WIRELESS SENSOR NETWORK (WSN)
FOR GARDENING SYSTEM

NURUL AKMAL BINTI MOHD SAMURI
51211208400

Report submitted to Fulfill the Partial Requirements
For the Bachelor of Engineering Technology (Hons) in Data Communications
Universiti Kuala Lumpur

NOVEMBER 2011

DECLARATION

I declare that this report entitle "*Wireless Sensor Network (WSN) for Gardening System*" is the results of my own research excepts as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :………………………………………………
Name :………………………………………………
Date :………………………………………………

APPROVAL

We have examined this report and verify that it meets the program and University requirements for the Bachelor of Engineering Technology (Hons) in Data Communications.

Date: 22 November 2011

Signature:………………………………………

Supervisor's Name:…………………………….

Official Stamp

ACKNOWLEDGEMENT

First of all I would like to thank to Allah for giving me all the courage and patient towards completing this project. I would also like to send my appreciation to my project advisor Ms Nor Khairiah Binti Ibrahim for giving me all the guidance throughout this 28 Week.

Nevertheless, my great appreciation dedicated to the people's around me such as family and friend that shows their support.

Lastly, the one that stay under same project advisor, Rohaima Bt Mazelan and Nur Atiqah Bt Najmuddin and those whom involve directly or indirectly with this project, thank you so much for the support.

.

ABSTRACT

Sensor network are currently an active research area mainly due to its potential of application. A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location. This project focuses on the use of Wireless Sensor Network (WSN) for gardening system. The objective of the project is to investigate how data travel in wireless medium transmission using wireless sensor network and garden's monitoring system. This project is designed to upgrade current irrigation system for garden which from manual or timer irrigation system to an automated irrigation system by controllable temperature, soil moisture and air humidity because they are the main factors to be control in precision agriculture.

TABLE OF CONTENT

TITLE	PAGE

Title Page	**i**
Declaration	**ii**
Approval	**iii**
Acknowledgement	**iv**
Abstract	**v**
Table of Content	**vi**
List of Tables	**ix**
List of Figures	**x**
CHAPTER 1 - INTRODUCTION	**1**
1.0 Introduction	**1**
1.1 Background of study	**1**
1.2 Problem Statements	**3**
1.3 Objectives of Wireless Sensor Network for Gardening System	**3**
1.4 Scope of Project	**4**
CHAPTER 2 - LITERATURE REVIEW	**5**
2.0 Introduction	**5**
2.1 Gardening System	**5**
2.2 Irrigation System	**9**
2.3 Wireless Sensor Network	**10**
2.4 Universal Asynchronous Receiver/Transmitter (UART)	**11**
2.5 C language Programming	**12**

CHAPTER 3 - METHODOLOGY **13**
3.0 Introduction **13**
3.1 Phase 1 - Preliminary Investigation **14**
3.2 Phase 2 - Analysis **14**
3.2.1 Hardware Implementation **20**
3.2.2 Software And Database Element **30**
3.3 Phase 3 - Design **34**
3.3.1 Hardware Development Process **35**
3.3.2 Software Development Process **38**
3.4 Phase 4 - Maintenance and testing **39**

CHAPTER 4 - RESULT AND ANALYSIS **41**
4.0 Introduction **41**
4.1 Interface Soil Humidity and Air Humidity Sensor with PIC 16F877A **42**
4.2 Interface Temperature Sensor with PIC 16F877A **44**
4.3 Interface user's limit value from PIC to Relay **46**
4.4 Zigbee Range Coverage **48**

CHAPTER 5 - CONCLUSION AND RECOMMENDATION **53**
5.0 Introduction **53**
5.1 Conclusion **53**
5.2 Problem and limitation **55**
5.3 Recommendations **56**

REFERENCES **57**
Books, Journals and Manuals **57**
Internet Website **60**

APPENDICES **62-74**

LIST OF TABLES

TITLE **PAGE**

Table 3.0 : PIC 16F877A Pin Description Summary **29**

LIST OF FIGURES

TITLE	Page
Figure 2.1 : Aquaponic System	6
Figure 2.2 : Biodynamic System	6
Figure 2.3 : Square foot System	7
Figure 2.4 : Hydroponic System	7
Figure 2.5 : Raised Bed Garden System	8
Figure 2.6 : Example of Irrigation System	9
Figure 3.0 : Design phase	13
Figure 3.1: Block diagram on project development	14
Figure 3.2: Block diagram of the Transmitter (Tx)	16
Figure 3.3: Block diagram of the Receiver (Rx)	18
Figure 3.4: Xbee Module	20
Figure 3.5: LCD Display	21
Figure 3.6: LED light	23
Figure 3.7: Relay	24
Figure 3.8: Humidity Sensor (probe type)	25
Figure 3.9: NPN transistor	25
Figure 3.10: Temperature Sensor (LM35DZ)	26
Figure 3.11: Microcontroller PIC16F876A	27
Figure 3.12: Proteus Design Suite 7	30
Figure 3.13 : Simulation using Proteus	31
Figure 3.14 : PICkit 2 Programmer	32
Figure 3.15: USB ICSP PIC Programmer	33
Figure 3.16: UIC-S Socket Board	33
Figure 3.17: Hardware Development Process Flow	35
Figure 3.18: PCB layout designing flow	36
Figure 3.19: Ironing the Layout Method	38
Figure 3.20: Software Development Process	38
Figure 4.0 : AIR HUMIDITY display	43
Figure 4.1 : SOIL HUMIDITY display	43
Figure 4.2 : TEMPERATURE display	45
Figure 4.3: User’s value limit set	47

CHAPTER 1

INTRODUCTION

1.0 Introduction

In this chapter, there will be the background of the system to develop briefly discussed. The problem statement, main objective and the scope of the project are also discussed in this chapter.

1.1 Background of study

Wireless technologies especially wireless sensors and sensors network, which integrate sensor technology, wireless communication technology, embedded computing technology and distributed information management technology, has been under rapid development. The advantages of wireless transmission are significant reduction and simplification in wiring and harness, allow otherwise impossible sensor applications, such as monitoring dangerous, hazardous, unwired or remote areas and locations, allow faster deployment and installation of various types of sensors, allows MEMS sensors to be integrated with signal-conditioning and radio units to form “motes” with extremely low cost, small size and low power requirement and mobility.

Types of wireless technologies being developed range from simple IrDA that uses infrared light for short-range, point-to-point communications, to wireless personal area network (WPAN) for short range, point-to multi-point communications, such as Bluetooth and ZigBee, to mid-range, multi-hop wireless local area network (WLAN), to long-distance cellular phone systems, such as GSM/GPRS and CDMA.

Gardening system is a system of taking care of the plants and frequently focuses on irrigation system. Irrigation system is the science of artificial application of water to the land and soil. Various types of irrigation have different techniques but mostly uses and found are dripper and sprinkler. In general, the goal is to supply the entire field uniformly with water, so that each plant has the amount of water it needs, neither too much nor too little. Most irrigation that is designed for home and garden equipped with auto timer. Users can set the time for the sprinkler to apply the water. Surely, it helps users as they do not have to drag the hose around the lawn and they can leave their plants unmonitored worry free.

Instead of all of the advantages, there are some disadvantages appear on this helpful equipment such as over irrigation because of poor distribution uniformity or management wastes water, chemicals, and may lead to water pollution and under irrigation or irrigation giving only just enough water for the plant (e.g. in drip line irrigation) gives poor soil salinity control which leads to increased soil salinity with consequent build up of toxic salts on soil surface in areas with high evaporation. This requires either leaching to remove these salts and a method of drainage to carry the salts away. When using drip lines, the leaching is best done regularly at certain intervals (with only a slight excess of water), so that the salt is flushed back under the plant's roots. It surely does help green lovers a lot but still it can kill the plants too.

Based on the drawback of previous and current irrigation and automatic timer watering system, this project had been design to improve this inadequate system for better planting with more accurate planting water irrigation factor's value. With user's controller limit value, this project will improve irrigation system with accurate amount of water needed at time needed and help save water.

1.2 Problem statement

Current irrigation system is sure to help gardeners, farmers and agriculturist as they do not have to drag water hose all over the field to water the field. Instead of the advantages, there are some drawbacks of the current irrigation system that can damage the plant by over irrigation or under irrigation.

Although current irrigation system has automatic water timer and the system will turn on irrigation system at the right time as how user control the timer, it does not accurate in water distribution for plant as it does not monitor the main factors for watering plants which are temperature, air humidity and soil humidity.

Even on monsoon season or dry season, user can set the automatic timer but they do not know the real value of water needed by the plant as they do not monitor the main factor value for water irrigation needed by the plants. In that case, problems for the plants still could be under irrigation or over irrigation.

1.3 Objectives of Wireless Sensor Network for Gardening System

The main objective of this project can be summarized as follows:

1. To design the user's controller water irrigation for gardening system.
2. To fabricate and test the system designed for gardening system.
3. To monitor the quality and data rate of Zigbee protocol transmission.

1.4 Scope of Project

The Wireless Sensor Network for Gardening System is an innovation of improving current water irrigation system. The scopes of this project consist of hardware and software. The basic idea is to provide user-controller of the hardware receiver board from the transmitter board that contain sensors that will send current condition of the plant to the receiver.

The three sensor components consist of temperature sensor, soil humidity and air humidity sensor connected to Zigbee to transmit data and a Liquid Crystal Display (LCD) to display the data. The main processor that controls the whole system is PIC16F877A. User can set the limit value of every sensor from the receiver board that is connected straight to relay that can be connected to any other application. If the current condition achieved the limit value of user’s set value, the relay will be turn on.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

In this chapter, the review the critical points of current knowledge and methodological approaches used is being reviewed. This literature reviews are secondary sources, and as such, do not report any new or original experimental work.

2.1 Gardening System ([1],[2],[4],[5],[9])

Gardening system is a system of taking care of the plants including all factors in planting such as type of plant, planting system; hydroponic system, raised bed, roof top, water treatment, fertilizer, irrigation watering, type of soil; clay, sand, alkaline, acidic, and many more factors.

Figure 2.1: Aquaponic System

Aquaponics is a sustainable food production system that combines a traditional aquaculture with hydroponics (cultivating plants in water) in a symbiotic environment. The system is as review on Figure 2.1. In the aquaculture, effluents accumulate in the water, increasing toxicity for the fish. This water is led to a hydroponic system where the by-products from the aquaculture are filtered out by the plants as vital nutrients, after which the cleansed water is recirculated back to the animals. The term aquaponics is a portmanteau of the terms aquaculture and hydroponic.

Figure 2.2: Biodynamic System

Biodynamic agriculture is a method of organic farming that treats farms as unified and individual organisms, emphasizing balancing the holistic development and interrelationship of the soil, plants and animals as a self-nourishing system without external inputs insofar as this is possible given the loss of nutrients due to the export of food. The example of the system as in Figure 2.2. As in other forms of organic agriculture, artificial fertilizers and toxic pesticides and herbicides are strictly avoided.

Figure 2.3: Square foot System

Square Foot Gardening is a technique of intensive planting. A system of laying out, planting, and maintaining a productive, attractive garden in any amount of space. The garden is based on a grid of 1-foot by 1-foot squares; as in Figure 2.3, with single seeds or plants placed in carefully determined spacing. The square foot system lets you make the most of your garden space to conserve the amount of water, soil conditioners, and labor needed to produce a maximum amount of food in that space. A square foot garden takes only one-fifth the space and work of a conventional single-row garden to produce the same harvest

Figure 2.4: Hydroponic System

Hydroponics is a method of growing plants using mineral nutrient solutions, in water, without soil. The example of the system as in Figure 2.4.Terrestrial plants may be grown with their roots in the mineral nutrient solution only or in an inert medium, such as perlite, gravel, mineral wool, or coconut husk.

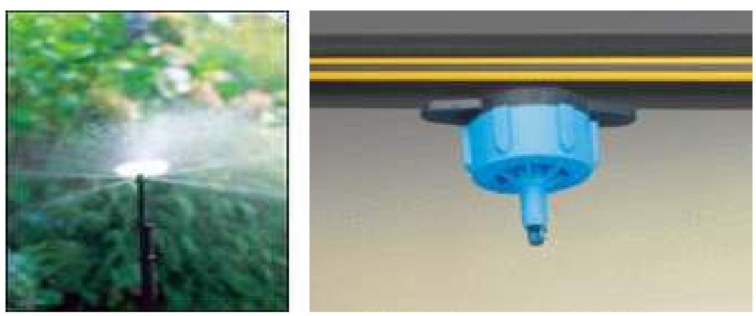
Figure 2.5: Raised Bed Garden System

Raised bed gardening is a form of gardening in which the soil is formed in 3–4 foot (1.0–1.2 m) wide beds, which can be of any length or shape. The soil is raised above the surrounding soil (6 inches to waist high), sometimes enclosed by a frame generally made of wood, rock, or concrete blocks, and enriched with compost. This design can be seen as in Figure 2.5. The vegetable plants are spaced in geometric patterns, much closer together than conventional row gardening. The spacing is such that when the vegetables are fully grown, their leaves just barely touch each other, creating a microclimate in which moisture is conserved and weed growth suppressed. Raised beds produce a variety of benefits: they extend the planting season; they reduce the need to use poor native soil; and they can reduce weeds if designed properly. Since the gardener does not walk on the raised beds, the soil is not compacted and the roots have an easier time growing. The close plant spacing and the use of compost generally result in higher yields with raised beds in comparison to conventional row gardening. Waist high raised beds enable the elderly and the sick to grow vegetables without having to bend over to tend them.

2.2 Irrigation System ([2],[3],[8])

Irrigation may be defined as the science of artificial application of water to the land or soil. It is used to assist in the growing of agricultural crops, maintenance of landscapes, and revegetation of disturbed soils in dry areas and during periods of inadequate rainfall. Additionally, irrigation also has a few other uses in crop production, which include protecting plants against frost, suppressing weed growing in grain field and helping in preventing soil consolidation.

In contrast, agriculture that relies only on direct rainfall is referred to as rain-fed or dry land farming. Irrigation systems are also used for dust suppression, disposal of sewage, and in mining. Irrigation is often studied together with drainage, which is the natural or artificial removal of surface and sub-surface water from a given area. There are many type of irrigation system available in market depending on users' desire and type of gardening system that the user uses. Figure 2.2 is the examples of some common widely use irrigation system.



Micro Jets Drip Irrigation

Figure 2.6: Examples of Irrigation System

2.3 Wireless Sensor Network ([6],[12],[13],[14],[15],[16])

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of nodes from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting.

A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes"(demo video) of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

2.4 Universal asynchronous receiver/transmitter (UART) (4,[6],[12],[13],[14])

A universal asynchronous receiver/transmitter is a type of "asynchronous receiver/transmitter", a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA RS-232, RS-422 or RS-485. The universal designation indicates that the data format and transmission speeds are configurable and that the actual electric signaling levels and methods (such as differential signaling etc.) typically are handled by a special driver circuit external to the UART.

A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. UARTs are now commonly included in microcontrollers. A dual UART, or DUART, combines two UARTs into a single chip. Many modern ICs now come with a UART that can also communicate synchronously; these devices are called USARTs (universal synchronous/asynchronous receiver/transmitter).

A UART usually contains the following components:

i. a clock generator, usually a multiple of the bit rate to allow sampling in the middle of a bit period.
ii. input and output shift registers
iii. transmit/receive control
iv. read/write control logic
v. transmit/receive buffers (optional)
vi. parallel data bus buffer (optional)
vii. First-in, first-out (FIFO) buffer memory (optional)

2.5 C language Programming (5)

C programming language as it applies to embedded microcontroller applications. This programming needed programmer to declaring variables and constant do loops, testing the logic analyzer. In c programming there are several types of loops. Next, the important part is software to interface with PIC. Software used is C programming.

C compiler is software to write, simulate and burn the program into PIC. To ensure written programming worked or not, it can combine with software Proteus. From this software, program can be simulating with circuit.

CHAPTER 3

METHODOLOGY

3.0 Introduction

The methodology part is used to define about how the project should be designed based on a few requirements. There are 6 main phase in order to complete the design requirements as in Figure 3.0. Every phase is explained in chapter 3.1,3.2,3.3 and 3.4:

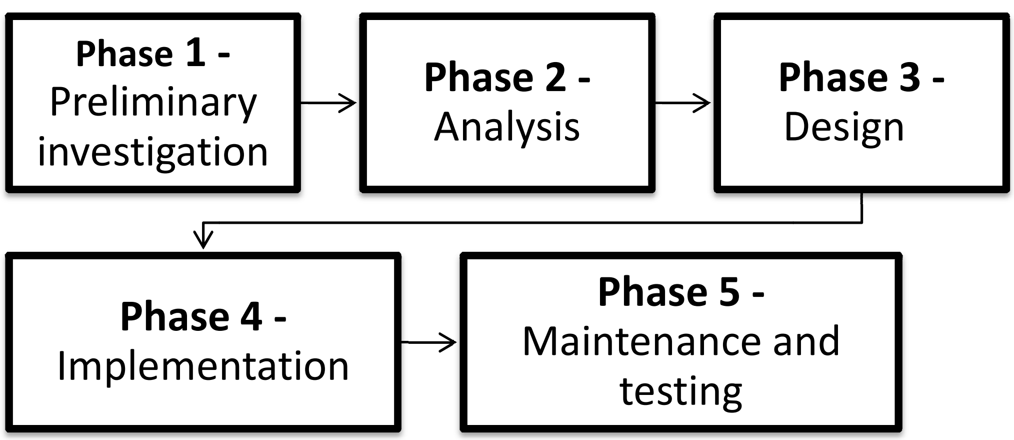


Figure 3.0: Design phase

3.1 Phase 1 - Preliminary Investigation

For the first phase, the project requirement need to be determined based on the problem statements, objectives and scope that had been determined earlier. In order to get information related to the project, the following method was done:

i. Find and investigate all documentation related this topic from books, journal, thesis, websites and others.
ii. Discuss with supervisor, specialist lecturer, technician and colleagues on improving the current system.

3.2 Phase 2 - Analysis

The system will be divide with 2 common parts consists of combination between transmitter board and receiver board as shown in Figure 3.1 below:

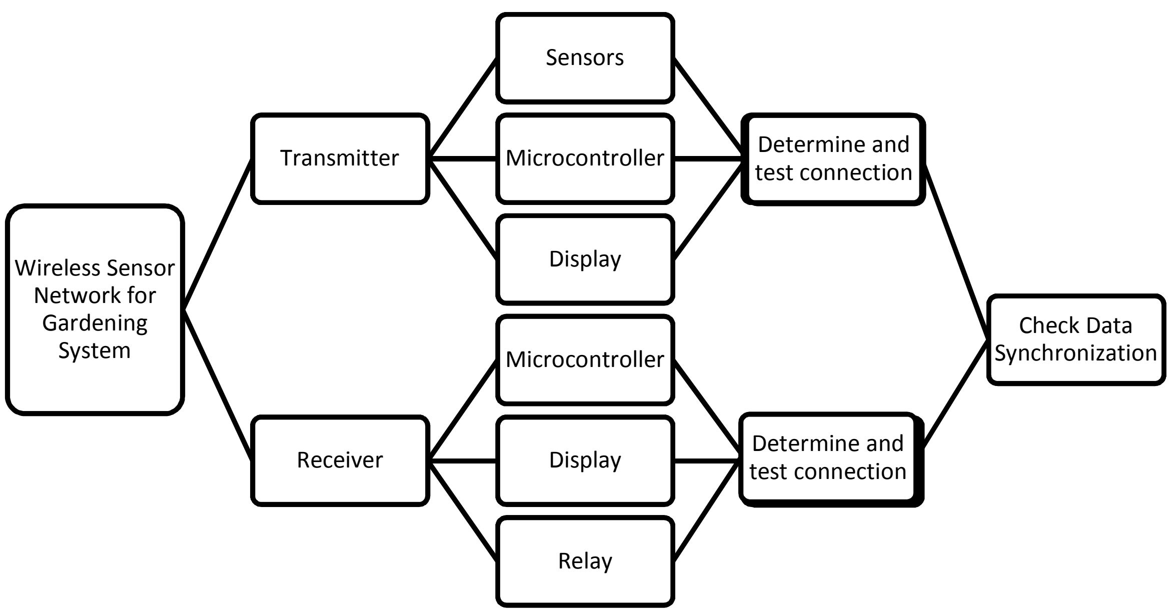


Figure 3.1: Block diagram on project development

From Figure 3.1, there are three sensors to be setup which are temperature, humidity and soil moisture. All of them will collect data and pass the data to the microcontroller that will be programmed with appropriate protocol to control data flow, to tune the data and to control the nodes in the sensors. After the data had been processed in the microcontroller, the data will then be displayed on LCD display and the data will be sending by Zigbee to the receiver board and will be display on LCD display. If the received data value exceeded user's limit value, relay will be on.

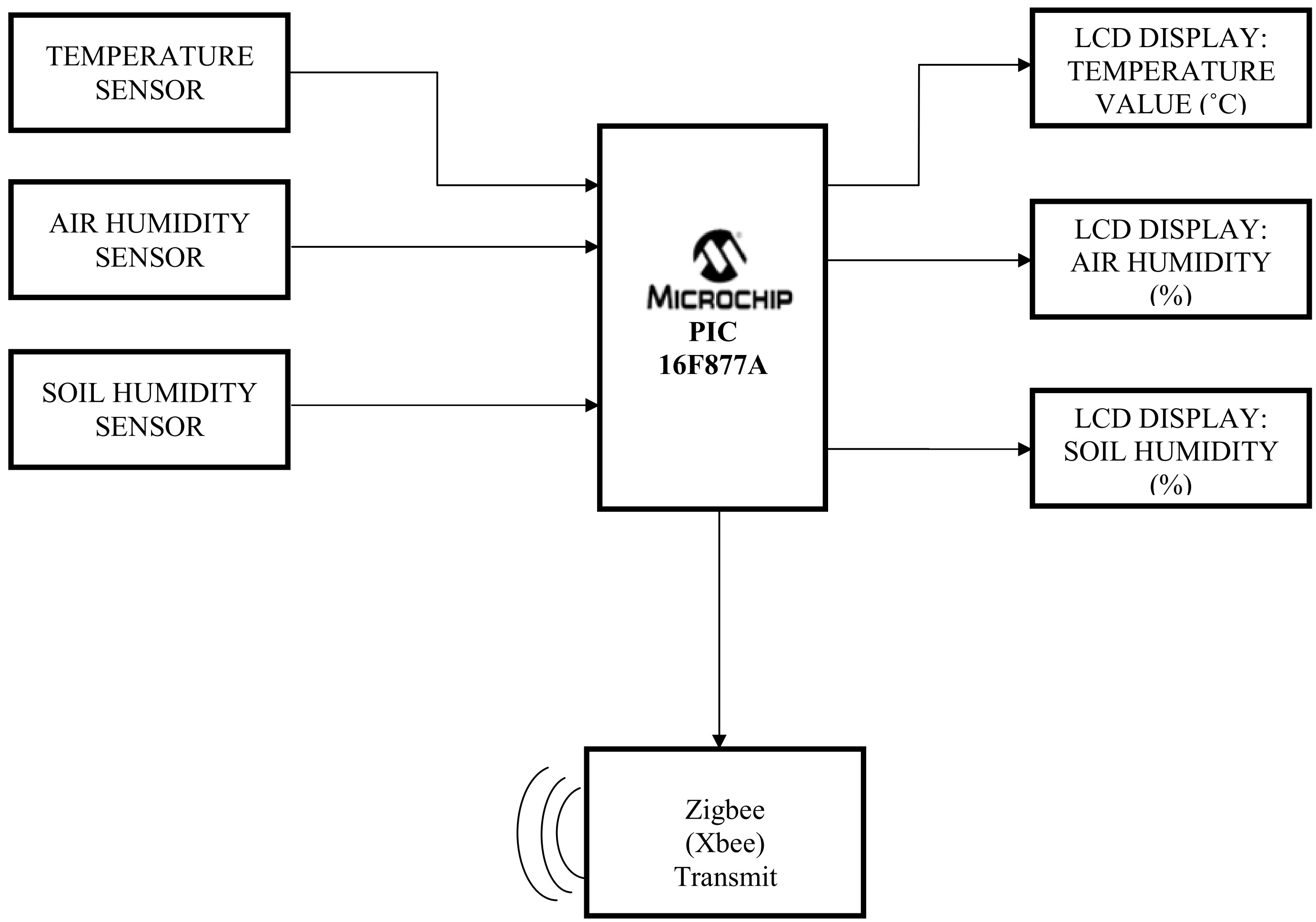


Figure 3.2: Block diagram of the Transmit (Tx)

Referring to Figure 3.2, all sensors which are temperature, air humidity and soil humidity sensors will send analog signal which is voltage to PIC 16F877A. The programmed microcontroller will then convert the voltage signal to the sensors' measurement. Voltage signal from humidity sensors will be converted into percentage humidity and voltage from temperature sensor will be converted into ˚C according to the calculation programmed in the microcontroller. The calculated value will then be displayed on LCD display and will be transmit to the receiver board.

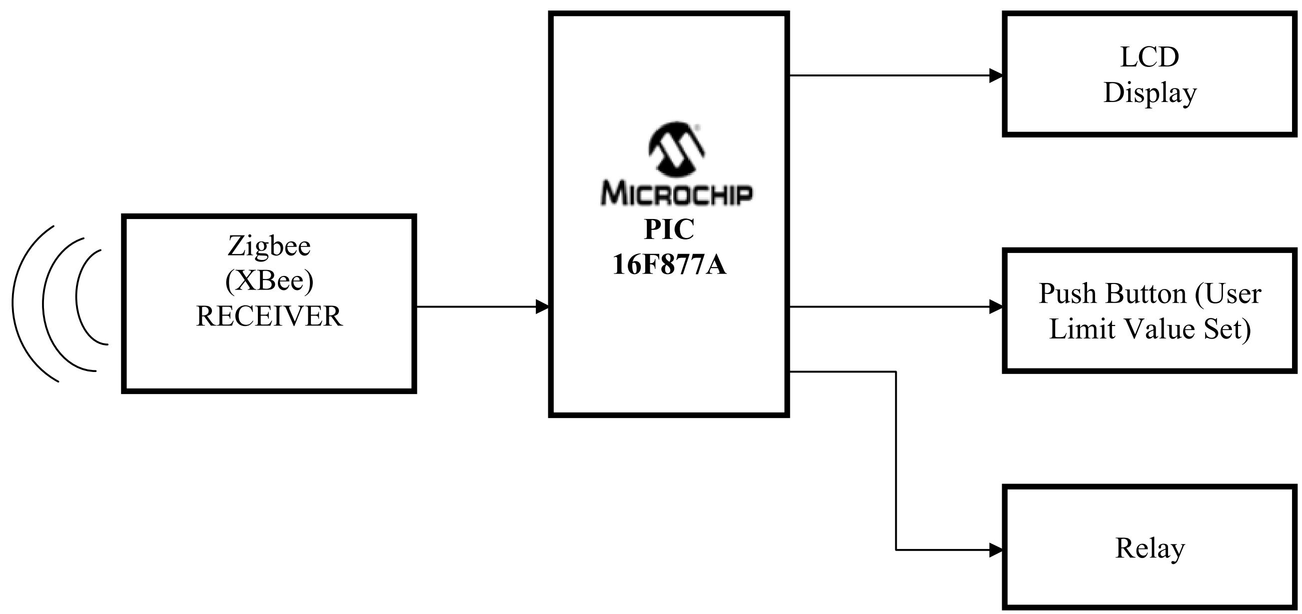


Figure 3.3: Block diagram of the Receive (Rx)

Referring to block diagram on Figure 3.3, Zigbee will receive the transmitted value from the transmitter board and will display the value on LCD display. If the received value exceeded the user limit value set, relay will then be turn on.

3.2.1 Hardware Implementation

3.2.1.1 Transmitter and Receiver Zigbee (XBee)

Figure 3.4: Xbee Module

ZigBee is a standard for wireless data transmission just like Wi-Fi and Bluetooth. The entire device (wireless device) is function to move the desired data to the next carrier in the communications path until the data is delivered to the target end-point.MaxStream is a well-known manufacturer of components for wireless communication. ZigBee is one of the MaxStream products. There are two versions of ZigBee that are available from MaxStream which are XBee and XBee PRO. Both versions are functionally identical and pin compatible. The only difference is the transmit power, which is 1mW maximum for XBee and 63mW maximum for XBee PRO.

ZigBee's mission is to cut the traditional wires between sensors, wired slaves devices, and the microcontrollers and microprocessors they serve. There is a transmit buffer and a receive buffer, and each buffer provides a temporary parking place for 100 bytes. Data can arrive from both directions at the same time – the data to be transmitted coming from the UART, and the data received by the antenna from the RF link. When the antenna is receiving data, it cannot transmit data at the same time. For this reason, the data to be transmitted is parked in the transmit buffer for a while, and the received data is stacked up in the receive buffer.

As soon as the data stream from the RF end stops, the XBee module switches the antenna from receive to transmit and empties the transmit buffer by sending its content out on the ether. At the same time, the UART empties the receive buffer by sending the data in it to your application. An application with a large amount of data to send can easily overload the transmit buffer.

MaxStream provides a 'full' alarm to deal with this problem. As soon as the application has filled all but the last 17 bytes of the transmit buffer (which means 83 bytes are waiting to be transmitted), pin 12 goes high to signal to the system that it has to stop filling the buffer for a while. Pin 12 goes low again after the content of the transmit buffer has been reduced to 66 bytes. This can be regarded as a sort of software hysteresis.

3.2.1.2 LCD Display

Figure 3.5: LCD Display

An HD44780 Character LCD is a liquid crystal display (LCD) display device designed for interfacing with embedded systems. These screens come in common configurations of 8x1, which is one row of eight characters, 16x2, and 20x4, among others. The largest commonly manufactured configuration is 40x4 characters, and actually requires two individually addressable HD44780 controllers with expansion chips.

Character LCDs can come with or without backlights, which may be LED, fluorescent, or electroluminescent. Character LCDs use a standard 14-pin interface. If the screen has a backlight, it will have 16 pins.

The backlight power rail differs often, the polarity can be different and some screens need an external resistor. Usually the supply voltage is 5V DC. Character LCDs can operate in 4-bit or 8-bit mode. In 4 bit mode, pins 7 through 10 are unused and the entire byte is sent to the screen using pins 11 through 14 by sending 4-bits (nibble) at a time.

3.2.1.3 LED

Figure 3.6: LED light

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LEDs emitted low-intensity red light, but modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness.

The LED is based on the semiconductor diode. When a diode is forward biased (switched on), electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the color of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor.

An LED is usually small in area (less than 1 mm^2), and integrated optical components are used to shape its radiation pattern and assist in reflection. LEDs present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, faster switching, and greater durability and reliability. However, they are relatively expensive and require more precise current and heat management than traditional light sources.

3.2.1.4 Relay

Figure 3.7: Relay

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

3.2.1.5 Humidity sensor

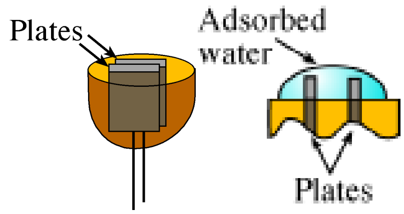


Figure 3.8: Humidity sensor (probe type)

The water sensor as in the project which based on Figure 3.8, a transistor is use to amplify the signal from the humidity (water) sensor. The humidity sensor works by electrical conductivity; when water vapor condenses on the plates it forms a thin film that conducts electricity between the two plates. For this project the sensor is the probe that will complete the circuit electric current with the present of water. In that sense the water sensor is like a resistor except that its resistance value changes depending on the amount of water in the air.

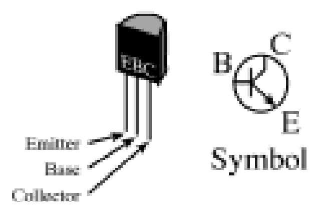


Figure 3.9: An NPN Transistor

A transistor is an amplifier that will convert the very small change in resistance from the humidity sensor into a larger resistance change that you will measure with your voltmeter. It has three connections; the collector, the base (also called the gate), and the emitter. The collector and the emitter are like the connections to a pipe that carries water, and the base is like the faucet. Ordinarily the base is closed and the resistance between the collector and emitter is very large. When a small voltage is applied to the base, it reduces the resistance between the collector and the emitter, allowing current to flow. Your kit contains a transistor which is marked with the letters E, B, and C, which stand for emitter, base, and collector, respectively.

3.2.1.6 Temperature sensor

Figure 3.10: Temperature sensor (LM35DZ)

LM35 temperature sensor is used to convert the temperature scale that captured the magnitude of voltage. This sensor has high precision source, the second leg as the output and the third leg connected to ground. The images and characteristics of LM35 IC is as follows:

i. Can be calibrated directly to the Celsius scale.
ii. Linear scale factor is more than 10mV/°C.
iii. Range of temperature between -55°C to 150°C.
iv. Work on the voltage 4 volts to 30 volts.
v. Operating current less than 60 µA.
vi. Low output impedance 0.1Ω for 1mA load.

LM35 works by changing the temperature into voltage. The LM35 can be applied easily in the same way as other integrated circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.0.1°C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 would be at an intermediate temperature between the surface temperature and the air temperature.

As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperature where condensation can occur. These devices are sometimes soldered to a small light weight heat fin, to decrease the thermal time constant and speed up the response in slowly moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

3.2.1.7 Microcontroller (PIC16F877A)

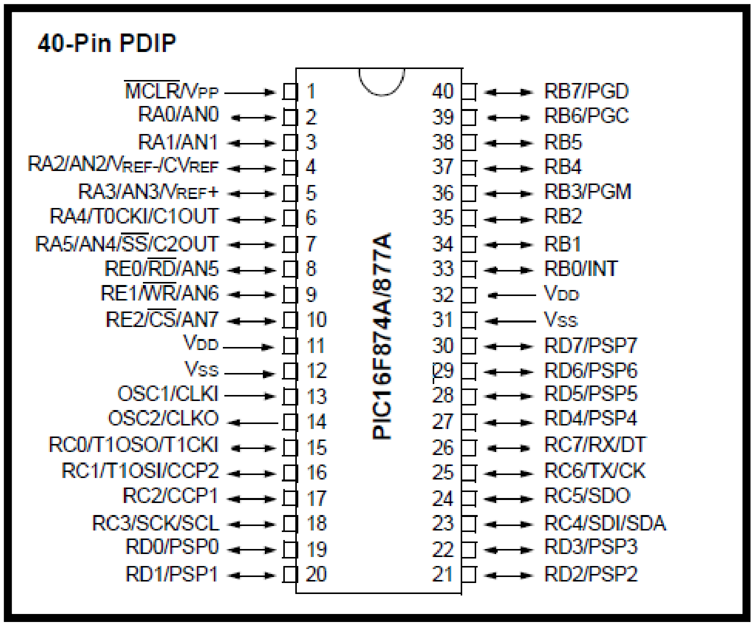


Figure 3.11: Microcontroller PIC16F876A

The Microcontroller used is PIC16F877A as in Figure 3.11. This microcontroller act as the main component which interfaces with the reader in order to save or store the data from the sensors and display on LCD. The features of PIC16F877A as follows:

i. 256 bytes of EEPROM data memory
ii. Self programming
iii. ICD (In Circuit Debugging function)
iv. 2 Comparators
v. 8 channels of 10-bit Analog-to-Digital (A/D) converter
vi. 2 capture/compare/PWM functions
vii. The synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I²C™) bus.
viii. Universal Asynchronous Receiver Transmitter (USART)

The pin used in the project are describe on Figure 3.12 as below:

Pin	Function	Description
1	Memory clear	Reset purpose. It will erase the memory of the PIC when needed without need to shut down the circuit.
2	Analog control	For variable resistor that control air humidity sensor
3	Analog control	For variable resistor that control the soil humidity sensor
5	Variable	Range sensors' control
8	Analog control	For variable resistor that control temperature sensor
11,32	Supply	5v is the maximum voltage that PIC can accept.
12,31	Ground	Must be connected to the PIC.
13,14	Clock pulse	Needs to be connected to the crystal but crystal cannot connect independently so it needs a capacitor with it. 20MHz is the maximum speed and the fastest speed that the PIC can go but it contains more power consumption.
25	Transmit	Transmit the data from PIC to XBee.
26	Receive	Receive data from XBee to PIC.
33-40	LCD Display	Connect to LCD Display

Table 3.1: PIC 16F877A Pin Description Summary

3.2.2.0 Software And Database Element

3.2.2.1 Proteus Design Suite 7

Figure 3.13: Proteus Design Suite 7

Proteus is software for microprocessor simulation, schematic capture, and printed circuit board (PCB) design. It is developed by Labcenter Electronics. The Proteus Design Suite combines schematic capture, SPICE circuit simulation, and PCB design to make a complete electronics design system.

The Proteus Design Suite includes:

- ISIS - A networking tool very similar to Eagle, but with the possibility to simulate programmable ICs like Microchip PIC, Atmel etc.
- ARES - for PCB layouts, automatic item placement and/or routing can be achieved by importing the schematic from ISIS.

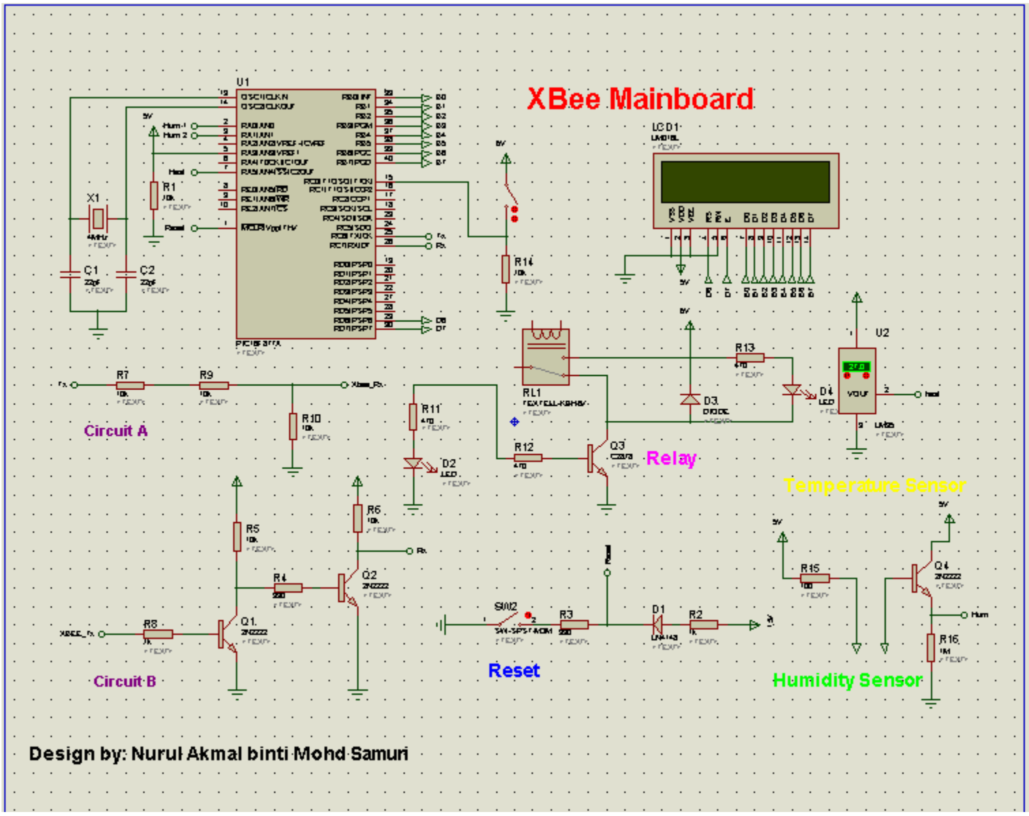


Figure 3.14: Simulation using Proteus

Proteus software is used to simulate the hardware and the software (program) to ensure that the circuit and the program is function correctly. The project simulation for WSN for Gardening System as in Figure 3.14.To simulate the Xbee module and the main board, the Xbee module is connected to the serial communication on computer and the baud rate is set same as PIC in Proteus software in order to ensure it can communicate each other.

The C program can be import or upload it to the virtual PIC in Proteus software so that we can try and see the functionality of the hardware and the program before we construct the real hardware.

3.2.2.2 PICkit 2 Programmer

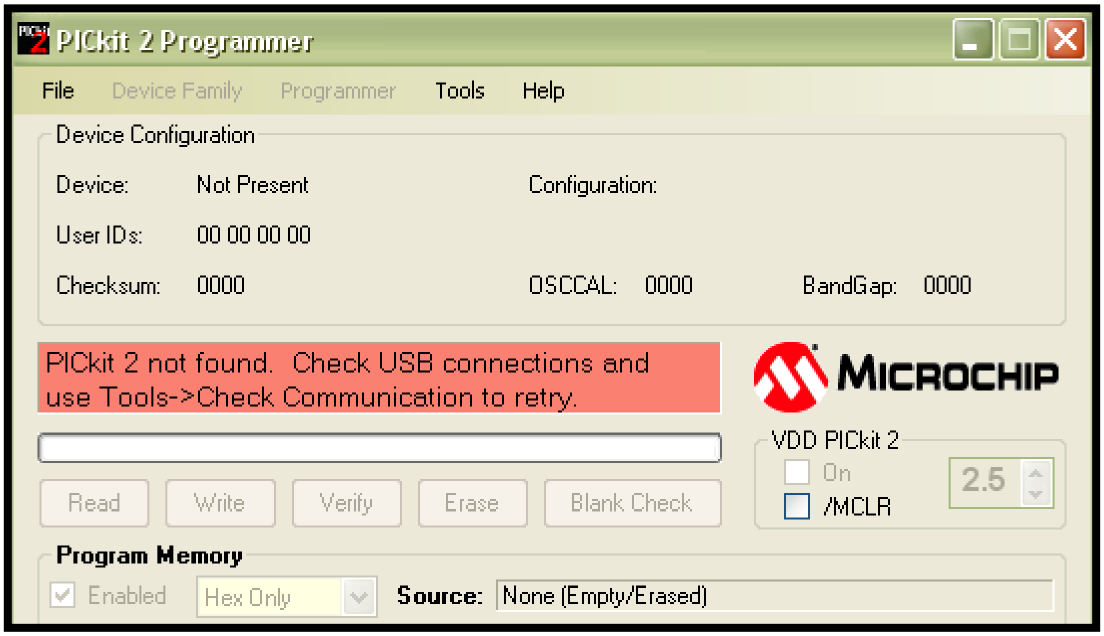


Figure 3.15: PICkit 2 Programmer

The PICkit 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products.

With Microchip powerful MPLAB Integrated Development Environment (IDE) the PICkit 2 enables in-circuit debugging on most PIC microcontrollers. In-Circuit-Debugging runs halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

3.2.2.3 USB ICSP PIC Programmer

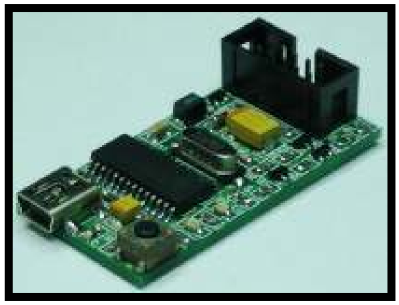


Figure 3.16: USB ICSP PIC Programmer

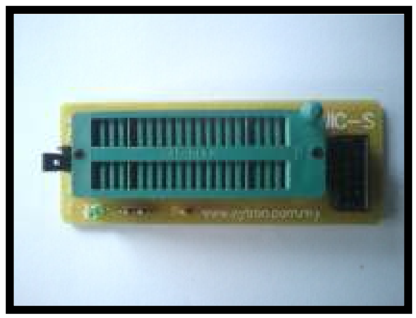


Figure 3.17: UIC-S Socket Board

UIC00A offers low cost yet reliable and user friendly PIC USB programmer solutions for developer, hobbyist and students. It is designed to program popular Flash PIC MCU which includes PIC12F, PIC16F and PIC18F family. It can also program 16bit PIC MCU. On board ICSPTM (In Circuit Serial Programming) connector offers flexible method to load program. It supports on board programming which eliminate the frustration of plug-in and plug-out of PIC MCU.

This also allow user to quickly program and debug the source code while the target PIC is on the development board. Since USB port is commonly available and widely used on Laptop and Desktop PC, UIC00A is designed to be plug and play with USB connection. This programmer obtained it power directly from USB connection, thus no external power supply is required, making it a truly portable programmer. This programmer is ideal for field and general usage. UIC00A offers reliable, high speed programming and free windows interface software.

3.3 Phase 3 - Design

Design is the planning that lays the basis for the making of every object or system. Before student start designing WSN for Gardening System, student has to state requirement of the design to be followed.

The major requirements of the receiver and transmitter system are;-

i. It should send and receive data in maintain condition for monitoring.
ii. It should have a user friendly interface.
iii. It should be durability system.
iv. It should be manual and automatic system for help industries workers.
v. It should have synchronization of data.

The user control value requirements;

i. It should be user friendly.
ii. The design must be simple and easy to understand.
iii. The circuit and the microcontroller interface system must be synchronously connected.

3.3.1 Hardware Development Process

A hardware development process is a structure imposed on the development of a hardware product. Synonyms include hardware life cycle and hardware process. PCB designing can be including as part of Hardware development. Figure 3.10 is the model that student use upon hardware completion. Each step describes how student develop the hardware. In this Hardware designing there are several software used upon completion which is Proteus 7 Professional.

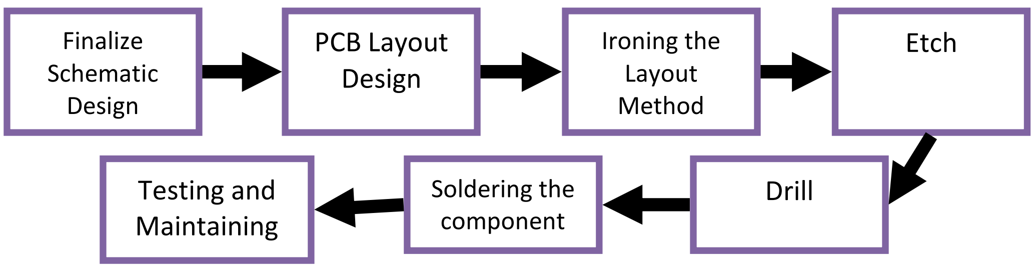


Figure 3.18: Hardware Development Process Flow

Step 1: Finalize Circuit Design

Everything starts with the circuit design. Without a circuit there is no need for a PCB. Before student even begins to lay out PCB, student must have a complete and accurate schematic diagram. In today's world of modern computing, the circuit design is capture directly in a schematic. Schematic design can be obtain from source such internet and books. Some of the schematic design student obtains from lecturer and intensive class student attend.

Step 2: PCB Layout Design

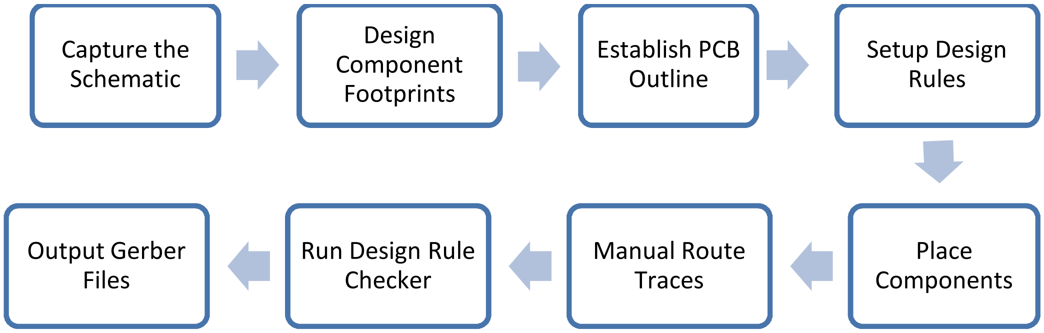


Figure 3.19: PCB layout designing flow

In general "capturing the schematic" is the process by which each component is drowns electronically and is interconnected with each other. Once the schematic is complete its time to draw the physical outline of each of the components. These outlines are what are placed on the PCB in copper to allow the components to be soldered to the printed wiring board.

i) **Establish the PCB Outline** means each project will have restrictions related to the board outline. This should be determined in this step since an idea of component count and area should be known.

ii) **Setup Design Rules** – Before placement thought you should setup the design rules to ensure that components or traces aren't too close together.

iii) **Place Components** – Now it's time to move each component onto the PCB and begin the tedious work of making all those components fit together. This is where you'll find that PCB design is really a jigsaw puzzle.

iv) **Manual Route Traces** – It's necessary to manually route critical traces. Clocks. Power. Sensitive analog traces. Once that's complete you can turn it over to Step 9.

v) **Runs Design Rule Checker** – Most PCB design software packages have a very good setup of design rule checkers. Make easy to violate PCB spacing rules and this will pinpoint the error saving from having to redesign the PCB.

vi) **Output Gerber Files** – Once the board is error free it's time to output the gerber files. These files are universal and are needed by the PCB fabrication houses to manufacture your printed circuit board.

Step 3: Ironing the Layout Method

Figure 3.20: Ironing the Layout Method

Figure 3.20 above shows the printing process to the PCB board using iron. Then the process followed by etching, drilling via, soldering the components and maintenance and testing. Figure 3.21 below shows all completed circuit of 'Receiver Circuits'.

3.3.2 Software Development Process

In this sub topic, student will discuss the flow of software upon its completion. Software development is a structure imposed on the development of a software product. Synonyms include software life cycle and software process. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process.

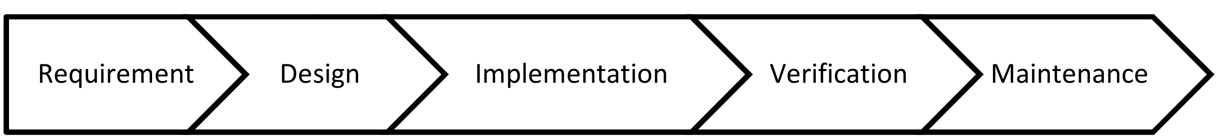


Figure 3.21: Software Development Process

Figure 3.21 show the process flow of developing software in general method. Requirement is the early part that can be consider as planning. Good planning can lead to a good result. Design is the part where student try to develop general process of the system for example by developing a flow chart and list all possible input output (I/O) related.

Implementation is the part of the process where student actually program the code for the project. Verification is also called Software testing is an integral and important part of the software development process. This part of the process ensures that defects are recognized as early as possible.

Maintaining and enhancing software to cope with newly discovered problems or new requirements can take far more time than the initial development of the software. It may be necessary to add code that does not fit the original design to correct an unforeseen problem or it may be that a customer is requesting more functionality and code can be added to accommodate their requests.

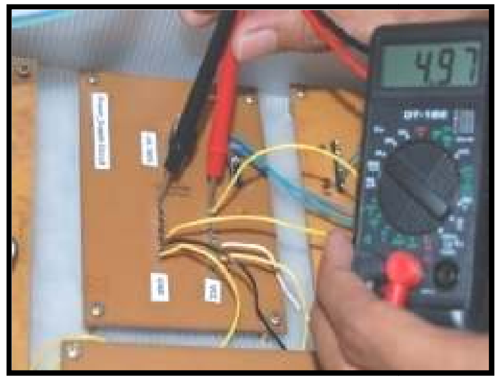
3.4 Phase 4 - Maintenance and Testing

The last process involved on testing and check the availability of the overall system include the reader, architecture design, and the performance of the whole database system to response when required and fulfill the objective requirement.

In this sub topic, student explains the activity that has been covered for hardware part and will be more detail for software part in Chapter 4. Overall, this system function correctly as it three level sensor transmitted through wireless RF and monitoring with Visual Basic (GUI) and record total level condition.

1) Testing Power supply

This is to test the supply circuit provide 5V from after passing through voltage regulator.



2) Testing Main circuit

This is to test the functionality of PIC to response to a simple program. For example, by displaying data.

CHAPTER 4

RESULTS AND DISCUSSION

4.0 Introduction

This chapter will define the final result of the project from starting of the block diagram to the end. There will be 4 parts of testing and maintaining the system which is;

1. Interface Soil Humidity and Air Humidity Sensor with PIC 16F877A.
2. Interface Temperature Sensor with PIC 16F877A.
3. Interface user’s limit value from PIC to Relay.
4. Zigbee Range Coverage.

4.1 Interface Soil Humidity and Air Humidity Sensor with PIC 16F877A

This testing objective is to check the string that humidity sensor will be sending to microcontroller (PIC). The analog signal received by the sensor will be times 0.1 for percentage humidity convert to be display on LCD.

```
switch(mode){
case 0:{
        value[0]=(int)(0.1*read_a2d(0));                //Analog Signal to humidity
        value[1]=(int)(0.1*read_a2d(1));
        value[2]=(int)(0.488*read_a2d(5));
        uart_transmit(0xF1);
        uart_transmit(set);
        uart_transmit((value[0]>>7)&0b01111111);
        uart_transmit(value[0]&0b01111111);
        uart_transmit((value[1]>>7)&0b01111111);
        uart_transmit(value[1]&0b01111111);
        uart_transmit((value[2]>>7)&0b01111111);
        uart_transmit(value[2]&0b01111111);
break;}
```

```
    case 0:{
          if(mode==1) set=rf[1];

          lcd_goto(0x00);
          lcd_string("  AIR HUMIDITY  "); //LCD display AIR HUMIDITY
          lcd_goto(0x40);
          lcd_string("      ");      //LCD display value
          convert(value[0],10);
          for(i=3;i>=1;i--) display(di[i]);
          lcd_string("%      ");

          switch(pb){
          case 2:{
                set=3;
                __delay_ms(200);
          break;}}
    break;}
    case 1:{
          if(mode==1) set=rf[1];

          lcd_goto(0x00);
          lcd_string("  SOIL HUMIDITY ");//LCD display SOIL HUMIDITY
          lcd_goto(0x40);
          lcd_string("      ");      //LCD display value
          convert(value[1],10);
          for(i=3;i>=1;i--) display(di[i]);
          lcd_string("%      ");

          switch(pb){
          case 2:{
                set=3;
                __delay_ms(200);
          break;}}
    break;}
```

Result

As the interfacing the value from sensor and display on LCD, the result shown is displayed on LCD.

Figure 4.0: AIR HUMIDITY display

Figure 4.1: SOIL HUMIDITY display

Figure 4.0 and 4.1 shows the display on LCD along with the sensor valued sense from the water sensor probes. As both sensors sense humidity value, they were built with the same schematic except they are being applied on different elements.

4.2 Interface Temperature Sensor with PIC 16F877A.

This testing objective is to check the string that temperature sensor will be sending to microcontroller (PIC). The analog signal received by the sensor will be times 0.4888 for degree Celsius convert to be display on LCD.

```
switch(mode){
case 0:{
        value[0]=(int)(0.1*read_a2d(0));
        value[1]=(int)(0.1*read_a2d(1));
        value[2]=(int)(0.488*read_a2d(5));          //Analog signal convert to temperature
        uart_transmit(0xF1);
        uart_transmit(set);
        uart_transmit((value[0]>>7)&0b01111111);
        uart_transmit(value[0]&0b01111111);
        uart_transmit((value[1]>>7)&0b01111111);
        uart_transmit(value[1]&0b01111111);
        uart_transmit((value[2]>>7)&0b01111111);
        uart_transmit(value[2]&0b01111111);
break;}

case 2:{
        if(mode==1) set=rf[1];

        lcd_goto(0x00);
        lcd_string("  TEMPERATURE   "); //LCD display TEMPERATURE
        lcd_goto(0x40);
        lcd_string("     ");       //LCD display value
        convert(value[2],10);
        for(i=3;i>=1;i--) display(di[i]);
        lcd_write(0);
        lcd_string("C      ");

        switch(pb){
        case 2:{
                set=3;
                __delay_ms(200);
        break;}}
break;}
```

Result

From the interface program that is burned to the PIC 16F877A, the output to the LCD display is as shown.

Figure 4.2: TEMPERATURE display

From Figure 4.2, the program on microcontroller display TEMPERATURE and its value based on the program burned into the PIC. The value from temperature sensor used which is LM35DZ which is an analog signal of output voltage has been converted into degree Celsius as shown in the program.

4.3 Interface user's limit value from PIC to Relay

This testing objective is to set the user limit value to microcontroller (PIC).

```
case 3:{
        lcd_goto(0x00);
        lcd_string("SET AIR HUMIDITY");
        lcd_goto(0x40);
        lcd_string("      ");
        convert(limit[0],10);
        for(i=3;i>=1;i--) display(di[i]);
        lcd_string("%      ");

        switch(pb){
        case 2:{
                set++;
                __delay_ms(200);
        break;}
        case 4:{
                if(limit[0]<1000) limit[0]=limit[0]+10;
                __delay_ms(200);
        break;}
        case 1:{
                if(limit[0]>0) limit[0]=limit[0]-10;
                __delay_ms(200);
        break;}}
break;}
case 4:{
        lcd_goto(0x00);
        lcd_string("SET SOIL HUMID  ");
        lcd_goto(0x40);
        lcd_string("      ");
        convert(limit[1],10);
        for(i=3;i>=1;i--) display(di[i]);
        lcd_string("%      ");

        switch(pb){
        case 2:{
                set++;
                __delay_ms(200);
        break;}
        case 4:{
                if(limit[1]<1000) limit[1]=limit[1]+10;
                __delay_ms(200);
        break;}
        case 1:{
                if(limit[1]>0) limit[1]=limit[1]-10;
                __delay_ms(200);
        break;}}
break;}
case 5:{
        lcd_goto(0x00);
        lcd_string("SET TEMPERATURE ");
        lcd_goto(0x40);
        lcd_string("      ");
        convert(limit[2],10);
        for(i=3;i>=1;i--) display(di[i]);
        lcd_string("%      ");

        switch(pb){
        case 2:{
                set++;
                __delay_ms(200);
        break;}
        case 4:{
                if(limit[2]<1000) limit[2]=limit[2]+10;
                __delay_ms(200);
        break;}
        case 1:{
                if(limit[2]>0) limit[2]=limit[2]-10;
                __delay_ms(200);
        break;}}
break;}
```

Result

The result for the user set limit value can be change according to user's desire. User can increase and decrease value of set limit according to the value on LCD display.

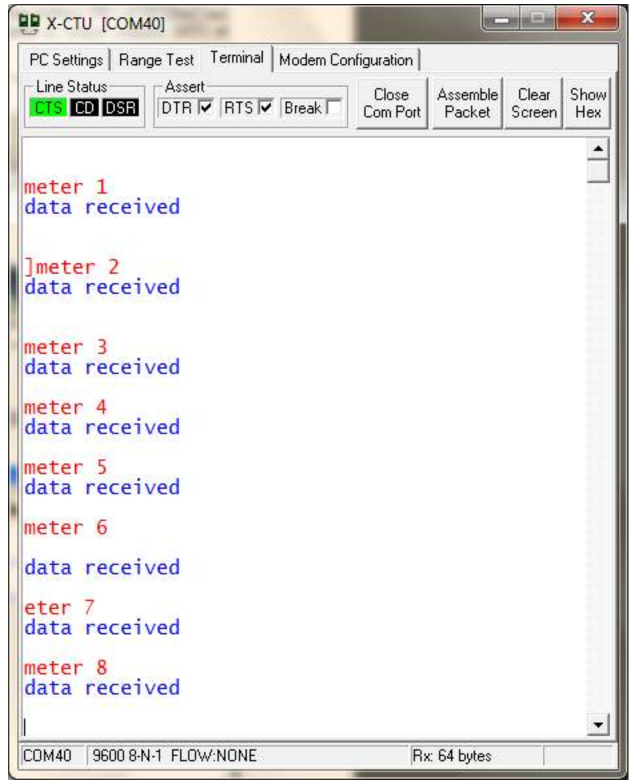
Figure 4.3: User's value limit set

From Figure 4.3, the display on LCD shows set value based on the sensors used. User can set value based on their desire with increasing and decreasing value.

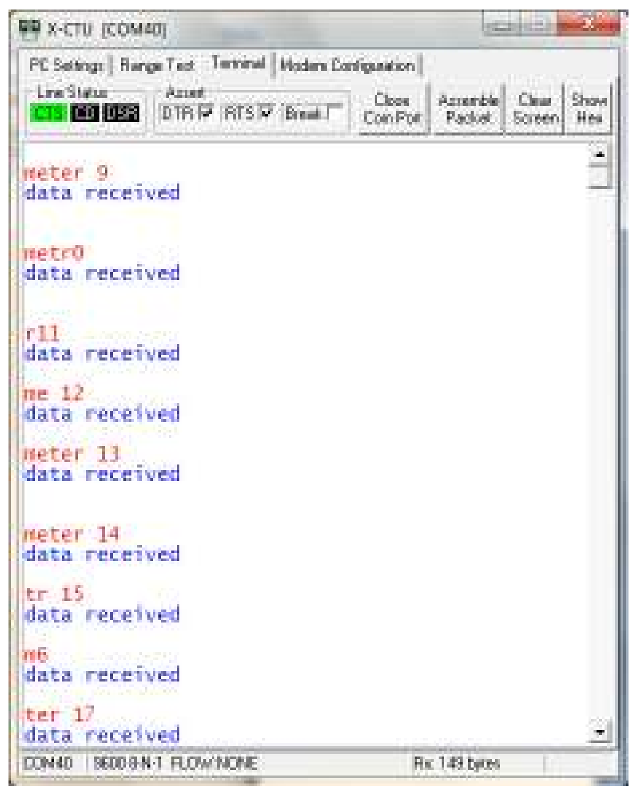
4.4 Zigbee Range Coverage

The Zigbee range coverage testing has been done in order to observe how efficiency Zigbee act as a transceiver which a Zigbee module can transmit and receive data. This testing has been done using 2 Zigbee's module. One module act as a transmitter (pink frame) and another one act as a receiver (white frame). When the transmitter sends the data, the receiver will acknowledge the transmitter which makes both modules as a transceiver to see how Zigbee communicate with each other. The testing of both Zigbee is in line of sight. The result is as shown below.

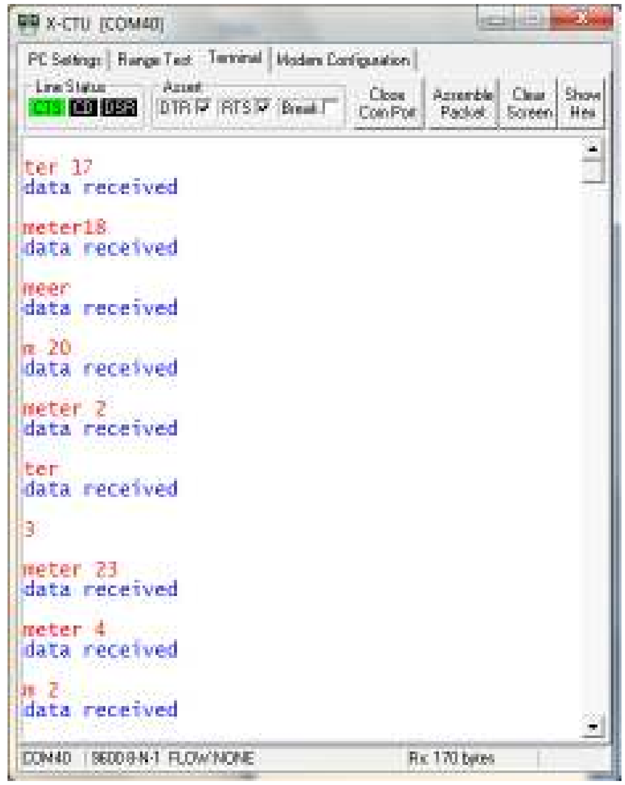

X-CTU [COM24]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS CD DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
]
meter 2
data received
meter 3
data received
meter 4
data received
meter 5
data received
meter 6
data received
meter 7
data received
meter 8
data reeived
COM24
9600 8-N-1 FLOW:NONE
Rx: 170 bytes


X-CTU [COM40]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS CD DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
meter 1
data received
]meter 2
data received
meter 3
data received
meter 4
data received
meter 5
data received
meter 6
data received
eter 7
data received
meter 8
data received
COM40
9600 8-N-1 FLOW:NONE
Rx: 64 bytes


X-CTU [COM24]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS CD DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
data received
meter 7
data received
meter 8
data reeived
meter 9
data received
meter 10
data received
meter 11
meter 11
data received
meter 12
daa recei
meter 13
data received
COM24
9600 8-N-1 FLOW:NONE
Rx: 242 bytes

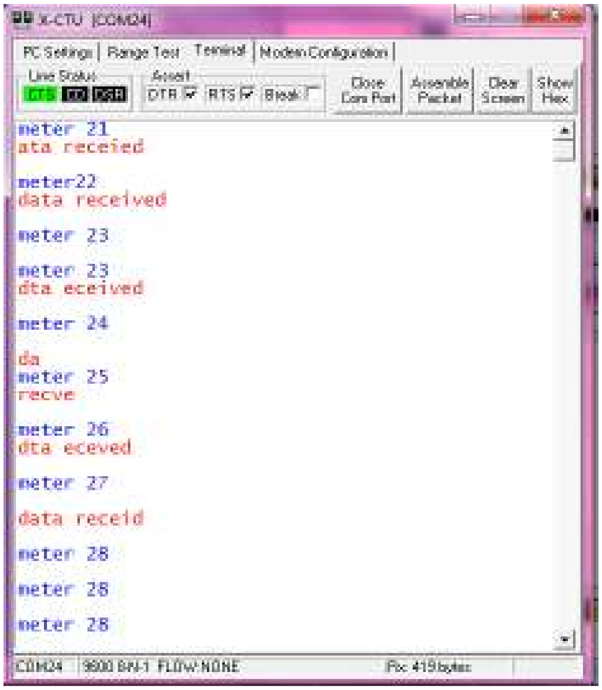

X-CTU [COM40]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS CD DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
meter 9
data received
metr0
data received
r11
data received
me 12
data received
meter 13
data received
meter 14
data received
tr 15
data received
m6
data received
ter 17
data received
COM40
9600 8-N-1 FLOW:NONE
Rx: 149 bytes

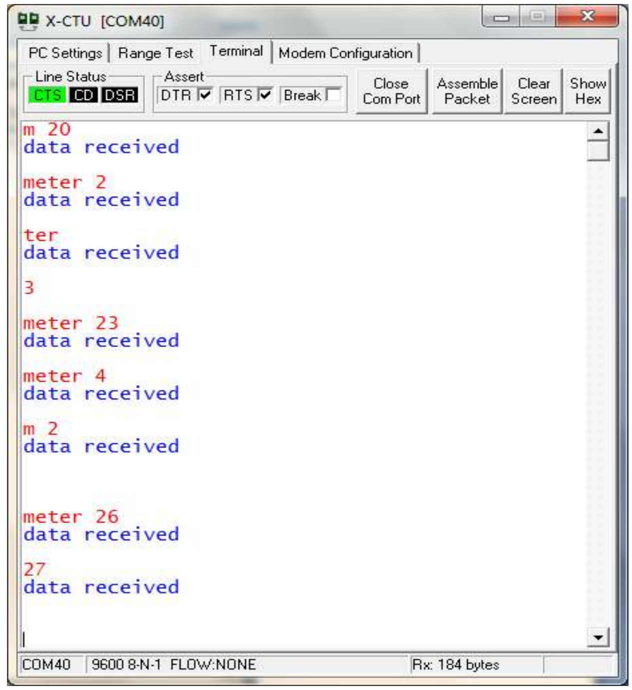

X-CTU [COM24]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS
CD
DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
daa recei
meter 13
data received
meter 14
data received
meter 15
data received
meter 16
]
meter 16
data received
meter 17
meter 17
data rec
meter 18
data received
meter 19
data received
COM24 9600 8-N-1 FLOW:NONE
Rx: 328 bytes

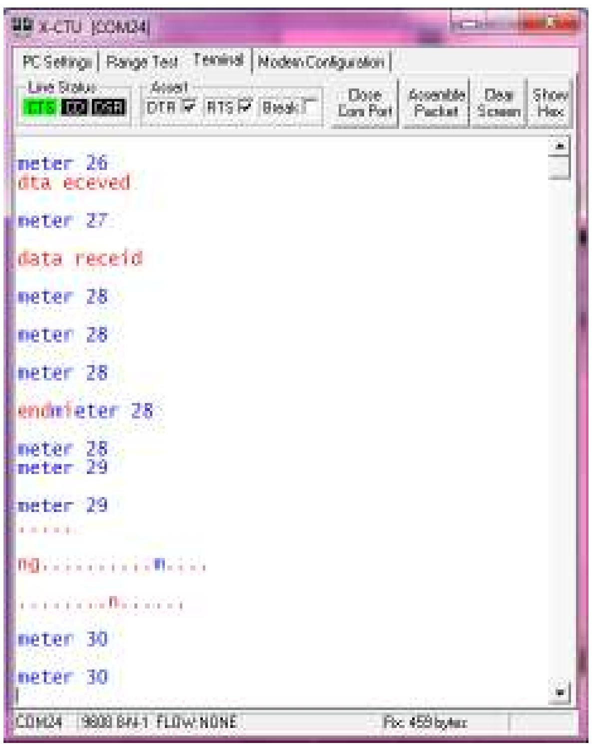

X-CTU [COM40]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS
CD
DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
ter 17
data received
meter18
data received
meer
data received
m 20
data received
meter 2
data received
ter
data received
3
meter 23
data received
meter 4
data received
m 2
data received
COM40 9600 8-N-1 FLOW:NONE
Rx: 170 bytes

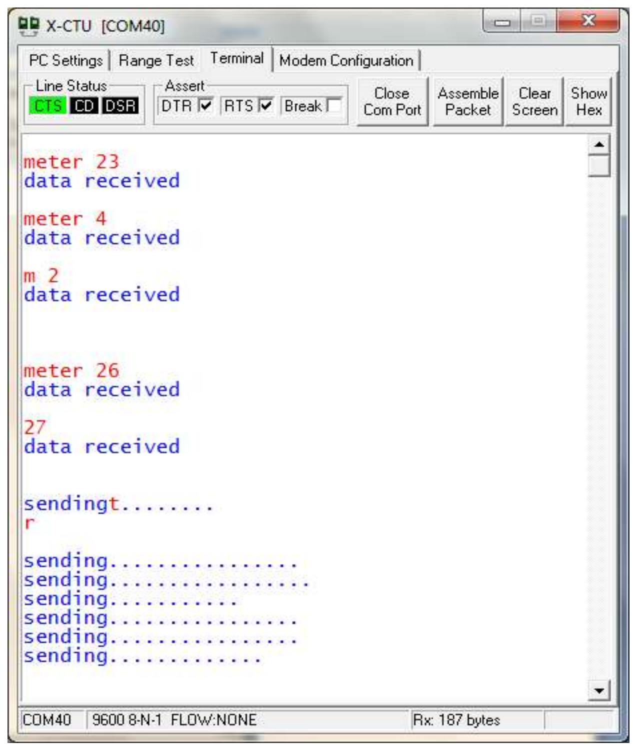

X-CTU [COM24]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS
CD
DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
meter 17
meter 17
data rec
meter 18
data received
meter 19
data received
meter 20
data received
meter 21
ata receied
meter22
data received
meter 23
meter 23
dta eceived
meter 24
da
meter 25
recve
COM24 9600 8-N-1 FLOW:NONE
Rx: 394 bytes


X-CTU [COM24]
PC Settings
Range Test
Terminal
Modem Configuration
Line Status
CTS
CD
DSR
Assert
DTR
RTS
Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
meter 21
ata receied
meter22
data received
meter 23
meter 23
dta eceived
meter 24
da
meter 25
recve
meter 26
dta eceved
meter 27
data receid
meter 28
meter 28
meter 28
COM24 9600 8-N-1 FLOW:NONE
Rx: 419 bytes


X-CTU [COM24]
PC Settings | Range Test | Terminal | Modem Configuration
Line Status
CTS CD DSR
Assert
DTR RTS Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
meter 21
ata receied
meter22
data received
meter 23
meter 23
dta eceived
meter 24
da
meter 25
recve
meter 26
dta eceved
meter 27
data receid
meter 28
meter 28
meter 28
COM24 9600 8-N-1 FLOW:NONE
Rx: 419 bytes


X-CTU [COM40]
PC Settings | Range Test | Terminal | Modem Configuration
Line Status
CTS CD DSR
Assert
DTR RTS Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
m 20
data received
meter 2
data received
ter
data received
3
meter 23
data received
meter 4
data received
m 2
data received
meter 26
data received
27
data received
COM40 9600 8-N-1 FLOW:NONE
Rx: 184 bytes


X-CTU [COM24]
PC Settings | Range Test | Terminal | Modem Configuration
Line Status
CTS CD DSR
Assert
DTR RTS Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
meter 26
dta eceved
meter 27
data receid
meter 28
meter 28
meter 28
endmeter 28
meter 28
meter 29
meter 29
meter 30
meter 30
COM24 9600 8-N-1 FLOW:NONE
Rx: 459 bytes


X-CTU [COM40]
PC Settings | Range Test | Terminal | Modem Configuration
Line Status
CTS CD DSR
Assert
DTR RTS Break
Close Com Port
Assemble Packet
Clear Screen
Show Hex
meter 23
data received
meter 4
data received
m 2
data received
meter 26
data received
27
data received
sendingt........
r
sending...............
sending................
sending...........
sending...............
sending...............
sending............
COM40 9600 8-N-1 FLOW:NONE
Rx: 187 bytes

Result

From the range coverage testing, it can be observe that Zigbee's transmission data can be up to 28 meters. They cannot communicate after 28 meters. At meter 7, loss of data has started and continues until meter 28 till they cannot communicate with each other anymore.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.0 Introduction

This chapter will describe the overall thought of this thesis of Final Year Project. There are few problem and limitation that were faced during duration were stated and recommendation for future development of this project.

5.1 Conclusion

Recent developments in electronic technology provide a wide range scope of applications for environmental monitoring system. For this matter, the most appropriate Wireless Sensor Network (WSN) has been chosen based on popularity, suitable, common used and past projects. The most appropriate WSN for monitoring is Zigbee based on its characteristics. This project has received various great comments as WSN is widely used and develop these days.

The important part of the system which is transmitter successfully send string to microcontroller and able to differentiate three type of sensors, “AIR HUMIDITY”, “SOIL HUMIDITY” and “TEMPERATURE”. This will allow the system to manipulate data received for various kind of output such as LCD display, LED and Relay.

To make the Receiver module communicates within Transmitter module, student use a Radio Frequency which is Xbee modules and this task success completely. As the data received, the microcontroller will give command to relay whether to on or off depending on data received if the data exceeded user’s limit value.

Overall, the system functions as expected even though student facing a lot of difficulties throughout the completion. The objective of the project is successfully fulfilled.

5.2 Problem and Limitation

1. PIC 16 series - The microcontroller (PIC 16F877A) used has its own limitation in terms of memory, to put many sensor or input, there have limited.

2. Xbee module - The distance of the “Transmitter Module” has its limit which able to communicate within 28 meter range and has some great loss after that.

3. RF Range – The signal can be clash when more of the RF module (Xbee module) is use in one place. The solution is to use encoder and decoder for select the signal address in 8 bits. .

4. Design – The design of both transmitter and receiver board can be smaller and neater etching if the board can be etch with UV box.

5.3 Recommendation

1) Hardware

i) The wireless Xbee transmitter and receiver module need a higher technology device for communication system such as Xbee PRO that can exceed more that 1.0 KM communication range.

ii) The board system can be built in compact design to put in strategic places and reduce the size.

iii) Replacing the etching board with Arduino for easier interfacing without have to use voltage regulator and replacing PIC 16F877A with Rabbit which act as microcontroller but easier program.

REFERENCES

1) BOOKS, JOURNALS AND MANUALS

[1] Andrzej Pawlowski, Jose L. Guzman, Francisco Rodriguez, Manuel Berenguel, Jose Sanchez and Sebastian Dormido, 2009. Simulation of greenhouse climate monitoring and control with wireless sensor network and event-based control. Sensors, 9(1): 232-252. DOI: 10.3390/s90100232.

[2] Carlos Serodio, J. Boaventura Cunha, Raul Morais, Carlos Couto, Joao Monteiro, 2001. A networked platform for agricultural management systems. Computers and Electronics in Agriculture, 31 (1): 75-90. DOI: 10.1016/S0168-1699(00)00175-7.

[3] Hui Liu, Zhijun Meng, Shuanghu Cui, 2007. A wireless sensor network prototype for Environmental monitoring in greenhouses. Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, 2007 (WiCom 2007), 21-25 Sept. 2007, pp: 2344 - 2347.

[4] Teemu Ahonen, Virrankoski, Reino Elmusrati, Mohammed, 2008. Greenhouse monitoring with wireless sensor network. Proceedings of the IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications, 2008 (MESA 2008), 12-15 Oct. 2008, pp: 403-408. DOI: 10.1109/MESA.2008.4735744.

[5] Xiliang Zhang, Weihua Zhang, Xiarong Zhang, Zhezhuang Xu, Feng Zhang, 2008. Optimized deployment of cluster head nodes in wireless network for the greenhouse. Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008 (WiCOM '08), 12-14 Oct. 2008, pp: 1 - 5.

[6] Xiuhong Li, Zhongfu Sun, Tianshu Huang, Keming Du, Qian Wang, Yingchun Wang, 2006. Embedded wireless network control system: an application of remote monitoring system for greenhouse environment. Proceedings of the Multi-conference on Computational Engineering in Systems Applications, IMACS Oct. 2006, pp: 1719 - 1722.

[7] Xuemei Li, Yuyan Deng, Lixing Ding, 2008. Study on precision agriculture monitoring framework based on WSN. Proceedings of the 2nd International Conference on Anticounterfeiting, Security and Identification, 2008. ASID 2008, 20-23 Aug. 2008, pp: 182 – 185 Digital Object Identifier 10.1109/IWASID.2008.4688381.

[8] Yang Shifeng, Ke Jing, Zhao Jimin, 2007. Wireless monitoring system for aquiculture environment. Proceedings of the IEEE International Workshop on Radio-Frequency Integration Technology, 2007 (RFIT2007), 9-11 Dec. 2007, pp: 274 - 277.

[9] Zhou Yiming, Yang Xianglong, Guo Xishan, Zhou Mingang, Wang Liren, 2007. A Design of Greenhouse Monitoring & Control System Based on ZigBee Wireless Sensor Network. International Conference on Wireless Communications, Networking and Mobile Computing, 2007 (WiCom 2007), 21-25 Sept. 2007:2563 - 2567.DOI: 10.1109/WICOM.2007.638

[10] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263 270, Seattle, Washington, USA, August 1999, ACM.

[11] lan F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks. IEEE Communications Magazine," pp. 102-114, August 2002.

[12] M. Beigl, A. Krohn, T. Zimmer, and C. Decker, "Typical Sensors needed in Ubiquitous and Pervasive Computing," In First International Workshop on Networked Sensing Systems (INSS) 2004, pp. 153-158, June 2004.

[13] E. M. Tapia, L. L. Stephen S. Intille, and K. Larson, "The Design of a Portable Kit of Wireless Sensors for Naturalistic Data Collection," In Proceedings of Pervasive2006, pp. 117-134, May 2006.

[14] M. Beigl, A. Krohn, T. Riedel, T. Zimmer, C. Decker, and M. Isomura, "The uPart experience: Building a wireless sensor network,"nIn Proceedings of the ACM/IEEE Fifth International Conference on Information Processing in Sensor Networks, pp. 366-373, Memphis, USA, April 2006.

[15] Kidd, C.D., R.J. Orr, G.D. Abowd, C.G. Atkeson, I.A. Essa, B. MacIntyre, E. Mynatt, T.E. Starner, and W. Newstetter, "The Aware Home: A living laboratory for ubiquitous computing research," in Proceedings of the Second International Workshop on Cooperative Buildings CoBuild'99. 1999.

[16] S. S. Intille, K. Larson, E. Munguia Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research," in Proceedings of PERVASIVE 2006, vol. LNCS 3968, Springer-Verlag, pp. 349-365, 2006.

2) INTERNET WEBSITE

1. Home Automation – gives more information and idea in designing the home automation circuit.
   http://www.home-automation.org/

2. Greenhouse Monitoring System – gives more information about monitoring system in the greenhouse.
   http://www.absoluteautomation.com/applications/greenhouse/index.html

3. Demo: Wireless Communication Protocol
   http://www.youtube.com/watch?v=6lhT5FmCyII

4. MPLAB Integrated Development Environment – tells about MPLAB and how to use MPLAB software to design the program.
   http://www.swarthmore.edu/NatSci/echeeve1/Ref/MPLab/MPLab_Intro.html

5. Serial and UART Tutorial – gives information about UART which is a software that are used to connect zigbee to the laptop. It also teaches on how to use UART.
   http://www.freebsd.org/doc/en/articles/serial-uart/

6. Zigbee wireless standards – gives information about zigbee and its advantages.
   http://www.ember.com/zigbee_index.html

APPENDIX A

MPLAB Program

LCD Interfacing Test Program

```
*************************************************************
        ****/
#include <htc.h>
#include "lcd.h"
//Chip Settings
__CONFIG(1,0x0200);
__CONFIG(2,0X1E1F);
__CONFIG(3,0X8100);
__CONFIG(4,0X00C1);
__CONFIG(5,0XC00F);
//Simple Delay Routine
void Wait(unsigned int delay)
{       for(;delay;delay--)
                __delay_us(100);
}
void main()
{       //Let the Module start up
        Wait(100);

        //Initialize the LCD Module
        LCDInit(LS_BLINK);
        //Clear the Module
        LCDClear();

        //Write a string at current cursor pos
        LCDWriteString("Good Is Great !!");
        Wait(20000);
```

```
//Now Clear the display
LCDClear();
LCDWriteString("God Bless all !!");

//Goto POS (X=0,Y=1 i.e. Line 2)
//And Write a string
LCDWriteStringXY(0,1,"<**************>");
Wait(20000);

//Write Some Numbers
for(char i=0;i<100;i++)
{
        LCDClear();
        LCDWriteInt(i,3);
        Wait(3000);
}
LCDClear();
LCDWriteString("    The  End    ");

//Loop Forever
while(1);
}
```

MPLAB Program in Microcontroller

```
//Project: Xbee
//PIC: PIC16F877A
//Crystal Frequency: 20MHz



#include <pic.h>
#include <htc.h>
__CONFIG(0x3F32);
#define PB1         RC0
#define PB2         RC1
#define PB3         RC2
#define   LED            RD2
#define   LCD_RS         RD6
#define LCD_EN           RD7
#define LCD_DATA    PORTB
#define   LCD_PULSE() ((LCD_EN=1),(LCD_EN=0))
#define BAUD        9600
#define _XTAL_FREQ  20000000

void pic_init(void);
void lcd_init(void);
void lcd_write(unsigned char c);
void lcd_clear(void);
void lcd_goto(unsigned char pos);
void lcd_string(const char *s);
char read_button(void);
int read_a2d(unsigned char channel);
void convert(int no, char base);
void display(char number);
void uart_init(void);
void uart_transmit(char data);

char z,di[10],rf[10],set,mode=1;
//mode=0 for transmitter, mode=1 for receiver
unsigned int counter;

static void interrupt isr(void)
{
if(RCIF==1){
     if(RCREG==0xF1) z=0;
     rf[z]=RCREG;
     z++;}
if(TMR0IF==1){
     TMR0IF=0;
     if(counter<39062) counter++;
```

```
	else{
		counter=0;
		if(set<2) set++;
		else if(set==2) set=0;}
}
}

main()
{char pb=0;
int i,value[3],limit[3];
pic_init();							//initialize
PIC
lcd_init();							//initialize
LCD
uart_init();						//initialize UART
limit[0]=400;
limit[1]=400;
limit[2]=300;
for(;;){
	switch(mode){
	case 0:{
		value[0]=(int)(0.1*read_a2d(0));
		value[1]=(int)(0.1*read_a2d(1));
		value[2]=(int)(0.488*read_a2d(5));
		uart_transmit(0xF1);
		uart_transmit(set);
		uart_transmit((value[0]>>7)&0b01111111);
		uart_transmit(value[0]&0b01111111);
		uart_transmit((value[1]>>7)&0b01111111);
		uart_transmit(value[1]&0b01111111);
		uart_transmit((value[2]>>7)&0b01111111);
		uart_transmit(value[2]&0b01111111);
	break;}
	case 1:{

	if((limit[0]>value[0])|(limit[1]>value[1])|(limit[2]>v
alue[2])) LED=1;
		else LED=0;
		pb=read_button();

		value[0]=(rf[2]<<7)+rf[3];
		value[1]=(rf[4]<<7)+rf[5];
		value[2]=(rf[6]<<7)+rf[7];
	break;}
	}

	switch(set){
```

```
     case 0:{
          if(mode==1) set=rf[1];

          lcd_goto(0x00);
          lcd_string("  AIR HUMIDITY  ");    //LCD display
AIR HUMIDITY
          lcd_goto(0x40);
          lcd_string("      ");          //LCD display value
          convert(value[0],10);
          for(i=3;i>=1;i--) display(di[i]);
          lcd_string("%      ");

          switch(pb){
          case 2:{
               set=3;
               __delay_ms(200);
          break;}}
     break;}
     case 1:{
          if(mode==1) set=rf[1];

          lcd_goto(0x00);
          lcd_string("  SOIL HUMIDITY ");//LCD display SOIL
HUMIDITY
          lcd_goto(0x40);
          lcd_string("      ");          //LCD display value
          convert(value[1],10);
          for(i=3;i>=1;i--) display(di[i]);
          lcd_string("%      ");

          switch(pb){
          case 2:{
               set=3;
               __delay_ms(200);
          break;}}
     break;}
     case 2:{
          if(mode==1) set=rf[1];

          lcd_goto(0x00);
          lcd_string("  TEMPERATURE   "); //LCD display
TEMPERATURE
          lcd_goto(0x40);
          lcd_string("     ");           //LCD display value
          convert(value[2],10);
          for(i=3;i>=1;i--) display(di[i]);
          lcd_write(0);
```

```
        lcd_string("C      ");

        switch(pb){
        case 2:{
             set=3;
             __delay_ms(200);
        break;}}
    break;}
    case 3:{
        lcd_goto(0x00);
        lcd_string("SET AIR HUMIDITY");
        lcd_goto(0x40);
        lcd_string("      ");
        convert(limit[0],10);
        for(i=3;i>=1;i--) display(di[i]);
        lcd_string("%      ");

        switch(pb){
        case 2:{
             set++;
             __delay_ms(200);
        break;}
        case 4:{
             if(limit[0]<1000) limit[0]=limit[0]+10;
             __delay_ms(200);
        break;}
        case 1:{
             if(limit[0]>0) limit[0]=limit[0]-10;
             __delay_ms(200);
        break;}}
    break;}
    case 4:{
        lcd_goto(0x00);
        lcd_string("SET SOIL HUMID  ");
        lcd_goto(0x40);
        lcd_string("      ");
        convert(limit[1],10);
        for(i=3;i>=1;i--) display(di[i]);
        lcd_string("%      ");

        switch(pb){
        case 2:{
             set++;
             __delay_ms(200);
        break;}
        case 4:{
             if(limit[1]<1000) limit[1]=limit[1]+10;
```

```
                __delay_ms(200);
            break;}
            case 1:{
                if(limit[1]>0) limit[1]=limit[1]-10;
                __delay_ms(200);
            break;}}
        break;}
        case 5:{
            lcd_goto(0x00);
            lcd_string("SET TEMPERATURE ");
            lcd_goto(0x40);
            lcd_string("      ");
            convert(limit[2],10);
            for(i=3;i>=1;i--) display(di[i]);
            lcd_string("%      ");

            switch(pb){
            case 2:{
                set++;
                __delay_ms(200);
            break;}
            case 4:{
                if(limit[2]<1000) limit[2]=limit[2]+10;
                __delay_ms(200);
            break;}
            case 1:{
                if(limit[2]>0) limit[2]=limit[2]-10;
                __delay_ms(200);
            break;}}
        break;}
        default:{
            set=0;
        break;}}
}}

void pic_init(void)
{
TRISA=0b00001011;
TRISB=0b00000000;
TRISC=0b10000111;
TRISD=0b00000000;
TRISE=0b00000001;
ADCON1=0b00000110;
OPTION=0b00000000;            //Internal instruction clock

if(mode==0) INTCON=0b10100000;
else INTCON=0b11000000;
```

```
PIE1=0b00100000;              //Enable RX interrupt
PORTA=0b00010000;
PORTB=0b00000000;
PORTC=0b00000000;
PORTD=0b00001000;
PORTE=0b000;
}

/* initialise the LCD - put into 4 bit mode */
void lcd_init(void)
{
__delay_ms(20);                    //delay for LCD Power Up
lcd_write(0x28);              //function set
lcd_write(0x0C);              //display on/off control
lcd_clear();                  //clear screen
lcd_write(0x06);              //entry mode set

LCD_RS=0;
lcd_write(0x40);
LCD_RS=1;
lcd_write(0b01110);
lcd_write(0b01010);
lcd_write(0b01110);
lcd_write(0b00000);
lcd_write(0b00000);
lcd_write(0b00000);
lcd_write(0b00000);
lcd_write(0b00000);
}

/* write a byte to the LCD in 4 bit mode */
void lcd_write(unsigned char c)
{
LCD_DATA=(LCD_DATA&0x0F)|(c&0xF0);
LCD_PULSE();
LCD_DATA=(LCD_DATA&0x0F)|((c<<4)&0xF0);
LCD_PULSE();
__delay_us(40);
}

/* clear LCD and goto home */
void lcd_clear(void)
{
LCD_RS=0;
lcd_write(0x1);
__delay_ms(2);
}
```

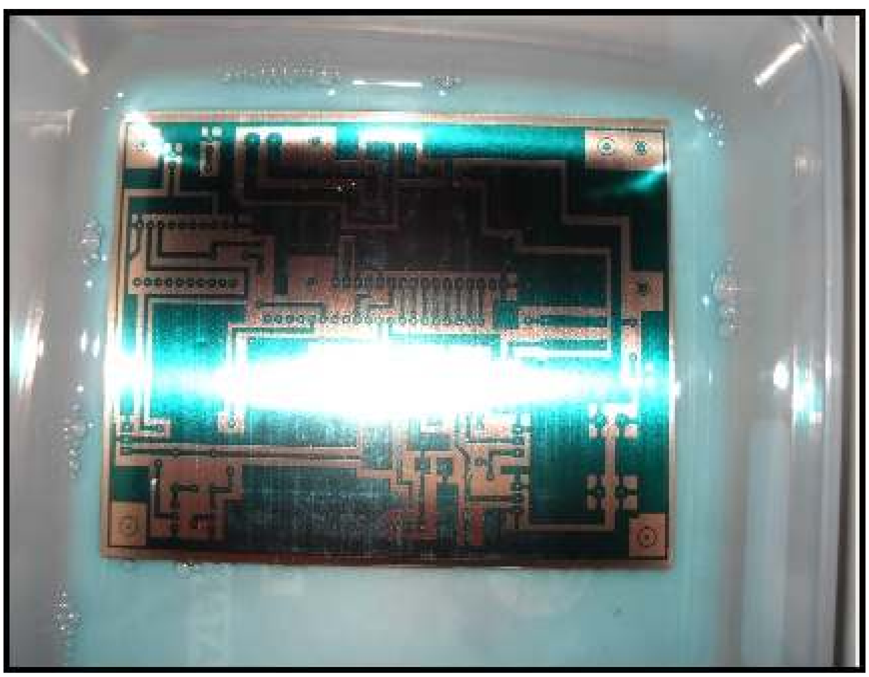
```
/* write a string of chars to the LCD */
void lcd_string(const char *s)
{
LCD_RS=1;                // write characters
while(*s)
lcd_write(*s++);
}

/* go to the specified position */
void lcd_goto(unsigned char pos)
{
LCD_RS=0;
lcd_write(0x80+pos);
LCD_RS=1;
}

char read_button(void)
{char i=0;
if(PB1==1) i=i+1;
if(PB2==1) i=i+2;
if(PB3==1) i=i+4;
return i;
}

int read_a2d(unsigned char channel)
{ADCON0=0b00000001;                    //Turn on A/D
module
ADCON1=0b10000000;                     //configures
analog and voltage reference pins
ADCON0=(ADCON0&0xC7)|(channel<<3);     //select analog
input channel
__delay_ms(2);
ADGO=1;
     //initiate conversion on the selected channel
while(ADGO==1) continue;               //wait until
conversion done
return(256*ADRESH+ADRESL);}

void convert(int no, char base)
{char i;
for(i=0;i<=9;i++) di[i]=0;
i=0;
do{
     di[i]=no%base;
     no=no/base;
     i=i+1;}
```

```
while(no!=0);
}

void display(char number)
{switch(number){
case 0:{lcd_string("0");break;}
case 1:{lcd_string("1");break;}
case 2:{lcd_string("2");break;}
case 3:{lcd_string("3");break;}
case 4:{lcd_string("4");break;}
case 5:{lcd_string("5");break;}
case 6:{lcd_string("6");break;}
case 7:{lcd_string("7");break;}
case 8:{lcd_string("8");break;}
case 9:{lcd_string("9");break;}
case 10:{lcd_string("A");break;}
case 11:{lcd_string("B");break;}
case 12:{lcd_string("C");break;}
case 13:{lcd_string("D");break;}
case 14:{lcd_string("E");break;}
case 15:{lcd_string("F");break;}
default:{lcd_string("?");break;}
}}

void uart_init(void)
{
TXSTA=0b10100000;
RCSTA=0b10010000;
SPBRG=(int)(_XTAL_FREQ/(64.0*BAUD)-1);
}

void uart_transmit(char data){
while(TXIF==0) continue;
TXREG=data;
}
```

APPENDIX B

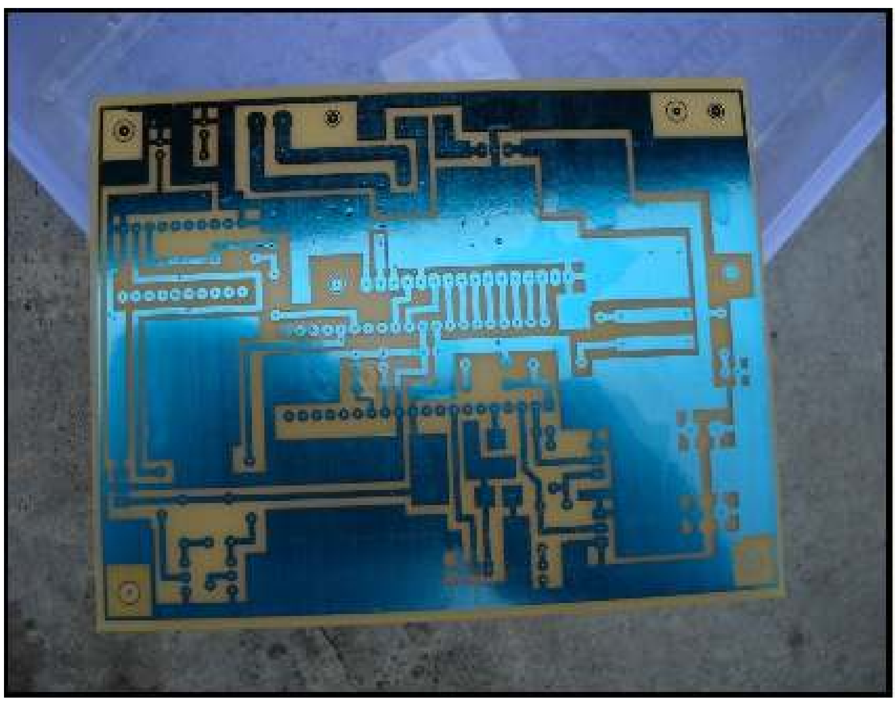
Etching Process

Etching Process

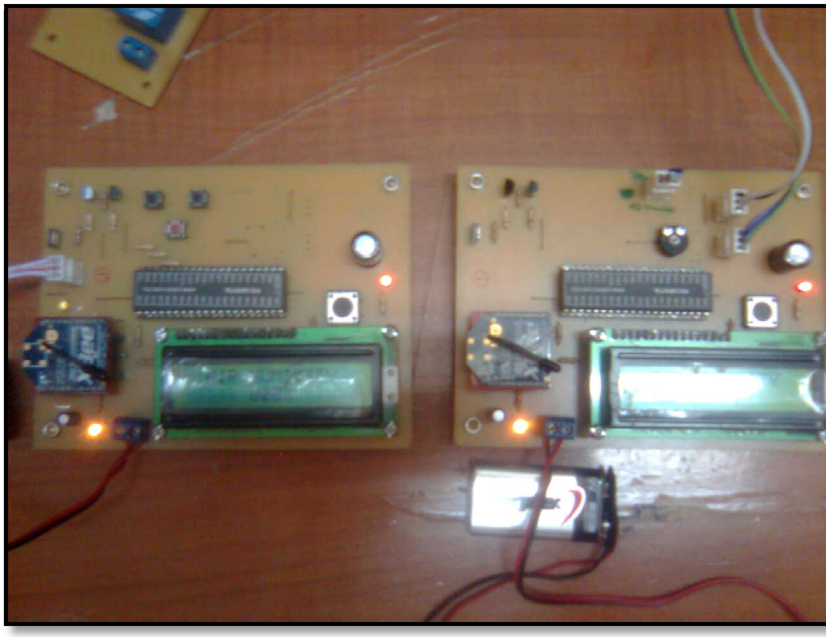


PCB layout after etching process

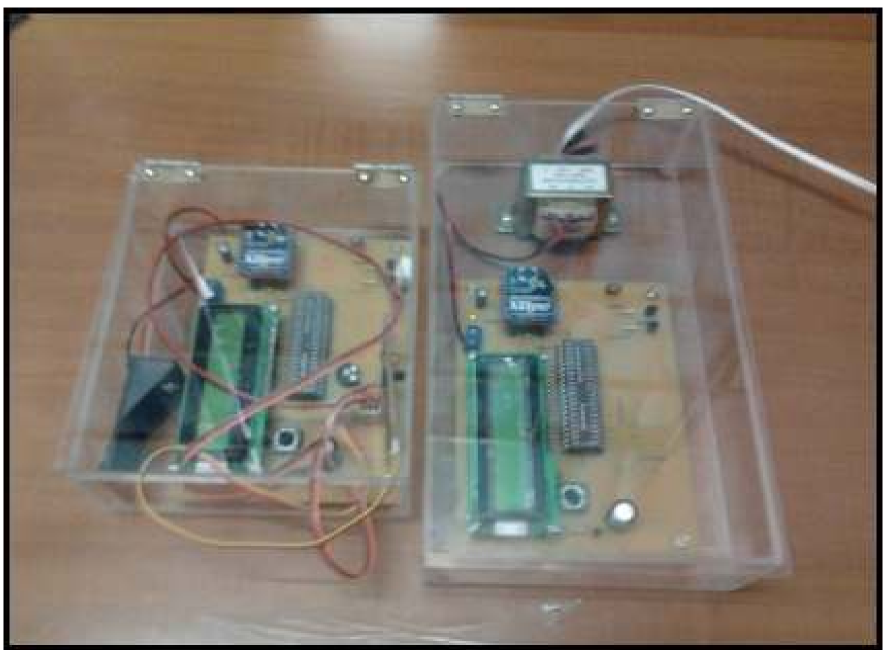
Drilling process



Complete PCB board



Transmitter and receiver board



Transmitter and receiver board with casing