I hereby declare that I have read through this report entitled "**MICROCONTROLLER AND ZIGBEE BASED PUBLIC TRANSPORTATION (BUS) AWARENESS SYSTEM**" and found that it has comply the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering (Control, Instrumentation, And Automation).

Signature: ....

Supervisor‟s Name: Assoc. Prof. Mohd Ariff B. Mat Hanafiah

Date : 1st June 2015

MICROCONTROLLER AND ZIGBEE BASED PUBLIC TRANSPORTATION (BUS) AWARENESS SYSTEM

ALI KHALED ALI AL-FARAWI

A report submitted in partial fulfillment of the requirements for the degree of Electrical Engineering (Control, Instrumentation And Automation)

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

JUNE 2015

I declared that this thesis entitled "**MICROCONTROLLER AND ZIGBEE BASED PUBLIC TRANSPORTATION (BUS) AWARENESS SYSTEM**" is the result of my own research except as cited in the reference. The report has not been accepted for any degree and is not concurrently submitted candidature of any other degree.

Signature: ....

Name: ALI KHALED ALI AL-FARAWI

Date : 1st June 20

To my beloved father and mother who always there for me

Khaled Ali Al-farawi *and* ***Amani Abdulqadir Al-mihsini***

To my supervisor and lecturer, for their guidance and encouragement

Prof Mohd Ariff Bin Mat Hanafiah

To my friends, for their unconditional support

Dedicated in thankful appreciation for your supporting, encouragement and best wishes

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my supervisor *PROF MADYA MOHD ARIFF* BIN *MAT* HANAFIAH for his sincere guidance along the project. Secondly, I also would like to thank my panels, lecturers and friends for their contributions into my project by giving comments and advice to improve it.

Lastly, I would also like to thank my parents for their prayers and support along the journey of my study. I wouldn‟t be able to make my way without them.

ABSTRACT

The problems that people face every day at the bus stop can be sometimes very troublesome as it makes people miss their buses or get confused to take which bus. Thus in this paper a system is designed using ZigBee wireless communication to link the buses that arrives at the bus stop with the bus stop. The objective of this product is that it will prevent people from missing their buses and it will avoid people to lose their buses between many other buses. The product will be prototype of two buses and one bus stop, when the bus stop button is pressed the installed LCD on the bus stop will show a message that the bus is ordered and the led and the buzzer on the bus will energized to notify the bus driver that there is passengers on that bus stop. When the bus arrived at the bus stop the message of the arrival of the specific bus will be visualized on the LCD and the installed buzzer on the bus station will energized to notify the passengers, at the same time the buzzer and the led on the bus will turn off. When the bus leave the bus stop, the LCD will show that the bus is leaving and the buzzer on the bus stop will turn off. The methods used in this paper to achieve the objective are to use the ZigBee because it is more reliable and has the least power consumption. The logic signal from the button is then raised to a proper voltage being able to be sent over the wireless communications link. The product will be consisting of two ZIGBEE representing the specific buses and one ZIGBEE at the bus stop. The Arduino is used as a controller.

TABLE OF CONTENT

CONTENTS	PAGE
ACKNOWLEDGEMENT	V
ABSTRACT	VI
TABLE OF CONTENT	VII
LIST OF TABLES	IX
LIST OF FIGURES	X
CHAPTER 1	1
INTRODUCTION	1
1.1 Project background	1
1.2 Problem statement	5
1.3 Objectives	6
1.4 Scope	6
1.5 Report outline	7
CHAPTER 2	8
LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Overview of ZigBee	8
2.2.1 ZigBee Alliance	9
2.2.2 History of ZigBee	10
2.2.3 ZigBee vs. Other wireless standasrds	11
2.2.4 XBee vs XBee Pro	13
2.3 XBee	15
2.4 Arduino Uno	16
2.5 Other components overview	17
2.5.1 LC04 converter	17
2.5.2 16x2 LCD screen	17

2.5.3 Buzzer 19

2.5.4 Reed switch sensor 20

2.5.5 LED 20

2.5.6 Capacitors 21

2.6 Related Works 22

2.7 Summarization of Literature Review 24

CHAPTER 3 25

METHODOLOGY 25

3.1 Introduction 25

3.2 Project Methodology 26

3.2.1 Project Flowchart 27

3.3 Hardware Implementation 28

3.3.1 The First Part: The Bus Stop 28

3.3.2 The Second Part: The Bus 33

3.3.3 Software and programming 36

CHAPTER 4 42

RESULTS & DISCUSSION 42

4.1 Result overview 42

4.2 Project result procedures 47

CHAPTER 5 54

CONCLUSION 54

REFERENCES 55

APPENDICES 57

Appendices A: The Bus Stop Arduino Programming Codes 57

Appendices B: Project flow chart 70

LIST OF TABLE

Table	Title	Page
Table 2.1	Comparison of Wireless and Wired communication	8
Table 2.2	ZigBee vs. Other wireless standards	11
Table 2.3	Comparison of 3 Wireless communication channels	12
Table 2.4	Specifications of the Xbee	14
Table 2.5	LCD screen pin description	18
Table 3.1	First part components	28
Table 3.2	Second Part components	33

LIST OF FIGURES

Figure	Title	Page
Figure 1.1	Layers Standards	3
Figure 2.1	Xbee vs Xbee Pro	13
Figure 2.2	Xbee Module	15
Figure 2.3	Arduino Uno Board	16
Figure 2.4	LCD Screen	18
Figure 2.5	Buzzer	19
Figure 2.6	Smt buzzer basic drive circuit	19
Figure 2.7	Reed Switch Sensor	20
Figure 2.8	LED	20
Figure 2.9	Capacitors	21
Figure 3.1	Project Methodology Flowchart	25
Figure 3.2	Project Overview	26
Figure 3.3	Project Flowchart	27
Figure 3.4	Bus Station Block Diagram	29
Figure 3.5	Xbee S2	30
Figure 3.6	LC04 converter connection	31
Figure 3.7	LCD Pin connection	32
Figure 3.8	The Bus Block Diagram	34
Figure 3.9	The Bus circuit	35
Figure 3.10	The Bus power supply circuit	35
Figure 3.11	Coordinator API Settings	37

Figure 3.12	Router1 AT Settings	38
Figure 3.13	Router1 AT I/O Settings	39
Figure 3.14	Router2 AT Settings	40
Figure 3.15	Router2 AT I/I Settings	41
Figure 4.1	The Bus Stop Prototype	42
Figure 4.2	The Bus Prototype	43
Figure 4.3	Bus1 and Bus2	44
Figure 4.4	The bus stop inside view	45
Figure 4.5	The bus stop prototype top view	45
Figure 4.6	The bus inside view	46
Figure 4.7	The bus top view	46
Figure 4.8	The bus stop starting	47
Figure 4.9	The bus stop starting part2	47
Figure 4.10	The bus stop detecting	48
Figure 4.11	The bus stop detecting 2	48
Figure 4.12	Ordering the first bus	49
Figure 4.13	Notifying the first bus	50
Figure 4.14	The bus ordered	50
Figure 4.15	The first bus arriving	51
Figure 4.16	The first bus leaving	52
Figure 4.17	Notifying the second bus	52
Figure 4.18	The second bus arriving	53
Figure 4.19	The second bus leaving	53

CHAPTER 1

INTRODUCTION

1.1 Project Background:

ZigBee is a wireless networking standard that is aimed at remote control and sensor applications which is proper for operation in harsh radio environments and in isolated locations where there is a little or no ability to access the wired link.

ZigBee technology builds on IEEE standard 802.15.4 which defines the physical and MAC layers. Moreover, ZigBee defines the application and security layer specifications enabling mutual operation between products from different manufacturers. In this way ZigBee is a part of the 802.15.4 specification.[1]

With the applications for remote wireless sensing and control growing faster than ever, it is estimated that the market size could reach hundreds of millions of dollars as early as 2007. This makes ZigBee technology a very attractive alternative for many modern applications that are required to be automated or wirelessly controlled.

The distances that can be accomplished transmitting starting with one station then onto the next reach out up to around 70 meters. All that much more prominent distances may be come to by handing-off information starting with one hub then onto the next in a system, though [2].

The control and monitoring of the sensor and the controllers of 802.15.4 family of ZigBee doesn"t need high data throughput, thus the application can consume little power. The possibility of remote, battery controlled sensors, low power utilization is a key necessity. Sensors, lighting controls, security and numerous more applications are all contender for the new innovation [1].

The devices and the control unit would all need a typical standard to empower clear correspondence. ZigBee is such a standard for inserted application programming and has been approved in late 2004 under IEEE 802.15.4 Wireless Networking Standards [1].

ZigBee is a set up situated of determinations for remote individual region organizing (WPAN), i.e., computerized radio associations in the middle of PCs and related gadgets. This sort of system takes out utilization of physical information transports like USB and Ethernet links. The gadgets could incorporate phones, hand-held advanced collaborators, sensors and controls situated inside of a couple meters of one another [2].

- **Architecture:**

Despite the fact that WPAN infers a scope of just a couple meters, 30 feet on account of ZigBee, the system will have a few layers, so outlined as to empower intrapersonal correspondence inside of the system, association with a system of more elevated amount and at last an uplink to the Web.

The ZigBee Standard has evolved standardized sets of solutions, called „layers'. These layers provide the features that make ZigBee very attractive. Being of low cost, being able to by easily implemented, reliable data transfer, short-range operations. Moreover, having very low power consumption and adequate security features adds to the benefits of ZigBee. In the following paragraphs the layers of the ZigBee is explained.

- **The first layer is the Network and Application Support layer**

This layer can deal with immense quantities of hubs. This level in the ZigBee structural engineering incorporates the ZigBee Device Object (ZDO), client characterized application profile(s) and the application Support (APS) sub-layer. The APS sub-layer's obligations incorporate support of tables that empower coordinating between two devices and correspondence among them. The viewpoint that recognizes different devices that works in the working space of any device.

The second layer is the physical (PHY) layer. The IEEE802.15.4 PHY physical layer accommodates high levels of integration by using direct sequence to permit simplicity in the analog circuitry and enable cheaper implementations. Figure 1 shows the MAC layer.

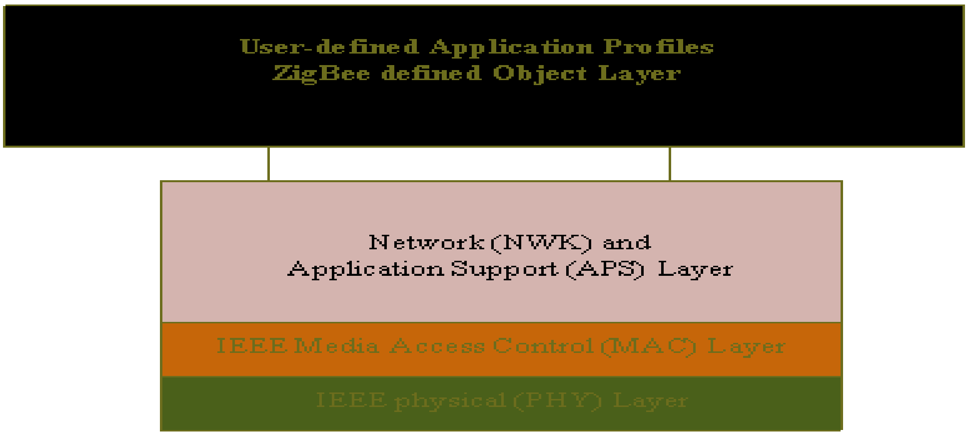

Figure 1.1 "Layers Standards"

The third layer is the MAC (Media Access Control) layer which is determined to work in one of the three licenses free groups at 2.4 GHz, 915 MHz for North America and 868 MHz for Europe. Thusly the standard has the capacity work the world over, in spite of the fact that the precise details for each of the groups are marginally diverse. At 2.4 GHz there are an aggregate of sixteen distinct channels accessible, and the greatest information rate is 250 kbps. For 915 MHz there are ten channels and the standard backings a most extreme information rate of 40 kbps, while at 868 MHz there is stand out channel and this can bolster information exchange at up to 20 kbps [1]

- **Data Transfer:**

The data is transferred in packets that have a maximum size of 128 bytes, allowing for a maximum payload of 104 bytes. Although this may appear low when compared to other systems, the applications in which 802.15.4 and ZigBee are likely to be implemented should not require very extreme data rates[2].

The standard backings 64 bit IEEE addresses and in addition 16 bit short addresses. The 64 bit addresses remarkably recognize each gadget in the same way that gadgets have an interesting IP address. When a system is situated up, the short addresses can be utilized and this empowers more than 65000 hubs to be supported [1].

It also has an optional super frame structure with a method to synchronize the time and give high priority to some massages that need to be processed faster. To achieve this, a guaranteed time slot mechanism has been incorporated into the specification. This enables these high priority messages to be sent across the network as swiftly as possible [2].

As it is discussed in the MSC layer, each band has a different frequency to use and thus, the modulation techniques also change based on the type of band in used. Direct succession spread range (DSSS) is utilized as a part of all cases. Be that as it may, the 868 and 915 MHz groups the genuine type of balance is parallel stage movement keying. For the 2.4 GHz band, counterbalance quadrature stage movement keying (O-QPSK) is utilized.

In perspective of the way that frameworks may work in vigorously congested situations, and in territories where levels of superfluous obstruction is high, the 802.15.4 specification has fused a mixed bag of components to guarantee exceedingly dependable operation. These incorporate a quality evaluation, beneficiary vitality location and clear channel appraisal. CSMA (Carrier Sense Multiple Access) systems are executed to focus when to transmit, and along these lines pointless conflicts are kept away from.

Many years ago, when Bluetooth technology was presented, it was believed that Bluetooth would make WIFI repetitive. Yet, the two coincide well today, so do numerous different Wireless standards like Wireless HART and ISA100.11a. At that point why might we require another WPAN standard like Zigbee? The answer is, the application center of Zigbee Alliance - ease and low power for vitality productive and savvy wise gadgets. In addition, Zigbee and Bluetooth have diverse application center. Notwithstanding of every one of their similitudes, and regardless of the way that both are in view of the IEEE 802.15 measures, the two are diverse in innovation and additionally scope. Bluetooth is made with cellular telephones as its focal point of universe empowering media move at rates in abundance of 1 Mbps while Zigbee is based with accentuation on low information rate control framework sensors highlighting slower information of only 250 kbps.

1.2 Problem Statement:

Nowadays, many people miss their buses at the bus stop due to many interruptions occurred between them and their environment. Moreover, many people get into trouble when trying to catch their buses when there are many buses waiting at the bus stop as well as for the bus drivers may waste their time by going through every bus stop without knowing if there are passengers waiting. Moreover, disabled people also encounter some problems due to their disability.

However this problem can be solved by building microcontroller and ZIGBEE Based Public Transportation Bus Awareness System. In this system, there will be a connection created between the bus and the bus stop so as to ensure the people at the bus stop will be aware of the presence of the buses. In addition, the system will notify the driver about the availability of the passengers at the bus stop, so the driver will avoid going through bus stops free of passengers. Not only that, the system will provide a LCD screen and buzzer to help the normal people and the disabled people to notice the presence of the bus.

1.3 Objectives:

- To make a data communication between the bus and the bus stop.
- To design a system that will aware the people of the presence of the bus.
- To distinguish between the buses and their destinations accordingly.
- To handle the communication that is between the bus and the passengers automatically and effortlessly.

1.4 Scope

The scope of this project is to use ZigBee Radio Frequency data transfer as a means of communication. Also, Arduino microcontroller is been used in this project as the controller of the system. Moreover, there will be an electronic notice board at the bus stop to alert the people of the presence of the bus. This project will be done by using a LCD screen to notify the people. Thus, in this project only the presence of the bus at the bus stop will be notified to the people at the bus stop and the making of bus schedule that needs further components like GPS will not be covered.

1.5 Report Outline

There are five chapters in this thesis which are introduction, literature review, methodology, result and discussion and finally conclusion and recommendation. Each chapter will discuss its own aspects related to the project.

Chapter one is the introduction for the project. Problem statement, object and scope of the project along with the summary of works have been discussed in this chapter. Then, chapter two discusses more on the theory and literature. Besides that, this chapter also discusses the type of Arduino used for the project, the sensor chosen, and the software involve in programming the Xbee and Arduino.

Chapter three focuses on the methodology and approaches on the project. This includes the programming of the software and hardware development of the project. Results and discussion are presented in chapter four. Lastly, chapter five is the conclusion for the whole project. Some future suggestions such as a functional addition and hardware improvement of the project are also mentioned.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction:

This chapter goes through the past related works that have been done on the protocol communication between the bus and passengers for the sake of implements and design a solution for enhancing the public transportation administration services based on wireless connection such as ZigBee, Gprs, Bluetooth, Gsm and Wi-Fi. Besides that this section discusses in details on the components and instruments used for this project.

Wireless Communication	Wired Communication
No need for physical communication	Need physical communication medium which is cable
No problem in extending the detection range of the signal	Need to change cables length if need to change connection distance
Reduce cost from buying cables	Need extra cost for cables purchasing
Not affected by environment condition	Get affected by environment condition
Less affected by noise produce from surrounding machinery	Easily affected by noise produce from surrounding machinery
Changeable data transfer rate	Data transfer rate is according to the material of the wires
Possible for mobile applications	Not suitable for mobile application

Table 2.1 "Comparison of Wireless and Wired communication"

2.2 Overview of ZigBee:

ZigBee is a new wireless technology. Moreover, it is technological standard created for control and sensor networks based on the IEE 802.15.4 standard, ZigBee created from ZigBee alliance. Plus, ZigBee alliance includes many leading companies such as Philips, Motorola, Intel, HP they are all members of the alliance. Zigbee is a specification for (WPANs) operating at 868 MHz, 915 MHz and 2.4 GHz. A WPAN means (wireless personal area network) and it‟s a network for interrelationship or interconnecting an individual's devices. Using XBee devices in WPAN can communicate at speeds of up to 250 Kbps while physically separated by distances up to 100 meters in typical circumstances and greater distances in an ideal environment. XBee is based on the 802.15.4 specification and the Institute of Electrical and Electronics Engineering Standards Association (IEEE-SA) approved it [3].

ZigBee supplies for high information throughput in applications where the obligation cycle is low. This makes ZigBee perfect for home, business and modern mechanization where control gadgets and sensors are ordinarily utilized. Such gadgets work at low power levels, and this in conjunction with their low obligation cycle (ordinarily 0.1% or less), interprets into long battery life. Application appropriate to ZigBee incorporate warming, ventilation and ventilating (HVAC), lighting frameworks, fire detecting and the discovery, interruption identification and warning of strange events. ZigBee is good with most topologies including shared, star system and cross section systems.

2.2.1 ZigBee Alliance

The ZigBee Alliance is an assembly of organizations cooperating to characterize an open universal standard for making low-power wireless networks. The target of ZigBee Alliance is to make a detail characterizing how to manufacture diverse systems topologies with information security lineaments and interoperable application profiles. The enforcement incorporates organizations from a wide spectrum of classes, from chip manufactures to system integration companies.

The primary particular was confirmed in 2004 and the original of ZigBee products had come to the business in 2005. A major test for the partnership is to make the interoperability to work among different products. To take care of this issue, the ZigBee Alliance has characterized diverse profiles, depending on what kind of classification the product fits in with.

2.2.2 History of ZigBee

- ZigBee-style systems started to be conceived around 1998, when numerous installers understood the both Wi-Fi and Bluetooth would be unsatisfactory for some applications. Specifically, many architects saw a requirement for self-sorting out specially appointed advanced radio systems.
- The IEEE 802.15.4 standard was completed and finished in May 2003.
- In the mid-year of 2003, Philips Semiconductors, a noteworthy cross section system supporter, stopped the speculation. Philips Lighting has, then again, proceeded with Philips cooperation and Philips remains a promoter part on ZigBee Alliance Board of Directors.
- The ZigBee Alliance reported in October 2004 that the participation had dramatically multiplied in the former year and had developed to more the 100 part organizations in 22 countries. By April 2005 enrollment had developed to more than 159 organizations and by December 2005 participation had passed 200 organizations.
- The ZigBee specifications were confirmed on 14 December 2004.
- The ZigBee Alliance declares public availability of Specification 1.0 on 13 June 2005, known as ZigBee 2004 specification.
- The ZigBee Alliance reports the finishing and prompt part accessibility of the upgraded adaptation of the ZigBee Standard in September 2006, known as ZigBee 2006 Specification.
- Amid the last quarter of 2007, ZigBee PRO the upgraded ZigBee particular was concluded.

2.2.3 XBee vs. Other Wireless Standards

Table 2.1outlines some of the key characteristic of ZigBee and how it stacks up against other common wireless standards.

WIRELESS CONNECTIVITY TECHNIQUES			
	Bluetooth	ZigBee	Wi-Fi 802.11
Data rate	1 Mbit/s	20, 40, and 250 kbits/s	11 and 54 Mbits/s
Range	10 m	10 to 100 m	Up to 100 m
Networking topology	Ad-hoc, small networks	Ad-hoc,peertopeer, star, or mesh	Point to hub
Frequency	2.4 GHz	868 MHz (Europe), 900 to 928 MHz (North America), 2.4GHz(worldwide)	2.4 and 5 GHz
Power consumption	Low	Very low	High
Typical applications	Inter-devicewirelesscon-nectivity, e.g., phones, PDAs,laptops,headsets, cameras, printers, serial cable replacements	Industrial control and monitoring, sensor networks, buildingautomation, toys, games	Wireless local-area network (WLAN) connectivity, broadbandInternet, security cameras

Table 2.2: "ZigBee vs. Other wireless standards"

ZigBee looks rather like Bluetooth however straightforward, has a lower information rate and invests the vast majority of its energy resting. This characteristic implies that a hub on a ZigBee system ought to have the capacity to keep running for six months to three years on only two AA batteries.

The operational range of ZigBee is 10-100 compared to 10m for Bluetooth (without a power amplifier). ZigBee sits Bluetooth in terms of data rate, the data rate of ZigBee is 250 kbps at 2.4 GHz, 40 kbps at 915 MHz and 20 kbps at 868 MHz whereas that of Bluetooth is 1 Mbps.

	WiFi/Android	Bluetooth 2.0+	XBEE 802.15.4
Range	Typcial 46m from Hotspot indoors	class 1: up to 100 m (30 typical), class 2: up to 30 m (10 typical)	30 m typical
Expense	$100 xbee wifi kit, assuming network is in-place	$25-100, not dependent on external network	$40-80 typical
Ease of Integration	Relatively simple plug-and-play	Requires adapting/hacking existing product	Simple plug and play
Data Rate	72 Mbps typical for 802.11n*	1 Mbps possible, less typical	250 kbit/s, 100 typical
Operating Frequency	2.4-5 GHz	2.4–2483.5 MHz	902-928 MHz (radio)
Power Consumption	up to 300mW dependent on data rate	up to 50 mW dependent on data rate	peak 150mW
Interface	drop-in board + browser or phone app	drop-in board + bluetooth controller (eg PS3 controller)	drop-in board + analog or digital controller, phone possible
Support	Extensive home networking support, limited RC applications.	Limited RC application due to low range. Limited third party support.	Large amount of support for RC application, from manufacturer and third party users.
Summary	Dependent on in-place router/wifi network. Susceptible to drop-out. Handshaking errors common. Fast transfer speeds, but excessive for hexacopter application.	More than adequate transmission speed, yet low transmission range. RC application not common, resulting in poor product support and few integration options.	Adequate transfer speed for RC application, acceptable transmission range. Low cost and easy integration with copious support.
	*Based on network configuration/hardware		

Table2.3 "Comparison of 3 Wireless communication channels"

Bluetooth's protocol is more complex since it is geared towards handling voice, images and file transfers in ad-hoc networks. Bluetooth devices can support scatter nets of multiple smaller non-synchronized networks (piconets) and it only allows up to 7 slave nodes in a basic master-slave piconet set-up.

When ZigBee node is powered down, it can wake up and get a packet in around 15 mile second whereas a Bluetooth device would around 3 second to wake up and respond.

2.2.4 XBee Vs XBee Pro

The XBee and XBee-Pro RF Modules were built to meet IEEE 802.15.4 models and backing the extraordinary needs of ease, low-control remote sensor systems. The modules oblige negligible power and give dependable conveyance of information between gadgets. The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin perfect with one another.

Figure 2.1 "Xbee vs Xbee Pro"

Specification	XBee	XBee-PRO (S2)	XBee-PRO (S2B)
Performance			
Indoor/Urban Range	up to 133 ft. (40 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant
Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled	50mW (+17 dBm) 10mW (+10 dBm) for International variant	63mW (+18 dBm) 10mW (+10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps	250,000 bps
Data Throughput	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)
Serial Interface Data Rate (software selectable)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)
Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled	-102 dBm	-102 dBm
Power Requirements			
Supply Voltage	2.1 - 3.6 V	3.0 - 3.4 V	2.7 - 3.6 V
Operating Current (Transmit, max output power)	40mA (@ 3.3 V, boost mode enabled) 35mA (@ 3.3 V, boost mode disabled)	295mA (@3.3 V) 170mA (@3.3 V) international variant	205mA, up to 220 mA with programmable variant (@3.3 V) 117mA, up to 132 mA with programmable variant (@3.3 V), International variant
Operating Current (Receive))	40mA (@ 3.3 V, boost mode enabled) 38mA (@ 3.3 V, boost mode disabled)	45 mA (@3.3 V)	47 mA, up to 62 mA with programmable variant (@3.3 V)
Idle Current (Receiver off)	15mA	15mA	15mA
Power-down Current	< 1 uA @ 25°C	3.5 µA typical @ 25°C	3.5 µA typical @ 25°C
General			
Operating Frequency Band	ISM 2.4 GHz	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960 x 1.297 (2.438cm x 3.294cm)	0.960 x 1.297 (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA, or U.FL Connector	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA or U.FL Connector	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA or U.FL Connector
Networking & Security			
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Number of Channels	16 Direct Sequence Channels	14 Direct Sequence Channels	15 Direct Sequence Channels
Channels	11 to 26	11 to 24	11 to 25
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
Agency Approvals			
United States (FCC Part 15.247)	FCC ID: OUR-XBEE2	FCC ID: MCQ-XBEEPRO2	FCC ID: MCQ-PROS2B
Industry Canada (IC)	IC: 4214A-XBEE2	IC: 1846A-XBEEPRO2	IC: 1846A-PROS2B
Europe (CE)	ETSI	ETSI (International variant)	ETSI (10 mW max)

Table 2.4 "Specifications of the Xbee"

2.3 XBee

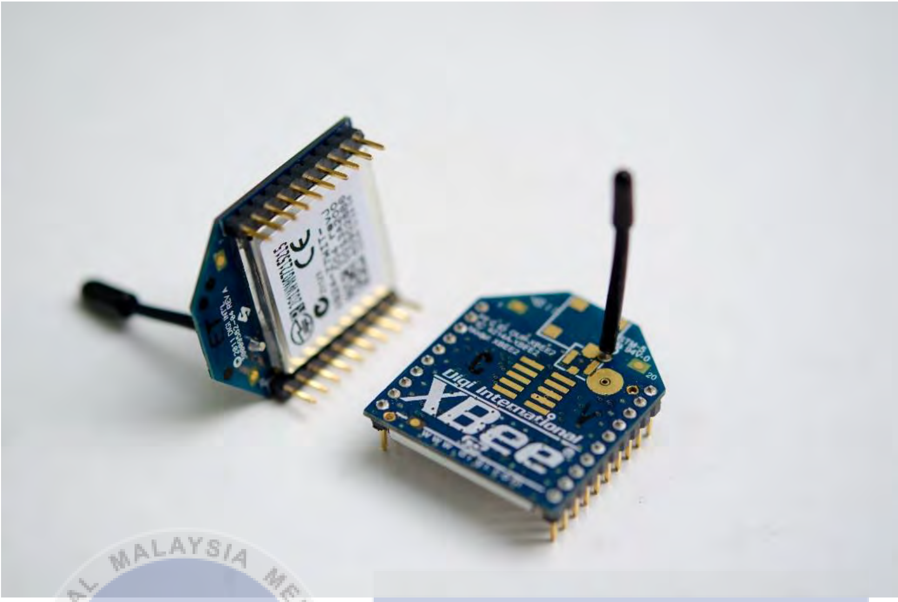

Figure 2.2 "XBee Module"

ZigBee wireless protocol as shown in figure 2.2 provides to network a set of autonomous devices with standard radio frequency transceiver to perform some networked task. In the proposed system a vehicular RF takes the role of XBee end devices while as reader and writer module takes the role of XBee coordinates. Digi XBee 802.15.4 V as shown in figure 2.2 are the easiest-to-use, most reliable and cost-effective RF devices.

2.4 Arduino Uno

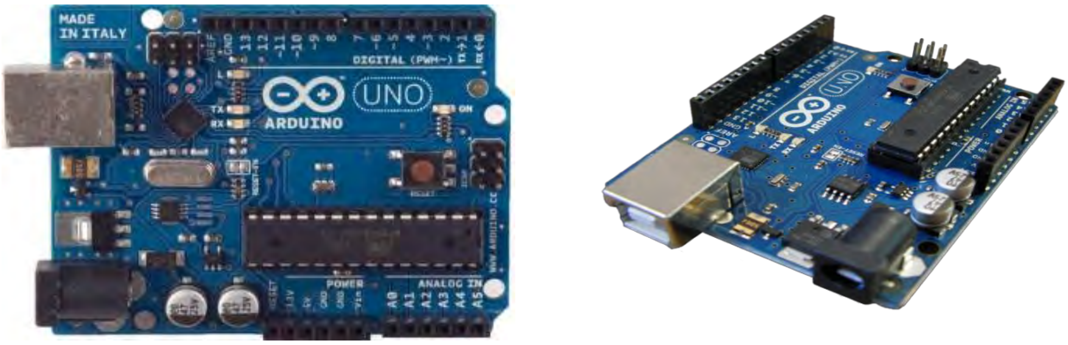

Figure 2.3 "Arduino Uno Board"

The Arduino Uno as indicated in Figure 2.3 is a microcontroller board in light of the ATMEL microcontroller ATmega328. It has 14 advanced info or yield pins (of which 6 can be utilized as PWM yields), 6 simple inputs, a 16 MHz precious stone oscillator, a USB association, a force jack, an ICSP header, and a reset catch. It contains everything expected to backing the microcontroller; basically join it to a PC with a USB link or force it with an AC-to-DC connector or battery to begin [10]. The force source is chosen naturally, since the Arduino Uno can be controlled by the USB association or with an outside force supply, the force source is chosen consequently.

Arduino is an open-source hardware prototyping stage taking into account adaptable, simple to-utilize equipment and programming. Arduino can sense the earth by getting data from a mixed bag of sensors and can influence its surroundings by controlling lights, engines, and different actuators. The microcontroller on the board is modified utilizing the Arduino programming dialect (in light of Wiring) and the Arduino advancement environment (in view of Processing). Arduino activities can be remain solitary or they can correspond with programming running on a PC.

2.5 Other Components overview

In this section there are some components that been used in this project in order to achieve the objective of this project. Also, to design the convenient and suitable circuits that‟s needed in this project.

2.5.1 LC04 Converter

Some voltage converters been installed with the system in this project. Voltage converter commonly used in many devices and many projects, since there are many devices uses different power supply. In order to provide the suitable voltage for the Arduino microcontroller which needs 5v and ZIGBEE which needs 3.3v, LC04 Converter is used in this project. LC04 converter converts the voltage between 3.3v and 5v and serves the both devices.

2.5.2 16x2 LCD screen

LCD (Liquid Crystal Display) screen is an electronic showcase module and locate an extensive variety of utilizations. A 16x2 LCD screen is extremely fundamental module and this sort normally utilized as a part of circuits and gadgets. This screen could show 16 characters for each line while there are two such lines. In this LCD every character is shown in 5x7 pixel framework. This LCD has two registers, to be specific, Command and Data.

The order register stores the charge directions given to the LCD. A summon is a guideline given to LCD to do a predefined undertaking like instating it, clearing its screen, setting the cursor position, controlling showcase and so forth. The information register stores the information to be shown on the LCD.[7]

- **Pin Diagram:**

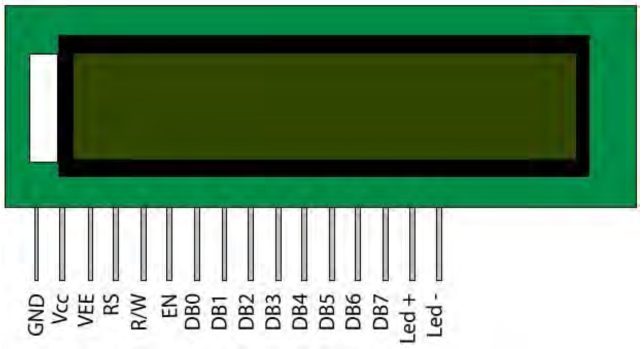

Figure 2.4 "LCD screen"

Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7		DB0
8		DB1
9		DB2
10	8-bit data pins	DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Table 2.5 "LCD screen pin description"

2.5.3 Buzzer

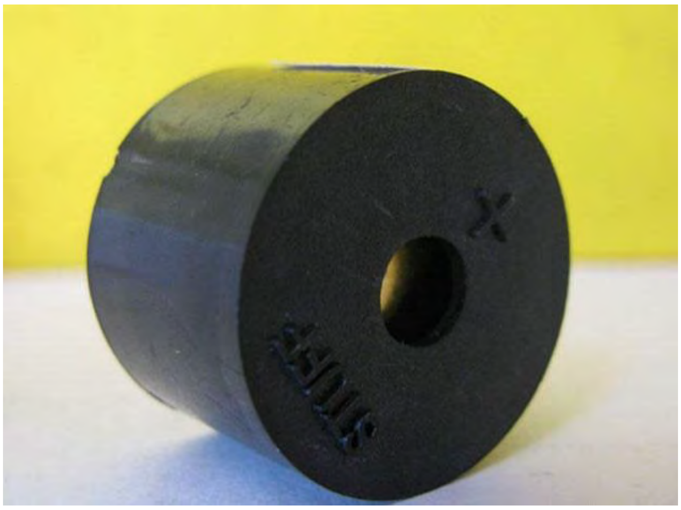

Figure 2.5 "Buzzer"

The buzzer is used in this project and the buzzer could be mechanical, electromechanical, or piezoelectric. In this project the piezoelectric buzzer is used. Piezo buzzer is an electronic gadget usually used to deliver sound. Light weight, straightforward development and low value make it usable in different applications like auto/truck switching marker, PCs, call ringers etc. [8]

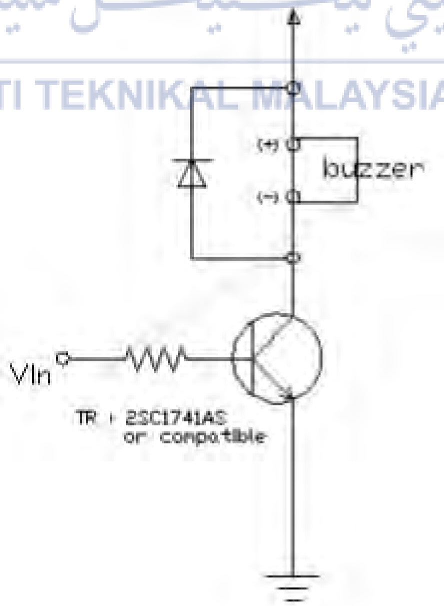

Figure 2.6 "Smt buzzer basic drive circuit"

2.5.4 Reed Switch Sensor

Figure 2.7 "Reed switch sensor"

A Reed sensor is a helpfully bundled reed switch that goes about as a nearness sensor when in the vicinity of an attractive field. No physical contact is needed. In this project precisely, this type of sensors is needed to detect the bus and sense it when it arrive at the bus stop. Moreover, one of the benefits of reed sensor that it does not require any physical contact between the bus and the bus stop which will make the process smooth and effortless.

2.5.5 LED

Figure 2.8 "LED"

A light-emitting diode (LED) is a semiconductor gadget that emanates noticeable light when an electric current goes through it. The light is not especially splendid, but rather in many LEDs it is monochromatic, happening at a solitary wavelength [11]. In this project LED is used to notify the bus driver when the bus ordered from the bus stop.

2.5.6 Capacitor

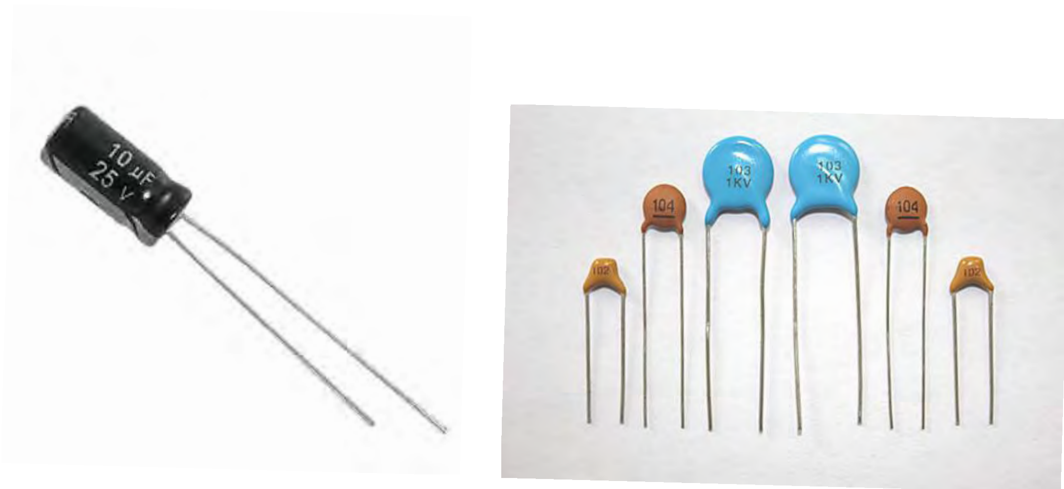

Figure 2.9 "Capacitors"

Capacitor is a uninvolved component that stores electric charge factually and briefly as a static electric field. It is made out of two parallel directing plates isolated by non-leading locale that is called dielectric, for example, vacuum, ceramic, air, aluminum, etc. [13]

The capacitance formula of the capacitor is represented by,

$$C = \frac{\epsilon A}{d}$$

In this project a 0.1uF ceramic capacitor and a 10uf electrolytic capacitor are used. The 0.1uF capacitor is very commonly used in various circuits. It used on many kinds of applications and circuits to decouple ICs from the power supplies. [12] As the 10uf electrolytic capacitor also common in this field, these capacitors are extraordinary transient/surge silencers and function admirably as charge pump tops for MAX232 circuits.

2.6 Related works:

In the previous works the ZigBee have been chosen to be acting as the transmitter and receiver for the bus alert systems. In some of them the signal is received by the Arduino or other microcontroller to drive a LCD or some notice word and etc.

For example, in one research paper by Lavanya[4], the project was for easy navigation for blind people. The idea have been presented through a transport framework utilizing remote sensor systems (WSNs).The visually impaired individuals in the transport station is furnished with a ZigBee unit which is perceived by the ZigBee in the transport and the evidence is made in the transport that the visually impaired individuals is available in the station. So the transport stops at the specific station. The fancied transport that the visually impaired need to take is told to him with the assistance of discourse acknowledgment framework HM2007. The visually impaired gives the information about the spot he needs to achieve utilizing receivers and the voice acknowledgment framework remembers it. The data is then examined by the microcontroller which creates the transport numbers comparing to the area gave by the visually impaired. These transport numbers are changed over into sound yield utilizing the voice synthesizer APR 9600. The ZigBee handset in the transport sends the transport number to the handset with the visually impaired and the transport number is declared to the visually impaired through the earphones. The visually impaired takes the right transport stopped before him and when the destination is come to it is reported by method for the GPS-634R which is associated with the controller and voice synthesizer which creates the sound yield. This task is additionally gone for helping the senior individuals for free route.

In another research by Feng[5], the research explain that the public traffic system mainly depends on driver‟s manual operation, which will inevitably encounter many problems such as punctuality of the bus‟s arrival on bus station. The research proposes a supervisory system based on GPRS and ZigBee technology, to improve the operation efficiency of bus monitoring system and realize intelligent transportation system. The research introduces the bus monitoring system from the aspect of both hardware design and software design. System takes it into accounts for the respective advantages and disadvantages of GPRS and ZigBee, and designs a feasible solution successfully, of practically significant. In conclusion the researcher integrates GPRS with ZigBee to make a supervisory system to alert the people of the bus arrival. The methods he use to achieve its objective is to use a ZigBee based devices with a flash memory to store the arrival time of the bus.

Moreover, in the previous works the ZigBee have been chosen to be acting as the transmitter and receiver for the bus alert systems. In some of them the signal is received by the Arduino or other microcontroller to drive a LCD or some notice word and etc.

As an example, in one research paper by Lavanya[4], a method have been developed to help blind people in the bus stop using the ZigBee and voice recognition kit. The methods used to achieve the objective are to use ZigBee as a means of signal transferring and Arduino to receive the signal and display it on LCD.

In another research by Feng[5], the researcher integrates GPRS with ZigBee to make a supervisory system to alert the people of the bus arrival. The methods he use to achieve its objective is to use a ZigBee based devices with a flash memory to store the arrival time of the bus.

2.7 Summarization of Literature Review

This chapter presented an overview of ZigBee, its applications, its characteristics, and some related works been done on it. Moreover, the components overview that are used in this project is discussed. The following are the key concepts that were described in the chapter:

- ZigBee, a wireless sensor and control network, was developed by the ZigBee Alliance.
- ZigBee applications can be used in home automation, commercial building automation, personal home health care, smart energy, and industrial process monitoring.
- The first ZigBee specification was published in 2004 and supported home control lighting; the ZigBee Alliance no longer supports the 2004 specification.
- In 2006, the ZigBee Alliance published the ZigBee 2006 specification, which was a modification of ZigBee 2004 specification.
- In 2007, ZigBee published ZigBee and ZigBee PRO feature sets.
- ZigBee defines three main types of devices: the coordinator, router, and end device. In addition, devices can act as a trust center or gateway.
- The trust center performs authentication of devices joining the network, security management, and key distribution.
- ZigBee offers star, tree, and mesh topologies.
- It uses layer architecture for its protocol.
- It uses IEEE 802.15.4 for its physical and MAC layers.
- ZigBee protocol architecture consists of the application, application support sublayer, and network layers
- It offers direct, group, and broadcast addressing.
- A ZigBee end device can have 240 endpoints, where each endpoint may represent different application.
- The ZigBee 2007 specification supports tree and mesh topologies.
- The descriptions of the components.
- LC04 converter.
- 16x2 LCD screen.
- Buzzer and Capacitors and reed switch sensor.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This project was planned and designed to build and implement a hardware using ZIGBEE and Arduino microcontroller which will be used for three prototypes, the first one is the bus stop and the rest two is bus1 and bus2.

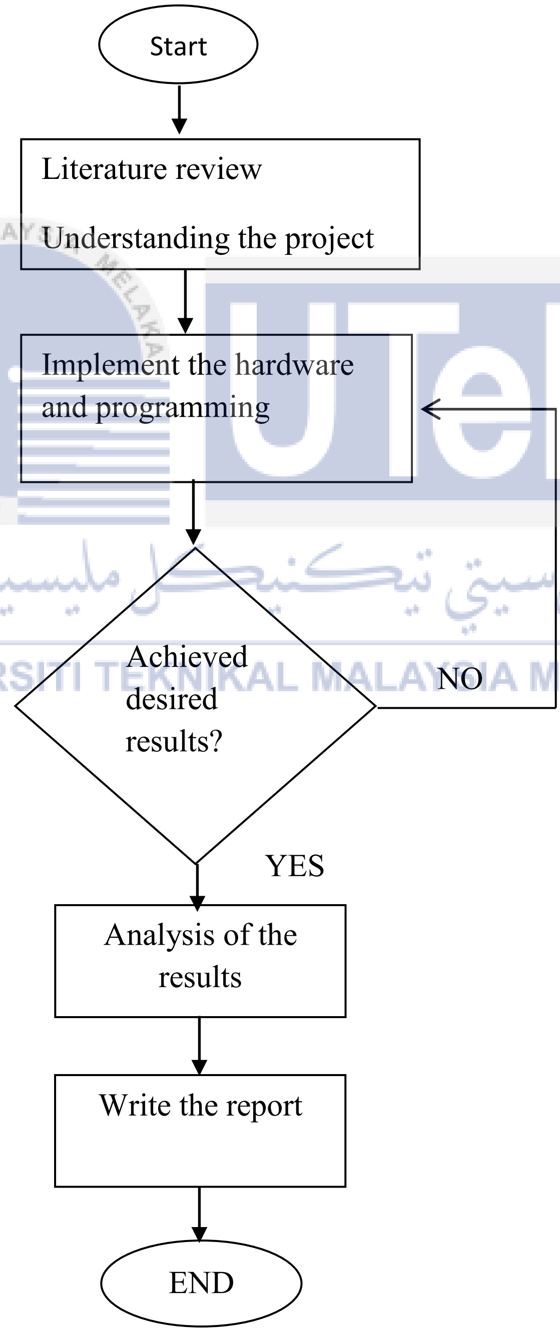

Figure 0.1 "Project Methodology Flowchart"

3.2 Project Methodology

The project was partitioned into two parts:

i. The first part consists of the hardware that will be used as the bus stop and it contains the Arduino microcontroller, ZIGBEE, LCD screen, buzzer and other elements. The Arduino microcontroller act as the heart of this system and the ZIGBEE of this module will act as the coordinator. For the sake of create the connection and to notify the passengers on the bus stop.

ii. The second part consists of the hardware that will be used as the bus and it consist of two buses and it contains the ZIGBEE, led, sensor, buzzer and other elements. The buses will receive the signal from the bus stop through the ZIGBEE which act as a router or end device.

Figure 3.2 "Project Overview"

3.2.1 Project Flowchart:

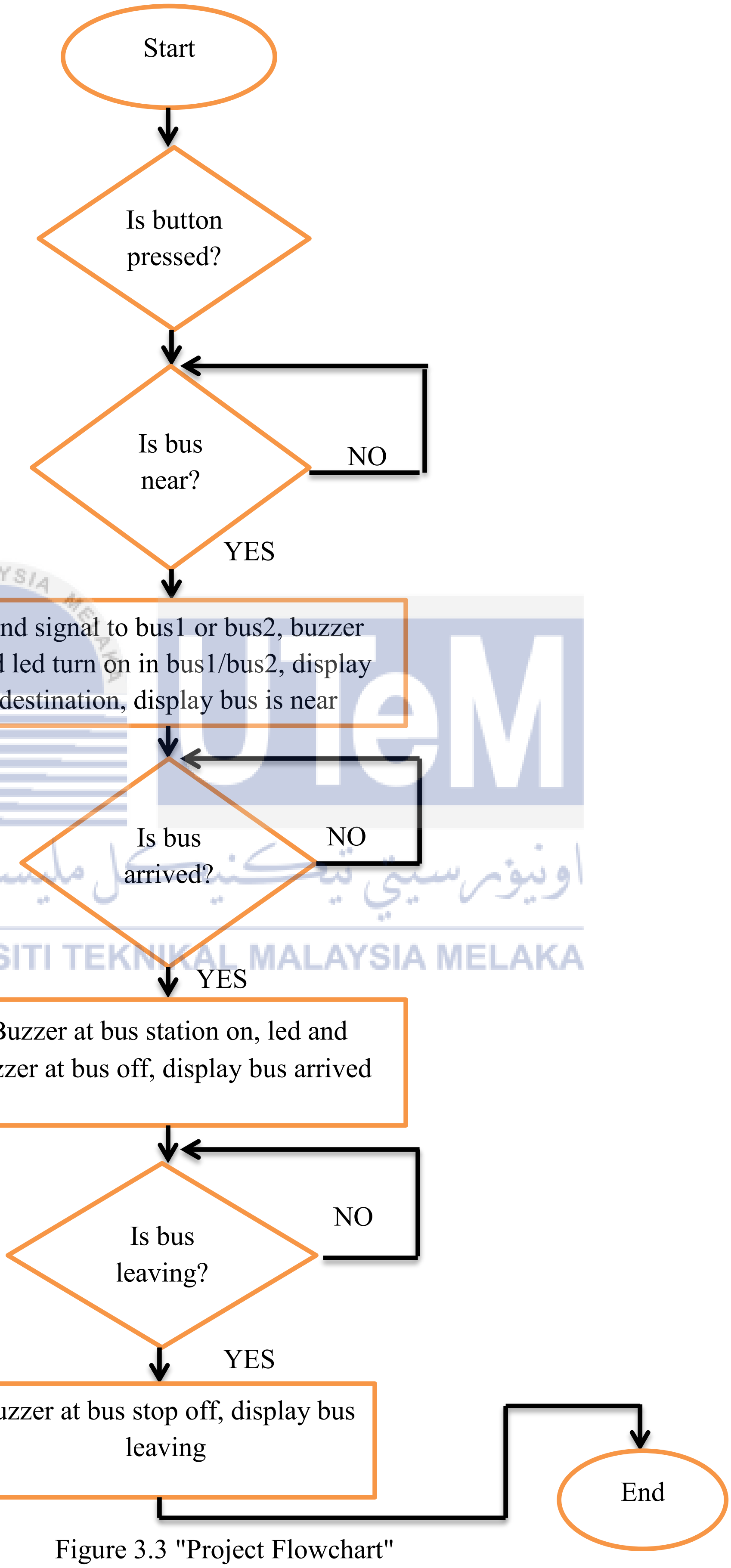

Figure 3.3 "Project Flowchart"

3.3 Hardware Implementation

In this project, there are two parts of hardware involved**.** The first one is the bus stop prototype and the second one is the buses prototype. This section will discuss on the components that been used in this project and will discuss on the design and function of the components that are connected to the Arduino microcontroller in order to build the project. Moreover, this section analyzes and explains the connections between the components and the Arduino microcontroller.

3.3.1 The First Part: The Bus Stop

For this part of the project, the bus stop hardware was implemented through collecting the needed components, designing the suitable circuit for the components and writing the suitable codes for the programming.

- **List Of Components:**

Arduino Uno	1 Unit
Xbee series 2	1 Unit
LC04 converter	1 Unit
LCD graphic screen 16x2	1 Unit
Buzzer	1 Unit
PC or Laptop	1 Unit
Reed switch sensor	1 Unit
Strip board	1 Unit
Wires	-
Button	2 Unit
Potentiometer	1 Unit

Table 3.1 "First part components"

- **Block diagram**

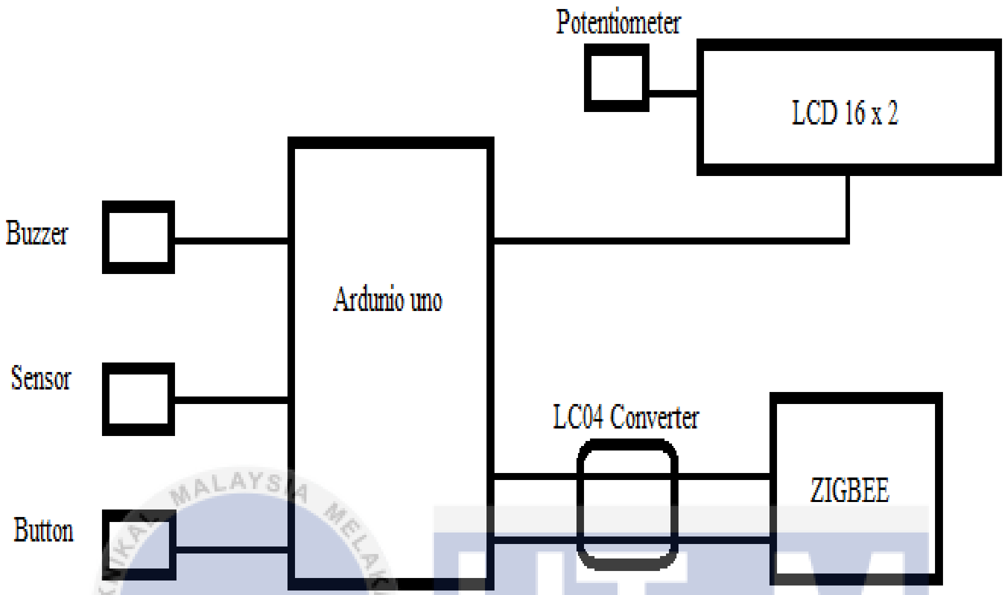

Figure 3.4 "Bus Stop Block Diagram"

The block diagram in figure 3.4 shows the design process of the Bus stop where the whole components connected to the Arduino microcontroller which act as the heart of the system. Also, it shows how the LC04 converter connected between the ZIGBEE and the Arduino uno since the Arduino needs 5 Volt and the ZIGBEE needs 3.3 Volt. Moreover, it shows the potentiometer that connected to the LCD for the contrast, lastly the LCD is displaying wither the bus1/bus2 arrived or leaving to or from the bus stop.

- **Discussion**

The required parts were chosen and obtained and were arranged to use. Then the connection part between the components set up afterwards with the Arduino uno. After that, it was the programming part and that‟s includes the settings of the XBee in each prototype. The troubleshooting and the testing part were after that. The progressions of the components were as:

a) XBee

Figure 3.5 "Xbee S2"

The ZIGBEE that been used for the implementation was Xbee s2. The features of it as below:

- 3.3V @ 40mA
- 250kbps Max data rate
- 2mW output (+3dBm)
- 400ft (120m) range
- Built-in antenna
- Fully FCC certified
- 6 10-bit ADC input pins
- 8 digital IO pins
- 128-bit encryption
- Local or over-air configuration
- AT or API command set

The XBee were connected to the Arduino microcontroller through the LC04 converter, since the voltage supply for the XBee is 3.3v and the voltage supply for the Arduino is 5v. For the connections as shown below:

Arduino PIN A0 → Dout in the ZIGBEE

Arduino PIN A1 → Din in the ZIGBEE

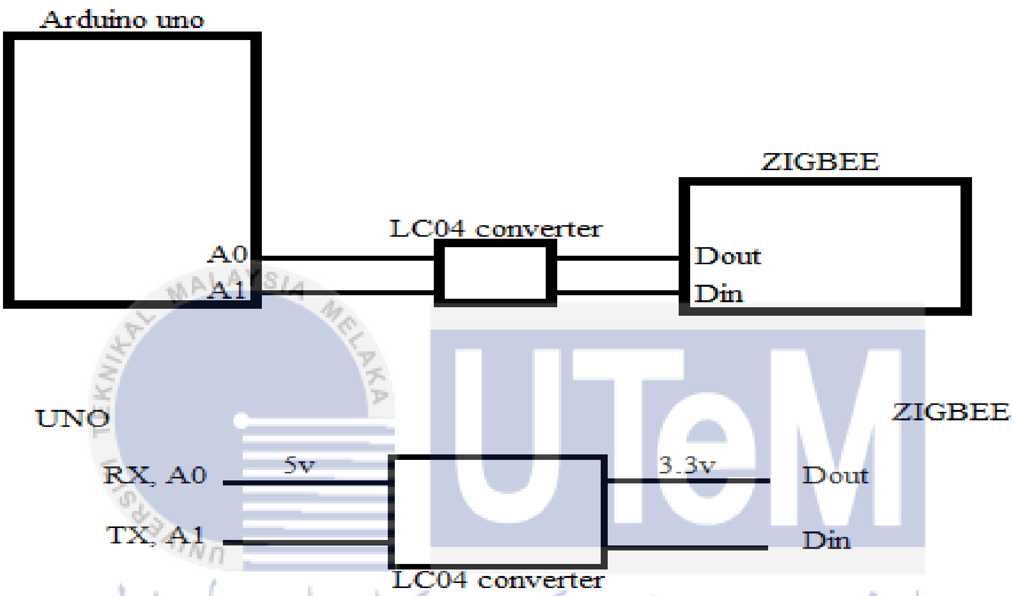

Figure 3.6 " LC04 converter connection"

b) LCD Graphic Screen

The LCD that was used in this experiment was LCD graphic green background with 16 x 2. The feature of it as below:

- Type: Character
- Display format: 16 x 2 characters
- Built-in controller: ST 7066 (or equivalent)
- Duty cycle: 1/16
- 5 x 8 dots includes cursor
- + 5 V power supply
- Optional: Smaller character size (2.95 mm x 4.35 mm)

And for the connections as shown below:

Arduino PIN 4 → D4
Arduino PIN 5 → D5
Arduino PIN 6 → D6
Arduino PIN 7 → D7
Arduino PIN 3 → Enable
Arduino PIN 2 → RS
Arduino PIN 5V → VDD

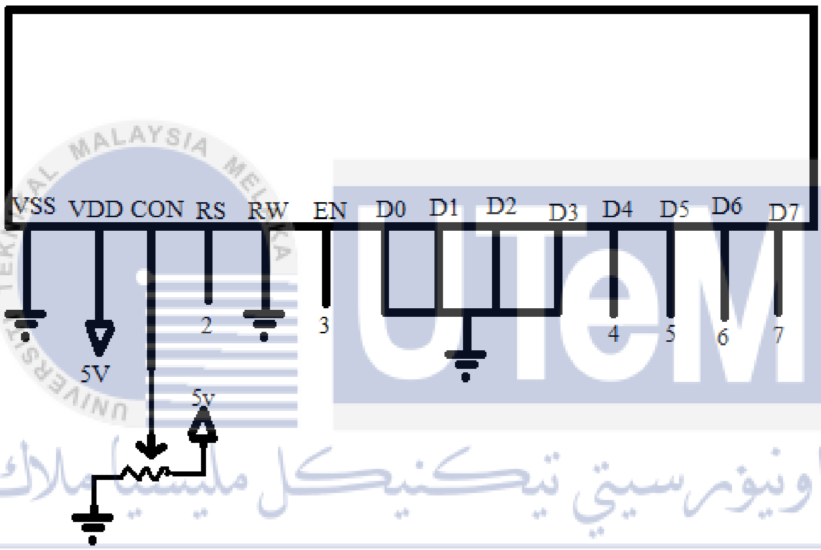

Figure 3.7 "LCD Pin connection"

c) Buzzer and Button

The buzzer and the button were connected to the Arduino directly through:

Arduino Pin A3 → Reset Button
Arduino Pin A4 → Buzzer
Arduino Pin A5 → Button

3.3.2 The Second Part: The Bus

For the second part, it was much easier than the first one. In this prototype the XBee acting as the router or end device. Other components were connected to the XBee since it is the heart of this prototype and to ensure the device is function as it supposed to be. Hence, in this project there are two buses been used and implemented, the first bus (Melaka MITC) and the second bus (Bukit Beruang). Moreover, the list of components as listed below:

- **List of Components**

Description	**Quantity**
Xbee series 2	2 Units
LED	2 Units
Strip Board	2 Units
PC or Laptop	1 Unit
Battery 9v	2 Units
330Ω resistor	4 Units
Buzzer	2 Units
LD33v Voltage Regulator	2 Units
1n4001 diode	2 Units
10KΩ resistor	2 Units
Reed switch sensor	2 Units
2N2222 transistor	2 Units
0.1μF Ceramic Capacitor	2 Units
10μF Electrolytic Capacitor	2 Units
Button	2 Units

Table 3.2 "Second Part components"

- **Block Diagram**

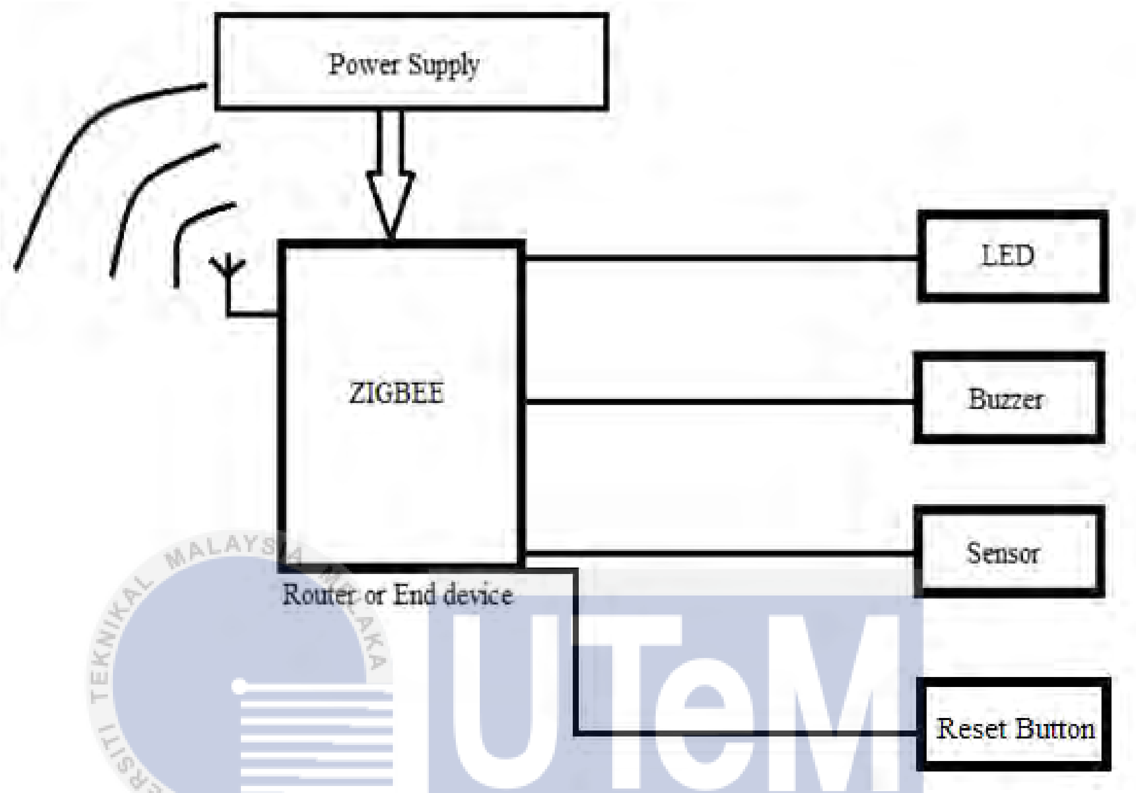

Figure 3.8 "The Bus Block Diagram"

The block diagram shows the second part of hardware where the ZIGBEE is the heart of the bus system. The led and buzzer connected in the bus to aware the bus of the presence of the passengers on the bus stop. The Reset button been added to both buses in this project. The reset button designed for the drivers of the buses, when this button pressed the led and buzzer on the bus will turn off until the bus arrive at the bus stop to avoid the noise and to avoid any repeated notification.

- **Discussion**

In this part of the project, the build of this prototype was not complicated due to its limited procedure. The ZIGBEE is the heart of this prototype and the other components such as led, buzzer and sensor were connected all to the ZIGBEE. Moreover, the ZIGBEE in the buses will act as the router or end device and will receive the signal from the ZIGBEE on the bus stop which act as the coordinator.

The circuit of the bus was as shown below:

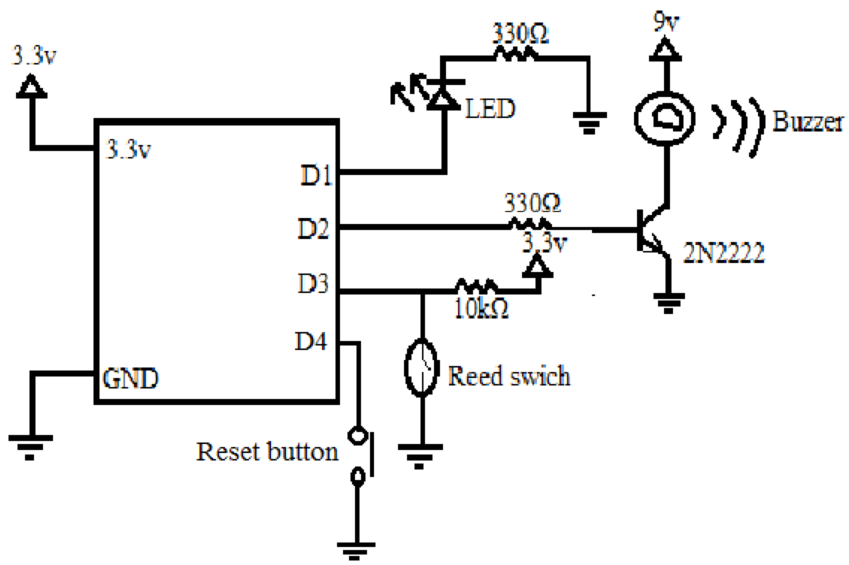

Figure 3.9 "The Bus Circuit"

The figure 3.9 shows the connections between the ZIGBEE and the other components as:

ZIGBEE Pin D1 ⟶ LED

ZIGBEE Pin D2 ⟶ Buzzer

ZIGBEE Pin D3 ⟶ Reed switch sensor

ZIGBEE Pin D4 ⟶ Reset button

And since the 9v battery been applied in this prototype, the voltage regulator LD33v was needed to supply the ZIGBEE which use 3.3v. The circuit below shows the connections:

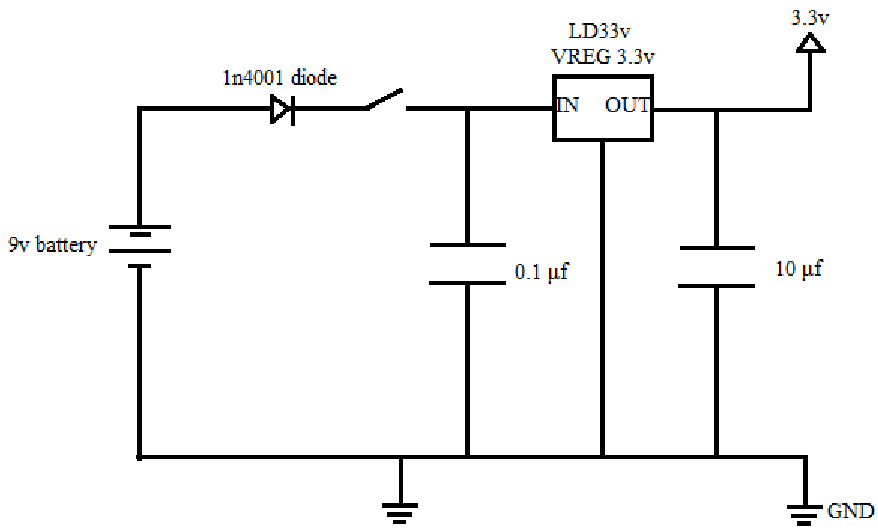

Figure 3.10 "The Bus power supply Circuit"

3.3.3 Software and Programming

So to accomplish the goals and the objectives of this project, Arduino microcontroller must be programmed through Arduino program, by writing the suitable codes using C++ language. All the codes for the Arduino programming are provided in APPENDIX A.

Then, the next step is to make the connection between the three Xbee devices on the bus1, bus2 and the bus stop. In order to create the connections between the Xbee devices wirelessly, the three Xbee must be set up their settings through the programming. The Xbee on the bus will act as the coordinator and the other two Xbee will act as the router or end device.

X-CTU software is the used software to program the Xbee in this project. Once the installation of X-CTU software is complete in the computer, each COM for each Xbee must be tested to ensure it connect. Moreover, XCTU software is supported for configuring and programming Xbee, WIFI modules. [14]

- **ZIGBEE settings**

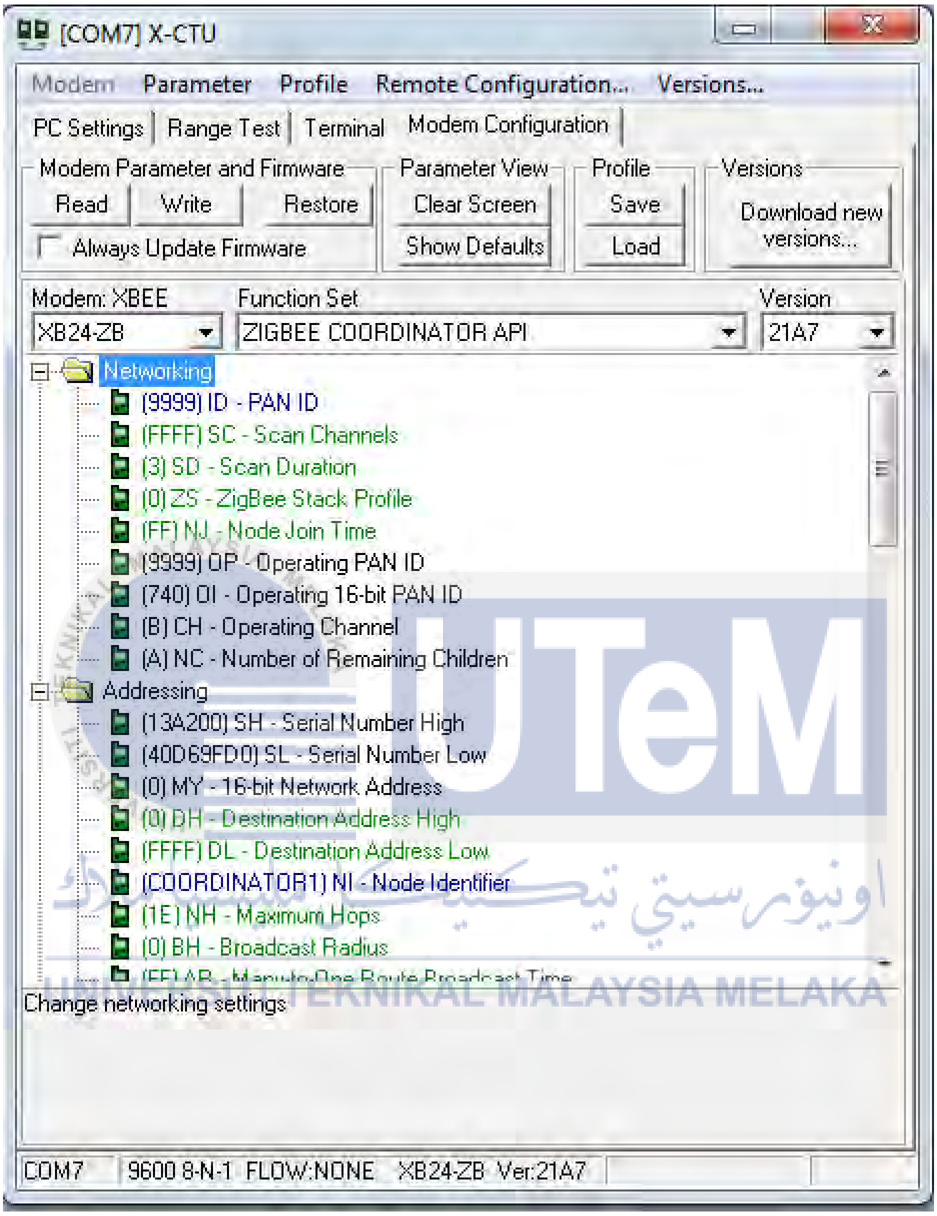

Figure 3.11 "Coordinator API Settings"

Figure 3.11 shows the settings for the ZIGBEE that used as the coordinator and uses API commands. This is the ZIGBEE on the bus stop.

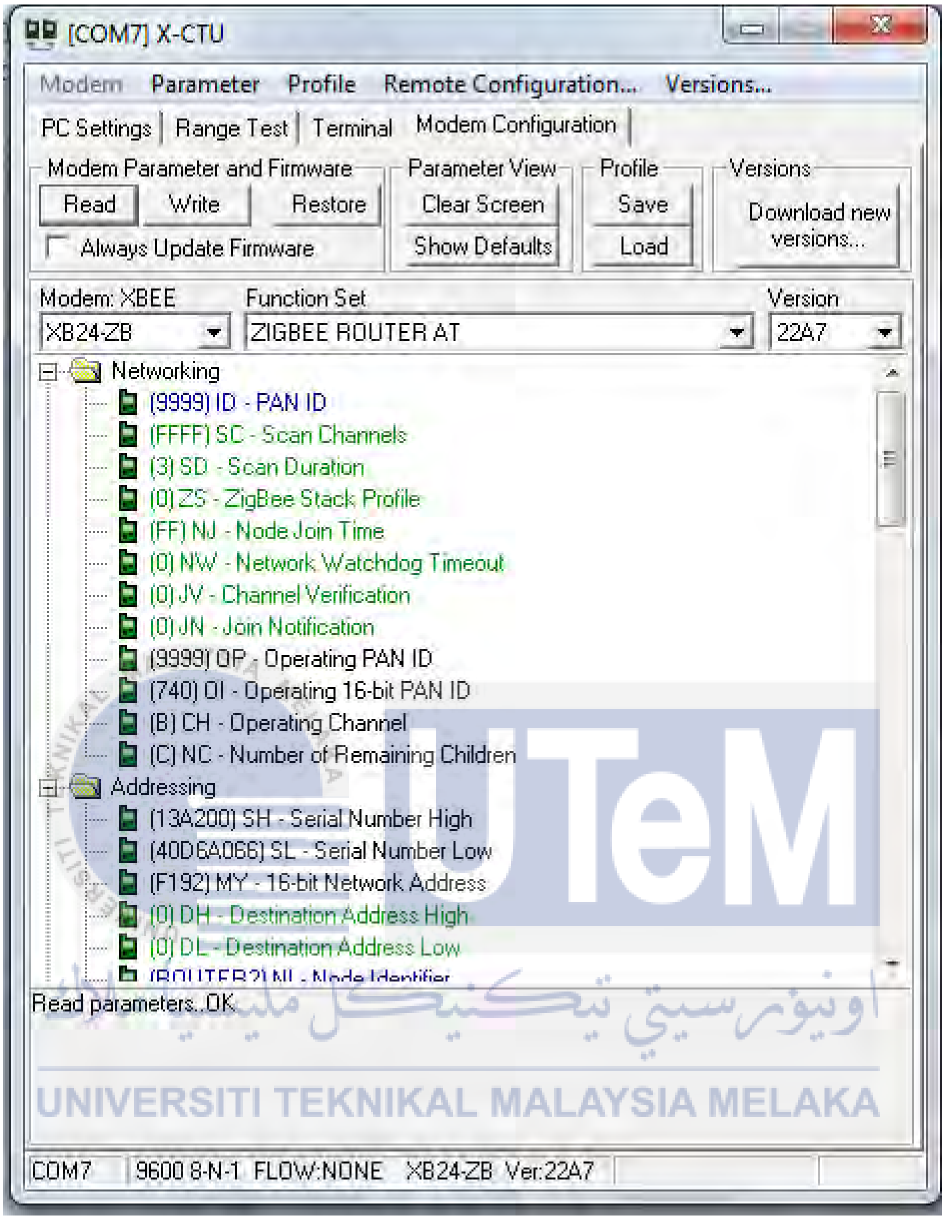

Figure 3.12 "Router1 AT Settings"

Figure 3.12 shows the settings for the ZIGBEE that used as the router1 and uses AT commands. This is the ZIGBEE in Bus1.

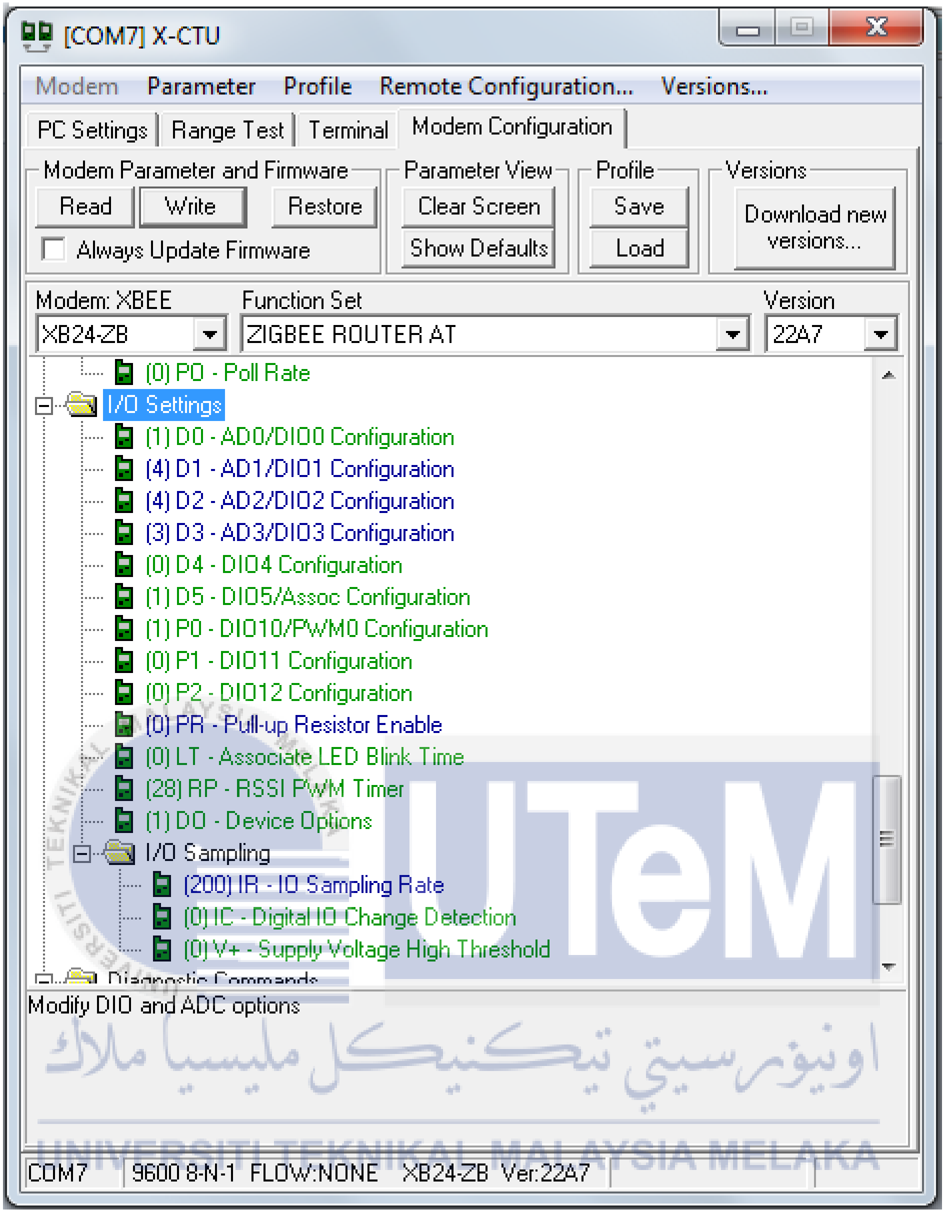

Figure 3.13 "Router1 AT I/O Settings"

This figure shows the I/O settings for the ZIGBEE on Bus1.

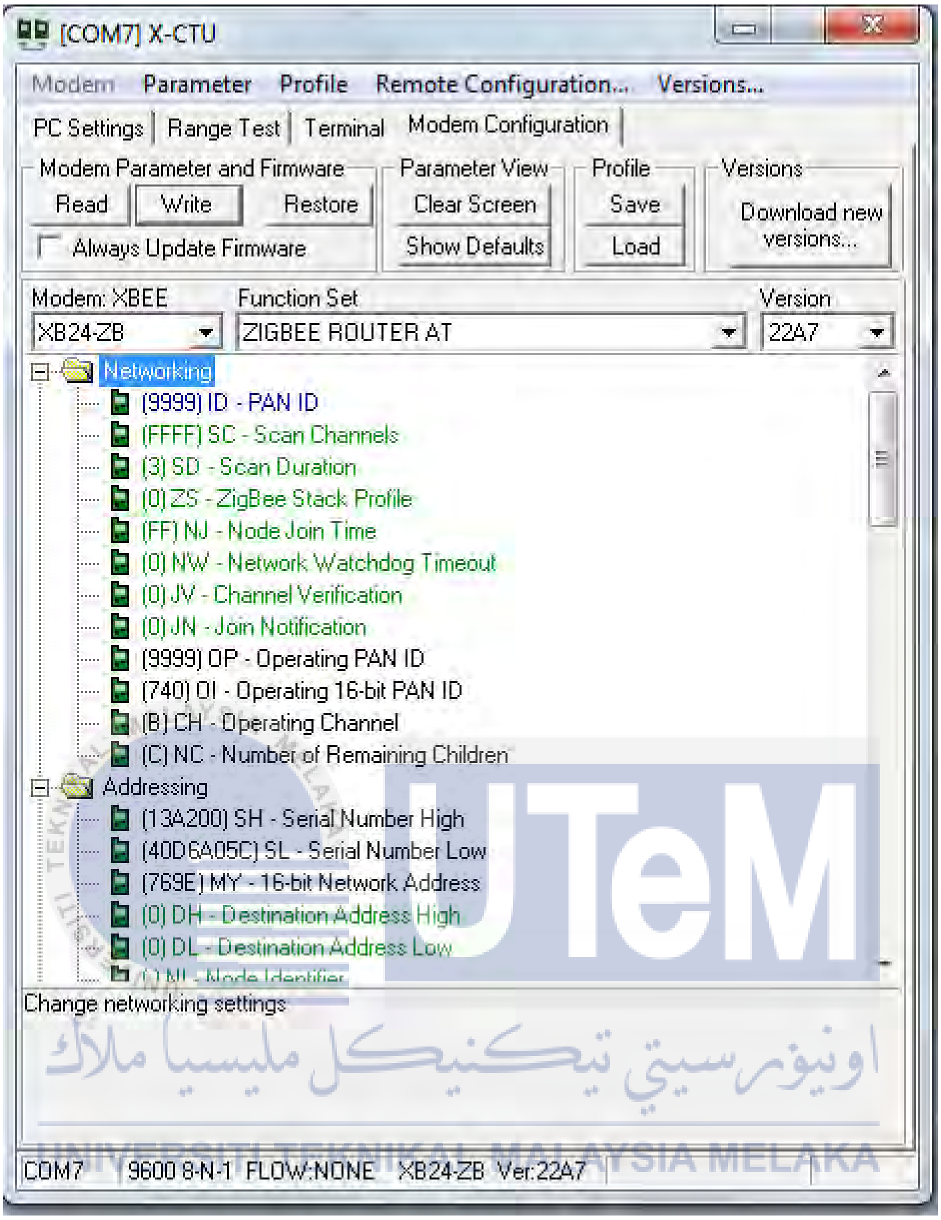

Figure 3.14 "Router2 AT Settings"

Figure shows the settings for the ZIGBEE that used as router2. This is the ZIGBEE in Bus2.

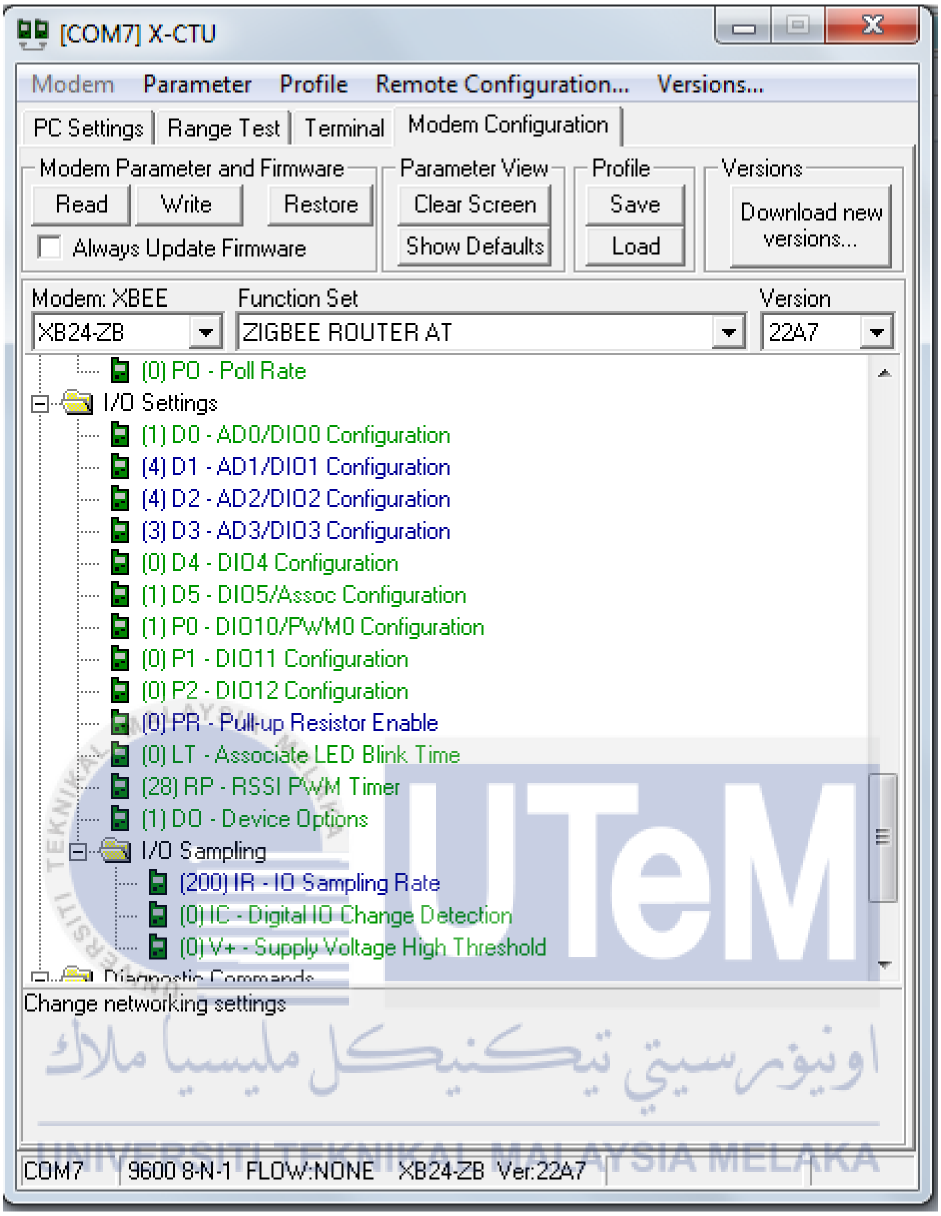

Figure 3.15 "Router2 AT I/O Settings"

This figure shows the I/O settings for the ZIGBEE on Bus1.

CHAPTER 4

RESULT

4.1 Result overview

The outcome of this project is a product that enables data communication between the buses and the bus stop to solve the problems that are faced during waiting at the bus stop. This product made of ZigBee with Radio Frequency as a means of communication with Arduino Uno as the controller. Moreover, a LCD and buzzer are installed at the bus stop to notify the people of the presence of the buses. Furthermore, a LED and buzzer are installed at the buses to notify the bus driver. Also, a reset button been added to the buses for the driver to press it after he receive the notification to avoid the noise and to avoid any repeated notification until the bus arrive at the bus stop.

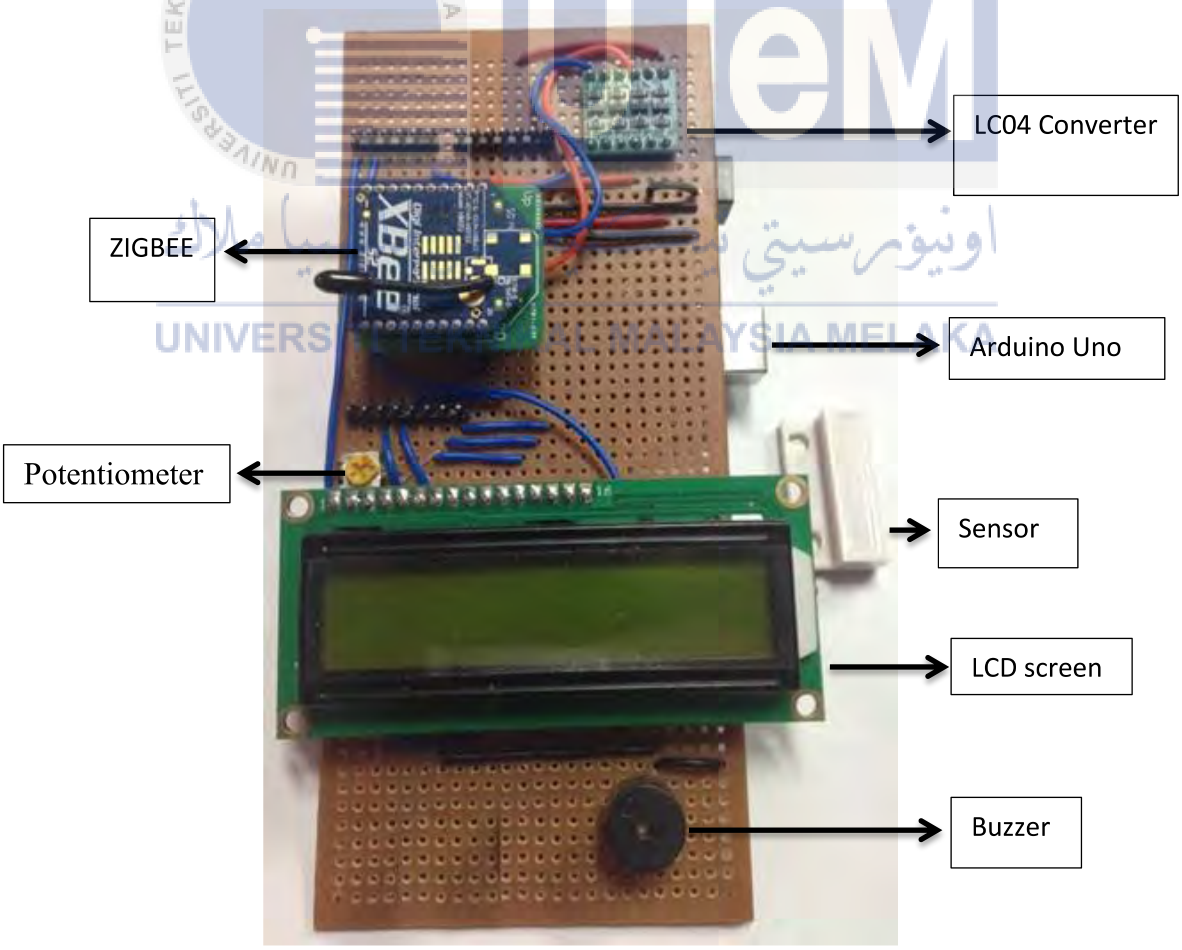

Figure 4.1 "The Bus Stop Prototype"

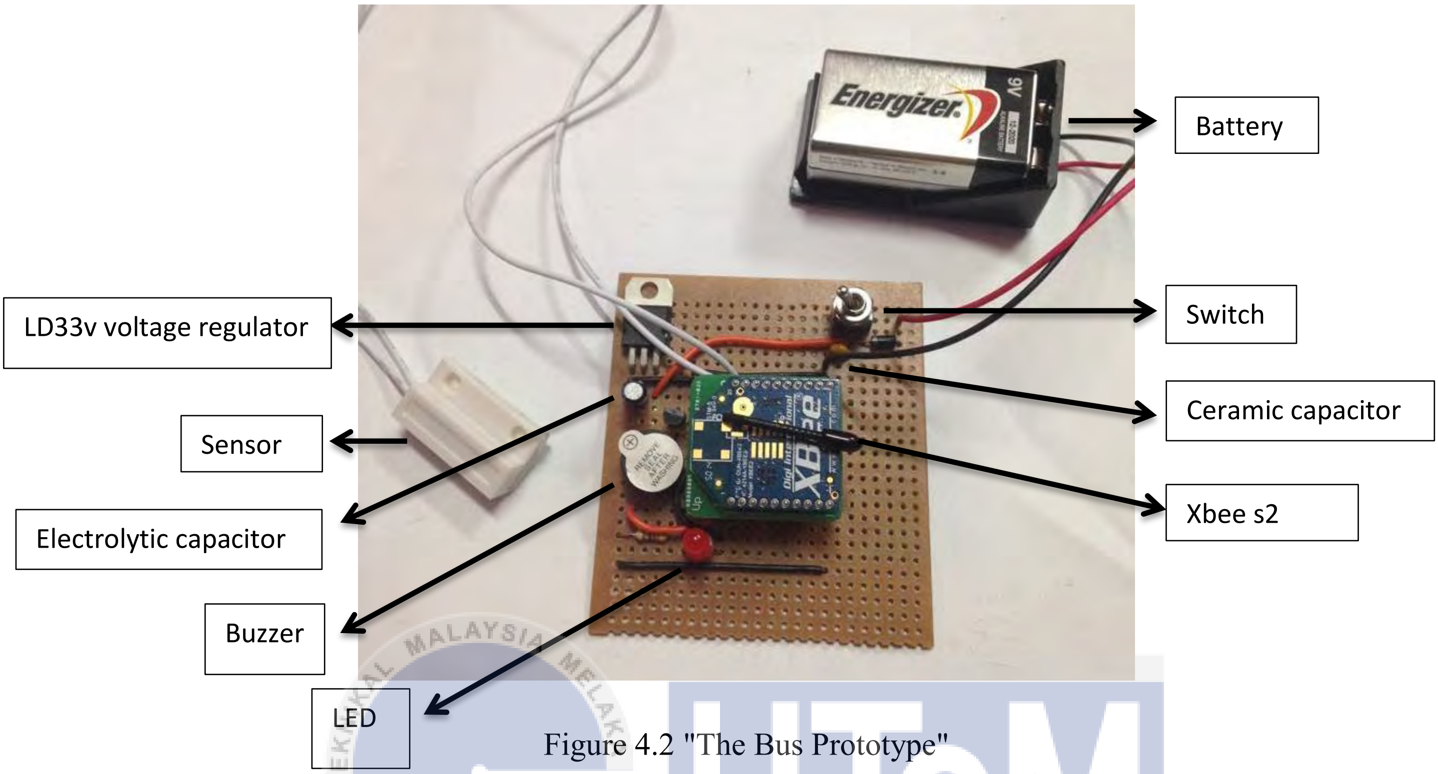

Figure 4.2 "The Bus Prototype"

Figure 4.1 shows the bus station prototype and shows the components of this prototype. Moreover, figure 4.2 shows the bus prototype and shows the components of this prototype. Hence, in this project as explained in the previous chapters there are two bus prototypes.

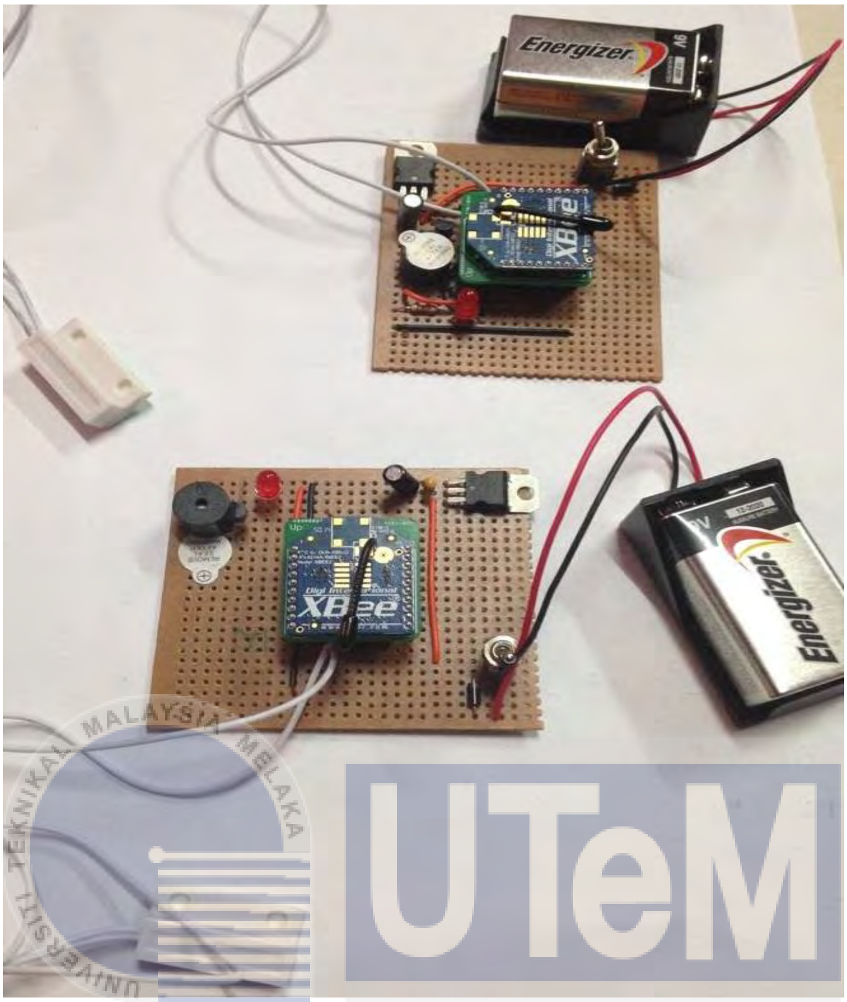

Figure 4.3 "Bus1 and Bus2"

Figure 4.3 shows the two buses prototypes bus1 and bus2.

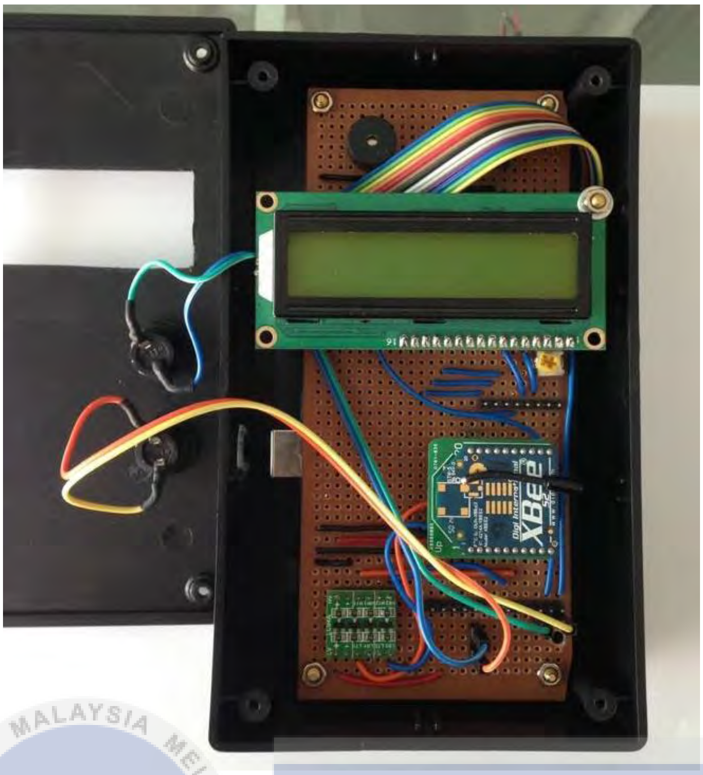

Figure 4.4 "the bus stop inside view"

Figure 4.5 "The bus stop prototype top view"

Figure 4.6 "The bus inside view"

Figure 4.7 "The bus top view"

4.2 Project result procedures

Figure 4.8 "The bus stop starting"

Figure 4.9"The bus stop starting part2"

Figure 4.8 and figure 4 show the first step when the Arduino is connected to the power supply. First the Arduino uno lamps is turned on and then the Xbee is started and initialized. At the same time the LCD shows the message for the starting and initializing step.

Figure 4.10 "The bus stop detecting"

The next thing LCD screen will shows this message as observed from figure 4.10 in case there was not any bus in the range.

Figure 4.11 "The bus stop detecting 2"

When the bus enter the range, this message will be displayed on the LCD screen as shown in figure 4.11

Figure 4.12 "Ordering the first bus"

Figure 4.12 shows the stop when the first bus ordered, also the message will appear on the LCD screen.

Figure 4.13 "Notifying the first bus"

After the bus been ordered the bus stop will send the signal to the desired bus to notify it which the buzzer and led will turn on in the bus and the LCD screen will show this message on the bus stop as shown above in figure 4.13.

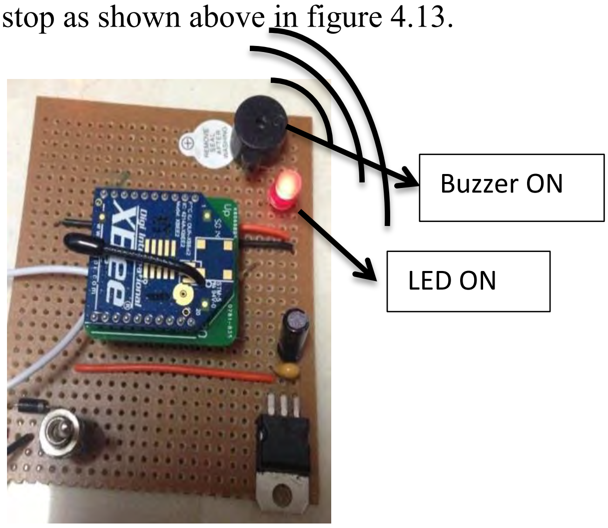

Figure 4.14 "The bus ordered"

Figure 4.15 "The first bus arriving"

Figure 4.15 shows the bus stop when the first bus arrived. At the same time when the bus arrives at the bus stop, the buzzer on the bus stop will turn on and the LCD screen will show the message of the arrival bus to aware the passengers on the bus stop of the presence of the bus.

Figure 4.16 "The first bus leaving"

After that, when the first bus leaves the bus stop the buzzer will turn off and the LCD screen will show this message as shown in figure 4.16.

Figure 4.17 "Notifying the second bus"

In figure 4.17 the second bus ordered and the bus stop send the signal to the second bus to notify it.

Figure 4.18 "The second bus arriving"

The same procedures applied for the second bus also. Figure 4.18 shows the bus stop when the second bus arrives.

Figure 4.19 "The second bus leaving"

CHAPTER 5

CONCLUSION

This project has described a system that is used to help the people in the bus stop from missing their buses and also to help them not to confuse with getting other buses. Moreover, this will help the normal people and also the disabled people as well in the bus stop by implementing the LCD screen and the buzzer. This system was implemented by utilizing the microcontroller and ZIGBEE to make a data communication between the bus and the bus stop. The system was successfully built and functions as aimed by the objective.

References:

[1] Ian Poole, "Zig Bee Technology Tutorial",

http://www.radio-electronics.com/info/wireless/zigbee/zigbee.php, viewed 11/13/2014

[2] GC, "Zig Bee tutorial", 02/18/2013,

http://www.tutorial-reports.com/wireless/zigbee/tutorial.php, viewed 11/13/2014

[3] GUTIERREZ, Jose. (2003) "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks (LR-WPANs)", ISBN 0-7381-3677-5SS95127, Published by the Institute of Electrical and Electronics Engineering, New York, USA. Retrieval September 2008.

[4] G.Lavanya M.E, Preethy. W, shameem.A, Sushmitha. R, "Passenger BUS Alert System for Easy Navigation for Blind", International conference on circuits, power and computing technologies, 2013.

[5] HE. Feng, LI. Lulu, Yin. Heng, Huang, Xia, "Bus monitoring system based on Zigbee and GPRS", International conference on computer distributed control and intelligent environmental monitoring, 2012.

[6] Alan Lowne, President, Saelig Co. Inc. , Hazel Doughty, Sales & Marketing Manager, LM Technologies Ltd. | *Mobile Dev Design* "Bluetooth Moves Beyond The Earpiece To Rule Other Applications" Jun 28, 2010.

[7] http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet

[8] http://www.engineersgarage.com/insight/how-piezo-buzzer-works

[9] "Reed Sensors and Magnets - MEDER electronic", www.meder.com/**reed-sensors**-magnets-us.html

[10] http://www.arduino.cc/en/Main/ArduinoBoardUnoSMD

[11] http://whatis.techtarget.com/definition/light-emitting-diode-LED

[12] https://www.sparkfun.com/products/8375

[13] http://www.electrical4u.com/what-is-capacitor-and-what-is-dielectric/#What-is-Capacitor

[14] http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu#overview

APPENDICES

APPENDIX A

The Bus Station Arduino Programming Codes

```
// include the library code:

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

#include <XBee.h>

#include <SoftwareSerial.h>

// Define SoftSerial TX/RX pins

// Connect Arduino pin 8 to TX of usb-serial device

uint8_t ssRX = A1;

// Connect Arduino pin 9 to RX of usb-serial device

uint8_t ssTX = A0;

// Remember to connect all devices to a common Ground: XBee, Arduino and USB-
Serial device

SoftwareSerial nss(ssRX, ssTX);

XBee xbee = XBee();

ZBRxIoSampleResponse ioSample = ZBRxIoSampleResponse();

int button1 = A5;

int buzzer = A4;
```

```
// Set DIO0 (pin 20) to Analog Input
uint8_t d0Cmd[] = { 'D', '0' };
uint8_t d0Value[] = { 0x2 };
uint8_t d1Cmd[] = { 'D', '1' };
uint8_t d1ValueLow[] = { 0x4 };
uint8_t d1ValueHigh[] = { 0x5 };
uint8_t d2Cmd[] = { 'D', '2' };
uint8_t d2ValueLow[] = { 0x4 };
uint8_t d2ValueHigh[] = { 0x5 };

// SH + SL of your remote radio
//XBeeAddress64 remoteAddress = XBeeAddress64(0x0013a200, 0x400a3e02);
XBeeAddress64 remoteAddress = XBeeAddress64(0x0000, 0xffff);
// Create a remote AT request with the IR command
//RemoteAtCommandRequest remoteAtRequest =
RemoteAtCommandRequest(remoteAddress, irCmd, irValue, sizeof(irValue));
RemoteAtCommandRequest remoteAtRequest =
RemoteAtCommandRequest(remoteAddress, d1Cmd, d1ValueLow,
sizeof(d1ValueLow));
// Create a Remote AT response object
RemoteAtCommandResponse remoteAtResponse = RemoteAtCommandResponse();
boolean switching = false;
//13A20040D6A066 //address router2
//
```

```
                //1234567890123456
String lineClear = "                ";
boolean busOrder = false;
boolean busAvailable = false;
boolean commandSent = false;
boolean commandSent2 = false;
boolean busState1 = false;
boolean busState2 = false;
boolean busState1Prev = false;
boolean busState2Prev = false;
boolean busNo = false;


void setup() {
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("lcd test row 1");
  lcd.setCursor(0,1);
  lcd.print("lcd test row 2");
  delay(1000);
  lcd.clear();
  lcd.print("Starting xbee...");
  Serial.begin(9600);
  // start soft serial
```

```
nss.begin(9600);
xbee.begin(nss);
pinMode(button1, INPUT_PULLUP);
pinMode(buzzer, OUTPUT);


// When powered on, XBee radios require a few seconds to start up
// and join the network.
// During this time, any packets sent to the radio are ignored.
// Series 2 radios send a modem status packet on startup.
// it took about 4 seconds for mine to return modem status.
// In my experience, series 1 radios take a bit longer to associate.
// Of course if the radio has been powered on for some time before the sketch runs,
// you can safely remove this delay.
// Or if you both commands are not successful, try increasing the delay.
  delay(5000);
  //delay(1000);
  lcd.setCursor(0,1);
  lcd.print("Xbee Initialized.");
  delay(1000);
  lcd.clear();
remoteAtRequest.setCommand(d1Cmd);
remoteAtRequest.setCommandValue(d1ValueLow);
remoteAtRequest.setCommandValueLength(sizeof(d1ValueLow));
```

```
 sendRemoteAtCommand();
}
void loop() {
 if(digitalRead(button1) == LOW)
  {
   lcd.clear();
   lcd.setCursor(0,1);
   lcd.print("Bus Ordered!");
   busOrder = true;
   sendRemoteAtCommand();
   remoteAtRequest.setCommand(d1Cmd);
   remoteAtRequest.setCommandValue(d1ValueHigh);
   remoteAtRequest.setCommandValueLength(sizeof(d1ValueHigh));
   sendRemoteAtCommand();
   remoteAtRequest.setCommand(d2Cmd);
   remoteAtRequest.setCommandValue(d2ValueHigh);
   remoteAtRequest.setCommandValueLength(sizeof(d2ValueHigh));
   sendRemoteAtCommand();
  }
 // it's a good idea to clear the set value so that the object can be reused for a query
 remoteAtRequest.clearCommandValue();
 xbee.readPacket(300);
 if (xbee.getResponse().isAvailable()) {
```

```
// got something
if (xbee.getResponse().getApiId() == ZB_IO_SAMPLE_RESPONSE) {
 xbee.getResponse().getZBRxIoSampleResponse(ioSample);
 Serial.print("Received I/O Sample from: ");
 Serial.print(ioSample.getRemoteAddress64().getMsb(), HEX);
 Serial.print(ioSample.getRemoteAddress64().getLsb(), HEX);
 Serial.println("");
 if (ioSample.containsDigital()) {
  Serial.println("Bus Available");
  busAvailable = true;
  //13 A2 00 40 D6 A0 66
  if( ioSample.getRemoteAddress64().getMsb() == 0x13A200 &&
ioSample.getRemoteAddress64().getLsb() == 0x40D6A066)
  {
   Serial.println("Bus One");
   busNo = false;
  }
  else
  {
   Serial.println("Bus Two");
   busNo = true;
  }
 }
```

```
// check digital inputs
for (int i = 0; i <= 12; i++) {
  if (ioSample.isDigitalEnabled(i)) {
    Serial.print("Digital (DI");
    Serial.print(i, DEC);
    Serial.print(") is ");
    Serial.println(ioSample.isDigitalOn(i), DEC);
  }
}
if( ioSample.isDigitalOn(3) == 1)
{
  busState1 = true;
  digitalWrite(buzzer, LOW);
}
else
{
  busState1 = false;
  digitalWrite(buzzer, HIGH);
}
if(busState1 == true)
{
  //lcd.clear();
  lcd.setCursor(0,1);
```

```
      lcd.print("Bus is Near..");
    }
  }
  else {
    Serial.print("Expected I/O Sample, but got ");
    Serial.println(xbee.getResponse().getApiId(), HEX);
  }
}

else if (xbee.getResponse().isError()) {
  Serial.print("Error reading packet.  Error code: ");
  Serial.println(xbee.getResponse().getErrorCode());
  busAvailable = false;
}
else
{
  Serial.print("No bus in range.");
  lcd.setCursor(0,1);
  lcd.print("No bus in range.");
  busAvailable = false;
}
if(busOrder && busAvailable && !commandSent)
{
```

```
sendRemoteAtCommand();

lcd.clear();

lcd.home();

if(busNo == false)

{

  lcd.print("Destination One");

}

else

{

  lcd.print("Destination Two");

}

lcd.setCursor(0,1);

lcd.print("Notifying Bus..");

remoteAtRequest.setCommand(d1Cmd);

remoteAtRequest.setCommandValue(d1ValueHigh);

remoteAtRequest.setCommandValueLength(sizeof(d1ValueHigh));

sendRemoteAtCommand();

remoteAtRequest.setCommand(d2Cmd);

remoteAtRequest.setCommandValue(d2ValueHigh);

remoteAtRequest.setCommandValueLength(sizeof(d2ValueHigh));

sendRemoteAtCommand();


commandSent2 == false;
```

```
commandSent = true;
}
Serial.print("Bus State:");
Serial.println(busState1);


if(busState1 == 0 && busState1Prev == 1 && commandSent2 == false)
{
  sendRemoteAtCommand();
  //lcd.home();
  //lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("Bus Arrived..");
  remoteAtRequest.setCommand(d1Cmd);
  remoteAtRequest.setCommandValue(d1ValueLow);
  remoteAtRequest.setCommandValueLength(sizeof(d1ValueLow));
  sendRemoteAtCommand();
  remoteAtRequest.setCommand(d2Cmd);
  remoteAtRequest.setCommandValue(d2ValueLow);
  remoteAtRequest.setCommandValueLength(sizeof(d2ValueLow));
  sendRemoteAtCommand();
  commandSent2 == true;
  commandSent = false;
  busOrder = false;
```

```
  }
  if(busState1 == 1 && busState1Prev == 0)
  {
    lcd.setCursor(0,1);
    lcd.print("Bus leaving..");
  }


  busState1Prev = busState1;
  busState2Prev = busState2;
  delay(300);
}
void sendRemoteAtCommand() {
  Serial.println("Sending command to the XBee");
  // send the command
  xbee.send(remoteAtRequest);
  // wait up to 5 seconds for the status response
  if (xbee.readPacket(5000)) {
    // got a response!
    // should be an AT command response
    if (xbee.getResponse().getApiId() == REMOTE_AT_COMMAND_RESPONSE) {
      xbee.getResponse().getRemoteAtCommandResponse(remoteAtResponse);
      if (remoteAtResponse.isOk()) {
        Serial.print("Command [");
```

```
    Serial.print(remoteAtResponse.getCommand()[0]);
    Serial.print(remoteAtResponse.getCommand()[1]);
    Serial.println("] was successful!");
    if (remoteAtResponse.getValueLength() > 0) {
      Serial.print("Command value length is ");
      Serial.println(remoteAtResponse.getValueLength(), DEC);
      Serial.print("Command value: ");
      for (int i = 0; i < remoteAtResponse.getValueLength(); i++) {
        Serial.print(remoteAtResponse.getValue()[i], HEX);
        Serial.print(" ");
      }
      Serial.println("");
    }
  }
  else {
    Serial.print("Command returned error code: ");
    Serial.println(remoteAtResponse.getStatus(), HEX);
  }
} else {
  Serial.print("Expected Remote AT response but got ");
  Serial.print(xbee.getResponse().getApiId(), HEX);
}
} else if (xbee.getResponse().isError()) {
```

```
    Serial.print("Error reading packet.  Error code: ");
    Serial.println(xbee.getResponse().getErrorCode());
  } else {
    Serial.print("No response from radio");
  }
}
```

Appendices B: Project flow chart

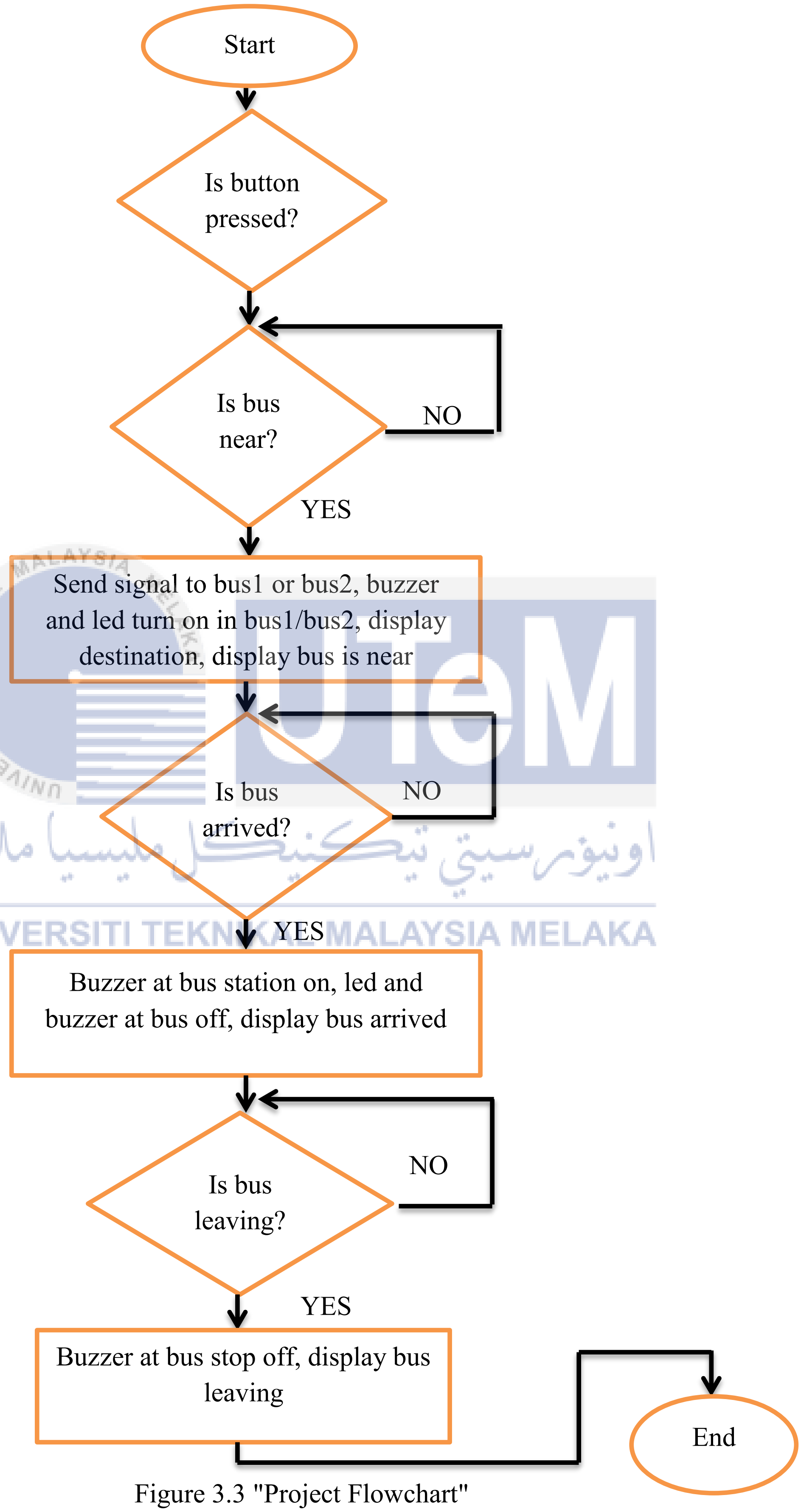

Figure 3.3 "Project Flowchart"