NTNU – Trondheim
Norwegian University of
Science and Technology

Tracking the presence of users by the presence of their device

Pavel Arteev

Master of Telematics - Communication Networks and Networked Services (2
Submission date: June 2014
Supervisor: Frank Alexander Krämer, ITEM
Co-supervisor: Joakim Klemets, ITEM

Norwegian University of Science and Technology
Department of Telematics

Title: Tracking the presence of user by the presence of their device
Student: Pavel Arteev

Problem description:

This work describes a system for tracking people's presence by their personal devices. Tracking is based on detecting the zones in which users are currently present. A zone can be a room in the building or just a specific place in a big area. Such a system can be for example installed at the university and help students to find in which auditorium professors are located at that moment. In addition, this system can be helpful at hospital for a fast detection of rooms in which nurses are present.

Bluetooth Low Energy technology was selected as one means for the device detection. This technology as a part of the Bluetooth standard was built especially to reduce power-consumption of a radio module. This allows devices to advertise their signal continuously over a long period, without being switched off. Advertising is a special phase of the Bluetooth Low Energy standard where devices periodically send small packets to make their presence known. Advertising packet contains UUID, which can be used as user's identification.

Raspberry Pis combined in a network should be used in order to get qualitative information about user's presence in certain zones. Each network node contains a Bluetooth adapter, which is used for scanning Bluetooth devices.

A purpose of this work is to examine how the network of Raspberry Pis can be used to track the location and handle Bluetooth advertising signals. Criteria such as accuracy, speed and scalability of the system should be given as a result of this work, and the system should be implemented and demonstrated.

Responsible professor: Frank Alexander Kraemer, ITEM
Supervisor: Joakim Klemets, ITEM

Abstract

Real-time locating systems (RTLS) are systems, which can automatically identify and track the objects. Such a system can be helpful at university in a way that students can find in which auditorium a professor is present at that moment or at hospital for the fast detection of rooms in which nurses are present.

In this thesis the prototype of the Real-time locating system based on the Bluetooth Low Energy technology was build and demonstrated. System was developed using Reactive Blocks SDK. The development, using this SDK, is turning the ordinary program codding process to the visual process. Raspberry Pi was chosen as hardware solution for the system. Work shows solutions on problems, which can occur during working with Bluetooth.

The results of experiments, which were performed to establish the quality of measure of RSSI value, were demonstrated using the developed system. The results include different combinations of independent variables such as: speed of movement of an object, a type of an object, a scanning interval, a type of network connection and a type of environment (with wall or without wall between the advertising signal and the zone scanner). Further, these results were used for a development of an algorithm for a zone detection. The results of the algorithm were demonstrated in additional experiments, which showed the possibility of the zone detection by the developed system.

A discussion about the correctness of technic approaches, which were used in the work, was presented in a conclusion. In addition, a future work that includes new ideas and suggestions for increasing and improving a quality of the zone detection was presented.

Preface

The work on this Thesis has been performed at the Department of Telematics at the Norwegian University of Science and Technology during the spring semester 2014.

The evaluation of the system was assist by Associate Professor, Frank Alexander Kraemer, who provided great advices and fundamental theory for the system development. My supervisor, PhD Joakim Klemets, put a great effort into the structure of experiments and the thesis as whole. Further thanks goes to the Nordic Semiconductor team, which gave me useful advices and tips for using the BLE technology in the project.

Last, but certainly not least; Thanks to Paula, who has supported me during the work and was correcting my 'language problems'.

Contents

List of Figures **vii**

List of Tables **ix**

List of Acronyms **xi**

1 Introduction **1**
1.1 Motivation 1
1.2 Research goals 2
1.3 Scope 2
1.4 Structure of Thesis 2

2 Background **3**
2.1 Real-time locating systems 3
2.2 Bluetooth Low Energy 4
2.3 Reactive Blocks SDK 5
2.4 Related work 7

3 System **9**
3.1 Messaging protocol 10
3.2 Zone Scanner 12
3.2.1 Bluetooth module 12
3.2.2 Zone scanner system architecture 14
3.3 Central Server 17
3.4 Advertising device 19
3.5 Conclusion 21

4 System experiment **23**
4.1 Experiment 1 24
4.2 Experiment 2 29
4.3 Experiment 3 31
4.4 Experiment 4 34
4.5 Experiment 5 36

4.6 Experiment 6 38
4.7 Experiment 7 42
4.8 Experiment 8 43
4.9 Experiment 9 46
4.10 Experiment 10 49
4.11 Experiment 11 52
4.12 Experiment's conclusion 54

5 Zone detection algorithm **55**
5.1 Sorting data module 55
5.2 Preparing data module 56
5.3 Processing data module 58
5.4 Conclusion 60

6 Experiments of the algorithm **61**
6.1 Experiment 1 62
6.2 Experiment 2 63
6.3 Experiment 3 65
6.4 Experiment 4 66
6.5 Conclusion 67

7 Discussion **69**

8 Conclusion **71**

9 Future work **73**

References **75**

Appendices

A USB reset **77**

B REST API server **79**

List of Figures

2.1 Reactive Blocks example 6

3.1 The architecture of the location system 10
3.2 Zone scanner architecture RaspberryBluetoothNetwork block 14
3.3 Zone scanner architecture BluetoothAgregator block 15
3.4 Zone scanner architecture BLEFastScanner block 16
3.5 Central server architecture block 19
3.6 Central Nordic evaluation kit 20

4.1 A general plan of the area for the system experiment 24
4.2 A general plan for the system experiments with out wall 25
4.3 Photo of the lab from the system experiment 26
4.4 System experiment 1 result 27
4.5 System experiment 1 normalization result 28
4.6 System experiment 2 result 30
4.7 System experiment 2 normalization result 31
4.8 System experiment 3 result 32
4.9 System experiment 3 normalization result 33
4.10 System experiment 4 result 35
4.11 System experiment 4 normalization result 36
4.12 BluetoothAgregator block interval setting part 37
4.13 System experiment 5 result 38
4.14 System experiment 5 normalization result 39
4.15 System experiment 6 result 40
4.16 System experiment 6 normalization result 41
4.17 System experiment 7 result 43
4.18 System experiment 7 normalization result 44
4.19 System experiment 8 result 45
4.20 System experiment 8 normalization result 46
4.21 System experiment 9 result 47
4.22 System experiment 9 normalization result 48
4.23 A general plan for the system experiments with wall 49

4.24 System experiment 10 result 50
4.25 System experiment 10 normalization result 51
4.26 System experiment 11 result 52
4.27 System experiment 11 normalization result 53

5.1 The general approach of the algorithm 56
5.2 Example of Savitzky-Golay filter 58
5.3 Algorithm block scheme 59
5.4 Example of working Neutral zone 60

6.1 Area for experiments of the location detection algorithm 62
6.2 Path of the movement for the experiments of algorithm 64

List of Tables

2.1 Technical comparison of classic Bluetooth and BLE 4

3.1 Location system messaging protocol format 11
3.2 LE advertising report 13
3.3 REST API operations 18
3.4 Bluelocation protocol structure 20

4.1 The configuration overview for the system experiments 25

5.1 Bluelocation protocol structure 56

6.1 Configuration of experiments for location detection algorithm 62
6.2 Results of experiment 1 of the algorithm 63
6.3 Results of experiment 2 of the algorithm 65
6.4 Results of experiment 3 of the algorithm 66
6.5 Results of experiment 4 of the algorithm 67

List of Acronyms

API Application programming interface.

AWS Amazon Web Services.

BLE Bluetooth low energy.

BR/EDR Basic Rate / Enhanced Data Rate.

BSD Berkeley Software Distribution.

ESM External State Machine.

GAP Generic Access Profile.

GPIO General-purpose input/output.

GPS Global Positioning System.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

I2C Inter-Integrated Circuit.

JAR Java Archive.

JNI Java Native Interface.

JSON JavaScript Object Notation.

LE Low Energy.

MAC Media access control.

MQTT MQ Telemetry Transport.

NTNU Norwegian University of Science and Technology.

RFID Radio-frequency identification.

RSSI received signal strength indicator.

RTLS Real-time locating system.

SDK software development kit.

SPI Serial Peripheral Interface.

TCP Transmission Control Protocol.

UART Universal asynchronous receiver/transmitter.

URL Uniform resource locator.

USB Universal Serial Bus.

UUID Universally unique identifier.

Chapter 1
Introduction

This work is showing the Real-Time Locating System (RTLS) solution based on the Bluetooth Low Energy (BLE) technology. Such a system can be helpful at university in a way that students can find in which auditorium a professor is present at that moment or at hospital for the fast detection of rooms in which nurses are present.

The system is developed by using Reactive Blocks Software Development Kit (SDK), which turn the ordinary program codding process to the visual process. BlueZ library was used in order to communicate with the Bluetooth module on a Linux machine. However, during the developing process there were some problems with using Bluetooth, such as Bluetooth module became frozen during scanning or a problem of getting the Received Signal Strength Indicator (RSSI) value when using ordinary BlueZ library utilities.

Experiments with the developed system were carried out in a laboratory of the Medical Faculty at Norwegian University of Science and Technology (NTNU). Experiment results showed the possibility of detection the object location.

1.1 Motivation

In my previous work [Art13], I tried to use a BLE technology to establish an appearance of an object in a specific zone. The results of the work showed that it is possible to use BLE technology for establishing an approximate position of the objects by measuring the RSSI value. However, this work did not provide any experiments, which could show the quality of the object detection. Also, only one type of a Bluetooth device was used, which is actually not answering the question whether it is possible to detect any devices, or just devices which are specially developed for the system. This new work continues and extends the research of using BLE technology for the locating system. A RTLS was built for this project. RTLS is using the BLE technology for detecting the location of an object. This system is evaluated through experiments, which show the quality of detection of the object location using the BLE technology.

1.2 Research goals

For the successful result of the work, following goals were established:

- Develop a software architecture design of a Real-time locating system, which is using the BLE technology for establishing location of the objects;
- Provide empirical evidence by conducting experiments, which will help to establish the quality of detection of the object location using the BLE technology and show its limitations.

1.3 Scope

The are many interesting technologies which can be used for the RTLS, for example ultrasound or Radio-frequency identification (RFID). This thesis is only concentrating on using BLE technology for the locating system and all results of the work will be according to this technology. The BLE technology was chosen as a new technology, which is compared to the classical Bluetooth technology, cheaper and has less power consumption. Further, the development of the RTLS will be made from a software engineering point of view and not electrical engineering. The success of the result will depend on a quality of a hardware, but this work will use only hardware ready solutions. The hardware design can be considered as a future work.

1.4 Structure of Thesis

First chapter is a current chapter and it includes introduction to the work. Chapter 2 contains the theoretical background of the locating system, Bluetooth Low Energy technology and Reactive Blocks SDK as a SDK for development of the real-time systems. Chapter 3 provides the description of the system. Chapter 4 is an experiment chapter. The system developed in Chapter 3 was used to carry out the experiments. The algorithm for the zone detection is described in Chapter 5 and is based on the results from the experimental Chapter 4. Chapter 6 is an experiment chapter of the algorithm developed in the Chapter 5. Discussion is presented in the Chapter 7. The overall conclusion is presented in Chapter 8 and the prospects for the further work is described in the Chapter 9.

Chapter 2
Background

This chapter is a description of the background, which was used during the project. It describes Real-time locating systems (RTLS), Bluetooth Low Energy technology, Reactive Blocks SDK and previous work that are related to my project.

2.1 Real-time locating systems

Real-time locating systems (RTLS) are systems which can automatically identify and track the objects [fS08]. RTLS are basically indoor systems, for areas where it is necessary to monitor and track objects such as inside a building, but it can also be outside the buildings.

There are 3 roles in a system:

- RTLS tags - autonomous devices, which are physically connected to the real objects and communicate with the RTLS infrastructure in order to provide identification and tracking of the object;
- RTLS infrastructure – infrastructure, which includes the stationary locating stations with a defined position. Stationary stations are communicating with the RTLS tags, and with this it is possible to determine the position of the object;
- Server software – software, which is collecting all information about the RTLS tags from all stationary stations, making identifications of the objects and calculating objects positions.

In the RTLS, objects with the RTLS tags are detected by single or multiple locating stations of the RTLS infrastructure and then the server software identifies the object and calculates the position.

As the main characteristics of the RTLS we can refer these:

- a period of detection - how frequently system is updating information about the object;

– a precision of detection - the precision shows a range or a distance in which object will be detected with the same position;
– a quality of detection - percentage of true detection with the specific precision of a detection.

The most popular technology used for RTLS are:

– Global Positioning System (GPS);
– Radio technology;
– Optical technology (example: infrared);
– Audio technology (example: ultrasound).

In this project, the BLE technology, which is a part of the Bluetooth standard, was used.

2.2 Bluetooth Low Energy

Bluetooth is a radio communication link, operating on the unlicensed ISM band at 2.4 GHz frequency [PMoBS13b]. It allows to transfer data between the Bluetooth devices. Bluetooth has two forms of wireless technology. They are the Basic Rate / Enhanced Data Rate (BR/EDR) and Low Energy (LE) [PMoBS13a]. BR/EDR technology is sometimes called classic Bluetooth. Table 2.1 compares the technical details of calssic Bluetooth and BLE. It can be seen that BLE has less power consumption and low latency compared to the classic Bluetooth, which is making it more profitable for use in the RTLS.

Table 2.1: Technical comparison of classic Bluetooth and BLE

Technical Specification	Classic Bluetooth	Bluetooth low energy
Distance/Range	100m	50m
Data rate	1-3 Mbit/s	1 Mbit/s
Application Throughput	0.7-2.1 Mbit's	0.27 Mbit/s
Active slaves	7	Not defined
Security	56/128-bit	128-bit AES
Latency (from a non connected state)	100 ms	6 ms
Time to send data	100 ms	3 ms
Voice	Yes	No
Power consumption	1	0.01 to 0.5
Peak current consumption	< 30 mA	< 20 mA

BLE is a wireless technology design, which became a part of Bluetooth v4 specification from 2010. Originally, Nokia developed it in 2006. This technology operates on the same 2.4 Ghz frequency as a BR/EDR Bluetooth, which allows the Bluetooth device to share a single radio antenna. The Bluetooth devices, which implement both LE and BR/EDR, are called dual-mode devices. Therefore these both technologies are not compatible.

BLE was designed in order to decrease the cost and a power consumption of the modules and aimed basically on home, fitness and healthcare sectors.

Data, which are transferred between the LE devices, are positioning in two types of events. These events are Advertising and Connection events. Advertising events are necessary before establishing the connection, and help the devices to "find" each other. After the connection becomes established between two devices, the advertising event is ended and connection events begin.

The Bluetooth system defines Generic Access Profile (GAP) as a base profile for all Bluetooth devices. GAP profiles include different layers, which are different for BR/EDR and LE technologies [PMoBS13c].

For LE, GAP defines 2 groups with roles. They are connectable and non-connectable groups. Connectable group includes Broadcaster and Observer roles. Broadcasters are type of devices, which are only transmitting the signal. Broadcasters use advertising to broadcast the data. Observers are type of devices, which are only receiving the broadcast data from the broadcasters, but do not establish the connection. Non-connectable group includes Peripheral and Central roles. Devices in Peripheral role support single connection and are usually connected to the central devices to transfer some information. Devices in Central role support multiple connections with the peripheral devices, they are collecting and computing the data from all, connected to them peripheral devices. As an example for the connectable group we can look on the sport watches with a heart rate and GPS sensors. Watch is a device in a central role, which listens to the data from the GPS and heart-rate sensors. GPS and heart-rate sensors are devices in peripheral roles, that are connected to the watch and transfer the location and heart-rate information.

For the locating system, I used Broadcaster and Observer roles. RTLS tags were Broadcasters and stationary locating stations were Observers. For the Observer, it is possible to measure the RSSI. RSSI shows power, which is present in the received signal. For the BLE devices, the RSSI value is in the range from -127 to 21 and the unit is dBm [PMoBS13f].

2.3 Reactive Blocks SDK

Reactive Blocks is a SDK for developing Java applications [AS14a]. The main idea of this SDK is to develop the application using the visualized blocks. This approach makes it easy to understand, develop and maintain the system. The Reactive Blocks is a combination of diagrams and Java code [AS14b].

The main components of the SDK are:

– External State Machine (ESM) - it is describing the internal behavior of the input and the output pins of the blocks;
– An activity diagram - describing the internal processes of the blocks;
– Java methods - describing the details of the operations inside the blocks.

The example below, presented on the Figure 2.1, shows the example of the simple application made by Reactive Blocks SDK. The application is printing out the phrase 'Hello!' on the console, periodically with interval 1 second.

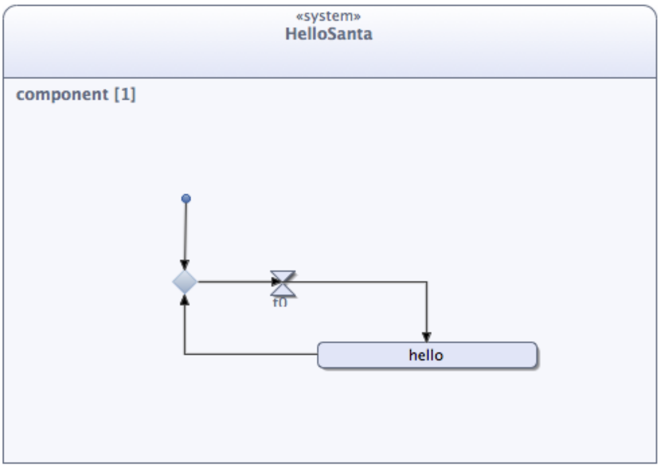

Figure 2.1: Reactive Blocks example

This is an example of a simple application, but it is already showing how easy it is to understand the system developed by this block. The application flow starts with an initial node, then a timer node, which is setting delay for 1 second, and then printing of the phrase 'Hello!' in operation hello. Operation hello is implementing ***System.out.println("Hello!")*** code. After 'Hello' is printed for the first time, the application flow returns to the timer. Finally there is a loop, where the phrase is being printed every second.

During the development of complexity systems, it is possible to reuse some blocks and by this decrease the development time.

The locating system developed, are based on the Reactive Blocks SDK. The study of how to use SDK was easy and understandable, and increased the possibility to start to develop the system fast.

2.4 Related work

Compared to the previous work [Art13], where Bluetooth Low Energy technology was used to establish an appearance of an object in a specific zone, this work shows the Real-time locating system based on the BLE technology, which is able to monitor for the object location using a collaboration of several scanning stations.

A location system using the ultrasound is introduced and described in the master thesis 'Capability Assessment of Indoor Positioning Systems' [Lan09]. The project is showing a possibility to detect objects inside the zone. One disadvantage of using the ultrasound in a location system is that the ultrasound, which is generated by the RTLS tag, has a direction. Consequently, if the receiver is behind the transmitter, it is not possible to detect the object. As a solution, several receivers around the zone or several transmitters, which are directed to different directions, can be used. However, increasing the amount of the devices will increase the cost of the system and also increase the difficulty of maintaining the system. Next disadvantage is a long period of transmission of the signal, which is 5 seconds minimum. Also, since the ultrasound waves cannot pass through the walls, the detection can be made only in the area of one open zone. Therefore, if an object will stand in a blind area[1], system will not detect even the approximate location of the object.

Most of the studies made on the location systems, where Bluetooth is used, are based on the BR/EDR technology described in Section 2.2. For example, in the work 'Bluetooth Tracking without Discoverability' following problems were registered [HH09]:

- High Tracking Latency - each scan takes 10.24s, a system have a slow update;
- Devices in the System must be Discoverable. The ordinary mobile phones were used in the project. Usually, these devices have limited time during which the device can be in a discoverable Bluetooth mode. The limitation of the discoverable mode is caused by a high power consumption of the BR/EDR Bluetooth technology.

In this project we are using BLE, which has less power consumption compared to the classical Bluetooth. Also, in BLE system, the scan interval takes 250ms which is much better and faster compared to 10.24 seconds.

There are very few examples of works which are using Bluetooth Low Energy due to the fact that this technology is new. This is one of the reasons why I decided to orientate my work on a research of BLE technology and further on an experiment, which is using this type of the technology. However, it is important to compare this work with currently 'modern' technologies on the market such as iBeacon technology developed by Apple. iBeacon devices are usually small devices which are advertising

[1] blind area - the area where the object can not be detected by the system.

a signal using the BLE technology. If iBeacon device is close to the user, it is possible to catch the advertising signal by special application. Advertising signal contains UUID of the iBeacon devices, which is possible to detect by phone application, to which the iBeacon device was close to at the current time. This approach can be used for example in the shops for advertising products. iBeacon device can be installed in the area with the products which are necessary to advertise and when user will appear in this area, the phone application will show to the user the product name and a price. The main difference between the system developed in this work and iBeacon, is that iBeacon technology does not track nearby users but it is only advertising the signal to the users phone. The system developed in this work allows to track the objects.

Chapter 3
System

This chapter is a description of a system, which was developed in order to evaluate experiments, which are testing whether it is possible to use BLE as a technology for the indoor location system.

As technical requirements for the system, it needs to:

1. Receive the BLE advertising signal from the BLE peripheral devices, which are in an advertising mode in different zones;
2. Be able to measure RSSI value of the received signal;
3. Collect the information about scanned devices from different zones in one place;
4. Establish the location of the devices using the measurements of RSSI values. As a location result, system needs to identify and appoint a zone in which advertising device was present;
5. Have an interface, which allows communication with other systems in order to provide a location information.

The architecture of the location system is presented on the Figure 3.1.

Advertising devices used for the experiments are Bluetooth Low Energy (BLE) peripheral devices which are configured to be in an advertising mode. Zone scanners are nodes which are installed in different zones of the building and are receiving the advertising signal. Likewise, zone scanners are measuring the RSSI values of received signals and send this information to a central server. A message server is providing a communication bridge between the zone scanners and the central server. The central server is collecting all information about the advertising devices from the zone scanners. The central server is further calculating the location of the advertising devices. Also, it has a communication interface that is providing a possibility to exchange the location information with other systems.

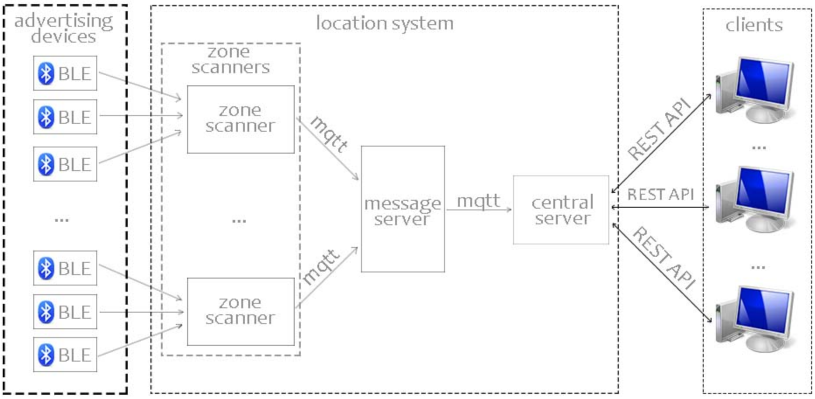

Figure 3.1: The architecture of the location system

3.1 Messaging protocol

A messaging protocol is responsible for data exchange between the zone scanners and the central server. MQ Telemetry Transport (MQTT) protocol was chosen as a basis for the message exchange. The reason why this protocol was chosen is because Reactive Blocks SDK provides a ready solution as a building block for the MQTT connection.

MQTT is machine-to-machine protocol or "internet of things" protocol [8]. It is a lightweight protocol and is allowed to connect the remote devices into the one network group. The protocol consists of a broker and clients. Broker is a central node and it is usually a dedicated server which is located in internet and all MQTT messages are going through this server. Clients are subscribed to the channel topics on the broker. Each channel topic is making an associative group for the clients where they can exchange their information. If different clients subscribe to the same channel, they become a part of one group. Clients can send messages to the channel, and can listen the channel in order to receive messages.

As an example of a location system, I am using channel topic *'bluelocation'* for exchanging information between the zone scanners and the central server. The zone scanners are sending information to the channel and the central server is listening for the channel. By this mechanism, the central server can receive messages from the zone servers.

As a broker, Mosquito server was used in this project. Mosquitto is an open source server with Berkeley Software Distribution (BSD) license based on the MQTT v3.1 protocol [Mos14]. Broker was run on the ubuntu server located on the Amazon

Web Services (AWS). AWS are cloud services, which allow to run the virtual servers in internet [Ama14].

MQTT message basically consists of the channel topic and a payload. Payload is a data, which is usually specific for the system. In a payload, JavaScript Object Notation (JSON) format for the data formatting was chosen to be used. JSON is a light-weight text format, based on a subset of JavaScript programming language [Int13] and consists of two structures:

- A key-value structure;
- Lists of objects, for example arrays.

Table 3.1: Location system messaging protocol format

ZoneInformation	
timestamp	long
zone_id	String
zone_ip	String
device_array	Array of DeviceInformation

DeviceInformation	
time	long
device_id	String
advertising_da	String
rssi	double

Table 3.1 is showing, how JSON format is structured for the location system. A message consists of two types of objects named ZoneInformation and DeviceInformation. ZoneInformation is a description of the zone scanner that consists of:

- timestamp - when message was sent;
- zone id - zone identification. Media Access Control (MAC) address of the network interface was used as an identification;
- zone_ip - ip address of the zone scanner;
- device_array - consists of an array of the devices, which were scanned by the zone scanner.

DeviceInformation is a description of the advertising device, which was detected by the zone scanner. Object consists of next fields:

- time - when devices was detected last time;
- device_id - a specific id for the device such as MAC or Universally Unique Identifier (UUID) if it was included to the advertising data;
- advertising_data - Advertising data field of the advertising protocol;
- rssi - measured RSSI value of the advertising device.

3.2 Zone Scanner

A zone scanner is responsible for the detection of the advertising devices, measuring the RSSI and sending identified information to the central server. A single-board computer Raspberry Pi was chosen as hardware. It was a good solution for the zone scanner in on order to simplify and decrease the time of the development process. Software for the project was written on the Java using the ReactiveBlocks SDK. Wi-Fi or Ethernet can be used for the network connection. Raspbian linux system was used as an operation system.

3.2.1 Bluetooth module

Bluetooth scanner is based on the Bluetooth micro module, which was made as a USB dongle by company Targus and supports the BLE technology [Tar14]. BlueZ is responsible for the communication between the Bluetooth module and the operation system. Bluez is a Bluetooth stack for the Linux, which implemented a specification of Bluetooth for the Linux systems. For this project, it was necessary to implement the scanning of the advertising devices with BLE technology. Utility 'hcitool', which is a part of the BlueZ library, allowed the scanning for the BLE devices and for the normal Bluetooth devices. To start the scanning of the BLE devices it is necessary to use parameter *'lescan'*. The example of the command is presented below:

```
Command:
hcitool lescan
Output:
LE Scan ...
12:88:FF:FF:11:99 touch
12:88:FF:AA:11:99 (unknown)
```

As a result, a MAC address and the name of the detected Bluetooth device are obtained. At this stage of a system development, there were two problems found within the scanning which were making the locating system impossible to implement.

First, during the scanning mode, the Bluetooth module became *'frozen'* after some minutes (1-2 min) and did not give any response back. The module did not respond even after it was programmatically restarted by *hcitool* utility. Different parameters, such as a type of the scan (active or passive) or an interval of the scan have been tried but the only result was that after some interval, the module became frozen again. One solution was to physically disconnect Bluetooth module from the USB port and then connect it back. The Bluetooth module then started to work again for few minutes. After making some research on Internet about this type of problem, the solution was found. The solution included a sending of the *USBDEVFS_RESET* signal to the USB port in order to restart the module. Based on this solution, a small utility, which is making a reset of the USB port, was written. The code example is presented in the appendix page 77.

Second problem was that *hcitool* returned only MAC address of the devices but it did not return either the advertising data nor the RSSI value. Since BlueZ is an Open Source it was possible to rewrite the *hcitool* utility for needs of this project. As a background theory, the Core Specification of the Bluetooth System v4.1 was used. In a specification, it is written that LE Meta Event packet is used to encapsulate the LE Controller specific events [PMoBS13d]. One of these events is LE Advertising Report Event. The Table 3.2 represents the fields of the advertising report.

Table 3.2: LE advertising report

Event	Event Code	Event parameters
LE Advertising Report	0x3E	Subevent_Code, Num_Reports, Event_Type[i], Address_Type[i], Address[i], Length[i], Data[i], RSSI[i]

The LE Advertising Report event indicates that some information was received during the scan mode [PMoBS13e]. A useful field, for this project, from the protocol is Subevent Code, which is supposed to be equal 0x02 in order to have the Advertising Report. Also, field Data is an advertising data, which can be useful since it can include extra information, for example UUID and RSSI. The RSSI value only exist when the RSSI field value is in the range -127<=N<=+20 [PMoBS13f]. The unit for the RSSI value is dBm.

The final utility for the scanning device was written using the C code. As an output, utility returns continuously the information about the scanned devices, such as device's MAC address, advertising data and RSSI value.

Since the entire location system was written using the Java language, it was necessary to make a bridge between the C application and Java application. Java Native Interface (JNI) was used for this purpose. JNI is allowed for the Java code, which is ran on the Java Virtual Machine, to call native applications and libraries written on the other languages, for example C or C++.

3.2.2 Zone scanner system architecture

The zone scanner software description is presented in this chapter. Using the Reactive Blocks SDK it was easy to split the development in different modules. Thus 3 modules, BLEFastScanner, BluetoothAgregator and RaspberryBluetoothNetwork were made.

The architecture of the system block is presented on the Figure 3.2 .

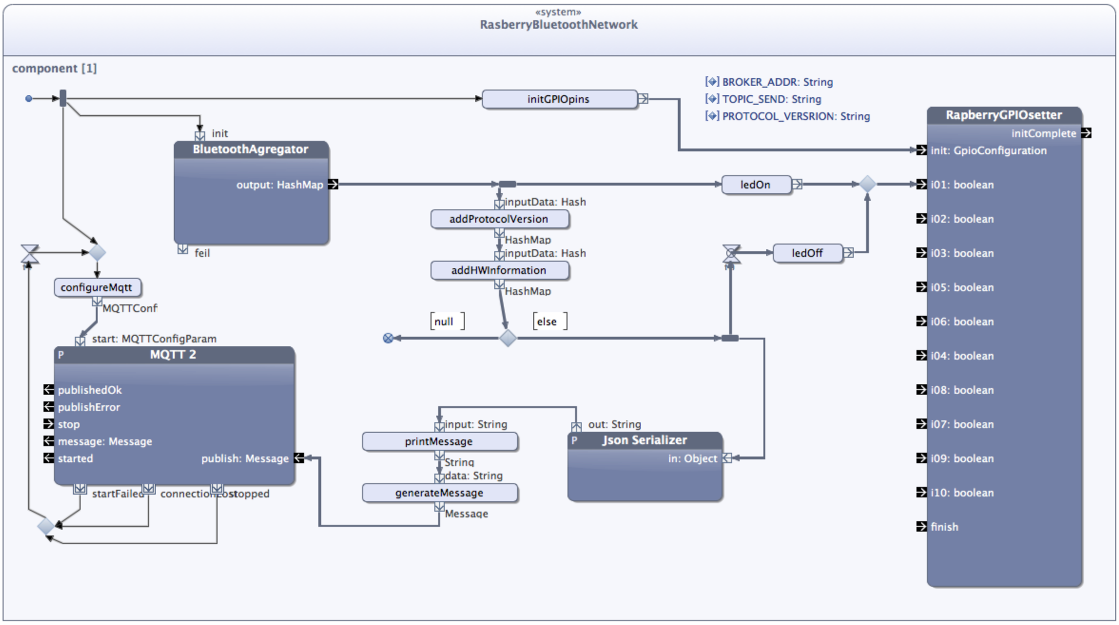

Figure 3.2: Zone scanner architecture RaspberryBluetoothNetwork block

System block is a main block for the application and it consists of local blocks, such as Bluetooth Agregator, MQTT, RaspberryGPIOSetter, Json Serializer. BluetoothAgregator block is responsible for collecting data from the Bluetooth module, RaspberryGPIOSetter serves as communication with the General-Purpose Input/Output (GPIO) port of Raspberry Pi, JsonSerializer is converting Java object to the JSON format data and MQTT block is responsible for sending of information to the MQTT broker. The evaluation flow of detecting the advertising devices by the zone scanner is next. When the zone scanner detects the advertising devices, block BluetoothAgregator is sending out, through pint, output information about the advertising devices. RasbperryGPIOsetter receive this information and trigging the pin 1 of the GPIO port (the LED can be connected to this pin, and will indicate when the zone scanner detected the advertising devices). As next step, Information is coming to the JsonSerializer, which converts Java object to the JSON data. After this, JSON data is coming to the MQTT block. MQTT block is sending out this information to the MQTT broker.

The architecture of the BluetoothAgregator is illustrated on the Figure 3.3.

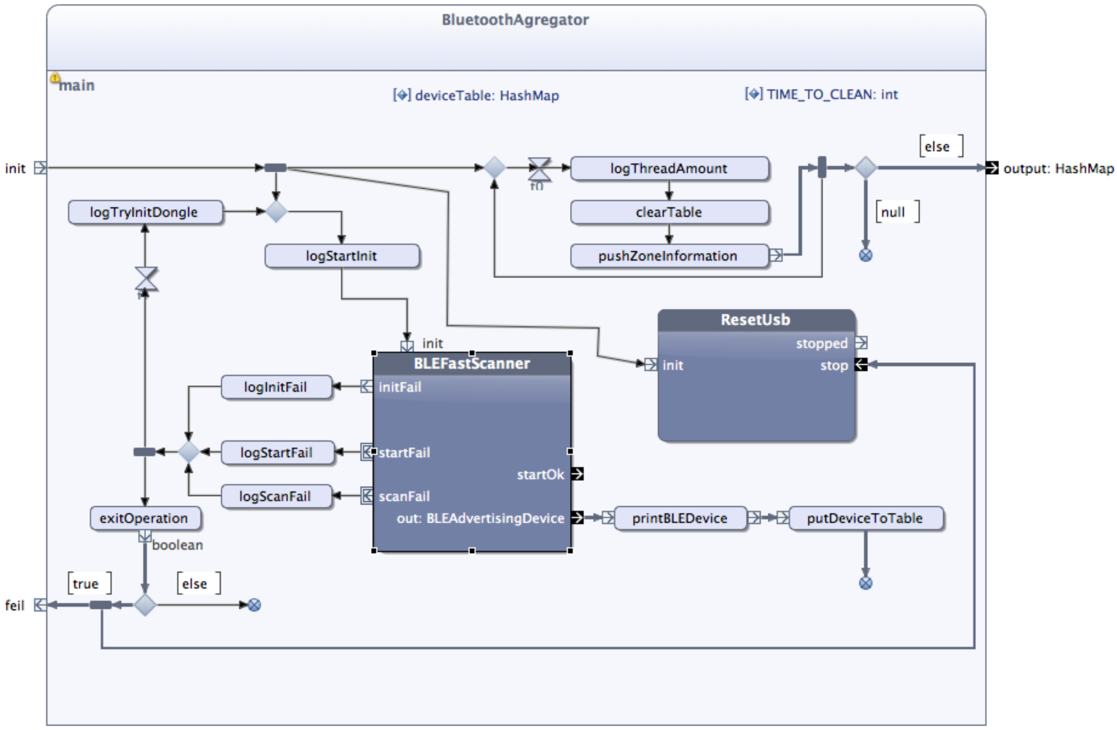

Figure 3.3: Zone scanner architecture BluetoothAgregator block

BluetoothAggregator block is aggregating the information about the advertising device into the table. A reset Universal Serial Bus (USB) module was implemented in this ResetUSB block, which is helping to prevent Bluetooth module from being frozen. During the initialization, the block starts a timer, which periodically pushes out the information about the BLE advertising devices from the aggregated table through a pin output as a HashMap Object. BLEFastScanner block is a connector between the software and the Bluetooth module, which is scanning for the device. BLEFastScanner block architecture is presented on the Figure 3.4.

The block is based on JNI for the Bluetooth scan utility, which was described above. As a flow, the block is starting with Bluetooth module initialization, where

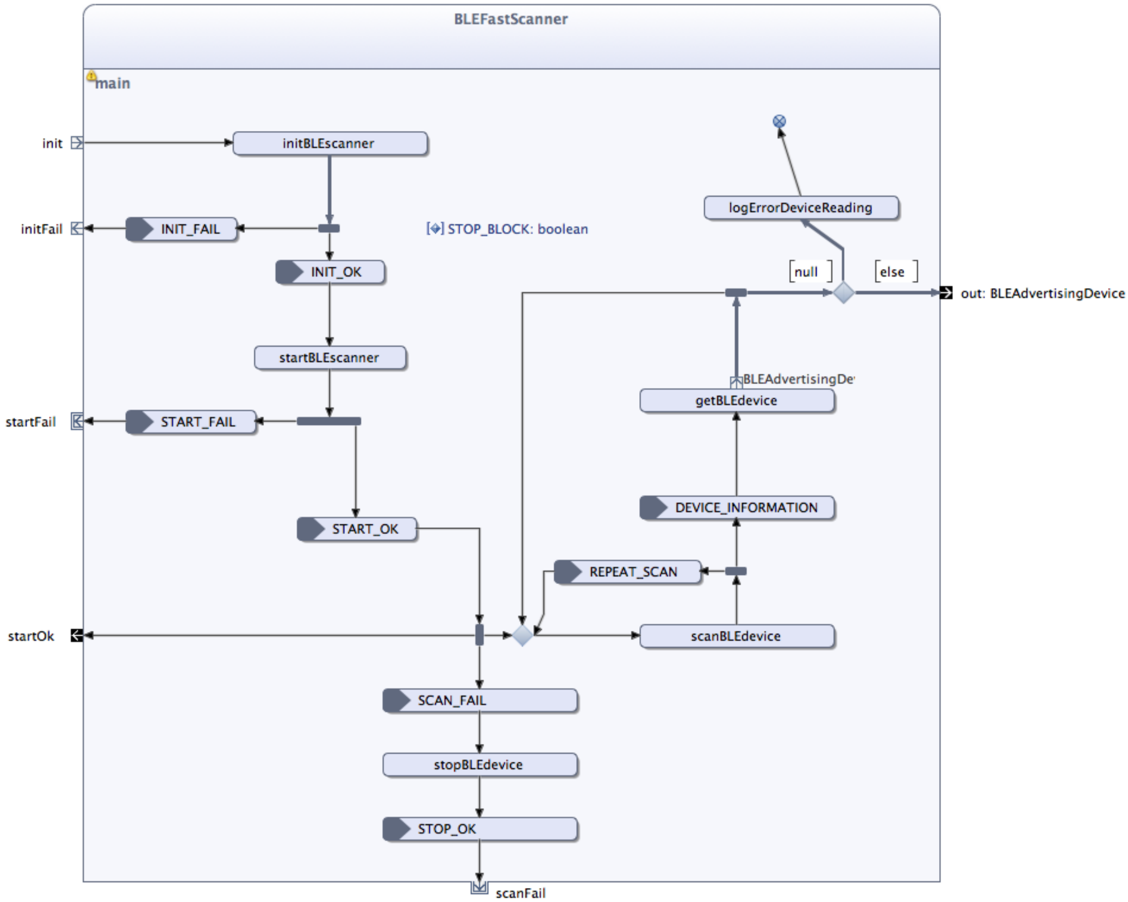

Figure 3.4: Zone scanner architecture BLEFastScanner block

it is setting Bluetooth module to be in scan mode. Next, the scanning process is starting. After the start of the scanner, there is a waiting period for the advertising information, which is coming when Bluetooth module receives the advertising signal from the BLE advertising device. The advertising information obtained is sent out from the block from the pin *'out'*.

The application for Raspberry Pi was packed to one file called Java Archive (JAR) in order to make an easy distribution.

3.3 Central Server

Software for the central server is written using the Reactive Blocks SDK. Since SDK is based on the Java language it is possible to make a cross-platform application. Cross-platform application is a type of application, which can be run on different systems [Wik]. A good example is the Java application that can be run on the Linux, Windows or Mac without or with minor changes in the code.

The role of the central server is to receive the messages with the location information from the zone scanners, store this information and to communicate with other systems. As a communication link with other systems, REST Application Programming Interface (API) was chosen for the use.

REST is a communication design style for distributed hypermedia systems [Fie00]. Clients are making request on the server through Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) protocols. Requests consist of different HTTP methods such as GET, PUT, POST and DELETE. Each of the method is responsible for one type of an operation. A method GET is responsible for returning information from the server, a method POST for the creating new information on the server, a method PUT to change information on the server and a method DELETE to delete the information. Also as part of the request, clients are sending parameters to determine which operation server needs to work. Parameters can be transferred in the body of the HTTP request or as an request link, for example *http://serverexample.com/api/zones*, Here, we are telling the server that we would like to use the api and get an information about the zones. In this case, *'zones'* were a parameter for the method. As a response, server returns the status code, which indicates that request was successful or unsuccessful, and also information data, which is usually JSON format data.

Table 3.3 shows which operations were implemented on the location server. The table consists of the operation names and operation descriptions.

Software architecture for the central server is presented on the Figure 3.5. During the start, the central server is initializing the MQTT block for receiving information from the zone scanners. For the MQTT module initialization, central server is setting the ip address and port of the MQTT broker and subscribing to the specific channel topic. Topic *'bluelocation'* was used for the location system. Two HashMap tables, they are TABLE_DEVICE and TABLE_ZONE, were used as a storage for the location information. HashMap is a hashed table of the Map class [Ora14] and is based on the Key-Value approach. Key-Value approach is able to store and associate the object with the key value. It is also able to get back the object from the HashMap using the key value. For the TABLE_DEVICE, the key value is an advertising device id, for the TABLE_ZONE, the key value is a zone id. It could be possible to use

Table 3.3: REST API operations

Operation	Description
/api/devices/	Operation returns information about the all devices, which was detected by the system, as an array. Device information includes zone where it was detected and RSSI value measured on this zone.
/api/device/<device_id>/	Operation returns information about the specific device, which was detected by the system. <device_id> paramater shows, information of which device need to be returned
/api/deviceZone/<device_id>/	Operation returns the zone_id in which the device is presented now. This operation is based on the algorithm of the zone determination.
/api/zones/	Operation returns information about the all zones which is in the system as an array. Zone information includes zone_id, zone_ip and array of devices which was detected by this zone.
/api/zone/<zone_id>/	Operation returns information for the specific zone. Information includes zone_id, zone_ip and array of devices which was detected by this zone.

arrays instead of HashMaps, but in order to find the object by specific id, it would be necessary to go through all elements of the array and compare the id with the object id. HashMap approach is making searching process much faster. The approach with using 2 tables was chosen in order to decrease the searching time for a specific device or a specific zone and improve the aggregation of the location information. So, for example, if we want to get information about the specific advertising device id, the system needs to look in the table TABLE_DEVICE and for the zone in the table TABLE_ZONE.

Afterward, the MQTT block receives the message from a zone scanner. The Message is coming out from the pin *'message'*. Message data is a sequence of bytes, which represent a JSON string.

After the JSON string deserialization to the Map object, a timestamp is added, which shows, when the message was received. Next step is changing the advertising device address. By default, the address is a MAC address of the BLE advertising device, but if the UUID was encapsulated inside the advertising data, it is possible to use this UUID instead of MAC address.

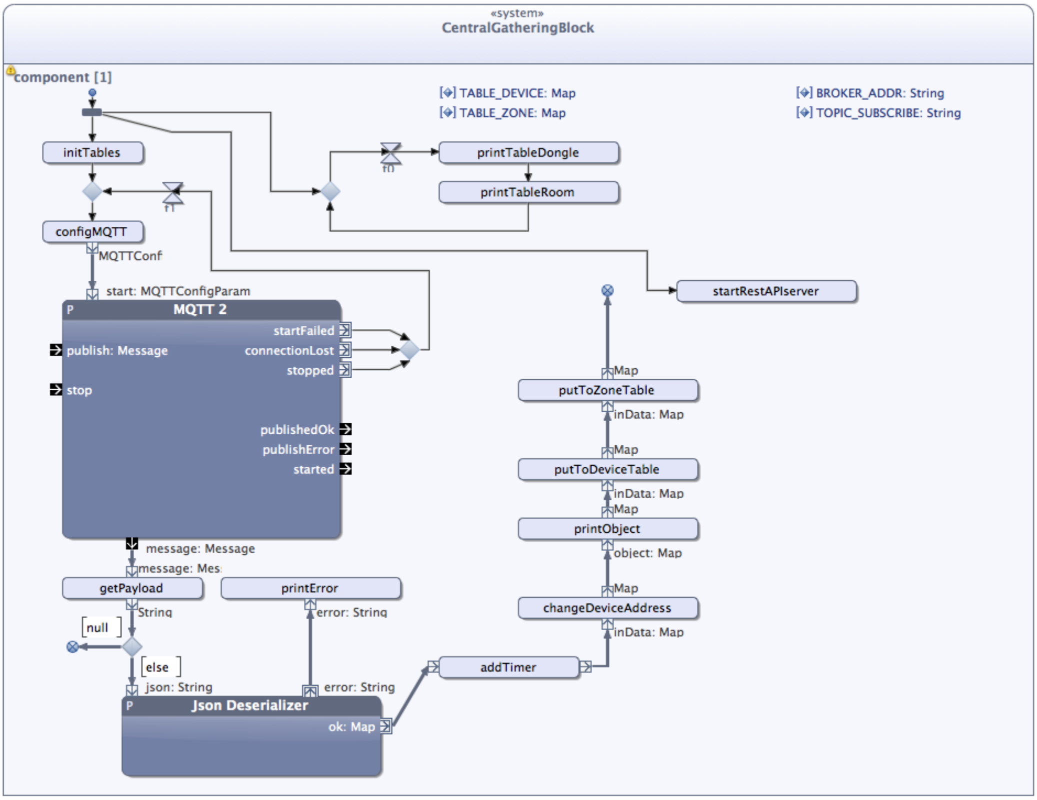

Figure 3.5: Central server architecture block

The final step is to aggregate the data of the MQTT message and put it to the device and the zone tables.

The REST API module was written using the SocketServer java class. This class basically opens the Transmission Control Protocol (TCP) port on the local system and listen for the incoming HTTP requests on it. After receiving the HTTP request REST API module is parsing the Uniform Resource Locator (URL) path of the request in order to get the operation which is necessary to implement on the server. The sample of the code for the parsing is presented in the appendix page 81.

3.4 Advertising device

For the system tests, it was necessary to develop a BLE advertising device, which will be in advertising mode with a custom advertising data. Bluetooth board from the evaluation KIT from the Nordic Semiconductor, which is presented on the Figure 3.6, was used for the project.

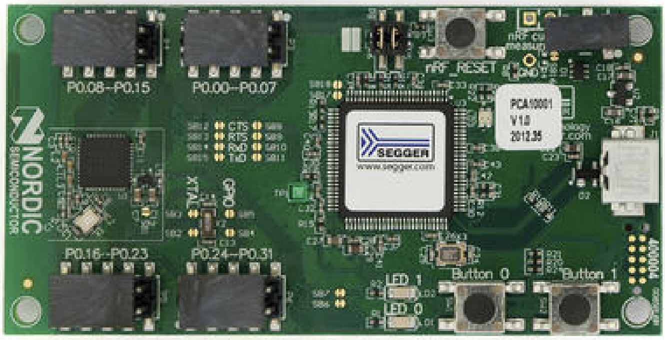

Figure 3.6: Central Nordic evaluation kit

The board is based on the Nordic Bluetooth chip nRF51822. nRF51822 is a system on a chip which is well suited for BLE and 2.4GHz ultra low-power wireless applications [Sem13]. The chip is based on ARM CortexTM-M0 CPU [Sem13]. It has GPIO and communication interfaces such as Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C) and Universal Asynchronous Receiver/Transmitter (UART). BLE peripheral applications on the controller are based on the S110 SoftDevice, which is BLE Peripheral/ Broadcast protocol stack solution.

The application developed for this board is configuring it in BLE non-connectable advertising mode, with advertising period of 250ms. I developed a special protocol "bluelocation" for the locating system and integrated it to the advertising data.A structure of protocol is presented on the Table 3.4. Where TYPE field is showing that the advertising data is specific for the location system, UUID is the identification of the device and TX POWER is the transmission power of the signal for the Nordic board. Later in the text I will use a name *'Nordic beacon'* for this advertising device.

Table 3.4: Bluelocation protocol structure

14bytes		
3 bytes	10 bytes	1 byte
TYPE	UUID	TX POWER

3.5 Conclusion

This chapter was a description of the location system, which was developed in order to conduct experiments, which will show if it is possible to use the Bluetooth Low Energy as a technology for the indoor location system.

The final system needs to meet the technical requirements, which were established in the beginning of the Chapter 3. Such requirements were:

1. To receive the BLE advertising signal from the BLE peripheral devices which are in advertising mode in different zones;
2. To be able to measure RSSI value of the received signal;
3. To collect the information about the scanned devices from different zones in one place;
4. To establish the location of the devices using the measurements of RSSI values. As a location result, system needs to identify and appoint a zone in which advertising device was present;
5. To have an interface which allows communication with other systems in order to provide location information.

At this point, the system description answered only 4 requirements out of 5. To receive and measure the RSSI of the BLE advertising signal in different zones, the approach of using Raspberry Pies with the Bluetooth modules for each zone was used. A collection of the scanned information from the scanner zones is implemented on the central server. As an interface to communicate with other systems, REST API interface was used.

For now, the unclosed requirement is to establishing the zone in which the advertising device was present. However, before that, the first thing, which was decided to do, is to provide series of experiments with establishing the quality of measuring the RSSI value. Chapter 4 shows the results of this experiment.

Chapter 4
System experiment

The designed and built system described in the Chapter 3 was evaluated through experiments. The goal was to check the hypothesis whether it is possible to establish a location of an object using the RSSI value. Eleven experiments with different configurations were made.

A laboratory in a medical faculty at NTNU was used for experiments. The laboratory simulated a hospital floor with a corridor and rooms for patients. A general plan of the area for the experiment is presented on a Figure 4.1. Area consists of three rooms and one hall, which is connected to the room 1. Two types of room compositions were used during the experiment. For the experiments that included room number 1 and room number 2 an open space was used, i.e. without barrier on the radio signal way. Experiments between room 1 and room 3 presented a situation in which a wall can be on the way of the radio signal. Each experiment was carried out 10 times. The final result of the experiments was used for location detection algorithm and analyzing the quality of the measurements.

The two devices which were advertising a signal were Nordic beacon 3.4 and iPad. The application for simulating BLE signal on iPad is called 'LightBlue'. iPad was chosen as an independent advertising device to check the possibility of tracking a random device.

Two types of network connections were used during the experiments: Ethernet and Wi-Fi. Since Wi-Fi is a radio signal, which operates on the same frequency as a Bluetooth, 2.4 GHz, it was decided to check if Wi-Fi signal will have an influence on quality of measuring RSSI value of Bluetooth devices.

Wi-Fi network connection was made to the private Wi-Fi zone using own router. This solution was chosen to provide clear experiments with measuring RSSI of Bluetooth signal and to remove probability of error sending packages during the connection to the public Wi-Fi network.

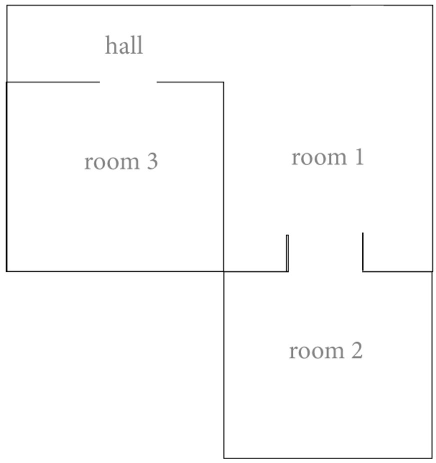

Figure 4.1: A general plan of the area for the system experiment

The independent variables for the experiment are: the speed of movement of the advertising device, type of advertising device, scanning interval of the zone scanner, type of network connection and type of environment (with wall or without wall between the advertising signal and zone scanner). The RSSI value of the advertising device was observed as a dependent variable.

The configuration overview for the experiments is presented in the Table 4.1. The table shows a configuration of each of the eleven experiments. Column 'speed' shows the speed of observed advertising device. Column 'Type of BLE Device' shows if iPad or Nordic beacon was used as an observed advertising device. Column 'Wall' shows type of environment, with or without the wall in between the zone scanner and the advertising device. Column 'Scanning interval' shows how often the zone scanner was scanning for the advertising devices. Column 'Network type' shows which type of connection was used, Ethernet or Wi-Fi.

4.1 Experiment 1

Experiment 1 was performed in an area without a wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2. The path of the experiment is marked on the figure with the green line. Red objects on the figure are the zone scanners. In this experiment room 1 is marked as zone 1 and room 2 is marked as zone 2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters Figure 4.3. Zone scanners were configured to send information about the scanned area every

Table 4.1: The configuration overview for the system experiments

Experiment number	*Independent variables*				
	Speed (m/s)	*Type of BLE Device*	*Wall*	*Scaning interval(ms)*	*Network type*
1	0	Nordic beacon	NO	1000	Ethernet
2	1.4	Nordic beacon	NO	1000	Ethernet
3	0	iPad	NO	1000	Ethernet
4	1.4	iPad	NO	1000	Ethernet
5	0	Nordic beacon	NO	250	Ethernet
6	0	Nordic beacon	NO	250	Wi-Fi
7	0	iPad	NO	250	Wi-Fi
8	1.4	Nordic beacon	NO	250	Wi-Fi
9	1.4	iPad	NO	250	Wi-Fi
10	0	Nordic beacon	YES	250	Wi-Fi
11	1.4	Nordic beacon	YES	250	Wi-Fi

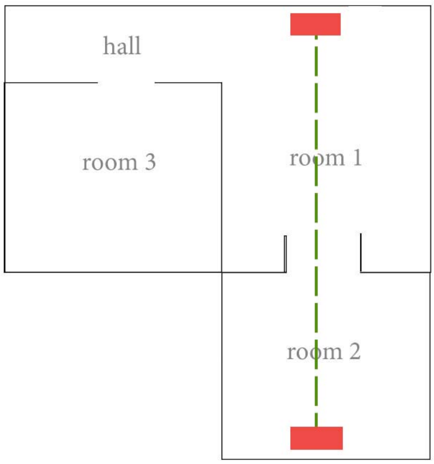

Figure 4.2: A general plan for the system experiments with out wall

second. Ethernet connection was used as a network connections. Nordic Beacon was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during a standing position and included 10 attempts for each meter with interval of one minute.

Results of the experiment are presented on the Figure 4.4. Blue and red dots show RSSI measurements obtained for zone 1 and zone 2 on different distances. A blue and a red line show the average of the 10 RSSI measurements, which were made for each zone. Black and green high/low bars show one standard deviation around

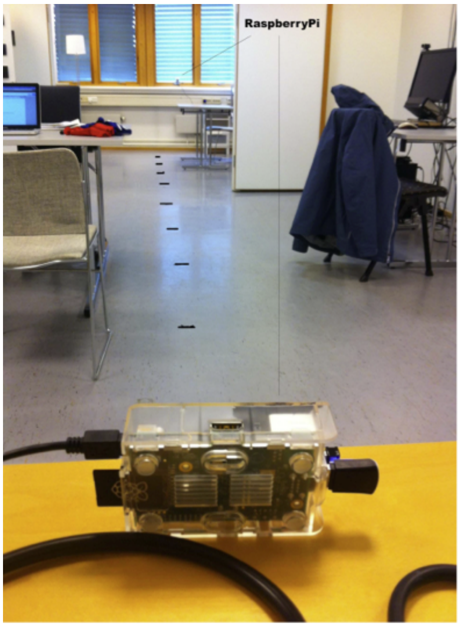

Figure 4.3: Photo of the lab from the system experiment

the average value for the zones on different distances.

Standard deviation is a measure of dispersion in a frequency distribution, equal to the square root of the mean of the squares of the deviations from the arithmetic mean of the distribution [sta14]. The formula (4.1) shows the equation for standard deviation, where σ is a standard deviation, X_i are values of the set and μ is a mean value of the set. In the experiment one set includes RSSI, which were measured on one specific distance for one zone.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \mu)^2}{N}}; \mu = \frac{\sum_{i=1}^{N} X_i}{N}. \tag{4.1}$$

The higher standard deviation the larger the range of the spread of the values. In statistics there is a ‘three-sigma rule’ for standard deviation in normal distribution., which tells that :

1. 68.27% of the values lie within one standard deviation of the mean;
2. 95.45% of the values lie within two standard deviation of the mean;
3. 99.73% of the values lie within two standard deviation of the mean.

On the graphs, only one standard deviation is presented.

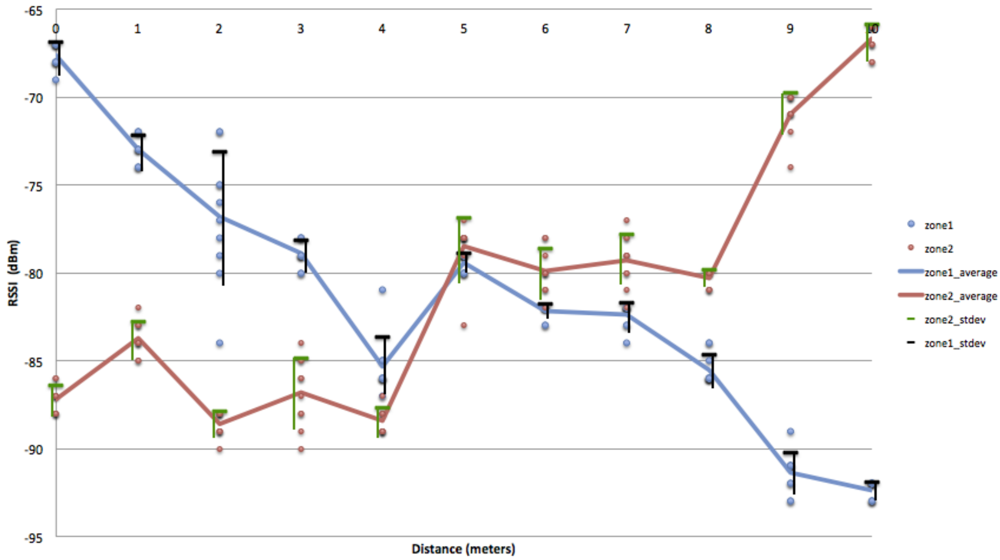

Figure 4.4: System experiment 1 result

First point to observe is a standard deviation of the RSSI values for zone 1 and zone 2 during the same distance of different experiment repeats. For the zone 1, the average of standard deviations was 1.07. For the zone 2, the average of standard deviations was 1.1, which is almost equal to the standard deviation of the zone 1. Zone 1 had great standard deviation equaled to 3.7 on the distance of 2 meters. This is possible to see from the graph. This outcome was caused by two measurements, which were -72dBm and -84dBm. These measurements could be caused by the signal measurement error on a device or the signal interference. Further we will call it unexpected measurements since we do not know the nature of this behavior.

A tendency of changing RSSI value while changing the distance can be observed from the average measurement of zones on the blue and the red line. Lines show that, generally, RSSI value is decreasing while moving away from the particular zone and increasing while moving closer towards the zone scanner. Sometimes, this tendency showed unexpected results. On the distance of 5 meters for example, the value for zone 1 was increasing instead of decreasing. In addition, on the distance 1 meter, the value for zone 2 was decreasing instead of increasing and on intervals between 2 and 4 and between 5 to 8 RSSI value was almost the same. Also on the distance of 5 meters we can see that RSSI values of both zones unexpectedly increased. We can assume now that changing of the RSSI value can be dependent on both zones and try to make normalization of the measurements by using (4.2).

$$normalization\_value = Zone1\_RSSI - Zone2\_RSSI \tag{4.2}$$

Graph Figure 4.5 shows the result after applying normalization formula. Blue dots present difference between zones RSSI values, at the same time and on the same distance, calculated by FormulaNormalization.

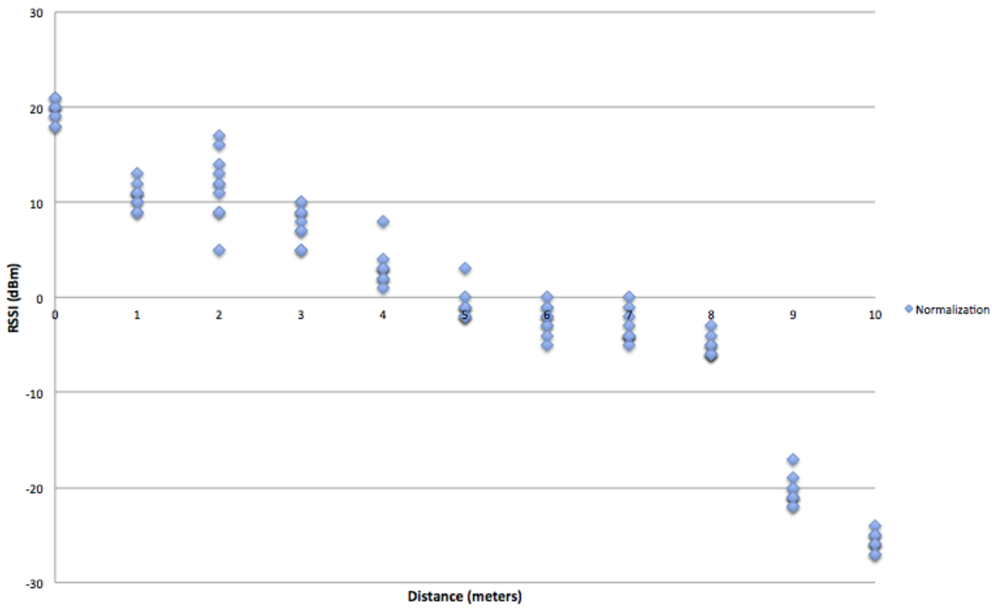

Figure 4.5: System experiment 1 normalization result

This approach can be used to find out in which zone an object is present or to establish the position of an object within the zones. We can see, that on the interval from 0 to 4 meters, the normalization values were positive. Therefore it can be assumed that object was present in the zone 1. On the distance between 5 and 7 meters, the normalization results were both positive and negative which does not allowed us to tell precisely in which zone the object was. On the other hand it can be assumed that these results showed that object was in the middle of two zones. On the interval between 8 and 10 meters, the results were negative which is providing a conclusion that the object was in the zone 2.

To conclude this experiment, it was found that RSSI value depends on the distance from advertising device to the object and the closer the object is to the zone the higher the RSSI value is. Standard deviation of RSSI values for the same zone can vary on different distances and sometimes can be high because of unexpected measurements. Additionally, it was found that normalization by formula (4.2) can be used in order to establish in which zone the user is present, under the conditions that were presented in the experiment.

4.2 Experiment 2

Experiment 2 was performed in an area without the wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every second. Ethernet connection was used as a network connection. Nordic Beacon was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during movement with the speed 1.4 m/s. One experiment try included moving from zone 1 to zone 2 and10 tries with interval of 1 minute were made during the experiment.

The goal of the experiment was to check how RSSI values will change if the object will move. Results are presented on the Figure 4.4.

Standard deviation of the RSSI values showed different results comparing with the experiment 1. For the zone 1, the average of standard deviations was 1.98. For the zone 2, the average of standard deviations was 2.34. The average of standard deviation increased comparing with the experiment 1, where measurements were made while standing. Also, higher values were obtained for the zone 2 than they were for the zone 1. Great standard deviation for the zone 1 was found on the distance of 0 meters and it equaled 3.07. For the zone 2, great standard deviation was on the distance of 10 meters and it equaled 3.95. Since great deviations were noticed only in one point for each zone, it can be assumed that it happened because of unexpected measurement of RSSI.

Average values of the RSSI measurements for the two zones, which are presented as a blue and a red line, showed the same tendency, i.e. RSSI values are changing while the distance is changing. The same tendency was observed in experiment 1. The values of the blue and the red line in this experiment were ranging from -88dBm to -76dBm but in experiment 1 it was between -63dBm and -92dBm. Considering this fact, we can assume that during walking the RSSI values are decreasing. The cross of the blue and the red line on the graph shifted to the zone 2 compared to

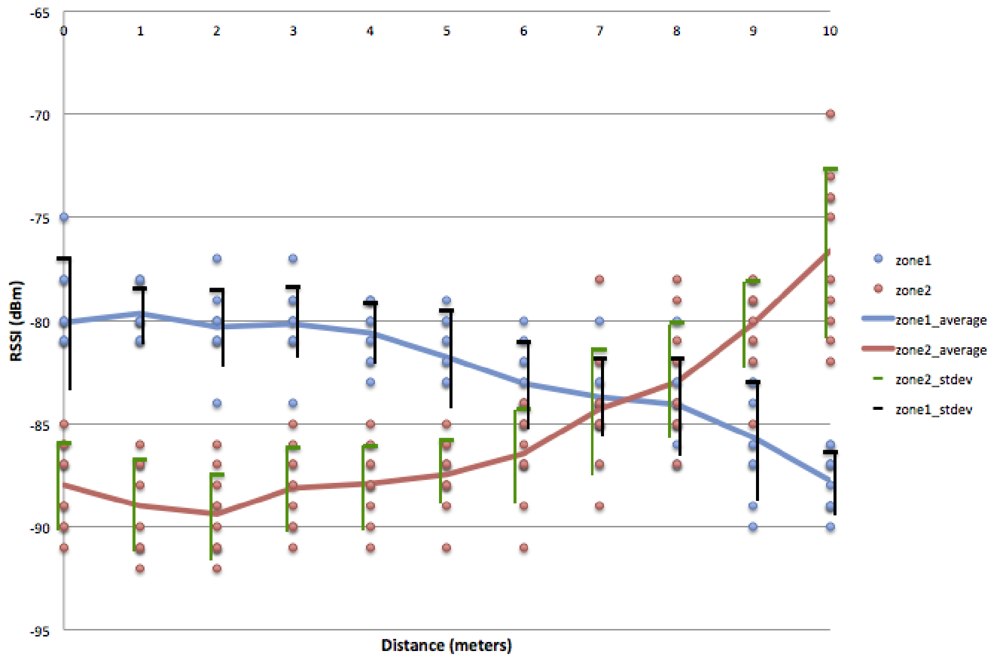

Figure 4.6: System experiment 2 result

graph on the Figure 4.4. This is because our scanning interval was 1 second but during this time it is possible to move around 1.4 meters.

The results of normalization process calculated by formula (4.2) are presented on the Figure 4.7.

It can be observed that spreads of the values are bigger comparing with the graph on the figure 2 from experiment 1, because of unexpected measurements. On the interval from 0 to 4, we can assume that object was present in the zone 1 since all values are positive. On the interval from 5 to 6, few values are negative but most of the values are positive which can tell that object was still in the zone 1. Values on the distance 7 and 8 show that there is equal amount of positive and negative values. This is because the object was in the middle and as mentioned before, since the period of the scan is 1 second the center shifted to zone 2. Values on the distance 9 and 10 show that the object was in zone 2.

From the results we got during the experiment 2, it can be concluded that if the object has a speed, the standard deviation of the measurements will be greater

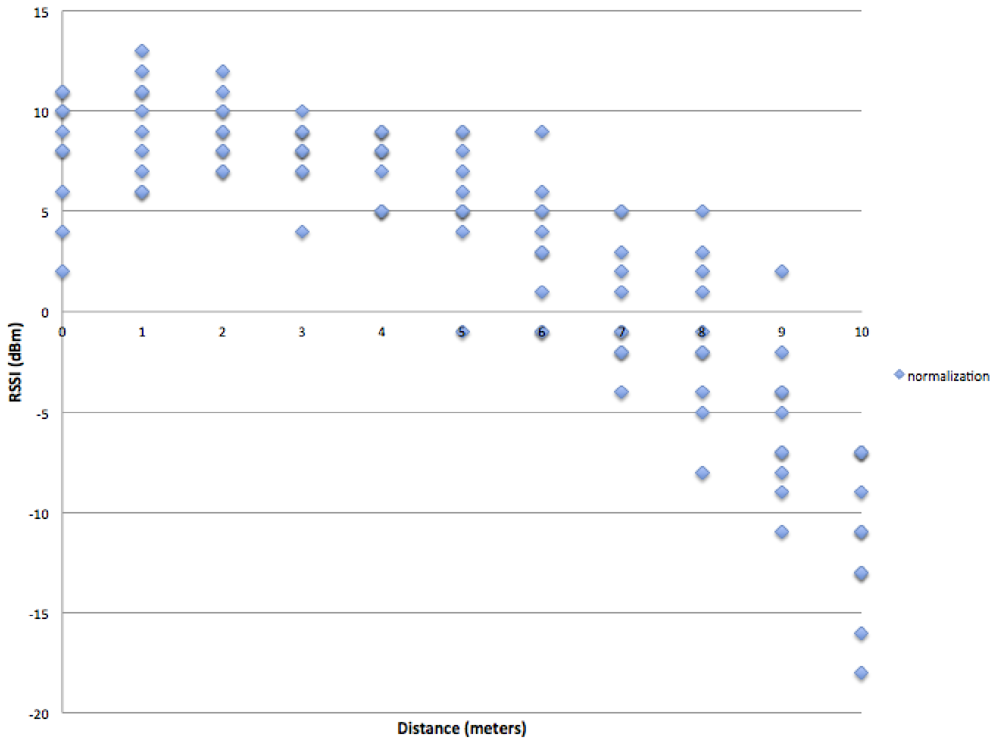

Figure 4.7: System experiment 2 normalization result

compared to the measurements on the same distance in a standing position. Also, RSSI value range is lower during the movement than it is during the standing. Sending interval parameter does not have an impact on the measuring parameters while the object is standing but during walking it can be reason of incorrect measurement of the zones' crossing place. The normalization formula (4.2) provides almost the same results for the zone detection as it did experiment 1, which gives an opportunity to assume that this formula can be used for a future algorithm of the zone detection.

4.3 Experiment 3

This experiment was made to check whether it is possible to use random devices, which are advertising BLE signal in order to detect their position.

The experiment was performed in an area without a wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every second. Ethernet connection was used as a network connection. In this experiment, instead of Nordic beacon, iPad was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during a standing position and included 10 attempts for each meter with interval of one minute.

Results of the experiment are presented on the Figure 4.8.

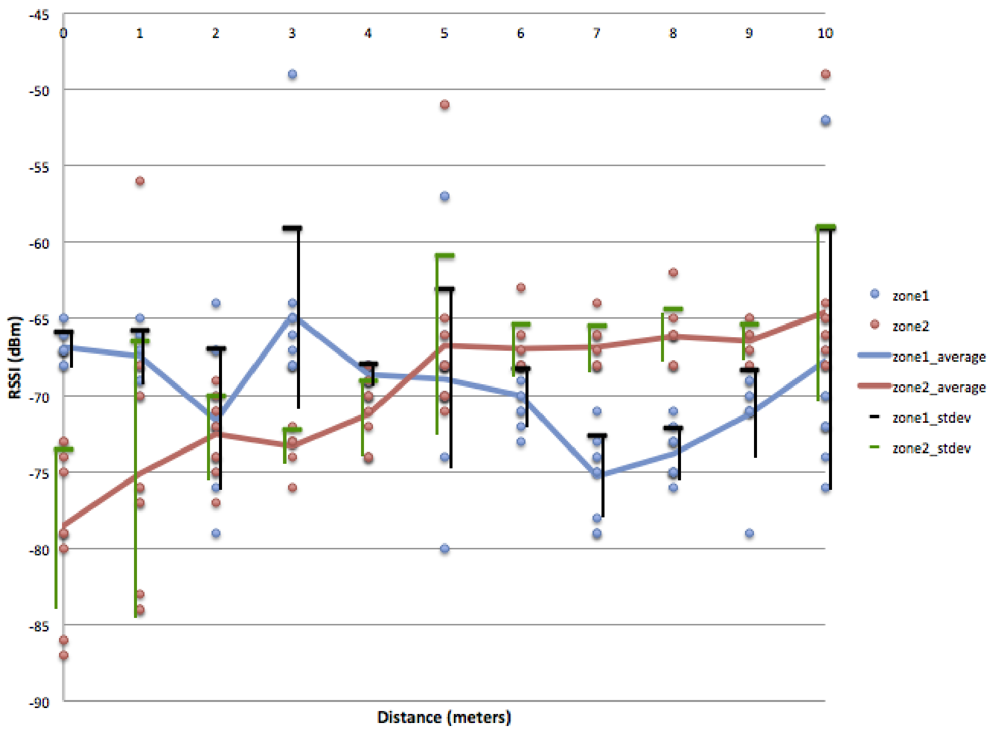

Figure 4.8: System experiment 3 result

The spread of the RSSI values showed chaotic results. On the distance of 1, 3, 5 and 10m, for example, few measurements were high. The average of standard deviations for the zone 1 was 3.36. For the zone 2, the average of standard deviations was 3.33, which is almost the same like for zone 1. This values are larger comparing with the experiment 1 and experiment 2, where Nordic beacon was used. This large deviation can cause an error while detecting in which zone the object was present.

Graph shows that all RSSI values were higher than values in experiments 1 and 2 for the Nordic beacon. This result shows that transmission power of the iPad is higher than transmission power of the Nordic beacon. Using this observation, it can be assumed that if a device has a stronger transmission power it can cause an incorrectly high RSSI value inside the building because of higher probability of the signal interference.

The blue and the red line show the tendency of an increase in RSSI value while being closer to the object and a decrease while being further from the object. However, a blue line, which represents the zone 1, has a strange behavior on the distances 3, 8, 9 and 10m. It showed that the RSSI values were increasing instead of decreasing. The nature of this phenomenon is not clear and as assumed before, it could be caused by strong transmission power of the BLE signal on the iPad. On the other hand, the RSSI values on the red line show a stable growing tendency.

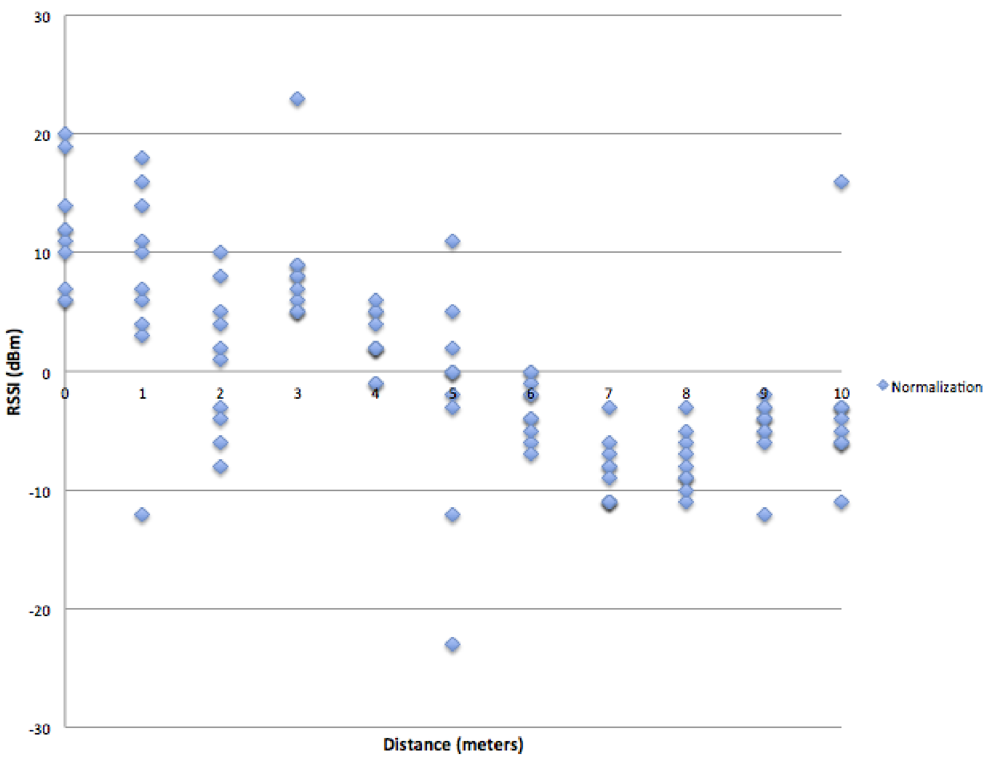

Figure 4.9: System experiment 3 normalization result

Figure 4.9 shows the result of normalization process calculated by formula (4.2).

The general tendency of the graph on the Figure 4.9 shows that while changing the position of the RSSI values from zone 1 to zone 2, the normalization value changes from positive to negative. Few errors can beseen on the distance 1, 3 and 10m. On the distance 2m, the amount of positive values equal to the amount of negative values. Additionally, the same behavior can be observed in the middle of the two zones. This would consequently cause an error in zone detection.

To conclude, during this experiment it was found that the standard deviation of the RSSI values was higher for the iPad than for the Nordic beacon obtained in experiments 1 and 2. The normalization process shows that an unexpected result can occur, like the one on the distance 2m on the Figure 4.9, which will cause incorrect detection of the zone. On the other hand, the general tendency of a change in RSSI value within the position change was similar to the tendency in experiments 1 and 2. Considering the results obtained in this experiment, it can be concluded that it is possible to use the iPad or other random advertising devices for the zone detection with a different level of errors detection.

4.4 Experiment 4

The goal of the experiment was to figure out if the movement of iPad will have an influence on the RSSI values.

Experiment 4 was performed in an area without the wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every second. Ethernet connection was used as a network connection. iPad was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during movement with the speed 1.4 m/s. One experiment try included moving from zone 1 to zone 2and 10 tries with interval of 1 minute were made during the experiment.

Graph on the Figure 4.10 shows the results of the experiment.

The average of standard deviations for the zone 1 was 4.52. For the zone 2, the average of standard deviation was 3.71. Standard deviations for both zones were higher but more stable comparing with the experiment 3.

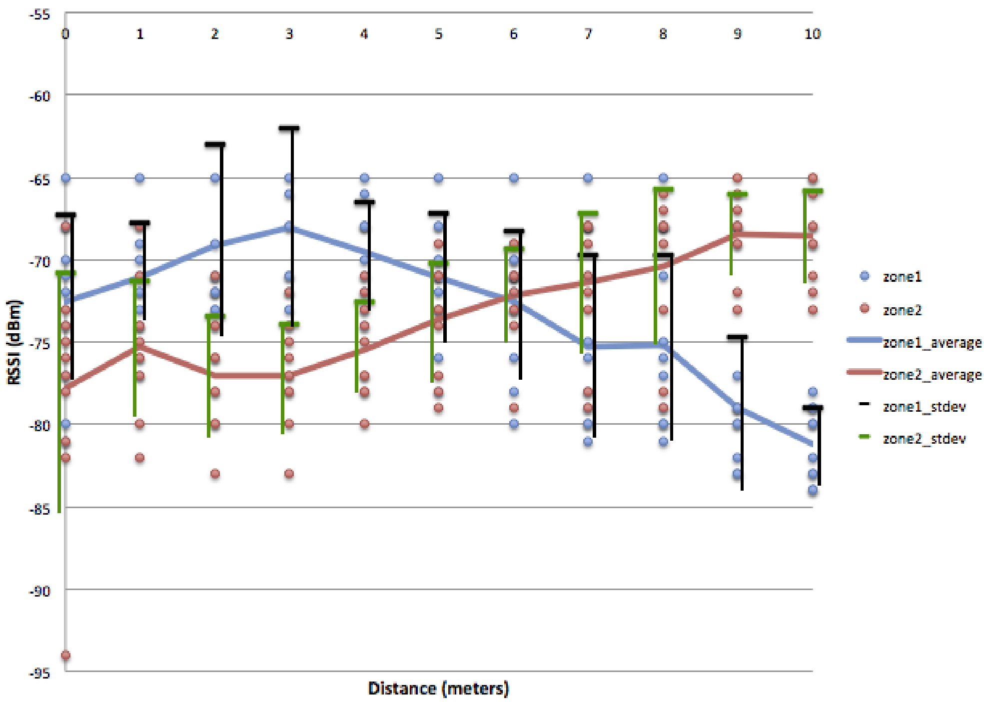

Figure 4.10: System experiment 4 result

The blue and red line, which both represent the average RSSI values, are showing the correct tendency, which is that RSSI value is changing when the distance is changing. The error values can be seen in zone 1 on the interval between 0 and 3, where RSSI values have tendency to increase instead of decrease. This phenomenon can be observed on the blue line.

The results of the normalization are presented on the Figure 4.11. Generally, the RSSI values are less spread and contain less unexpected values compared to the experiment 3. Unexpected results were obtained on the distances 0, 1 and 2m. These results could be caused by the interference. But let's assume that on the interval from 0 to 4m the object was in the zone 1, from 5 to 8m the object between zones 1 and 2 and from 9 to 10 the object was in zone 2. Similarly like in experiment 2, it can be seen that the middle of the two zones was shifted to the zone 2. Such deviation could be caused by 1 second scanning interval on the Raspberry Pi.

The experiment showed that during the movement the standard deviation of the RSSI values for iPad was higher compared to the spread of RSSI values during

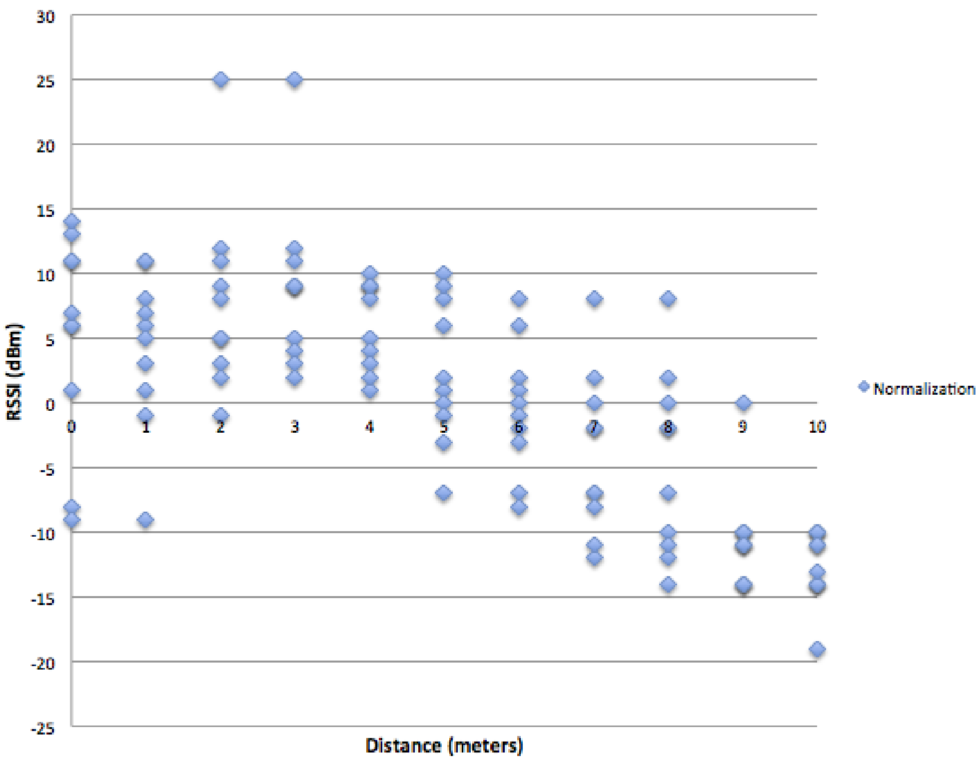

Figure 4.11: System experiment 4 normalization result

standing. Few unexpected values occurred during the experiment but in general, the results after normalization process showed that it is possible to detect the zone in which the measurement was made.

4.5 Experiment 5

The goal of this experiment was to determine whether changing the interval of scanning on the Raspberry Pi will not have an influence on the RSSI values during standing when using Nordic beacon. It was expected that results of this experiment will be similar to the results of experiment 1.

Interval can be changed in the BluetoothAgregator block of the Rasbperry Pi. The part of the block, which is responsible for the interval is presented on the Figure 4.12. A timer before operation logThreadAmount is setting interval with which information is pushed to the MQTT channel. The interval for this experiment was decreased from 1 second, which was used in previous experiments, to 250 milliseconds.

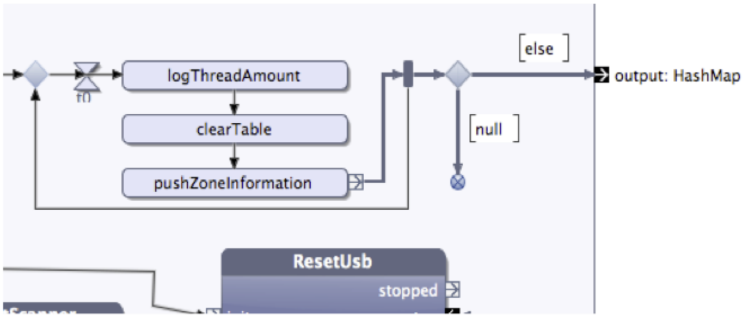

Figure 4.12: BluetoothAgregator block interval setting part

The experiment is performed in an area without a wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every 250 milliseconds. Ethernet connection was used as a network connection. In this experiment, Nordic beacon was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during a standing position and included 10 attempts for each meter with interval of one minute.

The results of the experiments are presented on the Figure 4.13.

Standard deviations show similar result to experiment 1. For zone 1, the average of standard deviations was equal to 1.15 and for the zone 2 it was equal to 1.25. In experiment 1 we got 1.1 and 1.07,respectively. Here I will assume that changing the interval of scanning does not have an influence on the deviation of the RSSI during standing.

The red and the blue line show the same tendency like in experiment 1. Only on the distance 2 in zone 1 the line went up instead of going down. The nature of this behavior is not clear.

The results of the normalization process by formula (4.2), are presented on the Figure 4.14.

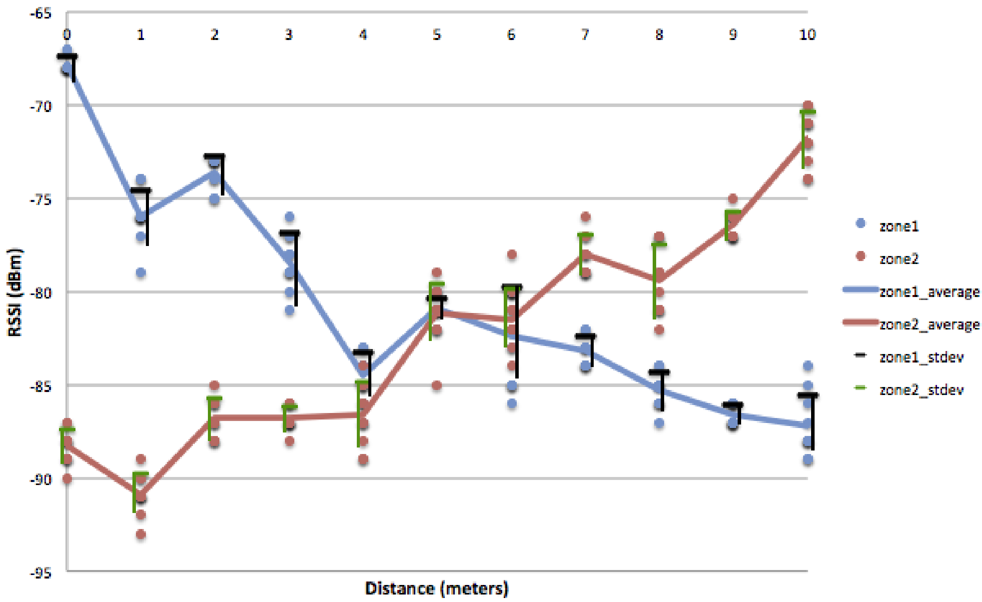

Figure 4.13: System experiment 5 result

Results are similar to the experiment 1. On the distances of 5 and 6 meters, there is an equal amount of negative and positive values, which is suggesting that advertising device was in the middle of two zones.

This experiment showed that changing the scanning interval on Raspberry Pi does not have an influence on the measuring of the RSSI during standing.

4.6 Experiment 6

In this experiment, Wi-Fi as a network connection for the zone scanners was included. The goal of this experiment was to observe how Wi-Fi signal will influence the RSSI values of Bluetooth signal during the standing position.

Experiment 6 was performed in an area without a wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every 250 milliseconds. Wi-Fi connection was used as a network connection. Nordic beacon was used as an advertising device.

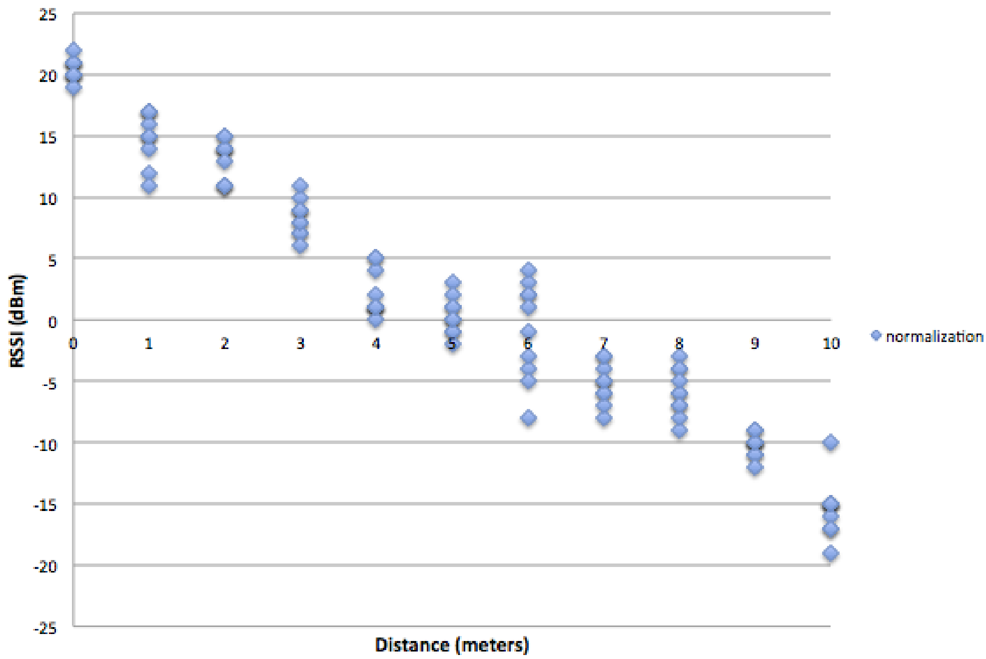

Figure 4.14: System experiment 5 normalization result

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during a standing position and included 10 attempts for each meter with interval of one minute.

The results of the experiment are presented on the Figure 4.15.

This experiment is compared to the experiment 1. The start point to observe is a standard deviation of the RSSI values on the same distance. The average of standard deviation for zone 1 was 2.22. The maximal standard deviation, 5.17, was obtained on the distance 5. This was caused by an unexpected measurement of the RSSI, which equaled -64dBm. The average of standard deviation for zone 1 in this experiment was two times bigger compared to the average of standard deviations in the experiment 1, where it was 1.07. The average of standard deviations for zone 2 was 1.3. The results show that Wi-Fi signal increased the standard deviation of the RSSI values.

The tendency of changing RSSI value with the change of a distance, which is

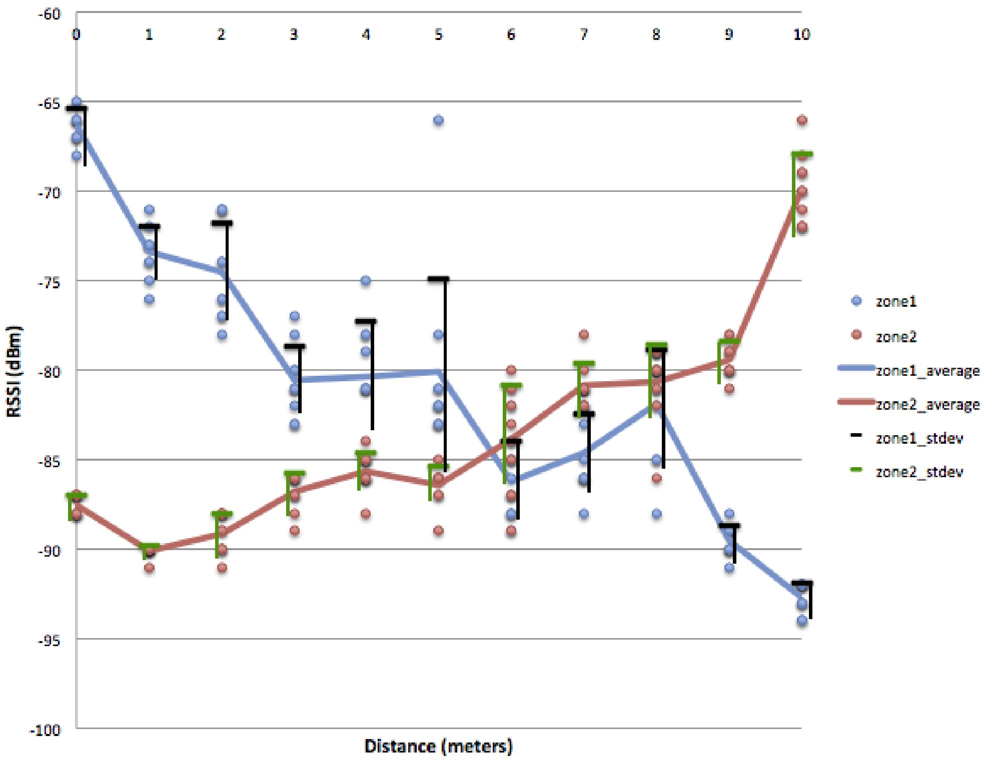

Figure 4.15: System experiment 6 result

represented by the blue and the red line, shows the correct pattern, i.e. an increase in RSSI values while decreasing the distance between the advertising device and the zone scanner and a decrease while increasing the distance. The error behavior was found on the distance 7 and 8m for the zone 1. Instead of decreasing, graph shows increasing tendency. This could be caused by interference in this zone or an influence of Wi-Fi module on the measurements. In general, both, the blue and the red line show the same tendency as in experiment 1.

The results of the normalization, calculated by formula (4.2), are presented on the Figure 4.16.

We can see that on interval from 0 to 5m, the normalization values are positive. This trend is suggesting that the advertising device was in the zone 1. On the distance 6m, the number of positive and negative values is equal since the device was in the middle of zones. On the distance 7m, mainly negative values were obtained because

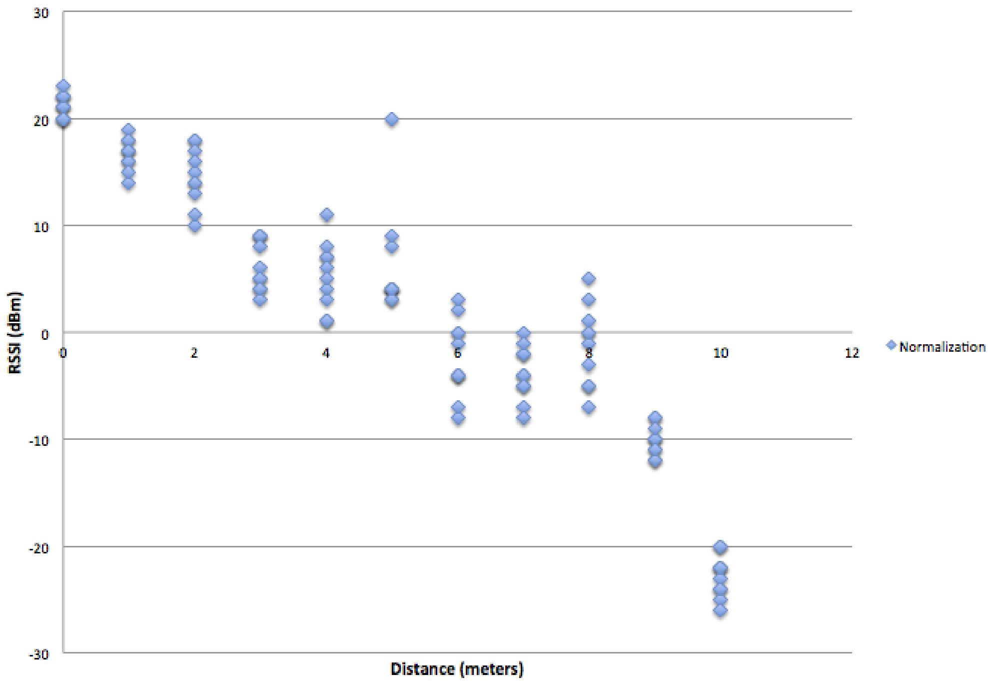

Figure 4.16: System experiment 6 normalization result

the advertising device entered zone 2. Distance 8m is with unexpected values and shows the same behavior like when device was in the middle. On the distances 9 and 10m, only negative values were recorded. This shows that advertising devices were in the zone 2. Comparing with the experiment 1, almost the same results were achieved with the unexpected behavior on the distance 8.

This experiment showed that Wi-Fi has an impact on the standard deviation of the RSSI values and generally, we got higher average of standard deviations. The tendency of changing RSSI value with the change of distance appears to be the same as the one in experiment 1. The results of normalization process showed unexpected behavior on the distance 8m but in general, results aree similar to those reached in the experiment 1. To conclude this experiment, it has to be said that the use of Wi-Fi as a network link has an influence on RSSI values , but the results can be still useful for the zone detection.

4.7 Experiment 7

This experiment was carried out order to observe if Wi-Fi signal will decrease the quality of measurement of the RSSI values of the Bluetooth signal transmitted from the iPad during the standing position and to establish if the results will be useful for the zone detection. The results of the experiment are compared to the experiment 3.

Experiment was performed in an area without a wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every 250 milliseconds. Wi-Fi connection was used as a network connection. iPad was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during a standing position and included 10 attempts for each meter with interval of one minute.

Results are presented on the Figure 4.17.

The average of standard deviations for the zone 1 was 9.54. For the zone 2, the average of standard deviations was 8.12. Standard deviations increased compared to the experiment 3.

On the blue and the red line, which both show the tendency of changing the RSSI values, it can be observed that the red line is almost horizontal which is indicating a wrong tendency. Moreover, the blue line has errors on the distance 4m. Comparing with experiment 3, the results became worse since the tendency of changing the RSSI values is not so clear as before.

The results of the normalization process calculated by formula (4.2) are presented on the Figure 4.18.

From these results it can be seen that the spread of the normalized values is greater than the spread in experiment 3. On the interval from 0 to 5m, most values are positive which means that object was located in the zone 1. On the interval from 6 to 10m, most of the values are negative which means that object was in the zone 2. Some of the values could provoke the error in the zone detection. As an example, on the distance 3m the RSSI value was -30dBm, the same value could be expected on the distances 8, 9 or 10m. The unexpected results can be due to an addition of Wi-Fi connection to the system.

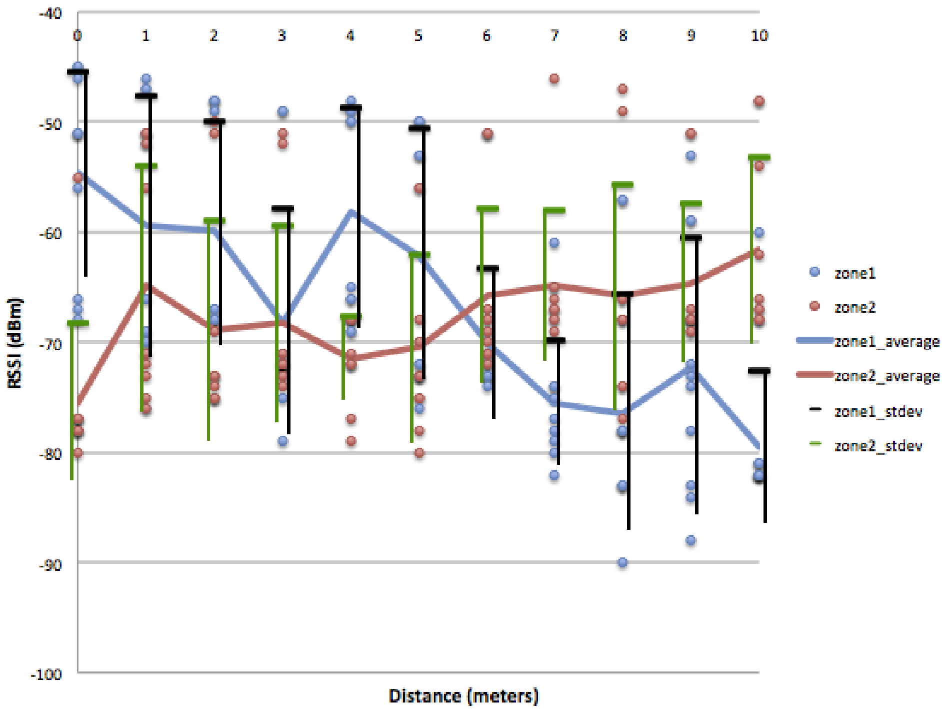

Figure 4.17: System experiment 7 result

This experiment showed, that Wi-Fi connection influence the measurement of the RSSI and increases the amount of unexpected measurements. Compared to the experiment 3, the standard deviations of the RSSI values became greater on the same distance. Also, after the normalization process it became difficult to establish in which zone the measurement was made. But even after obtaining such results, it is still possible to detect the zone with some probability.

4.8 Experiment 8

In experiment 2, there was a problem that the cross of RSSI values was shifted to the zone 2. This could have been caused by the interval of sending information from the zone scanner. Experiment 8 was carried out in order to find out whether the cross can shift closer to the middle of the zones when decreasing the sending interval.

The experiment was performed in an area without a wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

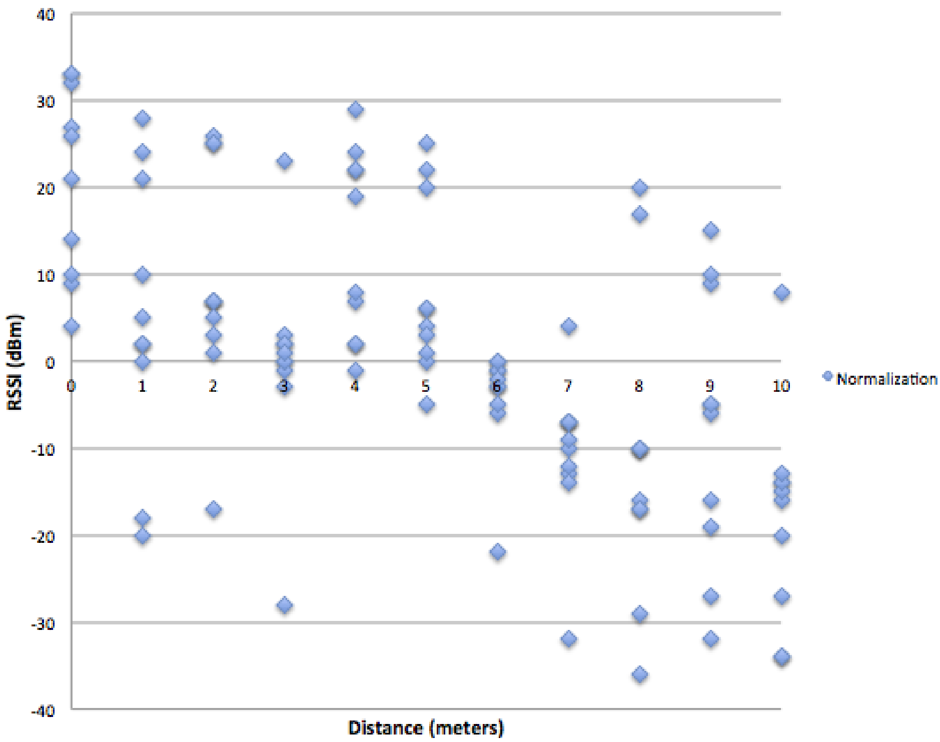

Figure 4.18: System experiment 7 normalization result

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every 250 milliseconds. Wi-Fi connection was used as a network connection. Nordic beacon was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during movement with the speed 1.4 m/s. One experiment try included moving from zone 1 to zone 2 and 10 tries with interval of 1 minute were made during the experiment.

The results of the experiment are presented on the Figure 4.19.

For the zone 1, the average of standard deviations was equal to 3.96. For the zone 2, the average of the standard deviations was equal to 3.18. Compared to the experiment 6, where the averages of standard deviations were 2.22 for zone 1 and 1.30

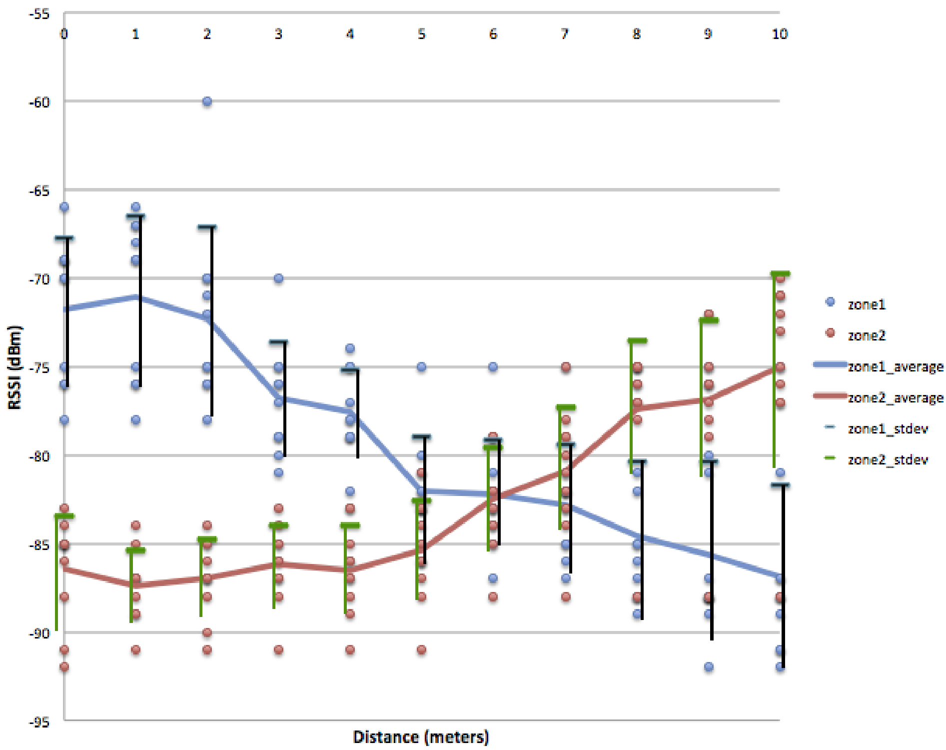

Figure 4.19: System experiment 8 result

for zone 2, it can be assumed that movement has an influence on the measurement of the RSSI value. The same tendency was achieved in experiment 2.

The red and the blue line, which represent the tendency of changing RSSI values while the distance is changing, show that RSSI values are decreasing while moving away from the particular zone and increasing while moving closer towards the zone scanner. Cross of the lines is found on the distance of 6 meters, which is closer to the middle. This shows that decreasing the interval of scanning on the zone scanner affects the RSSI value.

The results of the normalization process calculated by formula (4.2) are presented on the Figure 4.20.

The results from this experiment show a greater spread of the normalization values, on the same distance, comparing with the experiment 2. This was caused by Wi-Fi signal. However, this would not have an effect on the zone detection, which

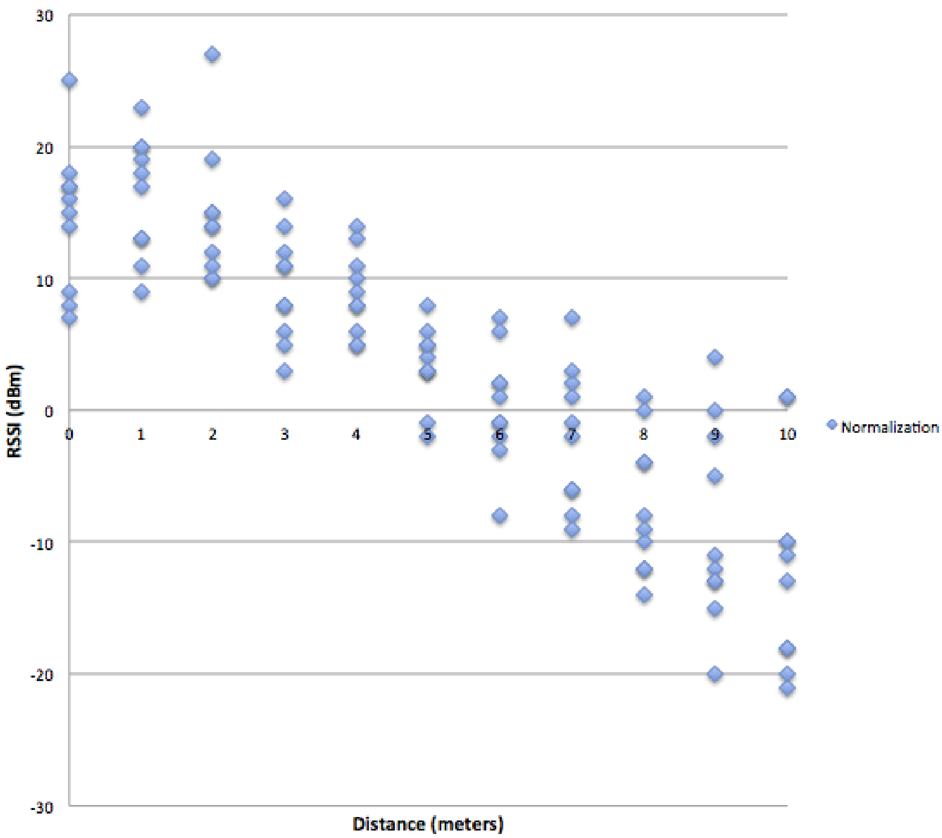

Figure 4.20: System experiment 8 normalization result

means that it was still possible to detect the correct zone from the values.

To conclude, the experiment pointed out that the Wi-Fi signal has an influence on the measurement of the RSSI value of the Bluetooth signal from the Nordic beacon. But on the other hand, the results of normalization still give the possibility to detect the correct zone. Also, by changing the interval of sending of information on the zone scanner to lower, the resolution of the RSSI values during walking was improved.

4.9 Experiment 9

In this experiment the signal from iPad was measured during movement. Before, the results of experiment 7 showed that after Wi-Fi connection was added, the quality of measurement of the RSSI values decreased for iPad, but results were still useful enough to be able to detect the location of the iPad. Adding a movement while measuring the RSSI is increasing the standard deviation. Therefore, this experiment needs to show if iPad will be useful in the location detection during movement.

The experiment was performed in an area without a wall between the zone scanners. A general plan of the experiment is presented on the Figure 4.2.

The distance between zone scanners (Raspberry Pi) equaled 10 meters. Zone scanners were configured to send information about the scanned area every 250 milliseconds. Wi-Fi connection was used as a network connection. iPad was used as an advertising device.

For process evaluation, measurements were made every meter within ten meters, including the starting position marked as 0m. The direction of position change was from zone 1 to zone 2. Measurements were obtained during movement with the speed 1.4 m/s. One experiment try included moving from zone 1 to zone 2 and 10 tries with interval of 1 minute were made during the experiment.

The results are presented on the Figure 4.21.

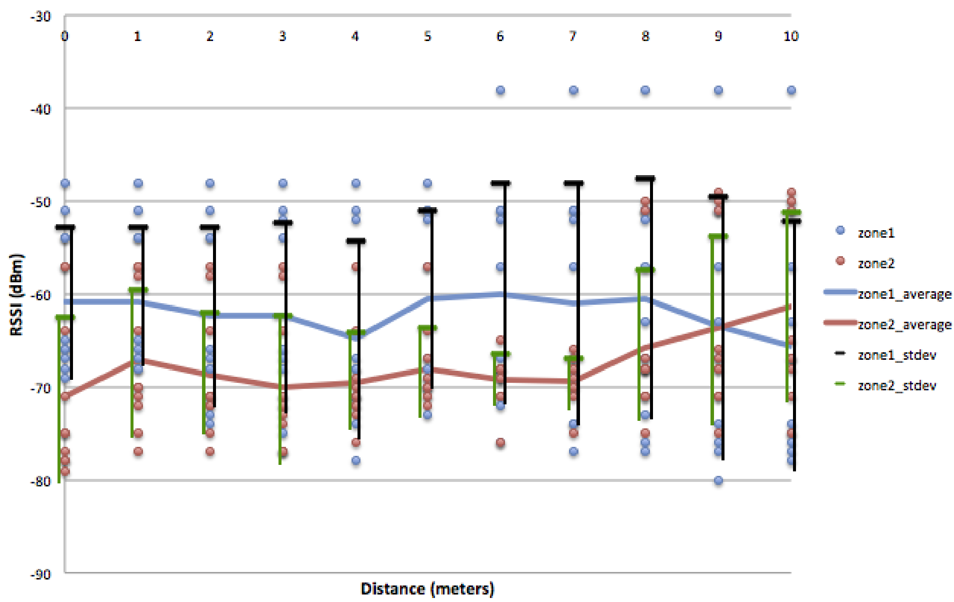

Figure 4.21: System experiment 9 result

In this experiment, standard deviations of the RSSI values was similar to the results achieved in experiment 7. For the zone 1, the average of the standard deviations was 11. For the zone 2, the average of standard deviations was 6.74.

The red and the blue line, which represent the tendency of changing RSSI values while the distance is changing, show this tendency only on the distances of 8, 9 and 10 meters.

The results of the normalization process should reveal whether it is possible to detect the zone in which the advertising device is present. Results are presented on the Figure 4.22.

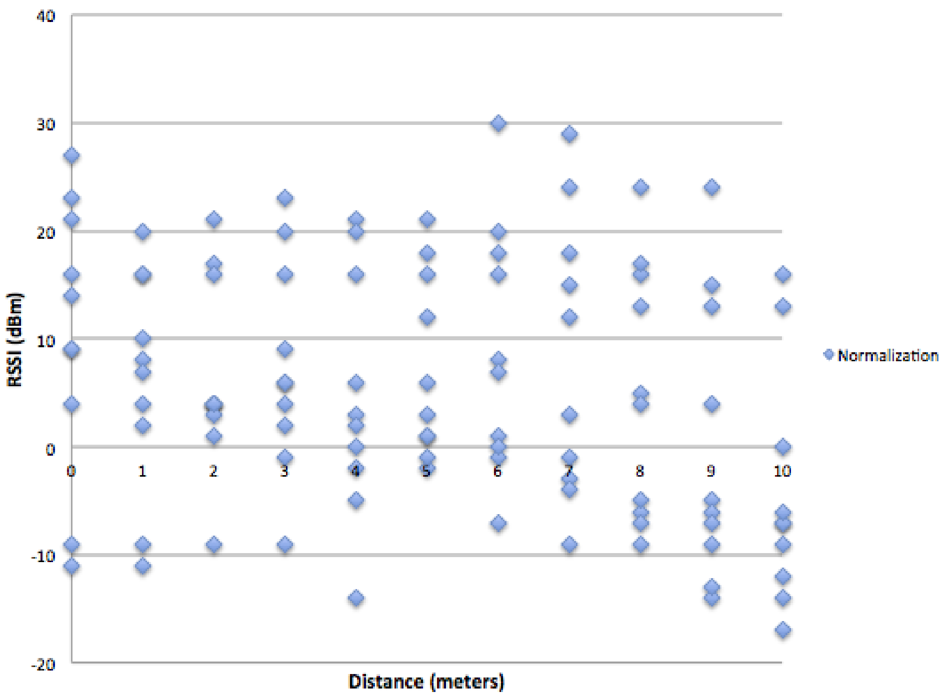

Figure 4.22: System experiment 9 normalization result

The results of normalization show major spreads of the values, which is making it difficult to detect the zone. On the distances of 4, 6, 7, 8 and 10 m, it is not possible to estimate the zone in which the advertising signal is present, since the results are not correct. In addition, when these results were compared with the results obtained in experiment 4, it was found that extra unexpected results were acquired, which were most likely caused by the Wi-Fi signal.

This experiment shows that, a movement does not affect the standard deviation.

However, the normalization process shows results where it is not possible to correctly detect the zone in which iPad was present. This behavior, caused by Wi-Fi signal, has and impact on the measurements of the RSSI value of the iPad and erases the possibility to use the data for the zone detection.

This experiment revealed that it might be difficult to maintain the detection of a random device by the system developed in the Chapter 3. The results can be chaotic and will depend on the strength of the transmission power of the device. Consequently, it has been decided to continue experiments only with the advertising devices which were developed especially for the system with the configured transmission power of Bluetooth signal such as Nordic beacon.

4.10 Experiment 10

All previous experiments were carried out in the environment where there was no wall between the advertising device and the zone scanner. In experiments 10 and 11, the environment with the wall was used in order to prove that it is possible to detect the zone of the advertising device while it is located behind the wall.

A general plan of the experiment is presented on the Figure 4.23. The path of the experiment is marked on the figure with the green line. Red objects on the figure are the zone scanners. The direction of position change was from room 1 to room 3. In this experiment room 1 is marked as zone 1 and room 3 is marked as zone 2.

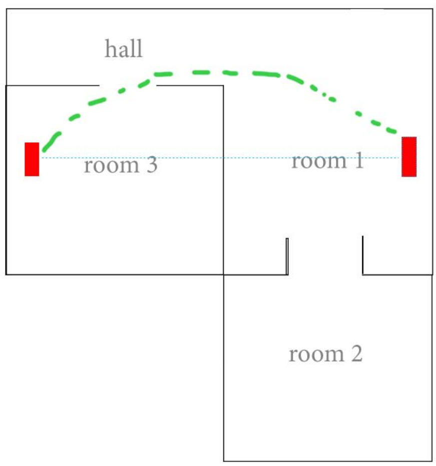

Figure 4.23: A general plan for the system experiments with wall

The experiments were carried out in order to prove the initial hypothesis, i.e. it is possible to detect the region where advertising device change position from zone 1 to zone 2. Zone 1 includes room 1 and hall, and zone 2 includes just room 3. The interval between 0 and 3 meters includes room 1, interval between 3 and 6 meters includes hall and interval between 6 and 7.5 meters includes room 3.

The total distance of the path was 7.5 meters. Measurements were made every 1.5 meter, including the starting position marked as 0m. These measurements were obtained during a standing position and included 10 tries for each position with interval of one minute.

Results of the experiment are presented on the Figure 4.24.

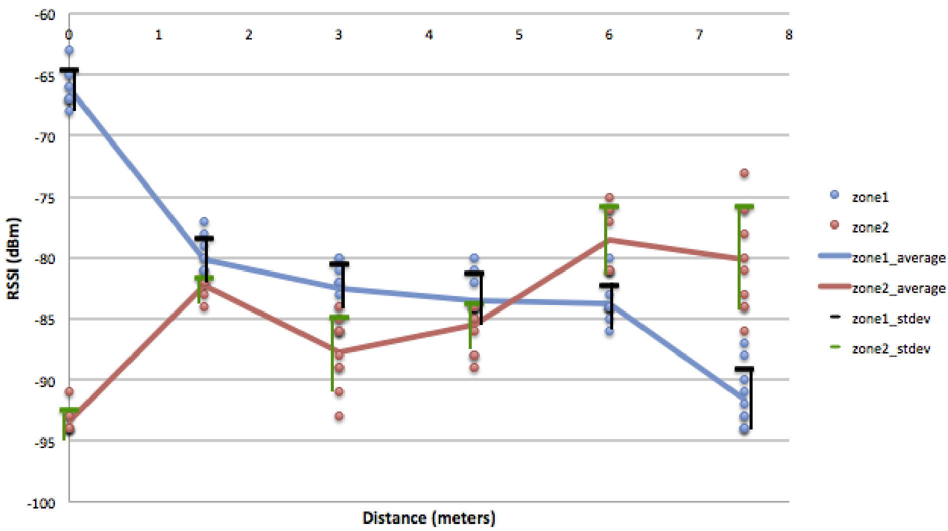

Figure 4.24: System experiment 10 result

The average of standard deviation for the zone 1 equaled 1.90. In the experiment 6 the average was 2.2. As we can see, the standard deviations of these experiments were close. For the zone 2 in this experiment, the average of standard deviations was 2.2. Compared to the experiment 6, where we got 1.30, the results for this experiment were higher, but this was presumably caused by the wall which was between the advertising device and the zone scanner.

The tendency of the blue and the red line seems to be correct according to the distance change. Lines show that, generally, RSSI value is decreasing while moving away from the particular zone and increasing while moving closer towards the zone scanner.

The cross of the blue and the red line appears on the distance of 5m which actually belongs to the region where there is a position change from hall to room 3.

Graph on Figure 4.25, shows the result after applying normalization by formula (4.2).

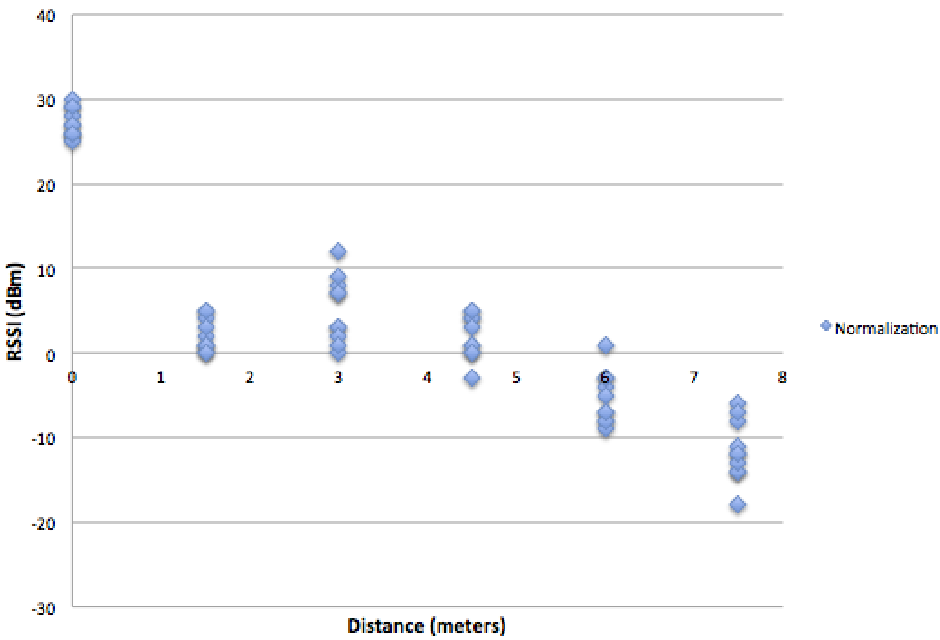

Figure 4.25: System experiment 10 normalization result

From graph it can be seen that on the distance of 0m, all RSSI values are positive which points out that the object was located in the zone 1. On the distances of 1.5, 3 and 4.5 meters, the majority of RSSI values re also positive numbers since the advertising device was still in the zone 1. From the distance 6 and above, the majority of RSSI values are negative numbers, which shows that the device was in the zone 2. Also it can be seen that the change between the two zones is in an area where the hall is connected to the room 3. This can be used for the zone cross detection. It was expected that in the distances of 1.5, 3 and 4.5m all values will be positive, however, because of some unexpected values some of the values were negative.

From the results obtained in this experiment it can be assumed that a wall in between the signal has an effect on the signal transmission and increases the spread of the RSSI values. The detection of the zone after normalization process is possible and also it is possible to detect the cross of the zones.

4.11 Experiment 11

The evaluation of the experiment 11 was performed in the same environment as experiment 10, i.e. with the room 1 and the hall as zone 1 and room 3 as zone 2. For the process evaluation, measurements were carried out every 1.5 meter, including the starting position marked as 0m. Measurements were obtained during walking from zone 1 to zone 2 with the speed 1.4 meters per second. 10 attempts of the experiment were performed with a duration of 1 minute each.

The goal of the experiment was to find out how the RSSI values will change when the object will move. The experiment covered only a movement with an average walking speed 1.4 m/s. Results are presented on the Figure 4.26.

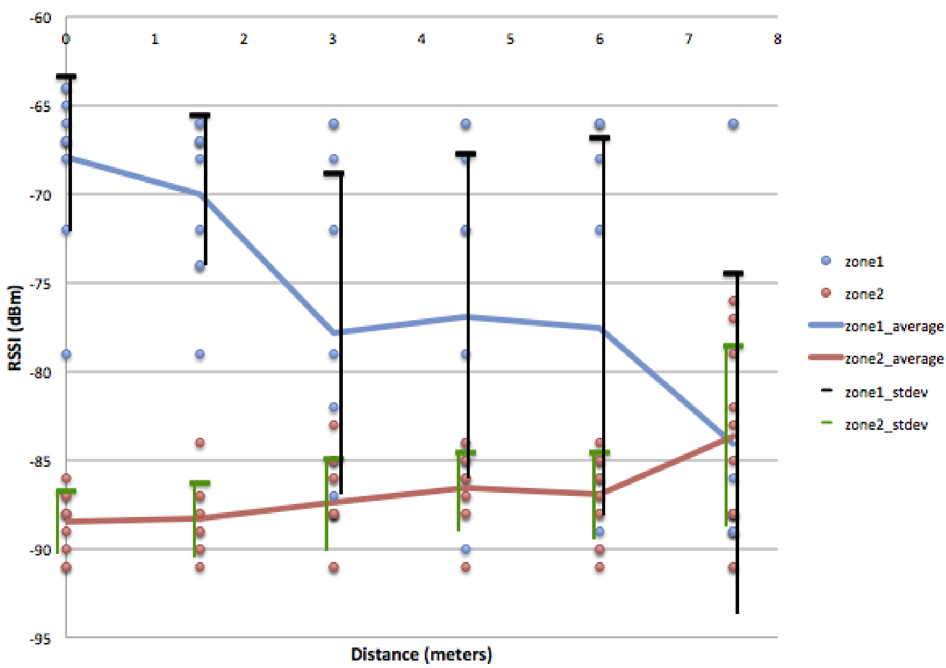

Figure 4.26: System experiment 11 result

For the zone 1, the average of the standard deviations was 7.92. For the zone 2, the average of the standard deviations was 2.62. Throughout this experiment, zone 1 had large values for the standard deviations which cannot be explained.

The tendency of the blue and the red line seems correct according to the distance change. Lines show that, generally, RSSI value is decreasing while moving away from the particular zone and increasing while moving closer towards the zone scanner. But, the cross of the lines appears on the distance 7.7m instead of 5m, which is caused by errors during measurement in the zone 1.

Graph on Figure 4.27 shows the result after applying normalization by formula (4.2).

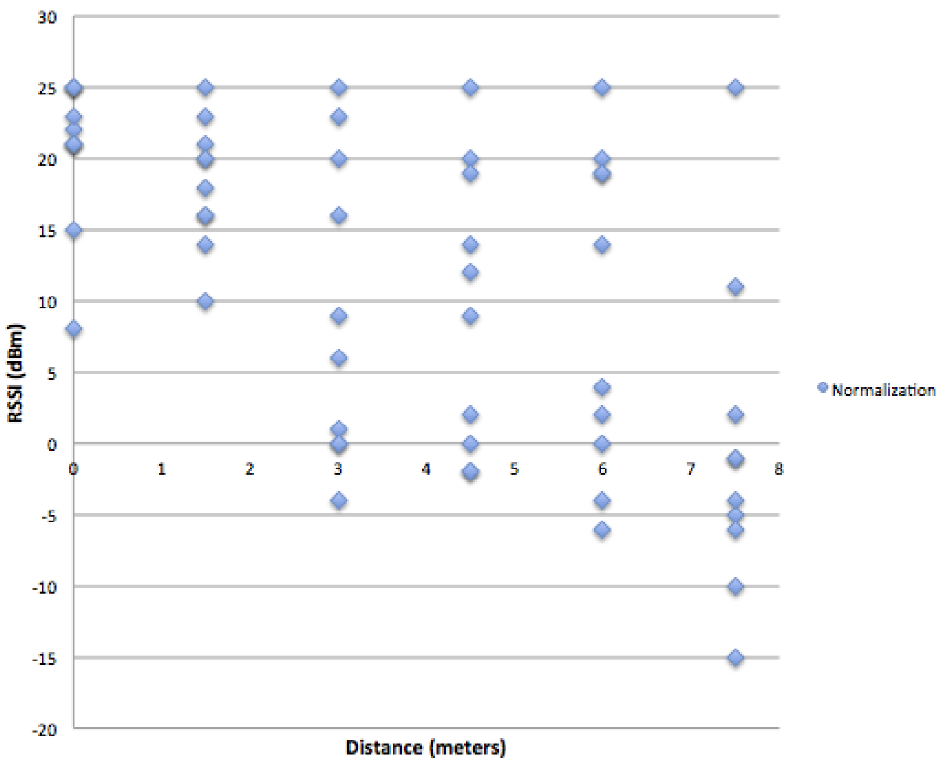

Figure 4.27: System experiment 11 normalization result

On the graph we can see that the spreads of the normalization values are bigger comparing with the previous experiment, experiment 10. On the distances between 0 and 6 meters, the RSSI values are mostly positive which means that the device was located in the zone 1. On the distance of 7.5 meters, most of the values are negative which means that the device was in the zone 2. Even with a large spread, it was possible to make a decision in which zone the advertising object was present.

This experiment shows that a movement of the object increases the spread of the RSSI values on the same distance but the results of normalization give an opportunity to detect in which zone the object was located.

4.12 Experiment's conclusion

The experiments were made with different parameters, including different environment configuration with the wall or without, different network configuration with Ethernet or Wi-Fi connection, different intervals of scanning and different advertising devices. More experiments with other parameter configurations could have been done, but it is assumed that the parameters, which were used here, provid a good experiment basis and the results can be used for the zone detection algorithm development. Further, new experiments with a working detection algorithm will be provided.

Throughout the experiments it was found that it is difficult to detect the zone of a random device correctly, mainly because of a different transmission power of the signal. Therefore, there needs to be separation within the detection of the system internal devices such as Nordic beacon and random devices. Sending interval parameter does not have an impact on the measurement of parameters while the object is standing but while walking it can be reason of incorrect measurement of the zones' crossing place.

Using Wi-Fi as a network connection for the zone scanners has an influence on the signal measurements and the increasing standard deviation of the RSSI values, but the zone detection is still possible. Consequently it is possible to use Wi-Fi in the zones without Ethernet network.

The standard deviation of the RSSI values also depends on the speed of the object and the presence of the wall between the advertising device and the zone scanner.

Normalization process provided by formula (4.2) can be used for the zone detection. The result of the formula depends on RSSI of two zones, which is making the correction of the result during the error measurements on one zone.

During the detection algorithm development we need to take in account that RSSI values can jump, and also the value of normalization can be triggering from positive to negative, and other way round, around zero.

Chapter 5
Zone detection algorithm

In this chapter, I am describing the algorithm for the zone detection, which is located on the central server presented in the Chapter 3 in Section 3.3. Results and conclusions from the experiment Chapter 4 became a base for the algorithm development process.

Three general tasks that need to be solved by algorithm:

1. Device classification (internal and guest devices);
2. Problem with “jumping” of RSSI values;
3. Using the normalization process for establishing the zone.

The general approach of the algorithm is presented on the Figure 5.1. Algorithm was split into 3 modules. The first module, 'Sorting data', includes an algorithm for device classification of the guest or internal device. Internal devices are devices, which are specially developed and configured for the system in order to have best quality of the location detection and have unique identification, which allows the system to recognize them. Guest devices are devices, which are not specially developed for the system and the detection of the location of such devices can be incorrect. For example iPad, which was used in the experiment can be count for a guest device. Second module, 'Preparing data', prepares data before any calculation. Here, an algorithm for solving problem with the jumping of the RSSI values is included. Third module includes logic for the zone detection. The approach with three modules was chosen in order to make the improvement of the algorithm easier in the future. So if there would be other experiments with another type of algorithms, it is necessary only to make changes in a specific module, instead of changing the whole detection algorithm.

5.1 Sorting data module

The sorting data module includes an algorithm, which is making classification for the advertising devices. Devices can be divided into two groups: guest and internal.

Figure 5.1: The general approach of the algorithm

Guest devices are devices, which are unknown for the system, and the results of the zone detection can be random. Internal devices are devices that were specially developed for the system with a known transmission power and a special configuration, which makes this device recognizable by the system.

The information provided in the advertising packet of the BLE signal was chosen for the device classification. In the Chapter 3 Section 3.4, a special protocol 'bluelocation' was developed, which was integrated into the advertising data of the advertising packet. Protocol is presented on the Table 5.1 and consists of the protocol type field, the UUID field and the transmission power field.

Table 5.1: Bluelocation protocol structure

14bytes		
3 bytes	10 bytes	1 byte
TYPE	UUID	TX POWER

The protocol type field can be used for the detection of 'bluelocation' protocol and also for the device classification. The UUID field can be used for the device identification inside the system. Next is the principle of the classification algorithm. After the central server received the array with the scanned devices from the Raspberries Pi, it looked for each device in the array and detected if the "bluelocation" protocol is presented in the advertising data. If so, the device was classified as an internal device. If the protocol was not present, device was recognized as a guest device.

5.2 Preparing data module

This module is preparing the RSSI values of the detected devices for the zone detection. RSSI values of the detected devices are sequences of values. The jumping of the RSSI values can be presented in the sequence as impulses. To prevent these impulses in a Digital Signal Processing (DSP), smoothing filters can be used. Smoothing filters in general are based on shift and multiply technique. The simplest smoothing filter can be presented as a replacement of value in a sequence by an average value of the

N neighbor points. Formula (5.1) shows the example of the filter using 5 neighbor points. S_j represent the smoothed value in the point j and $Y_(j-2)$, $Y_(j-1)$, Y_j, $Y_(j+1)$, $Y_(j+2)$ are neighbor points in a sequence.

$$S_j = \frac{(Y_(j-2) + Y_(j-1) + Y_j + Y_(j+1) + Y_(j+2))}{5} \tag{5.1}$$

Different filters have different advantages and disadvantages. In this project, the Savitzky-Golay filter described in the article [Sch11] was used. This filter is less effective in reducing the impulses in a sequence, but more effective in saving the original values in a sequence. The filter is using a low-degree polynomial approach for finding of a smoothed value.

For the project, 5-point quadratic polynomial with pre-calculated coefficients was used [PAG90], and it is presented in the formula (5.2).

$$S_j = \frac{(-3 * Y_(j-2) + 12 * Y_(j-1) + 17 * Y_j + 12 * Y_(j+1) - 3 * Y_(j+2))}{35} \tag{5.2}$$

S_j - smoothed value in the point j. $Y_(j-2)$, $Y_(j-1)$, Y_j, $Y_(j+1)$ and $Y_(j+2)$ are neighbor points. The results of this formula, which were used on a set of RSSI data obtained during experiments, are presented on Figure 5.2. Measurements were obtained while walking away from the scanned zone with a speed 1 km/h and the measurement period 250 ms. The total distance was 10 meters. Nordic beacon was used as the advertising device during the measurements.

In Figure 5.2, the blue line represents the real data, which shows the changing of RSSI value with the time. Red line simulates data after applying smoothing filter by formula (5.2). Impulse on the blue line can be observed on a 12 second interval. By impulse, it is meant a situation, where RSSI value is unexpectedly growing. On the other hand, this impulse on the red line is smoothed. Also, it can be seen that the red line is shifted to the right, on 1 second interval, compared to the blue line. This could be caused by the fact that the algorithm is using 5 previous measurements and consequently as a result of the smoothing filter. This is a disadvantage of using such algorithm. Smoothing can improve the sequence by smoothing the impulses but on the other hand it is adding a delay in the system. Before judging if this delay is admissible for the system, experiments have to be carried out.

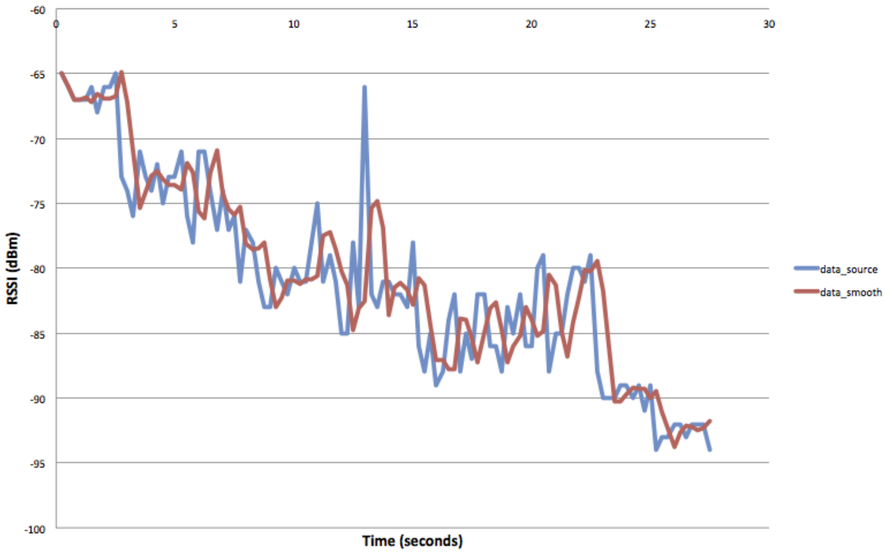

Figure 5.2: Example of Savitzky-Golay filter

5.3 Processing data module

The main task of the processing data module is to calculate in which zone the advertising device is present at the current moment according to the RSSI values. Detection algorithm presented on the Figure 5.3 shows the process of zone establishment.

First thing to do is to look if the advertising device was detected at least in two zones. If not, then we need to check if the device was detected at least by one zone. If the device was not detected even in one zone, the zone detection can be finished and an unknown status for the device zone would be set. If the advertising device was detected at least in one zone, the zone can be assigned for the device. If the device was detected in more than one zone then the two zones with the highest RSSI values are chosen and the normalization process for the values using (5.3) is applied. *Zone1_RSSI* is a zone with the highest RSSI value for advertising device, *normalized_value* is >= 0.

$$normalized\_value = Zone1\_RSSI - Zone2\_RSSI \tag{5.3}$$

NeutralZone parameter helps to prevent triggering between zones after normalization process. While the normalization value is less or equal to the NeutralZone parameter, the detection algorithm, as a result, is using the old zone, which was assigned before. If the normalization value is bigger than NeutralZone parameter

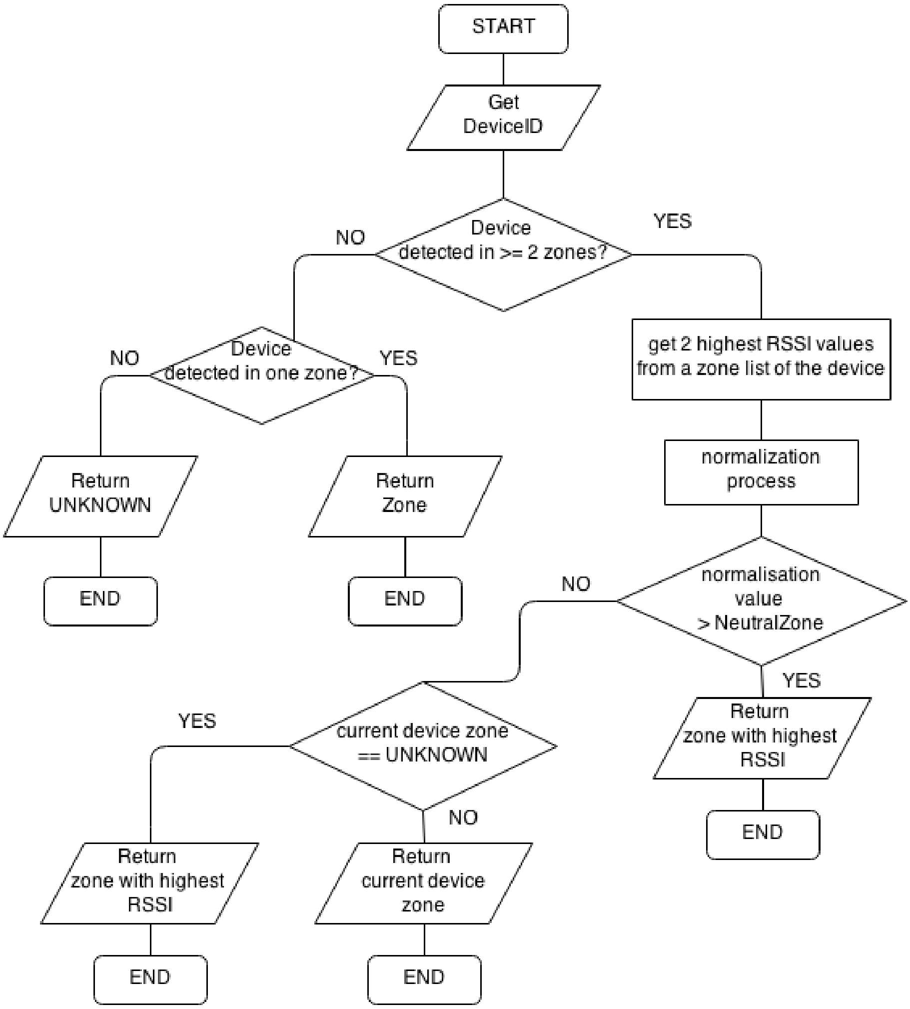

Figure 5.3: Algorithm block scheme

then the algorithm assign a new zone to the advertising device if this zone was not assigned before.

An example of how the NeutralZone parameter works is presented on the Figure 5.4. Graph represents the normalization process from the experiment 8. In this experiment, Nordic beacon was used as an advertising device, and the measurements were taken while walking from zone 1 to zone 2. The distance between scanning stations was 10 meters. NeutralZone area is presented between the red lines. In the example I was using NeutralZone parameter equaled to 5dBm.

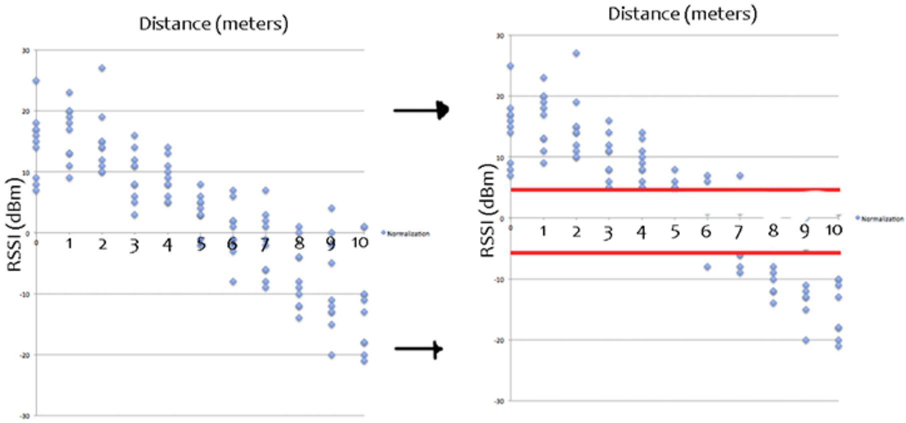

Figure 5.4: Example of working Neutral zone

The left graph represents the real data and the right graph represents how the NeutralZone parameter works. Consequently all values, which were in NeutralZone area, were discarded by the system. There are 6 places on the left graph where the triggering between the zones could occur. It was on the distances 5, 6, 7, 8, 9 and 10 meters. On these distances, both positive and negative results of the normalization process were recorded, which makes it harder to establish a zone in which the object was present. On the right graph, there are only 2 such places, on the distances of 6 and 7 meters. NeutralZone parameter can be configured and need to be estimate by experiments. In addition, NeutralZone parameter depends on the environment. In the future, the system can use a default NeutralZone parameter, which equals 5dBm or another, which can be configured for each pair of zones by experiments.

5.4 Conclusion

The techniques used in algorithms for solving the tasks introduced in the beginning of the chapter such as: a device classification, the RSSI value jumping and a zone detection triggering were used as basis for the zone detection algorithm. Experiments need to show if this design is applicable for the zone detection or not.

Chapter 6

Experiments of the algorithm

This chapter describes a series of experiments that aims to evaluate the developed location detection algorithm, which was described in chapter Chapter 5. As an input, the algorithm takes the latest RSSI value from two scanning stations for the same advertising device, and as result it returns the id of the scanning station to which the advertising device was determined to be the closest. Experiments need to show if it is possible to establish the location of the advertising device using RSSI values from several scanning stations.

The area in which the experiments were performed is presented in the Figure 6.1. Four scanning stations were used in the experiments. These were installed in room 1, 2, 3 and 4. There was no scanning station in a 'hall', but since the 'hall' has a wide connection with a 'room1', in the experiments it is considered as a continuation of the 'room1'. The Nordic beacon was used as an advertising device. Zones 'hall', 'room1' and 'room2' were imaginarily divided into small blocks. For example, the 'room1' consist of blocks 1, 2, 3 and 4, the 'room 2' consist of blocks 1 and 2 and the 'hall' consist of blocks 1 and 2. All measurements in all experiments were made in the middle of the blocks. Each experiment was carried out 10 times.

The independent variables for the experiment are: the speed of movement of the advertising device and the amount of advertising devices in the experimental area. The result of the algorithm, which shows in which zone the advertising devices is present, is a dependent variable, which is later observed directly in the experiment.

The configuration overview for the experiments is presented in the Table 6.1. The table shows the configuration of each of the four experiments. Column 'speed' shows the speed of observed advertising device and column 'BLE devices' shows the amount of the BLE devices present in the experiment area. Throughout the experiments only one device was moving. The location of the other devices was static in the room 1.

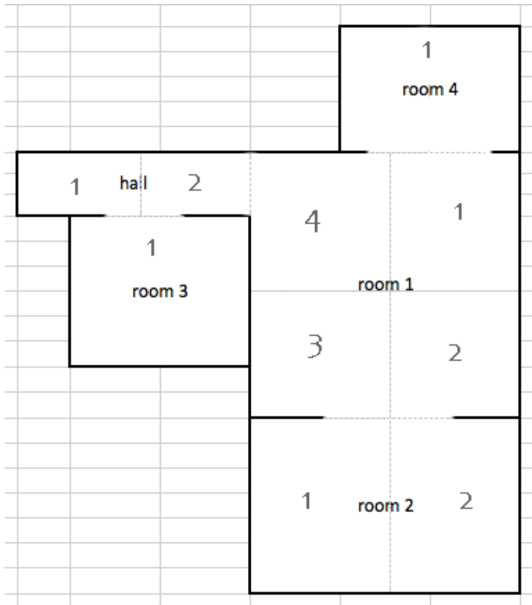

Figure 6.1: Area for experiments of the location detection algorithm

Table 6.1: Configuration of experiments for location detection algorithm

Experiment Number	*Independent variables*	
	speed (m/s)	*BLE devices*
1	0	1
2	1.4	1
3	0	3
4	1.4	3

6.1 Experiment 1

Observations of experiment 1 were made while standing in the middle of the blocks presented on the Figure 6.1. 10 tries for each block were carried out and only one device at the time was present in the area of experiment.

The hypothesis that it is possible to detect the location of the object while standing using the developed algorithm was tested here.

The results obtained in the experiment 1 are presented in the Table 6.2.

Column 'Room Number' shows, in which room the measurements were made. Column 'Block number' shows in which block of the room the measurements were

Table 6.2: Results of experiment 1 of the algorithm

Room Number	Block Number	Amount of detections for rooms				Expected room	Detection Errors (%)	Percentage Success (%)
		1	2	3	4			
1	1	8	0	0	2	1	20	80
	2	8	2	0	0	1	20	80
	3	10	0	0	0	1	0	100
	4	10	0	0	0	1	0	100
2	1	0	10	0	0	2	0	100
	2	1	9	0	0	2	10	90
hall	1	0	0	10	0	3	0	100
	2	10	0	0	0	1	0	100
3	1	0	0	10	0	3	0	100
4	1	0	0	0	10	4	0	100

made. Columns 'Amount of detection for rooms' shows how many times the algorithms detected that the object is either in room 1, 2, 3 or 4. Column 'Expected room' shows a result of the algorithm, which was expected. There was no separate scanning station in the 'hall'. Therefore, it was expected that block 1 of the 'hall' would have 'room3' as a result and block 2 of the 'hall' would have 'room1' as a result. These expectations were according to the nearest scanning station. Column 'Detection Errors(%)' shows the percentage of unexpected results out of all tries. Column 'Percentage Success(%)' shows the percentage of expected results out of all tries.

Seven of the results showed a correct detection, this is proved by 10 successfully obtained results out of 10 expected results. However, in three locations (room 1 block 1, room 1 block 2 and room 2 block 1) unexpected results were obtained. In the room 1 block 1, the algorithm showed, two times, that an object was present in the room 4. This could happen because of the fact that the room 4 is close to the room 1 block 1. In the room 1 block 2, the algorithm showed, also two times, that an object was present in the room 2. This result could happen because of the fact that the room 2 is close to the room 1 block 2. In the room 2 block 2, the algorithm showed, only one time, that an object was present in the room 1. This could be because of the fact that the room 1 is close to the room 2 block 2.

Experiment showed the possibility to detect a location of an object using the algorithm developed in the Chapter 5 while standing. However, unexpected results appeared in the blocks, which were near to the other rooms.

6.2 Experiment 2

Observations of experiment 2 were made while moving. The path of the movement is presented on the Figure 6.2 and is represented by the green arrows. The path of movement follows as: room 1 block 1 -> room1 block 2 -> room 2 block 2 -> room

2 block 1 -> room 1 block 3 -> room 1 block 4 -> hall block 2 -> room 3 -> hall block 1 -> hall block 2 -> room 1 block 4 ->room 4.

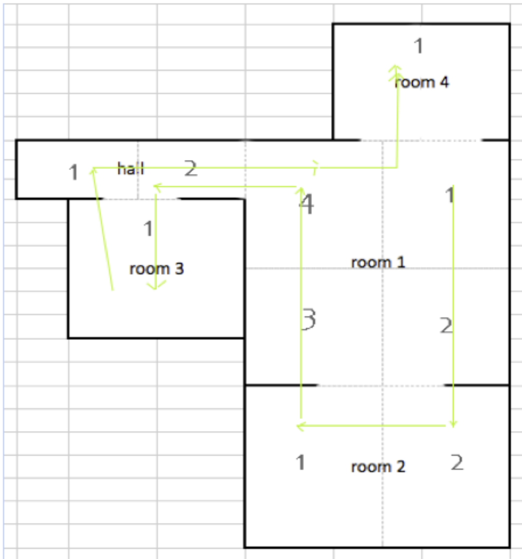

Figure 6.2: Path of the movement for the experiments of algorithm

The detection of a zone by algorithm was made in the middle of each block. 10 tries for each block were carried out and only one device at the time was present in the area of experiment.

A hypothesis, which was tested in this experiment states, that during a movement the amount of expected results in the blocks will be lower comparing with the experiment 1.

The results of the experiments are presented on the Table 6.3.

The final results showed that the expected results were obtained only in three cases, in room 1 block 1, room 1 block 4 and room 4 block 1. 90 % of expected results were obtained in room 1 block 2, room 2 block 1, hall block 2, room 3 block 1 and hall block 1. The lowest percentage of expected results were obtained in the hall block 2. The final values show that an object was detected in the room 1 six times or in the room 3 four times. This could have been caused due to a lack of a scanning station in the hall, and the fact that the hall is between room 1 and room 3. From the table, it can be seen that the percentage of successful results decreases within the change of a location between blocks from different rooms, but increases within a change of blocks in the same room. For example, when changing a location from room 1 block 2 to room 2 block 2, the percentage of successful results decreased

Table 6.3: Results of experiment 2 of the algorithm

Room Number	Block Number	Amount of detections for rooms				Expected room	Detection Errors (%)	Percentage Success (%)
		1	2	3	4			
1	1	10	0	0	0	1	0	100
1	2	9	1	0	0	1	10	90
2	2	3	7	0	0	2	30	70
2	1	1	9	0	0	2	10	90
1	3	8	2	0	0	1	20	80
1	4	10	0	0	0	1	0	100
hall	2	9	0	1	0	1	10	90
3	1	1	0	9	0	3	10	90
hall	1	1	0	9	0	3	10	90
hall	2	6	0	4	0	1	40	60
1	4	8	0	1	1	1	20	80
4	1	0	0	0	10	4	0	100

from 90% to 70%. When changing a location from room 2 block 1 to room 1 block 3, the percentage of successful results decreased from 90% to 80%. However, when changing a location from room 2 block 2 to room 2 block 1, the percentage increased from 70% to 90%. The increasing tendency was also spotted when a location changed from room 1 block 3 to room 1 block 4. A decrease in a percentage of the successful results while changing the location between rooms can be explained by the fact, that after the location was changed from one room to another, the algorithm was still showing a result from previous room.

The experiment showed, that the percentage of successful results of location detection is lower when it includes movement compared to the experiment 1 where the measurements were made while standing. The low percentage of successful results was caused by the fact that when object changed the position from one room to another, the system was still showing a result from the previous room.

6.3 Experiment 3

Previous experiments were assessed with only one advertising device in the experimental area. In this experiment, 2 more advertising devices are added. The purpose of such innovation is to test if several advertising devices present in the same area will increase the amount of error results of the location detection algorithm.

Observations of this experiment were made while standing in the middle of the blocks presented on the Figure 6.1. 10 tries for each block were carried out and three advertising devices were present in the area of experiment at the time. Only one device was transported and its location was observed. Other 2 devices were statically located in the room 1.

A hypothesis, which was tested here, states that, increasing the amount of advertising devices in the same experimental area will increase the amount of unexpected location detections while standing.

The results of the experiments are presented on the Table 6.4.

Table 6.4: Results of experiment 3 of the algorithm

Room Number	Block Number	Amount of detections for rooms				Expected room	Detection Errors (%)	Percentage Success (%)
		1	2	3	4			
1	1	8	0	0	2	1	20	80
	2	7	3	0	0	1	30	70
	3	8	0	0	2	1	20	80
	4	7	0	1	2	1	30	70
2	1	2	8	0	0	2	20	80
	2	1	9	0	0	2	10	90
hall	1	2	0	8	0	3	20	80
	2	8	0	1	1	1	20	80
3	1	0	0	10	0	3	0	100
4	1	1	0	0	9	4	10	90

The minimal percentage of successful results compared to expected results was 70% in this experiment. Comparing with experiment 1, where only one advertising device was used in the experimental area, the percentage of the successful results in this experiment was in general lower. Also, in a room 1 block 4, the object was once detected in the room 3.

The experiment showed that the amount of advertising devices present in one room, influences the quality of algorithm detection. Critically speaking, it decreases its quality.

6.4 Experiment 4

This experiment, like experiment 2, was observed during movement, but the amount of advertising devices present in the same area has been increased. Moving path is presented on the Figure 6.2. Green arrows represent the path of the movement. The experimental path follows as: room 1 block 1 -> room1 block 2 -> room 2 block 2 -> room 2 block 1 -> room 1 block 3 -> room 1 block 4 -> hall block 2 -> room 3 -> hall block 1 -> hall block 2 -> room 1 block 4 ->room 4.

The detection of a zone by algorithm was made in the middle of each block. 10 tries for each block were carried out and three advertising devices were present in the area of experiment at the time.

A hypothesis, which was tested here states that, increasing the amount of advertising devices in the same experimental area will increase the amount of error results of the location detection algorithm while moving.

The results of the experiment are presented on the Table 6.5.

Table 6.5: Results of experiment 4 of the algorithm

Room Number	Block Number	Amount of detections for rooms				Expected room	Detection Errors (%)	Percentage Success (%)
		1	2	3	4			
1	1	9	0	0	1	1	10	90
1	2	9	1	0	0	1	10	90
2	2	4	6	0	0	2	40	60
2	1	3	7	0	0	2	30	70
1	3	6	2	0	2	1	40	60
1	4	7	2	0	1	1	30	70
hall	2	6	0	3	1	1	40	60
3	1	2	0	8	0	3	20	80
hall	1	1	0	9	0	3	10	90
hall	2	6	0	4	0	1	40	60
1	4	7	0	1	2	1	30	70
4	1	4	0	0	6	4	40	60

In general, results showed that the percentage of successful detections decreased compared to the experiment 2. In the room 1 block 3, two detections were recorded as room 4, which is actually odd since room 1 block 3 and room 4 are not neighboring zones. On the other hand, it is possible to see the scanning station of room 4 from the room 1 block 3, since the entrance to the room 4 from room 1 is wide and without a door. So this could be a reason why such result was obtained.

The results of experiment 4 showed that the amount of the successful results of detection algorithm decreased after increasing the amount of advertising devices in the experimental area. However, it is important to realize that even when detection was not correct, the algorithm detected only the neighboring room but did not ”jump” to totally random room.

6.5 Conclusion

The four experiments showed the possibility of detecting the location, using algorithm described in the Chapter 5.

Experiments that were using one advertising device in the experimental area showed better results of detection compared to the experiments where three advertising devices were used in the experimental area, this could be caused by the interference. However, the unsuccessful detections were usually detections of a neighboring room to the room where object was actually present.

Additionally, when moving from one room to another, the detection algorithm showed that the object was still present in the room from where the advertising device was coming.

Chapter 7 Discussion

System design. What are the positive and negative sides of the design?

Using the Raspberry Pi for the project as a zone scanner allowed to make a small-size and low-cost solution. The next advantage is that on Raspberry Pi it is possible to run Rasberian linux system with a native support of Java, which makes the application developed by the Reactve Blocks SDK possible to be run. Personally, I find Raspberry Pi to be a good system, which can be running continuously 24/7. However, throughout the project development some problems with Wi-Fi and erasing SD card while switching off from the power were found, but these problems can be solved.

In this project, BLE was used instead of classical Bluetooth. However, an example of system where classic Bluetooth was used, was introduced in the background chapter in section Related work. A problem of that work was a big latency of device, around 10 seconds. In this project the latency was only 250ms. The great latency of the related work presented, may have been caused by using a normal phone with limited interval as an advertising device.

In this work, a centralized system is used to determine the location of a device. This approach is good for situations where it is necessary to track objects by system. Since all calculations are on the central server, the RTLS tag can be a simple device that is just advertising a signal. This allows to downgrade the cost of the RTLS tag.

Savitzky-Golay filter was used for the data processing in order to smooth the RSSI. On one hand, the filter helped to prevent impulses, but on the other hand, it was adding delay to the zone detection. Also, it might be a good decision to try another types of filters and make the experiments with them.

Throughout the experiments, it was found that RSSI values of one device on different zones are dependent. This observation was used in a zone detection algorithm, where the RSSI from two zone scanners was used in order to establish the zone where the object was present. There is a possibility that using RSSI values from three zone scanners could provide a better zone detection, however, this hypothesis would need to be proved or disproved in a set of new experiments.

One disadvantage of the algorithm at the moment is that, the algorithm is not able to say that ‘the location cannot be determined’ and it always detects a room as long as a zone scanner is able to detect the beacon.

Experiment results. Is it possible to use developed system for detecting the location of the object?

The experiments in which three advertising devices were present in one area, did not show a limitation on how many more advertising devices can be present in one area without completely disturbing the location detection. The larger amount of advertising devices can cause wrong location detection because of signal interference. So, in future, it is necessary to make new series of experiments where more advertising devices will be used, in order to establish the dependence of a quality of a zone detection on the amount of advertising devices.

The algorithm is not able to say that ‘the location cannot be determined’ but instead, it is giving the neighboring room as a result. Consequently, there comes a conclusion that it is not possible to use this system in places, where a wrong zone detection can cause some damages. However, this system can be used to estimate an approximate position of an object or a person, for example at university in a way that students can find approximately where a professor is located.

Chapter 8
Conclusion

In conclusion to the done work, I want to return back to the goals of the research, which were introduced in the Chapter 1, and show how these goals was achieved.

Software architecture:

The goal: To show the software architecture design of a Real-time locating system, which is using the BLE technology for establishing location of the objects.

The software for the Real-time locating system was developed using the Reactive Block SDK. System consists of zone scanners and a central station. Zone scanners are nodes, which are installed in different zones of the building and are receiving the advertising signal. Likewise, zone scanners are measuring the RSSI values of received signals and send this information to a central server through MQTT protocol. The central server is further calculating the location of the advertising devices. Also, it has a communication interface that is providing a possibility to exchange the location information with other systems through Rest API.

Experiments results:

The goal: To provide series of experiments which will help to establish the quality of detection of the object location using the BLE technology and also to show its limitations.

Experiments that were measuring RSSI values with different types of advertising devices and were carried out in different environment are shown in the Chapter 4. Based on the results of these experiments, the algorithm for detecting the location of the object was developed in Chapter 5. The final experiments in Chapter 6 showed the results of the algorithms. Finally, it was found that the increasing amount of the advertising devices will decrease the quality of the location detection. However, there will still be some kind of precision, that is, the algorithm will detect at least the neighboring room to the room in which the advertising device is actually present. There were 44 different situations used in the experiments, including different amount of advertising devices and different speed of advertising object. The minimal percentage of the successful detections was 60%, noted in 6 situations. 70% of successful detections were noted in 6 situations, 80% in 10 situations, 90% in 11 situations and 100% in 11 situations.

Chapter 9
Future work

I think that the developed system provided good results and can be possibly used as a basis for Real-time locating system. However some further work can be done in order to improve the results.

In my opinion, a future work needs to be concentrated firstly on fixing problems with the hardware:

- To try another Bluetooth modules. In this experiment, Bluetooth modules produced by company Targus were used, which were getting problems during the scanning mode and were becoming frozen. As a solution, it will be good to look into Nordic evaluation kit and build a USB Bluetooth module, using this platform. With a S120 SoftDevice stack, it is possible to develop Bluetooth device in a central role. I will mark this part of future work as the most important because the quality of the Bluetooth module will depend on quality of the locating system;
- Problems with a Wi-Fi connection. To solve these problems, other drivers or another Wi-Fi modules of different companies need to be tried;
- Problem with corrupting the SD card during the switching off. Since the file system corruption was happening while writing something on SD card, it is necessary to switch off Raspberry Pi by special button, which can be connected to the GPIO port and can be used to switch off the system programmatically.

References

[Ama14] Amazon. Amazon web services, http://aws.amazon.com/products/, 2014.

[Art13] Pavel Arteev. An application for coarse indoor positioning of users using bluetooth. Technical report, Norwegian University of Science and Technology, Department of Telematics, 2013.

[AS14a] Bitreactive AS. Reactive blocks, http://www.bitreactive.com, 2014.

[AS14b] Bitreactive AS. Types of blocks, http://reference.bitreactive.com/reference/types-of-blocks.html, 2014.

[Fie00] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures, Chapter 5, Representational State Transfer (REST)*. PhD thesis, University of California, Irvine, 2000.

[fS08] International Organization for Standardization. Iso/iec 19762-5, 2008.

[HH09] Simon Hay and Robert Harle. Bluetooth tracking without discoverability. In Tanzeem Choudhury, Aaron Quigley, Thomas Strang, and Koji Suginuma, editors, *Location and Context Awareness*, volume 5561 of *Lecture Notes in Computer Science*, pages 120–137. Springer Berlin Heidelberg, 2009.

[8] International Business Machines Corporation (IBM). *MQTT V3.1 Protocol Specification*, 2010.

[Int13] Ecma International. *The JSON Data Interchange Format*, October 2013.

[Lan09] Andreas Dypvik Landmark. Capability assessment of indoor positioning systems. Technical report, Norwegian University of Science and Technology, Department of Computer and Information Science, 2009.

[Mos14] Mosquitto. An open source mqtt v3.1/v3.1.1 broker, http://mosquitto.org/documentation/, 2014.

[Ora14] Oracle. Java doc, http://docs.oracle.com/javase/7/docs/api/java/util/hashmap.html, 2014.

[PAG90] University of Manchester Manchester England M13 9PL Peter A. Gorry, Department of Chemistry. General least-squares smoothing and differentiation by the convolution (savitzky-golay) method. *ANALYTICAL CHEMISTRY*, 62:570–573, MARCH 15, 1990.

[PMoBS13a] Inc. Promoter Members of Bluetooth SIG. *Bluetooth Specification Version 4.1, Architecture and Terminology Overview, ch. 1 General description, p. 17*, December 2013.

[PMoBS13b] Inc. Promoter Members of Bluetooth SIG. *Bluetooth Specification Version 4.1, Architecture and Terminology Overview, ch. 1.4 Nomenclature, p.24*, December 2013.

[PMoBS13c] Inc. Promoter Members of Bluetooth SIG. *Bluetooth Specification Version 4.1, Architecture and Terminology Overview, ch. 6.2 Generic Access Profile, p. 100*, December 2013.

[PMoBS13d] Inc. Promoter Members of Bluetooth SIG. *Bluetooth Specification Version 4.1, Host Controller Interface Functional Specification, ch. 7.7.65 LE Meta Event, p. 935*, December 2013.

[PMoBS13e] Inc. Promoter Members of Bluetooth SIG. *Bluetooth Specification Version 4.1, Host Controller Interface Functional Specification, ch. 7.7.65.2 LE Advertising Report Event, p. 938*, December 2013.

[PMoBS13f] Inc. Promoter Members of Bluetooth SIG. *Bluetooth Specification Version 4.1, Host Controller Interface Functional Specification, ch. 7.7.65.2 LE Advertising Report Event, p. 939*, December 2013.

[Sch11] R.W. Schafer. What is a savitzky-golay filter? *Signal Processing Magazine*, 28:111–117, 16 June 2011.

[Sem13] Nordic Semiconductor. *nRF51822, Product Specification v2.0*, 2013.

[sta14] Dictionary.com unabridged. Jun 2014.

[Tar14] Targus. *Targus Bluetooth® 4.0 Micro USB Adapter for Laptops*, 2014.

[Wik] Wikipedia. Cross-platform, http://en.wikipedia.org/wiki/cross-platform.

Appendix A

USB reset

Algorithm A.1 USB reset code on linux.

```
#include <errno.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>

#include <linux/usbdevice_fs.h>

int main(int argc, char **argv)
{
    const char *usbtoreset;
    int fd;
    int rc;

    if (argc != 2)
    {
        return 1;
    }

    usbtoreset = argv[1];

    // Trying to open description
    fd = open(usbtoreset, O_WRONLY);
    // get en error on opening
    if (fd < 0)
    {
        return 1;
    }
```

```
    if(ioctl(fd, USBDEVFS_RESET, 0)<0)
    {
        return 1;
    };

    close(fd);
    return 0;
}
```

Appendix B

REST API server

Algorithm B.1 REST API server initialization

```
public void startRestAPIserver() {
    try {
        SocketServer server;
        server = new SocketServer(SERVER_PORT, new IHttpRequestListener(){

        @Override
        public HttpResponse receiveRequest(HttpRequest req) {
            if (req == null) {
                System.out.println("Error: NULL request");
                return null;
            }
            req.getHeader();

            String responceAnswer  = httpRouter(req.getUrl().getPath());

            if(responceAnswer!=null){
                HttpResponse resp = Utils.genRespOK();
                resp.setBody(responceAnswer);
                return resp;
            }else{
                HttpResponse resp = Utils.genRespBadReq();
                resp.setBody("Wrong reques");
                return resp;
            }
        }});

        Thread t = new Thread(server);
        t.start();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Algorithm B.2 REST API server router function

```
private String httpRouter(String path){
    if(path == null)
        return null;

    String []values = path.split("/");

    if(values==null)
        return null;
    if(values.length<2)
        return null;

    String value = values[1];
    if(value.equals("api")){
        if(values.length>2){
            value = values[2];
            if(value.equals("users")){
                if(values.length==3){
                    return this.apiUsers();
                }
            }else if(value.equals("user")){
                if(values.length>3){
                    value = values[3];
                    if(values.length==4){
                        return this.apiUserId(value);
                    }
                }
            }else if(value.equals("userZone")){
                if(values.length>3){
                    value = values[3];
                    if(values.length==4){
                        return this.apiUserZoneById(value);
                    }
                }
            }else if(value.equals("zones")){
                if(values.length==3){
                    return this.apiZones();
                }
            }else if(value.equals("zone")){
                if(values.length>3){
                    value = values[3];
                    if(values.length==4){
                        return this.apiZoneId(value);
                    }
                }
            }
        }
    }
return null;
}
```

Algorithm B.3 REST API server function apiUserZoneById

```
private synchronized String apiUserZoneById(String id){
    if(id==null){
        String resultString = "{\"api_status\":\"error\",";
        resultString += "\"api_result\":\"missed ID\"}";
        return resultString;
    }

    String detectedZone = zoneDetection.getZoneForDevice(id);

    String resultString = "{\"api_status\":\"ok\",";
    resultString += "\"api_result\":";

    resultString += "{\"user_id\":\""+id+"\"";
    resultString += ",\"zone\":\""+detectedZone+"\"}}";

    return resultString;
}
```

Algorithm B.4 REST API server function apiUsers

```
private synchronized String apiUsers(){
    String resultString = "{\"api_status\":\"ok\",";
    resultString += "\"api_result\":";
    resultString += "{\"users\":[";

    Iterator<Entry<String, Dongle>> it = this.TABLE_DEVICE.\
                                        entrySet().iterator();
    while (it.hasNext()) {
        Map.Entry<String, Dongle> pairs = (Map.Entry<String, \
                                         Dongle>)it.next();

        if(pairs.getValue().getRooms().size()==0)
            continue;

        resultString += "{";

        Dongle user = pairs.getValue();

        resultString += "\"user_id\":\""+user.getUuid()+"\"";
        resultString += ",\"zones\":[";

        Iterator<Entry<String, Map<String, Object>>> innerIterator = \
                    pairs.getValue().getRooms().entrySet().iterator();

        boolean first = true;
        while(innerIterator.hasNext()){
            Map.Entry<String, Map<String, Object>> room = (Map.Entry<String,\
                                 Map<String, Object>>)innerIterator.next();

            long currentTime = System.currentTimeMillis();
            Map<String, Object>value = room.getValue();
        try{
                long messageTime = (long) value.get("time");
                long timeSinceLastUpdate = currentTime - messageTime;
                timeSinceLastUpdate /=1000;

                if(timeSinceLastUpdate < 60){
                    Map<String, Object> zone = room.getValue();
                    String zone_id = (String) zone.get("zone_id");
                    String zone_ip = (String)zone.get("zone_ip");
                    double zone_rssi = (double) zone.get("rssi");

                    if(!first){
                        resultString +=",";
                    }else{
                        first = false;
                    }
```

```
                    resultString += "{";

                    resultString += "\"zone_id\":\"" + zone_id +"\"";
                    resultString += ",\"zone_ip\":\"" + zone_ip +"\"";
                    resultString += ",\"rssi\":" + zone_rssi;
                    resultString += ",\"last_update\":" + timeSinceLastUpdate;
                    resultString += ",\"time\":" + messageTime;
                    resultString += "}";
                }
            }catch(Exception e){}
        }

        resultString += "]";
        resultString += "}";
        if(it.hasNext()){
            resultString += ",";
        }
    }
    resultString += "]}}";

    return resultString;
}
```

Algorithm B.5 REST API server function apiUserId

```
private synchronized String apiUserId(String id){
    if(id==null){
        String resultString = "{\"api_status\":\"error\",";
        resultString += "\"api_result\":\"missed ID\"}";
        return resultString;
    }

    Dongle user = this.TABLE_DEVICE.get(id);
    if(user == null){
        String resultString = "{\"api_status\":\"error\",";
        resultString += "\"api_result\":\"Can not find the user\"}";
        return resultString;
    }

    String resultString = "{\"api_status\":\"ok\",";
    resultString += "\"api_result\":";

    resultString += "{\"user_id\":\""+user.getUuid()+"\"";
    resultString += ",\"zones\":[";

    Iterator<Entry<String, Map<String, Object>>> innerIterator = \
                            user.getRooms().entrySet().iterator();
    boolean first = true;

    ArrayList<HashMap<String, Double>> roomsArray = \
                            new ArrayList<HashMap<String, Double>>();

    while(innerIterator.hasNext()){
        Map.Entry<String, Map<String, Object>> room = (Map.Entry<String, \
                            Map<String, Object>>)innerIterator.next();
        long currentTime = System.currentTimeMillis();
     Map<String, Object>value = room.getValue();
        try{
            long messageTime = (long) value.get("time");
            long timeSinceLastUpdate = currentTime - messageTime;
            timeSinceLastUpdate /=1000;

            if(timeSinceLastUpdate < 60){
                Map<String, Object> zone = room.getValue();
                String zone_id = (String) zone.get("zone_id");
                String zone_ip = (String)zone.get("zone_ip");
                double zone_rssi = (double) zone.get("rssi");

                // Adding room to array list
                HashMap<String, Double> roomToArray = new HashMap<String,\
                                                    Double>();
                roomToArray.put(zone_id, zone_rssi);
                roomsArray.add(roomToArray);
```

```
                if(!first){
                    resultString +=",";
                }else{
                    first = false;
                }

                resultString += "{";
                resultString += "\"zone_id\":\"" + zone_id +"\"";
                resultString += ",\"zone_ip\":\"" + zone_ip +"\"";
                resultString += ",\"rssi\":" + zone_rssi;
                resultString += ",\"last_update\":" + timeSinceLastUpdate;
                resultString += ",\"time\":" + messageTime;

                resultString += "}";

            }
        }catch(Exception e){}
    }

    resultString += "]";
    resultString += "}}";

    return resultString;
}
```

Algorithm B.6 REST API server function apiZones

```
private synchronized String apiZones(){
    String resultString = "{\"api_status\":\"ok\",";
    resultString += "\"api_result\":";

    resultString += "{\"zones\":[";
    Iterator<Entry<String, Room>> it = this.TABLE_ZONE.entrySet().\
                                                          iterator();
    while (it.hasNext()) {
        resultString += "{";
        Map.Entry<String, Room> pairs = (Map.Entry<String, \
                                                      Room>)it.next();
        Room room = pairs.getValue();

        resultString += "\"zone_id\":\""+room.getHw()+"\"";
        resultString += ",\"zone_ip\":\""+room.getIp() + "\"";
        resultString += ",\"users\":[";

        Iterator<Entry<String, Map<String, Object>>> innerIterator = \
                        pairs.getValue().getDevices().entrySet().iterator();

        boolean first = true;
        while(innerIterator.hasNext()){
            Map.Entry<String, Map<String, Object>> dongle = (Map.Entry<String,\
                        Map<String, Object>>)innerIterator.next();

         long currentTime = System.currentTimeMillis();
         Map<String, Object>value = dongle.getValue();
            try{
                long messageTime = (long) value.get("time");
                long timeSinceLastUpdate = currentTime - messageTime;
                timeSinceLastUpdate /=1000;
                if(timeSinceLastUpdate < 60){
                    Map<String, Object> user = dongle.getValue();
                    String user_id = (String) user.get("user_id");
                    double user_rssi = (double) user.get("rssi");

                    if(!first){
                        resultString +=",";
                    }else{
                        first = false;
                    }

                    resultString += "{";
                    resultString += "\"user_id\":\"" + user_id +"\"";
                    resultString += ",\"rssi\":"+user_rssi;
                    resultString += ",\"last_update\":" + timeSinceLastUpdate;
                    resultString += ",\"time\":" + messageTime;
                    resultString += "}";

                }
```

```
            }catch(Exception e){}

        }

        resultString += "]";
        resultString += "}";
        if(it.hasNext()){
            resultString += ",";
        }
    }

    resultString += "]}}";

    return resultString;
}
```

Algorithm B.7 REST API server function apiZoneId

```
private synchronized String apiZoneId(String id){
    if(id==null){
        String resultString = "{\"api_status\":\"error\",";
        resultString += "\"api_result\":\"missed ID\"}";
        return resultString;
    }

    Room room = this.TABLE_ZONE.get(id);

    if(room == null){
        String resultString = "{\"api_status\":\"error\",";
        resultString += "\"api_result\":\"Can not find the zone\"}";
        return resultString;
    }

    String resultString = "{\"api_status\":\"ok\",";
    resultString += "\"api_result\":";
    resultString += "{\"zone_id\":\""+room.getHw()+"\"";
    resultString += ",\"zone_ip\":\""+room.getIp() + "\"";
    resultString += ",\"users\":[";

    Iterator<Entry<String, Map<String, Object>>> innerIterator = \
                        room.getDevices().entrySet().iterator();
    boolean first = true;
    while(innerIterator.hasNext()){
        Map.Entry<String, Map<String, Object>> dongle = (Map.Entry<String,\
                        Map<String, Object>>)innerIterator.next();

        long currentTime = System.currentTimeMillis();
        Map<String, Object>value = dongle.getValue();
        try{
            long messageTime = (long) value.get("time");
            long timeSinceLastUpdate = currentTime - messageTime;
            timeSinceLastUpdate /=1000;
            if(timeSinceLastUpdate < 60){
                Map<String, Object> user = dongle.getValue();
                String user_id = (String) user.get("user_id");
                double user_rssi = (double) user.get("rssi");
```

```
                if(!first){
                    resultString +=",";
                }else{
                    first = false;
                }

                resultString += "{";
                resultString += "\"user_id\":\"" + user_id +"\"";
                resultString += ",\"rssi\":"+user_rssi;
                resultString += ",\"last_update\":" + timeSinceLastUpdate;
                resultString += ",\"time\":" + messageTime;
                resultString += "}";
            }
        }catch(Exception e){}

    }

    resultString += "]}}";
    return resultString;
}
```