A REPORT

ON

DATA AQUISITION THROUGH WIRELESS SENSOR NETWORKS

By

Akhilesh Agrawal 2010B3A3493H

Rishav Rej 2010AAPS064H

UNDER THE SUPERVISION OF

Dr.KCS Murti, Computer Science Department

At

Birla Institute of Technology and Science, Pilani Hyderabad Campus

A REPORT

ON

DATA AQUISITION THROUGH WIRELESS SENSOR NETWORKS

Submitted in partial fulfilment of the

Study Oriented Project BITS C331

By

Akhilesh Agrawal 2010B3A3493H

Rishav Rej 2010AAPS064H

UNDER THE SUPERVISION OF

Dr.KCS Murti, Computer Science Department

At

Birla Institute of Technology and Science, Pilani Hyderabad Campus

CERTIFICATE

This is to certify that the report entitled, -- ---- -- and submitted by -------------------------------- ID No. ------- -------- --- in partial fulfillment of the requirements of BITS C331 Computer Projects embodies the work done by him/her under my supervision.

Signature of the supervisor

Name:

Designation:

Date:

Acknowledgements

I would like to thank Prof. V.S.Rao, Director, BITS, Pilani Hyderabad campus who has been our inspiration. I thank Dr.KCS Murti our Instructor for his constant guidance for the fulfilment of the course and giving valuable suggestions explaining the importance of Computer Oriented Project in our higher studies .We are really indebted to him. The extensive discussions and presentations on the topic of project was indeed exemplary. It did help us impart great knowledge from it. We look forward to do more projects under your esteemed guidance.

(Rishav Rej) (Akhilesh Agrawal)

Abstract

This report deals with the design, development and implementation of a temperature Sensor using ZigBee module interfaced with the PIC microcontroller.

The main aim of the work undertaken in this report is to sense the temperature and to display the result on the CENTRAL SERVER using the ZigBee technology. ZigBee operates in the industrial, scientific and medical (ISM) radio bands; 868 MHz in Europe,915MHz in the USA and 2.4 GHz in most jurisdictions worldwide. The technology is intended to be simpler and cheaper than other WPANs such as Bluetooth. The most capable ZigBee node type is said to require only about 10 % of the software of a typical Bluetooth or Wireless Internet node, while the simplest nodes are about 2%. However, actual code sizes are much higher, more like 50 % of the Bluetooth code size. In this work undertaken in the design & development of the temperature sensor, it senses the temperature and after amplification is then fed to the micro controller, this is then connected to the ZigBee module, which transmits the data and at the other end the ZigBee reads the data and displays on to the CENTRAL SERVER. The software developed is highly accurate and works at a very high speed. The method developed shows the effectiveness of the scheme employed**.**

Keywords—Zigbee, Microcontroller, PIC, Transmitter, Receiver, Synchronous, Blue tooth, Communication.

Table of Contents:

I. Introduction

This section gives a brief introduction about the work, which describes all the components namely Zigbee, temperature sensor, PIC18F4550. Zigbee is a wireless network protocol specifically designed for low data rate sensors and control networks. Also, a brief literature survey of the work related to the research topic is also presented in the following paragraphs Zigbee is a consortium of software, hardware and services companies that have developed a common standard for wireless, networking of sensors and controllers. While other wireless standards are concerned with exchanging large amounts of data, Zigbee is for devices that have smaller throughout needs.

The other driving factors are low cost, high reliability, high security, low battery usage, simplicity and interoperability with other Zigbee devices. Compared to other wireless protocols, Zigbee wireless protocol offers low complexity. It also offers three frequency bands of operation along with a number of network configurations and optional security capability. It requires a supply voltage in the range of 2.8V to 3.3V. Hence, we design a power supply, which converts 230V to 3.3V. Here, we use the whip antenna to transmit the temperature sensed by LM35 temperature sensor to the receiving section. The LM35 series are precision integrated circuit temperature sensor whose output voltage is linearly proportional to the Celsius temperature.

The LM35 thus has an advantage over the 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 microcontroller used here is PIC18F4550. It belongs to a class of eight-bit microcontrollers of RISC architecture & a Program Memory (FLASH) for storing a written program. Since the memory is made in FLASH technology, it can be programmed and cleared more than once & makes this microcontroller suitable for device development. It has inbuilt ADC and USART. In the receiving section, the temperature is displayed on 16 u 2 backlit CENTRAL SERVER. If the temperature displayed exceeds 40 degree Celsius, then the buzzer goes on.

The report is organized in the following sequence:

Introduction about the work undertaken in this report and the relevant literatures were presented in the previous paragraphs. Introduction about the microcontroller & its design is considered in the section 2.Section 3 depicts about the background literature about the temperature sensor. The transmitter & receiver part is presented in section 4. Section 5 describes about the ZigBee concepts & its design. The design and development of the temperature sensor is presented in section 6. Working principle is presented in the next section. This is followed by the conclusions in section 8, followed by the references.

II. DESCRIPTION ABOUT THE MICROCONTROLLER

This section gives a brief idea about the PIC microcontroller, its advantages over microprocessors, its core features, block diagram, pin diagram and its description.

A. INTRODUCTION

Circumstances that we find ourselves today in the field of microcontrollers had their beginnings in the development of technology of integrated circuits. This development had made it possible to store hundreds of thousands of transistors into one chip. That was a prerequisite for production of microcontrollers, and adding external peripherals such as memory, input-output lines, timers and other made the first computers. Further increasing of the volume of the package resulted in creation of integrated circuits. These integrated circuits contained both processor and peripherals. That is how the first chip containing a microcomputer, or what would later be known as a microcontroller came out.

B. MICROCONTROLLER VERSUS MICROPROCESSORS

Microcontroller differs from a microprocessor in many ways. First and the most important is its functionality. In order for a microprocessor to be used, other components such as memory, or components for receiving and sending data must be added to it. In sort that means that microprocessor is the very heart of the computer. On the other hand, microcontroller is designed to be all of that in one. No other external components are needed for its application because all necessary peripherals are already built in to it. Thus, we save the time and space needed to construct devices.

C. PIC18F4550

It belongs to a class of eight bit microcontrollers of RISC Architecture.

1) Program Memory (FLASH)

This concept is used for storing a written program. Since memory is made use of in FLASH technology, it can be programmed and cleared more than once, it makes this microcontroller suitable for device development.

2) EEPROM

It is that device wherein the data memory needs to be saved when there is no supply. It is usually used for storing important data that must not be lost if supply suddenly stops. For instance, one such data is an assigned temperature in temperature regulators. If during a loss of supply this data is lost, we would have to make the adjustment once again upon return of supply. Thus our device loses on self-reliance.

3) RAM

Data memory used by a program during its execution. In RAM are stored all inter-results on temporary data that are not crucial to running a device during a loss of supply.

4) PORTA, PORTB AND PORTC

These are physical connections between the microcontroller and the outside world. PORTA has five pins, PORTB and PORTC has eight pins.

5) FREE TIMER

It is an eight-bit register inside a microcontroller that works independently of the program. On every fourth clock of the oscillator, it increments its value until it reaches the maximum, and then its starts counting over again from zero. As we know the exact timing between each two increments of the timer contents, timer can be used for measuring the time, which is very useful with some devices.

6) CENTERAL PROCESSING UNIT

It has a role of connective elements between other blocks in the microcontroller. It coordinates that work of other blocks and executes the user program.

D. CISC AND RISC

It has already been said that PIC18F4550 has RISC architecture. This term is often found in computer literature. Harvard architecture is a newer concept than von-Neumann. It rose out of the need to speed up the work of a microcontroller. In Harvard architecture, data bus and address bus are separate. Thus, a greater flow of data is possible through the central processing unit, and of course, a greater speed of work. Microcontrollers with Harvard architecture are also called “RISC microcontrollers”.RISC stands for reduced instruction set computer. Microcontrollers with von-Neumann's architecture are called as the "CISC microcontrollers". The title CISC stands for 'complex instruction set computer'. Since PIC18F4550 is a RISC microcontroller that means that it has a reduced set of instructions, more precisely 35 instructions.

E. MICROCONTROLLER CORE FEATURES

- High performance RISC CPU
- Only 35 Instructions to learn
- All Single cycle instructions except for program branches, which are two cycles
- Operating speed : DC-20MHz clock input
- DC-200ns instruction cycle
- Up to 8K u 14 words of FLASH program memory
- Up to 368 u 8 bytes of data memory (RAM)
- Up to 256 u 8 bytes of EEPROM data memory
- Power-on reset (POR)
- Power saving SLEEP mode
- Low-power, high-speed CMOS FLASH / EEPROM Technology
- Wide operating voltage range: 2V TO 5.5 V
- Low power consumption:
- < 2mA typical @ 5V, 4MHz, 20PA typical @ 3V, 32 kHz, < 1PA typical standby current

F. PIN DIAGRAM

The pin diagram of PIC18F4550 is as shown in the Fig. 1.

G. DEVICE OVERVIEW

PIC18F4550 device comes in 40-pin package. This does not have a parallel slave port implemented.

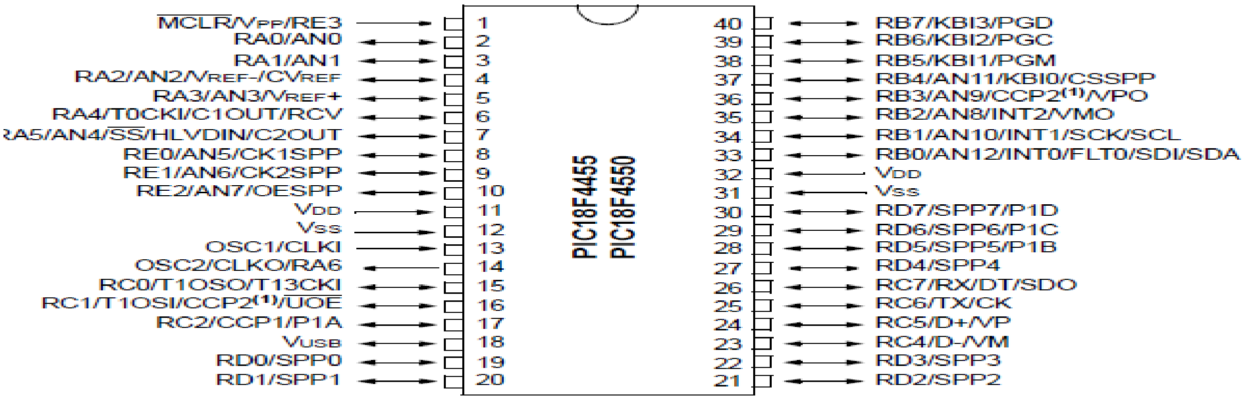
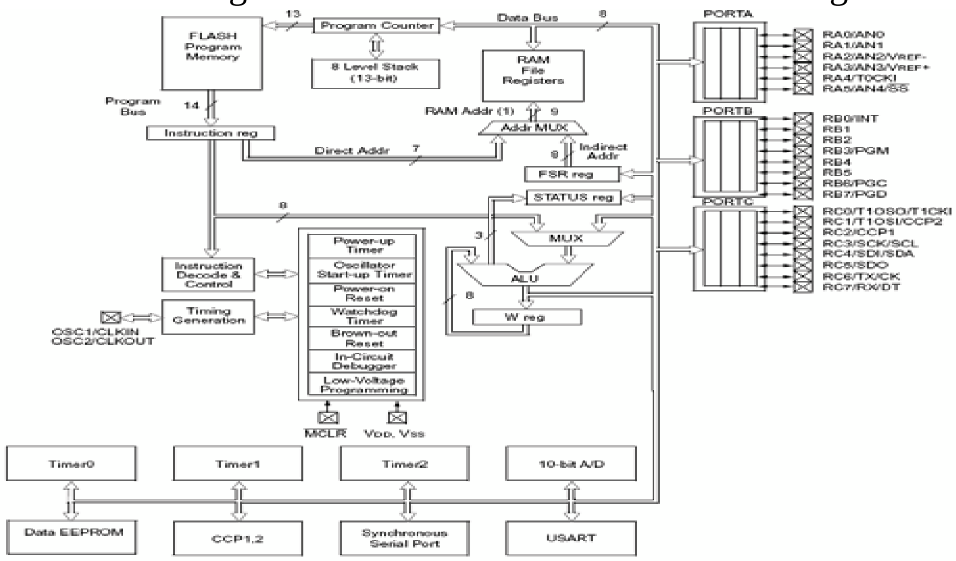


Fig. 1 Pin details of PIC PC

H. PIC18F4550 BLOCK DIAGRAM

The block diagram of the PIC18F4550 is shown in Fig. 2.



Device	Program FLASH	Data memory	Data EEPROM
PIC18F4550	4K	192 Bytes	128 Bytes

Fig. 2 Block diagram of the PIC18F4550 microcontroller

I. STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory. The STATUS register can be the destination for any instructions, as with any other register. If the STATUS register is the destination for any instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable, therefore, the result of an instruction with the STATUS register as destination may be different intended. For example, CLRF STATUS will clear upper three bits and set the Z bit. This leaves the STATUS register as 000u u1uu (where u = uncharged). It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to after the STATUS register, because these instructions do not affect the Z, C or DC bits from the STATUS register. For other "instructions set summary."

J. OPTION_REG REGISTER

The OPTION_REG register is a readable and writable register, which contains various control bits to configure the TMRO pre-scaler / WDT post-scaler (single, assignable register known also as pre-scaler), the external INT interrupt TMRO and the week pull-ups on PORTB. The detailed description about each bit of status register and option register is also studied prior to the design.

K. MEMORY ORGANIZATION

There are three memory blocks in PIC18F4550. The program memory and data memory has separate buses so that concurrent access can occur and is detailed in this section.

L. Program memory organization

PIC18F4550 have program counter capable of addressing an 8K u 14 program memory space. The reset vector is at 0000h and the interrupt vector is at 0004h.

M. Data memory organization

The Data Memory is partitioned in to multiple banks, which contain the general-purpose register and special function register bits RP1 (STATUS <6>) and RP0 (STATUS <5>) are the bank select bits as shown in the table 1.

TABLE 1 REGISTER BANKS

RP1 :	Bank
00	0
01	1
10	2
11	3

N. PROGRAM MEMORY MAP AND STACK

The program memory map and stack organization with its addresses was also studied in brief prior to the design of the temperature sensor. It consisted of 8 stack levels. The on chip program memory was divided into pages. The program memory address ranges from 0000h to 1FFFh. When a call instruction is executed, the address of the next instruction will be stored into the stack memory. The stack works on first in first out manner. After the return instruction is executed, the address stored in the stack is retrieved and loaded back into the program memory.

III. BACKGROUND LITERATURE SURVEY ABOUT THE TEMPERATURE SENSOR

This section gives a brief information about the temperature sensor LM35, its connection diagram and its features.

A. GENERAL DESCRIPTION

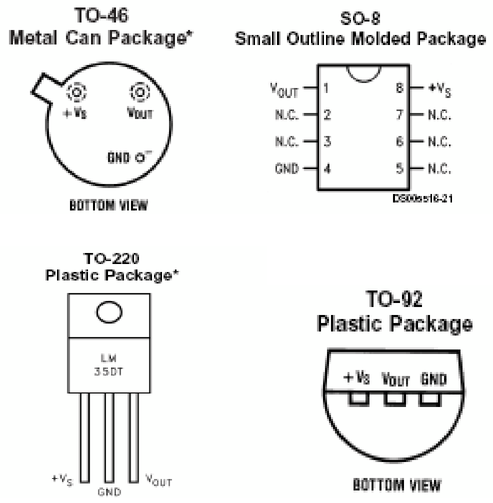
The LM35 series IC's are precision integrated circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 thus has an advantage over the linear temperature sensors calibrated in Kelvin as the as the user is not required to subtract a large constant voltage from its output to obtain convent centigrade scaling. The LM35 does not require any external calibration to provide typical accuracies of + or – ¼qC at room temperature and + or – ¾qC over a full –55 to +150qC temperature range.
The LM35's low output impedance linear output and
Precise inherent calibration make interfacing to readout or control circulatory especially easy. It can be used with single power supplies, or with plus and minus supply; it has very low self-heating less than 0.1 degrees Celsius in still air. The LM35 is rated to operate over a –55q to +150qC temperature range, while the LM35 is rated for a –40q to +110qC range. The LM35 series is available packaged in hermetic TO-46 transistor packages. The LM35D is also available in an 8 lead surface mount small outline package and a plastic TO-220 package.

B. FEATURES

Calibrated directly in Celsius (centigrade). Linear + 10.0 m V / qC scale factor.
0.5q C accuracy guarantee able (at +25q C). Rated for full -55q to +150q C range. Operates from 4 to 30 volts.
Low self-heating, 0.08qC in still air.
Low impedance output, 0.1 : for 1 mA load.

C. CONNECTION DIAGRAM

The connection diagram for LM35 packages is shown in Fig. 3. Here we are using TO-220 plastic package temperature sensor. It has three leads namely +Vs, ground and V_{out}



IV. UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER AND TRAMSMITTER

This section gives a brief description of the USART and the registers used for its operation at the transmitter and receiver section. The universal synchronous asynchronous receiver and transmitter (USART) module is one of the two serial input or output modules. USART is also known as a serial communications interface or SCI. The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices such as analog to digital or digital to analog integrated circuits. The USART used here is inbuilt in the PIC microcontroller. The USART can be configured in the following modes [4].

Asynchronous (full duplex) Synchronous-master (half duplex)
Synchronous - slave (half duplex)

The registers used in the operation of USART are

- TXSTA (TRANSMIT CONTROL AND STATUS REGISTER)
- RCSTA (RECEIVE CONTROL AND STATUS REGISTER)
- The TXSTA is used to control the transmission of data.
- The RCSTA is used to control the reception of data.

A. REGISTERS

TXSTA transmit status and control register (Add 98h) was also studied from the data manuals. RCSTA receive status and control register (Add 18h) was also studied from the data manuals.

B. USART BAUD RATE GENERATOR (BRG)

Baud rate supports both asynchronous and synchronous moods of the USAIT.The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode, bit BRGH (TXSTA <2>) also controls the baud rate. In asynchronous mode, bit BRGH is ignored. Given the desired baud rate and FOSC, the nearest integer value for the SPBRG registers can be calculated using the formula shown in the table. From this, the error is baud rate can be determined. The table 2 gives the formula to calculate the Baud rate.

C. USART ASYNCHRONOUS MODE

In this mode, the USART uses standard non-return to zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is eight bits. The USART transmits and receives the LSB first. The USART'S transmitter and receiver are functionally independent; but use the same data format and baud rate. Asynchronous mode is stopped during SLEEP. Asynchronous mode is selected by clearing bit synchronous (TXSTA<4>).

D. USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is as shown in Fig. 4. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in soft wear. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the stop bit is transmitted the TSR is loaded with new data from the TXREG register (if available).Note that the TSR register is not mapped in data memory so it is not available to the user. Flag bit TXIF is set when enable bit TXEN is set.TXIF is cleared by loading the TXREG. Transmission is enabled by setting enable bit TXEN (TXSTA<5>).The actual transmission will not occur until TXREG register has been loaded with data and the baud rate generator (BRG) has produced a shift clock. The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG.

A back-to-back transfer is thus possible. Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RC6/ TX / CK pin will revert to hi-impedance. In order to select 9- bit transmission, transmit bit TX9 (TXSTA <6>) should be set and the ninth bit should be written before writing the 8-bit data to the TXREG register. This is because of a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit may be loaded in the TSR register.

E. STEPS FOLLOWED IN ASYNCHRONOUS TRANSMISSION

- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
- Enable the asynchronous serial port by clearing bit synchronous and setting bit SPEN.
- If interrupts are desired, then set enable bit TXIE.
- If 9-bit transmission is desired, then set transmit bit TX9.
- Enable the transmission by setting bit TXEN, which will also set bit TXIF.
- If 9-bit transmission is selected the ninth bit should be loaded in bit TX9D.
- Load data to the TXREG register (starts transmission)

F. USART ASYNCHRONOUS RECEIVER

The receiver block diagram is as shown in the Fig. 5. The data is received on the RC7 / RX / DT pin and drives the data recovery block. The data recovery block is actually high speed shifter operating at u 16 times the baud rate, where as the main receive serial shifter operates at the bit rate or at FOSC. Once asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA <4>). The heart of the receiver is the receive (serial) the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set.

The actual interrupt can be enabled / disabled by

Setting/clearing enable bit RCIE (PIE1<5>) flag bit is a read only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two deep FIFO). It is possible for two bytes of data to be received and transfer RCREG FIFO and the third byte to begin shifting RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full, the overrun error bit OERR (RCSTA <1>) will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register is inhibited, so it is essential to clear error bit OERR if it is set. Framing error bit

FERR (RCSTA <2>) is set if a stop bit is detected as clear. Bit FERR and the ninth receiver bit or buffer the same way as the receive data. Reading the RCREG will load bit RX9D and FERR with new values therefore it is essential for the user to read the RCSTA register before reading RCREG register in order not to lose the old FERR and RX9D information.

G. STEPS FOLLOWED IN ASYNCHRONOUS RECEPTION

- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired set bit BRGH.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If interrupts are desired, then set enable bit RCIE.
- If 9-bit reception is desired, then set bit RX9.
- Enable the reception by setting bit CREN.
- Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
- Read the RCSTA register to get ninth bit (if enable) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG Register.
- If any error occurred, clear the error by clearing enable bit CREN.

V.CONCEPTS OF ZIGBEE DESIGN

This section gives the information about Zigbee, which is the major component used, its characteristics and its working, differences between Blue tooth and Zigbee, different topologies used to form the network and the applications of Zigbee.

A. INTRODUCTION TO ZIGBEE

Fig. 6 ZIGBEE chip

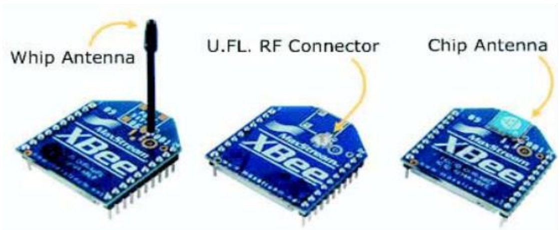
Zigbee is a wireless network protocol specifically designed for low data rate sensors and control networks as shown in Fig. 6. Zigbee is a consortium of software, hardware and services companies that have developed a common standard for wireless, networking of sensors and controllers. While other wireless standards are concerned with exchanging large amounts of data, Zigbee is for devices that have smaller throughout needs. The other driving factors are low cost, high reliability, high security, low battery usage, simplicity and interoperability with other Zigbee devices.

Compared to other wireless protocol that Zigbee wireless protocol offers low complexity. It also offers three frequency bands of operation along with a number of network configurations and optional security capability. In health care, Zigbee can be used for patient monitoring process control, assuring compliance with environmental standards and energy management. Used correctly, Zigbee enabled devices can give a warning before a breakdown occurs so that repairs can be made in the most cost effective manner. They will be used for controlling our home entertainment systems, lights, garage door openers, alarms, panic buttons and many other uses.

B. DIFFERENCE BETWEEN BLUETOOTH AND ZIGBEE

Zigbee looks rather like blue tooth but is simpler, has a lower data rate and spends most of its time in snoozing. This characteristic means that a node on a Zigbee network should be able to run for six months to two years on just two AA batteries. The operational range for it is 10 to 75 meters compared to 10 meters for blue tooth (without a power amplifier). Zigbee sits below blue tooth in terms of data rate. The data rate of Zigbee is 250 kbps at 2.4 GHz 40 kbps at 915 MHz and 20 kbps at 868 MHz where as that of blue tooth is 1 Mbps.

Zigbee uses a basic master slave configuration suited to static star networks of many infrequently used devices that talk via small data packets. It allows up to 254 nodes. Blue tooth protocol is more complex since it is geared towards handling voice, images and file transfers in ad hoc networks, blue tooth devices can support scatter nets of multiple smaller non synchronized networks. It only allows up to 8 slave nodes in a basic master slave Pico net setup. When Zigbee node is powered down, it can wake up and get a packet in around 15 milliseconds where as a blue tooth device would take around 3 seconds to wake up and respond.



C. HOW DOES ZIGBEE WORK?

Zigbee hardware typically consist of an eight bit microcontroller combined with a miniature transceiver a small amount (example 32 KB) of flash memory and RAM. Most of the Zigbee stack is provided in ASIC. Zigbee operates with ISM 2.4 GHz frequency band and is pin for pin compatible with maxstream's Zigbee product.

There are three radio frequencies used for Zigbee radio frequency communications 2.4 GHz with 16 channels and a data rate of 250 kbps for worldwide coverage, 868 MHz with a single channel and a data rate of 20 kbps in Europe and

915
MHz with 10 channels and a data rate of 40 kbps in America. For comparison even at 250 kbps the data throughput is only about one tenth that of blue tooth.

Another wireless networking solution but more than sufficient for monitoring and controlling usage. Broadcast range for Zigbee is approximately 70 meters. Theoretically Zigbee networks can contain up to 64 k (65,536) network nodes. Current testing has not reached anywhere near that level. The name ZigBee is said to come from the domestic honeybee, which uses a zigzag type of dance to communicate important information to other hive members.

D. DIFFERENT TYPES OF ANTENNA OF ZIGBEE

The 3 different antennas for Zigbee are as shown in the Fig.
7 such as the Whip antenna, UFL RF connector & the Chip antenna. The chip and integrated whip antennas are suited for any application, but are especially useful in embedded applications. Since the radios do not have any issue radiating through plastic cases or housings, the antennas can be completely enclosed in those types of applications.

The UFL Connector is used in conjunction with an adaptor cable that can be connected to a dipole or gain antenna if the housing is metal or if that solution is more desirable mechanically. Range can differ somewhat with different antenna types, so that should be a consideration when choosing what type of antenna you want to use.

E. DIFFERENT TYPES OF TOPOLOGIES

The three types of topologies that Zigbee supports are shown in Fig. 8 such as the Star topology, Peer to peer topology & the Cluster tree.

1) STAR TOPOLOGY

In the star topology, the communication is established between the devices and a single central controller called PAN coordinator. The PAN coordinator may be mains powered while the devices will most likely be battery powered. Applications that benefit from this topology include home automation, personal computer (PC) peripherals, toys and games. After an FFD is activated for the first time it may establish its own network and become the PAN coordinator. Each star network chooses a PAN identifier, which is not currently used by any other network within the radio spear of influence. This allows each star network to operate independently.

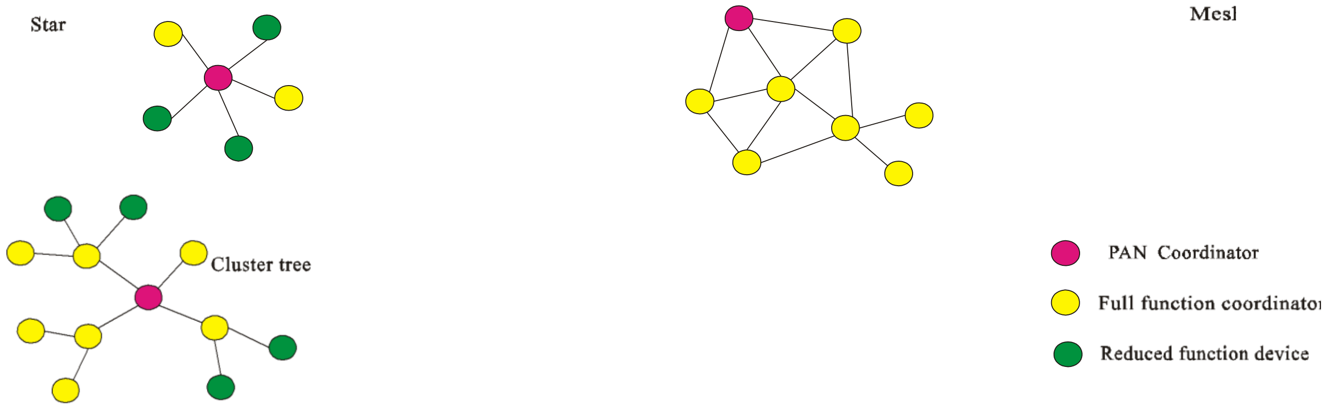


Fig. 8 Different Network Topologies

2) PEER TO PEER TOPOLOGY

In the Peer-to-Peer topology, there is also one PAN coordinator.In contrast to star topology, any device can communicate with any other device as long as they are in range of one another. A Peer-to-Peer network can be ado, self-organizing and self-healing. Applications such as industrial control and monitoring, wireless sensor networks assert and inventory tracking wood benefit from such a topology. It also allows multiple hope to root massages from any device to any other device in the network. It can provide reliability by multipath rooting.

3) CLUSTER TREE TOPOLOGY

Cluster tree network is special case of Peer-to-Peer network in which most devices are FFD's and an RFD may connect to a cluster tree network as a leave node at the end of a branch. Any of the FFD can act as a coordinator and provide synchronization services to other devices and coordinators.

Only one of this coordinator however is the PAN coordinator. The PAN coordinator forms the first cluster by establishing itself as the cluster head (CLH) with a cluster identifier (CID) of zero, choosing an unused PAN identifier, and broadcasting beacon frames to neighbouring devices. A candidate device receiving a beacon frame may request to join the network at the CLH.

If the PAN coordinator permits the device to join it will add this new device as a child device in its neighbour list. The newly joined device will add the CLH as its parent in its neighbour list and begin transmitting periodic beacons such that other candidate devices may then join the network at that device. The advantage of this clustered structure is the increased coverage area at the cost of increased message latency.

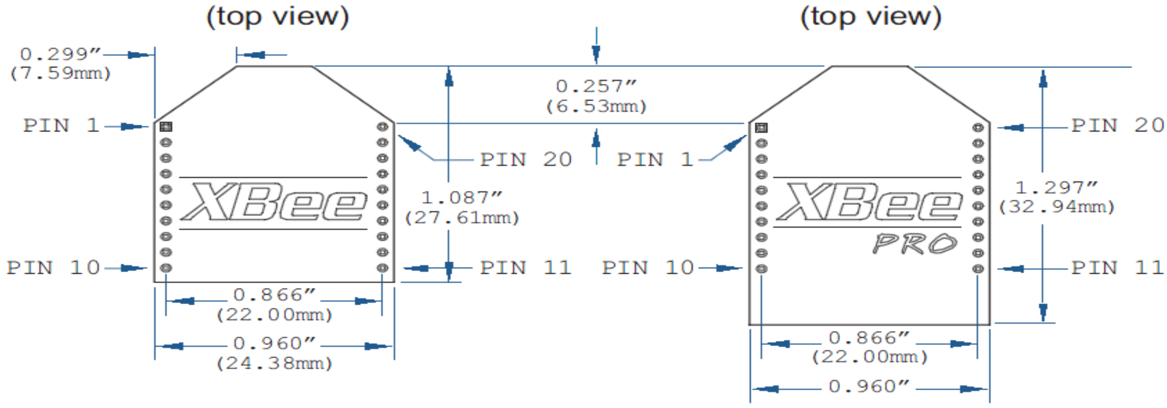


Fig. 9 Top View of Zigbee

F. PRODUCT SUMMARY

- ISM 2.4 GHz operating frequency.
- 1 milli watt (0dBm) power output (up to 100 m range).
- U.FL RF connector, chip or whip antenna options.
- Industrial (-40 to 85 degree Celsius) temperature rating.
- Approved for use in the United States, Canada and
- Europe.
- Advanced networking and low power modes supported.

G. TOP VIEW OF ZIGBEE

The top view of the Zigbee is as shown in the Fig. 9.

H. PERFORMANCE, NETWORKING, POWER, GENERAL AND PHYSICAL PROPERTIES OF ZIGBEE & KEY FEATURES

- Price to performance value.
- Low power consumption.

- Receiver sensitivity.
- Worldwide acceptance in USA, CANADA & EUROPE.

Systems that contain Zigbee modules can operate under the certifications obtained by maxstream's. Further testing is not required.

I. RANGES FOR ZIGBEE

Indoor / urban range up to 100c (30m)
Outdoor line-of-sight range up to 300c (100m) Transmit power output
1mw (0dBm)
Power – down current <10P$
Operating frequency 2.4 GHz

J. ZIGBEE ALSO OFFERS

Low power consumption: optimized for low battery operation.
License free operation in the 2.4GHz band.

Simple protocol definition can be implemented on low cost micro controllers.
Hundreds of devices per network exists.

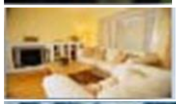
Network flexibility:star, cluster tree or mesh configuration.
Data rate up to 250 kbps.
Small size – The developed solution will be less than 9mm
& 9mm.

K. APPLICATIONS OF ZIGBEE

There are a number of applications that can benefit from the Zigbee protocol: Building automation networks, home security systems, industrial control networks, remote metering and PC peripherals are some of the many possible applications.

Security systems and lighting controls.

Home automation and building control.



Home appliances and fire alarms.

Monitoring of remote systems. Sensor data capture in embedded Networks.

VI. OVERALL WORKING OF THE UNIT

This section explains the block diagram of the designed & fabricated system, working of the transmitter and receiving section with the circuit diagram and flow chart.

A. OVER VIEW

The temperature is sensed by the LM35 temperature sensor. The output of LM35 is a voltage signal, which is very weak. This signal is not enough to display 25 degree centigrade of room temperature. At room temperature LM35 gives 0.3 milli- volts of voltage so an amplifier is used to amplify this voltage to 0.5 milli-volts, which produces 25 degree centigrade at CENTRAL SERVER. The amplifier amplifies this signal.

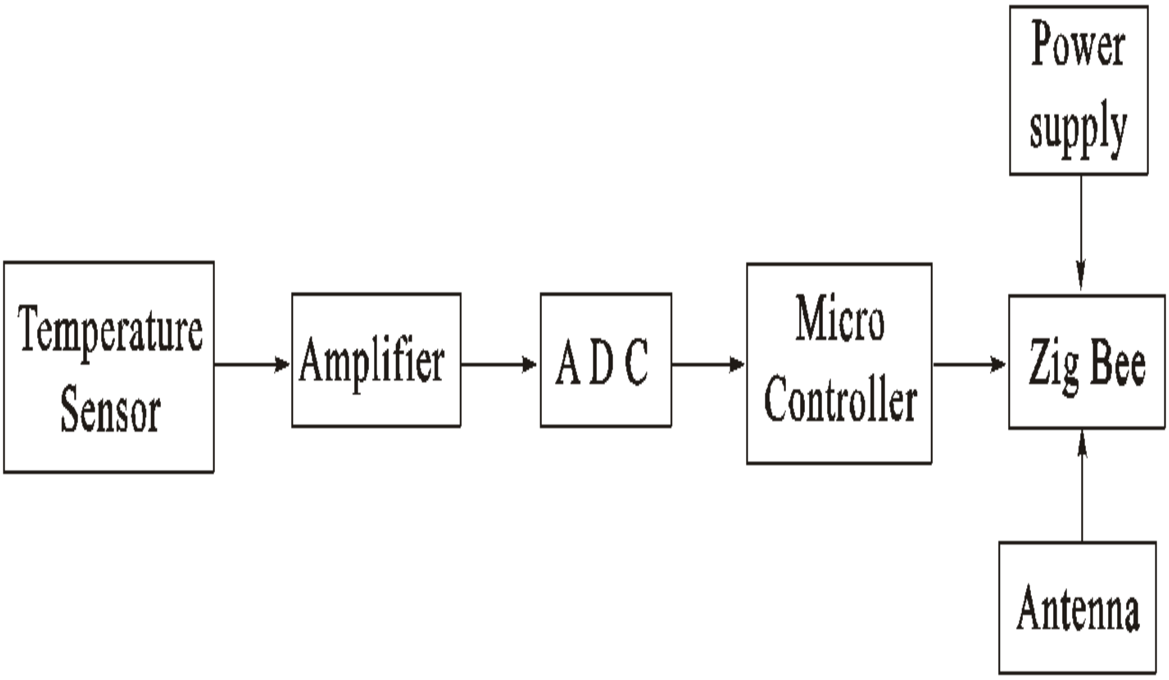


Fig 14. Block diagram of transmitting section

The output of the amplifier is given to the PIC microcontroller as shown in the Fig. 14. The inbuilt ADC of PIC microcontroller converts this analog data to digital form. The parallel data received is converted to serial data by the inbuilt USART. The Zigbee with whip antenna transmits this data. Power supply of 2.8V to 3.3V is designed for the operation of Zigbee. The Zigbee at the receiving side receives the transmitted data.
The similar power supply is designed for receiving Zigbee as in the transmitter as shown in the Fig. 15. This data is fed to the PIC microcontroller through pin 18. The serial data received is converted into parallel data by the inbuilt USART and is displayed on the four byte CENTRAL SERVER. The buzzer is connected to pin 15 of PIC microcontroller. If the sensed temperature exceeds 40 degree C, the buzzer is on.

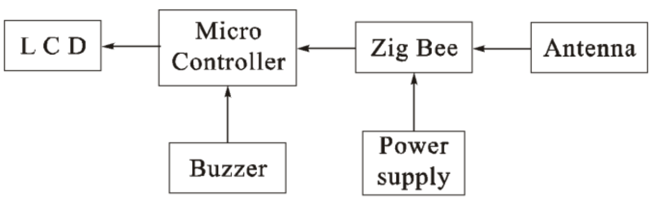


Fig. 15 Block Diagram of receiving section

The block diagram of the system is shown in the Figs. 14 &
15 respectively. It gives the over view of the designed & implemented system and information about different components used. The LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibration as it is inherently calibrated.

It outputs 10mv for each degree of centigrade temperature. The amplifier in the above diagram is used to increase the strength of the signal, which can be accessed by the microcontroller. Microcontroller uses binary values but the output of amplifier is analog. Hence we need an analog to digital converter to translate the analog signals to digital numbers, so that the microcontroller can access them. The ADC chip used here is 0808.
Microcontroller used here is the PIC18F4550. It has Harvard architecture, whose memory organization is divided into data memory and program memory as separate units. It has three ports namely PA, PB, PC. It has four register banks with thirty-five-instruction set. It has inbuilt ADC and USART. Zigbee is a recently developed

two-way wireless communications protocol designed to meet very low Power consumption and low cost (half that of Blue tooth) requirements. Protocol features include automatic network configuration, full handshaking for packet transfers (reliable data transfer), data rate of 20kbps at 868 MHz, 40kbps at 915 MHz and
250kbps at 2.4 MHz, power management features. Zigbee needs an input power between 2.8 to 3.3v. Hence, we design a power supply, which converts 230v to 3.3v. Here, we use whip antenna to transmit the signal from transmitter to the receiver [8].

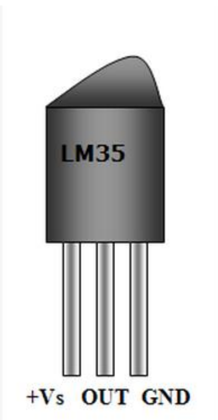
B. TRANSMITTER

- The temperature is sensed by LM35 temperature sensor.
- The output of LM35 is a voltage signal, which is very weak. The amplifier amplifies this signal.
- The output of the amplifier is given to the PIC
- Microcontroller.
- This analog data is converted to digital form by the inbuilt
- ADC.
- The parallel data received is converted to serial data by the inbuilt USART.
- The Zigbee with whip antenna transmits this data.
- Power supply of 2.8V to 3.3V is designed for the operation of Zigbee.

VII. Interfacing the different parts together

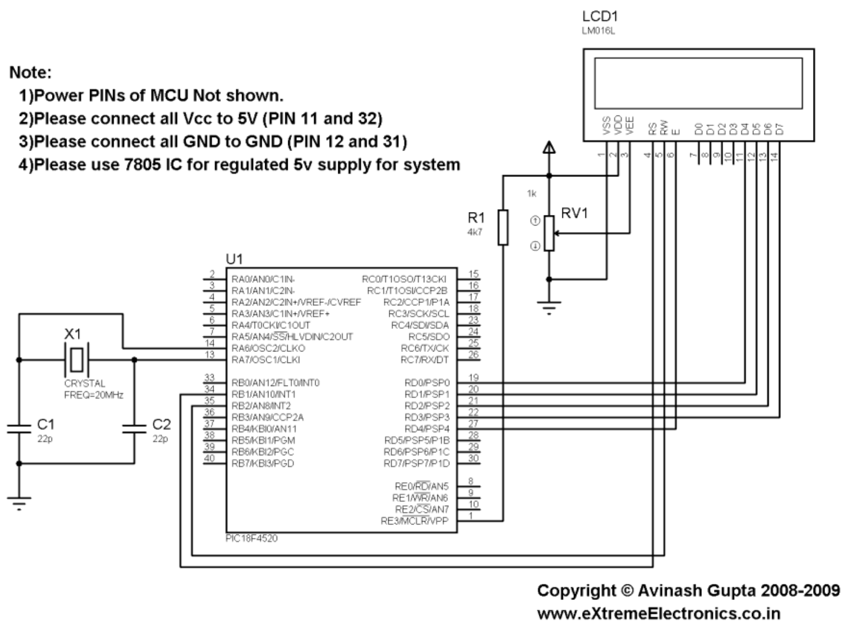
1. INTERFACING LM-35(TEMP SENSOR WITH PIC18F4550)

LM35 is a 3 pin device as show below Connect the +Vs Pin to 5v and GND to GND. The output must be connected to the analog input pin 0 of the PIC18F4520 MCU. It is labelled AN0 in the datasheet. It is pin number 2 on the 40 pin package.



It is also called **RA0** because it is shared with PORTA0.

2. INTERFACING LCD WITH PIC18F4550



3. INTERFACING Xbee with PIC18F4550

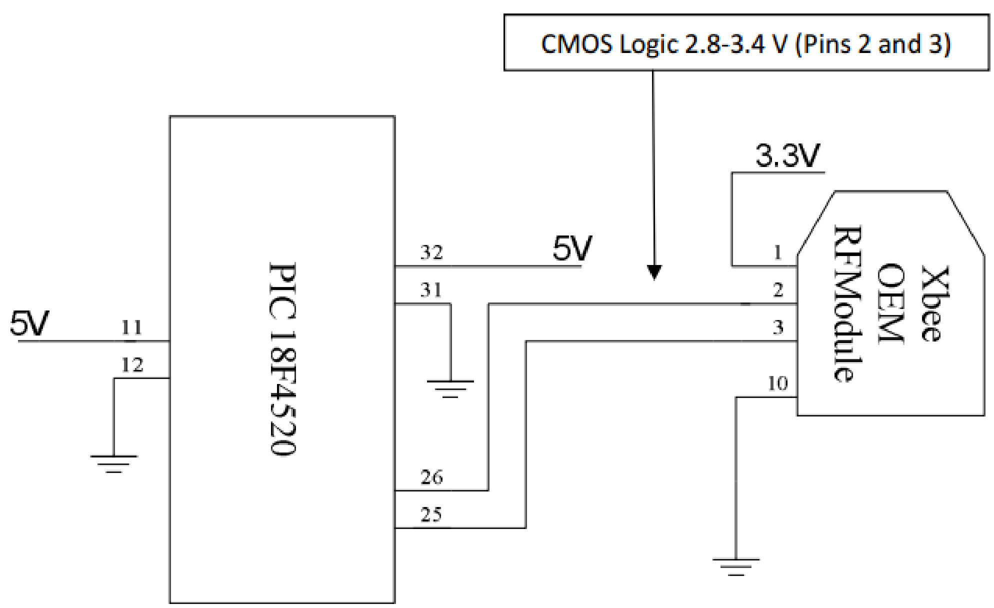


Figure 4: Circuit Schematic PIC to Xbee

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTA is a bidirectional I/O port.
RA0/AN0	2	19	19			
RA0				I/O	TTL	Digital I/O.
AN0				I	Analog	Analog input 0.
RA1/AN1	3	20	20			
RA1				I/O	TTL	Digital I/O.
AN1				I	Analog	Analog input 1.
RA2/AN2/VREF-/CVREF	4	21	21			
RA2				I/O	TTL	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (low) input.
CVREF				O	Analog	Analog comparator reference output.
RA3/AN3/VREF+	5	22	22			
RA3				I/O	TTL	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (high) input.
RA4/T0CKI/C1OUT/RCV	6	23	23			
RA4				I/O	ST	Digital I/O.
T0CKI				I	ST	Timer0 external clock input.
C1OUT				O	—	Comparator 1 output.
RCV				I	TTL	External USB transceiver RCV input.
RA5/AN4/$\overline{SS}$/HLVDIN/C2OUT	7	24	24			
RA5				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
$\overline{SS}$				I	TTL	SPI™ slave select input.
HLVDIN				I	Analog	High/Low-Voltage Detect input.
C2OUT				O	—	Comparator 2 output.
RA6	—	—	—	—	—	See the OSC2/CLKO/RA6 pin.

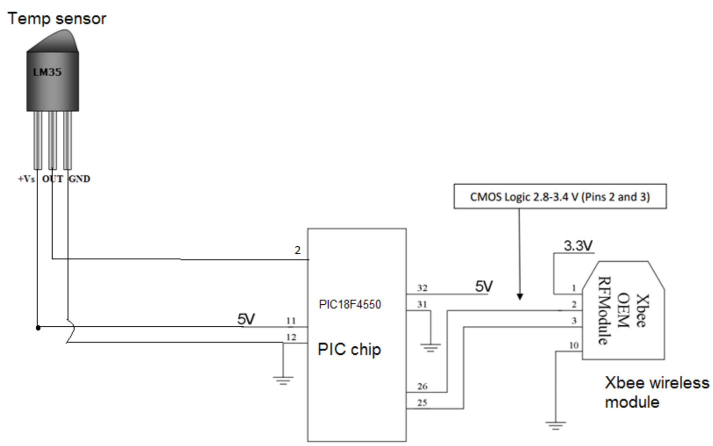
Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
$\overline{MCLR}$/VPP/RE3	1	18	18			Master Clear (input) or programming voltage (input).
MCLR				I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP				P		Programming voltage input.
RE3				I	ST	Digital input.
OSC1/CLKI	13	32	30			Oscillator crystal or external clock input.
OSC1				I	Analog	Oscillator crystal input or external clock source input.
CLKI				I	Analog	External clock source input. Always associated with pin function OSC1. (See OSC2/CLKO pins.)
OSC2/CLKO/RA6	14	33	31			Oscillator crystal or clock output.
OSC2				O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	—	In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6				I/O	TTL	General purpose I/O pin.

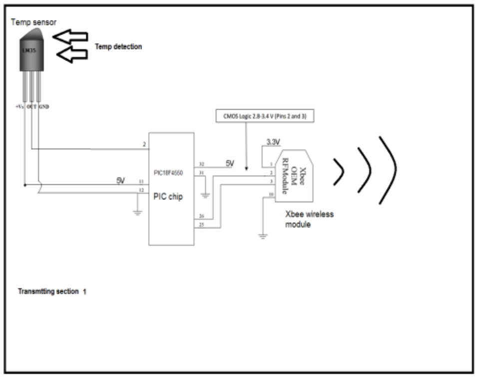
Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	15	34	32			
RC0				I/O	ST	Digital I/O.
T1OSO				O	—	Timer1 oscillator output.
T13CKI				I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2/ $\overline{\text{UOE}}$	16	35	35			
RC1				I/O	ST	Digital I/O.
T1OSI				I	CMOS	Timer1 oscillator input.
CCP2(2)				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
$\overline{\text{UOE}}$				O	—	External USB transceiver $\overline{\text{OE}}$ output.
RC2/CCP1/P1A	17	36	36			
RC2				I/O	ST	Digital I/O.
CCP1				I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output.
P1A				O	TTL	Enhanced CCP1 PWM output, channel A.
RC4/D-/VM	23	42	42			
RC4				I	TTL	Digital input.
D-				I/O	—	USB differential minus line (input/output).
VM				I	TTL	External USB transceiver VM input.
RC5/D+/VP	24	43	43			
RC5				I	TTL	Digital input.
D+				I/O	—	USB differential plus line (input/output).
VP				I	TTL	External USB transceiver VP input.
RC6/TX/CK	25	44	44			
RC6				I/O	ST	Digital I/O.
TX				O	—	EUSART asynchronous transmit.
CK				I/O	ST	EUSART synchronous clock (see RX/DT).
RC7/RX/DT/SDO	26	1	1			
RC7				I/O	ST	Digital I/O.
RX				I	ST	EUSART asynchronous receive.
DT				I/O	ST	EUSART synchronous data (see TX/CK).
SDO				O	—	SPI™ data out.

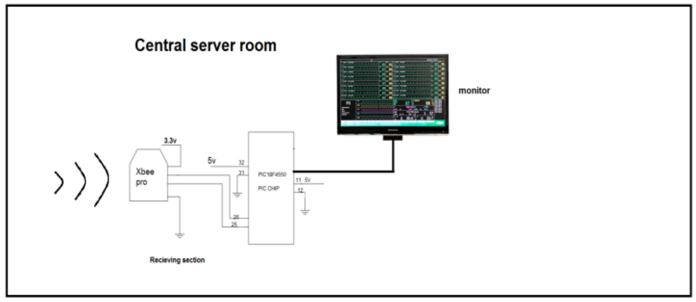
Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/AN12/INT0/FLT0/SDI/SDA	33	9	8			
RB0				I/O	TTL	Digital I/O.
AN12				I	Analog	Analog input 12.
INT0				I	ST	External interrupt 0.
FLT0				I	ST	Enhanced PWM Fault input (ECCP1 module).
SDI				I	ST	SPI™ data in.
SDA				I/O	ST	I²C™ data I/O.
RB1/AN10/INT1/SCK/SCL	34	10	9			
RB1				I/O	TTL	Digital I/O.
AN10				I	Analog	Analog input 10.
INT1				I	ST	External interrupt 1.
SCK				I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL				I/O	ST	Synchronous serial clock input/output for I²C mode.
RB2/AN8/INT2/VMO	35	11	10			
RB2				I/O	TTL	Digital I/O.
AN8				I	Analog	Analog input 8.
INT2				I	ST	External interrupt 2.
VMO				O	—	External USB transceiver VMO output.
RB3/AN9/CCP2/VPO	36	12	11			
RB3				I/O	TTL	Digital I/O.
AN9				I	Analog	Analog input 9.
CCP2[(1)]				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
VPO				O	—	External USB transceiver VPO output.
RB4/AN11/KBI0/CSSPP	37	14	14			
RB4				I/O	TTL	Digital I/O.
AN11				I	Analog	Analog input 11.
KBI0				I	TTL	Interrupt-on-change pin.
CSSPP				O	—	SPP chip select control output.
RB5/KBI1/PGM	38	15	15			
RB5				I/O	TTL	Digital I/O.
KBI1				I	TTL	Interrupt-on-change pin.
PGM				I/O	ST	Low-Voltage ICSP™ Programming enable pin.
RB6/KBI2/PGC	39	16	16			
RB6				I/O	TTL	Digital I/O.
KBI2				I	TTL	Interrupt-on-change pin.
PGC				I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD	40	17	17			
RB7				I/O	TTL	Digital I/O.

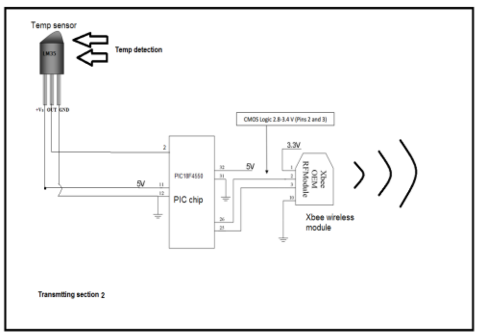
Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTD is a bidirectional I/O port or a Streaming Parallel Port (SPP). These pins have TTL input buffers when the SPP module is enabled.
RD0/SPP0	19	38	38			
RD0				I/O	ST	Digital I/O.
SPP0				I/O	TTL	Streaming Parallel Port data.
RD1/SPP1	20	39	39			
RD1				I/O	ST	Digital I/O.
SPP1				I/O	TTL	Streaming Parallel Port data.
RD2/SPP2	21	40	40			
RD2				I/O	ST	Digital I/O.
SPP2				I/O	TTL	Streaming Parallel Port data.
RD3/SPP3	22	41	41			
RD3				I/O	ST	Digital I/O.
SPP3				I/O	TTL	Streaming Parallel Port data.
RD4/SPP4	27	2	2			
RD4				I/O	ST	Digital I/O.
SPP4				I/O	TTL	Streaming Parallel Port data.
RD5/SPP5/P1B	28	3	3			
RD5				I/O	ST	Digital I/O.
SPP5				I/O	TTL	Streaming Parallel Port data.
P1B				O	—	Enhanced CCP1 PWM output, channel B.
RD6/SPP6/P1C	29	4	4			
RD6				I/O	ST	Digital I/O.
SPP6				I/O	TTL	Streaming Parallel Port data.
P1C				O	—	Enhanced CCP1 PWM output, channel C.
RD7/SPP7/P1D	30	5	5			
RD7				I/O	ST	Digital I/O.
SPP7				I/O	TTL	Streaming Parallel Port data.
P1D				O	—	Enhanced CCP1 PWM output, channel D.

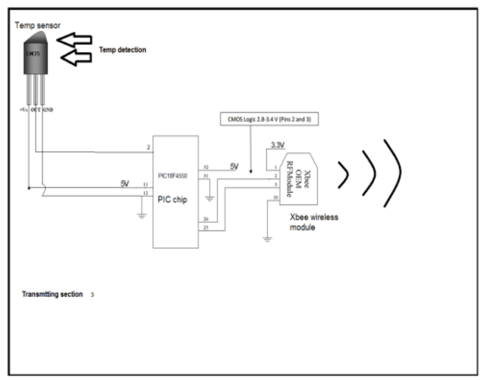
VIII. Pin Diagram for the combined connections




Temp sensor
Temp detection
CMOS Logic 2.8-3.4 V (Pins 2 and 3)
3.3V
5V
PIC18F4550
PIC chip
Xbee OEM RFModule
Xbee wireless module
Transmitting section 1


Central server room
monitor
3.3v
5v
Xbee pro
PIC18F4550
PIC CHIP
Recieving section


Temp sensor
Temp detection
CMOS Logic 2.8-3.4 V (Pins 2 and 3)
3.3V
5V
PIC18F4550
PIC chip
Xbee OEM RFModule
Xbee wireless module
Transmitting section 2


Temp sensor
Temp detection
CMOS Logic 2.8-3.4 V (Pins 2 and 3)
3.3V
5V
PIC18F4550
PIC chip
Xbee OEM RFModule
Xbee wireless module
Transmitting section 3

IX. Algorithm and code for the system

The algorithm:

1. First the value of the data is read from LM35 by programming the PIC 18f4500 using the internal ADC module of PIC 18f4500.The data from the LM35 is read into the PIC from the following pseudo code.

```
unsigned int val;

val=ADCRead(0);  (Read Channel 0)

voltage= ((val)/1023.0)*5;

voltage=((val)/1023.0)*5*1000); (Voltage is in mV)

t=((val)/1023.0)*5*100); (t is in degree centigrade)
```

2. The data from LM35 is received continuously and stored in a pointer. As the xbee is interfaced with the pic 18f4500, the xbee module (room) is commanded to send the data value to the other xbee module (server) which is interfaced with pic module. Also the server PIC is coded to return data to the room PIC to change the temperature of the target room.

Code for LM-35

```
//unsigned int val;

//val=ADCRead(0); //Read Channel 0

//voltage= ((val)/1023.0)*5;

//voltage=((val)/1023.0)*5*1000); //Voltage is in mV

//t=((val)/1023.0)*5*100); //t is in degree centigrade


#include <htc.h>

#include <math.h>

#include "lcd.h"

//Chip Settings

__CONFIG(1,0x0200);

__CONFIG(2,0X1E1F);

__CONFIG(3,0X8100);

__CONFIG(4,0X00C1);

__CONFIG(5,0XC00F);

//Simple Delay Routine

void Wait(unsigned int delay)

{
```

```
  for(;delay;delay--)
      __delay_us(100);
}
//Function to Initialise the ADC Module
void ADCInit()
{
   //We use default value for +/- Vref
   //VCFG0=0,VCFG1=0
   //That means +Vref = Vdd (5v) and -Vref=GEN
    //Port Configuration
   //We also use default value here too
 //All ANx channels are Analog
 /*
      ADCON2
       *ADC Result Right Justified.
      *Acquisition Time = 2TAD
      *Conversion Clock = 32 Tosc
   */
  ADCON2=0b10001010;
}
```

```
//Function to Read given ADC channel (0-13)
unsigned int ADCRead(unsigned char ch)
{
   if(ch>13) return 0;  //Invalid Channel
    ADCON0=0x00;
   ADCON0=(ch<<2);   //Select ADC Channel
   ADON=1;  //switch on the adc module
   GODONE=1; //Start conversion
   while(GODONE); //wait for the conversion to finish
   ADON=0;  //switch off adc
   return ADRES;
}
void main()
{
   //Initialize the ADC Module
   ADCInit();
   while(1)
   {
      unsigned int val; //ADC Value
      unsigned int t;      //Temperature
```

```
        val=ADCRead(0);    //Read Channel 0

        t=round(val*0.48876); //Convert to Degree Celcius

        Wait(1000);

    }

}
```

Wireless Transmission

```
void uart_init(void)
{
SPBRG=129; //set baud rate as 9600 baud
BRGH=1; //baud rate high speed option
TXEN=1; //enable transmission
TX9 =0; //8-bit transmission
RX9 =0; //8-bit reception
CREN=1; //enable reception
SPEN=1; //enable serial port
}
```

//UART Transmit and Receive coding configuration:

```
unsigned int uart_rec(void) //receive uart value
{
unsigned int rec_data; //here the data is the output value of LM35,t
while(RCIF==0); //wait for data
rec_data = RCREG;
return rec_data; //return the data received
}

void uart_send(unsigned int data)
{
```

```
while(TXIF==0); //only send the new data after
TXREG=data; //the previous data finish sent
}

void uart_data(const int *s)
{
while(*s)uart_send(*s++); // UART sending data as pointer elements
}
```

//Now, we can write the coding to send command for setting SKXBee Source address and //Destination address. However, we have to refer the AT Command Mode table. The main //reason is the command code exactly same with the code we have using X-CTU software for //setting the parameter.

//For initialize the Source address and Destination address to SKXBee1 module, we can write //the coding in *"void main()"* code as:

```
void xbee_init(void)
{
uart_data(&quot;+++&quot;); //send command to enter XBee Pro command mode
delay_ms(200); // waiting for finish sending and XBee respond

uart_data(&quot;atmy1111&quot;); //send command to setting Source address
uart_send(0xD); // 0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

uart_data(&quot;atwr&quot;); // send &quot;WR&quot; (write)command to SKXBee
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

uart_data(&quot;atdl2222&quot;); // send command to setting Destination
address
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

uart_data(&quot;atwr&quot;); //send &quot;WR&quot; (write)command to SKXBee
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
```

```
delay_ms(200);

uart_data(&quot;atcn&quot;); // send command for Exit AT Command Mode
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

}
```

//For initialize the Source address and Destination address to SKXBee2 module, we can write
//the coding in **another microcontroller PIC** source code *“void main()”* as:

```
void xbee_init(void)
{
uart_data(&quot;+++&quot;); //send command to enter XBee Pro command mode
delay_ms(200); // waiting for finish sending and XBee respond

uart_data(&quot;atmy2222&quot;);//send command to setting Source address
uart_send(0xD); // 0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

uart_data(&quot;atwr1111&quot;); // send &quot;WR&quot; (write)command to
SKXBee
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

uart_data(&quot;atdl22&quot;); // send command to setting Destination address
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

uart_data(&quot;atwr&quot;); //send &quot;WR&quot; (write)command to SKXBee
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

uart_data(&quot;atcn&quot;); // send command for Exit AT Command Mode
uart_send(0xD); //0XD is hexa code of &quot;Enter&quot;key
delay_ms(200);

}
```

X. CONCLUSION

Temperature sensing unit is studied & tested successfully. The successful operation of the unit was also carried out in the presence of numerous tests. We have done for one parameter, but the same can also be done for 2 or more parameters like pressure, viscosity and humidity in real time applications. This system is cost-effective and time efficient in its operation and works as high speed and is very accurate in its operation.

The development of wireless solution within the standards organization has the advantage of bringing several views together to define a better solution. The quick development of the standard was due to the proactive participation of several developers and users of the technology. The focus of Zigbee development was on maintaining simplicity by concentrating on the essential requirements that will leverage a successful standard.

The main goal of this effort has been to address applications that could benefit from wireless connectivity and enable new ones that cannot use higher-end wireless application. The value will be in the application, not in the technology. Some of the future developments could also be discussed as follows. With the facilities provided, we were successful in sensing the temperature as one of the data and displaying the same on the CENTRAL SERVER.

We can add other parameters say voltage, pressure, viscosity, humidity etc. to the existing work done in this report or for transfer of data using Zigbee in the place of internet and could have made it more efficient.

XI. REFERENCES

[1]. Sepoo Scottsdale, AZUSA, "Znterim Meeting Minuts", Scan Middletom, Philip semiconductors.

[2]. www.eg3.com/zigbee

[3]. www.zigbee.org

[4]. www.palowireless.com/zigbee

[5]. www.enbedded.com/showarticle.jhtml

[6]. www.wisegeetc.com

[7]. G Ding, Z Sahinoglu, B Bhargava, P Orlik, J Zhang , "Reliable Broad cast en Zigbee networks", Proc. of IEEE conference on Zigbee.

[8]. U Mohammad, "Performance Evaluation of IEEE 802.15.4 Protocol for Automotive application", IEEE Manuals.

[9]. Building wireless sensor networks – Robert faludi