

College of Engineering, Department of Electrical and Computer Engineering

EE/COMPE 491W Senior Design

**Final Report**

**Team HANDS-EMG**

**ECE Team 9 – Hand Activity Neural Detection using sEMG**

**EE Members:**

Kelly Hubbard, Jayden Sumbillo, Kirk Young

**CE Members:**

Noah Marosok, Blake Pearson

**Sponsor:**

SDSU Department of Electrical and Computer Engineering

SDSU Smart Biomedical Systems Lab

|  |  |  |  |
| --- | --- | --- | --- |
| P18C1T1#yIS1  Kelly Hubbard  Electrical Engineering | P22C2T1#yIS1  Jayden Sumbillo  Electrical Engineering | | P26C3T1#yIS1  Kirk Young  Electrical Engineering |
| P31C4T1#yIS1  Noah Marosok  Computer Engineering | | P35C5T1#yIS1  Blake Pearson  Computer Engineering | |
| P40C6T1#yIS1  Dr. Hakan Toreyin  PI & Advisor | | | |

**Table of Contents**

[Executive Summary 3](#_Toc184844797)

[System Description 4](#_Toc184844798)

[System Test 11](#_Toc184844799)

[Key Technological Challenges and Strategies 13](#_Toc184844800)

[Risk Analysis and Prototype 14](#_Toc184844801)

[Key Integration Steps 25](#_Toc184844802)

[Work Breakdown Structure 30](#_Toc184844805)

[Task List and Schedule 33](#_Toc184844806)

[Cost and Financing 35](#_Toc184844808)

[Bill of Materials 36](#_Toc184844809)

**Table of Figures**

[Figure 1: Machine Learning Process 5](#_Toc184844831)

[Figure 2: Top-Down Mechanical Drawing 6](#_Toc184844832)

[Figure 3: Side View Mechanical Drawing 6](#_Toc184844833)

[Figure 4: Use Case Flow Diagram 8](#_Toc184844834)

[Figure 5: Device Block Diagram 9](#_Toc184844835)

[Figure 6: Signal Flow Diagram 10](#_Toc184844836)

[Figure 7: System Test Diagram 12](#_Toc184844837)

[Figure 8: Initial Risk Cube 14](#_Toc184844838)

[Figure 9: AFE Signal Acquisition Test Setup 16](#_Toc184844839)

[Figure 10: uC to AFE Communication Test Setup 17](#_Toc184844840)

[Figure 11: AFE Scope Output 19](#_Toc184844841)

[Figure 12: Communication Output 20](#_Toc184844842)

[Figure 13: ML Model Layer Diagram 22](#_Toc184844843)

[Figure 14: Classification Accuracy Plots 23](#_Toc184844844)

[Figure 15: Final Risk Cube 24](#_Toc184844845)

[Figure 16: Integration Plan Diagram 29](#_Toc184844846)

[Figure 17: Work Breakdown Structure (WBS) 32](#_Toc184844847)

[Figure 18: Project Milestones 34](#_Toc184844848)

[Figure 19: Critical Path 34](#_Toc184844849)

[Figure 20: BOM with Pricing 36](#_Toc184844850)

[Figure 21: BOM Item Descriptions 36](#_Toc184844851)

**Table of Tables**

[Table 1: Exercise Descriptions 18](#_Toc184844852)

[Table 2: Budget Allocation 35](#_Toc184844853)

# Executive Summary

This report provides a comprehensive overview of the prototyping activities conducted to mitigate risks identified during the development of our senior design project. Our surface electromyography (sEMG) sensor system is designed to classify hand movements using muscle activity signals from the forearm that are processed via an analog front-end module and a machine learning (ML) model. This report details the risks addressed, the mitigation strategies implemented, the results obtained, and the subsequent changes made to the project.

The two key risks targeted in this initial phase of prototyping were the accuracy of the ML classification algorithm and the configuration of the analog front-end (AFE) module. Addressing these risks was crucial to ensure reliable movement recognition and clean signal processing for accurate data acquisition. To mitigate these risks, our prototype activity was conducted, which included designing and testing the ML classification pipeline on a PC, configuring and validating the programmable gain amplifier (PGA) of the AFE module, and establishing communication between the AFE and microcontroller.

The results of the prototyping activities showed the validity of our system design and provided important insights into our focuses moving forward. We successfully developed and tested a machine learning model with acceptable accuracy for initial prototyping, validating the feasibility of our classification approach. Communication between the analog front-end module and the microcontroller was established, confirming the compatibility of these devices and ability to program the AFE. Finally, we identified the optimal gain settings for the programmable gain amplifiers (PGAs), resulting in clean and usable sEMG signals. However, the risk associated with the ML model’s initial accuracy was not entirely resolved and remains an area of focus as the project progresses. This recognition will guide the following prototyping phases.

# System Description

The HANDS-EMG device is a battery-powered surface electromyography sensor system designed to classify hand movements using machine learning, by interpreting the muscle activity signals captured from the user’s forearm. By utilizing sEMG technology, it provides a non-invasive method to monitor and analyze the muscle signals in real time. This functionality is useful for applications in prosthetics and rehabilitation where accurate and efficient movement recognition is essential. The device is portable, and battery powered, making it a practical and reliable tool for improving accessibility.

The machine learning model is trained and implemented through the process seen below in Figure 1. Training begins with raw EMG data from the public NinaPro dataset, which is processed using custom signal processing algorithms in MATLAB to extract meaningful features. These features and their associated classifications are input into a Python-based training script to build and evaluate the model. Once trained, the model is converted into a TensorFlow Lite file for deployment on the STM32 microcontroller. The microcontroller receives raw data from the analog front-end, processes it to extract features with digital signal processing, and uses the pretrained inference engine to classify movements and output the results in real time.

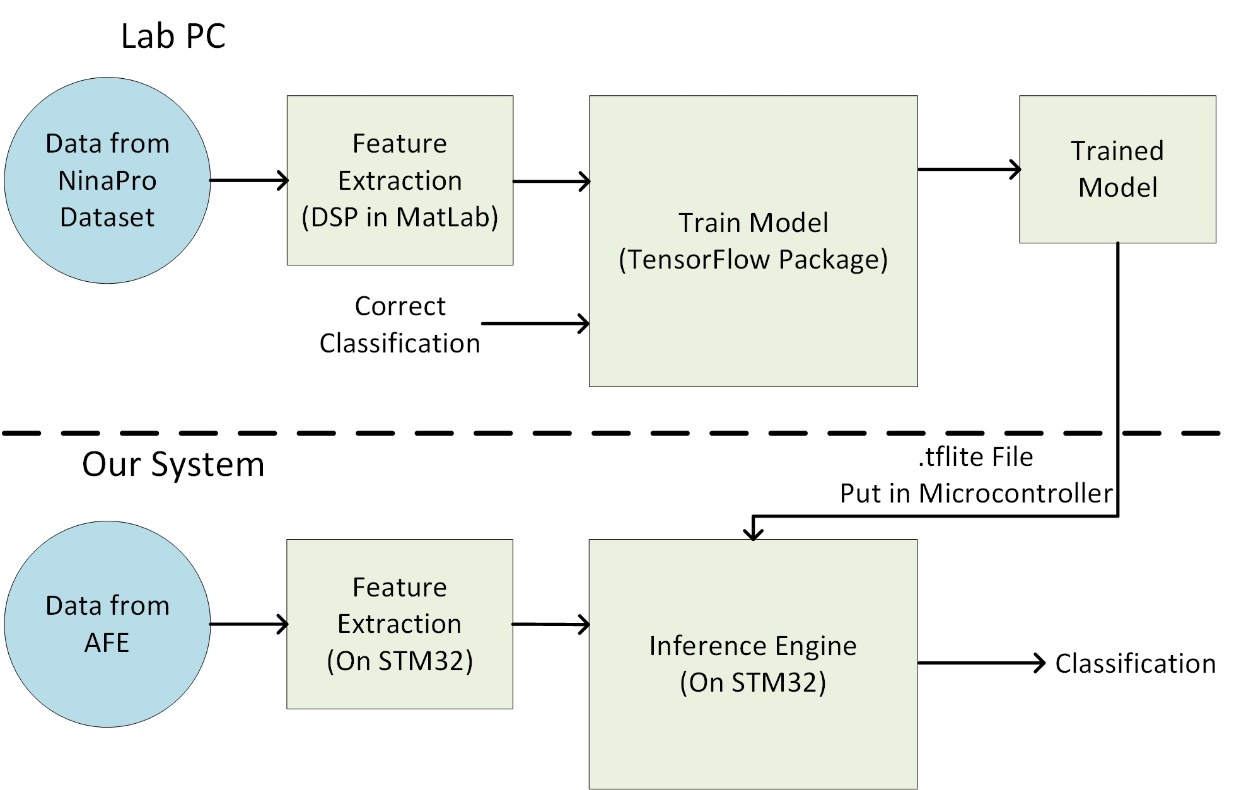


Figure 1: Machine Learning Process

*Figure 1 demonstrates the machine learning process that is conducted to attain a working model on our microcontroller. The process starts at the top left and works downwards.*

The device utilizes four channels of wet electrodes placed on the user’s forearm to capture the sEMG signals. The device is designed to be arm mounted and will not exceed the dimensions of 80 x 60 x 10mm and will not exceed a mass of 40g. A physical sketch including dimensions can be seen below in Figure 2 and Figure 3.

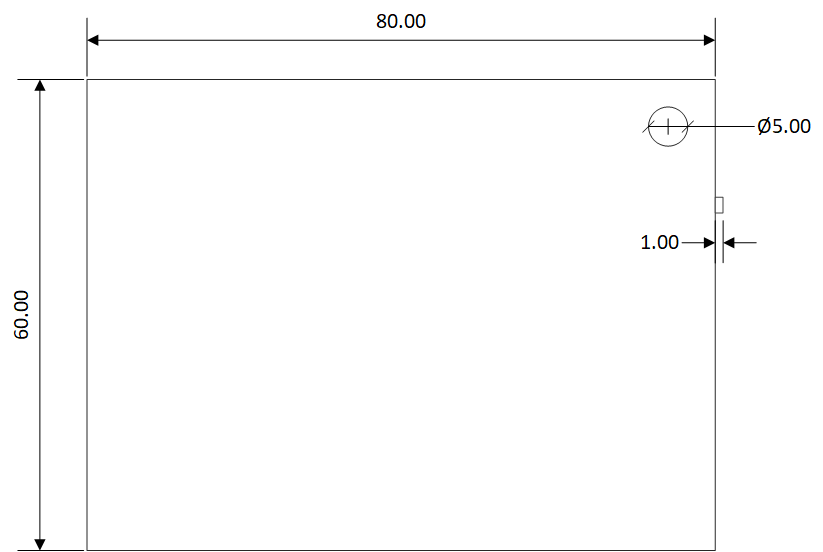


Figure 2: Top-Down Mechanical Drawing

*Figure 2 is the mechanical drawing of our physical device from a top view, the units are in mm. Notice the hole in the top right, this is for a status LED.*

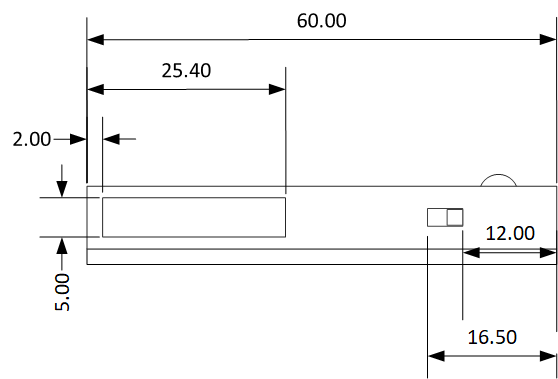


Figure 3: Side View Mechanical Drawing

*Figure 3 is the mechanical drawing of our physical device from a side view, the units are in mm. Notice the hole in the left, this is for I/O connections. The ON/OFF switch can also be seen towards the right.*

Our system is comprised of three main subsystems, each with specific tasks necessary to ensure the proper operation of our device. These subsystems are the ADS1299 analog front-end module (AFE), STM32 microcontroller, and the power management integrated circuit (PMIC). The AFE is a critical component, procured specifically for its ability to handle the complexity of sEMG signal acquisition. It amplifies and digitizes the weak bioelectric signals (typically in the range of 10uV to 10mV) from the forearm electrodes, ensuring that they are clean and suitable for further processing. The STM32 microcontroller serves as the central processor, performing feature extraction and classification using the machine learning model. The PMIC ensures stable power delivery to all components, enabling the device to operate with a 3.7V lithium-ion battery. These details and the interactions between the subsystems are illustrated below in the block diagram in Figure 5.

The analog signals captured by the four channels of wet electrodes are processed through our procured analog front-end module which operates at a sample rate of 2KHz. The AFE digitizes the signals and transmits them to our microcontroller via the SPI protocol operating at a data transfer rate of 4MHz. Within the microcontroller, a pre-trained machine learning model classifies these hand movements with an accuracy of no less than 70%. This process can be visualized below in our signal flow chart seen in Figure 6.

The device is powered by a 3.7V rechargeable lithium-ion battery, selected for its portability and energy density. The battery has a capacity of 400mAh, which will provide adequate charge to sustain the device’s power demands. Power is managed directly by the PMIC which ensures stable voltage levels and efficient battery charging through the USB Micro B port. The PMIC regulates and outputs the required voltages for each subsystem: 3.3V for the STM32 microcontroller and 3.3V/5V for the ADS1299 AFE. The total current consumption of the device is estimated to not exceed 10.43mA under normal operation, which allows for ~38 hours of continuous use.

To begin using the device, the user will prepare their forearm, ensuring the area is clean and dry. Eight electrodes are placed radially around the proximal forearm, spaced approximately 2 cm apart, and positioned 2-3 cm distal from the elbow crease. These electrodes are then connected to the corresponding input pins on the device for each channel. A reference electrode is placed on the elbow and connected to the designated REF pin. Once the electrodes are in place, the user runs the simulator software on a PC, connects the device to the PC via USB Micro B, and powers on the device. As the user moves their hand, the device processes the sEMG signals and transmits the classified movements to the PC, where a visual simulation displays the corresponding hand movements in real time. These steps can be seen below in Figure 4.

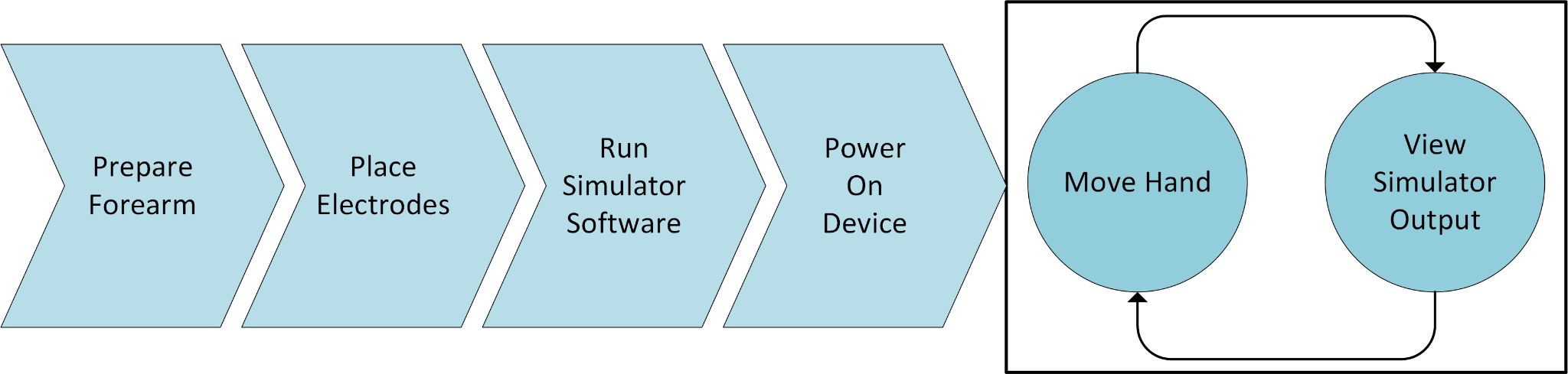


Figure 4: Use Case Flow Diagram

*Figure 4 shows the steps a user will take to begin using the device. On the right, the infinite use loop can be seen demonstrating prolonged use.*

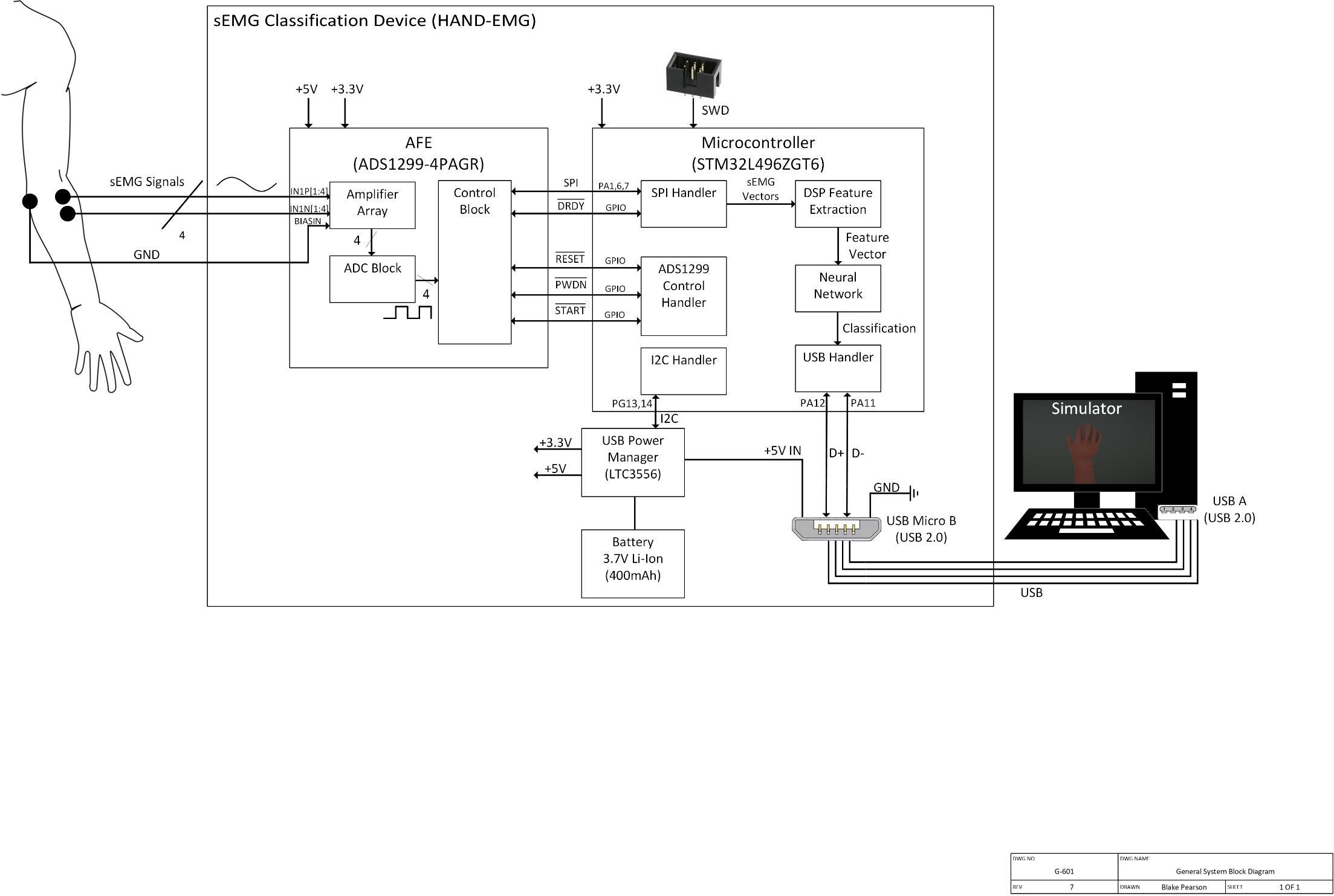


Figure 5: Device Block Diagram

*Figure 5 shows the detailed block diagram for our device. Notice the 4 major components: AFE, microcontroller, power manager, and simulator. These subsystems work in conjunction to perform the desired classification task*

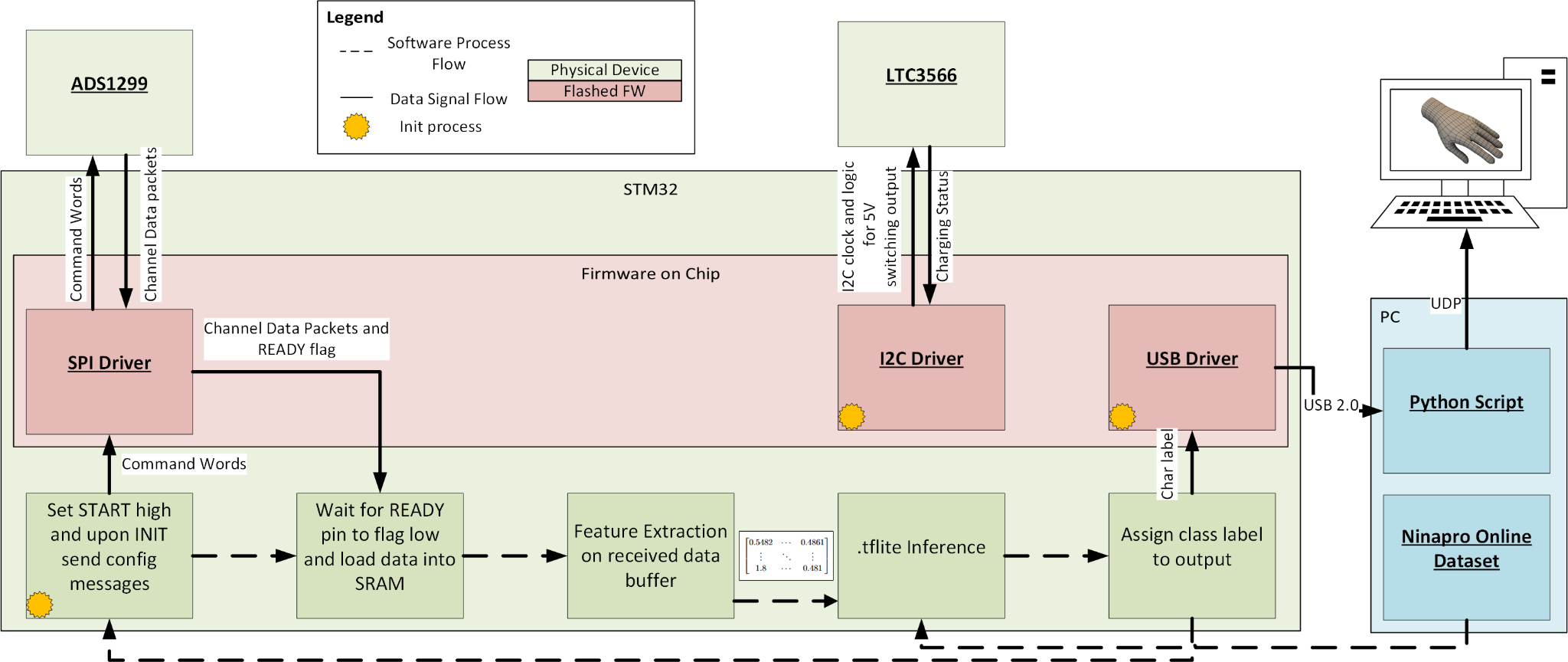


Figure 6: Signal Flow Diagram

*Figure 6 shows the path a signal will take in our software from acquisition from the AFE (ADS1299) to displaying in the simulator. This diagram provides a comprehensive overview of the innerworkings of our software.*

# System Test

For our system test we’ve designed it to demonstrate the fully integrated system’s ability to recognize and classify hand gestures using sEMG signals. The test ensures the key functionalities are working as expected. This includes the signal acquisition from the electrodes, system processing, and simulator display.

**Procedure**

1. Ensure contact area for wet electrodes is clean and dry, then place eight radially around the upper forearm 2 cm apart positioned 2-3 cm from the elbow crease.
2. Once the electrodes are connected to the system, turn the device and simulator on
3. Perform 5 hand gestures 5 times each pausing momentarily between repetitions and be sure to have clear and consistent movements for accurate classification

* First Clench: Clench the hand into a fist, relax and repeat
* Thumbs Up: Perform a thumb-up, relax and repeat
* Peace Sign: Perform a peace sign, relax and repeat
* Pointing Index Finger: Pointing index, relax and repeat
* Abduction Of All Fingers: All fingers, extended hand, relax and repeat

**Expected Results**

Our simulator will correctly display the performed gestures with less than 30% error during all repetitions.

**Equipment Required**

* HANDS-EMG Device
* Electrode Leads
* Wet Gel Electrode Pads
* Laptop

**Success Criteria**

Our system will be defined as successful if

1. The simulator accurately recognizes and displays the gestures performed by the user with 70% accurately displayed
2. The software outputs align with the expected classifications

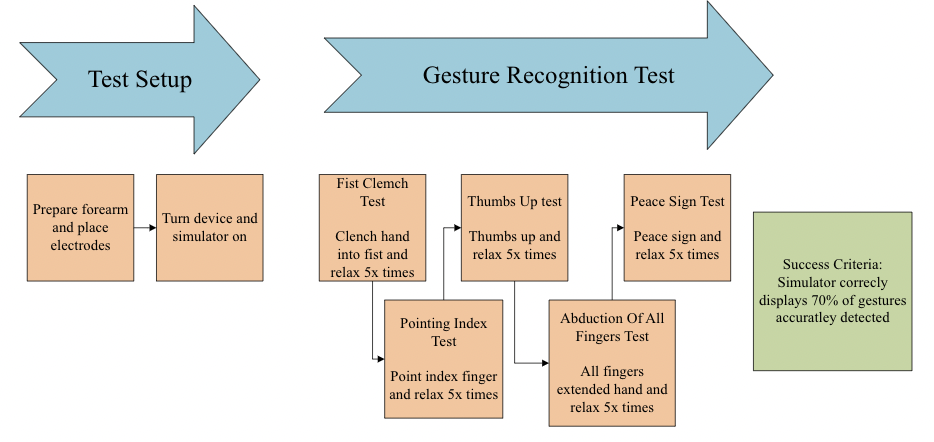


Figure 7: System Test Diagram

*Figure 7 shows the steps our team will take during the marketing demonstration to prove our device is complete. Success criteria can be seen in the green box.*

# Key Technological Challenges and Strategies

The development of our gesture recognition system presents a multifaceted challenge. This section outlines the key technical challenges and proposed strategies to address them, ensuring the successful realization of the project in Senior Design B.

The primary challenge lies in achieving high accuracy for the machine learning model. To address this, we will continue to leverage public datasets for training and validation. We will also refine the model using data collected from the analog front-end module (AFE) to improve its performance and adaptability to real-world scenarios.

The limited memory and processing capabilities of microcontrollers pose a significant constraint on the size of our machine learning model. To overcome this, we will design a lightweight model that fits within the device's constraints. Additionally, we will employ an incremental approach, adjusting the model's complexity based on the evolving demands during prototyping.

Effective signal processing is crucial for extracting meaningful features from the raw sensor data. We will implement DSP algorithms in C to process the digitized signals from the AFE, ensuring compatibility with the training data. Aligning the processed data with the public dataset will further enhance the accuracy of the classification process and getting outputs like the public dataset presents a challenge. Our strategy for this is to begin prototyping early and work through breaks to ensure the data is aligned.

The integration of multiple components on a single PCB presents a complex design challenge. To mitigate risks, we will validate the design through Design Rule Checks (DRCs) and prototyping. Additionally, we will employ techniques such as separate analog and digital ground planes and optimized signal routing to reduce noise and ensure reliable signal transmission.

Successful system integration hinges on the seamless communication and coordination between the AFE, microcontroller, and power management system. To achieve this, we will design individual PCBs for each core module, allowing for independent testing and verification. Early identification and resolution of potential design or communication issues will be crucial to the overall system's performance.

# Risk Analysis and Prototype

**Initial Risk Analysis**

Prior to prototyping this design, primary risks need to be identified and assigned a priority. In this case, the risks to be analyzed should be pulled from the previous section, where they have been explained in detail. By utilizing a risk cube in Figure 8 figure reference, which assigns priority of risks to be addressed as a function of probability of the risk event occurring, and the severity of the risk as it pertains to total system performance.

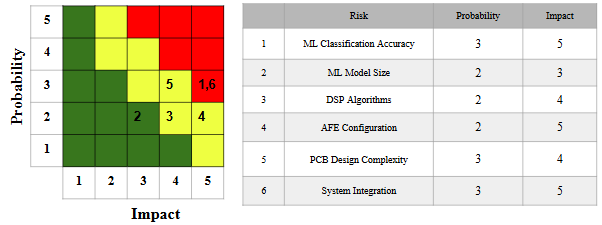


Figure 8: Initial Risk Cube

During prototyping of the system, the goal and tasks associated with prototyping need to be specific to address one or more of the risks listed above. Primarily, the prototyping stage should address risks that lie within the red areas of the risk cube plot. It was therefore decided to create prototype activities to address risk 1,2, and 4.

**Prototype Activity #1:**

This activity involved utilizing the AFE evaluation kit device with the electrodes to find the optimal gain configuration in the onboard PGA. This ensures the AFE device produces minimal digital noise and signal clipping with a consistent signal strength that matches closely with the training data. Additionally, the SPI communication between the STM32 and the AFE was attempted to be established. This was done to verify the ability to send commands to the AFE from the microcontroller and query data in response. Below in Figure 9 add fig REFERENCE, the experimental setup is shown. Electrodes are placed as described in the system description section and connected to the AFE which connects to Texas Instruments’ platform specific software.

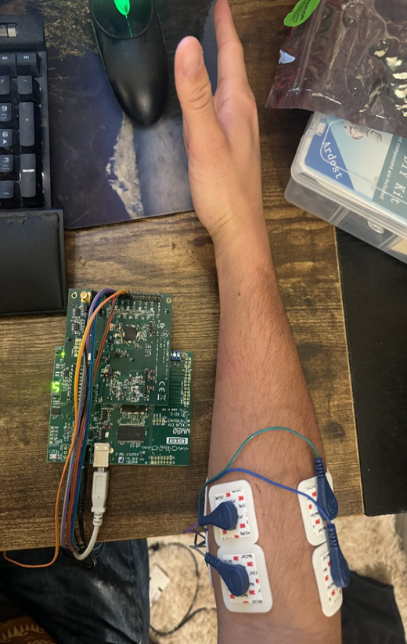


Figure 9: AFE Signal Acquisition Test Setup

*Figure 9 shows the test setup used for determining the PGA settings on the AFE. Our group member Kelly was hooked up to the electrodes for this prototype activity.*

Additionally, the test setup for digital serial communication is seen in figure Figure 10 which shows both the STM32 and AFE connected using the logic analyzer.

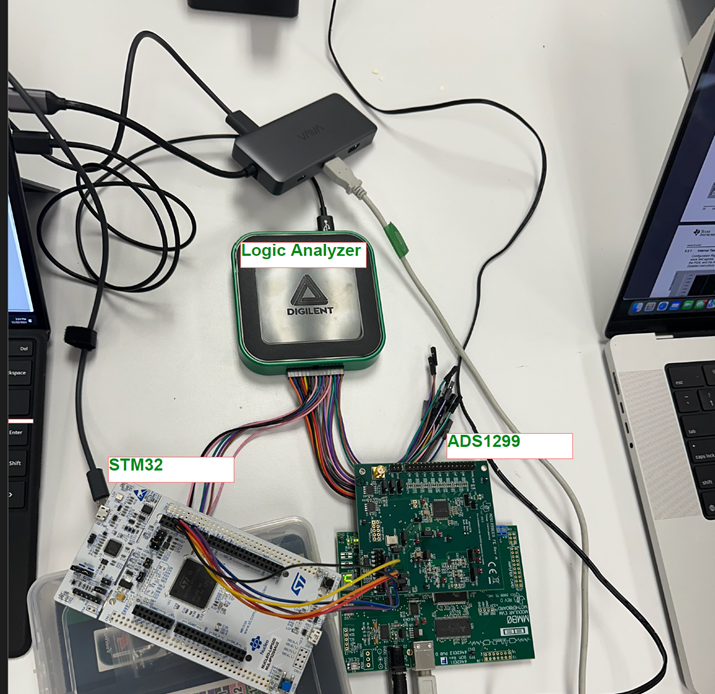


Figure 10: uC to AFE Communication Test Setup

*Figure 10 shows the connections made to test microcontroller communication with the AFE. A logic analyzer is used in this setup to watch the data being transferred between the devices.*

The exercises to be performed were prescribed from the Ninapro dataset and are all part of one exercise. This includes 18 unique hand movements Table 1 to include for experimental purposes.

Table 1: Exercise Descriptions

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exercises | | | | | | | | |
| 1 | Thumb up | P185C4T2#yIS1 | **7** | Pointing index | P188C7T2#yIS1 | **13** | Wrist flexion | P191C10T2#yIS1 |
| 2 | Extension of index and middle, flexion of others | P195C13T2#yIS1 | **8** | Adduction of extended fingers | P198C16T2#yIS1 | **14** | Wrist extension | P201C19T2#yIS1 |
| 3 | Flexion of ring and little finger extension of others | P205C22T2#yIS1 | **9** | Wrist supination (axis middle finger) | P208C25T2#yIS1 | **15** | Wrist radial deviation | P211C28T2#yIS1 |
| 4 | Thumb opposing base of little finger | P215C31T2#yIS1 | **10** | Wrist pronation (axis middle finger) | P218C34T2#yIS1 | **16** | Wrist ulnar deviation | P221C37T2#yIS1 |
| 5 | Abduction of all fingers | P225C40T2#yIS1 | **11** | Wrist supination (axis little finger) | P228C43T2#yIS1 | **17** | Wrist extension | P231C46T2#yIS1 |
| 6 | Fingers flexed together in fist | P235C49T2#yIS1 | **12** | Wrist pronation (axis little finger) | P238C52T2#yIS1 |  |  |  |

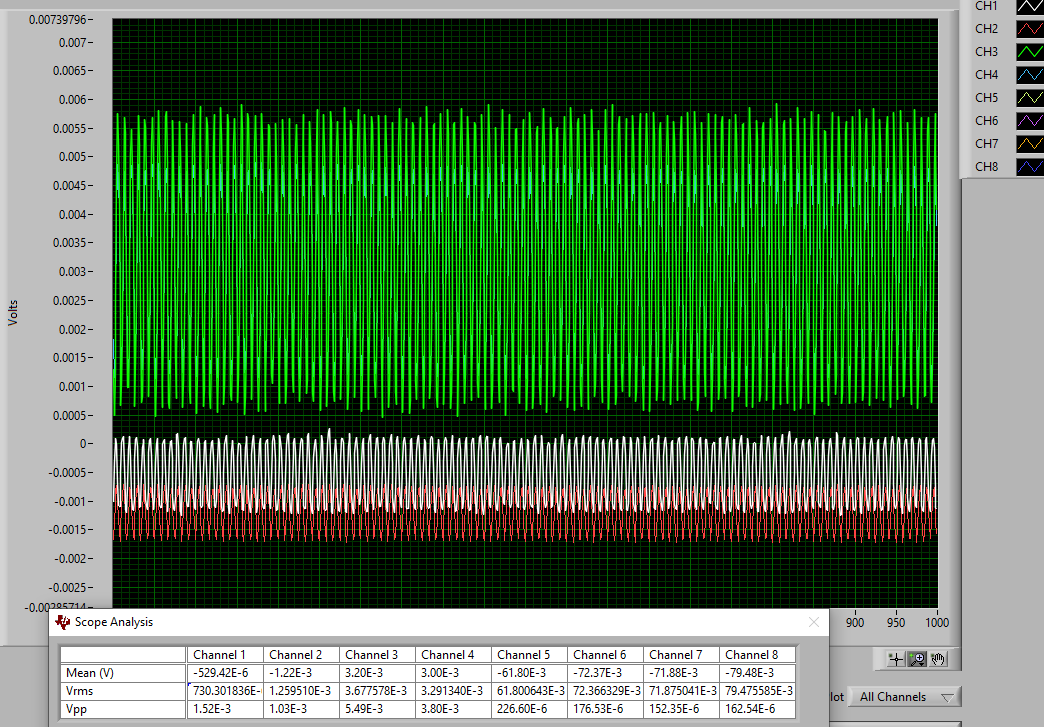


Figure 11: AFE Scope Output

*Figure 11 shows the output of the AFE measured with the internal oscope on the eval kit’s daughter card. This was taken during the prototype activity when testing PGA values.*

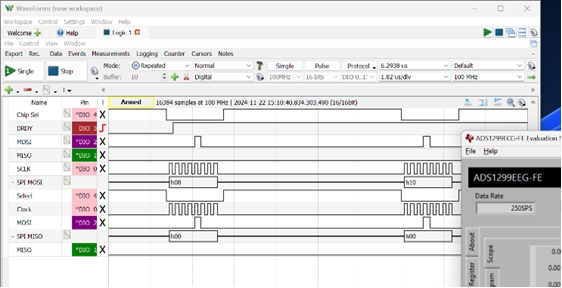


Figure 12: Communication Output

*Figure 12 shows the output of the logic analyzer when digital communication between the AFE and microcontroller was occurring. It shows that the command words were successfully sent.*

**Result**

The result of this prototyping effort left us with satisfactory results. The signal on the onboard scope of the AFE closely aligned with our benchmark Ninapro dataset within a range of ±10mV. The signals were visually inspected and showed no observable noise and no apparent clipping throughout all the movements displayed in the TI software. The communication between the two devices was also a success, as the microcontroller successfully sent control words to the AFE, and received data in response as shown in figure Figure 11 and Figure 12.

**Validation Analysis**

1. Signal Amplitude Consistency (Vpp, Vmin, Vmax,)

* The peak-to-peak voltages (Vpp) range consistently between ~1mv to ~5mv, which aligns with expected sEMG signal amplitudes for differential electrodes. The Vmin and Vmax values do not show any clipping or saturation as there was no flattening out of the waveforms. This indicates the PGA gain of 12 was the right choice for capturing the signal range without any wave distortion.

2. Nosie Rejection (Vrms and Mean V)

* Vrms values are stable across the exercises, indicating consistent signal power. The Mean V are near zero for most channels and exercises, confirming minimal DC offset/drift

3. Channel Correlation

* When comparing Vrms, Vpp, and Mean V across the four channels, their values remain consistently comparable within all the exercises. This shows that the wet electrodes are in fact capturing the signals uniformly. Any variations are expected with the different muscle activation patterns.

4. Range Across Exercises

* Exercises with higher muscle activation patterns such as exercise 11 (Figure 6) showing up to around 5.5mv on channel 3 display higher Vpp and Vrms, while less intensive hand configurations show lesser values. This was important as we saw it could differentiate muscle activation levels accurately and can confirm it as channel 3 was on the upper forearm for exercise 11 which was intensive in that region.

5. Absence of Clipping or Distortion

* None of the Channels had any extreme Vpp values that would indicate clipping or distortion. With a stable Vrms and bounded Vpp, we can confirm the PGA gain setting effectively avoids saturation while capturing usable signals for the ML model to interpret.

6. Microcontroller Communication

* We were able to successfully send control words and receive data from the ADS1299 using our microcontroller. The test setup for this can be seen below in Figure 12. Where the logic analyzer is set up to read the SPI lines between the ADS and the microcontroller. In Figure 11 we can see the data read on the SPI lines clearly showing the proper commands (START h08 and RDATAC h10) being sent to the ADS.

**Prototype Activity #2:**

For machine learning prototyping, we are addressing the identified risk 1 and risk 2. Risk 1 is specifically to guide the overall performance of the model. Choosing this as a high priority risk assures that the overall system performs as optimally as possible. Initially, it is good practice to make the model as basic as possible for a given application. For our application, we are limited by the size constraints of the microcontroller, as stated in risk 2. Thus, careful consideration needs to be made to make sure that there is a fine balance between classification accuracy and total file size and computational load during inference.

**Prototyping Strategy and Objectives**

The primary goals of prototyping the TensorFlow model lies in ensuring the model is not too large to fit on the 1MB flash on our device, and during an inference phase, the model cannot use more than 80% of the 320KB SRAM to prevent computational overload for other processes. Thus, we will create a basic model architecture, perform analysis on that model to ensure it fits within physical device constraints, and then train and evaluate the performance. Below are steps that can summarize this prototyping process:

1. Choose relevant channels that provide the most classification accuracy, and extract relevant features to decrease computational load
2. Declare architecture within the TensorFlow library in Python with consideration for: hyperparameters, model depth, model span, and optimizer settings

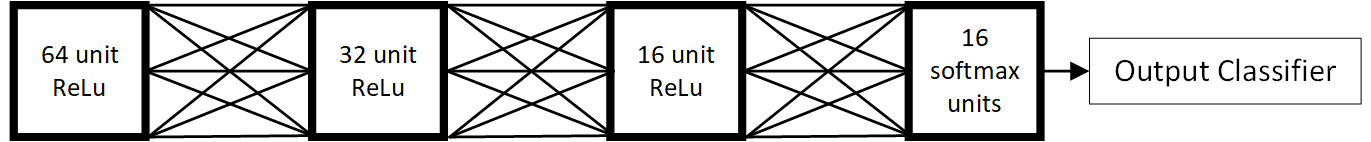


Figure 13: ML Model Layer Diagram

1. Split, shuffle and feed data into the model monitoring performance after each epoch
2. Evaluate model performance and repeat steps 2-4 as needed

**Result**

The model described in Figure 13 has 5,761 total parameters. Multiplying this by 4 bytes we get that there is a total space required of 22.5KB, which fits well below the required threshold of 800KB

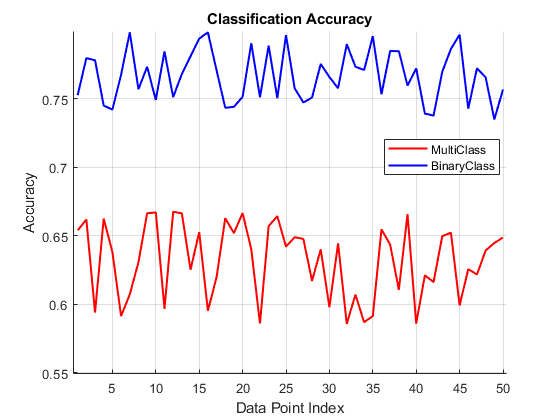


Figure 14: Classification Accuracy Plots

After evaluating the performance of the model with 50 runs with a static number of epochs at 50, the model performed with an average accuracy of 58.3%, and a peak performance of 67.01%. The average metric here is not necessarily significant due to the stochastic nature of machine learning algorithms, but including here aims to clarify the performance of the model for generalizability. This value obviously does not meet our predefined threshold, or the current accuracy rate of competing literature thus further investigation was performed to determine if the problem was with the model, or the data. To do this the problem definition was changed to perform a binary classification with class 0 representing a resting position, and class 1 representing an active position. In practice, this simply means changing the format of the input data, i.e. skipping one hot encoding, and changing the output layer activation function to a single unit sigmoid function. A sigmoid is defined as the logit transform of the data, where the output is a probability, and an output is found by a predefined threshold probability. The default of this value is 0.5. After running the same script, with the same number of iterations and epochs, the average performance was 73.4% with a peak performance of 80.1% accuracy. This can be visualized in Figure 14.

It also must be briefly mentioned that feature extraction may not be performed exactly as expected on the device if quantization is not considered. For this, certain considerations have already been applied. The functions that extract features in MATLAB have an adjustable size in output bit width, and additionally, the data received from the AFE is in a 12-bit format, while the STM32 can natively perform operations in 32-bit precision floating point. For these reasons, the risk impact was not placed as a high priority, but needed to be mentioned to ensure the reader has a comprehensive understanding of the current prototype.

Since sEMG actuations can be performed via simple thresholding with near perfect performance, an accuracy of below 99% is unacceptable. This presents several potential avenues for further investigation. Firstly, further data analysis needs to be conducted on raw channel signals to determine what channels of the ten available to us are of most significance to determining the output of the set of movements within the exercise. This involves performing a principal component analysis to see which channels explain the most variance in the data and dropping the last four channels. Then we perform feature extraction and investigate the performance from there. The second method of triaging the performance involves altering the architecture of the model completely. It may be necessary to create a 12x12 2D feature map of each channel and use 2D convolutional layers to extract the proper inherent temporal, and spatial relationships within the data. This would have to be done with caution to ensure, as before, that the size of the model does not surpass our physical capabilities.

**Risk Adjustment**

After our prototyping activities were performed, we were left with an adjusted idea of the risks in this project. In figure Figure 15 risk cube below, the adjusted risks are shown.

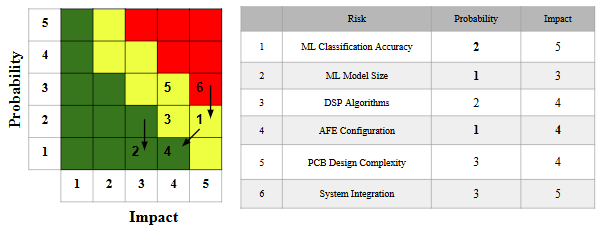


Figure 15: Final Risk Cube

Risk 4 and 2 were moved to the lowest probability of occurring, as the prototype activities ensured that they will be extremely unlikely to occur, as verified by our results. Risk 1, however, was not entirely addressed yet was still moved to a lower probability level. This is because likely causes were identified, and steps will be taken to adjust the performance to a desired level.

# Key Integration Steps

To ensure a seamless integration process, all subsystems were assigned a well-defined plan that involves various testing procedures. Each step is intended to validate the functionalities of all subsystems independently, ensuring reliable performance before integrating them into a single device. The following subsections outline the key integration steps required to achieve a fully functional system.

## **Procedure**

The table below details all major subsystems, alongside their specific testing procedures that are performed to ensure system readiness, and an accurate output.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Subsystem | Testing Procedure | Expected Result |
| **1** | Power Management | * Test voltage and current stability under typical and maximum load conditions * Verify battery charging with Micro USB * Verify ability to power on/off device | * Provide stable power delivery (3.3V/5V) to all components * Able to operate for a minimum of 6 hours. * Device power switch shuts off voltage to other subsystems |
| **2** | Microcontroller | * Test feature extraction with NinaPro data * Validate firmware for subsystem control * Ensure uploaded ML Model makes proper classification with NinaPro data | * Features extracted within the microcontroller match our features extracted in MATLAB that we used to train the model * Send control signals to other subsystems and verify expected change * Each NinaPro exercise data should classify the same as it is in the data set |
| **3** | Analog Front End | * Connect user with electrodes to the subsystem, confirm that sEMG data is being collected and digitized | * sEMG data when performing exercises matches NinaPro data set closely |
| **4** | Communication & Control Protocols | * Verify SPI communication between ADS1299 (AFE), STM32 (uC), and LTC3556 (PMIC) using a logic analyzer * Test USB communication between STM32 and PC simulator | * Reliably transfers data between all core modules with no interruptions in SPI and USB Micro B communication |
| **5** | Electrode Placement & Mount | * Test electrode placement on forearm to determine the optimal position for consistent signal acquisition * Secure device in a custom-designed arm mount | * Wet gel electrodes are placed on validated locations on the forearm supported by testing data * Acquire uniform and stable signals from all four electrode channels |
| **6** | Simulator Interface | * Validate real-time demonstrations of classified hand gestures through the PC simulator | * Accurately demonstrates real-time hand gestures with minimal latency |

## **Integration Plan**

The figure below showcases the sequential plan of manufacturing, testing, and integrating all subsystems into one functional device.

1. **There is a total of three major subsystems, with each being manufactured as a separate printed circuit board (PCB): power management integrated circuit (PMIC), analog front end (AFE), and the microcontroller (uC).** 
   1. The PMIC is first tested to determine its reliability in power delivery (critical test)
   2. After validation and verification, the PMIC is connected to the remaining two subsystems solely as the main power supply.
2. **The AFE and uC are tested in parallel through their own specific testing procedures**
   1. AFE module undergoes a Signal Acquisition Test to ensure reliable data capture (critical test)
   2. The uC undergoes Flash and ML Classification Tests
      1. Flash Test (minor test) is performed to validate the uC’s accuracy in programming and capability to properly send and receive commands to other subsystems via communication interfaces
      2. ML Classification Test (critical test) is performed to verify the uC’s capability to handle the model and accurately perform classification operations
3. **All three major subsystems, after being independently validated and verified, are connected together as a prototype** 
   1. Power delivery circuit powers both the AFE and uC
   2. AFE captures and processes test signals
   3. The uC receives the processed signals, classifies them as hand movements, monitors the charging operations of the PMIC
4. **The final integrated PCB is designed, manufactured, and tested**
5. The PCB is precisely designed to minimize the amount of noise interference and heat dissipation through the following methods:

* Ensure analog and digital components are kept separate
* Use short signal paths
* Opt for a multilayer board to isolate power and signal planes

1. The integrated system is tested to ensure it functions as intended

* All operations described in Step 3 are repeated and verified



Figure 16: Integration Plan Diagram

*Figure 16 shows our integration plan beginning in Senior Design B with grey indicating setup procedures, red indicating critical tests, and blue indicating minor tests.*

# Work Breakdown Structure

The provided Work Breakdown Structure (WBS) seen in Figure 17 offers a comprehensive overview of our project’s structure, breaking down the HANDS-EMG Device into smaller, manageable work packages. The WBS is organized into four levels, with each level representing increasing detail.

* Level 1: This top level encompasses the entire project.
* Level 2: Here, the project is divided into five major work packages: Core Deliverables, Hardware, Software, Firmware, and Final Assembly & Test.
* Level 3: Each of these major work packages is further subdivided into more specific tasks. For instance, the “Hardware” package includes subtasks like "Procurement," "Industrial Design," and "PCB Design."
* Level 4: The lowest level provides the most granular breakdown of the project, listing specific tasks and deliverables such as "Electrodes," "Microcontroller," and "AFE PCB" under the "Hardware" package.

**Key Components and Outcomes**

The WBS highlights several key components that are essential to the project’s success. These include:

* **Hardware:** This encompasses the physical components of the device, such as the AFE, microcontroller, sensors, and the PCB. The team is responsible for procuring these components, designing the PCB, and assembling the device.
* **Software:** The software component involves the development of the simulator, as well as the machine learning model for gesture recognition. The team will be responsible for training the ML model, creating the user interface, and ensuring smooth integration of the software with the hardware.
* **Firmware:** This component focuses on the low-level software that controls the hardware, such as the AFE and microcontroller. The team will develop the code necessary for data acquisition, signal processing, and communication with the higher-level software.

By effectively managing these key components and deliverables, our team can ensure the timely and successful development of the HANDS-EMG device.



Figure 17: Work Breakdown Structure (WBS)

# Task List and Schedule

The provided Gantt link below visually represents the project timeline, breaking down the project into five distinct phases. Each phase is associated with specific milestones and activities, clearly indicating the project's progress and key deliverables. Our critical path can be seen below in Figure 19. Our project milestones can be seen below in Figure 18.

[https://prod.teamgantt.com/gantt/list/?ids=4072715&public\_keys=9NPPKWp2U8NQ#](https://prod.teamgantt.com/gantt/list/?ids=4072715&public_keys=9NPPKWp2U8NQ)

## P396#y2P396#y1

Figure 18: Project Milestones

Figure 19: Critical Path

# Cost and Financing

Our project operates within a $1,000.00 budget, made possible by a generous grant from the Andrew Y. J. Szeto Rehabilitation Engineering and Assistive Technology Endowment Fund. This funding has been strategically allocated across three critical phases: Prototyping, Integration, and Testing. To date, we have allocated $950.00, with expenditure covering essential components such as the ASD1299 AFE module, STM32 microcontroller and PCB fabrication. Our allocated budget is based on expected costs and the bill of materials that can be seen in Figure 20and Figure 21 below.

Table 2: Budget Allocation

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Description | Components | Allocated Budget |
| Phase 1 | Initial Prototyping | Electrodes, Microcontroller, AFE, and PMIC eval kit | $400.00 |
| Phase 2 | Subsystem Integration Pairing | None | $0.00 |
| Phase 3 | Total Subsystem Integration | Li-ion battery, temporary platform | $50.00 |
| Phase 4 | Subsystem Integration on PCB |  | $200.00 |
| Phase 5 | Total System Integration | Device enclosure, finalized PCB | $300.00 |

### Bill of Materials

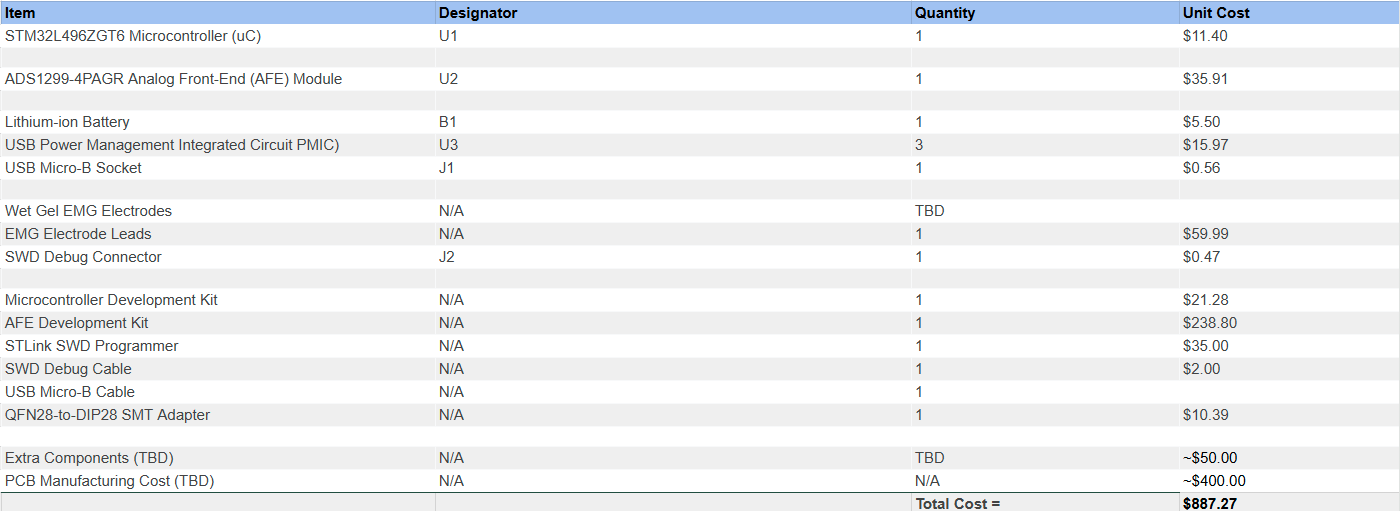


Figure 20: BOM with Pricing

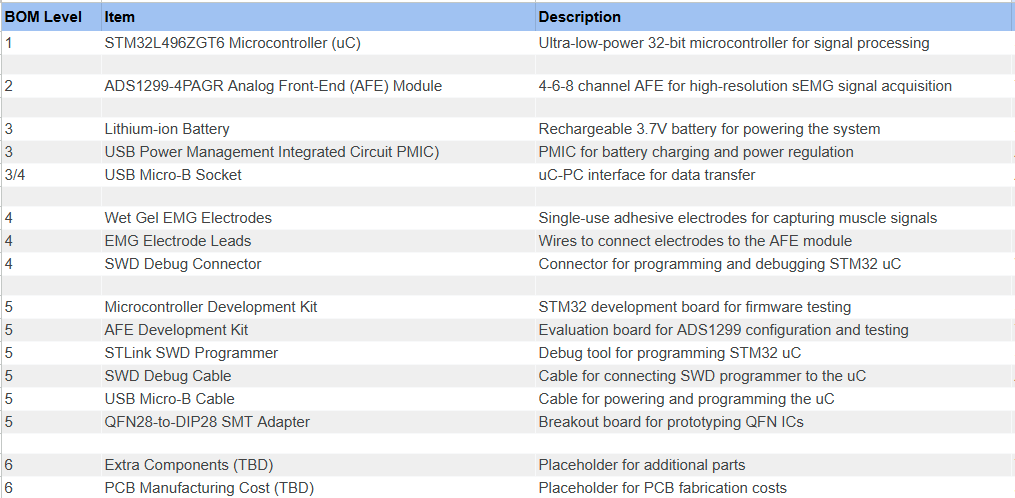


Figure 21: BOM Item Descriptions

[BOM Link](https://docs.google.com/spreadsheets/d/1DCkLrGYhf4FIt8fBN8eB10ucAKSWCuDaVUdKTYzPCas/edit?usp=sharing)