MOBILE PHONE APPLICATION DEVELOPMENT FOR INTERROGATING EXTERNAL SENSOR

by

Ag Shahrin Afwan B Ag Tahir
11673

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronic Engineering)

MAY 2011

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

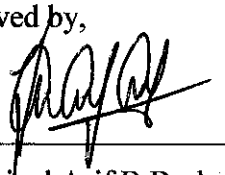
MOBILE PHONE APPLICATION DEVELOPMENT FOR INTERROGATING EXTERNAL SENSOR

by

Ag Shahrin Afwan B Ag Tahir

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 by,

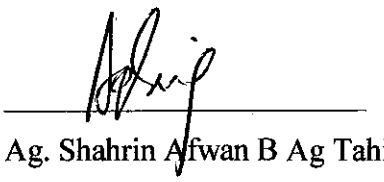


(Dr. Zainal Arif B Burhanudin)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
MAY 2011

CERTIFICATION OF ORIGINALITY

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



Ag. Shahrin Afwan B Ag Tahir

ABSTRACT

Recently, Smartphone processing power shows a remarkable performance on doing multitask function set by the user. As the manufacturing industry has come to a further development on data inspection by such sensor for example; Temperature Sensor, Pressure Sensor, Level Sensor, etc, both task of acquiring and analyzing data will need to be carried out at different time. By utilizing mobile phone as the main target data acquisition, data are easily access at the end of the user finger tips. Literature review comprises all the keywords and terms that we need to know before we can proceed to the system development. The methodology focus on more on project flow, hardware and software used such as, Programmable Integrated Circuit (PIC) Microcontroller, Bluetooth Module, Lithium Polymer Battery, O2 XDA Mobile Phone, vxHpc Software, Windows Visual Studio 2008, MPLAB 8.10 and etc. As for the result, the external system already able to communicate and interconnect with mobile phone with the ability of vxHpc software to act as hyper terminal for mobile phone. The external device are attached with internal temperature sensor (LM35DZ) and able to be connected with 3 external sensors as the users are able to interrogate the whole system by giving command and getting specific sensor data from external sensor device. As for conclusion, the system can be already be used for specific application as the microcontroller are already comprises eighty percent of its entire program memory utilization.

ACKNOWLEDGEMENT

Alhamdulillah, I praised utmost to ALLAH S.W.T for giving me the chance to complete the final year project for my bachelor degree in Electrical and Electronic Engineering, Universiti Teknologi PETRONAS. I would like to express my gratitude to all those who have contributed and gave the possibility to complete the project. Highest appreciation goes to my respectable supervisor, Dr Zainal Arif B Burhanudin, for his relentless patience and guidance throughout the project. Furthermore, thank you to AP Dr Mohamad Naufal B Mohamad Saad for his constructive comments and suggestion.

I also would like to thank you UTP Robocon Team and UTP Eurobot Team under the supervision of Electrical and Electronic Engineering Department, and lead by AP Dr Mohamad Naufal B Mohamad Saad for the opportunity to apply my technical and engineering knowledge in prestigious robotic competition, such as Robocon 2009 and Eurobot 2011 Competition. Not forgetting, my fellow colleges and friends for their assistants, ideas, moral support and motivation throughout the project.

Finally, I would like to dedicate my special thanks to my parents, Hj Ag Tahir B. Said and Hjh Rizah Bt Hj Shuhailie and my whole family for continuous support and encouragement.

AG SHAHRIN AFWAN B AG TAHIR

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLES

LIST OF ABREVIATION

PDA	Personal Digital Assistant
EDA	Enterprise Digital Assistant
DSP	Digital Signal Processing
IR	Infra Red
RF	Radio Frequency
WiFi	Wireless Fidelity
uC, µC, MCU	Micro-Controller
NOR	Non-volatile Memory
ROM	Read Only Memory
RAM	Random Access Memory
I/O	Input Output
UART	Universal Asynchronous Receive Transmit
SDK	Software Development Kits
PANs	Personal Area Networks
GHz	Giga Hertz
EM	Electromagnetic
LAN	Local Area Network
WPA	Wifi Protection Access encryption

CHAPTER 1
INTRODUCTION

1.1 Background of study

Recently application for mobile phone has been developed to make use of the mobile phone processing power and its user friendliness [1]. Many applications have been developed by application developers for example [2]:

- vxHpc Software (Terminal Emulation for Windows Mobile software) [3],
- Global Positioning System - GPS (Program to allocate user location) [4],
- SportyPal (Used to track healthy sport activity carried out by user) [2] , and
- Cell Mapper (program used to map the extent of your network providers) [2], etc.

Mobile application development is the process by which applications are developed for hand held devices such as personal digital assistants (PDA), enterprise digital assistants (EDA) or mobile phones [5]. These applications are either pre-installed on phones during manufacture, or downloaded by customers from application stores and other mobile software distribution platforms [6].

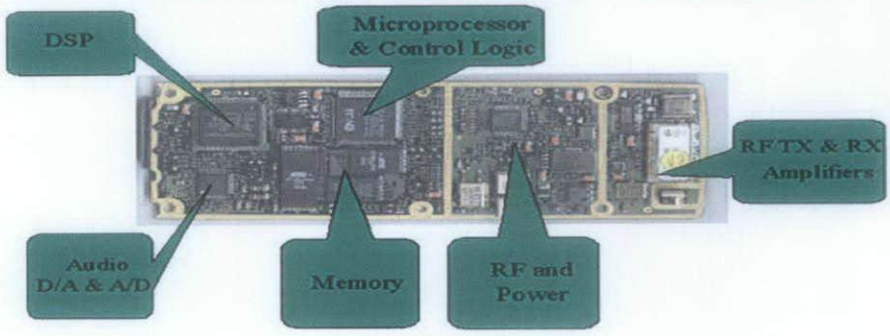


Figure 1: Microprocessor and DSP Chip in Mobile Phone [2]

Inside a mobile phone, it has a powerful Microprocessor and Digital Signal Processing (DSP) chip [7]. The Digital Signal Processing is designed to perform signal manipulation calculations at high speed [7]. This DSP is rated up to 40 MIPS (Millions of Instructions per Second) and handles all the signal compression and decompression [7]. The microprocessor and memory (RAM and ROM) inside handle all of the housekeeping chores for the keyboard and display, which deals with command and control signaling with the base station and also coordinate the rest of the functions on the board [7].

1.2 Problem Statement

Manufacturing Industrial application often needs data to be inspected for example:

- Temperature sensor,
- Pressure sensor,
- Level sensor,
- Fiber Optic Sensor,
- Color Sensor,
- Accelerometer, Positioning Sensor
- Gyro Sensor, etc.

All these sensor data need to be logged and later analyzed by the system inspector [8]. These two processes consume times and need to be performed separately. With available Smartphone processing power, this entire task is at the fingertips of the user. Since people are using advances Smartphone nowadays, then the idea to use it as a device to interact with the device are suitable for data logging, hence store and analyze all the data receive.

There are many other wireless transmission modules that can be used such as IR modules, RF modules, and WIFI. Although, not every module is suitable since the communication range, data transmission speed, price, power consumption and functionality vary from one to another [9].

1.3 Objective and Scope of Study

The objectives / deliverable of the projects are:

i) Device with micro-controller that is capable of establishing communication with Smartphone

ii) Application on Smartphone capable of acquiring, processing and displaying data from the device

The scope of the project includes the development of the mobile phone application, circuit design, circuit fabrication, data acquisition testing, and sensors or input testing.

1.4 Relevancy of the Project

This project is relevant to Electrical and Electronic Engineering academic syllabus of University Technology PETRONAS (UTP). It incorporates knowledge in Communication System, Micro-Controller, Circuit theory, Multimedia Network, and Network Analysis. In addition, it also enhances project management and communication skills.

1.5 Feasibility of the Project within the Scope and Time Frame

For this project, the first semester will cover formulation of methodology and development of mobile phone application. The second semester will consist of full detail in circuit design, full integration of phone and external device/controller.

CHAPTER 2
LITERATURE REVIEW

2.1 Microcontroller

A microcontroller is a small computer on a single integrated circuit containing all the microprocessor sub-part such like core, memory, and programmable input/output peripherals [10]. It is commonly abbreviated to µC, uC or MCU. The program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM [10]. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications [11].

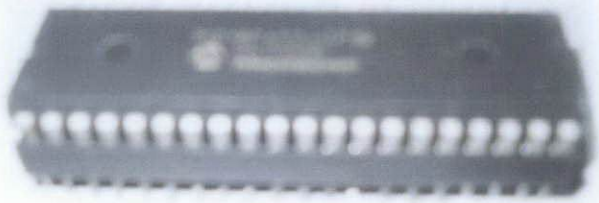


Figure 2: Programmable IC (PIC) Microchip

PICs Microcontroller is a leading microcontroller family from Microchip [12]. PICs are available in very small packages with only few pins and also as powerful 32-bit microcontrollers with many peripheral modules, I/O pins and offer UART capabilities to be integrated with Bluetooth modules [13].

2.2 Windows Mobile Development SDKs

Microsoft typically releases Windows Phone Software Development Kits (SDKs) that work in conjunction with their Visual Studio development environment [14]. These SDKs include emulator images for developers to test and debug their applications while writing them. Microsoft also distributes Visual Studio 2008 / 2005 Professional Editions, and server / database counterparts to students as downloads free of charge via its Dream Spark program [15].

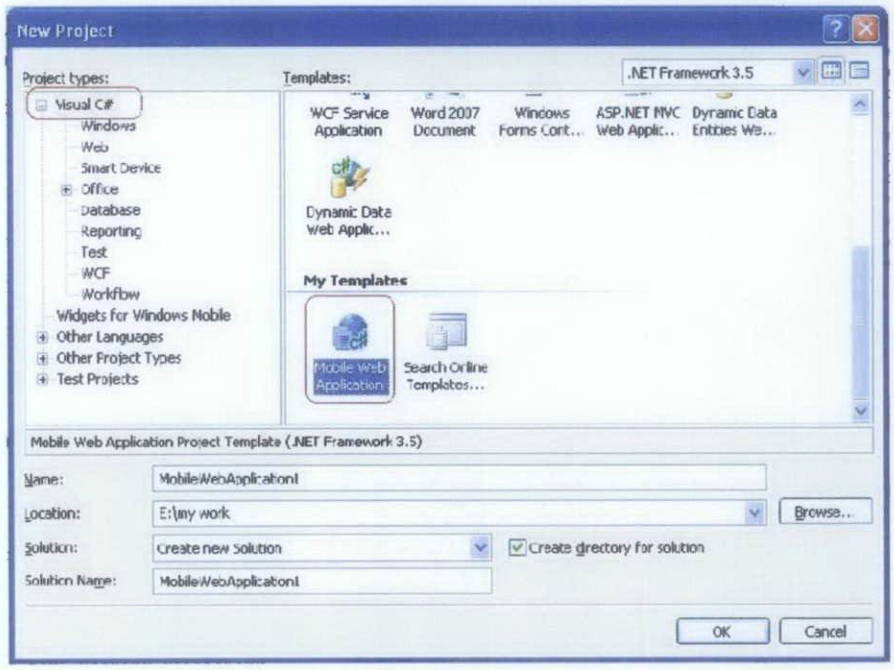


Figure 3: Windows Mobile SDKs Layout Project

These SDKs application development tools will ease new developers since they do not need to master certain language such as Java. The windows mobile SDKs provide tool and libraries to start developing windows mobile application using Visual Studio 2008 [16]. This SDK combine with Visual Studio to create a powerful mobile device development platform while providing the tools and features necessary to make developing application simpler .

2.3 Bluetooth

Bluetooth is an open wireless technology standard, used to transfer and communicate data over short distances. In other words, it is an electronic cable-replacement. Using a tiny, inexpensive radio chip that produces a short-range, low-cost radio frequency, Bluetooth eliminates the need for cables running between computer, printer, and mobile phone [17].

Bluetooth also used for exchanging data over short distances (using short wavelength radio transmissions) from fixed and mobile devices, creating personal area networks (PANs) with high levels of security [18]. Created by telecoms vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization. Today Bluetooth is managed by the Bluetooth Special Interest Group [19].

An interesting aspect of the technology is the instant formation of networks once the Bluetooth devices come in range of each other. Bluetooth adopts a flexible multiple piconet structure for communication which consists of two or more devices that occupy the same physical channel. A piconet is a collection of devices connected via Bluetooth technology in an ad hoc fashion which consists of master and up to seven slaves. A master unit, which initiates the communication, uses point to point or point to multipoint communication to transmit and receive signals from slave or slaves [19].

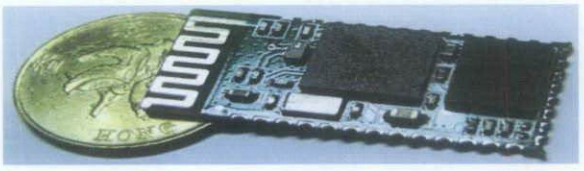


Figure 4: Bluetooth Module

The Bluetooth Module is an electronic device which has a feature of connecting an embedded microcontroller to a PC or mobile phone. This module is used to convert UART with 115200bps, one start bit, 8 data bits, one stop bit, and no parity bit format to Bluetooth UART protocol [20]. It cannot talk to a same device which means it cannot perform any connection between 2 embedded devices [19].

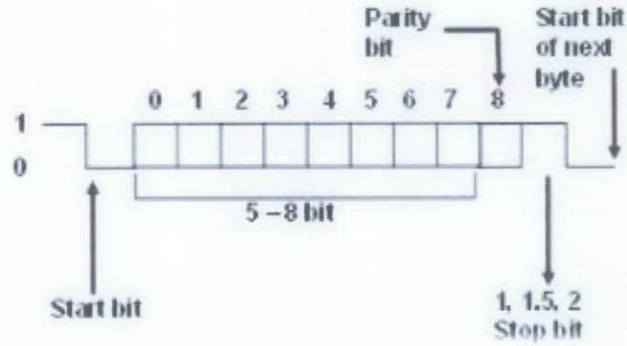


Figure 5: Start, Data, Parity and Stop bit for UART [20]

Bluetooth Module Configuration Block was shown in the Figure below telling what the main component inside the module is and how it communicates.

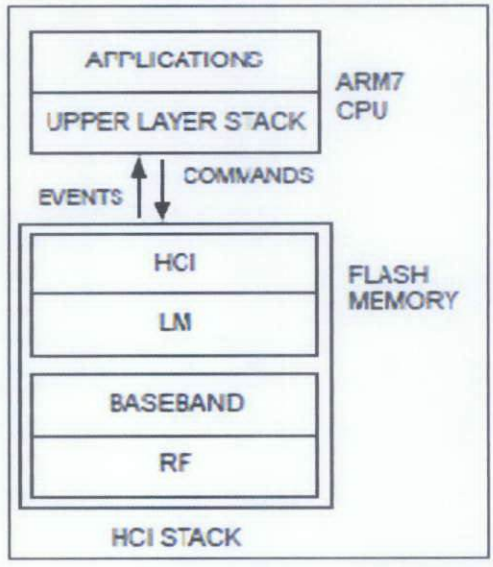


Figure 6: Bluetooth Module Block Diagram [21]

2.4 Infra-Red Communication

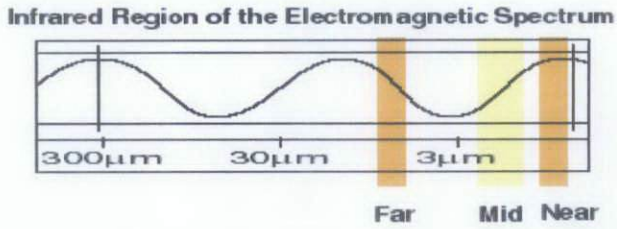


Figure 7: Infra-Red Region of the EM spectrum [22]

Infra red is an energy radiation which its frequency are below our eyes sensitivity. IR communication units communicate with its designated appliances using light waves in the infrared spectrum [22]. The communication layer which is formed by the emitter, usually emits the signal at a certain frequency, between 30 - 60 kHz, most often at 36 kHz, and the IR receives modules on appliances tunes to that particular frequency to read input from users. For an instance, infrared is fairly reliable and does not cost much to build into devices. The only drawback that limit the implementation of Infra Red are that both transmitter and receiver modules have to be in line of sight (LOS) and a transmitter cannot send data to multiple receiver at the same time [22].

Some major difference between Bluetooth and Infra-Red Communication are shown in the table below:

Table 1: Major Differences between Bluetooth and Infra Red Communication[23]

Communication Type	**IrDA-Data**	**Bluetooth**
Physical Media	Infrared	RF (2.4 GHz)
Communications Range	Up to at least 1m	10cm to 100m
Connection Type, Direction	Point-to-Point, Narrow Angle (30 degrees)	Multipoint, Omni-directional
Maximum Data Rate	4Mbps (16Mbps on the way)	1Mbps (aggregate)

Security	Physical limitations offer some built-in protection	Authentication, encryption, spread spectrum

2.5 Wi-Fi

Figure 8: Wi-Fi Trademark Logo

A Wi-Fi or Wireless Fidelity enabled device such as a personal computer, video game console, Smartphone, or digital audio player can connect to the Internet when within range of a wireless network connected to the Internet. The coverage of one or more (interconnected) access points called hotspots when offering public access generally comprises an area the size of a few rooms but may be expanded to cover many square miles, depending on the number of access points with overlapping coverage. Wi-Fi allows the deployment of local area networks (LANs) without wires for client devices, typically reducing the costs of network deployment and expansion. Spaces where cables cannot be run, such as outdoor areas and historical buildings, can host wireless LANs [24].

As of 2010 manufacturers are building wireless network adapters into most laptops. The price of chipsets for Wi-Fi continues to drop, making it an economical networking option included in even more devices. Wi-Fi has become widespread in corporate infrastructures [24].

Different competitive brands of access points and client network-interfaces can inter-operate at a basic level of service. Products designated as "Wi-Fi Certified" by the Wi-Fi Alliance are backwards compatible. "Wi-Fi" designates a

globally operative set of standards: unlike mobile phones, any standard Wi-Fi device will work anywhere in the world [24].

Some major differences between Bluetooth and WIFI are shown in the table below:

Table 2: Major Differences between Bluetooth and Wifi [25]

Specifications authority:	Bluetooth SIG	IEEE, WECA, WiFi
Ease of Use:	Fairly simple to use. Can be used to connect upto seven devices at a time. It is easy to switch between devices or find and connect to any device.	It is more complex and requires configuration of hardware and software.
Primary Devices:	Mobile phones, mouse, keyboards, office and industrial automation devices	Notebook computers, desktop computers, servers
Range:	10 meters	100 meters
Security:	More secure than WiFi as it covers shorter distances and has a 2 level password protection	Less secure. It has all the risks associated with any other network. If someone accesses one part, the rest can also be accessed
Power Consumption:	Low	High
Frequency:	2.4 GHz	2.4 GHz
Bandwidth:	Low (800 Kbps)	High (11 Mbps)
Hardware requirement:	Bluetooth adaptor on all the devices connecting with each other	Wireless adaptors on all the devices of the network, a wireless router and/or wireless access points

2.6 Cytron Bluetooth KC Wirefree

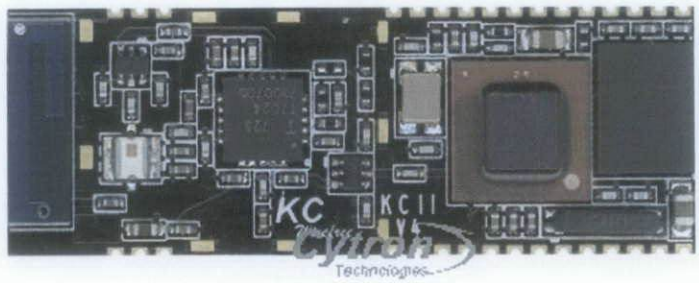


Figure 9: KC Wirefree Bluetooth Module

KC Wirefree Bluetooth OEM Amplified Module is designed for maximum wireless range and penetration [26]. The powerful onboard amplifier boosts both the transmission and reception signals which typically exceed Bluetooth minimum specifications for a class 1 rated device. This module is a surface mount PCB circuit that provides fully embedded which it is already can be used for Bluetooth wireless technology. The firmware is easy to customize for external device interaction, or for optimizations such as minimal power consumption, high speed response, and other proprietary features. Custom firmware is easily pre-loaded into these highly tuned and tested modules so that they are ready to install without additional procedures [26].

Some other features of KC Wirefree Bluetooth Module are [26]:

- • Integrated chip antenna
- • 128-bit encryption security
- • Amplified transmission and reception
- • Range up to 100m LOS
- • SPI interface, up to 24MHz
- • Multipoint capability

2.7 Temperature Sensor (LM35DZ)

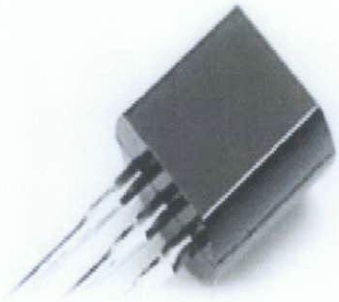


Figure 10: Temperature (Centigrade) Sensor [27]

The LM35DZ series are precision integrated temperature sensors which the output voltage is linearly proportional (centigrade) temperature. This sensor does not need external calibration or trimming to provide typical accuracies [27]. It also gives an output impedance, linear output, and precise inherent calibration sensor making interfacing with microcontroller to readout or control circuitry more easily. Some other feature of this Temperature sensor [27]:

- Calibrated directly in ° Celsius
- Linear + 10.0mV/° Celsius scale factor
- 0.5° Celsius accuracy guarantee able (at +25° Celsius)
- Rated from 0 to 150° Celsius for Basic Centigrade and from -55 to 150 ° Celsius for full Centigrade Measurement
- Suitable for remote applications
- Low cost due to wafer level trimming
- Operates from 4 to 30 volts
- Low self heating, 0.08° Celsius in still air
- Low impedance output, 0.1ohm for 1mA load

2.8 High Performance Serial and Telnet Communications Software for Windows Mobile – vxHpc

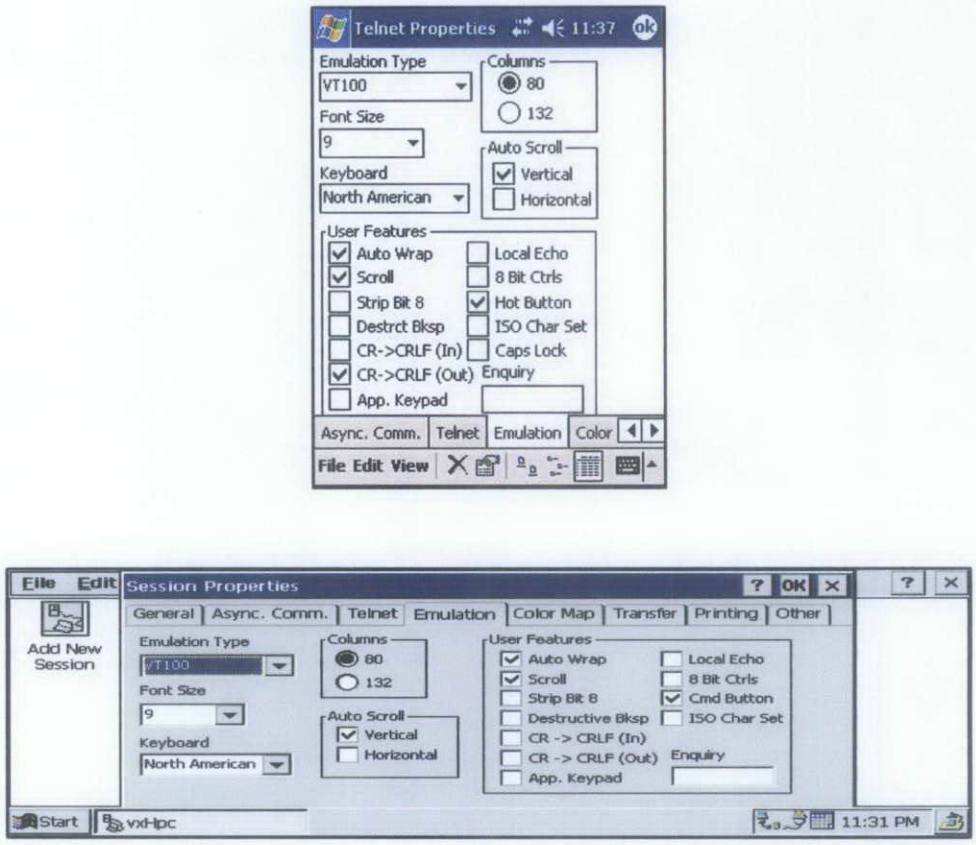


Figure 10: vxHpc terminal emulation software [3]

VxHpc is a serial and telnet communication software developed by the Cambridge Computer Corporation for Windows Mobile operating system Smartphone [3a]. This software works similar as HyperTerminal software in Windows XP which cooperates asynchronous connection that eases the communication between mobile phone and the external sensor device [3].

Some Features that available in this software are [3]:

- Multiple communications protocol capability.
    - Direct connect serial
    - Modem async.
    - Telnet (TCP/IP)
    - Infrared, Bluetooth, Cellular, Wi-Fi
- Full color support.
    - Assign a color to any visual attribute.
    - Supports host generated color attribute commands.
- International keyboard capabilities.
    - North American, British, Canadian French, Danish, Dutch, Finnish, French, German, Italian, Portugese, Spanish, Swedish, Swiss
- Various font sizes. Supports 5 point to 20 point
- Stylus support
    - vxHpc allows copy/paste between host sessions and Windows CE / Pocket PC documents.
- Capture screen and printer text to any selected file.
- File transfer.
    - Kermit, X modem (checksum and CRC), X modem 1K, X modem 1KG, Y modem, Y modem G, Z modem, Text/ASCII
    - Execute a file transfer from a script
- Hot keys for frequently used commands and keyboard remapping.
- Data scope capability for monitoring external devices and performing data logging.
- Trace mode for debugging.
- Backup and restore session properties.
- Developer user friendly and easily to integrate with external device.
- Operates on all Handheld, Handheld Pro, Palm-size and Pocket PCs
- Windows CE 1.0, 2.0, 2.11, 3.0, 4.0, 4.2, 5.0 and 6 compatible

CHAPTER 3

METHODOLOGY

3.1 Methodology Identification

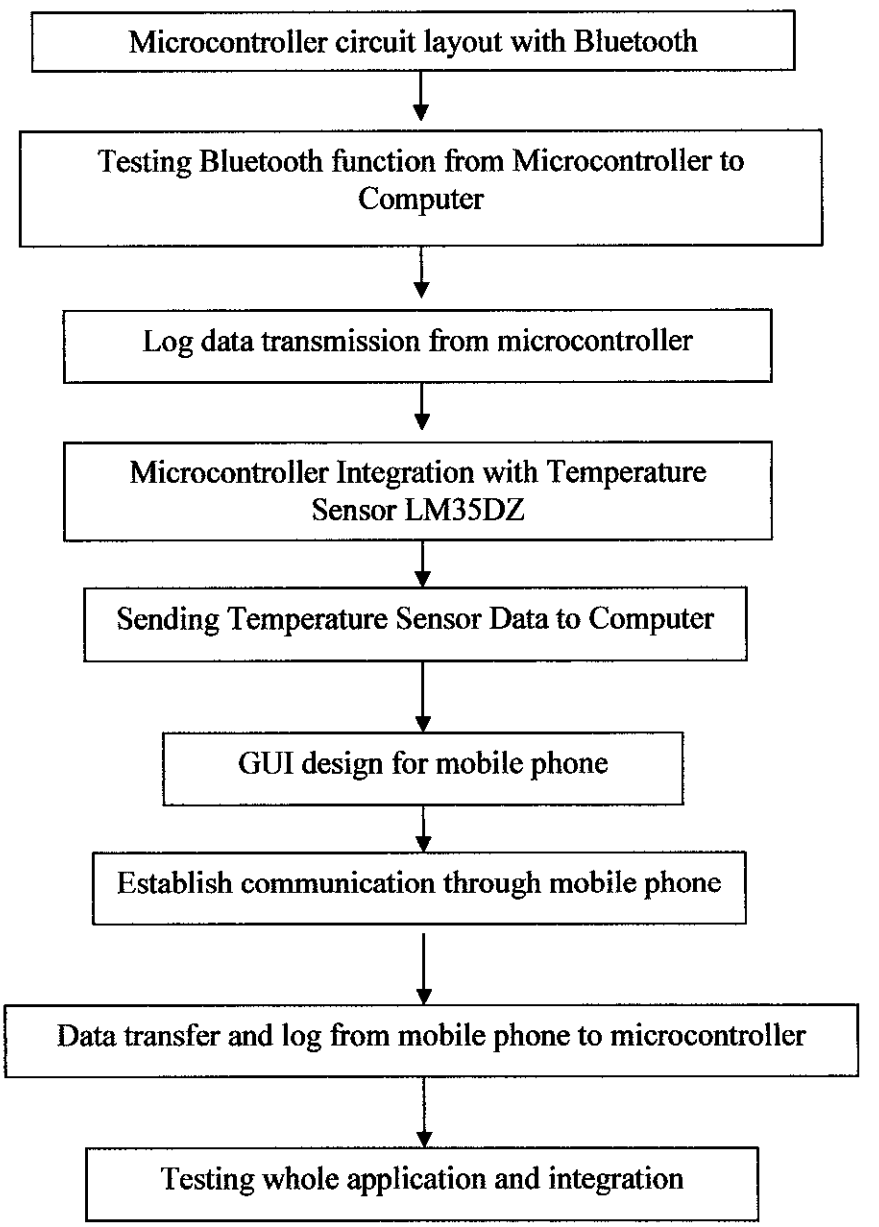


Figure 12: Process flow chat

3.2 Hardware / Tools and Software

Table 3: List of software needed

Software	Function / Use
Visual Studio	Web Editing software for SDKs
Windows Mobile SDKs	Application Development for windows mobile
PSPICE	Circuit design and simulations
MATLAB	Data calculations
EAGLE	Circuit and PCB design
MPLAB 8.1	Programming platform for microcontroller
C compiler	Compile c programming to a command that 16f877A microcontroller understand
vxHpc	Terminal Emulation for Windows mobile

Table 4: List of hardware needed

Hardware	Functions / Use
Battery (Lithium Polymer)	Supply power
Microcontroller 16f877A	Microcontroller used for external hardware to sent data
Bluetooth Module	Bluetooth module to be connected to PIC 16f877A using UART
PIC Programmer Hardware	Used to program PIC 16F877A
LM35DZ Temperature Sensor	Temperature Sensor used as the input sensor
O2 XDA Mobile Phone	Smartphone Equipped with windows mobile 5.0

CHAPTER 4
RESULT & DISCUSSION

4.1 Microcontroller Circuit Layout with Bluetooth

The flow below shows the inter-relationship between the entire main components for the system. Microcontroller is connected directly to the Temperature sensor while phone and microcontroller are connected by Bluetooth communication. The Bluetooth modules perform its function as the UART converter to serial communication (asynchronous communication) so the phone will be able to sent the right command to the external sensor device as well as receiving any data submitted by the microcontroller device.

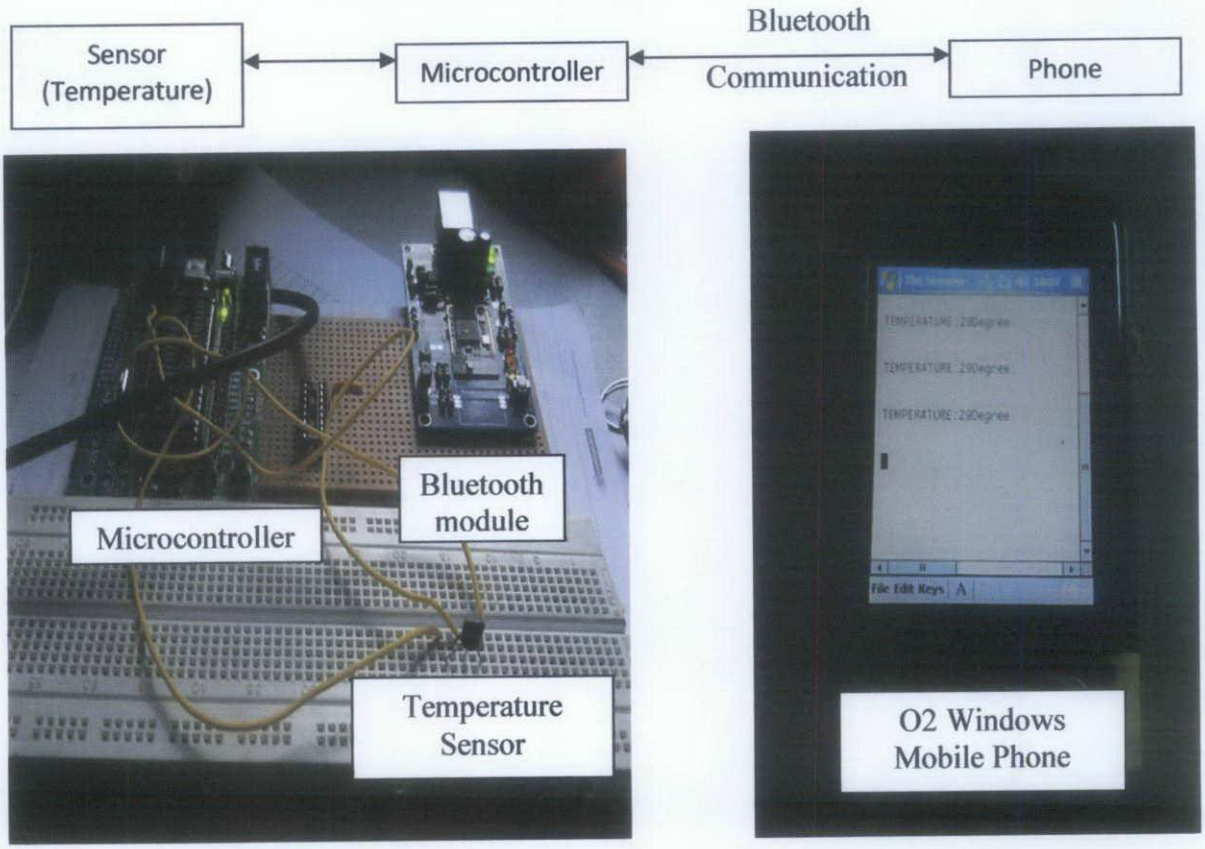


Figure 13: Integration of Temperature Sensor and Microcontroller to Phone

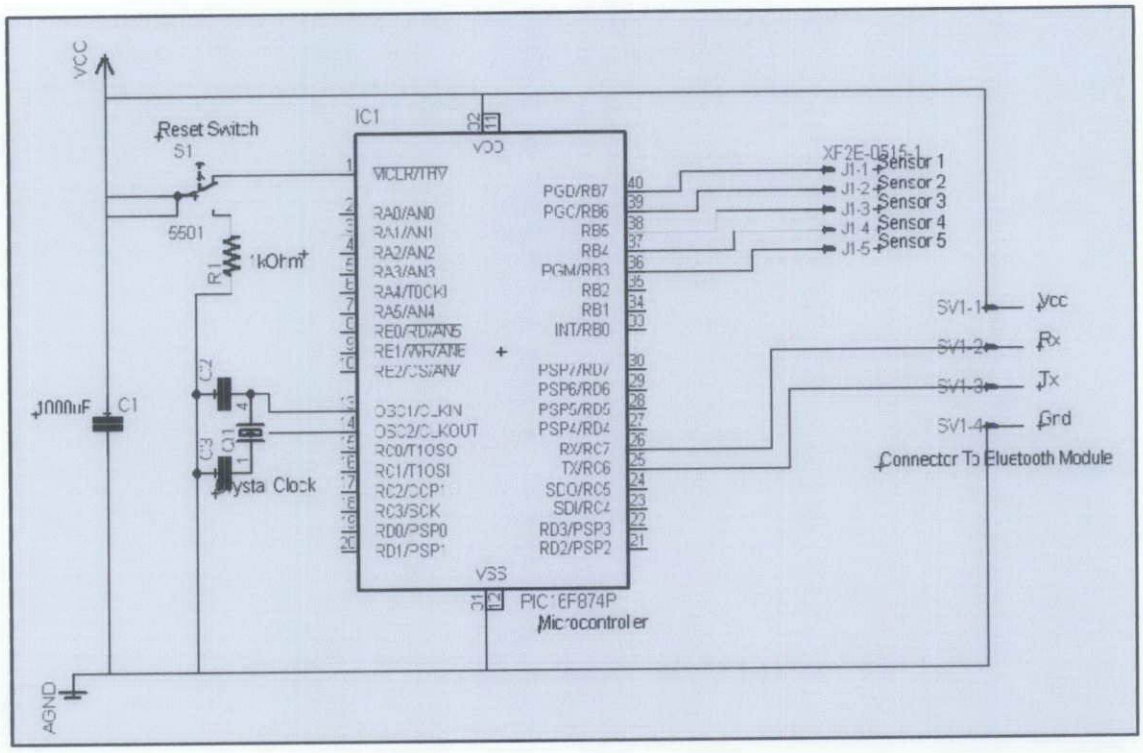


Figure 14: Circuit Design Layout

The design by using Eagle 5.1 software in Figure 14 shows the prototype circuit for the external sensor device. It uses 5 digital I/O as the inputs connectors. As for the Bluetooth Module, it will simply connected to the RX and TX of the PIC while their supply will be common for all components.

Since the prototype will be using LM35DZ as the analog sensors, it will be directly connected to one of the analog ports at pot A. This sensor will become an internal sensor where it will be permanently attached to the circuit as above.

4.2 Log Data Transmission to Computer using Hyper Terminal

The log data between Computer and Microcontroller through Bluetooth is a testing method for the entire external device to check whether it is ready to send the real data or not. For this testing, the sensor value has been ignored. Some configuration need to be provided to enable asynchronous communication between computers/laptop hyper terminal and the device itself.

As soon as the computer/laptop hyper terminal and the external device were connected, a test of sending 'OK' using that computer/laptop to the device has been perform as a command submission before the device can send the correct data that we need. This is some sort of like giving command or interrogation to the device, so that it would not give endless data. This test was carried out successfully and after that command submission, the data from the external device transmission commence as shown below:

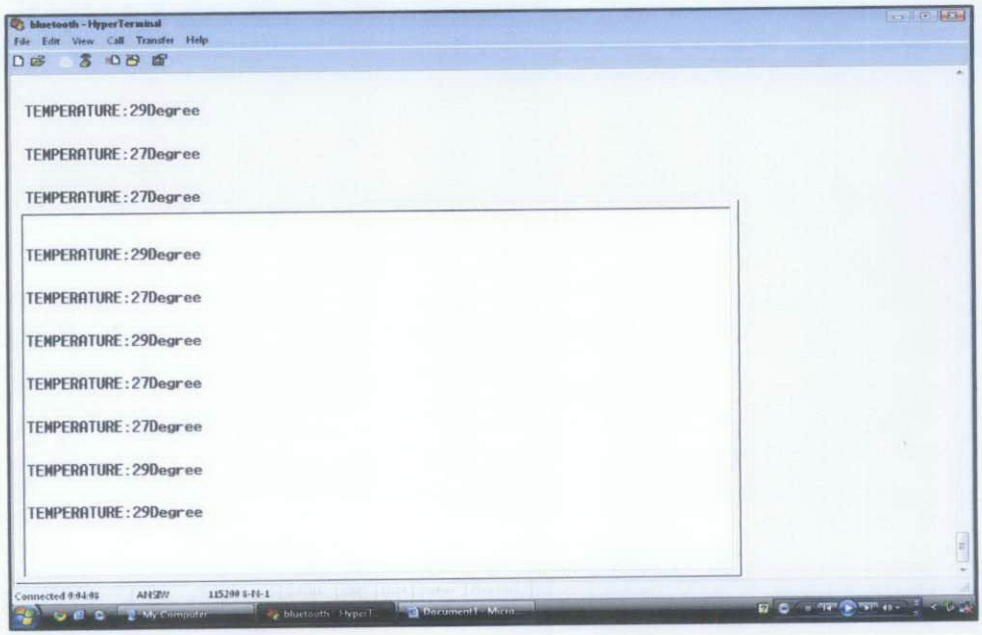


Figure 15: Data Log of the Bluetooth communication to computer

A full range signal test with respect to the distance between external devices and computer/laptop Bluetooth ports has been carried out. The result of the test can be accessed table 3 in appendix F. From the result, the maximum distance

range that the Bluetooth can go is up to 33 meters, where signal conditioning becomes poorer as the distance increased. An example of miscellaneous error is shown below which data sent to the hyper terminals is not readable.

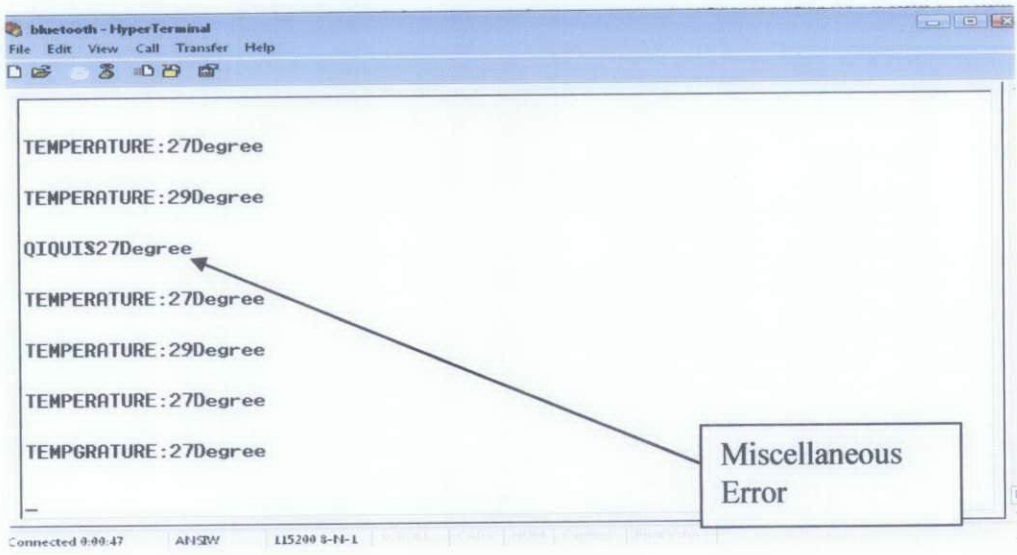


Figure 16: Miscellaneous error when the signal conditions worsen

4.3 Microcontroller Integration with Temperature Sensor

To integrate temperature sensor, the value of the voltage output of the temperature sensor need to be matched with the programming inside the microcontroller.

The PIC16F877A which used as the microcontroller has an 8 bit analog to digital (ADC) 0 to 5 volt converter built-in which needs to be programmed.

Hence:

Voltage Level: 0 -5 Volt = level

= 256 levels

= 0 – 255

So every 1 bit change = —— = 0.0196volt

In the data sheet of the temperature sensor, every 1 degree increased will change the output voltage for 0.010volt. Hence, the range of the output temperature is from 0 volt to 1.500 volt for temperature range from 0 – 150 degrees Celsius. The conversion and temperature calibration into programming of the PIC was put in table 4 in Appendix G. From the result, we can expect a low efficiency in PIC memory since it has a 2 degrees Celsius error where it jumps for every odd numbers. This problem can be tackle by using a higher bit range of PIC Microcontroller such as 18F where more memories are applied.

4.4 Sending Temperature Sensor Data to Computer

Since communication testing and temperature sensor data calibration has been carried out, it is time for real temperature data submission to computer/laptop. At first a normal room temperature measure and test were carried out to check the reliability of the temperature data. The data received was around 29 and 31 degrees Celsius.

Next, temperature rise testing will be carried by applying heat to the temperature sensor. In this test, sudden temperatures changes need to be apply to the temperature sensor. By using an iron (as shown in figure below), heat is applied to the temperature sensor, hence producing change on the data from a normal room temperature to a higher temperature up to 70 degrees Celsius.

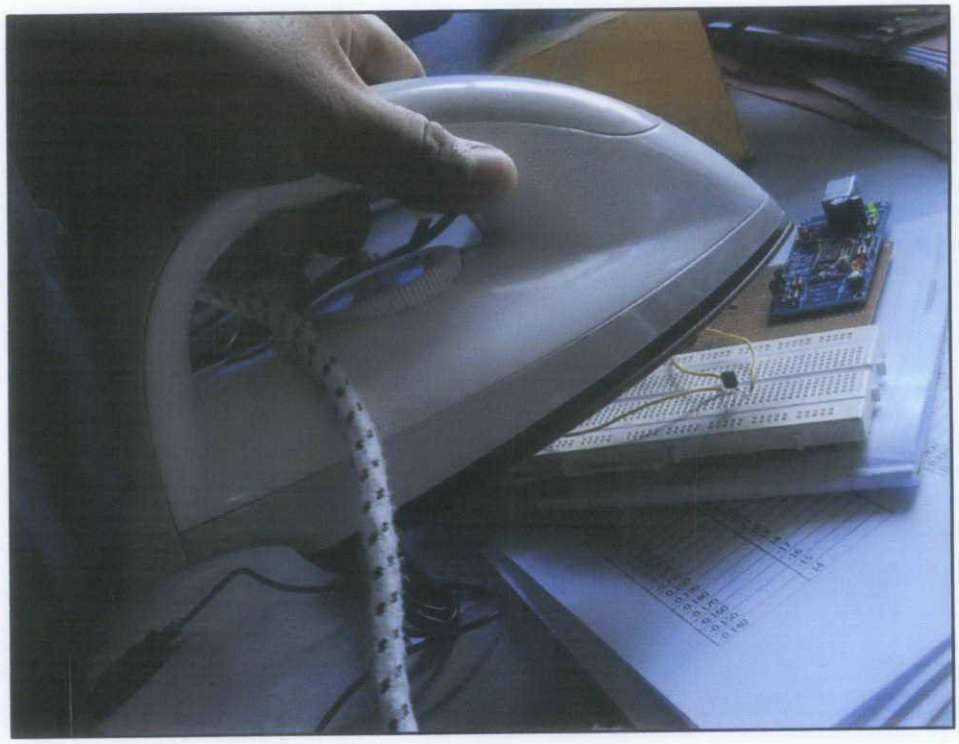


Figure 17: Heating up the temperature sensor

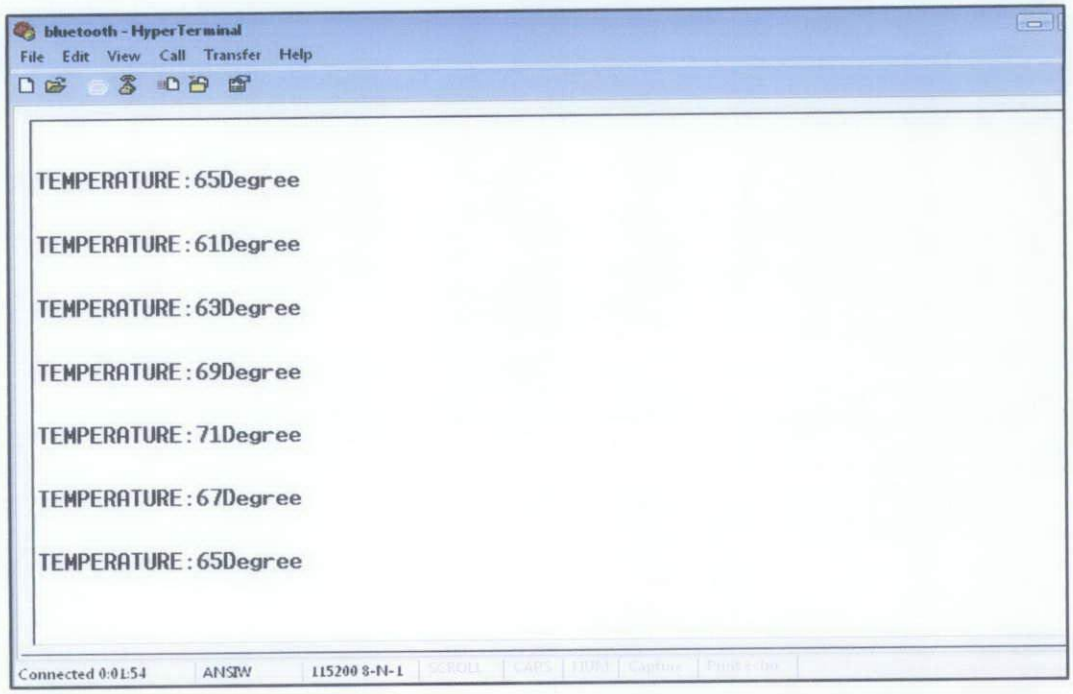


Figure 18: Temperature sensor data after heating

4.5 GUI for Mobile Phone

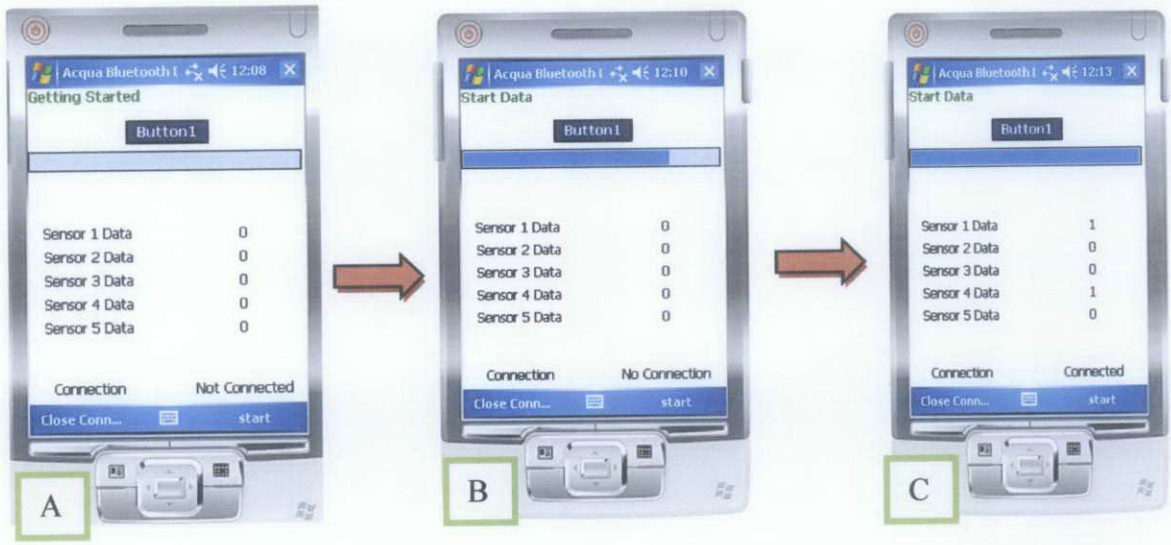


Figure 19: Visual studio Mobile SDK 5.0 emulator shows the start page of the application

By using Visual Studio, the basic phone application has been prepare, where it will acquire 5 digital sensor output and it will obtain the data through Bluetooth. The application for the mobile phone is simple and user friendly where anyone can use it for any applicable application.

In the first picture (A), this was the starting menu for the application where in consist of few button such as Button 1, Start Button, and Close Connection Button. Button 1 is used to start the Bluetooth communication where, if it is clicked, the status bar will start operating showing the communication progress – refer second picture (B). Then after the progress show 100% finished, we can start acquiring data by clicking the Start button, then all the sensor data value will changed as shown in picture 3(C).

4.6 Establishment of Communication from Mobile Phone to Sensor Device

As the prototype of the external device was ready to send any data to the mobile phone a test by using mobile phone must be carried out to ensure the reliability of the system. By installing vxHpc terminal emulation serial communication for windows mobile to the O2 Xda Smartphone, the test is ready to be commission.

At first, software setup need to be reconfigured as follow:

- Async. Comm tab:
  - Select Port – Bluetooth
  - Configure Tab :
    - Baudrate: 115.2kbps
    - Data Bit : 8
    - Parity: None
    - Stop: 1
    - Flow Control: none
- Telnet Tab:
  - Check Box – Send Nul After CR
  - Emulation Tab:
    - Emulation – VT100
    - Font Size – 10
    - Column – 80
    - Auto Scroll – Vertical
    - User Feature Checkbox
      - Auto Wrap
      - Scoll
      - CR-> CRLF(in)
      - CR-> CRLF(out)
      - App. Keypad
      - Local echo
      - Hot Button

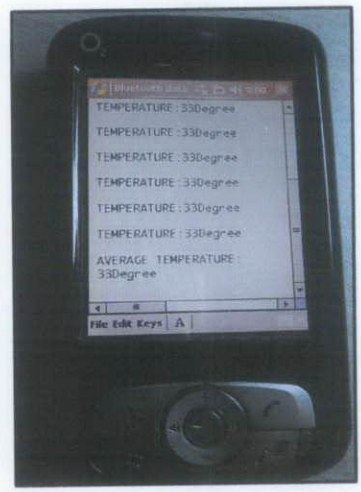


Figure 20: Phone shows the temperature data sent by the sensor device

After the phone software has been setup, we can start data interrogation by sending command 'ts' which denotes temperature sensor command in order to acquire temperature data. Then, 10 sets of temperature data sent to the mobile phone. Right after the last 10^{th} data, it will calculate the average temperature data to be send to the mobile phone. Similarly with digital sensor, we can use other command to select only digital sensor which is denotes to 'ds' and the device will send the condition of 3 digital sensors attached to the prototype digital port. As there are any condition changes on the sensor logic, it will keep updating its condition.

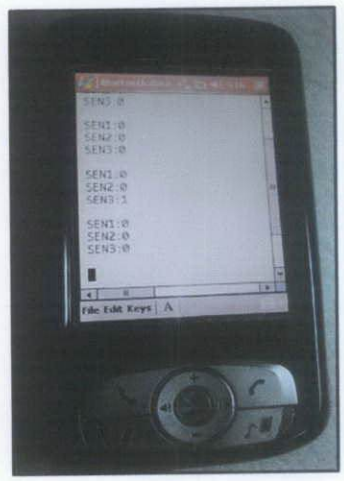


Figure 21: Phone shows the digital data sent by the sensor device

4.7 Final Prototype and Full Integration

To protect the circuit of the prototype, a circuit box was obtained to store the microcontroller device inside and all the important parts such as, battery, Bluetooth module, switch, and general input outputs for the device.

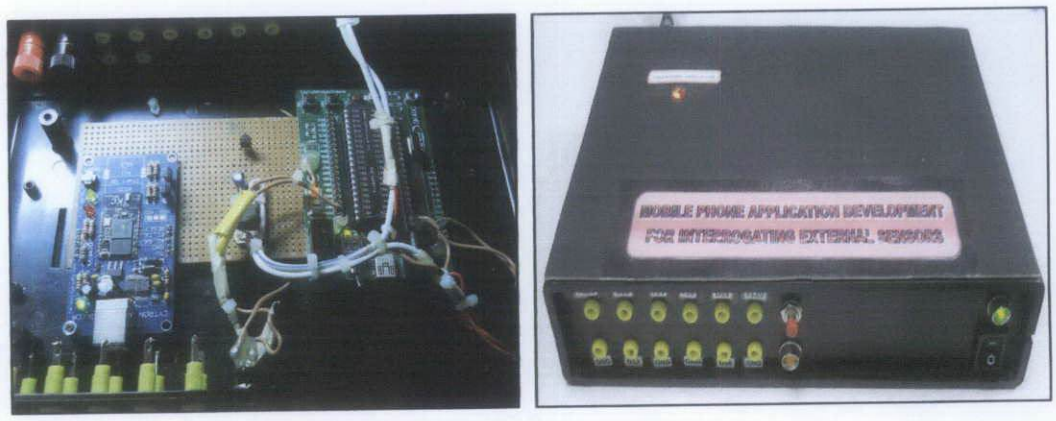


Figure 22: Final Prototype of External Device

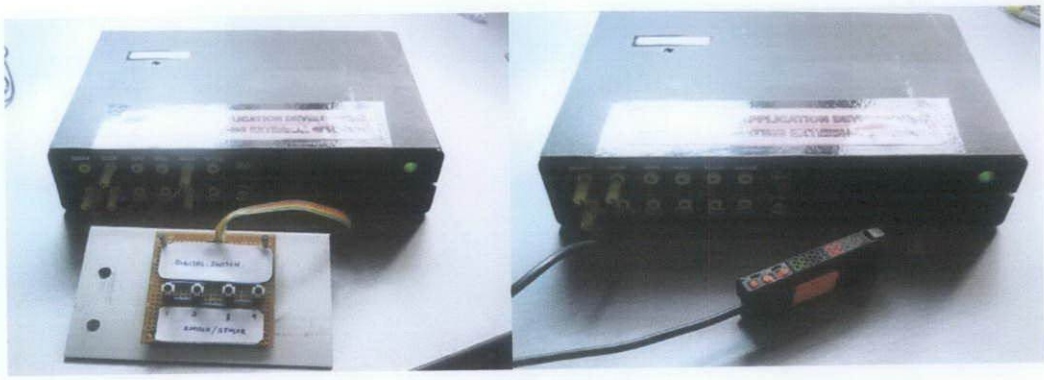


Figure 23: Integration with Digital Sensor – Limit switches and Fiber Optic Amplifier Sensor

The integration with available digital sensor was performed by attaching fiber optic amplifier sensor to the prototype digital port. By giving command for digital mode, the prototype will change its digital condition when the fiber optic sensor detects object within its detection range. One example of using this application (fiber optic sensor) is calculating quantity of rejected products in manufacturing plant for quality monitoring system.

4.7.1 PIC Memory Utilisation

As the prototype are been able to do multiple task on gathering, analyzing, and sending the sensor data to the mobile phone, the PIC Microcontroller program memory has already reached to about eighty percent of its entire program storage. In this case, adding more sensor or function of the device itself is not possible.

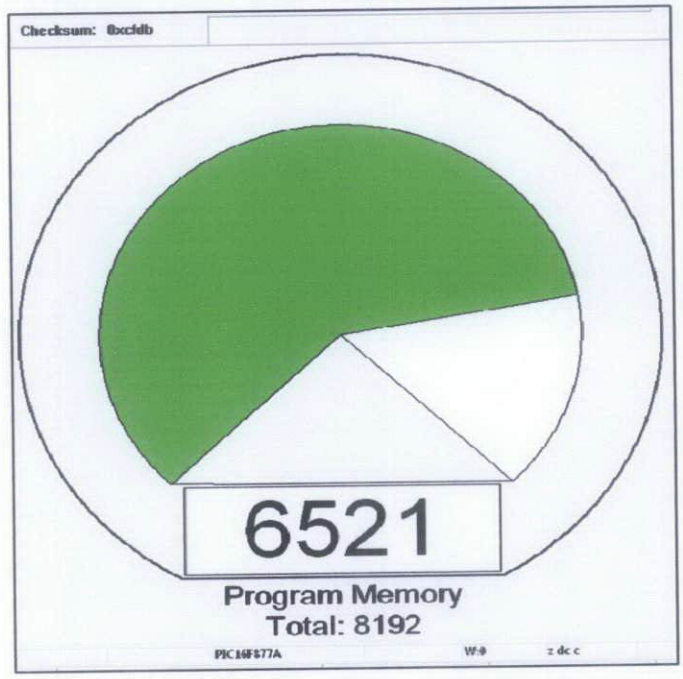


Figure 24: Program Memory of PIC 16F877A

Hence, the solution to solve the problems of the memory problem, the PIC Microcontroller that need to be used will need to be revised as more internal memory of the PIC is needed for the program memory and ADC data bit conversion as mention. Another method on solving the problem is by attaching more PIC to perform several tasks that divided by its own group for example:

- Digital Sensor Port Tasking
- Analog Sensor Port Tasking
- RX/TX Data utilization Tasking
- Data Analyze Tasking, and
- General Input / Output Tasking

CHAPTER 5
CONCLUSION

5.1 Conclusion

As a conclusion, this project has been progressed to a level where it can already be used for specific applications. Although program memory of the PIC 16F877A Microcontroller has reached eighty percent of its memory, it served its purpose to gather, analyze and send the data to the mobile phone. The digital port and analog port also now can be easily plug and play to the device since a simple connectors used to ease operation. The software developed for the application still need to be revised for further development since it still cannot be fully integrated with the mobile phone. With the help of vxHpc software by Cambridge Computer Corporation, it can establish communication between phone and the external device.

5.2 Recommendation

Future development of the application will need another revision to enhance the user interface (UI). More application sensors will need to be considered so that this project will be more valuable and attractive to potential user. It is recommended that advance phone such as IPhone or Android based system to be used for future work because it has powerful processing power hence faster in processing data, better user interface, big resolution, and accepted by many people for a good operating system based Smartphone. Last but not least, the power consideration for the prototype should be analyze critically so that the user will be able to know what type of supply should be used other than a rechargeable lithium polymer battery.

REFERENCES

[1] Murray, S. (2010, May 4). The Advancements of Mobile Phone Technology. Retrieved August 18, 2010, from http://ezinearticles.com/?The--Advancements- of- Mobile- Phone- Technology&id=4226818

[2] Freeware for Smartphone, Retrieved April 26, 2011 from http://www.smartphone-freeware.com/

[3] Cambridge Computer Computer Corporation vxHpc Software, Retrieved April 26, 2011 from http://www.cambridgevx.com/vxhpc.html

[4] GPS Software for Mobile Phone, Retrieved April 26, 2011 from http://www.gpss.force9.co.uk/

[5] Industrial PDA (EDA), Retrieved April 26, 2011 from http://www.ieimobile.com/downloads/2011_pda_Series_Brochure.pdf

[6] Apple Iphone AppStore, Retrieved April 26, 2011 from http://www.apple.com/iphone/

[7] Marsyal Brain, 1 April 2000, Inside cell phone. Retrieved August 18, 2010, from http://electronics.howstuffworks.com/inside-cell-phone.htm

[8] CAS Data Logger, Retrieved April 26, 2011 from http://www.dataloggerinc.com/

[9] Bradshaw, B. (2009, January 8). Wireless Power For Small Appliances. Retrieved August 18, 2010, from http://ezinearticles.com/?Wireless-¬Power- For- Small- Appliances&id=1860055

[10] PIC Microcontroller, Retrieved April 26, 2011 from, http://www.gnupic.org/

[11] Popa, O. G. (2005, November 11). SPI Bus: Theory and Implementation. Retrieved August 18, 2010, from http://ezinearticles.com/?SPI-Bus:--Theory-and-Implementation&id=95089

[12] Microchip Technology PIC Microcontroller, Retrieved April 26, 2011 From http://www.microchip.com/

[13] Funa, I. (2010, June 22). Popular Microcontrollers. Retrieved August 18, 2010, from http://ezinearticles.com/?Popular-Microcontrollers&id=4527607

[14] Windows SDKs, Retrieved April 26, 2011 From http://msdn.microsoft.com/en-us/windows/bb980924

[15] Microsoft Corp, (2008, February 18), Microsoft Gives Students Access to Technical Software at No Charge to Inspire Success and Make a Difference Retrieved 18 August,2010, from http://www.microsoft.com/Presspass/press/2008/feb08/02-18GSDPR.mspx

[16] Ramana G.V. (2007, September 19), Visual Studio 2008 – New features, Retrieved April 26 from http://msdn.microsoft.com/en-us/windows/bb980924

[17] Etinger, M. (2010, July 12). Bluetooth Accessory Questions - What is the Bluetooth Special Interest Group?. Retrieved August 18, 2010, from

http://ezinearticles.com/?Bluetooth- Accessory- Questions- - - What- is- the--Bluetooth- Special- Interest- Group?&id=4654051

[18] Bradley Mitchell (1999), PAN – Personal Area Network, Retrieved April 26, 2011from http://compnetworking.about.com/od/networkdesign/g/bldef_pan.htm

[19] High Tech Traveler,(2010) Bluetooth On The Road Retrieved August 18, 2010, from http://www.hoovers.com/business-information/-- pageid__13751--/global-hoov-index.xhtml

[20] Asynchronous Communication, Retrieved April 26, 2011 from http://www.webopedia.com/TERM/A/asynchronous.html

[21] KC11- Amplified Bluetooth Module Datasheet, Retrieved April 26, 2011 from www.cytron.com.my

[22] National Aeronautics and Space Administration (NASA), Infra Red Retrieved February 25, 2011 from http://science.hq.nasa.gov/kids/imagers/ems/infrared.html

[23] How Stuff Works – How Remote Controls Work, Infra red Retrieved February 25, 2011 from http://electronics.howstuffworks.com/remote-control2.htm

[24] Nnamdi Charles Akalugwu, Justas Bugnevicius, Christine Evers, Farhad Pourtaran, International University Bremen, (2003, November 4) Bluetooth Introduction, Retrived Apri 26, 2011 from http://www.faculty.iu-bremen.de/birk/lectures/PC101-2003/17bluetooth/bluetooth/intro.html

[25] Wikipedia – Wi-Fi (Wireless Fidelity, 2011) retrieved February 25, 2011 from http://www.webopedia.com/TERM/W/Wi_Fi.html

[26] WiFI vs Bluetooth - Comparison between WiFI & Bluetooth (2011), Retrieved April 26, 2011 from http://www.freewimaxinfo.com/wifi-vs-bluetooth-comparison.html

[27] LM35 Datasheet, Precision Centigrade Temperature Sensors, Retrieved Jan 1, 2011 From www.cytron.com.my

APPENDIX A: FYP1 GANTT CHART

Action Plan / Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Project Proposal															
Topic Research and Study															
Submission Of Preliminary Report															
System Block Configuration															
Submission of Progress Report															
Seminar															
Circuits and Part Identification															
Submission Of Interim Report Draft															
Submission Of Interim Report															
Oral Presentation															

Chart 1: FYP1 Project Flow

Semester Break

Proposed Time Frame

Suggested Time Frame

WEEK ** / ** ACTIVITIES	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Microcontroller circuit layout with Bluetooth	X													
Testing Bluetooth function from Microcontroller to computer	X	X												
Log data transmission from microcontroller		X												
Microcontroller Integration with Temperature Sensor LM35Z			X											
Sending Temperature Sensor Data to computer			X	X										
GUI for mobile phone				X	X	X								
Establish communication through mobile phone								X	X	X	X			
Data transfer and log from mobile phone to microcontroller										X	X	X		
Testing whole application and integration											X	X	X	X

APPENDIX C

SOURCE CODE OF VISUAL STUDIO 2008

MICROSOFT MOBILE SDK's

```
Public Class Form1


    Private Sub Start_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Start.Click
        Label1.Text = "Start Data"                                  //label for text
        Label13.Text = "No Connection"
        Label3.Text = "0"
        Label5.Text = "0"
        Label7.Text = "0"
        Label9.Text = "0"
        Label11.Text = "0"
        ProgressBar1.Value = 0


        For i = 0 To 200                                            // progress bar simulate
            ProgressBar1.Value = i * 0.5
            i = i + 1
        Next i

    End Sub

    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem2.Click
        Label13.Text = "Connected"                                  // message text changed
        Label3.Text = "1"
        Label5.Text = "0"
        Label7.Text = "0"
        Label9.Text = "1"
        Label11.Text = "0"
    End Sub


    Private Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem1.Click
        Label13.Text = "No Connection"
        Label3.Text = "0"
        Label5.Text = "0"
        Label7.Text = "0"
        Label9.Text = "0"
        Label11.Text = "0"
    End Sub

        Private Sub Form1_KeyDown(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles MyBase.KeyDown
        If (e.KeyCode = System.Windows.Forms.Keys.Up) Then
```

```
        'Up
    End If
    If (e.KeyCode = System.Windows.Forms.Keys.Down) Then
        'Down
    End If
    If (e.KeyCode = System.Windows.Forms.Keys.Left) Then
        'Left
    End If
    If (e.KeyCode = System.Windows.Forms.Keys.Right) Then
        'Right
    End If
    If (e.KeyCode = System.Windows.Forms.Keys.Enter) Then
        'Enter
    End If

  End Sub
End Class
```

APPENDIX D

SOURCE CODE FOR PIC16F877A

```
//Author: Ag Shahrin Afwan B Ag Tahir
//Project: Mobile Phone Application Development for Interrogating Sensors Final Year Project 2010-2011
//Project description:Developing software and hardware for mobile data transmission
// Date :2010 - 2011
//
//=============================================================================
// Include file
//=============================================================================
#include <pic.h>


//=============================================================================
// Configuration
//=============================================================================

__CONFIG(0x3F32);


//=============================================================================/
/Define
//=============================================================================
#define ADC_DATA ADRESH                         //define adc port
#define LED1            RB5
#define LED2            RB6
#define LED3            RB7
#define sen1            RB4
#define sen2            RB1
#define sen3            RB0

//=============================================================================
// Function Prototype
// User can write all the necessary function here
//=============================================================================

unsigned char a;
int s1,s2,s3;

void temp_disp(int analog);
void digital_disp();
void delay(unsigned long data)
{
        // this is a delay function for user to use when the program need a delay
        // this function can be call by type : delay(xxxx),
        // user can replace the 'xxxx' with a value to determine how long the program
        // should delay for, the bigger the value, the longer the time of the delay
        for( ;data>0;data-=1);
}

void init(void)                                 // subroutine to initialize
{
        TRISA = 0b11111111;
        TRISB = 0b00011111;
        SPBRG=0x0A;                                     // set baud rate as 115200 baud
        BRGH=1;
        TXEN=1;
        CREN=1;
        SPEN=1;
}

void display(unsigned char c)           // subrountine to display the text on the screen
{
        while (TXIF == 0);
        TXREG = c;
}
```

```
void display2(unsigned int z)                    // subrountine to display the text on the screen
{
        while (TXIF == 0);
        TXREG = z;
}

unsigned char receive(void)                              // subrountine to receive text from PC
{
        while (RCIF == 0);
        a = RCREG;
        return a;
}

void setup_adc ( void )
{
        ADCON0 = 0b10000000;        //Left justified, RA0, RA1 and RA3 as ADC input
        ADCON1 = 0b01000100;        //Fosc/32, channel 0, ADC not active, ADC off
}

unsigned char read_adc ( unsigned char channel )
{
        switch (channel)
        {
                case 1:
                ADCON0 = 0b10000001;
                break;
                case 2:
                ADCON0 = 0b10001001;
                break;
                case 3:
                ADCON0 = 0b10011001;
                break;
                default:
                ADCON0 = 0b10000000;
        }

        delay(5);                        // delay for a while
        ADGO = 1;                                // start conversion
        while (ADGO) continue;
        ADON = 0;                                // Off ADC module
        return ADRESH;                   // return ADC result

}



//=============================================================================
//      Main Function
//      This is the main function where program start to execute
//=============================================================================
void main(void)
{
        int analogs;
        init();
        int sel_sensor;
        int iter;
        int avrg;

        iter=0;
        avrg=0;
        sel_sensor=0;
        LED1=1;
        LED2=0;
        LED3=1;

        while(1)                                 // Wait for 'ok' to be entered by user
        {
                a = receive();
                if (a == 't')
                                {
                                        a = receive();
```

```
                                        if (a == 's')
                                        {
                                        sel_sensor=1;
                                        break;
                                        }
                                }


                        else if(a == 'd')
                                {
                                        a = receive();
                                        if (a == 's')
                                        {
                                        sel_sensor=2;
                                        break;
                                        }
                                }

        }


        display(0x0a);                          //Go to new line
        display(0x0d);

if(sel_sensor==1)
{
        while(iter<10)
        {
                analogs = read_adc(1);
                avrg = avrg+analogs;

                display('T');                           // Text will display on Hyperterminal after 'ts' is entered
                display('E');
                display('M');
                display('P');
                display('E');
                display('R');
                display('A');
                display('T');
                display('U');
                display('R');
                display('E');
                display(':');

                temp_disp(analogs);             //call function display data temperature
                display('D');
                display('e');
                display('g');
                display('r');
                display('e');
                display('e');

                display(0x0a);                          //Go to new line
                display(0x0d);

                LED1=!LED1;
                LED2=!LED2;
                LED3=!LED3;
                delay(180000);
                iter++;
        }




display('A');
display('V');
display('E');
display('R');
display('A');
```

```
display('G');
display('E');
display(' ');

display('T');
display('E');
display('M');
display('P');
display('E');
display('R');
display('A');
display('T');
display('U');
display('R');
display('E');
display(':');
display(0x0a);                          //Go to new line
//display(0x0d);

avrg=avrg/10;
temp_disp(avrg);
display('D');
display('e');
display('g');
display('r');
display('e');
display('e');
sel_sensor=0;
}


if(sel_sensor==2)
{
                display('D');                                   // Text will display on Hyperterminal after 'ds' is entered
                display('I');
                display('G');
                display('T');
                display('A');
                display('L');
                display(':');
                display(0x0a);
                delay(80000);
                digital_disp();
                s1=sen1;s2=sen2;s3=sen3;

        while(1)
        {
                if(!((sen1==s1)&&(sen2==s2)&&(sen3==s3)))
                {
                digital_disp();
                s1=sen1;s2=sen2;s3=sen3;
                }

        delay(200000);
        }

}
}


void digital_disp()
{
        if(sen1&&sen2&&sen3)
        {
        display('S');
        display('E');
        display('N');
        display('1');
        display(':');
        display('0');
                display(0x0a);                          //Go to new line
                display(0x0d);
```

```
display('S');
display('E');
display('N');
display('2');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('3');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
        display(0x0a);                  //Go to new line
        display(0x0d);
}

else if(sen1&&sen2&&!sen3)
{
display('S');
display('E');
display('N');
display('1');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('2');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('3');
display(':');
display('1');
        display(0x0a);                  //Go to new line
        display(0x0d);
}

else if(sen1&&!sen2&&sen3)
{
display('S');
display('E');
display('N');
display('1');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('2');
display(':');
display('1');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('3');
display(':');
display('0');
```

```
            display(0x0a);                  //Go to new line
            display(0x0d);
            display(0x0a);                  //Go to new line
            display(0x0d);
    }

    else if(sen1&&!sen2&&!sen3)
    {
    display('S');
    display('E');
    display('N');
    display('1');
    display(':');
    display('0');
            display(0x0a);                  //Go to new line
            display(0x0d);
    display('S');
    display('E');
    display('N');
    display('2');
    display(':');
    display('1');
            display(0x0a);                  //Go to new line
            display(0x0d);
    display('S');
    display('E');
    display('N');
    display('3');
    display(':');
    display('1');
            display(0x0a);                  //Go to new line
            display(0x0d);
            display(0x0a);                  //Go to new line
            display(0x0d);
    }

    else if(!sen1&&sen2&&sen3)
    {
    display('S');
    display('E');
    display('N');
    display('1');
    display(':');
    display('1');
            display(0x0a);                  //Go to new line
            display(0x0d);
    display('S');
    display('E');
    display('N');
    display('2');
    display(':');
    display('0');
            display(0x0a);                  //Go to new line
            display(0x0d);
    display('S');
    display('E');
    display('N');
    display('3');
    display(':');
    display('0');
            display(0x0a);                  //Go to new line
            display(0x0d);
            display(0x0a);                  //Go to new line
            display(0x0d);
    }

    else if(!sen1&&sen2&&!sen3)
    {
    display('S');
    display('E');
    display('N');
    display('1');
```

```
display(':');
display('1');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('2');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('3');
display(':');
display('1');
        display(0x0a);                  //Go to new line
        display(0x0d);
        display(0x0a);                  //Go to new line
        display(0x0d);
}

else if(!sen1&&!sen2&&sen3)
{
display('S');
display('E');
display('N');
display('1');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('2');
display(':');
display('1');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('3');
display(':');
display('0');
        display(0x0a);                  //Go to new line
        display(0x0d);
        display(0x0a);                  //Go to new line
        display(0x0d);
}

else if(!sen1&&!sen2&&!sen3)
{
display('S');
display('E');
display('N');
display('1');
display(':');
display('1');
        display(0x0a);                  //Go to new line
        display(0x0d);
display('S');
display('E');
display('N');
display('2');
display(':');
display('1');
        display(0x0a);                  //Go to new line
        display(0x0d);
```

```
            display('S');
            display('E');
            display('N');
            display('3');
            display(':');
            display('1');
                        display(0x0a);                          //Go to new line
                        display(0x0d);
                        display(0x0a);                          //Go to new line
                        display(0x0d);
            }
                        LED1=!LED1;
                        LED2=!LED2;
                        LED3=!LED3;
}


void temp_disp(int analog)
{
if(analog==0)
{
                        display('0');
}

else if(analog==1)
{
                        display('1');
}

else if(analog==2)
{
                        display('3');
}

else if(analog==3)
{
                        display('5');
}

else if(analog==4)
{
                        display('7');
}

else if(analog==5)
{
                        display('9');
}

else if(analog==6)
{
                        display('1');
                        display('1');
}

else if(analog==7)
{
                        display('1');
                        display('3');
}

else if(analog==8)
{
                        display('1');
                        display('5');
}

else if(analog==9)
{
                        display('1');
                        display('7');
}
```

```
else if(analog==10)
{
                    display('1');
                    display('9');
}

else if(analog==11)
{
                    display('2');
                    display('1');
}

else if(analog==12)
{
                    display('2');
                    display('3');
}

else if(analog==13)
{
                    display('2');
                    display('5');
}

else if(analog==14)
{
                    display('2');
                    display('7');
}

else if(analog==15)
{
                    display('2');
                    display('9');
}

else if(analog==16)
{
                    display('3');
                    display('1');
}

else if(analog==17)
{
                    display('3');
                    display('3');
}

else if(analog==18)
{
                    display('3');
                    display('5');
}

else if(analog==19)
{
                    display('3');
                    display('7');
}

else if(analog==20)
{
                    display('3');
                    display('9');
}

else if(analog==21)
{
                    display('4');
                    display('1');
}
```

```
else if(analog==22)
{
                    display('4');
                    display('3');
}

else if(analog==23)
{
                    display('4');
                    display('5');
}

else if(analog==24)
{
                    display('4');
                    display('7');
}

else if(analog==25)
{
                    display('4');
                    display('9');
}

else if(analog==26)
{
                    display('5');
                    display('1');
}

else if(analog==27)
{
                    display('5');
                    display('3');
}

else if(analog==28)
{
                    display('5');
                    display('5');
}

else if(analog==29)
{
                    display('5');
                    display('7');
}

else if(analog==30)
{
                    display('5');
                    display('9');
}

else if(analog==31)
{
                    display('6');
                    display('1');
}

else if(analog==32)
{
                    display('6');
                    display('3');
}

else if(analog==33)
{
                    display('6');
                    display('5');
}
```

```
else if(analog==34)
{
                    display('6');
                    display('7');
}

else if(analog==35)
{
                    display('6');
                    display('9');
}

else if(analog==36)
{
                    display('7');
                    display('1');
}


else if(analog==37)
{
                    display('7');
                    display('3');
}

else if(analog==38)
{
                    display('7');
                    display('5');
}

else if(analog==39)
{
                    display('7');
                    display('7');
}

else if(analog==40)
{
                    display('7');
                    display('9');
}

else if(analog==41)
{
                    display('8');
                    display('1');
}

else if(analog==42)
{
                    display('8');
                    display('3');
}

else if(analog==43)
{
                    display('8');
                    display('5');
}

else if(analog==44)
{
                    display('8');
                    display('7');
}

else if(analog==45)
{
                    display('8');
                    display('9');
```

```
}
else if(analog==46)
{
                    display('9');
                    display('1');
}

else if(analog==47)
{
                    display('9');
                    display('3');
}
else if(analog==48)
{
                    display('9');
                    display('5');
}
else if(analog==49)
{
                    display('9');
                    display('7');
}
else if(analog==50)
{
                    display('9');
                    display('9');
}
else if(analog==51)
{
                    display('1');
                    display('0');
                    display('1');
}
else if(analog==52)
{
                    display('1');
                    display('0');
                    display('3');
}
else if(analog==53)
{
                    display('1');
                    display('0');
                    display('5');
}
else if(analog==54)
{
                    display('1');
                    display('0');
                    display('7');
}
else if(analog==55)
{
                    display('1');
                    display('0');
                    display('9');
}
else if(analog==56)
{
                    display('1');
                    display('1');
```

```
                    display('1');
}

else if(analog==57)
{
                    display('1');
                    display('1');
                    display('3');
}

else if(analog==58)
{
                    display('1');
                    display('1');
                    display('5');
}

else if(analog==59)
{
                    display('1');
                    display('1');
                    display('7');
}


else if(analog==60)
{
                    display('1');
                    display('1');
                    display('9');
}

else if(analog==61)
{
                    display('1');
                    display('2');
                    display('1');
}

else if(analog==62)
{
                    display('1');
                    display('2');
                    display('3');
}

else if(analog==63)
{
                    display('1');
                    display('2');
                    display('5');
}

else if(analog==64)
{
                    display('1');
                    display('2');
                    display('7');
}

else if(analog==65)
{
                    display('1');
                    display('2');
                    display('9');
}

else if(analog==66)
{
                    display('1');
                    display('3');
                    display('1');
```

```
}

else if(analog==67)
{
                        display('1');
                        display('3');
                        display('3');
}

else if(analog==68)
{
                        display('1');
                        display('3');
                        display('5');
}

else if(analog==69)
{
                        display('1');
                        display('3');
                        display('7');
                        display('D');
                        display('e');
                        display('g');
                        display('r');
                        display('e');
                        display('e');
}

else if(analog==70)
{
                        display('1');
                        display('3');
                        display('9');
}

else if(analog==71)
{
                        display('1');
                        display('4');
                        display('1');
}

else if(analog==72)
{
                        display('1');
                        display('4');
                        display('3');
}

else if(analog==73)
{
                        display('1');
                        display('4');
                        display('5');
}

else if(analog==74)
{
                        display('1');
                        display('4');
                        display('7');
}

else if(analog==75)
{
                        display('1');
                        display('5');
                        display('0');
}

else
```

```
{
                display('O');
                display('u');
                display('t');
                display(' ');
                display('o');
                display('f');
                display(' ');
                display('r');
                display('a');
                display('n');
                display('g');
                display('e');
                display(' ');
}

}
```

APPENDIX E

LM35 TEMPERATURE SENSOR DATASHEET

November 2000

National Semiconductor

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

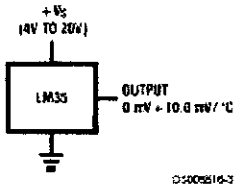
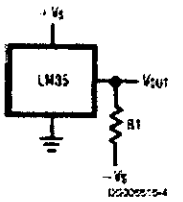


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S/50\ \mu A$
V_{OUT} = +1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

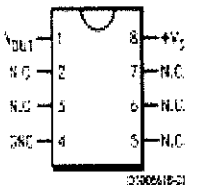
Connection Diagrams

TO-46
Metal Can Package*

*Case is connected to negative pin (GND)

Order Number LM35H, LM35A-H, LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

SO-8
Small Outline Molded Package



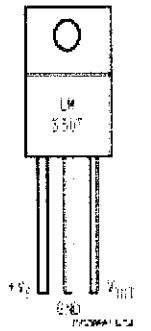
N.C. = No Connection

Top View
Order Number LM35DM
See NS Package Number M08A

TO-92
Plastic Package

Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

TO-220
Plastic Package*



*Tab is connected to the negative pin (GND).
Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+35V to −0.2V
Output Voltage	+6V to −1.0V
Output Current	10 mA
Storage Temp.,	
TO-46 Package,	−60°C to +180°C
TO-92 Package,	−60°C to +150°C
SO-8 Package,	−65°C to +150°C
TO-220 Package,	−65°C to +150°C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300°C
TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	−55°C to +150°C
LM35C, LM35CA	−40°C to +110°C
LM35D	0°C to +100°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	T_A=+25°C	±0.2	±0.5		±0.2	±0.5		°C
	T_A=−10°C	±0.3			±0.3		±1.0	°C
	T_A=T_{MAX}	±0.4	±1.0		±0.4	±1.0		°C
	T_A=T_{MIN}	±0.4	±1.0		±0.4		±1.5	°C
Nonlinearity (Note 8)	T_{MIN}≤T_A≤T_{MAX}	**±0.18**		**±0.35**	**±0.15**		**±0.3**	°C
Sensor Gain (Average Slope)	T_{MIN}≤T_A≤T_{MAX}	**+10.0**	**+9.9, +10.1**		**+10.0**		**+9.9, +10.1**	mV/°C
Load Regulation (Note 3) 0≤I_L≤1 mA	T_A=+25°C	±0.4	±1.0		±0.4	±1.0		mV/mA
	T_{MIN}≤T_A≤T_{MAX}	**+0.5**		**+3.0**	**+0.5**		**+3.0**	mV/mA
Line Regulation (Note 3)	T_A=+25°C	±0.01	±0.05		±0.01	±0.05		mV/V
	4V≤V_S≤30V	**±0.02**		**±0.1**	**±0.02**		**±0.1**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25°C	56	67		56	67		µA
	V_S=+5V	**105**		**131**	**91**		**114**	µA
	V_S=+30V, +25°C	56.2	80		56.2	80		µA
	V_S=+30V	**105.5**		**133**	**91.5**		**116**	µA
Change of Quiescent Current (Note 3)	4V≤V_S≤30V, +25°C	0.2	1.0		0.2	1.0		µA
	4V≤V_S≤30V	**0.5**		**2.0**	**0.5**		**2.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.5**	**+0.39**		**+0.5**	µA/°C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	T_J=T_{MAX}, for 1000 hours	±0.08			±0.08			°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM26, LM35C (Note 7)	T_A=+25°C	±0.4	±1.0		±0.4	±1.0		°C
	T_A=−10°C	±0.5			±0.5		±1.5	°C
	T_A=T_{MAX}	±0.8	±1.5		±0.8		±1.5	°C
	T_A=T_{MIN}	±0.8		±1.5	±0.8		±2.0	°C
Accuracy, LM35D (Note 7)	T_A=+25°C				±0.8	±1.5		°C
	T_A=T_{MAX}				±0.9		±2.0	°C
	T_A=T_{MIN}				±0.9		±2.0	°C
Nonlinearity (Note 8)	T_{MIN}≤T_A≤T_{MAX}	±0.5		±0.5	±0.2		±0.5	°C
Sensor Gain (Average Slope)	T_{MIN}≤T_A≤T_{MAX}	+10.0	+9.8, +10.2		+10.0		+9.8, +10.2	mV/°C
Load Regulation (Note 3) 0≤I_L≤1 mA	T_A=+25°C	±0.4	±2.0		±0.4	±2.0		mV/mA
	T_{MIN}≤T_A≤T_{MAX}	±0.5		±5.0	±0.5		±5.0	mV/mA
Line Regulation (Note 3)	T_A=+25°C	±0.01	±0.1		±0.01	±0.1		mV/V
	4V≤V_S≤30V	±0.02		±0.2	±0.02		±0.2	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25°C	56	80		56	80		µA
	V_S=+5V	105		153	91		138	µA
	V_S=+30V, +25°C	56.2	82		56.2	82		µA
	V_S=+30V	105.5		161	91.5		141	µA
Change of Quiescent Current (Note 3)	4V≤V_S≤30V, −25°C	0.2	2.0		0.2	2.0		µA
	4V≤V_S≤30V	0.5		3.0	0.5		3.0	µA
Temperature Coefficient of Quiescent Current		+0.39		+0.7	+0.39		+0.7	µA/°C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	T_J=T_{MAX} for 1000 hours	±0.08			±0.08			°C

Note 1: Unless otherwise noted, these specifications apply: −55°C≤T_J≤+150°C for the LM35 and LM35A; −40°C≤T_J≤+110°C for the LM35C and LM35CA; and 0°C≤T_J≤+100°C for the LM35D. V_S=+5Vdc and I_{LOAD}=50 µA, in the circuit of *Figure 2*. These specifications also apply from +2°C to T_{MAX} in the circuit of *Figure 1*. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400°C/W, junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-220 package is 90°C/W junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and 10mV/°C times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in °C).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of *Figure 1*.

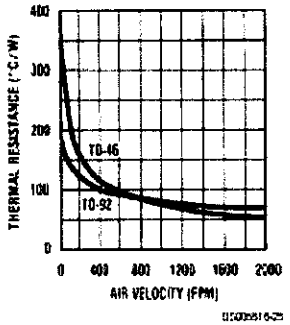
Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

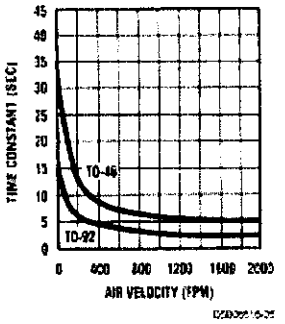
Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

Typical Performance Characteristics

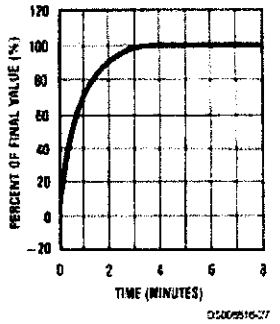
Thermal Resistance Junction to Air



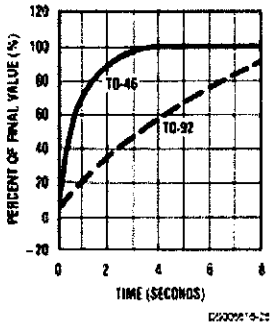
Thermal Time Constant



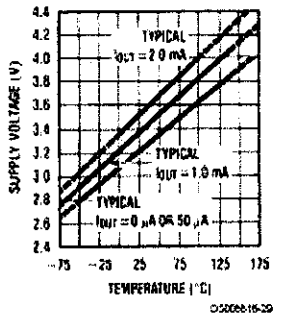
Thermal Response in Still Air



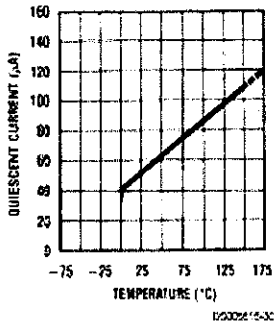
Thermal Response in Stirred Oil Bath



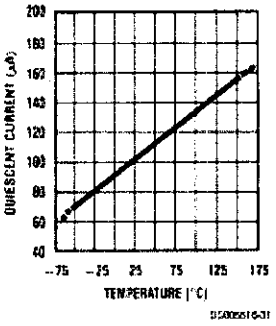
Minimum Supply Voltage vs. Temperature



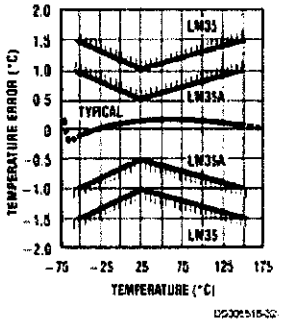
Quiescent Current vs. Temperature (In Circuit of *Figure 1*.)



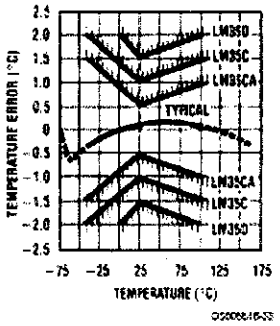
Quiescent Current vs. Temperature (In Circuit of *Figure 2*.)



Accuracy vs. Temperature (Guaranteed)

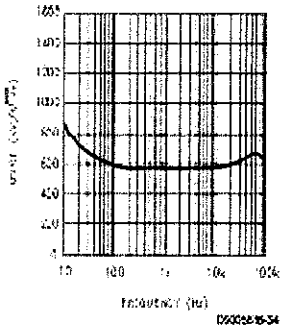


Accuracy vs. Temperature (Guaranteed)

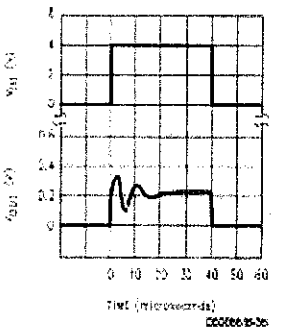


Typical Performance Characteristics (Continued)

Noise Voltage



Start-Up Response



Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V− terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, θ_{JA})

	TO-46, no heat sink	TO-46*, small heat fin	TO-92, no heat sink	TO-92**, small heat fin	SO-8 no heat sink	SO-8* small heat fin	TO-220 no heat sink
Still air	400°C/W	100°C/W	160°C/W	140°C/W	220°C/W	110°C/W	90°C/W
Moving air	100°C/W	40°C/W	90°C/W	70°C/W	105°C/W	90°C/W	26°C/W
Still oil	100°C/W	40°C/W	90°C/W	70°C/W			
Stirred oil	50°C/W	30°C/W	45°C/W	40°C/W			
(Clamped to metal, infinite heat sink)		(24°C/W)				(55°C/W)	

*Wakefield type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

**TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

Typical Applications

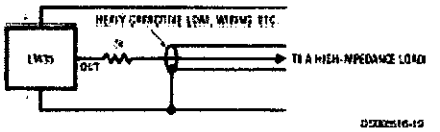


FIGURE 3. LM35 with Decoupling from Capacitive Load

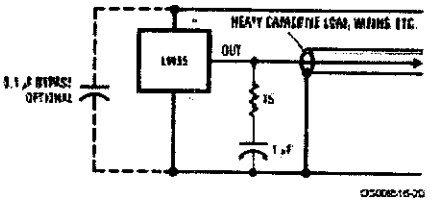


FIGURE 4. LM35 with R-C Damper

CAPACITIVE LOADS

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pf without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor; see *Figure 3*. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see *Figure 4*.

When the LM35 is applied with a 200Ω load resistor as shown in *Figure 5*, *Figure 6* or *Figure 8* it is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients etc, as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from V_{IN} to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 μF from output to ground are often useful. These are shown in *Figure 13*, *Figure 14*, and *Figure 16*.

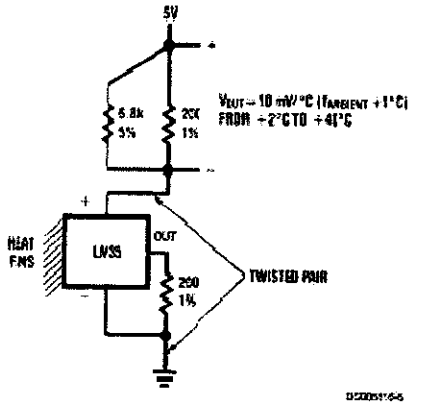


FIGURE 5. Two-Wire Remote Temperature Sensor (Grounded Sensor)

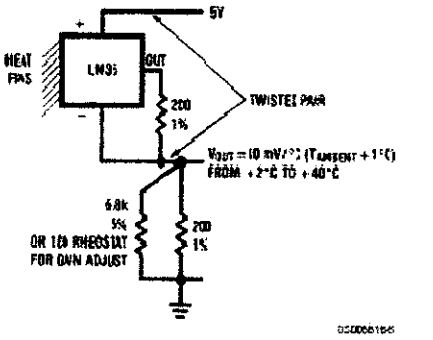


FIGURE 6. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

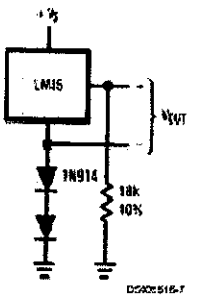


FIGURE 7. Temperature Sensor, Single Supply, −55° to +150°C

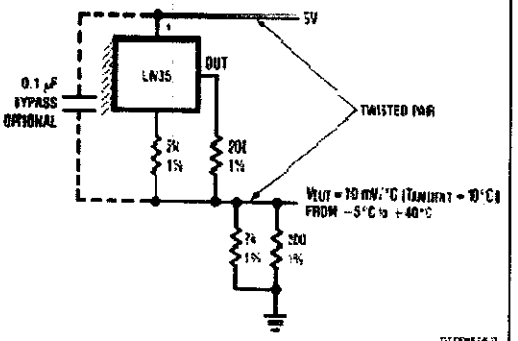


FIGURE 8. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

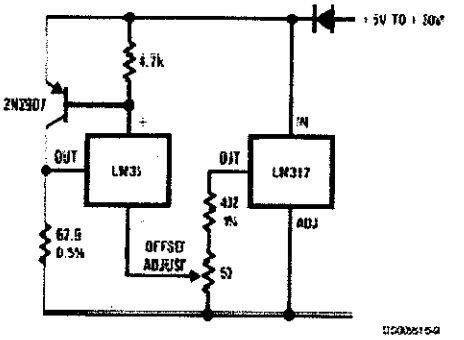


FIGURE 9. 4-To-20 mA Current Source (0°C to +100°C)

Typical Applications (Continued)

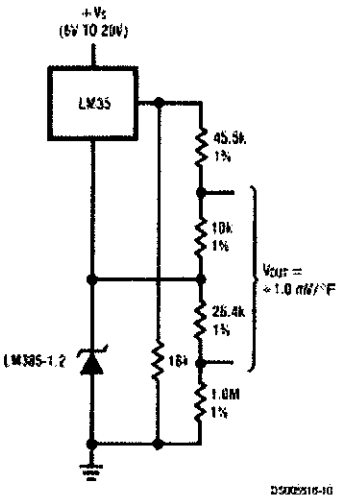


FIGURE 10. Fahrenheit Thermometer

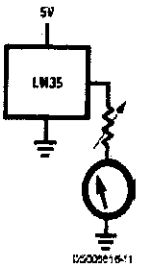


FIGURE 11. Centigrade Thermometer (Analog Meter)

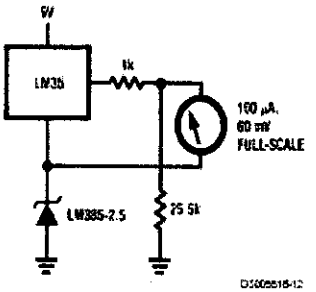


FIGURE 12. Fahrenheit ThermometerExpanded Scale Thermometer
(50° to 80° Fahrenheit, for Example Shown)

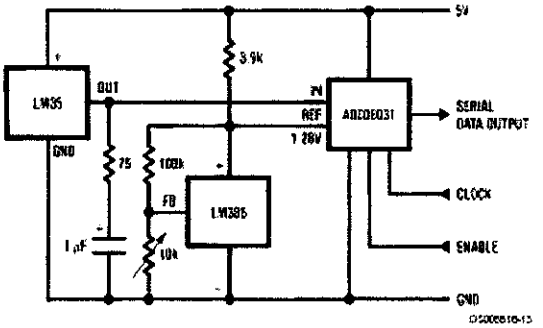


FIGURE 13. Temperature To Digital Converter (Serial Output) (+128°C Full Scale)

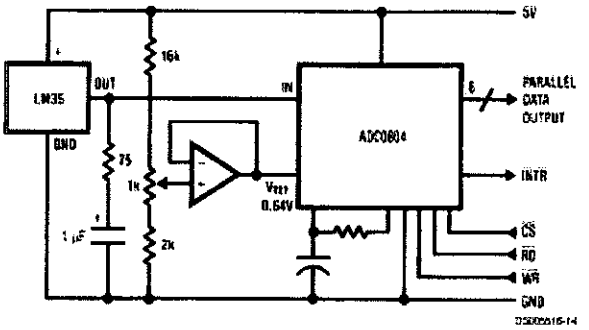
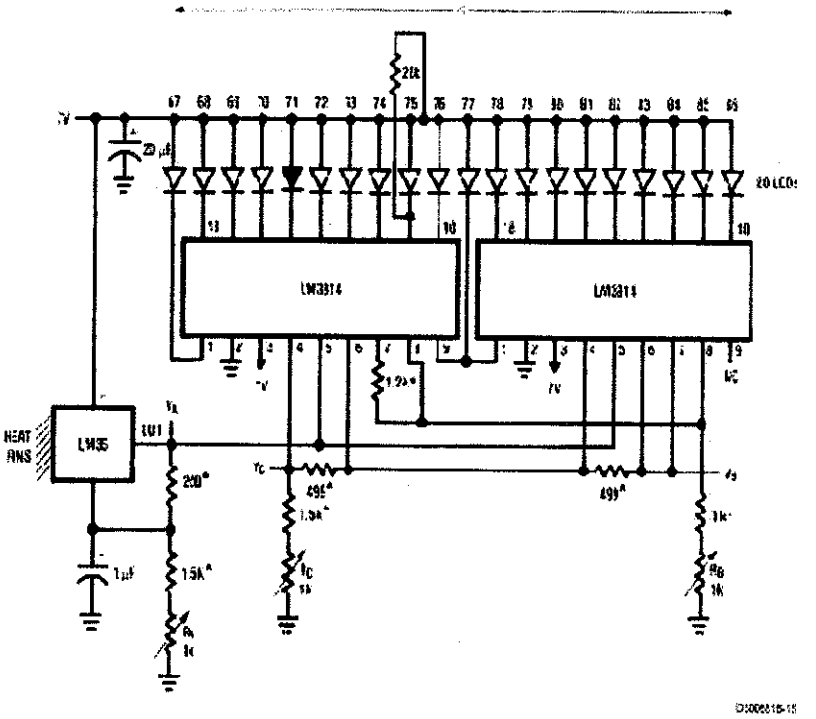


FIGURE 14. Temperature To Digital Converter (Parallel TRI-STATE™ Outputs for Standard Data Bus to µP Interface) (128°C Full Scale)

Typical Applications (Continued)



*=1% or 2% film resistor
Trim R_B for V_O=3.075V
Trim R_C for V_C=1.955V
Trim R_A for V_A=0.076V × 100mV/°C × $T_{ambient}$
Example, V_A=2.275V at 22°C

FIGURE 15. Bar-Graph Temperature Display (Dot Mode)

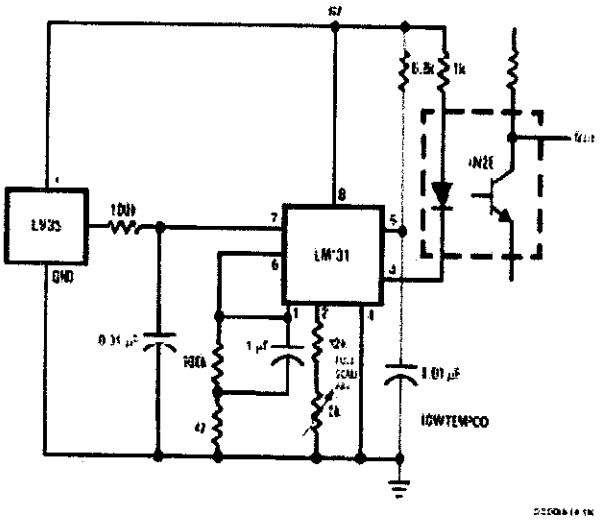
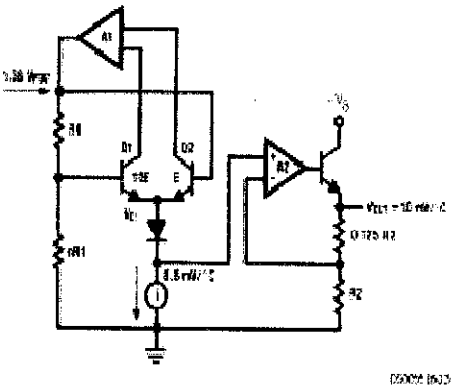
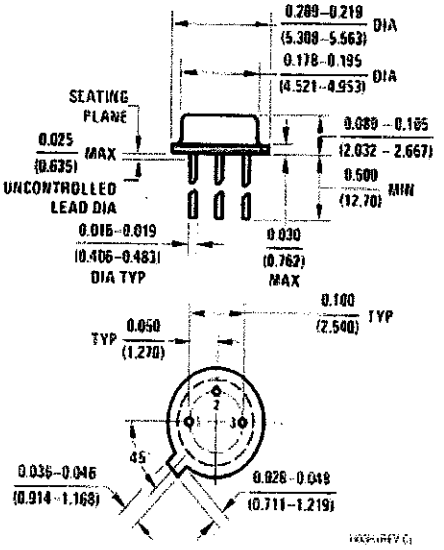


FIGURE 16. LM35 With Voltage-To-Frequency Converter And Isolated Output (2°C to +150°C; 20 Hz to 1500 Hz)

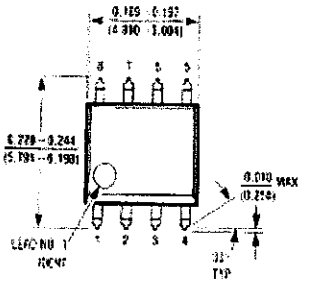
Block Diagram

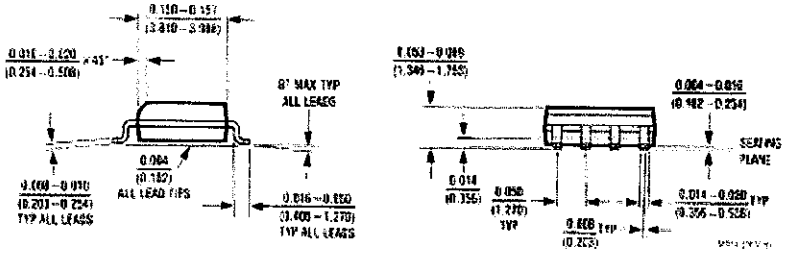


Physical Dimensions inches (millimeters) unless otherwise noted



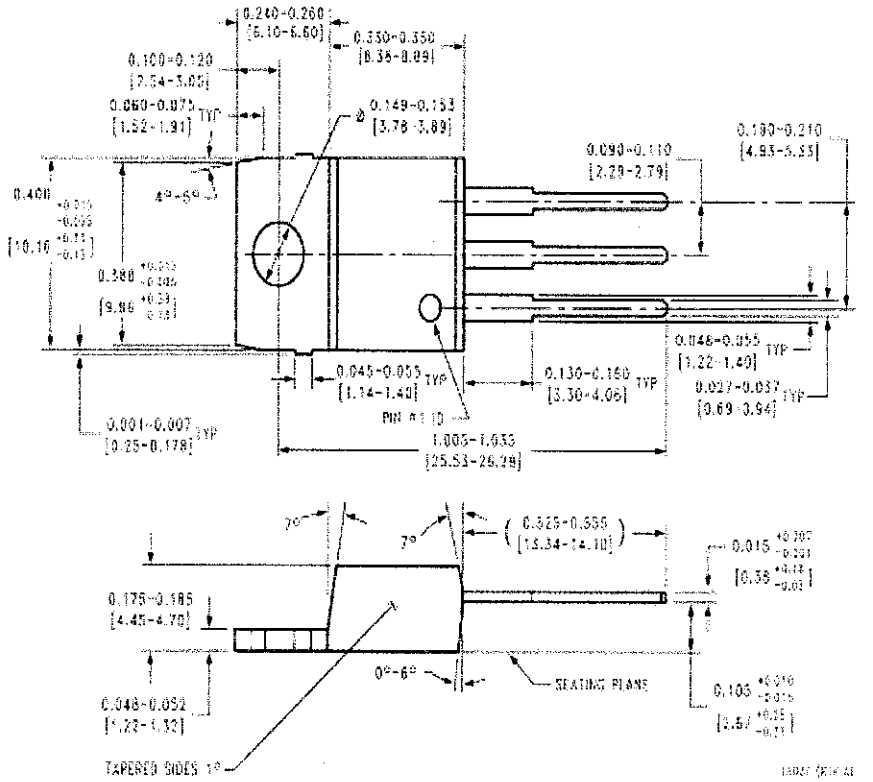
TO-46 Metal Can Package (H)
Order Number LM35H, LM35AH, LM35CH, LM35CAH, or LM35DH
NS Package Number H03H





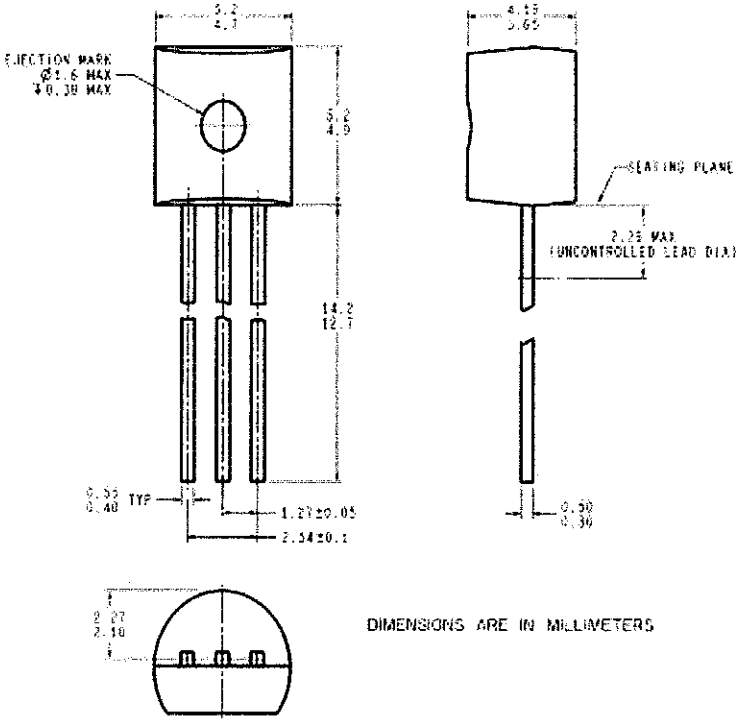
SO-8 Molded Small Outline Package (M)
Order Number LM35DM
NS Package Number M08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Power Package TO-220 (T)
Order Number LM35DT
NS Package Number TA03F

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



TO-92 Plastic Package (Z)
Order Number LM35CZ, LM35CAZ or LM35DZ
NS Package Number Z03A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

APPENDIX F

SIGNAL CONDITION TEST WITH RESPECT TO DISTANCE BETWEEN EXTERNAL DEVICES AND COMPUTER/LAPTOP BLUETOOTH

Table 5: Signal condition and effect to the data receives

No	Meter (for every 3 feet count)	Signal Strength	Data Condition
1	1.0	Excelent	- Signal receive without any error or delay
2	2.0	Excelent	- Signal receive without any error or delay
3	3.0	Excelent	- Signal receive without any error or delay
4	4.0	Excelent	- Signal receive without any error or delay
5	5.0	Excelent	- Signal receive without any error or delay
6	6.0	Excelent	- Signal receive without any error but sometimes have a transmission delay 0.5 to 1.0 second
7	7.0	Excelent	- Signal receive without any error but sometimes have a transmission delay 0.5 to 1.0 second
8	8.0	Excelent	- Signal receive without any error but sometimes have a transmission delay 0.5 to 1.0 second
9	9.0	Excelent	- Signal receive without any error but sometimes have a transmission delay 0.5 to 1.0 second
10	10.0	Good	- Signal receive sometimes with noise and have a transmission delay 0.5 to 1.0 second
11	11.0	Good	- Signal receive sometimes with noise and have a transmission delay 0.5 to 1.0 second
12	12.0	Good	- Signal receive sometimes with noise and have a

			transmission delay 0.5 to 1.0 second
13	13.0	Good	- Signal receive sometimes with noise and have a transmission delay 0.5 to 1.0 second
14	14.0	Good	- Signal receive sometimes with noise and have a transmission delay 0.5 to 1.0 second
15	15.0	Good	- Signal receive sometimes with noise and have a transmission delay 0.5 to 1.0 second
16	16.0	Good	- Signal receive sometimes with noise and have a transmission delay 1.5 to 2.0 second
17	17.0	Good	- Signal receive sometimes with noise and have a transmission delay 1.5 to 2.0 second
18	18.0	Good	- Signal receive sometimes with noise and have a transmission delay 1.5 to 2.0 second
19	19.0	Good	- Signal receive sometimes with noise and have a transmission delay 1.5 to 2.0 second
20	20.0	Good	- Signal receive sometimes with noise and have a transmission delay 1.5 to 2.0 second
21	21.0	Poor	- Signal receive with noise and have a transmission delay 1.5 to 2.0 second
22	22.0	Poor	- Signal receive with noise and have a transmission delay 1.5 to 2.0 second
23	23.0	Poor	- Signal receive with noise and have a transmission delay 1.5 to 2.0 second
24	24.0	Poor	- Signal receive with noise and have a transmission delay 1.5 to 2.0 second
25	25.0	Poor	- Signal receive with noise

			and have a transmission delay 1.5 to 2.0 second
26	26.0	Poor	- Signal receive with noise and have a transmission delay 1.5 to 2.0 second
27	27.0	Poor	- Signal receive with noise and have a transmission delay 1.5 to 2.0 second
28	28.0	Poor	- Signal receive with noise and have a transmission delay 1.5 to 2.0 second
29	29.0	Bad	- Signal receive with many error and have a transmission delay 3.5 to 5.0 second
30	30.0	Bad	- Signal receive with many error and have a transmission delay 3.5 to 5.0 second
31	31.0	Bad	- Signal receive with many error and have a transmission delay 3.5 to 5.0 second
32	32.0	Bad	- Signal receive with many error and have a transmission delay 3.5 to 5.0 second
33	33.0	Bad	- Signal receive with many error and have a transmission delay 3.5 to 5.0 second
34	34.0	No signal	No more data receive: - Need to reset connection

APPENDIX G

TEMPERATURE SENSOR ANALOG TO DIGITAL CONVERSION CALLIBRATION

Table 6: Sensor and PIC Calibration:

PIC ADC Value	Temperature (degree)
0	0
1	1
2	3
3	5
4	7
5	9
6	11
7	13
8	15
9	17
10	19
11	21
12	23
13	25
14	27
15	29
16	31
17	33
18	35
19	37
20	39
21	41
22	43
23	45
24	47
25	49
26	51
27	53
28	55
29	57
30	59
31	61
32	63
33	65
34	67
35	69
36	71
37	73
38	75
39	77
40	79
41	81
42	83
43	85
44	87
45	89
46	91
47	93
48	95
49	97
50	99
51	101
52	103
53	105
54	107
55	109
56	111
57	113
58	115
59	117
60	119
61	121
62	123
63	125
64	127
65	129
66	131
67	133
68	135
69	137
70	139
71	141
72	143
73	145
74	147
75	149

PIC ADC resolution bits = 8bits since $2^8 = 256$. The values can represent the ranges from 0 to