Heart-Monitor

by CardioTech

Bandara D.M.D.V.
Undergraduate (Biomedical engineering)
Department Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

Jayalath K.A.D.M.P.
Undergraduate (Biomedical engineering)
Department Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

Pushpakumara H.M.R.M.
Undergraduate (Biomedical engineering)
Department Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

Tilakarathna U.A.
Undergraduate (Biomedical engineering)
Department Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

February 28, 2023

Contents

Abstract

Our project is a heart monitor that outputs and displays the ECG waveform. The challenge is to amplify the small ECG voltage signal of range 0.1mV - 10 mV which gets largely affected by noise. To achieve this a leg drive circuit, an instrumentation amplifier, a high pass filter, a low pass filter and a notch filter were used. The bandwidth considered was 160 Hz. To achieve this, only analog electronic components were used. Digital electronic components were used only in the digital circuit. Simulation testing, breadboard testing, and PCB testing was done to test the functionality of the circuit. A 3D enclosure was designed and built to suit the device.

1 Introduction

An electrocardiogram (ECG) is a simple test that can be used to check the heart's rhythm and electrical activity through sensors attached to the skin, which detects electrical signals generated by the heart each time it beats. The voltage between right arm and left arm is amplified while having a feedback mechanism through the right leg. These voltages of right arm, left arm and right leg are taken from ECG leads (Figure 1). The amplitude of the voltage signal has a range of 0.1 mV-10 mV; while its frequency falls into 0.01 Hz-250 Hz. Amplifying this signal becomes a challenge due to the small amplitude of the ECG raw signals and their ease of corruption due to noise and disturbances such as power line interference, electrostatic potentials, RFI, electrode contact noise, stray capacitance, nearby electronic devices and also bio signal artifacts introduced into an ECG by subject movement, respiration and muscle tension. In certain situations, noise can override the ECG waves and make the amplified signal useless.

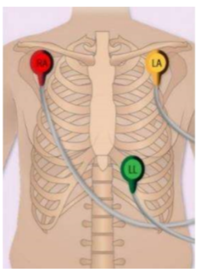


Figure 1: 3 Lead ECG placement

Our project is an ECG device, which amplifies and displays the above mentioned voltage waveform. We only used analog electronic equipment except for the display unit. In this report we have explained how we achieved this. For the ECG waveform generation, we have used below mentioned circuits to achieve the final output.

1. Right Leg Drive Circuit.
2. Instrumentation Amplifier.
3. 3^{rd} Order active Butterworth high pass filter.
4. 5^{th} Order active Bessel Thompson low pass filter.
5. Notch Filter.

And two separate circuits, with a PCB for each was used for power supply and display screen.

In each filter circuit, we have used variable resistors to adjust the gains, adjust and reduce noise and disturbances depending on the location, and unstable power supplies. The total approximate gain achieved through adjustments in the circuit is 1000.

2 Design Specifications

2.1 Filter Specifications

The ECG falls onto the frequency range of 0.01 Hz - 250 Hz. The American Heart Association (AHA) recommends a minimum bandwidth of 150 Hz for children between the ages of 12 to 16 years; and a minimum bandwidth of 125 Hz for adults. To filter out the signal to this range from the noise and other disturbances, three filters are used. The high pass filter cut off frequency we used was 0.05 Hz and the low pass filter cut off frequency was 160 Hz. Conductive gel between leads and the body also supports minimizing noise.

The other main issue is the power line interference. To avoid this a notch filter of 50 Hz is used in the circuit.

The overall gain of the ECG waveform should be above 1000 as per the standards. Therefore, the filter gain is made such that this goal is achieved. Variable resistors are used in the filters to increase and decrease the gain as necessary.

2.1.1 Op amp Selection

To reduce the high noise added to the small raw ECG signal, an op amp with high CMRR is used in the instrumentation amplifier. Generally, the

CMRR should at least be above 100dB, therefore, we used the LM4562 opamp which has a typical CMRR of 120dB.

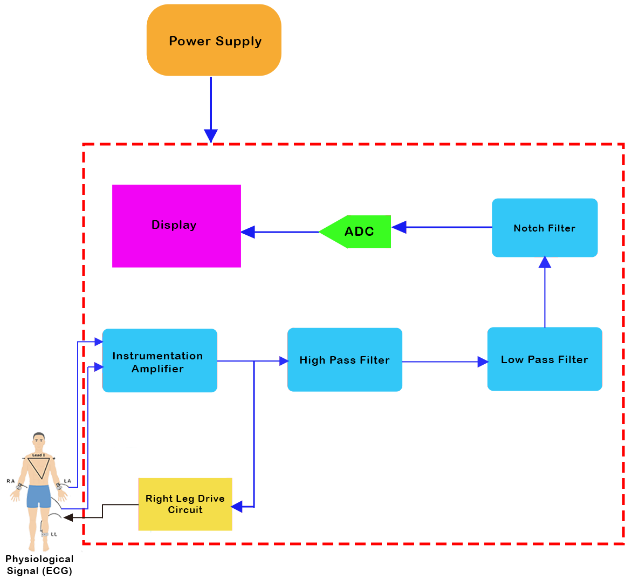
2.2 Sampling Rate

The sampling ADC sampling rate chosen was 15,000 samples per second. This is the maximum sampling rate of the ADC of the Atmega328pU chip.

3 Functionality Description

3.1 ECG Circuit

Block Diagram



3.1.1 Right Leg Drive Circuit

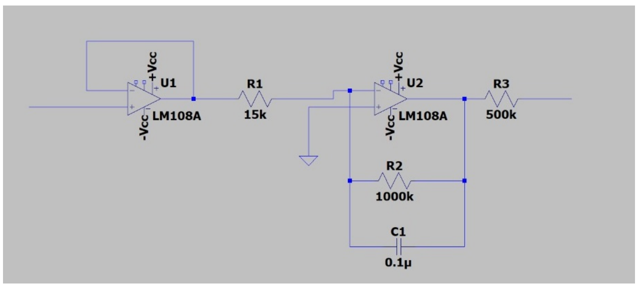


Figure 2: Right Leg Drive

A right leg drive circuit is used to cancel the common-mode signal between the left and right arm electrodes by inverting, amplifying, and then feeding the signal back to the body through the right leg electrode.The high value resistors used in this circuit also ensures that the patient connected does not get in contact with a bigger current.

3.1.2 Instrumentation amplifier

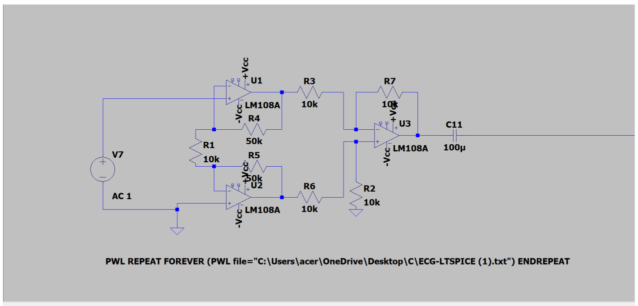


Figure 3: Instrumentation amplifier

This circuit is used to amplify the signal, while removing the common noise of the voltages from ECG leads. The three inputs are converted to the single output through this circuit. The gain of our circuit in the instrumentation amplifier is kept in the range 20-50. We have used a variable resistor at R1(Refer figure 1) to adjust the gain as necessary.

3.1.3 High Pass Filter

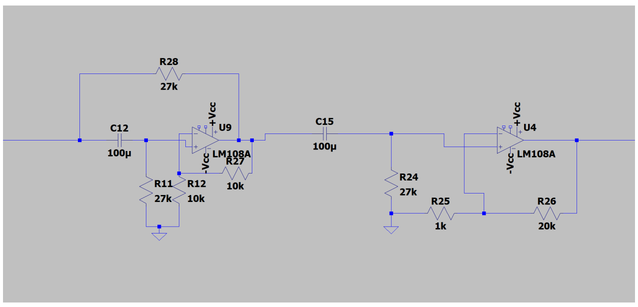


Figure 4: High Pass Filter

Here we have used 3rd order active inverting Butterworth filter. This consists of a cascaded second order active non inverting filter and a first order active inverting filter. This circuit eliminates the DC offset developed between the electrodes, and also amplifies the signal further. The gain of the second order filter is 2, and the minimum gain of the first order filter connected is 20, making the minimum overall gain of the high pass filter $20 \times 2 = 40$. The gain of the first order filter can be further increased when required through the variable resistor connected.

3.1.4 Low Pass Filter

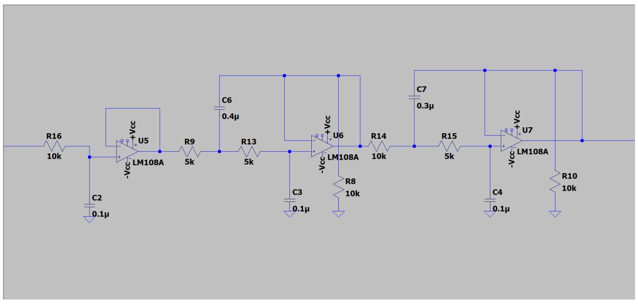


Figure 5: Low Pass Filter

Here, we used a 5^{th} order Bessel Thompson non inverting filter. Compared to the Butterworth filter, since the group delay and the signal distortion caused is absent, we chose this filter. This consists of a series cascaded first order active filter, and two second order active filters. The gains of each are unity, causing the overall gain of the low pass filter a unity. No adjustments to the gain are made, and therefore, we didn't use a variable resistor here.

3.1.5 Notch Filter

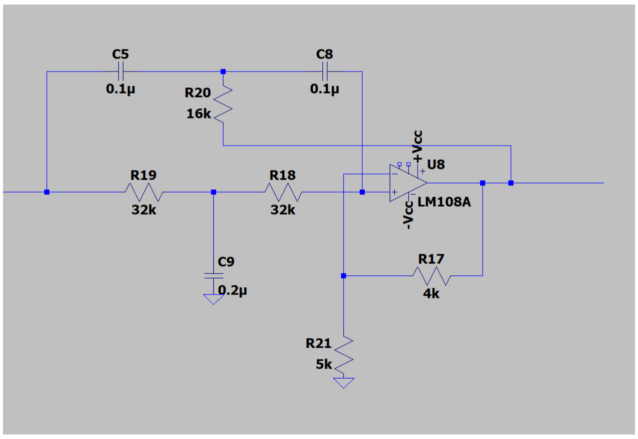


Figure 6: Notch Filter

Here we have used a Twin-T type notch filter. This is used to eliminate the power line interference. Therefore the notch filter frequency is 50 Hz. This requires a small transition bandwidth or high Q factor to achieve the steeper roll-off. A Twin-T notch filter is one of the few RC networks capable of providing an infinite deep notch at a particular frequency. Therefore, we used this particular filter. The Q factor, or else known as the quality factor determines the quality of the filter. To adjust this as necessary, a variable resistor is used as. The notch frequency depends on three resistor values. To adjust that when necessary, three tuners are used.

3.2 Display Circuit

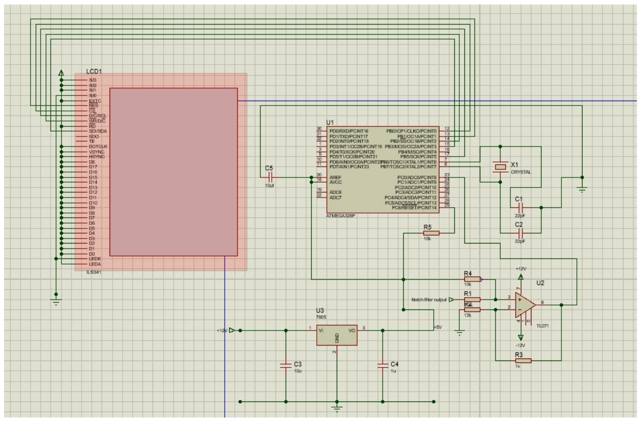


Figure 7: Display

The output from the notch filter is directed through an Atmega328pU chip to a display to plot the ECG waveform and display it in real time. In the circuit, we bring down the 12V supply to 5V (Using a L7805 regulator) to power the Atmega chip and the display. Another 2-pin outlet is also designed in here to power the fan. The code is provided in the **appendix (cite)**

3.3 Power Supply Circuit

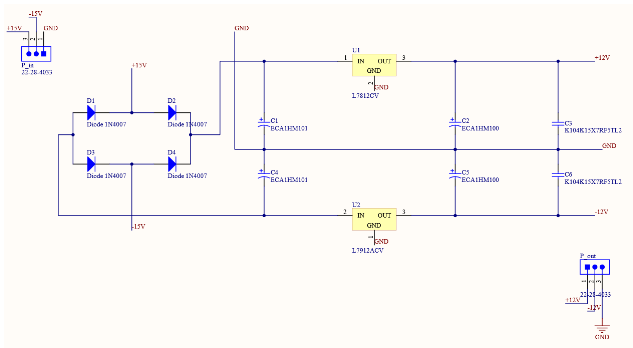


Figure 8: Power Supply

As we need a power supply of +12V, 0V and -12V s for the functionality of the op amps, we built a separate circuit to provide the required voltages. Here, the AC voltage of household of 230 V, is sent through a center tap transformer that outputs two AC signals of 30 V peak to peak value. These two signals are sent through rectifier diode circuit to make the signals DC, and then through 7812 to regulate the DC voltage to +12 V and then through 7912 to regulate -12 V.

4 Design Parameters

- **Gain resistor value of instrumentation amplifier:**

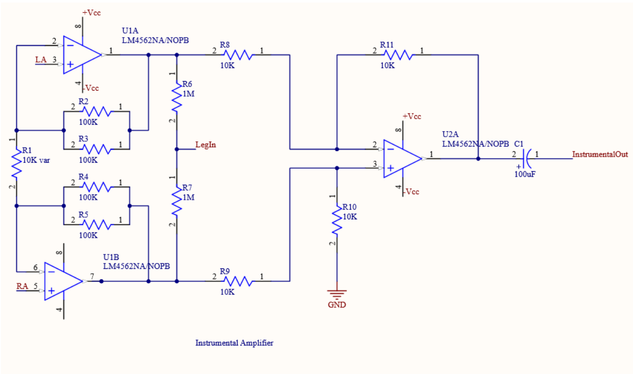


Figure 9: Instrumental amplifier

$$Gain \; = \left(\frac{R10}{R9}\right)\left(\frac{2R + R1}{R1}\right)$$
$$(\frac{10k\Omega}{10k\Omega}(1 + \frac{2 \times 50k\Omega}{R1}$$

$$Here, \;\; R = R4//R5 = R2//R3$$

$$Minimum\ Gain = 1 + \frac{2 \times 50\ k\Omega}{10k\Omega}$$
$$= 11$$

We can increase this by changing the value of the R1 variable resistor, and get the gain to the required range of 20 - 50.

- **The cut off frequency of the high pass filter:**

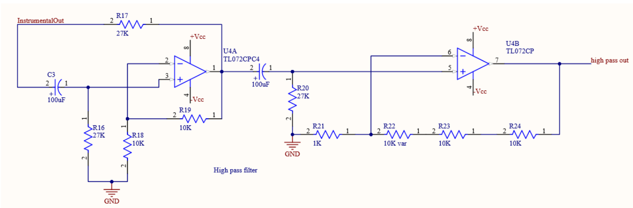


Figure 10: High pass filter

$$f_n = \frac{1}{2\pi RC} = \frac{1}{2\pi \times R16 \times C3}$$
$$= \frac{1}{2\pi \times 27k\Omega \times 100\mu F}$$
$$= 0.0589Hz$$

The resistor and capacitor values were taken such that the required cut off frequency can be achieved through the available components in the market.

- **Gain of the high pass filter:**

consider, Gain 2 order = Gain of the second order active filter

Gain 1 order = Gain of the first order active filter

$$Gain \;\; = Gain\ 2\ order \;\; \times Gain\ 1\ order$$

$$Gain\ 2\ order \;\; = 1 + \frac{R19}{R18}$$
$$= 1 + \frac{10k\Omega}{10k\Omega}$$
$$= 2$$

$$Gain\ 1\ order = 1 + \frac{R22 + R23 + R24}{R18}$$

This can be changed by varying the value of the $10k\Omega$ variable resistor at R21.

$$Maximum\ gain = 1 + \frac{R22 + R23 + R24}{R21}$$
$$= 31$$

- **Cut off frequency of the low pass filter:**

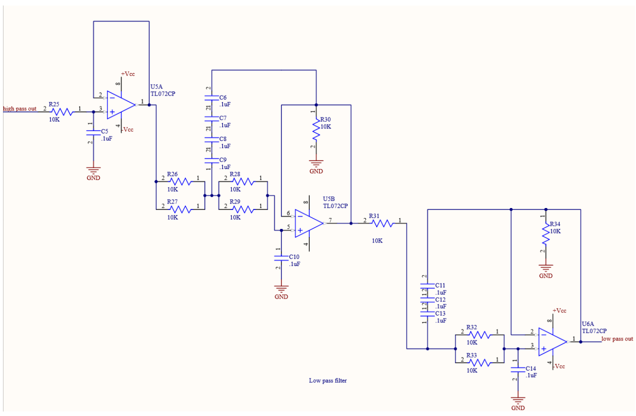


Figure 11: low pass filter

$$f_n = \frac{1}{2\pi RC}$$
$$= \frac{1}{2\pi \times R25 \times C5}$$
$$= \frac{1}{2\pi \times 10k\Omega \times 0.1\mu F}$$
$$= 159.154Hz$$

- **Cut off frequency of the notch filter**

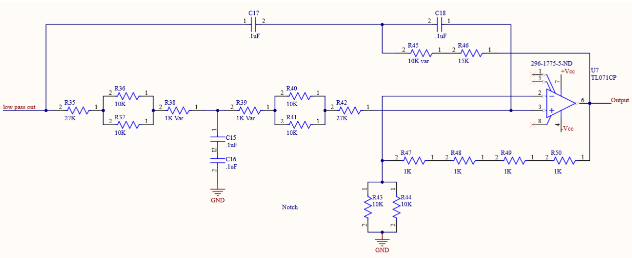


Figure 12: low pass filter

$$f_n = \frac{1}{2\pi RC}$$

$$Here,\ R = 27k\Omega + 10k\Omega//10k\Omega + 1k\Omega(variable)$$

$$f_n = \frac{1}{2\pi \times 32k\Omega \times 0.1\mu F}$$
$$= 49.73Hz$$

The capacitor values and resistor values were chosen as available market values. Three variable resistors were used at R38, R39 and R45 to maintain the Twin T configuration, by mitigating the effect of tolerances.

The other variable resistor at R48, was used to adjust the gain.

- **Gain of the notch filter:**

$$\begin{aligned} Gain &= 1 + \frac{R47 + R48 + R49 + R50}{R43//R44} \\ &= 1 + \frac{4k\Omega}{5k\Omega} \\ &= 1.8 \end{aligned}$$

5 PCB Design and Soldering

Three PCBs, namely a power circuit, a heart monitor circuit, and a display circuit, were designed using Altium Designer for printing. The ECG PCB is of 4 layers (**Figure 13**). The 3rd layer carries +12V and -12V power supply routings and the Top and bottom layers are used for routing. Power supply PCB(**Figure 14a**) was routed in a single layer but printed in 2 layer(Remaining layer grounded) to reduce the capacitance building up. Ground pours are poured to both single layers and the remaining layer to reduce further noise. Except for the display circuit (**Figure 14b**), the rest has been imported, and the display PCB is etched. All of them are soldered with relevant components; all the ICs are coupled with IC bases for easy replacement; and interconnections with accessories outside the circuit have been done with pin headers and jumpers.

6 Enclosure Design

The enclosure was designed by solidworks (**Figure 15**).

It contains a removable top lid, and a removable front. The removability helps to fix and replace components. The holes surrounding the box ensures that the heat flow to the outside of the box, and thus the components do not get damaged. A fan is also connected inside to help this.

The front contains the display and also the knobs for tuning the filters. The switch on top switches off the device, and can be used for temporary situations. And the switch at the back, switches off the transformer as well, saving the power loss that happens at the transformer. The enclosure also contains holes containing the adapters to connect the power cable and also to connect the 3.5mm jack.

7 Simulation and Testing

We first simulated our entire circuit using LTSpice and the display using Proteus(as shown in **Figure 7**) Then it's implemented on a breadboard with minor adjustments to obtain a signal with the least amount of noise possible. From the changes in osciloscope display, we can see the overall improvement from the testing. (**Figure 17 & 18**)

8 Results

Final product is shown below.

Though we faced some issues with display, the final result from the osciloscope is as shown in **figure 19**.

9 Task allocation

Index No.	Contribution
200061N	circuit and PCB designing
200247P	Circuit designing, Display and coding
200488E	Circuit and PCB designing
200664P	Circuit and solidwork(enclosure) designing

References

I Du WY, Jose W. Design of an ECG sensor circuitry for cardiovascular disease diagnosis. Int J Biosen Bioelectron. 2017;2(4):120–125. DOI: 10.15406/ijbsbe.2017.02.00032

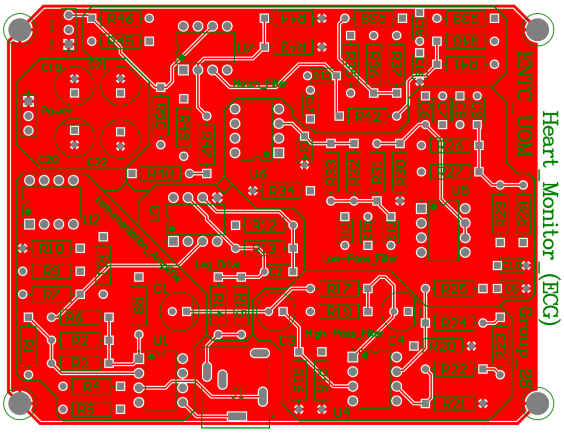
II Article : Stack_ exchanege/choosing-the-right-op-amp-for-ecg-application

III Article: Robu.in/Dual Power Supply

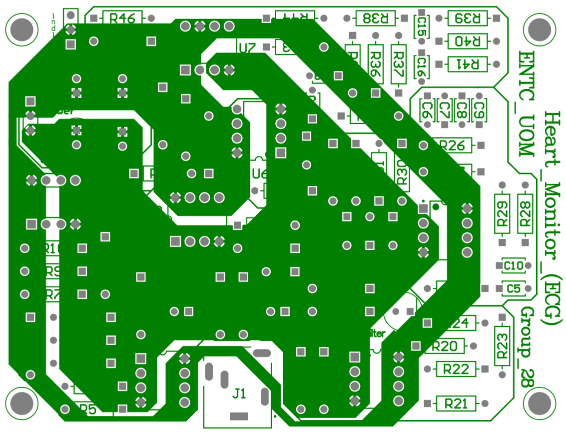
IV Display Code Reference: github.com/KrisKasprzak

Appendix I: PCB's and Enclosure design

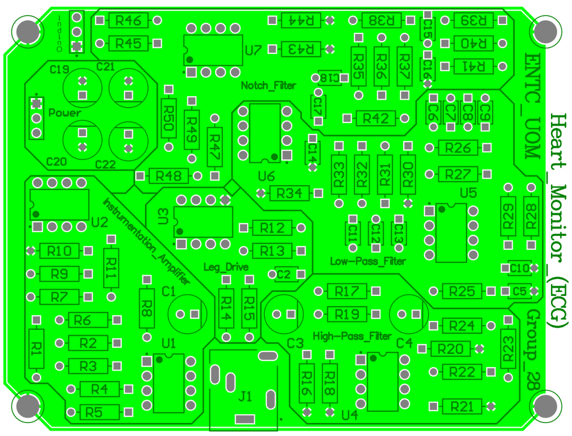
PCB Design with Altium designer



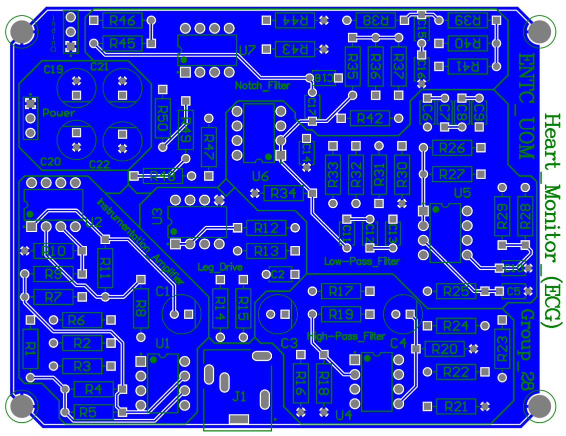
(a) Top layer



(b) Power layer

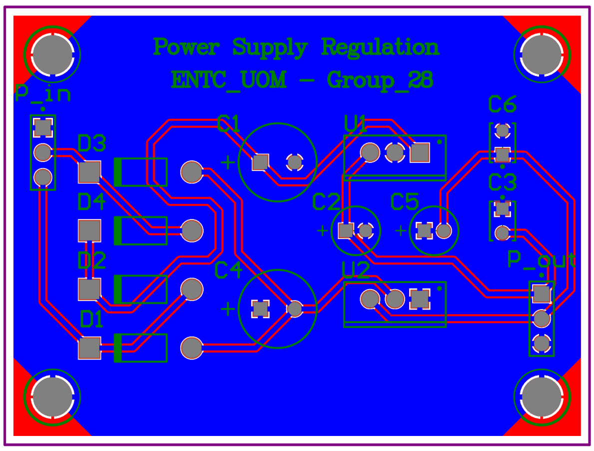


(c) Ground

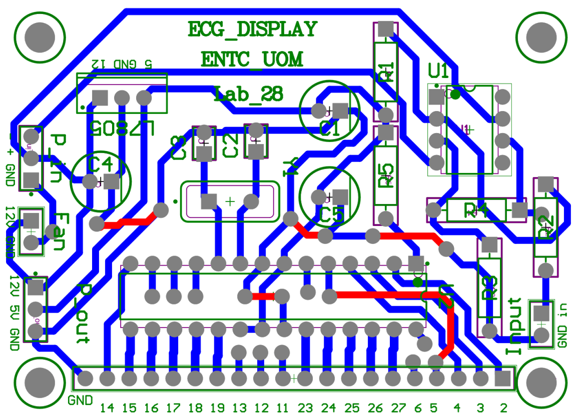


(d) Bottom Layer

Figure 13: ECG Main circuit



(a) power supply



(b) Display

Enclosure with solidworks

Figure 15: enclosure

Appendix II: Simulation and Testing

LT spice simulation of the circuit

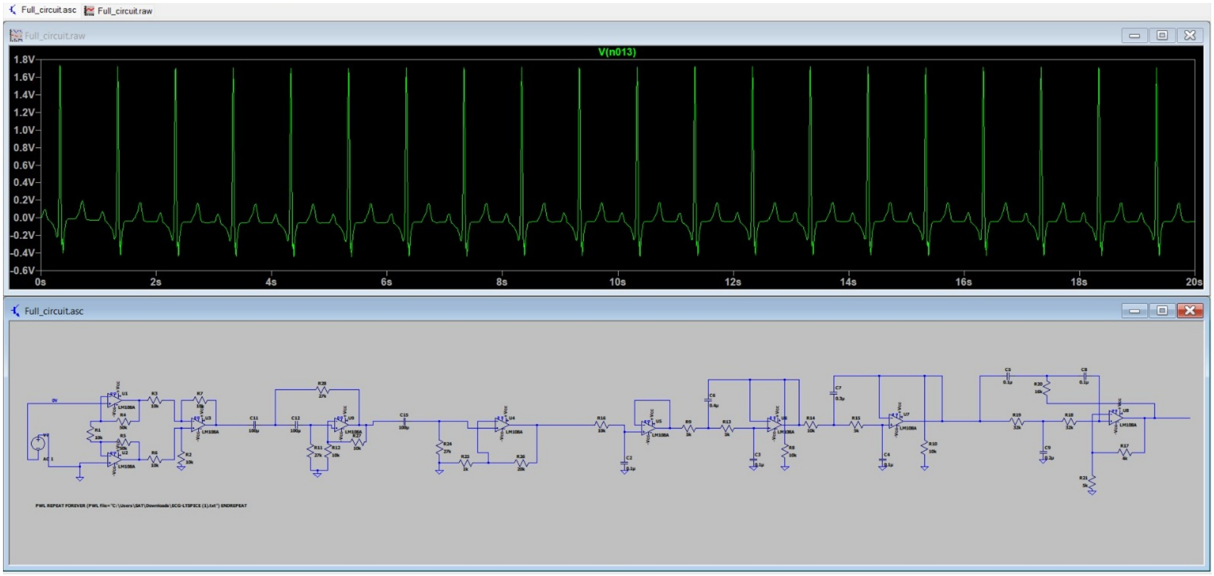


Figure 16: LT-Spice simulation

Breadboard implementation and testing

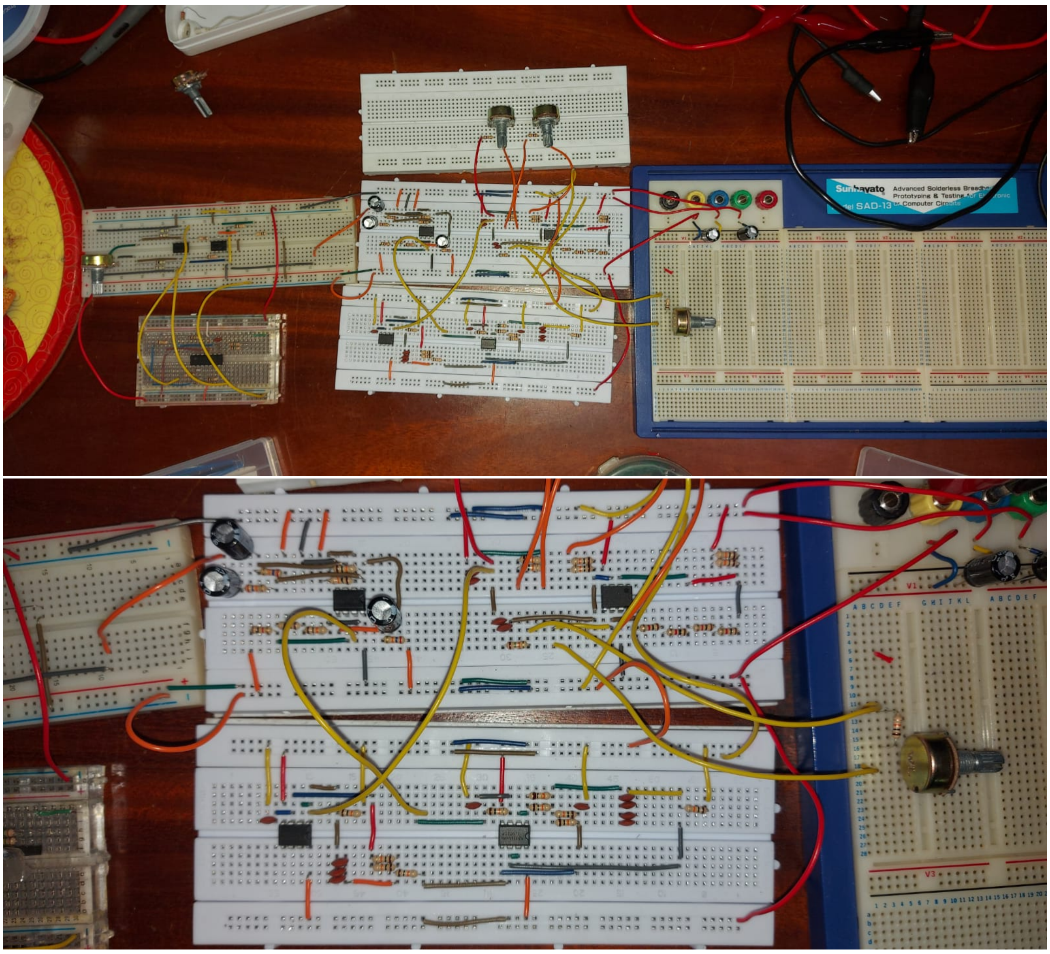


Figure 17: Breadboard

Results from Breadboard implementation and tuning

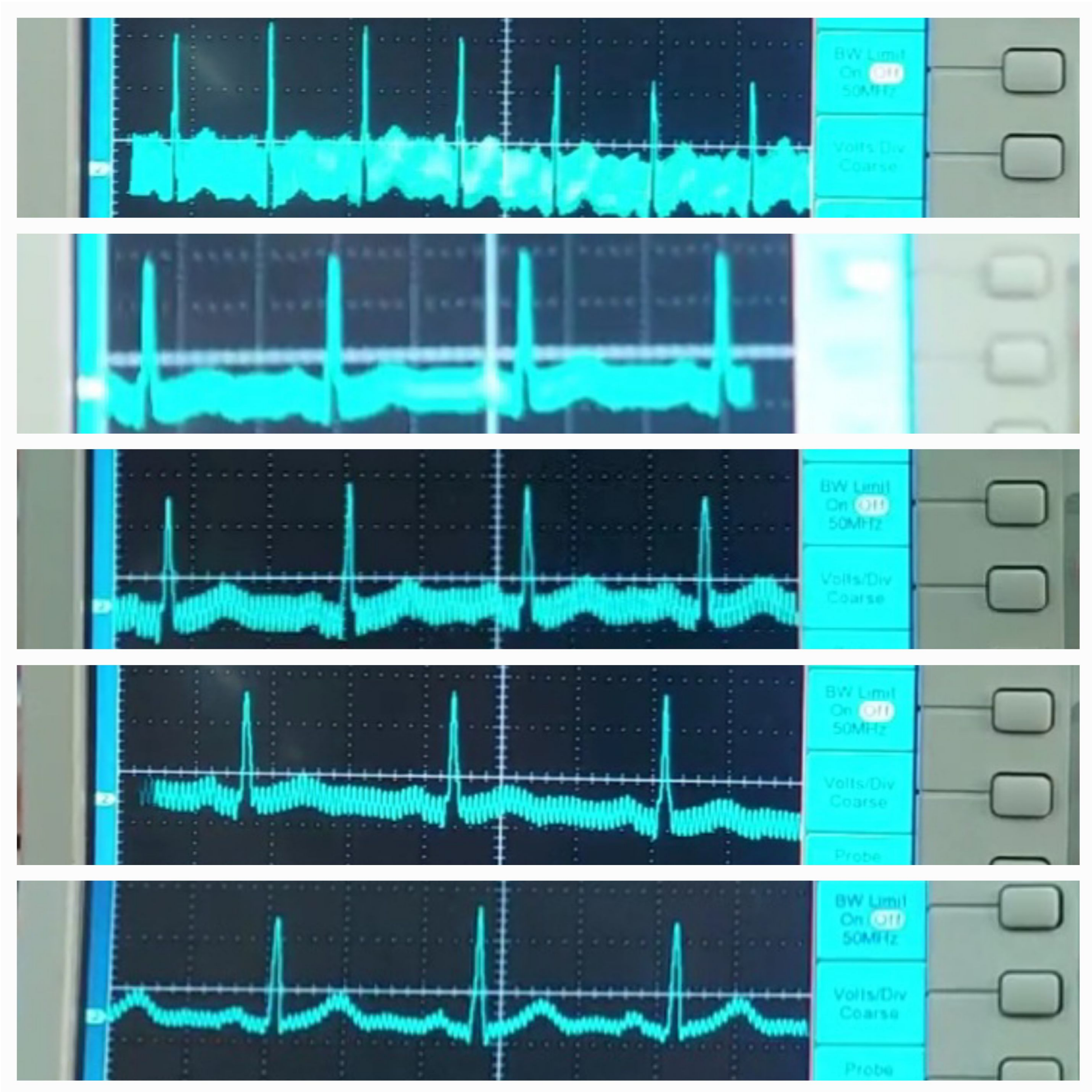


Figure 18: Osciloscope readings from breadboard tuning

Final Result After finalizing the Product.

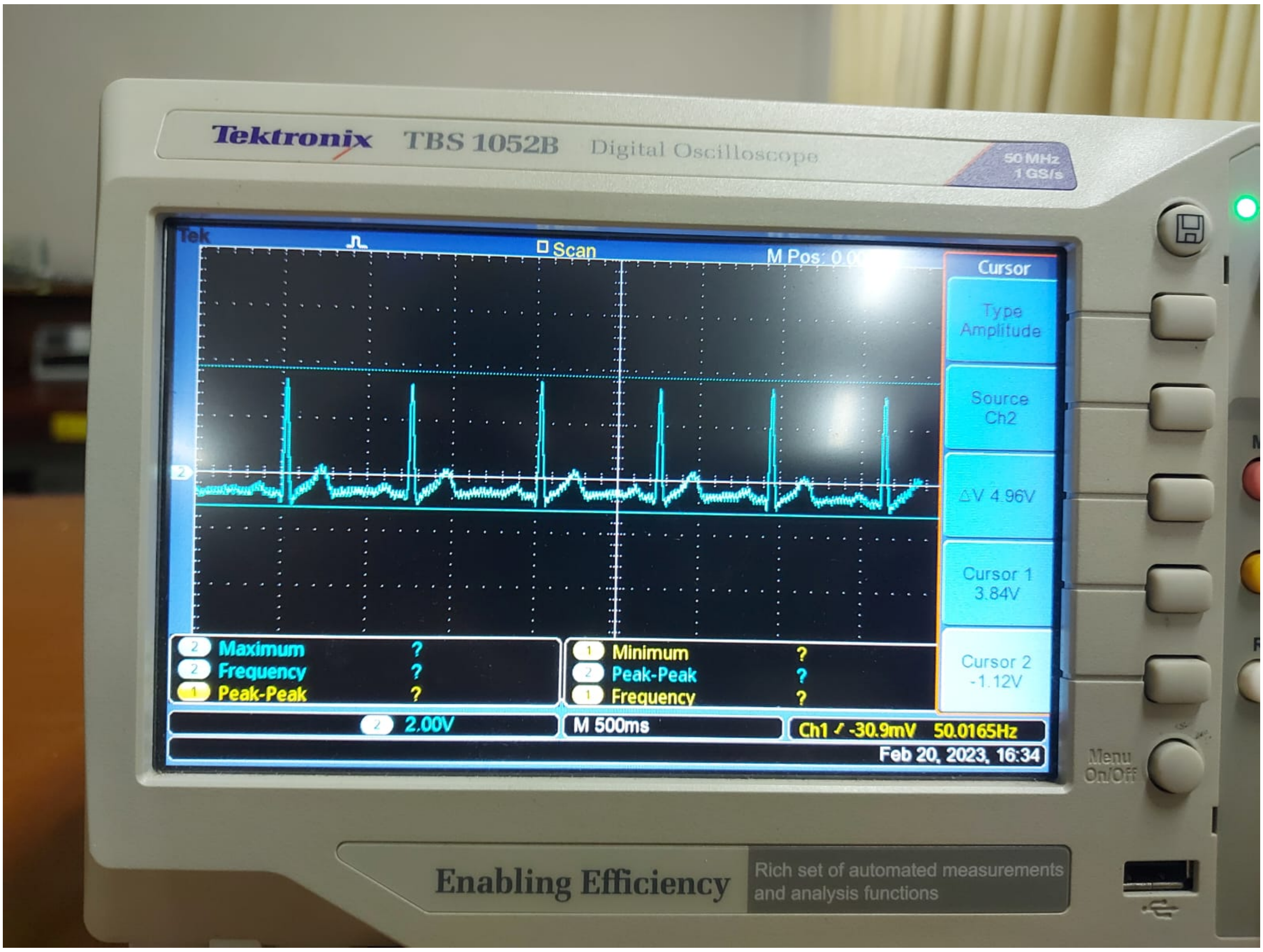


Figure 19: Final Output

Appendix III: Display Code

```
#include <Adafruit_GFX.h>
#include <Adafruit_TFTLCD.h>

#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4
#define ADJ_PIN A5

boolean display1 = true;
boolean display2 = true;
boolean display3 = true;
boolean display4 = true;
boolean display5 = true;

double vo;
double ox , oy;

Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

void Graph(Adafruit_TFTLCD &d, double x, double y, double gx, double gy,
double w, double h, double xlo, double xhi, double xinc, double ylo,
double yhi, double yinc, String title, String xlabel, String ylabel,
unsigned int gcolor, unsigned int acolor, unsigned int pcolor,
\unsigned int tcolor, unsigned int bcolor, boolean &redraw) {

  double ydiv, xdiv;
  // initialize old x and old y in order to draw the first point
  of the graph
  // but save the transformed value
  // note my transform funcition is the same as the map function,
  except the map uses long and we need doubles
  //static double ox = (x - xlo) * ( w) / (xhi - xlo) + gx;
  //static double oy = (y - ylo) * (gy - h - gy) / (yhi - ylo) + gy;
  double i;
  double temp;
  int rot, newrot;

  if (redraw == true) {

    redraw = false;
    ox = (x - xlo) * ( w) / (xhi - xlo) + gx;
    oy = (y - ylo) * (gy - h - gy) / (yhi - ylo) + gy;
    // draw y scale
    for ( i = ylo; i <= yhi; i += yinc) {
      // compute the transform
      temp =  (i - ylo) * (gy - h - gy) / (yhi - ylo) + gy;

      if (i == 0) {
        d.drawLine(gx, temp, gx + w, temp, acolor);
      }
```

```
      else {
        d.drawLine(gx, temp, gx + w, temp, gcolor);
      }

      d.setTextSize(1);
      d.setTextColor(tcolor, bcolor);
      d.setCursor(gx - 40, temp);
      // precision is default Arduino--this could really use some format control
      d.println(i);
    }
    // draw x scale
    for (i = xlo; i <= xhi; i += xinc) {

      // compute the transform

      temp =  (i - xlo) * ( w) / (xhi - xlo) + gx;
      if (i == 0) {
        d.drawLine(temp, gy, temp, gy - h, acolor);
      }
      else {
        d.drawLine(temp, gy, temp, gy - h, gcolor);
      }

      d.setTextSize(1);
      d.setTextColor(tcolor, bcolor);
      d.setCursor(temp, gy + 10);
      // precision is default Arduino--this could really use some format control
      d.println(i);
    }

    //now draw the labels
    d.setTextSize(2);
    d.setTextColor(tcolor, bcolor);
    d.setCursor(gx , gy - h - 30);
    d.println(title);

    d.setTextSize(1);
    d.setTextColor(acolor, bcolor);
    d.setCursor(gx , gy + 20);
    d.println(xlabel);

    d.setTextSize(1);
    d.setTextColor(acolor, bcolor);
    d.setCursor(gx - 30, gy - h - 10);
    d.println(ylabel);


  }

  //graph drawn now plot the data
  // the entire plotting code are these few lines...
  // recall that ox and oy are initialized as static above
  x =  (x - xlo) * ( w) / (xhi - xlo) + gx;
  y =  (y - ylo) * (gy - h - gy) / (yhi - ylo) + gy;
  d.drawLine(ox, oy, x, y, pcolor);
```

```
  d.drawLine(ox, oy + 1, x, y + 1, pcolor);
  d.drawLine(ox, oy - 1, x, y - 1, pcolor);
  ox = x;
  oy = y;

}

/*
  End of graphing functioin
*/

void setup() {
  Serial.begin(9600);
  tft.begin(0x9341);
  tft.fillScreen(BLACK);
  tft.setRotation(1);
  pinMode(ADJ_PIN, INPUT);

  for (int x = 0; x < 60; x += 1) {
    vo = analogRead(ADJ_PIN) / 204.6;
    Graph(tft, x, vo, 50, 290, 390, 260, 0, 60, 10, 0, 5, 1, "ECG MONITOR",
    " Time [s]", "Heart rate", DKBLUE, RED, GREEN, WHITE, BLACK, display1);

    delay(100);
  }

  delay(1000);
  tft.fillScreen(BLACK);

  for (int x = 0; x < 60; x += 1) {
    vo = analogRead(ADJ_PIN) / 204.6;
    Graph(tft, x, vo, 50, 290, 390, 260, 0, 60, 10, 0, 5, 1, "ECG MONITOR",
    " Time [s]", "Heart rate", DKBLUE, RED, GREEN, WHITE, BLACK, display2);

    delay(100);
  }

  delay(1000);
  tft.fillScreen(BLACK);   for (int x = 0; x < 60; x += 1) {
    vo = analogRead(ADJ_PIN) / 204.6;
    Graph(tft, x, vo, 50, 290, 390, 260, 0, 60, 10, 0, 5, 1, "ECG MONITOR",
    " Time [s]", "Heart rate", DKBLUE, RED, GREEN, WHITE, BLACK, display3);

    delay(100);
  }

  delay(1000);
  tft.fillScreen(BLACK);

  for (int x = 0; x < 60; x += 1) {
    vo = analogRead(ADJ_PIN) / 204.6;
    Graph(tft, x, vo, 50, 290, 390, 260, 0, 60, 10, 0, 5, 1, "ECG MONITOR",
    " Time [s]", "Heart rate", DKBLUE, RED, GREEN, WHITE, BLACK, display4);

    delay(100);
```

```
  }

  delay(1000);
  tft.fillScreen(BLACK);

  for (int x = 0; x < 60; x += 1) {
    vo = analogRead(ADJ_PIN) / 204.6;
    Graph(tft, x, vo, 50, 290, 390, 260, 0, 60, 10, 0, 5, 1, "ECG MONITOR",
    " Time [s]", "Heart rate", DKBLUE, RED, GREEN, WHITE, BLACK, display5);

    delay(100);

    /* code repeat similarly for 30 more segments chznging only the variable
    name of "display#"/*
  }

  delay(1000);
  tft.fillScreen(BLACK);
  }
void loop(void) {
}
\end
```