

ENGR 103 - Spring 2014
Freshman Engineering Design Lab

“Simple 2-lead wearable ECG monitor counting, documenting, and analyzing heart rate ”
Project Final Report

Group Members: [Steven Alpert, sma357@drexel.edu]
[Katherine Chu, kc847@drexel.edu]
[Michael Tarakjian, mjt85@drexel.edu]

Technical Advisor: [Marek Swoboda, ms354@drexel.edu]

Abstract:

This design project utilizes Arduino technology to monitor heart rates. The final product is a wearable heart rate monitor that checks for irregularities in the rhythm of the user’s heart rate and notifies them through a LCD display. This was meant to reduce the amount of unrecognized and untreated heart rate abnormalities. The goal of the project was to create a wearable, non-invasive device for people who need constant monitoring of their heart rates. The project consisted of assembling hardware and circuitry and programming the Arduino to detect irregularities in a heart rate and to notify the user. The Arduino was programmed to read through data from a pulse sensor and compare certain data points to preset normal values. This comparison allowed the Arduino to determine whether or not a heart rate was irregular. If it was determined to be abnormal, the device notified the user through visuals and audio. The device is also capable of connecting to a computer or smartphone over Bluetooth and can output data through a terminal application. Technical challenges included managing to connect all the components in a way that was compact and portable. Programming the Arduino to analyze ECG signals also proved to be a challenge as abnormalities in a heart rate can be very subtle. The result of this project was a functional Arduino based device that continuously checks the user’s heart rate for irregularities in rhythm.

1. Introduction

1.1 Problem Overview:

The goal of this project was to design a wearable Arduino based device to measure and record heart rates. Arduino boards and sensors were used to prototype a wearable electrocardiogram monitor that connects to a user's mobile device. The project was aimed at helping the elderly or ill who live alone. People at risk of heart disease frequently go without treatment because they are not aware when there is something wrong with their heart rate. The device will report any abnormalities in the rhythm of the heart rate to the user so they can seek medical help for any heart rate abnormalities. The connection of the device to the user's cell phone would allow the device to communicate any irregularities in the user's heart rate to emergency medical services or someone else who could help the wearer. This project was meant to improve quality of life and minimize risk of untreated arrhythmia for those who need constant heart rate monitoring.

Portability was the most important design constraint that had to be kept in mind while designing the device. Since the device was intended to be worn, it had to be designed to be as light and compact as possible while still being functional. The ECG monitor also needed to be put together in a way where all the components and wiring would be secure when the user moved. To serve its purpose as an alarm for any irregularities detected, the device also had to be able to easily notify the user of any possible discrepancies in the rhythm of their heart rate. The design and prototype also had to be finished and working by the end of the ten weeks provided.

1.2 Existing Solutions:

Portable ECG monitors are used by people who have a history or are at risk of heart disease. An example of this kind of heart monitor can be seen in Figure 1. The portable ECG monitors in use today are one lead handheld devices that record and save several heart rate readings over a long period of time. These readings are then downloaded onto a computer and given to a physician for analysis. These ECG monitors help doctors find heart problems in their patients in early stages.



Figure 1: A portable ECG monitor used for patients at risk of heart disease

There is also an ECG monitor, seen in Figure 2, that uses a smartphone to display and record the user's heart rate data. This is a newer model that snaps onto the back of a smartphone as a phone case. Two exposed leads on the back allow the user to take their heart rate reading which then shows up on the phone's screen. The data can then be shared with a doctor through email or messaging apps on the phone. [1]



Figure 2: The AliveCor ECG monitor phone case

Unlike existing portable ECG monitors, the device design for this project has the user wearing it constantly. Existing portable heart monitors are user triggered, meaning that they only record a heart rate and share recorded data when the user actively decides to. The design of this project is more similar to the standard heart monitors used in hospitals as it is a continuous reading of the user's heart rate. The prototype of this project also does not read the heart rate, or heartbeats per minute, of the user. It keeps track of the rhythm of the user's heart rate to detect irregular rhythms in the user's heart rate. And while other portable ECG monitors have the ability to share data, they do not do any data analysis and cannot tell whether a heart rate is normal or not. The design for this project analyzes the user's heart rate and notifies the user of irregularities.

1.3 Project Objective:

The project was to design a portable and wearable Arduino based ECG monitor. The purpose of the device was to notify the user of anything wrong with the rhythm of their heart rate. The device worked by picking up an ECG signal from a pulse sensor that the user strapped onto their wrists and leg. The Arduino would then run a program to read the incoming signals and analyze them. If the heart rate's rhythm was abnormal, the LCD screen would flash a message and the speaker would go off to notify the user of the detected irregularity in their heart rate. To ensure that the device was wearable, it was designed to be light and small, using the least components necessary to function. The device is covered in an antistatic bag and held in a waist pack to allow the user to carry it with ease.

The prototype and final deliverable for the project will be a wearable Arduino device that measures the wearer's heart rate with an ECG sensor that the user will wear. A sketch of the design can be seen in Figure 3. The ECG sensor will be one that the user wears on the wrists and leg. The arduino will also be worn on the user in a carrier bag. The Arduino will include a LCD display and speaker to notify the user of an irregularity in their heart rate. A Bluetooth module on the Arduino board will allow for the device to be connected to a smartphone for future work on communicating heart rate abnormalities to emergency medical services if needed. The prototype will run a program that finds the peak of a heart beat. Using this, the device then calculates the time between each heartbeat, or time between the peaks of the heart rate. If the reading of the user's heart rate deviates too far from a set normal time between peaks, the device will notify the user of the abnormality. The device detects sporadic appearances of the QRS complexes (or peaks in an ECG reading, Figure 3 is a diagram of a standard ECG reading) in a heart rate because an irregular pattern of QRS complexes can be an indication of arrhythmia [2]. An ECG reading where the heart's rhythm is irregular is considered abnormal [3], and the device is programmed to detect irregular rhythms in the user's heart rate. The normal time between peaks will have to be set by a medical professional as this value is unique from person to person.

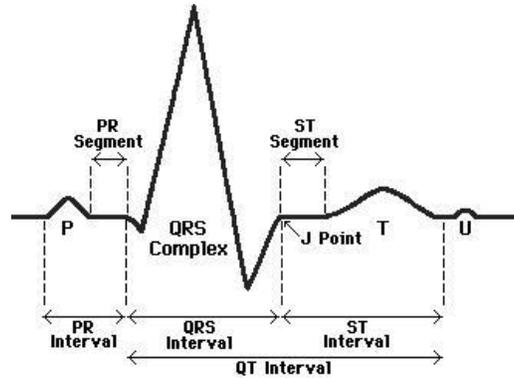


Figure 3: A diagram of a normal ECG reading

2. Technical Activities

2.1 Project Timeline

Table 1: Project Timeline

Task	Wk.1	Wk.2	Wk.3	Wk.4	Wk.5	Wk.6	Wk.7	Wk.8	Wk.9	Wk.10
Literature Study	x	x	x	x						
Electrical Design		x	x	x	x	x	x			
Programming Arduino				x	x	x	x	x	x	x
System Integration						x	x	x	x	x
Testing							x	x	x	x
Final Report Preparation						x	x	x	x	x

2.2 Hardware Assembly and Electrical Design:

A general timeline of how the project progressed can be seen in Table 1. The first three weeks were dedicated to connecting all the components to the Arduino and to have them working. A general sketch of how each component was connected can be seen in Figure 4. At this stage, the only code that was being used was the sample code that came with each piece of hardware. The first activity was to connect the ECG shield and sensors to the Arduino and to pick up an ECG signal. This was accomplished by loading the manufacturer's sample code onto the Arduino and displaying the signals on the manufacturer's PC software, ElectricGuru. With the Arduino connected to a computer, it was possible to identify a regular heartbeat. This indicated that the shield and probes were functioning properly. The LCD display and speaker were then connected to the Arduino. This involved soldering the leads of the LCD display to the pins it came with. Both were set up with the help of tutorials found online. Figure 5 shows the ECG shield, LCD display, and speaker connected to the Arduino. The sensors used in the design were worn on both wrists and the leg of the user and had to be connected to an ECG shield. A small, rechargeable battery pack was also plugged into the Arduino so it could be carried around and worn.

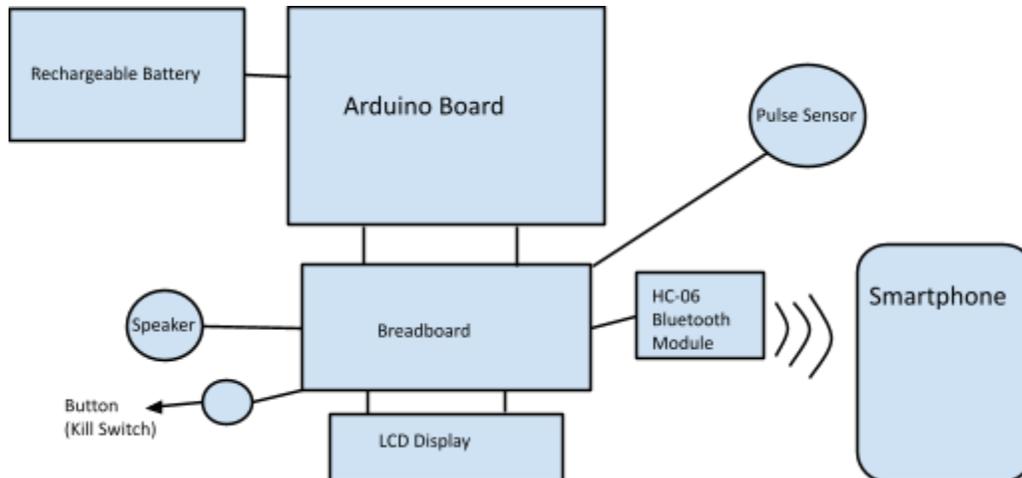


Figure 4: A sketch of the design of the prototype

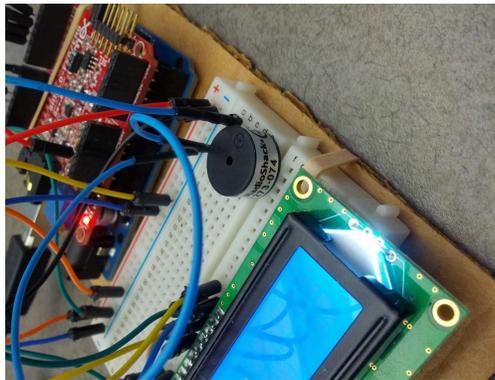


Figure 5: The ECG shield on the Arduino board in the back. The speaker and LCD Display are connected to the breadboard in the front

A Bluetooth module was also connected to the Arduino board. This, like with the LCD and speaker, was done with the help of tutorials found online. The digital pins 0 and 1 on the Arduino corresponded to TX and RX (transmit and receive) on the Bluetooth, respectively, this is shown in Figure 6 and explained in [4]. This meant that the same data sent over USB connection to a computer previously could be sent over a Bluetooth connection. This was tested by sending data to an Arduino Bluetooth Terminal app and comparing it to data that was received through USB connection to a computer.

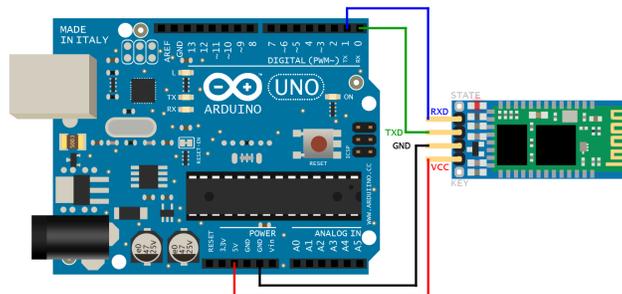


Figure 6: A diagram showing how the Bluetooth module was connected

2.3 Arduino Program:

Programming of the device started in week four and continued through week ten. The programming of the Arduino started with running several different sample codes from online sources to find out how the Arduino received and outputted data. The software and code from the manufacturer of the ECG shield and sensors were used to get a reading of a normal heart rate. This code was then altered to output the ECG signal as numerical values, these values were plotted on Excel to ensure that the readings from the ECG shield were accurate. The device calculates a moving average of the ECG signals to filter or smooth out the data. One of these plots is seen in Figure 7, a plot like this showed that the data was accurate and could be used for analysis. These numerical values corresponding to the electrical signal picked up by the sensors were then displayed live on the LCD display (as seen in Figure 8) using the `analogRead()` and `print()` functions. A splash screen was also coded for the LCD. This functioned as the visual notification of the device. The device is also capable of audio notification through the speaker. A tone (which sounds like the NBC chimes) plays at the startup of the device. There is also code for the speaker to beep loudly should the device detect an irregularity in a user's heart rate.

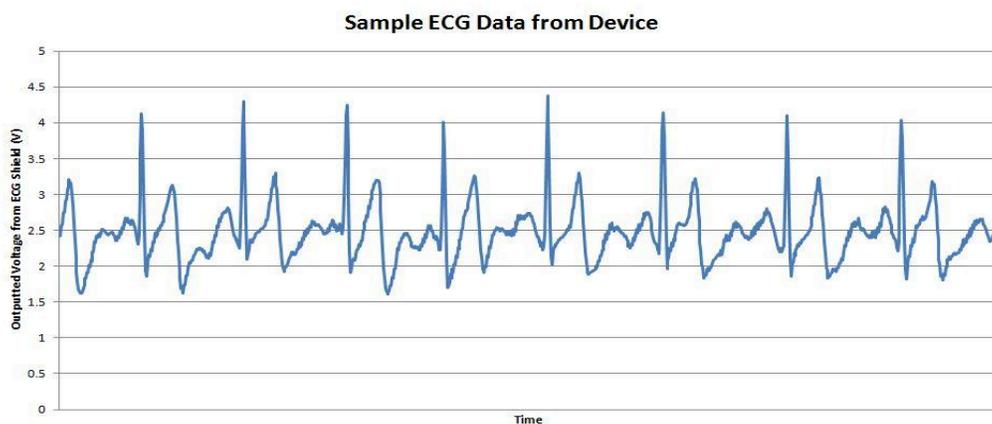


Figure 7: ECG signals outputted by the ECG shield translated to voltage. These were then plotted on Excel to make an electrocardiogram



Figure 8: ECG reading displayed on the LCD

With code written for the notification of the user, the focus then went to coding for actual analysis of the data. The program for the Arduino reads through data from the sensors, finds the peaks of the user's heart rate, then calculates the time between the peaks of the user's heart rate. This serves to keep track of the rhythm of the user's heart rate. If the time between the peaks of the heart rate deviates too far from a set normal (more than 2.5 standard deviations from the mean time between peaks) for the user then the Arduino records the data point as a

“bad” point. Statistical analysis, focusing on standard deviations, was used to differentiate normal heart rates from arrhythmia. Heart rates under normal conditions have a standard deviation of time between heartbeats that is significantly lower than the standard deviation of heart rates under arrhythmia conditions [5]. If the time between the peaks of the heart rate is within an acceptable range from a set normal, then it is recorded as a “good” point. The Arduino then calculates the proportion of “good” points out of the last several data points. If this proportion is too low, the Arduino alerts the user of an abnormality in the rhythm of their heart rate. If the proportion of “good” points is above the cutoff indicating no irregularities in the rhythm of the user’s heart rate, then the Arduino starts the loop again. There is also a button attached to the Arduino that wipes previously acquired data and resets the device. The whole program is detailed in Figure 9.

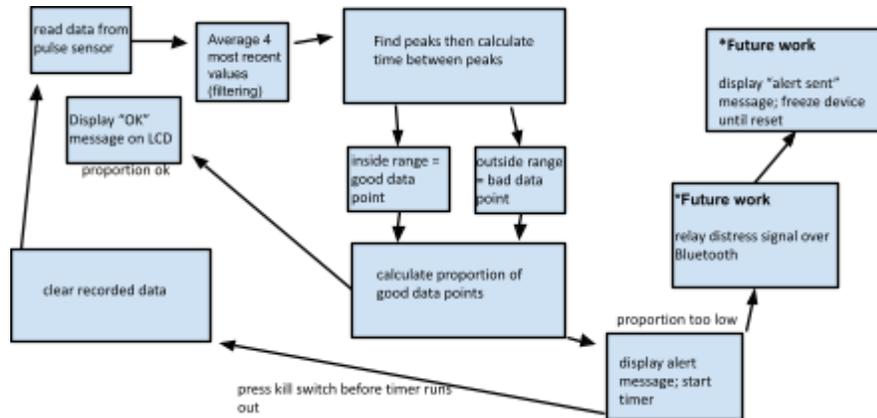


Figure 9: A flowchart of the Arduino program

2.4 Budget:

Table 2: Project Budget

Category	Item	Quantity	Cost
Arduino	Arduino Uno	1	25
Sensors	Pulse Sensor	1	25
Connectivity	HC-06 Bluetooth Module	1	11
Peripherals	Speaker	1	3
	LCD Display	1	10
Miscellaneous	Breadboard and Wires	1	10
	Rechargeable Battery Pack	1	15
	Switches	12	6
	Antistatic Bag	1	8
	Waist Pack	1	0(donated)
Total			113 dollars

Table 2 shows a summary of the project budget. The budget was broken down into several categories.

2.4.1 Arduino: This category in the budget of the project consisted of the Arduino Uno board used in the design to record and analyze data from the sensors. The board cost about twenty five dollars.

2.4.2 Sensors: The sensors used in the design strapped onto the user's arms and legs and cost twenty five dollars.

2.4.3 Connectivity: The device utilizes a HC-06 Bluetooth module to allow for wireless communication of data to smartphones and computers. The Bluetooth module cost eleven dollars.

2.4.4 Peripherals: The main purpose of the project was to notify users of heart rate abnormalities, this was done through an LCD display module and a simple speaker. The LCD display module was purchased for ten dollars and the speaker cost three dollars.

2.4.5 Miscellaneous: This category included all the wires and the breadboard used to connect all of the components and the items used to hold and carry the device. The battery and reset button on the device was also included in this category. In total, the breadboard, wires, bags, battery and buttons cost about thirty dollars.

3. Results

The prototype for the project is a wearable Arduino device that measures the rhythm of the wearer's heart rate with an ECG sensor that the user will wear. The ECG sensors are ones that the user wears on the arms and leg. The Arduino will also be worn on the user in a carrier bag. The Arduino includes a LCD display and speaker to notify the user of an irregularity in their heart rate. A bluetooth module is included on the Arduino board to allow for connection to a cell phone. A reset switch is also attached to the Arduino to stop the device's notification system should it detect a false irregular rhythm in the user's heart rate. The whole device, ready to wear, can be seen in Figure 10. The Arduino board, Bluetooth module, breadboard and speaker are kept inside the pouch to prevent damage during use. The reset button and LCD display module are placed on top of the waist pack for ease of use. The sensors and battery are kept in the front pocket of the waist pack for easy access.

The Arduino is programmed to read through a user's heart rate, find the peaks of individual heart beats, and calculate the time between the peaks. This allows the Arduino to keep track of the rhythm of the user's heart rate. If the time between peaks in the user's heart rate deviates far from their normal values, the Arduino is programmed to notify them through a message that comes up on the LCD display and a noise that plays on the speaker. The reset button on the Arduino is programmed to stop all data reading, display, noise, and cell phone functions and resets the device.

The original objective was to have the Arduino able to notify the user of an irregularity in their heart rate and to allow for notification of a doctor or emergency medical services. The device is able to detect large deviations from a normal rhythm in the user's heart rate. The device is also able to connect with a user's smartphone or computer through Bluetooth. The Bluetooth connection allows for wireless communication of the ECG signal.

Testing was done by attaching the device to various people. This ensured that the device was picking up the right signals. To test the devices ability to detect irregularities in a user's heart rate, the sensors were removed abruptly from test subjects or jiggled around. Because the sensors are not very accurate, this created interference that distorted or eliminated the reading, making the ECG signal appear as an abnormal heart rate. If the device's notification system was set off, that meant that it detected irregularities in the user's heart rate correctly. Testing like this was done several times on three different individuals. When the test subjects weren't moving, the signals that the device picked up were normal, did not trigger the notification system, and when graphed appeared like the one shown in Figure 7. When interference, or "heart rates" with abnormal rhythms were created, the device's notification system was set off. This indicated that the device worked as intended. The process through which the device determines whether a user's heart rate rhythm was normal or abnormal is explained in Figure 9.



Figure 10: The finished prototype

4. Future Work

While the device works, there are some aspects of it that could be improved. The initial design of the project included a smartphone app which would be controlled by the Arduino. This was not made due to a shortage of time. If there were more time allotted for the project, this would be the first thing to be added to the project as it would increase the effectiveness of the device the most. Because the ECG signals can be erratic and inaccurate at times, other improvements include more advanced filtering of the signals received by the Arduino from the sensors. The current program for the device filters data with a moving average. Other ideas for filtering included using the slopes of the signals received by the Arduino. The device can also only detect large deviations from a normal heart rate. If the device were more sensitive to irregularities in the rhythm of the user's heart rate, the notification system would be set off too often. This is also a downfall in other, commercially available, heart monitors as problems in a person's heart rate can be very subtle [1]. This can be fixed by using a more accurate, medical grade ECG sensor with more leads. The signals from each lead would then have to be analyzed. This could be a future improvement of the design. The normal values for a user's heart rate has to be preset in this design. Currently, this was done by taking a reading of a user's heart rate and finding the average time between the peak of each QRS complex. Future work on the device could include programming the device to be able to calculate normal values for a user for comparison of data. If this were done, the device would be easier to use as it would not require a medical professional to find the standard time between heartbeats for an individual when they are healthy. A more complex program could be written for the device in the future to detect specific heart problems. This could be done by having the device detect all parts of an ECG signal. The current program only detects the R peak of a QRS complex in an ECG reading. A better program would include code that accounts for irregularities of the user's P, T, and U waves.

5. References

- [1] Hickey, K , *Detection of Recurrent Atrial Fibrillation Utilizing Novel Technology*, Journal of Atrial Fibrillation , vol 6 , no 4, p.50 - 52
- [2] Ashley, EA, Niebauer, J, *Cardiology Explained*. Remedica ; ed 1, Vol 1. ;.2004.
- [3] Chernecky, CC, Berger, BJ, *Laboratory Tests and Diagnostic Procedures*. Saunders; ed 5; 2008.
- [4] Maker Media Inc., "Serial Bluetooth Products User Instructional Manual ". Retrieved April , 2014 Available: http://makezineblog.files.wordpress.com/2014/03/hc_hc-05-user-instructions-bluetooth.pdf
- [5] GS. Nayak, C. Puttamadappa, A. Surekha Kamatha, , *Statistical Analysis of ECG signals for Arrhythmia Detection* , IFMBE Proceedings , vol 21 , no 4, p.251 - 253