Heart Rate Monitor With Bluetooth Functionality and Group Based Application

Matthew Rowe, *MBR35 University of Pittsburgh Student.* Benjamin Tomasulo, *BJT23 University of Pittsburgh Student.* Professor Mingui Sun, *Project Supervisor and University Of Pittsburgh Professor - Neurosurgery.*

Abstract - In the last decade there has been a strong push towards self monitored exercise and companies selling quality heart rate monitors have seen a sky-rocket in their sales. With companies bringing out the most modern and wearable forms of heart rate monitors every month the market has quickly become a competitive battlefield for sellers. In addition, with the introduction of smart phones into society the integration of these with exercise apps and Bluetooth linked heart rate monitors has become a large part of the fitness industry.

Despite the growth and expansion of this rapidly progressing technology we have identified a field that has not yet been penetrated - a wearable heart rate monitor that has group applications in the form of GPS tracking and heart rate data streaming.

I. Introduction

We have drawn inspiration from designs currently on the market. One product presently available is the Zoll LifeVest [1] which is an external defibrillator. This vest bridges the gap for patients between when they get diagnosed with a critical risk of cardiac arrest and when they get an internal defibrillator implanted. Using a wide range of algorithms this product can detect when a patient is experiencing a cardiac event such as ventricular tachycardia (cardiac arrest), and electrodes on the vest deliver an electric shock to re-establish the regular beat of the heart.

We also drew inspiration from the plethora of apps for heart rate monitors. However, there are currently no group based apps that allow the monitoring and mapping of large groups of people. In addition to personal use, the app not only has application to group based competitions such as marathons and paintball, but also has become a proof of concept for a further project which could be used by the military.

Our design utilises the vest of the Zoll design and also the app component of modern products. We measure and analyse the heart rate of the user and tackle the problems of arrhythmia. Should the subject's heart rate fail or go into an irregular beat, emergency services are contacted with the patient's GPS location and any further data which the user shall input in the setup phase for the app (medical history etc).

II. Vest Construction

A heart rate has typically been measured through the use of an Electrocardiogram (ECG) - this equipment measures the electrical impulses of the heart. At the start of each heart beat there is an electrical depolarization at an edge of the heart muscle. This causes the muscle to contract rapidly and this contraction spreads to the next cell and so forth throughout the entirety of the heart. The sinoatrial node is the natural pacemaker of the heart and the ECG measures the net electrical activity of this muscle.

Initially there is a depolarisation of the atrium using a small current. After this there is a short stall as the atrial contraction pumps blood into the ventricles, which is then followed by the standard depolarisation of the ventricles which is shown as the main strokes on the ECG machine.

Although common electrodes are very accurate they are not suited to normal living conditions due to their intrusion on a person's lifestyle. One of the more prevalent modes of detecting heart rate is through the use of a chest strap which measures the electrical impulses of the heart in a similar manner to the ECG but without a plethora of wires.

We also investigated a number of other methods. Wristwatches can detect a heartbeat by measuring the pulsation in the ventral aspect of the wrist on the side of the thumb. Although these are easy to operate, by the time blood reaches the capillaries in your wrist the rate at which is pulses does not always reflect your current heart rate - and this becomes even more apparent in heart rates above 100 BPM.

A further method of heart rate calculation is performed using light. By illuminating your finger tip capillaries with an LED, a sensor adjacent to this light source measures the dilation of the arterial vessels. However, in order to get accurate readings with this method there must be no movement or muscle tensing. [2]

In order to be as accurate as possible and to ensure that the heart rate measurements can be taken while the subject is in exercise, our project utilises the chest strap model where the electrical impulses of the heart are used to measure the heart rate. One common problem with chest strap however, is that the sensing electrode pads are not always sufficiently pressed against the skin. As such, we implemented a compression vest (commonly used in sporting events) with electrodes interwoven into its design. We used the idea of the Zoll Lifefest to construct this but made it a less intrusive design and more comfortable for the user. The vest removed the issue of sensor pressure onto the body, as the electrodes are pressed against the skin due to the already existing design of the compression vest. (See below for an image of the vest in Figure 1) Note also that sweat is not a problem as that will only increase the conductivity of the electrodes.

Figure 1 - Vest Construction

III. Amplifier Circuit

One large part of a heart rate monitor is making the heart signal readable by the chosen microcomputer. The electrical impulses across the heart without amplification are around ~0.01V peak and as such without amplification a heart rate calculation is near impossible. In addition, if not dealt with properly, static introduced to the subject’s body can trump this voltage as can small amounts of noise introduced into the circuit.

Therefore, an amplifier circuit needed to be created which not only had inbuilt filters (remove noise) but also had a stable amplification ability. The circuit that we chose to employ was that from a highly rated 1980 paper which looked specifically at a Ground-free ECG amplifier with two inputs. The circuit (seen below in Figure 2) firstly utilises a filter followed by a differential gain amplifier. After completion the circuit could be seen to work perfectly, and included below in Figure 3 is an Oscilloscope screenshot demonstrating its operation (Vmax = ~2V).

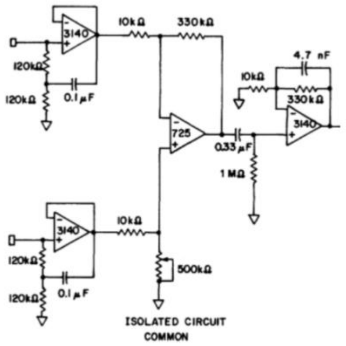

Figure 2 - Amplifier Schematic

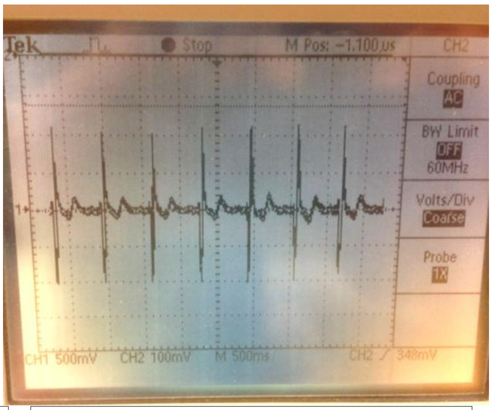

Figure 3 - Oscilloscope Reading

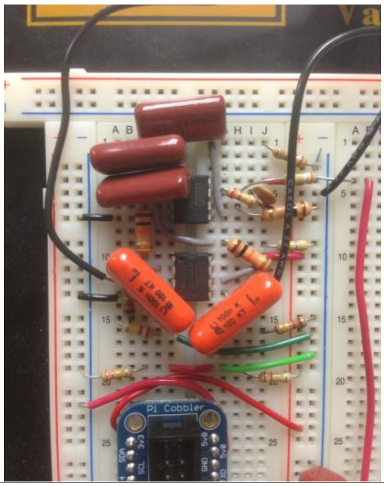

Figure 4 - Breadboard Circuit

As a concluding note- we noticed that there was still noise introduced from body contact, such as brushing against materials and rubbing of the user's shoes on the ground whilst wearing the device. To eliminate this we used a third electrode on the vest (positioned away from the heart) connected to the circuit ground on the amplifier. We found that this eliminated the majority, if not all of the irregular noise introduced from grounding issues and static charge build up on the user. See Figure 4 above for the circuit as seen on the breadboard. (It is also worth noting that although this is seemingly a large circuit, if this design was to go into large scale production, a semiconductor chip could be created the size of a quarter which could contain all of the circuitry seen below)

IV. Digital Signal Processing

There were various ways that we could calculate the heart rate. One of these would have been using a Fourier transform yet this requires a plethora of iterations and on a Raspberry Pi (where processing speed isn't as fast as a modern computer) the speed of code execution is very important. In the end, we opted for peak detection coded in Python - where each peak would be located in the data set, with the distance between each peak giving the time interval between heart beats (and therefore the heart rate).

Another important part of the code was establishing the sample frequency. A sample frequency of 200Hz was used. This was found to not only be adequate for full signal reconstruction in the digital domain, but it also did not saturate the CPU (saturation was noted at faster sampling frequencies). As for the calculation period, every 3 seconds the heart rate would be calculated on the previous 6 seconds of heart rate data. We tried many different frequencies for this and we came to the conclusion that a three second update was adequate but too much variability was introduced if just calculated on 3 seconds of data. As such, we extended the array length to 6 seconds of data and this combination produced the least error when tested (see section XIII for testing results). Finally, in order to ensure turbulent data would not over-influence the heart rate output, the adjusted output was calculated using the following algorithm - 55% current reading, 25% previous reading, 15% 2nd last reading, 5% 3rd last reading.

It is worth noting that one thing we did not foresee was that the Raspberry Pi did not have analogue inputs. As such, an Analogue to Digital conversion circuit was created with a MCP3008 10-bit ADC chip. This uses SPI python code to effectively read in data at the time of sampling.

In addition, (see the Figures below for common Arrhythmia ECG recordings) in order to calculate exact arrhythmia, Fourier transform and amplitude diminishment algorithms were used. However, we found this caused the digital signal processing code execution time to exceed what was allowable in order to get continual measurements from the ADC circuit at the constant sampling frequency. Due to this we adopted a different approach. In order to calculate the peaks (recall that we are using peak detection to locate heart beats in the data), a dynamic threshold is used based on a combination of the mean and standard deviation of the data. The dynamics of the arrhythmia outlined in Figures 6 and 7 will cause the data to fall below this threshold and as such the phone will read a zero heart rate - giving the same health implications as a critically endangered patient and emergency services will be contacted. As such, we feel the arrhythmia heart condition identification has been incorporated effectively whilst optimising CPU usage of the microcomputer.

Figure 5 - Normal Heart Beat:

Figure 6 - Ventricular Tachycardia:

Figure 7 - Ventricular Fibrillation:

[1]

V. Bluetooth Connection

There are several different communication methods we could have chosen for communication between a smartphone and the microcomputer. Most notable of which were wired, Wi-Fi, Bluetooth, and NFC. While a wired connection would be the most reliable, it would be restrictive to the end user as they would presumably still want to use their phone in its initial purpose of a phone. In addition the 3 wireless methods are more than reliable enough for the purposes of this device. Using Wi-Fi however, would tie up the phone's Wi-Fi antenna from connecting to the internet, so this runs into the same problem as a wired connection in terms of being restrictive towards the end user with respects to their phone. On the other hand however, neither NFC or Bluetooth restricts the end user in any way. While certain properties of NFC would be beneficial over Bluetooth, in the end Bluetooth was chosen for its more universal adoption by smartphone manufacturers and end user familiarity with use.

While this connection only needs to communicate from one's pocket to one's belt (where the microcomputer would be mounted), in testing we found communication would still be reliable through walls and several people apart. Indeed, the Bluetooth devices commonly found in phones are intended to have a range of 10m (33ft). [3] (It should be noted that the walls tested through contained chicken wire, which acts as a Faraday cage to Wi-Fi signals yet allows Bluetooth signals through, further demonstrating the superiority of Bluetooth over Wi-Fi.)

Our communication protocol options through Bluetooth were limited by Android, which only allows the RFCOMM protocol. This is a TCP like protocol, where reliability is guaranteed. While the L2CAP protocol, a UDP like protocol (where reliability would not be guaranteed, thereby not over-saturating the microcomputer with transmission threads should a connection error occur) would be preferable for this type of app, this would require the end user to “root” their phones, something most end users are not familiar or comfortable with. We did find however, that RFCOMM was more than adequate and the sending time was more than quick enough.

VI. App Creation

The app is simple to use and intuitive, with 3 basic screens. On the start up screen, there is a connect button used to connect to the microcomputer, a display of heart rate with a “status” below, an age selection field (used in determining a precise heart rate status), buttons to switch to the other two screens, and a quit button to exit out of the app. In order to start up the device, all the user must do is allow power to the microcomputer, wait ~30 seconds for the device to start up, and hit “connect” on the app. The device will then keep functioning until the “quit” button is pressed and the Raspberry Pi shut down. (If the app is in any way interrupted during operation, hitting the connect button will reconnect and establish data sending once more.

The first of the two other screens is a map screen. Using the map screen will auto-zoom on your location in Google Maps and show you the relative locations of other users of the same main server the user is connected to. The map screen also has a back button to take you back to the main screen.

The third screen is the list view screen. This displays all the users connected to the same main server as the end users with their heart rate and status. All 3 screens retain the local users heart rate and status displayed prominently at the top of the screen. In the event that tachycardia or bradycardia is suspected, an email will be sent to a predetermined address alerting them of the user's heart issues. This feature is intended to demo emergency services contact without actually engaging them.

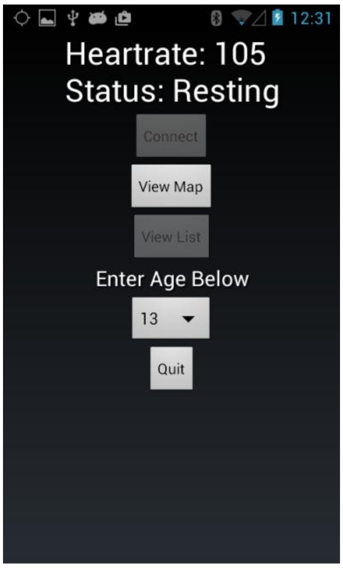

Figure 8 - Home Screen

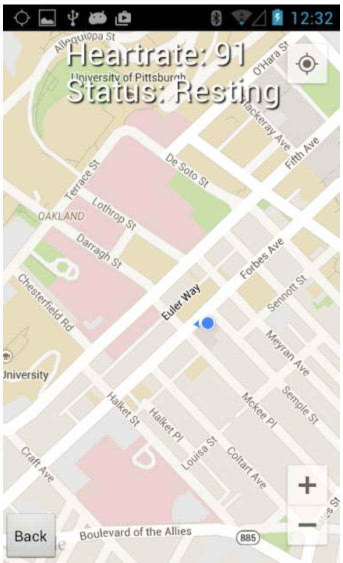

Figure 9 - Map Screen

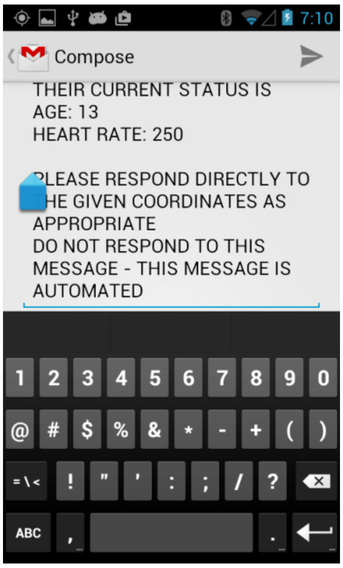

Figure 10 – Automatic Distress Email Screen

VII. Server Connection and Development

The last feature of this design is the Server. The app connects to a main server over the internet using the TCP protocol to allow for the group functionality in the map and list view screens. The server stores an array of phone objects and sends this array to the phones to allow the users to view the heart rate, status, and locations of other users. The test network is single server, but there is no reason why groups could not use their own servers instead with simple replacement of the IP address. In addition, this server was constructed to demonstrate that a user (such as an event coordinator, emergency service for a large scale event of a military commanding officer) sitting on the other side of the world can view up to date heart rate and GPS data of all participating members. A simple visual program could be created to then display all of the heart rate data alongside pins drop at up to date locations on a map.

VIII. Testing and Validation

There was a rigorous testing scheme for this design. First of all, the optimal sampling and calculation period was developed as has been already explained. Optimisation of code was a high priority as a constant yet fast sampling frequency was needed. Isolation testing was also completed on all components. After integration however, a final test was conducted with results being compared against the Polar H7 Heart Rate Sensor[4]. (Although this is not a hospital grade ECG this product also uses electrical impulses from the heart to calculate the heart beat of the user and is an industry standard and a trusted brand name in heart rate monitors.)

Both devices were worn simultaneously (it was double checked that this did not affect the performance of each individual monitor) and tests were run and both resting and raised heart rate levels. Although the calculation periods were not identical, outputs could easily be compared. Regularly they differed by <5% and at maximum of ~7% when exercising. As can be seen, the monitor we have constructed is accurate and a fully functioning product.

IX. Evaluation

Although we believe our project is a definite success, in the mindset of improvement we identified a number of areas which could have further development. One of these is a slight redesign to encompass armed forces applications. If this product was to be used in the military, there would be no smart phone being used, yet the code for the app (written in Java) could be easily translated and run on the microcomputer. Companies specialise in satellite internet dongles for total/95% global internet coverage (with GPS included) and this product, when combined with the Raspberry Pi, can transmit data to a server from a remote location almost anywhere on earth. In combat situations this data would be life saving and with a product weighing only ~150g this would be a great addition to the number of electronics a modern-day soldier has on their body.

Other than this, with our current product there are only two improvements that we identify. One of these is industrial construction of the vest with hidden electrodes and wires running through layers of the fabric. The other is fabrication of a joint ADC and amplifier circuit so that it could be included in a custom computer for mass production. (If mass production was not of interest then the integrated circuit would just be used on its own small breadboard with the Raspberry Pi still in use.)

X. Conclusion

In this paper, we have shown that a new effective, efficient and innovative heart rate monitor with group based application has been successfully constructed over the course of the semester. As a standalone product it has been very sustainable socially, environmentally and economically. It is a non-intrusive design and does not hinder anyone's daily routines. It has made use of the current smart phone market and also uses a microcomputer which is reusable. In the end the whole design would cost ~$80 including purchasing of the Raspberry Pi.

In conclusion, we have constructed a stable amplifier circuit which also effectively eliminates noise, a program written in Python hosted on a Raspberry Pi to successfully calculate the heart rate of the user and Bluetooth to transfer this data to the smart phone app created. This app has the ability to notify emergency services should your heart fail (or go into arrhythmia) and with this data streamed to a server, the data can be viewed from a select computer anywhere in the world. It is truly a unique design and its low cost makes it an affordable solution for group and also personal heart rate monitoring in modern society.

References

[1] "Welcome to ZOLL LifeVest." ZOLL Medical Corporation. ZOLL Inc, 2014. Web. 26 Sept. 2014. <http://lifevest.zoll.com/>

[2] Klobucher, Derek. "Heart Rate Monitors." Heart Rate Monitors: How to Choose. N.p., n.d. Web. 26 Sept. 2014. <http://www.rei.com/learn/expert-advice/heart-rate-monitor.html>

[3] Wright, Joshua. "Dispelling Common Bluetooth Misconceptions". Security Laboratory. N.p. N.d. Web. 30 Sept. 2014. <http://www.sans.edu/research/security-laboratory/article/bluetooth>

[4] Polar. "Heart Rate Monitors and GPS Sport Watches". H7 heart rate sensor. N.p. N.d. Web. 10 Nov 2014. <http://www.polar.com/us-en/products/accessories/H7_heart_rate_sensor>