HEART BEAT MONITOR

“A LIFE SAVER”

A PROJECT REPORT

Submitted by

Gautham .B	1110107026
Jagan C Venkatesh	1110107037
Karnith Shasti Vishak.K	1110107040
N.Mohammed Insaf	1110107053

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2015

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report titled, “HEART BEAT MONITOR, A LIFE SAVER” is the bonafide work of “GAUTHAM.B, JAGAN C VENKATESH, KARNITH SHASTI VISHAK.K, MOHAMMED INSAF.N” who carried out the project work under my supervision.

SIGNATURE

Dr. M.Bharathi Ph.D.,
Associate Professor
Electronics & Communication
Kumaraguru College of Technology
Coimbatore.

SIGNATURE

Dr. Rajeswari Mariappan M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
Electronics & Communication
Kumaraguru College of Technology
Coimbatore.

The candidates with Register numbers 1110107026, 1110107037, 1110107040 and 1110107053 are examined by us in the project viva-voce examination held on …………………….

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First we would like to express our praise and gratitude to the Lord, who has showered his grace and blessing enabling us to complete this project in an excellent manner. He has made all things beautiful in his time.

We express our sincere thanks to our beloved Joint Correspondent, **Shri. Shankar Vanavarayar** for his kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr.R.S.Kumar** M.E., Ph.D., who encouraged us with his valuable thoughts.

We would like to express our sincere thanks and deep sense of gratitude to our HOD, **Dr. Rajeswari Mariappan** M.E., Ph.D., for her valuable suggestions and encouragement which paved way for the successful completion of the project.

We are greatly privileged to express our deep sense of gratitude to the Project Coordinator **Ms.S.Nagarathinam** M.E., Assistant Professor (SRG), for her continuous support throughout the course.

In particular, we wish to thank and express our everlasting gratitude to the Supervisor **Dr.M.Bharathi** Ph.D.,Associate Professor for her expert counselling in each and every steps of project work and we wish to convey our deep sense of gratitude to all teaching and non-teaching staff members of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

TABLE OF CONTENT

ABSTRACT

More than 2 million people are at high risk of facing heart-attack. This project helps by acting as a life saver in attending to these situations immediately. The pulse rate plays a key role in the project.

This project presents the design and development of a heart rate monitor using fingertip sensor. The device uses the optical technology to detect the flow of blood through the finger and offers the advantage of portability over tape-based recording systems. This project is measuring heartbeat using the pulse oximetry logic .If the measured heartbeat rate is below the critical threshold of heart-attack then a help signal can be sent to the hospital including the latitude and longitude position of the victim, So that the medical aid is given to victim automatically.

Evaluation of the device on real signals shows accuracy in heart rate estimation, even under intense physical activity. The performance of Heart Rate Measuring device was compared with signal represented on an oscilloscope and manual pulse measurement of heartbeat, giving excellent results. This proposed Heart Rate Measuring device is economical and user friendly.

LIST OF ABBREVATIONS

BSN	Body Sensor Network
PHI	Personal Health Information
LED	Light Emitting Diode
AC	Alternating Current
DC	Direct Current
RMS	Root Mean Square
ADC	Analog to Digital Converter
PPG	Photo Plethysmo Graphy
EN	Enable
BPM	Beats per Minute
RAM	Random Access Memory
SSP	Synchronous Serial Port
PSP	Parallel Slave Port
BOR	Brown out Time
VREF	Voltage Reference
EEPROM	Electrically Erasable Programmable Read Only Memory
ICSP	In-Circuit Serial Programming
WDT	Watchdog Timer
ICD	In-Circuit Debug

PC	Program Counter
SFR	Special Function Register
HCI	Host Controller Interface
GSM	Global System for Mobile communication
OSS	Operations support system
GPS	Global Positioning System
JVM	Java Virtual Machine
J2EE	Java 2 platform Enterprises Edition
JSP	Java Server Page
JDT	Java Development Tool

LIST OF FIGURES

1. INTRODUCTION

In our present day scenario, mobile Healthcare (m-Healthcare) system has been envisioned as an important application of pervasive computing to improve health care quality and save lives, where miniaturized wearable and implantable body sensor nodes and smart phones are utilized to provide remote healthcare monitoring to people who have chronic medical conditions such as diabetes and heart disease.

Specifically, in an m-Healthcare system, medical users are no longer needed to be monitored within home or hospital environments. Instead, after being equipped with Smartphone and wireless body sensor network (BSN) formed by body sensor nodes, medical users can walk outside and receive the high-quality healthcare monitoring from medical professionals anytime and anywhere. Each mobile medical user's personal health information (PHI) such as heart beat, blood sugar level, blood pressure and temperature and others, can be first collected by BSN, and then aggregated by Smartphone via Bluetooth. Finally, they are further transmitted to the remote healthcare Centre via 3G networks only when it exceeds the pulse rate threshold. Based on these collected PHI data, medical professionals at healthcare centre can continuously monitor medical users' health conditions and as well quickly react to users' life-threatening situations and save their lives by dispatching ambulance and medical personnel to an emergency location in a timely fashion.

Although m-Healthcare system can benefit medical users by providing high-quality pervasive healthcare monitoring, the flourish of m-Healthcare system still hinges upon how we fully understand and manage the challenges facing in m-Healthcare system, especially during a medical emergency. To clearly illustrate the challenges in m-Healthcare emergency, we consider the following scenario. In general, a medical user's PHI should be reported to the healthcare centre every 5 minutes for normal remote monitoring. However, when he has an emergency medical condition, for example, heart attack, his BSN becomes busy reading a variety of medical measures, such as heart rate, blood pressure, and as a result, a large amount of PHI data will be generated in a very short period of time. However, since Smartphone is not only used for healthcare monitoring, but also for other applications, i.e., phoning with friends, the smart phone's energy could be insufficient when an emergency takes place. Although this kind of unexpected event may happen with very low probability, the reliability of m-Healthcare system is still challenging in emergency.

Recently, opportunistic computing, as a new pervasive computing paradigm, has received much attention. Essentially, opportunistic computing is characterized by exploiting all available computing resources in an opportunistic environment to provide a platform for the distributed execution of a computing-intensive task. For example, once the execution of a task exceeds the energy and computing power available on a single node, no other opportunistically contacted nodes can contribute to the execution of the original task by running a subset of task, so that the original task can be reliably performed. Obviously, opportunistic computing paradigm can be applied in m-Healthcare emergency to resolve the challenging reliability issue in PHI process. However, PHI is personal information and very sensitive to medical users.

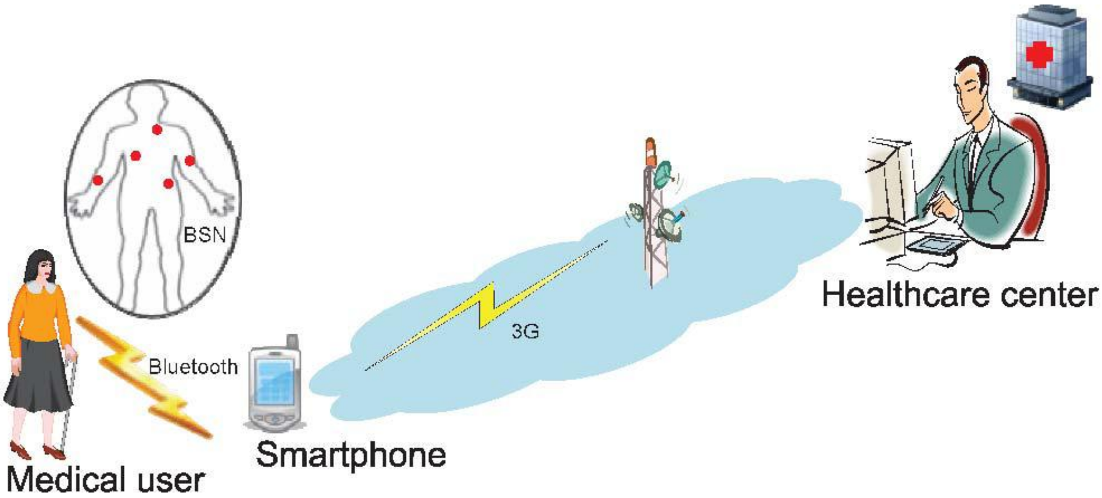


Fig 1.1 General diagram of heart-beat monitoring system

Fig 1.1 shows the overview of how the heart-beat monitoring system works. The pulse rate of patient is detected and transmitted to Smartphone using Bluetooth module. Then using the GSM network of mobile we transmit the pulse rate data to the healthcare centre only during an emergency.

2. HARDWARE DISCRIPTION

This section describes about the hardware modules like Bluetooth module, microcontroller, global positioning system, global system for mobile communication, temperature sensor and IR pulse sensor that has been used to determine the pulse rate of the medical patient and transmit it to the Smartphone for further processing.

2.1 BLOCK DIAGRAM OF HEART-BEAT MONITOR

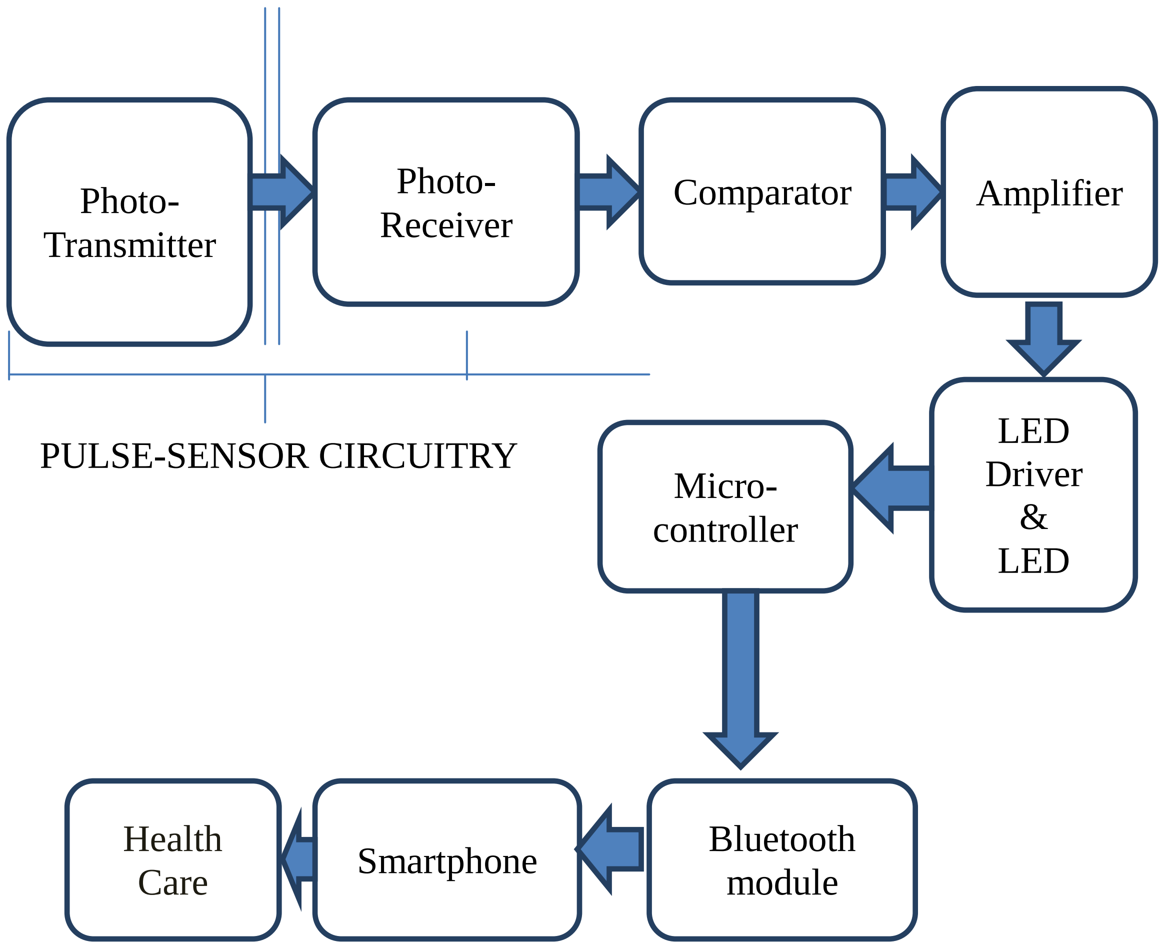


Fig 2.1 Block diagram of heart-beat monitor

Fig 2.1 shows the various modules used in determining the pulse rate and transmitting it to the healthcare centre. Here the pulse is determined using an IR sensor which consists of photo transmitter, photo receiver, comparator and amplifier. The count of the pulses can be seen visually using an LED and for processing purpose we use a microcontroller which keeps track of the number of counts. The values from the microcontroller are passed on to the Smartphone via Bluetooth module and when the pulse rate exceeds a certain threshold value it is sent to the health care and immediate medical aid is provided.

2.1.1 DATA FLOW DIAGRAM OF HEART-BEAT MONITORING SYSTEM

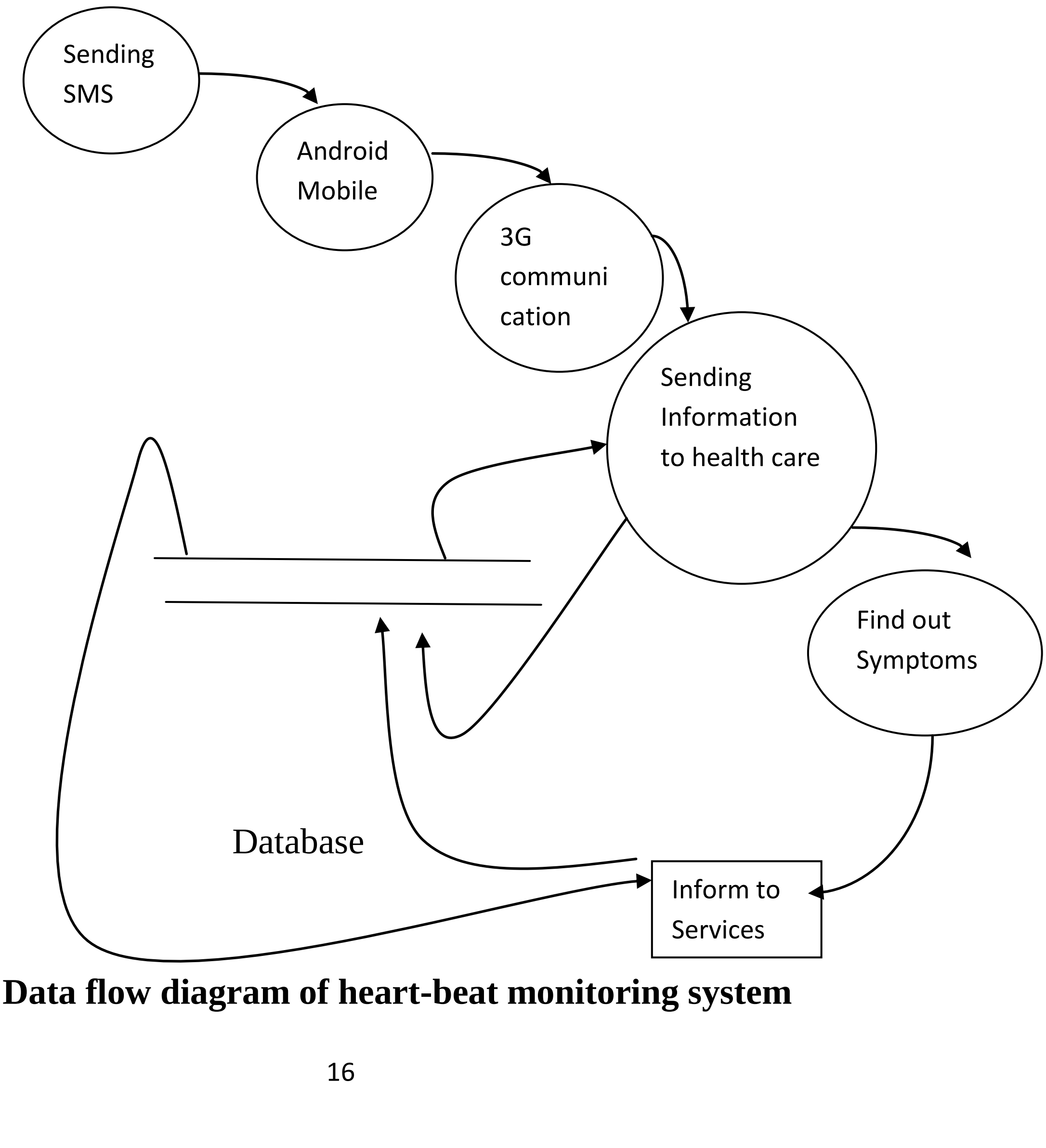


Fig 2.2 Data flow diagram of heart-beat monitoring system

Fig 2.2 shows the step by step process by which the determined pulse rate data is transmitted from BSN network to the Healthcare centre.

2.1.2 CLASS DIAGRAM

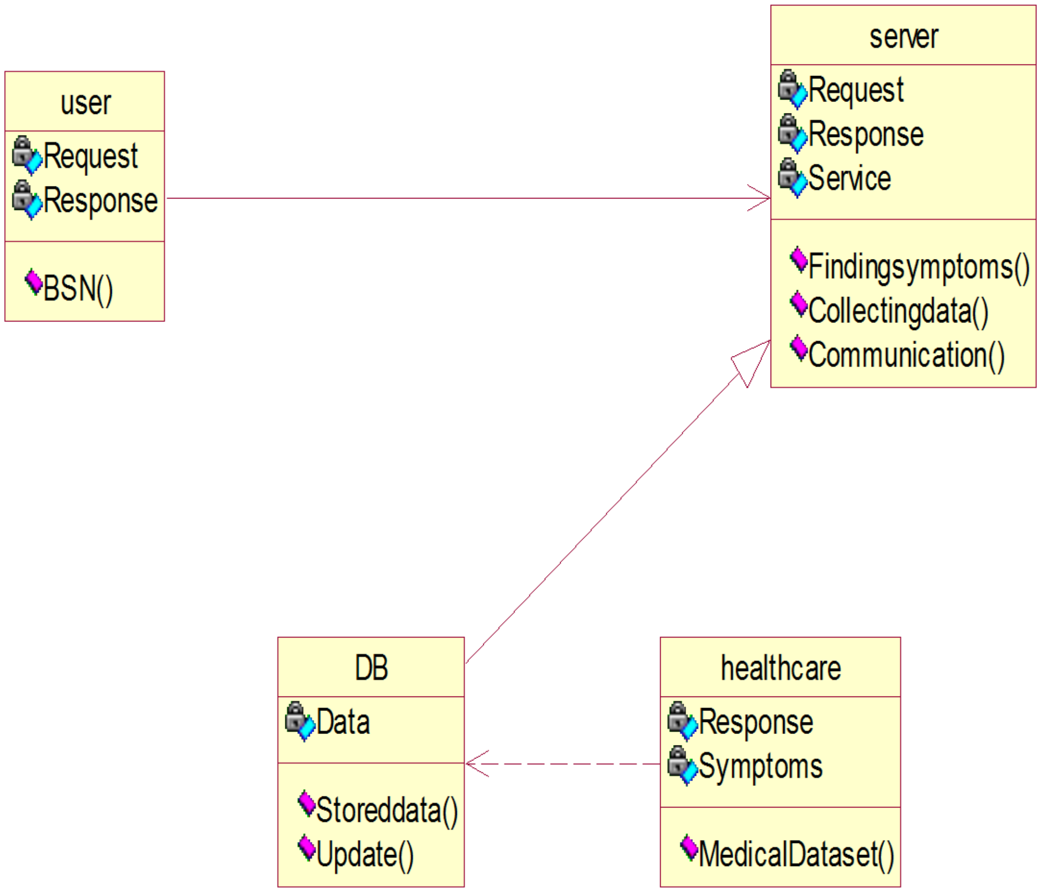


Fig 2.3 Client-server model of Heartbeat monitoring system

Fig 2.3 shows the client-server model which describes the various packages and modules used in Java script.

2.1.3 SEQUENCE:

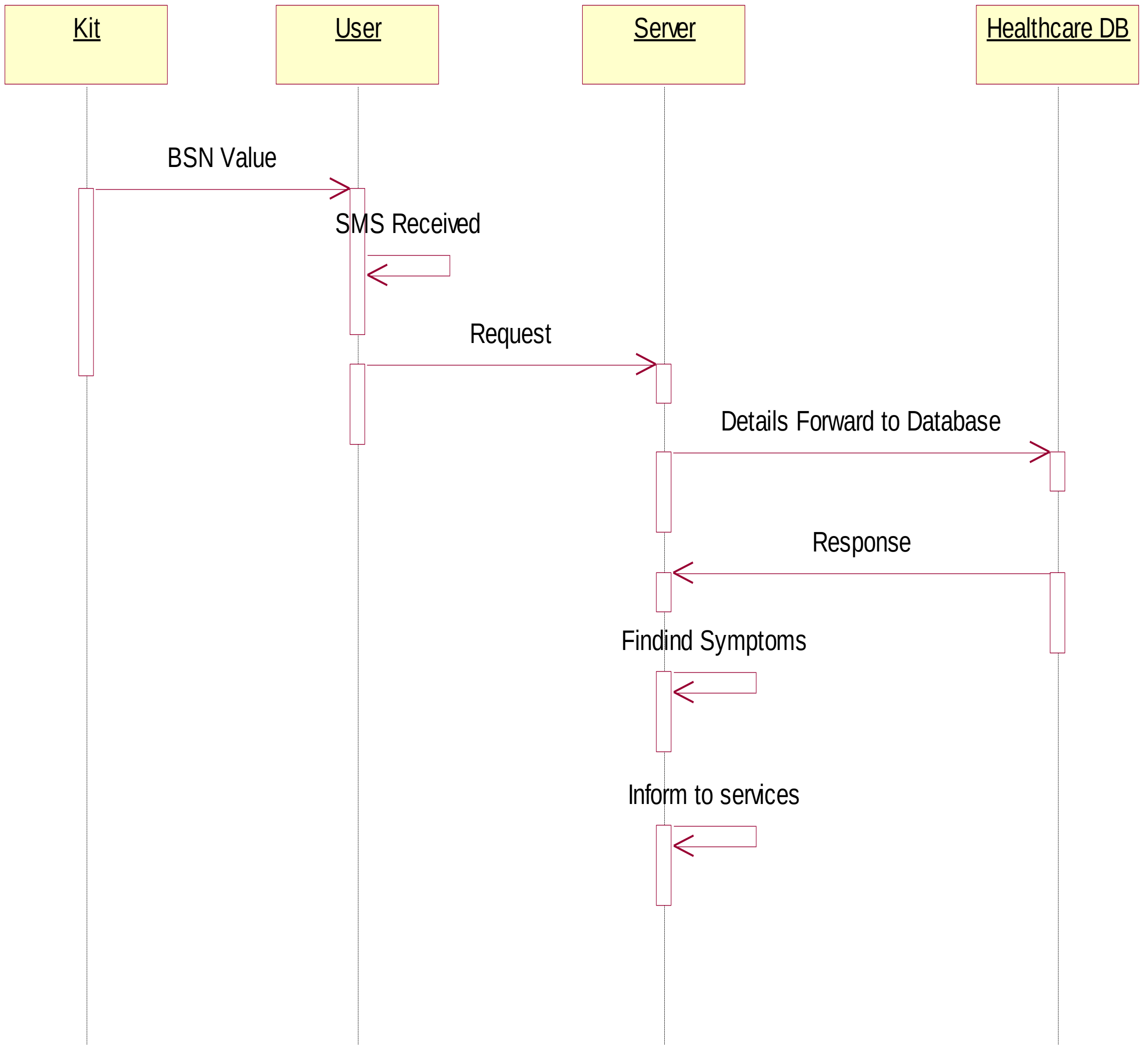


Fig 2.4 Communication sequence in heart-beat monitoring system

Fig 2.4 shows the communication sequence that takes place between BSN and Bluetooth module. This also gives the communication sequence between Bluetooth to the Server and server to the healthcare centre.

2.2 POWER SUPPLY UNIT

Circuit diagram

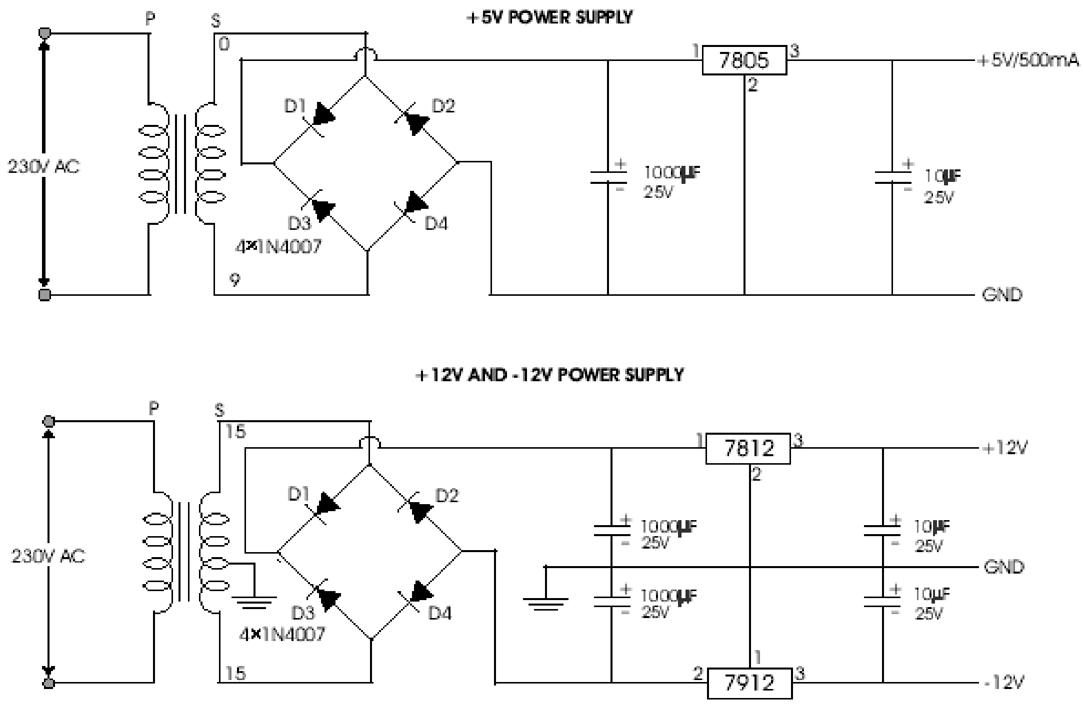


Fig 2.5 Circuit Diagram of Power Supply

Working principle

The AC voltage, typically 220V rms, is connected to a transformer, which steps that ac voltage down to the level of the desired DC output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.

A regulator circuit removes the ripples and also remains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes.

Block Diagram of Power supply

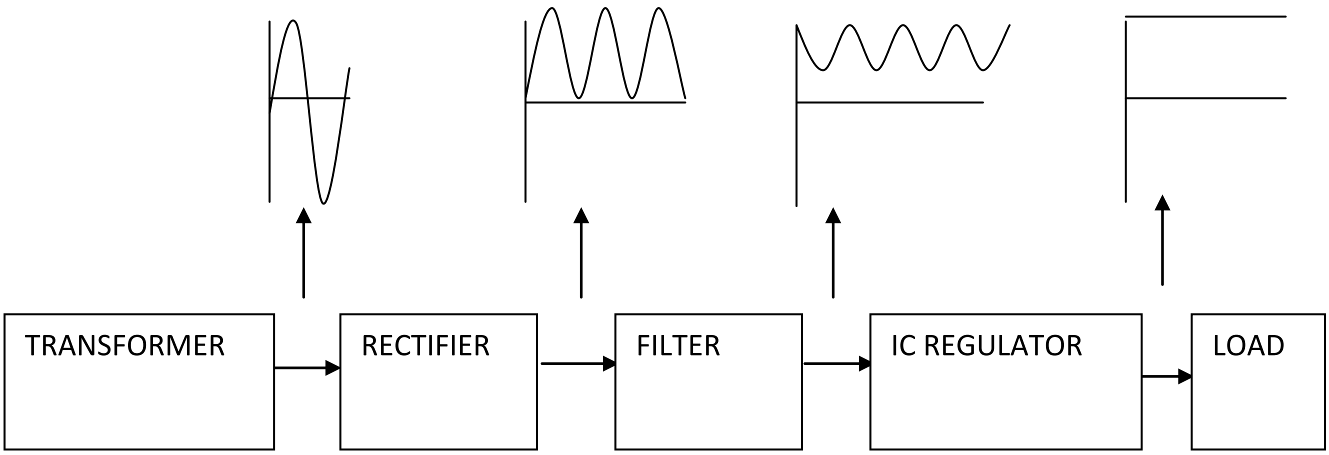


Fig 2.6 Block diagram of power supply

Transformer

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op–amp. The advantages of using precision rectifier are it will give peak voltage output as DC; rest of the circuits will give only RMS output.

Bridge rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.
The path for current flow is from point B through D1, up through RL, through D3, through the secondary of the transformer back to point B. this path is indicated by the solid arrows. Waveforms (1) and (2) can be observed across D1 and D3.

One-half cycle later the polarity across the secondary of the transformer reverse, forward biasing 2 and D4 and reverse biasing D1 and D3. Current flow will now be from point A through D4, up through RL, through D2, through the secondary of T1, and back to point A. This path is indicated by the broken arrows. Waveforms (3) and (4) can be observed across D2 and D4. The current flow through RL is always in the same direction. In flowing through RL this current develops a voltage corresponding to that shown waveform (5). Since current flows through the load (RL) during both half cycles of the applied voltage, this bridge rectifier is a full-wave rectifier.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit.

This may be shown by assigning values to some of the components shown in views A and B. Assume that the same transformer is used in both circuits. The peak voltage developed between points X and y is 1000 volts in both circuits. In the conventional full-wave circuit shown—in view A, the peak voltage from the Centre tap to either X or Y is 500 volts.

Since only one diode can conduct at any instant, the maximum voltage that can be rectified at any instant is 500 volts.

The maximum voltage that appears across the load resistor is nearly-but never exceeds-500 volts, as result of the small voltage drop across the diode. In the bridge rectifier shown in view B, the maximum voltage that can be rectified is the full secondary voltage, which is 1000 volts. Therefore, the peak output voltage across the load resistor is nearly 1000 volts. With both circuits using the same transformer, the bridge rectifier circuit produces a higher output voltage than the conventional full-wave rectifier circuit.

2.3 IC VOLTAGE REGULATORS

Voltage regulators [8] comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustable set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to tens of watts.

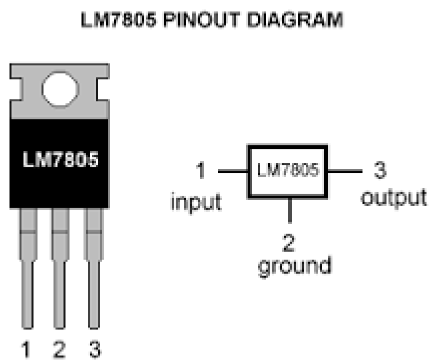


Fig 2.7 LM7805 pin-out diagram

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

2.4 IR PULSE SENSOR

The new version of Pulse sensor [4] uses reflective optical sensor for sensing heart beat. The use of pulse sensor simplifies the build process of the sensor part of the project as both the infrared light emitter diode and the detector are arranged side by side in a leaded package, thus blocking the surrounding ambient light, which could otherwise affect the sensor performance. It is designed such that it has a printed circuit board for it, which carries both sensor and signal conditioning unit. And its output is a digital pulse which is synchronous with the heart beat. The output pulse can be fed to either an ADC channel or a digital input pin of a microcontroller for further processing and retrieving.

The principle of photoplethysmography (PPG) is a non-invasive method of measuring the variation in blood volume in tissues using a light source and a detector. Since the change in blood volume is synchronous to the heart beat, this technique can be used to calculate the heart rate. Transmittance and reflectance are two basic types of photoplethysmography. For the transmittance PPG, a light source is emitted in to the tissue and a light detector is placed in the opposite side of the tissue to measure the resultant light. Because of the limited penetration depth of the light through organ tissue, the transmittance PPG is applicable to a restricted body part, such as the finger or the ear lobe. However, in the reflectance PPG, the light source and the light detector are both placed on the same side of a body part. The light is emitted into the tissue and the reflected light is measured by the detector. As the light doesn't have to penetrate the body, the reflectance PPG can be applied to any parts of human body. In either case, the detected light reflected from or transmitted through the body part will fluctuate according to the pulsatile blood flow caused by the beating of the heart.

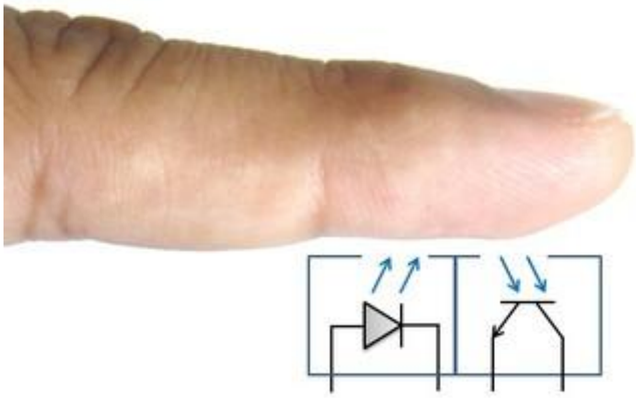


Fig 2.8 IR Pulse sensor

The following picture shows a basic reflectance PPG probe to extract the pulse signal from the fingertip. A subject’s finger is illuminated by an infrared light-emitting diode. More or less light is absorbed, depending on the tissue blood volume.

Consequently, the reflected light intensity varies with the pulsing of the blood with heart beat. A plot for this variation against time is referred to be a photoplethysmographic.

The PPG signal has two components, frequently referred to as AC and DC. The AC component is mainly caused by pulsatile changes in arterial blood volume, which is synchronous with the heart beat. So, the AC component can be used as a source of heart rate information. This AC component is superimposed onto a large DC component that relates to the tissues and to the average blood volume. The DC component must be removed to measure the AC waveform with a high signal-to-noise ratio. Since the useful AC signal is only a very small portion of the whole signal, an effective amplification circuit is also required to extract information.

The output (VSENSOR) from the sensor is a periodic physiological waveform attributed to small variations in the reflected IR light which is caused by the pulsatile tissue blood volume inside the finger. The waveform is, therefore, synchronous with the heart- beat. The following circuit diagram describes the first stage of the signal conditioning which will suppress the large DC component and boost the weak pulsatile AC component, which carries the information.

Operation

The operation of the board is very simple. After powering the board from a 3-5.5V supply, the Enable (EN) pin must be pulled high to activate the IR sensor. Next, place the tip of your forefinger gently over the sensor on its face. Your finger should be still and should not press too hard on the sensor. Within a couple seconds the circuit stabilizes and you will see the LED flashing synchronously with your heart beat. You can feed the output signal (Vout) to either a digital I/O or an ADC input pin of the microcontroller for measurement of the heart beat rate in BPM. The output voltage waveform can also be viewed on an oscilloscope. I connected Digilent's Analog Discovery tool to check the input PPG and the output waveforms from the two LPF stages.

2.5 TEMPERATURE SENSOR

The silicon band gap temperature sensor [8] is an extremely common form of temperature sensor (thermometer) used in electronic equipment. Its main advantage is that it can be included in a silicon integrated circuit at very low cost. The principle of the sensor is that the forward voltage of a silicon diode is temperature-dependent, according to the following equation:

$$V_{BE} = V_{G0}\left(1 - \frac{T}{T_0}\right) + V_{BE0}\left(\frac{T}{T_0}\right) + \left(\frac{nKT}{q}\right)\ln\left(\frac{T_0}{T}\right) + \left(\frac{KT}{q}\right)\ln\left(\frac{I_C}{I_{C0}}\right)$$

where

T = temperature in Kelvin

T_0= reference temperature

V_{G0} = band gap voltage at absolute zero

V_{BE0} = band gap voltage at temperature T_0 and current I_{C0}

K = Boltzmann's constant

q = charge on an electron

n = a device-dependent constant

By comparing the band gap voltages at two different currents, I_{C1} and I_{C2}, many of the variables in the above equation can be eliminated, resulting in the relationship:

$$\Delta V_{BE} = \frac{KT}{q} \cdot \ln\left(\frac{I_{C1}}{I_{C2}}\right)$$

An electronic circuit, such as the Brokaw band gap reference, that measures ΔV_{BE} can therefore be used to calculate the temperature of the diode. The result remains valid up to about 200 °C to 250 °C, when leakage currents become large enough to corrupt the measurement. Above these temperatures, materials such as silicon carbide can be used instead of silicon.

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor

- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 for 1 mA load

Fig 2.9 LM35 sensor

The LM35 - An Integrated Circuit Temperature Sensor

You can measure temperature more accurately than a using a thermostat.

The sensor circuitry is sealed and not subject to oxidation, etc. The LM35 generates a higher output voltage than thermocouples and may not require that the output voltage be amplified.

2.6 MICROCONTROLLERS

The heart of the project is the microcontroller. Here a PIC16F877A microcontroller is used to calculate the number of pulse count in a given instance and is transferred to the smart phone through Bluetooth module

2.6.1 High-Performance RISC CPU:

• Only 35 single-word instructions to learn

• All single-cycle instructions except for program branches, which are two-cycle

• Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle

• Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory

• Pin out compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

2.6.2 Peripheral Features:

• Timer0: 8-bit timer/counter with 8-bit prescalar

• Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock

• Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler

• Two Capture, Compare, PWM modules

- Capture is 16-bit, max resolution is 12.5 ns

- Compare is 16-bit, max. Resolution is 200 ns

- PWM max. Resolution is 10-bit

• Synchronous Serial Port (SSP) with SPI™ (Master mode) and I2C™ (Master/Slave)

• Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection

• Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)

• Brown-out detection circuitry for Brown-out Reset (BOR)

2.6.3 Analog Features:

• 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)

• Brown-out Reset (BOR)

• Analog Comparator module with:

- Two analog comparators

- Programmable on-chip voltage reference (VREF) module

- Programmable input multiplexing from device inputs and internal voltage reference

- Comparator outputs are externally accessible

Special Microcontroller Features:

• 100,000 erase/write cycle Enhanced Flash program memory typical

• 1,000,000 erase/write cycle Data EEPROM memory typical

• Data EEPROM Retention > 40 years

• Self-reprogrammable under software control

• In-Circuit Serial Programming™ (ICSP™) via two pins

• Single-supply 5V In-Circuit Serial Programming

• Watchdog Timer (WDT) with its own on-chip RC Oscillator for reliable operation

• Programmable code protection

• Power saving Sleep mode

• Selectable oscillator options

• In-Circuit Debug (ICD) via two pins

2.6.4 CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

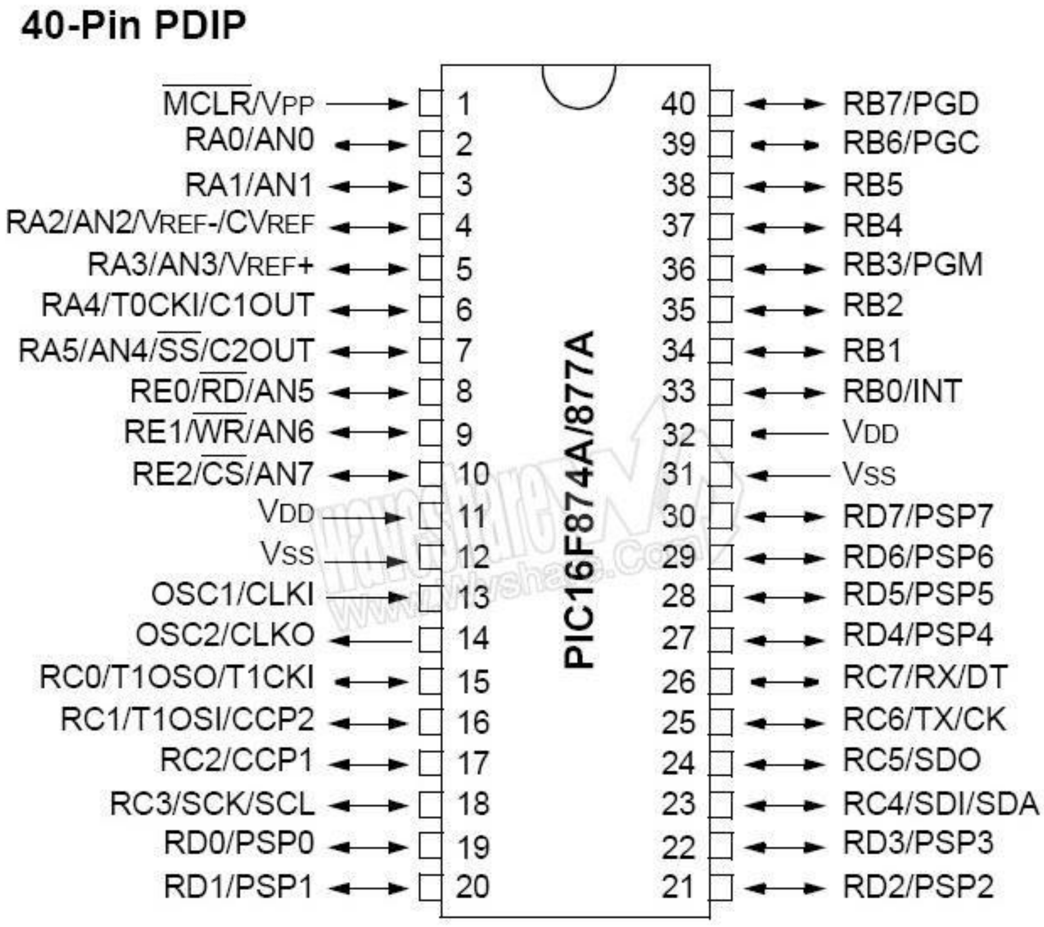


Fig 2.10 PIN diagram of PIC16F874A

2.6.5 DEVICE OVERVIEW

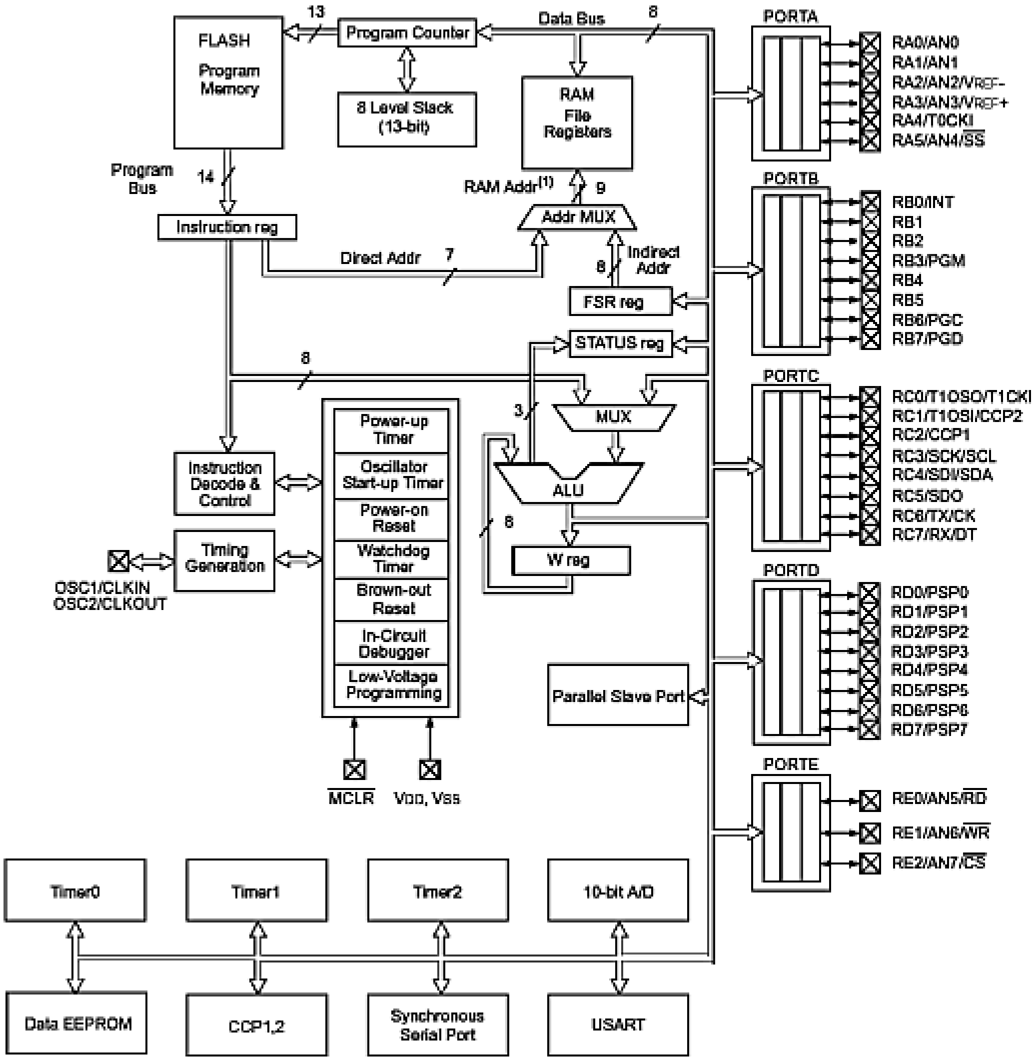
This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40 pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices.

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

Fig 2.11 Architecture of PIC16F874A

2.6.6 Memory Organization of PIC16F877

The memory of a PIC 16F877 chip is divided into 3 sections. They are

1. Program memory
2. Data memory and
3. Data EEPROMD

2.6.6.1 Program memory

Program memory contains the programs that are written by the user. The program counter (PC) executes these stored commands one by one. Usually PIC16F877 devices have a 13 bit wide program counter that is capable of addressing 8K×14 bit program memory space. This memory is primarily used for storing the programs that are written (burned) to be used by the PIC. These devices also have 8K*14 bits of flash memory that can be electrically erasable /reprogrammed. Each time we write a new program to the controller, we must delete the old one at that time. The figure below shows the program memory map and stack.

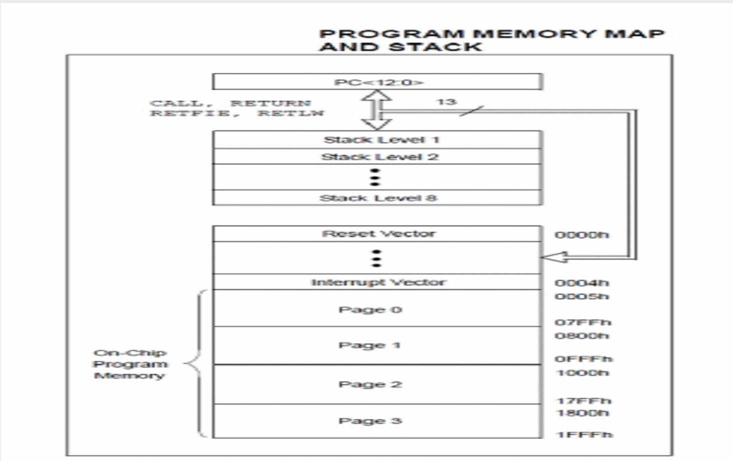


Fig 2.12 PIC16f877 Program Memory

2.6.6.2 PIC16F87XA Data Memory Organization

The data memory of PIC16F877 is separated into multiple banks which contain the general purpose registers (GPR) and special function registers (SPR). According to the type of the microcontroller, these banks may vary. The

PIC16F877 chip only has four banks (BANK 0, BANK 1, BANK 2, and BANK4). Each bank holds 128 bytes of addressable memory.

The banked arrangement is necessary because there are only 7 bits are available in the instruction word for the addressing of a register, which gives only 128 addresses. The selection of the banks are determined by control bits RP1, RP0 in the STATUS registers Together the RP1, RP0 and the specified 7 bits effectively form a 9 bit address. The first 32 locations of Banks 1 and 2, and the first 16 locations of Banks2 and 3 are reserved for the mapping of the Special Function Registers (SFR's).

BANK	RP0	RP1
0	0	0
1	1	0
2	0	1
3	1	1

Table 1 Bank access of RP1 & RP0 of the STATUS register

2.6.6.3 Data EEPROM and FLASH

The data EEPROM and Flash program memory is readable and writable during normal operation (over the full VDD range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers. There are six SFRs used to read and write this memory:

• EECON1

• EECON2

• EEDATA

• EEDATH

• EEADR

• EEADRH

The EEPROM data memory allows single-byte read and writes. The Flash program memory allows single-word reads and four-word block writes. Program memory write operations automatically perform an erase-before write on blocks of four words. A byte write in data EEPROM memory automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device for byte or word operations.

2.7 BLUETOOTH MODULE

In this project, we use Bluetooth module shown in fig 2.13, to transfer pulse count and temperature of the victim from microcontroller to the smartphone. In order to do this first the Bluetooth module and the smart phone needs to be paired.

Bluetooth wireless technology [6] is a 2.4GHz ISM-band open industry standard for short range wireless communication, which is capable of voice and data transfer (up to 723 kbps data transfer in ACL link or up to 3 simultaneous voice connections in SCO links in a piconet).

In this document, the SEMCO-Bluetooth Modules are presented, which consists of RF, baseband, and link manager protocol together with Host Controller Interface (HCI) / HCI-UART functionality implemented according to the Bluetooth specification version 1.1. SEMCO-Bluetooth Modules are designed for use as a universal Bluetooth module compliant for the Bluetooth specification version 1.1, which can be applied to hand-held phones, PDAs, headsets, PCs and PC-peripherals, et cetera for wireless voice and data communication. SEMCO-Bluetooth Modules or Evaluation Board Demo Kits are available by Samsung Electro-Mechanics for development.

Fig 2.13 Bluetooth kit v.1.1

2.8 GLOBAL SYSTEM FOR MOBILE COMMUNICATION

Global System for Mobile Communications(GSM), originally Groupe Spécial Mobile [2], is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe protocols for second-generation (2G) digital cellular networks used by mobile phones.

Communication between the health care/server and the victim’s smart phone is carried out through GSM.

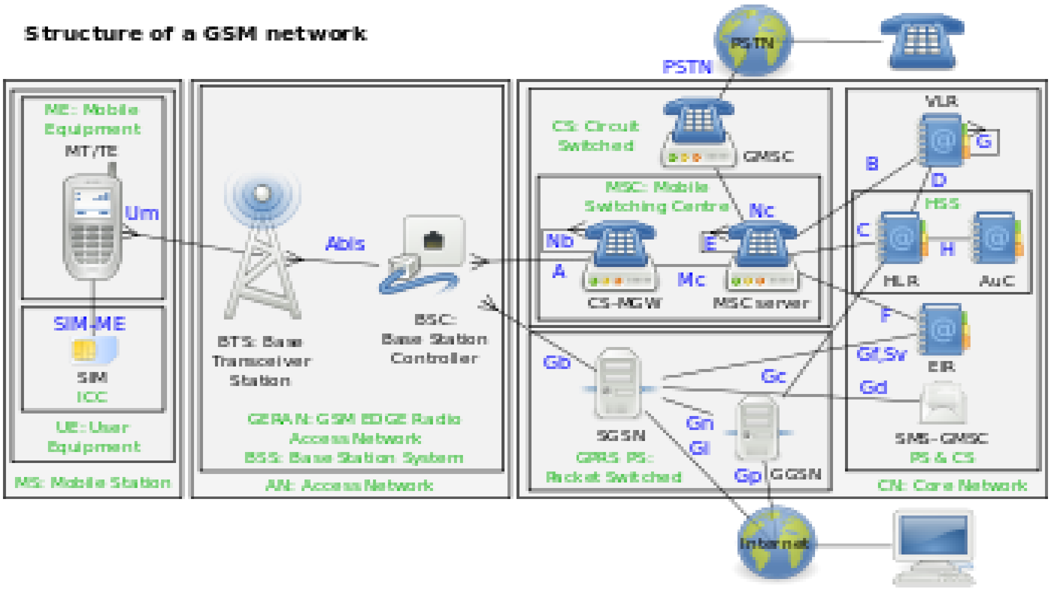


Fig 2.14 Structure of a GSM network

2.8.1 Network structure

The network is structured into a number of discrete sections:

- Base Station Subsystem – the base stations and their controllers explained
- Network and Switching Subsystem – the part of the network most similar to a fixed network, sometimes just called the "core network"
- GPRS Core Network – the optional part which allows packet-based Internet connections
- Operations support system (OSS) – network maintenance.

GSM is a cellular network, which means that cell phones connect to it by searching for cells in the immediate vicinity. There are five different cell sizes in a GSM network—macro, micro, Pico, femto, and umbrella cells. The coverage area of each cell varies according to the implementation environment. Macro cells can be regarded as cells where the base station antenna is installed on a mast or a building above average rooftop level. Micro cells are cells whose antenna height is under average rooftop level; they are typically used in urbanareas. Picocells are small cells whose coverage diameter is a few dozen metres; they are mainly used indoors. Femtocells are cells designed for use in residential or small business environments and connect to the service provider's network via a broadband internet connection. Umbrella cells are used to cover shadowed regions of smaller cells and fill in gaps in coverage between those cells.

Indoor coverage is also supported by GSM and may be achieved by using an indoor picocell base station, or an indoor repeater with distributed indoor antennas fed through power splitters, to deliver the radio signals from an antenna outdoors to the separate indoor distributed antenna system. These are typically deployed when significant call capacity is needed indoors, like in shopping centres or airports. However, this is not a prerequisite, since indoor coverage is also provided by in-building penetration of the radio signals from any nearby cell.

2.8.2 GSM carrier frequencies

GSM networks operate in a number of different carrier frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G), with most 2G GSM networks operating in the 900 MHz or 1800 MHz bands. Where these bands were already allocated, the 850 MHz and 1900 MHz bands were used instead (for example in Canada and the United States). In rare cases the 400 and 450 MHz frequency bands are assigned in some countries because they were previously used for first-generation systems.

Regardless of the frequency selected by an operator, it is divided into timeslots for individual phones. This allows eight full-rate or sixteen half-rate speech channels per radio frequency. These eight radio timeslots (or burst periods) are grouped into a TDMA frame. Half-rate channels use alternate frames in the same timeslot. The channel data rate for all 8 channels is 270.833 kbit/s, and the frame duration is 4.615 ms.

The transmission power in the handset is limited to a maximum of 2 watts in GSM 850/900 and 1 watt in GSM 1800/1900.

2.9 GLOBAL POSITIONING SYSTEM:

The Global Positioning System (GPS) [2] is a space-based global navigation satellite system that provides reliable location and time information in all weather and at all times and anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible by anyone with a GPS receiver.

In our project GPS is used to find the location of the victim when the latitude and longitude data of the victim is provided.

The GPS consists of three parts: the space segment, the control segment, and the user segment. The U.S. Air Force develops, maintains, and operates the space and control segments. GPS satellites broadcast signals from space, which each GPS receiver uses to calculate its three-dimensional location (latitude, longitude, and altitude) plus the current time.

The space segment is composed of 24 to 32 satellites in medium Earth orbit and also includes the boosters required to launch them into orbit. The control segment is composed of a master control station, an alternate master control station, and a host of dedicated and shared ground antennas and monitor stations. The user segment is composed of hundreds of thousands of U.S. and allied military users of the secure GPS Precise Positioning Service, and tens of millions of civil, commercial, and scientific users of the Standard Positioning Service (see GPS navigation devices).

2.9.1 Applications

GPS has become a widely used aid to navigation worldwide, and a useful tool for map-making, land surveying, commerce, scientific uses, tracking and surveillance, and hobbies such as geo caching and way marking. The precise time reference provided by GPS is used in many applications including the scientific study of earthquakes and as a time synchronization source for cellular network protocols.

In addition, GPS has, in the words of the website gps.gov, become a mainstay of transportation systems worldwide, providing navigation for aviation, ground, and maritime operations. Disaster relief and emergency services depend upon GPS for location and timing capabilities in their life-saving missions. The accurate timing provided by GPS facilitates everyday activities such as banking, mobile phone operations, and even the control of power grids. Farmers, surveyors, geologists and countless others perform their work more efficiently, safely, economically, and accurately using the free and open GPS signals.

2.10 CIRCUIT DIAGRAM

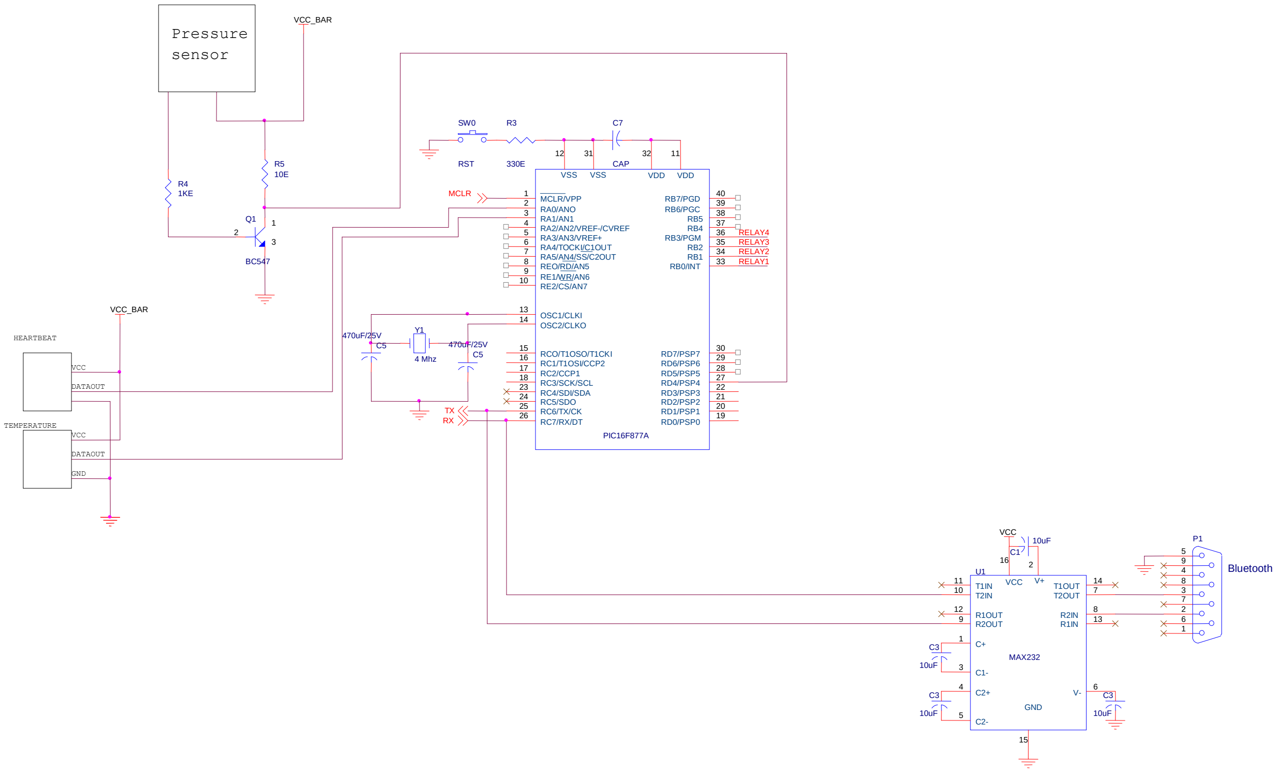


Fig 2.15 Circuit diagram of Heart-beat monitor

2.10.1 Circuit description

The pulse rate of the patient is detected using an IR pulse sensor. It is a clip based sensor in which the patient finger is placed in between the IR transmitter and receiver. When the heart pumps out the blood there will be no light received at the photo-receiver. Whereas when the heart pumps in the blood, the light transmitted form IR transmitter will be received by the receiver. Thus the time slots during which the photo-receiver receives the light is used to determine the pulse rate of the patient. This pulse rate is indicated by the blinking LED in the circuit. The LED output is connected to the PIC microcontroller. The microcontroller is programmed in such a way that it counts the pulse rate per minute. The temperature of patient is monitored using the LM35 sensor. It is interfaced with the microcontroller. The values counted in the microcontroller are transferred to the smartphone using bluetooth module version1.1. The bluetooth module synchronizes itself with the bluetooth of the smartphone and it readily transfers the counted pulse rate and temperature value to the smartphone. The smartphone has the developed android application which will accumulate the received values. In case of emergency during which the pulse rate or temperature value goes below / above the critical threshold value then the android application in the smartphone will transfer the obtained pulse rate value and temperature value of the patient to the healthcare server with help message. To enhance it better, the GPS feature of the smartphone is used to determine the exact longitude and latitude value of the current location of the patient. On determining the exact location of patient it transfers this location details to the healthcare server along with the help signal that is being sent. Also using the GSM feature of smartphone help messages are sent to the close relatives and family doctor. The healthcare server when updated with help message and location map of the victim they send an ambulance service to the exact location of the victim immediately.

3. SOFTWARE REQUIREMENTS

This section describes the software tools like java programming, J2EE application, java server page, XML, serverlets, eclipse, netbeans and android that have been used to interface the hardware components to the smartphone.It also describes the software required to compute the medical data obtained and activate a help signal to transmit it to the healthcare centre during an emergency situation.

3.1 JAVA (PROGRAMMING LANGUAGE):

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is considered by many as one of the most influential programming languages of the 20th century, and widely used from application software to web application.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of their Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Class path.

3.2 J2EE APPLICATION

A J2EE application or a Java 2 Platform Enterprise Edition application is any deployable unit of J2EE functionality. This can be a single J2EE module or a group of modules packaged into an EAR file along with a J2EE application deployment descriptor. J2EE applications are typically engineered to be distributed across multiple computing tiers.

Enterprise applications can consist of the following:

- EJB modules (packaged in JAR files);
- Web modules (packaged in WAR files);
- connector modules or resource adapters (packaged in RAR files);
- Session Initiation Protocol (SIP) modules (packaged in SAR files);
- application client modules;
- Additional JAR files containing dependent classes or other components required by the application;
- Any combination of the above.

3.3 JAVA SERVER PAGE

Java Server Pages (**JSP**) [7] is a Java technology that allows software developers to dynamically generate HTML, XML or other types of documents in response to a Web client request. The technology allows Java code and certain pre-defined actions to be embedded into static content.

The JSP syntax adds additional XML-like tags, called JSP actions, to be used to invoke built-in functionality. Additionally, the technology allows for the creation of JSP tag libraries that act as extensions to the standard HTML or

XML tags. Tag libraries provide a platform independent way of extending the capabilities of a Web server.

JSPs are compiled into Java Servlets by a JSP compiler. A JSP compiler may generate a servlet in Java code that is then compiled by the Java compiler, or it may generate byte code for the servlet directly. JSPs can also be interpreted on-the-fly reducing the time taken to reload changes

JavaServer Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server- and platform-independent.

3.3.1 Architecture OF JSP:

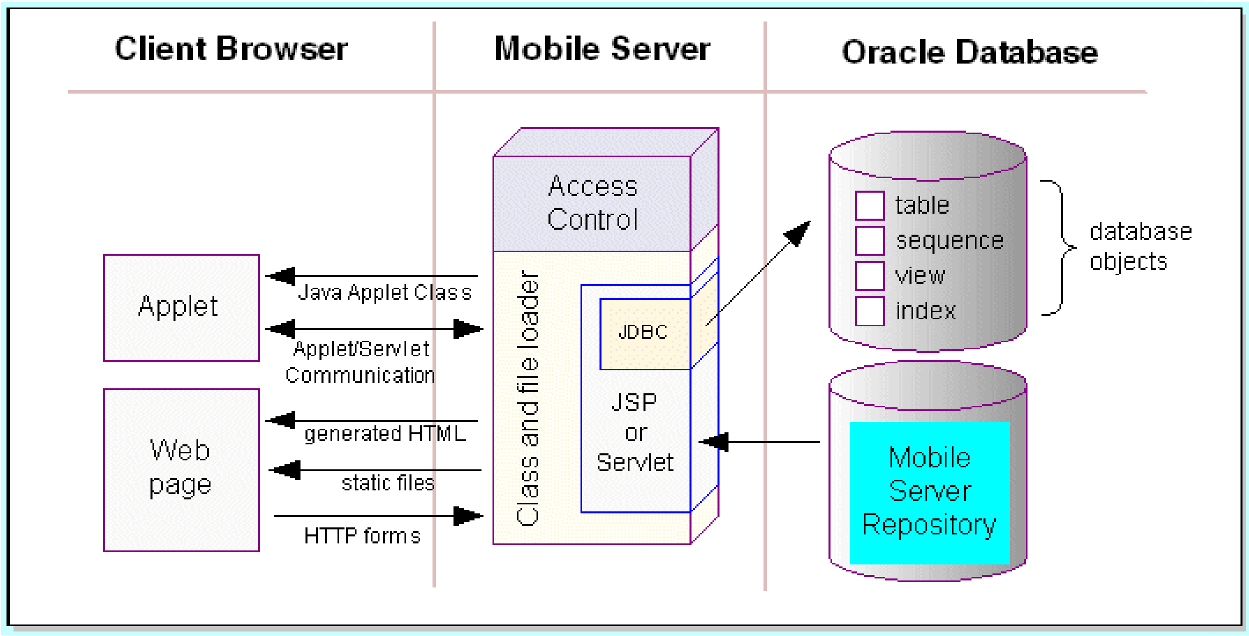


Fig 3.1 Architecture of JSP

3.3.2 Advantages of JSP:

- Active Server Pages (ASP). ASP is a similar technology from Microsoft. The advantages of JSP are twofold. First, the dynamic part is written in

Java, not Visual Basic or other MS-specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

- Pure Servlets. JSP doesn't give you anything that you couldn't in principle do with a servlet. But it is more convenient to write (and to modify!) regular HTML than to have a zillion println statements that generate the HTML. Plus, by separating the look from the content you can put different people on different tasks: your Web page design experts can build the HTML, leaving places for your servlet programmers to insert the dynamic content.
- Server-Side Includes (SSI). SSI is a widely-supported technology for including externally-defined pieces into a static Web page. JSP is better because it lets you use servlets instead of a separate program to generate that dynamic part. Besides, SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.
- JavaScript. JavaScript can generate HTML dynamically on the client. This is a useful capability, but only handles situations where the dynamic information is based on the client's environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can't access server-side resources like databases, catalogs, pricing information, and the like.
- Static HTML. XML, Regular HTML,XML of course, cannot contain dynamic information. JSP is so easy and convenient that it is quite feasible to augment HTML pages, XML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would preclude its use in all but the most valuable instances.

3.4 XML

Introduction:

MIDP devices have memory constraints when it comes to code, both in terms of the amount of code you can store on the device, and memory available to applications at runtime. So, keeping the size of applications and features in check is of paramount importance to the J2ME developer. That's where small-sized XML parsers come into play.

XML parsers

This section describes the XML parsing process and introduces some small XML parsers for MIDP.

XML parsing process

The XML parsing process operates in three phases:

1. XML input processing. In this stage, the application parses and validates the source document recognizes and searches for relevant information based on its location or its tagging in the source document; extracts the relevant information when it is located; and, optionally, maps and binds the retrieved information to business objects.

2. Business logic handling. This is the stage in which the actual processing of the input information takes place. It might result in the generation of output information.

3. XML output processing. In this stage, the application constructs a model of the document to be generated with the Document Object Model (DOM). It then either applies XSLT style sheets or directly serializes to XML.

An application that implements such a processing model is called an XML parser. You can integrate an XML parser into your Java applications with the Java API for XML Processing (JAXP). JAXP allows applications to parse and transform XML documents using an API that is independent of any particular XML processor implementation. Through a plug-in scheme, developers can change XML processor implementations without altering their applications.

3.5 SERVLETS

The Java Servlet API [7] allows a software developer to add dynamic content to a Web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the Java counterpart to non-Java dynamic Web content technologies such as PHP, CGI and ASP.NET. Servlets can maintain state across many server transactions by using HTTP cookies, session variables or URL rewriting.

The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of a Web container and a servlet. A Web container is essentially the component of a Web server that interacts with the servlets. The Web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

A Servlet is an object that receives a request and generates a response based on that request. The basic servlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The package javax.servlet.http defines HTTP-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the

Web server and a client. Servlets may be packaged in a WAR file as a Web application.

Servlets can be generated automatically by JavaServer Pages (JSP), or alternately by template engines such as WebMacro. Often servlets are used in conjunction with JSPs in a pattern called "Model 2", which is a flavour of the model-view-controller pattern.

Servlets are Java technology's answer to CGI programming. They are programs that run on a Web server and build Web pages. Building Web pages on the fly is useful (and commonly done) for a number of reasons:

- The Web page is based on data submitted by the user. For example the results pages from search engines are generated this way, and programs that process orders for e-commerce sites do this as well.
- The data changes frequently. For example, a weather-report or news headlines page might build the page dynamically, perhaps returning a previously built page if it is still up to date.
- The Web page uses information from corporate databases or other such sources. For example, you would use this for making a Web page at an on-line store that lists current prices and number of items in stock.

3.6 ECLIPSE

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, R, Ruby(including Ruby onRails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system. Its runtime system is based on Equinox, an implementation of the OSGi core framework specification.

In addition to allowing the Eclipse Platform to be extended using other programming languages, such as C and Python, the plug-in framework allows the Eclipse Platform to work with typesetting languages like LaTeX and networking applications such as telnet and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK, with support for other version control systems provided by third-party plug-ins.

With the exception of a small run-time kernel, everything in Eclipse is a plug-in. This means that every plug-in developed integrates with Eclipse in exactly the same way as other plug-ins in this respect, all features are "created equal". Eclipse provides plug-ins for a wide variety of features, some of which are through third parties using both free and commercial models. Examples of plug-ins include for UML, for Sequence and other UML diagrams, a plug-in for DB Explorer, and many others.

The Eclipse SDK includes the Eclipse Java development tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat filespace allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards.

Eclipse implements uses the graphical control elements of the Java toolkit called SWT, whereas most Java applications use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also uses an intermediate graphical user interface layer called JFace, which simplifies the construction of applications based on SWT. Eclipse was made to run on Wayland during a GSoC-Project in 2014.

3.7 NETBEAN SOFTWARE:

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++, and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

NetBeans Platform

Framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications, no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)
- Window management
- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library
- Integrated development tools

NetBeans IDE is a free, open-source, cross-platform IDE with built-in-support for Java Programming Language

GUI design tool:

Formerly known as project Matisse, the GUI design-tool enables developers to prototype and design Swing GUIs by dragging and positioning GUI components.

The GUI builder has built-in support for JSR 295 (Beans Binding technology), but the support for JSR 296 (Swing Application Framework) was removed in 7.1.

NetBeans JavaScript editor:

The NetBeans JavaScript editor provides extended support for JavaScript, Ajax, and CSS.

JavaScript editor features comprise syntax highlighting, refactoring, code completion for native objects and functions, generation of JavaScript class skeletons, generation of Ajaxcallbacks from a template and automatic browser compatibility checks.

CSS editor features comprise code completion for styles names, quick navigation through the navigator panel, displaying the CSS rule declaration in a List View and file structure in a Tree View, sorting the outline view by name, type or declaration order (List & Tree), creating rule declarations (Tree only), refactoring a part of a rule name (Tree only).

The NetBeans 7.4 and later uses the new [Nashorn] JavaScript engine developed by Oracle

3.8 ANDROID

It is a free, open source mobile platform [9]. Linux-based, multi process, Multithreaded OS. Android is not a device or a product it’s not even limited to phones you could build a DVR, a handheld GPS, an MP3 player, etc. Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

- Makes mobile development easy.
- Full phone software stack including applications
- Designed as a platform for software development
- Android is open
- Android is free
- Community support

3.8.1 Features

- Application framework enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open source WebKit engine
- Optimized graphics powered by a custom 2D graphics library; 3Dgraphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- SQLite for structured data storage.
- Media support for common audio, video, and still image formats(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Telephony (hardware dependent)
- Bluetooth, EDGE, 3G, and WiFi (hardware dependent)
- Camera, GPS, compass, and accelerometer (hardware dependent)
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE.

3.8.2 Linux Kernel

Android relies on Linux version 2.6 for core system services such as

- security
- memory management
- process management
- network stack
- driver model
- The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

3.8.3 Android Runtime:

- Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.
- Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently.
- The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint.
- The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

3.8.4 Libraries:

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.

- System C Library
- Media Library
- Surface Manager
- LibWebCore
- SGL
- 3D libraries
- Free Type
- SQLite

3.8.5 Development Tools:

The Android SDK includes a variety of custom tools that help you develop mobile applications on the Android platform. Three of the most significant tools are:

- Android Emulator -A virtual mobile device that runs on our computer - use to design, debug, and test our applications in an actual Android run-time environment
- Android Development Tools Plugin -for the Eclipse IDE - adds powerful extensions to the Eclipse integrated environment
- Dalvik Debug Monitor Service (DDMS) -Integrated with Dalvik -this tool let us manage processes on an emulator and assists in debugging
- Android Asset Packaging Tool (AAPT) – Constructs the distributable Android package files (.apk)
- Android Debug Bridge (ADB) – provides link to a running emulator. Can copy files to emulator, install .apk files and run commands.

3.8.6 Architecture

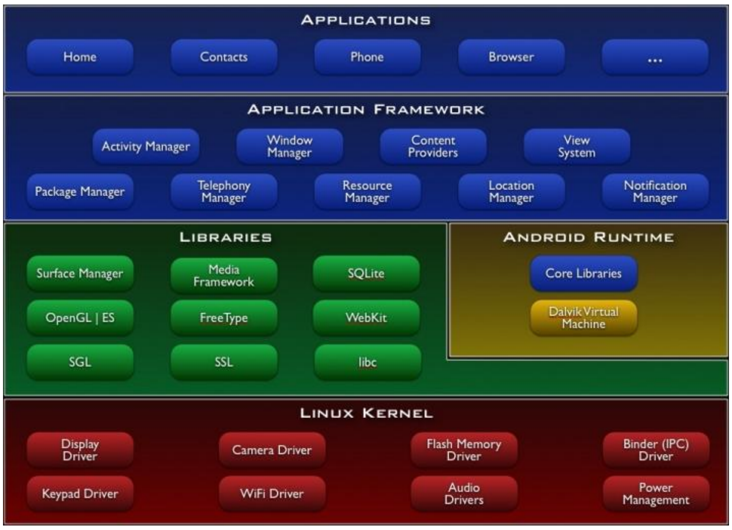


Fig 3.2 Architecture of Android development tool

There are four building blocks for an Android application:

- Activity - a single screen
- Broadcast Receiver- to execute in reaction to an external event(Phone Ring)

- Service - code that is long-lived and runs without a UI(Media Player)
- Content Provider - an application's data to be shared with other applications

Android Building Blocks

These are the most important parts of the Android APIs:

- AndroidManifest.xml
  - the control file-tells the system what to do with the top-level components
- Activities
  - An object that has a life cycle-is a chunk of code that does some work.
- Views
  - an object that knows how to draw itself to the screen
- Intents
  - A simple message object that represents an "intention" to do something.
- Notifications
  - is a small icon that appears in the status bar(SMS messages)
  - for alerting the user
- Services
  - It is a body of code that runs in the background.

4. CONCLUSION

The project helps by acting as a life saver in attending to the critical medical situations immediately. The device uses the optical technology to detect the flow of blood through the finger and offers the advantage of portability over tape-based recording systems. The performance of HRM device was compared with signal represented on an oscilloscope and manual pulse measurement of heartbeat, giving excellent results. This proposed Heart Rate Measuring (HRM) device is economical and user friendly. This project also shows how a smartphone can be used for the most critical and constructive purpose of saving a life in addition to its own features. However, since smartphone is not only used for healthcare monitoring, but also for other applications, i.e., phoning with friends, the smartphone's energy could be insufficient when an emergency takes place. Although this kind of unexpected event may happen with very low probability, m-Health care system is still challenging in emergency. Recently, opportunistic computing, as a new pervasive computing paradigm, has received much attention.

5. REFERNCES

1. “Microcontroller and Embedded System”, Rajiv kapadia,Jaico publishing house,2004.
2. R. Fensli, "A Wireless ECG System for Continuous Event Recording and Communication to a Clinical Alarm Station", Proc of the 26th Annual International Conference of the IEEE EMBS, 2004.
3. Mohamed Fezari, Mounir Bousbia-Salah, and Mouldi Bedda, “Microcontroller Based Heart Rate Monitor”, *The International Arab Journal of Information Technology*, Vol. 5, No. 4, 2008.
4. R.G. Landaeta, O. Casas, and R.P. Areny, “Heart rate detection from plantar bioimpedance measurements”, *28th IEEE EMBS Annual International Conference*, USA, 2006, pp. 5113-5116.
5. H. Shim, J.H. Lee, S.O. Hwang, H.R. Yoon, and Y.R. Yoon, “Development of heart rate monitoring for mobile telemedicine using smartphone”, *13th International Conference on Biomedical Engineering (ICBME 2008)*, Singapore, 2008, pp. 1116-1119.
6. A. Toninelli, R. Montanari, and A. Corradi, “Enabling Secure Service Discovery in Mobile Healthcare Enterprise Networks,” in Transcation IEEE Wireless Comm., vol. 16, no. 3, pp. 24-32, June 2009.
7. www.sparkfun.com
8. http://developer.android.com