ECG Signal Processing and Real-time Wireless Transmission

Mrs. Monali Chaudhary[#1], Ronit Makhijani[#2],Sanjay Badlani[#3],Sayan Daskarmakar[#4],Namrata Pai[#5]

#1 Assistant Professor, Department of Electronics and Telecommunication, VESIT, Mumbai, monali.chaudhary@ves.ac.in, +919867550252.

#2 B.E Student, Department of Electronics and Telecommunication, VESIT, Mumbai, ronit.makhijani@ves.ac.in, +917276514123.

#5 B.E Student, Department of Electronics and Telecommunication, VESIT, Mumbai, sanjay.badlani@ves.ac.in, +919860808069.

#3 B.E Student, Department of Electronics and Telecommunication, VESIT, Mumbai.

Sayan.daskarmakar@ves.ac.in, +919594031522.

#4 B.E Student, Department of Electronics and Telecommunication, VESIT, Mumbai, namrata.pai@ves.ac.in, +918767686124.

ABSTRACT:This paper aims at shrinking the time required to diagnose a patient and complete theformalities to admit a patient when he/she is brought to the hospital.To achieve this, we measureall the essential parameters like ECG, blood pressure, sugar level, etc. in ambulance itself and upload on hospital website using FTDI module. Along with this, a provision of online form filling is also provided on the website. A low cost, modular and flexible health care device is designed which includes amplification using AD620 instrumentation amplifier and operational amplifiers,analog filters viz., low pass (LPF), high pass (HPF) and notch filter to process the ECG signal. Digital filters are also implemented using Atmega32 microcontroller. A real time ECG as well as other parameters of the subject can be remotely monitored over the hospital website.

Key words: ECG, FTDI, analog filters, digital filters, hospital website, microcontroller, instrumentation amplifier.

INTRODUCTION

Nowadays, Telemedicine Technology is emerging enormously and is providing efficient solutions in medical field. Fusion of telemetry and medicine has anability to generate interactive healthcare systems like home-monitoring, tele-consultation, tele-learning, etc. However, emergency cases are found to be increasing as the number of people suffering from heart diseases is increasing at an alarming rate. Delay in the treatment may affect a patient severely or may also prove fatal. Also, theECG machine is an expensive device and its use

for the measurement of heart rate only is not economical. Keeping all these scenarios in mind, this paper describes the design of wirelessly monitoring the real time ECG of a patient in an inexpensive way. Here, the doctor in the hospital can have a view of vital statistics of a subject who is being brought to the hospital in an ambulance.

The Electrocardiography (ECG) signal represents the unique heart beat behavior. The characteristics of the ECGsignal, including the heart rate, the PR interval (as shown by P-R segment), the QRS duration (as shown by Q-R-S segment), the QT interval, etc., as shown in *Figure 1*, are important evidence for doctors to diagnose diseases.
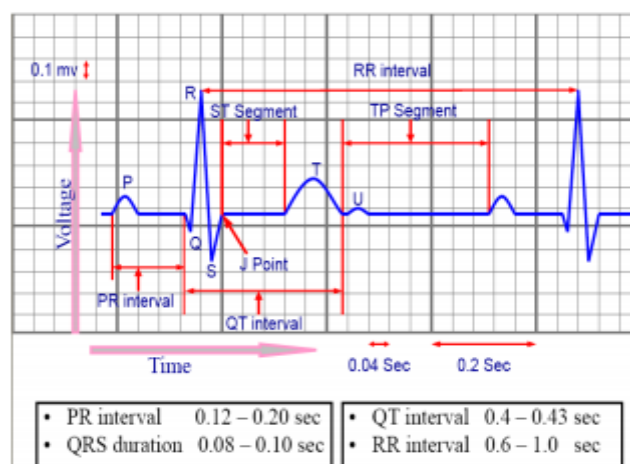

Figure 1: Typical ECG waveform

ECG is a bipolar low-frequency weak signal. Normally, ECG signal frequencies vary in the range of 0.05-100Hz, mainly concentrate in the range of 0.05-35Hz. Typical amplitude range is from 10μV to 5mV. Such small amplitude signals need to be amplified before further processing. Also, to use an ECG signal for diagnosis, it needs to be filtered to remove DC signals and artifacts as well as the high-frequency interference signals from the natural environment and the body. Any obvious variation of theECG waveform induced by ECG processing may cause a misdiagnosis.

PROPOSED MODEL

Block representation of proposed ECG model is as shown in *Figure 3*. The system is divided into five parts-

1. ECGAcquisition
2. Pre-Amplification
3. Analog Filtering And Amplification
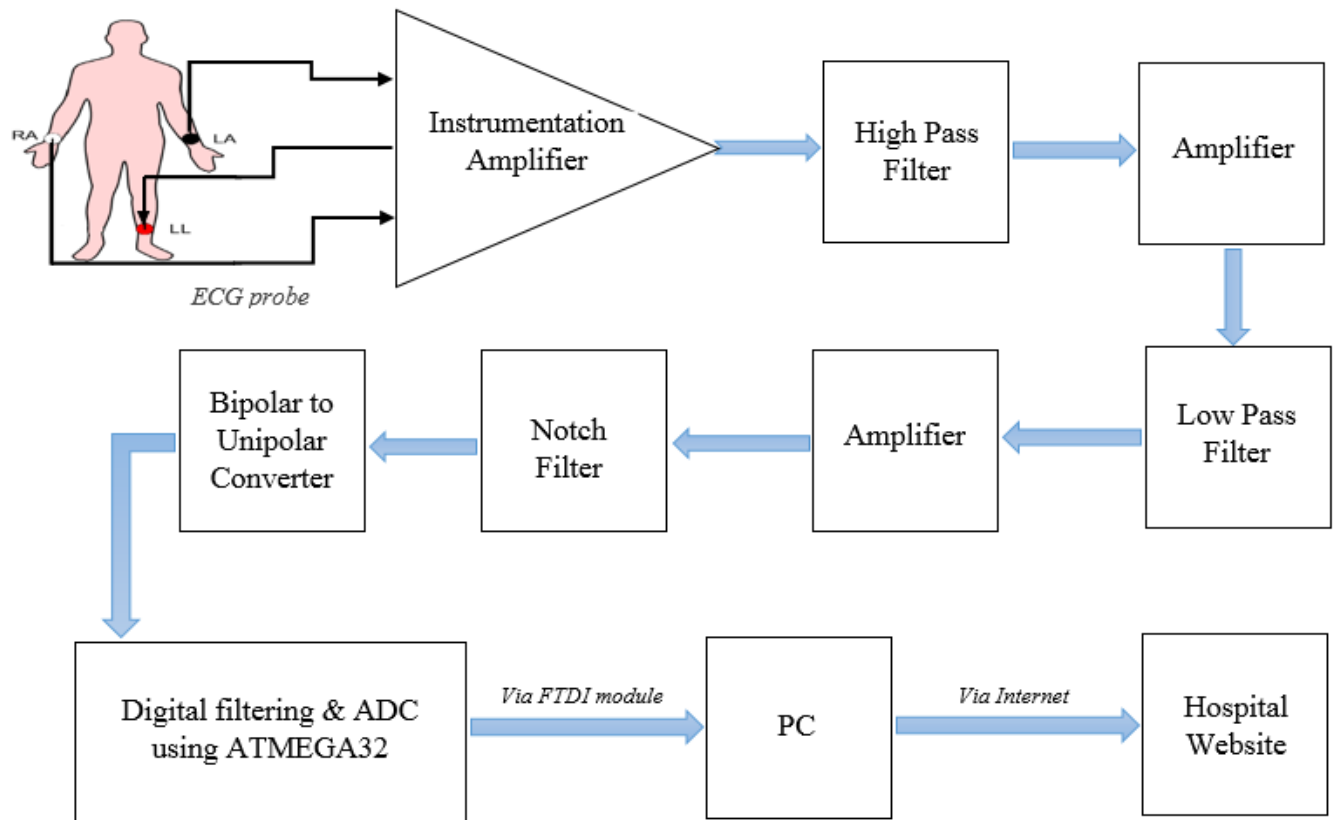4. ADC And Digital Filtering
5. Wireless Transmission

Figure 3: Block Diagram of proposed ECG model

1. ECG acquisition

Ag/AgCl electrodes are used that acts as a skin surface transducer to measure the electrical signal of heart. A standard clinical ECG includes 12 different leads, but in our design we'll focus on leads I, II and III as shown in *Figure 2*.
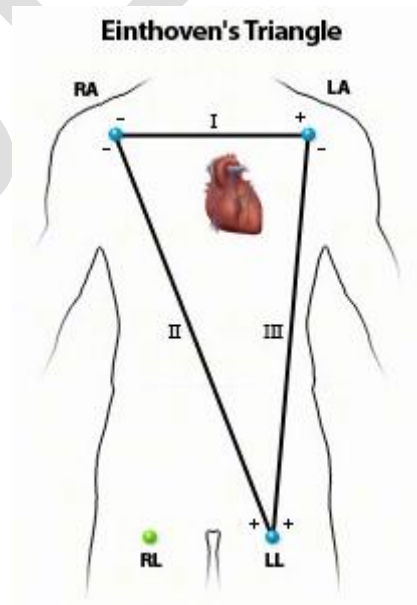

Figure 2: Lead Position forming Einthoven's triangle

Leads on the Right Arm (RA), left arm (LA), and Left leg (LL) form a triangle known as Einthoven's Triangle. The measured signals from LA and RA are subtracted while feedback to the LL acts as a ground reference to cancel the interference.

2. Pre amplification

As we know, ECG signal is a weak signal of few millivolts. Also, noise may get added to ECG signal during acquisition process. Handling such a small signal is tedious as it is difficult to distinguish between signal and noise. So, ECG signal needs to be amplified before further processing. We require an overall gain of 1000 to raise the ECG signal to volts. This is achieved using 3 amplification stages of gain 10. The first stage is pre- amplification using AD620. Circuit diagram of pre amplifier is as shown in *Figure 4*.
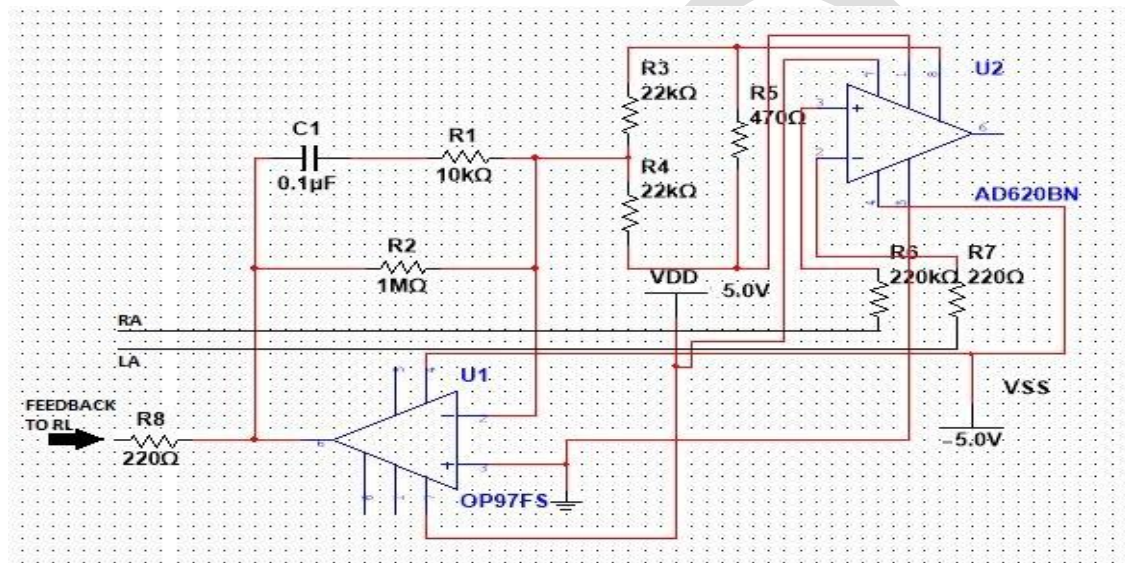

Figure4: Pre amplification using AD620 instrumentation amplifier

The AD620 instrumentation amplifier is a low cost, highaccuracy amplifier, with CMRR greater than 100 dB to nearly 1 kHz. To set the gain of AD620, only a single external gain resistor, RG is required and the gain is calculated as (Gain = 1 + 49.4 k/RG). The gain was selected to be 10 by setting RG to 4.7KΩ. Thus, the 3mV amplitude of our signal is amplified to approximately 30 to 33mV.

A right leg drive circuit is used to cancel the interference and improve the CMRR. The right leg drive inverts and amplifies the average common mode signal back into the patient's right leg. This action cancels 50 Hz noise from AC power and creates a cleaner ECG output signal. The more gain in feedback loop, more is the value of common mode rejection ratio. Cancelling noise in this way relaxes the attenuation needed from the common mode rejection of the instrumentation amplifier.

3. Analog Filtering and Amplification

In pre amplification process, along with ECG signal, noise also gets amplified. To avoid distortion in ECG signal, this noise needs to be eliminated without affecting the original

signal. Analog filters are assigned to remove unwanted noise from signal. Sources of noise that affects ECG signal are as follows:

- Baseline wander (low frequency noise)
- Power line interference (50Hz or 60Hz noise from power lines)
- Electromyographic(EMG) noise (This noise is very difficult to remove as it is in the same region as the actual signal.)

Table 1 shows the solution to eliminate these noise signals.

Table 1: Analog filters used to remove unwanted noise

Sr.No.	Noise Type	Filter Used To Remove Noise	Cutoff Frequency
1	Baseline Interference	High Pass Filter	0.5 Hz
2	Power Line Wander	Notch Filter	50 Hz
3	EMG Noise	Low Pass Filter	100 Hz

i.   High Pass Filter (HPF) : removal of baseline wander

Baseline wander is a low-frequency component present in the ECG system. It may be caused from a variety of noise sources including perspiration, respiration, body movements, and poor electrode contact. This can cause problems in the analysis of the ECG waveform. Its spectral content is usually confined to an interval below 1Hz (between 0.15 and 0.3 Hz).The heart rate can drop to 40 bpm, implying the lowest frequency to be 0.67 Hz. Therefore, the cut-off frequency should be such that the ECG remains undistorted while the baseline wander must be removed. A HPF is implemented with cutoff frequency of 0.5 Hz as shown in *Figure 5*.
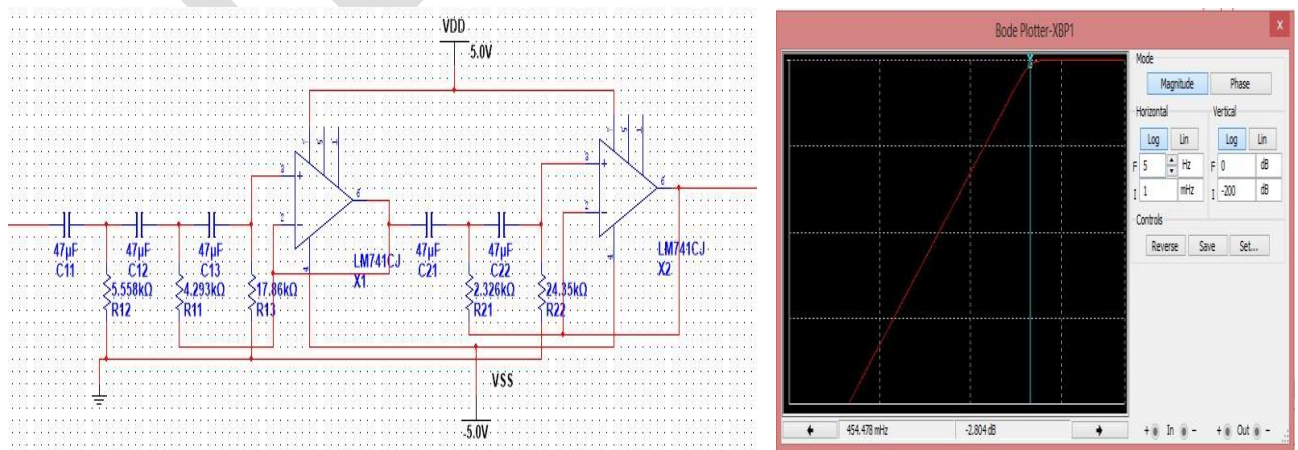

Figure 5: High Pass Filter with Frequency Response Plot

HPF designed is a 4th order Butterworth filter. It is an active filter as it uses OPAMPTL072.

ii.   Notch Filter : removal of Power Line Interference

Power Line noise is high frequency noise caused due to improper grounding of the ECG equipment. This noise degrades the signal quality. The amplitude of power-line noise is very large and generally gets coupled into the system even after prevention. It is a narrow-band noise centred at 50 Hz (or 60 Hz) with a bandwidth of less than 1 Hz. It is removed by implementing a notch filter which is shown in *Figure 6*.
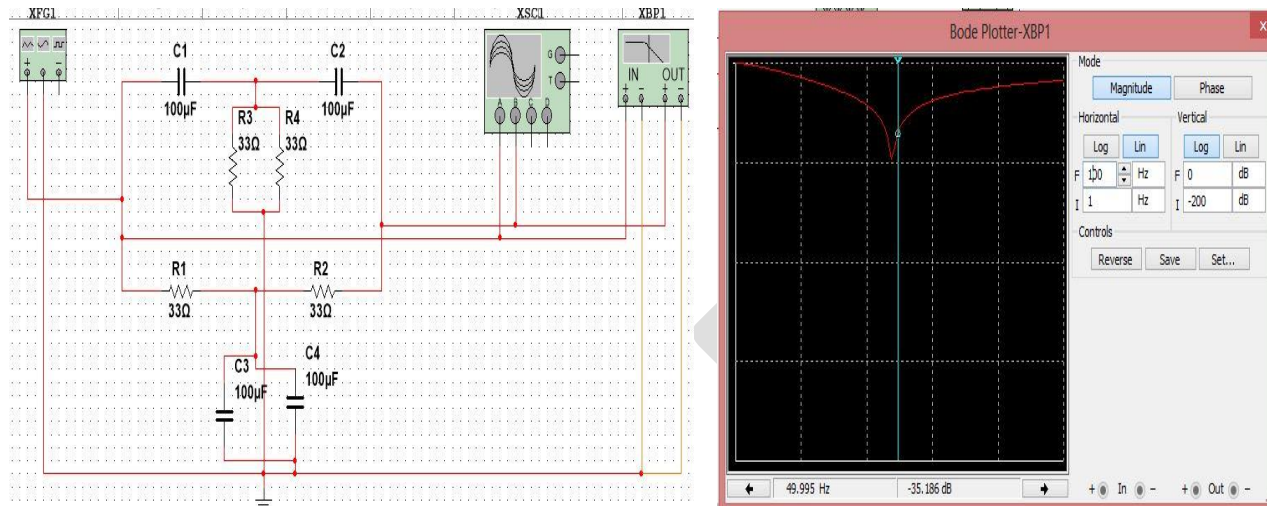

Figure 6: Notch filter with Frequency Response Plot

A passive notch filter is designed at 50 Hz. It filters the signal as well as noise that appears at 50 Hz.

iii.   Low Pass Filter (LPF) : removal of Electromyographic (EMG) Noise

EMG is mainly caused by the electrical activity of skeletal muscles during periods of contraction or due to a sudden body movements. Muscle noise can cause severe problems as low-amplitude waveforms can be obstructed. Muscle noise is difficult to remove because it is a wideband signal and its frequency spectrum with PQRST complex. Thus, filtering of muscle noise introduces some error in ECG signal. It has a frequency range between dc and 10,000 Hz with amplitude of 10% level. The high frequency components is discarded using a low pass filter designed with 100 Hz of cutoff frequency. Designed LPF is as shown in *Figure 7*.
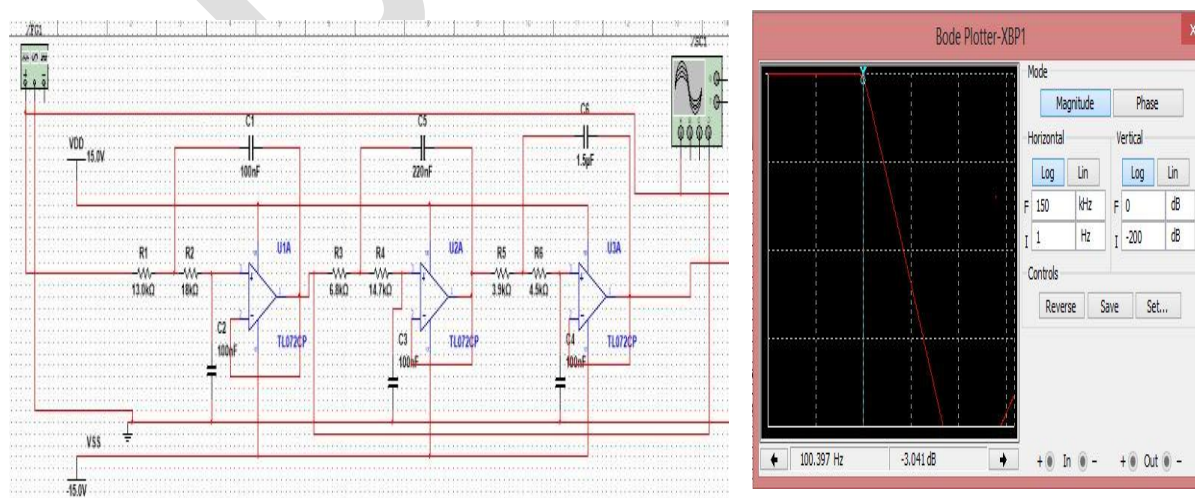

Figure 7: Low Pass Filter with Frequency Response

Similar to HPF, LPFis also an active Butterworth filter. OPAMP TL072 is used in filter circuit. To improve the performance of filter, it is designed with order of 6. Cutoff frequency of filter is 100 Hz.

Between two adjacent filtering circuits, an amplifier is used. It is designed using IC LM741 with gain 10 as shown in *Figure 8*.
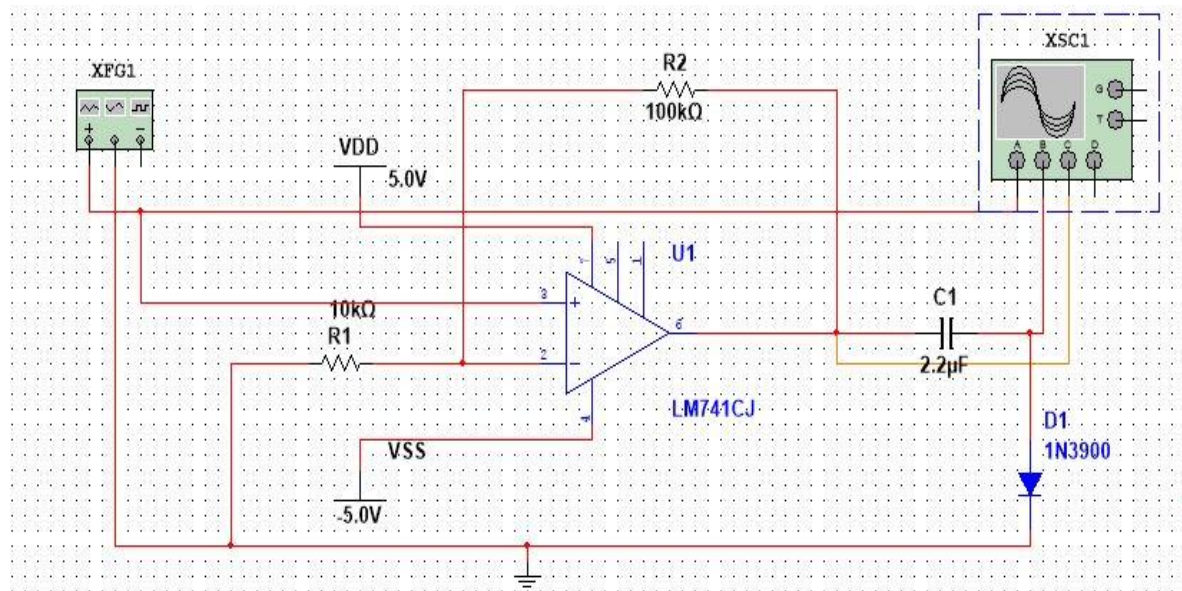

Figure 8: Amplifier circuit

Two such stages of amplification are used as shown in block diagram in*Figure 3*. All the Filtering and Amplifying circuits are tested using MULTISIM software.

4. ADC and Digital Filtering

Digital signals are preferred for wireless transmission as they are more immune to noise. Other advantages of digital transmission include- typically use less bandwidth and can be encrypted. On the other hand, though analog filters have eliminated the unwanted noise but to enhance the quality of output and obtain the ECG signal in nearly accurate shape, the signal was digitally filtered. To achieve this, ADC along with digital low-pass, notch and band-pass filters were implemented using low cost and high performance ATMEGA32 microcontroller. Digital processing is preferred as it can be reprogrammed, easily duplicated and digital signals can be easily stored which makes remote processing possible. To reduce the complexity associated with coding of digital filters, a simplified code algorithm using c-language for ADC and filtering was developed. The ADC sampling rate was set the same (500 Hz) as the filters' sampling rate to achieve the proper performance.

Since ATMEGA32 cannot process analog signals, so bipolar to unipolar circuit was used to shift the signal above zero level. Bipolar-to-unipolar converter uses OP97 operational amplifier in non-inverting mode. Circuit for bipolar-to-unipolar conversion is as shown below in *Figure9*.
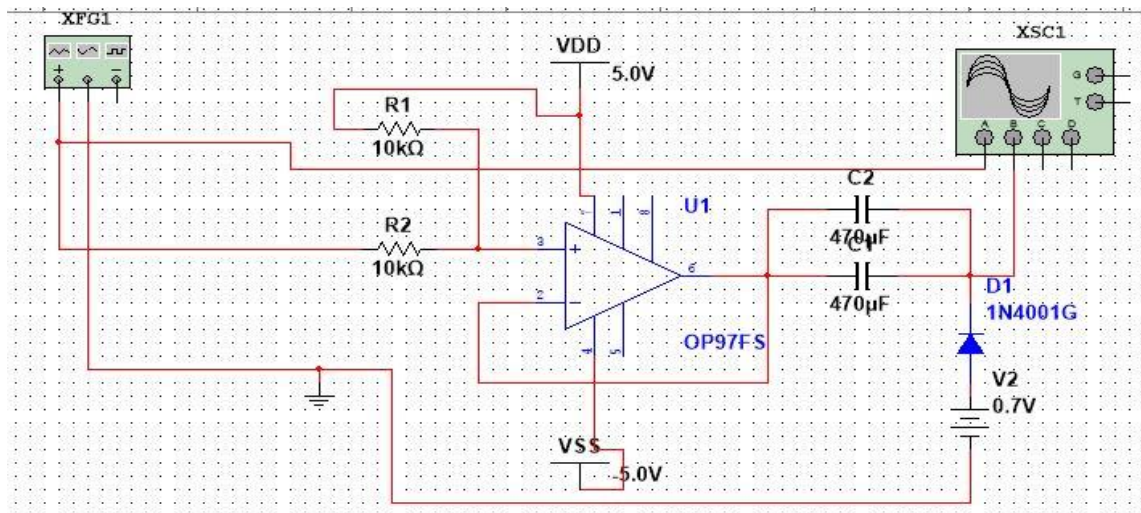
Figure 9: Bipolar-to-Unipolar converter

5.   Wireless Transmission

After Filtering, amplification and digitizing the ECG signal, the last and very important stage of our design is transmitting the signal to hospital server via internet. The digitized signal is brought to PC using a FTDI FT232R module. FTDI module acts as interface between UART port of ATMEGA 32 and USB port of PC or laptop.The received data is converted to 'Comma Separated Values (CSV)' format. This CSV file is then sent to hospital web server. The ECG graph is reproduced on hospital webpage by extracting the data from CSV file. This whole process can be divided into following hops:

i.   Microcontroller to PC

ii.   PC to Web Server

iii.   Reproducing Graph on Web Page

iv.   Microcontroller to PC

For communicating the microcontroller with the PC we have adopted a USB to UART module which communicates between them.A Universal Asynchronous Receiver/Transmitter, abbreviated UART is a piece of computer hardware that translates data between parallel and serialforms. Theuniversal designation indicates that the data format and transmission speeds are configurable.UART is used in conjunction with FT232R module.UART is coded to transmit the digital data serially to FT232R. Code for UART is provided in Appendix. In PC, the data can be seen in HyperSerialPort (HSP). HyperSerialPort is a Serial Terminal Emulator that works with any Serial Port including USB to Serial Virtual COM Ports. The HSP has several built in test tools that assist in the development of embedded electronics such as: Python Scripting Evaluator, Auto-Send Charactersand External Trigger etc. It is built on the .NET Framework 4.0 from Microsoft and designed for Windows 7 and 8 Systems.HSP allows to save the file in .csv format.

FT232R module

Figure 10: FT232R

The FT232R is a USB to serial UART interface with optional clock generator output and the new FTDI chip security dongle feature as shown in Figure 10. This little breakout is built around the FT232RL IC from FTDI, with an internal oscillator, EEPROM, and a 28-pin SSOP package.USB to serial designs using the FT232R have been further simplified by fully integrating the external EEPROM, clock circuit and USB resistors onto the device.

   v.   PC to Hospital Web Server

CSV file is uploaded on web server via internet provided through dongle. With high speed internet, it is possible to transfer the file within seconds.

   vi.   Reproducing Graph On Web Page

Data from .csv file is parsed using a data parsing tool. Many soft wares are available which can extract data from .csv files. One such software is Parse-O-Matic. The data retrieved from .csv file is used to reproduce an ECG graph on Hospital web page using PHP and Jquerry. The frame work of website was done using HTML. Designed web page is as shown in *Figure 11*.
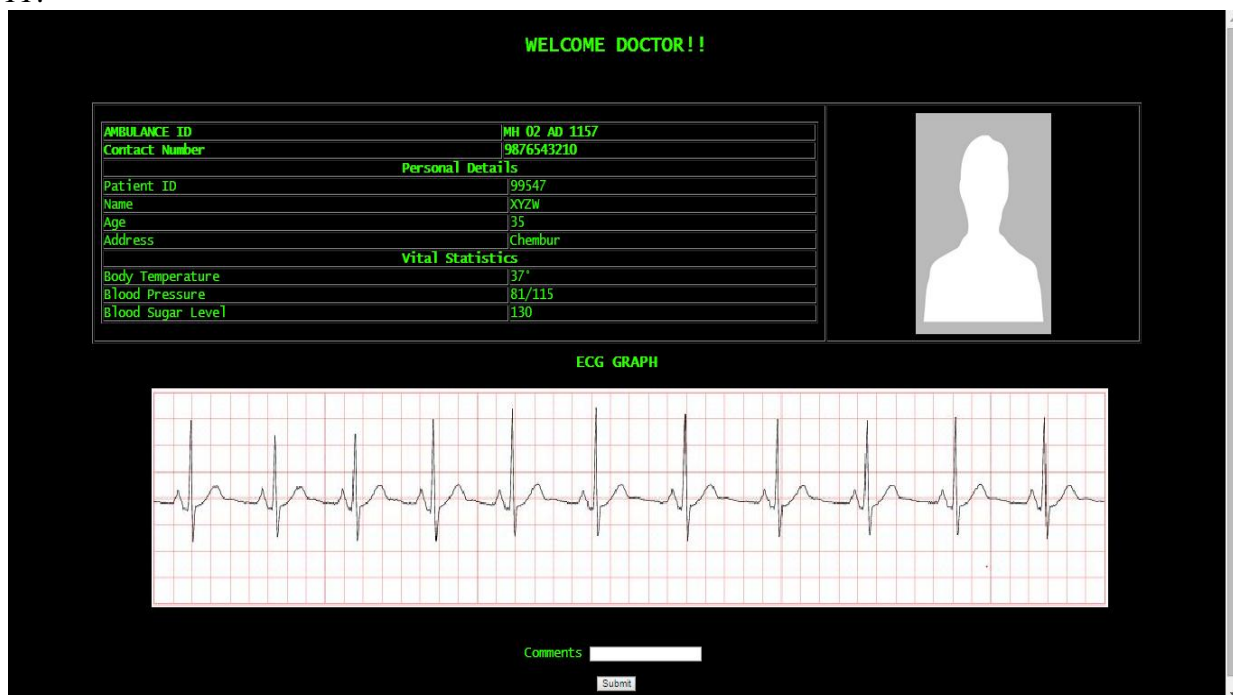

Figure 11: Snapshot of Web page designed for Online Monitoring

CONCLUSION

The designed health care system is a fullproof prototype of processing and wireless transmission of ECG signal. It is an economical and efficient model capable of cracking medical emergency situations. The proposed concept of online monitoring of ECG and online form filling will definitely help to compress the time wasted when patient is brought to the hospital. The further research may include discovery of new techniques of real-time transmission of ECG signal as wireless technologies are advancing day by day.

REFERENCES

[1]   M. Arpitchoudhary, K. Kumar, R. Kumar, S. Singh, "Design anddevelopment of low cost microcontroller based ECG system for realtime

[2]   Analysis", International Journal of Electronics, vol. 2, vol. 3m pp.121- 129, September 2012.

[3]   Analog Fundamentals of the ECG Signal Chain, a Texas Instruments technical presentation prepared by Matthew Hann.

[4]   K Siddique-e Rabbani, A RaihanAbir, A K M Bodiuzzaman, BJMP,vol4, issue1,2011, Design and development of low cost personal computer based ECG monitor

[5]   Joseph J. Carr and John M. Brown, "Introduction to Biomedical Equipment Technology", Pearson Education Fourth Edition. ISBN: 81- 7758-883-4

[6]   P. S. Hamilton, A comparison of adaptive and nonadaptive filters for the reduction of powerline interference in the ECG. IEEE Trans. Biomed. Eng. 1996; 43:105–109.

[7]   http://www.academia.edu/8433789/Digitally_Filtered_ECG_Signal_Using_Low-Cost_Microcontroller

[8]   socrates.berkeley.edu/~phylabs/bsc/Multisim/multisim_tutorial_2.pdf

[9]   www.petervis.com/C/testing%20uart/testing%20the%20uart.html

[10]   http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

[11]   www.CardiacBasicPhysiology.com

APPENDIX: C-CODE FOR UART

```
/*
 * GccApplication1.c
 *
 * Created: 3/22/2015 11:25:12 AM
 *  Author: sameer
 */


//#define F_CPU 16000000
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdlib.h>
#include <inttypes.h>
// global variables
static int8_t data, output;
static float xn, xn1, xn2, xn3, xn4, xn5, xn6;
static float yn, yn1, yn2, yn3, yn4, yn5, yn6;
static float zn, zn1, zn2, zn3, zn4 ;
static float wn, wn1, wn2, wn3, wn4;
```

```c
voidadc_init();
void delay_100us(int n);
voiduart_init(uint16_t);
uint8_t adc_read();
uint8_t filter_adc(uint8_t xn);
voiduart_write_char(uint8_t);
int main(void)
{
        DDRB=0xFF;
        adc_init(); // initialize ADC
        uart_init(103);
        while (1) // loop forever
        {
                data=adc_read();
                output=filter_adc(data);
                uart_write_char(output);
                PORTB= output;
                //delay 2ms to make sure that the sampling
                //frequency is 500Hz (fs=500Hz --> 2ms)
                // 0.7ms is the time wasted in filtering
                delay_100us(20-7);
        }
}
voiduart_init(uint16_t ubrr_value)
{
        UBRR0L = ubrr_value;
        UBRR0H = (ubrr_value>>8);
        UCSR0C = (1<<USBS0)|(3<<UCSZ00);
        UCSR0B = (1<<RXEN0) |(1<<TXEN0);
}
voidadc_init()
{
        ADCSRA |= 1<<ADPS2; //16 pre-scalar
        ADMUX |= 1<<ADLAR;
        ADMUX |= 1<<REFS0;
        ADCSRA |= 1<<ADIE;
        ADCSRA |= 1<<ADEN;
}
uint8_t adc_read()
{
        // start single conversion
        // write '1' to ADSC
        ADCSRA |= (1<<ADSC);
        // wait for conversion to complete
        // ADSC becomes '0' again
        // till then, run loop continuously
        while(ADCSRA & (1<<ADSC));
        return (ADCH);
}
uint8_t filter_adc(uint8_t data)
{
        xn= (float) data; // adc data
        //------Lowpass Filter---------//
        yn=0.0018*xn+ 0.0111*xn1+ 0.0277*xn2+ 0.0369*xn3+
        0.0277*xn4+ 0.0111*xn5+ 0.0018*xn6 +2.6189*yn1 -
        3.36580*yn2 +2.4851*yn3 -1.0978*yn4 +0.2704*yn5 -
        0.0288*yn6;
```

```
        //-----Notch Filter------------//
        zn= yn - 1.618*yn1 + yn2 + 1.5164*zn1 - 0.8783*zn2;
        //-----Bandpass Filter---------//
        wn=0.0627*zn -0.1254*zn2 +0.0627*zn4 +3.2235*wn1 -
        3.9425*wn2 +2.2129*wn3 -0.4940*wn4;
        output= wn; // filtered data
        //update filters? variables
        xn6=xn5;
        xn5=xn4;
        xn4=xn3;
        xn3=xn2;
        xn2=xn1;
        xn1=xn;

        yn6=yn5;
        yn5=yn4;
        yn4=yn3;
        yn3=yn2;
        yn2=yn1;
        yn1=yn;
        zn4=zn3;
        zn3=zn2;
        zn2=zn1;
        zn1=zn;
        wn4=wn3;
        wn3=wn2;
        wn2=wn1;
        wn1=wn;
        return output;
}
voiduart_write_char(uint8_t output)
{
        while(!(UCSR0A &(1<<UDRE0)))
        UDR0 = output;
}
void delay_100us(int n) //delay 100us
{
        inti,j;
        for (j=0; j<n; j++)
        {
                for (i=0; i<100; i++)
                {
                        asm("nop");
                }
        }
}
```