Lab 3: Portable EKG Amplifier with WiFi Functionality

Susan Gilbert, Stewart Holloway, Shalini Ranmuthu, Priyanka Vaidya
University of Texas at Austin, BME 374L
4/19/16

Table of Contents

I. INTRODUCTION

In this lab we created a portable WiFi enabled EKG monitor. To do so, we designed a printed circuit board (PCB) which takes inputs from the electrodes attached the chest. Its output goes to a web browser which we also developed to display 30 seconds of EKG data.

Designing a portable EKG came with its own set of challenges: there was much less area to lay components out, and so we had to be very careful in how we designed our amplifier. It was similar in idea to the biomedical amplifier we made in lab 1, however since the EKG in this case was battery operated we did not have to electrically isolate the patient side of the portable amplifier. Additionally, we had to work with lower power, which means a lower headroom for signal. Furthermore, we had to bias the incoming signal appropriately so that it fell within a voltage range appropriate for our IC's. The details of this work are described in the following report.

II. CIRCUIT DESIGN

A. Circuit Schematic

The circuit schematic was generated using DipTrace software and is reprinted in a landscape orientation below in order to make all the components visible (Fig. 1). The patterns for all components besides the resistors and capacitors were provided to us as a separate library and imported to build the schematic.

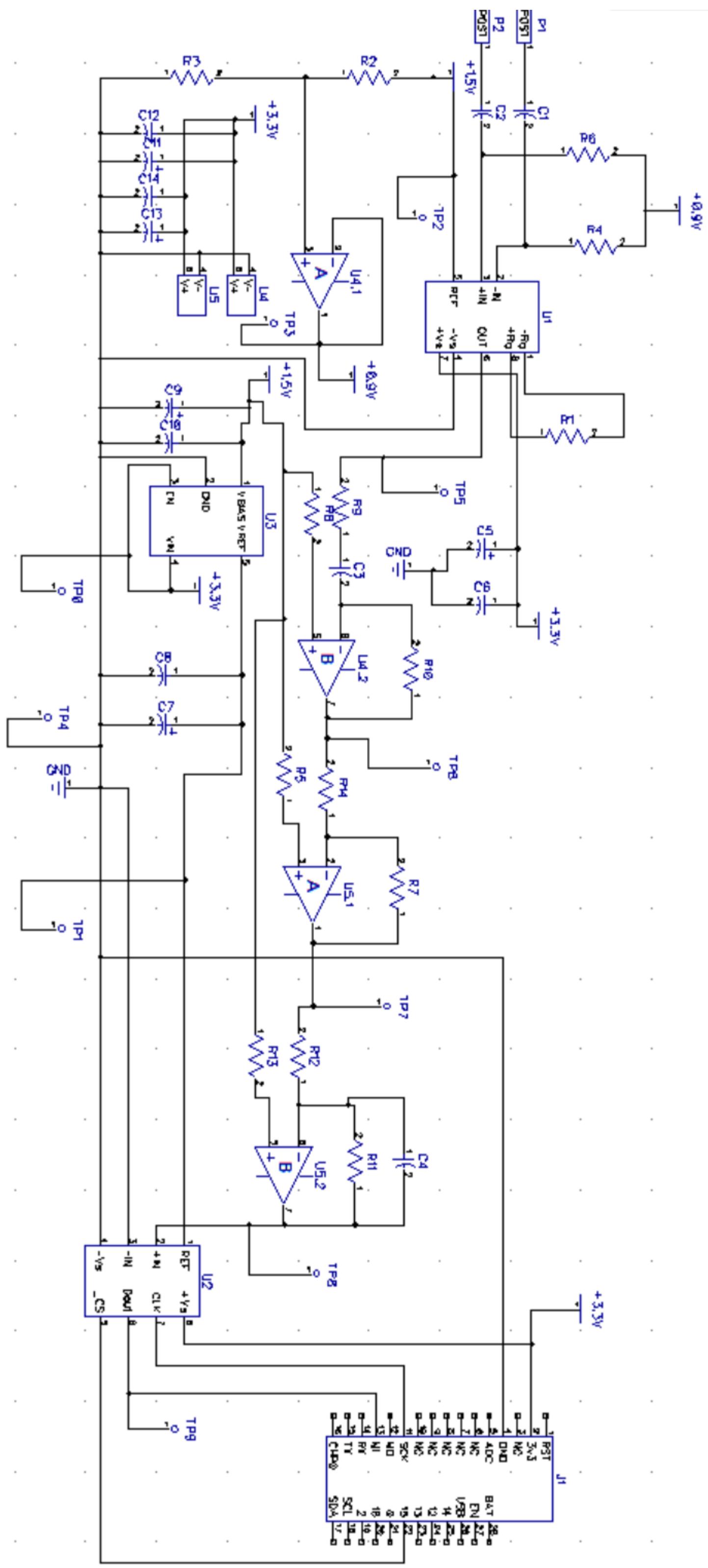


Figure 1: Circuit Schematic

B. Description

1. Instrumentation Amplifier (U1)

The INA we used was an AD623AN chip from Analog Devices, which we set up using a single-supply configuration.[1] The inputs from the two electrodes were connected to the binding posts, which were then connected to the inverting and non-inverting terminals of the INA (pins 2 and 3). We used a gain resistor of R1 = 11k in order to achieve a theoretical gain of 10.1, which was calculated using the given gain equation of 1+(100k/Rg). The INA has a reported CMRR of about 100dB at a gain of 10. The supplied +Vs and -Vs (pins 7 and 8) are 3.3V and ground, respectively (all other components were also set up using the same single-supply setup which is described in section 2). The two inputs to the INA were offset by a 0.9V supply generated by a voltage divider (discussed later) to bias the input towards the midpoint of the allowed 0.15-1.5V input range. Passive high pass filters are present at the input terminals of the INA in order to remove DC bias that might move our signal outside of our INA's voltage range. These filters used C1 and C2 of 1uF and R6 and R4 of 1M, giving a cutoff of 0.159 Hz. The output from pin 6 is biased around a virtual ground reference of 1.5V that is supplied to pin 5. The resulting signal can then be processed and better utilize the full 0-3V range of the ADC if it is centered around a virtual ground 1.5V since this allows for bipolar signal swing.

2. Power Supply and References (U3, U4.1)

The Huzzah board (discussed later) is connected to a 3.7V Lithium ion polymer battery, and pins 4 and 2 provide ground and 3.3V supplies for the entire PCB. [5] This 3.3V supply is used as +Vs for the INA, the ADC, and the op-amps. It is also connected to the Vin supply and enable pins (pins 4 and 3) of the REF1930 3V reference chip (U3).[2] The ground pin (pin 2) of the reference chip is connected to ground. The chip generates a Vref of 3V at pin 5 which is used as the voltage reference for the ADC. The chip also generates a Vbias of 1.5V (half of Vref) at pin 1 which is used as the virtual ground reference for the INA, op-amp filters, and gain stages.

A voltage divider was also created to divide down the 1.5V supply from the REF1930 to create the 0.9V supply used to bias the electrode inputs to the INA. An R2 of 1k and an R3 of 1.5k were used to divide down 1.5V to 0.9V. A gain-1 buffer was also created using one of the TLV2462IP op-amps (U4.1) so that the voltage divider sees a high input impedance at the next stage. This helps ensure that the voltage generated is in fact 0.9V (no leakage). The TLV2462IP chips are manufactured by Texas Instruments and contain two op-amps per chip that share the same +Vs and -Vs.[3] The gain-one buffer configuration involves connecting the output to the inverting input, and connecting the desired 0.9V to the non-inverting input so that the output will be this same value.

3. High Pass Filter (U4.2)

The output of the INA was input to a high pass filter. We designed an active high pass filter using the other half of the TLV2462IP mentioned previously (U4.2) in an inverting configuration. In our case, pin 5 was the positive input, pin 6 was the negative input, and pin 7 was the output. We used a 1M resistor for both R9 and R10 giving us a gain of 1, where gain is calculated as the ratio of R10/R9. The cutoff frequency was 0.159 Hz (same as before) since our capacitor value was 1 uF and our resistor value was 1M. The cutoff frequency can be calculated as $fc = 1/2\pi RC$. The bias resistor, R8, was connected to the non-inverting terminal and was made to be equal to R10 and R9 in parallel which is 499k. The other side of the bias resistor was connected to the virtual ground reference of 1.5V.

4. Gain Stage (U5.1)

The output of the high pass filter was input to a gain stage. We designed our gain stage using the first half of a new TLV2462IP chip (U5.1) in an inverting configuration. In our case, pin 3 was the positive input, pin 2 was the negative input, and pin 1 was the output. We used a 82k resistor for R7 and a 1k resistor for R14 giving us a gain of 82, where gain is calculated as the ratio of R7/R14. The bias resistor, R5, was connected to the non-inverting terminal and was made to be equal to R7 and R14 in parallel which is 1k. The other side of the bias resistor was connected to the virtual ground reference of 1.5V.

5. Low Pass Filter (U5.2)

The output of the gain stage was the input for our low pass filter. We designed an active low pass filter using the other half of the TLV2462IP mentioned previously (U5.2) in an inverting configuration. In our case, pin 5 was the positive input, pin 6 was the negative input, and pin 7 was the output. R11 and R12 were both 11k so the gain was 1 where gain can be calculated as the ratio of R11/R12. The cutoff frequency we had was 308 Hz since our capacitor was 0.047 uF and our resistor was 11 KOhms. The cutoff frequency can be calculated as $fc = 1/2\pi RC$. The bias resistor, R13, was connected to the non-inverting terminal and was made to be equal to R11 and R12 in parallel which is 4.99k. The other side of the bias resistor was connected to the virtual ground reference of 1.5V.

The overall gain of the signal path should therefore be 10.1 (INA) * 82 (gain stage) = 828. This is equal to 20log(828) = 58.4 dB gain.

6. ADC and Huzzah (U2, J1)

The output of the low pass filter was input to a successive approximation register ADC. We used the MCP3201 chip from Microchip (U2) which has a 12-bit resolution and SPI serial interface.[4] The supplied +Vs is 3.3V from the Huzzah board and -Vs is connected to ground (pins 8 and 4). The sampling rate is reported as 50 samples per second (ksps) for a +Vs of 2.7V and 100 ksps for a +Vs of 5V, so our sampling rate can be linearly approximated as 63 ksps for our +Vs of 3.3V. The non-inverting input is connected to the output of the low pass filter while the inverting input is connected to ground (pins 2 and 3). A reference voltage of 3V was provided to pin 1 from the REF1930 chip. A voltage range of 3V divided by 2^12 bits gives a resolution of 0.73mV. Pins 5-7 of the MCP3201 set up the serial interface from the IC. Pin 5 is the serial clock, pin 6 is the serial data output line, and pin 7 is the chip select line.

The Huzzah board is a programmable modular component that facilitates WiFi data transfer.[5] It is essentially an onboard antenna module attached to a PCB that allows for additional troubleshooting functionality, as well as pin control for analog input, GPIO, UART, and power inputs. We used the output from the MCP3201 ADC to sync the clock (pin 11) and provide an input to the MISO input of the Huzzah board (pin 13). The chip select from the ADC was also connected to GPIO 15 (pin 22). A 105mAh 3.7V battery was attached to the Huzzah board (not shown in the schematic), which allowed the Huzzah board to provide the 3.3V and ground supplies for the PCB (pins 2 and 4).

7. Bypass Capacitors and Test Points

Bypass capacitors of 1uF and 10pF were used to stabilize the +Vs lines for the op-amps and INA, as well as the 3V Vref and 1.5V Vbias supplied by the REF1930. The two different values of bypass capacitors help dampen AC ripple or noise across a range of frequencies in the DC +Vs supply. The 10pF attenuates the higher frequency noise and the 1uF attenuates the lower frequency noise. The capacitors were placed as close to the ICs as possible in order to minimize the line inductance and series resistance between the bypass capacitor and the IC, with the smaller capacitor being placed closer to the IC (pin 1).

Test points (TP) were added to supply lines and key points in the signal processing pathway in order to be able to troubleshoot any problems with the circuitry. A table of test points and their respective purposes can be seen in Table 1.

Table 1: List of Test Points and their purpose

Test Point	**Purpose**	**Test Point**	**Purpose**
TP0	3.3V	TP5	INA output
TP1	3V	TP6	HPF output
TP2	1.5V	TP7	Gain stage output
TP3	0.9V	TP8	LPF output
TP4	ground	TP9	ADC output

C. PCB Layout

The printed circuit board (PCB) was generated by exporting the schematic to the layout functionality of the DipTrace software (Fig. 2). A two layer board was generated, with the top layer containing all of the traces and the bottom layer being a copper pour that was connected to the ground net. All ground connections were made using a via to this copper pour plane. No 90 degree trace angles were used when components were routed in order to avoid issues with etching of corners and resonant properties. The final size of the PCB was 3” x 3”. After conducting a design rule check within DipTrace, we exported the final PCB files (NC Drill and Gerber files) and verified them at http://www.freedfm.com. After we passed this additional design check, we emailed the files to Kaarthik and he had the PCB made for us.

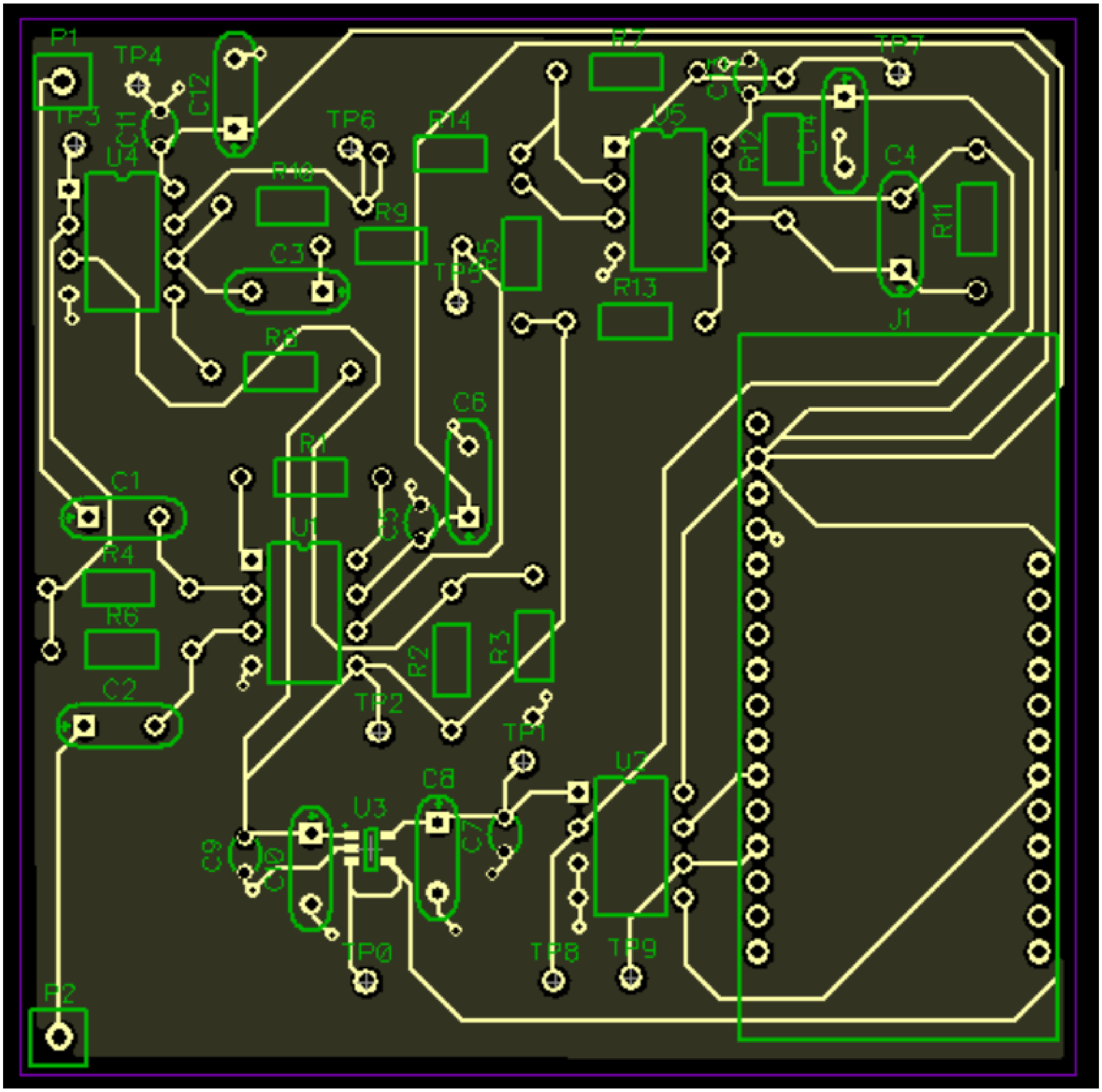


Figure 2: PCB layout (board outline in purple, top silkscreen in green, top traces in white, and bottom copper pour in gray)

D. Bill of Materials

The total cost of the components needed for our PCB is calculated in Table 2 to be $43.23. In addition to the cost of the circuit components, we used Vermed electrodes purchased from medexsupply.[6] Two electrodes are needed per subject. When bought in bulk, the cost of a single electrode is approximately 17 cents. Furthermore, two snap leads were required to connect the electrodes to the PCB board and can be purchased from any distributor at a cost of approximately $3.[7] The PCB board, minus all of the circuit components, cost approximately $33 to manufacture, bringing the total cost of our prototype to about $83. If produced in bulk, this cost could be reduced significantly.

Table 2: Bill of Materials

Name	Value	Quantity	RefDes	Manufacturer	Unit Price	Total Price	Distributor
CAP200	1uF	8	C1, C2, C3, C6, C8, C10, C12, C14	Nichicon	0.26	2.08	Digikey
CAP200	0.047uF	1	C4	Kemet	0.24	0.24	Digikey
CAP1000AP	10pF	5	C5, C7, C9, C11, C13	Vishay BC Components	0.2	1	Digikey
RES400	11k	3	R1, R11, R12	Stackpole Electronics Inc.	0.1	0.3	Digikey
RES400	1k	3	R2, R5, R14	Stackpole Electronics Inc.	0.1	0.3	Digikey
RES400	1.5k	1	R3	Stackpole Electronics Inc.	0.1	0.1	Digikey
RES400	1M	4	R4, R6, R9, R10	Stackpole Electronics Inc.	0.1	0.4	Digikey
RES400	82k	1	R7	Stackpole Electronics Inc.	0.1	0.1	Digikey
RES400	499k	1	R8	Yageo	0.1	0.1	Digikey
RES400	4.99k	1	R13	Stackpole Electronics Inc.	0.1	0.1	Digikey
HUZZAH		1	J1	Adafruit	15.95	15.95	Adafruit
BIND_POST		2	P1, P2	Keystone Electronics	0.36	0.72	Digikey
AD623AN		1	U1	Analog Devices	5.21	5.21	Digikey
MCP3201		1	U2	Microchip	2.28	2.28	Microchip
REF1930		1	U3	Texas Instruments	2.8	2.8	Digikey
TLV2462IP		2	U4, U5	Texas Instruments	2.8	5.6	Digikey
Li-Polymer battery	105mAh 3.7V	1		PKCELL	5.95	5.95	Adafruit
Total						43.23	

III. MEASUREMENTS AND ANALYSIS

A. Bode plot of ECG Amplifier

1. Testing Setup

To acquire the Bode plot of the ECG amplifier, one input terminal of the board was connected to AI0 of the ELVIS board, and the other input terminal was connected to the ground of the ELVIS board. We then used the Bode plotter VI provided by Labview to collect our data with an input amplitude of 3 mV, because this is the maximum amplitude we would expect from our EKG. The rest of the settings can be found below (Fig. 3).

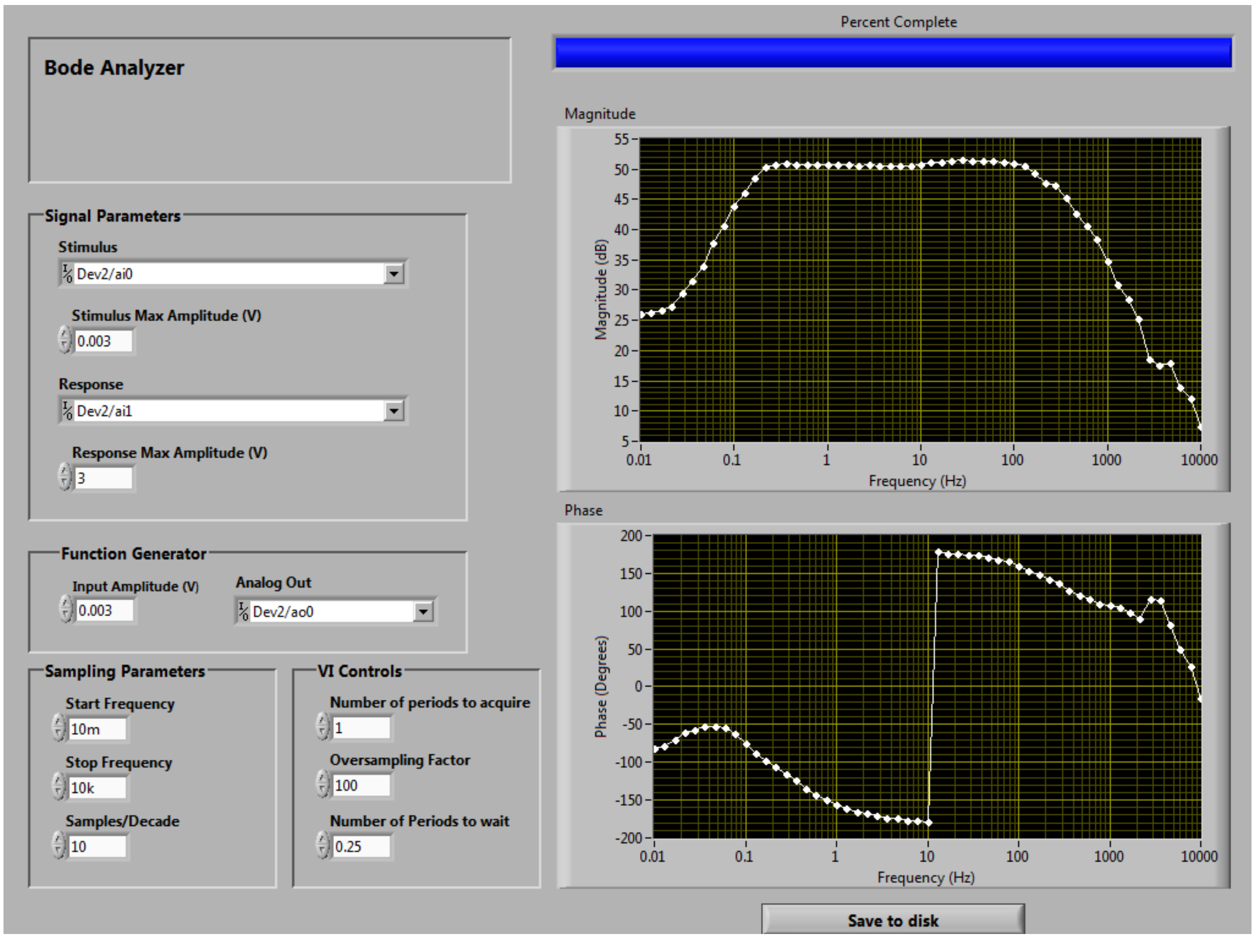


Figure 3: LabView Software Settings for collection of Bode Plot data

2. Results

Below we have shown our bode plot results for frequencies between 10^-2 and 10^4 Hz (Fig. 4). As you can see, our gain in the passband (approximately 50 dB) is close to our expected 58.4 dB (gain of 828). Furthermore, we see -3dB corner frequencies at 0.25 Hz and 300 Hz, based on the points where our gain drops below 47 dB. We expected our cutoffs to be 0.159 Hz and 307.8 Hz, so these values are very close. With single pole filters, we expect to see a -20 dB/decade drop off, which is confirmed when we see the ~30 dB gain at 1000 Hz and 10 dB at 10,000 Hz.

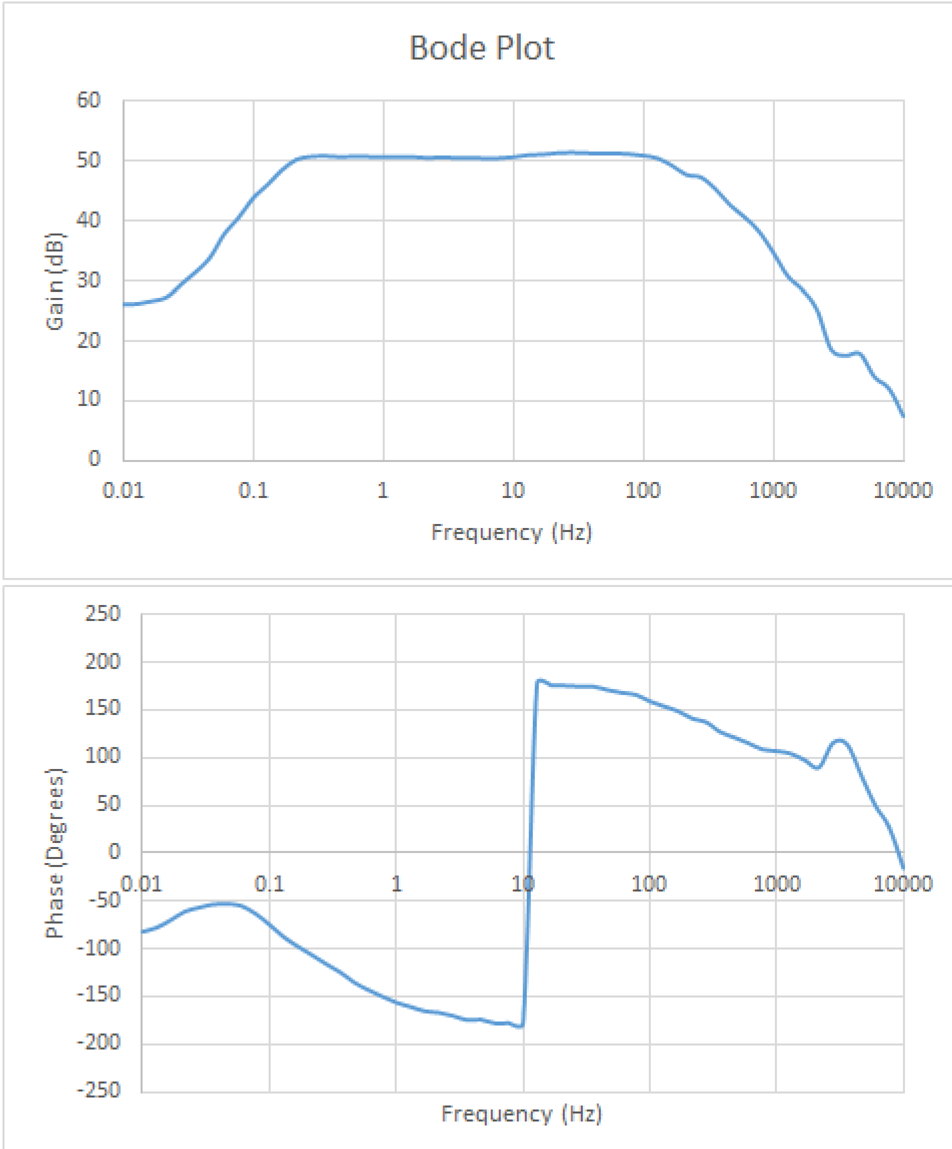


Figure 4: Bode Plot of portable EKG device: Magnitude and Phase

The theoretical filter responses for the low pass and high pass filters can be found below (Fig. 5). By comparing these theoretical graphs with the actual measured graphs above, you can see how our instrument's frequency response is very similar to the superposition of the two theoretical graphs in both magnitude and phase response. Inserted below is the MATLAB code used to generate the theoretical Bode plots in Figure 6.

```
%%FilterPlotter
%Prepared by Stew Holloway 021716
```

```
%Theoretical Bode Plots
R = 11e3;
C = .047e-6;
H_LPF = tf(1, [2*pi*R(1)*C 1]);

figure(1);
bode(H_LPF);
title('LPF Theoretical Bode Plot');

%Theoretical Bode Plots
R = 1e6;
C = 1e-6;
H_HPF = tf([2*pi*R(1)*C 0], [2*pi*R(1)*C 1]);

figure(2);
bode(H_HPF);
title('HPF Theoretical Bode Plots');
```

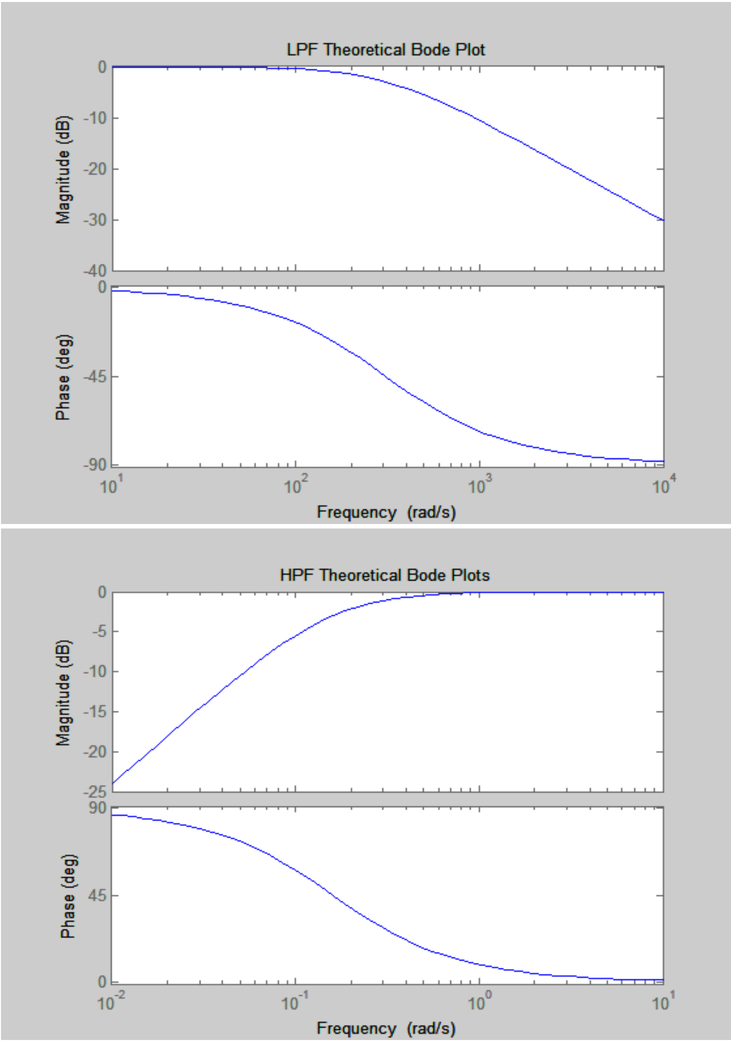


Figure 5: Theoretical Bode Plots of HPF and LPF (made in MATLAB)

B. ECG amplitude vs. time plots

Capacitive coupling with environmental AC noise around us will create an unwanted 60 Hz potential in the order of volts, which is orders of magnitude higher than the biopotential we are planning to measure. Therefore, the test subject was asked to sit inside a Faraday cage to eliminate the unwanted noise. To obtain the ECG amplitude versus time plot, a lead II configuration was placed on the test subject. The positive lead electrode was placed on the left leg and the negative lead electrode was placed on the right shoulder. The positive lead was then attached to P2 of the board we designed, whereas the negative lead was attached to P1. The output of the board, TP8, was connected to the positive lead of AI0 on the ELVIS board. The negative lead of AI0 was connected to AIGND of the ELVIS board.

We designed a LabVIEW program using a Data Acquisition Module in order to record the signal from our biomedical instrumentation amplifier (Fig. 6). This program allows the user to enter the sampling rate and total recording time into the front panel, which also displays the signal output by the amplifier. We recorded 5 seconds of data at a sampling rate of 1000.

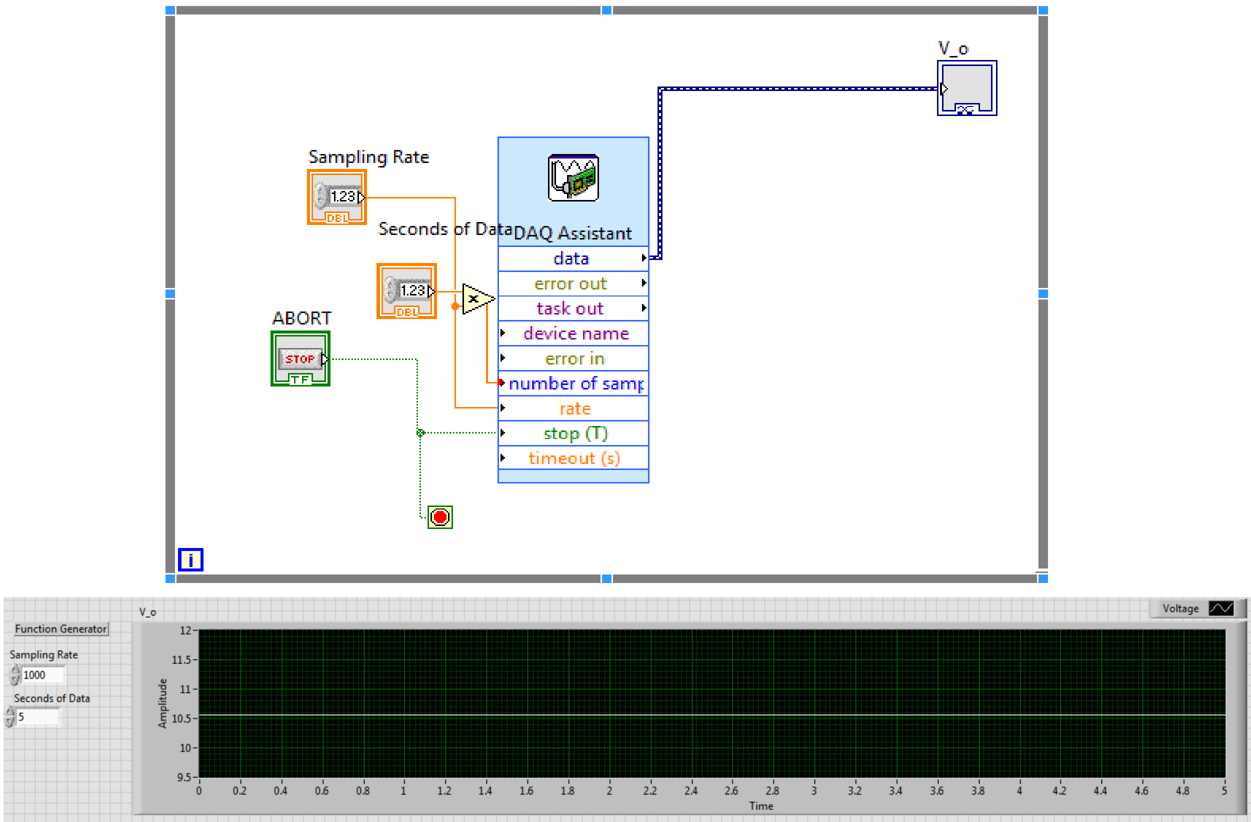
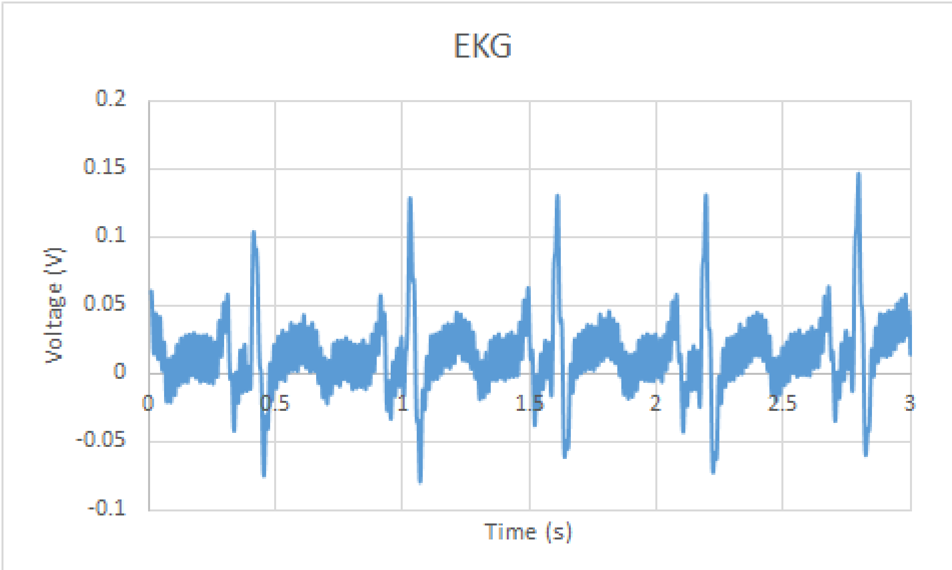


Figure 6: LabVIEW block diagram and front panel for EKG monitoring

The resulting waveform was exported and put through a simple averaging filter via Microsoft excel, where 0.02 seconds of data were averaged to create a single data point. This significantly cleaned up our signal and allowed us to see the important features of our waveform. However, the averaging filter has a tendency to attenuate signal and can eliminate features that are less than 0.02 seconds in length. The resulting waveforms can be found below (Fig. 7).



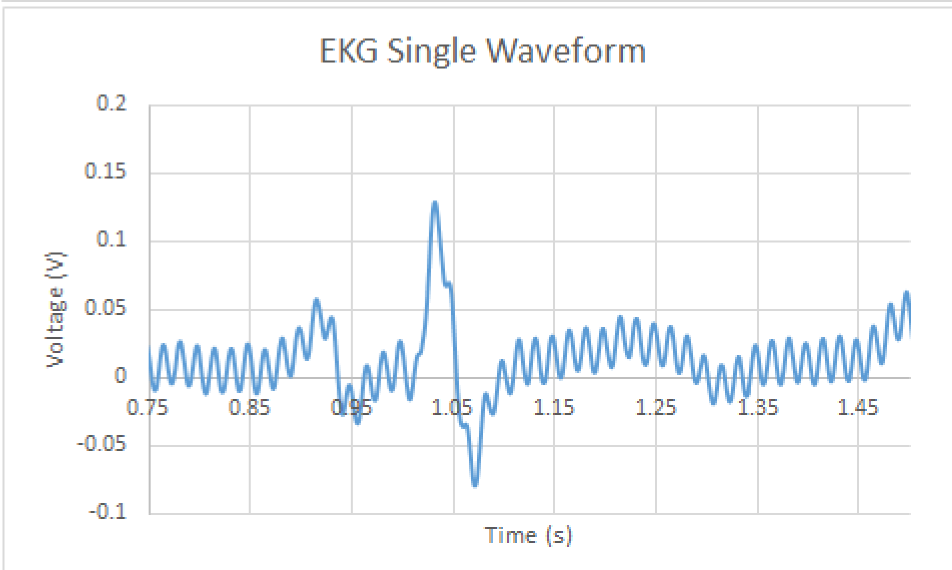


Figure 7: Signal averaged EKG waveform gathered from the ELVIS board

Since we measured Lead II (Left Leg to Right Arm), we expect to see a small P wave (atrial depolarization), a QRS complex (ventricular depolarization), and a T wave (ventricular repolarization). We can see all of these features in the signal above, albeit the signal is very noisy due to 60 Hz and other interference. The amplitude is not as large as we expected, since a 3 mV max amplitude should be amplified to approximately 2.5 Volts, however this lack of gain could be explained by attenuation of our averaging filter, issues in the manufacturing or soldering of our board, or even errors in electrode placement. The presence of the passive high pass filter at the inputs to our INA could also reduce the CMRR of our amplifier since they are likely not matched exactly (although they are very close). This could cause additional noise and attenuation.

C. Power Dissipation

To measure the power dissipation, a 1 ohm resistor was inserted in between the positive terminal of the battery and the positive terminal of the battery input on the Huzzah board, which had the unedited ESP8266_WebServer_Data_Logger.ino code loaded onto it (described further in the “Software” section). The voltage was measured across this resistor over a time interval of 10 seconds by connecting one end of the resistor to positive AI0 and the other end to negative AI0 on the ELVIS board and using the same LabVIEW program as above (Fig. 6). The raw data was exported to Excel and is depicted below (Fig. 8). We then converted voltage to current using ohm’s law (I = V/R). The formula for root mean square of a signal over a time interval is $f_{rms} = sqrt(1/(T_2 - T_1) * Integral(f(t)^2)dt$ where the integral is taken from T1 to T2. Using this formula where f(t) is the current recording over time, we calculate a root mean square current of 45.9 mA. This calculation was performed using Excel functions. Multiplying by the 3.7V voltage rating of the battery gives us a average power dissipation of 0.17 Watts (P = IV). Based on the capacity rating of the battery (105 mA*hrs.), we would expect our battery lifetime to be 105 mA*hrs /45.9 mA, or about 2 hours and 17 minutes.

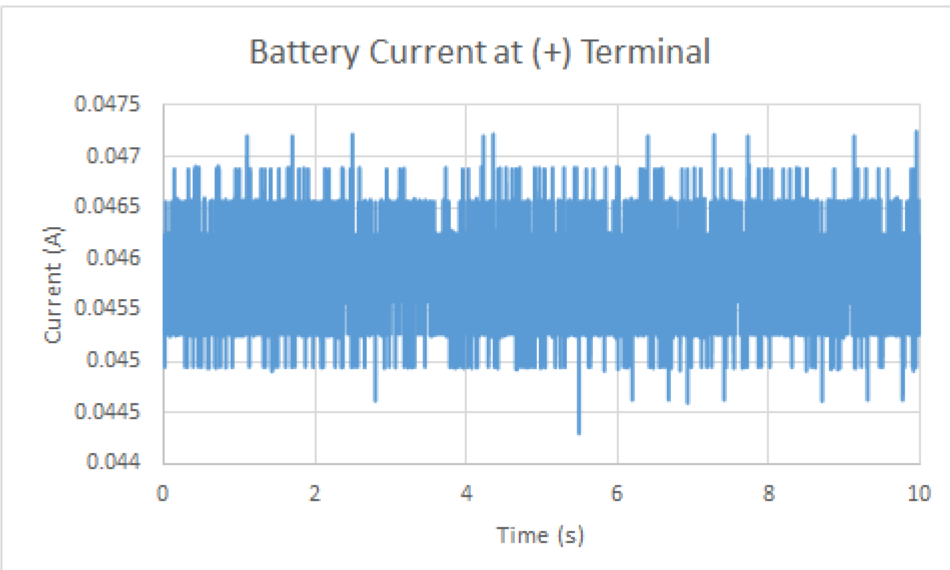


Figure 8: Raw Data for Battery Current Measurement

D. Noise Analysis

In order to record the noise, we connected both of the inputs of our board to AIGND on the ELVIS board. We also connected the output (TP8) to AI1 on the ELVIS board and recorded the output via the NI dynamic signal analyzer. The raw data as well as the settings on the dynamic signal analyzer can be seen below (Fig. 9). Next, we exported the data to excel and integrated over our bandwidth of interest (approximately 1 Hz to 300 Hz). This gave us a V^2 value of 0.079 V and a final measured thermal noise estimate of 0.282 V.

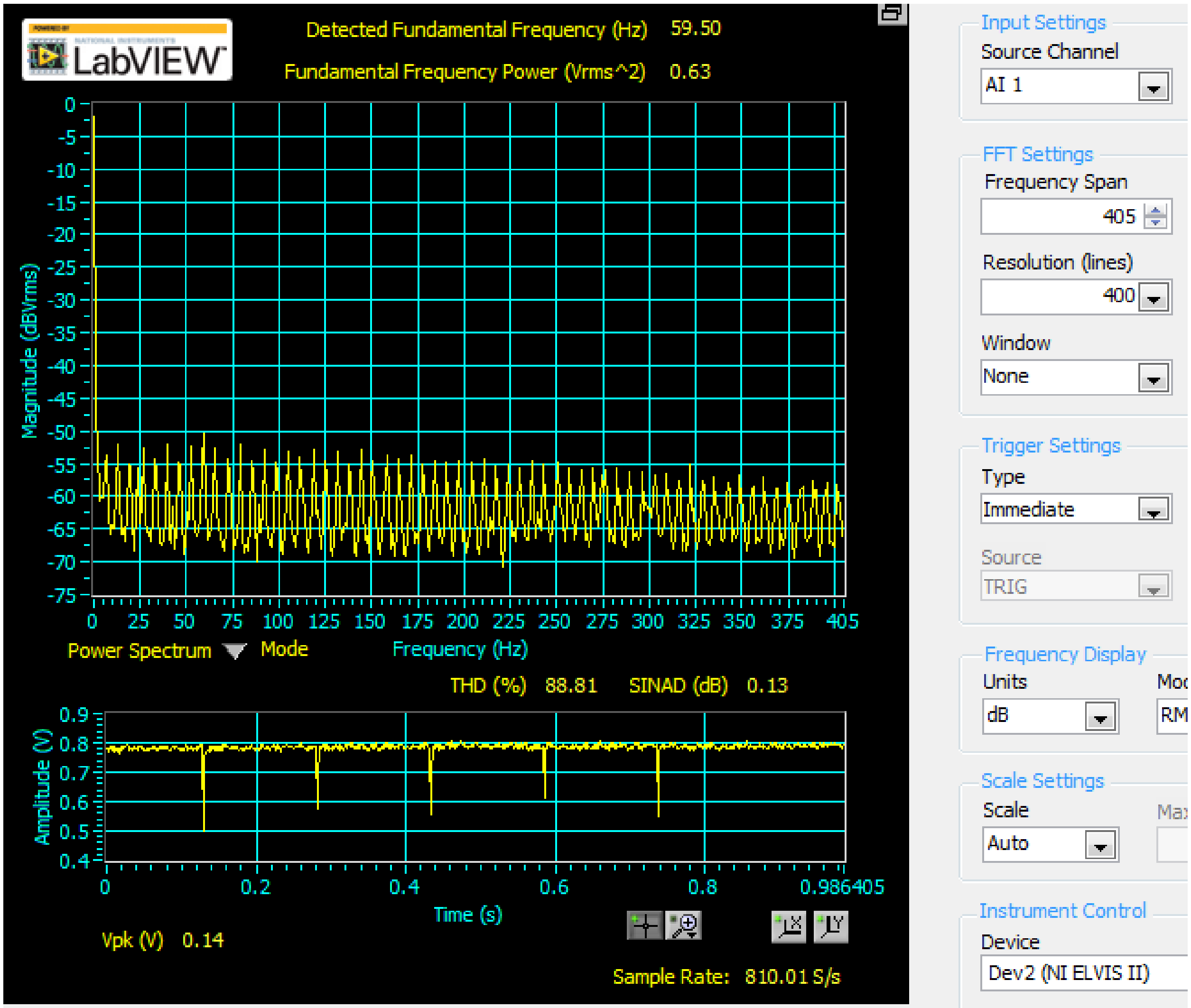


Figure 9: Dynamic Signal Analyzer Raw Output

We also calculated the theoretical thermal noise based on the components in our PCB. The handwork is detailed in the following scan (Fig. 10).

$$V_t^2 = \left[4KT\left(\Sigma R\right) + V_n^2 + i_n^2 \left(\Sigma R^2\right)\right] BW$$

$(1.602\times10^{-19})(273.15)$
$= 1.65\times10^{-20}$

INA 623:

Stage #1

$V_n = 35 nV/\sqrt{HZ}$ $\quad V_n^2 = 1.225\times10^{-15}$

$i_n = 1.5 pA\ p.p$ $\quad i_n^2 = 2.25\times10^{-24}$

$R = 1M + 1M + 11K \xrightarrow{R_{tot}} 2011000$ $\quad R^2 = 4.044121\times10^{12}$

HPF (TLV 24621P)

Stage #2

$V_n = 16 nV/\sqrt{HZ}$ $\quad V_n^2 = 2.56\times10^{-16}$

$i_n = .13 pA$ $\quad i_n^2 = 1.69\times10^{-26}$

$R = 499k + 1M + 1M \xrightarrow{R_{tot}} 2499000$ $\quad R^2 = 6.245001\times10^{12}$

G: 82 (TLV 24621P) " "

Stage #3

$V_n = 16 nV/\sqrt{HZ}$ " "

$i_n = .13 pA$

$R = 82k + 1k + 1k \xrightarrow{R_{tot}} 84000$ $\quad R^2 = 7056000000$

LPF: (TLV 24621P) " "

Stage #4

$V_n = 16 nV/\sqrt{HZ}$ " "

$i_n = .13 pA$

$R = 11k + 11k + 4.99k \xrightarrow{R_{tot}} 26990$ $\quad R^2 = 728460100$

$V_t^2 = [9.13678715\times10^{-12}] BW \times 82 \times 10.09 \rightarrow$ $\quad V_n \times G + V_n \times G$

$V_t^2 = [5.06136317\times10^{-13}] BW \times 82$

$V_t^2 = [1.07182517\times10^{-13}] BW \times 82$ $\quad BW = 307.843 - .159$

$V_t^2 = [4.72166098\times10^{-15}] BW \times 1$ $\quad = 307.684$

$\underline{V_t^2} \longrightarrow \sqrt{\ } \rightarrow \underline{V_t}$

Stage 1: $2.32576650\times10^{-6} \rightarrow 1.52511196\times10^{-3}$

Stage 2: $3.70959\times10^{-9} \rightarrow 6.0906\times10^{-5}$ $\quad \Sigma V_t \rightarrow 1.59315\times10^{-3}$

Stage 3: $4.443640\times10^{-11} \rightarrow 6.666063\times10^{-6}$

Stage 4: $2.1957744\times10^{-13} \rightarrow 4.685909\times10^{-7}$

$\boxed{\Sigma V_t = 1.59\times10^{-3}}$

Figure 10: Hand Calculations for our amplifier's theoretical thermal noise

We calculated a V^2 value of 2.52 x 10^-4 and a final theoretical thermal noise estimate of 0.00159V. These values are quite a bit lower than the actual measured values, but this is to be expected as the EKG signal that we gathered from our amplifier was quite a bit noisier than we expected it to be. This additional noise could come from 60 Hz or other coupled noise that was not rejected at the inputs of the INA, exacerbated by the fact that we had two analog filters placed before our INA. The components of these passive filters may have varied slightly from each other (thermally or just slight manufacturing variability), which would lessen the effective CMRR of our first stage. Furthermore, although we wouldn’t expect the 60 Hz interference to be as bad in a portable device because there is less capacitive coupling to the device itself, we recorded our EKG through the ELVIS board analog inputs, which are coupled to power line interference. We noticed that when our device is completely isolated from power lines (when we tested with a battery operated device sending WiFi to a server), there is significantly less noticeable noise.

IV. SOFTWARE

A. Software Setup

In order to make our data from the amplifier and the Huzzah module accessible via a web browser, we programmed the Huzzah with the “ESP8266_WebServer_Data_Logger.ino” file provided by Kaarthik Rajendran. We had some issues with the serial port/COM interface, as well as issues importing the “time_ntp.h” header file, however with Kaarthik’s helpful guidance we were able to compile the code successfully within the Arduino IDE and then load it onto the board. The serial interface was set up to work with a baud rate of 115200.

The software had to be tested at Shalini’s apartment where the WiFi protocol was more simple. On campus, the Huzzah board was not compatible with the campus WiFi setup, and setting up a virtual router on one of our laptops gave us issues because the IP was continuously changing as the connection failed and reset. At Shalini’s apartment, the Huzzah continued to stream to the same IP address even as the WiFi connection failed and reset.

B. Software documentation

In the interest of saving space, paper, and trees, we will document important sections of the code rather than the entire ESP8266_WebServer_Data_Logger.ino file.

The section below creates the global variables “ssid” and “password” that correspond to the router that the board will connect to. These need to be altered based on the WiFi that you would like to use.

```
// WiFi connection
const char* ssid = "bme374l";
const char* password = "rylander";
```

A global WiFiServer object (from the WiFiUdp.h library) is created, and the ADCword global variable is defined. Based on how the data is transferred from the Huzzah board, we must define two bytes (upperADC and lowerADC) in order to collect all of the bits before we shift and combine in order to form the full ADC word. Next, we edited the code to include a global variable “ADC_Voltage”, which can store the value after we convert from the ADC output to an actual voltage.

```
// Create an instance of the server on Port 80
WiFiServer server(80);

/////////////////////
// ADC Definitions //
/////////////////////
byte upperADC;
byte lowerADC;
int ADCword;
int ADC_Voltage;
```

The WiFiStart() function contains the code that sets up the WiFi connection. First, a WiFi.begin() function is called, and then the serial interface waits until a WiFi.status() flag is set to “Connected” status. Next, we use the WiFiServer object defined earlier globally to start the server (with a server.begin() function call).

```
void WiFiStart()
```

The setup function, as in most Arduino sketches, is used to setup the Serial Interface. Kaarthik has also used this function to define pins, manage memory, and calculate important metrics about the incoming data.

```
void setup()
```

In the MakeTable and makeList functions, we write an big HTML table and a big google chart. Instead of defining the these as strings, we calculate the amount of memory required and then hard-write the data as a binary stream of bytes. This technique gives us more control rather than using a predefined data structure, however hard-writing can also be risky if you do not calculate correctly as it can overwrite other nearby data and cause strange problems in your program.

```
unsigned long MakeTable (WiFiClient *pclient, bool bStream)
unsigned long MakeList (WiFiClient *pclient, bool bStream)
```

HTTP headers and footers are defined for documentation purposes. We edited the footer by adding our name to document the fact that the code has been (very slightly) edited.

```
String MakeHTTPHeader(unsigned long ulLength)
String MakeHTTPFooter()
```

The loop() function, as in all arduino sketches, is where the magic happens. The first step of this loop is the data logging, where the byte inputs are recorded. These bytes are shifted and combined in order to form a full 12-bit ADC word. Next, we edited the code to convert from ADC value to voltage value:

void loop()

We added the following line of code to plot the actual voltage instead of simply the ADC value. Because our 12-bit ADC had its input reference set to 3V, the bit output goes from an ADC value of 0 (corresponding to 0V) to 4096 (corresponding to 3V). By dividing the ADC value by 4096 and multiplying by 3V, we can record an actual voltage output. The resolution of this measurement will be 3/2^12, or 0.73 mV.

ADC_Voltage = (ADCword/4096)*3;

These values are then saved in a couple of vectors that will be used to generate a chart later on (pfADC and pulTime). Next come some server checks, and then finally a bunch of HTML formatting. This HTML arranges and defines our components on the webpage. We edited this code so that the axes would be labelled correctly according to our changes. Unfortunately, the edited code could not be uploaded to the Huzzah board due to COM port issues with the laptop supplying the code. However, the resulting web browser graphs from the unedited code can be found below (Fig. 11).

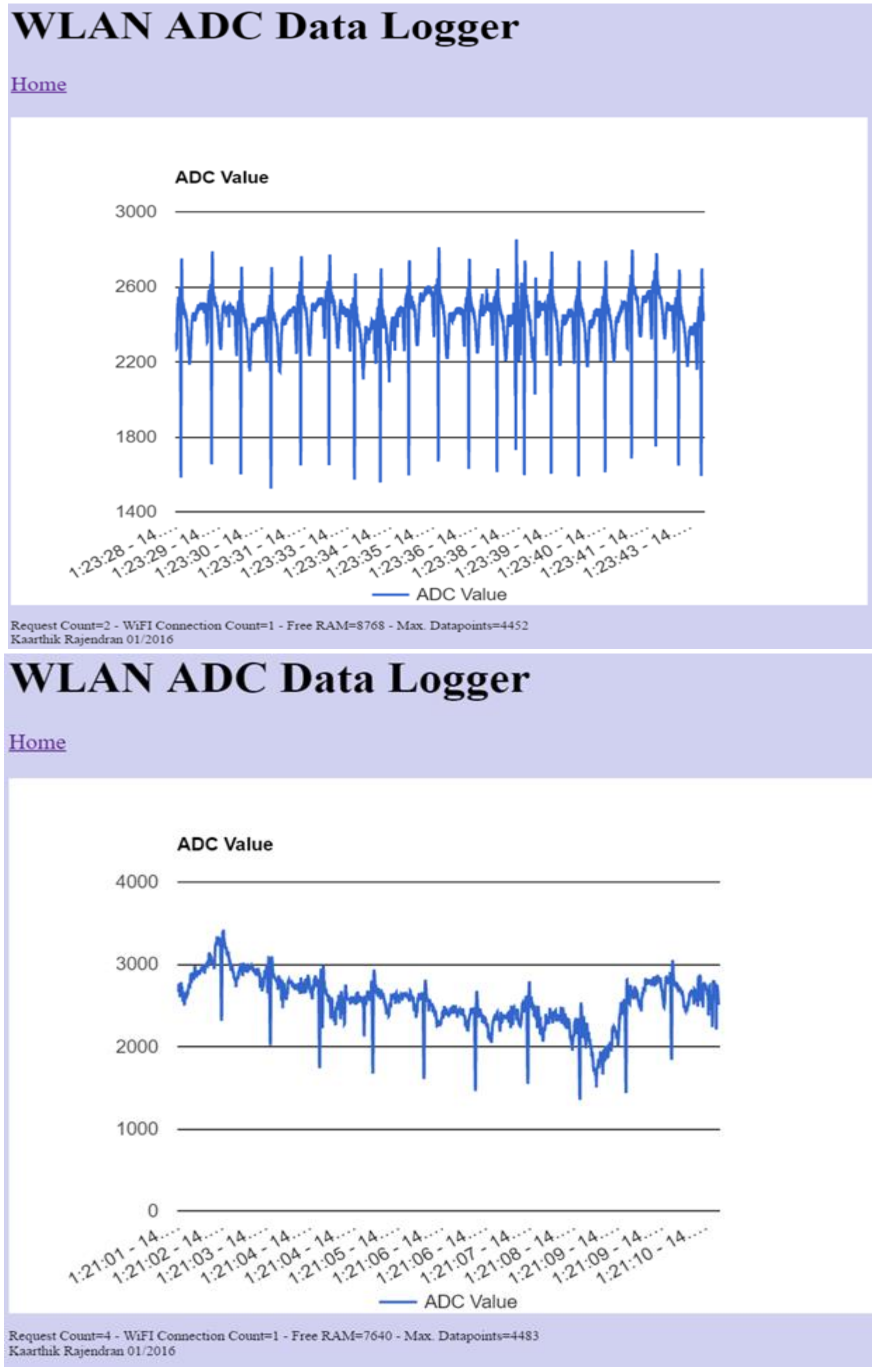


Figure 11: Screenshots of 2 different subjects' EKG (accessed using a laptop web browser)

We see the appropriate waveforms on the graph, with the y axis as voltage and the x value as a timestamp corresponding to the current time. Through this exercise, we were able to understand the difficulties of setting up a wireless system. Because our data needs to be collected at a relatively high sampling rate (compared to most web applications), the data is collected more or less in batches. This way, data can be transferred to the server much less frequently. If we wanted to record in real-time, we likely would have had to pay a subscription fee to an online service in order to get higher data transfer rates.

V. CONCLUSION

In this lab we learned the practical applications of designing a portable instrumentation device. Some of the issues we encountered were designing and troubleshooting a portable PCB, and also setting up a method to record the signal wirelessly. Although we do not have to deal with the headache of electrically isolating the patient, it is still a challenge to design a robust amplifier that fits in a compact package.

In order to take this prototype further, we could design a case for our device that could be attached comfortably to a person. We could also subscribe to a service that would allow for higher data transfer rates, which would allow us to record and even filter our signal in real time.

VI. BONUS MATERIAL

Although there were opportunities to earn bonus points, we felt that our time would be better spent working on the required components of Lab 3. Therefore, regrettably, we did not finish the bonus assignments.

VII. INDIVIDUAL CONTRIBUTION

The schematic, PCB layout, and PCB verification were done by Shalini. Nikki assisted with some of the routing. Priyanka calculated passive component values and located the components to put on the PCB. Shalini soldered the components onto the PCB. Stewart and Nikki did the calculations for the filters as well as the hardware testing and measurements. In terms of the report, Shalini and Priyanka wrote about the circuit design. Nikki and Stewart wrote about the measurements and analysis. We all worked on getting the software to work, especially Shalini. We all worked on the remaining sections of the report.

VIII. REFERENCES

1. “Single-Supply, Rail-to-Rail, Low Cost Instrumentation Amplifier.” Analog Devices. Web. 18 April 2016. http://www.analog.com/media/en/technical-documentation/data-sheets/AD623.pdf (AD623)
2. “REF19xx Low-Drift, Low-Power, Dual-Output, VREF and VREF / 2 Voltage References.” Texas Instruments. Web. 18 April 2016. http://www.ti.com/lit/ds/symlink/ref1930.pdf (REF1930).
3. “Family of Low-Power Rail-To-Rail Input/Output Operational Amplifiers With Shutdown.” Texas Instruments. Web. 18 April 2016. http://pdf1.alldatasheet.com/datasheet-pdf/view/29010/TI/TLV2462IP.html (TLV)
4. “2.7V 12-Bit A/D Converter with SPI™ Serial Interface.” Microchip. Web. 18 April 2016. http://ww1.microchip.com/downloads/en/DeviceDoc/21290D.pdf (MCP3201)
5. “Adafruit HUZZAH ESP8266 breakout.” Adafruit. Web. 18 April 2016. https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/downloads

6. “SilveRest^{TM} - Adult Diagnostic Resting EKG Electrodes.” Vermed. Web. 18 April 2016. http://www.vermed.com/Electrodes/SilveRest.aspx
7. “To patient monitor 5 lead ECG EKG cables ECG/EKG Lead wires SET SNAP ON STYLE.” Ebay. Web. 18 April 2016. http://www.ebay.com/itm/like/271795583032?lpid=82&chn=ps&ul_noapp=true