BRADLEY
University

Real-time Electrocardiogram Monitoring

Project Proposal

Department of Electrical and Computer Engineering

Calvin Walden, Edward Sandor, and Nicholas Clark

Advisors: Dr. Yufeng Lu and Dr. In Soo Ahn

December 5, 2016

Table of Contents

I. Introduction

Arrhythmias are irregular heartbeats that occur when the electrical signals controlling the heart's muscular contractions become malformed[1]. While these occur occasionally in healthy people, frequent occurrences can be a symptom of heart disease. The most common form of arrhythmia is premature ventricular contraction (PVC)[2]. When experienced in succession, PVCs cause a patient's heart to fail to circulate the necessary volume of blood through the body. Holter monitors allow physicians to use electrocardiograms (ECGs) to monitor a patient's heartbeat and diagnose irregularities[3]. However, with present technology, the ECG data must be analyzed after it is acquired, and arrhythmias cannot be detected as they are occurring[4].

Previously, the project *Real-time Heart Monitoring and ECG Signal Processing*[5] addressed the need for real-time ECG signal processing, and successfully implemented a PVC detection algorithm. While the final product could process ECG data directly acquired from a sensor in real-time, it was only used to test pre-recorded, benchmark ECG data from the MIT-BIH arrhythmia database.

This project aims to implement a system that can be realized as a standalone medical device that could be comfortably worn by an outpatient. Taking this project closer to this goal involves re-evaluating the implementation of the PVC detection algorithm and the choice of embedded platform for operability, connectivity, and battery life. Interface hardware and software for ECG sensors will also be added.

II. Previous Work

The project *Real-time Heart Monitoring and ECG Signal Processing*[5] looked into the application of detecting arrhythmias in real-time. It aimed to improve the Holter monitor by implementing the Pan-Tompkins and Template Matching algorithms on the low-power Texas Instruments TI CC3200 microcontroller development platform. The portable and mobile system demonstrated that it could successfully transmit an alert message wirelessly in the event of an arrhythmia as it occurred.

Due to the limited memory and computing power in the chosen platform, some modifications to the ECG signal processing algorithms were made. This was due in part to the combined processor load of the real-time signal processing, wireless communication, and data logging on the device.

The previous project planned to design a complete system to process ECG data acquired in real-time from sensors. However, a sensor interface was not implemented, and benchmark ECG data demonstrating various arrhythmias was used throughout its development instead.

III. Applicable Standards

Standards are guidelines agreed upon in an industry that insure understandable and successful implementation and interfacing of systems. There are four applicable standards that have been identified for this project.

1. IEEE 802.11

This IEEE standard outlines a wireless networking topology, in this case, WiFi. This project is based on a Raspberry Pi 3 embedded platform, which is IEEE 802.11n compliant at 2.4GHz.

2. IEEE 11073

This IEEE standard outlines the communication between a medical device and an external device such as a phone or computer. It is important because it states how the communication must occur in real-time, so that there may be consistent data transfer between devices. This might be considered at a later point in the project if it is applicable for a device ready for production.

3. MISRA C

The Motor Industry Software Reliability Association (MISRA) has developed coding standards for C language. This ensures safety, security, portability, and reliability of C code across embedded systems. Developers across numerous technical industries have adopted these standards, including the medical device industry.

4. RS-232

This serial communication specification is the initial form of communication to external devices from the device in this project. Data and settings can be transferred between this device and a host computer.

5. FDA Guidance

It is important to note that the Food and Drug Administration (FDA) provides guidance for ECG devices under the EC-11 and EC-13 standards; however, these standards do not fully apply to the device in this project.

IV. System Level Design

The overall project system consists of a number of subsystems. The main system controller is the primary system as it arbitrates communication and activity between all of the other subsystems.

1. System Block Diagram

The system block diagram is shown in Figure 1. The main system controller is at the center, which is the primary controller for the system. It communicates with the DSP in order to read processed data and enable signal processing. It also stores the processed data, and produces alerts based on the data processing results.

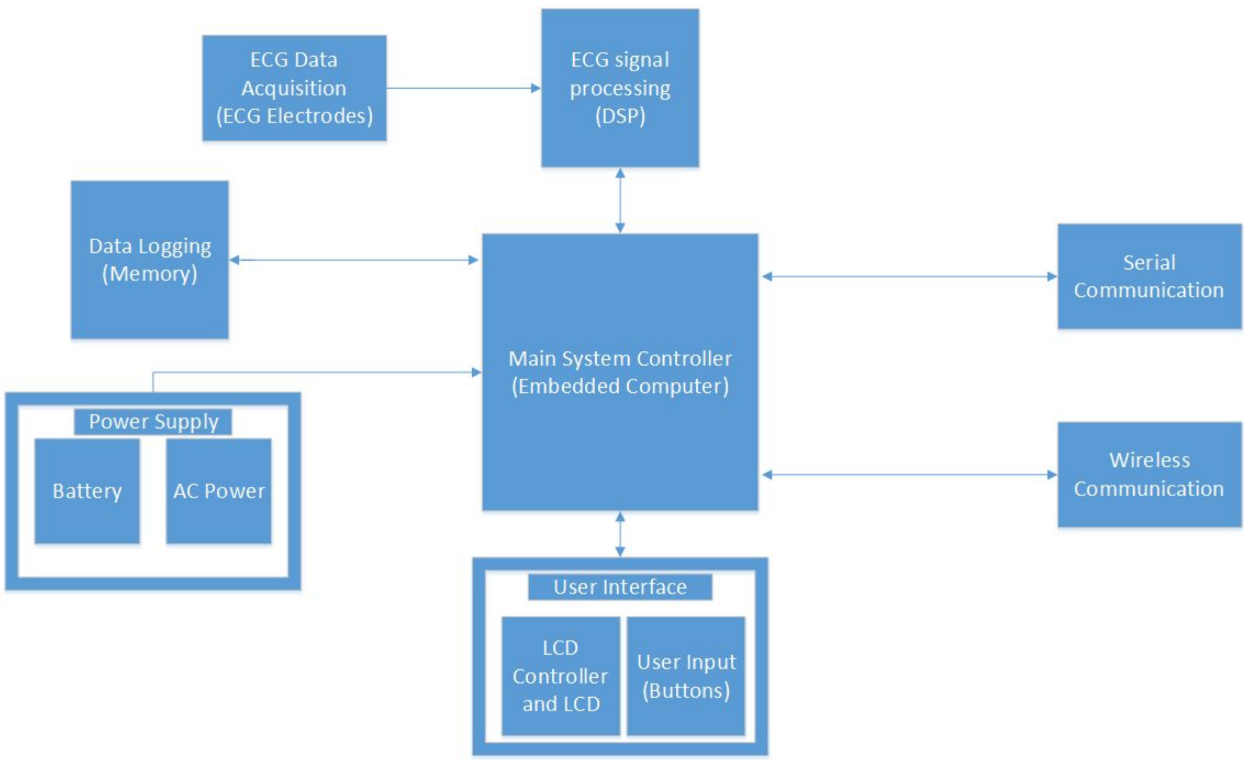

Figure 1 - System block diagram

2. Subsystems

The subsystems in this project are shown in Table 1.

Table 1 - Subsystems

Subsystem	**Description**
Main system controller	Controls functionality and data flow between all other subsystems; largely implemented on main microprocessor. User interface elements, wireless communication, and serial communication are implemented exclusively on this subsystem.
ECG data acquisition	Acquires ECG signal data from ECG sensor interface or stored benchmark data to provide to ECG Signal Processing subsystem. Raw ECG data may undergo prefiltering.
ECG signal processing	Processes and analyzes raw data ECG data; largely implemented as DSP hardware/software.
Serial communication	Interfaces main system controller and external computer for setting device parameters and viewing debugging information.
Wireless communication	Sends event information from ECG signal analysis to client.
LCD controller	Interfaces with LCD to display system status and parameters, depending on selected system mode and user settings.
User input	Push buttons provide input to main system controller to select modes.
Data logging	Stores history of ECG signal data taken over specified time period in a rolling buffer; writes all system events to a logfile for debugging purposes.
Power supply	Selects between a battery or AC power supply and provides each subsystem with regulated power.

3. Functionality Flowchart

The flowchart in Figure 2 shows the progression of one system cycle. The main system controller enables ECG data acquisition and digital signal processing. The DSP then processes the raw data and transmits it to the main system controller. The main system controller then stores the data, and transmits a message if PVC is detected.

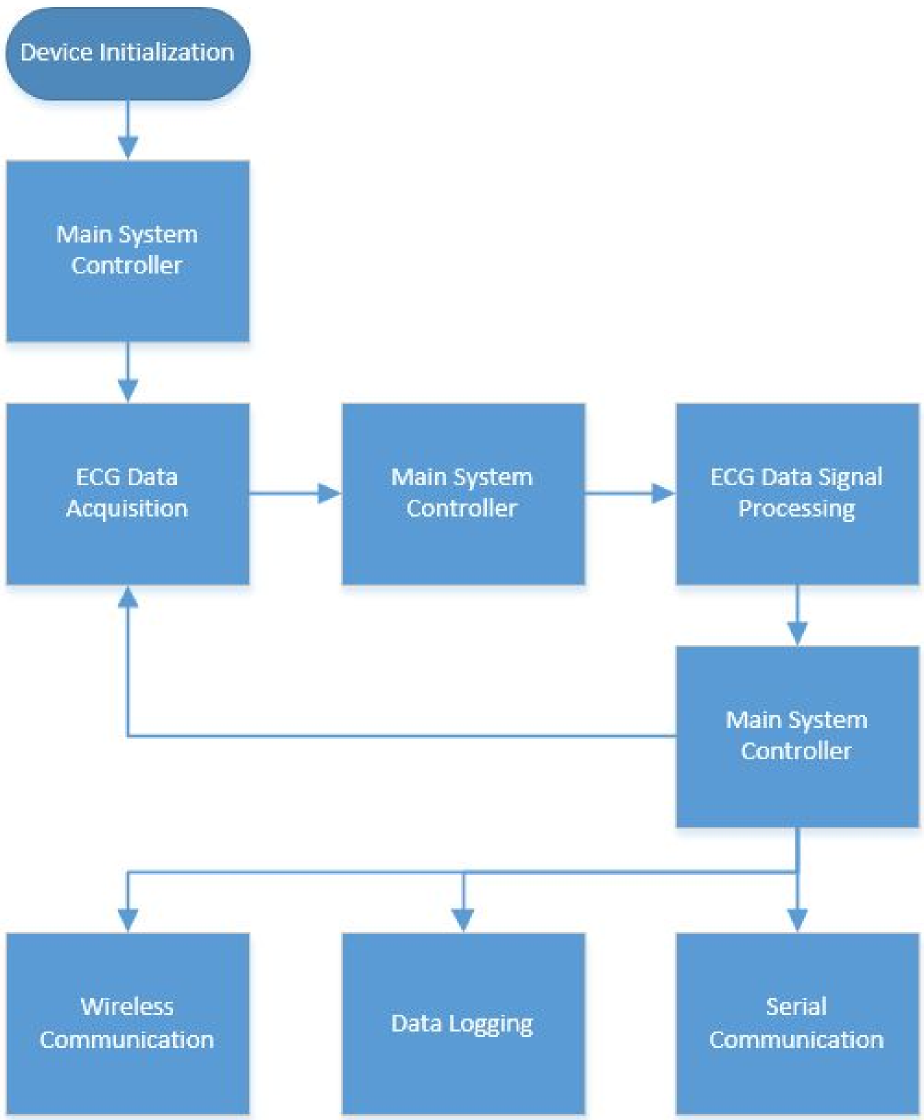

Figure 2 - System flowchart

4. Hardware Components

Table 2 provides a list of hardware components used in the system. It also shows which subsystems are implemented on each piece of hardware, and what communication interface is used between the component and the main system controller, if applicable.

Table 2 - Hardware components

Hardware Component	**Implemented Subsystem(s)**	**Interface to main system controller**
Raspberry Pi 3	Main system controller	N/A
	Data logging	Built-in
	Serial communication	USB
	Wireless communication	Built-in
Texas Instruments TMDX5515EZDSP	ECG data acquisition	SPI
	ECG signal processing	SPI
Adafruit RGB Negative 16x2 LCD+Keypad Kit for Raspberry Pi	LCD controller	I^2C
	User input	I^2C
Battery	Power Supply	Power supply input (from USB)

V. Completed Engineering Efforts

There are a number of engineering efforts completed to date. These include research, simulation, design, and experimentation.

1. Research

A significant amount of research has been expanded for the project, centered around the understanding of ECG signal and the algorithms used to detect PVCs.

Arrhythmias

There are six types of arrhythmias that have been identified through this research. These conditions are listed in Table 3.

Table 3 - Arrhythmias[1]

Arrhythmia	Description
Atrial Fibrillation	Disorganized upper chamber contraction
Tachycardia	Fast heart rate
Bradycardia	Slow heart rate
Conduction Disorders	Irregular heartbeat
Premature Contraction	Early heartbeat
Ventricular Fibrillation	Disorganized lower chamber contraction

From this group of six, atrial fibrillation (AF) has been identified as one of the most common and deadly form of arrhythmias. When an AF occurs, the upper chambers of the heart begin to quiver instead of pumping blood, which causes blood to pool and clot over time[6]. The clot can then enter the circulation system, and create a blockage causing a heart attack or stroke. Since the symptoms caused by this, irregular heartbeat, dizziness, fatigue, and shortness of breath, are like those of other medical conditions, the best way to identify AF is through the use of an ECG. In the ECG waveform sampled on a patient experiencing AF, there is no distinct P-wave. This is due to the quivering of the atria in the upper chambers of the heart. This quivering also results in irregular RR-intervals.

Similarly, PVC is a common arrhythmia, but appears to a patient as a skipped beat. While it is not a deadly arrhythmia, PVC can be a serious condition if left untreated[7]. PVC results in inefficient blood circulation, which can lead to organ damage. Like AF, PVC has no distinct P-wave and irregular RR-intervals, and has a noticeably wider QRS complex.

QRS Peak Detection

In order to identify most arrhythmias, there needs to be some form of QRS peak detection. Two sets of research have identified the Pan-Tompkins (Hamilton-Tompkins) algorithm as the simplest and most efficient algorithm to use on embedded devices[8][9]. This algorithm cleans the raw ECG signal using a bandpass filter then emphasises the QRS signal using differentiation, squaring, and an moving average filter. The algorithm then applies a set of rules to set an adaptive threshold to detect QRS based on the bandpass filtered signal and the QRS emphasised signal.

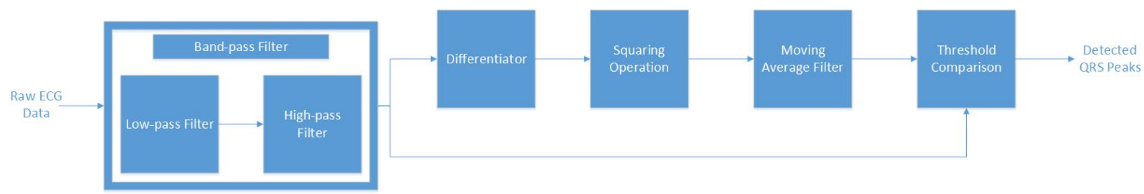

Figure 3 - Flowchart of the Pan-Tompkins Algorithm

Template Matching

After QRS peaks have been detected, a template matching algorithm is used to determine if a beat is healthy or is an arrhythmia.[11] This algorithm performs a wavelet transform on the raw data. When first started, the algorithm creates a template for a healthy heartbeat for the particular patient based on the best of the first beats detected. When a QRS peak is detected by the Pan-Tompkins algorithm, the raw signal is correlated with the template and the beat is determined to be healthy based on a threshold.

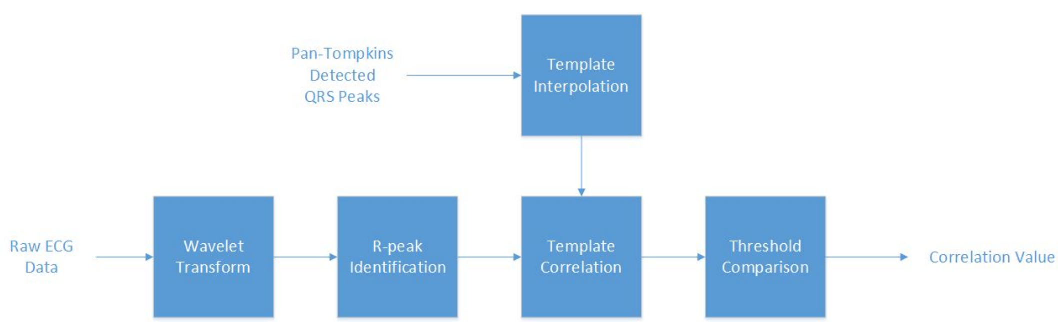

Figure 4 - Flowchart of the Template Matching Algorithm

2. Simulation

A new MATLAB simulation of the algorithm has been developed to prepare its hardware implementation on the new architecture. Compared with the simulation work from previous project, the new simulation evaluated the complexity of algorithm. The code is to be optimized for data acquisition on the TMDX5515EZDSP digital signal processor. So far, the Pan-Tompkins algorithm has been implemented and tested using benchmark ECG data from the MIT-BIH database. Figure 5 shows the result of identifying the R peaks of three QRS complexes. Next, the template matching algorithm is to be implemented in the new simulation.

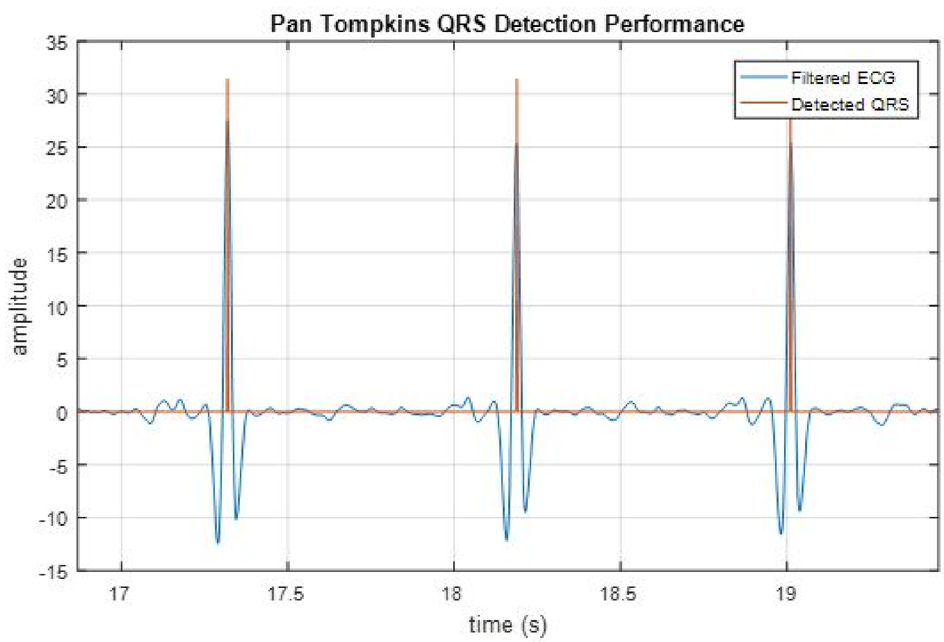

Figure 5 - Simulation result of the Pan Tompkins QRS algorithm

3. Design

The system level design is shown in Figure 1. Subsystem functionality is described in Table 1. This includes the ECG data processing core and the main system controller.

4. Experimentation

Initial experimentation has been performed with the AD8232 single lead ECG pre-filter board. Figure 6 shows the experimental setup. The ECG pre-filter board is powered by the 3.3 volt DC output of the voltage regulator on the Raspberry Pi 3.

A single ECG sensor lead is attached to the chest of one of the group members. A sample signal from the ECG pre-filter board's output pin is acquired by an oscilloscope (refer to Figure 7). It shows that the sample, with prominent QRS peaks, could be processed by the Pan-Tompkins algorithm. It is also observed that there is a significant amount of 60 Hz noise present in the acquired ECG signal. It can be filtered out using a bandpass filter in the signal processing subsystem.

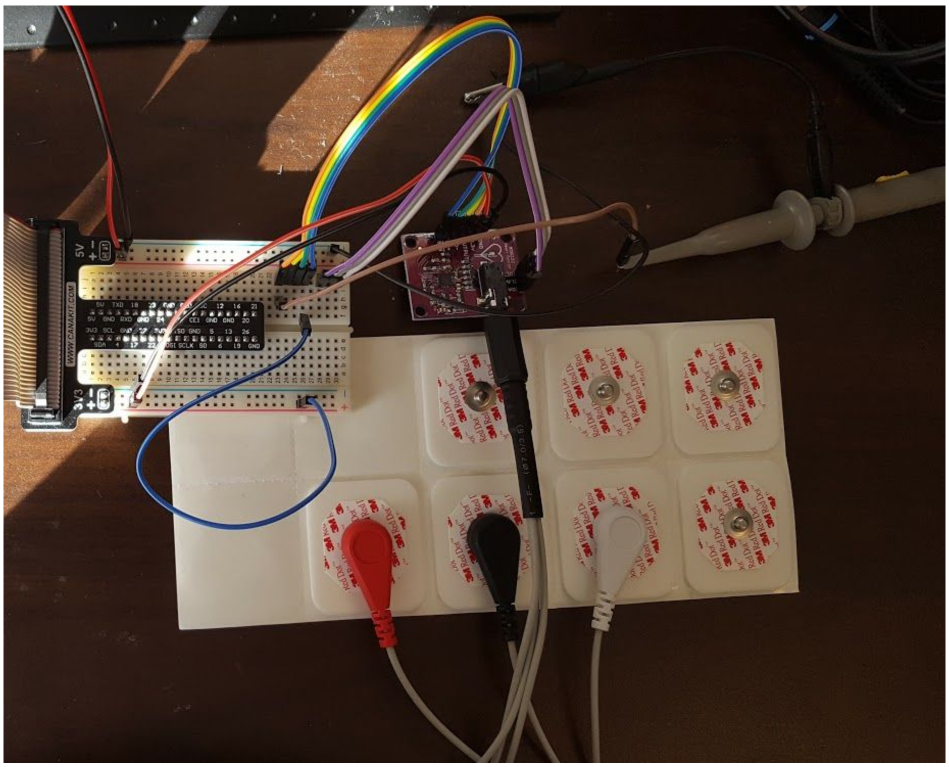

Figure 6 - Setup used to test the ECG pre-filter board using the Raspberry Pi 3 voltage regulator

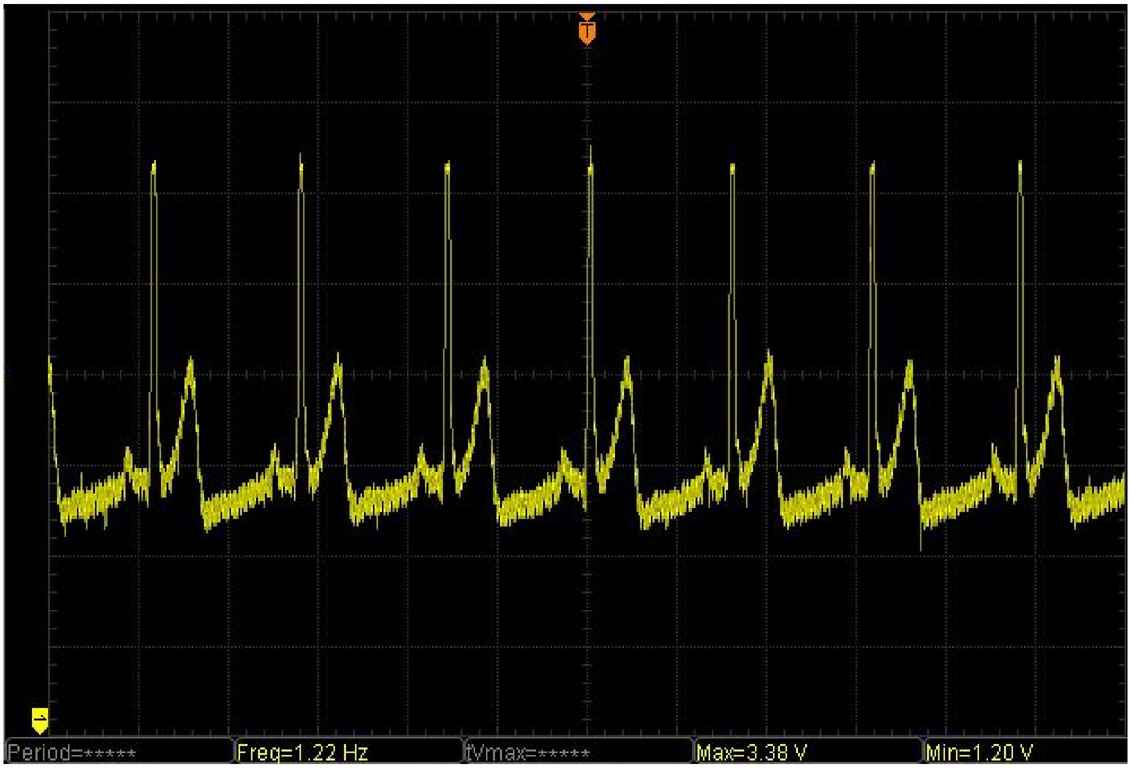

Figure 7 - ECG signal from the pre-filter board

VI. Parts List

Table 4 lists the parts used in the project.

Table 4 - Part list

Part	Vendor	Part #	Quantity	Price	Total
TI TMDX5515EZDSP	Digikey	296-25752-ND	1	$82.03	$82.03
3.5mm TRS male plug	Digikey	CP-3502-ND	3	$2.16	$6.48
Adafruit RGB Negative 16x2 LCD+Keypad Kit for Raspberry Pi	Adafruit	1110	1	$24.95	$24.95
Pi-EzConnect Terminal Block Breakout HAT	Adafruit	2711	1	$19.95	$19.95
Raspberry Pi 3	Amazon	B01CD5VC92	1	$36.91	$36.91
SanDisk 8GB Micro SD	Amazon	B008HK25BC	1	$9.99	$9.99
3M Red Dot Monitoring Electrode - Model 2560 - Bag of 50	Amazon	B0015TI4G2	1	$13.30	$13.30
Single Lead AD8232 Heart Rate Monitor ECG Development Kit	Amazon	B01LZK4479	2	$12.00	$24.00
SHIELD-EKG-EMG-PRO Cable for Gel ECG Electrodes	Amazon	B00K9487UC	2	$19.02	$38.04
				Total:	**$255.65**

VII. Project Management

Project management is essential for the success of any project. Outlined below are the division of labor and project schedule.

1. Division of Labor

It is important to determine each team member's role to ensure that all aspects of the project are addressed adequately. This gives the group direction, and allows each member to specialize and focus on a particular set of systems, leading to a successful project.

Table 5 - Division of Labor

Team Member	**Role**
Calvin Walden	● Linux implementation, including ○ DSP interface daemon ○ Wireless and serial communication daemons ○ User interface ○ Data management and logging
Edward Sandor	● Signal processing algorithm ● DSP implementation
Nicholas Clark	● Sensors board development ● Software implementations

2. Project Schedule

The project schedule has been broken down into weekly tasks for the group. This outlines the tasks from the present day to final presentations in April of 2017.

Table 6 - Project Schedule

Week Of	**Work To Be Completed**
11/21/16	● MATLAB simulation of Pan-Tompkins algorithm
11/28/16	● Project Proposal Presentation (12/1) ● Determine complexity of Pan-Tompkins algorithm

12/5/16	• Project Proposal (12/5) • Receive ordered parts
Winter Break	• Begin software development for embedded computer and DSP
1/16/17	• Implement initial Pan-Tompkins algorithm on choice platform • Begin interfacing board for embedded computer • Interface embedded computer with DSP
1/23/17	• Refine Pan-Tompkins algorithm, add Template Matching algorithm • Trial wireless communication and SMS
1/30/17	• Trial ECG data from MIT-BIH database • Refine wireless communication and SMS
2/6/17	• Refine Pan-Tompkins and Template Matching algorithms • Trial ECG data from MIT-BIH database • Begin embedded computer serial communication
2/13/17	• Begin real-time ECG testing • Add LCD and pushbutton interface to interfacing board • Begin UI development
2/20/17	• Refine Pan-Tompkins and Template Matching algorithms • Continue real-time ECG testing • Continue UI development
2/27/17	• Complete real-time ECG testing • Complete UI development
3/6/17	• Continue system testing and tuning
Spring Break	• Continue system testing and tuning
3/20/17	• Complete all lab work • Continue written deliverables
3/27/17	• Continue written deliverables • Update Student Scholarship Expo poster
4/3/17	• Finalize final report draft
4/10/17	• Finalize draft of presentation slides • Finalize Student Scholarship Expo poster (4/10) • Event: Student Scholarship Expo (4/11)
4/17/17	• Complete all written deliverables

4/24/17	• Event: Presentation of project • Event: Project demonstration

VIII. References

[1] "About Arrhythmia." [Online]. Available: http://www.heart.org/HEARTORG/Conditions/Arrhythmia/AboutArrhythmia/About-Arrhythmia_UCM_002010_Article.jsp.

[2] "Why Arrhythmia Matters." [Online]. Available: http://www.heart.org/HEARTORG/Conditions/Arrhythmia/WhyArrhythmiaMatters/Why-Arrhythmia-Matters_UCM_002023_Article.jsp.

[3] M. AlGhatrif and J. Lindsay, "A brief review: history to understand fundamentals of electrocardiography," J Community Hosp Intern Med Perspect, vol. 2, no. 1, Apr. 2012.

[4] J. A. Z. Justo, R. A. G. Calleja, and A. M. Diosdado, "Acquisition software development for monitor Holter prototype signals and its use for pre-diagnosis of cardiac damage based on nonlinear dynamic techniques," in AIP Conference Proceedings, 2016, vol. 1747, p. 90001.

[5] F. Bamarouf, C. Crandell, and S. Tsuyuki, "Real-time heart monitoring and ECG signal processing," Bradley University, May 2016.

[6] E. Bailey, J. Bolkhovsky, J. Sorrentino, and L. Anderson, "Automatic detection of atrial fibrillation and atrial flutter," Worcester Polytechnic Institute, Apr. 2012.

[7] "Premature Ventricular Contractions PVC - Cardiology - Highland Hospital - University of Rochester Medical Center." [Online]. Available: https://www.urmc.rochester.edu/highland/departments-centers/cardiology/conditions/premature-ventricular-contractions.aspx.

[8] H. Khamis, R. Weiss, Y. Xie, C. W. Chang, N. H. Lovell, and S. J. Redmond, "QRS detection algorithm for telehealth electrocardiogram recordings," IEEE Transactions on Biomedical Engineering, vol. 63, no. 7, pp. 1377–1388, Jul. 2016.

[9] N. M. Arzeno, Z.-D. Deng, and C.-S. Poon, "Analysis of first-derivative based QRS detection algorithms," IEEE Trans Biomed Eng, vol. 55, no. 2, pp. 478–484, Feb. 2008.