HEART RATE MONITOR USING ARDUINO MICROCONTROLLER

A MINI PROJECT REPORT

Submitted by

BARATH KUMAR.S	**(815120121008)**
BOOPATHI.D	**(815120121011)**
DHIVAHAR.P	**(815120121015)**
JAYAPRAKASH.M	**(815120121020)**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

DEPARTMENT OF BIO-MEDICAL ENGINEERING

DHANALAKSHMI SRINIVASAN INSTITUTE OF TECHNOLOGY

(Approved by AICTE & Affiliated to Anna University Chennai)

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2023

HEARTRATE MONITOR USING ARDUINO MICROCONTROLLER

A MINI PROJECT REPORT

Submitted by

BARATH KUMAR.S	**(815120121008)**
BOOPATHI.D	**(815120121011)**
DHIVAHAR.P	**(815120121015)**
JAYAPRAKASH.M	**(815120121020)**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

DEPARTMENT OF BIO-MEDICAL ENGINEERING

DHANALAKSHMI SRINIVASAN INSTITUTE OF TECHNOLOGY

(Approved by AICTE & Affiliated to Anna University Chennai)

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY : CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this report titled " **HEART RATE MONITOR USING ARDUINO MICROCONTROLLER"** is the bonafide work of **"BARATHKUMAR.S (815120121008), BOOPATHI.D (815120121011), DHIVAHAR.P(815120121015), JAYAPRAKASH.M (815120121020)"** who carried out the work under my supervision.

SIGNATURE

Dr. B. SUGANTHI M.E., Ph.D.,
HEAD OF THE
DEPARTMENT
DEPT. OF BIOMEDICAL
ENGINEERING
DHANALAKSHMI
SRINIVASAN INSTITUTE
OF TECHNOLOGY,
SAMAYAPURAM.

SIGNATURE

Mrs.M.DHAYALINI .,M.E
SUPERVISOR
ASSITANT PROFESSOR
DEPT.OF BIOMEDICAL
ENGINEERING
DHANALAKSHMI
SRINIVASAN INSTITUTE
OF TECHNOLOGY,
SAMAYAPURAM.

HEART RATE MONITOR USING ARDUINO MICROCONTROLLER

A MINI PROJECT REPORT

Submitted by

BARATH KUMAR.S	**(815120121008)**
BOOPATHI.D	**(815120121011)**
DHIVAHAR.P	**(815120121015)**
JAYAPRAKASH.M	**(815120121020)**

VIVA VOCE EXAMINATION

The viva voce examination of this project work has been done as a part of curriculum in Bachelor of Engineering degree in **"BIOMEDICAL ENGINEERING''** is held on …………

INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

We hereby declare that the work entitled " **HEART RATE MONITOR USING ARDUINO MICROCONTROLLER"** is submitted in partial fulfilment of requirement for the award of the degree in B.E (Biomedical) Anna University Chennai. It a record of our own work carried during the academic year 2022-2023 under supervision and guidance of **Dr. B. SUGANTHI M.E., Ph.D** Head of the Department and professor, Department of Biomedical Engineering, **Dhanalakshmi Srinivasan Institute of Technology, Samayapuram -621112.**

The extent and source of information are derived from the existing literature and have been indicated through dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other degree or diploma, either in this or any other university.

REGISTER NUMBER	NAME
815120121008	BARATH KUMAR.S
815120121011	BOOPATHI.D
815120121015	DHIVAHAR.P
815120121020	JAYAPRAKASH.M

PLACE: Samayapuram, Tiruchirappalli

DATE :

ACKNOWLEDGEMENT

At the outset we express our gratefulness to the **ALMIGHTY** who has been with us during each and every step we have taken towards the completion of this project.

We convey our sincere thanks to the chairman of our college **SHRI. A. SRINIVASAN**, without whose permission and patronage, this project would not have been a success and also to our beloved chairperson, **Mrs.K. ANANTHALAKSHMI KATHIRAVAN**, of our college for providing dexterities made available in our college.

We express our sincere thanks and gratitude to our respected principal of our college **Dr. S. SELVAKUMAR MCA M.Tech., Ph.D.**, for his kind permission to work on this project.

We are extremely grateful to **Dr. B. SUGANTHI M.E., Ph.D**., Head of the Department, Biomedical Engineering for her valuable guidance and the suggestions throughout this project work to complete within stipulated time.

We highly indebted our guide, **Mrs.M.DHAYALINI.,M.E.,** for her valuable advice and guidance without which we could not be possible to carry out this project in a commendable manner.

We express our gratefulness to our entire department faculty members for their contribution and cooperation at every stage of our project.

TABLE OF CONTENTS

ABSTRACT

Heart beat sensor is an electronic device that is used to measure the heart rate speed of the heartbeat. M body temperature, heart rate and blood pressure are the basic things that we do in order to keep us healthy. In measure the body temperature, we use thermometers and a sphygmomanometer to monitor the Arterial Pr Blood Pressure. Heart Rate can be monitored in two ways: one way is to manually check the pulse either at neck and the other way is to use a Heartbeat Sensor. Pulse oximetry is used in this project to detect the heartb fingers. When the heart expands (diastole) the volume of blood inside the fingertip increases and when contracts (systole) the volume of blood inside the fingertip decreases. The resultant pulsing of blood volume i fingertip is directly proportional to the heart rate and if you could somehow count the number of pulses in on that's the heart rate in beats per minute (bpm). For this an IR transmitter/receiver pair (LED) placed in clos with the fingertip. The pulsating refl converted to a suitable current or voltage pulse by the sensor. The sensor output is processed by suitable e circuits to obtain a visible indication (digital display). Le speed of the heartbeat Monitoring body temperature, heart rate and blood pressure are the basic things that we do in order to keep us healthy.In order to measure the body temperature, we use thermometers and a sphygmomanometer to monitor the Arterial Pressure or Blood Pressure.

LIST OF FIGURES

LIST OF ABBREVIATIONS

BIA	BIOELECTRICAL IMPEDANCE ANAYSIS
EAMC	EXERCISE ASSOCIATED MUSCLE CRAMPS
RF	RADIO FREQUENCY
SVM	SUPPORT VECTOR MACHINE
USG	URINE SPECIFIC GRAVITY
LCD	LIQUID CRYSTAL DISPLAY
PSU	POWER SUPPLY UNIT
ADC	ANALOG TO DIGITAL CONVERTOR
UART	UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER
SPI	SERIAL PERIPHERAL INTERFACE
PIC	PERIPHERAL INTERFACE CONTROLLER
EEPROM	ELECTRONICALLY ERASABLE PROGRAMMABLE READ ONLY MEMORY
RST	RESET
ALU	ARITHEMETIC LOGIC UNIT
MSSP	MASTER SYNCHROUS SERIAL PORT
CPU	CENTRAL PROCESSING UNIT
SSP	SYNCHROUS SERIAL PORT
FSR	FILE SELECT REGISTER

CHAPTER 1

1.1 INTRODUCTION

Traditional heart monitoring solutions exist for many years such as the Holter device which records the patient for 24 to 48 hours and is then analyzed afterwards by the cardiologist. The patient can 'wear' the device and go home and resume his/her normal activities. The main drawback of these solutions is when a major incident occurs during the monitoring phase which is recorded but no immediate action is taken to help the user.

As our project involves the design and development of medical equipment, as a precautionary measure, in order to minimize failures we take some necessary steps to intimate the respective doctors in time in case of emergencies using the GSM and through short mail service and see to it that such misfortunes are controlled and taken care.

The heart patient is monitored using various types of sensors (ECG, accelerometer, Oxygen). The sensor information is collected and transferred. One distinction of our solution compared to the others is that we can personalize the monitoring and we have mechanisms in place to locate the user in case of emergency whether the patient is indoors or outdoors. We detect life threatening arrhythmias and give the patient general information about their health when they are not in a dangerous situation. We can also store extra information for further use by the health providers.

The heart monitoring system designed in our project is such that if the first number dialed is busy, not reachable or out of service, the second number stored will be dialed and so on. Therefore, this special feature of the system makes it more advantageous and preferable.

To ensure good medical service:

- Be aware of patients medical record
- Co-operate with the various activities of the industry
- Take preventive measures in a proactive way so as to provide fast and best medical service.
- Take appropriate actions under emergency conditions.

1.2 Proposed Work

Some severe diseases and disorders e. g. heart failure needs close and continual monitoring procedure after diagnosis, in order to prevent mortality or further damage as secondary to the mentioned diseases or disorders. Monitoring these types of patients, usually, occur at hospitals or healthcare centers. Heart arrhythmias for instance, in many cases, need continual long-term monitoring. However, the patients are often too early released, owing to need of hospital bed for another patient on the waiting list, who needs to be hospitalized immediately.

1.3 Scope of the work

Long waiting time for hospitalization or ambulatory patient monitoring/treatment, are other well-known issues for both the healthcare institutions and the patients. This project provides healthcare authorities to maximize the quality and breadth of healthcare services by controlling costs. As the population increases and demand for services increases, the ability to maintain the quality and availability of care, while effectively managing financial and human resources, is achieved by this project. The use of modern communication technology in this context is the sole decisive factor that makes such communication system successful.

1.4 Design methodology

In transmitter circuit the Heat Beat is measured by LED and LDR, then it is applied to the microcontroller. The Microcontroller maintains the records of the measured readings. It compares the measured heart beat with the normal readings and checks it is within the normal range or not. If it is normal, then it keeps record of the same and the readings in SMS (Short Messaging Service) form to the specified mobile number. The time specified for sending message is given by the user.

CHAPTER 2

WORKING PRINCIPLE

2.1 BLOCK DIAGRAM

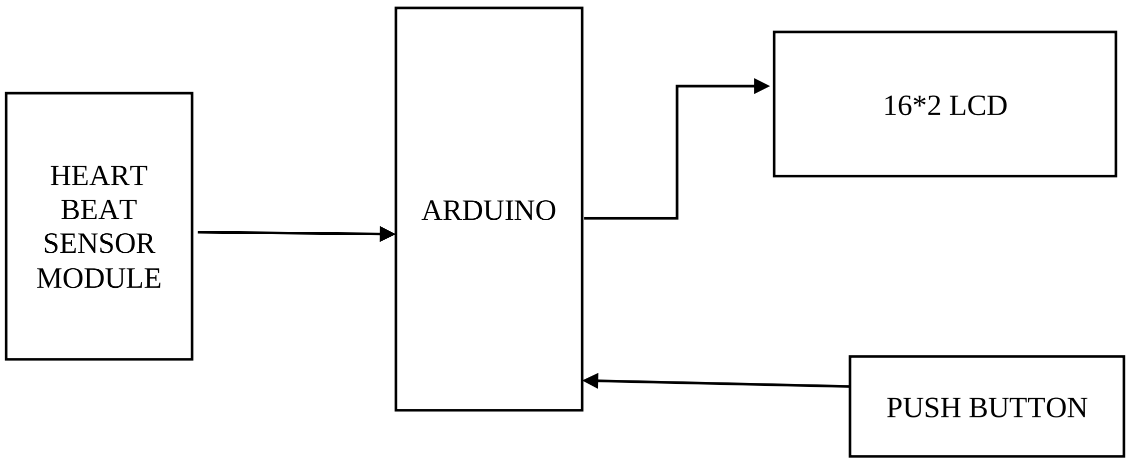

Fig 2.1 Block diagram of system design

In the above figure heart pulses are monitored continuously. There are seven sensors programmed with upper and lower limits for each of them as per the user's convenience. These sensors keep monitoring the respective parameter variations as a measure of precaution. The liquid crystal display displays the current sensor readings.

The sensor readings and the limits stored in the memory is compared by the microcontroller every time. Whenever the readings sensed is out of limits, the microcontroller fetches the phone number stored in the memory and gives command to the GSM circuit to the dial to the concerned person. If the first number dialed is busy, out of reach or switched off, then the next number is dialed. Thus, the phone numbers are dialed one after the other sequentially until any of the staff is reached. At the same time, microcontroller will send data to the computer.

Depend on the data received from the microcontroller; the computer outputs the required message through the GSM modem to intimate the person about the patients' status with some appropriate message delivery. After receiving the message regarding the Patient status of the device, he/she will send a code to the microcontroller to take the necessary action. Hence controlling any furthers hazards.

2.2 HEART BEAT SENSOR

This block is used to sense the heart beat with the help of an LED and an LDR. A continuous light from the LED should fall on the LDR and the finger of the patient is to be placed in between the LED and LDR.

The slight variation in the skin due to the heart beat is read by the LDR. The LDR output is fed to an operational amplifier to the digital level (0 and 5) which is then fed in to the microcontroller.

A person's heartbeat is the sound of the valves in his/her's heart contracting or expanding as they force blood from one region to another. The number of times the heart beats per minute (BPM), is the heartbeat rate and the beat of the heart that can be felt in any artery that lies close to the skin is the pulse.

2.3 ARUDINO MICROCONTROLLER

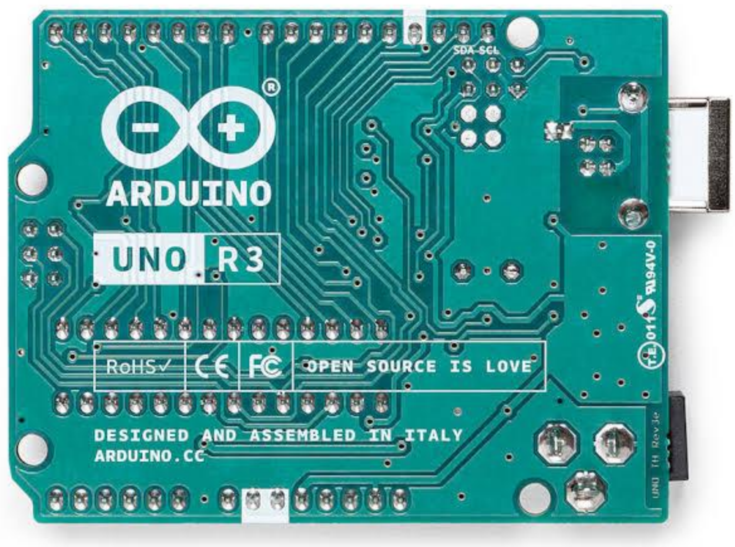

Fig 2.2 Arudino microcontroller

Arduino is an open-source electronics platform that has gained significant popularity among hobbyists, students, and professionals alike. It consists of both hardware and software components, designed to provide a simple and accessible way to create interactive projects.

At its core, Arduino boards are microcontrollers, small programmable devices that can be connected to various sensors, actuators, and other electronic components. These boards come in different shapes and sizes, but they all share a common feature: the ability to run programs written in the Arduino programming language, which is based on C/C++.

One of the key advantages of Arduino is its user-friendly nature. The Arduino software (IDE) provides a simplified programming interface, making it easier for beginners to get started with coding and electronics. It offers a range of built-in functions and libraries that abstract the complexities of low-level hardware programming, allowing users to focus on their project's logic and functionality.

Arduino boards are highly versatile and can be used for a wide range of applications. Whether you want to build a robot, control lights, create interactive art installations, or even develop prototypes for more complex systems, Arduino provides a flexible platform to bring your ideas to life. Its extensive community support ensures that you can find plenty of tutorials, examples, and projects to learn from and build upon.

Moreover, Arduino is known for its compatibility and expandability. The boards can communicate with other devices through various communication protocols such as UART, I2C, and SPI. Additionally, Arduino shields, which are add-on boards, extend the capabilities of the basic Arduino board by adding functionalities like Wi-Fi, Bluetooth, motor control, and more.

The open-source nature of Arduino fosters collaboration and innovation. Users can share their projects, libraries, and code with the community, contributing to the continuous growth and improvement of the platform. This collaborative environment has led to the development of numerous Arduino-based projects and initiatives worldwide.In summary, Arduino is a versatile and accessible electronics platform that empowers individuals to turn their ideas into reality. Its combination of hardware, software, and a supportive community makes it an excellent choice for anyone interested in learning about electronics, programming, and building interactive projects.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy- to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems.

Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

2.4 GSM

This GSM-Modem should be a plug and play GSM 900 / GSM 1800 / GSM 1900 modem. A direct and easy integration with RS232 and with in voltage range for the power supply.

A SIMCOM's GSM/GPRS modem is suitable for the GSM-SMS Transceiver System.

The features of SIMCOM-Modem:

- Triband GSM GPRS modem(EGSM 900/1800/1900 MHz) Designed for GPRS, data, fax, SMS and voice applications
- GPRS multi-slot class 10
- GPRS mobile station class B
- Designed for GPRS, data, fax, SMS and voice applications
- Fully compliant with GSM Phase 2/2+ specifications
- Built-in TCP/IP Protocol
- Built-in RTC in the module
- AT Command based.

CHAPTER 3

SYSTEM ELEMENTS DESCRIPTION

3.1 MICROCONTROLLER

The 89S52 Micro-controller is heart of this project. It is the chip that processes the User Data and executes the same. The software inherited in this chip manipulates the data and sends the result for visual display.The AT89S52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash Programmable and Erasable Read Only Memory (EPROM).

The device is manufactured using Atmel's high-density non-volatile memory technology and is compatible with the industry standard MCS-51TM instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel's AT89S52 is a powerful microcomputer 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, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry.

In addition, the AT89852 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 contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

FEATURES

- 8K Bytes of In-System Reprogrammable Flash Memory Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

3.2 ADC 0809

The ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a Successive Approximation register.

The 8-channel multiplexer can directly access any of 8-single ended. The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications.

3.2.1 Features:

- with 5 Easy interface to all microprocessors
- Operates ratio metrically or VDC or analog span adjusted
- voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- OV to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package
- Resolution : 8 Bits
- Total Unadjusted Error: +1/2 LSB and +1 LSB
- Single Supply: 5 VDC
- Low Power: 15 mW
- Conversion Time: 100 us

3.2.2 Functional Description

Multiplexer:

The device contains an 8-channel single-ended Analog signal multiplexer. A particular input channel is selected by using the address decoder. Table 1shows the input states for the address lines to select any channel. The address latched into the decoder on the low-to-high transition of the address latch enable signal. Converter:

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures.

The converter is partitioned into 3 major sections:

- 256R ladde r network
- Successive approximation register
- Comparator

256R Ladder Network:

The 256R ladder network approach is chosen so that the digital codes do not go missing (monotonic). In addition to this 256R network does not cause load variations on the reference voltage. The bottom resistor and the top resistor of the ladder network are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occur when the analog signal has reached +12 LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

Successive Approximation Register:

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

Comparator:

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements. The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset error.

3.3 LIQUID CRYSTAL DISPLAY

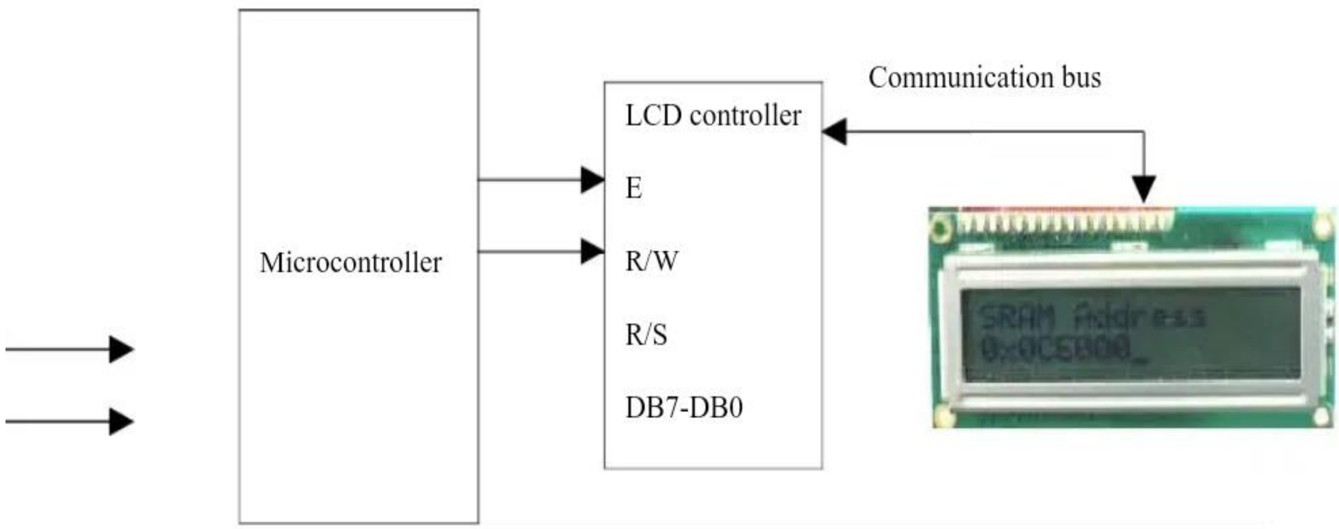

Fig 3.1 Schematic diagram of an LCD

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

The LCD controller provides a relatively simple interface between a processor and an LCD. LCDs can be added quite easily to an application and use as few as three digital output pins for control.There are different types of LCDs such as reflective LCD, absorption LCD, dot matrix LCD.

Each type of LCD is able to display multiple characters. In addition, each character may be displayed in normal or inverted fashion. The LCD may permit a character to be blinking or may permit display of a cursor indicating the" current" character. Such functionality would be difficult to be implemented using software.

Thus, an LCD controller is used to provide a simple interface to an LCD, perhaps eight data inputs and one enable input. This byte may be a control word, which can be an instruction or data word. The most common connector used for the 44780 based LCDs is 14 pins in a row, with pin centers 0.100" apart.

3.4 I²C BUS

F-C stands for inter-integrated circuit. This was designed by Philips but now a number of semiconductor device manufacturers are making devices compatible with FC bus. This FC bus is used mainly with single-chip micro controller based systems that require general-purpose circuits like EEPROM, RAM, real time clock, LCD & audio/video tuning circuits. A key advantage of this is that only two lines can connect multiple devices. All devices have built-in addresses.

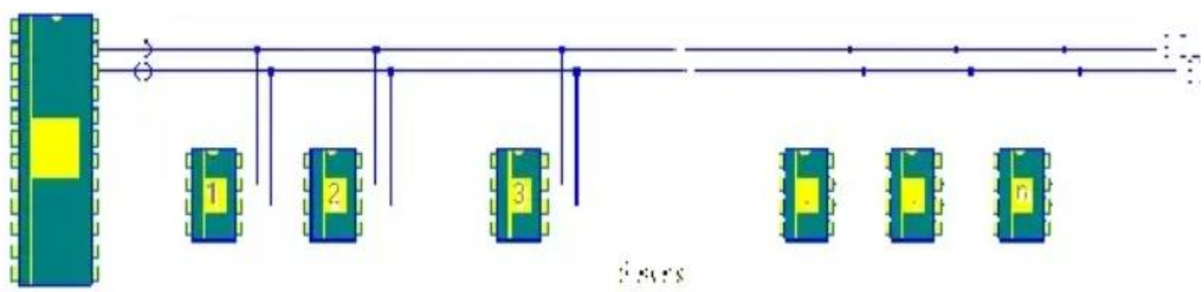

Fig 3.2 Shows a master connected to various device using 12C

In circuit with multiple devices, one device (usually the micro controller) takes the role of master. Another device (only one of the multiple devices at any one time) acts as a slave device. Master device takes control of SCL; i.e. SCL is set low or high under the control of master (usually micro controller). Slave device accepts the data from micro controller (e.g. writing into memory) or sends the data to micro

controller (reading from memory) under the control of master device.
Four different conditions exist in FC bus transfers. These are START, STOP, BIT TRANSFER (read/write) & ACKNOWLEDGE.

Normal data bit write/read

During transfer of data from master (micro controller) to slave device (e.g. EEPROM), SDA is set to logic'0' or logic' l' only when SCL is low. After small delay, SCL is pulsed high to clock in data. During read operation, SDA is an input line & its logic state is clocked in with SCL going high. Master (micro controller) can then read the level of SDA.

Start condition

Start is a special condition where SDA changes its state from high to low when SCL is high. Both SCL & SDA are controlled by master (micro controller).

Stop condition

Stop is also a special condition where SDA goes from low to high when SCL is high. Both SCL & SDA are controlled by master (micro controller).

Acknowledge

After transmitting eight data bits from micro controller to the device (e.g. EEPROM), direction of SDA line is reversed. One more clock pulse is given by micro controller. During this period, the slave device sets SDA to low. This indicates the acceptance of data by receiving device (e.g. EEPROM). When data is read from slave device (e.g. EEPROM), after reading eight bits, direction of SDA is reversed. SDA is set low (to send acknowledge) or high & then SCL is pulsed.

How to write a byte in EEPROM?

Typical example of writing into memory (device address A0 hex) location 58 (hex) with data byte 30 (hex) is given below:

S10100000A01011000A00110000AP

Where.

S-Start

P-Stop

A-Acknowledge

1 & 0-Data bits

Description of the above sequence is as follows:

- Action is started with start condition generated by master (micro controller).
- 1010 0000 (A0h) is transmitted as address for EEPROM.
- EEPROM responds with acknowledge during next clock pulse..
- 0101 1000 (58h) is transmitted as byte address with EEPROM.
- Finally data byte 0011 0000 (30h) is transmitted.
- EEPROM acknowledges during next clock pulse.
- At the end, master (micro controller) generates stop condition.

3.5 EEPROM

The Microchip Technology Inc. 24LC04B/08B is a 4Kbit or 8 bit electrically erasable PROM (EEPROM). The device is organized as two or four blocks of 256- x 8-bit memory with a 2-wire serial interface. Low voltage design permits operation down to 2.5 volts with typical standby and active currents of only 5 mA and 1 mA respectively. The 24LC04B/08B also has a page-write capability for up to 16 bytes of data. The 24LC04B/08B is available in the standard 8-pin DIP, 8-lead surface mount SOIC, MSOP and TSSOP package.

Features

- Single supply with operation down to 2.5V
- Low power CMOS technology
- 1 mA active current typical
- 10 micro ampere standby current typical at 5.5V
- 5 micro ampere standby current typical at 3.0V
- Organized as two or four blocks of 256 bytes
- 2-wire serial interface bus, 12CTM compatible
- Schmitt trigger, filtered inputs for noise Suppression Output slope control to eliminate ground bounce
- 100 kHz (E-temp) and 400 kHz (C/I-temp.) Compatibility
- Self-timed write cycle (including auto-erase)

Functional Description

The 24LC04B/08B supports a bi-directional 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as transmitter and if receiving data, as receiver. The bus has to be controlled by a master device which generates the serial clock (SCL), controls the bus access and generates the START and STOP conditions, while the 24LC04B/08B works as slave. Both master and slave can operate as transmitter or receiver, but the master device determines which mode is activated.

Bus Characteristics

The bus protocols are defined as:

- Data transfer may be initiated only when the bus is not busy
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted as a START or STOP condition defined.

Bus Conditions:

Bus not Busy (A)

Both data and clock lines remain HIGH.

Start Data Transfer (B)

A HIGH to LOW transition of the SDA line while the clock (SCL) is HIGH determines a START condition. condition. All commands must be preceded by a START.

Stop Data Transfer (C)

A LOW to HIGH transition of the SDA line while the clock (SCL) is HIGH determines a STOP condition. All operations must be ended with a STOP condition.

Data Valid (D)

The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data. Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of the data bytes transferred between the START and STOP conditions are determined by the master device.

Acknowledge

Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit. The device that acknowledges, has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Device Addressing

A control byte is the first byte received following the start condition from the master device. The control byte consists of a 4-bit control code, for the 24LC04B/08B this is set as 1010 binary for read and writes operations. The next three bits of the control byte are the block select bits (B2, B1, and B0). B2 is a 'don't care' for both the 24LC04B and 24LC08B; B1 is a 'don't care' for the 24LC04B. The master device to select which of the two uses them or four 256-word blocks of memory are to be accessed. These bits are in effect the most significant bits of the word address. The last bit of the control byte defines the operation to be performed. When set to '1', a read operation is selected and when set to '0', a write operation is selected. Following the start condition, the 24LC04B/08B monitors the SDA bus checking the device type identifier being transmitted. Upon a 1010 code, the slave device outputs an acknowledge signal on the SDA line. Depending on the state of the R/W bit, the 24LC04B/08B will select a read or write operation.

3.6 GSM MODEM

GSM is a most demanding system with the full range of digital techniques, via, equalization, frequency, hopping, sophisticated speech coding, error correction coding, echo cancellation block interleaving and advanced modulation provided to maximize the performance. The degree of processing is such that the battery current drain of the integrated circuits in the mobile is comparable with the current required to provide the RF power for the transmitter.

The GSM air interface provides the physical link between the mobile and the network. GSM is a digital system employing time division multiple access (TDMA) technique and operates at 900 MHz. The CEPT(conference of European post and telecommunication) has made available two frequency bands in the GSM system : (1) 890 MHz for the mobile to base station (up link), and (2) 985 MHz to 960 MHz for the base station to mobile (down link). These 25 MHz bands are divided into 124 pairs of carriers spaced by 200 MHz. Each of the carriers is divided into 8 TDMA time slots of 0.577 m sec length, such that the frame length is 4.615 m sec. the recurrence of each time slot makes up one physical channel, such that each carrier can support eight physical channels, both in up link & down link directions.

Features of GSM

The primary objective of GSM is to provide a full roaming mobile telephony service. Three broad categories of service provided by GSM are

- Tele services
- Bearer services
- Supplementary services

Tele services

Tele services are the services that are provided on a user terminal basis. Paramount tele services include voice communication and facsimile transmission.

Bearer Services

For Bearer services, the terminal equipment is provided by the user, the responsibility of the network service provide ending at the point of Connection. Data rates between 300 and 9600 bps fall into this category.

Supplementary Services

Supplementary services will be developed along the lines of ISDN services but will vary from country to country. GSM uses ISO and OSI model.

Working Principle

When you dial a number on the keypad of the phone, the handset transmits the digits, through the built-in radio transceiver, to a nearby cell. A group of transceiver station is controlled by a base station controller, which in turn is connected to a mobile switching centre. The MSC, in turn, is linked to other cellular and fixed line networks.All the switching functions within a GSM network are handled by MSC, which is the intelligence of the network and performs function like call routing, cell control, switching, plus all accounting and charging activities.

Once a cell is forwarded MSC, it determines how to route the call and set up the required link to enable the conversation. If the call is destined for a fixed the MSC sends it to DoT's public telephone exchange, over a leased line, which then switches the call to the desired telephone. However, if the call is destined for another mobile phone, things become more complicated. First, the MSC has to figure out where the desired mobile phone is, and then forward the call to the cell which is nearest to it.But how does the MSC figure out where a particular cell phone is? It is assisted by cellphone in this task. When the handset is powered on, it initializes itself and scans the control channels. These control channels are special RF used by the cell transmitter to send and receive control data. Based on the strength of received signal, the handset assigns itself to a specific cell. In this process it informs the cell of its location so that it can be passed out the handset keeps monitoring the data which is sent on the control channel till its own ID is paged-and then puts itself into receiving mode.However, if hand set is mobile, that is,if the user is travelling by car while calling, the cellular system also needs to keep track of the phone and the call is progress, so that it can automatically switch the call to another cell as the caller moves from one area to another. This process of switching calls between cells is user transparent, and is called cell hand off.

Strength of GSM

- GSM is the first to apply the TDMA scheme developed for mobile radio systems. It has several distinguishing features,
- Roaming in European countries.
- Connection to ISDN through RS box.
- Use of SIM cards.
- Control of transmission power.
- Frequency hopping.
- Discontinuous transmission.

GSM COMMANDS

This document describes the AT-command based messages exchanged between an application and the products in order to manage GSM related events or services. To check the modem whether it is functioned or not by using command, AT cmd - Enter the AT in hyper terminal or in serial port, the modem will responses as OK.

To check whether the SIM is there or not there in the modem, AT+CPIN? -> Modem responses as "CPIN READY" or "+CME ERROR" Some commands are

Call Control commands:

The ATD command is used to set a voice, data or fax call.

ATD<nb>; enter where nb is the destination phone number

Ex: ATD+919845598455; -> Voice call

Modem Responses -> OK or BUSY or NO ANSWER or NO CARRIER

Hang-Up Command:

The ATH command is used by the application to disconnect the remote user.

ATH -> modem responses as OK,

Answer a call:

ATA -> to answering the incoming call, modem responses as OK.

Redial last telephone number:

ATDL-> Redial the last number for ex, last number is 9845598455, then modem will dial the number automatically.

Automatic answer ATSO

This ATSO parameter determines and controls the product automatic answering mode.

CHAPTER: 4

SOFTWARE TOOLS

INTRODUCTION TO EMBEDDED SYSEMS

Embedded systems have grown tremendously in recent years, not only in their popularity but also in their complexity. Gadgets are increasingly becoming intelligent and autonomous. Refrigerators, air-conditioners, automobiles, mobile phones etc are some of the common examples of devices with built in intelligence. These devices function based on operating and environmental parameters.

The intelligence of smart devices resides in embedded systems. An embedded system, in general, in co-operates hardware, operating systems, low-level software binding the operating system and peripheral devices, and communication software to enable the device to perform the pre-defined functions. An embedded system performs a single, well-defined task, is tightly constrained, is reactive and computes results in real time.

Let us take a detailed look at these features of embedded systems:

Single functioned: An embedded system executes a specific program repeatedly. For example, a pager is always a pager. In contrast a desktop system executes a variety of programs like spreadsheets, word processors, etc. However there are exceptions where in an embedded system's program is updated with newer program versions. Cell phones are examples of being updated in such a manner.

Tightly constrained: All computing systems have constraints on design metrics but those on embedded systems can be especially tight. A design metric is a measure of an implementation's features, such as cost, size performance and power.

Reactive and real time: Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without delay.

4.2 Embedded Hardware

All embedded systems need a microprocessor, and the kinds of microprocessors used in them are quite varied. A list of some of the common microprocessor families is the ZILOG Z8 family, Intel 805/80188/x 86 families, Motorola 68k family and the PowerPC family.

4.3 Embedded Software

The software for the embedded systems is called firmware. The firmware will be written in assembly languages for time or resource critical operations or using higher- level languages like C or embedded C. The software will be simulated using micro code simulators for the target processor. Since they are supposed to perform only specific tasks these programs are stored in Read Only Memories (ROM's).

4.4 Application areas for embedded systems

Embedded software is present in almost every electronic device you use today. There is embedded software inside your watch, cellular phone, automobile, thermostats, Industrial control equipment and scientific and medical equipment. Defence services use embedded software to guide missiles and detect aircraft's. Communication satellites, medical instruments and deep space probes would have been nearly impossible without these systems. Embedded systems cover such as broad range of products that generalization is difficult.

Here are some broad categories.

- Aerospace and Defense Electronics (ADE)
- Consumer/Internet applications
- Data Communications
- Digital imaging
- Medical electronic Mobile data infrastructures

EMBEDDED C

Using Embedded C with the 8051: Why Use a High Level Language?

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

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

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

By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have C compilers available, which support ANSI C. All of this is not to say that assembly language does not have its place. In fact, many embedded systems (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only way to go. One of the great things about the C language is that it allows you to perform low-level manipulations of the hardware if need be, yet provides you with the functionality and abstraction of a higher order language.

KEIL C(EMBEDDED C) VS ANSI C:

The Keil compiler provides the user with a superset of ANSI C with a few key differences. For the most part these differences allow the user to take advantage of the architecture of the 8051. Additional differences are due to limitations of the 8051.

Data Types & SFR'S

Keil C has all the standard data types available in ANSI C plus a couple of specific data types, which help, maximize the use of the 8051's architecture. The following table shows the standard data types and the number of bytes they take on the 8051. It should be noted that integers and longs are stored with the most significant byte in the lower address (MSB first).

In addition to these standard data types the compiler supports a bit data type. A variable of type 'bit' is allocated from the bit addressable segment of internal RAM and can have a value of either one or zero. Bit scalars can be operated on in a manner similar to other data types. Their type is promoted for operations with higher data types such as char or int. Arrays of bits and pointers to bit variables are not allowed.

The special function registers of the 8051 are declared using the type specifier 'sfr' for an eight-bit register or 'sfr16' for a 16-bit register such as DPTR. In these declarations, the name and the address of the SFR are provided in the code. The address must be greater than 80 hex. Bits of the bit addressable SFRS can be declared by using the 'sbit' type. This type cannot be applied to any SFR, which is not normally bit addressable.

Downsizing Your Variables

On an eight-bit machine like the 8051, wide use of data types whose size is greater than eight bits will be a large waste of processing power and memory. Obviously, the most preferred type for variables will be unsigned char since it only uses one byte.

Use Unsigned Types

The reasoning behind this is that the 8051 does not support signed arithmetic, and the extra code required by a signed value as opposed to an unsigned value will take away from your overall processor resources.

Stay Away from Floating Point

Floating-point applications take a lot of time to execute. One is better off to avoid performing floating-point calculations.

Make Use of bit Variables

When you are using flags, which will only contain a one or a zero, use the bit type instead of an unsigned char. This will help make your memory reserves go farther, since you will not be wasting seven bits. Additionally, bit variables are always in internal RAM and therefore will be accessed in one cycle.

Use Locals instead of Globals

Variables, which are declared to be global data, will be less efficient than use of local variables. The reason for this is that the compiler will attempt to assign any local variables to internal registers. In comparison, global data may or may not be in internal data depending on your declaration. In cases where the global are assigned to XDATA by default (such as large memory model programs) you have given up speedy access. Globals and locals can be forced into any memory area. We can optimize the speed of the program by placing the most frequently accessed variables in internal RAM. Additionally, code will be smaller since it takes less instructions and setup to access variables in internal RAM than it does to access variables in external RAM.

Use Macros Instead of Functions

In addition to using intrinsics, you can also make your code more readable by implementing small operations such as a read from a latch or code to enable a certain circuit as macros. Instead of duplicating one or two lines of code all over the place, you can isolate duplicate code into a macro, which will then look like a function call. The compiler will place the code inline when it emits object code and will not generate a call. The code will be much easier to read and maintain since the macro can be given a name, which describes its operation.

EXPRESS PCB

Express PCB is a very easy to use Windows application for laying out printed circuit boards. This software is used for PCB designing.

Beginning a New Layout

- Begin a new layout by running Express PCB. Open a new file. If you are designing a four-layer board, select Board properties .

- In the main window, the yellow rectangle defines the perimeter of the PC board. Set The size of your board by moving three of its four corners (the upper left corner is fixed at (0, 0). Move the corners by dragging them with the mouse, or by double-clicking them and entering coordinates. Additional corners can be added to the perimeter to change its shape.

- Save the file.

- The easiest way to place components in your board layout is to use the Component Manager under view or click the button located on the top toolbar to display the Component Manager.

- Select one of these categories:

- Library components - Components that are included with the program.

- Custom components - Components that you have drawn.

- Favorite components - Components or symbols that you have book-marked.

- orientation (rotated up, left, down, or right).

- Left click to place the component.

- Filled Planes are used to add ground or power planes to a circuit, usually on double-sided boards.

- Corners in traces allow them to bend. They are displayed as small square blocks at the ends of traces.

- ExpressPCB uses standard Edit commands for Copy, Cut and Paste. To copy

- An item or several items, first select them.

- Labelling should be done on the silkscreen layer to avoid shorting of components.

- Then finally there is the print option to get prints of the desired layers. Thus the Express PCB is an extremely useful tool for realizing PCB's.

OVERVIEW ON KIEL MICROVISION 2

Keil development tools for the 8051 Microcontroller Architecture support every level of software developer from the professional applications engineer to the student just learning about embedded software development.

The industry-standard Keil C Compilers, Macro Assemblers, Debuggers, Real- time Kernels, Single-board Computers, and Emulators support all 8051 derivatives and help you get your projects completed on schedule.The Keil 8051 Development Tools are designed to solve the complex problems facing embedded software developers.

When starting a new project, simply select the microcontroller you use from the Device Database and the uVision IDE sets all compiler, assembler, linker, and memory options for you.The Keil uVision Debugger accurately simulates on-chip peripherals (I²C, CAN, UART, SPI, Interrupts, I/O Ports, A/D Converter, D/A Converter, and PWM Modules) of your 8051 device. Simulation helps you understand hardware configurations and avoids time wasted on setup problems. Additionally, with simulation, you can write and test applications before target hardware is available.

When you are ready to begin testing your software application with target hardware, use the MON51, MON390, MONADI, or FlashMON51 Target Monitors, the ISD51 In-System Debugger, or the ULINK USB-JTAG Adapter to download and test program code on your target system.The Vision IDE from Keil, combines project management, make facilities, source code editing, program debugging, and complete simulation in one powerful environment. uVision helps you get programs working faster than ever while providing an easy-to-use development platform. The editor and debugger are integrated into a single application and provide a seamless embedded project development environment.

The Vision2 IDE is Windows-based software development platforms that combines a robust editor, project manager, and make facility. µVision2 integrates all tools including the C compiler, macro assembler, linker/locator, and HEX file generator. Vision2 helps expedite the development process of your embedded applications by providing the following features: The 'device database' which automatically sets the assembler, compiler, and linker options for the chip you selects. This prevents you from wasting your time configuring the tools and helps you get started writing code faster.

A robust project manager', which lets you create several different configurations of your target from a single project file. Only the Keil µVision IDE allows you to create an output file for simulating, an output file for debugging with an emulator, and an output file for programming an EPROM--all from the same project file.

CHAPTER 5

SCOPE FOR FUTURE ENHANCEMENT

In the current version of the system, you can place parameters (heartbeats) on the web. However, there is still room for improvement. Other parameters such as blood pressure, blood sugar. BMI (body mass index), and waist circumference can also be measured. The patient's complete health status is then recorded and available on the internet. This makes it much easier for doctors to monitor the progress of a patient's health and give advice on the patient's health. You can extend your system to include additional features such as: B. Links to emergency services, a list of major doctors and their areas of expertise, hospitals and their special facilities, and more. Physicians can create awareness of illnesses and their symptoms via mobile applications. From the results obtained from the evaluation and analysis, the system is better for patients and physicians to improve their medical evaluation.

- Enables parallel development and aims at building platform for real time remote health monitoring Examples are Mobihealth, Telemedicare

- Interfacing the GPS system enables the global tracking of the patient

- It lays platform for development of advance medical equipments .

CHAPTER 6

RESULT

- The Heartbeat Monitoring System is the part of Patient Monitoring System, can be extended to measure other parameters of patien like ECG & temperature etc.

- Temperature is measured using DS1820, which gives temperature to current conversions in 200ms and it does not require any external circuitry. Hence it is simple and easy to measure temperature. This measured temperature is displayed over an LCD display via microcontroller.

- Heart Beat is measured by passing an high intensity red light through a finger which is collected by LDR, amplified and displayed over an LCD display via microcontroller.

- These two measured parameters are encoded and transmitted via GSM Modem. On receiving the readings and current status of patient, the doctor can take necessary actions or suggest his sub-ordinates for the same.

REFERENCE

1. Arulananth.T.S, Shilpa.B, "Fingertip based heart beat monitoring system using embedded systems", 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) Volume. 2, 2017.

2. Dr.Gurjar.AA, Neha.A.Sarnaik, "Heart Attack Detection By Heartbeat Sensing using Internet of Things". International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 03 Mar-2018, 2018.

3. Heart beat Sensing and Heart Attack Detection Using internet of things: 10T Aboobacker sidheeque, Arith Kumar, K. Sathish, (IJESCE) International Journal Of Engineering Science and Computing, Volume 09, Issue 07 July 2020.

4. Kumar M.A and SekharY.R "Android Based Health Care Monitoring System," 2nd International Conference On Innovations in Information Embedded and Communication Systems, ICHECS, IEEE,2016.

5. Lei Song, Yongcai Wang, Ji-Jiang Yang, Jianqiang Li, "Health Sensing by Wearable Sensors and Mobile Phones", IEEE 16th International Conference on e- Health Networking, Oct. 2014.

6. Monitoring using Wireless Body Area Network Mohammad Wajih Alam, Tanin Sultana and Mohammad Sami Alam International Journal of Bio Science and Bio Technology Vol.8, No.1 (2016).

7. Ponugumatla Kalyan, Mr. Gouri Shankar Sharma, "10T Based Heart Function Monitoring and Heart Disease Prediction System", IJSART Volume 3, Issue 12 DECEMBER 2017.

8. Sahana S Khamitkar's "loT based System for Heart Rate Monitoring", International Journal of Engineering Research & Technology (IJERT) Vol. 9 Issue 07, July-2020.

9. Selvathi. D, Sankar, V. & Venkatasubramani. H, "Embedded based automatic heart attack detector and intimator" International Conference on Innovations in Information, Embedded and Communication Systems (ICHECS) Volume 7, Issue 2 2018.

10. Ufoarch S.U. Oranugo C.O, Uchechukwu M.E, Heartbeat Monitoring and Alert System Using GSM Technology, International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-9 Issue-7, May 2020.

11. Warsuzarina Mat Jubadi, Siti Faridatul Aisyah. "Heartbeat Monitoring Alert via SMS" 2009 IEEE Symposium on Industrial Electronics and Applications October 4-6, 2009, Kuala Lumpur, Malaysia

APPENDIX

```
#include<LiquidCrystal.h>
//LiquidCrystal lcd(8,9,10,11,12,13);
LiquidCrystal lcd(8,9,13,12,11,10);
int SEN1  = A0;
int SEN2  = A1;
int SEN3  = A2;
int SEN4  = A3;
String webdata="0000";
void setup()
{
lcd.begin(16,2);
Serial.begin(9600);
//pinMode(A0,INPUT);
//pinMode(A1,INPUT);
//pinMode(A2,INPUT);
//pinMode(A3,INPUT);
pinMode(2,OUTPUT); // FROM ONLINE
pinMode(3,OUTPUT);
pinMode(4,INPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT); // FROM SENSOR VALUE
pinMode(7,OUTPUT);
```

```
pinMode(A5,OUTPUT);
lcd.setCursor(0,0);
lcd.print("HEARTRATE SENSOR");
lcd.setCursor(0,1);
lcd.print("   MONITERING");
delay(3000);
lcd.clear();
}
void loop()
{
    //lcd.clear();
    lcd.setCursor(3,0);// column, row
    lcd.print("HAERTRATE SENSOR");
    lcd.setCursor(0,1);
    lcd.print("    ");
    int x = map(analogRead(SEN1),0,1024,50,100);
    String y="";
    lcd.setCursor(4,1);
    lcd.print(x);
    if(x > 80)
    {
    digitalWrite(2,HIGH);
    y=" High";
      lcd.setCursor(10,1);
```

```
     lcd.print("HIGH  ");
    }
    else
    {
     digitalWrite(2,LOW);
     y=" NORMAL ";
      lcd.setCursor(10,1);
      lcd.print("NORMAL ");
    }
    Serial.print("Field1=");
    Serial.print(x);
    Serial.print(y);
    Serial.print("&Field2=00");
Serial.println(" ");
    //delay(1000);
delay(2500);
 }
```