

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: LOW COST PC BASED DATA LOGGING SYSTEM

Academic Session: MAY 2013

I LUI YONGJIAN

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

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

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

60A TAMAN BUNGA RAYA
06000, JITRA,
KEDAH DARUL AMAN

Mr. Lee Wai Kong
Supervisor's name

Date: 1 AUGUST 2013

Date: _____

LOW COST PC BASED DATA LOGGING SYSTEM

BY

LUI YONGJIAN

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

**Faculty of Information and Communication Technology
(Perak Campus)**

MAY 2013

DECLARATION OF ORIGINALITY

I declare that this report entitled “**LOW COST PC BASED DATA LOGGING SYSTEM**” is my own work except as cited in the reference. The report has not been accept for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : Lui YongJian

Date : 1 AUGUST 2013

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Mr. Lee Wai Kong. The supervision and support that he gave truly help the progression and smoothness of engage in a Low Cost PC based data logging system project. The co-operation is much indeed appreciated. I truly thank him for his support and guidance he gave me throughout the whole project. He has wide knowledge in this area and has been great in explaining information regards to this project. Without him, this project won't be a success

Finally, I also like to thanks to my parents and my family for their love, support and continuous encouragement throughout the course. Their supports have turned some difficulties into uncomplicated task for me to handle.

ABSTRACTS

Data Acquisition is a system to collect and gather raw data from real world. Mostly raw data collected by data acquisition are in analog form through real sensors. The raw data mostly were recorded by hand on or through strip chart recorders in the older age. In this information age, there are various type data loggers and data acquisition products have been designed in information technology. This is due to the implementation of multifunction in microcontroller which allows data loggers and data acquisition can be more advance and become high-end products in nowadays.

In this project, a low cost PC based data logger is developed to fulfill the needs of remote data collection and storage. The data logger works around in Microchip dsPIC30F3011 microprocessor with SKDS40A microcontroller. The low cost PC based data logger system is designed and developed to measure light intensity, resistive by bending, humidity and temperature by real sensors. The data collected will be display in GUI interface PC based data logging with real time data measurement and real time graph. In the end, the objective of this project had been achieved.

TABLE OF CONTENTS

TITLE	i
DECLARATION OF ORIGINALITY	ii
ACKNOWLEDGEMENTS	iii
ABSTRACTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	ix
LIST OF TABLES	xi
LIST OF ABBREVIATIONS.....	xii
CHAPTER 1 INTRODUCTION	1
1-1 Project Background.....	1
1-2 Motivation and Problem Statement.....	2
1-3 Project Scope.....	2
1-4 Project Objective.....	3
CHAPTER 2 LITERATURE REVIEW	4
CHAPTER 3 METHODOLOGY AND TOOLS	6
3-1 Methodology.....	6
3-2 Tools	8
3-2-1 Hardware.....	8

3-2-1-1 SKDS40A Microcontroller	8
3-2-1-2 dsPIC30F3011 microprocessor	10
3-2-2 Software	10
3-2-2-1 MPLAB IDE	11
3-2-2-2 PICkit 2 Programmer	11
CHAPTER 4 PROJECT SPECIFICATION	12
4-1 Planning	12
4-2 Design	13
4-2-1 Constructing data acquisition system.....	13
4-2-2 Collecting data from microcontroller through real sensors and send it to PC.....	13
4-2-3 Developing PC based software to display data and plot graphs	14
4-2-4 Generating a report which contains data collected and graphs through automation from PC based software	15
4-3 Verification Plan	16
CHAPTER 5 IMPLEMENTATION AND TESTING	32
5-1 Overall GUI Interface PC based Data Logging System	32
5-2 Data Logging for 1 Week	40
5-3 Discussion.....	43
CHAPTER 6 CONCLUSION	44

6-1 Problem Encounter.....	44
6-2 Future Work.....	45
REFERENCE	46
APPENDIX A HARDWARE SETUP DATA LOGGING SYSTEM	A-1
A-1 Data Acquisition System	A-1
A-2 Data Acquisition System with SKDS40A microcontroller....	A-2
APPENDIX B PC BASED DATA LOGGING SOFTWARE INTERFACE.....	B-1
B-1 Data Logging Software Interface.....	B-1
B-2 Automation Report Generation.....	B-2
APPENDIX C PROJECT FILES FUNCTION.....	C-1
C-1 Project File Description - MPLAB IDE.....	C-1
C-2 Project File Description – Microsoft Visual Basic	C-1
C-3 Project File Description – VBA in Word.....	C-2
APPENDIX D CODES FOR PROJECT FILES FUNCTION	D-1
D-1 MPLAB IDE Project Code Files	D-1
D-1-1 system.h	D-1
D-1-2 uart.h	D-2
D-1-3 delay.h.....	D-3
D-1-4 uart.c	D-4
D-1-5 delay.c.....	D-6
D-1-6 dataLogging.c.....	D-7

D-2	Microsoft Visual Basic Project Code Files	D-11
	D-2-1 dataLogging.vb.....	D-11
D-3	VBA in Word Project Code Files	D-33
	D-3-1 inputDataForm.frx	D-33
	D-3-2 reportGenMain.bas.....	D-42
APPENDIX E COMPANY PORTFOLIO LIST DATABASE EXCEL		E-1
APPENDIX F AUTOMATION REPORT		F-1

LIST OF FIGURES

Figure Number	Title	Page
Figure 3-1-F1	Basic Elements of Data Logging System	6
Figure 3-2-1-1-F1	SKDS40A 40 pins dsPIC Start-Up Kit	9
Figure 3-2-1-2-F1	Pin Diagram of dsPIC30F3011	10
Figure 4-3-F1	Starting of GUI interface data logging system	16
Figure 4-3-F2	No COM port is selected	17
Figure 4-3-F3	No COM port is connected	18
Figure 4-3-F4	No COM port is selected	19
Figure 4-3-F5	'Read' button clicked	20
Figure 4-3-F6	COM port is unplug when reading data	21
Figure 4-3-F7	Viewing real-time graph by click on check box	22
Figure 4-3-F8	Zoom in the real-time graph	23
Figure 4-3-F9	Right click pops out context menu for zoom selection	24
Figure 4-3-F10	'Pause' button clicked	25
Figure 4-3-F11	'Save Data' button clicked	26
Figure 4-3-F12	'Report Generation' button clicked	27
Figure 4-3-F13	'Cancel' button clicked in automation report	28
Figure 4-3-F14	User type in company name	29
Figure 4-3-F15	The list of company name will pop out	29
Figure 4-3-F16	'Exit' button is clicked	30

Figure 4-3-F17	'Ok' button is clicked when asked exiting program	31
Figure 5-1-F1	GUI Interface PC based Data Logging System	32
Figure 5-1-F2	Hide all channel's graph	33
Figure 5-1-F3	Channel 1 graph showed by tick 'Ch11' check box	34
Figure 5-1-F4	Channel 2 graph showed by tick 'Ch12' check box	35
Figure 5-1-F5	Channel 3 graph showed by tick 'Ch13' check box	36
Figure 5-1-F6	Channel 4 graph showed by tick 'Ch14' check box	37
Figure 5-1-F7	Channel 1 & 3 graph showed	38
Figure 5-1-F8	Real-time measurement based on data received	38
Figure 5-1-F9	All data received and graph are clear	39
Figure 5-2-F1	Data Logging for 1 week	40
Figure 5-2-F2	Data Received and Data Measurement for 4 channels	41
Figure 5-2-F3	Graph for Channel 1 & Channel 2	42
Figure 5-2-F4	Graph for Channel 3 & Channel 4	42
Figure 5-3-F1	Temperature is equal or greater than 90 °C	43
Figure A-1-F1	Data Acquisition System	A-1
Figure A-2-F1	Data Acquisition System with SKDS40A	A-2
Figure B-1-F1	Data Logging Software Interface	B-1
Figure B-2-F1	Automation Report Generation	B-2
Figure E-F1	Company Portfolio List Database	E-1
Figure F-F1	Automation Report Front Cover	F-1
Figure F-F2	Automation Report Data Measurement and Graph	F-2
Figure F-F3	Automation Report Graph Channel 3 and Channel 4	F-3
Figure F-F4	Automation Report Last Page	F-4

LIST OF TABLES

Table Number	Title	Page
Table 2-T1	Comparison of Specification by Different Type	5
Table 3-1-T1	Explanation Operation of Basic Elements	7
Table 3-2-1-T1	Components of Hardware Tools	8
Table 3-2-1-T2	Plug and Use Features of SKDS40A microcontroller	9
Table A-1-T1	Type of sensor on breadboard	A-1
Table A-2-T1	Type of hardware used in project	A-2
Table C-1-T1	Project File Description - MPLAB IDE	C-1
Table C-2-T1	Project File Description - Microsoft Visual Basic	C-1
Table C-3-T1	Project File Description – VBA in Word	C-2

LIST OF ABBREVIATIONS

<i>ADC</i>	Analog to Digital Converter
<i>EEPROM</i>	Electrically Erasable Programmable Read-Only Memory
<i>SRAM</i>	Static Random Access Memory
<i>PC</i>	Personal Computer
<i>UART</i>	Universal Asynchronous Receiver/Transmitter
<i>RAM</i>	Random-Access Memory
<i>USB</i>	Universal Serial Bus
<i>LAN</i>	Local Area Network
<i>TFT</i>	Thin-film-transistor
<i>LCD</i>	Liquid Crystal Display
<i>ICSP</i>	In-Circuit Serial Programming
<i>PIC</i>	Peripheral Interface Controller
<i>I/O</i>	Input/Output
<i>LED</i>	Light-Emitted Diode
<i>RISC</i>	Reduced Instruction Set Computing
<i>CPU</i>	Central Unit Processor
<i>DSP</i>	Digital Signal Processing
<i>PWM</i>	Pulse-Width Modulation
<i>VBA</i>	Visual Basic for Applications
<i>GUI</i>	Graphics User Interface
<i>CSV</i>	Comma-separated Values
<i>PNG</i>	Portable Network Graphics

CHAPTER 1 INTRODUCTION

1-1 Project Background

Data logging is a process to collect, analyze and store the data for later use. It is a process to record events during a test or measurement with the use of a system or a product. The human brain and its memory, the nature's creation, no doubt is the best data logging mechanism. Data loggers can possibly collect the information in term of accuracy is essential where there is the need to collect information faster than a human. Data logging also implies the control of how sensor collects and analyzes the data which commonly used in scientific experiments and in monitoring systems.

A data logger is an electronic instrument that can record digital or analog measurements over a period of time. It consists of an ADC, microcontroller and a data storage device. Data loggers have an on-board memory that is large enough to hold data that is recorded over a longer period of time. The memory in these data logger maybe flash memory, EEPROM or SRAM that is battery backed. Data loggers are provided with real time clocks to record the date and time of acquisition which automatically make a record of the readings of the instruments located at different location. User can determine the type of information that want recorded. The advantage of data logger is they can operate independently of a computer and available in various shapes and sizes.

Data logging and recording is a very common measurement application. In its most basic form, data logging is the measurement and recording of physical or electrical parameters over a period of time. The data can be temperature, strain, displacement, flow, pressure, voltage, current, resistance, power, or any of a wide range of other parameters.

Real-world data logging applications are typically more involved than just acquiring and recording signals, typically involving some combination of online analysis, offline analysis, display, report generation, and data sharing. Moreover, many data logging applications are beginning to require the acquisition and storage of other types of data, such as recording sound and video in conjunction with the other parameters measured during an automobile crash test.

1-2 Motivation and Problem Statement

The portable data loggers have limited memory. Attaching it to a PC will increase the size of data saved in PC. At the same time, saved data can be analyzed using more advanced software like Matlab, Octave and etc. When a data logger is attached to PC, user can also remotely monitor this measurement PC to check the most updated data reading from data logger.

1-3 Project Scope

A microcontroller will be used to construct the data acquisition system. Internal ADCs are connected to multiple sensors to collect data. The collected data will be sent to PC via UART communication. PC based software will be developed to store these data and display it on screen at the same time. Moreover, a report automation which allows user to import the data collected and graph into Microsoft Word for further inquiry. A complete test environment including the real sensors will be constructed in the end of this project.

1-4 Project Objective

Nowadays, the data loggers become an important tool for data collection and research purpose. Thus, it is essential to have proper basic and simple analysis data logging system. This project will introduce the PC based data logging system. The objective for this project:

- Constructing the data acquisition system
- Collecting data from microcontroller through various real sensors and send it to PC
- Developing PC based software to display data and plot graphs on screen
- Generating an automation report which contain the collected data and graphs through automation

CHAPTER 2 LITERATURE REVIEW

Data loggers are widely used in indoor, outdoor and underwater environments where data is needed and the convenience of battery power is preferred. It can record different variety of energy and environment measurements such as humidity, temperature, pressure, light intensity, wind speed, pulse signals and so on.

Nowadays, there are several type of data logger had been designed and produce in the market which have their own unique specification and function. Below Table 2-T1 showed the comparison of specification with three types of data logger products:

<i>Data Loggers Products</i>	<i>ZR-RX70</i>	<i>DrDAQ</i>	<i>DI-710</i>
Number of Channels	8	14	16
Maximum Sampling rate	10us per min	100kHz (USB Streaming)	4800 Hz(Standard) 8000Hz(Stand-Alone)
Memory	Internal RAM: 1, 000,000 points Internal Flash memory: 256 MB	No	SD Card: 16MB to 2GB
PC Connection	USB Drive, USB and LAN	USB 2.0	USB and Ethernet
Built-In Sensor	No	Yes	No

LCD Screen Display	Yes	No	Yes
Price	RM 1525 / \$500	RM 498.68 / \$163.50	RM 1521.59 / \$499

Table 2-T1 Comparison of Specification by Different Type of Data Logger Products

The ZR-RX70 is a portable multi-logger which has built in 5.7 inch TFT LCD to display the data in graph form. It can be used for laboratories, oscillation tests and environmental tests (ZR-RX70 2011). On the other hand, DrDAQ is a low-cost data logger that simply plugs into the USB port of any Windows PC which has built in sensors for light, temperature and sound. Besides that, it also can be acts an oscilloscope and a signal generator by using the software PicoScope to analyze the data (Educational Data Acquisition, n.d). Moreover, The DATAQ Instruments DI-710 Series is a family of instruments for general purpose and stand-alone data logger data acquisition applications. There are several options include interface type, input voltage range and PC-connected or standalone data logger operation (Data Logger System 2013).

Comparing all these three products, the ZR-RX70 are the data logger that include the LCD display for data collected which plotted in graph form which can see the result on the spot while comparing the other two products which requires connected to PC for viewing and analyze the data collected.

However, even though the all three products are fully provided by necessarily function and specification, the prices still consider expensive as high-end data logger. In this project, a low cost version of data logger with lower specs will be built.

CHAPTER 3 METHODOLOGY AND TOOLS

3-1 Methodology

Every data logging application, from manually recording weather patterns in the 15th century, to logging the experimental parameters of a fusion reactor test in the 21st century, can be broken down into a set of five common functional requirements which shown in Figure 3-1-F1 (A Review of Data 2010). The first step of data logging application is to acquire analog data which measuring the physical phenomena by hardware data logger. Secondly, the data will be processed by any method which configure by programming while data still collecting simultaneously. Thirdly, logging is the important and necessarily requirement for data logging application. Fourthly, in order to extract useful information from data acquired, the data collected will be going through offline analysis. Lastly, the analysis data collected will be going through display, sharing and reporting depends on configuration of data logging application.

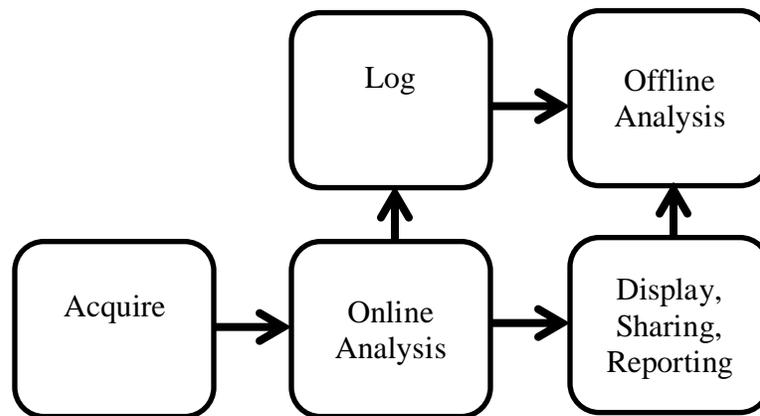


Figure 3-1-F1 Basic Elements of Data Logging System

Basic Elements	Details
Acquire	The sensors and data logger hardware are required for data collection in the process of conversion of physical phenomena into digital signals.
Online Analysis	To convert the binary value data into readable value or standard unit measurement for real sensors before the process of storing data is proceeded.
Log	The data are storing the analyzed data from the previous steps.
Offline Analysis	To analysis the data after the data is stored.
Display, Sharing, Reporting	To present and display the data, there is a creation of any reports for details display purpose. However, the data also can be presented from online analysis depends on user requirement. There are several components which are digitized hardware, data storage for long term and data logging system for data acquisition, analysis and presentation.

Table 3-1-T1 Explanation Operation of Basic Elements of Data Logging System

In this project, the hardware setup of data logger will be the first step to take on. There are various real sensors to be connected to the ADC inputs of microcontroller such as resistive humidity sensor (HR202), temperature sensor (LM35), flex sensor, light sensor and buzzer. After setup the hardware, the configuration of real sensors operation will be setup through software program and load into the microprocessor after done the coding operation of real sensors. Moreover, a design of a simple graph will be include which contains the data collection from the real sensor through the microcontroller. The details will be explained in CHAPTER 4 PROJECT SPECIFICATION [1].

3-2 Tools

3-2-1 Hardware

There is several hardware tools needed to be setup for data logger. The components of tools involved are listed at below Table 3-2-1-T1:

No	Components of Tools	Description
1	SKDS40A	40 pins dsPIC start-up kit
2	UIC00B	USB ICSP PIC Programmer V2010
3	UC00A	USB to UART converter
4	IC-PIC-30F3011	Microprocessor of dsPIC30F3011
5	Resistive Humidity (HR202)	Sensors for detect humidity in air
6	LM35	Sensors for detect temperature
7	Flex sensor	Sensors for detect the bending flex by resistive value
8	Light sensor	Sensors for detect light intensity
9	Buzzer	Sensors acts as alarm by beep sound

Table 3-2-1-T1 Components of Hardware Tools

3-2-1-1 SKDS40A Microcontroller

It is a microcontroller start-up kit designed for Microchip 40-pin dsPIC30F. It is designed to support dsPIC30F3011, dsPIC30F4011, dsPIC30F3014 and dsPIC30F4013. This board comes with basic components for user to begin project development. However, all interface and program should be developed by user. It offers plug and use features by referring to Table 3-2-1-T2 (Cytron Technologies 2011):

No	Plug and Use Features
1	Compact, powerful, flexible and robust start-up platform
2	Save development and soldering time
3	No extra components required for the dsPIC to function
4	All I/O pins are nicely labeled to avoid miss-connection by users
5	Ready connector for UIC00A/B (low cost USB ICSP PIC Programmer) which simple and fast method to load program
6	No more frustrated work plugging dsPIC out and back for re-programming
7	Program can be loaded in less than 5 seconds with UIC00A/B
8	2 x Programmable switch and 2 x LED indicator
9	Socket for external crystal oscillator and it is changeable
10	Ready with pad for parallel 2x16 LCD (optional)
11	Jumper to select UART1, UART1A (Alternative) and UART2
12	Ready with pad for UC00A (USB to UART Converter)
13	All the necessities to eliminate users difficulty in using dsPIC

Table 3-2-1-1-T1 Plug and Use Features of SKDS40A microcontroller

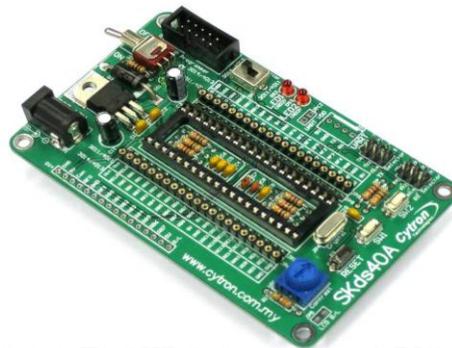


Figure 3-2-1-1-F1 SKDS40A 40 pins dsPIC Start-Up Kit

3-2-1-2 dsPIC30F3011 microprocessor

It is an enhanced flash 16 bit digital signal controller which is high performance modified RISC CPU, DSP engine, peripheral. Motor control PWM module and quadrature encoder interface module. Below is the pin diagram of dsPIC30F3011 (Microchip 2005):

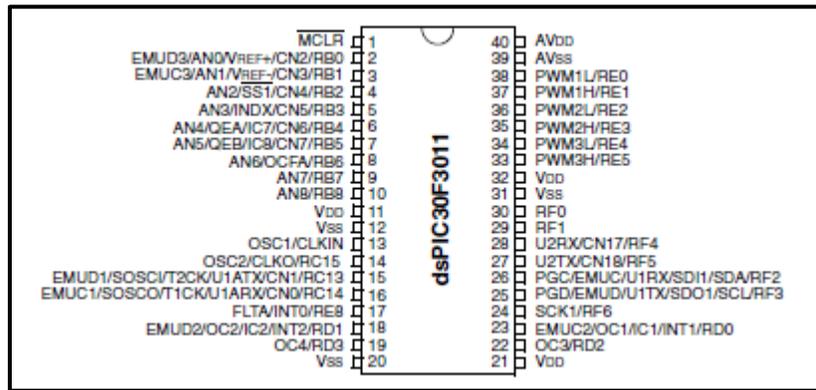


Figure 3-2-1-2-F1 Pin Diagram of dsPIC30F3011

3-2-2 Software

There is several software are needed to design data logging application software and microcontroller code. The software are listed at below:

- MPLAB Integrated Development Environment (IDE)
- PICkit 2 Programmer
- Microsoft Visual Basic 2010 in the Microsoft Visual Studio 2010
- Visual Basic for Applications (VBA) in Microsoft Word 2010

3-2-2-1 MPLAB IDE

This software is a free and integrated toolset for the development of embedded applications employing Microchip's PIC and dsPIC microcontrollers. It is called an IDE because it provides a single integrated environment to develop code for embedded microcontrollers. It runs as a 32-bit application and easy to use and includes a host of free software components for fast application development and super-charged debugging. It is also serves as a single, unified graphical user interface for additional Microchip and third party software and hardware development tools. It also contains simulator hardware interface which allow user to be easier to debug the error before load into the microprocessor (MPLAB Integrated 2012).

3-2-2-2 PICkit 2 Programmer

It is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. It mostly supports all families of 8-bit, 16-bit and 32-bit Microchip microcontrollers which are baseline (PIC10F, PIC12F5XX, and PIC16F5XX) and midrange (PIC12F6XX, PIC16F, PIC18F, PIC24, dsPIC30, dsPIC33 and PIC32). It also has functionality for in-circuit debugging on most PIC microcontrollers (PICkit2 Development 2012).

CHAPTER 4 PROJECT SPECIFICATION

4-1 Planning

In this section, all the information and requirement on designing low cost PC based data logging system have been gathered. The information for this project is available from dsPIC30F3010/3011 Data Sheet (Microchip 2005), Mastering Visual Basic 2010 eBook (Evangelo, 2012) and Getting Started with VBA in word 2010 website (Shauna, 2010). For this project, Microsoft Visual Basic was used as the major tool in designing PC based software. Each step was taken to build a low cost PC based data logging system according to following sequence:

- i. Constructing the data acquisition system
- ii. Collecting data from microcontroller through various real sensors and send it to PC
- iii. Developing PC based software to display data and plot graphs on screen
- iv. Generating an automation report which contain the collected data and graphs through automation from PC based software

In the following section will explained the process of each step had been taken to design low cost PC based data logging system until the end of project.

4-2 Design

4-2-1 Constructing data acquisition system

In this section, there are several real sensor used in this project which is resistive humidity sensor (HR202), temperature sensor (LM35), flex sensor, light sensor and buzzer. These are major real sensor to collect real world data by analog value through microprocessor except buzzer which operates as alarm when the temperature sensor detects certain level of value in degree Celsius. All these sensors are plug into the breadboard and build some simple circuit which allows normal operation for all these sensors to integrate with microcontroller. The circuit of the data acquisition system can refer to A-1 Data Acquisition System [2].

4-2-2 Collecting data from microcontroller through real sensors and send it to PC

To collect data from sensors, microcontroller is one of the important tools to interpret the analog value by ADC in microprocessor. Without conversion to analog to digital, human being is impossible to interpret the data collected from real sensors. In this section, SKDS40A and dsPIC30F3011 are used as combination of hardware tools to collect and interpret the data collected by real sensors. To program the dsPIC30F3011 microprocessor, MPLAB IDE is one of the software programming tools to configure the operation for collect data through real sensors.

In this project, C language used as MPLAB IDE programming language to write a program which allows do conversion of analog data into digital data. After done conversion, the data will be sending to PC via UART transmission by sequence. The data acquisition system with microcontroller can refer to A-2 Data Acquisition System with SKDS40A microcontroller [3].

4-2-3 Developing PC based software to display data and plot graphs

This is main section for this project because it is the main components to display the data collected send from microcontroller via UART transmission. To develop a PC based data logging software, GUI interface is mostly used in software developed application and easy to interface compares with command-line interfaces.

In this project, Microsoft Visual Basic is used to develop a PC based GUI interface data logging system to communicate with microcontroller through UART. Visual Basic as a programming language had been used to program the GUI interface. There are multiple functionality of GUI interface is designs which are:

- i. Displaying data on four separately channel according to real sensors
- ii. Displaying real-time measurement based on collected data which are average, standard deviation and peak-to-peak voltage
- iii. Real-time graph based on collected data for maximum number of four series with zoom in and zoom out function
- iv. Saving data and graph into CSV files and PNG format for further inquiry for long term period
- v. Generating an automation report through VBA Microsoft Word based on all data collected in GUI interface into a Word report to review for future purpose

The PC based GUI interface data logging system can be refer to B-1 Data Logging Software Interface [4].

4-2-4 Generating a report which contains data collected and graphs through automation from PC based software

Nowadays, report is commonly used to record the data and information which allows user to interpret. In this project, a template of report had been created which allow user to key in necessary information through form to import the data collected and graph directly into report for their reference. The VBA in Word is used as to program an automation report which using same format of report as called as template to create multiple similar report but with different name or data inserted into the report. This allow user to reuse same template report without redo the same format again for next time usage which reduce their spending time on creation of report.

In this project, a form GUI interface of automation report had been designed which can refer to B-2Automation Report Generation [5]. There are two section of this design form which are existing company and new user company. For the existing company, there are a list in the combo box list to show which connected to database of Microsoft Excel which contain the details of previously company had been typed in the data through form Word. This can be referring to APPENDIX E COMPANY PORTFOLIO LIST DATABASE EXCEL [6]. The details of company will be saved into the Microsoft Excel for future inquiry and retrieve again for next time usage.

4-3 Verification Plan

To make sure the PC based software to meet user's requirement, there are some verification testing need to proceed to prevent bug occur in certain circumstance. Below are screen shot of PC based GUI interface data logging system:

- i. The start of GUI interface of PC based GUI interface data logging system

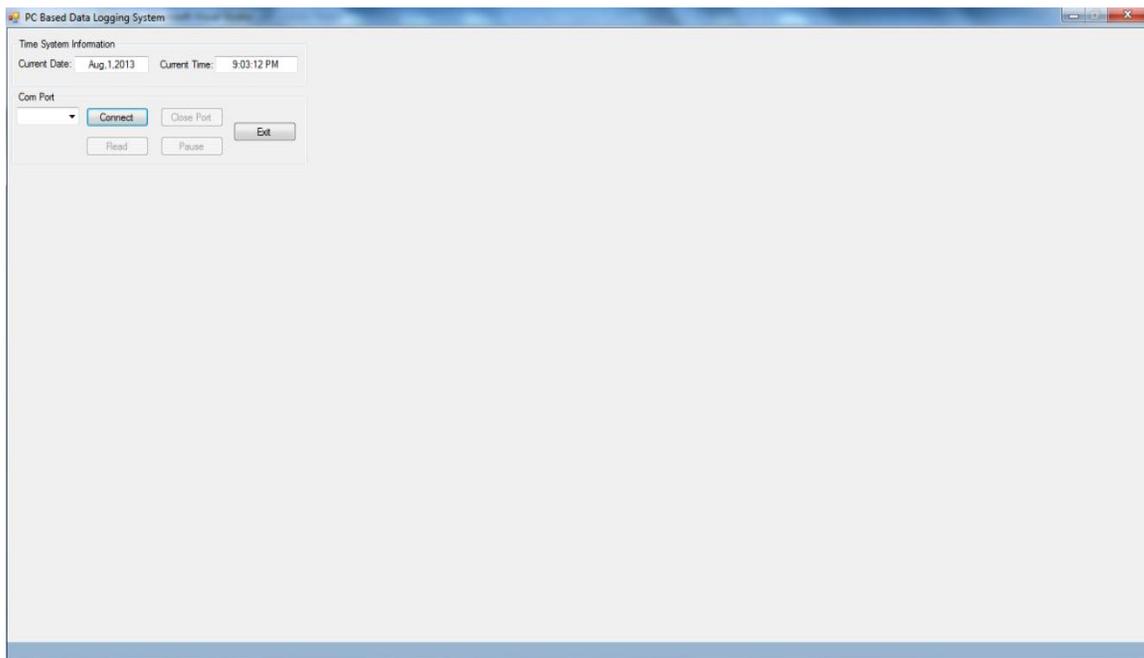


Figure 4-3-F1 Starting of GUI interface PC based data logging system

- ii. When user click 'Connect' button while no COM port is selected

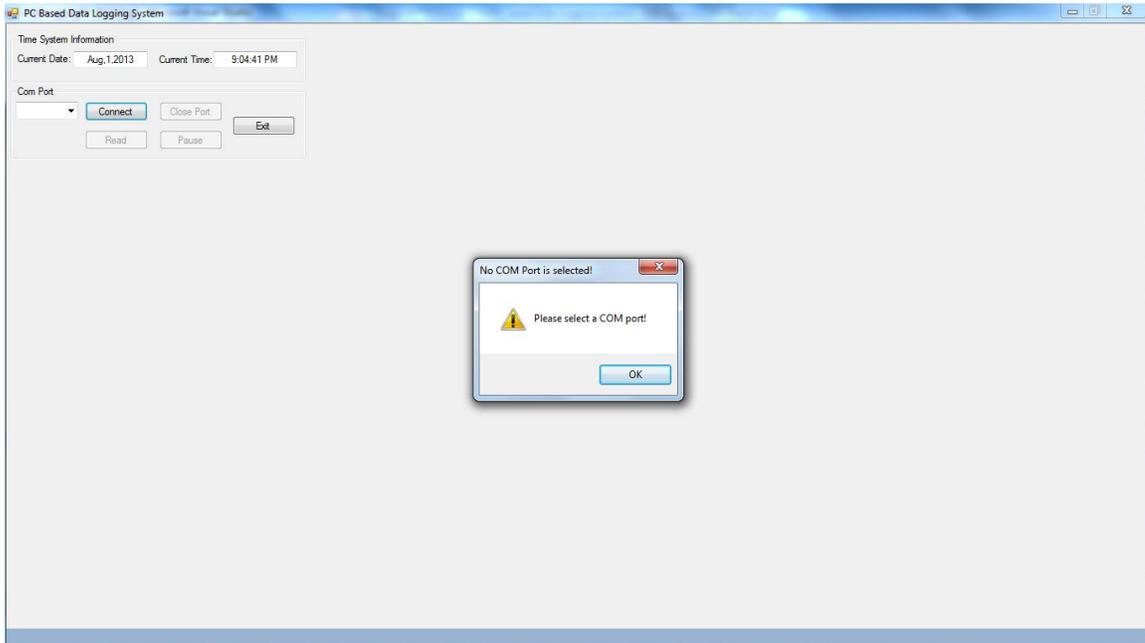


Figure 4-3-F2 No COM port is selected

- iii. When COM port name is selected but is not connected, the error message will pop out

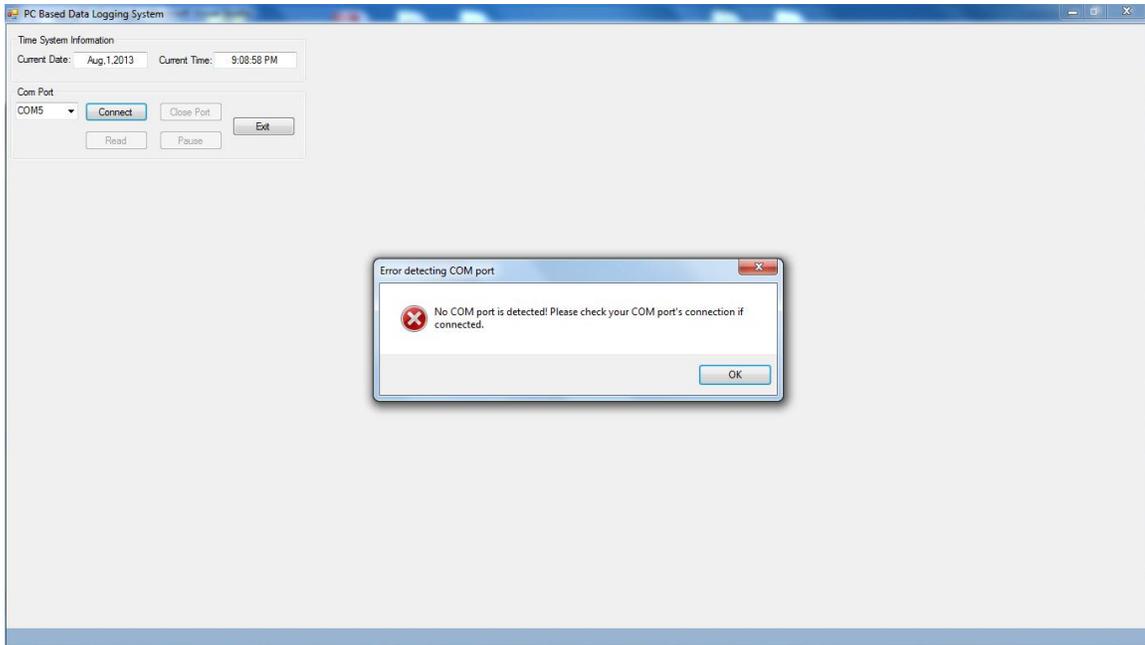


Figure 4-3-F3 No COM port is connected although COM port name is selected

- iv. When COM port is connected, user clicks on the 'Connect' Button

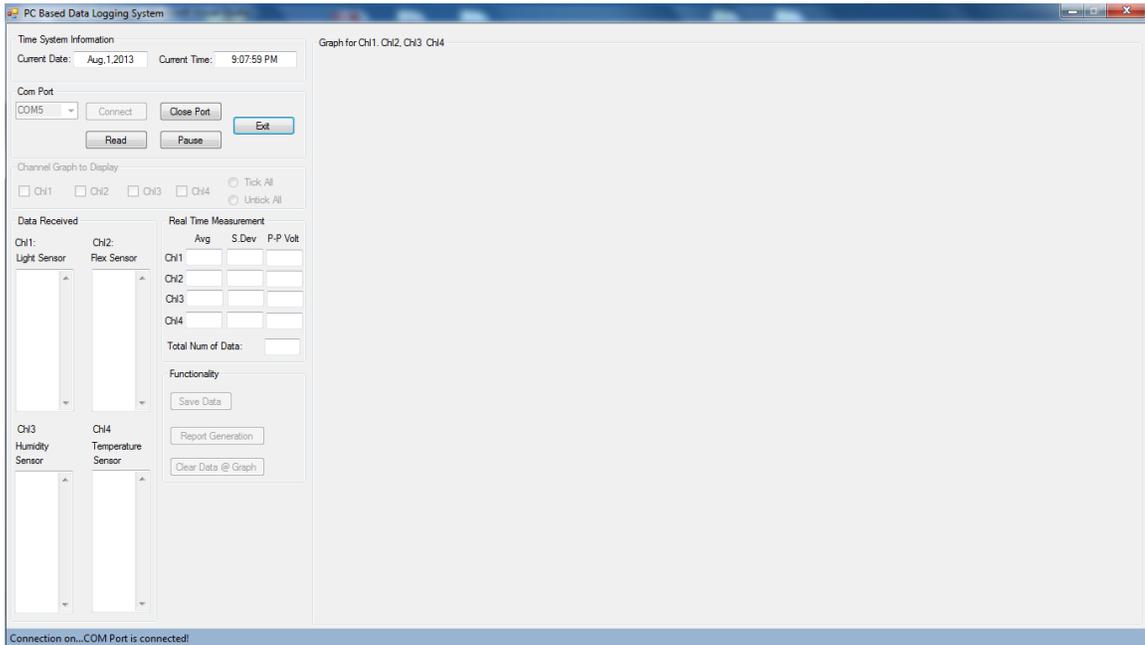


Figure 4-3-F4 No COM port is selected

- v. When user click on 'Read' button to start reading data from microcontroller through UART

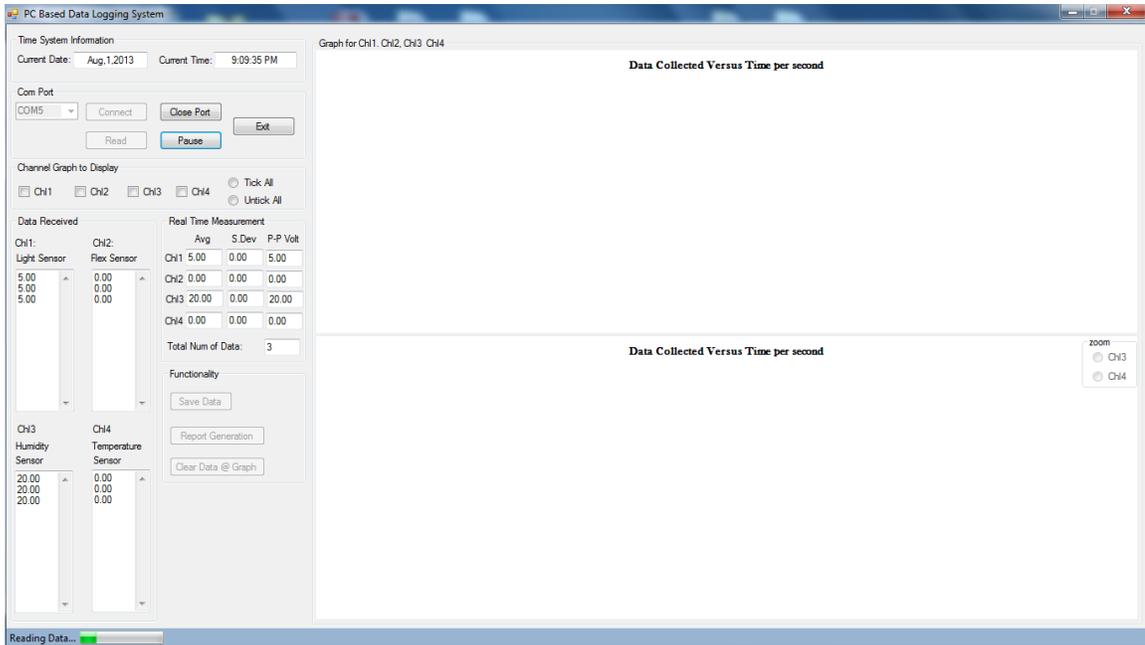


Figure 4-3-F5 'Read' button clicked

- vi. When COM port is unplug while still reading data will pop out error message

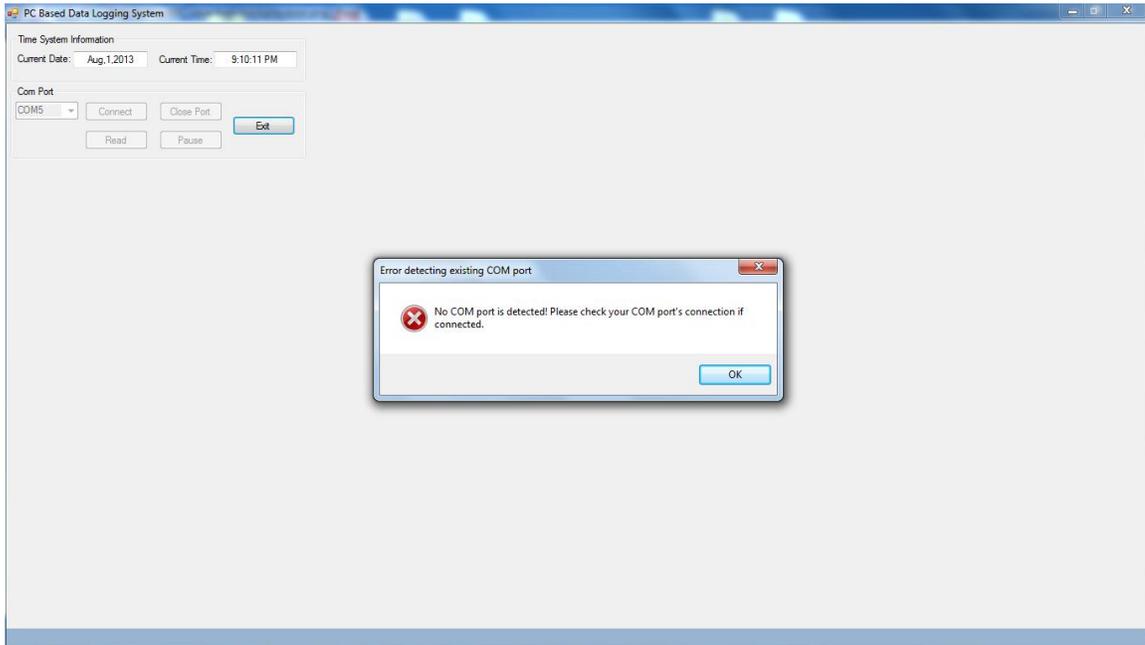


Figure 4-3-F6 COM port is unplug when reading data from microcontroller

- vii. User also can view real-time graph by click on check box in ‘Channel Graph to display’ to select or deselect the graph to be view or hide

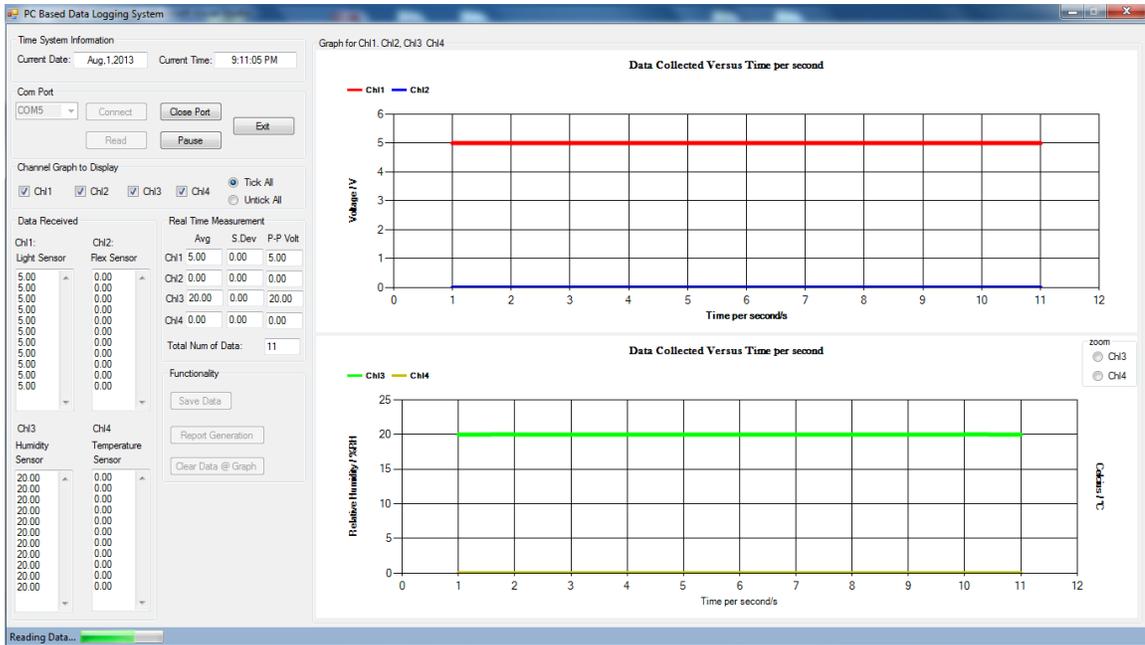


Figure 4-3-F7 Viewing real-time graph by click on check box

- viii. User also can view the graph more details by zoom in into the graph by selecting the part which to be view more detail

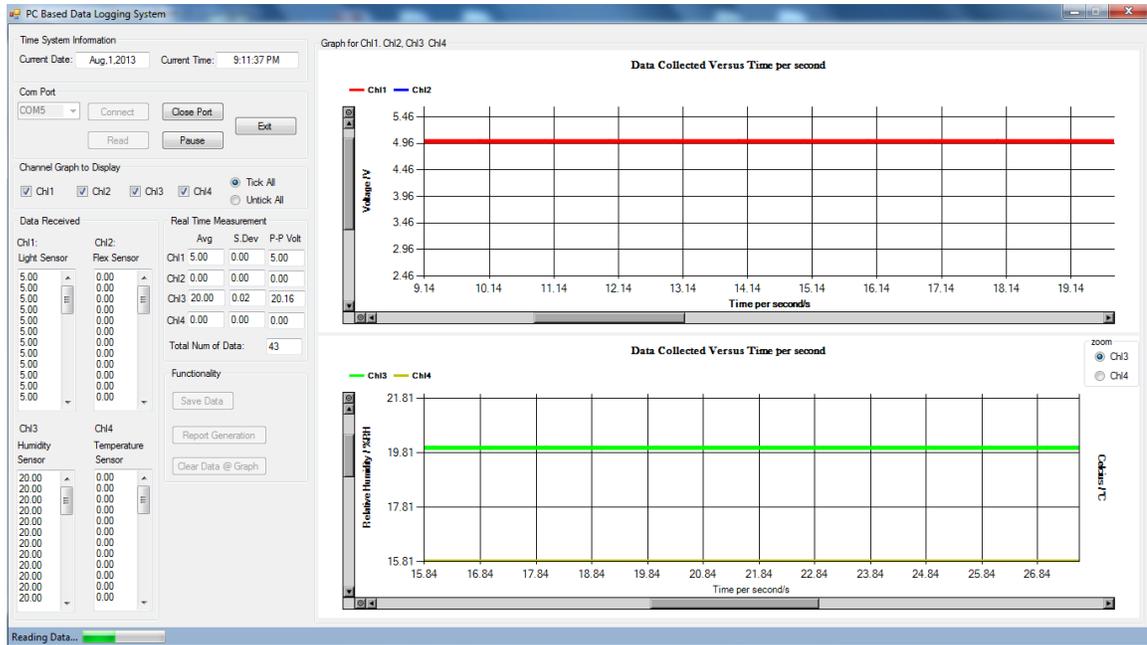


Figure 4-3-F8 Zoom in the real-time graph

CHAPTER 4 PROJECT SPECIFICATION

- ix. User also can zoom out the graph by right click to display out a small context menu to do their selection or zoom out

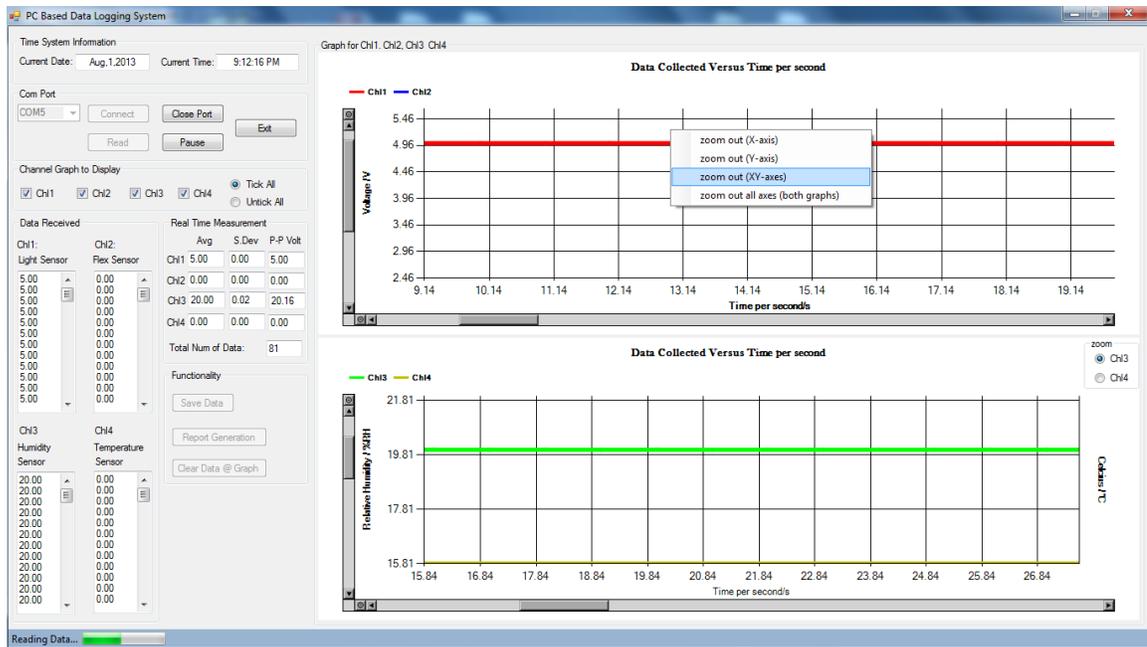


Figure 4-3-F9 Right click pop out context menu to allow zoom out selection

CHAPTER 4 PROJECT SPECIFICATION

- x. When user click on 'Pause' button, the program will stop retrieve data from microcontroller

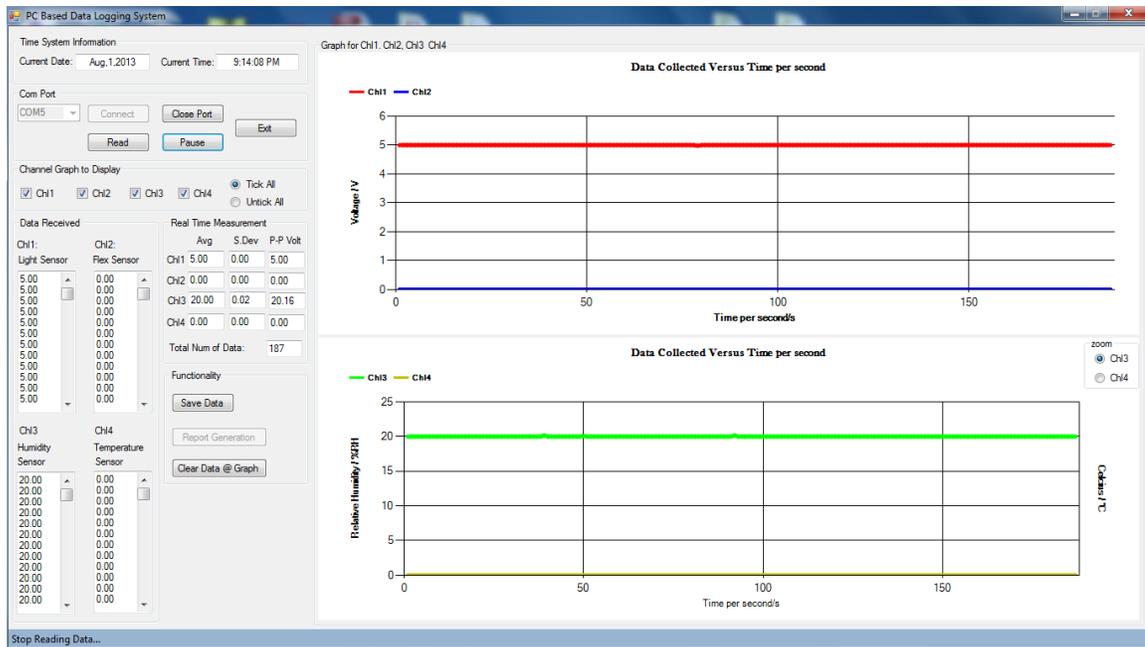


Figure 4-3-F10 'Pause' button clicked

- xi. User can save data by click on 'Save Data' button

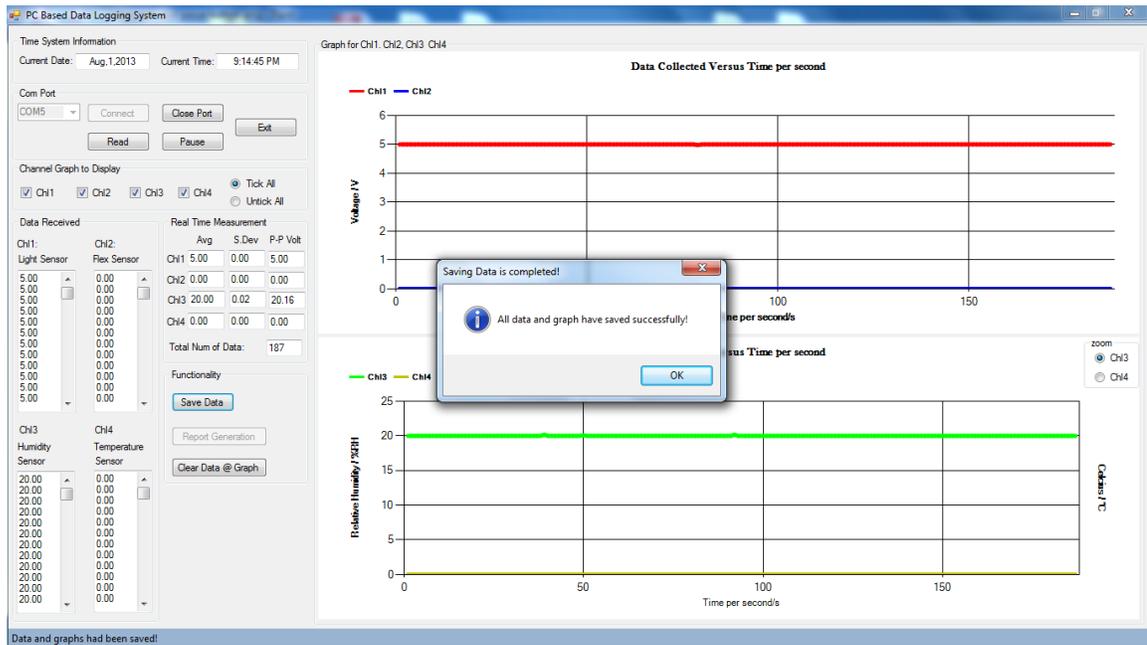


Figure 4-3-F11 'Save Data' button clicked

- xii. When user click on 'Report Generation' button, the automation report will execute

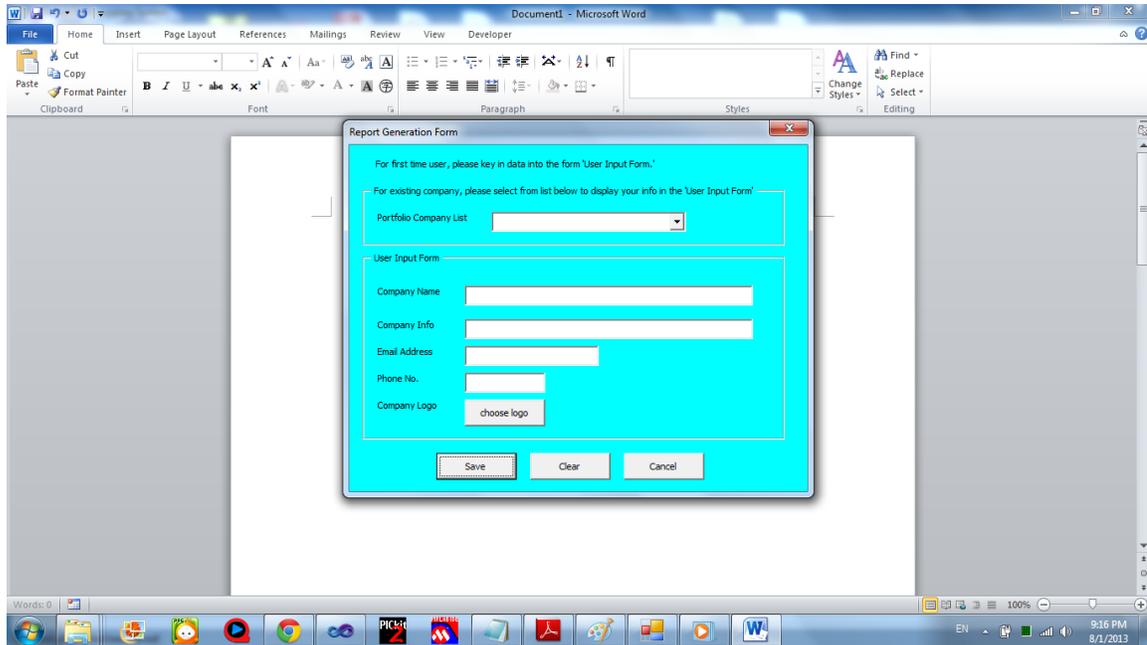


Figure 4-3-F12 'Report Generation' button clicked

xiii. When user click 'cancel' button

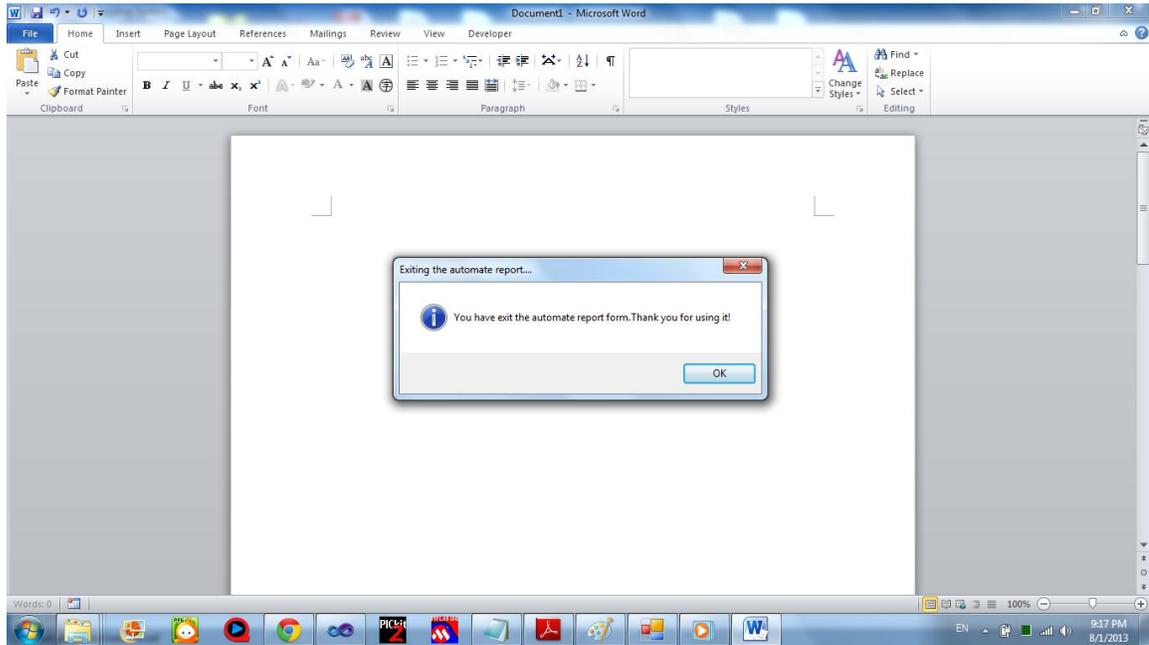


Figure 4-3-F13 'Cancel' button clicked in automation report

CHAPTER 4 PROJECT SPECIFICATION

- xiv. If user type in company had been typed before, the system will pop out message to inform user and pop out the combo list to let user choose back

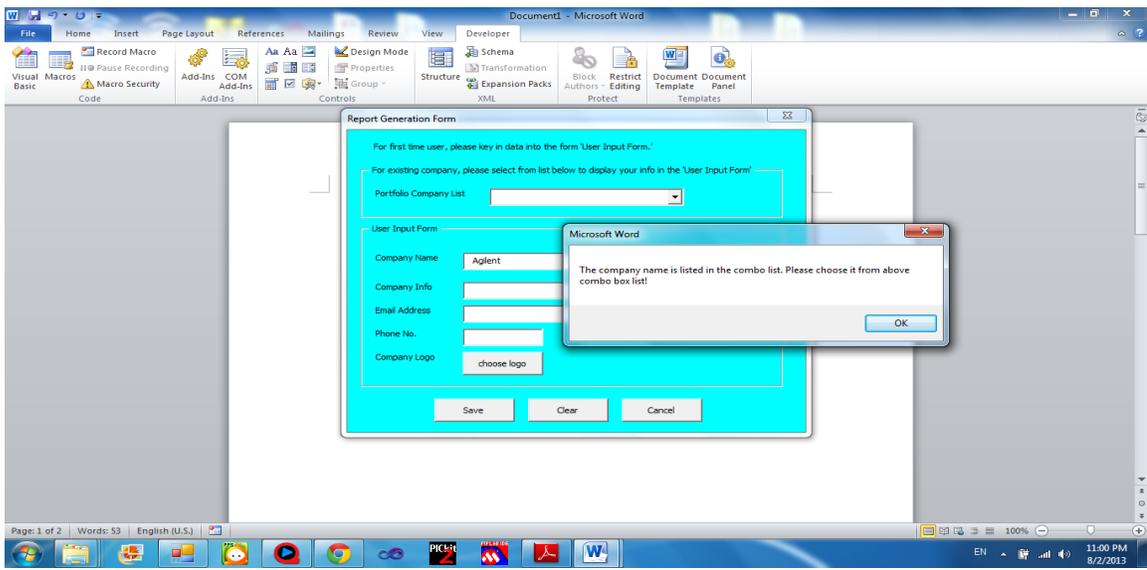


Figure 4-3-F14 User type in company which found in database Microsoft Excel

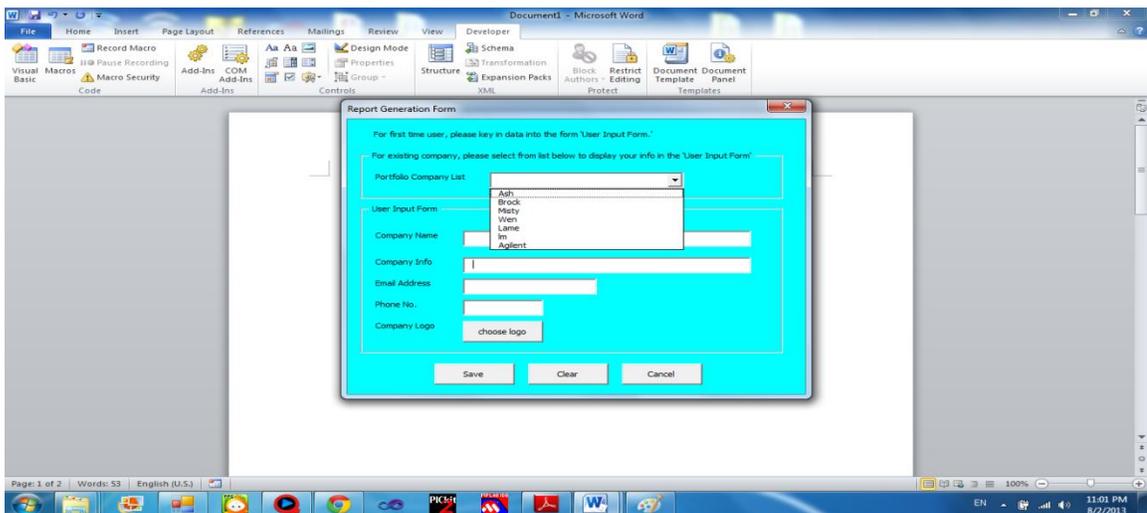


Figure 4-3-F15 The list of company name will pop out to let user choose back

xv. When user click 'Exit' button in PC based data logging system

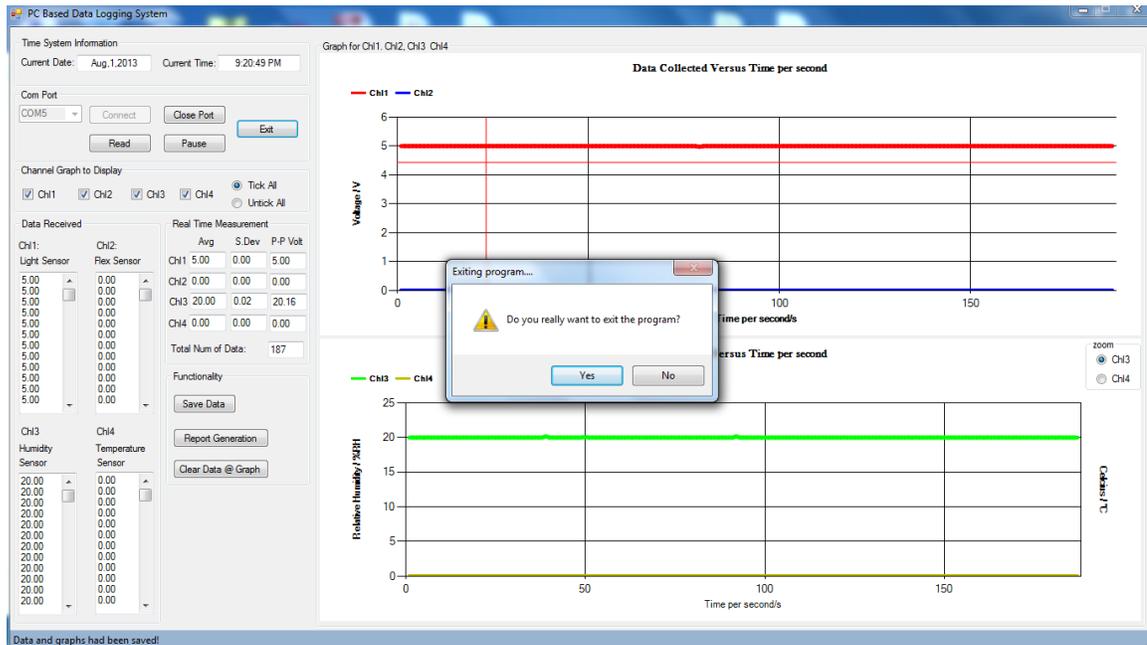


Figure 4-3-F16 'Exit' button is clicked

xvi. When use click ok for exit program

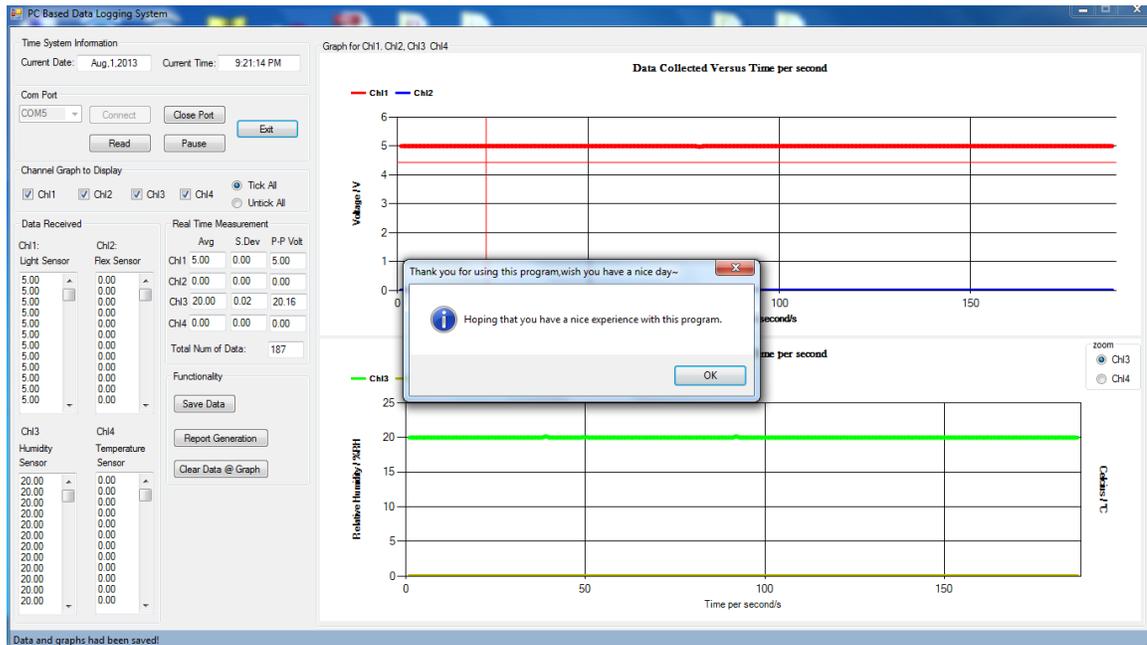


Figure 4-3-F17 'Ok' button is clicked when asked exiting program

CHAPTER 5 IMPLEMENTATION AND TESTING

5-1 Overall GUI Interface PC based Data Logging System

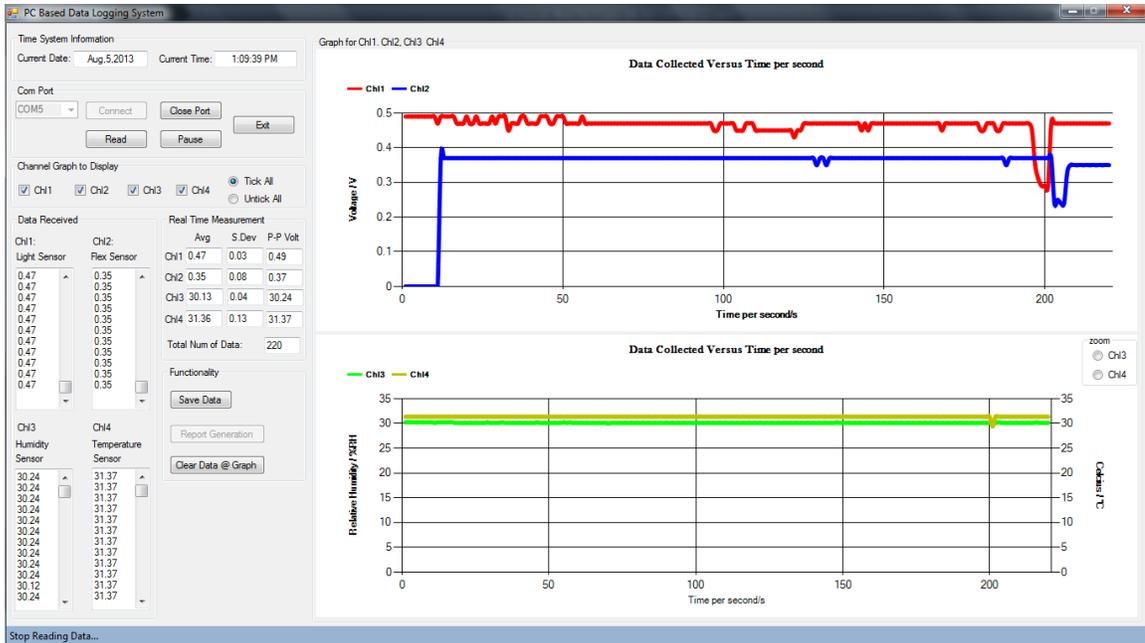


Figure 5-1-F1 GUI Interface PC based Data Logging System

CHAPTER 5 IMPLEMENTATION AND TESTING

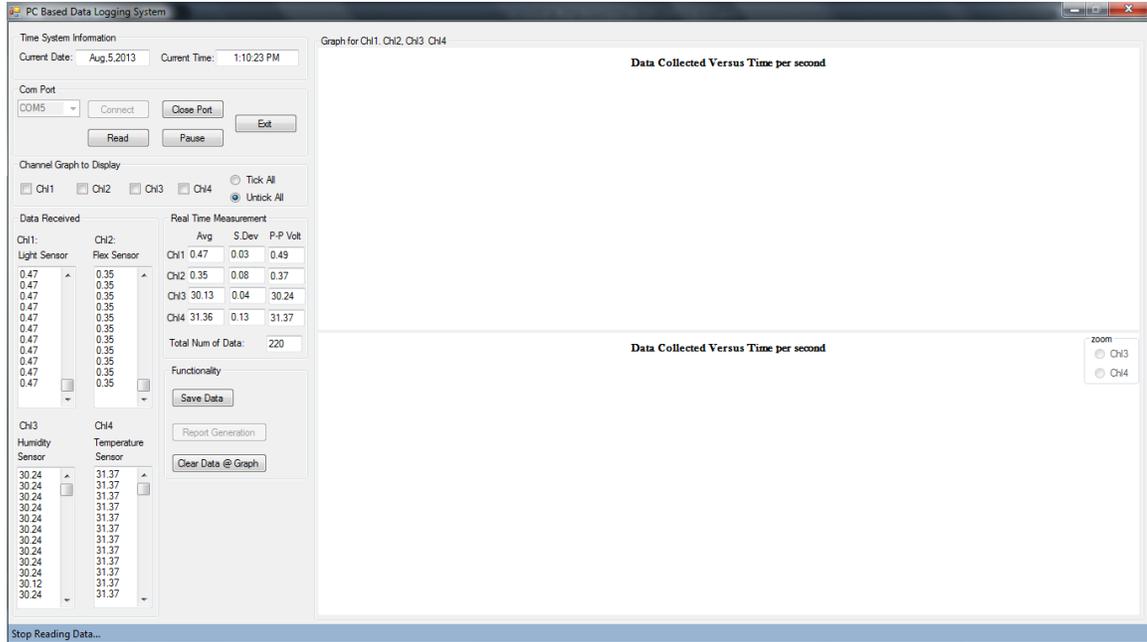


Figure 5-1-F2 Hide all channel's graph by click on 'Untick All' radio button

CHAPTER 5 IMPLEMENTATION AND TESTING

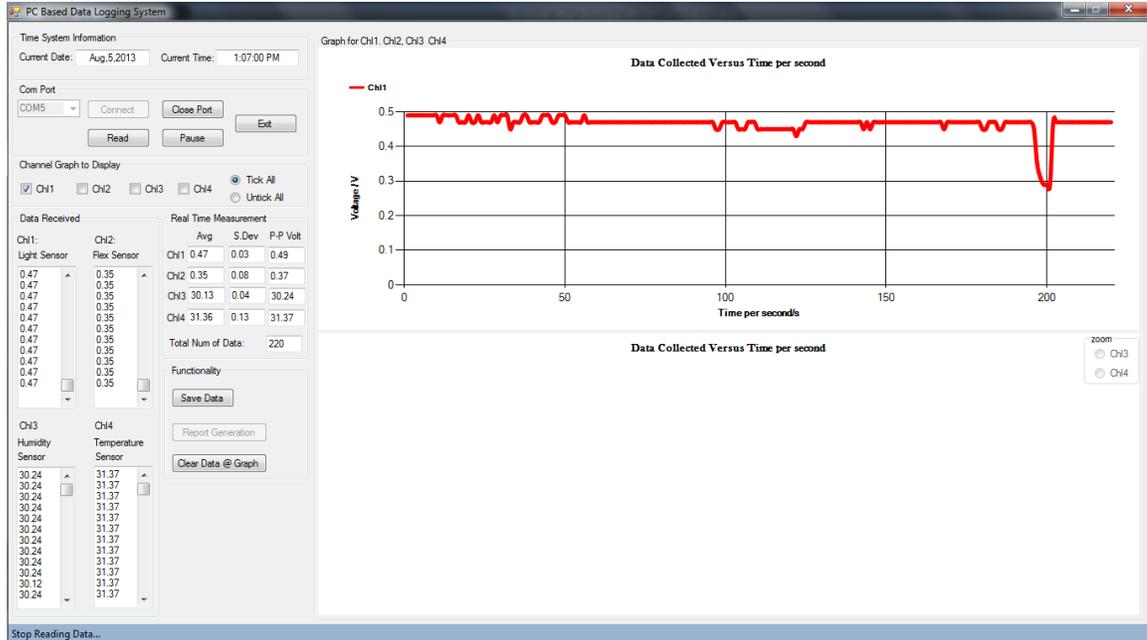


Figure 5-1-F3 Channel 1 graph showed by tick 'Ch1' check box

CHAPTER 5 IMPLEMENTATION AND TESTING

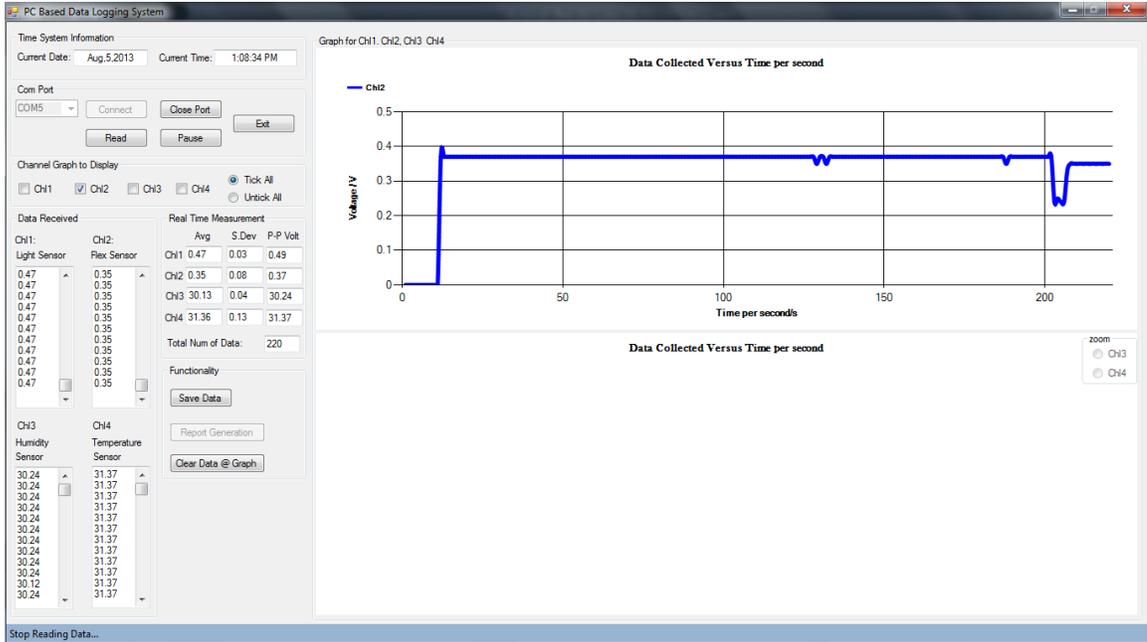


Figure 5-1-F4 Channel 2 graph showed by tick 'Ch12' check box

CHAPTER 5 IMPLEMENTATION AND TESTING

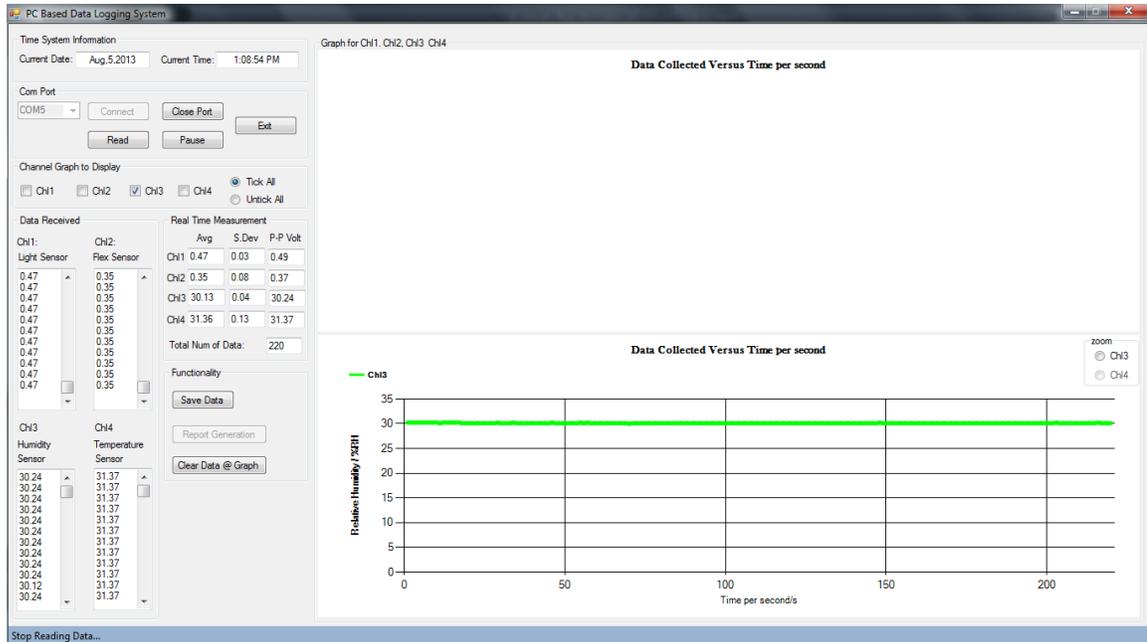


Figure 5-1-F5 Channel 3 graph showed by tick 'Ch3' check box

CHAPTER 5 IMPLEMENTATION AND TESTING

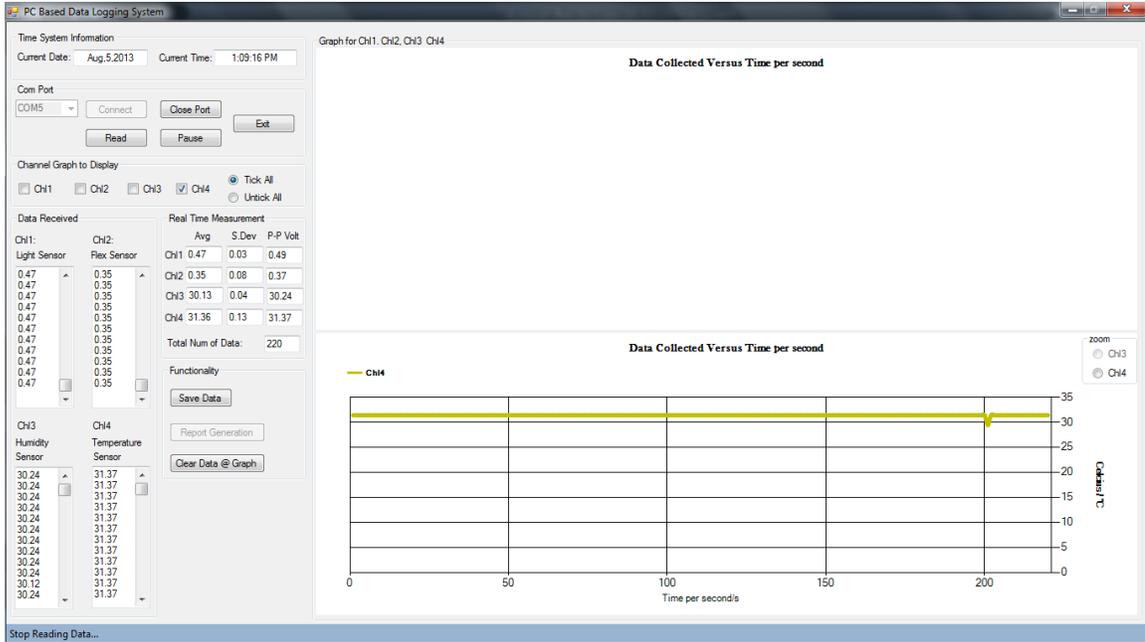


Figure 5-1-F6 Channel 4 graph showed by tick 'Ch14' check box

CHAPTER 5 IMPLEMENTATION AND TESTING

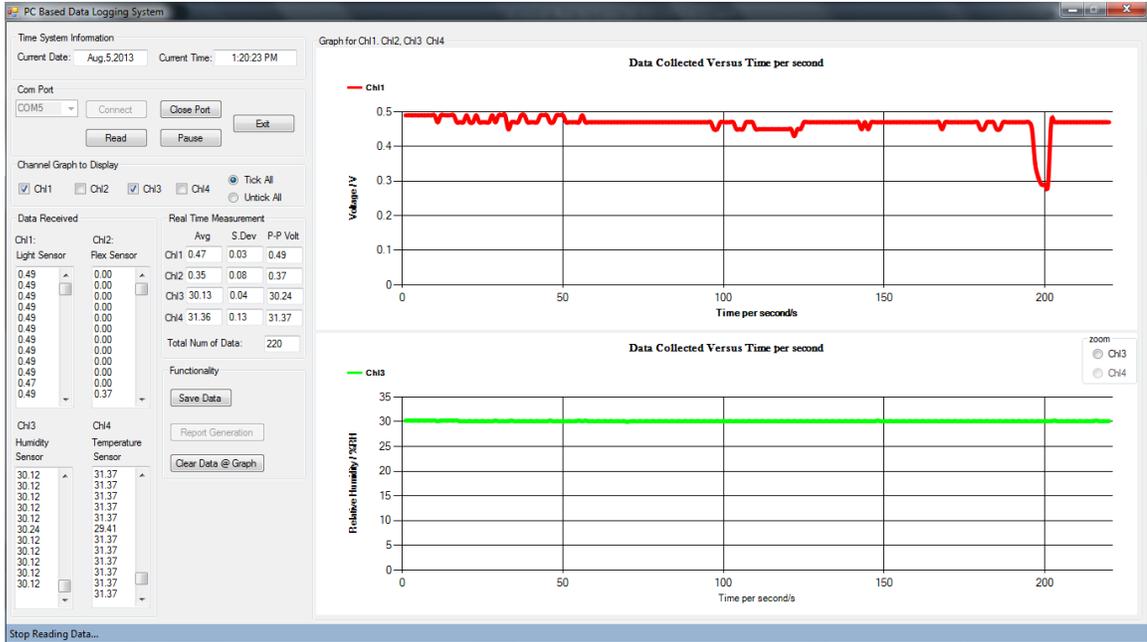


Figure 5-1-F7 Channel 1 & 3 graph showed by tick 'Ch1' and 'Ch3' check box

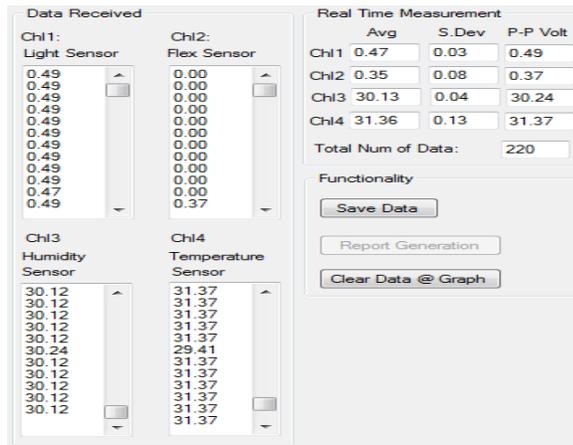


Figure 5-1-F8 Real-time measurement based on data received from microcontroller

CHAPTER 5 IMPLEMENTATION AND TESTING

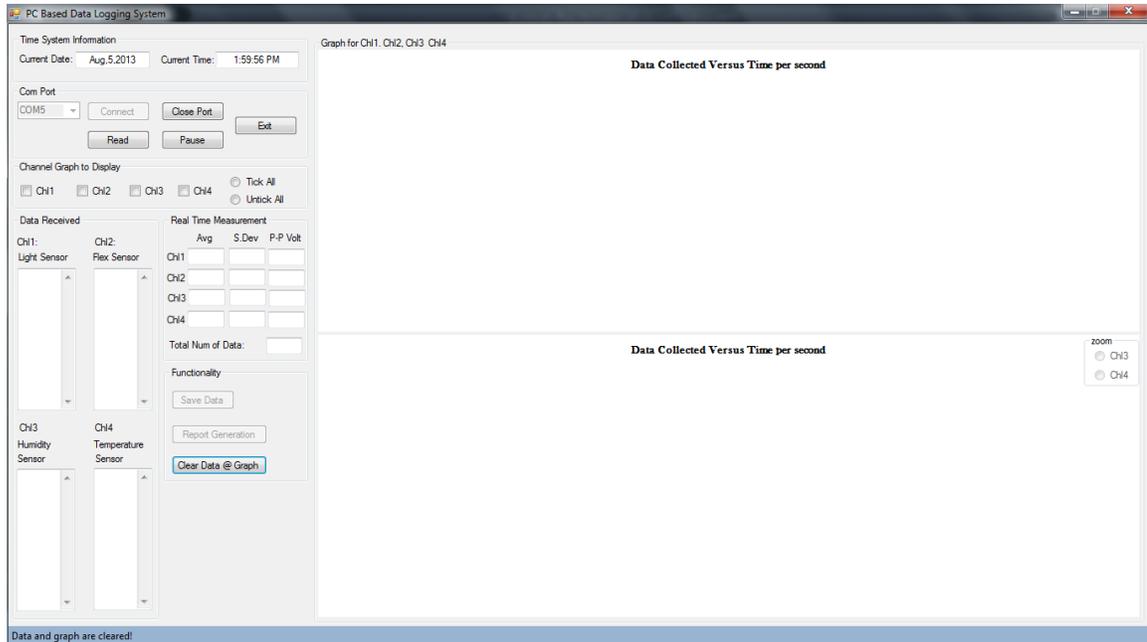


Figure 5-1-F9 All data received and graph are clear when 'Clear Data @ Graph' button is clicked

5-2 Data Logging for 1 Week

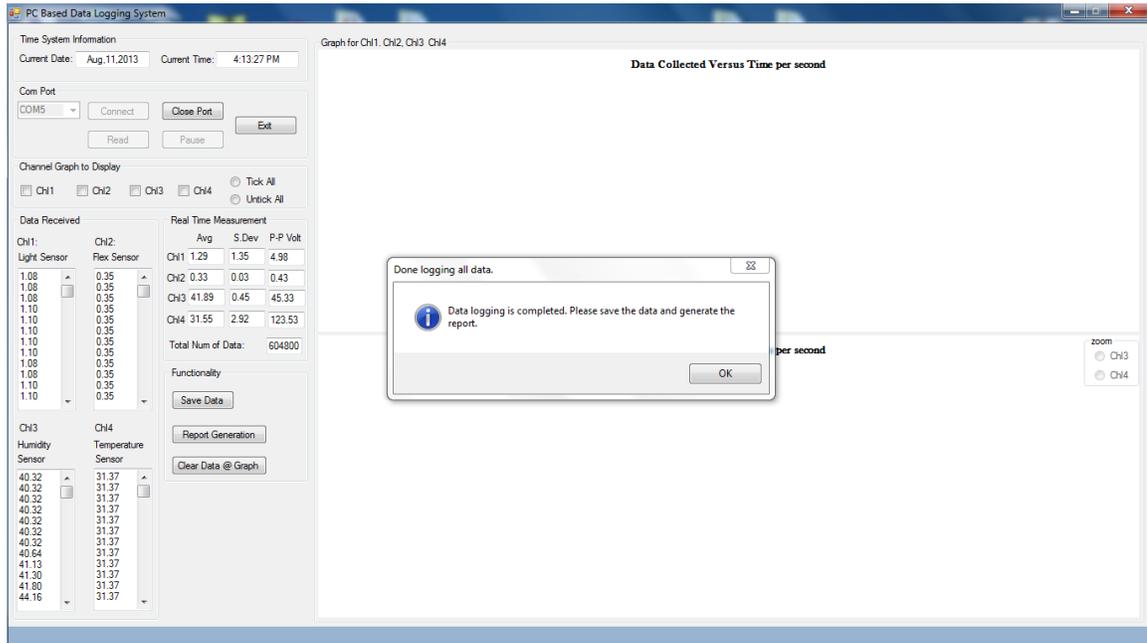


Figure 5-2-F1 Data Logging for 1 week

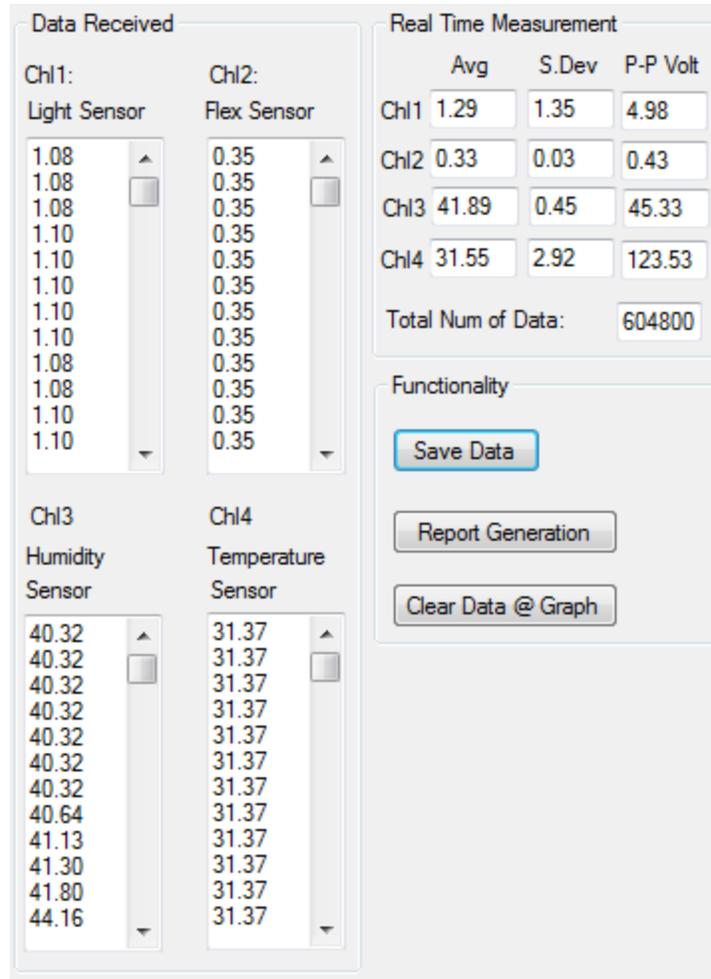


Figure 5-2-F2 Data Received and Real Time Data Measurement for 4 channel

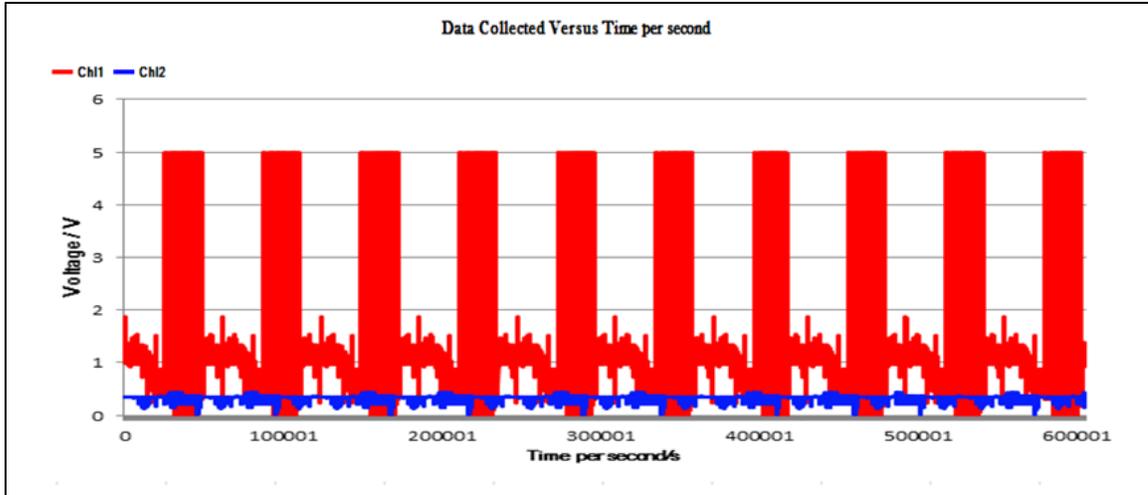


Figure 5-2-F3 Graph for Channel 1 & Channel 2

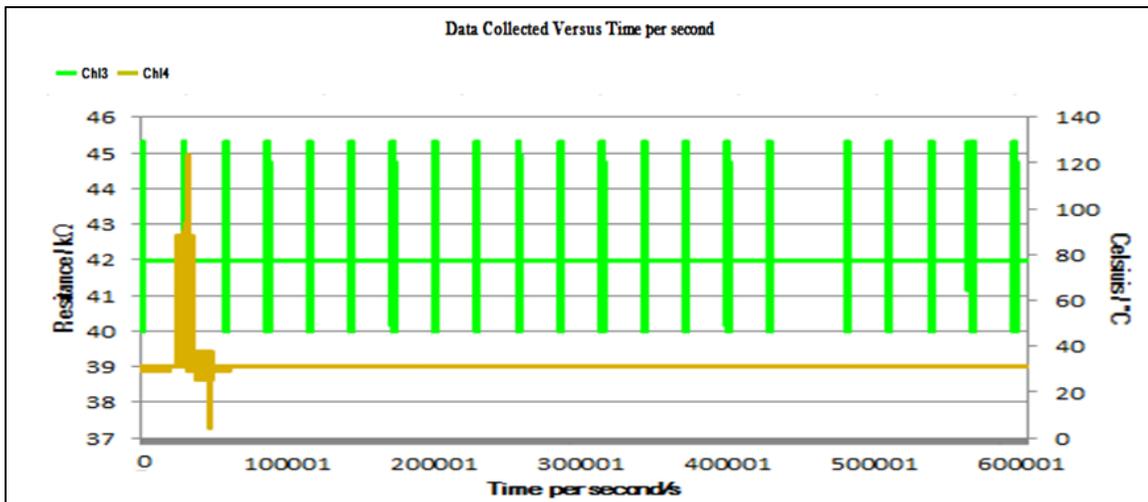


Figure 5-2-F4 Graph for Channel 3 & Channel 4

5-3 Discussion

After going through testing PC based data logging system, all four type sensor data had been collected and calculate the real-time measurement based on data collected. Besides that, the graph also plotted based on the data collected. When the temperature is equal or greater than 90 °C, the buzzer will beep 1 times for notification for high temperature had been detected by LM35. Below Figure 5-3-F2 shown the temperature is greater than 90 °C and active the buzzer to beep:

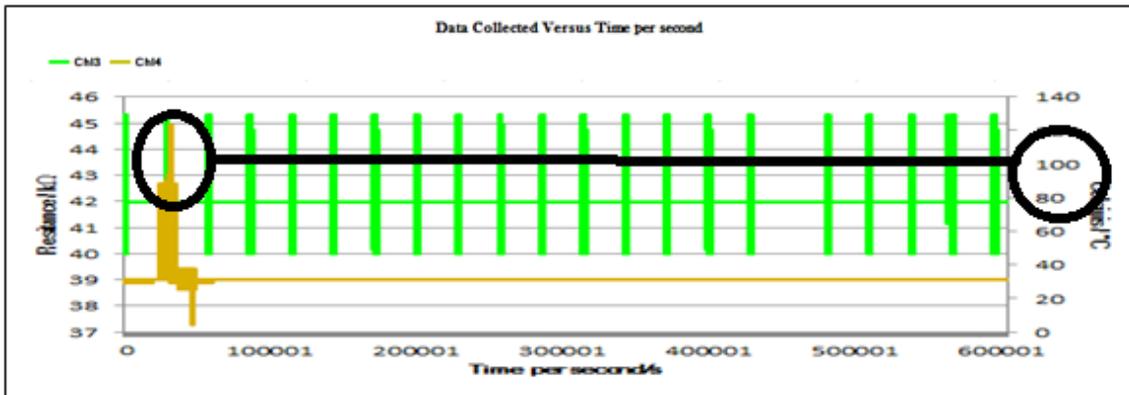


Figure 5-3-F1 Temperature is equal or greater than 90 °C

After done logging data for 1 week which set limit in the system, the program will stop collect data from microcontroller and notify user to save data and generate a report which contain data measurement and graph. The automation report can refer to APPENDIX F AUTOMATION REPORT [7].

CHAPTER 6 CONCLUSION

In this trend technology, PC based data logging system become more invaluable tool for collect and analyze differential type of data which has the real time results with sensors and it is able to respond to parameters. The limitation of microcontroller memory become the motivation of developing the PC based data logging system for more efficiency and effective way in collecting and storing differential environmental data in real world. This is the reason for developing PC based data logging system in this project for improve the data storage of microcontroller and can analyze more detail about the collected data from real sensors.

The design of low cost PC based data logging system application had been done in this project. There are several real sensor implemented in this data logging system to collect the data and display on the PC software application designed in this project with some graphical interface. The data collected and graph had been saved and can be import to Microsoft Word through automation of VBA which allows user to further usage. In overall, this project has met the project scope and its objective as stated, so it can be conclude that this project is successful.

6-1 Problem Encounter

Going through this project, there are several problem encounter and manage to solve by other solution. One of the problems is the data collected cannot display in multiple text box in the software application simultaneously due to some timer configuration. After going some several test and error, this problem had been solved by using correct timer

configuration and some configuration to retrieve data from the microprocessor through UART.

On the other hand, the graph interface only allows 2 type series data to be display in one graph which cannot add on Y-axis further more if have multiple series data need to be display. To solve this problem, the only way is to allow multiple graphs to be display in the screen application.

6-2 Future Work

Although this project's scope and objective are achieved, there are some improvements can be done in this project. First, the data collected can be saved into the EEPROM which allows the data logging system can collect data without using PC interface. After done collect the data in certain interval time, the user can retrieve the data in the EEPROM to display in software application which is more flexible in real world application.

Second, the graph interface can use other method to plot multiple series data in one graph but it will take longer time to done if lack of knowledge in this section. Besides that, it also can be improved by allow user to select the type of measurement of real sensor through the software application which is more flexible. Moreover, user can allows choosing the type of sensor to be display on screen for further configuration.

REFERENCE

REFERENCE

- A Review of Data Logging and Recording Techniques*, 2010. Available from:
<<http://www.ni.com/white-paper/2693/en/>>. [23 April 2013].
- Cytron Technologies 2011, *40 pins dsPIC Start-Up Kit SKds40A*. Available from:
<http://www.cytron.com.my/usr_attachment/SKds40A_Users_Manual.pdf>. [14 April 2013].
- Data Logger System*, 2013. Available from:
< <http://www.dataq.com/products/hardware/di710.htm>>. [23 April 2013].
- Education Data Acquisition*, n.d. Available from:
< <http://www.picotech.com/educational.html>>. [23 April 2013].
- Evangelos, P. 2012, *Mastering Microsoft Visual Basic 2010*, Indianapolis, Indiana.
Available from: SYBEX. [30 July 2013].
- Microchip 2005, *dsPIC30F4011/4012 Data Sheet*. Available from:
<<http://ww1.microchip.com/downloads/en/DeviceDoc/70141c.pdf>>. [14 April 2013].
- MPLAB Integrated Development Environment*, 2012. Available from:
<http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469>. [30 July 2013].
- PICKit 2 Development Programmer/Debugger*, 2012. Available from:
<http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023805&redirects=pickit2>. [30 July 2013].
- Shauna, K. 2010, *Getting Started with VBA in word 2010*. Available from:
<[http://msdn.microsoft.com/en-us/library/office/ff604039\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/office/ff604039(v=office.14).aspx)>. [30 July 2013].
- ZR-RX70 Portable Multi-logger*, 2012. Available from:
< http://www.omron-ap.com/product_info/ZR-RX70/index.asp>. [23 April 2013].

APPENDIX A HARDWARE SETUP DATA LOGGING SYSTEM

A-1 Data Acquisition System

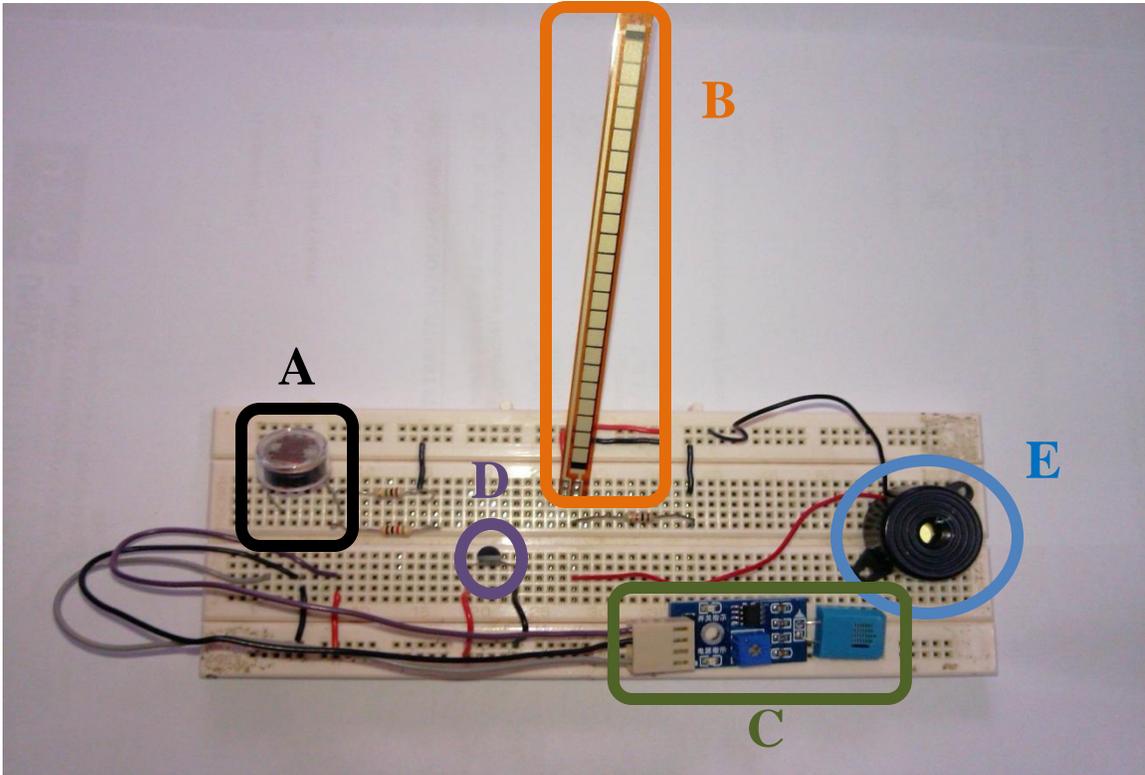


Figure A-1-F1 Data Acquisition System

Label	Type of Sensor
A	Light
B	Flex
C	Humidity (HR202)
D	Temperature (LM35)
E	Buzzer

Table A-1-T1 Type of sensor on breadboard

A-2 Data Acquisition System with SKDS40A microcontroller

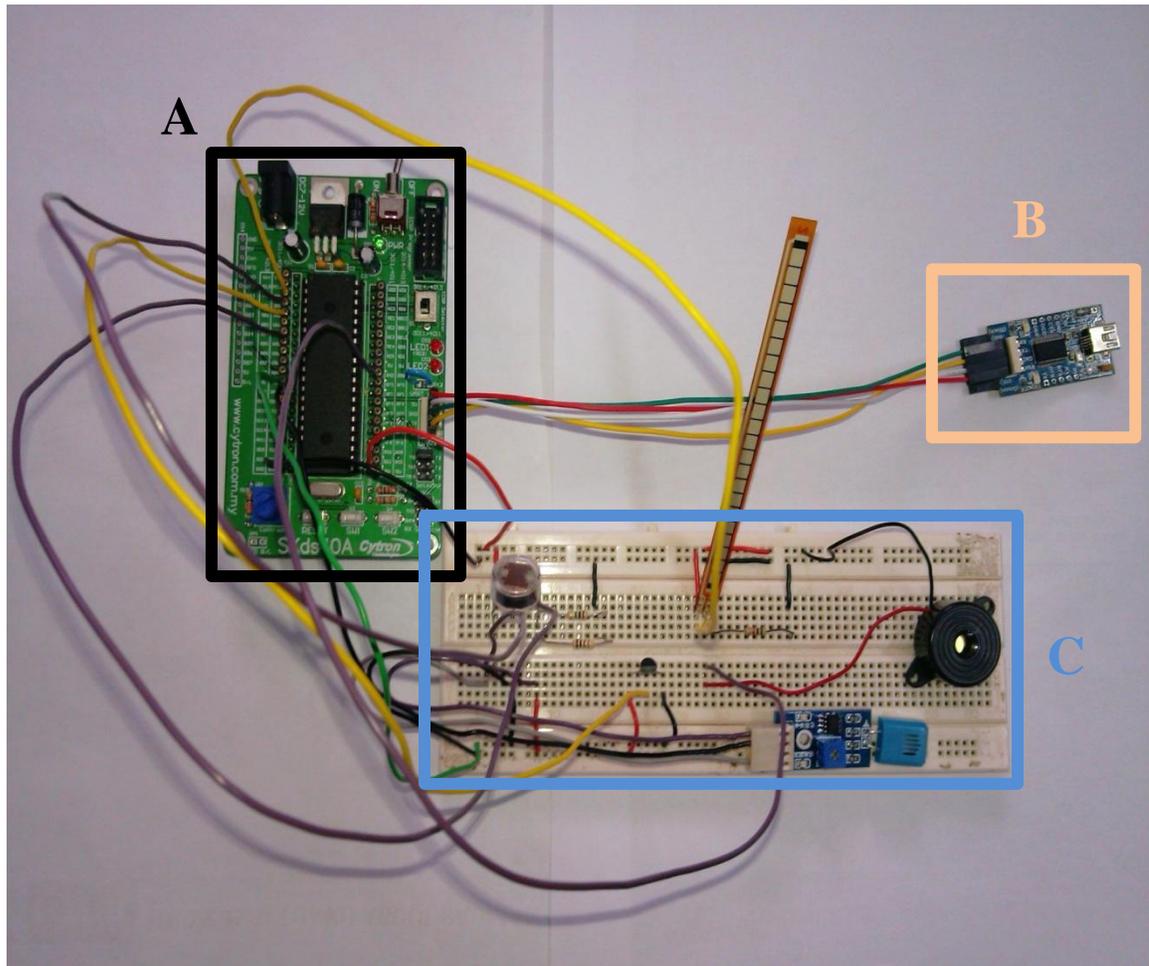


Figure A-2-F1 Data Acquisition System with SKDS40A microcontroller

Label	Type of Hardware
A	SKDS40A
B	USB to UART Converter (UC00A)
C	Data Acquisition System

Table A-2-T1 Type of hardware used in project

APPENDIX B PC BASED DATA LOGGING SOFTWARE INTERFACE

B-1 Data Logging Software Interface

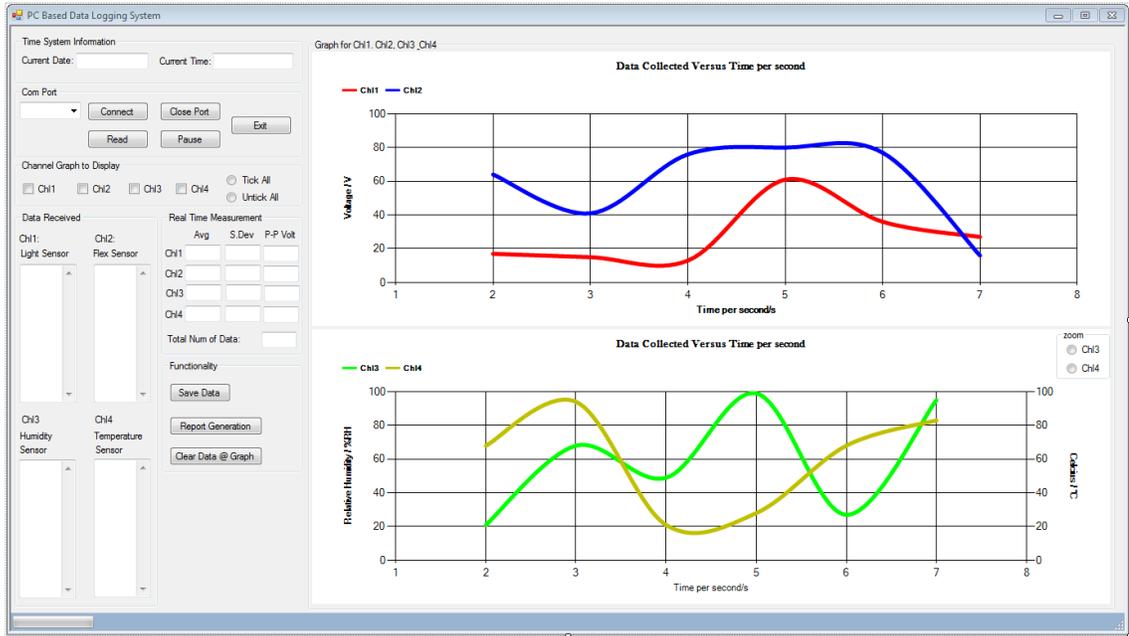


Figure B-1-F1 Data Logging Software Interface

B-2 Automation Report Generation

Report Generation Form

For first time user, please key in data into the form 'User Input Form.'

For existing company, please select from list below to display your info in the 'User Input Form'

Portfolio Company List

User Input Form

Company Name

Company Info

Email Address

Phone No.

Company Logo

Figure B-2-F1 Automation Report Generation

APPENDIX C PROJECT FILES FUNCTION

APPENDIX C PROJECT FILES FUNCTION

C-1 Project File Description - MPLAB IDE

No	File Name	Description
1	<i>system.h</i>	Initialize the oscillator frequency, UART baud rate of microprocessor (dsPIC30F3011) and I/O connections of microcontroller (SKDS40A).
2	<i>uart.h</i>	Declare function declaration for UART module.
3	<i>delay.h</i>	Declare function declaration for delay module.
4	<i>uart.c</i>	Initialize the UART module and functionality.
5	<i>delay.c</i>	Initialize the delay function.
6	<i>dataLogging.c</i>	Main program to read ADC value from real sensor through four I/O pins then send it to PC through UART

Table C-1-T1 Project File Description - MPLAB IDE

C-2 Project File Description – Microsoft Visual Basic

No	File Name	Description
1	<i>dataLogging.vb</i>	Designed GUI interface of PC based data logging application to retrieve data from microcontroller via UART and display data collected for each four channel, real-time measurement and real-time graph based on data collected.

Table C-2-T1 Project File Description - Microsoft Visual Basic

APPENDIX C PROJECT FILES FUNCTION

C-3 Project File Description – VBA in Word

No	File Name	Description
1	<i>inputDataForm.frx</i>	Designed GUI interface form for record user's company name, company's info/description, company's telephone, company's email address and company's logo.
2	<i>reportGenMain.bas</i>	Main program to do generate a report through template designed with certain format which contain all company's info and data collected through application.

Table C-3-T1 Project File Description – VBA in Word

APPENDIX D CODES FOR PROJECT FILES FUNCTION

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-1 MPLAB IDE Project Code Files

D-1-1 system.h

```
/******  
*   Final Year Project - PC Based Data Logging System           *  
*   =====                                                    *  
*   Name: Lui YongJian                                         *  
*   Course: Computer Engineering (CT)                          *  
*   Student ID: 09ACB08053                                     *  
*   Date Last Modified: 29 July 2013                           *  
*   Copyrighted (c) 2013 . All rights reserved                 *  
*   Remarks: This is the header file that describes the system *  
*   properties                                                  *  
*****/  
  
// Oscillator Frequency  
#define _XTAL_FREQ          8000000  
  
// Default UART baud rate  
#define UART_BAUD          9600  
  
// I/O Connections  
#define LED1                LATDbits.LATD2  
#define LED2                LATDbits.LATD3  
  
#define SW1                 PORTEbits.RE8  
#define SW2                 PORTDbits.RD0  
  
#define BUZZER              LATFbits.LATF0
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-1-2 uart.h

```
/******  
*   Final Year Project - PC Based Data Logging System                               *  
*   =====                                                                    *  
*   Name: Lui YongJian                                                            *  
*   Course: Computer Engineering (CT)                                           *  
*   Student ID: 09ACB08053                                                       *  
*   Date Last Modified: 29 July 2013                                           *  
*   Copyrighted (c) 2013 . All rights reserved                                 *  
*   Remarks: This file provides the functions for the UART module               *  
*****/  
  
#ifndef _UART_H  
#define _UART_H  
  
/******  
*   Function: uart1_initialize                                                    *  
*   Description: initialize the UART1 module                                     *  
*****/  
extern void uart1_initialize(void);  
  
/******  
*   Function: uc_uart1_receive                                                    *  
*   Description: receive one byte of data using UART1                           *  
*****/  
extern unsigned char uc_uart1_receive(void);  
  
/******  
*   Function: uart1_transmit                                                      *  
*   Description: transmit one byte of data using UART1                          *  
*****/  
extern void uart1_transmit(unsigned char uc_data);  
  
#endif
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-1-3 delay.h

```
/******  
*   Final Year Project - PC Based Data Logging System           *  
*   =====                                                    *  
*   Name: Lui YongJian                                         *  
*   Course: Computer Engineering (CT)                          *  
*   Student ID: 09ACB08053                                     *  
*   Date Last Modified: 29 July 2013                           *  
*   Copyrighted (c) 2013 . All rights reserved                 *  
*   Remarks: This file provides the functions for delay        *  
*****/  
  
#ifndef _DELAY_H  
#define _DELAY_H  
  
/******  
*   Function: delay_ms                                         *  
*   Description: delay in miliseconds                          *  
*****/  
extern void delay_ms(unsigned int ui_value);  
  
#endif
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-1-4 uart.c

```
/******  
*   Final Year Project - PC Based Data Logging System                               *  
*   =====                                                                    *  
*   Name: Lui YongJian                                                            *  
*   Course: Computer Engineering (CT)                                           *  
*   Student ID: 09ACB08053                                                       *  
*   Date Last Modified: 29 July 2013                                           *  
*   Copyrighted (c) 2013 . All rights reserved                                  *  
*   Remarks: This file provides the functions for the UART module               *  
*****/  
  
#include <p30F3011.h>  
#include "system.h"  
#include "uart.h"  
  
/******  
*   Function: uart1_initialize                                                    *  
*   Description: initialize the UART1 module                                     *  
*****/  
void uart1_initialize(void)  
{  
    // Set the Baud Rate.  
    U1BRG = 129;                        // ((Fosc/4)/(16 * Baud rate)) - 1  
                                        // 129 = 9600bps baudrate for 10MHz crystal with 8xPLL  
                                        // 20 = 57600bps baudrate for 10MHz crystal with 8xPLL  
                                        // 10 = 115200bps baudrate for 10MHz crystal with 8xPLL  
    U1MODEbits.ALTI0 = 0; // 1 = Communicate using U1ATX and U1ARX pins (alternative).  
                          // 0 = Communicate using U1TX and U1RX pins.  
    U1STAbits.URXISEL = 0; // Interrupt flag bit is set for every character received.  
    IPC2bits.U1RXIP  = 4; // UART1 Receive Interrupt Priority = 4.  
    U1STAbits.OERR   = 0; // Clear the Receive Overflow Flag.  
    IFS0bits.U1RXIF  = 0; // Clear the UART1 Receive Interrupt flag.  
    IEC0bits.U1RXIE  = 0; // Disable UART1 Receive Interrupt.  
    U1MODEbits.UARTEN = 1; // Enable UART1.  
    U1STAbits.UTXEN  = 1; // Enable UART1 Transmit.  
}
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```

/*****
*   Function: uc_uart1_receive                                     *
*   Description: receive one byte of data using UART1           *
*****/
unsigned char uc_uart1_receive(void)
{
    // Wait until there is data available in the receive buffer.
    while (U1STAbits.URXDA == 0);

    // Clear the overflow bit and return the received data.
    U1STAbits.OERR = 0;
    return U1RXREG;
}

/*****
*   Function: uc_uart1_receive                                     *
*   Description: transmit one byte of data using UART1         *
*****/
void uart1_transmit(unsigned char uc_data)
{
    // Wait until the transmit buffer is ready for new data.
    while (U1STAbits.UTXBF == 1);

    // Transmit the data.
    U1TXREG = uc_data;
}

```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-1-5 delay.c

```
/******  
*   Final Year Project - PC Based Data Logging System           *  
*   =====                                                    *  
*   Name: Lui YongJian                                         *  
*   Course: Computer Engineering (CT)                          *  
*   Student ID: 09ACB08053                                     *  
*   Date Last Modified: 29 July 2013                           *  
*   Copyrighted (c) 2013 . All rights reserved                 *  
*   Remarks: This file provides the functions for delay        *  
*****/  
  
#include <libpic30.h>  
#include "system.h"  
#include "delay.h"  
  
/******  
*   Function: delay_ms                                         *  
*   Description: delay in miliseconds                           *  
*****/  
void delay_ms(unsigned int ui_value)  
{  
    __delay32(_XTAL_FREQ/4000 * ui_value);  
}
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-1-6 dataLogging.c

```
/******  
*   Final Year Project - PC Based Data Logging System                               *  
*   =====                                                                    *  
*   Name: Lui YongJian                                                            *  
*   Course: Computer Engineering (CT)                                           *  
*   Student ID: 09ACB08053                                                       *  
*   Date Last Modified: 29 July 2013                                           *  
*   Copyrighted (c) 2013 . All rights reserved                                 *  
*   Remarks: This is the main program for the SKds40A + dsPIC30F3011          *  
*****/  
  
#include <stdio.h>  
#include <p30F3011.h>  
#include "uart.h"  
#include "system.h"  
#include "delay.h"  
  
/******  
*   DEVICE CONFIGURATION WORDS                                                  *  
*****/  
// Primary Oscillator Mode = XT with PLL 8x.  
// Clock Switching and Monitor = Off.  
_FOSC(CSW_FSCM_OFF & XT_PLL8);  
  
// Watchdog Timer = Off.  
_FWDT(WDT_OFF);  
  
// Brown Out Reset = Off.  
// POR Timer Value = 64ms.  
// Master Clear = Enabled.  
_FBORPOR(PBOR_OFF & PWRT_64 & MCLR_EN);  
  
// Code Protection = On.  
_FGS(CODE_PROT_ON);  
  
/******  
*   Declaring functions and variables                                          *  
*****/  
void port_initialize();
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
void configure_pins();
void buzzerBeep(unsigned char count);

unsigned int ADCdata;
unsigned char indicatorFlag = 0;
int *ADCPtr;
int count;

/*****
* MAIN FUNCTION
*****/
int main()
{
    // Configuration on ports, UART & ADC
    port_initialize();
    uart1_initialize();
    configure_pins();

    // Turn ADC on
    ADCON1bits.ADON = 1;

    // Waiting software send indicator through UART
    indicatorFlag = uc_uart1_receive();

    /* If indicatorFlag = '+' then start sampling data
       If indicatorFlag = '-' then back to main()
       If indicatorFlag = '=' then active buzzer sound */
    if (indicatorFlag == '+')
    {
        LED2 = ~LED2;
        ADCPtr = &ADCBUF0;
        IFS0bits.ADIF = 0;
        while (IFS0bits.ADIF);
        for (count = 1; count < 5; count++)
        {
            delay_ms(50);
            ADCdata = *ADCPtr++;
            ADCdata = ADCdata / 4;
            uart1_transmit(ADCdata);
            delay_ms(50);
        }
    }
}
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
    }
}
else if (indicatorFlag == '-')
{
    main();
}
else if (indicatorFlag == '=')
{
    buzzerBeep(1);
}
}
/*****
* Function: port_initialize
* Description: clear all port and initialize I/O direction
*****/
void port_initialize()
{
    LATB = 0;
    LATC = 0;
    LATD = 0;
    LATE = 0;
    LATF = 0;

    TRISB = 0;
    TRISC = 0;
    TRISD = 0;
    TRISDbits.TRISD0 = 1;
    TRISE = 0;
    TRISEbits.TRISE8 = 1;
    TRISF = 0;
}

/*****
* Function: configure_pins
* Description: configure analog inputs and ADC configuration
*****/
void configure_pins()
{
    TRISB = 0x01FF; // All inputs are port B
    ADPCFG = 0xFFD8; // ANALOG INPUT = RB0, RB1, RB2 & RB5
}
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
ADCHS = 0x0005;    // AN5 is the positive input of CH0
ADCON1 = 0x00EC;   // implement simultaneous sampling, sampling begins
                  // immediately after last conversion completes & auto
                  // convert
ADCON2 = 0x030C;   // interrupts at the completion of conversion for each
                  // 4th sample, convert the sample in CH0, CH1, CH2 & CH3
ADCON3 = 0x0309;   // Auto sampling time is 3 Tad (Tad = 2 Tcy)
}

/*****
 * Function: buzzerBeep
 * Description: active the buzzer's sound based on count
 *****/
void buzzerBeep(unsigned char count)
{
    while (count-- > 0)
    {
        BUZZER = 1;
        delay_ms(50);
        BUZZER = 0;
        delay_ms(50);
    }
}
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-2 Microsoft Visual Basic Project Code Files

D-2-1 dataLogging.vb

```
' Final Year Project - PC Based Data Logging System GUI
' =====
' Name: Lui YongJian
' Course: Computer Engineering (CT)
' Student ID: 09ACB08053
' Date Last Modified: 26 July 2013
' Copyrighted (c) 2013 . All rights reserved

Imports System
Imports System.Math
Imports System.IO.Ports
Imports System.Threading
Imports System.ComponentModel
Imports System.Windows.Forms.DataVisualization.Charting

Public Class dataLoggingGuiForm

    ' Declare and intiliaze the variable to be used in the subroutines
    Dim currDate As String = Format(Now, "MMM, d, yyyy")
    Dim currTime As String = Format(Now, "h:mm:ss tt")
    Dim comPort As Array
    Dim chl As Integer = 1
    Dim dataReceived As Queue(Of String) = New Queue(Of String) ' 1 day = 86400 data per
    seconds, 7 day = 604800

    ' =====
    ' GUI Form Load & Time System Information & Close Form Section '
    ' =====
    ' This subroutine is activated when the form is loaded ro check whether got COM port
    is connected or not
    Private Sub dataLoggingGuiForm_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'set the current date and curent time
        dateTB.Text = currDate
        timeTB.Text = currTime
        realTimeTimer.Start() ' refresh the time and date every seconds
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
comPortDetectTimer.Start()

' set the graph & progress bar & selected group box invisible on startup & set
group boxes invisible
graphVisi(False)
gpBoxVisi(False)
readDataProgreeBar.Visible = False

'set the buttons and graph display enabled or disable
readBtn.Enabled = False
pauseBtn.Enabled = False
clsBtn.Enabled = False
saveDataBtn.Enabled = False
clrDataBtn.Enabled = False
rptGenBtn.Enabled = False
chlGrpDisGp.Enabled = False
End Sub

' This function to enable or disable the visible of graph
Sub graphVisi(ByVal var)
    graph.Visible = var
    graph2.Visible = var
    zoomBox.Visible = var
End Sub

' This subroutine is to auto detect com port
Private Sub comPortDetectTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles comPortDetectTimer.Tick
    comPort = IO.Ports.SerialPort.GetPortNames()
    portCmbBox.Items.AddRange(comPort)

    If (portCmbBox.Items.Count > 0) Then
        portCmbBox.SelectedIndex = 0
        comPortDetectTimer.Stop()
    End If
End Sub

' This subroutine refresh the current time every 1 seconds
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Private Sub realTimeTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles realTimeTimer.Tick
    dateTB.Text = currDate
    currTime = Format(Now, "h:mm:ss tt")
    timeTB.Text = currTime
End Sub

' This subroutine close the connection of COM port and exit the GUI program if user
click 'Yes' by click form close button
Private Sub GUI_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
    If MessageBox.Show("Do you really want to exit the program?", "Exiting
program...", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) =
Windows.Forms.DialogResult.No Then
        e.Cancel = True
    Else
        MessageBox.Show("Hoping that you have a nice experience with this program.",
"Thank you for using this program,wish you have a nice day~", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        SerialPort1.Close()
    End If
End Sub

' Exit Button '
' This subroutine same as GUI_FormClosing function
Private Sub exitBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles exitBtn.Click
    Me.Close()
End Sub

'' ===== ''
'' Com Port Section ''
'' ===== ''

' Connect Button '
' This subroutine initliaze the connection of COM port
Private Sub connectBtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles connectBtn.Click
    If portCmbBox.SelectedIndex = -1 Then
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        MessageBox.Show("Please select a COM port!", "No COM Port is selected!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning)
    Exit Sub
Else
    With SerialPort1
        .PortName = portCmbBox.Text
        .BaudRate = 9600 'set to 9600 to synchronsn with the microcontroller
        .DataBits = 8
        .Parity = IO.Ports.Parity.None
        .StopBits = 1
        .Encoding = System.Text.ASCIIEncoding.GetEncoding(28591) 'set to
receive ASCII char from 0 - 255
        Try
            .Open()
        Catch ex As Exception
            statusBar.Text = ""
            MessageBox.Show("No COM port is detected! Please check your COM port's
connection if connected.", "Error detecting COM port", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
        Exit Sub
    End Try
End With

    statusBar.Text = "Connection on...COM Port is connected!"

' reset check box & clear text box
rstCheckBoxFunc(False)
clrTBFunc("")
clrRtTBFunc("")

' set group boxes to visible while the graph invisible on startup
gpBoxVisi(True)
graphVisi(False)
tickAllradioBtn.Checked = False
untickAllradioBtn.Checked = False

' set the buttons and graph display enabled or disable
portCmbBox.Enabled = False
connectBtn.Enabled = False
readBtn.Enabled = True
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        clsBtn.Enabled = True
        pauseBtn.Enabled = True
        saveDataBtn.Enabled = False
        clrDataBtn.Enabled = False
        chlGrpDisGp.Enabled = False
    End If
End Sub

' This function to reset the checkbox
Sub rstCheckBoxFunc(ByVal var)
    Ch11ChkBox.Checked = var
    Ch12ChkBox.Checked = var
    Ch13ChkBox.Checked = var
    Ch14ChkBox.Checked = var
End Sub

' This function to clear the real time measurement text box
Sub clrRtTBFunc(ByVal var)
    avgCh11.Text = var
    avgCh12.Text = var
    avgCh13.Text = var
    avgCh14.Text = var

    stDevCh11.Text = var
    stDevCh12.Text = var
    stDevCh13.Text = var
    stDevCh14.Text = var

    ppVCh11.Text = var
    ppVCh12.Text = var
    ppVCh13.Text = var
    ppVCh14.Text = var
End Sub

' Read Button '
' This subroutine initliaze the start time to read data from microcontroller and
display data into text boxes
Private Sub readBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles readBtn.Click
    dataRecTimer.Start()
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
statusBar.Text = "Reading Data..."
readDataProgreeBar.Visible = True
progreeBarTimer.Start()

' start to plot graph but invisible for user based on control
graphMeasTimer.Start()
If Ch11ChkBox.Checked = True Or Ch12ChkBox.Checked = True Or Ch13ChkBox.Checked =
True Or Ch14ChkBox.Checked = True Then
    hideSeriesFunc(True)
Else
    hideSeriesFunc(False)
    radioBtnCh3.Enabled = False
    radioBtnCh4.Enabled = False
End If

'set the buttons and graph display enabled or disable
graph.Visible = True
graph2.Visible = True
zoomBox.Visible = True
readBtn.Enabled = False
saveDataBtn.Enabled = False
clrDataBtn.Enabled = False
chlGrpDisGp.Enabled = True
End Sub

' This function to disable the display of series in chart
Sub hideSeriesFunc(ByVal var)
    graph.Series("Ch11").Enabled = var
    graph.Series("Ch12").Enabled = var
    graph2.Series("Ch13").Enabled = var
    graph2.Series("Ch14").Enabled = var
End Sub

' Pause Button '
' This subroutine stop receive data from microcontroller through UART
Private Sub pauseBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles pauseBtn.Click
    stopFunc()
    readDataProgreeBar.Visible = False
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
statusBar.Text = "Stop Reading Data..."

progreeBarTimer.Stop()
graphDataTimer.Stop()
realTimeMeasTimer.Stop()
graphMeasTimer.Stop()

' set the buttons enabled or disable
readBtn.Enabled = True
saveDataBtn.Enabled = True
clrDataBtn.Enabled = True
End Sub

' Close Port Button '
' This subroutine close the COM port connection
Private Sub clsBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles clsBtn.Click
stopFunc()
realTimeMeasTimer.Stop()
graphDataTimer.Stop()
graphMeasTimer.Stop()
SerialPort1.Close()
portCmbBox.Text = ""
portCmbBox.SelectedIndex = -1

' turn off the progress bar
readDataProgreeBar.Visible = False
statusBar.Text = "Connection COM Port is closed!"

' set the buttons enabled or disable
portCmbBox.Enabled = True
connectBtn.Enabled = True
readBtn.Enabled = False
pauseBtn.Enabled = False
saveDataBtn.Enabled = True
clsBtn.Enabled = False
End Sub

' This subroutine to allow the graph and real time measurement to be display in real
time
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Private Sub graphMeasTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles graphMeasTimer.Tick
    If dataReceived.Count > 0 Then
        graphDataTimer.Start()
        realTimeMeasTimer.Start()
        If ChlBox4.Text <> "" Then
            detectHighTemNmaxDataLog(ChlBox4)
        End If
    End If
End Sub

' This function to active the buzzer to sound when temperature sensor reading >=
90° ?C & detect max data logging
Sub detectHighTemNmaxDataLog(ByVal chlBox)
    Dim data() As String
    Dim dataRng() As String
    Dim dataTrim As Double
    Dim val As Double
    Dim cntData As Long

    data = Split(chlBox.Text.Trim, vbCr)

    For i = LBound(data) To UBound(data)
        dataRng = Split(data(i).Trim)
        dataTrim = CDBl(dataRng(0))
        val = dataTrim
        cntData = cntData + 1
    Next

    If val >= 90 Then
        SerialPort1.Write("=")
    End If

    dataLoggingNumMax(cntData)
End Sub

' This function is allow the maximum number of data can be logged into the system
Sub dataLoggingNumMax(ByVal varCnt)
    If varCnt = 604800 Then
        dataRecTimer.Stop()
    End If
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        realTimeMeasTimer.Stop()
        graphDataTimer.Stop()
        graphMeasTimer.Stop()
        progreeBarTimer.Stop()
        statusBar.Text = ""
        readDataProgreeBar.Visible = False
        readBtn.Enabled = False
        pauseBtn.Enabled = False
        saveDataBtn.Enabled = True
        rptGenBtn.Enabled = True
        clrDataBtn.Enabled = True
        MessageBox.Show("Data logging is completed. Please save the data and generate
the report.", "Done logging all data.", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
End Sub

' This function to stop all timers in GUI form
Sub stopFunc()
    SerialPort1.WriteLine("-") ' Send indicator to stop sampling
    dataRecTimer.Stop()
    graphDataTimer.Stop()
End Sub

' These 2 subroutine to allow user to tick all check box or untick all
Private Sub tickAllradioBtn_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tickAllradioBtn.CheckedChanged
    If tickAllradioBtn.Checked = True Then
        rstCheckBoxFunc(True)
    End If
End Sub

Private Sub untickAllradioBtn_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles untickAllradioBtn.CheckedChanged
    If untickAllradioBtn.Checked = True Then
        rstCheckBoxFunc(False)
    End If
End Sub

' This subroutine display progress bar when reading data
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Private Sub progreeBarTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles progreeBarTimer.Tick
    readDataProgreeBar.PerformStep()
    If readDataProgreeBar.Value = 170 Then
        readDataProgreeBar.Value = 0
    End If
End Sub

'' =====
'' Serial Port Data Received Section ''
'' =====
' This subroutine receives data from microcontroller through UART
Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    dataReceived.Enqueue(SerialPort1.ReadExisting())
End Sub

' This subroutine save the collected data from microcontroller through UART into array
called buffer
Private Sub dataRecTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles dataRecTimer.Tick
    Try
        SerialPort1.WriteLine("+") ' Send indicator to start sampling data
    Catch ex As Exception
        statusBar.Text = ""
        gpBoxVisi(False)
        pauseBtn.Enabled = False
        clsBtn.Enabled = False
        readDataProgreeBar.Visible = False
        dataRecTimer.Stop()
        MessageBox.Show("No COM port is detected! Please check your COM port's
connection if connected.", "Error detecting existing COM port", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End Try

    SyncLock dataReceived
        While dataReceived.Count > 0
            Try
                chkBoxSelect(ch1)
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        Catch ex As Exception
            statusBar.Text = ""
            readDataProgreeBar.Visible = False
            dataRecTimer.Stop()
            MessageBox.Show("There is some error occur in the process of
collecting data. Please check your micrococontroller!", "Error retrieve data from
microcontroller!", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End Try

        chl += 1
        If chl > 4 Then
            chl = 1
        End If
    End While
End SyncLock
End Sub

' This fuction to enable the data display into 4 channels in text box
Sub chkBoxSelect(ByVal chl)
    Select Case chl
        Case 1
            ChlBox1.Text &= (AscW(dataReceived.Dequeue()) * (5 / 255)).ToString("F2")
& Environment.NewLine
        Case 2
            ChlBox2.Text &= (AscW(dataReceived.Dequeue()) * (5 / 255)).ToString("F2")
& Environment.NewLine
        Case 3
            ChlBox3.Text &= (10 / (AscW(dataReceived.Dequeue()) * (5 / 255)) *
100).ToString("F2") & Environment.NewLine
        Case 4
            ChlBox4.Text &= (AscW(dataReceived.Dequeue()) * (5 / 255) *
100).ToString("F2") & Environment.NewLine
    End Select
End Sub

'' ===== ''
'' Real-Time Measurement ''
'' ===== ''

' This subroutine to display the measurement on real time
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Private Sub realTimeMeasurement_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles realTimeMeasTimer.Tick
    avgCal (ChlBox1, avgCh11)
    avgCal (ChlBox2, avgCh12)
    avgCal (ChlBox3, avgCh13)
    avgCal (ChlBox4, avgCh14)

    standDevCal (ChlBox1, stDevCh11)
    standDevCal (ChlBox2, stDevCh12)
    standDevCal (ChlBox3, stDevCh13)
    standDevCal (ChlBox4, stDevCh14)

    ppVoltCal (ChlBox1, ppVCh11)
    ppVoltCal (ChlBox2, ppVCh12)
    ppVoltCal (ChlBox3, ppVCh13)
    ppVoltCal (ChlBox4, ppVCh14)
End Sub

' These 3 subfunction is for measurement data which are average, standard deviation &
peak-peak voltage
Sub avgCal (ByVal chlBox, ByVal display)
    Dim data() As String
    Dim dataRng() As String
    Dim dataTrim As Double
    Dim sum As Double
    Dim avg As Double

    data = Split(chlBox.Text.Trim, vbCr)

    For i = LBound(data) To UBound(data)
        dataRng = Split(data(i).Trim)
        dataTrim = Cdbl(dataRng(0))
        sum += dataTrim
    Next

    avg = sum / (UBound(data) + 1)
    display.Text = avg.ToString("F2")
End Sub

Sub standDevCal (ByVal chlBox, ByVal display)
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Dim data() As String
Dim dataRng() As String
Dim dataTrim As Double
Dim val As Double
Dim cntData As Long
Dim sum As Double
Dim sumSqr As Double

data = Split(chlBox.Text.Trim, vbCr)

For i = LBound(data) To UBound(data)
    dataRng = Split(data(i).Trim)
    dataTrim = Cdbl(dataRng(0))
    val = dataTrim
    cntData = cntData + 1
    sum = sum + val
    sumSqr = sumSqr + val * val
Next

countDataTextBox.Text = cntData
display.Text = Sqrt((sumSqr - (sum * sum / cntData)) / (cntData -
1)).ToString("F2")
End Sub

Sub ppVoltCal(ByVal chlBox, ByVal display)
    Dim data() As String
    Dim valA() As String
    Dim maxVal As Double
    Dim dataRng() As String
    Dim dataTrim As Double
    Dim val As Double

    data = Split(chlBox.Text.Trim, vbCr)
    valA = Split(data(0).Trim)
    maxVal = Cdbl(valA(0))

    For i = LBound(data) To UBound(data)
        dataRng = Split(data(i).Trim)
        dataTrim = Cdbl(dataRng(0))
        val = dataTrim
    Next
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        maxVal = Max(val, maxVal)
    Next

    display.Text = maxVal.ToString("F2")
End Sub

'' =====''
'' Functionality Section ''
'' =====''
' Save Data Button '
' This subroutine saves the content of the Data Received text box
Private Sub saveDataBtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles saveDataBtn.Click
    statusBar.Text = "Data and graphs had been saved!"

    saveGraph(graph, "E:\data_logging\bin\graph_image\graph1.png",
ChartImageFormat.Png)
    saveGraph(graph2, "E:\data_logging\bin\graph_image\graph2.png",
ChartImageFormat.Png)

    saveDataFile("E:\data_logging\bin\data_file\data_1.csv", "Data Collected in
Channel 1", ChlBox1)
    saveDataFile("E:\data_logging\bin\data_file\data_2.csv", "Data Collected in
Channel 2", ChlBox2)
    saveDataFile("E:\data_logging\bin\data_file\data_3.csv", "Data Collected in
Channel 3", ChlBox3)
    saveDataFile("E:\data_logging\bin\data_file\data_4.csv", "Data Collected in
Channel 4", ChlBox4)

    MessageBox.Show("All data and graph have saved successfully!", "Saving Data is
completed!", MessageBoxButtons.OK, MessageBoxIcon.Information)

    'set the buttons enabled or disable
    rptGenBtn.Enabled = True
End Sub

' This functon to save the graph image in the form
Sub saveGraph(ByVal gp, ByVal location, ByVal imgType)
    gp.SaveImage(location, imgType)
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
End Sub

' This function to save the data in text box into data file
Sub saveDataFile(ByVal location, ByVal contextTitle, ByVal chlBox)
    Dim currDate As String = Format(Now, "d/MMM/yyyy")
    Dim currTime As String = Format(Now, "h:mm:ss tt")

    My.Computer.FileSystem.WriteAllText(location, contextTitle & vbCrLf & "Date: " &
currDate & vbCrLf & "Time: " & currTime & vbCrLf, False)
    My.Computer.FileSystem.WriteAllText(location, chlBox.Text, True)
End Sub

' Report Generation Button '
' This subroutine generate report about company nam, company info & so on with raw
data & measurement result
Private Sub rptGenBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rptGenBtn.Click
    'statusBar.Text = "Open report template and generating it..."
    System.Diagnostics.Process.Start("E:\data_logging\report_gen\automate_report.dotm")
End Sub

' Clear Data @ Graph Button '
' This subroutine clear the content of the Data Received text box and reset graph
Private Sub clrDataBtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles clrDataBtn.Click
    statusBar.Text = "Data and graph are cleared!"
    clearSeriesFunc()
    clrTBFunc("")
    clrRtTBFunc("")
    tickAllradioBtn.Checked = False
    untickAllradioBtn.Checked = False
    radioBtnCh3.Checked = False
    radioBtnCh4.Checked = False
    rstCheckBoxFunc(False)
    countDataTextBox.Text = ""

    'set the buttons enabled or disable
    saveDataBtn.Enabled = False
    rptGenBtn.Enabled = False
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
' This function to clear the series point in chart
Sub clearSeriesFunc ()
    graph.Series("Ch11").Points.Clear()
    graph.Series("Ch12").Points.Clear()
    graph2.Series("Ch13").Points.Clear()
    graph2.Series("Ch14").Points.Clear()
End Sub

'' =====
'' Graph Section ''
'' =====
' Plotting real-time graph
Private Sub graphDataTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles graphDataTimer.Tick
    plotGraph1 ()
    plotGraph2 ()
    plotGraph3 ()
    plotGraph4 ()
End Sub

' These 4 subroutines plot the graph based on collected data in Data Received Group
text box
Sub plotGraph1 ()
    Dim iData() As String
    Dim xValue As Integer
    Dim yValue() As String

    graph.Series("Ch11").Points.Clear()
    iData = Split(Ch1Box1.Text.Trim, vbCrLf)

    For i As Integer = 0 To UBound(iData)
        xValue += 1
        yValue = Split(iData(i).Trim)           'get the y-value for plot
data in graph
        graph.Series("Ch11").Points.AddXY(xValue, Cdbl(yValue(0))) 'plot the graph
based on collected data
    Next
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Sub plotGraph2()  
    Dim iData2() As String  
    Dim xValue2 As Integer  
    Dim yValue2() As String  
  
    graph.Series("Ch12").Points.Clear()  
    iData2 = Split(Ch1Box2.Text.Trim, vbCr)  
  
    For i As Integer = 0 To UBound(iData2)  
        xValue2 += 1  
        yValue2 = Split(iData2(i).Trim)           'get the y-value for plot  
data in graph  
        graph.Series("Ch12").Points.AddXY(xValue2, CDb1(yValue2(0))) 'plot the graph  
based on collected data  
    Next  
End Sub  
  
Sub plotGraph3()  
    Dim iData3() As String  
    Dim xValue3 As Integer  
    Dim yValue3() As String  
  
    graph2.Series("Ch13").Points.Clear()  
    iData3 = Split(Ch1Box3.Text.Trim, vbCr)  
  
    For i As Integer = 0 To UBound(iData3)  
        xValue3 += 1  
        yValue3 = Split(iData3(i).Trim, )       'get the y-value for  
plot data in graph  
        graph2.Series("Ch13").Points.AddXY(xValue3, CDb1(yValue3(0))) 'plot the  
graph based on collected data  
    Next  
End Sub  
  
Sub plotGraph4()  
    Dim iData4() As String  
    Dim xValue4 As Integer  
    Dim yValue4() As String
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
graph2.Series("Ch14").Points.Clear()
iData4 = Split(Ch1Box4.Text.Trim, vbCr)

For i As Integer = 0 To UBound(iData4)
    xValue4 += 1
    yValue4 = Split(iData4(i).Trim)           'get the y-value for plot
data in graph
    graph2.Series("Ch14").Points.AddXY(xValue4, Cdbl(yValue4(0))) 'plot the
graph based on collected data
Next
End Sub

' Channel Graph to Display based on checkbox Section
' These 4 subroutine is to allow user to choose which channel graph to display or not

' Ch11 check box '
Private Sub Ch11ChkBox_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Ch11ChkBox.CheckedChanged
    If Ch11ChkBox.Checked = True Then
        graph.Series("Ch11").Enabled = True
    Else
        graph.Series("Ch11").Enabled = False
    End If
End Sub

' Ch12 check box '
Private Sub Ch12ChkBox_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Ch12ChkBox.CheckedChanged
    If Ch12ChkBox.Checked = True Then
        graph.Series("Ch12").Enabled = True
    Else
        graph.Series("Ch12").Enabled = False
    End If
End Sub

' Ch13 check box '
Private Sub Ch13ChkBox_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Ch13ChkBox.CheckedChanged
    If Ch13ChkBox.Checked = True Then
        graph2.Series("Ch13").Enabled = True
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        radioBtnCh3.Enabled = True
    Else
        graph2.Series("Ch13").Enabled = False
        radioBtnCh3.Enabled = False
        radioBtnCh3.Checked = False
    End If
End Sub

' Ch14 check box '
Private Sub Ch14ChkBox_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Ch14ChkBox.CheckedChanged
    If Ch14ChkBox.Checked = True Then
        graph2.Series("Ch14").Enabled = True
        radioBtnCh4.Enabled = True
    Else
        graph2.Series("Ch14").Enabled = False
        radioBtnCh4.Enabled = False
        radioBtnCh4.Checked = False
    End If
End Sub

' Zoom Function
' These 2 subroutine to determine which y-axis to zoom in or zoom out for graph2 by
radio button

' zoom Ch13 Radio Button '
Private Sub radioBtnCh3_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles radioBtnCh3.CheckedChanged
    If radioBtnCh3.Checked = True Then
        graph2.ChartAreas(0).CursorY.AxisType = AxisType.Primary
        zoomXY (True)
    Else
        graph2.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
        graph2.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)
        zoomXY (False)
    End If
End Sub

' zoom Ch14 Radio Button '
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Private Sub radioBtnCh4_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles radioBtnCh4.CheckedChanged
    If radioBtnCh4.Checked = True Then
        graph2.ChartAreas(0).CursorY.AxisType = AxisType.Secondary
        zoomXY(True)
    Else
        graph2.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
        graph2.ChartAreas(0).AxisY2.ScaleView.ZoomReset(1000)
        zoomXY(False)
    End If
End Sub

' This function is to enable or disable zoom in or zoom out for XY-axis in graph &
graph2
Sub zoomXY(ByVal var)
    graph2.ChartAreas(0).CursorX.IsUserEnabled = var
    graph2.ChartAreas(0).CursorX.IsUserSelectionEnabled = var
    graph2.ChartAreas(0).CursorY.IsUserEnabled = var
    graph2.ChartAreas(0).CursorY.IsUserSelectionEnabled = var
End Sub

' These 8 subroutine to let user zoom out the graph view to original view through
right click context menu
' graph context menu
Private Sub ZoomOutXaxisToolStripMenuItem_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ZoomOutXaxisToolStripMenuItem.Click
    graph.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
End Sub

Private Sub ZoomOutYaxisToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ZoomOutYaxisToolStripMenuItem.Click
    graph.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)
End Sub

Private Sub ZoomOutXYaxesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ZoomOutXYaxesToolStripMenuItem.Click
    graph.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
    graph.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Private Sub ZoomOutAllAxesbothGraphsToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ZoomOutAllAxesbothGraphsToolStripMenuItem.Click
    ' zoom out graph
    graph.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
    graph.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)

    ' zoom out graph2
    graph2.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
    graph2.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)
    graph2.ChartAreas(0).AxisY2.ScaleView.ZoomReset(1000)
End Sub

' graph2 context menu
Private Sub ZoomOutXaxisToolStripMenuItem1_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ZoomOutXaxisToolStripMenuItem1.Click
    graph2.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
End Sub

Private Sub ZoomOutYaxisToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ZoomOutYaxisToolStripMenuItem1.Click
    graph2.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)
    graph2.ChartAreas(0).AxisY2.ScaleView.ZoomReset(1000)
End Sub

Private Sub ZoomOutXYaxesToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ZoomOutXYaxesToolStripMenuItem2.Click
    graph2.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
    graph2.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)
    graph2.ChartAreas(0).AxisY2.ScaleView.ZoomReset(1000)
End Sub

Private Sub ZoomOutAllAxesbothGraphsToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ZoomOutAllAxesbothGraphsToolStripMenuItem1.Click
    ' zoom out graph
    graph.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
    graph.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)

    ' zoom out graph2
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
graph2.ChartAreas(0).AxisX.ScaleView.ZoomReset(1000)
graph2.ChartAreas(0).AxisY.ScaleView.ZoomReset(1000)
graph2.ChartAreas(0).AxisY2.ScaleView.ZoomReset(1000)
End Sub

'' =====''
'' Sub function for shared common usage ''
'' =====''
' This function to enable or disable the visible selected group box
Sub gpBoxVisi(ByVal var)
    chlGrpDisGp.Visible = var
    dataRecGp.Visible = var
    realTimeMeasurementGp.Visible = var
    functionalityGp.Visible = var
    graphGp.Visible = var
End Sub

' This subfunction to start the graph timer
Sub startGraphTimer()
    graphDataTimer.Start()
End Sub

' This subfunction clear collected data in Data Received Box in GUI form
Sub clrTBFunc(ByVal var)
    ChlBox1.Text = var
    ChlBox2.Text = var
    ChlBox3.Text = var
    ChlBox4.Text = var
End Sub
End Class
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-3 VBA in Word Project Code Files

D-3-1 inputDataForm.frx

```
' Final Year Project - PC Based Data Logging System
' =====
' Name: Lui YongJian
' Course: Computer Engineering (CT)
' Student ID: 09ACB08053
' Date Last Modified: 29 July 2013
' Copyrighted (c) 2013 . All rights reserved
```

Option Explicit

```
Public dataFilled As Boolean
Dim locTable As Word.Range
Dim tmpNo As String
Dim nameDetected As Boolean, saveLogo As Boolean

''''''''''variable for routine comList_Change, UserForm_Activate, getDataBtn_Click()&
calData()''''''''''
Dim dataRng As Excel.Range, locRng As Excel.Range

' This subroutine clear data in the userform and enabled user key in new data
Private Sub clearDataBtn_Click()
    Me.comName.Value = ""
    Me.comInfo.Value = ""
    Me.phone.Value = ""
    Me.emailAdd.Value = ""
    Me.comList.Value = ""

    Me.comName.Enabled = True
    Me.comInfo.Enabled = True
    Me.phone.Enabled = True
    Me.emailAdd.Enabled = True
End Sub

' This function to check the typed company name is existing in the combo list or not
Sub chkName()
    Set excelApp = CreateObject("Excel.Application")
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Set excelBk =  
excelApp.Workbooks.Open("E:\data_logging\report_gen\database_portfolioCompanyList.xlsx")  
Set excelSht = excelBk.Sheets(1)  
  
With Me.comName  
For Each dataRng In excelSht.Range("comNameList")  
If dataRng <> "" Then  
If dataRng = .Value And Me.comName.Enabled = True Then  
MsgBox "The company name is listed in the combo list. Please choose it  
from above combo box list!"  
.Value = ""  
Me.comList.DropDown  
nameDetected = True  
Exit For  
End If  
nameDetected = False  
End If  
Next dataRng  
End With  
  
exlCls  
End Sub  
  
' This subroutine to check the company is existing or not in database excel file  
Private Sub comName_Exit(ByVal Cancel As MSForms.ReturnBoolean)  
If Me.comName.Value <> "" Then  
chkName  
Else  
Exit Sub  
End If  
End Sub  
  
' This function is to get the graph image  
Sub getGraphImg()  
Dim graphBox As InlineShape  
Dim graphRng As Range  
Dim graphBox2 As InlineShape  
Dim graphRng2 As Range  
  
ActiveDocument.Undo
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Set graphRng = ActiveDocument.Bookmarks("graph").Range
Set graphRng2 = ActiveDocument.Bookmarks("graph2").Range
graphRng.Text = vbNullString
graphRng2.Text = vbNullString
Set graphBox =
ActiveDocument.InlineShapes.AddPicture(fileName:="E:\data_logging\bin\graph_image\graph1.p
ng", LinkToFile:=False, Range:=graphRng)
Set graphBox2 =
ActiveDocument.InlineShapes.AddPicture(fileName:="E:\data_logging\bin\graph_image\graph2.p
ng", LinkToFile:=False, Range:=graphRng2)
graphBox.LockAspectRatio = False
graphBox2.LockAspectRatio = False
graphBox.Height = InchesToPoints(4)
graphBox2.Height = InchesToPoints(4)
ActiveDocument.Bookmarks.Add "graph", graphRng
ActiveDocument.Bookmarks.Add "graph2", graphRng2
End Sub

' This subroutine is to list down the company name list in the combo box of InputDataForm
which available in excel file
Private Sub UserForm_Activate()
Set excelApp = CreateObject("Excel.Application")
Set excelBk =
excelApp.Workbooks.Open("E:\data_logging\report_gen\database_portfolioCompanyList.xlsx")
Set excelSht = excelBk.Sheets(1)

For Each dataRng In excelSht.Range("comNameList")
If dataRng <> "" Then
Me.comList.AddItem dataRng.Text
Else
Exit For
End If
Next dataRng

exlCls
End Sub

' This subroutine is auto insert existing user data into userform by let user select their
company name in combo list
Private Sub comList_Change()
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Set excelApp = CreateObject("Excel.Application")
Set excelBk =
excelApp.Workbooks.Open("E:\data_logging\report_gen\database_portfolioCompanyList.xlsx")
Set excelSht = excelBk.Sheets(1)
Set dataRng = excelSht.Range("comNameList", excelSht.Cells(excelSht.Rows.count,
1).End(xlUp))

Set locRng = dataRng.Find(What:=Me.comList.Value, LookIn:=xlValues)

'retrieve user data from excel file
Me.comName.Value = locRng.Offset(0, 0).Value
Me.comInfo.Value = locRng.Offset(0, 1).Value
Me.phone.Value = locRng.Offset(0, 2).Value
Me.emailAdd.Value = locRng.Offset(0, 3).Value

'Not allow user to edit the data filled in the userform
Me.comName.Enabled = False
Me.comInfo.Enabled = False
Me.phone.Enabled = False
Me.emailAdd.Enabled = False

exlCls
End Sub

Private Sub cnlBtn_Click()
Me.Hide
End Sub

' This subroutine is to declare data file
Sub dataDeclaration()
Dim csvFileName1 As String
Dim csvFileName2 As String
Dim csvFileName3 As String
Dim csvFileName4 As String
Dim locTable As Word.Range

Set dataVars = ActiveDocument.Variables
Set excelApp = CreateObject("Excel.Application")

csvFileName1 = "E:\data_logging\bin\data_file\data_1.csv"
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
csvFileName2 = "E:\data_logging\bin\data_file\data_2.csv"  
csvFileName3 = "E:\data_logging\bin\data_file\data_3.csv"  
csvFileName4 = "E:\data_logging\bin\data_file\data_4.csv"  
Set excelBk1 = excelApp.Workbooks.Open(csvFileName1)  
Set excelBk2 = excelApp.Workbooks.Open(csvFileName2)  
Set excelBk3 = excelApp.Workbooks.Open(csvFileName3)  
Set excelBk4 = excelApp.Workbooks.Open(csvFileName4)  
Set excelSht1 = excelBk1.Sheets(1)  
Set excelSht2 = excelBk2.Sheets(1)  
Set excelSht3 = excelBk3.Sheets(1)  
Set excelSht4 = excelBk4.Sheets(1)  
End Sub  
  
' These 4 function to do calculation based on data collected  
Sub calData1()  
    dataDeclaration  
    dataVars("avgVal1") = Round(excelApp.WorksheetFunction.Average(excelSht1.Range("A4",  
excelSht1.Range("A" & excelSht1.Rows.count).End(-4162))), 2)  
    dataVars("stDevVal1") = Round(excelApp.WorksheetFunction.StDev(excelSht1.Range("A4",  
excelSht1.Range("A" & excelSht1.Rows.count).End(-4162))), 2)  
    dataVars("ppVal1") = Round(excelApp.WorksheetFunction.Max(excelSht1.Range("A4",  
excelSht1.Range("A" & excelSht1.Rows.count).End(-4162))), 2)  
  
    excelBk1.save  
    excelBk1.Close SaveChanges:=True  
    excelApp.Quit  
End Sub  
  
Sub calData2()  
    dataDeclaration  
    dataVars("avgVal2") = Round(excelApp.WorksheetFunction.Average(excelSht2.Range("A4",  
excelSht2.Range("A" & excelSht2.Rows.count).End(-4162))), 2)  
    dataVars("stDevVal2") = Round(excelApp.WorksheetFunction.StDev(excelSht2.Range("A4",  
excelSht2.Range("A" & excelSht2.Rows.count).End(-4162))), 2)  
    dataVars("ppVal2") = Round(excelApp.WorksheetFunction.Max(excelSht2.Range("A4",  
excelSht2.Range("A" & excelSht2.Rows.count).End(-4162))), 2)  
  
    excelBk2.save  
    excelBk2.Close SaveChanges:=True  
    excelApp.Quit
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

End Sub

Sub calData3()

```
    dataDeclaration
    dataVars("avgVal3") = Round(excelApp.WorksheetFunction.Average(excelSht3.Range("A4",
excelSht3.Range("A" & excelSht3.Rows.count).End(-4162))), 2)
    dataVars("stDevVal3") = Round(excelApp.WorksheetFunction.StDev(excelSht3.Range("A4",
excelSht3.Range("A" & excelSht3.Rows.count).End(-4162))), 2)
    dataVars("ppVal3") = Round(excelApp.WorksheetFunction.Max(excelSht3.Range("A4",
excelSht3.Range("A" & excelSht3.Rows.count).End(-4162))), 2)
```

```
    excelBk3.save
    excelBk3.Close SaveChanges:=True
    excelApp.Quit
```

End Sub

Sub calData4()

```
    dataDeclaration
    dataVars("avgVal4") = Round(excelApp.WorksheetFunction.Average(excelSht4.Range("A4",
excelSht4.Range("A" & excelSht4.Rows.count).End(-4162))), 2)
    dataVars("stDevVal4") = Round(excelApp.WorksheetFunction.StDev(excelSht4.Range("A4",
excelSht4.Range("A" & excelSht4.Rows.count).End(-4162))), 2)
    dataVars("ppVal4") = Round(excelApp.WorksheetFunction.Max(excelSht4.Range("A4",
excelSht4.Range("A" & excelSht4.Rows.count).End(-4162))), 2)
```

```
    excelBk4.save
    excelBk4.Close SaveChanges:=True
    excelApp.Quit
```

End Sub

' This function allows close the excel file

Sub exlCls()

```
    excelBk.save
    excelBk.Close SaveChanges:=True
    excelApp.Quit
```

End Sub

' This subroutine prompt user to insert their company logo

Private Sub insertLogoBtn_Click()

```
    Dim picName As String
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Dim picBox As InlineShape
Dim picRng As Range

With Application.Dialogs(wdDialogInsertPicture)
    save = .Show
    If save = 0 Then
        saveLogo = False
        Exit Sub
    Else
        picName = .Name
        ActiveDocument.Undo
        Set picRng = ActiveDocument.Bookmarks("logo").Range
        picRng.Text = vbNullString
        Set picBox = ActiveDocument.InlineShapes.AddPicture(fileName:=picName,
LinkToFile:=False, Range:=picRng)
        ActiveDocument.Bookmarks.Add "logo", picRng
        If picName <> "" Then

ActiveDocument.Sections(2).Headers(wdHeaderFooterPrimary).Range.InlineShapes.AddPicture
fileName:=picName
            ActiveDocument.ActiveWindow.ActivePane.View.SeekView =
wdSeekCurrentPageHeader
                End If
            ActiveDocument.ActiveWindow.View.SeekView = wdSeekMainDocument
            saveLogo = True
        End If
    End With
End Sub

' This subroutine check everything details to make sure no empty field left
Private Sub okBtn_Click()
    If Me.comName.Enabled = True Then
        If Me.comName.Value <> "" Then
            chkName
            If nameDetected = True Then
                Exit Sub
            End If
        End If
    End If
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
Select Case ""
    Case Me.comName.Value
        MsgBox "Please fill in your company name!", vbExclamation, "Compulsory To Fill
In!"
        Me.comName.SetFocus
        Exit Sub
    Case Me.comInfo.Value
        MsgBox "Please fill in your company info!", vbExclamation, "Compulsory To Fill
In!"
        Me.comInfo.SetFocus
        Exit Sub
    Case Me.emailAdd.Value
        MsgBox "Please fill in your email address!", vbExclamation, "Compulsory To
Fill In!"
        Me.emailAdd.SetFocus
        Exit Sub
    Case Me.phone.Value
        MsgBox "Please fill in your phone number!", vbExclamation, "Compulsory To Fill
In!"
        Me.phone.SetFocus
        Exit Sub
End Select

If saveLogo = False Then
    MsgBox "Please insert company logo!", vbExclamation, "Compulsory To Insert!"
    Exit Sub
End If

calData1
calData2
calData3
calData4
getGraphImg

Me.dataFilled = True
Me.Hide
End Sub

' This subroutine is to check the phone format
Private Sub phone_Change()
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
If Me.phone.Value = "" Then Exit Sub
tmpNo = Me.phone.Value

If (tmpNo Like "##-#####") Or (tmpNo Like "###-#####") Then
    Exit Sub
Else
    If (tmpNo Like "#####") Then
        tmpNo = Left(tmpNo, 2) & "-" & Right(tmpNo, 7)
        Me.phone.Value = tmpNo
        Exit Sub
    Else
        If Not IsNumeric(tmpNo) Then
            MsgBox "Please enter a valid number for phone number!"
            Me.phone.Value = ""
            Me.phone.SetFocus
        End If
    End If
End If
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

D-3-2 reportGenMain.bas

```
' Final Year Project - PC Based Data Logging System
'
' =====
' Name: Lui YongJian
' Course: Computer Engineering (CT)
' Student ID: 09ACB08053
' Date Last Modified: 29 July 2013
' Copyrighted (c) 2013 . All rights reserved
```

Option Explicit

```
Public excelApp As Object, excelBk As Object, excelSht As Object, dataSelected As Object
Public excelBk1 As Object, excelSht1 As Object, excelBk2 As Object, excelSht2 As Object,
excelBk3 As Object, excelSht3 As Object, excelBk4 As Object, excelSht4 As Object
Public dataVars As Word.Variables
Dim saveComplete As Boolean
Public save As Integer
```

```
' This subroutine always initialize the report template
```

```
Sub AutoNew()
    rstVar
    executeForm
End Sub
```

```
' The subroutine reset all the variable in template report to empty
```

```
Sub rstVar()
    With ActiveDocument.Variables
        .Item("companyName").Value = " "
        .Item("companyInfo").Value = " "
        .Item("phoneNo").Value = " "
        .Item("email").Value = " "
        .Item("avgVal1").Value = " "
        .Item("stDevVal1").Value = " "
        .Item("ppVal1").Value = " "
        .Item("avgVal2").Value = " "
        .Item("stDevVal2").Value = " "
        .Item("ppVal2").Value = " "
        .Item("avgVal3").Value = " "
        .Item("stDevVal3").Value = " "
        .Item("ppVal3").Value = " "
    End With
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        .Item("avgVal4").Value = " "  
        .Item("stDevVal4").Value = " "  
        .Item("ppVal4").Value = " "  
    End With  
    updateVar  
End Sub  
  
' This subroutine is to update the variable in template report with user's key in data or  
reset data  
Sub updateVar()  
    Dim varData As Word.Range  
    Dim iData As Long  
  
    iData = ActiveDocument.Sections(1).Headers(1).Range.StoryType  
    For Each varData In ActiveDocument.StoryRanges  
        Do  
            varData.Fields.Update  
            Set varData = varData.NextStoryRange  
        Loop Until varData Is Nothing  
    Next  
End Sub  
  
' This subroutine is to call the form to allow user to key in required data and insert  
into template report  
Sub executeForm()  
    Dim dataForm As inputDataForm  
  
    Set dataVars = ActiveDocument.Variables  
    Set dataForm = New inputDataForm  
  
    With dataForm  
        .Show  
        If .dataFilled Then  
            dataVars("companyName").Value = .comName.Text  
            dataVars("companyInfo").Value = .comInfo.Text  
            dataVars("phoneNo").Value = .phone.Value  
            dataVars("email").Value = .emailAdd.Text  
            If .comList.Value = "" Then  
                savePortfolio  
            End If  
        End If  
    End With  
End Sub
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
        Unload dataForm
        updateVar
        saveFile
        If saveComplete = True Then
            Application.ActiveDocument.Close False
            Application.Quit
        End If
    Else
        MsgBox "You have exit the automate report form. Thank you for using it!",
vbInformation, "Exiting the automate report...."
    End If
End With

    Set dataForm = Nothing
    Set dataVars = Nothing
End Sub

' This subroutine is allows prompt user to save file
Sub saveFile()
    With Application.Dialogs(wdDialogFileSaveAs)
        save = .Show
        If save = -1 Then
            saveComplete = True
        End If
    End With
End Sub

' This subroutine is to save the user data into excel for future retrieve
Sub savePortfolio()
    Dim dataSelected As Range
    Dim dataRow As Long, no As Long
    dataRow = 5
    no = 1

    Set excelApp = CreateObject("Excel.Application")
    Set excelBk =
excelApp.Workbooks.Open("E:\data_logging\report_gen\database_portfolioCompanyList.xlsx")
    Set excelSht = excelBk.Sheets(1)

    While excelSht.Range("B" & dataRow) <> ""
```

APPENDIX D CODES FOR PROJECT FILES FUNCTION

```
no = no + 1
dataRow = dataRow + 1
excelSht.Range("B" & dataRow).Select
Wend

With ActiveDocument.Variables
    excelSht.Range("B" & dataRow).Value = no
    excelSht.Range("C" & dataRow).Value = .Item("companyName").Value
    excelSht.Range("D" & dataRow).Value = .Item("companyInfo").Value
    excelSht.Range("E" & dataRow).Value = .Item("phoneNo").Value
    excelSht.Range("F" & dataRow).Value = .Item("email").Value
End With

excelBk.save
excelBk.Close
excelApp.Quit
End Sub
```

APPENDIX E COMPANY PORTFOLIO LIST DATABASE EXCEL

Company Portfolio List Database

No.	Company Name	Company Info	Telephone No.	Email Address
1	Agilent Technologies	Business of Chemical Analysis, Life Science, Diagnostics and Genomics and Electronic Measurement	03-7727880	ccc-smt@agilent.com
2	Pico Technology	PC Oscilloscope and Data Acquisition Products	03-7005912	sales@picotech.com
3	KKI INSTRUMENTS (M) SDN BHD	Sales of MSR Data Logger	07-4352540	sales@meter.com.my
4	CAS Dataloggers	Distributor of data loggers, paperless recorders and data acquisition equipment	03-7001022	support@dataloggerinc.com
5	ACR System Inc.	Designing and building data loggers	04-5911128	sales@acrsystems.com

Figure E-F1 Company Portfolio List Database

APPENDIX F AUTOMATION REPORT

APPENDIX F AUTOMATION REPORT



PC BASED DATA LOGGING REPORT

Company Name: Pico Technology - PC Oscilloscope and Data Acquisition Products

Tel No: 03-7005912 Email Address: sales@picotech.com

Date Created: 11 August 2013

Figure F-F1 Automation Report Front Cover

APPENDIX F AUTOMATION REPORT

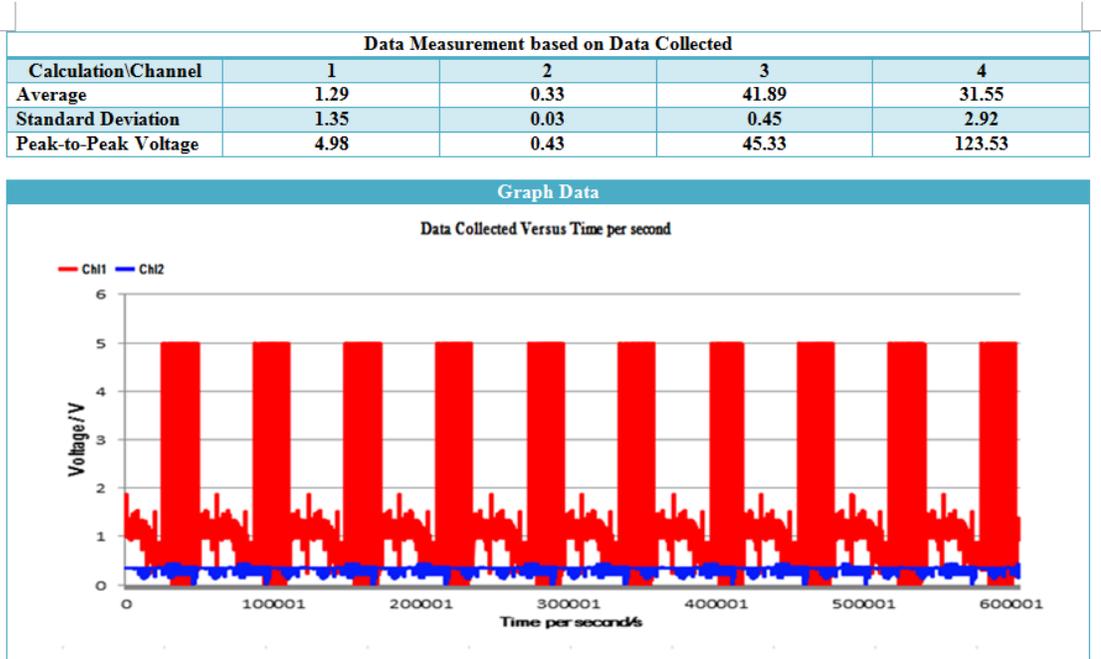


Figure F-F2 Automation Report Data Measurement and Graph Channel 1 and Channel 2

APPENDIX F AUTOMATION REPORT

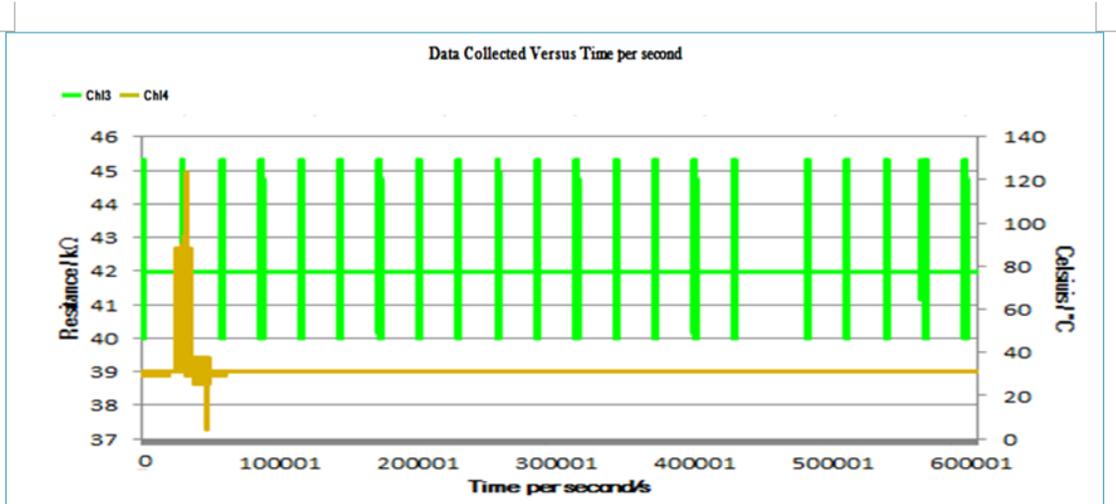


Figure F-F3 Automation Report Graph Channel 3 and Channel 4

APPENDIX F AUTOMATION REPORT

Data Collected File			
data_1.csv	data_2.csv	data_3.csv	data_4.csv
			
* Please click on excel icon to open the data files.			

Figure F-F4 Automation Report Last Page