

# Systematic Design and HRV Analysis of a Portable ECG System Using Arduino and LabVIEW for Biomedical Engineering Training

Mehdi Shokouinejad, Michelle Chiang, Samuel Lines, Fa Wang, Willis Tompkins, and John G. Webster  
Department of Biomedical Engineering, University of Wisconsin-Madison, Madison WI 53706, USA  
Email: john.webster@wisc.edu, mehdi@bme.wisc.edu

**Abstract**—This paper presents a systematic design for a fully operational portable electrocardiogram (ECG) system for training and educational purposes in Biomedical Engineering (BME) curriculum courses. The proposed ECG system also includes heart rate variability (HRV) analysis using LabVIEW, as well as step-by-step methods for the hardware and software design consideration. The instrumentation is developed to interface as a shield for an Arduino Uno. Furthermore, the study of the ECG signal includes acquisition of a real-time ECG signal, signal filtering and processing. ECG feature extraction is based on the wavelet transform for HRV analysis in LabVIEW. The proposed system has been tested with the MIT-BIH arrhythmia database and was able to achieve a sensitivity of 93.5%. Researchers will develop a valuable toolkit in the process of designing this system, and if the system is further developed to meet clinical standards, it could have major applications in less accessible areas.

**Index Terms**—ECG, physiological measurement system design, ECG signal processing, LabVIEW, Arduino, engineering education

## I. INTRODUCTION

Cardiovascular Disease (CVD) is the number one cause of death each year [1], [2]. Electrocardiography (ECG) plays a central role in the diagnosis of cardiovascular disease by reading the electrical activity of the heart in a non-invasive manner [3], [4]. ECG information is used for a variety of purposes, such as diagnosis of heart disease, monitoring of nominal heart activity, and other recording applications.

The ECG is broadly used in clinics to examine patients' heart conditions, and its effectiveness has been proven through its extensive use in the diagnosis of heart conditions.

The proposed system is a fully operational portable electrocardiogram (ECG) system for training and educational purposes in Biomedical Engineering (BME) curriculum courses.

Incorporating ECG design into curriculum is largely beneficial for students and researchers interested in BME. From a biological standpoint, they will learn to understand the electrical nature of arguably the most important physiological organ. The electrical signals of

the heart, gathered from the skin, are illustrative in determining the condition of the heart. If a patient is suffering from abnormal heart pathology, the electrical signals generated by the patient are likely to differ from those of a normal individual. Therefore, studying the electrical signal generated by the heart can provide a useful means to diagnose patients non-invasively. From a technical standpoint, students and researchers will develop a useful toolkit for BME design. They will gain important technical skills, such as how to acquire a real-time, biological signal and how to use hardware and software to process the signal. They will learn how to design and fabricate a Printed Circuit Board (PCB), how to use a microcontroller to develop a portable medical device system, and how to use LabVIEW for important ECG feature extraction.

Incorporating ECG design into BME curriculum can also be beneficial for society. This is because the development of accessible, portable ECG systems has real clinical applications worldwide. A portable ECG system would provide a handy ambulatory monitoring system that patients can carry around and have their heart conditions continuously monitored. Cardiovascular disease is an extremely prevalent issue, ECG design is an extremely relevant field of study, and the optimization of the proposed system could contribute to mitigating the global issue of heart disease mortality.

The purpose for this study is to create step-by-step platform that equips students and researchers to develop physiological signal acquisition systems. The proposed use for this platform includes various BME teaching and research applications, such as medical instrumentation courses and undergraduate or graduate independent research. An overall design specification is explained and adopted in Section II.A. The hardware design is provided in Sections II.B and II.C. The software implementation is then outlined in Sections II.D and II.E. The experiment results and statistics for the developed device are presented in Section III. Finally, Section IV provides further discussion of this study.

## II. MATERIALS AND METHODS

### A. Overall Design Specifications

We designed a two-channel ECG amplifier to allow simultaneous ECG data recording for both lead I and

lead II. The bandwidth was set to 0.05 to 150 Hz with an anti-aliasing filter of greater than  $-100$  dB/decade for the low-pass filter [5]. The range of frequency for measurement of this physiological measurement is at low frequency in contrast to some other physiological circuitry application which can be in high frequencies [6]-[8]. The portable ECG was constructed as a daughter board for the Arduino Uno microcontroller, which was programmed to sample the signal at 750 Hz [9]. All analog circuitry was placed on a single dual layer Printed Circuit Board (PCB) for shielding purposes. The analog data can then be processed using a combination of LabVIEW and Arduino based coding algorithms.

The Arduino takes care of the front end data collection and correction. The digitized ECG signals are sent to the computer for processing. LabVIEW handles the processing and extrapolation of denoised ECG data for varying degrees of measurement and interpretation. The ECG feature extraction was based on the Daubechies wavelet (db06) transform for its similarity to the QRS wave of the ECG signal. After the wavelet transform, the algorithm searched for the maximum positive point and minimum negative point pair using the high frequency scale [10], [11], which determined the R wave positions. A search window was implemented to determine the P wave and T wave at low frequencies. Based on the Pan-Tompkins algorithm and ECG feature extraction R-R intervals, P wave, QRS complex were determined, and used in the LabVIEW's HRV analysis suite. Using ten sample ECG sets from the MIT-BIH arrhythmia database, results of the LabVIEW HRV analysis were collected and compared to information posted on the MIT-BIH database.

### B. Hardware - Analog Circuit Design

The analog circuitry consists of four primary signal processing blocks that allow for amplification and filtration of the raw ECG signal to meet the system specification listed in Table I. The stages in order of signal processing are a preamplifier, Band-Pass Filter (BPF), low-pass filter and Programmable Gain Amplifier (PGA). The complementary circuits used to aid in functionality are the Analog Common Voltage (ACOM) generator and the Driven Right Leg (DRL) circuit. Power is supplied to each of these stages by the output pins on

the microcontroller that connect to the PCB. The microcontroller itself is powered by a laptop via the USB power cable. The separation of the collective system into these subsections allows for specifications to be verified during the development of the system. The relationship between these circuit topologies is demonstrated in Fig. 1.

TABLE I. PORTABLE ECG SYSTEM SPECIFICATION AND FEATURES

| 1. Amplifier specifications:   |  |
|--|--|
| Differential Gain  | 432–1728 V/V                             |
| –3 dB Bandwidth  | 0.05–150 Hz                              |
| Anti-aliasing filter   | $\geq -100$ dB/decade                    |
| Input $V_{cm}$   | $\sim 0$ V                               |
| Output to be referenced in the center of the non-saturating range  | ( $\sim 2.5$ V)                          |
| Power Supplies   | +5 V, GND from laptop                    |
| Driven Right Leg implementation  | Used to reduce Common Mode Interference. |
| 2. Microcontroller with on-board analog to digital converter (ADC) 10-bit ADC, successive approximation (SAR-type) |  |
| Sampling rate ( $F_s$ )  | 750 Hz                                   |
| $V_{ds}$   | +5 V <sub>dc</sub>                       |
| 3. Features  |  |
| a. Real-time display of Lead I, Lead II, Lead III, aVR, aVL and aVF ECG signals on computer                        |  |
| b. Implement QRS detection algorithm and Heart rate average over 10 beats and refreshed after every beat           |  |
| c. HRV Analysis  |  |

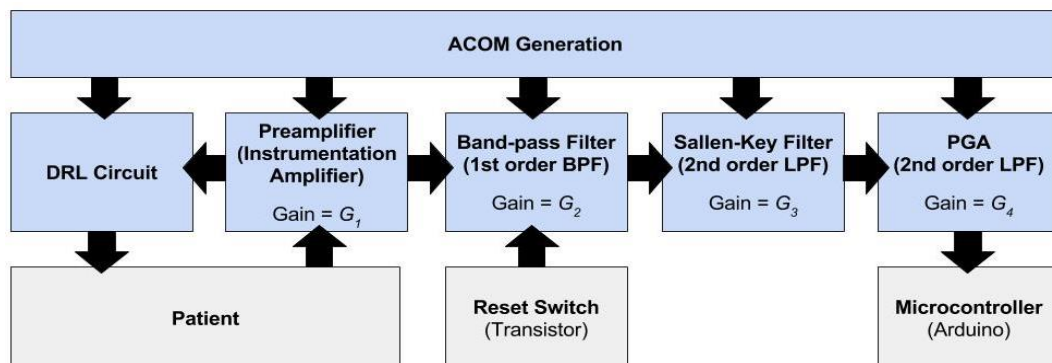


Figure 1. Block diagram of the ECG lead amplifier.

The gain of the whole system is equal to the product of the gains of each circuit block, as demonstrated in (1).

$$G_T = G_1 * G_2 * G_3 * G_4 \quad (1)$$

In order to meet the specifications set in Table I, each stage was designed to have a certain gain so that the product of all stages yielded a total gain,  $G_T$ , equal to 432-1728 V/V. Because the Sallen-Key filter is a unity gain filter, it does not change the gain of the system; therefore, the total gain is dependent only on the gain of the preamplifier, band-pass filter, and PGA. Here, reasonable gains were set for the first two blocks, and the gain for the fourth block (PGA) was determined accordingly.

#### 1) Analog Common Voltage (ACOM) generator

The ACOM generator is a power block that will supply 2.5 V elsewhere in the circuit. This circuit topology includes a voltage divider with two equivalent resistors to divide the 5 V power source in half. The resulting 2.5 V signal is isolated using a voltage follower buffer. This will provide the 2.5 V baseline for the ECG. The capacitance within this pseudo-ground circuit is added for stability of the DC output.

#### 2) Preamplifier stage (instrumentation amplifier)

The purpose of this stage is to provide both high input impedance and a high common mode rejection ratio, which helps to improve the signal-to-noise ratio of the system as a whole. The main component is the LT1167 instrumentation amplifier. This amplifier operates with power rails of 0 V and 5 V called as a single supply. Since the range is non-negative the signal is centered on the midpoint voltage,  $V_{\text{offset}} = 2.5$  V. This offset is applied using the ACOM generator circuit. To acquire proper resistance values for the gain of the preamplifier at this stage, (2) was utilized.

$$G_1 = 1 + \frac{49.4 \text{ k}\Omega}{R_G} \quad (2)$$

In order to achieve a gain of 10.96, the resistance,  $R_G = 4.96 \text{ k}\Omega$ , is found by applying the formula. This resistance must then be split into two equal parts. The resulting two equivalent resistors are used calculate the average of the input signal, which then act as an input to the DRL circuitry (see Section II.B-6)).

#### 3) Band-Pass stage

This stage starts with a high-pass filter input that has a cutoff frequency of 0.05 Hz. This cutoff frequency is set by the values of the resistor and capacitor as shown in (3).

$$f_c = \frac{1}{2\pi\tau} = \frac{1}{2\pi RC} \quad (3)$$

Picking the capacitor to be 10 uF and applying (3) yields a resistance value of roughly 330 k $\Omega$ . With these specifications, if the circuit is saturated, there will be a time delay for the DC baseline to return to the equilibrium baseline, 2.5 V. This is a result of the high time constant of the high-pass portion of the band-pass filter. To eliminate this effect, a reset switch is added to discharge the capacitor and drive the output back to

$V_{\text{offset}}$ . This is done by using a BJT connected to the ACOM 2.5 V DC source at the emitter junction, and the source collector connected to the side of the capacitor that is closest to the amplifier within that block. This is activated by applying a voltage from the Arduino to the base in order to forward bias the transistor (see Section II.D-1)).

Within the feedback loop of this active filter is a low-pass filter. Since there is a total of five low-pass filters within the system, (3) is no longer applicable to the frequency calculation. To acquire a steeper roll-off for the low-pass filter,  $n$  filter stages can be connected in series. The resulting transfer function is:

$$G = \frac{1}{(1+\alpha_1*s)(1+\alpha_2*s)...(1+\alpha_n*s)} \quad (4)$$

In the case that all filters have the same cut-off frequency,  $f_c$ , the coefficients become:

$$\alpha_1 = \alpha_2 = \dots = \alpha_n = \alpha = \sqrt[n]{\sqrt{2} - 1} \quad (5)$$

As a result,  $f_c$  of each partial filter is  $1/\alpha$  times higher than  $f_c$  of the overall filter [12]. Provided this information, the case of five series low-pass filters, (3) must be multiplied by a correction factor, as shown below.

$$f_c = \frac{\sqrt[n]{\sqrt{2}-1}}{2\pi\tau} = \frac{\sqrt[n]{\sqrt{2}-1}}{2\pi RC} \quad (6)$$

This new formula gives a desired cutoff frequency of 150 Hz when a time constant  $\tau = RC$  of  $4.09*10^{-4}$  is used. This means that for each low-pass filter within the system, the product of the resistance and capacitance must equal this value. To maximize gain within the band-pass stage, the smallest capacitor value (330 pF) is used to allow for a maximized feedback resistance. The corresponding resistance is calculated to be 1.2 M $\Omega$  to meet all constraints. The gain of this filter,  $G_2$ , is then equal to the feedback resistance divided by the input resistance, which equates to -3.64.

#### 4) Low-pass stage

The low-pass topology that is employed is called Sallen-Key. It is a second-order filter and has unity gain characteristics. As long as the resistor values and capacitor values are chosen with the same tau value as previously calculated (see Section II.B-3)), the filter will have the correct frequency response. Due to the unity gain characteristic, all components will equate to each other ( $R14$  will be equivalent to  $R15$  and  $C8$  will be equivalent to  $C9$ ). The value of  $C$  is chosen to be 330 nF, which leads to an  $R$ -value of 1.24 k $\Omega$  when the tau equivalency is used.

#### 5) Programmable Gain Amplifier (PGA) stage

This stage consists of a low-pass filter leading into a non-inverting operational amplifier and is immediately followed by another low-pass filter. The reasoning behind the programmable aspect of this stage is for compensation due to variable electrical activity of the heart. Since we want to be able to scale the output signal to a visible level, the gain must be adjusted. The values

of the resistors and capacitors in the low-pass filters were picked to be 330 nF and 1.24 kΩ, just as above. To find the gain of the non-inverting operational amplifier, the following (7) is used:

$$G_4 = 1 + \frac{R_f}{R_i} \quad (7)$$

In this equation,  $R_f$  is the total resistance in the feedback loop and  $R_i$  is the input resistance. This block is called a programmable gain amplifier because  $R_f$  contains a 100 kΩ potentiometer, which can be manipulated. The DS1804 NV, nonvolatile digital potentiometer, is utilized and has 100 unique positions. In this case, these values range from  $R_{varL}$ , 1 kΩ, to  $R_{varH}$ , 100kΩ, in increments of 1kΩ. The device provides an ideal method for low-cost trimming applications using a Central Processing Unit (CPU) or manual control input with minimal external circuitry (see Section II.D-2)). To achieve the required gain range, the system of equations (8)-(9) must be solved. Here,  $G_{TL}$  is the lower bound of the total gain and  $G_{TH}$  is the upper bound of the total gain, which are 432 and 1728 V/V, respectively.

$$G_{TL} = G_1 * G_2 * G_3 * \left(1 + \frac{R_{varL} + R_{f1}}{R_i}\right) \quad (8)$$

$$G_{TH} = G_1 * G_2 * G_3 * \left(1 + \frac{R_{varH} + R_{f1}}{R_i}\right) \quad (9)$$

In these equations, 39.89 is the product of the individual gains of the previous blocks,  $R_{f1}$  is a constant resistor in the feedback loop, and  $R_i$  is the input resistor. Solving this system and picking the closest resistor values available gives an  $R_{f1}$  of 26.7 kΩ and a  $R_i$  of 6.8 kΩ.

The parametric simulation in Fig. 2 shows the frequency response for the PGA system at different gain settings. The bottommost trace represents the minimum possible gain for the system, when  $R_{var} = 1$  kΩ. The topmost trace represents the maximum possible gain for the system, when  $R_{var} = 100$  kΩ. The middle trace depicts the frequency response when  $R_{var} = 10$  kΩ to further demonstrate the relationship between the variable resistor and the gain. The key here is that all three traces demonstrate comparable frequency responses. Thus, the major advantage of the PGA is that the user can have manipulation over the output gain without altering other behaviors of the system in order to cover the wide range of the amplification.

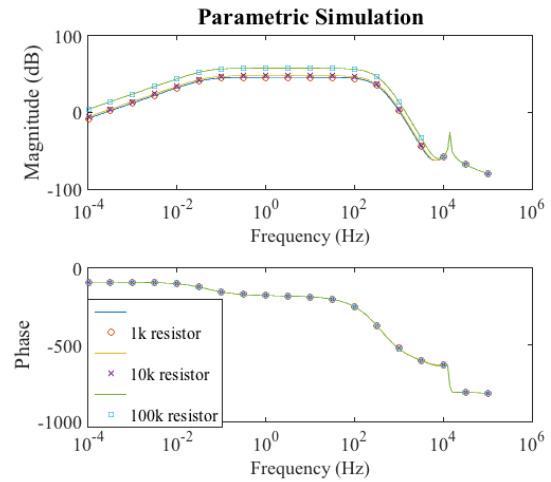


Figure 2. Bode plot of the simulated output of the ECG lead at minimum gain, maximum gain, and one gain in between minimum and maximum.

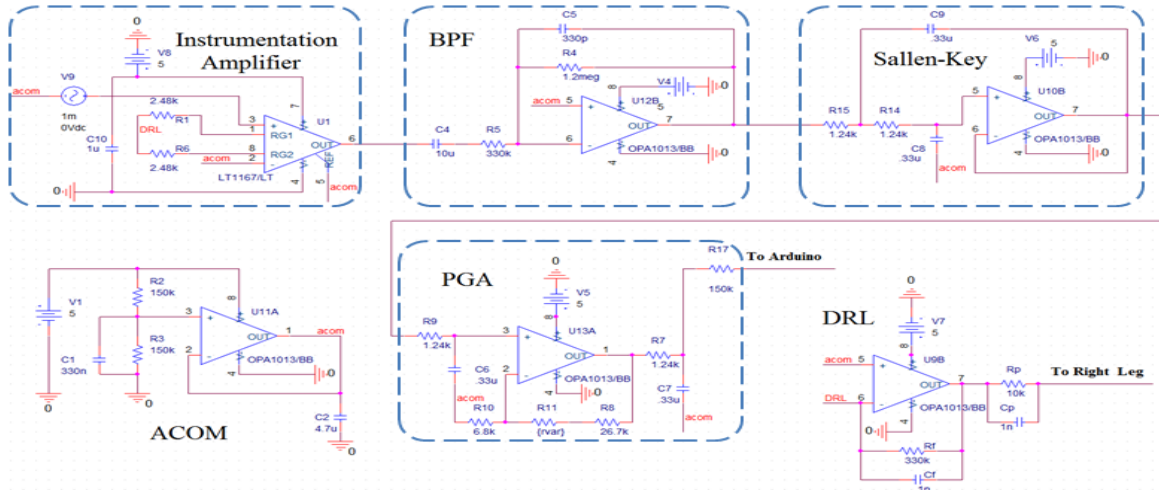


Figure 3. Circuit diagram of a single ECG lead. Two identical circuits were combined to create the ECG shield. Regarding the operational amplifiers, an LT1167 was used for the instrumentation amplifier and the TLV274 was used for all other op-amps. When implementing the transistor that serves as a reset switch, it is attached to the capacitor, C4, on the band-pass filter circuit block (see Section II.B-3)).

#### 6) Driven Right Leg (DRL) circuit

One of the most important performance parameters for ECG system applications is the Common Mode Rejection (CMR) [13]. This is because the differential between the two raw input voltages is very small. Driving common mode voltage through a resistor is one technique to improve CMR [13]. Thus, the driven right

leg circuit is added to the design to further minimize common mode voltage [14].

In this case, a low resistance,  $R_p$  (see Fig. 3), is used to drive the common mode voltage into the patient's body. Initially, the body is at a positive common mode voltage, so the DRL inverts and amplifies the signal from between the two equivalent resistors in the preamplifier

(see Section II.B-2)). The amplified negative signal is then fed back into the patient's right leg, which pulls down the common mode voltage, thereby increasing the CMRR.

The closed loop gain of this system,  $G_5$ , can be calculated using (10), where  $Z_f$  is the total feedback impedance (11).

$$G_5 = \frac{Z_f}{Z_i} = \frac{Z_f}{R_G/2} = 2 * \frac{Z_f}{R_G} \quad (10)$$

$$Z_f = \frac{R_f}{1+sR_fC_f} \quad (11)$$

This feedback loop improves CMR by  $(1 + G_5)$  [10].

In summary, the DRL provides stability to the system, while ultimately improving CMR. If the third electrode is connected to a DRL, the effective resistance to common is reduced to  $R_p$ , which also serves as a current-limiting resistor [14]. The DRL circuit block is also used to help remove a significant amount of noise due to power-line interference, as well as to prevent any potentially dangerous currents from flowing through the patient [13], [14].

#### 7) Additional components

In addition to the primary blocks, several other components are added onto the PCB in order to transition the circuit from a breadboard prototype to a final shield version. The audio jack is added in order to connect the electrode leads from the body to the PCB. An additional screw terminal is placed on the PCB to account for the extra lead needed for the DRL. There are also a series of header pins that match the pinout of the Arduino so that the PCB can interface readily with the software (see Fig. 3). Several other header pins are also added to the schematic to either serve as access (or test) points or to create sockets for removable components.

#### C. CAD Drawing of ECG Circuit

The Printed Circuit Board (PCB) design is based on the schematic developed from the ECG circuit hardware described in Section II.B. This PCB is designed and constructed in Eagle, a PCB design freeware. The process of exporting the schematic to Eagle involves recreating each component and connection on the Cadence PSPICE schematic. Then the entire single channel design gets repeated to create a dual channel ECG circuit. However, only a single DRL channel is used to negate interference from the measurements.

Each component placed on the Eagle schematic is required to have a specified footprint that identifies its physical form on the PCB. On this board file, the connections from the schematic are preserved. Each component is placed on a 2.3 in x 3 in (1 in = 0.0254 m) board and connected via copper traces, following the connections on the schematic. Trace widths vary from 8 mil to 32 mil (1 mil =  $2.54 \times 10^{-5}$  m) depending on the type of connection. Power or signal connections are generally given larger traces to reduce noise and ensure solid connectivity. For power and ground, separate

copper planes, called polygons, create a similar phenomenon to ensure power signal integrity (see Fig. 4).

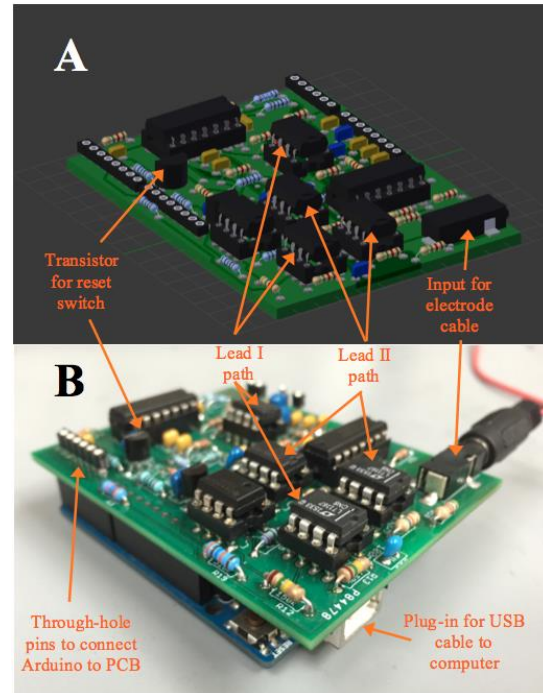


Figure 4. (a) Conceptual illustration of electronics package suitable for onboard testing in an Arduino UNO platform. (b) PCB layout of the ECG shield. PCB was a two-layer board.

#### D. Software – Arduino

The software for this design consists of two parts: the firmware and the user interface. The firmware is the code on the microcontroller that acts independent of the user. In this design, the firmware contains code that is used to acquire data from the PCB and coordinate communication with the computer. The computer runs a main application that serves as a user interface to display the data that has been gathered.

##### 1) Arduino firmware

The firmware code is implemented on a microcontroller. The Arduino Uno R3 was selected for this project due to its availability, wide community support and its efficiency in high-speed prototyping [6]. Fig. 5 demonstrates the Arduino firmware implementation flowchart. The firmware on the Arduino is used to gather data, maintain communication with the computer, and autonomously control various aspects of the ECG leads. The implementation and description of the Arduino code can be found in the flowchart below, which includes both automatic gain and defibrillation desaturation control. These abstract blocks will be discussed in more detail in Sections II.D-2) and II.D-3).

The flowchart details the implementation of the Arduino firmware. Upon startup, the Arduino initializes serial communication with the computer and LabVIEW Graphical User Interface (GUI) as well as a timer interrupt that will be used to set the sampling rate at 750 Hz to adhere to the Nyquist-Shannon sampling theory [15], [16]. After initialization, there is an infinite loop that allows for data acquisition as well as control of the



ECG lead circuit board. Every timer interrupt, data is sampled. After the data is sampled, the code checks to see if it is time to auto-calibrate the ECG leads. Finally, there is a check to see if either lead is saturated. If neither one has saturated, the sampled ECG lead data is transmitted to the computer.

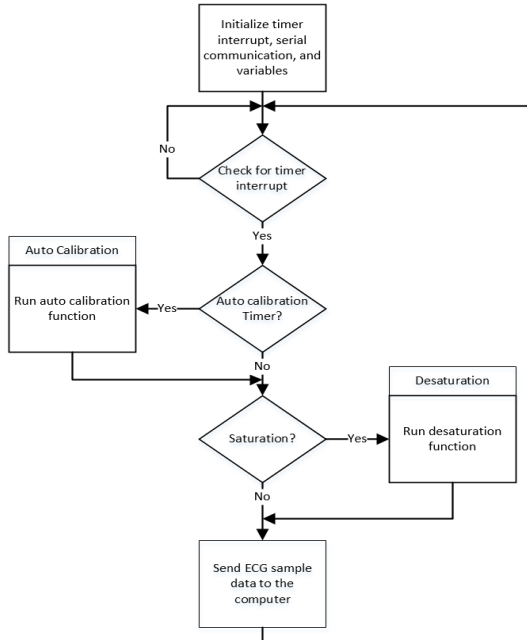


Figure 5. Flowchart for the Arduino firmware.

The firmware on the Arduino is very simple as it mainly provides a platform to easily send the data to the computer. It does so by using the Circular Byte Buffer implementation for Arduino, which is the memory efficient way to read and write data from an array-backed circular buffer [17]. That being said, both of the main automatic features of the Arduino, that of the gain control and desaturation switch, have been described below as algorithms to allow for straightforward implementation.

## 2) Automatic gain calibration

Automatic gain calibration of the ECG system allows the Arduino to control the gain of the amplifiers in real time to allow for the best signal acquisition. The algorithm for automatic gain calibration can be seen in Algorithm 1.

This algorithm works by comparing the peaks of the ECG QRS single to a threshold value, and determining whether the ECG lead gain should be increased or decreased. By comparing the peak ECG QRS signal to the threshold value, it is possible to determine the exact gain needed to reach the desired ECG amplification. Once this extra gain has been determined, the current gain of the PGA must be determined (see Section II.B-5)). To do this, (7) is modified by calculating  $R_f$  as the series sum of the two resistors that make up the effective  $R_f$  and then plugging in the actual resistances for both  $R_f$  and  $R_i$ . Since  $R_f$  is made up of a resistor and a digital potentiometer, the resistance of the digital potentiometer must be calculated by multiplying its current wiper position by the step in resistance from moving one wiper

position. With the calculation of the current gain of the PGA stage determined, it is possible to find the new wiper position that will lead to the desired ECG lead amplification. By rearranging the formula used to calculate the current gain of the PGA, and multiplying it by the necessary amplification calculated in the first step, the new digital potentiometer position is determined. With the new wiper position known, it is compared to the previous wiper position to determine how far it needs to move. If the wiper position needs to move in the negative direction,  $loopRun_i$  will be negative. To overcome the negative number of for loop iterations, the absolute value of  $loopRun_i$  should be used and the wiper direction should be changed from up to down.

### Algorithm 1: Implementation of Automatic Gain Calibration

**Define:**  $x_i$  to be the average of the last five samples of  $ECG_i$ ,  $M_i$  to be the max of  $x_i$  since last calibration, **threshold<sub>i</sub>** to be the desired level of the QRS complex of  $ECG_i$ , and **dPot<sub>i</sub>** to be the position of the digital potentiometer in the programmable amplifier of  $ECG_i$ .  
**Data:**  $x_i$ ,  $M_i$ , **threshold<sub>i</sub>**, **dPot<sub>i</sub>**  
**Results:** **dPot<sub>i</sub>**  
**for each ECG do**  
    **if**  $M_i \neq 5$  **then**  
        calibrate  $ECG_i$   
    **end if**  
     $amp = threshold_i / M_i$   
     $currGain_i = 1 + (dPot_i * 1000 + 26700) / 6800$   
     $newDpot_i = ((Amp_i * currGain_i + 1) * 6800) - 26700 / 1000$   
     $loopRun_i = newDpot - dPot_i$   
    **for loopRun**  
        increment or decrement **dPot<sub>i</sub>**  
    **end for**  
**end do**

### 3) Desaturation function

The algorithm for saturation check is simple yet important because saturation of the ECG will prevent the proper use of the ECG, Algorithm 2.

### Algorithm 2: Implementation of desaturation Function

**Define:**  $A_i$  to be the array of the last 75 samples of  $ECG_i$  and **satThresh<sub>i</sub>** to be the saturation threshold for  $ECG_i$   
**Data:**  $A_i$   
**Results:** true if saturated else false  
**for each ECG do**  
     $avg_i = average(A_i)$   
    **if**  $avg_i > satThresh_i$   
        engage desaturation method;  
    **end if**  
**end do**

In order to determine if saturation has occurred, a moving window of the previous 75 sampled data points is kept. This corresponds to the last 0.1 seconds of ECG data given the 750 Hz sampling rate. If the ECG is ever saturated for more than 0.1 seconds, the desaturation switch will be engaged to bring the ECG back down to baseline. In order to determine if saturation has occurred, the last 75 samples are averaged. If this average is above a threshold set to the previous max value, then it is said to have saturated and the switch will be activated.

### E. Software – LabVIEW

After data collection on the Arduino, the raw ECG data is transmitted to a LabVIEW executable that does all of the signal processing, lead calculations, heartbeat detection, and arrhythmia detection. This executable also displays all of the data to the user via a custom GUI. The GUI provides the user with an easy way to see the ECG data. This allows for determination of how the heart is beating and for visual recognition of any sort of potential pathologies. In Fig. 6, ECG lead I, II, and III are displayed on the left along with the augmented lead aVR, aVL, and aVF on the right.

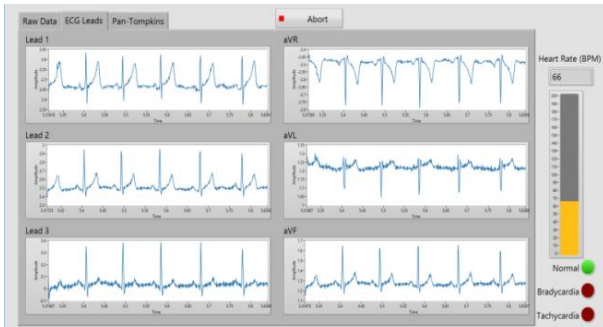


Figure 6. Front Panel of the LabVIEW ECG GUI. Displayed are ECG leads I, II, and III as well as aVR, aVF, and aVL.



Figure 7. Flowchart for the LabVIEW code.

Data processing was accomplished through various different built-in LabVIEW VIs and was used for signal denoising, feature extraction as well as lead calculation. Implementation of the LabVIEW code is outlined and described in the flowchart in Fig. 7.

#### 1) LabVIEW code implementation flowchart

Once the data was received from the Arduino, the raw data is denoised. This abstract block will be discussed in more detail in Sections II.E-2). Then, two parallel processes occurred in LabVIEW. The denoised ECG data was used to calculate the various leads using the Einthoven limb leads and Einthoven triangle [18]. At the same time, data from both lead I and lead II was used to detect the QRS complex of each beat. To do this, a modified version of the Pan-Tompkins peak detection algorithm was implemented. The implementation of this algorithm can be seen in Section II.E-3). Furthermore, Wavelet based ECG feature extraction was deployed in order to perceive the P and T wave position, which is discussed in more detail in Section II.E-4). After all of the calculations for the various lead have been made, the QRS complexes have been detected, and the Beats Per Minute (BPM), R-R interval, P-R interval, and R-T interval were calculated, all the data was displayed on the GUI for easy access.

#### 2) Wavelet based ECG signal denoising

After recording a certain period of time (e.g.: 1 minutes), before the further signal processing, raw data should be denoised. The noise in the ECG signal may come from 1) other physiological signals, like Electromyography (EMG) and respiration signal; 2) electrode contact noise; 3) motion artifacts. These noises could be modeled as white noise or color noise [19], which could be denoised based on Wavelet Transform (WT). The WT of the ECG signal  $f(x)$  is defined as (12):

$$W_s f(x) = f(x) * \Psi_s = \frac{1}{s} \int_{-\infty}^{\infty} f(t) \Psi\left(\frac{x-t}{s}\right) dt \quad (12)$$

Scale factor  $s$  determines the basis function. Corresponding coefficients like  $d_1$ ,  $d_2$ ,  $d_3$ , etc. give frequency domain information. WT with higher  $s$  value will focus on the low frequency domain, and vice versa. Prototype wavelet  $\Psi_s$  is chosen as Daubechies wavelet (db06) due to similarity of the shape to the QRS wave of ECG signal. Previous studies apply golobla thereshold for all levels of WT. On the other hand, Poornachandra et al. introduced a level dependent threshold for ECG denosing [20] which using this approach can enable us to achieve higher signal to noise ratio (SNR). The threshold is defined in (13):

$$t_{l,k} = \frac{\sigma_k \sqrt{2 \log(n)}}{S_{l,k} + b} \quad (13)$$

where  $\sigma_k$  is the variance of the noise. If  $\sigma_k$  is unknown, it could be estimated via median absolute deviation [21] shown in (14):

$$\sigma_k = \frac{\text{median}|x|}{0.6745} \quad (14)$$

$S_{l,k}$  in (13) is scale level dependent parameter (15):

$$S_{l,k} = 2^{(L-K)/L} \quad (15)$$

$L$  is a total or deepest decomposition wavelet transform level, which normally ranges from 3 to 5. And  $K$  is the current level which thresholding is applying on.

Based on  $t_{l,k}$  threshold, we apply soft threshold to the wavelet coefficient at each level (16):

$$\delta_t(x) = \text{sgn}(x)(|x| - t_{l,k}) \quad (16)$$

Then the ECG signal was reconstructed based on these modified coefficients shown in Fig. 8.

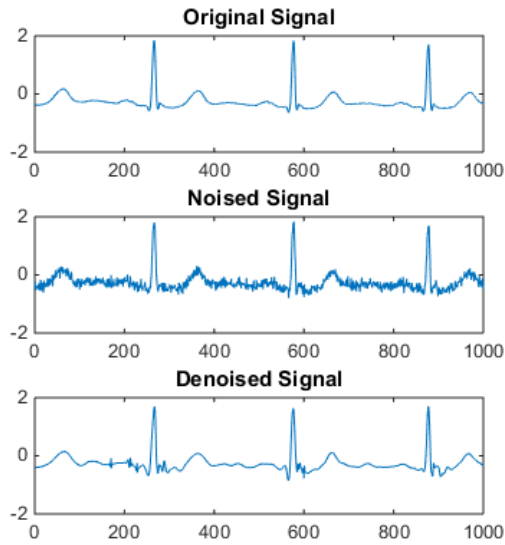


Figure 8. Demonstrates the signal denoise result. Denoising with db06 and level 5.

### 3) Pan-Tompkins algorithm in LabVIEW

A modified implementation of the Pan-Tompkins algorithm created with the use of built in LabVIEW point by point functions for real time QRS detection shown in Algorithm 3 [22], [23].

Algorithm 3: Pseudocode of Pan-Tompkins Algorithm

**Define:**  $E_i$  to be the data from ECG<sub>i</sub>

**Data:**  $E_i$

**Result:** R

**for a signal**  $E_i$  **do**

$B = \text{Band-pass filter}(E_i) \text{ low cutoff} = 5\text{Hz} \text{ high cutoff} = 17\text{Hz};$

$D = \text{Point-by-point derivative}(B);$

**if**  $D < 0$

$D = 0$

**end if**

$S = \text{Square}(B)$

$A = \text{avg}(S_k, S_{k-1}, S_{k-2}, S_{k-3}, S_{k-4})$

**end do**

Upon entering the Pan-Tompkins QRS detection Sub-VI, each data point passes through a band-pass filter

composed of a cascaded high-pass and low-pass filter. The cutoff frequency for the high-pass filter is 5 Hz and the cutoff frequency of the low-pass filter is 17 Hz. This creates a pass band from 5 to 17 Hz. After the band-pass filter, the point-by-point derivative of the signal is taken. Wired to the point-by-point derivative is the time between samples, which is dictated by the sampling rate of the Arduino. Next to further isolate the QRS complex values below zero are eliminated. This helps remove noise. Finally, the signal is squared and amplified to create peaks for each QRS complex. Fig. 9 shows the LabVIEW implementation of the algorithm described in Algorithm 3.

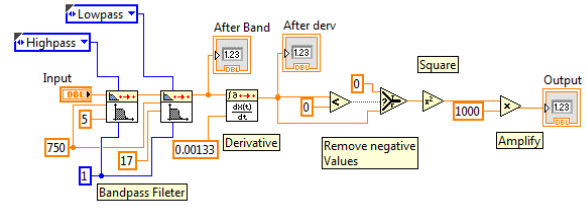


Figure 9. LabVIEW implementation of the modified Pan-Tompkins algorithm.

### 4) Heart rate calculation

Using the output of the Pan-Tompkins QRS detection algorithm, heart rate was calculated using last ten detected beats shown in Algorithm 4.

Algorithm 4: Pseudocode of Heart Rate Calculation

**Define:**  $P$  to be the output of the Pan-Tompkins algorithm,  $f$  to be a flag to determine if current peak is still occurring,  $HR$  to be previously calculated heart rate, and  $\text{thresh}$  to be the threshold used to signal a peak has been detected.

**Data:**  $P, f, HR, \text{thresh}$

**Results:** heartRate

**initialize**  $f = \text{false};$

**if**  $\sim f \ \&\& \ P > \text{thresh}$  **then**

$f = \text{true};$

$\text{beatArrat}[i] = \text{current time} - \text{previous beat time};$

$i = (i+1)\%10;$

$\text{heartRate} = \text{avg}(\text{beatArray}[] \text{ over all } i)$

**else**

$\text{heartRate} = HR;$

**end if**

**if**  $f \ \&\& \ P < \text{thresh}$  **then**

$f = \text{false};$

**end if**

Heart rate was determined using the output of the Pan-Tompkins Sub-VI. Each time a peak from the Pan-Tompkins goes over the threshold value set for the heart rate calculation, the time of the beat minus the time of the previous beat is stored in a circular array of length 10. This captures the time between each beat. The points from the array are then averaged to get the average time



between beats from the last ten beats. With the average time per beat, heart rate is easily back calculated.

#### 5) Wavelet based ECG feature extraction

The denoised signal is also can be sent to the feature extraction sub VI in LabVIEW for ECG feature extraction shown in Fig. 7, which is based on the discrete wavelet transform(DWT). Original signal, which is "s" in Fig. 10, will be decomposed by a high pass filter and a low pass filter. High pass filter result would be finally transformed to the level 1 detail coefficients d1. Low pass filter result, which is transformed to the approximation coefficients, would be treated as the next level original signal, and repeated to get level 2 detail coefficients d2. This progress would repeated 5 times for a typical ECG signal and get level 1 to level 5 detail coefficients d1 to d5. Remaining approximation coefficients is a5 and is also shown in Fig. 10.

Typical one cardiac cycle signal and its wavelet transform is shown in Fig. 10:

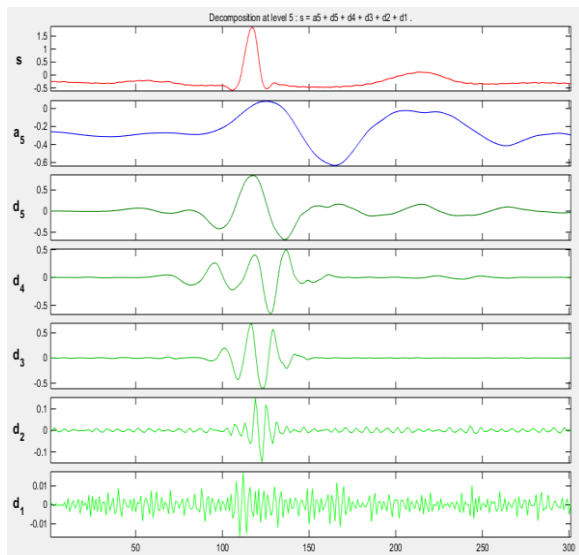


Figure 10. DWT result with db06 and level 5.

QRS complex is much clearer at high frequency scale like d2. After the WT, and rejecting all the isolated maximum points, algorithm tries to search for the local maximum ( $I_{max}$ ) positive point and local minimum ( $I_{min}$ ) negative point pair, which absolute value of ( $I_{max} - I_{min}$ ) is larger than preset threshold; the zero-crossing point between this pair is the position of the R wave [10]. If algorithm could not find this pair for more than 2 seconds by preset threshold, lower threshold will be chose to search back again.

Search windows for P wave detection which windows location is decided by QRS position and length is depending on the R-R interval are performed to search at low frequency scale, like d5. If one pair of local maximum positive point and local minimum negative point could be found and larger than preset threshold, a P wave can be detected. Similar approach is also applied to find the T wave shown in Fig. 11.

R-R interval, P-R interval, R-T interval could be found based on the position of QRS complex, T wave and P wave. Further Heart rate variability (HRV) analysis like

Root Mean Square of the Successive Differences (RMSSD) and NN50 which is the the number of pairs of successive NNs (same as R-R, normal beats) that differ by more than 50 ms could also be calculated based on this information.

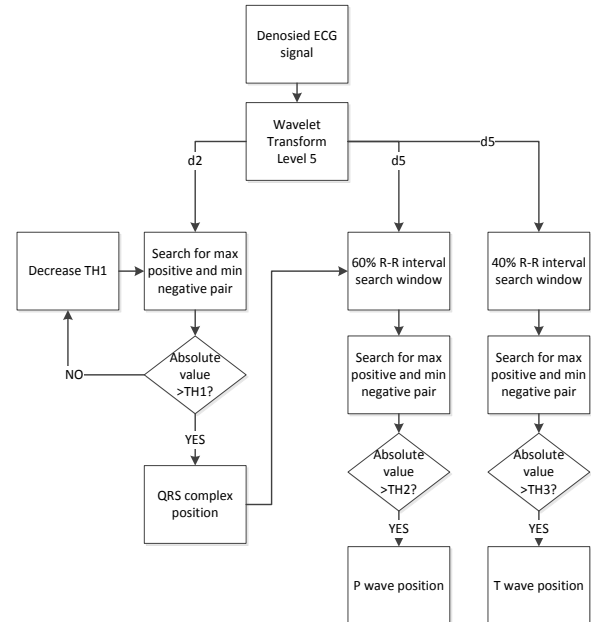


Figure 11. Demonstrates flowchart of Wavelet based ECG feature extraction.

### III. RESULTS

The LabVIEW algorithm was tested using 25 one-minute long ECG data sets from the MIT-BIH Arrhythmia Database. Using this database, detected beats were compared to the annotated information with each ECG signal. To maintain consistency throughout the tests, ECG beat detection was done on data from lead II. The algorithm that was implemented in this project was able to achieve a sensitivity of 93.5%. The percentage of correctly classified heartbeats from the LabVIEW HRV analysis varied from 89% to 100% with an average of 97.46%. The average error rate for the detected RR interval was 0.12% The RMSSD error rate was 1.69%. The NN50 error rate was 1.95%.

### IV. DISCUSSION AND CONCLUSION

Here, the systematic design for a fully operational ECG system has been described in detail. The skills acquired from a project like this can be really useful for experimental or lab purposes in biomedical engineering curriculum.

The authors are aware that the ADS1298 from Texas Instruments exists, which is an integrated circuit (IC) chip that can be used in the manufacturing of ECG systems. For advanced scholars that can afford the chip and to whom this product is available, this could be incorporated in the design; however, in order to make this design more reproducible to engineers worldwide, the steps in this paper are available. Understanding the ECG design at a less abstract and more fundamental level makes this type of design more robust and adaptive

to other applications. The authors believe that this paper can provide helpful information and will be useful for training purposes without using complex ICs.

LabVIEW is graphical programming software that is widely used in industry as well as academia. Understanding how to implement different algorithms and utilize the available toolboxes can equip students to take on many different projects and prepare them for their careers [24]. Additionally, the Arduino Uno was selected for use in this project due to its availability, wide community support, and its efficiency in high-speed prototyping. Furthermore, the impact of this design has potential to span beyond academic education at the college and research level.

For instance, the authors hope that by providing this toolkit, and designing it based on the widely accessible material, that it would encourage students K-12 to pursue BME and the study of the physiological signals [24], [25]. Finally, training biomedical engineers across the globe to understand the basics of how to acquire and study the ECG signal may be the key to developing cardiac diagnostic equipment or new signal processing approaches in their future career that can help combat the issues of CVD and other disorder such as sleep apnea mortality rates worldwide [26]-[28].

#### ACKNOWLEDGMENT

The authors would like to thank Dr. Amit Nimunkar, Keum San Chun, Samuel Jensen, Shawn Patel, Alex Nguyen, Adib Keikhosravi, Lucas Hurtley, Gustavo Zach Vargas, and Robert Scott Carson, for guidance in the early design phases of the project. Funding for this project was provided by the Department of Biomedical Engineering, University of Wisconsin-Madison.

#### CONFLICTS OF INTEREST

The authors declare no conflict of interest.

#### REFERENCES

- [1] J. Xu, K. D. Kochanek, S. L. Murphy, and B. Tejada-Vera, "Deaths: Final data for 2007," *National vital statistics reports: From the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System*, vol. 58, pp. 1-19, 2010.
- [2] T. Yamashita and S. R. Kunkel, "The association between heart disease mortality and geographic access to hospitals: County level comparisons in Ohio, USA," *Social Science & Medicine*, vol. 70, pp. 1211-1218, 2010.
- [3] J. P. Clarys, "Electromyography in sports and occupational settings: an update of its limits and possibilities," *Ergonomics*, vol. 43, pp. 1750-1762, 2000.
- [4] M. Lamontagne, "Application of electromyography in sport medicine," in *Rehabilitation of Sports Injuries*, Springer, 2001, pp. 31-42.
- [5] N. Meziane, S. Yang, M. Shokouinejad, J. Webster, M. Attari, and H. Eren, "Simultaneous comparison of 1 gel with 4 dry electrode types for electrocardiography," *Physiological Measurement*, vol. 36, p. 513, 2015.
- [6] F. Wang, X. Zhang, M. Shokouinejad, B. J. Iskandar, J. E. Medow, and J. G. Webster, "A novel intracranial pressure readout circuit for passive wireless LC sensor," *IEEE Transactions on Biomedical Circuits and Systems*, pp. 1-10, 2017.
- [7] Z. Xuan, et al., "Invasive and noninvasive means of measuring intracranial pressure: A review," *Physiological Measurement*, vol. 38, p. R143, 2017.
- [8] M. S. Maragheh, S. R. Sandock, A. Kumar, and Y. H. Jung, "Smart patch for wound management," USA Patent, 2015.
- [9] A. D'Ausilio, "Arduino: A low-cost multipurpose lab equipment," *Behavior Research Methods*, vol. 44, pp. 305-313, 2012.
- [10] C. Li, C. Zheng, and C. Tai, "Detection of ECG characteristic points using wavelet transforms," *IEEE Transactions on biomedical Engineering*, vol. 42, pp. 21-28, 1995.
- [11] M. Shokouinejad, et al., "Sleep apnea: A review of diagnostic sensors, algorithms, and therapies," *Physiological Measurement*, vol. 38, p. R204, 2017.
- [12] T. Kugelstadt, "Active filter design techniques," *Op Amps for Everyone: Design Reference*, pp. 271-281, 2008.
- [13] V. Acharya, "Improving common-mode rejection using the right-leg drive amplifier," *Texas Instruments*, pp. 1-11, 2011.
- [14] B. B. Winter and J. G. Webster, "Driven-right-leg circuit design," *IEEE Transactions on Biomedical Engineering*, pp. 62-66, 1983.
- [15] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 21-30, 2008.
- [16] C. E. Shannon, "Communication in the presence of noise," *Proc. of the IRE*, vol. 37, pp. 10-21, 1949.
- [17] P. Morin, *Open Data Structures: An Introduction*, Athabasca University Press, 2013.
- [18] M. B. Conover, *Understanding Electrocardiography*, Elsevier Health Sciences, 2003.
- [19] M. A. Awal, S. S. Mostafa, M. Ahmad, and M. A. Rashid, "An adaptive level dependent wavelet thresholding for ECG denoising," *Biocybernetics and Biomedical Engineering*, vol. 34, pp. 238-249, 2014.
- [20] S. Poornachandra, "Wavelet-based denoising using subband dependent threshold for ECG signals," *Digital Signal Processing*, vol. 18, pp. 49-55, 2008.
- [21] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, pp. 1200-1224, 1995.
- [22] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, pp. 230-236, 1985.
- [23] W. J. Tompkins, *Biomedical Digital Signal Processing*, Editorial Prentice Hall, 1993.
- [24] J. Travis and J. Kring, *LabVIEW for Everyone*, Prentice-Hall, 2007.
- [25] A. J. Nimunkar, X. Zhang, M. Shokouinejad, and J. G. Webster, "Promoting active learning in biomedical engineering classes through blended instruction," *Age*, vol. 24, p. 1.
- [26] M. Shokouinejad, "Computer-Based sleep apnea detection and use of rebreath device ("Smart CO<sub>2</sub>") to treat sleep apnea," The University of Wisconsin - Madison, 2017.
- [27] I. D. Santos, et al., "Sleep apnea therapy device that automatically adjusts the fraction of inspired carbon dioxide," US 15/376,180 Patent, December 2016.
- [28] C. Fernandez, S. Rusk, N. Glattard, and M. Shokouinejad, "Computational phenotyping in polysomnography: Using interpretable physiology-based machine learning models to predict health outcomes," *J. Sleep Res.*, vol. 40, p. A26, 2017.



**Mehdi Shokouinejad** (SM'12, M'17) received his B.Sc. degree in electrical engineering from the University of Tehran, in 2011, and the M.Sc. and the Ph.D. degrees in Biomedical Engineering at the University of Wisconsin-Madison (UW-Madison), WI, USA, in 2013 and 2016, respectively. He joined Boston Scientific Corporation as a senior R&D scientist, MN, USA, in 2017. His main areas of research interest are the development of non-invasive and invasive physiological measurement and biomedical signal processing methods for assessment of the cardiovascular, brain and respiratory system (sensors, identification algorithms, and therapeutic devices). Dr. Shokouinejad was the recipient of the 2016 IEEE-EMBS, NSF Young Professionals

Contributing to Smart and Connected Health, FL, USA. He won the First Place Grand Award at the 2014 Qualcomm Innovation competition, WI, USA. He has been extensively involved in organizing and contributing to American research societies such as Annual Biomedical Society Meetings.



**Michelle K. Chiang** was born in Yokosuka, Japan. In 2015, she received her B.Sc. degree in biomedical engineering with a certificate in Japanese studies from the University of Wisconsin - Madison (UW-Madison, WI, USA). Subsequently, she continued at UW-Madison and received her M.S. degree in biomedical engineering in 2017. Chiang joined Johnson Health Tech. Co., Ltd. (Cottage Grove, WI, USA) in 2017 as an

Electrical Project Engineer in the Research and Development department. Her research interests include embedded systems development and medical instrumentation.



**Sam Lines** received his B.Sc. and M.Sc. degrees in Biomedical engineering from the University of Wisconsin-Madison (UW-Madison), WI, USA, in 2015 and 2017 respectively. He joined Plexus Engineering Solutions in 2017 as a software engineer focusing on medical device software.



**Fa Wang** (M'17) received the B.Sc. degree in Biomedical engineering from Fudan University, Shanghai, China in 2011, and Ph.D. degrees in Electrical and Computer Engineering from the University of Wisconsin-Madison (UW-Madison), WI, USA, in 2017. Dr. Wang joined Philips in 2017 as an Electrical design engineer focus on the MRI receiving coil design. His research interests include the hardware and

the software for the physiological measurement.



**Willis J. Tompkins** received the B.S. and M.S. degrees in electrical engineering from the University of Maine, Orono, Maine, USA in 1963 and 1965, respectively, and the Ph.D. degree in biomedical electronic engineering from the University of Pennsylvania, Philadelphia, Pennsylvania, USA in 1973. He is currently Professor Emeritus of Biomedical Engineering at the University of Wisconsin-Madison, where he has been on the faculty since 1974. He previously served for five years as Chair of the Department of Electrical and Computer Engineering. He has published four textbooks including: 1) Biomedical Digital Signal Processing, Prentice Hall, 1993; 2) Design of Microcomputer-Based Medical Instrumentation, Prentice Hall, 1981 (with J. G. Webster); and 3) Interfacing Sensors to the IBM PC, Prentice Hall, 1988 (with J. G. Webster). His research interests include development of microprocessor-based medical instrumentation and real-time computer processing of electrocardiograms. Dr. Tompkins is a Life Fellow of the IEEE (Institute of Electrical and Electronics Engineers), a Founding Fellow of the AIMBE (American Institute for Medical and Biological Engineering), and an Inaugural Fellow of BMES (Biomedical Engineering Society). He is a past President of the IEEE EMBS (Engineering in Medicine and Biology Society). He is a past Chair of the Biomedical Engineering Division of the ASEE (American Society for Engineering Education).



**John G. Webster** (M'59-SM'69-F'86-LF'97) received the B.E.E. degree from Cornell University, Ithaca, NY, USA in 1953, and the M.S.E.E. and Ph.D. degrees from the University of Rochester, Rochester, NY, USA in 1965 and 1967, respectively. He was a Highly Cited Researcher, King Abdulaziz University, Jeddah, Saudi Arabia. He is Professor Emeritus of Biomedical Engineering at the University of Wisconsin-Madison, USA. In the field of medical instrumentation he taught undergraduate and graduate courses in bioinstrumentation and design. He does research on a sleep apnea device that uses no increased pressure and a hydrocephalus shunt valve controlled by a pressure sensor. He is the editor of Medical instrumentation: application and design, Fourth Edition New York, John Wiley & Sons, 2010, and has developed 25 other books including the Encyclopedia of medical devices and instrumentation, Second Edition, New York: John Wiley & Sons, 2006, The physiological measurement handbook, CRC Press, Boca Raton, FL, 2015, Webster, J. G. and H. Eren (ed.), The measurement, instrumentation and sensors handbook, 2nd Ed. CRC Press, Boca Raton, FL, 2014, Eren, H. and Webster, J. G. (ed.), Telemedicine and electronic medicine, CRC Press, Boca Raton, FL, 2016, and 200 research papers. Dr. Webster is a fellow of the Institute of Electrical and Electronics Engineers, Instrument Society of America, American Institute of Medical and Biological Engineering, and Institute of Physics. He has been a member of the IEEE-EMBS Administrative Committee and the NIH Surgery and Bioengineering Study Section. He is the recipient of the 2001 IEEE-EMBS Career Achievement Award.