ETH zürich

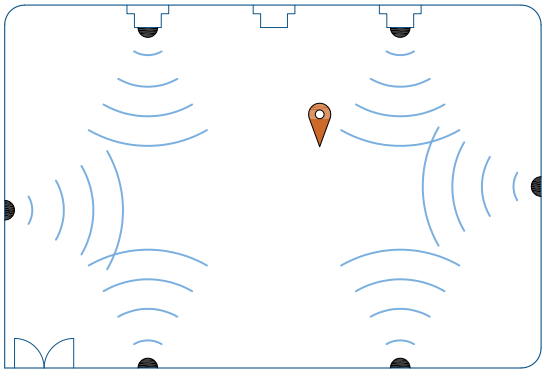
Institut für Integrierte Systeme
Integrated Systems Laboratory

DEPARTMENT OF
INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Spring Semester 2020

Bluetooth Low Energy Positioning on FPGA

Master Thesis



Jan Kreisinger
kreisja2@fel.cvut.cz

14. 8. 2020

Advisors: Mauro Salomon, msalomon@iis.ee.ethz.ch
Stefan Lippuner, lstefan@iis.ee.ethz.ch

Professors: Quiting Huang, huang@iis.ee.ethz.ch
Pavel Hazdra, hazdra@fel.cvut.cz

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kreisinger** Jméno: **Jan** Osobní číslo: **439601**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra mikroelektroniky**

Studijní program: **Elektronika a komunikace**

Specializace: **Elektronika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Implementace určování polohy využívající Bluetooth Low Energy v FPGA

Název diplomové práce anglicky:

Bluetooth Low Energy Positioning on FPGA

Pokyny pro vypracování:

1. Get familiar with the BLE PHY specification and identify the signal processing blocks required to detect and decode broadcasted packets. Understand GFSK modulation/demodulation and the architecture and interface of the Celeste core.
2. Get familiar with the Matlab simulation environment. Learn how to generate Bit Error Rate plots. Understand the implemented algorithms for preamble detection and GFSK demodulation. Learn how to generate test vectors for the Celeste core. Use the existing Matlab frame work to genearate a reference waveform.
3. Identify all the building blocks that will be integrated on the FPGA and understand how they will interface. Integrate the Celeste core into an existing system that includes a CPU and an RF controler block.
4. Adapt the Celeste core for FPGA synthesis. Synthesise the entire system and map it on an FPGA.
5. Develop the required software to interact with the Celeste core. Understand how to control the RF subsystem. Develop a program that detects and decodes BLE broadcasted packets. Implement RSSI measurements.
6. Test the implemented system and evaluate its performance with RF measurements. Use the developed RSSI measurement to implement BLE based positioning. Build a test set-up to measure the positioning performance.

Seznam doporučené literatury:

[1] N. Gupta. Inside Bluetooth Low Energy. Artech House Remote Sensing Library. Artech House, 2013.

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Pavel Hazdra, CSc., katedra mikroelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Prof. Dr. Qiuting Huang, Department of Information Technology and Electrical Engineering, ETH Zurich

Datum zadání diplomové práce: **28.01.2020** Termín odevzdání diplomové práce: **15.08.2020**

Platnost zadání diplomové práce: **30.09.2021**

prof. Ing. Pavel Hazdra, CSc. podpis vedoucí(ho) práce	prof. Ing. Pavel Hazdra, CSc. podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. Datum převzetí zadání

Podpis studenta

Declaration of Originality

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date ....

signature

Acknowledgments

I would like to sincerely thank to the project supervisors Mauro Salomon and Stefan Lippuner for their valuable advices and professional guidance throughout the whole project.

I would also like to thank to professor Quiting Huang for giving me the opportunity to come to ETH and carry out this project.

My thanks go to professor Pavel Hazdra as well for guarantying this project at the side of FEL CTU.

Abstract

Indoor positioning systems are becoming more and more popular. Bluetooth Low Energy is one of the most frequently used technologies within this field thanks to its simplicity, low energy consumption, and low costs. The goal of this project was to develop an indoor positioning testbed based on a previously designed Bluetooth digital baseband accelerator called Celeste. Celeste was modified within this project to provide better receiver characteristics. It was integrated to a RISC-V SoC and the whole system was implemented on an FPGA. EvaLTE board, which was previously developed at IIS, was used as an RF and analog subsystem. The function of the whole system was verified first with CMW500 RF tester and then with signals from commercial BLE advertisers. A positioning application based on the linear least squares algorithm was developed and evaluated in an experimental setup. The main issues regarding the position estimation accuracy identified during the application development were beacons in non-line-of-sight, an unstable RF environment, and differences between advertising channels. Evenly distributed measurements of signal strength over all the advertising channels were used to mitigate the effect of multipath propagation. The negative effects of the unstable RF environment and the beacons in non-line-of-sight were successfully attenuated by excluding measurements from the beacons with weak signal strength from the position calculation. This approach improved the position estimation accuracy by up to 60% comparing to the conventional approach of using measurements from all the available beacons. The achieved mean error in the position estimation was 2.75 m.

Abstrakt

Vnitřní lokalizační systémy se stávají čím dál populárnějšími. Bluetooth Low Energy je jednou z nejpoužívanějších technologií v tomto oboru díky své jednoduchosti, nízké energetické náročnosti a nízkým finančním nákladům. Cílem této práce je navrhnout lokalizační systém založený na dříve vyvinutém digitálním Bluetooth akcelerátoru zvaném Celeste. Tento akcelerátor slouží k detekci a demodulaci Bluetooth paketů v základním frekvenčním pásmu. Celeste bylo v rámci této práce modifikováno pro zlepšení jeho detekčních a demodulačních parametrů. Poté bylo integrováno do RISC-V SoC a celý tento systém byl implementován na FPGA. Jako RF a analogový subsystém byla použita deska EvaLTE, dříve vyvinutá na IIS. Funkce celého systému byla ověřena pomocí CMW500 RF testeru a také pomocí signálů z BLE advertiserů. Software pro lokalizaci založený na metodě nejmenších čtverců byl vyvinut a jeho funkce byla experimentálně ověřena. Největší výzvou pro přesnost odhadu pozice byla nestabilita RF prostředí, rozdíly mezi jednotlivými kanály určenými pro broadcast a BLE vysílače, které byly mimo přímou viditelnost. Pro potlačení rozdílů mezi kanály byla použita rovnoměrně rozložená měření síly signálu ve těchto kanálech. Z výpočtu pozice byla vyloučena měření z několika vysílačů pro potlačení negativních dopadů nestabilního RF prostředí a vysílačů mimo přímou viditelnost, a to ta s nejslabším signálem. Tento navržený přístup zlepšil přesnost lokalizace až o 60% ve srovnání s konvenčním přístupem, který využívá všechny dostupné vysílače. Dosažená průměrná chyba v odhadu pozice byla 2.75 m.

Contents

List of Figures

List of Tables

List of Acronyms

AA . . . . . .Access Address

APB . . . . . .Advanced Peripheral Bus

BER . . . . . .Bit Error Rate

BLE . . . . . .Bluetooth Low Energy

BLER . . . . . .Block Error Rate

CDF . . . . . .Cumulative Distribution Function

GFSK . . . . . .Gaussian Frequency Shift Keying

IPS . . . . . .Indoor Positioning System

LLR . . . . . .Log Likelihood Ratio

LLS . . . . . .Linear Least Square

LOS . . . . . .Line of Sight

NLOS . . . . . .Non-Line of Sight

NRZ . . . . . .Non-Return to Zero

PMR . . . . . .Packet Miss Rate

PULP . . . . . .Parallel Ultra Low Power

RSSI . . . . . .Received Signal Strength Indication

RTL . . . . . .Register Transfer Level

SNR . . . . . .Signal to Noise Ratio

Chapter 1

Introduction

The indoor positioning system (IPS) is a key technology of the Internet of Things (IoT) and Industry 4.0 applications. Based on thw IndustryARC report [1] the indoor positioning and navigation market is expected grow to $23.6 billion in 2023 compared to $6.92 billion in 2017.

The satellite navigation systems such as GPS or Galileo cannot be used for the indoor positioning due to high path loss of the satellite signal inside buildings. The indoor positioning application can be built on the current telecommunication technologies like Wi-Fi, RFID or Bluetooth Low Energy (BLE). Among these technologies BLE is especially attractive due to its simplicity, low-cost and low energy consumption.

The indoor positioning is used in many applications e.g. logistics, indoor navigation, localized product promotion, location based analytics etc. The proximity estimation using Bluetooth was the leading technology used in 2020 COVID-19 pandemic for contact tracking, while maintaining a high level of the personal data protection.

The Bluetooth communication standard was introduced in 1998 and BLE in 2010 as part of Bluetooth 4.0. Its newest version, which will be used throughout this thesis, is Bluetooth 5.2 released in the end of 2019. These standards are developed and maintained by the Bluetooth Special Interest Group (SIG). BLE is used for applications on battery supplied devices such as smart technologies and wearables requiring lower data rates with loose latency constraints. Thanks to that, the indoor positioning and navigation systems using BLE can reach a great number of users.

The position of a device can be estimated based on measurements from several reference beacons with known positions within the range of the device. The BLE beacons are continuously broadcasting a constant data on BLE advertising channels.

Based on the type of the measurements several methods of position estimation can be distinguished. Generally used methods are Received Signal Strength Indication (RSSI), Angle of Arrival (AoA), Time of Arrival (ToA) and Time Difference of Arrival (TDoA). The BLE technology usually uses the RSSI measurements for the position estimation, which is typically less accurate than the ToA measurements but simpler to implement [2]. The direction finding technology using AoA or Angle of Departure indication was

introduced in Bluetooth 5.1.

Fingerprinting and multilateration methods are the two major approaches used for the position estimation.

The position estimation using the fingerprinting method requires two steps. First, a large number of the RSSI measurements at known positions is collected over the area, where the IPS is to be implemented. These fingerprints are then stored in a memory of the device or on a server. Then, the application compares its own measurements with the fingerprints acquired in the previous step, and based on that, it estimates the position of the device [2]. This method gives very good results regarding the position accuracy, but it requires a large amount of the memory and taking fingerprints is very time consuming.

Multilateration method is based on the RSSI measurements and a path loss model of the area, where the IPS is implemented. The distances from the particular beacons can be calculated based on the measured RSSIs and the mathematical model of the RF environment. The distances from at least three different beacons must be known for the position estimation. The multilateration method is not as accurate as the fingerprinting, but it is very simple to implement and it does not require large datasets.

1.1. State of the Art of Bluetooth Positioning

The first attempts to use Bluetooth for the positioning were performed at the beginning of the last decade [3]. The reported mean distance error was 3.8 m. A significant development in this field came with the introduction of the BLE standard in 2010.

BLE is described as Wi-Fi competitor in "Meta-review of Indoor Positioning Systems" [4]. Comparing to Wi-Fi routers, BLE devices can be battery-powered and thanks to the low energy consumption they can work for several months or even years depending on their settings. Such devices are usually cheap, small and easy to install, which is a great advantage over the Wi-Fi IPS systems. BLE IPS is typically more accurate thanks to a denser beacon grid. On the other hand, Wi-Fi is more resistant to fast-fading thanks to the wider channel bandwidth [4] and Wi-Fi network is usually already available.

BLE usually uses the RSSI measurements and either the fingerprinting or multilateration method for the position estimation. The mean error achieved with the fingerprinting is between 1.0 m and 2.0 m, while the multilateration usually reaches mean error from 1.5 m to 3.0 m [4]. The reported density of the beacons in the environment is between 3 m^2 per beacon and 132 m^2 per beacon. Several studies ([5], [6]) report that the accuracy improves with the higher density of the beacons in a defined area, but at a certain density the accuracy saturates.

Faragher and Harle [5] describe two significant issues affecting the position estimation accuracy using BLE. They observed different RSSI variations in the three advertising channels caused by an uneven channel gain and a multipath interference. The reason behind this nature is the narrow bandwidth and the wide spacing of the channels (see Fig. 2.1). The different channels show deep fades at different positions. As a solution for this issues it is suggested to use evenly distributed measurements in the three advertising

channels for the position estimation.

The other part of their research is focused on an adequate samples filtering, the beacon's advertising period and the length of the scanning window. They did not observe considerable differences between maximum, median and mean filtering method. They were able to achieve less than 2.5 m distance error in 95% of all cases using the fingerprinting method.

Huang et al. [7] tried to approach this issue by separating the samples from the three advertising channels and establishing a separate model for each of them. They report that using all the three channels aggregated produced a higher standard deviation than using each of them separately. They characterize the quality of the data using the variance of the measured RSSIs. With Weighted Linear Least Square algorithm they achieved less than 2.4 m distance error in 90% of all cases.

It was proposed in [8] to use BLE gateway's RSSI measurements stored on server as a reference to correct the actual blind node's measurements. Thanks to that they were able to obtain mean distance error less than one meter using the multilateration method and Kalman filter.

It is worth noting that most of the researches were performed using smartphones available at the time, while an original hardware and firmware will be used within this thesis.

1.2. Project Goals and Their Evaluation

The main scope of this project is to develop an FPGA positioning testbed based on previously designed subsystems. The positioning capabilities of such a system will be analyzed. The starting point of this thesis is a BLE digital baseband accelerator called Celeste, which was developed in a previous semester project [9]. Celeste was designed for a positioning application and besides a BLE receiver it includes a power estimator, which makes it sufficient for the positioning with the RSSI measurements. As this core was not tested with the real BLE signals, some changes before its actual integration were necessary. A Matlab hardware true model of Celeste (also developed in previous project) had to be adapted together with those changes.

The verified core will be integrated to an SoC on an FPGA as an Advanced Peripheral Bus (APB) peripheral. The EvaLTE board [10] will provide an RF and analog interface for the whole setup. The function of the whole system must be verified with the real BLE signals either from commercial devices or generated by a radio communication tester as is CMW500 [11].

The development of an embedded software written in *C* programing language for controlling Celeste, analyzing the received packets and calculating the position from the measured RSSIs will be a significant part of this project. The experimental setup with BLE beacons will be installed in an ETH building. The performance of the developed IPS will be evaluated in the end of the project.

The main performance benchmarks of IPS systems are described in [12]. The emphasis will be mainly laid on the accuracy (mean distance error) of such a system and the

precision analysis using cumulative distribution function (CDF). The other important aspect is the position calculation rate, which determines the ability of the application to track moving objects.

1.3. Structure of the Thesis

Chapter 2 lays the theoretical context of this work. BLE physical and link layers are described in this section, followed by a description of commonly used protocols of the BLE beacons and the used positioning algorithm.

The architecture of Celeste and principles of the other used subsystems are described in Chapter 3.

Chapter 4 is dealing with the necessary modifications of Celeste core and its Matlab model. Celeste integration to the SoC, its implementation on the FPGA and the functional verification of the whole system are also described in this chapter.

The challenges negatively affecting the positioning accuracy were identified based on overnight RSSIs measurements, which will be described in Chapter 5 together with the function of the positioning application.

The final measurements and their evaluation can be found in Chapter 6.

At the end of this thesis, within Chapter 7, the achieved results are summarized and possible future progress of this project is outlined.

Chapter 2

Theoretical Background of BLE Positioning

The goal of this chapter is to describe technology and algorithms used throughout the project.

As previously mentioned, BLE is widely used technology for data transfer. It is in many aspects similar to Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR). The following sections focus on the physical and link layers of BLE. The knowledge of these two layers is essential for understanding the BLE digital baseband accelerator operation. Celeste includes a BLE receiver baseband accelerator and a power estimator, thus it is suitable for the positioning method using RSSI measurements from a grid of beacons. The information presented in the first two sections of this chapter are taken from Bluetooth Core specification [13] and BLE advertising packet guide [14].

2.1. BLE PHY

The BLE physical layer is responsible for the transmission of a raw bitstream and its receiving on the receiver side. BLE uses the unlicensed 2.4 GHz band, particularly the frequencies from 2.402 GHz to 2.480 GHz. This band is divided into 40 channels of 2 MHz bandwidth. The channels with indices 37, 38 and 39 are used for advertising. These channels are chosen so they do not interfere with the Wi-Fi channels (see Fig. 2.1)

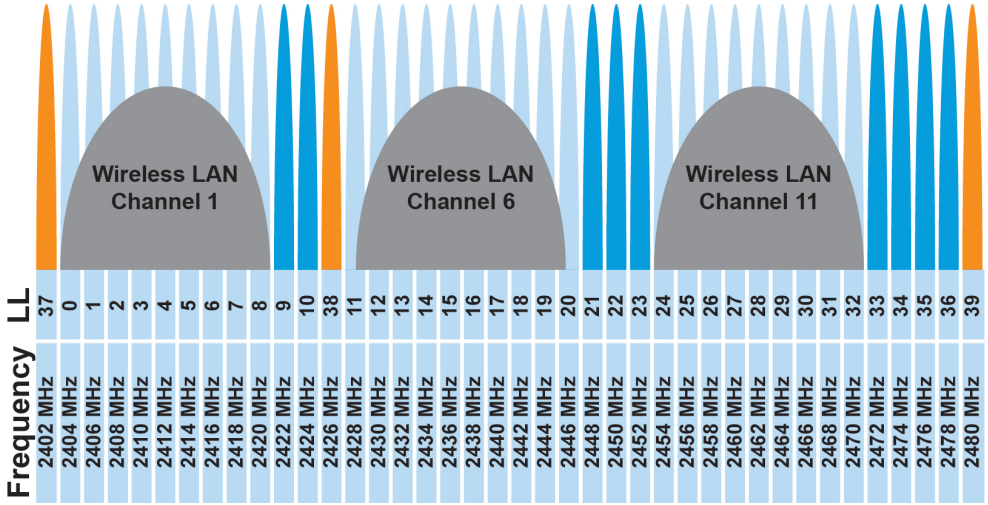


Figure 2.1.: BLE RF channels, taken from [15]

BLE adopts the frequency hopping in a predefined sequence in order to reduce the interference and fading. BLE advertises such as beacons usually successively use all the three advertising channels.

Gaussian Frequency Shift Keying (GFSK) modulation with the modulation index 0.45 to 0.55 is used in BLE. GFSK is described by following block diagram (Fig. 2.2).

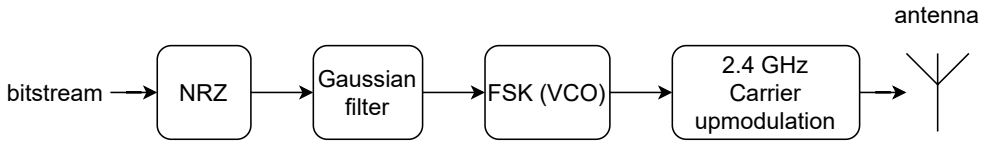


Figure 2.2.: GFSK modulation

The bitstream is first coded to a non-return to zero (NRZ) sequence, where the logical one is represented by value 1 and the logical zero is represented by value -1. This sequence is then filtered by a Gaussian filter. The impulse response of such filter is a Gaussian function, which helps to reduce the sideband power and the interference with the neighboring channels. The sequence is then modulated by an FSK modulator and it is finally transmitted on the desired channel. The offset from the center frequency during the packet transmission should not exceed ±150 kHz [13].

Celeste uses a Phase-shift discriminator for the GFSK demodulation, which produces Log-Likelihood Ratios (LLR). It can be decided, based on the LLR polarity, if the received symbol is the logical zero or one. The phase-shift discriminant method calculates the phase difference between two consecutive samples. It can be calculated following Eq. (2.1) and Eq. (2.2).

$$S_n e^{j\alpha}(S_{n-1}e^{j\beta})^* = S_k S_{n-1}^* e^{j\alpha-\beta} \tag{2.1}$$

$$\alpha - \beta = \arctan \frac{\Im(S_k S_{k-1}^* e^{j\alpha-\beta})}{\Re(S_k S_{k-1}^* e^{j\alpha-\beta})} \tag{2.2}$$

$S_k e^{j\alpha}$ is the current sample and $(S_{k-1} e^{j\beta})^*$ is the complex conjugate of the previously received sample. S is an amplitude of the signal. α and β are phases of the two consecutive samples. The other commonly used methods for the GFSK demodulation are Zero-Crossing Detection or Delay-Locked Loop.

The BLE specification defines two modulation schemes with 1 Msym/s and 2 Msym/s. The transmitted packets can be either in coded or uncoded format. Since the uncoded packets are used during advertising, only their structure and processing will be described in Section 2.2.

2.2. BLE LIN

The link layer is responsible for preparation of the packets for the transmission. Besides that it enables error detection and correction. BLE LIN specifies packet formats, a state machine allowing communication between the devices and bitstream processing.

2.2.1. Uncoded Packet Format

The content of the BLE uncoded packet is described by Fig. 2.3.

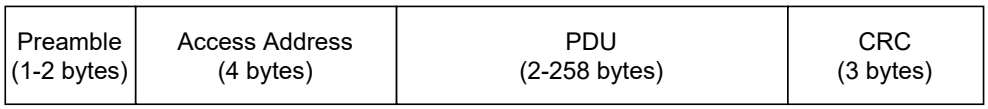


Figure 2.3.: BLE uncoded packet format

The transmission of the BLE uncoded packet starts with a preamble, which serves for synchronization purpose. The preamble is a bit sequence of alternating 0 and 1. It starts with the same bit value as is the Least Significant Bit (LSB) of an Access Address (AA), which follows the preamble. The preamble consists of 1 byte for 1 Msym/s modulation scheme and 2 bytes for 2 Msym/s.

The AA always has the value 0x8E89BED6 during advertising and it is transmitted LSB first. The AA is followed by a payload that consists of a two-byte header and up to 255 bytes of data. At the end of the packet is a three-byte CRC, which is transmitted Most Significant Bit (MSB) first. The CRC calculation is described in Section 2.2.3

PDU of Advertising Packets

Advertising PDU is described in Fig. 2.4. PDU type is transmitted LSB first, the next four bits follow in the same order as they are depicted in the figure. The payload length block is transmitted LSB first. The payload can be further divided into several blocks based on the application. It typically includes six-bytes Advertiser Address or Tx Power

indication. The advertiser address (MAC) is transmitted LSB first, the rest of the data is transmitted LSB first by byte.

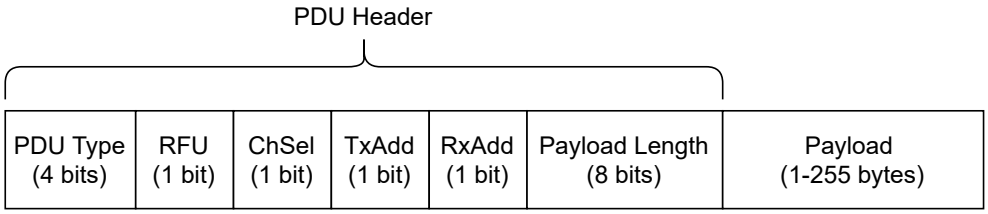


Figure 2.4.: BLE advertising PDU format

2.2.2. Link Layer States

The link layer defines the state machine of 7 states depicted in Fig. 2.5. Advertising and Scanning states are essential for the designed application. The advertising devices are broadcasting packets in periodic advertising events on the three advertising channels. Those can be used in any order. The packets are transmitted during an advertising interval, which can be any multiple of 625 µs between 20 ms and 10.485 s. This interval is followed by a pseudo-random delay within the range 0 ms to 20 ms. Every scanning device can detect those packets.

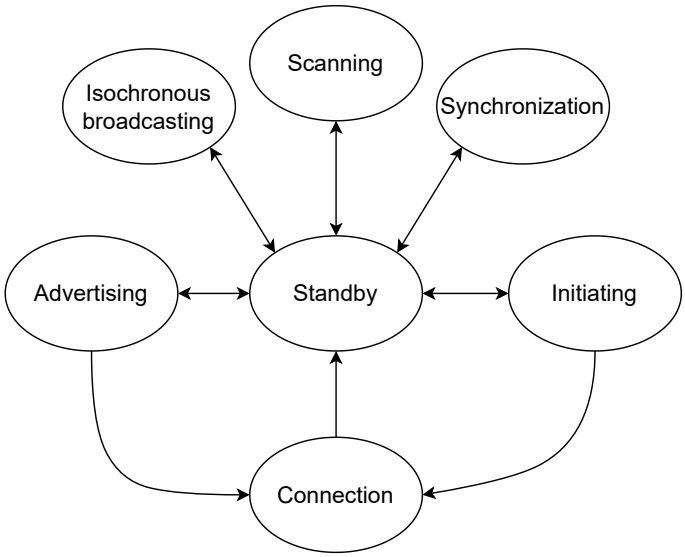


Figure 2.5.: BLE link layer state machine

2.2.3. Bit Stream Processing

The BLE bitstream processing of an uncoded packets consists of Cyclic Redundancy Check (CRC) generation and whitening. The CRC is used to detect the packets received

with errors. The CRC is calculated on the whole PDU and it is appended to the end of the transmitted sequence. The CRC is calculated independently on the receiver side and it is compared to the received CRC. A 24-bit linear feedback shift register (LSFR) defined by a polynomial in Eq. (2.3) is used for the CRC generation.

$$x^{24} + x^{10} + x^6 + x^4 + x^3 + x + 1 \tag{2.3}$$

The register is initialized by 0x555555 with its LSB corresponding to the LSB of the LSFR. This value is valid for all the advertising channels. The bitstream is fed to the LSFR in the transmitting order. The resulting CRC is transmitted MSB first.

The data is further randomized (whitened) to remove long sequences of ones or zeros. The data is whitened in the same order as they are coming from the CRC generator. The whitening sequence is generated by a 7-bit LSFR defined by a polynomial in Eq. (2.4).

$$x^7 + x^4 + 1 \tag{2.4}$$

The generated whitening sequence is XORed with the coming bitstream. The register's LSB is initialized by 1. The rest of the register's bits are initialized by the channel index (37, 38 or 39 for advertising channels) with MSB at second lowest bit of LSFR. The data has to be whitened by the same process on the receiver side to obtain the original bit sequence.

2.3. BLE Beacon

BLE beacons are devices that are only advertising in specified time intervals. They can transmit several bytes of information including an Universally Unique Identifier UUID, a reference RSSI, their own identification or a link to a website. They are used for positioning, localized product promotion or as wireless sensors. AltBeacon, URIBeacon, Eddystone and iBeacon are the most common protocols.

iBeacon has a great advantage over the rest of them, since it is widely supported. The protocol is maintained by the Apple company. AltBeacon, on the other hand, has better data space usage and it is an open source platform. URIBeacon serves only for URL advertising [16]. It was decided to use iBeacon for its wide usage among commercial devices.

2.3.1. iBeacon

The iBeacon protocol consists of 30 or 31 bytes. 20 bytes out of them are reserved for user data. iBeacon payload is described in Fig. 2.6.

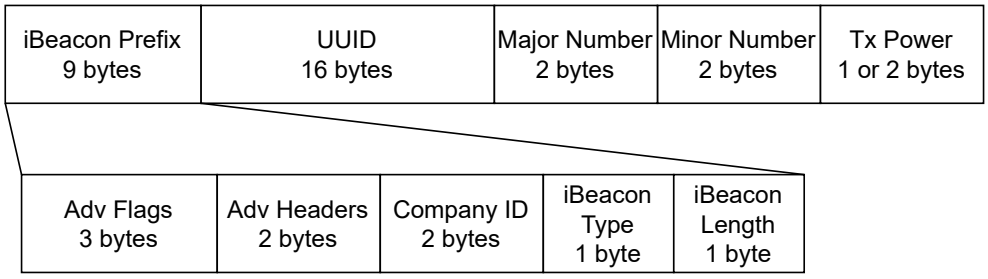


Figure 2.6.: iBeacon protocol

UUID is an application specific number. The major number identifies a compact set of the beacons (e. g. within one building's floor or a branch of a shop) and the minor number identifies the particular beacons within this set. Tx Power can consist of one or two bytes. iBeacon reports RSSI value in one meter distance in this field.

2.4. Path Loss Model

The distance from the beacons can be estimated based on a mathematical model and RSSI measurements. The path loss model describes the loss in the signal power over the distance from a transmitter. It is caused by the signal propagation, distortions, reflections etc. [2]. The signal power loss has a logarithmic dependence on the distance and for the RSSIs measurements is defined by Eq. (2.5).

$$RSSI(d_i) = RSSI(d_0) + 10n \log\left(\frac{d_i}{d_0}\right) + M_i + X_i \tag{2.5}$$

d_i is the distance from the transmitter, d_0 is a distance, where a reference RSSI was measured and n is an environment dependent variable. It typically reaches values from 2 to 5. M_i describes the small scale fading and can be diminished by filtering and X_i is an environment dependent zero-mean log-normal variable characterizing the large scale fading [2].

2.5. Multilateration

The multilateration is a method to estimate the device's (receiver's) location based on the known distances from the beacons (transmitters) and their positions. The multilateration was chosen to be used within this project, because it can achieve sufficient accuracy, while maintaining the main BLE advantages such as low cost and simplicity. The advantage of the multilateration method is that, besides the known location of the beacons, it does not require any site specific data. The multilateration principle is illustrated in Fig. 2.7.

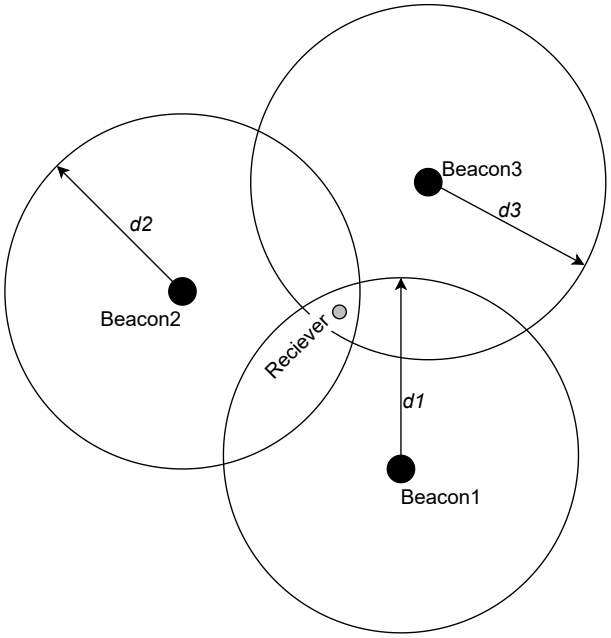


Figure 2.7.: Illustration of multilateration method

The position estimation problem is defined as search for solution of Eq. (2.6), which describes the unknown position $\boldsymbol{x}$ as a function of the distances $\boldsymbol{d}$ from the known points and noise $\boldsymbol{w}$.

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{d} + \boldsymbol{w} \tag{2.6}$$

The solution can be found if at least three distances from three different beacons are known. The distances from the beacons can be estimated using the following equation derived from Eq. (2.5).

$$d_i = d_0 10^{\frac{RSSI(d_0) - RSSI(d_i)}{10n}} + M_i + X_i \tag{2.7}$$

X_i and X_i are both zero-mean, normal distributed random variables. Parameters of path loss model $RSSI\,(d_0)$ and n has to be determined based on measurements in the environment.

Approaches used to find the solution for problem Eq. (2.6) can be divided to linear and nonlinear. The nonlinear approach is more accurate, but it requires noise statistics and it is computationally more demanding [2]. The commonly used nonlinear methods are Newton-Raphson, Gauss-Newton and steepest descend method [2].

The linear approach, on the other hand, is less accurate, but it is simpler to implement. The linear least square (LLS) method and the weighted linear least square (WLLS) method are the most common linear algorithms. LLS was chosen for this thesis for its simplicity and computational efficiency.

2.5.1. Linear Least Square method

The LLS search for solution of linearized Eq. (2.6) by minimizing the sum of errors from the estimated position. Matrix of the known beacons' positions is described in Eq. (2.8).

$$\boldsymbol{A} = \begin{pmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \ddots & \vdots \\ -2x_k & -2y_k & 1 \end{pmatrix} \tag{2.8}$$

x_k and y_k are the known coordinates of k beacons.

The estimates of the distances d_k from k beacons, calculated using Eq. (2.7), form matrix $\boldsymbol{b}$ in Eq. (2.9).

$$\boldsymbol{b} = \begin{pmatrix} -d_1^2 - x_1^2 - y_1^2 \\ -d_2^2 - x_2^2 - y_2^2 \\ \vdots \\ -d_k^2 - x_k^2 - y_k^2 \end{pmatrix} \tag{2.9}$$

And finally the solution of the LLS algorithm is described by the matrices in Eq. (2.10) and Eq. (2.11)

$$\boldsymbol{\Theta} = \left(\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{b} \tag{2.10}$$

$$\boldsymbol{\Theta}^{\boldsymbol{T}} = \begin{pmatrix} x & y & x^2 + y^2 \end{pmatrix} \tag{2.11}$$

x and y are the position coordinates of the receiver.

Eq. (2.6) and Eq. (2.9) to Eq. (2.11) are taken from [2].

Chapter 3

Description of Used Subsystems

The main part of this chapter is dedicated to the Celeste Bluetooth digital baseband core, which will be adapted and integrated to an SoC. It is followed by a description of the PULP based SoC, to which was Celeste integrated. The used FPGA, RF subsystem and software development tools are described in the end of this chapter.

3.1. Architecture of the Celeste Core

Celeste is a Bluetooth receiver digital baseband accelerator written in VHDL and it was designed in a semester project in 2019 [9]. All the information in this section was taken from this thesis. Celeste was provided as VHDL code together with its Matlab hardware true model. Besides detecting and receiving Bluetooth packets it is able to estimate the noise and the signal power. Detected and demodulated packets are stored in the memory in form of LLRs, where they can be read from through APB and further processed.

Complex input signal from an external Analog-to-Digital Converter (ADC) is in form of signed 9 bit integer IQ samples. The imaginary part is mapped to the upper 10 bits of the signal and the real part of the signal is mapped to the lower 10 bits.

Celeste was designed for up to 80 bits long preamble. The preamble length and its type can be configured in the APB registers.

Celeste was developed to be used with oversampling ratio (OR) of 8. Data sheet of the Celeste core can be found in Appendix A of [9].

The block diagram of the Celeste architecture can be found in Fig. 3.1. The function of each block will be described shortly.

3.1.1. Finite Impulse Response (FIR) Filter

Every sample coming from the ADC is first filtered by a low pass FIR filter. There are altogether 19 taps available within the hardware, but only 17 of them are used. The taps' values are stored in the APB registers and they can be reconfigured. Filter's coefficients

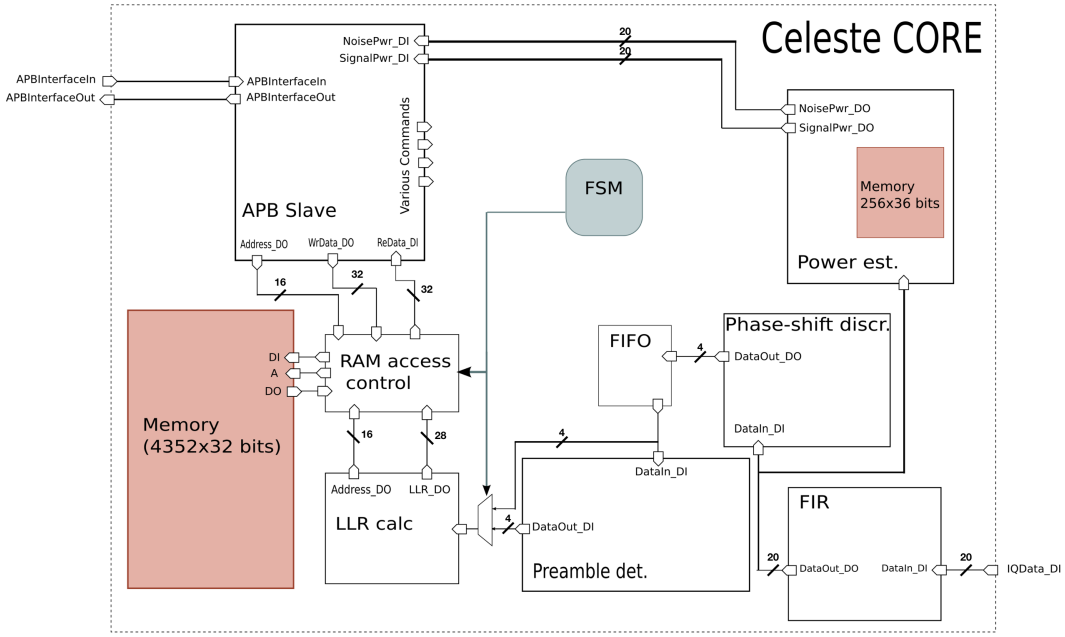


Figure 3.1.: Block diagram of the Celeste core, taken from [9]

are written as signed Q0.4[1]. The cut off frequency of the filter is 680 kHz at -3 dB and attenuation in the stop band is -15 dB.

3.1.2. Phase-Shift Discriminator

Phase-Shift discriminator was implemented in the form of the Coordinate Rotation Digital Computer (CORDIC) algorithm for the phase difference calculation. It is an iterative algorithm that computes an angle necessary to rotate one vector to the position of another vector [17]. The basic CORDIC algorithm was adjusted to calculate arctangent approximation of the two consecutive samples following equation Eq. (2.2). The phase between them is calculated in 8 iterations. The output phase signal is in form of signed Q-3.6. In addition, the implemented design filters out samples (phase is set to zero) with a high phase difference as they are assumed to be produced by the noise.

3.1.3. Preamble Detector and FIFO

The preamble is detected by cross-correlation of the samples coming from the phase-shift discriminator and a reference preamble stored in the APB registers. The stored preamble has to be coded to the NRZ before the correlation calculation. The preamble detector operation can be simplified to four stages.

[1] Q formating for describing representation of fixed point numbers will be used throughout this thesis. Q$m.f$ means that the number is represented by m integer bits and f fractional bits.

It is in the IDLE state after reset and it will change to COARSE SEARCH, when it receives start command (see Section 3.1.6). In that state the core receives samples and is searching for the preamble in every eighth received sample (one sample per symbol). When a certain threshold of correlation is reached, the state will change to FINE SEARCH.

In this state the detector will search for the highest correlation within the next new 104 samples. This value was chosen to avoid the local maxima in the cross-correlation. Within this state are used all available samples for the cross-correlation calculation.

When the next new 104 samples are received, the state will change to FLUSH. All samples stored in the preamble detector are then read by the LLR calculator. All following samples can be read by the LLR calculator directly from the FIFO to avoid unnecessary energy consumption in by the preamble detector. Then the preamble detector will change back to IDLE.

3.1.4. LLR Calculator

When the preamble is found, the LLR calculator starts summing eight consecutive phase signal samples and saving them to RAM. Four LLRs are stored in one 32-bit register.

3.1.5. Power Estimator

The power estimator includes memory for 256 samples. It begins to save samples coming from FIR filter, when the preamble search is started. The oldest samples in the memory will be overwritten by the new ones. The power is calculated, when the preamble is found. The 128 oldest samples that were received before the preamble was found are used for the noise power calculation and 128 that are yet to come will be used for signal power calculation. Both the noise and signal powers are calculated following Eq. (3.1).

$$P = \frac{1}{N} \sum_{n=1}^{N=128} (I_n^2 + Q_n^2) \tag{3.1}$$

I_n and Q_n are the real part, the imaginary part respectively, of IQ samples.

The average simulated error of the power estimation is SNR dependent and it reaches values of ca. 9% down to 4% of actual transmitting power for SNR from 12 dB to 22 dB. The more detailed evaluation of the power estimator can be found in [9].

3.1.6. APB Slave and LLR RAM

An APB slave is part of Celeste for simpler integration to a processor system. The memory with the stored LLRs can be accessed through this slave. There are three essential registers for the Celeste's operation:

- 0xFF00 - write 0x01 for synchronous reset
- 0xFF01 - write 0x01 to start search for a new packet
- 0xFF0C - number of the LLRs to be received

A detailed description of all the registers can be found in Appendix A of [9].

The LLR RAM consists of 4352 registers of 32-bit width, which means it can store up to 17408 LLRs.

3.1.7. Final State Machine (FSM)

The Celeste's operation is controlled by an FSM of five states depicted in Fig. 3.2.

The core is in the IDLE state after the reset and it changes to PRESEARCH, when starting signal at 0xFF01 is received. The preamble detector is activated in this state.

When the preamble is found, the core transfers to the FLUSH state that follows corresponding state of the preamble detector (Section 3.1.3). The power estimation and the LLR summation are started during FLUSH state.

LLRs are calculated and stored to the memory during MESRECEIVE state. When the number of received LLRs reaches value set in 0xFF0C register, the core transfers to the MESSAGEREADY state, where it generates *MsgRdy_SO* output signal and in the next clock cycle the state changes back to IDLE.

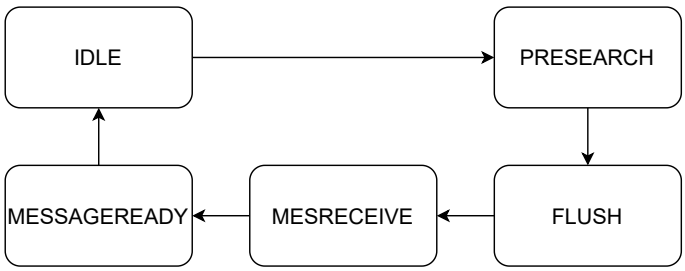


Figure 3.2.: Block diagram of the Celeste's FSM

3.2. Matlab HW True Model

It is necessary to verify the performance of the hardware for the worst case scenario before its actual implementation. There are two fundamental characteristics that describe performance of a digital baseband accelerator. First, Bit Error Rate (BER) and its dependency on SNR, and second, Packet Miss Rate (PMR) and its dependency on SNR.

Both can be simulated with the Celeste's Matlab HW true model. It was created within a group project [18] and adjusted for Celeste within [9]. Its block diagram can is depicted in Fig. 3.3.

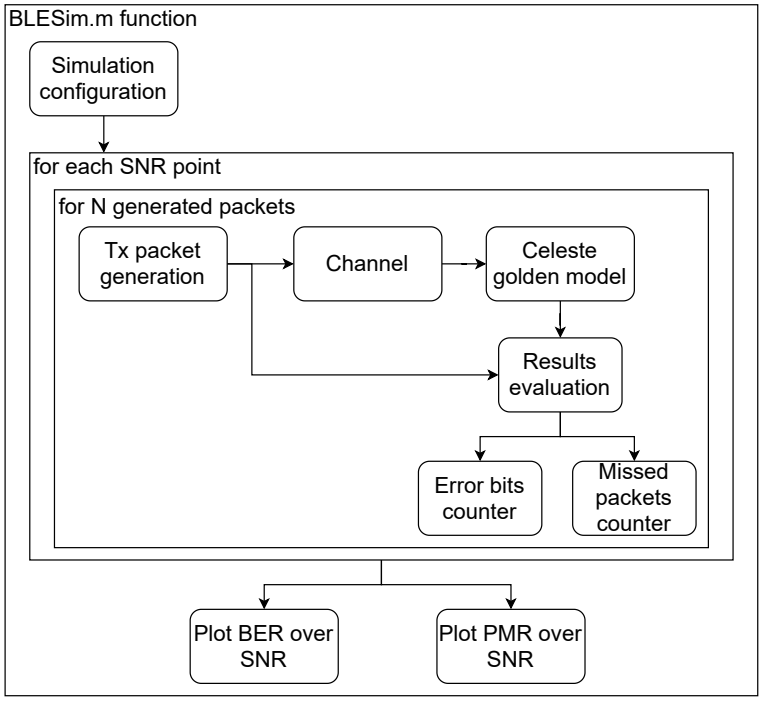


Figure 3.3.: Block diagram of Matlab golden model of Celeste

The simulation loads its configuration and generates N packets per SNR point. The length of the packet, the preamble type and the modulation parameters can be configured. The generated signal passes through the Channel block, where a frequency offset, time offset and Additive White Gaussian Noise can be appended to the signal based on the channel configuration. The time offset consists of 200 bits and 5 samples before the packet. Simulations with the frequency offset and the random time offset are not reported within [9]. The generated frame is then passed to the Celeste golden model, which corresponds to previously described Celeste architecture. The model is divided into the same blocks as the actual hardware. Each block is able to generate stimuli and responses files for Register Transfer Level (RTL) simulations. Hard decision is done on the calculated LLRs in order to compare the bits of the received packet with the bits of the generated one.

The missing packet is indicated either if the preamble is not found at all within the transmitted frame or if the difference between the actual and the estimated position of the preamble's first sample is higher than 3 samples. Bits of missed packets are not included to BER calculation. BER and PMR plot of the Celeste model before any modification is displayed in Fig. 3.4.

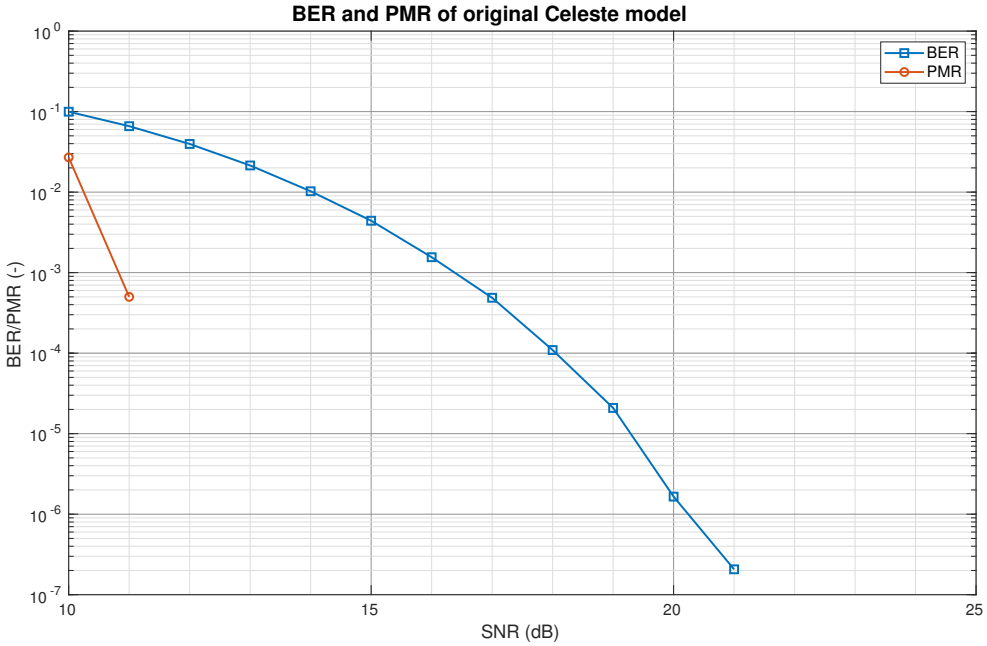


Figure 3.4.: BER and PMR performance of original golden model of Celeste

3.3. PUPL Processor System

A RI5CY based SoC [19] was provided for the integration of the Celeste core. It is based on the Parallel Ultra Low Power (PULP) platform developed as a joint project of IIS group from ETH and Energy-Efficient Embedded Systems (EEES) group from Bologna University. The goal of this platform is to develop an open-source ultra-low-power processor architecture with high performance scalability for the IoT applications [20].

RI5CY is a 32 bit core based on the RISC-V instruction set architecture with the PULP specific instruction set extension and a 4-stage pipeline [21].

The SoC was previously designed at IIS. It is a complex RF system consisting of an RF-transceiver, quad-core RI5CY processor and several HW accelerators to optimize the system's power consumption [19]. It supports Assisted-GPS and cellular Observed Time Difference of Arrival (OTDOA) positioning.

3.4. FPGA Testbed

Zynq UltraScale+ MPSoC evaluation kit by Xilinx was used for implementation of the whole system on the FPGA. It features 504k of programmable logic cells, 38 Mbit of memory and 1728 DSP slices [22]. It provides Ethernet, UART and FPGA Mezzanine Card (FMC) connectivity. These interfaces were used for interaction with the designed application and for connection of RF subsystem.

3.4.1. RF Subsystem

EvaLTE PCB [10] was provided for this project to serve as RF transceiver. It was designed to support 2G, 3G and 4G cellular communication for FPGA based prototyping. It is connectable to the FPGA via FMC. The gain of the board as well as the RF frequency and band can be set from the SoC.

3.5. Software Development Tools

Software was developed in *C* programing language using Eclipse SDK. The software designed during this thesis is using a previously designed platform developed for the SoC, where will be Celeste integrated. It is based on the Little Kernel (LK) platform, which is a small operating system developed by Travis Geiselbrecht for embedded systems with limited memory space [23].

The RISC-V GNU compiler toolchain was used for code compilation and ARM-USB-TINY-H for programming and the *C* code debugging.

Chapter 4

Hardware Modification and Integration

This chapter is focused on the description of necessary changes to Celeste and its integration to the SoC. This chapter is divided into three sections.

The changes in Celeste and their impact on the core's performance will be explained in Section 4.1. Each step was at first verified by the Matlab simulations and the implemented changes were tested by the RTL simulations using the test vectors generated by the hardware true model.

Integration of Celeste to the SoC is described in Section 4.2. A C library for Celeste control was developed and it will be described in this section as well.

After the successful implementation on the FPGA, Celeste was tested by the reference BLE signal generated by the CMW500 RF tester. These measurements and their evaluation are described in Section 4.3.

4.1. Modification of Celeste for FPGA Implementation

The Celeste core described in the previous chapter had to be further tested and modified for the desired application. Format of the advertising packet and correct modulation index had to be set to the Matlab HW true model at the first place as is described in Section 4.1.1.

As was mentioned previously the core's model was not simulated with frequency offset and random sample offset before the packet, which both well represent the real world conditions. These simulations revealed problems regarding worse BER and PMR performance, which will be described in Section 4.1.2 and Section 4.1.3.

4.1.1. Modulation Index and Advertising Packet

The performance of Celeste HW true model had to be verified for receiving BLE advertising packets. The original Celeste was evaluated using an 80 bits preamble, which

is used in the coded packets. This preamble consists of 10 repetitions of 0b00111100. Employing only an 8 bits preamble that is used in the advertising packets results in a high packet loss. For that reason, it is better to detect the packets using both the preamble and the access address in one synchronization word. Altogether 40 bits are used for the synchronization, which is a half of the coded packet's preamble, so a drop in the PMR performance is expected. The whole iBeacon packet including the preamble, access address, PDU and CRC is 368 bits long. This value was used for the following Matlab simulation. The threshold of the coarse search had to be adjusted to the preamble length.

The modulation index used in evaluation of original Celeste was $h = 0.315$, which is valid value for Bluetooth BR/EDR. Modulation index used in BLE is between $h = 0.450$ and $h = 0.550$ as was mentioned in Section 2.1. The modulation index was simulated in three values, but only the simulation of $h = 0.450$ is depicted in Fig. 4.1 as they all provided similar results. This value was also used in all the next simulations.

Fig. 4.1 shows that PMR has a floor of around 5% even at high SNR. This is not a satisfactory result and its correction will be described in Section 4.1.3. BER was improved by a higher modulation index comparing to Fig. 3.4.

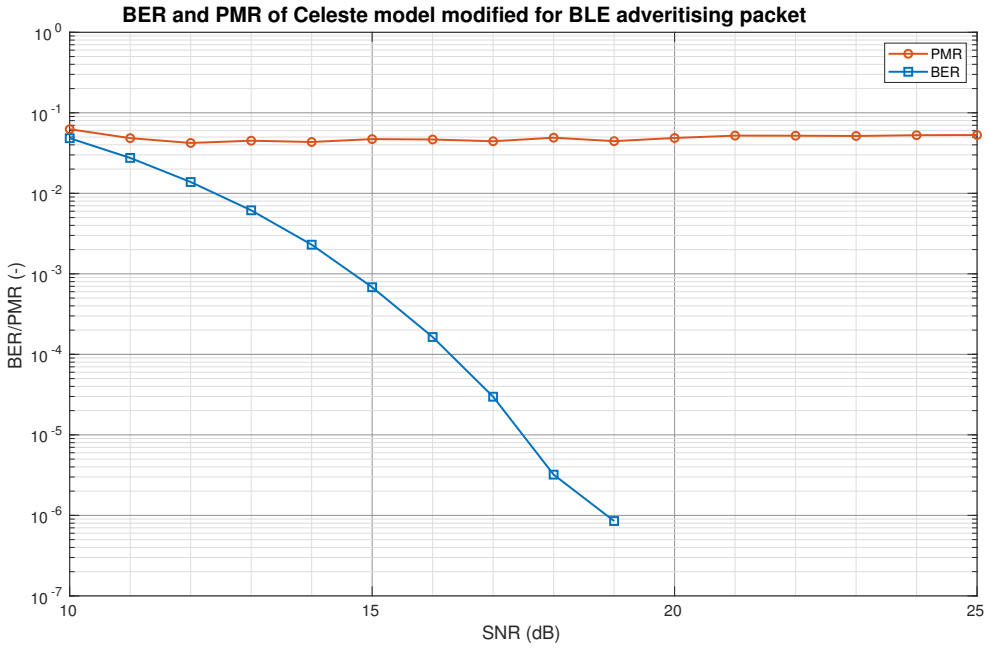


Figure 4.1.: BER performance of Celeste in settings for BLE advertising packet

4.1.2. Frequency Offset Compensation

The maximal frequency offset during the whole packet defined by [13] is $\pm 150\,\text{kHz}$. The simulation of Celeste (see Fig. 4.2) with a random offset between $-150\,\text{kHz}$ and $150\,\text{kHz}$ from the central frequency showed significant drop in the BER performance, but almost no change in PMR. Due to the high frequency offset, which is close to the

actual frequency used for the data modulation (225 kHz for $h = 0.45$), the samples can be wrongly interpreted. Considering that any incorrectly received bit will result in a CRC error, and thus in losing the whole packet, such results are not sufficient for the given application. It would take too much time to receive enough valid packets.

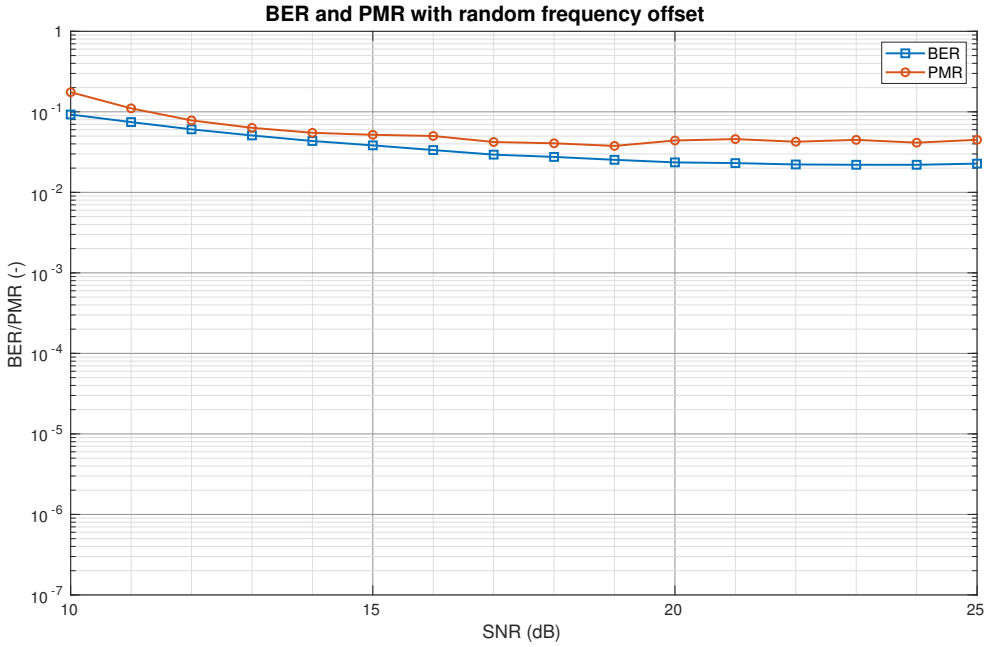


Figure 4.2.: BER and PMR performance of Celeste with random frequency offset

As was proved in [18], the frequency offset results in a shift of the mean value of the LLRs and it can be compensated using this knowledge. In case there would be no frequency offset, the mean values of the synchronization word in the NRZ coding and the received synchronization word in the LLRs would be equal. From these mean values a compensation value can be calculated and every received LLR can be corrected following equation Eq. (4.1).

$$LLR_{rxc} = LLR_{rx} + (\overline{SYNC}_{NRZ} - \overline{SYNC}_{rx}) \tag{4.1}$$

LLR_{rx} are the received LLRs and LLR_{crx} are the compensated LLRs. $\overline{SYNC}_{NRZ}$ is the mean value of the synchronization word in the NRZ coding and $\overline{SYNC}_{rx}$ is the mean value of the received synchronization word represented by the LLRs.

Further simulations showed that using the phase samples in signed Q-3.6 does not provide sufficient accuracy for the frequency offset compensation. It was necessary to increase the phase samples' fixed point representation to signed Q0.7 to achieve satisfactory results. The simulations for the three different fixed point and the floating point representations are shown in Fig. 4.3. The threshold value for the coarse preamble search and the maximum signal phase value had to be adjusted for each of the fixed point representation. There is an improvement compared to Fig. 4.2 already with the

original phase signal (orange curve). But still a lot of received packets would result in a CRC error.

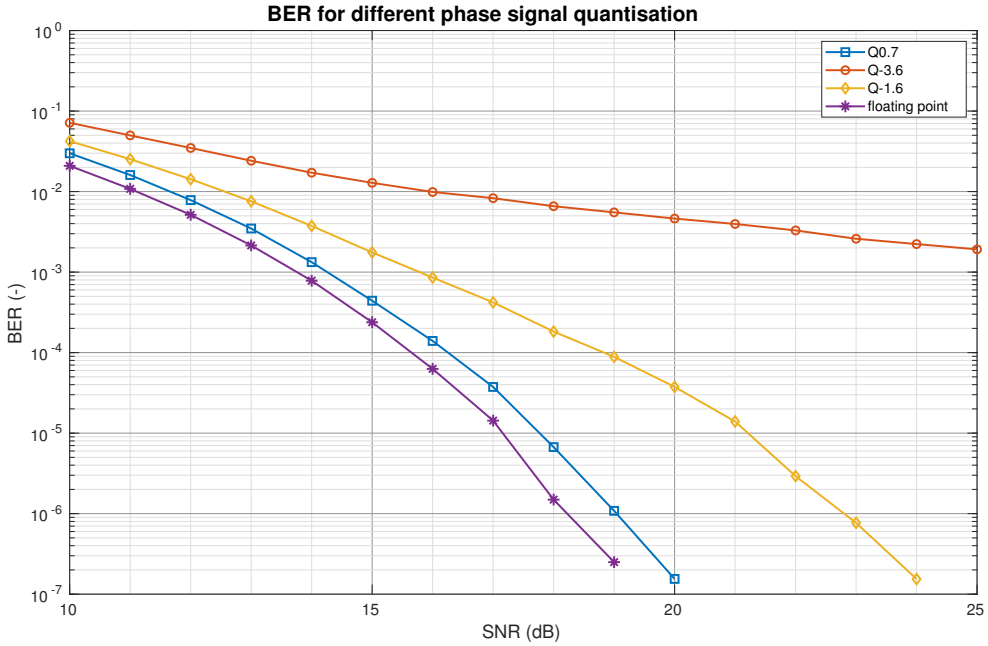


Figure 4.3.: BER performance of Celeste with random frequency offset and its compensation, comparison between different fixed point and floating point representations

To implement those changes to hardware the signal path from the phase shift discriminator to the LLR calculator had to be extended to signed Q0.7. This will come with area costs especially in the preamble detector, which has a long shift register in the signal path. Wider changes are required in the LLR calculator. The frequency offset compensation was implemented to HW as is described by block diagram in Fig. 4.4. The LLR calculator starts summing the phase samples, when the preamble is found. The LRRs that correspond to the synchronization word are stored in a 80 samples long shift register. When the whole synchronization word is received, its mean value is calculated following equation Eq. (4.2).

$$\overline{SYNC}_{rx} = \frac{1}{L}\sum_{l=1}^{L} LLR_l \tag{4.2}$$

L is length of the synchronization word. Value of $\frac{1}{L}$ is stored in an APB register as signed Q0.7 to avoid division in hardware. A new register is also assigned for the preamble length value and for $\overline{SYNC}_{NRZ}$.

The compensation value is calculated and it is stored in a register for debugging purpose. The LLRs leaving shift register are compensated and four of them are arranged to one 32 bit memory word.

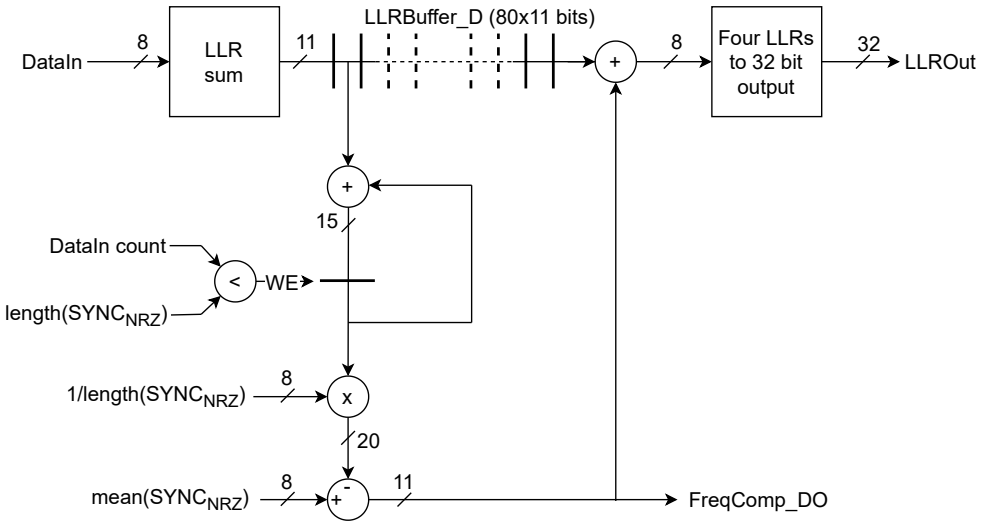


Figure 4.4.: Simplified block diagram of frequency offset compensation in LLR calculator

4.1.3. Preamble Detector Improvement

Celeste with the implemented frequency offset compensation was simulated with a random time offset before the packet. This offset is represented by a random number of bits (up to 200 bits) in front of the actual packet and by a random number of samples (0 to 7 samples).

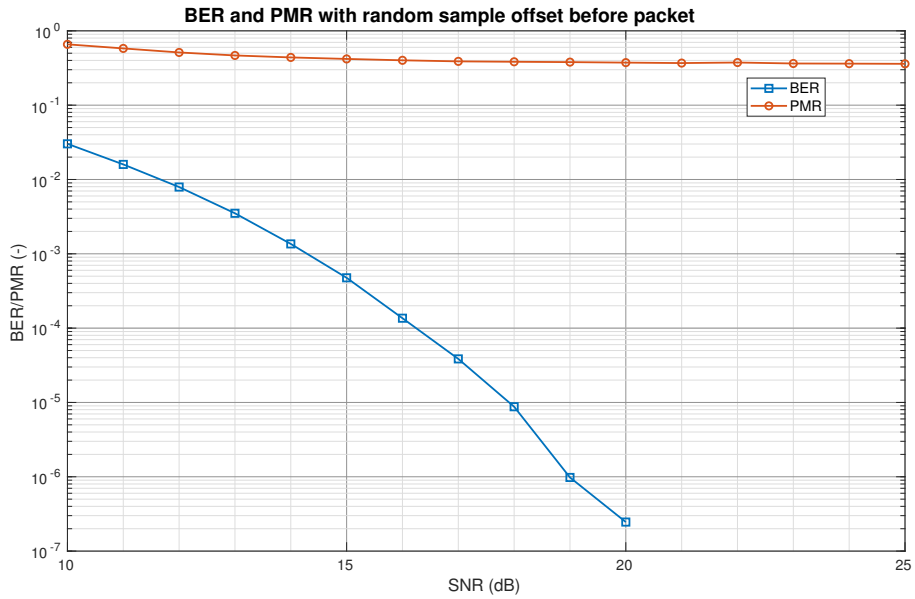


Figure 4.5.: BER and PMR performance of Celeste with random time offset

The time offset simulation uncovered a problem that resulted in high packet loss (see Fig. 4.5), but BER stayed unaffected. Drop in the PMR performance is caused by the preamble detector's architecture. Fig. 4.6 shows the preamble in the NRZ coding filtered by the Gaussian filter. The positive samples represent a binary 1 and the negative samples a binary 0. In the coarse search only every eighth sample is used for the cross-correlation calculation. This approach was used for lower power consumption.

The samples highlighted by the red color are the first samples of each symbol. It is clear from this figure that the highest correlation will be reached if the samples with offset four $(8l + 4)$ are used. On the other, hand using the samples with zero offset $(8l + 0)$ will produce a correlation value close to zero. Such an approach causes loss of up to 40% of all packets, because lot of packets will have lower correlation value than coarse search threshold and they will not be detected.

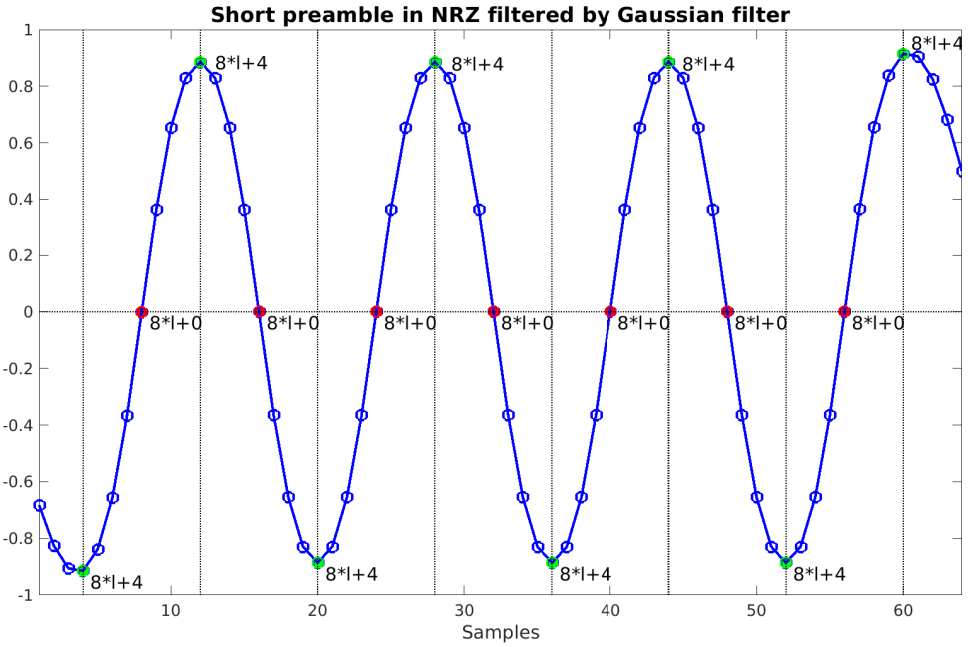


Figure 4.6.: Short preamble coded to NRZ and filtered by Gaussian filter

This can be solved by using more samples during the coarse preamble search that are evenly distributed over the symbols during the coarse preamble search. Fig. 4.7 shows different settings in the coarse preamble search. Fig. 4.7 shows that the employment of every second sample proved to be sufficient for PMR characteristic improvement.

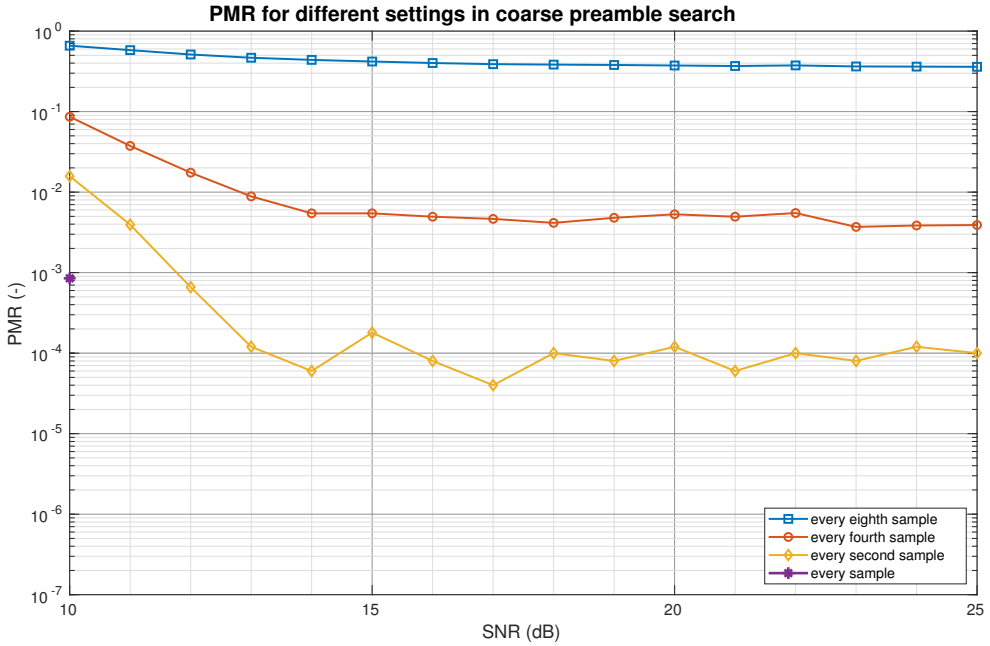


Figure 4.7.: PMR performance of Celeste with different number of samples used in the coarse preamble search

The infrastructure for implementation of these changes to the hardware was already available. The only required modification is to change the internal preamble detector's sample counter to count to two instead of to eight. Thus these changes will not result in larger chip area, but they could cause higher power consumption.

4.1.4. Minor Changes to Celeste

FIFO

It was observed in the RTL simulation that Celeste gets stalled, when the FIFO is filled before the core is started. This was caused by false operation of the FIFO's internal sample counter. When the FIFO is filled, it will not acknowledge any more samples coming from the phase shift discriminator and the write require signal from the same block will stay high. The samples can still be read from the FIFO by the preamble detector, but the FIFO's internal sample counter will not update as it can decrement only if the write require from phase shift discriminator is low. The FIFO is then still considered as full and it will not accept any new samples. The preamble detector then reads the same samples over and over again. This issue was solved by wiring the core start signal to the FIFO and letting it accept the samples only, when the core is started.

Power Valid Flag

It was discovered that if a packet was found too soon after the core had been started, the power estimation could provide wrong results. This is caused by the memory size in the power estimator. When the packet is found before the memory is filled the power is calculated on partly empty registers. A power valid flag was implemented to avoid using the wrong data for the distance and position calculation. This flag signalizes that at least 256 samples had been received before the packet was detected and that the values of the signal and noise power were successfully stored in the corresponding APB register. This signal was mapped to an APB register as well.

"First LLRs Ready" Output Signal

As was described in Section 3.1, the LLR calculation is stopped when a certain number of the received samples defined in register 0xFF0C is reached and *MsgRdy_SO* signal is generated. This could cause complications as the actual number of bits within the packet is not known beforehand.

The packet length can be read from the second byte of the PDU header (see Fig. 2.4). This knowledge was used for the implementation of another output signal called *FirstLLRsRdy_SO* indicating that enough samples to estimate the packet's length from its PDU header were received. These samples can be read by the software and based on the information from the PDU header, the actual length of the packet can be set to 0xFF0C register. Another APB register, which contains the number of LLRs from the beginning of the preamble until the end of the PDU header, must be implemented.

APB Registers

The APB slave and its registers had to be adjusted following the changes described above. A register, where the number of received samples in FIR filter is stored, was added for debugging purpose. The mapping of registers and their purpose is described in Appendix A. The registers starting with address 0xFF12 upwards were implemented within this project. The length of the individual registers was adjusted based on the used number of bits.

4.1.5. Impact of the Modifications to Celeste

Fig. 4.8 depicts the final performance of the Celeste core after all the changes. BER was improved by the frequency offset compensation and by using wider phase signal. PMR was reduced by increasing the number of employed samples in the coarse preamble search.

The Block Error Rate (BLER) was simulated for comparison with the CMW500 measurements. Block error is sum of the missed packets and packets with at least one wrong bit within their PDU. The floor in BLER is caused by the same floor in the PMR characteristics.

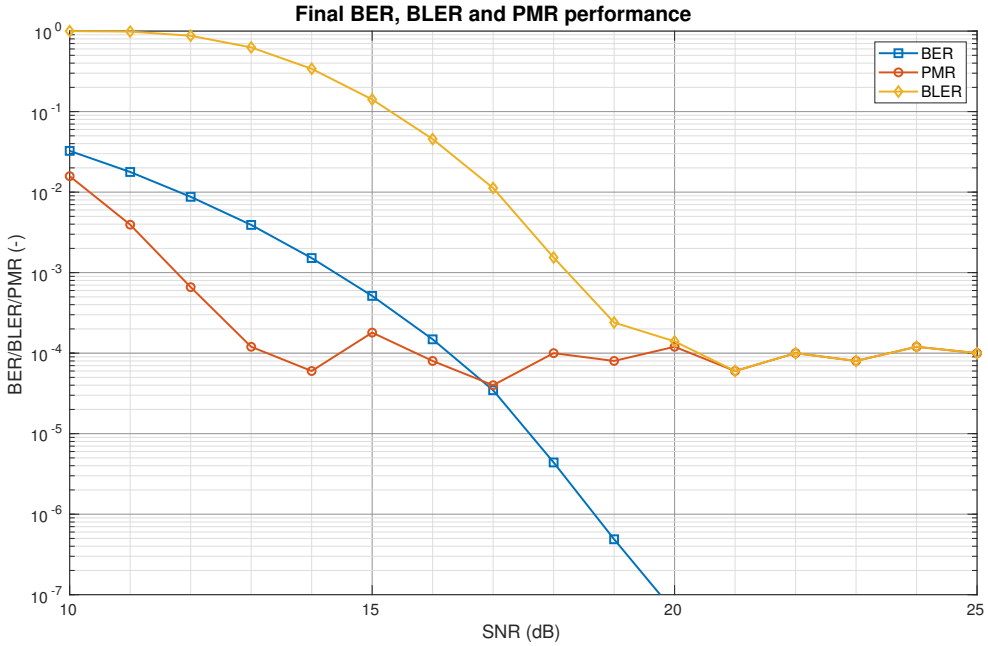


Figure 4.8.: BER, PMR and BLER performance of Celeste after all the modifications

Those changes had impact on the area of the accelerator. Celeste was synthesized for UMC65 technology after all mentioned changes to compare the area with the original core. Area of preamble detector and FIFO increased almost twice, which corresponds to twice wider phase samples signal. LLR calculator became a considerably more complex system. The area of the APB slave decreased due to usage of only the necessary number of bits. The overall core area increased by ca. 25%. A detailed overview of the impact of the previously mentioned changes on the Celeste area can be found in Table 4.1. Values of original Celeste's area are taken from [9].[1]

[1]Gate Equivalent (GE) in UMC65 technology is 1.44 μm^2

Block	Area of original Celeste	Area after modifications	Area after modifications	Change
	(GE)	(GE)	(μm^2)	
FIR filter	5621	5974	8603	6%
Phase discriminator	3648	3672	5288	1%
FIFO	3070	5715	8229	86%
Preamble detector	37107	70213	101107	89%
LLR calculator	641	8797	12667	1272%
APB slave	3956	3447	4964	-13%
LLR memory	102286	102286	147292	0%
Power estimator	9483	9494	13672	0%
Celeste core	**165812**	**207253**	**298444**	**25%**

Table 4.1.: Area summary of Celeste after all mentioned modifications

4.2. System Integration

The Celeste core was integrated as APB peripheral to the RI5CY based SoC. Celeste input was changed to signed Q3.12 to be compatible with IQ samples bus within the SoC. This signal was then resized within Celeste to signed Q3.6. Signals *MsgRdy_SO* and *FirstLLRsRdy_SO* are mapped to interrupts of the processor system. The whole system was programmed on Xilinx Zynq UltraScale+ MPSoC. The SoC is using clock frequency 100 MHz. The simplified block diagram of the integrated system is in Fig. 4.9.

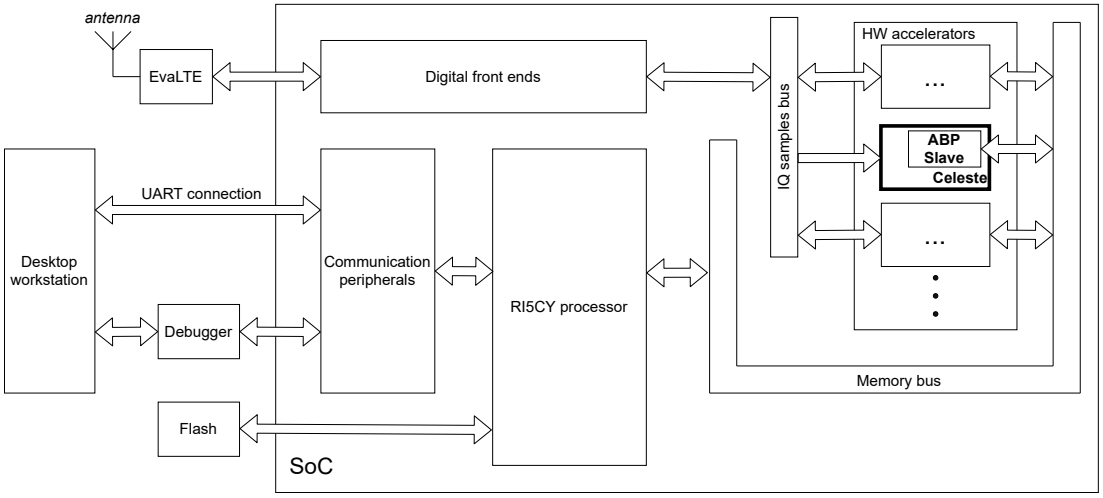


Figure 4.9.: Simplified overview of the whole implemented system

The FPGA with the implemented system and the connected debugger, EvaLTE board and antenna is pictured in Fig. 4.10. Interface of Celeste with all its input and output signals can be found in Appendix A.

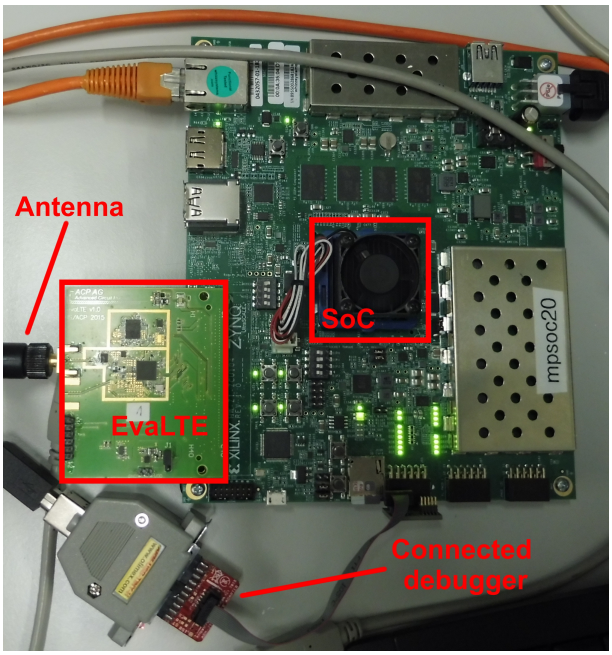


Figure 4.10.: Picture of the whole system during measurements

4.2.1. Basic Packet Processing

A library called *ble_common* was developed for easier use of Celeste. It contains the Celeste's address space definition, basic constants (lengths of the individual blocks of an advertising packet, initial values for whitening and CRC calculation, etc.) and essential functions for controlling Celeste and the received packet processing.

A software was written to allow measurements and the system's evaluation. The application is using two interrupts for receiving a message. The first interrupt occurs, when 56 LLRs are received. The interrupt routine reads this amount of the LLRs and based on the content of the PDU header it sets the number of LLRs to be received. When the core receives this number of the LLRs, the second interrupt occurs. The power valid flag is checked before reading the message. If the power is not valid, the packet is discarded and the core is started over again.

When the power flag is valid, the interrupt sets a flag signaling the main loop that a packet was received. The whole message in LLRs is read to the memory of the SoC from Celeste. LLRs are converted to bits and those are stored in 8 bit words and the message is whitened. The whitening sequence is calculated using *C* implementation of the LSFR and the data is XORed with it. The CRC is calculated afterwards using look-up table and if it matches the last three bytes of the message, the packet and corresponding power

value are used further. Bits of the packet must be reordered based on the information from Section 2.2 to match the RISC-V endianness and bit numbering.

4.3. Evaluation of the Integrated System

The successfully integrated system was first tested by the CMW500 RF tester using arbitrary waveforms generated by the Matlab framework. Reference BLE signals generated by the tester were used after primary evaluation of the function of the whole system. A simple application was developed for this evaluation and for the raw RSSIs measurements. This application detects a BLE packet, processes it and prints its content to the terminal. It also reads and prints calculated power of the received packet. There are two more functions that can be run from the terminal.

The first function called `ble_bler_measure` serves for the PMR and CRC error measurement. This function captures given number of packets and returns the PMR and CRC error rate. PMR can be calculated if the period, at which the packets are transmitted, is known. A missed packed is indicated if there was no packet received within the whole transmitting period plus the time of the packet itself.[2] The packet could be already detected, but it might not be stored in the memory yet, if only the transmitting period would be used.

The application also calculates the average power over the whole measurement.

Another function is `power_stability`, which measures the signal's power fluctuation over a longer time window. It captures packets in the chosen period, either from the testing equipment or the beacons. In case of the beacons measurement the packets can be captured in all the advertising channels separately. The function calculates their mean value and the standard deviation. When the function finishes, it reports the measured RSSIs over time in format that is easily transferable to Matlab for further processing.

4.3.1. CMW500 Measurement Over the Cable

The system was first tested over a cable with ca. 2 dB attenuation to estimate the mapping constant between the received RSSI and the calculated digital power value. It was observed that using power gain higher than 80 dB does not bring any improvement regarding the sensitivity.

The sensitivity of the system was evaluated by over the cable measurement using BLER characteristic. It is a sum of missed packets and CRC errors divided by the number of transmitted packets. The packets generated by CMW500 were transmitted with 50% duty cycle at the period of 656 µs for packets of 368 bits. The packet is considered as missed if it was not received within 1.5 multiple of this period. Results of these measurements can be found in Fig. 4.11. In the figure are measurements using 80 dB gain in all the three advertising channels. 10,000 packets were transmitted at each measuring point.

[2] One bit lasts 1 µs.

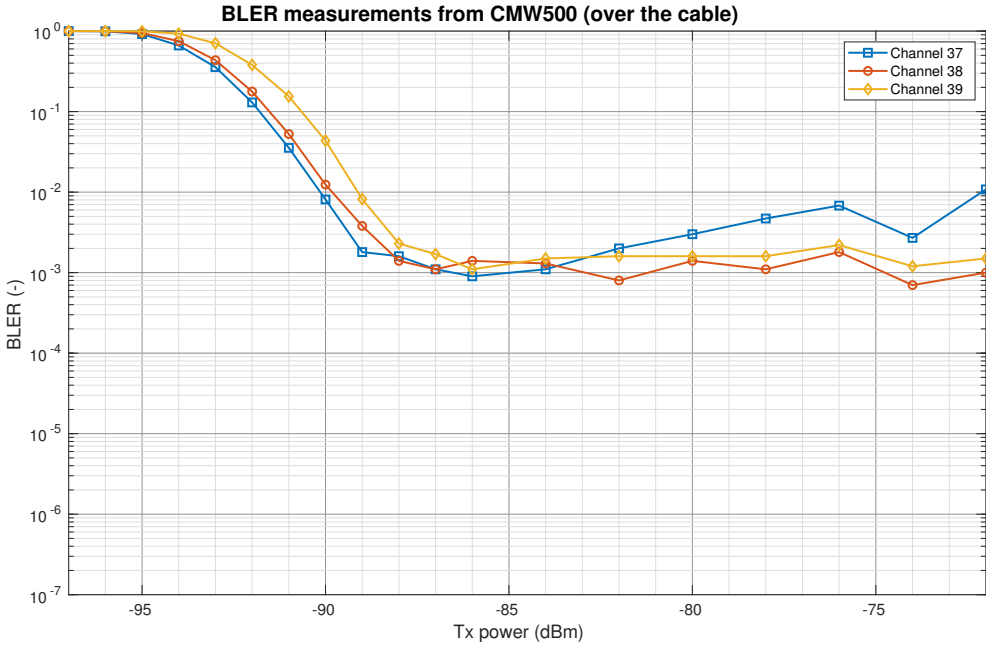


Figure 4.11.: BLER measurements in the three advertising channels with CMW500 over the cable

Using high gain for strong signal causes the RF IC saturation, which means the calculated power is not correct. The packet's power is considered as valid only within certain boundaries.

Three gain values (40 dB, 70 dB and 80 dB) that provide sufficient sensitivity range were chosen for simplification. An RSSI mapping constant c_{RSSI} was calculated from known cable loss L, transmitting power P_{tx}, calculated digital power P_{rxD} and used gain G (all values in dB) following Eq. (4.3).

$$c_{RSSI} = P_{rxD} - P_{tx} - G + L \tag{4.3}$$

This constant was calculated for the three gain values in the three advertising channels. The average values for each channel and gain can be found in Table 4.2. Small differences between the channels are probably caused by an uneven channel gain.

Gain (dB)	c_{RSSI} (dB)		
	Channel 37	**Channel 38**	**Channel 39**
40	50.82	50.64	49.65
70	51.17	50.94	49.87
80	51.15	50.66	49.72

Table 4.2.: Measurement of constant c_{RSSI} for mapping to RSSI

These values are calculated from measurements within P_{tx} range, where more than 99% of the transmitted packets were correctly received and P_{rxD} was not saturated. P_{tx} was stepped by 2 dB in this range and 10,000 packets were transmitted at each point. The average value used for the RSSI calculation is 50.51 dB.

4.3.2. CMW500 Measurement Over the Air

The sensitivity was evaluated also in over the air measurement. P_{tx} was stepped by 1 dB in "steep slope" part of the characteristic and by 2 dB in the flat part. 10,000 packets were transmitted at channel 37 at each point. The FPGA was placed ca. 1 m away from CMW500. The floor in the sensitivity characteristic is partly caused by presence of other advertisers within the measuring site. It was not possible to control their transmitting power or position, which means the measurement will be always partly violated by their presence.

Results depicted in Fig. 4.12 prove that the system is able to receive and process real BLE advertising packets and it is ready to be used in an positioning application.

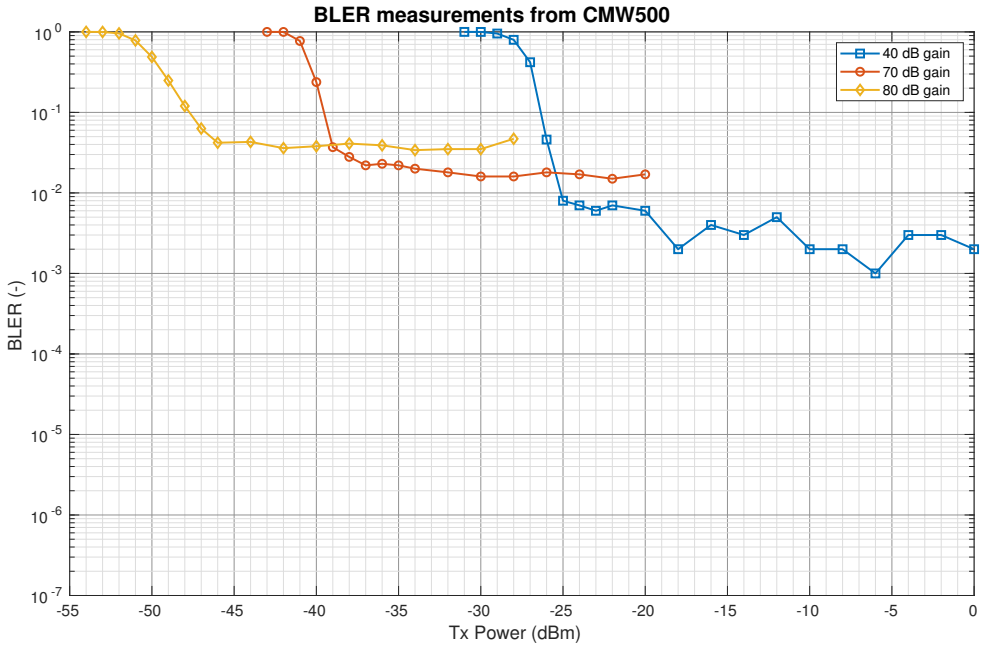


Figure 4.12.: Measurement of Celeste with CMW500 over the air

Chapter 5

Positioning Application Development

This chapter is focused on the positioning application description, which can be found in Section 5.1. The path loss model and the filtering approach were determined based on the RSSIs measurements reported in Section 5.2.

5.1. Positioning Application

The application was desinged on previously developed platform for the SoC, which is based in the Little Kernel [23].

The application is started from the terminal using the command `position`. The first argument after the command is an integer number of positions that should be calculated. The positioning application operates in seven states, which are depicted in Fig. 5.1. The states are within a loop, which can be ended only from END APP state.

The application enters its execution in the STARTUP state, where drivers of the physical layers are reset and the application sets the two used interrupts. The application starts Celeste, minor and global timeouts and transfers to the IDLE state. There is one minor timeout per beacon, which indicates if there are more than 4 seconds elapsed between reporting position and the last received packet from the corresponding beacon. The global timeout is implemented to control if there are no received packets at all. It is checked within the IDLE state and if there is more than 5 seconds from the last received packet the application transfers to the END APP state.

In the IDLE state the application waits for a received packet. Packets are received using the two interrupts as was described in the previous chapter. When the packet is received and the calculated power is valid, the whole message and calculated power are read from the Celeste core. A new gain value and a new frequency are set to the RF block and Celeste is started over again. The application steps through the three gain values (40 dB, 70 dB and 80 dB) within each advertising channel after each received packet. This approach was chosen to ensure that the RSSIs are received evenly from all the beacons and all the channels.

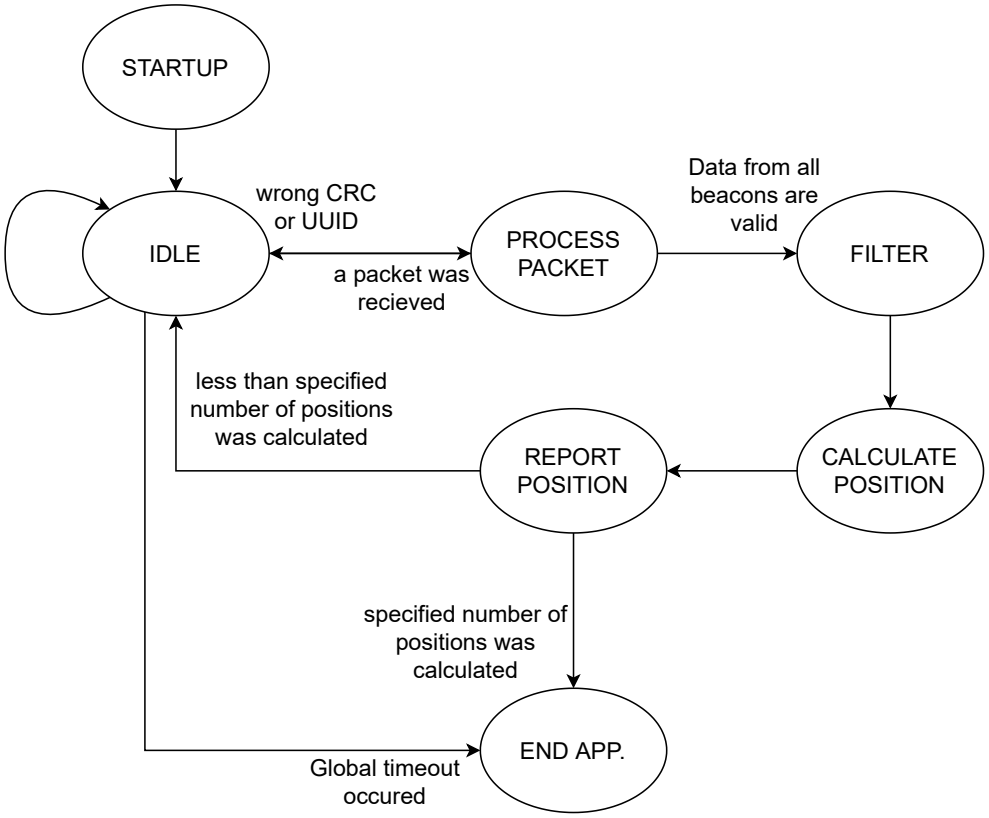


Figure 5.1.: Block diagram of the positioning application

The received packet is handled within the PROCESS PACKET, which was described in Section 4.2.1. If both the CRC and UUID are valid and the packets power is within range for the used gain, the RSSI value is calculated and saved to an array corresponding to the received minor number. The reference RSSI from the received packet is saved as well and the application then checks if there is enough available data to calculate the position.

It transfers to the FILTER state if 9 new RSSIs (three from each channel) are received from each beacon, since the last calculated position. If there is not enough new data, it goes back to the IDLE state.

The filtering within the FILTER state is described in Section 5.2.5. The application then transfers to the CALCULATE POSITION state, where the distance from each beacon is calculated and matrices $\boldsymbol{A}$ and $\boldsymbol{b}$ (see Eq. (2.8) and Eq. (2.9)) are formed from the known beacons' positions and the calculated distances. The position is estimated using the LLS algorithm and then the application goes to the REPORT POSITION state.

The minor timeouts are checked and reported. The position is printed to the terminal and it is saved to an array. The application is ended if the specified number of positions was received, otherwise it goes back to IDLE.

In END APP RF block is stopped and the application ends its operation.

5.1.1. Positioning Application Timing Analysis

Spending too much time on the calculations and packet processing could result in losing other packets. For that reason, the Celeste core is always started just after reading a new packet from its memory.

The time demanding processes are summarized in Table 5.1. The measured time includes functions' calls. The position calculation, which consists of five matrix operations, takes most of the time during the run of the application. The sum of all the demanding operations is 693 µs, which is sufficiently low considering the advertising period of 100 ms.

Process	**Time**
Packet processing	198 µs
Matrices formation	32 µs
LLS calculation	463 µs
SUM	693 µs

Table 5.1.: Summary of time demanding processes within the positioning application

5.2. Measurements with BLE Beacons

Preliminary measurements were done with BLE beacons designed for an indoor positioning application at IIS within a group project in 2019 [24]. They were used for functional verification of the developed FPGA testbed and for debugging of the whole positioning application.

In the end was decided to use commercial Minew E5 beacons [25] for the final experimental setup, because more of them were available. They are built on nRF52 IC series compatible with BLE 5.0. An Android and iOS application is available for their configuration. The type of the advertising protocol (iBeacon, Eddystone or URL) and the payload data can be configured in six independent slots. The advertising period and the transmit power can be configured as well.

5.2.1. Path Loss Measurement

The path loss of signal was first measured from 0.5 m to 9.0 m distance in half meter steps with the FPGA fixed and a moving beacon. Fig. 5.2 shows that the aggregation of channels gives more stable results than using the channels individually. A comparison between individual channels and their aggregation will be further investigated in Section 5.2.3.

The logarithmic characteristic of the fading shows that fluctuation of the RSSIs from the beacons that are further will result in a higher error in the distance estimation than the same fluctuation from the beacons that are closer.

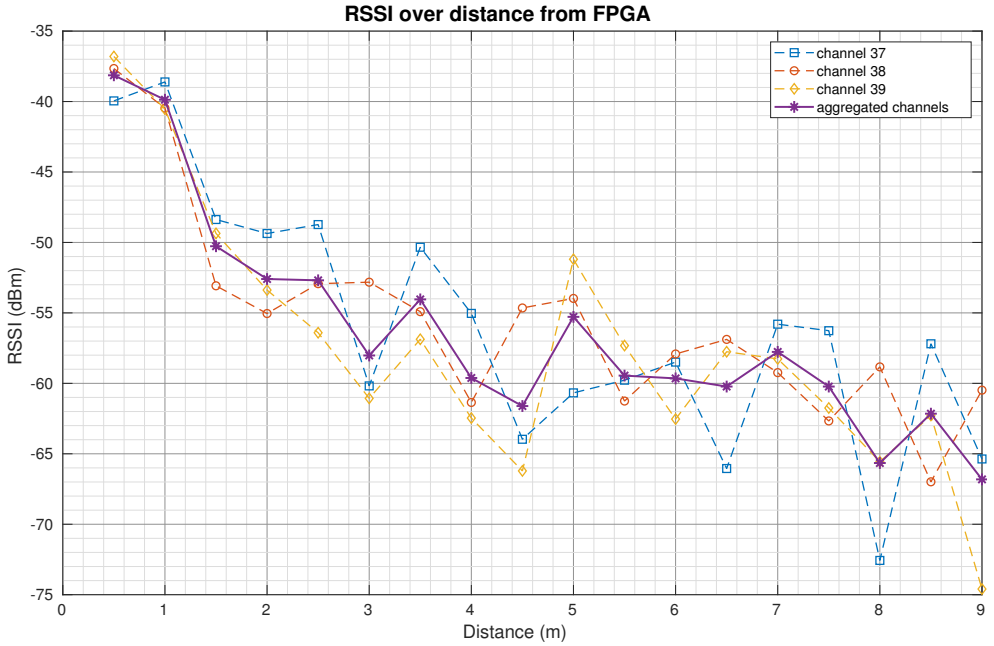


Figure 5.2.: Path loss measurement

5.2.2. Experimental Setup

The six Minew beacons were set in an office room of approximately 130 m^2. During the measurements the room was under usual use of three to nine people. The greatest obstacle considering the Line of Sight (LOS) were computer screens with their upper edge at a height of 130 cm. The beacons were placed on walls at a height of 200 cm and FPGA's antenna port at a height of 100 cm. Fig. 5.3 displays a simplified experimental setup. A detailed description with all dimensions can be found in Appendix B. The upper wall consists of three columns and large windows between them.The beacons were set to 0 dBm transmit power and the advertising period was set to 100 ms.

The beacons are marked as *Bx*, where *x* stands for their minor number. The rest of the marked points are the measuring positions. Index *a* identifies positions used for model determination and index *b* is used for control positions.

Next subsections describe RSSI measurements at *a* points. The goal of these measurements was to identify the RF environment, and based on that, choose the best approach for the distance calculation and the position estimation. Different behavior of the individual channels was observed during these measurements. Another studied aspect the stability of the environment, mainly the movement of the people within the room. Both of these attributes of the environment are described in Section 5.2.3.

The measurements used in Section 5.2.3, Section 5.2.5 and Section 5.2.6 run for one hour at the positions with index *a*. One packet was received from each beacon and each channel every 5 seconds. Only at the position *P1a* the measurement run for ca. 20 hours. These measurements had three goals - find the best filtering method, decide

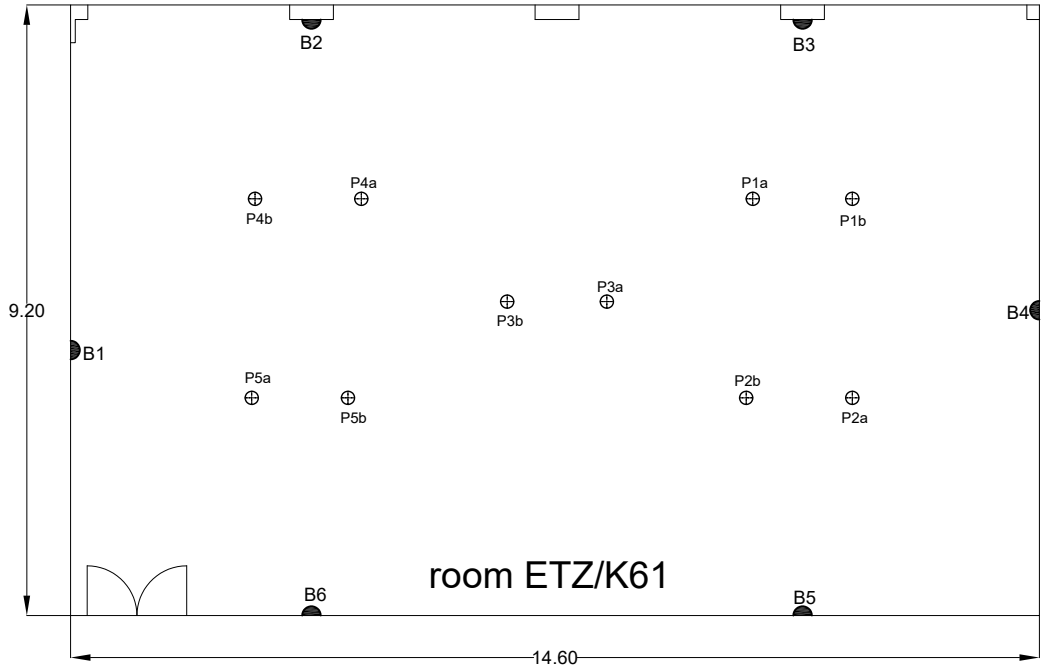


Figure 5.3.: Illustration of the experimental setup

which channel should be used for measuring the RSSIs and for the path loss model estimation.

5.2.3. Channel Differences and RSSI Fluctuation

Different behavior of the individual channels was described already in [5].

Fig. 5.4 shows the measurement from position *P1a* from beacon *B1*. Around 3 hours after the beginning of the measurement the environment became stable and stayed so for ca. 8 hours. Considering that the measurement was started at 9 pm., it implies that the people using the room have the greatest impact on the measured RSSIs. In the first three hours of the measurement a delay in capturing packets caused by waiting for the valid packets from all the channels and all the beacons is visible.

RSSIs of the channels fluctuate by ca. 0.5 dB around a stable value in the still environment. The difference between the channels is around 2 dBm. In the unstable environment the difference between the channels at one moment can be up to 10 dBm. Considering logarithmic model described in Section 2.4 such fluctuations could cause error of several meters in distance estimation.

Behavior described above is even more visible in Fig. 5.5, which is the same measurement for beacon *B2*. This beacon was closer, but it was in Non Line of Sight (NLOS). It as apparent from these two figures that unstable environment, the beacons in NLOS and the differences between the channels will be the greatest challenges for the position estimation.

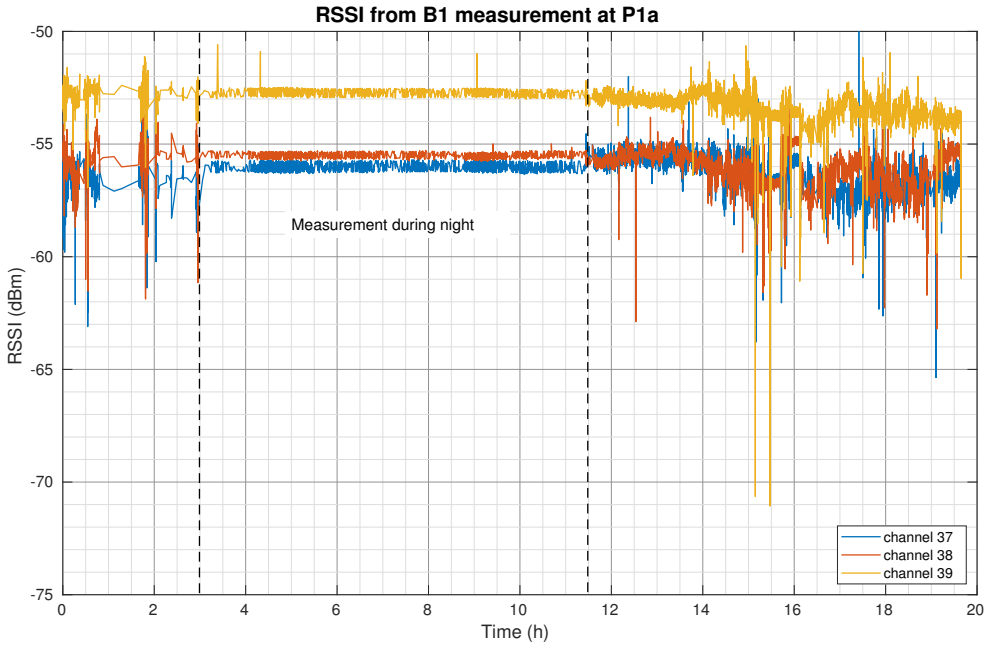


Figure 5.4.: RSSI received from *B1* in overnight measurement

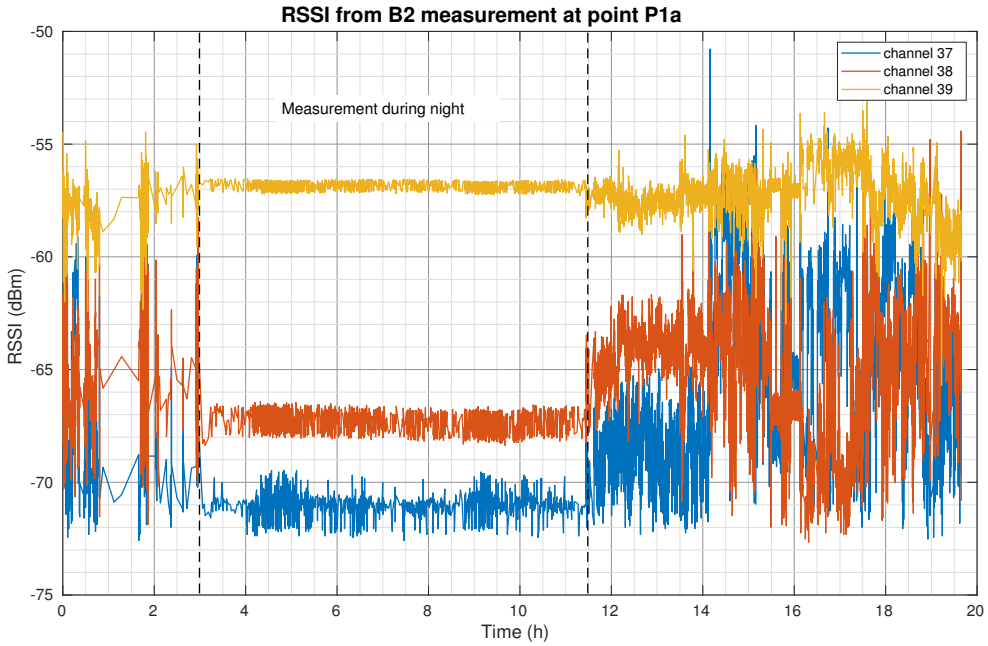


Figure 5.5.: RSSI received from *B2* in overnight measurement. Beacon was in NLOS.

Aggregated Channels Compared to a Single Channel

The same measurements as were described before were used to find out if using measurements from all the channels evenly would bring better results than using measurements in a single channel. The differences between minimal and maximal value within the whole measurement for each channel separately and the aggregated channels are compared in Table 5.2. The data in the table is from the measurement at point *P3a*, which is almost in the middle of the room. A moving mean window of 3 samples was used on the individual channels to make them comparable with the aggregated channels.

The highest difference between the minimal and maximal value for each channel and the aggregated channels were compared to choose, which measurement has the highest potential. The highest value also means the greatest error in the distance estimation. The analysis from the rest of the measuring points gave similar results.

	Min/Max Difference			
Beacons	**Channel 37**	**Channel 38**	**Channel 39**	**Aggregated channels**
	(dB)	(dB)	(dB)	(dB)
B1	3.80	3.24	2.83	3.79
B2	**9.75**	4.71	4.28	9.15
B3	8.53	6.60	6.10	6.41
B4	9.04	**9.83**	6.30	**9.30**
B5	6.19	7.04	3.82	5.08
B6	7.69	5.93	**7.34**	8.68

Table 5.2.: Differences between the minimal and maximal RSSI value during the measurement at position *P3a* for the different channels

The highest value of the difference between minimal and maximal value in each column is in bold letters. The values in Table 5.2 show that the best result provides the channel 39 and the second best result is in "Aggregated channels" column. Those two will be used in the future measurements.

5.2.4. Preliminary Constrained Model

The first approach to establish a path loss model was to measure the signal path loss within the setup, similarly as it was done in Section 5.2.1. Parameters $RSSI(d_0)$ and n of the path loss model in Eq. (5.1) had to be determined based on the measurements. d_0 is 1 m distance from the beacon.

$$RSSI(d_i) = RSSI(d_0) + 10n \log\left(\frac{d_i}{d_0}\right) \tag{5.1}$$

The path loss was measured between *B3* and *B5* already installed in the setup. Sorted

arrays of RSSIs and distances were provided to the Matlab function `fit`. The model with two unknowns was specified using the `fittype` function.

Listing 5.1: Path loss modeling in Matlab

```
ft = fittype('RSSI_d0-10*n*log10(x)');
f = fit(l',RSSI',ft,'StartPoint',[-40 3]);
```

Model given by this measurements was further optimized at position *P1a*.

$$MODC : RSSI(d_i) = -42 + 10 \cdot 1.380 \cdot \log(d_i) \tag{5.2}$$

Two limiting assumptions were done for this model. First, maximal valid RSSI is -60 dBm due to the maximal possible distance from any beacon within the room. And second, if the estimated position is outside of the room, up to three beacons with the weakest signal strength are excluded from the calculation. If the estimated position is still outside of the room all received RSSIs are flushed and the measurement starts over again. These constraints were chosen in order to try to achieve the highest possible accuracy. As this model has a very limited validity for a single room only, a more general approach for model determination is described in Section 5.2.6.

5.2.5. Filtering

The reliability of the position estimation can be improved by applying a relevant filtering method on the measured RSSIs. Several moving window filters were tested to find the best approach.

The length of the window was based on the preliminary measurement using the constrained model (Eq. (5.2)) in point *P1a*. Analysis from the other points gave similar results. A moving mean filter with different lengths was applied on samples, but it was assumed that similar results regarding window length would be achieved with any other type of the filter. Based on the results depicted in Fig. 5.6, it was decided to use moving window of 30 samples from each beacon as it seems to be robust enough and still acceptably short. All the data would be exchanged once in 3 seconds in the ideal case considering the advertising period is 100 ms. This time will probably be longer considering the random delay after each packet, BLER in Fig. 4.12 and the fact that the packet does not have to be received with the valid power.

The measured data was filtered by a mean, median, maximum and 75% moving windows. The RSSIs measurements from aggregated channels are used in this evaluation.

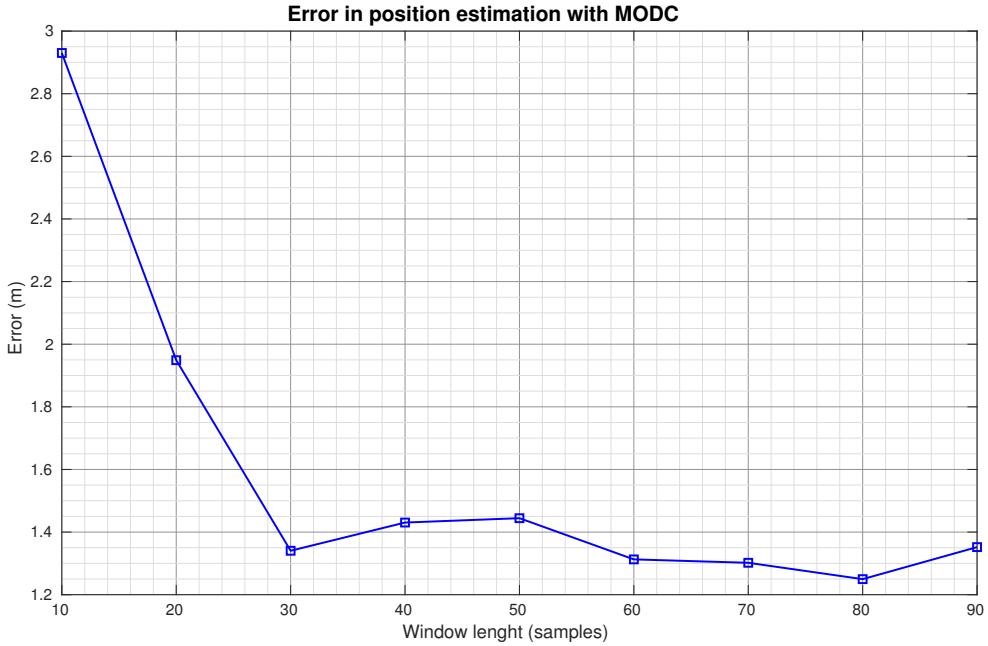


Figure 5.6.: Dependency of an error in the position estimation on the length of the moving mean filter

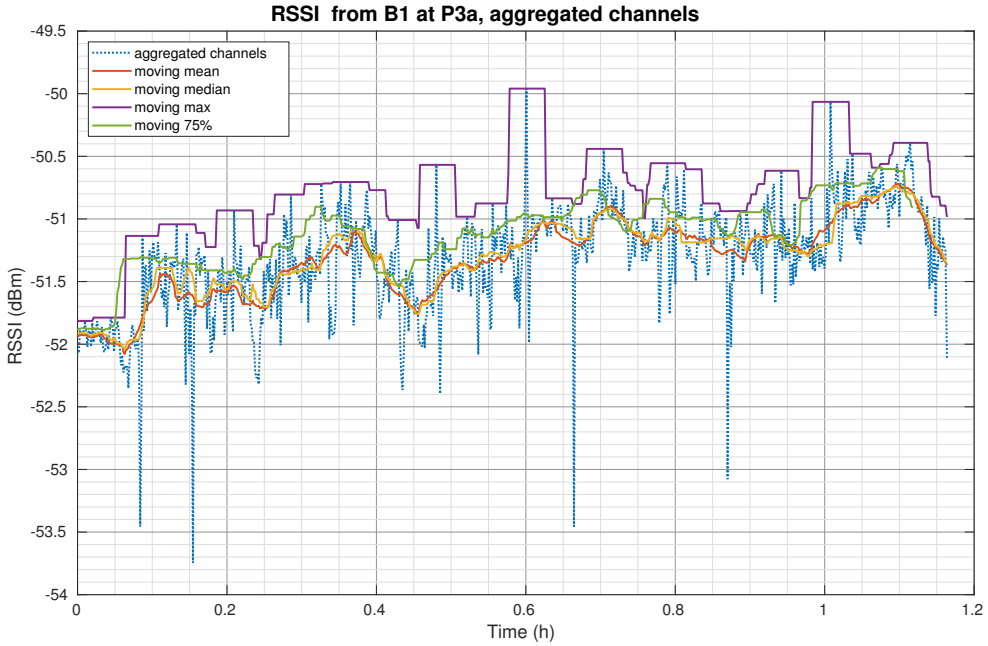


Figure 5.7.: Filtered RSSI from the aggregated channels and beacons *B1* at the position *P3a*

Fig. 5.7 shows an example of the original and the filtered data. The dotted curve is the raw aggregated channels' RSSIs. The moving median and moving mean gave comparable results. The moving maximum is very stable during long periods, but it is susceptible to very short spikes within the data, which could cause high error.

Results of the analysis for all the beacons at position *P3a* are summarized in Table 5.3. The filtered data were classified using the difference between minimal and maximal value of the whole measurement.

	Min/Max Difference			
Beacons	**Mean**	**Median**	**Maximum**	**75%**
	(dB)	(dB)	(dB)	(dB)
B1	1.36	1.30	1.86	1.30
B2	1.10	1.01	1.43	0.93
B3	1.42	1.39	1.72	1.38
B4	1.30	1.01	1.58	0.95
B5	**1.74**	**1.74**	2.00	**1.85**
B6	1.63	1.51	**3.91**	1.68

Table 5.3.: Differences between minimal and maximal RSSI value during the whole measurement for the different moving filters

The measurements from the other points gave similar results for the average and the mean moving filter. The mean moving filter was chosen because it provided a better result in 3 out of 5 measurements. The moving median filter would be a valid choice as well.

5.2.6. Path Loss Model Determination

The logarithmic path loss model was determined based on the measurements described before. The mean values of the RSSIs over the whole measurement were used for the model determination.

There were chosen two approaches for the model determination. First, data from all the beacons was included to the model calculation using the previously mentioned Matlab functions and second, the data from a beacon that is the least reliable were excluded from model calculation.

Altogether 20 calculated models, one for each channel and aggregated channels at each position, are summarized in Table 5.4. It can happen that some of the measurements were so influenced by fading that the model calculated by Matlab was not following expected logarithmic characteristic. These models were excluded from the final model determination.

Positions	Channel 37		Channel 38		Channel 39		Aggregated channels	
	$RSSI(d_0)$ (dBm)	n (-)	$RSSI(d_0)$ (dBm)	n (-)	$RSSI(d_0)$ (dBm)	n (-)	$RSSI(d_0)$ (dBm)	n (-)
P1a	*-54.62*	*0.541*	*-55.14*	*0.324*	*-54.66*	*0.052*	*-54.80*	*0.306*
P2a	**-42.34**	**1.517**	*-45.59*	*1.086*	**-37.47**	**2.259**	**-41.80**	**1.620**
P3a	*-53.52*	*-0.326*	*-53.37*	*-0.456*	**-45.18**	**1.148**	*-50.96*	*0.122*
P4a	**-38.72**	**1.965**	*-57.20*	*-0.591*	**-40.59**	**1.833**	*-45.50*	*1.069*
P5a	**-42.20**	**1.179**	**-37.67**	**1.614**	**-41.28**	**1.701**	**-40.38**	**1.498**

Table 5.4.: Summary of the models acquired at the five different measuring points

The numbers in the bold letters are valid model coefficients. As mentioned above, the models with $n < 1.100$ were excluded from the model determination.

The most of the valid models were found in the channel 39. The mean value of $RSSI(d_0)$ (-41 dBm) was set to the `fittype` function instead of $RSSI(d_0)$ and n was recalculated. The final model used mean of the recalculated n (see Eq. (5.3)).

$$MOD1 : RSSI(d_i) = -41 + 10 \cdot 1.761 \cdot \log(d_i) \tag{5.3}$$

The models acquired from the channel 37 and the aggregated channels were calculated in a similar way and they provided very similar results. For that reason only the model given by the aggregated channels was included in the future measurements. This model is specified by Eq. (5.4).

$$MOD2 : RSSI(d_i) = -41 + 10 \cdot 1.520 \cdot \log(d_i) \tag{5.4}$$

The second approach was to exclude from the datasets one beacon per channel with the highest standard deviation at the particular position. The results of the model fitting for such a situation are given in Table 5.5.

Positions	Channel 37		Channel 38		Channel 39		Aggregated channels	
	$RSSI(d_0)$ (dBm)	n (-)	$RSSI(d_0)$ (dBm)	n (-)	$RSSI(d_0)$ (dBm)	n (-)	$RSSI(d_0)$ (dBm)	n (-)
P1a	*-54.91*	*0.308*	*-55.44*	*0.090*	*-65.78*	*-1.193*	*-60.02*	*-0.278*
P2a	**-31.73**	**2.540**	*-46.74*	*1.024*	**-38.20**	**2.219**	**-43.66**	**1.521**
P3a	*-53.87*	*-0.368*	*-48.70*	*0.061*	*-48.84*	*0.570*	*-51.73*	*-0.043*
P4a	**-40.75**	**1.595**	**-40.11**	**1.326**	**-44.46**	**1.192**	**-44.48**	**1.255**
P5a	*-44.71*	*1.040*	**-37.99**	**1.541**	*-44.53*	*1.073*	**-40.06**	**1.572**

Table 5.5.: Summary of the models acquired at the five different measuring points with one beacon excluded

The models with $n < 1.100$ were again excluded from the model determination. The most of the valid coefficients were obtained from the aggregated channels. The model was established the same way as mentioned previously.

$$MOD4 : RSSI(d_i) = -43 + 10 \cdot 1.329 \cdot \log(d_i) \tag{5.5}$$

It seems from Table 5.4 that the coefficients given by channel 39 could provide good results. For that reason, the model from the channel 39 in Table 5.5 was used as well. For n coefficient calculation was used $RSSI(d_0)$ value from Eq. (5.5).

$$MOD3 : RSSI(d_i) = -43 + 10 \cdot 1.479 \cdot \log(d_i) \tag{5.6}$$

All the determined models are summarized in Table 5.6.

Designation	$\boldsymbol{RSSI(d_0)}$ (dBm)	$\boldsymbol{n}$ (-)	**Source**	**Note**
MOD1	-41	1.761	channel 39	all beacons used
MOD2	-41	1.520	aggr. channels	all beacons used
MOD3	-43	1.329	channel 39	one beacon excluded from calc.
MOD4	-43	1.479	aggr. channels	one beacon excluded from calc.
MODC	-42	1.380	path loss	constrained (see Section 5.2.4)

Table 5.6.: Summary of all the used models

5.2.7. Beacons Redundancy

The beacons in NLOS and the unstable environment have a significant negative impact on the RSSI stability. Especially the beacons in NLOS mean always higher signal attenuation, and thus considerably longer distance is estimated than it actually is. It could be beneficial to exclude the unreliable data from the position estimation as they do not improve its accuracy, more likely they increase the error.

Two different approaches were evaluated. Within the CALCULATE POSITION state can be excluded data from up to two beacons based either on the standard deviation of the RSSIs or those with the weakest signal strength.

Chapter 6

Results

The goal of this chapter is to describe the final evaluation of the positioning capabilities of the whole system. As was mentioned at the beginning of this thesis the main emphasis is laid on the position estimation accuracy and the position calculation rate. Both of these characteristics are described in the two following sections. The results achieved during these measurements are compared to the state of the art in Section 6.4. All the following measurements were done with the commercial beacons advertising in 100 ms intervals with 0 dBm transmit power.

6.1. Position Estimation Accuracy

The accuracy of the position estimation is a key characteristic of any IPS. Besides searching for the best model for the given environment, different approaches were considered to improve the positioning accuracy:

- Using a single channel comparing to using all the channels evenly
- Excluding the measured data based on the standard deviation or the strength of the signal
- Constraining a model to a specific room

The rest of the chapter is divided into sections corresponding to the outlined approaches.

All of the following measurements were done in a complex RF environment, when the room was under usual use of up to 9 people. The only exception is the constrained model, of which performance was measured only in presence of the operator.

6.1.1. Channel 39 Versus Aggregated Channels

Preliminary evaluation confirmed that using all the channels evenly provides higher accuracy than using a single channel. The goal of these measurements was also to

exclude some of the path loss models from the final evaluation.

The channel 39 was chosen for this comparison because two out of five models were determined based on measurements in this channel. The experimental setup is identical with the one described in Chapter 5. Only the measurements from points *P1a*, *P2a* and *P3a* were used during this preliminary evaluation. At each point was the position calculated 24 times. Constrained model was not considered in this evaluation.

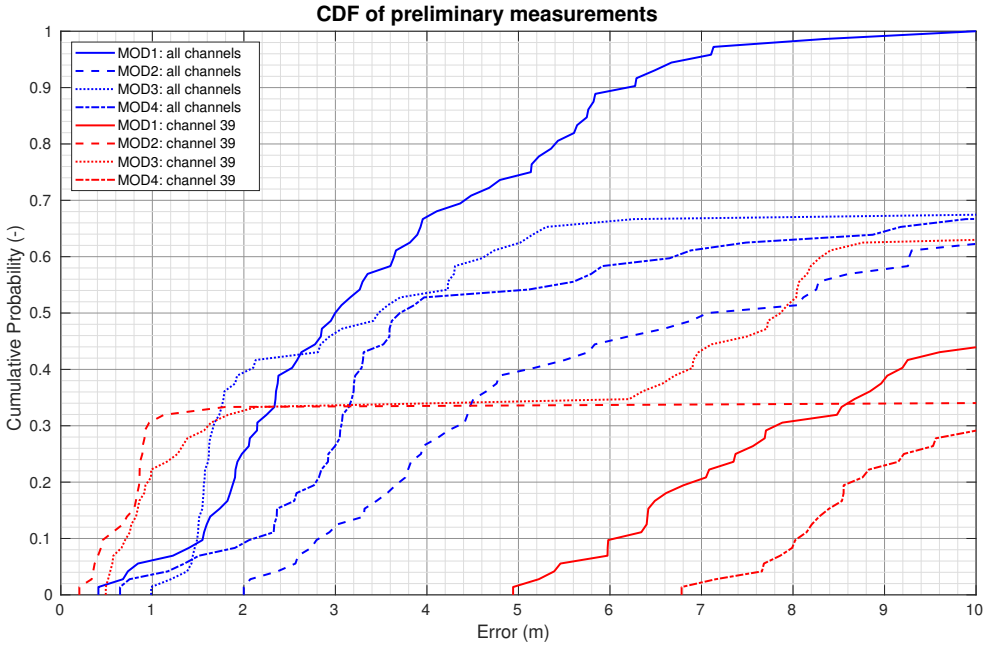


Figure 6.1.: CDF of all the models in the channel 39 and aggregated channels

The CDFs of these measurements are depicted in Fig. 6.1. It is clear from this figure that using all the channels provides overall higher and more stable accuracy than using the channel 39 separately.

Even though MOD1 and MOD3 were calculated from the measurement in the channel 39, they do not provide better results than the aggregated channels using the same model. It shows that the measurements in channels 37 and 38 had significantly lower reliability than the measurements in the channel 39, which were used for MOD1 and MOD3 calculation.

Based on these preliminary measurements only the aggregated channels and models MOD1 and MOD3 were considered for the subsequent evaluations.

6.1.2. Excluding Beacons From Position Calculation

Excluding unreliable beacons from the position calculation is a very simple approach, that comes with minimal costs considering the power consumption and the software complexity. It was considered to exclude up to two beacons based on:

1. the highest standard deviation
2. the weakest signal strength

Excluding Beacons Based on the Standard Deviation

It was decided to try to exclude data based on the standard deviation, because the fluctuation of measured RSSI significantly increases if the beacon is in NLOS or in an unstable environment.

The position was estimated in all the 10 measuring points marked in Fig. 5.3. The results for the control points only can be found in Appendix C. The position was calculated at each point 99 times.

Fig. 6.2 illustrates that excluding one or two beacons based on the standard deviation provides higher accuracy within MOD1. The results are not very convincing within MOD3. This could be caused by very unstable environment, where more than data from two beacons was significantly violated. The accuracy is still increased in the lower part of the plot. It can be said that excluding beacons based on the standard deviation of RSSIs can bring higher accuracy, but this increase is not very significant. The position estimation accuracy is worse in Fig. 6.2 than in Fig. C.1, but the overall assumption is the same

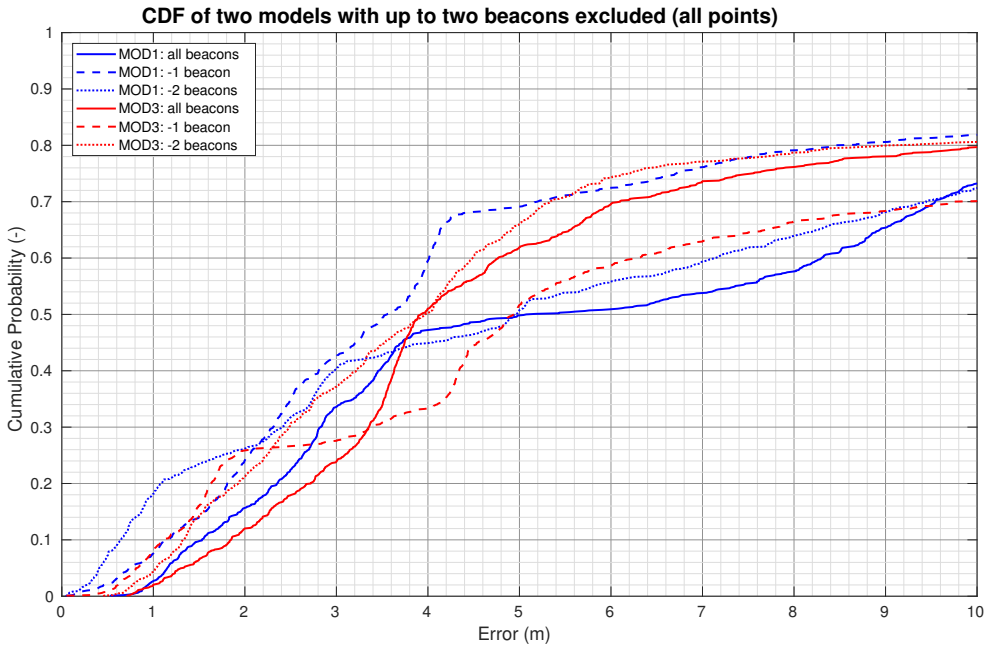


Figure 6.2.: CDF of MOD1 and MOD3 with up to two beacons excluded from position calculation based on the highest SD of RSSI. Measurements in all the 10 points.

Improvement in the position estimation accuracy, while excluding beacons based on

the standard deviation compared to using all beacons is summarized in Table 6.1. The measurements from all the ten points were used in this comparison. The negative value means the exclusion brought worse accuracy than using all the six beacons. The results in Table 6.1 shows that excluding beacons based on the highest standard deviation is not reliable approach. The mean error was improved only in MOD1 by 2%, while excluding two beacons with the highest standard deviation.

	μ	50%	95%
	(%)	(%)	(%)
MOD1: -1	-7	30	-86
MOD1: -2	4	2	-10
MOD3: -1	-2	-26	40
MOD3: -2	-1	-2	-8

Table 6.1.: The effect of excluding beacons from position calculation based on the highest SD relative to using all the beacons

Excluding Beacons Based on the Signal Strength

Similar measurements as in Section 6.1.2 were repeated with up to two beacons with the weakest signal excluded.

Fig. 6.3 depicts CDF of these measurements in all the 10 points. CDF of only the control points only can be found in Fig. C.2. The accuracy in position estimation is significantly improved in both MOD1 and MOD3 by excluding one or two beacons with the weakest signal. In Fig. C.2 the accuracy of MOD3 is improved up to 60% of all the cases. The drop in the rest of the characteristics can be explained by unstable environment, where the data from more than two beacons was violated.

The overall accuracy in Fig. 6.3 is worse than in Fig. C.2, but it supports the previously described assumption.

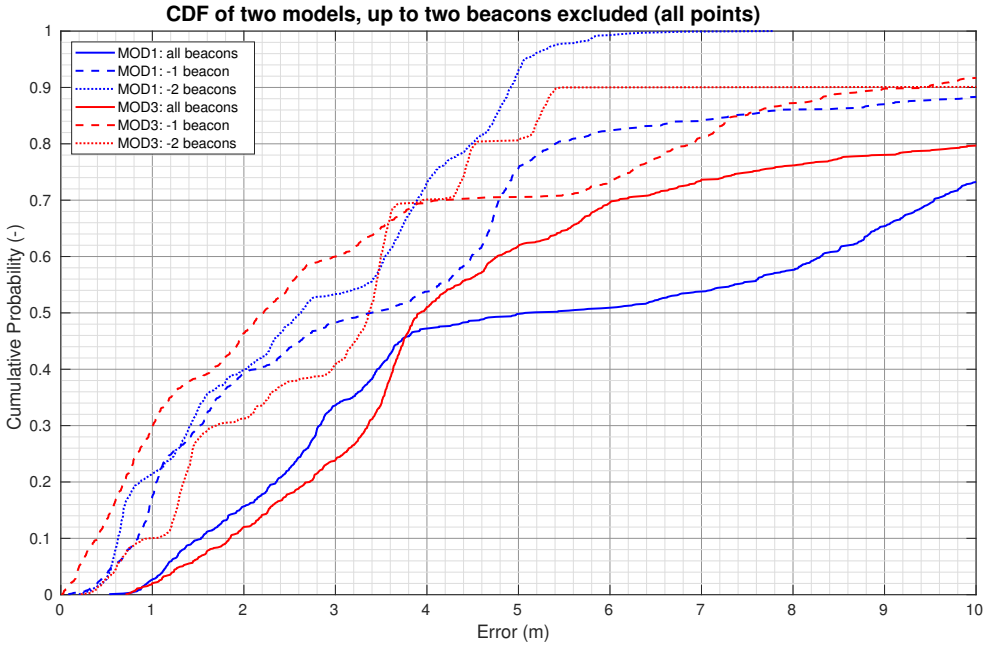


Figure 6.3.: CDF of MOD1 and MOD3 with up to two beacons with the weakest signal excluded from the position calculation. Measurements in all the 10 points.

The comparison of the achieved results relative to usage of all the beacons is summarized in Table 6.2. The measurements from all the 10 points were used in this comparison. The accuracy is significantly improved for all the configurations. The results are much better while excluding two beacons in MOD1. The number of the excluded beacons does not seem to be as important in MOD3.

Excluding beacons based on their signal strength proved to be reliable approach to improve positioning accuracy.

	μ	**50%**	**95%**
	(%)	(%)	(%)
MOD1: -1	35	33	2
MOD1: -2	60	48	68
MOD3: -1	57	43	66
MOD3: -2	54	14	62

Table 6.2.: The effect of excluding the beacons with the weakest signal from the position calculation relative to using all the beacons

Comparison of the Two Exclusion Approaches

Fig. 6.2 and Fig. 6.3 show that the position estimation accuracy can be significantly improved by excluding some data from the position calculation. Excluding data based on the signal strength proved to be a better approach than excluding the data based on the standard deviation. It could be caused by a too short filtering window, which is not able to capture a stronger signal fluctuation.

A comparison for the measurements in only the control points can be found in Appendix C.

The three best results from the two approaches for MOD1 are compared in Fig. 6.4. The accuracy of all the measurements is lower than considering only the five control points. A stable improvement of the accuracy was not observed, while excluding the data based on the standard deviation. The accuracy improved by 2.40 m in 50% of all the cases, while excluding two beacons based on the weakest signal strength.

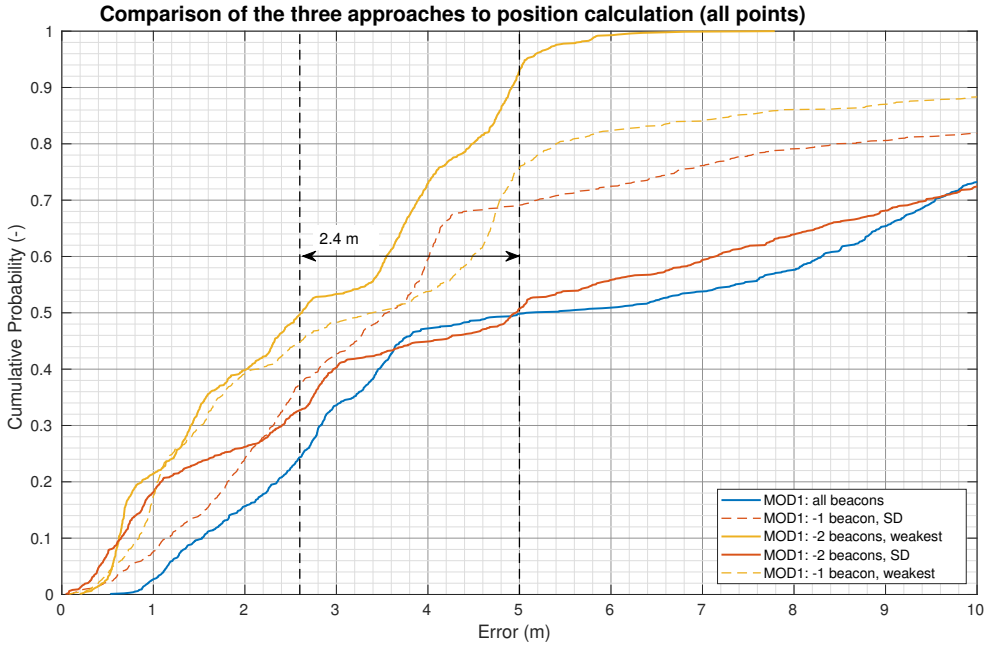


Figure 6.4.: Comparison of the three different approaches to the position estimation. Measurements in all the 10 points.

These results show that the beacons with a stronger signal have a higher reliability and that it is beneficial to exclude some of the beacons from the position calculation if there is more of them available.

6.1.3. Comparison of the Constrained and the Unconstrained Models

The constrained model was described in Section 5.2.4. In order to try to achieve the highest possible accuracy this model was measured without any other people present

in the room. Both of the models were measured in all the 10 positions marked in Appendix B. The unconstrained model was measured during the week, while the room was under regular use.

A comparison of both of the configurations in only the control points (seeFig. C.4) shows that MODC does not provide higher accuracy. In case we consider measurements from all 10 points (see Fig. 6.5) the constrained configuration demonstrates slightly higher accuracy than the unconstrained one. MODC does not change much within this two plots. The unconstrained configuration, on the other hand, has lower accuracy considering all the 10 measurements. This could mean that constraining provides more robust solution, but it compromises the overall accuracy. It also seems that MODC is not as reliable as MOD1. Constraining comes with high costs regarding portability of the system and calculation rate (see Section 6.2).

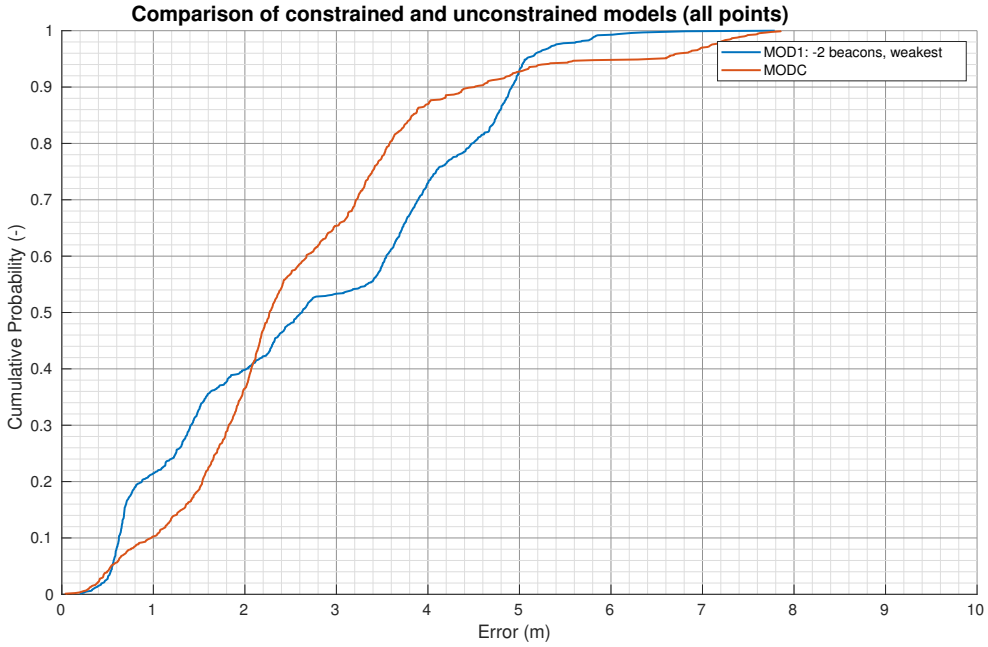


Figure 6.5.: Comparison of the constrained and the unconstrained model (configuration). Measurements in all the 10 points.

	μ (m)	50% (m)	90% (m)	95% (m)
MOD1: -2, weakest	2.75	2.63	4.92	5.08
MODC	2.65	2.29	4.51	6.44

Table 6.3.: Comparison of the constrained and unconstrained models (configurations)

The achieved results of the best unconstrained and constrained configurations are summarized in Table 6.3. Only the measurements in all the ten points are considered.

6.2. Position Calculation Rate

The position calculation rate is a key characteristic describing the tracking ability of the system. The elapsed time was saved after each calculated position.

The mean time needed for one position calculation in the unconstrained models is 4.4 s and for the constrained model it is 11.2 s. This is caused by excluding all $RSSI < -60$ dBm.

This result means that the constrained model (MODC) does not have the ability to track the devices almost at all, while the unconstrained model provides a sufficient calculation rate for real-time tracking of very slow devices. The calculation rate could be improved by using shorter length of the filtering window which would compromise the accuracy.

6.3. Real-time Tracking Evaluation

A demonstration of real-time tracking of the FPGA over the room was prepared as a part of the project evaluation. The prepared *Python* script is reading the calculated positions from the FPGA trough a serial connection. The script tries to read position each 0.5 s. When a new position is available, it is immediately plotted. Fig. 6.6 illustrates an example output of the tracking using MOD1 with two beacons with the weakest signal excluded from the position calculation. The blue curve is an approximate track of the device and the red curve is the calculated track. The installed beacons are depicted in the figure as well. During the demonstration was observed that the position is estimated each 2 s to 6 s.

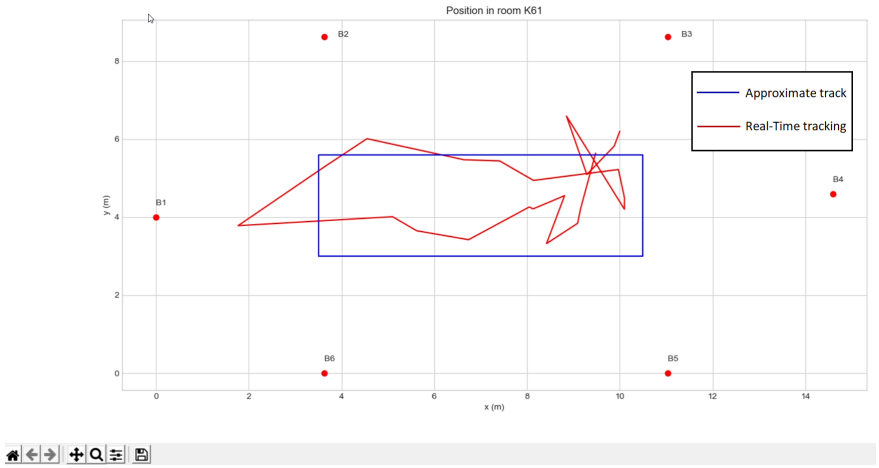


Figure 6.6.: Example output of the tracking application

6.4. Comparison to Related Work

The unconstrained model achieved mean error 2.75 m. This value is comparable to the state of the art described in [4].

The achieved error of < 5.1 m in 95% of all the cases is approximately two times worse than the two mentioned researches [5] and [7]. Both of the researches mention comparable beacons densities. Acheived results are compared to these two publications in Table 6.4.

	μ	90%	95%	Method	Density of beacons
	(m)	(m)	(m)		(m^2 per beacon)
[5]	ca. 1.2	2.0	2.5	fingerprinting	39
[7]	N/A	2.4	ca. 2.6	WLLS	27
MOD1: -2, weakest	2.8	4.9	5.1	LLS	22

Table 6.4.: Comparison of suggested approach to the state of the art

The first mentioned publication ([5]) is using the fingerprinting method for position calculation, but it does not provide any information on the instability of the environment caused by other people using the same area.

In [7] they used the WLLS algorithm and they were classifying the quality of the RSSIs based on their variance. This approach did not provide as good results in the setup used within this project as classification based on the signal strength. It was also observed that the standard deviation of the RSSIs is lower in aggregated channels than in separate channels, while [7] observed the opposite. In general it is difficult to compare researches on indoor positioning systems done in very different environments.

The measurements in this chapter were done in a real life environment (with exception mentioned in Section 6.1.3), where the movement of people in the room had the main influence on RSSI stability. From the studied researches such an environment was reported only in [7].

The calculation rate was considered in [5], where they studied filtering windows of 0.5 s to 2.0 s, which is faster than calculation rate that was achieved by this project.

The calculation rate is also a big issue of the current work. The application is able to calculate real time position only for very slow devices.

Chapter 7

Conclusion and Future Work

All the main goals of this project were achieved. The Celeste Bluetooth baseband accelerator was modified based on the Matlab simulations. BER and PMR were considerably improved by the described changes. These modifications were implemented in hardware and adapted Celeste was integrated to a PULP based SoC and the whole system was implemented on an FPGA. The whole system was successfully evaluated by measurements with the CMW500 RF tester and commercial BLE beacons. A library in *C* programing language was developed for simpler controlling of Celeste.

An application was designed in *C* to verify the system's positioning capabilities. The whole system was tested in a setup with the commercial beacons in a complex RF environment. The challenges negatively affecting the positioning accuracy were identified based on the overnight RSSI measurements. The main issues from this point of view were differences between the channels, beacons in NLOS and an unstable RF environment. The effect of the differences between the channels was reduced by using all the channels evenly.

Two different approaches were considered to improve the position estimation accuracy. Excluding two beacons with the highest standard deviation of the RSSIs brought improvement of 2% in the mean error comparing to using all the available beacons. This approach did not proved to be reliable. The mean error in the position estimation was improved by 60% by excluding two beacons with the weakest signal strength from position calculation. It is a significant improvement, especially, when considering very simple implementation of such a decision in the positioning algorithm.

A model constrained to a specific room was compared to the conventional approach. It provided mean error improved by 4% at cost of 2.5 times lower calculation rate and lower portability of the system.

The achieved mean error of the best unconstrained configuration was 2.75 m. The error was < 5.08 m in 95% of all the cases. The constrained configuration achieved mean error 2.65 m. The error of this configuration was < 6.44 m in 95% of all the cases. The achieved mean error is comparable to the state of the art, but such a comparison is difficult considering different experimental setups. The achieved values are sufficient

for a low cost IPS.

The Celeste core requires further optimization regarding chip area and energy consumption. Hard bit decision, whitening and CRC calculation should be moved to the hardware, which could save considerable amount of the processing time and energy.

The accuracy could be further improved by employing more complex algorithms for the position calculation. Separate information from individual channels could be used to improve the accuracy as it was suggested in [7]. It would be beneficial to do a long-term evaluation of the suggested approach and verify its portability to another environment.

Appendix A

Celeste's APB registers

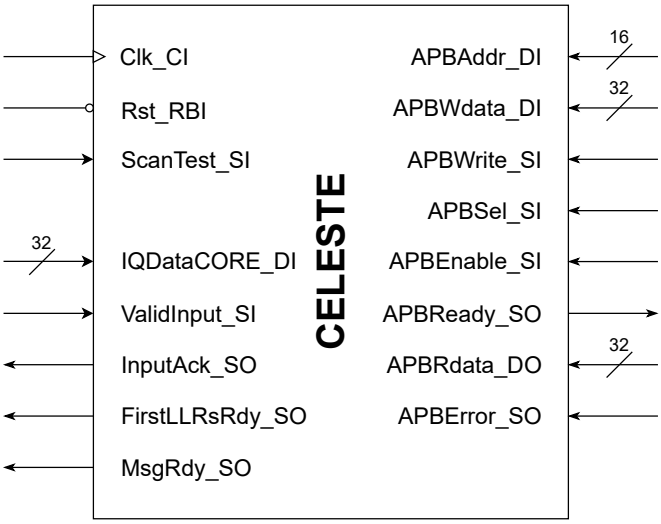


Figure A.1.: Celeste entity as it was integrated to the SoC

Address	Name	Reset value	Qm.f	Description
0xFF00	BLE_SYNCH_RES			Write 0x01 for synchronous reset
0xFF01	BLE_START_SEARCH			Write 0x01 to start BLE core
0xFF02	BLE_PREAMBLE_LOW	0x00000000		Lower 32 bits of synch. word
0xFF03	BLE_PREAMBLE_MID	0x7D917100		Middle 32 bits of synch. word
0xFF04	BLE_PREAMBLE_UP	0x0000556B		Upper 16 bits of synch. word
0xFF05	BLE_PREAMBLE_MASK_LOW	0x00000000		Lower 32 bits of synch. word mask
0xFF06	BLE_PREAMBLE_MASK_MID	0xFFFFFF00		Middle 32 bits of synch. word mask
0xFF07	BLE_PREAMBLE_MASK_UP	0x0000FFFF		Upper 16 bits of synch. word mask
0xFF08	BLE_FIR_COEF_LOW	0x22100FF0	sQ0.4	Lower eight FIR coefficients
0xFF09	BLE_FIR_COEF_MID	0x00122333	sQ0.4	Middle eight FIR coefficients
0xFF0A	BLE_FIR_COEF_UP	0x000000FF	sQ0.4	Upper three FIR coefficients
0xFF0B	BLE_COARSE_THR	0x672	sQ7.7	Threshold for coarse preamble search
0xFF0C	BLE_PACKET_LEN	0x4290	uQ15.0	Expected length of BLE packet
0xFF0D	BLE_SIG_PWR	0x0	uQ20.0	Calculated signal power, read only
0xFF0E	BLE_NOISE_PWR	0x0	uQ20.0	Calculated noise power, read only
0xFF0F	BLE_SET_BIST	0x0		Set BISTs, see [9] for details
0xFF10	BLE_BIST_RES	0x0		Bits 4 to 0: BISTs result (1 => failed, 0 => passed) Bits 9 to 5: BIST done flag (see [9] for details)
0xFF11	BLE_CORE_STATUS	0x0		Bits 1 to 0: Core FSM State: 00: IDLE; 01: PRE-SEARCH; 10: FLUSH; 11: MESRECEIVE[1] Bit 2 is power calculation valid flag
0xFF12	BLE_MAX_PHASE	0x36	sQ0.7	Maximal phase shift value.
0xFF13	BLE_PREAMBLE_LEN	0x28	uQ7.0	Length of synch. word
0xFF14	BLE_PREAMBLE_COEF	0x03	sQ0.7	$\frac{1}{length(preamble)}$ for frequency offset calculation
0xFF15	BLE_PREAMBLE_ID_MEAN	0x0D	sQ0.7	Mean value of synchronization word in NRZ
0xFF16	BLE_FREQ_OFFSET_FIX	0x0	sQ3.7	Frequency offset compensatory value, read only
0xFF17	BLE_FIRST_LLR_NR	0x38	uQ15.0	Number of LLRs to generate *FirstLLRsRdy_SO*
0xFF18	BLE_FIR_SAMPLE_CNT	0x0	uQ32.0	Counter of samples entering FIR, read only

Table A.1.: Description of Celeste's APB registers

[1] MESSAGEREADY state is executed within one clock cycle and is not reported to the register.

Appendix B

Experimental Setup

Figure B.1.: Beacons 5, 6 and 1 installed in the experimental setup

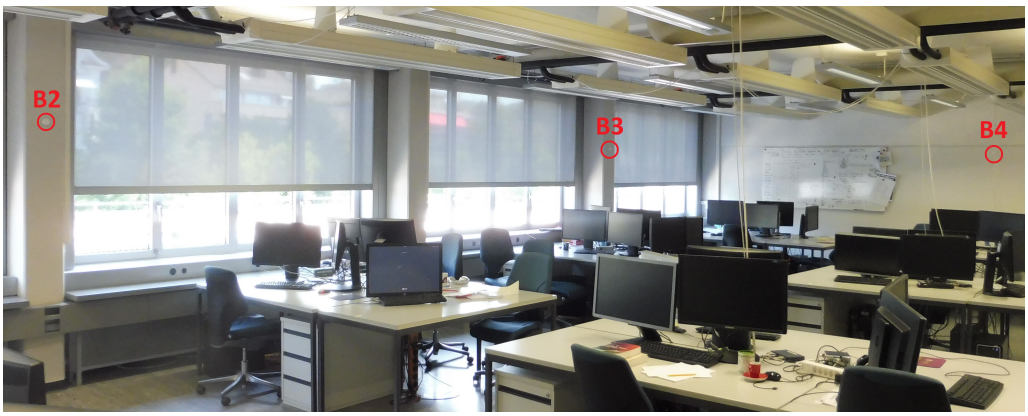


Figure B.2.: Beacons 2, 3 and 4 installed in the experimental setup

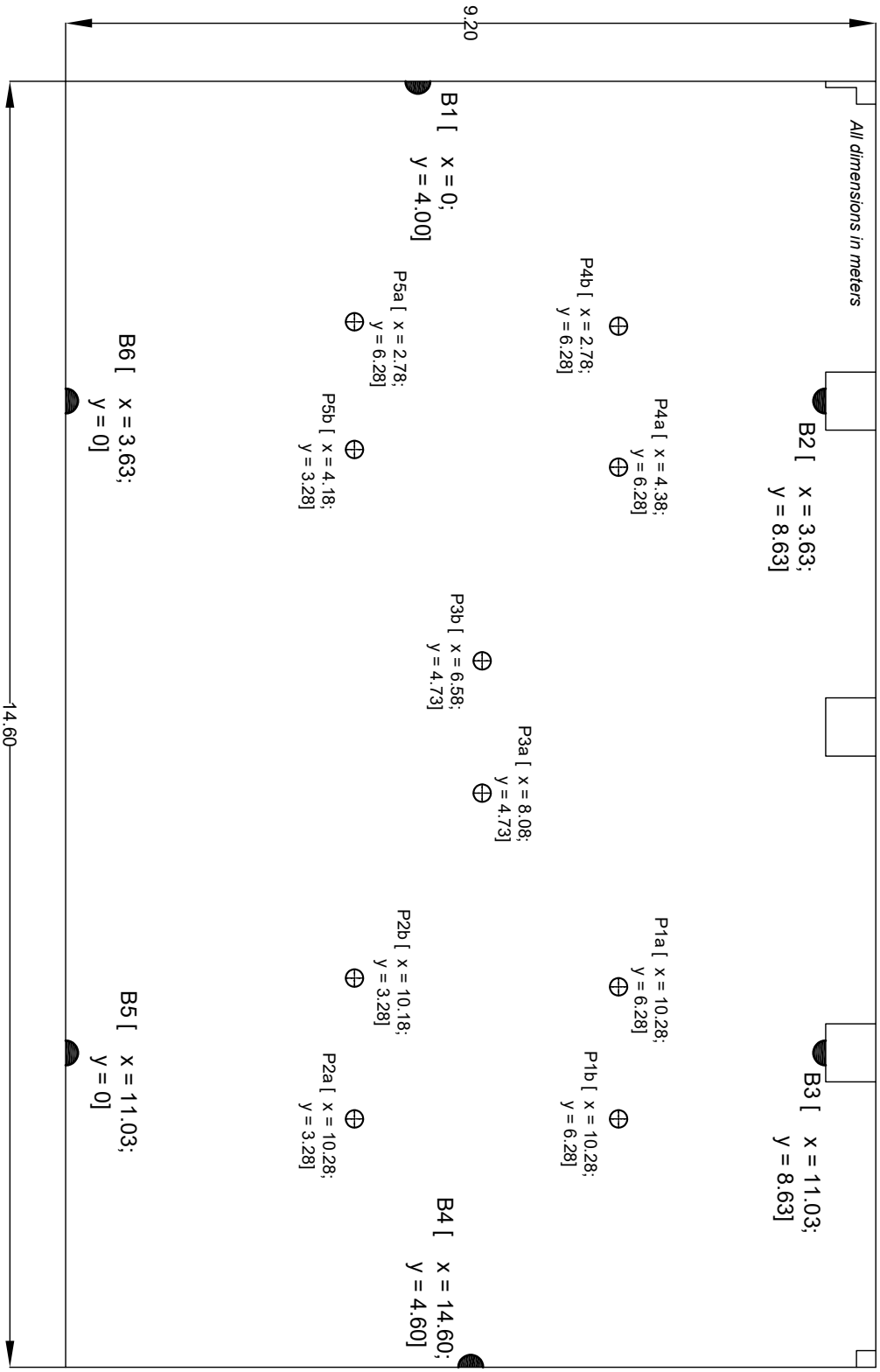


Figure B.3.: Detailed description of the experimental setup

Appendix C

Measurement Results in Control Points Only

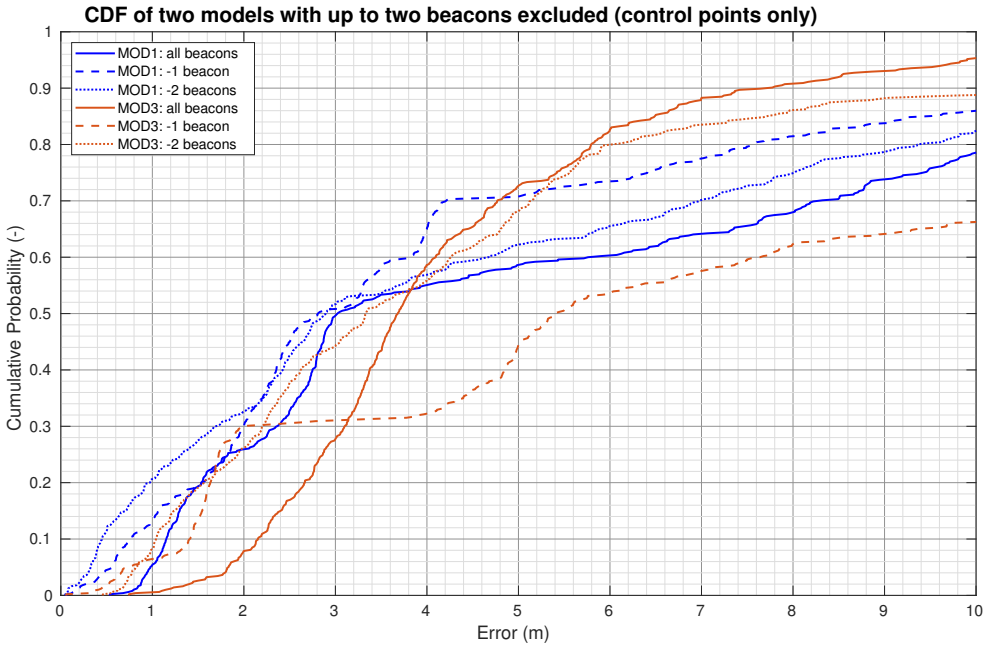


Figure C.1.: CDF of MOD1 and MOD3 with up to two beacons excluded from the position calculation based on the highest SD of RSSI. Measurements in the control points only.

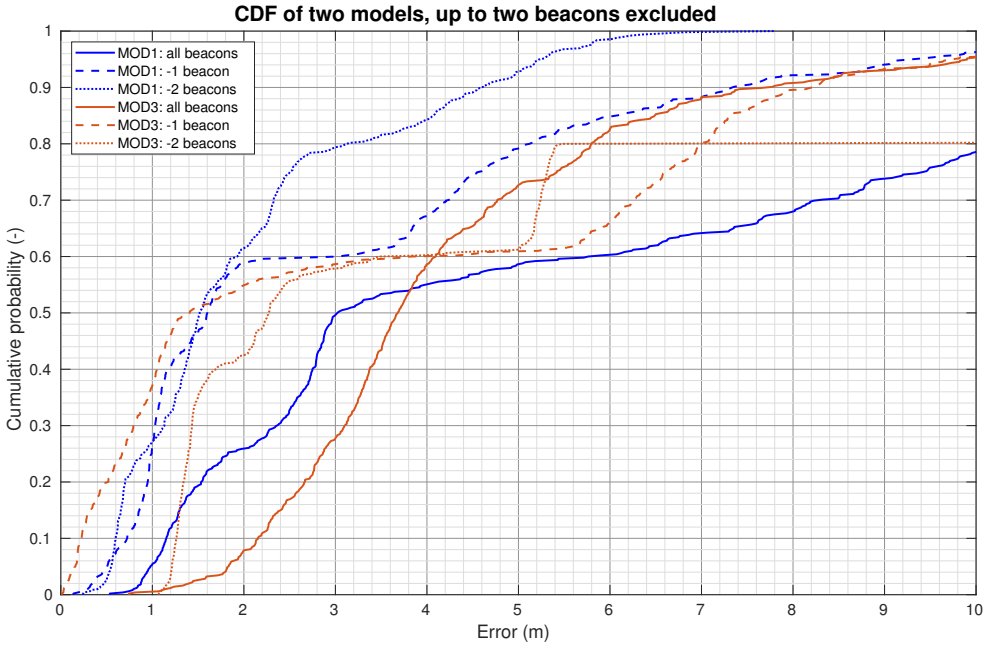


Figure C.2.: CDF of MOD1 and MOD3 with up to two beacons excluded from the position calculation based on the weakest signal. Measurements in the control points only.

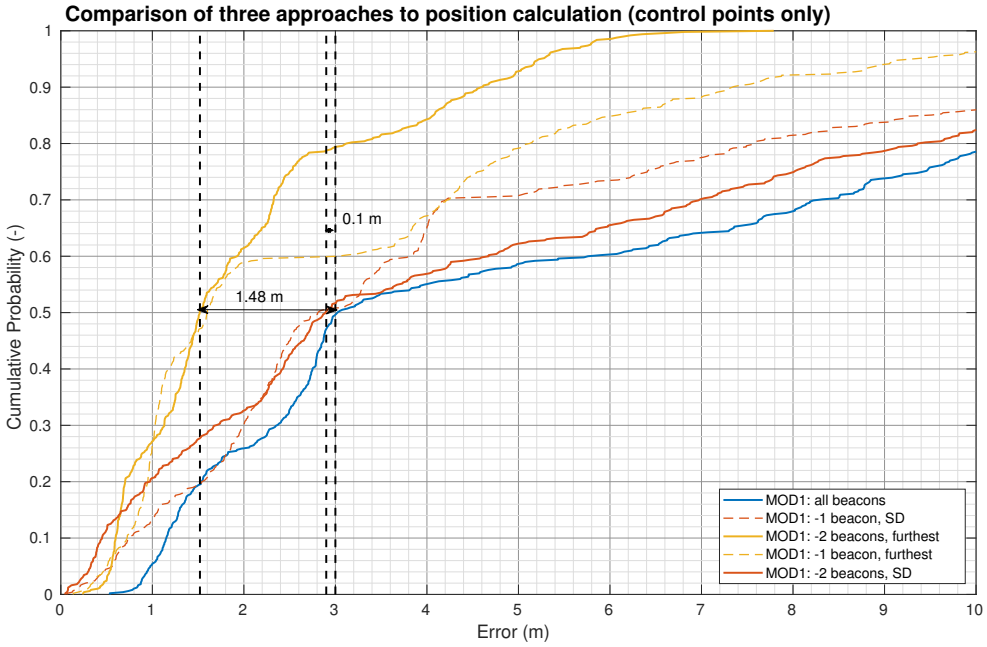


Figure C.3.: Comparison of the three different approaches to the position estimation. Measurements in the control points only.

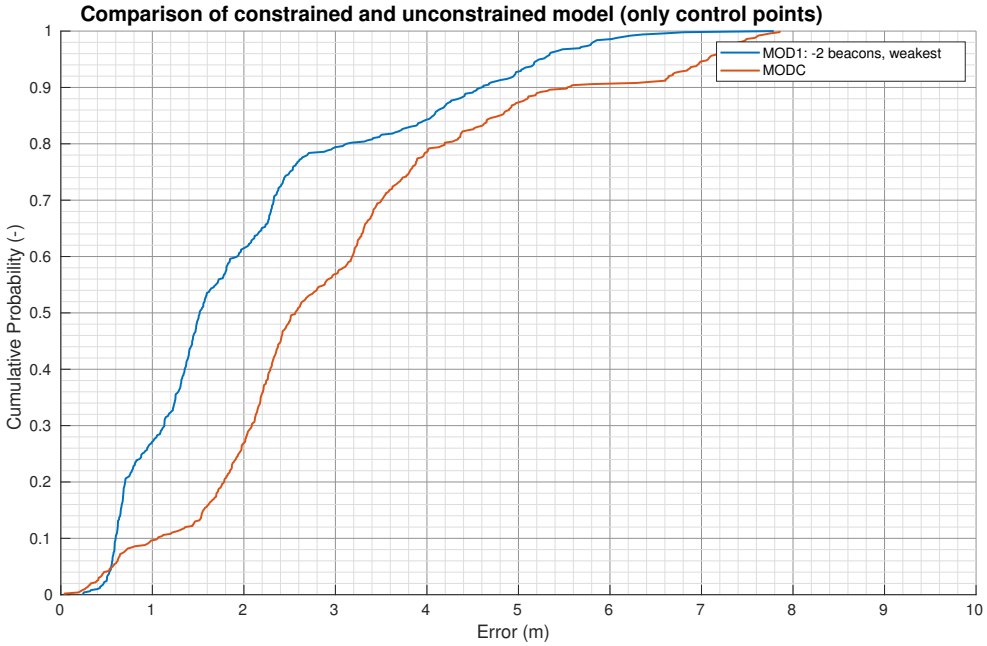


Figure C.4.: Comparison of constrained and unconstrained model. Measurements in the control points only.

Appendix D

Task Description

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Integrierte Systeme
Integrated Systems Laboratory

Master Thesis at the Department of
Information Technology and Electrical Engineering

Spring Semester 2020

Jan Kreisinger

Bluetooth Low Energy Positioning on FPGA

24. 02. 2020

Professor: Qiuting Huang, ETZ J95, Tel. +41 44 632 5240, huang@iis.ee.ethz.ch
Advisor 1: Mauro Salomon, ETZ J90, Tel. +41 44 632 9343, msalomon@iis.ee.ethz.ch
Advisor 2: Stefan Lippuner, ETZ J90, Tel. +41 44 632 6089, lstefan@iis.ee.ethz.ch

Start: 10.02.2020
End: 10.08.2020

1 Introduction

The Internet of Things (IoT) is expected to become one of the key drivers for the semi-conductor industry in the upcoming years. Thereby, a major part of the IoT will consist of moving devices such as wearables or autonomous vehicles. For these kinds of applications the knowledge of the current location is essential.

Solely relying on Global Navigation Satellite Systems (GNSS) such as GPS is insufficient for many use cases, since these signals cannot be received indoors. Even though there is no well established technology yet, Bluetooth Low Energy (BLE) [1] beacons are gaining interest as low cost solutions for indoor positioning.

2 Project Description

Wireless receivers can typically be decomposed into three subsystems, as depicted in Fig. 1:

- *RF Subsystem:* Downconversion from Radio Frequency (RF), analog to digital conversion.
- *DBB Processing:* Physical Layer (PHY) processing (time and frequency synchronization, data demodulation and decoding).
- *CPU Subsystem:* Higher layers of communication protocol, application software.

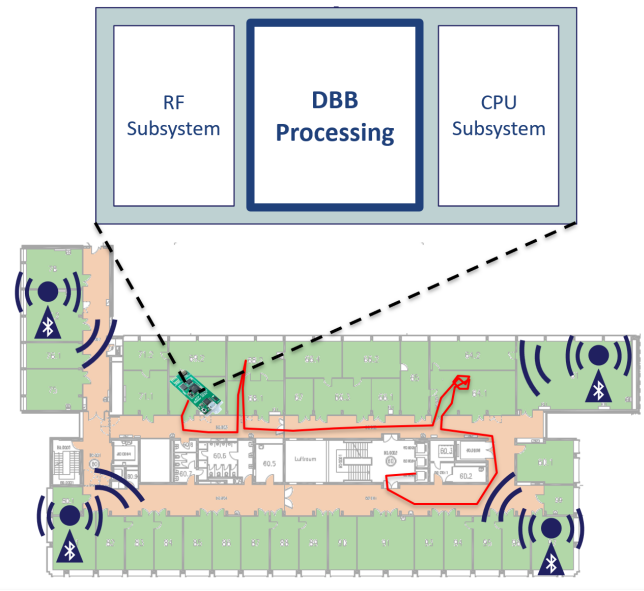


Figure 1: Bluetooth Receiver for Indoor Positioning

The goal of this project is to implement an FPGA based BLE [2] positioning testbed, including the embedded software to perform position fixes with the developed system. The CPU subsystem and the Digital Baseband (DBB) processing will be maped on a Xilinx UltraScale+ FPGA while an EvaLTE [3] board will be used as RF subsystem. Celeste [4], a BLE DBB receiver developed in a previous work performed at the institute, will be used as starting point for this project. Celeste was designed for the UMC65 technology and supports BLE packet detection and demodulation as well as received signal power measurements. The Celeste core will be ported to the FPGA and integrated into an existing processor system as an APB [5] peripheral.

In a second step the firmware to control the Celeste core and the RF IC will be developed on the FPGA testbed. The detection of BLE broadcasting channels will be implemented and verified first. Then, the functionality of the received signal power measurement will be investigated and used to develop a scheme to estimate the location of the device in the presence of multiple BLE beacons with known locations.

Throughout the project different approaches will be used for testing depending on the desired debugging capabilities:

- Register Transfer Level (RTL) simulations of the Celeste core alone or of the entire digital subsystem,
- reception of signals generated with an arbitrary wave generator (ARB) using a Matlab generated baseband signal,
- reception of standard compliant BLE signals generated wiht Bluetooth test equipment,
- over the air detection and demodulation of true BLE beacon signals.

3 Milestones

In order to achieve the project goals stated above the project has the following milestones:

1. *Theory and Previous Work:* Get familiar with the BLE PHY specification [1, 2] and identify the signal processing blocks required to detect and decode broadcasted packets. In particular understand Gaussian Frequency Shift Keying (GFSK) modulation and demodulation. Understand the architecture and interface of the Celeste core [4].
2. *Matlab Model:* Get familiar with the Matlab simulation environment. Learn how to generate Bit Error Rate (BER) plots. Understand the implemented algorithms for preamble detection and GFSK demodulation. Learn how to generate test vectors for the Celeste core. Use the existing Matlab frame work to genearate a reference waveform to test the complete system with an ARB.
3. *System Integration:* Identify all the building blocks that will be integrated on the FPGA and understand how they will interface. Integrate the Celeste core into an existing system that includes a CPU and an RF controler block. The CPU system with the RF controller block is based on previous work and will be provided by the assistants. Realize all required changes to the Celeste core.

4. *FPGA Synthesis:* Adapt the Celeste core for FPGA synthesis. Synthesise the entire system and map it on an FPGA.

5. *Software Development:* Develop the required software to interact with the Celeste core. With the provided examples understand how to control the RF subsystem. Develop a program that detects and decodes BLE broadcasted packets. Implement Received Signal Strength Indicator (RSSI) measurements using the signal power measurement capabilities of the Celeste core.

6. *Positioning and Measurements:* Test the implemented system and evaluate its performance with RF measurements. Use the developed RSSI measurement to implement BLE based positioning. Build a test set-up to measure the positioning performance. Prepare a demonstration for the final presentation.

7. *Documentation:* Document your code (Matlab, VHDL, C, ...) with comments. Prepare and hold a presentation, and write a report.

4 General Recommendations

The following are some recommendations for this project:

- Regularly use version control to save the progress you have made. With git it is good practice to create a new commit at least once an hour. Have a look at [6] for a quick introduction to git.
- Draw block diagrams of your architecture (e.g., by using Inkscape). It is important to note that VHDL coding is greatly simplified by having up-to-date block diagrams at hand.
- While coding VHDL, use the IIS standard coding style as documented in [7, 8, 9]
- Use the *DZ Wiki* located at [10] for information about the design flow. There are detailed answers to many of the questions that might arise during the project.
- If you are looking for references or other papers use IEEE Xplore at [11] and the ETH library at [12].

5 Project Realization

5.1 Meetings

Weekly meetings will be held between the students and the assistants. The exact weekly meeting time and location will be determined to fit the schedule of the assistants. These meetings will be used to evaluate the status and progress of the project. If you like to discuss details of your work, please provide appropriate and up-to-date figures and block diagrams.

5.2 Report

Documentation is an important and often overlooked aspect of engineering. The report of the work should be written in English. Any form of word processing software is allowed for writing the reports, nevertheless the use of LaTeX with Inkscape is strongly encouraged by the IIS staff. The report has to be presented at the end of the project.

The final report must also contain (in the appendix)

- Presentation Slides: The final slides of your presentation.
- Task Description: This document.
- List of the directory structure and the most important files of this project (unless the files are under version control).
- Declaration of Originality: ETH requires a signed declaration of originality [13], which also has to be part of the report.

5.3 Presentation

There will be a presentation (20 min presentation and 5 min Q&A) at the end of this project to present your results to a wider audience. The exact date has to be determined.

6 Project Plan

A tentative project plan is depicted in Fig. 2

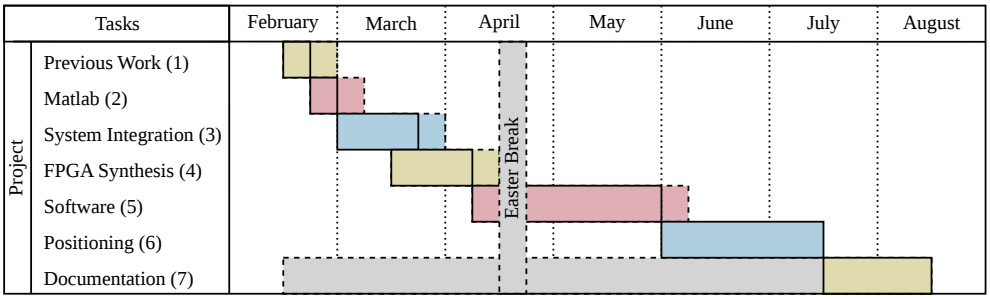


Figure 2: Tentative project plan. The numbers in parentheses refer to the milestones.

References

[1] N. Gupta. *Inside Bluetooth Low Energy*. Artech House Remote Sensing Library. Artech House, 2013.

[2] Bluetooth SIG. Core Specification Version 5.2. `https://www.bluetooth.com/specifications/bluetooth-core-specification`, January 2019.

[3] evaLTE A 2G 3G 4G Cellular Transceiver FMC. `http://iis-projects.ee.ethz.ch/index.php/EvaLTE`, January 2016.

[4] Lorenz Becker-Sander and Robin Gadola. CELESTE: A Bluetooth Receiver for Indoor Positioning, July 2019.

[5] ARM. AMBA 3 APB v1.0 Protocol Specification, August 2004.

[6] Roger Dudler. git - the simple guide. `http://rogerdudler.github.io/git-guide/`.

[7] DZ. Microelectronics Design Center. `http://www.dz.ee.ethz.ch`.

[8] Hubert Käslin. *Digital Integrated Circuits*. Cambridge University Press, 2008.

[9] DZ. Naming Conventions. `http://eda.ee.ethz.ch/index.php/Naming_Conventions`.

[10] DZ. Microelectronics Design Center Wiki. `http://eda.ee.ethz.ch`.

[11] IEEE. IEEE Xplore Digital Library. `http://ieeexplore.ieee.org`.

[12] ETH. ETH-Bibliothek. `http://www.library.ethz.ch`.

[13] ETH. Declaration of Originality. `https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/declaration-originality.pdf`.

Zurich, 24. 02. 2020 Prof. Dr. Qiuting Huang

Appendix E

Declaration of Originality

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Bluetooth Low Energy Positioning on FPGA

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
KREISINGER	JAN

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, Switzerland, 10. 08. 2020	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Bibliography

[1] IndustryARC, "Indoor Positioning and Navigation Market - Forecast (2020-2025)," accessed June 30, 2020. [Online]. Available: https://www.industryarc.com/Report/43/global-indoor-positioning-navigation-market.html

[2] R. Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice, and Advances*. IEEE Press, 2019.

[3] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. D. Hamalainen, "Experiments on local positioning with Bluetooth," in *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*, 2003, pp. 297–303.

[4] G. M. Mendoza-Silva, J. Torres-Sospedra, and J. Huerta, "A Meta-Review of Indoor Positioning Systems," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[5] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[6] M. Ji, J. Kim, J. Jeon, and Y. Cho, "Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, 2015, pp. 92–95.

[7] B. Huang, J. Liu, W. Sun, and F. Yang, "A Robust Indoor Positioning Method based on Bluetooth Low Energy with Separate Channel Information," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[8] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, "Indoor Positioning Algorithm Based on the Improved RSSI Distance Model," *Sensors*, vol. 18, p. 2820, 08 2018.

[9] L. Becker-Sander and R. Gadola, *CELESTE: A Bluetooth Receiver for Indoor Positioning*, Jul. 2019.

[10] B. Weber, H. Kröll, S. Altorfer, and Q. Huang, "Cellular Baseband Development Platform with an open RF Interface," *2015 Wireless Innovation Forum European Conference on Communications*, vol. 37, no. 6, pp. 26–30, 2015.

[11] Rhode&Schwarz, *R&D CMW Radio Communication Tester: Platform Overview*, Rhode&Schwarz, Sep. 2019.

[12] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.

[13] SIG, *Bluetooth Core Specification v5.2*, Bluetooth Special Interest group, Dec. 2019.

[14] D. Burnett, "All BLE guides are wrong," accessed July 15, 2020. [Online]. Available: https://inst.eecs.berkeley.edu/~ee290c/sp18/lec/Lecture7A.pdf

[15] A. A. Circcuits, "Bluetooth and WLAN frequencies," accessed July 15, 2020. [Online]. Available: https://www.allaboutcircuits.com/uploads/articles/Bluetooth_and_WLAN_frequencies.jpg

[16] A. Blackstone, "Understanding the different types of BLE Beacons," accessed July 15, 2020. [Online]. Available: https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/

[17] U. of Oslo, "CORDIC Algorithm COordinate Rotation DIgital Computer," accessed July 18, 2020. [Online]. Available: https://www.uio.no/studier/emner/matnat/ifi/INF5430/v12/undervisningsmateriale/dirk/Lecture_cordic.pdf

[18] J. Brütch and J. Stoffel, *Bluetooth 5 Receiver: From Radio-Frequency to Bit Log-Likelihood Ratios*, Jun. 2018.

[19] M. Korb, S. Willi, B. Weber, H. Kröll, A. Traber, S. Altorfer, D. Tschopp, J. Rogin, E. Dornbierer, M. Salomon, S. Lippuner, L. Wu, and Q. Huang, "A Dual-Mode NB-IoT and EC-GSM RF-SoC Achieving -128-dBm Extended-Coverage and Supporting OTDOA and A-GPS Positioning," in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, 2018, pp. 286–289.

[20] IIS and EEES, "PULP Platform - Project info," accessed July 18, 2020. [Online]. Available: https://pulp-platform.org/projectinfo.html

[21] M. Gautschi, P. D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gürkaynak, and L. Benini, "Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2700–2713, 2017.

[22] Xilinx, *ZCU104 Evaluation Board - User Guide*, Xilinx, Oct. 2018.

[23] T. Geiselbrecht, "Little Kernel," accessed August 4, 2020. [Online]. Available: https://github.com/littlekernel/lk/wiki/Introduction

[24] M. Dreher and S. Cortesi, *Indoor Localization with BLE 5.0*, Jun. 2019.

[25] Minew, "E5 Location Beacon," accessed July 18, 2020. [Online]. Available: https://www.minew.com/products/e5-location-beacon.html