**CHAPTER 1**

**INTRODUCTION**

* 1. **Overview of the Project:**

The main objective of this project is that human health care by means of using wireless communication .The main module consists of HEART BEAT SENSOR, TEMPERATURE SENSOR are used to know their respective conditions of the victim. After noticing the readings of the patient, when the readings are abnormal than those values are sent to the doctor through message, this helps the doctor diagnolysis the report of the victim and then suggests medicine by mobile communication or in emergency can attend him with less delay.

The Microcontroller here used is AT89S52 which is a type of reprogrammable Microcontroller Programmed.

**1.2 Requirements of the Project:**

**(I) Hard Requirements:**

* Microcontroller AT89S52
* ZIGBEE Module

**(II) Software Requirements:**

* KEIL Software
* ISP to burn the Chip

**(III) Language Used:**

* Embedded C

Generally the present paper existing is that the Measurement of Heart Rate using the RF Transmitter and Receiver and now we are improving the existing paper with the impronised way of calculating the Heart Beat by using Measurement of Heart Rate using Zigbee Transmitter and Receiver. This will give the opportunity for continuously Monotoring for the patient. Moreover it will Display the Heart Beat For every 20 seconds and it will Display in the Doctor room. This will give the continuous attention about the Patient and the sudden danger can be found immediately.

**1.3Embedded System:**

**1.3.1 Introduction to Embedded System:**

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used every day, but very few people realize that a processor and software are involved in the preparation of their lunch or dinner.

This is indirect contrast to the personal computer in the family room. It too is comprised of computer hardware and software and mechanical components (disk drives, for example). However, a personal computer is not designed to perform a specific function rather; it is able to do many different things. Many people use the term general-purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer will do wish it. One customer may use it for a network file server another may use it exclusively for playing games, and a third may use it to write the next great American novel.

Frequently, an Embedded System is a component within some larger System. For example, modern cars and trucks contain many Embedded Systems. One Embedded System controls the anti-lock brakes, other monitors and controls the vehicle's emissions, and a third displays information on the dashboard. In some cases, these Embedded Systems are connected by some sort of a communication network, but that is certainly not a requirement.

At the possible risk of confusing you, it is important to point out that a general-purpose computer is itself made up of numerous Embedded Systems. For example, my computer consists of a keyboard, mouse, video card, modem, hard drive, floppy drive, and sound card-each of which is an Embedded System. Each of these devices contains a processor and software and is designed to perform a specific function. For example, the modem is designed to send and receive digital data over analog telephone line. That's it and all of the other devices can be summarized in a single sentence as well. If an Embedded System is designed well, the existence of the processor and software could be completely unnoticed by the user of the device. Such is the case for a microwave oven, VCR, or alarm clock.

In some cases, it would even be possible to build an equivalent device that does not contain the processor and software. This could be done by replacing the combination with a custom integrated circuit that performs the same functions in hardware. However, a lot of flexibility is lost when a design is hard-cooled in this way. It is much easier, and cheaper, to change a few lines of software than to redesign a piece of custom hardware.

**1.3.2 History and Future:**

Given the definition of Embedded Systems earlier is this chapter; the first such Systems could not possibly have appeared before 1971. That was the year Intel introduced the world's first microprocessor. This chip, the 4004, was designed for use in a line of business calculators produced by the Japanese Company Busicom. In 1969, Busicom asked Intel to design a set of custom integrated circuits-one for each of their new calculator models. The 4004 was Intel's response rather than design custom hardware for each calculator, Intel proposed a general-purpose circuit that could be used throughout the entire line of calculators. Intel's idea was that the software would give each calculator its unique set of features.

The Microcontroller was an overnight success, and its use increased steadily over the next decade. Early Embedded applications included unmanned space probes, computerized traffic lights, and aircraft flight control Systems. In the 1980s, Embedded Systems quietly rode the waves of the microcomputer age and brought microprocessors into every part of our kitchens (bread machines, food processors, and microwave ovens), living rooms (televisions, stereos, and remote controls), and workplaces (fax machines, pagers, laser printers, cash registers, and credit card readers).

It seems inevitable has the number of Embedded Systems will continue to increase rapidly. Already there are promising new Embedded devices that have enormous market potential; light switches and thermostats that can be central computer, intelligent air-bag Systems that don't inflate when children or small adults are present, pal-sized electronic organizers and Personal Digital Assistants (PDAs), digital cameras, and dashboard navigation Systems. Clearly, individuals who possess the skills and desire to design the next generation of Embedded Systems will be in demand for quite some time***.***

**1.3.3 Real Time Systems:**

One subclass of Embedded is worthy of an introduction at this point. As commonly defined, a real-time System is a computer System that has timing constraints. In other words, a real-time System is partly specified in terms of its ability to make certain calculations or decisions in a timely manner. These imPortant calculations are said to have deadlines for completion. And, for all practical purposes, a missed deadline is just as bad as a wrong answer.

The issue of what if a deadline is missed is a crucial one. For example, if the real-time System is part of an airplane's flight control System, it is possible for the lives of the passengers and crew to be endangered by a single missed deadline. However, if instead the System is involved in satellite communication, the damage could be limited to a single corrupt data packet. The more severe the consequences, the more likely it will be said that the deadline is "hard" and thus, the System is a hard real-time System. Real-time Systems at the other end of this discussion are said to have "soft" deadlines.

All of the topics and examples presented in this book are applicable to the designers of real-time System who is more delight in his work. He must guarantee reliable operation of the software and hardware under all the possible conditions and to the degree that human lives depend upon three System's proper execution, engineering calculations and descriptive paperwork.

**1.3.4 Application Areas:**

Nearly 99 per cent of the processors manufactured end up in Embedded Systems. The Embedded System market is one of the highest growth areas as these Systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on.

**Consumer Appliances:**

At home we use a number of Embedded Systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc. Today’s high-tech car has about 20 Embedded Systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now becoming Embedded Systems. The palmtops are powerful Embedded Systems using which we can carry out many general-purpose tasks such as playing games and word processing.

**Office Automation:**

The office automation products using an Embedded Systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

**Industrial Automation:**

Today a lot of industries use Embedded Systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The Embedded Systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

**Medical Electronics:**

Almost every medical equipment in the hospital is an Embedded System. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonoscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

**Computer Networking:**

Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are Embedded Systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router’s function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end Systems (desktop computers) we use to access the networks, are Embedded Systems.

**Telecommunications:**

In the field of telecommunications, the Embedded Systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are Embedded Systems. The network equipment includes multiplexers, multiple access Systems, Packet Assemblers Dissemblers (PADs), sate11ite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest Embedded Systems that provide very low-cost voice communication over the Internet.

**Wireless Technologies:**

Advances in mobile communications are paving way for many interesting applications using Embedded Systems. The mobile phone is one of the marvels of the last decade of the 20’h century. It is a very powerful Embedded System that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centres are also powerful Embedded Systems.

**Insemination:**

Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all Embedded Systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio communication test set etc. are Embedded Systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming Portable facilitating easy testing and measurement in the field by field-personnel.

**Security:**

Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and store. Developing Embedded Systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are Embedded Systems. Embedded Systems find applications in every industrial segment- consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric Systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

**Finance:**

Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded as Any Time Money) machines. Smart Card, of the size of a credit card, has a small micro-controller and memory; and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society. Well, the list goes on. It is no exaggeration to say that eyes wherever you go, you can see, or at least feel, the work of an Embedded System.

**1.4 Organization of the Project:**

* In Chapter 2 the overall System Block Diagram of Measurement of Heart Rate using Wireless System is explained along with its description.
* In Chapter 3 the Microcontroller AT89S52 Pin diagram and all its description is explained.
* In Chapter 4 the description about components are explained.
* In Chapter 5 the Software is explained
* In Chapter 6 Source Code of the project is shown.
* In Chapter 7 Results are shown.
* In Chapter 8 Conclusion and Future Scope are shown.

**CHAPTER 2**

**BLOCK DIAGRAM AND EXPLANATION**

**2.1 Block Diagram**

The Fig:2.1 indicates the block diagram for Measurement of Heart rate using ZigBee.When the finger is placed on the sensor and it is sensed by the IR and the send the signal to the Microcontroller then it switches the relay into on and thus the heart beat get measured for the given time.

ZIGEE Transmitter

Microcontroller Device

Amplifier Circuit

Finger Tip Sensor



Microcontroller AT89S52

ZIGBEE

Receiver

LCD Display

**Fig 2.1: Block Diagram of Heart Monitoring System**

**2.2 Explanation:**

**2.2.1 Finger tip Sensor:**

The Sensor circuit consists of LED and Photo diode. The Finger tip is placed over the sensor assembly then IR LED transmits an Infrared Light into Finger tip, a part of reflected light is sensed by Photo Diode. The intensity of Light is depends on the blood volume inside the fingertip. The amplitude of Reflected signal is low so it can be amplified by Amplifier circuit.

LM35 is a integrated temperature sensor. It generates an analog voltage depending on the temperature of the patient’s body. The sensor output voltage is linearly proPortional to the body temperature. LM35 generates a higher output voltage than thermo couples. The sensor can measure temperature and generate signal that is sent to a Microcontroller. The data are then transmitted by the ZigBee to the PC. The sensors are connected to the I/O Port of the 89S52Microcontroller. The output voltage is converted into temperature by a simple conversion factor. As shown in Fig :2.1 the temperature sensor measures the temperature and converts it into electrical signal. The electrical signal is then processed by a Microcontroller and the Hyper Terminal software. Finally, it is displayed in the monitoring unit.

**2.2.2 Amplifier Circuit:**

The Reflected IR signal detected by photo diodeis fed to Amplifier Circuit.The Circiut Consists of Signal Conditioning Unit.This unit formed with filters and two stage operational amplifiers.With high gain amplifier ,the amplitude of reflected light can be converted tn to pulse.That filters remove the un wanted signals and boost the desired pulse signal.The cut off frequency of both filters are set to 2.5Hz,so it can measure.

**2.2.3 ZigBee Module:**

ZigBee Networks are called personal area networks (PAN). Each network contains a 16-bit identifier called a PAN ID and is represented in the Fig:2.2.



**Fig 2.2: ZigBee Module**

The upper layers, shown in Figure 2.2, consist of a network layer, which provides network configuration, manipulation, and message routing.Here, ZigBee for wireless transmission. The Doctor can get arecord of a particular patient‟s information by just accessing the database of the patient on his PC which is continuously updated through ZigBee receiver module.

ZigBee has lower power consumption. Hence, ZigBee is generallyused for 24 hours monitor of communication transmission Systems. Compared to Bluetooth, ZigBee provides higher network flexibility and a larger number of nodes, and a better transmission range with low power consumption.Large number of nodes enables the expansion of such Systems.

Temperature and Heart Beat will be continuously transmitted and monitored through wireless technology ZigBee . This System does not require the patient to be confined to his bed and allows him to move around freely in his room within a specific distance from the doctor’s monitor. Depending on the size of the hospital, several such nodes might be required resulting in a much higher System infrastructure cost. A ZigBee node is connected to every patient monitor System that consumes very low power and is extremely small in size.

**2.2.4 Liquid Crystal Display :**

Liquid Crystal Displays (LCD) offer several advantages over traditional Cathode-Ray Tube displays that make them ideal for several applications. Of course, LCD’s are flat and they use only a fraction of the power required by cathode-ray tubes. They are easier to read and more pleasant to work with for long periods of time than most ordinary video monitors. There are several tradeoffs as well, such as limited view angle, brightness, and contrast, not to mention high manufacturing cost.32x2 LCD is used in this project to display data to user. There are two rows and 32 columns. It is possible to display 32 characters on each of the 2 rows. It has two registers, command register and data register.

**Features of 16\*2 LCD Display:**

It displays the 224 different symbols

* Low power consumption
* TTL & CMOS Compatible

CHAPTER 3

**MICROCONTROLLER (AT89S52)**

**3.1 Introduction:**

Microcontrollers producers have been struggling for a long time for attracting more and more choosy customers. Every couple of days a new chip with a higher operating frequency, more memory and more high-quality A/D converters comes on the market.

Nevertheless, by analyzing their structure it is concluded that most of them have the same (or at least very similar) architecture known in the product catalogs as “8051 compatible”. What is all this about? The whole story began in the far 80s when Intel launched its series of the Microcontrollers labeled with MCS 051. Although, several circuits belonging to this series had quite modest features in comparison to the new ones, they took over the world very fast and became a standard for what nowadays is mint by a word Microcontroller.

The reason for success and such a big popularity is a skillfully chosen configuration which satisfies needs of a great number of the users allowing at the same time stable expanding (refers to the new types of the Microcontrollers). Besides, since a great deal of software has been developed in the meantime, it simply was not profitable to change anything in the Microcontroller’s basic core. That is the reason for having a great number of various Microcontrollers which actually are solely upgraded versions of the 8051 family. What is it what makes this Microcontroller so special and universal so that almost all the world producers manufacture it today under different name.

**3.2Description:**

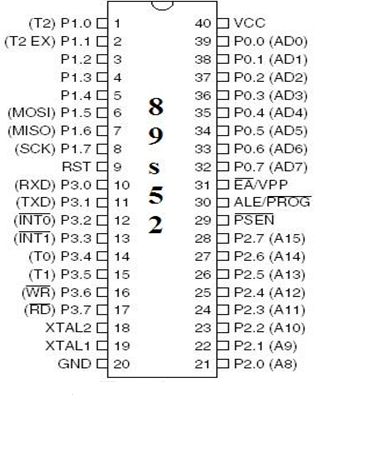
The AT89S52 is a low-power, high-performance CMOS 8-bit Microcontroller with 8K bytes of in-System programmable Flash memory. The device is manufactured using Atmel’s high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pin-out. The on-chip Flash allows the program memory to be reprogrammed in-System or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-System programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful Microcontroller which provides a highly-flexible and cost-effective solution to many Embedded control applications. The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial Port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial Port, and interrupt System to continue functioning. The Power-down mode saves the RAM con-tents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

**3.3 Features of AT89S52:**

* 8K Bytes of Re-programmable Flash Memory.
* RAM is 256 bytes.
* 2.7V to 6V Operating Range.
* Fully Static Operation: 0 Hz to 24 MHz.
* Two-level Program Memory Lock.
* 256x 8-bit Internal RAM.
* 32 Programmable I/O Lines.
* Two 16-bit Timer/Counters.
* Six Interrupt Sources.
* Programmable Serial UART Channel.
* Low-power Idle and Power-down Modes

As it is shown in the Fig: 3.1 the 8051 Microcontroller have nothing impressive at first sight.

The whole configuration is obviously envisaged as such to satisfy the needs of most programmers who work on development of automation devices. One of advantages of this Microcontroller is that nothing is missing and nothing is too much. In other words, it is created exactly in accordance to the average user‘s taste and needs. The other advantage is the way RAM is organized, the way Central Processor Unit (CPU) operates and Ports which maximally use all recourses and enable further upgrading.

  
 **Fig 3.1: Pin Diagram of AT89S52 Microcontroller**

**3.4 89S52 Microcontroller pins:**

**Pins 1-8:** Port 1each of these pins can be configured as input or output.

**Pin 9: RS** Logical one on this pin stops Microcontroller’s operating and erases the contents of most registers. By applying logical zero to this pin, the program starts execution from the beginning. In other words, a positive voltage pulse on this pin resets the Microcontroller.

**Pins 10-17:** Port 3 Similar to Port 1, each of these pins can serve as universal input or output. Besides, all of them have alternative functions

**Pin 10:** RXD Serial Asynchronous communication input or Serial Synchronous communication output.

**Pin 11:** TXD Serial asynchronous communication output or Serial synchronous communication clock output.

**Pin 12:** INT0 Interrupt 0 input

**Pin 13:** INT1Interrupt 1 input

**Pin 14:** T0Counter 0 clock input

**Pin 15:** T1Counter 1 clock input

**Pin 16:** WR Signal for writing to external (additional) RAM

**Pin 17:** RD Signal for reading from external RAM

**Pins 18, 19:** X2, X1 Internal oscillator input and output. A Quartz Crystal Oscillator which determines operating frequency is usually connected to these pins. Instead of quartz crystal, the miniature ceramics resonators can be also used for frequency stabilization. Later versions of the Microcontrollers operate at a frequency of 0 Hz up to over 50 Hz.

**Pin 20:** GND Ground

**Pin 21-28:** Port2 if there is no intention to use external memory then these Port pins are configured as universal inputs/outputs. In case external memory is used then the higher address byte, i.e. addresses A8-A15 will appear on this Port. It is important to know that even memory with capacity of 64Kb is not used ( i.e. note all bits on Port are used for memory addressing) the rest of bits are not available as inputs or outputs.

**Pin 29:** PSEN if external ROM is used for storing program then it has a logic-0 value every time the Microcontroller reads a byte from memory.

**Pin 30:** ALE Prior to each reading from external memory, the Microcontroller will set the lower address byte (A0-A7) on P0 and immediately after that activates the output ALE. Upon receiving signal from the ALE pin, the external register (74HCT373 or 74HCT375 circuit is usually Embedded) memorizes the state of P0 and uses it as an address for memory chip. In the second part of the Microcontroller’s machine cycle, a signal on this pin stops being emitted and P0 is used now for data transmission (Data Bus). In this way, by means of only one additional (and cheap) integrated circuit, data multiplexing from the Port is performed. This Port at the same time used for data and address transmission.

**Pin 31:** EA By applying logic zero to this pin, P2 and P3 are used for data and address transmission with no regard to whether there is internal memory or not. That means that even there is a program written to the Microcontroller, it will not be executed, the program written to external ROM will be used instead. Otherwise, by applying logic one to the EA pin, the Microcontroller will use both memories, first internal and afterwards external (if it exists), up to end of address space.

**Pins 32-39:** Port 0 Similar to Port 2, if external memory is not used, these pins can be used as universal inputs or outputs. Otherwise, P0 is configured as address output (A0-A7) when the ALE pin is at high level (1) and as data output (Data Bus), when logic zero (0) is applied to the ALE pin.

**Pin 40:** VCC Power supply.

**Input/ Output Ports (I/O Ports):**

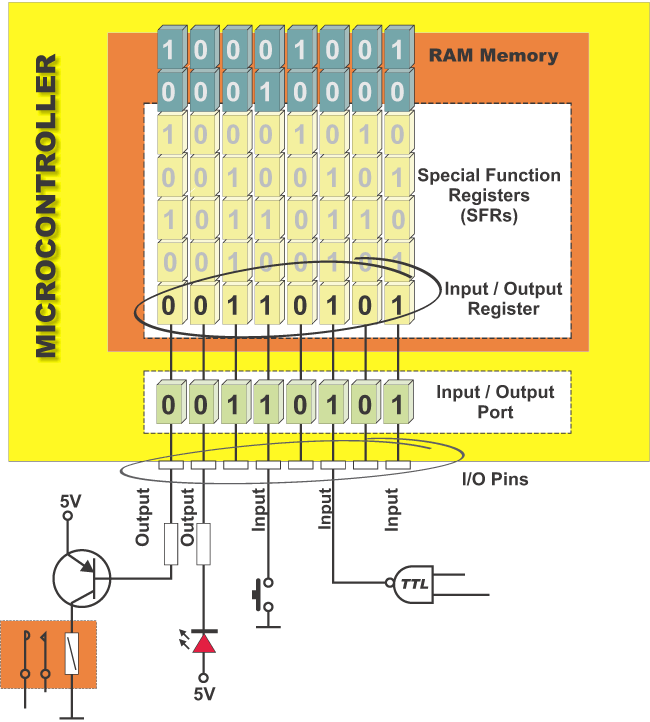
All 89S52 Microcontrollers have 4 I/O Ports, each consisting of 8 bits which can be configured as inputs or outputs. This means that the user has on disposal in total of 32 input/output lines connecting the Microcontroller to peripheral devices.

A logic state on a pin determines whether it is configured as input or output: 0=output, 1=input. If a pin on the Microcontroller needs to be configured as output, then logic zero (0) should be applied to the appropriate bit on I/O Port. In this way, a voltage level on the appropriate pin will be 0.

Similar to that, if a pin needs to be configured as input, then a logic one (1) should be applied to the appropriate Port. In this way, as a side effect a voltage level on the appropriate pin will be 5V (as it is case with any TTL input). This may sound a bit confusing but everything becomes clear after studying a simplified electronic circuit connected to one I/O pin.

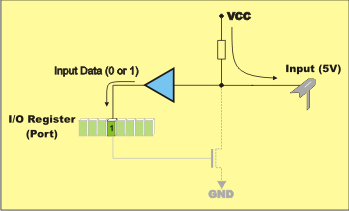
**Input/Output (I/O) Pins:**

This is a simplified overview of what is connected to a pin inside the Microcontroller is shown in the Fig: 3.2.

It concerns all pins except those included in P0 which do not have Embedded pull up transistor.

**Fig 3.2: Input/Output (I/O) Pins**

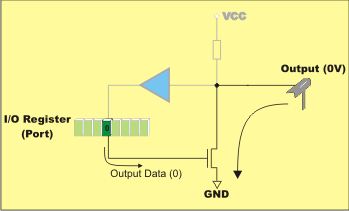
**Output Pin:**

The logic zero (0) is applied to a bit in the P register. By turning output FE transistor on, the appropriate pin is directly connected to ground and is shown in the Fig: 3.3

**Fig: 3.3 Output Pin Connections**

**Input Pin:**

The logic one (1) is applied to a bit in the P register. Output FE transistor is turned off. The appropriate pin remains connected to voltage power supply through a pull-up resistor of high resistance .A logic state (voltage) on any pin can be changed or read at any moment. A logic zero (0) and logic one (1) are not equal. The logic one (0) represents almost short circuit to ground. Such a pin is configured as output and is shown in Fig: 3.4



**Fig: 3.4 Input Pin Connections**

A logic one (1) is “loosely” connected to voltage power supply through resistors of high resistance. Since this voltage can be easily “pulled down” by an external signal, such a pin is configured as input.

**Port 0:**

It is specific to this Port to have a double purpose. If external memory is used then the lower address byte (addresses A0-A7) is applied on it. Otherwise, all bits on this Port are configured as inputs or outputs.

Another characteristic is expressed when it is configured as output. Namely, unlike other Ports consisting of pins with Embedded pull-up resistor (connected by its end to 5 V power supply), this resistor is left out here. This, apparently little change has its consequences:

If any pin on this Port is configured as input then it performs as if it “floats”. Such input has unlimited input resistance and has no voltage coming from “inside”. When the pin is configured as output, it performs as “open drain”, meaning that by writing 0 to some Port’s bit, the appropriate pin will be connected to ground (0V). By writing 1, the external output will keep on “floating”. In order to apply 1 (5V) on this output, an external pull-up resistor must be Embedded.

Only in case P0 is used for addressing external memory (only in that case), the Microcontroller will provide internal power supply source in order to establish logical ones on pins. There is no need to add external pull up resistors.

**Port 1:**

This is a true I/O Port, because there are no role assigning as it is the case with P0. Since it has Embedded pull-up resistors it is completely compatible with TTL circuits.

**Port 2:**

Similar to P0, when using external memory, lines on this Port occupy addresses intended for external memory chip. This time it is the higher address byte with addresses A8-A15. When there is no additional memory, this Port can be used as universal input-output Port similar by its features to the Port 1.

**Port 3:**

Even though all pins on this Port can be used as universal I/O Port, they also have an alternative function. Since each of these functions use inputs, then the appropriate pins have to be configured like that. In other words, prior to using some of reserve Port functions, a logical one (1) must be written to the appropriate bit in the P3 register. From hardware perspective, this Port is also similar to P0, with the difference that its outputs have a pull-up resistor Embedded.

**3.5 89S52 Memory Organization:**

The Microcontroller memory is divided into Program Memory and Data Memory. Program Memory (ROM) is used for permanent saving program being executed, while Data Memory (RAM) is used for temporarily storing and keeping intermediate results and variables. Depending on the model in use (still referring to the whole 8051 Microcontroller family) at most a few Kb of ROM and 128 or 256 bytes of RAM can be used.

All 89S52 Microcontrollers have 16-bit addressing bus and can address 64 kb memory. It is neither a mistake nor a big ambition of engineers who were working on basic core development. It is a matter of very clever memory organization which makes these controllers a real “programmers” tidbit.

**Program Memory:**

If the EA pin is connected to GND, all program fetches are directed to external memory. On the AT89S52, if EA is connected to VCC, program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

**Data Memory:**

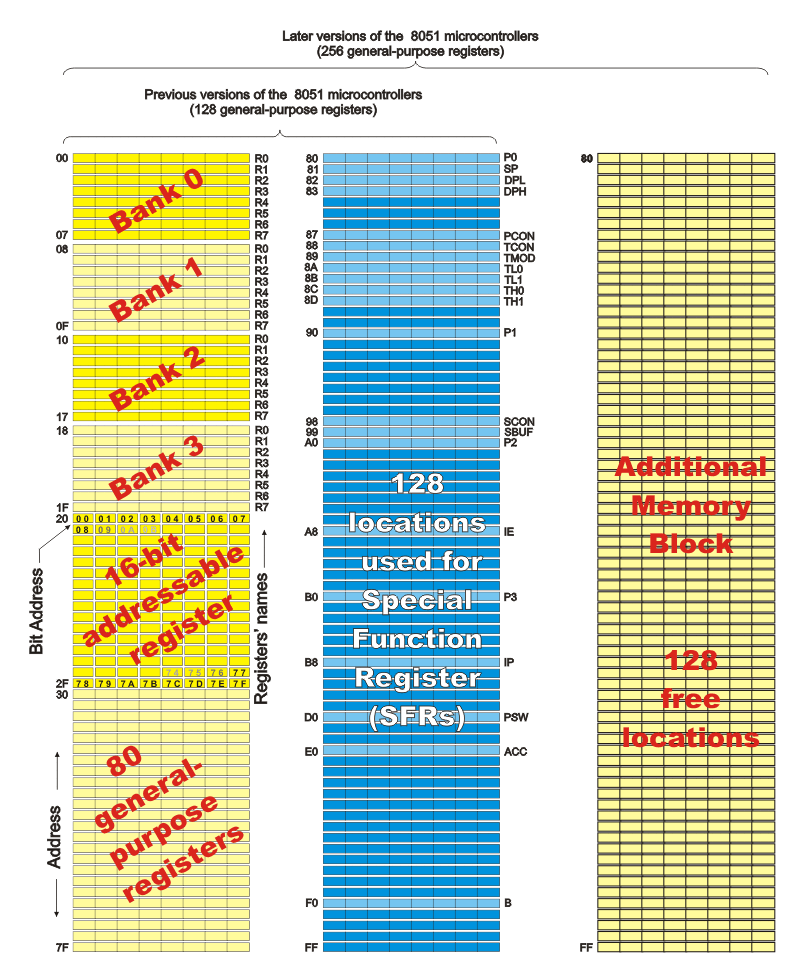
The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space. When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions which use direct addressing access the SFR space. For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2). MOV 0A0H , #data Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H). MOV @R0, #data Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

As already mentioned, Data Memory is used for temporarily storing and keeping data and intermediate results created and used during Microcontroller’s operating. Besides, this Microcontroller family includes many other registers such as: hardware counters and timers, input/output Ports, serial data buffers etc. The previous versions have the total memory size of 256 locations, while for later models this number is incremented by additional 128 available registers. In both cases, these first 256 memory locations (addresses 0-FFh) are the base of the memory. Common to all types of the 8051 Microcontrollers. Locations available to the user occupy memory space with addresses from 0 to 7Fh. First 128 registers and this part of RAM is divided in several blocks.

The first block consists of 4 banks each including 8 registers designated as R0 to R7. 4Prior to access them, a bank containing that register must be selected. Next memory block (in the range of 20h to 2Fh) is bit- addressable, which means that each bit being there has its own address from 0 to 7Fh. Since there are 16 such registers, this block contains in total of 128 bits with separate addresses (The 0th bit of the 20h byte has the bit address 0 and the 7th bit of the 2Fh byte has the bit address 7Fh). The third group of registers occupy addresses 2Fh-7Fh ( in total of 80 locations) and does not have any special purpose or feature.

**Additional Memory Block of Data Memory:**

In order to satisfy the programmers’ permanent hunger for Data Memory, producers have Embedded an additional memory block of 128 locations into the latest versions of the 89S52 Microcontrollers. Naturally, it’s not so simple. The problem is that electronics performing addressing has 1 byte (8 bits) on disposal and due to that it can reach only the first 256 locations. In order to keep already existing 8-bit architecture and compatibility with other existing models a little trick has been used. Using trick in this case means that additional memory block shares the same addresses with existing locations intended for the SFRs (80h- FFh) is shown in the Fig:3.5. In order to differentiate between these two physically separated memory spaces, different ways of addressing are used. A direct addressing is used for all locations in the SFRs, while the locations from additional RAM are accessible using indirect addressing.

****

**Fig 3.5: Register Banks**

**Addressing:**

While operating, processor processes data according to the program instructions. Each instruction consists of two parts. One part describes what should be done and another part indicates what to use to do it. This later part can be data (binary number) or address where the data is stored. All 89S52 Microcontrollers use two ways of addressing depending on which part of memory should be accessed.

* Direct Addressing
* InDirect Addressing

**Direct Addressing:**

On Direct Addressing, a value is obtained from a memory location while the address of that location is specified in instruction. Only after that, the instruction can process data (how depends on the type of instruction addition, subtraction, copy…). Obviously, a number being changed during operating a variable can reside at that specified address.

For example the address is only one byte in size ( the greatest number is 255), this is how only the first 255 locations in RAM can be accessed in this case the first half of the basic RAM is intended to be used freely, while another half is reserved for the SFRs.

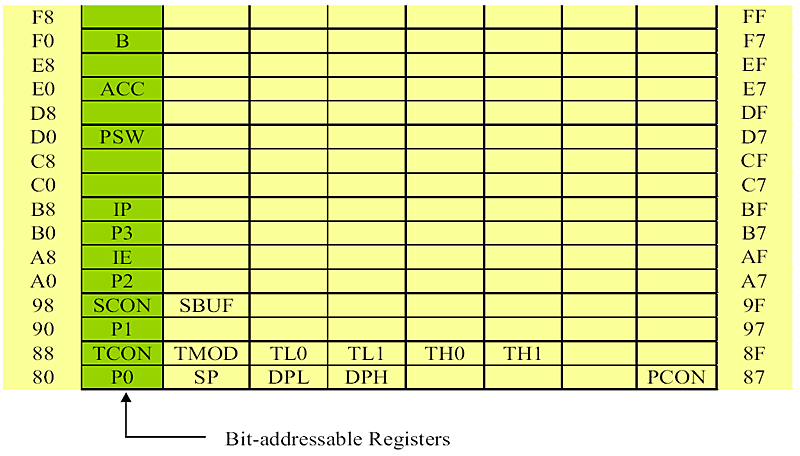
**Indirect Addressing:**

On Indirect Addressing, a register which contains address of another register is specified in the instruction. A value used in operating process resides in that another register. For example:

Only RAM locations available for use are accessed by indirect addressing (never in the SFRs). For all latest versions of the Microcontrollers with additional memory block (those 128 locations in Data Memory), this is the only way of accessing them. Simply, when during operating, the instruction including “@” sign is encountered and if the specified address is higher than 128 (7F hex.), the processor knows that indirect addressing is used and jumps over memory space reserved for the SFRs. On Indirect Addressing, the registers R0, R1 or Stack Pointer are used for specifying 8-bit addresses. Since only 8 bits are available, it is possible to access only registers of internal RAM in this way (128 locations in former or 256 locations in latest versions of the Microcontrollers). If memory extension in form of additional memory chip is used then the 16-bit DPTR Register (consisting of the registers DPTRL and DPTRH) is used for specifying addresses. In this way it is possible to access any location in the range of 64K.

**3.6 Special Function Registers (SFRs):**

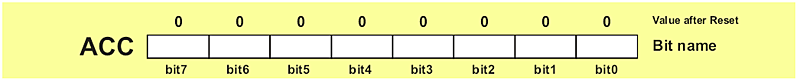
SFRs are a kind of control table used for running and monitoring Microcontroller’s operating is shown in the Fig: 3.6. Each of these registers, even each bit they include, has its name, address in the scope of RAM and clearly defined purpose ( for example: timer control, interrupt, serial connection etc.). Even though there are 128 free memory locations intended for their storage, the basic core, shared by all types of 8051 controllers, has only 21 such registers. Rest of locations is intensionally left free in order to enable the producers to further improved models keeping at the same time compatibility with the previous versions. It also enables the use of programs written a long time ago for the Microcontrollers which are out of production now.

****

**Fig 3.6: Special Function Registers**

**A Register (Accumulator):**

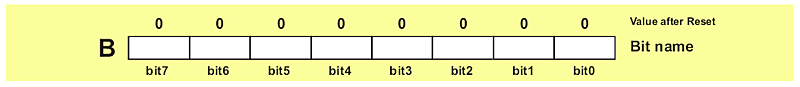
This is a general-purpose register which serves for storing intermediate results during operating and is shown in the Fig: 3.7. A number (an operand) should be added to the accumulator prior to execute an instruction upon it. Once an arithmetical operation is performed by the ALU, the result is placed into the accumulator. If a data should be transferred from one register to another, it must go through accumulator. For such universal purpose, this is the most commonly used register that none Microcontroller can be imagined without (more than a half 8051 Microcontroller's instructions used use the accumulator in some way).

****

**Fig 3.7: A Register**

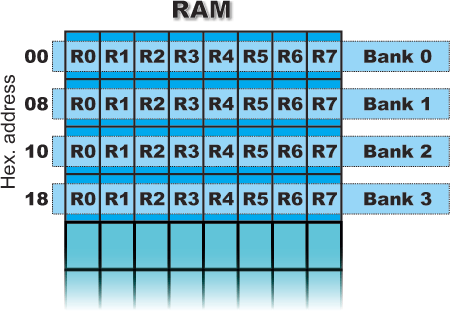
**B Register:**

B register is used during multiply and divide operations which can be performed only upon numbers stored in the A and B registers. All other instructions in the program can use this register as a spare accumulator (A).The B Register is represented in the Fig: 3.8.

During programming, each of registers is called by name so that their exact address is not so important for the user. During compiling into machine code (series of hexadecimal numbers recognized as instructions by the Microcontroller), PC will automatically, instead of registers name, write necessary addresses into the Microcontroller.

**Fig 3.8: B Register**

**R Registers (R0-R7):**

****

**Fig 3.9: R Register**

This is a common name for the total 8 general purpose registers (R0, R1, R2 ...R7) and is shown in the above Fig: 3.9. Even they are not true SFR’s, they deserve to be discussed here because of their purpose. The bank is active when the R registers it includes are in use. Similar to the accumulator, they are used for temporary storing variables and Intermediate results. Which of the banks will be active depends on two bits included in the PSW Register. These registers are stored in four banks in the scope of RAM.

**3.7 Quartz Crystal Oscillators:**

One of the most important features of any oscillator is its *frequency stability*, or in other words its ability to provide a constant frequency output under varying load conditions. Some of the factors that affect the frequency stability of an oscillator include: temperature, variations in the load and changes in the DC power supply.

Some of the factors that affect the frequency stability of an oscillator include: temperature, variations in the load and changes in the DC power supply. Frequency stability of the output signal can be improved by the proper selection of the components used for the resonant feedback circuit including the amplifier but there is a limit to the stability that can be obtained from normal LC and RC tank circuits. To obtain a very high level of oscillator stability a **Quartz Crystal** is generally used as the frequency determining device to produce another types of oscillator circuit known generally as a **Quartz Crystal Oscillator**, (XO) is shown in the Fig: 3.10.



**Fig 3.10: Quartz Crystal Oscillator**

When a voltage source is applied to a small thin piece of quartz crystal, it begins to change shape producing a characteristic known as the Piezo-electric effect. This piezo-electric effect is the property of a crystal by which an electrical charge produces a mechanical force by changing the shape of the crystal and vice versa, a mechanical force applied to the crystal produces an electrical charge.

Then, piezo-electric devices can be classed as [Transducers](http://www.electronics-tutorials.ws/io/io_1.html) as they convert energy of one kind into energy of another (electrical to mechanical or mechanical to electrical). This piezo-electric effect produces mechanical vibrations or oscillations which are used to replace the LC tank circuit in the previous oscillators. There are many different types of crystal substances which can be used as oscillators with the most important of these for electronic circuits being the quartz minerals because of their greater mechanical strength.

The crystals characteristic or resonant frequency is inversely proPortional to its physical thickness between the two metalized surfaces. A mechanically vibrating crystal can be represented by an equivalent electrical circuit consisting of low resistance, large inductance and small capacitance as shown in the above Fig :3.10.A quartz crystal has a resonant frequency similar to that of a electrically tuned tank circuit but with a much higher Q factor due to its low resistance, with typical frequencies ranging from 4kHz to 10MHz. The cut of the crystal also determines how it will behave as some crystals will vibrate at more than one frequency. Also, if the crystal is not of a parallel or uniform thickness it has two or more resonant frequencies having both a fundamental frequency and harmonics such as second or third harmonics. However, usually the fundamental frequency is more stronger or pronounced than the others and this is the one used. The equivalent circuit above has three reactive components and there are two resonant frequencies, the lowest is a series type frequency and the highest a parallel type resonant frequency.

An amplifier circuit will oscillate if it has a loop gain greater or equal to one and the feedback is positive. In a **Quartz Crystal Oscillator** circuit the oscillator will oscillate at the crystals fundamental parallel resonant frequency as the crystal always wants to oscillate when a voltage source is applied to it. However, it is also possible to "tune" a crystal oscillator to any even harmonic of the fundamental frequency, (2nd, 4th, 8th etc.) and these are known generally as Harmonic Oscillators while Overtone Oscillators vibrate at odd multiples of the fundamental frequency, 3rd, 5th, 11th etc). Generally, crystal oscillators that operate at overtone frequencies do so using their series resonant frequency.

**3.8 Schematic Diagram**:

5v  Sensor input

**Fig 3.11: Schematic Representation of the Project**

**Pin Connections:**

In this when the Finger is placed in the Sensor which takes the signals from the body and sends to the Amplifier circuit. When the interruption occurs in the sensor mechanism then command signal will be sent to the microcontroller.

Microcontroller works only when the command receives from the Amplifier circuit. LCD will shows the timer in counting the Heart Beat for 20 seconds of time.

Also the beats for every 20 seconds the message is sent to the Doctor room and is displayed in the Monitor of the Doctor’s room by using the ZigBee receiver.The information is collected by the ZigBee transmitter and is collected by the ZigBee receiver.The range of ZigBee is 100 meters.

Microcontroller has 40 Pins and 4 Ports. They are Port 0,Port 1,Port 2,Port 3.In this Microcontroller is used to count the heart beat for every 20 sec which will be displayed on LCD.

Basically, LCD has 16 pins. Pins of 4,5,6 are named as Enable, RD/WR,RS.

These are given to the Port 2 i.e,2.5, 2.6, 2.7 pins.Pin 7 to Pin 14 are data pins. These are given to Port 0 of Microcontroller. Pin 3 of LCD is Contrast pin which is connected with variable resistor to set the brightness of the display.Pin 1 and Pin 2 are shorted with LED+ and LED- of Pin 15 and pin 16.

The supply given to the Microcontroller by voltage regulator(IC7805).Pin 3 of IC7805 is given to Pin 40,pins of IC7805 (pin 1,pin2) are given to battery supply.

Microcontroller (Pin 18,19) are XTAL1,XTAL2 are the heart of the Microcontroller which are used to control the frequency by crystal Oscillator.This crystal oscillator connected with disc capacitors for accurate clock frequency.The 9th pin of Microcontroller is Reset,which is formed with capacitor and resistor to clear the previous data.

**CHAPTER 4**

**COMPONENTS DESCRIPTION**

**4.1 Voltage Regulator:**

**4.1.1 LM 78XX Series Voltage Regulator:**

The LM 78XX series of the three terminal regulations is available with several fixed output voltages making them useful in a wide range of applications is shown in the Fig4.1. One of these is local on card regulation. The voltages available allow these regulators to be used in logic Systems, instrumentation and other solid state electronic equipment. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents. The LM78XX series is available in aluminium to 3 packages which will allow over 1.5A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. The LM 78XX is available in the metal 3 leads to 5 and the plastic to 92. For this type with adequate heat sinking. The regulator can deliver 100mA output current.

The advantage of this type of regulator, it is easy to use and minimize the number of external components.

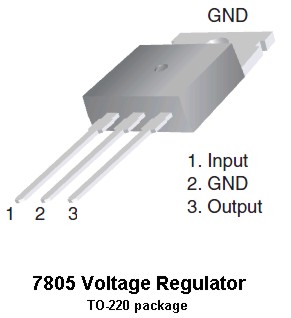
The following are the features voltage regulators:

* Output current in excess of 1.5A for 78 and 78L series.
* Internal thermal overload protection.
* No external components required.
* Output transistor sage area protection.
* Internal short circuit current limit.
* Available in aluminium 3 package.

**4.1.2 Positive Voltage Regulator:**

The positive voltage regulator has different features like

* Output current up to 1.5A.
* No external components.
* Internal thermal overload protection.
* High power dissipation capability.
* Internal short-circuit current limiting.
* Output transistor safe area compensation.
* Direct replacements for Fairchild microA7800 series.



**Fig 4.1: Pin Representation of Voltage Regulator**

**4.1.3 Features:**

* Output Current up to 1A.
* Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V.
* Thermal Overload Protection.
* Short Circuit Protection.
* Output Transistor Safe Operating Area Protection**.**

**4.2 BATTERIES:**

An electric battery is a device consisting of one or more [electrochemical cells](http://en.wikipedia.org/wiki/Electrochemical_cell) that convert stored chemical [energy](http://en.wikipedia.org/wiki/Energy) into electrical energy. Each cell contains a positive terminal, or [cathode](http://en.wikipedia.org/wiki/Cathode), and a negative terminal, or [anode](http://en.wikipedia.org/wiki/Anode). [Electrolytes](http://en.wikipedia.org/wiki/Electrolyte) allow ions to move between the electrodes and terminals, which allows current to flow out of the battery to perform work.

**4.2.1 Introduction to SMF Battery:**

SMF batteries measure created in an eco-friendly, ISO Certified & trendy plant with a huge producing capability and square measure being sold-out worldwide. There are differences between SMF batteries and other tubular batteries. In SMF Batteries no distilled water or effort is needed. There are a wide selection of SMF battery on the market to suit all applications of standby power needs like UPS, electrical converter and Emergency Lights, communication System, hearth Alarm & Security Systems, Railway communication, Electronic group action and money Registers, star Lanterns and Systems, etc. The SMF batteries are available industrial plant charged conditions and have a high period thereby requiring longer time intervals between recharging of batteries available.

**4.2.2 Categories and Types of Batteries:**

Batteries are classified in to two types

* Primary Battery
* Secondary Battery

Primary batteries irreversibly transform chemical energy to electrical energy. When the supply of reactants is exhausted, energy cannot be readily restored to the battery.

Secondary batteries can be recharged; that is, they can have their chemical reactions reversed by supplying electrical energy to the cell, approximately restoring their original composition.

Some types of primary batteries used, for example, for [telegraph](http://en.wikipedia.org/wiki/Telegraphy) circuits, were restored to operation by replacing the electrodes. Secondary batteries are not indefinitely rechargable due to dissipation of the active materials, loss of electrolyte and internal corrosion.

**Primary Batteries:**

Primary batteries or [primary cells](http://en.wikipedia.org/wiki/Primary_cell), can produce current immediately on assembly. These are most commonly used in Portable devices that have low current drain, are used only intermittently, or are used well away from an alternative power source, such as in alarm and communication circuits where other electric power is only intermittently available. Disposable primary cells cannot be reliably recharged, since the chemical reactions are not easily reversible and active materials may not return to their original forms. Battery manufacturers recommend against attempting recharging primary cells.

In general, these have higher [energy densities](http://en.wikipedia.org/wiki/Energy_densities) than rechargeable batteries, but disposable batteries do not fare well under high-drain applications with loads under 75 [ohms](http://en.wikipedia.org/wiki/Ohm) (75Ω).Common types of disposable batteries include [zinc–carbon batteries](http://en.wikipedia.org/wiki/Zinc%E2%80%93carbon_batteries) and [alkaline batteries](http://en.wikipedia.org/wiki/Alkaline_batteries).

**Secondary Batteries:**

Secondary batteries, also known as [secondary cells](http://en.wikipedia.org/wiki/Secondary_cell), or [rechargeable batteries](http://en.wikipedia.org/wiki/Rechargeable_batteries), must be charged before first use; they are usually assembled with active materials in the discharged state. Rechargeable batteries are (re)charged by applying electric current, which reverses the [chemical reactions](http://en.wikipedia.org/wiki/Chemical_reaction) that occur during discharge/use. Devices to supply the appropriate current are called chargers.

Other Portable rechargeable batteries include several sealed "dry cell" that are useful in applications such as [mobile phones](http://en.wikipedia.org/wiki/Mobile_phone) and [laptop computers](http://en.wikipedia.org/wiki/Laptop). Cells of this type include [Nickel–Cadmium](http://en.wikipedia.org/wiki/Nickel%E2%80%93cadmium_battery)(NiCad),[Nickel–Zinc](http://en.wikipedia.org/wiki/Nickel%E2%80%93zinc_battery)(NiZn),[Nickel Metal Hydride](http://en.wikipedia.org/wiki/Nickel_metal_hydride_battery) (NiMH), and [Lithium-Ion](http://en.wikipedia.org/wiki/Lithium-ion_battery) (Li-ion) cells. Li-ion has by far the highest share of the dry cell rechargeable market. NiMH has replaced NiCad in most applications due to its higher capacity, but NiCad remains in use in [power tools](http://en.wikipedia.org/wiki/Power_tool), [two-way radios](http://en.wikipedia.org/wiki/Two-way_radio), and [medical equipment](http://en.wikipedia.org/wiki/Medical_equipment).

**4.2.3 Features of SMF:**

SMF is SEALED MAINTENANCE FREE type battery.

* **Sealed/Maintenance-Free:**

The valve regulated, spill-proof construction of the Power-Sonic battery allows trouble-free, safe operation in any position. There is no need to add electrolyte, as gases generated during over-charge are recombined in a unique “oxygen cycle.”

* **Design Flexibility:**

These batteries may be used in both series & parallel mode to obtain the choice of voltage & capacity. Due to recent design breakthroughs, some of these batteries can be used in either cyclic or standby mode.

* **Deep discharge recovery:**

Special separators, advanced plate composition, and a carefully balanced electrolyte System have greatly improved the ability of recovering from excessively deep discharge.

* **Compact:**

Power-Sonic batteries use state of the art design, high grade materials, and a carefully controlled plate-making process to provide excellent output per cell. The high energy density results in superior power/volume and power/weight ratios.

* **High Discharge Rate:**

Low internal resistance allows discharge currents of up to ten times the rated capacity of the battery. Relatively small batteries may thus be specified in applications requiring high peak currents.

* **High Discharge Rate:**

Low internal resistance allows discharge currents of up to ten times the rated capacity of the battery. Relatively small batteries may thus be specified in applications requiring high peak currents.

* **Rugged Construction:**

The high impact resistant battery case is made either of non-conductive ABS plastic or styrene. Large capacity batteries frequently have polypropylene cases. All of these case materials impart great resistance to shock, vibration, chemicals and heat.

**4.2.4 Construction:**

**Electrolyte:**

Immobilized dilute sulphuric acid: - H2SO4.

**Container:**

The case material is ABS, a high-impact proof plastic resin, styrene, or a polypropylene-polyethene co-polymer with resistance to chemicals & flammability.

**Leak proof Design & Operational Safety:**

Power-Sonic batteries have been approved for shipment by air, both by D.O.T. and I.A.T.A. U.L.’s component recognition program for emergency lighting and power batteries lists Power-Sonic under file numbers MH14328 and MH14838.

**Terminals:**

Depending on the design, batteries come-up either with AMP fast on type terminals made with tin-platted brass, post type terminals of same composition with threaded nut & bolt hardware, or heavy-duty flag terminals made of lead-alloy. A special alloy is used as sealing material surrounding the terminals.

**Plate construction:**

Plate-construction is the key to produce a good battery. So, a latest technology is implemented to cast the grids from lead-cadmium alloy free of antimony. The small amounts of calcium & tin imparts strength to the walls of the electrode plates. Lead oxide paste is added to the grid to form an electrically active material. In the charged state, negative plate paste is pure lead that of positive lead.

**Theory of Operation:**

**The basic electrochemical reaction equation in a lead-acid battery can be written as follows:**

Pb + 2H2SO4 + PbO2 PbSO4 + 2H2O + PbSO4

Electrolyte

Active Material on Positive Plate

Active material on Negative plate

**4.2.5 Battery Cell Performance:**

**Capacity and Discharge:**

A battery's capacity is the amount of [electric charge](http://en.wikipedia.org/wiki/Electric_charge) it can deliver at the rated voltage. The more electrode material contained in the cell the greater its capacity. A small cell has less capacity than a larger cell with the same chemistry, although they develop the same open-circuit voltage. Capacity is measured in units such as [amp-hour](http://en.wikipedia.org/wiki/Amp-hour) (A·h).The rated capacity of a battery is usually expressed as the product of 20 hours multiplied by the current that a new battery can consistently supply for 20 hours at 68 °F (20 °C), while remaining above a specified terminal voltage per cell. For example, a battery rated at 100 A·h can deliver 5 A over a 20-hour period at [room temperature](http://en.wikipedia.org/wiki/Room_temperature). The fraction of the stored charge that a battery can deliver depends on multiple factors, including battery chemistry, the rate at which the charge is delivered (current), the required terminal voltage, the storage period, ambient temperature and other factors. Batteries that are stored for a long period or that are discharged at a small fraction of the capacity lose capacity due to the presence of generally irreversible side reactions that consume charge carriers without producing current. This phenomenon is known as internal self-discharge. Further, when batteries are recharged, additional side reactions can occur, reducing capacity for subsequent discharges. After enough recharges, in essence all capacity is lost and the battery stops producing power.

**Battery Lifetime:**

Available capacity of all batteries drops with decreasing temperature. In contrast to most of today's batteries, the [Zamboni pile](http://en.wikipedia.org/wiki/Zamboni_pile" \o "Zamboni pile), invented in 1812, offers a very long service life without refurbishment or recharge, although it supplies current only in the nanoamp range. The [Oxford Electric Bell](http://en.wikipedia.org/wiki/Oxford_Electric_Bell) has been ringing almost continuously since 1840 on its original pair of batteries, thought to be Zamboni piles.

**Self-discharge:**

Disposable batteries typically lose 8 to 20 percent of their original charge per year when stored at room temperature (20°–30°C). This is known as the "self-discharge" rate, and is due to non-current-producing "side" chemical reactions that occur within the cell even when no load is applied. The rate of side reactions is reduced for batteries are stored at lower temperatures, although some can be damaged by freezing.

Old rechargeable batteries self-discharge more rapidly than disposable alkaline batteries, especially [nickel](http://en.wikipedia.org/wiki/Nickel)-based batteries; a freshly charged Nickel Cadmium (NiCd) battery loses 10% of its charge in the first 24 hours, and thereafter discharges at a rate of about 10% a month. However, newer [low self-discharge Nickel Metal Hydride (NiMH) batteries](http://en.wikipedia.org/wiki/Low_self-discharge_NiMH_battery) and modern Lithium designs display a lower self-discharge rate (but still higher than for primary batteries).

**Physical component changes:**

The active material on the battery plates changes chemical composition on each charge and discharge cycle, active material may be lost due to physical changes of volume; further limiting the number of times the battery can be recharged.

Most nickel-based batteries are partially discharged when purchased, and must be charged before first use. Newer NiMH batteries are ready to be used when purchased, and have only 15% discharge in a year.

**Storage:**

Battery life can be extended by storing the batteries at a low temperature, as in a [refrigerator](http://en.wikipedia.org/wiki/Refrigerator) or [freezer](http://en.wikipedia.org/wiki/Freezer), which slows the side reactions. Such storage can extend the life of alkaline batteries by about 5%; rechargeable batteries can hold their charge much longer, depending upon type. To reach their maximum voltage, batteries must be returned to room temperature; discharging an alkaline battery at 250 mA at 0°C is only half as efficient as at 20°C. Alkaline battery manufacturers such as [Duracell](http://en.wikipedia.org/wiki/Duracell) do not recommend refrigerating batteries.

**4.2.6 Hazards:**

**Explosion:**

A battery explosion is caused by misuse or malfunction, such as attempting to recharge a primary (non-rechargeable) battery, or a [short circuit](http://en.wikipedia.org/wiki/Short_circuit). Car batteries are most likely to explode when a short-circuit generates very large currents. Car batteries produce [hydrogen](http://en.wikipedia.org/wiki/Hydrogen), which is very explosive, when they are overcharged (because of [electrolysis](http://en.wikipedia.org/wiki/Electrolysis) of the water in the electrolyte). The amount of overcharging is usually very small and generates little hydrogen, which dissipates quickly. However, when "jumping" a car battery, the high current can cause the rapid release of large volumes of hydrogen, which can be ignited explosively by a nearby spark, for example, when disconnecting a [jumper cable](http://en.wikipedia.org/wiki/Jumper_cable).

When a battery is recharged at an excessive rate, an explosive gas mixture of hydrogen and oxygen may be produced faster than it can escape from within the battery, leading to pressure build-up and eventual bursting of the battery case. In extreme cases, battery acid may spray violently from the casing and cause injury. Overcharging—that is, attempting to charge a battery beyond its electrical capacity—can also lead to a battery explosion, in addition to leakage or irreversible damage.

**Leakage:**

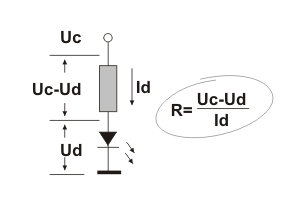
Many battery chemicals are corrosive, poisonous or both. If leakage occurs, either spontaneously or through accident, the chemicals released may be dangerous. For example, disposable batteries often use a zinc "can" both as a reactant and as the container to hold.

**4.3 IR LED & PHOTODIODES:**

**4.3.1 IR LED:**

Here the 5v is applied to IR LED via 330 ohms resistor in forward bias, the IR LED emits constant IR light in the free space. The light is focused on the photodiode kept at opposite to the IR LED, connected in reverse bias in series with 10 k ohms resistor. The junction is connected to the Port P1.0 of Microcontroller. When the IR light is focused on the photodiode, which maintains very low resistance due to that the P1.0 maintains low i.e. 0 V. When any obstruction is there in between IR LED and photodiode, there is no IR rays focused on the photodiode that maintains very high resistance due to that the Port maintains high signal i.e. 5V (bit-1).Like this we are connected 8 sensors for Port 1 (P1.0 to P1.7). These 8 sensors are organized in 4 rows of each 2 sensors.

Light-Emitting Diodes are elements for light signalization in electronics. They are manufactured in different shapes, colors and sizes. For their low price, low consumption and simple use, they have almost completely pushed aside other light sources- bulbs at first place. They perform similar to common diodes with the difference that they emit light when current flows through them and is shown in the Fig: 4.2.



**Fig 4.2: Light Emitting Diode**

It is important to know that each diode will be immediately destroyed unless its current is limited. This means that a conductor must be connected in parallel to a diode. In order to correctly determine value of this conductor, it is necessary to know diode’s voltage drop in forward direction, which depends on what material a diode is made of and what color it is. Values typical for the most frequently used diodes are shown in table below: As seen, there are three main types of LEDs. Standard ones get full brightness at current of 20mA. Low Current diodes get full brightness at ten time’s lower current while Super Bright diodes produce more intensive light than Standard ones. Since the 8051 Microcontrollers can provide only low input current and since their pins are configured as outputs when voltage level on them is equal to 0, direct connecting to LED’s is carried out as it is shown on figure (Low current LED, cathode is connected to output pin).

**4.3.2 Photo Diode:**

A photodiode is a type of photo detector capable of converting light into either current or voltage, depending upon the mode of operation. Photodiodes are similar to regular semiconductor diodes except that they may be either exposed (to detect vacuum UV or X-rays) or packaged with a window or optical fibre connection to allow light to reach the sensitive part of the device. Many diodes designed for use specifically as a photodiode will also use a PIN junction rather than the typical PN junction.

## **4.3.2.1 Principle of Operation:**

A photodiode is a PN junction or PIN structure. When a photon of sufficient energy strikes the diode, it excites an electron, thereby creating a mobile electron and a positively charged electron hole. If the absorption occurs in the junction's depletion region, or one diffusion length away from it, these carriers are swept from the junction by the built-in field of the depletion region. Thus holes move toward the anode, and electrons toward the cathode, and a photocurrent is produced.

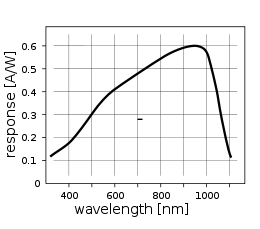
### **4.3.2.2 Photovoltaic Mode:**

When used in zero bias or photovoltaic mode, the flow of photocurrent out of the device is restricted and a voltage builds up. The diode becomes forward biased and "dark current" begins to flow across the junction in the direction opposite to the photocurrent. This mode is responsible for the photovoltaic effect, which is the basis for solar cells in fact, a solar cell is just a large area photodiode.

### **4.3.2.3 Photoconductive mode:**

In this mode the diode is often reverse biased, dramatically reducing the response time at the expense of increased noise. This increases the width of the depletion layer, which decreases the junction's capacitance resulting in faster response times. The reverse bias induces only a small amount of current (known as saturation or back current) along its direction while the photocurrent remains virtually the same. The photocurrent is linearly proportional to the luminance

Although this mode is faster, the photoconductive mode tends to exhibit more electronic noise. The leakage current of a good PIN diode is so low (< 1nA) that the Johnson–Nyquist noise of the load resistance in a typical circuit often dominates. The response is shown in the Fig: 4.3.

[](http://en.wikipedia.org/wiki/File:Response_silicon_photodiode.svg)

**Fig 4.3: Response of A Silicon Photo Diode Vs. Wavelength of The Incident light**

**4.4 PCB DESIGN:**

Design and Fabrication process of printed circuit boards.

**4.4.1 Introduction:**

Printed Circuit Boards, or PCB’s, from the core of electronic equipment in domestic and industrial fields. Some of the areas where PCB’s are intensively used are computers, process control, telecommunication and instrumentation.

**4.4.2 Manufacturing:**

The manufacturing process consists of two methods;1) Print and etch. 2) Print, plate and etch. The single sided PCB’s are usually made using the print and etch method. The double sided Plate Through-Hole (PTH) boards are made by the plate and method.

The production of multi layer boards uses both the methods. The inner layers are printed and etch while the outer layers are produced by print, plate and etch after pressing the inner layers.

**Software:**

The software used in our project to obtain the schematic layout is MICROSIM.

**Panelisation:**

Here the schematic is transformed in to the working positive/negative films. The circuit is repeated conveniently to accommodate economically as many circuits as possible in a panel, which can be operated in every sequence of subsequent steps in the PCB process. This is called panelisation. For the PCB boards the next operation is drilling.

**Drilling:**

PCB drilling is the state of the art operation. Very small holes are drilled with high speed. CNC drilling machines, giving a wall finish with less or no smear or epoxy, required for void free through hole plating.

**Plating:**

It is the heart of the PCB manufacturing process. The holes drilled in the board are treated both mechanically before depositing the copper by the electro less copper plating process.

**Etching:**

Once a multi layer board is drilled and electro less copper deposited, the image available in the form of a film is transferred on to the outside by photo printing using a dry film printing process. The boards are then electrolytic plated on to the circuit pattern with copper and tin. The tin plated deposit serves an etch resist when copper in the unwanted area is removed by the conveyor’s spray etching machines with chemical etch ants. The etching machines are attached to automatic doping equipment, which analysis and controls etchants concentrations.

**Solder mask:**

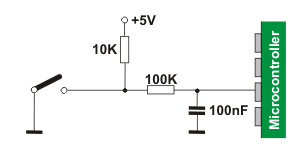
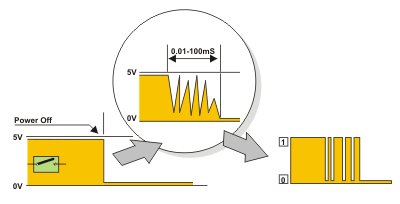
Since a PCB design may call for very close spacing between conductors, a solar mask has to be applied on the both sides of the circuitry to avoid the bridging of conductors. The solder mask ink is applied by screening. The ink is dried, exposed to UV, developed in a mild alkaline solution and finally cured by both UV and thermal.

**Hot Air Levelling:**

After apply the solder mask, the circuit pads are soldered using the hot air leveling process. The bare bodies fluxed and dipped in to a molten solder bath. While removing the board from the solder bath, hot air is blown on both sides of the boards through air knives in the machines, leaving the board soldered and leveled. This is one of common finishes given to the boards. Thus the double side plated through whole printed circuit board is manufactured and is now ready for the components to be soldered.

**4.5 PUSH BUTTONS:**

Nevertheless, it is not so simple in practice.This is about something commonly unnoticeable when using these components in everyday life. It is about contact bounce- a common problem with mechanical switches. If contact switching does not happen so quickly, several consecutive bounces can be noticed prior to maintain stable state. The reasons for this are: vibrations, slight rough spots and dirt. Anyway, whole this process does not last long (a few micro- or milliseconds), but long enough to be registered by the Microcontroller. Concerning pulse counter, error occurs and is shown in the Fig: 4.4.



**Fig 4.4: Switch & Interfacing of Switch**

The simplest solution is to connect simple RC circuit which will “suppress” each quick voltage change. Since the bouncing time is not defined, the values of elements are not strictly determined. In the most cases, the values shown on figure are sufficient.If complete safety is needed, radical measures should be taken The circuit, shown on the figure (RS flip-flop), changes logic state on its output with the first pulse triggered by contact bounce. Even though this is more expensive solution (SPDT switch), the problem is definitely resolved! Besides, since the condenser is not used, very short pulses can be also registered in this way. In addition to these hardware solutions, a simple software solution is commonly applied too: when a program tests the state of some input pin and finds changes, the check should be done one more time after certain time delay. If the change is confirmed it means that switch (or pushbutton) has changed its position. The advantages of such solution are obvious: it is free of charge, effects of disturbances are eliminated too and it can be adjusted to the worst-quality contacts.

**4.6 LIQUID CRYSTAL DISPLAY:**

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

* The declining prices of LCDs.
* The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.
* Incorporation of a refreshing controller into the LCD, thereby 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.

These components are “specialized” for being used with the Microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD. The Schematic Representation of LCD is given in the Fig:4.5.

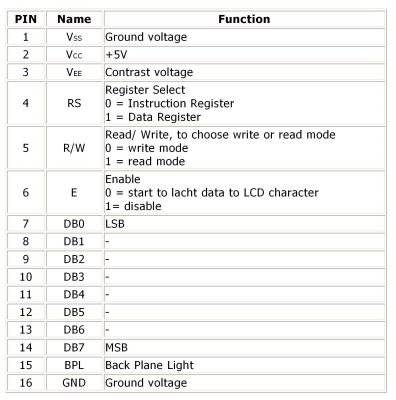
**Fig 4.5: Liquid Crystal Display**

A model described here is for its low price and great possibilities most frequently used in practice. It is based on the HD44780 Microcontroller (Hitachi) and can display messages in two lines with 16 characters each. It displays all the alphabets, Greek letters, punctuation marks, mathematical symbols etc. In addition, it is possible to display symbols that user makes up on its own. Automatic shifting message on display (shift left and right), appearance of the pointer, backlight etc. are considered as useful characteristics.

**Pins Functions:**

There are pins along one side of the small printed board used for connection to the Microcontroller. There are total of 14 pins marked with numbers (16 in case the background light is built in). Their function is described in the table below:

**Table 4.1: Pin Functions of LCD**



**LCD screen:**

LCD screen consists of two lines with 16 characters each. Each character consists of 5 x 7 dot matrix. Contrast on display depends on the power supply voltage and whether messages are displayed in one or two lines. For that reason, variable voltage 0-VDD is applied on pin marked as VEE. Trimmer potentiometer is usually used for that purpose. Some versions of displays have built in backlight (blue or green diodes). When used during operating, a resistor for current limitation should be used (like with any LED).The Schematic Representation is given below in the Fig: 4.6

**Fig 4.6: Liquid Crystal Display Power supply**

**LCD Basic Commands:**

All data transferred to LCD through outputs D0-D7 will be interpreted as commands or as data, which depends on logic state on pin RS:

* RS = 1 - Bits D0 - D7 are addresses of characters that should be displayed. Built in processor addresses built in “map of characters” and displays corresponding symbols. Displaying position is determined by DDRAM address. This address is either previously defined or the address of previously transferred character is automatically incremented.
* RS = 0 - Bits D0 - D7 are commands which determine display mode. List of commands which LCD recognizes are given in the table below:

**LCD Connection:**

Depending on how many lines are used for connection to the Microcontroller, there are 8-bit and 4-bit LCD modes. The appropriate mode is determined at the beginning of the process in a phase called “initialization”. In the first case, the data are transferred through outputs D0-D7 as it has been already explained. In case of 4-bit LED mode, for the sake of saving valuable I/O pins of the Microcontroller, there are only 4 higher bits (D4-D7) used for communication, while other may be left unconnected. Consequently, each data is sent to LCD in two steps: four higher bits are sent first (that normally would be sent through lines D4-D7), four lower bits are sent afterwards. With the help of initialization, LCD will correctly connect and interpret each data received. Besides, with regards to the fact that data are rarely read from LCD (data mainly are transferred from Microcontroller to LCD) one more I/O pin may be saved by simple connecting R/W pin to the Ground. Such saving has its price. Even though message displaying will be normally performed, it will not be possible to read from busy flag since it is not possible to read from display.

**LCD Initialization:**

Once the power supply is turned on, LCD is automatically cleared. This process lasts for approximately 15mS. After that, display is ready to operate. The mode of operating is set by default. This means that:

* Display is cleared
* Mode

DL = 1 Communication through 8-bit interface

N = 0 Messages are displayed in one line

F = 0 Character font 5 x 8 dots

Display/Cursor on/off

D = 0 Display off

U = 0 Cursor off

B = 0 Cursor blink off

* Character entry

ID = 1 Addresses on display are automatically incremented by 1

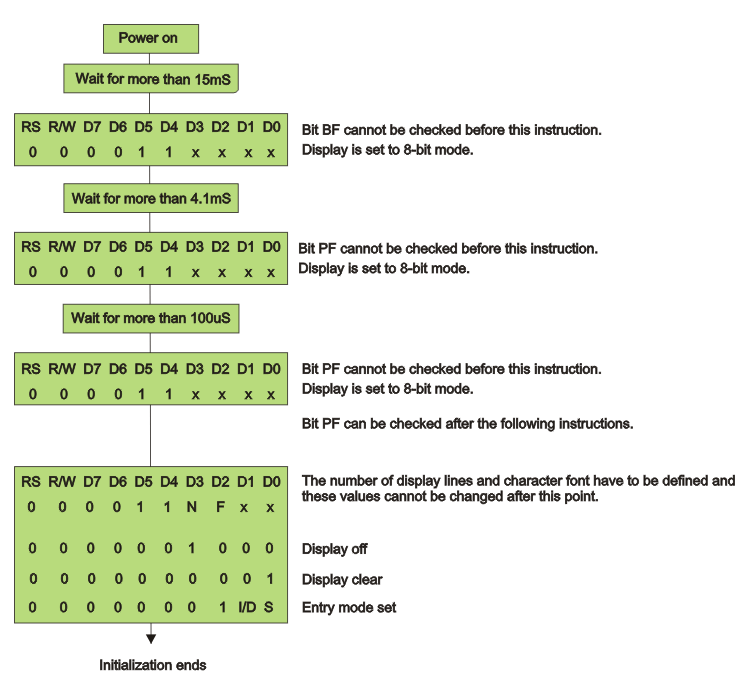
S = 0 Display shift off

Automatic reset is mainly performed without any problems. Mainly but not always! If for any reason power supply voltage does not reach full value in the course of 10mS, display will start perform completely unpredictable. If voltage supply unit cannot meet this condition or if it is needed to provide completely safe operating, the process of initialization by which a new reset enabling display to operate normally must be applied.

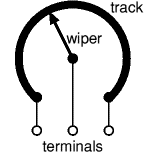
Algorithm according to the initialization is being performed depends on whether connection to the Microcontroller is through 4- or 8-bit interface. All left over to be done after that is to give basic commands and of course- to display messages. The Representation is given in below the Fig: 4.7

**Contrast Control:**

To have a clear view of the characters on the LCD, contrast should be adjusted. To adjust the contrast, the voltage should be varied. For this, a preset is used which can behave like a variable voltage device. As the voltage of this preset is varied, the contrast of the LCD can be adjusted.

**Fig 4.7: Procedure On 8-Bit Initialization**.

**Potentiometer:**

Variable resistors used as potentiometers have all three terminals connected. This arrangement is normally used to vary voltage, for example to set the switching point of a circuit with a sensor, or control the volume (loudness) in an amplifier circuit. If the terminals at the ends of the track are connected across the power supply, then the wiper terminal will provide a voltage which can be varied from zero up to the maximum of the supply and is shown in the Fig: 4.8



**Fig 4.8: Variable Resistor (Potentiometer)**

**Presets:**

These are miniature versions of the standard variable resistor. They are designed to be mounted directly onto the circuit board and adjusted only when the circuit is built. For example to set the frequency of an alarm tone or the sensitivity of a light sensitive circuit. A small screwdriver or similar tool is required to adjust presets. Presets are much cheaper than standard variable resistors so they are sometimes used in projects where a standard variable resistor would normally be used and is shown in the Fig 4.9:

Multi turn presets: These are used where very precise adjustments must be made. The screw must be turned many times (10+) to move the slider from one end of the track to the other, giving very fine control.

|  |
| --- |
| **Fig 4.9: Preset Symbol** |



**Lcd Interfacing With The Microcontroller:**

VCC

FOR BACKLIGHT PURPOSE

GND

VCC

PRESET

(CONTRASTCONTROL)

GND

4 (RS) 1

5 (R/W) 2

6(EN) 3

D0  **LCD**

D1

D2

D3

D4

D5 15

D6 16

D7

D0

D1

D2

D3

D4

D5 15

D6 16

D7

P2.4

P2.5

P2.6

**89S52**  P0.0

P0.1

P0.2

P0.3

P0.4

P0.5

P0.6

P0.7

**Fig 4.10: Interfacing of LCD**

The interfacing from pin to pin is given in the above Fig: 4.10. The LCD is a 16\*2 LCD it has 16 pins. PORT 0(P0.0 – P0.7) set as data Port in 8 bit mode LCD display connect D0-D7 with PORT 0

* Ground the first lead of LCD (VSS) display.
* Connect VCC to 5v Power Supply
* Connect VEE with a Pot. It is used to adjust the Contrast of LCD display.
* Pins of 15 & 16 leads are used to power the back light of LCD.15 pin with 5v and Ground 16pin.
* Remaining pins of LCD (Pins 4, 5, 6) are connected to Port 2 pins(Port 2.5,2.6,2.7) of Microcontroller.

**4.7 FLOW CHART FOR TRANSMITTING PART:**

Start

Wait for ‘3’ sec to print & display

“Patient Monitoring System” on LCD display

No

LM 35 If Te=0

Print High Temperature

Yes

Print Low Temperature

No

Print Heart Beat=0 &timer=\*\*

Start Count Beat

Yes

Heart Beat count ++ and timer ++

Print Heart Beat on LCD display

After 20 sec stop the counting and send Msg through Zigbee

Process count value & send it as string Through Zigbee Transmitter

**Fig 4.11: Flow chart at transmitting session**

**4.8 FLOW CHART FOR RECEIVING PART:**

Start

Print ‘L’ On Monitoring System

If Te=1

After 20 sec the Data received from ZigBee transmitter

No

Yes

Print ‘H’ On Monitoring System

Print heart beat & temperature “\*\*\*\*”

on Monitoring System

**Fig 4.12: Flow chart at Receiving session**

There are different flow charts and the flow chart for Transmitting session is given in the Fig: 4.11 and the the Receiving section is given in the Fig :4.12

**CHAPTER 5**

**DESCRIPTION ABOUT KEILμVISION**

SOFTWARE

In this chapter software’s used for programming and dumping are explained.

**KEIL µVISION SOFTWARE:**

**5.1 Introduction:**

The µVision IDE from Keil combines project management, make facilities, source code editing, program debugging, and complete simulation in one powerful environment. The µVision development platform is easy-to-use and helping you quickly create Embedded programs that work. The µVision editor and debugger are integrated in a single application that provides a seamless Embedded project development environment. The µVision IDE is the easiest way for most developers to create Embedded applications using the Keil development tools. To launch µVision, click on the icon on your desktop or select Keil µVision4 from the Start Menu.A Project is the collection of all the source files as well as the compiler, assembler, and linker settings required to compile and link a program. µVision includes several robust features that make project management easy.

* **Device Support:**

One of the hardest parts of starting a new project is selecting the right mix of compiler, assembler, and linker options for the particular chip you use. µVision provides the [Device Database](http://www.keil.com/dd/) which makes this tedious task easy. When you create a new project, you select the chip you will use and µVision sets all the necessary assembler, compiler, and linker options automatically.

* **Symptom:**

The µVision device database contains all 8051, 251, C16x/XC16x/ST10, and ARM7/ARM9/Cortex-M standard products. However, there are some custom devices and there will be future devices that are not part of this database (they will be added as they are created, of course). If you need to work with an unlisted MCU, you have two alternatives:

* Select a generic device.
* Add a new device to the database.
* **Selecting a generic device:**

Under Generic listing in the device database, for 8051, 251, C16x/XC16x/ST10 you will find the following generic devices:

* 8031 (all Variants)
* 8032 (all Variants)
* 8051 (all Variants)
* 8052 (all Variants)
* C166 (all Variants) - Supports CPUs with no extended instruction set
* C167 (all Variants) - Supports CPUs with an extended instruction set

For ARM and Cortex devices, under the ARM listing in the device database, you will find the following generic devices:

* ARM7 (Big Endian)
* ARM7 (Little Endian)
* ARM966E-S (Big Endian)
* ARM966E-S (Little Endian)
* ARM9E-S (Big Endian)
* ARM9E-S (Little Endian)
* Cortex-M0
* Cortex-M0 SDK
* Cortex-M1
* Cortex-M1 (Altera)
* Cortex-M3
* Cortex-M4
* Cortex-M4 FPU
* Cortex-R4
* Cortex-R4F

You may select one of these devices and then specify any necessary chip options in the Target Dialog.

* **Adding a new device:**

From the File menu, select the Device Database item. µVision opens a dialog which shows the device database where you may add and modify the existing devices.

To add a new device there may be some steps

* Select a CPU that is similar to the device you want to use. Be sure to double-click to bring the settings for this device into focus.
* Change the name of the chip vendor.
* Select the appropriate CPU family.
* Enter the part number in the device text box.
* Modify the description to match your device.
* Change the options for the tools (see below).
* Single-click in the database window (to change focus). This enables the Add button.
* Click Add to add the new device.

To remove a device there may be some steps

* Locate the device you want to remove in the **Data Base** tree list box and left-click the name once to highlight it.
* Press the **Del** key to remove the device from the Device Database.

Note: You will not receive a warning when you press the **Del** key. The device is deleted immediately.

* Adjusting the Options...

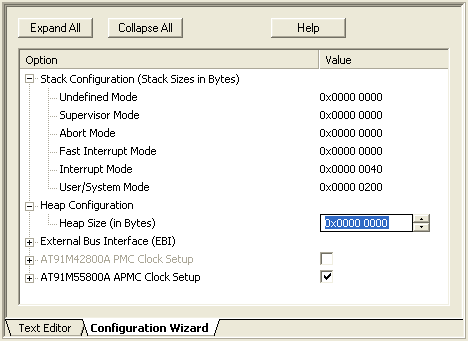
In the Options box, **CPU=** specifies the basic tool settings for the chip. The parameters for **CPU=** are:

* **IRAM (*range*)**  
  Address location of the on-chip IRAM.
* **XRAM (*range*)**  
  Address location of the on-chip XRAM.
* **XRAM2 (*range*)**  
  Address location of a 2nd on-chip XRAM area (i.e. for C167CS).
* **XRAM3 (*range*)**  
  Address location of a 3rd on-chip XRAM area.
* **IROM (*range*)**  
  Address location of the on-chip (flash) ROM.  
  For classic C166: start address must be 0; IROM is split into two sections if size is above 32KB; range specifies physical ROM size.
* **IROM2 (*range*)**  
  (for 8051 and variants only) Address location of 2nd on-chip ROM.
* **XROM (*range*)**  
  Address location of external (flash) ROM.
* **XROM2 (*range*)**  
  Address location of 2nd external ROM.
* **XROM3 (*range*)**  
  Address location of 3rd external ROM.
* **ICAN (*range*)**  
  Address location of the on-chip CAN module. Use for C167 and variants only.
* **CPUTYPE (*variant*)**  
  Specify CPU variant for ARM based controllers. Currently only ARM7, ARM9, Cortex-M and Cortex-R4 devices are supPorted.
* **EBIG**  
  Default to BIG Endean for ARM based controllers.
* **ELITTLE**  
  Default to LITTLE Endean for ARM based controllers.
* **ESEL**  
  Allow selection of the endianiess for ARM based controllers.
* **CLOCK (*val*)**  
  Default CPU clock speed (in Hz) used when you select the device. For example CLOCK (12000000), indicates a 12MHz oscillator.
* **FPU**   
  (for ARM9 only) Specifies default VFPU usage for some ARM9 devices.
* **FPU2**   
  (for Cortex-M4 only) Specifies default VFPU usage for some Cortex-M4 devices.
* **MASK\_REV (*val*)**  
  Specifies the mask revision number.
* **MDU\_F120**  
  Use the Multiply/Accumulate Unit of Silicon Labs C8051F12x device variants.
* **MDU\_R515**  
  Use the Multiply/Divide Unit of Cast/Evatronix R80515.
* **MOD167**  
  Use the extended instruction set of the C167 and variants.
* **MOD517DP**  
  Enable Infineon specific multiple DPTR registers.
* **MOD517AU**  
  Enable the Infineon specific Arithmetic Unit.
* **MODA2**  
  Enable Atmel specific multiple DPTR registers (like on AT89S8252).
* **MODAB2**  
  Enable Analog Devices specific multiple DPTR registers.
* **MODC2**  
  Enable Cast/Evatronix specific multiple DPTR registers (R80515).
* **MOD\_CONT**  
  Enable support for the Dallas Contiguous Mode.
* **MODDA**  
  Enable Dallas specific Arithmetic Accelerator.
* **MODDP2**  
  Enable Dallas specific multiple DPTR registers.
* **MODH2**  
  Enable Hynix/ST uPSD33xx uPSD34xx multiple DPTR registers.
* **MODP2**  
  Enable Philips specific multiple DPTR registers. (Note also some Atmel devices are using this variant).
* **MODV2**  
  Use the Infineon XC16x / ST Super10 instruction set extensions.
* **MX**  
  Enable support for the Philips 80C51MX architecture.
* **MXP**  
  Enable support for the Philips Smart MX SmartCard architecture.
* **DPX**  
  Enable 24-bit DPTR register for the Analog Devices ADuC812.
* **PMW**  
  Enable the PCON.PMW feature that allows using MOVX instructions to write into code space for the Evatronix R8051XC.
* **DPC**  
  Enable the data pointer control registers on the Evatronix R8051XC that provide auto-increment features for the DPTR registers.
* **BSE**  
  Enable the Bank Switch Enable feature in the register DPSEL.3 for the Evatronix R8051XC.
* **PSOC**  
  Enable the generation of interrupt vectors for Cypress PSoC.

**5.2 STARTUP CODE:**

Configuring start-up code can be one of the most frustrating aspects of Embedded software development .The µVision IDE automatically includes the appropriate start-up code (base on the device you select) and provides a known foundation from which to start.

The [Configuration Wizard](http://www.keil.com/uvision/ide_edt_cfgwizard.asp) helps you set start-up options for your target hardware using familiar dialog controls and is shown in Fig: 5.1.



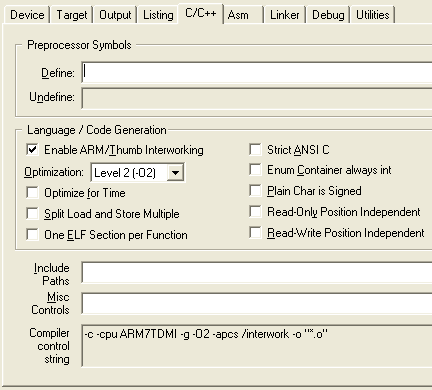
**Fig 5.1:µvision 2 Dialog Box 1**

**5.3 OPTION SETTINGS:**

µVision lets you set the options for all files in a target, a group, or even a single source file.

Options for Target ButtonClick the **Options for Target** button on the toolbar to change the project options for the currently selected target.

In the Project Workspace, you may right-click the target, group, or source file to open the options dialog specific to that item. The Schematic Representation is given below in the Fig: 5.2.



**Fig 5.2: µvision 2 Dialog Box 2**

The Options Dialog offers several Tabs where you specify option settings:

* The **Device** tab allows you to select the device for this target.
* The **Target** tab allows you to specify the memory model and memory parameters. You may enter the external (or off-chip) memory address ranges under External Memory. When you start a new project, you typically only need to setup the options on this tab.
* The **Output** tab allows you to specify the contents of the output files generated by the assembler, compiler, and linker.
* The **Listing** tab allows you to configure the contents of the listing files.
* The **C/C++**, **Asm**, and **Linker** tabs allow you to enter tool-specific options and display the current tool settings.
* The **Debug** tab configures the µVision Debugger.
* The **Utilities** tab configures Flash memory programming for your target System.

**5.4 SOURCE FILES:**

The source files in your µVision project display in a Project Workspace.

* Each Project can be configured to generate one or more Targets. Each Target has its own option settings and output file name that you may define. You may create one Target for testing with the simulator and another Target for a release version of your application that will be programmed into Flash ROM.
* Within a Target, you may have one of more file Groups which allow you to associate source files together. Groups are useful for grouping files into functional blocks or for identifying engineers in a software team.
* The Project Workspace window displays the target (AT91M55800A in this case) along with all file groups and source files that comprise your project.

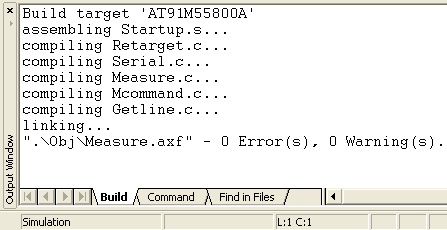
The Project menu provides access to all dialogs for project management including...

* **New Project**... which creates a new project.
* **Targets, Groups, Files**... which add components to a project. The Local menu in the Project window allows you to add files to the project.
* **Open Project**... which opens an existing project.

**5.5 BUILDING PROJECTS:**

The Project menu provides access to all dialogs for project management including...

* **New Project**... which creates a new project.
* **Targets, Groups, Files**... which add components to a project. The Local menu in the Project window allows you to add files to the project.
* **Open Project**... which opens an existing project.



**Fig 5.3: µvision 2 Dialog Box 3**

You may double-click on an error or warning to immediately begin editing the file with the problem--even while µVision continues compiling your source files in the background. The line numbers for errors and warnings are synchronized even after you make changes to the source file(s) also represented in the above Fig 5.3:

To get more information about a particular error message, select the message and press **F1** for full help text.

If you enable global optimizations, µVision re-compiles your source files to achieve the most optimal global use of registers.

* **Editing files:**

Color syntax highlighting and text indentation are optimized for editing C source code. Most editor functions are quickly accessed from the toolbar. The editor is available while debugging.

This gives you a natural debugging environment that helps you quickly identify and correct errors in your source code.

* **Code Completion:**

New editor features enhance your productivity while developing C/C++ source code. The Code Completion List and Function Parameter information helps you to keep track of symbols, functions, and parameters. Dynamic Syntax Checking validates the program syntax compilation.

* **Executing Code:**

µVision provides run controls to start, stop, and step through your Embedded application. You may use breakpoints and trace memory to conditionally halt program execution and review previously executed program code.

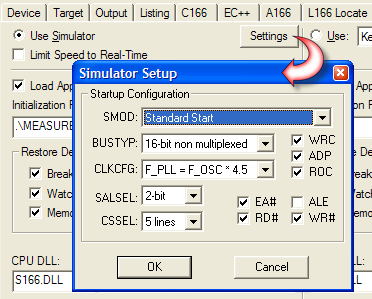
* **Reset:**

Many devices self-configure based on the state of certain pins at reset time and is represented in Fig: 5.4.

The Settings button opens the Simulator Setup dialog where the µVision Simulator allows you to configure the reset parameters of these devices.

Parameters like bus configuration, watch dog timer setup, and clock selection are available.

Reset ButtonWhen you click the **Reset** button on the toolbar to reset the Microcontroller, µVision restarts the simulation using the new reset options.



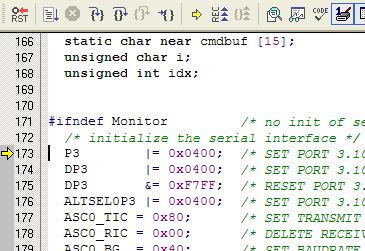
**Fig 5.4: µvision 2 Dialog Box 4**

* **Run/Stop:**

Starting and stopping program execution may be controlled using commands you enter in the Output Window or using buttons on the toolbar. Click the **Run** button on the toolbar to begin executing your target program in the µVision debugger and is represented in Fig: 5.5.

The **Run** button executes code until a [breakpoint](http://www.keil.com/uvision/db_exe_breakpoints.asp) is reached.

Stop Execution ButtonClick the **Stop** button on the toolbar to halt program execution.



**Fig 5.5: µvision 2 Dialog Box 5**

In the Output Window, type **g, label** to execute program code and stop when *label* is reached.

You may also type **G** to run and **Ctrl-C** to stop.

**5.6 SINGLE STEPPING:**

The µVision supPorts various methods of single-stepping through your application.

Step Into ButtonClick the **Step Into** button on the toolbar to execute a single instruction or line of code (single-step). **Step Into** single-steps into all called functions.

Step Over ButtonClick the **Step Over** button on the toolbar to execute a single instruction or line of code (single-step). **Step Over** executes all called functions without stepping into them.

Step Out ButtonClick the **Step Out** button on the toolbar to step out of a function and return to the caller.

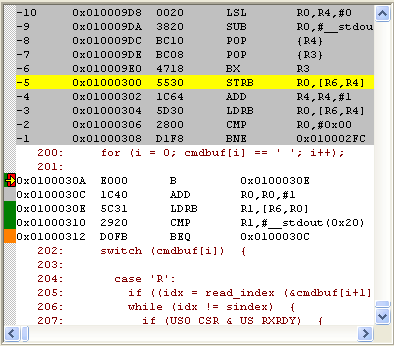
Run to Cursor ButtonClick the **Run to Cursor** button on the toolbar to begin executing your target program until the current cursor line is reached.

* **Execution Stepping:**

As your program runs, the µVision debugger optionally stores trace information for each instruction that executes. Trace recording lets you analyze the program flow prior to a breakpoint or manual halt.

Toggle Trace Recording ButtonThe **Toggle Trace Recording** button on the toolbar enables or disables trace recording.  
  
View Trace ButtonThe **View Trace Records** button on the toolbar displays the trace buffer (shaded in grey).

As you scroll through the trace records, information displayed in the register window is updated to reflect register values when the selected instruction executed represented in Fig: 5.6



**Fig 5.6: µvision 2 Dialog Box 6**

**5.7 BREAKPOINTS:**

Breakpoints are trigger points in your program that halt execution or execute a debugger function. Breakpoints may be triggered by reading an instruction from program memory (an execution breakpoint), reading or writing a memory location (a memory access breakpoint), or by calculating a true value for a conditional expression (a conditional breakpoint).

The µVision IDE allows you to set breakpoints while writing your source code and while debugging your program. Breakpoints you set while editing are activated in your debug session. You may right-click or use the buttons on the toolbar to set breakpoints on source lines and is represented in Fig: 5.7

Insert Breakpoint ButtonThe **Insert/Remove Breakpoint** button toggles (sets or clears) an execution breakpoint on the current line.

Enable Breakpoint ButtonThe **Enable/Disable Breakpoint** button enables or disables an existing breakpoint on the current line. The breakpoint is not removed.

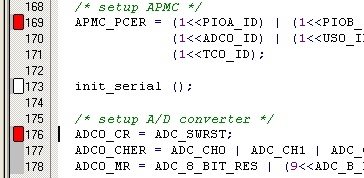
Disable All Breakpoints ButtonThe **Disable All Breakpoints** button disables all breakpoints in the program. The breakpoints are not removed.

Kill All Breakpoints ButtonThe **Kill All Breakpoints** button removes all breakpoints in the program.

The status of each line of code is marked in the **Attributes** column of the editor window.

Enabled BreakpointRed blocks mark enabled breakpoints.

Disabled BreakpointWhite blocks mark disabled breakpoints.

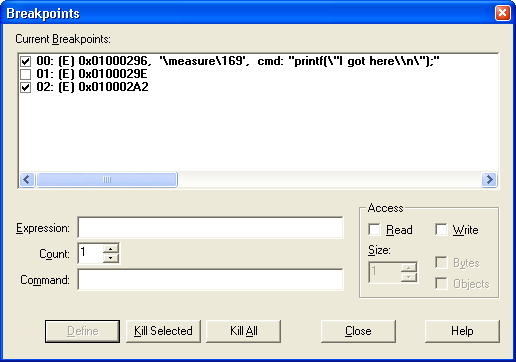


**Fig 5.7: µvision 2 Dialog Box 7**

The µVision debugger provides several ways for you to create and manage breakpoints in your program.

* You may double-click a line of code to toggle an execution breakpoint for that line. You may also use buttons on the toolbar to toggle an execution breakpoint.
* You may enter all types of breakpoints using the command line in the Output Window.
* You may open then Breakpoints dialog where you may edit and redefine all types of breakpoints.

Advanced breakpoint features are available from the Breakpoints dialog and is represented in Fig: 5.8



**Fig 5.8: µvision 2 Dialog Box 8**

By default, breakpoints you create are saved and restored in subsequent debugging sessions.

**5.8 CORE SIMULATION:**

The µVision Simulator allows you to debug programs using only your PC and device simulation drivers provided by Keil and various third-party developers. A good simulation environment, like µVision, does much more than simply simulate the instruction set of a Microcontroller — it simulates your entire target System including interrupts, start-up code, on-chip peripherals, external signals, and I/O.

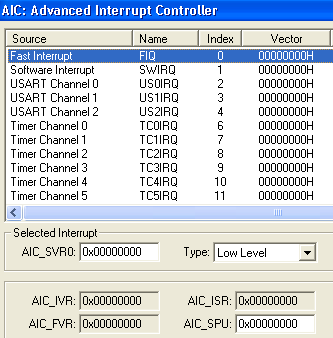
* **Interrupt Simulation:**

Interrupts are fully supported and properly simulated in the µVision Debugger. Interrupts are triggered and executed exactly as they would be in a real target System.

An Interrupt System dialog box is designed specifically for each supPorted device. Each interrupt source including all interrupt configuration options are displayed.

You may use this dialog to interactively enable and disable interrupts, change the interrupt priority, enable or disable an interrupt request, and change the global interrupt flag.

Further more, interrupt simulation enables you to set breakpoints within and stop program execution inside an interrupt service routine represented in given Fig: 5.9

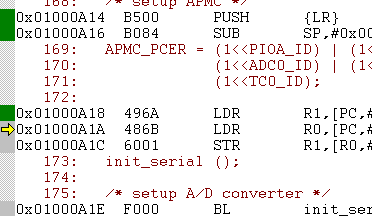


**Fig 5.9: µvision 2 Dialog Box 8**

* **Instruction Simulation:**

The µVision Debugger provides complete instruction set simulation for all ARM7, ARM9, Cortex-M3, XC16x, C16x, ST10, 251, and 8051 devices.

When debugging your program, op-codes are interpreted and executed as their corresponding instructions would be. You may view program disassembly in mixed mode or in assembly code. The Schematic Representation is given in the Fig: 5.10.

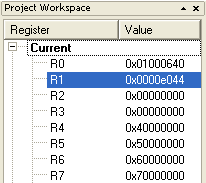


**Fig: 5.10 µvision 2 Dialog Box 10**

All registers and flags are updated as each instruction executes. Results display in the Register Tab of the Project Workspace.

As you step through your program, affected registers are highlighted.

Instruction timings are accurately simulated so you can easily determine how long a function or module takes to execute. Timing is cycle-accurate for deterministic parts and is represented Fig: 5.11.



**Fig 5.11: µvision 2 Dialog Box 11**

**5.9 DEBUGGER FUNCTIONS:**

The µVision debugger incorporates a C function language you can use to respond to program output and create input stimulus. A number of built-in debugger functions, like printf, memset, and rand, are available.

You may create**:**

* Signal functions to simulate analog and digital inputs to the CPU.
* User functions that extend the command scope of the debugger and combine repetitive actions.

Signal functions describe the behaviour of external hardware and supply input to the Microcontroller's I/O pins. Signal functions run in the background while the µVision debugger simulates your target program.

There are several ways you can invoke debugger functions:

* From the Command Line in the Output Window,
* From a button in the [Toolbox](http://www.keil.com/uvision/db_sim_toolbox.asp),
* From a [breakpoint](http://www.keil.com/uvision/db_exe_breakpoints.asp) definition.

**5.10 TOOLBOX FUNCTIONS:**

The µVision Debugger Toolbox contains user-configurable buttons you may assign to user or signal [functions](http://www.keil.com/uvision/db_sim_functions.asp) or to debugger commands.

Toolbox ButtonClick the Toolbox button on the toolbar to display the Toolbox Window.

When you click on a Toolbox button, the debugger executes the associated function or command. Toolbox buttons may be used at any time, even while executing your target program.

You may define Buttons for the Toolbox in the Command window Which is represented in Fig: 5.12.



**Fig 5.12: µvision 2 Dialog Box 12**

* **Agsi Drivers:**

AGSI is an Application Program Interface (API) third-party developers can use to create simulation drivers that interface directly with the Keil µVision Simulator. The purpose of AGSI is to provide a flexible, high-fidelity simulation environment for Embedded software development.

AGSI drivers are DLLs that are created using Microsoft Visual C++ and template files provided by Keil. You can find more information in:

AGSI drivers can simulate such things as:

* LCD Panels with full graphic display capability.
* Keyboards.
* External peripherals and devices.
* Complete System hardware and environment.

Third-Party AGSI Drivers

* [Digital Core Design (DCD): DCD IP Core Simulation](http://www.keil.com/uvision/agsi/383.htm)
* [Lab centre Electronics: Proteus VSM](http://www.keil.com/uvision/agsi/193.htm)
* [MEBA: 8051 State Machine](http://www.keil.com/uvision/agsi/183.htm)
* [MEBA: I&sup2;C Simulation](http://www.keil.com/uvision/agsi/151.htm)
* [MEBA: LCD Simulation (HD44780 Controller)](http://www.keil.com/uvision/agsi/186.htm)
* [MEBA: LED simulation](http://www.keil.com/uvision/agsi/184.htm)
* [MEBA: Signal Generator](http://www.keil.com/uvision/agsi/185.htm)

**5.11 PERIPHERAL SIMULATION:**

The µVision debugger simulates the on-chip peripherals of numerous Microcontrollers. When you select a Microcontroller from the [device database](http://www.keil.com/uvision/ide_prj_dd.asp) to configure your project, µVision automatically configures the debugger's peripheral simulator for you. With its logical and timing simulation, it is possible to test an application before the target hardware is even completely designed. The simulator makes it easy to test hardware defects and critical situations which are difficult to debug with real hardware.

|  |
| --- |
| * [**A/D Converter**](http://www.keil.com/uvision/db_sim_prf_adc.asp): A/D Converter Simulation allows you to easily configure the A/D Converter and simulate voltages are corresponding device pins. |
| * [**I/O Port**](http://www.keil.com/uvision/db_sim_prf_portio.asp): I/O Port Simulation allows you convenient dialogs to configure and monitor I/O Port status. |
| * [**Timer/Counter**](http://www.keil.com/uvision/db_sim_prf_timer.asp): Timers and Counters are accurately simulated allowing you to control and measure events precisely. |
| * [**Watchdog Timer**](http://www.keil.com/uvision/db_sim_prf_watchdog.asp): Watchdog Timer Simulation helps you determine how often and where to reset the watchdog in your application. |
| * [**Capture/Compare**](http://www.keil.com/uvision/db_sim_prf_capcom.asp): Capture/Compare Simulation allows you to view and change the configuration of the cap com units fond on many devices. |
| * [**Serial Communications**](http://www.keil.com/uvision/db_sim_prf_uart.asp): Serial Communications Simulation provides a serial window and dialog that help you configure and communicate using your device's on-chip UART. |
|  |
| * [**I²C Communications**](http://www.keil.com/uvision/db_sim_prf_i2c.asp) I²C Communications Simulation allows you to configure I²C and view messages sent and received. Debugger scripts may be used to generate and respond to I²C messages. |
|  |
| * [**FLASH Memory**](http://www.keil.com/uvision/db_sim_prf_flash.asp): FLASH Memory Simulation gives you access to all control registers of on-chip FLASH/EE memory. All memory contents may be viewed and modified in real-time. |
| * [**Power-Saving Modes**](http://www.keil.com/uvision/db_sim_prf_power.asp): Power-Saving Modes Simulation simulates the Idle and Power Down Modes found on many devices. |

**5.12 TARGET DEBUGGING:**

The µVision Debugger allows you to debug programs running on target Systems using target drivers provided by Keil and various third-party developers.

|  |
| --- |
| [**Target Monitor**](http://www.keil.com/uvision/db_trg_monitor.asp): A target monitor interfaces your PC to your target System using RS-232 and software. |
| [**Flash Programming**](http://www.keil.com/uvision/db_trg_flash.asp): Flash programming uses a target interface to download your target program to Flash memory. |
| [**AGDI Drivers**](http://www.keil.com/uvision/db_trg_agdi.asp): AGDI drivers interface the µVision Debugger to third-party hardware or provide additional debugger features. |

|  |
| --- |
|  |
|  |

**CHAPTER 6**

**SOFTWARE IMPLEMENTATION**

**CODE:**

#include <AT89X52.h>

#include "AT\_serial.h"

#define beat P1\_0

#define lm35 P2\_0

#define display Lcd8\_Display(0xc0,"heartbbeat:",11);

unsigned char temp,te,respiration,heart\_beat,count,sec,v[10],a;

void beat\_count();

void Bee(unsigned char x);

void main()

{

Lcd8\_Init();

Lcd8\_Display(0x80,"PATIENT MONITRNG",16);

Lcd8\_Display(0xC0," SYSTEM ",16);

Delay\_(55000,3);

display;

while(1)

{

beat\_count();

if(lm35)

{

te=0;

Lcd8\_Display(0x80,"LOW TEMP",8);

}

if(!lm35)

{

Lcd8\_Display(0x80,"HIGH TEMP",8);

te=1;

}

// if(sec>30) { sec=0; send(); }

if(sec>20)

{

//if(num\_ready)value\_compare();

if(te)

{

Serial\_Out('H');

}

if(!te)

{

Serial\_Out('L');

}

//Bee(respiration);

Bee(heart\_beat);

Enter;

sec=heart\_beat=respiration=0;

}

}

}

void beat\_count()

{

if(!beat&&!y) { Delay(800); heart\_beat++; y=1; }

if( beat&& y) { y=0; Delay(800); }

Lcd8\_Decimal2(0xCB,heart\_beat);

Lcd8\_Decimal2(0xCE,sec);

}

void timer0(void) interrupt 1

{

count++;

if(count>=15){count=0; sec++;}

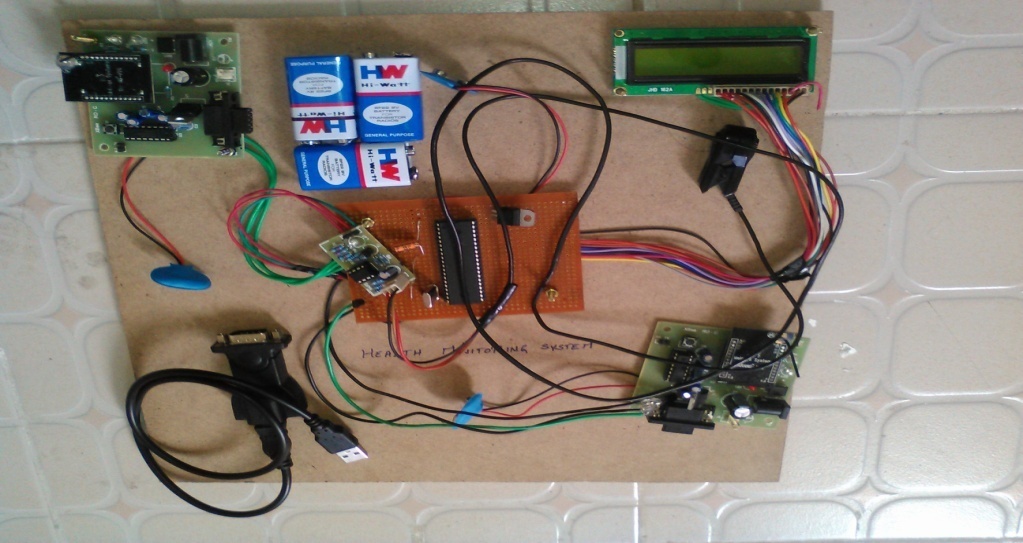
}

**CHAPTER 7**

**RESULT ANALYSIS**

In this chapter the final result of the project& conclusion are explained.

The code for the Microcontroller is written in Embedded C and the code is executed using the Keil Software.



**Fig 7.1: Total project kit**

When the System gets turned on Patient Monitoring System is displayed on the LCD the figure is represented given below Fig: 7.2



**Fig 7.2: Liquid Crystal Display on the Kit**

It needs to take some time for the start of the System then it find out the room temperature and start counting the heart beat up to 20sec it is represented in the given Fig 7.3:



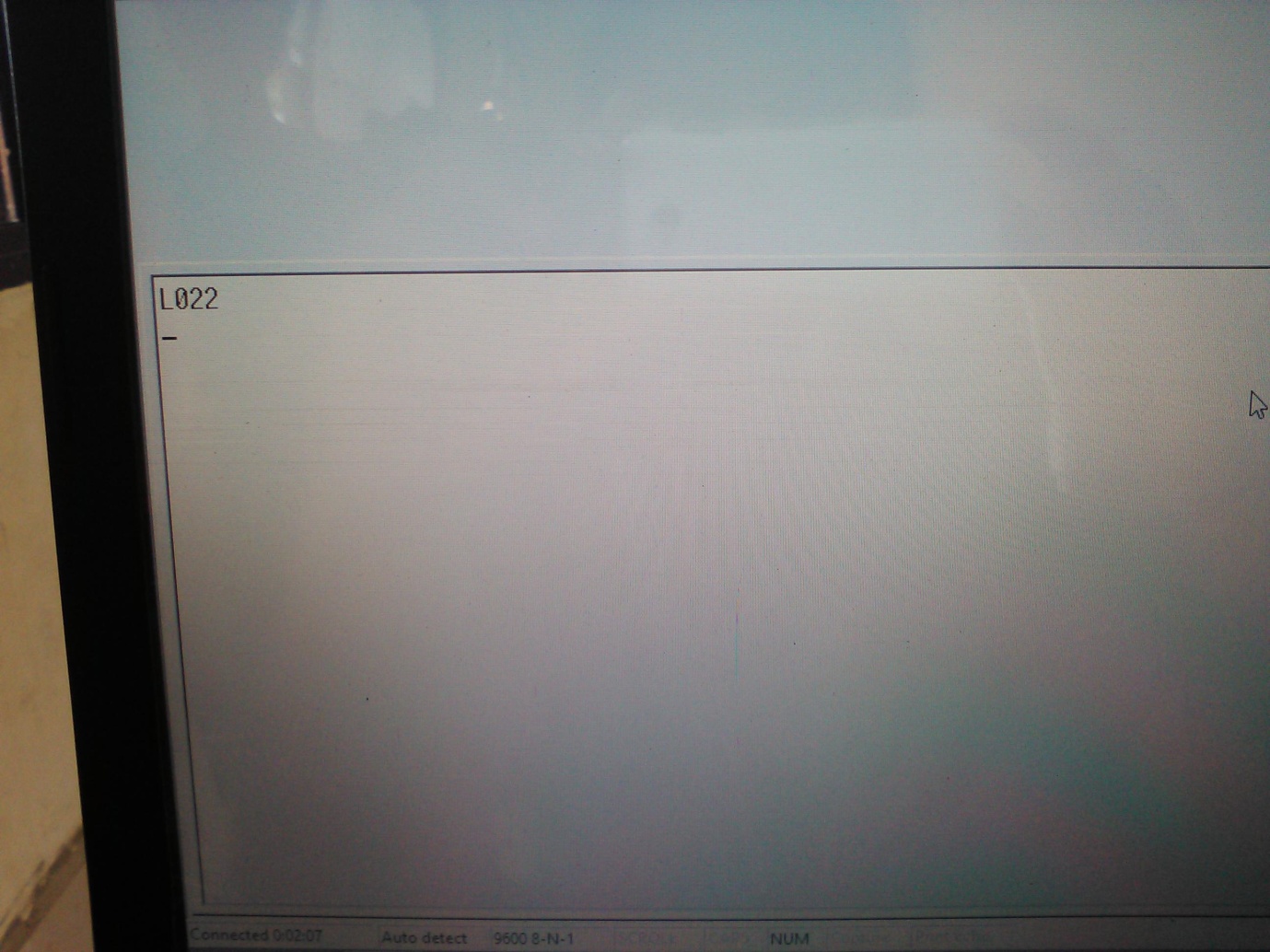
**Fig 7.3: Liquid Crystal Display showing the Temperature on the kit initially**

The patient total beats for 20 seconds is counted by microcontroller it will be displayed in LCD. According to our blood volume the Heart Beat is varied time to time it was resented in Fig 7.4:



**Fig 7.4: LCD showing the Temperature on the kit after 20 seconds**

The Heart Beat count is transmitted to Receiver By using the ZIGBEE Module. So The Resulting output is displayed on the Monitor in the Doctor Room. It was represented in Fig: 7.5



**Fig 7.5: Temperature and the Heart Beat on the Display of the Patient room**

CHAPTER 8

**CONCLUSION AND FUTURE ENHANCEMENT**

**8.1 CONCLUSION:**

In this project we represent By using this prototype circuit containing 89S52, ZigBee, LCD and other hardware circuit so that the messages can be transferred at fixed time intervals to the corresponding medical expert to give necessary precautions to take care about the patient. This System has the following features:

* 1. 89S52 MCU consumes low power with suitable devices for interconnection.
  2. Continuous monitoring of patients is done which is simple by using ZigBee. The device is designed to provide a continuous access to a person’s heart rate and temperature monitoring & inform through wireless communication.

The heart beat sensor which detects heart beat is interfaced to Microcontroller along with LCD, which display the heart beat rate. The goal of the project is to reduce the hospitalization and assistance cost. Health monitoring application is mainly proposed to provide alerts for medical health monitoring staff for the patients when needed. The device can be improved in certain areas as listed below: ( i ) A graphical LCD can be used to display a graph of the change of heart rate over time. ( ii ) Sound can be added to the device so that a sound is output each time a pulse is received. (iii). Serial output can be attached to the device so that the heart rates can be sent to a PC for further online or offline analysis

**8.2 FUTURE ENHANCEMENT:**

For the future scope of our project we need to consider the GSM model instead of ZIGBEE.

By this we can also send the message not only to the patient care taker but also to their family members. This yields to the results that continuous monitoring will be done for the patient.

**REFERENCES**

* Weidong Xiao, William G. Dunford, Patrick r. Palmer andAntoine Capel,

Regulation of Photovoltaic voltage,IEEETrans .Industrial Electronics. vol. 54

no.3, pp. 13651373, June 2012.

* Barth, H. Schaeper, C. Schmidla, T. Nordmann, H. Kiel,M. van der Broeck, H.

Yurdagel, Y. Wieczorek, C. Hecht, F.Sauer, D.U. Development of a universal

adaptive batterycharger as an educational project.Power Electronics Specialists

Conference, 2008. PESC2008. IEEE19 June 2010,

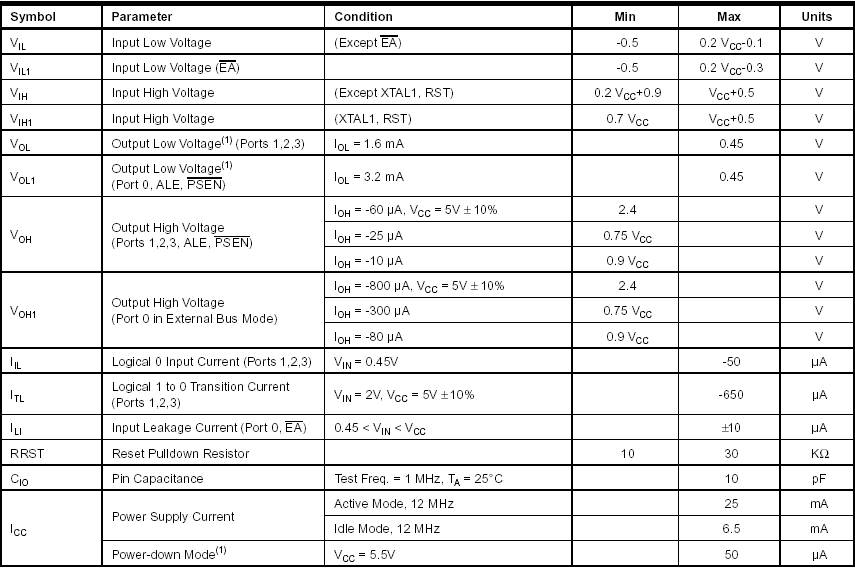
* Rodriguez, Jose; et al. (August 2002). "MultilevelInverters: A Survey of

Topologies, Controls, andApplications" IEEE Transactions on Industrial

Electronics(IEEE) 49 (4): 724–738

* http/: electronics\_components.globalspec
* http://electroniccomponents.globalspec.com/LearnMore/Electrical\_Electronic\_Components/Power\_Supplies\_Conditioners/Voltage\_Multipliers

**APPENDIX-A**

**DC Characteristics of AT89C52:**

**APPENDIX-B**

**Regulator LM7805**

**D.1 Absolute Maximum Ratings:**

Absolute maximum ratings are those values beyond which damage to the device may occur. The datasheet specifications should be met, without exception, to ensure that the System design is reliable over its power supply, temperature, and output/input loading variables. Fairchild does not recommend operation outside data sheet specifications.

Table 1:

****