

INDOOR POSITIONING SYSTEM FOR WAREHOUSE ASSET TRACKING

BY

WONG ZHENG YAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER ENGINEERING (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2019

REPORT STATUS DECLARATION FORM

Title: _____

Academic Session: _____

I _____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

Supervisor's name

Date: _____

Date: _____

INDOOR POSITIONING SYSTEM FOR WAREHOUSE ASSET TRACKING

BY

WONG ZHENG YAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER ENGINEERING (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2019

DECLARATION OF ORIGINALITY

I declare that this report entitled “**INDOOR POSITIONING SYSTEM FOR WAREHOUSE ROBOT**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I cannot express enough thanks and appreciation to my supervisor, Dr Lee Wai Kong for his continuous support and encouragement. I offer my sincere appreciation for the bright opportunity provided for me to engage in an indoor positioning system design. The advice given by Dr Lee has been a great help in realizing this project.

My completion of this project could not have been accomplished without the support of my friends and family for standing by my side during hard times. Finally, I would like to express my very great appreciation to Ms CY who has been by my side for the whole time through the project, encouraging and motivating me every time I am drowning in the pool of frustration and on the edge of giving up.

ABSTRACT

This project is an IPS design project that aims to design a warehouse robot capable of delivering goods and merchandise within a warehouse on its own without the need of human aid. This paper provides the concept, methodology and design of the IPS through implementing an Arduino Mega combining various modules. The reason behind Arduino Mega chosen to be the brain of this project is that it has a more powerful ATmega2560 microcontroller with a larger SRAM, Flash Memory and I/O pins compared to the ATmega328p of the Arduino Uno. This project focus on how to enable the robot to accurately calculate its indoor position through RFID tags deployed across the test field. Combining various method and algorithms is very challenging, as all the calculations have to be tally. Additionally, another challenge of this project is that the information fed back to the Arduino for calculation have to be ensured to be accurate. This problem arises since all the modules including the communication between Arduino and the robot's stm32 are using serial/UART, except for the RFID module which uses SPI. The base of this project is the RFID module which enables the robot to initially positions itself, and then rectify its location either through DR or RSSI value or both. The output material of this project is mainly Arduino IDE and a simple web page to monitor or control the robot and some test run results.

TABLE OF CONTENT

DECLARATION OF ORIGINALITY	i
ACKNOWLEDGEMENTS	ii
ABSTRACT	iii
TABLE OF CONTENT	iv
LIST OF FIGURES	vi
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
CHAPTER 1: INTRODUCTION	1
1.1.1 Passive Reader Active Tag (PRAT):	4
1.1.2 Active Reader Passive Tag (ARPT):	4
1.1.3 Active Reader Active Tag (ARAT):	5
1.2.1 Low Frequency (LF):	5
1.2.2 High Frequency (HF):	5
1.2.3 Ultra-High Frequency (UHF):	5
CHAPTER 2: LITERATURE REVIEW	6
2.1 Dead Reckoning (DR):	7
2.2 Triangulation:	8
2.3. Radio frequency identification (RFID):	8
CHAPTER 3: SYSTEM DESIGN	13
3.1 Full system diagram	13
3.2 Hardware Components	14
3.3 Hardware design and wiring	22
3.4 Software design and coding	24
3.4.1 Cytron ESP8266 Wi-Fi Shield	24
3.4.2 MFRC522 RFID Module	27
3.4.3 Serial Commands for Controlling Robot	28
3.4.4 XBee Coordinator	29
3.4.5 MySQL Database	33
3.4.6 Webpage (HTML/PHP)	35
CHAPTER 4: METHODOLOGY AND TOOLS	40
4.1 Methodology	40
4.2 Software Tools	41

CHAPTER 5: FUNCTIONAL SPECIFICATION ANALYSIS AND VERIFICATION.....	47
5.1 RFID Test.....	47
5.2 RSSI Test	49
5.2.1 RSSI Test 1	49
5.2.2 RSSI Test 2	50
5.2.3 RSSI Test 3	51
5.3 Full System Test	53
CHAPTER 6: CONCLUSION	59
REFERENCES	61
APPENDIX.....	A-1
POSTER.....	P-1
TURNITIN RESULT.....	T-1

LIST OF FIGURES

Figure Number	Title	Page
2.1	Estimating aircraft position using Dead Reckoning	7
2.2	Positioning based on TOA/RTOF measurements (Liu et al. 2007)	8
2.3	Multilateration technique by Bekkali et al. (2007)	9
2.4	Multilateration technique by Huang et al. (2015)	9
2.5	A UWB positioning system as buiding infrastructure	11
2.6	Matrix based tag deployment (Qin et al. 2017)	12
3.1	Full system diagram	13
3.2	Robot base design	14
3.3	Robot's on board STM32 microcontroller module	15
3.4	Arduino Mega 2560	16
3.5	XBee RF Wire Antenna	17
3.6	Cytron's SKXBee starter kit module	18
3.7	MFRC522 RFID Module	19
3.8	RFID 13.56 MHz Tags	19
3.9	16 Bytes Block 0 of 125 kHz RFID tag	20
3.10	Cytron ESP8266 Wi-Fi Shield	20
3.11	Hardware connections diagram	22
3.12	Cytron ESP8266 Wi-Fi Shield Libraries	24
3.13	Wi-Fi Shield Initialization	24
3.14	ESP8266 Serial and Internet Connection	25
3.15	HTTP GET Request	26
3.16	Close Connection	26
3.17	MFRC522 Libraries	27
3.18	MFRC522 Module Initialization	27
3.19	MFRC522 Reading UID	28
3.20	Serial Commands for Robot	28
3.21	Writing commands to robot	28
3.22	Serial1 definition for XBee Module	29
3.23	Low Address of Router 1 to 4	29

3.24	AT Mode and ATCN Command	30
3.25	ATDL Command	31
3.26	ATDB Command	31
3.37	Sending Data Packets to Desired Router	32
3.38	Location Database	33
3.39	History Database	34
3.40	Webpage Display	36
3.41	MySQLi Connection	36
3.42	PHP GET Method	37
3.43	Search and Update Database	38
3.44	HTML Code	39
4.2	Arduino IDE with description	41
4.3	XCTU Software Diagram	42
4.4	XAMPP for windows	43
4.5	MySQL/MariaDB database GUI	44
4.6	phpMyAdmin Directory/Workspace	44
4.7	Naming for Database	45
4.8	Database Created Pop-Up	45
4.9	Creating Tables	46
5.1	RFID Test Setup	47
5.2	RSSI Test 1	49
5.3	RSSI Test 2	50
5.4	RSSI Test 3 Setup	51
5.5	Full System Test	53
5.6	Zone 4 Test	54
5.7	Zone 4 Test Result	54
5.8	Zone 3 Test	55
5.9	Zone 3 Test Result	55
5.10	Zone 2 Test	56
5.11	Zone 2 Test Result	56
5.12	Zone 1 Test	57
5.23	Zone 1 Test Result	57
5.24	All Rounded Test Result	58

LIST OF TABLES

Table Number	Title	Page
5.1	RFID full test result	31
5.2	RFID UID test result	31

LIST OF ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
DR	Dead Reckoning
FCC	Federal Communications Commission
FDMA	Frequency-Division Multiple Access
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HF	High Frequency
I2C	Inter-Integrated Circuit
IPS	Indoor Positioning System
LF	Low Frequency
LOS	Line-of-Sight
OFDMA	Orthogonal Frequency-Division Multiple Access
PID	Proportional Integral Derivative
RFID	Radio-Frequency Identification
RSSI	Received Signal Strength Indicator
RTOF	Round-Trip Time of Flight
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TOA	Time of Arrival
TOF	Time of Flight
UART	Universal Asynchronous Receiver-Transmitter
UAV	Unmanned Ariel-Vehicles
UHF	Ultra-High Frequency
UHF	Ultra-High Frequency
UWB	Ultra-Wide Band

CHAPTER 1: INTRODUCTION

Indoor positioning systems (IPS) have been around for a long time, utilizing sensors, nodes, and communication technologies combined with various methods to locate objects explicitly in indoor environments. The IPS technology enables us to compute the location of an object with significant accuracy depending on the method and algorithms used. The market of IPS technology is getting bigger by day since the possibilities and opportunities it can bring is tremendous. There are of course tons of various journals and surveys on IPS technologies up to this day which we would come to that later in this paper.

Despite that, many of those journals are obsolete due to the fact that the area of indoor positioning systems is constantly changing and evolving fast. One of the many reasons why the IPS technology has been a success in the past was because the flexibility of it. The location or position of an object or device, sometimes made to a wearable can be easily implemented using sensor nodes embedded into the device.

Location pin-pointing has first been successful when implemented to outdoor surroundings using technology like GPS (Xu 2007). The Global Positioning System or mainly known as GPS relies on satellite technology. The GPS signal is broadcasted to the user along with the forecasted geo-positions of the satellites. The action of measuring the ranges between the few concurrently observed satellites is the fundamental technique or method in GPS technology. Through a few of the known positions of the observed satellites, combined with the measured distances or range between the satellites and the receiver, the location and position of the receiver can be computed or calculated. The position change determined later is derived to become the velocity of the receiver. The most important applications of the GPS are positioning and navigating. (Xu 2007, p20).

GPS can provide high accuracy geolocation along with the timeframe information to a receiver anywhere on Earth given that there is an unobstructed line-of-sight (LOS) to

CHAPTER 1: INTRODUCTION

minimum four GPS satellites. The line-of-sight propagation is coin as an important characteristic in electromagnetic wave propagation. Electromagnetic wave propagation is the operational fundamentals of all wireless communications and telecommunications, such as GPS, RFID, FDMA, OFDMA. Electromagnetic waves that travel in a direct path from the source or transmitter to the receiver are said to have achieved LOS, without being reflected, refracted or diffracted.

GPS technology has since been an immense influence turning to a necessity affecting our daily life through various applications in guidance, mapping, and positioning (Haykin 2005). In contrast to that, in an indoor environment, the serviceability of the GPS is very limited since there are much attenuation and absorption of the GPS signal by the atmosphere, plus the lack of LOS, making it quite expendable. GPS signals also get highly attenuated after they cross through walls. Furthermore, having a precision of up to 50 metres is highly ineffective when it comes to locating a smaller object with respect to the building the object is inside to. For instance, GPS are advantageous when it comes to geolocating a person or device with respect to the entire world. Thus, the demand for a more significant and specialized methods and technology for IPS or indoor location systems has been widely established.

Many surveys and articles have been written throughout the years based on various IPS or IPS related topics, implementing different technologies such as RFID, Bluetooth, image processing and inertial sensors (Lionel et al. 2004), (Haverinen & Kempainen 2009), (Kriz et al. 2016), (Dabove et al. 2015). Since we will only be doing our research in IPS and not for outdoor positioning, the atmosphere and terrain would not be a problem for us, but rather some other miscellaneous object that comes into obstruction of our transmitter and receiver. On the other hand, similar techniques and methods for GPS geolocating can be administered in IPS, such as dead reckoning (DR).

The term “dead reckoning” is the process of calculating an objects current position by using a prior determined position. DR is mostly used in navigation with respect to a 2-Dimension plane. The latter position of an object is estimated based upon known speed

CHAPTER 1: INTRODUCTION

and direction through path integration. Alone, DR technology has become obsolete with the birth of GPS, but combined, considering the situation of loss of GPS signal, DR can aid through navigating.

There are various wireless technologies used for wireless IPS, 1) the algorithm of location positioning or the method of determining location based on numerous measurements of the signal such as Time of Flight (TOF), Time of Arrival (TOA) and Received Signal Strength Indicator (RSSI); 2) the physical layer or location sensor infrastructure e.g. transmitter and receiver which is the communicating link between the static and mobile device; 3) via vision-based localization, i.e., capturing and analysing images to locate an object (Liu et al. 2007), (Qin et al. 2017).

Measurement between hardware components of the system includes both the transmission and reception of signals. “A wireless IPS must at least consist of two separate hardware components in order for it to work: a measuring unit and a signal transmitter. The measuring unit usually comprises as the major part of the system.” (Liu et al. 2007).

In the end of this project, a fully functional warehouse robot with IPS on-board equipped with various sensors that can detect obstacles and pinpoint its location based on the deployed RFID tags will be produced. This project includes the DR algorithm and RFID positioning techniques. The objectives of this project is to develop a fully functional warehouse robot that can detect obstacles in all four directions, detect and reading position information from deployed RFID tags, read and detect RSSI value transmitted used for DR, and lastly reaching the intended destination via lane recognition

This project aims to more accurately position a robot in a given vicinity with the aid of various sensors and algorithm. As positioning is very important for the warehouse robots to reach its intended destination when conveying the parcel or package, the

CHAPTER 1: INTRODUCTION

ability to pinpoint the location for the robot is vital. The target of this project is to improve at least 30% of the efficiency of such robots. The location tracking systems enables the warehouse robot to be tracked and monitor within the working grid.

However, this project only focus on a single robot and not a swarm of robots. To successfully implement a swarm of warehouse robots, a single robot must first be constructed and can work properly on its own before integrating more robots into the system. To implement such systems with many robots, new functions such as location broadcasting must be implemented so that each robot can know not only its position but other robots position too in order to avoid path collision and task dispute hazard (two robots aiming to do the same task).

RFID technology implements electromagnetic fields to automatically identify and track (in some cases) RFID tags attached or embedded onto objects. RFID tags which contain electronically-stored information has two main forms, passive tags, and active tags. Active tags have an embedded power source such as a battery or local power source and usually can function up to hundreds of meters from the RFID reader. On the other hand, passive RFID tags gets its energy emitted from a nearby RFID reader, and then sending back the information stored within the tag itself. RFID systems can be classified into three main types of tag and reader.

1.1.1 Passive Reader Active Tag (PRAT):

PRAT systems use passive readers which can only receive radio frequency signals from active tags which has its own power source that also in turn powers the passive reader.

1.1.2 Active Reader Passive Tag (ARPT):

ARPT systems has an active reader (with own power source), which transmit interrogator signals and powering up the passive tags, and in turn receive information from the tags for purposes such as authentication.

1.1.3 Active Reader Active Tag (ARAT):

ARAT systems uses both active readers and active tags awoken by interrogator signal from the reader to lower the power consumption. The range for ARAT systems is from low to ultra-high.

The commonly use RFID systems can be categorized into – Low Frequency (LF), High Frequency (HF), Ultra-High Frequency (UHF).

1.2.1 Low Frequency (LF):

LF systems have frequencies ranging from 120 to 150 kHz, with a range within 10 cm, low data speed, and usually used in animal identification or factory data collection. Usually, LF RFID tags can be read multiple times but can only be write once.

1.2.2 High Frequency (HF):

HF systems usually has frequencies of 13.56 MHz, with range from 10 cm to 1 m, low to moderate data speed. Commonly used for smart cards, identification cards. One famous manufacturer for HF ID cards is Mifare. HF RFID tags can be read and write multiple times.

1.2.3 Ultra-High Frequency (UHF):

UHF systems has frequencies ranging from 860 to 960 MHz, with range up to 100m, moderate to high data speed, commonly used in defence or industrial application, usually used along with active tags.

CHAPTER 2: LITERATURE REVIEW

The plan of this project is to overcome problems such that when the signal of a tag supposed to be received by one reader is read by another reader, or a reader reads multiple tags simultaneously. Using just raw data from readers and tags lead to error prone readings and very low accuracy on locating the object. The main problem is not that there will be no data return, but rather how accurate and reliable is the data return, and in turn does it accurately depict the location of our target objects.

Most surveys and journals about IPS has been published throughout the past few decades, which most of them are out dated such as (Hightower & Borriello 2001), (Want et al. 1992) and (Haverinen & Kemppainen 2005). Furthermore, obsolete technologies like Bluetooth implemented in (Kriz et al. 2016) needs more beacon or localization nodes that uses more power to sustain the system.

To classify, there are mainly four different system topologies for location positioning systems, namely self-positioning, remote positioning, indirect self-positioning and indirect remote positioning (Drane et al. 1998). In an indirect remote positioning system, a self-positioning system sends position data to a remote or mobile location, which fulfil our target in creating an IPS through implementing passive HF RFID tags that sends position data to robots that are mobile.

Since then, many researchers have used RFID technology in hopes to create a sustainable IPS by implementing both passive (without build in power supply) and active (with power supply) RFID tags, combined with algorithm calculating TOA, TOF, RTOF, and RSSI values (Saab & Nakad 2011), (Bekkali et al. 2007). The most often problem and challenge faced while using RFID in IPS, was the distance between the tags and the reader. It is not always easy to accurately model the radio wave propagation in indoors. To ensure that, a clear LOS must be achieved, specific site parameters i.e., floor layout and other mobile objects within the vicinity must be known beforehand.

Methods used in RFID based IPS system are usually based on monitoring the RSSI value broadcasted by the transmitter covering a particular vicinity. When the device or object's distance increase from the transmitter, the signal strength decreases proportionally, and the time taken for the signal to reach the receiver increases gradually. Furthermore, RFID based scheme IPS is also widely implemented as a navigation aid, marking checkpoints by deploying passive RFID tags. Three basic approaches are being used – dead reckoning, triangulation, radio frequency identification.

2.1 Dead Reckoning (DR):

Before GPS, DR based systems are implemented by calculating the current position according to the last known position, typically serve as an enhancement to IPS such as (Laftchiev et al. 2015) and (Lee et al. 2010). For instance, a DR-based system called Inertial Navigation System (INS) uses accelerometers and gyroscopes to continuously calculate the velocity, direction or orientation, and the position of a moving object in the absence of GPS. Combining DR to IPS, the position localization errors can be lowered to a smaller degree.

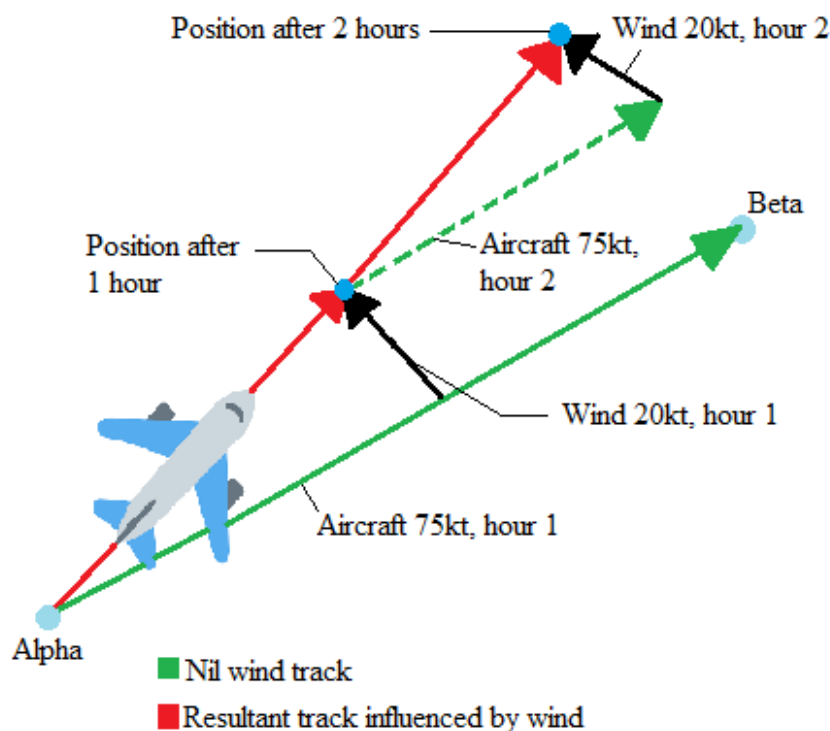


Figure 2.1 Estimating aircraft position using Dead Reckoning

2.2 Triangulation:

Triangulation utilize the geometric properties of triangles to estimate and calculate the target location. There are two main derivations, lateration and angulation (Liu et al. 2007). Lateration is also called range measurement technique, as it estimates an objects location through measuring its distance from multiple reference nodes. The TOA and TDOA rather than the RSSI value are measured. RTOF in some cases, is also measured and used as a range estimation. On the other hand, angulation locates an object computing angles relative to multiple points used as reference. In this project, we focus on short range, low cost reader tag and indoor environment.

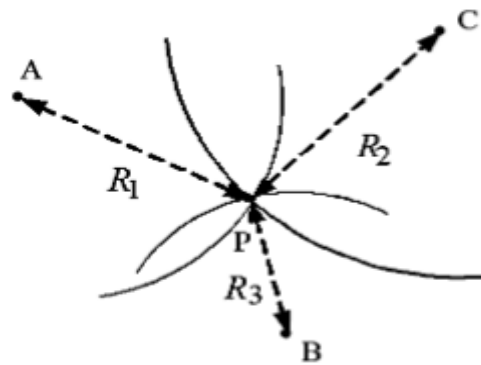


Figure 2.2 Positioning based on TOA/RTOF measurements (Liu et al. 2007)

2.3. Radio frequency identification (RFID):

RFID based systems have long been used as an aid to navigation or positioning system be it outdoor or indoor (Chon et al. 2015), (Gandhiraj & Kumar 2015). Furthermore, RFID based schemes have also been proposed such as lane recognition and indoor autonomous vehicle localization (Digiampaolo & Martinelli 2012).

IPS in general has much different approach than outdoor localization, such that the terrain and climate would not be much of a big problem to indoor localization. The challenge for building an IPS is however, more on the range inside a building. To accurately pinpoint an object within a 3D spectrum, it is much more complicated and error prone. In order to calculate an object within the range of multiple nodes that form a partition of 3-dimension vector which combines with other partition to complete a whole building, the x-axis, y-axis and z-axis have to be all measured with very little accuracy.

CHAPTER 2: LITERATURE REVIEW

Huang et al. (2015) implements probabilistic RFID map generation, Kalman filtering AND Heron-Bilateration location estimation, which has been used successfully in different prediction applications or state determination of a system despite the existence of more sophisticated filters. The result of the system implementing both probabilistic RFID map generation and Kalman filtering is quite accurate. However, this type of approach tends to have a high value of path loss signal strength. Besides, the deployment of the RFID tags by Bekkali et al. (2007) were in a random fashion, making it even harder to accurately position the robots, since their multilateration technique was used also to detect the location of unknown tags. However, in Huang et al's (2015), they deployed the RFID reader or tags in a more consistent fashion. On the contrary, more readers were deployed to find the targeted tag rather than deploying multiple tags with one single reader.

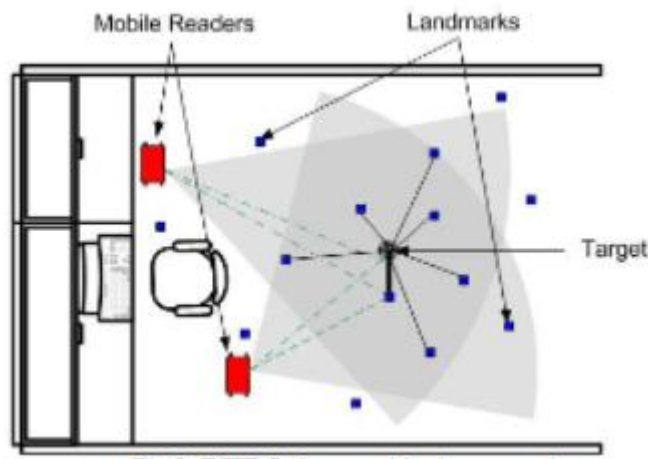


Figure 2.3 Multilateration technique by Bekkali et al. (2007)

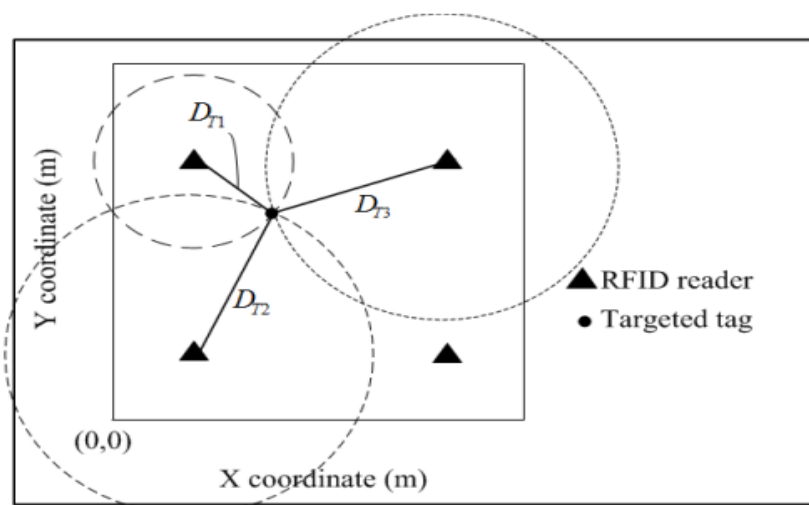


Figure 2.4 Multilateration technique by Huang et al. (2015)

CHAPTER 2: LITERATURE REVIEW

To resolve this problem, we would only focus on positioning the robot and the RFID tags would be deployed using a special matrix based scheme which would be discussed further later in this paper. In other words, all of our location transmitter i.e. RFID tags, RSSI transmitter would be at a fixed location, making it relatively simpler for the robot to locate itself.

Furthermore, the RFID system used must be at least ultra-wide frequency (UHF) to successfully implement such system, which cost a lot when implemented at a large scale. Despite, the accuracy, the downside to this system is that it does not estimate the location of the object when the distance is too vast from the reader. In other words, the localization only occurs when the target object can close enough to the receiver or reader.

Such UHF RFID based IPS for autonomous vehicle can also be seen implemented in Digiampaolo & Martinelli's (2014). The duo also implemented Kalman filter combined with UHF RFID which requires RFID antennas to be realized. In other words, the cost will be high.

To overcome such problems, the RFID used in this project would only be low cost HF RFID tags and readers. This would greatly decrease the cost of building an IPS even the RFID tags are heavily deployed. Although the range of HF RFID systems have a mere range of 2cm, but we can transform the former 3-dimension spectrum into a 2-dimension which increases the accuracy exponentially. On the other hand, to conquer the problem of the object out of range, DR and triangulation techniques can be combined with the RFID system to estimate and position the target object when it is out of range.

IPS systems also suffer from expensive equipment's such as Tiemann et al.'s (2015) which implements ultra-wide band (UWB) integrated with unmanned aerial-vehicles (UAV) in the environment absent of global navigation satellite system (GNSS). By using triangulation methods and hybridization with GNSS, they successfully implemented UWB based IPS. However, the downside of UWB is that it is only permitted to work between 3.1 GHz and 10.6 GHz by FCC, and there is a big trade off involving the performance and the range. As extra data means longer messages or lower processing gain, whilst the power level is strictly controlled, less gain means less range (Ingram et al. 2004, p712).

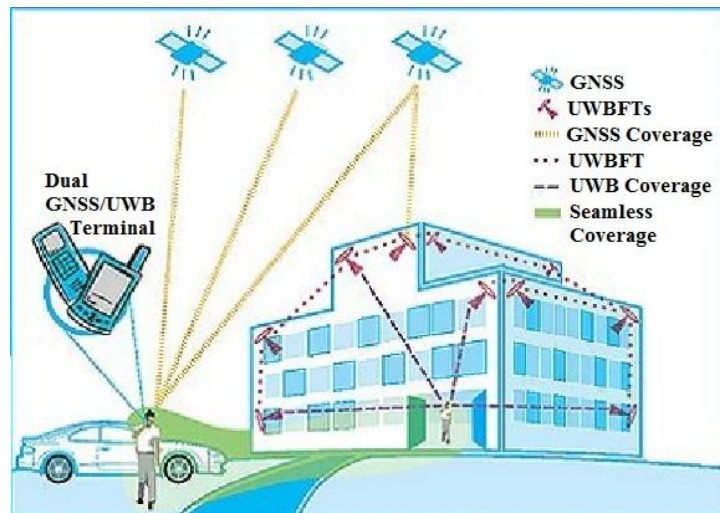


Figure 2.5 A UWB positioning system as building infrastructure

Once again, the problem mentioned above can be solved through using ordinary HF RFID tags and reader. There are no restrictions to such systems and the performance of such system is not dependent on the range per se, meaning there is no trade off since we will not be transmitting long data messages, rather just the ID of the RFID tag for identification (positioning) purposes.

One of the most recent papers on RFID based localization system for automated vehicles was written by Qin et al. (2017). RFID tags and readers used in their paper are much similar to what we are going to use in this paper. HF RFID are deployed in a matrix based scheme to enhance the localization accuracy, combined with their own error cognitive algorithm to further augment the reliability of the system, where the vehicles with higher localization accuracy are used to correct the locations of the vehicles with lower localization accuracy (Qin et al. 2017, p9946).

The above system has quite a high accuracy, but in order to do so, it has to be combined with GPS to perform that well. Furthermore, the system above was designed for vehicles driving outdoor, which take much considerations into terrain and weather, which will not be a big challenge when it comes to implementing IPS. On the other hand, the matrix based tag deployment scheme and tag clustering scheme which we are going to adopt in this project has a profound result when successfully implemented.

CHAPTER 2: LITERATURE REVIEW

Our approach on tag deployment will be similar to the one in (Qin et al. 2017), implementing a of matrix based scheme for our HF passive RFID tags. However, in Qin et al.'s (2017) paper, their positioning system is designed for automated vehicles which drives at a much faster speed on rough surface or terrain. Due to the two factors mention, the main weakness or struggles that they faced is that the transmission between the RFID reader and tag may be unreliable due to high mobility and environmental issues. If the vehicles move at a very fast speed, the transmission may not even be successful since a specific time period is required for one successful transmission between the tag and the reader.

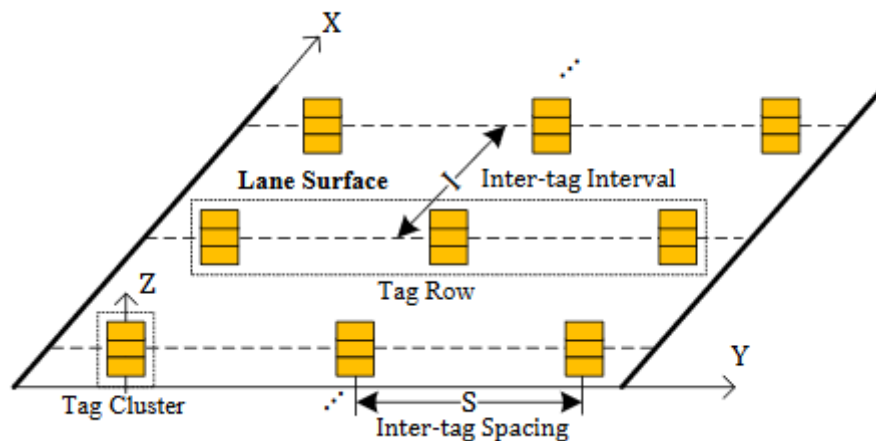


Figure 2.6 Matrix based tag deployment (Qin et al. 2017)

For both problems, it would not be much of an affect to our IPS since warehouse robots are only going to travel inside the warehouses flat surfaces and the speed of the robot would be significantly smaller than an automobile. Despite that, a nominal speed will be set for the robots to achieve the best result in terms of the fastest speed a robot can move over a RFID tag in order for one successful transmission to occur.

Overall, by keeping the distance between the tags and the reader relatively close, whilst all the location of tags is known, the RFID reader on the robots are able to receive the data transmitted from the tags.

3.1 Full system diagram

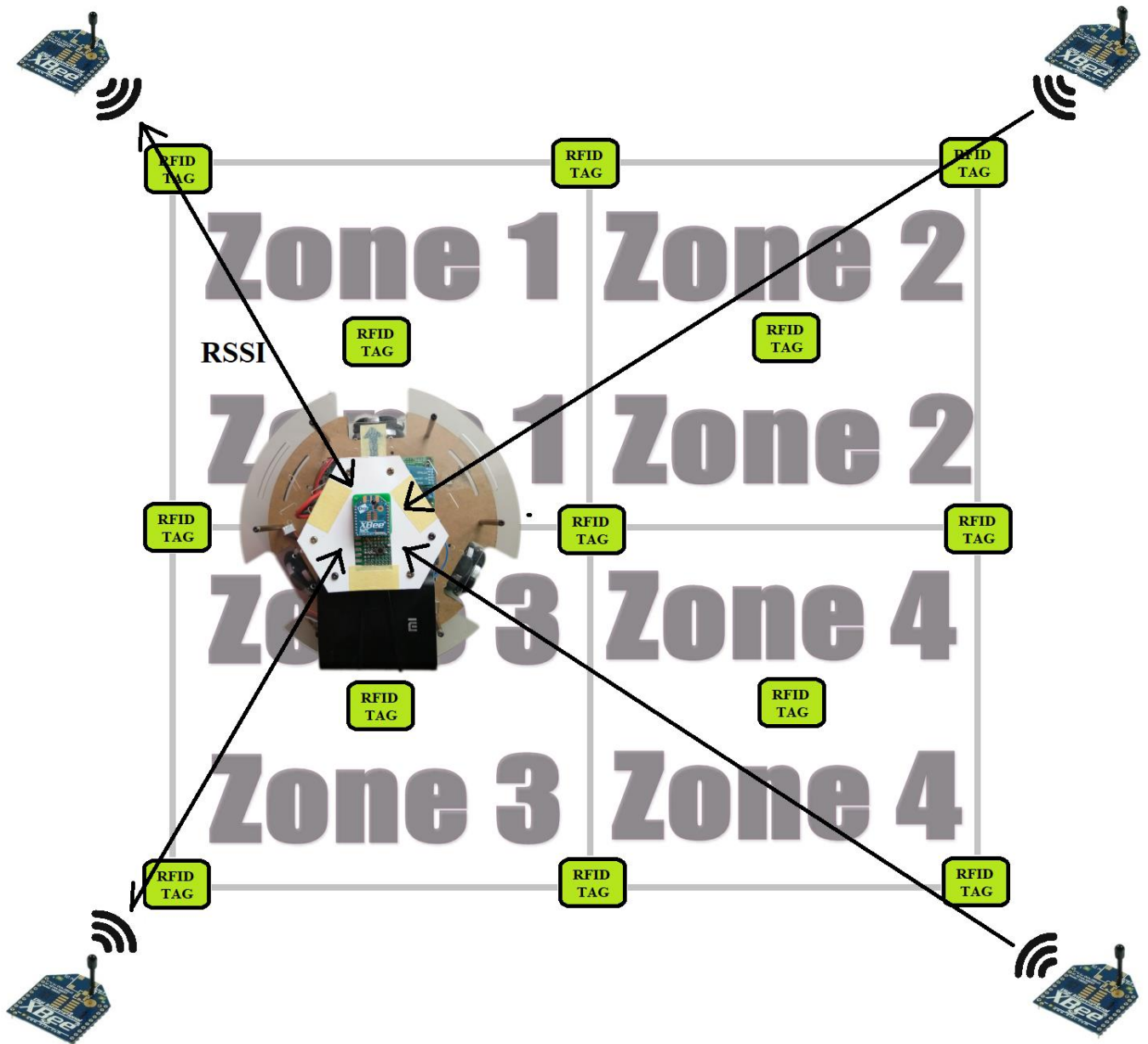


Figure 3.1 Full System Diagram

3.2 Hardware Components

The base of this project will of course be the robot which comes with an onboard STM32 microcontroller preloaded with the stock firmware. The robot can be controlled through PID/CAN or serial UART, and can move in multidirectional due to the special design of the robot's encoder. Despite that, due to mechanical constraints and design, the robot will sometimes deviate from its intended path. Hence, extra care and configuration have to be taken to ensure all three speed encoders will move in the intended direction and speed. However, that is not the concern of this project. We only focus on the positioning of the robot and aim to detect the robots position even after it deviates.

In this project, the robot is interfaced and control through UART. The robot which runs on the stock firmware has two interfacing modes, Mode 1 with prefixed movement direction and tweakable speed, and Mode 2 with tweakable movement direction and speed. Mode 1 is used in this project for simplicity since we only need the robot to move in the fixed directions of forward, backward, left and right. Speed of the encoders must also be configured properly since moving too fast would affect the robot from reading both the RFID and RSSI values.

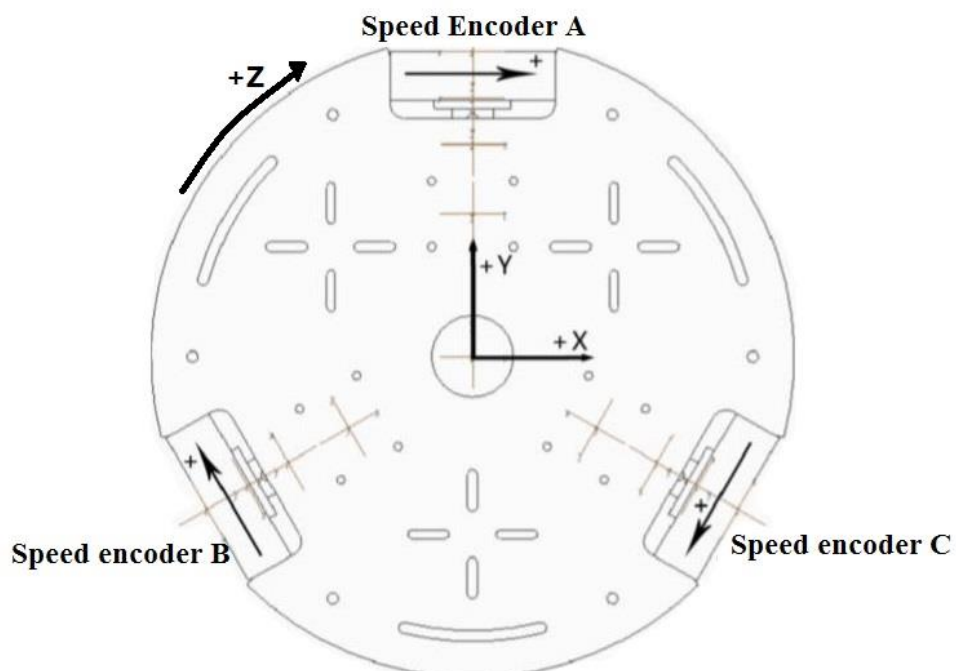


Figure 3.2 Robot's base design

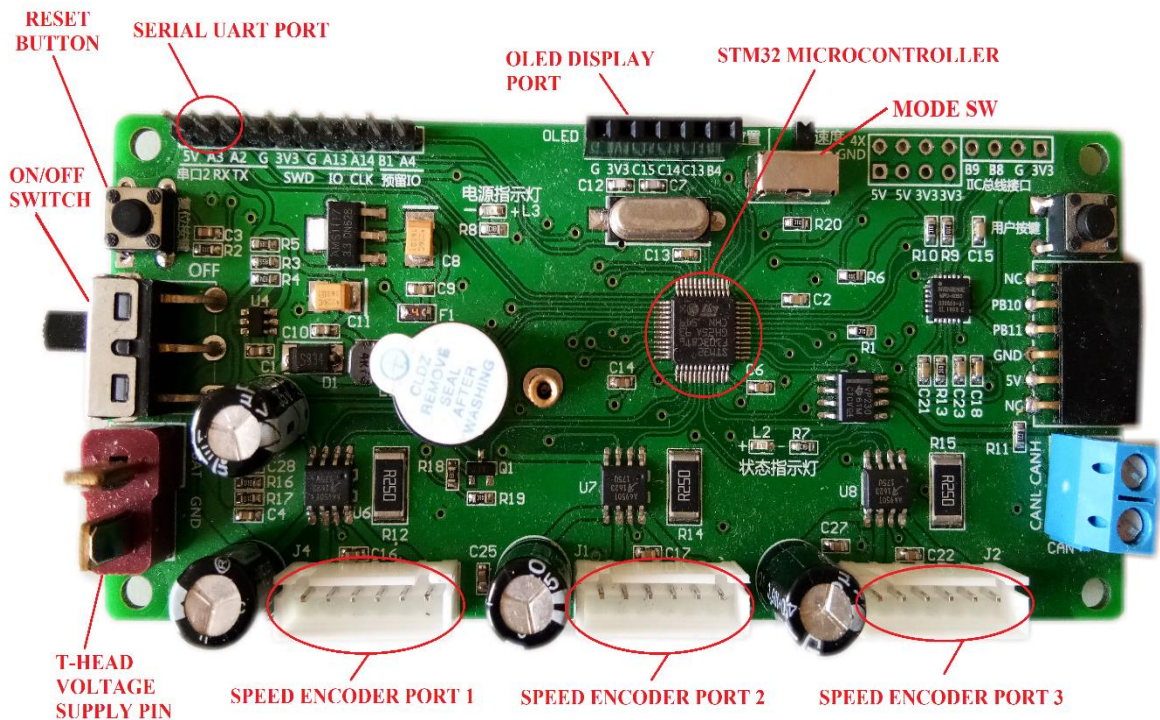


Figure 3.3 Robot's on board STM32 microcontroller module

The on board STM32 microcontroller module comes with all the required module and bare components to operate the robot. Users can even directly plug in an OLED display to view the current status of the robot. The microcontroller module also has a T-Head voltage supply pin that can be connected to a battery pack. The switch on the top right hand side just next to the OLED display port is the mode switch which users can just flip and change from mode 1 to mode 2 and vice versa.

With the initial requirements gathered on hand, a quick design involving all the parts and hardware can be done. To implement the IPS for the warehouse robot, there must be a microcontroller which serves as a brain to our entire system. In our case where many other modules are required, an Arduino Mega 2560 is used, since Arduinos are the best when it comes to interfacing with sensors and other miscellaneous modules.

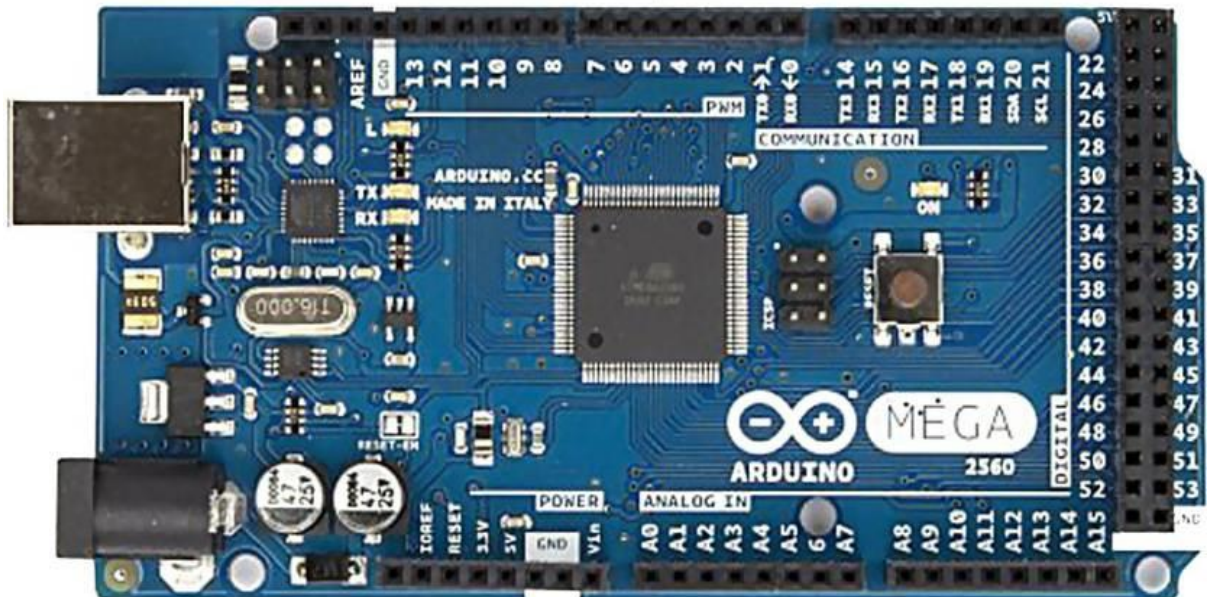


Figure 3.4 Arduino Mega 2560

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. With 54 digital I/O pins and 16 analog inputs onboard, it is perfect for robotics projects as such. The Arduino Mega 2560 can be programmed through the use of the official Arduino IDE. The biggest benefits of an Arduino microcontroller is its ready to use structure. With the required libraries installed and configured, users can start programming right away without needing to worry for drivers. As the Arduino Mega 2560 has many on-board components that are ready without any extra configuration such as the powerful ATmega 2560 microcontroller, a 5 Volts regulator, multiple hardware serial communication interface, an oscillator, ADC, a burner, LEDs and headers for the connections. By a mere USB cable, we can program the Arduino from our desktop. The full pinout diagram can be seen at the appendix.

CHAPTER 3: SYSTEM DESIGN

To achieve the project with low cost hardware, the task of reading RSSI values will be separated from the RFID system itself, since RFID systems with RSSI transmitter and receiver usually has a higher price. To compensate that, we are going to use Xbee RF Zigbee module, which is has an antenna capable of transmitting and receiving RSSI values.

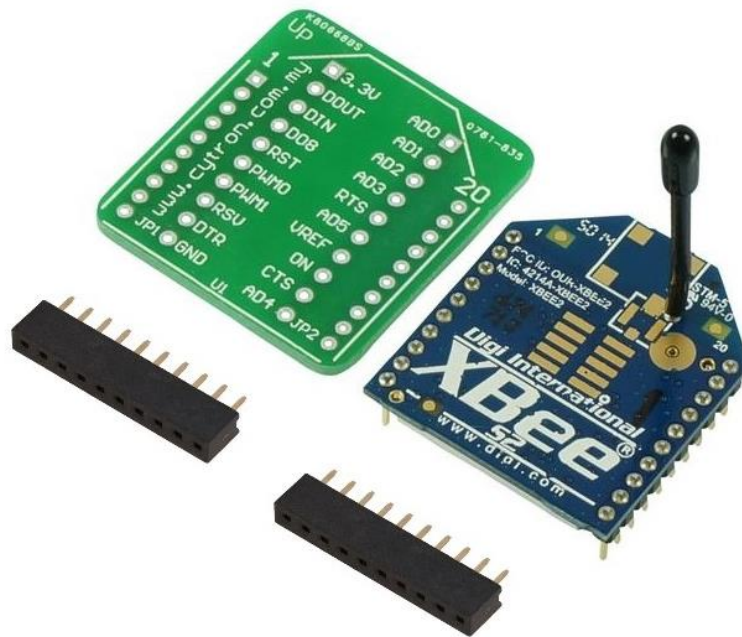


Figure 3.5 Xbee 2mW RF Wire Antenna

The **XBee 2mW RF module** such as the one above provides wireless connectivity to the warehouse robot at a cost effective rate. This low-power module will be put on top of the warehouse robots, and an extra four Xbee will be position at each corner of the test space. The Xbee on the robot will constantly communicate with the XBees in the corner, receiving the RSSI values sent from them. The Xbee 2mW RF modules only operate on 3.3V and has a built in antenna that has a maximum output of 2mW or +3dBm. Xbee modules can operate up to 120m maximum in outdoors and 60m maximum indoors.



Figure 3.6 Cytron's SKXBee starter kit module

For the four XBees located at the corner, they are installed on the Cytron SKXBee XBee starter kit module which acts as a breakout board and provide many useful features. The complication arise when trying to utilize the XBee modules since they require only 3.3V to operate and the pins are a 2mm pins which is not compatible with any printed circuit boards, prototype board or breadboard. Cytron's SKXBee can support communication range up to 100 meters and most importantly has a 5V Plug and Play USB UART interface that acts as its power supply source too. The module also has LEDs for power, Rx and Tx indicator.

The RFID module used in this project is the MFRC522 HF RFID module. The MFRC522 communicates with RFID tags up to 2cm using a 13.56MHz electromagnetic field, then sends the data to an Arduino board through SPI or I2C communication. When the robot equipped with the RFID reader moves near to the deployed tags, it will obtain its exact location through the information transmitted by the tags. By doing so, it can help reset the error for the DR calculation, as new current readings have been obtained when the robot approaches the RFID tags.

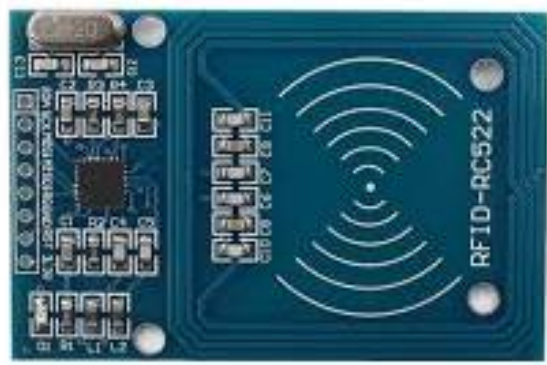


Figure 3.7
MFRC522 RFID Module



Figure 3.8
MIFARE RFID 13.56 MHz Tags

To break it down how the RFID reader and tags works, we must first understand on how the data are stored in the tags. Each tag contains 16 sectors and each sector has 4 blocks. Each block has 16 bytes totalling up to 64 bytes per block and 1K bytes per card. 3 blocks are used to store data in each block and 1 block is used to store secret authentication and access keys.

What we are interested in for this project is not to manipulate the information on top of the card; rather we would like to differentiate the tags from one another. The easiest ways is by reading the UID from each tag which is stored in sector 0 block 0. Within this particular sector and block, it is reserved for manufacturer data and not writable by user since it would damage and corrupt the tag.

Inside the 16 bytes manufacturer data, the first 4 bytes is the tags UID, which is what I extracted out from the tags every time a tag goes near to the RFID module. By only reading 4 bytes off the tags every time, this ensures efficient and fast reading speed (compared to reading the whole 1KB in a go). With the extracted UID, the tags can be easily differentiated and associated with a location within the test grid for the warehouse robot.

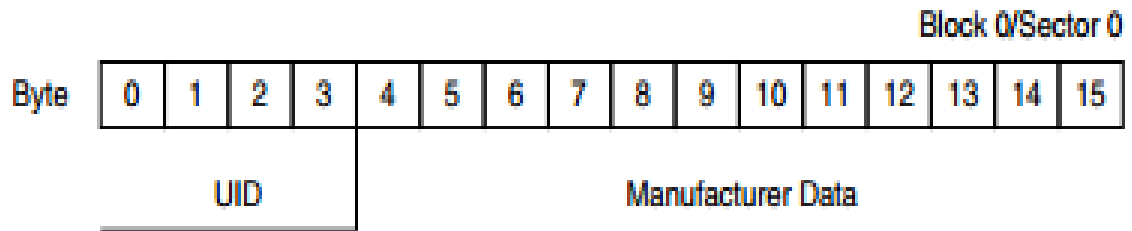


Figure 3.9 16 Bytes Block 0 of 125 kHz RFID tag

To connect to the internet for displaying the real time location of the robot to enable the user to see, a Wi-Fi shield is required. The Wi-Fi shield used in this project is a Cytron manufactured Rev2.0 Wi-Fi shield with a on board integrated ESP8266 module which supports 802.11 b/g/n, Wi-Fi Direct (P2P), integrated TCP/IP protocol stack.

The shield can be interfaced using hardware serial pins D0 and D1 on any Arduino, or users can choose to interface it using software serial, which the manufacturer kindly added pins on top of the shield so users can just use a jumper pin to connect the desired pins as the serial UART interface.

The brain of the Wi-Fi shield, the ESP8266 WROOM-02 is a low power 32-bit Microcontroller Wi-Fi module, which is based on the ESP8266 chip. Developed by Espressif System, the ESP8266 is the best option for connecting IoT projects to the internet since it only consume a very little power having an excellent power consumption rating.

The ESP8266 module that is equipped with a on board +3dBi PCB antenna only supports Wi-Fi frequency of 2.4 GHz, meaning it does not support Wi-Fi with frequency of 5 GHz, which can be found in most new version of routers or modems. It also supports encryption and security such as WEP, WEP64, WEP128, WPA-PSK, WPA2-PSK, TKIP and AES.

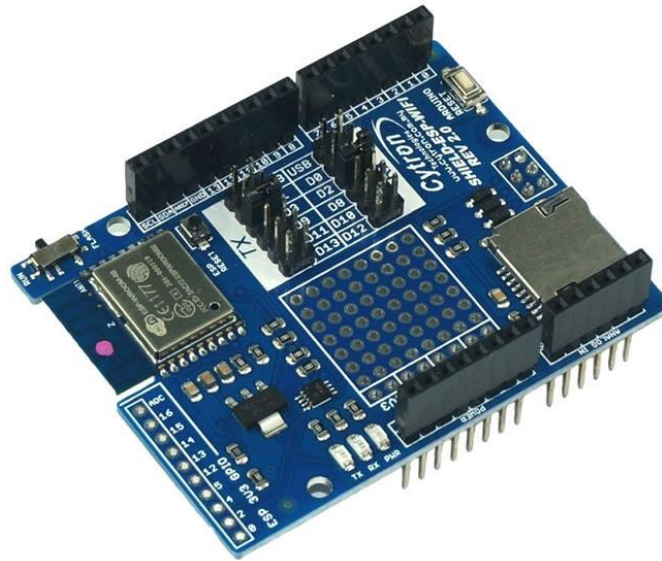


Figure 3.10 Cyttron ESP8266 Wi-Fi Shield

For this project, the area of movement would be divided into grids. RFID tags would be placed on every intersecting line in the grid. In total 4 XBee would be used in the grid system for transmitting and receiving the RSSI value. Each corner of the grid has one XBee installed, and an additional one would be installed onto the robot itself. When the robot is inside the grid, through RSSI readings from each XBee in the respective corner, the XBee installed on the robot can calculate its distance from each corner, relatively estimating its location inside the grid.

With all the aforementioned parts and components above integrated, a system combining DR, RFID (MFRC522) and RSSI (XBee) is created to perform accurate positioning for the warehouse robot. As the base of this project, the Arduino Mega 2560 is connected to the Wi-Fi Shield using software serial, although the Arduino Mega has several hardware serial ports. The reason for this is because the shield does not have enough pins to be plugged into the Arduino Mega, which requires extra cables flying over the top connecting them, plus extra codings and configuration have to be made, making it very inefficient. The Arduino communicates with the robot, Wi-Fi shield and XBee all through serial communications. However, the RFID module communicates with Arduino through SPI.

3.3 Hardware design and wiring

The aforementioned components are interconnected as the figure below.

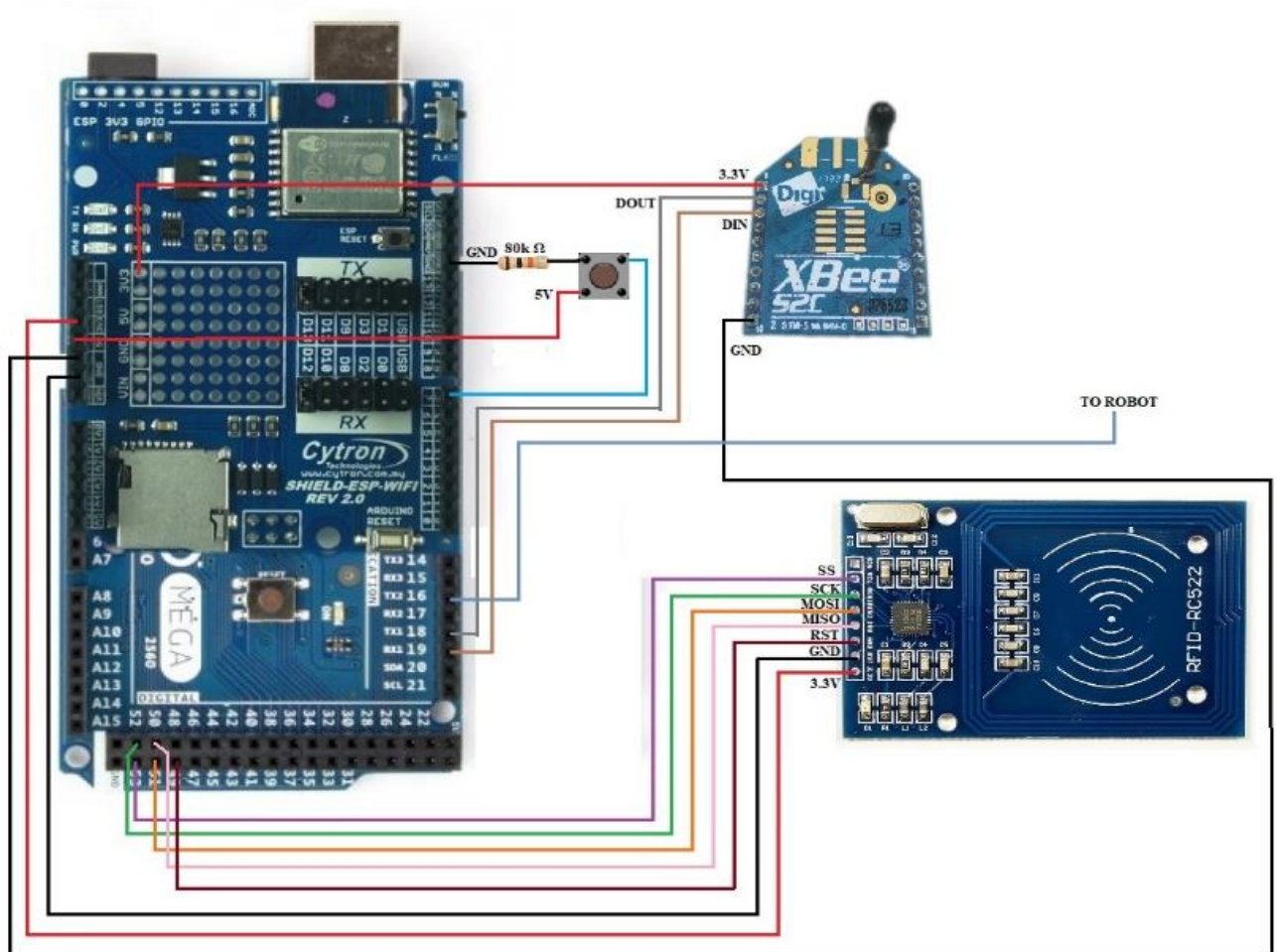


Figure 3.11 Hardware connections diagram

The Cytron ESP8266 Wi-Fi Shield is directly stacked upon the Arduino Mega 2560. Two jumper wires are used to choose the Rx and Tx pins for the Wi-Fi shield to communicate with Arduino. In this project, only the ground and Rx pin of the robot's UART pin will be used, since information is only sent to enable the robot to move, and no data is required to be sent back from the robot. Additionally, both the Arduino and on board STM32 microcontroller has their own power supply, hence the 5V power supply pin is omitted from the connection. However, they have to be grounded together, hence the GND pins are connected.

CHAPTER 3: SYSTEM DESIGN

The MFRC522 RFID module is interfaced with the Arduino Mega 2560 using SPI protocol. The 3.3V and GND pins are connected to the respective ports. Pin 49 on the Arduino Mega 2560 is used as the rest pin. The pins for SS (slave select), SCK (clock), MOSI (master-out-slave-in), and MISO (master-in-slave-out) is connected through pins 53, 52, 51 and 50 respectively.

The XBee module that acts as the coordinator is connected to the Arduino using 4 pins. A 3.3V power supply, GND, Rx and Tx pins for serial UART communication. Hardware serial Serial1 is used for the communication for the XBee module, hence it is connected through pin 18 as Tx and pin 19 as Rx for Serial1. The 3.3V power supply however, is not connected directly to the Arduino since the 3.3V port has been taken by the MFRC522 RFID module.

The solution to overcome this is fairly simple and direct, by utilizing a breakout prototype board on top of the Cytron Wi-Fi Shield. The manufacturer wired a few of the frequent used pin to the prototype including the 3.3V port. Consequently, the 3.3V needed will be directly supplied from the on board breakout 3.3V pin.

3.4 Software design and coding

3.4.1 Cytron ESP8266 Wi-Fi Shield

To implement the Wi-Fi shield, the correct libraries have to be imported first. Usually, generic libraries can be found in Arduino's IDE library, but since this shield is a after market proprietary device, the libraries have to be directly imported from the manufacturers website.

Below are the headers that should be included into the project file when working with the Cytron ESP8266 Wi-Fi shield. The software serial library is included too since we are using pin 12 and 13 as our software serial communication with the Arduino Mega 2560.

```
1 #include <CytronWiFiShield.h>
2 #include <CytronWiFiClient.h>
3 #include <SoftwareSerial.h>
4
```

Figure 3.12 Cytron ESP8266 Wi-Fi Shield Libraries

To setup for connection with the Wi-Fi, variables like Wi-Fi name, password, port that is going to be used and IPV4 address have to be created. A client instance should also be initialized. The "192.168.x.x" should be replaced with whatever IPv4 address of the device the shield is trying to connect to.

```
5 const char* ssid = ""; // Wi-Fi Name
6 const char* pass = ""; // Wi-Fi Password
7 const char* host = "192.168.x.x"; // IPV4 Address
8 const int httpPort = 80;
9 ESP8266Client client;
10
```

Figure 3.13 Wi-Fi Shield Initialization

CHAPTER 3: SYSTEM DESIGN

In the setup section, the serial communication with the Wi-Fi shield is commence and once success, the shield would attempt to connect to the Internet through Wi-Fi. Once connection is established, it will print out some information including firmware, connection status, RSSI and IP address. Any failure in connections will result in the shield looping indefinitely, requiring human assist to reset it.

```
13 //Start serial communication with WiFi shield
14 if(!wifi.begin(12, 13)){
15     Serial.println(F("Error talking to shield"));
16     while(1); // Stay here if fail to communicate with shield
17 }
18 Serial.println(wifi.firmwareVersion());
19 Serial.print(F("Status: "));Serial.println(wifi.status());
20
21 // Start WiFi connection
22 Serial.println(F("Starting wifi connection ..."));
23 if(!wifi.connectAP(ssid, pass)){
24     Serial.println(F("Error connecting to WiFi"));
25     while(true); // Stay in loop if fail to connect to WiFi
26 }
27 Serial.print(F("Connected to "));Serial.print(wifi.SSID());
28 Serial.print(F(", RSSI :"));Serial.println(wifi.RSSI());
29 Serial.print(F("IP address: "));Serial.println(wifi.localIP());
30
```

Figure 3.14 ESP8266 Serial and Internet Connection

In our application, data is sent to the web server through a simple HTTP GET request. The code snippet on the next page describes the creation of the request. The complete request string can be seen at line 97 to 100. The variable content is the UID read from the RFID that will be shown in the MFRC522 RFID module section.

The figure is on the next page.

```
95     if(client.connect(host, httpPort)){
96         String url = "/connect.php?test_str=" + content + "test_str2=" + content2;
97         Serial.print("requesting URL: ");
98         Serial.print(url);
99
100        client.print("GET " + url + " HTTP/1.1\r\n" +
101        "Host: " + host + "\r\n" +
102        "User-Agent: Location seeker\r\n" +
103        "Connection: close\r\n\r\n");
104
105        Serial.println("    GET Done");
106    }
107    else Serial.println("Failed to connect to server");
```

Figure 3.15 HTTP GET Request

Last but not least, every time a connection is opened, it is closed upon exiting the loop and reopen on the next request.

```
115     if(client.connected()){
116         client.stop();
117     }
```

Figure 3.16 Close Connection

3.4.2 MFRC522 RFID Module

To utilize the MFRC522 RFID Module, the libraries below have to be added. SPI library is needed since this module is interfaced through SPI connection.

```
1 #include <SPI.h>//include the SPI bus library
2 #include <MFRC522.h>//include the RFID reader library
```

Figure 3.17 MFRC522 Libraries

Initialization is done for the MFRC522 Module. Pins for reset and slave select are also saved as variables for use in the SPI protocol later on.

```
10 #define SS_PIN 53
11 #define RST_PIN 49
12 // instantiate a MFRC522 reader object.
13 MFRC522 mfrc522(SS_PIN, RST_PIN);
14 //create a MIFARE_Key struct named 'key'
15 //which will hold the card information
16 MFRC522::MIFARE_Key key;
17 String content= "";
18
```

Figure 3.18 MFRC522 Module Initialization

The built in function in the MFRC522 library is called to detect the presence of a RFID tag within its receiving range. Once detected, the UID of the target RFID tag would be saved in the string variable named content.

The UID is read byte by byte and then concatenated together, while transforming the value into uppercase for easier readability. However, in this project, the UID read from the RFID tag is directly appended with “%20” to transform it into a HTTP GET Request string form.

By changing “%20”：“%20” to “ 0”：“ ” at line 40, the user can get a normal UID. The difference with the GET Request string form is that in the string form, the space is appended with %20 since the URL is encoded and sent over the Internet using ASCII character set.

```

36 if ( mfr522.PICC_IsNewCardPresent() ) {
37     if ( mfr522.PICC_ReadCardSerial() ) {
38         for (byte i = 0; i < mfr522.uid.size; i++) {
39             if (i!=0) {
40                 content.concat(String(mfr522.uid.uidByte[i] < 0x10 ? "%200" : "%20"));
41             }
42             content.concat(String(mfr522.uid.uidByte[i], HEX));
43         }
44         content.toUpperCase();
45         Serial.println(content); //Debug use
46         if(content.substring(0)=="27%2037%2063%202A") // Do something ;
47         else if(content.substring(0)=="D7%20D1%206B%202A") // Do another thing;
48     }
49     content = "";
50     delay(500);
51 }

```

Figure 3.19 MFRC522 Reading UID

3.4.3 Serial Commands for Controlling Robot

Serial commands are save as bytes and then sent to the robot using serial UART. The commands for moving include forward, backwards, left, right, anticlockwise spin and clockwise spin. Last command is to reset the robot.

```

21 byte fwd[10]      = {255,254,1,0,0,0,16,0,0,0};
22 byte bwd[10]     = {255,254,1,0,0,0,16,0,0,2};
23 byte lft[10]    = {255,254,1,0,16,0,0,0,0,4};
24 byte rht[10]    = {255,254,1,0,16,0,0,0,0,0};
25 byte anticlkw[10] = {255,254,1,0,0,0,0,0,16,7};
26 byte clkw[10]   = {255,254,1,0,0,0,0,0,16,0};
27 byte rst[10]    = {255,254,1,0,0,0,0,0,0,0};

```

Figure 3.20 Serial Commands for Robot

The command bytes is sent to the robot using serial write.

```

56 Serial2.write(fwd, 10);
57 Serial2.write(bwd, 10);
58 Serial2.write(rst, 10);

```

Figure 3.21 Writing commands to robot

CHAPTER 3: SYSTEM DESIGN

Extra care must be made while coding since serial communication is heavily used for the main communication media. If not configured and done properly, data bits or bytes might get mixed up between modules which can result in the robot going haywire doing task we did not intend to.

3.4.4 XBee Coordinator

Hardware serial 1 is used for the connection between XBee and Arduino. Serial1 of the Arduino Mega 2560 have to be predefined.

```
2 #define xbee Serial1
```

Figure 3.22 Serial1 definition for XBee Module

Constant variables containing the address values of the router XBees are stored. A dummy address variable was created if none of the four routers are the intended destination. The upper address of each routers are the same, hence it is not required.

```
10 const String L = "0";  
11 const String R1L = "41809D36";  
12 const String R2L = "4176DB81";  
13 const String R3L = "41809D0C";  
14 const String R4L = "4176DC55";
```

Figure 3.23 Low Address of Router 1 to 4

To configure the XBee in AT mode, the coordinator XBee have to enter AT command mode. To exit AT command mode, the ATCN command have to be used.

The code for entering AT command mode and ATCN can be seen in the figure on the next page.

```

64 // ##### Enter AT Command Mode #####
65 xbee.print("+++");
66 delay(1000);
67 if(xbee.read()==79 && xbee.read()==75 && xbee.read()==13) Serial.print(""); //Serial.println("OK");
68 else Serial.print(""); //Serial.println("Error");
69
70 // ##### ATCN Command #####
71 xbee.print("ATCN\r");
72 delay(1000);
73 while((info = xbee.read())!=-1) Serial.print("");

```

Figure 3.24 AT Mode and ATCN Command

The whole procedure of obtaining a RSSI value from a single router are as follow.

- Step 1: Enter AT command mode
- Step 2: ATDL command
- Step 3: ATCN command
- Step 4: Send packet(s) to desired router
- Step 5: Enter AT command mode
- Step 6: ATDB command
- Step 7: ATCN command

ATDL is a command used to change the lower destination address of the desired router, whereas ATDB command is used for obtaining the RSSI value of the last received data packet.

The code snippet figure of ATDL and ATDB can be seen on the next page.

```

108 // ##### Change Destination Addr LOW #####
109 xbee.print("ATDL "); xbee.print(RL); xbee.print("\r");
110 delay(1000);
111 if(xbee.read()==79 && xbee.read()==75 && xbee.read()==13) Serial.print(""); //Serial.println("OK");
112 else Serial.print(""); //Serial.println("Error");
113 //Serial.print("ATWR ");
114 xbee.print("ATWR\r");
115 delay(1000);
116 if(xbee.read()==79 && xbee.read()==75 && xbee.read()==13) Serial.print(""); //Serial.println("OK");
117 else Serial.print(""); //Serial.println("Error");

```

Figure 3.25 ATDL Command

```

139 // ##### ATDB Command #####
140 xbee.print("ATDB\r");
141 delay(1000);
142
143 for(int g=0;g<2;g++){
144     db = xbee.read();
145     if(g==0){
146         if(db=='A' || db=='B' || db=='C' || db=='D' || db=='E' || db=='F'){ temp = ((int)db - 55); ischar = true;}
147         else { temp = ((int)db - 48)*16; ischar = false;}
148     }
149     else if(g==1){
150         if (db==13){
151             if(ischar) temp += 0;
152             else temp /= 16;
153         }
154         else if(db=='A' || db=='B' || db=='C' || db=='D' || db=='E' || db=='F') temp += ((int)db - 55);
155         else temp += ((int)db - 48);
156     }
157 }
158
159 rssi[num-1] = temp;
160 temp = xbee.read();
161 temp = 0;

```

Figure 3.26 ATDB Command

The value obtained using ATDB have to be recalculated since it returns either a byte or two bytes of value in Hex which represents one single value. The complication arises when the reply of RSSI values are not fixed in size, either one byte or two bytes. Suitable intervals and delays are added between all AT commands to ensure the responsiveness and overcome the latency and propagation delay of the XBee modules.

Data packets can be easily transmitted out regardless of the destination XBee device. Any messages that are enclosed in `xbee.print(“”)` function will be sent as text messages.

```
200 xbee.print("To Router "); xbee.print(num); xbee.print("\r");
201 delay(500);
```

Figure 3.37 Sending Data Packets to Desired Router

Lastly, the RSSI values for all four router XBees are stored in an array. The four values are computed and a comparing algorithm will define which four zone on the “playground” the robot is located on. The comparing method can be simplified into such.

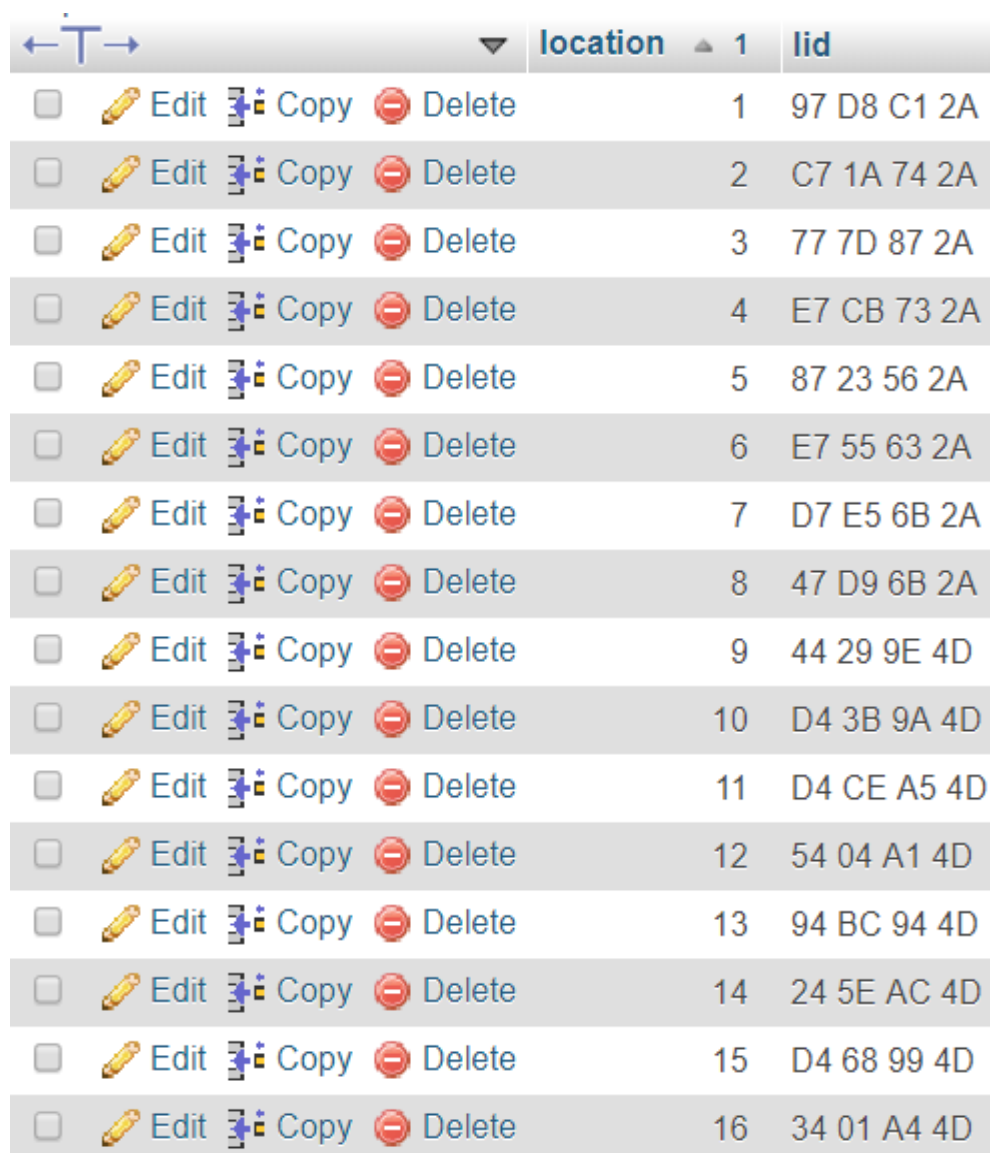
The first case is where there exist an absolute minimum value meaning no more than one value can have the minimum value, will be determined as the zone the robot is in. i.e. router 1 returns the smallest RSSI value, robot is in zone 1.

The second case where two and only two values have the same minimum value among the four, the other two values will be compared and the adjacent zone of the smaller value of the other two value will be determined as the zone the robot is in. i.e. router 1 and 2 returns the smallest RSSI value, compare value of router 3 and 4. If router 3 has smaller value, robot is in zone 1, or vice versa.

The third and last case is where at least three of the RSSI values are the same, then a cross calculation is done using four pairs of values router 1 + router 2, router 3 + router 4, router 1 + router 3 and router 2 + router 4.

3.4.5 MySQL Database

Two tables are created in total to store the data location of each and every RFID tag deployed within the “playground”. The table can be increased and decreased based on the users need or application required. The second table is created to store the location information sent from the Arduino to the localhost server.



	location	id	lid
<input type="checkbox"/> Edit Copy Delete		1	97 D8 C1 2A
<input type="checkbox"/> Edit Copy Delete		2	C7 1A 74 2A
<input type="checkbox"/> Edit Copy Delete		3	77 7D 87 2A
<input type="checkbox"/> Edit Copy Delete		4	E7 CB 73 2A
<input type="checkbox"/> Edit Copy Delete		5	87 23 56 2A
<input type="checkbox"/> Edit Copy Delete		6	E7 55 63 2A
<input type="checkbox"/> Edit Copy Delete		7	D7 E5 6B 2A
<input type="checkbox"/> Edit Copy Delete		8	47 D9 6B 2A
<input type="checkbox"/> Edit Copy Delete		9	44 29 9E 4D
<input type="checkbox"/> Edit Copy Delete		10	D4 3B 9A 4D
<input type="checkbox"/> Edit Copy Delete		11	D4 CE A5 4D
<input type="checkbox"/> Edit Copy Delete		12	54 04 A1 4D
<input type="checkbox"/> Edit Copy Delete		13	94 BC 94 4D
<input type="checkbox"/> Edit Copy Delete		14	24 5E AC 4D
<input type="checkbox"/> Edit Copy Delete		15	D4 68 99 4D
<input type="checkbox"/> Edit Copy Delete		16	34 01 A4 4D

Figure 3.38 Location Database

CHAPTER 3: SYSTEM DESIGN

The Location table has two columns in total, column one holding the location value and two rows can have the same value. The reason being more than one RFID tag can be deployed at a certain location for clustering purpose. However, the RFID UID stored in the second column acts as the primary key of the location table since all RFID tags has a unique identification value where no RFID UID can be exactly the same. The UID are stored in uppercase letters and separated by a space character.

The reason for the creation of the Location table is to store the RFID UID values on the cloud rather than storing them all on the Arduino. This reduces the workload of the Arduino since it is already busy dealing with many other operations, hence increasing the whole performance speed of the system in localizing the robot. Furthermore, when the system gets larger, the Arduino will eventually run out of memory.

timestamp	location	zone
2019-07-06 13:29:45	1	<i>NULL</i>
2019-07-06 13:30:02	0	<i>NULL</i>
2019-07-06 13:30:02	14	2
2019-08-16 14:30:02	5	3
2019-08-16 14:31:02	12	1
2019-08-16 14:30:02	5	3
2019-08-16 14:30:02	2	2
2019-08-16 14:30:02	10	1
2019-08-16 14:30:02	9	1
2019-08-16 14:30:02	8	2
2019-08-16 14:30:02	13	4
2019-08-16 14:30:02	7	3
2019-08-16 14:30:02	1	1
2019-08-16 14:30:02	7	3
2019-08-16 14:30:02	9	2

Figure 3.39 History Database

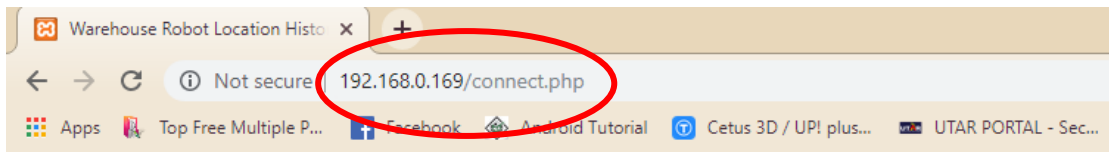
The second table namely the History table has three columns in total. Column number one stores the timestamp of each location update of the warehouse robot. Although the timestamp is not set as the primary key since it is not applicable in this project, the value for timestamp would be unique in this process since a single HTTP request would take more than one second to be generated and then successfully sent to the localhost server database.

Nonetheless, the two tables need to be created and the location table must have the RFID UID value and the respective location value. Each time the Arduino sends a HTTP GET request to the localhost server, the server would extract information from the Location table and then input into the History table. If both tables are missing, the whole system would not be able to work.

3.4.6 Webpage (HTML/PHP)

The webpage is a simple display with a table of the warehouse robot location history. The webpage is written using HTML for the structure and display of the webpage, and PHP was use for the server side scripting.

The webpage can be accessed through the URL [ip address of device running server]/[.php file]. In this project, the URL is simply 192.168.0.169/connect.php. The page auto refreshes every three seconds to update any new readings or new database entry. The page have to be a browser side refresh rather than auto update anytime the server receives new data is because PHP is an older server scripting language and does not support such functionality.



Warehouse Robot Location History

Timestamp	Location	Zone
2019-08-23 02:40:46	0	1
2019-08-23 02:41:59	0	4
2019-08-23 02:43:09	0	4
2019-08-23 02:43:16	9	4
2019-08-23 02:44:26	9	3
2019-08-23 02:45:39	9	3
2019-08-23 02:45:47	11	2
2019-08-23 02:46:53	11	2
2019-08-23 02:48:04	11	1
2019-08-23 02:48:16	1	1
2019-08-23 02:49:19	1	1
2019-08-23 02:50:29	1	1
2019-08-23 02:51:40	1	1
2019-08-23 02:52:51	1	3
2019-08-23 02:54:02	1	3
2019-08-23 02:55:13	1	3
2019-08-23 02:55:23	4	3

Figure 3.40 Webpage Display

For the PHP script, the connection between the client and database server have to be establish before performing any actions.

```
// ##### Setup connection between database #####
$con1 = mysqli_connect("localhost", "root", "", "location");

if (mysqli_connect_errno()){
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$con = mysqli_connect("localhost", "root", "", "testdb");

if (mysqli_connect_errno()){
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
//echo nl2br("Well done connect OK !!\n");
```

Figure 3.41 MySQLi Connection

CHAPTER 3: SYSTEM DESIGN

Variables are then created for the HTTP GET Method request.

```
// #####  
$test_str = "";  
$lid = "";  
$zon = "";  
if(isset($_GET["test_str"])&&isset($_GET["zone"])){  
    echo "OK  ";  
    $test_str = $_GET["test_str"];  
    $zon = $_GET["zone"];  
    if($test_str=="") $lid = "emp";  
    else                $lid = $test_str;  
}  
  
if($zon==""){  
    $spe_q = "SELECT zone FROM history ORDER BY timestamp DESC LIMIT 1";  
    $last_zone = $con->query($spe_q);  
    $zon = mysqli_fetch_array($last_zone);  
}
```

Figure 3.42 PHP GET Method

Both the UID and zone number sent from the Arduino will be stored and later used to update the database. If no new zone number is updated, the previous zone number would be used as the next input.

Using the acquired RFID UID from earlier, the server would compare and look for the location number with that particular UID and returns the value to be inputted into the next database update.

Finally, after the three variables timestamp, location number and zone number is prepared, they would be used as the input for the next database tuple input. Then HTML for the client side will process the webpage and display it in the browser.

```

// ##### Search for location number in location table #####
if (!$lid==""){
    if($lid=="emp"){
        $sq = "SELECT location FROM history ORDER BY timestamp DESC LIMIT 1";
        $last_loc = $con->query($sq);
        if (!$result0) {
            echo "SELECT failed: (" . $con->errno . ") " . $con->error;
        }
        $loc = mysqli_fetch_array($last_loc);
    }
    else{
        $q0 = "SELECT location FROM location where lid = ".$lid."";
        $result0 = $con->query($q0);
        if (!$result0) {
            echo "SELECT failed: (" . $con->errno . ") " . $con->error;
        }
        $loc = mysqli_fetch_array($result0);
    }
}

// ##### Store history in to history table #####

$timestamp = date("Y-m-d H:i:s") ;

$q1 = "INSERT INTO history (timestamp, location, zone) VALUES ('".$timestamp."','".$loc["location"]."','".$zon["zone"]."')";

if (!$con->query($q1) ) {
    echo "INSERT failed: (" . $con->errno . ") " . $con->error;
}
//echo "ADD Ok !!";
}

$q2 = "SELECT * FROM history ";
$result = $con->query($q2);

```

Figure 3.43 Search and Update Database

```

<!DOCTYPE html>
<html>
  <head>
    <title>Warehouse Robot Location History Table</title>
    <meta http-equiv="refresh" content="3; URL=/connect.php">
  </head>
  <body background="cyan">
    <big><h1>Warehouse Robot Location History</h1></big>

    <font size="4"><table border="1" cellspacing="1" cellpadding="1">
      <tr>
        <td> &nbsp;   Timestamp&nbsp;   </td>
        <td> &nbsp;   Location&nbsp;   </td>
        <td> &nbsp;   Zone&nbsp;   </td>
      </tr>

      <?php
        //if (!$lid==""){
        if($result!==FALSE){
          while($row = mysqli_fetch_array($result)) {
            printf("<tr><td> &nbsp;   %s </td><td> &nbsp;   %s&nbsp;   </td><td> &nbsp;   %s&nbsp;   </td></tr>",
              $row["timestamp"], $row["location"], $row["zone"]);
          }
          mysqli_free_result($result);
          mysqli_close($con);
        }
        //}
        mysqli_close($con1);
      ?>

    </table></font>
  </body>
</html>

```

Figure 3.44 HTML Code

CHAPTER 4: METHODOLOGY AND TOOLS

4.1 Methodology

Prototyping methodology would be used for describing the phases for this project.

The stages or phases can be seen through the diagram below.

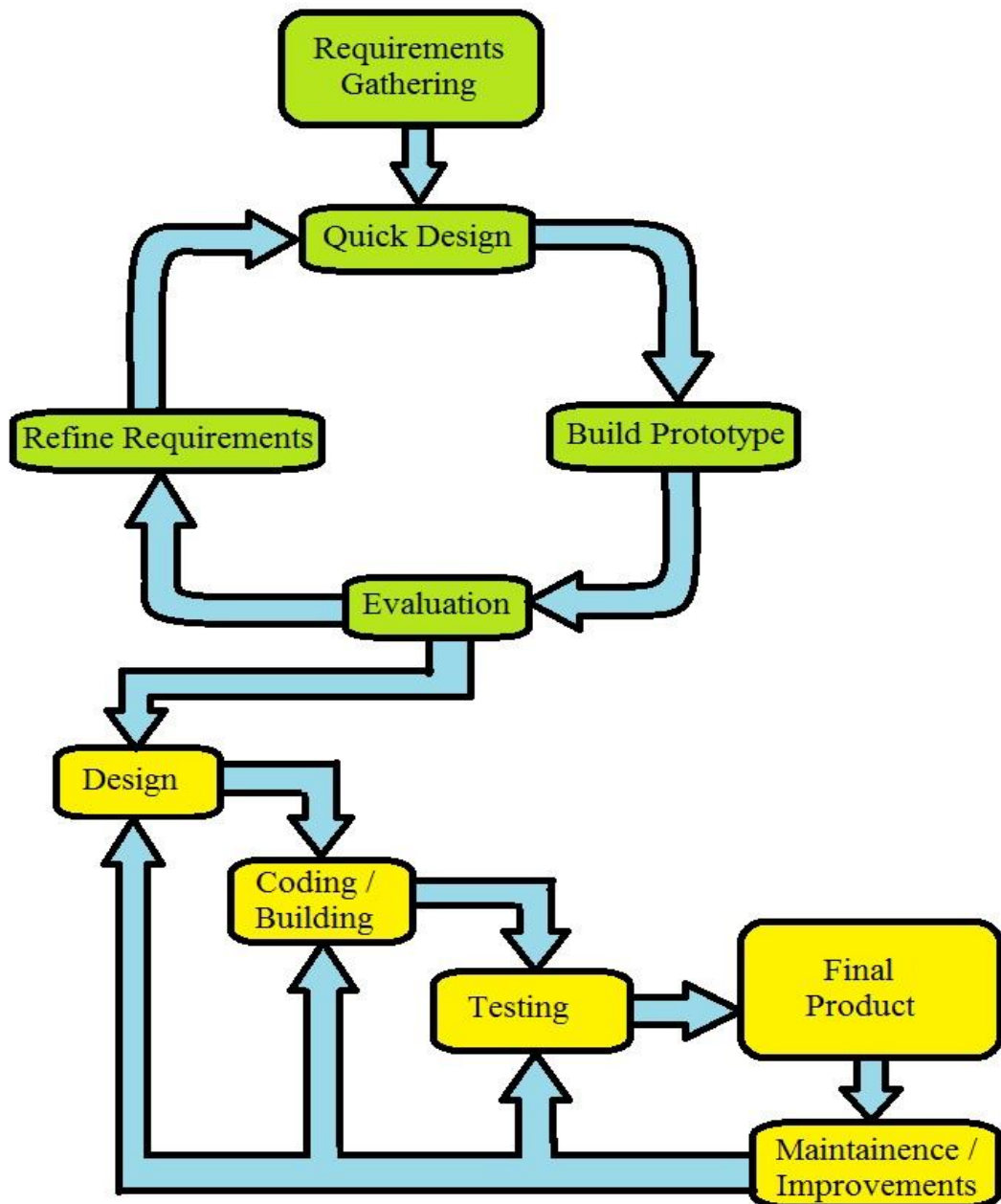


Figure 4.1 Prototyping Model

4.2 Software Tools

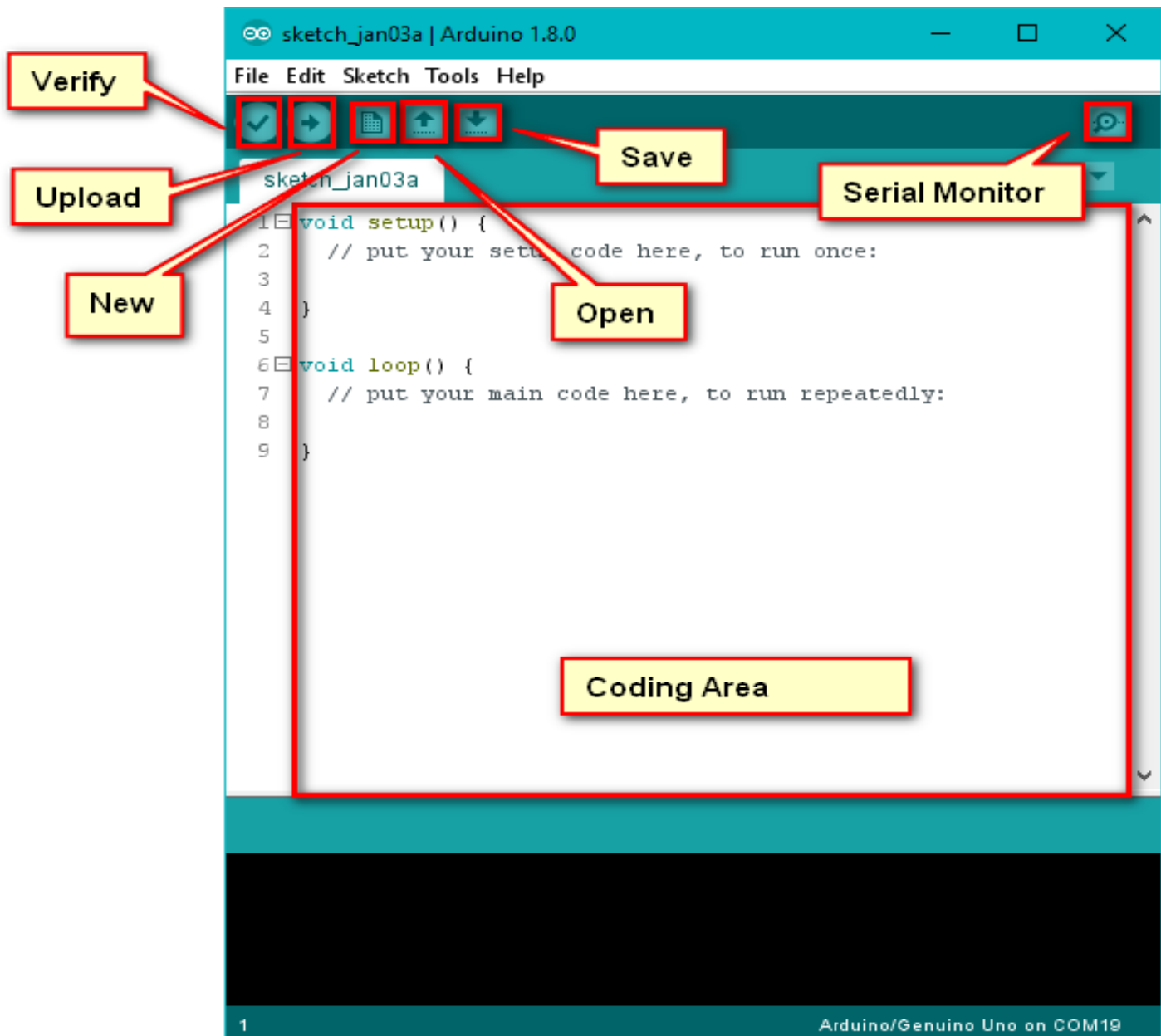


Figure 4.2 Arduino IDE with description

The Arduino Mega 2560 microcontroller will be programmed and loaded using the official Arduino IDE as above. The software can be downloaded from Arduino's official website. It is the best IDE to use for programming Arduino since it has a very good library system, one can even install an after market library directly within the software, or just simply load in a zip file, the IDE would do the rest for you. Furthermore, loading the ATmega2560 can be done by just pressing one key. Additionally, the compiler also has an error checking mechanism that helps check syntax and other problems.

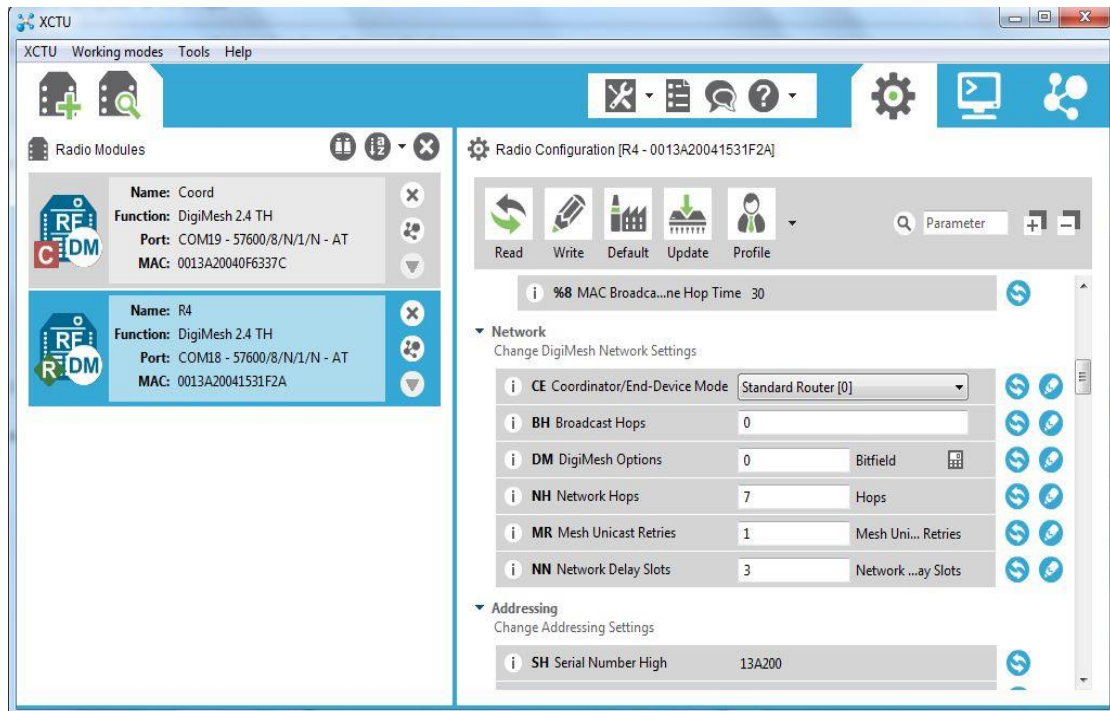


Figure 4.3 XCTU Software Diagram

The configuration on the XBee is done on the official software XCTU developed by Digi International. Before using the XBee modules, they must be either configured through XCTU as a coordinator, a router, or an end device. Within our mesh network, there will only be a single XBee module acting as a coordinator, and the other four would act as routers.

The XBee modules can be set to either API or AT mode. XBee in API mode need to be interfaced using the xbee.h library whereas XBee in AT mode can be directly interfaced using serial UART. AT command mode is used in this project for simplicity and reliability. AT commands are sent to the XBee through Arduino UART Tx as byte per byte and the return value is received through the Rx port of Arduino.

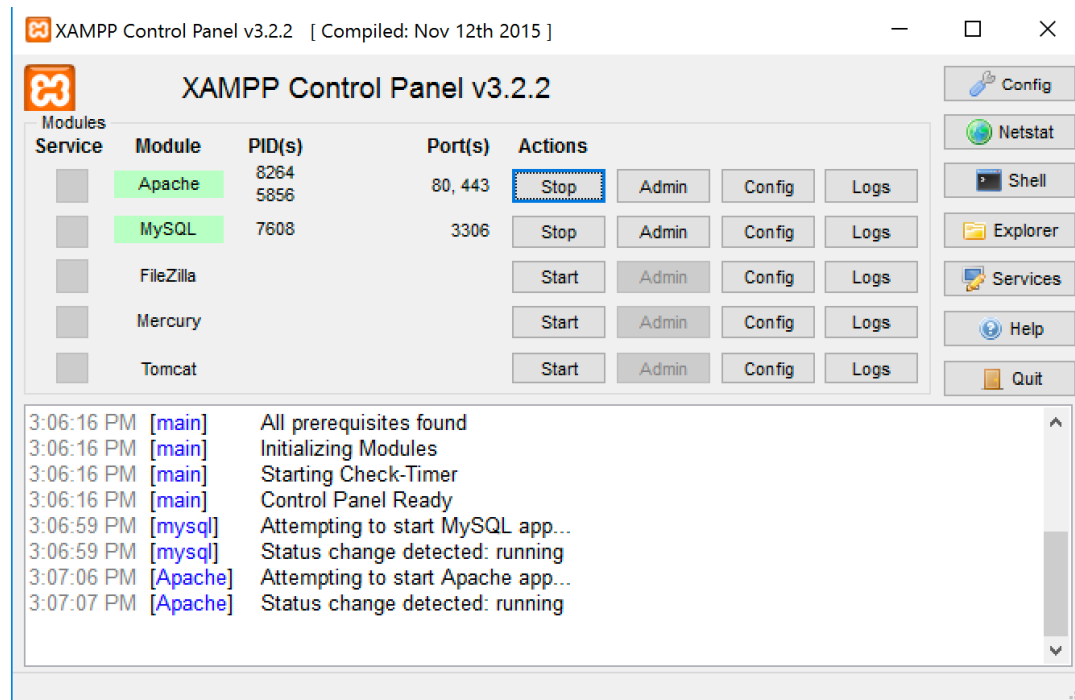


Figure 4.4 XAMPP for windows

The simple web page and database was built on top of the the local host server developed using Apache in Windows XAMPP - Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). The software is completely free under the GNU license since it is an open source software.

The website files can be stored in the apache directory and once the apache server is started, users can go to the desired webpage using the URL 127.0.0.1/[directory/filename], 127.0.0.1 being the commonly used IPv4 loopback address of the localhost. If users desire to connect to the server from another device or computer, the IP v4 address to be use would be the IP of the computer or device running the Apache server.

On the other hand, the database is created using a GUI after starting the MariaDB or previously known as MySQL. The creation of database is fairly easy through the MySQL interface.

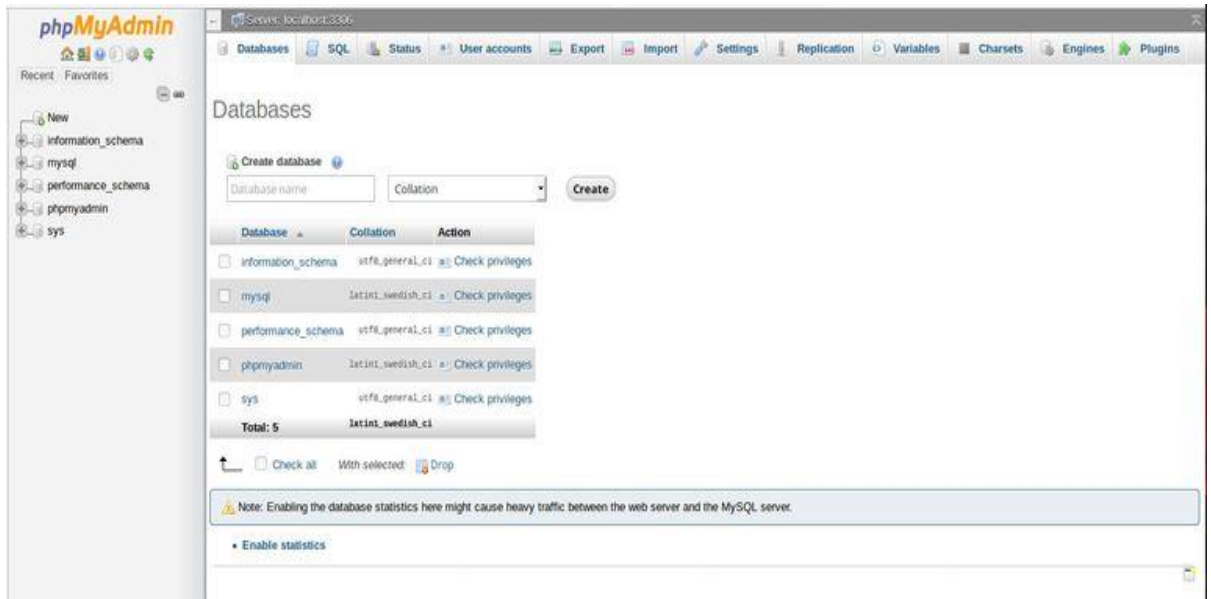


Figure 4.5 MySQL/MariaDB database GUI

To create a table in the localhost server database, the user can easily do so through using the GUI. The steps are shown below.

Step 1: On the index page with URL as such //localhost/phpmyadmin/index.php, locate the New option (first option) on the left-hand directory/workspace.

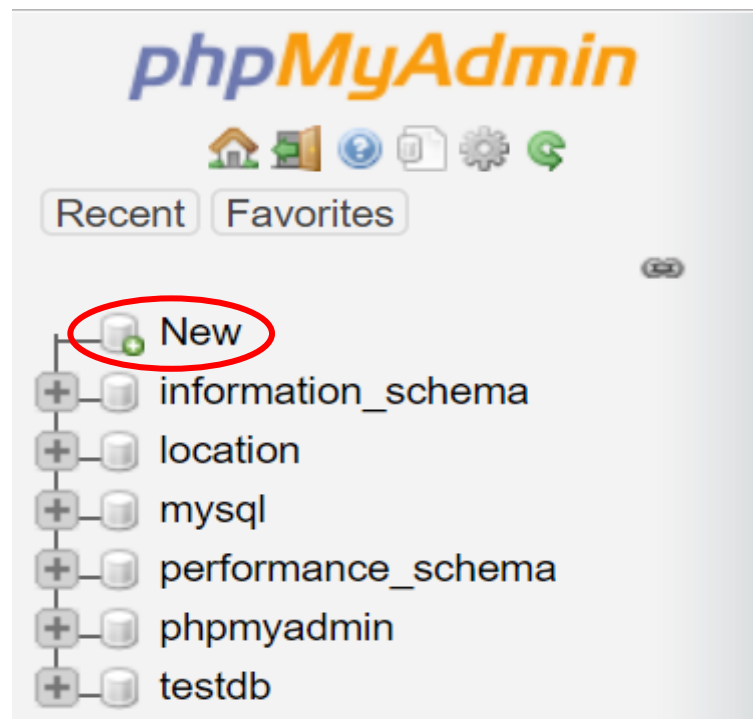


Figure 4.6 phpMyAdmin Directory/Workspace

Step 2: Input the desired name of the database table and press create.

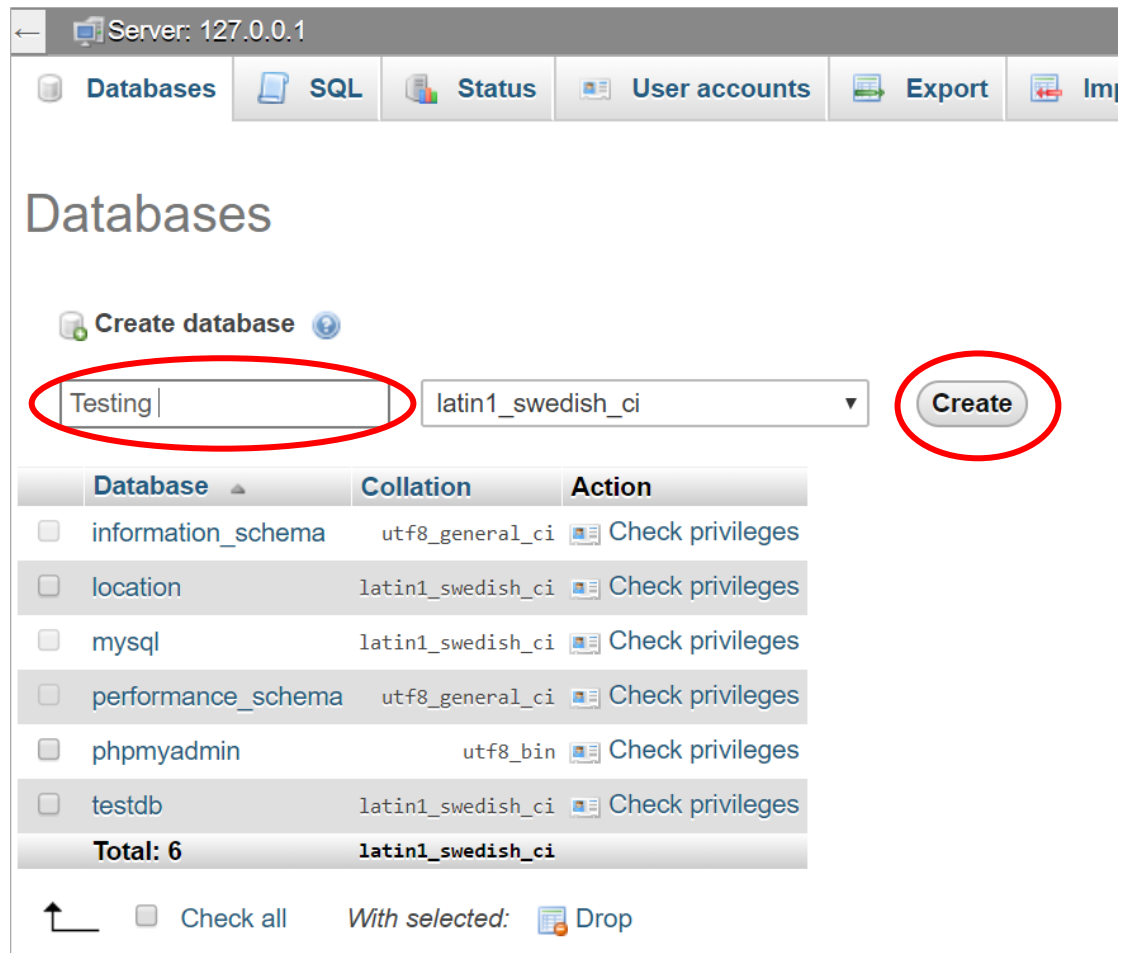


Figure 4.7 Naming for Database

Once the empty database is created, a pop-up message will indicate the database successfully created. The database can be seen on the left-hand directory and user will see an interface for creating tables within this database.

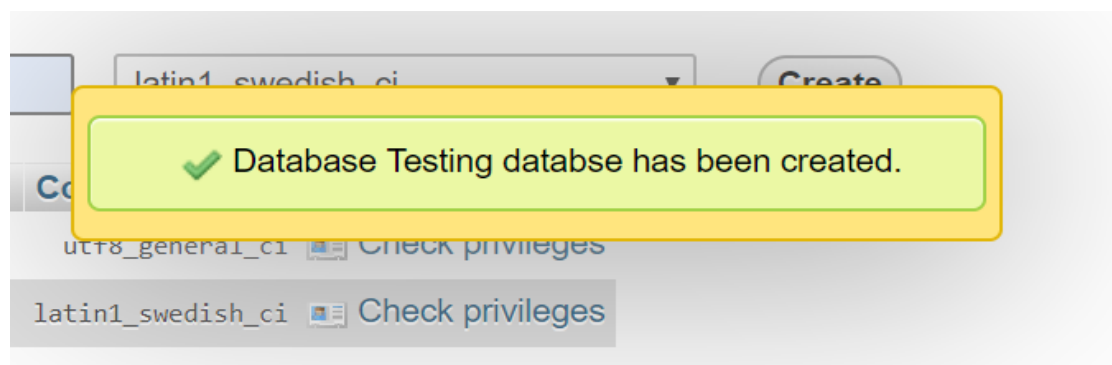
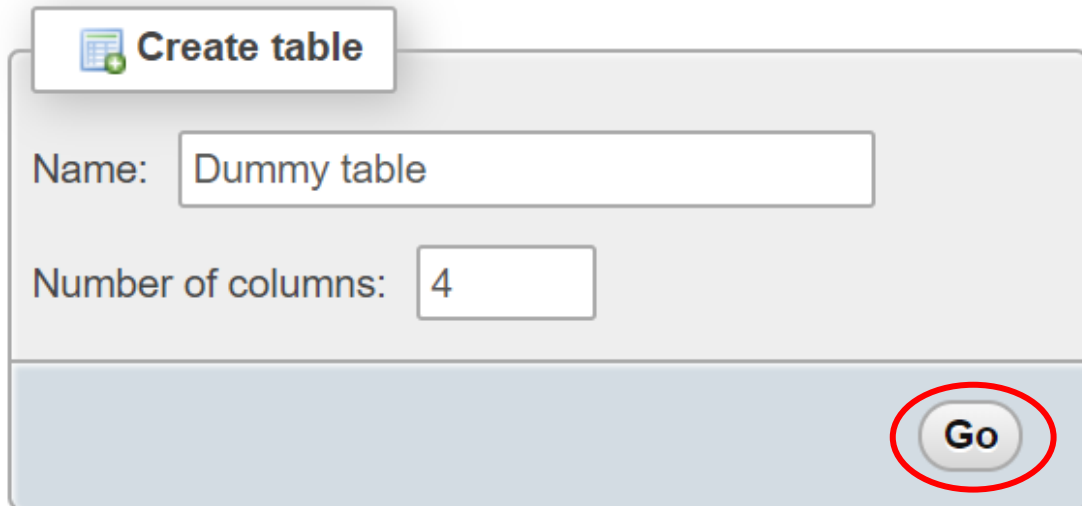


Figure 4.8 Database Created Pop-Up

Step 3: Input the desired name and choose the desired number of column for the new table. By pressing the Go button, the table will be created.



The image shows a web-based form titled "Create table". The form has a light gray background and a darker gray footer. At the top left, there is a small icon of a table with a green plus sign, followed by the text "Create table". Below this, there are two input fields. The first is labeled "Name:" and contains the text "Dummy table". The second is labeled "Number of columns:" and contains the number "4". In the bottom right corner of the form, there is a circular button with the text "Go" inside it, which is circled in red.

Figure 4.9 Creating Tables

Step 4: The next step will be to fill in the attributes such as names, type, length, default values etc for each and every column. Each column represents the data the user plans to store in them. Once the user has finished this step, the empty table is being created successfully.

To update any tables, there are mainly two methods. The first method is by simply going into the MySQL GUI and then normal SQL commands can be inputted. The second method which is implemented in this project is updating the tables tuple by tuple when a HTTP GET request is detected and received.

CHAPTER 5: FUNCTIONAL SPECIFICATION ANALYSIS AND VERIFICATION

The tests done for this project can mainly be categorized into two parts, testing of the MFRC522 RFID module and RFID tags and the testing of RSSI readings based on the XBee modules. Since the readings of both the RFID and RSSI are deterministic in the positioning system, ensuring the readings are consistent and stable are vital.

5.1 RFID Test

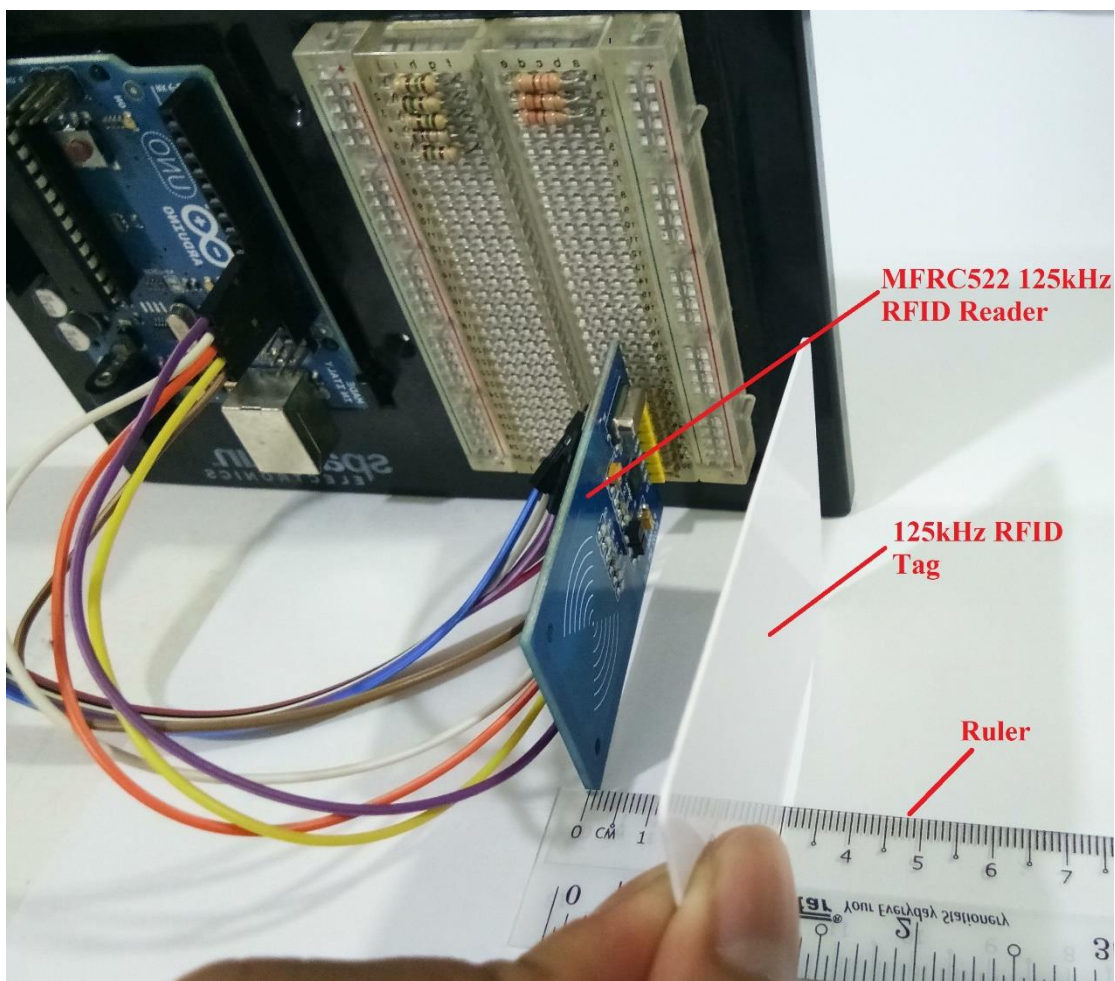


Figure 5.1 RFID Test Setup

The setup for the RFID test using the MFRC522 RFID module is as above. A ruler is placed starting from the RFID module. The RFID tag is placed at intervals of 0.5cm starting from 5cm decreasing gradually until 0cm, directly on top of the RFID module.

The result of the test can be seen in the table next page.

CHAPTER 5: FUNCTIONAL SPECIFICATION ANALYSIS AND VERIFICATION

Length (cm)	Able to Read
5.0	X
4.5	X
4.0	X
3.5	X
3.0	X
2.5	X
2.0	✓
1.5	X
1.0	X
0.5	X
0.0	X

Table 5.1 RFID full test result

Length (cm)	Able to Read
5.0	X
4.5	X
4.0	X
3.5	X
3.0	X
2.5	X
2.0	✓
1.5	X
1.0	X
0.5	X
0.0	X

Table 5.2 RFID UID test result

CHAPTER 5: FUNCTIONAL SPECIFICATION ANALYSIS AND VERIFICATION

The test for RFID From the RFID test result, we can deduce that the RFID is very consistent and reliable in the given range. The rate of failure is almost zero, since the data or UID can be read from the RFID tags every time it is in range. The full range of the MFRC522 RFID module is $2\pm 0.15\text{cm}$.

5.2 RSSI Test

RSSI test are carried out utilizing three to five XBees. The normal procedure of reading the RSSI values are looped and a total of 300 values are recorded. The test was done to determine the stability and reliability of the RSSI system.

5.2.1 RSSI Test 1

Test 1 is done using one XBee coordinator and two XBee routers. Router 1 and Router 2 are used in this test.

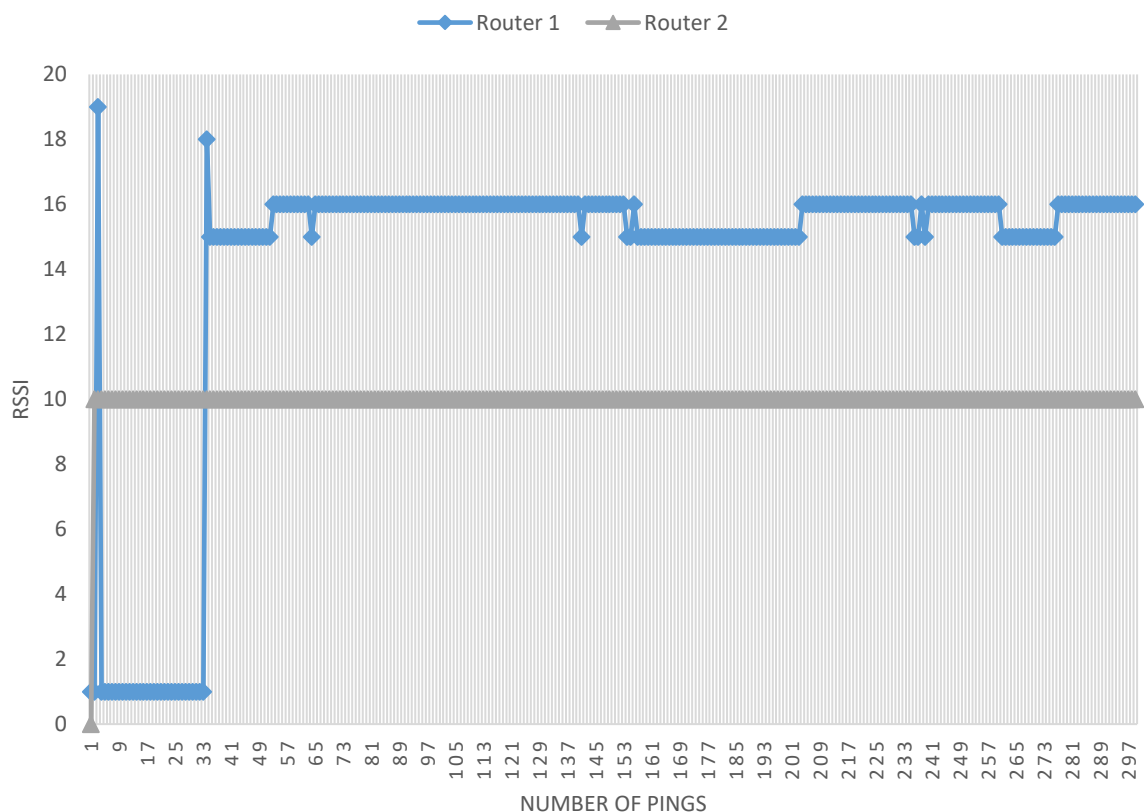


Figure 5.2 RSSI Test 1

CHAPTER 5: FUNCTIONAL SPECIFICATION ANALYSIS AND VERIFICATION

Based on the result of test 1, other than the initial drop down of router 1's RSSI, both readings of router 1 and 2 are stable after ping number 33+. The reason for the drop in reading was due to a misinterpretation of the data replied by the XBee module. A simple fix in the code solved the problem and the next test is continued.

5.2.2 RSSI Test 2

Test 2 is carried out with Router 1 and Router 2.

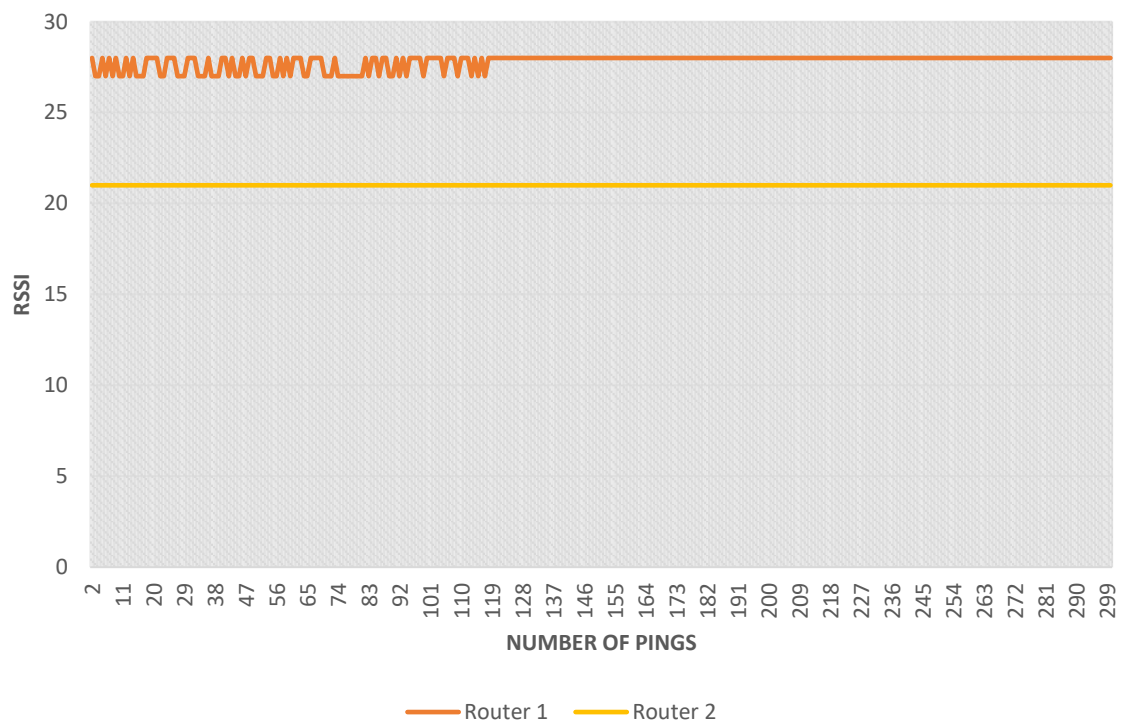


Figure 5.3 RSSI Test 1

The readings are considered stable and reliable in this test. Note that distance is not yet put into consideration in this few test, the purpose is just to determine the reliability and consistency of the readings in less constraints environment. Fluctuations in readings are due to the test are conducted indoors which has walls and furniture that deflect, refract or diffract the signal waves, sometimes giving a false reading of high gain in RSSI value.

5.2.3 RSSI Test 3

This test was done with all five of XBee modules. The coordinator was already installed on top of the robot and the other four XBee modules at the four corner.

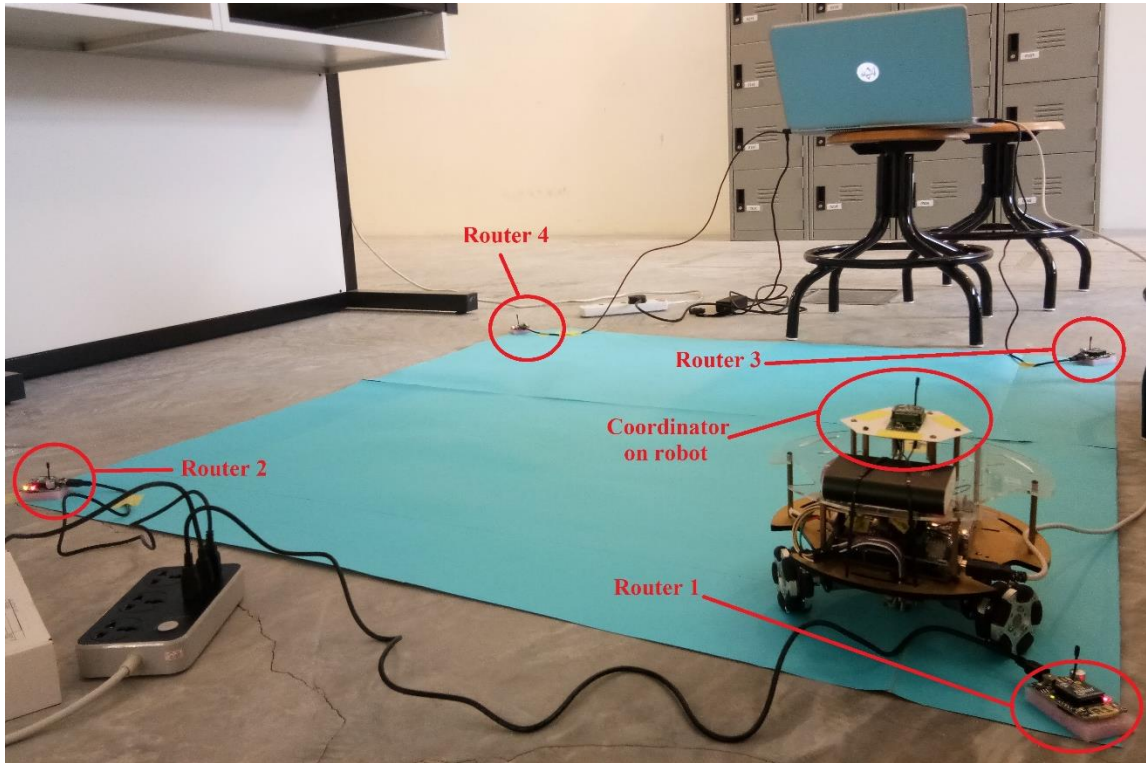


Figure 5.4 RSSI Test 3 Setup

Before the test could be run, the “playground” of the robot have to be fully set up like the figure above. It is best to tape the wires onto the ground or other places in order to reduce the interference of LOS between the coordinator XBee and the router XBees.

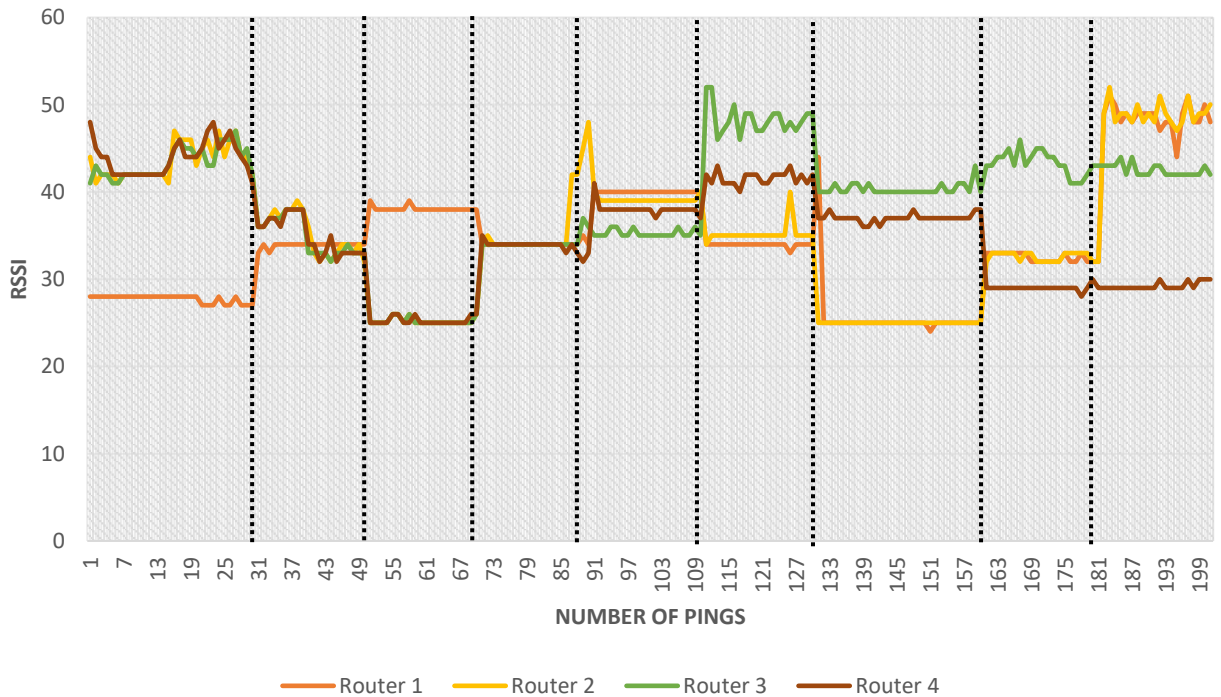


Figure 5.5 RSSI Test 3

The robot was moved around the grid and stayed at each position for a specific time to finish twenty readings. The first position has thirty readings since it was given more time since it need some time to stabilize after each startup. The third last position was also left there thirty readings just to give it a random scenario to see how it affected the final reading. In total, two hundred readings were recorder and plotted.

Notice that the higher the RSSI value gets, meaning the distance of router and coordinator is very far from each other, the more spikes there is in the graph. Comparing it to the graph plotted in RSSI test 1 and 2, the readings are much more stable and consistent in a shorter distance.

Hence, a balance between distance and stability have to be achieved when implementing XBees for the RSSI values.

5.3 Full System Test

After integrating both the RFID and RSSI system together, it is time to run a full system diagnose test. The setup for the test is as below.

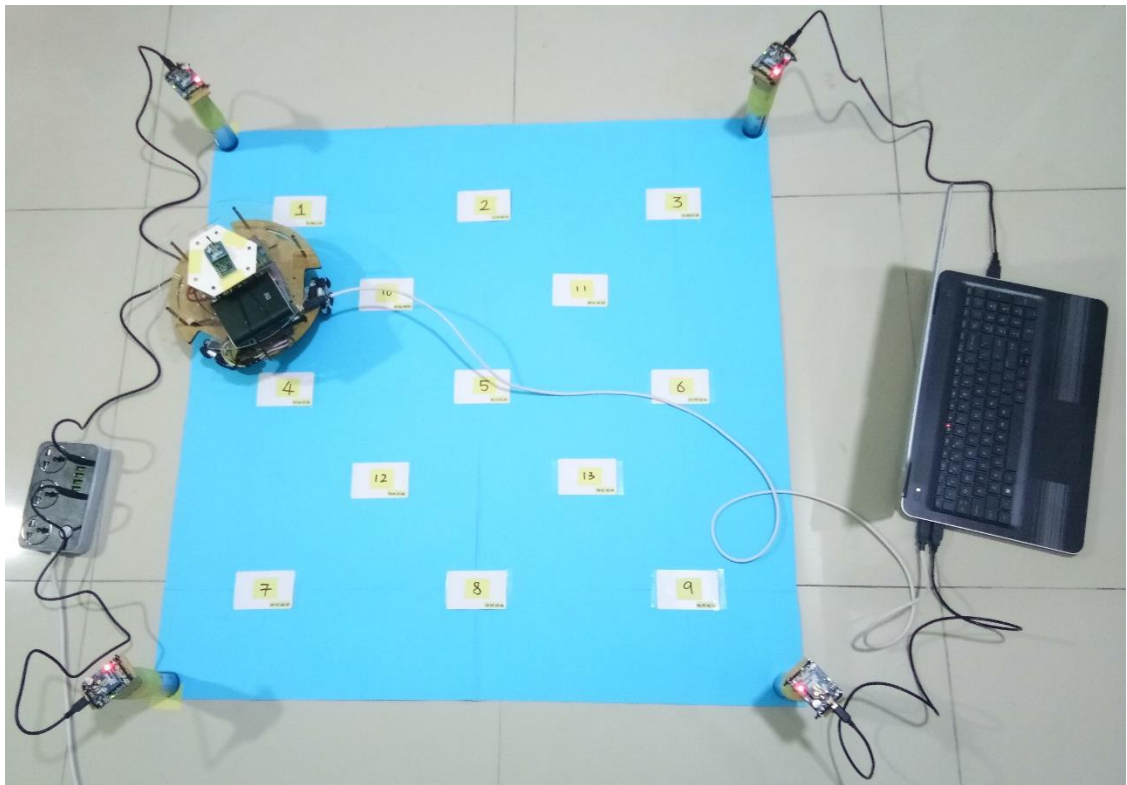


Figure 5.5 Full System Test

The test was done by moving the robot across the “playground”, passing over different RFID tags and entering different zones. The test result were very accurate.

The results for the test can be seen on the next page.

CHAPTER 5: FUNCTIONAL SPECIFICATION ANALYSIS AND VERIFICATION

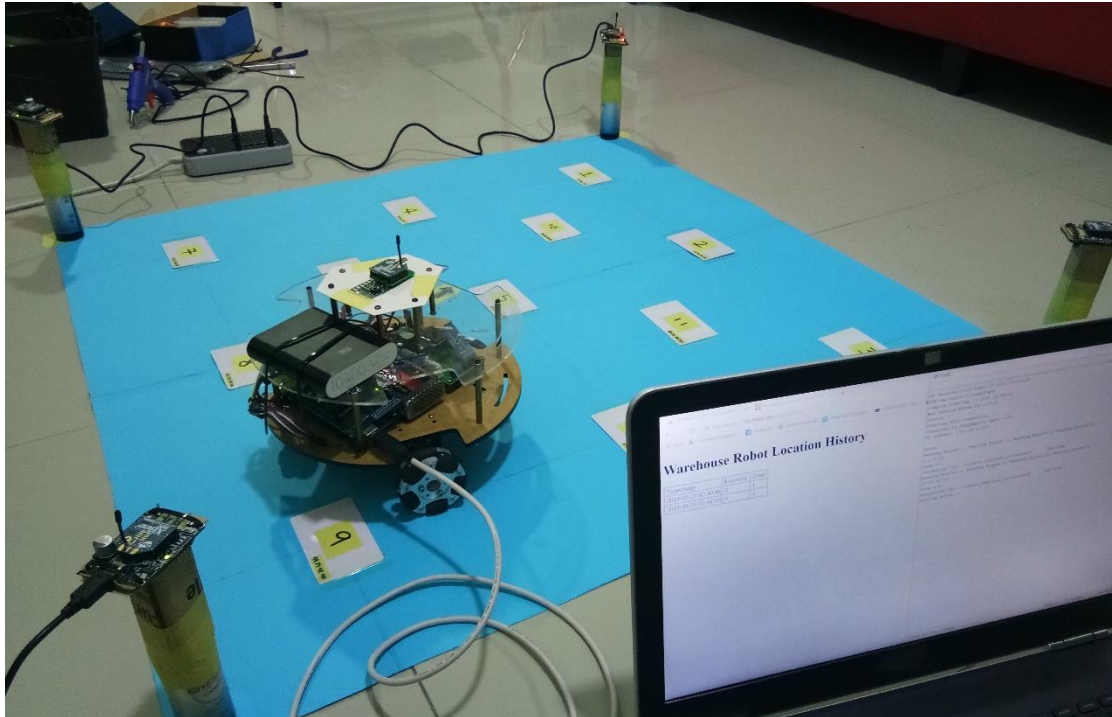


Figure 5.6 Zone 4 Test

Timestamp	Location	Zone
2019-08-23 02:40:46	0	1
2019-08-23 02:41:59	0	4

Figure 5.7 Zone 4 Test Result

The result of the RSSI shows an accurate calculation on which zone the robot is in. The location value is zero is because the robot have not pass by any RFID tags, hence the previous value would be repeated.

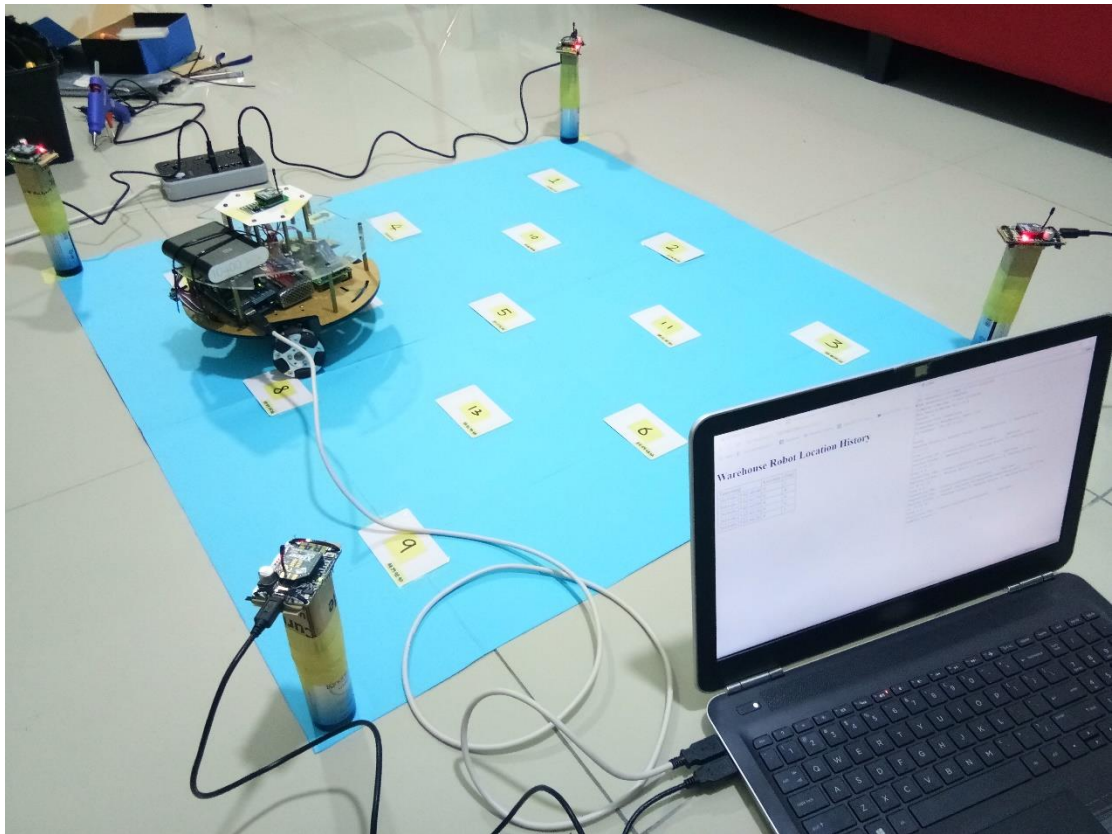


Figure 5.8 Zone 3 Test

Timestamp	Location	Zone
2019-08-23 02:40:46	0	1
2019-08-23 02:41:59	0	4
2019-08-23 02:43:09	0	4
2019-08-23 02:43:16	9	4
2019-08-23 02:44:26	9	3

Figure 5.9 Zone 3 test Result

The robot passes by RFID tag number nine before going into zone 3, located in zone four, proving the RFID system is working fine. After going into zone 3, the obtained RSSI value indicates that the robot resides in zone 3, proving the RSSI system working as intended.

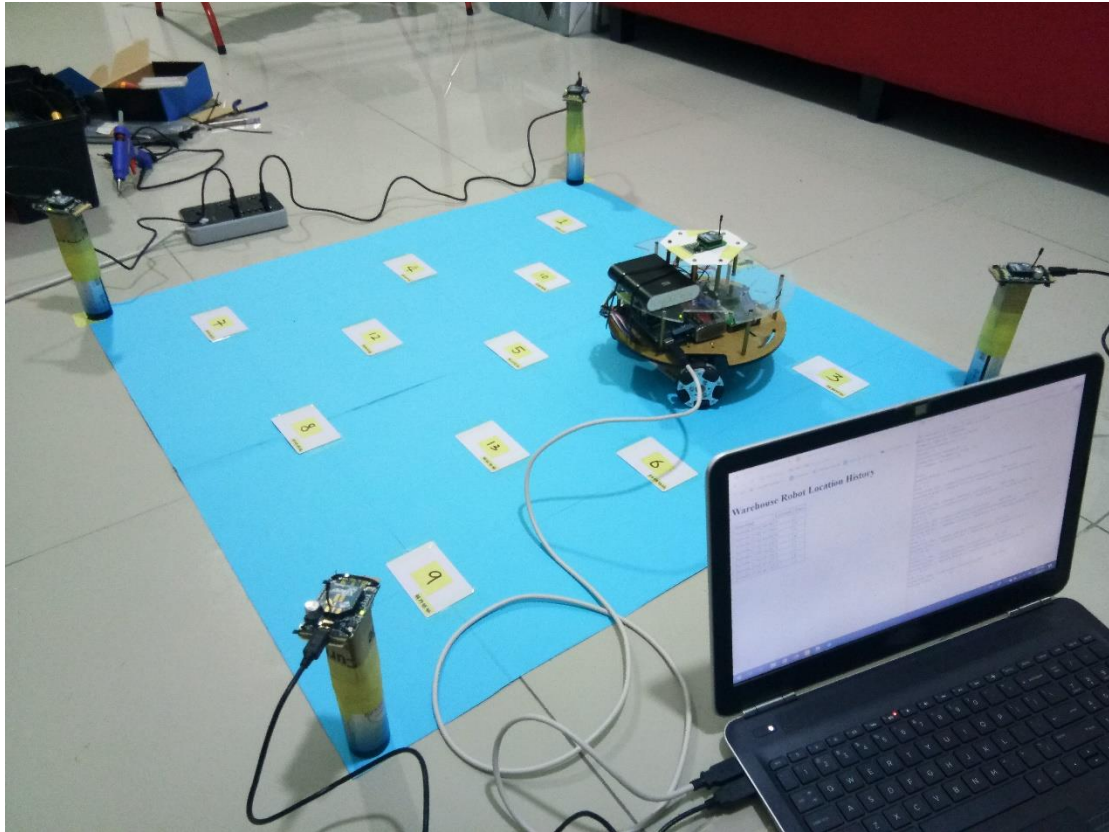


Figure 5.10 Zone 2 Test

Timestamp	Location	Zone
2019-08-23 02:40:46	0	1
2019-08-23 02:41:59	0	4
2019-08-23 02:43:09	0	4
2019-08-23 02:43:16	9	4
2019-08-23 02:44:26	9	3
2019-08-23 02:45:39	9	3
2019-08-23 02:45:47	11	2
2019-08-23 02:46:53	11	2

Figure 5.11 Zone 2 Test Result

The robot passes by RFID tag 11 that resides in zone 2, showing RFID system is working as intended. A further RSSI system test proved that it is working since it indicates the robot is in zone 2.

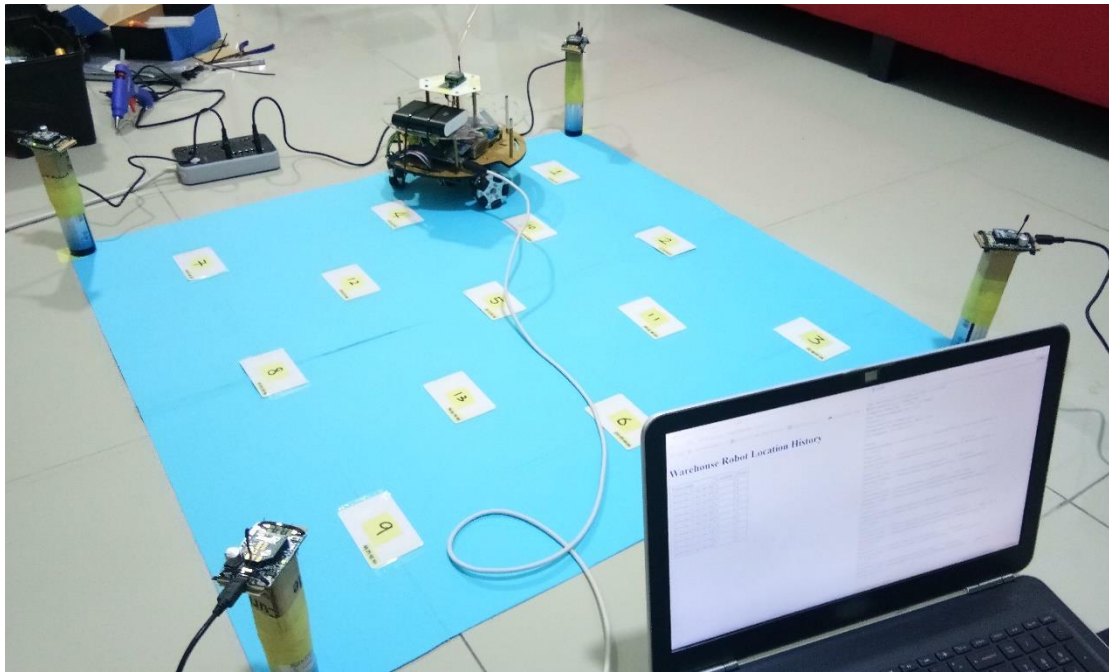


Figure 5.12 Zone 1 Test

Timestamp	Location	Zone
2019-08-23 02:40:46	0	1
2019-08-23 02:41:59	0	4
2019-08-23 02:43:09	0	4
2019-08-23 02:43:16	9	4
2019-08-23 02:44:26	9	3
2019-08-23 02:45:39	9	3
2019-08-23 02:45:47	11	2
2019-08-23 02:46:53	11	2
2019-08-23 02:48:04	11	1
2019-08-23 02:48:16	1	1

Figure 5.13 Zone 1 Test Result

In this test , the system first detects the robot in zone 1 using RSSI before detecting any RFID tags. Then the robot is brought to tag number one and the result was immediately updated and displayed on the browser.

CHAPTER 5: FUNCTIONAL SPECIFICATION ANALYSIS AND VERIFICATION

Finally, the robot is brought to various points within the “playground” and the results prove that the system is indeed working as intended to. Switching between multiple points did not affect the readings of the system. The only constraint is whenever the RSSI system is running, the robot does not change its location, since it needs time to finish reading all four RSSI values from the XBee routers.

Timestamp	Location	Zone
2019-08-23 02:40:46	0	1
2019-08-23 02:41:59	0	4
2019-08-23 02:43:09	0	4
2019-08-23 02:43:16	9	4
2019-08-23 02:44:26	9	3
2019-08-23 02:45:39	9	3
2019-08-23 02:45:47	11	2
2019-08-23 02:46:53	11	2
2019-08-23 02:48:04	11	1
2019-08-23 02:48:16	1	1
2019-08-23 02:49:19	1	1
2019-08-23 02:50:29	1	1
2019-08-23 02:51:40	1	1
2019-08-23 02:52:51	1	3
2019-08-23 02:54:02	1	3

2019-08-23 02:55:13	1	3
2019-08-23 02:55:23	4	3
2019-08-23 02:56:29	4	1
2019-08-23 02:56:38	5	1
2019-08-23 02:57:45	5	2
2019-08-23 02:58:57	5	2
2019-08-23 02:59:03	3	2
2019-08-23 02:59:08	3	2
2019-08-23 03:00:11	3	4
2019-08-23 03:00:16	13	4
2019-08-23 03:00:22	13	4
2019-08-23 03:00:28	13	4
2019-08-23 03:00:33	13	4
2019-08-23 03:00:39	13	4
2019-08-23 03:00:44	13	4

Figure 5.24 All Rounded Test Result

CHAPTER 6: CONCLUSION

The aim to build a fully functional warehouse robot equipped with IPS is fulfilled through combining various sensors and algorithms, integrated and working together to perform the task of localization. Warehouse robots with IPS can efficiently transfer packages and parcel without human aid, hence can increase the efficiency and productivity of a warehouse drastically if implemented successfully. Furthermore, a well design warehouse robot is not as bad as human when discussing about error prone since robots do not get tired and distracted.

Using RFID as the base of the IPS is brilliant as it is very a low power mechanism which uses passive RFID tags that do not require constant external power source for it to work. In addition, the reader itself also only uses a minimal 3.3 Volts to function, making it energy efficient. Furthermore, using an Arduino Mega as the microcontroller for the sensors and components only and not including the encoders is also a good idea since the encoders needs a higher voltage which requires extra circuitry and hardware to implement. Hence, the hard work of driving the encoders is given to the on board STM32 of the robot.

Integrating the Xbee modules into the IPS is also a novel idea since it act as a secondary level of localization and asset tracking. Based on the RSSI values, the robot can be tracked and determine which zone the robot is in. Whenever the RFID system fails to pinpoint the exact location of the robot, the RSSI system will be able to determine the zone the robot is in. This fulfils the objective of asset tracking and localization of the warehouse robot.

There are several constraints and challenges that are notable in this project. The RFID system is actually the best and fastest way as a positioning system, but due to the mechanical error of the robot which is it tends to slightly deviate from its intended path. The biggest challenge in implementing the RSSI system is utilizing the XBee modules. Since the XBee modules were not built for such application, it does not give the best

CHAPTER 6: CONCLUSION

result in RSSI values especially in a constraint environment as such in this project. Working indoors with many objects around definitely affect the RSSI readings a lot, sometimes resulting in error when trying to position the robot.

In the future, to assure the robot is always on its correct path, an image processing technique can be added to this project. Lines of the grid will be used as a guide to assure the robot will always only move in line. Additionally, the system can be further improved by changing the XBee modules into XBee modules with higher output power, or completely scrap the XBee and implement the RSSI positioning system with other wireless system with greater range and output power, i.e. LoRa. Implementing the RSSI system may perhaps be possible by just utilizing the Wi-Fi module since the Wi-Fi module is already a necessary part for the system.

Finally yet importantly, this project serves as a very good opportunity to learn how to integrate and combine different parts and devices to become a workable IPS. Though the journey in completing was tough and very challenging, much have been learned through this project.

REFERENCES

REFERENCES

- Bekkali, A, Sanson, H & Matsumoto, M, 2007, *RFID Indoor Positioning based on Probabilistic RFID Map and Kalman Filtering*, Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communication
- Chon, HD, Jun, S, Jung, H & An, W, 2005, *Using RFID for accurate positioning*, Positioning, vol. 3, no. 1&2, pp. 32–39
- Dabove, P, Ghinamo, G & Lingua, AM, 2015, *Inertial sensors for smartphones navigation*, SpringerPlus, vol. 4, no. 1, article 834
- Digiampaolo, E & Martinelli, F, *A passive uhf-RFID system for the localization of an indoor autonomous vehicle*, IEEE Transactions on Industrial Electronics, vol. 59, no. 10, pp. 3961–3970, 2012
- Drane, C, Macnaughtan, M & Scott, C, 1998, *Positioning GSM telephones*, IEEE Communications Mag., vol. 36, no. 4, pp. 46–54, 59
- Ganghiraj, S & Kumar, VD, 2015, *RFID assisted navigation system for transportation*, International Journal of Innovation Research in Science, Engineering and Technology, vol. 4, Special Issue 13
- Haverinen, J & Kemppainen, A, 2009, *Global indoor self-localization based on the ambient magnetic field*, Robotics and Autonomous Systems, vol. 57, no. 10, pp. 1028–1035
- Haykin, S, 2005, *Cognitive radio: brain-empowered wireless communications*, IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 201–220
- Hightower, J & Borriello, G, 2001, *Location systems for ubiquitous computing*, Computer, vol. 34, no. 8, pp. 57–66.
- Kriz, P, Maly, F & Kozel, T, 2016, *Improving indoor localization using Bluetooth low energy beacons*, Mobile Information Systems, vol. 2016, Article ID 2083094
- Lionel, MN, Liu, Y, Lau, YC & Patil, AP, 2004, *LANDMARC: indoor location sensing using active RFID*, Wireless Networks, vol. 10, no. 6, pp. 701–710
- Liu, H, Darabi, H, Banerjee, P & Liu, J, 2007, *Survey of wireless indoor positioning techniques and systems*, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 37, no. 6, pp. 1067–1080
- Mulloni, A, Wagner, D, Barakonyi, I & Schmalstieg, D, 2009, *Indoor positioning and navigation with camera phones*, IEEE Pervasive Computing, vol. 8, no. 2, pp. 22–31
- Qin, H, Peng, Y & Zhang, W, 2017, *Vehicles on RFID: Error Cognitive Vehicle Localization in GPS-less Environments*, IEEE Transactions on Vehicular Technology, vol. 66, no. 11, pp. 9943–9957

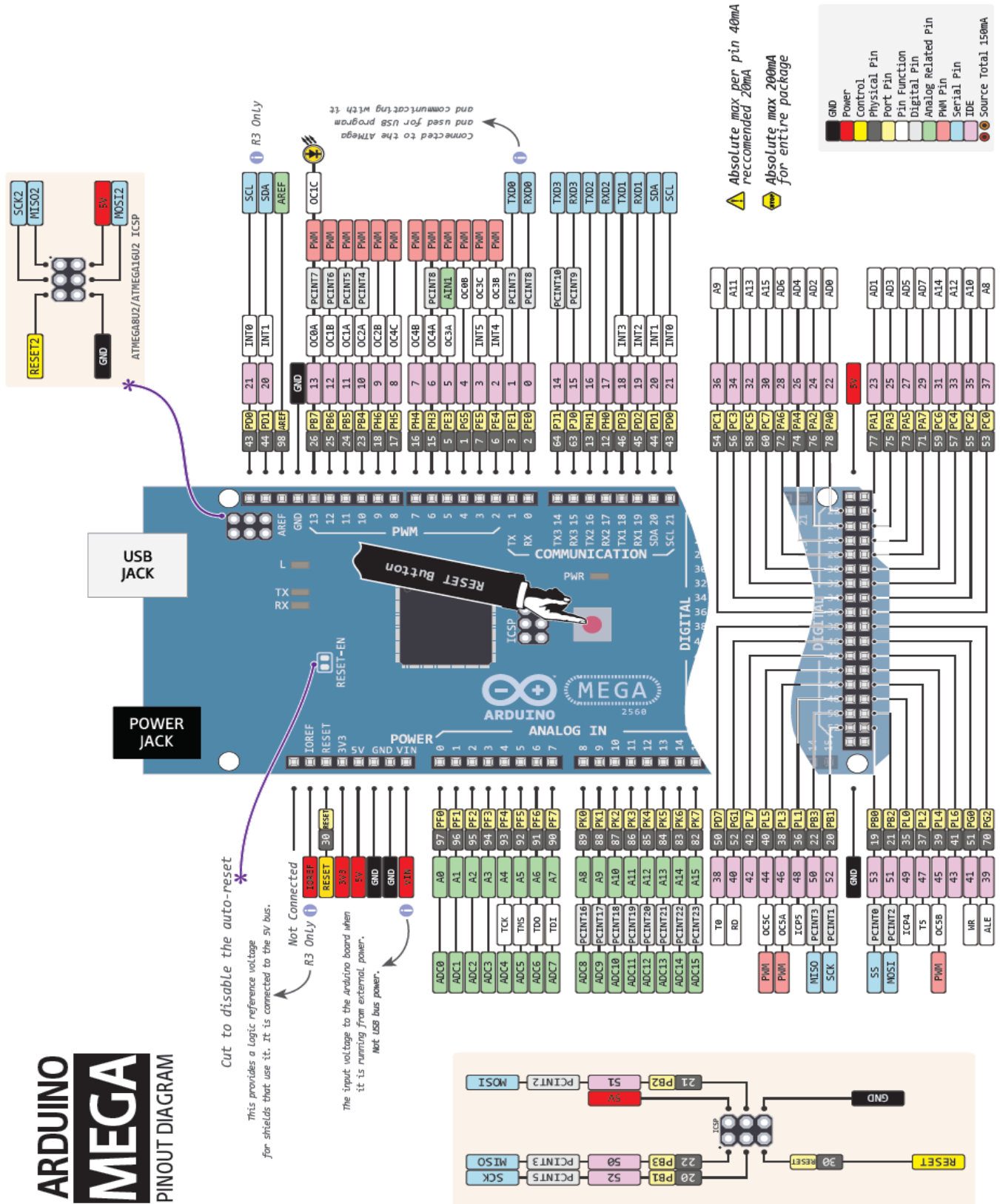
REFERENCES

Saab, SS & Nakad, ZS, 2011, *A Standalone RFID Indoor Positioning System Using Passive Tags*, IEEE Transactions on Industrial Electronics, Vol. 58, No. 5, pp. 1961–1970

Want, R, Hopper, A, Falcão, V & Gibbons, J, 1992, *The active badge location system*, ACM Transactions on Information Systems, vol. 10, no. 1, pp. 91–102

Xu, GC, 2007, *GPS Theory, Algorithms and Applications*, 2nd ed. Springer Berlin Heidelberg, New York

ARDUINO MEGA PINOUT DIAGRAM



Indoor Positioning System For Warehouse Asset Tracking

INTRODUCTION :

A WAREHOUSE ROBOT THAT CAN LOCATE ITSELF WITHIN THE WAREHOUSES' VICINITY THROUGH COMBINING VARIOUS POSITIONING SYSTEM METHODS. THIS ROBOT CAN BE MONITORED OR CONTROLLED THROUGH A WEB BASED INTERFACE.

OBJECTIVE :

- 1) DETECT AND READ POSITION INFORMATION FROM RFID TAGS
- 2) DETECT AND READ RSSI VALUES TRANSMITTED
- 3) LOCATING ITSELF WITHIN THE GRID SYSTEM

METHODOLOGY :

- 1) DEVELOPED USING EMBEDDED C IN ARDUINO IDE
- 2) HTML & PHP FOR WEBSITE DEVELOPMENT
- 3) PROJECT PHASES:

DESIGN-> PROTOTYPE -> EVALUATION -> REFINE

DISCUSSION :

AN INNOVATIVE DESIGN WHICH COMBINES VARIOUS LOCALIZATION TECHNIQUES IN ACHIEVING A SMART IPS FOR ASSET TRACKING IN THE FUTURE, EQUIPPED WITH A MORE ROBUST WIRELESS SYSTEM, AND HAVING A MORE INTRICATE AND SLEEKY DESIGN, IT CAN BE A VERY EFFICIENT YET AESTHETIC WAREHOUSE ROBOT

FINAL SYSTEM :

A ROBUST AND EFFICIENT WAREHOUSE ROBOT PRODUCING ACCURATE LOCALIZATION CALCULATIONS FOR ASSET TRACKING AND MONITORING

TURNITIN RESULT

TURNITIN RESULT

The screenshot shows a Turnitin submission interface. The top navigation bar includes the user's name 'Zheng Yan Wong' and the document title 'FYP 2 Wong Zheng Yan'. The main content area is split into two panels. The left panel displays the document text, with sections for 'ACKNOWLEDGEMENTS' and 'ABSTRACT'. The right panel, titled 'Match Overview', shows a large red '2%' match percentage. Below this, a list of three sources is shown, each with a match percentage of 1%:

Rank	Source	Match Percentage
1	Submitted to Universiti ... Student Paper	1%
2	Hui Liu. "", IEEE Transa... Publication	1%
3	Hua Qin, Yang Peng, W... Publication	1%

TURNITIN RESULT

Turnitin - Google Chrome
turnitin.com/newreport_printview.asp?eq=0&eb=1&esm=-1&oid=1162200517&sid=0&n=0&m=2&svr=58&r=51.508663096855045&lang=...

This is a preview of the print version of your report. Please click "print" to continue or "done" to close this window.
[print](#) [done](#)

Turnitin Originality Report

Processed on: 23-Aug-2019 04:29 +08
ID: 1162200517
Word Count: 10577
Submitted: 2

FYP 2 Wong Zheng Yan By Zheng Yan Wong

1% match (student papers from 10-Apr-2017)
[Submitted to Universiti Tunku Abdul Rahman on 2017-04-10](#)

Similarity Index	Similarity by Source
2%	Internet Sources: 1% Publications: 1% Student Papers: 2%

1% match (publications)
[Hui Liu. "", IEEE Transactions on Systems Man and Cybernetics Part C \(Applications and Reviews\), 11/2007](#)

1% match (publications)
[Hua Qin, Yang Peng, Wensheng Zhang. "Vehicles on RFID: Error-Cognitive Vehicle Localization in GPS-Less Environments", IEEE Transactions on Vehicular Technology, 2017](#)

ACKNOWLEDGEMENTS I cannot express enough thanks and appreciation to my supervisor, Dr Lee Wai Kong for his continuous support and encouragement. I offer my sincere appreciation for the bright opportunity provided for me to engage in an indoor positioning system design. The advice given by Dr Lee has been a great help in realizing this project. My completion of this project could not have been accomplished without the support of my friends and family for standing by my side during hard times. Finally, I would like to express my very great appreciation to Ms CY who has been by my side for the whole time through the project, encouraging and motivating me every time I am drowning in the pool of frustration and on the edge of giving up. ABSTRACT This project is an IPS design project that aims to design a warehouse robot capable of delivering goods and merchandise within a warehouse on its own without the need of human aid. This paper provides the concept, methodology and design of the IPS through implementing an Arduino Mega combining various modules. The reason behind Arduino Mega chosen to be the brain of this project is that it has a more powerful ATmega2560 microcontroller with a larger SRAM, Flash Memory and I/O pins compared to the ATmega328p of the Arduino Uno. This project focus on how to enable the robot to accurately calculate its indoor position through RFID tags deployed across the test field. Combining various method and algorithms is very challenging, as all the calculations have to be tally. Additionally, another challenge of this project is that the information fed back to the Arduino for calculation have to be ensured to be accurate. This problem arises since all the modules including the communication between Arduino and the robot's stm32 are using serial/UART, except for the RFID module which uses SPI. The base of

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	
ID Number(s)	
Programme / Course	
Title of Final Year Project	

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: _____ % Similarity by source Internet Sources: _____ % Publications: _____ % Student Papers: _____ %	
Number of individual sources listed of more than 3% similarity: _____	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Signature of Co-Supervisor

Name: _____

Name: _____

Date: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	
Student Name	
Supervisor Name	

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Cover
	Signed Report Status Declaration Form
	Title Page
	Signed form of the Declaration of Originality
	Acknowledgement
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. <hr style="width: 80%; margin-left: 0;"/> (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. <hr style="width: 80%; margin-left: 0;"/> (Signature of Supervisor) Date:
---	---