BME88A: EKG Final Design Report

Eduardo Hirata, Henry Hinton, Pavle Jeremic

March 12, 2015

1 Overview

An electrocardiogram, also known as EKG or ECG, is a noninvasive device that monitors electrical activity produced by the natural electrical system caused by the contraction of heart muscle. This process of depolarization and repolarization causes allows the heart to pump blood through the body. The electrical impulses generated by each heartbeat are registered by the EKG's electrodes and translated by the device into a waveform. Analysis of this waveform by personnel with requisite training can be used to assess heart condition.

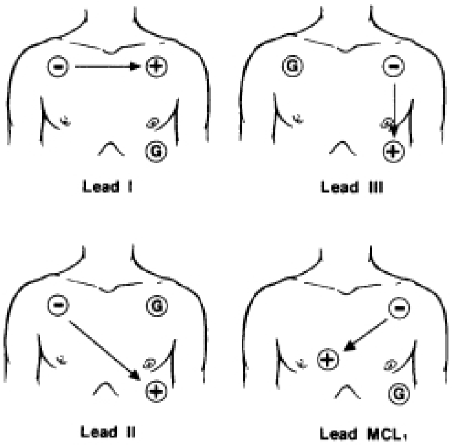


Figure 1: Various positions for 3-lead EKG electrodes. Observe that the ground is not along the path of the electrical signal between the two charged electrodes. [1] Image copied from RTBoard, *Electrocardiography Devices.*

Our design uses three electrodes to measure the electrical activity produced by the heart. By positioning two of the electrodes across the heart and the remaining one elsewhere on the body (not in between the two measuring electrodes) as a reference ground, we can generate a voltage across the skin over the heart as seen in Fig.1 . The resulting difference will be small, however, so we need an instrumentation amplifier to provide gain to render the signal interpretable. This amplified signal will be sent to an Arduino (Sparkfun Redboard). When connected to a computer, the Arduino's serial input can be recorded, filtered, and graphed.

2 PQRST

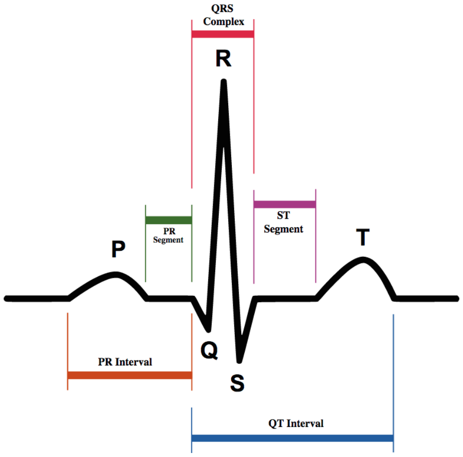


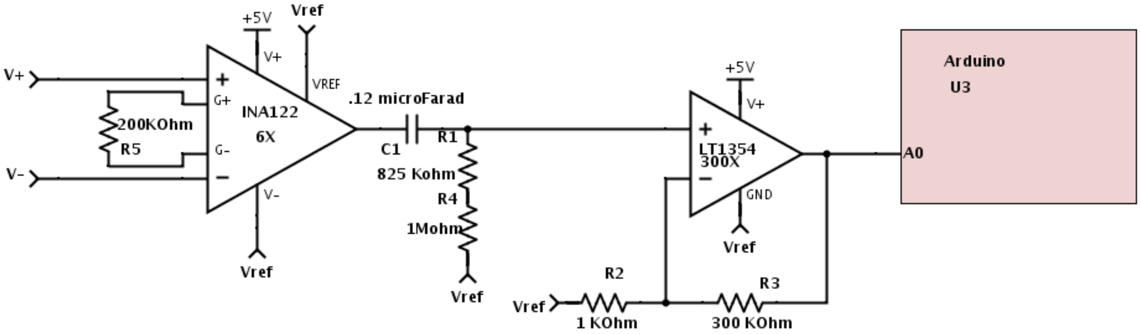
Figure 2: Schematic representation of an EKG reading. The QRS complex represents the ventricular depolarization and contraction of the heart chambers. [2] Image copied from Wikipedia article on Electrocardiography.

For this design we are looking to record the electrical signal of the heartbeat's QRS complex. The QRS complex is the combination of the three main graphical deflections observed on the electrocardiogram produced after the P and T wave as seen in Figure 2. The P wave is the first short upward spike recorded in an ECG, and indicates the contraction of atria that pump blood into the ventricles. The QRS represents the ventricular depolarization and contraction, and it is graphed as a downward deflection, Q; a high upward deflection, spike R, and another downward S wave. Physiologically normal interval durations within the PQRST process happen quickly, in the order of tens of milliseconds. We expect a normal QRS interval to last around 60ms. Overall, we are looking for the electrical signal produced by the heart's contractions; these should be around 0.5mV (before analog amplification). [5]

Due to variability in the signal baseline as a direct result of autonomous muscle contraction, however, we must process the baseline with a bandpass filter. [5] Doing this in software will require post-processing (filtering the data after it has been collected), which we are currently achieving using the scipython bandpass filter module and the scipython notch filter. A bandpass filter is designed to remove any frequencies above frequency B (f_B) and below frequency A (f_A) where $f_A < f_B$. For the EKG, a bandwidth from 0.8 to 100 Hz is optimal, as those parameters represent the normal range of the human heartbeat frequency (48 bpm to 180 bpm respectively), and include the diagnostically relevant higher frequency features of the EKG waveform. The heartbeat of a human could be determined with a bandpass of approximately 0.8 to 3.0 Hz, but the diagnostically-relevant peak intervals would be expunged by such a filter. The notch filter is an inversion of the bandpass filter in that it will remove any frequencies between f_A and f_B. In the EKG, the notch filter will be used to remove the 60Hz noise produced by inductive capacitance of the surrounding (mains-level) electronics. Both filters will be imported from the SciPython module. [7]

The final data set will be fed into a graphing program, which will then output data to two modules: the Graphic User Interface (GUI) Module and the Diagnostic Reader Module (DRM). The DRM will provide basic heart rhythm analysis, which will then output to the GUI. The GUI will be a highly-simplified interface that will allow the user to change some of the parameters of the program to improve usability, possibly including bandpass filter parameter alterations to better serve the needs of the user.

3 Circuit Design



3.1 Components

Circuit Parts:

- Universal ECG EKG Electrodes Electrodes
- Sparkfun Redboard Arduino
- INA122 Instrumentation Amplifier
    - ∗ CF14JT200K Resistor(R5:200 kOhm)
- LT1354 Operational Amplifier
    - ∗ CF14JT1K00 Resistor(R2:1 kOhm)
    - ∗ PR01000103003JR500 Resistor(R3:300 kOhm)
- DC Filter
    - ∗ C330C124KCR5TA Capacitor(C1: .12 (MICRO)Farad)
    - ∗ MRS25000C8253FRP00 Resistor (R1:825 kOhm)
    - ∗ HVR2500001004FR500 Resistor (R4:1 Mohm)
- Wires

3.2 Modules

Hardware Modules:
Human: Only prerequisite is a beating heart (and consent).

Instrumentation Amplifier: Voltage fluctuations may be lower than the Arduino's resolution ($\geq$5mV), so we must use the amplifier to boost the signal to a level we can monitor. Since the signal will be approximately 0.5 mV but the DC noise could be as high as 300 mV [5], the instrumentation amplifier could be saturated by a boosted noise signal. As such, the primary purpose of the instrumentation amplifier will be to "convert [a] differential signal to [a] single-ended signal referenced to Vref" [5], rather than to boost the voltage. Since the gain function of the INA122 is the folowing:

$$Gain = 5 + \frac{200kOhm}{RGain} \tag{1}$$

It was determined that a safe gain would be 6X, corresponding to a RGain of 200 kOhm. Arduino: Converts analog input to digital output for Laptop using a serial output. Computer: Runs software/code and displays data.

Software Modules:

- Register: Generates a list of voltage differential values and time recorded, pairing the two values.
- Noise Filters:
  - Bandpass Filter: Removes noisy frequencies higher and lower than the target bandwidth, imported from SciPython. [7]
  - Notch Filter: Removes 60 Hz noise generated by surrounding electronics.
- Grapher: Plots all data points and connects to generate EKG waveform, imported from matplotlib [8]
- GUI: Basic interface with ability to change graphing options and potentially activate/disable diagnostic screening or change diagnostic parameters manually.

Hypothetical Software Module: Diagnostic: Determines average distance between waveform peaks.

4 Current Design Plans

We plan on using a Texas Instruments INA122PA-ND instrumentation amplifier to find the differential signal from our electrodes, as mentioned above the gain will be minimized to 6X in order to not saturate the INA122. [6] The LT1354 Operational amplifier will provide most of the voltage gain at approximately 300X voltage gain.

5 Cost

These are the fiscal costs associated with this project, not including man hours. The following assumes that a USB cable and a laptop is available to the reader. Ordering extras is highly recommended.

Sparkfun Redboard:	$20
Breadboard TW-E40-1020:	$8.98
Wires:	$4.95
Texas Instruments INA122PA-ND:	$6.35 (3X)
LT1354 Operational Amplifier:	$4.27 (3X)
Capacitor:	$2.64 (3X)
MRS25000C8253FRP00 Resistor:	$0.44 (3X)
HVR2500001004FR500 Resistor:	$0.47 (3X)
CF14JT1K00 Resistor:	$0.10 (3X)
PR01000103003JR500 Resistor:	$0.37 (3X)
CF14JT200K Resistor:	$0.10
Electrodes: Provided by Dr. Karplus	$0.00
Total Cost:	Including Shipping/Markup: $59.61

6 Code

```
’’’
Created on Mar 8, 2015

@author: Henry Hinton, Pavle Jeremic, Eduardo Hirata
’’’
#import required modules
from sys import argv
import numpy
import scipy
import matplotlib.pyplot as plt
#arrays to store values
time = []
vol = []
#read filename from terminal input
filename = argv[1]
#open file
txt = open(filename, ’r’)
print(’Using file: ’ + filename) #verify filename with user
for line in txt: #skip commented lines in data file
    if line.startswith(’#’):
        continue
    fields = map(float, line.split()) #create tuple with time and values
    if len(fields) >= 2: #verify tuple
        time.append(fields[0])
        vol.append(fields[1])
print("Using %d values" % len(time)) #show number of data points

for i in range (0, len(time)):
    print("%.7f %.7f" % (time[i], vol[i])) #print data to terminal
plt.plot(time, vol) #use matplotlib to plot data

plt.show() #open window with plot
txt.close() #close the file safely
```

The above code is our current graphing module that has successfully converted the example data provided by Dr. Karplus [5] and generated a waveform as seen in Figure 3. In the next two days we will implement the bandpass and notch filters from scipy. Immediately afterwards we plan on completing the Diagnostic Module and a GUI. Current plans indicate that we will most likely import GUI functionality from the matplotlib module due to time constraints.

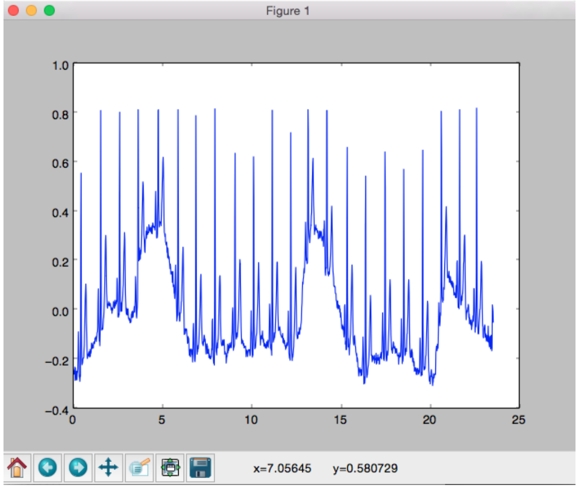


Figure 3: image of raw EKG waveform with no software filtering, generated from static data provided by Dr. Karplus [5]. It is immediatly apparent that the unfiltered signal is almost useless for data extraction without further filtering.

References

[1] ”Electrocardiography Devices.” Electrocardiography Devices. N.p., n.d. Web. 24 Feb. 2015.
http://rtboardreview.com/public/equipment_room/ats_ecg.htm

[2] Cardiac Waveform. Digital image. Electrocardiogram. *Wikipedia*, n.d. Web. 24 Feb. 2015.
http://en.wikipedia.org/wiki/Electrocardiography#mediaviewer/File:QRS_complex.png.

[3] ”How to Read an Electrocardiogram (ECG). Part One: Basic Principles of the ECG. The Normal ECG.” N.p., n.d. Web. 24 Feb. 2015.
http://www.southsudanmedicaljournal.com/archive/may-2010/how-to-read-an-electrocardiogram-ecg.-part-one-basic-principles-of-the-ecg.-the-normal-ecg.html.

[4] Karplus, Kevin. ”Two-Stage EKG.” Web log post. Gasstationwithoutpumps. Wordpress, 14 July 2012. Web. 17 Feb. 2015.
https://gasstationwithoutpumps.wordpress.com/2012/07/14/two-stage-ekg/.

[5] Personal communication with Dr. Karplus regarding typical EKG sensor output.

[6] Data sheet for Texas Instruments INA122PA-ND instrumentation amplifier. Web. March 1, 2015.
http://www.ti.com/lit/ds/symlink/ina122.pdf

[7] Eric Jones and Travis Oliphant and Pearu Peterson and others. SciPy: Open source scientific tools for Python, 2001-present.March 1, 2015 http://www.scipy.org/”

[8] Hunter, J. D. Matplotlib: A 2D graphics enviroment. Computing in Science & Engineering vol.9 #3 pg. 90-95 publisher: IEEE Computer Society, 2007. http://matplotlib.org/index.html