1.INTRODUCTION

Industrial safety is one of the main aspects of industry specially mining industry. In the mining industry safety is a very vital factor. To avoid any types of unwanted phenomena all mining industry follows some basic precaution and phenomena. Communication is the main key factor for any industry today to monitor different parameters and take necessary actions accordingly to avoid any types of hazards. To avoid loss of material and damaging of human health, protection system as well as faithful communication system is necessary inside the underground mines.

To increase both safety and productivity in mines, a reliable communication must be established between workers, moving in the mine, and a fixed base station. Inside mines, the wired communication system is not so effective. The reliability and long life of conventional communications systems in harsh mining environments has always been a problem. Inside mines due to uncomfortable situation the installation cost as well as maintenance cost is high for wired communication networks. It is very difficult to reinstall the wired communication system inside mines after a landslide or damage due to any reason. Due to roof fall, if by any means some workers trapped inside mines, to maintain the continuity of the communication system is very much important to know the actual position and condition of the trapped workers. To monitor other parameters during this condition it is very much necessary to maintain the communication system as usual.

Accordingly, development of mine monitoring system to accurately detect temperature, pressure, flammable and poisonous gas and to track underground miners and vehicles on real-time has significant meaning to safety production and rescue of coal mine disaster. Coal mine safety monitoring system based on wireless sensor network can timely and accurately reflect dynamic situation of staff in the underground regions to ground computer system. A hybrid tunnel radio propagation model consisting of the free space propagation and the modified waveguide propagation can replace the wired communication system. But, using this popular radio communication inside mines has some disadvantages. When radio signals are transmitted, diffraction, attenuation, multi-path and scattering are often very serious. So, wireless communication is the burning need today for the fast, accurate, flexible safety and production process in underground mines. So we have decided to select Zigbee network for this purpose which has several advantages.

**1.1Aim:**

* To develop a social relevant project
* To learn ZIGBEE technology
* To develop a embedded system to provide safety to the mine Workers
* To learn embedded system design

**1.2 Requirements:**

**1.2.1 Hardware Used :**

* Sensor Circuit
* Comparator Circuit
* ZIGBEE Module
* LCD Module
* Power supply module

**1.2.2 Software Used :**

* Keil Microvision4
* Philips Flash Magic
* Embedded C for coding

**1.3 Overview Of the project:**

The developed system can be divided into two sections. First is a hardware circuit that will be attached with the body of the mine workers. This may be preferably fitted with the safety helmet of the workers also which should be mandatory in the premises of any underground mines. An additional stand by system can be fitted with the wrist of the underground mine workers if required. The circuit has a sensor module consisting of some MEMS based sensors that measures real-time underground parameters like temperature, humidity concentration of different gases inside mines etc. Gas concentration is meant for the harmful gases like methane and carbon-monoxide etc. Some of the gases are toxic and some are inflammable. A microcontroller is used with the sensors to receive the sensor outputs and to take the necessary decision. The microcontroller can store data as required by the user for maintaining of records.

Once temperature is more than the safety level preprogrammed at microcontroller, microcontroller decodes beep alarms through the buzzer as shown in Fig. 1 and the alert message is displayed in LCD screen. Again, once the measured humidity value is more than the safety level preprogrammed at microcontroller, it beeps the buzzer and display message in the LCD. Similarly when gas concentration crosses the safety level, microcontroller initiates the buzzer. In all such cases, this will send an alarm through an urgent message to the ground control terminal through zigbee. The microcontroller data is transmitted through two separate boards i.e. ZigBee transmission module to the data collector or receiver module. At the other end of the zigbee module receives the the data at the ground section and displays the alarm message in the monitor to take immediate action to rescue the mine workers.

**1.4 Block Diagram:**

PC

UART to USB CONVERTER

ZIGBEE RECIEVER

THERMISTOR

POWER MODULE

ZIGBEE TRANSMITTER

GAS SENSOR

HUMIDITY SENSOR

LCD

BUZZER

COMPARATOR

COMPARATOR

COMPARATOR

LPC

2148

**2.Thermistor**

A **thermistor** is a type of resistor whose resistance varies significantly with temperature, more so than in standard resistors. The word is a portmanteau of thermal and resistor. Thermistors are widely used as inrush current limiters, temperature sensors, self-resetting over current protectors and self-regulating heating elements.

Thermisters differ from resistance detectors (RTD) in that the material used in a thermistor is generally a ceramic or polymer, while RTDs use pure metals. The temperature response is also different; RTDs are useful over larger temperature ranges, while thermistors typically achieve a higher precision within a limited temperature range, typically −90 °C to 130 °C.

**Sensitivity:**

Sensitivity is a function that describes how much resistance changes with a change in temperature. According to Johnson, resistance changes of 10 percent per degree Celsius are not uncommon. Therefore, if a thermistor has a nominal value of 10,000 ohms at laboratory temperature (20 degrees Celsius), a 1 degree Celsius change in temperature could cause a 1,000 ohm change in resistance.

**Construction:**

Thermistors are most commonly formed into the shape of beads, disks, and rods. Their size can vary from as small as 1 mm and as large as several cm in diameter. A smaller thermistor will react faster to temperature changes. Different materials, such as epoxy, are used to surround and protect the thermistor from damage.

**Response time:**

Response time is how quickly a thermistor responds to a change in temperature. A 1-mm thermistor bead in an oil bath can react as quickly as half a second. The same bead in open air may have a response time of nearly 10 seconds. As the thermistor's size increases, so does its response time. If a thermistor is being used in a temperature control system, a faster response time results in better temperature control.

**Signal conditioning:**

Due to a thermistor's non-linearity, it is difficult to use alone as a temperature measurement device. The thermistor is usually connected to a divider circuit with a fixed resistor of the same nominal value. The measurement instrument then senses the change in resistance of the thermistor and converts it to a temperature reading.

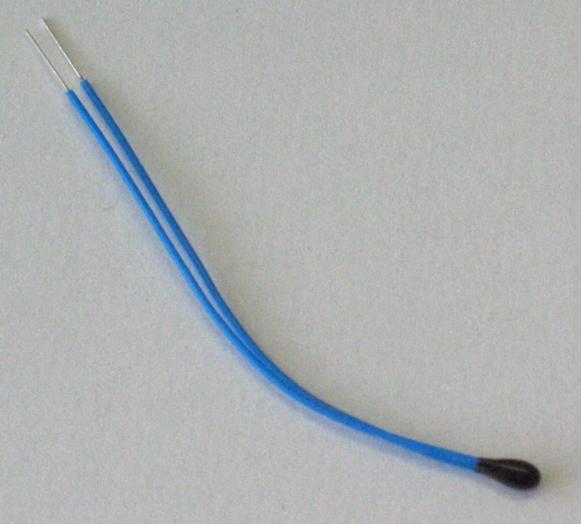
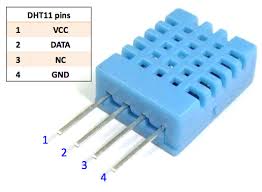


Fig: Thermistor

**3. Moisture Sensor**

Moisture/Humidity is the presence of water in air. The amount of water vapor in air can affect human comfort as well as many manufacturing processes in industries. The presence of water vapor also influences various physical, chemical, and biological processes. Humidity measurement in industries is critical because it may affect the business cost of the product and the health and safety of the personnel. Hence, **humidity sensing** is very important, especially in the control systems for industrial processes and human comfort.

Controlling or monitoring humidity is of paramount importance in many industrial & domestic applications In all such applications and many others, **humidity sensors** are employed to provide an indication of the moisture levels in the environment.

   
 Fig: Humidity Sensor

Resistive type humidity sensors pick up changes in the resistance value of the sensor element in response to the change in the humidity. Thick film conductor of precious metals like gold, ruthenium oxide is printed and calcinated in the shape of the comb to form an electrode. Then a polymeric film is applied on the electrode; the film acts as a humidity sensing film due to the existence of movable ions. Change in impedance occurs due to the change in the number of movable ions.

**4. Gas sensor**

A gas detector is a device which detects the presence of various gases within an area, usually as part of a safety system. This type of equipment is used to detect a gas leak and interface with a control system so a process can be automatically shut down. A gas detector can also sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave the area. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals.

Gas detectors can be used to detect combustible, flammable and toxic gases and oxygen depletion. This type of device is used widely in industry and can be found in a variety of locations such as on oil rigs, to monitor manufacture processes and emerging technologies such as photovoltaic. They may also be used in firefighting.

**MQ2:**

Sensitive material of MQ-2 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, the sensor’s conductivity is more higher along with the gas concentration rising. Please use simple electro-circuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensitity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

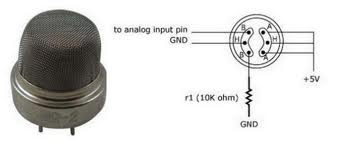


Fig: MQ2 Gas Sensor.

5. **LCD**

5.1 LCD Introduction:

A liquid crystal display (LCD) is a low cost, low power device capable of displaying text and images.LCDs are extremely common in embedded system, since such system often do not have video monitors like those that come standard with standard desktop systems. LCDs can be found in numerous common devices like watches, fax and copy machines and calculators.

The LCD controller provides a relatively simple interface between a processor and an LCD. LCDs can be added quite easily to an application and use as few as three digital output pins for control.

There are different types of LCDs such as reflexive LCD,absorption LCD, dot matrix LCD. Each type of LCD is able to display multiple characters. In addition, each character may be displayed in normal or inverted fashion. The LCD may permit a character to be blinking or may permit display of a cursor indicating the current character. Such functionality would be difficult to be implemented using software.

Thus, an LCD controller is used to provide a simple interface to an LCD, perhaps eight data inputs and one enable input. This byte may be a control word, which can be an instruction or data word.

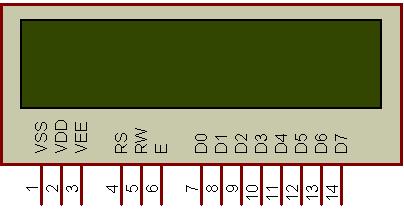
The most common connector used for the Hitachi’s H44780 based

LCDs in 14 pins in row, with pin centers 0.100” apart.

In recent years the LCD is finding widespread use replacing LED’s. this s due to the following reasons;

* The declining prices of LCDs
* The ability to display no’s, characters and graphics. This is contrast to led’s , which are limited to numbers and few characters.
* Incorporation of a refreshing controller into the LCD, there by relieving the CPU of the task of refreshing the LCD. in contrast, the led must be refreshed by the CPU to keep displaying the data.
* Ease of programming for characters and graphics.

**5.2 LCD pin descriptions:**



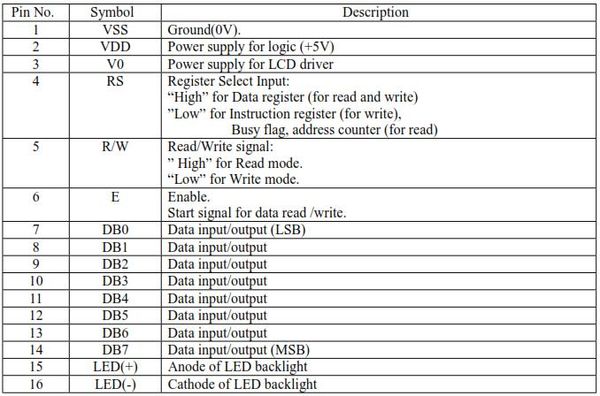


Table: LCD Pin Description.

5.3 LCD Command Codes:

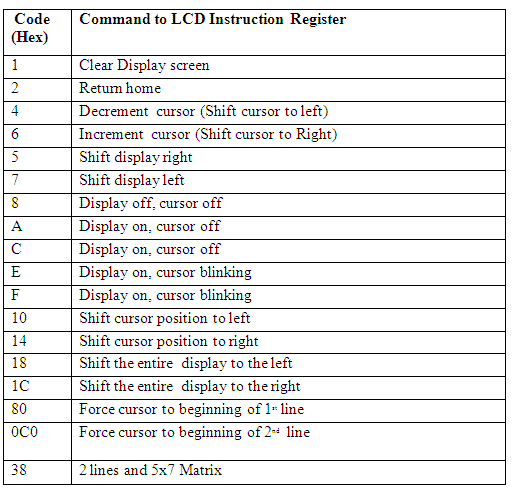


Table: LCD Command Codes

5.4 Connection of LCD with Microcontroller:

Microcontroller

LCD CONTROLLER

E

R/W

R/S

DB7-DB0

Communication Bus



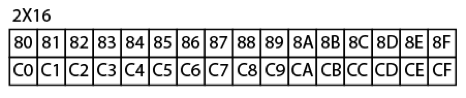


Fig: LCD Addressing(2 Rows and 16 Columns)

**6. BUZZER**

The buzzer subsystem produces a 2 KHz audible tone when powered. The buzzer will sound when the signal coming into the driver is high. It must be connected to a transistor, Darlington or transducer driver subsystem.

**Understanding the buzzer circuit**

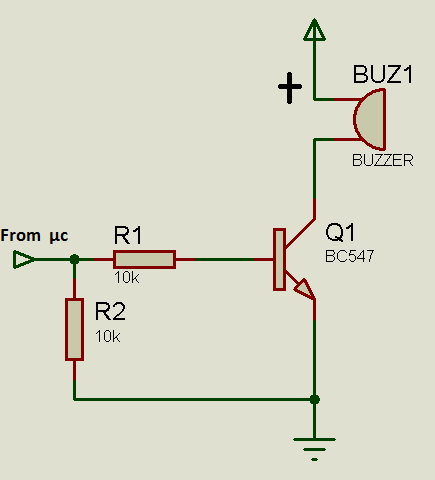


Fig: Showing Buzzer Circuit Connection

The buzzer is connected between the supply rail(+V) and the input signal. This acts as a load on the driver. When the input signal coming into the buzzer subsystem is low, a potential difference across the buzzer causes current to flow. It is this flow of current that causes the buzzer to sound.



Fig: Buzzer

**7. MICROCONTROLLER (lpc2148)**

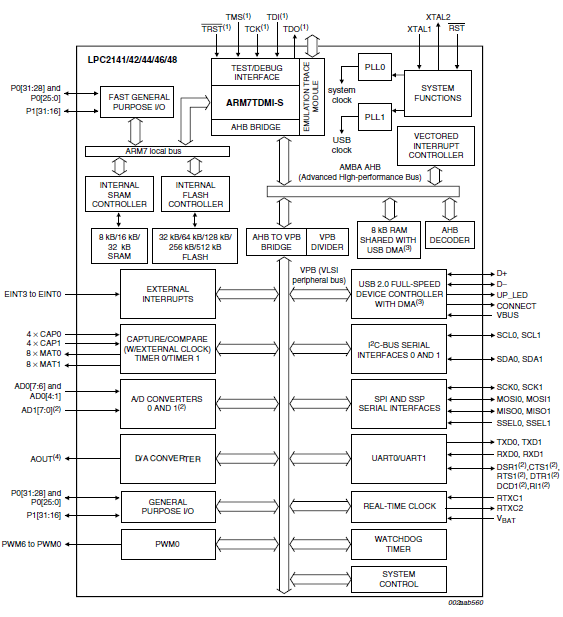
7.1 Features:

* 16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
* 8 kB to 40 kB of on-chip static RAM and 32 kB to 512 kB of on-chip flash memory. 128-bit wide interface/accelerator enables high-speed 60 MHz operation.
* In-System Programming/In-Application Programming (ISP/IAP) via on-chip boot loader software. Single flash sector or full chip erase in 400 ms and programming of 256 bytes in 1 ms.
* Embedded ICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip Real Monitor software and high-speed tracing of instruction execution.
* USB 2.0 Full-speed compliant device controller with 2 kB of endpoint RAM. In addition, the LPC2146/48 provides 8 kB of on-chip RAM accessible to USB by DMA.
* One or two (LPC2141/42 vs. LPC2144/46/48) 10-bit ADCs provide a total of 6/14 analog inputs, with conversion times as low as 2.44 μs per channel.
* Single 10-bit DAC provides variable analog output (LPC2148 only).
* Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog.
* Low power Real-Time Clock (RTC) with independent power and 32 kHz clock input.
* Multiple serial interfaces including two UARTs (16C550), two Fast I2C-bus (400 kbit/s), SPI and SSP with buffering and variable data length capabilities.
* Vectored Interrupt Controller (VIC) with configurable priorities and vector addresses.
* Up to 45 of 5 V tolerant fast general purpose I/O pins in a tiny LQFP64 package.
* Up to 21 external interrupt pins available.
* 60 MHz maximum CPU clock available from programmable on-chip PLL with settling time of 100 μs.
* On-chip integrated oscillator operates with an external crystal from 1 MHz to 25 MHz.
* Power saving modes include ideal and Power-down.
* Individual enable/disable of peripheral functions as well as peripheral clock scaling for additional power optimization.
* Processor wake-up from Power-down mode via external interrupt or BOD.
* Single power supply chip with POR and BOD circuits: CPU operating voltage range of 3.0 V to 3.6 V (3.3 V ± 10 %) with 5 V tolerant I/O pads.

7.2 Description:

The LPC2148 microcontrollers are based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine microcontroller with embedded high speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and a unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30 % with minimal performance penalty. Due to their tiny size and low power consumption, LPC2148 are ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale. Serial communications interfaces ranging from a USB 2.0 Full-speed device, multiple UARTs, SPI, SSP to I2C-bus and on-chip SRAM of 8 kB up to 40 kB, make these devices very well suited for communication gateways and protocol converters, soft modems, voice recognition and low end imaging, providing both large buffer size and high processing power. Various 32-bit timers, single or dual 10-bit ADC(s), 10-bit DAC, PWM channels and 45 fast GPIO lines with up to nine edge or level Sensitive external interrupt pins make these microcontrollers suitable for industrial control and medical systems.

7.3 Block Diagram:



(1) Pins shared with GPIO.

(2) LPC2144/46/48 only.

(3) USB DMA controller with 8 kB of RAM accessible as general purpose RAM and/or DMA is available in LPC2146/48 only.

(4) LPC2142/44/46/48 only

Fig: Block Diagramof LPC2148

7.4 PIN Diagram:

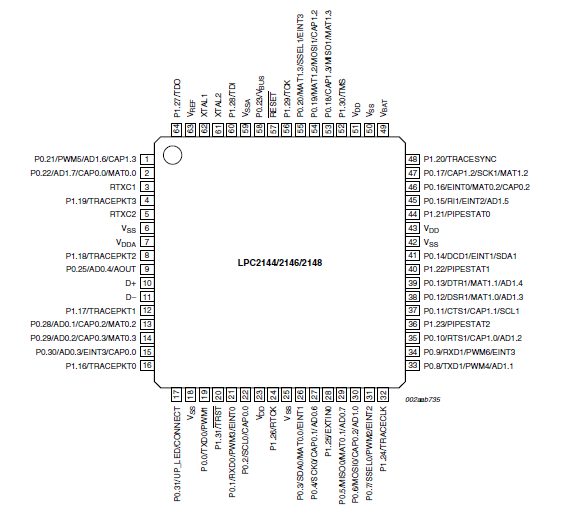


Fig: PIN Diagram of LPC2148

7.5 Pin Description:

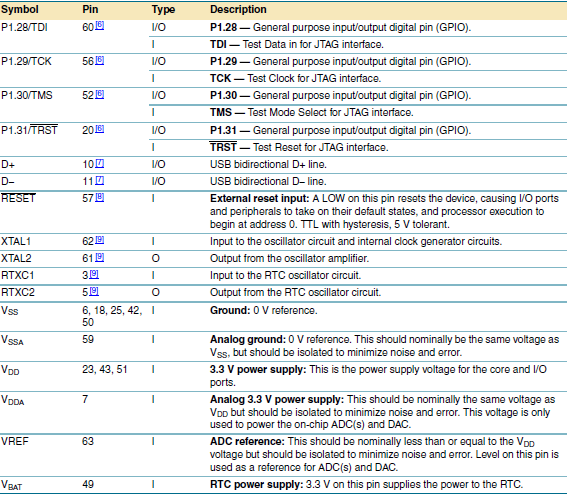


Fig: PIN Description

[1] 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control.

[2] 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control. If configured for an input

function, this pad utilizes built-in glitch filter that blocks pulses shorter than 3 ns.

[3] Open-drain 5 V tolerant digital I/O I2C-bus 400 kHz specification compatible pad. It requires external pull-up to provide an output

functionality.

[4] 5 V tolerant pad providing digital I/O (with TTL levels and hysteresis and 10 ns slew rate control) and analog input function. If configured

for an input function, this pad utilizes built-in glitch filter that blocks pulses shorter than 3 ns. When configured as an ADC input, digital

section of the pad is disabled.

[5] 5 V tolerant pad providing digital I/O (with TTL levels and hysteresis and 10 ns slew rate control) and analog output function. When

configured as the DAC output, digital section of the pad is disabled.

[6] 5 V tolerant pad with built-in pull-up resistor providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control.

The pull-up resistor’s value typically ranges from 60 kΩ to 300 kΩ.

[7] Pad is designed in accordance with the Universal Serial Bus (USB) specification, revision 2.0 (Full-speed and Low-speed mode only).

[8] 5 V tolerant pad providing digital input (with TTL levels and hysteresis) function only.

[9] Pad provides special analog functionality.

**7.6 Board Schematics:**

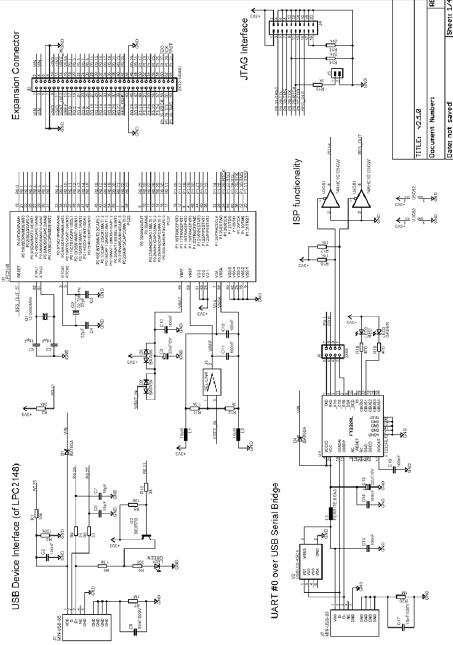
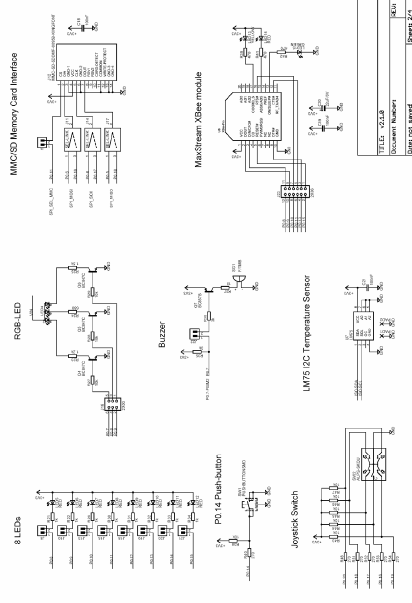


Fig: Board Schematics

Board Schematics Contd…..



7.7 Architecture Overview:

The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed Complex Instruction Set Computers

(CISC).This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective Processor core. Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory. The ARM7TDMI-S processor also employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue. The key idea behind Thumb is that of a super-reduced instruction set. Essentially, the ARM7TDMI-S processor has two instruction sets:

**•** The standard 32-bit ARM set.

**•** A 16-bit Thumb set.

The Thumb set’s 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM’s performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code. Thumb code is able to provide up to 65 % of the code size of ARM, and 160 % of the performance of an equivalent ARM processor connected to a 16-bit memory system. The particular flash implementation in the LPC2141/42/44/46/48 allows for full speed execution also in ARM mode. It is recommended to program performance critical and short code sections (such as interrupt service routines and DSP algorithms) in ARM mode. The impact on the overall code size will be minimal but the speed can be increased by 30% over Thumb mode.

7.7.1 On-chip Flash Program Memory:

The LPC2141/42/44/46/48 incorporate a 32 kB, 64 kB, 128 kB, 256 kB and 512 kB flash memory system respectively. This memory may be used for both code and data storage. Programming of the flash memory may be accomplished in several ways. It may be programmed In System via the serial port. The application program may also erase and/or program the flash while the application is running, allowing a great degree of flexibility for data storage field firmware upgrades, etc. Due to the architectural solution chosen for an on-chip boot loader, flash memory available for user’s code on LPC2148 is 500 kB. The LPC2148 flash memory provides a minimum of 100,000 erase/write cycles and 20 years of data-retention.

7.7.2 On-chip Static Ram:

On-chip static RAM may be used for code and/or data storage. The SRAM may be accessed as 8-bit, 16-bit, and 32-bit. LPC2148 provides 32 kB of static RAM respectively. In case of LPC2146/48 only, an 8 kB SRAM block intended to be utilized mainly by the USB can also be used as a general purpose RAM for data storage and code storage and execution.

7.7.3 Fast general purpose parallel I/O (GPIO):

Device pins that are not connected to a specific peripheral function are controlled by the GPIO registers. Pins may be dynamically configured as inputs or outputs. Separate registers allow setting or clearing any number of outputs simultaneously. The value of the output register may be read back, as well as the current state of the port pins. LPC2148 introduce accelerated GPIO functions over prior LPC2000 devices:

**•** GPIO registers are relocated to the ARM local bus for the fastest possible I/O timing.

**•** Mask registers allow treating sets of port bits as a group, leaving other bits

unchanged

**•** All GPIO registers are byte addressable.

**•** Entire port value can be written in one instruction.

Features:

**•** Bit-level set and clear registers allow a single instruction set or clear of any number of bits in one port.

**•** Direction control of individual bits.

**•** Separate control of output set and clear.

**•** All I/O default to inputs after reset.

7.7.4 UARTs:

The LPC2148 each contains two UARTs. In addition to standard transmit and receive data lines, the LPC2148 UART1 also provide a full modem control handshake interface. Compared to previous LPC2000 microcontrollers, UARTs in LPC2148 introduce a fractional baud rate generator for both UARTs, enabling these microcontrollers to achieve standard baudrates such as 115200 with any crystal frequency above 2 MHz. In addition, auto-CTS/RTS flow-control functions are fully implemented in hardware (UART1 in LPC2144/46/48 only).

Features:

* + 16 byte Receive and Transmit FIFOs.
  + Register locations conform to ‘550 industry standard.
  + Receiver FIFO trigger points at 1, 4, 8, and 14 bytes
  + Built-in fractional baud rate generator covering wide range of baudrates without a need for external crystals of particular values.
  + Transmission FIFO control enables implementation of software (XON/XOFF) flow control on both UARTs.
  + LPC2144/46/48 UART1 equipped with standard modem interface signals. This module also provides full support for hardware flow control (auto-CTS/RTS).

7.7.5 SPI serial I/O controller:

The LPC2148 each contains one SPI controller. The SPI is a full duplex serial interface, designed to handle multiple masters and slaves connected to a given bus. Only a single master and a single slave can communicate on the interface during a given data transfer. During a data transfer the master always sends a byte of data to the slave, and the slave always sends a byte of data to the master.

Features:

**•** Compliant with Serial Peripheral Interface (SPI) specification.

**•** Synchronous, Serial, Full Duplex, Communication.

**•** Combined SPI master and slave.

**•** Maximum data bit rate of one eighth of the input clock rate.

7.7.6 General purpose timers/external event counters:

The Timer/Counter is designed to count cycles of the peripheral clock (PCLK) or an externally supplied clock and optionally generate interrupts or perform other actions at specified timer values, based on four match registers. It also includes four capture inputs to trap the timer value when an input signal transitions, optionally generating an interrupt. Multiple pins can be selected to perform a single capture or match function, providing an application with ‘or’ and ‘and’, as well as ‘broadcast’ functions among them. The LPC2148 can count external events on one of the capture inputs if the minimum external pulse is equal or longer than a period of the PCLK. In this configuration, unused capture lines can be selected as regular timer capture inputs, or used as external interrupts.

Features:

**•** A 32-bit timer/counter with a programmable 32-bit prescaler.

**•** External event counter or timer operation.

**•** Four 32-bit capture channels per timer/counter that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.

**•** Four 32-bit match registers that allow:

**–** Continuous operation with optional interrupt generation on match.

**–** Stop timer on match with optional interrupt generation.

**–** Reset timer on match with optional interrupt generation.

**•** Four external outputs per timer/counter corresponding to match registers, with the following capabilities:

**–** Set LOW on match.

**–** Set HIGH on match.

**–** Toggle on match.

**–** Do nothing on match.

7.7.7 Watchdog timer:

The purpose of the watchdog is to reset the microcontroller within a reasonable amount of time if it enters an erroneous state. When enabled, the watchdog will generate a system reset if the user program fails to ‘feed’ (or reload) the watchdog within a predetermined amount of time.

Features:

**•** Internally resets chip if not periodically reloaded.

**•** Debug mode.

**•** Enabled by software but requires a hardware reset or a watchdog reset/interrupt to be disabled.

**•** Incorrect/Incomplete feed sequence causes reset/interrupt if enabled.

**•** Flag to indicate watchdog reset.

**•** Programmable 32-bit timer with internal pre-scaler.

**•** Selectable time period from (TPCLK × 256 × 4) to (TPCLK × 232 × 4) in multiples of TPCLK × 4.

7.7.8 Crystal oscillator:

On-chip integrated oscillator operates with external crystal in range of 1 MHz to 25 MHz. The oscillator output frequency is called fosc and the ARM processor clock frequency is referred to as CCLK for purposes of rate equations, etc. fosc and CCLK are the same value unless the PLL is running and connected.

7.7.9 PLL:

The PLL accepts an input clock frequency in the range of 10 MHz to 25 MHz. The input frequency is multiplied up into the range of 10 MHz to 60 MHz with a Current Controlled Oscillator (CCO). The multiplier can be an integer value from 1 to 32 (in practice, the multiplier value cannot be higher than 6 on this family of microcontrollers due to the upper frequency limit of the CPU). The CCO operates in the range of 156 MHz to 320 MHz, so there is an additional divider in the loop to keep the CCO within its frequency range while the PLL is providing the desired output frequency. The output divider may be set to divide by 2, 4, 8, or 16 to produce the output clock. Since the minimum output divider value is 2, it is insured that the PLL output has a 50 % duty cycle. The PLL is turned off and bypassed following a chip reset and may be enabled by software. The program must configure and activate the PLL, wait for the PLL to Lock, then connect to the PLL as a clock source. The PLL settling time is 100 μs.

7.7.10 Memory mapping control:

The Memory Mapping Control alters the mapping of the interrupt vectors that appear beginning at address 0x0000 0000. Vectors may be mapped to the bottom of the on-chip flash memory, or to the on-chip static RAM. This allows code running in different memory spaces to have control of the interrupts.

7.7.11 Power Control:

The LPC2141/42/44/46/48 supports two reduced power modes: Idle mode and Power-down mode. In Idle mode, execution of instructions is suspended until either a reset or interrupt occurs. Peripheral functions continue operation during Ideal mode and may generate interrupts to cause the processor to resume execution. Idle mode eliminates power used by the processor itself, memory systems and related controllers, and internal buses. In Power-down mode, the oscillator is shut down and the chip receives no internal clocks. The processor state and registers, peripheral registers, and internal SRAM values are preserved throughout Power-down mode and the logic levels of chip output pins remain static. The Power-down mode can be terminated and normal operation resumed by either a reset or certain specific interrupts that are able to function without clocks. Since all dynamic operation of the chip is suspended, Power-down mode reduces chip power consumption to nearly zero. Selecting an external 32 kHz clock instead of the PCLK as a clock-source for the on-chip RTC will enable the microcontroller to have the RTC active during Power-down mode. Power down current is increased with RTC active. However, it is significantly lower than in Ideal mode. A Power Control for Peripherals feature allows individual peripherals to be turned off if they are not needed in the application, resulting in additional power savings during active and idle mode.

**8. ZIGBEE**

8.1 Introduction:

Zigbee wireless network technology is a new standard launched and made by Zigbee alliance. The alliance , founded in august 2001, is a fast growing and non-profit organization , and it aims to provide consumers with more flexible and easier electronic products. The second half of 2002, four large corporations including the british company Invensys, Mitsubishi electric corporation , Motorola and the dutch gaint Philips semiconductor corporation jointed together to announce that they would join the ‘Zigbee Alliance’ to invent the next generation wireless communication standard named “Zigbee” , which became a significant milestone in the development process. In October 2004, the Zigbee Alliance announced a version 1.0 of zigbee protocol, and in December 2005 Version 1.1. This protocol is developed based on IEEE 802.15.4.

Zigbee uses free frequency bands of 2.4 GHz and 900 MHz, and its transmission rate is 20 Kbps to 250 Kbps. In this project we are using standard Zigbee wireless network modules. The Zigbee Module and protocol have been successfully applied to power systems, medical and some other fields.

8.2 Why Zigbee:

Zigbee is a worldwide open standard for wireless radio networks in the monitoring and control fields. The standard was developed by the Zigbee Alliance (an association of international Companies) to meet following principal needs:

* Low cost
* Ultra-low power consumption
* Use of unlicensed radio bands
* Cheap and easy installation
* Flexible and extendable networks
* Integrated intelligence for network set-up and message routing

Some of the above requirements are related – for example, the need for extremely low power consumption is motivated by the use of battery powered nodes which can be installed cheaply and easily, without any power cabling, in difficult locations.

For instance, if you try to talk to someone in the room next to this room via mobile phone, RF signals are first sent to the ground station antenna of the mobile phone company, after which they pass through the mobile phone company’s server and then once again return to the ground station antenna before finally reaching the mobile phone in the next room. Makes for quite the detour, a technology with RF signals that need not pass through the mobile phone company’s ground station antennas, but rather can directly connect to the mobile phone in the next room via wireless communication. With the Zigbee protocol, P2P communications have expanded even further, and it has now become possible to build mesh networks, which allows communication with multiple terminals simultaneously. Compared to Bluetooth technology, the Zigbee protocol allows one to one 65,000 nodes of communication,while the Bluetooth one allows one to only seven nodes, though the data rate in zigbee based products is slower. Additionally, the Zigbee standard requires much less power to the extent that Zigbee and devices can operate for a hundred days, or even several years with a battery. Moreover, Zigbee technology is easy to apply, compact in size, and cheap enough to become more versatile in various applications. It is not easy to think of the possible benefits of P2P mesh networking in a practical application. One real application seriously being investigated is automated meter reading, AMR, of utility meters for electricity, water, gas and heat. AMR using a P2P mesh network is expected to contribute to energy conservation by real time monitoring of consumption. Another application could be a disaster-monitoring system used during a major earthquake or other natural disaster. The self-recovery function of mesh-networking will enable the monitors to better grasp the extent of damage via sensor networks. Environment monitoring, via the placement of sensors throughout forests to gather environmental parameters for use in nature conservation efforts, may be another possibility. However, these application areas are just the beginning. If the development of other users for mesh networks advances, there would be good potential for us to see this technology used in a broad range of fields. So Zigbee is the suitable technology for this kind of application.

If the GSM Modem is used in our project instead of Zigbee, it needs external service provider(Airtel, Vodafone), but whereas when Zigbee modem in our project itself acts as a Transmitter and Receiver avoiding connecting to service provider and also avoiding call congestion problem, etc. Another reason that Zigbee is used in our project is to avoid EMI interference in Air. So Zigbee is the suitable technology for this kind of application.

8.3 ZIGBEE Module Description:

**ZigBee** is the set of specs built around the [IEEE](http://www.wisegeek.com/what-is-the-ieee.htm) 802.15.4 wireless protocol. The IEEE is the Institute of Electrical and Electronics Engineers, a non-profit organization dedicated to furthering technology involving electronics and electronic devices. The **802 group** is the section of the IEEE involved in network operations and technologies, including mid-sized networks and local networks. **Group15** deals specifically with [wireless networking](http://www.wisegeek.org/what-is-wireless-networking.htm) technologies. The standard itself is regulated by a group known as the **ZigBee Alliance**, with over 150 members worldwide.

ZigBee technology builds on IEEE standard 802.15.4 which defines the physical and MAC layers. Above this, ZigBee defines the application and security layer specifications enabling interoperability between products from different manufacturers. In this way ZigBee is a superset of the 802.15.4 specification

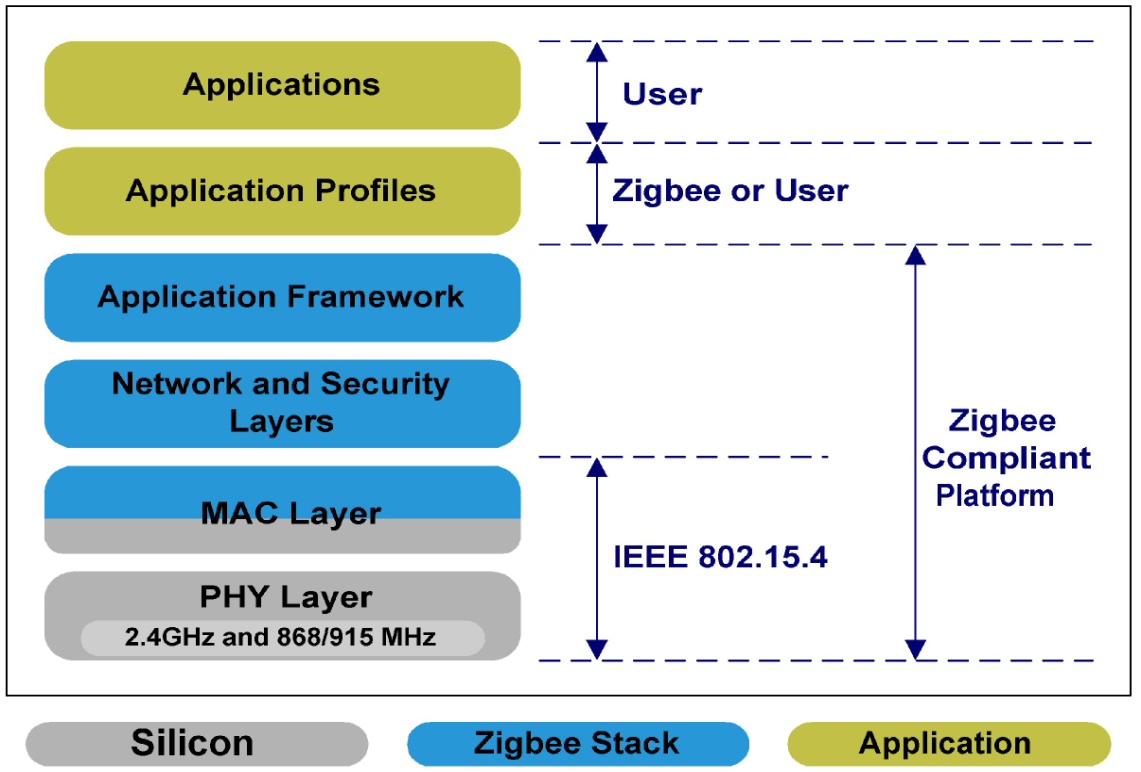


Fig: Zigbee Layers

Zigbee layers

Physical Layer:

* The IEEE 802.15.4 standard provides 3 freqency bands  to Zigbee:
* 868.3 MHz for Europe, with one channel at 868.3 MHz and raw data rate of 20 kb/s   (figure 3)
* 915 MHz for North American, with 10 channels between 902.0 and 928.0 MHz and a raw data of 20 kb/s (figure 2)
* 2.400 GHz - for world wide application.Here we have 16 channels between 2.4 and 2.4835 GHz and a data rate of 250 kb/s

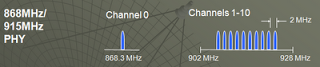


Fig: 868/915 MHz

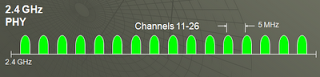


Fig: 2.4 Ghz

The modulation techniques also vary according to the band in use. Direct sequence spread spectrum (DSSS) is used in all cases. However for the 868 and 915 MHz bands the actual form of modulation is binary phase shift keying. For the 2.4 GHz band, offset quadrature phase shift keying (O-QPSK) is employed.

MAC Layer

The IEEE 802.15.4 MAC sub-layer controls access to the radio channel using a CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) mechanism. In view of the fact that systems may operate in heavily congested environments, and in areas where levels of extraneous interference is high, the 802.15.4 specification has incorporated a variety of features to ensure exceedingly reliable operation . These include a quality assessment, receiver energy detection and clear channel assessment. CSMA (Carrier Sense Multiple Access) techniques are used to determine when to transmit, and in this way unnecessary clashes are avoided.

## Upper layers (ZigBee)

Above the physical and MAC layers defined by 802.15.4, the ZigBee standard itself defines the upper layers of the system. This includes many aspects including the messaging, the configurations that can be used, along with security aspects and the application profile layers.

There are three different network topologies that are supported by ZigBee, namely the star, mesh and cluster tree or hybrid networks. Each has its own advantages and can be used to advantage in different situations.

The star network is commonly used, having the advantage of simplicity. As the name suggests it is formed in a star configuration with outlying nodes communicating with a central node.

Mesh or peer to peer networks enable high degrees of reliability to be obtained. They consist of a variety of nodes placed as needed, and nodes within range being able to communicate with each other to form a mesh. Messages may be routed across the network using the different stations as relays. There is usually a choice of routes that can be used and this makes the network very robust. If interference is present on one section of a network, then another can be used instead.

Finally there is what is known as a cluster tree network. This is essentially a combination of star and mesh topologies.

Both 802.15.4 and ZigBee have been optimised to ensure that low power consumption is a key feature. Although nodes with sensors of control mechanisms towards the centre of a network are more likely to have mains power, many towards the extreme may not. The low power design has enabled battery life to be typically measured in years, enabling the network not to require constant maintenance.

## Data transfer

The data is transferred in packets. These have a maximum size of 128 bytes, allowing for a maximum payload of 104 bytes. Although this may appear low when compared to other systems, the applications in which 802.15.4 and ZigBee are likely to be used should not require very high data rates.

The standard supports 64 bit IEEE addresses as well as 16 bit short addresses. The 64 bit addresses uniquely identify every device in the same way that devices have a unique IP address. Once a network is set up, the short addresses can be used and this enables over 65000 nodes to be supported.

It also has an optional superframe structure with a method for time synchronisation. In addition to this it is recognised that some messages need to be given a high priority. To achieve this, a guaranteed time slot mechanism has been incorporated into the specification. This enables these high priority messages to be sent across the network as swiftly as possible.

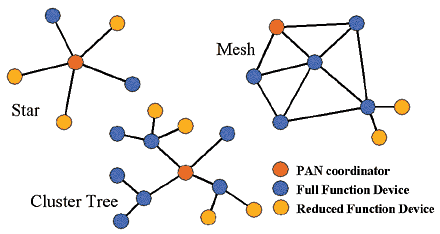
After receiving a data packet, the receiver performs a 16-bit cyclic redundancy check (CRC) to verify that the packet is not corrupted. If the result is positive, an acknowledgement packet is sent back. If the CRC result is negative (the packet was corrupt), the packet is dropped and no acknowledgement is sent.



Fig : ZIGBEE Module

**8.3 Network Topology:**

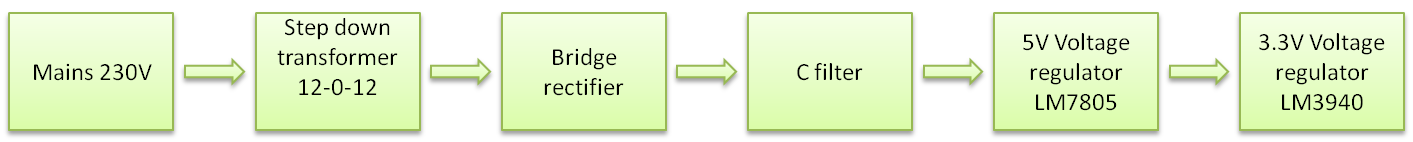
Topology Models



**9.HARDWARE DETAILS**

**9.1 Power Supply Module:**

The microcontroller and other get power from power supply unit. Here AC voltage is converted to non regulated DC using rectifier, ripples are removed using filter and the non regulated DC is fed to the voltage regulator LM7805 to get the required 5V DC. And to generate 3.3V LM3940 Voltage regulator is used.



9.1 Components of Power supply Module:

Transformer:

The main source of power supply is a transformer. The maximum output power of power supply is dependent on maximum output power of transformer .We determine power from its current and voltage rating. e.g. If there is a transformer of 12V, 500mA then maximum power delivered by transformer is 6Watt.It means we can drive a load from this transformer up to 6w. In our project our maximum power requirement is 1watt.So to provide this power we use 12V/250mA transformer. The maximum output power of this transformer is 4watt.It means it can easily drive load up to 4 watt.

Bridge Rectifier:

Bridge Rectifier is a circuit which is used to convert ac to dc. Every electronic circuit requires a dc power supply for rectification. Bridge rectifier internally has four diodes to convert AC to DC.

C filter:

After rectification we obtain dc supply from ac but it is not pure dc it may have some ac ripples .To reduce these ripples we use filters. It comprises of two filters –low frequency ripple filter and high frequency ripple filter. To reduce low frequency ripples we use electrolytic capacitor. The voltage rating of capacitor must be double from incoming dc supply. It blocks DC and passes ripples to ground.

Voltage Regulator:

Regulator is a device which provides constant output voltage with varying input voltage. There are two types of regulators -

* Fixed voltage regulator
* Adjustable regulator

We have used fixed voltage regulator LM78XX last two digits signify output voltage. The voltage for our system is 5V that is why we have used 7805 regulator which provides 5V

from 12V dc.

Output filter:

It is used to filter out output ripple if any.

9.2 Interfacing Thermistor to Microcontroller:

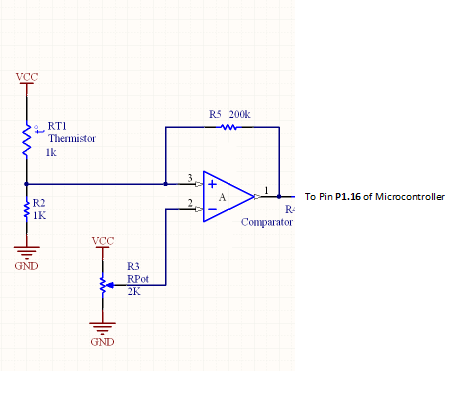


Fig: Interfacing Thermistor to microcontroller

as we can see from the circuit that the reference voltage has been fixed at the inverting terminal of the comparator circuit i.e at the 2nd pin of the op-amp used. After repeated experimental analysis it has been found that the corresponding voltage level reflected by the thermistor and the resistance combination at the 3rd pin of the op-amp for a temperature of 40 degree centigrade is approximately 3.93V. If the temperature goes increases beyond 40 degrees, the voltage level will go below 3.93V ar the 3rd pin.

So we have given a fixed reference voltage of 3.93V at the inverting terminal of the comparator circuit. Initially as the temperature is below 40 degrees, the input to the 3rd pin of comparator circuit is always greater than or equal to 3.93V. So the 3rd pin has a higher input voltage level than the 2nd pin which drives the output of the comparator always HIGH. As the temperature increases beyond 40 degrees, the voltage level at 3rd pin decreases below 3.93V and hence the input voltage level of 2nd pin becomes higher than that of the 3rd pin which drives the output of the comparator LOW.

So this LOW signal is read by the Microcontroller and it alerts us by alarming the buzzer, displaying messages through LCD and by sending alert messages through Zigbee communication Protocol.

9.3 Interfacing Moisture Sensor to Microcontroller:

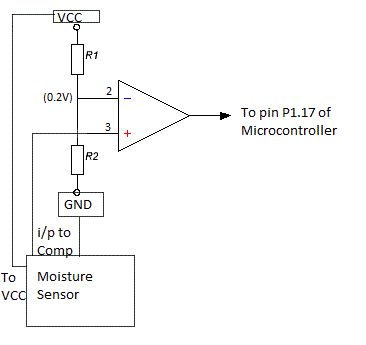


Fig: Interfacing Moisture Sensor to microcontroller

as we can see from the circuit that the reference voltage has been fixed at the inverting terminal of the Comparator circuit i.e at the 2nd pin of the op-amp used. After repeated experimental analysis it has been found that the corresponding voltage level reflected by the Moisture Sensor at the 3rd pin of the op-amp for a humidity level below 80% is approximately 0.2V. If the humidity level increases beyond 80%, the voltage level will go above 0.2V the 3rd pin.

So we have given a fixed reference voltage of 0.2V at the inverting terminal of the comparator circuit. Initially as the humidity level is below 80%, the input to the 3rd pin of comparator circuit is always less than or equal to 0.2V. So the 3rd pin has a lower input voltage level than the 2nd pin which drives the output of the comparator always LOW. As the humidity level increases beyond 80%, the voltage level at 3rd pin increases above 0.2V and hence the input voltage level of 3nd pin becomes higher than that of the 2nd pin which drives the output of the comparator HIGH.

So this HIGH signal is read by the Microcontroller and it alerts us by alarming the buzzer, displaying messages through LCD and by sending alert messages through Zigbee communication Protocol.

9.4 Interfacing Gas Sensor with Microcontroller:

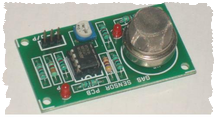


Fig: Gas Sensor with In-built Comparator circuit

**+ Vcc**

**- GND**

**O/p To pin P1.18 of Microcontroller**

Fig: Connection information of Gas Sensor.

We are using a MQ-2 gas sensor which has an in-built comparator circuit in it. The sensor has only 3 connectable pins viz. **+**, **-** and **o/p**. Here the o/p pin is the output from the comparator circuit. The ‘+’ pin of the sensor is connected to the Vcc, the – pin is connected to the GND and the o/p pin of the sensor is connected to the pin P1.18 of the microcontroller. The sensor is so designed that if any poisonous or hazardous Gas is detected, then the comparator circuit output will go HIGH and the o/p pin of the sensor circuit will go HIGH. Then this HIGH signal is read by the microcontroller pin P1.18 and the microcontroller is so designed that it alerts us by operating the alarm and by displaying alert messages through LCD. Alert messages are also sent to the base station through Zigbee communication.

9.5 LCD Connection Details:



Fig: LCD connection details

The LCD can be used is a 16 pin LCD. It can be used in either of the two modes viz. 8-bit mode or 4-bit mode. In our project we are using our LCD in 4-bit mode. As shown in the figure above, Vcc pin and the Background Led light pin is connected to the 5V pin of the microcontroller. The Background Led is used to keep the LED screen lighted. The ground pin is connected to ground. The LCD has 8 data lines (D0-D7). Out of these pins we are using only four data lines, D4-D7. So these four pins of LED are connected to pins P0.10 – P0.13 of the microcontroller. Rest of the data lines i.e D0-D3 is grounded. The control pin RS of LED is connected to pin P1.24 of the microcontroller. The EN control pin of the LED is connected to the pin P1.25 of the microcontroller. In our project we are using the LED only for writing data into it. So R/W pin is grounded to perform write operation by the microcontroller. So as soon as any alert is sensed by the microcontroller, it sends the alert message to the LED through its data lines and the alert message gets displayed into the LED screen.

**10. SOFTWARES USED**:

10.1 Embedded C:

When designing software for a smaller embedded system with the 8051, it is very commonplace to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempered to write the software in assembly language.

The trouble with projects done with assembly code is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher level language like C can directly address these issues. A program written in C is easier to read than an assembly program.

Since C program processes great structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A higher order language such as C allows a developer to write code , which resembles the a human’s thought process more closely than does the equivalent assembly code. The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable.

By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have c compilers available, which supports ANSI C.

All of this is not to say that assembly language does not have its place. In fact, many embedded system (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only one way to go. one of the great things about the C language is that it allows you to perform low level manipulations of the hardware if needed be, yet provides you with the functionality and abstraction of a higher order language.

The code is written using Keil C software. Code written in assembly language is difficult to understand so easily understandable, high level language Keil C is preferred.

Embedded development tools from Keil software support all 8051 compatible devices.

Keil offer an extensive range of evaluation boards and starter kits to quick start to your development. Keil development tools for the 8051 microcontroller architecture support every level of software developer from he professional applications engineer to the student just learning about the embedded software development.

The industry-standard Keil C compilers, macro assemblers , Debuggers , real time kernels, single board computers, and Emulators support all 8051 derivatives and help you get your projects completed on schedule. The keil 8051 development tools are designed to solve the complex problems facing embedded software developers

10.2 OVERVIEW ON KEIL MICROVISION 4

Keil development tools for the 8051 microcontroller architecture support entry level of software developer from the professional applications engineer to the student just learning about embedded software development.

The industry-standard Keil C compilers, macro assemblers, Debuggers, real time kernels, single-board computers, and emulators support all 8051 derivatives and help you get your projects completed on schedule.

The Keil 8051 development tools are designed to solve the complex problems facing embedded software developers.

* When starting a new project, simply select the microcontroller you use from the device database and the µvision IDE sets all compiler, assembler, linker, and memory options for you.
* The Keil µvision Debugger accurately simulates on-chip peripherals (I2C, CAN, UART, and SPI, Interrupts, I/O ports, A/D converter, D /A Converter and PWM modules) of your 8051 devices. Simulation helps you understand hardware configurations and avoids time wasted on setup problems. Additionally, with simulation, you can write and test applications before target hardware is available.
* When you are ready to begin testing your software application with target hardware, use the MON51, MON390, MONADI, or FlashMON51 target monitors, the ISD51 In-system debugger, or the LINK USB-JTAG Adapter to download and test program code on your target system.

The µvision IDE from Keil, combines project management, make facilities, source code editing, program debugging, and complete simulation in one powerful environment. µvision4 helps you get programs working faster than ever while providing an easy to use development platform. The editor and debugger are integrated into a single application and provide a seamless embedded project development environment.

The µvision4 IDE is windows based software development platforms that combines a robust editor, project manager, and make facility. µvision4 integrates all tools including the C compiler, macro assembler, linker/locator, and HEX file generator. µvision4 helps expedite the development process of your embedded applications by providing the following features:-

* The ‘device database’ which automatically sets the assembler, compiler, and linker options for the chip you selects. This prevents you from wasting your time configuring the tools and helps you get started writing code faster.
* A robust ‘project manager’, which lets you create several different configurations of your target from a single project file. Only the Keil µvision IDE allows you to create an output file for simulating, an output file for debugging with an emulator, and an output file for programming an EPROM---all from the same project file.
* An integrated ‘make’ facility with automatic dependency generation. You don’t have to figure out which header files and include files are used by which source files. The keil compilers and assemblers do that automatically.
* Interactive error correction. As your project compiles, errors and warnings appear in an output window. You may make corrections to the files in your project while µvision continues to compile in the background. Line numbers associated with each error or warnings are automatically resynchronized when you make changes to the source.

10.3 FLASH MAGIC

Flash magic is a software tool used to download the program from computer to the microcontroller.

**11. Working of the system:**

The various sensors i.e gas-sensor, humidity sensor and the thermistor are connected to comparator circuits. Each comparator circuit is so designed that it changes its output state if the threshold limit of the various parameters i.e presence of poisonous gas, high temperature and high humidity, etc. exceeds a threshold limit which is predefined experimentally. The output of these comparators is read by microcontroller. If the parameters readings are found to cross the desirable threshold limit, the microcontroller is programmed to alert us by operating the buzzer. An arrangement is also made to display various alert messages through an LCD.

The alert message is also transmitted through Zigbee communication. Zigbee uses UART port of the microcontroller at the transmitter side to receive the messages sent by it. At the receiver side, we have used a UART to USB converter to directly fed the messages received by the Zigbee receiver to the computer and the messages are displayed on the computer screen. So, whenever the parameters readings have crossed a desired level, the designed system alerts in following three ways:

* By sendinf alarm signal through Buzzer.
* By displaying alert messages through LCD.
* By sending alert message to the database via a Zigbee module so as to take quick action in order to handle the situation.

**12. FLOWCHART**

Get the values from comparators o/p

to the Microcontroller

Values exceeding threshold??

NO

YES

Buzzer ON

Display the alert message on

LCD Screen

Is Zigbee

Ready?

NO

YES

Calling Zigbee Function for

transmitting or Receiving

Display alert messages on COMPUTER screen.

**13. CONCLUSION:**

The mine safety system thus provides protection and safety measure which is of utmost importance for saving life during hazardous mine disasters. The design which we have done is a basic model of Mine safety system consisting of a microcontroller and Zigbee module which can be improvised and used effectively to save life and resources.

**14. BIBLIOGRAPHY**

1. E. K. Stanek, “Mine Electro-technology Research: The Past 17 Years”, *IEEE transactions on industry applications*, Vol. 24, No. 5, 1988, pp. 818-19
2. X. Ma, Y. Miao, Z. Zhao, H. Zhang, J. Zhang, “A novel approach to Coal and Gas Outburst Prediction Based on Multi-sensor Information Fusion”, *Proc. IEEE international conference on automation and* *logistics*, 2008, pp 1613-18.
3. Rong Yan. Design of Mine Safety Monitoring System Based on the Wireless Network [D]. Ji’nan: Shandong University, 2007.
4. LPC2148 datasheets and various sensors datasheets.

**13.APPENDIX**

13.1 Program Code:

LPC2148 Code:

#include <LPC21xx.H>

#include<stdio.h>

#include<string.h>

#include "gpio.h"

#include "lcd.h"

#include"gsm.h"

#include"uart.h"

void del\_2ms()

{

unsigned char i;

for(i=0;i<=250;i++);

}

void del\_10ms()

{

unsigned char i;

for(i=0;i<5;i++)

{

del\_2ms();

}

}

void del\_100ms()

{

unsigned char i;

for(i=0;i<10;i++)

{

del\_10ms();

}

}

/\*

void del\_500ms()

{

unsigned char i;

for(i=0;i<5;i++)

{

del\_100ms();

}

}

\*/

void del\_1sec()

{

int i;

for(i=0;i<400;i++)

{

del\_100ms();

}

}

void delaym()

{

int i;

for(i = 0; i < 300; i++)

;

}

void delay1()

{

int k;

for(k = 0; k < 2000; k++)

; /\* 25 \*/

}

int main()

{

int i,a=0,x=0,y=0,z=0;

Gpio\_init\_pin\_sel0(SELECT\_PORT0\_GPIO);

Gpio\_init\_pin\_sel1(SELECT\_PORT0\_GPIO);

Gpio\_init\_pin\_sel2(SELECT\_PORT1\_GPIO);

Gpio\_init\_pin\_sel0(SELECT\_UART1);

Uart\_Init(UART1,BAUD\_9600,CHAR\_LENGTH\_8,PARITY\_NO,STOP\_BIT\_1,FIFO\_TRIG8);

Gpio\_Set\_output\_port0(PORT0\_PIN10 | PORT0\_PIN11 | PORT0\_PIN12 | PORT0\_PIN13);

Gpio\_Set\_output\_port1(PORT1\_PIN24 | PORT1\_PIN25 | PORT1\_PIN19);

IOCLR1 = 0x00080000;

delaym();

lcdinit();

printLcd(" WELCOME TO ");

del\_1sec();

del\_1sec();

for(i=0;i<40;i++)

{

delaym();

}

clear();

printLcd(" EFFICIENT MINE ");

printLcd2(" SAFETY STSTEM ");

del\_1sec();

del\_1sec();

for(i=0;i<100;i++)

{

delaym();

}

clear();

printLcd(" USING ZIGBEE ");

printLcd2("BASED EMBEDDED ");

del\_1sec();

del\_1sec();

for(i=0;i<100;i++)

{

delaym();

}

clear();

printLcd(" SYSTEM ");

printLcd2(" ");

del\_1sec();

del\_1sec();

IOSET1 = 0x00080000;

for(i=0;i<100;i++)

{

delaym();

}

clear();

//////////////////////////////////////////////////////////////////////////

a=0;

printLcd(" OK ");

printLcd2(" ");

del\_1sec();

del\_1sec();

del\_1sec();

IOCLR1 = 0x00080000;

while(1)

{

if(((IOPIN1&0x00010000)==0x00000000) && x==0)

{

x=1;

IOSET1 = 0x00080000;

clear();

printLcd(" WARNING ! ");

printLcd2(" TEMP ==> >40C ");

Uart\_Puts(" !! WARNING LEVEL TEMPERATURE ==> >40C !! ");

del\_1sec();

del\_1sec();

IOCLR1 = 0x00080000;

a=0;

}

if(((IOPIN1&0x00020000)==0x00020000) && y==0)

{

y=1;

IOSET1 = 0x00080000;

clear();

printLcd(" WARNING ! ");

printLcd2("HUMIDITY : >80%");

Uart\_Puts(" !! WARNING LEVEL HUMIDITY ==> >80% !! ");

del\_1sec();

del\_1sec();

IOCLR1 = 0x00080000;

a=0;

}

if(((IOPIN1&0x00040000)==0x00040000) && z==0)

{

z=1;

IOSET1 = 0x00080000;

clear();

printLcd(" WARNING ! ");

printLcd2(" GAS DETECTED ");

Uart\_Puts(" !! WARNING LEVEL GAS DETECTED !! ");

del\_1sec();

del\_1sec();

IOCLR1 = 0x00080000;

a=0;

}

if(((IOPIN1&0x00070000)==0x00010000) && a==0)

{

x=0;

y=0;

z=0;

clear();

printLcd("SYSTEM PARAMETER");

printLcd2(" NORMAL ");

Uart\_Puts(" !! SYSTEM IS IN NORMAL CONDITION !! ");

del\_1sec();

del\_1sec();

a=1;

}

}/// while(1)

}/// main