Miami University

Infrared Heart Rate Sensor

By: Jim Kolonis, Jon Kosir, Jordan Ma, and Drew Sanders
ECE 387
Dr. Tareq Khan
5 / 13 / 2015

Abstract

To begin, in the interest of looking into a useful product under the topic of biomedical applications for this project, we began looking into some practical ideas of human vital monitoring. On this subject of vital monitoring, we arrived as the interesting idea of human heart rates and how to improve upon the already developed fields. Of course, when including the requirements and the structure of this project, we ultimately arrived of monitoring a person's heart rate remotely, while trying to minimize cost and increasing usability. In our project, we utilize an Arduino UNO with Bluetooth and Android interface software in conjunction with an Infrared Heart Rate monitor that can attach to a patient's ear, via an clip. With the hardware in mind, we would then turn our view towards the goal of making this system robust and convenient to use. How this was accomplished, was by the idea of looking into the idea of making the heart rate monitoring a mobile platform that patients can wear with them while moving around, and without being too cumbersome.

Introduction

With advances in technology booming, and the ever present need of increased precision and usability in the fields of medicine and biomedical engineering, the applications for such technology have also grown. One subject in particular that would benefit greatly from these advances are the monitoring of a human patient's vital signs, such as: temperature, heart rate, blood pressure, cognitive responses, and so on. The interests of this project focuses on providing a flexible, mobile platform that can be attached to a patient for monitoring with the ease of not being a cumbersome set of machines that could hamper the patient's mobility, or prevent them from moving at all.

For this project, the investigation into the monitoring of a patient's heart rate was chosen, both to satisfy specifications and the goal of providing a useful product that could potentially be looked into further in the future. To best deliver these results, a Infrared Grove Ear-Clip Heart Rate Sensor (model: MED03212P) was chosen. As mentioned, this model of heart rate sensors use a dual sensor -- one sensor that is embedded on one side of the ear clip, while the second sensor is on the opposite side. While attached to the patient's ear, the sensor utilizes infrared light to transmit through the patient's ear to detect the rate and / or presence of blood moving through their veins. Pending on the rate of blood flow, the sensor then transmits this data through the Arduino Uno unit which then converts the data into a readable heart rate that can be view via a computer monitor or android phone with the proper application downloaded.

Table of Contents

Related Work

When looking at the topic of biomedical engineering for this project, we arrived at many other similar projects during our research. Looking at these project, we then analyzed what their overall goals were, how they achieved their embedded system designs, and what types of sensors they utilized for their specific projects. Primarily, we looked into projects on the website: medicarduino.net, where many similar projects can be located. In particular, a few of these projects use Arduino boards to calculate a patient's heart rate, such as the project: "A DIY photoplethysmographic sensor for measuring heart rate." While this project was accomplished using a different arduino platform and sensor then our's, it helped to give us an idea of what to do in our own project, and how to vary the project from what we researched.

From these projects, we decided to run with the idea of detecting a patient's heart rate via a sensor of some kind. Looking into the other projects, such as another project on medicarduino.net titled: "An easy way to send your heartbeat to the cloud". From this project, we saw that they would send the data in the Arduino to an android phone with the use of an application. For our project, we decided on the same sort of heartbeat sensor that was used, but of course, varying our embedded system with the requirement of converting the Arduino code to C++ code. Thus, our idea was to incorporate the use of the heart rate sensor for patients via a mobile platform was thought of (as opposed to them staying still and hooked up directly to a desktop or laptop computer monitor).

System Architecture

The basic architecture for our embedded system, as can be seen in figure 1. To protect the components of the Bluetooth HC-06, a voltage divider was used. The Transmitter (TX) on the Arduino outputs 5V maximum while the Receiver (RX) on the Bluetooth HC-06 can receive 3.3V maximum.

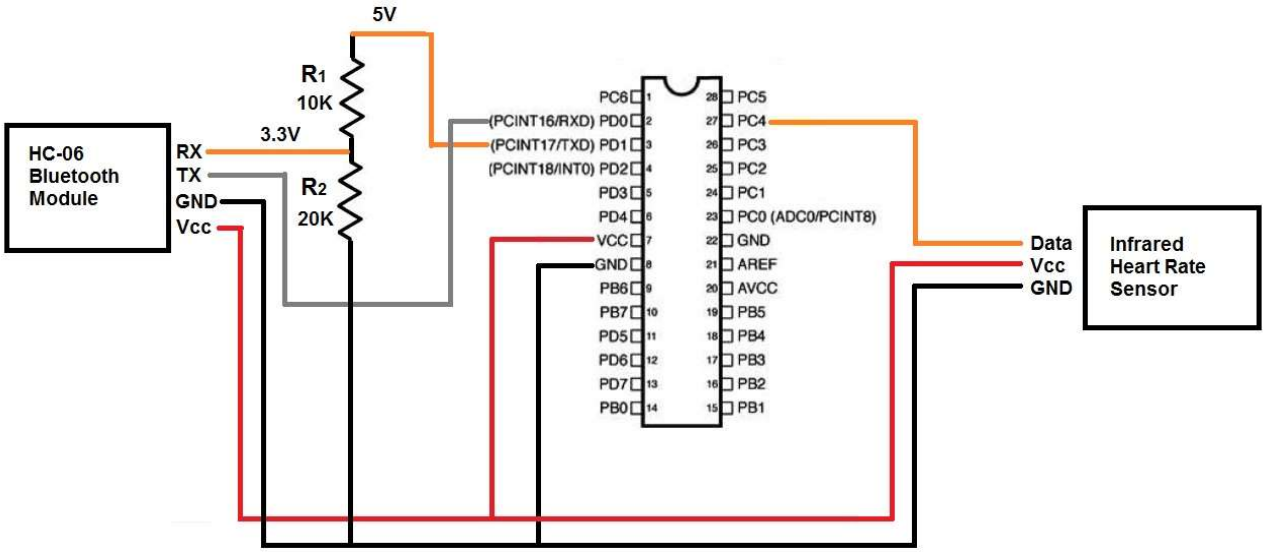


Figure 1 Basic System Layout

Also presented in figure 2, is the actual Infrared Grove Ear-Clip Heart Rate Sensor (model: MED03212P) that we use as a part of our system.

Figure 2 Ear Clip Heart Rate Sensor

As can be seen in the figures below, we designed a custom physical enclosure for our embedded system components. This enclosure was designed using a free 3D modeling software called TinkerCad, which was then sent to a 3D printer to be printed. All material was printed using ABS plastic filament (Black and White). In figure 3 (below), the mobile enclosure is open to show where each separate component was placed. Figure 4 presents the mobile enclosure in it's ready to use state, where detachable clips can be seen. The clips are for both holding the enclosure together, as well as presenting somewhere a patient can attach to themselves, via belt-loops.

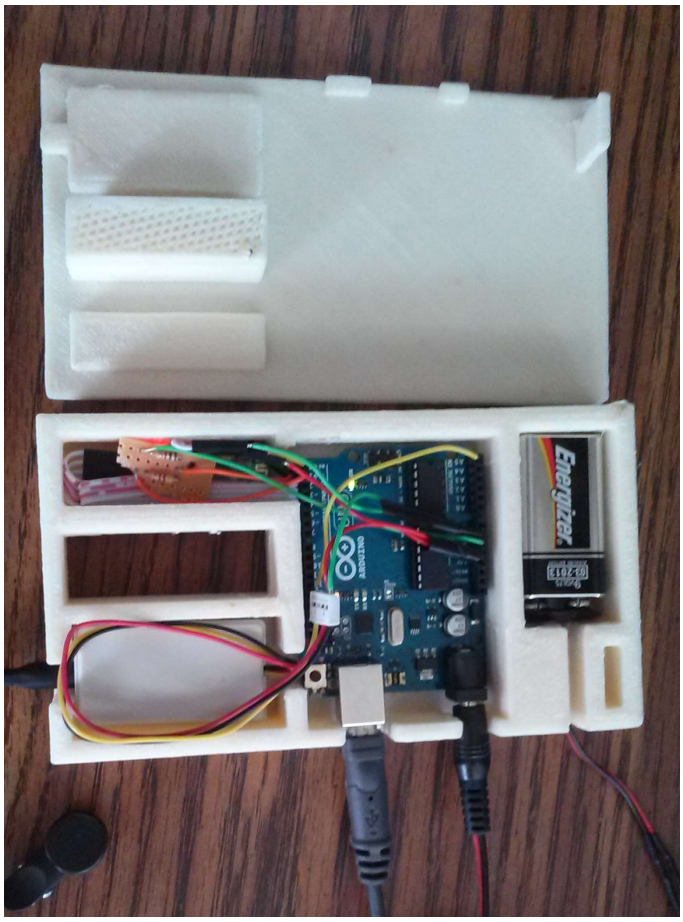


Figure 3 Mobile Enclosure Disassembled

Figure 4: Mobile Enclosure Assembled

Embedded Firmware

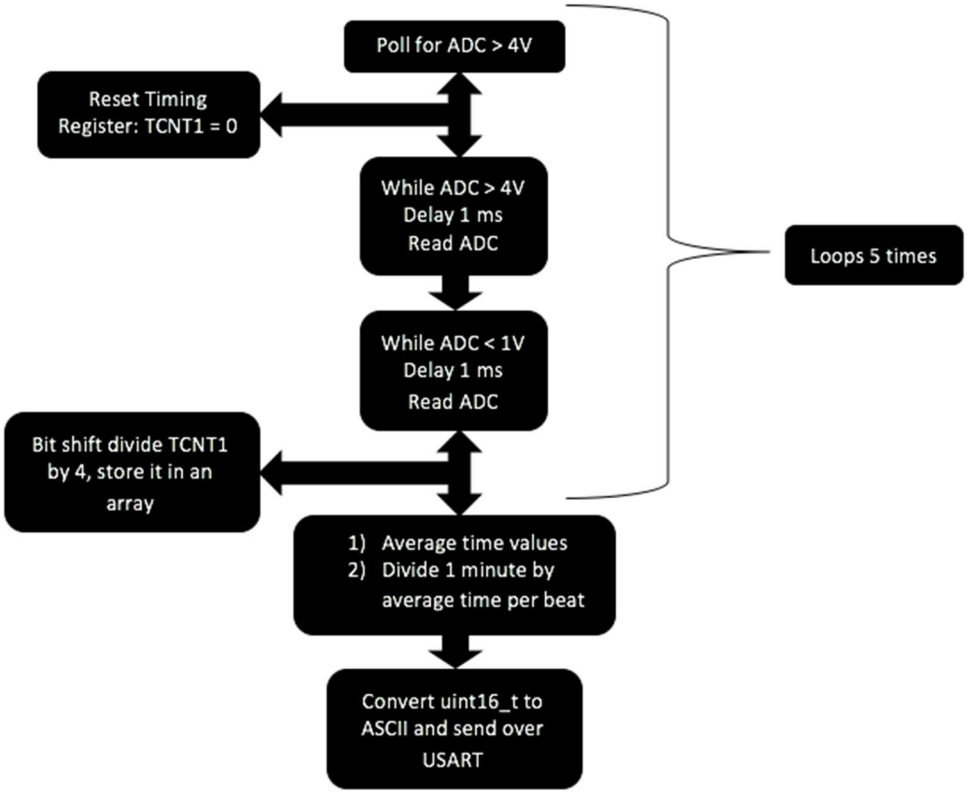


Figure 5 Firmware Structure

Android Application

For the Android application development portion of this project our team decided to use MIT's Application Inventor 2. This decision was not made from the start but ended up to be the more promising route. We first began development in Android's SDK using the Java language, this however ended up to be a more involved and time consuming endeavor then we realized. After much effort was put into developing in Android's SDK we made this realization and began looking for a new solution. That is when we found MIT's Application Inventor 2 and we decided to use it instead.

Application Inventor 2 provided us with a very simplistic and easy to use framework and allowed us to get an application developed for the Android platform even with the time constraints of this project. App Inventor does not need written code, but instead it uses a drag and drop GUI platform which writes the code for the user. The user is able to choose objects and place them on a mock screen and organize them in the way they see fit using the many options available. This allows for easy design development of the front end of the Android application. Once the front end is built you can then start working on the backend of the application.

The backend development uses blocks which are organized very much like regular code, it has methods, variables, logic statements and much more. As seen in

Figure 5 we were able to use some of App Inventor's pre-set methods to connect to the Arduino's bluetooth. Using Android's SDK this would have been much more complicated. The pre-set method blocks in Application Inventor were able to assign a button action which creates a list of available bluetooth clients that the device is able to pick up. Then once the user sees the client they would like they can click on its name and the Android device connects to that client. The blocks you see below those two blocks are just some simple variable instantiations.

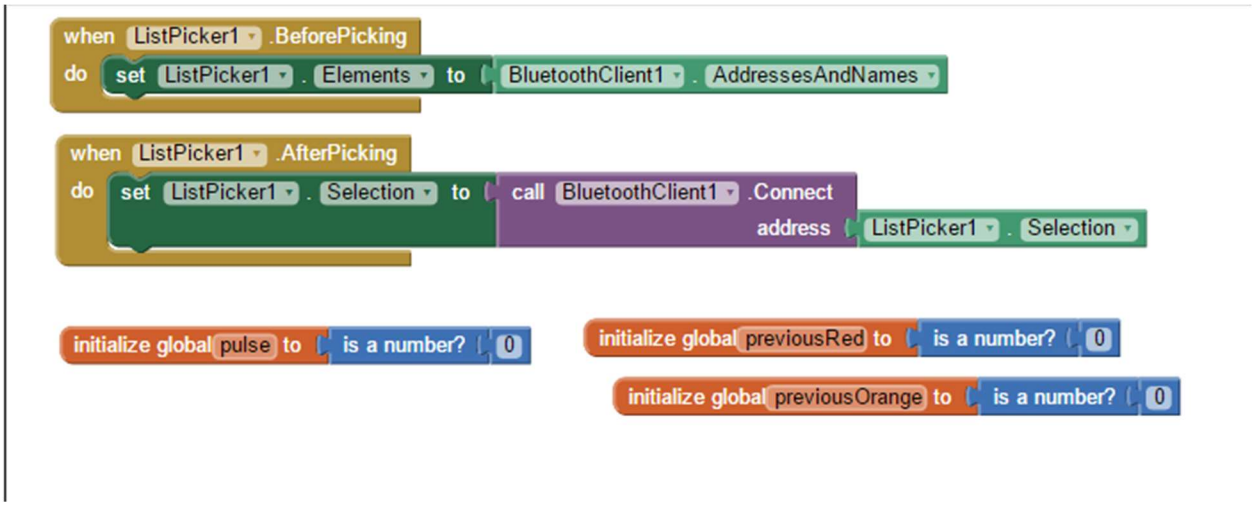


Figure 5: Bluetooth Connection

In the figure below you can see the rest of how our application was programed. On a set clock timer it calls to the Arduino for data to be received. Then it stores that data in a variable as well as a TinyDB. The TinyDB in this code is not used. We had more plans for analyzing the data the app would receive from the Arduino, but we weren't able to include them due to lack of time. Using the blocks we then set label1 which is displayed to the user on the front end of the app to the current heart rate the Arduino has sent. It checks if the value is within the bounds of one of the thresholds. If it isn't the background stays black and no notification is sent. Otherwise, depending on if the user's heart rate is mildly high or severely high, the apps screen will turn red or orange and send an appropriate response to the user.

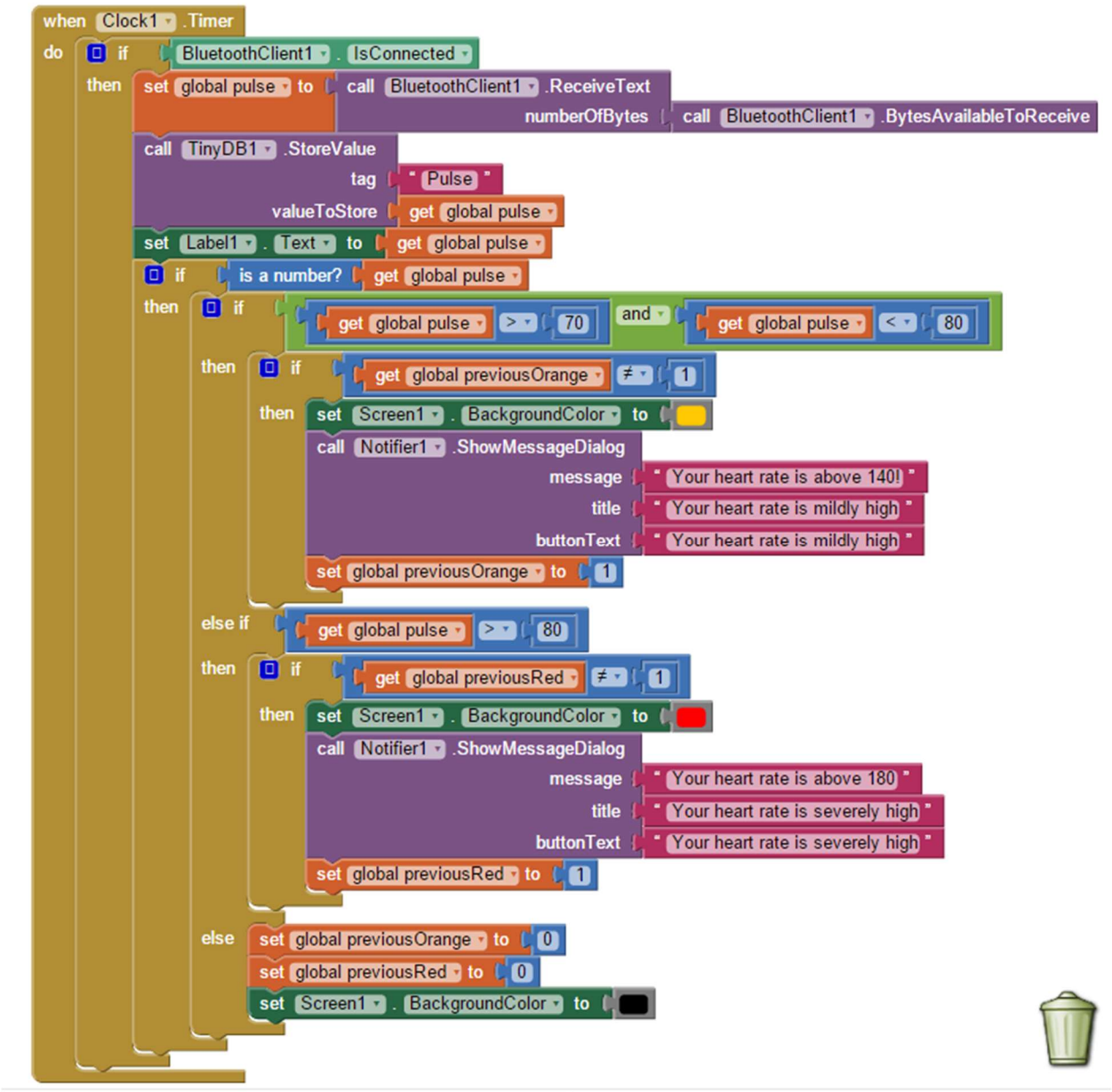


Figure 6: Application Data Analysis

Results

Utilizing the MIT App. Inventor program on a laptop computer, the creation of the heart rate monitoring application for an Android phone was successfully created. Once downloaded to one's phone, the application can then be started and receive transmitted data from the IR heart rate sensor via Bluetooth. As can be seen in the figure below (figure 5), the screen will display a patient's heart rate on the phone's screen in BPM.

As the app. runs, it will constantly send data to the phone and continue to monitor a patient's heart rate, and is able to send a warning message at two different

threshold stages once a patient's heart rate becomes higher and higher. If the patient's heart rate reaches the first threshold, the screen will flash to a yellow background, and sends a notification to the android phone notifying them of a "mildly high" state. If the patient's heart rate reaches the second threshold, the screen will flash to a red background, warning the patient their heart rate is too high.

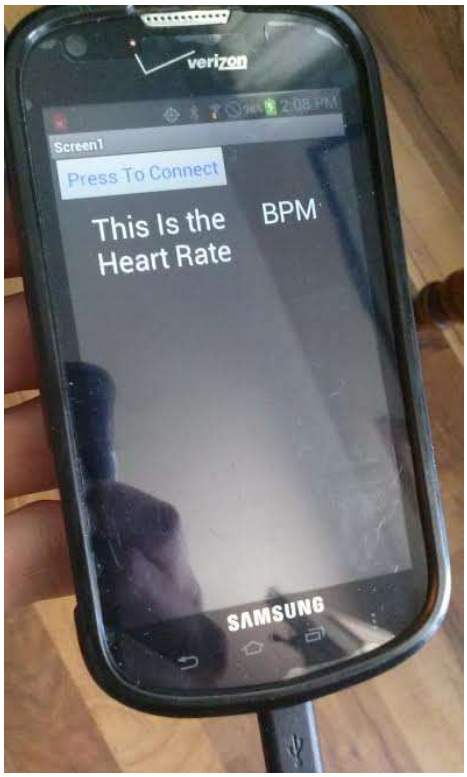


Figure 7 Application Screenshot

Battery Life Estimations

To power our devices within the mobile enclosure, we incorporated the use of two different 9V battery types during our experimentations. First, we used a Alkaline battery that has an approximate capacity of 565 mAH, and an estimated battery life of 6.35 hours. This battery type was obviously a short-lived source, but we were able to look into another 9V battery type that proved to have a longer life span. The second battery type we then experimented with was a 9V Lithium battery, that had an approximate capacity of 1200 mAH, and an estimated battery life of 13.5 -- effectively, doubling our original battery life with an Alkaline battery and allowing, of course, the operation of the IR sensor and the mobile enclosure to last longer.

Analysis of Final Product

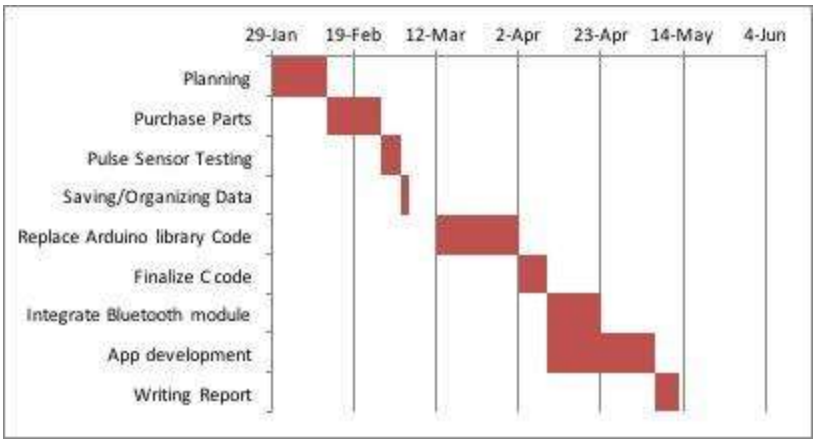


Table 1: Power Consumption of Hardware Components

Device	Voltage	Current	Power
IR sensor	5V	7.3 mA	36.5 mW
BlueTooth	5V	43 mA	215 mW
Arduino			549.5 mW
		Total Power:	801 mW

Table 2: Battery Life

9V Battery Type	Battery Capacity	Expected Battery Life
Alkaline	565 mAH	6.35 Hours
Lithium	1200 maH	13.5 Hours

Table 3: List of Software, Hardware, and their Prices

Device	Function	Price
Arduino Uno	Connection of Components, Programming of Microcontroller	$24.00 (Amazon.com)
Ear-Clip Heart Rate Sensor	Collection of Pulse Data	$12.90 (SeeedStudio.com)

HC-06 Bluetooth Module	Serial Communication to Android	$8.79 (Amazon.com)
Physical Enclosure	Protection and Convenient Packaging of Components	Approximately $5.00 Material Cost in 3D Printer Filament
Arduino IDE	Programming of Microcontroller	FREE
MIT App Inventor	Programming of Android Application	FREE

Table 4: Division of Work

Member	Role
Jim Kolonis	Hardware Management
Jon Kosir	Programming; Application Management
Jordan Ma	Resource Management
Drew Sanders	Programming Management

Discussion of Ethical, Social and Economical Issues

In the creation of our Pulse Monitoring system, we considered several other issues in addition to the design of the system.Our target audience is very large, consisting of athletes, those exercising for leisure, as well as medical patients. From an economical standpoint, our system is fairly inexpensive. Currently, all the parts total to $49.79. The cost could be further reduced by utilizing just the ATmega microprocessor instead of the proprietary Arduino device that surrounds the microprocessor.

From a social and ethical standpoint, our system offers two benefits: portability and privacy. Our system's portability is beneficial to patients that need to monitor their heart rate in a clinical setting, but would like to spend time away from their bed or room. Our system's privacy is beneficial to users that would like to keep their health information private, as it can only be accessed on the user's mobile device once they set the Bluetooth device PIN.

Conclusion

This report outlines the design process for an infrared heart rate sensor with wireless data communication capabilities. The sensor is passed several thresholds and executes functions based on a software outline.This outline - in C code - is uploaded to a prototyping board and processed by an AT Mega 328P. Using Bluetooth, data is sent to an Android device where the heart rate is displayed in beats per minute.

Future Work

While we are currently finished with our embedded design project, we believe there is plenty of opportunity for future work with this particular system. Several improvements to both the software and the hardware could be made, in addition to what has already been constructed. In future projects, the design of our own project could be implemented, and also include other sensors that can read and then display the other vital signs of the patient (such as temperature, blood pressure, cognitive responses, ect.).

Other improvements that could be made in future projects that can further build upon our current project could include: improvements to the physical enclosure (perhaps a company-made, mass production model), further minimizing the arduino circuitry and component arrangement, further application development (for other phone models outside of Android), software improvements (minimizing and clearing the code, improving efficiency, ect.), and finding a longer-lived power source to provide adequate power for the entire system. With all this room to experiment and improve upon our existing system, we believe that there is great potential in investing in such mobile devices as they will provide new opportunities and advantages for the medical technology community and beyond.

References

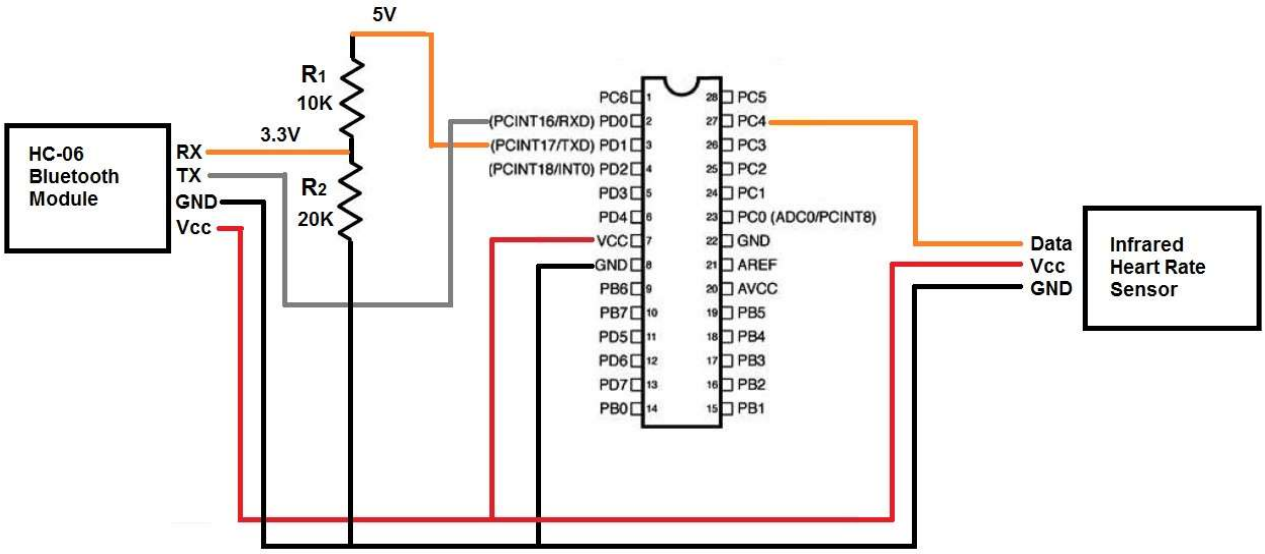
http://medicarduino.net/

http://www.seeedstudio.com/wiki/Grove_-_Ear-clip_Heart_Rate_Sensor

https://developer.android.com/training/index.html

http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

Appendix A:


5V
R1
10K
3.3V
R2
20K
HC-06
Bluetooth
Module
RX
TX
GND
Vcc
PC6 1
(PCINT16/RXD) PD0 2
(PCINT17/TXD) PD1 3
(PCINT18/INT0) PD2 4
PD3 5
PD4 6
VCC 7
GND 8
PB6 9
PB7 10
PD5 11
PD6 12
PD7 13
PB0 14
28 PC5
27 PC4
26 PC3
25 PC2
24 PC1
23 PC0 (ADC0/PCINT8)
22 GND
21 AREF
20 AVCC
19 PB5
18 PB4
17 PB3
16 PB2
15 PB1
Data
Vcc
GND
Infrared
Heart Rate
Sensor