Chapter-1

Introduction

Nowadays, the world is facing many challenges in reducing energy consumption and global warming. In the same time, there are many technologies that can be used to resolve these problems and moreover support better living. Wireless Sensor networks (WSNs) are the technology that could provides ubiquitous computing. WSNs deploy many small sensor devices to detect environmental properties. WSNs are the combination of embedded system and wireless communication which allows data transmission among the sensor nodes over ad-hoc wireless networks. The ad hoc networks requires no existing infrastructures unlike those in WLAN or cellular networks. The brain of each WSN node is the microcontroller which processes readings from its own sensors and, in some cases, readings from adjacent nodes as well since the sensors in different nodes located near each other may be highly correlated and hence the amount of sensor data needed to be transmitted from the two sensor nodes can be cooperatively reduced. In this prototype system, we develop an embedded wireless sensor prototype system for temperature monitoring at the Integral University (I.U.). The system is used to remotely monitor temperature in each classroom. It provides necessary data for the whole energy consumption management of the air conditioning systems. We use the X-Bee modules based on the IEEE 802.15.4/Zigbee Wireless Personal Area Network (WPAN) standards to build a low-power, low-maintenance, and self-organizing WSN. Small size, low power, low cost and long battery life are the reasons of using ZigBee. We use the Arduino board which comes with ATmega168 or 328 for easy interfacing with the ZigBee module and for easy programming (in C) of the microcontroller[3]. The Arduino boards come with a library for interfacing with X-Bee module and for dealing with analog or digital inputs and outputs. For the temperature monitoring sensor, we use a low-cost analog sensor to show the proof of concept. The remaining of the paper is organized as follows. In Section II, we discuss some related description of project. Section III describes the methodology and working used in this project. Sections IV describe the testing environment and the results, respectively. We conclude the paper and discuss some future extensions of our current prototype in Section IV.

History

Wireless sensing technology can collect pressure, temperature, and flow measurements in remote and often unsafe locations that is common in the offshore/on land oil and gas industry without cables and the associated problems. However, the severe offshore conditions make it necessary to develop reliable and cost-effective real-time monitoring structures when building offshore control and monitoring systems. Nowadays, ZigBee RF standard is deployed. This has opened new perspectives for wireless control networks. ZigBee is powerful and easy to install because it was developed in order to be installed in a new or existing sensor network. RASHPETCO has many geographically distributed offshore platforms. Some types of ZigBee radio modules provide a transmission range over 40 km with multi-hop communication capability to extend the coverage. Many oil and gas applications, where ZigBee technology can be deployed, can be overseen[1].

Chapter-2

CONTENTS

S.no.	Components	Page no.
1.	Atmega-168	6-14
2.	Capacitors	15-16
3.	Crystal Oscilator	17-19
4.	DS18B20	20-22
5.	Express P.C.B.	23-28
6.	LCD 16X2	29-31
7.	UART Module	32-33
8.	X-Bee	34-43

ATMEGA-168

INTRODUCTION

The computer on one hand is designed to perform all the general purpose tasks on a single machine like you can use a computer to run a software to perform calculations or you can use a computer to store some multimedia file or to access internet through the browser, whereas the microcontrollers are meant to perform only the specific tasks, for e.g., switching the AC off automatically when room temperature drops to a certain defined limit and again turning it ON when temperature rises above the defined limit[5].

There are number of popular families of microcontrollers which are used in different applications as per their capability and feasibility to perform the desired task, most common of these are 8051, AVR and PIC microcontrollers. In this we will introduce you with AVR family of microcontrollers[3].

History of AVR

AVR was developed in the year 1996 by Atmel Corporation. The architecture of AVR was developed by Alf-Egil Bogen and Vegard Wollan. AVR derives its name from its developers and stands for Alf-Egil Bogen Vegard Wollan RISC microcontroller, also known as Advanced Virtual RISC[2].

AVR microcontrollers are available in three categories:

Tiny AVR – Less memory, small size, suitable only for simpler applications

Mega AVR – These are the most popular ones having good amount of memory (up-to 256 KB), higher number of in-built peripherals and suitable for moderate to complex applications.

Xmega AVR – Used commercially for complex applications, which require large program memory and high speed.

Features

- RISC Architecture with CISC Instruction set
- Powerful C and assembly programming
- Same powerful AVR microcontroller core
- Low power consumption
- Both digital and analog input and output interfaces.

Description

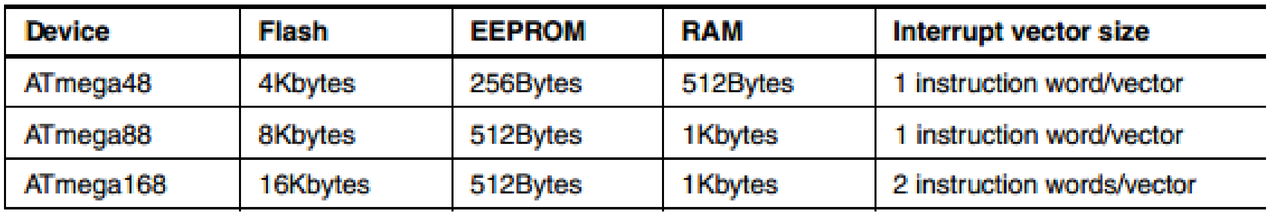
The Atmel ATmega48/88/168 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48/88/168 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed[8][9].

The Atmel ATmega48/88/168 provides the following features:

4K/8K/16K bytes of In-System Programmable Flash with Read-While-Write capabilities 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit,a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes.

The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The ATmega48, ATmega88 and ATmega168 differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupts vector sizes for the three devices[8].



Device	Flash	EEPROM	RAM	Interrupt vector size
ATmega48	4Kbytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88	8Kbytes	512Bytes	1Kbytes	1 instruction word/vector
ATmega168	16Kbytes	512Bytes	1Kbytes	2 instruction words/vector

PROCESS ARCHITECTURE

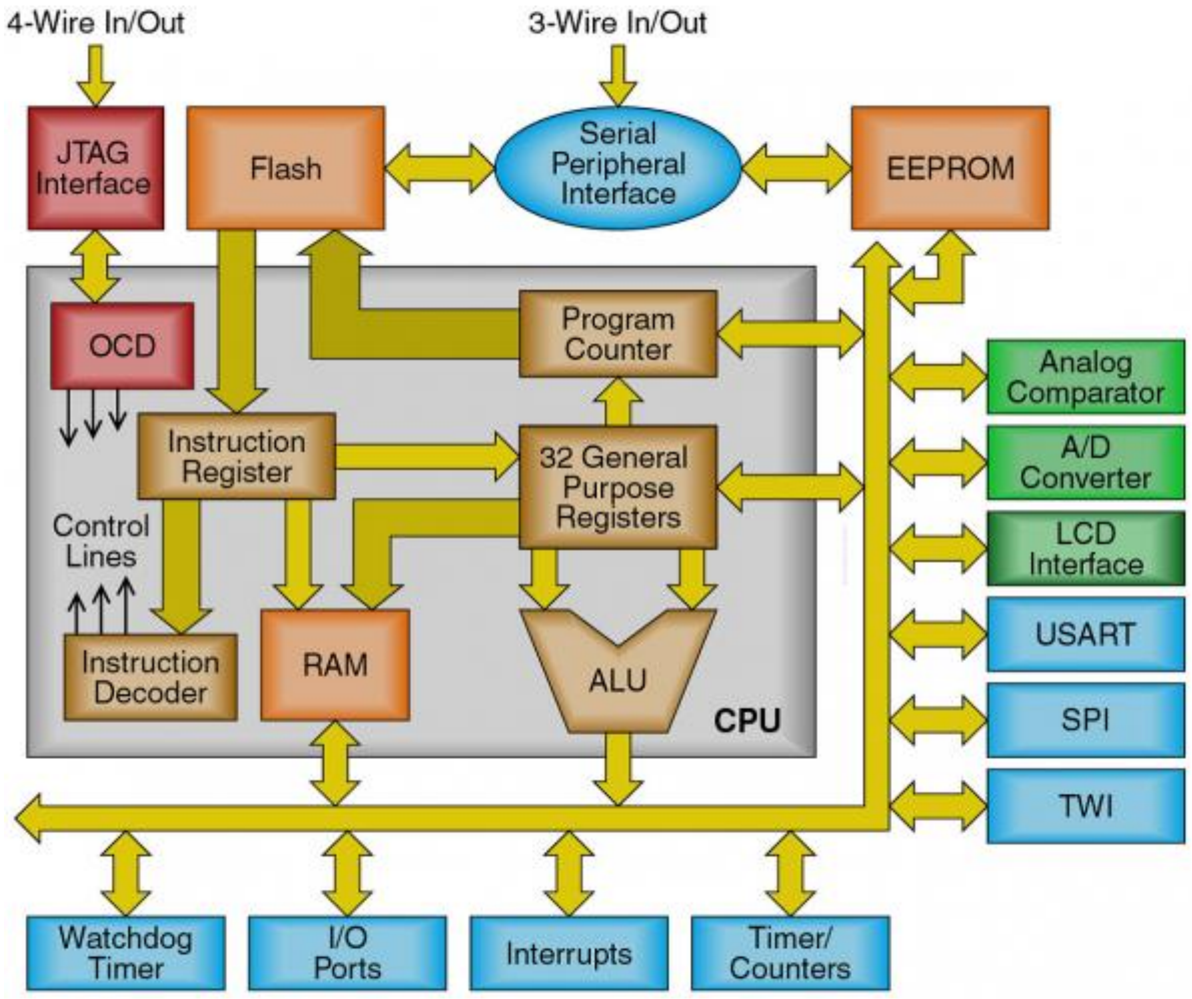


FIG. 2.1.1

AVR follows Harvard Architecture format in which the processor is equipped with separate memories and buses for Program and the Data information. Here while an instruction is being executed, the next instruction is pre-fetched from the program memory.

ALU: The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format.

Flash program memory: The ATmega48/88/168 contains 4K/8K/16K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 2K/4K/8K × 16. For software security, the Flash Program memory space is divided into two sections, Boot Loader Section and Application Program Section in ATmega88 and ATmega168.

EEPROM data memory: The Atmel ATmega48 /88/168 contains 256/512/512 bytes of data EEPROM memory. It is organized as a separate data space e, in which single bytes can be read and written.

Program counter: A program counter is a register in a computer processor that contains the address (location) of the instruction being executed at the current time. As each instruction gets fetched, the program counter increases its stored value by 1. After each instruction is fetched, the program counter points to the next instruction in the sequence. When the computer restarts or is reset, the program counter normally reverts to 0.

RAM: RAM stands for random access memory. This type of memory storage is temporary and volatile. You might have heard that if your system is working slowly you say that increase the RAM processing will increase. This is the case with RAM also if you increase the RAM the address basically increases for temporary processing so that no data has to wait for its turn. However the storage is temporary every time u boot your system the data is lost but when you turn on the system The BIOS fetch number of addresses available in the RAM. This memory supports read as well as write operations both.

INSTRUCTION EXECUTION SECTION (IES): It has the most important unit—instruction register and instruction decoder to control the flow of the instruction during the processing's.

INPUT/OUTPUT PORTS: To interact with the physical environment there are different input and output ports in every system like in PC we have VGA port to connect the monitor, USB port for flash memory connections and many more ports. Similarly ATMEGA 168 has its input and output ports with different configurations depending on the architecture like only input, only output and bi-directional input output ports[5].

PIN DIAGRAM AND DESCRIPTION

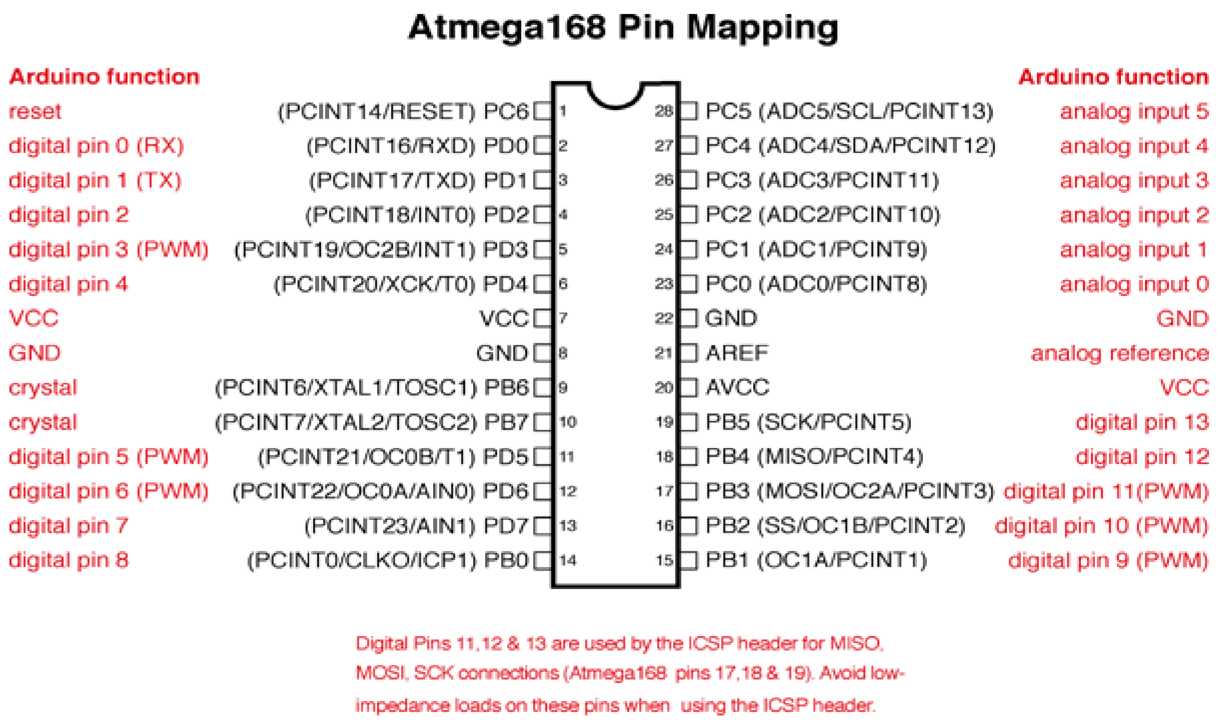


FIG. 2.1.2.

PIN DESCRIPTION:

VCC: Digital supply voltage.

GND: Ground.

Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull- up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source Capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit[2][6]. Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB7.6 is used as TOSC2.1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5.0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET:

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is un-programmed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guarantee to generate a reset[3].

Port D (PD7:0):

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

MINIMUM INTERFACE CIRCUIT FOR ATMEGA168 CONTROLLER:

According the minimum interface discussed for the microcontrollers earlier, the minimum interface circuit for ATMEGA168 is:

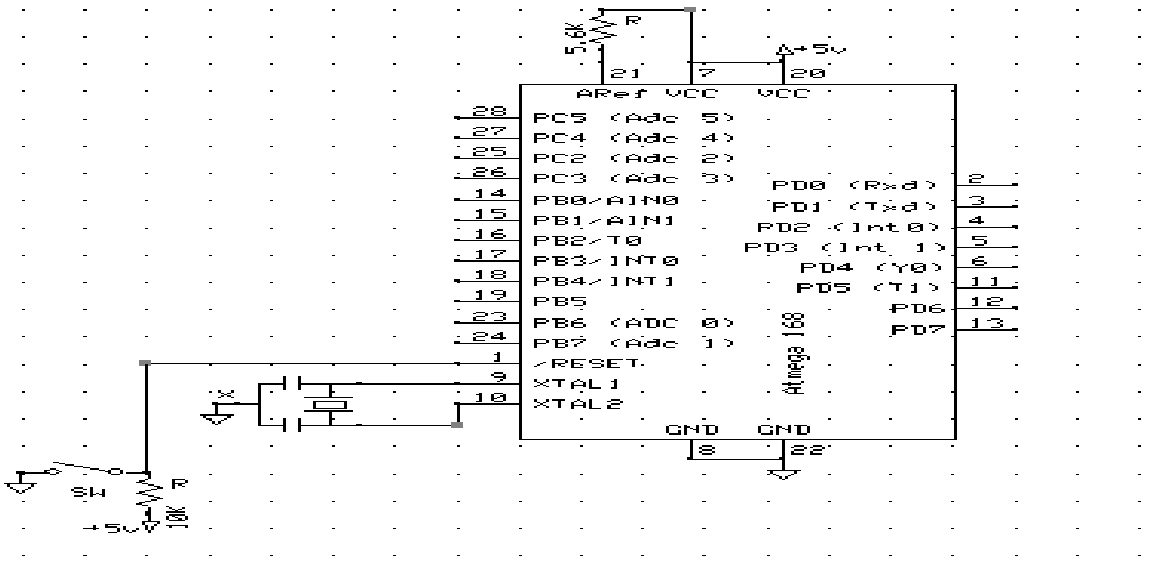


FIG. 2.1.3

1. THINGS TO REMEMBER ABOUT ATMEGA168 CONTROLLERS

The ATMEGA controllers are strong controllers but you have to take some small points in mind always like:

- When you go for the programming of atmega168 consider the pin no. as configures in red color in Pin diagram shown before (like controllers pin number 2 is digital pin number 0 for input or output ,pin23 is analog pin A0). Sao you will address the pin according to that number[7].

- Use the proper pin for proper input output interface that analog input should be configured at analog pin analog output should be configured on PWM pins and likewise the digital inputs and outputs.

CAPACITORS

INTRODUCTION

The function of capacitors is to store electricity, or electrical energy. The capacitor also functions as filter, passing AC, and blocking DC. The capacitor is constructed with two electrode plates separated by insulator. They are also used in timing circuits because it takes time for a capacitor to fill with charge. They can be used to smooth varying DC supplies by acting as reservoir of charge.

This symbol (⊣⊢)is used to indicate a capacitor in a circuit diagram. The capacitor is constructed with two electrode plates facing each other but separated by an insulator[4].

When DC voltage is applied to the capacitor, an electric charge is stored on each electrode. While the capacitor is charging up, current flows. The current will stop flowing when the capacitor has fully charged.

Breakdown Voltage

When using a capacitor, you must pay attention to the maximum voltage which can be used. This is the "breakdown voltage." The breakdown voltage depends on the kind of capacitor being used.

TYPES OF CAPACITORS

There are various types of capacitors available in the market. Some of them are as follows:

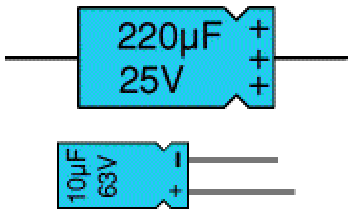
- Mica Capacitor
- Paper Capacitor
- Ceramic Capacitor
- Variable Capacitor
- Electrolytic Capacitor
- Tantalum Capacitor
- Film Capacitor

Here we used only two types of capacitor i.e. ceramic capacitor & electrolytic capacitor.

1. Polarized capacitors
2. Un-polarized capacitors

POLARIZED CAPACITORS

These are the capacitors having polarity. Basically these are of larger values than 1uf. For example below is the diagram of capacitor of 220 microfarad and having breakdown voltage 25V.

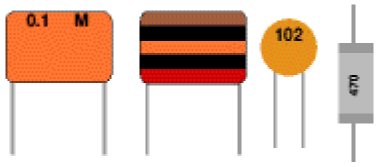


Electrolytic capacitors are polarized and they must be connected the correct way round, at least one of their leads will be marked + or -. They are not damaged by heat when soldering.

UNPOLARIZED CAPACITORS

Small value capacitors are un-polarized and may be connected either way round. They are not damaged by heat when soldering, except for one unusual type (polystyrene). They have high voltage ratings of at least 50V, usually 250V or so.

Many small value capacitors have their value printed but without a multiplier, so you need to use experience to work out what the multiplier should be.



For example 0.1 means 0.1µF = 100nF.

FIG. 2.2.2

VARIABLE CAPACITORS

Variable capacitors are mostly used in radio tuning circuits and they are sometimes called 'tuning capacitors'. They have very small capacitance values, typically between 100pF and 500pF (100pF = 0.0001µF).

FIG. 2.2.3

CRYSTAL OSCILLATOR

INTRODUCTION

A crystal oscillator is an electronic circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digital integrated circuits, and to stabilize frequencies for radio transmitters and receivers. The most common type of piezoelectric resonator used is the quartz crystal, so oscillator circuits designed around them were called "crystal oscillators".

A crystal oscillator is an electronic circuit that produces electrical oscillations at a particular designed frequency determined by the physical characteristics of one or more crystals, generally of quartz, positioned in the circuit feedback loop[9]. A piezoelectric effect causes a crystal such as quartz to vibrate and resonate at a particular frequency. Its fundamental frequency that can be hundreds of megahertz. The crystal oscillator is generally used in various forms such as a frequency generator, a frequency modulator and a frequency converter. Crystal oscillator circuits using crystal have a number of advantages in actual application since crystals show high frequency stability and stable temperature characteristic as well as excellent processing ability.

FIG. 2.3.1

COMONLY USED CRYSTAL FREQUENCIES

The Crystals can be manufactured for oscillation over a wide range of frequencies, from a few kilohertz up to several hundred megahertz. Many applications call for a crystal oscillator frequency conveniently related to some other desired frequency, so certain crystal frequencies are made in large quantities and stocked by electronics distributors.

CRYSTAL OSCILLATORS USED IN MICROCONTROLLERS

A microcontroller is disclosed that includes a crystal oscillator circuit that is programmable to provide multiple different levels of start-up current. In addition, the crystal oscillator circuit includes provision for selecting one of a plurality of different levels of capacitance. Furthermore, the crystal oscillator circuit includes a gate pass that includes circuitry for assuring predetermined start-up conditions are met. A feedback loop that includes an amplifier provides for steady-state operations that have low power consumption [10].

CRYSTAL OSCILATTORS OF DIFFERENT FREQUENCIES WITH USES

Frequency (MHz)	***Primary uses***
0.032768	Real-time clocks, quartz watches and clocks; allows binary division to 1 Hz signal (2^{15} × 1 Hz)
1.8432	UART clock; allows integer division to common baud rates. (= $2^{13} \times 3^{2} \times 5^{2}$. 16 × 115200 baud or 96 × 16 × 1200 baud)
2.4576	UART clock; allows integer division to common baud rates up to 38400
3.2768	*Allows binary division to 100 Hz (32768 × 100 Hz, or 2^{15} × 100 Hz)*
3.575611	PAL M color subcarrier
3.579545	NTSC M color subcarrier. Because these are very common and inexpensive they are used in many other applications, for example DTMF generators
3.582056	PAL N color subcarrier
3.6864	UART clock (2 × 1.8432 MHz); allows integer division to common baud rates
4.096	Allows binary division to 1 kHz (2^{12} × 1 kHz)

APPLICATIONS

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digital integrated circuits, and to stabilize frequencies for radio transmitters and receivers[10].

Virtually all microprocessors, micro-controllers, PICs and CPU's generally operate using a QUARTZ CRYSTAL OSCILLATOR as its frequency determining device to generate their clock waveform because as we already know, crystal oscillators provide the highest accuracy and frequency stability compared to resistor-capacitor(RC), inductor-capacitor (LC) oscillators. Quartz crystals are manufactured for frequencies from a few kilohertz up to several hundred Megahertz.

FIG. 2.3.2

DS18B20 TEMPERATURE SENSOR

INTRODUCTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems[4][5].

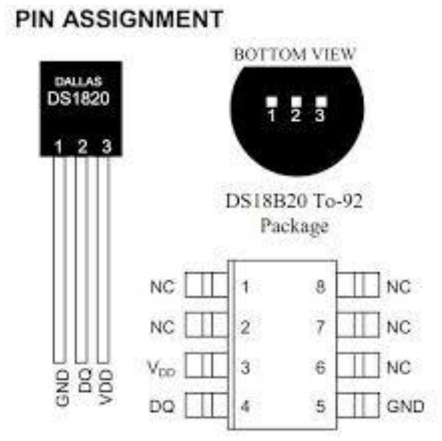


FIG. 2.4.1

KEY FEATURES

- Unique 1-Wire Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multi drop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- Nonvolatile Alarm Settings

- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System.

OPERATION MEASURING TEMPERATURE

The core functionality of the DS18B20 is its direct-to - digital temperature sensor. The resolution of the temperature sensor is user - configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5 °C, 0.25 °C, 0.125°C, and 0.0625°C, respectively. The default resolution at power- up is 12- bit. The DS18B20 powers up in a low - power idle state. To initiate a temperature measurement and A - to - D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2 - byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue "read time slots" (see the 1- Wire Bus System section) after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done[9][1]. If the DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pull-up during the entire temperature conversion.

The DS18B20 output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16- bit sign- extended two's complement number in the temperature register. The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. If the DS18B20 is configured for 12- bit resolution, all bits in the temperature register will contain valid data. For 11 - bit resolution, bit 0 is undefined. For 10 - bit resolution, bits 1 and 0 are undefined, and for 9- bit resolution bits 2, 1, and 0 are undefined[10].

INTERFACE TO CONTROLLER

The temperature sensor should be interface on one-wire communication pin of controller. The figure shows the interface.

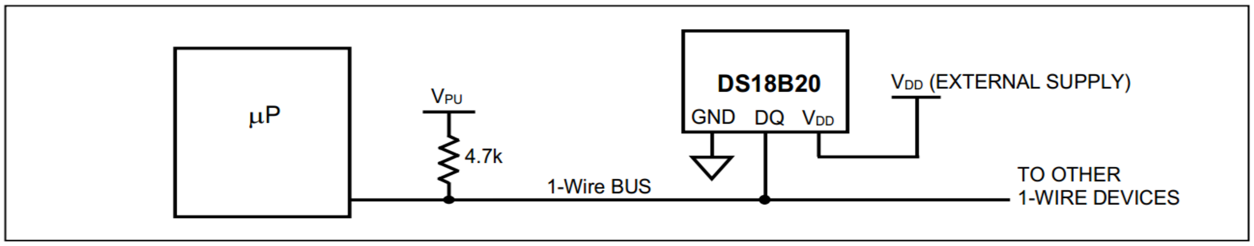


FIG. 2.4.2

APPLICATIONS

- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System
- Agricultural Equipment
- Audio Equipment
- Automotive
- Climate Control
- GPS Devices
- Hard Disk Drive
- Medical Equipment
- Set-Top Boxes
- Telecommunications.

EXPRESS PCB

INTRODUCTION

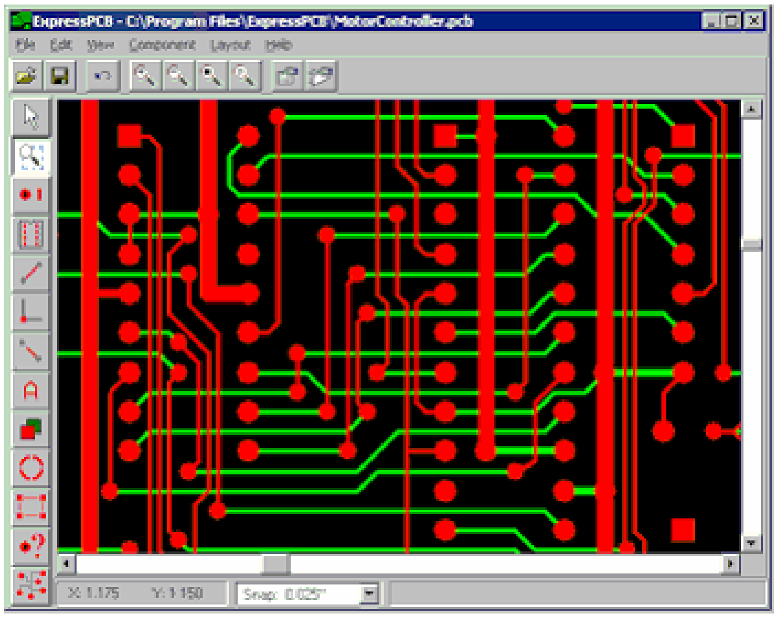
A PCB layout is required to place components on the PCB so that the component area can be minimized and the components can be placed in an efficient manner. The components can be placed in two ways, either manually or by software. The manual procedure is quiet cumbersome and is very inefficient. The other method is by the use of computer software. This method is advantageous as it saves time and valuable copper area[3]. There are various software's available for this purpose like-

- Express PCB
- Pad2pad
- Protel PCB
- PCB design

Many of them are loaded with auto routing and auto placement facility. The software that we have used here is **EXPRESS PCB.** This software has a good interface, easy editing options and a wide range of components.

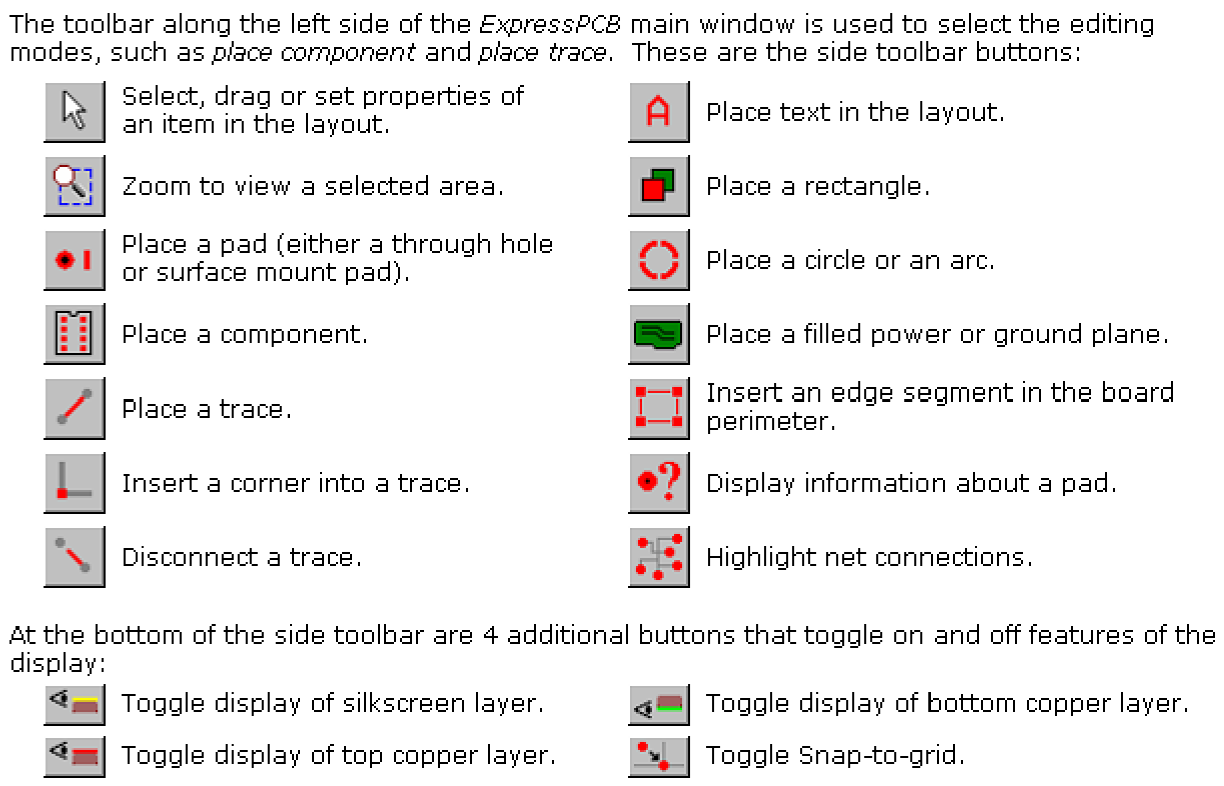
Express P.C.B.

Express PCB is a very easy to use Windows application for laying out printed circuit boards. There are two parts to Express PCB, Express SCH for drawing schematics and Express PCB for designing circuit boards. The software can be downloaded from the website www.expresspcb.com.


ExpressPCB - C:\Program Files\ExpressPCB\MotorController.pcb
File Edit View Component Layout Help
X: 1.175 Y: 1.150
Snap: 0.025"

There are lots of functions available in the software. This software is free of cost and also it is very easy to use. The different layers of the PCB can be viewed by just a click of a button on the interface. And we easily get its print on paper which is utilized for further processing. We can design single sided PCB as well as Double Sided PCB with this Software

THE SIDE TOOLBAR



The toolbar along the left side of the *ExpressPCB* main window is used to select the editing modes, such as *place component* and *place trace*. These are the side toolbar buttons:

Select, drag or set properties of an item in the layout.	Place text in the layout.
Zoom to view a selected area.	Place a rectangle.
Place a pad (either a through hole or surface mount pad).	Place a circle or an arc.
Place a component.	Place a filled power or ground plane.
Place a trace.	Insert an edge segment in the board perimeter.
Insert a corner into a trace.	Display information about a pad.
Disconnect a trace.	Highlight net connections.

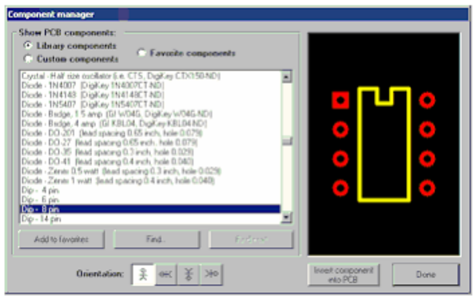
At the bottom of the side toolbar are 4 additional buttons that toggle on and off features of the display:

Toggle display of silkscreen layer.	Toggle display of bottom copper layer.
Toggle display of top copper layer.	Toggle Snap-to-grid.

FIG. 2.5.2

BEGINNING A NEW LAYOUT

- Begin a new layout by running Express PCB.
- In the main window, the yellow rectangle defines the perimeter of the PC board. Set the size of your board by moving three of its four corners (the upper left corner is fixed at 0, 0). Move the corners by dragging them with the mouse, or by double-clicking them and entering coordinates.



[✓ Library components] Custom components Favorite components

```
Crystal - Half size oscillator (i.e. CTS, DigiKey CTX150-ND)
Diode - 1N4007  [DigiKey 1N4007CT-ND]
Diode - 1N4148  [DigiKey 1N4148CT-ND]
Diode - 1N5407  [DigiKey 1N5407CT-ND]
Diode - Bridge, 1.5 amp  (GI W04G, DigiKey W04G-ND)
Diode - Bridge, 4 amp  (GI KBL04, DigiKey KBL04-ND)
Diode - DO-201  (lead spacing 0.65 inch, hole 0.079)
Diode - DO-27  (lead spacing 0.65 inch, hole 0.079)
Diode - DO-35  (lead spacing 0.3 inch, hole 0.029)
Diode - DO-41  (lead spacing 0.4 inch, hole 0.040)
Diode - Zener 0.5 watt  (lead spacing 0.3 inch, hole 0.029)
Diode - Zener 1 watt  (lead spacing 0.4 inch, hole 0.040)
Dip - 4 pin
Dip - 6 pin
Dip - 8 pin
Dip - 14 pin
```

- **Zooming and Panning:** The easiest way to move around your layout is with the scroll wheel on the mouse. Turning the wheel zooms in and out. Pressing the wheel and dragging the mouse pans.

PLACING THE COMPONENT

The easiest way to place components in your board layout is to use the Component Manager. Tip: Use the Find button to search for a PCB footprint by its name. To place a component:

1. Click the button located on the top toolbar to display the Component Manager.
2. Select one of these categories:· Library components - Components that are included with the program· Custom components - Components that you have drawn· Favorite components - Components or symbols that you have book-marked
3. From the list box, choose the item to insert then select the component's orientation (rotated up, left, down, or right) by clicking one of these buttons:
4. Press the Insert component into PCB button, then drag the component to the desired location.
5. Assign the component's Part ID (such as R1 or U2). To do this, select then double-click on the component to display its Component properties dialog box[4].

PLACING THE PADS

Individual pads are used both to build new components and as via. To insert a pad:

1. From the side toolbar, select or press the P shortcut key.
2. Select a pad size from the drop down list box on the top toolbar. There are several pad types: round, square, surface mount and via.
3. To place the pad, click on your layout at the desired location.

COMPONENT PLACEMENT

These are the steps to add traces to your layout. If you have drawn a schematic for your circuit, be sure to read the section Linking the Schematic and PCB before you begin.

1. From the side toolbar, select or press the T shortcut key.
2. Select the trace width from the drop down list box on the top toolbar: A width of 0.010" is a good default for digital and analog signals. For power lines, use traces 0.05" or wider.
3. Select the layer by clicking or by pressing the L shortcut key.
4. Move the mouse to the trace's first end point and click. Drag the trace to the second end point, and then click again. Continue placing trace segments (use the L key to change layers) until you have reached the final end point.

   Tip: Keep an eye on the status bar when connecting traces to components. It will display the pin and part number to which the trace is connected.
5. As you drag the trace, the L key changes layers, the Del key deletes the previous segment, the + and - keys zoom in and out, the G key toggles the snap-to-grid on and off. The Spacebar sets the trace, and the Esc key cancels it.
6. To complete the operation, press the Spacebar or click right.

WORKING WITH TRACES

Traces themselves cannot be moved. The path of a trace is determined by the straight line between its two connections. Therefore, to move a trace, you need to connect it to something different or to move what it is connected to. Corners in traces allow them to bend. They are displayed as small square blocks at the ends of traces. In your final PC board layout, the corners will not be included.

Corners can be dragged, inserted or deleted to change the trace's path. To insert a corner, select the required tab then click to choose the layer on which the corner will be placed then click on the trace at the point where you want to insert the corner. To disconnect a trace and reconnect it elsewhere, select desired tab from the side toolbar. Next, click on the trace near the point you want to disconnect, and then drag the trace to a new pin.

PLACING FILLED PLANE

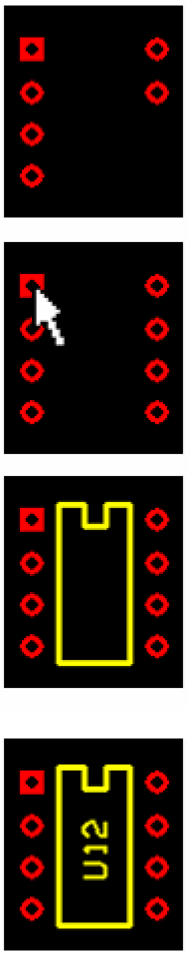
Filled Planes are used to add ground or power planes to a circuit, usually on double-sided boards. They can be placed on the top or bottom layers. The perimeter can have the shape of any polygon and the interior is automatically insulated from traces and pads.

To place a Filled Plane, select the proper tab , then on the top toolbar choose the Draw filled plane option, along with the layer. Next use the mouse to draw the perimeter. Do this by clicking the left mouse button at each corner, then clicking right after you have placed the last corner. Connect a pad to a plane by right clicking on the pad. In the popup menu, select one of the Top layer pad shape or Bottom layer pad shape options[7].

COPYING DELETING AND MOVING THE ITEMS

Express PCB uses standard Edit commands for Copy, Cut and Paste. To copy an item or several items, first select them. From the Edit menu choose Copy, and then from the Edit menu choose Paste. Deleting items from your layout is as easy as selecting them and pressing the Del key. There are three ways most items can be moved. Typically, items can be moved by selecting and dragging them. They can also be moved with the arrow keys by selecting them and then pressing Ctrl-right, Ctrl-left, Ctrl-up or Ctrl-down. Alternately, an item can be moved by changing the coordinates set in its properties dialog box[3][5].

Tip: To copy a group of items from one layout file to another, select and copy the items of interest into the clipboard. Load the second file into Express PCB and paste. It is not possible to copy items between two programs running at the same time.



MAKING CUSTOM COMPONENT:

1. Express PCB includes many components that you can use to create your PCB. However if you need a footprint not found in our library, you can easily build your own. Here is how:

FIG. 2.5.4

2. First add the pads by selecting and choosing a pad type on the top toolbar. Carefully position the pads with the correct spacing. It may be helpful to change the Snap-to-grid spacing in the Options dialog. Tip: If you cannot find a pad the size you need, create a custom pad.
3. Assign each pad a Pin number. This must be done if you link your PCB to its schematic. Assign pin numbers by selecting and then double clicking on each pad to display its Pad Properties dialog box. In the Pin number field, enter the pin number[2].
4. Draw the component outline for the new part on the silkscreen layer. Draw straight lines by selecting or circles and arcs using. Be sure to select the silkscreen layer before drawing the outline by clicking. The recommended line width for drawing component outlines is 0.012". Check the concerned tab in the toolbar.

\

16X2 LCD

INTRODUCTION

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD[8].

Features

- 5 x 8 dots with cursor
- Built-in controller (KS 066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

PIN DESCRIPTION

PIN NUMBER	SYMBOL	FUNCTION
1	V_{SS}	GND
2	V_{DD}	+ 3V or + 5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/W	H/L Read/Write Signal
6	E	H?L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/V_{EE}	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (OV)

LCD INTERFACE DIAGRAM

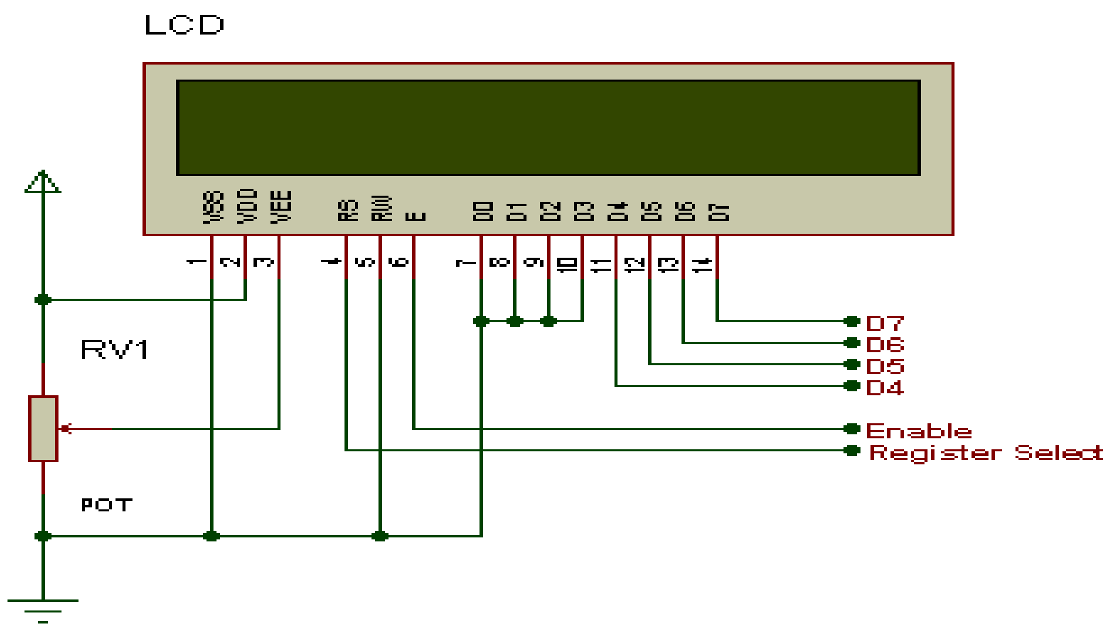


FIG. 2.6.1

Above is the connection diagram of LCD in 4-bit mode, where we only need 6 pins to interface an LCD. D4-D7 is the data pins connection and Enable and Register select are for LCD control pins. We are not using Read/Write (RW) Pin of the LCD, as we are only writing on the LCD so we have made it grounded permanently. If you want to use it, then you may connect it on your controller but that will only increase another pin and does not make any big difference. Potentiometer RV1 is used to control the LCD contrast. The unwanted data pins of LCD i.e. D0-D3 are connected to ground[5].

USB TO UART MODULE

INTRODUCTION

USB to RS232 TTL Module contains a CP2102 single-chip USB to UART bridge which converts data traffic between USB and UART formats. The chip includes a complete USB 2.0 full-speed function controller; bridge control logic and a UART interface with transmit/receive buffers and modem handshake signals. The Module interfaces to microcontrollers through a 6 Pin 2.54mm single row pin header interface can easily be connected to any microcontroller or FPGA and DSPs.

FEATURES

* Brand new and high quality.
* Stable and reliable chipset CP2102.
* USB specification 2.0 compliant with full-speed 12Mbps.
* Standard USB type A male and TTL 6pin connector.
* 6 pins for 3.3V, RST, TXD, RXD, GND & 5V.
* All handshaking and modem interface signals.
* Baud rates: 300 bps to 1.5 Mbps.
* Byte receive buffer: 640 byte transmit buffer.
* Hardware or X-On/X-Off handshaking supported.
* Event character support Line break transmission.
* USB suspend states supported via SUSPEND pins.
* Temperature range: -40°C to +85°C.
* Supports Windows 98SE, 2000, XP, Mac OS X & Linux.
* Size: 42mm x 15mm.

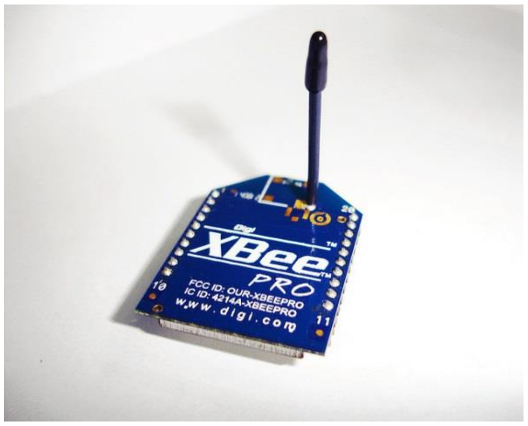
APPLICATIONS

- Microcontroller Board USB Interface
- ARDUINO Projects
- PND (Portable Navigation Device)
- GPS mouse and Bluetooth GPS receiver
- CPLD and FPGA Interfacing.

X-BEE TECHNOLOGY AND MODULE

INTRODUCTION

The X-Bee RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



This is wireless module permits long distance transmission as compared with normal RF Module. It works in full duplex mode that each x-bee module is transmitter as well as receiver .One more advantage of these modules that they are interference free that is we can use multiple X-bee transmitter and receiver of same frequency in same region. However use multiple receiver are possible in Normal RF but we cannot use multiple transmitter in same RF range[7][10].

KEY FEATURES

1. **Long Range Data Integrity**
    - Indoor/Urban: up to 300’ (90 m), 200' (60 m) for International variant
    - Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
    - Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
    - Receiver Sensitivity: -100 dBm
    - RF Data Rate: 250,000 bps

2. **Power Requirements (Low Power)**
    - Supply voltage : 2.8 – 3.4 V
    - TX Peak Current: 250mA (150mA for international variant
    - RX Current: 55 mA (@3.3 V)

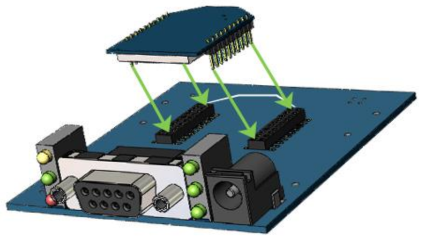
2. **ADC and I/O line support**
    - Analog-to-digital conversion, Digital I/O Line Passing

3. **Advanced Networking & Security**
    - Retries and Acknowledgements
    - DSSS (Direct Sequence Spread Spectrum)
    - Each direct sequence channels has over 65,000 unique network addresses available
    - Source/Destination Addressing
    - Uni-cast & Broadcast Communications Point-to-point, point-to-multipoint and peer-to-peer topologies supported
4. **Easy-to-use**
    - No configuration necessary for out-of box RF communications
    - Free X-CTU Software (Testing and configuration software)
    - AT and API Command Modes for configuring module parameters
    - Extensive command set
    - Small form factor

5. **Worldwide Acceptance**
    - Systems that contain X-Bee®/X-Bee-PRO® RF Modules inherit Digi Certifications.
    - ISM (Industrial, Scientific & Medical) 2.4 GHz frequency band
    - Manufactured under **ISO 9001:2000** registered standards

6. **General**
    - Operating Frequency   ISM 2.4 GHz
    - Operating Temperature : -40 to 85º C (industrial)
    - Antenna Options: Integrated Whip, Chip or U.FL Connector, RPSMA Connector.

MOUNTING CONSIDERATIONS



The X-Bee RF Module was designed to mount into a receptacle (socket) and there-fore does not require any soldering when mounting it to a board. The X-Bee Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

OPERATION

Serial Communications: The X-bee RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Digi proprietary RS-232 or USB interface board).

UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

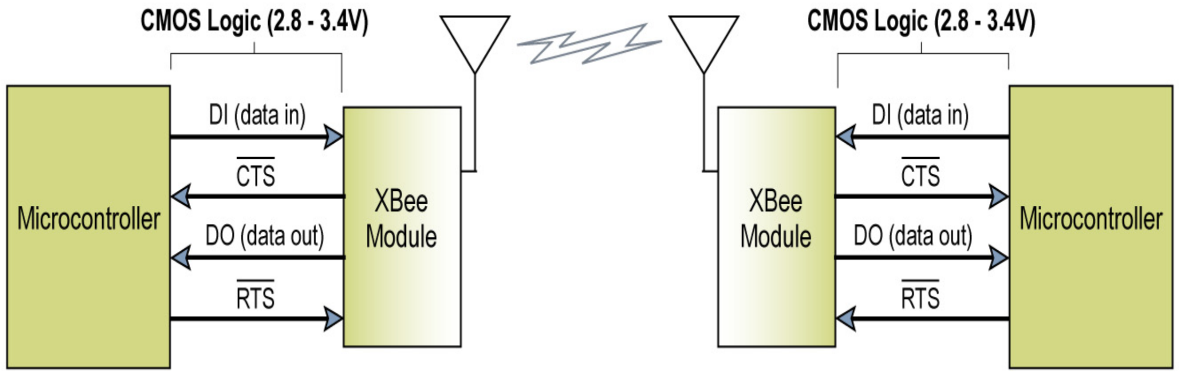


FIG. 2.8.2

Serial Data

Data enters the module UART through the DI pin (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted. Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

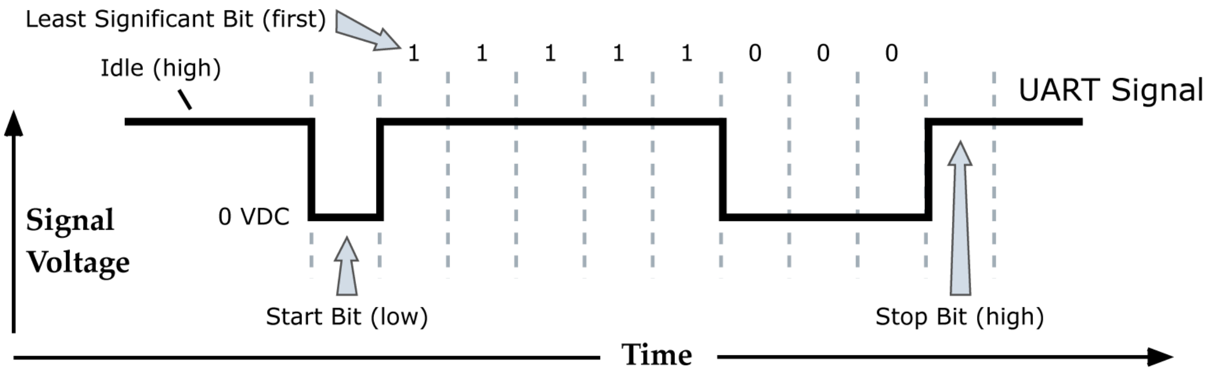


FIG. 2.8.3

UART data packet 0x1F as transmitted through the RF module

Serial communications depend on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits). The UART baud rate and parity settings on the X-bee module can be configured with the BD and SB commands, respectively[6].

Transparent Operation

By default, X-bee RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement. All UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin[1][3][4].

API (APLLICATION PROGRAMMING INTERFACE)

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

- Transmit Data Frames (received through the DI pin (pin 3)) include:
- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)
- Receive Data Frames (sent out the DO pin (pin 2)) include:
- RF-received data frame

- Command response
- Event notifications such as reset, associate, disassociate, etc.

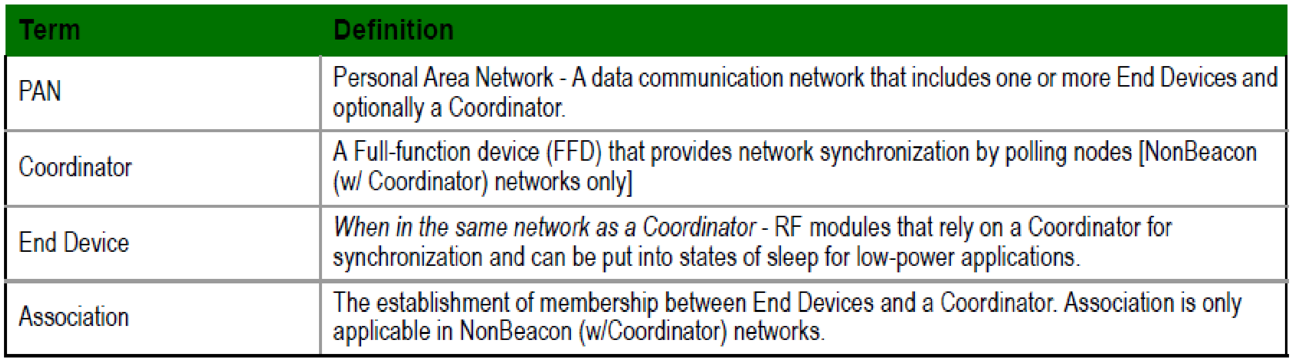
The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets[10].

The API operation option facilitates many operations such as the examples cited below:

- Transmitting data to multiple destinations without entering Command Mode
- Receive success/failure status of each transmitted RF packet
- Identify the source address of each received packet.

X-BEE NETWORKS

The following terms will be used to explicate the network operations:



Term	Definition
PAN	Personal Area Network - A data communication network that includes one or more End Devices and optionally a Coordinator.
Coordinator	A Full-function device (FFD) that provides network synchronization by polling nodes [NonBeacon (w/ Coordinator) networks only]
End Device	*When in the same network as a Coordinator* - RF modules that rely on a Coordinator for synchronization and can be put into states of sleep for low-power applications.
Association	The establishment of membership between End Devices and a Coordinator. Association is only applicable in NonBeacon (w/Coordinator) networks.

Antennas (X-Bee-PRO RF Module)

The following antenna types have been tested and approved for use with the X-Bee Module:

Antenna Type: Yagi RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss.

Antenna Type: Omni-directional RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss (EIRP Maximum of 14 dBm).

Antenna Type: Flat Panel RF module was tested and approved with 19 dBi antenna gain with 4.8 dB cable-loss.

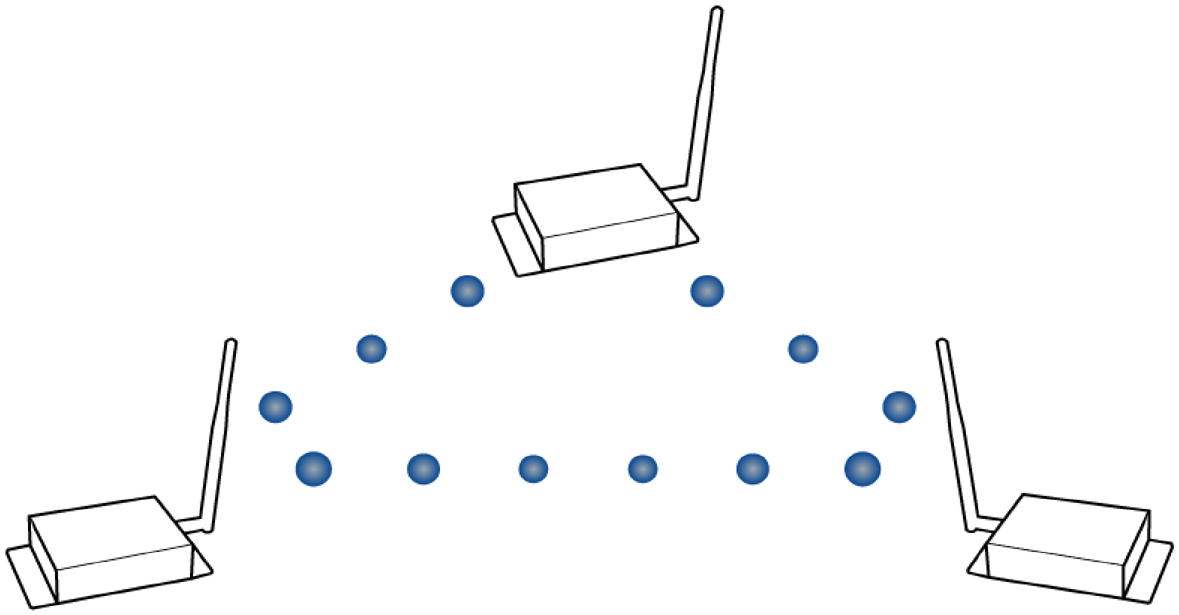


FIG. 2.8.4

MODES OF OPERATION

X-Bee RF Modules operate in five modes.

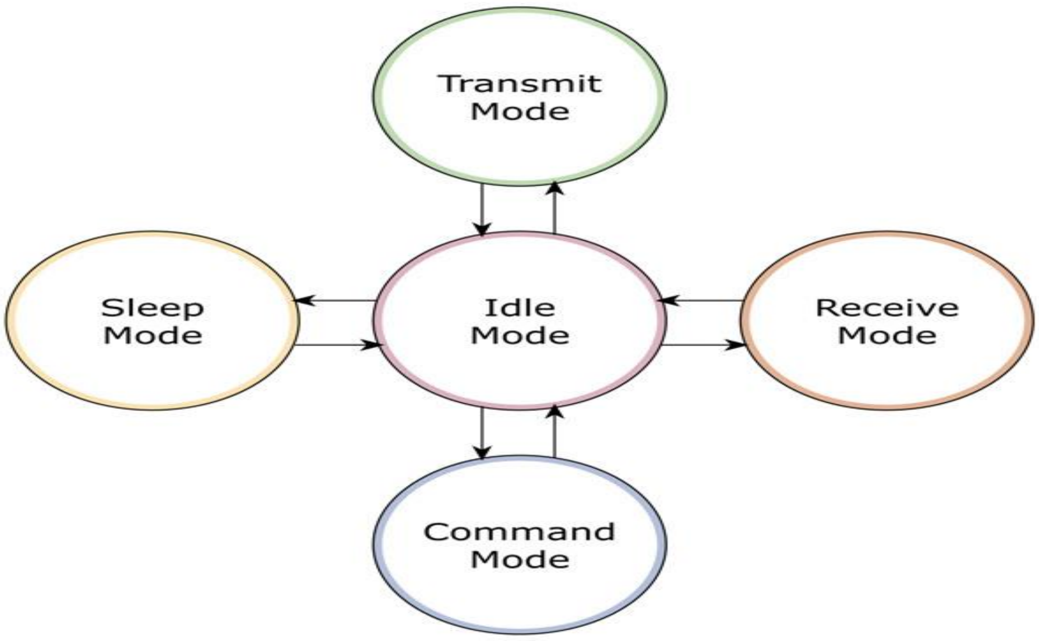


FIG. 2.8.5

Modes of Operation

1. **Idle Mode**

   When not receiving or transmitting data, the RF module is in Idle Mode. The module shifts into the other modes of operation under the following conditions:

   • Transmit Mode (Serial data is received in the DI Buffer)

   • Receive Mode (Valid RF data is received through the antenna)

   • Sleep Mode (Sleep Mode condition is met)

   • Command Mode (Command Mode Sequence is issued)

2. **Transmit/Receive Modes**

   RF Data Packets

   Each transmitted data packet contains a Source Address and Destination Address field. The Source Address matches the address of the transmitting module as specified by the MY (Source Address) parameter (if MY >= 0xFFFE), the SH (Serial Number High) parameter or the SL (Serial Number Low) parameter. The <Destination Address> field is created from the DH (Destination Address High) and DL (Destination Address Low) parameter values. The Source Address and/or Destination Address fields will either contain a 16-bit short or long 64-bit long address[5][7].

The RF data packet structure follows the 802.15.4 specification.

3. Direct and Indirect Transmission

There are two methods to transmit data:

• Direct Transmission - data is transmitted immediately to the Destination Address

• Indirect Transmission - A packet is retained for a period of time and is only transmitted after the destination module (Source Address = Destination Address) requests the data.

Indirect Transmissions can only occur on a Coordinator. Thus, if all nodes in a network are End Devices, only Direct Transmissions will occur. Indirect Transmissions are useful to ensure packet delivery to a sleeping node. The Coordinator currently is able to retain up to 2 indirect messages.

Direct Transmission A Coordinator can be configured to use only Direct Transmission by setting the SP (Cyclic Sleep Period) parameter to "0". Also, a Coordinator using indirect transmissions will revert to direct transmission if it knows the destination module is awake[5].

To enable this behavior, the ST (Time before Sleep) value of the Coordinator must be set to match the ST value of the End Device. Once the End Device either transmits data to the Coordinator or polls the Coordinator for data, the Coordinator will use direct transmission for all subsequent data transmissions to that module address until ST time occurs with no activity (at which point it will revert to using indirect transmissions for that module address). "No activity" means no transmission or reception of messages with a specific address. Global messages will not reset the ST timer.

Indirect Transmission To configure Indirect Transmissions in a PAN (Personal Area Network), the SP (Cyclic Sleep Period) parameter value on the Coordinator must be set to match the longest sleep value of any End Device. The sleep period value on the

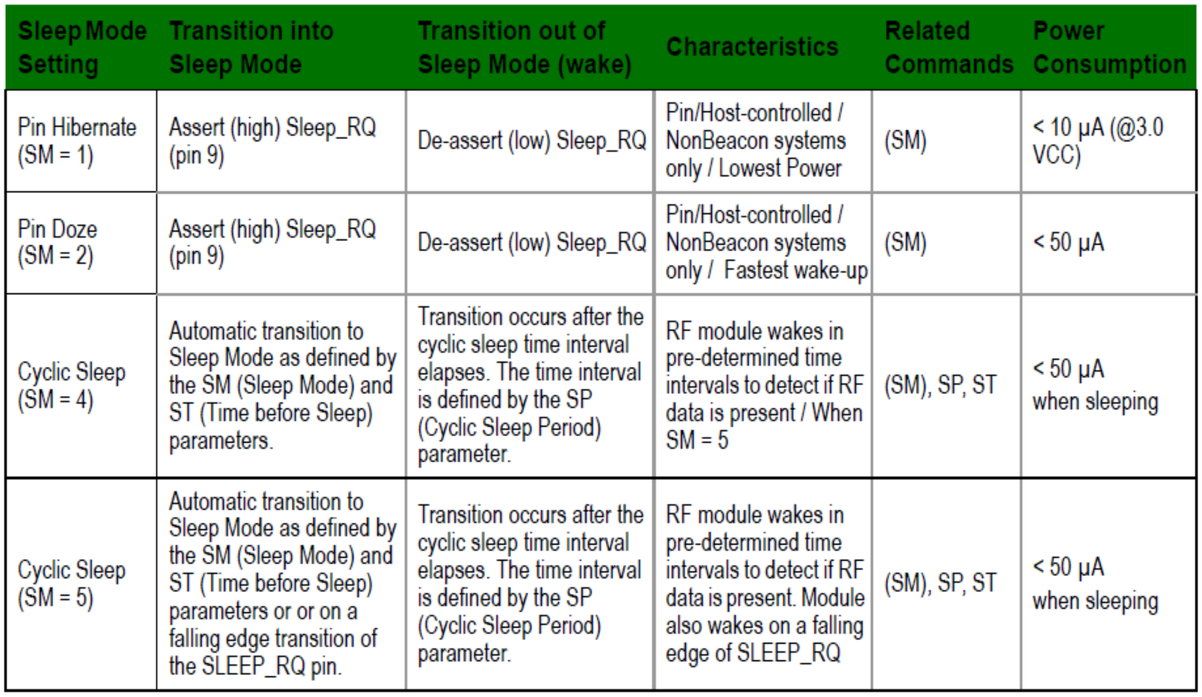
Coordinator determines how long (time or number of bea-cons) the Coordinator will retain an indirect message before discarding it. An End Device must poll the Coordinator once it wakes from Sleep to determine if the Coordinator has an indirect message for it. For Cyclic Sleep Modes, this is done automatically every time the module wakes (after SP time). For Pin Sleep Modes, the A1 (End Device Association) parameter value must be set to enable Coordinator polling on pin wake-up. Alternatively, an End Device can use the FP (Force Poll) command to poll the Coordinator as needed[4][3].

4. Sleep Mode

Sleep Modes enable the RF module to enter states of low-power consumption when not in use. In order to enter Sleep Mode, one of the following conditions must be met (in addition to the module having a non-zero SM parameter value):

• Sleep_RQ (pin 9) is asserted and the module is in a pin sleep mode (SM = 1, 2, or 5)

• The module is idle (no data transmission or reception) for the amount of time defined by the ST (Time before Sleep) parameter.



Sleep Mode Setting	Transition into Sleep Mode	Transition out of Sleep Mode (wake)	Characteristics	Related Commands	Power Consumption
Pin Hibernate (SM = 1)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Lowest Power	(SM)	< 10 µA (@3.0 VCC)
Pin Doze (SM = 2)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Fastest wake-up	(SM)	< 50 µA
Cyclic Sleep (SM = 4)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present / When SM = 5	(SM), SP, ST	< 50 µA when sleeping
Cyclic Sleep (SM = 5)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters or or on a falling edge transition of the SLEEP_RQ pin.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present. Module also wakes on a falling edge of SLEEP_RQ	(SM), SP, ST	< 50 µA when sleeping

Sleep Mode Configurations

5. **Command Mode**

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming characters are interpreted as commands. Two Command Mode options are supported:

- AT Command Mode
- API Command Mode

AT Command Mode

To Send AT Commands:

Send AT commands and parameters using the syntax shown below.

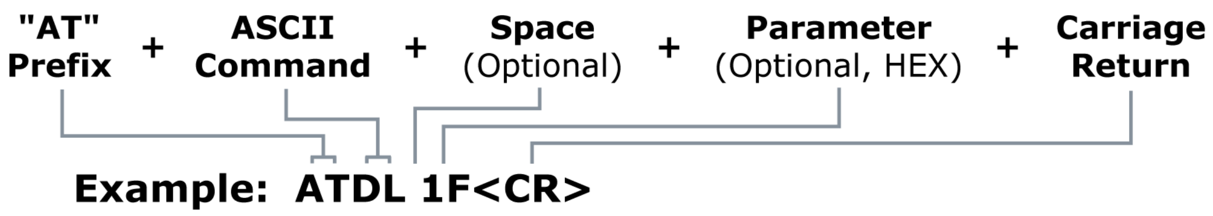


FIG. 2.8.7

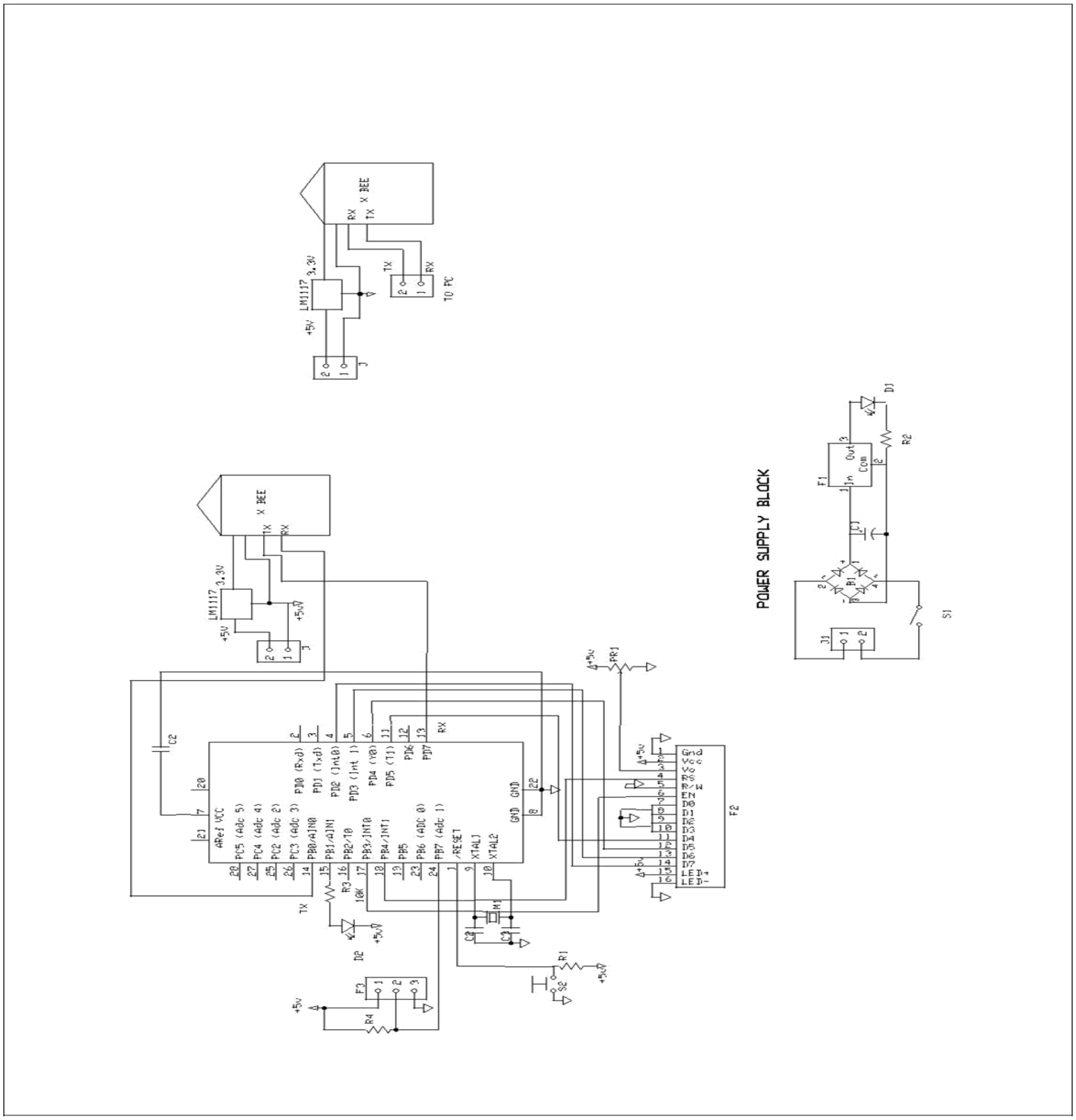
Chapter-3

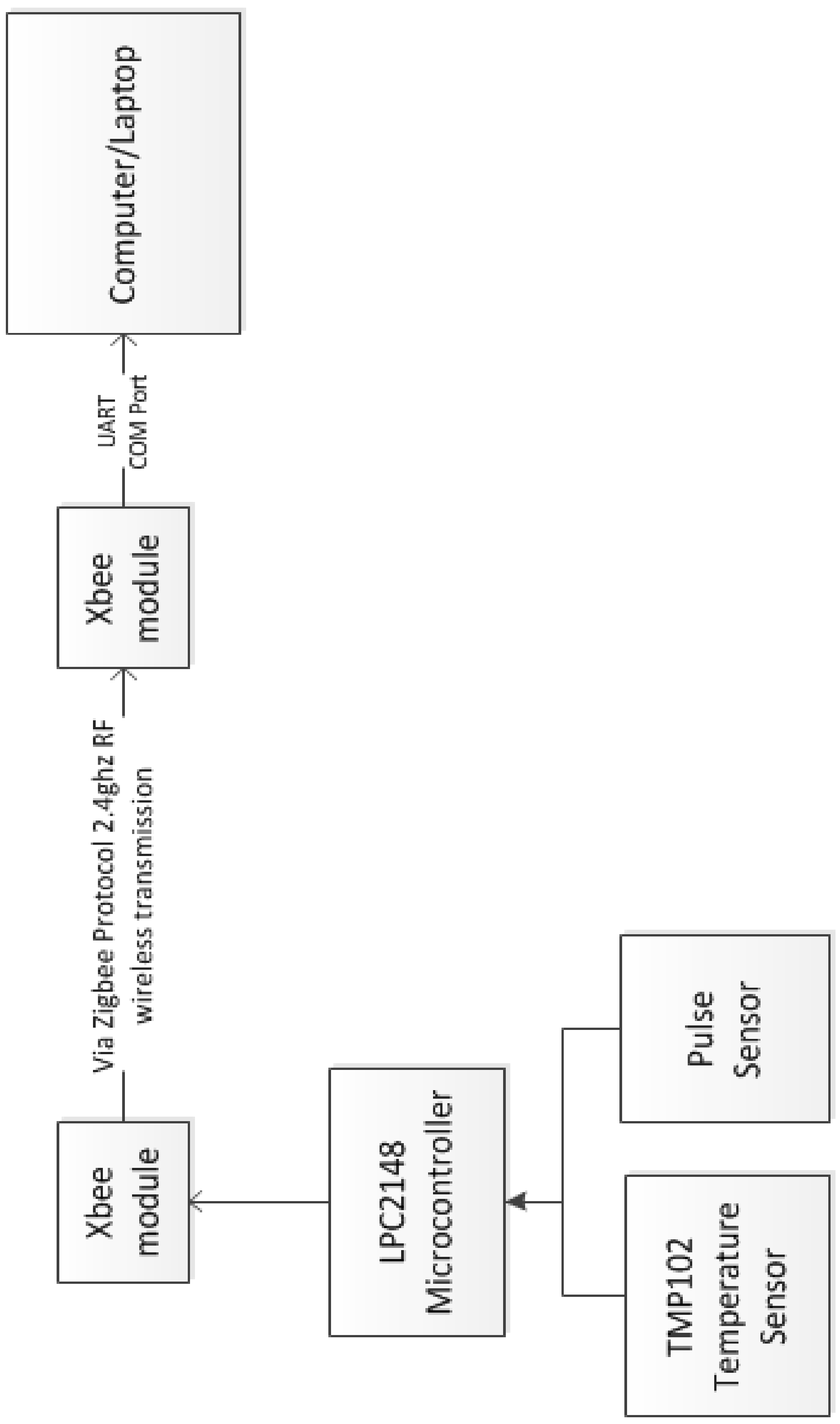
Description of Project

Wireless sensing technology can collect pressure, temperature, and flow measurements in remote and often unsafe locations that is common in the offshore/on land oil and gas industry without cables and the associated problems. However, the severe offshore conditions make it necessary to develop reliable and cost-effective real-time monitoring structures when building offshore control and monitoring systems. Nowadays, ZigBee RF standard is deployed. This has opened new perspectives for wireless control networks. ZigBee is powerful and easy to install because it was developed in order to be installed in a new or existing sensor network. RASHPETCO has many geographically distributed offshore platforms. Some types of ZigBee radio modules provide a transmission range over 40 km with multi-hop communication capability to extend the coverage. Many oil and gas applications, where ZigBee technology can be deployed, can be overseen[2][8].

This is wireless module permits long distance transmission as compared with normal RF Module. It works in full duplex mode that each x-bee module is transmitter as well as receiver .One more advantage of these modules that they are interference free that is we can use multiple X-bee transmitter and receiver of same frequency in same region. However use multiple receiver are possible in Normal RF but we cannot use multiple transmitter in same RF range[6].

Circuit Diagram





Block Diagram

Working

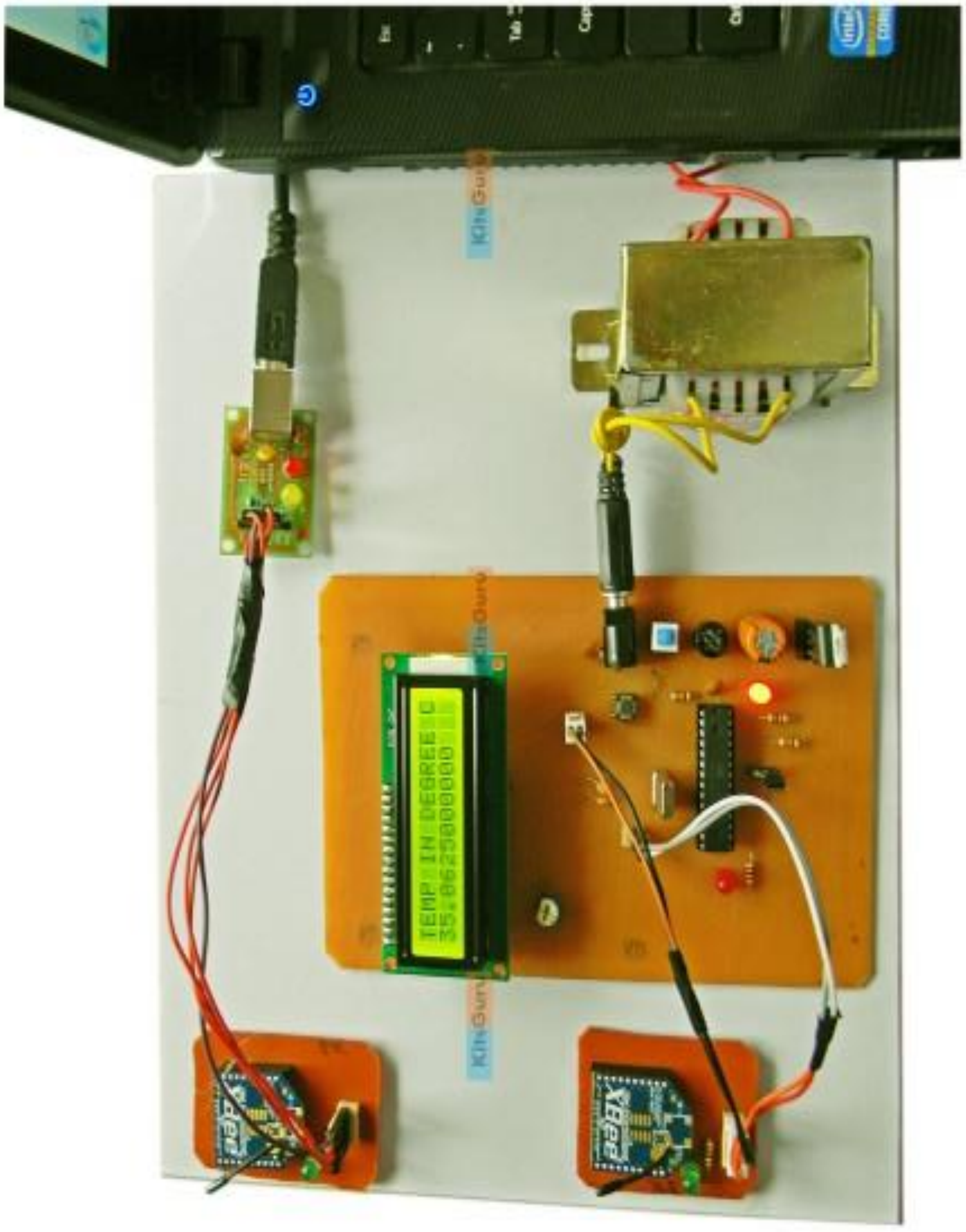


FIG. 3.3

POWER SUPPLY: The input is 230V AC which is step down using the transformer (12-0-12) .The 12V ac input is fed to the bridge diode to gives 12V pulsating DC. This DC voltage is filtered through the capacitor to remove the ripples. The filtered DC is fed to 7805 regulator to fetch +5v regulated output. This regulated voltage is given to all the components to function properly.

Here we are monitoring the temperature on computer wirelessly using x bee. The data is controlled in full duplex mode wirelessly using x bee wireless modules. As we send the data from the controller, it goes to the x bee module. Another x bee module is mounted with the computer. This x bee connected with the pc using an USB to serial converter such as FTDI. To see the data on the computer, we are using Dock light software to monitor temp. Here temperature sensor is DB18B20 that works on one wire protocol. The microcontroller keeps on reading data from sensor and will keep on transmitting. Different power supply is given to different units, for example microcontroller is given 5 v power supplies to operate using 7805 voltage regulator. X-bee modules works on 3.3v power supply; to provide 3.3V we are using LM1117[4].

TESTING

Connect the black terminal of the Digital Multimeter to the ground of the supply source and turn the knob to 20V DC voltage.

1. Make sure the notch of all the IC’s including microcontroller is same as given above.

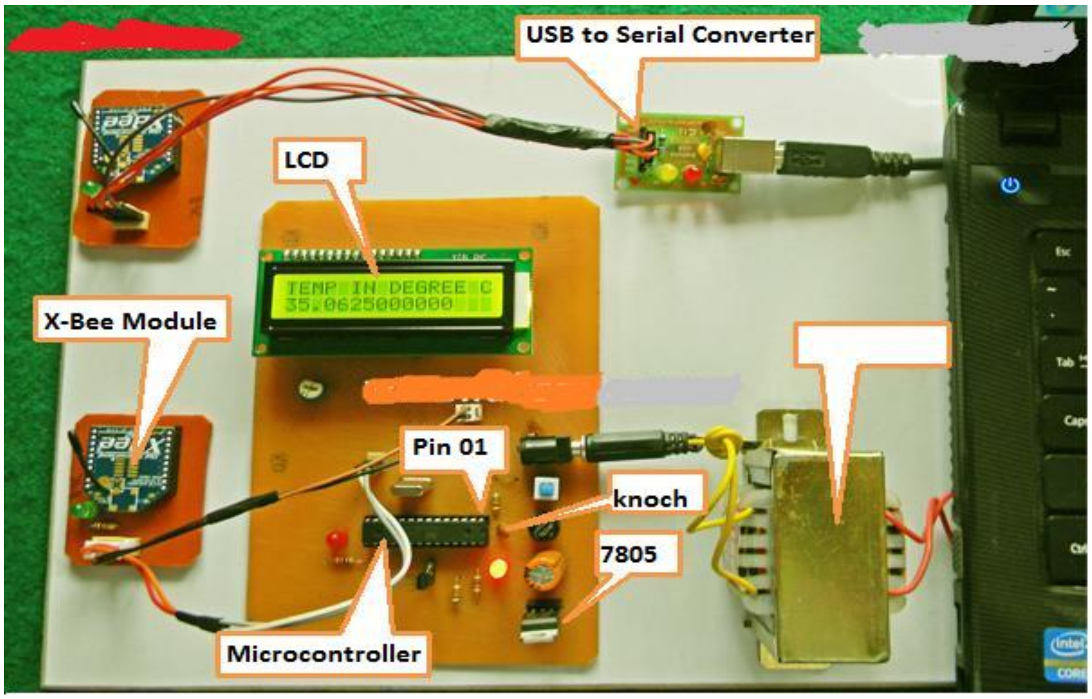


FIG.3.4

2. Check the continuity and short circuit of the PCB's.
3. Check the voltages at pin o (output) of 7805 and it should be +5 volts.
4. Check the voltage at pin 7 and 20 of the microcontroller it should be +5 volts.
5. Check the UART connector and see RX and TX.
6. Connect to TX of PCB with RX of UART.
7. Connect to RX of PCB with TX of UART.
8. Check the voltage at pin 2 and pin 15 of the LCD, It should be +5V
9. LCD should display all the messages.

Testing of Serial Port Hardware & Software:

Step 1: Testing of USB to UART device

1. See the name of USB to UART device.(eg: FTDI converter)
2. Download and install the driver in your computer if it is not provided to you.
3. Restart the computer.

Connect the device with the computer.

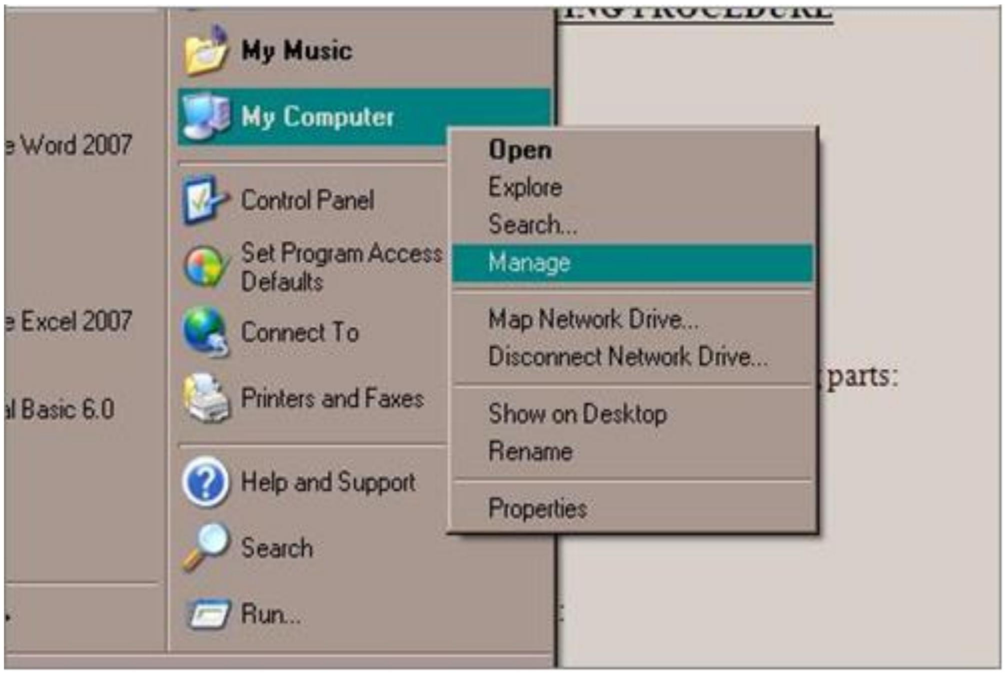


FIG. 3.5

5. Right click on the my computer icon as shown below.
6. Click on the device manager as shown below.

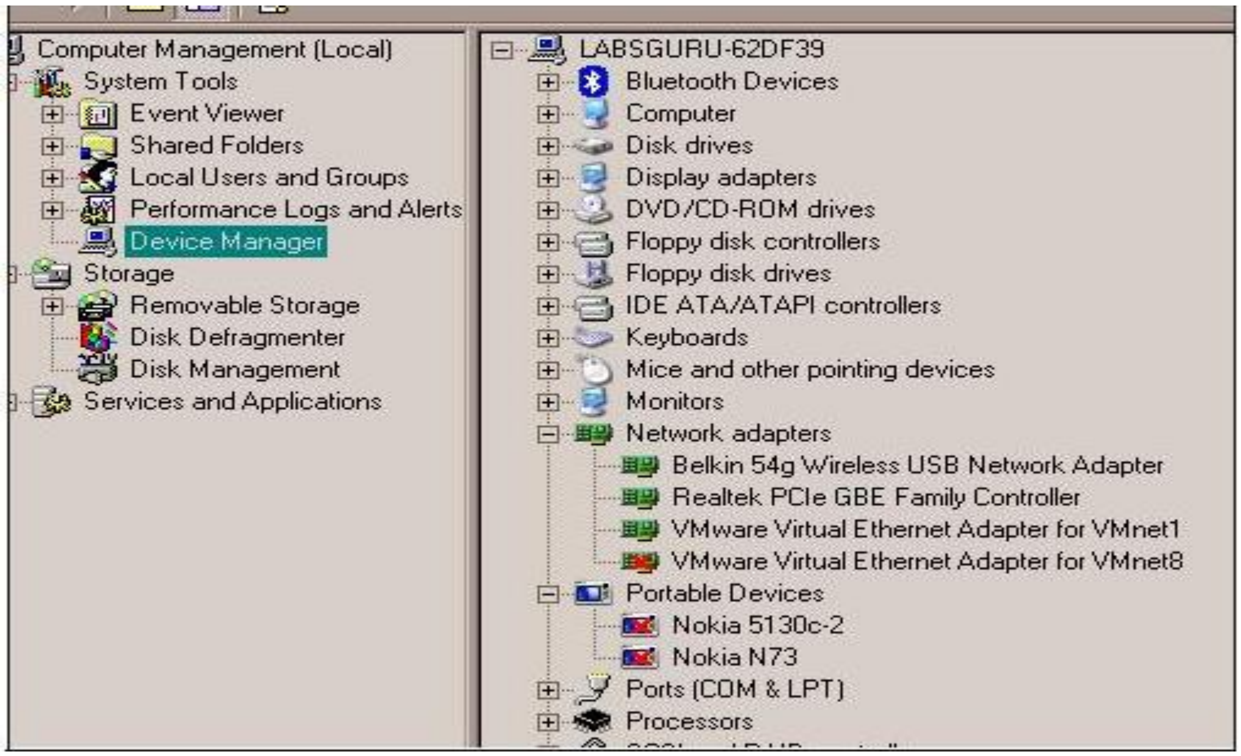


FIG. 3.6

7. As shown below if ‘**Ports (COM&LPT)’** appears then click on it, if not then device is not properly connected with the computer or the driver is not properly installed. (repeat the steps from 4 - 7)

   As soon as the **Ports (COM&LPT)’** then click on it. One or more ‘Communication Ports’ may be there, for example in above image, there are two com numbers COM1 and COM2. If there is more than one communication port, then remove and reinsert the device to the computer, the new com generated will be the right com number. Remember the com number; this is an address that will be used further in your project[9].

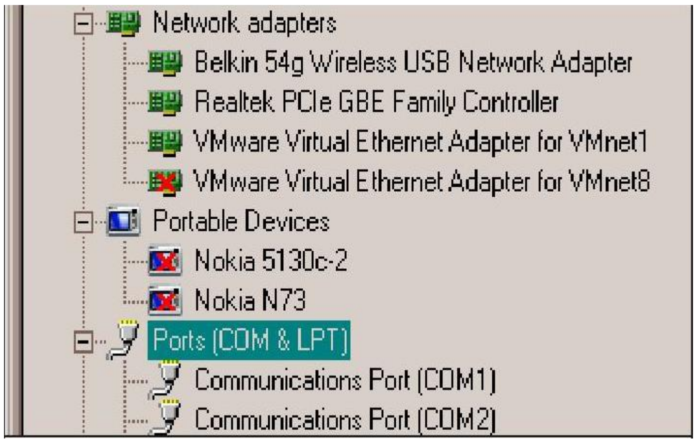
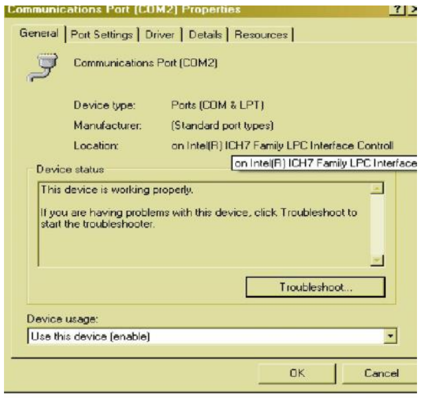

Network adapters
Belkin 54g Wireless USB Network Adapter
Realtek PCIe GBE Family Controller
VMware Virtual Ethernet Adapter for VMnet1
VMware Virtual Ethernet Adapter for VMnet8
Portable Devices
Nokia 5130c-2
Nokia N73
Ports (COM & LPT)
Communications Port (COM1)
Communications Port (COM2)

FIG. 3.8

8. If the com number is more than of single digit, say COM10, COM11 etc then change this number as follows.
    a. Double click on the ‘**Communication Port (COM N)**’ the following window will appear.
    b. Click on port settings, the following window will appear.

    Click on ‘Advanced’ tab and click on Drop down list to select any COM number (between 1 to 9).


Communications Port (COM2) Properties

General | Port Settings | Driver | Details | Resources

Communications Port (COM2)

Device type: Ports (COM & LPT)

Manufacturer: (Standard port types)

Location: on Intel(R) ICH7 Family LPC Interface Controll

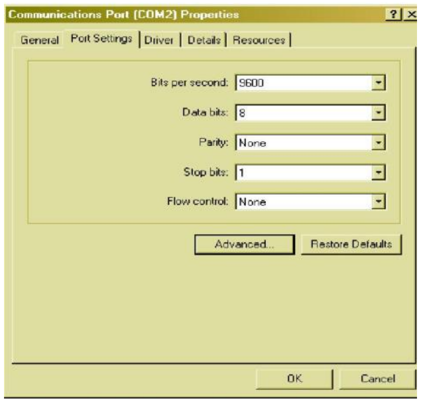
Device status

This device is working properly.

If you are having problems with this device, click Troubleshoot to start the troubleshooter.

Device usage:

Use this device (enable)


Communications Port (COM2) Properties

General | Port Settings | Driver | Details | Resources

Bits per second: 9600

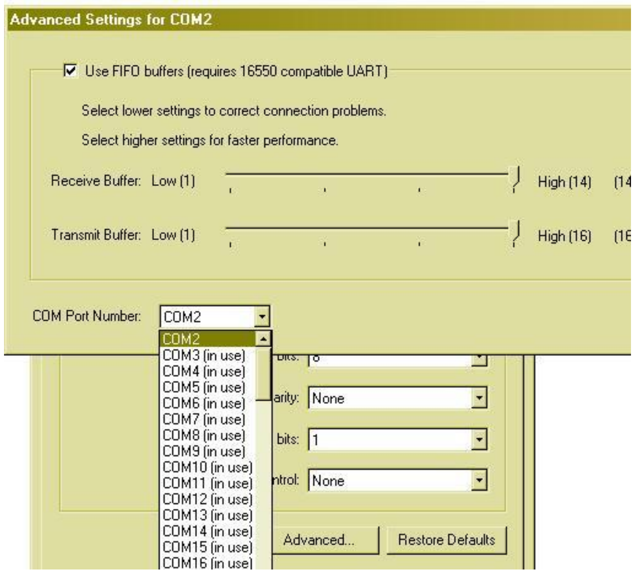
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

FIG. 3.9


Advanced Settings for COM2

[✓] Use FIFO buffers (requires 16550 compatible UART)

Select lower settings to correct connection problems.

Select higher settings for faster performance.

Receive Buffer: Low (1) High (14) (14

Transmit Buffer: Low (1) High (16) (16

COM Port Number: COM2

COM2
COM3 (in use)
COM4 (in use)
COM5 (in use)
COM6 (in use)
COM7 (in use)
COM8 (in use)
COM9 (in use)
COM10 (in use)
COM11 (in use)
COM12 (in use)
COM13 (in use)
COM14 (in use)
COM15 (in use)
COM16 (in use)

arity: None

bits: 1

ntrol: None

FIG. 3.10

d. Go on clicking OK and OK and finally the number of the COM will be changed. Remember this com number to run the project.

Step 2:

1. On the USB to UART converter see the pin outs, there should be the following minimum pin outs **RX, TX, VCC (+5V),** and **GND**.

2. On the PCB board there are 6 pins as shown below. We have to use only 4 pins out of it as
   Shown below. Connect should be as follows:
   VCC of PCB with VCC of converter module
   GND of PCB with GND of converter module
   TX of PCB with RX of converter module
   RX of PCB with TX of converter module
   Open the project and run it by pressing ‘F5’. And select the COM number.

Write the COM number in the above text box which we found in previous steps and click on SET button. If some error occurs then try it again. Now you can run your project.
If all the above parameters are met, then the testing part is complete and we can run our Project.

Caution: Check the orientation of LCD, and all IC.

After doing the coding on microcontroller the project initiate the working of sensing the temperature and now after done with coding we have to work on the power supply part and after connecting its adapter in external socket. Now we have to start downloading the software named ‘Docklight’ it monitors/displays the temperature while sensing with the device named DS18B20 (A Temperature Sensor) and the data received by the x-bee receiver.

After done the receiving section the docklight software installed in electronic device such as p.c., laptop, desktop, etc. the docklight software start monitoring the temperature analysed by the temperature sensor[3].

Result

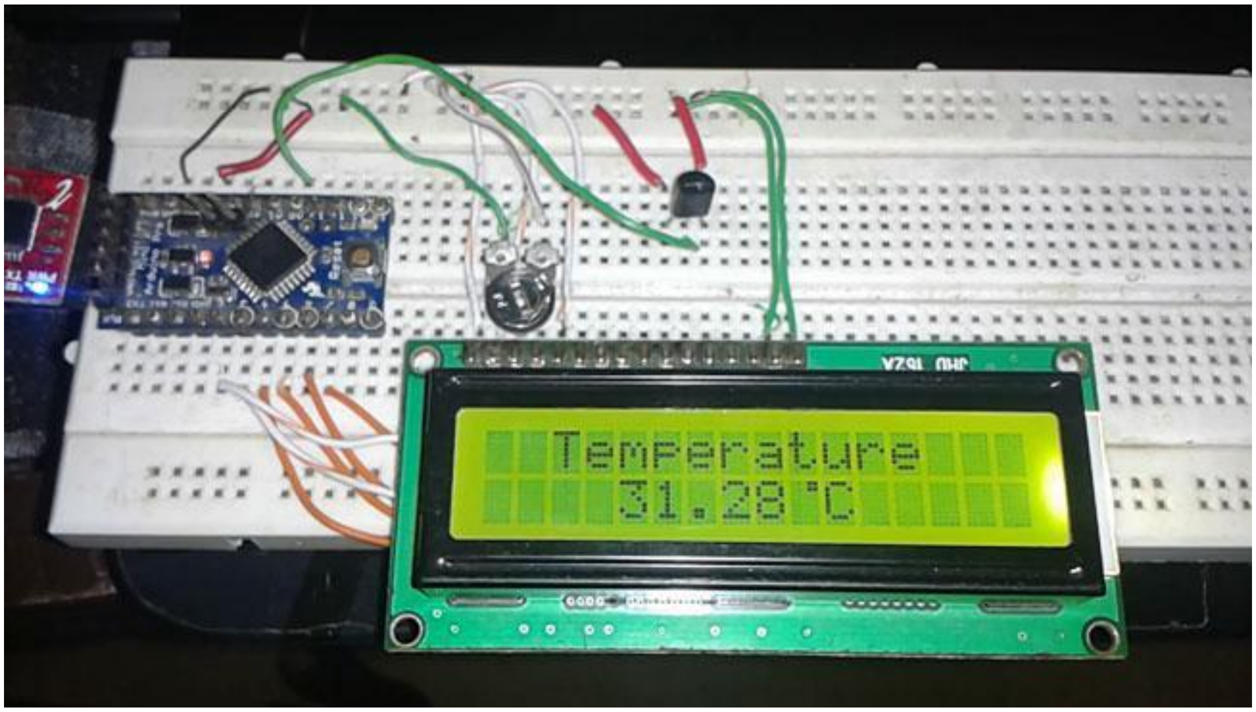


FIG. 3.4

The x-bee transmitter transmits the information of heat which is sensed by the DS18B20 Temperature sensor and also receives by its end x-bee receiver which is connected to the monitor by USB UART.

A. Field-Test Temperature Measurement we performed a field test for temperature monitoring between 8am to 4pm. The system collected the temperature readings every 1 hour and got the result.

B. Polling time The time required for each polling to a node can be measured by using the function millis (); to realize the distance between Coordinator and End Device. This testing counts the 2-way transmission time.

C. The result is being shown in figure is that when the temperature rises or lows it indicates on the lcd digitally and we easily read out by on monitor.

Chapter-4

APPLICATION

Home Automation-The Zig-Bee Home Automation profile is likely to be the first Zig-Bee application profile to hit the market place in volume and also holds promise to be the first application space where multiple products from multiple vendors are truly inter-operable allowing users to mix and match products to enhance their digital lifestyle. Lighting control, thermostats, occupancy and motion sensors, security systems, door and window sensors, as well as fixed and mobile keypads all occupy the X-Bee home automation space and can be bound together to make sophisticated home automation behaviors.

Building automation-Wireless sensing and control mesh networks can make building automation easier and more efficient by combining lighting, HVAC, security, safety systems, and other monitoring networks into a single platform.

Industrial plant monitoring-Wireless sensing and control mesh networks provide accurate and efficient IPM, and are also ideal to deploy in hazardous environments in which you want to minimize human exposure.

Conclusion

Component Characteristics	Specifications
Microcontroller	Atmega128L
Performance	< 16 MIPS throughput
In-system programmable Flash Memory	128 KB
RAM	16 KB
Configuration EEPROM	4 KB
Operating Voltage	2.7V to 5.5V
Current consumed	2.5 mA 5.5 mA

This project helps us to determine the temperature of various places wirelessly and at the eye futuristic scope is very useful.

It contains less power consumption, measure approximately 85 degree centigrade temperature and show it on monitor/display screen. It’s also very cost effective and convenient device for measure temperature wirelessly.

Future Prospects

In this paper, we built a prototype of an embedded wireless sensor network based on easy-to-use Arduino microcontroller board and Xbee module. We consider a temperature monitoring application to demonstrate the proof-of-concept of our system. The collected temperature data can be stored into the MySQL Database and retrieved later for analysis. Extensions of our current work include an extension from a star network to a mesh network which will be useful for deploying sensor networks in large areas like in buildings with multiple rooms and multiple floors.

A cost reduction for each node can be achieved by removing the Arduino board. This takes advantage of the fact that each wireless sensor node can be equipped with an XBee module alone without a microcontroller. This can be done because the XBee module can automatically sample the sensor inputs and report back to the coordinator. It is called Zigbee I/O Sampling that is Xbee module can read sampling values on its pin by itself. So, if we can use only Xbee module, price per each node will decrease.

References

[1] Stefan Poslad, Ubiquitous Computing: Smart Devices, Environments and Interactions, Wiley, 2009

[2] IEEE 802.15 WPAN Task Group 4 (TG4), http://www.ieee802.org

[3] Arduino, http://arduino.cc/.

[4] SparkFun Electronics, Temperature Sensor - LM335A, http://www.sparkfun.com

[5] Wireless Sensor Network Research Group, http://www.sensornetworks.org

[6] Won-Suk Jang, William M. Healy, Miroslaw J. Skibniewski, Wireless sensor networks as a part of a web-based building environmental monitoring system, Automation in Construction Volume 17, Issue 6, August 2008, Pages 729-736

[7] Digi International Inc,XBee ZNet2.5/XBee-PRO ZNet2.5 OEM RF Modules, Product Manual v1.x.4x - ZigBee Protocol For OEM RF Module Part Numbers: XB24-BxIT-00x,Digi International Inc.11001 Bren Road East Minnetonka, MN 55343877 912-3444 or 952 912-3444 http://www.digi.com

[8] Sabri.C, Mechatronic, 1st-Edition, John Wiley&sons, Inc., 2007, ISBN 978-0-471-47987-1

[9] James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach: International Version, 5/E,Pearson Higher Education.

[10] Timothy Boronczyk, Martin E. Psinas, PHP and MySQL: Create Modify Reuse, Wrox (May 5, 2008)

APPENDIX

Component Characteristics	Specifications
Microcontroller	**Atmega128L**
Performance	**< 16 MIPS throughput**
In-system programmable Flash Memory	**128 KB**
RAM	**16 KB**
Configuration EEPROM	**4 KB**
Operating Voltage	**2.7V to 5.5V**
Current consumed	**2.5 mA** **5.5 mA**

PCB DESIGN AND MODULE STRUCTURE:

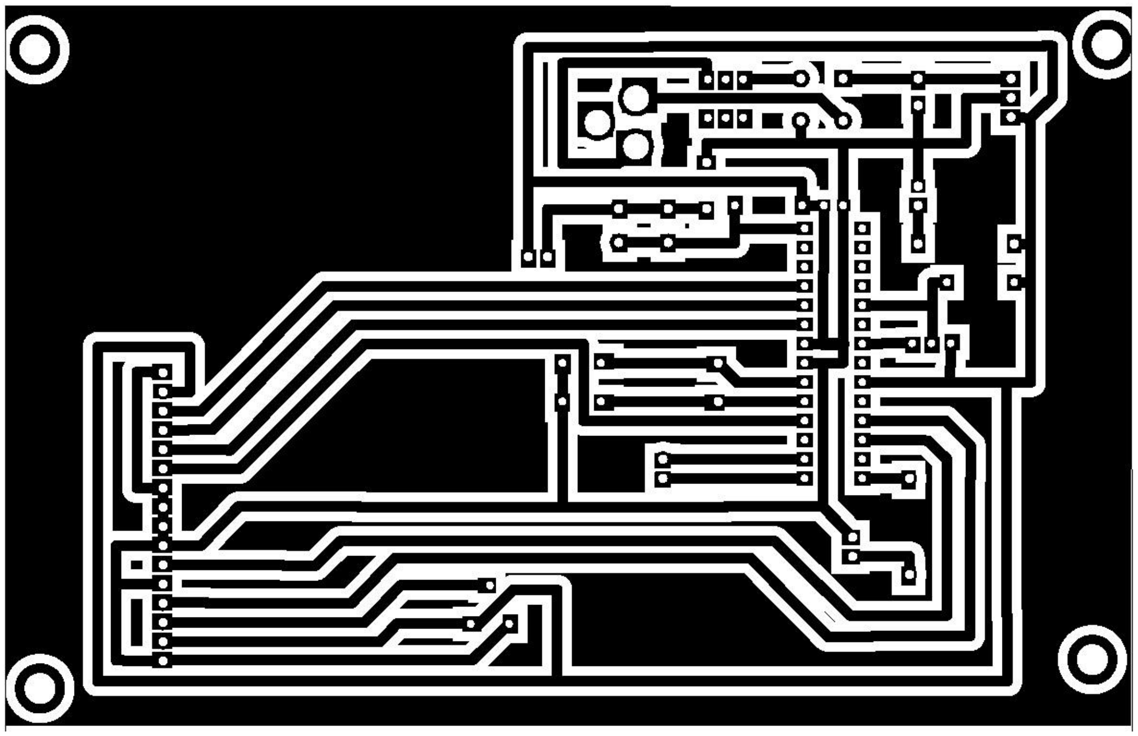


FIG. 4.1

Coding In Microcontroller:

```
NewSoftSerial xbee(7,8);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float temp=0;
void setup(void)
{
lcd.begin(16, 2);
Serial.begin(9600);
xbee.begin(9600);
Serial.println("X BEE BASED TEMP MONITORING SYSTEM");
xbee.println("X BEE BASED TEMP MONITORING SYSTEM");
tone(9,3000,100);// indicator
sensors.begin();
}
void loop(void)
{
sensors.requestTemperatures();
Serial.println("DONE");
delay(1000);
Serial.print("Temperature for the device 1 (index 0) is: ");
temp=sensors.getTempCByIndex(0);
Serial.println(temp,DEC);
xbee.println(temp,DEC);
tone(9,3000,100);// indicator
lcd.clear();
lcd.print("TEMP IN DEGREE C");
lcd.setCursor(0, 1);
lcd.print(temp,DEC);
}
Dim TEMP As String
Private Sub Command1_Click()
MSComm1.CommPort = Text1.Text
MSComm1.PortOpen = True
End Sub
Private Sub MSComm1_OnComm()
TEMP = MSComm1.Input
Label1.Caption = TEMP
Text2.Text = Text2.Text & Now & " " & TEMP
End Sub
Private Sub Label2_Click()
End Sub
```