# Playing Pong for Stroke Rehabilitation Therapy Using a Real-Time Wireless Wearable Electromyogram Sensing System

A Major Qualifying Project submitted to the Faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the degree of Bachelor of Science

**Dilan Altiparmak**
Electrical & Computer Engineering

**Alexandria Miera**
Electrical & Computer Engineering

**Alicia Salvalzo**
Electrical & Computer Engineering

August 24, 2022

Report Submitted to

**Professor Edward A. Clancy, Ph.D.**
Worcester Polytechnic Institute
Department of Electrical & Computer Engineering

## Acknowledgements

The team would like to personally thank:

Professor Clancy, Kiki Rajotte, and He Wang for their continued support and guidance throughout the project.

Bill Appleyard for ordering various parts of the project.

## Executive Summary

This paper discusses the device that is designed and prototyped that uses EMG to control a game of Pong. A motivation for developing this device was in rehabilitation from stroke.

The project consists of three main parts, which will be further explained in the oncoming chapters of this report: Analog Front End, Microcontroller, and Pong App. Figure 1 shows the flow and components of the system.

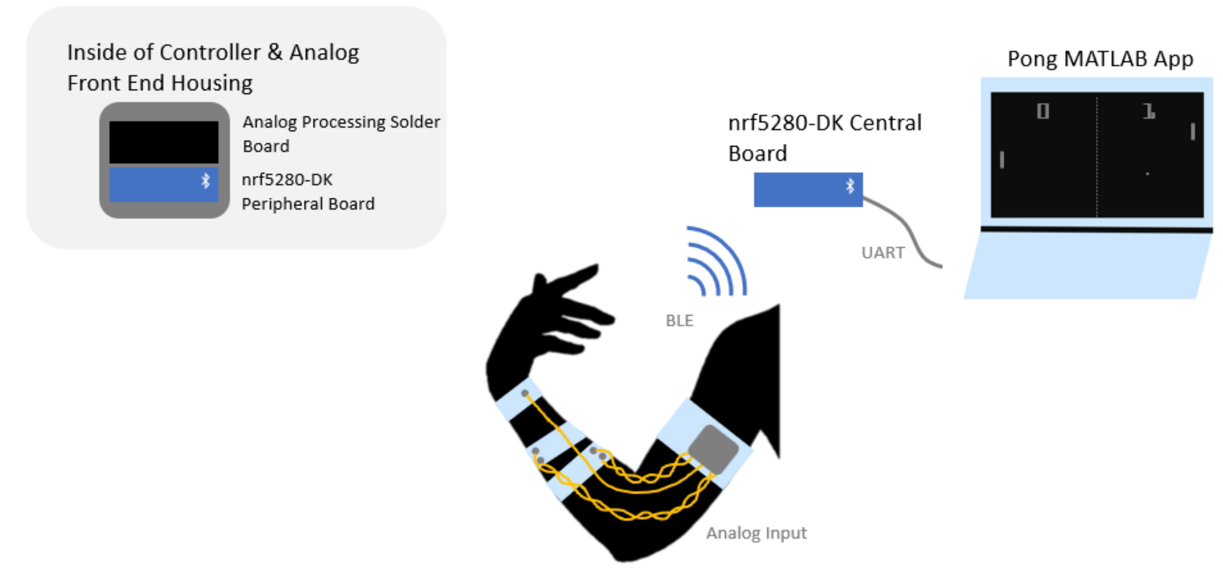

Inside of Controller & Analog Front End Housing
Analog Processing Solder Board
nrf5280-DK Peripheral Board
nrf5280-DK Central Board
Pong MATLAB App
0
1
UART
BLE
Analog Input

**Figure 1.** System Design

The analog front end consists of the electrodes, circuitry, and housing. For the electrodes, an armband was designed, and they are placed on the forearm to ensure more comfortable movement for the user. These electrodes are connected to the circuitry, which consists of a solder board circuit with the contents shown in the Figure below. The housing of the front end is an armband with the solder board and a microcontroller unit.

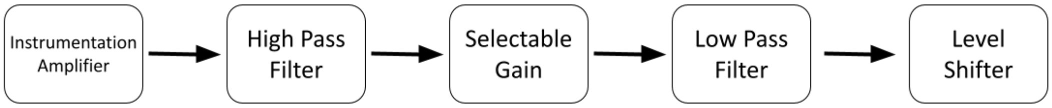


**Figure 2.** Analog Front End Components

The microcontrollers (MC) used for this project were NRF52840 by Nordic Semiconductors. Segger Embedded Studio was used as the SDK, and C was the programming language of the microcontroller. One microcontroller was used as the central node, and the other was used as the peripheral node. While one node, the central one, stayed connected to the computer, the other one, peripheral, was placed on the user on a secondary armband. In order for these modules to communicate with one another and transmit data, a BLE protocol was established. The data that is transmitted to the central node was sent to Matlab for further processing.

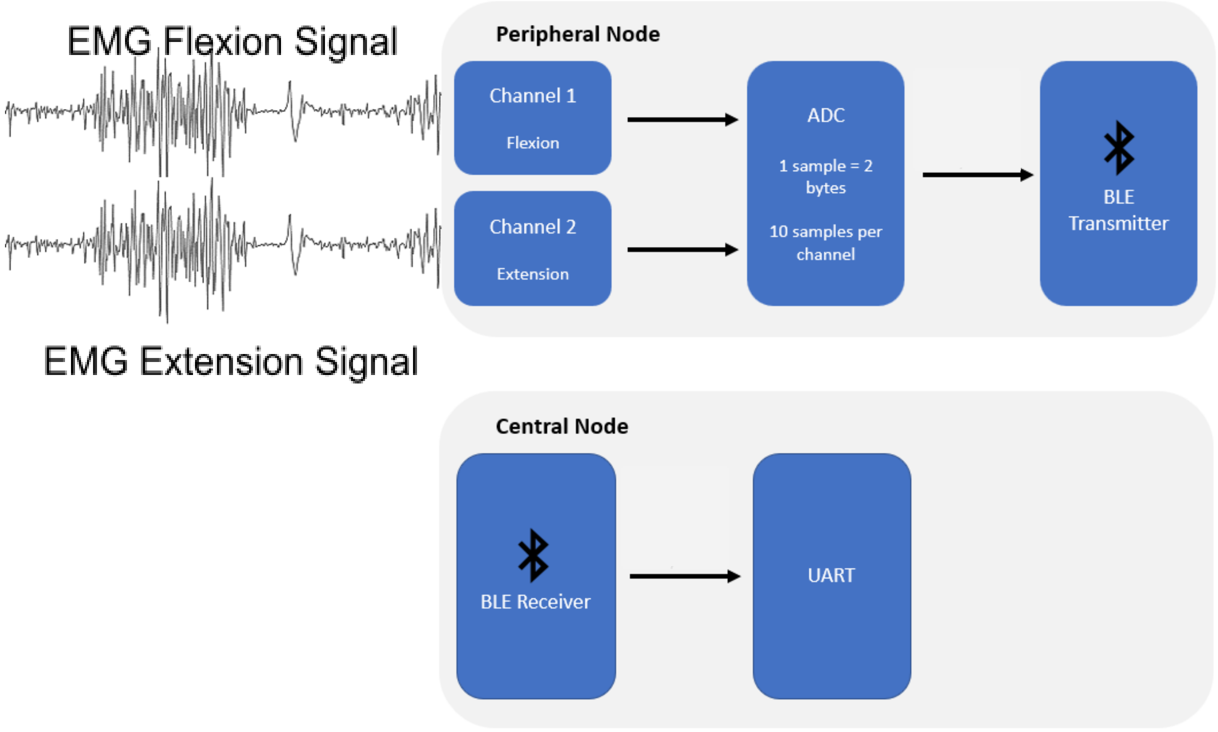


**Figure 3.** Microcontroller Process

The application for the Pong game and the EMG processing was done in Matlab. The signal from the two EMG channels on the arm are taken in as the data, in a multiplexed form, and they are demultiplexed as the first step of the process. After that, the data from both channels go into the signal processing function, and output as the cleaned EMG signals. The last step is to take the difference between the processed signals, i.e., flexion EMG minus extension EMG, to assess the direction and the intensity of the player's desired movement. This finding is then used in the Pong Game to control the paddle.

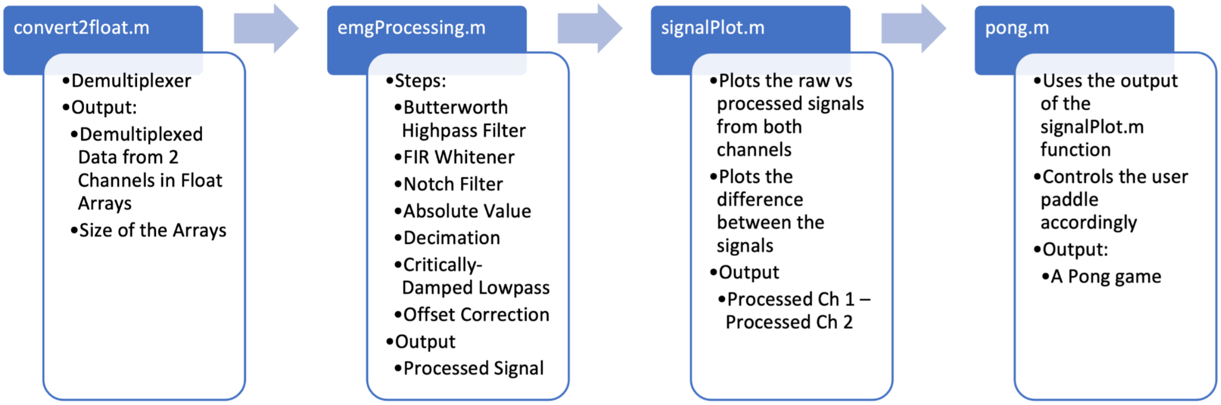

convert2float.m
•Demultiplexer
•Output:
•Demultiplexed Data from 2 Channels in Float Arrays
•Size of the Arrays
emgProcessing.m
•Steps:
•Butterworth Highpass Filter
•FIR Whitener
•Notch Filter
•Absolute Value
•Decimation
•Critically-Damped Lowpass
•Offset Correction
•Output
•Processed Signal
signalPlot.m
•Plots the raw vs processed signals from both channels
•Plots the difference between the signals
•Output
•Processed Ch 1 – Processed Ch 2
pong.m
•Uses the output of the signalPlot.m function
•Controls the user paddle accordingly
•Output:
•A Pong game

**Figure 4.** Matlab Software Flowchart

The final version of the system was tried on a sample dataset, and two individuals. We found that there are different thresholds and values for each individual's muscles, and the system is adjusted to fit the voltage range for different settings. According to our findings, it is recommended that the electrodes be placed as one that is very close to the elbow crease and one that is far from the wrist, in the middle of the forearm, preferably applied with electrode gel in between. The final appearance of the system is shown in Figure 5.

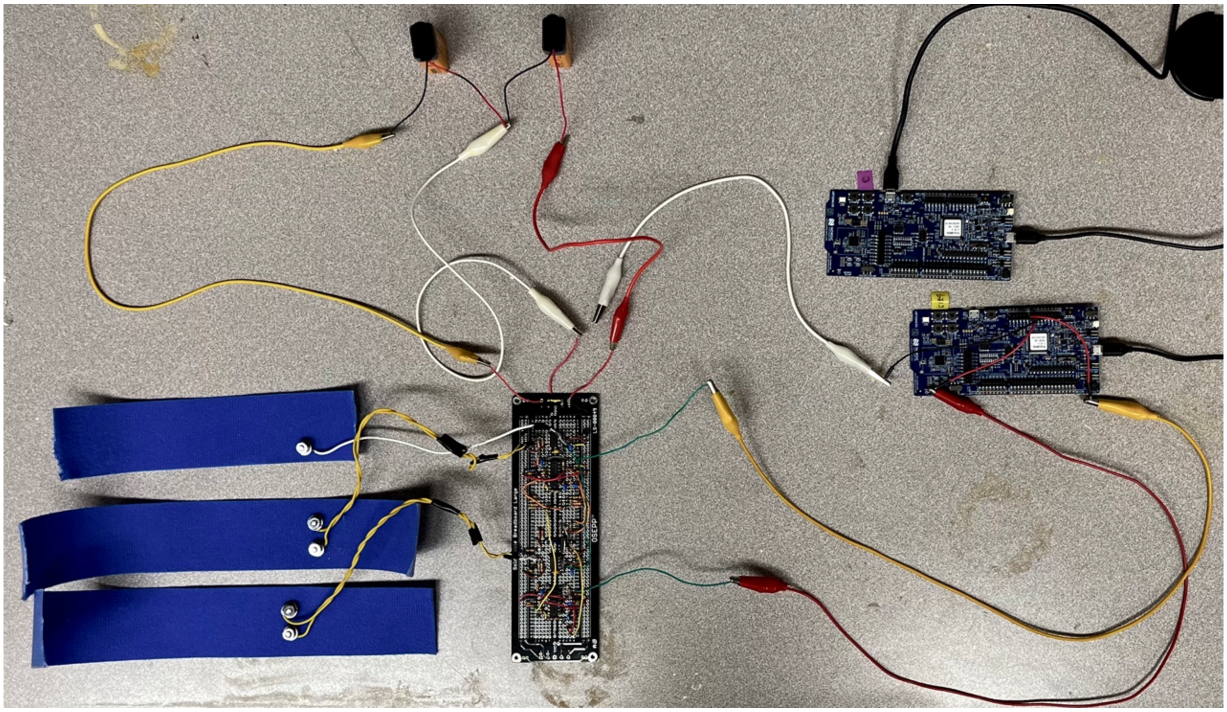


**Figure 5.** Final System

In the future, the housing can be designed to provide a more secure placement of the electrodes and more contact surface. Also, the circuitry can be turned into a PCB, to prevent any possible errors or offsets with the wiring. Furthermore, the microcontroller should be decreased in size to provide a more comfortable use (a smaller form factor is available), and the Matlab program should be rearranged to provide a faster execution, and better communication with the serial port.

**Acronyms**

(AAR) Accelerated Address Resolver

(ADC) Analog-to-Digital Converter

(AES CCM) Advanced Encryption Standard-Counter

(ATT) Attribute Protocol

(BLE) Bluetooth Low Energy

(CPU) Central Processing Unit

(DFT) Discrete Fourier Transform

(DK) Development Kit

(ECB) Electronic Code Block

(EMG) Electromyography

(ENOB) Effective Number of Bits

(FPU) Floating Point Unit

(GAP) Generic Access Profile

(GATT) Generic Attribute Profile

(GPIO) General input and output

(I/O) Input output pins

(IMU) Inertial Measurement Unit

(IoT) Internet of Things

(L2CAP) Logical Link Control & Adaptation Protocol

(MCU) Microcontroller

(MUAP) Motor Unit Action Potential

(NFC) Near Field Communication

(OOP) Object-Oriented Programming

(PCB) Printed Circuit Board

(POP) Procedure-Oriented Programming

(PPI) Programmable Peripheral Interconnect

(RAM) Random Access Memory

(RMS) Root Mean Square

(ROM) Read-only Memory

(RX) Receive

(sEMG) Surface Electromyography

(SIG) Special Interest Group

(SINAD, S/N+D, or SNDR) Signal to Noise and Distortion Ratio

(SMP) Security Manager

(SNR) Signal to Noise Ratio

(SoC) System on Chip

(TX) Transmit

(UART) Universal asynchronous Receiver/Transmitter

(USB) Universal Serial Bus

(WBAN) Wireless Body Area Network

## Abstract

This paper analyzes how electromyography (EMG) can be safely adapted to rehabilitation and/or entertainment purposes in the form of games; for the scope of this project, Pong Game. To answer this question, a system that is designed to be attached to the arm and assess the EMG signals from a person was designed with the help of electronic circuitry, nrf52840 microcontrollers, Segger Embedded Studio, and C programming language, and MATLAB. Our results have shown that there exists a strong correlation between the intensity of muscle effort and the signal amplitude. Results also revealed that the accuracy of the signal acquisition heavily relied on the placement of the electrodes and the contact area of electrodes with the skin. sEMG usage for the Pong game can become an important tool for the physical therapy of muscle or movement-related diseases, and an alternative to keyboard- or joystick-controlled games.

**Table of Contents**

**1. Introduction 13**

1.1 User Requirements 14

1.2 Product Requirements 14

**2. Literature Review 15**

2.1 Electromyography 15

2.1.1 Parts of the Motor Unit Action Potential 15

2.2 Wireless EMG Sensors 18

2.3 Analog Front End 19

2.3.1 Signal Characteristics 19

2.3.2 Maximizing EMG Signal Quality 19

2.3.3 Components of an EMG Sensor Analog Front End 20

2.3.3.1 Instrumentation Amplifiers 20

2.3.3.2 Band-Pass Filter 22

2.3.3.4 Non-Inverting Amplifier 24

2.3.4 Existing EMG Sensor Analog Front End 25

2.4 MCU 30

2.5 Bluetooth Low Energy 31

2.6 Analog-to-Digital Converter 34

2.6.1 Signal-to-Noise Ratio 35

2.6.2 Signal to Noise and Distortion Ratio 36

2.6.3 Effective Number of Bits 36

2.6.4.1 SINAD Calculation using DFT 39

2.6.5 Increasing ADC Resolution 40

2.6.5.1 Oversampling and Decimation 40

2.7 Pong 43

**3. System Design 49**

3.1 Module Definitions 49

3.2 Design Options 49

3.2.1 Phase I Design Options 49

3.2.1.1 Analog Front End 49

3.2.1.2 Controller 50

3.2.1.3 Pong App 50

3.2.2 Phase II Design Options 53

3.3 Final Design 57

**4. Analog Front End** **58**
4.1 Introduction 58
4.2 Design 59
4.2.1 Front End Housing Design Options 59
4.2.2 Front End Circuit Design Options 60
4.2.2.1 Power Supply 60
4.2.2.2 Instrumentation Amplifier 68
4.2.2.3 High Pass Filter 71
4.2.2.4 Selectable Gain 74
4.2.2.5 Low Pass Filter 74
4.2.2.6 Level Shifter 75
4.2.3 Electrode Design Options 77
4.2.4 Final Design 81
4.3 Testing 82
4.3.1 Methodology 82
4.3.2 Results & Observations 82
4.3.2.1 Multisim 82
4.3.2.2 Breadboard 87
4.3.2.3 Solder Board 95
4.4 Discussion 97
4.4.1 Design 97
4.4.2 Testing 97

**5. Controller** **98**
5.1 Introduction 98
5.2 Design 98
5.2.1 Design Options 98
5.2.2 Final Design 101
5.2.3 Technical Specifications 102
5.2.3.1 MCU - nRF52840-DK 105
5.2.3.1 SoC - nRF52840 107
5.2.4 Flow of Information 108
5.3 Testing 111
5.4.1 Methodology 111
5.4.2 Results & Observations 112
5.3 Discussion 113

5.3.1 Design 113
5.3.2 Testing 114

**6. Pong Application 115**
6.1 Introduction 115
6.2 Design 116
6.2.1 Design Options 116
6.2.1.1 Programming Language 117
6.2.1.2 Platform 119
6.2.1.3 Software Complexity 119
6.2.1.4 Software Tools and Design 121
6.2.1.5 Target User 124
6.2.2 Final Design 125
6.3 Testing 127
6.3.1 Methodology 127
6.3.2 Results & Observations 127
6.4 Discussion 137

**7. System Testing 138**
7.1 Methodology 138
7.2 Results & Observations 138

**8. System Results & Observations 147**

**9. System Discussion 149**
9.1 Use of Standards 149
9.2 Ethics and  Safety 149

**10. Conclusion & Future Work 150**

**11. References 151**

**12. Appendix 161**
Appendix A: Instrumentation Amplifier MATLAB Code  [E. A. Clancy] 161
Appendix B: High-Pass Filter MATLAB Code  [E. A. Clancy] 162
Appendix C: Low-Pass Filter MATLAB Code [E. A. Clancy] 165

# 1. Introduction

Stroke is the third leading cause of death and disability in the world. Currently, over one hundred million people are living with the effects of stroke aftermath [Feigin et al., 2022] which include paralysis, numbness, pain in the hands and feet, and other physical, occupational, and speech-related impairments [CDC 2, 2022]. Post-stroke, there are generally four defined clinical stages: the first 2 weeks are defined as the acute stages, weeks 3-11 are defined as the subacute stage, weeks 12-24 are defined as the early chronic stage, and more than 24 weeks is defined as the chronic stage [Wu et al., 2015]. Approximately 50 - 80% of stroke patients experience upper limb impairment in the acute phase and 40-50% are affected in the chronic phase [Hussain et al., 2018]. Patients who receive rehabilitation fewer days after stroke symptom onset have better functional outcomes and shorter rehabilitation length of stay [Maulden et al., 2005]. However, in practice, the time and cost of rehabilitation can be a restriction for patients [Hosseiniravandi et al., 2020].

Telerehabilitation approaches have been developed and tested to address the time and cost it takes for upper-limb rehabilitation. Generally, there are three types of upper-limb telerehabilitation approaches: virtual reality technologies, smart wearable devices, and robots [Moulaei et al., 2022]. Mixed reality telerehabilitation technologies are computer-assisted technologies that place the patient in a virtual or augmented environment with the purpose of gamifying the rehabilitation process. For example, Feng & Winters utilized force-feedback joysticks, driving wheels, and pointing devices in conjunction with computer software to remotely assess and train patients with stroke-induced upper limb impairments [Feng & Winters, 2005] while Tsekleves et al. used Nintendo Wii remotes [Tsekleves et al., 2014]. Generally, these systems, as well as robot-assisted rehabilitation systems [Razani et al., 2020] [Gopura et al., 2009] [Chen, 2021], require expensive and stationary hardware that may not be accessible to all patients. However, Qiu et al. used hand-tracking software to track, train, and assess patients with stroke-induced upper limb impairments [Qiu et al., 202]. Although the system is more accessible, it lacks direct biomedical signal feedback which can provide data on motor unit recruitment capability, fatigue, synergies, co-contraction, etc. [Capanini et al., 2020] and may help clinicians quantify muscle activity [Steele et al., 2020]. The wireless “Myo” armband [Uncrate, 2022] and the armband by CTRL-Labs [Rodriguez, 2022] have been developed to use surface electromyography (sEMG) and inertial measurement unit (IMU) [CT1] data in conjunction with motor assessments and gesture recognition [Maceira-Elvira et al., 2019]. However, “Myo” armband has since been discontinued [Sawers, 2018] and CRTL-Labs has been acquired by Facebook. sEMG systems developed by Noraxon and Delysis provide high-quality data but at a relatively high cost [Delsys, 2022] [Noraxon, 2022]. Therefore, a niche market for low-cost and open-source sEMG and/or IMU armbands is available. Yang et al. built a wireless sEMG armband that utilized machine learning to control a dexterous robotic hand for stroke rehabilitation [Yang et al., 2018]. Similar systems have been designed, however further research is still needed on human-computer interaction applications [Razfar et al., 2021]. Therefore, this project focuses on the initial design of a low-cost open-source real-time wearable sEMG system used to control the game of Pong. This research aims to create a user-friendly and cost-effective upper-body stroke rehabilitation and monitoring system.

### *1.1 User Requirements*

For the initial design, the user requirements are focused on functionality and ease of use because the system will be tested in an environment that requires rapid succession demos. At minimum users will expect that they can simply don and doff the system and then proceed to control a game of pong using sEMG.

### *1.2 Product Requirements*

The product requirements are focused on the minimum features required to validate the system idea which includes:

- An analog front end that transduces and processes the signals
- A controller that digitizes and transmits the signals to the device the application is running on
- An application and device that uses the signals to control the game of Pong.

Each of the above sections makes up one of the three system modules and each module is further broken down in chapters 4, 5, and 6.

# 2. Literature Review

## *2.1 Electromyography*

Electromyography (EMG), measures the electrical activity of the muscle as a response to nerve activation [Johns Hopkins Medicine, 2022]. Electrical signals from the muscles can be observed by attaching electrodes to the patient's skin. Sometimes, the electrodes are attached beneath the skin's surface using a needle or wire to make direct contact with the muscle itself [Johns Hopkins Medicine, 2022].

The EMG signal is the electrical expression caused by neuromuscular activation during muscular contraction [Bronlund et al., 2013]. When the muscle is not contracted, there is no electrical signal from the muscle.

The most elementary functional unit of a muscle is called a motor unit. A motor unit is composed of a motor nerve and the muscle fibers that it innervates. When the motor unit is activated, it creates a motor unit action potential (MUAP). Repetitive trains of MUAPs (MUAPTs) from many active motor units create the EMG signal [Bronlund et al., 2013].

### *2.1.1 Parts of the Motor Unit Action Potential*

The MUAP can be represented by five different parameters (see Figure 2): amp = amplitude; dur = duration; p = phase; t = turn; BL = baseline [Rodríguez-Carreño et al., 2012]. Amplitude is the voltage difference from minimum to maximum peaks. A MUAP's amplitude is determined by the few muscle fibers (typically less than eight) located within a 0.5mm radius from the electrode. The amplitude of an EMG signal normally has a maximum peak-to-peak value of around 10 mV. Amplitude varies depending on the size of the motor unit [Rodríguez-Carreño et al., 2012].

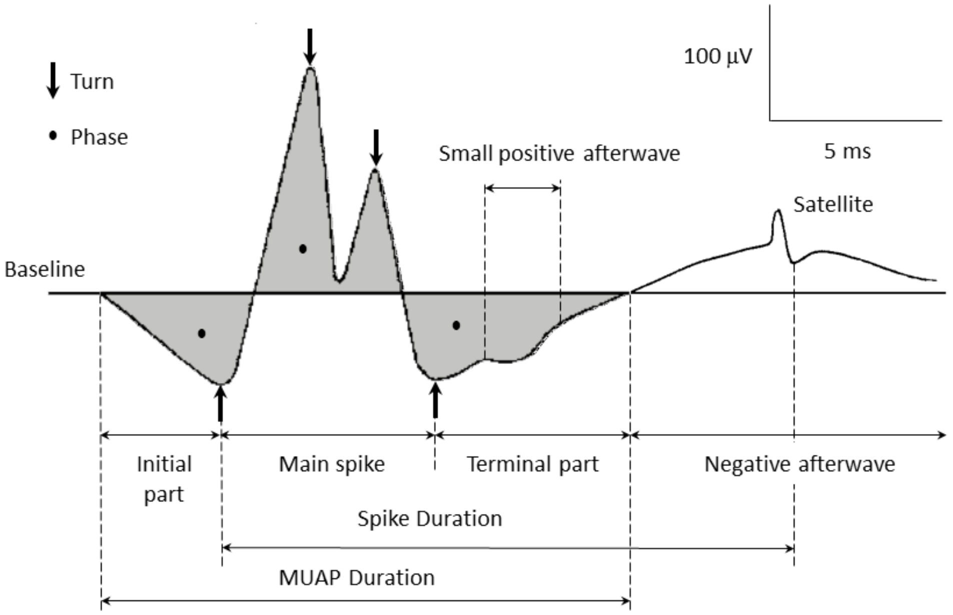


**Figure 2.1.1** Parts of the MUAP [Rodríguez-Carreño et al., 2012]

The phase is the section of the MUAP that falls between two baseline crossings. The number of phases in a single MUAP usually ranges from two to four. A polyphasic MUAP has more than four phases. In Figure 2.1.1, there are three phases, meaning this MUAP is considered normal [Rodríguez-Carreño et al., 2012].

A peak (change in direction) in an EMG signal is called a turn. If a signal has more than five turns, it is considered a polyturn or complex MUAP.

The MUAP begins at the start of activation of the motor unit and ends at the end of the repolarization phase, the time in which the motor unit discharges and bioelectric changes take place. This is known as the duration of the MUAP. The duration of the MUAP ranges from 8 to 14ms. Duration also varies depending on the size of the motor unit [Rodríguez-Carreño et al., 2012].

Shapes and time between MUAPs in EMG signals provide information for the diagnosis of neuromuscular disorders. This diagnosis is done via needle electrodes. The nature and characteristics of the EMG signals can be understood using proper hardware [Hussain et al., 2006]. Surface electrodes can be used to detect muscle activity to control prosthetics which aids the physically disabled and amputated population [Jamal, 2012].

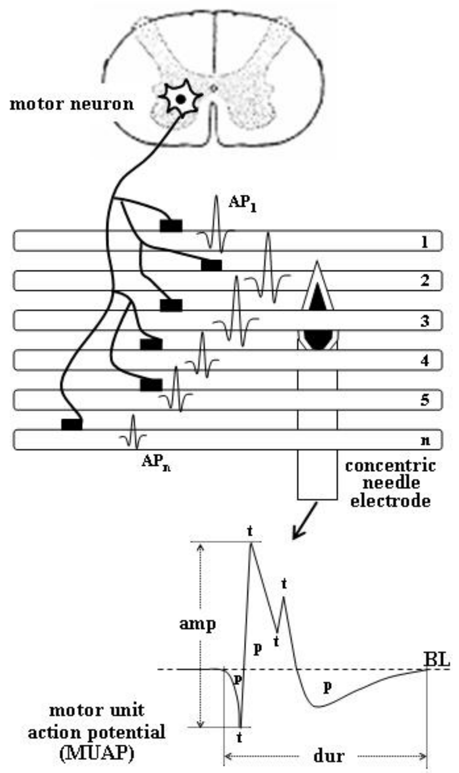

motor neuron
AP1
1
2
3
4
5
n
APn
concentric needle electrode
t
t
amp
p
t
p
BL
p
motor unit action potential (MUAP)
t
dur

**Figure 2.1.2** Motor Unit with “n” Muscle Fibers [Rodrı́guez-Carreño et al., 2012]

Figure 2.1.2 above represents a muscle motor unit with “n” muscle fibers. Each muscle fiber has its own action potential (AP) which are summed together to represent the MUAP at that moment in time.

The method of needle EMG seen in Figure 2 will not be the method used in this project. This project will use surface electrodes, which transduce a summation of MUAPTs from a group of motor units. Figure 2.1.3 shows a surface electrode attached to the skin.

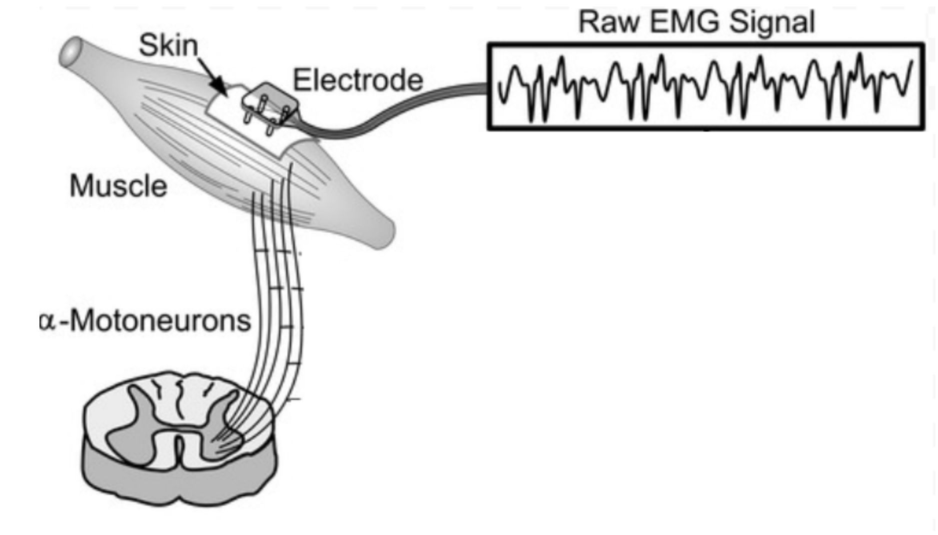


**Figure 2.1.3** sEMG Signal Using Surface Electrode [Modified from De Luca et al., 1982]

### *2.2 Wireless EMG Sensors*

Traditional sEMG, as well as other patient data collection devices, are or have been wired. However, as the world becomes increasingly interconnected through the internet of things (IoT) the market for (IoT) continues to grow. It is estimated to be a 2.4 billion dollar United States (U.S.) industry by 2029 [Fortune Business Insights, 2022]. As a result, wireless technology and research continue to be improved such that data collection and transfer is fast, reliable, and secure with respect to wired technologies. This is especially true for wireless body area networks (WBAN) which are a collection of wireless sensors placed on the surface of or inside the body to record physiological parameters such as the electrical signals of the muscles [Rahat et al., 2018].

The term wireless body area sensor networks (WBASN), which is synonymous with WBAN, was first used by Van Dam et al. in 2001 [Van Dam et al., 2001]. Although, the research dates back to 1996 in the work of Zimmerman on near-field intrabody communication [Zimmerman, 1996]. WBAN research has grown in popularity [Scopus Search Results], in part because it enables medical practitioners and patients to monitor their health and track their progress outside of traditional medical settings such as hospitals and doctor offices. And so, the portability of wireless sensors that make up WBANs is a key characteristic of their success. But, WBANs come with their challenges, including heterogeneous devices, traffic, energy efficiency, environmental factors, security, authentication, privacy, biocompatibility, quality of signal, interference, coexistence, and wireless propagation characteristics [Hasan et al., 2019].

Today, companies like Delsys and Noraxon have wireless sEMG systems available which are intended for human movement research, clinical, and educational settings [Delsys, 2022] [Noraxon, 2022]. However, such technology is not intended for consumers and for long-term use. Current research looks at ways in which sEMG can be used for medical monitoring, rehabilitation, and control schemes. Yang et al. researched ways in which facial sEMG could be used to monitor pain for patients who are unable to self-report [Yang et al., 2018a]. In addition, they researched hand-gesture recognition using sEMG and machine learning to support the stroke rehabilitation process [Yang et al., 2018b]. Similarly, Mendez et al. researched the feasibility of using sEMG hand-gesture recognition for prosthetic control and in virtual reality environments [Mendez et al., 2017]. Xi et al. researched how to detect falls and daily activities such as walking using sEMG and machine learning  [Xi et al., 2017]. Likewise, Luo et al. researched how to detect phases of the gait cycle [Luo et al., 2020].

## *2.3 Analog Front End*

The analog front end of the wireless EMG sensing circuit must include transduction, filtering, and amplification capabilities. The raw EMG signal contains noise artifacts from electronics equipment, ambient noise, motion artifacts, and inherent instability of the signal itself  [Hussain et al., 2006]. This section will discuss the characteristics of the EMG signal and how the signal can be altered with an analog front end to extract the desired information.

### *2.3.1 Signal Characteristics*

The typical amplitude of a sEMG signal ranges from 0 - 10 mV prior to amplification [Hussain et al., 2006]. Amplitude varies depending on the size of the muscle [Rodríguez-Carreño et al., 2012] and the effort level, among other factors. This signal must be amplified to be processed digitally by the analog-to-digital converter (ADC). This amplification range is dependent on the input voltage specifications of the ADC being utilized.

The typical frequency of an EMG signal ranges from 0-500 Hz, with the most dominant frequency range being 50 - 150 Hz. The EMG signal is unfortunately susceptible to a large amount of noise from various sources. Some external noise factors are impossible to avoid, these being radio transmission devices, fluorescent lights, and power line interference from electrical wires. Power line noise is found at 50 or 60 Hz and their harmonics.

Motion artifacts may also be created due to instability of the electrode skin interface and movement of the electrode cables. These motion artifacts fall within the frequency range of 0 - 20 Hz [Jamal, 2012].

### *2.3.2 Maximizing EMG Signal Quality*

To extract the desired information from an EMG signal, analog filtering techniques must be applied on the front end of the system. A signal often undergoes  “signal conditioning stages” prior to being digitized [De Luca et al.,  1982]

Maximizing the quality of the EMG signal can be done by having a high signal-to-noise ratio and limiting distortion of the signal, especially on the signal peaks [Hussain et al., 2006]. The signal-to-noise ratio is the ratio of signal power to unwanted signal or noise power [Cadence PCB Solutions, 2022].

*2.3.3 Components of an EMG Sensor Analog Front End*

An EMG sensor analog front end usually consists of three main stages: instrumentation amplifier, band-pass filter, and selectable gain amplifier.

2.3.3.1 Instrumentation Amplifiers

An instrumentation amplifier is a type of differential amplifier that amplifies the difference between two signals, while maintaining a very high input impedance. In the case of an EMG sensor, the difference is between two EMG electrodes [Ibrahim et al., 2021].

The skin of a human has a high output impedance, which requires the input of the instrumentation amplifier to have a high input impedance. This can be understood by thinking of the skin's impedance and the front end circuit impedance as two resistors in a voltage divider. If both resistances are the same, only half of the input voltage will be measured across the circuit. If the circuit impedance is greater than the skin impedance, more voltage will be applied across the circuit. [megcircuitsprojects, 2022].

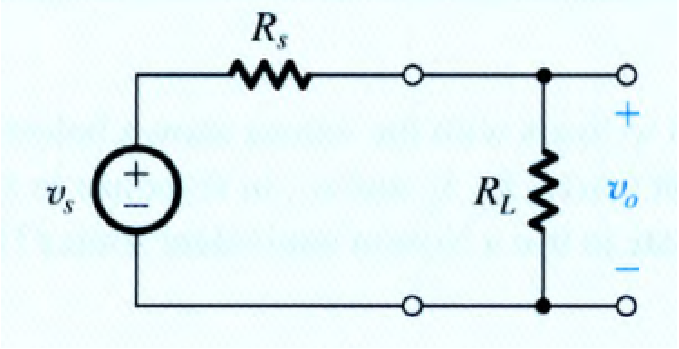


**Figure 2.3.1** Voltage Divider Schematic [Sedra & Smith, 2020]

The voltage divider equation, based in the voltage divider circuit shown in Figure 2.3.1, is as follows:

$$V_0 \;=\; V_S\,(\frac{R_L}{R_L + R_S})$$

*[Sedra & Smith, 2020]*

If we replace $V_S$ with the EMG signal, $V_0$ with voltage being measured across the circuit, and resistances $R_S$ and $R_L$ with skin impedance and circuit impedance, respectively, we get the equation:

$$V_{measured} = V_{signal}\left(\frac{Circuit\ Impedance}{Circuit\ Impedance + Skin\ Impedance}\right)$$

Since circuit impedance appears in both the numerator and denominator of this ratio, the higher the circuit impedance, the more input voltage will be measured across the circuit.

A high input impedance can be obtained by using an instrumentation amplifier (Figure 2.3.2). An instrumentation amplifier can be modeled using two non-inverting amplifiers and one differential amplifier. The gain of the instrumentation amplifier is [Austerlitz, 2003]:

$$Vout = [B - A] * \frac{R4}{R2}\left[1 + \left(\frac{2 \cdot R1}{R3}\right)\right],$$

where A and B are the two input voltages. The circuit provides a high input impedance and a low output impedance and has tunable gain by adjusting the values of the resistors. Resistors R2 and R4 are often the same value, making R4/R2 = 1. R3 sets the gain of the amplifier circuit, which is known as the gain resistor. It is helpful to add a large capacitor in series with R3 so that the DC potential difference between the two inputs A and B (i.e., their offset) does not get amplified.

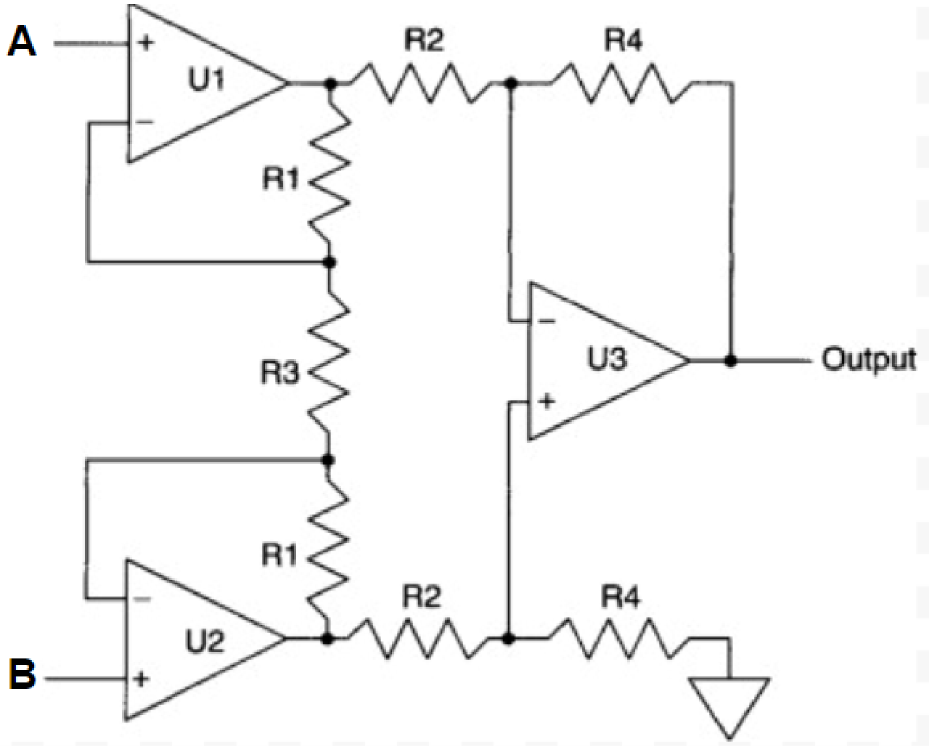


**Figure 2.3.2** Instrumentation Amplifier Model (Modified from Austerlitz, 2003)

2.3.3.2 Band-Pass Filter

The second stage of the AFE circuit should include a band-pass filter. A band-pass filter consists of a low-pass and high-pass filter.

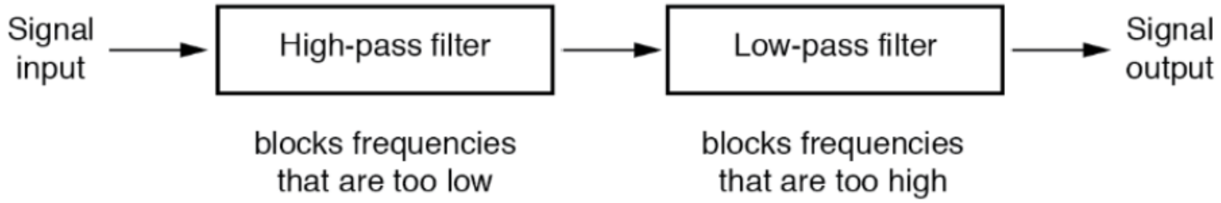

Signal input
High-pass filter
Low-pass filter
Signal output
blocks frequencies that are too low
blocks frequencies that are too high

**Figure 2.4.3** Block Diagram of Band-Pass Filter [Modified from *allaboutcircuits.com, n.d.*]

The band-pass filter will remove noise artifacts that are not part of the desired EMG signal and provide anti-aliasing filtering. As stated previously, the typical frequency of noise artifacts are either 50 or 60 Hz (and their harmonics) from the power lines and 0 - 20 Hz from motion. Filtering is needed to block these frequency ranges from passing through, as well as frequencies above half the sampling rate.

In this project, the high-pass filter should come before the low-pass filter, as seen in Figure 5. This configuration eliminates the DC offset from the instrumentation amplifier stage to avoid saturation.

A band-pass filter can be used to eliminate noise that is outside the desired EMG frequency range. The 0 - 20 Hz motion artifact from movement of the electrodes can be attenuated by a high-pass filter and any frequencies above the desired EMG range can be attenuated by a low-pass filter. The 60 Hz frequency from the United States power lines can be attenuated by a narrow notch filter, if desired (In some systems, this filtering is performed digitally after sampling by the ADC). The notch filter is a type of band-pass filter with a very narrow and steep stop band [Electronics Tutorials, 2022]. This allows for the small frequency range of 10Hz from the power lines that is within the EMG signal frequency range to be filtered out efficiently. This notch is usually set for around 2 Hz to provide some buffer room to make sure all components of the power line frequency are removed.

A common band-pass filter circuit topology is the Sallen-Key topology. High and low pass Sallen-Key filters can be easily cascaded together, making it easy to create a well-functioning band-pass filter. When designing the magnitude response of the filters, it is desired to have a flat magnitude response in the pass band to accurately depict the EMG signal. This can be achieved using a Butterworth low and high pass filter with Sallen-Key topology. Butterworth filters provide a flat magnitude response in the pass band which is desirable for an accurate filtered signal [Electronics Tutorials, 2022].

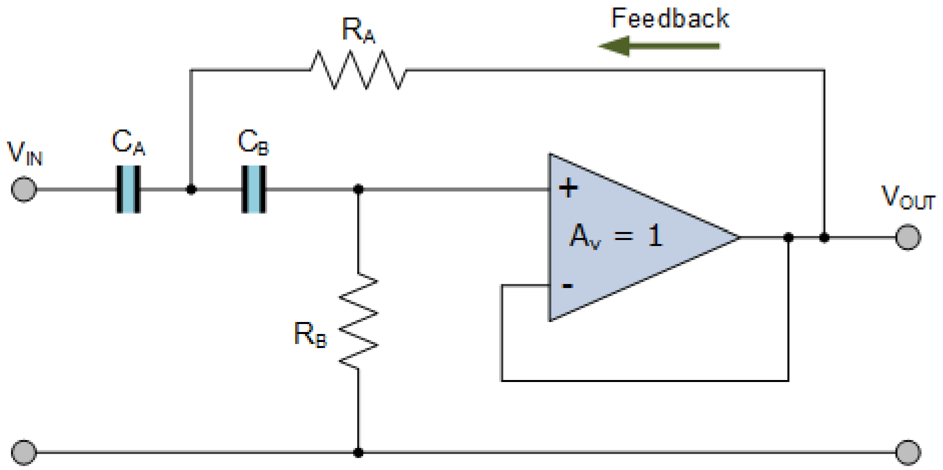


**Figure 2.3.4** Unit-Gain High-Pass Sallen-Key Circuit Topology [Electronics Tutorials, 2022]

$$f_C = \frac{1}{2\pi\sqrt{R_A R_B C_A C_B}}$$

**Figure 2.3.5** Butterworth Filter Cut-Off Frequency Equation  [Electronics Tutorials, 2022]

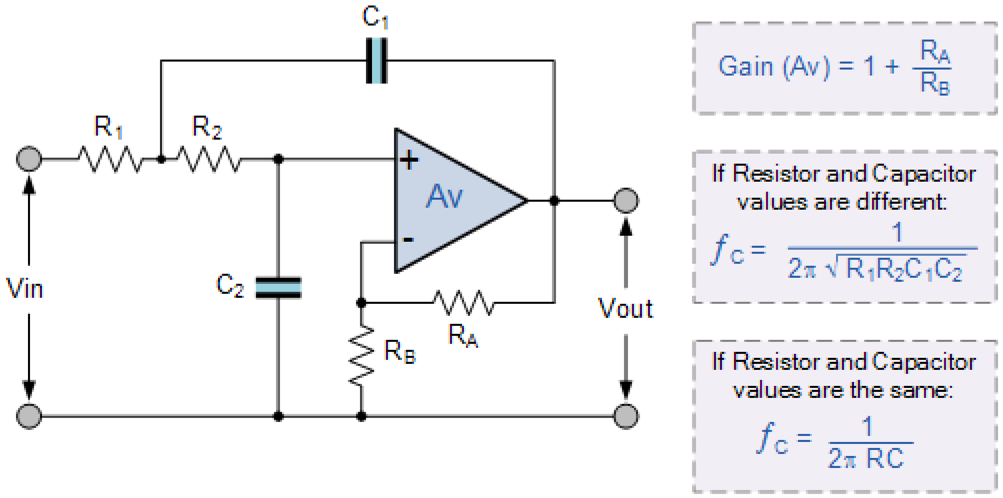

C₁
R₁
R₂
Av
+
-
Vin
C₂
R_B
R_A
Vout
Gain (Av) = 1 + $\frac{R_A}{R_B}$
If Resistor and Capacitor values are different:
$f_c = \frac{1}{2\pi\sqrt{R_1R_2C_1C_2}}$
If Resistor and Capacitor values are the same:
$f_c = \frac{1}{2\pi RC}$

**Figure 2.3.6** Low-Pass Sallen-Key Circuit Topology [Electronics Tutorials, 2022]

The output from the instrumentation amplifier stage is passed through to the input of the high-pass Butterworth filter, then through the low-pass Butterworth filter to create the band-pass filtering effect.

2.3.3.4 Non-Inverting Amplifier

A non-inverting amplifier will provide the correct amount of gain to the signal so that the signal can be processed by the analog-to-digital converter. The amount of gain needed depends on the voltage range of the ADC that is chosen.

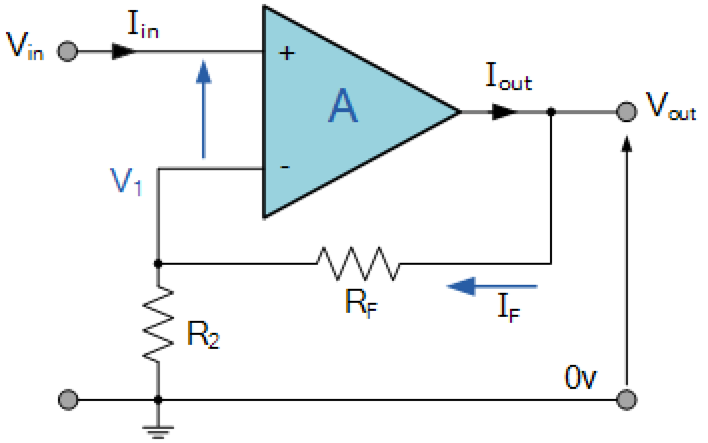


**Figure 2.3.7** Non-Inverting Amplifier Circuit [Electronics Tutorials, n.d.]

The gain equation of a non-inverting amplifier is:

$$Gain = 1 + \frac{RF}{R2}$$

The output voltage of the non-inverting amplifier is calculated by Vin ● Gain:

$$Vout \ = \ Vin\,(1 \ + \ \frac{RF}{R2})$$

An inverting amplifier could also be used instead of the non-inverting configuration. The purpose of this amplification stage is to provide gain to the signal so that it can be processed by the ADC, so the configuration of this stage (inverting vs. non-inverting) does not matter.

The input (Vin) to the non-inverting amplifier will be the output of the band-pass filter circuitry. The output (Vout) will become the final desired EMG signal which then becomes the input to the ADC. The ADC will take the signal and digitize it so that it can be used by the application.

### *2.3.4 Existing EMG Sensor Analog Front End*

A previous MQP team from March 2021 completed a design of an analog front end circuit for EMG use [Ibrahim et al., 2021]. Their design consists of two main stages: an instrumentation amplifier and band-pass filter with selectable gain.

The instrumentation amplifier used in the March 2021 MQP was the AD620. This instrumentation amplifier includes two internal resistors of value 24.7kΩ, for an overall internal resistance of 49.4kΩ. The gain of the amplifier can be set by using an external resistor, RG. The gain of this amplifier can be calculated by the following formula: $Gain \ = \frac{49.4k\Omega}{RG} + \ 1$ .

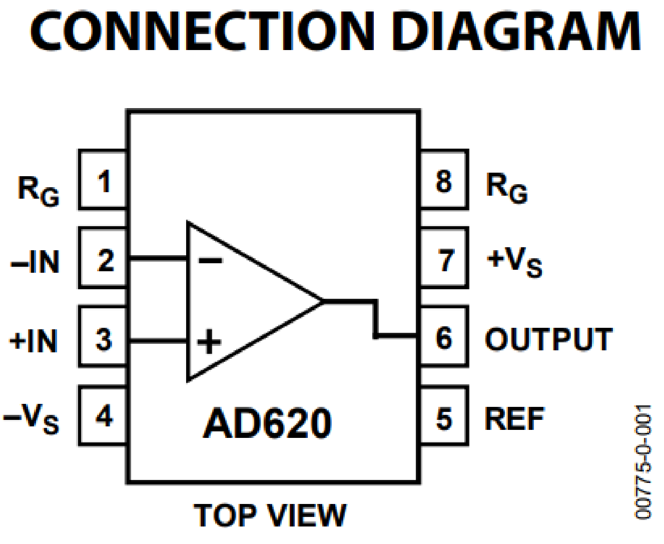


**Figure 2.3.8** Pin Connections of AD620 Amplifier [*Analog.com*]

To test the circuit, the MQP team built the Multisim schematic in Figure 2.3.9. A 1kHz, 15mV peak sine wave was used to replicate the raw EMG signal from the electrodes. The RG value, seen as R1 in the Figure, was chosen as 1kΩ to provide a gain of 50. The 100uF capacitor was placed in series with R1 to ensure a gain of 1 at DC. Capacitors C5 and C6 were used as decoupling capacitors to keep the input voltages stable.

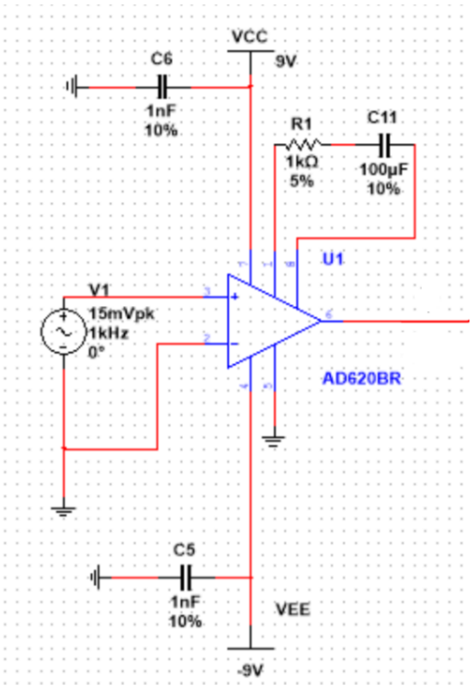


**Figure 2.3.9** EMG Instrumentation Amplifier Simulation Circuit [Ibrahim et al., 2021]

This schematic in Figure 2.3.10 is a band-pass filter with selectable gain capability [Ibrahim et al., 2021]. This circuit includes a fourth-order high-pass filter, a selectable gain stage, and fourth-order low-pass filter. This portion of the analog front end takes care of filtering the unwanted noise out of the raw EMG signal gathered by the electrodes.

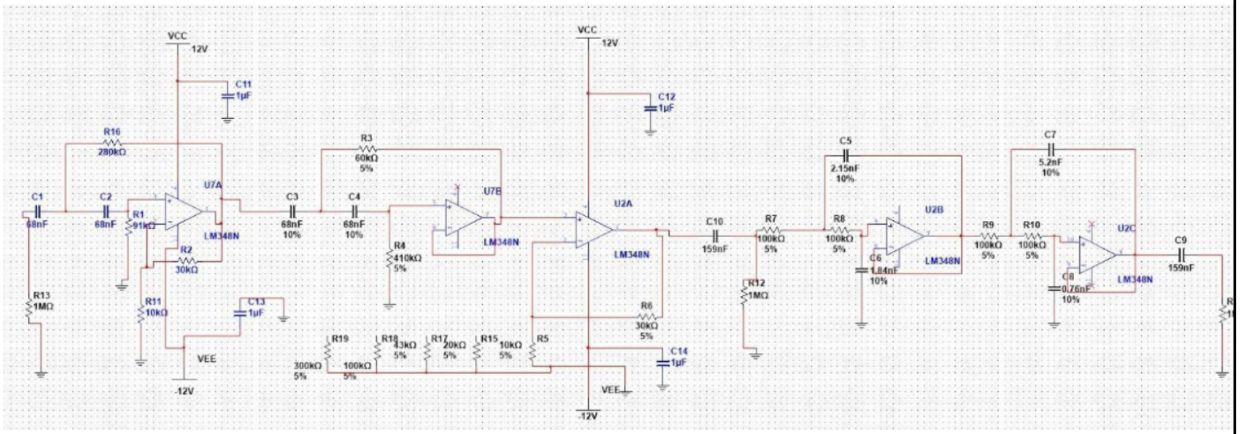


**Figure 2.3.10** Band-Pass Filter with Selectable Gain [Ibraham et al., 2021]

The eighth-order band-pass filter stage is made up of a fourth-order, low-pass Butterworth filter and fourth-order, high-pass Butterworth filter.

The high-pass filter stage (Figure 2.3.11) was created using capacitor values to provide a cutoff frequency of 15 Hz and resistor values to provide a gain of 2.

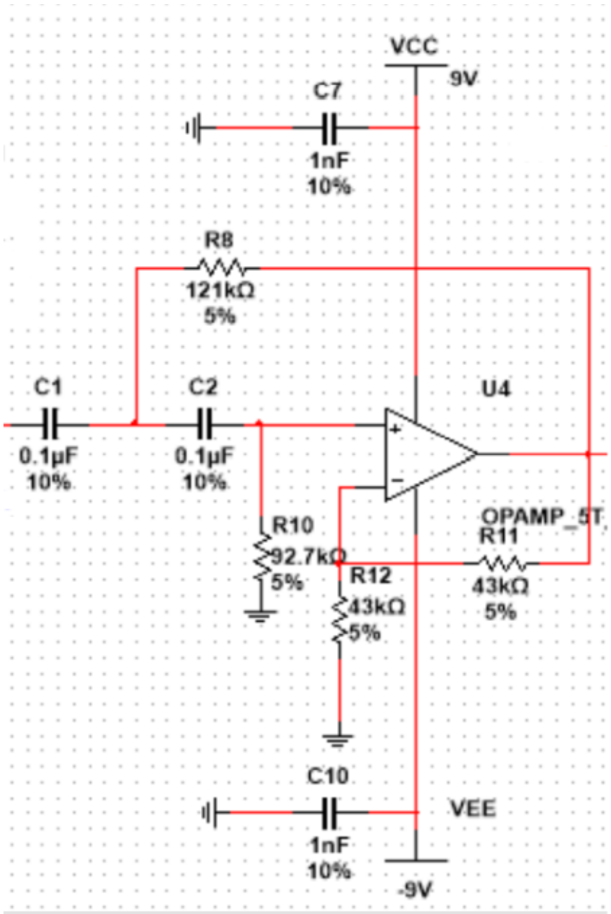


**Figure 2.3.11** High-Pass Filter Stage [Ibraham et al., 2021]

The low-pass filter stage (Figure 2.3.12) was created using capacitor values for a cutoff frequency of 800 Hz and a gain of 1. No components were added to the negative feedback loop to create the gain of 1.

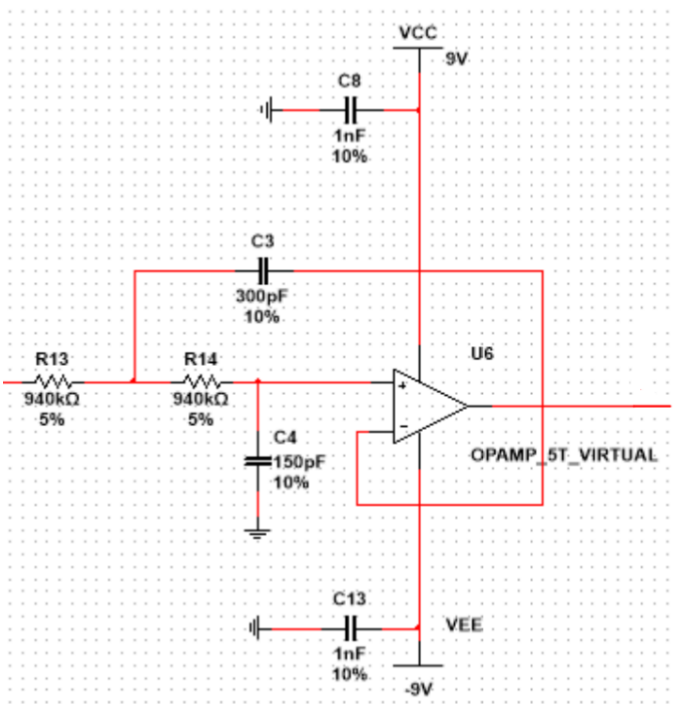


**Figure 2.3.12** Low-Pass Filter Stage [Ibraham et al., 2021]

The selectable gain stage was created using a non-inverting amplifier. The value of R3 can be varied by the resistors in Figure 2.3.14 to create the change in non-inverting gain. The resistor values available are 10kΩ, 20kΩ, 43kΩ, 100kΩ, and 300kΩ. The selectable gain was created to make sure that the smallest possible EMG signal of 10 mV can be amplified to the 3.3 V range of the ADC. An RC circuit was added to the output of the amplifier to eliminate any offset from the previous two stages without altering the entire pass-band [Ibraham et al., 2021].

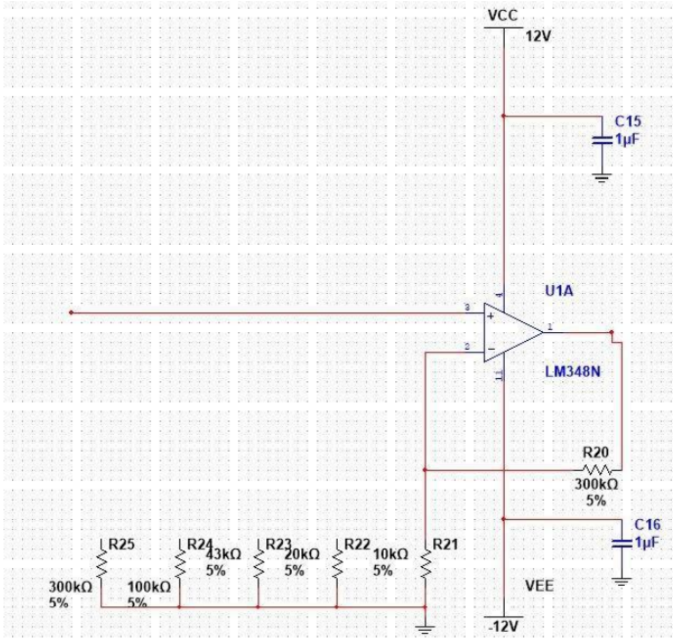


**Figure 2.3.13** Selectable Gain Stage [Ibraham et al., 2021]

### *2.4 MCU*

A microcontroller (MCU) is a single-chip computer that, as the name suggests, is small and intended to control applications [Ibrahim, 2014]. MCUs used for WBANs, and IoT applications alike, are constrained to be cost-efficient and area-efficient while still meeting real-time constraints and maximizing battery life [Chéour et al., 2020].

The first MCU was introduced by Intel back in 1976 as the 8048 which was part of the MCS-48 family [Raghunathan, 2021]. The 8048 was an 8-bit MCU that contained a central processing unit (CPU), random access memory (RAM), read-only memory (ROM), and input and output (I/O) pins [Raghunathan, 2021]. Since, MCUs have increased in bit size, i.e., the size of the registers, the number of memory addresses, and the largest numbers that can be processed. In addition, there have been improvements in the performance and or size of the processing units, communication interfaces, and peripherals.

In 1971, the first CPU was released by Intel as the Intel 4004 [Cass, 2018]. From then to now, improvements have been made to the size, speed, and use of the transistors that make up the CPUs. In 1895, the first wireless communication device, known as the wireless telegraph, was created by Guglielmo Marconi [Falciasecca & Valotti, 2009]. Then, in the late 1990s Bluetooth was developed and released [Wells, 200], and later on, Bluetooth low energy was developed in 2004 and released in 2006 under the name of Wibree [Honkanen et al., 2004]. During this time, other short-range wireless communication technologies were also developed, released, and integrated into MCUs including Zigbee, Thread, and near-field communication (NFC).

As the technology has advanced over the years, opportunities for feasible information of things (IoT) devices have grown resulting in a wide variety of MCUs catered towards IoT and wearable applications. MCUs commonly used in IoT applications for medicine include the megaAVR series, ABR x mega series, Arm Cortex M series, TI MSP 430, and Intel 8051 [Behmanesh et al., 2020] [Chéour et al., 2020].

For this project, the Nordic nRF52840 development kit (DK) MCU is used. The MCU interfaces with the nRF52840 system on chip (SoC) which supports a wide variety of short-range communication technologies, including Bluetooth low energy, as well as a host of security features and sub-power modes making it an option for IoT, advanced wearable, and interactive entertainment devices [Nordic, n.d. a].

### *2.5 Bluetooth Low Energy*

Bluetooth low energy (BLE) is a successor of Bluetooth classic and was designed by Nokia back in 2004 to support low power-constrained applications [Honkanen et al., 2004]. In 2006, Nokia officially publicized the technology under the name of “Wibree” and in 2010, the Bluetooth Special Interest Group (SIG) first introduced the technology in their 2010 4.0 Core Specifications document [Bluetooth SIG, 2010]. Since, SIG has updated the specifications in their 2021 5.3 Core Specifications document [Bluetooth SIG, 2021]. For this project, the 5.0 specifications are used.

A stack is a set of protocols used to enable wireless communication. BLE supports two different stacks, the BLE stack, and the Bluetooth mesh stack. This project will use the BLE stack which is made up of nine layers as shown in Figure 2.5.1. The two architectural components of the BLE stack are the host and controller which interface with each other via the host controller interface. The functionality of each layer is described in Table 2.5.1. For this project, the software that enables BLE communication comes embedded in the nRF52840 SoCs.

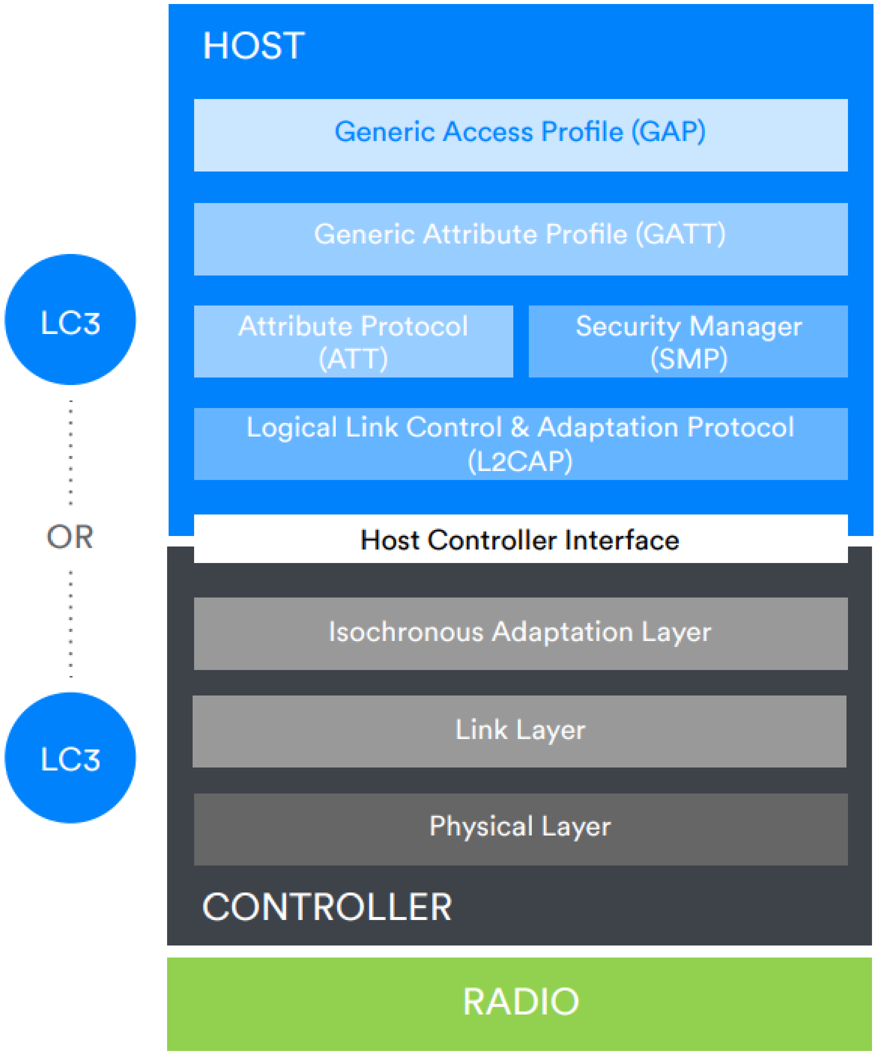

HOST
Generic Access Profile (GAP)
Generic Attribute Profile (GATT)
Attribute Protocol (ATT)
Security Manager (SMP)
Logical Link Control & Adaptation Protocol (L2CAP)
Host Controller Interface
Isochronous Adaptation Layer
Link Layer
Physical Layer
CONTROLLER
RADIO
LC3
OR
LC3

**Figure 2.5.1** BLE stack [Nordic, 2021]

| Layer | Details |
|---|---|
| Physical Layer | Defines how transmitters and receivers are used to encode and decode digital data for transmission and receipt in addition to other radio-related properties including symbol rate, protocol data rate, max application data rate, error detection, error correction, time division, transmission power, and receiver sensitivity |
| Link Layer | Defines types of packets transmitted over the air and the rules of communication. There are different states, as shown in Figure 2.4.2, that affect how the link operates. Each state is described in Table 2.4.2. |
| Isochronous Adaptation Layer | Allows different frame durations to be used by devices using isochronous communication. If the frame duration of the transmitter differs from that of the receiver, the isochronous adaption layer is used to resolve this conflict. |
| Host Controller Interface | The Host Controller Interface (HCI) defines a standardized interface via which a host can issue commands to the controller and a controller can communicate with the host. |
| Logical Link Control & Adaptation Protocol (L2CAP) | Responsible for protocol multiplexing, flow control, and segmentation and reassembly of service data units (SDUs). Service data units are packets that have been transferred to a lower-level layer by an upper-level layer. |
| Attribute Protocol (ATT) | A protocol used by an ATT client and an ATT server that allows the discovery and use of data in the server’s attribute table. It defines how data is represented. |
| Security Manager (SMP) | Supports the execution of security-related procedures such as pairing, bonding, and key distribution. |
| Generic Attribute Profile (GATT) | Defines higher-level data types based on the attributes held in the attribute table (see 11. The Attribute Protocol). Those data types are called services, characteristics, and descriptors. It also defines a series of procedures involved in using these data types via the Attribute Protocol (ATT). The relationship between services, characteristics, and descriptors are as shown in Figure 2.4.3. |
| Generic Access Profile (GAP) | Defines procedures concerned with device discovery and establishing connections between two devices. |

**Table 2.5.1** BLE layer descriptions derived from Nordic, 2021

### *2.6 Analog-to-Digital Converter*

Analog-to-digital converter (ADC) is a device that converts analog signals into digital signal form by taking discrete samples from the continuous signal and representing them in binary form. [Actel Corporation, 2007] While most data are characterized by analog signals, various processors can gather, interpret and operate on data only in digital form. Sampling is the first step in the conversion, in which the ADC performs the conversion periodically by taking samples from the input rather than continuously performing the operation. This converter's main function is to convert the continuous time-domain analog signal to a discrete amplitude digital signal [Jamuna S et al., 2018].

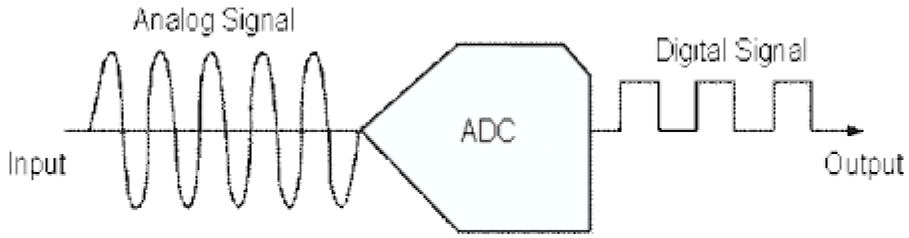


**Figure 2.6.1** Analog-to-Digital Converter [Jamuna S et al., 2018]

ADCs are usually characterized in three ways: input voltage range, resolution, and bandwidth, or conversion rate. [Actel Corporation, 2007]

| Term | Definition |
|---|---|
| Input Voltage Range | The range of common-mode signal that the ADCs operation will remain linear. Determined by the reference voltage ($V_{ref}$) [Actel Corporation, 2007] |
| Resolution | Function of the number of bits. The input voltage value is approximated using $2^n$ steps (n = number of bits in ADC) [Actel Corporation, 2007] |
| Bandwidth (Conversion Rate) | Represents the maximum number of conversions that can be performed by the ADC every second. The bandwidth is restrained by the architecture and key performance characteristics [Actel Corporation, 2007] |

**Table 2.6.1** General ADC characteristics

ADC performance is measured using the effective number of bits (ENOB), along with SINAD (Signal-to-Noise Distortion Ratio), SNR (Signal-to-Noise Ratio), THD (Total Harmonic Distortion), THD + N (Total Harmonic Distortion plus Noise), and SFDR (Spurious Free Dynamic Range).

| Measure | Definition | Generic Formula |
|---|---|---|
| Signal-to-Noise Ratio (SNR) | Comparison of the desired signal power and noise power [Kieser et al., 2005] | $10log(Signal/Noise)$<br>**[Eq. 2.6.1]** |
| Signal-to-Noise and Distortion Ratio (SINAD, S/N+D, SNDR) | Variation between the ideal signal and the measured signal for a specified number of measurements [intel.com, n.d.] | $\frac{P_{signal} + P_{noise} + P_{distortion}}{P_{noise} + P_{distortion}}$<br>$= (6.02 * 1.76B),\ B = no\ of\ Bits\ in\ AD$<br>**[Eq. 2.6.2]** |
| Total Harmonic Distortion (THD) | Ratio of the RMS value of the signal to the mean value of the root-sum-square of its harmonics [Kester, 2009] | $\frac{\sqrt{V_2^2 + V_3^2 + ... + V_n^2}}{V_1}$<br>[Microsemi, 2015]<br>**[Eq. 2.6.3]** |
| Total Harmonic Distortion and Noise (THD + N) | Ratio of the RMS value of the signal to the mean value of the root-sum-square of its harmonics plus all the noise components [Kester, 2009] | $\frac{\sqrt{A_2^2 + A_3^2 + ... + A_n^2}}{\sqrt{A_1^2 + A_2^2 + ... + A_n^2}}$<br>[ni.com, 2022]<br>**[Eq. 2.6.4]** |
| Spurious Free Dynamic Range (SFDR) | Ratio of the signal RMS value to the worst spurious signal RMS value [Kester, 2009] | $20log(\frac{Amplitude\ of\ Fundamental\ RMS}{Amplitude\ of\ Largest\ Spur\ RMS})$<br>[ni.com, 2020]<br>**[Eq. 2.6.5]** |
| Effective Number of Bits (ENOB) | Measure of analog-to-digital converter through the test frequency and signal-to-noise ratio [maximintegrated.com, n.d.] | $ENOB = \frac{SINAD - 1.76\ dB}{6.02}$<br>[Kester, 2009]<br>**[Eq. 2.6.6]** |

**Table 2.6.2** Definitions and Formulas of ADC Performance Measures

### *2.6.1 Signal-to-Noise Ratio*

Signal-to-noise ratio (SNR) is the fundamental frequency signal power level ($P_S$) to the noise power level ($P_N$) ratio, which can be seen in Equation 1 above.

The ideal theoretical SNR is calculated directly from the resolution (N Bit), where an ideal ADC with a linear characteristic is considered. The ADC would quantize the incoming signal, which would include quantization noise in the system. The signal and noise power would make it possible to calculate the SNR. [Schaefer, 2011] In the case of using a full-scale sine-wave as the input, the equation would be:

$$SNR = (6.02 * 1.76B)dB, where\ B\ denotes\ the\ number\ of\ Bits\ in\ the\ ADC \qquad \text{(Eq. 2.6.7)}$$

*2.6.2 Signal to Noise and Distortion Ratio*

SINAD is the fundamental frequency signal power level ($P_S$) to the noise plus distortion power level ($P_{N+D}$) ratio, expressed in Equation 2. SINAD is the ratio of the amplitude of the input signal to the RMS sum of all the other spectral components, except for DC. A higher SINAD value signifies a better system. [Lundberg, 2002]

*2.6.3 Effective Number of Bits*

The ENOB is the number of bits when both noise and distortion are considered. As can be viewed from the equation 2.6.6, ENOB is identical to SINAD, with a difference in vertical axis scaling and offset. [Lundberg, 2002] ENOB is the number of bits in a quantizer that would produce the exact same value of SINAD in a system with quantization as the entire source of error. [LeRoy Corporation, 2011]

The derivation of the ENOB equation is laid out below for an ideal ADC. The maximum error of an ideal converter when digitization is ±½LSB. The quantization error for any AC signal that spans more than a few LSBs can be approximated by an uncorrelated sawtooth waveform having a peak-to-peak amplitude of *q*, the weight of an LSB. This error is shown in Figure 3. [Kester, 2009]

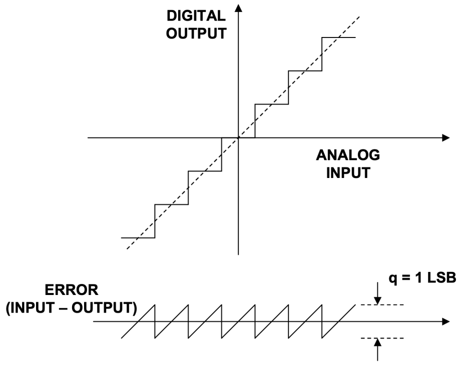


**Figure 2.6.2** Ideal ADC Quantization Noise [Kester, 2009]

The sawtooth error is calculated in Equation 6.

$$e(t) \; = \; st, \; -\frac{q}{2s} \; < \; t \; < \; +\frac{q}{2s}, \textit{ where}$$

*s = Slope of the quantization noise as a function of time*

**[Eq. 2.6.8]**

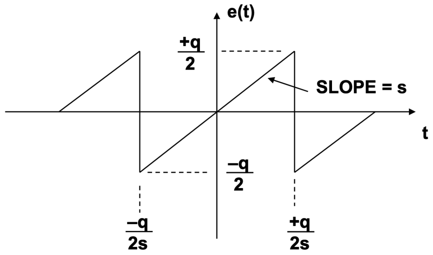


**Figure 2.6.4** Quantization Noise as a Function of Time [Kester, 2009]

The mean square value of e(t):

$$\overline{e^2(t)} \; = \; \frac{s}{q} \int_{-q/2s}^{+q/2s} (st)^2 dt$$

**[Eq. 2.6.9]**

Simple Integration and Simplifying:

$$\overline{e^2(t)} \; = \; \frac{q^2}{12}$$

**[Eq. 2.6.10]**

Root Mean Square Quantization Error:

$$RMS\ Quantization\ Noise \; = \; \sqrt{\overline{e^2(t)}} \; = \; \frac{q}{\sqrt{12}}$$

**[Eq. 2.6.11]**

While they need to alias back into the Nyquist bandwidth and produce an rms noise equal to $q/\sqrt{12}$, this waveform forms harmonics extend past the Nyquist bandwidth of dc to $f_s/2$.

The quantization noise is roughly Gaussian and distributed approximately uniformly across the Nyquist bandwidth from DC to fs/2, as pointed out by W. R. Bennett of Bell Laboratories. The fundamental premise is that the quantization noise is incidental to the input signal. When the sampling clock and signal are harmonically related, the quantization noise is correlated and the energy is concentrated in the signal's harmonics, but the RMS value remains approximately q/12. Assuming a full-scale sine wave, it is possible to calculate the theoretical SNR at this point:

Input Full-Scale Sine Wave:

$$Input\ Sine\ Wave\ =\ v(t)\ =\ \frac{q2^N}{2} sin(2\pi ft),\ where$$
$$N\ =\ number\ of\ Bits\ in\ the\ ADC$$

**[Eq. 2.6.12]**

RMS Value of the Input Signal:

$$rms\ value\ of\ input\ =\ \frac{q2^N}{2\sqrt{2}}$$

**[Eq. 2.6.13]**

RMS Signal-to-Noise and Distortion Ratio for an ideal N-Bit Converter:

$$SINAD\ =\ 20log_{10} \frac{rms\ value\ of\ FS\ input}{rms\ value\ of\ quantization\ noise}$$

**[Eq. 2.6.14]**

$$SINAD\ =\ 20log_{10}\left[\frac{q2^N/2\sqrt{2}}{q/\sqrt{12}}\right]\ =\ 20log_{10}2^b\ +\ 20log_{10}\sqrt{\frac{3}{2}}$$

**[Eq. 2.6.15]**

Therefore, the SINAD equation would be equal to Eq. 2.6.2, and the ENOB Equation would be equal to Eq. 2.6.6.

2.6.4 Quantifying and Analyzing Noise Distortion

Figure 2 shows the setup for the analysis, which includes a Discrete Fourier Transform (DFT) analysis to quantify noise distortion.

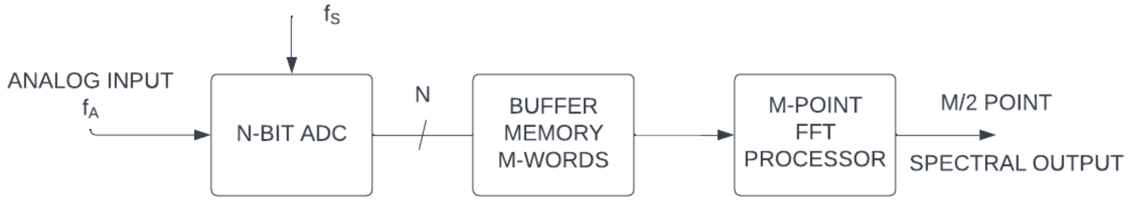


**Figure 2.6.3** Generalized Test Setup for DFT Analysis of ADC Output [Modified From: Kester, 2009]

The DFT analyzes a spectrum of M/2 points in the frequency domain with a total frequency range of dc to $f_s/2$, where M denotes the number of samples used for the Discrete Fourier Transform and $f_s$ denotes the sampling rate. $f_s/M$ accounts for the resolution of the DFT, and the theoretical noise floor is equal to theoretical SNR plus the DFT process gain, 10*log(M/2). The noise used for the SNR calculation is a noise that extends over the whole Nyquist bandwidth (dc to $f_s/2$) - generally, a sinusoid, - and the DFT takes the role of a narrowband spectrum analyzer with the $f_s/M$ bandwidth. [Kester, 2009]

In addition to measuring the amplitude of the harmonic component of the digitized signal, the DFT output can also be used to determine the amplitude of the harmonics and noise components. The harmonics can be distinguished from the other components due to their location in the spectrum. [Kester, 2009]

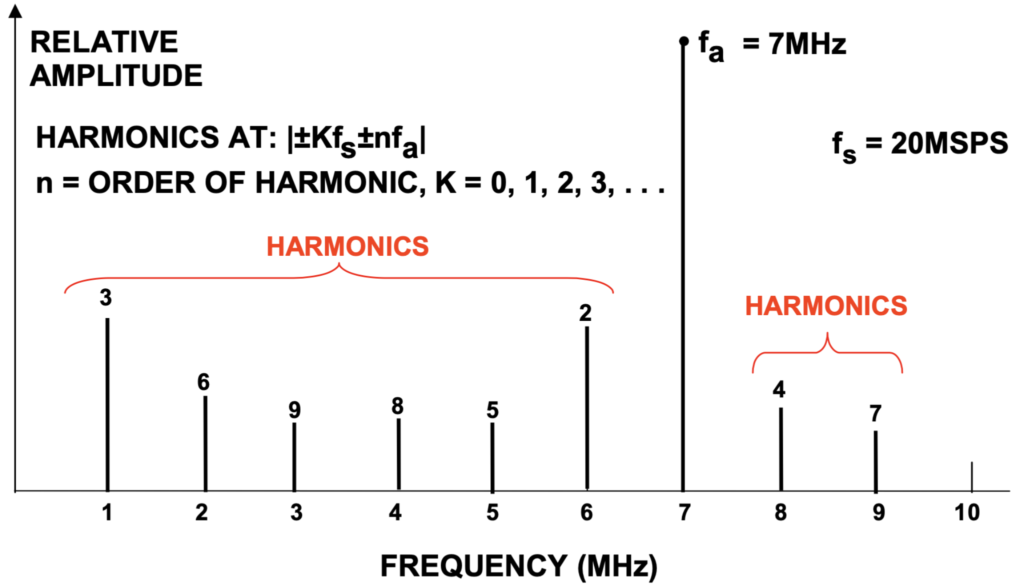


**Figure 2.6.4** Harmonics in the Spectrum

2.6.4.1 SINAD Calculation using DFT

In an M-point DFT of a sine wave, with the fundamental signal being in frequency bin M and having an amplitude $A_m$, the SINAD can be calculated from the DFT Amplitudes, as shown in Equation 2.6.4.1.1:

$$SINAD = 10log\left[A_m^2 \left(\sum_{k=1}^{m-1} A_k^2 + \sum_{k=m+1}^{M/2} A_k^2\right)^{-1}\right]$$

**[Eq. 2.6.16]**

Usually, several bins around the fundamental frequency are ignored to avoid any spectral leakage. SINAD is dependent on the input signal frequency and amplitude, and it degrades at high frequency and power. Results should be represented in plots of SINAD vs frequency for a constant amplitude input or SINAD vs amplitude of a constant-frequency input. [Lundberg, 2002]

*2.6.5 Increasing ADC Resolution*

Even though ADCs mostly offer a specific number of bits of resolution, there are several methods to enhance it: oversampling, averaging, and decimation. Figure 1 shows the plots for 10-bit resolution, as well as enhanced resolutions of 11 and 12 bits.

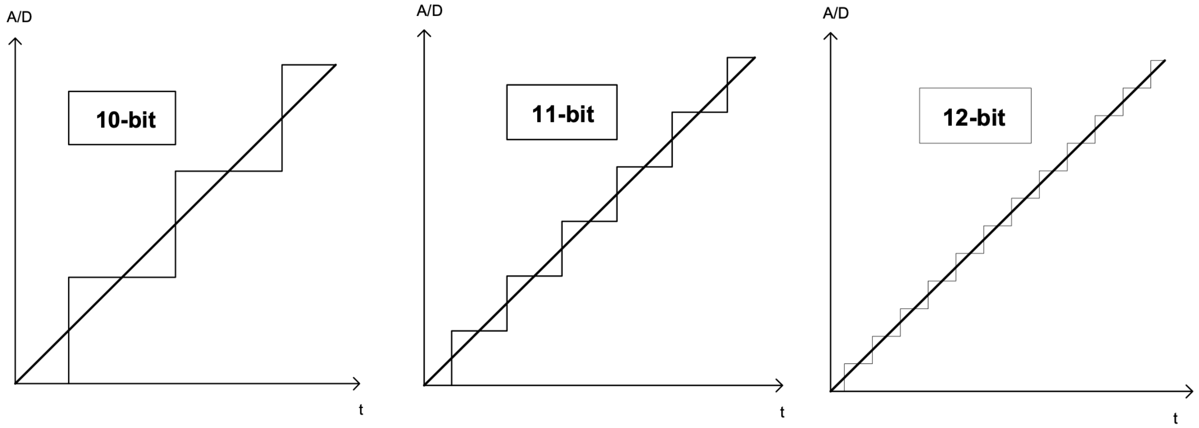


*Figure 2.6.5-a.* 10-Bit Resolution Plot

*Figure 2.6.5-b*. 11-Bit Resolution Plot

*Figure 2.6.5-c.* 12-Bit Resolution Plot

**Figure 2.6.5** Demonstration of original vs enhanced resolutions [Atmel Corporation, 2005]

2.6.5.1 Oversampling and Decimation

Sampling rate is the frequency at which the ADC samples the input analog signal, in Hertz. A signal has to be sampled twice as fast as the bandwidth to accurately construct the waveform, according to the Nyquist Theorem. If this is not achieved, there exists a folded-frequency, which causes the signal to be aliased. Twice of the signal frequency, which is the minimum, is called the Nyquist Frequency. *[Atmel Corporation, 2005]* As the change in signal gets faster, the higher the sampling rate should be.

$$f_{nyquist} > 2 * f_{signal}$$

**[Eq. 2.6.17]**

Sampling at frequency rates higher than the Nyquist Frequency is called oversampling. To achieve each additional bit of resolution, n, there is a need to oversample the signal four times. *[Atmel Corporation, 2005]*

$$f_{oversampling} = 4^{n} * f_{nyquist}$$

**[Eq. 2.6.18]**

There are some constraints to this method *[Atmel Corporation, 2005]*:

- The signal component should not be significantly volatile during a conversion.
- Noise should be present, but independent from the signal.
- The amplitude of this noise should be at least equal to 1LSB.

This method not only increases the resolution but also enhances the SINAD, thus enhancing ENOB as well. The process of increasing the resolution from 10-bit to 11-bit is shown in Figure 6.

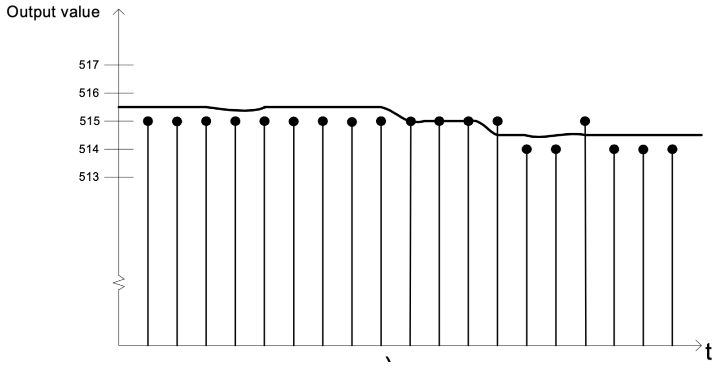


*Figure 2.6.6-a*

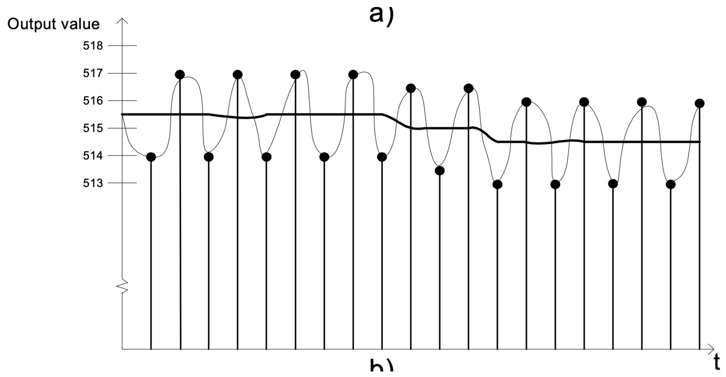


*Figure 2.6.6-b*

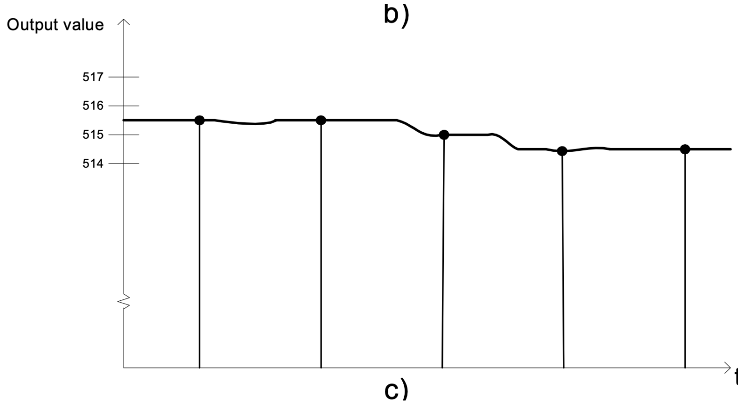


*Figure 2.6.6-c*

**Figure 2.6.6** Enhancing Resolution through Oversampling [Atmel Corporation, 2005]

Oversampling and decimation would work as long as the noise in the signal is sufficient to toggle the least significant bit of the ADC. If the noise does not exist, an artificial Gaussian or a periodical noise might be added to the signal, which is called dithering. The period of the noise signal should not be more than the period of the m samples. The amplitude should be at least 1 LSB, and the mean value of the noise must be zero. Otherwise, there might be an offset, which is shown in Figure 2.6.7.

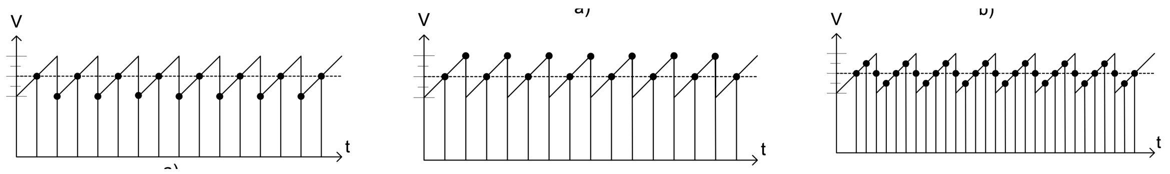


*Figure 2.6.7-a* *Figure 2.6.7-b* *Figure 2.6.7-c*

**Figure 2.6.7** Offset by Inadequate Sampling [Atmel Corporation, 2005

The first plot in Figure 2.6.7 generates a negative offset, and the second plot causes a positive one. In the third plot, since sampling is adequate and sufficient, there is no offset.

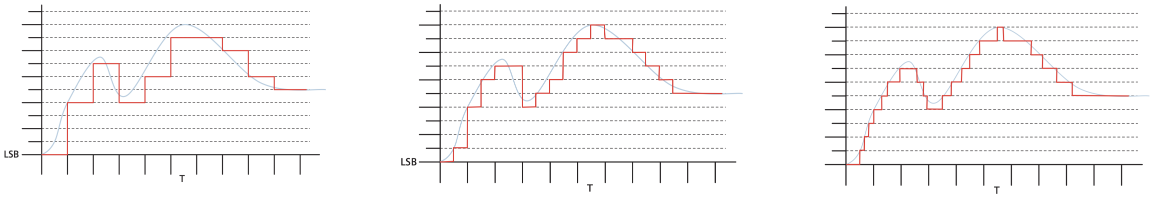


*Figure 2.6.8-a* Input Waveform Sampled at 1x Rate

*Figure 2.6.8-b* Input Waveform Sampled at 2x Rate

*Figure 2.6.8-c* Input Waveform Sampled at 8x Rate

**Figure 2.6.8** Enhanced Resolution by Oversampling [Actel Corporation, 2007]

ADC resolution could be improved through oversampling in bandlimited signals, which basically increases the sampling rate. Oversampling refers to sampling of the signal at a rate significantly higher than the Nyquist frequency. Despite the fact that oversampling does not directly provide impact to the ADC resolution, it assists in more accurate tracking of the input signal by utilizing the existing ADC dynamic range in a more effective manner, as shown in Figure 8. [Actel Corporation, 2007]

Increasing sampling rate further than that would cause multiples of the same result for each value, while not providing any visible improvement. At the same time, with oversampling being easy to implement, it requires a complex circuit and high power. [Zheng et al., 2021] Changes in system clock rate, sample acquisition times and the sampling sequence, sample intervals could independently be managed for each analog input. Even though oversampling suffices for most applications, further processing could be implemented to improve the digital representation of the signal. [Actel Corporation, 2007]

### *2.7 Pong*

Games are often incorporated into electromyography applications for muscle training, prosthetic adjustment or solely entertainment purposes. Research conducted using Nintendo Wii Fit Balance Board as the main tool has shown that physical and cognitive skills could be improved through games. [Lek et al., 2017]

The game of Pong is a table tennis, where the user controls a ball using a bat (shown in Figure 1). It is one of the games that require muscle movement and could be used in EMG applications. The game requires the player to move the bat around to block the opponent’s ball from entering the scoring region. [Lek et al., 2017]

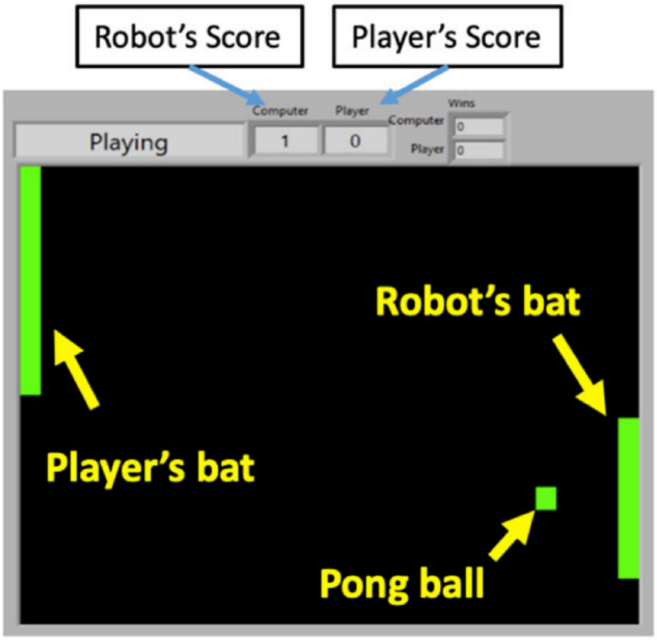


**Figure 2.7.1** Pong Game [Hau et al., 2021]

An EMG application of a Pong game is shown in Figure 2. The two bats are moved around to keep the opponent striking a goal. One of the bats are controlled by the computer while the other one is controlled by the player. The Figure also demonstrates the voltage changes on the two muscles used for the study. The flow chart for the application is shown in Figure 3.

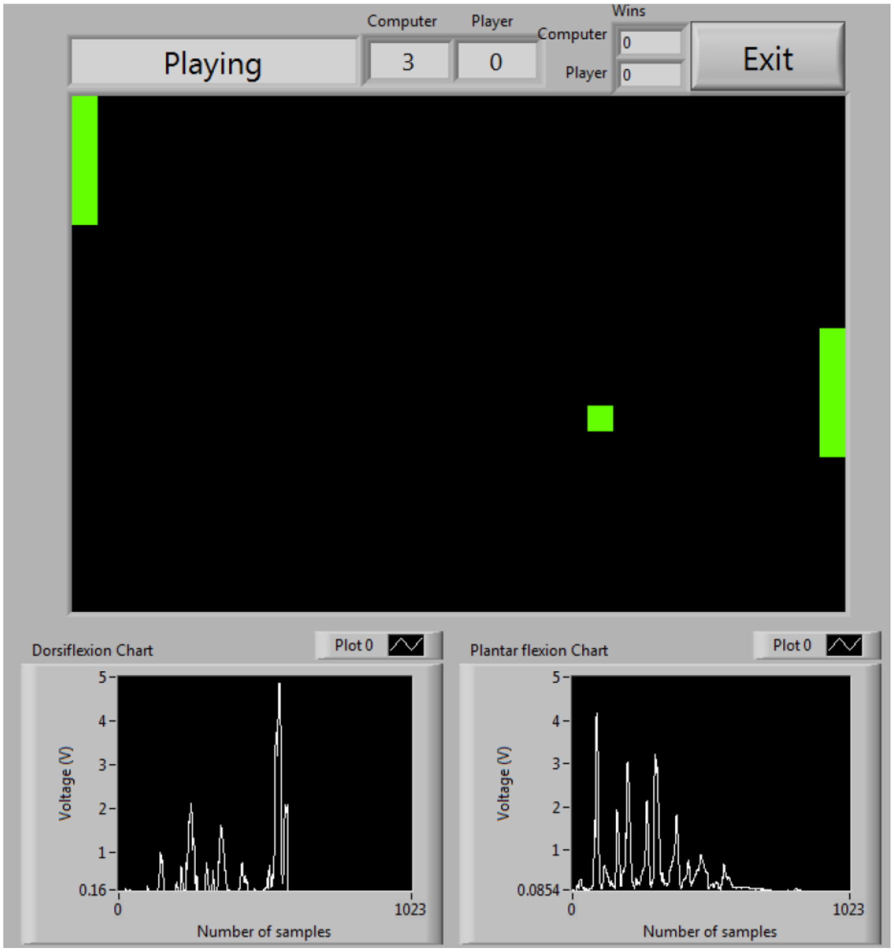


**Figure 2.7.2** Pong Game with EMG Application [Lek et al., 2017]

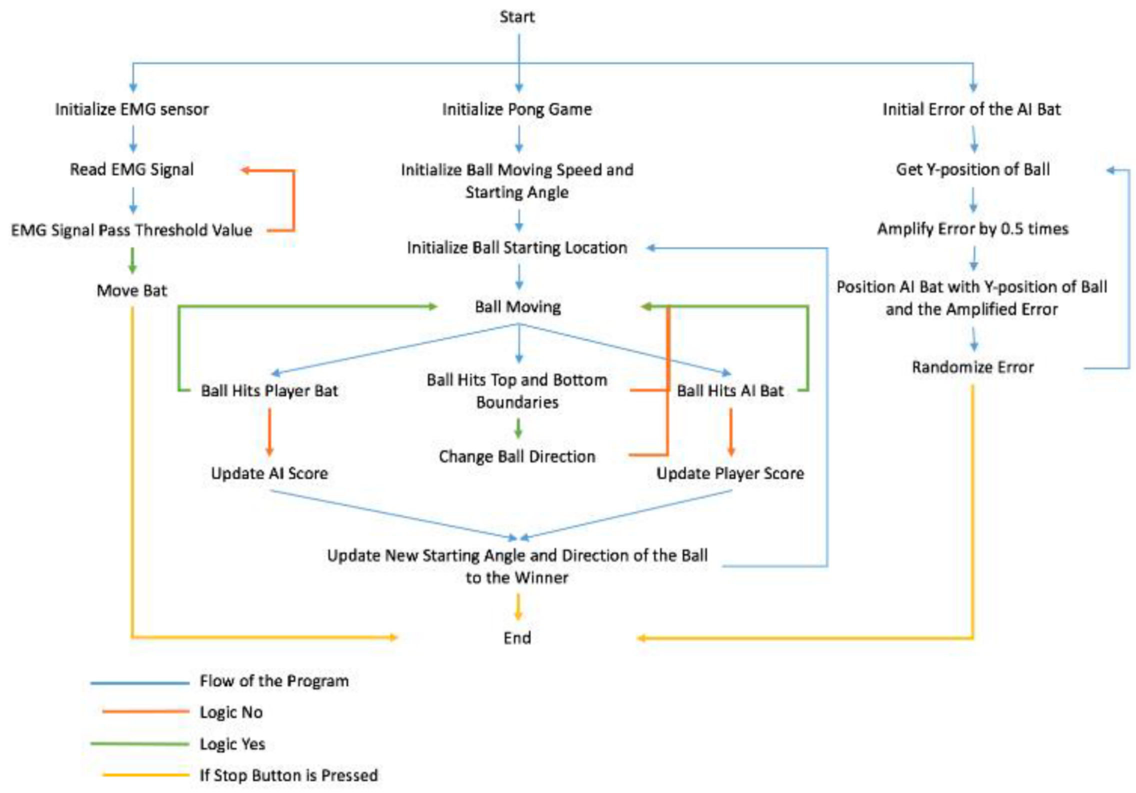
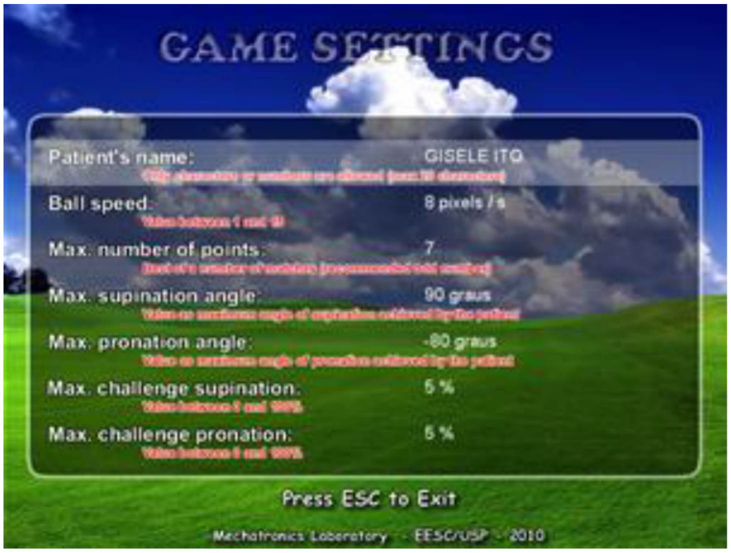

Start
Initialize EMG sensor
Initialize Pong Game
Initial Error of the AI Bat
Read EMG Signal
Initialize Ball Moving Speed and Starting Angle
Get Y-position of Ball
EMG Signal Pass Threshold Value
Amplify Error by 0.5 times
Initialize Ball Starting Location
Position AI Bat with Y-position of Ball and the Amplified Error
Move Bat
Ball Moving
Randomize Error
Ball Hits Player Bat
Ball Hits Top and Bottom Boundaries
Ball Hits AI Bat
Change Ball Direction
Update AI Score
Update Player Score
Update New Starting Angle and Direction of the Ball to the Winner
End
Flow of the Program
Logic No
Logic Yes
If Stop Button is Pressed

**Figure 2.7.3** Software Architecture of EMG based Pong Game [Lek et al., 2017]

The Pong game could be used in many modular places in the body, including ankles, wrists, elbows. When the player applies force to the plate on the system by contracting their muscles, which allows EMG to control the paddle, and the applied force exceeds the threshold, the bat is moved down. If the opposite happens, the bat will be moved up.

A study conducted by Andrade et al. has shown that Pong could be used for rehabilitation purposes after distal radius fractures on the arms. It was aimed to be used in rehabilitation centers by physicians and therapists; therefore, the interface was created for a business environment rather than individual use.



GAME SETTINGS

Patient's name: GISELE ITO
[illegible]

Ball speed: 8 pixels / s
Values between 1 and 10

Max. number of points: 7
[illegible]

Max. supination angle: 90 graus
Value as maximum angle of supination achieved by the patient

Max. pronation angle: -80 graus
Value as maximum angle of pronation achieved by the patient

Max. challenge supination: 5 %
Values between 0 and 100%

Max. challenge pronation: 5 %
Values between 0 and 100%

Press ESC to Exit

Mechatronics Laboratory - EESC/USP - 2010

**Figure 2.7.4** Software Interface for the EMG Pong Game to be used at Rehabilitation Centers [Andrade et al., 2010]

The system provided feedback through the difference between pronation and supination in the muscles, which proved to be effective over time. As the difference incidences increased, the more the muscles were effectively moving. The improvement in the patient's movement abilities are shown in Figure 6.

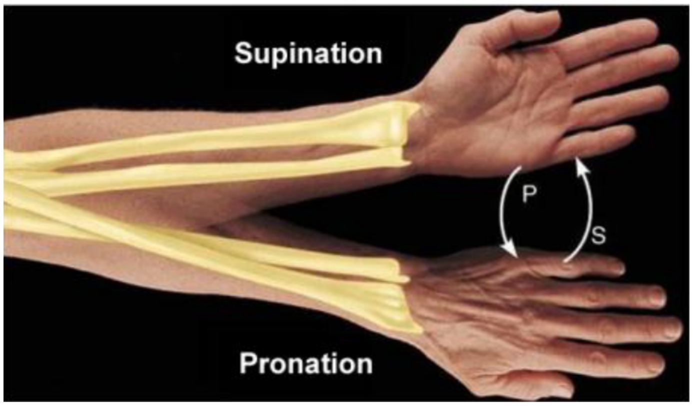


**Figure 2.7.5** Pronation and Supination of the Arm [Andrade et al., 2010]

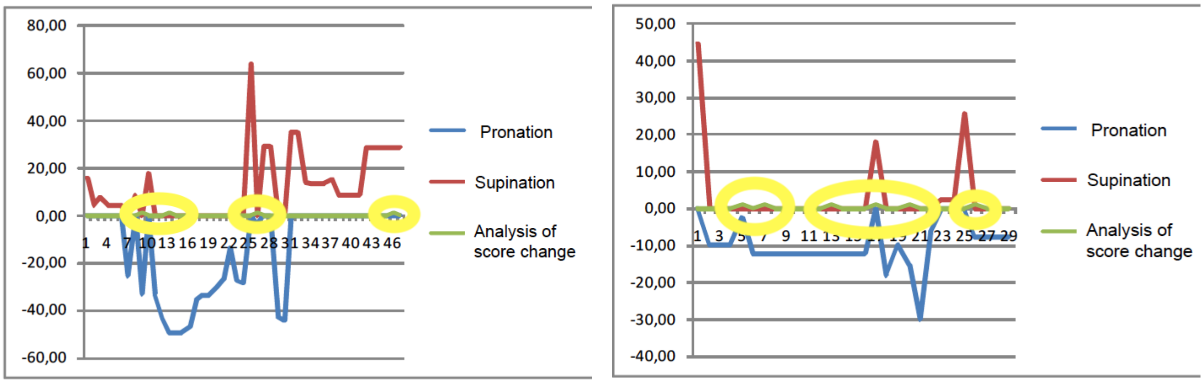


*Figure 2.6.6-a*

*Figure 2.6.6-b*

**Figure 2.6.6** Changes in Difference Between the Pronation and Supination of Muscles while playing the Pong Game in 2 Different Sessions [Andrade et al., 2010]

# 3. System Design

## *3.1 Module Definitions*

The system is broken down into three modules as shown in Figure 3.1.1. The analog front end transduces and processes the raw sEMG signal. The processed signal is then input into the controllers which digitize, transmit, receive, and send the signal out to the application. The digitized signal is then further processed and used to control the pong application. Each module is further broken down into elements as shown in Figures 4.1.1, 5.2.4, and 6.1.1.

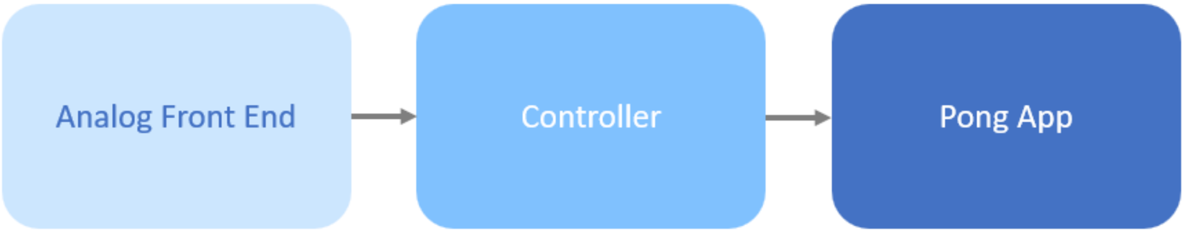


**Figure 3.1.1** Modules

## *3.2 Design Options*

The design options for each individual module can be found in sections 4.2, 5.2, and 6.2. Due to the time constraints of this project, only two overall system designs were considered. However, there are a variety of system design options that are possible and should be considered in future iterations.

### *3.2.1 Phase I Design Options*

The Phase I system will utilize the design decisions highlighted in Figure 3.2.1, 3.2.2, and 3.3.3 and visualized in Figure 3.2.4.

#### 3.2.1.1 Analog Front End

A Velcro armband placed on the forearm was chosen because the screw-based electrodes can easily be integrated into the material. In addition, the material makes the system easy to don and doff which meets the user requirements discussed in section 1.1. For the controller and front end housing, the armband was chosen because the system will be easy to don and doff and also remain close to the electrodes, satisfying both the user and product requirements discussed in sections 1.1 and 1.2. A solder board was chosen for the analog front end because it decreased the size and robustness of the system in comparison to a breadboard but did not come with the lead times of a PCB. The ICs chosen were based on functionality, cost, and availability. The AD620 is an instrumentation amplifier that complies with the voltage range our system operates within. It is an older part; however, there were extra of these components already available to us in the lab from prior MQP teams, so only a few needed to be ordered.

The AD620 is also a relatively cheap part at roughly $23.00 per unit from distributors like Digikey and Mouser. The LM348N was chosen because it is a quad-operational amplifier, which saves space in the design. Only one quad operational amplifier IC was needed per channel, rather than four individual operational amplifier ICs per channel. The LM348N is also very low cost at $0.61 from distributors like Digikey and Mouser. The voltage regulator was chosen based on availability in the lab and desired output voltage. The LM317BT voltage regulator was used by previous MQP teams, so our team already had some available to us. This voltage regulator was also adjusTable and allowed us to set the output voltage to the desired 1.8 V.

### 3.2.1.2 Controller

The nRF52840-DK was chosen as both the peripheral and central board because the software development kits (SDKs) and boards have previously been used in the lab. Therefore, the graduate students within the lab were well versed on the boards and were available if any questions arose.

### 3.2.1.3 Pong App

MATLAB was chosen as the programming language and environment for the pong application because MATLAB code to display the signal output of the central board had previously been developed by the graduate students within the lab. Similarly, a PC based app was chosen because the code developed by the graduate students was PC-based. One-player pong was chosen so that the user could adjust the settings of the computer-based player to fit their rehabilitation needs. Due to time constraints, only a demonstrable version of the app was developed over a fully-functional app.

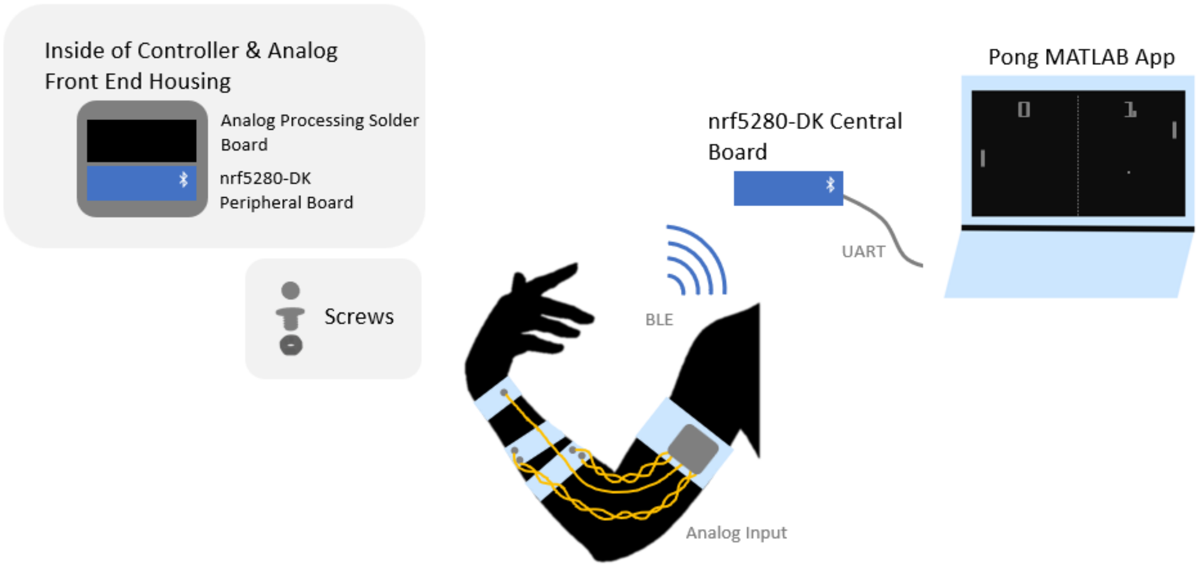

Inside of Controller & Analog Front End Housing
Analog Processing Solder Board
nrf5280-DK Peripheral Board
Screws
nrf5280-DK Central Board
Pong MATLAB App
0
1
UART
BLE
Analog Input

**Figure 3.2.4** Phase I design

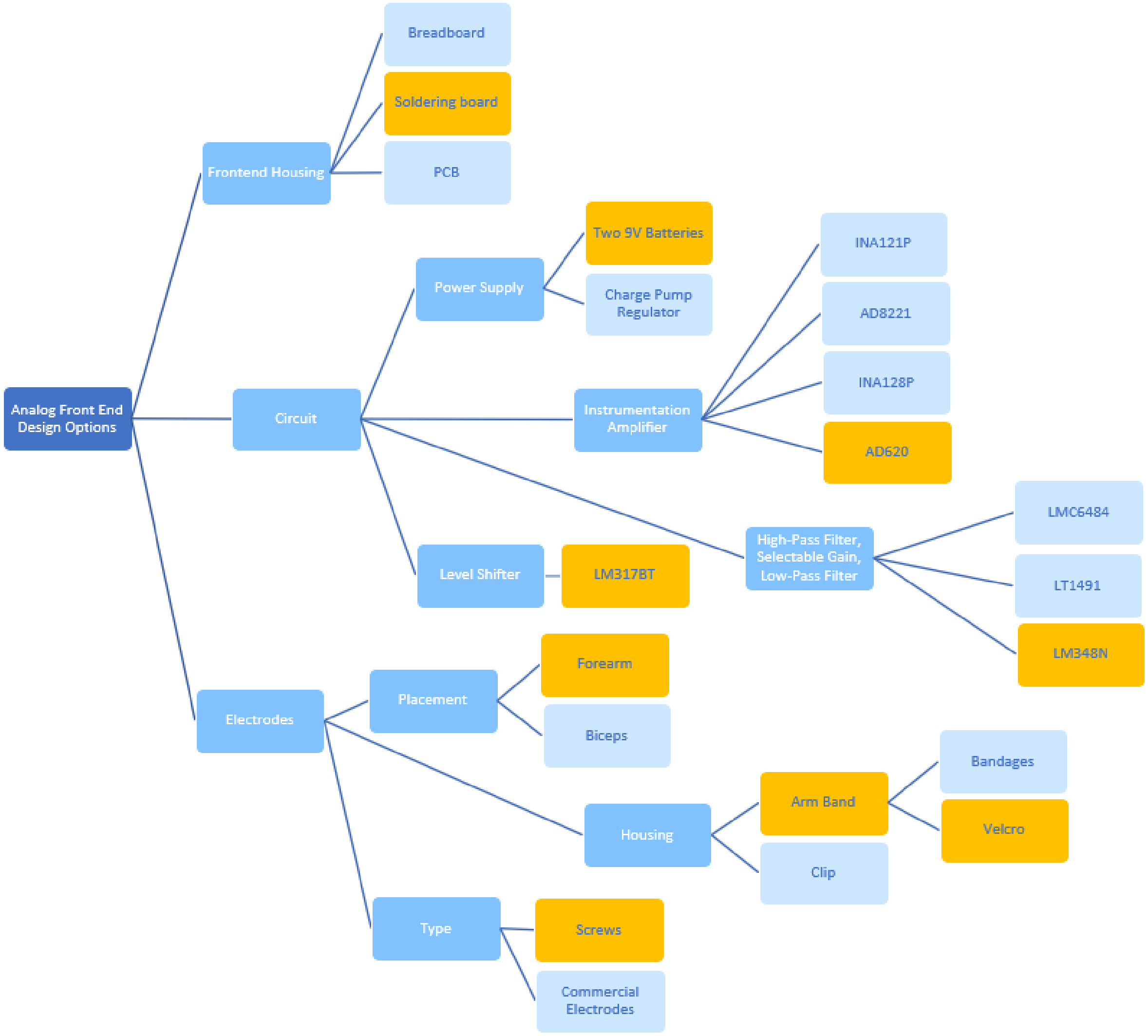


**Figure 3.2.1** Phase I analog front end design decisions

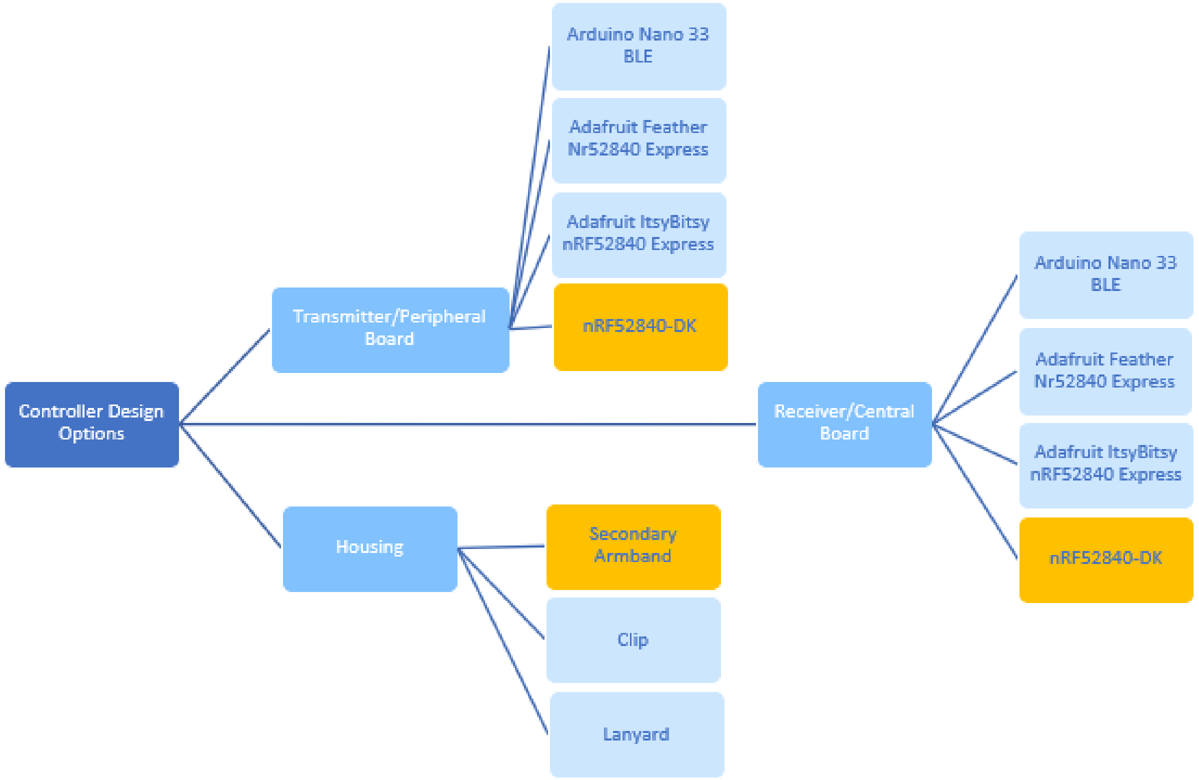

Controller Design Options
Transmitter/Peripheral Board
Arduino Nano 33 BLE
Adafruit Feather Nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Receiver/Central Board
Arduino Nano 33 BLE
Adafruit Feather Nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Housing
Secondary Armband
Clip
Lanyard

**Figure 3.2.2** Phase I controller design decisions

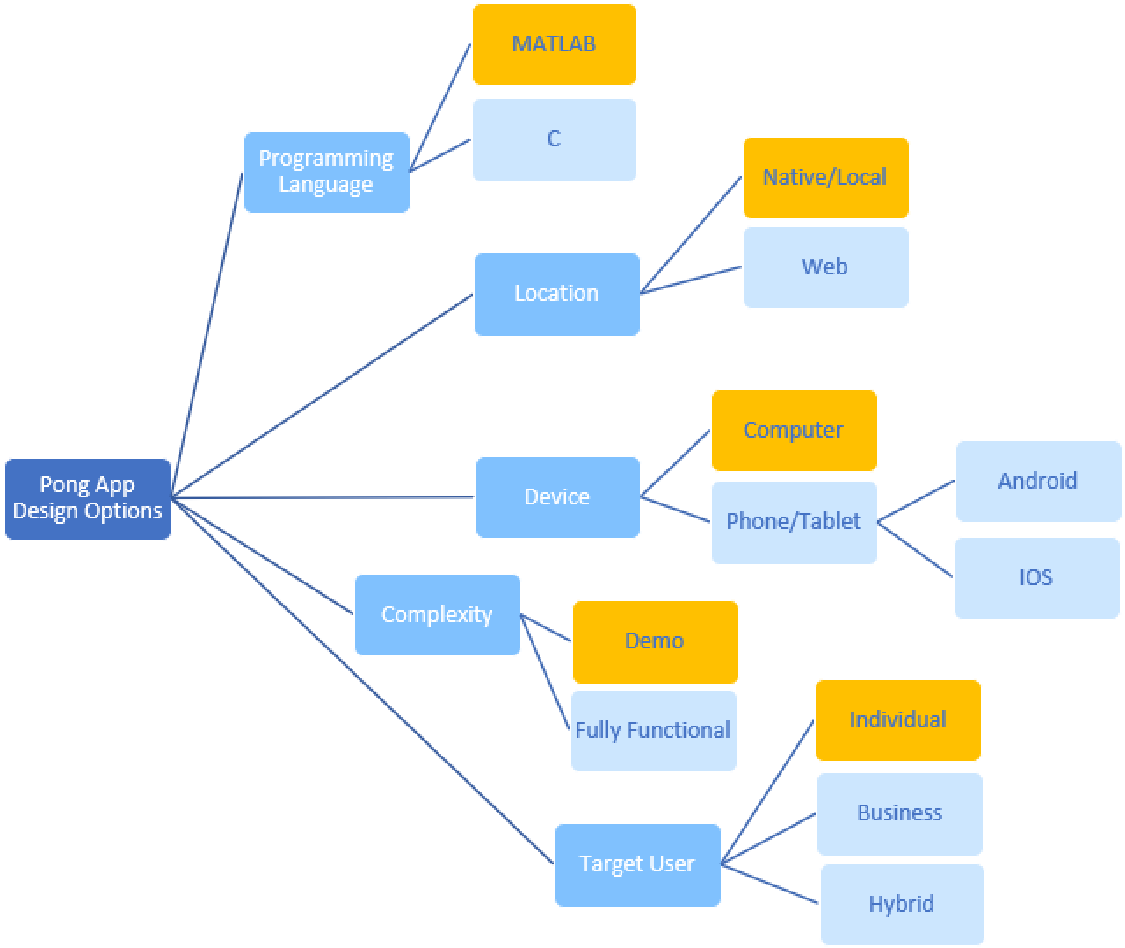


**Figure 3.2.3** Phase I pong app design decisions

*3.2.2 Phase II Design Options*

The Phase I system will utilize the design decisions highlighted in Figure 3.2.5, 3.2.6, and 3.3.7 and visualized in Figure 3.2.8. The design decisions are similar to that of phase I with slight adjustments to decrease the footprint size of the system and increase its efficiency and usability. For the analog front end housing, a PCB board was chosen instead of a solder board to decrease the system's footprint. Similarly, the Adafruit Feather nr52840 Express boards were chosen instead of the nRF52840-DKs. For the app complexity, a fully functional app was chosen to be developed over a demonstrable app so that future developers can easily understand and adapt the code to fit their needs.

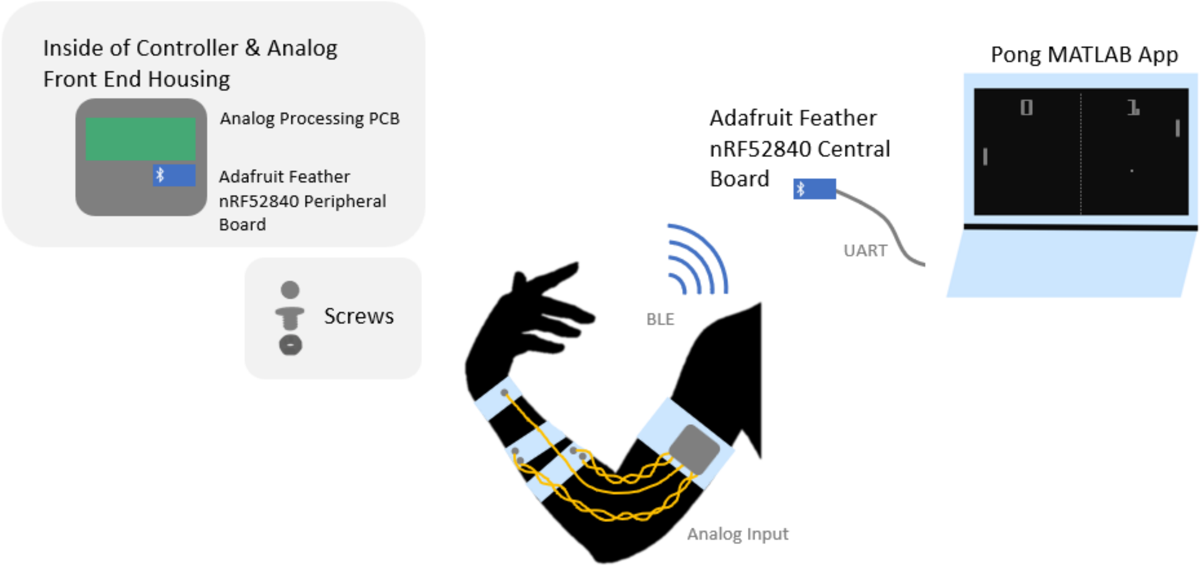

Inside of Controller & Analog Front End Housing
Analog Processing PCB
Adafruit Feather nRF52840 Peripheral Board
Screws
BLE
Analog Input
Adafruit Feather nRF52840 Central Board
UART
Pong MATLAB App
0
1

**Figure 3.2.8** Phase II design

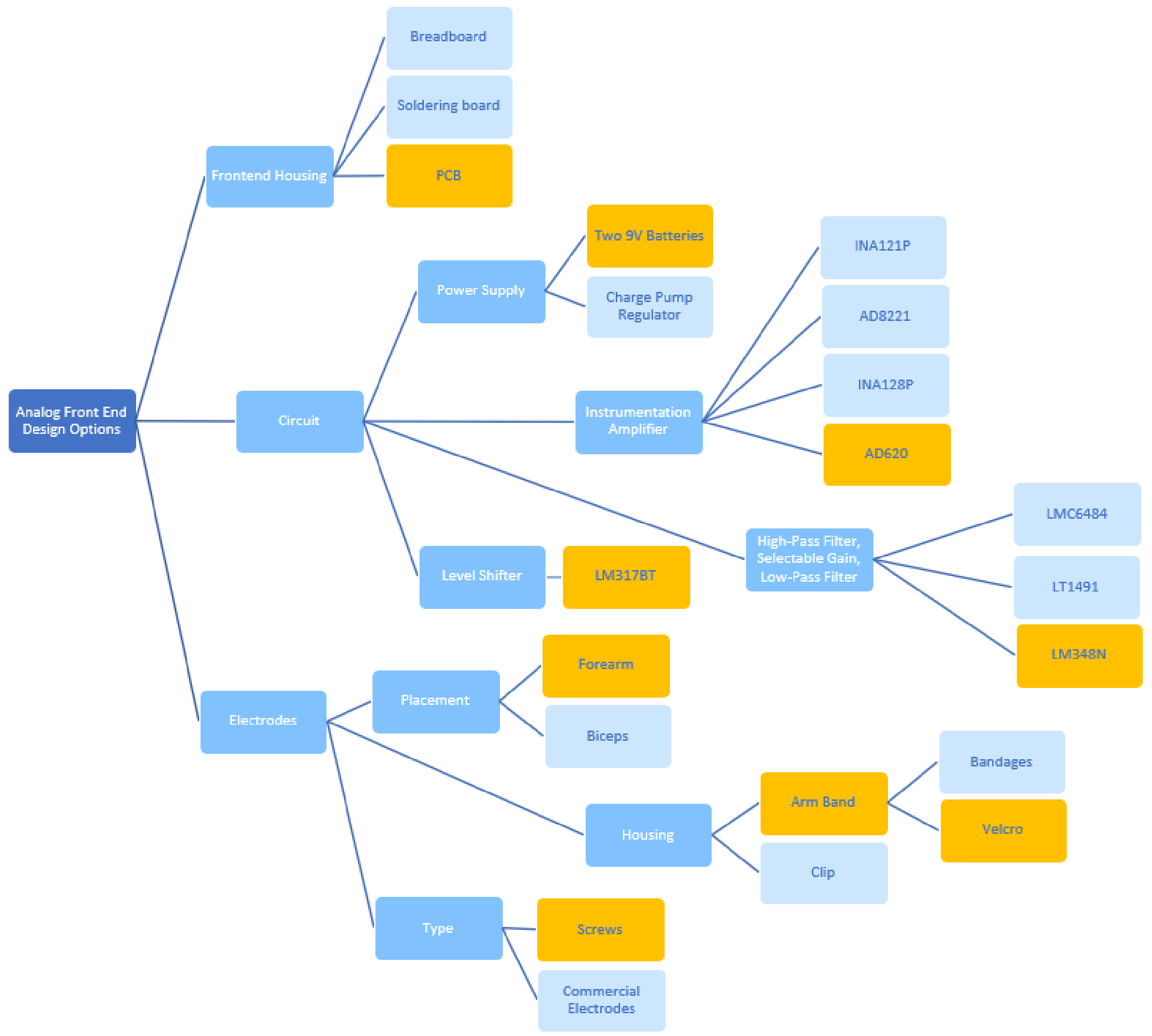


**Figure 3.2.5** Phase II analog front end design decisions

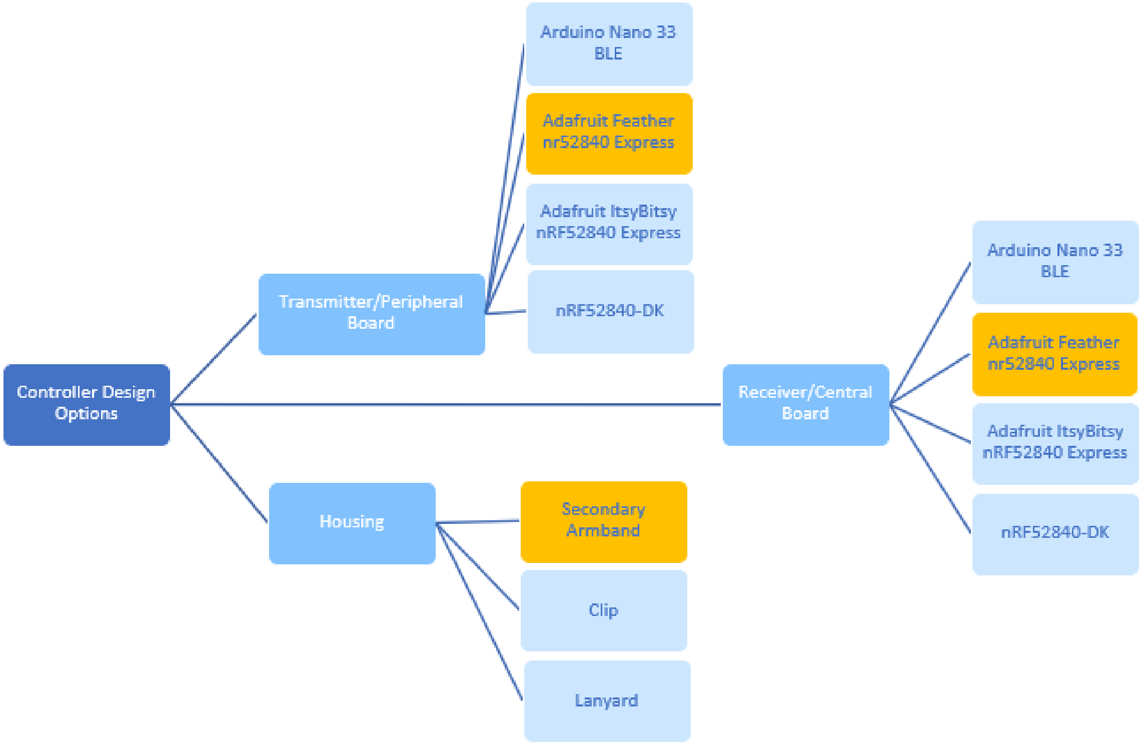

Controller Design Options
Transmitter/Peripheral Board
Arduino Nano 33 BLE
Adafruit Feather nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Receiver/Central Board
Arduino Nano 33 BLE
Adafruit Feather nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Housing
Secondary Armband
Clip
Lanyard

**Figure 3.2.6** Phase II controller design decisions

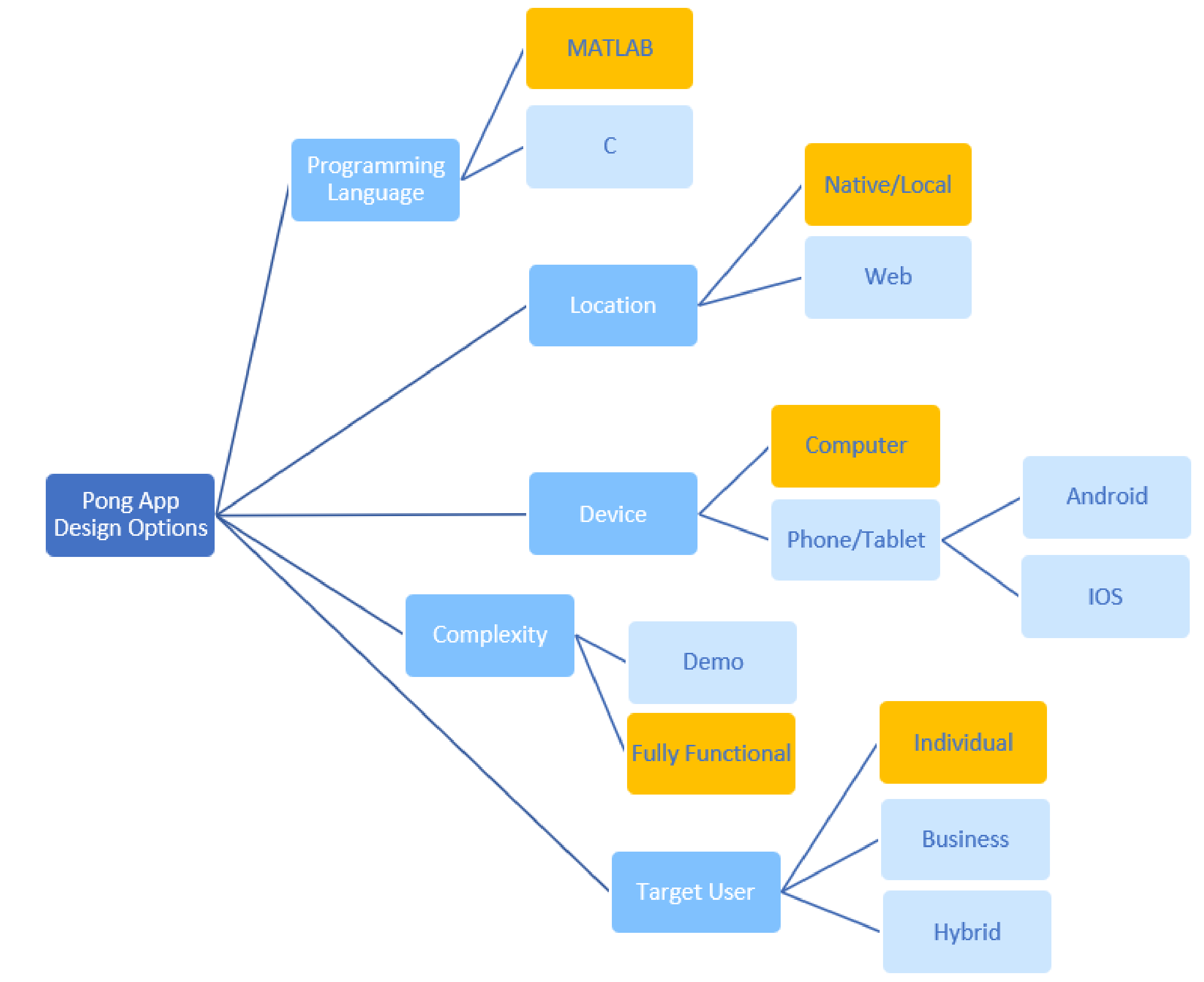


**Figure 3.2.7** Phase I pong app design decisions

### *3.3 Final Design*

The two system design options considered were phase I and phase II designs. The phase I design option was intended as a proof of concept that met the minimum user and product requirements as described in chapters 1.1 and 1.2, respectively. The phase II design was a reach design that was intended to decrease the system's footprint size and increase the system's usability and efficiency. For this project, only the phase I design was implemented in the final design. However, future projects should consider the phase II design as an option.

# 4. Analog Front End

### *4.1 Introduction*

In order to utilize the EMG signal, it must go through an amplification and filtering process. The EMG signal must be amplified, band-pass filtered to remove unwanted frequencies that are outside of the EMG frequency range, and level shifted to provide only positive values to the unipolar ADC.

The analog front end circuitry requires the following stages as seen in Figure 4.1.1:

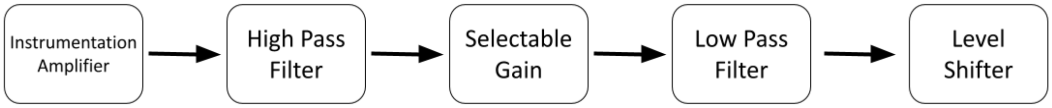


Figure 4.1.1 Analog Front End Flow Chart

### *4.2 Design*

Each analog front end design component is shown below in Figure 4.2.0. The options for each dark blue design element will be discussed in the following subsection.

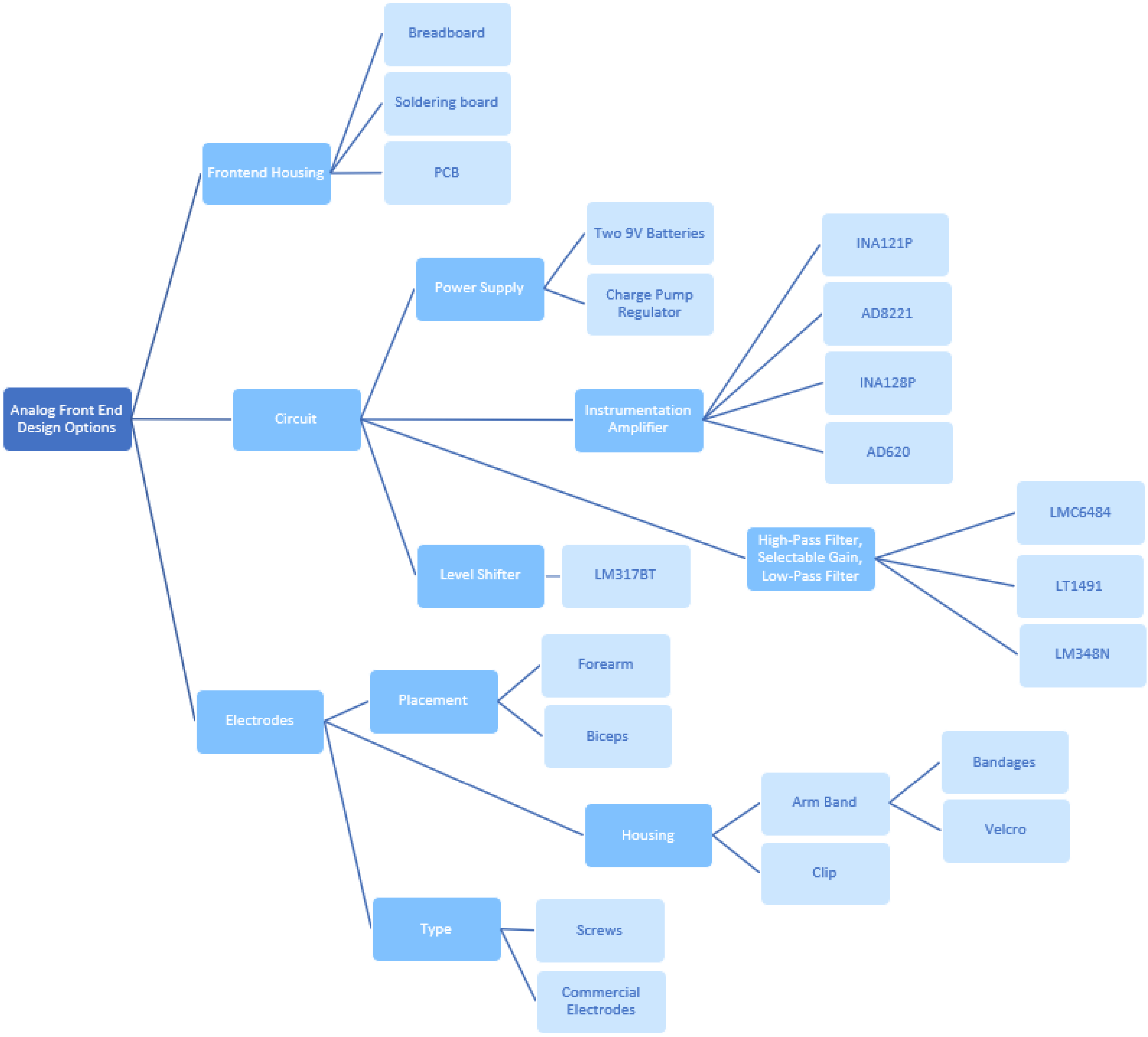


**Figure 4.2.0** Analog Front End Design Options

#### *4.2.1 Front End Housing Design Options*

The analog front end circuitry could be constructed on either a breadboard, solder board, or printed circuit board (PCB). The easiest option given the time constraints of this project is beginning with a breadboard setup for ease of testing and proof of concept. Once the circuit functionality is proven, the solder board can be constructed which would help shrink down the size of the entire front end circuit. The final revision of the front end would be a PCB of small enough size that small, surface-mount components can be used and the circuit could be worn on the body, either on a lanyard or armband.

Below is a table discussing the pros and cons of each option:

| Breadboard | | Solder Board | | PCB | |
|---|---|---|---|---|---|
| Pros | Cons | Pros | Cons | Pros | Cons |
| Easiest to construct | Large | More compact than breadboard | Not wearable by user | Small | Time consuming to design |
| Easy for debugging | Not wearable by user | Good soldering experience | N/A | Could be worn by the user in armband | Would need to order, long lead time |
| No need to purchase any additional items | N/A | N/A | N/A | Good design experience | Extra expense |

**Table 4.2.1** Analog Front End Housing Comparison

### *4.2.2 Front End Circuit Design Options*

#### 4.2.2.1 Power Supply

The front end circuit requires a minimum of 4.3 V to operate effectively. The ADC that is used in this project runs on a voltage range of 0-3.6 V, and a 0.7 V buffer is added on top of the required 3.6 V to account for the output voltage range of the operational amplifiers.  There are two options for providing power to the front end circuit:

1. *Two batteries wired in series with a ground reference in between*
   When using this approach, the simplest option would be using two 9V batteries wired in series with the ground reference in between. This would provide the necessary dual voltage power supply (“plus” battery voltage and “minus” battery voltage) that the front end circuit requires. A dual supply of +/- 9 V is more than the required + 4.3 V, but using only two 9 V batteries is simpler than requiring four 3 V batteries (which would instead provide +/- 6 V). In this case, the microcontroller that is not part of the front end circuit would need to be powered by its own battery because it cannot be powered by +9 V.

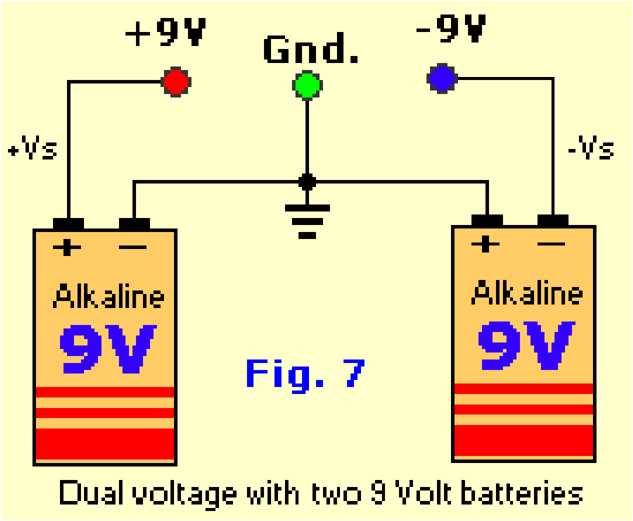


**Figure 4.2.1** Dual Voltage Power Supply [*electronics.stackexchange.com,* 2022]

The two 9 V batteries would be placed next to the analog front end circuit. Wires would need to be soldered to the battery terminals to provide a path for the voltage to connect to the front end circuit. A 9 V battery holder with built-in wire leads could be used to prevent the need of soldering wires. An example battery holder with leads is the *9 V Cell Densi-Pak Holder with Wire Leads* from Keystone Electronics (Figure 4.2.2). The price of one holder is $2.06 from *Mouser.com.*

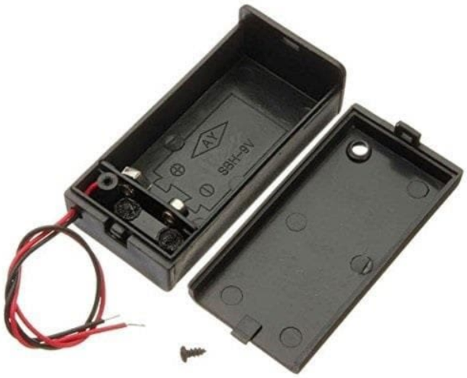


**Figure 4.2.2** Keystone Electronics 245 9V Battery Holder [*Mouser.com*, 2022]

2. *One battery and charge pump regulator*
   The second option is using one battery and a charge pump regulator. A charge pump regulator takes one input voltage and provides a dual-supply output voltage. This would allow our design to operate from one battery rather than two. However, the charge pump regulator would need to be purchased and integrated into the analog front end. Two charge pump regulator options are the MAX768 from Maxim Integrated and LM27762 from Texas Instruments.

**MAX768**

The MAX768 provides a dual power supply of +/- 5 V in Dual Mode™ operation: the output voltage is preset to +5 V and -5 V.

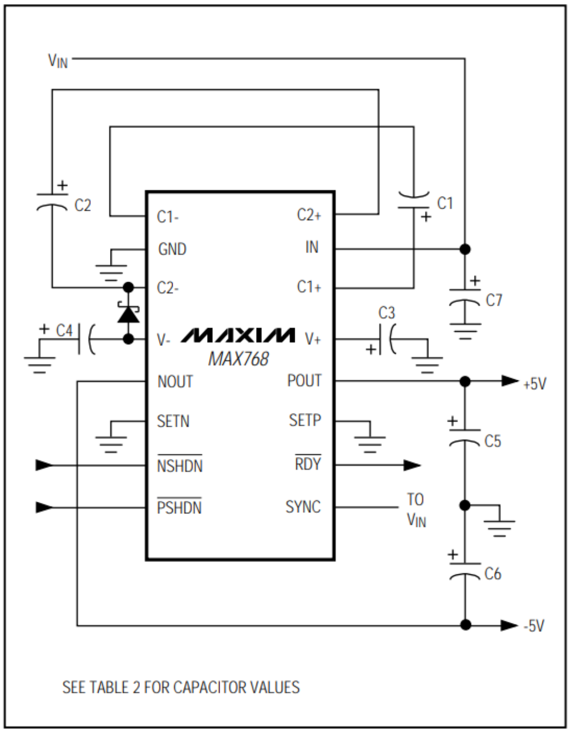
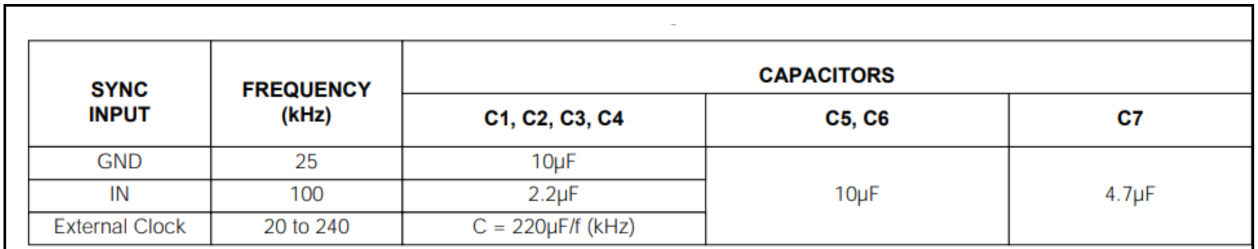


**Figure 4.2.3** MAX768 Standard Application Circuit [*Maxim Integrated*, 2022]

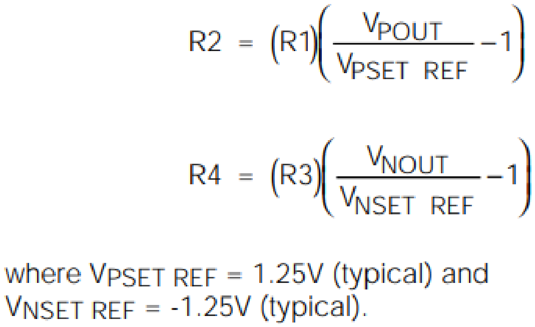
The MAX768 requires setting external capacitors to drive the chip to provide the typical 5 V dual-supply. These capacitors are set to values given in the following table:



| SYNC INPUT | FREQUENCY (kHz) | CAPACITORS | | |
|---|---|---|---|---|
| | | C1, C2, C3, C4 | C5, C6 | C7 |
| GND | 25 | 10µF | 10µF | 4.7µF |
| IN | 100 | 2.2µF | | |
| External Clock | 20 to 240 | C = 220µF/f (kHz) | | |

**Figure 4.2.4** External Capacitor Values MAX768 [*Maxim Integrated*, 2022]

The dual-supply output voltage can be adjusted from the preset +/- 5 V condition to an output voltage between +/-1.25 V and +/-11 V by the addition of external resistors R1, R2, R3, & R4 (see Figure 4.2.6). The resistors R1 and R3 must be selected between 100kΩ and 400kΩ. This adjustable output option would allow for setting the dual-supply output to the desired + 4.3 V range needed for this project. The following equations are used to determine the R2 and R4 resistor values for the desired output voltage (VPOUT and VNOUT):



$$R2 = (R1)\left(\frac{V_{POUT}}{V_{PSET\ REF}} - 1\right)$$

$$R4 = (R3)\left(\frac{V_{NOUT}}{V_{NSET\ REF}} - 1\right)$$

where $V_{PSET\ REF}$ = 1.25V (typical) and $V_{NSET\ REF}$ = -1.25V (typical).

**Figure 4.2.5** Resistor Equations for AdjusTable Output [*Maxim Integrated*, 2022]

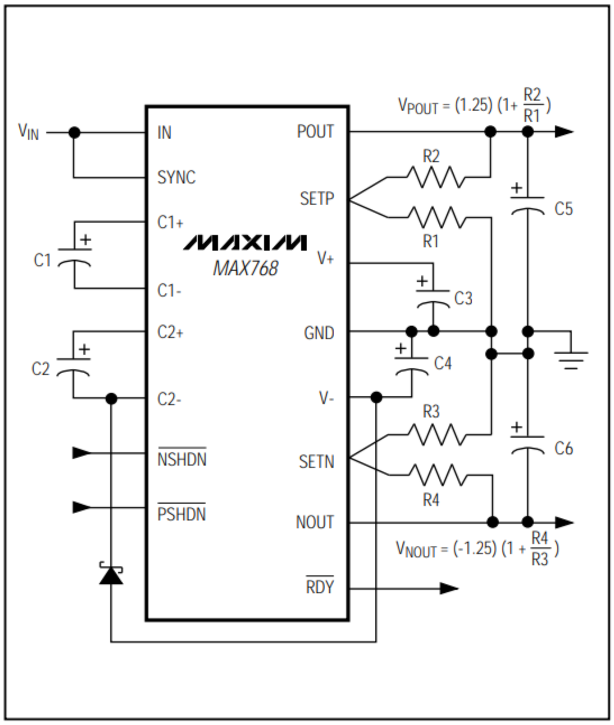


**Figure 4.2.6** MAX768 Adjustable Configuration [*Maxim Integrated*, 2022]

Possible resistor values can be calculated for the + 4.3 V range. The following is one example of the many resistor combinations that can be used:

1. Solve for the necessary resistor ratio of R2/R1 and R4/R3.

$$\frac{R2}{R1} = \frac{VPOUT}{VPSET\ REF} - 1$$

$$\frac{R4}{R3} = \frac{VNOUT}{VNSET\ REF} - 1$$

2. Plug-in reference voltages and desired voltages.

$$\frac{R2}{R1} = \frac{4.3}{1.25} - 1$$
$$\frac{R2}{R1} = 2.44$$

$$\frac{R4}{R3} = \frac{-4.3}{-1.25} - 1$$
$$\frac{R4}{R3} = 2.44$$

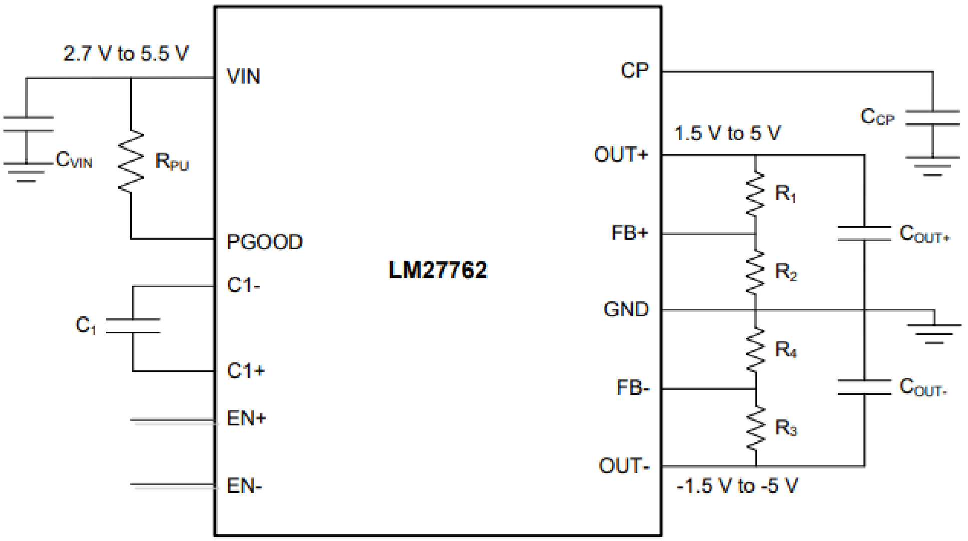
3. Pick a value between 100kΩ and 400kΩ for R1 and R3.

R1 and R3 were chosen to be 300kΩ because it is a common resistor value. 300kΩ also creates a necessary resistance for R2 and R4 that is close to a common resistor value of 750kΩ.

$$\frac{R2}{300k\Omega} = 2.44$$
$$R2 = 2.44 \cdot 300k\Omega$$
$$R2 = 732k\Omega \approx 750k\Omega$$
$$\therefore R4 = 732k\Omega \approx 750k\Omega$$

**LM27762**

TheLM27762 Low-Noise Positive and Negative Output Integrated Charge Pump from Texas Instruments provides an adjustable dual-supply voltage output, similar to the MAX768. The LM27762 output voltage range is smaller than that of the MAX768, ranging between +/-1.5 – 5V. This would not be an issue for this project because we are aiming for a minimum supply voltage of + 4.3V.



Copyright © 2016, Texas Instruments Incorporated

**Figure 4.2.7** LM27762 Standard Application Circuit [*Texas Instruments*, 2016]

The output voltage is set similarly to the MAX768 by adjusting external resistors R1, R2, R3, and R4.

$$\begin{aligned} VOUTP &= 1.2 \cdot (R1 + R2)/R2 && \textit{with } R2 \textit{ no less than } 50k\Omega \\ VOUTN &= -1.22 \cdot (R3 + R4)/R4 && \textit{with } R4 \textit{ no less than } 50k\Omega \end{aligned}$$

Possible resistor values can be calculated for + 4.3V range. The following is one example of the many resistor combinations that can be used:

**Positive Voltage Output:**

$$VOUTP = 1.2 \cdot (R1 + R2)/R2$$

1. Substitute desired output voltage.

$$4.3 = 1.2 \cdot (R1 + R2)/R2$$
$$\frac{4.3}{1.2} = (R1 + R2)/R2$$

2. Set R2 to a value greater than 50kΩ. Setting R2 = 200kΩ:

$$\frac{4.3}{1.2} = (R1 + 200k\Omega)/200k\Omega$$
$$\frac{4.3}{1.2} \cdot 200k\Omega = (R1 + 200k\Omega)$$
$$(\frac{4.3}{1.2} \cdot 200k\Omega) - 200k\Omega = R1$$
$$R1 = 516.7k\Omega \approx 510k\Omega$$

**Negative Voltage Output:**

$$VOUTN = -1.22 \cdot (R3 + R4)/R4$$

1. Substitute desired output voltage.

$$-4.3 = -1.22 \cdot (R3 + R4)/R4$$
$$\frac{-4.3}{-1.22} = (R3 + R4)/R4$$

2. Set R4 to a value greater than 50kΩ. Setting R4 = 200kΩ:

$$\frac{-4.3}{-1.22} = (R3 + 200k\Omega)/200k\Omega$$
$$\frac{-4.3}{-1.22} \cdot 200k\Omega = (R3 + 200k\Omega)$$
$$(\frac{-4.3}{-1.22} \cdot 200k\Omega) - 200k\Omega = R3$$
$$R3 = 504.9k\Omega \approx 510k\Omega$$

Table 4.2.2 provides a side-by-side comparison of the MAX768 and LM27762 charge pump regulators:

| Name | **MAX768** | **LM27762** |
|---|---|---|
| Input Voltage Range (V) | 3.0 – 5.5 | 2.7 – 5.5 |
| Positive Output Voltage (V) | Preset: 4.81 – 5.15 (Typ. = 5.0)<br>Adjustable: 1.25 – 11 | Adjustable: 1.5 – 5 |
| Negative Output Voltage (V) | Preset: -5.10 – -4.82 (Typ. = -5.0)<br>Adjustable: -1.25 – -11 | Adjustable: -1.5 – -5 |
| Package Type | Surface-Mount | WSON (not ideal) |

**Table 4.2.2** Charge Pump Regulator Comparison

### 4.2.2.2 Instrumentation Amplifier

The instrumentation amplifier stage provides a differential gain between the two electrode sites. This can be accomplished using an instrumentation amplifier such as the AD620. The AD620 will be expanded upon in more detail because we had several available to use at the time of this design process. The AD620 was also used in previous MQPs regarding ECG/EMG sensors. However, there are many options for instrumentation amplifiers that can be used, which will be listed below.

**AD620**

The AD620 by Analog Devices is a low cost, high efficiency instrumentation amplifier that allows for gain between 1 and 10,000. The gain is set easily by one external resistor, RG. The AD620 is low noise, low input bias current, and low power, making it the perfect option for an EMG application [Analog Devices, 2003-2011]. This device is becoming obsolete, but its availability in a DIP package makes it very useful for prototyping on a solderless DIP protoboard.

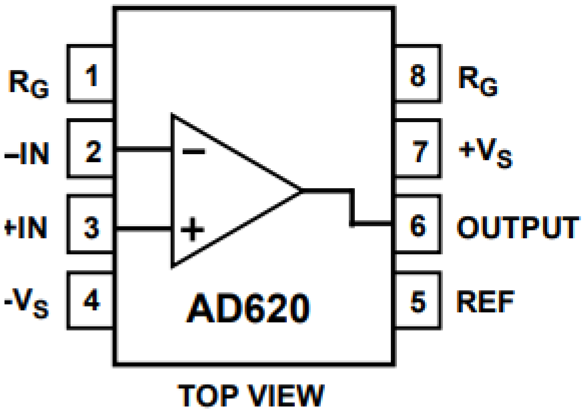


**Figure 4.2.8** AD620 Pin Diagram [Analog Devices, 2003-2011]

The external resistor RG sets the gain via the following formula:

$$G = \frac{49.4k\Omega}{RG} + 1$$

For this project, a gain of 20 is sufficient. A gain of 20 allows for good noise reduction while allowing for a reasonable capacitor value for Cg. A capacitor Cg is added in series with RG to set the DC gain to 1, rather than 20. This creates an effect where low-frequency signals (20Hz and below) are attenuated.
Using the gain equation:

$$\begin{aligned} 20 &= \frac{49.4k\Omega}{RG} + 1 \\ 19 &= \frac{49.4k\Omega}{RG} \\ RG &= \frac{49.4k\Omega}{19} \\ RG &= 2.6k\Omega \end{aligned}$$

The value of Cg was chosen using a MATLAB script called “iamp_stage_v01” from a previous MQP by Ibrahim et al. (see Appendix A for MATLAB script). This script plots the magnitude response of the instrumentation amplifier over a set range of frequencies. In this case, we care about the frequency range of an EMG signal, so a range of 0 to 200 Hz was chosen for the frequency sweep.

The formula for the magnitude response of the AD620 instrumentation amplifier is as follows:

$$H(w) = 1 + \frac{49.4k\Omega}{RG + \frac{1}{j \cdot w \cdot Cg}}$$

The value for RG was calculated above as 2.6kΩ. Several Cg values were tested to find a value where the frequency range of 0-20 Hz was attenuated in the stop band.

After testing various Cg values, a Cg value of 10uF was chosen because it attenuated the 0-20 Hz frequency range similar to a high pass filter. The “stop band” also has a sharp roll-off in this frequency range, which is ideal (see Figure 4.2.9).

$$Cg = 10uF$$

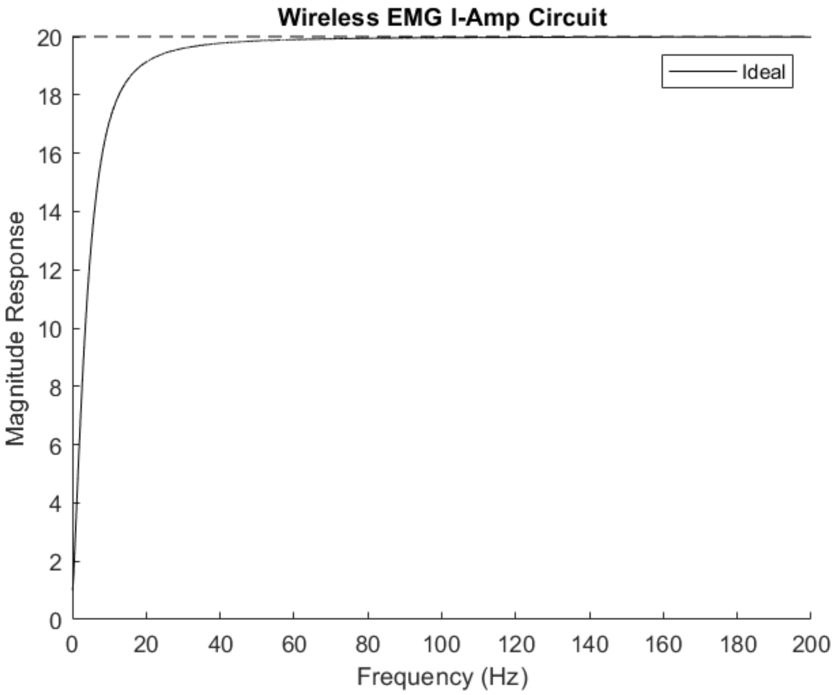


**Figure 4.2.9** Magnitude Response with Capacitor Value of 10uF

Figure 4.2.9 shows the magnitude response of the instrumentation amplifier stage. The DC gain equals one. The frequency is attenuated around 20 Hz, which is what was desired given our design specifications.

**Other Instrumentation Amplifiers**

Table 4.2.3 compares several other options for instrumentation amplifiers. All of these options are capable of providing the system with the desired 20 V/V gain.

| Name | INA121P | AD8221 | INA128P |
|---|---|---|---|
| Supply Voltage Range (V) | +/- 2.25 – 18 | +/- 2.3 – 18 | +/- 2.25 – 18 |
| Gain (V/V) | 1 – 10,000 | 1 – 1,000 | 1 – 10,000 |
| Gain Equation | 1 + (50kΩ/RG) | 1 + (49.4 kΩ/RG) | 1 + (50kΩ/RG) |

**Table 4.2.3** Instrumentation Amplifier Comparison

4.2.2.3 High Pass Filter

The high pass filter stage eliminates DC offset from the signal, attenuates any motion artifacts, and provides a necessary gain to the signal so that it is within the necessary range to become input to the ADC. The ADC runs on a voltage range of 0 - 3.6 V. This means the signal must be amplified to 3.6 V. The voltage range of an EMG signal was determined as approximately a maximum of +/-0- 10 mV, which is an overall range of 20 mV. The necessary minimum gain to accomplish this desired voltage is determined by the following:

$$\begin{gathered} EMG\ signal\ range \ \cdot \ Gain \ = \ Desired\ Voltage \\ 20\ mV \ \cdot \ Gain \ = \ 3.6\ V \\ Gain \ = \ \frac{3.6\ V}{20\ mV} \\ Gain \ = \ 180 \end{gathered}$$

However, since the front end has a gain of 20, the actual gain needed is:

$$High\ Pass\ Filter\ Gain \ = \ 180/20 \ = \ 9$$

The previous project group (Ibrahim et al.) used a fourth-order high pass filter. However, for this project, the decision was made to use a second-order high pass filter to keep the circuit operation simpler, lower power, and occupy less space. The second-order filter will still provide a fast enough roll-off in the stop band for our application. For example, Figure 4.2.10 shows the increase in steepness of the stop band when the filter order is increased. The blue line shows a single-order high pass filter versus the red line second-order filter.

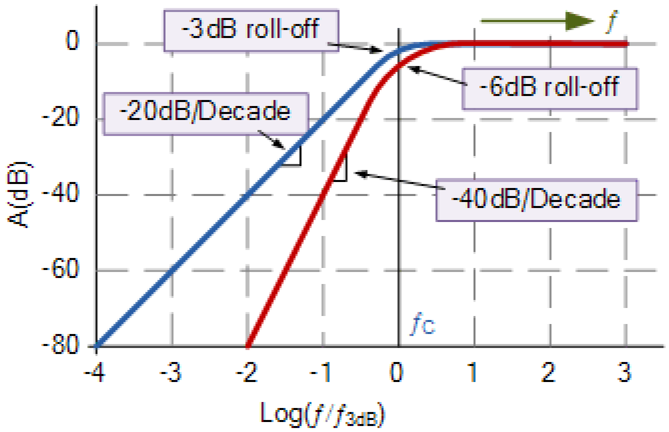
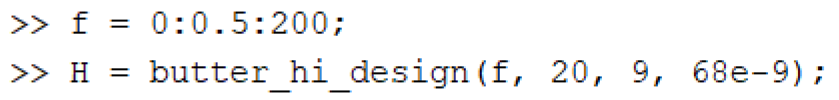


**Figure 4.2.10** High-Pass Filter Frequency Response [*electronics-tutorials.ws*, 2022]

To determine the necessary resistor and capacitor values for the calculated gain of 9, a MATLAB script called “butter_hi_design” [E. A. Clancy] was used (Appendix B). A function to plot the magnitude response was used: function H = butter_hi_design(f, Fc, A, Stage1, Stage2, Stage3, Stage4, Stage5). The function requires input for the frequency range (f), desired cutoff frequency (Fc), desired gain (A), and StageX, where one capacitor value (C1) must be provided for each stage X (C2 will automatically be set equal to C1).

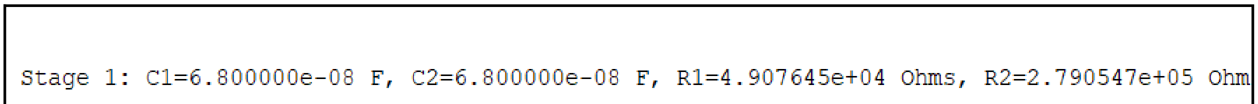
It was determined that the high-pass filter in this front end circuit will be second-order (Stage 1). Below is the input to the MATLAB command window for creating the second-order filter:



```
>> f = 0:0.5:200;
>> H = butter_hi_design(f, 20, 9, 68e-9);
```

**Figure 4.2.11** MATLAB Command Window for High-Pass Filter

The frequency range for evaluation of the high-pass filter was set for 0 to 200 Hz to replicate the desired frequency range of the EMG signal. The cutoff frequency was chosen as 20 Hz to attenuate the noise and motion artifacts within the 0 - 20 Hz range. The gain was set as 9 based on the calculation done above. The capacitor value for C1 was chosen as 68 nF because it is a readily available capacitor value. The output of the MATLAB script provides the necessary component values and the magnitude response plot of the filter.



```
Stage 1: C1=6.800000e-08 F, C2=6.800000e-08 F, R1=4.907645e+04 Ohms, R2=2.790547e+05 Ohm
```

**Figure 4.2.12** High-Pass Filter Component Values

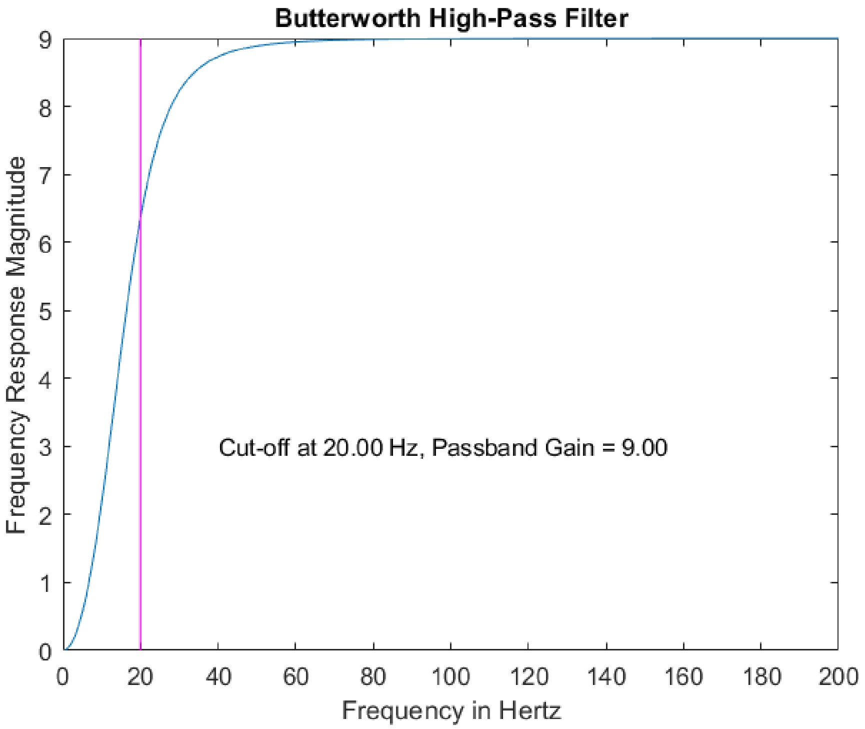


**Figure 4.2.13** High-Pass Filter Design Magnitude Response

To save space when constructing the circuit, quad operational amplifiers should be used. Table 4.2.4 provides options for operational amplifiers that are within the required operating voltage specifications.

| Name | LM348N | LMC6484 | LT1491 |
|---|---|---|---|
| Input Operating Voltage (V) | +/- 4 – 18 | +/- 3 – 15.5 | +/- 2 – 44 |
| Number of Channels | 4 | 4 | 4 |

**Table 4.2.4** Operational Amplifier Comparison

### 4.2.2.4 Selectable Gain

The selectable gain stage uses a non-inverting amplifier with the ability to change the gain it produces. The gain needed varies depending on the voltage of the EMG signal. This stage ensures that even the lowest EMG signals can be seen by the ADC.

As discussed in Section 2.3.3.4, the gain equation for a non-inverting amplifier is $Gain = 1 + \frac{R2}{R1}$. The value of R2 can be chosen as any resistor value. For example, R2 can be set as 300kΩ. The array of R1 resistor values should create gains of 1, 2, 4, 8, and 16. Table 3.1.1.4 shows the necessary resistor values for the desired gains.

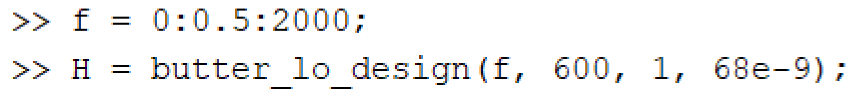
| Desired Gain (V/V) | R1 Value (Ω) |
|---|---|
| 1 | 0 |
| 2 | 300k |
| 4 | 100k |
| 8 | 43k |
| 16 | 20k |

**Table 4.2.5** Selectable Gain Resistor Values

### 4.2.2.5 Low Pass Filter

The low-pass filter stage is designed in a similar fashion to the high-pass filter stage. A MATLAB script called “butter_lo_design” [E. A. Clancy] was used to determine the necessary component values for a desired cutoff frequency (Appendix C). The desired cutoff frequency for the low-pass filter is 600 Hz. This attenuates any noise that is outside the overall EMG signal range (above 600 Hz).

The function H used in the MATLAB script was similar to the high-pass function, except it utilizes the equation for frequency response of a low-pass filter, rather than high-pass: function H = butter_lo_design(f, Fc, A, Stage1, Stage2, Stage3, Stage4, Stage5). The function accepts the same variables as the high-pass filter: frequency range, cut-off frequency, gain, and C1 capacitor value for each stage. For this design, the frequency range was set from 0 to 2000 Hz to make sure the desired low-pass cut-off frequency could be reached in the frequency sweep. The cut-off frequency as stated previously is 600 Hz. The necessary gain for this stage is a gain of 1 because there is no further desire to amplify the signal in this stage. The filter is only attenuating the high frequencies above 600 Hz and maintaining all frequencies below 600 Hz. The filter is being designed as second-order so only one C1 capacitor value needs to be provided (C2 will automatically be set equal to C1). C1 was chosen as 68 nF to match the capacitor used in the high-pass filter. This is not a requirement to have the capacitor values match, but it was done for ease of purchasing. The MATLAB command window is shown below:



```
>> f = 0:0.5:2000;
>> H = butter_lo_design(f, 600, 1, 68e-9);
```

**Figure 4.2.14** MATLAB Command Window for Low-Pass Filter

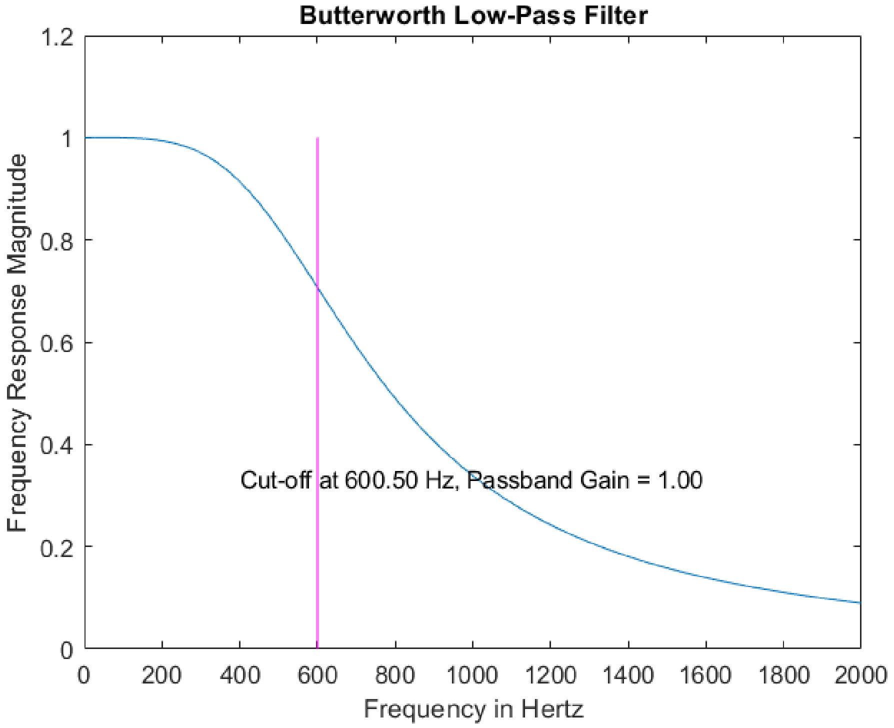


**Figure 4.2.15** Low-Pass Filter Design Magnitude Response

### 4.2.2.6 Level Shifter

The level shifter circuit is used to provide the ADC with a unipolar signal. Prior to this stage, the EMG signal is bipolar, meaning it contains both positive and negative voltages. The ADC can only read positive voltage values, therefore the EMG signal must be offset by one-half of the voltage range of the ADC. The ADC range is 3.6 V, so centering the signal at 1.8 V would effectively account for the negative voltages of the EMG signal.

A voltage regulator, in combination with an adder circuit (discussed subsequently), such as the LM317BT from STMicroelectronics can be used to adjust the EMG signal voltage to the desired range. The output voltage range of this voltage regulator is 1.2 V to 37 V, which is suited for this application. The circuit diagram for using this device properly with protection diodes is shown in Figure 4.2.16.

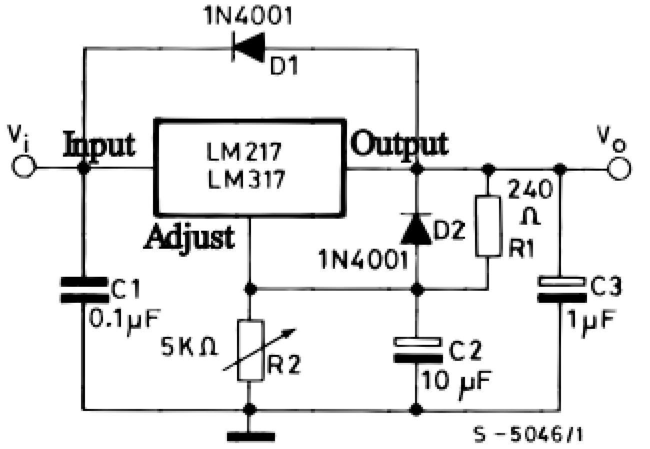


**Figure 4.2.16** Voltage Regulator Operating Circuit [STMicroelectronics, 2022]

The value of the voltage regulator is set by the following equation:

$$VO \ = \ (1.25 \ * \ (1 + \frac{R2}{240})) \ + \ (IADJ \ * \ R2)$$

R1 is a fixed value of 240 Ω to internally set the 1.25 V reference voltage. For a desired VO of 1.8 V, the value for R2 must be calculated. The error term IADJ*R2 can be neglected in this equation because the value of IADJ must be less than 100 uA, which is very small. The R2 resistor value necessary for the 1.8 V output of the voltage regulator was calculated to be 105.6 Ω. The actual values used in the physical implementation of the voltage amplifier are 200 Ω and 51 Ω in series for R1 and 100 Ω and 10 Ω in series for R2.

$$VO \ = \ VREF \ * \ (1 + \frac{R2}{R1})$$
$$1.8\,V \ = \ 1.25\,V \ * \ (1 + \frac{R2}{R1})$$
$$1.44 \ = \ (1 + \frac{R2}{R1})$$
$$0.44\,V = \frac{R2}{R1}$$

An example of the level shifter circuit from Ibrahim et al. is shown below. This circuit used the same voltage regulator output formula and required choosing the necessary values of R15 and R16 in the schematic.

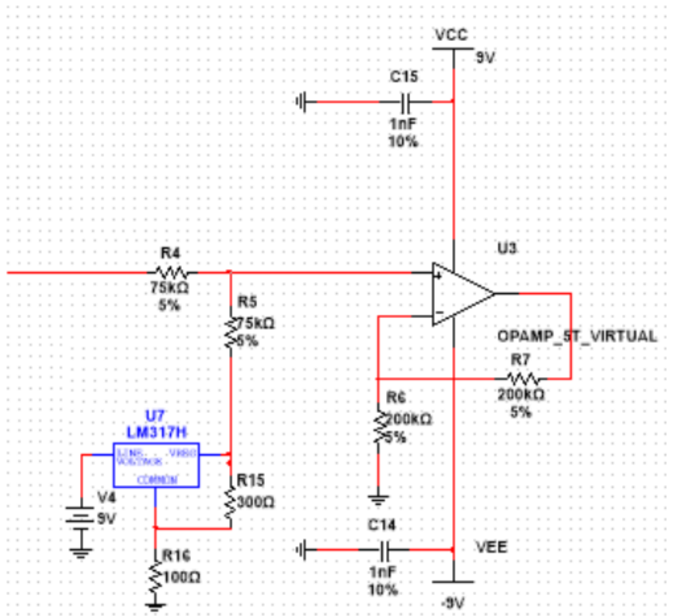


**Figure 4.2.17** Level Shifter Circuit [Ibrahim et al., 2021]

### *4.2.3 Electrode Design Options*

There are five electrodes placed on the arm to collect sEMG signals. There are two options for the types of electrodes that can be used:

1. Commercial surface electrodes
2. Stainless steel screws

The pros and cons for each option can be seen in Table 4.2.3.1. The screw electrodes are cheap and reusable. However, the signal quality can be lower in comparison to commercial surface electrodes. If commercial surface electrodes are used, then the pads will likely have to be periodically replaced.

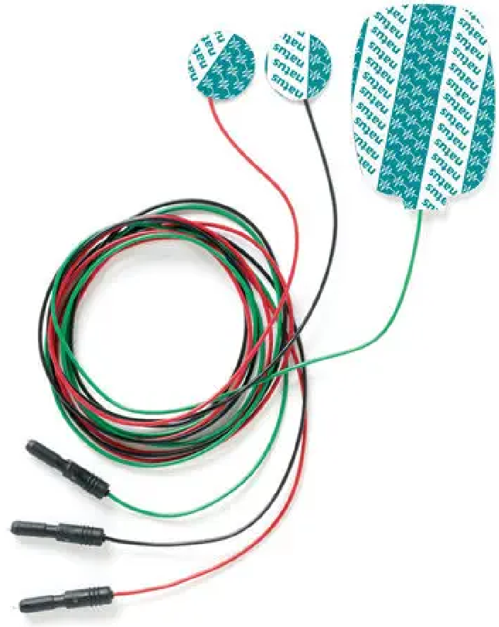


**Figure 4.2.3.1** Example of Surface Electrodes [*Medicalexpo.com*, 2022]

**Figure 4.2.3.2** Screw [Home Depot, 2022]

| | Screw | Traditional Surface Electrodes |
|---|---|---|
| Pros | Cheap | High quality signal |
| Cons | Lower quality signal | Expensive |

**Table 4.2.3.1** Electrode type pros and cons

Electrodes will be placed on the arm to collect sEMG signals. There are two options of where the electrode pairs can be placed on the arm:

1. Biceps-Triceps
2. Forearm (anterior-posterior)

The pros and cons for each option are shown in Table 4.2.3.2. The biceps-triceps are expected to have quality sEMG signals because of the lack of hair in the area. However, a higher range of arm motion will be required to control pong which may be tiring for the participant. Alternatively, the forearm may have lower-quality signals but will require a lower range of arm motion.

| | Biceps-Triceps | Forearm |
|---|---|---|
| Pros | Less hair (better quality) | Low range of arm motion |
| Cons | High range of motion | More hair (lesser quality) |

**Table 4.2.3.2** Electrode placement pros and cons

If the screw-based electrodes are chosen, they must be housed in one of the following two options as shown in Figure 4.2.3.3:

1. Armband
2. Clip

The pros and cons for each option are shown in Table 4.2.3.4. The armband will be made of either Velcro or bandages that will wrap around the biceps-triceps or forearm. However, the armband is subject to sliding. Alternatively, the clip provides high pressure around the participant’s arm and can easily be donned and doffed. Though, the clip may not fit all users.

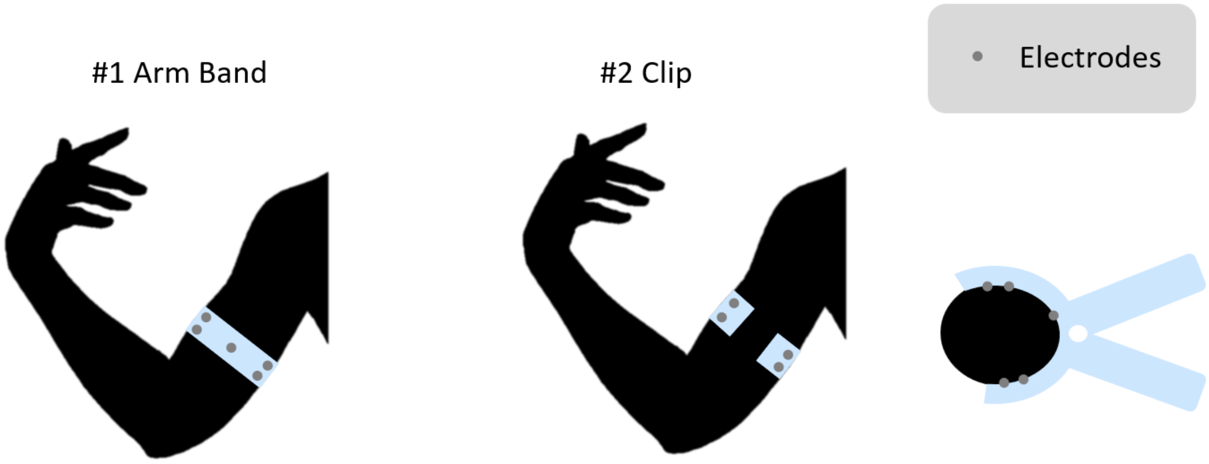


**Figure 4.2.3.3** Electrode housing design options

| | Arm Band | Clip |
|---|---|---|
| Pros | Adjustable applied pressure,<br>Flexible | High applied pressure,<br>Simple donning and doffing |
| Cons | Subject to sliding | May not fit all participants |

**Table 4.2.3.3** Electrode housing pros and cons

If the armband is chosen, then the pros and cons of the armband cloth type must also be considered as listed in Table 4.2.3.4. Velcro allows the user to easily don and doff the electrodes and is not as breathable as cloth. The lack of airflow causes sweat accumulation which may improve signal quality. However, cloth is less rigid than Velcro making it more comfortable for the user.

|  | Velcro | Bandages |
|---|---|---|
| Pros | Easy to don and doff,<br>Non-breathable | Flexible |
| Cons | Rigid | Requires a clip to hold the wrap in place,<br>Breathable |

**Table 4.2.3.4** Armband material pros and cons

The complete electrode design options are as shown in Figure 4.2.3.4.

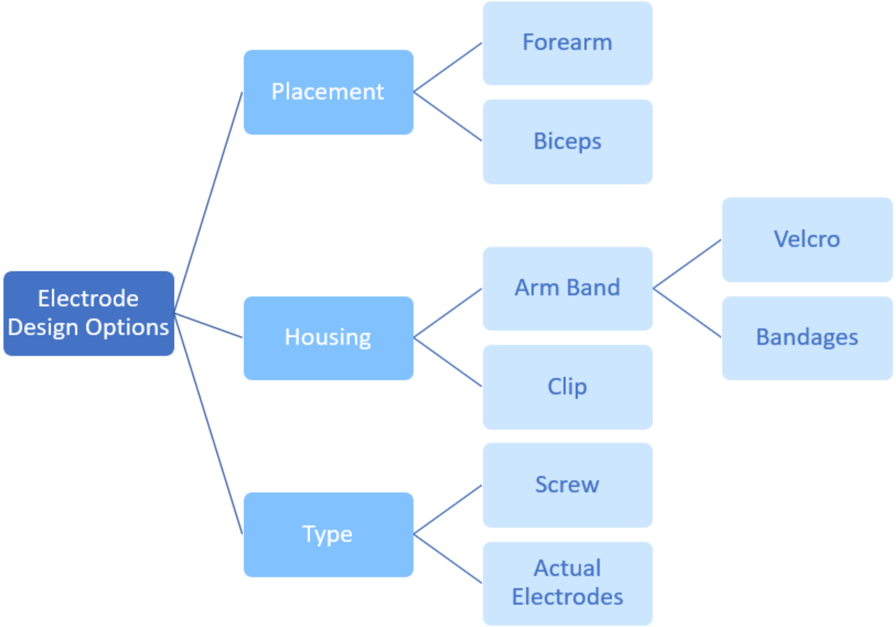


**Figure 4.2.3.4** Electrode design options

### *4.2.4 Final Design*

The final design of the analog front end section of the design consisted of a breadboard design and solder board design The breadboard design consisted of two breadboards, 9V batteries, and Velcro armbands for electrode housing. Once the functionality of the system was proven on the breadboards, the solder board design was created and implemented. The solder board was originally placed on a flat surface for testing, but eventually was built into an armband worn by the user.

## *4.3 Testing*

### *4.3.1 Methodology*

The testing for the analog front end circuit consisted of MATLAB simulations, breadboard, and solder board testing. The MATLAB simulations tested the front end in stages, beginning with the instrumentation amplifier, then adding the additional stages. The breadboard was tested with only one channel since the second channel would simply be a duplicate of the first channel. The breadboard was first tested with the function generator and power supply to prove functionality. Once functionality was proven, the breadboard was tested with electrodes and battery power. The solder board was tested with both input channels to ensure that all solder connections were correct and nothing was shorted on the board. The first round of solder board testing was conducted using the bench power supply and function generator to ensure proper functionality at each stage of the circuit. Then, the electrodes and battery power were used to test the output of the front end and ensure that accurate EMG signals were being provided to the microcontroller.

### *4.3.2 Results & Observations*

#### 4.3.2.1 Multisim

**Instrumentation Amplifier**

The instrumentation amplifier stage consists of the AD620 and necessary gain resistor value (RG) to produce our desired gain of 20. This value of RG was calculated previously in the design options section of the report (see section 4.2.1.2). RG was determined to be 2.6kΩ. The 10uF capacitor was included to provide a gain of 1 at DC. The 0.01uF capacitors on the power rails were used as decoupling capacitors to keep the supply voltage to the circuit stable and free of AC noise.

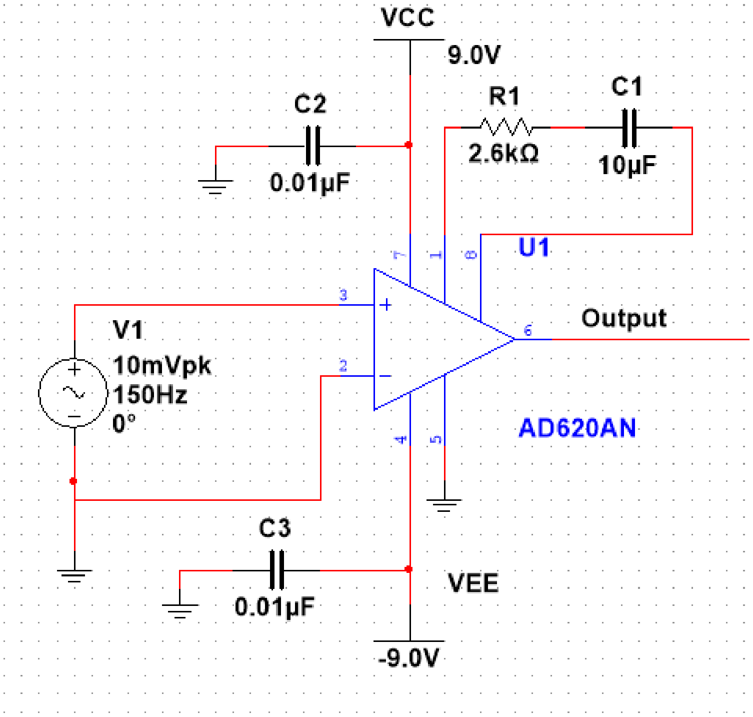
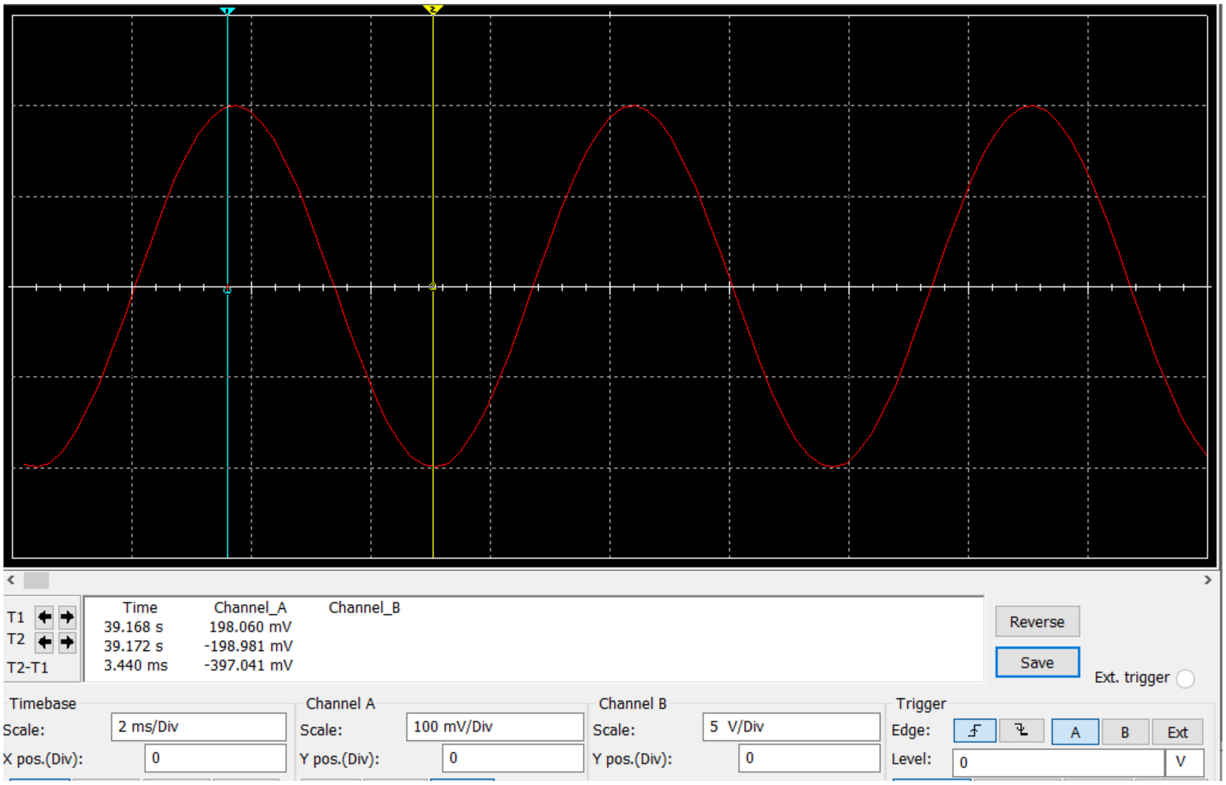


**Figure 4.3.1** Instrumentation Amplifier Schematic

This stage was tested using an input signal of 10mVpk and 150Hz. This test signal was chosen since it is at the maximum end of our expected EMG signal range. Only one input signal was applied to the input of the instrumentation amplifier for ease of testing. This way one of the signals being compared is 0V, making it easy to observe the desired gain of 20 to the initial input signal. In this case, the positive terminal input signal was 10 mVpk 150 Hz and the negative terminal input signal was 0 Vpk 0 Hz. Therefore, we expect to see an output signal of 200 mVpk 150 Hz when the gain of 20 is applied. This gain was observed in Figure 4.3.2.



```
T1      Time        Channel_A     Channel_B
        39.168 s    198.060 mV
T2      39.172 s    -198.981 mV
T2-T1   3.440 ms    -397.041 mV
```

Timebase Scale: 2 ms/Div; X pos.(Div): 0

Channel A Scale: 100 mV/Div; Y pos.(Div): 0

Channel B Scale: 5 V/Div; Y pos.(Div): 0

Trigger Edge: A; Level: 0 V

**Figure 4.3.2** Instrumentation Amplifier Multisim Output

### Band-Pass Filter with Selectable Gain

The design of the band-pass filter stage is discussed beginning in section 4.2.1.3. Below is the final schematic design in Multisim.

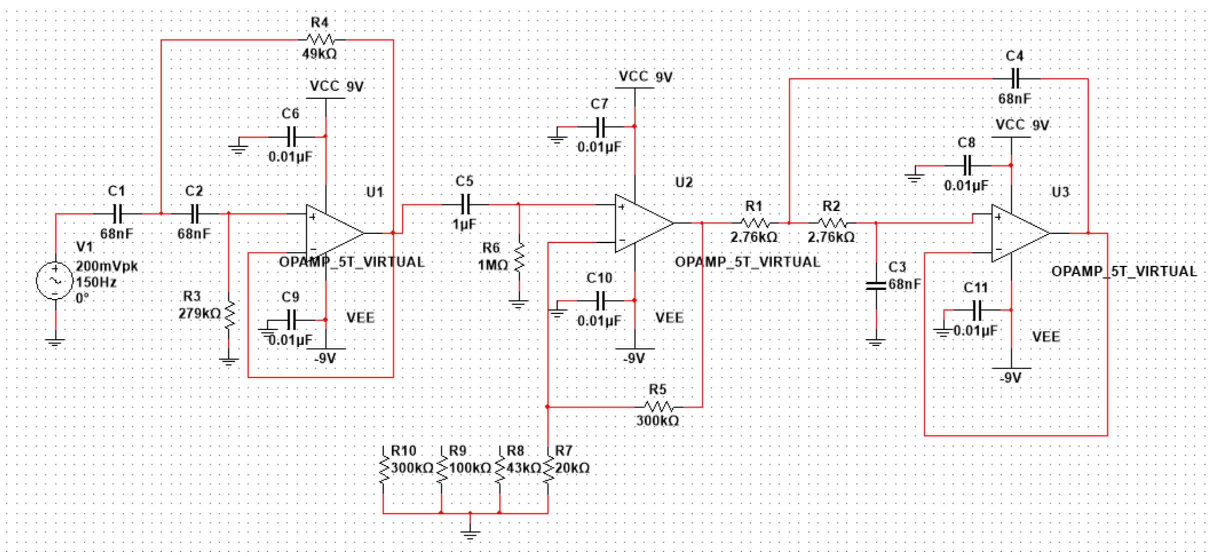
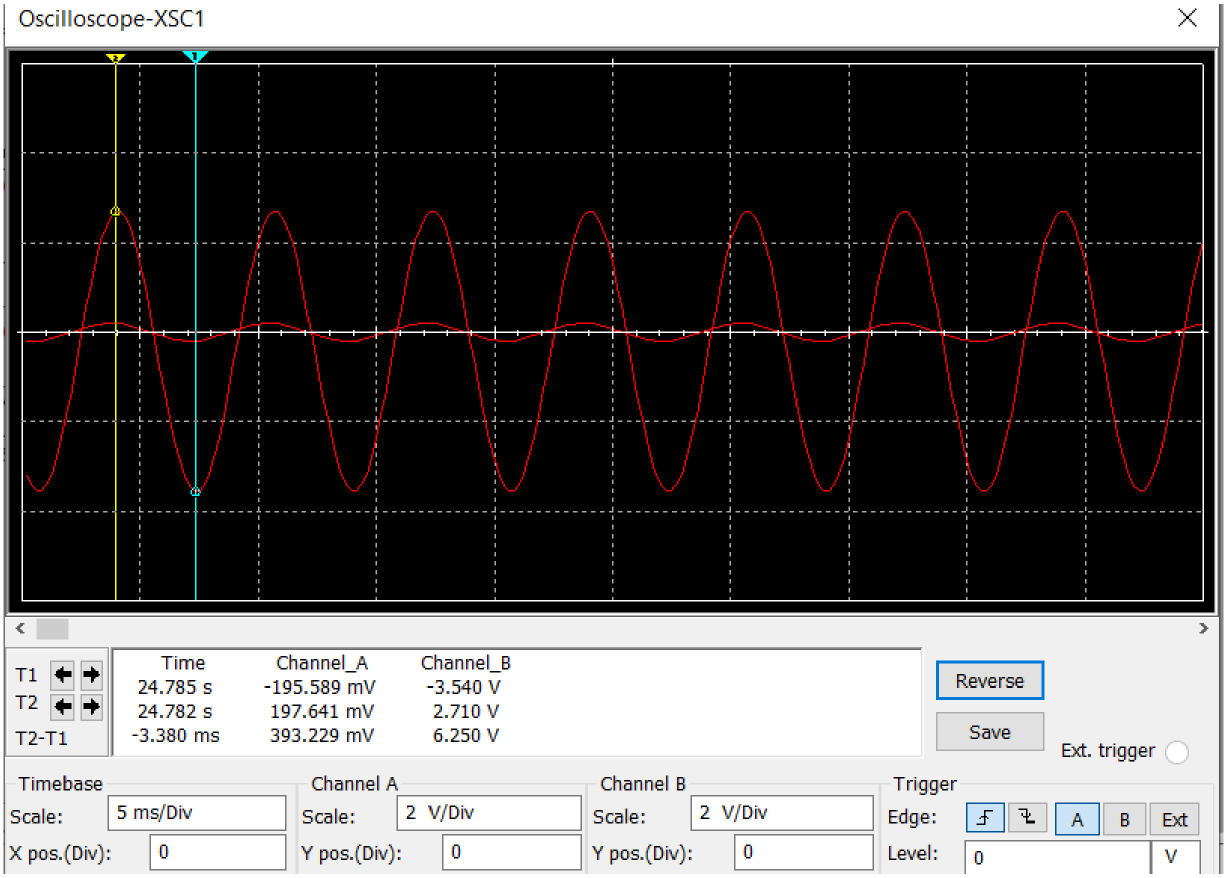


**Figure 4.3.3** Band-Pass Filter with Selectable Gain Schematic

The band-pass filter stage was tested with an input signal of 100 mVpk 150 Hz to replicate the output of the instrumentation amplifier stage. The input of this stage should be amplified by whichever gain was chosen via the selecTable gain resistor (either R7, R8, R9, or R10 in the schematic). For testing purposes, the gain resistor chosen was the 20kΩ resistor, meaning a gain of 16 was expected. From the oscilloscope output, the ratio of output to input voltages is 2.710 V / 197.6mV = 13.76 and -3.540 V / -195.59 mV = 18.10. These values average out to a gain of 15.93, which is approximately a gain of 16.



|  | Time | Channel_A | Channel_B |
|---|---|---|---|
| T1 | 24.785 s | -195.589 mV | -3.540 V |
| T2 | 24.782 s | 197.641 mV | 2.710 V |
| T2-T1 | -3.380 ms | 393.229 mV | 6.250 V |

Timebase
Scale: 5 ms/Div
X pos.(Div): 0

Channel A
Scale: 2 V/Div
Y pos.(Div): 0

Channel B
Scale: 2 V/Div
Y pos.(Div): 0

Trigger
Level: 0 V

**Figure 4.3.4** Band-Pass Filter with Gain of 16 Multisim Output

The frequency response of the band-pass filter was also observed. This analysis was completed to verify the cutoff frequencies of 20 Hz and 600 Hz. The magnitude of the pass band was also verified and should be 20log(16) or approximately 24.08 dB. Figure 4.3.5 verifies the correct cutoff frequencies and correct pass band magnitude.

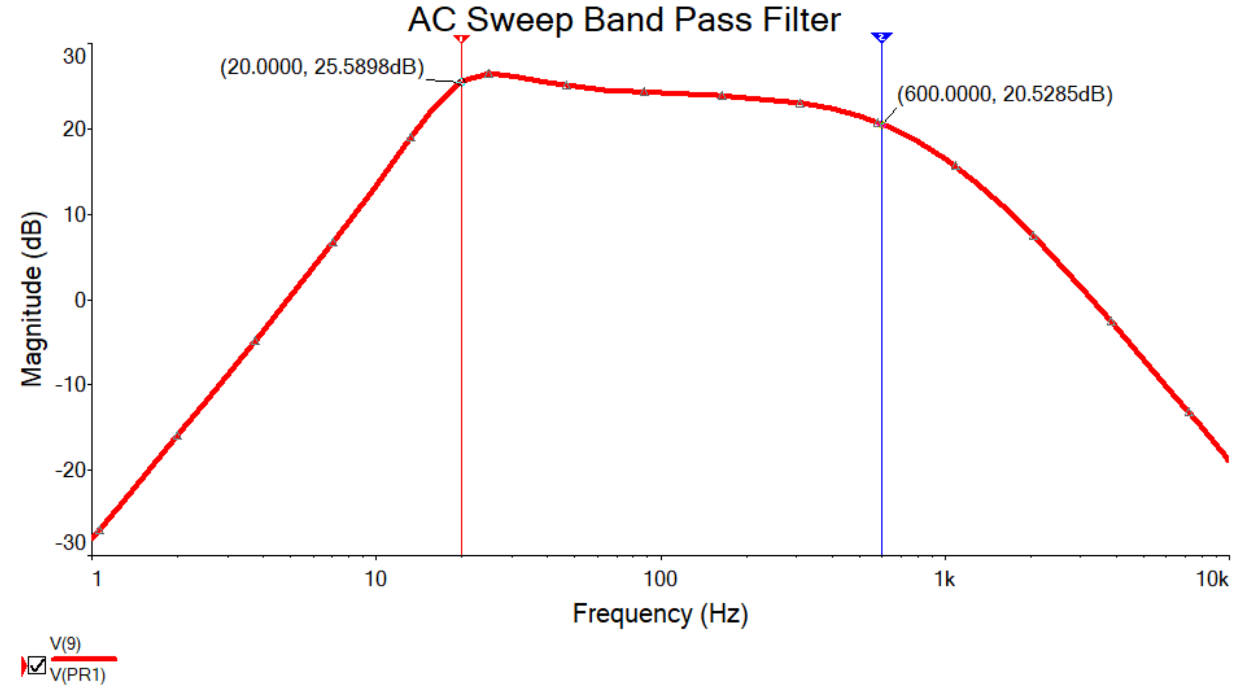


**Figure 4.3.5** Frequency Response of Band-Pass Filter

**Voltage Regulator**

The voltage regulator was tested using Multisim Online. The LM317H voltage regulator being used in our design was not available in Multisim Version 14.3 that was being used to test the other stages of the circuit. Therefore, the online version was used to verify the 1.8 V output of the voltage regulator as seen in Figure 4.3.6. A voltage probe was placed on the output terminal and a voltage of 1.80 V was read.

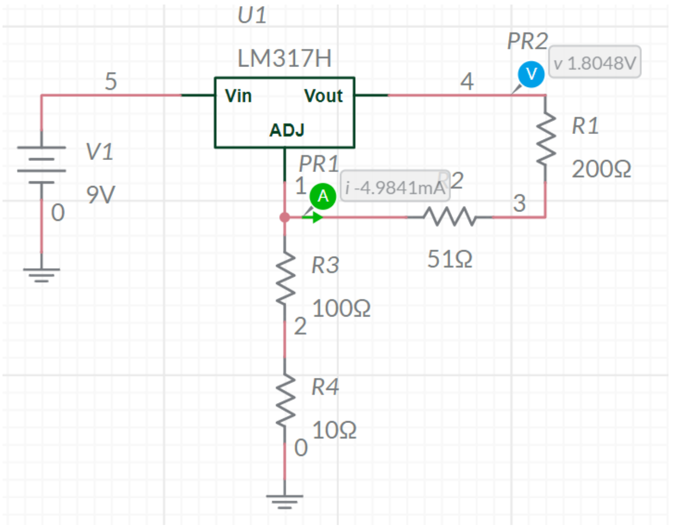


**Figure 4.3.6** Voltage Regulator Test Schematic

**Overall Front End**

The overall front end design is shown in Figure 4.3.7. Two duplicates of this single “channel” design make up the overall front end circuit. One channel is used for the anterior electrodes and one channel is used for the posterior electrodes.

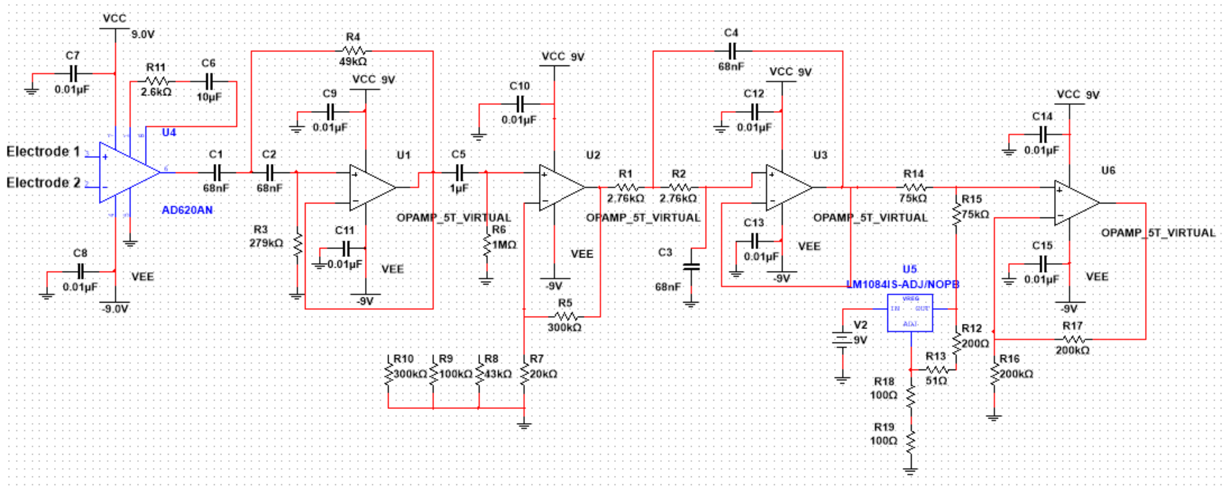


**Figure 4.3.7** Overall Front End Schematic (One Channel)

**Note:** The LM1084 WAS NOT USED IN THIS DESIGN. This is just a placeholder since the LM317H was not available for use in Multisim Version 14.3. The design of the entire circuit was attempted in Multisim Online using the correct voltage regulator, LM317H; however, the online software limits the number of components allowed in the design, so this was not possible.

### 4.3.2.2 Breadboard

The first step in breadboard testing was testing one channel (one breadboard) using a dual power supply and function generator to prove circuit functionality.

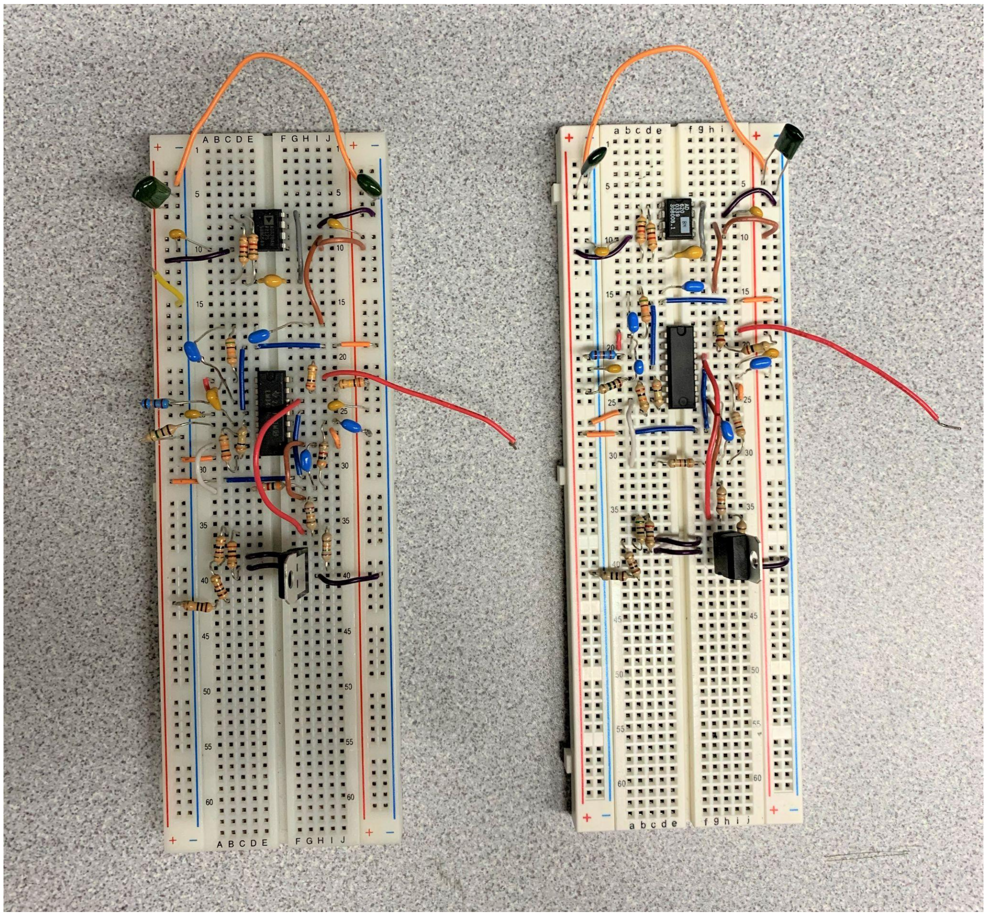


**Figure 4.3.8** Two Front End Channels on Breadboard

The system was tested using an input signal of 10 mVpk @ 150 Hz. The testing was broken down into each stage of the system, cascading each stage together once prior stages were proven functional.

The first test completed was on the instrumentation amplifier stage. The input signal of 10 mVpk @ 150 Hz was applied and an output of approximately 200mVpk was observed. This outcome is expected since the instrumentation amplifier stage has a gain of 20.

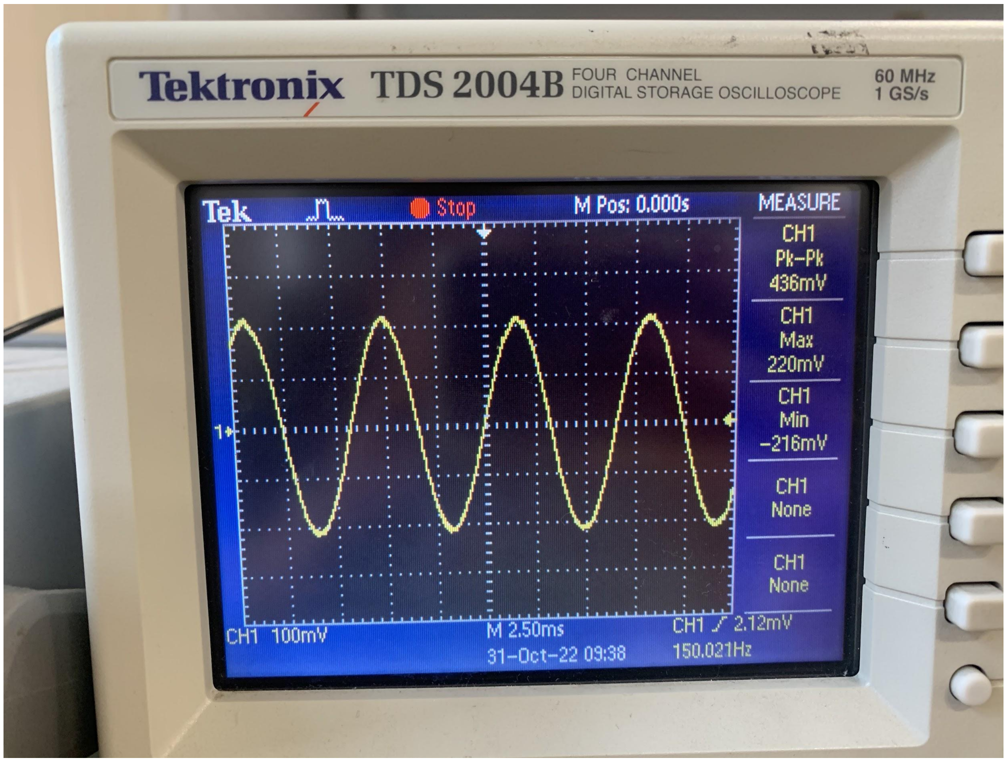


**Figure 4.3.9** Oscilloscope Output of Instrumentation Amplifier

The band pass filter stage was then cascaded to the instrumentation amplifier stage. The output of the instrumentation amplifier stage is shown in yellow and the output of the band pass stage is blue. A gain of 16 was tested, so an output of approximately 3.5 Vpk was expected. This gain was observed below in Figure 4.3.10.

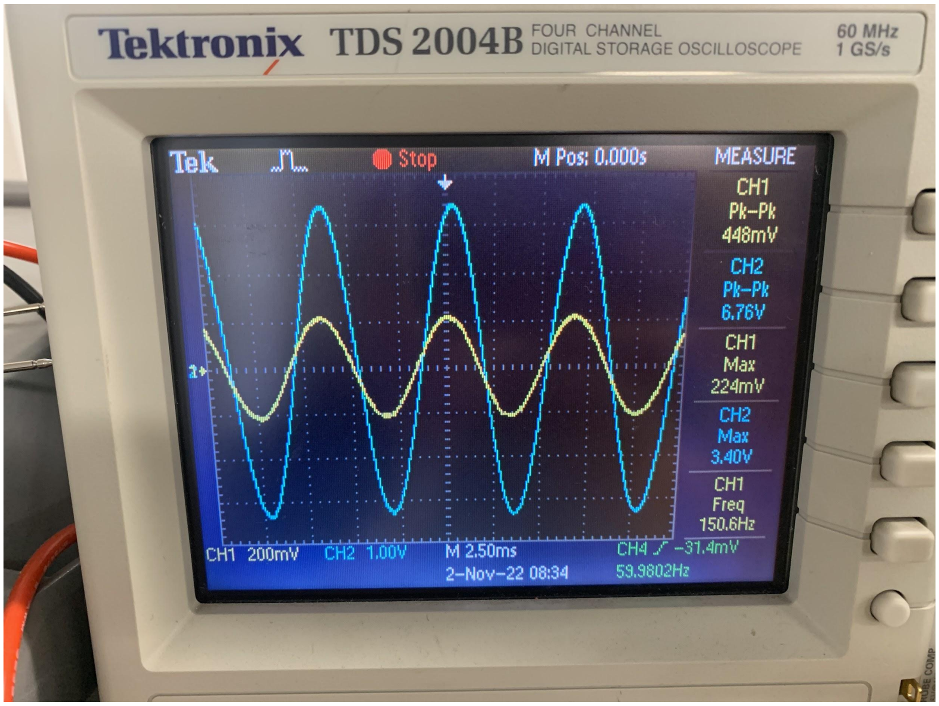


**Figure 4.3.10** Oscilloscope Output of Band-Pass Filter with Gain of 16

The overall front end circuit was then tested with an input signal of 20 mVpk @ 150 Hz and a selectable gain of 1 for ease of analysis. Figure 4.3.11 shows the output of the system level shifted by 1.8V to be within the range of the ADC. The signal is now centered around this 1.8V mark, rather than at zero which produces negative values that cannot be read by the ADC.

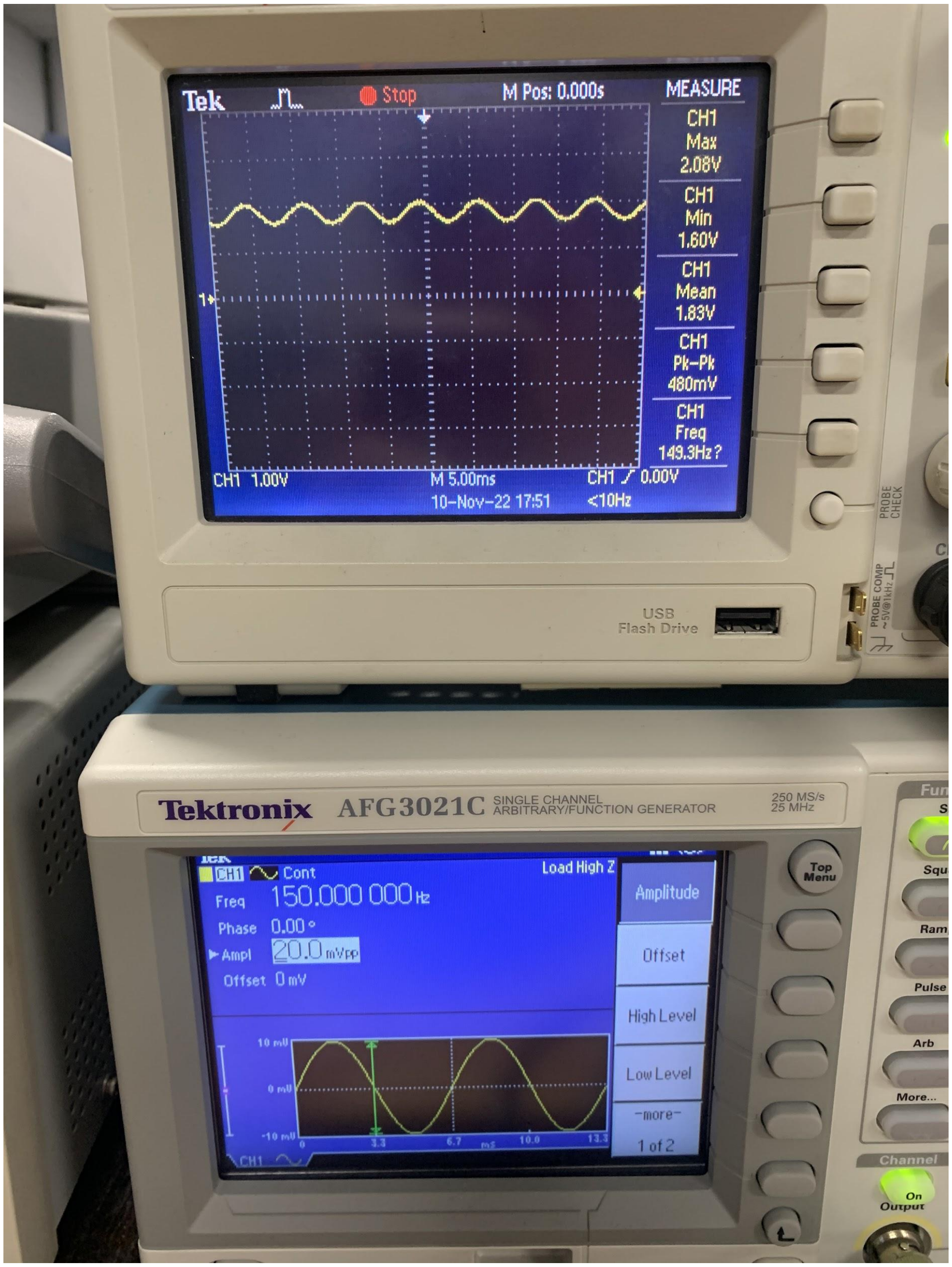


**Figure 4.3.11** Oscilloscope Output of One Breadboard Channel with Gain of 1

Once the functionality of one breadboard was proven, a second board was constructed and both boards were tested using battery power and electrodes. Two 9V batteries were used to create the dual 9V power supply as seen in Figure 4.3.12. Battery clip leads were attached to the batteries to provide an easy connection to the breadboard without the need of soldering directly to the battery terminals.

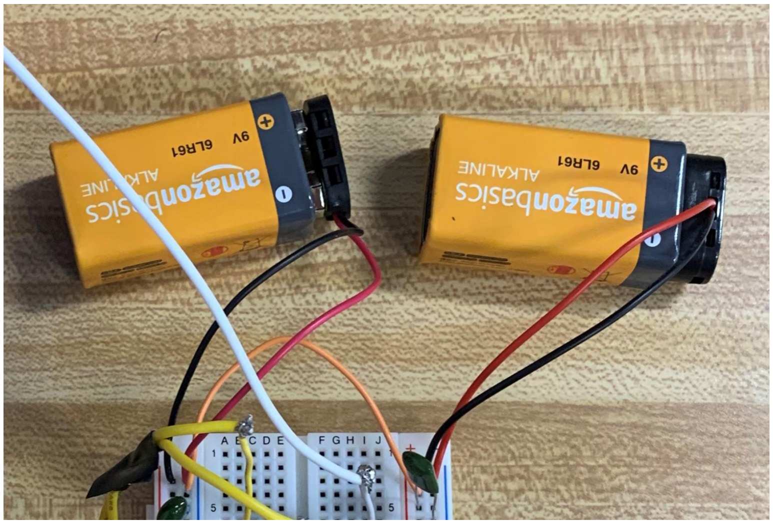


**Figure 4.3.12** Dual Power Supply Battery Setup

The electrodes used were made of screws, nuts, and washers. Two electrodes were used per channel and one ground reference electrode was used for both channels, for a total of five electrodes. Figure 4.3.13 shows the electrodes on separate bands to place on the different forearm locations for flexion, extension, and ground reference.

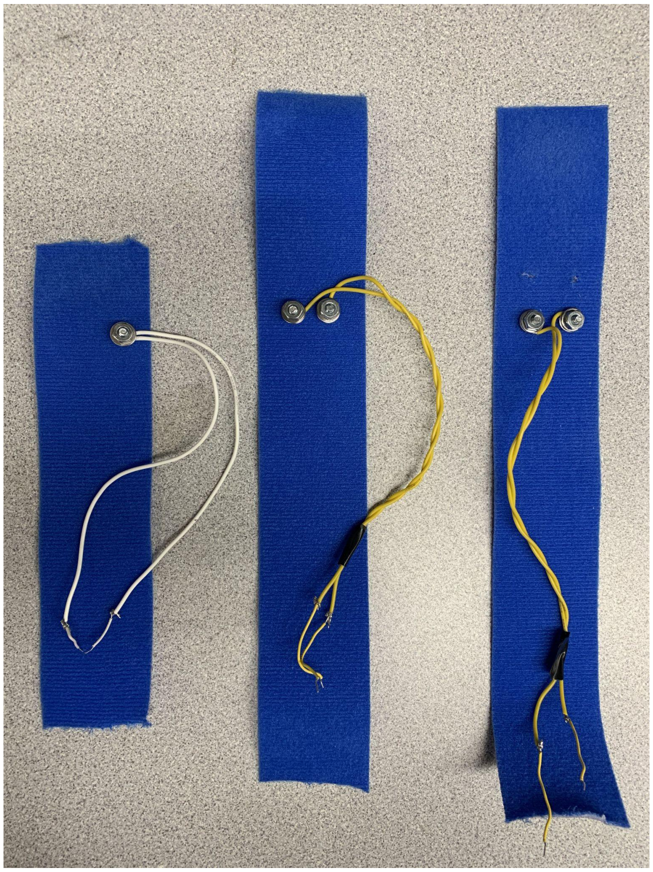


**Figure 4.3.13** Electrode Arm Bands

Both channels were tested for flexion and extension of the wrist. Each breadboard channel was connected to a different armband that was placed on either the flexion muscle or extension muscle. When flexion of the wrist occurs, the channel connected to the flexion electrodes should show an EMG response while the extension channel should show no response. The same applies to the extension of the wrist muscle; the extension channel should show EMG response while the flexion channel remains quiet. Activating the different flexion and extension muscles is our indicator of how the paddle moves left or right in the pong application.

Figures 4.3.14 and 4.3.15 show both channels of the breadboard functioning as expected. The purple signal is the output of the flexion channel and the yellow signal is the output of the extension channel. When the flexion channel is “activated,” the EMG signal is observed on the flexion channel and the extension channel is flat. The opposite behavior is observed when the extension channel is activated instead.

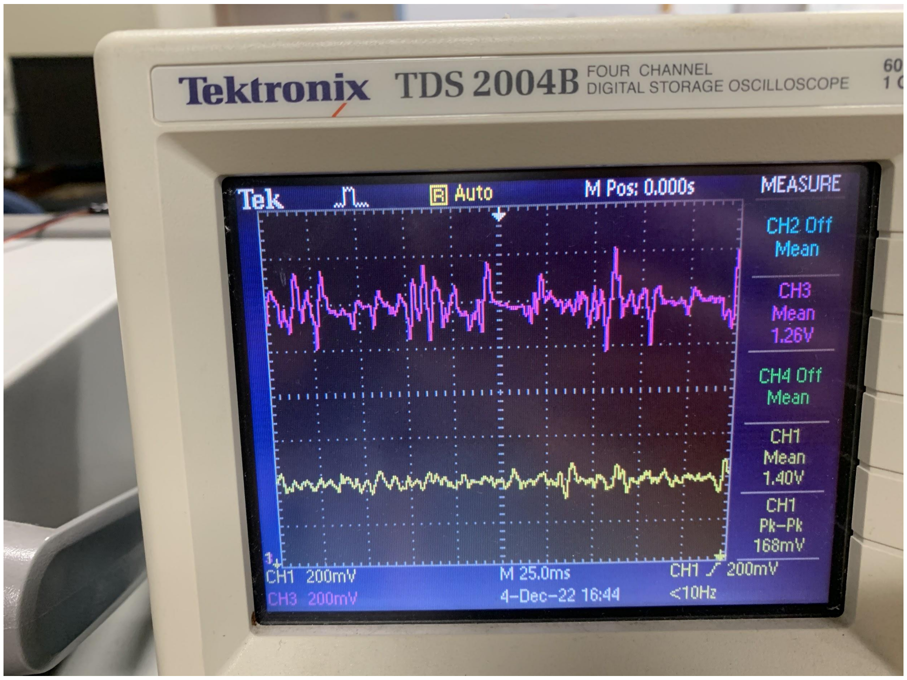


**Figure 4.3.14** Oscilloscope Output from Breadboard Showing Flexion of the Wrist

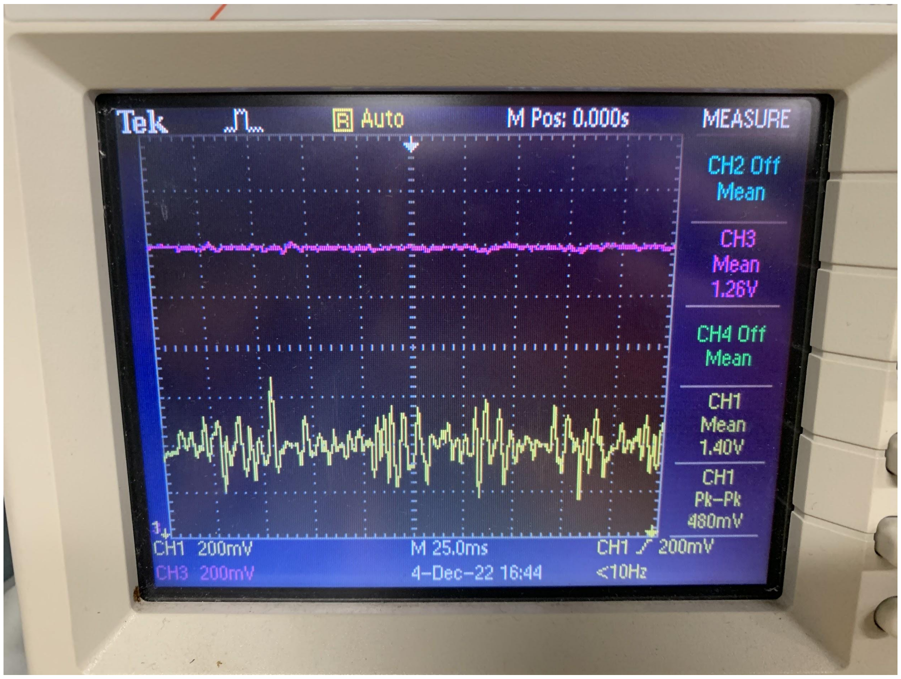


**Figure 4.3.15** Oscilloscope Output from Breadboard Showing Extension of the Wrist

#### 4.3.2.3 Solder Board

The solder board was constructed to contain both channels on an individual board. This provides a more compact design. The solder board used was a solderable breadboard, meaning the board was set up exactly the same as a normal breadboard with power rails and horizontal terminal strips. The solder board contained two AD620 instrumentation amplifiers, two LM348N quad operational amplifiers, and one LM317 voltage regulator. Only one LM317 voltage regulator was needed because it can be used for both channels and saved space on the board. An image of the finished solder board is shown in Figure 4.3.16.

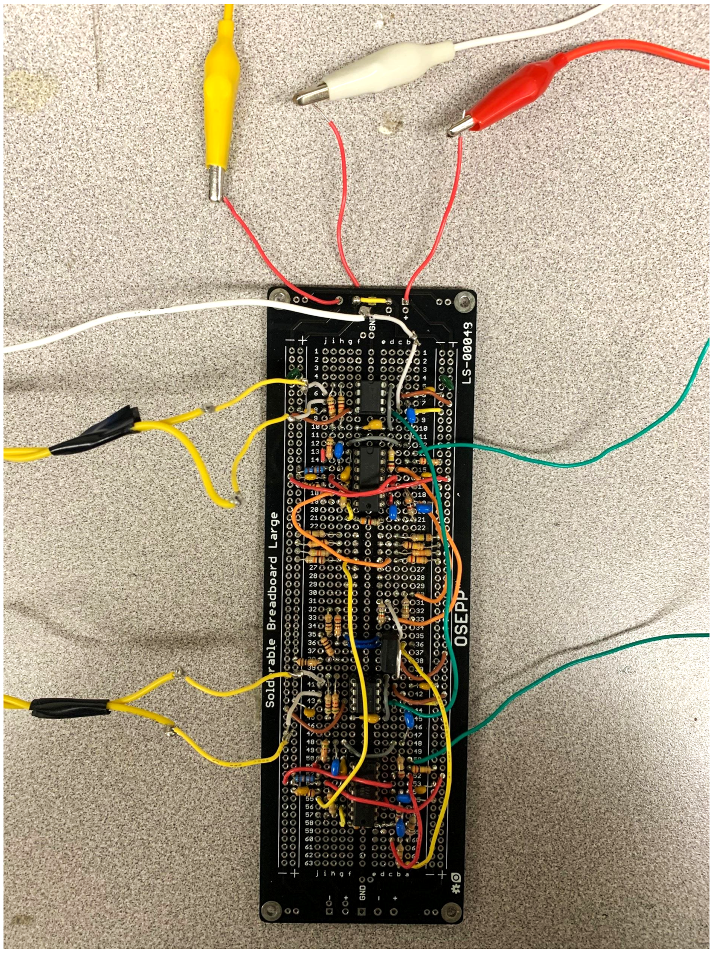


**Figure 4.3.16** Front End Solder Board

The solder board was initially tested the same way as the breadboard. The bench power supply and function generator were used and the output of each channel was observed. Once this functionality was confirmed, the solder board was tested using the electrodes and battery power. The output was observed using a MATLAB program created by one of the graduate students. This program acts as an oscilloscope and shows the output of each channel. The flexion and extension of the wrist was observed to show that the front end was working properly to provide the correct input to the microcontroller.

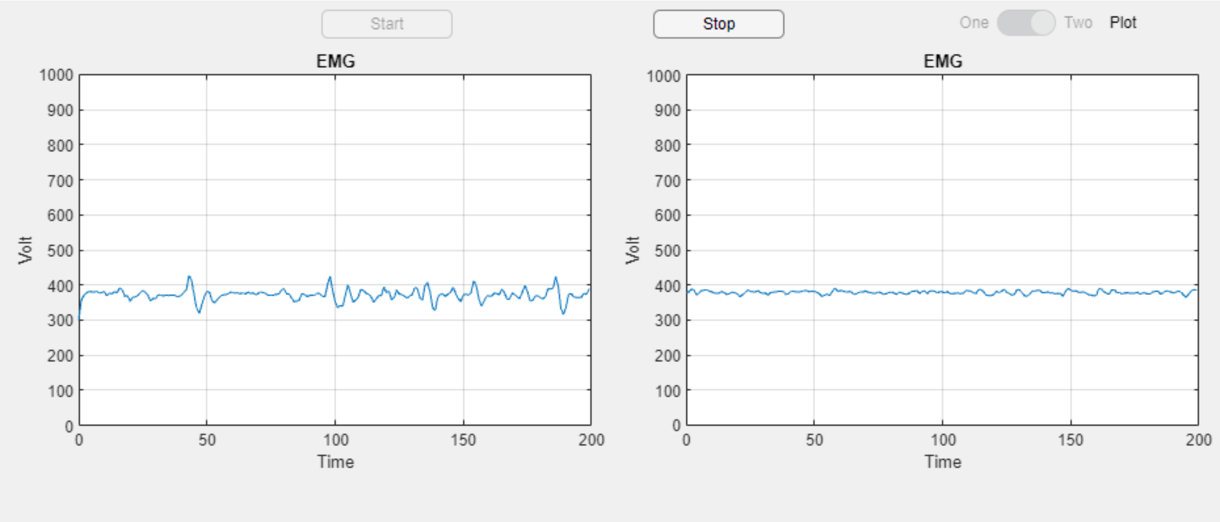
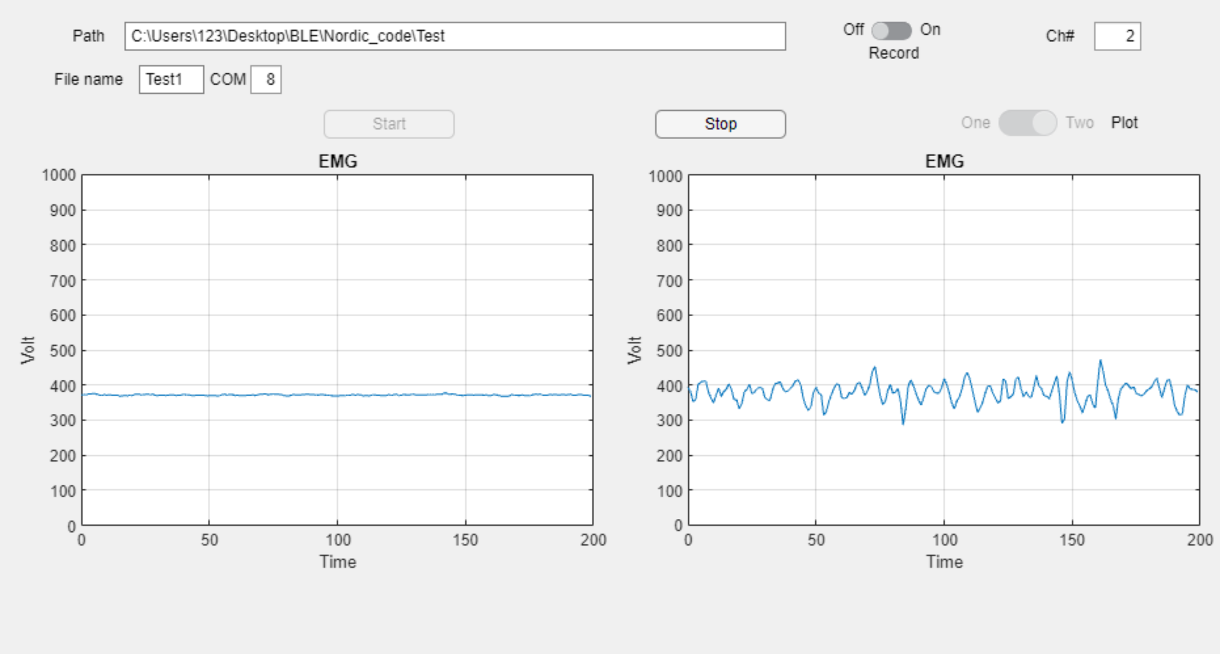


**Figure 4.3.17** Wrist Flexion Solder Board Output (Flexion Channel at Left)


Path C:\Users\123\Desktop\BLE\Nordic_code\Test

File name Test1 COM 8

Off On Record

Ch# 2

**Figure 4.3.18** Wrist Extension Solder Board Output (Extension Channel at Right)

## *4.4 Discussion*

### *4.4.1 Design*

The design of the front end circuitry can be improved by using newer IC components. For example, the AD620 is an older instrumentation amplifier that is relatively efficient and low-cost. However, there is a newer Texas Instrument instrumentation amplifier called INA128 that has the same Vmin and Vmax specifications but performs better by having a higher common mode rejection ratio (CMRR) and lower offset voltage [Texas Instruments, 2022].

The integration of a power rocker switch and single-pole-single-throw (SPST) switch for selectable gain would both be helpful additions to the front end design of the solder board. Both of these switches were ordered; however, they did not arrive in time to be implemented before the completion of this project. A rocker switch should be added to easily turn “on” and “off” the batteries. This would prevent the batteries from draining when the system is not in use. The selectable gain switch would allow the gain resistor to easily be selected from person to person. Rather than removing and replacing the necessary resistor, the switch for the desired resistor value would be turned “on” while all other resistor values remain “off.” This design would allow for a more time-efficient system when switching between users.

The design could be taken a step further and be designed using PCB software and manufactured. This would provide a much smaller version of the system that could easily be implemented into some sort of wearable like inside of the electrode armband or on a necklace around the user’s neck.

### *4.4.2 Testing*

Testing of the front end circuit could be improved by making sure the MQP lab space has the proper equipment needed for testing. The front end circuit test setup had to be moved from the personal MQP lab out into a shared lab for testing. Each time testing was done for that day, the test setup had to be disassembled. It would be more time efficient and provide more consistent results if the test setup was not moved around and the same test equipment was used.

As for the process of test steps, the procedure could have definitely been improved when assembling and testing the solder board. The solder board was constructed all at one time. This was because the solder board was constructed over Thanksgiving break when there was no access to test equipment. Looking back, it would have been more beneficial to construct the solder board in stages and verify each stage of the circuit at a time before adding new components. This would have prevented debugging the entire board at once, which makes it more difficult to identify where the problem is occurring.

# 5. Controller

## *5.1 Introduction*

In this project, two controllers are used. One controller is designated as the central node and the other is designated as the peripheral node. The peripheral node is used to digitize the signal from the analog front end and send the information to the central node via BLE. Once the data has been received on the central node, it is sent to the MATLAB pong app through a serial port. In previous iterations of the project, the Nordic nRF52840-DK boards were used. However, for this project, alternative controllers are considered.

## *5.2 Design*

### *5.2.1 Design Options*

There are three options for alternative controllers to the nRF52840-DK:

1. Arduino Nano 33 BLE
2. Adafruit Feather nRF52840 Express
3. Adafruit ItsyBitsy nRF52840 Express - Bluetooth LE

The pros and cons of the three options are listed in Table 5.2.1. The Arduino Nano 33 is the second largest controller of the three. However, it contains an integrated IMU that can provide additional data for game control or rehabilitation purposes. The Adafruit Feather is the largest controller and contains a built-in battery connector. Adafruit ItsyBitsy is the smallest controller and contains an extra 2 MB of QSPI Flash memory.

| Name | Arduino Nano 33 BLE | Adafruit Feather nRF52840 Express | Adafruit ItsyBitsy nRF52840 Express |
|---|---|---|---|
| Power Source Options | ● USB<br>● Pin | ● USB<br>● Battery<br>● Pins | ● USB<br>● Pins |
| External Memory | N/A | N/A | 2 MB of QSPI flash |
| Additional Features | LSM9DS1TR IMU | N/A | N/A |
| Size (mm) | 45 x 18 | 51 x 23 | 36 x 18 |
| Cost ($) | 22.35 | 24.95 | 19.95 |

**Table 5.2.1** Key differences between nRF52840 development boards

The controller of choice will also need to be housed appropriately. If the clip option is chosen for electrode housing, then there are three options for where the controller will be housed as shown in Figure 5.2.1:

1. On a band
2. On the clip
3. On a lanyard

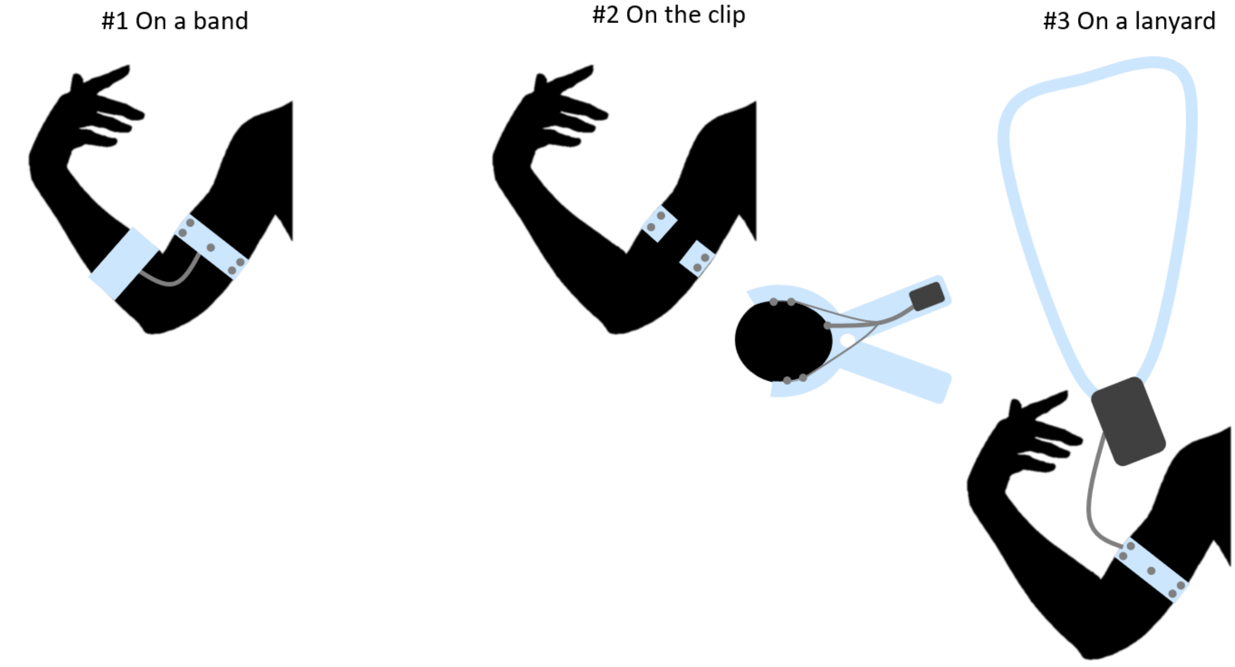


**Figure 5.2.1** Three design options for controller housing

If the armband option is chosen for the electrode housing, then either a secondary armband or a lanyard can be used to house the controller. For any of the three options, the controller housing will need to be 3D printed. Each 3D print will slightly differ for each of the three attachment sites. The pros and cons of each option are listed in Table 5.2.2.

| | Band | Clip | Lanyard |
|---|---|---|---|
| Pros | Can be placed on any limb of the body | All-in-one system | Simple to don and doff |
| Cons | Complex to don and doff | Controller may be difficult to place on the clip | Subject to movement which may affect electrode connections |

**Table 5.2.2** Pros and cons of the three design options for controller housing

For any of the three options, the controller can connect and transmit data to one of the three receivers:

1. Smartphone
2. Laptop
3. Controller

If a smartphone is used, then a mobile application will need to be made or used in order to process the data which may vary between Apple and Android products. However, if a laptop or controller is used, then a desktop application will suffice. Further explanation of the application design options can be found in Table 5.2.3.

If a controller is used to receive data, then there are two options for the controller that can be used:

1. Adafruit or Arduino MCUs
2. nRF52840-DK

The pros and cons of the two options are listed in Table 5.2.3. If either an Adafruit or Arduino controller is used, the team will need to learn how to set up and use the associated software development kits. However, if the nRF52840-DK is used, the team can use the boards that are set up as receivers in the lab. In addition, the Nordic software development kit has been well established in the lab so any problems that arise are likely to be quickly resolved through communication with graduate students.

|  | Adafruit or Arduino MCU | nRF52840-DK |
|---|---|---|
| Pros | Smaller form factor | Ready for use in the lab |
| Cons | New software development kit | Larger form factor |

**Table 5.2.3** Pros and cons of the two controller receiver options

The complete controller design options are as shown in Figure 5.2.2.

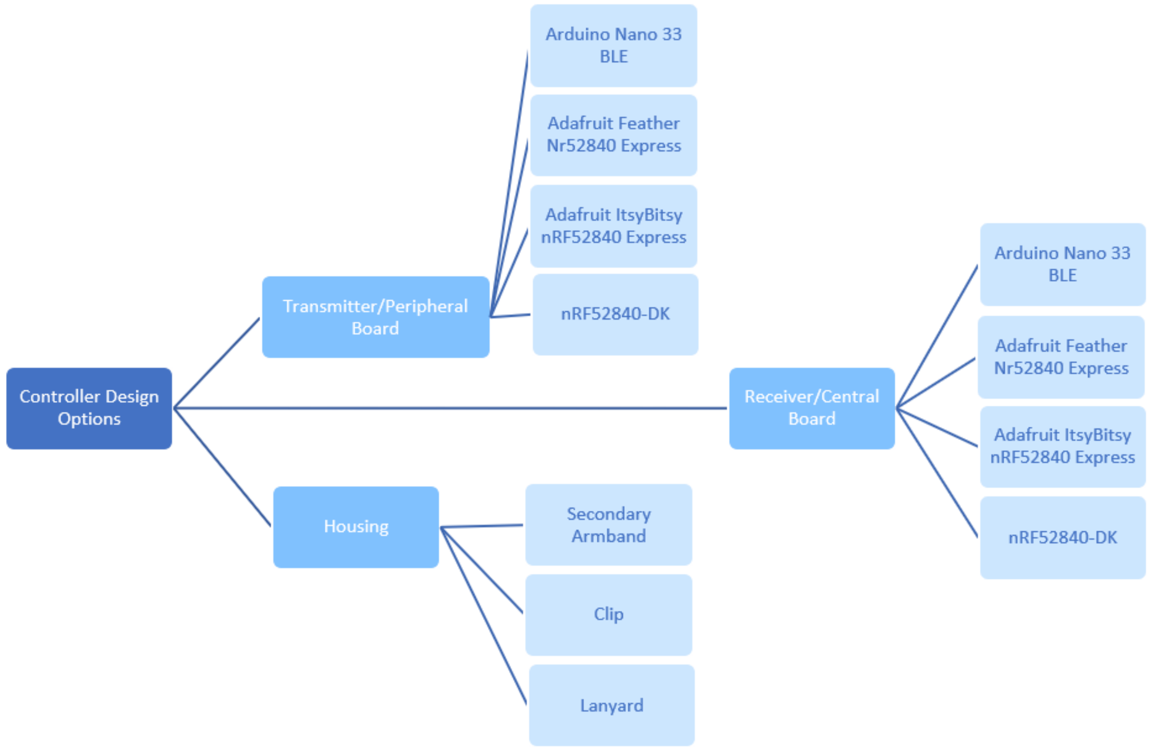

Controller Design Options
Transmitter/Peripheral Board
Arduino Nano 33 BLE
Adafruit Feather Nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Receiver/Central Board
Arduino Nano 33 BLE
Adafruit Feather Nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Housing
Secondary Armband
Clip
Lanyard

**Figure 5.2.2** Controller design options

### *5.2.2 Final Design*

The final design options for the controls are shown in Figure 5.2.3. The nRF52840-DKs were chosen to be both the central and peripheral nodes because of time restraints. The controllers had been well established within the lab which allowed the team to ask graduate students for help if issues arose. The controller housing was chosen to be a secondary armband that sat on the user's biceps-triceps. This location reduced the distance the signal from the electrodes had to travel in order to reach the peripheral node.

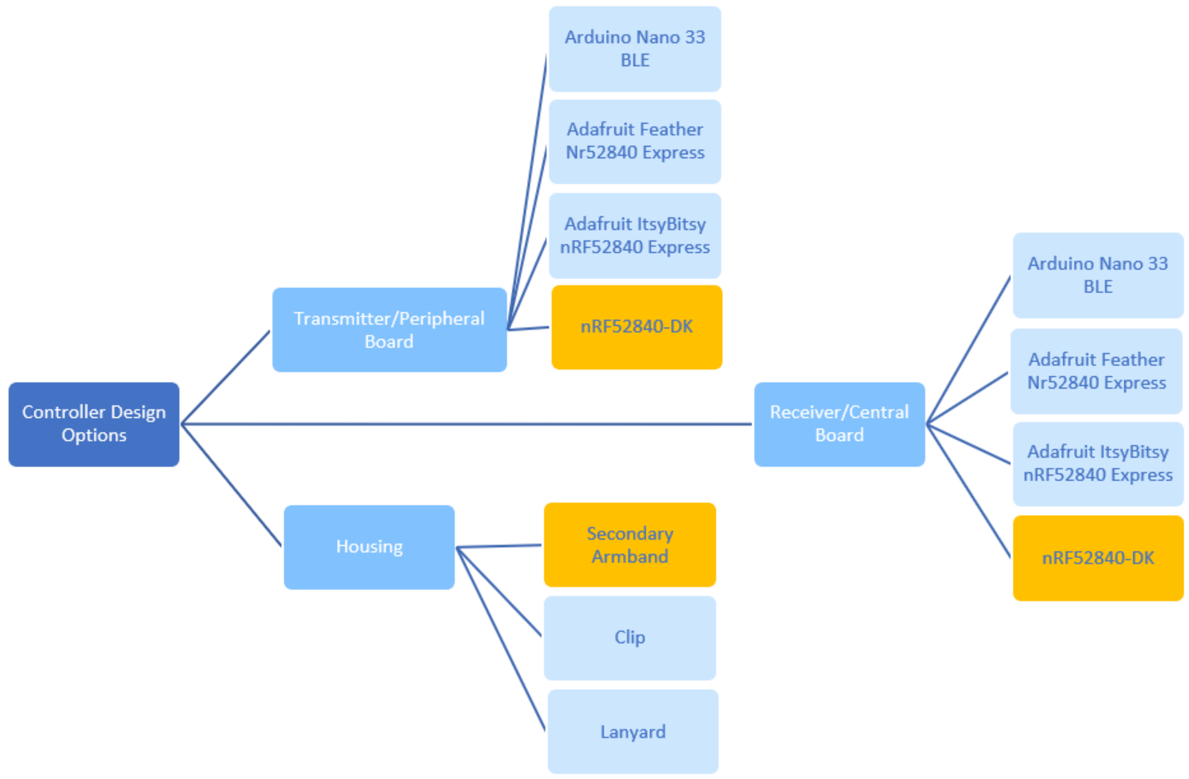

Controller Design Options
Transmitter/Peripheral Board
Arduino Nano 33 BLE
Adafruit Feather Nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Receiver/Central Board
Arduino Nano 33 BLE
Adafruit Feather Nr52840 Express
Adafruit ItsyBitsy nRF52840 Express
nRF52840-DK
Housing
Secondary Armband
Clip
Lanyard

**Figure 5.2.3** Final controller design options

### *5.2.3 Technical Specifications*

The controller can be broken up into ten elements as shown in Figure 5.2.4 where each element is either part of the MCU or SoC. To clarify, the controller is the combination of the MCU and SoC where the MCU interfaces with the SoC and provides additional options for developers. The analog-to-digital converter (ADC) takes the signal from the analog front end and converts it into a digital signal. Since the nRF52840-DK uses the nRF52840 SoC, it has an on-chip 12-bit ADC. The digitized signal is then sent to the BLE component within the SoC which then packages and transmits the signal to the central node from the peripheral node. When processing the signal, the MCU or SoC memory, general inputs and outputs (GPIO), central processing unit (CPU), peripherals, and security may be used. The central node will be connected to a PC via USB-C so that the received signal can be further processed and used in the MATLAB Pong application. The nRF52840-DK supports USB-C and battery-power options. Descriptions of each element described above can be found in Table 5.2.4. Specifics on the nRF52840-DK MCU and nRF52840 SoC can be found in sections 5.2.3.1 and 5.2.3.2.

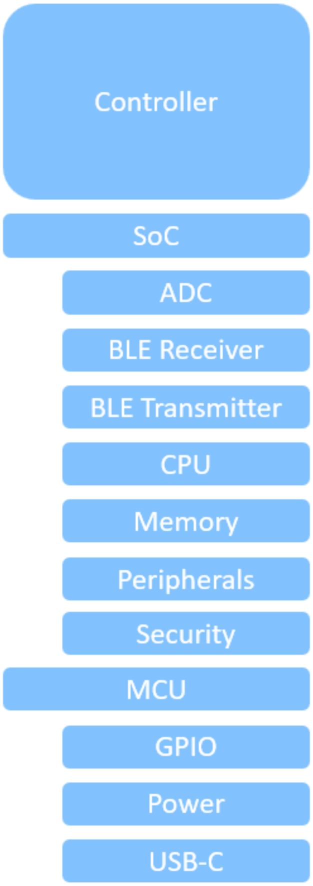


**Figure 5.2.4** Controller Elements

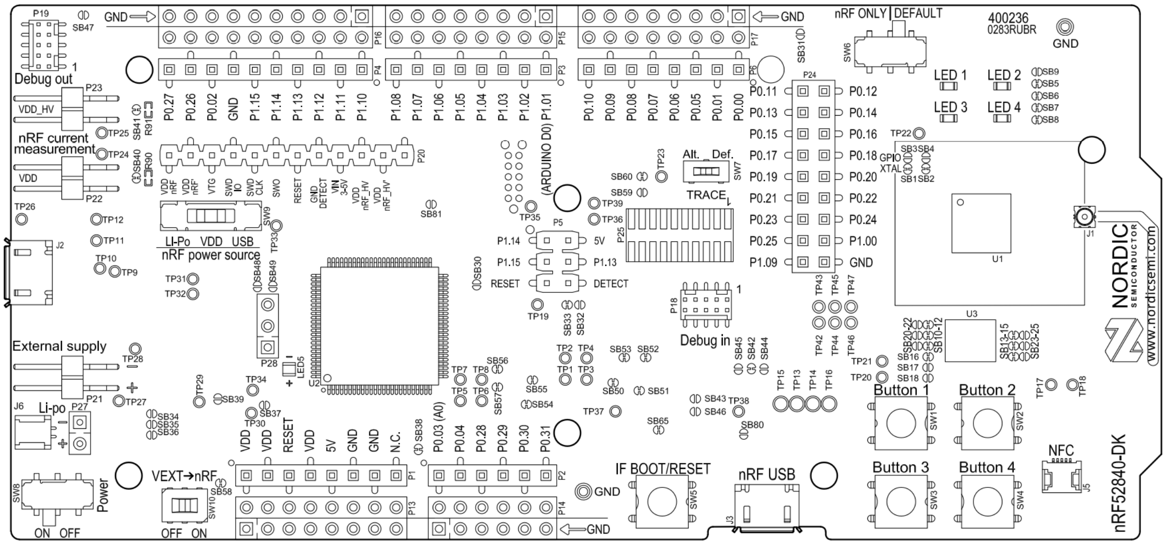
| Specification | Sub-specification | Definition |
|---|---|---|
| Memory | Random Access Memory (RAM) | Short-term volatile memory used to hold data and information of currently running programs |
| - | Flash | Nonvolatile memory used to store and or transfer persistent data and information |
| General inputs and outputs (GPIO) | - | Means to interface with the MCU and its components |
| Analog Inputs & Outputs | Analog to Digital Converter (ADC) | Component used to convert an analog signal to a digital signal |
| Wireless Communication Interfaces | Bluetooth Low Energy (BLE) | Wireless short-distance communication technique that operates in the 2.4 GHz frequency band and is intended to work for distances up to 100 m<br>On-air data rate - speed at which data can be transmitted |

| | | |
|---|---|---|
| | | TX power - power at which data can be transmitted<br>Sensitivity - lowest power signal that the receiver can process<br>Radio current consumption - current used when transmitting data |
| Serial Interfaces | Universal asynchronous Receiver/Transmitter (UART) | Asynchronous wired communication technique that is typically used for device to device communication i.e. board to board |
| - | Universal Serial Bus (USB) | Synchronous wired communication technique that that is typically used for device to device communication i.e. controller to laptop |
| Processing Units | Clock Speed | Cycles per second the CPU can execute |
| - | Floating Point Unit (FPU) | Module designed to optimize floating-point number operations |
| - | Bits | Number of bits of data that can be processed in one cycle |
| Power Saving Modes | - | Modes that restrict power to certain parts of the controller therefore limiting particular features<br>System off mode - CPU and ongoing tasks are terminated<br>Constant latency - CPU wakeup latency and PPI task response are constant<br>Low-power - automatic power management system chooses the most efficient supply option [Nordic, 2021 a] |
| Peripherals | Watchdog Timer | Timer used to detect and recover from malfunctions |
| - | Sensors | Components used to gather information about the external environment |
| - | Counter | Module that stores the number of times an event has occurred. |
| - | Oscillator | Module that produces clock signals |
| Security | Advanced Encryption Standard-Counter (AES CCM) | Modules designed to encrypt radio packets without CPU involvement [Nordic, n.d b] |
| - | Electronic Code Block (ECB) | Module used to implement the “Resolvable Private Address Generation” described in the Bluetooth Core specification v4.0 [Nordic, 2021 b] |
| - | Accelerated Address Resolver (AAR) | Module used to implement the "Resolvable Private Address Resolution Procedure" described in the Bluetooth Core specification v4.0 [Nordic, 2021 b] |

**Table 5.2.4** Controller specification definitions [Nordic, n.d. a]

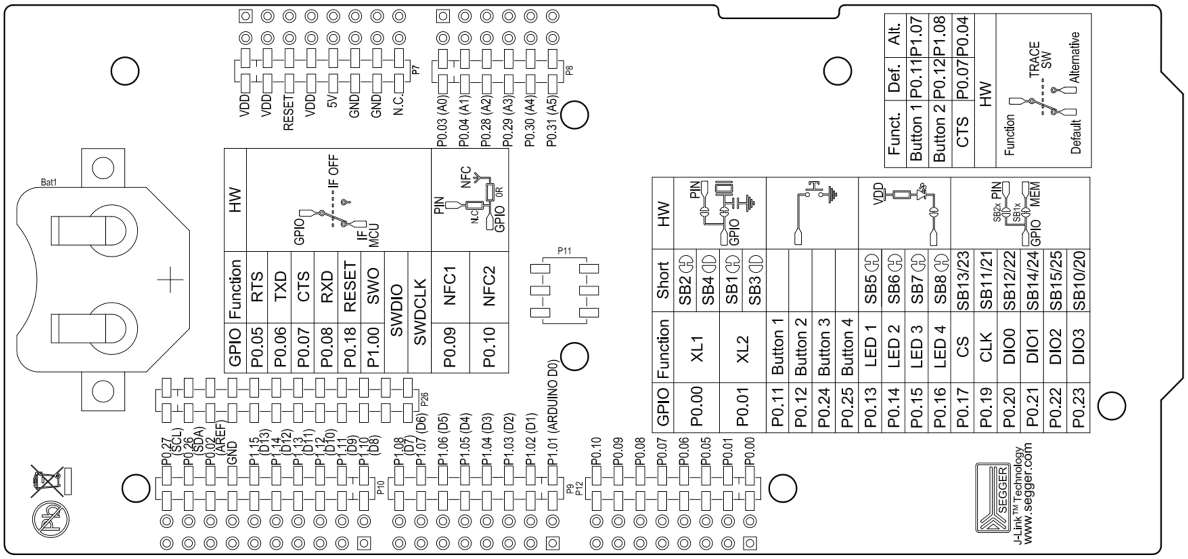
### 5.2.3.1 MCU - nRF52840-DK

The nRF52840-DK will be used as both a transmitter and receiver in the Phase I design. The MCU interfaces with the nRF52840 SoC and provides a user-friendly power supply, power source, external memory, and GPIO options. Further information can be found in the following sections and the details on the board layout can be found in Figures 5.2.5 and 5.2.6.



Debug out
nRF current measurement
VDD_HV
VDD
External supply
Li-po
Power
ON OFF
VEXT→nRF
OFF ON
Li-Po VDD USB
nRF power source
GND
P0.27 P0.26 P0.02 GND P1.15 P1.14 P1.13 P1.12 P1.11 P1.10
P1.08 P1.07 P1.06 P1.05 P1.04 P1.03 P1.02 P1.01
(ARDUINO D0)
P0.10 P0.09 P0.08 P0.07 P0.06 P0.05 P0.01 P0.00
P0.11 P0.13 P0.15 P0.17 P0.19 P0.21 P0.23 P0.25 P1.09
P0.12 P0.14 P0.16 P0.18 P0.20 P0.22 P0.24 P1.00 GND
nRF ONLY DEFAULT
400236
0283RUBR
LED 1 LED 2 LED 3 LED 4
Alt. Def.
TRACE
P1.14 P1.15 RESET 5V P1.13 DETECT
Debug in
Button 1 Button 2 Button 3 Button 4
NFC
IF BOOT/RESET
nRF USB
VDD VDD RESET VDD 5V GND GND N.C.
P0.03 (A0) P0.04 P0.28 P0.29 P0.30 P0.31
NORDIC SEMICONDUCTOR
www.nordicsemi.com
nRF52840-DK

**Figure 5.2.5** nRF52840-DK front view [Nordic, 2020]



VDD VDD RESET VDD 5V GND GND N.C.
P0.03 (A0) P0.04 (A1) P0.28 (A2) P0.29 (A3) P0.30 (A4) P0.31 (A5)
P7
P8
Bat1
P11
P26
P10
P9
P12
P0.27 (SCL) P0.26 (SDA) P0.02 (AREF) GND P1.15 (D13) P1.14 (D12) P1.13 (D11) P1.12 (D10) P1.11 (D9) P1.10 (D8)
P1.08 (D7) P1.07 (D6) P1.06 (D5) P1.05 (D4) P1.04 (D3) P1.03 (D2) P1.02 (D1) P1.01 (ARDUINO D0)
P0.10 P0.09 P0.08 P0.07 P0.06 P0.05 P0.01 P0.00
GPIO
IF OFF
IF MCU
PIN
NFC
N.C
0R
GPIO
PIN
GPIO
VDD
SB2x
SB1x
PIN
MEM
GPIO
TRACE SW
Function
Default
Alternative
SEGGER
J-Link™ Technology
www.segger.com

| GPIO | Function | HW |
|---|---|---|
| P0.05 | RTS | |
| P0.06 | TXD | |
| P0.07 | CTS | |
| P0.08 | RXD | |
| P0.18 | RESET | |
| P1.00 | SWO | |
| SWDIO | | |
| SWDCLK | | |
| P0.09 | NFC1 | |
| P0.10 | NFC2 | |

| GPIO | Function | Short | HW |
|---|---|---|---|
| P0.00 | XL1 | SB2 | |
| | | SB4 | |
| P0.01 | XL2 | SB1 | |
| | | SB3 | |
| P0.11 | Button 1 | | |
| P0.12 | Button 2 | | |
| P0.24 | Button 3 | | |
| P0.25 | Button 4 | | |
| P0.13 | LED 1 | SB5 | |
| P0.14 | LED 2 | SB6 | |
| P0.15 | LED 3 | SB7 | |
| P0.16 | LED 4 | SB8 | |
| P0.17 | CS | SB13/23 | |
| P0.19 | CLK | SB11/21 | |
| P0.20 | DIO0 | SB12/22 | |
| P0.21 | DIO1 | SB14/24 | |
| P0.22 | DIO2 | SB15/25 | |
| P0.23 | DIO3 | SB10/20 | |

| Funct. | Def. | Alt. |
|---|---|---|
| Button 1 | P0.11 | P1.07 |
| Button 2 | P0.12 | P1.08 |
| CTS | P0.07 | P0.04 |
| HW | | |

**Figure 5.2.6** nRF52840-DK back view [Nordic, 2020]

**Power Supply Options**

The MCU can be powered by one of the following six options:

1. 5V universal serial bus (USB)  for the interface MCU (J2)
2. 5V USB for the nRF52840 system on chip (SoC) (J3)
3. Lithium polymer (Li-Po) batteries (J6, P27)
4. 3 - 5V VIN pin (P20)
5. 1.7 - 3.6V External supply (P21)
6. 3V coin cell battery (Back)

For this project, both option 1 and option 6 are used to power the controller.

**GPIO**

The MCU GPIOs are available from the connectors P1, P2, P3, P4, P5, P6, and P24 as shown in Figure 5.2.7. For this project, connectors P1 and P2 were used. More specifically, P0.03 was used as the channel 1 input for the flexion signal and P0.04 was used as the channel 2 input for the extension signal. In addition, a GND pin was used.

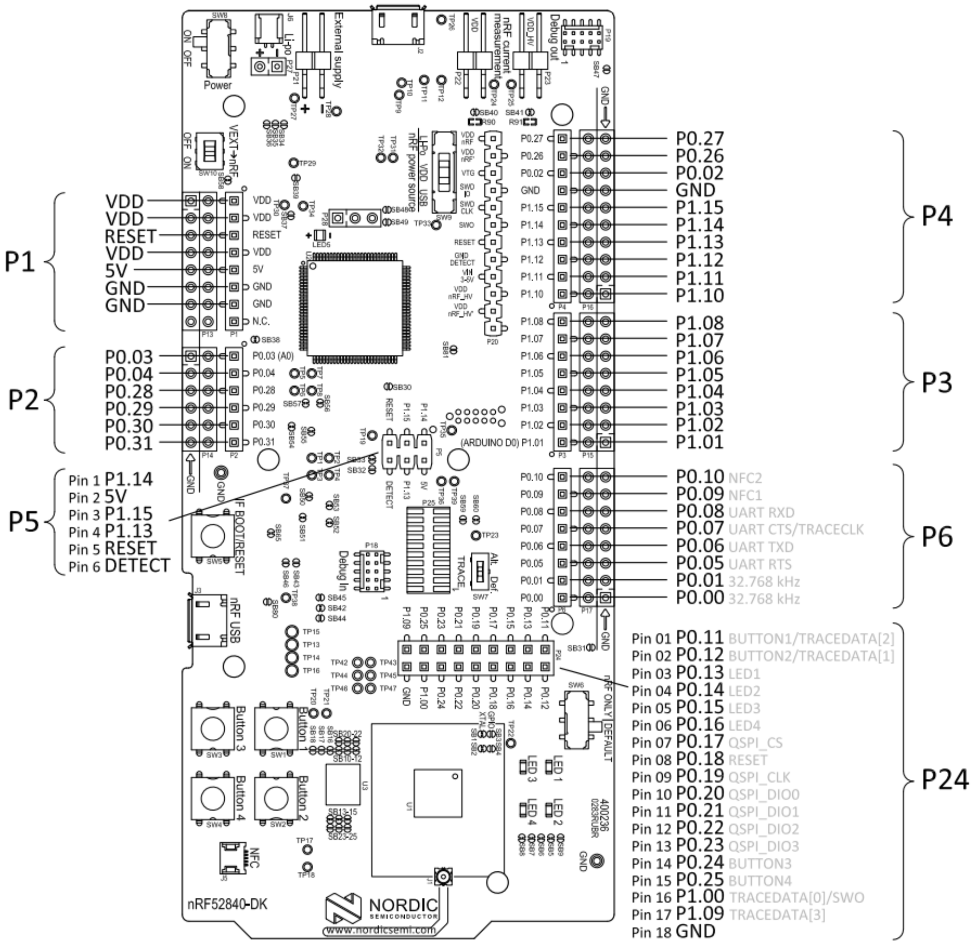


**Figure 5.2.7** nRF52840-DK GPIO connectors [Nordic, 2020]

#### 5.2.3.1 SoC - nRF52840

As mentioned previously, the MCU interfaces with the nRF52840 SoC. The specifications of the SoC are shown in Table 5.2.5.

| Specification | Details |
|---|---|
| Microprocessor | 64 MHz 32-bit Arm Cortex-M4 with FPU |
| Memory | 1 MB Flash + 256 KB RAM |
| On-air data rate | BLE: 2 Mbps/1 Mbps/500 kbps/125 kbps |
| TX power | Programmable from +8 dBm to -20 dBm in 4 dB steps |
| Sensitivity | BLE: -103 dBm at 125 kbps -95 dBm at 1 Mbps |
| Radio current consumption DC/DC at 3 V | 16.40 mA at +8 dBm TX power,<br>6.40 mA at 0 dBm TX power,<br>6.26 mA in RX at 1 Mbps |
| Oscillators | 64 MHz from 32 MHz external crystal or internal 32 kHz from crystal, RC or synthesized |
| System current consumption DC/DC at 3 V | 0.4 µA in System OFF, no RAM retention<br>1.86 µA in System OFF, full RAM retention<br>0.97 µA in System ON, no RAM retention<br>2.35 µA in System ON, full RAM retention 3.16 µA in System ON, full RAM retention and RTC |
| Hardware security | 128-bit AES CCM, ECB, AAR |
| Security subsystem | Arm TrustZone CryptoCell 310 |
| Digital interfaces | USB 2.0,<br>2 x UART, |
| Analog interfaces | 12-bit 200 ksps ADC |
| Peripherals | 5 x 32 bit timer/counter,<br>3 x 24 real-time counter,<br>watchdog timer |
| Voltage supply | 1.7 to 5.5 V |

**Table 5.2.5** nRF52840 SoC specifications [Nordic, n.d. a]

### *5.2.4 Flow of Information*

The flow of information of the controllers is shown in Figure 5.2.8. Ports P0.03 (channel 1) and P0.04 (channel 2) are connected to the signal outputs of the analog front end for both flexion and extension, respectively. The signals are then sampled with a 12-bit ADC and multiplexed into a 40-byte unsigned 8-bit int array of 40 elements as shown in Figure 5.2.9. Each sample is represented by 2 bytes and a total of 10 samples are taken per channel for each Bluetooth packet. The BLE module on the peripheral node then sends the data to the central node. The individual sample bytes are read through a ring buffer and stored in a 40-byte signed 16-bit int array of 20 elements as shown in Figure 5.2.10. Finally, the samples are packaged as a float resulting in an 80-byte float array of 20 elements as shown in Figure 5.2.11. The array is then sent to a laptop through a serial port on the computer containing the MATLAB pong application.

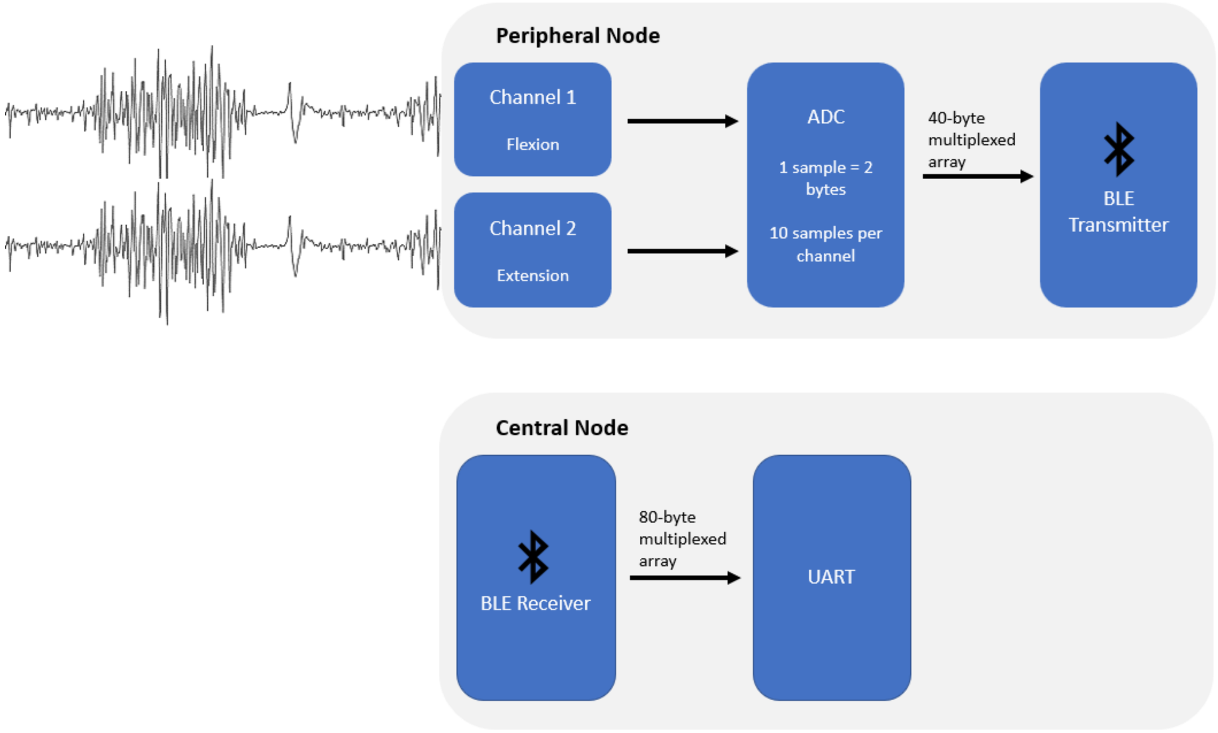


**Figure 5.2.8** Controller flow of information

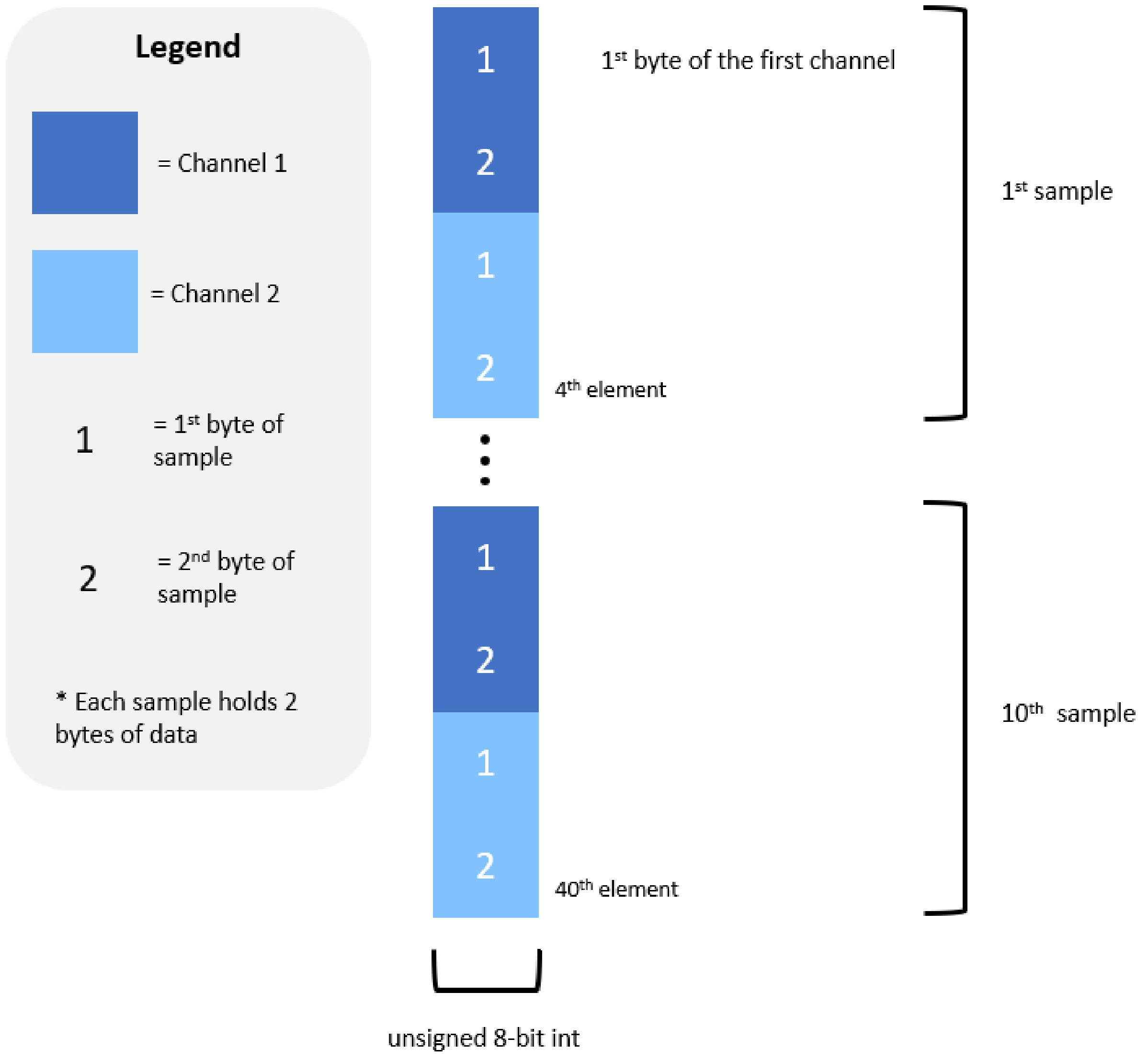


**Figure 5.2.9** 40-byte multiplexed array format

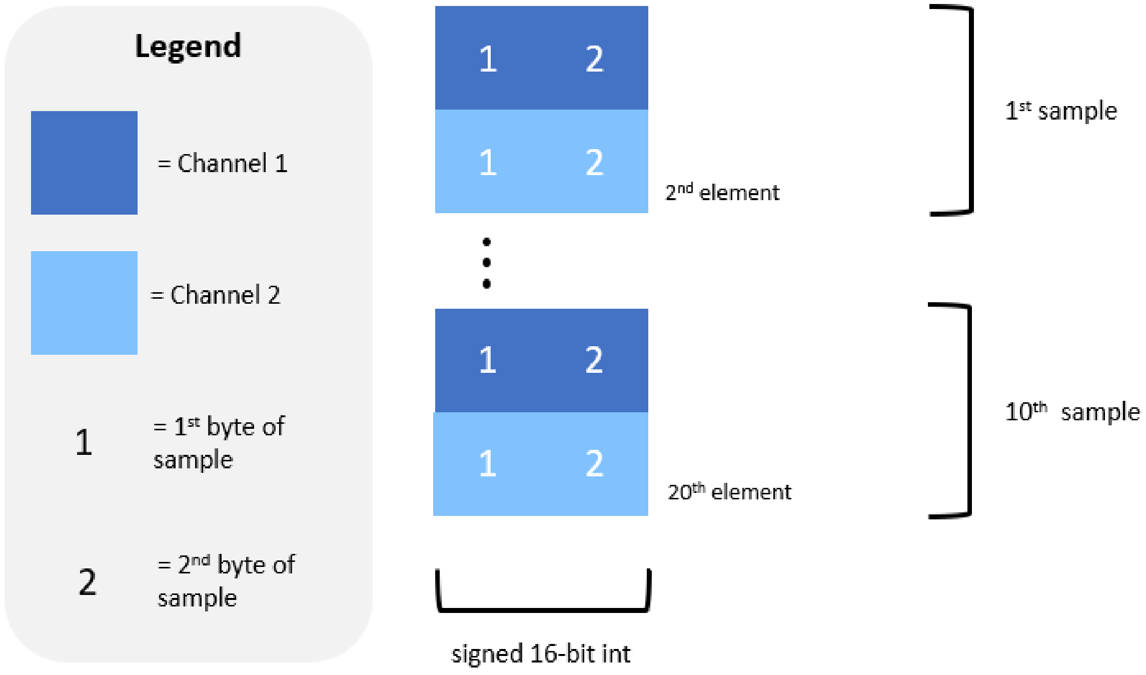


**Figure 5.2.10** 40-byte multiplexed array format

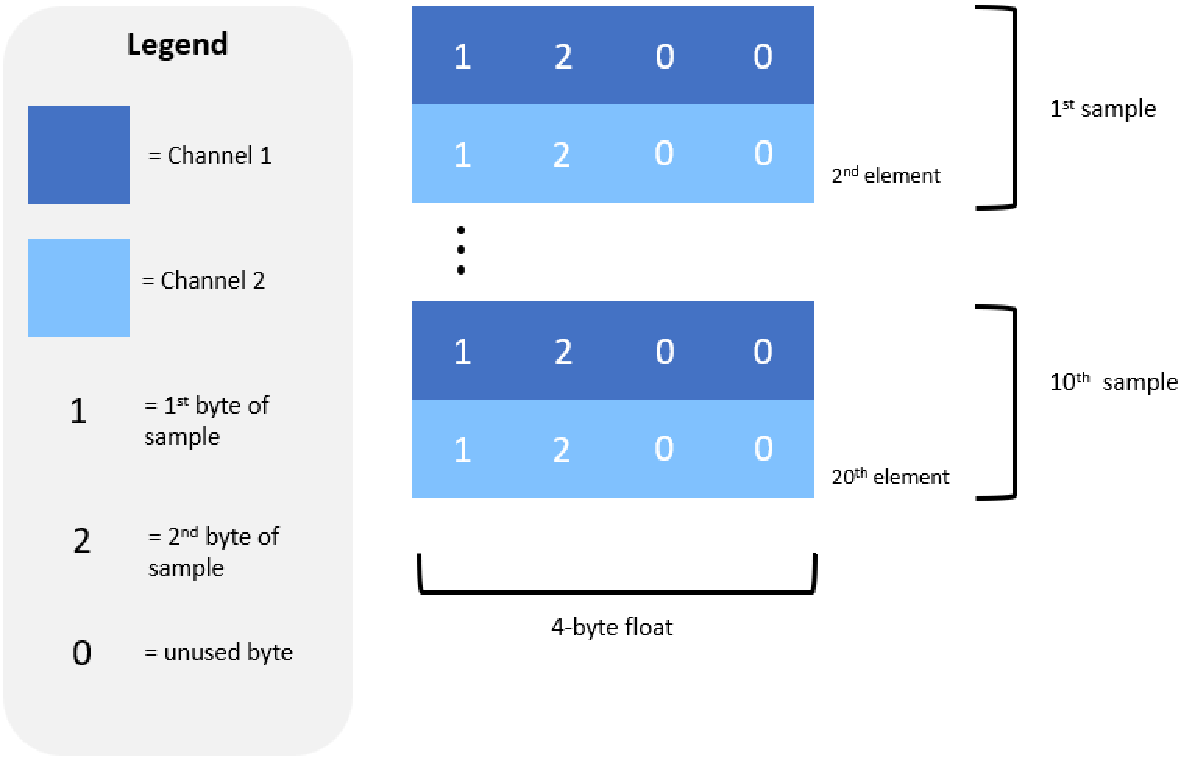


**Figure 5.2.11** 80-byte multiplexed array format

### *5.3 Testing*

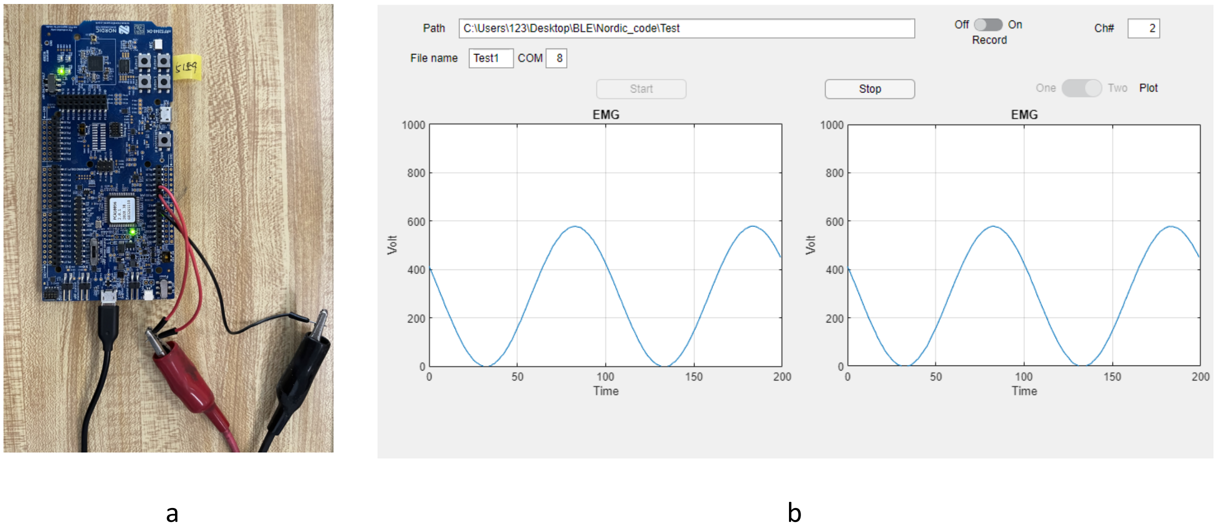
#### *5.4.1 Methodology*

To test if the controllers were properly sampling, transmitting, receiving, and processing data, a single-channel function generator was used to generate test signals that were then sampled and sent from the peripheral node to the central node. From the central node, the signal was sent through a serial port where it was captured, processed, and graphed with a Nordic Real-time MATLAB Application developed by the lab's graduate students. The EMG data are expected to be all positive values; therefore, the application does not handle negative input signals. This means that the test signal used must be positively offset.

Two tests were used to ensure data communication was functional. In the first test, a 10 Hz 1Vpp test signal with a positive 0.5V offset was input into channels 1 and 2 to ensure that the flow of information between the controllers and laptops was working. In the second test, channel 2 was connected to ground while channel 1 remained connected to the test signal to ensure that the data are properly being demultiplexed in MATLAB.

### *5.4.2 Results & Observations*

For the first test, the resulting peripheral node configuration and the signal received and displayed in the Nordic Real-time MATLAB App are shown in Figures 5.3.1a and 5.5.3.1b, respectively.



Path C:\Users\123\Desktop\BLE\Nordic_code\Test

File name Test1 COM 8

Record: Off/On

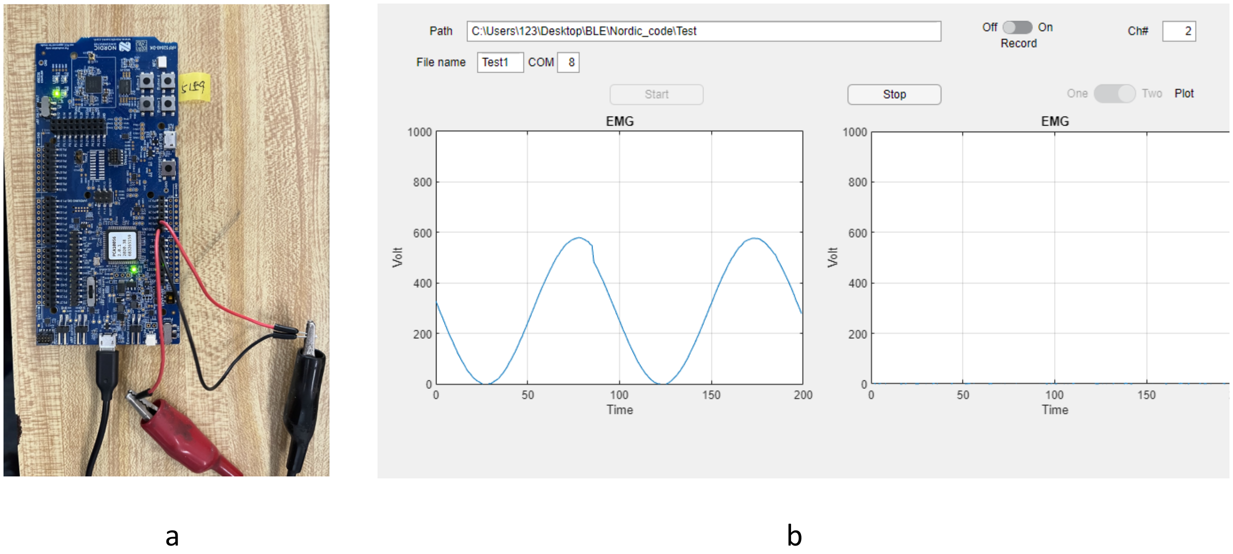
Ch# 2

Plot: One/Two

**Figure 5.3.1** Test 1 **a)** peripheral node configuration and the **b)** processed test signal graphically displayed on the Nordic Real-time MATLAB Application

In Figure 5.3.1a, channels 1 and 2 of the peripheral node are connected to the function generator outputting the test signal. A ground connection between the peripheral node and function generator is also needed to ensure a common ground. In Figure 5.3.1b, the processed signal from the function generator is displayed. The graph on the left represents channel 1 and the graph on the right represents channel 2. The x-axis of the graph designates time in ms and the y-axis designates voltage in mV. However, the y-axis is not properly scaled and therefore does not accurately represent the voltage of the test signal.

For the second test, the resulting peripheral node configuration and the signal received and displayed in the Nordic Real-time MATLAB App are shown in Figures 5.3.2a and 5.3.2b, respectively.


Path C:\Users\123\Desktop\BLE\Nordic_code\Test

Off On Record

Ch# 2

File name Test1 COM 8

**Figure 5.3.2** Test 2 **a)** peripheral node configuration and the **b) p**rocessed test signal graphically displayed on the Nordic Real-time MATLAB Application

In Figure 5.3.2a, channel 1 of the peripheral node is connected to the function generator outputting the test signal while channel 2 is connected to ground. The graph format for Figure 5.3.2b is the same as 5.3.1b. All signals were correctly received and displayed.

### *5.3 Discussion*

#### *5.3.1 Design*

The nRF52840-DKs were chosen for the final design which ultimately made the overall system's footprint larger than anticipated. During testing, two of the boards stopped working unexpectedly. Fortunately, the graduate students had extra boards. The reason for the board's failure is still unknown. Future iterations of the controller hardware should utilize smaller boards to decrease the footprint of the system. For example, the Adafruit Feather nRF board has the same processing capabilities as the nRF52840-DK and would roughly decrease the footprint by a factor of 8. In addition, the controller code should be optimized to send the data as stored when initially sampled as shown in Figure 5.2.9. Currently, the 2-byte sample data is initially stored in a 40-byte unsigned 8-bit int array of 40 elements. However, throughout the controller code, it is converted into a 40-byte signed 16-bit int array of 20 elements and then later an 80-byte float array of 20 elements. Given that the data are expected to be all positive values, unsigned int values are adequate. In addition, because the 2-byte sample data is stored in a 4-byte float, 2 bytes of data are not being used. As a result, the data sent to the laptop are two times larger than required. By adjusting the controller code to send the data as it is originally stored, as shown in Figure 5.2.9,  the responsiveness of the system may increase and the controller battery usage may decrease.

### *5.3.2 Testing*

Based on Figure 5.8b, it is shown that the controllers are properly sampling and transmitting data to MATLAB because the signals from channel 1 and channel 2 are identical in both frequency and amplitude. The frequency of the displayed signal is 10 Hz. However, the amplitude cannot be properly measured because the y-axis on the graph is not scaled appropriately. It is assumed that the amplitude of the displayed signal is equal to the test signal. Figure 5.9b shows that the data are being properly read into and demultiplexed in MATLAB because the channel 1 graph displays the test signal while the channel 2 graph displays the ground signal. The same issues discussed in Figure 5.8b are also true for 5.9b. In the Nordic Real-time application, the graphical display tended to lag more when run on a personal computer (PC) with relatively lower processing power. However, both the microcontroller code and the Nordic Real-time application can be further optimized to improve performance across all devices.

## 6. Pong Application

### *6.1 Introduction*

In order to ensure accuracy and consistency with sEMG application, it must go through several steps of processing on the software side. The incoming sEMG signal through the serial port must be filtered, taken the absolute value of, decimated by 10, and offset corrected, so that it is plotted as a signal that can be understood and used with the game. The following steps in the software would be needed:

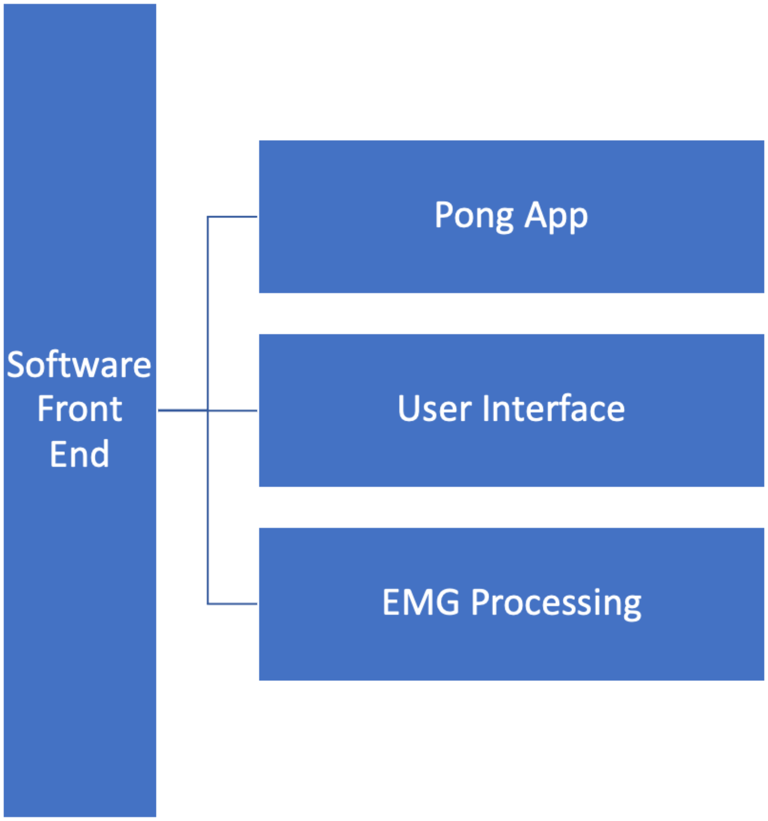


**Figure 6.1.1** Software Elements

## *6.2 Design*

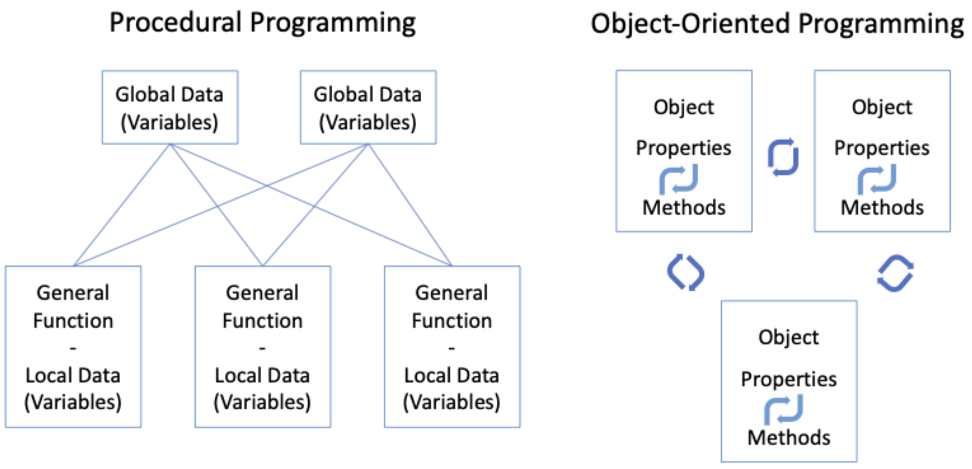

Procedural Programming
Global Data (Variables)
Global Data (Variables)
General Function - Local Data (Variables)
General Function - Local Data (Variables)
General Function - Local Data (Variables)
Object-Oriented Programming
Object
Properties
Methods
Object
Properties
Methods
Object
Properties
Methods

**Figure 6.2.1.1** Procedural vs OOP [*Establish the Fundamental Concept*, n.d.]

### *6.2.1 Design Options*

The software has different options from various aspects, which can be seen in the figure below.

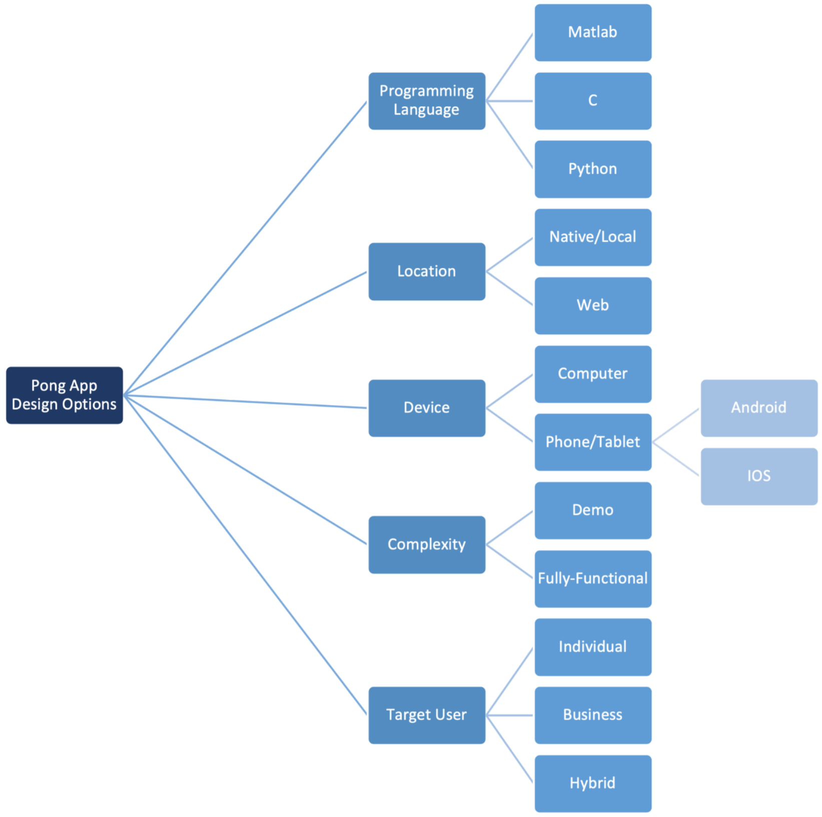


**Figure 6.2.1.2** Software Design Options

### 6.2.1.1 Programming Language

There are three useful programming languages for this application: Matlab, C, and Python. C is a low-level compiled language, while Matlab and Python are high-level scripting languages. The most prominent difference Matlab and Python hold from C is that they do not require any compilation. The code is interpreted as soon as the developer executes the file, in most cases. There are benefits and challenges to this. First of all, this accounts to the developer being exempt from memory management, and able to do dynamic typing, and interactive sessions. However, this decreases the execution speed. For the score of this project, it is important to note that both options would be sufficient.

Programs in languages like Matlab and Python are significantly shorter when compared to C, and it takes less development time. On the other hand, Matlab provides a very specific environment that is aimed at signal analysis and scientific applications to be developed. This brings a significant advantage as it provides built-in functions for many of the tasks to be accomplished, as well as providing a developed and well-visualized plotting environment for signals. As for Python, it requires many different built-in, or open-source functions that can be used for various functions, making it a sufficient alternative to Matlab programming language for scientific applications.

It is important to note that Matlab has a significant cost to obtain a license, which makes it a disadvantage for the user. On the other hand, Python is a free and open-source language. C IDEs are mostly free, which makes both languages more advantageous than Matlab when it comes to the financial aspects. However, C, which is a very powerful and broad programming language, is known for its complexity. It requires very complex memory allocation algorithms, which makes it time-consuming and complicated with a requirement for mastery, but, at the same time, this makes it a more flexible and in-depth programming language for the developer. The main advantages of such compiled languages are their speed of execution and efficiency. [Fangohr, 2004]

While Python seems to be the most viable option among the three, runtime errors are very likely during the development process. The main reason for this is its simplicity. Python does not require a variable to possess a data type, which means if a variable is initialized as an integer at first, it might end up being used to store a string. This would result in runtime errors. With that said, C requires the developer to perform the garbage collection manually, which is a very complex procedure, and it does not support Object-Oriented Programming (OOP).

| | C [Difference Between C and Python, 2022] | Matlab [Advantages of MATLAB \| Disadvantages of MATLAB, n.d.] | Python [Prasanna, 2022] |
|---|---|---|---|
| Pros | Very powerful language, with a high potential to move around<br>High compilation speed | Very specific use for signal analysis and scientific applications<br>Device independent plots and Figures | More functionality with less code<br>Free and open-source<br>Easy-to-understand |
| Cons | Requires complex algorithms<br>Does not support OOP<br>Garbage Collection done manually | Costly<br>Slower than C | High memory consumption<br>Slower than C<br>Runtime Errors |

**Figure 6.2.1.3** Pros and Cons of programming languages

#### 6.2.1.2 Platform

The application will either be a web or a native app. While a web app provides cross-functionality to the app, it requires an active internet connection and the app needs to be much smaller in size. Web apps are beneficial in terms of not requiring a download or installation. The languages that could be used for web app development would be Javascript, CSS, HTML, and Python. As for the native app, it provides a much faster and more efficient structure, and it has access to more resources inside the device, such as a camera, GPS, Health app, etc., which provides greater functionality. Also, it is easier to build with more development tools, Software Development Kits (SDKs), and many more. However, if a cross-platform application were to be built, it would mean having to build everything from scratch or incompatibility with some devices.

If a native application were to be developed, there would be different options as to which device it would be developed for. If a laptop/computer was chosen, the user experience would be better, since it would provide a larger view, and make it easier to manage big tasks, as well as eliminate the need for secondary processing. If a smartphone were to be used, there would be the need to choose between the operating systems. If an app was built for iOS, this means that the application would be coded in Swift or Objective-C, which are languages that are unique to iOS app development. Among those, Swift would be more convenient and require less coding; however, it would be possible to create much faster and much more complex software using Objective-C. Also, due to the organization of the programming environment, iOS application development would be comparatively easier than Android development. There is also a considerably limited range of devices which results in less compatibility, but provides less accountability with limited screen dimensions to be dealt with. Also, the iOS programming languages use view controllers to manage an entire screen or one of its parts. iOS app development has a reliable and consistent structure, but it has the disadvantage of being incompatible with many devices on the market.

If the Android framework was aimed to be the home environment of the application, this would mean that the compatible mechanism would help target different user bases. On the other hand, this capability of operation on a wide range of devices causes device fragmentation. Also, Android devices use Java as their main programming language, which provides the benefit of being much simpler to navigate than C-based programming languages. Android IDEs use partitions instead of view controllers to manage screens. This means that the application is broken down to fragments and activities, where activity accounts for one application screen. Therefore, creating a project with multiple screens means managing dozens of activities at the same time. While Android is beneficial in terms of being open-source and cross-functionality, it is error-prone and hard to manage.

#### 6.2.1.3 Software Complexity

In terms of software options, the application could be designed either as a demo or as a fully-functional app. There are certain benefits and challenges to both.

1. *Demo Application*
    a. Functionality: The demo application would come with parts that are incomplete either in the front end or the back end of the application. With that said, the application would be functional enough to demonstrate the uses of the device and the place of the software in the final product.
    b. User-friendliness: Having a demo application would mean that the software would be hard to navigate through for the user. As mentioned before, it would lack significant aesthetics and features that would make the application easy to use for the user. Demo application would be a very primitive demonstration of what might be done with the application.
    c. Requirements: Since it would be less complex than a fully-functional application, it would require less space and memory, as well as time and effort.

2. *Fully-Functional Application*
    a. Functionality: The fully-functional application would come with many features and much higher quality. Every step of it, both in the front end and the back end would be complete. It would contain minimal bugs.
    b. User-friendliness: This version would be much more user-friendly since it would have the shape of a real application, rather than a program that lacks the physical features. It would be very easy to use, and it would be a high-level demonstration of all the features.
    c. Requirements: Since this application is much more complex to build than the demo application, it would require more space and memory, as well as time and effort.

| | Demo App | Fully-functional App |
|---|---|---|
| Pros | ● Takes up less space and memory<br>● Requires less time and effort<br>● Good enough for demonstrating the functionality | ● High quality<br>● User-friendly<br>● Fewer bugs<br>● More functional<br>● More features |
| Cons | ● Missing Parts<br>● Less User-Friendly<br>● Not bug-free | ● Takes up more space and memory<br>● Requires more time and effort |

**Figure 6.2.1.4** Pros and Cons of App Complexity Options

### 6.2.1.4 Software Tools and Design

In order to decide between different software tools and design, the first step would be to compare different software structure approaches. The program could either be written by taking a procedural programming approach or an OOP one. This decision would mainly depend on the programming language to be used. For example, if C were to be used, the Procedural Programming approach would be necessary, while OOP would be a must for Python. However, if Matlab were to be used, both options would be possible, and there would be benefits and drawbacks for each.

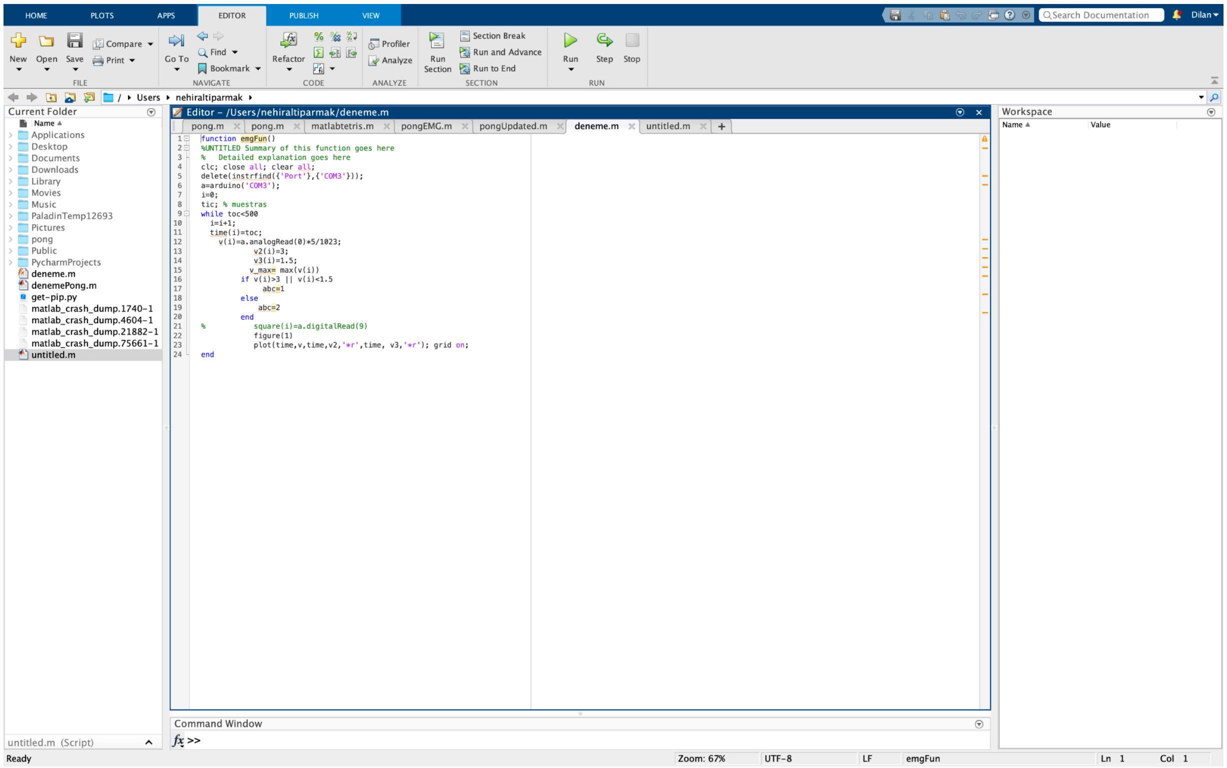
First of all, the biggest advantage of procedural programming over OOP is that the runtime is significantly faster with it on Matlab. With that said, OOP approach's encapsulation feature assists the developer(s) with avoiding collisions between different parts of the program. It also reduces redundancy in the code by creating objects and attributes for each and defining the object for each use. This structure of the OOP approach also fosters the modularity of the program too.

On the other hand, while procedural programming is complicated to extend and harder to manage complex programs, it makes the code easier to understand. Also, it is easy to invoke, modify and reuse a function.

| | Procedural Programming [Zahr, 2015] | Object-Oriented Programming [Why Use Object-Oriented Design - MATLAB & Simulink, n.d.] |
|---|---|---|
| Pros | ● Easier to understand<br>● Easy to invoke and reuse the function<br>● Significantly faster runtime with this | ● Encapsulation helps avoid collisions between different parts<br>● Reduces redundancy<br>● Fosters modularity |
| Cons | ● Complicated to extend<br>● Hard to manage long & complex programs | ● Slower runtime |

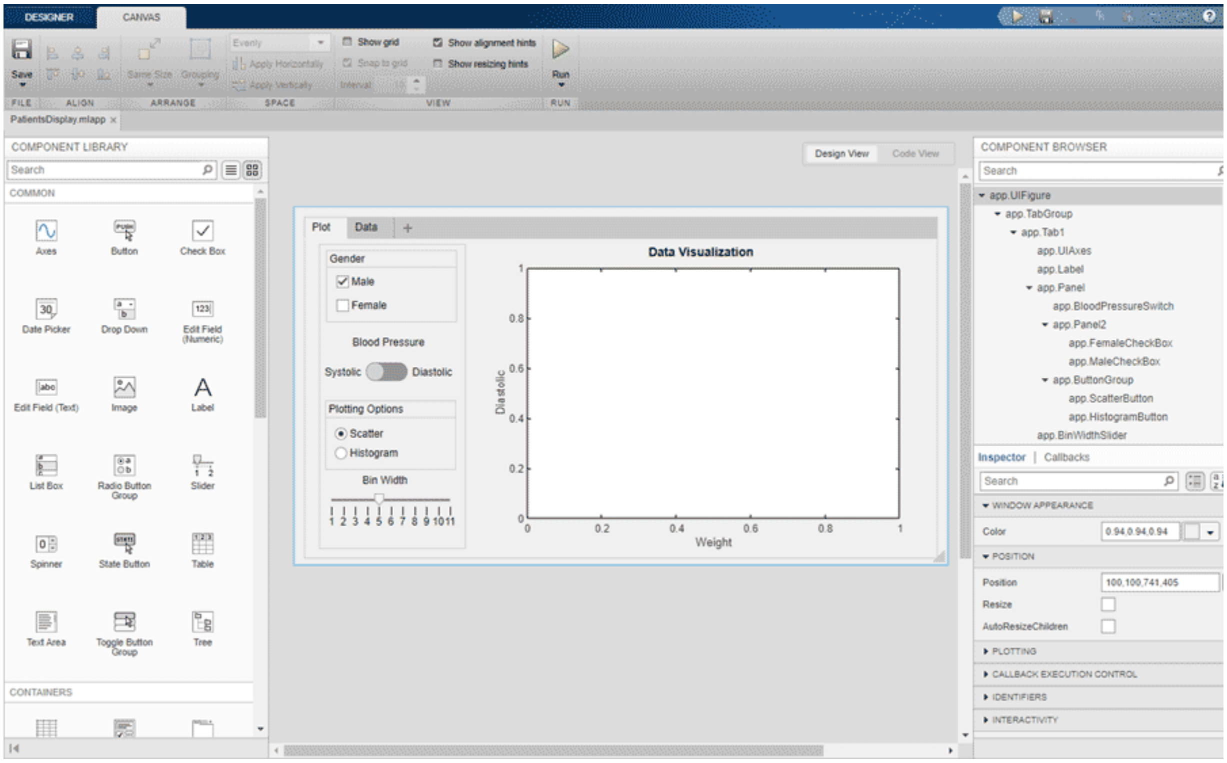
**Figure 6.2.1.5** Pros and Cons of Procedures & OOP on Matlab

For the design of the app, there are two different options for the code: Matlab Script and App Designer. Matlab Script provides the regular programming environment for the developer (Figure 6.2.1.1), while App Designer has a different interface for development (Figure 6.2.1.2).



```
function emgFun()
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
clc; close all; clear all;
delete(instrfind({'Port'},{'COM3'}));
a=arduino('COM3');
i=0;
tic; % muestras
while toc<500
  i=i+1;
  time(i)=toc;
    v(i)=a.analogRead(0)*5/1023;
            v2(i)=3;
            v3(i)=1.5;
           v_max= max(v(i))
         if v(i)>3 || v(i)<1.5
              abc=1
         else
             abc=2
         end
%           square(i)=a.digitalRead(9)
            figure(1)
            plot(time,v,time,v2,'*r',time, v3,'*r'); grid on;
end
```

## Figure 6.2.1.6 Interface for Matlab Script



Color 0.94 0.94 0.94

Position 100,100,741,405

Resize □

AutoResizeChildren □

## Figure 6.2.1.7 Interface for App Designer

Both of the options that have been mentioned above have various advantages and drawbacks. First of all, Matlab Script can be coded both in a Procedural or Object-Oriented Programming approach; with that said, App Designer only works with the OOP approach. Script requires significantly less code than the App designer, which makes it easier to understand and modify. On the other hand, the script has much less room for improvement in terms of GUI, while the App Designer is aesthetically-pleasing and involves many visual components with specific positions and physical options for each component.

| | Matlab Script | App Designer |
|---|---|---|
| Pros | ● Requires less code<br>● More efficient<br>● Very straightforward<br>● Can be used with both procedural programming and OOP | ● Aesthetically pleasing<br>● Involves a range of visual components |
| Cons | ● Has limited capacity in terms of UI | ● Hard to understand<br>● Has a very specific way of coding and it is nearly impossible to change the structure |

**Figure 6.2.1.8** Pros and Cons of Software Tools

If Matlab Script were to be chosen as the tool for programming, two different initialization and programming methods for the UI could be chosen: uiFigure() or Figure(). While uiFigure() is a function to be called when app functions are required, Figure() is for data visualization and exploration. However, both could be used for either goal. uiFigure() has more functionality for purposes such as apps with buttons, sliders etc. However, Figure() is easier to create, maintain and understand with fewer features for apps, and uiFigure() will involve longer code.

| | uiFigure() Family | Figure() Family |
|---|---|---|
| Pros | ● For App Functions<br>● Has more functionality | ● For Data Visualization and Exploration<br>● Easier to create, maintain and understand |
| Cons | ● Longer code | ● Fewer features for app creation |

**Figure 6.2.1.9** Pros and Cons of uiFigure() and Figure()

### 6.2.1.5 Target User

The application could have different options as to target users. Depending on the use of the product, the app can have two target users: individuals or businesses. If the app were to be designed for businesses, it would be aimed at physiotherapists and physicians to use. This would bring some specific features to hospitals and/or physiotherapy centers, and a specified area of use to the application and the device. Creating an application for business use might also entail being able to deal with multiple users at once, and would require a more complex database structure.

If the application was designed for individual use, it would be accessible to more people. It could also have more areas of use, as well as being easier to make. It would be significantly harder to use for multiple people and it would not contain any of the hospital/rehabilitation center specific features.

| | Individual | Business |
|---|---|---|
| Pros | ● Accessible to more people | ● Specific features |
| Cons | ● Hard to use with multiple people | ● Complicated requirements |

**Figure 6.2.1.10** Pros and Cons of Target User Options

### *6.2.2 Final Design*

For the purposes of this project, Matlab was chosen as the main programming language due to its scientific nature. As for the platform, to perform a more consistent analysis, keeping the application on a PC was a better option than mobile or web apps. Also, the program became a hybrid of the demo and the fully-functional one. While the multiplayer option is not working, the rest of the program is fully functional. This also ended up in choosing individual use over business use for the current version.

For the structure of the program, since a real-time application relies heavily on the speed of the program, a POP approach became a better fit than an OOP one. Also, this meant that we chose to create the program in the form of a Matlab Script instead of the App Designer. Also, since the application did not need too complicated visual components, Figure() was chosen over uiFigure().

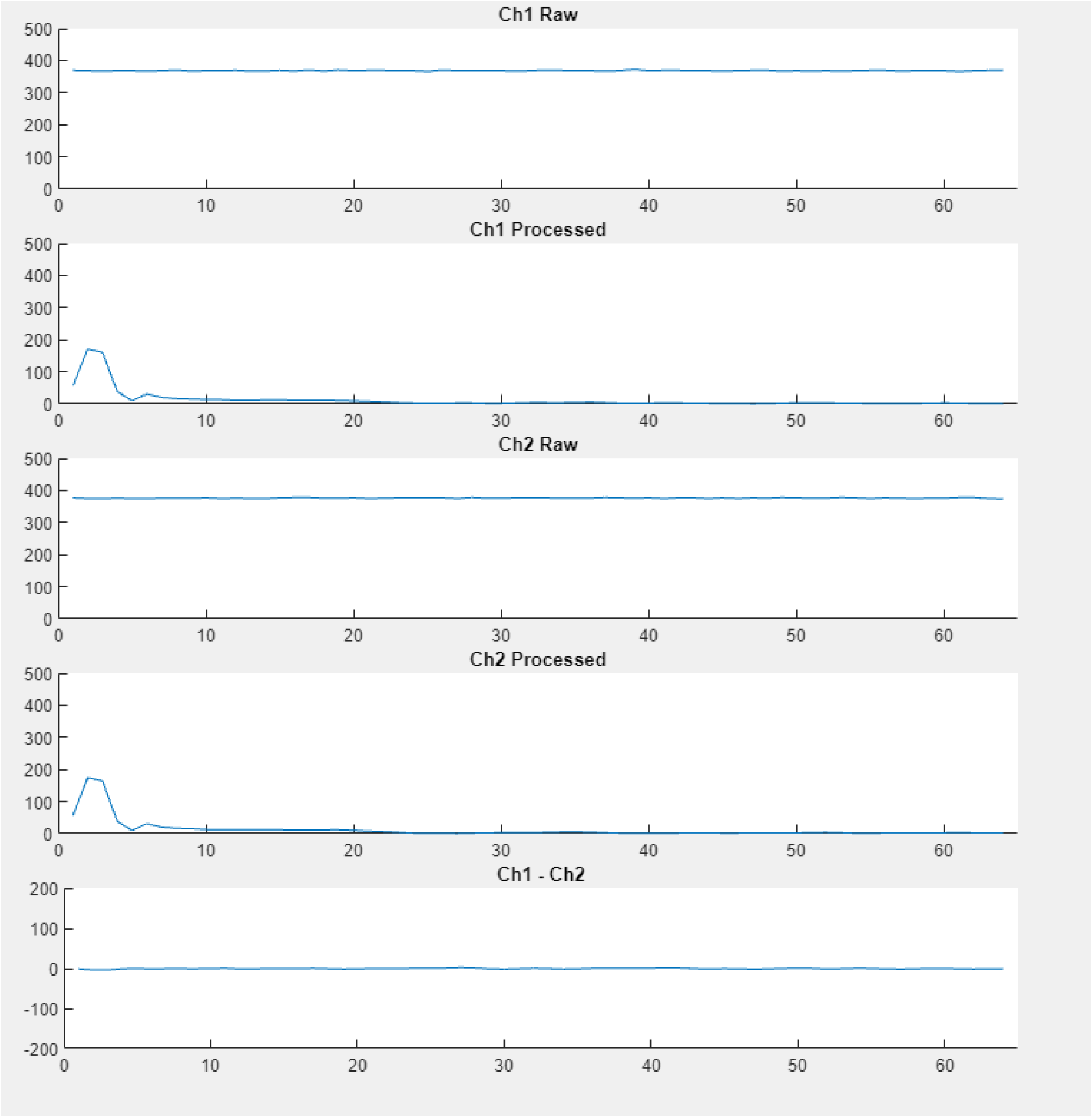


**Figure 6.2.2.1** EMG Plot Screen

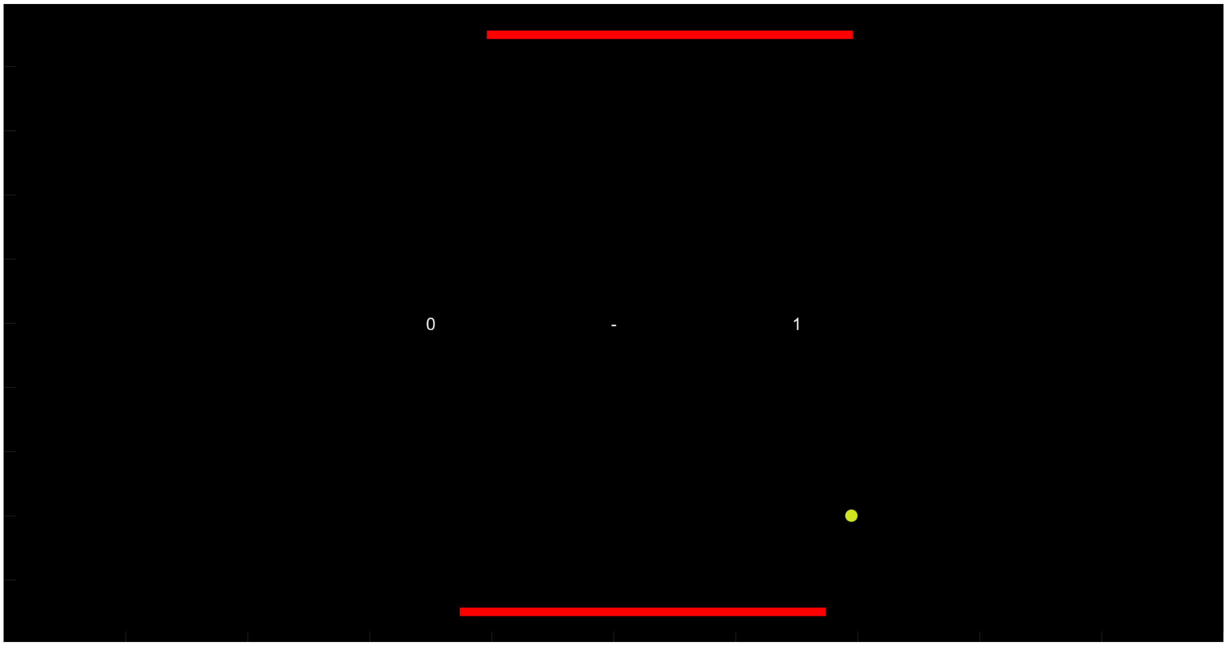


**Figure 6.2.2.2** Pong Game UI

### *6.3 Testing*

Testing for the software involved error tracing, bug fixes, and performance improvements.

#### *6.3.1 Methodology*

The methodology of the testing for the software involved the following steps:

For the overall software:

1. Create a sample non-real-time dataset using the random number generator
2. Test out the filter codes and processing function using the sample dataset
3. Create the serial port and test out the connection
4. Test out the filter codes and processing function using the data from the port
5. Test the pong paddles with the non-real-time dataset
6. According to values observed on the serial port data after processing, change the range of values for pong
7. Test the pong app with the real-time dataset

Also, frequent control points were added to the system by printing out the output variables.

#### *6.3.2 Results & Observations*

For the first two steps of the software testing, a sinusoidal signal with 20,000 elements were created. For the noise, a dataset of 20,000 elements were created and multiplied with each other to obtain the signal shown in Figure 6.3.2.1.

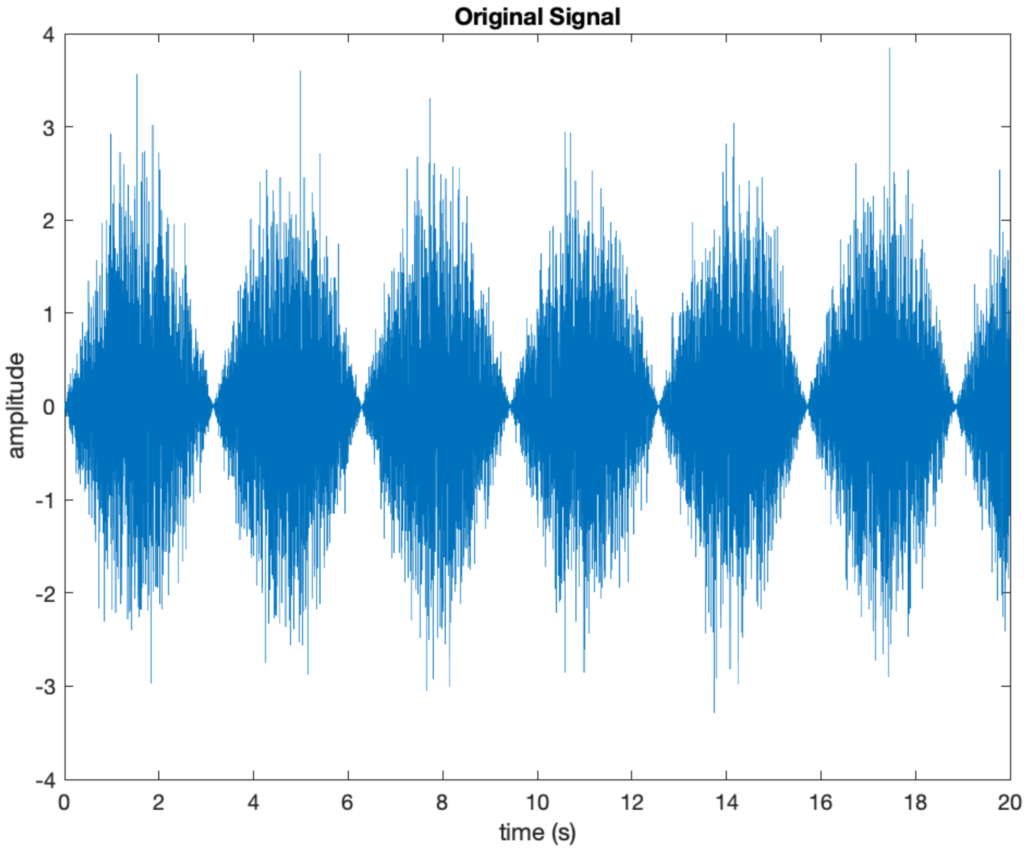


**Figure 6.3.2.1. Test Data Plot**

This data went over several steps of filtering and tuning, High Pass Filter, FIR Whitener, Notch Filter, Absolute Value, Decimation, Critically Damped-Lowpass Filter, and Offset Correction. Since the sampling rate was 1000 Hz, a lowpass filter at the beginning was not needed.

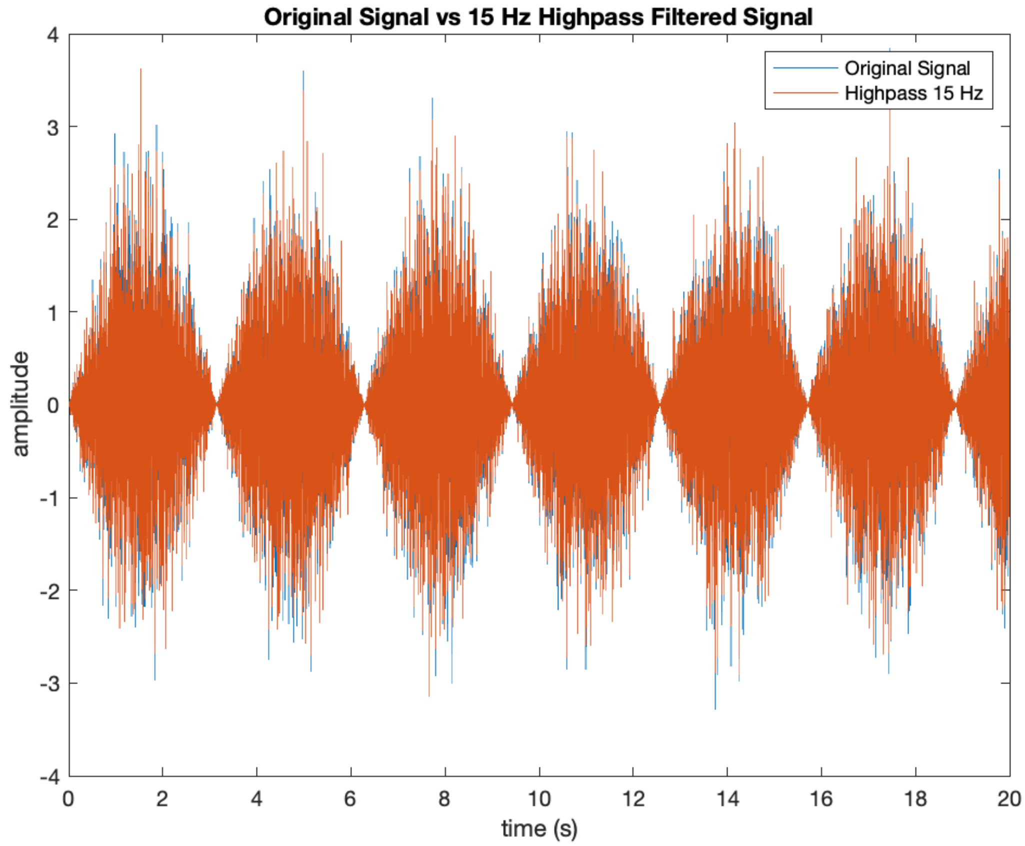


**Figure 6.3.2.2** High-pass filter. The signal was first put into a 2nd order 15 Hz highpass filter, which attenuated the signals that are lower than 15 Hz. The results of the filtering process can be seen in Figure 6.3.2.2.

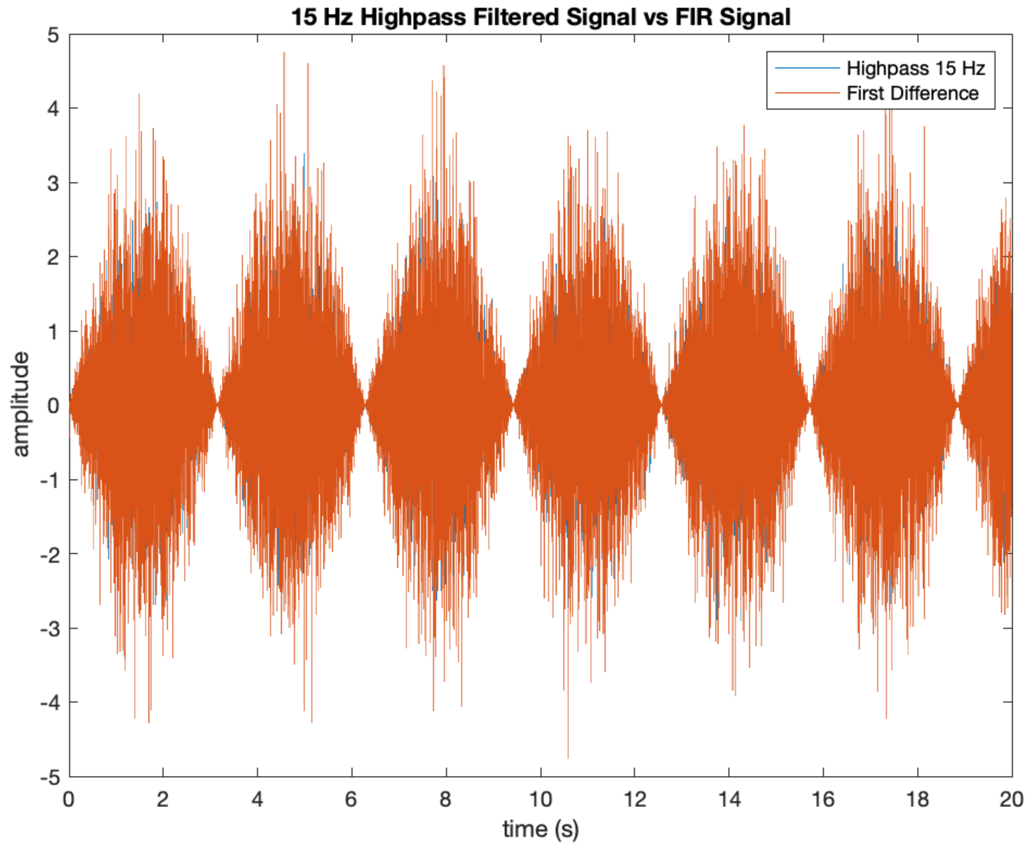


**Figure 6.3.2.3** FIR Whitener. The first difference of the data was taken through the second step, which was to apply an FIR filter to the signal with coefficients of 1 and -1. The results of the filter when compared to the previous filter (Highpass) could be observed in Figure 6.3.2.3.

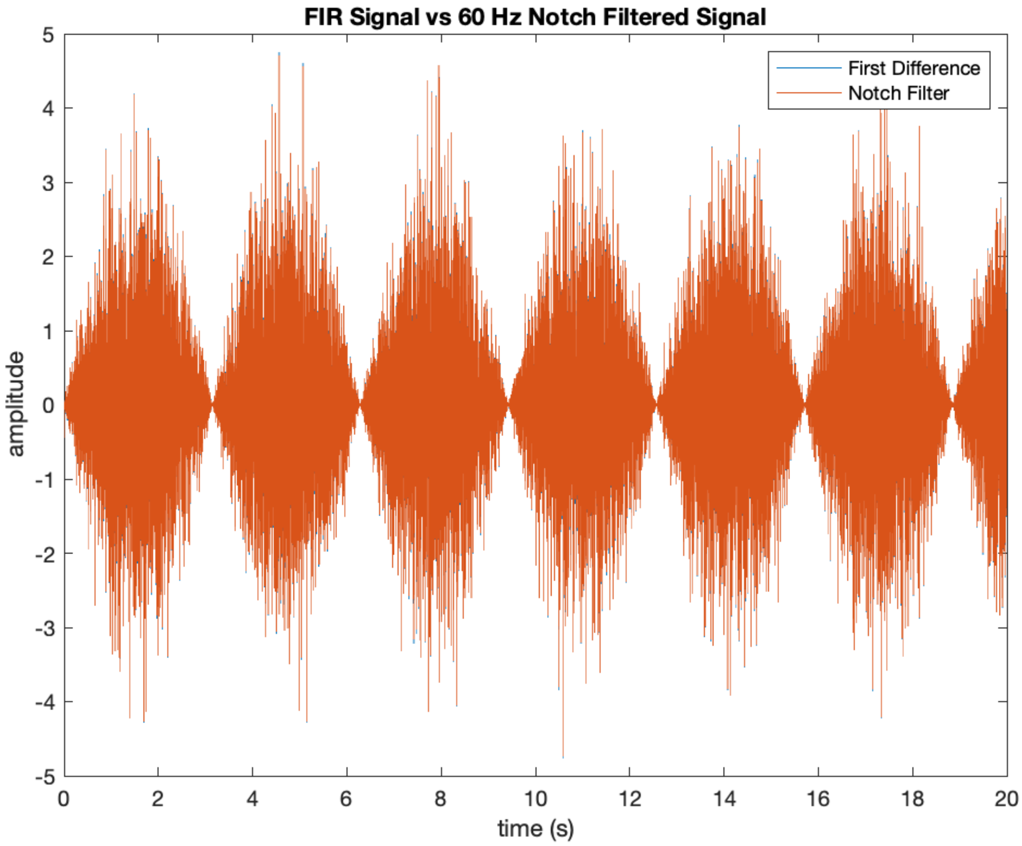


**Figure 6.3.2.4** Notch filter plot. The third step was to apply a Notch Filter of 60 Hz to the signal. The results compared to the FIR Whitener results can be found in Figure 6.3.2.4, and the frequency response of the filter is shown in Figure 6.3.2.5.

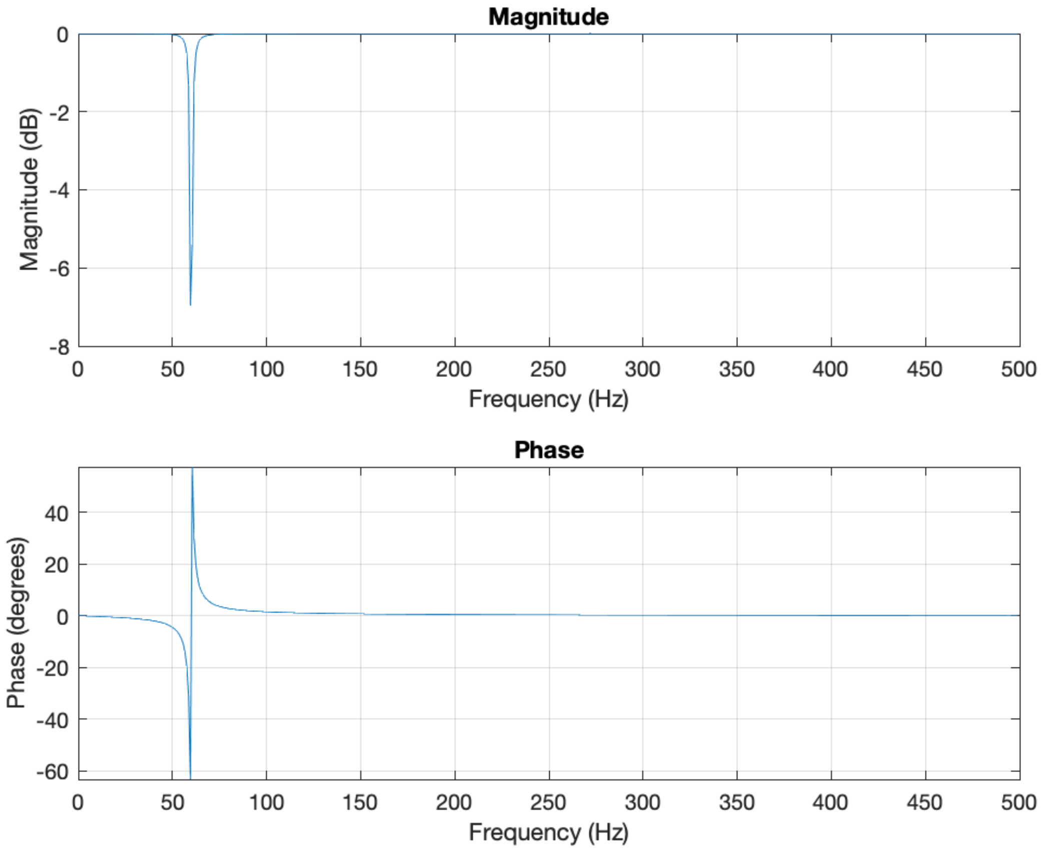


**Figure 6.3.2.5** Frequency Response of Notch Filter

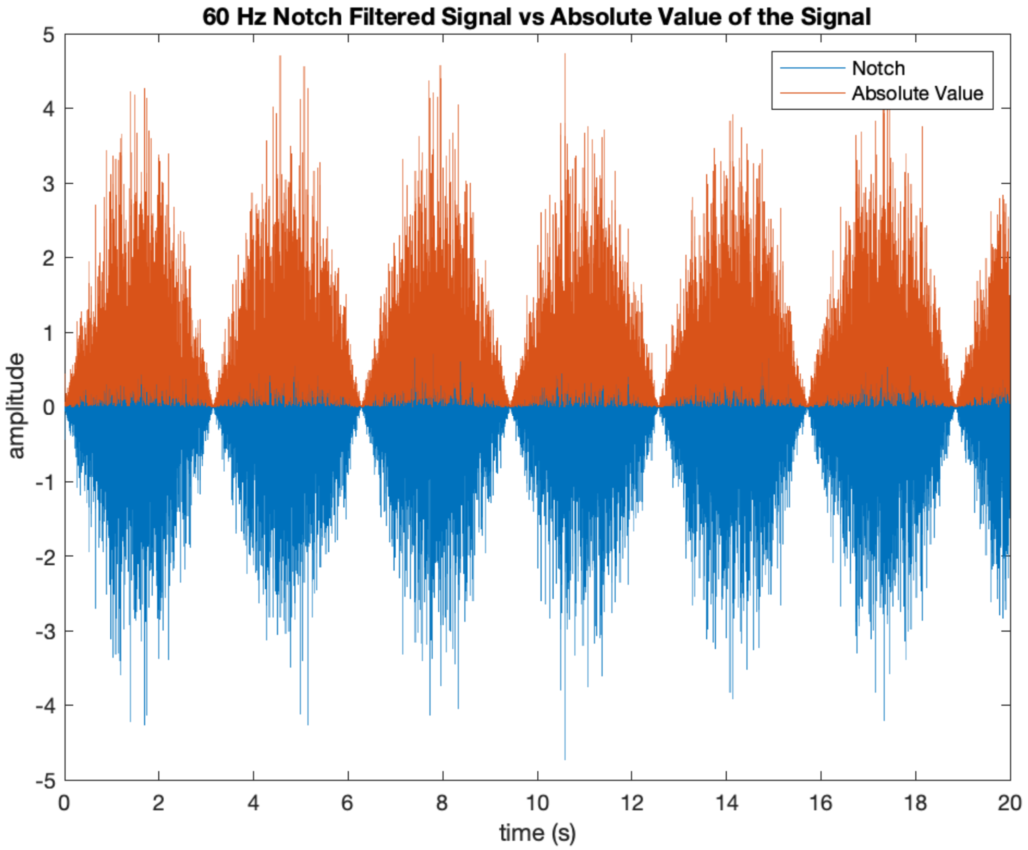


**Figure 6.3.2.6** Absolute value plot. As the fourth step of the processing, the absolute value of the signal was taken, as seen in Figure 6.3.2.6.

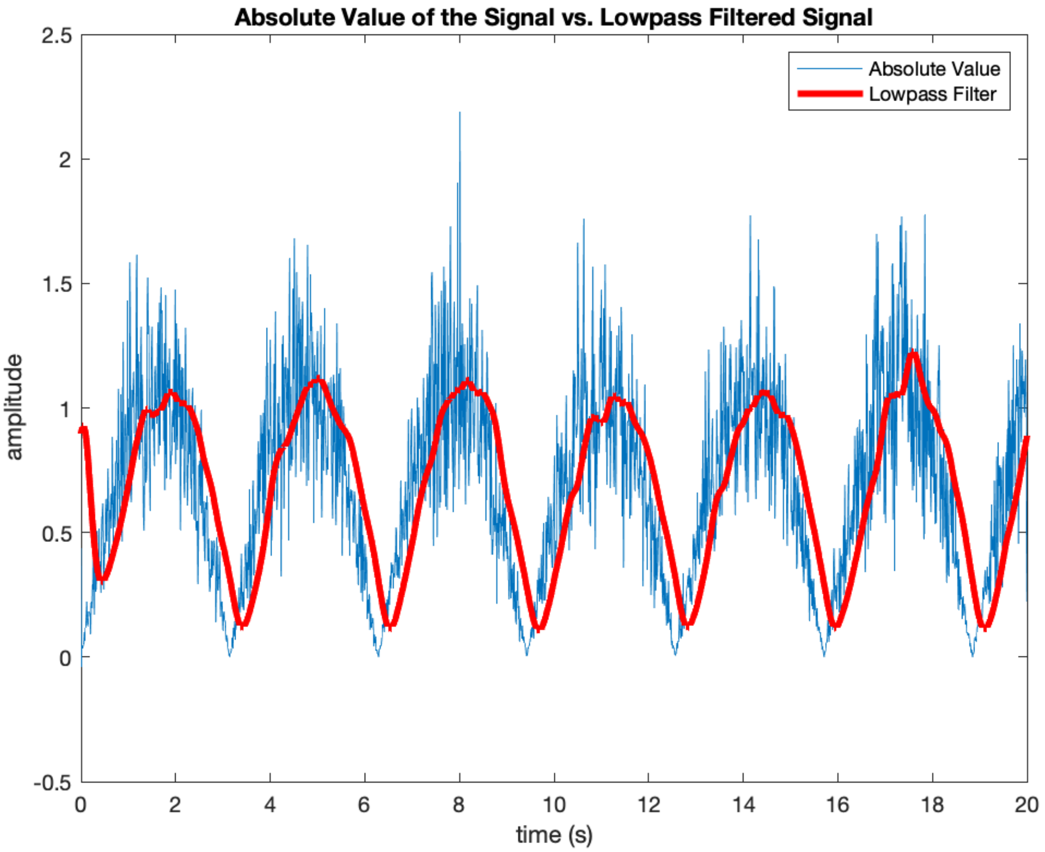


**Figure 6.3.2.7** Low-pass filter plot. The signal was low-pass filtered, using the algorithm described in the William Boyd Thesis. [cite] This filter was designed to be a critically damped fourth-order lowpass filter with 1Hz of the cutoff frequency. The results of this filter are shown in Figure 6.3.2.7, and the step response of the filter is shown in Figure 6.3.2.8.

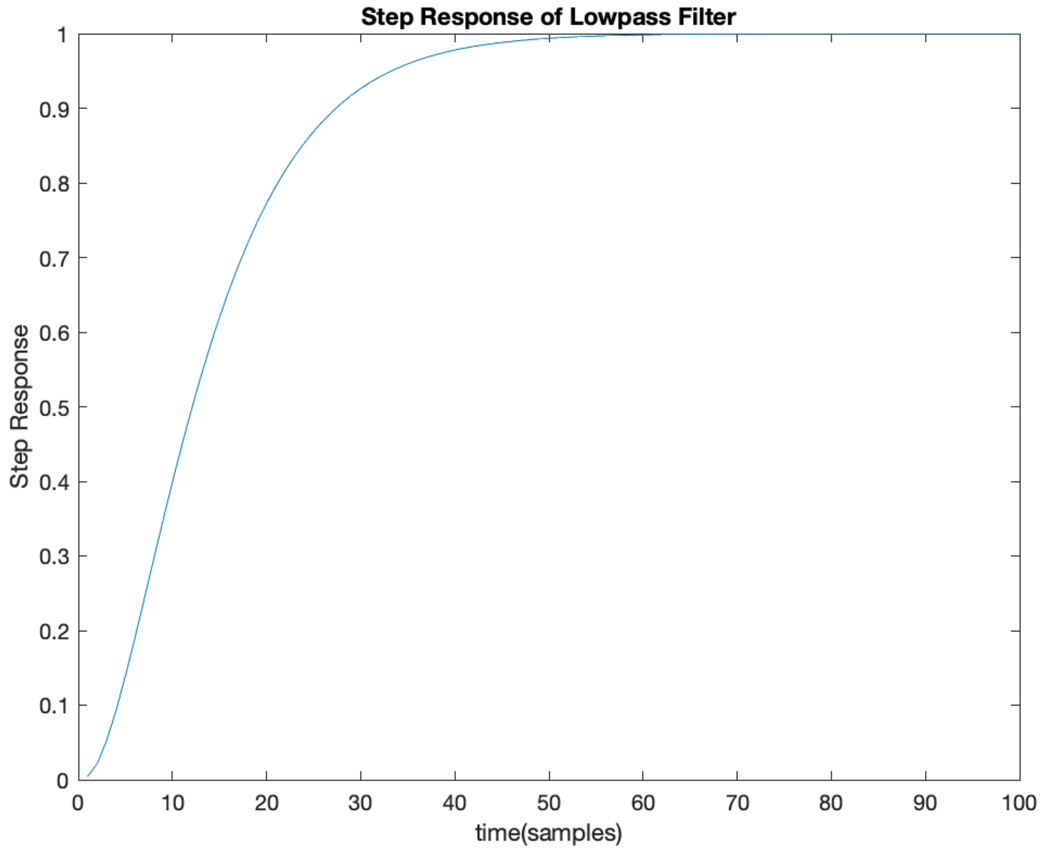


**Figure 6.3.2.8** Low Pass Filter Step Response

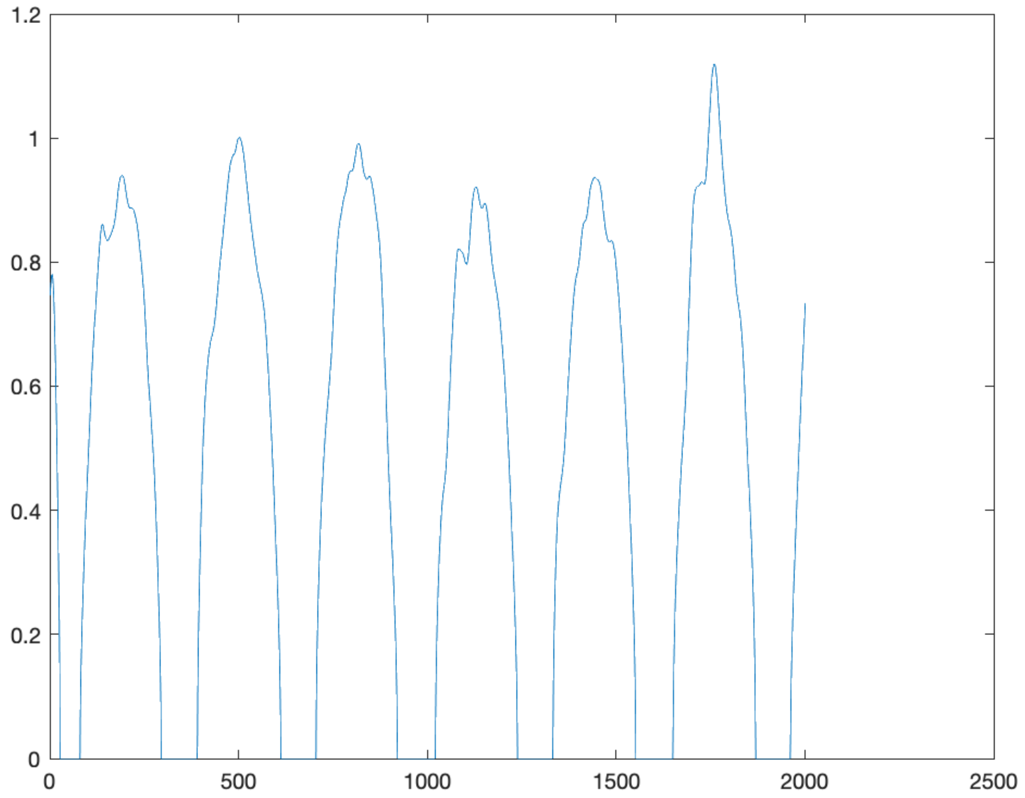


**Figure 6.3.2.9** Offset Correction Plot. In order to get more stable results for the pong game, an offset correction was performed. The noise was identified to be 0.5. The offset-corrected signal can be seen in Figure 6.3.2.9.

### *6.4 Discussion*

The software structure became easy to understand and modify through Procedural programming; therefore, it is likely that it was a better option than using the Object Oriented Programming approach.

Currently, the syntax is a mixture of the pre-2019 and post-2019 Matlab syntax. It is important to note that since the pre-2019 functions will be removed in future releases, the code should be updated accordingly. Another point to consider in terms of the syntax is that minimizing the loops and performance-decreasing code would increase the speed of the program if needed for other sorts of applications.

While the program does not necessarily need a multiplayer option, it would be better for the case of producing multiple of the same device. Also, a crucial improvement would be to increase the efficiency of the processing algorithm with further fine-tuning.

The testing process has proven that non-real-time testing of the code does not provide enough insight into the adjustments of the real-time application. Therefore, a separate testing step with possibly a real-time device emulator in between would give better results.

As a result of this testing process, several main errors have been identified and solved.

a. The program slowed down when the serial port started inputting data into the software. After the testing process was conducted, and the length of the bits transmitted was observed, it was realized that the default number of input bytes was 512 on Matlab, but this project required 40.
b. The serial port reading function read the same values from the start. After testing was conducted on the plotting function, it was found that the loop structure that the function was in made it start from the beginning of the sequence, which made the function repeat only the first bit of the sequence.

After the full testing of the software structure of the project, plenty of steps can be taken in the future to improve upon the existing framework:

1. Features for increasing the ADC resolution could be added. As mentioned broadly in the Literature Review chapter, in order to make signal acquisition and processing more effective and accurate, the quality of the raw data must be increased. For this, the methods mentioned before, such as Oversampling and Decimation or Averaging, could be used depending on the circuitry and the purpose of the overall system.
2. The visuals of the program can be improved through more detailed calculations and an approach that is more centered towards the graphics.
3. The number of loops can be reduced to increase the execution speed.

# 7. System Testing

### *7.1 Methodology*

To test if the system was properly collecting, transmitting, and receiving data, a single-channel function generator was used to generate test signals that were input into the electrodes of the analog front end. Before testing, all channel 1 and channel 2 electrodes were connected to the function generator's ground reference to ensure that external noise was minimized before testing with a live signal. Two tests were used to ensure that data collection and communication were functional.

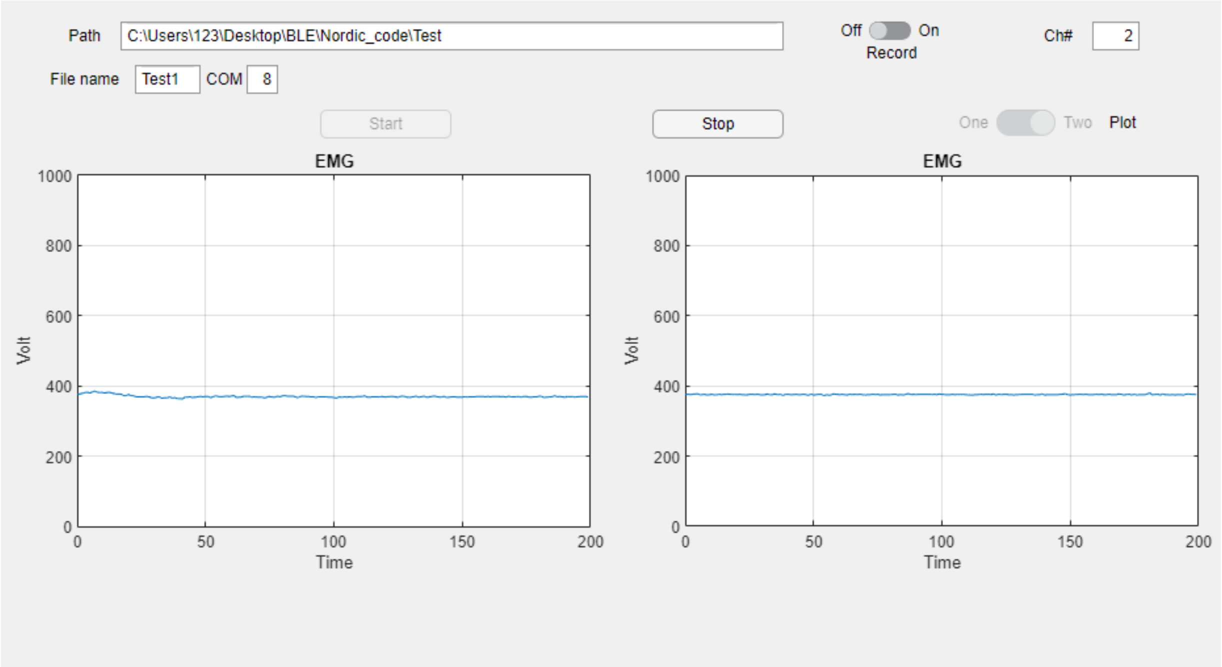
In the first test, a 150 Hz 20 mVpp test signal was input into one of the channel 1 electrodes. The second channel 1 electrode, reference electrode, and both channel 2 electrodes were connected to ground. The same methodology was repeated using a 10 Hz 20 mVpp signal. The test was then repeated for channel 2. The signals collected, transmitted, and received were then plotted in the MATLAB application developed by the lab's graduate students.

In the second test, the same methodology for the first test was used. However, the signals were plotted in a MATLAB application developed by one of the team members.

### *7.2 Results & Observations*

Before a live signal was input into either channel 1 or channel 2 electrodes, the electrodes were connected to ground and displayed in the MATLAB application developed by the lab's graduate students as shown in Figure 7.2.1 and the MATLAB application developed by one of the team members as shown in Figure 7.2.6. In Figure 7.2.1, the graph on the left and right display channel 1 and channel 2 signals respectively. The x-axis of the graph designates time in ms and the y-axis designates voltage in mV. However, the y-axis is not properly scaled and therefore does not accurately represent the voltage of the test signal. In Figure 7.2.6 the top two graphs display channel 1 raw signals and processed signals, respectively. The following two graphs display channel 2 raw and processed signals, respectively. Finally, the last graph represents the difference between channel 1 and channel 2 processed signals. The application developed by a student on the team uses similar graphing techniques to that of the application developed by the lab's graduate students and therefore the x and y axis are the same. In both Figures 7.2.1 and 7.2.6, the signals displayed are at around the same amplitude because the level shifter shifts the input signal up to about 1.8 volts. In addition, both signals do not display any noise.

For the first test, the resulting 150 Hz 20 mVpp and 10 Hz 20mVpp signals for channel 1 are displayed in Figures 7.2.2 and 7.2.3, respectively. In Figure 7.2.2, the peaks of the channel 1 signal appear to have periodic noise. In Figure 7.2.3 the noise seems to be eliminated with a lower frequency signal. The error may be related to how the data is being plotted. However, in Figure 7.2.2, the channel 2 signal appears to have the same frequency of periodic noise that was present in the channel 1 signal. The noise may be attributed to 60 Hz noise that originates from some source external to the system such as the lights. The resulting 150 Hz 20 mVpp and 10 Hz 20mVpp signals for channel 2 are displayed in Figure 7.2.4 and 7.2.5, respectively. In Figure 7.2.4, the peaks of the channel 2 signal appear to have the same periodic noise present in channel 1 of Figure 7.2.2. Similarly, in Figure 7.2.5, the periodic noise is eliminated with a lower frequency signal. However, the periodic noise that was present in channel 2 of Figure 7.2.2 is not present in channel 1 of Figure 7.2.4 indicating that channel 2 may be inherently more prone to noise than channel 1 due to a difference in the connection quality between the circuits.

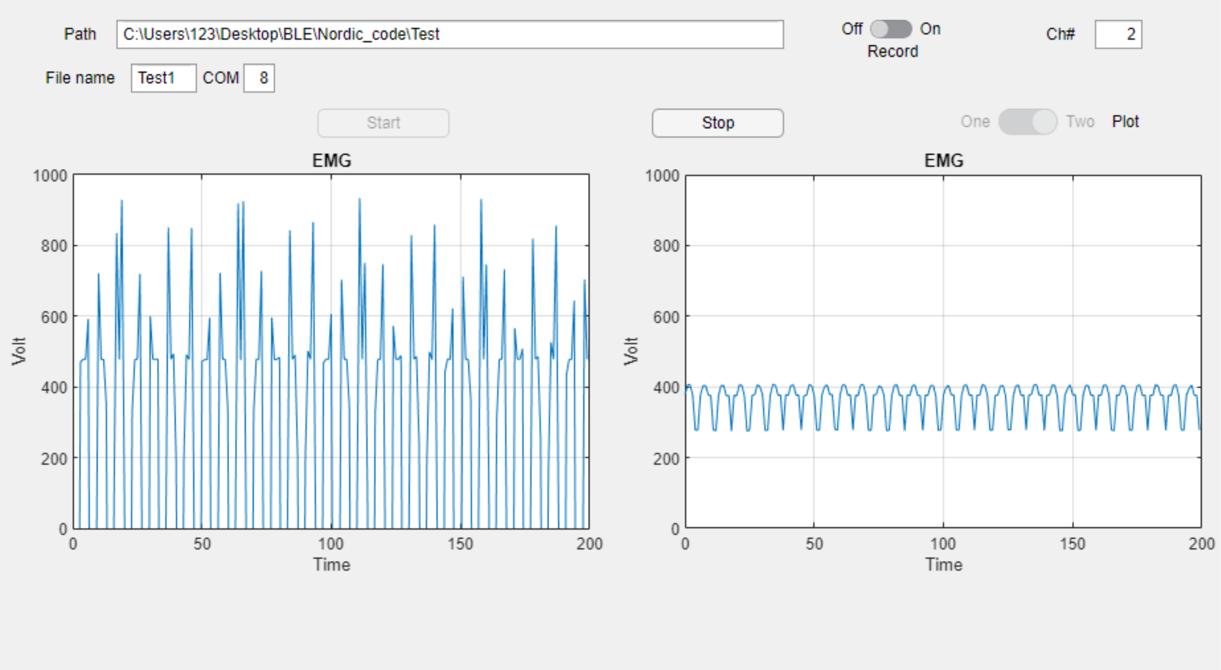


Path C:\Users\123\Desktop\BLE\Nordic_code\Test

File name Test1 COM 8

Ch# 2

**Figure 7.2.1** graphical display of all channel 1 and channel 2 electrodes grounded using the MATLAB application developed by the lab's graduate students

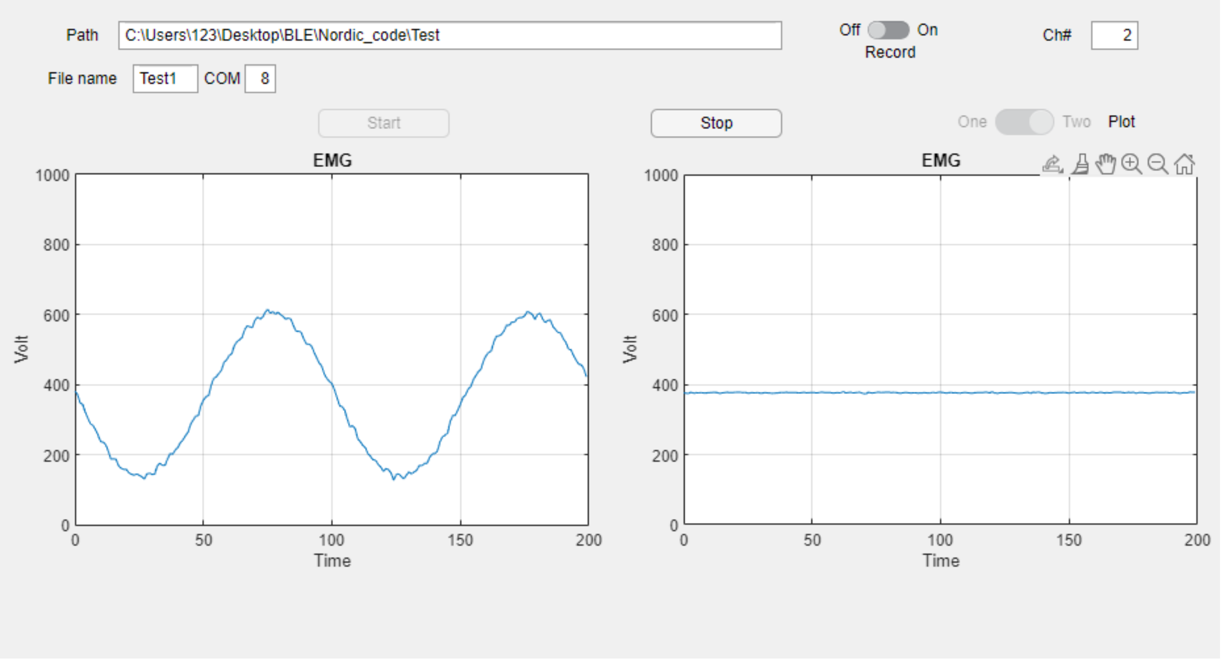


Path C:\Users\123\Desktop\BLE\Nordic_code\Test

File name Test1 COM 8

Ch# 2

**Figure 7.2.2** graphical display of a 150 Hz 20 mVpp signal sent through one channel 1 electrode using the MATLAB application developed by the lab's graduate students

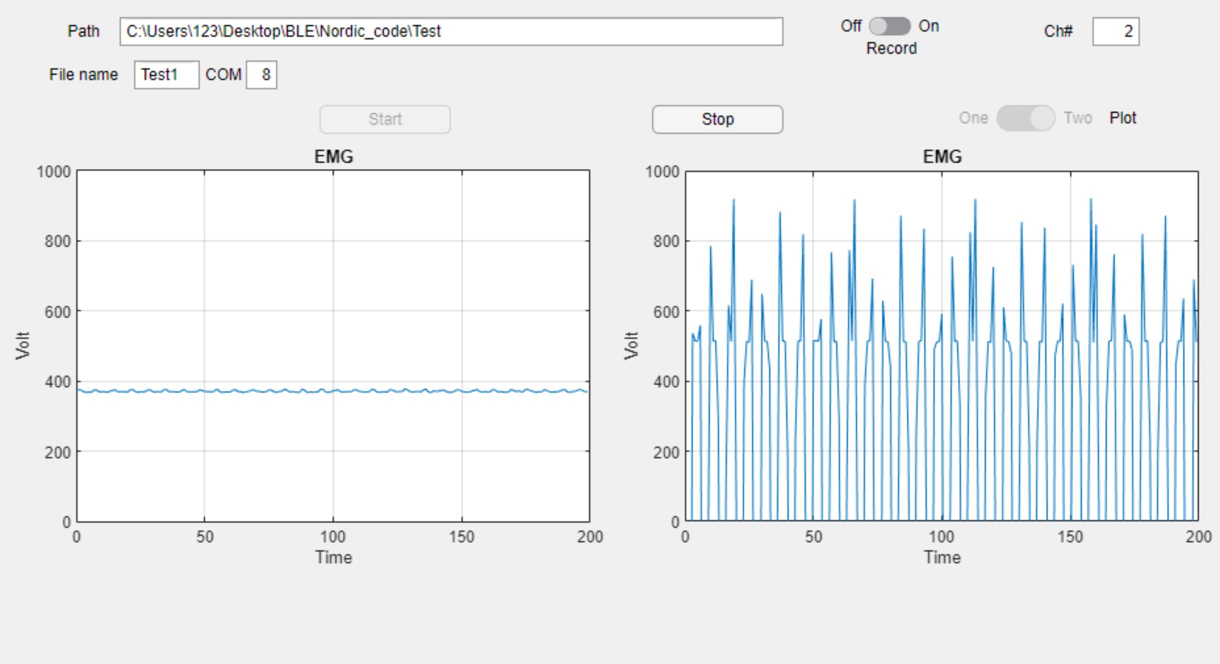


Path C:\Users\123\Desktop\BLE\Nordic_code\Test

File name Test1 COM 8

Ch# 2

**Figure 7.2.3** graphical display of a 10 Hz 20 mVpp signal sent through one channel 1 electrode using the MATLAB application developed by the lab's graduate students

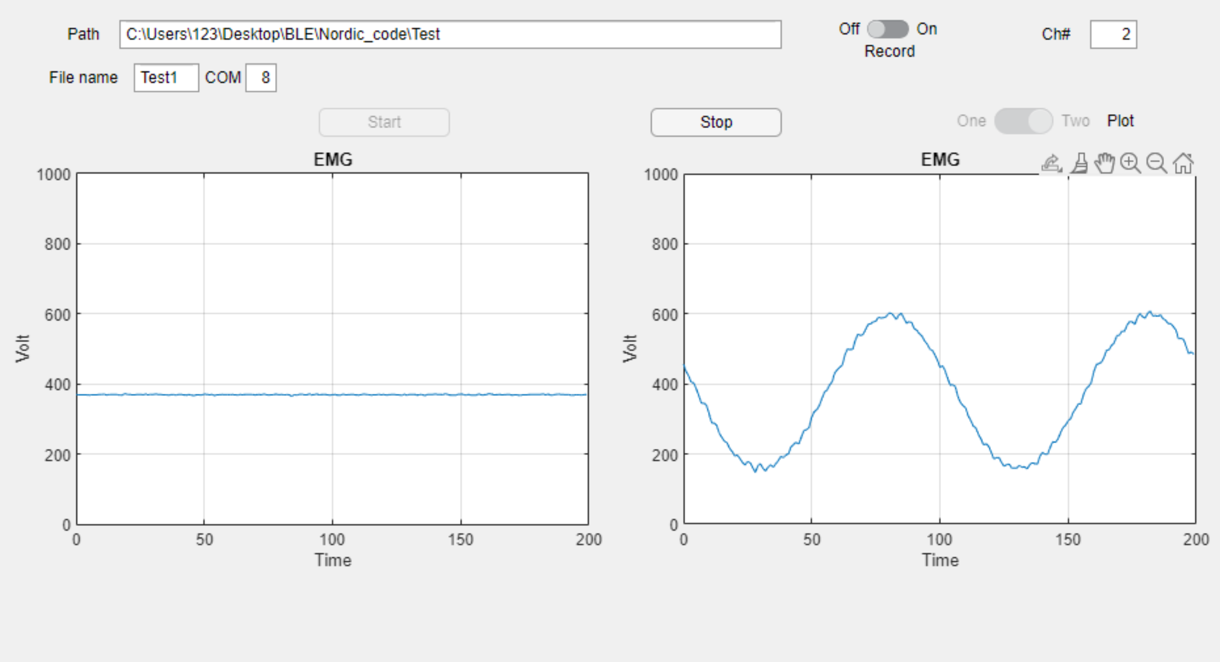


Path C:\Users\123\Desktop\BLE\Nordic_code\Test

File name Test1 COM 8

Ch# 2

**Figure 7.2.4** graphical display of a 150 Hz 20 mVpp signal sent through one channel 2 electrode using the MATLAB application developed by the lab's graduate students



Path C:\Users\123\Desktop\BLE\Nordic_code\Test

File name Test1 COM 8

Ch# 2

**Figure 7.2.5** graphical display of a 10 Hz 20 mVpp signal sent through one channel 2 electrode using the MATLAB application developed by the lab's graduate students

The same results are true for the second test as explained in the first test meaning that the signals input into channel 1 and 2 of the analog front end are being properly collected, transmitted, and received.

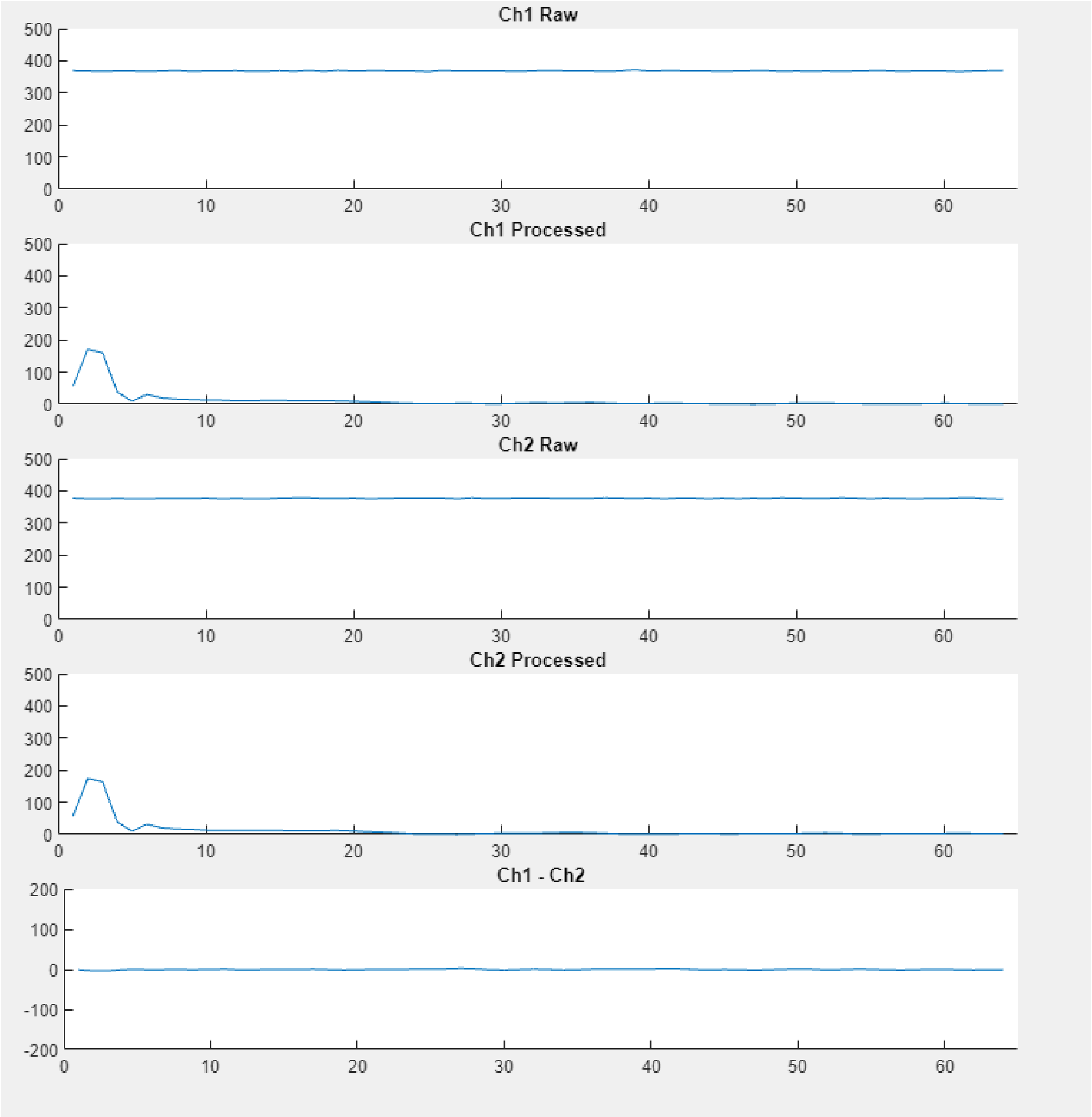


**Figure 7.2.6** graphical display of all channel 1 and channel 2 electrodes grounded using the MATLAB application developed by one of the team members

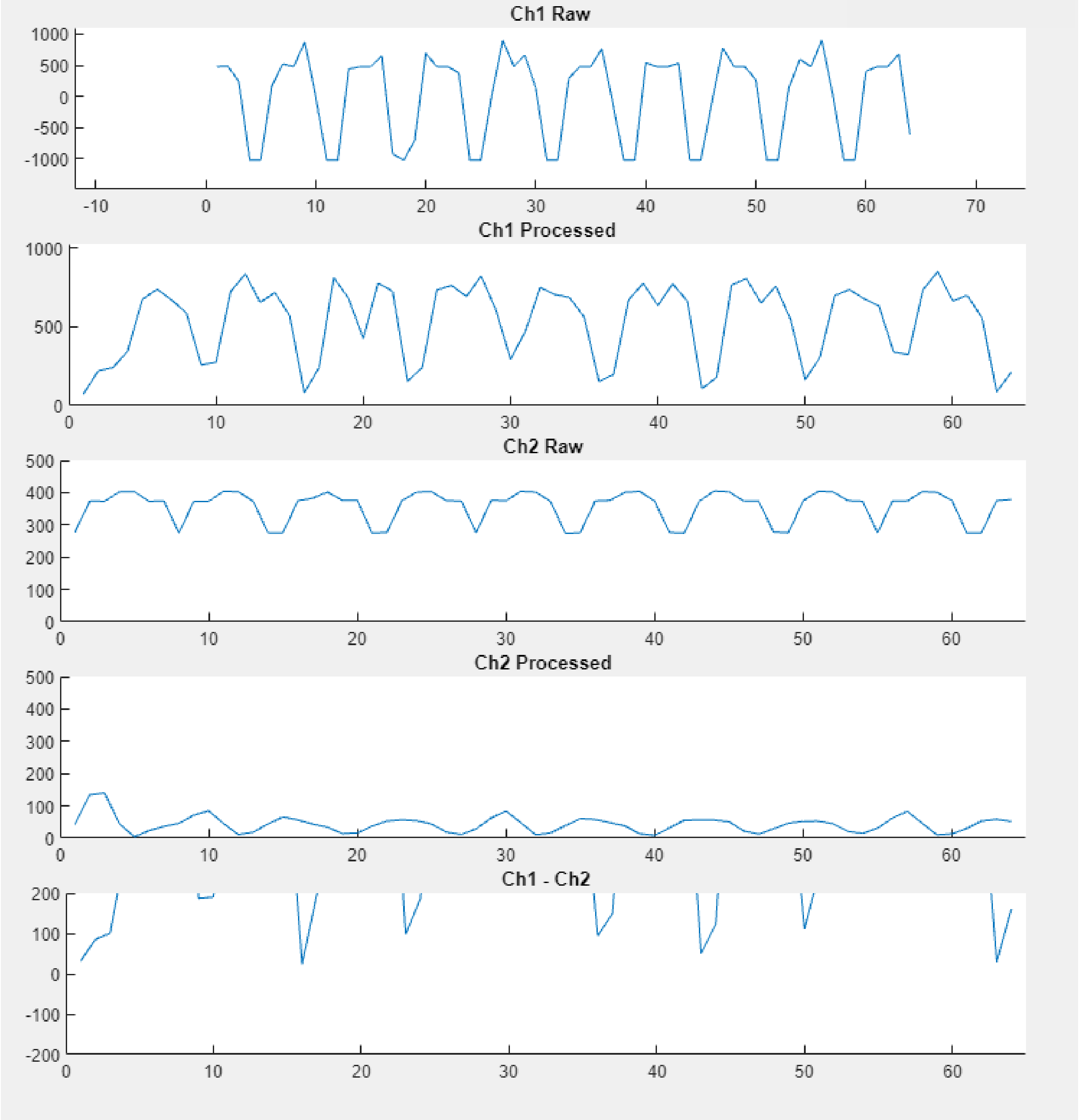


**Figure 7.2.7** graphical display of a 150 Hz 20 mVpp signal sent through one channel 1 electrode using the MATLAB application developed by one of the team members

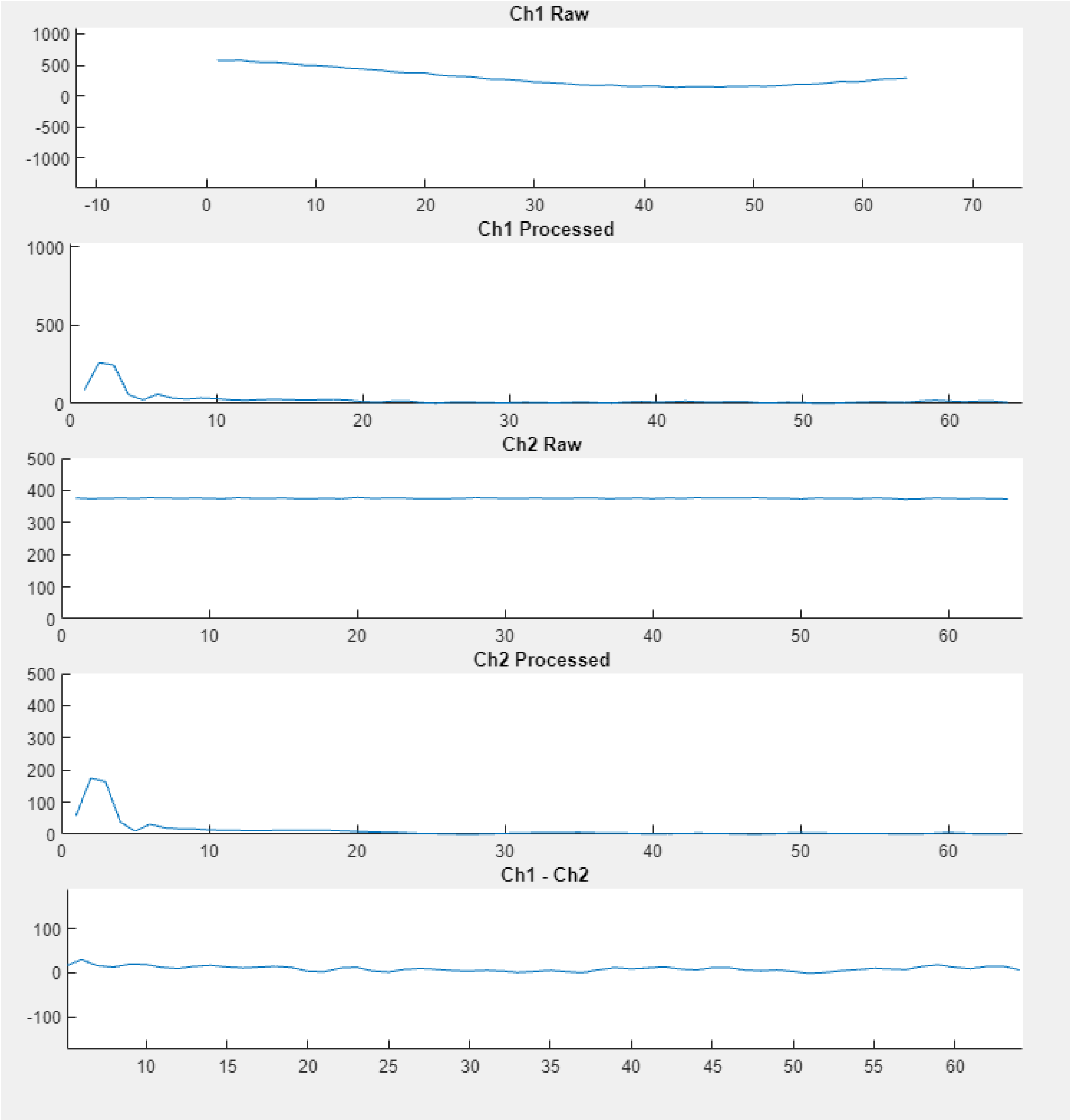


**Figure 7.2.8** graphical display of a 10 Hz 20 mVpp signal sent through one channel 1 electrode using the MATLAB application developed by one of the team members

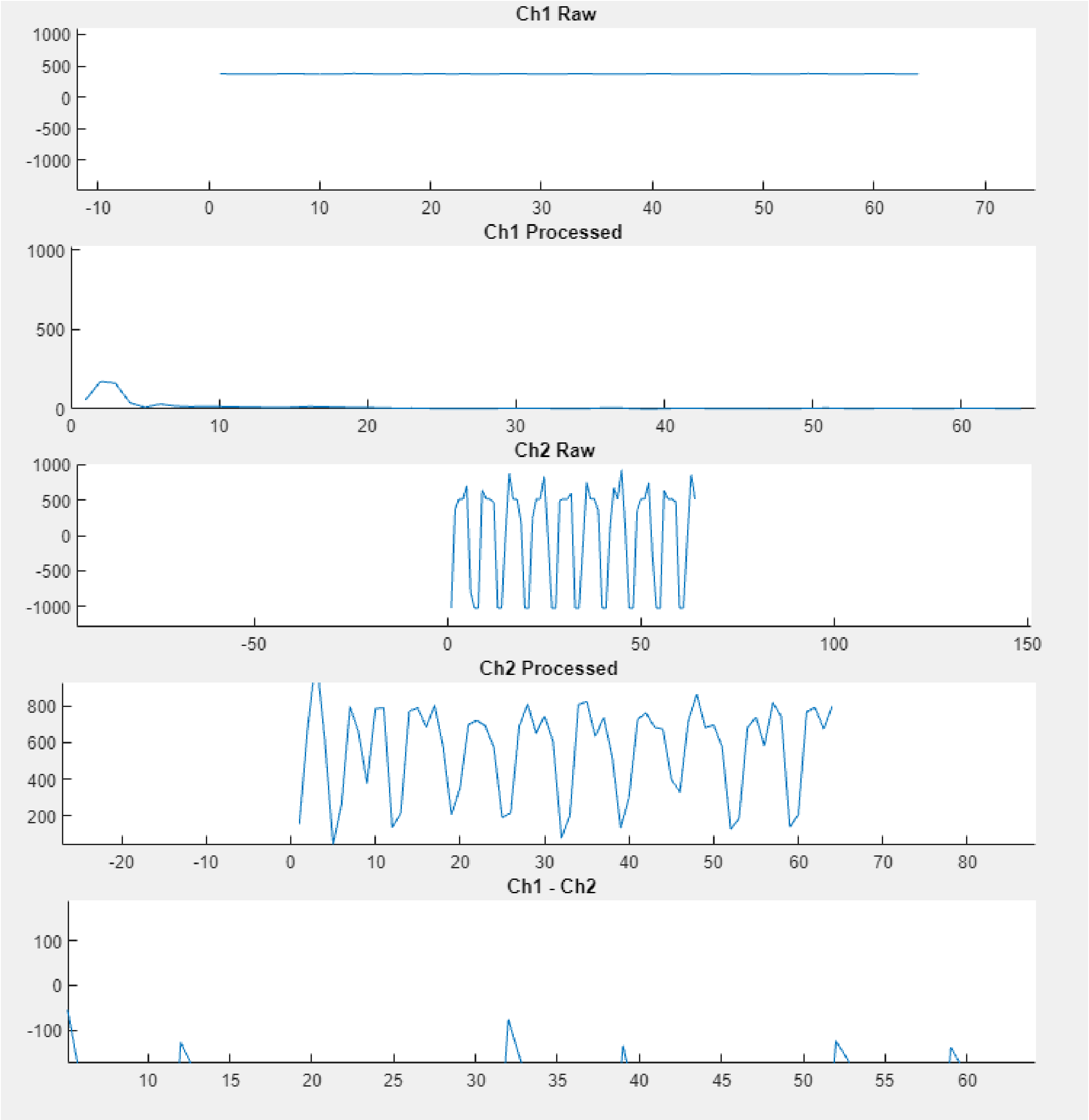


**Figure 7.2.9** graphical display of a 150 Hz 20 mVpp signal sent through one channel 2 electrode using the MATLAB application developed by one of the team members

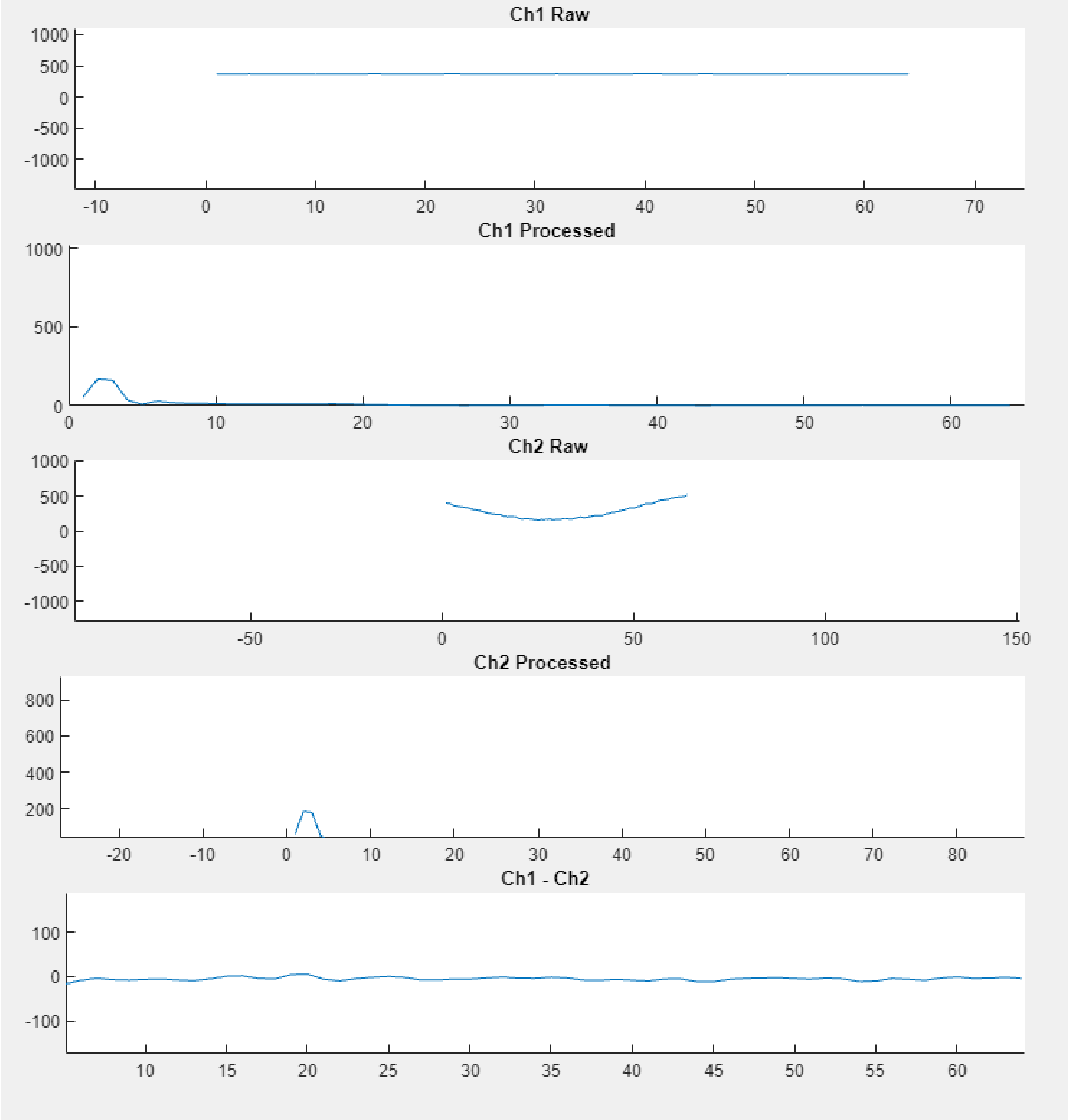


**Figure 7.2.10** graphical display of a 10 Hz 20 mVpp signal sent through one channel 2 electrode using the MATLAB application developed by one of the team members

# 8. System Results & Observations

The overall system functioned as desired when testing was completed. The analog front end correctly captured the desired flexion and extension EMG signals from the forearm. The microcontroller took these signals and digitized them and then sent them off to the Pong application. The Pong application then used these digitized values to control the paddle movement.

The player's paddle moved left and right based on whether or not the player was flexing or extending their wrist. The responsiveness of the paddle was sufficient to play the game without much delay. The paddle moved quickly in response to flexion or extension of the wrist. We were able to confirm that an entire game of Pong could be played using our system. Below are images of the working Pong application. Attached to this report is a video of the successful Pong application being played using our system, titled "PongEMGRecording." In the video, the user's EMG is controlling the bottom paddle.

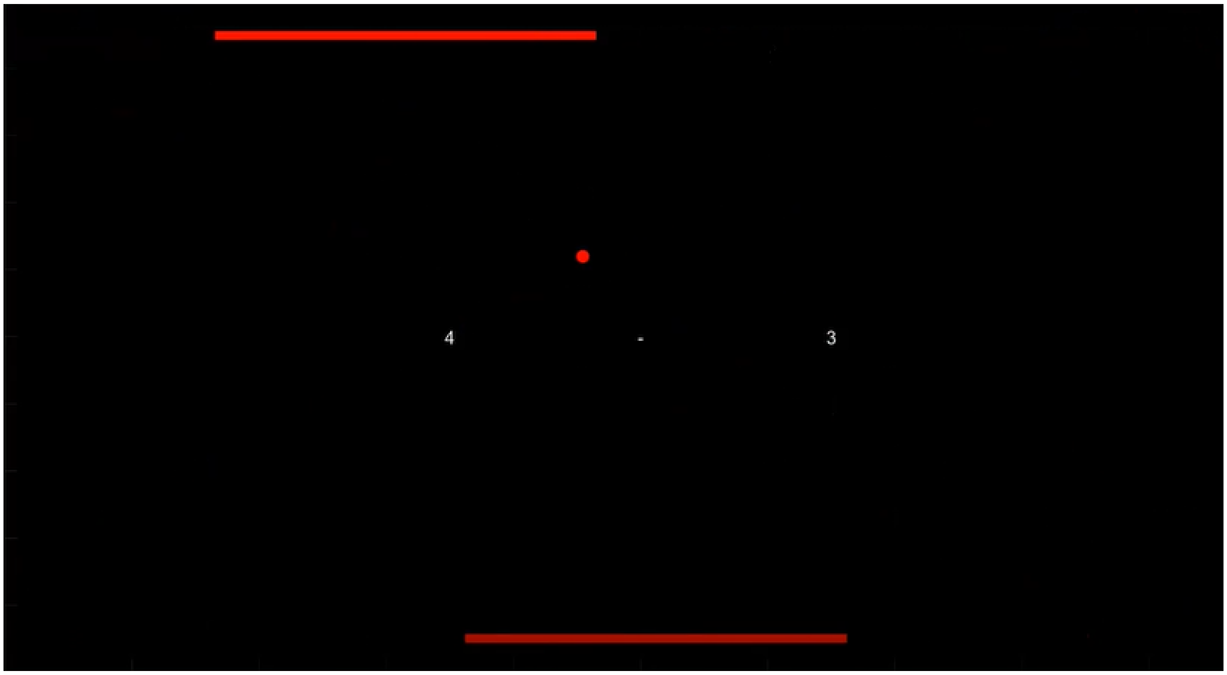


**Figure 8.1** Pong Game Being Played By User Via Bottom Paddle

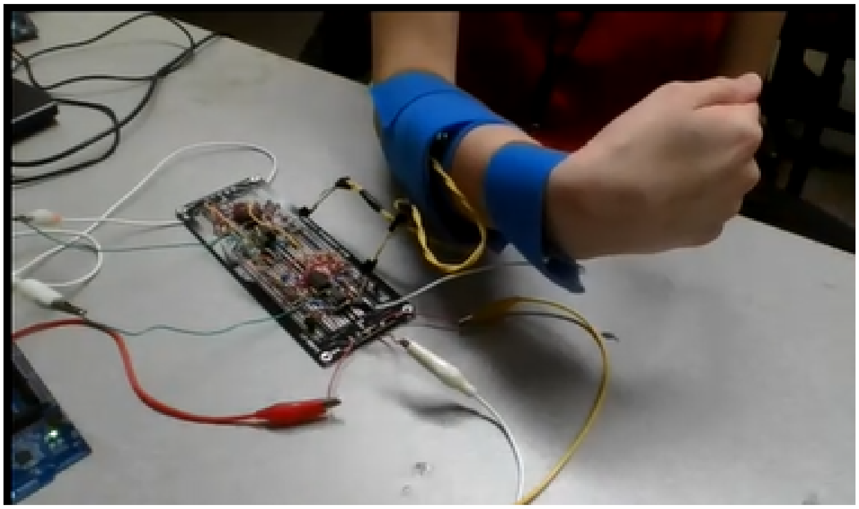


**Figure 8.2** Flexion of Wrist During Pong Game

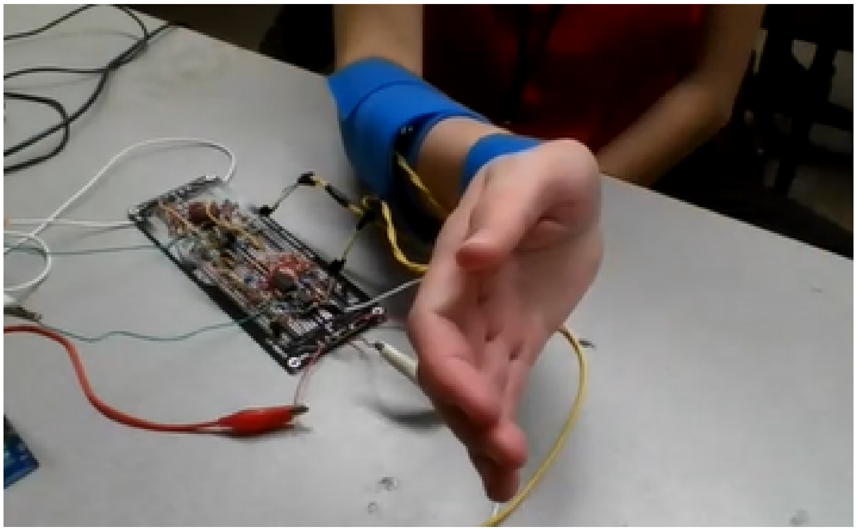


**Figure 8.3** Extension of Wrist During Pong Game

# 9. System Discussion

The system functioned as expected based on our design goal. We were able to create a system that could detect EMG signals from the body and use them to play a game that could aid in stroke rehabilitation. However, some improvements to the system can be made. The electrode armband can sometimes feel itchy and uncomfortable for the user. Changing out the Velcro material for a softer, smoother material like an Ace bandage could be implemented. The solder board could be switched out for a PCB and integrated into the armband for a smaller, more compact design. This would allow more freedom for the user to move their arm, rather than needing to rest their arm on the table directly next to the front end. The responsiveness of the paddle in the Pong game could also be improved by continuing to fine-tune the voltage range of the input signals, improving the EMG filtering, and adjusting the paddle speed in the MATLAB code.

## *9.1 Use of Standards*

Our team followed various standards and protocols throughout our design process. The Bluetooth Low Energy (LE) standard was used to communicate between the peripheral and central boards. The USB standard was used to power both the peripheral and central boards and also provided a line of communication between the central board and the laptop running the MATLAB application. A USB Micro B output on the central board was connected to a USB-A input on the laptop via a data cable to provide this line of data communication between the board and the laptop. The board was powered by an additional USB Micro B and USB-A connection, however, this cable was a power cable rather than a data cable. The IEEE numeric standards of integers and floating point values were used throughout the Segger code for the central and peripheral board communication and MATLAB script to create the Pong game.

## *9.2 Ethics and Safety*

Our team ensured to test our system with both ethics and safety in mind. Our system was only tested on members of the MQP team so no data was gathered from any test subjects other than ourselves. Also when testing the system on humans, we ensured that battery power was used to prevent possible electric shock from a DC power supply.

## 10. Conclusion & Future Work

Our research proved that the system provides an efficient EMG signal acquisition and analysis. The housing of the design was chosen to be on an armband, but with that said, the accuracy of the system could be improved by choosing a different design. More stable and comfortable housing could be adopted in future versions.

The circuitry of the system was first designed as a breadboard. To avoid short circuits and possible wiring issues during testing and application, a solderboard was then created. While the solderboard helps with most of the possible problems, it would be helpful to choose the PCB option in the future. This would also help with decreasing the size of the front end.

The microcontroller that is currently being used is a part of a development kit. It is significantly big in size, and this could be reduced by using a different microcontroller kit. The same microcontroller unit could be used, but a dongle from Nordic Semiconductors, or, as previously mentioned in the design options section of the Microcontroller, the Adafruit Feather can also be helpful.

The app is currently in the form of a demo, rather than a complete app, even though it has the full functionality that was aimed for. A more aesthetic design and a more user-friendly control would be needed for commercial use. Also, a machine learning algorithm that learns the muscle movements and value ranges from the sample datasets could be implemented. The application can be moved to the web, and it could be used on any device, not only computers. The most needed improvement would be to decrease the number of loops used in the processing algorithm to increase the speed and make the program compatible with even the slowest devices.

Ultimately, the system is currently in its Phase I design, and it is fully-functional in terms of EMG processing and implementation of the game. For any future work, it could be upgraded to Phase II, and tools that are helpful to medical professionals, such as a multi-user interface, easier placement of electrodes, etc., are possible areas of improvement.

# 11. References

*245 Keystone Electronics: Mouser. Mouser Electronics. (n.d.). https://www.mouser.com/ProductDetail/Keystone-Electronics/245?qs=Cb2nCFKsA8qYA1CMmEO%252BaA%3D%3D*

Adafruit, (n.d.). Adafruit Feather NRF52840 Express. adafruit industries blog RSS. https://www.adafruit.com/product/4062

*Advantages of MATLAB | Disadvantages of MATLAB*. (n.d.). https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-MATLAB.html

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/comst.2015.2444095

Alkhafaf, O. S., Wali, M. K., & Al-Timemy, A. H. (2020). Improved Prosthetic Hand Control with Synchronous Use of Voice Recognition and Inertial Measurements. IOP Conference Series: Materials Science and Engineering, 745(1), 012088. https://doi.org/10.1088/1757-899x/745/1/012088

Amazon Basics 4 Pack CR2032 3 Volt Lithium Coin Cell Battery. Amazon b. (n.d.). Retrieved October 14, 2022, from https://www.amazon.com/AmazonBasics-CR2032-Lithium-Coin-Cell/dp/B0787K2XWZ

Analog Devices. (2016) Precision Instrumentation Amplifier. https://www.analog.com/media/en/technical-documentation/data-sheets/AD8221-EP.pdf

"Analog Devices. (n.d.) Dual/Quad Over-The-Top Micropower Rail-to-Rail Input and Output Op Amps https://www.mouser.com/datasheet/2/609/14901afd-2954046.pdf
Micropower Rail-to-Rail
Input and Output Op Amps
"
Andrade, K. O., Ito, G. G., Joaquim, R. C., Jardim, B., Siqueira, A. A. G., Caurin, G. A. P., & Becker, M. (2010). A Robotic System for Rehabilitation of Distal Radius Fracture Using Games. 2010 Brazilian Symposium on Games and Digital Entertainment. https://doi.org/10.1109/sbgames.2010.26

ASET - The Neurodiagnostic Society (1997, August) Invasive Electrode Techniques Retrieved from https://www.aset.org/files/public/Invasive_Electrode_Techniques.pdf

AVR121: Enhancing ADC resolution by oversampling [Review of AVR121: Enhancing ADC resolution by oversampling]. (n.d.). Atmel. Retrieved October 13, 2022, from https://ww1.microchip.com/downloads/en/Appnotes/doc8003.pdf

Bai, Y. W., Cheng, C. Y., Lu, C. L., & Huang, Y. S. (2005, May). Design and implementation of an embedded remote ECG measurement system. In 2005 IEEE Instrumentation and Measurement Technology Conference Proceedings (Vol. 2, pp. 1401-1406). IEEE.

Basic Total Harmonic Distortion (THD) Measurement [Review of Basic Total Harmonic Distortion (THD) Measurement]. Microsemi. (2015). https://www.microsemi.com/document-portal/doc_view/134813-an30-basic-total-harmonic-distortion-thd-measurement

Behmanesh, A., Sayfouri, N., & Sadoughi, F. (2020). Technological Features of Internet of Things in Medicine: A Systematic Mapping Study. Wireless Communications and Mobile Computing, 2020, 1–27. https://doi.org/10.1155/2020/9238614

Campanini, I., Disselhorst-Klug, C., Rymer, W. Z., & Merletti, R. (2020). Surface EMG in Clinical Assessment and Neurorehabilitation: Barriers Limiting Its Use. Frontiers in Neurology, 11. https://doi.org/10.3389/fneur.2020.00934

Cass, S. (2018, July 2). Chip Hall of Fame: Intel 4004 Microprocessor. IEEE Spectrum. https://spectrum.ieee.org/chip-hall-of-fame-intel-4004-microprocessor

CDC. (2019). Stroke Treatment. Centers for Disease Control and Prevention. https://www.cdc.gov/stroke/treatments.htm

Centers for Disease Control and Prevention. (2021, May 25). Stroke Facts. CDC. https://www.cdc.gov/stroke/facts.htm

Chen, X., Gong, L., Wei, L., Yeh, S.-C., Da Xu, L., Zheng, L., & Zou, Z. (2021). A Wearable Hand Rehabilitation System With Soft Gloves. IEEE Transactions on Industrial Informatics, 17(2), 943–952. https://doi.org/10.1109/tii.2020.3010369

Cheour, R., Khriji, S., abid, M., & Kanoun, O. (2020). Microcontrollers for IoT: Optimizations, Computing Paradigms, and Future Directions. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). https://doi.org/10.1109/wf-iot48130.2020.9221219

Core Specification 4.0 – Bluetooth® Technology Website. (n.d.). Www.bluetooth.com. https://www.bluetooth.com/specifications/specs/core-specification-4-0/

Core Specification 5.3 – Bluetooth® Technology Website. (n.d.). Www.bluetooth.com. Retrieved

October 13, 2022, from https://www.bluetooth.com/specifications/specs/core-specification-5-3/

*Difference Between C and Python*. (2022, June 30). InterviewBit. https://www.interviewbit.com/blog/difference-between-c-and-python/

Dogan Ibrahim. (2014). PIC microcontroller projects in C : basic to advanced. Elsevier/Newnes.

Energizer 9V batteries, Max Premium 9 volt battery alkaline, 4 count. Amazon.com. (n.d.). https://www.amazon.com/Energizer-9-Volt-Alkaline-Batteries-4-Count/dp/B00IA8M42G

Ensworth, J. F., & Reynolds, M. S. (2017). BLE-Backscatter: Ultralow-Power IoT Nodes Compatible With Bluetooth 4.0 Low Energy (BLE) Smartphones and Tablets. IEEE Transactions on Microwave Theory and Techniques, 65(9), 3360–3368. https://doi.org/10.1109/tmtt.2017.2687866

Establish the fundamental concept. (n.d.). OpenClassrooms. https://openclassrooms.com/en/courses/4542221-deepen-your-knowledge-by-learning-object-oriented-programming-oop-with-swift/4661391-establish-the-fundamental-concept

Everbilt m4 zinc-plated flat washers (4-pack) . The Home Depot. (n.d.). https://www.homedepot.com/p/M4-Zinc-Plated-Flat-Washers-4-Pack-803528/204276428

Everbilt M4-0.7 x 8 mm Phillips flat head zinc plated machine screw (3-pack) . The Home Depot. (n.d.). https://www.homedepot.com/p/Everbilt-M4-0-7-x-8-mm-Phillips-Flat-Head-Zinc-Plated-Machine-Screw-3-Pack-802998/204274525

Feigin, V. L., Brainin, M., Norrving, B., Martins, S., Sacco, R. L., Hacke, W., Fisher, M., Pandian, J., & Lindsay, P. (2022). World Stroke Organization (WSO): Global Stroke Fact Sheet 2022. International Journal of Stroke, 17(1), 18–29. https://doi.org/10.1177/17474930211065917

G. Falciasecca and B. Valotti, (2009) "Guglielmo Marconi: The pioneer of wireless communications," 2009 European Microwave Conference (EuMC), pp. 544-546, doi: 10.23919/EUMC.2009.5296358.

Glossary Definition for ENOB. (n.d.). Www.maximintegrated.com. Retrieved October 13, 2022, from https://www.maximintegrated.com/en/glossary/definitions.mvp/term/ENOB/gpk/109

Gopura, R. A. R. C., Kiguchi, K., & Li, Y. (2009). SUEFUL-7: A 7DOF upper-limb exoskeleton robot with muscle-model-oriented EMG-based control. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. https://doi.org/10.1109/iros.2009.5353935

Hasan, K., Biswas, K., Ahmed, K., Nafi, N. S., & Islam, M. S. (2019). A comprehensive review of wireless body area network. Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2019.06.016

Hau, C. T., Gouwanda, D., Gopalai, A. A., Low, C. Y., & Hanapiah, F. A. (2021). Gamification and Control of Nitinol Based Ankle Rehabilitation Robot. Biomimetics, 6(3), 53. https://doi.org/10.3390/biomimetics6030053

Hee, C. L., Chong, T. H., Gouwanda, D., Gopalai, A. A., Low, C. Y., & binti Hanapiah, F. A. (2017). Developing interactive and simple electromyogram PONG game for foot dorsiflexion and plantarflexion rehabilitation exercise. 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). https://doi.org/10.1109/embc.2017.8036815

Honkanen, M., Lappetelainen, A., & Kivekas, K. (n.d.). Low end extension for Bluetooth. Proceedings. 2004 IEEE Radio and Wireless Conference (IEEE Cat. No.04TH8746). https://doi.org/10.1109/rawcon.2004.1389107

Hosseiniravandi, M., Kahlaee, A. H., Karim, H., Ghamkhar, L., & Safdari, R. (2020). Home-based telerehabilitation software systems for remote supervising: a systematic review. International Journal of Technology Assessment in Health Care, 36(2), 113–125. https://doi.org/10.1017/s0266462320000021

Hussain, N., Alt Murphy, M., & Sunnerhagen, K. S. (2018). Upper Limb Kinematics in Stroke and Healthy Controls Using Target-to-Target Task in Virtual Reality. Frontiers in Neurology, 9. https://doi.org/10.3389/fneur.2018.00300

Improving ADC Results Through Oversampling and Post-Processing of Data (2007). [Review of Improving ADC Results Through Oversampling and Post-Processing of Data]. Actel Corporation. https://www.microsemi.com/document-portal/doc_view/131569-improving-adc-results-white-paper

Internet of Things Market Size, Regional Analysis and Growth Rate 2026. (2019). Fortunebusinessinsights.com. https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307

Jeon, K. E., She, J., Soonsawad, P., & Ng, P. C. (2018). BLE Beacons for Internet of Things Applications: Survey, Challenges, and Opportunities. IEEE Internet of Things Journal, 5(2), 811–828. https://doi.org/10.1109/jiot.2017.2788449

Kester, Walt. (2009). Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor.

Khan, R. A., & Pathan, A.-S. K. (2018). The state-of-the-art wireless body area sensor networks: A survey. International Journal of Distributed Sensor Networks, 14(4), 155014771876899. https://doi.org/10.1177/1550147718768994

Kieser, R., Reynisson, P., & Mulligan, T. J. (2005). Definition of signal-to-noise ratio and its critical role in split-beam measurements. ICES Journal of Marine Science, 62(1), 123–130. https://doi.org/10.1016/j.icesjms.2004.09.006

*Lecture 6 Advanced MATLAB: Object-Oriented Programming*. (n.d.). [Slide show] https://math.lbl.gov/~mjzahr/content/cme292/spr15/lec/lec06.pdf

Leong, S. C., Tang, Y. M., Toh, F. M., & Fong, K. N. K. (2022). Examining the effectiveness of virtual, augmented, and mixed reality (VAMR) therapy for upper limb recovery and activities of daily living in stroke patients: a systematic review and meta-analysis. Journal of NeuroEngineering and Rehabilitation, 19(1). https://doi.org/10.1186/s12984-022-01071-x

Luo, R., Sun, S., Zhang, X., Tang, Z., & Wang, W. (2020). A Low-Cost End-to-End sEMG-Based Gait Sub-Phase Recognition System. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28(1), 267–276. https://doi.org/10.1109/tnsre.2019.2950096

Maceira-Elvira, P., Popa, T., Schmid, A.-C., & Hummel, F. C. (2019). Wearable technology in stroke rehabilitation: towards improved diagnosis and treatment of upper-limb motor impairment. Journal of NeuroEngineering and Rehabilitation, 16(1). https://doi.org/10.1186/s12984-019-0612-y

Maulden, S. A., Gassaway, J., Horn, S. D., Smout, R. J., & DeJong, G. (2005). Timing of Initiation of Rehabilitation After Stroke. Archives of Physical Medicine and Rehabilitation, 86(12), 34–40. https://doi.org/10.1016/j.apmr.2005.08.119

Maxim Integrated. (n.d.) Low-Noise, Dual-Output, Regulated Charge Pump https://datasheets.maximintegrated.com/en/ds/MAX768.pdf

McDermott-Wells, P. (2005). What is Bluetooth? IEEE Potentials, 23(5), 33–35. https://doi.org/10.1109/mp.2005.1368913

Medical Expo. (n.d.). Natus® - EMG electrode by Natus Medical: Medicalexpo. Medical Expo. Retrieved October 13, 2022, from https://www.medicalexpo.com/prod/natus-medical/product-76900-927303.html

Mendez, I., Hansen, B. W., Grabow, C. M., Smedegaard, E. J. L., Skogberg, N. B., Uth, X. J., Bruhn, A., Geng, B., & Kamavuako, E. N. (2017). Evaluation of the Myo armband for the classification of hand motions. 2017 International Conference on Rehabilitation Robotics (ICORR). https://doi.org/10.1109/icorr.2017.8009414

Mishra, S., Singh, N. K., & Rousseau, V. (2016). Sensor Interfaces. System on Chip Interfaces for Low Power Design, 331–344. https://doi.org/10.1016/b978-0-12-801630-5.00010-4

Moulaei, K., Sheikhtaheri, A., Nezhad, M. S., Haghdoost, A., Gheysari, M., & Bahaadinbeigy, K. (2022). Telerehabilitation for upper limb disabilities: a scoping review on functions, outcomes, and evaluation methods. Archives of Public Health, 80(1). https://doi.org/10.1186/s13690-022-00952-w

Myo Gesture Control Armband. (n.d.). Uncrate.com. Retrieved October 13, 2022, from https://uncrate.com/myo-gesture-control-armband/

Nokia builds a better Bluetooth - Technology - International Herald Tribune. (2006, October 3). The New York Times. https://www.nytimes.com/2006/10/03/technology/03iht-nokia.3018027.html

Nordic Semiconductor Infocenter. (n.d.). Infocenter.nordicsemi.com. https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52832.ps.v1.1%2Fccm.html

Nordic Semiconductor Infocenter. (n.d.). Infocenter.nordicsemi.com. Retrieved October 13, 2022, from https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52832.ps.v1.1%2Faar.html

Nordic Semiconductor nRF52840-DK. Symmetry Electronics. (n.d.). https://www.symmetryelectronics.com/products/nordic-semiconductor/nrf52840-dk/?utm_source=OemSecrets&utm_medium=compref&utm_campaign=oemsecrets&utm_term=nrf52840-dk

nRF52840 [Review of nRF52840]. Nordic Semiconductor. (2021). https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.7.pdf

nRF52840 DK. Nordic Semiconductor. (2020). https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.7.pdf

nRF52840 High-end multiprotocol Bluetooth Low Energy (LE) SoC supporting: Bluetooth 5.3/Bluetooth mesh/Thread/ Zigbee/802.15.4/ANT. Nordic Semiconductor. https://nsscprodmedia.blob.core.windows.net/prod/software-and-other-downloads/product-briefs/nrf52840-soc-v3.0.pdf

Prasanna. (2022, January 7). *Advantages and Disadvantages of Python | Python Language Advantages, Disadvantages and Its Applications*. A Plus Topper. Retrieved December 15, 2022, from https://www.aplustopper.com/advantages-and-disadvantages-of-python/

Prime-line M4-0.70 class 8 metric zinc plated steel finished hex nuts (25-pack) . The Home Depot. (n.d.). https://www.homedepot.com/p/Prime-Line-M4-0-70-Class-8-Metric-Zinc-Plated-Steel-Finished-Hex-Nuts-25-Pack-9087796/310561417

Product Documentation - NI. (n.d.). Www.ni.com. Retrieved October 13, 2022, from https://www.ni.com/docs/en-US/bundle/labview-sound-and-vibration-toolkit/page/svtconcepts/lvac_thd_n.html

Qiu, Q., Cronce, A., Patel, J., Fluet, G. G., Mont, A. J., Merians, A. S., & Adamovich, S. V. (2020). Development of the Home based Virtual Rehabilitation System (HoVRS) to remotely deliver an intense and customized upper extremity training. Journal of NeuroEngineering and Rehabilitation, 17(1). https://doi.org/10.1186/s12984-020-00789-w

Raghunathan, K. R. (2021). History of Microcontrollers: First 50 Years. IEEE Micro, 41(6), 97–104. https://doi.org/10.1109/MM.2021.3114754

Ranzani, R., Lambercy, O., Metzger, J.-C., Califfi, A., Regazzi, S., Dinacci, D., Petrillo, C., Rossi, P., Conti, F. M., & Gassert, R. (2020). Neurocognitive robot-assisted rehabilitation of hand function: a randomized control trial on motor recovery in subacute stroke. Journal of NeuroEngineering and Rehabilitation, 17(1). https://doi.org/10.1186/s12984-020-00746-7

Razfar, N., Kashef, R., & Mohammadi, F. (2021). A Comprehensive Overview on IoT-Based Smart Stroke Rehabilitation Using the Advances of Wearable Technology. 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). https://doi.org/10.1109/hpcc-dss-smartcity-dependsys53884.2021.00205

Rodriguez, S. (2019, September 23). Facebook agrees to acquire brain-computing start-up CTRL-labs. CNBC; CNBC. https://www.cnbc.com/2019/09/23/facebook-announces-acquisition-of-brain-computing-start-up-ctrl-labs.html

S, D. J., P, D. D., & Shashikala, K. (2018). A Brief Review on Types and Design Methods of ADC [Review of A Brief Review on Types and Design Methods of ADC]. Dr. Jamuna S Journal of Engineering Research and Application, 8(6), Part-II. https://doi.org/https://doi.org/10.9790/9622-0806028591

Sawers, P. (2018, October 12). Amazon-backed wearables company Thalmic Labs kills its Myo armband, teases new product. VentureBeat. https://venturebeat.com/mobile/amazon-backed-wearables-company-thalmic-labs-kills-its-myo-armband-teases-new-product/

Scheme, E., & Englehart, K. (2011). Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. The Journal of Rehabilitation Research and Development, 48(6), 643. https://doi.org/10.1682/jrrd.2010.09.0177

Security Subsystems for Systems-on- Chip (SoCs) Common concepts and usage paradigms of security subsystems [Review of Security Subsystems for Systems-on- Chip (SoCs) Common concepts and usage paradigms of security subsystems]. (n.d.). NXP. Retrieved October 13, 2022, from https://www.nxp.com/docs/en/white-paper/Security-Subsystems-WP.pdf

Siemens DISW. (n.d.). Community.sw.siemens.com. Retrieved October 13, 2022, from https://community.sw.siemens.com/s/article/what-is-total-harmonic-distortion-thd

Signal-to-Noise-and-Distortion Ratio (SNDR). (2019). Intel. https://www.intel.com/content/www/us/en/docs/programmable/683852/current/signal-to-noise-and-distortion-ratio-sndr.html

Spurious-Free Dynamic Range (SFDR) Explained. (n.d.). Www.ni.com. Retrieved October 13, 2022, from https://www.ni.com/en-tr/support/documentation/supplemental/18/specifications-explained--spurious-free-dynamic-range--sfdr-.html

Steele, K. M., Papazian, C., & Feldner, H. A. (2020). Muscle Activity After Stroke: Perspectives on Deploying Surface Electromyography in Acute Care. Frontiers in Neurology, 11. https://doi.org/10.3389/fneur.2020.576757

STMicroelectronics. (n.d.)  1.2 V to 37 V adjustable voltage regulators. https://www.mouser.com/datasheet/2/389/lm217-1849593.pdf

Storr, W. (2022, August 6). *Butterworth filter design and low Pass Butterworth filters*. Basic Electronics Tutorials. Retrieved December 14, 2022, from https://www.electronics-tutorials.ws/filter/filter_8.html

Storr, W. (2022, August 6). Second order filter: Second order low pass filter design. Basic Electronics Tutorials. https://www.electronics-tutorials.ws/filter/second-order-filters.html

Texas Instruments. (2002) LM148, LM248, LM348 Quadruple Operational Amplifier https://www.ti.com/lit/ds/symlink/lm148.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1665705922922&ref_url=https%253A%252F%252Fwww.mouser.com%252F

"Texas Instruments. (n.d.) FET-Input, Low Power https://www.ti.com/lit/ds/symlink/ina121.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1665496106142&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%25

2Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fina121
INSTRUMENTATION AMPLIFIER"

Texas Instruments. (n.d.) INA12x Precision, Low-Power Instrumentation Amplifiers. https://www.ti.com/lit/ds/symlink/ina129.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1665455131113&ref_url=https%253A%252F%252Fwww.mouser.mx%252F

Texas Instruments. (n.d.) LM27762 Low-Noise Positive and Negative Output Integrated https://www.ti.com/lit/ds/symlink/lm27762.pdf?ts=1664900652103&ref_url=https%253A%252F%252Fwww.google.com%252F
Charge Pump Plus LDO

The Bluetooth® Low Energy Primer. (2022, May 19). Bluetooth® Technology Website. https://www.bluetooth.com/bluetooth-resources/the-bluetooth-low-energy-primer/?utm_source=internal&utm_medium=blog&utm_campaign=technical&utm_content=the-bluetooth-low-energy-primer

Tosi, J., Taffoni, F., Santacatterina, M., Sannino, R., & Formica, D. (2017). Performance Evaluation of Bluetooth Low Energy: A Systematic Review. Sensors, 17(12), 2898. https://doi.org/10.3390/s17122898

Tsekleves, E., Paraskevopoulos, I. T., Warland, A., & Kilbride, C. (2014). Development and preliminary evaluation of a novel low cost VR-based upper limb stroke rehabilitation platform using Wii technology. Disability and Rehabilitation: Assistive Technology, 11(5), 413–422. https://doi.org/10.3109/17483107.2014.981874

Tunc, M.A., Gures, E., & Shayea, I. (2021). A Survey on IoT Smart Healthcare: Emerging Technologies, Applications, Challenges, and Future Trends. ArXiv, abs/2109.02042.

Ultium EMG. (n.d.). Noraxon USA. Retrieved October 13, 2022, from https://www.noraxon.com/our-products/ultium-emg/

Velcro Armbands. PatchPanel. (n.d.). https://patchpanel.ca/products/velcro-armbands?variant=39153826053

*Why Use Object-Oriented Design - MATLAB & Simulink*. (n.d.). https://www.mathworks.com/help/matlab/matlab_oop/why-use-object-oriented-design.html

Wireless EMG. (n.d.). ADInstruments. Retrieved October 13, 2022, from https://www.adinstruments.com/research/human/exercise-physiology/wireless-emg-electromyography

WPI 3/8" printed lanyard with Hook. Worcester Polytechnic Institute Apparel & Spirit Store. (n.d.). https://wpi.spirit.bncollege.com/worcester-polytechnic-institute-engineers-wpi-3/8-printed-lanyard-with-hook/t-7811554713+p-4533885707536+z-8-3370889759?_ref=p-DLP%3Am-GRID%3Ai-r0c0%3Apo-0

Wu, F., Wu, T., & Yuce, M. (2018). An Internet-of-Things (IoT) Network System for Connected Safety and Health Monitoring Applications. Sensors, 19(1), 21. https://doi.org/10.3390/s19010021

Wu, P., Zeng, F., Li, Y., Yu, B., Qiu, L., Qin, W., Li, J., Zhou, Y., & Liang, F. (2015). Changes of resting cerebral activities in subacute ischemic stroke patients. Neural Regeneration Research, 10(5), 760–765. https://doi.org/10.4103/1673-5374.156977

Wu, T., Wu, F., Redoute, J.-M., & Yuce, M. R. (2017). An Autonomous Wireless Body Area Network Implementation Towards IoT Connected Healthcare Applications. IEEE Access, 5, 11413–11422. https://doi.org/10.1109/access.2017.2716344

Xi, X., Tang, M., Miran, S. M., & Luo, Z. (2017). Evaluation of Feature Extraction and Recognition for Activity Monitoring and Fall Detection Based on Wearable sEMG Sensors. Sensors, 17(6), 1229. https://doi.org/10.3390/s17061229

Xin Feng, & Winters, J. M. (n.d.). UniTherapy: A Computer-Assisted Motivating Neurorehabilitation Platform for Teleassessment and Remote Therapy. 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.https://doi.org/10.1109/icorr.2005.1501117

Yang, G., Deng, J., Pang, G., Zhang, H., Li, J., Deng, B., Pang, Z., Xu, J., Jiang, M., Liljeberg, P., Xie, H., & Yang, H. (2018). An IoT-Enabled Stroke Rehabilitation System Based on Smart Wearable Armband and Machine Learning. IEEE Journal of Translational Engineering in Health and Medicine, 6, 1–10. https://doi.org/10.1109/JTEHM.2018.2822681

Yang, G., Jiang, M., Ouyang, W., Ji, G., Xie, H., Rahmani, A. M., Liljeberg, P., & Tenhunen, H. (2018). IoT-Based Remote Pain Monitoring System: From Device to Cloud Platform. IEEE Journal of Biomedical and Health Informatics, 22(6), 1711–1719. https://doi.org/10.1109/jbhi.2017.2776351

Zahr. (n.d.). *Lecture 6 Advanced MATLAB: Object-Oriented Programming* [Slide show]. https://math.lbl.gov/~mjzahr/content/cme292/spr15/lec/lec06.pdf

Zimmerman, T. G. (1996). Personal Area Networks: Near-field intrabody communication. IBM Systems Journal, 35(3.4), 609–617. https://doi.org/10.1147/sj.353.0609

# 12. Appendix

### *Appendix A: Instrumentation Amplifier MATLAB Code [E. A. Clancy]*

```
function iamp_stage_v01
% Components.
Rg = 2600;
R1 = 24.7E3;
Cg = 5E-6;
% Frequency values at which to evaluate.
f = 0:0.01:200;
w = 2*pi*f;
fdata = [5,10,20,50,100,140,180,200];
mag_data =
[13.72340426,27.51937984,44.8,61.47540984,61.40684411,62.40310078,61.64122137,62.890
625];
phase_data = [64.8,64.8,51.84,21.6,21.6,0,0,0];
% Frequency response.
H = 1 + ( 2*R1 ./ (Rg + (1./(1j*w*Cg))) );
figure(1)
% Plot magnitude only.
%plot(fdata,mag_data,':b.')
title('Ideal and recorded magnitude response')
hold on
plot(f, abs(H),'k')
% Passband gain line.
Hpass = 1 + ( 2*R1 ./ Rg );
hold on, plot([f(1) f(end)], [Hpass Hpass], 'k--'), hold off
box('off')
xlabel('Frequency (Hz)')
ylabel('Magnitude Response')
title('Wireless EMG I-Amp Circuit')
legend('Ideal')
%legend('Recorded data','Ideal')
Return
```

## *Appendix B: High-Pass Filter MATLAB Code  [E. A. Clancy]*

```
function H = butter_hi_design(f, Fc, A, Stage1, Stage2, Stage3, Stage4, Stage5)
%
% H = b_hi_des(f, Fc, A, Stage1[, Stage2[, Stage3[, Stage4[, Stage5]]]])
%
%   Helps design an electronic circuit to build even-order, high-pass
% Butterworth filters of orders 2, 4, 6, 8, 10.  Plots the magnitude of
% the resulting frequency response.
%   See [Thomas Kugelstadt, "Chapter 16: Active
% Filter Design Techniques," Literature Number SLOA088, Texas Instruments
% Incorporated, Post Office Box 655303, Dallas, Texas  75265, 2001.
% Excerpted from "Op Amps for Everyone," Literature Number SLOD006A,
% Texas Instruments.  Available on the Internet at: http://www.ti.com].
%
% f:      Frequency axis (Hertz) for all calculations and plotting (vector).
% A:      Overall circuit gain (>=1).  Applied in 1st stage.
% Fc:     Desired cutoff frequency in Hertz (scalor).
% StageX: Up to 5 StageX arguments (one per stage) are permitted.  For each stage,
%         StageX is a vector of 1-4 elements, corresponding to C1, C2,
%         R1 and R2, respectively.  C1 must be supplied (in Farads) and
%         is the first vector element.  If a second argument is supplied,
%         then it is C2 (in Farads).  If C2 is not supplied, it is set
%         equal to C1.  If
%         a third argument is supplied, it is R1 (Ohms).  If not supplied,
%         it is set as required to form a Butterworth filter.  If a fourth
%         argument is supplied, it is R2 (Ohms).  If not supplied, it is
%         set as required to form a Butterworth filter.
%
% H: Resulting (complex-valued) frequency response corresponding to the
%    frequency axis f.
%
% USAGE RECOMENDATIONS:  To build a filter, initially call script with only
%   C1 and C2 specified for each stage.  Use the recommended
%   R1 and R2 values to find R1's and R2's that are manufactured.  Call the
%   script a second time with all values to see the resultant nominal frequency
%   response.
% Table of ai values.  For even-order Butterworth, all bi values equal 1.
aiTable = [1.4142    NaN    NaN    NaN    NaN;  % For 1-stage filter.
          1.8478 0.7654    NaN    NaN    NaN;  % For 2-stage filter.
          1.9319 1.4142 0.5176    NaN    NaN;  % For 3-stage filter.
          1.9616 1.6629 1.1111 0.3902    NaN;  % For 4-stage filter.
          1.9754 1.7820 1.4142 0.9080 0.3129]; % For 5-stage filter.

% Determine the number of stages.
Nstage = nargin - 3;
if Nstage<1 | Nstage>5, error('Must have 1-5 stages.'); end
if A<1, error('"A" must be >= 1.'); end

% Extract/develop parameters for each stage and build the frequency response.

H = ones( 1, length(f) );  % Initialize frequency response to unity.
w = 2*pi*f;               % Convert to radian frequency.
MagBig = A;               % Cascade passband gain.

for S = 1:Nstage

  if S>1, A = 1; end  % Set gain to one after first stage.

  % Prepare to parse parameter vector.
  eval(['Param = Stage' int2str(S) ';']);  % Copy parameters to a scratch vector.
  if length(Param)<1, error(['Stage ' int2str(S) ' parameter list < 1.']); end;
  if length(Param)>4, error(['Stage ' int2str(S) ' parameter list > 4.']); end;
  ai = aiTable(Nstage,S);

  % Parse parameter vector.
  % C1.
  C1 = Param(1);
  % C2.
  if length(Param)>1,  C2 = Param(2);  else,  C2 = C1; end
  % R1.
  if length(Param)>2,  R1 = Param(3);
  elseif A==1
    R1 = ( C1+C2 ) / ( 2*pi*Fc*ai*C1*C2 );
  else
```

```
    a = ((2*pi*Fc)^2) * C1*C2*C2*(1-A);
    b = -ai * 2*pi*Fc * C1*C2;
    c = C1 + C2;
    R1 = ( -b - sqrt(b*b - 4*a*c) ) / (2*a);
  end
  % R2.
  if length(Param)>3,  R2 = Param(4);
  else,  R2 = 1 / ( ((2*pi*Fc)^2)*R1*C1*C2 ); end

  % Print component values.
  fprintf('Stage %d: C1=%e F, C2=%e F, R1=%e Ohms, R2=%e Ohms\n', S, C1, C2, R1, R2);

  % Update frequency response.
  a = ( R2*(C1+C2) + R1*C2*(1-A) ) ./ (R1*R2*C1*C2);
  b = 1 ./ (R1*R2*C1*C2);
  Hstage = A ./ (  1 + ( a./(j*w) ) + ( b./(-w.*w) )  );
  H = H .* Hstage;
end

% Now, plot the frequency response magnitude.
plot( f, abs(H) )
xlabel('Frequency in Hertz')
ylabel('Frequency Response Magnitude')

L1 = find( abs(H) >= (MagBig*sqrt(2)/2) );
hold on, plot([f(L1(1)) f(L1(1))], [0 MagBig], 'm'), hold off
Thing = sprintf('Cut-off at %0.2f Hz, Passband Gain = %0.2f', f(L1(1)), MagBig);
text( max(f)/5, max(abs(H))/3, Thing);
title('Butterworth High-Pass Filter');

figure(gcf);

return
```

## *Appendix C: Low-Pass Filter MATLAB Code [E. A. Clancy]*

```
function H = butter_lo_design(f, Fc, A, Stage1, Stage2, Stage3, Stage4, Stage5)
%
% H = b_lo_des(f, Fc, A, Stage1[, Stage2[, Stage3[, Stage4[, Stage5]]]])
%
%   Helps design an electronic circuit to build even-order, low-pass
% Butterworth filters of orders 2, 4, 6, 8, 10.  Plots the magnitude of
% the resulting frequency response.
%   See [Thomas Kugelstadt, "Chapter 16: Active
% Filter Design Techniques," Literature Number SLOA088, Texas Instruments
% Incorporated, Post Office Box 655303, Dallas, Texas  75265, 2001.
% Excerpted from "Op Amps for Everyone," Literature Number SLOD006A,
% Texas Instruments.  Available on the Internet at: http://www.ti.com].
%
% f:      Frequency axis (Hertz) for all calculations and plotting (vector).
% A:      Overall circuit gain (>=1).  Applied in 1st stage.
% Fc:     Desired cutoff frequency in Hertz (scalor).
% StageX: Up to 5 StageX arguments (one per stage) are permitted.  For each stage,
%         StageX is a vector of 1-4 elements, corresponding to C1, C2,
%         R1 and R2, respectively.  C1 must be supplied (in Farads) and
%         is the first vector element.  If a second argument is supplied,
%         then it is C2 (in Farads).  If C2 is not supplied, it is set
%         equal to C1.  If
%         a third argument is supplied, it is R1 (Ohms).  If not supplied,
%         it is set as required to form a Butterworth filter.  If a fourth
%         argument is supplied, it is R2 (Ohms).  If not supplied, it is
%         set as required to form a Butterworth filter.
%
% H: Resulting (complex-valued) frequency response corresponding to the
%    frequency axis f.
%
% USAGE RECOMENDATIONS:  To build a filter, initially call script with only
%   C1 and C2 specified for each stage.  Use the recommended
%   R1 and R2 values to find R1's and R2's that are manufactured.  Call the
%   script a second time with all values to see the resultant nominal frequency
%   response.
% Table of ai values.  For even-order Butterworth, all bi values equal 1.
aiTable = [1.4142    NaN    NaN    NaN    NaN;  % For 1-stage filter.
        1.8478 0.7654    NaN    NaN    NaN;  % For 2-stage filter.
        1.9319 1.4142 0.5176    NaN    NaN;  % For 3-stage filter.
        1.9616 1.6629 1.1111 0.3902    NaN;  % For 4-stage filter.
        1.9754 1.7820 1.4142 0.9080 0.3129]; % For 5-stage filter.

% Determine the number of stages.
Nstage = nargin - 3;
if Nstage<1 | Nstage>5, error('Must have 1-5 stages.'); end
if A<1, error('"A" must be >= 1.'); end

% Extract/develop parameters for each stage and build the frequency response.

H = ones( 1, length(f) );  % Initialize frequency response to unity.
w = 2*pi*f;                % Convert to radian frequency.
MagBig = A;                 % Cascade passband gain.

for S = 1:Nstage

  if S>1, A = 1; end  % Set gain to one after first stage.

  % Prepare to parse parameter vector.
  eval(['Param = Stage' int2str(S) ';']);  % Copy parameters to a scratch vector.
  if length(Param)<1, error(['Stage ' int2str(S) ' parameter list < 1.']); end;
  if length(Param)>4, error(['Stage ' int2str(S) ' parameter list > 4.']); end;
  ai = aiTable(Nstage,S);

  % Parse parameter vector.
  % C1.
  C1 = Param(1);
  % C2.
  if length(Param)>1,  C2 = Param(2);  else,  C2 = C1; end
  % R1.
  if length(Param)>2,  R1 = Param(3);
  else
    a = ((2*pi*Fc)^3) * C2 * ( C1 + (1-A)*C2 );
    b = -ai * C2 * ((2*pi*Fc)^2);
    c = 2*pi*Fc;
    R1 = ( -b - sqrt(b*b - 4*a*c) ) / (2*a);
  end
  % R2.
  if length(Param)>3,  R2 = Param(4);
  else,  R2 = 1 / ( ((2*pi*Fc)^2)*R1*C1*C2 ); end

  % Print component values.
  fprintf('Stage %d: C1=%e F, C2=%e F, R1=%e Ohms, R2=%e Ohms\n', S, C1, C2, R1, R2);

  % Update frequency response.
  a = C1*(R1+R2) + (1-A)*R1*C2;
  b = R1 * R2 * C1 * C2;
  Hstage = A ./ (  1 + ( a*j*w ) - ( b*w.*w )  );
  H = H .* Hstage;
end

% Now, plot the frequency response magnitude.
plot( f, abs(H) )
xlabel('Frequency in Hertz')
ylabel('Frequency Response Magnitude')

L1 = find( abs(H) <= (MagBig*sqrt(2)/2) );
hold on, plot([f(L1(1)) f(L1(1))], [0 MagBig], 'm'), hold off
Thing = sprintf('Cut-off at %0.2f Hz, Passband Gain = %0.2f', f(L1(1)), MagBig);
text( max(f)/5, max(abs(H))/3, Thing);
title('Butterworth Low-Pass Filter');

figure(gcf);

return
```