

College of Engineering, Department of Electrical and Computer Engineering

EE/COMPE 491W Senior Design

**Prototype 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

|  |  |  |  |
| --- | --- | --- | --- |
| Kelly Hubbard  Electrical Engineering | Jayden Sumbillo  Electrical Engineering | | Kirk Young  Electrical Engineering |
| Noah Marosok  Computer Engineering | | Blake Pearson  Computer Engineering | |
| Dr. Hakan Toreyin  PI & Advisor | | | |

**Table of Contents**

[Executive Summary 4](#_Toc183197970)

[System Description 5](#_Toc183197971)

[Initial Identification of Risks 10](#_Toc183197972)

[Prototype Activity 13](#_Toc183197973)

[AFE Configuration Risk Mitigation 13](#_Toc183197974)

[Machine Learning Risk Mitigation 22](#_Toc183197975)

[Final Risk Analysis 24](#_Toc183197976)

**Table of Figures**

[Figure 1: Machine Learning Process 4](#_Toc183208899)

[Figure 2: Top-Down Mechanical Drawing (units in mm) 5](#_Toc183208900)

[Figure 3: Side View Mechanical Drawing (units in mm) 5](#_Toc183208901)

[Figure 4: Device Block Diagram 7](#_Toc183208902)

[Figure 5: Signal Flow Diagram 8](#_Toc183208903)

[Figure 6: Risk Cube 11](#_Toc183208904)

[Figure 7: Comparison of Vmin and Vmax with our configuration 14](#_Toc183208905)

[Figure 8: Comparison of Vpp with our configuration 15](#_Toc183208906)

[Figure 9: Waveform of Exercise 11 on Scope 17](#_Toc183208907)

[Figure 10: Setup for Performing Exercises 1-17 18](#_Toc183208908)

[Figure 11: Scope AFE Communication with STM32 20](#_Toc183208909)

[Figure 12: Microcontroller and AFE Communication Setup 20](#_Toc183208910)

[Figure 13: ML Model Layer Diagram 21](#_Toc183208911)

[Figure 14: ML Classification Accuracy Graph 24](#_Toc183208912)

[Figure 15: Updated Risk Chart 27](#_Toc183208913)

[Table 1: Identified Risks 11](#_Toc183208299)

[Table 2: Voltage Features of Measured sEMG signals in mV 15](#_Toc183208300)

[Table 3: Exercises Performed in Testing AFE 16](#_Toc183208301)

[Table 4: Updated Risk Table 25](#_Toc183208302)

# 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.

A diagram of a system

Description automatically generated

Figure 1: Machine Learning Process

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.

A screenshot of a computer

Description automatically generated

Figure 2: Top-Down Mechanical Drawing (units in mm)

A drawing of a diagram

Description automatically generated with medium confidence

Figure 3: Side View Mechanical Drawing (units in mm)

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 4.

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 5.

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.

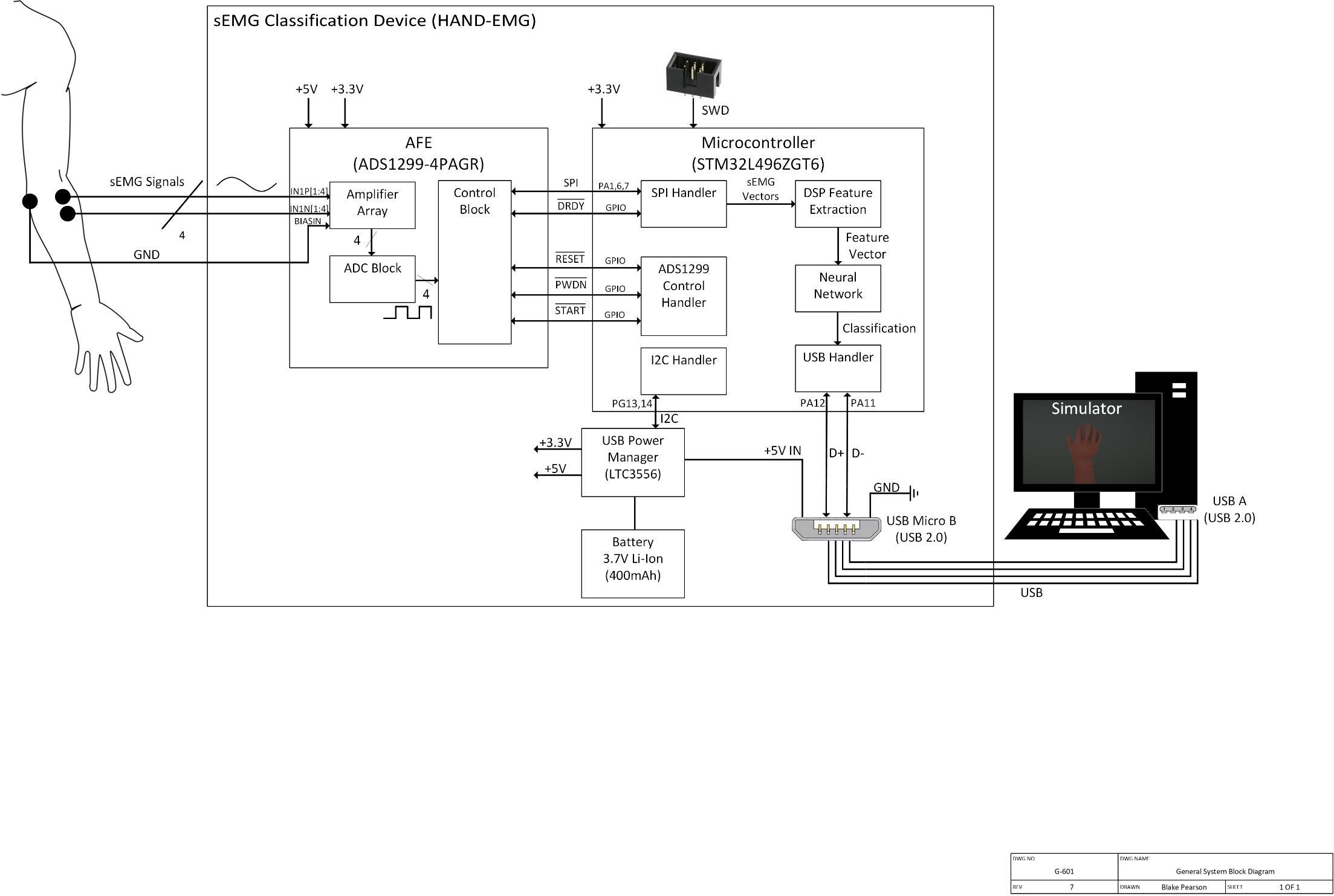


Figure 4: Device Block Diagram

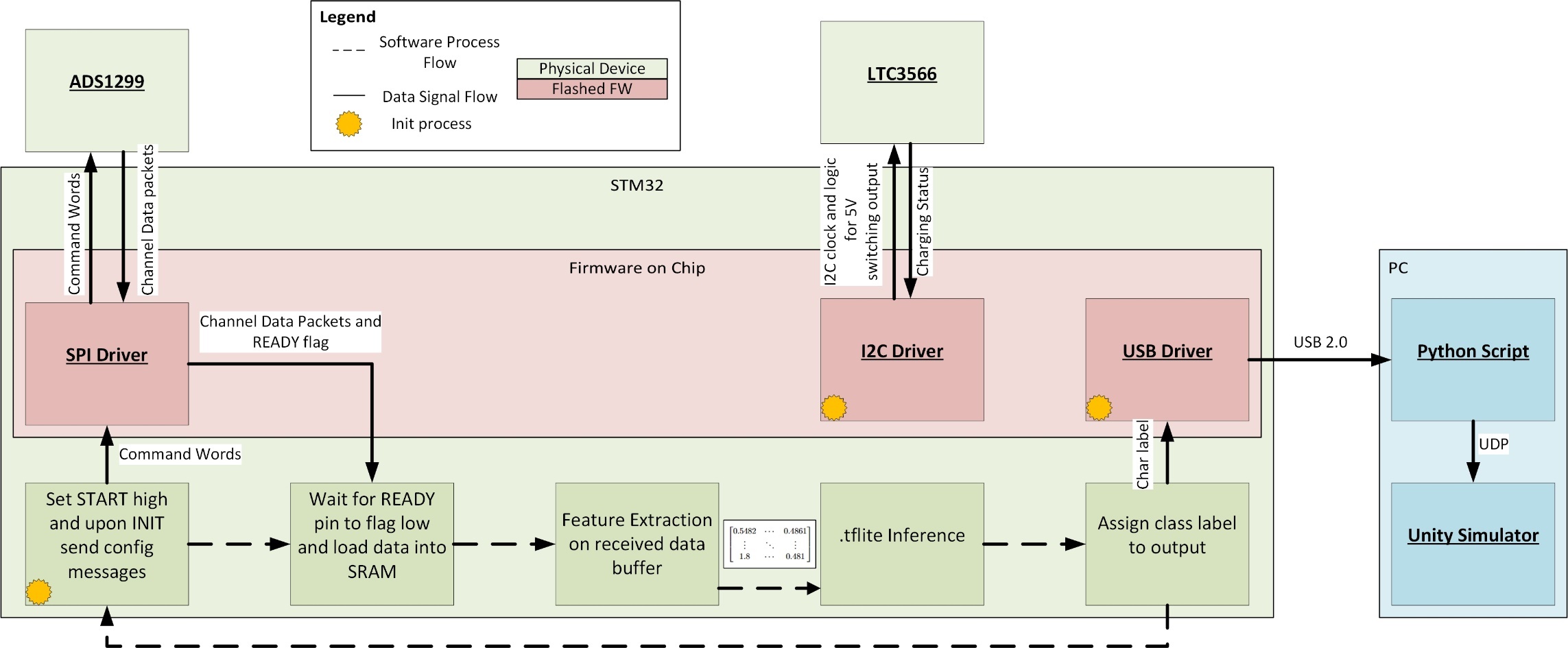


Figure 5: Signal Flow Diagram

# Initial Identification of Risks

Following the risk management meeting, six significant risks were identified, two of which were prioritized for initial mitigation activities due to their high impact on the project. These risks were determined through a combination of research, advisor feedback, and a thorough analysis of the project’s technical requirements. Each risk originates from the complexities involved in the design of a reliable sEMG based classifier on a resource constrained device. The risks identified include challenges related to the machine learning model accuracy, computational resource limitations, digital signal processing (DSP), analog front-end configuration, and PCB design complexity.

The first and perhaps most critical risk involves the accuracy of the machine learning model. The industry standard classification accuracy of 70% to 90% is desirable for our system to reliably interpret hand movements. If the model’s accuracy falls below 70%, the device’s will likely be ineffective and perform random movements. In researching this risk, our team has identified multiple datasets with over 180 acquisitions specifically for this application. This coupled with analysis of similar models shows that attaining this accuracy is possible.

The second risk is the size and computational demand of the machine learning model. Deploying an ML model on a resource-constrained microcontroller presents significant challenges, particularly regarding memory usage and processing speed. The STM32 microcontroller has limited RAM and flash memory, and balancing these constraints with the accuracy and complexity of the model is a delicate process. Analysis of existing literature and projects show EMG classification modules operating on similar devices.

The third risk pertains to digital signal processing (DSP) algorithms. The ADS1299 AFE module outputs raw sEMG data, which must be processed before feature extraction and classification. This risk arises from the need to align the processed data with the characteristics of the training dataset used for the ML model. Any discrepancies between the processed signals and the training data could result in inaccurate feature extraction, leading to poor classification performance.

The fourth identified risk is the configuration of the AFE module. The module’s numerous configuration options, including amplifier gain and sampling frequency, require careful tuning to ensure clean and reliable signal acquisition. Improper settings can result in noisy, distorted, or clipped signals, rendering them unsuitable for classification. This risk is particularly significant because the quality of the input signals directly impacts the performance of the ML model. Determining the correct combination of configuration parameters is a complex process that requires iterative testing and validation.

The fifth risk involves the complexity of PCB design. The system’s PCB must integrate three major ICs, including the AFE module, microcontroller, and power management circuit, with a combined chip area of 564mm² and a total of 236 pins. The high density of components and connections increases the likelihood of errors during the design and fabrication process. Additionally, ensuring proper signal routing and minimizing electromagnetic interference are critical to maintaining system performance. This risk is intensified by the limited space available on the PCB and the need for efficient layout strategies.

The final risk involves system integration, which encompasses the communication and functionality of the three major components: the power management circuit, the AFE module, and the microcontroller. Each of these subsystems operates with distinct control and communication protocols, making their integration a complex and potentially error-prone process. Since all subsystems are interdependent, any failure in communication between components, such as mismatched protocols or signal inconsistencies, would compromise the entire system. This makes system integration a critical risk to address, as successful communication and interoperability are essential for the device’s overall functionality and reliability.

These risks collectively represent the challenges to the success of our sEMG sensor system. While all five risks are significant, the machine learning classification accuracy, AFE configuration, and PCB design were identified as the most critical during the risk management meeting and prioritized for initial mitigation activities. A detailed table summarizing the risks and a risk cube visualizing their probability and impact is included below in Table 1 and Figure 6 to illustrate our considerations.

Table 1: Identified Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Risk | Probability | Impact | Mitigation |
| 1 | ML Classification Accuracy | 3 | 5 | Testing will take place with public data and AFE acquired data. |
| 2 | ML Computational Demand | 2 | 3 | ML model will be adjusted incrementally beginning with prototype phase 1. |
| 3 | DSP Algorithms | 2 | 4 | Select algorithms being prototyped in C. Existing literature for each DSP exists. |
| 4 | AFE Configuration | 2 | 5 | Signal acquisition testing began early with this prototype activity. |
| 5 | PCB Design Complexity | 3 | 4 | Simplified schematic, routing validation, and DRC. |
| 6 | System Integration | 3 | 5 | Modules will be tested individually with debugging tools to ensure proper technical characteristics before integration. |

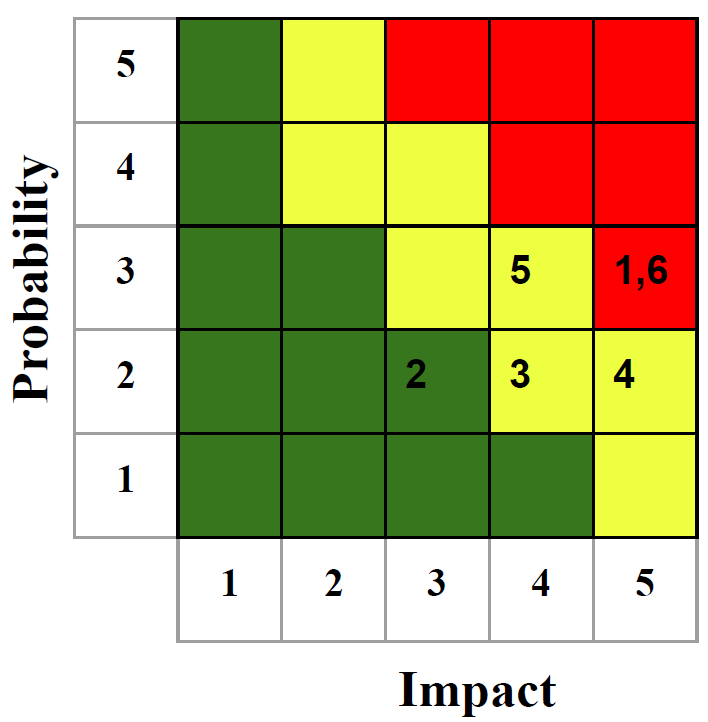


Figure 6: Risk Cube

# Prototype Activity

The following risks were selected for mitigation during the prototype activity due to their high impact on the project. Risk 1: ML classification accuracy, which has an impact rating of 5 because it directly determines the devices functionality. Risk 2: ML computational demand, which is a stretch goal that is aligned with addressing Risk 1 as the prototype tasks overlap. Risk 4: AFE configuration, which has an impact rating of 5 due to it being essential for ensuring clean and reliable signal acquisition. These risks, each scoring 4-5 out of 5 for impact, represent the most significant challenges to the project and were prioritized.

Our prototype activity was structured around two key tasks to mitigate these risks: determining the AFE configuration to address Risk 4 and developing an initial ML model to mitigate Risk 1, with a stretch goal of Risk 2. These tasks were chosen to account for the complexity of the project and to efficiently distribute work, allowing the entire team to make meaningful progress simultaneously across our high-impact areas. We aimed to ensure that critical system components were validated early in our development process, reducing overall project risk and uncertainty. Each task in our prototype activity will be discussed in detail below.

## AFE Configuration Risk Mitigation

**Selected Risk and Rationale**

The first risk we addressed was the configuration of the programmable gain amplifier (PGA) in the ADS1299 Analog Front-End (AFE) module (Risk #4). This risk was chosen since it’s critical to acquire clean and reliable sEMG signals which directly affect the accuracy of our machine learning model’s classification. The wrong gain settings could lead to noise amplification or signal clipping which would lead to improper signal classification.

**Prototyping Strategy and Objectives**

The main goal was to determine the optimal PGA gain setting that balanced signal acquisition sensitivity and noise rejection. The first tests were to analyze signal quality across the full range of PGA gain values that the ADS1299 supports using one hand configuration as a baseline for initial testing. After that, we validated the chosen gain with 17 hand movements across the four channels to confirm consistency in signal acquisition. These signal metrics were also compared to the NinaPro data set to act as a benchmark for signal characteristics. Below we list the steps taken in this prototyping.

1. Testing all supported PGA gain values (1, 2, 4, 6, 8, 12, 24) using one baseline hand configuration
2. Selecting the gain setting that gave a balance between signal clarity within the AFE’s range and noise rejection
3. Utilizing the chosen gain setting in signal acquisition across 17 hand configurations to ensure validity

A secondary goal of establishing communication between the ADS and microcontroller via the SPI protocol lines was also tackled. Establishing communication is necessary to address Risk 4 as the configuration is controlled with messages from the microcontroller.

**Risk Mitigation vs Elevation**

**Mitigating Results:** Risk mitigating results would consist of clean and consistent signals that closely align with our benchmark NinaPro data set within a range of +/- 10mV. These signals can be visually inspected to ensure no observable noise (such as 60Hz from wall power) and no apparent clipping. In addition, sending the control words to the ADS1299 should prove to be successful and verified on a logic analyzer for mitigating the risk.

**Elevating Results:** Significant deviations from our benchmark NinaPro data set or unstable signals across hand configurations would have elevated the risk significantly especially in probability, requiring further tuning or preprocessing. Observed clipping of signals or excess noise would prove to elevate the risk associated with the AFE configuration. If the control words sent from the STM32 are not registered on the ADS1299 this would also prove to elevate the risk.

**Test Setup and Procedures**

1. **Equipment:** ADS1299 Evaluation Board, 9 electrode leads, SPI connection for data transfer, and power cord for the ADS1299, Wet Electrodes.
2. **Signal Input:** Wet electrodes placed around the forearm to capture signals from four channels with a grounding point at the elbow.
3. **Procedure:**

* Signal metrics were recorded at least 3 times for each PGA gain setting using a single baseline hand configuration, exercise 1. These metrics were Vrms, Mean V, Vpp, Vmax, and Vmin.
* The PGA gain of 12 was selected as optimal. Although 24 was promising in signal acquisition, if there were signal spikes caused by muscle cramps or rapid movements, 12 was still able to record a clean signal and minimize excess noise or spikes that 24 had picked up.
* Validation involved testing the gain of 12 across all 17 hand configuration exercises for a broad applicability. These signals were analyzed for amplitude stability and noise levels as well as comparing them to averages from NinaPro dataset to ensure it aligned with those benchmarks.

**Results and Data Presentation**

Figure 7: Comparison of Vmin and Vmax with our configuration

Figure 8: Comparison of Vpp with our configuration

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exercises | Vavg 1 | Vrms 1 | Vpp 1 | Vavg 2 | Vrms 2 | Vpp 2 | Vavg 3 | Vrms 3 | Vpp 3 | Vavg 4 | Vrms 4 | Vpp 4 |
| 2 | -0.06 | 0.25 | 1.34 | -0.59 | 0.73 | 1.25 | 0.91 | 1.22 | 2.40 | 0.43 | 0.59 | 1.49 |
| 3 | -0.05 | 249.84 | 1.26 | -0.70 | 0.86 | 1.45 | 1.05 | 1.39 | 2.81 | 0.50 | 0.68 | 1.59 |
| 4 | -0.22 | 0.33 | 1.13 | -0.48 | 0.59 | 1.07 | 1.08 | 1.43 | 2.84 | 0.51 | 0.69 | 1.15 |
| 5 | -0.70 | 0.71 | 1.02 | -1.72 | 1.72 | 0.29 | 3.68 | 3.70 | 1.79 | 2.57 | 2.49 | 1.21 |
| 6 | -0.70 | 0.83 | 1.84 | -0.95 | 1.09 | 1.68 | 1.65 | 1.98 | 3.32 | 1.17 | 1.39 | 2.47 |
| 7 | -0.33 | 0.41 | 0.90 | -0.81 | 0.92 | 1.39 | 1.60 | 1.91 | 3.13 | 0.97 | 1.15 | 2.13 |
| 8 | -0.33 | 0.41 | 1.31 | -0.81 | 0.89 | 1.38 | 1.80 | 2.07 | 3.36 | 1.10 | 1.26 | 2.21 |
| 9 | -0.16 | 0.35 | 1.37 | -0.85 | 1.05 | 1.78 | 1.09 | 1.44 | 3.15 | 0.56 | 0.77 | 1.79 |
| 10 | -0.11 | 0.33 | 1.20 | -0.63 | 0.79 | 1.50 | 0.86 | 1.19 | 2.81 | 0.24 | 0.48 | 1.64 |
| 11 | -0.53 | 0.73 | 1.52 | -1.22 | 1.26 | 1.03 | 3.20 | 3.68 | 5.49 | 3.00 | 3.29 | 3.80 |
| 12 | -0.20 | 0.27 | 0.68 | -0.73 | 0.93 | 1.56 | 1.09 | 1.50 | 2.86 | 0.62 | 0.86 | 1.63 |
| 13 | -0.79 | 0.83 | 0.84 | -1.49 | 1.50 | 0.64 | 2.66 | 2.71 | 1.99 | 2.35 | 2.37 | 1.37 |
| 14 | -0.92 | 0.98 | 2.28 | -1.74 | 1.75 | 0.73 | 3.21 | 3.22 | 1.64 | 2.11 | 2.11 | 0.65 |
| 15 | -0.84 | 0.87 | 0.80 | -1.76 | 1.77 | 0.62 | 2.86 | 2.86 | 0.47 | 2.48 | 2.48 | 0.36 |
| 16 | -0.84 | 0.87 | 0.74 | -1.95 | 1.95 | 0.51 | 3.33 | 3.34 | 1.27 | 2.65 | 2.65 | 0.39 |
| 17 | -1.56 | 1.59 | 1.43 | -1.64 | 1.64 | 0.29 | 3.48 | 3.49 | 0.95 | 2.63 | 2.64 | 0.72 |

Table 2: Voltage Features of Measured sEMG signals in mV

Table 3: Exercises Performed in Testing AFE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exercises | | | | | | | | |
| 1 | Thumb up |  | **7** | Pointing index |  | **13** | Wrist flexion |  |
| 2 | Extension of index and middle, flexion of others |  | **8** | Adduction of extended fingers |  | **14** | Wrist extension |  |
| 3 | Flexion of ring and little finger extension of others |  | **9** | Wrist supination (axis middle finger) |  | **15** | Wrist radial deviation |  |
| 4 | Thumb opposing base of little finger |  | **10** | Wrist pronation (axis middle finger) |  | **16** | Wrist ulnar deviation |  |
| 5 | Abduction of all fingers |  | **11** | Wrist supination (axis little finger) |  | **17** | Wrist extension |  |
| 6 | Fingers flexed together in fist |  | **12** | Wrist pronation (axis little finger) |  |  |  |  |

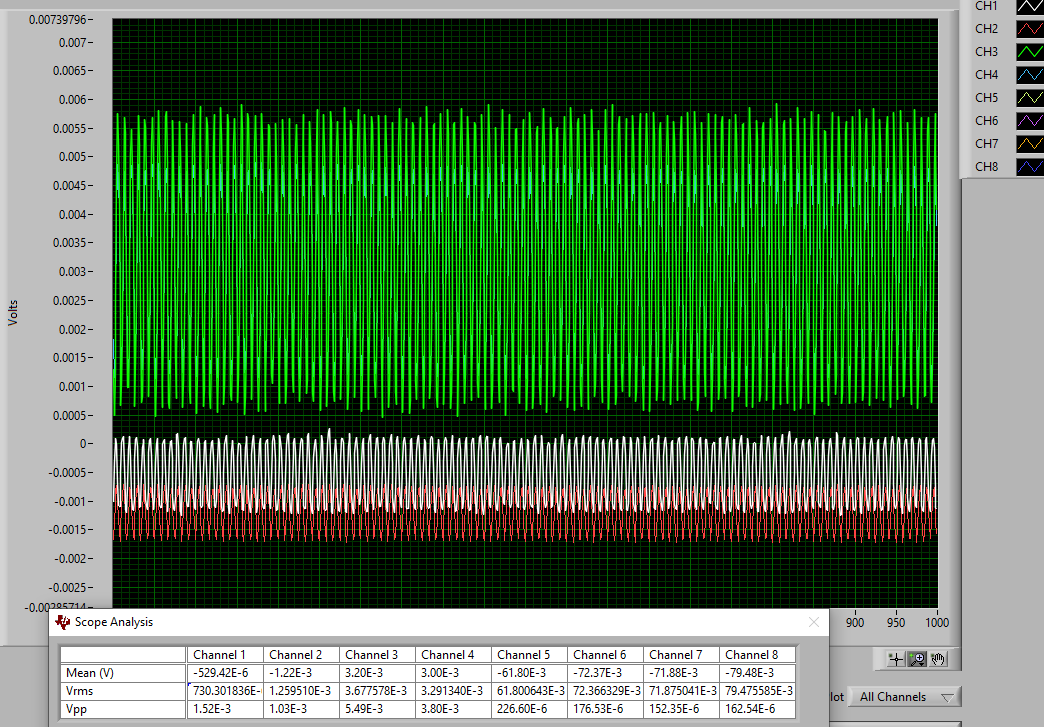


Figure 9: Waveform of Exercise 11 on Scope

(Figure 9 seen above was the waveform taken of Exercise 11 from Table 3 being performed and measured on the internal scope on the AFE Dev Kit)

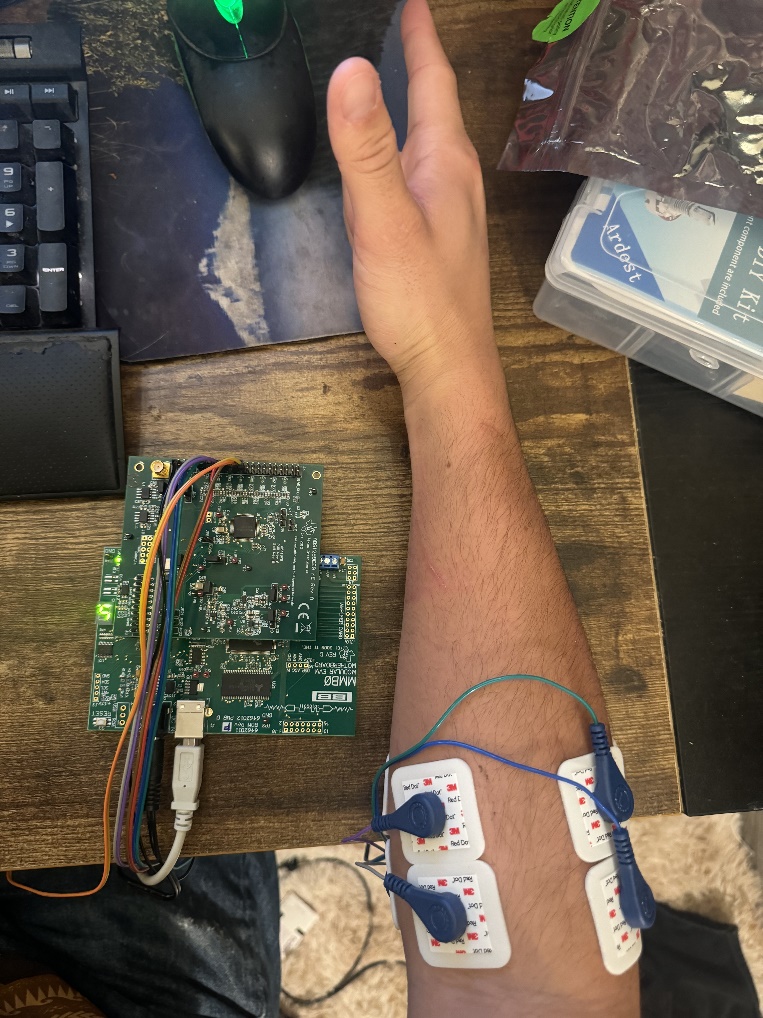


Figure 10: Setup for Performing Exercises 1-17

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.

`A screenshot of a computer

Description automatically generated

Figure 11: Scope AFE Communication with STM32

A computer circuit board with wires and a computer

Description automatically generated with medium confidence

Figure 12: Microcontroller and AFE Communication Setup

## Machine Learning Risk Mitigation

**Selected Risk and Rationale**

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

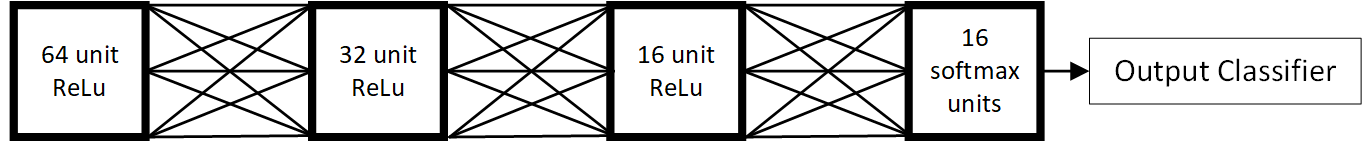


Figure 13: ML Model Layer Diagram

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

**Risk Mitigation vs Elevation**

**Mitigated Risks:** Mitigating the risks involved in this section would involve having a classification accuracy of at least 70% for an initial prototype baseline. This would show that for the selected data, the problem is resolvable, and the model can reasonably reach an optimal solution. In addition, this prototyping section needs to meet the physical constraints previously mentioned

**Elevated Risks:** In this section, there is a low probability of the actual elevation of risk. If average accuracy falls below 60%, that increases the probability of this risk occurring substantially. In addition, if the most basic models are not the size as calculated, and surpass the capacity of the device, that presents increased risk in the form of swamping the device so no other tasks can be performed during inference.

**Test Setup and Procedures**

The first consideration in building the model is seeing what exact functions and layers are supported in the TensorFlow Lite Micro library. We did not have reasonable time to write code for a hardware efficient custom function, so it was decided to use the preexisting libraries for this task. Essentially, the most basic of tasks are included such as 1D & 2D convolution, regular dense layers, overfitting functions, and basic tensor arithmetic functions.

The workflow for the actual test setup involved making algorithms to replicate the extraction of certain features. After this I can write a basic Python script to create the function of the steps listed in the prototyping strategy. This can be done in any IDE that allows compilation in Python. The true caveat here was working in conjunction with the team members working on the AFE configuration. This verifies that the training data is most closely matched to the data we receive from our device, and thus the device can make inferences as optimally as possible in its final iteration. The TensorFlow library offers functions that can evaluate your model with test data at the end of any training phase, which shows us the generalizable performance of the model most closely.

**Results and Data Presentation**

Following the above procedures, we were finally left with some preliminary results. Regarding risk 2, we can simply build and compile the model and view the actual file size. This is an accurate prediction of not only the size required on the flash memory, but due to the basic nature of the model, it is also an accurate prediction of the SRAM needed during inference. The size of the model was found to be 5,761 total parameters. Multiplying this by 4 bytes, we get 22.5KB, which fits well below the required threshold of 800KB (80% of total flash storage).

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.

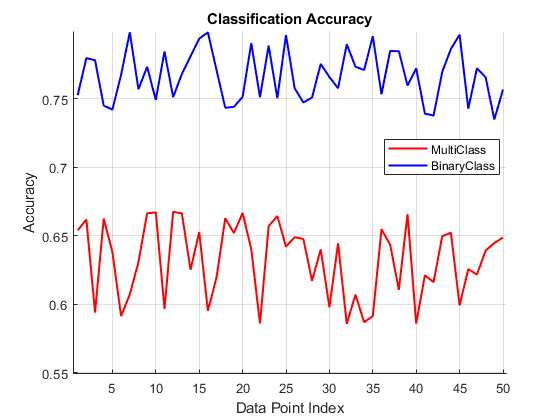


Figure 14: ML Classification Accuracy Graph

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 eight 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.

# Final Risk Analysis

After completing the prototype activity, the overall risk profile of the project has been updated to reflect the progress made in mitigating the key risks. These changes can be seen below Table 4. Figure 14 shows how our selected risks moved from their initial positions before the prototype tasks. The activity focused on addressing high-impact risks, including the machine learning classification accuracy (Risk 1), computational demand (Risk 2), and AFE configuration (Risk 4). The prototype results provided valuable insights into these risks, which enabled us to adjust our risk table and risk cube. Most risks had a significantly reduced probability while their impacts stayed the same. Risks 2 and 4 had their probability reduced to the lowest point, while Risk 1 remains partially unresolved and will require further attention as our project advances.

The probability of the AFE configuration risk was lowered to 1 due to the successful completion of the prototype task. During this task, the mitigating results metrics were established and thus the probability was lowered. Through extensive testing, we successfully identified and implemented the optimal gain and sampling settings for the AFE, ensuring reliable signal acquisition. Control words were successfully sent as well as receiving data from the ADS using our microcontroller, ensuring proper communication. The impact of this risk was not lowered, as our initial analysis correctly determined that any failure in the AFE configuration would still have a substantial effect on the overall system.

Regarding the risks involved in the machine learning model, the size of the model (risk 2) was of no serious consideration. This was accurately determined prior to prototyped designs, as therefore the risk probability has not changed from its location in the risk cube. The classification accuracy (risk 1), however, has changed. Despite the poor performance in the classification accuracy, it is already known what the prime suspect in the poor performance is. Thus, the probability of this risk occurring is lowered to 2. In both risks, the impact remains the same, as the whole goal of the project is to use a machine learning model to predict gestures in the hand.

To manage the remaining risks as our project continues, we will utilize specific strategies. For machine learning accuracy, additional training and model optimization will be prioritized as the project continues. To address the risks associated with the DSP algorithms, prototyping will begin in the upcoming weeks to design these algorithms in C on our microcontroller. Once the DSP algorithms are designed for the microcontroller, testing will begin taking place by feeding the features into the ML model within the microcontroller. To address the PCB design complexity, we will begin thorough design reviews and DRC runs to ensure manufacturability. As development continues, testing and debugging on each subsystem will happen individually to ensure proper technical characteristics. This enables us to work quickly and ensure that we address the risk of system integration. Overall, these strategies will help us manage the remaining risks and continue our prototyping in future iterations.

Table 4: Updated Risk Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Risk | Probability | Impact | Mitigation |
| 1 | ML Classification Accuracy | 2 | 5 | Testing will take place with public data and AFE acquired data. |
| 2 | ML Computational Demand | 1 | 3 | ML model will be adjusted incrementally beginning with prototype phase 1. |
| 3 | DSP Algorithms | 2 | 4 | Select algorithms being prototyped in C. Existing literature for each DSP processes exists. |
| 4 | AFE Configuration | 1 | 4 | Signal acquisition testing began early with this prototype activity. |
| 5 | PCB Design Complexity | 3 | 4 | Simplified schematic, routing validation, and DRC. |
| 6 | System Integration | 3 | 5 | Modules will be tested individually with debugging tools to ensure proper technical characteristics before integration. |

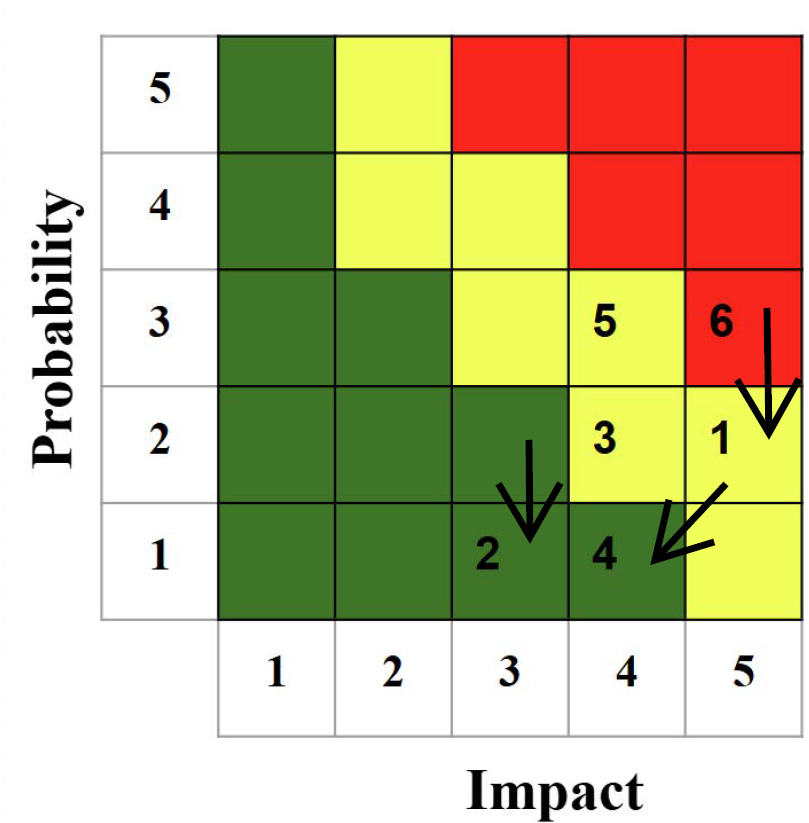


Figure 15: Updated Risk Chart