CONTROLLING SPEED AND DIRECTION OF STEPPER MOTOR USING XBEE MODULE

ISAAC KING YIU JOA

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **CONTROLLING SPEED AND DIRECTION OF STEPPER MOTOR USING XBEE MODULE**

SESI PENGAJIAN: 2008/2009

Saya **ISAAC KING YIU JOA (891206-12-5565)**

mengaku membenarkan tesis (Sarjana Muda/Sarjana /Doktor Falsafah)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

[ ]	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
[ ]	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
[√]	**TIDAK TERHAD**	

Disahkan oleh:

____________________________ (TANDATANGAN PENULIS)

____________________________ (TANDATANGAN PENYELIA)

Alamat Tetap:

5037, TAMAN HOT SPRING JAYA,
JALAN AIR PANAS,
91000, TAWAU, SABAH.

NORAZILA BINTI JAALAM

Tarikh: **20 JUNE 2012**

Tarikh: : **20 JUNE 2012**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Power System)"

Signature : ___________________________

Name : NORAZILA BINTI JAALAM

Date : 20 JUNE 2012

CONTROLLING SPEED AND DIRECTION OF STEPPER MOTOR USING XBEE MODULE

ISAAC KING YIU JOA

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Power System)

Faculty of Electrical & Electronics Engineering
University Malaysia Pahang

JUNE, 2012

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : ____________________

Author : ISAAC KING YIU JOA

Date : 20 JUNE 2012

To all my beloved family and friends

ACKNOWLEDGEMENT

Along the road of preparing this thesis, I was in contact with many people, tutors, lecturers and practitioners. They had contributed a lot towards my understand and thoughts related to my project. In particular, I would like to first express my sincere appreciation to my project supervisor, Madam Norazila binti Jaalam, thanks for the continuous leading, guidance and assistances.

Besides that, I am also indebted to Universiti Malaysia Pahang (UMP) for funding Degree Study. Special thanks to lab assistance Mr Azri Bin Idris for his continuous support and advice on handling project components.

Lastly my fellow postgraduate students thanks for all support and motivation along those days. My sincere appreciation also extends to all my colleagues for providing assistance at various occasions.

To all I am indebted. Without their support and interest, this thesis would not have been the same as presented here.

ABSTRACT

Nowadays the remote area is experiencing a great demand for varies field of engineering. Stepper motor, one of the most common motor widely used in controlling especially to achieve a precise measuring of a motor's rotor operation. Stepper motor is the best choice to be chosen for certain applications which requires high specification. Hence the aim for this project is to construct a stepper motor controller while enhancing its performance in terms of controlling.

The purpose of this project is to control the speed and direction of a stepper motor using Xbee module using wireless controlling method. The aim is to able to control the turn of stepper motor rotor direction clockwise or anti-clockwise and decrease or increase the speed. Instead of using in industrial application such as remote control device, valve operation or any other electrical device operation; the device also can be apply for home application such as camera monitoring.

ABSTRAK

Pada masa kini, penggunaan alat kawalan jauh semakin luas di kebanyakan bidang kejuruteran . Motor adalah salah satu alat yang paling biasa digunakan untuk melaksanakan sesuatu kerja. Namun untuk mendapatkan pengukuran yang tepat terutamanya bagi aplikasi yang memerlukan spesifikasi yang tinggi; ‘Stepper motor” adalah pilihan yang terbaik. Maka matlamat untuk projek ini adalah untuk membina sebuah alat kawalan jauh dengan meningkatkan prestasi dari segi kawalan.

Tujuan projek ini adalah untuk membina sebuah alat kawalan yang dapat mengawal kelajuan dan arah pusingan “stepper motor” dengan menggunakan modul Xbee. Dengan itu, motor tersebut boleh dikawal dengan kaedah tanpa wayar di mana kelajuan motor boleh dipertingkatkan atau diperlahankan. Pada masa yang sama, arah putaran motor juga boleh dikawal sama ada mengikut arah jam atau sebaliknya. Peranti ini bukan saja boleh diguna dalam kawasan perindustrian malah boleh juga digunakan di tempat tinggal seperti alat kawalan kamera pemantau.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

T	–	Time period in second (s)
f	–	Frequency in herts (Hz)
R1	–	Resistance in ohms (Ω)
R2	–	Resistance in ohms (Ω)
C1	–	Capacitance in farads (F)

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.0 Background of Study

In this modern era, electric motors are widely used in many fields of engineering and also our daily life electrical devices such as elevator. Specific types of electric motors are designed to meet particular applications. Remote and voltage input of an electric motors are specifically design to aid user easier on controlling and maintenance.

Technology grows faster and bigger, people keen to talk about efficiency, reliability and cost. Instead of using wired push button switches like old times, consumer now can control electrical device using wireless controller. Therefore, this project proposed to develop a wireless remote control for a stepper motor. This controller was developed using a module under ZigBee standard protocol called Xbee module. With a battery supported, this controller can be portable and ease the controller method within the working range.

Hence, as a future electrical engineer, this is an honor and responsibility to develop and share the technology in order to improve community life. This is how the basic idea that leads to this project of developing wireless stepper motor controller. The study and implementation of wireless stepper motor controller is believed to be helpful in controlling all other motor in the future.

1.1 Project Objectives

The vital objectives of this project are:

I. To develop a wireless remote controller for a stepper motor using ZigBee standard module.
II. To control the stepper motor speed; increase or decrease.
III. To control the stepper motor direction; clockwise or anti-clockwise.

1.2 Problems Statement

After doing some research and study, found that there are several problem generally faced in developing a controller. The problems finding are listed as below:

I. Wired controller is costly.
II. Difficult to install (complicated wiring connection).
III. More work to be done for maintenance.
IV. Instability wireless connection.
V. Fault tolerant connection.
VI. Network Configuration (complicated setting).

1.3 Project Scope

The project was proposed with a range of scope listed below:

I. Operating temperature: -30^{0}C ~ 70 ^{0}C.

II. Operation distance: 120 meters insight; 40m indoor/urban range as in datasheet.

III. Controlling stepper motor speed and direction only.

IV. Consist of 2 Xbee module; 1 transmitter and 1 receiver.

1.4 Thesis Outline

The thesis begins with an introduction of the background study about the project in electric motor and the controlling method in Chapter 1. The objectives, problems statement and the project scopes are also clearly stated in this chapter.

In Chapter 2, the chapter will bring out a review of some other relevant techniques used to controller stepper motor. Several examples and ideas which had been done by others are going to discuss in this chapter together with their feedback.

The next Chapter 3 would discuss the methodology of the project which will deal with the design of a wireless controller to control stepper motor. List of component and method would be further describe in this chapter to show how thing work in itself and together to achieve the objectives.

Chapter 4 would be about the result and discussion. In this chapter, the result of the project will be discussed in detail of how the controller works in overall. The flow of the project will be clearly stated in this chapter to show how the controller works.

For the last chapter, Chapter 5 would be the conclusion of the project where the achievements of the project would be listed. Besides that, future suggestion also included for future improvement of the project.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

Nowadays electrical motor is widely used in our daily life, no matter as small as in our laptop's hard disc till as big as in other electrical appliances such as fan, refrigerator, and washing machine and so on. Notice that, most of it is controlled by switches or button which was connected in cable on the electrical appliances; unlike television, a wireless remote control to switch on/off and channel selection. The purpose of this project is to develop a wireless remote control supported with Xbee module which able to control a stepper motor. This controller allows stepper motor to control it speed either to increase or decrease and rotation direction in terms of clockwise or anti-clockwise. Further description on each part supported with reviews and studies will be discussed below.

2.1 Water Quality Monitoring System

Several works had been done using ZigBee wireless network including Water Quality Monitoring System Using ZigBee Based Wireless Sensor Network [1]. Notice that, the stability of ZigBee module is meeting up to the capability to deploy a hoc or continuous monitoring purpose with numbers of sensor nodes. According to the ZigBee Alliance, ZigBee was eventually a communication standard which adopting the IEE 802.15.4 standard for reliable communication [2]. Besides that, ZigBee module also featuring low labor cost, easy to use, low power consumption and reliable data communication between sensor nodes.

2.2 Long Span Bridge Monitoring

Instead of using ZigBee for wireless sensor network (WSN) in water quality, ZigBee also used to monitor the condition of long span bridge [3]. After considering the disadvantages of currently used wire and cable for data communications such as high installation cost of communication and power supply for the sensors, difficult in the installation of steel pipeline for protecting the cables, sensors data distortions due to temperature changes on cables, noise affecting cables and sensor etc. ZigBee commonly used for short distance communication while CDMA (Code Division Multiple Access) infrastructure was used for long distance communication between sensors and the server system [4].

2.3 UPnP – ZigBee Internetworking Architecture Mirroring

Besides that, ZigBee network stability also allows to develop a multi hopping network [5]. Universal Plug-and-play (UPnP) is a good solution for integrating ZigBee networks with internet and for easily configuring ZigBee device. Although UPnP is very useful and widely used in many applications, integrating ZigBee and UPnP is a different approach of non-IP-based devices connection. This is because ZigBee network forms multi-hop connection and supported with limited nodes depending on how many module was in places. UPnP provide transparent interoperation, an UPnP-ZigBee gateway (UZG) creates/terminates virtual UPnP proxies acting as generic UPnP devices by synchronizing UZG with connected ZigBee network topology. This is enabled by injecting network monitoring functions into the ZigBee network and combining them into UZG [6].

2.4 Field Programmable Gate Array (FPGA)

There are many other wireless controllers that had been developed which available for stepper motor instead of ZigBee module. A reconfigurable wireless stepper motor controller based on Field Programmable Gate Array(FPGA) implementation is another great example which had been developed [7]. The idea is to design a controller using Very High Speed Integrated Circuit Hardware Description Language (VDHL) together implementing it on Spartan 3E FPGA. The controller is using Pulse-Width Modulation (PWM) technique to rotate the motor with a desired angle, speed and direction with a latent time of 10ms.

2.5 System on Programmable Chip (SOPC)

Another type subdivided controller of stepper motor based on system on Programmable Chip (SOPC) Technology [8]. The design method is based on the Task Flow Graph (TFG) model of the digital signal process algorithm and implemented with SOPC technology. Notice that, stepper motor system consist of 3 parts which are control section, driver section and stepper motor itself. Nios II CPU was selected for the SOPC system as the control core. This embedded system has a strong computing ability and a good networking configuration property [9]. In addition, due to the application of subdivided driving with SOPC, the stepper motor control system could meet the requirement such as some high precision positioning and accurate machining.

2.6 Single Microprocessor Controlling DC Stepper Motor

Stepper motor controlled by a single microprocessor was developed by sending pulse sequences to the motor winding in response to control commands [10]. Notice that the command pre-type and burned into the microprocessor. When the system initialized, the command executed by the code. The project also includes controlling the stepper motor rotate direction. The speed of the stepper motor rotation is controlled by a 16-bit timer. The timer was setup with a value that cause an underflow once every a variable second at a fix frequency [11].

2.7 Steering Camera by Stepper Motor towards Active Speaker

Many devices such as survey, monitoring and video recording of an event need an automatic camera steering [12]. This is highly important since the person who operates was located at another place. However the project was further suggested to uses a linear microphone array to get speech or audio signal from speakers integrating with a stepper motor based camera steering unit (CSU) [13]. These speech or audio then will compared to estimate direction of arrival (DOAs) of the speaker using multiple signal classification (MUSIC) algorithms. These estimated DOA will send signal to CSU which installed with a stepper motor for turning the camera to desire direction. Notice that, the experimental results propose that a high resolution MUSIC algorithm can be effective in steering camera especially during noisy environment too if the number of sensors in array is increased in the same time [14].

CHAPTER 3

METHODOLOGY

3.0 Introduction

In order to develop a wireless controller, several devices and software configuration are needed. The basic overall process flow of the project has been shown in Figure 3.1 as below. Generally the whole system was divided into a few main parts which include the controller, transmit medium and end device which was the execution device to move the stepper motor. However when come into solving each puzzle and put it all together to solve the matter, the whole process flow of the project had been illustrate in Figure 3.2.

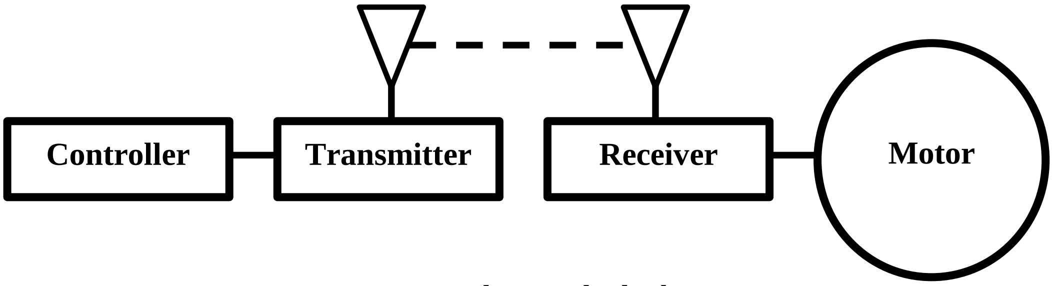

Figure 3.1: Fundamental Block Diagram

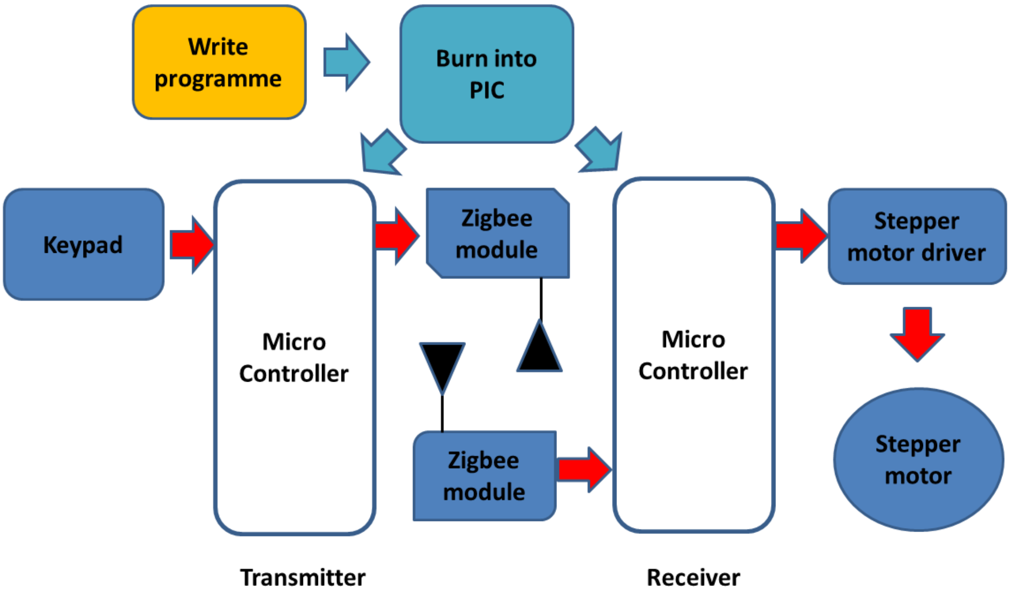

Figure 3.2: Overall Process Flow of Controlling Stepper Motor

3.1 ZigBee

3.1.1 Introduction

Millions of implementations built using ZigBee standards prove it can rely on the widest variety of smart and easy-to-use products for just about anywhere work, live or play [2]. For that reason this project are proposing to use Xbee module as the medium of transmit data, where this product was built according to ZigBee standards. Due to limited budget and stock availability, “XBEE 1mW Wire Antenna – Series 2” was selected to use in this project. Although there are other even high end types of Xbee, this product was still in the group of Series 2 which was the latest technology available in the market. Basically both Series 1 and Series 2 are the same in way of usage, just the manufacturing of the product had been improved in terms of current usage and transmitting range. Besides that, Series 2 also introduce Mesh connections which will be further discuss in Xbee network topologies. The sample of Xbee module has been shown in Figure 3.3.

Figure 3.3: XBEE 1mW Wire Antenna – Series 2

3.1.2 Xbee Network Topologies

First of all, ZigBee define three different types of device which are coordinator, router and end device. The definition, comparison, and function of each device had been stated clear in Table 3.1 below.

Table 3.1: Comparison of Coordinator, Router and End device.

Device	**Coordinator**	**Router**	**End device**
Function	Select channel and PAN ID to start a network.	Must join ZigBee PAN before it can transmit, receive or route data	Must join ZigBee PAN before it can transmit, receive or route data
Allow router and end device to join	Yes	Yes, only after joining.	No
Assisting in routing Data	Yes	Yes, only after joining.	No
Sleep	Not allow	Not allow	Allow

From the table, notice that Coordinator and Router plays important role in buffering radio frequency data packets for end device which was connected into the network. An example of such network has been illustrated shown below in Figure 3.4.

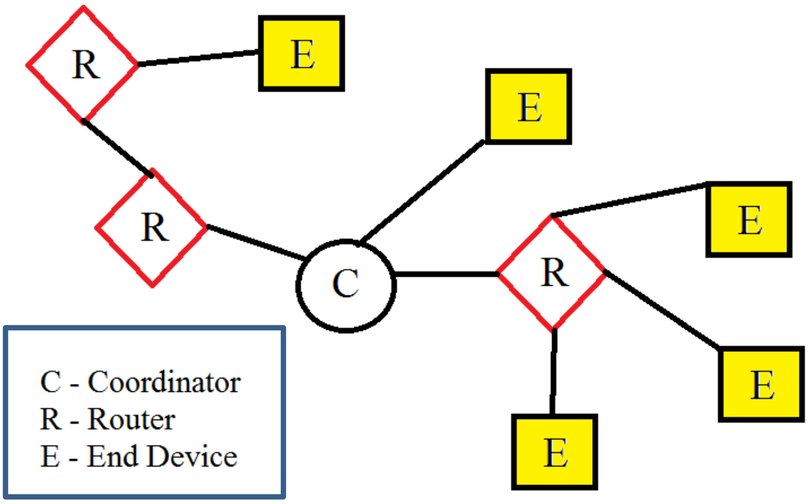

Figure 3.4: ZigBee Network Connection Example.

The network is called personal area networks, PANs and each network is defined with a unique PAN identifier called PAN ID. A coordinator is needed to configure a PAN ID to allow other devices to join. Devices can be manually set or discover nearby networks themselves and select to join.

Generally ZigBee network was set up base on three types of topology which is Point-to-point, Many-to-one and the latest released Mesh routing. Many-to-one routing is use when there are many devices sending data to the central collector. Notice that if the device itself had to discover a route before it can send data, the network could flood. In order to overcome this problem, instead of each device searching its own route, a single many-to-one route was set to send data from data collector while establishing a reverse route on all devices. The network has been illustrated in Figure 3.5. In that case, when a device send data to data collector, it will find a many-to-one route and transmit the data without searching a new route instead.

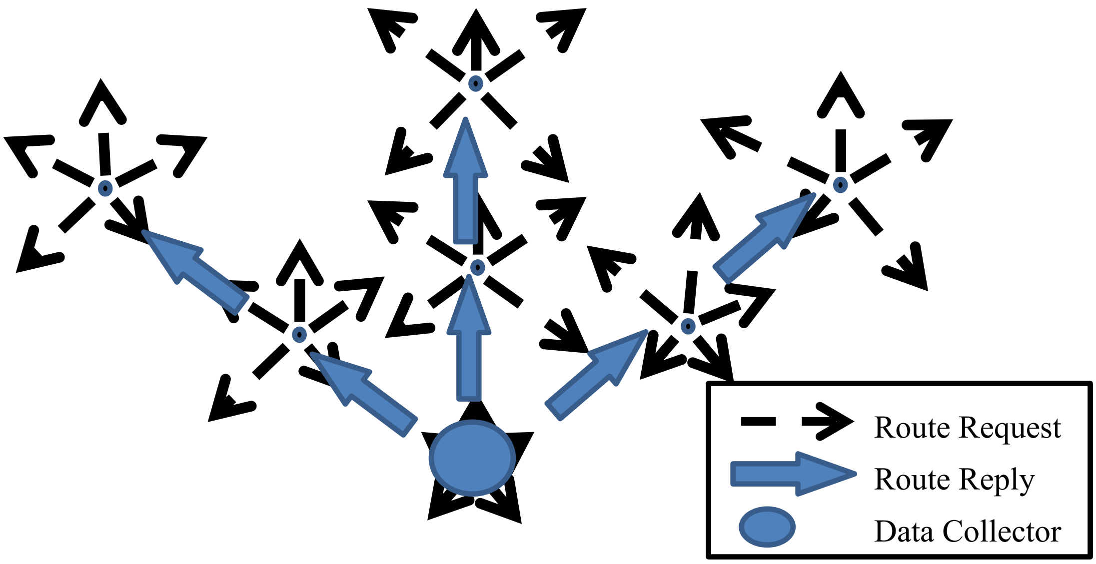

Figure 3.5: Many-to-One Network Illustrates.

Unlike many-to-one, Mesh routing is to build a route between source device and destination where the data was allowed to travel multiple nodes or also known as hops. Router and Coordinator took parts on building route between source and destination device using a process call route discovery. This process would begin from sending route request, compare the options and select the most ideal route path to send the data to destination. Figure 3.6 and Figure 3.7 shows an example on how the process works.

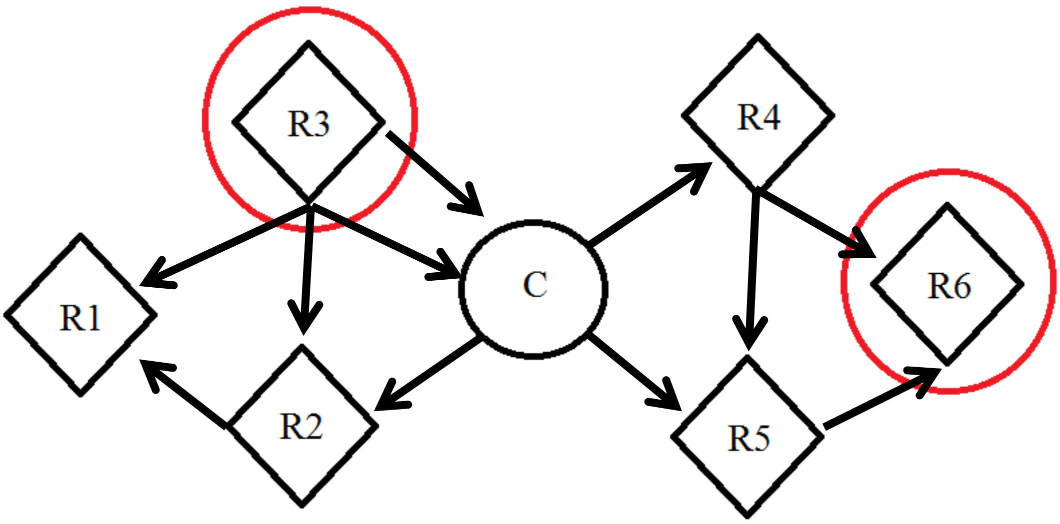

Figure 3.6: Sample Route Request for Data Sending from R3 to R6

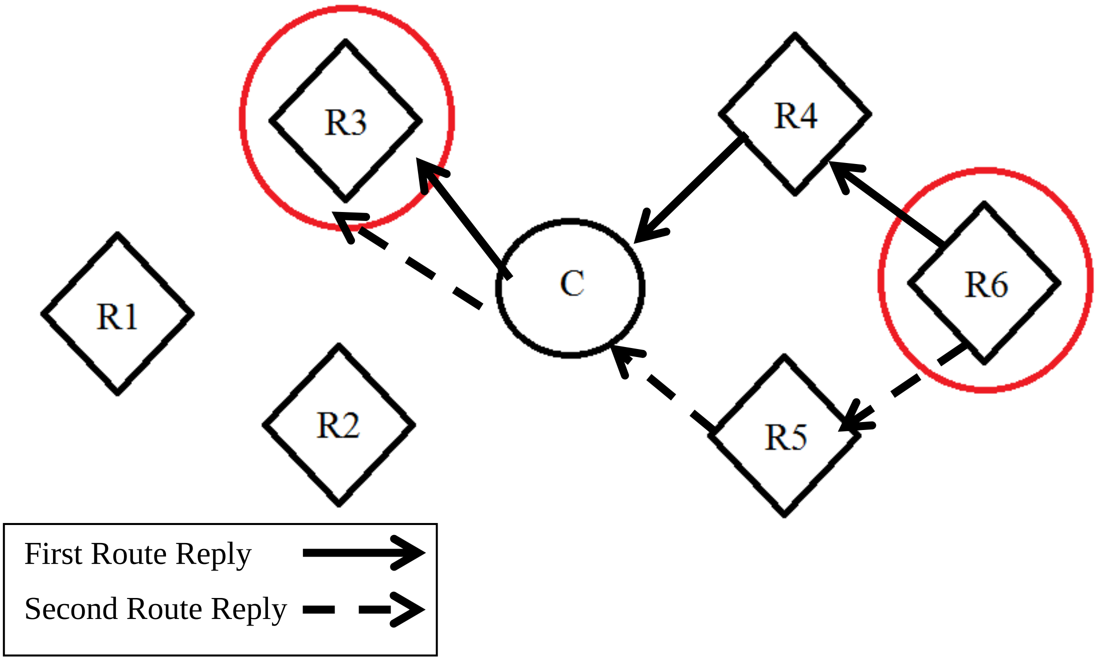

Figure 3.7: Route Reply from R6 to R3

Figure 3.6 shows that, R3 was sending multiple routes request to other router in order to search the suitable path. When the destination node receives a route request, it will automatically compare the path against other request. If it was better than any other, the destination node will transmit a route reply to the originate route request node. Instantly after receiving the reply, source node will packet up the data and send to destination shown in Figure 3.7.

Last is Point-to-point routing, this is the simplest routing communication where it was only consist of two points. Only one coordinator and one router are needed to establish this connection. After set the same PAN ID for both nodes, it will automatically connect to each other once both nodes were in transmitting range.

The method that going to be use in this project is Point-to-point routing, because it was only consist of two point; controller and stepper motor. The objective is to send data from the controller to another node which connected to the stepper motor driver and drive the stepper motor on command. The communication can be easily set up by using X-CTU software which will be further discuss in following subchapter.

Notice that all route discovery process is based on AODV (Ad-hoc On-demand Distance Vector routing) protocol. The process is accomplished using tables of each node that store the next hop for destination nodes. If next hop is not known route discovery with take place in order to find another path. Route discovery usually take place more often in a large network with communication between many different available nodes [2].

3.2 Controller Design

The controller method is simple and efficient where simple push button was selected for the design. In order to determine the numbers of button needed, first it must come to understand the objective of the project. Notice that main objective of the project is to control the speed (increase/decrease) and direction (forward/reverse) of the stepper motor. Additional an ON/OFF switch also include for starting or stopping the stepper motor. A reset button also added to reset both PIC18F4550 and Xbee module. In conclusion, 7 buttons was decided in the design for controller, the number and function of each button had been clearly stated in Table 3.2 below. The button will be connected to PIC18F4550 as an analog input, wiring connect is shown in Figure 3.8.

Table 3.2: List of Button and Function.

Button number	**Button Title**	**Function**
Button 1	On button	To enable stepper motor to start
Button 2	Off button	To disable stepper motor to stop.
Button 3	Forward button	To switch stepper motor rotation to forward direction.

Button 4	Reverse button	To switch stepper motor rotation to reverse direction.
Button 5	Speed increase button	To increase stepper motor rotation speed.
Button 6	Speed decrease button	To increase stepper motor rotation speed.
Button 7	Reset button	Reset button – to reset PIC18F4550 and Xbee

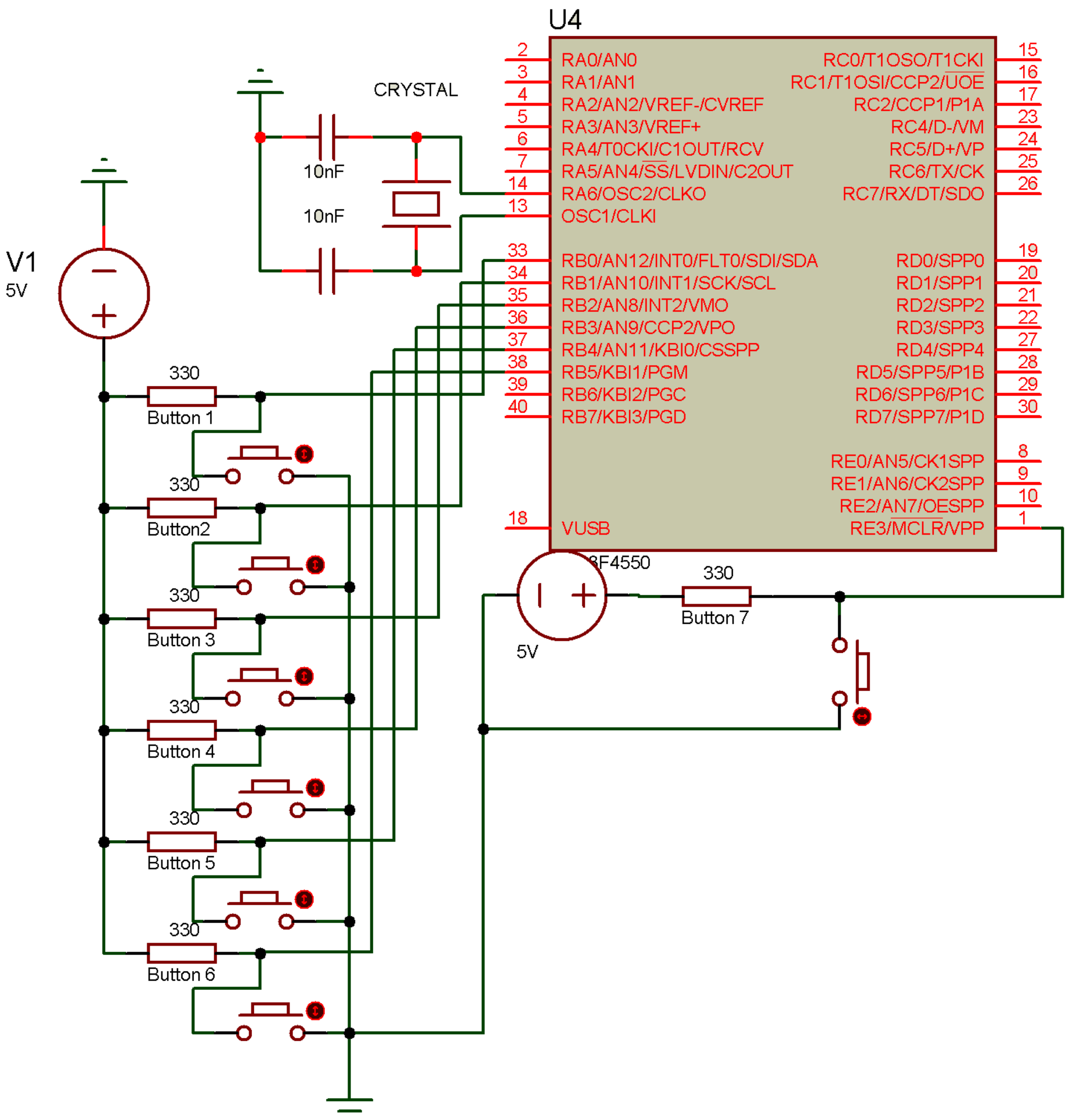

Figure 3.8: Button Wiring Connection.

Every button will be connected to PIC18F4550 as an analog input to send out order to execute each particular function. Wiring connection will be further discussed in PIC18F4550 subchapter. Speed availability has been manually set. 4 different speeds had been set in PIC18F4550 by using TIMER555 which will further discuss below. Controller method only discuss the way how analog input was send to the system to execute designate command, further control of stepper motor would be discuss in next subchapter in Stepper motor driver.

3.3 Stepper Motor

3.3.1 Introduction

Stepper motor defines as a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor consist of a permanent magnet rotating shaft (rotor) and electromagnets on the stationary part around the motor (stator). When each particular stator was electrified, rotor will divert their direction to it. The rotation of stepper motor was illustrated as in Figure 3.9 below.

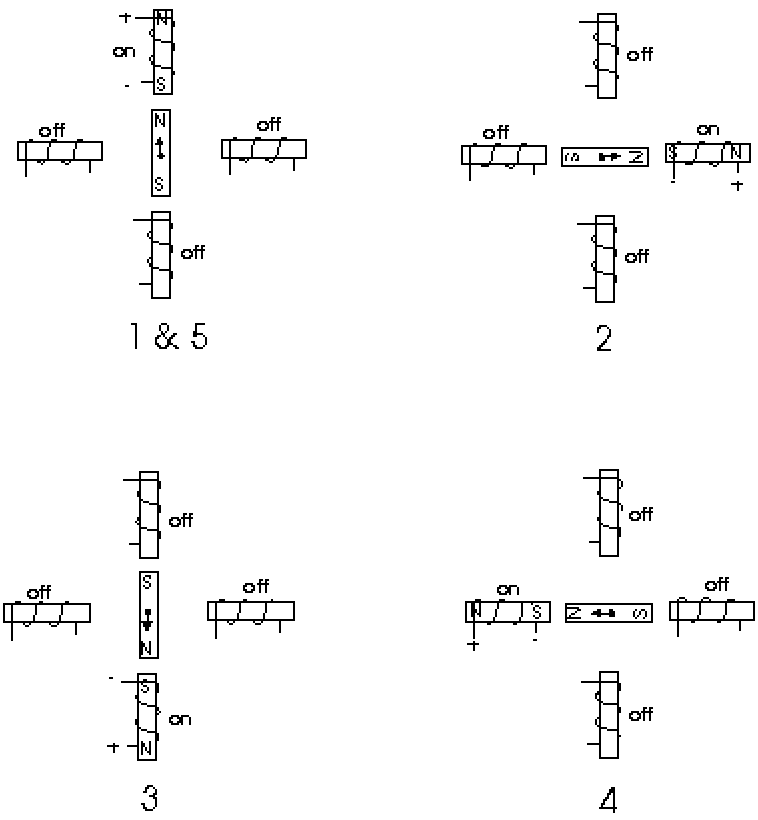

Figure 3.9: Stepper Motor Rotation.

Notice that, the North Pole of the rotor always point at the electrified magnetic bar. If the stator was supplied electricity in a designated order, the rotor will rotate in a smooth and continuous direction. If the order was reversed, the rotor will rotate at opposite direction.

3.3.2 Stepper motor

Generally there are two types of stepper motor which are Unipolar and Bipolar. Unipolar has four windings and easy to drive with low torque and speed while Bipolar has two winding and hard to drive with high torque and speed [X]. Since the aim of this project is to drive a stepper motor, unipolar stepper motor has been chosen to do the task. Base on the availability stepper motor from university store, it has been chosen with specification as listed in Table 3.3 below.

Table 3.3: Stepper Motor Specification.

Stepper motor type	Unipolar
Rated Voltage	12Vdc
Rated Current	16mA
Number of phase	4
Degree per step	7.5 degree

Unlike the other motor, stepper motor talks about high precision on each movement where each steps was fix a certain degree with very minor alignment. Instead of taking full step, stepper motor also can be controlled to make a half step rotation when higher precision step was needed. Figure 3.10 simulated how half step of the stepper motor can be made.

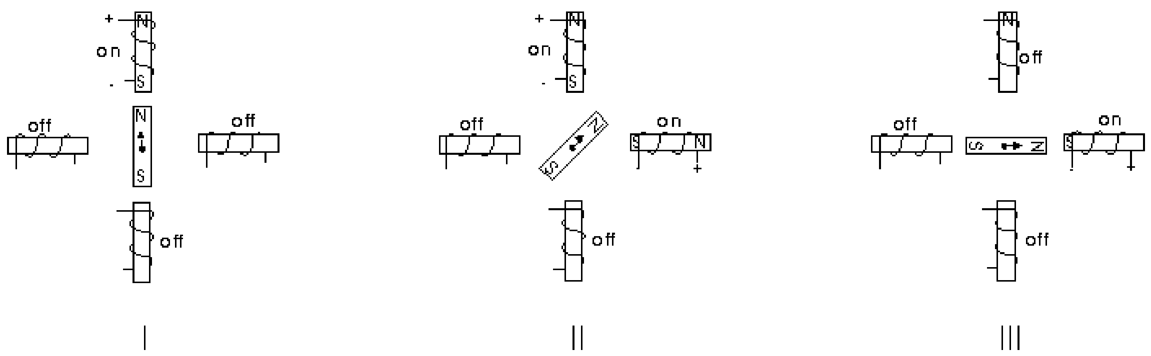

Figure 3.10: Stepper Motor Half Step Rotation.

The stepper motor come with 6 output wires. Each output connection had been listed in Table 3.4 below and sample of stepper motor outlook also included are shown in Figure 3.11.

Table 3.4: Stepper Motor Wires Color and Function.

Wire Color	**Function**
Red 1	Common wire, use to supply 12V
Red 2	Common wire, use to supply 12V
Black	Connected to Coil A, grounded when stepper motor step one.
Yellow	Connected to Coil B, grounded when stepper motor step one.

Grey	Connected to Coil C, grounded when stepper motor step one.
Orange	Connected to Coil D, grounded when stepper motor step one.

Figure 3.11: Stepper Motor Outlook.

3.3.3 Stepper Motor Driver

In order to drive a stepper motor from PIC18f4550, an IC ULN2003A is more than enough. Sample of connection was shown in Figure 3.12. Using the very basic of time delay in between each sequence, the speed and direction of the stepper motor can be easily controlled. However this does not works out when the input was transmitted from another PIC18F4550 due to the time delay. This is a major problem since the objective of the project is to develop a wireless controller.

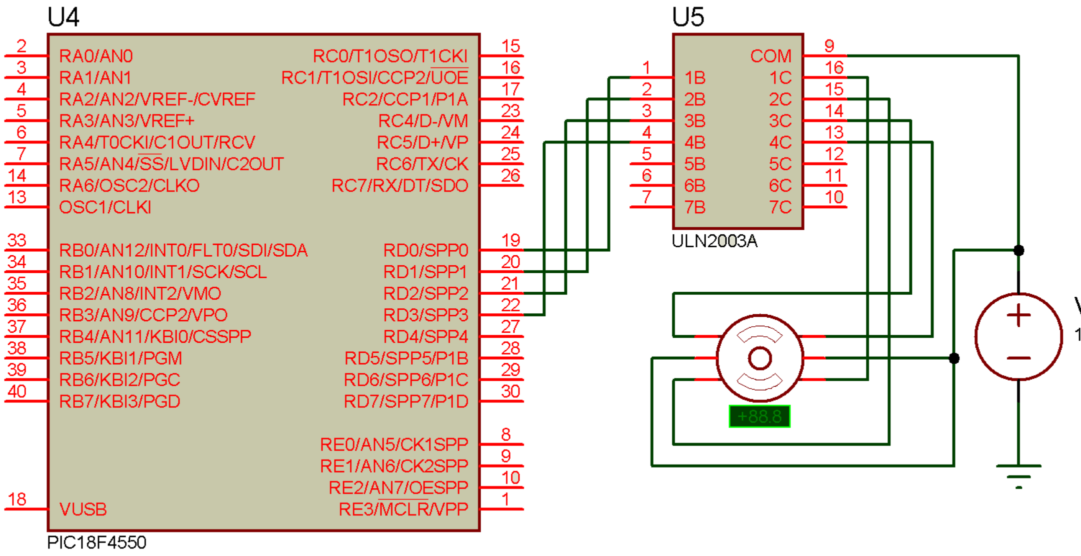

Figure 3.12: Controlling Stepper Motor using ULN2003A.

In order to overcome this problem, another method had been introduced by using two IC which is L297 and L298. In IC L297 the sequence of the stepper motor rotor has been manually set in the IC itself, varying the ENABLE pin, CW/CCW pin and CLOCK pin, the stepper motor on/off state, speed and direction can be controlled by one input. So now the data was able to successfully sent and receive at the end device. A sample of connection was shown in Figure 3.13.

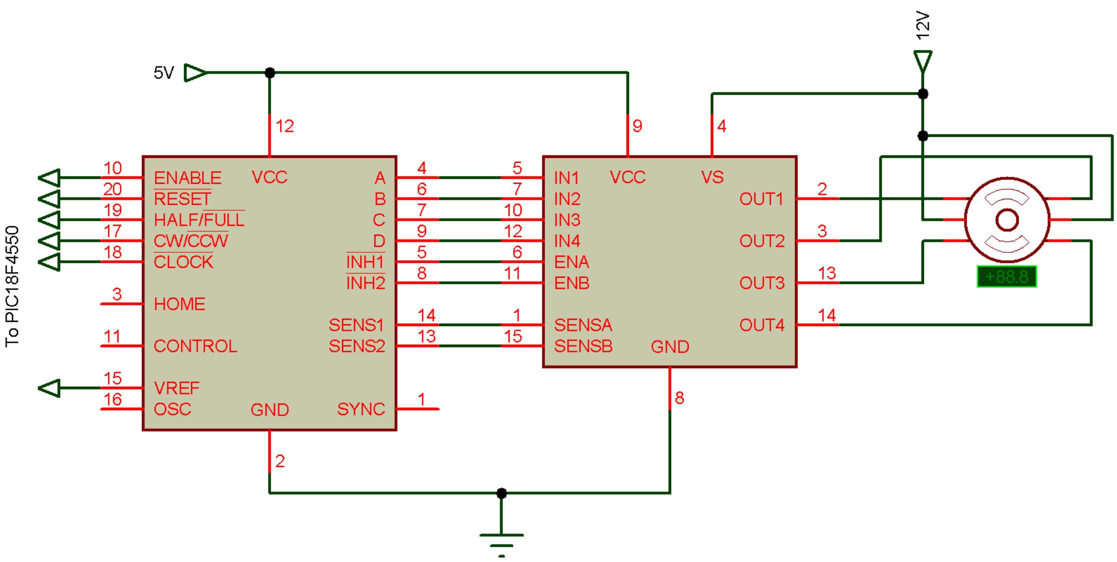

Figure 3.13: Controlling Stepper Motor using L297-L298.

Notice that L298 can handle up to 3A of output current, which was way too high since the stepper motor only need 16mA. L298 cost about Rm20 which would be waste. So the project had redesign by combining the advantage of IC L297 with IC ULN2003A. The connection was shown in Figure 3.14.

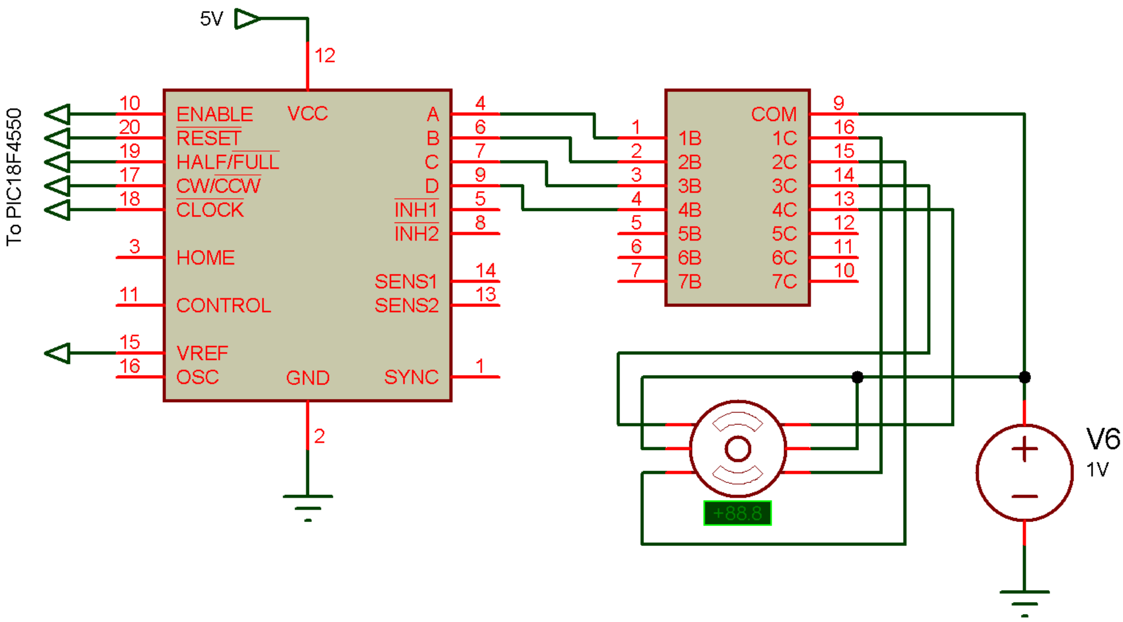

Figure 3.14: Controlling Stepper Motor using L297-ULN2003A

Important pins description and function had been tabled in Table 3.5 for IC L297 and Table 3.6 for IC ULN2003.

Table 3.5: Pin Description for IC L297

Pins	**Description and function**
RESET	- Reset pin use to reset IC. - This pin has to always HIGH when operate else the step will block at state 1 (ABCD = 0101).
ENABLE	- HIGH to allow voltage pass through. - LOW to block voltage pass through.
V_{ref}	- Reference voltage for chopper circuit. - To determine the peak load current.
CW/CCW	- Clockwise/counterclockwise direction control input.

CLOCK	- Step clock, an active low pulse on this input to advance the motor increment in one. - Directly control the rotation speed of the stepper motor.
V_s	- 5v input to turn on IC.
GND	- Ground terminal.
A,B,C,D	- 4 output in sequence generated by the IC itself.
HOME	- An open collector use to indicate when the IC back to state 1 (ABCD = 0101)
HALF/FULL	- Half/full step select input.

Table 3.6: Pin Description for IC ULN2003A

Pins	**Description and function**
COMMON	- Connect 12V supply same with stepper motor.
GND	- Ground terminal.
PIN 1,2,3,4	- Input from IC L297 (5V)
PIN 16,15,14,13	- Output to stepper motor (12V)

These both devices will be simulated using Protues ISIS Schematic Capture software and tested with real component. Analysis and modifying will keep update until the system was function properly as designated.

3.4 Pulse-width Modulation

3.4.1 Introduction

Pulse-width modulation also known as PWM is a general technique use in controlling the input power of electrical devices. It generates a pulse of high and low which directly manipulate the output voltage from a zero to five voltages. One of the examples of PWM usage is to control the speed of motor. CLOCK pin was designed to receive PWM input to manipulate the motor rotation speed. Notice that PIC18F4550 itself is able to generate a PWM output, but the frequency was just way too high to control the stepper motor rotation. There is a limit of frequency for a stepper motor to work in proper condition, because if the speed was too fast the driver itself will jump step and the motor rotation will not smooth. In order to overcome this matter, IC TIMER 555 was proposed to handle the pulse generation.

3.4.2 IC TIMER 555

NE555 also known as TIMER 555 is useful device use on handling precision timing circuit which capable to produce an accurate time delay or oscillation. Generally there are two types of operation known as monostable and astable. The different is monostable only generate 1 pulse until next change while astable generate continuous pulse which is what needed for the project to drive the stepper motor. Astable TIMER 555 operation can be set up as like connection shown in Figure 3.15.

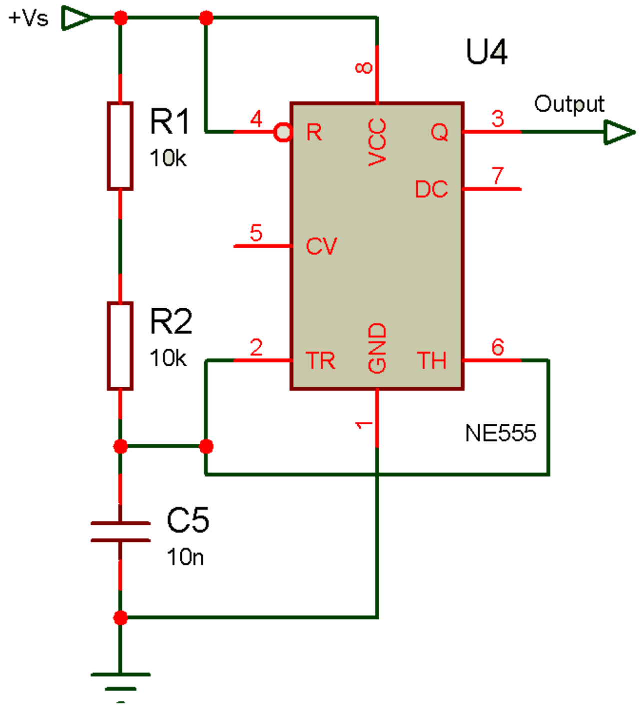

Figure 3.15: TIMER 555 astable circuit.

An astable circuit can produces a square wave shown in Figure 3.16; the high and low state duration is able to vary by changing the R1 and R2 value.

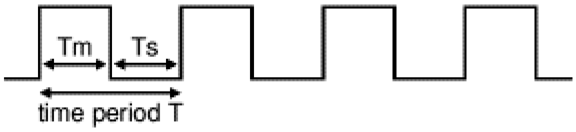

Figure 3.16: Square wave

The time period (T) of the square wave is the time of one complete cycle, usually number of cycles per second is determine by frequency (f). The calculation made is shown in Appendix B and list in Table3.7 below. Then it was manually set to implement for controlling stepper motor rotation speed.

Table 3.7: TIMER 555 Astable Frequencies.

Capacitance, C1 (µF)	**Resistor, R1 (kΩ)**	**Resistor, R2 (kΩ)**	**Frequency (Hz)**
10	0.47	5	13.51
10	1.00	10	6.80
10	5.00	50	1.36
10	10.00	100	0.68

However duty cycle had to be considered for the project in order to archive a smooth rotation, duty cycle must achieve 50% as close as possible. In order to do so, selecting resistor with considering the ration between R1 and R2 is very important. Calculation was made as shown in Appendix B and shown the result in Table 3.8 below.

Table 3.8: Duty Cycle and High/Low Time Period.

Resistor, R1 (kΩ)	**Resistor, R2 (kΩ)**	**Duty cycle, D (%)**	**High time, Tm (s)**	**Low time, Ts (s)**
0.47	5	52.24	0.039	0.036
1	10	52.38	0.077	0.070
5	50	52.38	0.385	0.350
10	100	52.38	0.770	0.700

From the Table 3.8 above, notice that the duty cycle is equal to 52.38% for each selection pair of resistor which had proven it was suitable since it was close to 50%. The high time and low time period also calculated to show the time taken for stepper motor to make the next step from each stop.

3.5 Microprocessor

3.5.1 Introduction

Nowadays in modern life, many electronic device use microprocessor to handle general task such as remote control, countdown timer, lighting advertisement and so on. Notice that this small device plays an important role to the whole system as like the mind of everything. Microprocessor is so general and can be programmed using different language such as C-language and Assembly language. For this project, PIC18F4550 was selected to do task. This is because it can be easily found in the market and widely used in many applications.

3.5.2 PIC18F4550

Basically in this project PIC18F4550 was used for 4 purposes which are receiving analog input, detect and transmit data, receive and read data, and finally execute output signal to drive the stepper motor. After discussing each and particular part of the project, now every single unit was place together to work. Figure 3.17 shows the whole wiring connection of each device.

The design was based on certain testing on each component to make sure it fit into. Simulation software, PROTUES also used to generate a simulation of the process to troubleshoot if any problems or design fault occur. PIC18F4550 also programed using PIC C Compiler.

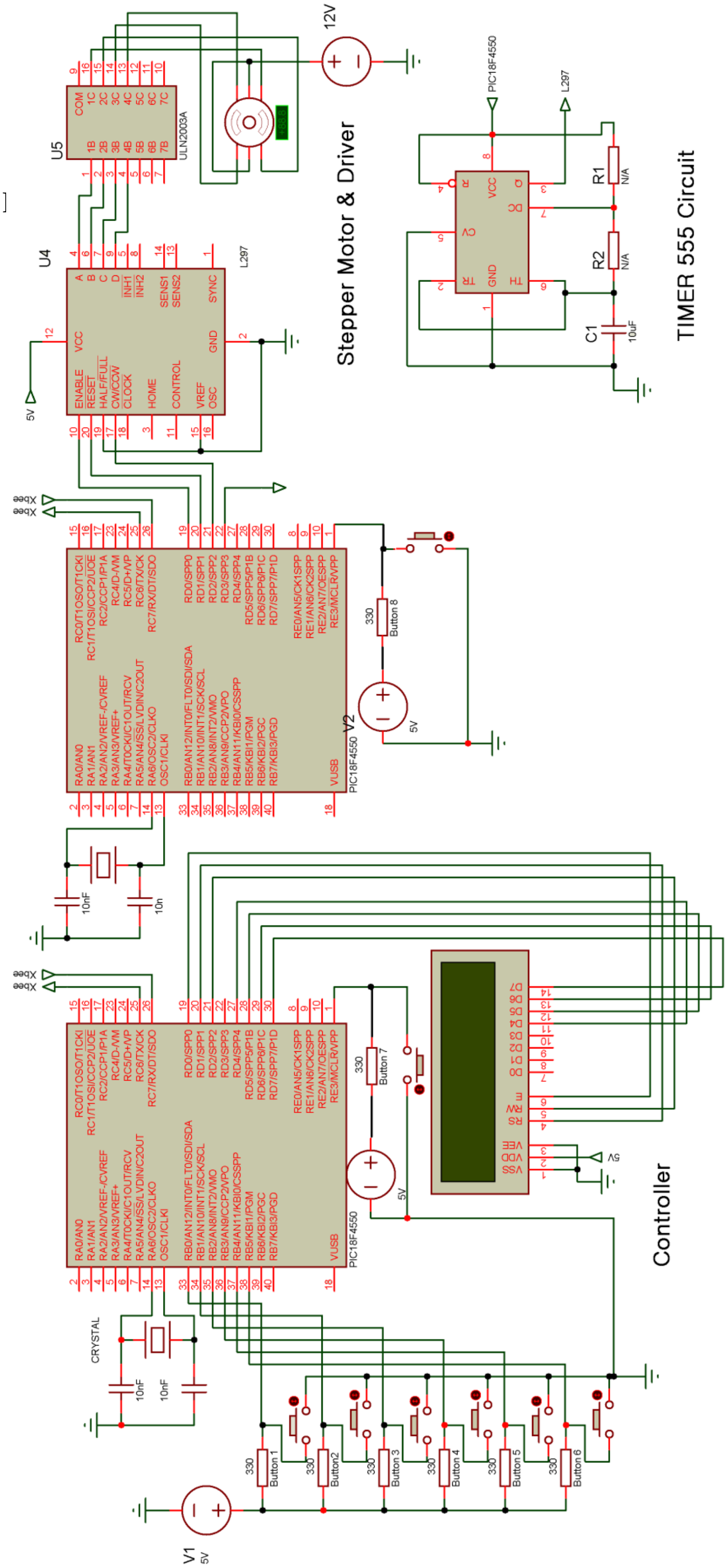

Figure 3.17: Overall Circuit Design

CHAPTER 4

RESULT AND DISCUSSION

4.0 Introduction

This chapter will going to discuss the outcome from the project. After long planned and hard work, the project had finally come to the end. The project had successfully constructed with every components integrated together in proper order. Further information and description will be discussed in several sub-chapters including Transmitting module, Receiving module and End device.

4.1 Overall Process Flow

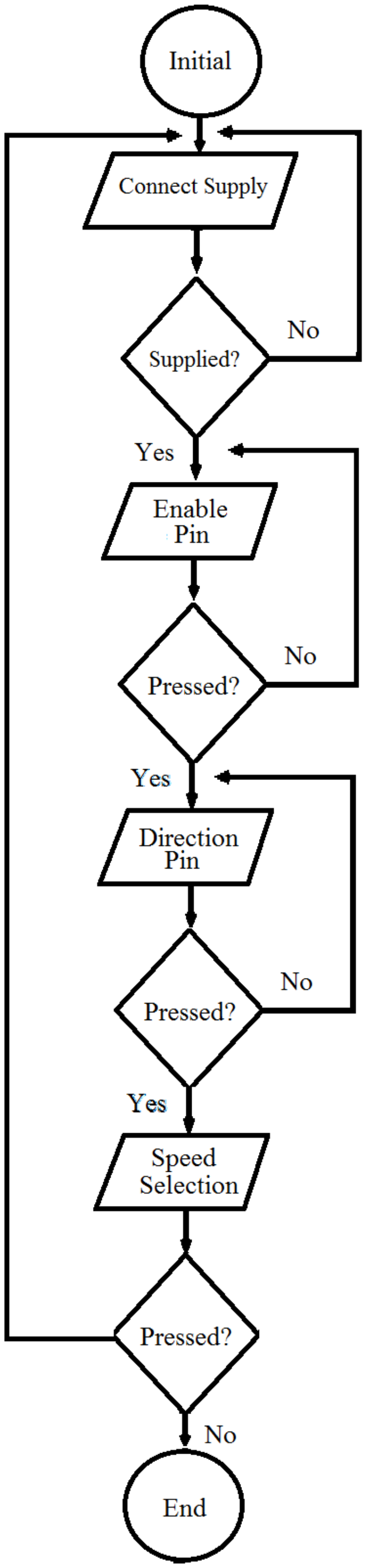

Figure 4.1: Project Process Flow.

The project starts with connecting the power supply into both transmitter and receiver as shown in Figure 4.1 above. Notice that at transmitter part, the device has an option of either supply by using a 12V DC socket or a 9V battery. Then power switch has to be turn on to supply the voltage into the devices. LCD will display “Hello Xbee!” to indicate the system was turn on. Next it will display “Xbee Initializing Please wait…” to allow the Xbee module to scan and set up the communication bridge and it will display “Controller Ready to use!”

Now user first has to press the “ON” button, first a signal will send from the controller (transmitter) to stepper motor driver (receiver) to put ENABLE pin high. LCD will indicate “Stepper Motor Energized”. Next user need to select the direction of the rotation by pressing button “FORWARD” or “REVERSE”. Initially stepper motor speed at zero, in order to initialize stepper motor to rotate user has to press button “INCREASE” to tap stepper motor speed increase by 1 level. 4 level speeds were designated in this project. “DECREASE” button press to tap stepper motor speed decrease by 1 level. The stepper motor will continue rotating at the same direction and speed upon selection, “STOP” button press to stop the stepper motor rotation.

4.2 Transmitting Module

In this project transmitting device would be the controller. The function of the controller is to detect analog input make by user and change the signal into digital input to be process by the microcontroller. The device is shown in Figure 4.2.

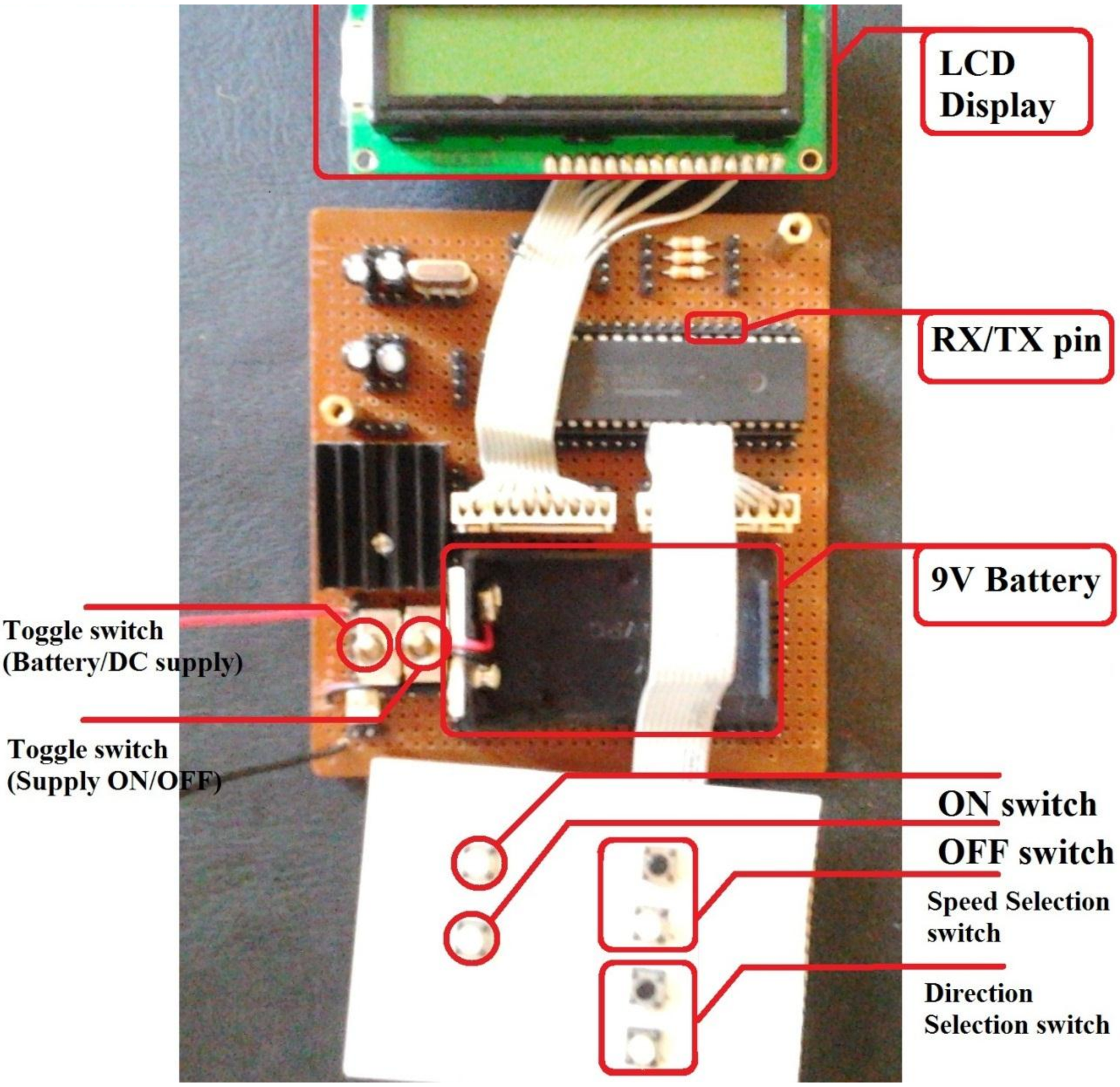

Figure 4.2: Transmitting Module.

4.2.1 LCD Display

In order to indicate what is happening in the system, a LCD 168A led display was included. The LCD will initially display as shown in Figure 4.3 when power supply turn on.

Figure 4.3: LCD Initialize Display.

When the speed and direction of the stepper motor had been set by pressing the push button the LCD will display as in Figure 4.4 and Figure 4.5 depending on the direction and speed selection. The list of code can refer in Appendix A.

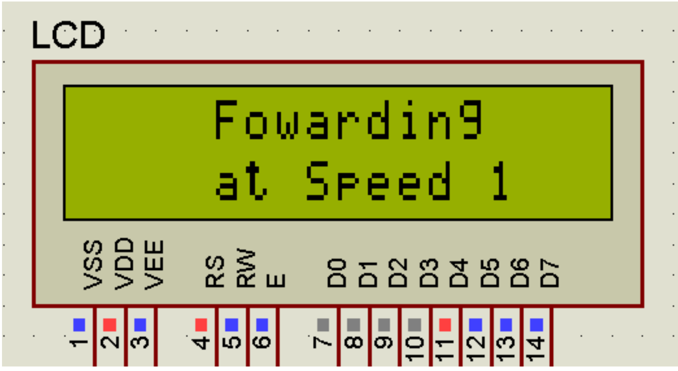

Figure 4.4: LCD Display When Stepper Motor In Speed 1 Forward Direction.

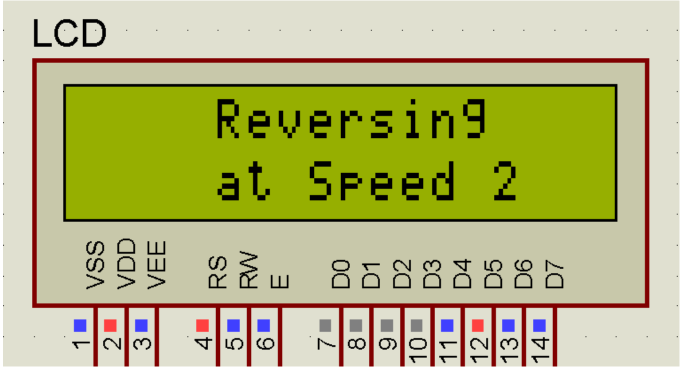

Figure 4.5: LCD Display When Stepper Motor in Speed 2 Reverse Direction.

4.2.2 Power Supply (Transmitting)

The project is working in the range of 4.5V to 6.0V DC current, where a DC adapter was needed. Voltage was then regulated by 7805 voltage regulator in order to achieve a stable and reliable 5V supply. This is part of the protection from supplying over current into the system which can lead to major failure of other devices. Since the project aim is to develop a wireless controller, in order to improve the portability of the controller a 9V battery was included as an internal supply for the system. Notice that the system run either by using the supply from a battery or from the DC socket. The circuit was successfully constructed using a two way toggle switch as shown in Figure 4.6.

Figure 4.6: Toggle Switch.

An ON/OFF supply toggle switch also added to control the incoming supply from supplying into the system. An LED was included to indicate the supply availability, where when toggle switch was switched on supplying the circuit the LED will light on.

4.2.3 Switches

Switch is device used to detect analog input made by user and convert it as a digital signal to send and interpret by microcontroller. In this project switch was constructed using the very basic of push button as shown in Figure 4.7.

Figure 4.7: Push Button.

For the controller seven switches were constructed. The list and function of each button were listed in Table 4.1.

Table 4.1: List of Button and Function.

Indication	**Function**
ON	- Send a digital input signal to turn on Stepper Motor Driver.
OFF	- Send a digital input signal to turn off Stepper Motor Driver.
FORWARD	- Send a digital input signal to set Stepper Motor rotation to forward direction.
REVERSE	- Send a digital input signal to set Stepper Motor rotation to reverse direction.
SPEED INCREASE	- Send a digital input signal to tap Stepper Motor speed increase by one level.

SPEED DECREASE	- Send a digital input signal to tap Stepper Motor speed decrease by one level.

Besides that, a RESET button also included in the circuit function as the master switch to reset the whole system.

4.2.3 Transmitting Medium

The digital inputs will be interpreted by a microcontroller and generate a data output to another microcontroller to proceed further instruction as programmed. A medium was needed to transmit this data in order to allow both microcontrollers to communicate. Since the project's objective is to control the stepper motor using ZigBee Standard, Xbee module was used as shown in Figure 4.8.

Figure 4.8: Xbee Module.

The system consists of two points only, Point-to-Point method was selected to perform the task. Notice that, Xbee module must be configured first in order to construct a bridge or communication link between both Xbee modules. The setup can be done by using special designed software, XCTU. The setting and successful interconnecting were shown in Figure 4.9 and Figure 4.10.

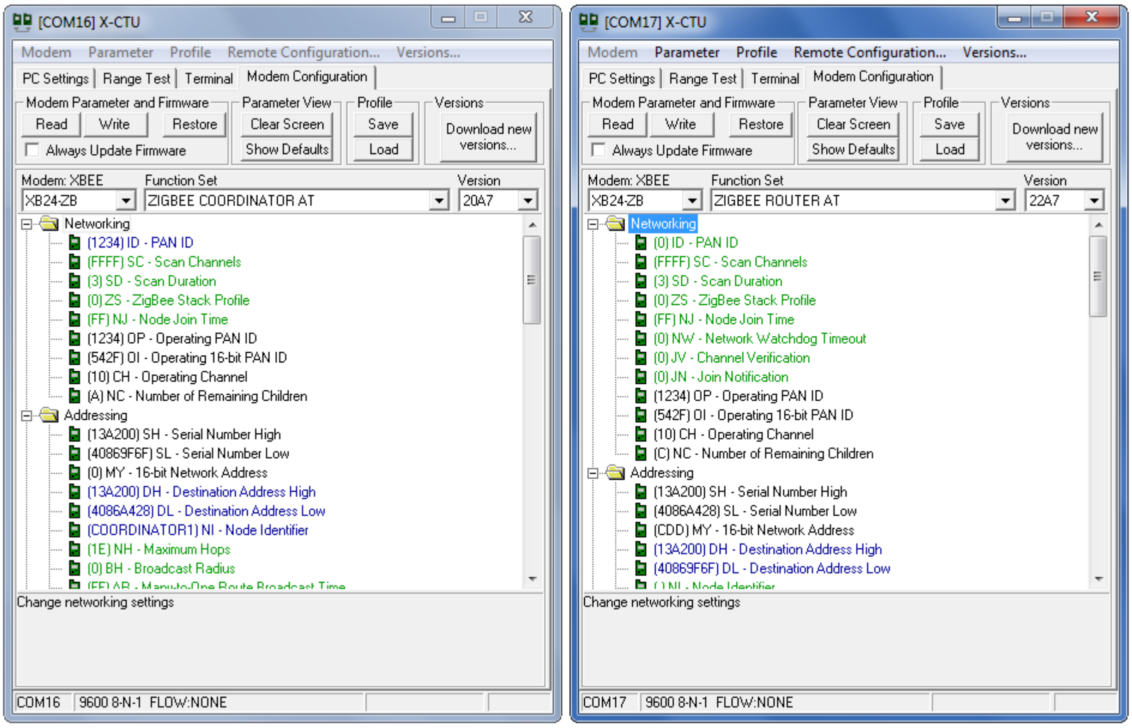

Figure 4.9: Xbee Module Configuration as in XCTU Software.

Figure 4.9 shows one of the Xbee set as COORDINATOR while another set as ROUTER, where both Xbee destination address was set in cross order. PAN ID set to configure the network availability.

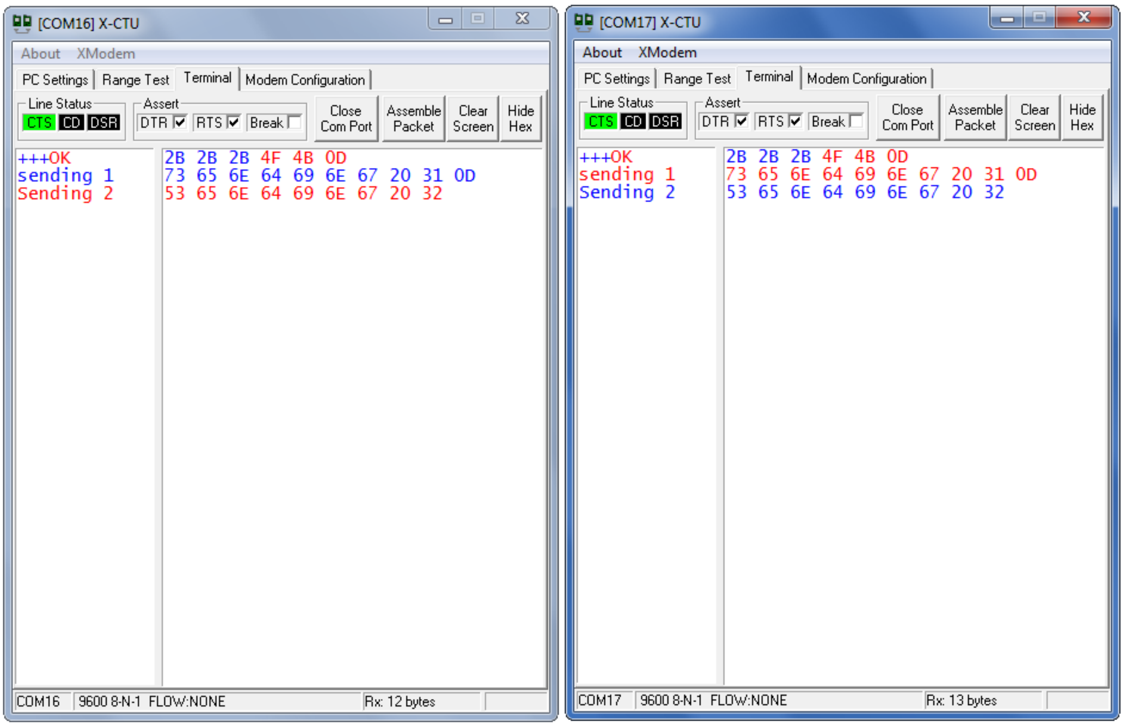

Figure 4.10: Xbee Module Communication Test as in XCTU Software.

Figure 4.10 shows both modules were successfully communicated where text was able to send and receive by each other. The RX/TX pin of both microcontroller and Xbee module were connected vice versa to send and receive data. Notice that, Xbee module function in the baud range of 9600.

4.3 Receiving Module

Receiving device is almost same like Transmitting device; both consist of a microcontroller and Xbee module works in range of 4.5V to 6.0V DC current. However unlike transmitting devices consist of many switches; receiving devices only consist of 1 RESET switch while the microcontroller was connected to stepper motor driver. In another word, microcontroller stays alert on order and directly divert it to the driver as programmed. Prototype work which had done was shown in Figure 4.11.

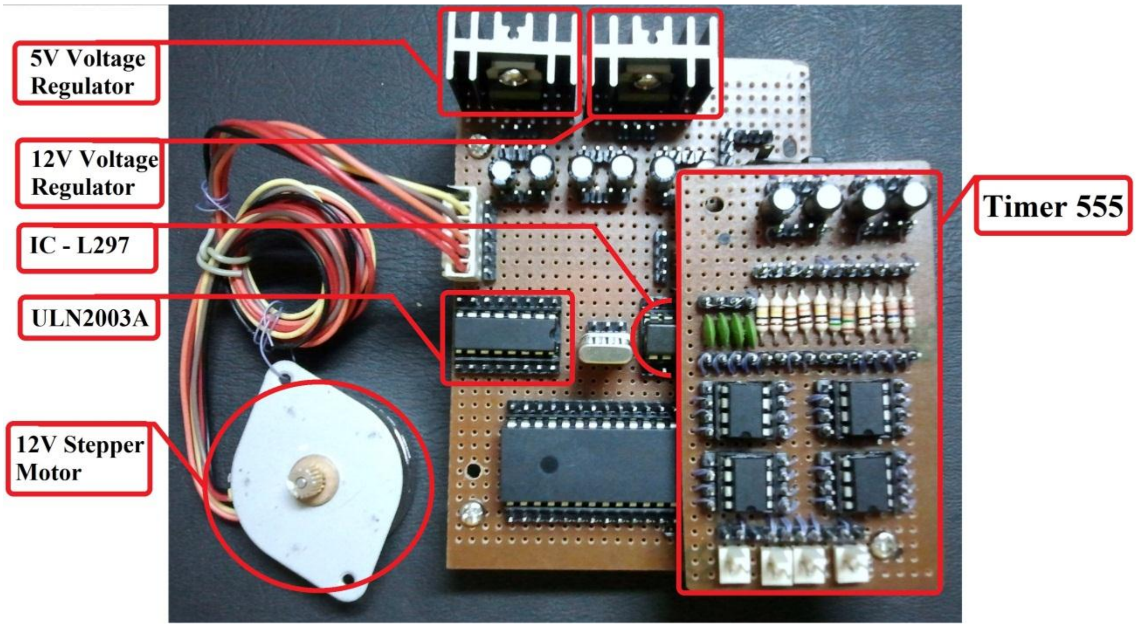

Figure 4.11: Receiving Module.

4.3.1 Power Supply (Receiving)

Unlike transmitting device, Receiving device consist of two different values of supply voltage which are 5V and 12V DC current. This is because the stepper motor was rated at 12V, a 5V supply was insufficient to turns it up. Additional 7812 voltage regulator shown in Figure 4.12 was included to produce a stable constant 12V DC current supply to drive stepper motor.

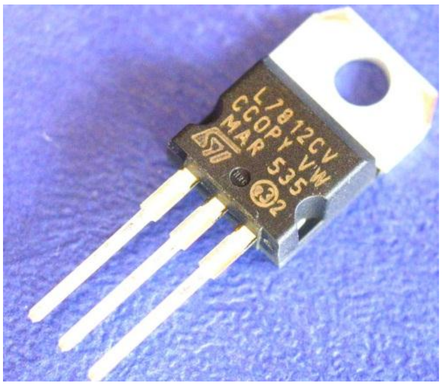

Figure 4.12: 7812 Voltage Regulator.

4.3.2 Timer 555

In order to control the stepper motor speed, a pulse wave modulation, PWM was needed in order to perform the task. Since the PWM generated by microcontroller itself was way too high to allow stepper motor work properly, PWM was manually generated using IC TIMER555. IC TIMER 555 shown in Figure 4.13 was a very useful device used to generate PWM where the frequency was able to manual control by manipulating the ratio of both resistor and capacitor value.

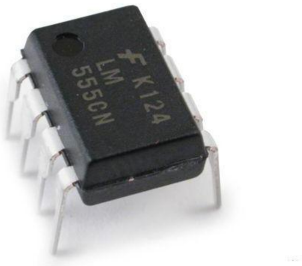

Figure 4.13: IC Timer 555.

Notice that, the frequency rate is directly proportional to the speed of stepper motor; higher the frequency faster the rotation speed. In these project four different speeds was setup up using 4 identical Timer 555. The waveform frequency generate by Timer 555 is able to calculate by formulae, please refer Appendix B.

4.3.3 L 297

IC L-297 as shown in Figure 4.14 is a very common driver use to drive stepper motor. The IC itself was able to generate a proper sequence to drive stepper motor to rotate. Besides that, the IC also able to control stepper motor rotation direction and speed by vary the clock input. In this project, three pin including ENABLE, CW/CCW and CLOCK were used where counting and fall step motion were ignored. Each and every pin will connected to microcontroller as digital input for control method.

Figure 4.14: IC L 297.

4.3.4 ULN2003A

ULN2003A act as a medium between the driver and stepper motor. Since IC L297 is unable to handle too high current, ULN2003A was placed to take up the task to drive the stepper motor. Four pins were direct connected in between IC L297 and stepper motor.

4.4 End Device

The end device of the project would be the stepper motor. Six wires were connected as shown in Figure 4.11 previously; two common 12V DC current supplies and four controlling input. A designated signal sequence will directly drive stepper motor to rotate through ULN2003A.

CHAPTER 5

CONCLUSION

5.1 Introduction

This is the final chapter of the thesis; outline of the project will be discussed comparing to the objectives that were set. The Important finding of the project will be further described below. Besides that, further suggestion work for the area of the project also included.

5.2 Conclusion

As a conclusion, the project had successfully constructed while fulfilling every objectives listed in Chapter 1. A wireless controller using Xbee module was successfully interpreted into both receiver and transmitter. Notice that, along the process of building the project; there are several important finding which direct or indirectly affect the outcome of the project. First of all, the setting set on Xbee module was unable to keep repeating sending a data. So the output was only sent at one time per single press, the receiver will only receive data at one time and the rest of action will be perform by programming coding which had preset in microcontroller. In that case, every action and output from microcontroller to stepper motor driver has to be break down into as simple as possible compartment.

The finding follows by IC ULN2003A. Although it was perfect enough to use this single IC to control stepper motor; but not in simplest form. This is because the order of sequence to drive stepper motor to spin has to be manually preset in microcontroller. In order to solve that problem, IC L297 was brought out to perform the task where it will generate the sequence automatically in proper order. The speed of changing each tap to another was decided by a CLOCK pin. A pulse was connected into this pin. The higher generated pulse supplied, the faster tap speed which directly increase stepper motor rotation speed. Instead of driving the stepper motor to rotate, IC L297 also able to control the direction of the stepper motor. Additionally it also can control stepper motor to work in half or full step and step count.

Besides that, instead of using Point-to Point method Xbee module also able to setup into Mesh and Many-to-One connection which can allow more devices to join. This would allow more devices to be able to control by a single coordinator.

5.3 Future Recommendations

Although project had successfully constructed, there are still many improvement that can be made. Recommendations and suggestions for future work were suggested in point below.

I. Improve Xbee Module Connection method

Instead of using Point-to-Point connection, Xbee also can be set up using Many-to-One connection. This would allow more devices able to control by a single controller. Device such sensor may take part as input or controlling the end device. Besides that, Xbee module contains of its own perspective language. Program can eventually manually save in Xbee itself instead of totally depend on microcontroller. Xbee connection stability and range of working also can be improved by using a high specification module, Xbee Series 2 – Pro. Notice that this device can provide working range at about two kilometer at sight.

II. Improve the Stepper motor driver

Notice that the stepper motor driver was made under a limited budget where only four speed selections were made. Design can be improve by improving the way the speed increase continuously instead of level by level tap. Upgrade stepper motor drive by fully make use of HALF/FULL and HOME pin. Notice that, IC L297 actually can control the stepper motor to step half or full step when it rotate. Besides that, HOME pin also allow it as a counter to counter the number of complete cycle had made. An output will send out every time sequence pass 0101. In that case, stepper motor can use for higher precision work which require accurate and district angle.

III. Improve Stepper Motor

The stepper motor used in this project is too normal in term of stepping angle and weight handling. If the stepper motor was changed with a higher specification one, the angle can be controlled more precisely. This is important for application which involves precise angle and distance travel or location allocating.

5.4 Commercialization

This project can be commercialized in many fields of working area such as production factory and radar telecommunication. This is because most of the work requires precise angle and displacement in which stepper motor able to provide. Besides that, the device also can be applied on most of the camera surveillance for monitoring purpose.

However controllers are usually connected in wire method. It is totally fine if the distance and numbers of control is less. When long distances with multiple controls are put into factor, it may costly for purchasing wires and difficult to install. That is where this project takes place in advantage by solving the problem with wireless connection. Notice that, each controller can be set individual address to define particular control line. As long as the device was within the working range, it can function properly with high speed of data transfer.

REFERENCES

[1] Mohd Adib B. Sarijari, Rozeha A. Rashid, Mohd Raziani Abd Rahim, Nur Hija Mahalin, "Wireless Home Security and Automation System Utilizing Zigbee Based Multi-hop Communication." IEEE 2008 6th National Conference on Telecommunication Technologies 26-27 August 2008.

[2] The Zigbee Alliance website [Online]. Available: http:// www.zigbee.org/(2009).

[3] Chae, M.J, Yoo, H.S, Kim, J.R., and Cho.M.Y., "Bridge Condition Monitoring System Using Wireless Network (CDMA and Zigbee)," 23rd International Symposium on Automation and Robotic in Construction ISARC, Tokyo, Japan 3-5 Oct 2006.

[4] Zulhani Rasin and Mohd Rizal Abdullah, "Water Quality Monitoring System Using Zigbee Based Wireless Sensor Network." International Journal of Engineering & Technology IJET Vol9 No 10 2007.

[5] Universal Plug and Play (UPnP) Device Architecture Reference Specification Version 1.0. Microsoft Corporation, June 2000.

[6] Seong H.K, Jeong S.K., hong S.P., " UPnP-ZigBee Internetworking Architecture Mirroring a Multi-hop ZigBee Network Topology." IEEE Transaction on Consumer Electronics, Vol. 55, No 3, August 2009.

[7] Nandha Kumar Thulasiraman, Haidar A.F, " A Reconfigurable Wireless Stepper Motor Controller FPGA Implementation." 2010 IEEE Symposium on Industrial Electronics and Applications, Penang, Malaysia October 3-5, 2010.

[8] Han Xi, Li Zheying, Wang Shuyinh, " Stepper Motor Subdivided Control System Based on SOPC Technology," Second International Symposium on Computational Intelligence and Design 2009.

[9] Niu Wenliang, Li Zheying, Han Xi, " DFG Model of Measurement Instrument Based on SOPC Technology," ISTM 2007, Proc 5-8 August 2007.

[10] Z.L.Kang and S.F.Qu , "A New Methodology For using Single Microprocessor To Control DC Stepper Motors" Industrial Technology, Page 543-545, December 1994.

[11] N.K. Sinha (ed), "Microprocessor-Based Control System" D. Reidel Publishing Company, 1986.

[12] D. Giuliani, M. Matassoni, M.Omologo, P.Svaizer, "Hands Free Continuous Speech Recognition in Noisy Environment Using a Four Microphone Array", Proc. ICASSP, vol.2, pp.273-276, Detroit 1995.

[13] R. Schmidt, ``Multiple emitter location and Signal parameter estimation," RADC Spectrum Estimation Workshop, pp. 243 - 258, 1979.

[14] Manoj K.M, M.M. Choudhary and Fumitoshi Matsuno, "Steering of Camera by Stepper Motor Towards Active Speaker Using Microphone Array" SICE Annual Conference, Page 19-24, August 2008.

APPENDICES

Appendix A 1: Programmed Source Code (Transmitting)

```
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//                    BASIC PIC SHORT COURSE                                  //
//                       MOTOR CONTROL                                        //
//                                                                            //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////

#include <18F4550.h>                              // PIC18F4550 HEADER FILE
#include <stdlib.h>

WATCH DOG TIMER, NO LOW VOLTAGE PROGRAMMING
#use delay (clock=20M)                            // 20 MHZ CRYSTAL

#include <lcd.c>                                  // LCD DISPLAY HEADER

////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//                    SWITCH                                                  //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////

#define but1     PIN_B0                 //Enable ON
#define but2     PIN_B1                 //Enable OFF
#define but3     PIN_B2                 //Foward Selection
#define but4     PIN_B3                 //Reverse Selection
#define but5     PIN_B4                 //Speed Increase
#define but6     PIN_B5                 //Speed Decrease



#define LCD_E     PIN_D0          // PIN E
#define LCD_RS    PIN_D1          // PIN RS
#define LCD_RW    PIN_D2          // PIN RW
#define LCD_D4    PIN_D4          // PIN D4
#define LCD_D5    PIN_D5          // PIN D5
#define LCD_D6    PIN_D6          // PIN D6
#define LCD_D7    PIN_D7          // PIN D7
```

```
void main(0)
{

////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                     PORT INITIALIZE                                    //
//                                                                        //
////////////////////////////////////////////////////////////////////////////

int counter;
set_tris_b(0xFF);              // SET ALL PORT B AS INPUT PORT
output_b(0xFF);                // RESET PORT B
lcd_init(0);                   // INITIALIZE LCD
counter = 0;
delay_ms(2000);

lcd_putc("\f");                // Clear LCD display
lcd_gotoxy(4,1);
lcd_putc("Hello Xbee");        // Initialize display
delay_ms(4000);

lcd_putc("\f");
lcd_putc("\fXBeeInitializing");
lcd_putc("\nplease wait...");
delay_ms(10000);

lcd_putc("\f");
lcd_gotoxy(4,1);
lcd_putc("Controller");
lcd_gotoxy(3,2);
lcd_putc("Ready to use!");

////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                     MAIN PROGRAM                                       //
//                                                                        //
////////////////////////////////////////////////////////////////////////////

while(REAL)                    // always repeat program below
{


///////////////on button ///////////////
if (!input(but1))              // when button on press
 {
 lcd_putc("\fStepper Motor");
 lcd_putc("\nEnergized");
 while(!input(but1))           //take action after button on is released
   {}
 delay_ms(100);                //short delay to debounce the switch
```

```
 printf("a");
 }

/////////////// off button ///////////////
else if (!input(but2))              // when button off press
 {
 lcd_putc("\fStepper Motor");
 lcd_putc("\nDenergized");
 while(!input(but2))              //take action after button off is released
   {}
 delay_ms(100);                    //short delay to debounce the switch
 printf("b");
}

/////////////// forward button ///////////////
else if (!input(but3))              // when button off press
 {
 lcd_putc("\fFORWARD");while(!input(but3))              //take action after button
off is released
   {}
 delay_ms(100);                    //short delay to debounce the switch
 printf("c");
}

 /////////////// reverse button ///////////////
 else if (!input(but4))              // when button off press
 {
 lcd_putc("\fREVERSE");
 while(!input(but4))              //take action after button off is released
   {}
 delay_ms(100);                    //short delay to debounce the switch
 printf("d");
}

///////////counter increase///////////
else if (!input(but5))             // when button forward press
 {

 while(!input(but5))               //take action after sw1 is released
   {}
 delay_ms(100);//short delay to debounce the switch
 counter++;
}

///////////counter decrease///////////
else if (!input(but6))             // when button forward press
 {

 while(!input(but6))               //take action after sw1 is released
   {}
```

```
 delay_ms(100);//short delay to debounce the switch
 counter--;
}

 ///////////////  counter speed ///////////////
else if (counter==1)
{printf("e");
lcd_putc("\nSpeed 1");}

else if (counter==2)
{printf("f");
lcd_putc("\nSpeed 2");}

else if (counter==3)
{printf("g");
lcd_putc("\nSpeed 3");}

else if (counter==4)
{printf("h");
lcd_putc("\nSpeed 4");}

 ///////////counter range////////////
else if (counter<=0)
{counter = 0;}

else if (counter>4)
{counter = 4;}     }} END
```

Appendix A 2: Programmed Source Code (Receiving)

```
////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                  BASIC PIC SHORT COURSE                                //
//                     MOTOR CONTROL                                      //
//                                                                        //
//                                                                        //
////////////////////////////////////////////////////////////////////////////

#include <18F4550.h>                          // PIC18F4550 HEADER FILE

WATCH DOG TIMER, NO LOW VOLTAGE PROGRAMMING
#use delay (clock=20M)                        // 20 MHZ CRYSTAL


////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                       SWITCH                                           //
//                                                                        //
////////////////////////////////////////////////////////////////////////////

#define on1 PIN_D0
#define on2 PIN_D1
#define on3 PIN_D2
#define on4 PIN_D3
#define on5 PIN_D4
#define on6 PIN_D5

char x;

////////////////////////////////////////////////////////////////////////////
//                                                                        //
//                       MAIN PROGRAMME                                   //
//                                                                        //
////////////////////////////////////////////////////////////////////////////

void main(0)
{
set_tris_d(0x00);                 // SET ALL PORT D AS OUTPUT PORT
output_d(0x00);                   // RESET PORT D

while(REAL)                       // always repeat program below
{
```

```
if (khbit())                    // repeat to detect data received
{x = getd(); }                   // data receive detected

if (x=='a')                     // if data received = "a"
{output_high(on5); }                // ENABLE pin high ( motor on )

else if (x=='b')                 // if data received = "b"
{output_low(on5); }                 // ENABLE pin low ( motor off )


else if (x=='c')                 // if data received = "c"
{ output_high(on6); }               // CW/CCW pin high ( rotate forward )


else if (x=='d')                 // if data received = "d"
{output_low(on6); }                  // CW/CCW pin low ( rotate reverse )


 else if (x=='e')                // if data received = "e"
{output_low(on2);                  // motor rotate at speed 1
output_low(on3);
output_low(on4);
output_high(on1);}

else if (x=='f')                 // if data received = "f"
{output_low(on1);                  // motor rotate at speed 2
output_low(on3);
output_low(on4);
output_high(on2);}

else if (x=='g')                 // if data received = "g"
{output_low(on1);                  // motor rotate at speed 3
output_low(on2);
output_low(on4);
output_high(on3);}

else if (x=='h')                 // if data received = "h"
{output_low(on1);                  // motor rotate at speed 4
output_low(on2);
output_low(on3);
output_high(on4);}

}} END
```

Appendix B1: TIMER 555 Frequency Calculations

The formulae to calculate the frequency rate of the output generation by TIMER 555 was show below.

$$T = 0.7\ x\ (R1 + 2R2)\ x\ C1$$

$$f = \frac{1}{T}$$

For Speed 1:

R1 = 10kΩ; R2 = 100kΩ; C1 = 10 μF

$T = 0.7\ x\ (10k + 2(100k))\ x$ 10μF

= 1.47s

$f = \frac{1}{1.47}$

$f = 0.68\ Hz$

For Speed 2:

R1 = 5 kΩ; R2 = 50kΩ; C1 = 10 μF

$T = 0.7\ x\ (5k + 2(50k))\ x$ 10μF

= 0.735s

$f = \frac{1}{0.735}$

$f = 1.361\ Hz$

For Speed 3:

R1 =1 kΩ; R2 = 10kΩ; C1 = 10 µF

$T = 0.7\ x\ (1k + 2(10k))\ x\ 10\mu F$

$= 0.147s$

$$f = \frac{1}{0.147}$$

$f = 6.803\ Hz$

For Speed 4:

R1 = 470Ω; R2 = 5kΩ; C1 = 10 µF

$T = 0.7\ x\ (470 + 2(5k))\ x\ 10\mu F$

$= 0.0733s$

$$f = \frac{1}{0.0733}$$

$f = 13.64\ Hz$

Appendix B2: TIMER 555 Duty Cycle Calculations

The duty cycle formulae as given use to calculate the percentage of the output signal for low period and high period.

$$Duty\ cycle = \frac{Tm}{Tm + Ts} = \frac{R1 + R2}{R1 + 2R2}$$

For Speed 1:

R1 = 10 kΩ; R2 = 100kΩ;

$$Duty\ cycle = \frac{10k + 100k}{10k + 2(100k)}$$

$Duty\ cycle =$ 0.5238 (52.38%)

For Speed 2:

R1 = 5kΩ; R2 = 50kΩ;

$$Duty\ cycle = \frac{5k + 50k}{5k + 2(50k)}$$

$Duty\ cycle =$ 0.5238 (52.38%)

For Speed 3:

R1 = 1kΩ; R2 =10kΩ;

$$Duty\ cycle = \frac{1k + 10k}{1k + 2(10k)}$$

$Duty\ cycle =$ 0.5238 (52.38%)

For Speed 4:

R1 = 0.47 kΩ; R2 =5kΩ;

$$Duty\ cycle = \frac{0.47k + 5k}{0.47k + 2(5k)}$$

$Duty\ cycle =$ 0.5224 (52.24%)