

**SENSOR WIRELESS INTERFACE MODULE**

By

**SAIFUL IMRAN BIN BERAHIM**

**FINAL PROJECT REPORT**

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

**Universiti Teknologi Petronas  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan**

**© Copyright 2006  
by  
Saiful Imran bin Berahim, 2006**

# **CERTIFICATION OF APPROVAL**

## **SENSOR WIRELESS INTERFACE MODULE**

by

Saiful Imran Bin Berahim

A project dissertation submitted to the  
Electrical & Electronics Engineering Programme  
Universiti Teknologi PETRONAS  
in partial fulfilment of the requirement for the  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

Approved:



---

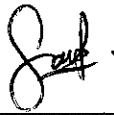
Ms. Suhaila Badarol Hisham  
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

December 2006

## **CERTIFICATION OF ORIGINALITY**

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



---

**Saiful Imran Bin Berahim**

## **ABSTRACT**

In the oil and gas industry, wired sensor network is the traditional and commonly used. Nowadays, there is a new trend in network communication which is the wireless sensor network. Hence, the wireless technology is now growing fast and keeps improving with time. The objectives of this project are to find an economical wireless solution for the wireless sensor network, design, build and test a short-range sensor wireless interface module. A lot of study and research in communications and instrumentation has to be made in order to design the system. The designed system consists of a RTD transmitter, A/D converters, signal interface, a WiFi Access Point and a WiFi enabled PC. Based on the surveys conducted in this project, the WiFi technology is suitable for this project because of its maximum transfer speed, maximum distance and maximum devices that it can connect at one time. This system also can reduce maintenance cost and save time.

## **ACKNOWLEDGEMENTS**

Alhamdulillah, with the greatest gratitude to the Almighty Allah for his gracious blessings throughout the whole period this project was undertaken. My deepest gratitude to my parents, who have provided me with their love and undivided attention through the sweet and sour moments. To my supervisor, Ms Suhaila Badarol Hisham, thank you so much for your guidance and support throughout this project. Her words of wisdom have encouraged me to rise again during the times I fall.

I would also like to express my sincere appreciation to the lecturers' involved, especially Mr Azlan Awang who has helped me regarding telecommunications. I have learnt a lot from him. To Mrs Siti Hawa, thank you so much for your cooperation that has helped me a lot in the lab. My gratitude also extends to all the lab technicians whose presences are truly meaningful and valuable.

I would like to take this opportunity to thank all my friends who has contributed to this project. Your assistance is duly acknowledged and noted.

# TABLE OF CONTENTS

<b>LIST OF TABLES .....</b>	<b>ix</b>
<b>LIST OF FIGURES .....</b>	<b>x</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>xi</b>
<b>CHAPTER 1 INTRODUCTION .....</b>	<b>1</b>
1.1 Background Of Study.....	1
1.2 Problem Statement .....	2
1.3 Objectives And Scope Of Studies.....	3
<b>CHAPTER 2 LITERATURE REVIEW .....</b>	<b>4</b>
2.1 The Wireless Sensor Networks .....	4
2.2 Bluetooth.....	5
2.3 WiFi .....	5
2.4 RFID.....	5
2.5 ZigBee .....	5
2.6 Microwave .....	6
<b>CHAPTER 3 METHODOLOGY .....</b>	<b>7</b>
3.1 Flow Of The Project.....	7
3.2 Comparing The Available Wireless Technology.....	9
3.3 Tools Used .....	12
3.4 Serial Interface .....	13
3.5 Microcontroller .....	13
3.5.1 Microcontroller Applications.....	13
3.5.2 PIC16F84 Microcontroller .....	15
3.5.3 PIC16F84 Microcontroller Circuitry.....	17
3.5.4 PIC16F84 Microcontroller Programming .....	18
3.6 Microsoft Visual Basic Environment.....	19
3.6.1 Microsoft Comm Control 6.0.....	21
3.7 PC Communication: Serial Port Interface.....	22
3.7.1 Serial Communication Circuitry .....	23
3.8 The ADC0804 Circuit Test .....	25

3.9	The System Design .....	27
3.10	TCP-Com Software.....	29
3.10.1	TCP-Com Settings .....	29
<b>CHAPTER 4 RESULTS &amp; DISCUSSIONS.....</b>		<b>32</b>
4.1	Hyperterminal Settings .....	32
4.2	The ADC Outputs .....	35
4.3	The Visual Basic Interface Program .....	37
4.4	Testing Range Of The WiFi Transmission.....	38
4.5	Problems Encountered .....	39
<b>CHAPTER 5 CONCLUSION &amp; RECOMMENDATIONS.....</b>		<b>41</b>
5.1	Conclusion .....	41
5.2	Recommendations.....	41
<b>REFERENCES.....</b>		<b>43</b>
<b>APPENDICES .....</b>		<b>45</b>
APPENDIX A	C CODES FOR PIC MICROCONTROLLER.....	46
APPENDIX B	VISUAL BASIC 6 PROGRAMMING .....	49
APPENDIX C	OVERALL CIRCUIT DIAGRAM .....	53
APPENDIX D	ASCII CODES TABLE.....	54
APPENDIX E	PIC16F84 DATASHEET .....	55
APPENDIX F	ADC0804 DATASHEET .....	56
APPENDIX G	MAX232 DATASHEET .....	57
APPENDIX H	RTD TRANSMITTER DATASHEET .....	58

## LIST OF TABLES

Table 3.1:	Table Of Comparison.....	10
Table 3.2:	Table Of Bluetooth And Wifi Comparison.....	11
Table 3.3:	Visual Basic Naming Conventions .....	20
Table 4.1:	Outputs Of The ADC0804 .....	36
Table 4.2:	Range And Connection Testing .....	38



## **LIST OF FIGURES**

Figure 2.1:	Wireless Sensor Network .....	4
Figure 3.1:	Flow Chart Of Project Work .....	8
Figure 3.2:	Serial Interface Details .....	13
Figure 3.3:	PIC16F84 Pin Out .....	15
Figure 3.4:	Components Of A Microcontroller.....	16
Figure 3.5:	PIC16F84 Circuit Diagram.....	17
Figure 3.6:	Visual Basic Editor.....	19
Figure 3.7:	Serial Port (9 Pin) .....	22
Figure 3.8:	RS-232 Pin Out On A DB-9 Pin Used For Asynchronous Data .....	24
Figure 3.9:	Serial Interface Circuit.....	25
Figure 3.10:	Free-Running Connection.....	26
Figure 3.11:	Overall Block Diagram Of The Project System .....	28
Figure 3.12:	Setting For The Transmitter Side .....	30
Figure 3.13:	Setting For The Receiver Side.....	31
Figure 4.1:	The VB Interface Program .....	37

## **LIST OF ABBREVIATIONS**

<b>CMOS</b>	<b>Complimentary Metal Oxide Semiconductor</b>
<b>CPU</b>	<b>Central Processing Unit</b>
<b>DOS</b>	<b>Disk Operating System</b>
<b>EPROM</b>	<b>Erasable Programmable Read Only Memory</b>
<b>GUI</b>	<b>Graphical User Interface</b>
<b>I/O</b>	<b>Input/Output</b>
<b>IC</b>	<b>Integrated Circuit</b>
<b>LED</b>	<b>Light Emitting Diode</b>
<b>OLEDB</b>	<b>Object Link Embedded Database</b>
<b>PC</b>	<b>Personal Computer</b>
<b>PIC</b>	<b>Programmable Integrated Circuit</b>
<b>PWM</b>	<b>Pulse Width Modulation</b>
<b>QBE</b>	<b>Query By Example</b>
<b>RAM</b>	<b>Random Access Memory</b>
<b>RDBMS</b>	<b>Relational Database Management System</b>
<b>RISC</b>	<b>Reduced Instruction Set Computer</b>
<b>RXD</b>	<b>Receive</b>
<b>SCI</b>	<b>Serial Communication Interface</b>
<b>SQL</b>	<b>Structured Query Language</b>
<b>TTL</b>	<b>Transistor-Transistor Logic</b>
<b>TXD</b>	<b>Transmit</b>
<b>UART</b>	<b>Universal Asynchronous Receiver/Transmitter</b>

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Background of Study**

Instrumentation is defined as "the art and science of measurement and control". Instrumentation can be used to refer to the field in which Instrument technicians and engineers work or it can refer to the available methods of measurement and control and the instruments which facilitate this. There are many kinds of instrument that are related to the instrumentation field. Instrumentation engineers largely design control and measurement systems primarily for industrial processes, although instruments are present in almost any modern industrial applications today.

Instrumentation plays a big role in the oil and gas industry. Most of the instruments are used for measurement and control. Measurement and control are very important in the industry to maintain the process of the plant at desired condition. In order to control, they need instrument that can measured the real value at the plant and compare it to the desired values. In the oil and gas industry, there are a lot of instruments installed in the plants. These instruments are generally connected to some sort of control system which responds to the instrument measurement. The response programmed into the control system manipulates control devices attached to the process. This cycle of changing of manipulated variables, measurement of controlled variables and proper response is the basic concept behind process control. The control systems used are often considered part of the field of instrumentation, and can include simple Programmable Logic Controllers (PLC) or Remote Terminal Units (RTU), or Distributed Control Systems (DCS). Inputs can vary from a few measured variables, to thousands of measured variables.

Wired sensor networks have been around for decades, with an array of gauges measuring temperature, fluid levels, humidity and other attributes on pipelines, pumps, generators and manufacturing lines. Many of these run as separate wired networks, sometimes linked to a computer. Now advances in silicon radio chips, coupled with cleverly crafted routing algorithms and network software are promising to eliminate those wires, and their installation and maintenance costs. This is where the idea of wireless network starts to work and been developed.

## **1.2 Problem Statement**

Most practitioners in oil and gas industry prefer to use hardwired sensors. However, there are several issues of concern on the hardwired sensors:-

*a) Space and Costs*

Wired sensors consume more space and high cost for maintenance because cables are laid on trays and ducts for routing all over the plant.

*b) Distance*

Wired sensors are normally located far from the control room and cables have to be routed all the way from the field to the operator room.

*c) The cable length*

The length of the cable affects the signal drop. If the signal drop is large, it will affect the signals.

*d) Maintenance difficulties*

Wired sensors are more difficult to maintain because technicians have to sift through bundles of cable and identify the plant documentations. This cost the company a lot of money, time and effort.

*e) Difficult areas*

Wired system network cannot reach some of the difficult areas. For example, cables cannot withstand very high temperature at the boiler area. Sensor

devices have the ability to withstand the very high temperature using special certified enclosure but not the cables.

This project aims to reduce costs with the same reliability of the wired technology system which is very demanding in the oil and gas industries.

### **1.3 Objectives And Scope Of Studies**

The main objectives of this project are:-

- a) To find an economical wireless solution for the wireless sensor network in terms of cost and time.
- b) To design a wireless interface module system.
- c) To build the wireless interface module system based on the design.
- d) To test the short-range sensor wireless interface module on a RTD transmitter.

In this project, the scopes of study are:-

- a) study of wireless communication technology,
- b) designing a wireless system,
- c) assembling the components for the supporting circuits and
- d) testing the system by connecting it to one of the RTD transmitter available in the EE Instrument Control Lab.

Objective of the project can be achieved as long as the project is on track and quick action is taken when there is a problem. In the oil and gas industry, the wireless system will help engineers and operators to monitor and maintain the plant process as the system is more economical.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 The Wireless Sensor Networks

The wireless sensor networks can use several different wireless technologies, including IEEE 802.11 WiFi, Bluetooth and radio frequency identification (RFID). But right now most of the action is with low-power radios that have a range of about 30 to 200 feet and data rates of up to around 300K bit/sec. Most of these, with their accompanying network software and APIs, are proprietary products. [1]

But the IEEE last year approved the 802.15.4 low-rate standard for a simple, short-range wireless network whose radio components could run several years on a single battery. Below is the diagram that shows what the wireless network need and how they are connected to each other. [2]

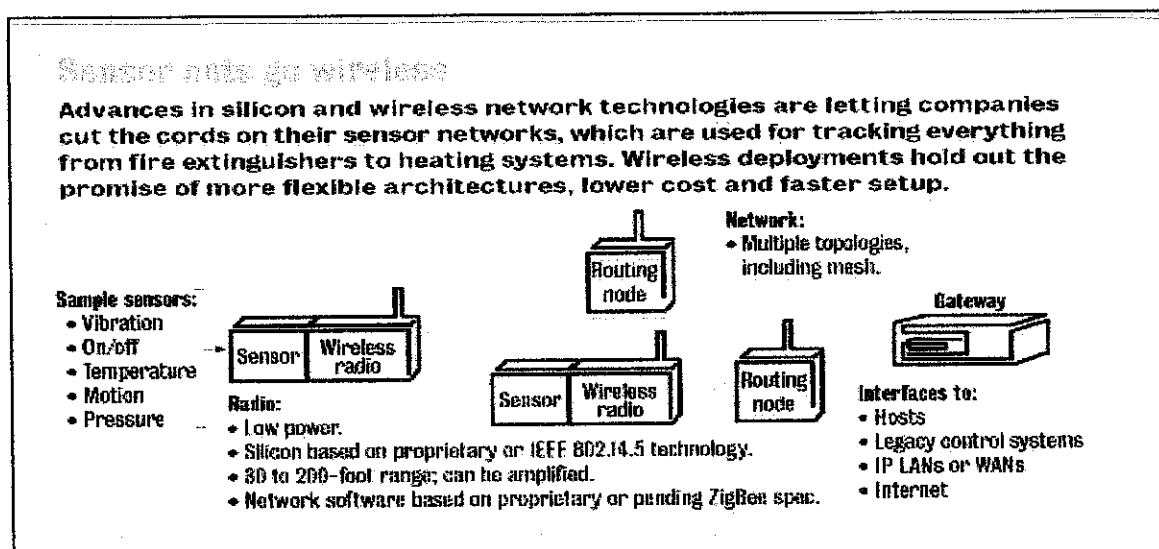


Figure 2.1: Wireless Sensor Network [1]

## **2.2 Bluetooth**

Bluetooth is a low-power wireless technology intended to replace cables and wires and let multiple devices discover and talk to each other at speeds up to 1M bit/sec over a range of roughly 30 feet. [3]

## **2.3 WiFi**

WiFi is the wireless way to handle networking. It is also known as 802.11 networking and wireless networking. The big advantage of WiFi is its simplicity. We can connect computers anywhere in your home or office without the need for wires. The computers connect to the network using radio signals, and computers can be up to 100 feet or so apart. [4]

## **2.4 RFID**

RFID operates in a low frequency range of 100 kHz-1.5 MHz or a high frequency range of 900 MHz-2.4GHz, which has an operating range up to 30m. RFID tags are very inexpensive, and are used in manufacturing and sales inventory control, container shipping control, etc. RFID tags are installed on water meters in some cities, allowing a metering vehicle to simply drive by and remotely read the current readings. They are also be used in automobiles for automatic toll collection. These networks used RF modems to convert the RS-232 signal to a radio signal and back again. Fluke (Everett, Washington) developed a digital voltmeter that could be configured to accept a voltage signal and transmit the signal over a dedicated radio frequency channel. [5]

## **2.5 ZigBee**

ZigBee is a published specification set of high level communication protocols designed to use small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). The relationship between IEEE 802.15.4 and ZigBee is analogous to that existing between IEEE 802.11 and the WiFi

Alliance. The ZigBee 1.0 specifications were ratified on December 14, 2004 and are available to members of the ZigBee Alliance. For non-commercial purposes, the ZigBee specification is available to the general public at the ZigBee Alliance homepage. The technology is designed to be simpler and cheaper than other WPANs such as Bluetooth. The most capable ZigBee node type is said to require only about 10% of the software of a typical Bluetooth or Wireless Internet node, while the simplest nodes are about 2%. However, actual node sizes are much higher, more like 50% of Bluetooth code size. ZigBee chip vendors announced 128-kilobyte devices. [6]

## 2.6 Microwave

Microwaves are electromagnetic waves with wavelengths longer than those of infrared light, but relatively short for radio waves. Microwaves have wavelengths approximately in the range of 30 cm (frequency = 1 GHz) to 1 mm (300 GHz). However, the boundaries between far infrared light, microwaves, and ultra-high-frequency radio waves are fairly arbitrary and are used variously between different fields of study. A credible definition comes from Pozar's text "*Microwave Engineering*", which states that the term microwave "refers to alternating current signals with frequencies between 300MHz ( $3 \times 10^8$  Hz) and 300GHz ( $3 \times 10^{11}$  Hz)." The existence of electromagnetic waves, of which microwaves are part of the higher frequency spectrum, was predicted by James Clerk Maxwell in 1864 from his famous Maxwell's equations. In 1888, Heinrich Hertz was the first to demonstrate the existence of electromagnetic waves by building apparatus to produce radio waves. The microwave range includes ultra-high frequency (UHF) (0.3-3 GHz), super high frequency (SHF) (3-30 GHz), and extremely high frequency (EHF) (30-300 GHz) signals. Above 300 GHz, the absorption of electromagnetic radiation by Earth's atmosphere is so great that it is effectively opaque, until the atmosphere becomes transparent again in the so-called infrared and optical window frequency ranges. [7]



## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Flow of the Project**

The flowchart below summarizes the project methodology for this semester and next semester. The project methodology consists of:-

*a) Preliminary research*

This is where studies and research from the internet, books and supervisor are done to get the idea to design the system. In this project, research and studies in wireless communication technology are done throughout the first semester.

*b) System Design*

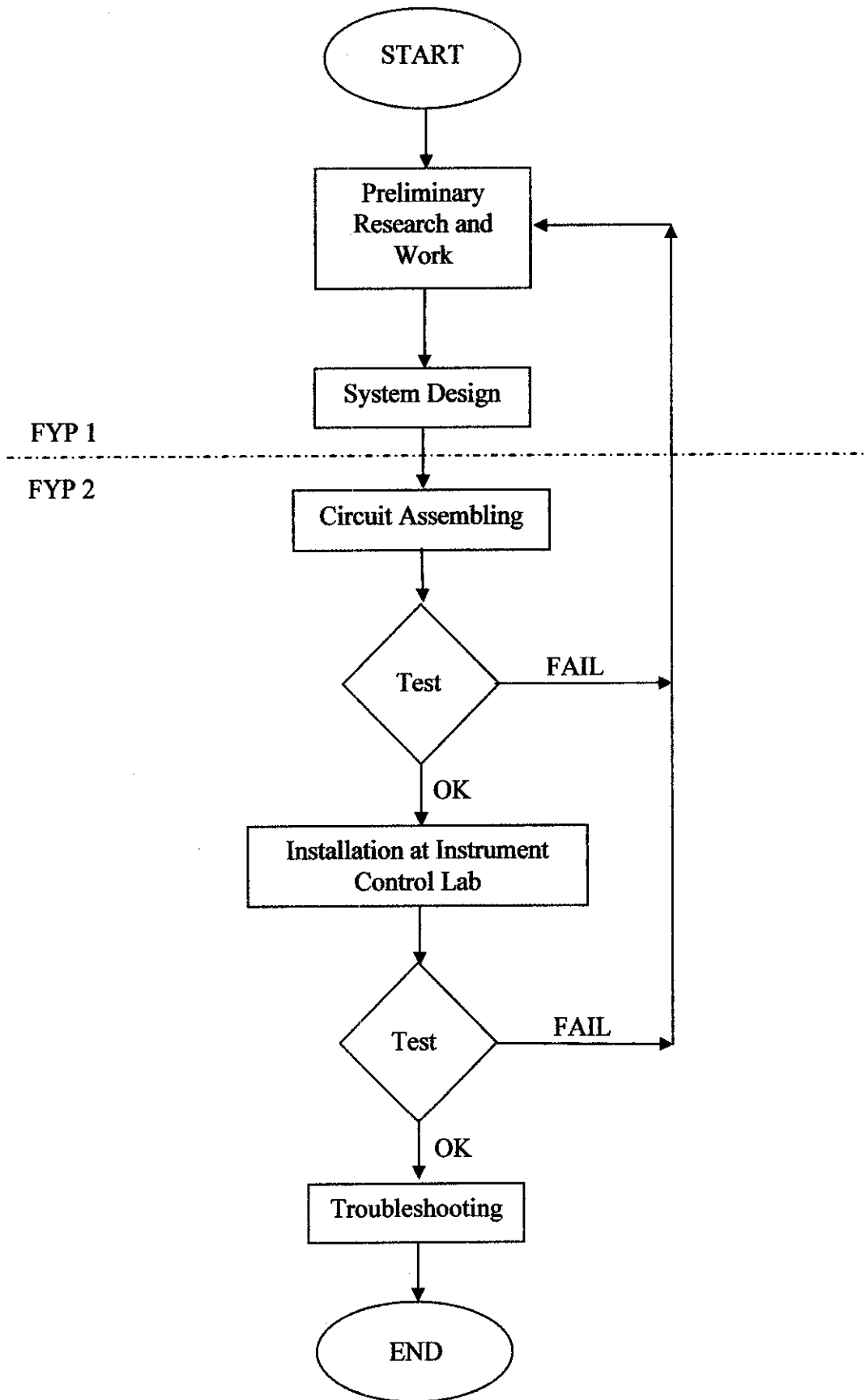
System design is where designing the system and simulates the supporting circuits using software like PSpice and EWB are done. In this stage, the important components of the system are identified carefully.

*c) Circuit Assembling*

In circuit assembling phase, lab works like fabricating the PCB and soldering the components of the circuit are done throughout the second semester.

*d) Installation and Testing*

After assembling the circuits, it must be install at the EE Instrument Control Lab where the circuit is connected to a RTD transmitter as the input. Configuration of the WiFi enabled PC will take place after that.



**Figure 3.1: Flow Chart of Project Work**

### **3.2 Comparing the Available Wireless Technology**

In this project, a lot of findings and research are made about the latest wireless techniques that are used today. Each technology has its own advantages and disadvantages. The technology to be used is selected based on the applications of interest. Based on findings from the internet and books, several wireless technologies have been identified. There are:-

- a) WiFi
- b) Bluetooth
- c) ZigBee
- d) Infrared (IR)
- e) Radio Frequency (RF)

Each of them is evaluated according to several criteria such as:-

- a) the maximum transfer speed that can be achieved
- b) maximum distance
- c) maximum devices that it can connect or communicate to
- d) frequency band used
- e) application used

Wireless Technology	Criteria				
	Frequency Band / Wavelength Used	Transfer Speed	Maximum Distance	Maximum Devices	Applications
<b>Bluetooth</b>	2.45 Ghz	Version 1.0 – up to 1 Mbps Version 2.0 – up to 3 Mbps	Up to 10m	8 devices simultaneously	It has been set aside by international agreement for the use of industrial, scientific and medical devices.
<b>WiFi</b>	IEEE 802.11b/g – 2.4Ghz  IEEE 802.11a – 5Ghz	802.11b – up to 11Mbps  802.11a/g – up to 54Mbps	Up to 300m in all directions	100 devices at one time	High Speed Wireless Access Network
<b>Infrared (IR)</b>	Wavelength = 700nm – 1mm	2.4Kbps – 16Mbps	10m	One at a time – “one-to-one technology”	Used in remote control systems
<b>ZigBee</b>	IEEE 802.15.4.2. 4 Ghz	250Kbps	10-75m	100 by mesh networks	Wireless Personal Area Network (WPAN)
<b>Radio Frequency (RF)</b>	Frequency = 10kHz – 300 000 MHz	300Kbps	10-60m	10 devices	Mobile phones (W-CDMA), WLAN, most modern Radars

**Table 3.1: Table of Comparison**

After discovering many types of wireless technology and comparing each of them, it is found that WiFi and Bluetooth are applicable to this project. Next, these two technologies are compared based on the transfer speed, application that are suitable to be used, security, ease of use, power consumption, its compatibility, maximum ranges and devices that it can connect to.

<b>Technologies</b>	<b>Bluetooth</b>	<b>WiFi</b>
<b>Criteria</b>		
<i>Speed</i>	Operates at 720kbps	Operates at 11Mbps
<i>Applications</i>	Cable replacement designed to connect devices point-to-point	Designed to hook up an entire network and also point-point devices.
<i>Security</i>	It is secure because it is widely used to connect the PC accessories such as the printer and speaker.	Although WiFi has much security threats. The security features are improving from time to time. It has several layers of security and firewall to protect users from the unwanted trouble.
<i>Ease of Use</i>	Bluetooth devices “advertise” their capabilities to others, and a single device can be connected to up to seven other devices at the same time. This makes it easy to find and connect to the device you are looking for or to switch between devices, such as two printers.	WiFi is a bit more complex. It requires the same degree of network management as any comparable wired network.
<i>Power</i>	Bluetooth has a smaller power requirement	WiFi has a bit more power requirement than Bluetooth
<i>Compatibility</i>	Bluetooth is more likely to interfere with WiFi than vice versa.	WiFi share the same band of frequencies with Bluetooth and could interfere with one another.
<i>Maximum range</i>	10 m	300 m
<i>Maximum devices</i>	Up to 8 devices at a time	Up to 100 devices at a time

**Table 3.2:** Table of Bluetooth and WiFi Comparison

Based on Table 3.2, both of the technology are comparable except for speed and maximum ranges. The WiFi technique is more beneficial than the Bluetooth for the project because it has very high speed transfer (11Mbps), maximum range (300 m) and the maximum devices that it can connect to (100 devices) compared to Bluetooth.

### **3.3 Tools Used**

In this project, the tools that are needed for the system are:-

**Hardware:**

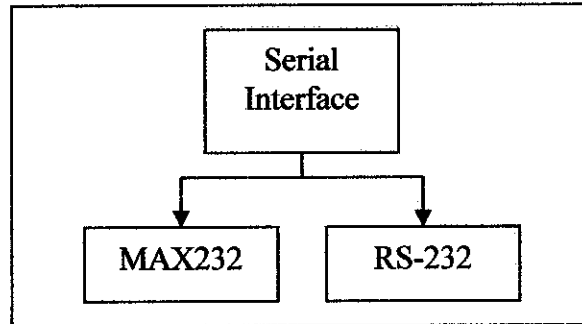
- a) Host Device (PC) – use as transmitter and receiver
- b) Sensor Devices (Yokogawa RTD transmitter) – the input
- c) Power Supplies – supply voltage for the system
- d) Cables (RS-232 and Ethernet) – connect between the PC and Access Point
- e) Wireless Device (e.g. WiFi Access Point) – transmit the signals wirelessly
- f) Oscilloscope (Optional) – to check the circuit for faulty

**Softwares:**

- a) Warp 13 – HEX code compiler
- b) TCP-COM by TALTec – convert serial data into TCP/IP data
- c) Microsoft Visual Basic 6 – design the interface
- d) PIC Programmer – to program the PIC microcontroller

### 3.4 Serial Interface

This interface comes in between the microcontroller circuitry and the serial port of the PC. The details are shown in the figure below.



**Figure 3.2:** Serial interface details

Basically, the data transmitted from the microcontroller will be managed by two important components. They are:

- MAX232
- RS-232

The MAX232 is a component that converts TTL logic levels (0V to 5V) to swing between 12V to -12V. From here, it will go through RS-232 to complete the communication with the serial port of the PC. This serial interface I used to connect the sensor to the PC to be transmitted using the WiFi Access Point.

### 3.5 Microcontroller

#### 3.5.1 Microcontroller Applications

The evolution of the microprocessor technology is marked by a major fork. One branch of the evolutionary tree is represented by Pentium chips and Power-PC chips which form the heart of personal computers. This mentioned branch is mainly focused upon enhancing raw computers. Meanwhile, the other branch, which is 30 times larger in unit volume, is represented by microcontrollers. Basically, in

microcontroller technology, the emphasis is upon the integration of many features needed in a single chip so that the chip has versatility to sense inputs and control outputs in a device or instrument. Mainly, the 8-bit microcontroller market is dominated by Motorola and Microchip Technology. All these devices are optimized to meet stringent performance requirements in cost-effective applications. [8]

After evolution in chip manufacturing, this embedded technology is no stranger is the daily-used applications. Generally, they can be divided to 5 broad markets:

- a) Consumer segment – home appliances and home entertainment
- b) Automotive – a modern car has nearly 50 microcontrollers providing intelligence and control, like keyless entry, antilock braking and air bas.
- c) Office automation – PCs, keyboards, copiers, printers etc
- d) Telecommunications – cell phones, pagers and answering machines
- e) Industrial products – door locks in hotel rooms, automatic faucets and industrial machinery

Among the main features that contribute to the wide popularity of microcontrollers are:

- a) Speed
- b) Instruction set simplicity
- c) Integration of operational features
- d) Programmable timer options
- e) Interrupt control
- f) Powerful output pin control
- g) I/O port expansion
- h) Serial programming via two pins
- i) EPROM support
- j) Mail-order support
- k) Free assembler and simulator



### 3.5.2 PIC16F84 Microcontroller

The PIC16F84 belongs to the family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers. All PICmicro™ microcontrollers employ an advanced RISC architecture. PIC16F84 have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with a separate 8-bit wide data bus. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set is used to achieve a very high performance level. [8]

PIC16F84 microcontrollers typically achieve a 2:1 code compression and up to a 4:1 speed improvement (at 20 MHz) over other 8-bit microcontrollers in their class. The PIC16F84 has up to 68 bytes of RAM, 64 bytes of Data EEPROM memory, and 13 I/O pins. A timer/counter is also available.

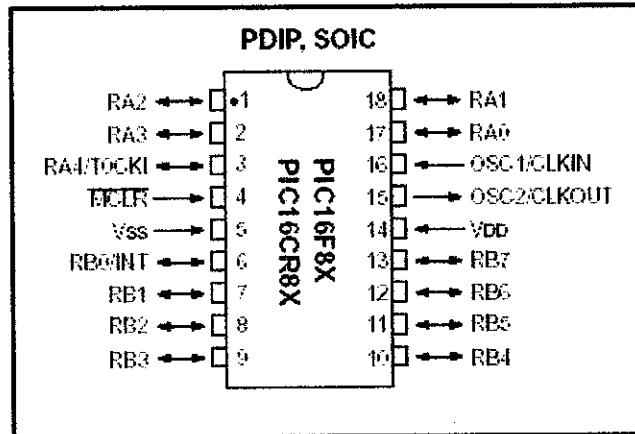
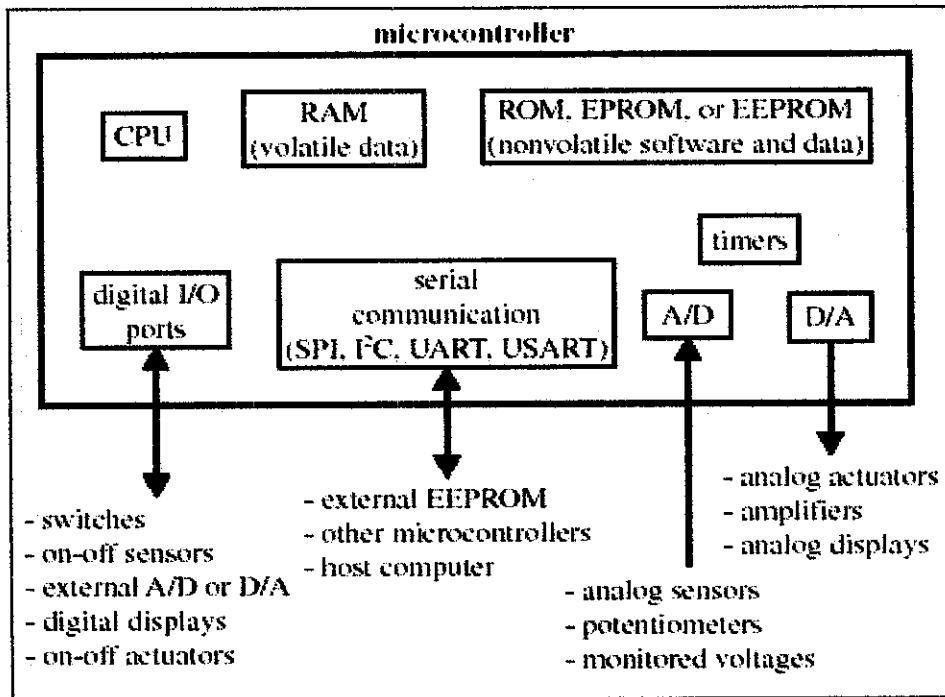


Figure 3.3: PIC16F84 pin out [8]

There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals.

The devices with Flash program memory allow the same device package to be used for prototyping and production. In-circuit reprogrammability allows the code to be

updated without the device being removed from the end application. This is useful in the development of many applications where the device may not be easily accessible, but the prototypes may require code updates. This is also useful for remote applications where the code may need to be updated (such as rate information). A simplified block diagram of the PIC16F84 is shown in Figure 3.4.



**Figure 3.4:** Components of a Microcontroller [8]

The PIC16F84 fits perfectly in applications ranging from high speed automotive and appliance motor control to low-power remote sensors, electronic locks, security devices and smart cards. The Flash or EEPROM technology makes customization of application programs such as transmitter codes, motor speeds, receiver frequencies and security codes extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease-of-use and I/O flexibility make the PIC16F84 very versatile even in areas where no microcontroller use has been considered before (example: timer functions; serial communication; capture, compare and PWM functions; and co-processor applications).

The serial in-system programming feature (via two pins) offers flexibility of customizing the product after complete assembly and testing. This feature can be used to serialize a product, store calibration data, or program the device with the current firmware before shipping.

### 3.5.3 PIC16F84 Microcontroller Circuitry

The microcontroller circuitry must be well designed for optimum usage. All the eight pins at PORT B are used for the inputs from the sensors. Meanwhile, the five pins at PORT A are used for the outputs and serial transmit pins.

From this circuit, the two pins on the left of the PIC16F84 are for the LEDs. The two pins on the right of the PIC16F84 are reserved for serial communication. The configuration is designed as such to provide a neat and tidy connection when constructed on the circuit board. This allows easier troubleshooting should there be any unexpected problems faced at any time.

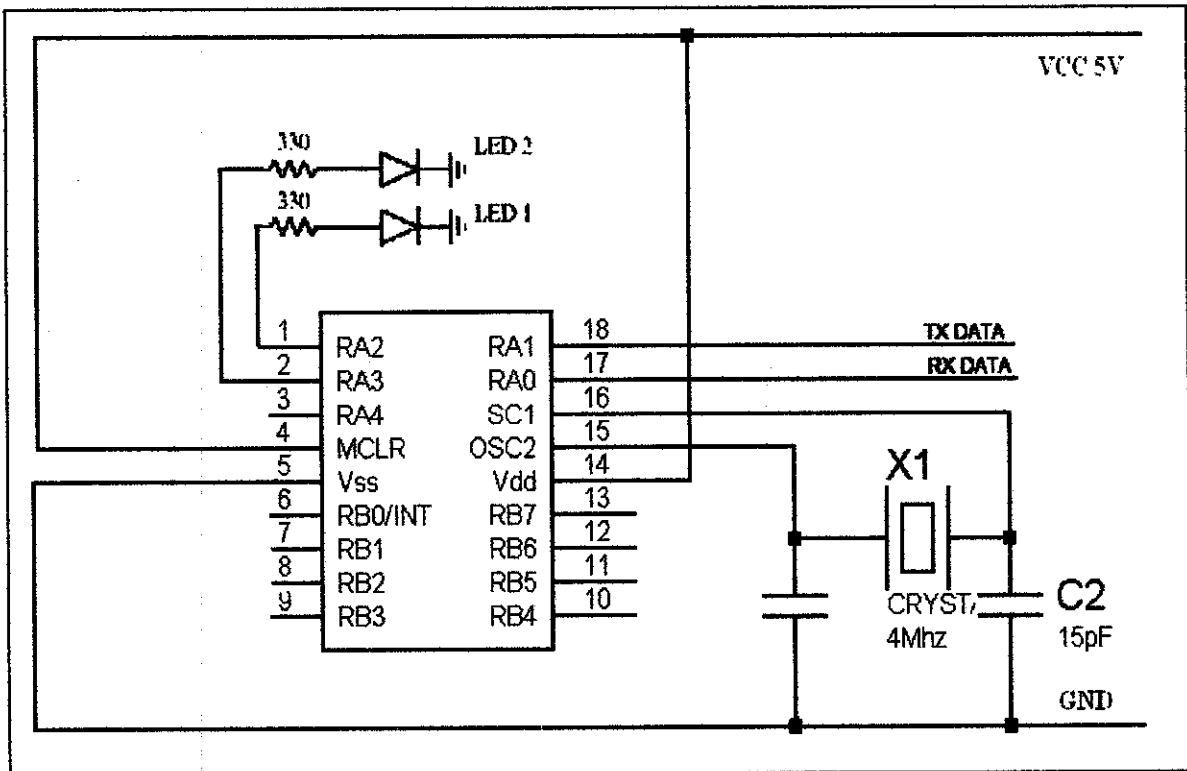


Figure 3.5: PIC16F84A Circuit Diagram

### 3.5.4 PIC16F84 Microcontroller Programming (refer Appendix A)

The C programming language is chosen to program the PIC16F84. The program will flow in such a way that the PIC16F84 will gather all the eight inputs into the PIC and produce an output that includes transmitting data output to the PC and vice versa. In order to run this program smoothly, several important initializations must be included. They are:

- PIC16F84 header file
- Declare the desired fuses
- Declare the usage of RS-232 which includes the baud rate, receive and transmit pins
- Define the delay or the clock oscillation speed

```
#include <16f84a.h>
#fuses XT,NOPROTECT,NOWDT

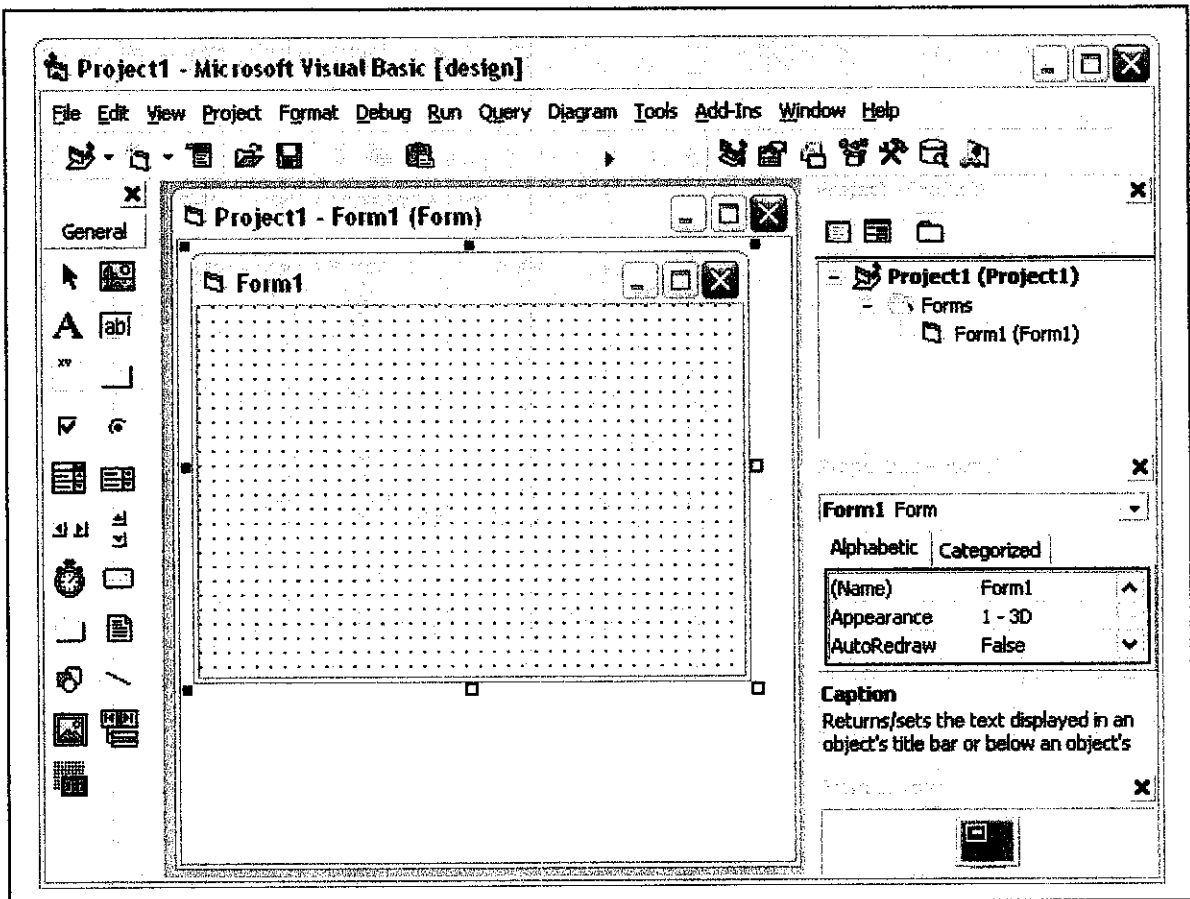
#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_A0, rcv=PIN_A1)
```

Initially, a flowchart of the whole operation is designed to guide the coding and the flow of the program. This flowchart shapes the programming codes in the C programming.

Before downloading a program into a PIC, the program must be verified and compiled with the C Compiler. The program will be converted into a hex file. Then, the EPROM of the PIC must be cleared and blanked. Once this stage is successful, the program can be downloaded into the PIC. This was done using the WARP-13 PIC programmer.

### 3.6 Microsoft Visual Basic Environment

The Microsoft Visual Basic is known to be among the most popular choice to create Windows GUI. In Visual Basic, new windows created are called forms. Elements (such as text boxes and buttons) that are placed inside a form are called controls. The Visual Basic allows event-driven programming, where the user's actions cause events, and each event in turn triggers a procedure that is associated with it.



**Figure 3.6:** Visual Basic Editor

The properties of creating an object model in Visual Basic are:

- a) Objects have properties and methods
- b) Forms and controls are examples of objects
- c) Properties describe an object. Examples: name, colour, size, or how they will behave
- d) Methods are actions associated with an object. Examples: move, clear and print

In order to write a proper Visual Basic project, there several important elements to learn and understand. The two vital steps are:

1. *Planning*

- Design the user interface
- Plan the properties
- Plan the Basic code - procedures are associated with the events, actions written in pseudo code

2. *Programming*

- Define the user interface - define objects
- Set the properties
- Write the Basic code

<b>Object</b>	<b>Prefix</b>
Form	frm
Command Button	cmd
Text Box	txt
Label	lbl
Option Button	opt
CheckBox	chk
Frame	fra
Horizontal Scrollbar	hsb
Vertical Scrollbar	vsb
Image	img
Picture Box	pic
Combo Box	cbo
List Box	lst
Shape	shp

**Table 3.3: Visual Basic Naming Conventions**

### 3.6.1 Microsoft Comm Control 6.0 (Refer to Appendix B)

Software on the computer is another essential part of the project. The software will interpret the data transferred from the PIC from the transmitter and then control the application based on the data from the transmitter. Software that is going to read from the serial port is designed using Visual Basic. The software is windows based and is very user friendly. It is object oriented programming.

Designing in Visual Basic environment will base on the object created. In this project a communication protocol like in Hyper Terminal is created and all the previous setting is duplicated into the Visual Basic form. Previously in Hyper Terminal, there is no user interface thus making the program not user friendly. In Visual Basic therefore the user interface is going to be tailored with graphic to its application. This will make the software more usable and friendly to user with the introduction of graphic.

In order for Visual Basic to establish connection with the serial port, a component called the Microsoft Comm Control 6.0 is applied. This component is represented by an icon of a phone.

The portion of the source codes that initializes the serial port using the Microsoft Comm Control 6.0 is presented below:

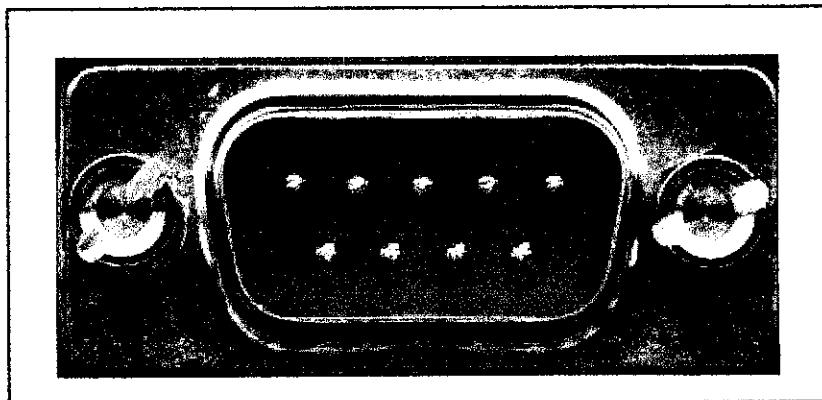
```
Private Sub Form_Load()  
    'Initialize  
    Comm.InputMode = 0 'take ASCII codes as inputs  
    Comm.CommPort = 1  
    Comm.Settings = "9600,N,8,1"  
    Comm.PortOpen = True 'open port  
    Comm.InputLen = 1 'limitation for input  
    Comm.RThreshold = 1  
End Sub
```

The COM settings will coincide with the serial port settings which are:

- Baud rate: 9600
- Parity bits: None
- Data bits: 8
- Stop bit: 1

### 3.7 PC Communication: Serial Port Interface

The serial port is an I/O device. An I/O device is just a way to get data into and out of a computer. There are many types of I/O devices such as serial ports, parallel ports, disk drive controllers, Ethernet boards, universal serial buses and many others. Most PC's have one or two serial ports. Each has a 9-pin connector or sometimes 25-pin on the back of the computer. Computer programs can send data (bytes) to the transmit pin (output) and receive bytes from the receive pin (input). The other pins are for control purposes and ground.



**Figure 3.7: Serial Port (9 Pin)**

The serial port is much more than just a connector. It converts the data from parallel to serial and changes the electrical representation of the data. Inside the computer, data bits flow in parallel, using many wires at the same time. Serial flow is a stream of bits over a single wire, such as on the Transmit or Receive pin of the serial connector. For the serial port to create such a flow, it must convert data from parallel inside the computer to serial on the transmit pin and conversely.



The advantages of using serial data transfer rather than parallel are:

- Serial cables can be longer than parallel cables. The serial port transmits a '1' as -3 to -25 volts and a '0' as +3 to +25 volts where as a parallel port transmits a '0' as 0v and a '1' as 5v. Therefore the serial port can have a maximum swing of 50V compared to the parallel port which has a maximum swing of 5 Volts. Therefore cable loss is not going to be as much of a problem for serial cables as they are for parallel.
- Less wires than parallel transmission. If your device needs to be mounted a far distance away from the computer then 3 core cable (Null Modem Configuration) is going to be a lot cheaper than running 19 or 25 core cable. However you must take into account the cost of the interfacing at each end.
- Microcontrollers have also proven to be quite popular recently. Many of these have in built SCI (Serial Communications Interfaces) which can be used to talk to the outside world. Serial communication reduces the pin count of these MPU's. Only two pins are commonly used, Transmit Data (TXD) and Receive Data (RXD) compared with at least 8 pins using an 8 bit Parallel method. Furthermore, it may also require a Strobe.

### ***3.7.1 Serial Communication Circuitry (Refer to Figure 3.9)***

After considering the odds, the serial port is chosen to transmit data between the PIC16F84 microcontroller and PC. The application of the serial port is feasible as it only utilizes 2 pins, which are transmitting and receive pins. Therefore, designing the PIC16F84 becomes convenient as only 2 pins are used which are transmit and receive pins.

Naturally, the output from the PIC16F84 needs to be transmitted to the PC so that the user could be alerted of any unwanted incidents. Among the transmission medium available include parallel and serial communication. As explained in earlier Chapter 3, the serial port is chosen as it proves to be more feasible.

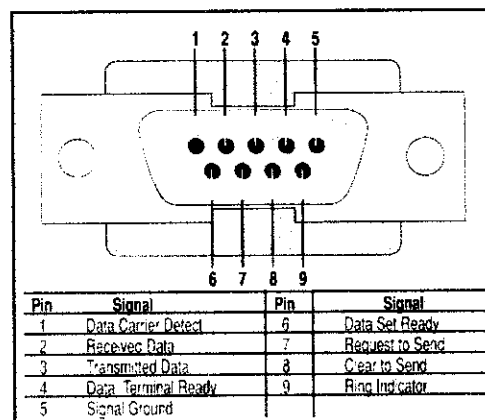
As far as interfacing between the PIC16F84 and PC is concerned, initialization is the first consideration by modifying the settings on both ends to accommodate each other. These settings include:

- Baud rate: 9600
- Parity bit: None
- Data bits: 8
- Stop bits: 1

These initializations will coincide with that of the PIC16F84. The two pins from PIC16F84 that interacts with the serial port are:

- Transmit pin (A0)
- Receive pin (A1)

Old PC's used 25 pin connectors but only about 9 pins were actually used so today most connectors are only 9-pin. Each of the 9 pins usually connects to a wire. Besides the two wires used for transmitting and receiving data, another pin (wire) is signal ground. The voltage on any wire is measured with respect to this ground. Thus the minimum number of wires to use for 2-way transmission of data is 3. Despite this method, it has also been known to work with no signal ground wire but with degraded performance and sometimes with errors.



**Figure 3.8:** RS-232 Pin Out on a DB-9 Pin Used for Asynchronous Data

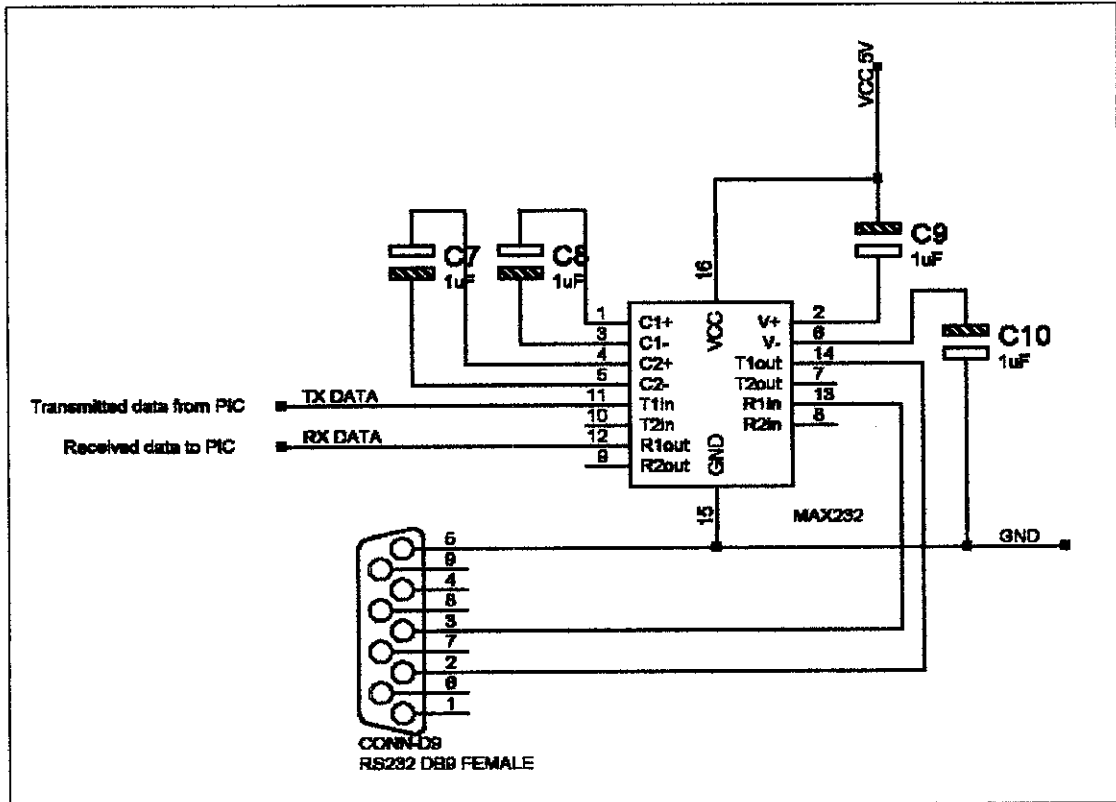


Figure 3.9: Serial Interface Circuit

### 3.8 The ADC0804 Circuit Test

There are many ways and degrees of complexity to test the ADC that are suggested in the datasheet. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code. For ease of testing, the  $V_{REF}/2$  (pin 9) should be supplied with 2.560V and a  $V+$  supply voltage of 5.12V should be used. This provides an LSB value of 20mV. If a full scale adjustment is to be made, an analog input voltage of 5.090V ( $5.120 - 1 \frac{1}{2}$  LSB) should be applied to the  $V_{IN}(+)$  pin with the  $V_{IN}(-)$  pin grounded.

The value of the  $V_{REF}/2$  input voltage should be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of  $V_{REF}/2$  should then be used for all the tests. The digital-output LED display can be decoded by dividing the 8 bits into 2 hex characters, one with the 4 most significant bits (MS) and

one with the 4 least-significant bits (LS). The output is then interpreted as a sum of fractions times the full scale voltage:

$$V_{OUT} = \left( \frac{MS}{16} + \frac{LS}{256} \right) (5.12)$$

For example, for an output LED display of 1011 0110, the MS character is hex B (decimal 11) and the LS character is hex (and decimal) 6, so:

$$V_{OUT} = \left( \frac{11}{16} + \frac{6}{256} \right) (5.12) = 3.64V$$

But, using the DC power supply provided by the lab technician, the voltage supplied only have 1 decimal place. Therefore, the output signal might have some error due to the voltages supplied and also the built-in decoder. It is concern for this project to decrease level of signal error as to make sure the system is reliable. The outputs of the ADC are connected to the inputs of the PIC. The PIC is programmed to read all the inputs and send the data based on the conditions that are set in the PIC program. For example, the outputs of the ADC give 0000 0000 when the input is 0V. In the PIC program, the condition has been set; if all inputs are 0, then send "0" to the PC and it goes for other values.

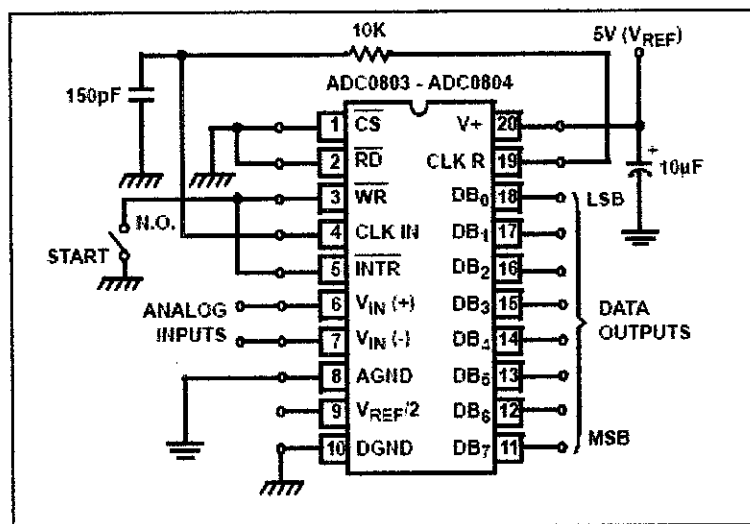


Figure 3.10: Free-running connection (See Table 4.3 for output results)

Calculations of the output in bits are:-

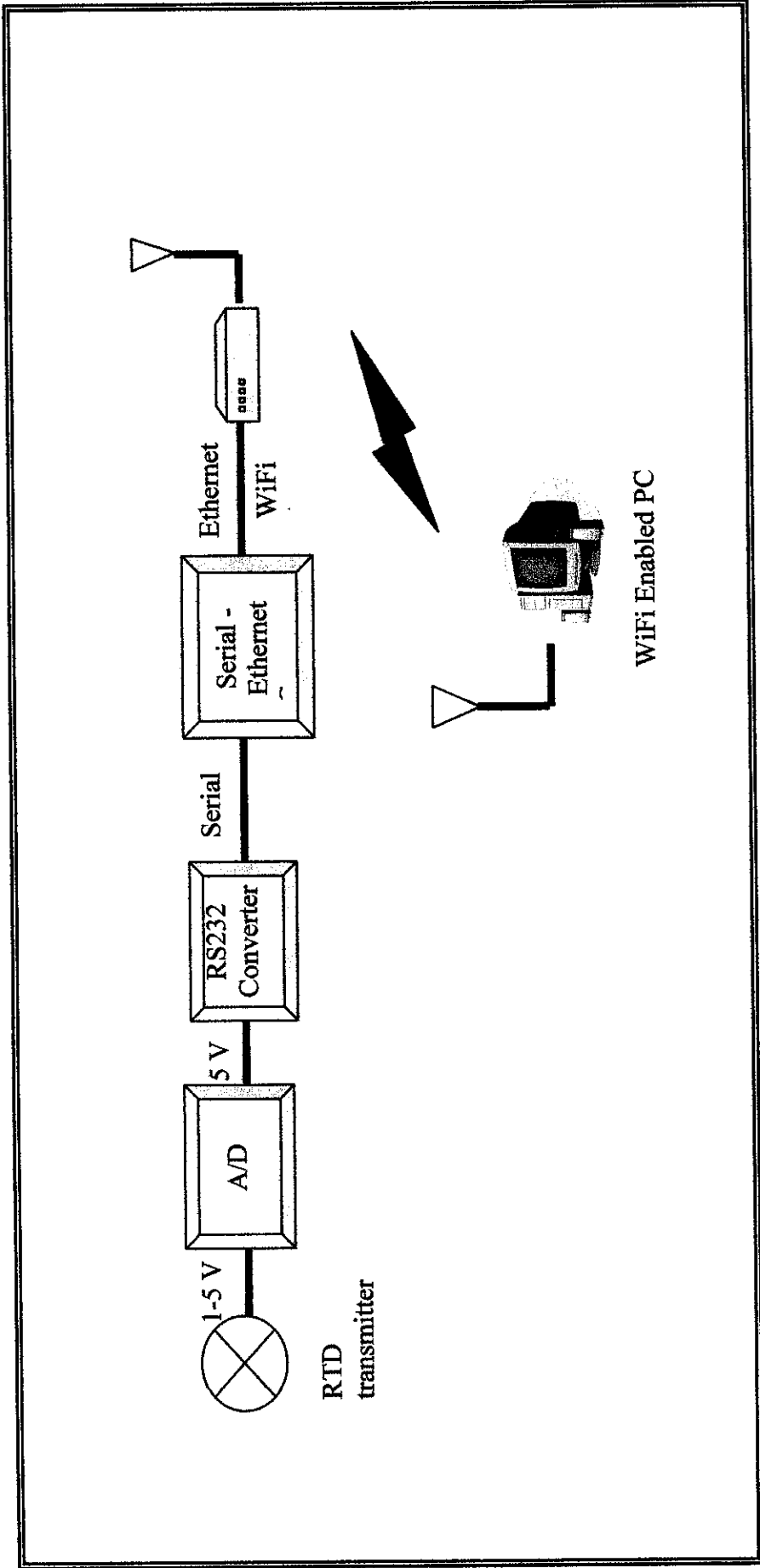
Example: The analogue input is 5V

$$8\text{-bit} = 2^8 = 256$$

We use 0-5V. So,  $5/255 = 0.0196\text{V}$  which means that 1-bit is 0.0196V. If 5V,  
 $5/0.0196 = 255_{10} = 1111\ 1111_2$

### **3.9 The System Design** (*Refer to Figure 3.11*)

Based on the research and survey conducted, the system needs a RTD transmitter, A/D converter, a Serial (RS-232) to Ethernet converter, a WiFi Access Point and WiFi enabled PC



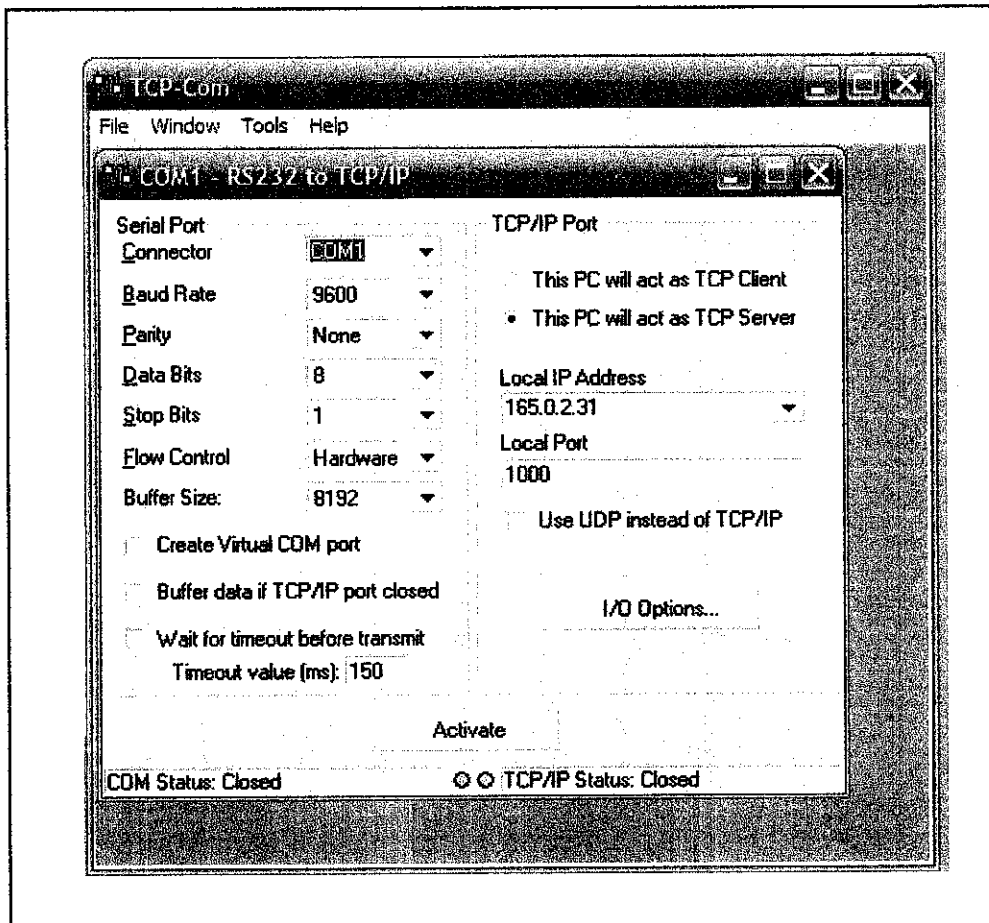
**Figure 3.11:** Overall Block Diagram of the Project System

### **3.10 TCP-Com Software**

TCP-Com is a software based RS232 to TCP/IP converter. TCP-Com allows any of the serial ports on your PC to send and receive data directly over any TCP/IP network. The functions of TCP-Com is to turn a PC into a multi-port “Serial Device Server” so that we can connect any RS232 serial device directly to a TCP/IP network and communicate with that device from any other workstation in the same network or across the Internet. TCP-Com can also create “Virtual” RS232 serial ports that are actually connections to a TCP/IP port. This allows us to use any existing serial communications software to send and receive data across a TCP/IP network. It can be attach to any RS232 device including, bar code scanners, sensors, gages, meters, telephone systems, laboratory, instruments, industrial controllers, modems, printers, etc. – to the serial ports of a PC and communicate with that device from any other computer in the same network (or across the Internet). This would allow us to use HyperTerminal or any other RS232 or TCP/IP communications software to read or write to the serial device from any PC connected to the same network. Then, we can convert the data into readable data such as graph or reading based on numbers using Microsoft Visual Basic 6.

#### ***3.10.1 TCP-Com Settings***

In this project, we need two PCs that are one act as transmitter and another as receiver. On the transmitter side, the PC retrieves data from the sensor and converts it into TCP/IP protocol. Then, the data is sent through the Access Point. The settings are as follows:-



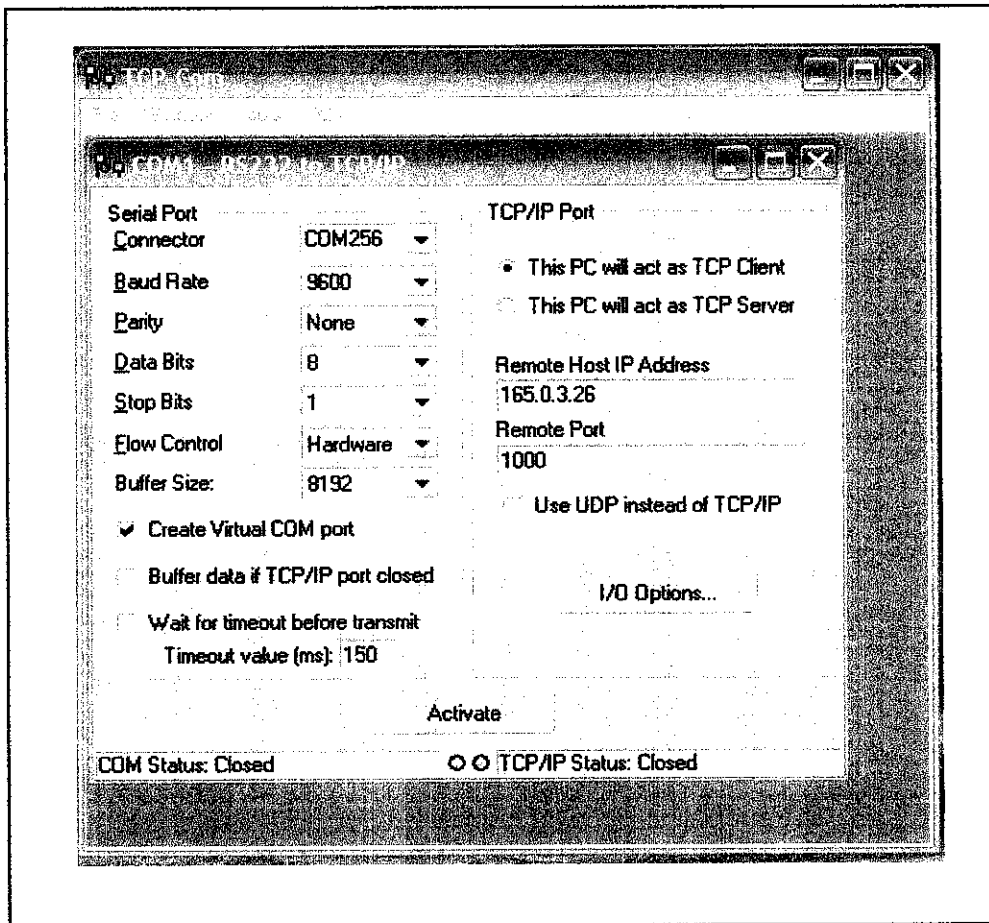
**Figure 3.12: Setting for the transmitter side**

From Figure 4.6, we can see that there are several settings have to be made.

- 1) Set the connector to COM1 because it will retrieve data from the sensor that is connected to the PC using COM1 port.
- 2) Set the baud rate, parity, data bits, stop bits and flow control according to the PIC program that we have set earlier.
- 3) Set the local IP according to the PC's IP address.
- 4) After this program is activated, it will open the COM1 port and also TCP/IP port. Now, the PC is set as server and will listen to any PCs that are connected to its network.



On the other hand, the settings for the receiver side are as follows:-



**Figure 3.13:** Setting for the receiver side

From Figure 4.7, there are several settings has to be made.

- 1) Set the connector to any COM numbers except COM1 because COM1 is used by the transmitter PC.
- 2) Set the baud rate, parity, data bits, stop bits and flow control according to the PIC program that we have set earlier.
- 3) Set the local IP according to the transmitter PC's IP address.
- 4) After this program is activated, it will open the COM port and also TCP/IP port. After connected to the transmitter PC, the data will be retrieving to this PC.

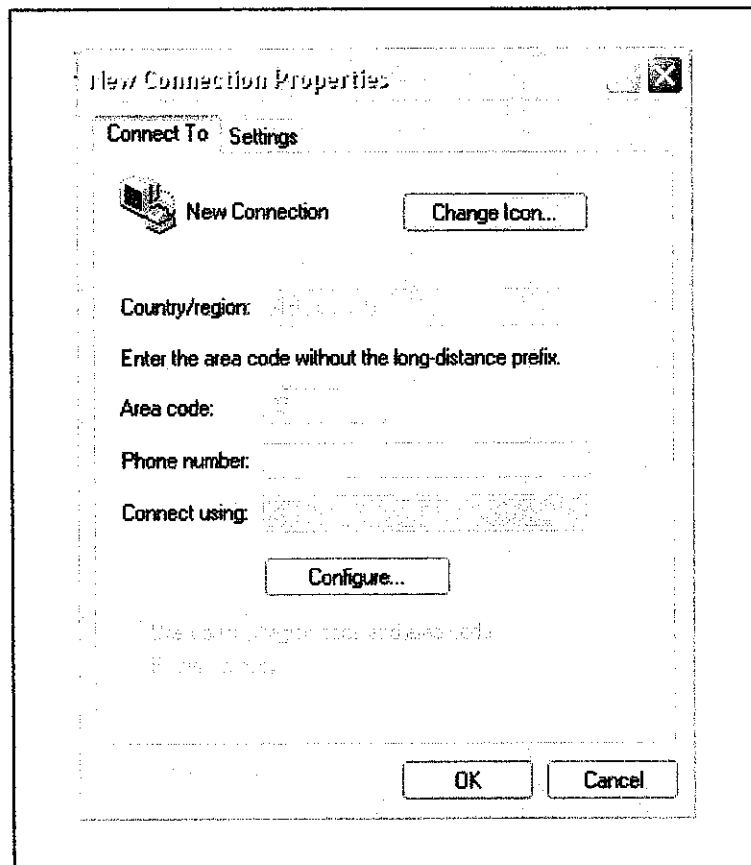
## CHAPTER 4

### RESULTS AND DISCUSSIONS

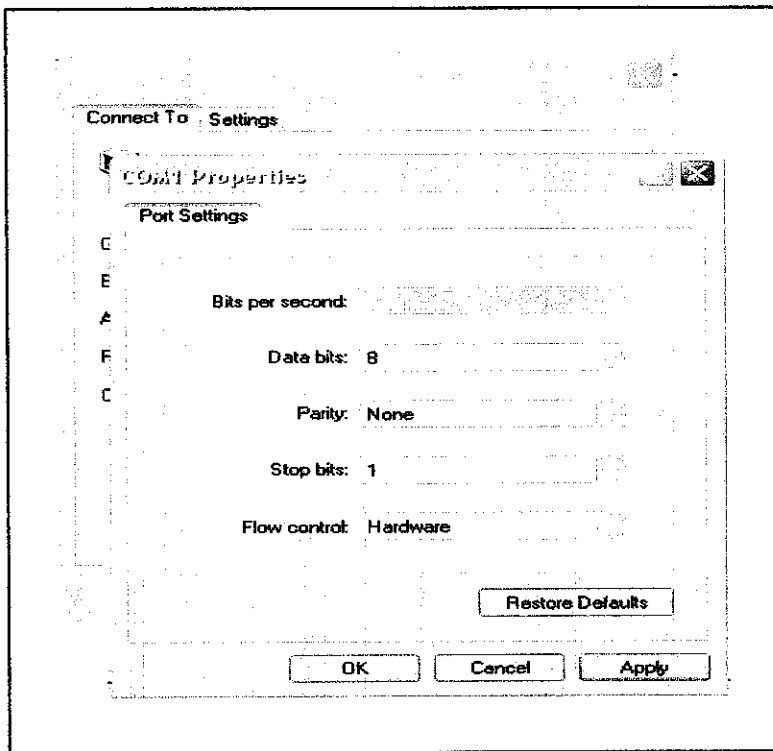
#### 4.1 HyperTerminal Settings

The circuit can be tested using Hyper Terminal. HyperTerminal is a program that you can use to connect to other computers, Telnet sites, and bulletin board systems (BBSs), online services, and host computers, using either your modem or a null modem cable. In this case, the PIC is identified as a modem with speed of 9600bps. The speed is already identified earlier in the program for the PIC as we can see in Appendix B. The configurations for the Hyper Terminal are as follows.

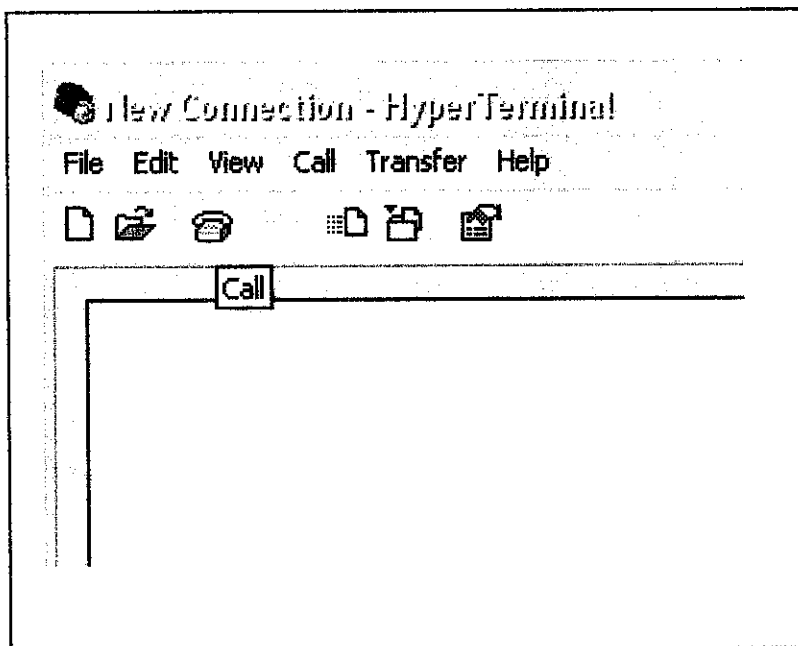
- 1) In the properties dialog, COM1 is selected



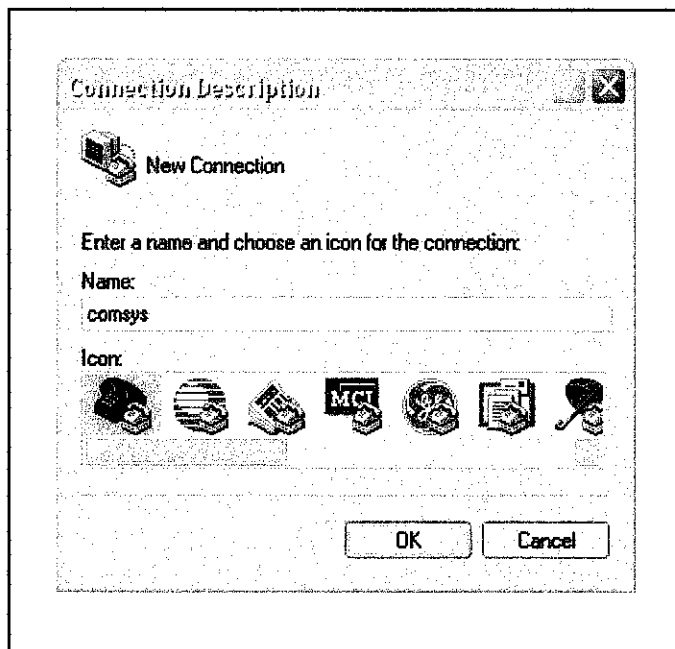
- 2) In the configure dialog, bits per second is set to 9600, then hit OK



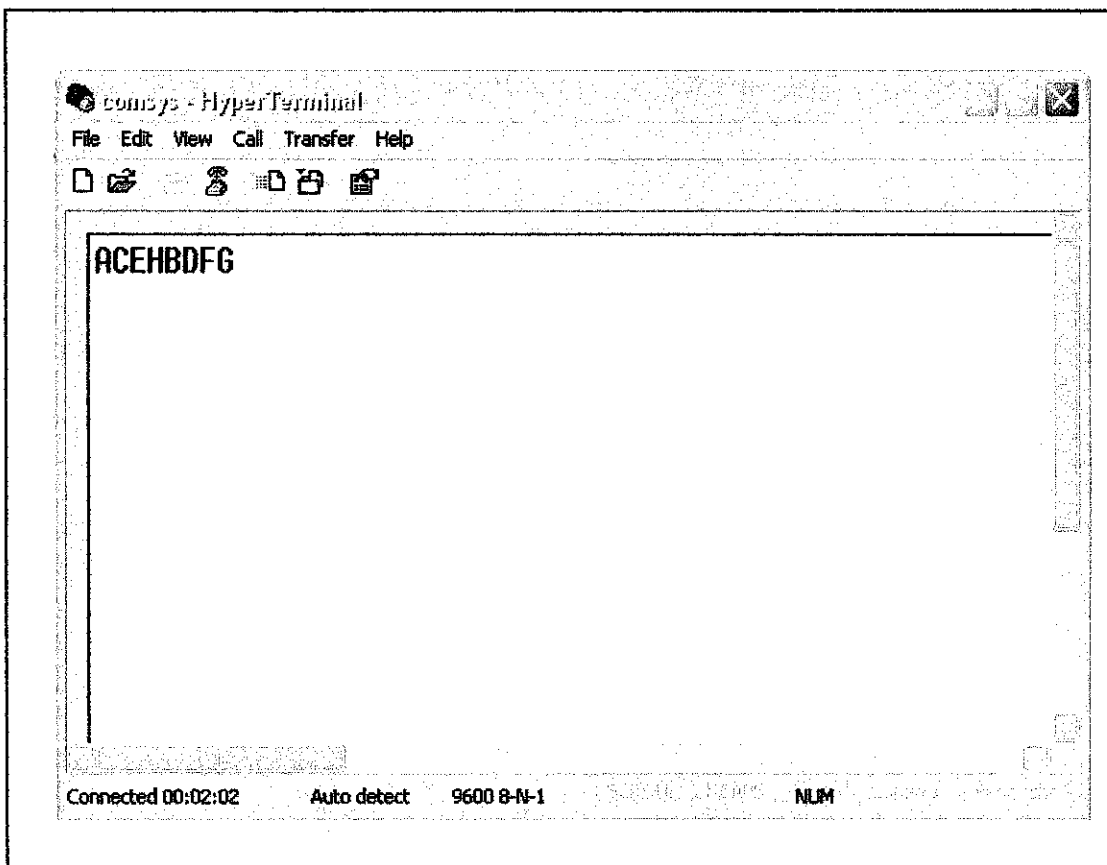
- 3) The call button is then selected



- 4) Give a name for the connection and hit OK



- 5) The transmitter is tested by giving analogue input and the output is observed as follows.



If all the circuits are working and the output is the same like the picture shown previously, the ADC0804, PIC and MAX 232 are working correctly. The HyperTerminal can be used for both PCs to check whether the data that are retrieved by the PC are the right one.

## **4.2 The ADC Outputs**

Based on the Table 4.3, we can see that expected values and the real values are almost the same. It is known that the ADC circuit is reliable and satisfactory.

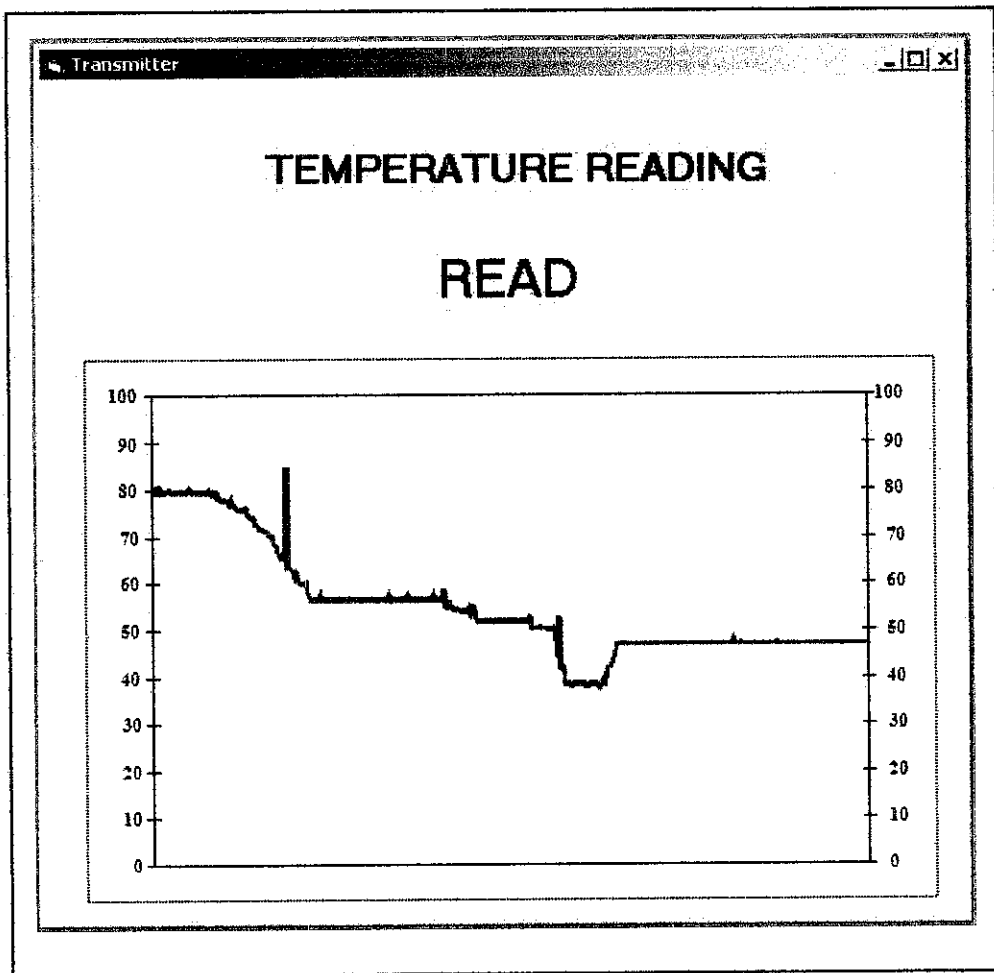
After completing the ADC, PIC and RS232 circuit, we have to connect them together in order to be able to communicate between the input and PC. Refer to Appendix C for overall circuit diagram.

Input Voltages	Expected ADC Outputs		Real ADC Outputs	
	Decimal	Binary	Binary	Multimeter Reading (V)
0.00	0.00	0000 0000	0000 0000	0.0000
0.25	12.75	0000 1100	0000 1100	0.2508
0.50	25.50	0001 1001	0001 1011	0.4980
0.75	38.25	0010 0110	0010 1111	0.7489
1.00	51.00	0011 0011	0011 0111	1.0059
1.25	63.75	0011 1111	0011 1111	1.2513
1.50	76.50	0100 1100	0100 1101	1.5103
1.75	89.25	0101 1001	0101 1011	1.7515
2.00	102.00	0110 0110	0110 1001	2.0043
2.25	114.75	0111 0010	0111 0011	2.2476
2.50	127.50	0111 1111	0111 1111	2.5003
2.75	140.25	1000 1100	1000 1101	2.7568
3.00	153.00	1001 1001	1001 1001	3.0059
3.25	165.75	1010 0101	1010 0111	3.2459
3.50	178.50	1011 0010	1011 1010	3.5041
3.75	191.25	1011 1111	1011 1111	3.7504
4.00	204.00	1100 1100	1100 1111	4.0118
4.25	216.75	1101 1000	1101 1011	4.2462
4.50	229.50	1110 0101	1110 1111	4.5133
4.75	242.25	1111 0010	1111 0111	4.7544
5.00	255.00	1111 1111	1111 1111	5.0015

Table 4.1: Outputs of the ADC0804

### 4.3 The Visual Basic Interface Program

Basically, there are eight input triggered by the PIC circuit. These are the character A, B, C, D, E, F, G and H. Each character is recognized as a case, in each case commands can be put to be executed. In the VB program, we can use these characters as the digital voltage. For example, if we send 5V analogue signal and the ADC will digitized it as 1111 1111. This will activate all the input pins of the PIC. So, all characters will be sent to the receiver PC. In VB 6, it will get all the characters and we can program it to interpret it as 5V when all characters are retrieved. Same goes to the others.



**Figure 4.1:** The VB interface program

Using the Visual Basic program, we can read and identify the data easily. Base on the VB program in Figure 4.1, the program will show the graph and the actual reading. From the graph, we can identify pattern of the process temperature.

#### 4.4 Testing Range of The WiFi Transmission

After completing the system and testing it at the Instrumentation Lab, the range must be tested in order to identify the maximum range that the WiFi can transmit the data without any problems. Table 4.2 summarize the testing made at Building 22 and 23.

Range (m)	Connection Speed (Mbps)
10	54
20	54
30	24
40	24
50	24
60	16
70	16
80	4
90	2
100	No connection

**Table 4.2:** Range and Connection Testing

Based on Table 4.2, it is known that the WiFi Access Point (AP) can be connected up to 90m range. This is normal because the AP that is used in this project is a home AP which can range up to 100m. This system cannot achieve 100m because the interference occurred at Building 22 and 23.



#### 4.5 Problems Encountered

Some problems encountered during the project this semester:-

a) *Vendors late response*

Late response from the vendors to enquire about the WiFi converter or Access Point.

b) *Sourcing hardwares*

Some of the ICs and components are not available in the component store. So, the components are sourced from Ipoh and Kuala Lumpur.

c) *Cost limitation*

Most of the WiFi Access Point sold in Malaysia has only the Ethernet ports which mean that the second option is to be chosen and the price is quite cheap. On the other hand, the WiFi Access Point which has the serial port (DB-9) is very expensive. Each cost around RM600-800.

d) *Availability of the WiFi Access Point.*

Looking at the first options, the WiFi Access Point that have the serial port (RS-232) are only available overseas while the WiFi Access Point that has the Ethernet port (RJ-45) is available in Malaysia.

e) *Buy in bulk*

Some vendor only sells the device above 10 pieces. The price for a piece is much higher than the bulk price.

f) *Technical support and guarantee*

Technical support and guarantee from the vendor is a must to make sure that the device is compatible with the system designed and to get familiar with the device especially when system configuration is needed to be done.

*g) Burning PIC with Warp 13*

The process of burning the program into the PIC may take many attempts as the programmer is prone to errors.

*h) Circuit faulty*

The circuitries also have to go through a series of troubleshooting. Upon checking every connections and components, some components were found to be faulty.

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATIONS**

#### **5.1 Conclusion**

Using hardwired sensors consumes more space, high cost for maintenance especially for inexpensive monitoring and predictive maintenance. The wireless communication is the latest technology that they can implement in their field or plant to cut cost of maintenance, save their time and the system is very simple.

Bluetooth are widely used to replace cables. Radio frequency (RF) can be easily interfered with other radio frequencies. ZigBee is still in research and development phase. Infrared (IR) is a one-to-one technology which means that the transmitter and receiver must be located at the line of sight of each other. It also can be easily interfered by other frequencies.

On the other hand, WiFi are widely used for wireless high speed LAN and internet access. It can provide wireless connection at the speed of up to 11 Mbps. It is widely used especially in networking and communications.

#### **5.2 Recommendations**

In-depth study and research in wireless communication are needed in order to successfully complete this project. The WiFi technology is suitable for this project because of its security, maximum distance and speed that it can achieved and also maximum device it can connect at one time. For the first semester, this project will

under go research and designing the system. The next semester, this project will be in the installation and troubleshooting phase. The system design must be reliable, less complexity and cost.

The system can be improved in terms of simplicity and market values. The system can be developed further so that it can be connected to the DCS network. It is hoped that the improved system successfully implemented in the real plant.

This system can be improved by replacing the PC that is used as the serial-Ethernet converter with the available serial-Ethernet hardware converter.

A more advance and improve system can be design to use the digital signals from the RTD transmitter instead using the analogue signals and replace the RS232 with RS485 which is more compatible and simple.

## REFERENCES

- [1]. F. L. Lewis , “Wireless Sensor Networks”, Smart Environments: Technologies, Protocols, and Applications., D.J. Cook and S.K. Das, John Wiley, New York, 2004.
- [2]. John Lewis, “New frontier for wireless: Sensor networks”, Network World, 06/07/04, [www.networkworld.com/wireless](http://www.networkworld.com/wireless)
- [3]. Julia Layton, Curt Franklin, “How Bluetooth Works”, Electronic Stuff, [www.howstuffworks.com/bluetooth](http://www.howstuffworks.com/bluetooth)
- [4]. Marshall Brain, “How WiFi Works”, Electronic Stuff, [www.howstuffworks.com/wifi](http://www.howstuffworks.com/wifi)
- [5]. F. Bennett, D. Clarke, J. B. Evans, A. Hopper, A. Jones, and D. Leask. Piconet: Embedded mobile networking. IEEE Personal Communications Magazine, 4(5):8–15, Oct. 1997.
- [6]. Wikipedia, “ZigBee”, <http://en.wikipedia.org/wiki/ZigBee>
- [7]. Wikipedia, “Microwave”, <http://en.wikipedia.org/wiki/Microwave>
- [8]. PIC16F84 Microcontroller Datasheet.

## OTHER REFERENCES

- 1) Wikipedia, "Analogue-to-Digital Converter",  
[http://en.wikipedia.org/wiki/Analogue-to-Digital\\_Converter](http://en.wikipedia.org/wiki/Analogue-to-Digital_Converter)
- 2) Lammert Bies, "Yost Serial Device Wiring",  
<http://www.lammertbies.nl/comm/cable/yost-serial-rj45.html>
- 3) Kevin Ross, "Rs-232 to TTL Cable", <http://www.seattlerobotics.org/>
- 4) TALTech, "TCP-COM RS-232 To TCP/IP Converter",  
<http://www.taltech.com/products/tcpcom.html>
- 5) "MSComm Control and Visual Basic"  
<http://www.ontrak.net>
- 6) "How to Check RS-232 Port to Verify Operation"  
[http://www.bb-europe.com/tech\\_articles](http://www.bb-europe.com/tech_articles)
- 7) "Understanding and Using Visual Basic", *Jared Hoylman*  
<http://www.renton.com>
- 8) "Using Visual Basic 6", *Brian Siler; Jeff Spotts*, MacMillan Computer Publishing.

## APPENDICES

## APPENDIX A

### C codes for PIC Microcontroller

```
#include <16f84a.h>
#fuses XT,NOPROTECT,NOWDT

#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_A0, rcv=PIN_A1)

#define ALL_OUT 0
#define ALL_IN 0xFF

void respond() {
    output_low(PIN_A2);
    output_low(PIN_A3);
    output_low(PIN_A4);
}

main () {
    int i;
    set_tris_B(0xFF);
    do {
        i=0;
        if (input(PIN_B0)!=0) {
            output_high(PIN_A2);
            output_high(PIN_A4);
            delay_ms(1);
            putchar(65);
            delay_ms(1);
            output_high(PIN_A3);
        }
        if (input(PIN_B1)!=0) {
            output_high(PIN_A2);
            output_high(PIN_A4);
            delay_ms(1);
            putchar(66);
            delay_ms(1);
            output_high(PIN_A3);
        }
        if (input(PIN_B2)!=0) {
            output_high(PIN_A2);
            output_high(PIN_A4);
            delay_ms(1);
        }
    } while(1);
}
```



```

    putchar(67);
    delay_ms(1);
    output_high(PIN_A3);
}
if (input(PIN_B3)!=0) {
    output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putchar(68);
    delay_ms(1);
    output_high(PIN_A3);
}
if (input(PIN_B4)!=0) {
    output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putchar(69);
    delay_ms(1);
    output_high(PIN_A3);
}
if (input(PIN_B5)!=0) {
    output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putchar(70);
    delay_ms(1);
    output_high(PIN_A3);
}
if (input(PIN_B6)!=0) {
output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putchar(71);
    delay_ms(1);
    output_high(PIN_A3);
}
if (input(PIN_B7)!=0) {
    output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putchar(72);
    delay_ms(1);
    output_high(PIN_A3);
}

```

```
} while (TRUE); }
```

## APPENDIX B

### Visual Basic 6 Programming

```
Public num As Integer
Public prev_num As Integer
Public total As Integer
Public inc As Integer
Public stopnow As Integer
Public realtotal As Integer
Public temp_data1 As Integer
```

```
Private Sub Form_Load()
    comm.InputMode = 0 'take ascii as input
    comm.CommPort = 1
    comm.Settings = "9600,N,8,1"
    comm.PortOpen = True 'open port
    comm.InputLen = 1 'limitation for input
    comm.RThreshold = 1
    total = 0
    num = 0
```

End Sub

```
Private Sub Comm_OnComm()
    Select Case comm.Input
        Case "A"
            inc = 1
            prev_num = num
            num = 1
            Call process
            If stopnow > 5 Then
                Data1.Caption = realtotal
            End If
```

```
Case "B"  
  inc = 2  
  prev_num = num  
  num = 2  
  Call process  
  If stopnow > 5 Then  
    Data1.Caption = realtotal  
  End If
```

```
Case "C"  
  inc = 4  
  prev_num = num  
  num = 3  
  Call process  
  If stopnow > 5 Then  
    Data1.Caption = realtotal  
  End If
```

```
Case "D"  
  inc = 8  
  prev_num = num  
  num = 4  
  Call process  
  If stopnow > 5 Then  
    Data1.Caption = realtotal  
  End If
```

```
Case "E"  
  inc = 16  
  prev_num = num  
  num = 5  
  Call process  
  If stopnow > 5 Then  
    Data1.Caption = realtotal  
  End If
```

```
Case "F"  
  inc = 32  
  prev_num = num  
  num = 6
```

```
Call process
If stopnow > 5 Then
    Data1.Caption = realtotal
End If
```

```
Case "G"
    inc = 64
    prev_num = num
    num = 7
    Call process
    If stopnow > 5 Then
        Data1.Caption = realtotal
    End If
```

```
Case "H"
    inc = 128
    prev_num = num
    num = 8
    Call process
    If stopnow > 5 Then
        Data1.Caption = realtotal
    End If
```

```
End Select
```

```
End Sub
```

```
Public Function draw_graph()
```

```
    'chart.Row = 1
```

```
    'chart.Data = realtotal
```

```
    'For i = 2 To 38
```

```
        'for temp data
```

```
        'chart.Row = i
```

```
        'chart.Data = 10
```

```
        'i = i + 1
```

```
    'Next
```

```
chart.Row = 1
```

```
temp_data1 = chart.Data
```

```
chart.Data = realtotal

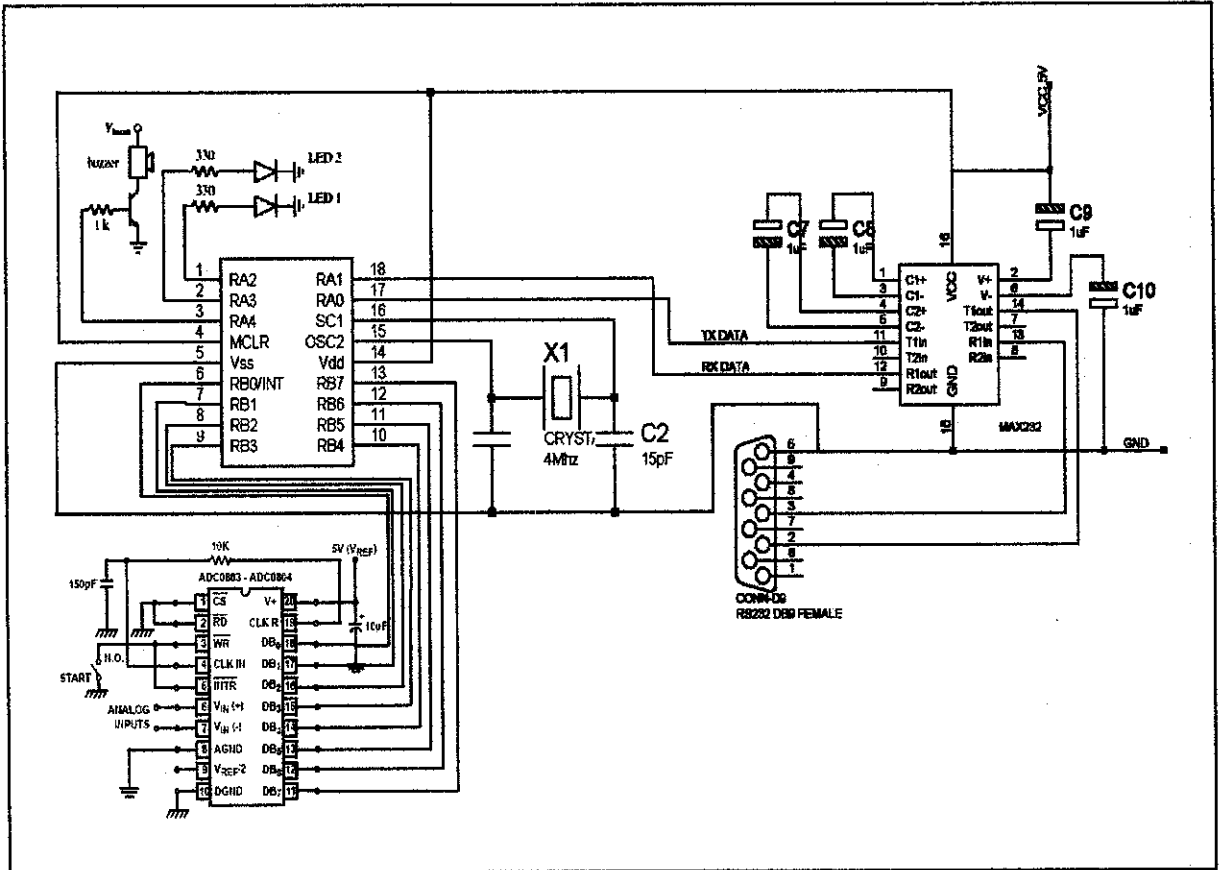
For i = 2 To 40
chart.Row = i
temp_data1 = chart.Data

Next

End Function
Public Function process()
If num <= prev_num Then
    realtotal = total
    total = inc
    stopnow = 10
    Call draw_graph
End If
If num > prev_num Then
    total = total + inc
    stopnow = 0
End If
End Function
```

# APPENDIX C

## Overall Circuit Diagram



## APPENDIX D

### ASCII Codes Table

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT (end of transmission)	36	24	044	&#36;	&	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1 (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2 (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3 (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4 (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
29	1D	035	GS (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: [www.ascitable.com](http://www.ascitable.com)



## APPENDIX E

### PIC16F84A Datasheet



# PIC16F84A

## 18-pin Enhanced FLASH/EEPROM 8-Bit Microcontroller

### High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt on-change
  - Data EEPROM write complete

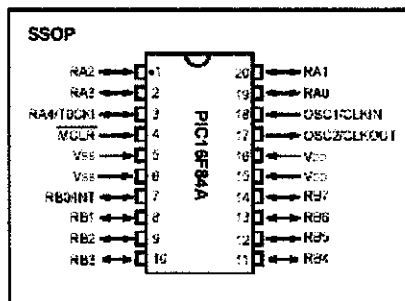
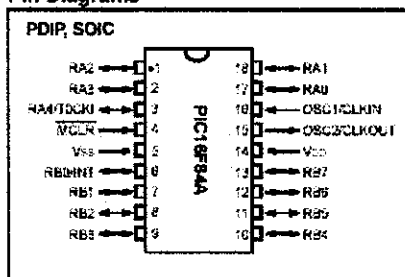
### Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

### Special Microcontroller Features:

- 10,000 erase/write cycles Enhanced FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

### Pin Diagrams



### CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 kHz
  - < 0.5 µA typical standby current @ 2V

# APPENDIX F

## ADC0804 Datasheet

### ADC0803, ADC0804

Data Sheet

August 2002

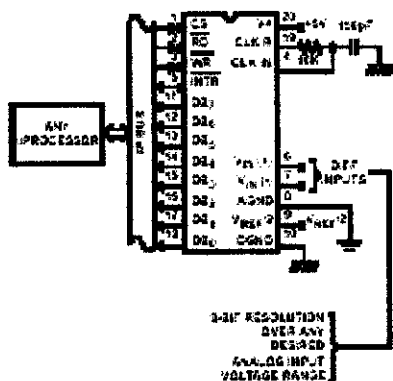
PN3064.4

#### 8-Bit, Microprocessor-Compatible, A/D Converters

The ADC0803A family are CMOS 8-Bit successively-approximated A/D converters which use a modified potentiometric ladder and are designed to operate with the 8080A control bus via three-state outputs. These converters appear to the processor as memory locations on 8-bit busses, and hence no interfacing logic is required.

The differential analog voltage input has good common-mode rejection and permits offsetting the analog conversion to voltage values. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage down to the full 8 bits of resolution.

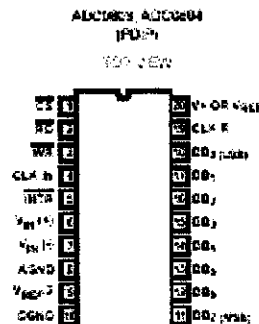
#### Typical Application Schematic



#### Features

- 80C43 and 80C80/85 Bus-Compatible - No Interfacing Logic Required
- Conversion Time  $t_{CONV} = 100 \mu s$
- Easy Interface to Most Microprocessors
- Will Operate in a Stand-Alone Mode
- Differential Analog Voltage Inputs
- Works with Bandgap Voltage References
- TTL-Compatible Inputs and Outputs
- On-Chip Clock Generator
- Analog Voltage Input Range (Single = 5V Supply)
- No Zero-Adjust Required
- 80C43 and 80C80/85 Bus-Compatible - No Interfacing Logic Required

#### Pinout



#### Ordering Information

PART NUMBER	ERROR	EXTERNAL CONDITIONS	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
ADC0803-01T	±1/2 LSB	VREFN = Adjusted for Correct Full-Scale Reading	0 to 70	DIP (16 PIN)	E203
ADC0804-01T	±1/2 LSB	VREFN = 2.56V (typical Adjustment)	0 to 70	DIP (16 PIN)	E203

©2002 Intersil Corporation. All rights reserved. Intersil Corporation is a wholly owned subsidiary of Intel Corporation. (Intersil is a registered trademark of Intel Corporation.) Copyright is taken in all other marks. All rights reserved.

# APPENDIX G

## MAX232 Datasheet

MAX232



### ±15kV ESD-Protected, +5V RS-232 Transceivers

#### General Description

The MAX232 is a CMOS IC that provides a complete RS-232 interface solution. It contains two full drivers and two full receivers, all of which are ESD-protected to ±15kV. The MAX232 is designed to be used with a single +5V power supply. It is available in a 16-pin DIP package.

The MAX232 is designed to be used with a single +5V power supply. It is available in a 16-pin DIP package. The MAX232 is designed to be used with a single +5V power supply. It is available in a 16-pin DIP package.

#### Applications

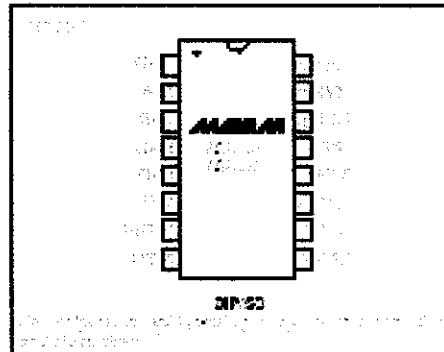
The MAX232 is used in a wide variety of applications, including data communications, instrumentation, and control systems.

Ordering information appears at end of datasheet.

#### Features

- ♦ ESD Protection for RS-232 I/O Pins:
  - ±15kV—Human Body Model
  - ±8kV—IEC1000-4-2 Contact Discharge
  - ±15kV—IEC1000-4-2 Air-Gap Discharge
- ♦ Latchup-Free (meets bipolar equivalents)
- ♦ Guaranteed 122kba Data Rate—Caplink™ Compatible
- ♦ Guaranteed 30µs Min. Setup Rate
- ♦ Operates from a Single +5V Power Supply

#### Pin Configurations



#### Selection Guide

PART	No. of RS-232 DRIVERS	No. of RS-232 RECEIVERS	RECEIVERS ACTIVE IN SHUTDOWN	No. of EXTERNAL CAPACITORS	LOW-POWER SHUTDOWN	TTL THREE-STATE
MAX232E	2	2	0	4	Yes	Yes
MAX232C	2	2	0	4	Yes	No
MAX232D	2	2	0	4	Yes	Yes
MAX232E	2	2	0	4	Yes	Yes
MAX232C	2	2	0	4	Yes	No
MAX232D	2	2	0	4	Yes	Yes
MAX232E	2	2	0	4	Yes	Yes
MAX232C	2	2	0	4	Yes	No
MAX232E	2	2	0	4	Yes	Yes
MAX232C	2	2	0	4	Yes	No
MAX232E	2	2	0	4	Yes	Yes

For more information, please contact your local sales office.



Maxim Integrated Products

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-0800

MAX202E-MAX213E, MAX232EMAX241E

# APPENDIX H

## RTD Transmitter Datasheet

Yokogawa Co., Ltd.

### General Specifications

Model YTA Series  
Temperature Transmitter  
With HART® Communication

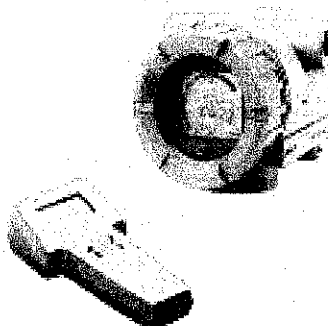
GS 01CS0T01-03E

The device temperature measured at the probe device and the ambient temperature are displayed on the digital display. Ambient air display is available in the case of the transmitter.

4-20 mA output can also be selected with DIP switch.

### FEATURES

- **Process Ranging And Monitoring Functions**  
Range type and range can be remotely set and monitored through HART® Communication.
- **On-line Communication**  
Output signal and communication signal do not interfere making on-line communication possible.
- **Self-Diagnostic**  
Loss of signal, defective ambient temperature, incorrect input setting, etc. can be diagnosed through HART® Communication.



### STANDARD SPECIFICATIONS

For items other than those described below, refer to each General Specifications sheet.

#### Applicable Model:

Model YTA020, YTA010 and YTA010 Temperature Transmitters

#### Conditions of Communication Line:

- Supply Voltage: 15.4 to 42 V DC
- Load Resistance: See Figure 1
- Min. pulse width: 24 A (200 μs) (1.51 mm diameter)
- Cable type: Single pair shielded or multiple pair twisted pair
- Maximum twisted-pair length: 10000 ft (3048 m)
- Maximum multiple pair twisted-pair length: 5000 ft (1524 m)

Use the following formula to determine cable length for a specific application:

$$L = \frac{50 \times 10^3}{100 - R} - 10000$$

Where:

- L = Length in feet or meters.
- R = Resistance in ohms (cable length resistance plus barrier resistance).
- C = Cable capacitance in pF or pF/m.
- C = Maximum allowed capacitance of twisted-pair cable.

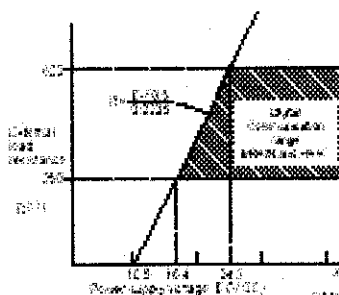


Figure 1. Relationship Between Power Supply Voltage and Current/Load Resistance

**YOKOGAWA** ◆

Yokogawa Electric Corporation  
1-2-1, Chikara-cho, Tsurumi-ku, Yokohama, Japan

Yokogawa Electric Company, Inc.  
300 Central Expressway, Redwood City, CA, USA