

March 13th, 2010

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
8888 University Drive, Burnaby, BC
VSA 1S6

RE: ENSC 440 Capstone Project Design Specification for the HeartMon™, a cardiovascular diagnostic device

Dear Dr. Rawicz:

Please find attached the Capstone Project Design Specification for the product HeartMon™, presented by Biomedical Engineering Solutions. We are designing and implementing a heart monitoring system that will be an improvement over the Holter monitor by including diagnostic capabilities and being more portable and accessible. The HeartMon™ is a preventative device, which will keep patients aware of their own health, save doctors time in making diagnoses, and save money in the healthcare industry.

This document provides the how and why to meet the requirements in our functional specification. The design specifications in this document apply to the proof-of-concept model only, although design improvements for future iterations will be discussed.

Our team is versatile and consists of five innovative and motivated individuals: Amir Kamyabnejad, Bobby Luk, Cheng Zhang, Eric Boyer, and Yash Trivedi. If there are any questions or concerns regarding our proposal, feel free to contact me by phone at 604-617-1478 or by e-mail at aka39@sfu.ca.

Sincerely,



Amir Kamyabnejad
Chief Executive Officer
Biomedical Engineering Solutions

Enclosure: *Proposal for a Heart Monitor System*

Design Specification for the HeartMon™

Project Team:	Amir Kamyabnejad Bobby Luk Chen Zhang Eric Boyer Yash Trivedi
Contact Person:	Amir Kamyabnejad aka39@sfu.ca
Submitted to:	Dr. Andrew Rawicz – ENSC 440 Michael Sjoerdsma – ENSC 305 School of Engineering Science Simon Fraser University
Issued Date:	March 13, 2011
Revision:	1.5

EXECUTIVE SUMMARY

This design specification for the HeartMon™ heart monitoring system provides a detailed description of the design and implementation of our prototype. This prototype will be used for proof-of-concept purposes and hence will meet the requirements marked as levels I and II in the Functional Specification BES documents.

This Design Specification document is divided into three main parts: Hardware, Software, and System. Each section will be discussed in detail.

The hardware section includes design specifications for the ECG electrodes and cables, patient protection, instrumentation amplifier, low-pass filter, high-pass filter, notch filter, and final gain stage. The derivations of the specific values for the filters are shown, accompanied with schematic diagrams. The filters are designed to capture the ECG signal bandwidth of 0.05Hz-100Hz and reject any noise outside this bandwidth. The major noise contributors are noise from muscle movement around 150Hz, RF signals in the air, EMI from surrounding electronics, and power line interference at 60Hz.

The system section includes the overall system design: mechanical design, top level design, microcontroller and Bluetooth. The task of the microcontroller is to receive ECG signals from the ECG circuit, perform Analog to Digital Conversion (ADC), do high order digital filtering, and send the output to the phone via Bluetooth.

The software section includes mainly the phone application parts: warning system, data logging system, and method of uploading to a server. The warning system will indicate if some heart-rate readings fall into a dangerous range. If a heart-rate reading in critical states persists, healthcare professionals are notified. The data logging system uses internal memory or SD memory card to log in the data. Finally, an uploading scheme is devised to upload the results onto a Google document that would be accessible to the doctor and health care professionals remotely.

Lastly, a test plan is devised to ensure the targets of HeartMon™ are met. This plan includes testing the individual parts of project separately first. That is, to see there's a signal coming from electrodes, the signal is filtered to the desired bandwidth, there's a bi-directional communication between microcontroller and cell phone, correct analysis of results on android software via giving test values for threshold, proper data logging, and successful uploading of the result file on Google server. False errors also are avoided by means of keeping track of the last 10 consecutive alarms and are tested against faulty measurements. Thereafter, the system will be integrated and tested as a whole.

The delivery date for the HeartMon™ prototype remains as mentioned in the Functional Specification, April 10, 2011.

Table of Contents

EXECUTIVE SUMMARY	ii
List of Figures.....	v
Acronyms.....	1
Glossary	1
1. Introduction.....	2
1.1. Scope.....	2
1.2. Audience	2
2. Overall System Design	3
2.1. Top-level System Design	3
2.2. Mechanical System Design.....	4
3. ECG Circuit and Electrodes	5
3.1. Overview	5
3.2. Patient protection, low-pass filter at 100Hz, instrumentation amplifier, and leg driver stage.....	6
3.3. Low-pass filter at 100Hz.....	9
3.4. High-pass filter at 0.5Hz.....	9
3.5. Notch filter at 60Hz.....	10
3.6. Final gain stage.....	11
3.7. ECG circuit power supply	11
4. Microcontroller and Bluetooth transmitter	12
4.1. Overview	12
4.2. Analog to Digital Conversion.....	13
4.3. Digital Filter	14
4.4. Bluetooth Transfer	14

5. Cell Phone Application	15
5.1. Overview	15
5.2. Platform, Programming Language, and Development Tools	15
5.3. User Interface	16
5.4. Basic Application Algorithm	16
5.5. Phone Warning Notification	18
5.6. Data Logging System	18
5.7. Data Uploading Scheme	19
5.8. Class Implementation	19
6. System Test Plan	20
6.1. Sensors	21
6.2. Hardware	21
6.3. Software	21
6.4. Integrated System	21
6.5. False Alarms	22
7. Conclusion	22
8. References	23

List of Figures

Figure 1. Top level architecture design	4
Figure 2: Overall mechanical design for the HeartMon™	5
Figure 3: Flow-chart of the HeartMon™'s ECG circuit	6
Figure 4: Input filter, instrumentation amplifier, and leg driver stage	6
Figure 5: 100Hz low-pass filter stage	9
Figure 6: 0.5Hz high-pass filter stage	10
Figure 7: Notch filter at 60Hz	10
Figure 8: Final gain stage	11
Figure 9: ECG voltage regulators	12
Figure 10. Flow-chart of microcontroller and Bluetooth	13
Figure 11: Cell phone application flow chart	15
Figure 12: User Interface	16
Figure 13. Basic flowchart of the phone application	17
Figure 14: User Authorization Process	19
Figure 15: Diagram for the structure of the HeartMon Application Classes	20

Acronyms

ADC	Analog to Digital Conversion
API	Application Programming Interface
CM	Common Mode
CMRR	Common-Mode Rejection Ratio
ECG	Electrocardiogram
EMI	Electromagnetic Interference
GPS	Global Positioning System
HPF	High-Pass Filter
IA	Instrumentation Amplifier
IDE	Integrated Development Environment
LPF	Low-Pass Filter
MCU	MicroController Unit
OA	Operational Amplifier
RF	Radio Frequency
SDK	Software Development Kit
USB	Universal Serial Bus

Glossary

Arduino	The microcontroller used in this project. The specific model used is the Arduino BT (the Bluetooth version).
Bluetooth	A proprietary open wireless technology standard for exchanging data over short distances (using short wavelength radio transmissions) from fixed and mobile devices, creating personal area networks (PANs) with high levels of security. [i]
Health Professional	An individual or an institution that provides preventive, curative, promotional or rehabilitative health care services in a systematic way to individuals, families or communities. [ii]

1. Introduction

Monitoring a patient's heart activity is very important in the healthcare industry. Heart monitoring is often performed after a patient has a heart attack or stroke, after having heart surgery, and after being prescribed medications that could affect the heart. However, most heart monitors currently available, such as the Holter monitor, do not offer adequate functionality for many applications. To solve this problem, Biomedical Engineering Solutions is developing the HeartMon™. This will be a pocket-size device that will measure the patient's heartbeat by means of three electrocardiogram (ECG) electrodes. The ECG signal will be digitized by means of a microcontroller unit (MCU), which will then send the signal to the user's cell phone by means of Bluetooth. The accelerometer and GPS embedded in the cell phone is used to track user's activity and location. The cell phone will contain an application that processes the ECG and accelerometer data to determine if the user's current heart activity is acceptable for their current level of activity. If it isn't then the user's doctor will be notified immediately, at which point emergency personnel will come. This document outlines the design specifications of this device.

1.1. Scope

BES will use this document and the HeartMon™ Functional Specification as guides during the development of the prototype HeartMon™. These design requirements will focus on the details of the level I and II functional requirements, as these are all that are necessary to build our prototype. After the prototype is complete and we are ready to refine our product into a retail device, we will focus on the level III functional requirements. Details of our design will be discussed at length and justification will be given for our particular design choices.

1.2. Audience

The intended audience of this document is the engineers of BES. They will use it as a reference as they progress in their work on the prototype. This document should also be readable for technically-knowledgeable investors interested in the HeartMon™.

2. Overall System Design

The overall system can be divided into the following components:

- 1) Top-level system design
- 2) Mechanical Design
- 3) ECG circuit and sensors
- 4) Microcontroller and Bluetooth transmitter
- 5) Cell phone application

2.1. Top-level System Design

The top level design divides the project into two stages, hardware and software. From a top level view, the ECG electrodes act as sensors when worn by the patient. The signal is then put through analog and digital filters to remove any unwanted interference before being sent to the phone. The phone then processes the ECG information along with other measurements such as the accelerometer and GPS. Diagnostics and feedback are then provided depending on the conditions. The hardware stage involves obtaining a signal from the electrodes and then conditioning it with the aid of analog filters before being fed to the microcontroller. The software stage involves digital filtering by the microcontroller and then diagnostic algorithms within the phone application. A top level system architecture design is illustrated in Figure 1, below.

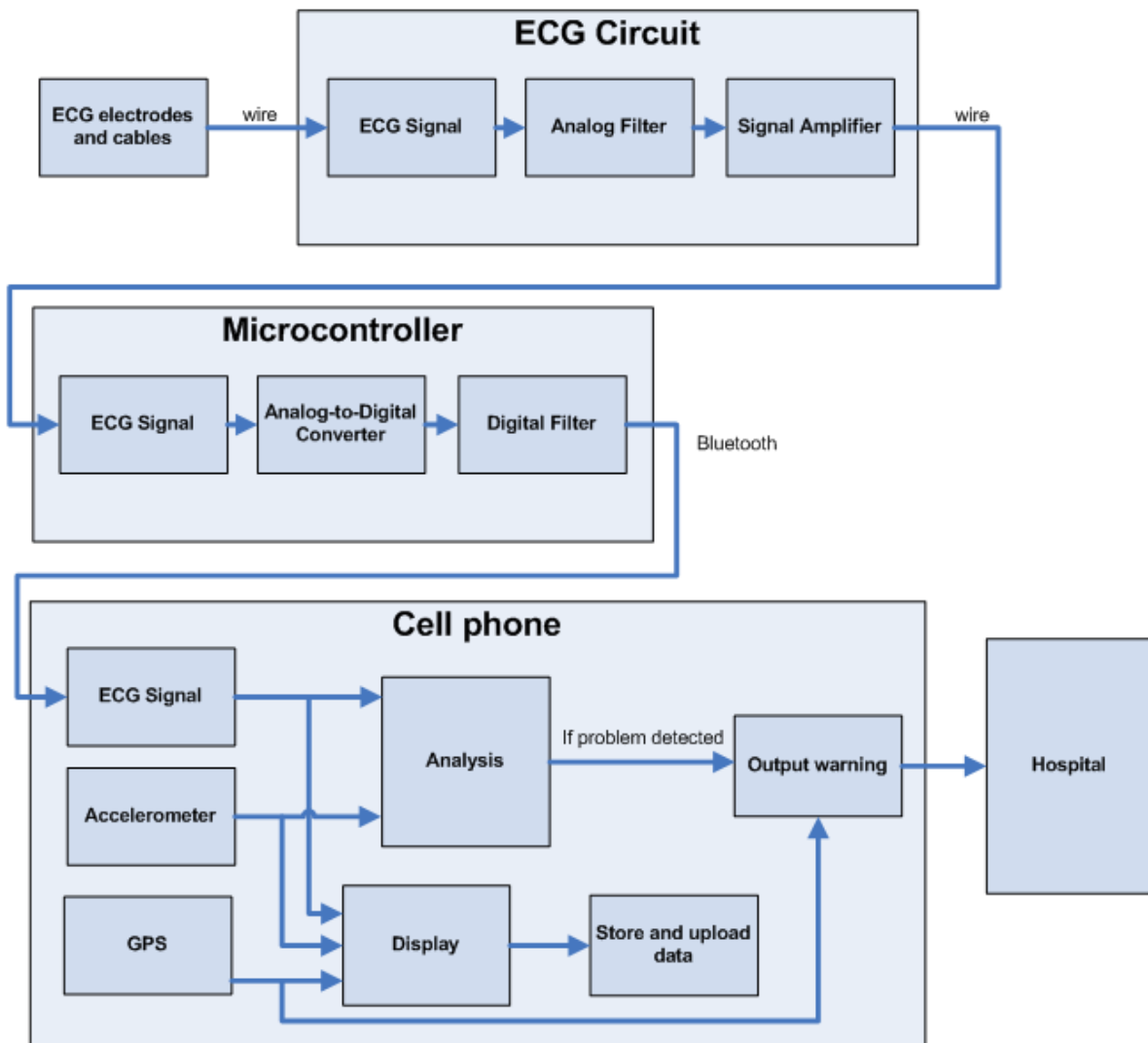


Figure 1. Top level architecture design

2.2. Mechanical System Design

Mechanically, the design consists of the HeartMon™ hardware as per the functional specification requirement [R1-II]. The HeartMon™ hardware pack includes a PCB for the analog filter circuit, the microcontroller with its Bluetooth module, and a battery power source. The general mechanical design of the HeartMon™ is shown in figure 2.

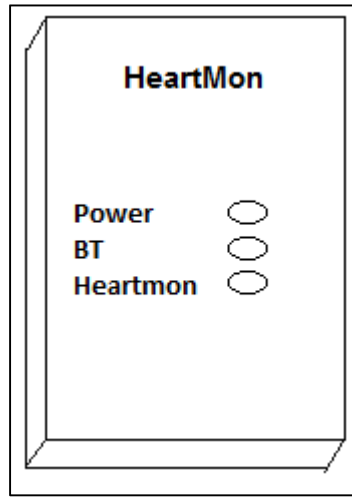


Figure 2: Overall mechanical design for the HeartMon™

3. ECG Circuit and Electrodes

3.1. Overview

The electrical signals produced by the heart during a heartbeat are approximately 0.5 - 5mV in amplitude [1], occur anywhere from 60 to 220 times per second [2], and have most of their frequency content in the range of 0.05Hz to 100Hz [2]. Also, the body can have a varying DC potential up to $\pm 300\text{mV}$, there can be fairly significant amounts of noise above 150Hz from muscle movement, and signals below 0.5Hz are produced from patient movement and slow shifts in the DC potential on the body. External electrical noise can also occur from RF signals in the air, EMI from surrounding electronics, and power line interference at 60Hz (50Hz in some countries). Because of how weak the actual ECG signal is, all of these sources of noise interfere quite strongly with it.

To obtain an ECG signal, electrodes are attached to the body at specific locations depending on the number of electrodes used, and the electrodes are connected to an ECG circuit via shielded cables. In the case of the HeartMon™, which uses three electrodes, one electrode is placed near the left shoulder, one is placed below and to the right of the heart, and one is placed on the right leg. The first two electrodes measure the actual ECG signal, while the third is used to cancel out common-mode noise in the body, discussed later.

The structure of the HeartMon™'s ECG circuit is shown in figure 3. Each portion of the circuit will be discussed in the next sections. It should be noted that the choice of ordering for the filters is arbitrary and that they can be implemented in any order without adverse effects. However, the final gain stage must come after the filters because the filters remove noise and the DC component, both of which could saturate the OA in the final gain stage. The digital filters are discussed in section 4 of this document because they are implemented in the Arduino microcontroller. Originally, BES was planning to use only digital filters because of their easier implementation and flexibility, but the analog filters used are essential to ensure that the signal sent to the Arduino is of high enough quality to be processed digitally.

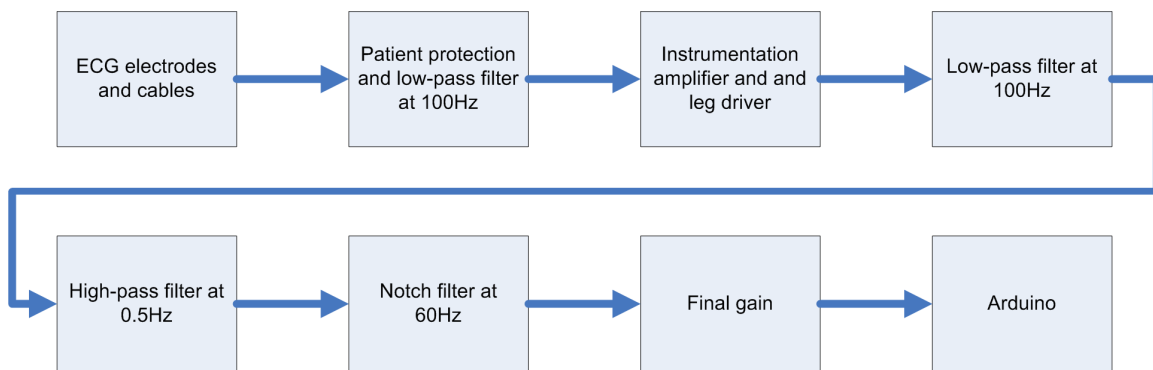


Figure 3: Flow-chart of the HeartMon™'s ECG circuit

3.2. Patient protection, low-pass filter at 100Hz, instrumentation amplifier, and leg driver stage

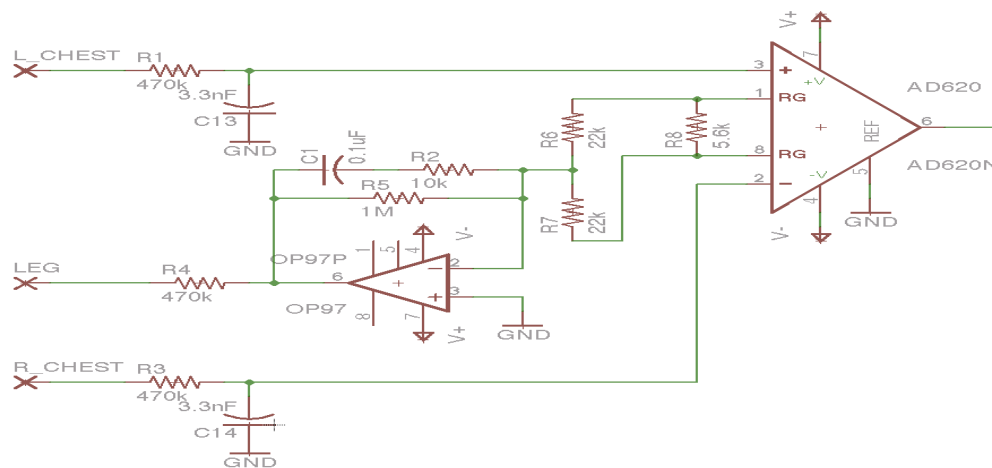


Figure 4: Input filter, instrumentation amplifier, and leg driver stage

Figure 4 shows the schematic of the first stage of the ECG circuit. The ECG signal from the chest electrodes first passes through 100Hz 1st-order low-pass filters (LPF) formed by R1 and C13 for the left electrode, and R3 and C14 for the right electrode. These filters attenuate RF interference and EMI, as well as muscle noise, and must be implemented before the instrumentation amplifier (IA) because otherwise the RF and EMI may be amplified to a level that saturates the IA. Muscle noise occurs above 150Hz, RF and EMI occur at much higher frequencies, and the ECG signal has frequency content up to only 100Hz. So, a -3dB frequency of 100Hz was chosen to maximize noise attenuation without affecting the signal. Using standard components, the -3dB frequency is

$$\begin{aligned}
 f &= \frac{1}{2\pi RC} \\
 &= \frac{1}{2\pi \cdot 470k\Omega \cdot 3.3nF} \\
 &= 103Hz
 \end{aligned}
 \tag{1}$$

which is close enough for our purposes.

Two resistors also serve the purpose of patient protection. The electrodes form a low-resistance electrical path through the patient's heart that can be as low as a few hundred ohms depending on the electrodes used and the skin's condition. So, even if a small voltage potential is produced across the electrode cables then the patient could be at risk. Therefore, high-value resistors must be placed in series with the ECG cables at the point at which they enter the ECG circuit to minimize the amount of current that can flow through the body.

The American Heart Association recommends that current through the body be limited to 10uA under no-fault and single-fault conditions [3]. In our circuit there will be no current through the electrodes under no-faults conditions, and the worst-case single-fault condition would be a short from either the +5V or -5V supply to an electrode. Assuming the worst case of 0Ω resistance in the body between the two chest electrodes, to ensure that less than 10uA flows through the body, there needs to be a total resistance of

$$\begin{aligned}
 R &= \frac{5}{10^{-5}} \\
 &= 500k\Omega
 \end{aligned}
 \tag{2}$$

from one chest electrode to the other. This would require 250kΩ resistors in series with the electrode cables. These resistors would have to be placed as close as possible to the attachment point of the ECG cables on the ECG circuit, or even be integrated into the ECG cables, to ensure that any circuit fault would have to pass through them.

To maximize patient safety we have decided to protect against double-fault conditions by using 500kΩ resistors on each electrode cable. This will ensure that the patient will be safe under the worst possible fault condition of the ECG circuit: both power supply rails being applied to each chest electrode. The use of 500kΩ resistors instead of 250kΩ resistors will have almost no effect on circuit performance thanks to the high input impedance of the instrumentation amplifier. For our actual circuit, we are using the closest standard value of 470kΩ.

Because the HeartMon™ is battery-powered and has no link to household power, and because ±5V is the absolute maximum voltage possible in the circuit (the switching power supplies used to power the ECG circuit are incapable of producing voltages higher than this because they do not use inductors), no further patient protection is required. If the HeartMon™ could be plugged in then further protection would be required, for example by means of opto-couplers. However, this is not necessary for the HeartMon™.

Next, the ECG signal is amplified by means of the instrumentation amplifier (IA). An IA is used instead of an operational amplifier (OA) because IAs have superior performance in almost all regards, such as low DC offset, low noise, high common-mode rejection ratio (CMRR), excellent long-term accuracy, and, most important, high input impedance. The high input impedance means that the high source impedance from the protection circuitry has little effect on the IA. For our circuit we chose the AD620, an IA commonly used for ECG circuits. This IA was also chosen because of its low supply current, which is a maximum of 1.3mA.

The gain of the IA is [1]

$$A = 1 + 49.4k / R_G + (49.4k / 2) / 22k \quad (3)$$

where R_G is the resistance between pins 1 and 8. The ECG signal has an amplitude of 0.5 – 5mV, and this is superimposed on a DC component that can be as high as ±300mV. Therefore, the IA must have a gain that is small enough to not saturate itself with a 300mV input. The IA is powered by ±5V, and the output swing of the AD620 is $(-V_S + 1.1)$ to $(+V_S - 1.2)$, so the maximum outputs are -3.9V and +3.8V. Therefore, the maximum gain that can be used is $3.8/0.3 = 12.7$ V/V. For our circuit we set R_G to 5.6kΩ, which produces a gain of 10.9. This will only bring the ECG amplitude up to 5 – 50mV, which is still very low, but further amplification is provided later.

The OP97 operational amplifier (OA) in this stage is used to cancel out common-mode (CM) noise in the patient's body. It applies an inverted version of the CM signal from the chest electrodes to the patient's right leg. This helps to reduce the CM content in the signal. This OA was chosen because of its low power consumption.

3.3. Low-pass filter at 100Hz

To further reduce muscle noise, RF, and EMI, the output of the IA is applied to a 2nd-order unity-gain Sallen-Key Butterworth LPF at 100Hz, shown in figure 5.

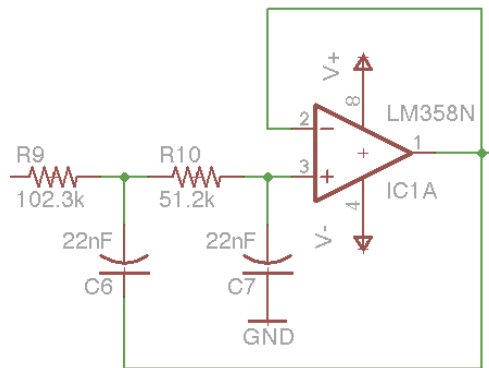


Figure 5: 100Hz low-pass filter stage

It would be optimal to implement a 6th- or 8th-order filter here because of how strong the muscle noise can be and because of that fact that it is less than an octave above the ECG signal's bandwidth, but due to space and power concerns, and components tolerances, this is not feasible. Instead, we simply use this 2nd-order filter to remove a reasonable amount of noise, and then use a higher-order digital LPF on the Arduino to further reduce the noise (discussed in section 4). The Sallen-Key configuration was used because of its low component count and the fact that it only uses one OA to implement a 2nd-order filter, which reduces power requirements.

This filter was designed using reference 4, an online active filter design program that uses filter design equations that can be found in almost any basic electronics textbook. Thus, the derivation of the component values is not given. Exact theoretical component values are given in Figure 6 and all figures that follow; actual component values will be chosen to be as close as possible to these values.

3.4. High-pass filter at 0.5Hz

To attenuate ultra-low frequency noise produced by body movement and shifts in the body's DC component, and to completely block the DC component, a 2nd-order unity-gain Sallen-Key Butterworth high-pass filter (HPF) at 0.5Hz is implemented after the LPF. Its schematic is shown in figure 6.

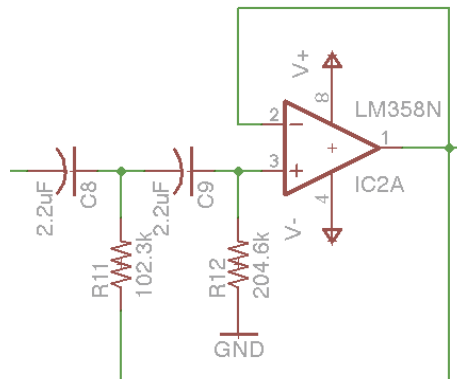


Figure 6: 0.5Hz high-pass filter stage

Although the bandwidth of the ECG signal is nominally 0.05Hz – 100Hz, we chose to use a HPF at 0.5Hz rather than 0.05Hz due to implementation limitations. A HPF at 0.05Hz would require excessively large capacitors and resistors, which would cause space requirement problems and noise problems, respectively. The loss in signal quality from this choice will be minimal.

Again, it would be optimal to implement a 6th- or 8th-order filter for this stage, but further filtering will be provided digitally. This filter simply provides the minimal filtering required for the digital circuitry.

3.5. Notch filter at 60Hz

To attenuate the 60Hz noise produced by household wiring, a Sallen-Key notch filter at 60Hz is implemented after the HPF. Figure 7 shows its schematic.

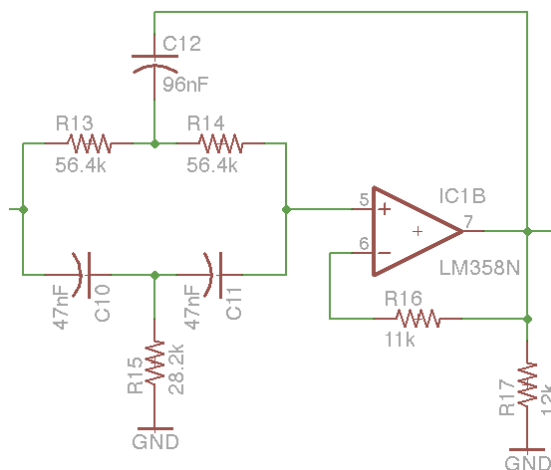


Figure 7: Notch filter at 60Hz

This filter has a bandwidth of 10Hz, which will allow it to attenuate noise that isn't exactly at 60Hz.

3.6. Final gain stage

The schematic for the final gain stage is shown in figure 8.

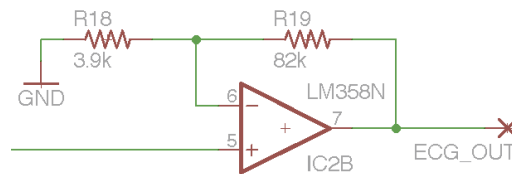


Figure 8: Final gain stage

This is simply a non-inverting amplifier with a gain of $+21V/V$. This amplifies the ECG signal so that the heartbeat from an average 22 year-old male at rest has an amplitude of 0.8V. This allows for plenty of headroom in the circuit for stronger heartbeats from individuals engaging in strenuous activity (the saturation limit of the op-amp is 3.8V). This gain can also be changed if needed while testing the prototype.

This stage provides the majority of the amplification for the circuit, and, as mentioned before, must be done after the filters to ensure that no noise is amplified to saturation limits. This stage uses non-inverting amplification to ensure that the output ECG signal has a positive voltage when the electrodes are connected properly, which is necessary because the inputs of the Arduino only accept voltages from 0 to +5V.

3.7. ECG circuit power supply

The IA and OAs in the ECG circuit must be powered by dual-polarity power supplies. To ensure that enough dynamic range is provided, this power supply should be at least $\pm 5V$, and this is the voltage we chose to use, partly due to the wide availability of +5V and -5V regulators. From the component datasheets it can be seen that the active components in the ECG circuit will require a maximum of 4.3mA, so the voltage regulators will need to be able to handle this.

To make the HeartMon™ user-friendly, the whole system will run off of two easily replaceable AA batteries. To ensure maximum compatibility, all major battery chemistries will be supported, meaning that supply voltages from the two AA's will range from 2.4V (NiCd and NiMH) to 3.2V (Nickel-Zinc), and thus must be supported.

Therefore, the ECG circuit must include a step-up voltage regulator with an input range of at least 2.4V – 3.2V and a fixed output of +5V, as well as a voltage inverter providing a fixed output of -5V. Both of

these regulators must be able to supply at least 4.3mA and should be as efficient as possible to minimize wasted power. To meet these requirements, we chose the MAX619 fixed +5V regulator and the LTC1046 voltage inverter. The MAX619 is powered by the two AA batteries to provide +5V, and the voltage inverter is connected to this +5V output to produce -5V. The schematic for these voltage regulators is shown in figure 9.

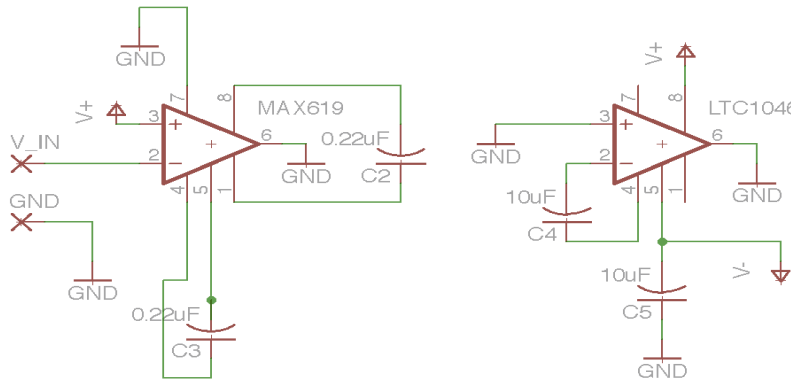


Figure 9: ECG voltage regulators

4. Microcontroller and Bluetooth transmitter

4.1. Overview

A microcontroller is used to receive ECG signal from ECG circuit. The microcontroller digitizes the received signal, performs an analog-to-digital conversion (ADC), and transmits the result to the Cell phone via a built-in Bluetooth (BT) module. The Microcontroller and BT flow chart is illustrated in Figure 10, below.

The Arduino BT V06 was selected for this project because of having our required functionalities: built-in Bluetooth, high scalability, easy to use, low cost, readily available, compatible with Android OS, and well documented.

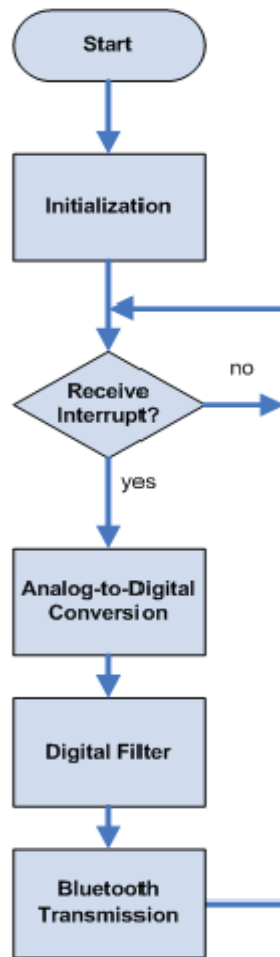


Figure 10. Flow-chart of microcontroller and Bluetooth

4.2. Analog to Digital Conversion

The microcontroller board, Arduino, is used to digitize the ECG signal obtained from the ECG circuit. This is essential as communication via Bluetooth and digital filtering can only happen with digitized data. Arduino takes an analog input on its PIN 4 analog input and converts it to a digital number between 0 and 255.

The specific microcontroller chip used on Arduino is ATmega168. This microcontroller has the capability of 10-bit analog to digital conversion (ADC) which satisfies our requirements.

4.3. Digital Filter

After digitizing the ECG signal, microcontroller performs digital filtering. Most of filtering is done via analog filtering mentioned in section 3. As mentioned previously, analog filtering the strong noise coming from muscle movement is not feasible. This muscle noise is very strong around 150Hz which is less than an octave above the ECG signal's bandwidth (0.05Hz – 100Hz).

Thus, a high-order Butterworth low-pass filter (LPF) is implemented within the microcontroller to facilitate the muscle noise removal. The order of this digital filter is determined according to equation 4 below. Instead, we simply

$$G = \frac{1}{\sqrt{1+\omega^{2n}}} \quad (4)$$

Where,

G = desired frequency response gain

ω = angular frequency in radians per seconds

n = number of reactive elements

The order of this digital filter will be set at 4 or 5 depending on the quality of signal observed in practice. The desired frequency response will be set at 100 Hz because this is the upper limit of the ECG signal's desired bandwidth.

The implementation will be performed in C++ since the Arduino is compatible with operation sketches written in C++.

4.4. Bluetooth Transfer

Once the signal is conditioned by digital LPF, it will be transmitted to the phone via the Bluetooth (BT) module on the microcontroller. The BT module used on Arduino BT V06 is BlueGigaWT11

The sampling rate for the BT transfer is set to milliseconds. This sampling rate was chosen to ensure that all relevant information in the ECG signal is preserved, and also to ensure that the data is not sent at an excessive rate, which would waste power and processing time.

The baud rate for the Bluetooth connection is set at 115,200 since it was observed to deliver the best results.

5. Cell Phone Application

5.1. Overview

The cell phone application will act as the user interface of the HeartMon™ product. It will receive heart rate information from the microcontroller wirelessly using Bluetooth to forego the need for a cable to connect the phone and the microcontroller. The cell phone application will be able to establish two-way Bluetooth communication with the microcontroller by using the open source “Amarino” library [5].

Along with the motion data taken by the accelerometer on the cell phone, the application will analyze and store the data, and issue warnings when necessary. The phone’s accelerometer was chosen rather than an accelerometer on the Arduino, as was originally going to be the case, due to space requirements. Figure 11 below illustrates the role of the application.

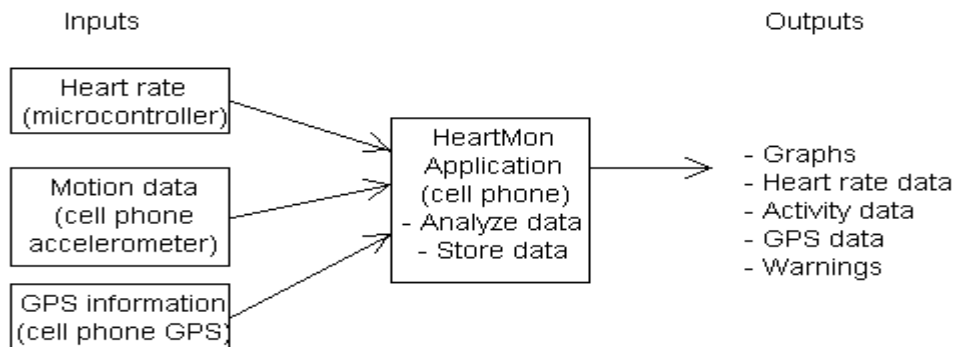


Figure 11: Cell phone application flow chart

5.2. Platform, Programming Language, and Development Tools

For the prototype, the platform of choice is the Android because it is open source and applications can be transferred to an Android phone via USB cable or internet download. This is in contrast to Apple’s closed iOS platform for the iPhone, which requires the process of registering a test device before an application can be tested on an iPhone. The BlackBerry platform supports applications written in Java and hence porting the application to the BlackBerry smartphone will require a minimal amount of effort. However, BlackBerry currently has only one device model (BlackBerry Torch) supporting both, accelerometer and GPS functionalities. The Android OS being freely available on a wide range of smartphones hence became the obvious choice.

Applications on the Android platform are developed using the Java programming language [7] so that is the language used. The Android SDK also provides many tools and APIs used for high-level programming. The IDE used to develop the application will be Eclipse SDK version 3.6.1 Helios because an Android plug-in for Eclipse has been provided by Android.

5.3. User Interface

The proposed user interface for the HeartMon™ is shown in Figure 12. The objective of the user interface is to be streamlined and show only necessary information.

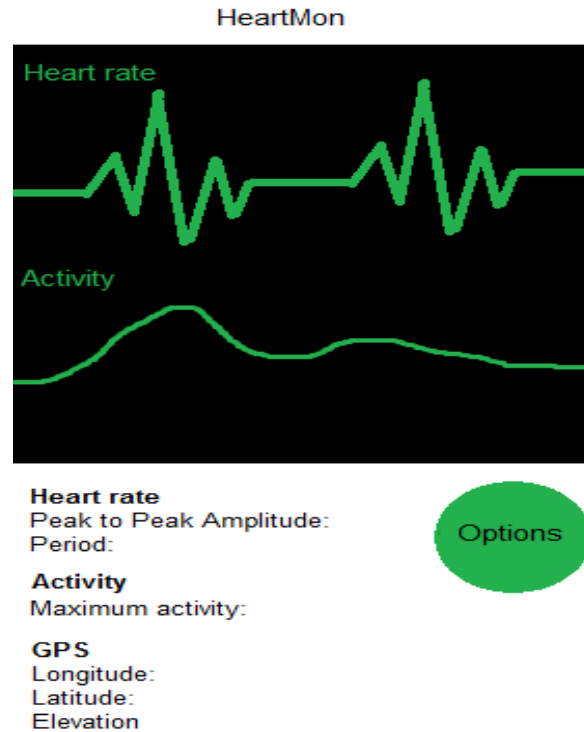


Figure 12: User Interface

When the application detects an anomaly it will issue a warning by playing a sound and displaying a pop-up in the middle of the user interface displaying the warning message. This pop-up will even appear over top of other applications.

5.4 Basic Application Algorithm

When the Bluetooth connection is established between the cell phone and the microcontroller, the application is ready to run. A flowchart of the basic program algorithm is shown in the Figure 13, below.

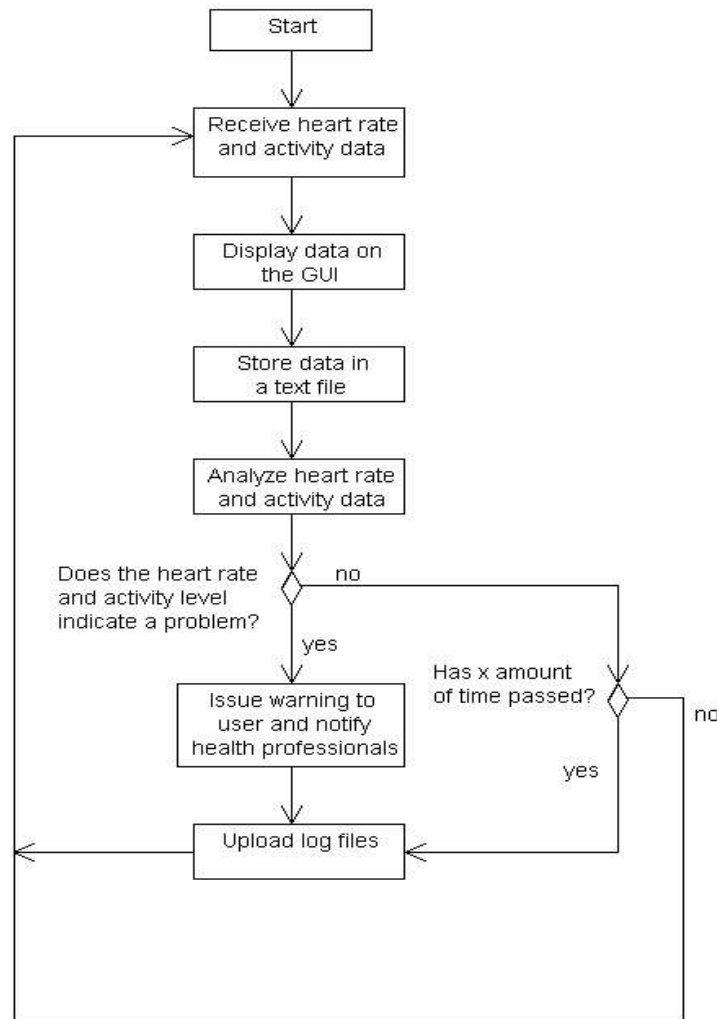


Figure 13. Basic flowchart of the phone application

The analysis algorithm looks at both the heart rate and activity data. The activity data is used to increase the accuracy of the analysis. For example, if the heart rate is elevated and the activity level is high, then the user’s heart is probably fine. On the other hand, if the heart rate is elevated and the activity level is low then it could be an indication of a heart problem.

The activity is detected by the accelerometer on the cell phone and the following equation 5 is used to determine the net activity

$$Net\ Activity = A_x + A_y + A_z \quad (5)$$

Where

A_x = the motion in the x-axis (the phone screen's horizontal axis)

A_y = the motion in the y-axis (the phone screen's vertical axis)

A_z = the motion in the z-axis (the direction toward the sky when the device is lying on its back on a table)

After an “x” amount of time has passed, the log files will be uploaded even if there is no emergency. For the prototype, “x” is set to be 1 day by default but in future iterations of the application the user will be able to choose how often the log files are uploaded.

5.5. Phone Warning Notification

If an irregularity is detected in the data collected from the Bluetooth based on the data analysis algorithm, a warning message will be displayed on the GUI to inform the user what sort of issue is occurring according to the analysis done by the application. The warning message is displayed on the top of screen for a duration of 1.5 minutes and it won't be overlaid or canceled by any other interface or interrupts so that the user won't miss the warning if an emergency occurs. To ensure that the user is able to notice the warning in a noisy public environment, the phone will vibrate and play a warning sound at full-volume, even if the phone is set to mute. The vibration lasts for 4 seconds and the user can customize the sound alarm by changing different sound effects in MP3 format.

Health professionals will be notified by a text message or phone call when the warning occurs.

5.6. Data Logging System

For the purposes of diagnostics, the application has the feature of logging the real-time data in the phone's built-in storage or an SD card. The data logging system is activated when the application is launched and keeps working in the background while the application is running. A line in the log file is made for each heartbeat, so the logging rate is approximately 1 – 2 times per second. Each stored log file contains a maximum of 100,000 lines of collected ECG readings, accelerometer readings, and corresponding timestamps in a spreadsheet format. The reason of limiting 100,000 lines of data in the log file is for the convenience of uploading the file to the Internet. Once 100,000 lines is reached, a new file is created. Each log file's name is its created time in the format of year-date-hour-minute-second so that it will be easy for the user or doctor to track any useful information. The data logging system will be deactivated once the application is terminated. Log files that are older than a week will be deleted to

reduce the load on the cell phone's storage memory. They are kept on the phone for a week as opposed to being deleted after upload for redundancy.

5.7. Data Uploading Scheme

In order to upload the data logs to online storage so that doctors and nurses can remotely access it, the application uses the Google Data Java client library that can interact with Google documents to upload the log files to the Google Documents server, from which the doctors and nurses can log in and access the files. To authorize the application to upload files to Google documents, first-time users of the application must authorize access to their Google data. Typically, the users see some text and a link or button directing them to authenticate (sign in) using their Google Account credentials. After they authenticate, users are directed to a URL where they can use the application to retrieve data from their Google data feed. This process is shown in Figure 14.

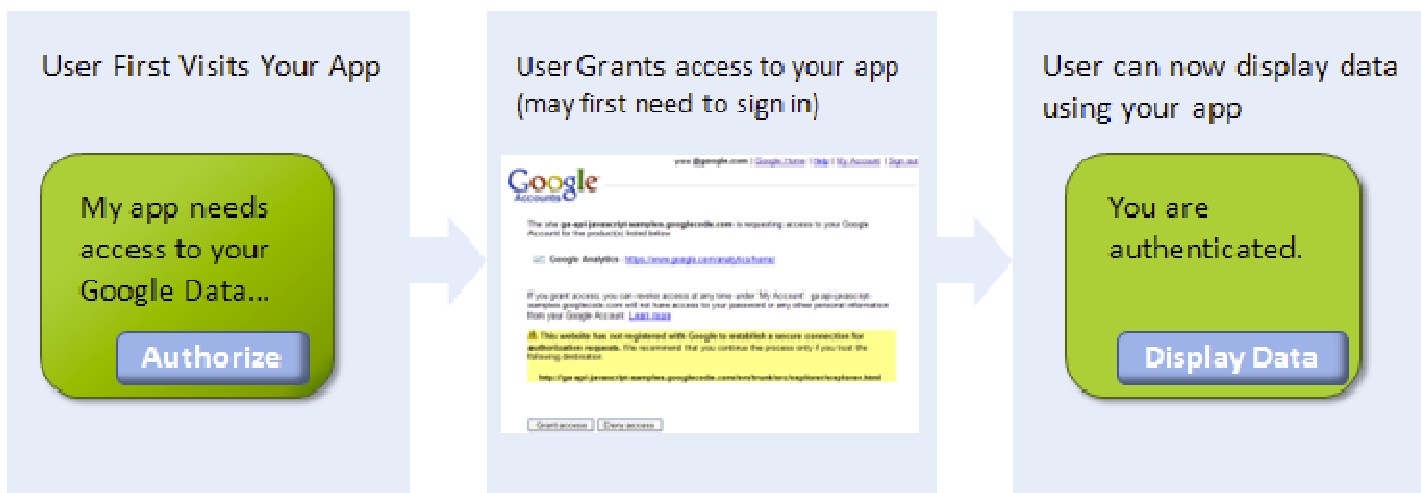


Figure 14: User Authorization Process [6]

The application will upload the log files created in the past 24 hours once every day.

5.8. Class Implementation

The three major classes in the application and the relationships between the classes are shown in Figure 15.

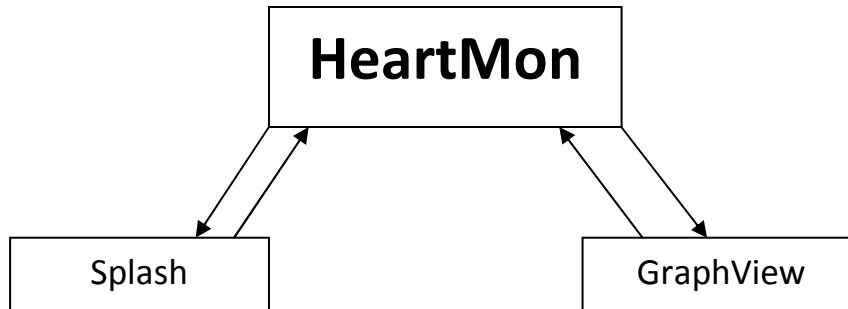


Figure 15: Diagram for the structure of the HeartMon Application Classes

Table 1 shows the responsibility of each class listed above.

Table 1: Class descriptions	
Class Name	Description
HeartMon	The HeartMon class acts as the entry point and the main function of the application. It is responsible for creating an activity for the application on the phone, displaying the application UI and handling all the application events, such as data analysis, warning notification, GPS location, data logging, and uploading data.
Splash	The splash display class is responsible for creating a splash screen with company info before entering the main UI of the application.
GraphView	The graph class is responsible for plotting the data received from the Bluetooth Arduino board on the application UI. The user will be able to see the graph once they enter the application UI.

6. System Test Plan

The system test plan will incorporate modularity in the beginning. Individual modules (sensors, hardware, and software) will be tested at first, and then the integrated system will be tested at the end.

6.1. Sensors

The readings from both types of sensors, ECG electrodes and the accelerometer will be tested using an oscilloscope. The range of values displayed by the readings will be verified against the expected values from datasheets.

6.2. Hardware

The ECG circuit will perform filtering and amplification on input signals from ECG sensors. The microcontroller will perform further filtering on input signals coming in from ECG circuit.

The analog and digital filtering are required to remove the body movement and shifts in body's DC noise below 0.5Hz; the noise from muscle activity above 100Hz, and a notch filter to remove noise from household power at 60Hz.

The output will once again be monitored using oscilloscopes to verify that the wanted signal is indeed within the desired bandwidth whereas the unwanted signal is not.

6.3. Software

At first, communication between the microcontroller and the Android phone over Bluetooth will be tested. Data will be sent back and forth to verify a bi-directional communication line. Accelerometer readings will be tested by performing various motions and verifying that they agree with the graph readings. GPS readings will be tested by obtaining the co-ordinates and verifying that they are accurate to about 10-50 meters. The algorithms to process the data will be tested by displaying output values of the samples received and plotting them on corresponding graphs. Also, values to trigger alarms and alerts will be arbitrarily provided to test critical cases. All the data during the tests should be logged and the log files are stored in phone SD memory. The log files will be tested by being viewed on the phone.

6.4. Integrated System

The overall system once integrated, will be subjected to various tests to ensure that the desired functions are being performed. Various scenarios will be incorporated to cover different situations. The ECG electrodes will be connected to a person who will display readings within the normal range. Correct measurement and basic functioning of the application will be verified from this. For another case, a person will perform jumping jacks with the accelerometer lying stationary to simulate irregular heart

behavior. In such a case an alert will be displayed. For the next case, the accelerometer will be moved vigorously with the person at rest which will display a different kind of alert. Similar experiments will be conducted based on the nature of the test.

6.5. False Alarms

BES recognizes the significance of detecting false alarms. For the prototype the values for the ECG signal amplitude, period, and ST segment will be measured. These values will be compared against some threshold values that will determine the well being of client. If the analysis detects a hard problem, the analyzing software will look for 10 consecutive and consistent indication of problem before it sends out an alarm. Then, it would be far less likely to have 10 false alarms in a row. More accurate error handling for false alarms will be implemented after completion of prototype.

7. Conclusion

This design specification document has discussed the proposed hardware and software design specifications used to implement functional requirements of the heart monitoring system *HeartMon™*. Justifications for the chosen hardware components and choices of solutions have been discussed. Specifications and flow-charts have been presented to illustrate how the overall system will perform, and a system test plan shows how the functionality and reliability of a proof-of-concept system will be verified. The proof of concept prototype model is expected to be finished and meet the capabilities and requirements of the *HeartMon™* system by April 10, 2011.

8. References

- [i] Wikipedia, “Bluetooth”, 2011. [Online] <http://en.wikipedia.org/wiki/Bluetooth> [Accessed 13 Mar, 2011]
- [ii] Wikipedia, “Healthcare Provider”, 2011. [Online] http://en.wikipedia.org/wiki/Health_professional [Accessed 13 Mar, 2011]
- [1] Analog Devices, “ECG Front-End Design is Simplified with MicroConverter”, 2003. [Online]. Available: <http://www.analog.com/library/analogdialogue/archives/37-11/ecg.html> [Accessed: 11 Mar, 2011]
- [2] Wikipedia, “Heart Rate”, 2011. [Online]. Available: http://en.wikipedia.org/wiki/Heart_rate [Accessed: 11 Mar, 2011]
- [3] American Heart Association, “Recommendations for Safe Current Limits for Electrocardiographs”, 1996. [Online]. Available: <http://circ.ahajournals.org/cgi/content/full/93/4/837> [Accessed: 11 Mar, 2011]
- [4] Online active filter design tool. [Online]. Available: <http://astro.u-strasbg.fr/~koppen/radio/Activee.html> [Accessed: 11 Mar, 2011]
- [5] B. Kaufmann, “Amarino”, 2009. [Online] Available: <http://www.amarino-toolkit.net/index.php/home.html> [Accessed 11 Feb, 2011]
- [6] Google Code, “userauthprocess”, 2011. [Online] Available: <http://code.google.com/apis/accounts/images/userauthprocess.png> [Accessed: 10 Mar, 2011]
- [7] Android Developers, “What is Android?”, 2011. [Online] <http://developer.android.com/guide/basics/what-is-android.html> [Accessed 19 Jan, 2011]
- [8] Arduino, “Arduinobt06”, 2010. [Online] http://arduino.cc/en/uploads/Main/arduino_bt06.pdf [Accessed 13 Mar, 2011]
- [9] Atmel, “Atml168”, 2010. [Online] http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf [Accessed 13Mar, 2011]
- [10] DataSheetArchive, “WT11”, 2010. [Online] <http://www.datasheetarchive.com/WT11%20BlueGiga-datasheet.html> [Accessed 13Mar, 2011]