

Copyright
by
Marco Antonio Moreno
2012

The report committee for Marco Antonio Moreno
certifies that this is the approved version of the following report:

An Android Hosted Bluetooth ECG Monitoring Device

APPROVED BY

SUPERVISING COMMITTEE:

Nan Sun, Supervisor

Jacob Abraham, Co-Supervisor

An Android Hosted Bluetooth ECG Monitoring Device

by

Marco Antonio Moreno, B.S.C.S.E.

REPORT

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2012

Dedicated to my family.

Acknowledgments

I would like to thank Professors Nan Sun and Jacob Abraham for their guidance on this project and report. Additionally, I wish to thank my friend and former co-worker Ron Graczyk for his advice and hardware debugging assistance.

An Android Hosted Bluetooth ECG Monitoring Device

Marco Antonio Moreno, M.S.E.

The University of Texas at Austin, 2012

Supervisors: Nan Sun
Jacob Abraham

This paper proposes a device capable of acquiring an electrocardiogram (ECG, EKG) signal to be hosted by a typical Android smartphone. Bluetooth is used as the data connection. Once acquired, the signal is graphed on the display of the smartphone. A basis of physiology behind the ECG is presented. The data acquisition system and the performance of the ECG amplification and supporting circuits are analyzed.

Table of Contents

Acknowledgments	v
Abstract	vi
List of Figures	ix
Chapter 1. Introduction	1
1.1 Extending the Smartphone Platform	1
1.2 Mobile Device Hardware Interfaces	2
1.3 Mobile Health Monitoring	3
Chapter 2. The Electrocardiogram	5
2.1 Important Diagnostic Tool	5
2.2 Einthoven's Triangle	6
2.3 Physiology of a Volume Conductor	9
2.4 Typical Lead-II Features	12
Chapter 3. The Data Acquisition System	14
3.1 The Arduino BT Platform	14
3.2 Android Platform and System Architecture	16
3.3 System Functionality	18
Chapter 4. ECG Front-End Signal Amplifier	20
4.1 Circuit Description	20
4.2 Implementation	23
4.3 Gain Calculation	25
4.4 Signal Recording and Spectrum Analysis	26
4.5 Signal To Noise Ratio	35
4.6 Power Consumption	38

Chapter 5. Discussion and Conclusions	40
5.1 Performance Results and Limitations	40
5.2 Future Work	40
Bibliography	42
Vita	45

List of Figures

2.1	ECG signals recorded by capillary electrometer (top) and string galvanometer (bottom)	7
2.2	Einthoven's Triangle	8
2.3	Heart Muscle as a Volume Conductor	10
2.4	Depolarization Wave Propagation to Differential Voltage . . .	12
2.5	Typical Lead-II Features	13
3.1	The ArduinoBT Platform	16
3.2	System Architecture	18
3.3	Lead-II Captured Data Real-Time Representation	19
4.1	The INA321 ECG Amplifier Circuit	21
4.2	ECG Amplifier Circuit Front-End for Arduino	24
4.3	ECG Front-End for Arduino	25
4.4	Left: 5mVpp Input, Right: 1.86Vpp Output (Gain = 372V/V)	26
4.5	Example Waveform Converted to Normalized Floating Point .	28
4.6	Modified Lead-II Configuration	29
4.7	Spectrum Analysis in Lab Environment (Signal Generator) . .	30
4.8	Spectrum Analysis in Lab Environment (Human Body)	31
4.9	Spectrum Analysis in Isolated Environment (Signal Generator)	32
4.10	Spectrum Analysis in Isolated Environment (Human Body) . .	33
4.11	Conventional SNR for Signal Generator Input	36
4.12	Results for SNR_{peak} and SNR_{crest}	36
4.13	Results for SNR_{inband}	36
4.14	SNR_{crest} measurements from [8]	37
4.15	SNR_{peak} measurements from [8]	37

Chapter 1

Introduction

1.1 Extending the Smartphone Platform

As smartphones and tablet computers have become increasingly popular, the market for value add components targeted to these devices continues to grow. The most ubiquitous way to sell content for mobile devices has been by selling “apps” in online marketplaces. These are usually pure software additions that utilize the existing platform’s hardware capabilities. The common set of hardware features for these devices has come to include high-resolution touch screens, powerful CPU’s, speakers and microphones, 3-axis accelerometers, and GPS, along with cellular, Wi-fi, near-field communications (NFC) and Bluetooth wireless data connectivity. These hardware interfaces provide a high quality standard environment for general application development. For more specialized applications, this set of hardware may not be sufficient, but does provide a capable basis as a user interface, network uplink, and recording platform.

1.2 Mobile Device Hardware Interfaces

So to provide some added functionality in the form of a hardware capability extension, what is the preferred method of integration? The answer to this is not straightforward and this discussion will be limited to Apple (iOS) and Android. In the case of Apple, iOS based applications can utilize the physical charging connector that is present on these devices. Interfaces on this connector include a UART type serial port, USB, audio, video, and firewire. To use any of these interfaces, however, requires special licensing [2]. For Android devices, the charging port usually takes the form of a standard USB connection and is much more open to add-on development. Using the USB Accessory APIs [3], it is possible to create external HW which is a USB host with the smartphone assuming the client role.

For wireless data connections, both Wi-fi and Bluetooth can suffice. The choice of these standards would depend on required data rate and power consumption. Wi-fi is better suited for higher bandwidth, higher power applications, while Bluetooth is preferred for lower speed and power. With Wi-fi, there is an added requirement of some form of router to connect both the mobile and add-on. In some cases, the smartphone itself can take on the “hot-spot” role, but this is not the standard. On the other hand, Bluetooth only requires a point to point connection to create a network. In addition to audio interfaces, Bluetooth supports a standard serial port type connection via the Serial Port Profile (SPP), also sometimes referred to as an RFCOMM connection. For Android devices, this type of Bluetooth support has been standard

since OS version 2.0 (API level 5) and still is at the time of this writing [1]. To do the same with an iOS device, however, extra licensing is required as is further Bluetooth authentication hardware [5]. Near-field communication is also an option, having found much use in the RFID and payment processing areas. However, the range on this medium is only a few centimeters and bandwidth is usually limited to small quantity data exchange.

1.3 Mobile Health Monitoring

Affordable healthcare is a difficult issue in today's world. As government agencies and private companies alike search for means of cutting costs, there may be market opportunity to leverage the ubiquity of mobile device technologies. Even in developing countries, capable low-end smartphone markets are expected to see growth in upcoming years [10]. There are several physiological data sets that could be interesting to the health industry for a variety of reasons. For example, one might experience certain symptoms that are recurring, but not readily reproducible in a clinical setting. A mobile device can be carried as normal with monitoring hardware to record various signals generated by the body, which may be uplinked to the network at any time. Further opportunities exist in the fitness and sports areas, where bio-feedback is helpful in tuning workouts for peak performance.

In order to explore the possibilities of smartphone health monitoring, this paper proposes a device capable of acquiring an electrocardiogram (ECG, EKG) signal to be hosted by a typical Android smartphone. Bluetooth is

used as the connection method due to the low power and bandwidth requirements of the ECG signal, and the desirability to link untethered. Furthermore, using Bluetooth allows the smartphone's USB connection to be reserved exclusively for the Android debugger, which is the most straightforward development setup. Once acquired, the signal is graphed on the display of the smartphone. The data acquisition system and the performance of the ECG amplification and supporting circuits will be analyzed. The physiology and acquisition method for the ECG signal is also presented.

Chapter 2

The Electrocardiogram

2.1 Important Diagnostic Tool

The electrocardiogram, also referred to as ECG or EKG interchangeably, is a measure of the electrical activity associated with heart function. When read by someone skilled in its interpretation, the graph of an ECG signal can indicate abnormalities in a patient's cardiac health. By the 1800's it was known that the heart produced electric potentials, but no proven system of non-invasive measurement had yet been devised. As we will see, the ECG voltage levels to be monitored at the skin's surface are less than 5mV in amplitude. This required sensitivity was a challenge to early scientists as the instrumentation at the time was based on various forms of physical deflection combined with some recording component. Today's instrumentation electronics have made capturing an accurate picture of the body's electrical activity much more accessible. However, a basic understanding of the signal's physiology is required to fully comprehend its importance.

2.2 Einthoven's Triangle

In the early years of the 1900's, a Dutch doctor named Willem Einthoven is credited as creating the first clinical quality ECG recording device. Einthoven studied an earlier method of low amplitude signal recording using an apparatus known as the Lippman electrometer (also called a capillary electrometer). This device consisted of a level of mercury suspended in a capillary tube with diluted sulfuric acid suspended above it by surface tension. Electrical activity introduced at electrodes placed onto the ends of the device would cause a vertical displacement of the mercury level. A recording over time was achieved by projecting the moving displacement onto scrolling photosensitized paper. After learning from and improving on the capillary electrometer approach, he set out to create a more accurate method of recording. He called his device the string galvanometer. Here, a thin quartz filament coated in silver is set in an electromagnetic field. When current is passed through the filament, it is deflected within the magnetic field. Once again, this movement was registered by photographic means onto a moving medium after magnification. Figure 2.1 shows an example of recorded signals acquired from Einthoven's devices [14]. In addition to recording the ECG with unprecedented precision, Einthoven also created a standard method and terminology for its signal acquisition that is still in use today.

The basic ECG configuration, called the "Einthoven Triangle", is a guide for electrode placement on the body to observe the heart's electrical activity from various perspectives. As different areas of the heart's muscle

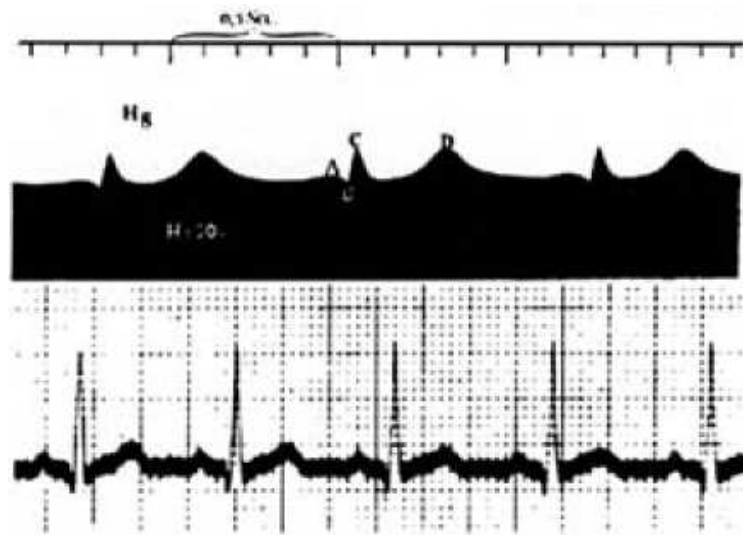


Figure 2.1: ECG signals recorded by capillary electrometer (top) and string galvanometer (bottom)

tissue experience depolarization and repolarization waves, the potentials across the 3 ECG leads making up this idealized triangle on the skin's surface can be observed to create a visualization of internal activity. The term “lead” in this context refers to a set of two differential electrodes placed across the heart to obtain a single ECG trace perspective. Figure 2.2 shows the standard 3 lead electrode placement. Note that the electrodes are named RA, LA, RL, and LL for Right Arm, Left Arm, Right Leg, and Left Leg respectively. The RL node is set to a reference potential by the measurement system in order to establish a common mode level for all differential leads. Additionally, depending on the implementation of the measurement system, this node may have active feedback to cancel common mode noise that is usually coupled into the body

by nearby power lines. The leads themselves are named Lead-I, Lead-II, and Lead-III and their values are defined in [7] as:

$$\begin{aligned} V_I &= (V_{LA} - V_{RL}) - (V_{RA} - V_{RL}) \\ V_{II} &= (V_{LL} - V_{RL}) - (V_{RA} - V_{RL}) \\ V_{III} &= (V_{LL} - V_{RL}) - (V_{LA} - V_{RL}) \end{aligned} \tag{2.1}$$

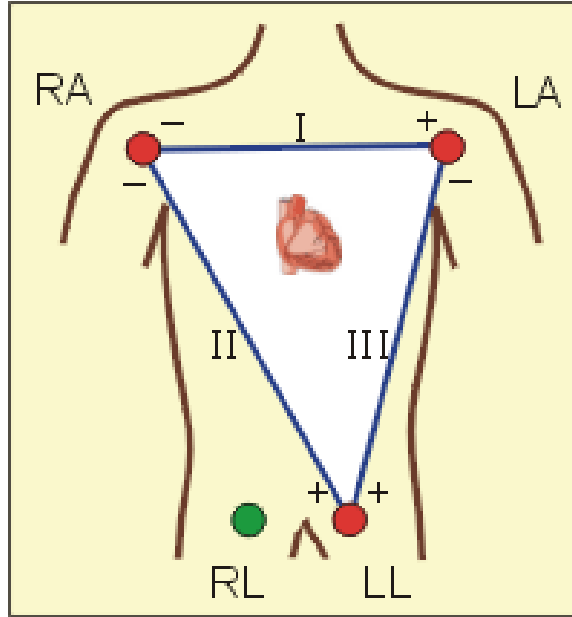


Figure 2.2: Einthoven's Triangle

We can now see that the signal from each of these leads is nothing more than a measure of the differential voltage between two electrodes. Despite this simplicity, the ECG is a powerful diagnostic tool. In order to understand how internal activity is inferred from external monitoring, we will look briefly at the physiology that is generating this voltage differential.

2.3 Physiology of a Volume Conductor

Cardiomyocytes are the cells that make up the myocardium, which is the muscle tissue in the walls that surround the heart. Each cardiomyocyte contains ions principally in the form of Potassium, Sodium, and Calcium. As these ions move in and out of each cell along their respective chemical gradients, a variation of surface to internal cell voltage potential is created. When at rest, these cells are said to be polarized, referring to the fact that the outer layer of each cell is about 90mV higher in potential than the inner area. When a cell becomes depolarized corresponding to the “firing” and thus contraction of the cell, the surface potential lowers. In aggregate, different regions of the heart muscle are polarized and depolarized during varying periods of the cardiac cycle. The potentials resulting from these transient states are the essence of what the electrocardiogram captures.

In order to visualize this, consider a uniform flat piece of heart muscle tissue placed between the leads of a voltmeter. Figure 2.3 (taken from [12]) demonstrates the concept of a volume conductor. At rest, the surface of the tissue is said to be polarized such that the electrical potential across the entire surface is constant and the meter will read a flat line. When a stimulus is introduced to the material, a chain reaction occurs which causes adjacent cells to fire and become depolarized. This creates a wave front, a depolarization wave, as the stimulus spreads throughout the muscle.

As the depolarized mass increases from the negative terminal to the positive, the voltmeter will register a positive increasing potential difference.

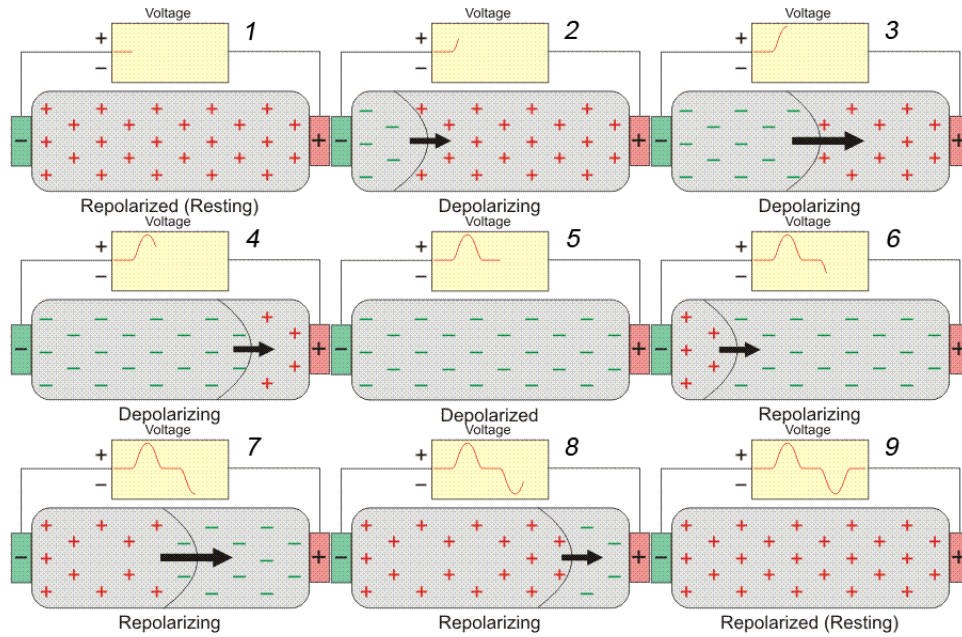


Figure 2.3: Heart Muscle as a Volume Conductor

This difference reaches a maximum value when half of the tissue is depolarized. Further depolarization will begin to bring the surface of the muscle back to a uniform potential, so the voltage across the terminals will now decrease back to zero. An exact opposite effect will occur when the tissue is repolarized in the same fashion. In fact, there are a set of rules that dictate how a volume conductor will register on a differential set of leads [12]. These principles are listed below and figure 2.4 is provided for further visualization.

1. A wave of depolarization traveling toward a positive electrode results in a positive deflection in the ECG trace.
2. A wave of depolarization traveling away from a positive electrode results

in a negative deflection.

3. A wave of repolarization traveling toward a positive electrode results in a negative deflection.
4. A wave of repolarization traveling away from a positive electrode results in a positive deflection.
5. A wave of depolarization or repolarization traveling perpendicular to an electrode axis results in a biphasic deflection of equal positive and negative voltages (i.e., no net deflection).
6. The instantaneous amplitude of the measured potentials depends upon the orientation of the positive electrode relative to the mean electrical vector.
7. The voltage amplitude is directly related to the mass of tissue undergoing depolarization or repolarization.

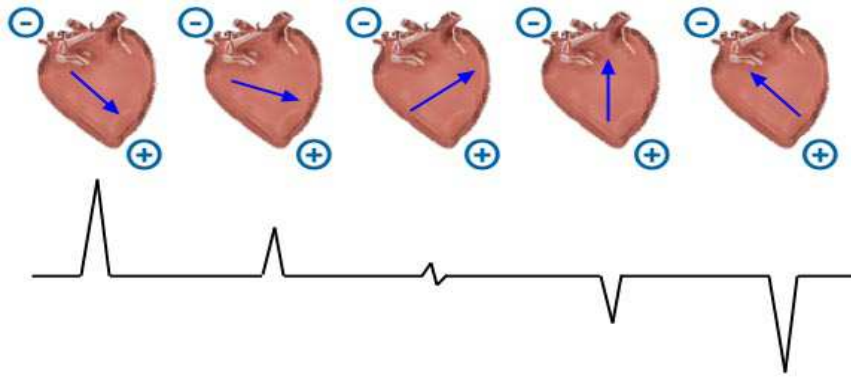


Figure 2.4: Depolarization Wave Propagation to Differential Voltage

2.4 Typical Lead-II Features

In many clinical settings, if only one lead may be monitored, it will be the Lead-II configuration. This is because Lead-II is oriented along the overall direction of depolarization, the majority of which travels from the RA node down to the LL node. For this reason, it is the most useful lead for detecting cardiac arrhythmias (abnormal electrical activity of the heart) [13]. The measurements for this report will therefore be taken in the Lead-II configuration.

Figure 2.5 shows the typical features of an ECG waveform when capturing only the Lead-II electrode placement of Einthoven's Triangle. [7] It can be seen from this graphic that each major wave or peak is a combination of several different factors whose explanation is beyond the scope of this report. However, this can be used to visually compare with the output of the proposed Android ECG capturing device. It is a goal of this project to be able to visually identify at least the P, Q, R, S and T peaks in recordings and on the

mobile device's screen.

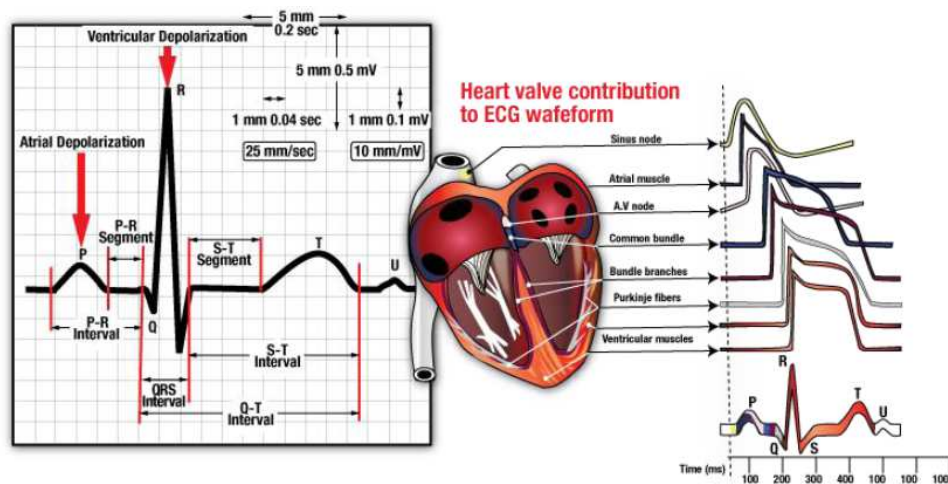


Figure 2.5: Typical Lead-II Features

Chapter 3

The Data Acquisition System

In order to bring sampled data into the Android environment, a data acquisition system is required. This system must be capable of sampling an analog signal at a near constant rate and transmit the resulting data to the Android device. Furthermore, once the data is received, there must be some means of displaying the ECG signal in a familiar format that an experienced medical professional would be able to read. The details of the components used to achieve all the required data acquisition functionality for this project are presented here.

3.1 The Arduino BT Platform

The Arduino BT development kit is an open source hardware product that consists of a small Atmel ATmega328 main processor and a WT11 Bluetooth transceiver by BlueGiga. The WT11 has its own internal processor that handles the Bluetooth protocol tasks and effectively presents the main processor with a UART interface. This gives the Arduino BT standard Bluetooth connectivity such that it can be paired with any device that implements the Bluetooth Serial Port Profile (SPP). Once the Bluetooth link has been

paired and connected, the ATMega processor may read and write bytes over the UART link to communicate with the remotely connected device.

The kit includes a software development environment that runs on a PC. The ATMega328 has been pre-programmed with a bootloader capable of transferring a new program from the PC over the Bluetooth link and burning it to system flash. After downloading, the new program is then run and will be run on subsequent restarts. Programming for the Arduino platform is greatly simplified by the library interfaces that are provided to read and write UART data, as well as perform A/D conversions. Under the hood is found the open-source WinAVR toolkit, which contains the C compiler for the ATMega328. Two rows of headers are present on the board to allow hardware expansion, which is where the ECG front-end amplifier will be placed. A debugging header also exists, but requires a special cable that is not included in the basic kit. The set of library functions supplied are very straightforward, so the programming tasks required for this platform did not warrant the use of a debugger.



Figure 3.1: The ArduinoBT Platform

3.2 Android Platform and System Architecture

The Android development environment consists mainly of the Eclipse IDE and a framework of Java based components targeted for mobile devices. The IDE comes with a mobile device emulator, which may be used for debugging many aspects of application development. This emulator is slow to load, but once it is up and running, it is a suitable debugging target. There is no Bluetooth emulation, however, so ultimately a real device needs to be used. The specific Android model used for this project is the HTC Droid Incredible.

Downloadable example code provided by the android.com web site was customized to achieve a working ECG monitoring device. Initial testing was done using the BluetoothChat example project. This example code provides a bare-bone server-client system capable of pairing two Bluetooth devices and

creating an RFCOMM socket connection between them. Furthermore, an input box with keyboard functionality is presented to the user to type messages to the remote device. The idea here is that you would run this example on two mobile devices and messages may be typed back and forth over the Bluetooth link which is shown on a screen log.

To receive the sampled data from the Arduino BT platform, the Bluetooth Chat buffer handling was modified to suit needs. A chat server module is responsible for passing buffers of raw data to and from the Bluetooth API's provided by the Android OS. A client module ultimately is the provider and consumer of the transmit and receive buffers. To verify the consistency of the received data, a test program was written for the Arduino BT platform which generated and transmitted 8-bit LFSR data. The Android side was also programmed with the same algorithm to generate LFSR data to compare with the incoming stream. The chat log functionality was modified to log communications statistics, and it was found that at least a 10 kilobyte per second transfer rate was sustainable with no errors at the application layer. The maximum UART rate on the Arduino BT is 115200 bits per second, which corresponds to 14400 bytes per second, but the theoretical limit for Bluetooth is 2.1M bits per second. For the ECG signal sampled at 200Hz, only 400 bytes per second data rate is required due to the fact that the 10-bit A/D converted codes are transmitted as 16-bit words.

With basic wireless connectivity achieved, the graphical presentation of the ECG signal was the next major piece of functionality to be implemented.

For this, an example called TouchPaint was customized to provide an oscilloscope type trace across the screen. With this example, the user is allowed to draw on the screen by dragging of the finger and the resulting marks on the screen will slowly fade away until they disappear entirely. This fading is achieved by the use of a block filling procedure which manipulates the alpha channel level of the final rendering. Modifications were made to replace the touch screen input with the incoming data from the Bluetooth server. Some further structural modifications to this example were made to create the EKGView component, which is the module responsible for rendering the ECG trace on the screen in real-time as data is received over Bluetooth. All these components are shown in figure 3.2 along with the Arduino BT platform's functionality to represent the full system architecture.

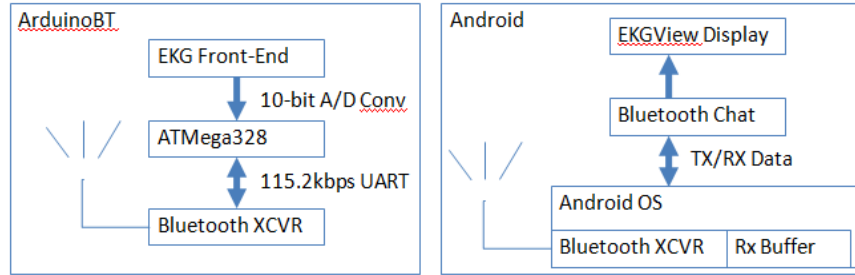


Figure 3.2: System Architecture

3.3 System Functionality

With all the system pieces in place, including the amplifier front end, system integration testing could be done. After starting the app on the An-

droid mobile device, a few menu items are used to connect to the Arduino BT board. Once connected, the Arduino BT will wait until a character is received from the Android side and will begin to make A/D conversions, which are transmitted back to the Android platform. A snapshot of the real-time system functionality is shown in figure 3.3 along with the visual identification of expected lead-II waveform features.

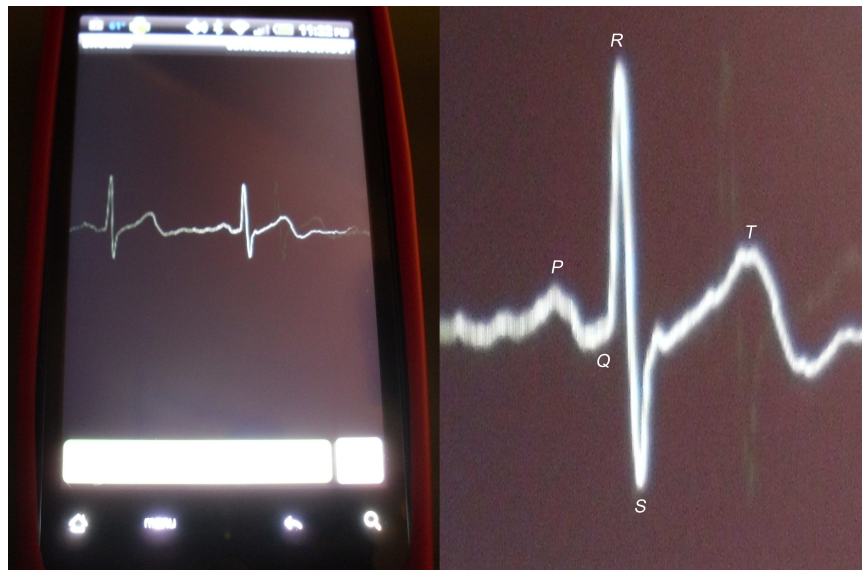


Figure 3.3: Lead-II Captured Data Real-Time Representation

Chapter 4

ECG Front-End Signal Amplifier

Electronics advances in the past hundred years since the original Einthoven experimentation have indeed made the ECG signal acquisition much more accessible. The first string galvanometer weighed in at 600 pounds and took up 2 rooms. Now a capable data acquisition system is small enough to be handheld and powered by a set of AA batteries. A proper commercial clinical quality ECG monitor will need to go through various certification processes in order to be sold in the medical market. However, a simpler non-commercial monitor can be constructed out of just a few semiconductor amplifier parts combined with the necessary discrete passive components.

4.1 Circuit Description

Figure 4.1 illustrates the circuit that was used in this project. It is based on the INA321 instrumentation amplifier (in-amp) from Texas Instruments and is presented in the datasheet [9]. The INA321 was chosen due to its rail-to-rail operation, which can be easily integrated into a single supply data acquisition system. If a bi-polar amplifier was to be used, the supply requirements would need to be modified to include both positive and negative

power supply voltages centered around a common ground. Since the Arduino development kit has the A/D converter referenced at a single 5V rail, this seemed to be the best choice.

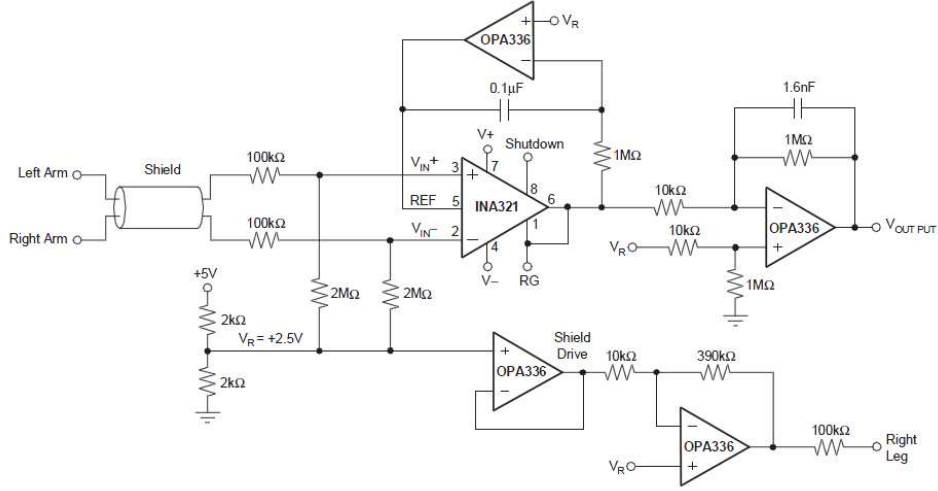


Figure 4.1: The INA321 ECG Amplifier Circuit

To begin with, the circuit has a few terminals to be connected externally. This system will have a bi-polar set of electrodes attached to the patient’s skin to create a single ECG “lead”. These electrodes will be run through a shielded cable and connected to the positive and negative inputs of the in-amp via the $100k\Omega$ isolation resistors. The shielding material of the cable will be connected to the “Shield Drive” node of the circuit. The “Right Leg” output is essentially a reference node for the bi-polar electrodes and will be run through the shielded cable as well to an electrode on the patient’s skin.

The circuit will be powered ideally by a single 5V supply rail, but should

also operate well from 2.7V to 5.5V as stated in the INA321's datasheet. The V_R node is set by the voltage divider on the 5V supply, and will ideally be the voltage midway between the supply and ground which is nominally 2.5V. This is used to set the common mode level of the inputs to the in-amp through the $2M\Omega$ resistors. Additionally, V_R sets the voltage level on the cable shielding through the unity gain configured op-amp, as well as the "Right Leg" driver through the inverting op-amp with gain of $390k\Omega/10k\Omega = 39$. The "Right Leg" driver is a negative feedback that applies an inverted representation of the common mode noise to the body in order to minimize its effect.

The REF input on pin 5 of the INA321 sets the zero output voltage level. This must be driven with a low impedance source to prevent CMRR degradation. Thus, an op-amp is used in an integrator configuration. The integrator effectively cancels any output offset and attempts to keep the output null level matched to V_R . The in-amp has a gain setting of 5 in the configuration shown and the final output op-amp is configured for a gain of $100k\Omega/10k\Omega = 100$ with a cutoff frequency of $1/(2\pi \times 1.6nF \times 1M\Omega) = 99.5Hz$. Thus, with a supply of 5V, V_{OUTPUT} is centered at 2.5V giving a total gain of $5 \times 100 = 500$. The differential inputs to the in-amp for a typical ECG are said to be less than 5mV with a frequency content of less than 100Hz. At 5mV, the input would ideally be gained to $0.005V \times 500 = 2.5V$, which would achieve rail-to-rail output.

4.2 Implementation

Figure 4.2 shows the schematic implementation of the example amplifier circuit from the INA321 datasheet. A couple of customizations were made to add some flexibility in the design. For instance pads for gain configuration resistors R9 and R10 were added to allow for gain to be set as desired depending on system performance. Furthermore, some resistors were added to allow microprocessor control of the INA321 shutdown pin, or just set it ON or OFF. An LED was also added to be used by software. Finally, the output op-amp low-pass filter capacitor value was modified to be $4.7\mu F$ in order to reduce the cutoff frequency to about $33.9Hz$. This was done to reduce the effects of 60 Hz noise coupling from nearby power lines. This schematic was fabricated as a 2 layer PCB that can mate with the Arduino development kit main board (see figure 4.3).

The PCB design flow was performed using the free PCB Artist tools from Advanced Circuits. Signal routing was mostly achieved using the autoroute feature of this software. Parts were placed near where expected good routes could be made and placement was iteratively refined until the number of routes made to connect all nodes was minimized. After routing, a copper fill ground plane area was made on the bottom layer. The 4 ground contacts at the corners of the board were added to allow the addition of an RF shielding material to cover the circuit in case the Bluetooth transceiver added noise to the system. This shielding was ultimately not necessary to the design.

During hardware debug, the $1M\Omega$ resistor R15 was found to be prob-

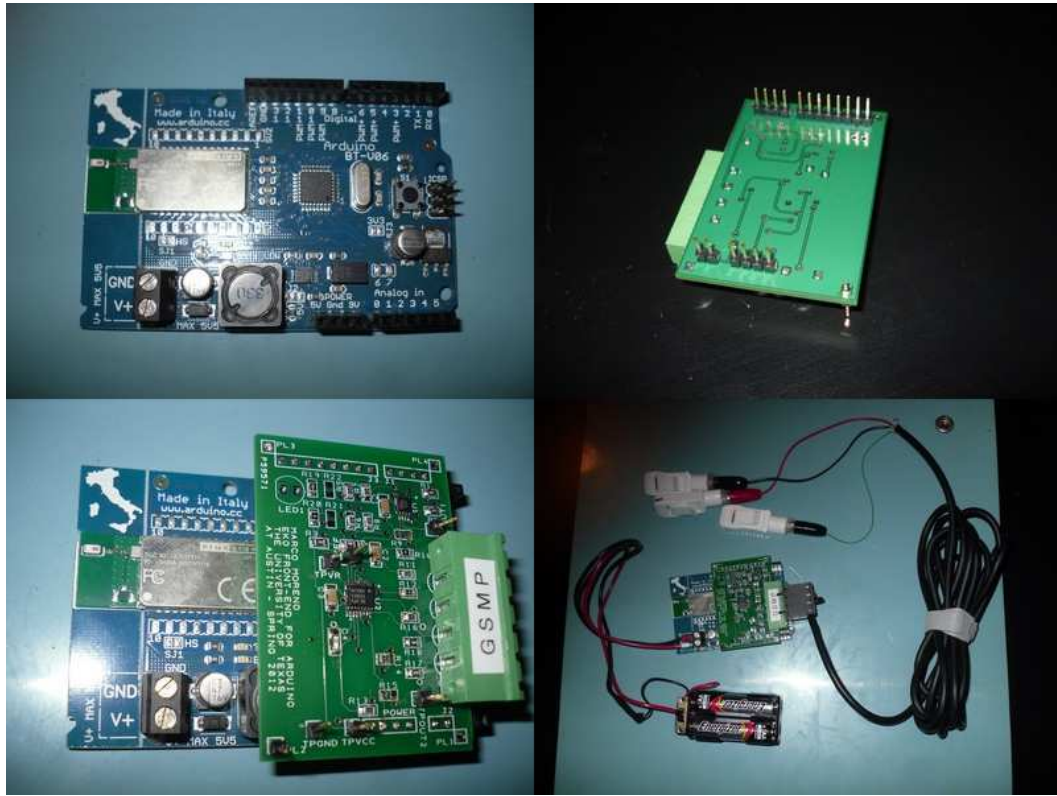


Figure 4.3: ECG Front-End for Arduino

4.3 Gain Calculation

Initial testing was done by injecting 5mV_{pp} , 20Hz sine wave given by a precision function generator and taking measurements with an oscilloscope. It is important to note that care was taken to not reference the system to earth ground, so both the signal generator and oscilloscope were powered without the third prong on their power cables being used. This way, the function generator could be referenced to the “Right Leg” node to better simulate the configuration when probing to a human subject. The output sine wave is

measured at 1.86Vpp, with a DC offset of V_R . This shows the actual system gain achieved to be $1.86/0.005 = 372V/V$ (see figure 4.4).

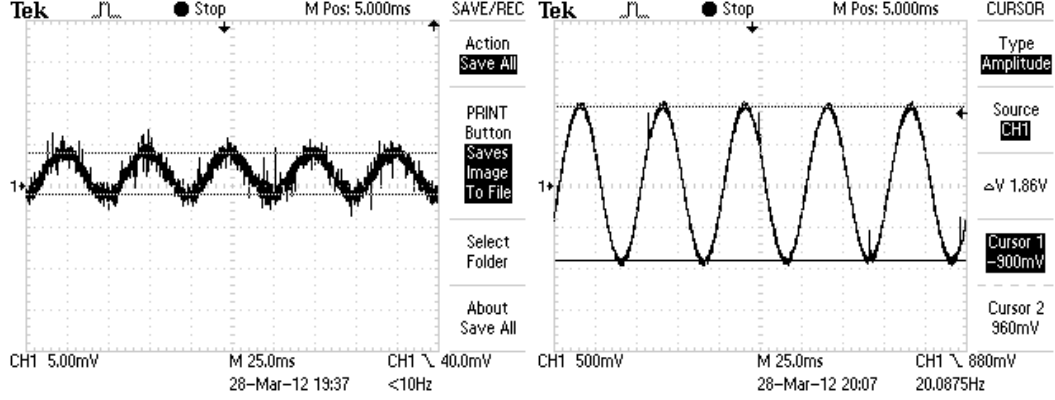


Figure 4.4: Left: 5mVpp Input, Right: 1.86Vpp Output (Gain = 372V/V)

4.4 Signal Recording and Spectrum Analysis

Further measurements were taken using the on-chip A/D converter of the Arduino's ATMega328 processor due to the fact that probing the board with oscilloscope probes further couples noise into the system. The processor was setup in a loop to capture the waveform in the following steps: 1) Perform A/D conversion, 2) Send 2-byte result over UART to BT transceiver, 3) Delay 5ms and repeat. Ideally, this should achieve a sampling rate of about 200Hz. Using a logic analyzer capable of capturing data sent over the UART w/ timestamping information, a rate test was performed. The captured data was exported and further processed using scripts written in the Lua programming language. It was determined that the average sampling rate is actually about

192Hz. This corresponds to an average of 5.209ms, which varied down to a minimum of 5.207ms and up to a maximum of 5.217ms. This puts the maximum sampling rate jitter at $\pm 8\mu s$. Note that the ATmega328 processor is capable of driving the A/D converter with a square wave input, which could make the sampling rate an interrupt driven function and minimize rate error and jitter. This method was not used due to time constraints.

Unfortunately, capturing data with the logic analyzer was also found to be coupling noise into the system and corrupting the A/D conversion accuracy. Further analysis was done by capturing the conversion results over the Bluetooth link. Since the Arduino's Bluetooth transceiver implements the standard Serial Port Profile (SPP), a PC with a Bluetooth USB dongle was used in combination with logging via TeraTerm, a serial terminal program. In this way, the 10-bit conversion results are sent over Bluetooth as 16-bit data to a binary log file and once again further processed with Lua scripts. This data was then converted to 32-bit floating point data in the range of +1 to -1 in order to import into Audacity, an audio processing software. To convert from 10-bit words to floating point, each sample is processed as $FloatSample = (TenBitSample - 512)/512$. The results of this are such that the conversion code of 512 will correspond to a floating point 0, the code of 1023 will be near +1 (0.998 actually) and the code of 0 will be at -1. This is illustrated in figure 4.5, which shows a captured ECG waveform displayed in Audacity. Notice also that the final output op-amp effectively inverts the polarity of the expected deflection. Therefore, the final floating point samples

are inverted to reverse this by multiplying by -1 . Finally, we can estimate the input voltage differential to the system if we use the peak shown at about the 0.2 level in this figure. The full scale reference is $5V$, with the null level being $2.5V$, so the voltage deflection presented to the A/D converter input is $0.2 \times 2.5V = 0.5V$. If we divide this by the system gain we get an input voltage of $0.5V/372 = 1.34mV$, which is near the expected value of about $1mV$ from figure 2.5.

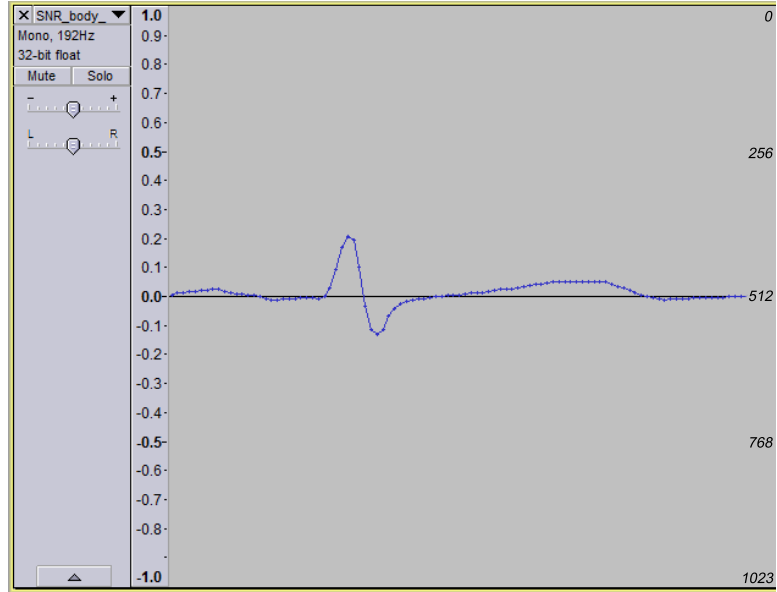


Figure 4.5: Example Waveform Converted to Normalized Floating Point

Two sets of recordings were taken in order to further study system performance. The first set was taken in a noisy electronics lab environment, which demonstrates the effects of power line noise coupling into the circuit. The system was then setup in an isolated environment, specifically a microwave

oven, in order to shield the system from external noise sources. In both cases, recordings were taken with the reference 20Hz tone input from a signal generator, as well as real human body input. For recording with human body input, typical gel adhesive electrodes were placed on the chest in a modified lead-II configuration as shown in figure 4.6. A shielded USB cable was modified to create the connection with the data acquisition system. To obtain baseline noise recordings, the minus and plus inputs to the in-amp were shorted.

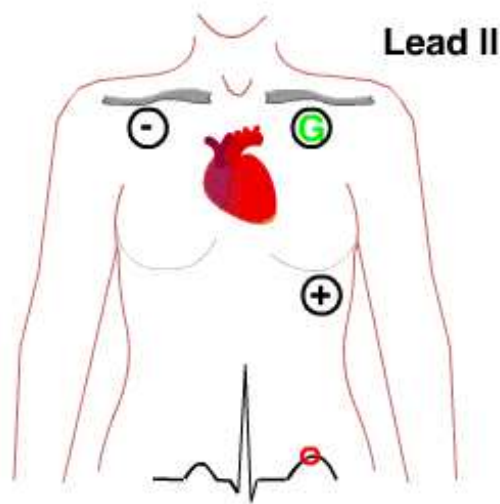


Figure 4.6: Modified Lead-II Configuration

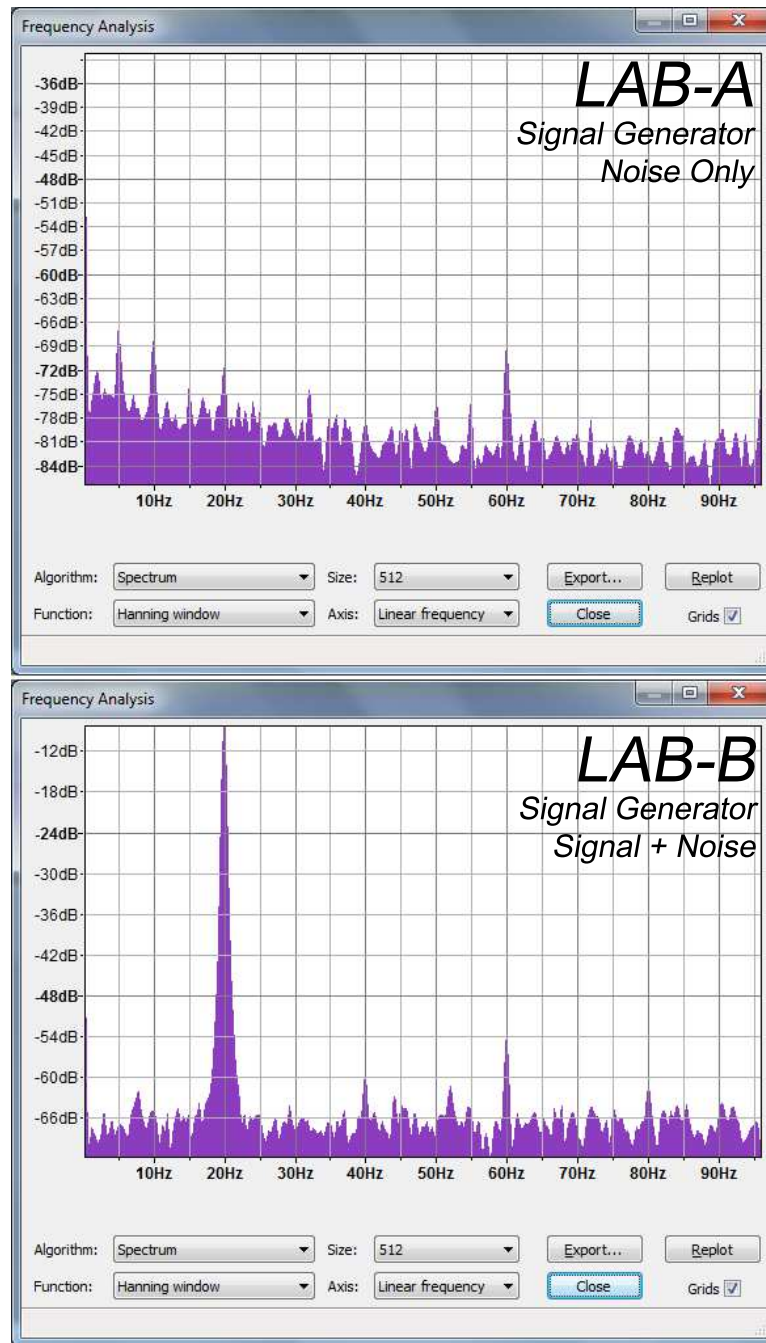


Figure 4.7: Spectrum Analysis in Lab Environment (Signal Generator)

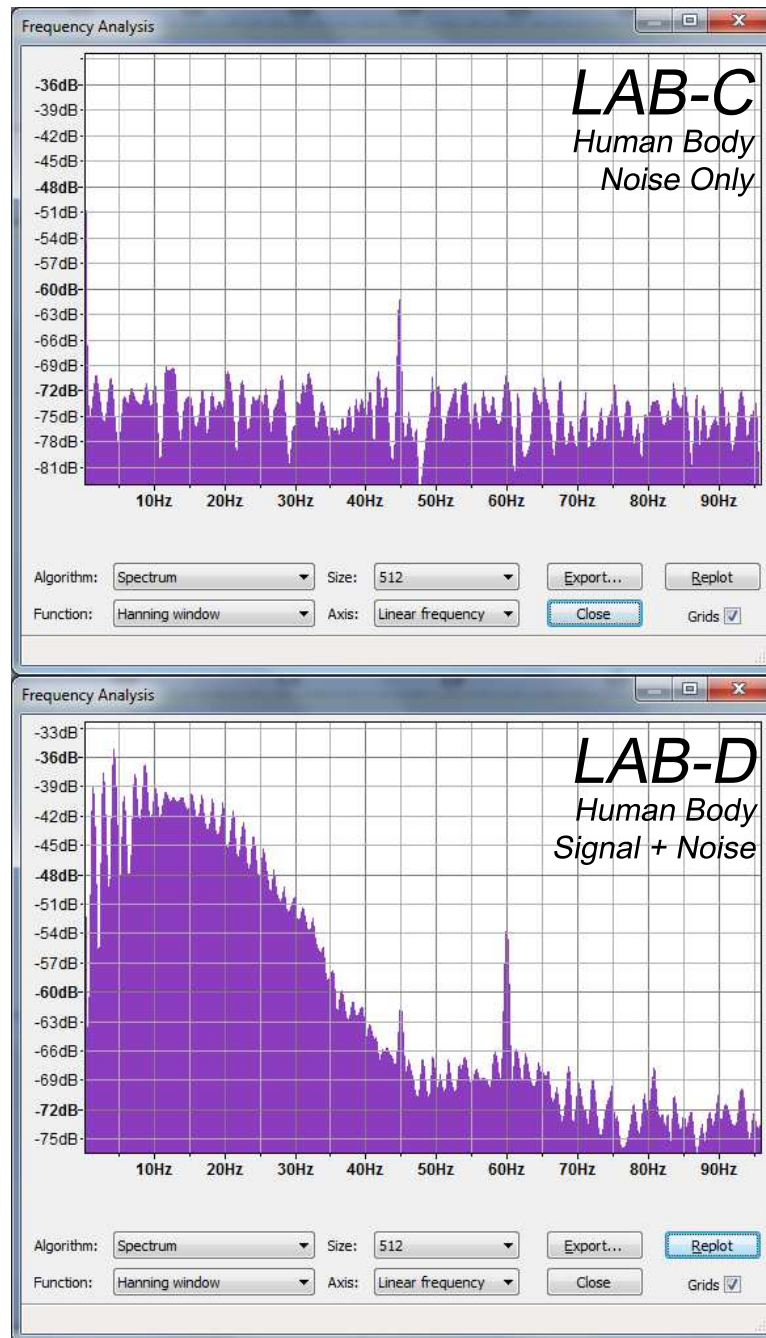


Figure 4.8: Spectrum Analysis in Lab Environment (Human Body)

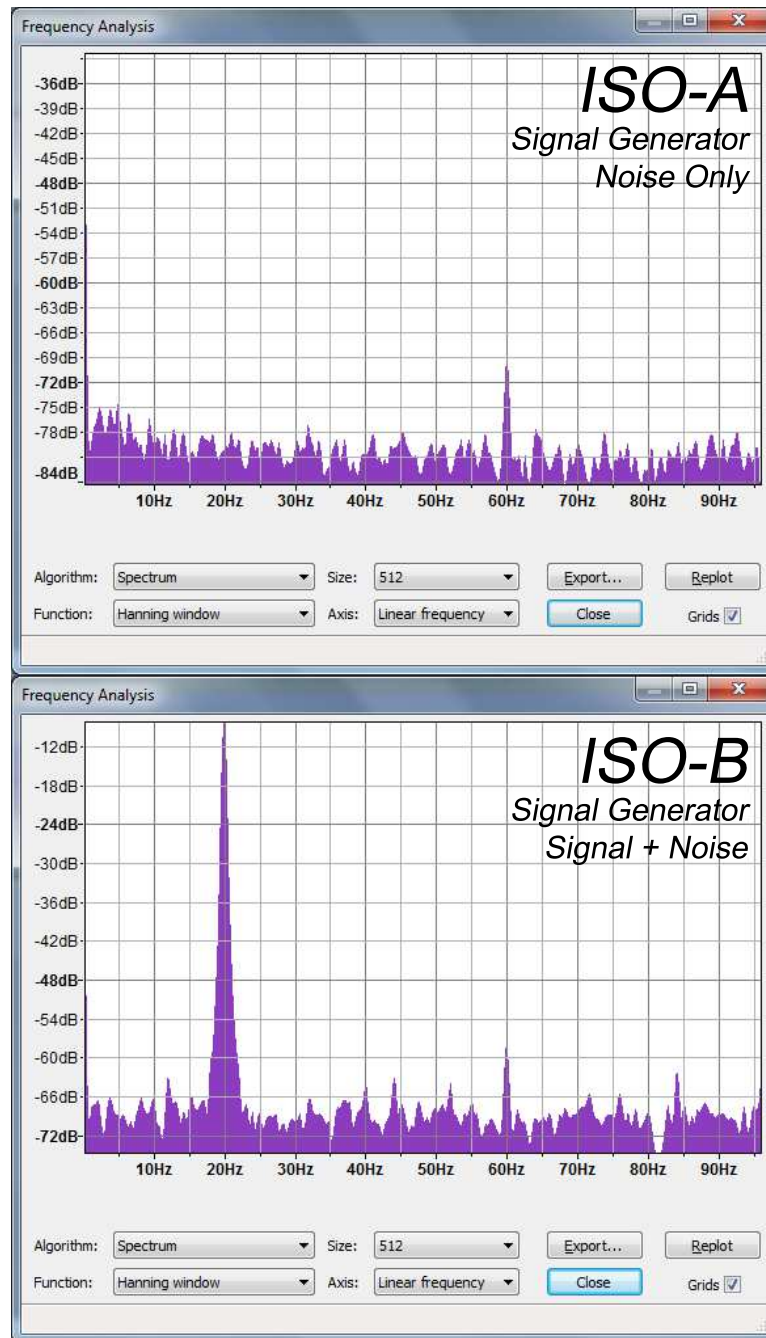


Figure 4.9: Spectrum Analysis in Isolated Environment (Signal Generator)

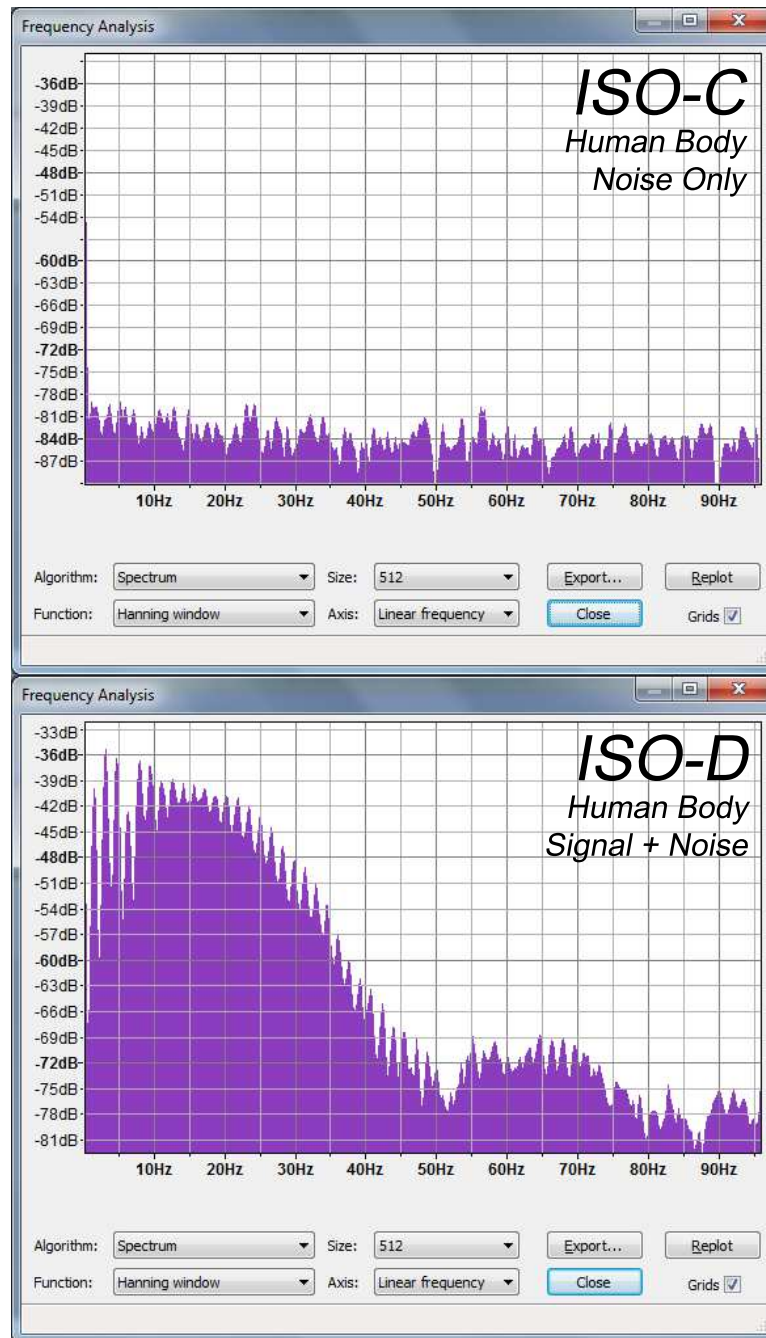


Figure 4.10: Spectrum Analysis in Isolated Environment (Human Body)

As expected, there is a peak at 60Hz in many of the recordings due to power line coupling. Figure 4.7 LAB-A shows a few other spikes under the 33Hz cutoff frequency of the output stage. In the presence of the 20Hz tone, LAB-B shows that the only significant frequency contributions are at the input frequency and 60Hz. Figure 4.8 LAB-C oddly indicates a lone noise contributor at 45Hz near the -62dB level. This is carried over into LAB-D at the same power level along with a now higher power 60Hz peak. The source of this 45Hz energy is unclear, but it is only present in the lab environment when testing with the human body.

Moving on to figure 4.9 ISO-A, we see that the signal generator is still coupling in power line noise. This may be explained by the fact that the signal generator reference was left connected to the “Right Leg” node during this recording. Furthermore, in all the isolated recordings, the microwave door needed to be left slightly ajar to allow both the instrumentation cable as well as the USB extender cable for the PC’s Bluetooth adapter. As with the LAB-B recording, the ISO-B graph also shows the only major contributors are at 20Hz and 60Hz. Figure 4.10 ISO-C shows that in the isolated environment, virtually no noise was present in the output. As with the LAB-D graph, the ISO-D recording shows significant frequency content distributed from DC up to around the expected cutoff frequency.

4.5 Signal To Noise Ratio

The SNR can be calculated in a conventional way using the recordings when the input is the signal generator.

$$SNR = 20\log_{10}(V_{RMS,signal}/V_{RMS,noise}) \quad (4.1)$$

Using this definition, the table in figure 4.11 gives the SNR calculations for both the lab and isolated environments. It appears that the isolated environment is only slightly better in SNR, but this may be expected as the spectrum analysis of both of these scenarios were very similar. However, this definition of SNR only holds meaning due to the fact that the input signal is a regular tone type. The information held in an ECG recording is different than the typical periodic content, so we will use a modified definition of signal quality in this case. Three altered versions of SNR have been proposed in [6] and are listed here along with designations to be used for later reference:

1. SNR_{peak} - Ratio of the R-peak amplitude to the noise amplitude in the isoelectric region
2. SNR_{crest} - Crest factor / peak-to-RMS ratio (the ratio of the peak value of a signal to its RMS value)
3. SNR_{inband} - Ratio between in-band (5 to 40 Hz) and out-of-band spectral power

Environment	$A_{\text{RMS,signal}}$	$A_{\text{RMS,noise}}$	Conventional SNR
Lab	0.272	0.011	27.86dB
Isolated	0.274	0.011	27.93dB

Figure 4.11: Conventional SNR for Signal Generator Input

These definitions are further defined in [8] along with empirical values that can be compared against. To obtain estimates for SNR_{peak} and SNR_{crest} , the peak-to-peak value of a single typical QRS complex is extracted. This value is then be compared against the maximum peak-to-peak value of the isoelectric region and the RMS value of the entire recording respectively. Finally, to obtain an estimated SNR_{inband} , the spectral data from a recording is exported from Audacity and the average power from 5Hz to 40Hz will be taken and compared to the average power from the remainder of the spectrum. These results are shown in figures 4.12 and 4.13. The SNR_{peak} and SNR_{crest} numbers may be found comparable to results from [8] in figures 4.14 and 4.15.

Environment	A_{p-p}	$A_{\text{isoelectric}}$	RMS	SNR_{peak}	SNR_{crest}
Lab	0.3496	0.176	0.0424	19.9	8.25
Isolated	0.3242	0.00976	0.0428	33.2	7.57

Figure 4.12: Results for SNR_{peak} and SNR_{crest}

Environment	P_{inband}	P_{outband}	SNR_{inband}
Lab	-47.5dB	-69.7dB	22.2dB
Isolated	-47.5dB	-73.8dB	26.3dB

Figure 4.13: Results for SNR_{inband}

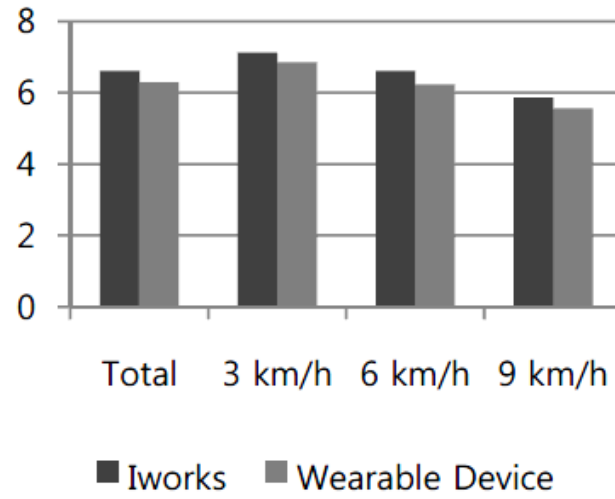


Figure 4.14: SNR_{crest} measurements from [8]

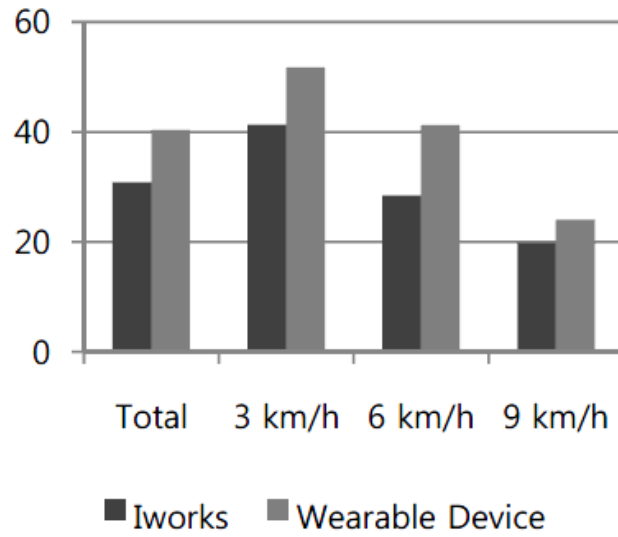


Figure 4.15: SNR_{peak} measurements from [8]

4.6 Power Consumption

For the Arduino BT module using a 4 AA series battery pack for power, the resulting supply voltage was 5.8V. The ECG front-end amplifier circuit by itself draws just 1.6mA. With the Bluetooth transceiver not in use, but still discoverable, the full system has an average current of 27mA. The current consumption would spike up to 41.6mA max briefly at about a 1 second interval. Most likely, this is due to the transceiver waking up periodically to perform some tasks. When actively transmitting 16-bit samples at 192Hz, the total system current was 50.3mA on average, and peaked at 53.6mA max. If we use an estimated 2000mAh for the AA battery pack, this system will run for about 37 hours. This would be sufficient for use as a 24 Holter monitor.

The Android platform is somewhat less straightforward to make power estimates on due to the variability in the system. To get an idea of what run time would be if the display was kept on solid, the data acquisition was left running starting with a full 1300mAh system battery. It took about 3 hours of run time to drain the battery to 15%, which would be down to about 200mAh remaining. So, if the system drained 1100mAh in 3 hours, then we can estimate current consumption at about 366mA. The Android battery information listed these factors for most power consumption:

1. 90% Display
2. 4% Android OS
3. 3% Wifi

4. 2% Cell Standby

5. 1% Bluetooth

If we were to disregard everything but the OS and Bluetooth usage, we can say that the system uses about 5% of the 366mA, which is about 18.3mA. This is useful to know if we were to modify the software to just log the data to the internal SD card. A SanDisk product manual states that the max current while writing to an SD card is 75mA. If we use this to make a rough estimate on how long we could log data with the screen off, the full 1300mAh battery should run this logging only system for about 14 hours max. The battery capacity would need to be about doubled to achieve a recording system capable of running for 24 hours.

Chapter 5

Discussion and Conclusions

5.1 Performance Results and Limitations

It can be seen that the proposed device performed adequately for the ECG signal with suitable run times depending on battery capacity. However, a commercially available form of this system would need to be certified against established medical practices and standards. As a learning device, this system can provide many examples of the noise artifacts that most commercially available systems have already filtered out. It could be a good platform for the evaluation of various signal processing techniques. Aside from the ECG signal, this system can be adapted to many other bio-potential signals which may be of interest to health monitoring. In fact, this system demonstrates that mobile device data acquisition is feasible, despite a few challenges. There is opportunity to use the demonstrated data connection for various other purposes such as robotic telemetry and remote control.

5.2 Future Work

Most future work on this system would be to add the various filtering and isolation pieces necessary to get a clinical quality signal. Apart from the

front end amplifier, the wireless connection would need more robustness features such as verifying the consistency of data in the presence of noise and low signal strength. The Arduino BT board is also lacking the necessary connections to implement hardware flow control to the Bluetooth UART, which would need to be corrected to prevent buffer overflow in the transceiver. A peak detector needs to be added to supply the classic beep indicator of heart pacing. This would also give rise to the calculation of pulse rate from the average distance between peaks.

Additionally, some work remains to get a correctly referenced display waveform, with the correct 1mm to 0.1mV vertical grid scale and a 25mm per second horizontal scale. This can pose challenges due to the variability of screen sizes and resolutions from different suppliers. Android application development in the Bluetooth area can apparently be a mixed bag. Each mobile device maker has the ability to make customizations to the system which can vary the behavior of the standard APIs. This can lead to situations where the app may work only on some platforms, and not others, even when the hardware capability is present. In general, this is an issue with Android development, which is not as pronounced in the iPhone due to the consistency among different iPhone and iPad devices.

Bibliography

- [1] android.com. Bluetoothsocket. 2012. Available at: "<http://developer.android.com/reference/android/bluetooth/BluetoothSocket.html>".
- [2] android.com. Mfi program. 2012. Available at: "<https://developer.apple.com/programs/mfi/>".
- [3] android.com. Usb accessory. 2012. Available at: "<http://developer.android.com/guide/topics/usb/accessory.html>".
- [4] Apple. ios: Supported bluetooth profiles. 2010. Available at: "<http://support.apple.com/kb/HT3647>".
- [5] connectBlue. Oem bluetooth iphone / ios accessory module obs414. 2012. Available at: "<http://www.connectblue.com/products/bluetooth-products/bluetooth-modules/bluetooth-iphone-ios-accessory-module-obs414/>".
- [6] Patrick McSharry Gari D. Clifford, Francisco Azuaje. *Advanced Methods And Tools for ECG Data Analysis*. Artech House Publishers, 2006.
- [7] Matthew Hann. Analog fundamentals of the ecg signal chain. Available at: "http://www.deyisupport.com/cfs-filesystemfile.ashx/___key/CommunityServer-Discussions-Components-Files/60/3264.TI_5F00_ECG_5F00_Fundamentals_5F00_condensed_5F00_v08.pdf".

- [8] Y.W. Jang I.B. Lee, S.C. Shin. Comparison of conductive fabric sensor and ag-agcl sensor under motion artifacts. *30th Annual International IEEE EMBS Conference*, 2008.
- [9] Texas Instruments Incorporated. micropower, single-supply, cmos instrumentation amplifier. 2000. Available at: "<http://www.ti.com/product/ina321>".
- [10] R. Colin Johnson. Smartphones' downward push going over top. 2011. Available at: "<http://www.smartertechnology.com/c/a/Social-Business/Smartphones-Downward-Push-Going-Over-Top/>".
- [11] Richard E. Klabunde. Volume conductor principles and ecg rules of interpretation. *Cardiovascular Physiology Concepts*, 2007. Available at: "<http://www.cvphysiology.com/Arrhythmias/A014.htm>".
- [12] Naomi L. Burtneck Larry P. Tilley. *ECG for the Small Animal Practitioner*. Teton NewMedia, 1999. Available at: "<http://research.vet.upenn.edu/smallanimalcardiology/ECGTutorial/ECGLeads/tabid/4961/Default.aspx>".
- [13] Dr Juliette Lee. Ecg monitoring in theatre. *Update in Anaesthesia*, 2000. Available at: "http://www.nda.ox.ac.uk/wfsa/html/u11/u1105_01.htm".
- [14] Joseph Varon Moises Rivera-Ruiz, Christian Cajavilca. Einthoven's string galvanometer. *Texas Heart Institute Journal*, 2008. Available at:

"<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2435435/?tool=pubmed>".

Vita

Marco Moreno was born in Dallas, Texas. He received the Bachelor of Science degree in Computer Science and Engineering from The University of Texas at Arlington in 1998. Since then, he has held various consulting and full time roles with emphasis on real-time embedded systems and audio/video. In 2009, he decided he had too much free time and began graduate studies in Electrical and Computer Engineering at The University of Texas at Austin. With graduation approaching, he is looking forward to having his life back. He may be reached at marcoamorenottx@gmail.com.

This report was typeset with L^AT_EX[†] by the author.

[†]L^AT_EX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T_EX Program.