INDOOR LOCALIZATION USING BLUETOOTH LOW ENERGY

Submitted by

RAJA VENKATA SATYA PHANINDRA CHAVA

117005103

In partial fulfilment for the award of the degree of

Bachelor of Technology

in

Electrical and Electronics Engineering

School of Electrical and Electronics Engineering

SASTRA University

Tirumalaisamudram

Thanjavur – 613401

June 2017

MISSOURI S&T

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Formerly University of Missouri-Rolla

5/29/2017

Letter of Completion

Name: Raja Venkata Satya Phanindra Chava

Register no: 117005103

Program: B. Tech. Electrical and Electronics Engineering

Title of the Project: Indoor Localization using Bluetooth Low Energy Project
Duration: February 11 to June 1st 2017

Phanindra Chava started his project on indoor localization of assets by using Bluetooth active and passive RFID devices. Since GPS cannot be utilized for indoor applications, active tags can be employed as satellites and an additional tag that will be attached to the asset will process signals from the other active tags to localize the asset. In this project, Phanindra Chava used a technique of using Bluetooth Low Energy (BLE) and Received Signal Strength Indication (RSSI) together for indoor localization. Bluetooth Low Energy (BLE) is pervasively available, is relatively cheap and has relatively low power consumption. Especially the fact that Bluetooth is integrated in a wide range of mobile devices makes its use attractive for indoor localization. Fingerprinting is utilized to collect data apriori and to estimate the location of the asset. This information is utilized to generate a logistic regression function which is subsequently used to estimate the actual location of the asset given the measuresed RSSI values. Phanindra Chava concluded that, despite challenges in fingerprinting and RSSI, BLE can be an alternative to traditional localization techniques.

Sincerely

Jagannathan Sarangapani, Fellow IEEE, Fellow IET, Fellow Inst. of Measu. & Control, UK

Dr. Jagannathan Sarangapani, Rutledge-Emerson Distinguished Professor & Site Director NSF I/UCRC on Intelligent Maintenance Systems

222 Emerson Electric Co. Hall • 301 West 16th Street • Rolla, MO 65409-0040

Phone: 573-341-6775 • Fax: 573-341-4532 • Email: sarangap@mst.edu • Web: www.mst.edu/~sarangap

BONA FIDE CERTIFICATE

Certified that the project work entitled, **"Indoor localization using Bluetooth Low Energy",** submitted to SASTRA University, Thanjavur by RAJA VENKATA SATYA PHANINDRA CHAVA (117005103) in partial fulfilment for the award of the degree of Bachelor of Technology in Electrical & Electronics Engineering is the work carried out under my guidance during the period February 2017 – June 2017.

Submitted for the University Examination held on ___________

External Examiner **Internal Examiner**

DECLARATION

I submit this project work entitled, **"Indoor localization using Bluetooth Low Energy",** to SASTRA University, Thanjavur in partial fulfilment of the requirements for the award of the degree of "**Bachelor of Technology**" in **"Electrical & Electronics Engineering".** I declare that it was carried out by me under the guidance of **DR. JAGANNATHAN SARANGAPANI**, PROFESSOR, DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY, USA.

Name : RAJA VENKATA SATYA PHANINDRA CHAVA

Register No. : 117005103

Signature :

ACKNOWLEDGEMENT

First of all, I express my gratitude to **Prof. R. Sethuraman**, Vice Chancellor, SASTRA University who facilitated and provided encouragement during the course of study. I extend my sincere thanks to **Dr. G. Bhalachandran**, Registrar, SASTRA University for providing the opportunity to pursue this project.

I extend my whole hearted thanks to **Dr. B. Viswanathan**, Dean, SEEE, SASTRA University for his moral support. I also thank **Dr. K. Vijayarekha,** Associate Dean, EEE who motivated me during the project work.

I would also express my gratitude to **Prof. M. Sridharan**, for giving me an opportunity to work on a project at Missouri University of Science and Technology (MST), USA. I am also very grateful for all his support throughout my stay in USA.

I owe a debt of deep gratitude to **Dr. Jagannathan Sarangapani**, Professor, Department of Electrical and Computer Engineering, Missouri University of Science and Technology, USA for offering me the opportunity to carry out this project at Missouri University of Science and Technology, USA and also for his invaluable support and sharing his extensive knowledge with me to make this project a reality.

Also, my deepest gratitude to my mentor **Krishnan Raghavan** for his continuous support and guidance throughout the process during the pursuit of my project work. His deep insight in the field and invaluable suggestions helped me in making progress.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BLE – Bluetooth Low Energy

RSSI – Received Signal Strength Indicator

WPS – Wi-Fi based Positioning System

RFID – Radio Frequency IDentification

IPS – Indoor Positioning System

GPS – Global Positioning System

CC2540EM – CC2540 Evaluation Modules

smartRF05EB – smartRF05 Evaluation Board

SOC_BB – System on Chip Battery Board

RF – Radio Frequency

PC – Personal Computer

USB – Universal Serial Bus

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver/Transmitter

csv – Comma Separated Values

TI – Texas Instruments

GPIO – General Purpose Input / Output

SRAM – Static Random Access Memory

IDE – Integrated Development Environment

GHz – Giga Hertz

Mbps – Megabits per second

KB – KiloBytes

AoA – Angle of Arrival

ABSTRACT

Technology is making the life of a human being easier. Development of a country is, now-a-days, parameterised by rate of progress of its technology. One such technological advancement is LOCALIZATION. Localization is defined as the process of finding the exact position of an object in an environment. Localization techniques, most importantly, are widely used for security and to facilitate human life.

While outdoor localization is highly developed and advanced, indoor localization is still in its nascent stages. Several techniques are available in the literature for indoor localization [1], most common ones include Wi-Fi based positioning system (WPS) and Radio Frequency Identification (RFID) based positioning system.

This project aims at developing a reliable indoor positioning system. In this project, a comprehensive methodology is developed for using Bluetooth Low Energy (BLE) and Received Signal Strength Indicator (RSSI) for indoor localization. Bluetooth Low Energy (BLE) is a low power consumption technology used to transmit data in the form of radio frequency signals with Bluetooth as a medium of connection. The advantage of using this technology over others [1] is that it is pervasively available, relatively cheap and consumes low power.

Firstly, a BLE hardware was setup and connection was established to record the RSSI values to facilitate localization. Next, data is collected in order to develop a generalised equation using MULTINOMIAL LOGISTIC REGRESSION facilitating real time prediction. Finally, the localization model is integrated with RFID handheld for localization and identification.

This report provides a detailed procedure of development of a model and its implementation for indoor positioning system using Bluetooth Low Energy.

The main conclusion of this report is that BLE is a potential alternative to traditional localization techniques and it can be used very efficiently for indoor localization.

CHAPTER 1 – INTRODUCTION

1.1 LOCALIZATION

Localization is the process of finding the exact position of an object in any environment. Depending on the environment under consideration, localization can be broadly classified into two categories:

- Outdoor localization
- Indoor localization / Indoor Positioning System (IPS)

One of the most popular localization technique is GPS which is readily available now-a-days. The fundamental question therefore is, If GPS is a readily available, what is the need for new localization techniques?

1.2 SIGNIFICANCE OF INDOOR POSITIONING SYSTEM

GPS signals are unavailable in indoor environments such as railway stations, malls, airport etc. due to its large margin of error. This is where indoor localization comes into significance. IPS can be used in situations where GPS technologies fail. In indoor environments, where GPS signals are likely to be blocked, use of radio frequency signals is a very efficient method.

1.3 CURRENT TECHNOLOGIES FOR INDOOR POSITION SYSTEM

1.3.1 WI-FI BASED POSITIONING SYSTEM (WPS) [1] [11]

WPS is widely used technology for indoor localization. Wi-Fi modems and MAC address of the access point can be used to find the neighbourhood of the object. The major drawback of such a method are

DISADVANTAGES:

- High power consumption
- Imprecise localization
- High cost of equipment like modems and routers

1.3.2 SENSOR CONCEPTS [2]

Many systems use physical measurements (like angle or distance) for indoor localization.

1.3.2.1 Angle of Arrival

Angle of Arrival (AoA) is the angle at which a signal arrives at the receiver. AoA is measured by highly directional sensors attached to the antennas. AoA usually uses triangulation by finding out 3 or more angle of arrivals for finding location.

1.3.2.2 Time of Arrival

Time of arrival is the time taken by the signal to propagate from the transmitter to receiver. Since the signal propagation rate is constant and known, time of travel can be used to directly calculate the distance. Measurements from 3 or more receivers can be used with trilateration to find out the location.

DISADVANTAGES:

- Large dependence on environmental conditions.
- Reflection and diffraction of signals can cause uncertainties and wrong outputs.
- High sensor cost.

1.3.3 OTHERS [2]

1. Radio Frequency Identification (RFID)

2. Ultrasound based system

3. Infrared (IR) based system

1.4 BLUETOOTH LOW ENERGY (BLE)

Bluetooth is a wireless technology that enables short range wireless communication among devices. Bluetooth Low Energy is a new specification of Bluetooth developed by Bluetooth Special Interest Group (SIG) [3]. BLE is version 4.0 of Bluetooth technology. It is also known as Bluetooth 4.0 or Bluetooth Smart. The major advantage of BLE over classic Bluetooth is its low power consumption and large battery life. The speciality of BLE is that the devices will be active only when data is getting transmitted and inactive otherwise. In this way, the device consumes less power.

Fig 1: Bluetooth Low Energy logo

1.4.1 States of operation

Bluetooth Low Energy devices operate in following states:

Standby: Does not transmit or receive packets

Advertising: Broadcasts advertisements in advertising channels

Scanning: Looks for advertisers

Initiating: Initiates connection to advertiser

Connection:

Master Role: Communicates with device in the Slave role.

Slave Role: Communicates with single device in Master Role.

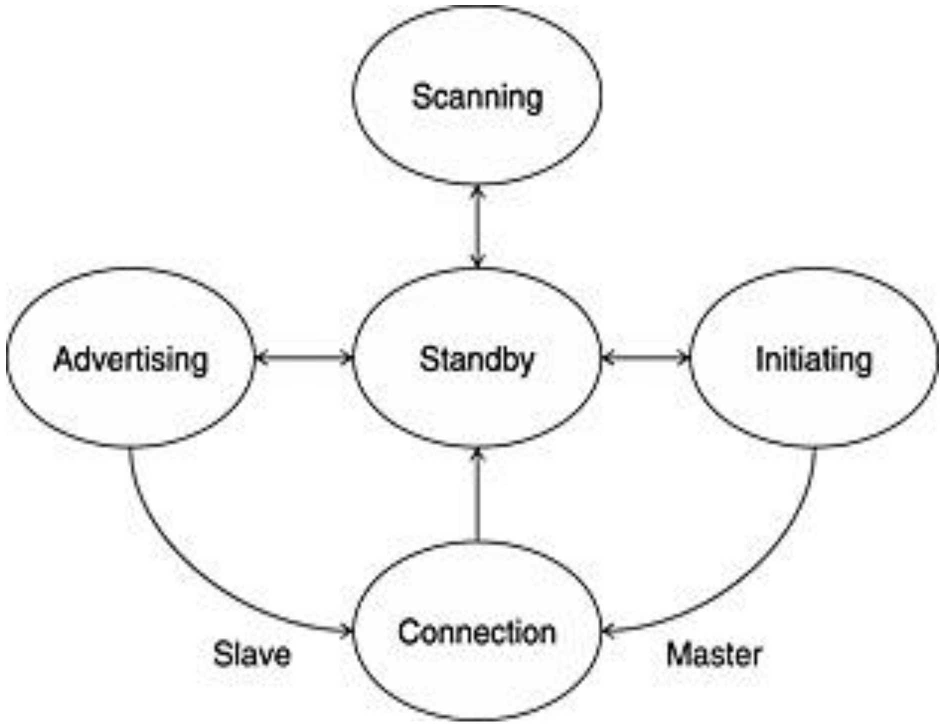

Fig 2: States of operation of BLE

1.5 CLASSICAL BLUETOOTH VS BLUETOOTH LOW ENERGY – A COMPARISON

SPECIFICATION	CLASSICAL BLUETOOTH	BLUETOOTH LOW ENERGY
Frequency Band	2.4 GHz	2.4 GHz
Data rate	1 to 3 Mbps	1 Mbps
Range	Up to 10 m	Up to 40 m
Power consumption	Low	Very low
Peak current consumption	<30 mA	<15 mA
Minimum time to send data	100 ms	3 ms
Battery life	Multiple weeks	Multiple months
Cost	Medium	Low
Accuracy	2 – 5 m	1 – 2 m
Developed for positioning	No	Yes

Table 1: Comparison of Classical Bluetooth and BLE

1.6 ADVANTAGES / DISADVANTAGES OF BLE

1.6.1 ADVANTAGES

- *Low power consumption:* Since BLE devices are active only when communication is happening and inactive/standby otherwise, BLE devices consumes very low energy or power.
- *High battery life:* Average battery life of a BLE module/beacon in 2 years.
- *Low cost of infrastructure*: Beacons are very low in cost compared to sensors or Wi-Fi modems and routers. Hence installation and working costs are very low comparatively.
- *High compatibility:* Bluetooth is best known personal area networking technology.
- *Pervasive availability of technology:* Bluetooth is mostly readily available in all the smart phones or electronic gadgets which makes it easier for communication and future uses.
- *Tight security:* BLE encryption and authentication is highly secured and data is safeguarded.

1.6.2 DISADVANTAGES

- The sample rate of RSSI is relatively very low.
- Less number of RSSI samples, than required, can be collected in real time in online phase on a moving object. Hence locating a moving object is relatively difficult.
- Large environmental changes can affect the prediction and result in wrong output.
- Range of Bluetooth is low compared to that of Wi-Fi.

1.7 PURPOSE OF THE PROJECT

The main motivation behind this project is to develop an indoor positioning system using Bluetooth Low Energy (BLE), Received Signal Strength Indicator (RSSI) and FINGERPRINTING method and finally integrate it with an RFID handheld. This framework can be used to locate the object as well as identify the object and record information corresponding to the object. Later this information can be transmitted and displayed on a mobile application.

There are four main stages of development in this project:

1. Hardware and setup
2. Experimentation and data collection
3. Data analysis and fingerprinting
4. Integration with RFID handheld and app development

The next chapter is focused on the first step of the development process.

CHAPTER 2 – HARDWARE AND SETUP

2.1 HARDWARE COMPONENTS

Main components used are:-

- CC2540 evaluation modules
- smartRF05 evaluation board
- System on Chip Battery Board (SOC_BB)

2.1.1 CC2540 module / Beacons [4]

CC2540 is a BLE module developed by Texas Instruments specifically for BLE applications. They are also called BEACONS. CC2540 system-on-chip has a high-performance and low-power 8051 microcontroller with an in-system-programmable flash memory of 128/256 KB, a SRAM memory of 8 KB, 21 GPIO ports, and other powerful features. CC2540 modules are connected via Bluetooth and they transmit and receive the radio frequency signals. They have an antenna connection for amplification of RF signals.

Fig 3: CC2540EM/beacon

2.1.2 smartRF05 evaluation board [5]

smartRF05 evaluation board is used as a medium of communication between the beacons and the PC. Two major functions of smartRF05EB are to:

1. Download the program into the beacon from the IDE.
2. Serial communication between the beacon and PC

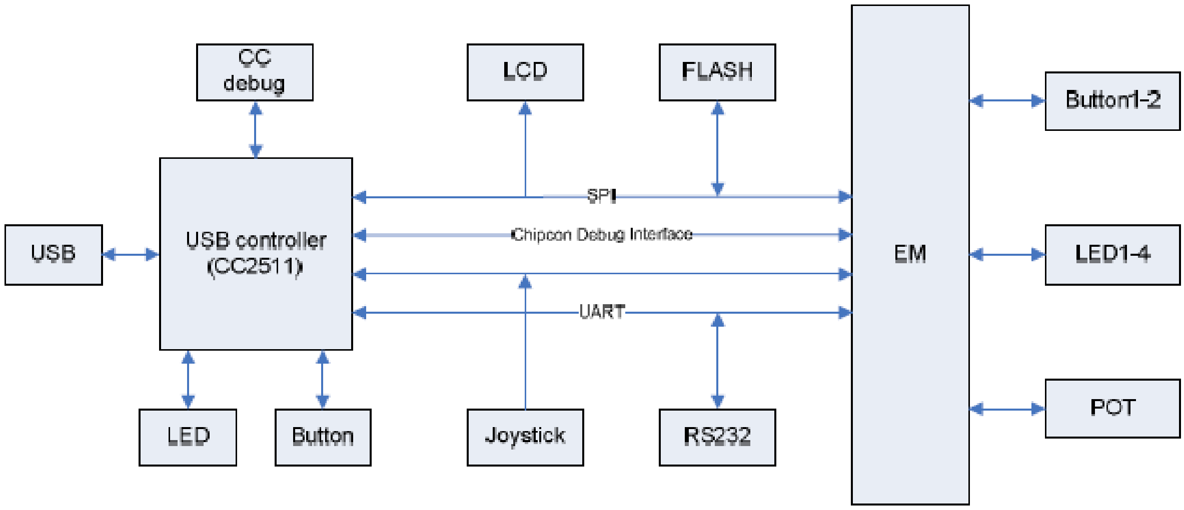

Fig 4: smartRF05EB block diagram

The board can communicate with the PC in four possible ways:

1. USB controller and USB connection to PC
2. Debug interface
3. SPI communication
4. UART RS-232 communication

In this case, UART serial communication is used using RS-232 interface and DB-9 serial cable. USB cable is used to power the CC2540EM through smartRF05EB.

Fig 5: smartRF05EB

2.1.3 System on Chip Battery Board [6]

The main function for this board is to power the CC2540EM with the use of two 1.5V AA batteries. This can also be used to download the code into the evaluation module and debugging of the chip.

Fig 6: System on Chip Battery Board

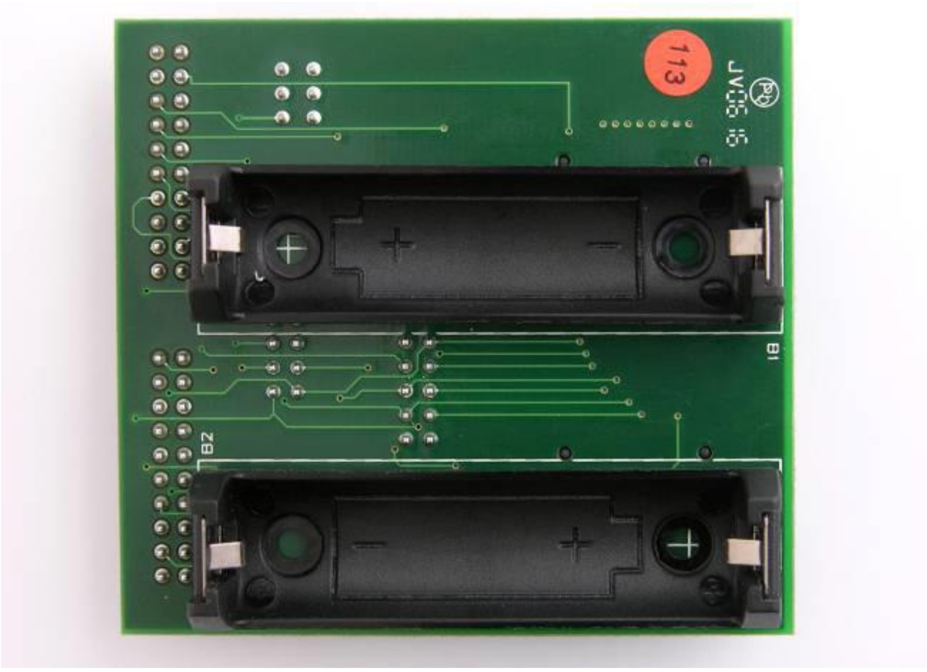

Fig 7: System on Chip Battery Board back side

2.2 IAR EMBEDDED WORKBENCH

CC2540EM can understand only HEX commands. Hence, IAR embedded workbench, an IDE developed by TI is used to write, edit, compile and debug a C program into a HEX file which is then downloaded on the CC2540EM using smartRF05EB.

2.3 RECEIVED SIGNAL STRENGTH INDICATOR

Received Signal Strength Indicator (RSSI) is a measurement of power of received RF signal compared to transmitted RF signal. When an RF signal is transmitted and received at the other end, there will be a loss of signal strength due to disturbances in the environment. This loss is taken into account and strength of the received signal compared to transmitted signal. Its unit is dB/dBm. Therefore, higher the RSSI number, stronger the signal. Thus, when an RSSI value is represented as a negative parameter, the closer the value to 0, the stronger the received signal is. In this project, RSSI is used as a parameter to locate the object.

2.4 HARDWARE SETUP

There are two types of beacons categorized according to our usage – master beacon and slave beacon. There are five beacons – four of them act as slaves and one acts as master. The master beacon is attached to the object to be localized and slave beacons are used to locate the object. Firstly, master beacon initiates and establishes a connection with the slave beacon, when this beacon is in the vicinity. Slave beacons transmit RF signal that are received by the master beacon. RSSI value are then evaluated on the master and are forwarded to the PC through serial communication (UART– RS232).

Slave beacons are programmed to automatically connect to the master beacon when in vicinity. Master beacon are programmed to receive the RF signal from the slave, compute the RSSI value and forward it to the PC using serial communication. Programs are written in C language in IAR Embedded Workbench and are downloaded to the beacons using smartRF05EB as HEX files. The slave beacons are then disconnected from smartRF05EB and powered through SOC_BB 1.1. The master beacon is still connected to smartRF05EB since the RSSI values should be sent to PC via serial communication for fingerprinting and further detection. A pictorial representation of this process is shown in Fig. 8.

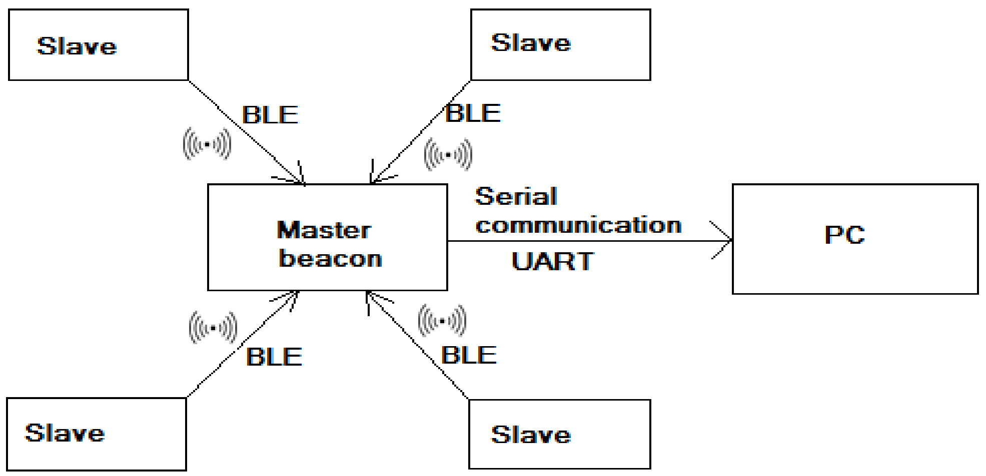

Fig 8: Hardware setup flow chart

CHAPTER 3 – EXPERIMENTATION AND DATA COLLECTION

3.1 POSITIONING TECHNIQUES

There are two main positioning techniques which can use RSSI measurements as a parameter, taking into account the characteristics of BLE technology. They are [7] [8]

- Trilateration
- Fingerprinting

3.1.1 TRILATERATION

Trilateration is a positioning technique which uses measured distances to determine the position of a point. The system measures distances from a point to, at least, three references, and forms circles where the intersection of these three will give the location of the point.

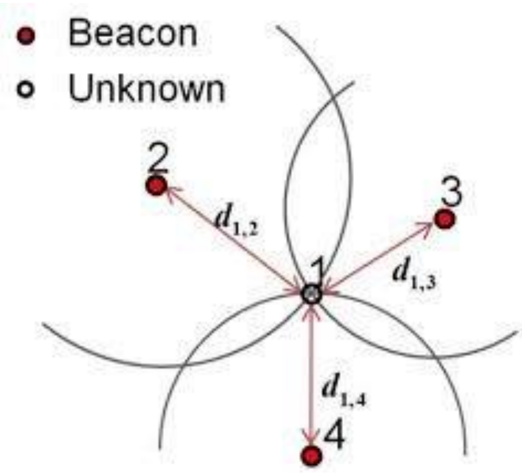

Fig 9: Trilateration representation

The distance between the master beacon and a particular slave beacon can be obtained from its corresponding RSSI value. RSSI is a measure of distance between two BLE devices.

Relationship between RSSI values and distance is represented by path loss model.

$$RSSI = RSSI' + 10 * n * \log\left(\frac{d}{d\prime}\right) + X_{\mathrm{g}}$$

Where RSSI' value is the signal power measured at distance d', n is multipath constant and X_g is fading constant.

• **Fading:** It is the time variation of the received signal. It depends on the environment and the movement of the devices.

• **Multipath:** It causes the signal to arrive in different ways to the device due to reflections, diffraction and scattering, depending on the indoor distribution such as furniture, walls, objects, etc.

3.1.2 FINGERPRINTING [7] [9] [12] [15]

Fingerprinting is a method where online data is compared, with a hypothesis model, to recorded offline data. RSSI values are recorded in order to estimate the position of a device. Firstly, an offline phase of the method is carried out by dividing the indoor room into a 2x3 grid with 6 divisions as 6 locations. Then, RSSI samples at each location are stored in a database. Secondly, in the online phase, few RSSI samples are gathered at unknown positions, of the same grid, which gives real time positioning of the object on which the RSSI samples is taken. These RSSI values are compared to the values stored in the database and an algorithm matches the actual value with the fingerprints obtained in the offline phase and locate the device with the positions that best fit the fingerprints.

In this project, the offline data is used to develop a generalized equation for location prediction using LOGISTIC REGRESSION which takes online samples of RSSI as input and gives the position corresponding to the input RSSI values as output.

3.1.3 TECHNIQUE UNDER CONSIDERATION

The implementation of fingerprinting with BLE is useful since trilateration is not a good technique due to problems such as [10]

1. Inaccurate distance measurement from RSSI values.
2. Inaccurate measurement of n and X_g parameters due to rapid changes in environmental conditions during and after experimentation.

Hence, FINGERPRINTING is used as a positioning technique and usage of TRILATERATION is ruled out in the scope of this project.

3.2 EXPERIMENTAL SETUP

- For the process of fingerprinting, an offline phase of the method is to be carried out by dividing the whole room into small areas with respective numeric positions. A rectangular part of our lab was divided into a 2x3 grid i.e. into 6 data points. Four slave beacons are put on the top at all the four corners of the experimental space.

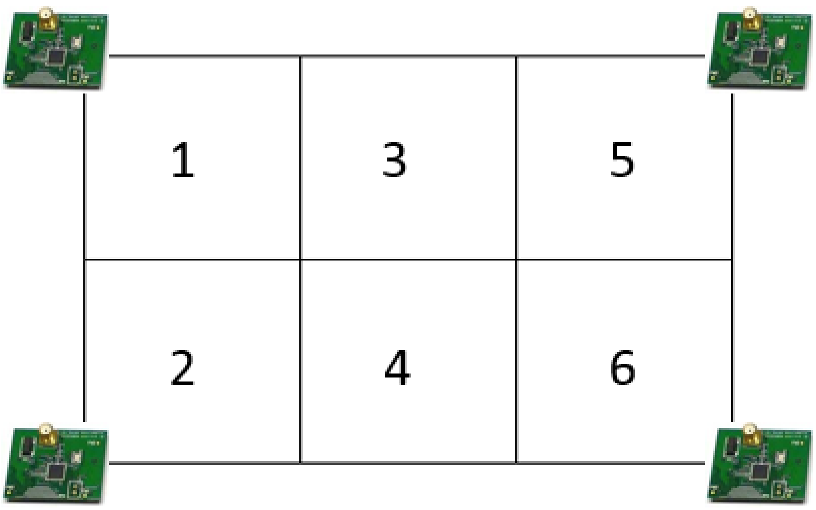

Fig 10: 2x3 grid division representation

Master beacon connected to smartRF05EB was put in the vicinity of all the four slave beacons so that it is connected to all of them via Bluetooth. Each data point was marked with its location on a paper.

3.3 OFFLINE PHASE / DATA COLLECTION FOR FINGERPRINTING

1. The master beacon is kept at all the 6 data points in the 2x3 grid.
2. Five thousand RSSI values from all the four slave beacons together are collected at each data point.
3. The 5000 RSSI values are serially sent to the PC and stored as csv files in the PC in a folder with its corresponding numeric position as the folder name.
4. These values are then divided according to its slave beacon address.
5. Then, every 5 values of a particular slave beacon is averaged so that the variation in RSSI values at a particular position from a particular beacon is minimised.
6. The final averaged RSSI values of all the four slave beacons at a particular position are again stored as csv files in the same folder.
7. The iteration was repeated four times, twice in a pre-determined sequence of positions and twice randomly, to account for various disturbances in the experimental space.
8. Data collected is composed of approximately 1000 averaged RSSI values i.e. 250 RSSI values of each beacon at any particular data point/position.
9. The whole process of data collection is performed using PYTHON programming language.
10. This amount of offline phase data is sufficiently large for development of a generalised equation to predict the real time position from online data fed.
11. A csv file is a comma separated values file which can store data in the form of plain text as a tabular columns.

12. The data serially received from master beacon is in the form of plain text. So, to store it in form of tabular columns, csv was used through python language.
13. The appearance of csv file is similar to an excel file and Microsoft excel can be used to open csv files.

	A	B
1	Slave	RSSI
2	69DF	49
3	41F3	62
4	6CDD	51
5	6CDD	59
6	459F	45
7	41F3	60
8	459F	45
9	41F3	59
10	6CDD	51
11	69DF	51
12	41F3	61
13	6CDD	57
14	459F	44
15	69DF	67
16	41F3	61

Fig 11: Pictorial representation of data saved

CHAPTER 4 – LOGISTIC REGRESSION BASED ANALYSIS

The algorithm explained in this chapter is intended to develop a generalised equation from the offline data which takes the online data as input and predicts the output. This algorithm is a subset of a broader set of algorithm known as machine learning.

4.1 MACHINE LEARNING

Machine learning is the ability of computers to learn on its own without being explicitly programmed. Machine learning explores the study and construction of algorithms that can learn and make predictions on data. Machine learning algorithms are used to make data-driven predictions or decisions by building a model, a generalised formula in our case, from sample inputs. Machine learning is used in cases when it is difficult to program explicit algorithms. Machine learning algorithms are used to produce reliable, repeatable decisions and results through learning from relationships and trends in data.

4.1.1 SIGNIFICANCE OF MACHINE LEARNING IN THIS PROJECT

Machine learning provides a machine ability to perform on new data after having experienced a learning data set. The learning data set is used to train a model is developed by the learner (machine) that enables it to produce accurate predictions in new cases.

In this project, the learning data set is the offline phase data, new cases are the online data provided, model will be a generalised equation taking the online data as input and the prediction will be the location of the master beacon.

4.1.2 ADVANTAGES

1. Future prediction: Recognize the pattern of past data and predict the future from the future data.
2. Can work on any amount of data, unlike human who can work on limited data.
3. No need of explicit programming.
4. Ability of machine to learn and work on its own.

4.1.3 DISADVANTAGES

1. Fails when finding patterns is difficult.
2. Fails when enough training data is not available.
3. Since the future is uncertain, it does not guarantee the accuracy of prediction.

THE MACHINE LEARNING ALGORITHM USED IN THIS PROJECT IS *MULTINOMIAL LOGISTIC REGRESSION*

4.2 MULTINOMIAL LOGISTIC REGRESSION

Multinomial logistic regression or multiclass logistic regression is used when there are more than two categories of outputs. The equation in the matrix form can be written as

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = f(W * \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T + C) \qquad (1)$$

Here, x_1, x_2, x_3, x_4 are ordered input RSSI values of 4 beacons at a particular point in the grid and y_1, y_2, y_3, y_4, y_5, y_6 are probabilities of the input point being data point represented by their subscript. Highest probability indicates exact position of the object.

Also,

$$W = \begin{bmatrix} \beta_{1,1} & \beta_{2,1} & \beta_{3,1} & \beta_{4,1} \\ \beta_{1,2} & \beta_{2,2} & \beta_{3,2} & \beta_{4,2} \\ \beta_{1,3} & \beta_{2,3} & \beta_{3,3} & \beta_{4,3} \\ \beta_{1,4} & \beta_{2,4} & \beta_{3,4} & \beta_{4,4} \\ \beta_{1,5} & \beta_{2,5} & \beta_{3,5} & \beta_{4,5} \\ \beta_{1,6} & \beta_{2,6} & \beta_{3,6} & \beta_{4,6} \end{bmatrix} \quad and \quad C = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{bmatrix}$$

$$f(t) = \frac{1}{1+e^{-(t)}} \tag{2}$$

$$\text{where} \quad t = WX^T + C \tag{3}$$

The equation in vector form can be written as

$$Y = f\big((W * X^T) + C\big), \tag{4}$$

where

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} \quad \text{and} \quad X = [\, x_1 \quad x_2 \quad x_3 \quad x_4 \,],$$

with Y Є R^{6x1} , X Є R^{4x1} , W Є R^{6x4} and C Є R^{6x1}. W is the coefficient matrix and C is the intercept matrix. Regression is used to find out the W and C i.e. the coefficients and intercepts of the logistic function for all the classes in the classification (locations in this case).

4.2.1 LOGISTIC FUNCTION

Logistic function takes any real input and always gives an output between 0 and 1 which is interpreted as probability. A logistic function f(t) is defined as follows:

$$f(t) = \frac{1}{1 + e^{-t}}$$

The graph of logistic function looks as below

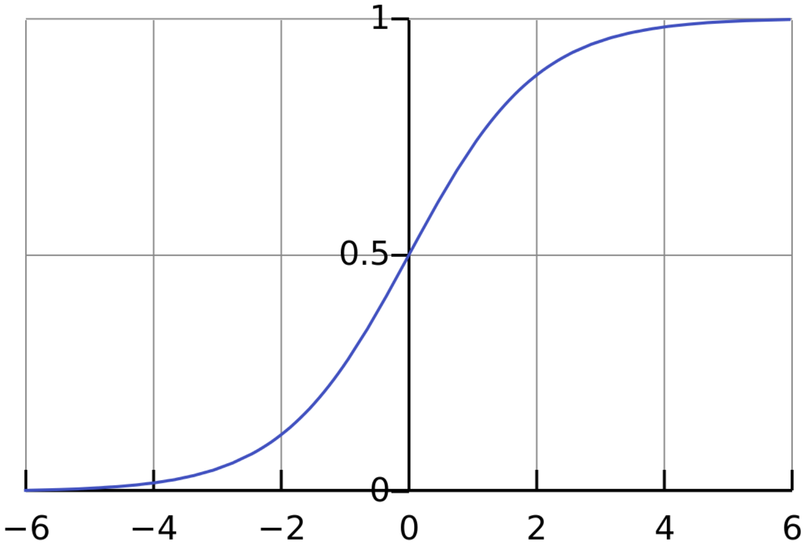

Fig 12: Logistic function graph

Where t is represented in x-axis and f(t) is represented in y-axis and t Є R

LOGISTIC REGRESSION IS USED TO FIND OUT W AND C VECTORS OF THE LOGISTIC FUNCTION USING REGRESSION ANALYSIS.

To understand how the logistic regression is done to find out the parameters, cost function is to be analysed.

4.2.2 COST FUNCTION

From equation (3), we know that

$$t = WX^T + C$$

Applying logistic function, f on both sides

$$T = f(t) = f(WX^T + C) \tag{5}$$

Suppose when an online data X is given, the output be ŷ. Then the cost function is given by

$$J(W;C) = -\hat{y} * \log(T) \tag{6}$$

where $\hat{y} \in R^{6x1}$. Suppose we have m data samples. Then cost function is defined as:

$$J(W;C) = \sum_{i=1}^{m} -\hat{y}^{(i)} * \log(T) \tag{7}$$

To find out W and C, J(W;C) is to be minimised. This procedure can be succinctly written as

1. Start with random initialization of W and C.
2. Keep changing W and C till we end up getting minimum of J(W;C).

This is analytically implemented using gradient descent.

4.2.3 GRADIENT DESCENT ALGORITHM

Gradient descent algorithm is an iterative optimization algorithm. Gradient descent algorithm is performed by repeating the update shown below till convergence.

$$W = W - \alpha \left(\frac{\partial \ J(W;C)}{\partial W} \right) \tag{8}$$

$$C = C - \alpha \left(\frac{\partial \ J(W;C)}{\partial C} \right) \tag{9}$$

Here, α is the learning rate. It can be manually chosen and also can be automatically updated by using various methods which are not discussed here. The significance of α is that

- If α is higher, gradient descent is done in higher steps and result is obtained quickly but it may not be accurate as the actual W and C matrix values can be over jumped.
- If α is lower, gradient descent is done in slower steps and result is obtained after more number of iterations but it will be more accurate comparatively.

One of the most important aspect of gradient descent algorithm is that

➢ W and C must be updated simultaneously in a single iteration, instead of doing one at a time, to yield accurate results.

In this way, the regression parameters are found and substituted in the logistic equation to find the probabilities of all the outputs.

➢ Finally, the online data (x_1, x_2, x_3, x_4) is fed to the logistic equation whose regression coefficients and intercepts are known through the probability that the object being in the locations, denoted by their position in the matrix, is obtained in the output matrix. For example, probability that (x_1, x_2, x_3, x_4) is from location 3 is found out by

$$y_3 = \frac{1}{1 + e^{-(\beta_{1,3}x_1 + \beta_{2,3}x_2 + \beta_{3,3}x_3 + \beta_{4,3}x_4 + c_3)}}$$

CHAPTER 5 - ANALYSIS AND RESULTS

The most important task of this project is to write a program to perform logistic regression on the offline phase data collected during experimentation and predicting the output with logistic function. The programming was done in python language.

CASE 1:

The experimental space was first divided into an 8x8 grid such that there are a total of 64 locations in the experimental space. In this case, the prediction of logistic regression is seen to be inaccurate. This is because the experimental space is small and the difference in RSSI values between the adjacent locations was not significant enough for the logistic regression to predict accurately.

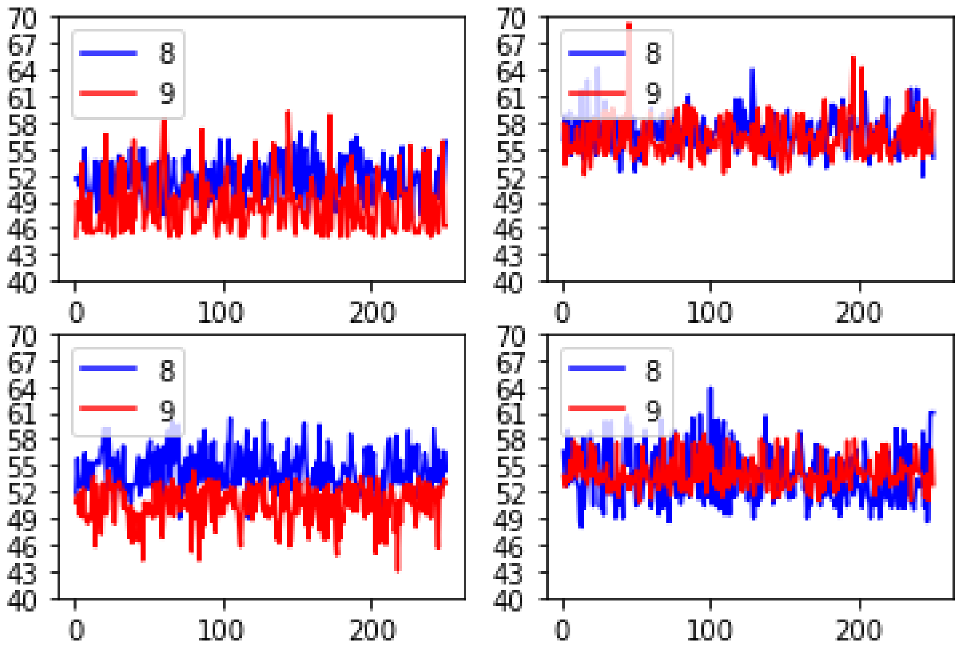

Fig 13: 8x8 simulation

The above figure represents RSSI values of 250 samples of the 4 beacons at two adjacent locations 8 and 9. It can be seen that there is no significant change in RSSI values. Y-axis represents RSSI values. First, second, third and fourth box represents beacon 1, 2, 3 and 4 respectively.

When logistic regression of approximately 1000 data samples were done for 8x8 grid, only 273 samples were predicted correctly and 751 were predicted incorrectly. So the accuracy was only 27% approximately.

```
In [1]: runfile('C:/Users/Phanindra/Desktop/New folder/for64/for64.py', wdir='C:/Users/
Phanindra/Desktop/New folder/for64')
C:\Users\Phanindra\Anaconda3\lib\site-packages\scipy\optimize\linesearch.py:285:
LineSearchWarning: The line search algorithm did not converge
  warn('The line search algorithm did not converge', LineSearchWarning)
C:\Users\Phanindra\Anaconda3\lib\site-packages\sklearn\utils\optimize.py:195:
UserWarning: Line Search failed
  warnings.warn('Line Search failed')
[[-0.6067966   0.08835491 -0.17247301 -0.20609702]
 [-0.26119267 -0.06926518 -0.06467544 -0.27650713]
 [-0.53925168  0.02487316 -0.03130769 -0.53552797]
 ...,
 [ 0.69240069  0.73132667  0.6813979   0.36796434]
 [ 0.6957899   0.77162335  0.35541993  0.67806567]
 [ 0.17085961  0.67264383 -0.02646399  0.0855141 ]]
[  47.79576841   37.41529447   57.55626963 ..., -137.23123805 -139.77803795
  -47.75994279]
FOR 64 locations
The number of samples predicted correctly are 273
The number of samples predicted incorrectly are 751
```

Fig 14: 8x8 accuracy

CASE 2:

The experimental space was then divided into a 4x4 grid i.e., into 16 locations. Even in this case, the locations were not able to predict accurately for the same reason as in case 1. When the online RSSI samples are collected, the RSSI samples of two adjacent locations were almost similar. Adjacent locations were being predicted instead the actual location. For example, when the object was in 10 position, the adjacent locations such as 9, 11, 6 or 14 are being predicted. Hence the output was inaccurate again.

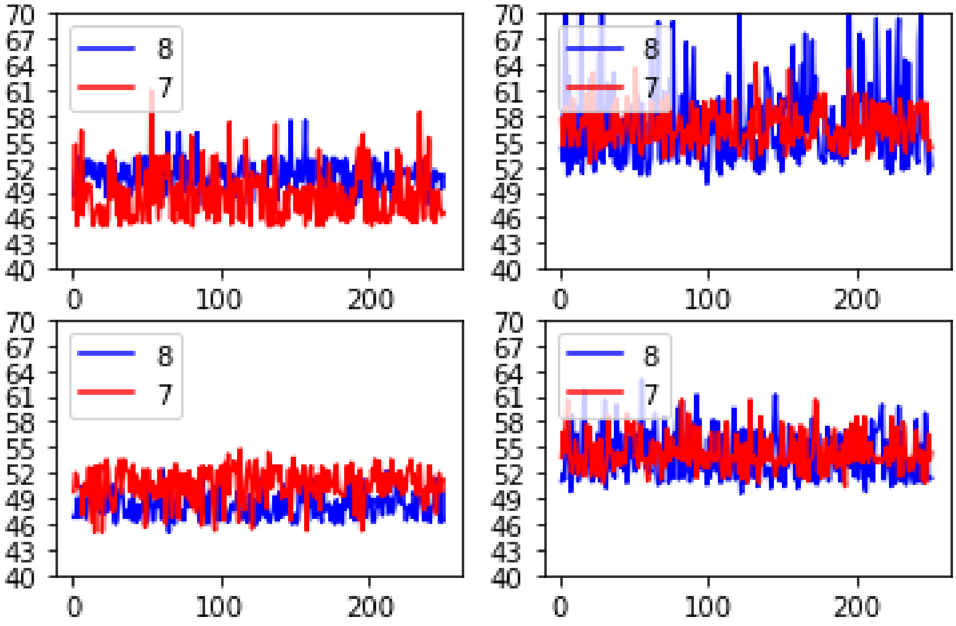

Fig 15: 4x4 simulation

The above figure represents RSSI values of 250 samples of the 4 beacons at two adjacent locations 7 and 8. It can be seen that there is no significant change in RSSI values. Y-axis represents RSSI values. First, second, third and fourth box represents beacon 1, 2, 3 and 4 respectively.

When logistic regression of approximately 1000 data samples were done for 4x4 grid, 655 samples were predicted correctly and 353 were predicted incorrectly. So the accuracy was 65% approximately which is still not good enough to predict a location.

```
In [1]: runfile('C:/Users/Phanindra/Desktop/untitled1.py', wdir='C:/Users/Phanindra/
Desktop')
[[-0.97706254 -0.35831687  0.04761915 -0.70786268]
 [-0.98141019 -0.35050539  0.26579651 -1.28740603]
 [-1.22316299  1.32950076 -2.48863755  0.83653808]
 ...,
 [ 0.58324936 -1.76059319  1.59418063  2.34111209]
 [ 1.93571931  1.96935238  0.81163326  0.46294335]
 [ 0.7693926   1.52148804  0.10301637 -0.09368442]]
[ 111.39399107  129.72266513   68.68978558 ..., -153.24990377 -287.86115193
 -120.85567996]
FOR 16 locations
The number of samples predicted correctly are 655
The number of samples predicted incorrectly are 353
```

Fig 16: 4x4 accuracy

CASE 3:

In this case, the experimental space was finally divided into 2x3 grid i.e., 6 locations considering its size. This time the adjacent locations was far enough for the RSSI values to vary significantly that the logistic regression could distinguish between locations and predict the output correctly.

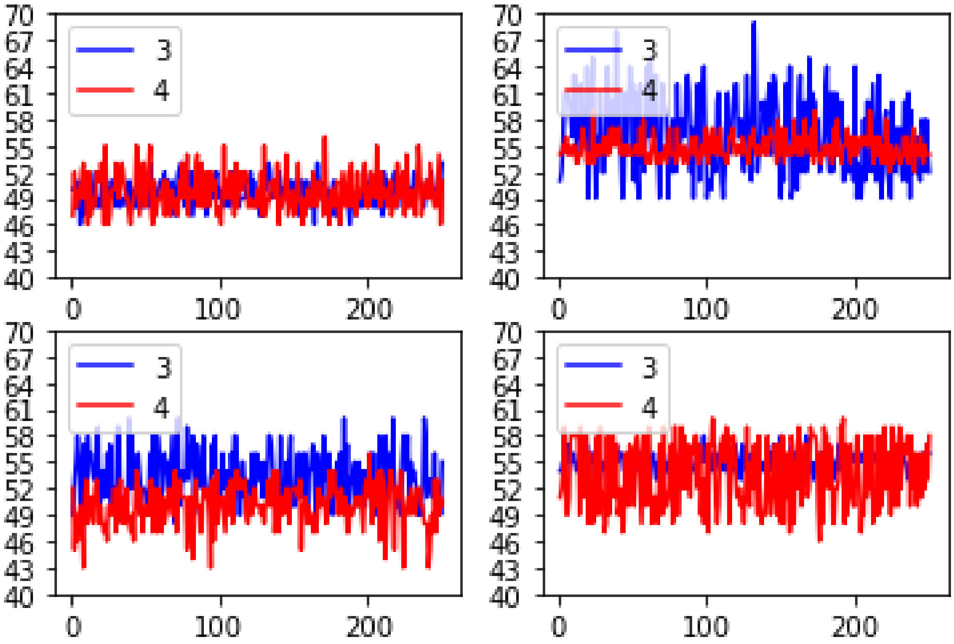

Fig 17: 2x3 simulation

The above figure represents RSSI values of 250 samples of the 4 beacons at two adjacent locations 3 and 4. It can be seen that there is a significant change in RSSI values. Y-axis represents RSSI values. First, second, third and fourth box represents beacon 1, 2, 3 and 4 respectively.

When logistic regression of approximately 1000 data samples were done for 2x3 grid, 874 samples were predicted correctly and 128 were predicted incorrectly. So the accuracy was 85% approximately with which exact location can be predicted with 100% certainty.

```
In [2]: runfile('C:/Users/Phanindra/Desktop/untitled2.py', wdir='C:/Users/Phanindra/
Desktop')
[[-0.27879998 -0.1675953  -0.55141319 -0.09946124]
 [ 0.17524216 -0.88826629  0.68758681 -0.13349286]
 [-0.24168931  0.51509126  0.68010211 -0.02440393]
 [-0.35448862 -0.36288186 -0.53309404 -0.19994965]
 [ 1.08020708  0.86798582  0.05926881  0.49819605]
 [-0.38047305  0.0356676  -0.34245068 -0.04088927]]
[  58.28019313    7.94205234  -48.41119511   76.74399221 -133.37250948
   38.8174669 ]
FOR 6 locations
The number of samples predicted correctly are 874
The number of samples predicted incorrectly are 128
```

Fig 18: 2x3 accuracy

To find the final location of the object, few data samples, say 20, are taken and location predicted by maximum number of samples is the final location of the object. For example, with 85% accuracy, 17 out of the 20 samples predict correctly. So, the final location will be the location predicted by those 17 samples.

When the master beacon was put at 2 location, the output was

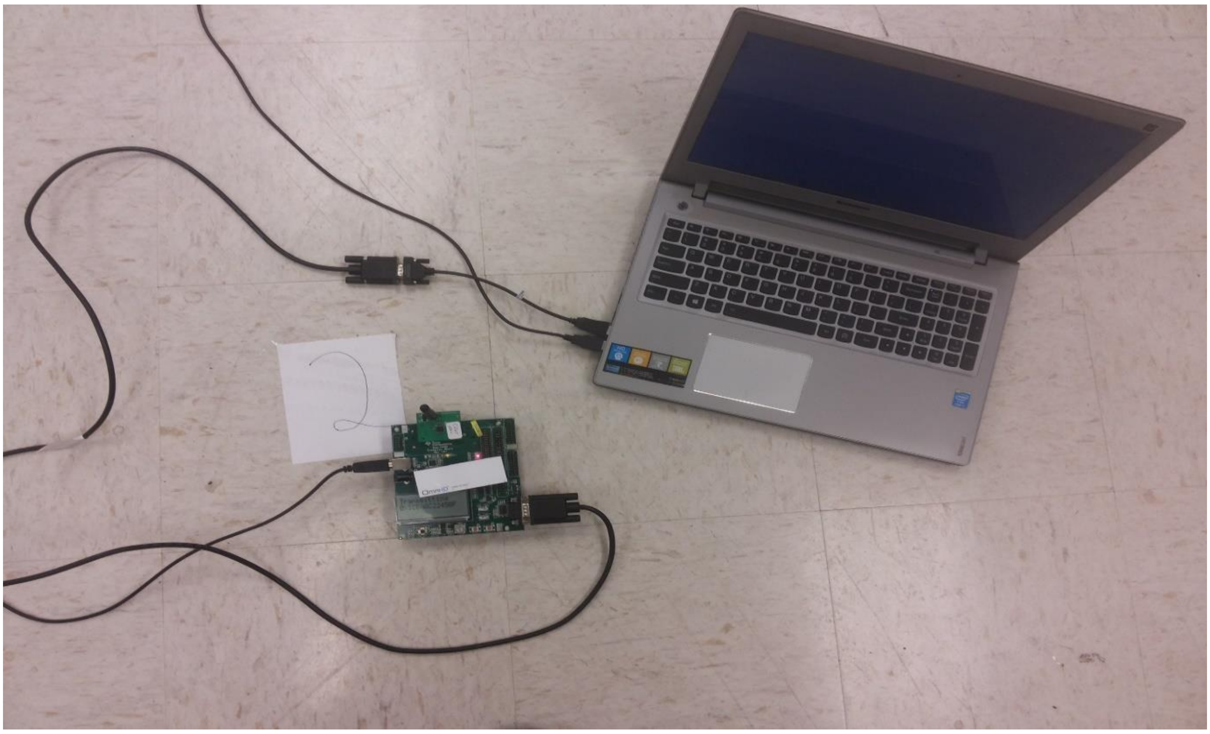

Fig 19: Object at location 2

```
In [1]: runfile('C:/Users/Phanindra/Desktop/codes/Demo.py', wdir='C:/Users/Phanindra/
Desktop/codes')
Starting to regress
[[-0.27879933 -0.16759507 -0.55141231 -0.09946109]
 [ 0.17524164 -0.88826834  0.68758574 -0.13349345]
 [-0.24168986  0.5150907   0.68010097 -0.02440434]
 [-0.35448766 -0.36288139 -0.53309273 -0.1999494 ]
 [ 1.080208    0.86798616  0.05926807  0.49819673]
 [-0.38047231  0.03566791 -0.34244984 -0.04088908]]
[  58.28010596    7.94228267  -48.41104366   76.74384966 -133.37256574
   38.81737111]
Finished Regression

Enter to start0
The object is at location 2
```

Fig 20: Output for object at location 2

When the object was placed at 3 location, the output was

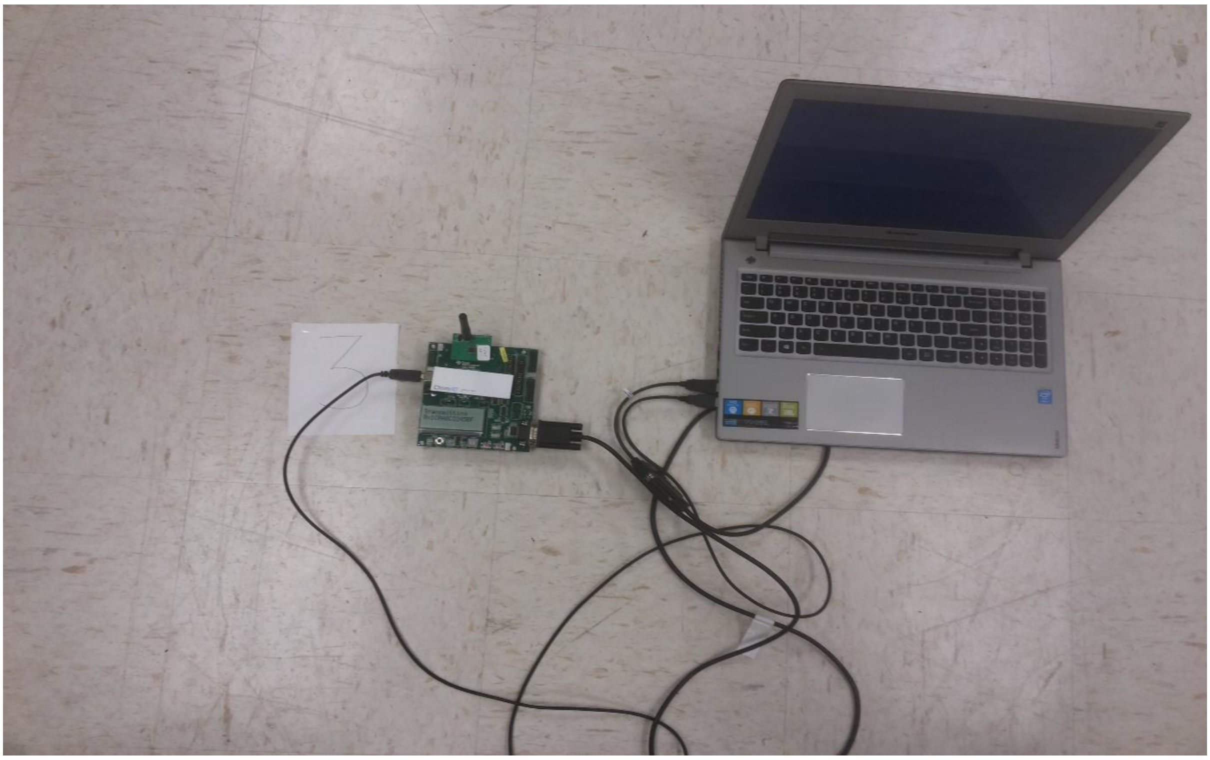

Fig 21: Object at location 3

```
In [1]: runfile('C:/Users/Phanindra/Desktop/codes/Demo.py', wdir='C:/Users/Phanindra/
Desktop/codes')
Starting to regress
[[-0.27879933 -0.16759507 -0.55141231 -0.09946109]
 [ 0.17524164 -0.88826834  0.68758574 -0.13349345]
 [-0.24168986  0.5150907   0.68010097 -0.02440434]
 [-0.35448766 -0.36288139 -0.53309273 -0.1999494 ]
 [ 1.080208    0.86798616  0.05926807  0.49819673]
 [-0.38047231  0.03566791 -0.34244984 -0.04088908]]
[  58.28010596    7.94228267  -48.41104366   76.74384966 -133.37256574
   38.81737111]
Finished Regression

Enter to start0
The object is at location 2

Enter to start0
The object is at location 3

Enter to start
```

Fig 22: Output for object at location 3

<u>SIMILARLY, THE MODEL WAS ACCURATELY ABLE TO LOCATE THE OBJECT IN ALL THE 6 LOCATIONS INSIDE THE EXPERIMENTAL SPACE OF THE 2X3 GRID</u>.

CHAPTER 6 – INTEGRATION WITH RFID HANDHELD

The final step in this development process is to integrate the BLE positioning system with a RFID handheld to locate an object as well as identify the object. The RFID handheld can read an RFID tag and get the tag number associated with that tag. The information of the object can be stored in an offline database and that information can be associated with the RFID tag number of an object. So when the tag number is identified by the RFID handheld, the tag number is matched with the information in the database. In this way, the object is identified and information of that object is known.

Fig 23: RFID handheld

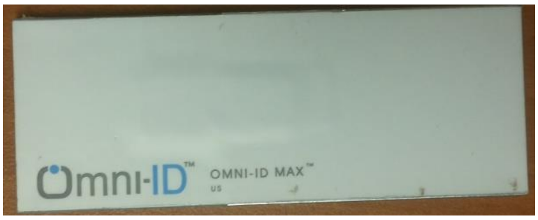

Fig 24: RFID tag

Also, an android application is developed to showcase the output of the final integrated model on a single screen. The output includes the tag ID, to which the information related to the object is associated with, and the location of the object.

The handheld is developed by Zebra technologies [18]. Also, there is an already developed android application by Zebra Technologies which can read the RFID tag and display the tag number [19]. Zebra Technologies made the source code of the application available to us. The application code (given by Zebra Technologies) [19] was therefore modified as per the project requirements and a feature to display the location of the object alongside the tag ID was added.

To display the location in the android application, the location information evaluated by the logistic regression python code in PC is sent to the android mobile with application developed installed in it via Bluetooth as a text file. The application code was edited in such a way that it reads the .txt file in the Bluetooth folder and displays the text in that file (which is the location of the object) alongside with the tag number.

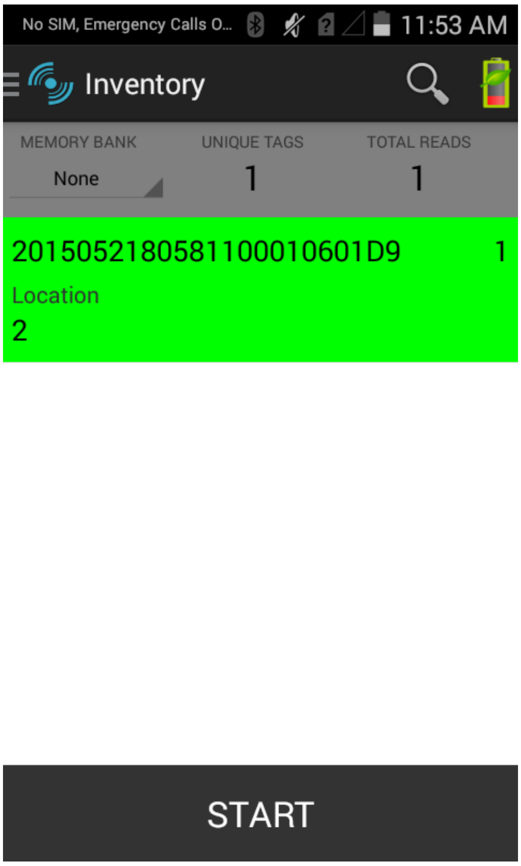

Fig 25: Sample output display of android app

In the above figure of the output screen, the number '2015052180581100010601D9' represents the tag ID and the number below that represents the location of the object.

So finally, when the master beacon is at 2 location, the output will be

Fig 26: Output display in application when object is at location 2

When the master beacon is at 3 location, the output is

Fig 27: Output display in application when object is at location 3

CHAPTER 7 – CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

Bluetooth Low Energy can be used as an alternative to traditional technologies used for localization and navigation such as GPS or Wi-Fi. A model for Indoor Positioning System is designed and developed which can be used to locate objects and used by a human as a navigation system in an indoor environment. This model uses low power, is highly economical and has high working life.

CC2540 beacons are used as BLE modules which communicates with the object to be located. RSSI from the beacons is used as a parameter for positioning. A small indoor environment was considered as experimental space and it was divided into locations. A model for localization was constructed using FINGERPRINTING and LOGISTIC REGRESSION. This model could accurately locate objects placed in all the locations of the experimental space. This localization model was integrated with an RFID handheld which, as a final model, can be used to locate and identify an object. The final output was displayed successfully in an android application developed.

This project thus fulfilled the intention of its development and the proposed model for indoor localization is found to be accurate and useful in real time.

7.2 FUTURE WORK

In this model, a master beacon along with the smartRF05EB is to be attached to the object. Also, the master beacon is communicating with the PC using DB-9 cable. But this model can be extended such that the attachment of beacon to the object to be located can be prevented and a small BLE chip, attached to the object or a mobile phone with Bluetooth enabled, can be directly used for navigation. Communication with the PC can also happen wirelessly.

This model can also be extended such that the use of PC to store the data can be eliminated and all the communication and storage can happen through a common server. The serial communication through DB-9 cable can be eliminated and data can be transferred to the server wirelessly.

Also, this model can be extended by integrating this with Wi-Fi based positioning system where WPS can tell the vicinity of the object and BLE positioning system can give the exact location in that vicinity.

APPENDIX

A.1 OFFLINE PHASE

```
import serial
import csv

serial = serial.Serial("COM1", 115200)
with open("data_unfiltered.txt", "w") as f:
    i=1
    while i<=5000:
        for v in range(1,22):
            data = serial.read()
            if v==21:
                f.write("\n")
            elif v==17:
                pass
            else:
                f.write(str(data))
        i+=1

with open("data_unfiltered.txt", "r") as f:
    with open("data_filtered.txt", "w") as f1:
        for s in range(5000):
            data1=f.readline()
            f1.write(data1[11:15]+","+data1[16:19]+"\n")

with open('data_filtered.txt', 'r') as f1:
    stripped_data = (line.strip() for line in f1)
    grouped = [line.split(',') for line in stripped_data]
    with open('data_filtered.csv', 'w') as f2:
        writer = csv.writer(f2)
        writer.writerow(['Slave', 'RSSI'])
        writer.writerows(grouped)

with open('data_filtered.csv', 'r') as f2:
    reader = csv.reader(f2)
    v=1,x=1,y=1,z=1
    fb1=open('data_filtered_b1.csv', 'w')
    fb2=open('data_filtered_b2.csv', 'w')
    fb3=open('data_filtered_b3.csv', 'w')
    fb4=open('data_filtered_b4.csv', 'w')
    for row in reader:
        for field in row:
            if field=="69DF":
                writer = csv.writer(fb4)
                if v==1:
                    writer.writerow(["Slave","RSSI"])
```

```
                v+=1
            writer.writerow(row)
        elif field=="41F3":
            writer = csv.writer(fb2)
            if x==1:
                writer.writerow(["Slave","RSSI"])
                x+=1
            writer.writerow(row)
        elif field=="6CDD":
            writer = csv.writer(fb1)
            if y==1:
                writer.writerow(["Slave","RSSI"])
                y+=1
            writer.writerow(row)
        elif field=="459F":
            writer = csv.writer(fb3)
            if z==1:
                writer.writerow(["Slave","RSSI"])
                z+=1
            writer.writerow(row)
    fb1.close()
    fb2.close()
    fb3.close()
    fb4.close()

v=[]
w=[]
with open('data_filtered_b1.csv', 'r') as fb1:
    reader=csv.reader(fb1)
    for row in reader:
        if row[1]!="RSSI":
            v.append(row[1])
z=len(v)%5
y=len(v)-z
a=int(y/5)
i=0
for r in range(a):
    s=0
    for x in range(5):
        s+=int(v[i])
        i+=1
    avg=(s/5)
    w.append(str(avg))
with open('data_filtered_b1_avg.csv', 'w') as fb1a:
    writer=csv.writer(fb1a)
    writer.writerow(["Slave","RSSI"])
    for t in w:
        writer.writerow(["6CDD",t])
```

```
q=[]
e=[]
with open('data_filtered_b2.csv', 'r') as fb2:
    reader=csv.reader(fb2)
    for row in reader:
        if row[1]!="RSSI":
            q.append(row[1])
z=len(q)%5
y=len(q)-z
a=int(y/5)
i=0
for r in range(a):
    s=0
    for x in range(5):
        s+=int(q[i])
        i+=1
    avg=(s/5)
    e.append(str(avg))
with open('data_filtered_b2_avg.csv', 'w') as fb2a:
    writer=csv.writer(fb2a)
    writer.writerow(["Slave","RSSI"])
    for t in e:
        writer.writerow(["41F3",t])

o=[]
p=[]
with open('data_filtered_b3.csv', 'r') as fb3:
    reader=csv.reader(fb3)
    for row in reader:
        if row[1]!="RSSI":
            o.append(row[1])
z=len(o)%5
y=len(o)-z
a=int(y/5)
i=0
for r in range(a):
    s=0
    for x in range(5):
        s+=int(o[i])
        i+=1
    avg=(s/5)
    p.append(str(avg))
with open('data_filtered_b3_avg.csv', 'w') as fb3a:
    writer=csv.writer(fb3a)
    writer.writerow(["Slave","RSSI"])
    for t in p:
        writer.writerow(["459F",t])

g=[]
h=[]
```

```
with open('data_filtered_b4.csv', 'r') as fb4:
    reader=csv.reader(fb4)
    for row in reader:
        if row[1]!="RSSI":
            g.append(row[1])
z=len(g)%5
y=len(g)-z
a=int(y/5)
i=0
for r in range(a):
    s=0
    for x in range(5):
        s+=int(g[i])
        i+=1
    avg=(s/5)
    h.append(str(avg))
print (h)
with open('data_filtered_b4_avg.csv', 'w') as fb4a:
    writer=csv.writer(fb4a)
    writer.writerow(["Slave","RSSI"])
    for t in h:
        writer.writerow(["69DF",t])
```

A.2 ONLINE PHASE

A.2.1 Demo_data.py

```
import csv
import numpy as np

def data_sample():
    import serial
    serial = serial.Serial("COM1", 115200)

    with open("data_unfiltered.txt", "w") as f:
        i=1
        while i<=500:
            for v in range(1,22):
                data = serial.read()
                if v==21:
                    f.write("\n")
                elif v==17:
                    pass
                else:
                    f.write(str(data))
            i+=1
```

```
with open("data_unfiltered.txt", "r") as f:
        with open("data_filtered.txt", "w") as f1:
            for s in range(500):
                data1=f.readline()
                f1.write(data1[11:15]+","+data1[16:19]+"\n")

    with open('data_filtered.txt', 'r') as f1:
        stripped_data = (line.strip() for line in f1)
        grouped = [line.split(',') for line in stripped_data]
        with open('data_filtered.csv', 'w') as f2:
            writer = csv.writer(f2)
            writer.writerow(['Slave', 'RSSI'])
            writer.writerows(grouped)

    with open('data_filtered.csv', 'r') as f2:
        reader = csv.reader(f2)
        v=1,x=1,y=1,z=1
        fb1=open('data_filtered_b1.csv', 'w')
        fb2=open('data_filtered_b2.csv', 'w')
        fb3=open('data_filtered_b3.csv', 'w')
        fb4=open('data_filtered_b4.csv', 'w')
        for row in reader:
            for field in row:
                if field=="69DF":
                    writer = csv.writer(fb4)
                    if v==1:
                        writer.writerow(["Slave","RSSI"])
                        v+=1
                    writer.writerow(row)
                elif field=="41F3":
                    writer = csv.writer(fb2)
                    if x==1:
                        writer.writerow(["Slave","RSSI"])
                        x+=1
                    writer.writerow(row)
                elif field=="6CDD":
                    writer = csv.writer(fb1)
                    if y==1:
                        writer.writerow(["Slave","RSSI"])
                        y+=1
                    writer.writerow(row)
                elif field=="459F":
                    writer = csv.writer(fb3)
                    if z==1:
                        writer.writerow(["Slave","RSSI"])
                        z+=1
                    writer.writerow(row)
        fb1.close()
        fb2.close()
        fb3.close()
```

```
    fb4.close()

v=[]
w=[]
with open('data_filtered_b1.csv', 'r') as fb1:
    reader=csv.reader(fb1)
    for row in reader:
        if row[1]!="RSSI":
            v.append(row[1])
z=len(v)%5
y=len(v)-z
a=int(y/5)
i=0
for r in range(a):
    s=0
    for x in range(5):
        s+=int(v[i])
        i+=1
    avg=(s/5)
    w.append(str(avg))
B1_rand = w

q=[]
e=[]
with open('data_filtered_b2.csv', 'r') as fb2:
    reader=csv.reader(fb2)
    for row in reader:
        if row[1]!="RSSI":
            q.append(row[1])
z=len(q)%5
y=len(q)-z
a=int(y/5)
i=0
for r in range(a):
    s=0
    for x in range(5):
        s+=int(q[i])
        i+=1
    avg=(s/5)
    e.append(str(avg))
B2_rand = e

o=[]
p=[]
with open('data_filtered_b3.csv', 'r') as fb3:
    reader=csv.reader(fb3)
    for row in reader:
        if row[1]!="RSSI":
            o.append(row[1])
z=len(o)%5
```

```
    y=len(o)-z
    a=int(y/5)
    i=0
    for r in range(a):
        s=0
        for x in range(5):
            s+=int(o[i])
            i+=1
        avg=(s/5)
        p.append(str(avg))
    B3_rand = p
    g=[]
    h=[]
    with open('data_filtered_b4.csv', 'r') as fb4:
        reader=csv.reader(fb4)
        for row in reader:
            if row[1]!="RSSI":
                g.append(row[1])
    z=len(g)%5
    y=len(g)-z
    a=int(y/5)
    i=0
    for r in range(a):
        s=0
        for x in range(5):
            s+=int(g[i])
            i+=1
        avg=(s/5)
        h.append(str(avg))
    B4_rand = h
    return np.array(B1_rand), np.array(B2_rand), np.array(B3_rand), np.array(B4_rand)
```

A.2.2 Demo_py

```
import os
import csv
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from math import exp
import bluetooth
from PyOBEX.client import Client

B1=np.ones((6,550))
B2=np.ones((6,550))
B3=np.ones((6,550))
B4=np.ones((6,550))
```

```
def test_regression():
    global B1
    global B2
    global B3
    global B4
    b1=[]
    b2=[]
    b3=[]
    b4=[]
    rootDir = r'C:\Users\Phanindra\Desktop\Data req\New folder\6 data pts'
    for dirName, subdirList, fileList in os.walk(rootDir):
        if len(fileList)>0:
            i=1
            for fname in fileList:
                if (fname[15]=='1'):
                    x=np.genfromtxt(dirName+'\\'+fname,unpack=True,
skip_header=1,delimiter=',')[1,:]
                    x=x.tolist()
                    b1.append(x)
                    i+=1
                if (fname[15]=='2'):
                    y=np.genfromtxt(dirName+'\\'+fname,unpack=True,
skip_header=1,delimiter=',')[1,:]
                    y=y.tolist()
                    b2.append(y)
                    i+=1
                if (fname[15]=='3'):
                    z=np.genfromtxt(dirName+'\\'+fname,unpack=True,
skip_header=1,delimiter=',')[1,:]
                    z=z.tolist()
                    b3.append(z)
                    i+=1
                if (fname[15]=='4'):
                    v=np.genfromtxt(dirName+'\\'+fname,unpack=True,
skip_header=1,delimiter=',')[1,:]
                    v=v.tolist()
                    b4.append(v)
                    i+=1

    j=0
    for i in range(6):
        v=b1[j]+b1[j+1]
        j=j+2
        for k in range(len(v)):
            B1[i][k]=v[k]
    j=0
    for i in range(6):
        v=b2[j]+b2[j+1]
        j=j+2
        for k in range(len(v)):
```

```
            B2[i][k]=v[k]
    j=0
    for i in range(6):
        v=b3[j]+b3[j+1]
        j=j+2
        for k in range(len(v)):
            B3[i][k]=v[k]
    j=0
    for i in range(6):
        v=b4[j]+b4[j+1]
        j=j+2
        for k in range(len(v)):
            B4[i][k]=v[k]
    B1=B1[:,:475]
    B2=B2[:,:475]
    B3=B3[:,:475]
    B4=B4[:,:475]

    with open('data.csv','w') as f:
        writer=csv.writer(f)
        writer.writerow(["position","beacon1","beacon2","beacon3","beacon4"])
        for i in range(6):
                    for x,y,z,v in zip(B1[i][:],B2[i][:],B3[i][:],B4[i][:]):
                writer.writerow([str(i+1),str(x),str(y),str(z),str(v)])

    df = pd.read_csv("data.csv")
    train_cols = df.columns[1:]
    out_cols = df.columns[0]
    v=np.array(df[train_cols])
    s=np.array(df[out_cols])
    s1 = [[i] for i in s]
    s2=np.array(s1)
    model = LogisticRegression(multi_class='multinomial',solver ='newton-cg',max_iter=1000)
    model = model.fit(v,s2.ravel())
    return model

def formulae(r1,r2,r3,r4):
    del op_e[:]
    del op[:]
    for i in range(6):
        b=(y[i][0]*r1)+(y[i][1]*r2)+(y[i][2]*r3)+(y[i][3]*r4)+z[i]
        op_e.append(b)
    for i in range(6):
        l=((exp(op_e[i]))/(1+(exp(op_e[i]))))
        op.append(l)

print "Starting to regress"
model = test_regression()
print model.coef_
print model.intercept_
```

```
print "Finished Regression"
from Demo_data import data_sample
np.savetxt('coef.csv', np.array(model.coef_), delimiter=',')
np.savetxt('intercep.csv', np.array(model.intercept_), delimiter=',')

print "Loading parameters"
y=np.loadtxt('coef.csv', delimiter =',')
z=np.loadtxt('intercep.csv', delimiter =',')

print "Loaded"
while True:
    x = input("Enter to start")
    B1_rand, B2_rand, B3_rand, B4_rand = data_sample()
    op_e=[]
    op=[]
    a=[0 for i in range(6)]
    for j in range(min([B1_rand.shape[0], B2_rand.shape[0], B3_rand.shape[0],
B4_rand.shape[0] ])):
        formulae(float(B1_rand[j]),float(B2_rand[j]),float(B3_rand[j]),float(B4_rand[j]))
        b=op.index(max(op))
        a[b]+=1
        count=(a.index(max(a)))+1
    print count

    address = '40:83:DE:A4:18:20'
    svc = bluetooth.find_service(address=address, uuid='1105')
    first_match = svc[0]
    port = first_match["port"]
    name = first_match["name"]
    host = first_match["host"]
    print("Connecting to \"%s\" on %s" % (name, host))
    client = Client(host, port)
    client.connect()
    client.put("test.txt", str(count))
    client.disconnect()
```

REFERENCES

[1] A. Baniukevic, et al., “Hybrid Indoor Positioning With Wi-Fi and Bluetooth: Architecture and Performance”. *International Conference on Mobile Data management,* IEEE 2013, 3-6 June 2013, pp. 207-216, doi: 10.1109/MDM.2013.30

[2] Wikipedia contributors. "Indoor positioning system." *Wikipedia, The Free Encyclopedia*”. Wikipedia, The Free Encyclopedia, 7 May. 2017. Web. 29 May. 2017

[3] Bluetooth SIG. Bluetooth Core Specification Version 4.0. Bluetooth, June 2010. https://www.bluetooth.org/en-us/specification/adopted-specifications. Retreived 2014-04-30

[4] CC2540 Development Kit Quick Start Guide : www.ti.com/lit/pdf/swru301

[5] SmartRF05 Evaluation Board User’s Guide : www.ti.com/lit/pdf/swru210a

[6] CC2431DK Development Kit User Manual Rev. 1.5 : www.ti.com/lit/pdf/swru076d

[7] F. Subhan, et al., “Indoor Positioning in Bluetooth Networks using Fingerprinting and Lateration approach”, *International Conference on Information Science and Applications*, April 2011, pp. 1-9

[8] Y. Wang, et al., “Bluetooth Positioning Using RSSI and Triangulation methods”, *Consumer Communications and Networking Conference (CNCC),* 2013, IEEE, 11-14 Jan 2013, Las Vegas, USA

[9] Yuanchao Shu, et al., “Gradient-Based Fingerprinting for Indoor Localization and Tracking”, *IEEE transactions on industrial electronics*, Vol 63, No. 4, April 2016

[10] A. Thottam Parameswaran, et al., “Is RSSI a Reliable Parameter in Sensor Localization algorithms-An Experimental Study”, *Field Failure Data Analysis Workshop (F2DA’09),* New York, 2009

[11] Eric Kim, “DeepBLE – Localized navigation using Low Energy Bluetooth”, Senior Design project, 2013-2014, Dept. of CIS, University of Pennsylvania, Philadelphia, PA

[12] Daan Scheerens, “Practical Indoor Localization using Bluetooth”, Master Thesis, Informations Systems Engineering, University of Twente, January 2012

[13] Georgia Ionescu, et al., “Improving Distance Estimation in Object Localization with Bluetooth Low Energy”, *SENSORCOMM 2014 : The Eighth International Conference on Sensor Technologies and Applications*

[14] L. Pei, et al., ”Inquiry-based Bluetooth Indoor Positioning via RSSI Probability Distributions”, *Second International Conference on Advances in Satellite and Space Communications*, 2010, pp. 151-156

[15] L. Pei, et al., “Using Inquiry-based Bluetooth RSSI Probability Distributions for Indoor Positioning”, *Journal of Global Positioning System*, Volume 9, Issue 2, 2010, pp. 122-130

[16] Silke Feldmann, et al., “An indoor Bluetooth based positioning system: Concept, implementation and experimental evaluation”, *International Conference on Wireless Networks*, 2003, pp. 109-113

[17] Pavel Kriz, et al., “Improving Indoor Localization Using Bluetooth Low Energy Beacons”, *Mobile Information Systems*, Volume 2016, Article ID 2083094, http://dx.doi.org/10.1155/2016/2083094

[18] RFD8500 RFID developer guide, MN002222A02, Revision A, March 2016

[19] RFD8500 RFID user guide, MN002065A02, Revision A, March 2016