TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

INDOOR LOCALIZATION SYSTEM FOR UNMANNED AERIAL VEHICLE

Master of Science Thesis

Examiner: Prof. Kari T. Koskinen
Examiner and topic approved by the Dean of the Faculty of Engineering Sciences on 28th March 2018

ABSTRACT

TAOUFIKUL ISLAM: Indoor Localization System for Unmanned Aerial Vehicle.
Tampere University of technology
Master of Science Thesis, 34 pages, 14 Appendix pages
March, 2019
Master's Degree Programme in Automation Engineering
Major: Factory automation & industrial informatics.
Examiner: Professor Kari T. Koskinen

Keywords: Indoor localization, Zigbee, XBee, Trilateration, Wireless sensor network.

Industries have gone through several stages of changes. Over the flow of time, automation and robotization in industries have become a big concern to maximize the production and minimize the time. Autonomous mobile robots have become very popular among industries because of its mobile nature. One variant of autonomous mobile robot is Unmanned Aerial Vehicle. UAV provides a wide scope of utility inside industrial environment. However, navigation of autonomous mobile robots in indoor environment is still a big concern.

The thesis has tried to develop a methodology for indoor localization that can be used for those autonomous mobile robots such as UAV. The concern of the thesis is to develop a methodology for indoor navigation that is stable, less electrical power and computational power demanding. First, the thesis studied state of the art existing solution related to the topic. Then the thesis has tried to select a potential approach to develop a methodology. After that the thesis has developed a methodology and implemented the methodology in real life lab environment. Finally, the thesis has discussed the result of the developed methodology.

The thesis mainly focused on developing an indoor localization methodology that can be used for UAV. Hence, implementation of the developed methodology has not been discussed in the thesis.

PREFACE

The research presented in this thesis has been carried out at Mechanical Engineering & Industrial Systems (MEI) at Tampere University of Technology between June 2018 to November 2018.

First, I would like to thank me because of my high hopes and patience to come this far. I am grateful to me being able to accomplish the dream of pursuing master's degree in Finland and all the sequential and contentious hard work required for that.

I present my sincere gratitude towards my thesis advisor Dr. Jussi Aaltonen. I have always found him as supportive and informative. During the total thesis research period his wisdom and guidance was my inspiration and motivation.

I am also thankful to my thesis supervisor Prof. Dr. Kari T. Koskinen. I find it as a subject of honor to do some research with prof. Kari.

I would like to conclude my preface speech by thanking my parents and my sisters. I am forever grateful to them for their continuous support and sacrifice toward me.

Tampere, 29.3.2019

Taoufikul Islam

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

APM	Absolute Positioning Method
AP	Access point
BLE	Bluetooth Low Energy
CMRR	Continuous Moving Range Recognition
EMI	Electro Magnetic Interference
e.g	exempli gratia/ for example
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GUI	Graphical User Interface
i.e	id est / that is
IEEE	Institute of Electrical and Electronics Engineers
INS	Indoor Navigation System
IPS	Indoor Positioning System
IR	Infra-Red
KHz	Kilo Hertz
LQP	Link Quality Profile
M2M	Multi point to Multi point
MAS	Multi Agent System
MEM	Map Environment Method
MFP	Magnetic Field Positioning
MLE	Maximum Likelihood Estimation
MMSE	Minimum Square Error Message

OEM	Original Equipment Manufacturer
OSI	Open System Interconnection
P2M	Point to Multi point
P2P	Point to Point
RFI	Radio Frequency Interference
RFID	Radio Frequency Identification
RPM	Relative Positioning Method
RSSI	Received Signal Strength Indicator
SCRR	Swift Communication Range Recognition
SLAM	Simultaneous Localization and Mapping
TDOA	Time Distance of Arrival
TOA	Time of Arrival
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
UWB	Ultra-Wide Band

1. INTRODUCTION

Over the flow of time, industry has gone through several stages of which is commonly termed as “Generation”. Currently, industry is in a state of transition from Industry Generation 3.0 to Industry 4.0. The concept behind industry 4.0 is completely different than all the previous generations. Industry 4.0 introduces the concept of totally decentralized distributed systems & internet of things[1]. In modern industry, autonomous mobile robot has become a very important issue to make the production line fast and economical. One of the biggest challenges for those autonomous indoor mobile robots is indoor navigation.

There are two main types of navigation system, indoor navigation and outdoor navigation. Outdoor navigation is relatively well developed because of the Global Navigation Satellite Systems (GNSS) /Global Positioning System (GPS) whereas Indoor navigation is way more challenging because of the nature of the navigation environment (indoor places) and the absence of GNSS/GPS.

Unmanned Aerial vehicle is the recent addition in industrial use. The scope of the UAV is ultra-wide. From the rescue to the monitoring, UAV provides an agile and faster support. UAV is a topic of choice in different industrial application because modern UAV are very small, lightweight and can go to a place what is very difficult or even impossible for a wheeled robot.

Due to the advancements of semiconductor and electrical devices, a lot of compliances are available for small sized UAV what gives opportunity of doing more complex operation using UAV. Due to the size and weight, UAV can be used in the indoor environment as well.

Most of the autonomous UAV uses GPS/GNSS system for their primary navigation system. However, as discussed previously, GNSS based navigation is not the solution for indoor environment. Many UAV uses advanced technology for navigation purpose in indoor environment. The result is very good and very accurate. Machine vision or image processing or SLAM devices are mainly used in this kind of indoor navigation solution. The biggest drawback of these kind of solutions is that they are very expensive to deploy, needs a lot of computational power and draws a lot of energy. Computational power and energy are a big concern for small sized UAV. The flight control system by itself is very computational power demanding. Adding some heavy computational power demanding INS/IPS algorithm based on machine vision, SLAM, or image processing limits the flight time and programming scopes for extra task of the UAV.

The thesis has developed an Indoor Localization methodology using Zigbee based WSN for Unmanned Aerial Vehicle. The idea is to develop a smart WSN based localization and navigation system what requires less computational power, less electrical power, stable, provides secure and fast enough data for locating and navigating a fast-moving object like UAV. The thesis also tries to implement Multi Agent System in localization and navigation process. The thesis will only focus on developing the methodology hence the implementation of the method will not be discussed or presented in the thesis.

2. LITERATURE REVIEW

There exist no universal standard for Indoor Navigation System(INS) or even Indoor Positioning System (IPS) , no standard has been specified by the Institute of Electrical and Electronics Engineers(IEEE)[2]. Researchers have been trying to come up with different ideas to make indoor navigation or positioning as smooth and agile as outdoor navigation system. Some promising methodologies and techniques exist at present.

2.1 Indoor positioning methods

There are mainly three kinds of widely accepted techniques for indoor positioning.

The first type is Relative Positioning Method (RPM). This is the cheapest option where the relative position of the robot is taken from the onboard sensor and then compared to the pre-defined reference position. A simple line follower robot with a rotary encoder is an example of this kind of technique. The initial position is taken from a known reference point, then the distance travelled & direction is determined based on the reading from the rotary encoder. The main disadvantages of this method are that, the errors get accumulated with distance and the errors increases at an exponential rate. One solution to minimize the error is to compare the value with intermediate reference point what makes the process complex and in efficient.

Absolute Position Method (APM) is the second type. This method stays in the middle position in terms of accuracy, cost and performance. The robot acts as beacon and its position / presence is determined by some deployed sensor network. This kind of method often uses a wireless sensor network and there is always a beacon (active/ passive) what is being detected by the wireless sensor network. A good example of this method is asset tracking using RFID tags. To track the inventory, a RFID tag is being placed on the target object. In this method, whole wireless sensor network is pre-defined and every WSN node has its pre-defined address and position. Thus, when a WSN node detects the target beacon, the result is compared with the detecting RFID node and the position of the target beacon is determined.

Finally comes the Map-Environment Method (MEM). This is the most expensive method with the highest accuracy, but this is also the most demanding in terms of resource and cost. This method uses SLAM devices or camera and powerful processor what can do machine vision and image processing, for example smart phone or single board computer.

Autonomous navigation using SLAM can be a good example of this method. The mobile robot uses the SLAM device to map the environment. Then the results obtained by the

SLAM device is being compared with the pre-defined class and category. The pre-defined class and category is a user defined algorithm to detect and classify the environment. For example, what is wall or obstacle and what is path. Then based on the pre-defined algorithm the robot tries to create its own trajectory to achieve the end goal. Navigation using camera and machine vision uses the same principal.

Throughout this research, focus will be put on the middle path i.e The Absolute Positioning Method (APM). Many researchers have already developed indoor navigation technique using the Absolute Positioning Method.

2.2 Indoor positioning using radio frequency technologies

Wireless Local Area Network or in short Wi-Fi is a popular method in digital communication. It follows IEEE 802.11 standard. The Wi-Fi technology fully supports OSI model hence a great number of things can be done using this technology. It is possible to get the RRSI values of a Wi-Fi transmitter what gives a scope to use this technology for IPS/INS. Several researches have been conducted to develop methodology and technique for INS/IPS using Wi-Fi. Ultra-Wide Band (UWB) has also been tested. Both showed some pros and cons. In their research work, Omar et al. has proposed a methodology to use Wi-Fi for indoor positioning[2]. In their research, they showed that their developed Dynamic Access point & Finger printing method has better accuracy and scalability than standard Wi-Fi access point finger printing method for IPS. A dynamic access point fingerprint is the sum of fixed Wi-Fi access point and temporary access point by the user. The proposed method uses the AP feature of the smartphones to make the dynamic access point. In addition, the accelerometer of the smartphone has been used to predict the movement patterns of a person to detect static or dynamic with the help of machine learning algorithm. This proposed method used MaxMean access point selection strategy to select a set of access points to improve the accuracy of the result. They have also concluded that 5 dynamic access point fingerprint gives the most promising result and the accuracy reduces with the increment of dynamic access point fingerprinting. The mobile hot spot or access point feature of modern smart phone has been used in the research, hence it also falls in the map-environment mapping method for IPS/INS.

Ultra-Wide Band (UWB) is another new way for Indoor Positioning or Indoor Navigation System. UWB falls in IEE 802.15.3a standard. Mentionable advantage of UWB is that it is immune to interference and has better penetration power for signal travel for example through concrete walls[3]. There are several ways for IPS/INS using UWB. Mainly Time of Arrival (TOA) and Time Difference of Arrival (TDOA) between sent and received signal has better efficiency among the rest of the other technique[3]. In their research, Ren et al. has done a comprehensive study on existing on different methodology for UWB based IPS/INS. They have made a special antenna and used TOA method. They presented a methodology that has an accuracy of 1.5 cm. In this research, effects of base- station configuration have been studied for UWB. A completely new type of antenna was developed for ToA estimation method that has been used in the research. Four configurations based on 3,4 and 8 base station position has been studied in their work. Least Square (LS) algorithm has been used for TOA calculation. They have concluded that 8 base station configuration gives the most satisfying result.

Bluetooth is a wireless technology that complies to IEE 802.15.1 standard. Key application of Bluetooth is Personal Area Network (PAN). Bluetooth offers low power and short-range application. Also, the number of connections in an established network is very limited. Bluetooth technology is another way that has been taken in research consideration to deploy in Indoor Navigation System (INS) or Indoor Positioning System (IPS). Particularly Bluetooth Low Energy (BLE) devices are commonly used in IPS/INS because, they consume low energy, meaning they can remain operational using a battery. Apple Inc. introduced iBeacon in 2013 that can be used for indoor positioning. This iBeacon can offer proximity sensing by tranceiving a universally unique identifier. Later, Google introduced Eddystone BLE beacons in late 2015 that is an alternative to iBeacons. However, BLE is less reliable and less agile than Wi-Fi because BLE signal not only gets attenuated in presence of Wi-Fi signal but also BLE signal has issues with Multipath propagation and signal fading[4]. In their research, Fernando et al. [4] has studied different techniques of IPS using BLE and has done a comprehensive performance comparison of different filters and algorithm to be used for IPS using BLE devices. They have argued that, triangulation along with dead reckoning and Kalman filter gives accuracy of less than 1 meter for IPS using BLE devices. Throughout their research, they have developed a better methodology for IPS using Google's Eddystone BLE beacons. This research used fingerprinting method to estimate the position along with three machine learning algorithms: Naive Bayes, SVM, Random Forest. They have also confirmed that random forest classifier algorithm gives the most accuracy in their methodology. According to the authors, their method gives satisfying accuracy in room-level.

Radio Frequency Identification (RFID) is very popular in short range identification application. RFID is very cheap to deploy and offers both active or passive tags. RFID has been used for asset tracking for quite long time. The principle is very simple, when those tags get a magnetic field, they get powered and information stored to them can be read and be used for identification. RFID can also be used for IPS/INS.

Some researchers have developed several improved and promising techniques for indoor navigation systems using RFID. For example, in their research paper Emi Nakamori et al. proposed one very accurate method for indoor navigation system using RFID[5]. The authors have conducted a comprehensive study on the Swift Communication Range Recognition (S-CRR) technique and have solved a major problem with the S-CRR method. The authors have also done a comprehensive performance comparison between their developed Continuous Moving Range Recognition (CM-RR) and S-CRR and have proved that their developed method has higher accuracy in tracking both the linear and angular movements of a moving robot. Their proposer CM-RR method showed better result for both straight moving model and in curved moving model. There are 4 steps in their developed method: first, detection of RFID by the long range, then detection of RFID by the short range, after that, re-detection of RFID by the long range, and finally re-detection of RFID by the long range.

2.3 Indoor positioning using Zigbee RF device

Zigbee is a wireless radio that is IEE 802.15.4 compliant. Zigbee has been developed for low power, high level communication to create a wireless personal area network. Main feature of Zigbee is pure mesh configuration.

In addition to different methodologies using different technology for IPS/INS, Zigbee wireless protocol has also been studied. Many researchers have already proposed methodology for IPS/INS based on Zigbee devices. In their research paper, Liu et al. have proposed a Zigbee based methodology for IPS. They have developed a fuzzy algorithm that converts RSSI value to distance. The algorithm used Minimum Square Error Message (MMSE) method with integration of 1-D Gaussian membership Function[6] and achieved relatively better accuracy than bare RSSI calculation.

Some other researchers have done RSSI based IPS using Zigbee devices in slightly time-consuming and relatively complex way. In their research, Dong et al. have proposed a Zigbee based IPS methodology. Their developed methodology has two stages: first one is the training session where RSSI values from static devices are being converted as unique fingerprint database using map matching algorithm. Then in second phase, measuring and converting the RSSI from the querying device and comparing with the database to determine its potion. In second phase the RSSI value has been stabilized using Medium Filter Algorithm[7].

Zhao et al.[8] have also developed a Zigbee based Indoor Positioning System (IPS). Their work is also based on Zigbee Link Quality Profile (LQP), a synonym of RSSI. They have put their effort in developing a cooperative localization algorithm combining Multi-Dimensional Scaling (MDS) and Maximum Likelihood Estimation (MLE).

Schwarzer et al. have also proposed a methodology for IPS using Zigbee devices [9]. They have claimed their methodology to be precise for distance measurement between two Zigbee devices (Nodes). They have resolved the issue of better accuracy using a coherent synthesis measurement and a special signal phase evaluation technique. They argue that their developed technique is immune to multi path propagation error. Their methodology is based on TDOA and they have used a technique to utilize the total 80MHz of Zigbee bandwidth by combining all the 16 channels of Zigbee using their algorithm.

Hikmet et al [10]. have presented a methodology of IPS/INS combining Infra -Red (IR) with Ultra Sonic Sound (USS) sound. In their methodology, they have made their own data frame, where IR is used as reference of start and stop of the frame followed by US

The main difference between the approach taken in this thesis and other methodology is that, this thesis research is trying to integrate a Xbee (Zigbee) infra-structure for the wireless sensor network, modulated Infra-Red for the beaconing and usage of Multi Agent System (MAS) to determine the shortest path. Even though, the thesis tries to develop a methodology for Indoor Navigation System (INS) using Xbee (a wireless device that uses Zigbee protocol), several researchers have already proposed several Zigbee based methodologies for INS/IPS. Most of Zigbee based INS/IPS methodologies are done by Received Signal Strength Indicator (RSSI). It has been found that RSSI gets easily attenuated by several factor for example : humidity on the air , obstacle in between devices (walls or similar things), Electro Magnetic Interference (EMI), Radio Frequency Interference (RFI)[6]. Thus INS/IPS based on RSSI is not a reliable solution. This is the reason why the thesis tries to replace RSSI with Infra-Red (IR) beaconing and tries to validate a better methodology and gained enough motivation to evaluate the result with existing methodologies.

2.4 Indoor positioning using micro electro-mechanical systems

Modern mobile phones are packed with various kinds of sensors what can be used for many applications. Those sensors are usually names as micro electro-mechanical systems or MEMS. Those MEMS provide various kinds of information such as magnetic field, orientation, position, gravity, velocity, acceleration. Most of the recent smart phones offers a digital compass, an orientation sensor, an acceleration sensor, a magnetometer and a step counter sensor. Those sensors can be also used for IPS /INS.

In his master's thesis, Vassilayev [11] introduced an improved way of indoor positioning by integrating some new sensors: step counter and step sensor. The research work mainly focused on indoor positioning and used smart phone as processor. Custom software has been developed to use the step counter sensor for indoor navigation purpose. Pedestrian Dead Rocking (PDR) has been used in the application and the noise has been reduced using a Kalman filter. The developed application, RunTracker was being integrated with

Google maps using Google Android SDK. About 35 meters of errors was reported during 3 tests on three scenarios what is claimed satisfactory by the author. One professional indoor navigation software named Navizone has been studied in this research and the author claimed better accuracy with his own developed RunTracker application. According to the author, the main bottle neck of the developed method was large distortion due to magnetic field. He has argued that it might be fixed by integrating a compass alongside of the step counter sensor.

Magnetic Field Positioning (MFP) is another excellent way that can be used for Indoor Positioning System (IPS) or Indoor Navigation System (INS). Every place has a unique magnetic field signature with respect to earth’s gravitational point and magnetic field. Hence, a unique database of magnetic field of a building and its indoor places can be made and then it can be used for IPS/INS with the help of devices equipped with magnetometer. Kim et al. [12] has developed an Indoor Navigation System (INS) for mobile robots that uses magnetic field mapping. The authors have argued that adding some rotational and positioning encoder increases the accuracy in INS for a mobile robot. One great problem of Magnetic Field Positioning is that, magnetic field fluctuates widely for a small change or re-arrangements of same elements. For example, numbers of people in a building. Also, electronic devices have effect in changing magnetic field footprint. In their research, P.Binu et al. has argued that the magnetic flux inconsistency can be fixed with Google’s Route Boxer Algorithm[13].They have also developed an INS methodology using MFP. They used android phone to do that and made an android app for that navigation what is only valid for a specific walking patter of a specific person[13]. In their method, first a complete map along with coordinate value is being generated and in second phase the relative position is being read and compared by the sensor of the smart phone.

Meanwhile some other research groups have developed a promising Indoor Navigation System for autonomous robots using a camera and image processing technique. For example, in their research paper “Indoor navigation for mobile robots using memorized omni-directional images and robot’s motion”[14], Tang et al. presented an Indoor Navigation System using 360-degree camera and image processing. Their system works in two stages. In the first stage the robot teaches itself by taking reference image then it compares any instant image with its stored reference images to determine its relative position. The authors have ensured the better accuracy of the method by using matching vertical edge with color information rendering.

Another research group has developed an Indoor Navigation System using SONAR technology. For example , in their research paper “Indoor localization and navigation for a mobile robot equipped with rotating ultrasonic sensors using a smartphone as the robot's brain”[15], Lim et al have proposed a system using a 360 degree SONAR sensor. The robot makes a geometric map of its environment using a 360-degree SONAR sensor array and then navigates itself along with obstacle avoidance algorithm.

3. DESCRIPTION OF THE DEVELOPED SYSTEM

The Indoor Localization System that has been developed in this thesis consists of three parts

3.1 Beacon:

The beacon is the object what will be detected and from detection of the object the navigation path will be generated. In this thesis, an active beacon has been used. The beacon is an active Infra -Red transmitter. In this thesis, a remote controller of a television has been used (Figure 1) as the transmitter beacon because the Television remote has modulated IR transmitter and the code for the button is standardized by specific protocol. The reason for choosing modulated IR is that it is immune to existing noise and can be used for multipurpose information for the same wavelength of light.

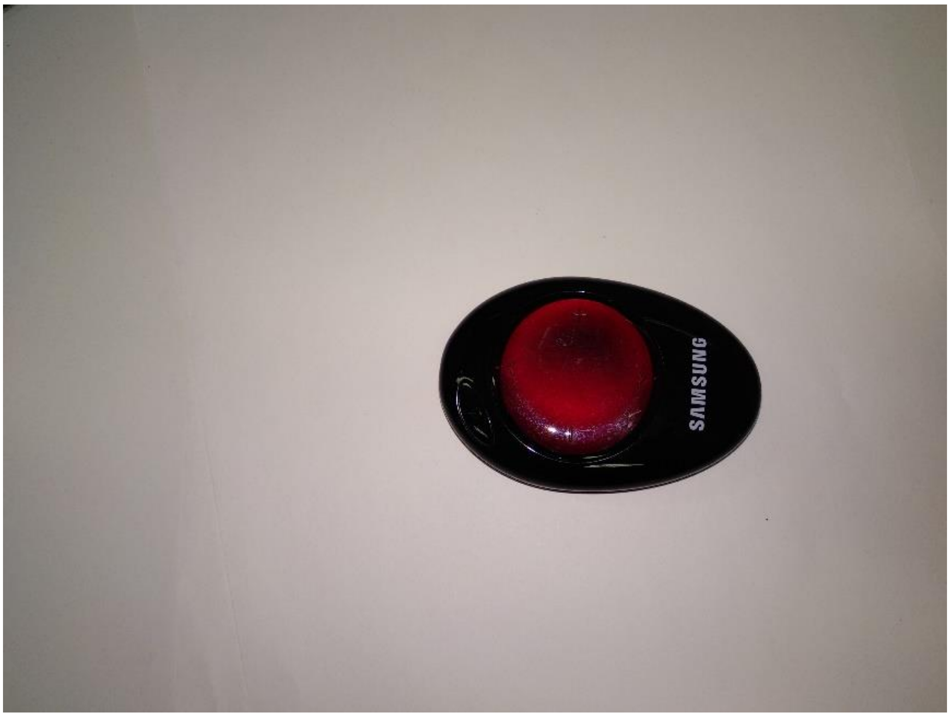

Figure 1. Samsung Television IR remote has been used as beacon.

3.2 XBee Wireless Sensor Network

XBee WSN is the heart of the designed navigation system of the thesis. The WSN is responsible for detecting the beacon and sending data to the computer. Each XBee node is equipped with an IR receiver (Figure 2 & Figure 3). As XBee devices are not user programmable, Arduino had been used to decode the IR message then the decoded message is passed to the Zigbee network.

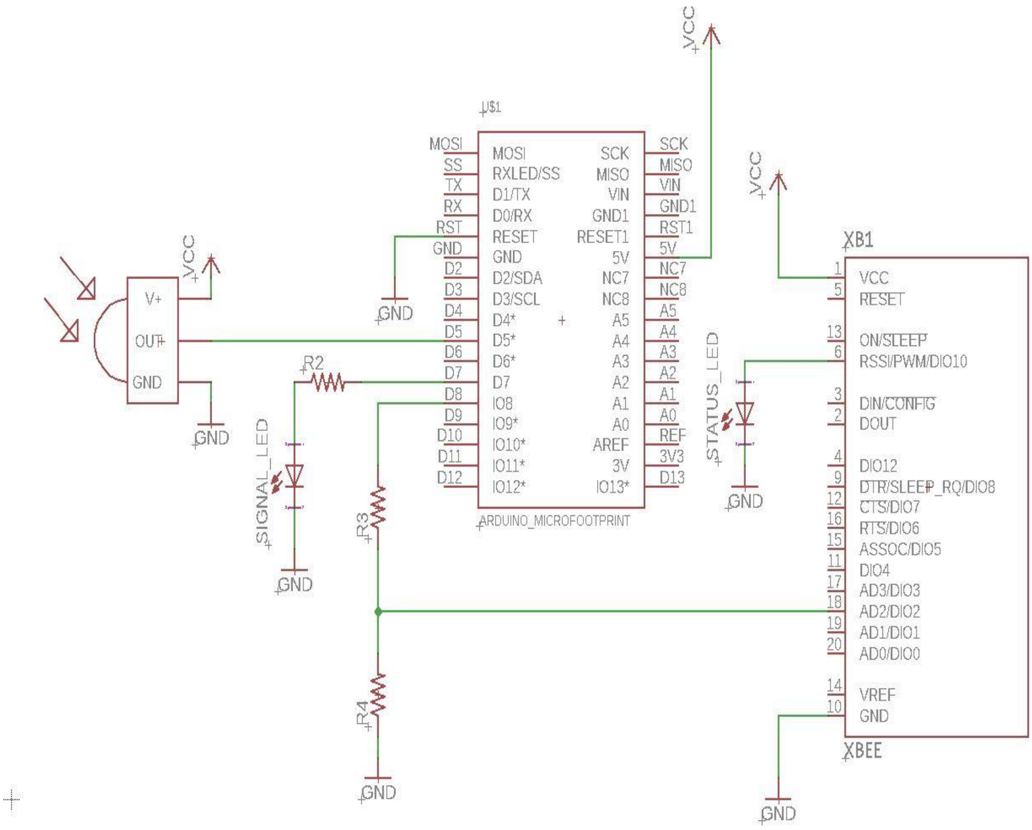

Figure 2. Schematic diagram of the WSN node.

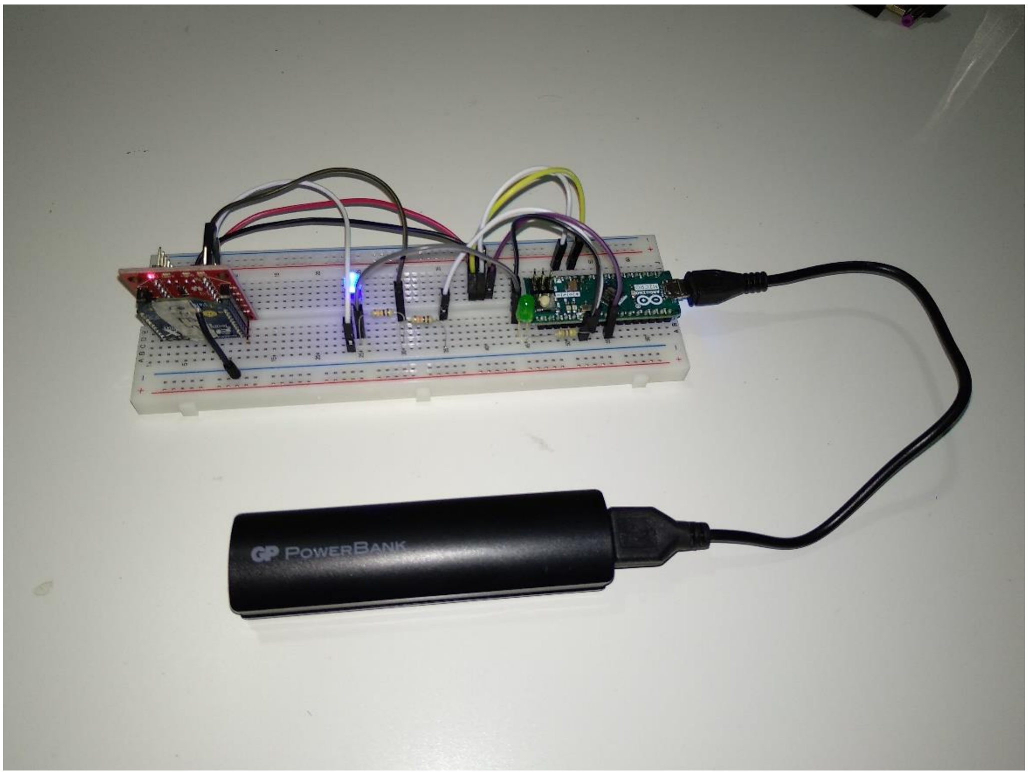

Figure 3. Actual XBee WSN node with power supply.

3.3 Data processing unit

This is where the data frames from the XBee network is being collected and then the data are converted and processed for localization and visualization purpose and most importantly navigation trajectory is made and is sent to the beacon (Figure 6). As the thesis has mainly focused on building a methodology, the implementation part has not been discussed or presented here.

In this data processing unit, the XBee coordinator and a Raspberry Pi single board computer has been used. With the help of XBee coordinator the XBee API data frames are being collected (Figure 4). The XBee supports only UART communication protocol. So, to collect data from the Xbee coordinator, an UART to USB converter has been used (Figure 5). The Raspberry Pi has two programs that accomplishes all the data processing. First program collects data from the USB port and parse it for http JSON response.

The second program is a java script what takes the JSON response and updates the Graphical User Interface (GUI).

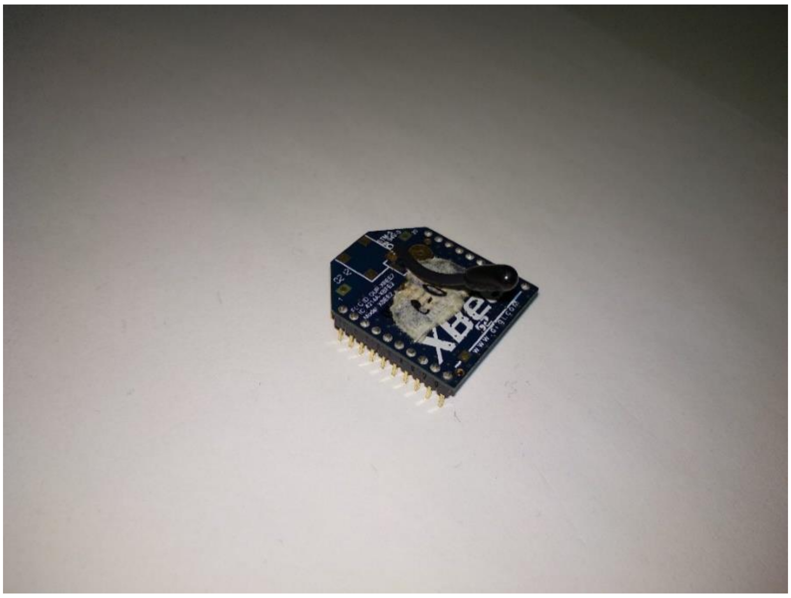

Figure 4. An XBee device manufactured by Digi International.

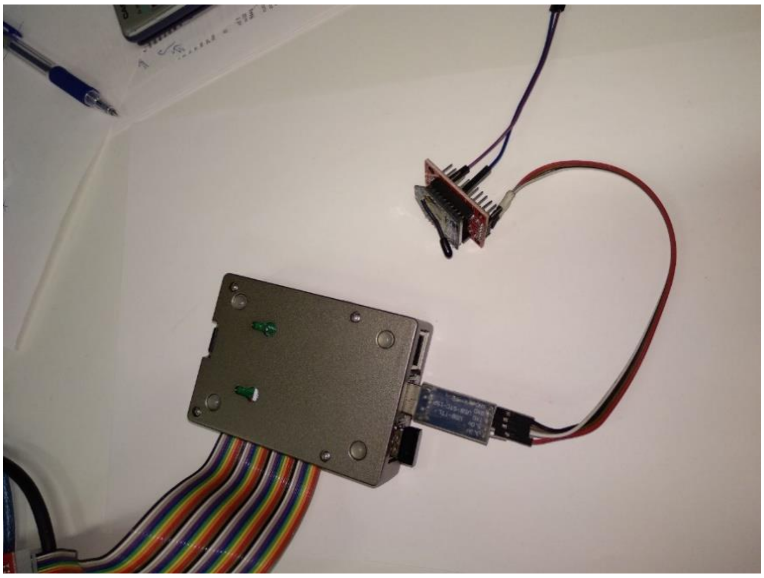

Figure 5. XBee coordinator connected to Raspberry Pi via USB to UART converter.

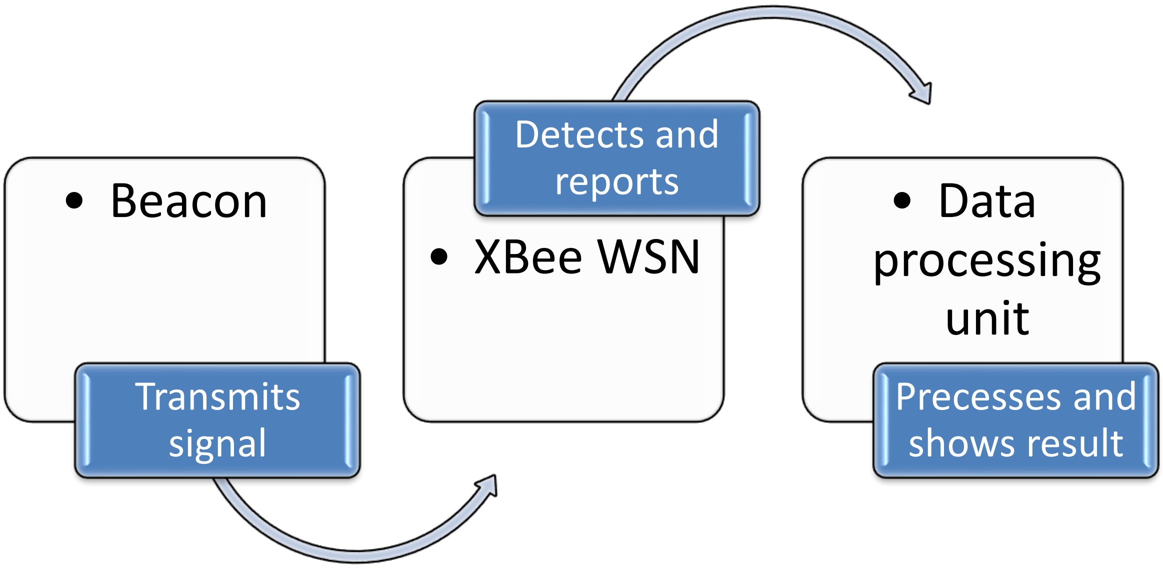

Figure 6. System architecture of the developed system.

3.4 Setup of the test environment:

The developed system has been deployed and tested in the MEI department of Tampere University of Technology. The GUI of the developed indoor navigation system (Figure 8) has been made according to the office architecture of the MEI department's RA room (Figure 7). The area has been chosen intentionally because it has obstacle, a door and thick wall. That allows the thesis to evaluate the performance of the developed system in complex situation. The thesis mainly focused on evaluating the possibility of indoor navigation using XBee devices. In addition to that the thesis worked on to find what kind of network configuration gives the best result, discovered whether the sensor network should be a smart WSN or just a normal WSN, should the communication between beacon and WSN be a simplex, half-duplex or full duplex communication.

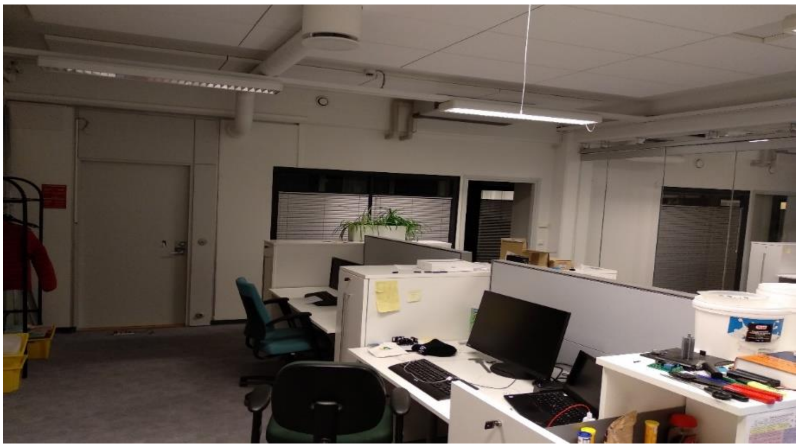

Figure 7: Image of the office room where the system has been tested

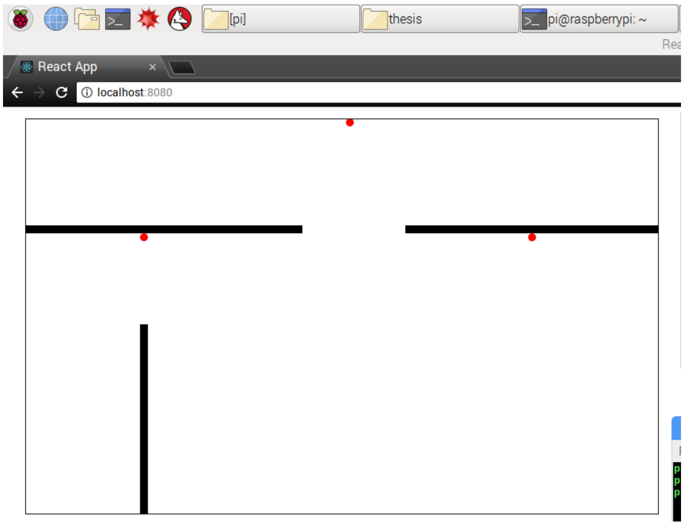

Figure 8: Graphical representation of the office layout.

For the data processing unit two application has been developed. First, the application named “Finder” runs in the raspberry pi (Figure 9). The job of the application is to collect the XBee API frame and parse it. After parsing the API frame, it publishes a JSON response to http port.

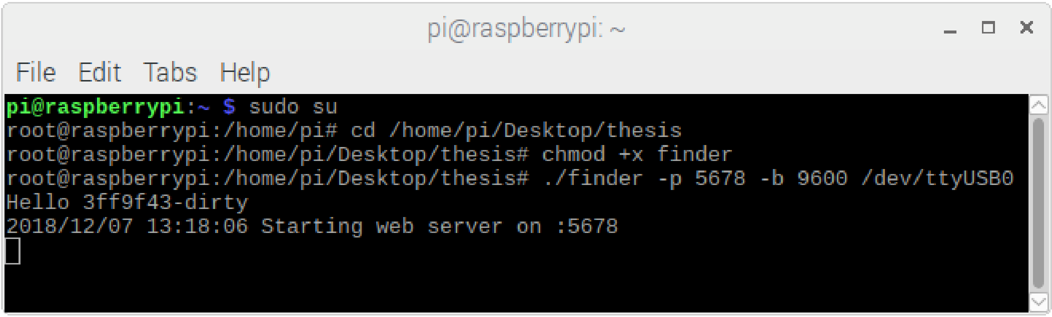

Figure 9: “Finder” application running in the Raspberry pi

Then we need to run the second application named “Build”. The function of the application is to request the parsed XBee API frame and update the GUI map based of the parsed information (Figure 10).

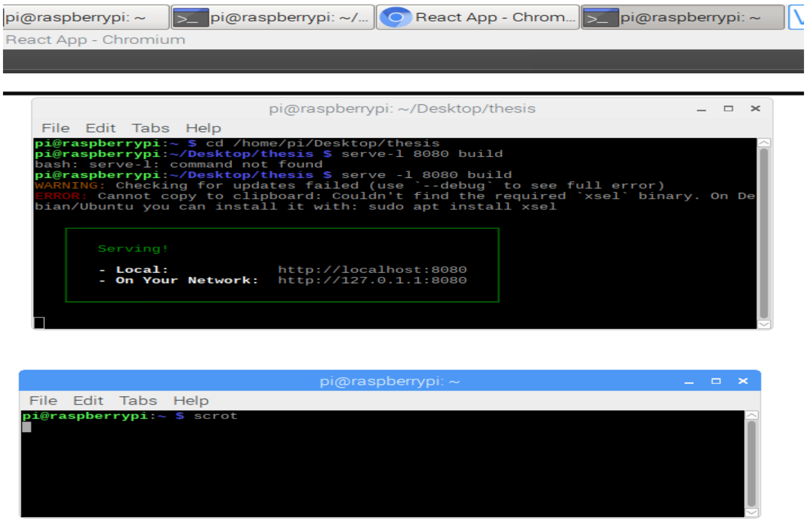

Figure 10: “Build” application running in the Raspberry Pi

4. SET THEORYTICAL MODEL

The Indoor Navigation System uses multi-lateration technique for localizing the beacon. This thesis used trilateration method to detect the location of the beacon node because only 3 nodes were available. In lateration method, one node represents one circle. The location of the beacon is determined by the intersection area of the circles. Thus, the more circle, smaller the intersection area hence the better locating accuracy. In this thesis VS1838B IR receiver from Elecrow [16] has been used to detect the IR beacon . The IR receiver has a range of about 10 meters in most of the practical situation. Thus, in this thesis each WSN node represents one circle having a radius of 10 meters in GUI side. The layout of the GUI has been made according to the test bed located in the office. The trilateration makes in total 8 zones (Figure 11). That can be explained using Venn diagram.

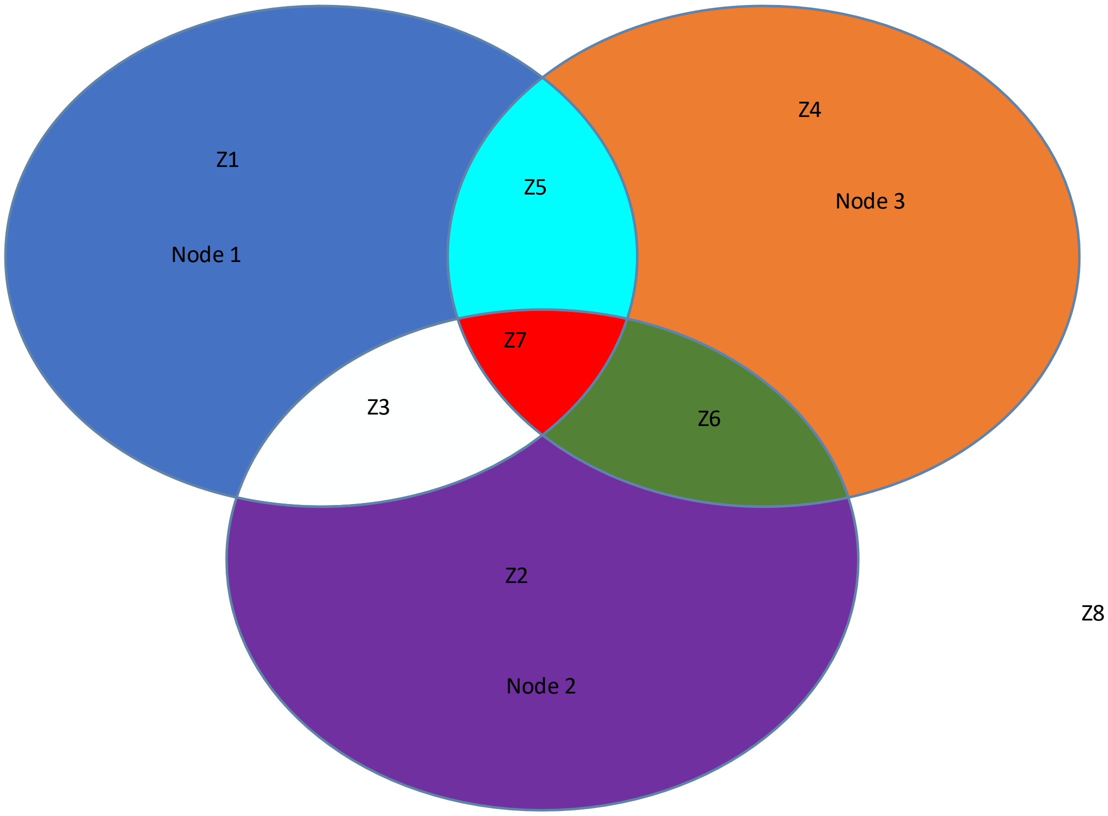

Figure 11. Zones created by the trilateration method.

The Zones can me mathematically modelled as below:

- **Zone 1:** Only Node 1 receives the signal: A-(A∩B)-(A∩C). This area is marked as Z1 in the above Venn diagram.
- **Zone 2:** Only Node 2 receives the signal: B- (B∩A)-(B∩C). This area is marked as Z2 in the above Venn diagram.
- **Zone 3:** Node 1&2 receives the signal: (A∩B)-(A∩B∩C). This area is marked as Z3 in the above Venn diagram.
- **Zone 4:** Only Node 3 receives the signal: C-(C∩A)-(C∩B). This area is marked as Z4 in the above Venn diagram.
- **Zone 5:** Node 1&3 receives the signal: (A∩C)-(A∩B∩C). This area is marked as Z5 in the above Venn diagram.
- **Zone 6:** Node 2&3 receives the signal: (C∩B)-(A∩B∩C). This area is marked as Z6 in the above Venn diagram.
- **Zone 7:** Node 1&2&3 receives the signal: (A∩B∩C). This area is marked as Z7 in the above Venn diagram.
- **Zone 8:** Null set: No node receives the signal: Ǿ. This area is marked as Z8 in the above Venn diagram.

5. NETWORK CONFIGURATION

Network topology refers the way devices are interconnected to each other (Figure 12). Based on the complexity, there are mainly five different types of network topology.

- Daisy chain or Line: In this topology all devices are connected in series to each other. It is least complex. All nodes must be alive in order to make a successful communication from top to bottom.
- Tree: This network arrangement is likely Line structure but with brunches on it.
- Bus Topology: In this network all devices are connected via a shared common bus. The bus may have simplex, half-duplex or full-duplex arrangement.
- Ring: In this network, last device is connected to the first device hence it is a closed loop network topology.
- Star: In this network topology the rest of devices are connected to a central device.
- Mesh network: In this network topology, devices are connected in such a way that communication to rest of the device are possible even if one or more intermediate devices are down.
- Fully connected or pure Mesh: This is the most complex network topology where every node is connected to every node. This is the most complex, expensive, reliable network.

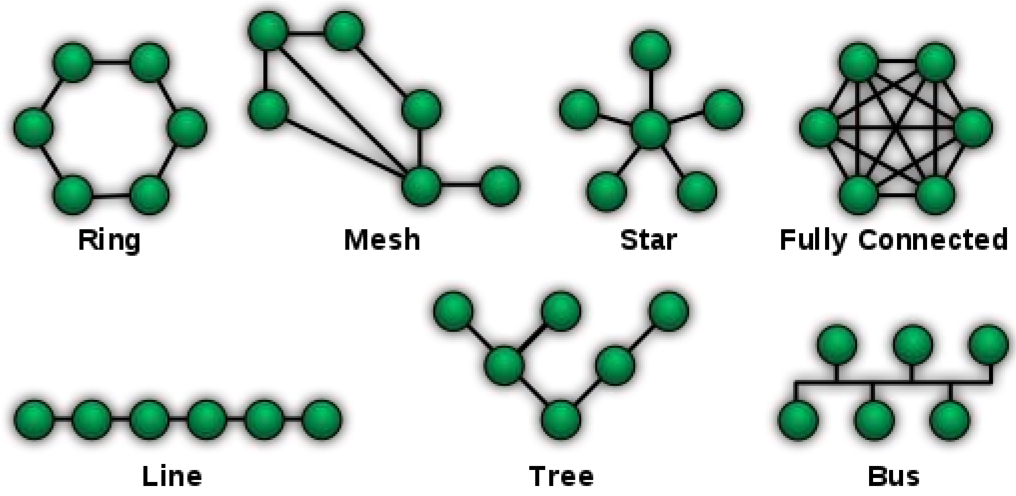

Figure 12. Different network topology[17].

The XBee support pure mesh network topology. The network topology is configurable with in the device by the OEM utility software. The software allows to setup Point to Point (P2P), Point to Multi point (P2M) or mesh and Multi Point to Multi Point (M2M) or pure mesh network topology. This reconfigurable network topology makes XBee a very useful Zigbee device based on application need. The point to point or pair to pair is fastest, point to multi point is faster but provides robust communication. M2M is the most connected network and it is the slowest because every data packet from every node travel through entire network before it arrives to coordinator of the network.

In the research work, P2M network topology has been used rather than M2M topology. The reason for that is the requirement of faster data frame while maintaining the mesh networking.

When configured for data sampling, the XBee devices exchanges data through API frames. Those API frames are unique to XBee modules. XBee coordinator makes an echo of a received API frame to its UART communication port. So, by connecting a USB to UART converter, those XBee API frames can be retrieved (Figure 13).

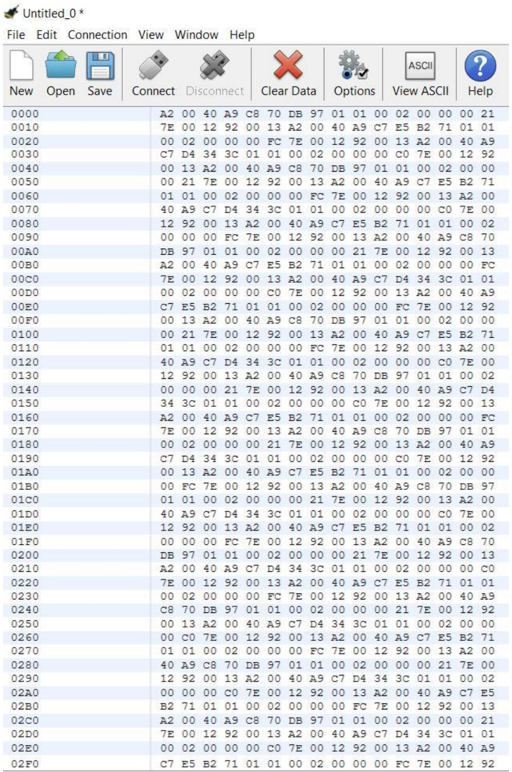

Figure 13. Received XBee data frame from the coordinator.

The detailed explanation of the XBee API frame (Table 1) is given below: In the developed system, there was 3 sensor nodes hence 3 types of XBee data frame is available.

Table 1. Possible data frames from each XBee node.

Node 1 ON	7E 00 12 92 00 13 A2 00 40 A9 C7 D4 B2 71 01 01 00 02 00 00 02 FC
Node1 OFF	7E 00 12 92 00 13 A2 00 40 A9 C7 D4 B2 71 01 01 00 02 00 00 00 FC
Node 2 ON	7E 00 12 92 00 13 A2 00 40 A9 C8 70 B2 71 01 01 00 02 00 00 02 FC
Node2 OFF	7E 00 12 92 00 13 A2 00 40 A9 C8 70 B2 71 01 01 00 02 00 00 00 FC
Node 3 ON	7E 00 12 **92 00 13 A2 00 40 A9** C7 E5 **B2 71 01 01 00 02 00 00 02** FC
Node3 OFF	7E 00 12 92 00 13 A2 00 40 A9 C7 E5 B2 71 01 01 00 02 00 00 00 FC

The breakdown of the XBee API frame from node 3(C7E5) while it got a digital sample data is elaborated below (Table 2):

Table 2. XBee data frame description**.**

Byte number	API frame value in HEX	
0	7E	Start Byte of XBee API frame. It is a defines and fixed delimiter for API frame.
1 2	00 12	16-bit frame length indicator of the incoming packet. This indicator excludes the start byte, frame length indicator and

		checksum. This is underlined in the previous table.
3	92	Frame type. 0x92 = data sample frame.
4	00	64-bit network address. It is unique to each XBee device and can never be changed.
5	13	
6	A2	
7	00	
8	40	
9	A9	
10	C7	
11	E5	
12	B2	16-bit network address assigned to the router and end device by the coordinator while joining the network. It is variable and can be different for same device.
13	71	
14	01	Type of data packet 0x01= Unicast packet, 0x02=Broadcast packet.
15	01	Fixed value defined by Digi.
16	00	Digital channel mask.
17	02	
18	00	Analogue channel mask.
19	00	
20	02	Digital sample data.
21	FC	Checksum.

6. PERFORMANCE EVALUATION

6.1 Results from the developed system

The results get visually presented with the help of GUI in computer screen. Raspberry Pi single board computer has been used for this GUI presentation. In the GUI the red dots represent the node what has not received signal from beacon node. The green dot represents a node what has received signal from beacon. The shaded gray area represents the possible location of the IR beacon. The results of localization technique based on trilateration method is presented below for each zone.

In Zone 1 (Z1), only node 1 is expected to get the signal. Thus, the possible location area would be A-(A∩B)-(A∩C). As we can see in the below picture (Figure 14), the circular gray zone is the area other than the common area of Node 2 and Node 3. This represents possible location of the beacon. We can see the green dot in the shaded gay zone and other two red dots meaning only Node 1 received the beacon signal where Node 2 and Node 3 did not receive any signal from the beacon.

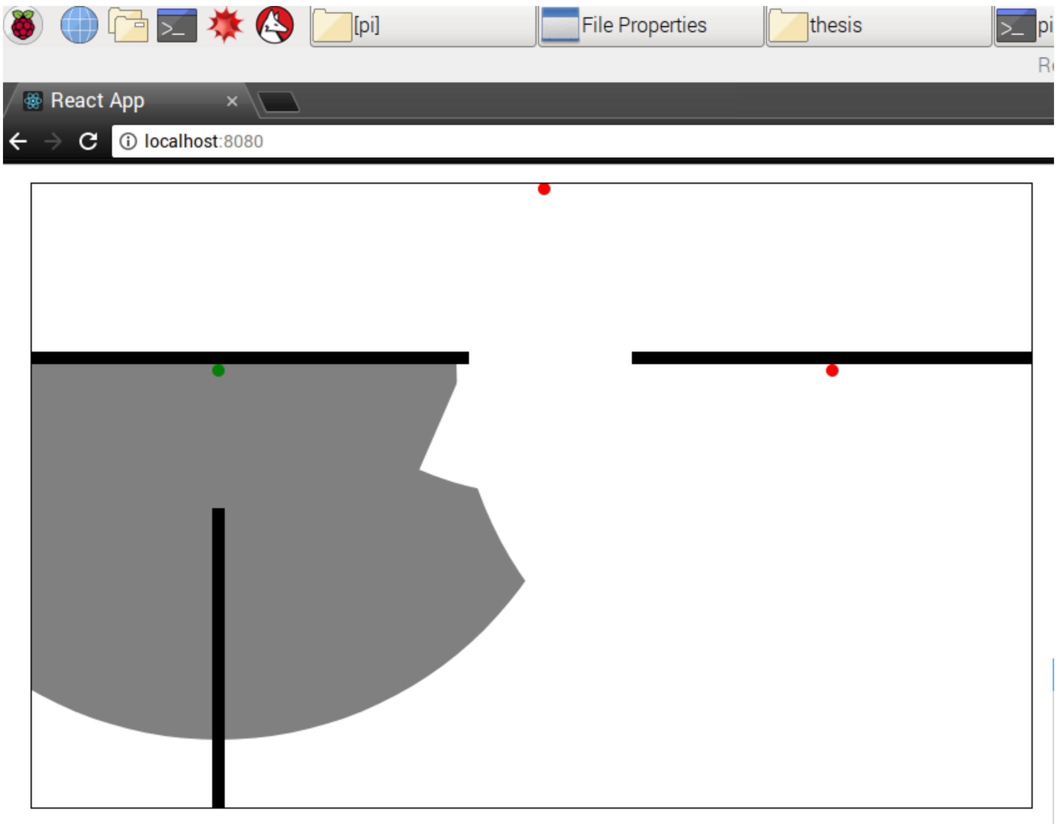

Figure 14. Graphical result of zone 1.

In Zone 2 (Z2), it is expected that only Node 2 will receive the signal. Hence expected result would be the only area that is not common part of Node 1 and Node 3. It can be explained by the set theory with this equation: $B-(B\cap A)-(B\cap C)$. We can see the shaded gray zone with green dot (Figure 15) meaning that only Node 2 has received the beacon signal whereas Node 1 & Node 3 has red dot meaning they have not received any signal from the beacon. The gray zone is the area what represents the possible location of the beacon.

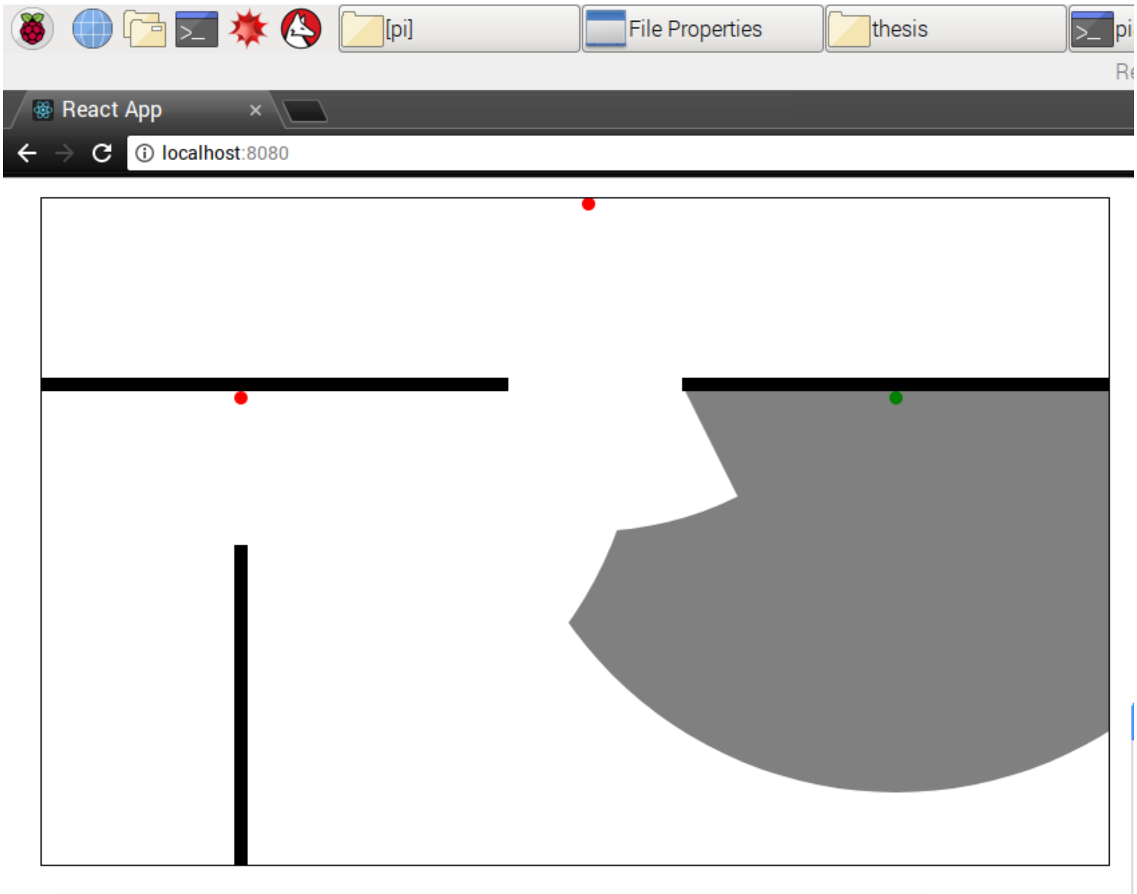

Figure 15. Graphical result for zone 2.

Zone 3 (Z3) is the case where Node 1 & Node 2 receives the signal from the beacon. So, the GUI should make gray zone to the common part of Node 1 and Node 2 except the common part of Node 3. This gray zone is the possible location of the beacon (Figure 16). This area is represented by the following equation: (A∩B)-(A∩B∩C). We can see the green dots for Node 1 & Node 2 meaning they have received the signal from the beacon where as red dot in Node 3 means it has not received any signal from the beacon. Also, we can see gray zone that intersects between Node 1 and Node 2.

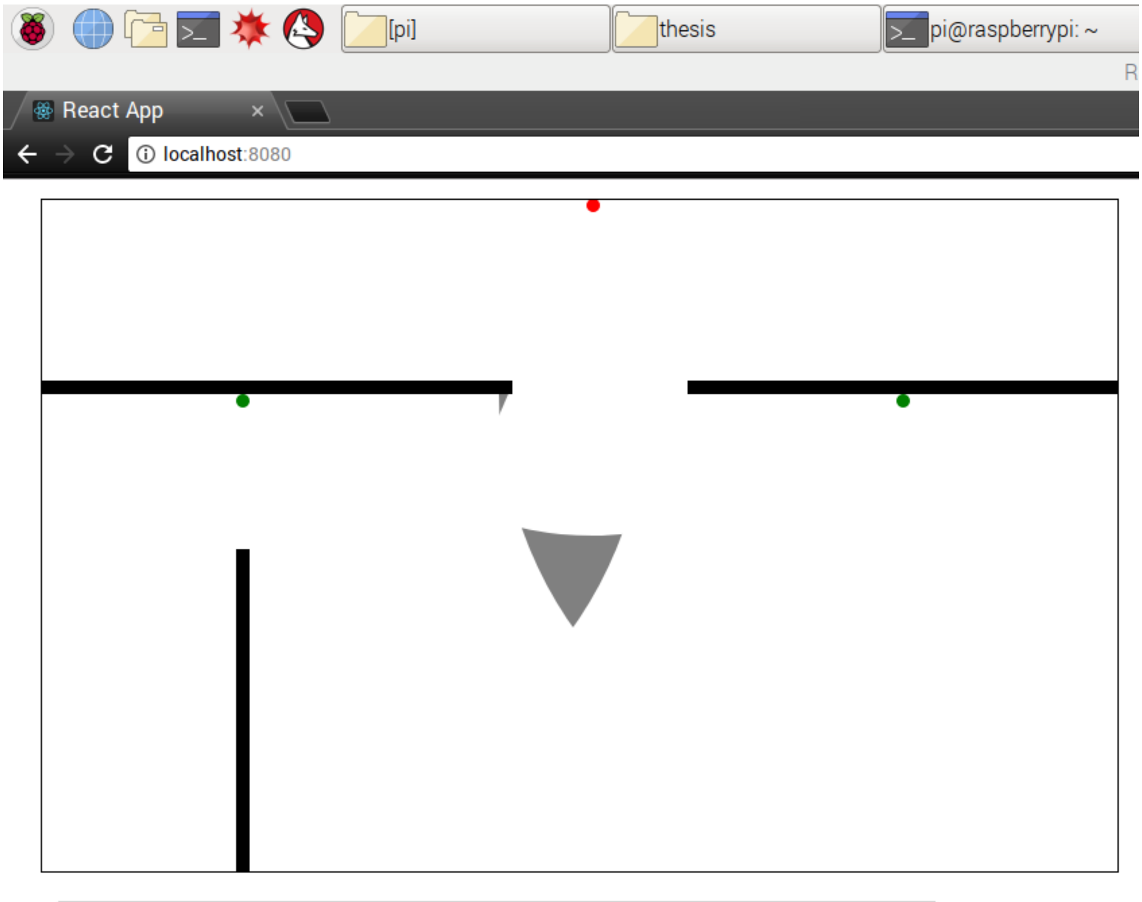

Figure 16. Graphical result for zone 3.

Zone 4 (Z4) is the case where only Node 3 receives the signal from the beacon and Node 1 & Node 2 does not receives any signal from the beacon. The expected gray zone (Figure 17) would the area of the Node 3 what does not intersects with Node 1 and Node 2. This gray zone is the possible location of the beacon. The set representation of the area is: C-(C∩A)-(C∩B). We can see in the picture a half gray circle. This is because the Node 3is attached on a wall and there is no area on the other side of the wall. We can also see a green dot for the Node 3 where as red dot for Node 1 and Node 2.

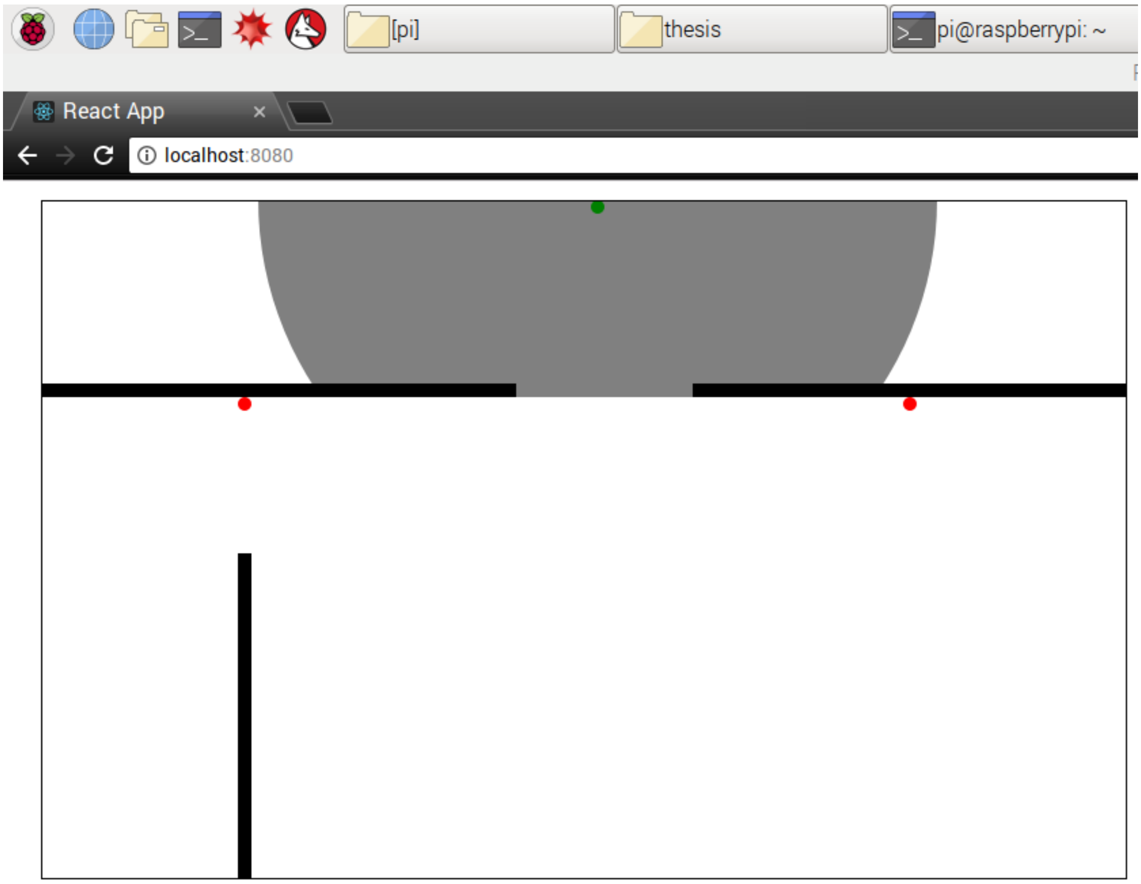

Figure 17. Graphical result of zone 4.

In Zone 5 (Z5) Node 1 & Node 3 receives the signal from the beacon. The result should show the intersecting area of Node 1 and Node 3 except the area common to Node 2. The set equation for this region is: (A∩C)-(A∩B∩C). In the picture (Figure 18) we can see the gray zone that intersect between Node 1 and Node 3 excluding the intersecting area of Node 2. The image also shows green dots to Node 1 & Node 3 while red dot for Node 2.

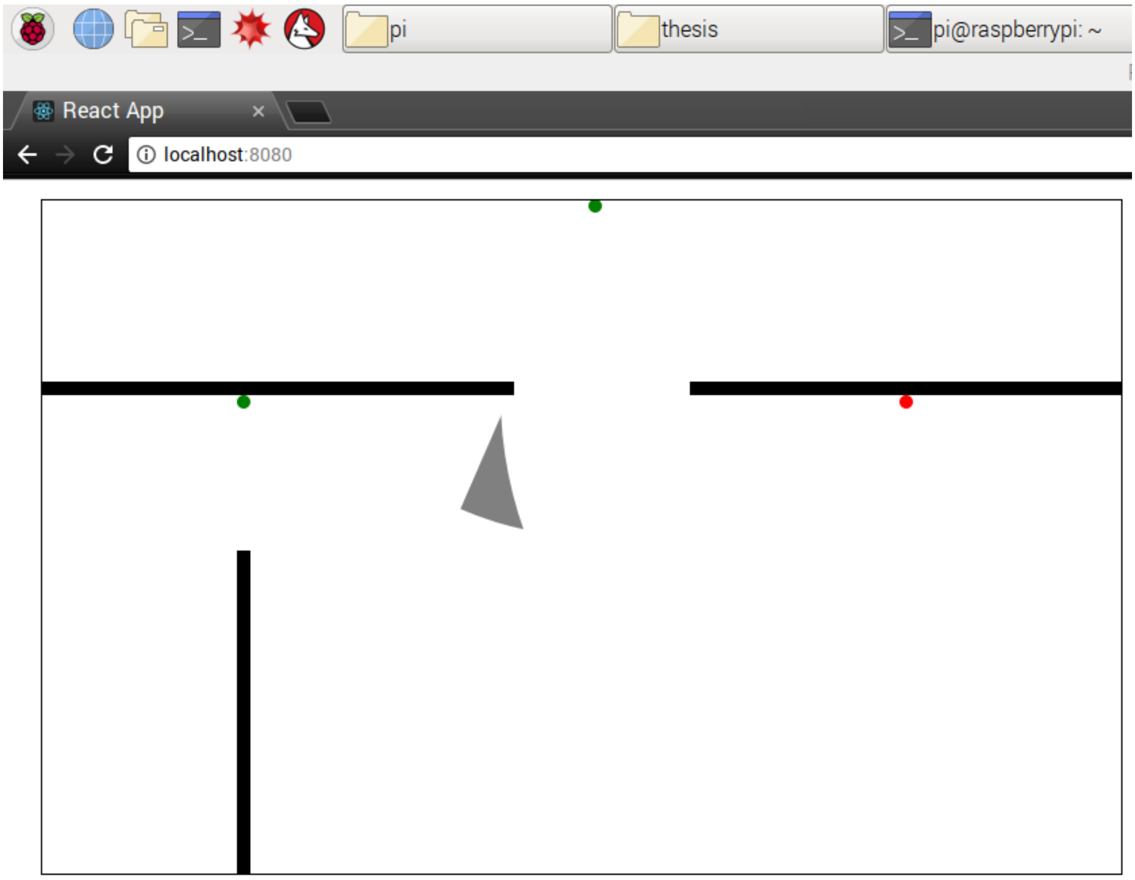

Figure 18. Graphical result of zone 5.

In Zone 6 (Z6) Node 2 and Node 3 receives the signal from the beacon. Hence the gray zone should be the intersecting zone of Node 2 & Node 3 excluding the area of Node 1. The set equation for the area is: (C∩B)-(A∩B∩C). This gray zone is the possible location of the beacon when Node 2 & Node 3 receives the signal. We can see green dots for Node 2 & Node3 where as red dot for Node 1.

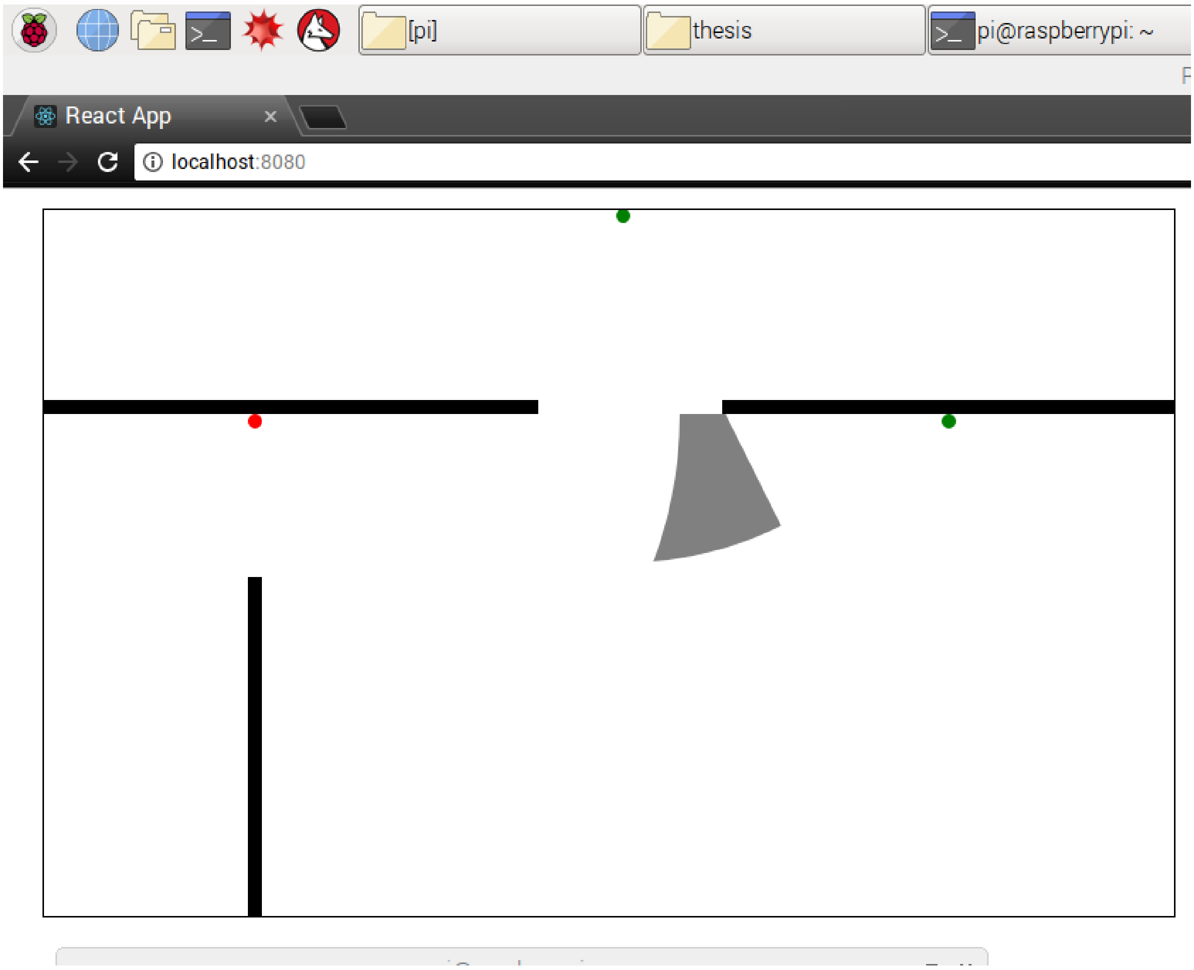

Figure 19. Graphical result of zone 6.

Zone 7 (Z7) is the area when the beacon is in such a position that Node 1, Node 2 and Node 3 receives the signal from the beacon. Theoretically it should the most compact area of the Venn diagram. This is the intersecting area of all the nodes. The set equation for the area is: (A∩B∩C). We can the gray zone is common (Figure 20) to all the Node zone. This time we see all green dots meaning all the node received the signal from the beacon.

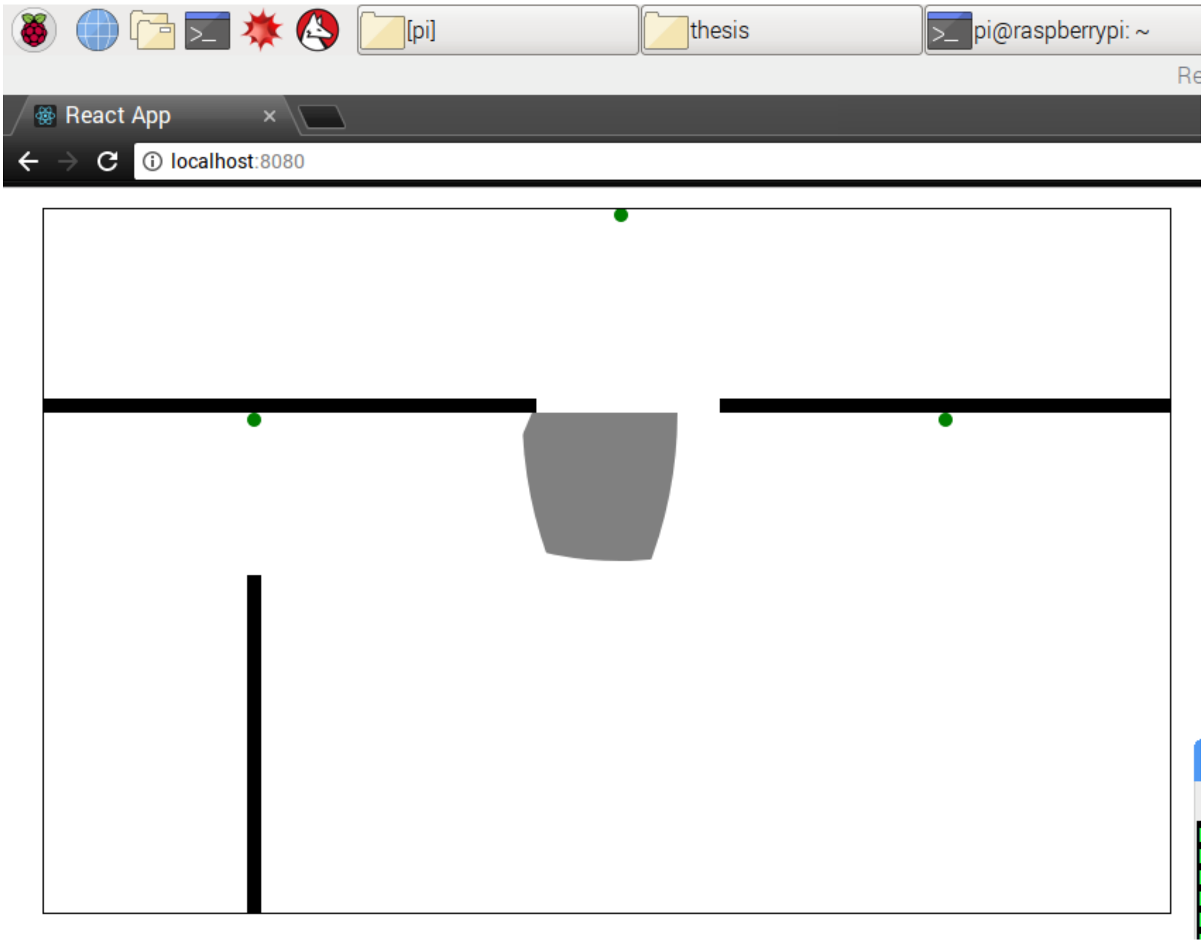

Figure 20. Graphical result of zone 7.

Zone 8 (Z8) is the area where none of the node receives signal from the beacon. In set theory it is identified as a null set: Ǿ. As we can see in the below picture (Figure 21) that all the nodes have the red dot. So, the conclusion is the beacon is not present in the area detectable by the Nodes.

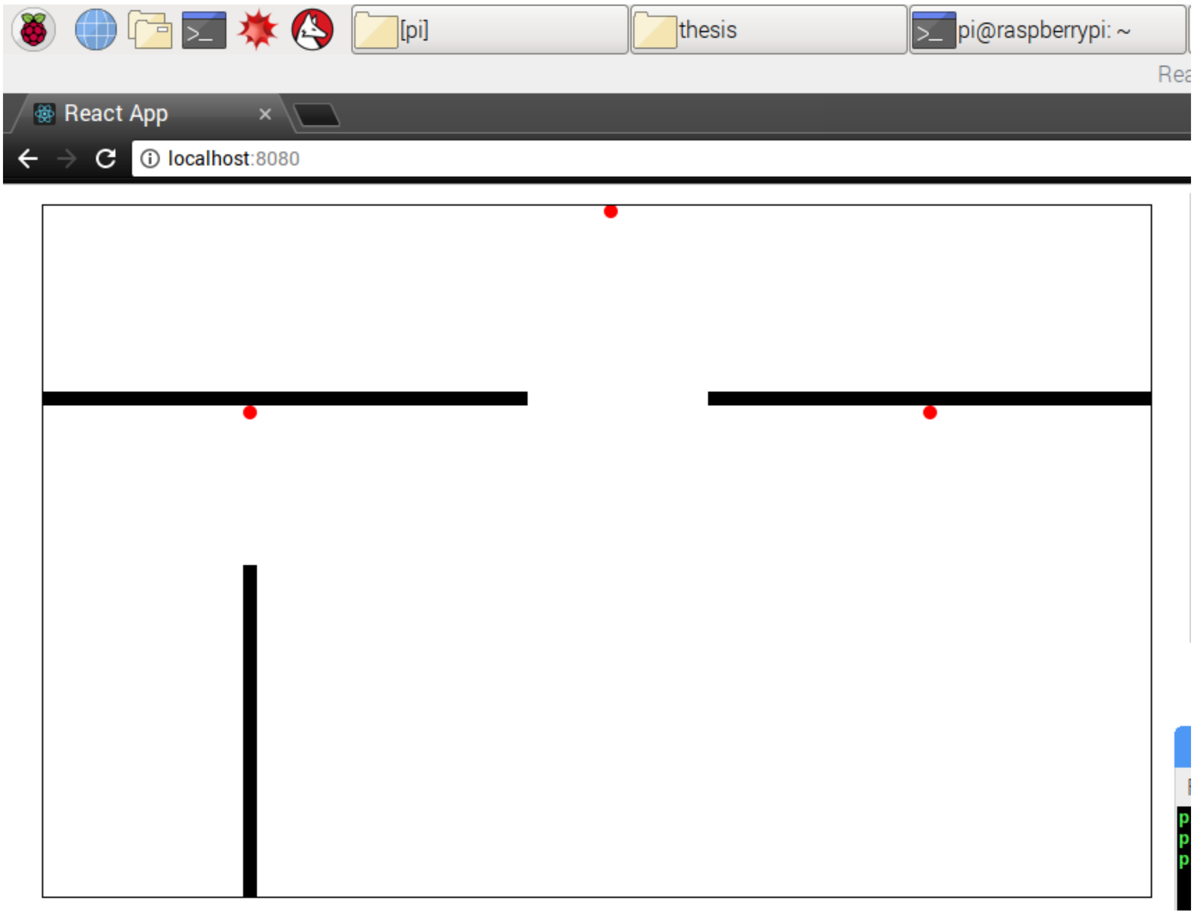

Figure 21. Graphical result of Zone 8.

6.2 Advantages and limitations of XBee in indoor navigation

The thesis has focused on developing a methodology for indoor localization what can be used for unmanned aerial vehicle. One methodology has been developed for indoor localizationusing XBee and IR beaconing. While developing the methodologies some advantage and limitations have been found for indoor localization using XBee devices. Based on the experiments of different configuration and settings, the thesis has found that Multicast setting of the XBee devices gives most promising result for this developed indoor localization method over Broadcast settings. That means mesh network gives optimum result compared to pure mesh network for the developed INS methodology using XBee devices.The developed system can localize the IR beacon and can represent the position of the beacon in the GUI.

The mentionable advantages of XBee for INS/IPS applications are:

- XBee devices provide excellent network stability. XBee has a self-healing networking capability what means any node can goes off and re-join without interrupting the entire network.
- XBee devices offers pure mesh networking.
- XBee devices provides sub networking and hardware encryption of data from device level.
- Unicast (specific addressing of device), Multicast (Group of devices addressing) or Broadcast (addressing to all device of the same network) is available in XBee devices.
- XBee provides 250kbps data rate in API configuration what makes an ideal choice for this INS application.

The considerable limitations of XBee devices for INS/IPS application are:

- While configured to broadcast mode each node allows 3 broadcast frames in every 4 seconds. The data frames have checksum that is being calculated in each node. This setting gives a very slow response required for this IPS methodology.
- 2mW version of the XBee series 2 has been in the thesis. It has been found that those XBee devices has a stable range of approximately 7 meters in most of the real-life experiment.
- XBee devices offers only Digital and Analogue input /output. It features only UART communication protocol what gives it a big bottle neck in terms of communication fearure. Also, the XBee devices are not user programmable that is another mentionable disadvantage.

7. CONCLUSION

An Indoor Positioning methodology for Unmanned Aerial Vehicle has been developed in this thesis. The focus of the thesis was to develop an indoor positioning methodology for UAV using XBee devices because of the low power, low data rate, high security, high network stability and Zigbee mesh networking features provided by XBee devices.

At first the methodology has been mapped. This has been accomplished by investigating state of the art systems and methods available till date related to the topic. Another big factor was the requirements of the thesis. After an intensive research, it has been found that localizing the object by Infra -Red beaconing along with XBee devices gives the satisfactory result required by the thesis. In addition, it has also been found that localizing an object with IR beaconing using Zigbee device is a unique approach compared to the existing methods. It is easily deployable, more accurate, less computational power demanding what makes methodology an energy efficient method for Zigbee based indoor positioning system.

After that the system has been designed and developed. The thesis chooses an active IR beaconing technique for the object to be detected. It has been found by the thesis that modulated IR is highly immune to natural noise for example day light or existing light. The thesis chooses 38 KHz IR remote protocol for IR beaconing.IR remote protocol has unique code value for each button and for each IR remote protocol. It not only gives the benefit of noise immunity but also existence of multiple beacons without interfering any other devices over same channel. The thesis has used SAMSUMG IR remote protocol for the beacon. The WSN node has been developed with an IR receiver that detects the emitted IR by the beacon, an Arduino micro-controller that decodes the IR signal and makes it readable for XBee device and a XBee router/end device what sends data to the XBee coordinator.

Finally. the data collection and processing unit has been developed. This has been accomplished with the help of a XBee coordinator and a Raspberry Pi single board computer. The XBee coordinator collects the XBee API data frames from all the XBee WSN Nodes and sends it to Raspberry Pi for further processing via a USB to UART converter. There are two programs that runs on the Raspberry Pi. One program collects the API frames, checks the error and parse it for the second program. The second program collect the parsed data and updates the result in GUI.

In conclusion, the thesis has successfully developed a Zigbee based indoor positioning methodology for UAV using XBee devices. The thesis has found active IR beaconing with XBee WSN gives the promising result. It has been discovered that multicast setup of XBee (Coordinator-Router-End device) is the ideal network topology for faster data required application like INS/IPS for UAV.

The methodology that has been developed in this thesis needs some external data to provide exact location of the UAV in the map. The developed methodology gives the UAV to locate its position in the area of a certain cell. The more beacon is present in one cell the small the intersecting area, hence the accurate the location. When the altitude of the UAV is known by some altitude sensors and its relative position with respect to surrounding is known with the help of distance sensors, the UAV system gets the ability to position itself to local map and calculate its position in both local and global coordinates using the information of its location on the cell area provided by the developed methodology. In addition to that, there is scope of implement Multi Agent System in WSN node hardware level. The thesis has implemented MAS at software level at the coordinator side. The developed method can be used not only for UAV application but also for any kind of IPS/INS application.

8. REFERENCES

[1] "Industrial revolutions: the 4 main revolutions in the industrial world." [Online]. Available: https://www.sentryo.net/the-4-industrial-revolutions/. [Accessed: 16-Nov-2018].

[2] O. Costilla-Reyes and K. Namuduri, "Dynamic Wi-Fi fingerprinting indoor positioning system," in *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, 2014, no. October, pp. 271–280.

[3] A. Ren, F. Zhou, A. Rahman, X. Wang, N. Zhao, and X. Yang, "A study of indoor positioning based on UWB base-station configurations," *Proc. 2017 IEEE 2nd Adv. Inf. Technol. Electron. Autom. Control Conf. IAEAC 2017*, pp. 1939–1943, 2017.

[4] F. Campana, A. Pinargote, F. Dominguez, and E. Pelaez, "Towards an indoor navigation system using Bluetooth Low Energy Beacons," *2017 IEEE 2nd Ecuador Tech. Chapters Meet. ETCM 2017*, vol. 2017–Janua, pp. 1–6, 2018.

[5] E. Nakamori *et al.*, "A new indoor position estimation method of RFID tags for continuous moving navigation systems," *2012 Int. Conf. Indoor Position. Indoor Navig. IPIN 2012 - Conf. Proc.*, no. November, 2012.

[6] F. T. Liu, C. H. Chen, Y. C. Kao, C. M. Hong, and C. Y. Yang, "Improved ZigBee module based on fuzzy model for indoor positioning system," *Proc. 2017 IEEE Int. Conf. Appl. Syst. Innov. Appl. Syst. Innov. Mod. Technol. ICASI 2017*, pp. 1331–1334, 2017.

[7] Z. Dong, C. Mengjiao, and L. Wenjuan, "Implementation of indoor fingerprint positioning based on ZigBee," *Proc. 29th Chinese Control Decis. Conf. CCDC 2017*, pp. 2654–2659, 2017.

[8] Y. Zhao, L. Dong, J. Wang, B. Hu, and Y. Fu, "Implementing indoor positioning system via zigbee devices," *Conf. Rec. - Asilomar Conf. Signals, Syst. Comput.*, no. 4, pp. 1867–1871, 2008.

[9] S. Schwarzer, M. Vossiek, M. Pichler, and A. Stelzer, "Precise Distance Measurement with IEEE 802 . 15 . 4 (ZigBee) Devices," vol. 4, no. 89, pp. 779–782, 2008.

[10] H. Yucel, R. Edizkan, T. Ozkir, and A. Yazici, "Development of indoor positioning system with ultrasonic and infrared signals," *INISTA 2012 - Int. Symp. Innov. Intell. Syst. Appl.*, pp. 1–4, 2012.

[11] A. Vassilyev, "Indoor Navigaton on Smartphones," no. April, 2014.

[12] H. S. Kim, W. Seo, and K. R. Baek, "Indoor positioning system using magnetic

field map navigation and an encoder system," *Sensors (Switzerland)*, vol. 17, no. 3, 2017.

[13] P. K. Binu, R. Akhil Krishnan, and A. Pranav Kumar, "An efficient indoor location tracking and navigation system using simple magnetic map matching," *2016 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2016*, 2017.

[14] L. Tang and S. Yuta, "Indoor navigation for mobile robots using memorized omni-directional images and robot's motion," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, no. October, pp. 269–274, 2002.

[15] J. Lim, S. Lee, G. Tewolde, and J. Kwon, "Indoor localization and navigation for a mobile robot equipped with rotating ultrasonic sensors using a smartphone as the robot's brain," *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2015–June, pp. 621–625, 2015.

[16] "full-text." [Online]. Available: https://www.elecrow.com/download/Infrared receiver vs1838b.pdf.

[17] "network topology - Google Search." [Online]. Available: https://www.google.com/search?q=network+topology&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwimkvGXp9rfAhUMlIsKHeaTD-MQ_AUIDigB&biw=1536&bih=763#imgrc=vRdBpzigMdP0wM: [Accessed: 07-Jan-2019].

9. APPENDIX

Arduino Code for IR remote manufacture detection:

```
// Include IR Remote Library by Ken Shirriff
#include <IRremote.h>

// Define sensor pin
const int RECV_PIN = 4;

// Define IR Receiver and Results Objects
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup(){
  // Serial Monitor @ 9600 baud
  Serial.begin(9600);
  // Enable the IR Receiver
  irrecv.enableIRIn();
}

void loop(){
  if (irrecv.decode(&results)){
      Serial.println(results.value, HEX);
      switch (results.decode_type){
        case NEC:
          Serial.println("NEC");
          break;
        case SONY:
          Serial.println("SONY");
          break;
        case RC5:
          Serial.println("RC5");
          break;
        case RC6:
          Serial.println("RC6");
          break;
        case DISH:
          Serial.println("DISH");
          break;
        case SHARP:
          Serial.println("SHARP");
          break;
        case JVC:
          Serial.println("JVC");
          break;
        case SANYO:
          Serial.println("SANYO");
          break;
        case MITSUBISHI:
          Serial.println("MITSUBISHI");
          break;
        case SAMSUNG:
          Serial.println("SAMSUNG");
          break;
        case LG:
          Serial.println("LG");
          break;
        case WHYNTER:
          Serial.println("WHYNTER");
          break;
        case AIWA_RC_T501:
```

```
            Serial.println("AIWA_RC_T501");
            break;
          case PANASONIC:
            Serial.println("PANASONIC");
            break;
          case DENON:
            Serial.println("DENON");
            break;
        default:
          case UNKNOWN:
            Serial.println("UNKNOWN");
            break;
        }
      irrecv.resume();
 }
}
```

Arduino Code for IR beacon:

```
// Include IR Remote Library
#include <IRremote.h>

// Define sensor pin
//Any digital pin will work
const int RECV_PIN = 4;

// Define LED pin constants
//led to show the result of the IR
//In our case we need to hold the LED for 5S wwhen we get the IR
const int redPin = 2;

// Define integer to remember toggle state
int togglestate = 0;

// Define IR Receiver and Results Objects
IRrecv irrecv(RECV_PIN);
decode_results results;


void setup(){
  // Enable the IR Receiver
  irrecv.enableIRIn();
  // Set LED pins as Outputs
  pinMode(redPin, OUTPUT);
  //pinMode(yellowPin, OUTPUT);
}


void loop(){
    if (irrecv.decode(&results)){

        switch(results.value){
        case 0XE0E040BF: //This is the Power button  value for SAM-SUNG IR remote protocol
        // Turn on LED for 2 Seconds
        digitalWrite(redPin, HIGH);
        delay(5000);
        digitalWrite(redPin, LOW);
        break;

    }
    irrecv.resume();
  }

}
```

Code for the Rasp berry Pi:

XBee data frame Parser:

```
package xbee

import (
	"bytes"
	"encoding/binary"
	"encoding/hex"
	"io"
)

type FrameType int

const (
	UnknownFrameType FrameType = iota
	DataSampleFrameType
)

type PacketType int

const (
	UnknownPacketType PacketType = iota
	UnicastPacketType
	BroadcastPacketType
)

// 7E 0012 92 0013A20040A9C7E5 B271 01 01 0002 00 0000 FC

// 7E 0012 92 0013A20040A9C7D4 BC24 02 01 0002 00 0002 4D
//
// start bit - 7E
// byte len before chksm - 0012 -> 18
// frame type - 92 (data sample frame)
// device id - 0013A20040A9C7D4
// net adrs - BC24
// packet type - 02 (broadcast packet)
// number of samples - 01 (fixed)
// digital chan mask - 0002 -> 0000000000000010
// analog chan mask - 00
// value - 0002
// chksm - 4D

type Frame struct {
	Len        int
	Type       FrameType
	DeviceID   string
	NetAddr    string
	PacketType PacketType
	SampleN    int
	DataD      map[int]bool
	DataA      map[int]int
}

func ParseFrom(r io.Reader, dataCh chan<- *Frame, errCh chan<-
error) error {
	buf := &bytes.Buffer{}
	for {
		byts := make([]byte, 1000)
		n, err := r.Read(byts)
		if err != nil {
			if err == io.EOF {
				return nil
			}
			return err
		}
		buf.Write(byts[:n])
		if buf.Len() < 20 {
			continue
```

```
		}

		tmpBuf := &bytes.Buffer{}
		for {
			b, err := buf.ReadByte()
			if err != nil {
				break
			}
			if b != 0x7e {
				continue
			}
			tmpBuf.WriteByte(b)

			if buf.Len() < 2 {
				break
			}

			byts := buf.Next(2)
			tmpBuf.Write(byts)

			l := int(binary.BigEndian.Uint16(byts))
			if buf.Len() < (l + 1) {
				break
			}

			tmpBuf.Write(buf.Next(l + 1))

			f, err := ParseFrame(tmpBuf.Next(tmpBuf.Len()))
			if err != nil {
				errCh <- err
				continue
			}
				dataCh <- f
			}
			buf.WriteTo(tmpBuf)
			tmpBuf.WriteTo(buf)
		}
}

func ParseFrame(data []byte) (*Frame, error) {
	// fmt.Println(hex.EncodeToString(data))
	if len(data) < 4 || data[0] != 0x7e {
		return nil, ErrInvalidData
	}

	if !checksum(data) {
		return nil, ErrChkSumMismatched
	}

	l := int(binary.BigEndian.Uint16(data[1:3]))
	if l != 18 {
		return nil, ErrUnsupportedLength
	}

	f := Frame{
		Len:      l,
		DeviceID: hex.EncodeToString(data[4:12]),
		NetAddr:  hex.EncodeToString(data[12:14]),
		SampleN:  int(data[15]),
	}

	switch data[3] {
	case 0x92:
		f.Type = DataSampleFrameType
	}
```

```
	switch data[14] {
	case 0x01:
		f.PacketType = UnicastPacketType
	case 0x02:
		f.PacketType = BroadcastPacketType
	}

	dmask := binary.BigEndian.Uint16(data[16:18])
	ddata := binary.BigEndian.Uint16(data[19:21])
	amask := data[18]

	f.DataD = map[int]bool{}
	f.DataA = map[int]int{}

	for i := 0; dmask != 0; i++ {
		if dmask&1 == 1 {
			f.DataD[i] = (ddata & 1) == 1
		}
		dmask = dmask >> 1
		ddata = ddata >> 1
	}

	for i := 0; amask != 0; i++ {
		if amask&1 == 1 {
			f.DataA[i] = int(binary.BigEndian.Uint16(data[19:21]))
		}
		amask = amask >> 1
	}

	return &f, nil
}

func checksum(data []byte) bool {
	sum := byte(0)
	for _, d := range data[3:] {
		sum = sum + d
	}
	return sum == 0xff
}
```

XBee Frame Error detection:

```
package xbee

import "errors"

var (
	ErrInvalidData       = errors.New("data is not a valid xbee data")
	ErrChkSumMismatched  = errors.New("checksum does not match")
	ErrUnsupportedLength = errors.New("length of packet is not supported")
)
```

Data Parser:

```
package main

import (
	"encoding/json"
	"flag"
	"fmt"
	"log"
	"net/http"
	"os"
	"strconv"
```

```
	"time"

	serial "go.bug.st/serial.v1"

	mhttp "github.com/taoufikul/go/http"
	"github.com/taoufikul/pi-xbee-finder/version"
	"github.com/taoufikul/pi-xbee-finder/xbee"
)

const usage = `
Usage:	finder [options] <tty port device>
Options:
	-b <baudrate> (default 115200)
	-p <web port> (default 8080)
`

func main() {
	fmt.Println("Hello", version.Version)

	var baudR, webP int
	flag.IntVar(&baudR, "b", 115200, "baud rate")
	flag.IntVar(&webP, "p", 8080, "web port")

	flag.Parse()

	serialP := flag.Arg(0)
	if serialP == "" {
		fmt.Println(usage)
		os.Exit(1)
	}

	ports, err := serial.GetPortsList()
	if err != nil {
		log.Fatal(err)
	}
	found := false
	for _, port := range ports {
		if port == serialP {
			found = true
		}
	}

	if !found {
		fmt.Println(serialP, "not found")
		fmt.Println("Available serial ports")
		for _, port := range ports {
			fmt.Println(port)
		}
	}

	devSig := map[string]bool{}
	dev := 1
	var devErr error

	port, err := serial.Open(serialP, &serial.Mode{BaudRate: baudR})
	if err != nil {
		log.Fatal(err)
	}
	defer port.Close()

	dataCh := make(chan *xbee.Frame, 100)
	errCh := make(chan error, 100)

	go func() {
		for err := range errCh {
			log.Println(err)
```

```
			devErr = err
		}
	}()

	go func() {
		sum := map[string]int{}
		cnt := map[string]int{}
		for data := range dataCh {
			// fmt.Printf("%#v\n", data)
			id := data.DeviceID
			val := 0
			if data.DataD[dev] {
				val = 1
			}
			sum[id] = sum[id] + val
			cnt[id]++
			if cnt[id] >= 10 {
				devSig[id] = (float64(sum[id])/float64(cnt[id]) > 0.5)
				sum[id] = 0
				cnt[id] = 0
			}
		}

	}()

	go func() {
		if err := xbee.ParseFrom(port, dataCh, errCh); err != nil {
			log.Fatal(err)
		}
	}()

	webHandler := func(w http.ResponseWriter, r *http.Request) {
		w.Header().Set("Access-Control-Allow-Origin", "*")
		data := map[string]interface{}{
			"data":  devSig,
			"error": devErr,
		}
		devErr = nil
		if err := json.NewEncoder(w).Encode(data); err != nil {
			log.Println(err)
		}
	}

	srvr := http.Server{
		Addr:    ":" + strconv.Itoa(webP),
		Handler: http.HandlerFunc(webHandler),
	}

	if err := mhttp.ManageServer(&srvr, 30*time.Second); err != nil {
		log.Fatal(err)
	}
}
```

__Graphical User Interface:__

```
import React, { Component } from 'react';

import { Rect, Circle, Wedge, Line, Stage, Layer, Group } from "react-konva";

export class OfficeMap extends Component {
	state = {
```

```
        color: "green",
};

constructor(props) {
        super(props)
        this.baseUrl = props.baseUrl;
        this.state = {
                dev1: {
                        key: props.dev1,
                        value: false,
                },
                dev2: {
                        key: props.dev2,
                        value: false,
                },
                dev3: {
                        key: props.dev3,
                        value: false,
                },
        }
}

componentDidMount() {
        this.fetchTimer = setInterval(
                () => this.fetchData(),
                1000
        );
}

componentWillUnmount() {
        clearInterval(this.fetchTimer)
}
```

```
        fetchData() {
                fetch(this.baseUrl)
                .then(res => res.json())
                .then(
                        (res) => {
                                let state = {};
                                for (let dev in res.data
) {
                                        if ( dev ===
this.state.dev1.key) {

        state["dev1"] = {key: dev, value: res.data[dev]};
                                        }
                                        if ( dev ===
this.state.dev2.key) {

        state["dev2"] = {key: dev, value: res.data[dev]};
                                        }
                                        if ( dev ===
this.state.dev3.key) {

        state["dev3"] = {key: dev, value: res.data[dev]};
                                        }
                                }
                                if (res.error) {
                                        con-
sole.log(res.error);
                                }
                                this.setState(state);
                        },
                        (error) => {
                                console.log(error);
                        }
                )
        }

        calcZone(d1, d2, d3) {
                let zone = d1;
```

```
        zone += (d2<<1);
        zone += (d3<<2);
        return zone;
    }

    renderZone0() {
        return (
            <Group>
            </Group>
        );
    }

    renderZone1() {
        return (
            <Group>
                <Wedge
                    fill={"gray"}
                    x={170}
                    y={165}
                    radius={300}
                    angle={180}
                />
                <Wedge
                    fill={"white"}
                    x={660}
                    y={165}
                    radius={300}
                    angle={180}
                    globalCompositeOperation={"destination-out"}
                />
                <Wedge
                    fill={"white"}
                    x={430}
                    y={20}
                    radius={250}
                    angle={50}
                    rotation={63.5}
                    globalCompositeOperation={"destination-out"}
                />
            </Group>
        );
    }

    renderZone2() {
        return (
            <Group>
                <Wedge
                    fill={"gray"}
                    x={660}
                    y={165}
                    radius={300}
                    angle={180}
                />
                <Wedge
                    fill={"white"}
                    x={170}
                    y={165}
                    radius={300}
                    angle={180}
                    globalCompositeOperation={"destination-out"}
```

```
                />
                <Wedge
                    fill={"white"}
                    x={430}
                    y={20}
                    radius={250}
                    angle={50}
                    rotation={63.5}
                    globalCompositeOperation={"destination-out"}
                />
            </Group>
        );
    }

    renderZone3() {
        return (
            <Group>
                <Wedge
                    fill={"gray"}
                    x={170}
                    y={165}
                    radius={300}
                    angle={180}
                    />
                <Wedge
                    fill={"gray"}
                    x={660}
                    y={165}
                    radius={300}
                    angle={180}
                    globalCompositeOperation={"destination-in"}
                />
                <Wedge
                    fill={"white"}
                    x={430}
                    y={20}
                    radius={250}
                    angle={50}
                    rotation={63.5}
                    globalCompositeOperation={"destination-out"}
                />
            </Group>
        );
    }

    renderZone4() {
        return (
            <Group clipX={20} clipY={20} clipWidth={800} clipHeight={145}>
                <Wedge
                    fill={"gray"}
                    x={430}
                    y={20}
                    radius={250}
                    angle={180}
                />
            </Group>
        );
    }

    renderZone5() {
```

```
        return (
            <Group>
                <Wedge
                    fill={"gray"}
                    x={170}
                    y={165}
                    ra-
dius={300}
                    an-
gle={180}
                />
                <Wedge
                    fill={"gray"}
                    x={430}
                    y={20}
                    ra-
dius={250}
                    angle={50}
                    rota-
tion={63.5}
                    global-
CompositeOperation={"destination-in"}
                />
                <Wedge
    fill={"white"}
                    x={660}
                    y={165}
                    ra-
dius={300}
                    an-
gle={180}
                    global-
CompositeOperation={"destination-out"}
                />
            </Group>
        );
    }
```

```
    renderZone6() {
        return (
            <Group>
                <Wedge
                    fill={"gray"}
                    x={660}
                    y={165}
                    ra-
dius={300}
                    an-
gle={180}
                    />
                <Wedge
                    fill={"gray"}
                    x={430}
                    y={20}
                    ra-
dius={250}
                    angle={50}
                    rota-
tion={63.5}
                    global-
CompositeOperation={"destination-in"}
                    />
                <Wedge
    fill={"white"}
                    x={170}
                    y={165}
                    ra-
dius={300}
                    an-
gle={180}
                    global-
CompositeOperation={"destination-out"}
                />
            </Group>
```

```
        );
    }

    renderZone7() {
        return (
            <Group>
                <Wedge
                    fill={"gray"}
                    x={660}
                    y={165}
                    radius={300}
                    angle={180}
                />
                <Wedge
                    fill={"gray"}
                    x={430}
                    y={20}
                    radius={250}
                    angle={50}
                    rotation={63.5}
                    globalCompositeOperation={"destination-in"}
                />
                <Wedge
                    fill={"gray"}
                    x={170}
                    y={165}
                    radius={300}
                    angle={180}
                    globalCompositeOperation={"destination-in"}
                />
            </Group>
        );
    }

    renderZone(zone) {
        switch (zone) {
            case 1:
                return this.renderZone1();
            case 2:
                return this.renderZone2();
            case 3:
                return this.renderZone3();
            case 4:
                return this.renderZone4();
            case 5:
                return this.renderZone5();
            case 6:
                return this.renderZone6();
            case 7:
                return this.renderZone7();
            default:
                return this.renderZone0();
        }
    }

    render() {
        let {state} = this;
        let zone = this.calcZone(state.dev1.value, state.dev2.value, state.dev3.value);
```

```
return (
<Stage width={window.innerWidth} height={window.innerHeight}>
<Layer
clipX={20}
clipY={20}
clipWidth={800}
clipHeight={500}
>
{this.renderZone(zone)}
<Circle
fill={state.dev1.value ? "green" : "red"}
x={170}
y={170}
radius={5}
fillPriority={100}
/>
<Circle
fill={state.dev2.value? "green" : "red"}
x={660}
y={170}
radius={5}
/>
<Circle
fill={state.dev3.value ? "green" : "red"}
x={430}
y={25}
radius={5}
/>
<Rect
x={20}
y={20}
width={800}
height={500}
stroke={"black"}
strokeWidth={2}
/>
<Line
points={[20, 160, 370, 160]}
stroke={"black"}
strokeWidth={10}
/>
<Line
points={[500, 160, 820, 160]}
stroke={"black"}
strokeWidth={10}
/>
<Line
points={[170, 280, 170, 520]}
stroke={"black"}
strokeWidth={10}
/>
</Layer>
```

```
                    </Stage>
                );
            }
        }
```