Atrial Fibrillation Detection Device

Design Document

Advisor: Mani Mina
Client: Dr. Rakshak Sarda
Contact: Benjamin Sjulson

Group:

Ian Abbott
Nicholas Dubois
Aaron Melcer
Donathan Morgan
Keegan Mumma
Seth Rickard

Table of Contents

Project Statement

With an ever increasing population comes an increased demand for healthcare. To accommodate the higher need, low cost and easy access testing equipment will become essential. Atrial fibrillation is an often undiagnosed heart condition that affects many around the world and is especially hard to detect in rural areas where expensive testing equipment is not always accessible. However, one piece of technology reaches almost every corner of the world: the smartphone. Our aim is to create a low cost heart monitor to pair with an Android phone to detect atrial fibrillation. These monitors could then be distributed with the free companion application to collect data and report on the subject's heart condition.

System Requirements

The solution will consist of four components: a power supply, an amplifier circuit, a bluetooth controller, and an Android application. The power supply will provide power to the amplification circuit and the bluetooth controller. The amplifier circuit will amplify and smooth the subject's heart rate for the bluetooth controller. The bluetooth controller will read the value from the amplifier circuit and send the digital value to the Android phone for processing by the Android application.

Power Supply

Functional

- Includes recharge circuit/rechargeable supply
- Provides 3.7V at 3800mAh
- Corresponds appropriately with the microcontroller unit (MCU) and instrumentation amplifiers

Non-functional

- Fits within a "pocket-sized" enclosure

- Light weight

Amplifier Circuit

Functional

- Modifies the low voltage heartbeat signal to be read by the MCU at a higher potential
- Low power for less draw on the battery circuit
- Low-pass filter for signal buffering
- Safety circuit to prevent damage to device or user injury
- Life span to accommodate lengthy use

Non-functional

- Occupies minimal area
- Utilizes low cost components

Bluetooth Controller

Functional

- Connect the bluetooth radio to the Android device
- Digitize the analog signal from the amplifier circuit
- Buffer the data when the Android device is disconnected
- Intelligent data summation of periods in between heartbeats

Non-functional

- Low power consumption
- Embedded program error resistant

Android application

Functional

- Connect to the bluetooth monitor and control the connection
- Record the values from the monitor and store them in a database
- Display the current monitor output as a graph
- Display old values from the monitor stored in the database
- Compress data at regular intervals to reduce storage size

Non-functional

- UI / UX follows android best practices, material design
- User interface is “intuitive” as it is possible that this will not be used by English speakers
- Abstracted strings for easy translation
- Quick response when pulling up large sets of data

System Analysis

The system's main objective is to be a low cost and reliable heart monitor for deployment in low income areas of the world. Therefore, components will be chosen for their durability and low price point. The final product should be able to be shipped across the globe, withstanding the harsh conditions of transport and use in these environments. As this is an embedded system with an unknown repair schedule it is assumed that the device must operate as long as physically possible. To ensure this, the software on the device must be extremely robust and fault tolerant. These factors will be taken into consideration in every facet of our design.

Cost

One of the major considerations for this project is the cost of the final product. If this solution is to be sold or given away in low income areas, then the components must be inexpensive to manufacture. The amplification circuit is comprised of commercially manufactured components and the primary part of the bluetooth connection subsystem is a Texas Instruments chip, both of which are mass produced, keeping the cost low. The largest cost is the Lithium-Ion battery used to power the circuit. The Android application, once programmed, will be distributed for free of charge on the Google Play store. These component selections were made in order to make the product available to the largest number of people.

Environment

After cost, the next consideration is the dependability of the system. The software for the bluetooth controller will follow the JPL Rules for Developing Safety Critical Code to increase the likelihood that the embedded code will have the longest uptime possible. This robustness will be mirrored by the hardware design. Another consideration is the durability of the hardware. The circuit and microcontroller will need some form of enclosure to keep them safe from damage.

User Interaction

The design of the solution, both hardware and software, should be easy to understand and operate, because the heart monitor may be used by people whose first language might not be english. This will be accomplished on the software side by use of icons and simple material design. The strings used in the application will be abstracted into language files to aid in internationalization. The hardware side will be as simple as possible to attach the leads and

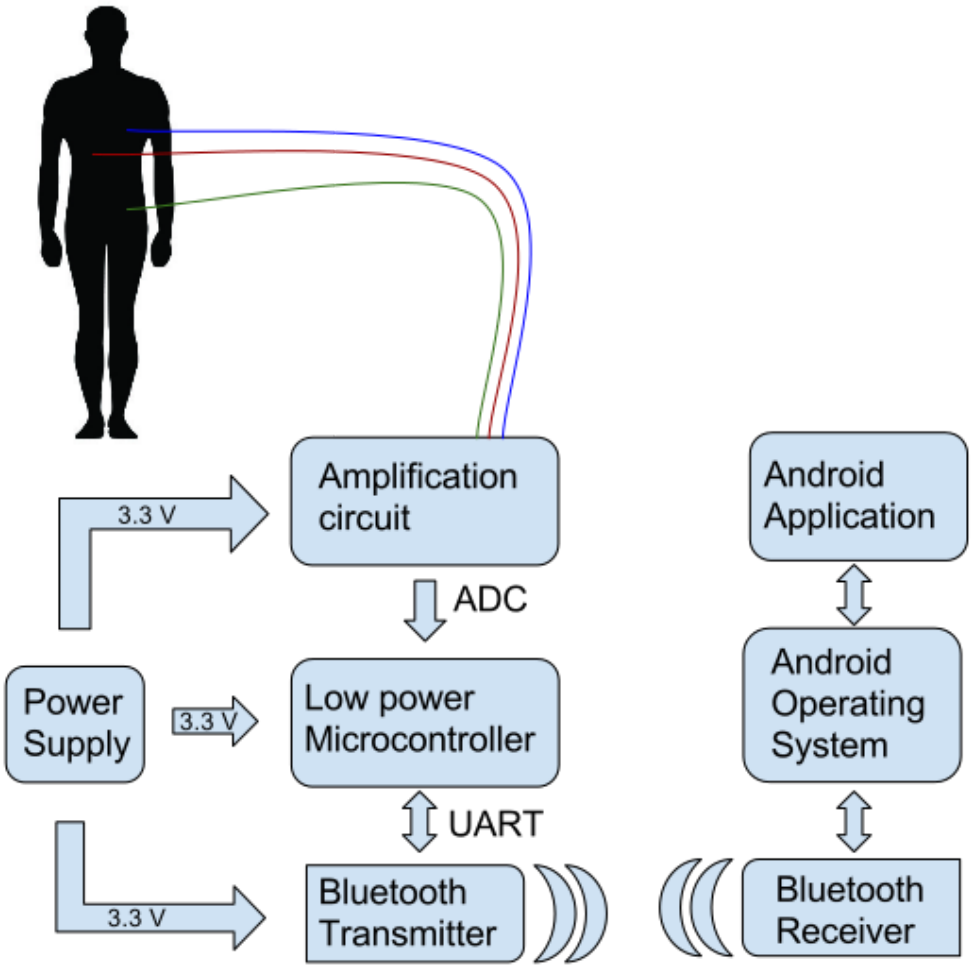
connect to the Android device. Pictures within the application will also help instruct how to operate the hardware and leads.

UI Design

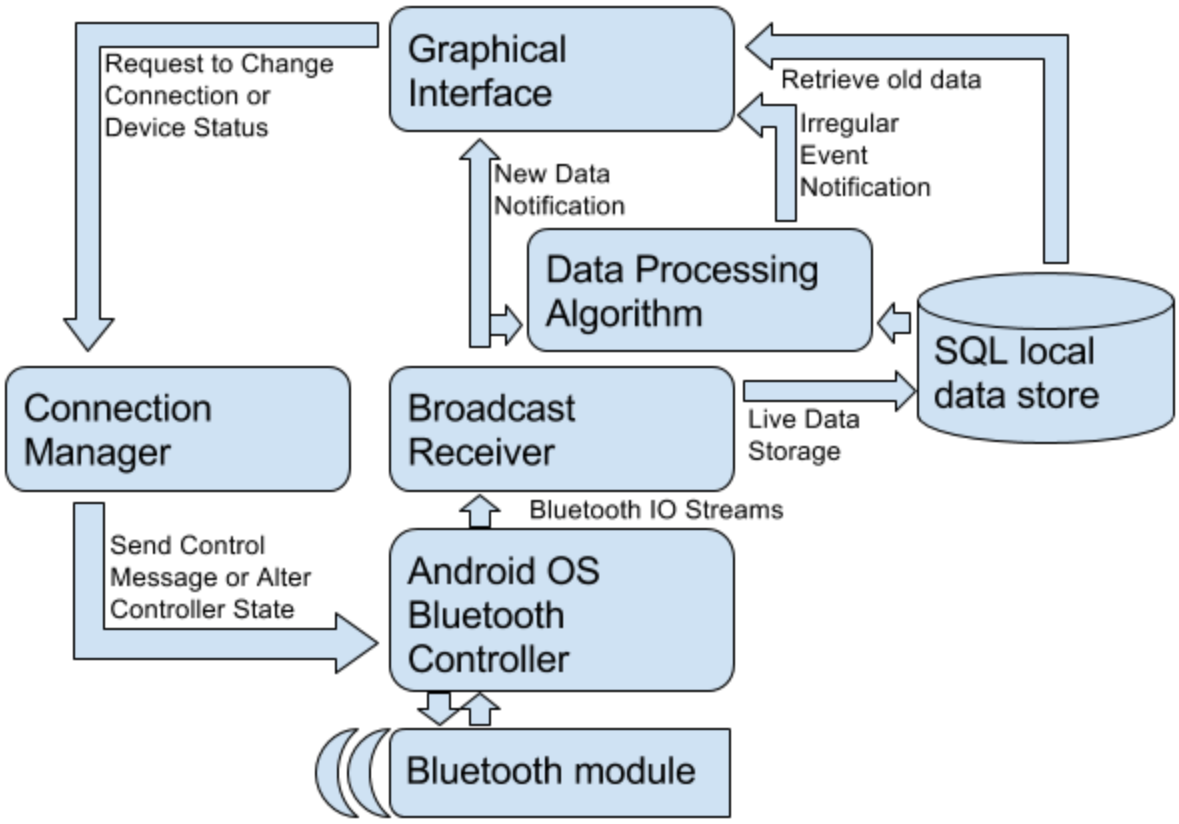
In order for the application to be most effective, it needs to be both simple and intuitive. In order to achieve this, we are going for a minimal approach in keeping a low number of direction options on each screen to create an easy navigation system. There is also a menu in the top right corner of each screen that will allow the user to jump to any screen from any point within the application. The actual heart rate will be represented in a graph that will be stored so the user can go back and view it at a later date, with atrial fibrillation detections being marked whenever they happen for easy viewing. For a visual representation of the application navigation, see the flow diagram, go to Appendix A. To see each screen individually, along with a description, go to Appendix B.

Block Diagrams

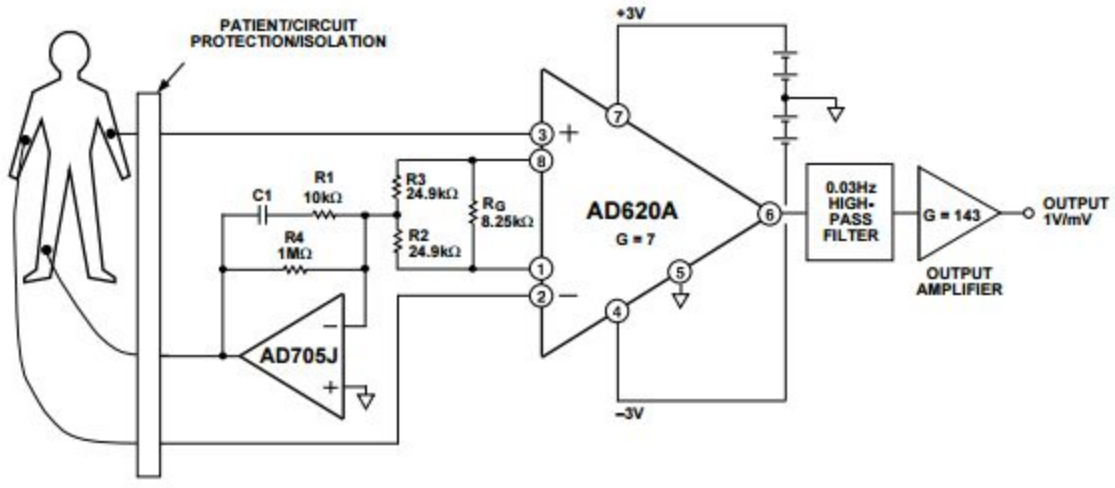
System Overview



Software Module Interaction



Amplifier Circuit Diagram



Standards

MCU

The Texas Instruments MSP430f5438a mixed signal processor was chosen as the microcontroller for the embedded monitor device. The msp-exp430f5438a experimentation board was chosen as the prototype development board due to the existing code examples for the board relating to the bluetooth controller.
http://www.ti.com/product/MSP430F5438A
http://www.ti.com/tool/msp-exp430f5438

Bluetooth Connection Core

We decided to go with the Texas Instruments cc2564moda as the final bluetooth controller for our design due to the available example material for the cc2564modnem evaluation board.
http://www.ti.com/product/CC2564MODA
http://www.ti.com/tool/cc2564modnem

Bluetopia

Texas instruments offers a royalty free Bluetooth SIG certified Bluetooth stack for use with their line of microcontrollers. This library was designed for the msp-exp430f5438 experimenter board by Stonestreet One.
http://www.ti.com/tool/tibluetoothstack-sdk

Bluetooth SPP

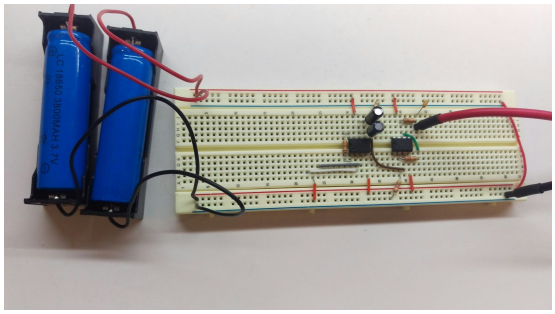
We chose to target the Serial Port Profile in the bluetooth standard due to its wide acceptance and more specifically its integration in the Android Development Kit.
https://developer.bluetooth.org/TechnologyOverview/Pages/SPP.aspx

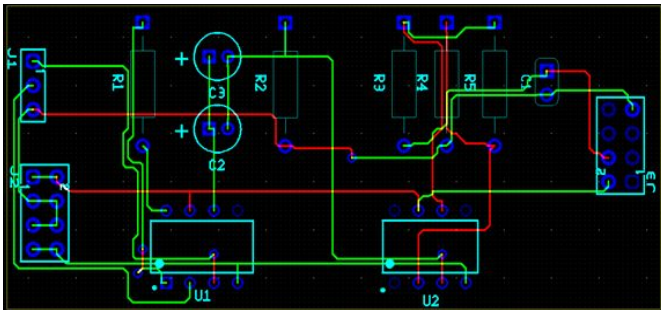
Android

The smartphone application was developed for Android using the ADK and Android Studio. This was chosen due to android's ease of development and quick distribution avenues.
http://developer.android.com/develop/index.html

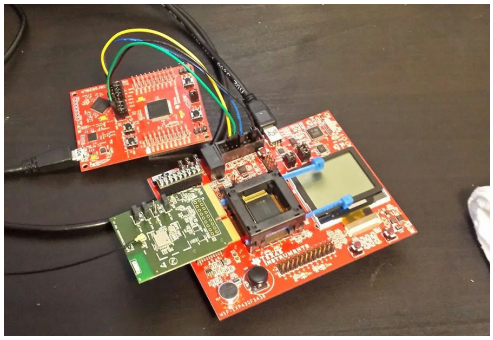
Implementation Details

Below is an image of the final circuit design on a Breadboard. Due to time constraints we were unable to have a PCB ordered and tested. A layout for the PCB was designed, and the layout follows the image below. The system is powered by two 3.7V Li-Ion batteries. Two integrated circuit amplifiers pick up the electro-cardio signal from the leads placed on the subject and amplify the signal. This signal is then sent to be digitized by the MCU.

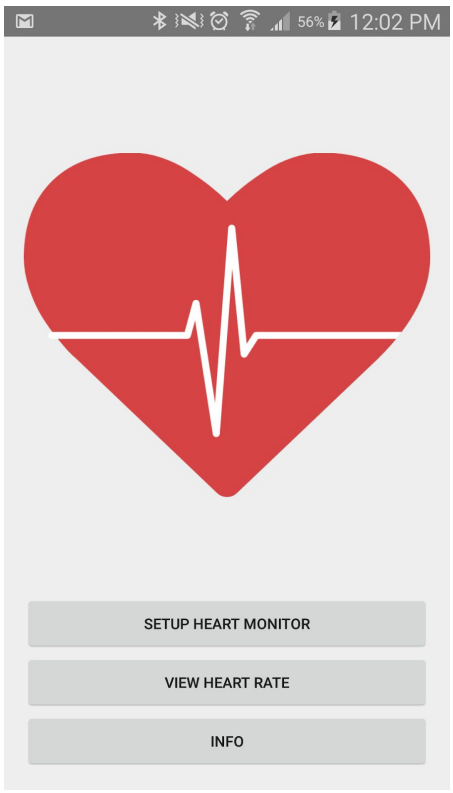


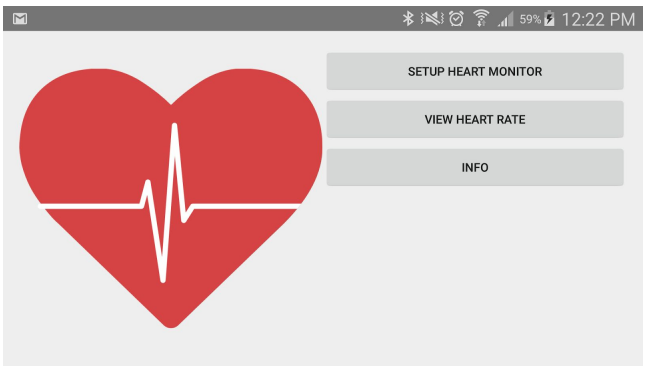


Below is an image of the msp-exp430f5438 experimenter board being programmed by the FET of an msp430f5529. Attached to the msp-exp430f5438 is the cc2564modnem evaluation daughter board. A layout of a PCB for the mcu and connection core was designed, but due to time constraints one was not able to be populated for testing.



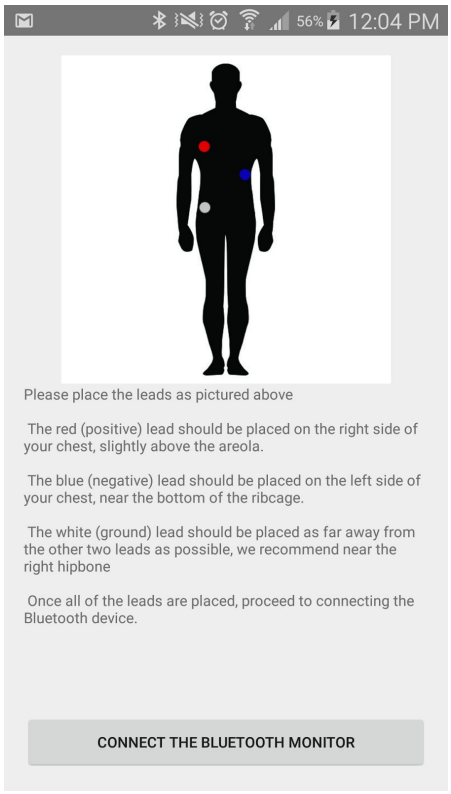
The android application consists of many screens. The easiest way to understand the implementation of the application is to look at each screen individually.

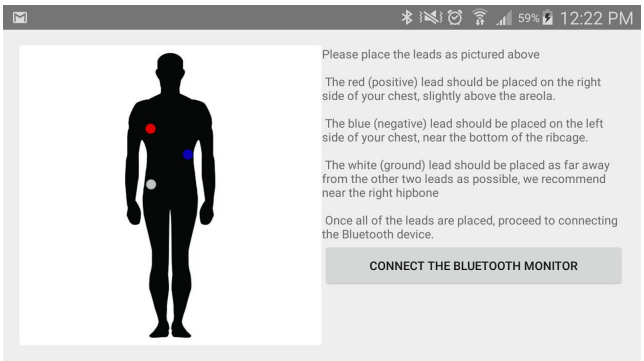




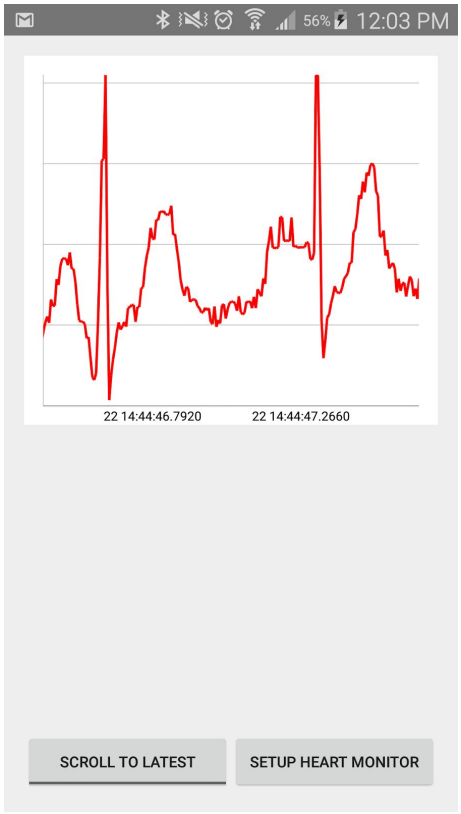
Home Screen: This is the first screen the user sees. It welcomes the user and prompts them with all of the possibilities.

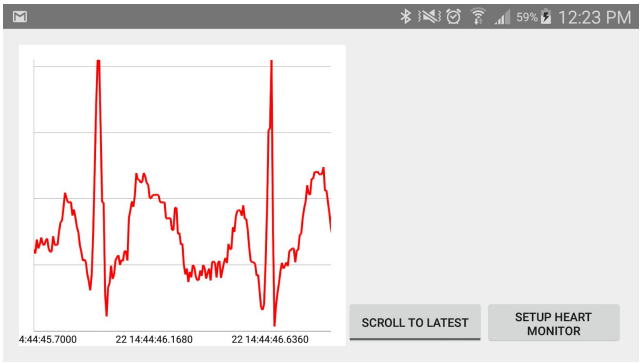
Buttons:
Setup Heart Monitor - takes the user to the setup screen, with information on how to setup the monitor and connect the bluetooth
View Heart Rate - takes the user to the ECG screen
Info - brings up a text box with legal information about the application





Setup Monitor: This screen informs the user of the proper placement of the leads. The picture above the text shows the approximate placement on a neutral form. This enables people of limited literacy to still place the leads with some accuracy.
Buttons:
Connect the Bluetooth Monitor - Brings up a screen for connecting the application to the bluetooth monitor





View Heartrate: This screen offers a live view of the values coming from monitor, or observe the values stored in the sqlite database.
Buttons:
Scroll to Last - this is a toggle button, in testing of the UI we found it impossible to have constantly updating values and scroll back to view earlier values. So when this toggle is set, the graph will update dynamically. And when it is clear, the graph will statically fill with all the data in the SQLite database.
Setup Heart Monitor - Takes the user to the setup screen.

Testing

The two major components of the system were tested functionally by passing data through. The amplification circuit, the Android application and MCU, and the whole system were tested on a lab bench.

Testing of the amplification circuit was begun by connecting the leads to one of the engineers as a test subject. The amplification circuit would then amplify the Electro Cardio signal, and the output was displayed on a bench oscilloscope.

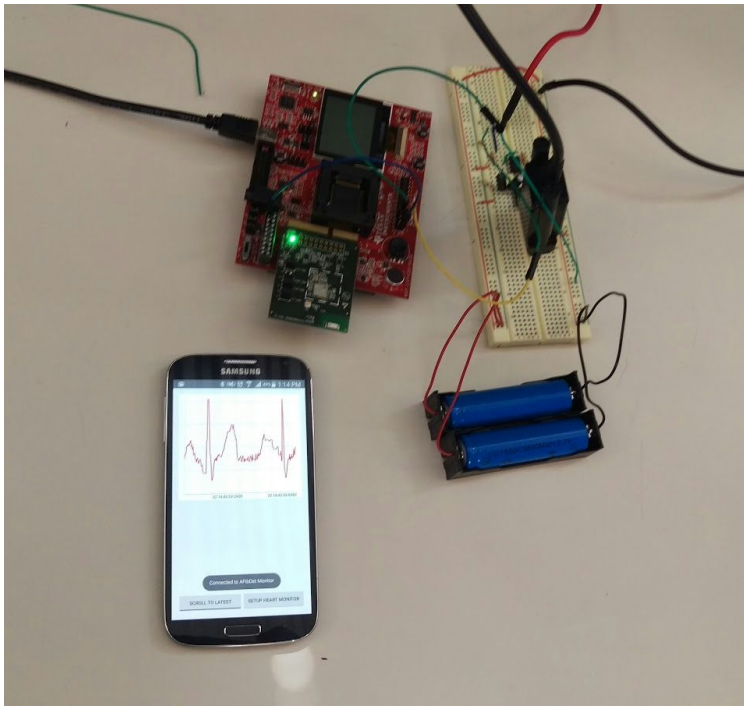
Testing of the Android application and MCU connection were done using a function generator. The function generator mocked the output of the amplification circuit to the MCU. This mock ECG was then digitized by the MCU and sent over bluetooth to the Android phone. The application then displayed the waveform live onscreen. This waveform was then compared with the mock waveform from the function generator.

The system as a whole was tested in much the same way as the constituent parts were. The leads were connected to an engineer and the amplification circuit. The amplification circuit was then connected to the MCU. The output was then viewed on the Android application to ensure the signal made the trip.

Appendix I: Operation Manual

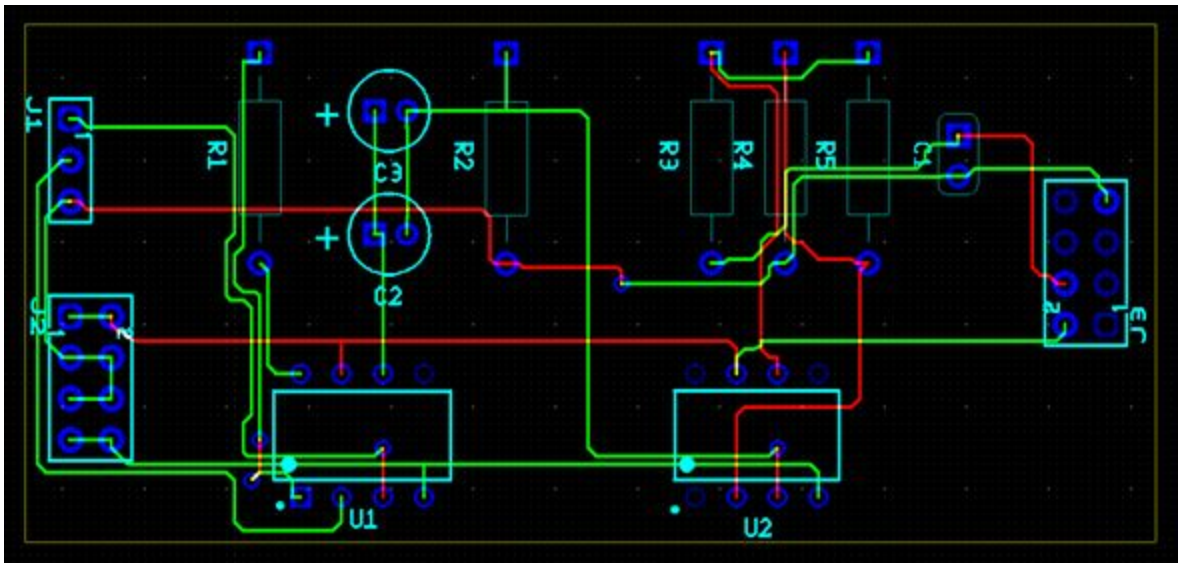
Demo Setup Instructions

Using Breadboard Layout



Using the breadboard as shown in the image above, the demo is carried out by simply attaching the electrodes to the body as shown in the android application, then connecting ground from the breadboard to the ground pin of Port X.Y on the Experimenter Board, and finally connecting the output of the circuit (the farthest right blue wire) to 6.7 of Port X.Y on the Experimenter Board. In order to run the demo, we checked using an oscilloscope to be sure that we were receiving a good signal from the detection circuit while at the same time checking for the signal on the application.

Using PCB



Due to the fact that we ran out of time to complete a full board-integration, the project must be demoed using the detection circuit board as shown in the layout above. Firstly, an adjustment to the board must be made in which Pin 2 of J3 is removed so as not to make connection with the Experimenter Board (this connection results in a fried MCU). Connector J3 is then plugged into the Experimenter Board such that Pin 1 of J3 connects to 5.1 of Port XY. The electrodes must then be connected to the body as shown in the android application, then connected to J1 such that the positive electrode signal connects to Pin 1, the negative electrode signal connects to Pin 2, and the ground signal connects to Pin 3. Lastly, the two lithium ion batteries must be connected to Connector J2 such that the positive lead of one battery connects to pin 1 of the connector and the negative lead connects to pin 3. The second battery is connected with the positive lead to pin 5 of J2 and the negative lead to pin 7 of J2. The Experimenter Board may now be powered on and the demo is carried out using the android application connected via bluetooth to the Experimenter Board.

Programming the software

MCU programming

The code for the msp-exp430f5438 should still be flashed on the msp430f5438a in the socket of the board. However, in the case that the code is corrupted or new code must be loaded into the flash, here are the instructions for flashing the device.

NOTE: the msp-exp430f5438 experimenter board does not have an onboard programmer. For this there are two options for programming. The instructions that follow are using the EZ-FET of an msp430f5529lp to program the msp-exp430f5438. If you have access to a 14 pin TI usb FET

device and wish to use that, follow the instructions for your device to connect to the target board and skip to the paragraph regarding Code Composer Studio.

With both the msp430f5529lp and the msp-exp430f5438 powered off, connect the following pins using female-female jumper cables. (NOTE: let LP be the msp430f5529lp launch pad device, and EXP be the msp-exp430f5438). LP GND to pin 9 of the jtag connector on the EXP. LP 3V3 to pin 2 of the jtag connector on the EXP. LP RST to pin 11 of the jtag connector on the EXP. LP TST to pin 8 of the jtag connector on the EXP. Now the EZ-FET of the msp430f5529 is connected to the msp-exp430f5438 using the Spy-Bi-Wire protocol. Plugging the msp430f5529lp into a computer with a usb micro will now allow you to program the msp-exp430f5438. The device should be visible in Code Composer Studio.

The use of Code Composer Studio is recommended over IAR or other embedded platforms, due to most of the code being written with CCSv6. Open Code Composer Studio with an empty workspace. Then import the project by going to File > Import, then selecting Code Composer Studio > CCS Project, and navigating to the directory containing the code. The code for the embedded device may be obtained as a zip from our project website. From there the project can be loaded and debugged as normal using the debug button near the top of Code Composer.

Android programming

Loading and debugging the android application requires an android device running at minimum SDK version 15, while SDK version 23 is recommended, and Bluetooth connectivity is required. Connect the Android device to a computer using a micro usb cable.

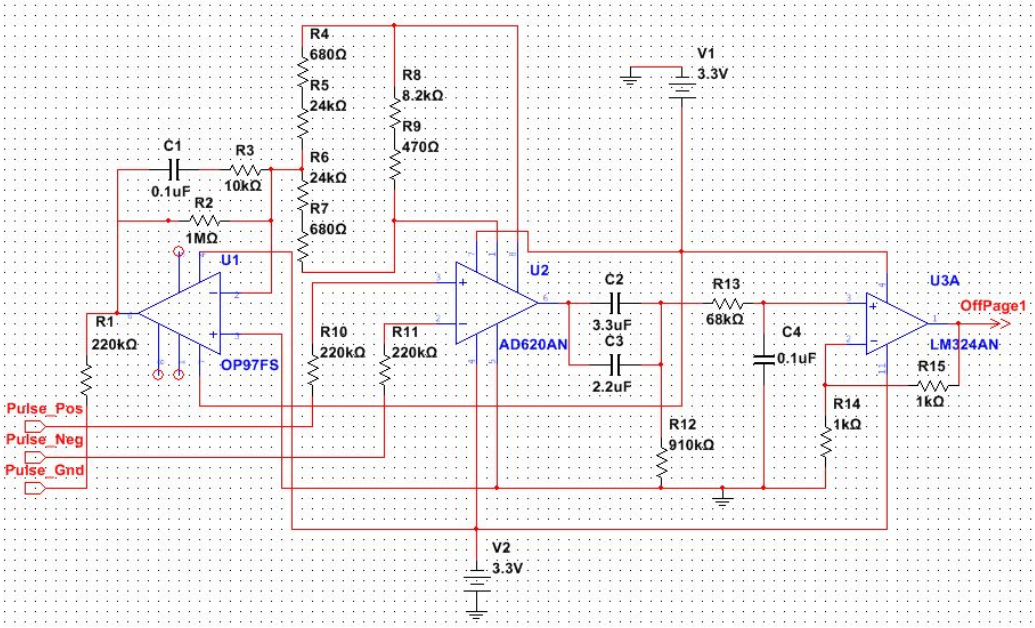
Open Android Studio and select Open Existing Project, then navigate to where the android code is stored on your machine. A zip of the project may be downloaded from the group website. Once loaded into Android Studio the application can be loaded and run onto the target device using the green triangle at the top of Android Studio. Make sure to select the correct device to program. Please note that building the Android application requires an internet connection as MPAndroidChart is included in the gradle build but not the repository, so building the application will download MPAndroidChart from a build repository.

Once the application is loaded onto the target device. Navigate to connecting the bluetooth monitor and select the device labeled AFibDet. A prompt for password may appear, this is because insecure pairing is not allowed due to security concerns, enter passcode 0000. Then navigate to the view heart rate activity to see the output of the monitor.

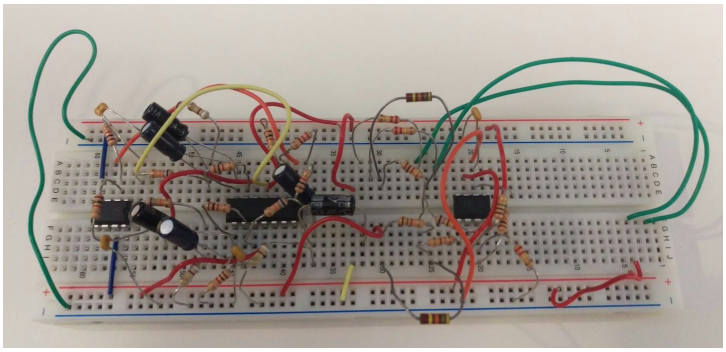
Appendix II: Alternative Design Versions

Amplification Circuit

There were several versions of the hardware design before the final design was settled on. The initial designs stemmed from a suggested circuit from the datasheet of the AD620 instrumentation amplifier which we used as the basis for our design.



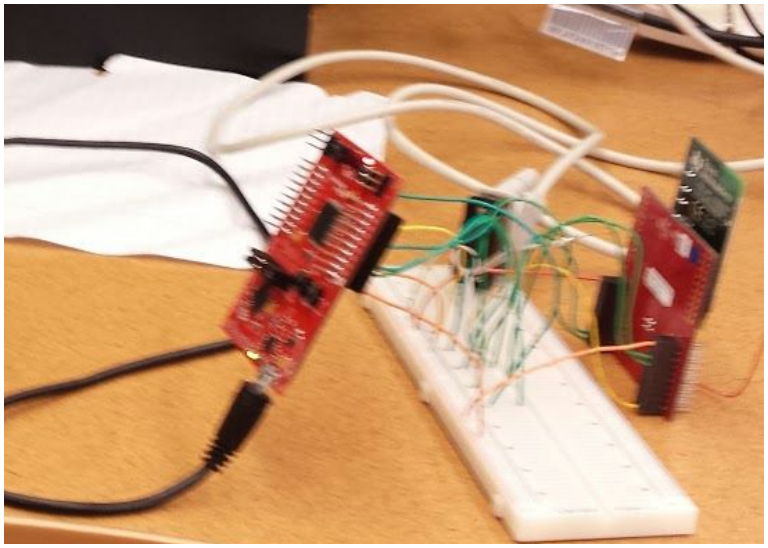
The original circuit design, as seen above, included a right leg drive (built around an OP97FS op-amp, U1) and included several off page amplifiers to attempt to bring the signal to a usable value. Though this design may have worked had we been able to establish a better connection on with the electrodes to the body, it also took up more space and contained several unnecessary components. This was designed based on the datasheet suggestions of the AD620 as well as other sources found whilst researching ways to detect a heart signal. The picture below shows another issue that may have been a factor, where we did not build the circuit cleanly and instead had a nest of wires and components that may have been acting as antennas for noise.



Many of our next attempts followed the same design as the one seen above, including a right leg drive and multiple amplifiers. Once we began to build strictly around the AD620 and attempted to first find a signal using only this component and the appropriate resistors, we were able to add on to create our final design. Though the right leg drive did help to alleviate some of the noise from the signal, it was not enough for us to consider it necessary.

MCU/Bluetooth Controller

The microcontroller and bluetooth controller had one major version prior to the current configuration. The original pair had the msp430f5529lp board with an EMadapter boosterpack, and the cc2564modnem evaluation daughter board (pictured below). Prototyping using this setup would have been significantly cheaper than using the msp-exp430f5438 because the launchpad did not have an onboard screen, joystick, speaker/amplifier, and accelerometer. It was believed that this configuration could be workable due to the existence of example code for the msp-exp430f5529 which had a different pinout, but the same MCU. Development with this setup was scrapped when it was found too difficult to adapt an unknown codebase to a foreign processor pinout, using an adapter that was not designed to interface with the chosen evaluation daughter board. In the end this endeavour was not a total waste as the onboard EZ-FET of the msp430f5529lp was used in lieu of an expensive TI jtag to program the msp-exp430f5438.



Android Chart

The original chart engine chosen for the Android application was GraphView. GraphView is a simple graphing option, available under the GPL version 2.0 with the linking exception, made by Jonas Gehring. It was chosen for our application due to it's ease of insertion and deletion of data points. This was key to the dynamic nature of our screen refreshing. Unfortunately, the onscreen rendering was done using Java canvas and paint objects. This simple approach to rendering resulted in immense rendering times for our datasets.

The application was then altered to use MPAndroidChart. MPAndroidChart is an open source graph engine, available under the Apache Licence Version 2.0, made by Philipp Jahoda. This library uses Android APIs for OpenGL and EGL to offload most of the work to the phone's GPU instead of the CPU. This allowed us to display our live dataset on screen without setting the phone on fire.

Appendix III: Other Considerations

Time was our major roadblock for this project. We expected our first design to work every time. This lead to delays in every aspect of our design. The microcontroller and bluetooth controller were not able to be mounted on a PCB because the original design changed. The PCB for the amplification circuit was not able to be printed and populated in time to be tested because the amplification circuit went through so many changes.

The major lesson learned by every member of the team is the interplay between every piece of the design, and the necessity to begin work as soon as possible. The importance of time management and the subdivision of labor cannot be stressed.

Appendix IV: Code

Due to the size of the code files, we will not include them verbatim here. Zipped project files with Git repos are available on our project website.