BIOMEDICAL PATIENT MONITORING SYSTEM

Project Report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

APURV GARG-061032

KRISHNA KOUNDIYNA-061083

HARINDER BHASIN-061151

Under the Supervision of

Ms. Vanita Rana

May-2010

Jaypee University of Information Technology

Waknaghat, Solan - 173234, Himachal Pradesh

CERTIFICATE

This is to certify that project report entitled "Biomedical Patient Monitoring System", submitted by Apurv Garg, Krishna Koundinya and Harinder Bhasin in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

Date:

Supervisor's Name: Ms. Vanita Rana

Designation: Lecturer,ECE

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Signature

1. **(Apurv Garg)**
   Roll No.061032

Signature

2. **(Krishna Koundinya)**
   Roll No.061083

Signature

3. **(Harinder Bhasin)**
   Roll No.061151

ACKNOWLEDGEMENT

The zeal to accomplish the task of formulating the project report on "Biomedical Patient Monitoring System" could not have been realised without the support and cooperation of the members of the faculty of ECE Department.

We express our gratitude and sincere thanks to **Ms. Vanita Rana** (Lecturer, Electronics and Communication Department) for providing us the opportunity to undertake the project under her able guidance. She helped us develop novel solutions to every problem and helped us emerge with good engineering acumen.

DATE:

Apurv Garg	**Krishna Koundinya**	**Harinder Bhasin**
(061032)	**(061083)**	**(061151)**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

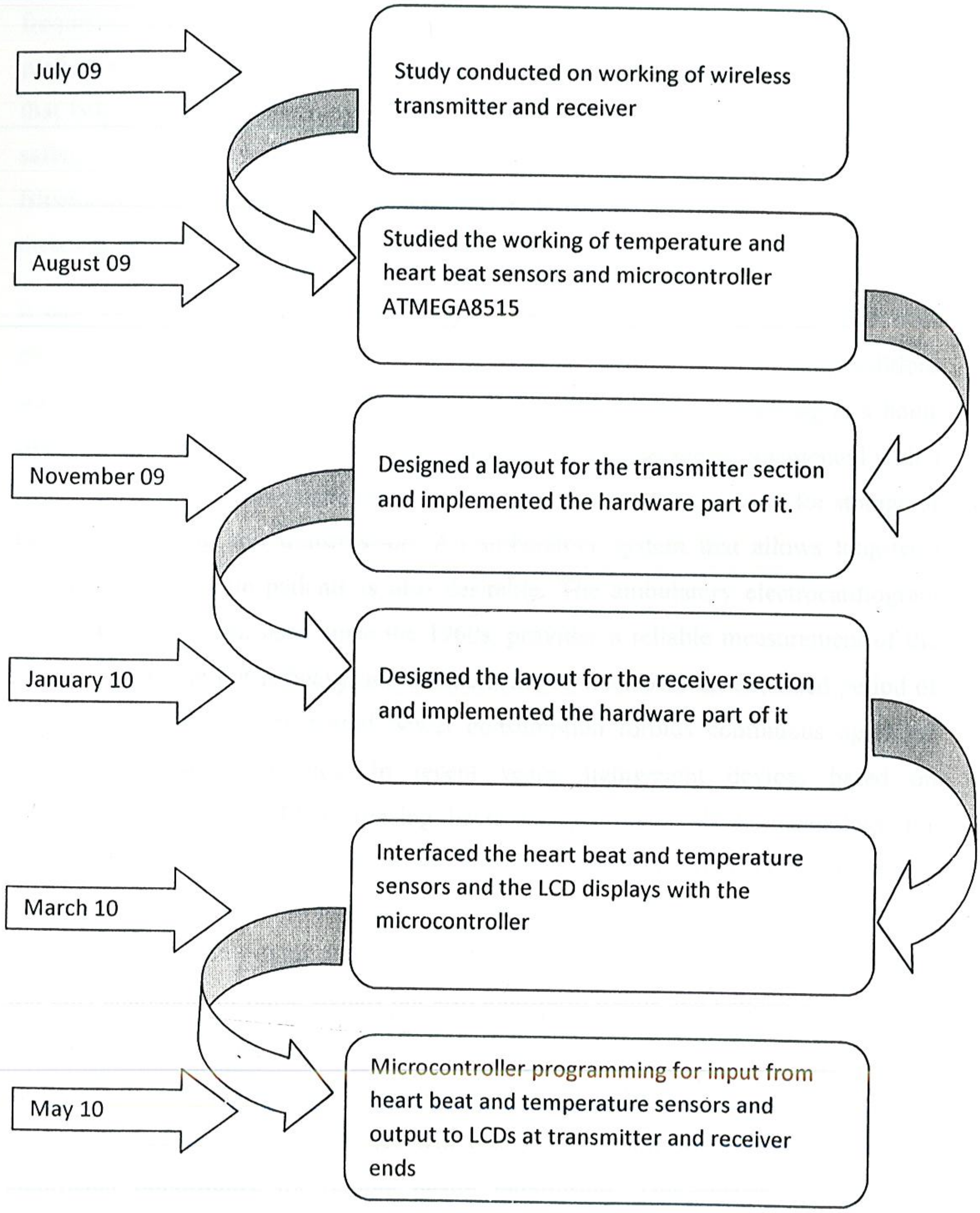
Patient monitoring systems is the term for all the various devices that are used to supervise patients. One category of such devices is devices that alerts if the patient gets into a critical state. Example of one such device is a heartbeat monitor. To a large extent computer based monitoring and intensive care unit systems have become cheap enough to be deployed on a large scale in many intensive care units around the world. The bedside has become an important point of displaying data. Bedside monitors have capabilities of intelligent monitoring, intelligent alarming, plug and play modules, TCP/IP and Ethernet networking and many other features provide easy, integrated monitoring in any facility.

Patient monitoring still need a lot of improvement there is still a need of mobility and real time monitoring. So, nowadays scope is shifting towards the wireless patient monitoring.

We thought of implementing a novel technique to whole concept of patient monitoring. We have tried to perform the wireless patient monitoring in this simply the vital information of the patient will be displayed in the doctor's cabin.

PROJECT PROCESS

The project idea has evolved over a period of one year and it is important that we discuss the efforts of the team and the process through which the project has evolved.



CHAPTER-1

INTRODUCTION

1.1. Why this project?

The popularity of wireless computing technologies has fuelled advances in radio frequency (RF) technology and continued decreases in their cost. Low cost, high-performance RF devices and systems are finding their way into various applications that help improve productivity, increase user convenience and enhance security and safety. Such prominent applications include wireless LANs, portable GPS receivers, Bluetooth-enabled laptops, printers, phones and headsets, building automation, remote metering and large-scale industrial sensor networks, among others.

It has been long recognized in the health care industry that long-term, continuous monitoring is a key element in preventive care for people with chronic conditions such as cardiovascular disease. A typical example of remote monitoring is a home care device, such as an electronic blood pressure or glucose meter, augmented with a low-power radio that transmits the measured data to a nearby computer for storage or further processing and transmission. An ambulatory system that allows long-term monitoring of mobile patients is also desirable. The ambulatory electrocardiogram (EKG) Holter device, used since the 1960s, provides a reliable measurement of the wearer's heartbeat but is heavy and cumbersome to wear over an extended period of time. In addition, its substantial power consumption forbids continuous operation using low-capacity batteries. In recent years, lightweight devices based on photoplethymographic (PPG) sensing have emerged as a viable technology for continuous measurement of vital signs. Wearable, wireless biosensors connected to self organizing, wireless sensor networks allow physicians to continuously monitor vital signs while the wearer performs normal activities, helping physicians to capture not only snapshots of pulse signals but also long-term trends and patterns that provide invaluable information about a patient's ongoing condition. The availability of advanced sensing devices and low-power RF combined with sophisticated; self-organizing networking technologies will enable new applications and represents a significant opportunity for remote health monitoring. This system will serve 3 requirements

1) The first is a small form factor so that these devices can fit inside or attach easily to a health monitoring device such as wrist or arm band, ring sensor or other wearable or implantable device.
2) The second requirement is extremely low power so that small, coin-size batteries can be used for an extended period of time. A low-power radio link is an indispensable element in a long-term monitoring system where measured data needs to be continuously transmitted from the sensing device to a data collection centre. For an RF device to run for a reasonable period of time on a coin size battery, the average power consumption rate should be on the order of one milli-ampere.
3) The third requirement is a highly sophisticated protocol for low latency, high scalability and high network responsiveness. This last requirement is particularly important for applications where patients are mobile, such as in assisted-living environments.

Continuous physiological monitoring requires small form factor devices, external battery-powered RF and a fast and efficient networking protocol. All of these elements are now in place and ready to be adapted to numerous biomedical applications. Moreover cardiac patients can be regularly monitored without the physical presence of the doctor.

Biomedical patient monitoring system is equipment used to monitor the patient remotely. Patients admitted to hospitals often find themselves with dozens of wires and cables strung from their every extremity trying to roll over at night resulting in a very large, expensive cat's cradle with the strings ending at sticky pads affixed to sensitive areas. So a solution is, the Wireless Patient Monitoring System, which would accept signals from sensors, beaming that data straight to the people who need to view it whether they be down the hall at the nurse's station or the doctor.

The need for patient monitoring is apparent in situations where the patient is:

- In unstable physiological regulatory systems, for example in the case of a drug overdose or anaesthesia.
- In a life threatening condition, for example where there are indications of a heart attack.

- In risk of developing a life threatening condition.
- In a critical physiological state.
- In spite of the improvement of communication link and despite all progress in advanced communication technologies, there are still very few functioning commercial wireless monitoring systems, which are most off-line, and there are still a number of issues to deal with.

Therefore, there is a strong need for investigating the possibility of design and implementation of an interactive real-time wireless communication system. In our project, a generic real-time wireless communication system will be designed and developed for short and long term remote patient-monitoring applying wireless protocol. The primary function of this system is to monitor the temperature and Heart Beat of the Patient and the Data collected by the sensors are sent to the Microcontroller. The Microcontroller transmits the data over the air.

1.2 Description

At the receiving end a receiver is used to receive the data and it is decoded and fed to Microcontroller, which is then displayed over the LCD display. If there is a dangerous change in patient's status an alarm is also sounded.

To enhance time-to time monitoring of patients in clinical and hospital atmosphere in order to provide the patient with accurate and relevant medication and proper care. It also helps doctors have access to the entire database of patient's present status and track their recovery even when the doctor is not able to physically monitor the patient.

Block Diagram Description

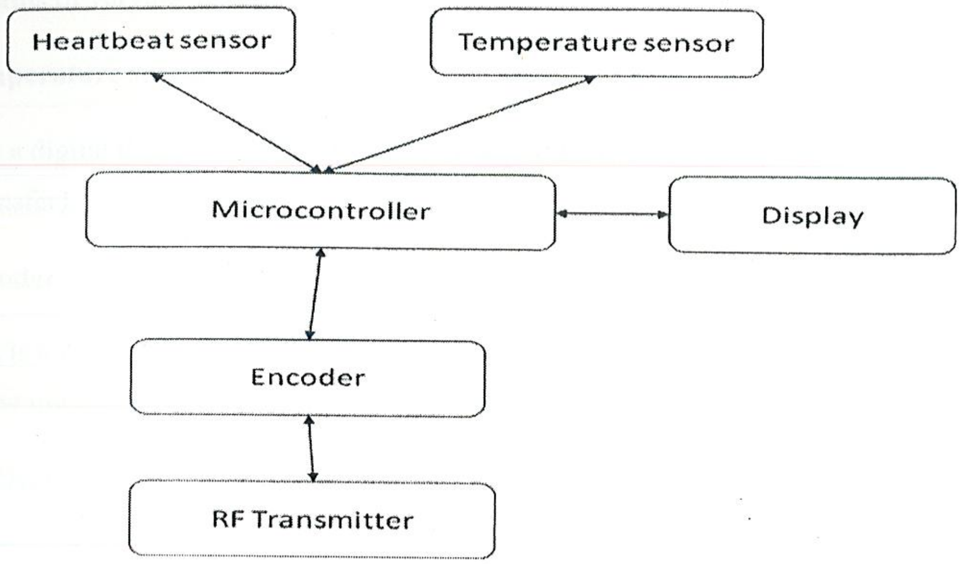


FIGURE1.1 Block Diagram of Transmitter

1.2.1 Transmitter

Microcontroller

Atmega8515 from ATMEL is being used. We use this because it reduces the amount of external hardware or internal software necessary to process the sensory data. Functions of micro controller in this are:

1. Internal counter is used to count the heartbeat.
2. Used to do display.
3. It is also used to interface the temperature sensor.

Liquid Crystal Display

Display used here is the LCD display. It is an intelligent LCD. It is a 16*2 LCD, which displays 32 characters at a time 16 will be on the 1st line and 16 will be on the 2nd line. There are two lines on the LCD and it works on extended ASCII code i.e. when ASCII code is send it display it on the screen. On the LCD total no of pins are 16 out of which 14 pins are used by the LCD and 2 are used for backlight. LCD is an edge trigger device i.e. from high to low.

Heartbeat sensor

It is a pair of IR led and phototransistor it senses the blood flow and gives the output in terms of varying voltages which is then calculated and counted.

Temperature sensor

It is a digital thermometer IC (DS1621) Temperature is read as a 9-bit value (2-byte Transfer).

Encoder

This is a 2^{12} series encoder HT12E. It is used to encode the data which is being sent by the microcontroller to be transmitted.

1.2.2Receiver

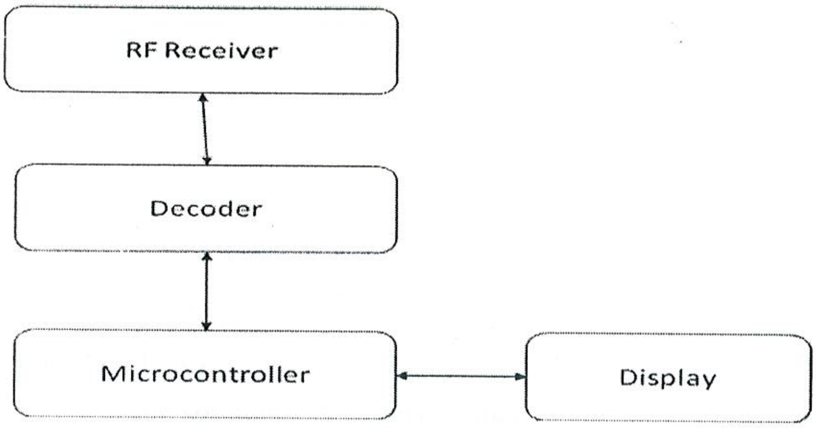


FIGURE1.2 Block Diagram of Receiver

Decoder

This is a 2^{12} series decoder HT12D. It is used to decode the data which is being received by RF receiver.

RF Receiver

It receives the encoded data sent by the RF Transmitter and gives it to decoder HT12D.

CHAPTER 2:

Theory of Sensors

In our project we will be using two sensors: Temperature Sensor and Heartbeat Sensor. Starting with temperature sensors, temperature sensors come in a wide variety and have one thing in common: they all measure temperature by sensing some change in a physical characteristic.

The seven basic types of temperature sensors to be discussed here are thermocouples, resistive temperature devices (RTDs, thermistors), infrared radiators, bimetallic devices, liquid expansion devices, molecular change-of-state and silicon diodes.

2.1 Thermocouples Sensor

Thermocouples are voltage devices that indicate temperature by measuring a change in voltage. As temperature goes up, the output voltage of the thermocouple rises - not necessarily linearly.

Often the thermocouple is located inside a metal or ceramic shield that protects it from exposure to a variety of environments. Metal-sheathed thermocouples also are available with many types of outer coatings, such as Teflon, for trouble-free use in acids and strong caustic solutions. A variety of thermocouples are available, suitable for different measuring applications (industrial, scientific, food temperature, medical research, etc.).

1) Type K (Chromel (Ni-Cr alloy) / Alumel (Ni-Al alloy)): The "general purpose" thermocouple. It is low cost and, owing to its popularity, it is available in a wide variety of probes. They are available in the −200 °C to +1200 °C range.
2) Type E (Chromel / Constantan (Cu-Ni alloy)): Type E has a high output (68 µV/°C) which makes it well suited to low temperature (cryogenic) use. Another property is that it is non-magnetic.
3) Type J (Iron / Constantan): Limited range (−40 to +750 °C) makes type J less popular than type K. The main application is with old equipment that cannot

accept modern thermocouples. J types cannot be used above 760 °C as an abrupt magnetic transformation causes permanent decalibration

4) Type N (Nicrosil (Ni-Cr-Si alloy) / Nisil (Ni-Si alloy)): High stability and resistance to high temperature oxidation makes type N suitable for high temperature measurements without the cost of platinum (B, R, S) types. They can withstand temperatures above 1200 °C.

Thermocouple types B, R, and S are all noble metal thermocouples and exhibit similar characteristics. They are the most stable of all thermocouples, but due to their low sensitivity (approximately 10 μV/°C) they are usually only used for high temperature measurement (>300 °C).

Thermocouples are most suitable for measuring over a large temperature range, up to 1800 K. They are less suitable for applications where smaller temperature differences need to be measured with high accuracy, for example the range 0--100 °C with 0.1 °C accuracy. For such applications other sensors are more suitable.

2.2 Resistive Temperature Detector Sensor (RTD)

A basic physical property of a metal is that its electrical resistivity changes with temperature. All RTD's are based on this principle. The heart of the RTD is the resistance element. Several varieties of semi-supported wire-wound fully supported bifilar wound glass, and thin film type elements are shown here.

Some metals have a very predictable change of resistance for a given change of temperature; these are the metals that are most commonly chosen for fabricating an RTD. A precision resistor is made from one of these metals to a nominal ohmic value at a specified temperature. By measuring its resistance at some unknown temperature and comparing this value to the resistor's nominal value, the change in resistance is determined. Because the temperature vs. resistance characteristics are also known, the change in temperature from the point initially specified can be calculated. We now have a practical temperature sensor, which in its bare form (the resistor) is commonly referred to as a resistance element.

Through years of experience, the characteristics of various metals and their alloys have been learned, and their temperature vs. resistance relationships is available in look-up tables. For some types of RTD's, there are also equations that give us the temperature from a given resistance. This information has made it possible for instrument manufacturers to provide standard readout and control devices that are compatible with some of the more widely accepted types of RTD's.

Platinum RTD's can measure temperatures from -200°C to 650°C. (IEC says -200°C to 850°C).

2.3 Bimetallic Devices Sensor

The Bimetallic strip is a mechanical temperature sensor element. It converts temperature to a mechanical displacement. This displacement may be coupled to a switch for simple on-off function, to a needle of an indicator, or to a position detector for electronic output. The most common application of the bimetallic strip is as a thermostat switch used for temperature and energy control.

A bimetallic strip is simply constructed from two strips of different metals bonded together. Typically a welding process is used for bonding, but rivets, bolts, adhesive and other fasteners can also be used.

The operation of the bimetallic strip relies on the different expansions rates of the two metals to temperature change (the different coefficients of thermal expansion of the metals).

As a matter of interest, the bimetallic strip can be scaled up or down. On a large scale, it can provide literally tones of force for mechanical control or other purposes. On a smaller scale, it can provide the force and movement for micro machine integrated circuits (MMIs).

Advantages of bimetallic devices are portability and independence from a power supply. However, they are not usually quite as accurate as are electrical devices, and you cannot easily record the temperature value as with electrical devices like thermocouples or RTDs; but portability is a definite advantage for the right application.

2.4 Thermometers

Thermometers are well-known liquid expansion devices. Generally speaking, they come in two main classifications: the mercury type and the organic, usually red, liquid type. The distinction between the two is notable, because mercury devices have certain limitations when it comes to how they can be safely transported or shipped.

For example, mercury is considered an environmental contaminant, so breakage can be hazardous. Be sure to check the current restrictions for air transportation of mercury products before shipping.

2.5 Change-of-state Sensors

Change-of-state temperature sensors measure a change in the state of a material brought about by a change in temperature, as in a change from ice to water and then to steam. Commercially available devices of this type are in the form of labels, pellets, crayons, or lacquers.

For example, labels may be used on steam traps. When the trap needs adjustment, it becomes hot; then, the white dot on the label will indicate the temperature rise by turning black. The dot remains black, even if the temperature returns to normal.

Change-of-state labels indicate temperature in °F and °C. With these types of devices, the white dot turns black when exceeding the temperature shown; and it is a nonreversible sensor which remains black once it changes color. Temperature labels are useful when you need confirmation that temperature did not exceed a certain level, perhaps for engineering or legal reasons during shipment. Because change-of-state devices are nonelectrical like the bimetallic strip, they have an advantage in certain applications. Some forms of this family of sensors (lacquer, crayons) do not change color; the marks made by them simply disappear. The pellet version becomes visually deformed or melts away completely.

Limitations include a relatively slow response time. Therefore, if you have a temperature spike going up and then down very quickly, there may be no visible response. Accuracy also is not as high as with most of the other devices more commonly used in industry. However, within their realm of application where you need a nonreversing indication that does not require electrical power, they are very practical.

Other labels which are reversible operate on quite a different principle using a liquid crystal display. The display changes from black color to a tint of brown or blue or green, depending on the temperature achieved. For example, a typical label is all black when below the temperatures that are sensed. As the temperature rises, a color will appear at, say, the 33°F spot - first as blue, then green, and finally brown as it passes through the designated temperature. In any particular liquid crystal device, you usually will see two color spots adjacent to each other - the blue one slightly below the temperature indicator, and the brown one slightly above. This lets you estimate the temperature as being, say, between 85° and 90°F.

Although it is not perfectly precise, it does have the advantages of being a small, rugged, nonelectrical indicator that continuously updates temperature.

2.6 Silicon band gap temperature sensor

Silicon band gap temperature sensor is a very ordinary form of temperature sensor or thermometer used in electronic equipment. The silicon bandgap temperature sensor is able to measure temperatures on computer chips with accuracy and speed. What are the advantages of using silicon bandgap temperature sensors?

One of the benefits of using the silicon bandgap temperature sensor is that the bandgap temperature sensor can be included in a silicon integrated circuit at a very low cost. Silicon temperature sensors are fast becoming important transducers in electronic systems because as the electronic systems become more complex, it is even more important to monitor critical temperatures. Silicon sensors are preferred because they are accurate, cheap, linear and can be integrated on the same IC as amplifiers and any other required processing functions. How does a silicon bandgap temperature sensor works? The principle of the bandgap temperature sensor is that the forward voltage of a silicon diode is dependant on the temperature. An electronic circuit, for example the Brokaw bandgap reference, can be used to calculate the temperature of the diode. The results that you obtained will stay valid up to about 200°C to 250°C when leakage currents become strong enough to corrupt the measurement. Once you go above 250°C in temperature, silicon carbide is used to replace silicon.

2.7 Infrared Sensors(IR)

Infrared sensors are noncontacting sensors. As an example, if you hold up a typical infrared sensor to the front of your desk without contact, the sensor will tell you the temperature of the desk by virtue of its radiation - probably 68°F at normal room temperature.

In a noncontacting measurement of ice water, it will measure slightly under 0°C because of evaporation, which slightly lowers the expected temperature reading.

Here in our project we are using DS1621 IC which is a digital thermometer. The DS1621 measures temperature using a bandgap-based temperature sensor because at low temperatures (i.e. body temperature) bandgap based temperature sensors work best.

In starting of our project we used LM35 temperature sensor IC but it has a drawback that it gives analog output so there is a need for an analog to digital converter (ADC) for converting the output to digital format before feeding it to microcontroller.

So we have come up with the idea of using DS1621 which has 9 Bit digital output. It has some more advantages over LM35 that are:

1) Temperature measurements require no external components
2) Temperature is read as a 9-bit value (2-byte transfer)
3) Converts temperature to digital word in less than 1 second
4) Thermostatic settings are user definable and non-volatile

So now we will be describing two sensors of our project in 2 parts:

Part 1: Temperature Sensor IC (DS1621)

Specifications of DS1621 are:

1) Pin Diagram

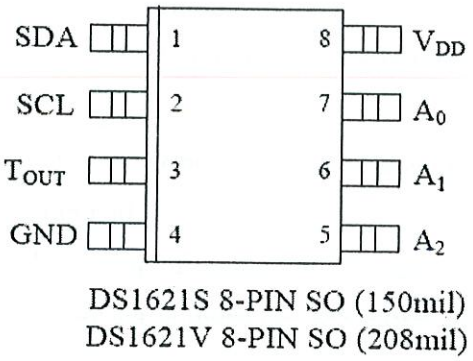


FIGURE-2.1 Pin Diagram of DS1621

2) Pin Description

SDA - 2-Wire Serial Data Input/output

SCL - 2-Wire Serial Clock

GND - Ground

TOUT - Thermostat Output Signal

A0 - Chip Address Input

A1 - Chip Address Input

A2 - Chip Address Input

VDD - Power Supply Voltage

3) Basic Description

The DS1621 Digital Thermometer and Thermostat provides 9-bit temperature readings, which indicate the temperature of the device. The thermal alarm output, TOUT, is active when the temperature of the device exceeds a user-defined temperature TH. The output remains active until the temperature drops below user defined temperature TL, allowing for any hysteresis necessary. User-defined temperature settings are stored in non-volatile memory so parts may be programmed prior to insertion in a system. Temperature settings and temperature readings are all communicated to/from the DS1621 over a simple 2-wire serial interface.

4) Operation

A block diagram of the DS1621 is shown in Figure 2.1The DS1621 measures temperature using a bandgap-based temperature sensor. A delta-sigma analog-to digital converter (ADC) converts the measured temperature to a digital value that is calibrated in °C; for °F applications, a lookup table or conversion routine must be used.

The temperature reading is provided in a 9-bit, two's complement reading by issuing the READ TEMPERATURE command.

Table 2.1 describes the exact relationship of output data to measured temperature. The data is transmitted through the 2-wire serial interface, MSB first. The DS1621 can measure temperature over the range of -55_C to +125_C in 0.5_C increments.

Functional Block

Diagram

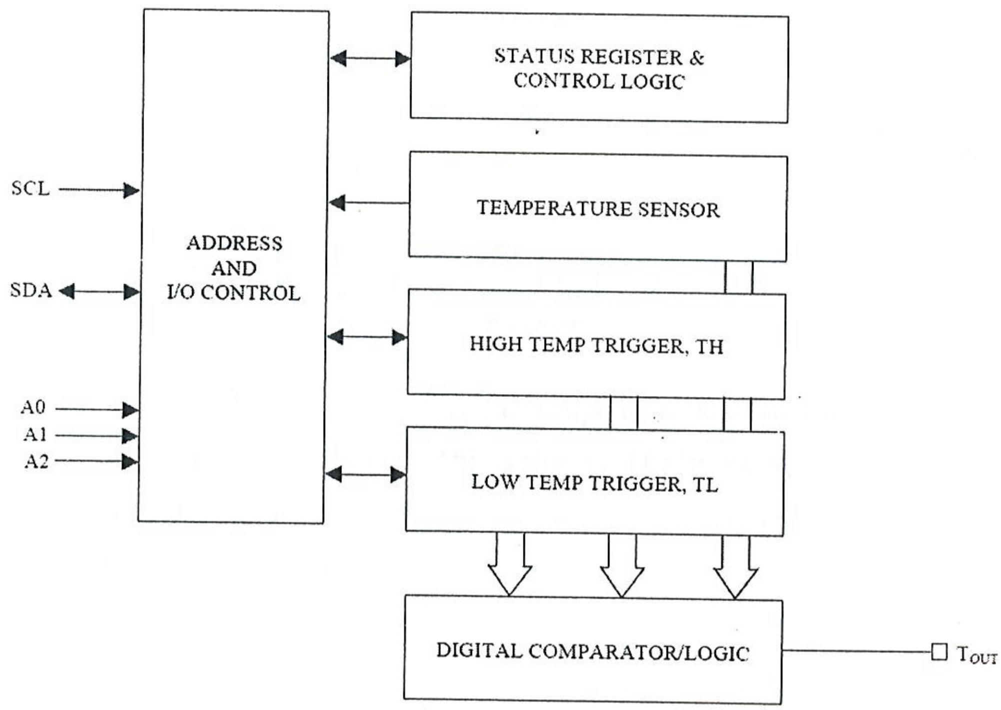


FIGURE-2.2 Functional Block Diagram

5) Temperature/DataRelationships

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	01111101 00000000	7D00h
+25°C	00011001 00000000	1900h
+½°C	00000000 10000000	0080h
+0°C	00000000 00000000	0000h
-½°C	11111111 10000000	FF80h
-25°C	11100111 00000000	E700h
-55°C	11001001 00000000	C900h

TABLE-2.1 Temperature Relationships

Since data is transmitted over the 2-wire bus MSB first, temperature data may be written to/read from the DS1621 as either a single byte (with temperature resolution of 1_C) or as two bytes. The second byte would contain the value of the least significant (0.5_C) bit of the temperature reading as shown in Table 1. Note that the remaining 7 bits of this byte are set to all "0"s.

Temperature is represented in the DS1621 in terms of a ½_C LSB, yielding the following 9-bit format

TEMPERATURE, TH, and TL FORMAT

MSB								LSB							
1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0

T = -25°C

Higher resolutions may be obtained by reading the temperature and truncating the 0.5_C bit (the LSB) from the read value. This value is TEMP_READ. A Read Counter command should be issued to yield the COUNT_REMAIN value. The Read Slope command should then be issued to obtain the COUNT_PER_C value. The higher resolution temperature may be then be calculated by the user using the following:

$$TEMPERATURE = TEMP\_READ - 0.25 + \frac{(COUNT\_PER\_C - COUNT\_REMAIN)}{COUNT\_PER\_C}$$

The DS1621 can be programmed to perform continuous consecutive conversions (continuous-conversion mode) or to perform single conversions on command (one-shot mode). The conversion mode is programmed through the 1SHOT bit in the configuration register.

In continuous conversion mode, the DS1621 begins continuous conversions after a Start Convert T command is issued. Consecutive conversions continue to be performed until a Stop Convert T command is issued, at which time the device goes into a low-power idle state. Continuous conversions can be restarted at any time using the Start Convert T command.

In one-shot mode, the DS1621 performs a single temperature conversion when a Start Convert T command is issued. When the conversion is complete, the device enters a low-power idle state and remains in that state until a single temperature conversion is again initiated by a Start Convert T command.

This covers the theory for temperature sensor IC i.e. DS1621

Now we will be discussing the next part i.e. a Heartbeat Sensor.

Part 2: Heartbeat Sensor

Firstly we tried to implement a heartbeat sensor based on a pair of LED and LDR but it didn't give appropriate readings due to interference of light and other surroundings and the circuit for that is as given:

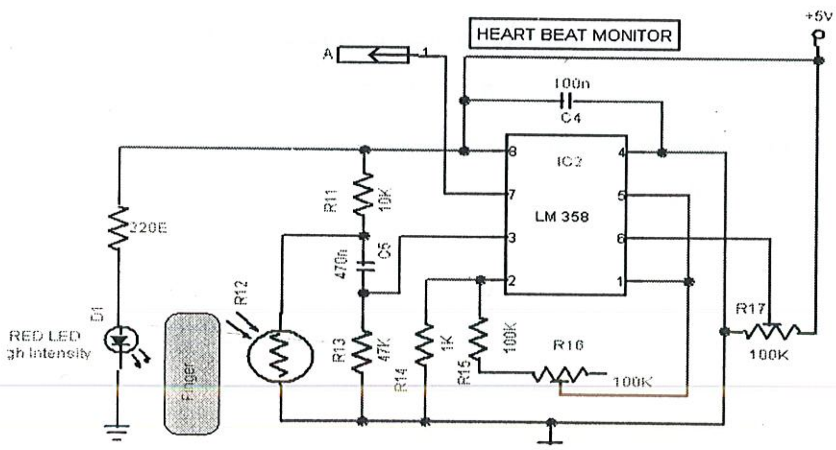


FIGURE-2.3 Heartbeat Monitor (LED and LDR based)

Thus we came up with an idea of using a heartbeat sensor based on IR LED and Phototransistor.

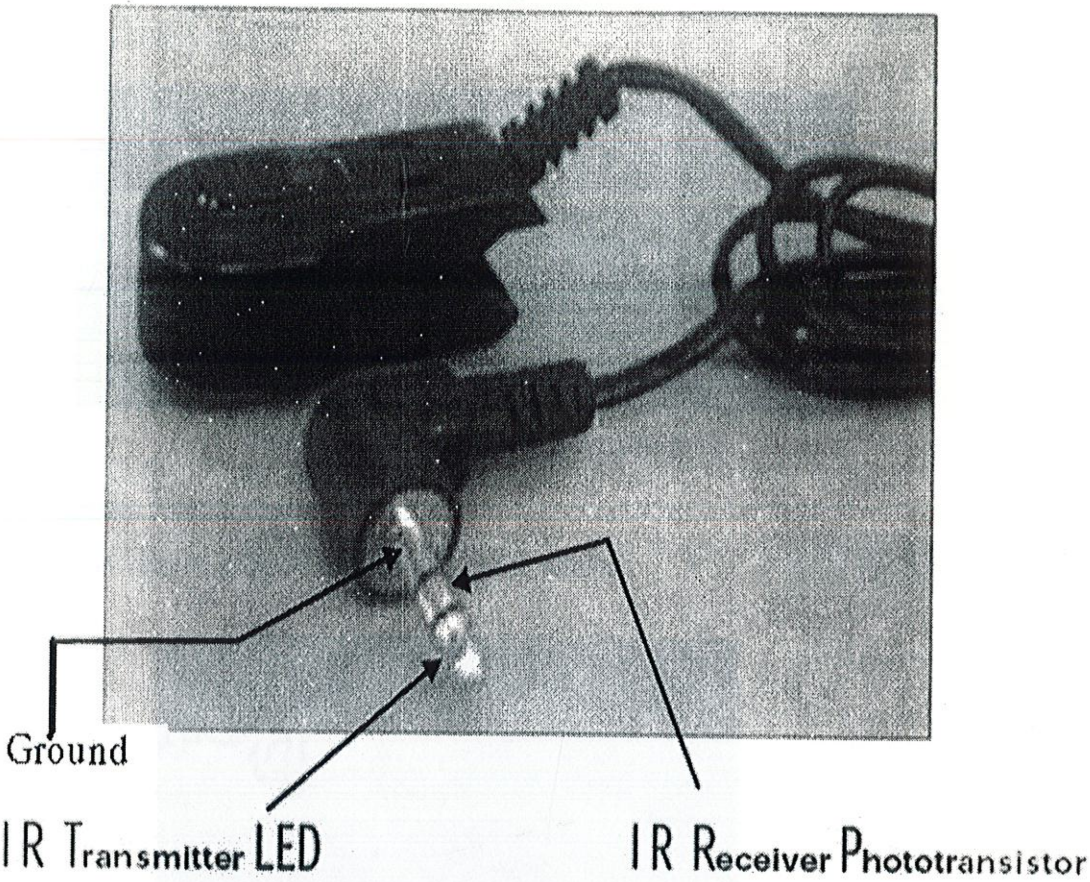


FIGURE-2.4 Heartbeat Sensor

These sensor clips consist of only a pair of IR photo transistor and receiver LEDs enclosed in a specially designed plastic clip housing and cable with stereo jack plug. The probe clip alone does not have any other circuit .To use the probe you must have some suitable application circuit or compatible pulse rate monitor.

It has 2 types of applications

1) Finger Clip Application

The sensor can be clipped on person's finger as shown.

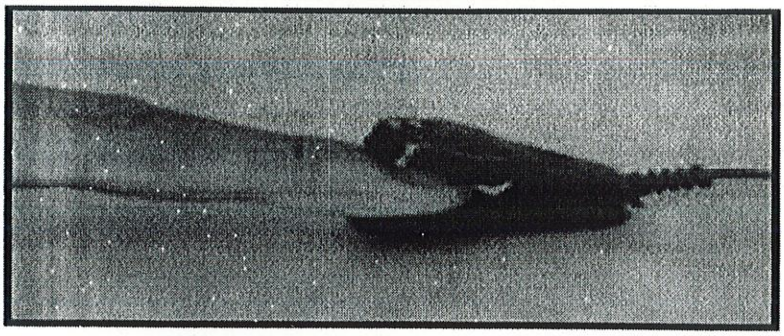


FIGURE-2.5 Finger clip Application

2) Earlobe Application

The sensor can be clipped on person's ear as shown.

FIGURE-2.6 Earlobe Application

This ends the description of the 2 sensors used in our project.

CHAPTER 3
Hardware Implementation

It is divided basically in 2 parts:

1) Transmitter Section
2) Receiver Section

Firstly we will look upon the transmitter section.

3.1 TRANSMITTER SECTION

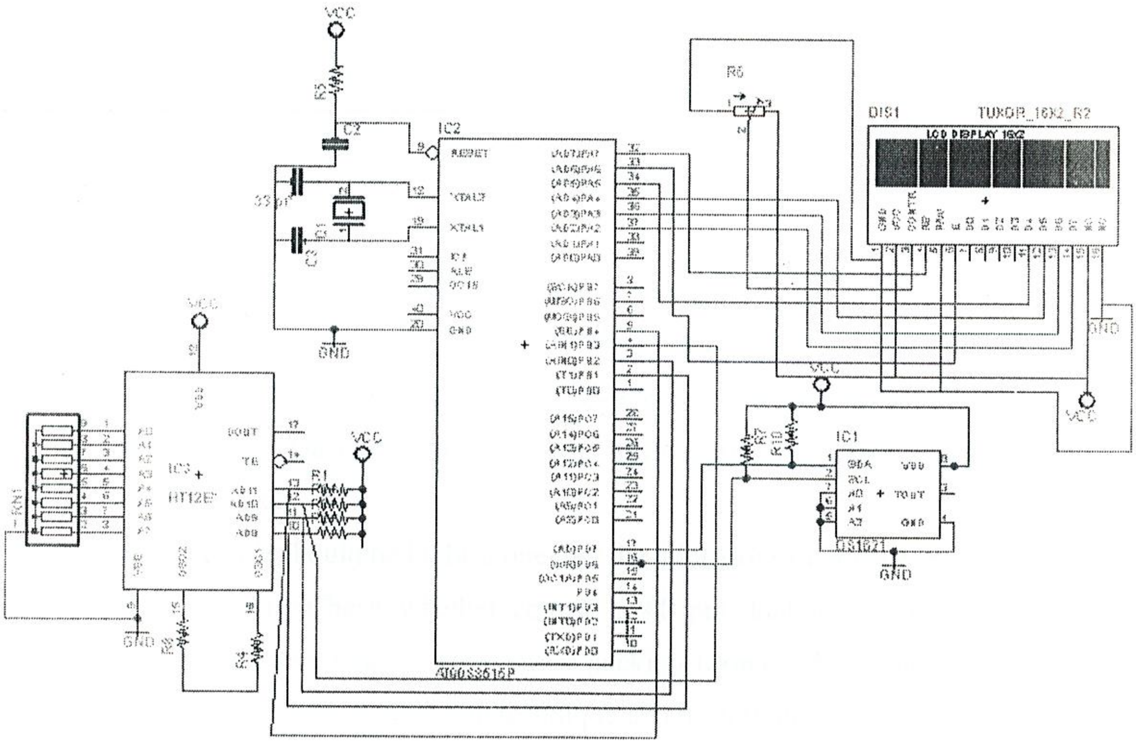


FIGURE-3.1 Transmitter Section

Description:

In our transmitter section we have basically following components:

1) ATmega8515
2) HT12E(Encoder IC)
3) DS1621(Temperature Sensor IC)
4) Heartbeat Sensor
5) LCD display(16*2)
6) RF Transmitter

We will now discuss these components in detail.

1) ATmega8515 Microcontroller

The criteria in choosing microcontrollers are as follows:

1. Meeting the computing needs of the tasks at hand efficiently and cost effectively
2. Availability of software development tools such as compilers, assemblers and debuggers.
3. Wide availability and reliable sources of microcontrollers.

Following are the criteria for selecting a microcontroller:

1. The first and the foremost criteria in choosing a microcontroller is that it must meet the task at hand effectively. Among other considerations in this category are:
   - Speed-It should be highest one that the microcontroller supports
   - Packaging-Check whether comes in 40 pin dual in line package or quad flat package or some other packing format. This is important in terms of space assembling technique and prototyping the end product.
   - Power consumption- This is especially critical for battery powered products.
   - The amount of RAM and ROM available on the chip.
   - The number of I/O pins and the timers available on the chip.
2. The second criterion in choosing a microcontroller is how easy it is in developing products around it. Key considerations include the availability of an assembler, debugger code efficient C language compiler, emulator, technical support and both in house and outside expertise.

3. The third criterion in choosing a microcontroller is its ready availability in needed quantities both at present and in future. For some designers this is even more important than first two criteria.

Since ATmega8515 is meeting all these requirements considerably well we have chosen this microcontroller.

1) Pin Diagram:

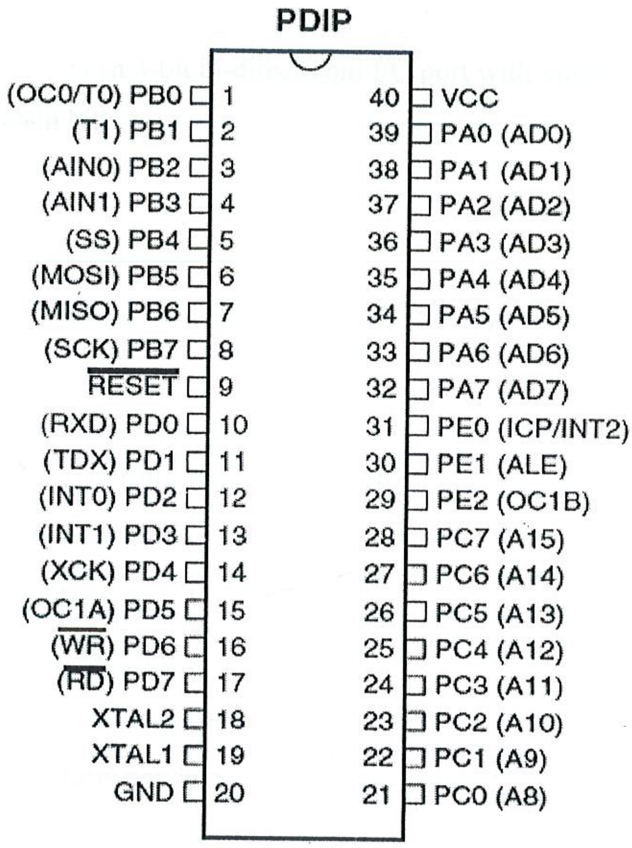


FIGURE-3.2 ATmega8515.Pin Diagram

2) Pin Description:

VCC - Digital supply voltage.

GND - Ground.

Port A (PA7, PA0) - Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

Port B (PB7,PB0) - Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

Port C (PC7..PC0) - Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit)

Port D (PD7..PD0) - Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

Port E (PE2..PE0) - Port E is an 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

RESET - Reset input.

XTAL1 - Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2 - Output from the inverting Oscillator amplifier.

2) HT12E(Encoder)

Features:

1) Operating voltage 2.4V~12V
2) Low power and high noise immunity CMOS technology.
3) Minimum 4 transmission words.
4) Built-in oscillator needs only 5% resistor.
5) Data code has positive polarity.
6) Minimal external components.

General Description:

The 2^{12} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12_N data bits. Each address/ data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a TE trigger on the HT12E further enhances the application flexibility of the 2^{12} series of encoders.

Pin Diagram:

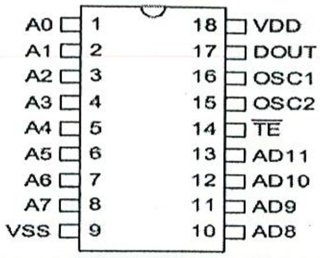


FIGURE-3.3 Pin Diagram of HT12E

Pin Description:

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS

TABLE-3.1 Pin Description of HT12E

Functional Description:

The 212 series of encoders begin a 4-word transmission cycle upon receipt of a transmission enable. This cycle will repeat itself as long as the transmission enable (TE or D8~D11) is held low. Once the transmission enable returns high the encoder output completes its final cycle and then stops as shown below.

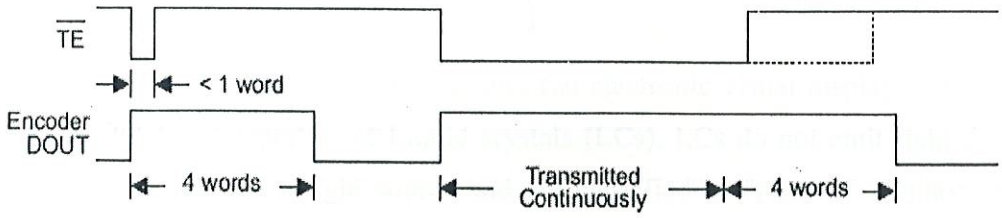


FIGURE-3.4 Functional Cycle

Flowchart:

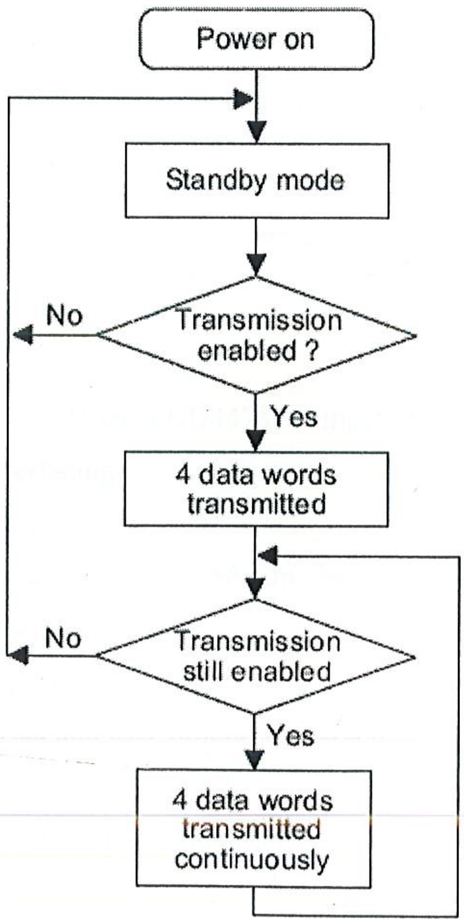


FIGURE-3.5 Flowchart for working of HT12E

3) **Temperature Sensor IC (DS1621)**

As discussed in previous section.

4) **Heartbeat Monitor**

As discussed in previous section.

5) **LCD Display**

A **liquid crystal display (LCD)** is a thin, flat electronic visual display that uses the light modulating properties of Liquid crystals (LCs). LCs do not emit light directly. LCDs therefore need a light source and are classified as "passive" displays. Some types can use ambient light such as sunlight or room lighting. There are many types of LCDs that are designed for both special and general uses. They can be optimized for static text, detailed still images, or dynamic, fast-changing, video content. They are used in a wide range of applications including: computer monitors, television, instrument panels, aircraft cockpit displays, signage, etc. They are common in consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones.

The LCD we are using in our project is a 16*2 Alphanumeric LCD.

Characteristics:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols
- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

The LCD that we are using looks like as shown in figure 3.7

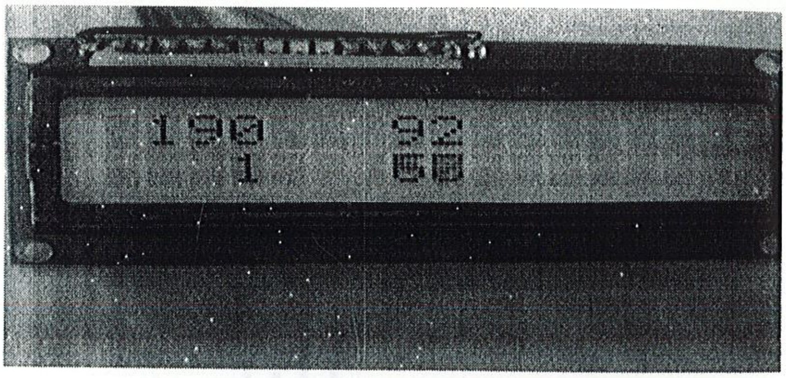


FIGURE-3.6 LCD

Pin Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	$R/\overline{W}$	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	NOT CONNECTED
16	NC	-	NOT CONNECTED

TABLE-3.2 Pin Specifications of LCD

6) RF Transmitter

The transmitter output is up to 8mW at 433.92MHz with a range of approximately 400 foot (open area) outdoors. Indoors, the range is approximately 200 foot.

The TWS-434 transmitter accepts both linear and digital inputs, can operate from 1.5 to 12 Volts-DC, and makes building a miniature hand-held RF transmitter very easy. The TWS-434 is approximately the size of a standard postage stamp.

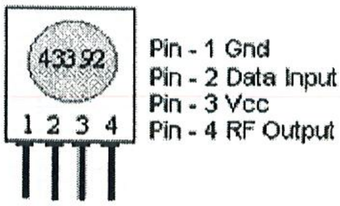


FIGURE-3.7 RF Transmitter pin diagram

So this ends the description of Transmitter section of our project.

3.2RECEIVERSECTION

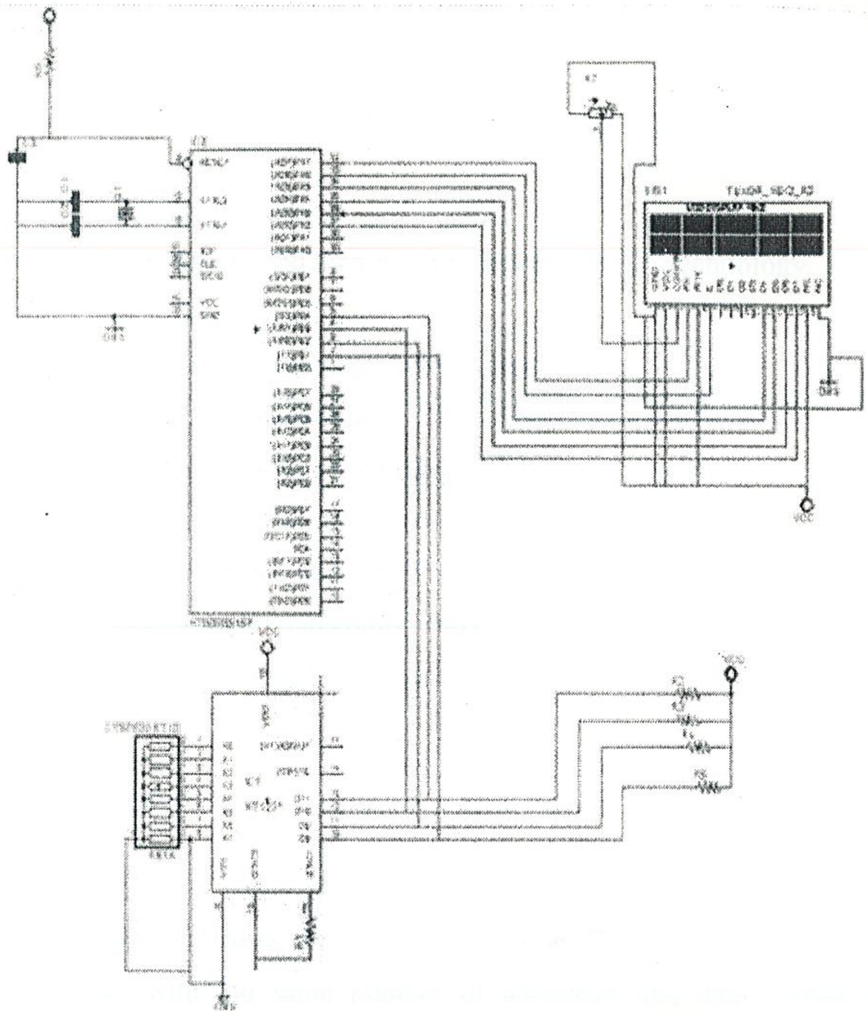


FIGURE-3.8 Receiver Schematic

3.2.1Description

In our receiver section we have basically following components:

1) ATmega8515 Microcontroller
2) HT12D(Decoder IC)
3) LCD display(16*2)
4) RF Receiver

We will now discuss these components in detail

1) ATmega8515

As discussed above

2) HT12D(Decoder IC)

Features:

1) Operating voltage: 2.4V~12V
2) Low power and high noise immunity CMOS technology
3) Low standby current
4) Capable of decoding 12 bits of information
5) Binary address setting
6) Received codes are checked 3 times
7) Built-in oscillator needs only 5% resistor
8) Easy interface with an RF or an infrared transmission medium
9) Minimal external components
10) Pair with Holtek's 2_{12} series of encoders

General Description:

The 2_{12} decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2_{12} series of encoders .For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen. The decoders receive serial addresses and data from a programmed 2_{12} series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. The 2_{12} series of decoders are capable of decoding informations that consist of N bits of address and 12_N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

Pin Diagram:

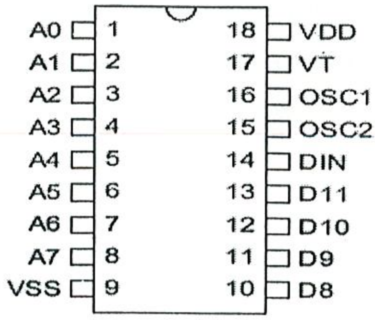


FIGURE-3.9 Pin Diagram of HT12D

Pin Description:

Pin Name	I/O	Internal Connection	Description
A0~A11 (HT12F)	I	NMOS Transmission Gate	Input pins for address A0~A11 setting These pins can be externally set to VSS or left open.
A0~A7 (HT12D)			Input pins for address A0~A7 setting These pins can be externally set to VSS or left open.
D8~D11 (HT12D)	O	CMOS OUT	Output data pins, power-on state is low.
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	Oscillator	Oscillator input pin
OSC2	O	Oscillator	Oscillator output pin
VSS	—	—	Negative power supply, ground
VDD	—	—	Positive power supply

TABLE-3.3 Pin Description

Functional Description:

The 212 series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 212 series of encoders. The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12_N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12_N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received. The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

Flowchart:

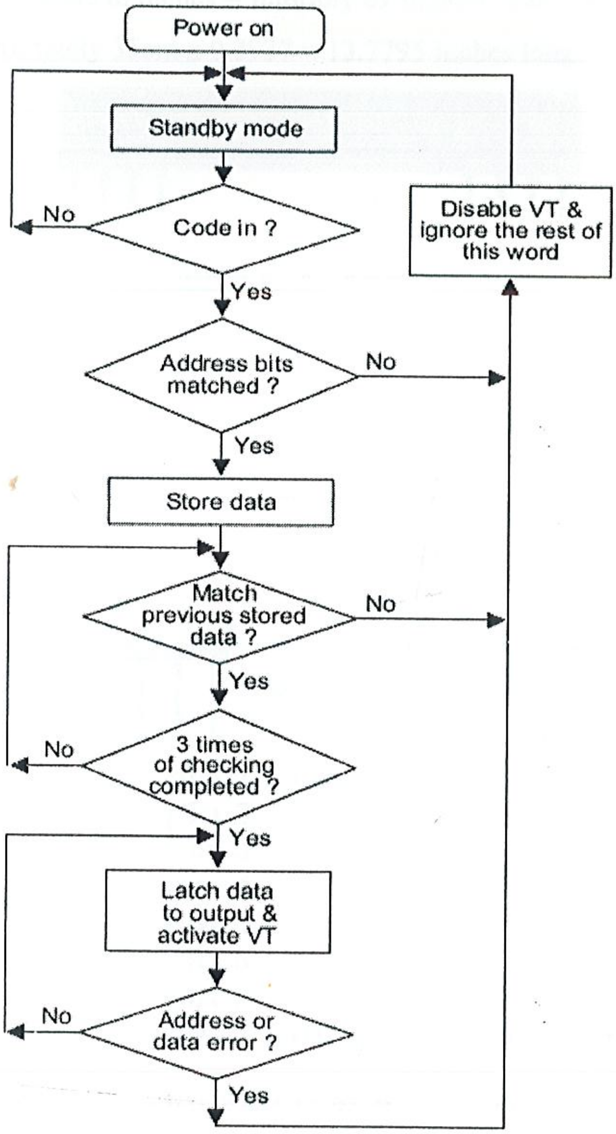


FIGURE-3.10 Flowchart for working of HT12D

3) **LCD Display**

As discussed in the transmitter section

4) **RF Receiver**

The Receiver also operates at 433.92MHz, and has a sensitivity of 3uV. The RWS-434 receiver operates from 4.5 to 5.5 volts-DC, and has both linear and digital outputs.

For maximum range, the recommended antenna should be approximately 35cm long. To convert from centimeters to inches -- multiply by 0.3937. For 35cm, the length in inches will be approximately 35cm x 0.3937 = 13.7795 inches long.

FIGURE-3.11 RF Receiver

Pin description of RF Receiver

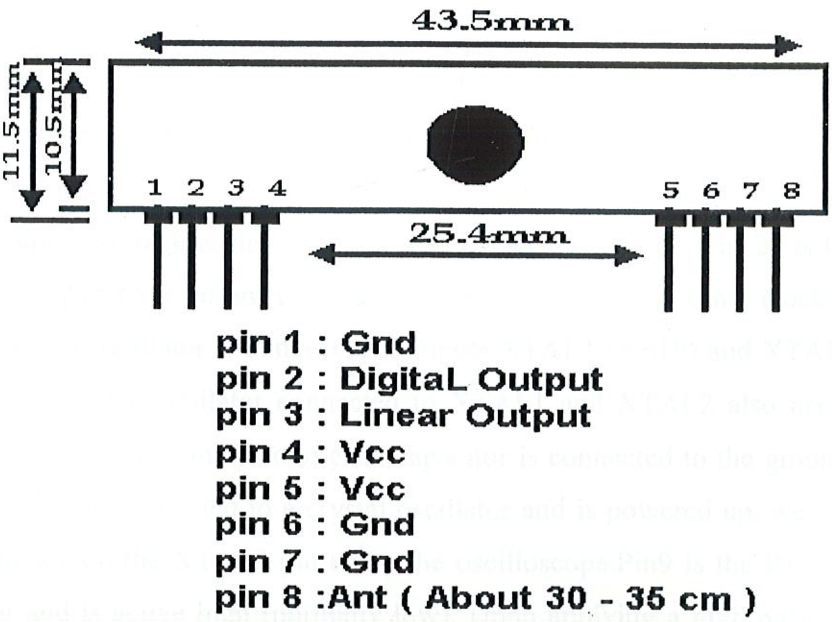


FIGURE-3.12 RF Receiver Pin out

So this ends the description of Receiver section of our project.

CHAPTER-4

Testing and software implementation

4.1 Testing Of Microcontroller

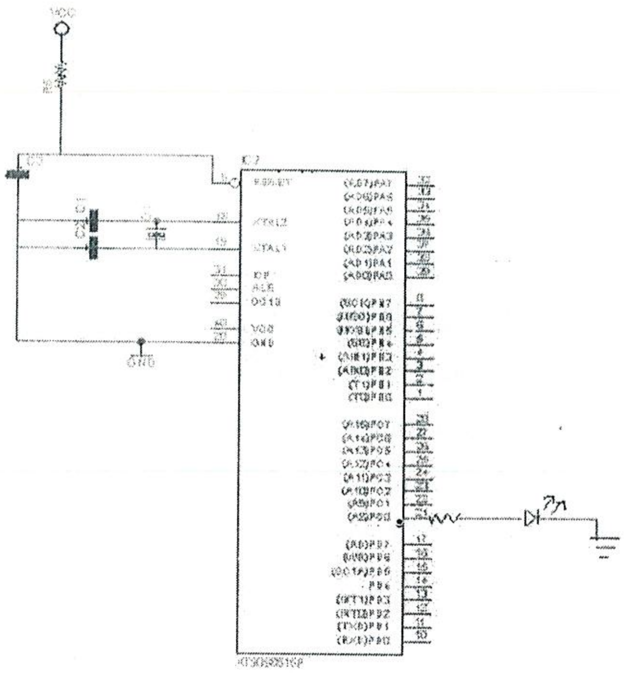


FIGURE-4.1 Testing of microcontroller

Pin 40 provides supply voltage to the chip. The voltage source is +5V. Pin 20 is the ground. The atmega8515 have an on- chip oscillator but require an external clock to run it. A quartz crystal oscillator is connected to inputs XTAL1 (pin19) and XTAL2 (pin18). The quartz crystal oscillator connected to XTAL1 and XTAL2 also needs two capacitors of 22pF value. One side of each capacitor is connected to the ground. When the atmega8515 is connected to a crystal oscillator and is powered up, we can observe the frequency on the XTAL2 pin using the oscilloscope.Pin9 is the RESET pin. It is an input and is active high (normally low). Upon applying a high pulse to this pin, the microcontroller will reset and terminate all activities. EA pin must be connected to Vcc to indicate that the code is stored internally.(On-chip ROM to store programs).

For testing of microcontroller a series combination of resistor of 470 ohm and a LED is connected to port 3.0 of the microcontroller. A program for blinking of LED with 50%duty cycle was stored in the on chip ROM and blinking of LED along with a square wave of 50% duty cycle was observed on oscilloscope.

4.2 TESTING OF LCD WITH MICROCONTROLLER

As shown in figure 16*2 line LCD is used for the display, connected to port 1 of microcontroller. Liquid crystal displays (LCD) are widely used in recent years as compares to LEDs, as the ability to display numbers, characters and graphics, incorporation of a refreshing controller into the LCD, their by relieving the CPU of the task of refreshing the LCD and also the ease of programming for characters and graphics. Before sending commands or data to the LCD module, the Module must be initialized. Once the initialization is complete, the LCD can be written to with data or instructions as required. Each character to display is written like the control bytes, except that the "RS" line is set. During initialization, by setting the "S/C" bit during the "Move Cursor/Shift Display" command, after each character is sent to the LCD, the cursor built into the LCD will increment to the next position (either right or left). Normally, the "S/C" bit is set (equal to "1").LCD pins brief description described below.

V_{CC}, V_{SS}, V_{EE}

The voltage V_{CC} and V_{SS} provided by +5V and ground respectively while V_{EE} is used for controlling LCD contrast. Variable voltage between Ground and Vcc is used to specify the contrast (or "darkness") of the characters on the LCD screen.

RS (register select)

If RS=0, the instruction command code register is selected, then allowing to user to send a command such as clear display, cursor at home etc.. If RS=1, the data register is selected, allowing the user to send data to be displayed on the LCD.

R/W (read/write)

The R/W (read/write) input allowing the user to write information from it. R/W=1, when it read and R/W=0, when it writing.

EN (enable)

The enable pin is used by the LCD to latch information presented to its data pins. When data is supplied to data pins, a high power, a high-to-low pulse must be applied to this pin in order to for the LCD to latch in the data presented at the data pins.

D0-D7 (data lines)

The 8-bit data pins, D0-D7, are used to send information to the LCD or read the contents of the LCD's internal registers. To displays the letters and numbers, we send ASCII codes for the letters A-Z, a-z, and numbers 0-9 to these pins while making RS =1. There are also command codes that can be sent to clear the display or force the cursor to the home position or blink the cursor.

We also use RS =0 to check the busy flag bit to see if the LCD is ready to receive the information. The busy flag is D7 and can be read when R/W =1 and RS =0, as follows: if R/W =1 and RS =0, when D7 =1(busy flag =1), the LCD is busy taking care of internal operations and will not accept any information. When D7 =0, the LCD is ready to receive new information.

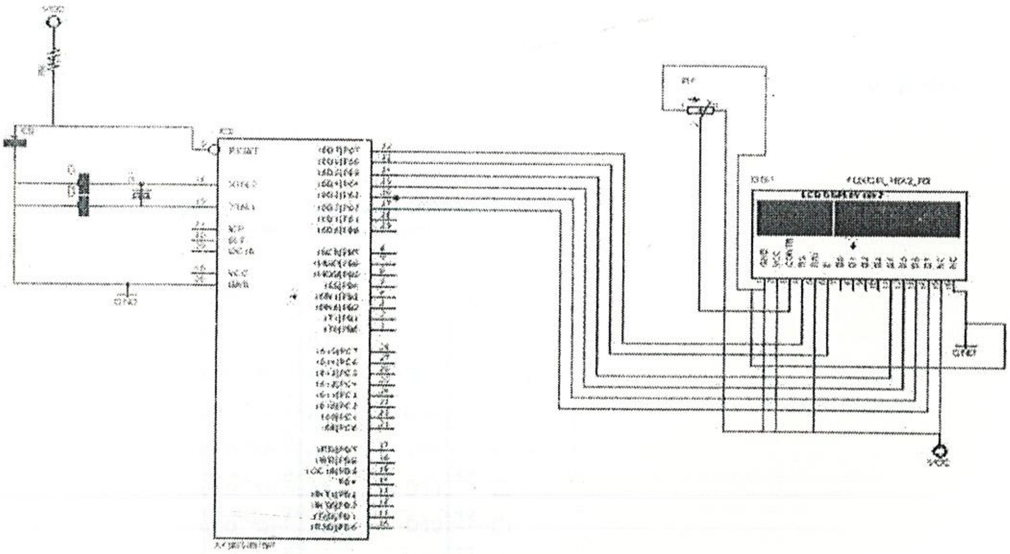


FIGURE-4.2: Testing Circuit of LCD with Microcontroller

4.3 TESTING OF TRANSMITTER & RECIEVER CIRCUIT

4.3.1 Transmitter

The data sent to microcontroller is processed and sent to encoder and the HT12E send this data to the RF transmitter. We have checked this manually the working of the transmitter by the following circuit diagram

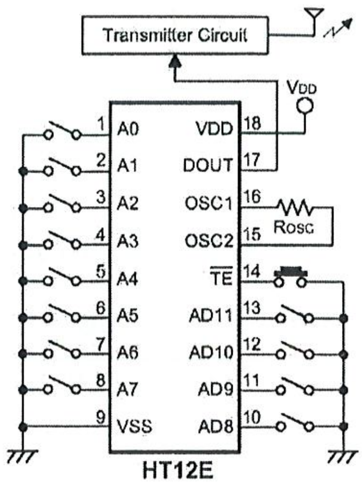


FIGURE-4.3 Block diagram of Encoder circuit

4.3.2 Receiver

The data received by the RF receiver is transmitted to the decoder HT12D. So we have checked the working of the encoder and decoder by directly connecting the wires as shown in the diagram

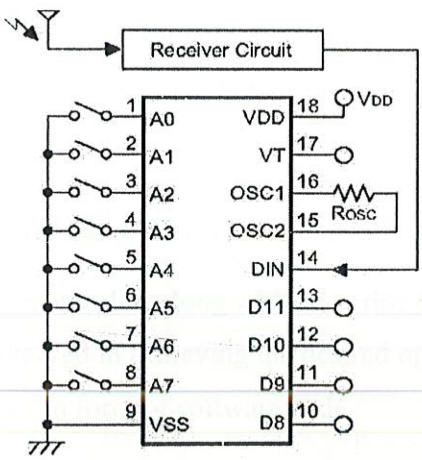


FIGURE- 4.4 Block diagram of Decoder circuit

4.4 SOFTWARE

The software design is a key element in the development of a project. For visualization of the different parameter on the LCD display, codevision AVR software is developed. The microcontroller chosen for the development of the system was the atmega8515. The atmega8515 has 4K of program memory and has the capability to write to its own memory. The use of a FLASH device for development also provides the option to use FLASH microcontrollers in the final design making the system fully upgradable. This allows modification of the microcontroller software to support expansion.

4.4.1 CODEVISION AVR SOFTWARE

Codevision AVR is Automatic Program Generator and In-System Programmer for the Atmel AVR family of microcontrollers it has easy to use Integrated Development Environment and ANSI C compatible Compiler. It has Editor with auto indentation, syntax highlighting for both C and AVR assembler, function parameters and structure/union members autocomplete.

The programming of the atmega8515 is done in embedded -C language. It is used because of the following advantages:

- To speed up computer operation.
- To reduce the size of the program.
- To write programs for special situations.
- To have better understanding of the computer operation.

4.5 SOFTWARE CODE

The microcontroller along with its various interfaces requires software to work on. The logic involved in achieving the desired operation has been carefully prepared and is noted down in form of software code.

4.5.1 Transmitter Section Code

```
#include <mega8515.h>

#define RXB8 1

#define TXB8 0

#define UPE 2

#define OVR 3

#define FE 4

#define UDRE 5

#define RXC 7

#define FRAMING_ERROR (1<<FE)

#define PARITY_ERROR (1<<UPE)

#define DATA_OVERRUN (1<<OVR)

#define DATA_REGISTER_EMPTY (1<<UDRE)

#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer

#define RX_BUFFER_SIZE 8

char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE<256

unsigned char rx_wr_index,rx_rd_index,rx_counter;

#else

unsigned int rx_wr_index,rx_rd_index,rx_counter;

#endif
```

```
// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;
// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
  {
  rx_buffer[rx_wr_index]=data;
  if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
  if (++rx_counter == RX_BUFFER_SIZE)
    {
    rx_counter=0;
    rx_buffer_overflow=1;
    };
  };
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
```

```
#define _ALTERNATE_GETCHAR_

#pragma used+

char getchar(void)

{

char data;

while (rx_counter==0);

data=rx_buffer[rx_rd_index];

if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;

#asm("cli")

--rx_counter;

#asm("sei")

return data;

}

#pragma used-

#endif


// USART Transmitter buffer

#define TX_BUFFER_SIZE 8

char tx_buffer[TX_BUFFER_SIZE];


#if TX_BUFFER_SIZE<256

unsigned char tx_wr_index,tx_rd_index,tx_counter;
```

```
#else
unsigned int tx_wr_index,tx_rd_index,tx_counter;
#endif
// USART Transmitter interrupt service routine
interrupt [USART_TXC] void usart_tx_isr(void)
{
if (tx_counter)
  {
  --tx_counter;
  UDR=tx_buffer[tx_rd_index];
  if (++tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
  };
}
#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter == TX_BUFFER_SIZE);
#asm("cli")
if (tx_counter || ((UCSRA & DATA_REGISTER_EMPTY)==0))
```

```
  {
  tx_buffer[tx_wr_index]=c;
  if (++tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
  ++tx_counter;
  }
else
  UDR=c;
#asm("sei")
}
#pragma used-
#endif
// I2C Bus functions
#asm
  .equ __i2c_port=0x12 ;PORTD
  .equ __sda_bit=4
  .equ __scl_bit=3
#endasm
#include <i2c.h>
// DS1621 Thermometer/Thermostat functions
#include <ds1621.h>
// Alphanumeric LCD Module functions
#asm
```

```
 .equu __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>
#include<stdio.h>
#include<delay.h>
 int count,hbeat,flag;
unsigned int msec,sec;
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
 count++;
}
// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer 1 value
TCNT1H=0xD8;
TCNT1L=0xF0;
// Place your code here
msec++;
if(msec==99) {msec=0; sec++;}
if(sec==30)
```

```
{
sec=0;
hbeat=count;
hbeat=hbeat*2;
count=0;
flag=1;
}
}
// Declare your global variables here
void main(void)
{
unsigned int temp=0;
unsigned char text[16],text1[16];
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xFF;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;
// Port E initialization
// Func2=In Func1=In Func0=In
// State2=T State1=T State0=T
PORTE=0x00;
DDRE=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
```

```
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1000.000 kHz
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x02;
TCNT1H=0xD8;
TCNT1L=0xF0;
ICR1H=0x00;
ICR1L=0x00;
```

```
OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// External Interrupt(s) initialization

// INT0: On

// INT0 Mode: Falling Edge

// INT1: Off

// INT2: Off

GICR|=0x40;

MCUCR=0x02;

EMCUCR=0x00;

GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x80;

// USART initialization

// Communication Parameters: 8 Data, 1 Stop, No Parity

// USART Receiver: On

// USART Transmitter: On

// USART Mode: Asynchronous

// USART Baud Rate: 600

UCSRA=0x00;
```

```
UCSRB=0xD8;
UCSRC=0x86;
UBRRH=0x03;
UBRRL=0x40;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
// I2C Bus initialization
i2c_init();
// DS1621 Thermometer/Thermostat initialization
// tlow: 50°C
// thigh: 55°C
// Tout polarity: 0
ds1621_init(0,50,55,0);
// LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("Welcome to BPMS");
 lcd_gotoxy(0,1);
lcd_putsf("Made by Harinder");
delay_ms(2000);
```

```
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Doctor XYZ");
 lcd_gotoxy(0,1);
 lcd_putsf("Patient no.1");
 delay_ms(2000);
lcd_clear();
// Global enable interrupts
#asm("sei")
while (1)
      {
     temp=ds1621_temperature_10(0x00);
      sprintf(text1,"%4u %4u",sec,msec);
      sprintf(text,"%4u %4u",temp,hbeat);
      lcd_gotoxy(0,0);
      lcd_puts(text);
      lcd_gotoxy(0,1);
      lcd_puts(text1);
      if (flag==1)
      {
      flag=0;
      printf("t%u#\n\r",temp);
```

```
delay_ms(300);

printf("p%u#\n\r",hbeat);

}

if(temp >= 205)

PORTB.2=1;

};

}
```

4.5.2 Receiver Section Code

```
#include <mega8515.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
   #asm
      .equ __lcd_port=0x1b
   #endasm
   #include <lcd.h>
#include<stdio.h>
#include<delay.h>
 unsigned int temp_p=0,hbeat;
float temp_dis,dis=12.3;
void get_val();
// External Interrupt 0 service routine
char ch=0x0D;
int start_rs232,data_ready;
#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
```

```
#define DATA_OVERRUN (1<<OVR)

#define DATA_REGISTER_EMPTY (1<<UDRE)

#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer

#define RX_BUFFER_SIZE 16

char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE<256

unsigned char rx_wr_index,rx_rd_index,rx_counter;

#else

unsigned int rx_wr_index,rx_rd_index,rx_counter;

#endif

// This flag is set on USART Receiver buffer overflow

bit rx_buffer_overflow;

// USART Receiver interrupt service routine

interrupt [USART_RXC] void usart_rx_isr(void)

{

char status,data;

status=UCSRA;

data=UDR;

//UDR=data;

if (((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)&& (start_rs232==1)&&(data!=ch))

 {

//  UDR=data;

  rx_buffer[rx_wr_index]=data;

  if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
```

```
if (++rx_counter == RX_BUFFER_SIZE)
  {
  rx_counter=0;
  rx_buffer_overflow=1;
  };
};
if(data==ch)
{
start_rs232=0;
rx_buffer[rx_wr_index]=255;
rx_wr_index--;
rx_counter--;
data_ready=1;
}
}
#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#asm("cli")
```

```
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif
#include <stdio.h>
 void reset_rs232()
{
start_rs232=1;
str_clear(rx_buffer,16);
rx_wr_index=0;
rx_counter=0;
rx_wr_index=0;
}
void getdata(char *s)
{    int i,x,y;
   int len=0,j=0;
   char text[16],temp[16],pul[16];
   len=strlen(s);
   for(i=0;i<len;i++)

{
   text[i]=*s;
         if (text[0]=='t')
          {
```

```
            if (text[i]== '#')
                {
                x=atoi(temp);
                temp_p=x ;
                }
            else if(isdigit(text[i])) {temp[j]=text[i];j++;}
          }
          else
         if (text[0]=='p')
         {
          if (text[i]== '#')
               {
               y=atoi(pul);
               hbeat=y;
               }
          else if(isdigit(text[i])) {pul[j]=text[i];j++;}
         } s++;
      }
}
// Declare your global variables here
void main(void)

{
unsigned char text[16],temp_t[16]="p12345#",text1[16];
// Declare your local variables here
PORTB=0x00;
```

```
DDRB=0xFF;
// USART initialization
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x03;
UBRRL=0x40;
/*UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x04;
UBRRL=0x7F;
*/
// LCD module initialization
lcd_init(16);
lcd_putsf("Welcome to BPMS");
delay_ms(50);
lcd_gotoxy(0,1);
lcd_putsf("Made by Harinder");
delay_ms(1000);
lcd_clear();
// Global enable interrupts
#asm("sei")
//start_rs232=1;
reset_rs232();
while (1)
```

```
{
if(temp_p >= 205)        PORTB.2=1;
sprintf(text1,"PULSE:%4u",hbeat);
sprintf(text ," TEMP:%2.1f",temp_dis);
lcd_gotoxy(0,0);
lcd_puts(text);
lcd_gotoxy(0,1);
lcd_puts(text1);
//Reciever
if(data_ready==1)
{
data_ready=0;
get_val();
reset_rs232();
temp_dis=(((float)temp_p)/10);
}
};
}
```

FUTURE PROSPECTS OF THE PROJECT

Despite lots of research in this field of wireless monitoring of patient by doctors, there has been very little effort in actual implementation of the concept which provides ample scope for the further developments of this project. A concept called mHealth a term used for the practice of medical and public health, supported by mobile devices in reference to using mobile communication devices like mobile phones can be implemented by using the basic concept of wireless transmission of patient's health status.

This concept of m-Health can be integrated in larger scale implementations like public health management systems where database of a large number of patients can be stored and monitored through wireless communication which can also be retrieved to maintain a proper record of the patient's health status and previous record. m-Health is a subdivision of e-Health Which deals with the usage of Information Coding Theory(ICT) like computers, communication satellites, patient monitors for health services and information. m-Health applications include the use of mobile devices in collecting community and clinical health data, delivery of healthcare information to practitioners, researchers, and patients, real-time monitoring of patient vital signs, and direct provision of care.

While m-Health certainly has application for industrialized nations, the field has emerged in recent years as largely an application for developing countries, stemming from the rapid rise of mobile phone penetration in low-income nations. The field, then, largely emerges as a means of providing greater access to larger segments of a population in developing countries, as well as improving the capacity of health systems in such countries to provide quality healthcare.

Within the m-Heath space, projects operate with a variety of objectives, including increased access to healthcare and health-related information (particularly for hard-to-reach populations); improved ability to diagnose and track diseases; timelier, more actionable public health information; and expanded access to ongoing medical education and training for health workers

The motivation behind the development of the m-Health field arises from two factors. The first factor concerns the myriad constraints felt by healthcare systems of developing nations. These constraints include high population growth a high burden of disease prevalence, low health care workforce,large numbers of rural inhabitants, and limited financial resources to support healthcare infrastructure and health information systems. The second factor is the recent rapid rise in mobile phone penetration in developing countries to large segments of the healthcare workforce, as well as the population of a country as a whole. With greater access to mobile phones to all segments of a country, including rural areas, the potential of lowering information and transaction costs in order to deliver healthcare improves.

The combination of these two factors has motivated much discussion of how greater access to mobile phone technology can be leveraged to mitigate the numerous pressures faced by developing countries' healthcare systems.

We can also envisage concepts like **Telehealth** wherein the delivery of health-related services and information via telecommunications technologies. Telehealth delivery could be as simple as two health professionals discussing a case over the telephone, or as sophisticated as using videoconferencing between providers at facilities in two countries, or even as complex as robotic technology.

Clinical Uses of Telehealth Technologies

- Transmission of medical images for diagnosis(often referred to as store and forward telehealth)
- Groups or individuals exchanging health services or education live via videoconference (real-time telehealth)
- Transmission of medical data for diagnosis or disease management (sometimes referred to as remote monitoring)
- Advice on prevention of diseases and promotion of good health by patient monitoring and follow-up.
- Health advice by telephone in emergent cases (referred to as teletriage)

Telemedicine is simple as two health professionals discussing a case over the telephone, or as complex as using satellite technology and video conferencing equipment to conduct a real-time consultation between medical specialists in two different countries. Telemedicine is most beneficial for populations living in isolated communities and remote regions and is currently being applied in virtually all medical domains. Specialties that use telemedicine often use a "tele-" prefix; for example, telemedicine as applied by radiologists is called Teleradiology. Similarly telemedicine as applied by cardiologists is termed as telecardiology; etc .Telemedicine is also useful as a communication tool between a general practitioner and a specialist available at a remote location.

One more future prospect of the project is its utility in monitoring the status of the patient by doctors when any one or both of them are on the move. Patient being taken to hospital in an ambulance and doctor travelling to the hospital can get constant updates about the patient status even when both of them are on the move is where our project can be instrumental.

CONCLUSION

The project has been successfully completed within the stipulated time frame. We have achieved the desired outputs on the LCD displays at the transmitting and the receiving ends.

The test results were very much satisfying and we have documented the entire concept under a research paper named "Proposed low cost model for implementation of remote patient monitoring in rural India" which is accepted in NIT Kurukshetra. Further prospects of this concept seem bright. The concept can be extended to implementations of Mobile Health networks development in rural India where there is severe scarcity of doctors and medical facilities. The vision is promising and absolutely possible to achieve and help rural population access better medical facilities.

BIBLIOGRAPHY

BOOKS:

- The 8051 Microcontroller And Embedded Systems Using Assembly And C(second edition),Pearson Education-2009

WEB PAGES:

- www.rickeyworld.com
- www.8051projects.com
- www.google.com
- www.wikipedia.com

RESEARCH PAPERS:

- Michael, M., Djamal, B.,Chirine, G., and Zakaria, M. "A Mediation for Web Services in a Distributed Healthcare Information System", *Proceedings of the IDEAS Workshop on Medical Information Systems: The Digital Hospital*, pp. 15-22, Sept. 1-4, 2004
- [2]Ng, P., Lo, L., Wells, O., Sloman, M., Peters, N., Darzi, A., Toumazou, C., and Yang, G. "Ubiquitous Monitoring Environment for Wearable and Implantable Sensors (UbiMon)" http://ubicomp.org/ubicomp2004/adjunct/posters/ng.pdf
- Chan, V., Ray, P., and Parameswaran, N. "A Multi-Agent Collaborative Framework for Mobile E-Health", *Proceedings of the 40th Hawaii International Conference on System Sciences*, Jan 3-6, USA, 20
- Schwaibold, M., Gmelin, M., Wagner, G., Schochlin, J., and Bolz, A. "Key factors for personal health monitoring and diagnosis devices". *Workshop of Mobile Computing in Medicine*, Germany, 2002.