Indoor positioning using fake BLE beacons

Autor: Albert Viaplana Ventura

Director: Dr. Lluís Garrido
Dr. Eloi Puertas i Prats

Realitzat a: Departament de Matematica i Informatica

Barcelona, 30 de juny de 2016
# Contents

## Introduction

1. **Motivation** .................................................. 1
2. **Objectives** .................................................. 1
3. **Structure of the project** .................................... 2

## State of the art

1. **What is a beacon?** .......................................... 3
2. **Beacon technologies** ........................................ 3
   - **WLAN** .................................................. 3
   - **Bluetooth** ............................................... 3
   - **Radio Frequency IDentification (RFID)** .................. 4
   - **Ultrasounds** ............................................ 4
3. **Filtering** .................................................... 4
   - **Simple Exponential Smoothing** .......................... 5
   - **Mean** .................................................... 5
4. **Measuring the distance** .................................... 5
   - **Time of Flight** .......................................... 5
   - **Received Signal Strength Indication** .................... 5
   - **Angle of Arrival (AOA)** ................................ 6
5. **Positioning techniques** ..................................... 6
   - **Lateration** ............................................... 6
     - **Least Mean Squares (LMS)** ............................ 7
   - **Triangulation** ........................................... 8
   - **Fingerprinting** .......................................... 8
   - **Particle Filter** .......................................... 8

## Design

1. **General structure** .......................................... 9
2. **The receiver** ............................................... 10
3. **The beacons** ................................................ 10
   - **Commercial beacons** .................................... 10
     - **Esmote** ............................................... 10

Chapter 1

Introduction

1.1 Motivation

GPS is a widely used technology, and its has become an essential. It has a lot of applications; from navigations to tracking of protected species. Nowadays, the majority of the population spends their time indoors, and this technology can’t be used in such circumstances, because the receiver needs a clear line of sight to the satellites.

In the last few years, the usage of beacons\(^1\) has been a trending topic, specially since the Bluetooth low energy (BLE) technology came out. A lot of start-up beacon-focused has been created, and big companies like Apple or Google have been investigating and developing their protocols based on such technology. There’s lots of applications for those beacons, like indoor navigation systems, or providing smart signs, which could be placed in a museum or supermarket, and as the user approaches, it would give detailed information of a painting, or special promotions for a product.

1.2 Objectives

With the rising market of the BLE, we wanted to explore how it could be used for localization purposes, which could eventually be applied to study how people moves in a building, for indoor maps, or for a robotics project.

The main goal of the project is to create an application that can localize a subject in an indoor environment, using BLE beacons as reference points. Our attempt is to achieve this goal by using beacons that will be built by us, which will use a radio frequency module that is not compatible with the BLE standard, but it is capable of “faking” certain type of BLE packets. This will give us a good knowledge about the BLE technology, get some experience working with beacons, and understand the location methods.

\(^1\)A device that sends a continuous or periodic signal
1.3 Structure of the project

This document will explain the whole system, starting from the definition of beacon, and the analysis of some academic papers, as well as commercial products; which technologies, noise filters, information, and localization methods have been used.

Once we have a general idea of what can be used, the design of our approach is described. It specifies the design of beacons, including which parts are they made of and which technologies are used, and the design of the receiver, which device and environment will be used.

Later on, it describes the implementation, explaining how are the beacons built, and how they can fake Bluetooth LE advertisement packets. The application is described as well, explaining how the positioning algorithm works, and which features it has.

Finally, some experiments and tests are illustrated, concluding with the conclusions of the project.
Chapter 2

State of the art

The goal of this project is to create a low cost system that will be able to locate a subject in a indoor environment, in real time. In this chapter a global view of the methods and technologies that can used will be given. It is divided in four parts; first, the concept of beacon will be described. Then, some technologies that a beacon can use, followed by two filtering techniques. Later on, the extraction of information from the beacon signal, and, finally some of the localization methodologies.

2.1 What is a beacon?

A beacon is a transmitter at a known location, which emits continuous or periodic radio signal with limited information content, for instance its identification or location. To send such signal, any radio technology can be used.

2.2 Beacon technologies

In this section will be discussing some of the technologies that could be used, in order to send and receive signals.

2.2.1 WLAN

Wireless local area network allows to connect a device to the local area network, following the IEEE 802.11 standard. Some companies like Accuware [8], have products that uses this technology to position an individual in a indoor environment, based on the Received Signal Strength Indication (RSSI2.4.2) of the signal.

2.2.2 Bluetooth

Bluetooth is a non-proprietary standard for wireless digital data communication over short distances, and its aim is to connect virtually every type of electronic device, like keyboards, headphones, etc. wirelessly. It uses frequencies from 2.4 to 2.485 GHz.
There are different variants, the most popular is the 2.1, with a maximum data rate of 1 Mbps. It was introduced in 2007, and it made it easier and more reliable to pair devices. With the Enhanced Data Rate (EDR) feature, which was introduced with the 2.0 variant, could increase the data transfer rate to a maximum of 3 Mbps, although the practical rate was 2.1 Mbps.

By 2010, the variant 4.0 was introduced, which included the classic Bluetooth, Bluetooth high speed and Bluetooth low energy (BLE) protocols. The Bluetooth low energy protocol is aimed to run at very low applications, running off a coin cell.

Since BLE was introduced, lots of companies build beacons that were using such technology. Apple introduced the iBeacon protocol / firmware. That protocol allows a iBeacon enabled device to wake up relevant apps (even if the app is closed) when it identifies a beacon. In 2015 Google announced Eddystone, another protocol / firmware. They are both Android and iOS compatible, the main difference among them is that the Eddystone can transmit three different frame types:
- Eddystone-URL, which is a beacon format for the physical web.
- Eddystone-UID, for apps. It is very similar to iBeacon.
- Eddystone-TLM, also known as Telemetry, bridges both app and browser users.

### 2.2.3 Radio Frequency IDentification (RFID)

It is a technology that can identify an object that is carrying a tag, using radio-frequency electromagnetic fields when they come close to a reader. It is considered as an alternative to the bar-code. There’s 2 types of RFID tags:
- The passive ones, which has no internal power supply, being powered by the electromagnetic energy transmitted from the reader. Short read range
- The active ones, with internal power supply. Those tags are constantly broadcasting their own signal. They have much longer read range.

It has been used for indoor positioning purposes, using passive tags [2] or active ones [3], again, using the RSSI to calculate the distance.

### 2.2.4 Ultrasounds

Ultrasounds are high frequency, inaudible sound waves. Those sound waves can be used to measure distances, by sending a pulse, capturing it within a window of time, and then calculate the distance based on the elapsed time. The Bat Ultrasonic Location System[4] uses ultrasounds to trilaterate the signals, in order to locate an object.

### 2.3 Filtering

It is very common to have noise in the received signals. In a building, most of the wireless devices are using the 2.4GHz band, which is used by a lot of technologies such as WLAN networks, car alarms, Bluetooth, microwave ovens...
The 2.4GHz band is very susceptible to factors that interfere with the signal, such as interferences, multipath propagations, absorptions... leading to an instability. To get an accurate localization, a stable signal is a must, then the signals have to be filtered.

### 2.3.1 Simple Exponential Smoothing

It acts as a Low-pass filter, emphasising recent events, smoothing out high frequency variations and revealing long term trends.

\[
s_0 = x_0 \\
n_{t} = \alpha x_t + (1 - \alpha)n_{t-1}, \quad t > 0
\]  

Where \(x\) is the received RSSI, \(s\) is the averaged value and \(\alpha\) is the smoothing constant, which is within the range from 0 to 1. If the smoothing constant is greater, more importance is given to the recent observations, and consequently more noise is included. If it’s smaller, more importance is given to the older observations, which can lead to improper reflection of the current state.

### 2.3.2 Mean

Consists in calculating the average of the received signals. Calculate the mean of the samples taken in a small period of time.

\[
s = \frac{x_1 + x_2 + ... + x_n}{n}
\]

Where \(x\) is a RSSI value.

### 2.4 Measuring the distance

In order to be able to estimate a position, relevant information has to be extracted from the received signals, for example the distance or the direction to the transmitter. Some of them will be discussed in this section.

#### 2.4.1 Time of Flight

Knowing the elapsed time from sending a packet until it is received, the distance can be calculated. It requires very precise clocks, as well as a perfect synchronization among the beacons and the receiver, in order to calculate elapsed time.

#### 2.4.2 Received Signal Strength Indication

Indicator of the power level being received by an antenna. The log-distance path loss model [1] is used to determine the distance among the receiver and the beacon. It is based on the relationship between the received power and the distance:

\[
p_i = p_0 - 10n\log_{10}\left(\frac{d}{d_0}\right)
\]
Where $p_i$ is the received power in dBm, $n$ is the path loss exponent (environment dependent), $p_0$ is the average power at a reference distance $d_0$, and $d$ is the distance from where $p_i$ was taken.

Knowing the power $p_i$ received at a distance $d$, and the power $p_0$ at the reference distance $d_0$, the path loss exponent can be calculated:

$$n = \frac{p_0 - p_i}{10 \log_{10} \left( \frac{d}{d_0} \right)}$$  

(2.4)

Finally, using the following expression, the distance can be estimated:

$$d = d_0 10^{\frac{p_0 - p_i}{10n}}$$  

(2.5)

(in cm if $d_0$ is in cm)

### 2.4.3 Angle of Arrival (AOA)

This method can determine the direction of propagation of a radio frequency wave incident on an antenna array, by measuring the time difference of arrival at individual elements of the array. Knowing those delays, ToA is used in the Triangulation based systems, like cellphone geolocation.

### 2.5 Positioning techniques

In this section positioning techniques will be discussed. Those techniques can be used once the information from the signals is extracted.

#### 2.5.1 Lateration

This is a widely used technique, used in technologies like GPS. It requires a minimum of 3 beacons (trilateration). Per each beacon, it draws a circle with center at the location of the known beacon, and with a radii equal to the distance among the beacon and the receiver. The unknown location of the receiver is the point where all the circles intersect.

Knowing the basic formula of a circle:

$$d^2 = x^2 + y^2$$  

(2.6)
2.5 Positioning techniques

Considering that the circle \( a \) is centred at a point \((x_a, y_a)\), each component of the point has to be subtracted to the corresponding component of the unknown position of the receiver, in order to get the length from both points. Now the previous formula can be expressed as:

\[
d_a^2 = (x - x_a)^2 + (y - y_a)^2
\]

(2.7)

Following the previous procedure, three equations are obtained:

\[
d_a^2 = ((x - x_a)^2 + (y - y_a)^2)
\]

(2.8)

\[
d_b^2 = ((x - x_b)^2 + (y - y_b)^2)
\]

(2.9)

\[
d_c^2 = ((x - x_c)^2 + (y - y_c)^2)
\]

(2.10)

The squares can be eliminated:

\[
d_a^2 = x^2 + x_a^2 - 2 * x * x_a + y^2 + y_a^2 - 2 * y * y_a
\]

(2.11)

\[
d_b^2 = x^2 + x_b^2 - 2 * x * x_b + y^2 + y_b^2 - 2 * y * y_b
\]

(2.12)

\[
d_c^2 = x^2 + x_c^2 - 2 * x * x_c + y^2 + y_c^2 - 2 * y * y_c
\]

(2.13)

Subtracting the first equation from the second:

\[
d_b^2 - d_a^2 = 2x(x_a - x_b) + x_a^2 - x_b^2 + 2y(y_a - y_b) + y_a^2 - y_b^2
\]

(2.14)

Likewise, the third from the second:

\[
d_c^2 - d_a^2 = 2x(x_a - x_c) + x_a^2 - x_c^2 + 2y(y_a - y_c) + y_a^2 - y_c^2
\]

(2.15)

Rearranging the equations to produce a variable to simplify the final equation:

\[
x(x_b - x_a) - y(y_b - y_a) = \frac{(x_a^2 - x_b^2) + (y_a^2 - y_b^2) + (d_b^2 - d_a^2)}{2} = V_a
\]

(2.16)

\[
x(x_b - x_c) - y(y_b - y_c) = \frac{(x_c^2 - x_b^2) + (y_c^2 - y_b^2) + (d_c^2 - d_b^2)}{2} = V_b
\]

(2.17)

To get the component \( x \) of the intersection point:

\[
y = \frac{V_a(x_b - x_c) - V_b(x_b - x_a)}{(y_a - y_b)(x_b - x_c) - (y_c - y_b)(x_b - x_c)}
\]

(2.18)

\[
x = \frac{y(y_a - y_b) - V_b}{(x_b - x_c)}
\]

(2.19)

If there’s no intersection point with the circles, there will be no real solution for the equation. If unstable signals are used, this is very likely to happen.

2.5.1.1 Least Mean Squares (LMS)

Least Mean Squares can approximate a solution of the equation when there’s no intersection. The LMS technique gives the closest point to the intersection. Ndubueze Chuku, Amitangshu Pal and Asis Nasipuri wrote a paper [6] where they used those techniques for positioning purposes.
2.5.2 Triangulation

Knowing the position of the beacons, and the angles from the receiver to each of the transmitter, the position of such receiver can be calculated [5]. With this method, Thales de Miletus measured the radius of the Earth’s orbit around the Sun.f.

2.5.3 Fingerprinting

This method consists in dividing the map in a grid, and each cell will represent an area. That grid is filled with the probability distribution of the characteristic attributes (like a human fingerprint), for instance RSSI values of each beacon. This is known as the calibration phase or offline stage. Once all the areas are described, the position can be estimated by comparing the values obtained with the areas described in the previous stage. An example can be seen in this paper [7], where the fingerprinting method with the RSSI of WLAN beacons, obtaining good results.

2.5.4 Particle Filter

The particle filter (also known as sequential Monte Carlo) is a technique for implementing recursive Bayesian filtering by Monte Carlo sampling. It consist in discretizing the problem into individual particles, each of which will represent one possible state of the model. The system should have enough particles to cover all kind of probability distribution, and any kind of evidence. The more particles the system has, the more accurate the result can be.

As explained in those articles [12], [15], [14] and [13], this method uses a nonparametric Bayesian Filter, which consists of specifying a prior probability distribution for an unknown parameter.

The filter is divided in 4 stages:

- Compute weights: A weight is assigned to each particle. The weight will always be greater than 0, so all the particles from the system will have at least a small chance to be chosen in the resampling stage. The sum of all the weights is 1.
- Weight normalization.
- Resample: A new set of particles is created, based on the old one. Each new particle is randomly chosen from the old one, having more chances if the value of the weight is greater.
- Prediction: The position of each particle from the new set is sightly modified, in order to cover the possible positions where the moving subject could be.
Chapter 3

Design

In this chapter the design of the system will be discussed, starting from the general structure, the characteristics of the receiver, the technologies, methods and parts of the beacons and a general description of the application.

3.1 General structure

The system will consist in a minimum of 3 of beacons distributed all over the area to cover. Those beacons will periodically broadcast a signal which will be captured by a receiver, and then it will calculate the current position.

![3 beacons placed in the UB hall, covering all the area.](image)

Figure 3.1: 3 beacons placed in the UB hall, covering all the area.

It is very important that a minimum of three beacons can be reached from any place in the area.
3.2 The receiver

A cellphone will be used as a receiver. It suits perfectly the needs for this project: the majority of the population already has one, it has great computation power, has a bunch of sensors that can be very useful for location purposes, and it’s easy to develop applications for it. In this case, I will be using a Sony Xperia Z3, running Android N DP3.

3.3 The beacons

The chosen technology for the beacons is BLE for various reasons:
- Can use a cellphone as a receiver.
- It is meant to have a super low power consumption.
- It is a cheap technology.
- Can send advertisement packets, which doesn’t require a paring process in order to capture them.

Such beacons will be continuously sending advertisement packets. An existing WLAN network could be used as well, but, because the system requires at least the signal of 3 beacons at all time, more AP would be required, increasing the cost of the project.

3.3.1 Commercial beacons

There are companies that are selling BLE beacons. The beacon market is growing really fast, in fact, it is said that the beacons are the future of costumer engagement. It is estimated that the market will double in 2016, and 400 million beacons will be shipped by 2019\(^1\).

3.3.1.1 Esmote

Esmote\(^2\) is one of the companies that are selling beacons:

\(^{1}\)http://www.nfcworld.com/2016/05/12/344713/ble-beacon-market-double-2016/  
\(^{2}\)http://estimote.com/
3.3 The beacons

For 59 USD (53.29 EUR) they offer a pack of 3 beacons, with a range up to 70 meters, with an ARM M0 Cortex, a battery life of 3 years (with a maximum of 5+ years), and integrated motion and temperature sensors. If they want to be used for indoor location, it is said in the website that two packs of 3 beacons are required, increasing the cost to 118 USD (106.57 EUR).

3.3.1.2 Unbranded alternatives

Unbranded alternatives can be found in online shops like Aliexpress, like this one:

This device is powered by an ARM M0 Cortex as well, combined with a Nordic NRF51822 chip, which gives it BLE compatibility. The price also includes a button battery. It has a price of 7,36 EUR (8,87 EUR including shipping).

Another option is to build our own beacons, using an Arduino.
3.3.2 Home made beacon using Arduino

Arduino is an easy to use open-source platform used to build electronics projects. It consists of two parts:

- The hardware, a programmable microcontroller.
- The Integrated Development Environment (IDE), which is used write and upload the to the board.

Arduino uses as simplified version of C++. For this project, the Pro Mini board is used, because it is the smallest and cheapest one. It is powered by an ATmega328 chip, and it can run at 3.3v and 8MHz, decreasing the power consumption.

![Arduino Pro Mini](image)

Figure 3.4: Arduino Pro Mini

The main specifications of the board are:

- 8 analog inputs, with a resolution of 10 bits. Two of those pins can be used for I²C communications (A4 for SDA and A5 for SCL).
- 14 digital input / output pins, each of them can provide a maximum of 40 mA, and has an internal pull-up resistor of 20-50 kOhms. Some of the pins have specialized functions:
  - Pin 0 and 1 can be used for UART TTL serial communications (0 for RX and 1 for TX).
  - Pin 2 and 3 can be configured to trigger an interruption.
  - Pins 3, 5, 6, 9, 10 and 11 can provide 8-bit pulse-width modulation (PWM).
  - Pin 13 has a LED connected.
  - Pins 10, 11,12 and 13 support SPI communication (10 for SS, 11 for MOSI, 12 for MISO and 13 for SCK).
- 32 kB for flash memory.
- 1 kB of EEPROM
- 2kB of SRAM
3.3 The beacons

- A clock speed of 8 MHz

Arduino-compatible boards are is being sold on Aliexpress as well. Those boards are basically a rip-off of the original Arduino:

![Figure 3.5: Arduino Pro Mini in Aliexpress](image)

For just 1,57 EUR (1,87 EUR including shipping). The microcontroller itself can’t send BLE packets, it needs a transmitter.

3.3.2.1 BLE Module

There are plenty of BLE modules that are compatible with the Arduino board. The cheapest one that is being sold by a trusty Aliexpress seller, is this one:

![Figure 3.6: Arduino compatible BLE module, in Aliexpress](image)

It has a CC2549 chip, and has a range up to 60 meters. Can be purchased for 3 EUR shipping costs included. //

Those could be used in this project, but I decided to build my own ones for a fraction of the price. Those home-made beacons will use a nRF24L01+ transmitter, controlled by an Arduino pro mini.

3.3.2.2 nRF24L01

Yet another option is to use a nRF24L01 transceiver. It’s a low cost, a ultra low power 2Mbps RF transceiver IC designed for operation in the world wide ISM frequency band at 2.400 - 24835GHz. It is made by Nordic Semiconductor. The IC integrates a RF transceiver, RF synthesizer, and an embedded baseband protocol engine (Enhanced ShockBurst) hardware protocol accelerator supporting a high-speed SPI interface for the application controller.
Variations

**nRF24L01**  This is the early version of the chip. It supports data rates of 1Mbps and 2Mbps. The transmission range is up to 70 meters at 1Mbps. Sold for 1 EUR, including shipping.

![nRF24L01 module, in eBay](image)

**Figure 3.7: nRF24L01 module, in eBay**

**nRF24L01+**  This is the updated version of chip, is fully compatible with the old one, and it supports a data rate of 250kbps, which can be used to extend the broadcast length. The transmission range is up to 100 meters at 256kbps. For an unbeatable price of 0,62 EUR shipping costs included.

![nRF24L01+ module, in Aliexpress](image)

**Figure 3.8: nRF24L01+ module, in Aliexpress**

**nRF24L01+ PA LNA**  It has the same chip, but it includes a power amplifier and a low noise amplifier. The transmission range is up to 1000 meters with a 2DB antenna, transmitting at 250kbps. Sold for 1,92 EUR including shipping and the antenna.
3.3 The beacons

3.3.2.3 The battery

The Arduino Pro Mini requires the input voltage to be within a range of 3.35 and 12 volts, which makes it perfect to be powered by a small one cell Lithium polymer battery, which voltage varies from 4.2 volts (fully charged) to 3.0 volts. This kind of batteries can be found in Aliexpress as well, like this one, for €2.31 EUR including shipping.

3.3.2.4 Total cost

For 7.2 EUR is possible to build a home made beacon, combining an Arduino, BLE module and a battery. It would have a range up to 60 meters.

On the other hand, if the nRF24L01+ module is used, it would have a range up to 100 meters, and the price would drop to 4.79 EUR.

The price would raise to 6.11 EUR if the long-range version of the nRF24L01+ is used, but the range would be 10 times greater, which translates to a more stable signal.

3.3.3 The chosen setup

I decided to build my own beacons, using an Arduino, as explained above. Since the range is much better using the nRF24L01+, and it only costs a fraction of the price, I decided to build the beacons using such modules. A second version using the long range version of the nRF24L01+ will be build as well.
3.4 The application

It will be an Android app, which will be capturing the advertisement packets from the known beacons, and applying an exponential smoothing filter to reduce the noise from the received RSSI. Periodically it will check if it has gathered enough packets to estimate the position. If so, the location will be calculated depending on the selected positioning technique, which could be lateration or particle filter.

3.5 Planning

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>80</td>
<td>Get an idea of the previous work done, read academic papers and understand them. Investigate which technologies and methods could be used</td>
</tr>
</tbody>
</table>

Analysis and design

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial systems</td>
<td>6</td>
<td>Search and study commercial location systems.</td>
</tr>
<tr>
<td>Alternative systems</td>
<td>6</td>
<td>Investigate if there are alternatives to the commercial ones. What could be used to build our own system.</td>
</tr>
<tr>
<td>Design</td>
<td>20</td>
<td>Design the environment, decide what to use and how to use it.</td>
</tr>
</tbody>
</table>

Implementation

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beacon software</td>
<td>10</td>
<td>Program the beacons to make them work as our system requires.</td>
</tr>
<tr>
<td>Android application</td>
<td>60</td>
<td>Write the Android application.</td>
</tr>
<tr>
<td>Test environment</td>
<td>15</td>
<td>Develop an application that can log the signals of the beacons. Develop a program to apply the filters to the logs, and plot the results for future analysis.</td>
</tr>
</tbody>
</table>

Installation and testing
### 3.5 Planning

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test the beacons</td>
<td>20</td>
<td>Test the different versions of the beacons in different environments.</td>
</tr>
<tr>
<td>General test</td>
<td>5</td>
<td>Test the Android application, look for bugs. Physical inspection of the boards.</td>
</tr>
<tr>
<td>Test the system</td>
<td>7</td>
<td>Install the system in a real environment, and perform tests to see if it works as expected.</td>
</tr>
</tbody>
</table>

#### Documentation

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write the documentation</td>
<td>100</td>
<td>Writing of this documentation</td>
</tr>
</tbody>
</table>
Chapter 4

Implementation

4.1 The beacons

The beacons are composed by two main parts: a microcontroller and the transmitter. The microcontroller tells the transmitter how it has to send the data, and the transistor sends it. In this section will be explained the components that the beacons are made of, and the software that makes it possible to simulate BLE advertisement packets.

4.1.1 Hardware

The nRF24L01 module is driven by an Arduino Pro Mini (3.3v version). The Arduino Pro Mini requires the input voltage to be within a range of 3.35 and 12 volts, which makes it perfect to be powered by a small one cell Lithium polymer battery, which voltage varies from 4.2 volts (fully charged) to 3.0 volts.

The transceiver has an operating voltage range of 1.9 to 3.6v, which means that it can’t be hooked directly to the battery. Instead, it will be connected to the VCC pin of the Arduino, which has a regulated power output of 3.3 volts. But connecting the nRF24L01 directly to the pin might lead to poor signal, and high packet loss rate, because in certain circumstances the regulator is not powerful enough. The nRF24L01 is a low consumption transceiver, but it has short peaks of high consumption when it starts a transmission, and the integrated regulator of the Arduino can’t supply such demand of power. To solve that, a capacitor is placed right before the nRF24L01 power input.
Figure 4.1: Beacon scheme

Figure 4.2: The PCB board
4.1 The beacons

(a) Arduino Pro Mini with a nRF24L01+

(b) Arduino Pro Mini with an amplified version of the nRF24L01+

Figure 4.3: Final version of the beacons.
4.1.2 Software

As Dimitry Grinberg explained in his article [9], it’s possible fake a Bluetooth LE advertisement packets using a cheap nRF24L01 module.

4.1.2.1 BLE Advertisement packet structure

The advertisement packet has following structure:

<table>
<thead>
<tr>
<th>Preamble 1 Byte</th>
<th>Access address 4 Bytes</th>
<th>Packet Data Unit (PDU) 6–39 Bytes</th>
<th>CRC 3 Bytes</th>
</tr>
</thead>
</table>

Figure 4.4: Structure of the advertisement packets.

Where the preamble field is either 10101010 or 01010101, and the access address field is always 0x8E89BED6.

The PDU is composed by:

<table>
<thead>
<tr>
<th>Header 2 Bytes</th>
<th>Payload 6–37 Bytes</th>
</tr>
</thead>
</table>

Figure 4.5: Detailed packet data unit section.

And the header is also broken down, in 6 parts:

<table>
<thead>
<tr>
<th>PDU type 4 bits</th>
<th>RFU 2 bit</th>
<th>TxAdd 1 bit</th>
<th>RxAdd 1 bit</th>
<th>Length 6 bits</th>
<th>RFU 2 bit</th>
</tr>
</thead>
</table>

Figure 4.6: Parts of the packet data unit’s header.

RFU stands for reserved for future use. The PDU Type can be:

- `ADV_IND (0000)`: Connectable unidirected advertising event
- `ADV_NONCONN_IND (0010)`: Non-connectable unidirected advertising event
- `ADV_SCAN_IND (0110)`: Scannable unidirected advertising event. Can provide additional information by scan responses. `ADV_NONCONN_IND` is used.

4.1.2.2 nRF24L01 packet structure

The device supports two packet structures: the Enhanced ShockBurst and ShockBurst. The difference among them, is that the Enhanced version uses a 9 bit Packet Control Field between the address field and the payload.

- Shockburst
- Enhanced ShockBurst

In this new structure, the CRC field is mandatory, and a new Payload Control Field is introduced. It is composed of 3 fields:
4.1 The beacons

4.1.2.3 From ShockBurst to BLE advertisement

As we can see, both BLE Advertisement and the nRF24L01’s structures are very similar, but, some problems must be fixed to achieve our goal:

1. The data whitening. BLE specifications requires that data whitening must be applied to the address and the payload.

2. The CRC checksum. The device is only capable of generating a 16-bit CRC checksum of the payload, and without data whitening. BLE requires a 24-bit CRC checksum with whitening. Luckily that feature can be disabled, so it can be computed by the microcontroller.

3. The order in which the address and payload are sent. The tranceiver sends the most significant bit first, but the BLE specification requires the opposite order.

Those three problems must be solved by the microcontroller. It will have to generate the data whitening an the CRC checksum, and it will have to reverse the address and payload. An example of the implementation can be seen here [?].

---

[1] This algorithm "whitens" the data in such a way that the 1s and 0s will be better balanced.[16]
4.2 The Android application

4.2.1 Positioning

Each time an advertisement packet is received, it checks if the address corresponds to any of the beacons, and, if so, if there was no previous RSSI received from this beacon, the value is stored. If there were previous values stored, it filters out the "peak" noise by calculating the difference among the stored RSSI and the received one. If the received RSSI signal is stronger, and the difference is greater than a threshold, the sample is rejected. It could happen that the receiver gets a lot of noise, while the user is moving. In this case, most of the RSSI would be rejected, and, at one point, the stored value would correspond to a position that is far from where the user actually is. For that reason, the comparison with the stored value is only done if the elapsed time is not greater than 750ms. Once the "peak" noise is filtered, an exponential smoothing filter is applied to prevent the "smooth" noise.

A timer calls a function when a defined window of time is elapsed. This function checks if at least one packet from each of the beacons have been received, and if so, the approximated distances from each beacon are calculated, using the last filtered RSSI value from each beacon. The path loss model is applied, and the results are stored. Then, the positioning method is applied:

4.2.1.1 Using the particle filter:

First, N particles are distributed randomly over the map. Each particle represents a possible position where the user might be in. Then, each beacon is evaluated, depending on the number of packets that have been received:

\[ W_{b_i} = \frac{\text{received_packets}}{\text{window_size}} \cdot \frac{\text{packet_time}}{\text{frequency}} \] (4.1)

Where \text{window_size} corresponds to the period of time that the application will be waiting packets. The result of the equation will determine the assigned weight. The more packets the system has received, the more reliable the signal is, because less noise has been filtered out. Depending on the weight of the beacon, it will have more influence when the the position of the user is calculated.

Then the weights of the particles are assigned following this equation:

\[ W_{p_i} = \sum_{j=1}^{\text{num_beacons}} W_{pb_{ij}} \] (4.2)

Where

\[ W_{pb_{ij}} = \begin{cases} 0.01 & \text{if } \text{diff_dist} \geq \text{max_dist} \\ (1 - (\text{diff_dist})/\text{max_dist})W_{n_i} & \text{if } \text{diff_dist} < \text{max_dist} \end{cases} \] (4.3)

\(W_{n_i}\) is the weight of the beacon, and \text{diff_dist} is the difference among the beacon obtained with the path loss model and the position of the particle \(j\).

\(\text{diff_dist} = |\text{distance}_n_i - \text{pos}_\text{particle}|\)
Then, all the weights are normalized, by:

\[ \text{norm}_{\text{w}}_{p_i} = \frac{W_i}{\text{sum\_weights}} \]

Where \( \text{sum\_weights} \) is the sum of weights of all the particles. The position of the user will be equal to the position of the heaviest particle.

Once the position is known, the resampling stage starts. A new set of particles is created, based on the old one. The particles from the old set are chosen randomly, following an arbitrary probability distribution fashion [11].

Finally the position of the particles the new set is modified a within a range of 0 to 10%, preventing them to be outside the map or in a protected area (like in the pool).

![Initialization of the particles](image1.png)

![Localization of the subject](image2.png)

(a) Initialization of the particles  
(b) Localization of the subject

Figure 4.10: Screenshots of the application using the particle filter.

### 4.2.1.2 Using the lateration combined with the least mean squares

The position of the user is determined by computing the lateration and least mean squares method, explained in the sections ?? when this method is used, the particles are not shown.
4.2.2 Calibration

In order to calculate the exact value of the path loss exponent, the application has a calibration mode. This mode first asks the user to select the reference beacon. Then, it has to place himself at a distance of one meter from such beacon, and then start capturing packets for 15 seconds. Once done, the average of the RSSI will be assigned to the $d_0$ variable. Later on, the user would have to move farther, at a distance of ten meters from the beacon, and start capturing packets for another 15 seconds. Finally, the average is calculated, and the path loss exponent as well, as described in the section 2.4.2.
4.2 The Android application

(a) Beacon selection
(b) Instructions
(c) Capturing
(d) Instructions
(e) Process done

Figure 4.12: Calibration process
4.2.3 Settings

The user can adjust:

**Number of particles**
The number of particles used when the particle filter mode is selected.

**Window of time**
The elapsed time among location calculations, in milliseconds.

**Debug information**
The application has a debug mode that shows real time information about:

- **General parameters:**
  - Path loss exponential.
  - Average RSSI at a meter.
  - Window of time.
  - Number of particles.

- **Beacon information:**
  - The mean RSSI within the window of time.
  - The deviation of the signals.
  - The number of packets received.
  - The assigned weight.
4.2 The Android application

Log generation
If this option is checked, each time the window of time is elapsed, the computed locations along with the number of packets received and the last RSSI value of each node will be written in a log file.

Positioning method
The positioning method that is used. Two choices available: Particle filter and iteration + least mean squares.
Chapter 5

Experiments and results

In this chapter will be explained the series of tests that were done to analyze if all the different parts of the system were working as expected. Later on, an updated planning of the project will be given.

5.0.1 Tests

A series of experiments were done to test the performance of the system. First, the received RSSI values were analyzed. In the following graphs we can see the received RSSI values of a non amplified beacon from different distances:
In the last graph we can see that the RSSI values are really low, and 54% of the packets are lost. This means that such beacons could only be used for localization in small rooms. The same tests were done using the amplified version.
(a) Received RSSI at 4 meters

(b) Received RSSI at 8 meters

(c) Received RSSI at 10 meters

(d) Received RSSI at 25 meters

Figure 5.2: RSSI logs using an amplified version of the nRF24L01+ transceiver
At a distance of 25 meters, still good values are received, but it seems to be the limit. If there’s more distance, the RSSI values are too weak, and more packets are lost. Observing at those graphs, we can deduce that:

- The noise is proportional to the distance; it increments when there’s more distance among receiver and transmitter.

- There’s two types of noise: the peak noise, and the smooth noise.

The peak noise is always incrementing the power of the signal, and it does it by a minimum of 18 dBm. It can be prevented by comparing the last valid signal captured, with the current one, and if the difference is greater than 18, we consider the signal as noise, rejecting it. Here are the results applying this technique:

(a) Received RSSI at 4 meters, "peak" noise filtered.

(b) Received RSSI at 8 meters, "peak" noise filtered.

Figure 5.3: RSSI logs filtering the "peak" noise, using a nRF24L01+ transceiver
(a) Received RSSI at 4 meters, "peak" noise filtered.

(b) Received RSSI at 8 meters, "peak" noise filtered.

Figure 5.4: RSSI logs filtering the "peak" noise, using an amplified version of the nRF24L01+ transceiver

With this simple filter, much better results are obtained. Now there’s just the “smooth” noise remaining, which is more difficult to eliminate. To obtain a steadier signal, the exponential smoothing filter is applied. Such filter has a smoothing constant, which is within the range of 0 to 1. If it is closer to one, the recent observation will have more importance. In the next graph we can see how the result varies depending on the value of the constant:
Figure 5.5: Influence of the smoothing constant.

If the subject is moving, we want the result to match the current location. For this reason, the smoothing constant should not be too low, because it would lead to results that might not correspond to the current location. On the other hand, it can’t be too high, because it would be more susceptible to noise. Good results are obtained with a smoothing constant of 0.2.

Results applying this filter to the previous logs:
(a) Received RSSI at 4 meters, both filters applied.

(b) Received RSSI at 8 meters, both filters applied.

(c) Received RSSI at 10 meters, both filters applied.

Figure 5.6: RSSI logs filtering both "peak" and "smooth" noise, using a nRF24L01+ transceiver
38 Experiments and results

(a) Received RSSI at 4 meters, both filters applied.

(b) Received RSSI at 8 meters, both filters applied.

(c) Received RSSI at 10 meters, both filters applied.

(d) Received RSSI at 25 meters, both filters applied.

Figure 5.7: RSSI logs filtering both "peak" and "smooth" noise, using an amplified version of the nRF24L01+ transceiver
Later on, the positioning methods were tested in a small room. In the next scheme we can see the dimensions of the room, where the beacons were placed, where the measurements were taken from.

![Figure 5.8: Testing environment.](image)

**Results obtained with the beacon that uses the nRF24L01+ module:**

**Particle filter method:**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Computed X</th>
<th>Computed Y</th>
<th>Error X</th>
<th>Error X</th>
<th>Total Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>0.95</td>
<td>6.55</td>
<td>0.95</td>
<td>1.45</td>
<td>2.4</td>
</tr>
<tr>
<td>4.8</td>
<td>0</td>
<td>3.34</td>
<td>3.00</td>
<td>1.46</td>
<td>3.0</td>
<td>4.46</td>
</tr>
<tr>
<td>4.8</td>
<td>8</td>
<td>3.05</td>
<td>4.00</td>
<td>1.75</td>
<td>4.00</td>
<td>5.75</td>
</tr>
<tr>
<td>2.4</td>
<td>4</td>
<td>2.50</td>
<td>2.47</td>
<td>0.1</td>
<td>1.53</td>
<td>1.63</td>
</tr>
</tbody>
</table>

**Lateration and least mean squares method.**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Computed X</th>
<th>Computed Y</th>
<th>Error X</th>
<th>Error X</th>
<th>Total Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>-1.23</td>
<td>9.40</td>
<td>1.23</td>
<td>1.4</td>
<td>2.63</td>
</tr>
<tr>
<td>2.4</td>
<td>4</td>
<td>1.99</td>
<td>2.69</td>
<td>0.41</td>
<td>1.31</td>
<td>1.72</td>
</tr>
<tr>
<td>4.8</td>
<td>0</td>
<td>-1.54</td>
<td>-1.62</td>
<td>6.34</td>
<td>1.62</td>
<td>7.96</td>
</tr>
<tr>
<td>4.8</td>
<td>8</td>
<td>3.35</td>
<td>6.23</td>
<td>1.45</td>
<td>1.77</td>
<td>3.22</td>
</tr>
</tbody>
</table>
Resuts obtained with the beacon that uses the amplified nRF24L01+ module:

**Particle filter method:**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Computed X</th>
<th>Computed Y</th>
<th>Error X</th>
<th>Error X</th>
<th>Total Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>1.77</td>
<td>5.93</td>
<td>1.77</td>
<td>2.07</td>
<td>3.84</td>
</tr>
<tr>
<td>2.4</td>
<td>4</td>
<td>3.63</td>
<td>3.35</td>
<td>1.23</td>
<td>0.65</td>
<td>1.88</td>
</tr>
<tr>
<td>4.8</td>
<td>0</td>
<td>1.51</td>
<td>4.13</td>
<td>3.29</td>
<td>4.13</td>
<td>7.42</td>
</tr>
<tr>
<td>4.8</td>
<td>8</td>
<td>1.16</td>
<td>4.53</td>
<td>3.64</td>
<td>3.37</td>
<td>7.11</td>
</tr>
</tbody>
</table>

**Lateration and least mean squares method.**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Computed X</th>
<th>Computed Y</th>
<th>Error X</th>
<th>Error X</th>
<th>Total Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>4.67</td>
<td>11.33</td>
<td>4.67</td>
<td>3.33</td>
<td>8</td>
</tr>
<tr>
<td>4.8</td>
<td>0</td>
<td>-26.98</td>
<td>-10.31</td>
<td>31.78</td>
<td>10.31</td>
<td>42.09</td>
</tr>
<tr>
<td>4.8</td>
<td>8</td>
<td>-35.71</td>
<td>5.83</td>
<td>40.51</td>
<td>2.17</td>
<td>42.68</td>
</tr>
<tr>
<td>2.4</td>
<td>4</td>
<td>23.00</td>
<td>25.90</td>
<td>20.6</td>
<td>21.9</td>
<td>42.5</td>
</tr>
</tbody>
</table>

Finally, the system was tested while the user was moving in a larger environment. First, the subject walked from one end to the other, capturing the RSSI from two beacons that were placed as seen in the scheme:

![Figure 5.9: Scheme illustrating the performed test.](image)

The values were filtered using the same filters as in the other tests, obtaining the following results:
(a) Results obtained using low range beacons.

(b) Results using the amplified beacons.

Figure 5.10: RSSI signals of the subject moving from one beacon to the other.

Again, the amplified beacon gave us the best results. As we can see in the next pictures, the faculty of Mathematics has a lot of obstacles in the way, such as columns, trees and people walking by. This might be the reason why we got such strange results with the amplified beacon. As we can see, the beacon with greater signal is the one that has more packets lost, when it should be the completely opposite.
42 Experiments and results

(a) Faculty of Mathematics.  
(b) Localization using the particle system, at the faculty of Mathematics.

Figure 5.11: RSSI signals of the subject moving from one beacon to the other.

5.1 Final planning

Once the project is done, the planning is updated with the real estimation of the dedicated hours.

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>100</td>
<td>Get an idea of the previous work done, read academic papers and understand them. Investigate which technologies and methods could be used</td>
</tr>
</tbody>
</table>

Analysis and design

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial systems</td>
<td>6</td>
<td>Search and study commercial location systems.</td>
</tr>
<tr>
<td>Alternative systems</td>
<td>8</td>
<td>Investigate if there are alternatives to the commercial ones. What could be used to build our own system.</td>
</tr>
<tr>
<td>Design</td>
<td>14</td>
<td>Design the environment, decide what to use and how to use it.</td>
</tr>
</tbody>
</table>

Implementation
## 5.1 Final planning

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beacon software</td>
<td>12</td>
<td>Program the beacons to make them work as our system requires.</td>
</tr>
<tr>
<td>Android application</td>
<td>72</td>
<td>Write the Android application.</td>
</tr>
<tr>
<td>Test environment</td>
<td>24</td>
<td>Develop an application that can log the signals of the beacons. Develop a program to apply the filters to the logs, and plot the results for future analysis.</td>
</tr>
</tbody>
</table>

### Installation and testing

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test the beacons</td>
<td>24</td>
<td>Test the different versions of the beacons in different environments.</td>
</tr>
<tr>
<td>General test</td>
<td>5</td>
<td>Test the Android application, look for bugs. Physical inspection of the boards.</td>
</tr>
<tr>
<td>Test the system</td>
<td>7</td>
<td>Install the system in a real environment, and perform tests to see if it works as expected.</td>
</tr>
</tbody>
</table>

### Documentation

<table>
<thead>
<tr>
<th>Task</th>
<th>Time (h)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write the documentation</td>
<td>120</td>
<td>Writing of this documentation</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusion and future work

In this chapter, the conclusions reached by the test of the system are exposed, and also future work is provided.

6.0.1 Conclusion

The goal of this project was to create a system that would be able to position a subject in an environment where the GPS is not accessible.

Two different types of beacons were built, one version used a nRF24L01+ module, and the other used an amplified version of the same module. We were able to fake the BLE advertisement packets, and the receiver was able to capture them.

After testing the beacons, we realized that the non-amplified module could not be used for large distances; the signal was not stable enough, and the received packet ratio was very poor. We got much better results with the second version, which was using the amplified module. The signal was much stronger, with an excellent received packet ratio.

For receiver part, an application was created, which was able to receive the packets, apply filters to the signals, and compute the location using two different methods; the particle filter and the lateration with least mean squares. A debug mode was created as well, to let the user know what was happening in the background, so it could have some feedback about the process.

Finally, the system was used in the faculty hall, but the results were not as good as expected. It was able to place the user in the map, but the results were not very accurate, and sometimes the location that was given was totally wrong. The reason of that was that the signals were too unstable, even after the calibration was performed. This was probably caused by the obstacles that were on the way, and maybe the WLAN network influenced as well. Using more beacons would have increased the accuracy, but due the lack of time, we were not able to build more beacons.
Despite the final results, we got to learn a lot about the technologies, how to work with them, and how to apply localization methods.

6.1 Future work

Several additions could be to this project if the time would have permitted:

- Use a technology that operates in a less saturated frequency, for instance WLAN 802.11n, which uses the 5GHz band.
- Improve the localization. It would be interesting to combine the BLE beacons with the WLAN network. This would reduce the noise caused by the interferences.
- Combine the signals of the beacons with the signals from the WLAN access points that are already installed in the faculty.
- Almost all the smartphones integrates a gyroscope, accelerometer and magnetometer. Using those sensors, the smartphones could be used as an inertial measurement unit (IMU). In this paper [17] a IMU is used for indoor positioning purposes. The usage of those sensors would increase the reliability of the system.
- Create a navigation application that could be use in a congress. The application could give information about the talks and events that are currently taking place. At the same time, the application could send the locations of the users to a server, in order to study how they move around, where are the spots where most of the people walked by...
Bibliography


[14] Miguel Martínez del Horno, 2015, Sistema de localización para entornos de interior basado en Android


[16] BLUETOOTH SPECIFICATION Version 1.0 B, section 7, Data Whitening http://grouper.ieee.org/groups/802/15/Bluetooth/core_10_b.pdf