Real Time ECG Feature Extraction and Arrhythmia Detection on a Mobile Platform

Jacob Chong and Kalvin Hallesy (29)

ABSTRACT

This paper details our implementation of a device that is capable of generating an ECG and detecting different types of arrhythmia in its user. The ECG is generated by using two metal dog tags to measure the potential difference across the user's body. This difference is then filtered both in hardware and software to make a clean ECG. Additional software filtering allows for extracting features from the ECG such as heart rate, RR intervals, and QRS intervals which are then used to detect whether the ECG exhibits bradycardia, tachycardia, premature ventricular contraction, and premature atrial contraction. Arrhythmias can be a sign of a serious health problem so being able to detect them at will with our device is important to our wellbeing.

1. INTRODUCTION

The heart is one of the most important parts of the body as it responsible for keeping blood moving throughout the body to supply cells from throughout the body with what they need. Arrhythmias are irregularities of the heartbeat and can be signs of problems with the heart which could be fatal.

The goal for our project is to allow for quick and easy arrhythmia detection. Despite being right inside of our chest, the heartbeat is not an easy thing to feel and it's even harder to notice if there is anything abnormal about one's beat. Our project extracts and clearly displays the Electrocardiograph (ECG) of its user and uses the ECG to detect different types of arrhythmias that may be present.

The path of a signal through our implementation must go through a series of steps before it can be used to detect any arrhythmias. First the signal must be acquired by measuring the voltage difference across the user's body. This difference goes through both analog and digital signal processing to extract a clean ECG which then goes through even more software filters to detect features from the ECG. Analysing these features is what lets us make conclusions about the different types of arrhythmias to be detected. Finally this data is displayed on a screen for easy readability, the heart rate also can be sent to a phone via bluetooth. This report will go into detail of what ECG features are needed, the process of extracting them, and how each component works to bring the vital information to the user.

2. HARDWARE DESCRIPTION

The hardware we used allows for an analog input signal to undergo both physical and digital signal processing and subsequently have the result be displayed visually and transmitted to a bluetooth device. As a result, this ECG hardware can be divided into four different components that are each responsible for a distinct task.

2.1 ANALOG FRONT-END

Due to the electric potential across the body being relatively small, the signal to noise ratio is much smaller than what is necessary. Another problem is that the Teensy reads from 0 V to 3.3 V from its input pins and a raw ECG signal is only a few millivolts at most. The Analog Front-Ends job is to solve both of these problems by filtering out some of the noise that would cause ECG data to be lost (by clipping past the Teensy's measurement threshold) and also amplify the signal to both be noticeable on a 0 to 3 V scale and be centered at roughly 1.15 V so it can be prepared for signal processing in the Teensy. The front-end is composed of freestanding hardware components on a breadboard.

2.2 TEENSY 3.1

The Teensy 3.1 is the microcontroller used to manage all digital signal processing, data storage, and communications with the LCD and bluetooth module; all of this and the ECG program itself is specified through our program which is written in Arduino. The signal from the analog front-end will still be quite noisy compared to what is expected from an ECG so a series of digital filters are used to extract the data from the signal including the QRS peak locations, QRS intervals, heart rate, and indicators of different types of arrhythmia. The Teensy simultaneously communicates with the LCD and bluetooth modules to display and transmit the extracted data.

2.3 ILI9341 LCD DISPLAY

The ILI9341 LCD Display is used to display the ECG and extracted data. It has a 320x240 pixel touchscreen color-LCD so we can display different features in different colors and also use the touchscreen as start indicator. It communicates with the Teensy via SPI.

2.4 ADAFRUIT BLUEFRUIT LE FRIEND

The Bluefruit is a low energy bluetooth module designed by Adafruit to connect to bluetooth enabled devices and transmit data wirelessly. We used it to graph the heart rate vs time recorded from the ECG. It also communicates to the Teensy using SPI.

3. ECG HARDWARE THEORY

As explained in section 2.1, the Analog Front-End has the job of taking in the input, reducing extreme noise, setting the 0 signal level to around 1.15 V, and amplifying the signal itself to readable levels by the Teensy. There are three main stages of the front-end that handle these characteristics.

3.1 BUFFER AMPLIFIER, VIRTUAL GROUND

Arguably the most important part of the front-end is to establish a virtual ground so that the 0 V line of the signal can sit at 1.15 V which is half of 3.3 V. This is because of the 0 V to 3.3 V range of the Teensy. The ECG will have both positive and negative components and the Teensy would not be able to read negative values without the shift. The operational amplifiers used in the circuit also need to have a positive and negative Vcc input so without making a virtual ground, there would be no negative Vcc. This is all accomplished by a buffer input with a positive input of 1.15V which is created by a voltage divider as shown in Figure 1. The amp simply stabilizes the voltage of the output.

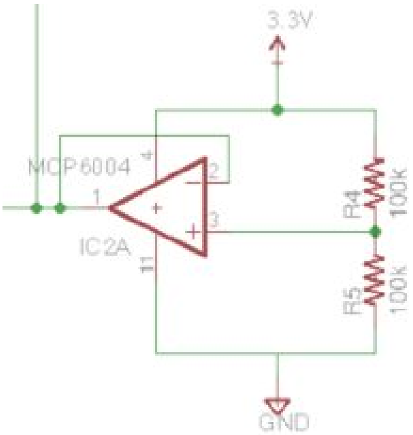

Figure 1. Buffer Amplifier for Virtual Ground

3.2 DIFFERENTIAL AMPLIFIER, INPUT

To measure a signal, we feed the voltage from each individual hand through their own dog tag and 100kΩ resistor into two inputs of an instrumentation amplifier (due to the high input impedance) which functions as a differential amplifier. This is a device that takes two inputs and outputs the difference in voltage between the two inputs, it also has a gain stage to amplify this output. This generates the crudest form of the ECG we have, but the output would still be viewable and resemble an ECG if the noise is low enough.

3.3 HARDWARE FILTERS

While the Teensy will do the bulk of the signal processing for feature extraction, a clean signal to the Teensy is still necessary. The analog world is very noisy and since our input is so sensitive, just a small amount of noise can cause the signal to spike. If the output goes above 3.3 V or below 0 V, the data is lost so we filter out some noise in hardware to prevent that from happening. We use a bandpass filter and 60Hz notch filter in series to accomplish this. We use an inverting bandpass filter op amp circuit that reduces the gain of frequencies outside of what would be found in an ECG. This signal then goes through a Twin-T 60 Hz notch filter to lower the noise from the North American power line.

4. ECG SOFTWARE THEORY

Raw ECG signal that Teensy recieves goes through the analog-digital conversion. This signal data received from hardware processing has lots of noise, such as baseline wander, power line interference, and muscle noise. These noises must be filtered out for the signal to be a meaningful data before analysis.

4.1 ANALOG TO DIGITAL CONVERSION

ECG signal received from hardware processing has to be converted into digital value, so that we are able to manipulate the data to display ECG signal and other useful informations. Our conversion implements Programmable Delay Block (PDB) interrupts for a timed sampling. We use 250Hz of sampling rate, which is given from the specification.

4.2 BASELINE WANDER

Baseline wander noise is a very low-frequency that can be caused by perspiration, respiration, and body movements. Since it is low-frequency, it can be filtered out by high-pass filter. We have tested various number of poles and frequencies of high-pass butterworth IIR filter, and finally implemented 3 poles of 0.5Hz frequency high-pass filter.

4.3 POWERLINE INTERFERENCE

Powerline interference is caused by electromagnetic fields from power lines with 60Hz frequency. Since it is mostly on 60Hz frequency, we have implemented low-pass filter to filter out high-frequency noises. Again, we have tested various number of poles and frequencies of low-pass butterworth IIR filter, and finally implemented 4 poles of 21Hz frequency low-pass filter. Because ECG is ranged between 0.5Hz and 20Hz, low-pass filter of 21Hz removes not only powerline interference, but also other high-frequency noises.

5. QRS & ARRHYTHMIA DETECTION

QRS complex from ECG signal is the most characteristic complex, and it gives us lots of information for heart rate and arrhythmia detection. We have implemented algorithm for detecting QRS complex, and the data is used for heart rate calculation and bradycardia, tachycardia, premature ventricular contraction (PVC), and premature atrial contraction (PAC) detection.

5.1 QRS COMPLEX DETECTION

QRS complex has the wave signal that sits in the frequency range of 4-20Hz. [1] We have tested implementing band-pass butterworth IIR filters with various number of poles and frequency ranges that can filter out other waves than QRS complex. we were able to successively detect only QRS complex with band-pass butterworth 3-pole IIR filter with frequency range of 7Hz - 20Hz. This filter is applied separately on the signal just for the QRS complex detection. Because there was a small delay with filter, we have put some delay on the original signal to be synced with filtered data.

After applying band-pass filter, Q wave in the filtered signal starts with having some negative value. Then R wave starts to reach very large positive value, then the signal again reaches negative value for the S signal. Minimum adaptive threshold is implemented, so that starting of Q wave and the end of S wave detection can be done. QRS interval time then can be calculated since Q and S are detected.

5.2 HEART RATE DETECTION

R waves can be found with filtered signal data from section 5.1, and the interval between them can be used for the heart rate detection. R signal is easily detected with very large positive value. We have implemented maximum adaptive threshold for finding R waves, and the interval time between them is used for heart rate analysis.

5.3 BRADYCARDIA & TACHYCARDIA

Bradycardia and tachycardia are arrhythmias with unusual heart rates. Average heart rate for relaxed normal person ranges between 60-100 bpm. Bradycardia is when someone has slow heart rate of slower than 60 bpm. Inversely, tachycardia is when someone has fast heart rate of faster than 100 bpm. [2] Since heart rate detection is done in section 5.2, detection of bradycardia and tachycardia is easily done by comparing them with the normal range.

5.5 PREMATURE VENTRICULAR CONTRACTION (PVC)

Premature ventricular contraction is where the heartbeat is initiated by the heart ventricles rather than by the Sinoatrial node. People with PVC has characteristic QRS interval of longer than 120ms. [2] Since QRS interval is found from section 5.1, we can check if found data is longer than 120ms.

5.6 PREMATURE ATRIAL CONTRACTION (PAC)

Premature atrial contraction occurs when a part of the atria depolarizes in advance from the Sinoatrial node to make an earlier heart beat than expected. The overall heart beat will stay the same as if PAC did not occur, the beats will just not have even intervals. [3] Therefore, to detect PAC, we look at the average interval between each beat and compare each individual beat to this average. If a beat comes too much earlier than expected (we have it set to 65% of the expected interval) it will increment a counter. If this counter exceeds 10% of the total amount of beats recorded during the 30-second interval, the PAC warning is triggered.

6. FUTURE WORK

While our implementation works for showing the real-time ECG and the data we extracted, I believe some worthwhile improvements would be to have the ECG able to be displayed at the end of the recording session and have the signal written to the screen by scrolling the contents left instead of writing the ECG across the screen and then refreshing it. Another would be to just polish the UI overall.

Being able to see the entire ECG after recording would allow for users to look over each individual beat and would let them look for conditions and abnormalities that our implementation is not designed to detect. Just being able to see it would be satisfying for the general user in any case. Allowing the screen contents to scroll instead of refresh and polishing the UI would make the device seem much more professional and smoother overall. Currently the implementation is a bit barebones outside of actual data extraction and I feel users would have a more enjoyable experience and perhaps even trust the device more if the interface was polished.

7. CONCLUSION

To evaluate the success of our project, we tested it out on a large variety of people with a range of different heartbeats. Our project would work on people with very small and fast QRS complexes and it would work just as well with large and slow QRS complexes. To see if it worked, we had a marker placed for on the screen above each QRS complex detected as long with as well as additional markers placed to show the beginning and ending of the wave. The markers lined up with our visual reference of where they should be in the waves.

Testing for arrhythmias was not so easy of a process, tachycardia could be easily achieved by doing some jumping jacks to increase the average heart rate to above 100 Hz; however, the others were not so easy. To test our algorithms for bradycardia and PVC, we were able to slow our time scale so the system would think the data was coming in at half the speed it actually was. This means that an 80 bpm heart rate would be 40 bpm which triggered the bradycardia warning. The same goes for PVC, a 90 ms QRS interval would be 180 ms which triggered the PVC warning. For PAC the best test was to occasionally tap one of the fingers while measuring to make a fake beat that was faster than expected. This was hard to manage since your regular heartbeat was still present, but adjusting the thresholds during testing allowed for the warning to be triggered meaning the algorithm at least works as expected.

Overall, our project was successful in displaying a clean ECG signal, the user’s heart rate, and detecting bradycardia, tachycardia, PVC, and PAC to the specifications outlined in the project description.

8. REFERENCES

[1] JIAPU PAN and Willis J. Tompkins. 1985. A Real-Time QRS Detection Algorithm. IEEE Transactions On Biomedical Engineering. BME-32, Article 3 (March 1985), 7 pages. DOI: https://courses.cs.washington.edu/courses/cse474/18wi/labs/l8/QRSdetection.pdf

[2] Abhilasha M. Patel, Pankaj K. Gakare, and A. N. Cheeran. 2012. Real Time ECG Feature Extraction and Arrhythmia Detection on a Mobile Platform. International Journal of Computer Applications (0975 – 8887) 44, Article 23 (April 2012), 6 pages. DOI: https://courses.cs.washington.edu/courses/cse474/18wi/labs/l8/pxc3878840.pdf

[3] M Visinescu, IAENG Member and B Gopakumaran. 2004. Automatic Detection of Conducted Premature Atrial Contractions to Predict Atrial Fibrillation in Patients after Cardiac Surgery. Computers in Cardiology 2004 (October 2004), 3 pages. DOI: https://www.researchgate.net/publication/265935519_Automatic_Detection_of_Conducted_Premature_Atrial_Contractions_to_Predict_Atrial_Fibrillation_in_Patients_after_Cardiac_Surgery