Android Electro Cardio Monitor

Jeffrey P. Frye, Jonathan D. Gibson,
Michael W. Sun, Cheng-Chieh Wang

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Android Electro Cardio Monitor (AECM) is a portable electrocardiograph system that can monitor a patients' heart rate with an Android mobile device. The main system contains an analog Front-End that captures the heart rate signal and wirelessly transfers the data using Bluetooth and an Android application that processes the data in real time and displays it on a graph. In addition to that, a Heart Rate Simulator with real previously recorded ECG data is used for this project.**

***Index Terms* — Biomedical Signal Processing, Bluetooth, Electrocardiography, Microprocessors, Operational Amplifiers, Portable Computers**

I. INTRODUCTION

Medical facilities across the world are becoming over crowed and they lack the necessary tools to efficiently diagnose patients. One such issue is monitoring patients' heart signal (electrocardiography). ECG monitors are expensive and typically can only monitor a single patient, making it costly to continuously monitor individuals while they are in the hospital. With the increasing number of people visiting physician and emergency departments, medical professionals need a new strategy for recording the vital signals of their patients. In the digital age that we live in, doctors and nurses in the medical field need to be able to quickly monitor patients' data, write down notes, and store this information in a timely fashion. The Android Electro Cardio Monitor (AECM) is the solution to this arising problem.

AECM simplifies the experience of getting an ECG for both the doctor and the patient. AECM uses an Android mobile device as a monitor to display the patient's heart rate signal. The application is user friendly and easy to use, which allows for not only doctors to be able to navigate it but also patients. This benefits patients, especially those with heart problems, because now they can simply take an ECG from home and send the recorded graphs to their doctor to analyze without having to go into a hospital.

The AECM hardware contains a heart rate simulator, analog front-end, and a Bluetooth module. The heart rate simulator serves as a system test, to ensure that the data being displayed through the application is accurate. The analog front-end is the hardware that the electrodes are plugged into and then attached across the heart. The Bluetooth digitizes and transfers the captured heart rate signal from the analog front-end to the Android device. All the contents of this system end are small enough to be carried around in a pocket.

II. SYSTEM OVERVIEW

A. Heart Rate Simulator

The ECG simulator is the forefront of the project since we have decided to not use a live person as our patient to retrieve the heart rate signal from. Since a person is not being used, an ECG simulator subsystem is a necessity. Data is available from an online database with individual patient's heart rate signals. This data can be downloaded as a text document with the data shown as samples. These values are from data that is sampled between 100 and 300 times per second. This data is stored onto the flash memory of a microprocessor as different heart rate signals can be selected by the user.

The ECG simulator uses data from PhysioBank, which was from an online database source. These databases were mainly developed at MIT and have been made available to the research community with support from the National Institutes of Health. Using these databases requires software to read and write the heart rate signals using their own functions which are part of the WFDB library. Although the software was free and available, the data files that we need are much smaller and can be downloaded directly from the website.

To use these databases, the PhysioNet website had their own GUI where we select the database that we are interested in. Then we choose a time frame to view and see the electrocardiogram.

Since we are creating an ECG Simulator, the ability to switch between three different signals is enough to check that everything is working correctly. From the large quantity of available ECG files, the ECG databases we are using in our simulator were selected so that an entire range of ECG signals could be used to test that the back end of the project, which includes the filters and circuits, handles several variations of signals. The three selected ECG signals are from the MIT-BIH Normal Sinus Rhythm Database, MIT-BIH Arrhythmia Database, and the QT Database.

When choosing a device to operate the simulator, several factors relating to the project came into play. First of all, since the simulator portion of the project should be reasonably small and hand-held. The simulator must come in a very small package and if needed can sometimes be operated by using low-power batteries. The microprocessor has to read in the data from the ECG data source. It has to be able to store the thousands of lines of data for each of the sample signals. Also, the chosen microprocessor needs to have enough pins to connect and interface with outside systems. For example, the microprocessor will be taking input from at least two push buttons. These push buttons will be used to scroll down to switch between the different records, while the other button will be used to select the signal that would be shown further down the line. The microprocessor will also be connected to an external display for showing the signal label when the user is selecting the exact signal to simulate. This display will be showing a label of the different sample signals the user is selecting between. Also, if any additional memory or converters are needed, pins will be used for this too. With all of these combined, our project needs at least twenty additional pins on the microprocessor.

The original MSP430 LaunchPad initially came to mind. Members in our group have already had previous experience programming the Launchpad in our embedded systems class and a workshop and are familiar with Code Composer Studio. Also, the Launchpad has an on-board emulation so no additional tools are needed to program and debug the project. The MSP430 LaunchPad uses ultra-low power operation and can be used for battery-operated applications which is feasible for our project at hand. In addition, MSP430G2553 came with DIP package, which is convenient for prototyping. The main features for this MSP430G2553 are shown in the Table 1.

TABLE I

MAIN FEATURES OF THE CHOSEN MICROPROCESSOR

CPU Speed	Up to 16 MHz
Flash Memory	16 KB
Analog to Digital Converter (ADC)	8 channel, 10-bit
Random Access Memory (RAM)	512 B
Communication	Up to 1 I2C, 2 SPI, 1 UART
Pins	20 pins
Timers	2 16-bit Timers

For our project, an LCD display is a requirement so that the user is able to visualize and see the label of which signal they are selecting to use for the simulator. The optimum words and phrases our group wants to be displayed are the number of the selected signal as well as a label of the title of that signal. For example, the label “Signal 1: MIT-BIH Arrhythmia Database” is the portion displayed. Since there are several different sizes of displays available, the text is shown as scrolling so that all words are still readable in a short time frame.

To output an analog heart rate signal using the individual signal's data from the online database, we thought of a different method since our microcontroller does not have a digital-to-analog converter (DAC) onboard. We decided it is much cheaper to use pulse width modulation (PWM) to implement a DAC. The resolution will be just eight bits, giving us 256 possible PWM widths. After the microcontroller outputs the pulses, the signal goes through low pass filters to filter out frequencies above about 150 Hz.

After converting the digital signal to an analog signal, we can obtain the simulated ECG signal. Nevertheless, the simulated ECG signal will be in the range of 0 to 5 V. This signal cannot be directly fed in our ECG reader because the ECG reader will not only clean the common-mode voltage, but it also amplifies it with the total gain of at least 1000 times. If the simulated output signal is amplified, there are two problems that occur.

1) The simulated output signal is clipped off by the power source of the gain amplifier, which means we will not be able to see anything.
2) The simulated output signal successfully going through the ECG analog front end. Nevertheless, the simulated output signal is amplified by 1000 times. The microcontroller we will be using is fried due to the high amplified simulated signal.

Therefore, we need to downgrade our simulated output signal. The method to downgrade the simulated signal is using an attenuator. The simplest attenuator can be created by voltage divider networks.

First of all, the simulated ECG signal is in the range of 0 V to 5 V. The actual ECG signal measured with the oscilloscope will have some negative components. First, we need to get rid of the DC offset of the simulated ECG signal. It is convenient to discuss the attenuation in terms of decibels relative to power. Assuming the amplitude of simulated ECG signal is 5 V, we need the ECG signal to be around 5 mV. This means the attenuation is

$$dB = 20\log\frac{Vi}{Vo} = 60\,dB. \quad (1)$$

We designed the following circuit shown in Fig. 1 to do this job. To attenuate the desired simulated ECG signal, we have a non-ideal second order low-pass filter with cutoff frequency around 80 Hz. Then a huge capacitor works as the DC filter. Finally, a simple voltage divider will attenuate the signal to whatever we wish to get in the end.

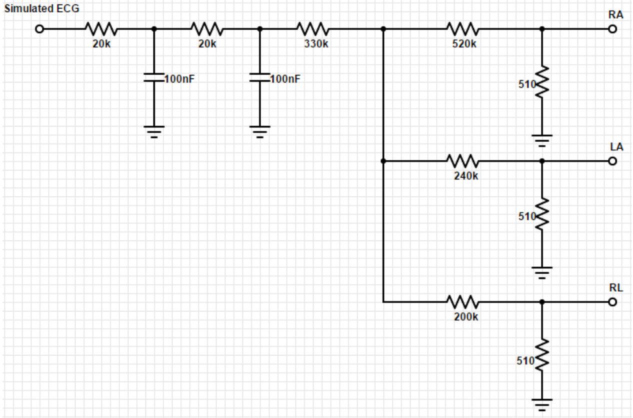


Fig. 1. Attenuator Design

To test the heart rate simulator subsystem, we used an oscilloscope to see what one of the heart rate signals looked like after it had gone through the attenuator design. A test signal from the MIT-BIH Arrhythmia Database is shown in Fig. 2.

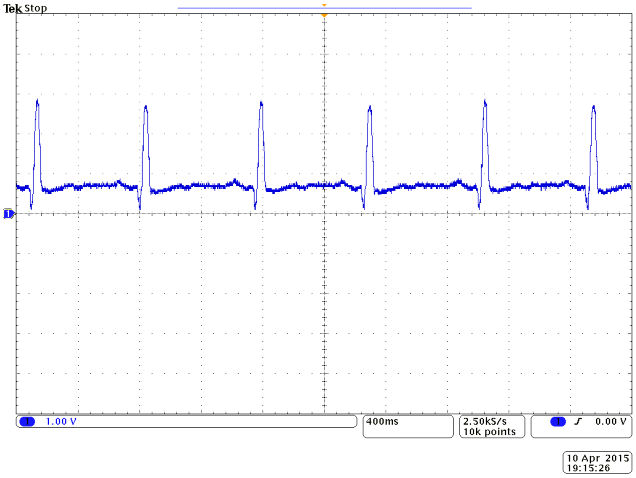


Fig. 2. Heart Rate Simulator Output

B. Analog Front-End

A good ECG signal measurement amplifies the small biopotential signal and filters the external and internal noise out. The main core of the analog front-end is divided into several subsystems: electrode, instrumentation amplifier, bandpass filter, notch filter, gain amplifier, and microcontroller. Following are the basic requirements.

1) Take a weak electric signal of biological origin and increase its amplitude with an amplifier. (Desired output between 0 – 5 V)
2) High input impedance so that ECG amplifier provides minimal loading to avoid distortion of the signal. A typical input impedance is 1 MΩ.
3) The input circuit must provide protection and minimize current flow appearing at the input terminals.
4) The output circuit that is used to drive the amplifier load should have a low output impedance in order to minimize the load regulation effect.
5) The ECG amplifier must be designed to be optimal in a particular frequency range as needed by the signal to obtain optimal signal to noise ratios.

A 3-lead electrocardiogram system is used in this project. The 3-lead system is the simplest ECG design and can be easily realized using basic circuit designs. The 3-lead system actually uses 3 electrodes unlike the 12-lead system that uses 10 electrodes. There were many placement options used in the 3-lead design. One way is placing an electrode on the interior side of each wrist. The final electrode is placed on either the left or right ankle. This placement of the electrodes creates what is called an Einthoven's triangle.

Three separate ECG graphs are derived from this placement depending on which electrode was designated to be the ground potential. The difference in voltage potential between the left and right arm electrodes with the leg electrode, designated as ground, is referred to as "Lead I". "Lead II" is the difference in voltage between the right arm and leg electrode, with the left arm designated as ground. Accordingly, "Lead III" is the potential difference between the left arm and leg electrode with the right arm referenced as ground.

The second stage is an instrumentation amplifier (INA), which has a very high Common-Mode Rejection Ratio (CMRR) and can also obtain a very high gain. A typical ECG signal varies from the microvolt to the millivolt range with the frequency range of 0.05 Hz to 150 Hz. Nevertheless, one of the biggest challenges of detecting small AC signals such as ECGs was that the ECG signal is often superimposed by a large electrode DC offset potential.

There are several INAs available in the market. Instead of using three op-amps to build an INA, we used one optimized instrumentation amplifier. For this project the INA128 was chosen. The INA128 is a low power, general purpose instrumentation amplifier that offers excellent accuracy. This instrumentation amplifier has a low offset voltage of 50 mV maximum with a low drift of 0.5 mV/°C max and low input bias current of 5 nA. It has a high CMRR of at least 120 dB and the inputs can be protected to ±40 V. It is known for having a wide supply range from ±2.25 V to ±18 V and a low Quiescent current of 700 mA. This all comes in an 8-pin plastic DIP.

A problem that needs to be addressed when designing the analog front-end is limiting the amount of current that can be drawn from the input of the instrumentation amplifier. Particularly there is a concern of electrical shock that might occur due to the fact that the amplified signal is being drawn from a living organism. A good designer must be concerned about prevention of macro and micro electric shock. A circuit protection/isolation design would provide protection.

According to Electronics Laboratory Safety Information and OSHA, the typical “cannot let-go” current (the current that makes a person lose muscular control) is about 6 - 30 mA. Even 1 mA could still be dangerous under certain conditions. For example, under dry conditions, human skin is very resistant. Wet skin dramatically drops the body’s resistance. We must design an input protection to limit the input current or over voltage to protect the ECG circuit against high voltages which occur. This can occur in the operating room when the ECG signal is combined with the use of an electrosurgical unit that will induce high transient voltages into the patient.

Many factors can cause signal interference in the analog front-end circuit, and careful analysis must be done to limit interference. There are many different ways noise can be introduced into a system, but for the case of designing circuits to measure signals in the body, the main interference that will be seen is from industrial noise and electrical signals in the body. The challenge was to filter out the interference and isolate the desired signal in the body to be amplified.

A notch filter efficiently filters out the power interference. A notch filter is a band-stop filter or band-rejection filter with a narrow stop band. To filter out the power interference for the United States, we designed a filter that passes all frequencies, except for the range of 59 to 61 Hz. This would be used to filter out the main hums from the 60 Hz power line, though its higher harmonics could still be present. The notch filter circuit used in our project is shown in Fig. 3.

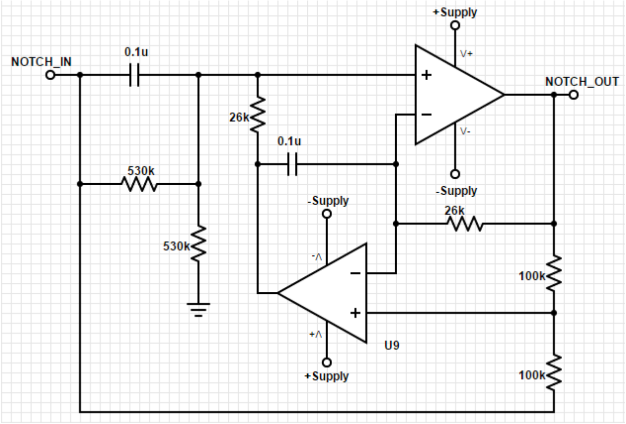


Fig. 3. Notch Filter Design

The body has many different electrical signals and isolating signals of interest (eg. Heart Beat) can be challenging. Everything from the brain, nerves, and muscles uses electrical signals to function. To limit interference from body functions that are not of interest, electrode placement needs to be carefully considered. The ideal placement of electrodes for an ECG measurement would be nearest to the heart.

The ECG signal varies from frequency 0.05 Hz to 150 Hz. A bandpass filter that constructs a low-pass filter cascaded with a high-pass filter would make a perfect bandpass filter. The high-pass filter comes first, so energy from it that stretches to infinite frequency will be low passed. The low-pass filter was designed to have a cutoff frequency of 150 Hz. The 4th order high-pass filter was designed to have a cutoff frequency of 0.05 Hz. Fig. 4 is the desired frequency response, where the red cursor is at 0.05 Hz and the blue cursor is at 150 Hz.

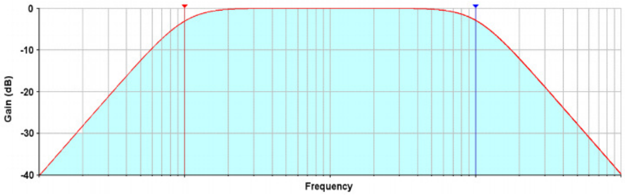


Fig. 4. Bandpass Filter Frequency Response

A second order high-pass filter topology is shown in Fig. 5.

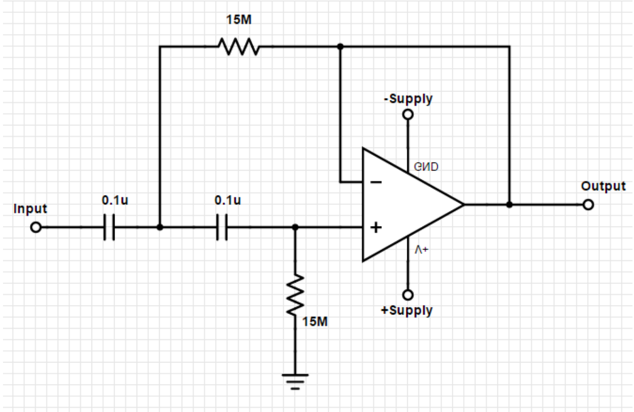


Fig. 5. Second Order High-Pass Filter

Next, we designed the 4th order low-pass filter. A second order low-pass filter is shown in Fig. 6. Connecting digital circuitry to sensor devices is simple if the sensor devices are inherently digital themselves. Switches, relays, and encoders are easily interfaced with gate circuits due to the on/off nature of their signals. However, when analog devices are involved, interfacing becomes much complex. What is needed is a way to translate analogy signals to digital numbers (binary).

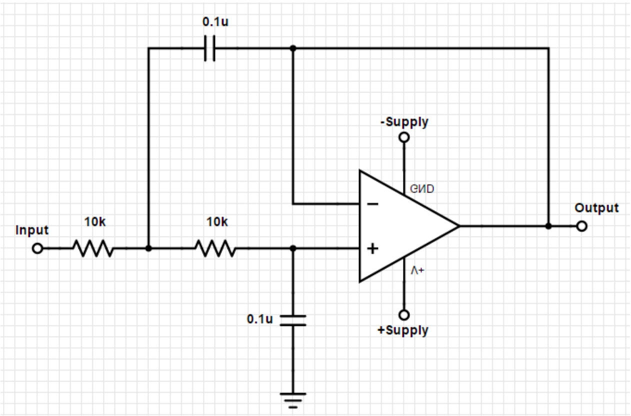


Fig. 6. Second Order Low-Pass Filter

An analog-to-digital converter (ADC) can input an analog electrical signal such as voltage or current and outputs a binary number. The simplest ADC is formed of a series of comparators, each one comparing the input signal to a unique reference voltage. The comparator outputs are connected to the inputs of a priority encoder circuit, which produces a binary sequence of output.

Another way to work with analog-to-digital converter is using a microcontroller. We decided to use the microcontroller chip such as the ATtiny85. The chip has an 8-channel, 10-bit ADC in quad flat (TQFP) and quad flat no-leads (QFN)/ micro-lead frame (MLF) package and 6-channel 10-bit ADC in Dual in-line package. It has master/slave SPI serial interface, and built-in clock distribution system.

The sampling rate of an ADC impacts greatly the accuracy of measurement of various ECG parameters. However, according to an online article, Digital Sampling Rate and ECG Analysis, the statistical study showed that no significant difference between results obtained at sampling rates of 500 and 250 samples per seconds. The recording made at 125 samples per seconds could be used in particular conditions when only the measurement of certain intervals is required. Therefore, more samples from the analog signal will not hurt.

In our application, we chose to send the data through a Bluetooth module. A Bluetooth module operates on the serial communication. Therefore, we must send the digital samples in a serial of binary codes in order for a Bluetooth module to be recognized. This means the speed of the ADC sending data is critical. Although there are a lot of IC ADC in the market, many ADC in the market require an external clock to operate, which mean they could operate at a different clock frequency.

The ATtiny85 met all of our requirements. In addition to that, an ATtiny85 chip costs around $2.84. Comparing with the price of ADC, a priority encoder, and SPI module, it would be about the same price. Nevertheless, each parts will be sent from different vendors. This would have delayed our schedule in the prototyping stage. With all these advantages of choosing a microcontroller, we were capable of building the prototype without worrying about the insufficient clock frequency, mismatched Baud rates, or delay shipping time for the parts.

ATtiny85 is the Atmel microcontroller. It is also used on the Arduino UNO. This product allowed us to build our project on a smaller scale without using a full size Arduino board. There are three types of integrated circuit package Atmel offers: TQFP, PDIP, and MLF. Since one of our group members already has an Arduino UNO board, we chose the DIP package. The DIP package is more flexible and we can program the microcontroller on an Arduino UNO board and then attach it on our PCB design.

C. Mobile Application

The mobile application is the main interface and application that the user utilizes. Mobile applications can be used on most Android phone and tablet displays. This application connects to the analog circuit via Bluetooth in order to receive the analog signals and heart rate frequencies. The heart rate monitor is specifically designed to be used in professional, medical, or hospital settings and used by medical professionals.

The mobile application is a user friendly, simple GUI (Graphical User Interface) that allows users to monitor the heart rate frequency sent from the electrodes connected to a patient's body (a simulator in our case).

The design for the mobile application is comprised of a few activities for navigation. The main activity has an area where the heart rate frequency will be displayed in the top part of the screen, and the bottom portion is where the user can view information about the heart rate. In landscape mode, the heart rate frequency makes up most of the screen with a more detailed view. The next activity consists primarily of the process of connecting and pairing via Bluetooth.

The application has an interface that is intuitive for the user and easy to use. The functionality calculates and displays the heart rate sample and ECG signal sent from our microcontroller device. Bluetooth is our primary way of communicating data between our Android device and microcontroller. A Bluetooth connection to the microcontroller is started from our application on the Android device. The user should enable Bluetooth on their Android device, and then they will be prompted to connect to the Bluetooth module. If the user's Android device does not support Bluetooth, the ECG will not work on their device. The Bluetooth connection will go through authentication and establishment, and then the user will be directed to the next activity to display the ECG heart rate signal on their device. The Use Case activities are shown using UML.

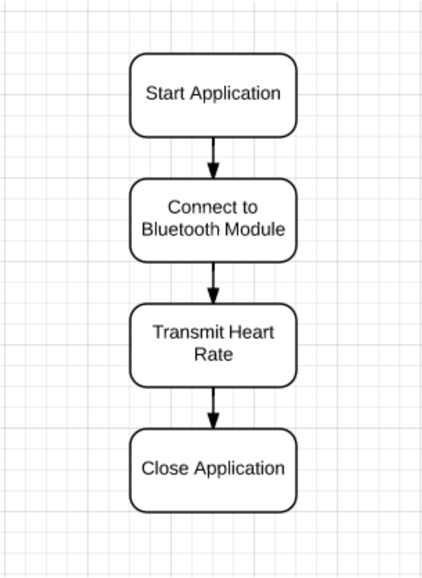


Fig. 7. Use Case Activities

Use case: Start Application - is representing when a user first launches the application, which is the very first activity.

TABLE II

START APPLICATION USE CASE

Name	Start Application
Actors	User
Pre-Condition	Android device with Bluetooth support
Description	Start up (First Activity) and Welcome Activity

TABLE III

CONNECT TO BLUETOOTH MODULE USE CASE

Name	Connect to Bluetooth Module
Actors	User
Pre-Condition	User starts Application
Description	Prompt the user for device discovery and connecting the devices

TABLE IV

TRANSMIT HEART RATE USE CASE

Name	Transmit Heart Rate
Actors	User
Pre-Condition	Bluetooth connection is established
Description	Begin reading data sent from Bluetooth module and display on Android device

TABLE V

DISCONNECT USE CASE

Name	Disconnect
Actors	User
Pre-Condition	Bluetooth connection is established
Description	User can disconnect from the Bluetooth module and exit the application

The Bluetooth module we decided to use for transmitting the ECG reading to the mobile device is the BTM-5 module. The BTM-5 module is a class 2 Bluetooth module that uses the BlueCore4-External chipset from the Bluetooth chipset supplier Cambridge Silicon Radio. This module supports both Master and Slave mode operations. It can be changed by the AT command configuration. Each Master/Slave BTM-5 pair automatically links with the default device address after power up.

After the link is established, the user can transmit and receive data via the UART interface with each other. BTM-5 module is also widely used on communication between the Arduino microcontroller and the mobile device. The Table VI is the specification of the BTM-5 module.

TABLE VI

BTM-5 SPECIFICATIONS

Parameters	**Specifications**
Operating Frequency Band	2.4 GHz-2.48 Ghz unlicensed ISM Band
Bluetooth Specification	V2.0+EDR
Output Power Class	-4 ~ 6 dBm adjustable, Class 2
Sensitivity	-80 dBm at 0.1% BER
Data Rate	Asynchronous: 2 Mbps (Max)
Operating Voltage	3.3 V
Host Interface	USB/UART
Audio Interface	PCM and Analog Interface
Flash Memory Size	8 Mbit
Operating Temperature	-20 ~ +55 ℃
Dimension	26.9mm(L) x 13mm(W) x 2.2mm(H)

Before we can do anything, the Android device and the microcontroller's Bluetooth module must be paired up. To accomplish this, we have to find the Android's Bluetooth settings and pick the microcontrollers Bluetooth module so we can pair them up. If the Bluetooth module is not appearing on the Android device, we need to verify that the Android device is connected to the module properly and that the microcontroller is getting power. Additionally we should unpair all other Bluetooth devices in the vicinity so we can make sure that our two devices will be able to correctly connect.

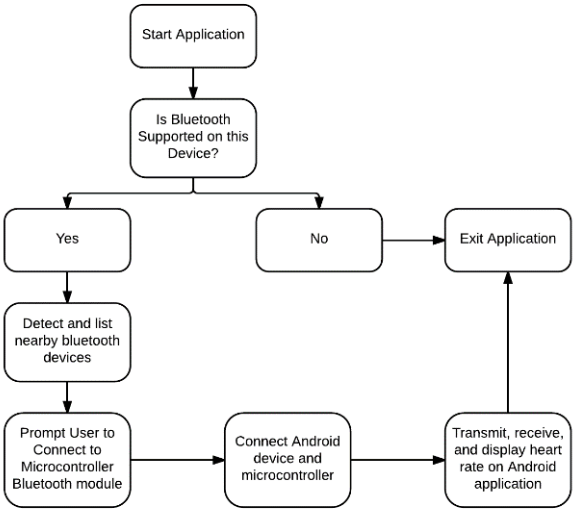


Fig. 8. Application Flow Chart

Fig. 8 shows a block diagram of the Bluetooth connection process between our Android device and microcontroller. First the user will start up the application, without much delay. The application will detect if the device supports Bluetooth. If it does, continue on and detect and list Bluetooth devices in the vicinity. If it does not support Bluetooth, it will close the application.

After detection of nearby Bluetooth devices and displaying them to the user, the user will be able to choose which Bluetooth device to connect to. Usually only our microcontroller Bluetooth module will be the only Bluetooth device to connect to in the area. Then it will go through the whole Bluetooth connection process and the two devices will be connected, being able to transmit and receive our ECG data that we will need to process and display on our application.

D. PCB Design

There are three custom printed circuit board designs used in this project. All PCB Designs were developed using CadSoft's Eagle design software. Each component on the PCB is used to accomplish a specific task of the project as required by the project specifications.

Without discussing each component of the PCB design in detail, Fig. 9, Fig. 10, and Fig. 11 will show the PCB layouts of the heart rate simulator, analog front end circuit, and microcontroller with Bluetooth Module. All components used in this project were through-hole components. The through-hole was used because we decided to test every part of system before mounting them on the boards.

First, the PCB of the ECG simulator is shown below.

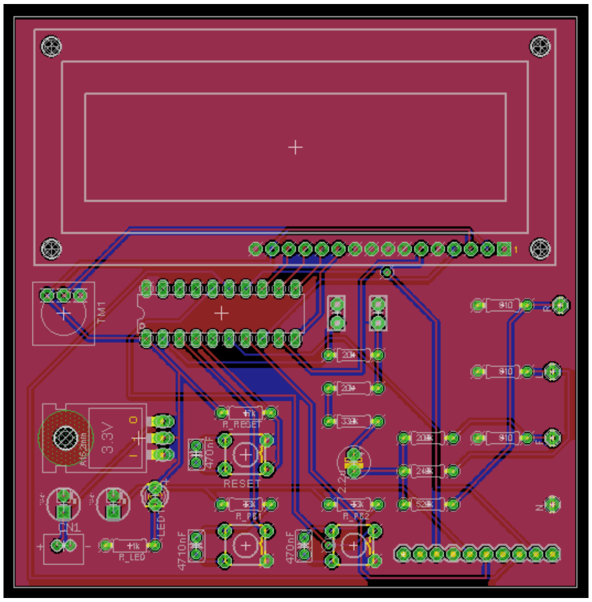


Fig. 9. Heart Rate Simulator PCB

Second, the PCB of analog front end is shown below.

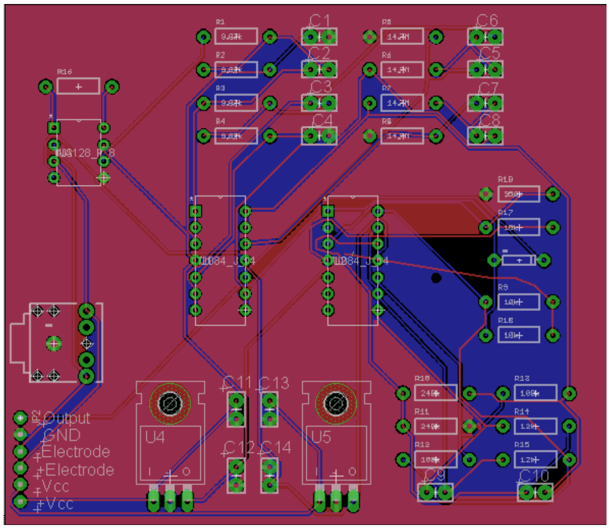


Fig. 10. Analog Front End PCB

Finally, the PCB of microcontroller with Bluetooth module is shown below. Notice this design is relative easy to read. The components in this PCB were only the Atmel Attiny85 microcontroller, the 5V voltage regulator, and an LED light to indicate the power. Also the Bluetooth Module we used in this project was already built with a breakout board. Therefore, it can be easily connected with some 4 pin male headers. This PCB was separated from the analog front circuit due to the digital and analog based circuit. The digital and analog circuit should not have a common ground. Otherwise, noise from the digital board will be introduced to the analog circuit.

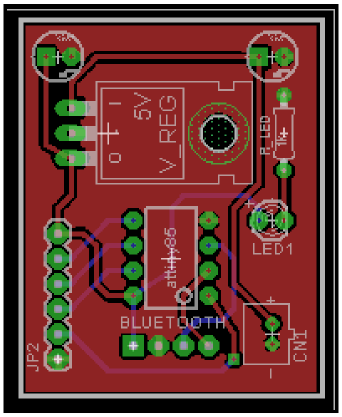
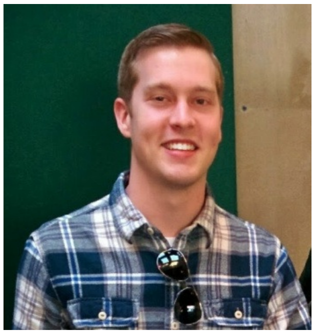


Fig. 11. Microcontroller with Bluetooth Module PCB

III. CONCLUSION

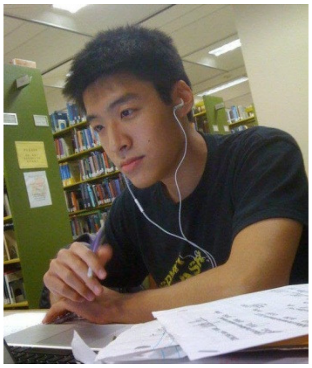
In conclusion, our project works as we expected. Although the functionality of the product is limited, we focused on getting the cheapest ECG monitor that not only doctors could obtain. Patients can also easily obtain such a product and use it with ease.

BIOGRAPHY



Jeffrey Frye is a senior majoring in Electrical Engineering at the University of Central Florida. He has interned at Lockheed Martin Missiles and Fire Control the past two summers, working mainly in FPGA design, circuit simulation and analysis, and MATLAB algorithms. After graduation, Jeffrey will be continuing his internship and pursuing his Master's degree in Electrical Engineering at UCF.

Jonathan Gibson is a senior electrical engineer at the University of Central Florida and is currently working at Earthrise Space Foundation as a systems engineer. He has previously interned at Lockheed Martin Missiles and Fire Control and worked in multiple engineering disciplines while there. In the future he plans to pursue a career in patent law with a primary focus in biomedical devices.



Michael Sun is currently a senior at the University of Central Florida. He plans to graduate and receive a Bachelor's of Science degree in Computer Engineering in May of 2015. In his last year, Michael has been an Automation Testing intern in the Quality Assurance department for an invoicing company called Viewpost. He was offered and has accepted a full time position after graduation in the same role. His interests are in Android application development and game development.

Cheng-Chieh Wang is currently a student at the University of Central Florida. He will be graduating in May 2015 with a Bachelor's degree in Electrical Engineering. His interests include the fields of electronics and digital signal processing. Following graduation, he intends to continue his education, pursing a Master's degree in Electrical Engineering.

REFERENCES

[1] Behutiye, Woubshet. "Android ECG Application Development." (n.d.): n. pag. Spring 2012. Web. 15 Nov. 2014.

[2] "Biopotential Amplifiers." SJSU EE Faculty Website. N.p., n.d. Web. 02 Dec. 2014. <http://www.engr.sjsu.edu/mkeralapura/>.

[3] Cache. "Interface MSP430 Launchpad with LCD Module (LCM) in 4 Bit Mode." Web log post. Co-Random Thoughts. N.p., n.d. Web. 1 Dec. 2014. <http://cacheattack.blogspot.com/2011/06/quick-overview-on-interfacing-msp430.html>.

[4] "Getting the Most out of Your Instrumentation Amplifier Design." (n.d.): n. pag. Amplifiers: Op Amps. Web. 1 Dec. 2014. <http://www.ti.com/lit/an/slyt226/slyt226.pdf>.

[5] MICHALEK, Paul J. "AN AUTHENTIC ECG SIMULATOR." Thesis. University of Central Florida, 2006. Web. 1 Dec. 2014. <http://etd.fcla.edu/CF/CFE0001214/Michalek_Paul_J_200608_MS.pdf>.