Wireless Sensor Data Logging System Design

By

Noorshafrina Binti Zulkalnain

FINAL REPORT

Submitted to the Electrical & Electronic Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Wireless Sensor Data Logging System Design

By

Noorshafrina Binti Zulkalnain

An Final Report Submitted to the
Electrical & Electronic Engineering Programme
Universiti Teknologi PETRONAS
in Partial Fulfillment of the Requirement for the
Bachelor of engineering (Hons)
(Electrical & Electronic Engineering)

Approved by,

____

(Dr. Mohd Zuki Bin Yusoff)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

MAY 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

____
(Noorshafrina Binti Zulkalnain)

DEDICATION

To

my parents,my mother,Norlela

my father,Zulklanain

And my siblings.

ACKNOWLEDGEMENT

It's been a long journey in UTP, 5 years living as a student and now finally as a final year student. The journey was great and many special moments I have been through together with my beloved friends. Learning is always a lifelong journey that demands a lot of dedication, passion and patience. Transitioning from a student to a scholar, one's route to knowledge quest is always unique in one's way. Some people manage to achieve their search for knowledge in a sweet and straightforward way. On contrary; some other people's knowledge seeking paths are bitter and full of winding ridges. Many sweats and brain juggling have been generated and many mind boggling of this Degree study proves that ultimate life balance and perseverance do pay.

During one year of my Final Year Project, I have worked with a great number of people whose contribution in assorted ways to the research and the making of the thesis deserved special mention. It is a pleasure to convey my gratitude to them all in my humble acknowledgment.In the first place,I would like to record my gratitude to my supervisor, Dr.Mohd Zuki supervision, advice, and guidance from the very early stage of this research as well as giving me great ideas throughout the work. Above all and the most needed, he provided me unflinching encouragement and support in various ways. Many thanks go in particular En.Azhar, the EE lab technician for giving ideas and advice to conduct the sensors experiment. I gratefully thank to the chemical lab technician, En.Sulaimen for lending me help in conducting the humidity and temperature sensor accuracy experiment. I would also acknowledge Mr. Hasrul Firdaus for his help in producing the printed circuit board for my project.I would also like to thank my friends for supporting me all along the way and creating a great friendship in the Universitiy. Finally, I would like to thank to my beloved mother, Norlela and father, Zulkalnain, who up brought me to be what I am today, and who always extends their prayers and best wishes.Last but not least,I would like to thank my siblings Noorazreen, Khairi, Asyraf, Syahmi for being there whenever needed and also for supporting me.

ABSTRACT

Wireless Sensor Data Logging System Design is a standalone electronic sensor device that captures and stores data through wireless communication. This system comprises two main integrated components; the Radio Frequency module and the Microcontroller based system. The main goal of this project is to design and construct a data logging system that effectively monitors the device's measurement values. In real life applications, most data monitoring system is a passive system. This type of system requires manned guarding on site to manage the devices. Therefore, a standalone data logging system offers a better enhancement system to replace the manned guarding method. The standalone data logger system can be applied by leaving the device alone in any place that requires the measurement of humidity and temperature. These data can be retrieved from EEPROM and transferred to a PC whenever needed by a user. A radio frequency module enables these data travels through wireless transmission medium, whereas the serial communication interface enables communication between the devices and PC. For diverse applications, an alarm system can be implemented if assets and security are the major concerns. The final report presents the development of a data logger system which is an integration of radio frequency module and the microcontroller-based system. The system monitors the device's measurement value via a Graphical User Interface. Basically, the system introduces a RF module to replace the hard wired scheme and produce a dynamic data transmission system. It is geared up with a PIC16F877A microcontroller to drive the outputs besides providing communication between devices and a PC. Overall, the project is the best platform to improve the traditional monitoring system and ignites another innovative invention in the future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PCB	Printed Circuit Board
UART	Universal Asynchronous Receiver Transmitter
EEPROM	Electrically Erasable Programmable Read Only Memory
I2C	Inter-Integrated Circuit
MCU	Microcontroller
PC	Personal Computer
LCD	Liquid Crystal Display
GUI	Graphical User Interface
FYP	Final Year Project
IDE	Integrated Development Environment
CCS	Code Composer Studio
PIC	Programmable Interface Controller
ICSP	In-Circuit Serial Programming
MCLR	Master Clear
VCC	Voltage Common Collector
PGD	Programming Data
PGC	Programming Clock
EN	Enable
R/W	Read or Write
SDA	Serial Data
SCL	Serial Clock

CHAPTER 1
INTRODUCTION

Wireless Sensor Data Logger System Design preface and background will be explained comprehensively in this chapter. All the information gathering and research were undertaken through many resources such as internets, books, journals and guidelines from lecturers. The elements that will be emphasized in this chapter are the background of study, problem statement, objectives and scope of study. The details discussed throughout this chapter will help the readers grasp the idea of the project and understand the concepts and principles applied.

1.1 Background of Study

.

The Wireless Sensor Data Logger System Design was designed based on the problems faced in the passive manned guarding system. Usually, the passive manned guarding system requires a person to monitor and guard the devices for the required time. Therefore, the data logger is designed to replace this less reliable and ineffective system. A data logger is a standalone sensor device that has the ability to store data in an external or internal memory. It refers to a system that is used to effectively measure and record important physical parameters such as humidity and temperature measurement. They are great portable device which can function independently without anyone to guard it. It can be taken anywhere and used in various situations. Whenever the measured data is needed, it is being collected. The sensors are the important hardware tools that actually take these measurements [1, 2].

Many industries around the world rely on the very regular use of such system especially the food and beverage industry. The data logger is helpful in controlling certain aspects when it comes to dealing with products that are being stored. The quality of a product must be assured in a good condition when it arrives to its final destination [3]. These devices are useful in restaurants to save a record of food temperature which are kept in the refrigerator. Bacteria increases and grows rapidly as it reaches the temperature between 4℃ to 60℃ [4]. Therefore, it is very important for the food industry to make sure the foods are at their proper temperature in order to prevent people from getting sick. Humidity measurement also plays an important role for the transport of some goods such as flowers [1, 3]. Humidity must be in high condition to avoid the flowers from getting dried out and wilt. These plants need to hold onto their vital moisture content to live [5].

Therefore, this wireless sensor data logger system was designed to satisfy the requirements stated above. It was developed based on the integration of a number of subsystems; the radio frequency module, the microcontroller-based systems, external EEPROM integration, sensor device, serial communication interface and GUI (Graphical User Interface). This integrated device is an efficient system where it is portable, accurate, less expensive and light weight [1]. These great qualities made the system very competitive and reliable for real application.

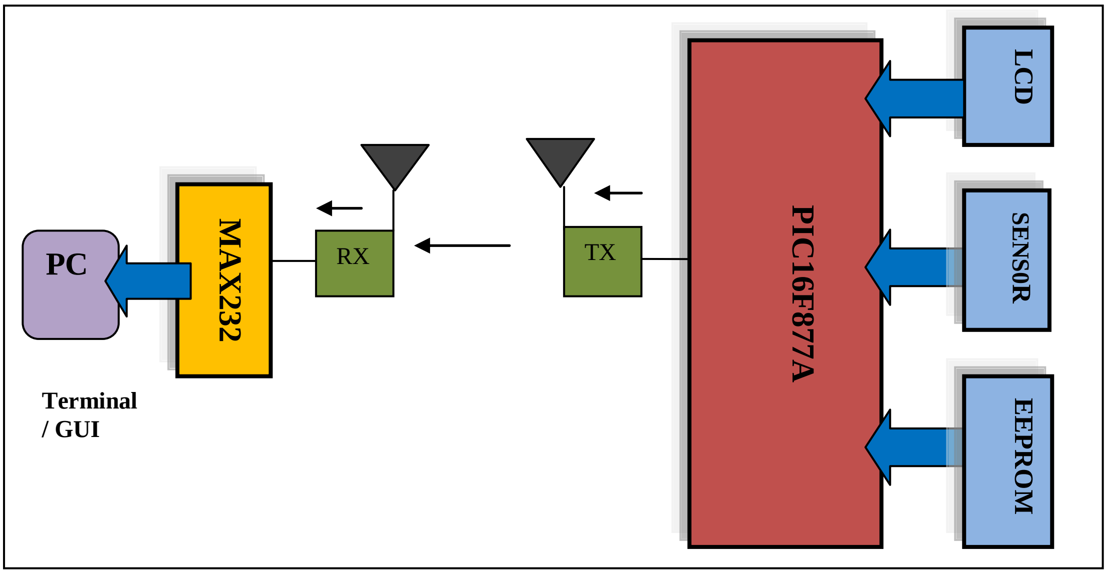

Figure 1: Block Diagram of wireless data logger system design using MCU.

The system is basically illustrated in the block diagram of Figure 1. Apparently; the measured data is obtained from the data logging system. The system was designed to store data received from the sensor that is attached to the microcontroller. The microcontroller is installed in the data loggers in order to interface with computer programs [1]. The value of each data will be transmitted via wireless communication medium utilizing radio frequency module at 433MHz operating frequency. Simultaneously, the data is transferred to serial communication interface via the serial communication system to enable communication between device and a PC. The system is made interactive with the aid of GUI for easy-handling purposes. Besides, the system is capable of monitoring the status in textual form via Terminal program. The system will operate in one direction communication where the value of the sensor is sent to the PC. Overall, the Wireless Sensor Data Logger System comprises the integration of hardware and software that offers an interactive, effective and reliable data logging system.

1.2 Problem Statement

1.2.1 Problem identification

Typically, most of traditional monitoring systems perform passive guarding system. Apparently, these systems are less reliable and ineffective. The hasty changes in technology nowadays made this existing system merely inconvenient. Consequently, a better approach should be implemented for an advancement of this passive system. The time management and quality of a product should be managed wisely in order to achieve a productive and smooth operation process. Some decisions must be made based upon the data and these decisions are important for the safety of human being. For instance the temperature of food must be recorded to keep of track the growth of bacteria in the food. Therefore, the *Wireless Sensor Data Logger System Design* perhaps introduces revolution for the passive manned guarding system.

1.2.2 Problem solution

The project essentially offers a dynamic and efficient system that handles the recorded data through a wireless standalone device that is displayed to PC. Basically, the project is based on the problem analysis basis and extends it to problem solving before it is implemented as a whole. It acts as a self contained unit that does not require any help from hosts to operate. Compared to conventional interface devices, this data logger has the capability to dump or transfer the data to a host system, if required. These data can be saved and analyzed for historical archive purposes [6]. The *Wireless Sensor Data Logger System Design* demonstrates the integration and application of theories in engineering discipline, which is a good platform for better understanding on engineering principle applications.

1.3 Project Objectives

The objectives of the project focus on the steps towards the final design of the data logger system which is based on engineering fundamentals and problem solving basis. The aim and goal of this project are as follows:

i. To design a wireless system that can record and save the sensor data into an external EEPROM chip for the required range of time.

ii. To integrate the radio frequency module with a microcontroller-based system to enable the data transfer through wireless transmission medium.

iii. To reduce and manage the wiring of a circuit by designing its printed circuit board.

iv. To design a graphical user interface that can display the values of measured and recorded data. This interface contributes a user-friendly system to the real environment.

1.4 Scope of Work

The scope of work in this chapter is based on elements listed as below:

i. Integration of sensor and EEPROM device with microcontroller.
ii. Wireless data transmission via a radio frequency module.
iii. Interface the data logger system with serial communication.
iv. Implementation of graphical user interface software design.

The work is based on the elements above which apparently consist of a wireless data logging system. The microcontroller offers various advancement designs for the whole system. The Radio frequency module introduces an alternative solution for data transmission and it is applicable for various fields. Radio frequency transmission medium offers wider coverage area compared to other mediums.

The serial communication is designed in order to communicate with PC which practically improves the data logger system. Finally, the graphical user is implemented to improve the data logging system which makes the system interactive to the end user.

CHAPTER 2
LITERATURE REVIEW

A literature review is one of the important development stages where the knowledge of each element used in the project is introduced. This particular chapter actually discusses the fundamental concepts applied in the project. The resources obtained through research via various sources are covered in this chapter. The literature review comprises external EEPROM interface, PIC16F877A microcontroller, serial communication interface, characteristics of wireless system, printed circuit board and graphical user interface via Visual C++.

2.1 Introduction of Microcontroller

A microcontroller is a single silicon chip which includes at a minimum microprocessor, program memory, data memory and an input output device. The word 'micro' reflects that the device is small while the word 'controller' refers to the use of it in control applications. An embedded controller is another term for a microcontroller since most of microcontrollers are built in the devices they control. The main difference between a microprocessor and a microcontroller is that a microprocessor requires several other components for its operation such as program memory, data memory, input output devices and an external clock circuit. On the other hand, a microcontroller consists of all the support chips embedded inside its single chip. Other additional components such as timers, counters and analog-to digital converters are included in certain microcontrollers. Thus, a microcontroller system can act as a large computer with hard disks, floppy disks and printers to a single-chip embedded controller. They also can be used and embedded into household goods and other electronic controlled devices such as refrigerators, implantable medical devices, remote controls, office machines, toys, appliances, microwave ovens and cookers [7].

A set of instructions stored in a memory of microcontroller can be operated by a microcontroller by fetching the instructions from its memory one by one; then these instructions are being decoded to carry out the required operations. The program languages used to program a microcontroller can either be an assembly language or a high level language. An assembly language is faster compared to a high level language, but it is hard to learn and maintain the program written because an assembly program consists of mnemonics. Different firms manufacture microcontrollers with different assembly languages; therefore, the user has to learn a new language for every new microcontroller used. High level languages are well known languages that facilitate the development of large and complex programs. User Programs which are loaded in the microcontroller's memory are executed. The data are received from input devices, manipulated and sent to output devices [7].

As a powerful tool, a microcontroller allows designers to design sophisticated input output data manipulation under program controls. They are classified by the number of bits they process. In most microcontroller-based applications, 8 bits are widely used and popular among the users. The 16 bits and 32 bits are expensive and not required in small or medium size general purpose applications compared to 8 bits, but they are much more powerful. The architecture of a microcontroller consists of a microprocessor, a memory and an input output. The central processing unit (CPU) and the control unit (CU) are the elements of the microprocessor. The CPU is referred as the brain of the microcontroller. Arithmetic and logic operations are performed here. The required instructions can be carried out by letting the CU control the internal operations of the microprocessor and send signals [7].

2.2 Architecture Overview of PIC16F877A

One of the most advanced and well known microcontrollers from Microchip is PIC16877A. In modern applications, the controller is widely used for experimental purposes since it is less expensive, high quality and easily available in market. It can be applied in various applications such as machine control applications, measurement devices, study purposes and so on. Compared to other microcontroller family series, PIC16F877A features all the components which modern microcontrollers normally have and also has more advanced and developed features [8]. The features, pin diagrams and specifications of PIC16F877A are shown in the datasheets from **APPENDIX E.**

From its data specifications**,** the microcontroller has 8K Word which is 14.2Kbytes Flash, 368 RAM, 256 EEPROM and 20MHz of operating frequency. The synchronous serial port can be configured as either 3 wired Serial Peripheral Interface or as the 2 wired Inter Integrated Circuit bus and a Universal Asynchronous Receiver Transmitter. These features makes this microcontroller chip ideal for more advanced level analog to digital applications in automotive, industrial, appliance and consumer applications [9]. Table 1 below summarizes the PIC16F877A specifications and other PIC16F87X as well:

Table 1: PIC16F877A and PIC16877X Microchip specifications [9].

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

By understanding the block diagram of PIC MCU, the idea of how to execute programs and manipulate data in the PIC MCU is easily understood. The block diagram is actually the architectural drawing of its inner workings. Processor block diagrams are basically similar for each of PIC MCU processor families. There are only certain things that might not be the same such as how data is accessed in different register banks, how data is indexed and stored in stacks. The internal block diagram of PIC16F877A microcontroller is shown in Figure 2 [10].

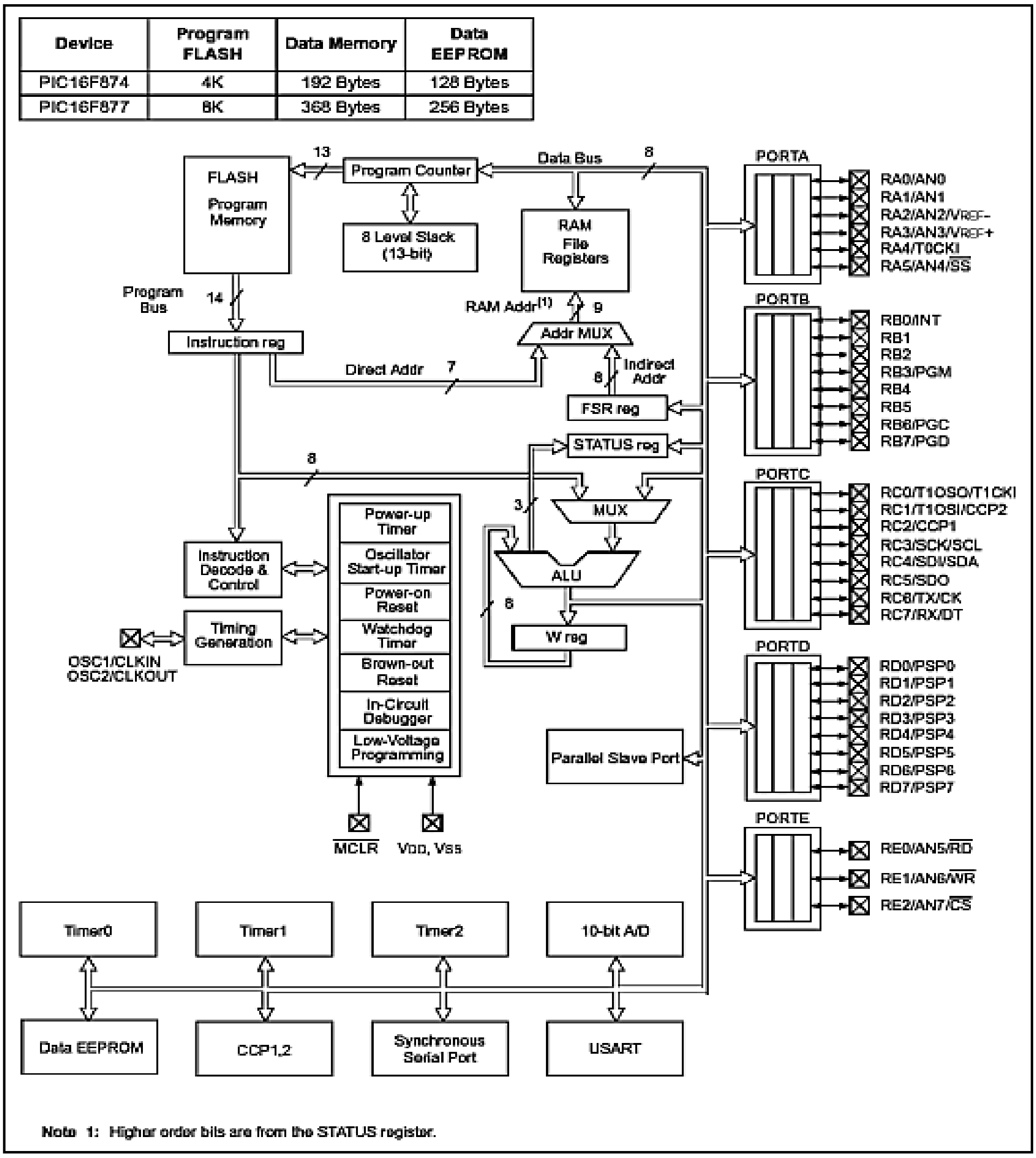

Figure 2: Internal block diagram of PIC16877A.

The arithmetic logic unit (ALU) provides basic arithmetic and bitwise operations for the processor of PIC microcontroller. The input-output registers and the data storage RAM registers are specific use registers that control the operation of the CPU. These registers sometimes can be called as hardware registers. It depends on the function they perform. Hardware registers can provide direct manipulation of functions that are invisible to the programmer such as the program counter which allows advanced program functions. The data storage registers, RAM, are known as file registers by Microchip. The registers have their own spaces because they are separated from the program memory [10] .This is called the Harvard architecture which is shown in Figure 3 below:

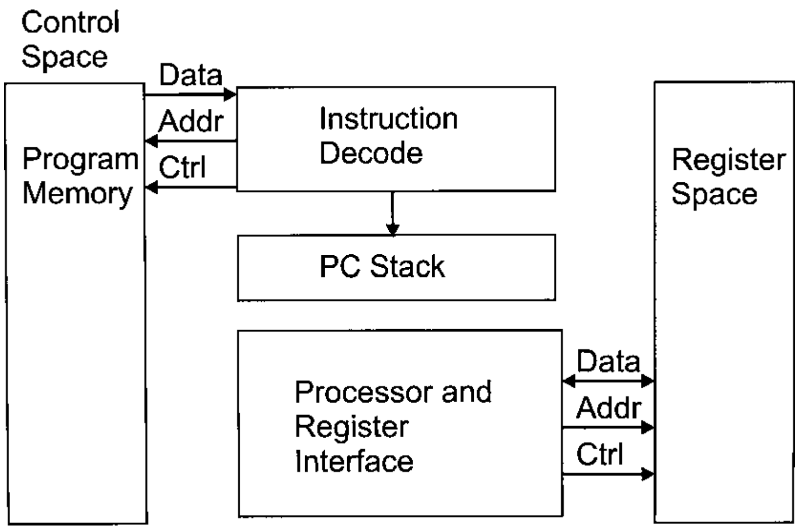

Figure 3: Harvard Architecture block diagram [10].

The purpose of this separation is to allow the program memory read instructions while the processor is accessing data and processing it. Therefore, the PIC microcontroller has the capability to execute software faster than many of its contemporaries. Instruction executions are performed based on the four clock cycles shown in Figure 4 below. Program memory will fetch the next instruction to be executed during an instruction execution cycle. The fetched instructions are latched in a holding or decode register. After an instruction has been fetched and is latched in a holding or decodes register, the program counter is incremented. This is shown in the first cycle, Q1, of Figure 4 [10].

In the next cycle, Q2, the data to be processed are read and put into temporary buffers. The data processing operations takes place during the third cycle, Q3. Last but not least, the resulting data value is stored during the last cycle, Q4 [10].

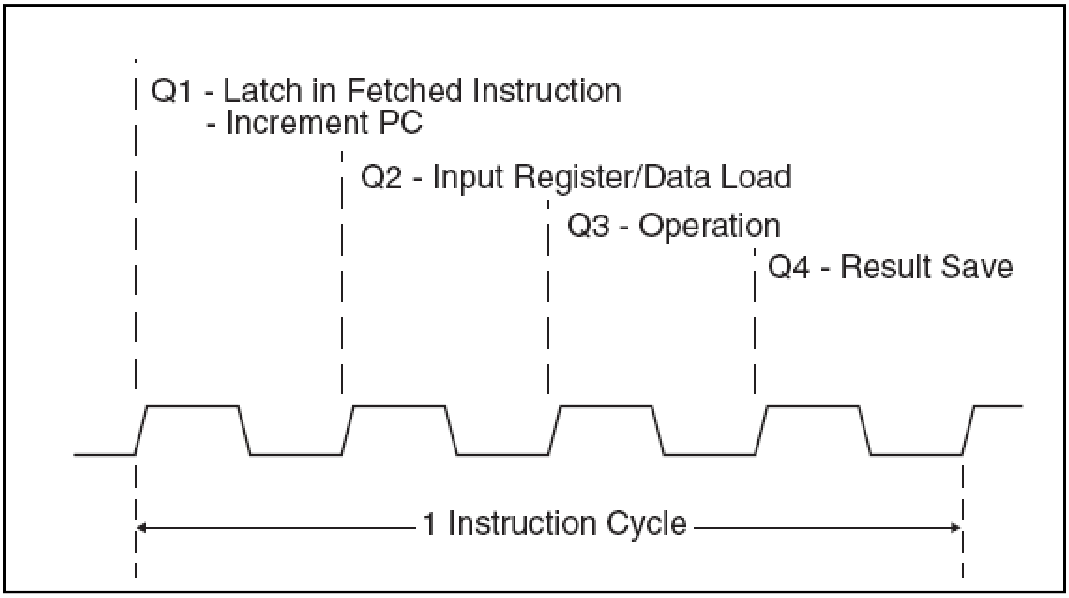

Figure 4: Four clock cycles for instruction execution.

The 7 address bits are explicitly defined as part of the instructions when accessing the PIC16 microcontroller family series. These 7 bits can specify up to 128 addresses in an instruction. The 128 register addresses can also be known as a bank**.** For the program counter, it maintains the current program instruction address in the program memory which contains the instructions for the PIC microcontroller processor. Each one is read out in sequence and being stored in the instruction register. The instruction decode and control circuitry will decode the program. The code that is executed takes place in the program memory. At each address, the content of the program memory consists of a full instruction. From the block diagram, a temporary holding register known as an accumulator is required to save a temporary value while the instruction fetches data from another register. Another alternative is by passing a constant value from the instructions. In this case, the accumulator used is the working register which is also called as the *w register* [10].

2.2.1 Parallel input/output ports

With respect to PIC16F877A, microcontroller ports are recognized and set according to the functionality of the required system. PIC16F877A is a family of PIC16 series which is more powerful in terms of its updated technology, enhancement of capacity and speed [10]. The pin diagram and its associated description are depicted in Figure 5 and Table 2 below, respectively.

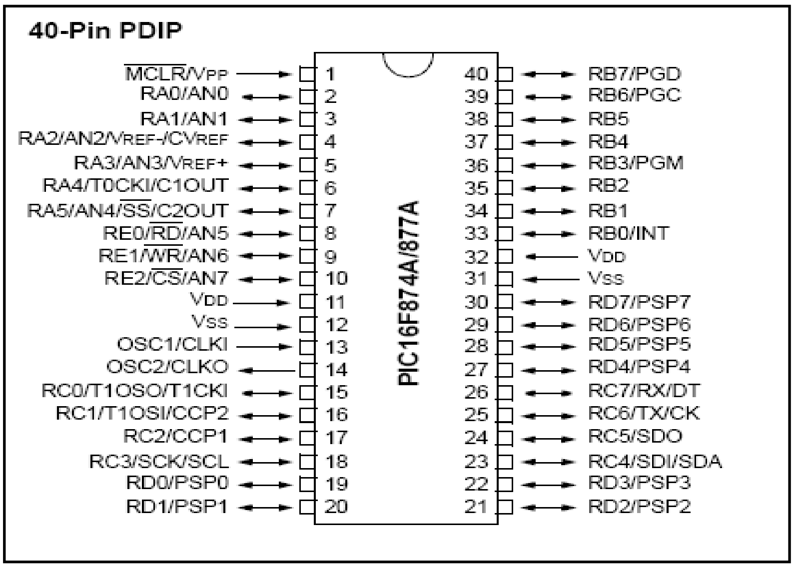

Figure 5: Pin diagram for PIC16F877A.

Table 2: Pin description for PIC16F877A.

ASSIGNED PIN	DESCRIPTION
RB7/PGD	Programming data for ICSP
RB6/PGC	Programming clock for ICSP
MCLR	Master clear for ICSP
VDD	Voltage power
VSS	Voltage ground
OSC1, OSC2	Oscillator 1 and 2 for crystal
RD4 - RD7	Port D4 to port D7 for input/output
SDA, SCL	Serial data and serial clock for I2C activity

2.2.2 The clock oscillator and instruction cycle

Any microcontroller is a complex electronic circuit, made up of sequential and combinational logic. At certain speed it steps in turn through a series of complex states, each state being dependent on the instruction sequence it is executing. Overall, the speed of the microcontroller operation depends on the clock frequency. Many essential timing functions are also derived based on clock frequency ranging from counters and timers functions to serial communications. Overall, the power consumption of the microcontroller strictly depends on the clock frequency where high operation speed uses more power compared to slow speed. Basically, the microcontroller has its specified range for its clock frequency. The selection of the clock frequency is up to the designer. The main clock signal is divided down by a fixed value into a lower frequency within a microcontroller. Each cycle of this slower signal is known as a machine cycle or an instruction cycle. In the action of the processor, the instruction cycle becomes the primary unit of time. For instance, it can be used to measure how long an instruction takes to execute. Basically, the original clock signal is retained to create time stages within the instruction cycle. In order to produce the instruction cycle time, the main oscillator signal in PIC16 series is divided by 4[11].

2.2.3 The timers module

In any microcontrollers, a timer is one of the important elements. Generally, a timer is a counter which is driven from either an external clock pulse or the microcontroller’s internal oscillator. It can be either 8 bits or 16 bits wide. Under the control of the program control, timers can load data. The program control can stop or start the timers. An interrupt can be generated by configuring the timers when a certain count is reached. The interrupts can be used by user program to carry out accurate timing-related operations inside the microcontroller [7].

2.2.4 Power supply and its operating conditions

The standard logic voltage of most microcontrollers are 5V. There are also microcontrollers that can operate as low as 2.7V and some will tolerate 6V without any problem. In the datasheets, the information about the allowed limits of power supply voltage is stated. Basically, the voltage regulator is used to obtain the required power supply voltage when the device is operated from a main adapter or batteries. For instance, a 5V regulator is required in order to operate the microcontroller from 5V using a 9V battery.

2.2.5 The power on reset

In microcontrollers, there is a built in power on reset that keeps the microcontroller in the reset state until all the internal circuitry has been initialized. It can start the microcontroller program back to the beginning and is known as the state on power up. The microcontroller also can be reset by an external reset button.

2.3 Programming PIC Microcontrollers

After the program is written and translated into executable code, the resulting HEX file is loaded to the target microcontroller's program memory with the help of a device programmer. Some microcontroller development kits include on-board device programmers, so the microcontroller chip does not need to be removed and inserted into a separate programming device.

2.3.1 In circuit serial programming (ICSP)

The In Circuit Serial Programming (ICSP) circuit must be connected to the MCU in order to burn the chip. ICSP is actually a method where it is easier to program a PIC Microchip without removing the chip from the development board.

The connection of ICSP with the microchip is simple. ICSP provides five connections from the PIC ICSP programmer to the developer's board as described in Table 3.

Table 3: ICSP pin connections to microcontroller.

PIN	DESCRIPTION
MCLR (MASTER CLEAR)	Programming voltage, reset button can connect here to reset the program of chip
VCC(VOLTAGE COLLECTOR)	Power voltage, usually 5V is used
GND(GROUND)	Zero voltage
PGD (PROGRAMMING DATA)	Connected to RB7 which is the ICSP Data (ICSPDAT)
PGC(PROGRAMMING CLOCK)	Connected to RB6 which is the ICSP Clock (ICSPCLK)

Figure 6 below shows the physical pin connections of ICSP with the PIC16877A microcontroller:

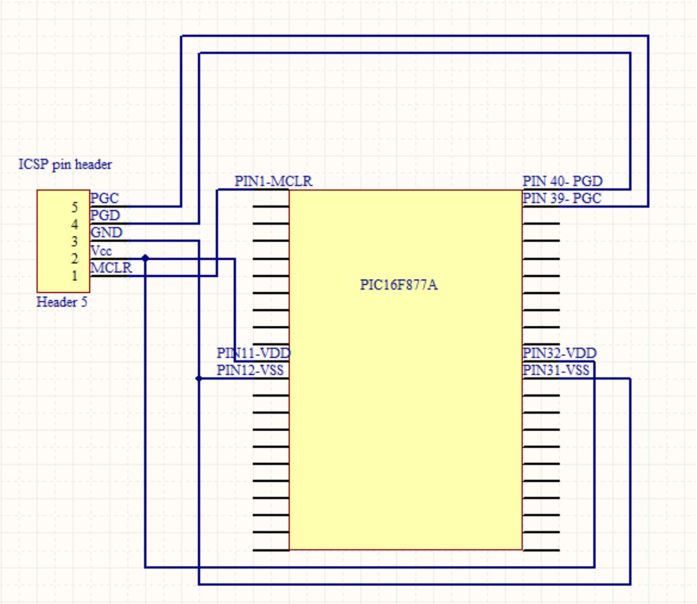

Figure 6: Pin connections of ICSP with PIC16F877A.

2.4 The Human and Physical Interface

Human interface can be devices that can give input and data response from the input data.Switches, keypads, sensors are some examples of input devices. While the output devices are the device that responds to the input device. It can be liquid crystal displays, motors, LEDs and so on.

2.4.1 Liquid crystal display interface

Liquid Crystal Displays (LCDs) consist of many types such as 1 line, 2 line and 4 line LCDs. We will be using a 1 line version with 16 characters. An LCD usually has 1 controller which can support about 80 characters. The LCD used has 14 pins with 2 extra pins. It is classified into 2 groups, which are serial and parallel connections. This LCD is a device where alphanumeric output can be displayed from microcontroller-based circuits. In serial LCD, it requires less input or output resources but they execute slower than the parallel LCD. LCD can be interfaced with various microcontrollers whether 4 bit or 8 bit [12]. Using a 4 bit LCD interface, one can reserve other ports of microcontroller for other functions. Figure 7 below illustrates the serial LCD connection. The description of the pins can be referred in **APPENDIX E.**

.

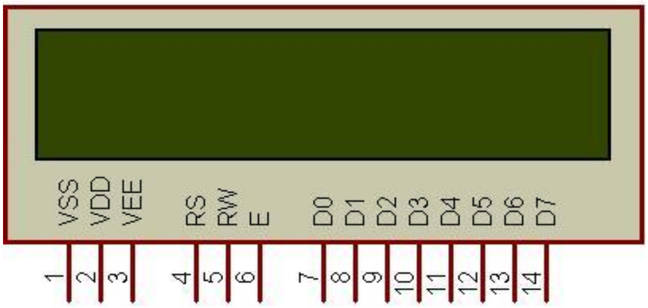

Figure 7: Serial LCD connection [12].

2.4.2 Humidity and temperature sensor

Humidity sensor is a sensor which measures and regularly reports the relative humidity in air. It is designed to sense relative humidity which measures both air temperature and moisture. The relative humidity is usually expressed as percentage which is the ratio of actual moisture in the air to the highest amount of moisture air of the measured environment. The warmer the air is, the more moisture it will be. Therefore, the relative humidity actually changes with fluctuations in temperature [13].

2.5 Serial Communication Overview

The serial communication is basically a method to send data into PC and vice versa. The serial communication interface makes communication between microcontroller and PC significant to a system. In addition, the computer programs are capable of sending data in bytes to transmit pin output and retrieve bytes from the receive pin input. The serial port converts data from parallel to serial forms; besides it changes the electrical representation of the data. Figure 8 below depicts a connection between PC and MAX232.

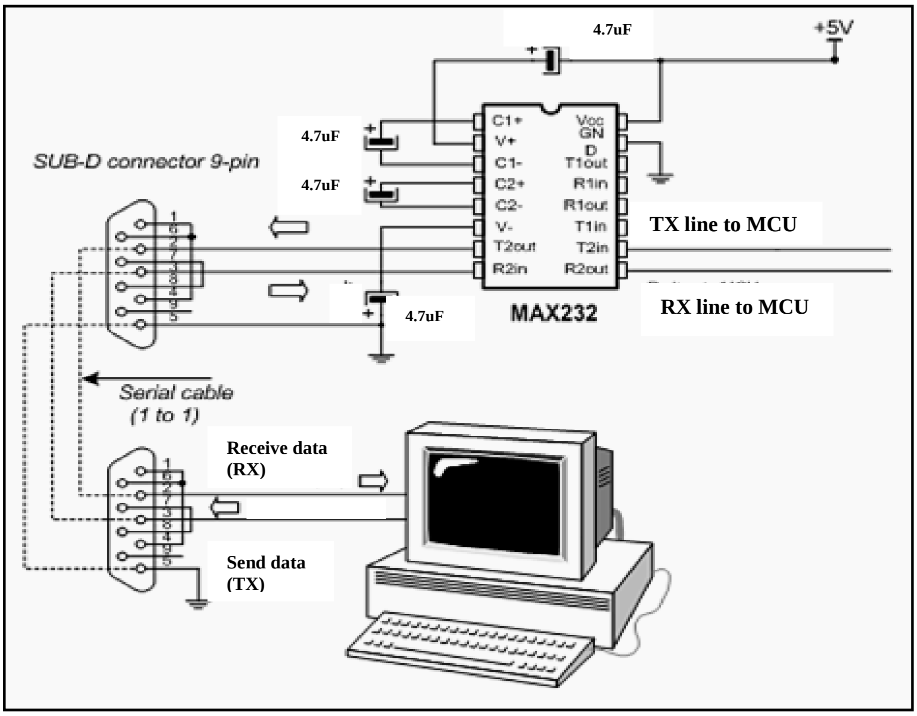

Figure 8: Connection between PC and MAX232.

2.5.1 Inter-intergrated circuit (I2C) protocol

The data communication of sensor is based on I2C method where the two I2C signals are serial data (SDA) and serial clock (SCL). Together, these signals make it possible to support serial transmission of 8-bit bytes of 7-bit-data device addresses plus control bits-over the two-wire serial bus. The device that initiates a transaction on the I2C bus is termed the master. The master normally controls the clock signal. A device being addressed by the master is called a slave. The data from humidity sensor is transmitted and received by the serial data and serial clock signals and being sent to the microcontroller [14].

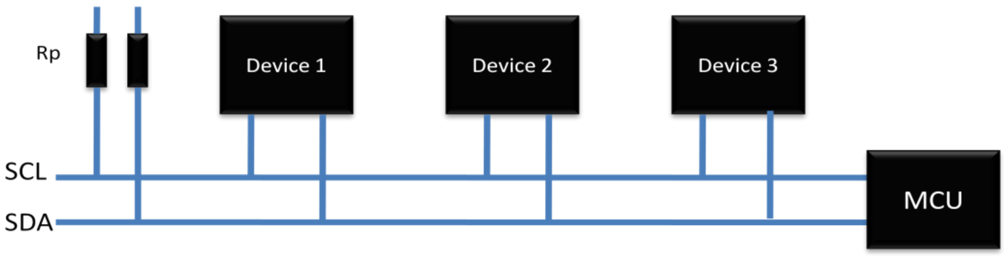

Figure 9: Communication configuration for I2C activity [14].

2.6 Radio Frequency (RF) Module for Wireless Communication

Radio frequency module ia an essential sub-system in the data logger system design. The subsystem is a wireless data link comprising radio frequency transmitter and receiver. TX434 and RX434 are selected for the system. These radio frequency modules require no licensing since the transmitter and receiver are used in accordance with low power devices such as in data logger applications.

Figure 10: The transmitter and receiver for RF module.

2.7 External EEPROM Memory Device

The external EEPROM is a storage device that can store data for a long term since it has more than 200 years of data retention. The external EEPROM can be connected through I2C protocol or Serial Peripheral Interface (SPI) protocol. It depends on the chip we used. We plan to use the 24L256 EEPROM chip to interface with the microcontroller by using the I2C protocol. This storage device interface concept can be applied to Secure Digital (SD) card and Universal Serial Bus (USB) device. This advanced, low power device has a write capability of up to 64 bytes of data and capable of both random and sequential reads up to 256K boundary. By using the external EEPROM, we can connect more than 1 EEPROM chip to create a memory of more than 256K bytes instead of using the built-in internal EEPROM. EEPROM uses floating gate technology. Its dimension is finer, so that it can exploit another means of charging its floating gate. This is known as Nordheim Fowler tunneling. With this method, it is possible to electrically erase the memory cell, as well as write to it. To allow this to happen, a number of switching transistors need to be included around the memory element itself, so the high density of EEPROM is lost. Generally, EEPROM can be written to and erased on a byte-by-byte basis. This makes it especially useful for storing single items of data. Both writing and erasing take finite time, up to several milliseconds, although a read can be accomplished at normal semiconductor memory access times [11].

2.8 Printed Circuit Board

This design is one of the last stages of development of circuit after the testing, troubleshooting and result of circuit passes the requirements. A Printed Circuit Board (PCB) design is also known as a printed wiring board. PCB has copper tracks connecting the holes through where the components are placed. It basically serves two purposes; it places the mounted components and provides the electrical connections between the components. The fabrication of printed circuit board is achieved by an etching process based on the Gerber file created.

CHAPTER 3
METHODOLOGY

Methodology is one of the important parts of the project development. We will explain the procedure identification process, the tools that will be used and also the proposed work overview. The process of the project development is segregated into few parts within two semester's time frame. In general, the process comprises the integration of hardware and software.

3.1 Procedure Identification

The procedures involved in Wireless Sensor Data Logger System Design are basically based on the overall block diagram illustrated in Figure 1 from Chapter 1. The procedures are identified to ensure that the project can be accomplished within the time frame provided. From the block diagram, the specific flow chart is illustrated in Figures 11 and 12 for FYP1 and FYP2 respectively. The first flow chart shows the procedure for the targeted work to be accomplished during FYP1, while the second flow chart shows the procedure work for FYP2. Generally, the process starts with some research of literature review and knowledge about the project such as the microcontroller features, serial communications, and wireless communications. By identifying their functions, we can start designing the circuit part by part.

For the first step, we plan to display the sensor data through the LCD display and integrate the sensor data with the microcontroller. Then, further research about the external EEPROM chip is done in order to connect the chip with the microcontroller. The serial interface allows communication between microcontroller and PC via serial port. The radio frequency module is integrated to the microcontroller to conduct a wireless data transmission. The last stage of system design is the design of PCB and GUI in order to make the project more presentable.

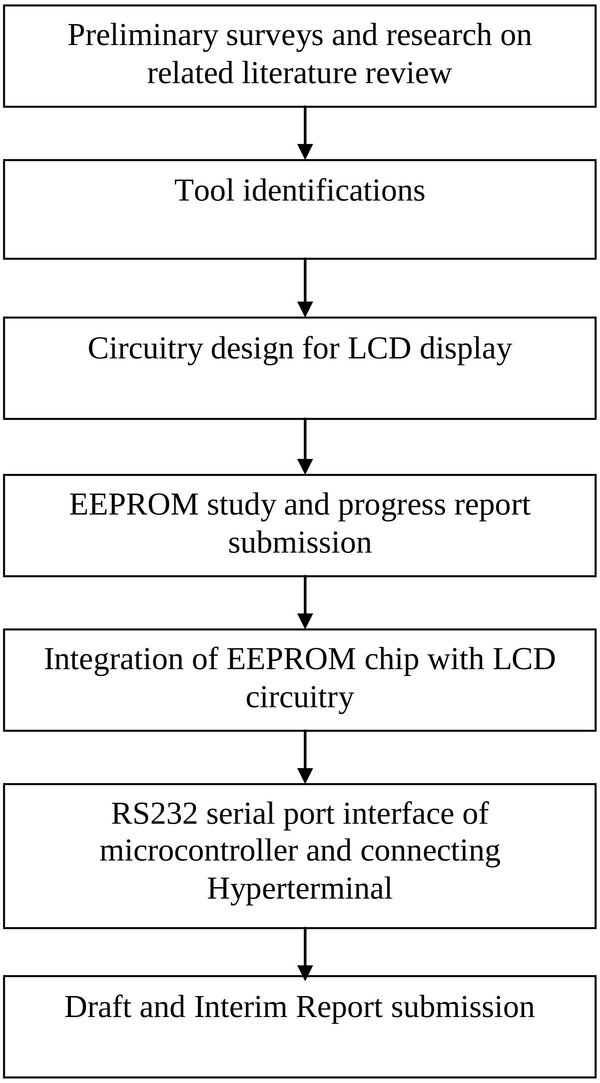

Figure 11: Procedure identification flow for FYP1.

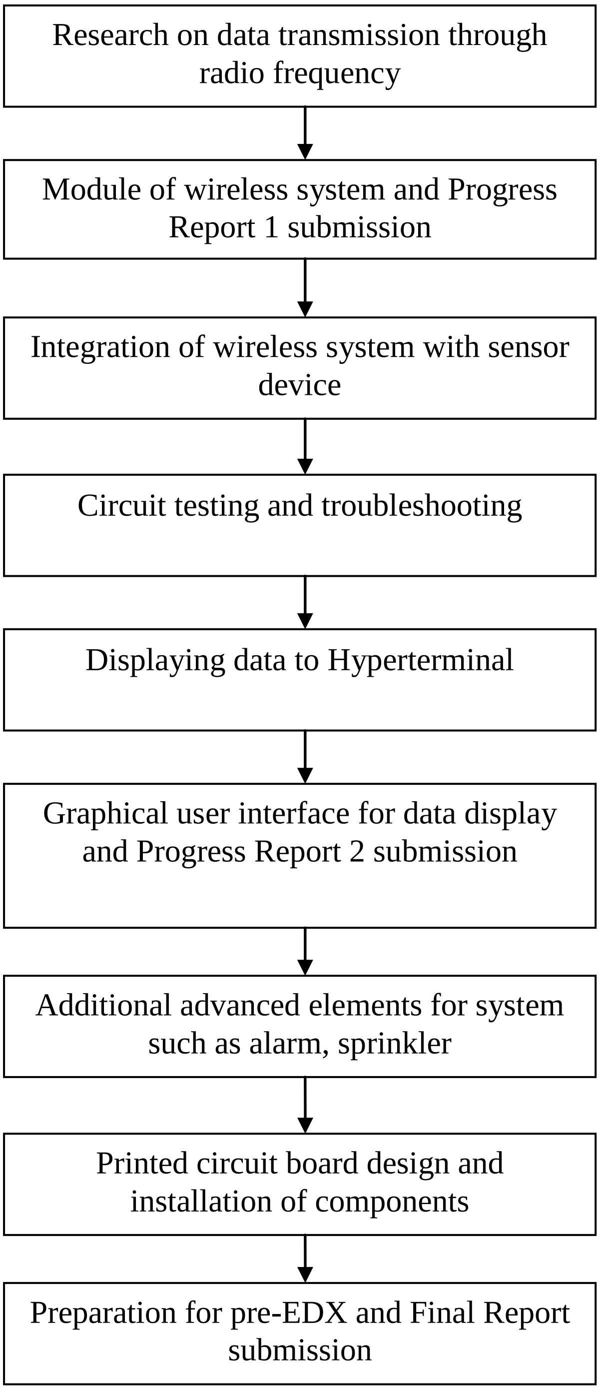

Figure 12: Flow procedure for FYP2.

3.2 Tools and Equipment Required

Tools play important roles in developing this data logger system. Since the project will be involving the software and hardware integration, both of software and hardware development tools are required. The tools that are proposed are the common software which mostly is widely used in electronic industries.

3.2.1 Software development tools

- MPLAB IDE and PICKIT 3
- CCS compiler
- ALTIUM Summer Designer - PROTEL
- Microsoft Visual Studio C++
- REALTERM

3.2.2 Hardware development tools

- Microcontroller PIC16F877A
- Humidity and temperature sensor
- Liquid crystal display (LCD)
- External EEPROM 24L256 chip
- MAX 232 level converter IC
- RS232 /RS485 serial port
- Personal computer

3.3 Proposed Work for Prototype Installation

The flow of installation for the prototype includes many crucial processes that should be done during the time frame given. Microcontroller theories are being applied to the project and new knowledge is discovered.

3.3.1 Clock oscillator calculation for PIC16F877A

For the microcontroller to operate, a clock is required to give a clock cycle. We use the crystal/ceramic timing devices that can be connected to the microcontroller through oscillator port denoted by OSC1 and OSC2. This timing device consists of a crystal oscillator plus two small capacitors. An instruction is executed by fetching it from the memory and then decoding it. This usually takes several clock cycles and is known as instruction cycle. The calculation for capacitors and crystal component are as follows:

Cp: Parasitic Capacitance, usually about 8pF

Ca: Actual value of capacitor, the capacitor used is 15pF

(Ca + Cp)/2 = (15p + 8p)/2 = 11.5

Therefore, the crystal that we will be using is 12 MHz.

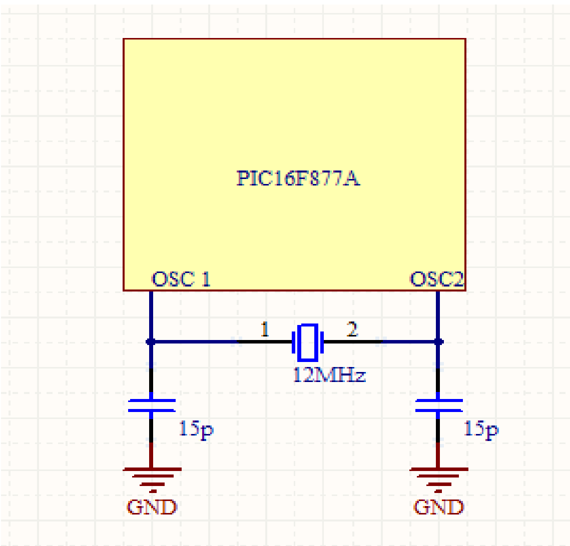

Figure 13: Schematic diagram for clock oscillator with PIC16F877A.

3.3.2 Programming development process

This flow diagram describes the basic process of developing a program using the microcontroller.

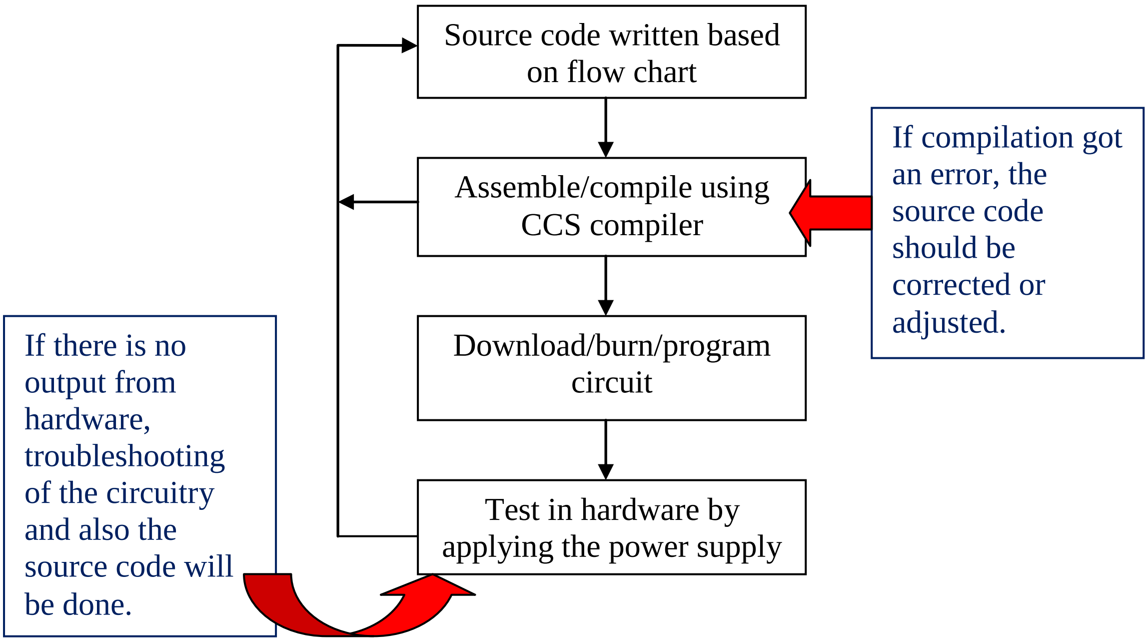

Figure 14: Flow diagram for programming development process.

3.3.3 Software programming installation

A circuit will not work without the microcontroller being programmed. Therefore, we learn and discovered the MPLAB IDE software together with the CCS compiler to create the program for digital alarm clock design. This software design can actually be created by various types of programming software. But we preferred using CCS compiler because the wizard and built-in functions in CCS compiler make it easy to create the basic settings based on the hardware device. They are also user friendly. The files are then imported to MPLAB IDE in order to use the PICKIT3 device to burn the program into the microcontroller chip.

3.3.4 Initialization mode of MCU chip

The initialization setting for the circuit is created by using the PIC Wizard in CCS Compiler.The screenshot for the initilazation of the microcontroller chip is depicted in Figure 15.

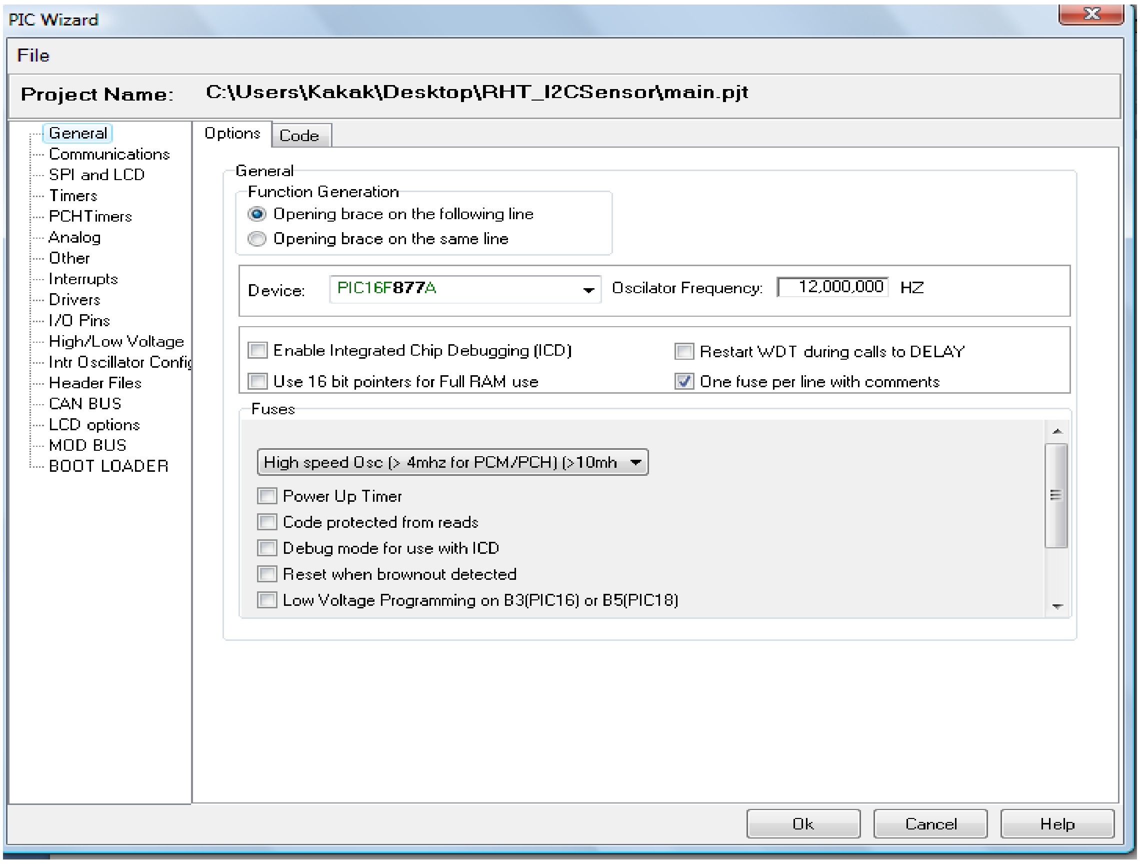

Figure 15: Initializing the microcontroller chip using the PIC Wizard.

The oscillator frequency used is about 12 MHz. For slower execution, a lower value of oscillator frequency can be used.The files included together in DigiPicco.c:

- Header file > **DigiPicco.h**
- LCD driver > **LCD.c**
- String library > **string.h**
- Standard Library > **stdlib.h**

Figure 16 below illustrates a sample C code for PIC16F877A initialization.

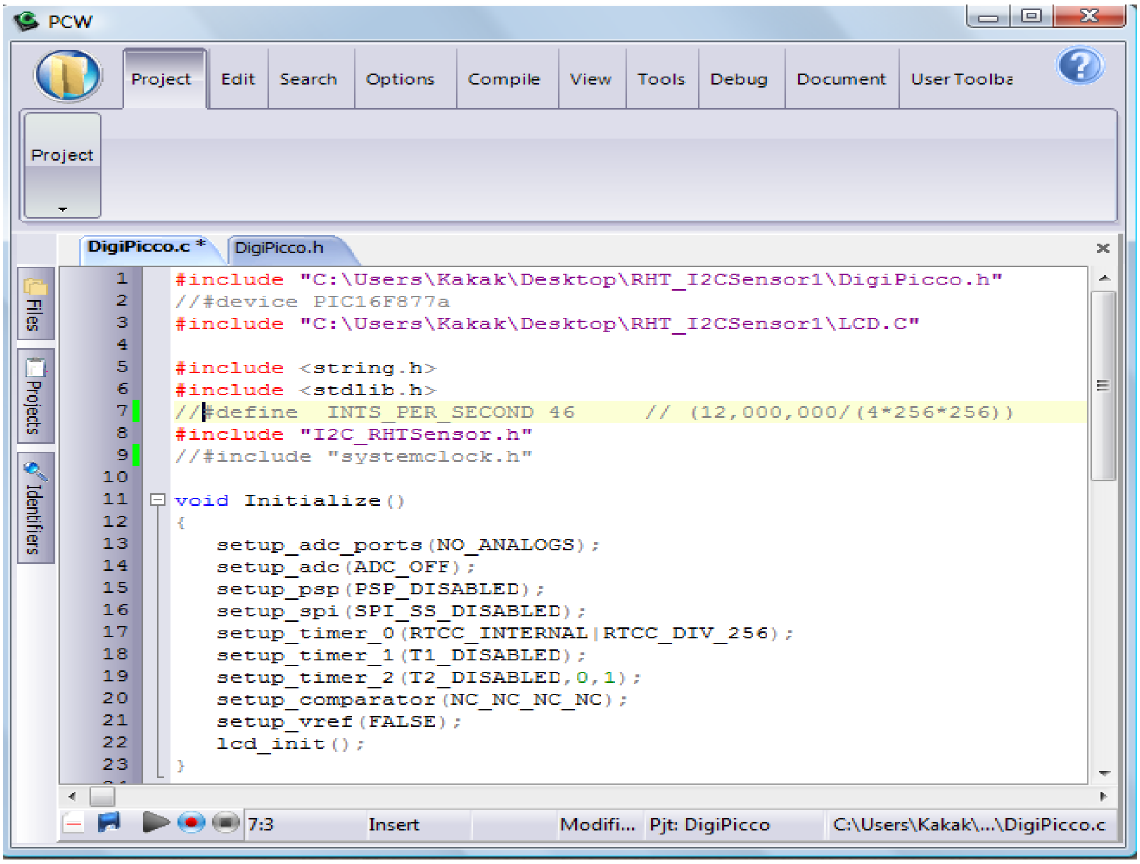

Figure 16: C programming code for initialization of PIC16F877A.

3.3.5 The interface for humidity and temperature sensor

The interface of humidity sensor to PIC16F877A based on the humidity and temperature sensor datasheet is given in Figure 17 below.

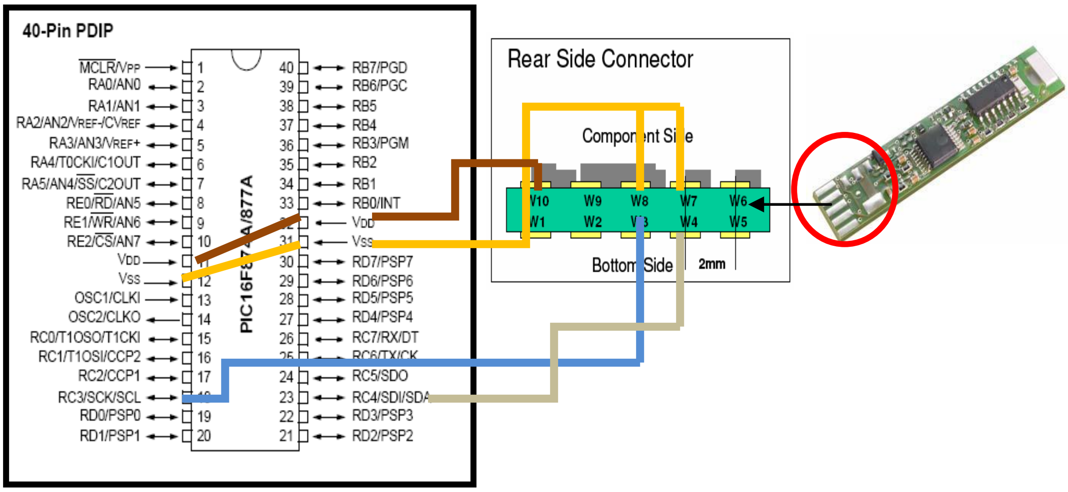

Figure 17: Connection of humidity sensor with PIC16F877A.

3.3.6 Interfacing 4-bit LCD

The first step to display output from LCD would require a sequence of program to be created. But in CCS compiler, a LCD driver has already created the sequence of the program. The following pins are the pin connections from LCD to microcontroller and these have been fixed by the driver. Only 4 bits are used for the LCD interfaces which are D7 to D4. It saves the use of other pins but responds slower than an 8 bit interface. Figure 18 below shows a LCD driver from CCS compiler.

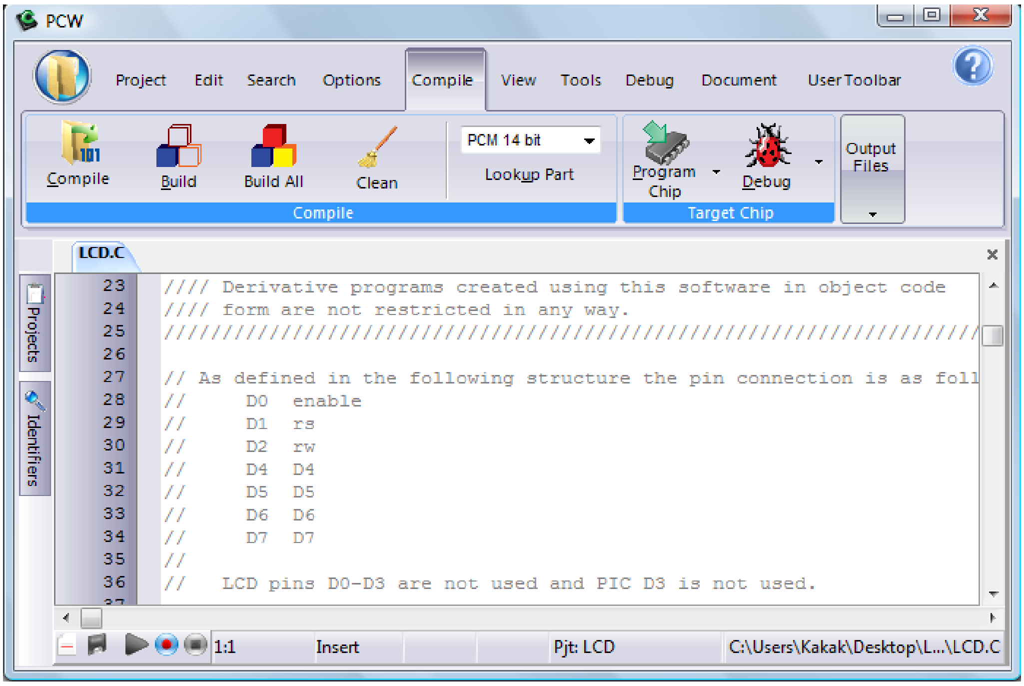

Figure 18: LCD driver from CCS compiler.

From the program code of LCD in **APPENDIX C,** the lcd_putc module is used to display a text or number on the LCD. The program below displays the value of relative humidity in percentage.

```
// The formula to convert hexadecimal of humidity and temperature into decimal value.
   Humidity = make32(0,0,HumH,HumL)*100/0x7FFF;
   Temperature = make32(0,0,TempH,TempL)*165/0x7FFF-40;
// lcd_putc module from driver is used to display value of humidity
   printf(lcd_putc,"\fR.Humidity=%LU%%\n",Humidity);
   printf(lcd_putc,"Temperature=%Lu C\n",Temperature);
```

3.3.7 Experimenting the accuracies of sensor

The accuracy of sensor was tested for both, the temperature and the humidity. Its accuracies can be detected by comparing the measured value with the actual value from other sources. The experiment took place at the Chemical Lab where the Isotech Model Jupiter 650 of constant temperature bath was used as the heating device. The temperature reading of sensor is compared with the master standard units of a digital thermometer. The data was recorded starting from 30 degree Celsius of temperature bath to 70 degrees of temperature bath. Figure 19 below shows the setup for the temperature heating experiment. Both sensors and master standard unit device were put into the space in the Isotech Jupiter machine. Here the surroundings inside the machine acted as the heating element. The values of both measured devices were recorded in a table and displayed in a plotted graph. The accuracy test for humidity is conducted by comparing the sensor with the hygrometer and the anemometer device. Both are humidity measurement devices that are basically used in industry applications.

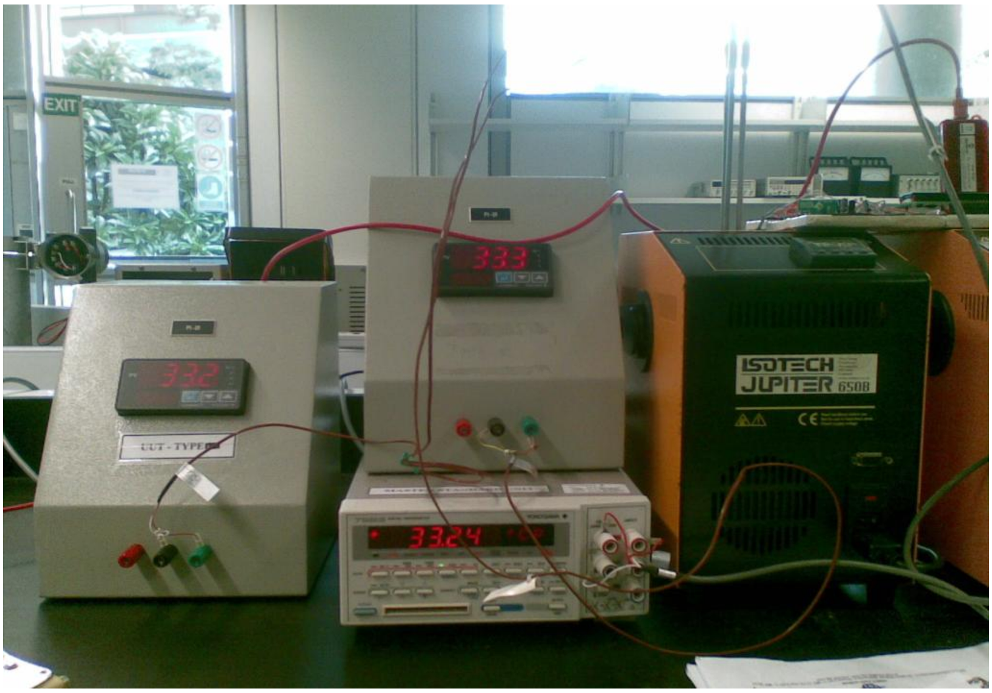

Figure 19: Setup of temperature measurement experiment.

The whirling hygrometer determines the percentage relative humidity (RH) by measuring the evaporation of water into the surrounding air. Two thermometers are placed in flowing air; one thermometer bulb was covered by a wet wick. The RH can be read off the slide rule calculator integrated into the hygrometer. To take the measurement of humidity, the instrument was opened by withdrawing the inner frame from the case. Then, thoroughly the wick was wet by placing the exposed end under cold running water or immersing it in water for about 30 seconds. This would wet both the exposed wick and that coiled in the wick container. The frame was rotated for 30 to 60 seconds at between 2 and 3 revolutions per second. When the hygrometer is closed the slide rule can be used to calculate the relative humidity percentage directly from the wet and dry temperatures. The calculator has two scales; the upper scale should be used for dry bulb temperature up to 20 degree Celsius. For higher temperature, the lower scale should be used. The steps to read the humidity reading of hygrometer, first, locate the wet bulb temperature on the relevant scale. The dry bulb temperature is aligned with the wet bulb. The reading of relative humidity is read from the centre scale at the location of the arrow.

Figure 20: The humidity measuring device; hygrometer and anemometer.

The anemometer is a digital humidity measurement device. This is straight forward compared to hygrometers because the reading of humidity is displayed directly from the anemometer. The devices measured the humidity of the surroundings from 8 am in the morning to 8 pm. The data measurements of sensor between devices are compared and plotted in a bar chart.

3.3.8 Serial Communication Interface

The serial communication allows the communication between the microcontroller and PC. The sensor data should be sent through this serial and displayed in the terminal called Realterm. The pin outs for serial communication is connected as illustrated in Figure 21 below. Two of input output pins of PIC16F877A microcontroller were configured as transmit pin and receive pin and connected directly to MAX232 level converter IC at pin 11 and 12, respectively. The MAX232 level converter IC converts microcontroller signal level 0V and +5V to +3V/+12V from a single supply of 5V. This is due to the fact that PIC16F877A microcontroller sends data serially in logic level of 0V for low logic, and +5V for high logic. However, RS232 serial port uses different logic levels, +3V and +12V for communication. Therefore, MAX232 converts the TTL logic level during data transmission.

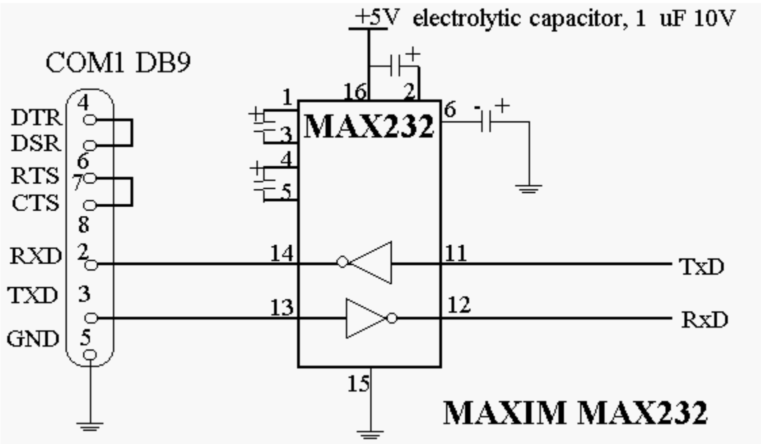

Figure 21: MAX 232 interface layout.

In order to allow the communication, several initializations are required to be executed. The initialization properties are set in the realterm as follows:

- Baud rate (bits per second): 2400
- Data bits: 8
- Parity bit: None
- Stop Bits: 1

3.3.9 External EEPROM Access

The closed up view for pin configuration and connections of EEPROM depicted in **APPENDIX B** is shown in Figure 22.The pin A0 is connected to 5V to get logic 1.Therefore, its address does not clash with the sensor's address that shares the same pin in the MCU circuitry. It is important for us to make sure different addresses are used between devices that share the same port of the MCU in order for them to function properly and be recognized. The EEPROM driver of 24LC256 is included from CCS Compiler library which is shown in **APPENDIX C**. As we see in program code number 5, the 24LC256 EEPROM library shows that both write and read operations follow the I2C protocol. In I2C protocol, the master initiates communications on the bus and controls the bus with one or more slave devices. Basically, it begins with a start condition and ends up with a stop condition.

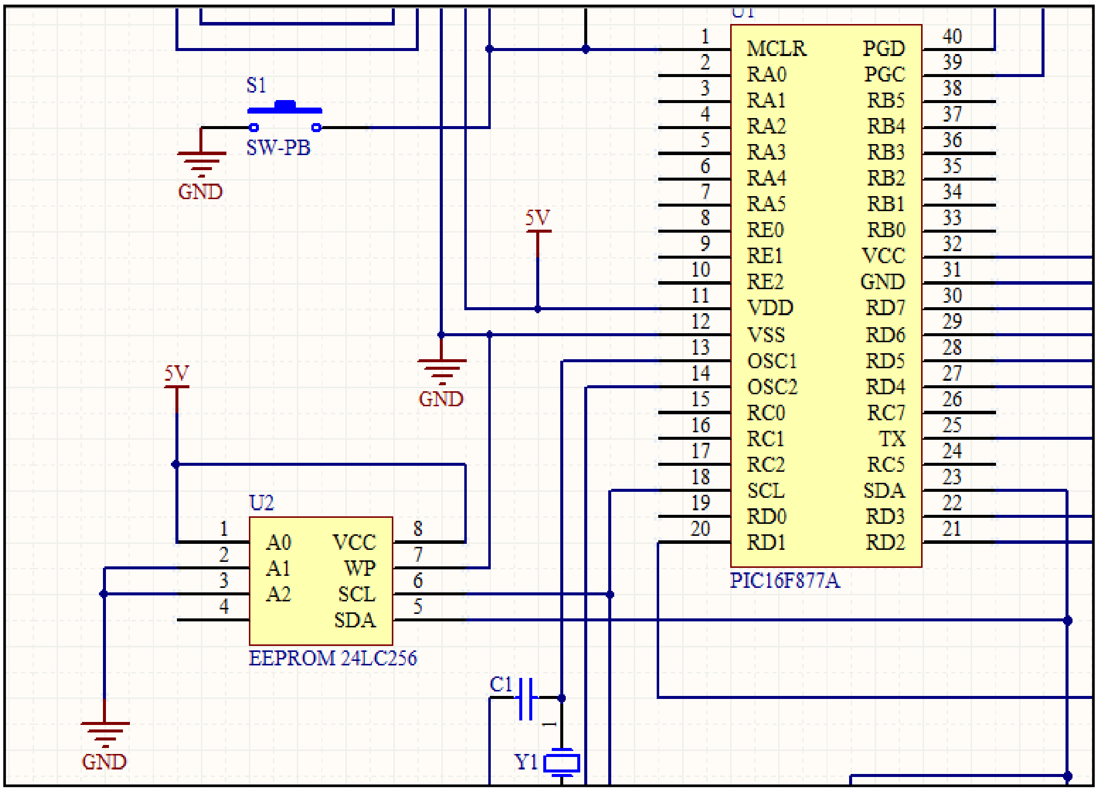

Figure 22: Close up view of EEPROM connections in the MCU circuitry.

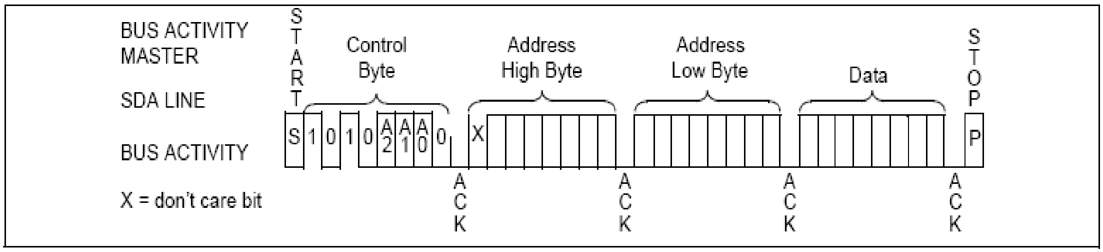

Figure 23: Write operation mode for 24LC256 EEPROM.

In Figure 23, the start condition is generated first. This figure relates with the code of the write operation where the slave actually writes the device address 0XA2 to get 1010 0010. From Figure 23, the first 4 bits refers to the control byte of EEPROM device for both write and read operation. The next three bits are the chip select bits A0, A1 and A2. These chip select bits depends on the user whether to use chip select A0, A1 or A2. The selected chip is connected to high logic to tell that we are using that chip select bits. In read operation, the $R/\overline{W}$ bit should be 0. From the program, it shows that A0 chip select was set to 5V. As shown in the datasheets in **APPENDIX E**, there are page-write and byte-write operations. For a byte write, one byte of data transfer taken place from the MCU to the EEPROM; the transfer is then acknowledged by the EEPROM. While in page write, data transfers can allow up to 16 bytes. The master generates a stop condition when everything was completed. A sample code to achieve a byte-write is shown below.

```
void write_ext_eeprom(long int address, BYTE data)
{
   short int status;
   i2c_start();
   i2c_write(0xa2);//a0=1
   i2c_write(address>>8);
   i2c_write(address);
   i2c_write(data);
   i2c_stop();
   i2c_start();
   status=i2c_write(0xa2);
}
```

Moreover, Figure 24 below depicts the EEPROM read operation.

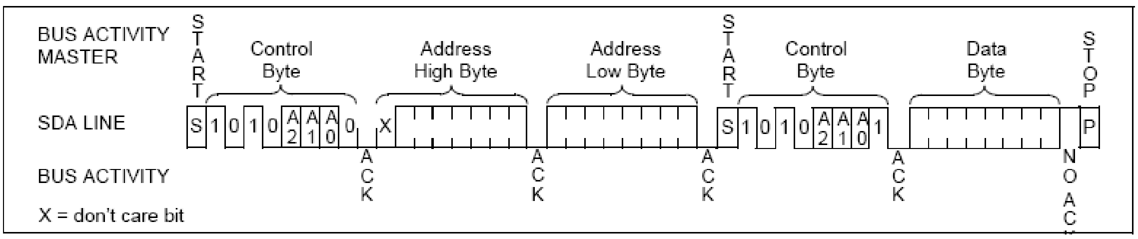

Figure 24: Read operation mode for 24LC256 EEPROM.

In the read mode, the $R/\overline{W}$ should be 1 to indicate it is in the read mode. The following sample code shows that the program starts with a start (i.e.; i2c_start( )) and ends with a stop (i2c_stop( )) conditions.

```
BYTE read_ext_eeprom(long int address) {
   BYTE data;
   i2c_start();
   i2c_write(0xa2);//a0=1
   i2c_write(address>>8);
   i2c_write(address);
   i2c_start();
   i2c_write(0xa3);
   data=i2c_read(0);
   i2c_stop();
   return(data);
}
```

In order to test whether or not the functionality of the EEPROM device works, or we used the input.c driver and included it in the main code. This input.c driver allows the user to kick in the data that he/she wants to save into one of the locations in the EEPROM device using a keyboard, and also allows the user to read back the value from the location. The program below explains that when the letter ‘R’ is received, the value from the EEPROM is read; while the letter ‘W’ indicates that the user wants to write a value to be saved in any location of the EEPROM.

```
#include "input.c

void main() {

   BYTE value, cmd;
   EEPROM_ADDRESS address;
   Initialize();
   printf("\r\nWelcome and Hye Noorshafrina\r\n");
   init_ext_eeprom();

   do {
      do {
         printf("\r\nRead or Write: ");
         cmd=getc();
         cmd=toupper(cmd);
         putc(cmd);
      } while ( (cmd!='R') && (cmd!='W') );

      printf("\n\rLocation: ");
```

3.4.0 Graphical User Interface Design

We had used the Microsoft Foundation Class wizard to create a new project workspace for the graphical user interface design. The location of project can be changed to a location that we want to save. In this project, the Win32 platform which refers to the recent versions of the windows operating system that runs in a 32-bit mode is used. This application generates an application that has built in functionality which when compiled; it implements the basic features of windows executable application. The Microsoft Foundation Class wizard is depicted in Figure 25.

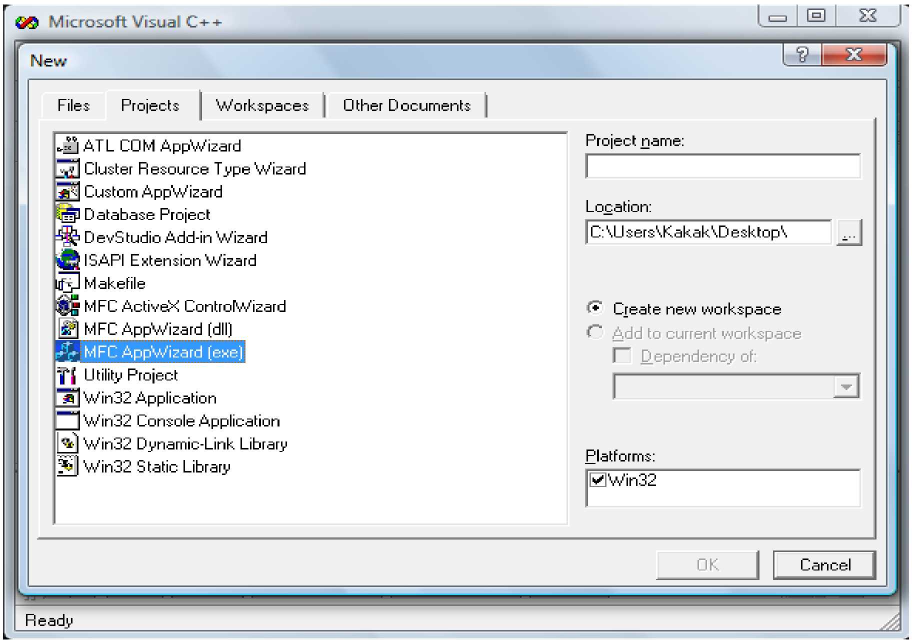

Figure 25: MFC wizard application in Microsoft C++.

A dialog-based GUI is implemented and created using a text file called a resource file which has file extension ".rc". This dialog box is a window that holds other windows controls and can be referred as a container. It is actually the primary interface that involves interaction between the user and computer. We can design the dialog box by selecting the boxes and placing them in the worksheet. But the variable for each of these boxes must be assigned because the user must program these variables according to their desired functions.

The dialog based type is selected as in Figure 26 below.

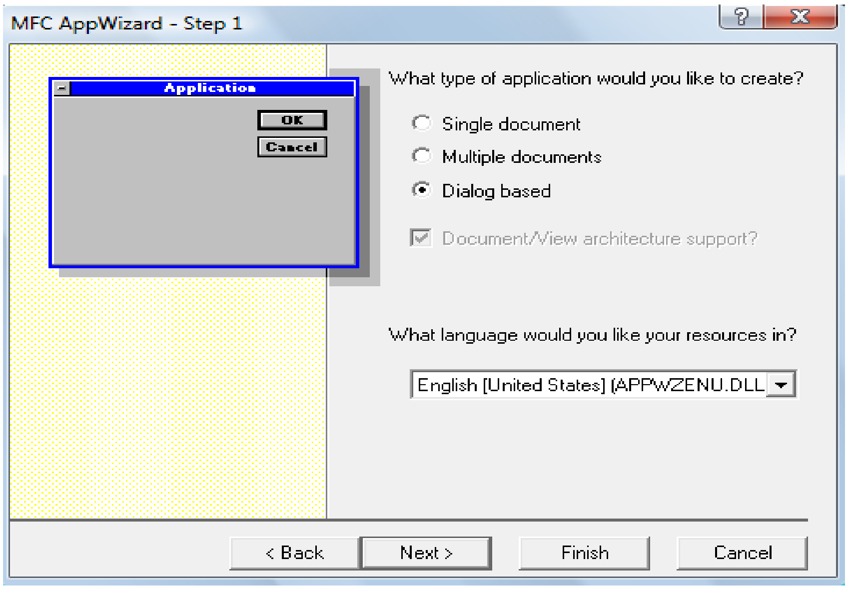

Figure 26: Screenshot for step 1 of GUI design.

The title of the dialog box for this project was entered as ‘Wireless Sensor Data Logger System Design’. Figure 27 shows step 2 for MFC wizard application.

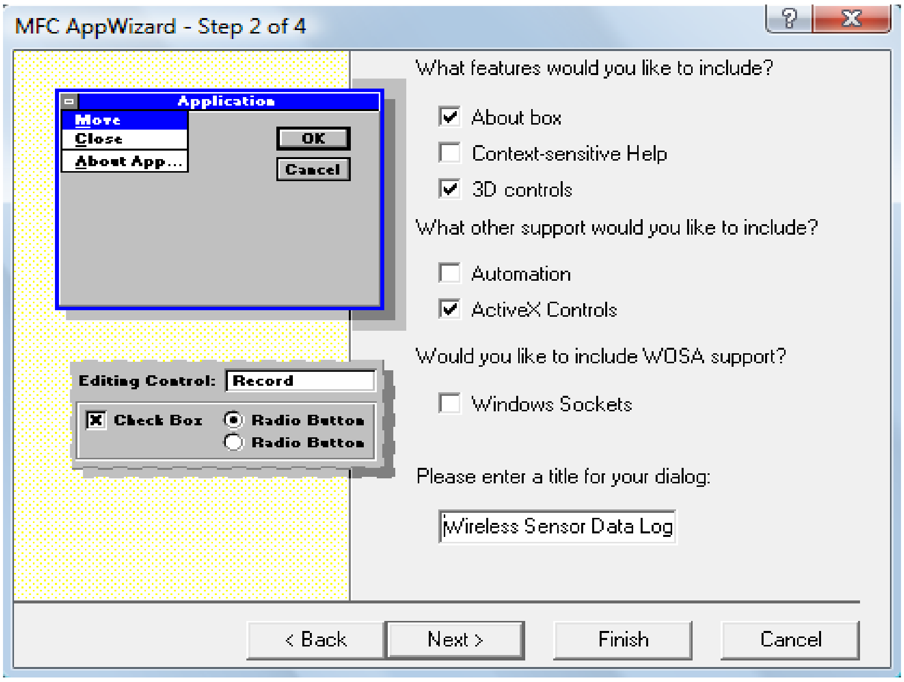

Figure 27: Screenshot for step 2 of GUI design.

Source code files will automatically be generated and any code modification can be made easily; Figure 28 below illustrates the options to generate such a source file.

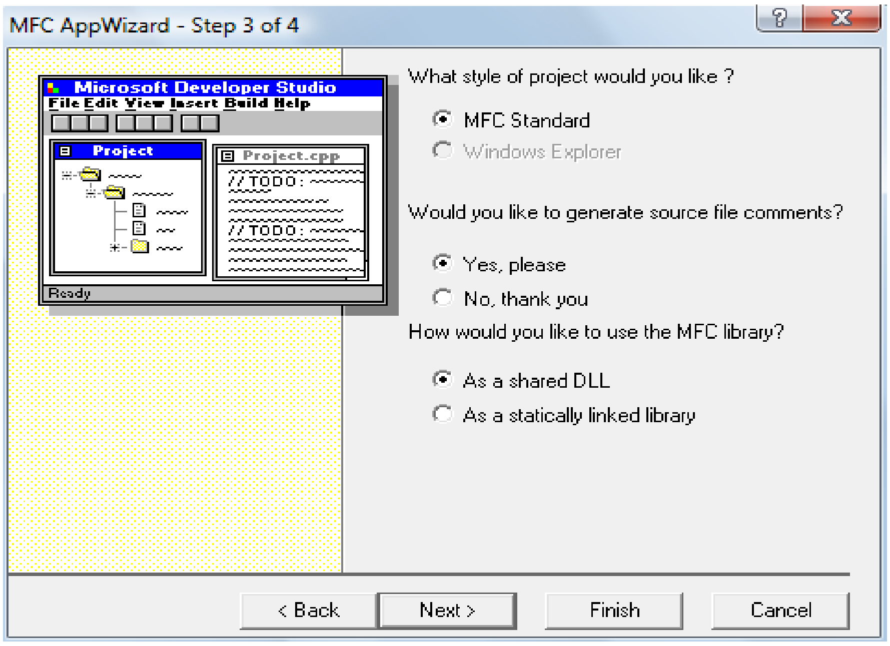

Figure 28: Screenshot for step 3 of GUI design.

Figure 29 below shows the design of graphical user interface with buttons and empty boxes which are arranged according to our desired functions. The 'close comm' and 'open comm' buttons allow the used port to be recognized.

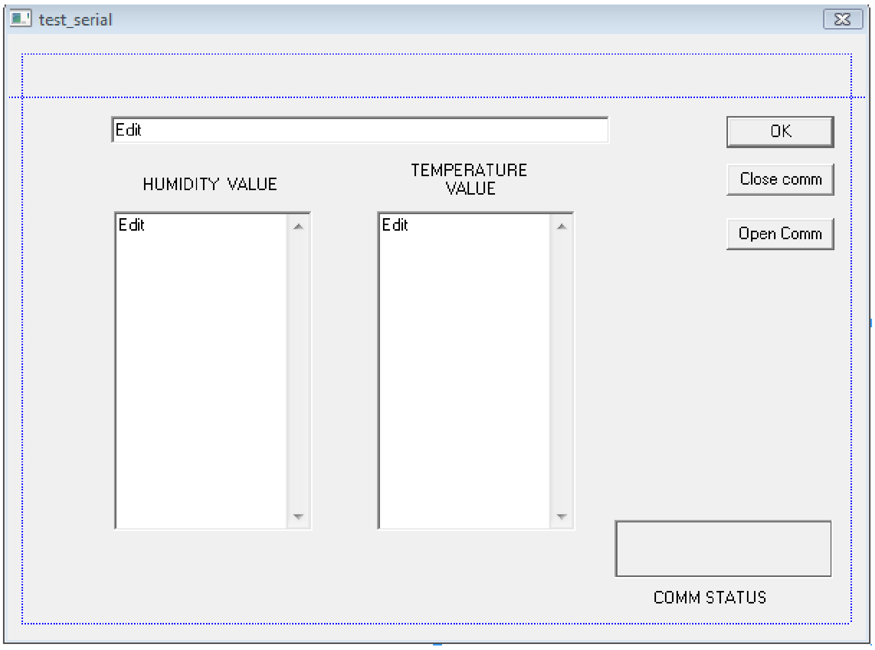

Figure 29: The view design of GUI interface.

The variables can be assigned by clicking the right side of mouse on the respective button box and by selecting the properties. This can be shown in Figure 30 below:

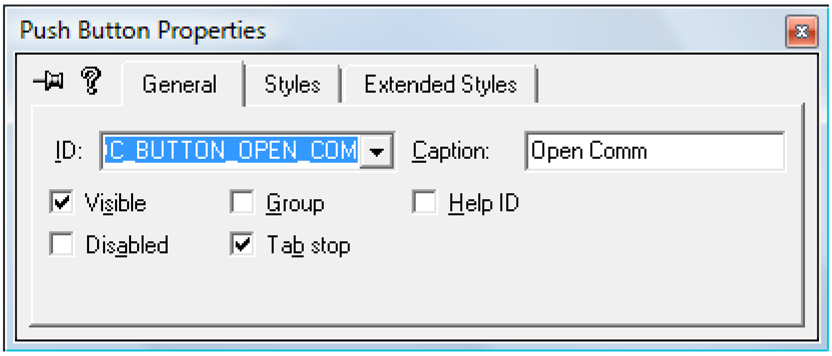

Figure 30: Properties for assigning variables to selected buttons.

A sample code below shows a situation where if the "open comm" button is pressed, the message of "comm. Port already open" will pop up.

```
void CTest_serialDlg::OnButtonOpenCom()
{

if(m_serial_flag)

{
        MessageBox("comm port already open") ;
        return  ;
}  }
```

The serial comm.unication used is 2400 baud rate with no parity bit and one stop bit

```
PortDCB.BaudRate = 2400;
PortDCB.fBinary = TRUE;
PortDCB.fParity = TRUE;
PortDCB.fOutxCtsFlow = FALSE;
PortDCB.fOutxDsrFlow = FALSE;
PortDCB.fDtrControl = DTR_CONTROL_ENABLE;

PortDCB.fAbortOnError = FALSE;

PortDCB.ByteSize = 8;
PortDCB.Parity = NOPARITY;
PortDCB.StopBits = ONESTOPBIT;
```

CHAPTER 4
RESULTS AND DISCUSSION

The results and findings obtained from the project are discussed thoroughly in this section. The process of project development involved masses of information and engineering principles. Apparently, the results presented in this chapter are the sensor data analysis through wireless and wired communication, PIC16F877A microcontroller circuitry and GUI interface. There are also other findings which will be explained further in the remaining sections of this chapter.

4.1 Sensor Output Display Test on LCD

From the first stage of circuit development, the measured output from the sensor is displayed by the LCD. Temperature and humidity data are sensed by the integrated sensor. The PICKIT3 connected through USB port supplies power to the circuit with 5 volts. The range of temperature that can be read by this sensor is around -25 to +85 degrees Celsius. In a normal condition, the room temperature should be around 27 degrees Celsius while the humidity is about 60 percents and above. The output display in Figure 31 shows the value of temperature and humidity on 1 November, 2010 at 1.36pm in room environment. Therefore, the measured data shows that the environment is in normal condition. Table 4 below shows the reading from the LCD display.

Table 4: LCD display of measured value from sensor.

SENSOR	MEASURED VALUE
HUMIDITY	67 %
TEMPERATURE	28 ℃

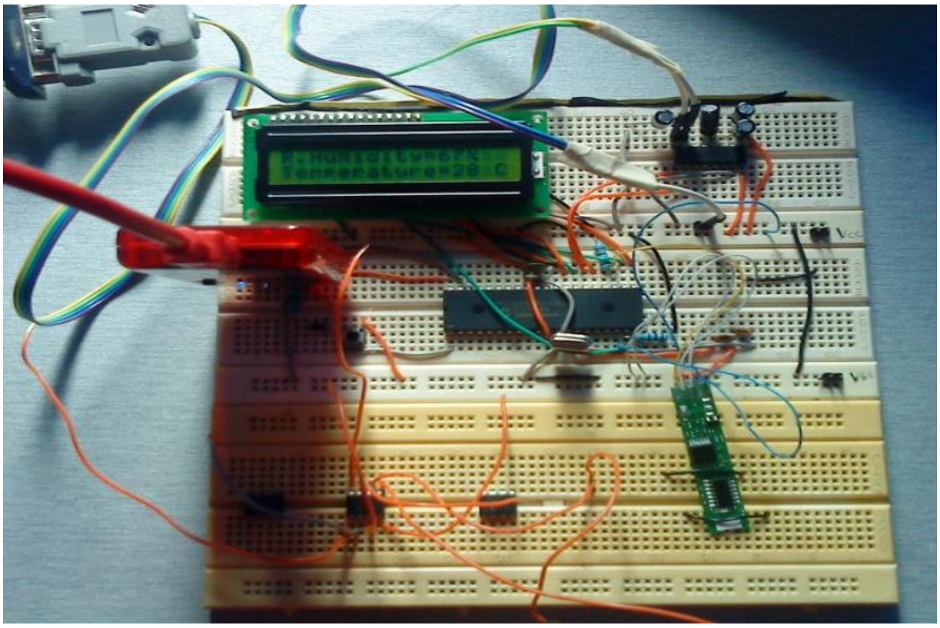

Figure 31: Output of temperature and humidity sensor data to LCD.

4.2 Accuracies of Sensor Board Compared to Other Devices

The accuracy test of the sensors is being run to compare them with standard measurements and to prove that they are accurate. By comparing these measurements with the standard reading, the accuracy level of the sensors could be determined. This is important since the sensors would be applied in real applications. The data analysis for the measurements of humidity and temperature are both tested in different methods. The method used has been already explained and discussed in the methodology section of the report.

4.2.1 Humidity data analysis

The humidity reading is compared with two different humidity devices, the hygrometer and the anemometer. These devices are common humidity measuring devices. The readings of the outside surrounding were taken from 8am in the morning until 8pm. Table 5 below shows the measured humidity data of the environment in normal weather, which is not too windy, not too hot and not raining.

Table 5: Measured value of humidity and temperature for measuring devices.

TIME	HYGROMETER	ANEMOMETER	SENSOR
8:00 AM	85	88	86
10:00 AM	72	77	75
12:00 AM	65	67	68
2:00 PM	71	68	70
4:00 PM	58	56	54
6:00 PM	62	64	66
8:00 PM	78	81	76

From the measured value, it shows that the reading of the integrated sensor is almost the same as the other two devices. The highest humidity percent is in the morning which is around 85 percents and above. This shows that the quantity of water in the air during 8am in the morning is high. While the lowest humidity percent is during 4pm in the evening. The weather at this time was quite hot; therefore the quantity of water is lower. The hygrometer and anemometer are common devices which gives the standard reading of the humidity in real life. Therefore, the integrated sensor is applicable in daily life applications and has an accurate humidity reading since its measured value is almost the same as the value of the devices. The result of humidity percent measured from the three devices is summarized in Figure 32.

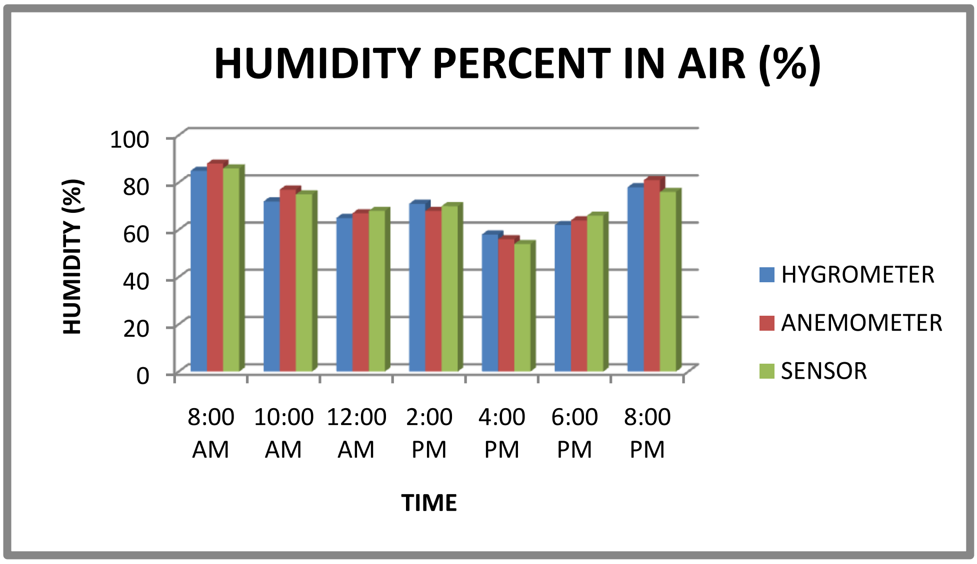

Figure 32: The measurements of humidity percentage in air.

4.2.2 Temperature data analysis

The accuracy of temperature is determined by comparing the measured data of the integrated sensor with the master sensor unit measurement. The bath temperature was used as a heating element. The temperature value of master standard unit (MSU) and value of sensor were recorded for each increasing value of bath temperature. The measurement was taken from 30 to 70 degree Celsius.

Table 6: The reading of temperature measured.

UNIT UNDER TEST (UUT)			SENSOR
MASTER STANDARD UNIT (MSU)			3 WIRE RESISTANCE THERMOMETER
NO	**BATH TEMP**	**MSU READING**	**UUT READING**
1	30	32.04	29
2	40	40.67	37
3	50	52.12	49
4	60	60.39	58
5	70	65.31	62

From the table, as the temperature bath increases, the measrurements of both MSU and sensors increase. The accuracy of the sensors are calculated based on the plotted graph using the standard deviation equation. A graph is plotted based on the recorded data shown in Figure 33 below.

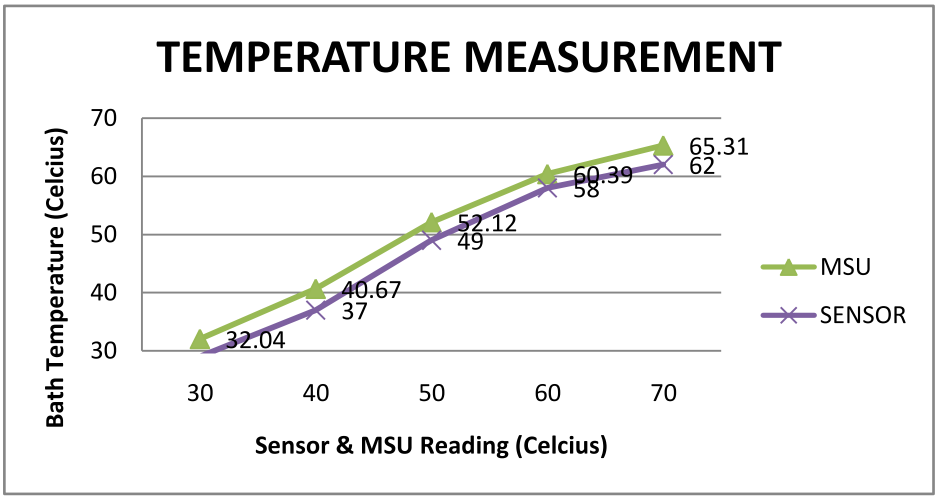

Figure 33: The measurements of temperature.

Using the standard deviation formula, the accuracy of the sensor can be determined as follows:

$$\sigma = \sqrt{\frac{\sum(x-\overline{x})^2}{N}}$$

σ = the standard deviation

x = each value in the population

$\bar{x}$ = the mean of the values

N = the number of values of population

The mean for sensor is calculated using:

$$\overline{x} = \frac{\sum x}{N}$$

$$= \frac{29 + 37+49+58+62}{5}$$

$$= 47$$

Using the mean to calculate:

$$\sum(x-\overline{x})^2 = (29-47)^2 + (37-47)^2 + (49-47)^2 + (58-47)^2 + (62-47)^2$$

$$= 774$$

Therefore, the standard deviation of sensor is around:

$\sigma = \sqrt{774/5} = 12.44$

The mean for MSU is calculated using:

$$\overline{x} = \frac{\sum x}{N}$$

$$= \frac{30.04 + 40.67 +52.12 +60.39+65.31}{5}$$

$$= 49.71$$

Using the mean to calculate:

$$\sum(x-\overline{x})^2 = (30.04-49.71)^2 + (40.67-49.71)^2 + (52.12-49.71)^2 + (60.39-49.71)^2 + (65.31-49.71)^2$$

$$= 831.86$$

Therefore, the standard deviation of sensor is around:

$$\sigma = \sqrt{831.86/5} = 12.9$$

From the calculation, it proves that the sensor gives an accurate temperature reading because the value of standard deviation of sensor is near with the value of the MSU standard deviation. Therefore, the sensor is applicable in industry applications like other temperature measurement devices. But, the sensor is easily integrated to a circuit compared to other devices.

4.3 EEPROM Output Test on Terminal

At the end of the EEPROM program test, the program manages to come out with a final output from the compilation of EEPROM 24256 C programming. The compilation is completed successfully with no errors. This is proved by the CCS compilation window in Figure 34 below.

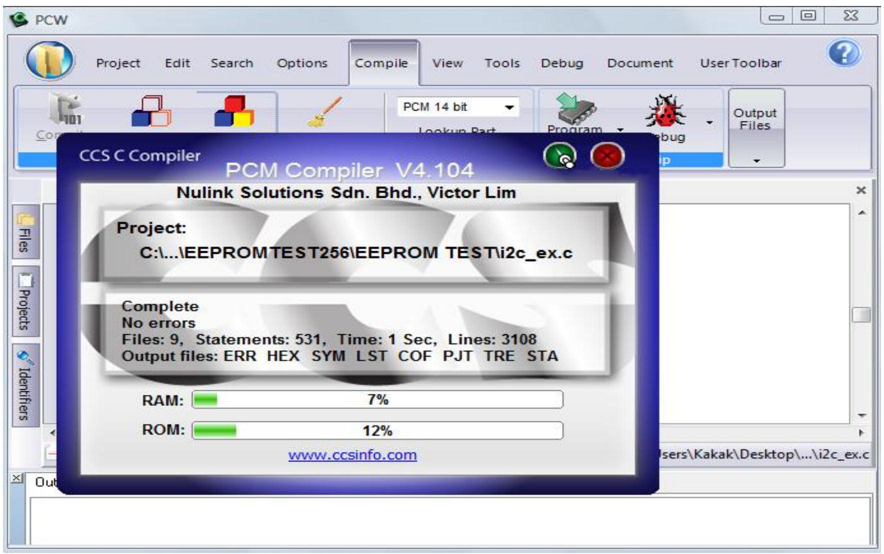

Figure 34: Compilation of EEPROM program.

The result illustrated in Figure 35 below shows that the value 45 is written by the user and saved in location 12. When the user wishes to read back the value from location 12, the result will show the value 45, which is the value that had been stored in location 12. This proves that the EEPROM chip program functions well.

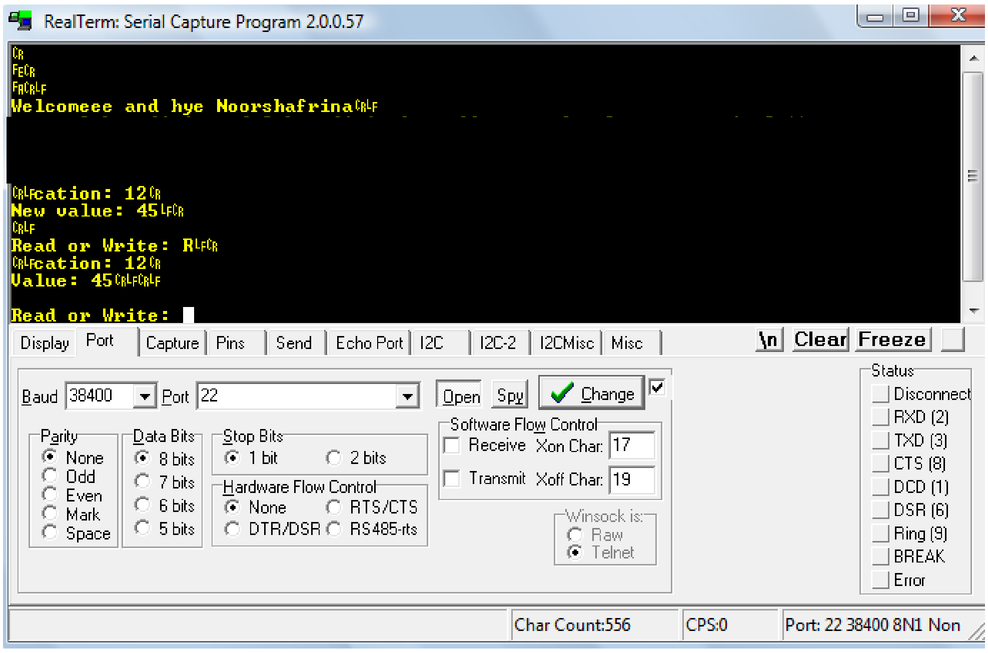

Figure 35: Result for external EEPROM programming with microcontroller.

4.4 Serial Communication Test on Terminal

Figure 36 below shows the result from RS232 communication in Realterm terminal.

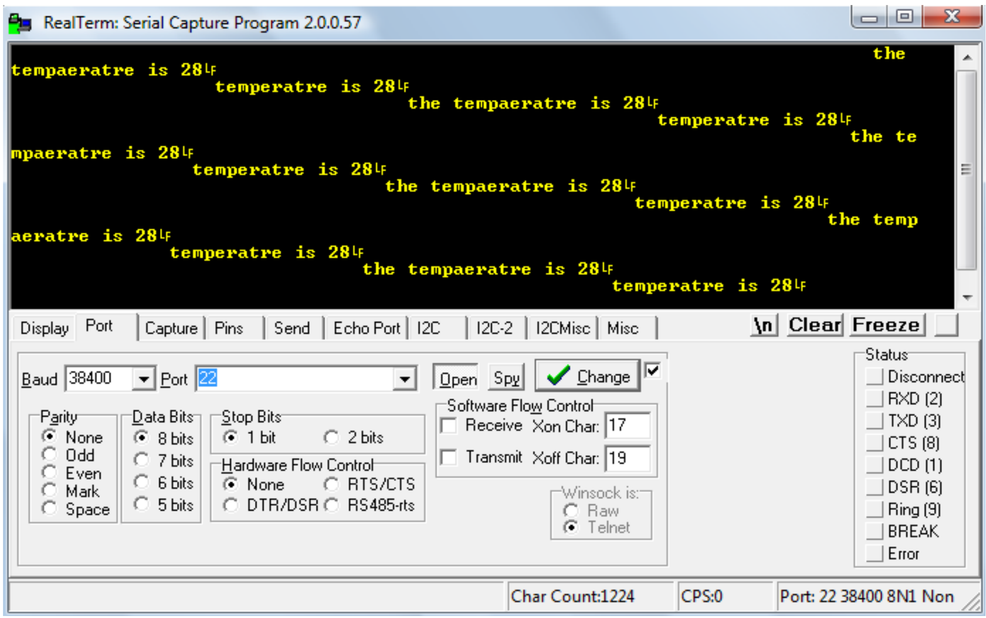

Figure 36: Result from RS232 communication in Realterm terminal.

This test is conducted in order to check the functionality of the serial communication before integrating the wireless communication device. From the program tested, the output display shows the temperature reading of the sensor module. From the result illustrated in Figure 36 above, it can be concluded that the serial port communicates with the PC and the value read by the sensors can be displayed in the RealTerm software.

4.4.1 Data Logger display through wired communication

The result from hard wired connection with PC using the serial port is the displayed data stored by EEPROM. From the program, the location for storing both data consists of 50 locations. The first 25 locations are the data of humidity, while the remaining locations are the data of temperature. The value from the locations are all the same because the time delay for each value to be read is 25micro seconds; therefore, it executes fast and the changes could not be easily detected during this duration of time. A faster delay time is purposely used in the experiment in order to avoid a slow response. In real applications, the exact delay that should be used can be changed in the program. Figure 37 and 38 below shows the reading of the stored data:

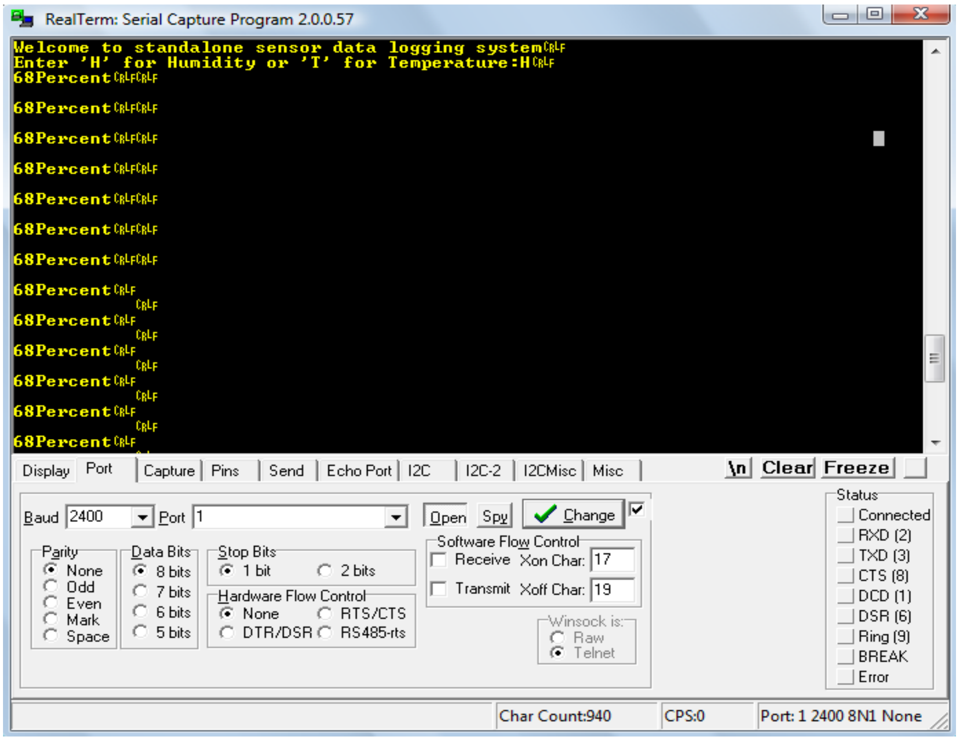

Figure 37: Display of 25 data of humidity from EEPROM.

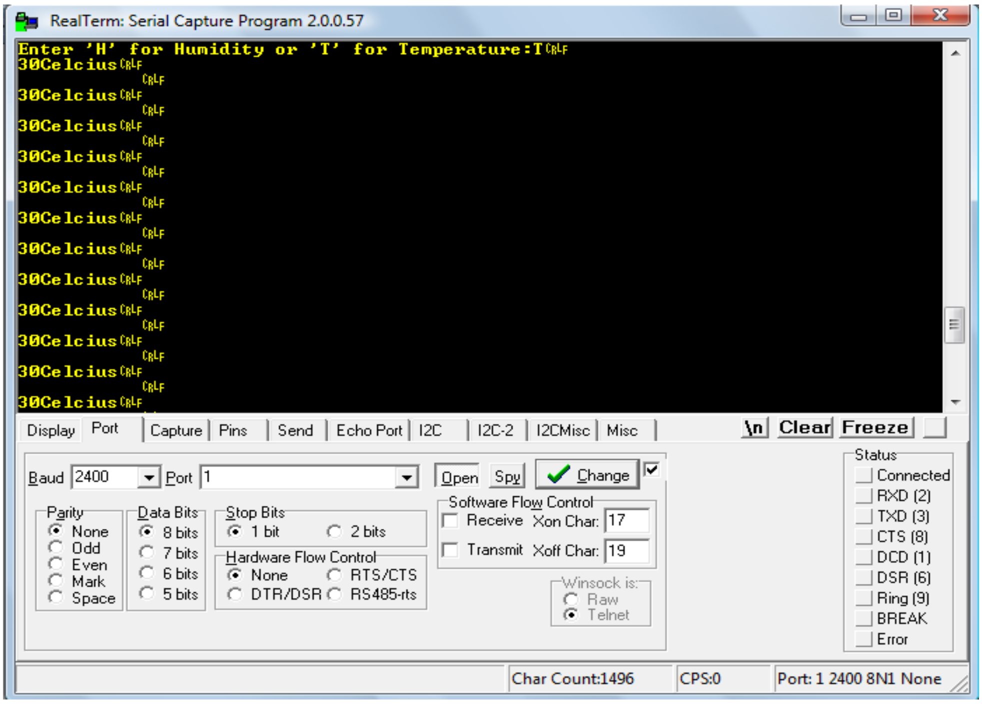

Figure 38: The display of 25 data of temperature from EEPROM.

4.4.2 Data logger display through wireless communication

Figure 39 below shows the data logger from a wireless communication using the radio frequency module. There are a lot of contaminations (garbage) due to noise and interference displayed in Realterm because of the instability of the RF wireless link. The terminal is basically a dummy terminal where it receives all the incoming data including the garbage.

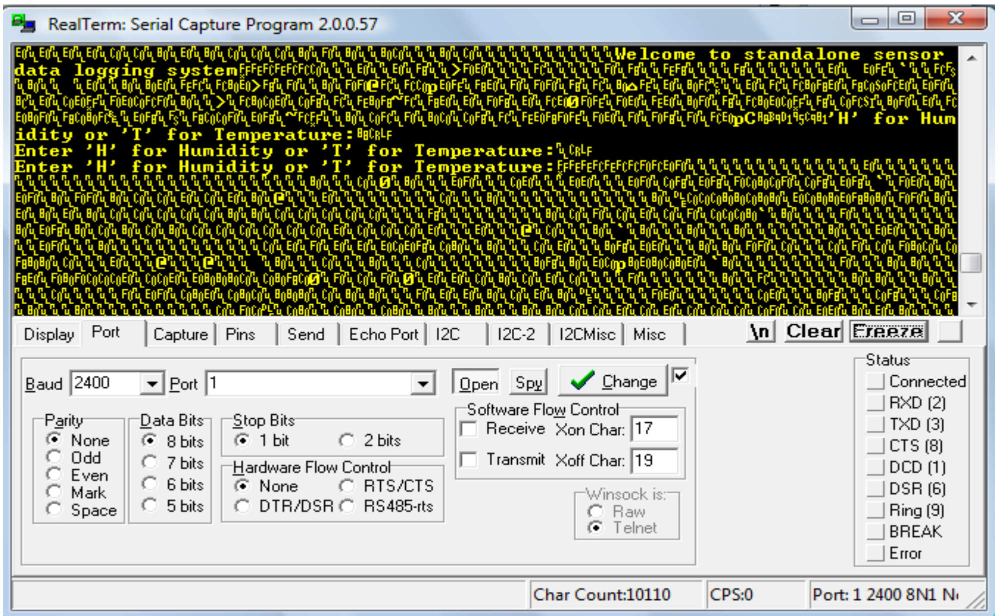

Figure 39: The display from wireless communication.

4.5 Graphical User Interface Display

As a result from the Figure 40, it shows that the data is displayed vertically. The amount of data sent are 50 data, 25 data for humidity values and another 25 values for temperature. We purposely programmed the EEPROM to read and record only 50 locations from the addresses in order to reduce the time of system development and prepare for other unfinished process. By pressing the “Open Comm” button, the “COMM STATUS” shows that the port has been opened and thread is started.

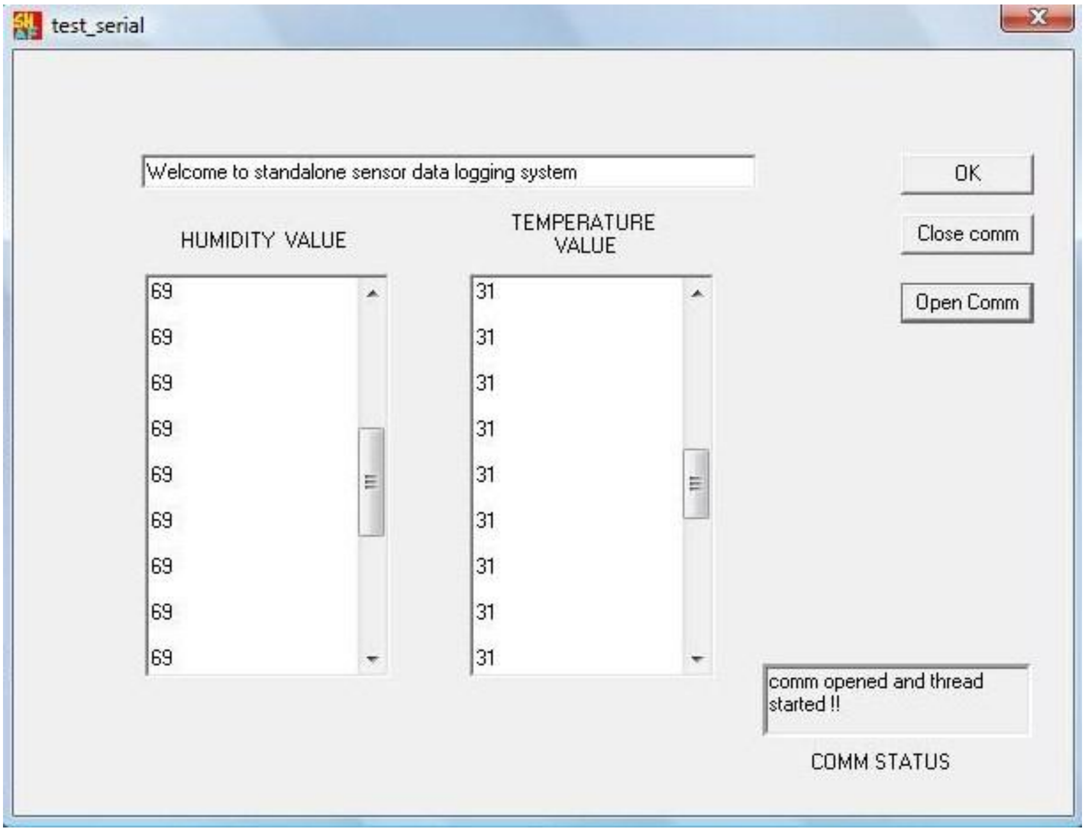

Figure 40: The serial communication through GUI.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

As a conclusion, at the end of project development process, the project basically meets the objectives and works appropriately as expected. However, there are some spaces or room of improvements for future enhancement. Conclusions and a few recommendations are explained in the section below.

5.1 Conclusion

In a nutshell, the Wireless Sensor Data Logger System proceeded as scheduled and has met its objectives. Through research, analytical and critical thinking, time management, planning and laboratory work; the objectives are met. From the objectives, the wireless communication between the integrated sensor MCU circuits with PC is communicating but the radio frequency module is not very stable and sends garbage to terminal. The printed circuit board has been fabricated and the GUI interface design has been working. The project gives a good practice to us in the embedded system knowledge and technical work which is a useful hands-on work in the future.

The project development was very tough and challenging due to time constraint and components availability. The project requires a frequent testing and troubleshooting processes which are very time consuming. The Wireless Sensor Data Logger system offers a reliable dynamic data logging system. Moreover, the project provides a platform for further advancement with better reliability and various applications in diverse fields.

5.2 Recommendations

In this section, recommendations will be made towards the Wireless Sensor Data Logger System. The recommendations are based on enhancement and improvement of the system besides reducing the mistakes that exists in the designed system. The recommendations for enhancement of the system will be beneficial especially when it deals with real applications. Thus, the recommendations are as below:

- **Implementing a recovery and security system that can accommodate the industries needs**

  A security system should be implemented together with the data logger system in order to respond to fault occurrences and be able to offer warning alarms and immediate corrective actions to the devices.

- **Improving the wireless communication by using a more stable radio frequency module or other better modules**

  The purpose of improving the wireless communication is to reduce or eliminate the garbage that is transferred to PC. A stabilized wireless link is required to make sure the correct data is transferred and sent to PC.

- **Implementing the time and date in the data logger design**

  For more advanced and systematic applications, the time and date should be implemented in order to record the data for the time and date required. Initially, the design of data logger system is designed without the date and time because of time limitations.

REFERENCES

[1] Enzine Articles, "*The importance of a data logger",* http://ezinearticles.com/?The-Importance-Of-A-DataLogger&id=4828095

[2] XZcution, "*For those who are curious about the data logger*", http://www.xzcution.com/for-those-who-are-curious-about-the-data-logger/

[3] Data Acquisition Networks, "*The Importance of data collection*", http://www.danmonitoring.com/data-loggers/the-importance-of-data collection.html

[4] USDA, "*The temperatures affect food*", http://www.fsis.usda.gov/factsheets/how_temperatures_affect_food/index.asp

[5] Post Harvest, "*Post Harvest Cooling/Storage for Cut flowers*",Extension Agriculture Engineer,University of Maryland http://www.bre.umd.edu/portacooler1.htm

[6] One omega, "*Introduction to data logging systems*", http://www.omega.com/techref/pdf/loggerintro.pdf

[7] Dogan Ibrahim.*Advanced PIC Microcontroller Projects in C,*The Newnes,Elsevier,2008

[8] Circuits Today, "*Introduction to PIC16877*", http://www.circuitstoday.com/pic-16f877-architecture-and-memory-organization

[9] Modtronix, "PIC16F877A", http://www.modtronix.com/product_info.php?products_id=29

[10] Myke Predko.Programming and Customizaing the PIC Microcontroller,TAB Electronics,Mc Graw Hill,3rd Edition,2008.

[11] Tim Wilmshurst.Designing embeded systems with PIC Microcontroller,The Newnes,Elsevier,1st Edition,2007.

[12] Rickeys World, "*LCD interfacing with microcontroller's tutorial*", http://www.8051projects.net/lcd-interfacing/introduction.php

[13]Wise Geek, "*What is humidity sensor"*,
http://www.wisegeek.com/what-is-a-humidity-sensor.htm

[14]Robot Electronics, "*Using the I2C Bus*",
http://www.robot-electronics.co.uk/acatalog/I2C_Tutorial.html

APPENDICES

APPENDIX A

GANT CHART

APPENDIX B
CIRCUIT DIAGRAM

APPENDIX C

PROGRAM CODES

APPENDIX D

PRINTED CIRCUIT BOARD

APPENDIX E

DATA SHEETS