



SAN DIEGO STATE UNIVERSITY

College of Engineering, Department of Electrical and Computer Engineering

EE/COMPE 491W Senior Design

Integration Readiness 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
System Description	5
Hardware Readiness.....	11
Power Management Board	11
Analog Front End Board	22
Microcontroller Board.....	31
Wiring Diagram.....	39
Software and Firmware Readiness.....	40
Inter-Board Communication	41
Machine Learning	43
Feature Extraction	46
Integration Plan.....	49

Table of Figures

FIGURE 1: MACHINE LEARNING PROCESS.....	5
FIGURE 2: TOP-DOWN MECHANICAL DRAWING	6
FIGURE 3: SIDE VIEW MECHANICAL DRAWING	7
FIGURE 4: USE CASE FLOW DIAGRAM.....	8
FIGURE 5: DEVICE BLOCK DIAGRAM.....	9
FIGURE 6: SIGNAL FLOW DIAGRAM	10
FIGURE 7: POWER MANAGEMENT PCB SIMPLE DIAGRAM.....	12
FIGURE 8: POWER MANAGEMENT PCB SCHEMATIC	13
FIGURE 9: POWER MANAGEMENT PCB CHARGE PUMP	14
FIGURE 10: POWER MANAGEMENT PCB CIRCUIT SIDE VIEW	15
FIGURE 11: POWER MANAGEMENT PCB COMPONENT SIDE VIEW	16
FIGURE 12: POWER MANAGEMENT PCB 3D RENDERING	16
FIGURE 13: POWER MANAGEMENT PCB FINAL ASSEMBLY.....	17
FIGURE 14: POWER MANAGEMENT PCB TEST SETUP	18
FIGURE 15: BATTERY DISCHARGE SETUP	18
FIGURE 16: DIGITAL DOMAIN VOLTAGE RIPPLE	20
FIGURE 17: POWER -2.5V BRING UP TIME.....	21
FIGURE 18: AFE PCB SIMPLE BLOCK DIAGRAM	22
FIGURE 19: AFE PCB SCHEMATIC POWER & I/O	24
FIGURE 20: AFE PCB SCHEMATIC OUTPUT ITEMS	24
FIGURE 21: AFE PCB SCHEMATIC DECOUPLING & PASSIVES.....	25
FIGURE 22: AFE PCB CIRCUIT SIDE VIEW	26
FIGURE 23: AFE PCB COMPONENT SIDE VIEW	27

FIGURE 24: AFE PCB TEST SETUP	29
FIGURE 25: AFE PCB FINAL ASSEMBLY.....	30
FIGURE 26: MICROCONTROLLER PCB SCHEMATIC	31
FIGURE 27: MICROCONTROLLER PCB SIMPLE BLOCK DIAGRAM	32
FIGURE 28: MICROCONTROLLER PCB CIRCUIT SIDE VIEW	33
FIGURE 29: MICROCONTROLLER PCB COMPONENT SIDE VIEW	34
FIGURE 30: MICROCONTROLLER PCB 3D RENDERING	35
FIGURE 31: MICROCONTROLLER PCB FINAL ASSEMBLY.....	36
FIGURE 32: MICROCONTROLLER PCB FLASH TEST	38
FIGURE 33: INTEGRATION WIRING DIAGRAM	39
FIGURE 34: SOFTWARE COMPONENT OVERVIEW	41
FIGURE 35: SPI COMMUNICATION WITH ADS1299 SCOPED.....	42
FIGURE 36: REVISED LIST OF ML GESTURES.....	43
FIGURE 37: ML NETWORK ANALYSIS FOR STM32	44
FIGURE 38: ML MODEL VALIDATION ON DEVICE.....	45
FIGURE 39: SUCCESSFUL FLASHING INDICATOR	45
FIGURE 40: OUTPUT WINDOW OF MATLAB SERIAL	47
FIGURE 41: QUANTIZATION ERROR FOR FEATURES	48
FIGURE 42: INTEGRATION FLOW CHART.....	50
FIGURE 43: TOTAL INTEGRATION TEST/SUCCESS CRITERIA.....	53

Table of Tables

TABLE 1: POWER MANAGEMENT PCB TESTING RESULTS	19
TABLE 2: INTER-BOARD COMMUNICATION TESTS.....	41
TABLE 3: LIST OF FEATURES FOR EXTRACTION	46
TABLE 4: INTEGRATION PLAN PROCEDURES.....	51

Executive Summary

This report provides an overview of the design, manufacturing, testing, and validation of all subsystem printed circuit boards (PCBs) and firmware components in preparation for system integration. Our system consists of three primary PCBs: the power management board, the analog front-end (AFE) board, and the microcontroller (uC) board. Each subsystem has undergone individual testing to verify functionality and performance, ensuring readiness for full integration.

The power management PCB was tested to confirm proper voltage regulation, stable power delivery, and transient response. The analog front-end PCB was validated to ensure proof of operation and communication, while the microcontroller PCB was tested for programming functionality and being operational. Additionally, firmware components, including the machine learning (ML) model and communication protocols, were evaluated. The ML model achieved an 85% classification accuracy and was validated on an evaluation kit using sample data. Communication protocols were tested and confirmed to function correctly using evaluation hardware. Feature extraction algorithms were also tested and compared with the MATLAB implementation to

With individual subsystem validation complete, our next step is system integration. This process will involve combining the power management, AFE, and microcontroller PCBs, running firmware on the complete system, and validating performance under real-world operating conditions. The results of this integration phase will confirm overall system functionality and guide any final refinements.

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.

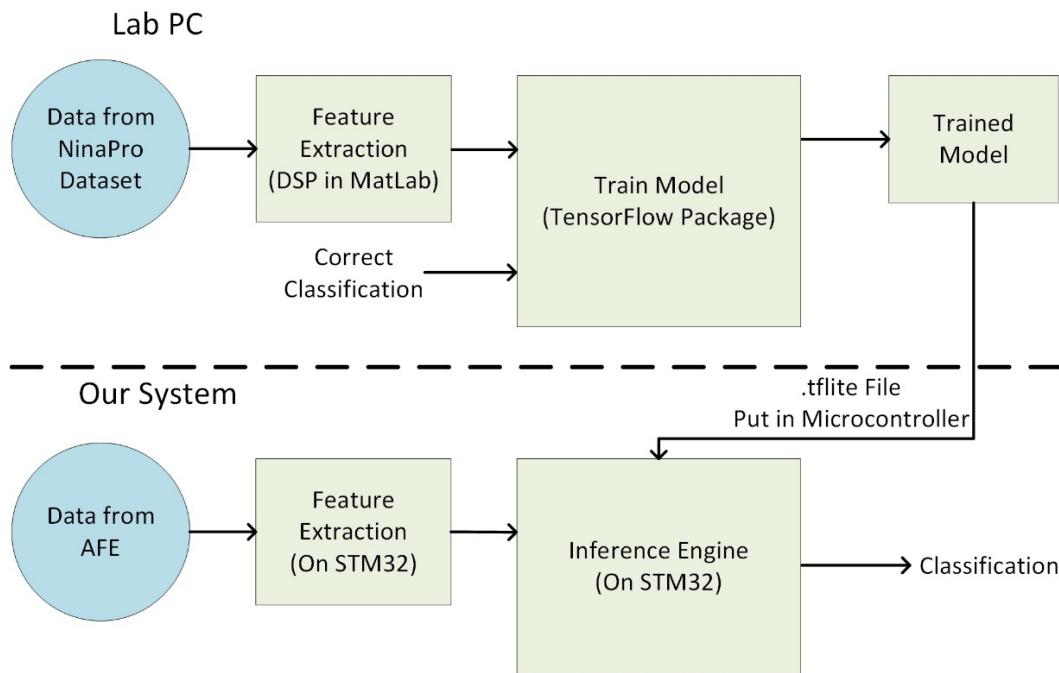


Figure 1: Machine Learning Process

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.

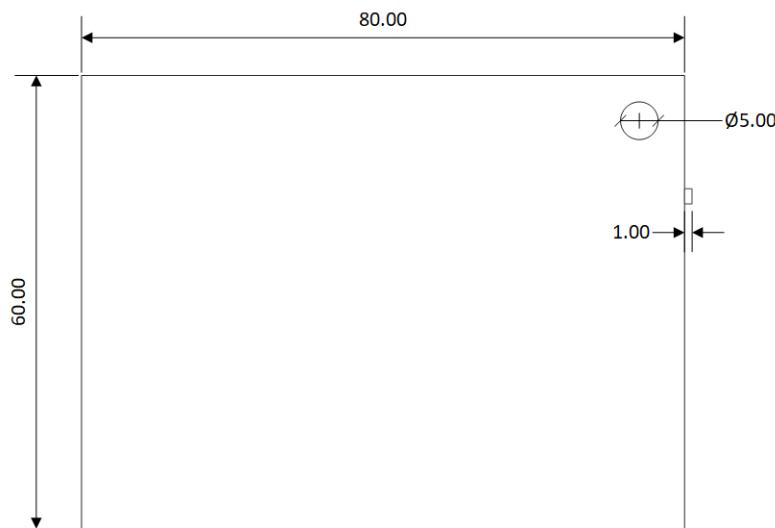


Figure 2: Top-Down Mechanical Drawing

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.

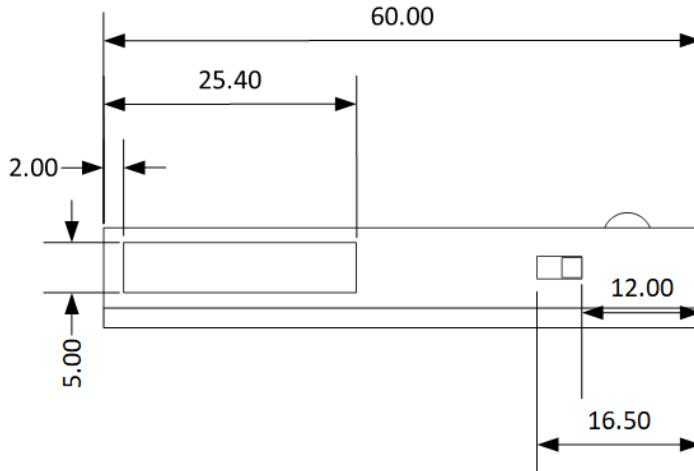


Figure 3: Side View Mechanical Drawing

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.

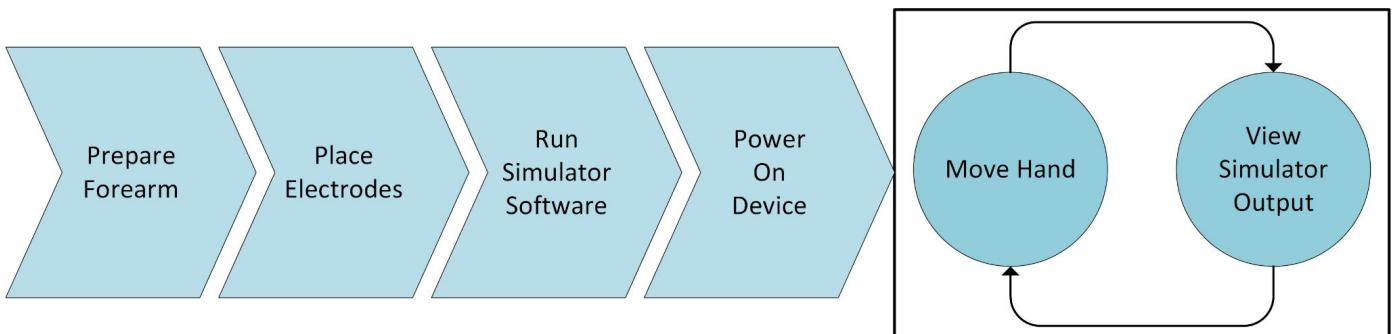


Figure 4: Use Case Flow Diagram

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.

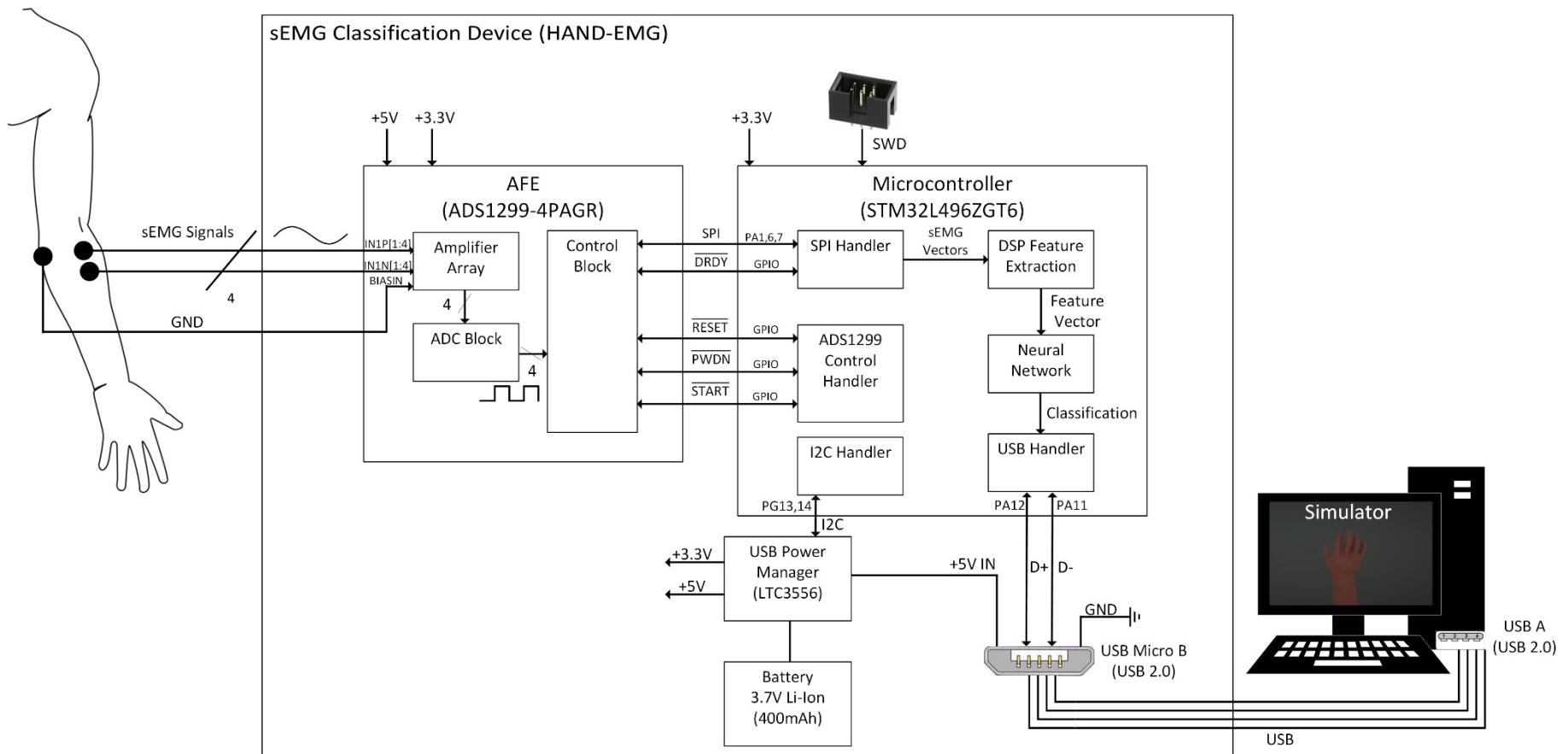


Figure 5: Device Block Diagram

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

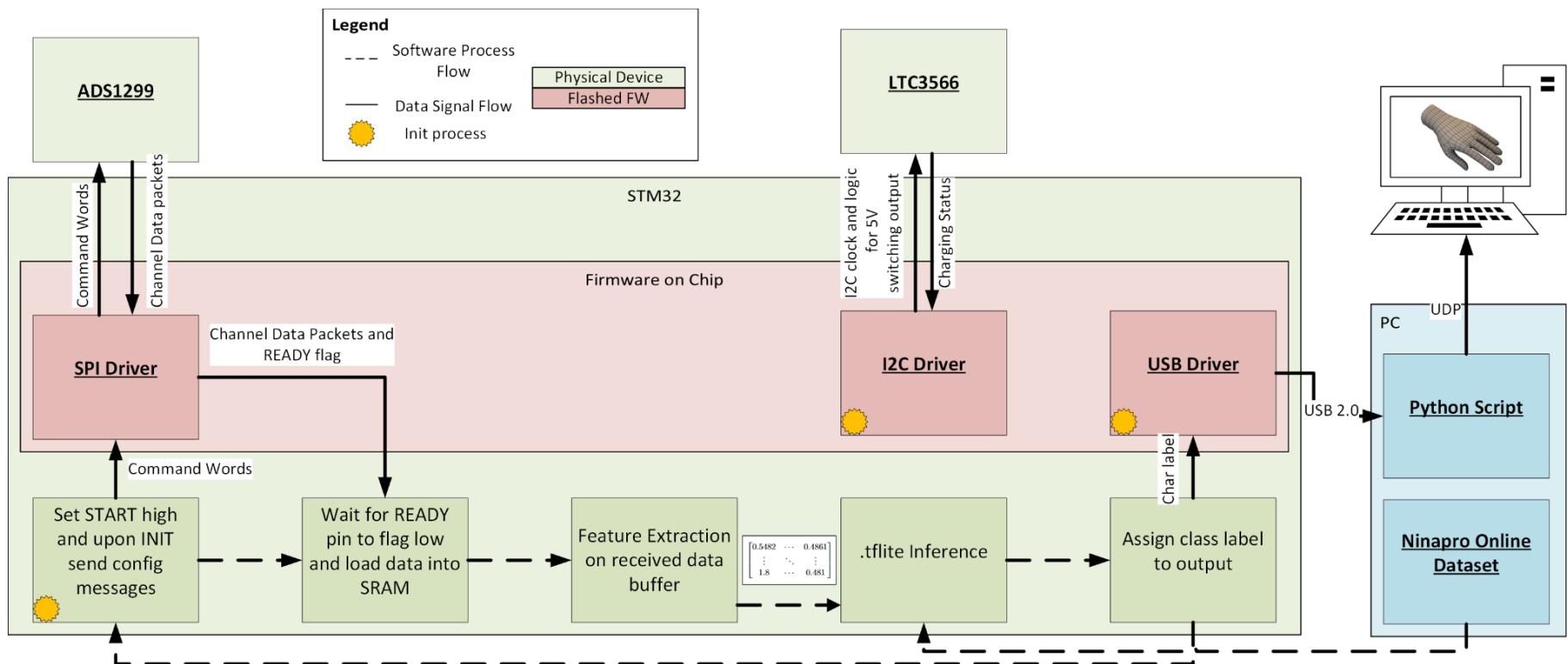


Figure 6: Signal Flow Diagram

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.

Hardware Readiness

As stated in the system description, our device contains three major subsystems that require testing and evaluation for hardware readiness. The systems are power management, analog front end, and microcontroller. Hardware readiness testing ensures that each subsystem functions correctly and meets our design specifications before full system integration. The power management system was evaluated for proper voltage regulation, stable power delivery, and the ability to control power rails via firmware. The analog front-end (AFE) was tested for proof of operation and communication with the microcontroller. The microcontroller subsystem was validated for programmability, stable operation, and successful execution of basic firmware components. The results of these tests confirm that each hardware subsystem is performing as expected and is ready for integration into the complete system.

Power Management Board

Introduction/Design Details

As the heart of our project, our power management board is critical for all systems in our device to function. Our power management system is based on the LTC3556, a highly integrated power management IC that provides multiple regulated outputs. The board generates a 3.3V rail to supply digital power to both the microcontroller and the digital section of the analog front end (AFE). To power the analog domain of the AFE, the design includes symmetric $\pm 2.5\text{V}$ rails, ensuring proper operation of the bipolar analog circuitry. The LTC3556 was selected for its high efficiency, compact form factor, and integrated battery charging capabilities, which simplify system design. The regulation performance and transient response of the LTC3556 ensures stable power delivery to sensitive analog and digital components. Proper layout techniques, including careful placement of bypass capacitors and ground plane considerations, were implemented to minimize noise and maintain signal integrity across the system.

Since the LTC3556 is an integrated circuit (with a QFN package) our team designed a custom PCB around this chip. Below in Figure 7 the simplified block diagram of our power management board is seen. The detailed implementation will be seen in later images such as the PCB schematic.

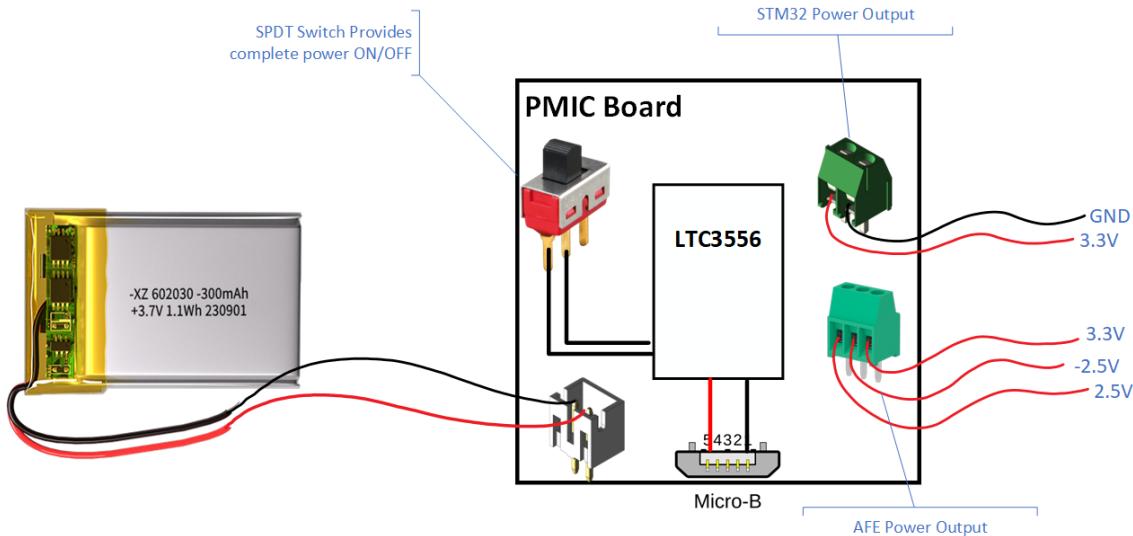


Figure 7: Power Management PCB Simple Diagram

Figure 7 shows a simple block diagram of the power management PCB without any passive components on it. The power management PCB has 5 major parts which can be seen in this block diagram. The screw terminals to hook up power to the other boards, the power switch to control power ON/OFF, the micro USB port for charging the battery, and the battery to power the device.

The following images show our power management PCB's schematics, layout, and 3D model.

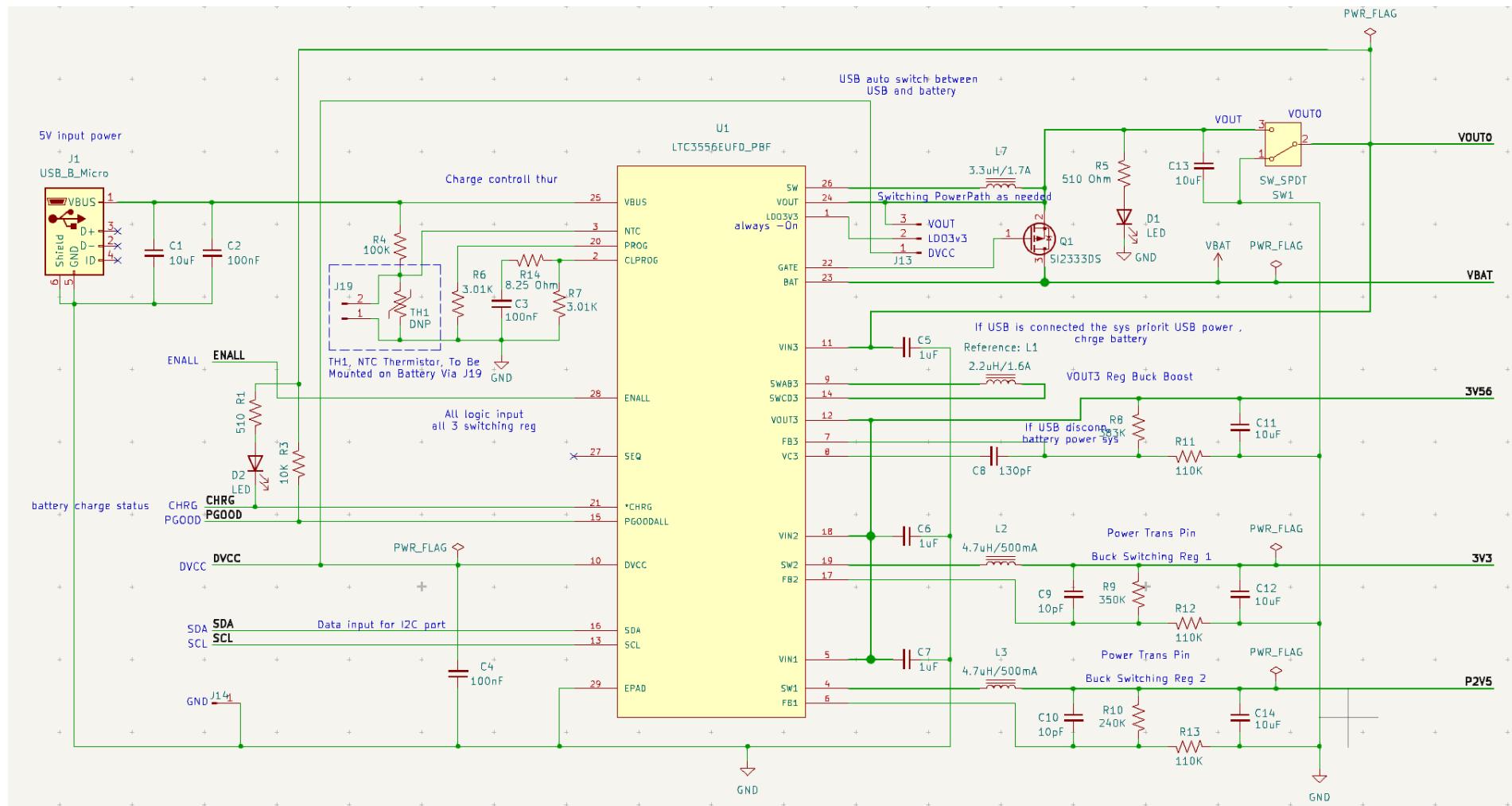


Figure 8: Power Management PCB Schematic

Figure 8 shows the schematic for the power management PCB. In the center of the schematic is the LTC3556, the main power regulator. The voltage outputs levels are controlled by a resistor voltage divider along their SW and FB pins. This schematic also the connection to the micro USB.

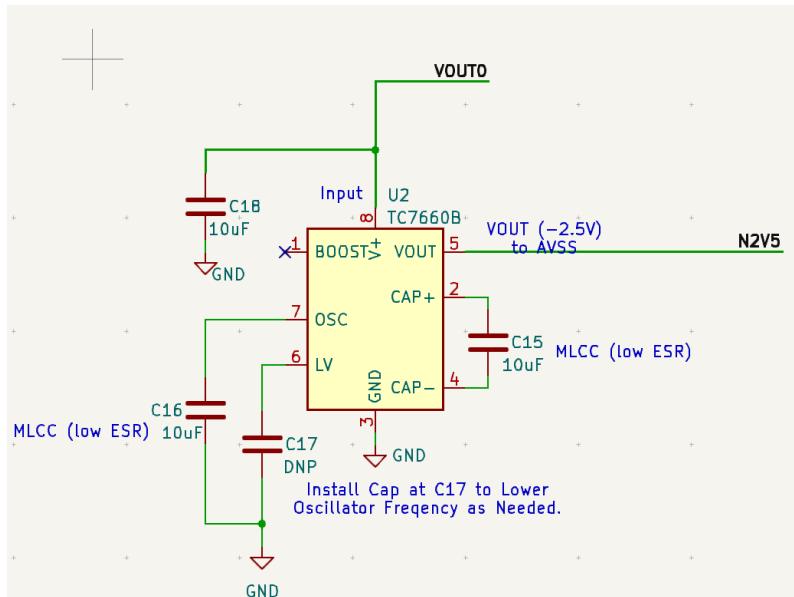


Figure 9: Power Management PCB Charge Pump

Figure 9 shows the charge pump that is responsible for generating the -2.5V output. The charge pump works by oscillating the voltage across C15.

The images above in Figure 8 and Figure 9 show the schematics for our power management PCB. The schematic in Figure 8 shows the main LTC3556 operation. On the right side of it the main DCDC regulator outputs are shown. These outputs are powered by VOUT0 which is controlled by the switch. VOUT0 is provided depending on the connected devices, the MOSFET Q1 will force VOUT0 to be powered by the 5V coming from the micro USB cable if it is present, otherwise VOUT0 will be powered by the battery. The schematic in Figure 9 shows the charge pump which is responsible for generating -2.5V.

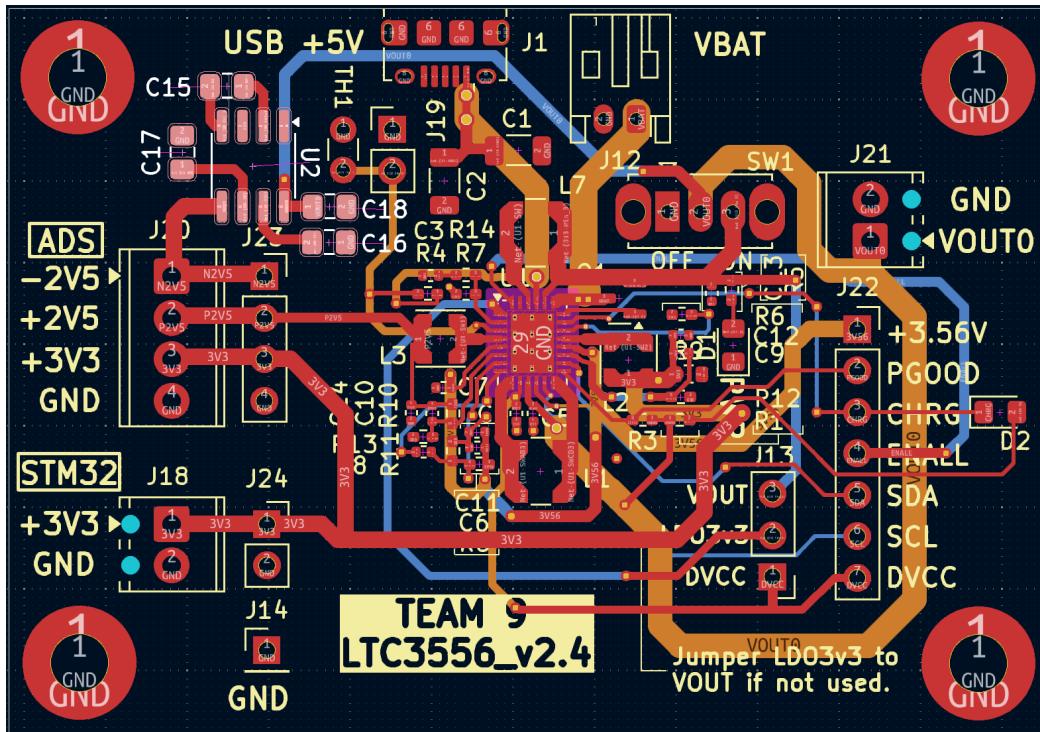


Figure 10: Power Management PCB Circuit Side View

Figure 10 shows the circuit side view for our power management PCB. This PCB is 4 layers. Note that the red traces and pads are the top layer copper, the orange traces are the second middle layer copper, and the blue traces are the bottom layer copper. The entire first middle layer (In1.Cu) is a ground plane.

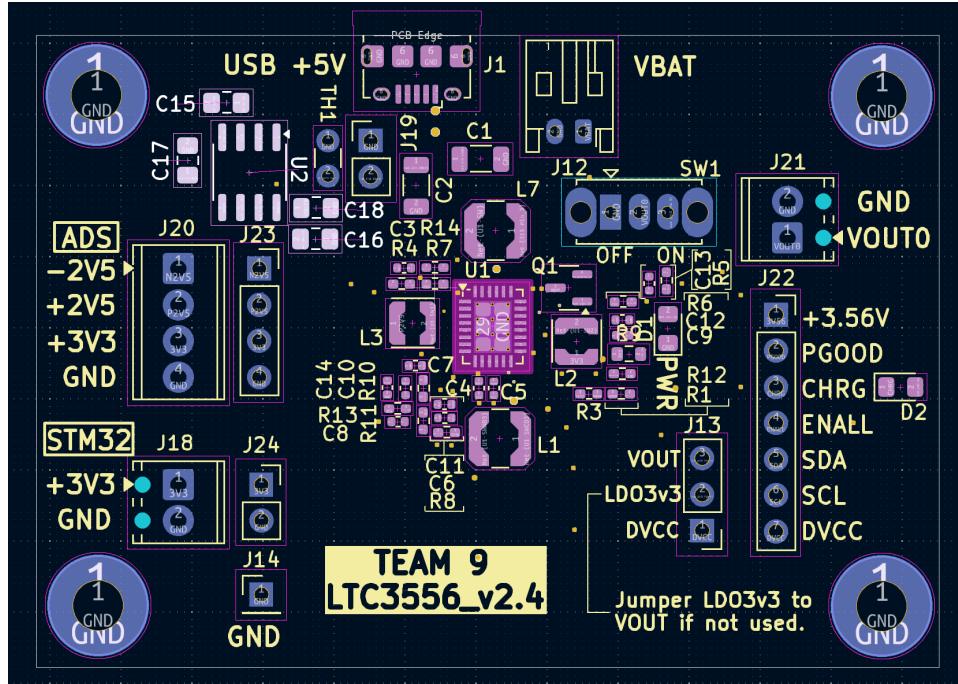


Figure 11: Power Management PCB Component Side View

Figure 11 shows the component side view of our power management PCB. Note the compact design and use of small passive components. This was necessary to place the decoupling components as close to the PMIC as possible to reduce noise.

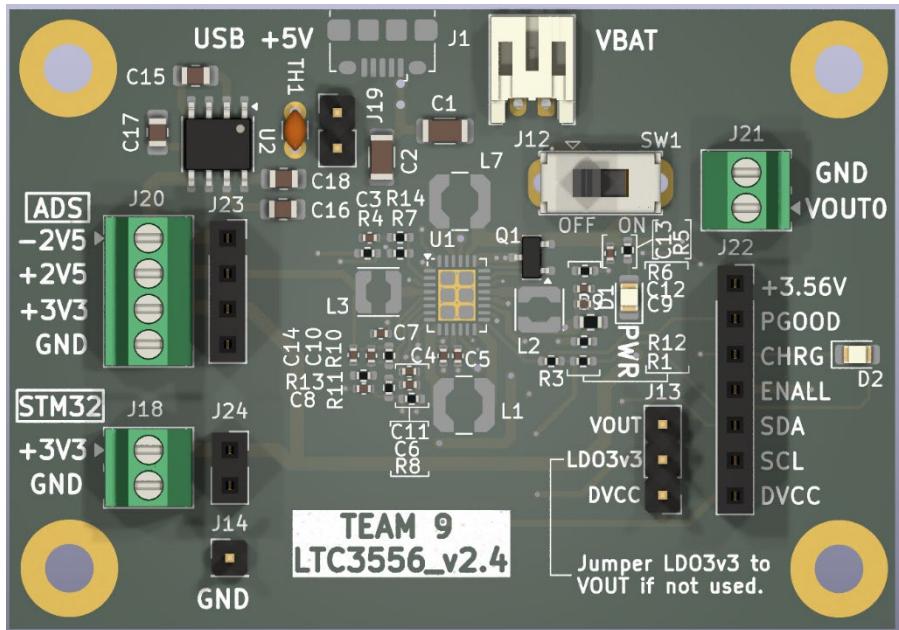


Figure 12: Power Management PCB 3D Rendering

Figure 12 is a visual of the assembled LTC3556 power management system with component placement and connectors. Note that there are a few missing 3D models.

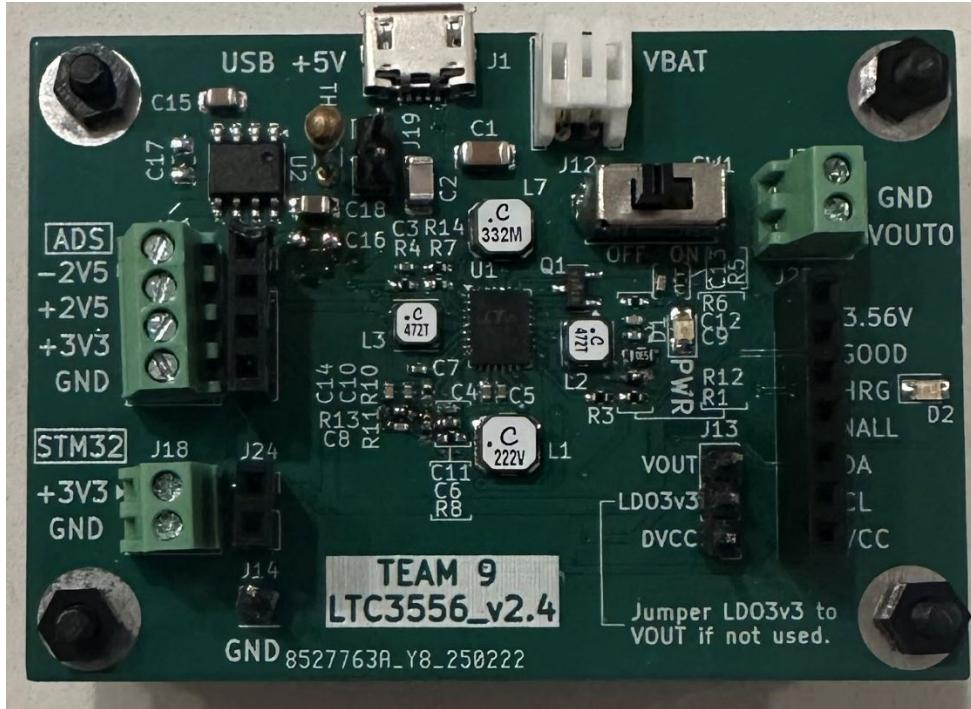


Figure 13: Power Management PCB Final Assembly

Figure 13 shows the final, fully assembled power management PCB. This PCB was assembled entirely in the senior design lab using the reflow oven. After thorough debugging, we determined that C16 had to be removed and multiple resistors had to be “teepeed” which can be seen on the board.

Final Assembled PCB

All hardware components have been developed or procured and passed basic operational tests. The LTC3556 PMIC and TC7660EOA charge pumps were tested under different load conditions. The PCB, ordered from JLCPCB, was inspected and assembled successfully. All components were soldered in the senior design lab using an oven reflow process for fine-pitch components such as 0402 resistors and capacitors. The final assembled PCB (post electrical debugging) can be seen above in Figure 13.

Testing and Analysis

To validate the performance of our power management board, several tests were conducted to ensure proper operation and reliability. A battery discharge and charge test was performed by connecting a battery across a 33Ω power resistor resulting in an approximate 100mA current draw for one hour. The battery voltage was monitored until a noticeable drop was observed, after which it was recharged using our PMIC board for an additional hour. The battery voltage successfully returned to its maximum level, confirming proper charging functionality.

Additionally, the regulated voltage outputs were measured using a digital multimeter (DMM), verifying that the board provided stable 3.3V, +2.5V, and -2.5V rails. The battery charging voltage was also tested with a DMM, showing that the PMIC supplied a steady 3.9V to charge the

battery. Lastly, a voltage ripple and bring up test was conducted using an oscilloscope, with captured waveforms confirming stable power delivery and minimal noise across the output rails. The test setup for the battery discharging can be seen below in Figure 15. The test setup for the remaining DMM and oscilloscope tests can be seen in Figure 14.

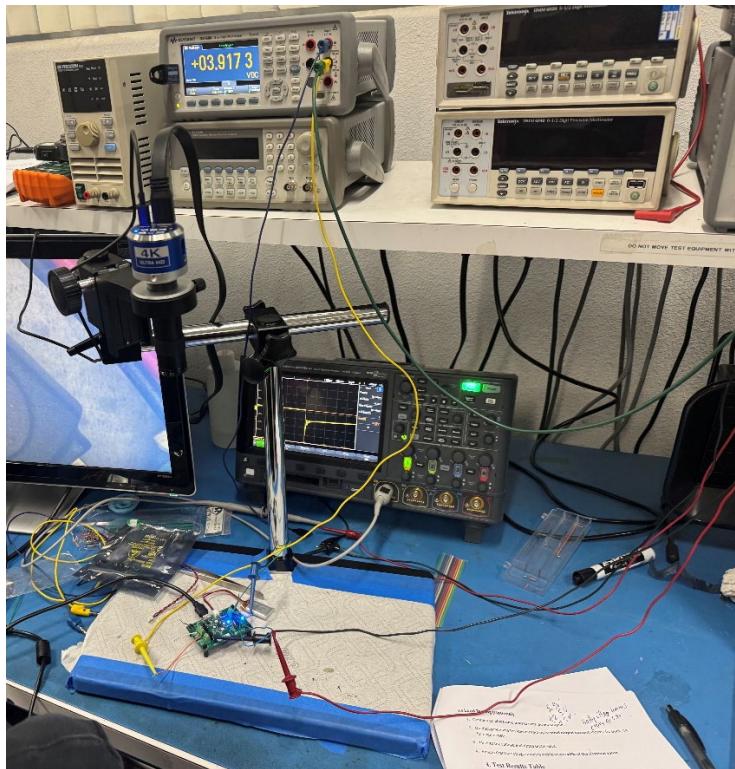


Figure 14: Power Management PCB Test Setup

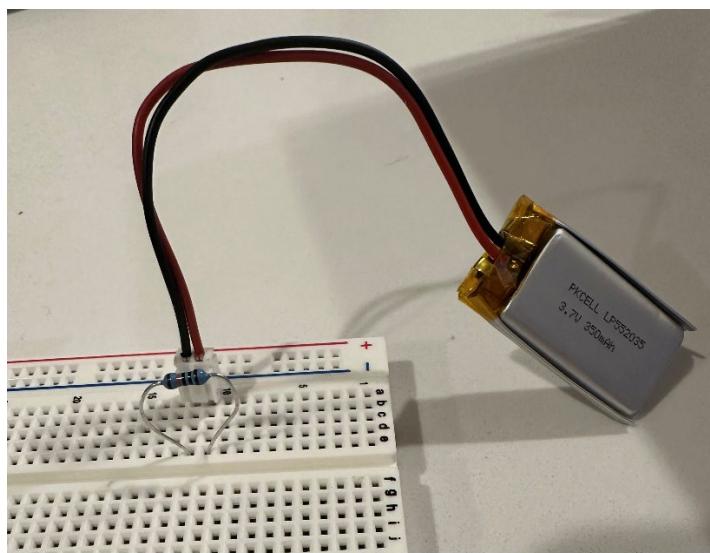


Figure 15: Battery Discharge Setup

Table 1: Power Management PCB Testing Results

Measurement	Voltage
Battery After Discharge	3.64V
Battery After Charging	3.76V
Voltage output on 3.3V line	3.6V
Voltage output on 2.5V line	2.54V
Voltage output on -2.5V line	-2.47V

The measured values confirm the proper operation of the power management board by demonstrating expected voltage regulation and battery charging behavior. After the discharge test, the battery voltage dropped to 3.64V, indicating a depletion consistent with the applied load. Following the charging cycle, the battery voltage increased to 3.76V, confirming that the PMIC is supplying charge to the battery. The 3.3V rail measured at 3.6V, which is within an acceptable range considering load conditions and potential measurement tolerances. The +2.5V and -2.5V outputs measured at 2.54V and -2.47V, respectively, showing that the PMIC is generating the expected bipolar supply for the analog front end with minimal deviation. These values confirm that the power management system is regulating voltages correctly and effectively charging the battery.

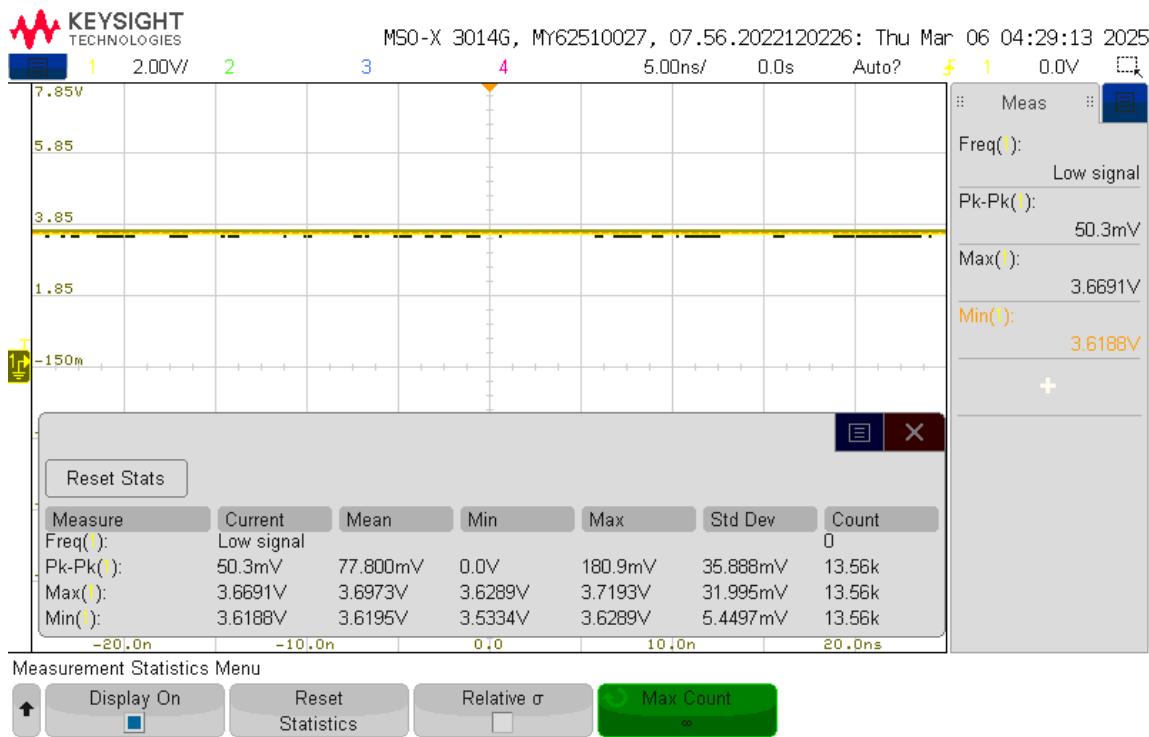


Figure 16: Digital Domain Voltage Ripple

Figure 16 shows the oscilloscope measurement of our power management PCB's 3.3V line. It was critical that this line's voltage ripple was analyzed to determine if there would be improper operation on the microcontroller and analog front end.

The oscilloscope measurements seen in Figure 16 above indicate that the voltage on the 3.3V line fluctuates between a minimum of 3.54V and a maximum of 3.71V, resulting in a peak-to-peak ripple of 0.17V. This level of ripple is within an acceptable range for our digital components without affecting functionality. The power supply specifications for our microcontroller and analog front end accommodates for such small deviations. The high-frequency nature of the voltage ripple causes it to be less visible on the oscilloscope waveform due to the instrument's sampling rate and display persistence, which can average out rapid fluctuations, making them less apparent in the captured trace.



Figure 17: Power -2.5V Bring Up Time

Figure 17 above shows that measurement of the -2.5V power lines bring up time. It's important to note that in the oscilloscope image, the time per div is 10 μ s.

Figure 17 above captures the bring up sequence of -2.5V power rail, measured using a single trigger on the oscilloscope. The measured bring up time <10 μ s, which aligns with the timing requirements for our analog front-end (AFE). This is crucial because the bipolar $\pm 2.5V$ analog supplies must be synchronized with 2^{12} clock cycles of each other to ensure stable operation. Given our 2.084MHz clock frequency, this calculation means the -2.5V rail must be stabilized within 2ms of other power rails.

This test was conducted under nominal load conditions in the design lab, confirming that the power sequencing meets design expectations. The successful demonstration of this power on behavior indicates that the power system is performing as intended, supporting further system integration.

Analog Front End Board

Electrical Design Details – Analog Front End

The analog front-end (AFE) portion acts as the nervous system, responsible for acquiring and conditioning the EMG signals that are later interpreted by the microcontroller. The ADS1299 was chosen as it was a well-documented IC in the biomedical field, providing low-noise, high-resolution signal acquisition. The AFE interfaces with the STM32 microcontroller via a SPI communication protocol. Proper decoupling and filter techniques were applied to ensure low noise from power and the input signals. With 8 input channels and a reference channel for a grounding point, all signals are filtered and processed inside the ADS1299 chip, to which the results are then fed through an SPI communication protocol.

The ADS1299 was designed with a custom PCB that follows the block diagram as shown below in Figure 18 from the input signals to the data sent to the microcontroller.

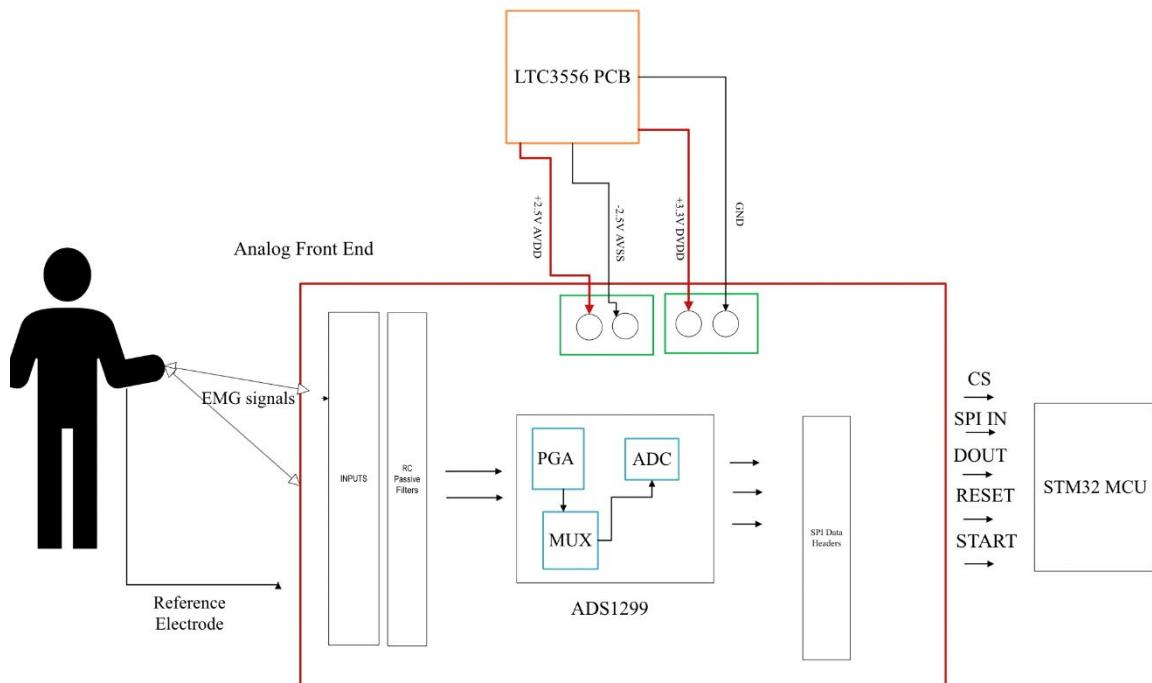


Figure 18: AFE PCB Simple Block Diagram

Figure 18 above shows the block diagram representation of the AFE PCB, from input, the processing, and the output

Below in Figure 19, Figure 20, Figure 21 is the full schematic of the ADS1299. The next two figures, Figure 22 and Figure 23, are the layouts of the board, with the first showcasing every component and its traces, and the second giving an easier follow-up of just the components.

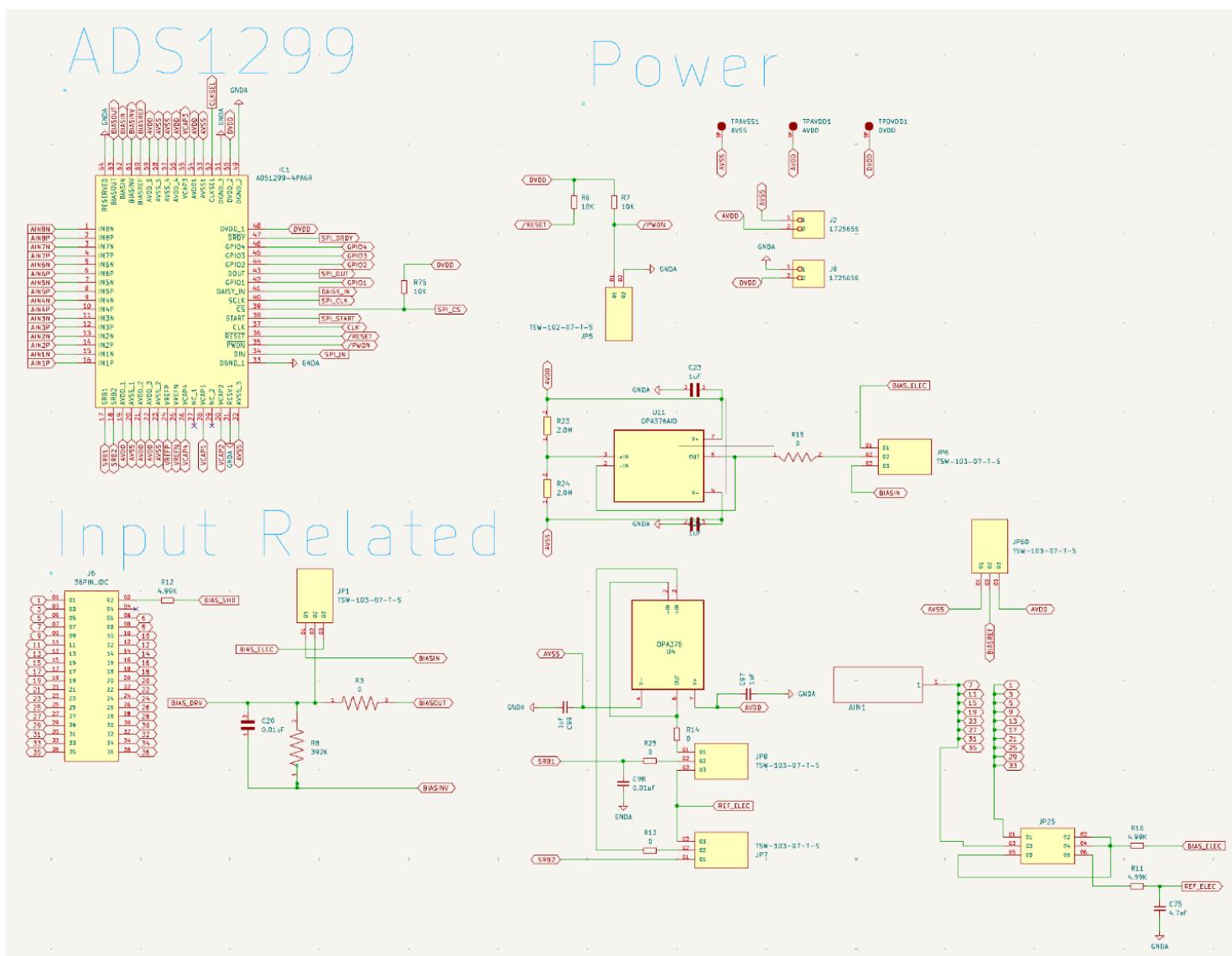


Figure 19: AFE PCB Schematic Power & I/O

Figure 19 is the schematic of ADS1299, Power, and Input related components and wirings for the layout

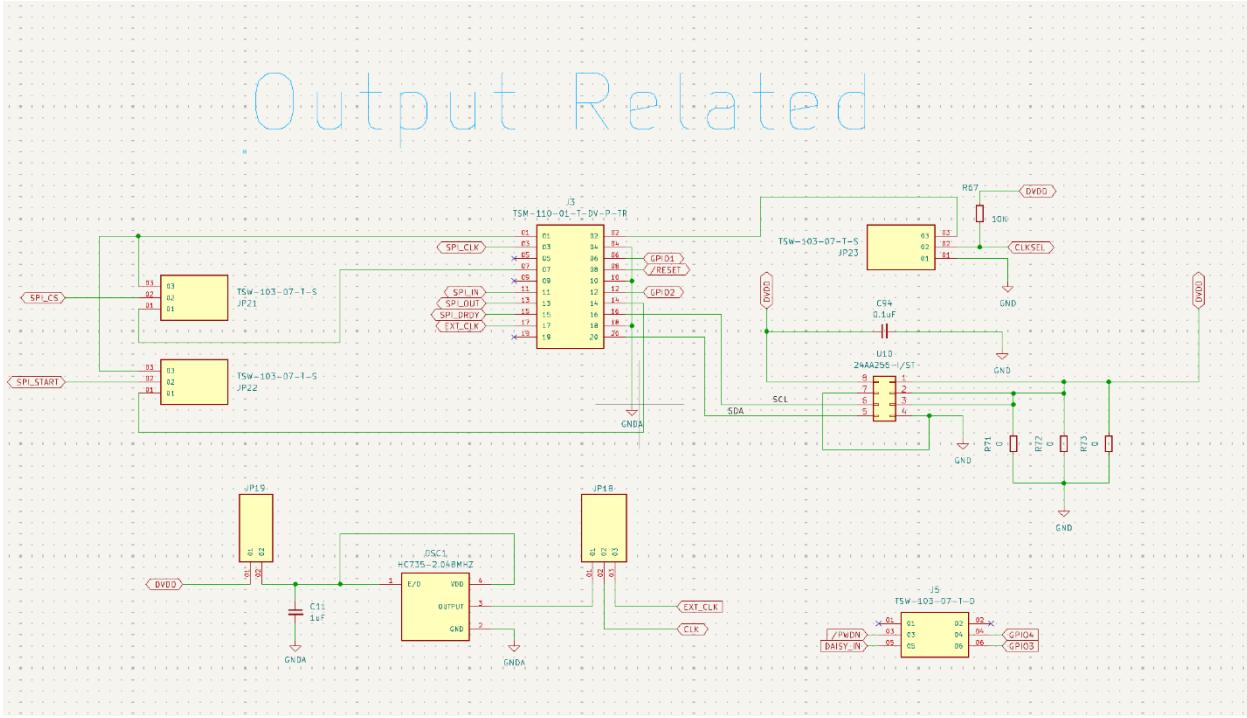


Figure 20: AFE PCB Schematic Output Items

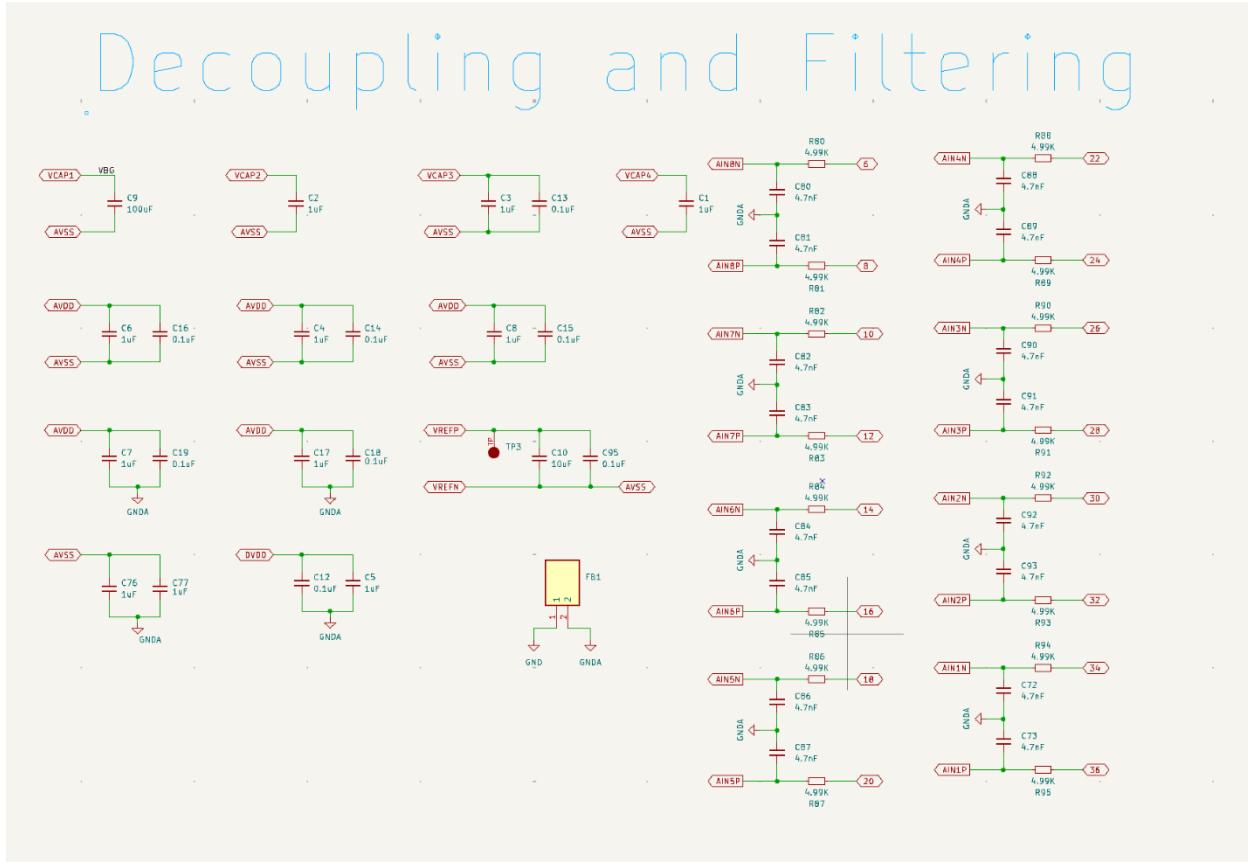


Figure 21: AFE PCB Schematic Decoupling & Passives

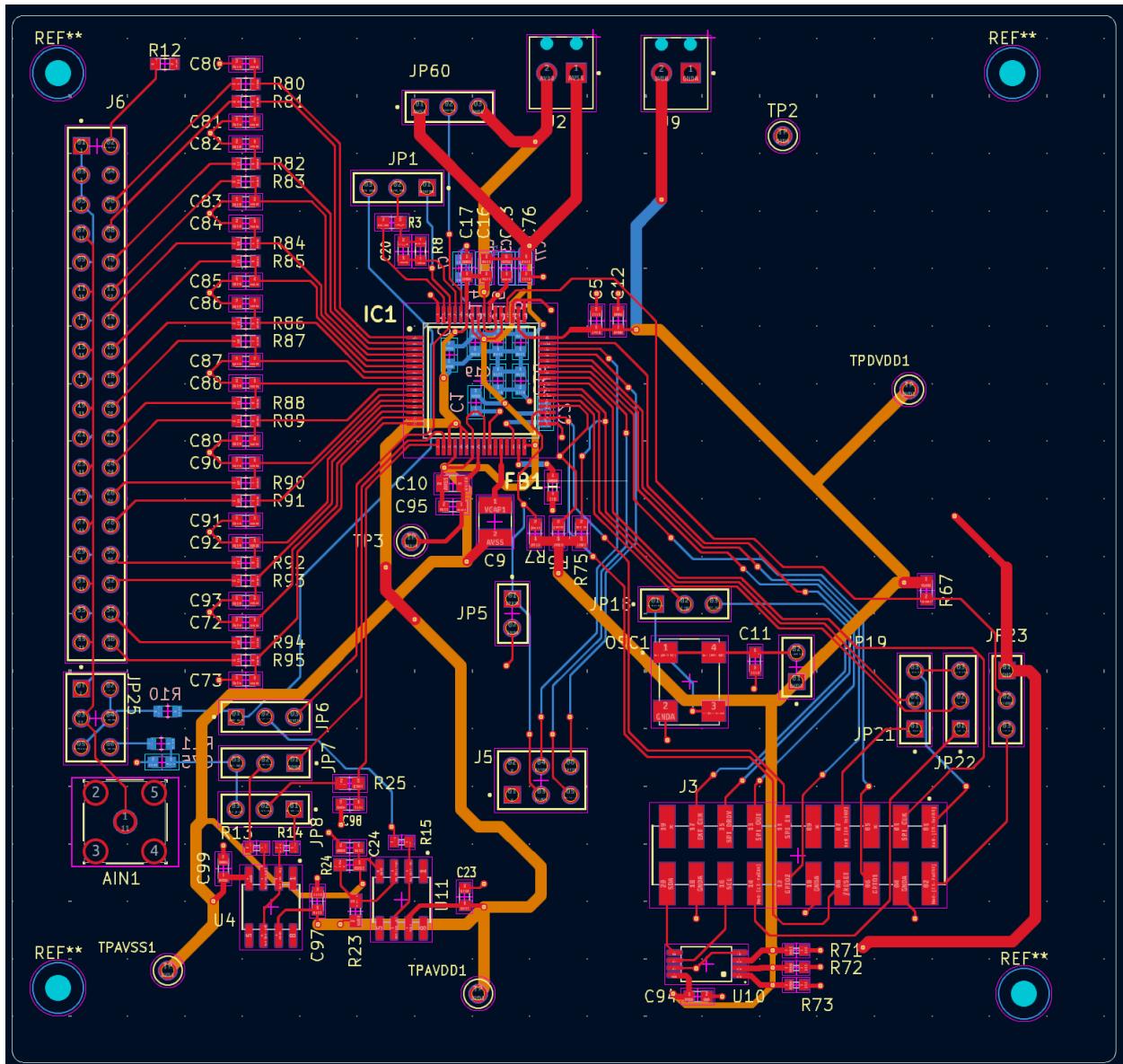


Figure 22: AFE PCB Circuit Side View

Figure 22 shows all layers of the board except the ground, including top, bottom, and power. These give bottom, top, and power traces to show all routes.

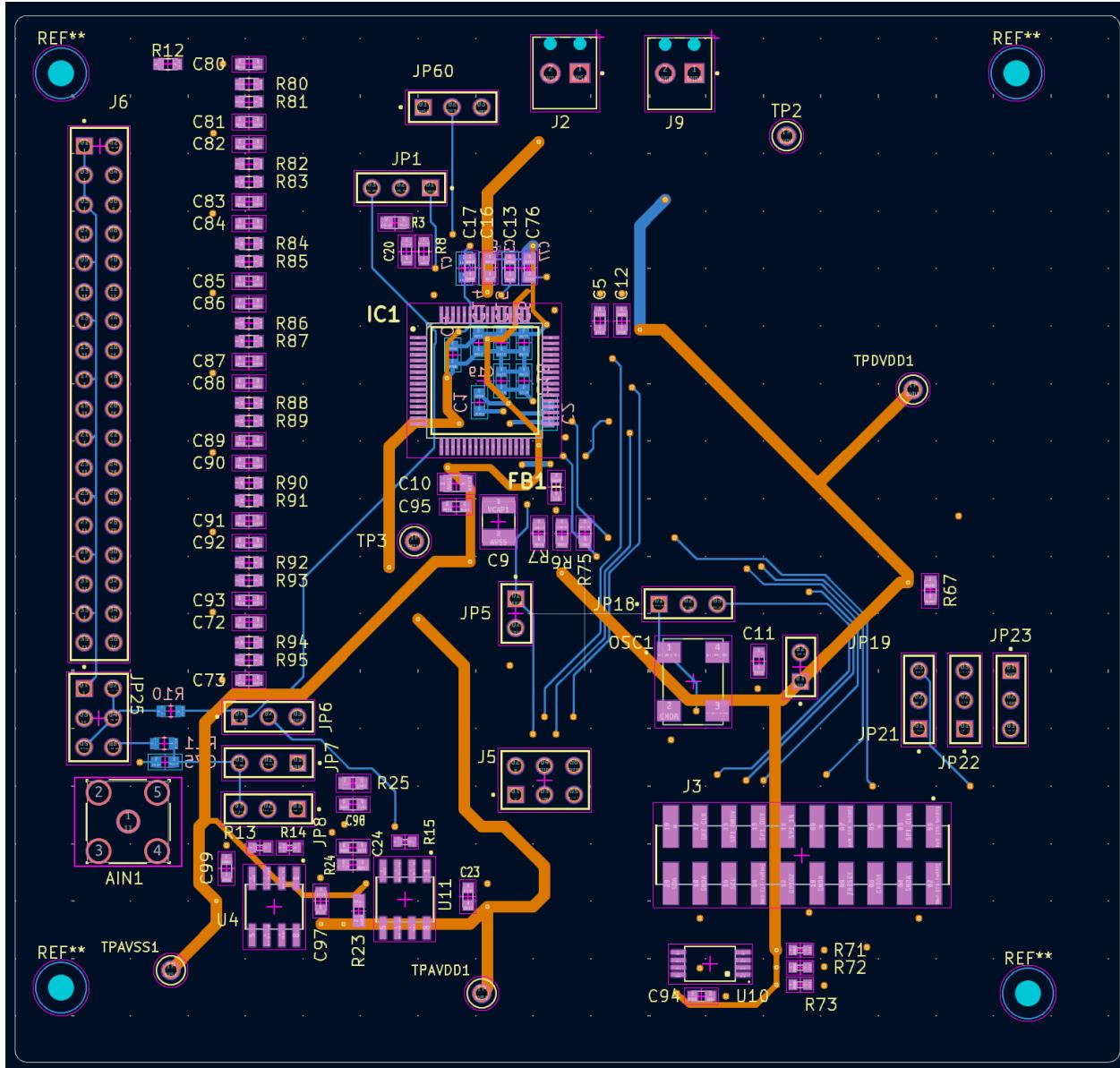


Figure 23: AFE PCB Component Side View

Figure 23 above shows the board without top layer traces to better visualize all the components and the large power routes.

Input/Output

Shown on the left side of our layout and final design figures (Figure 22 and Figure 23), we see J6 with 36 headers meant to act as the input terminals for the positive and negative leads for each EMG signal being inputted to the ADS1299. Before they enter the ADS, they go through an RC filter for noise reduction. JP25 holds the terminal for the reference and bias electrode placement. We are currently using the reference only, unless we determine with more testing that bias gives a cleaner result with a specific configuration of the RC configuration below JP1 near the top. JP6, JP7, and JP8 connect to the amplifiers U4 and U11 for the Bias Electrode configuration.

On the bottom right, there is J3 with a variety of headers. Although only the SPI lines are used, the rest of the I2C and other ports were connected in case other testing was necessary if the SPI lines somehow failed. JP21, JP22, and JP23 all have jumpers to enable the headers at J3 to be used without being set to 0. JP5 acted as additional GPIO pins in case they were needed, with JP5 as a physical power-down portion that would be activated with a jumper.

Power

At the top of the board ((Figure 22 and Figure 23), J2 and J9 are two screw terminals that take in power from our power management boards. J2 is responsible for the analog voltage +2.5 AVDD line and the -2.5 AVSS line. J9 takes the digital voltage +3.3V DVDD and the digital ground. The analog ground pin for AVDD and AVSS is provided next to J9 as TP2.

Decoupling capacitors were placed at the input points next to the pins that took in the power lines, ranging from 0.1uF to 10 uF, depending on the voltage running through them. Additional capacitors were placed near the Vcap pins as defined by the datasheet. FB1 was a ferrite bead added to connect the digital and analog grounds together and tie them to the ground on the ADS chip.

Test Points

TPDVDD1, TPAVSS1, and TPAVDD1 were all added as debugging points to check if proper power was flowing and to check for shorts

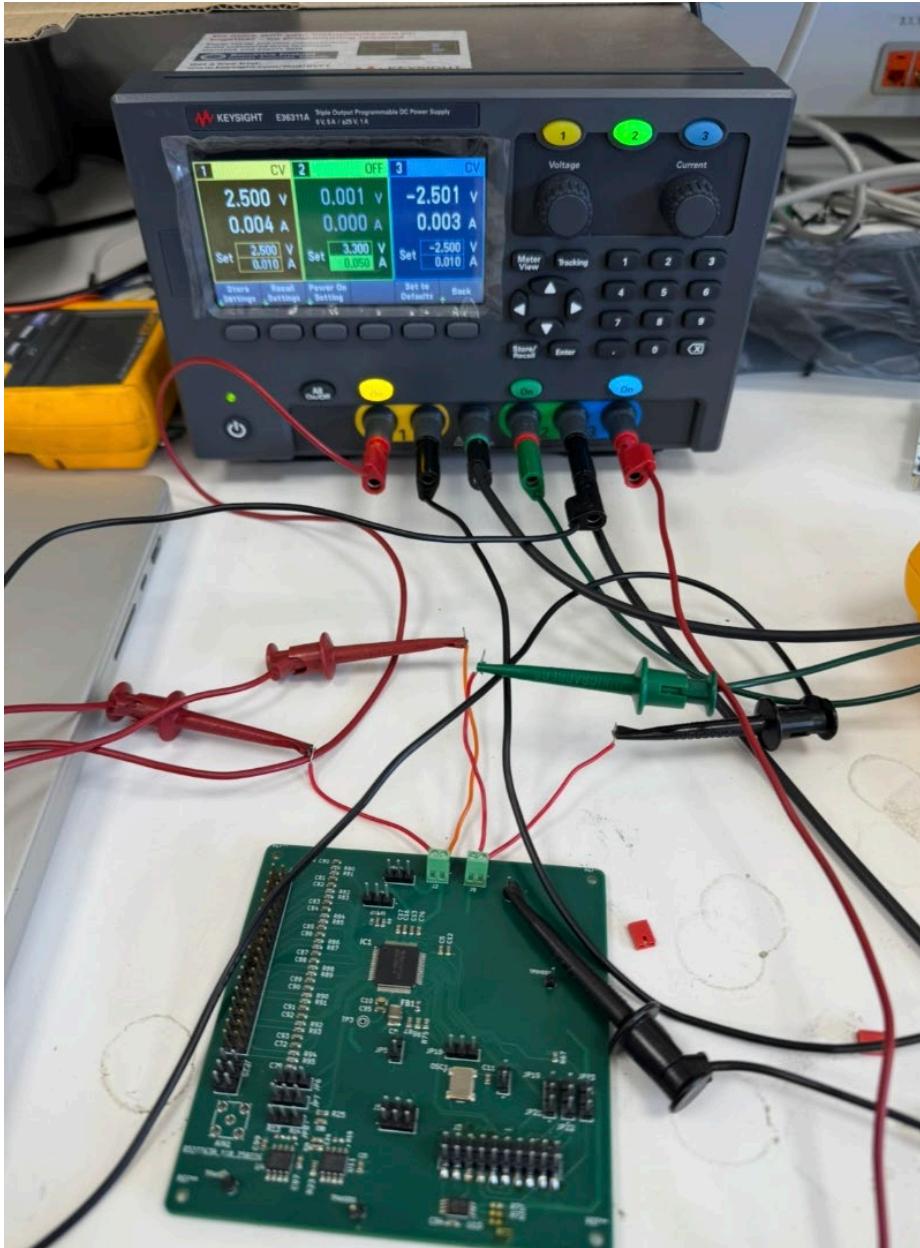


Figure 24: AFE PCB Test Setup

Figure 24 shows the testing setup for the ADS1299 with a power supply

Debugging the ADS1299

When Power was first applied to the AFE board as shown above in Figure 24, the +2.5V and –2.5V were supplied with the correct current, but the +3.3V reached a current limit that was placed on the power supply. To figure out what was happening, we first checked the continuity at each pin from the 3.3V input. Once it was confirmed there was a short, the layout was checked thoroughly and a zero-ohm resistor that connected the DVDD to a ground pin was found. This acted as a wire which caused a short, and once removed, the power was stabilized and checked via the test pins to confirm power was routed throughout the board. Figure 25 shows the three zero-ohm resistors removed below J3.

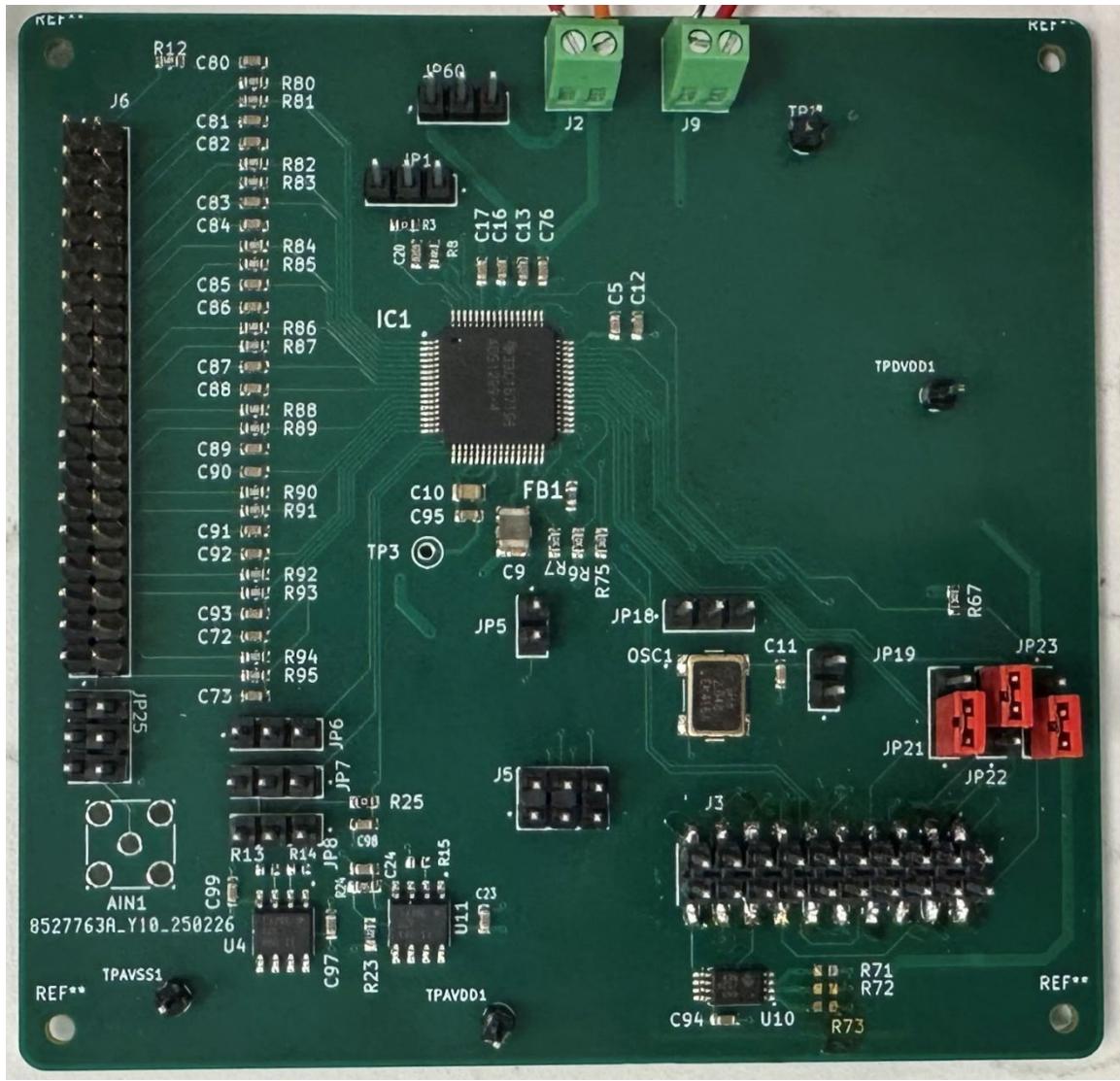


Figure 25: AFE PCB Final Assembly

Figure 25 is a picture of final ADS board after hardware debugging

Microcontroller Board

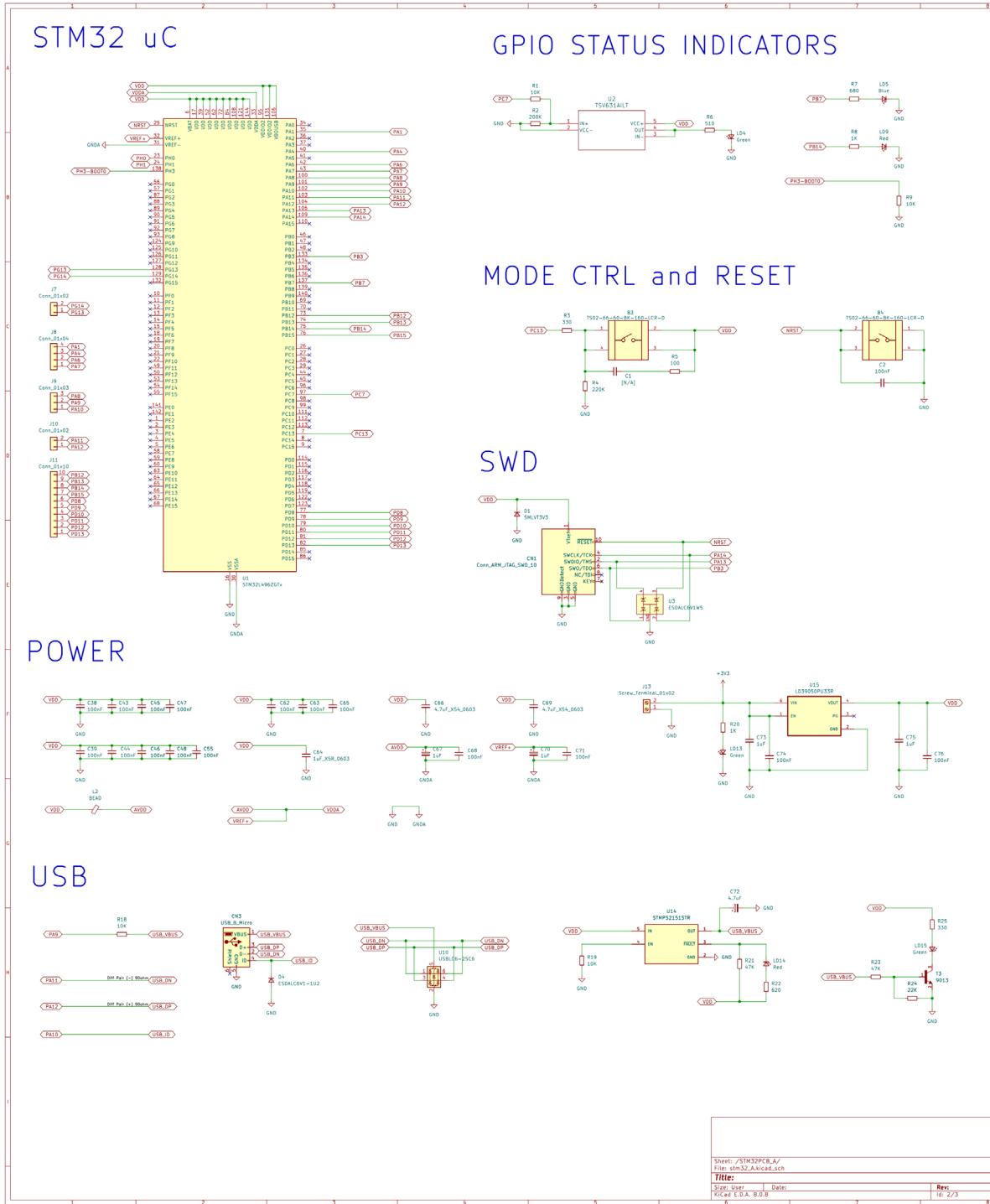


Figure 26: Microcontroller PCB Schematic

Figure 26 shows the finalized schematic of the MCU PCB. Modules include STM33 uC, power circuitry, USB interface, SWD interface, LED status indicators, and pushbutton configurations.

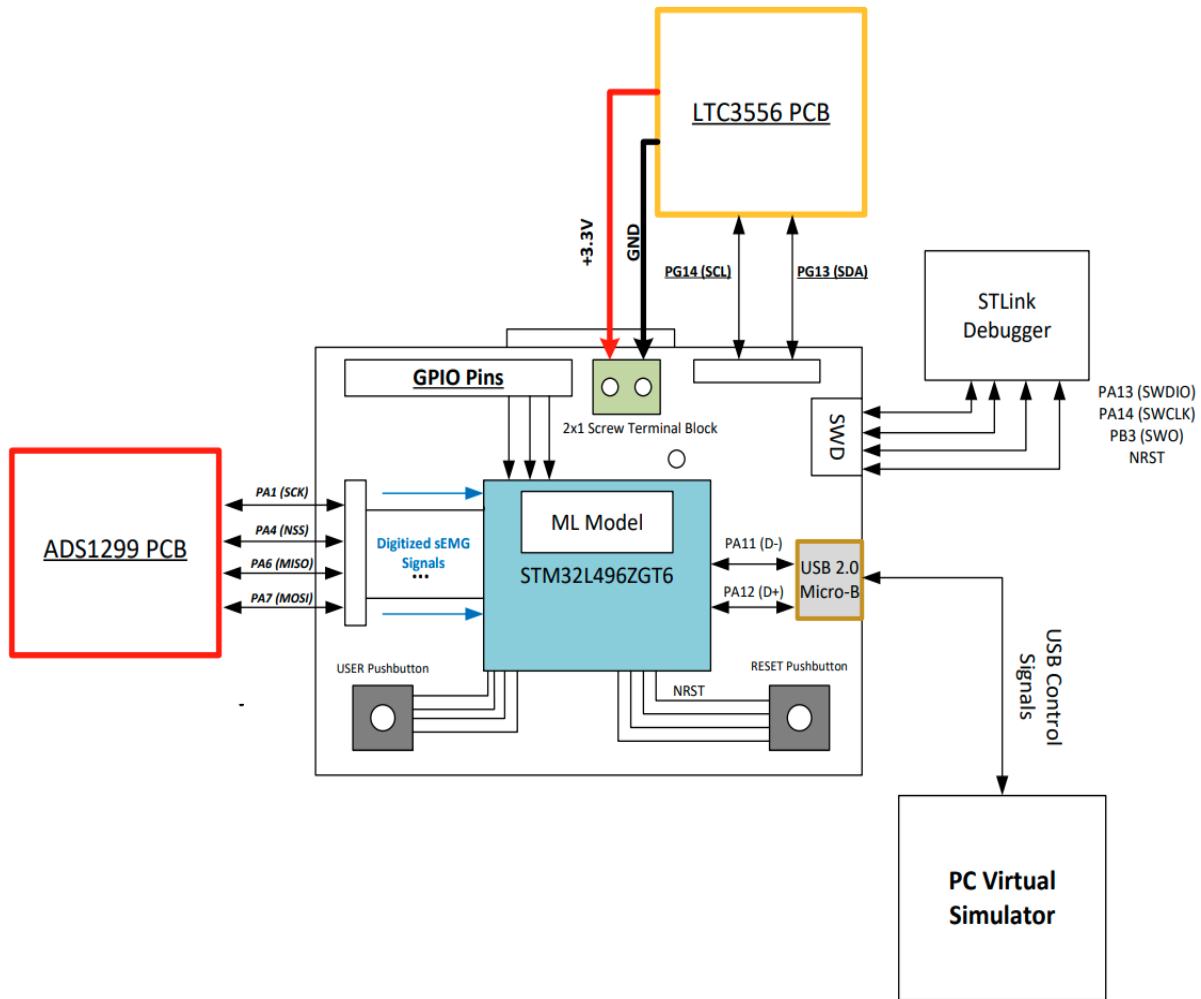


Figure 27: Microcontroller PCB Simple Block Diagram

Figure 27 shows the block diagram of all external/internal processes for the MCU PCB. Emphasizing the interfaces between PMIC, AFE, and USB Micro-B.

Electrical Design Details – Microcontroller Unit Selection

For the HANDS-EMG device, the *microcontroller unit* (MCU) PCB serves as the “brain” of this system. Moreover, this module is primarily responsible for tasks such as: processing all digitized surface EMG signals received from the AFE, classifying them as their corresponding gestures through feature extraction, and transmitting control signals to the PC simulator via USB Micro-B interface.

The specific MCU IC being used is the STM32L496GT6, which was chosen due to its state-of-the-art capabilities that closely align with system requirements. This MCU is considered

to be ultra-low-power as it is based on the power-optimized 32-bit, 80MHz *ARM Cortex-M4* core, specifically designed for energy-efficient applications such as the HANDS-EMG device. It operates on a power supply range of 1.71 V to 3.6 V, in which the regulated 3.3 V rail from the LTC3556 module will provide. The MCU provides up to 1 MB of non-volatile flash memory, which is exceedingly sufficient to upload the trained ML model for proper data classification.

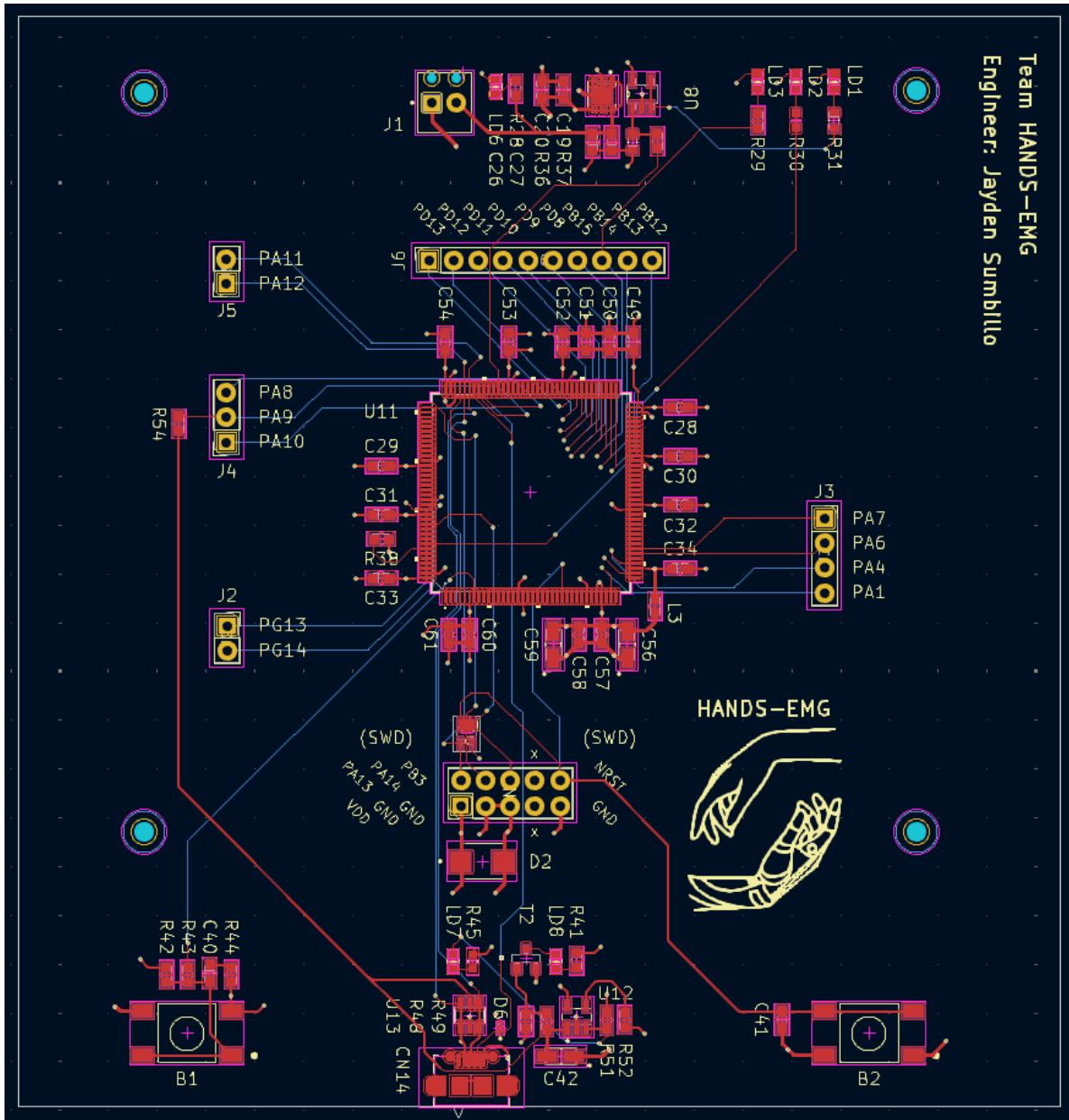


Figure 28: Microcontroller PCB Circuit Side View

Figure 28 shows the Top and Bottom layer (F.Cu/B.Cu) of the MCU PCB. These layers contain both signal and power traces used to wire all components. Inner copper layers are used as copper-filled planes for both VDD and GND.

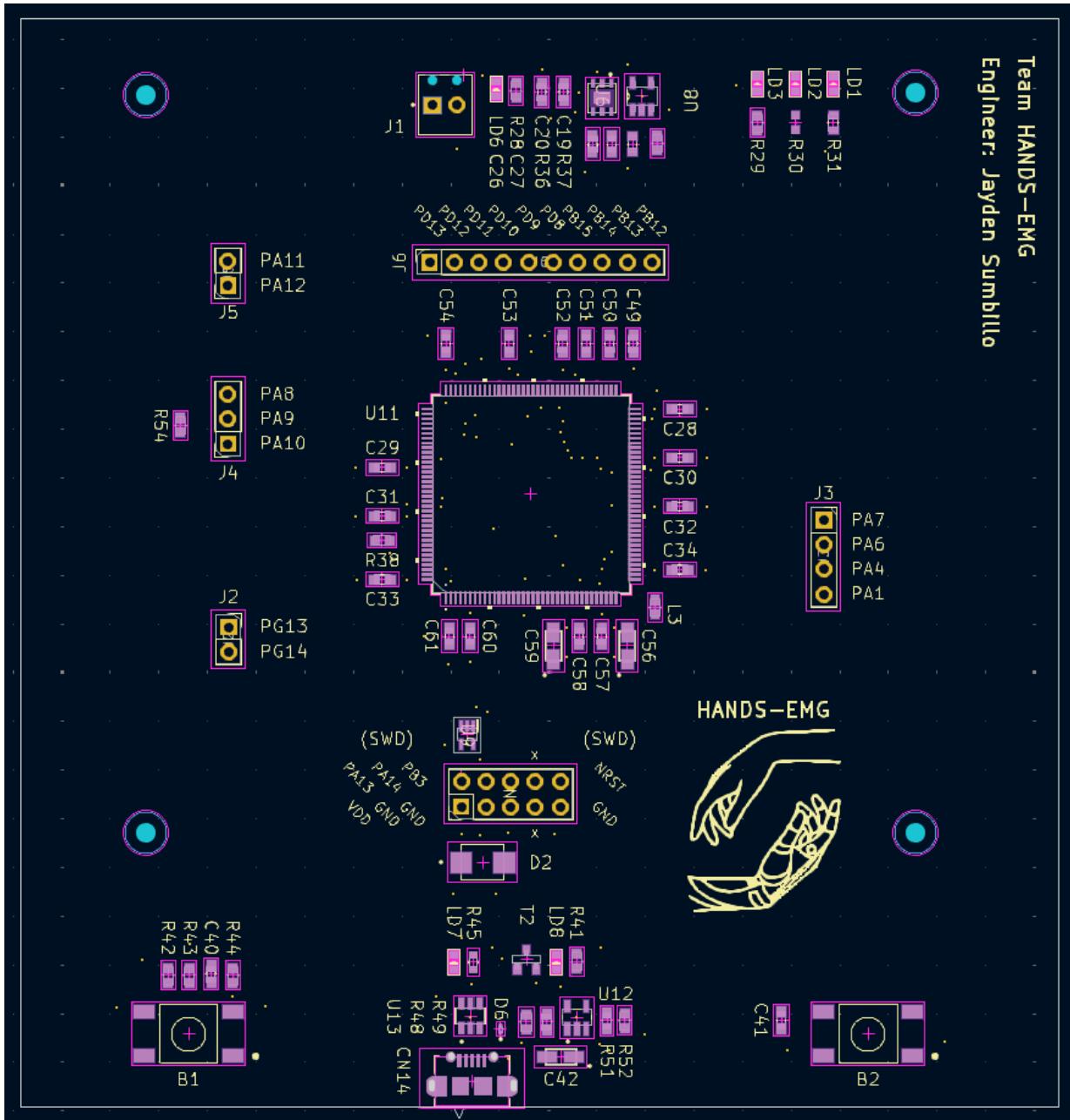


Figure 29: Microcontroller PCB Component Side View

Figure 29 shows all components of the MCU PCB are placed on the top layer (F.Cu) to minimize assembly costs. PCB includes various package types such as SMD, THT, and QFN.

Electrical Design Details – SWD Interface & Communication Lines

The prototype PCB for the MCU is designed to expose several pin segments for various functions. By using a 2x5 female header pin arrangement, this allows the ARM-based IC to be externally debugged and programmed through a Serial Wire Debug (SWD) protocol.

To communicate with the LTC3556 PMIC, pins PG14/PG13 are exposed to enable communication between the two modules via I₂C (shown on the left). For connecting to the ADS1299 AFE module, the board utilizes SPI line protocols (SCK, MISO, MOSI, etc.) that are exposed by pins PA1, PA4, PA6, PA7 shown on the right-hand side.

The remaining female headers are used to expose other general-purpose-input-output (GPIO) pins for flexible and contingent measures.

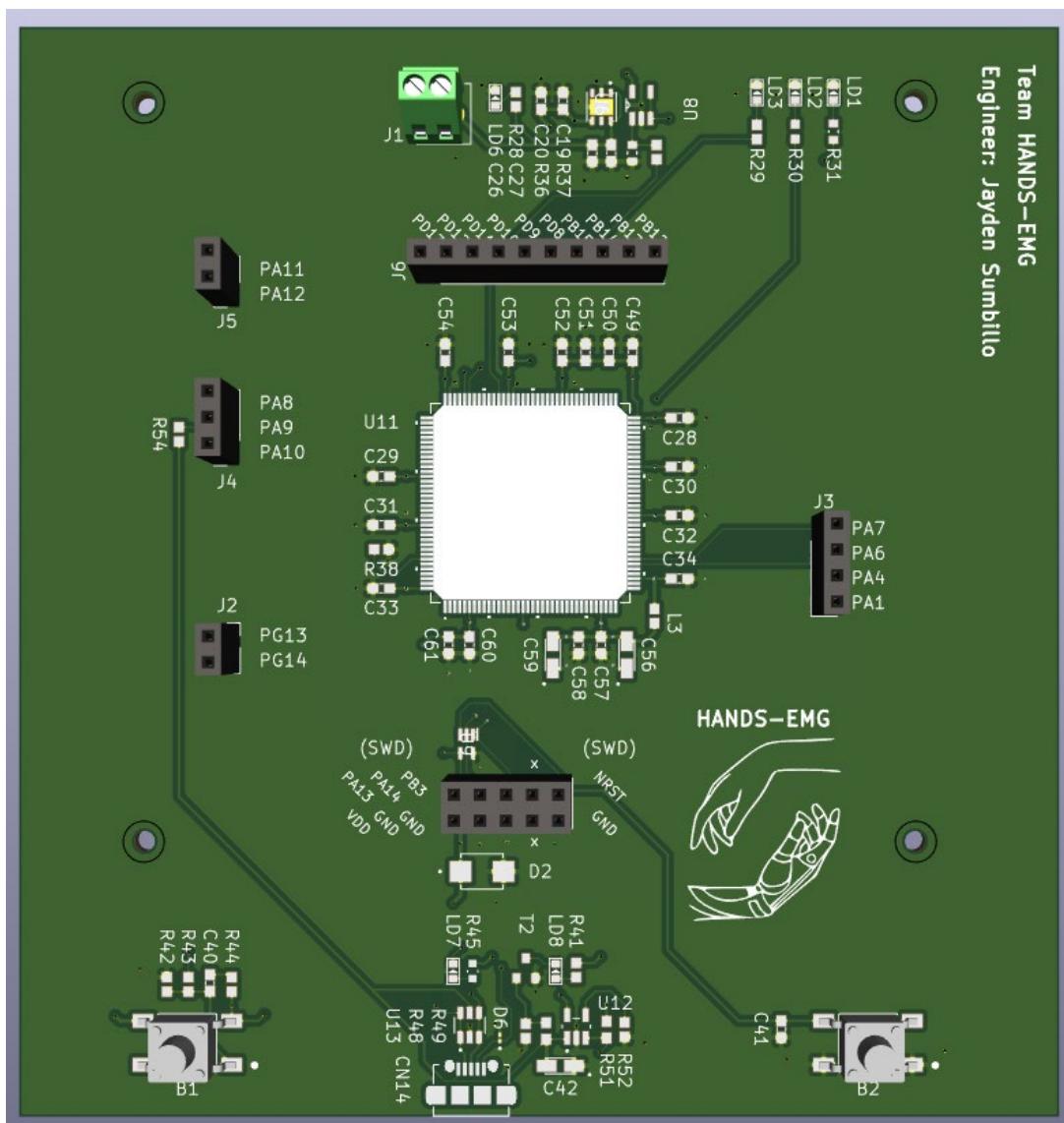


Figure 30: Microcontroller PCB 3D Rendering

Figure 30 is a KiCad 3D Rendered PCB – A visual of the footprints and placement of the STM32L496ZGT6 prototype board.

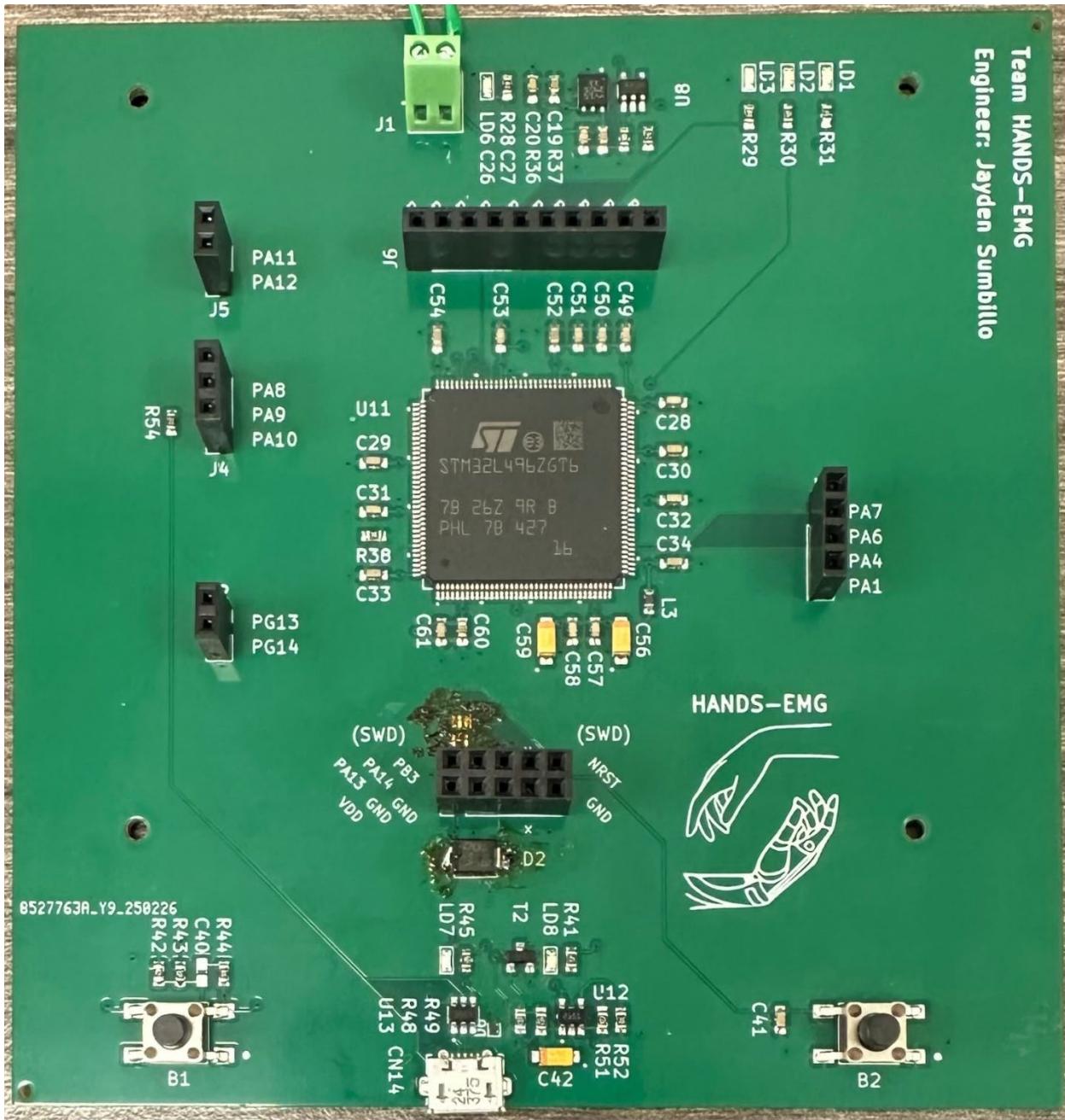


Figure 31: Microcontroller PCB Final Assembly

Figure 31 is the physical MCU PCB after fabrication and assembly. Diode (D2) was orientated 180 degrees for proper functionality. The ESD protection IC for the SWD interface was removed due to improper pad assignments.

Team HANDS-EMG
Engineer: Jayden Sumbillo

Electrical Design Details – Power Supply Connection

In order to provide a proper method of supplying power, the board includes a two-position screw terminal block that enables wires, 3.3 V rail and GND, to be securely connected to the PCB. This specific connector allows wires to be clamped down through the physical metal frame and screw. The input 3.3 V supply is fed into a power regulating circuitry that uses both an LDO IC and operational amplifier to regulate it into a “VDD” copper plane, used to power the rest of the circuit.

Electrical Design Details – USB 2.0 Micro-B Interface to PC

Shown at the bottom of the PCB in Figure 31, a USB 2.0 Micro-B connector is installed as an interface between the MCU and the PC simulator. All surrounding passive components and circuitry provide the proper protection required to prevent any distortion of the USB data lines (PA11/PA12). This allows the MCU to send clean control signals to the simulator based on the classifications made from the surface EMG signals.

Electrical Design Details – Onboard Components

The two tactile switch-pushbuttons, both placed at the bottom corners of the PCB, are assigned for different functions. The left pushbutton is a “USER” button that allows the MCU to switch between operating modes. The other pushbutton is used to reset the MCU IC through the NRST pin.

Several LEDs are installed around the board to visually indicate system behavior, such as whether power is being supplied, the event of data transfer, or SWD programming.

STM32 PCB Debugging

Once the MCU PCB design was finalized and verified through a series of DRC tests, the design files and BOM were sent to the manufacturer for both fabrication and assembly. After encountering issues with diode orientation and improper pad assignments, the components were carefully rearranged through manual soldering as seen in Figure 31.

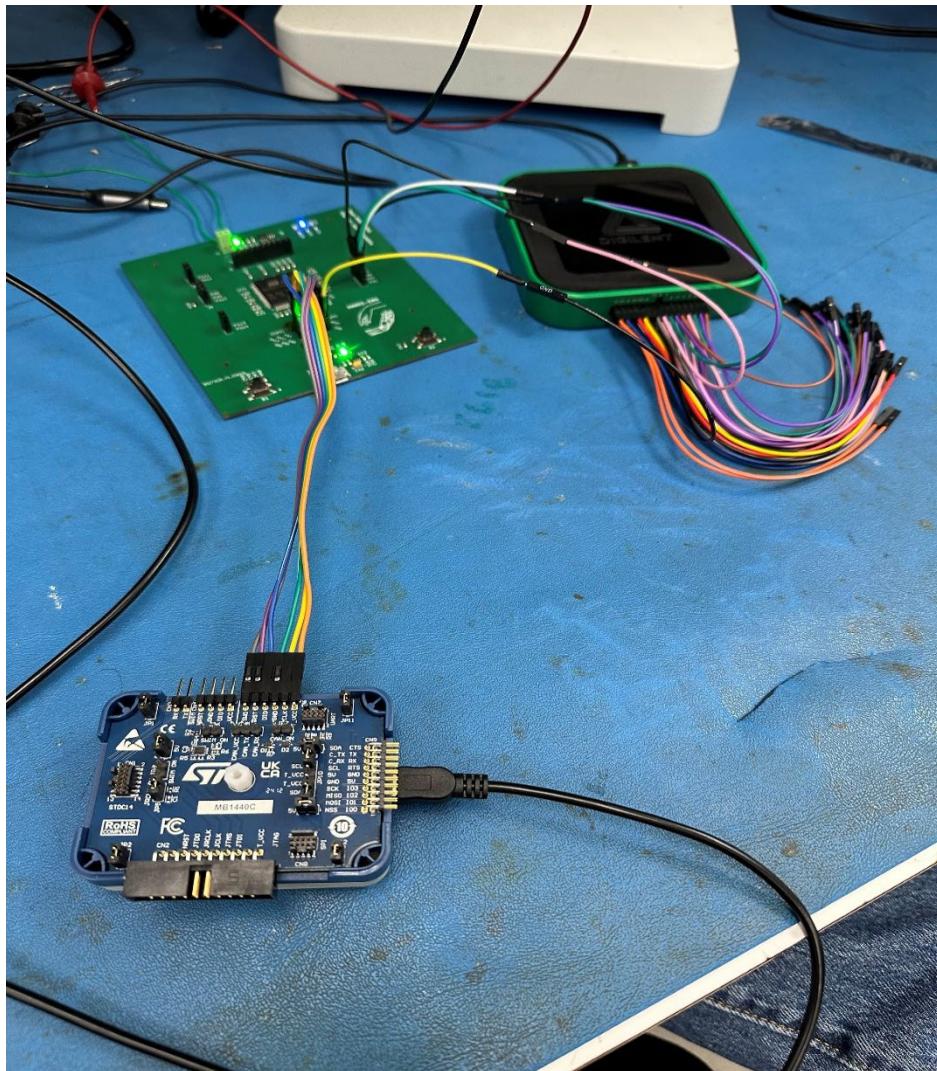


Figure 32: Microcontroller PCB Flash Test

Figure 32 is the test setup for the uC PCB program flash test.

As shown in Figure 32, the fully functional and powered-on MCU PCB is connected to the *ST-Link Utility* external debugger through the following pins: PA13 (SWDIO), PA14 (SWCLK), PB3 (SWO), and NRST (NRST). Prior to testing, the first task was to wipe the STM32's on-chip flash memory in order to ensure any previous firmware is removed. This was accomplished simply by utilizing the debugger's built-in "Erase" function. The device's memory was verified to be successfully cleared by assessing the console output through the STM32CubeIDE program.

To test the MCU's basic functionality using the external debugger, a simple “LED blink” code was uploaded to the board, with the purpose of flashing the on-board status indicator LED at various rates. Once verified, the next test involved the actual SWD debug interface. The interface was tested and verified by stepping through the prototype firmware and uploading the pre-trained ML model, resulting in zero present errors.

Wiring Diagram

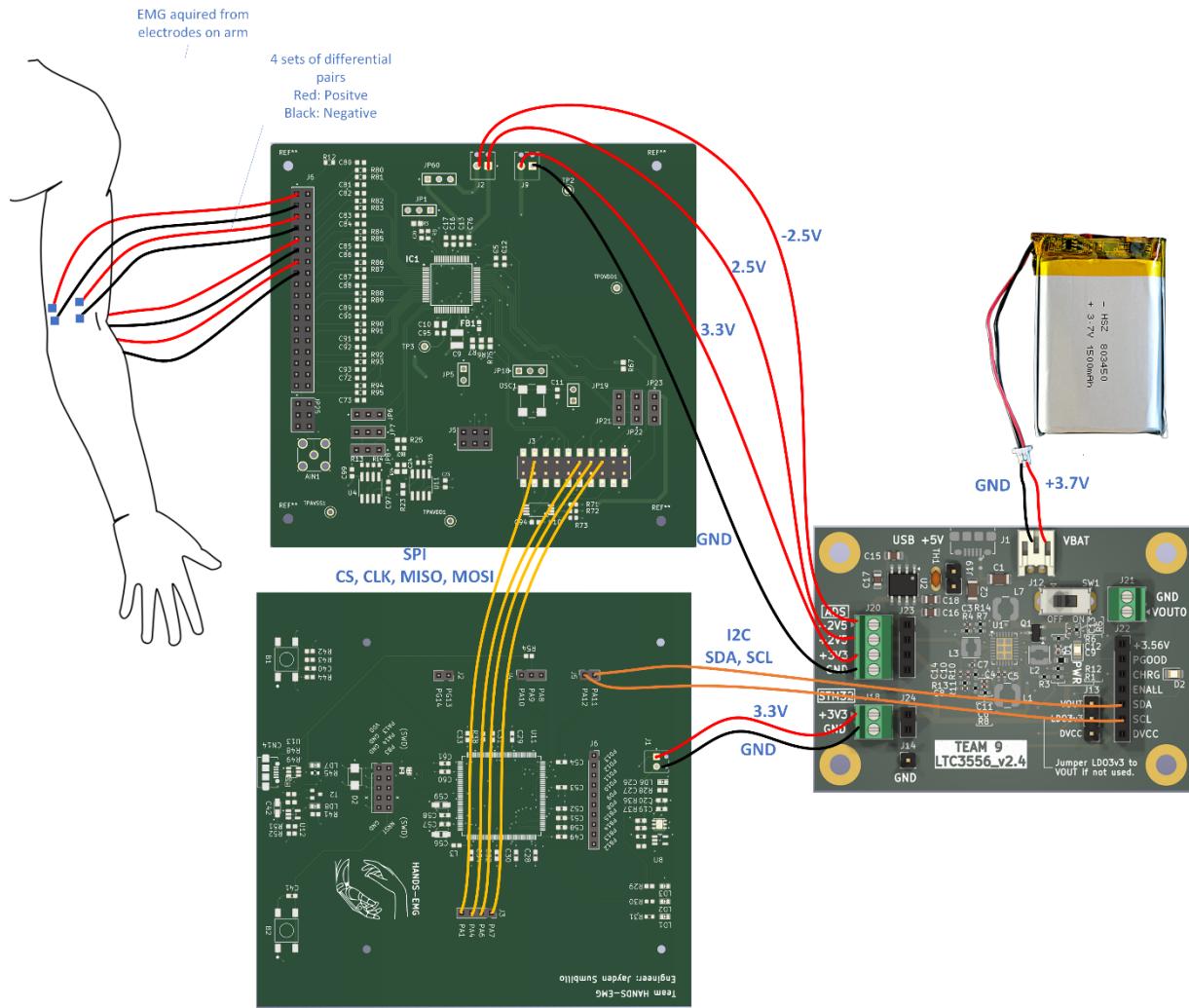


Figure 33: Integration Wiring Diagram

Figure 33 depicts how our 3 subsystems are wired together for operation. The AFE board is pictured on the top, and it is directly connected to the electrodes on the users arm. The microcontroller board is connected to the AFE through SPI communication lines (yellow wires) and also connected to the power management board through I2C communication lines (orange wires). The power management board is connected to the other subsystems to provide power as shown with the red wires in the figure. It is also connected to our single cell, 3.7V LiPo battery. All wires in the image above are 20 AWG.

Software and Firmware Readiness

The development of the system's software and firmware focused on three key components: inter-board communication, machine learning (ML) integration, and real-time feature extraction. Inter-board communication was implemented using SPI and I2C protocols to ensure seamless data exchange between subsystems. The SPI interface was developed for communication with the analog front-end (AFE) board and validated using evaluation kits and a logic analyzer, confirming correct data transmission. The I2C interface was implemented for controlling the power management IC (PMIC), allowing the microcontroller to enable and disable voltage rails as needed. Successful validation of these communication protocols ensures reliable subsystem interaction, a critical requirement for the full system integration phase.

The machine learning model and feature extraction firmware were also key software components developed for the system. The ML model was trained on the NinaPro dataset, achieving an accuracy of 85%, and was successfully deployed on a microcontroller evaluation kit. The testing confirmed that the model size and RAM usage were within operational limits, and validation using sample data demonstrated expected classification performance. Additionally, feature extraction firmware was developed to process incoming electromyography (EMG) signals in real time. This firmware was tested using sample data on evaluation kits, verifying its ability to extract relevant features efficiently. The successful development and validation of these software components establish a strong foundation for the final integration of all system elements.

Figure ## below shows a sketch of the firmware for the HARDS-EMG device. All project code resides within the STM32 microcontroller. The code runs on a finite state machine where it will query the analog front end for the read EMG signals. This prompts the machine feature extraction to extract the relevant features and trigger the machine learning model with the feature vector. After a classification had been reached, the firmware sends that classification over the USB to be displayed by the computer simulation software.

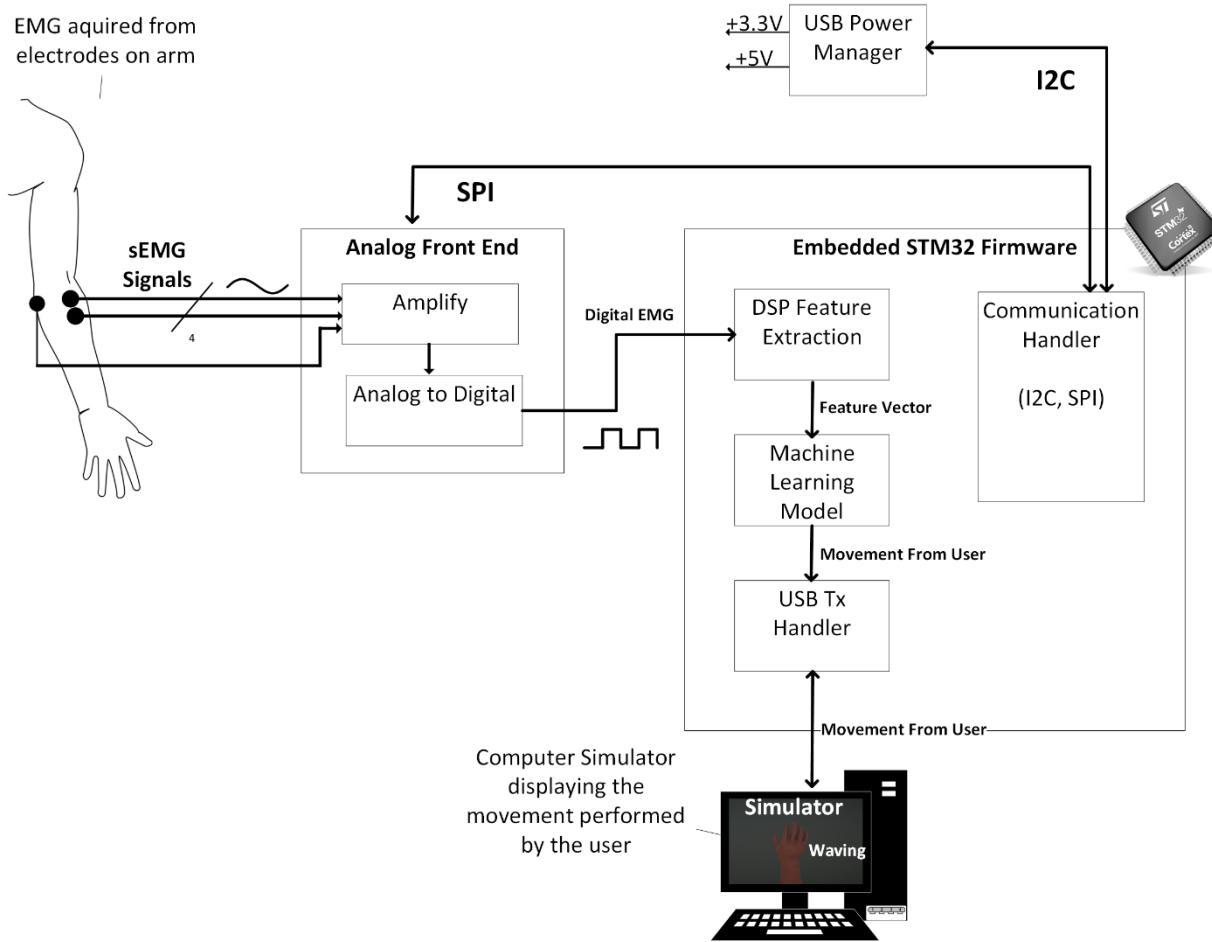


Figure 34: Software Component Overview

Figure 34 above shows our software components. Within the embedded software we have 3 major components. The feature extraction, machine learning model, and communication handler as seen above.

Inter-Board Communication

The inter-board communication firmware is responsible for initializing and managing data exchange between subsystems. Upon startup, the firmware uses the I2C protocol to enable the power rails on the PMIC board, ensuring that all necessary voltage lines are active before further system configuration. Once power is established, the firmware initializes SPI communication with the ADS1299, sending control words to configure its settings for signal acquisition. After configuration, the firmware continuously queries the ADS1299 for data, retrieving EMG signals for processing. The tests performed on the inter-board communication can be seen in Table 2 below.

Table 2: Inter-Board Communication Tests

Test	Description and Coverage	Procedure	Results
1	The I2C communication with the PMIC board was tested by sending	Connect the STM32 evaluation kit to the	The measured values of the voltage rails

	the 2 byte command word to our custom power management board. The 2 byte word enables all of the DCDC outputs and sets the battery charging current limit to 100mA.	SDA and SCL lines on the custom power management board. Flash the code to the STM32 as I2C communication is only performed on bring-up. Observe the voltage outputs on the PMIC board read their proper values.	were 3.64V, 2.56V, and -2.47V after the I2C communication took place. This data was presented in the hardware readiness section.
2	The SPI communication with the ADS1299 eval kit was tested by sending a byte command word and receiving data on our STM32 eval kit. This was monitored by a logic analyzer.	Connect the STM32 evaluation kit to the SPI (CS, CLK, MOSI, MISO, DRDY) lines of the ADS1299 evaluation kit. Flash the STM32, observe the control words being sent via SPI and the data being received over the logic analyzer.	The communication procedure can be visualized in Figure 35 below.

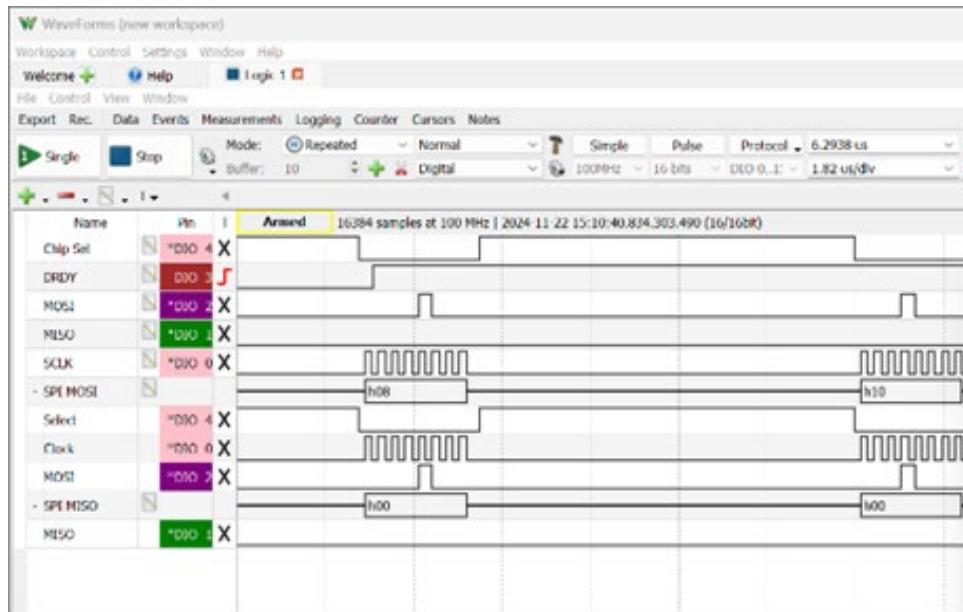


Figure 35: SPI Communication with ADS1299 Scoped

Figure 35 above shows the communication sequence with the ADS1299 eval kit scoped with a logic analyzer

Machine Learning

In this project, a machine learning algorithm is being utilized to perform real-time inference on surface electromyography (sEMG) signals captured by an analog-to-digital converter (ADC). The primary objective is to process these raw signals, extract meaningful features, and apply a pre-trained model to classify or predict motor activity, such as muscle contractions or gestures.

To properly decode the patterns presented by the sEMG signals, the machine learning model that is flashed to the STM chip must be able to perform in real world conditions, and not just on test data. Firstly, the model must show a good performance during the training phase. As covered in the final report for the previous semester, the model performance was poor, and it was discussed that further data analysis must be performed to make the problem simpler for the machine learning model to perform inferences. Utilizing a one-way variance analysis utilizing all channels from the NiNapro sEMG dataset, the gestures to be performed can be ordered from highest to lowest impact on the output classifier. Once this analysis is performed, the same method can be used in the opposite direction to find what channels are most useful to the prediction of these gestures. After this analysis was performed, figure Figure 36 shows what gestures are now capable of being predicted. As stated above, this analysis was performed in a bidirectional manner, but ultimately did not change the electrode placement that was previously specified.

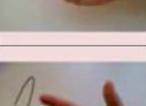
		Rest		
6	Fingers flexed together in fist		13	Wrist flexion 
	9 Wrist supination (axis: middle finger)		16	Wrist ulnar deviation 

Figure 36: Revised List of ML Gestures

Figure 36 is a revised list of gestures the device can predict reliably.

To verify the compatibility of the created model, and the microcontroller, STM has a prebuilt tool for uploading the machine learning model onto the device and performing test inferences to analyze the network requirements. As shown in figure Figure 37 the device is compatible with the crafted machine learning model and leaves room for a further increase in model complexity as it is well within budget for both flash and memory requirements.



Figure 37: ML Network Analysis for STM32

Figure 37 is a network analysis performed on the CubeMX platform, which demonstrates proper resource allocation given the hardware requirements

For isolated testing, the model inference must be performed on the eval kit prior to flashing to the custom microcontroller board. To do this without full system integration, the trained model can be loaded and evaluated on the evaluation board utilizing STM NanoEdgeAI Studio. This software can validate model architecture without the need for configuring communication protocols through C code. This simplifies the testing process of the machine learning model significantly by only needing the eval board, and a USB cable to thoroughly evaluate the ML performance. Figure Figure 38 shows the experimentation results of the ML model running on the target. The successful flashing of the target device with the tflite model does not have any outward indicator other than the LEDs onboard. For this, a single LED was turned on when the device entered its main loop, thus indicating the model is flashed successfully and is ready to receive data. Figure Figure 39 shows the physical state of the eval board at this point in the testing phase, indicating that the model is ready to receive data.

```

Saving validation data...
output directory: C:\Users\pears\stm32cubemx\emg_net_output
creating C:\Users\pears\stm32cubemx\emg_net_output\emg_net_val_io.npz
m_outputs_1: (10, 1, 1, 18)/float32, min/max=[0.002710, 0.172004], mean/std=[0.055556, 0.038388], nl_4
c_outputs_1: (10, 1, 1, 18)/float32, min/max=[0.002710, 0.172004], mean/std=[0.055556, 0.038388], nl_4
Computing the metrics...
Cross accuracy report #1 (reference vs C-model)
-----
notes: - the output of the reference model is used as ground truth/reference value
      - 10 samples (18 items per sample)
acc=100.00% rmse=0.000000013 mae=0.000000009 l2r=0.000000193 mean=0.000000 std=0.000000 nse=1.000000 cos=1.000000
Confusion matrix (axis=-1) - 18 classes (10 samples)

C0   0   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
C1   .   0   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
C2   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
C3   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .
C4   .   .   .   .   0   .   .   .   .   .   .   .   .   .   .   .   .   .
C5   .   .   .   .   .   0   .   .   .   .   .   .   .   .   .   .   .   .
C6   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .   .   .
C7   .   .   .   .   .   .   .   0   .   .   .   .   .   .   .   .   .   .
C8   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .   .   .
C9   .   .   .   .   .   .   .   .   .   0   .   .   .   .   .   .   .   .
C10  .   .   .   .   .   .   .   .   .   .   1   .   .   .   .   .   .   .
C11  .   .   .   .   .   .   .   .   .   .   .   0   .   .   .   .   .   .
C12  .   .   .   .   .   .   .   .   .   .   .   .   3   .   .   .   .   .
C13  .   .   .   .   .   .   .   .   .   .   .   .   .   0   .   .   .   .
C14  .   .   .   .   .   .   .   .   .   .   .   .   .   .   0   .   .   .
C15  .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   0   .   .
C16  .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   2   .
C17  .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   0

Evaluation report (summary)
-----
Output    acc      rmse      mae      l2r      mean      std      nse      cos      tensor
X-cross #1 100.00% 0.000000013 0.000000009 0.000000193 0.000000 0.000000 1.000000 1.000000 'nl_4'.

-----  

acc : Accuracy (class, axis=-1)  

rmse : Root Mean Squared Error  

mae : Mean Absolute Error  

l2r : L2 relative error  

mean : Mean error  

std : Standard deviation error  

nse : Nash-Sutcliffe efficiency criteria, bigger is better, best=1, range=(-inf, 1]  

cos : Cosine Similarity, bigger is better, best=1, range=(0, 1]  

Creating txt report file C:\Users\pears\stm32cubemx\emg_net_output\emg_net_validate_report.txt  

elapsed time (validate): 16.698s  

Validation ended!
```

Figure 38: ML Model Validation on Device

Figure 38 above demonstrates the physical result of running the AI Studio software, where the device only indicates that it has been flashed successfully.

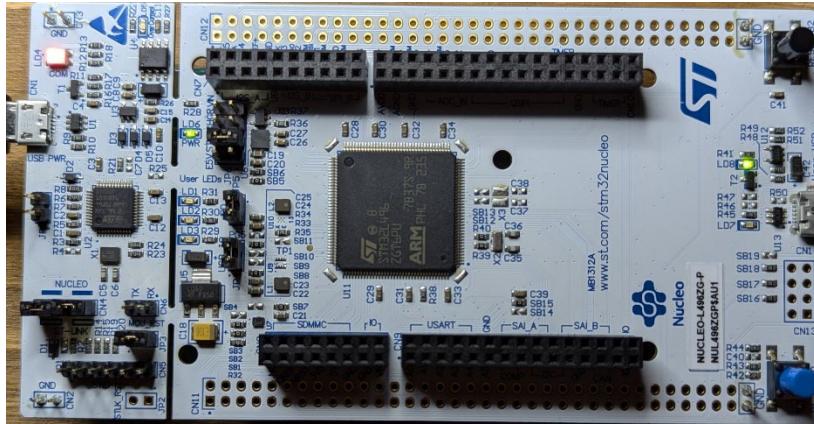


Figure 39: Successful Flashing Indicator

Figure 39 shows the physical indication lights showing successful flashing of the target device with the tflite model. Note the single indication LED on the right side of the image as described.

Feature Extraction

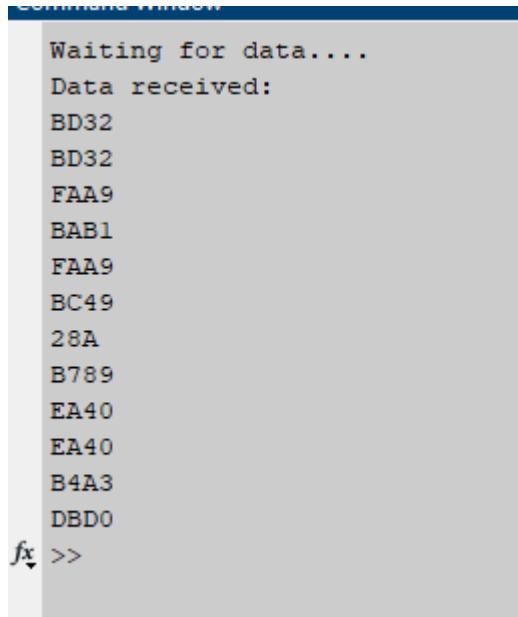
The other vital task to ensure proper functionality of the device, would be to verify that the training data for the ML model is in the same format as the data that will be received by the AFE. The AFE sends out data in 24-bit two's complement binary values, whereas the training data on the .tflite file are the quantized weights from double precision to 16 bit integer.

Tensorflow Lite natively supports quantization to 16 bits, and the compiler package from Tensorflow handles these operations by itself. The data from the AFE, however, needs to match the 16 bit integer to work properly. To test this, a window from the Ninapro dataset was flashed onto the device, and after the feature extraction was performed. The list of features that are extracted from each window are shown in Table Table 3.

Table 3: List of Features for Extraction

Statistical Metric	Abbreviation	Formula
Integrated EMG	IEMG	$\sum_{n=1}^N x[n] $
Mean Absolute Value	MAV	$\frac{1}{N} \sum_{n=1}^N x[n] $
Simple Square Integrated	SSI	$\sum_{n=1}^N x[n]^2$
Root Mean Square	RMS	$\sqrt{\frac{1}{N} \sum_{n=1}^N x[n]^2}$
Variance	VAR	$\frac{1}{N-1} \sum_{n=1}^N x[n]^2$
Myopulse Percentage Rate	MYOP	$\frac{1}{N} \sum_{n=1}^N f(x[n])$, $f(a) = \begin{cases} 1 & \text{if } a > T \\ 0 & \text{otherwise} \end{cases}$
Waveform Length	WL	$\sum_{n=1}^{N-1} x[n+1] - x[n] $
Difference Absolute Mean Value	DAMV	$\frac{1}{N-1} \sum_{n=1}^{N-1} x[n+1] - x[n] $
Second-Order Moment	M2	$\sum_{n=1}^{N-1} (x[n+1] - x[n])^2$
Difference Variance Version	DVARV	$\frac{1}{N-2} \sum_{n=1}^{N-1} (x[n+1] - x[n])^2$
Difference absolute standard deviation value	DASDV	$\sqrt{\frac{1}{N-1} \sum_{n=1}^{N-1} (x[n+1] - x[n])^2}$
Willison Amplitude	WAMP	$\sum_{n=1}^{N-1} f(x[n+1] - x[n])$, $f(a) = \begin{cases} 1 & \text{if } a > T \\ 0 & \text{otherwise} \end{cases}$

When the feature extraction test is performed on target, a COM port can be opened on MATLAB, to more easily compare the intended feature values, and the generated feature values from the firmware. Figure Figure 40 shows the received data from the device in hexadecimal format.



The screenshot shows the MATLAB Serial Output Window. The window title is 'Serial Window'. The text area contains the following output:

```
Waiting for data....  
Data received:  
BD32  
BD32  
FAA9  
BAB1  
FAA9  
BC49  
28A  
B789  
EA40  
EA40  
B4A3  
DBD0  
fx >>
```

Figure 40: Output Window of MatLab Serial

Figure 40 shows the output window of the MATLAB serial window showing the features generated from the device in hexadecimal format.

Since the data is already in MATLAB, it makes the comparison easier by utilizing the same process in which the sample window was loaded onto the board. Utilizing the fixed-point toolkit on the expected features, as calculated in double precision float, Figure 41 shows the quantization error between the intended values and the generated values from the target device.

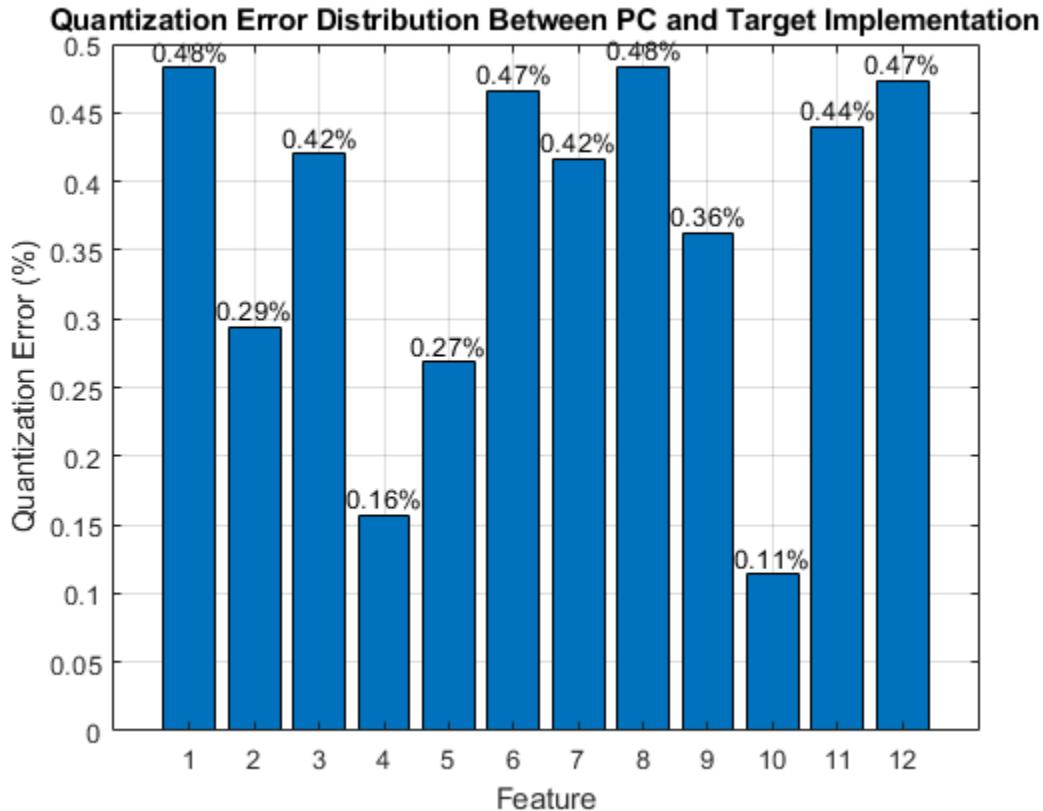


Figure 41: Quantization Error for Features

Figure 41 is the graph of the quantization error for each feature generated from the target device.

As Figure 41 would show, the quantization error is rather low. This is since the device can be preloaded with information, such as the mean and standard deviation of the training data features, the generated features can be normalized and thus do not exceed the representable range of numbers in a Q1.15 fixed point format. It must be noted however, that this process must be closely monitored during integration, as the data from the ADC can potentially make the quantization error much larger if the voltage range is not the same from the training set to the real time recordings.

Integration Plan

Our integration plan involves wiring the designed subsystems together and controlling them with our firmware. The process in which we will do this can be seen below in Figure 42. Furthermore, Table 4 below will describe each integration step and its respective validation.

The integration plan begins with us wiring the power delivery PCB to both of our subsystem PCBs, the AFE and microcontroller. In this configuration, a couple of steps will be performed. The power output will be tested and validated for integrity while the load is applied. While the power delivery board is under load, the battery life of the device will be timed before the device dies. Moving on, the microcontroller and analog front end will be wired together and the communication and command sequences from the microcontroller will be validated. Then the firmware for the microcontroller will acquire signals from the analog front end and these signals will be analyzed. Following this, the ML model will be uploaded, and the USB communication will be tested. Finally, the microcontroller will be tuned for power consumption and speed and the total system validation will be tested. Total system validation requirements can be seen in Figure 43 below.

The chart below in Figure 42 is our integration plan which is described in further detail in Table 4

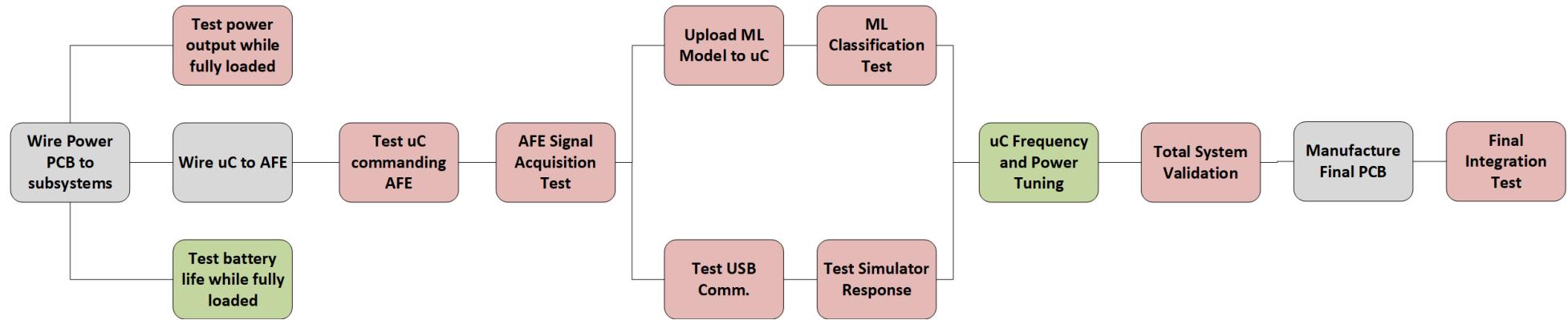


Figure 42: Integration Flow Chart

Table 4: Integration Plan Procedures

Integration Step	Procedure	Anticipated Result
Test power output while fully loaded	Connect oscilloscope probe to each power output of the power delivery PCB. Set trigger to unacceptable low voltage settings (3V for 3.3V line, 2.2V for 2.5V line, -2.2V for -2.5V line). Set trigger to single. Apply load by flashing firmware to uC and turning on AFE. Monitor scope for trigger.	No scope trigger. Voltage levels remain acceptable, no transient response that would cause device failure.
Test battery life while fully loaded	After the power output is tested while fully loaded, remain in the same setup. Allow the device to run until the power management board determines that the battery is discharged and stops operation.	Battery life is >6 hours. Take the time recorded during this procedure and apply to final report as a device metric.
Test microcontroller commanding AFE	After wiring the SPI communication lines to the analog front end. Flash the firmware to send the command words to the analog front end.	Analog front end responds to the sent command words and will begin continuous data transfer to the microcontroller.
AFE signal acquisition test	Trigger the microcontroller to acquire signals from the analog front end. Validating the signal integrity is necessary for this procedure, enable the firmware to transfer signals to a laptop via the Micro USB port. Use MATLAB on the laptop to plot these signals and verify their similarity to the recorded signals from the AFE eval kit.	Signals acquired from the AFE match the signals acquired from the evaluation kit. The signals are free from visible noise and the voltage levels match accordingly.
Test USB communication	Enable the USB OTG on the microcontroller and send a single byte representing a	The USB Host (PC) receives the byte representing classification. This will be

	classification to the USB HOST (PC).	visualized using a PuTTY terminal.
Test simulator response	Start the simulator on the PC. Select the USB port as the input to the simulator. Enable the USB OTG on the microcontroller and send a single byte representing a classification to the USB HOST (PC).	The host PC receives the byte representing classification and displays the action attached to the classification byte.
Upload ML model to microcontroller	Using the STM32CubeIDE upload the ML model to the microcontroller via SWD. Run the built-in validation on STM32CubeIDE with the “test on device” setting enabled.	The validation output will show classifications of the validation data with a general accuracy of >80%.
ML classification test	Upload the full firmware to the microcontroller. Allow the firmware to run in cycles for at least 3 minutes, observing no classification output. Hook the user up to the electrodes. Instruct the user to perform one of the movements within the classification set.	The microcontroller outputs the movement the user performed. This should happen multiple times.
Microcontroller power and frequency tuning	Connect the battery to the DMM to measure current. Hook up all subsystems as normal. Experiment with flashing different set clock speeds on the microcontroller to get faster classifications. Find a sweet spot with classification speed and power consumption.	Classifications that have less than one second of latency and a power consumption that leaves us with 6 hours of battery life.
Total System Validation	Follow the procedure listed in Figure 43 below.	Anticipated result is seen in the green “Success Criteria” box within Figure 43 below.
Manufacture Final PCB	Combine all 3 subsystems into a single PCB design. Take the issues learned while debugging and integrating and apply them to this final PCB. To put simply, the wires	Final PCB that can be mounted within an enclosure.

	in the wiring diagram will become internal traces.	
Final Integration Test	Follow the procedures listed in Figure 43 below for the final PCB.	Anticipated result is seen in the green “Success Criteria” box within Figure 43 below.

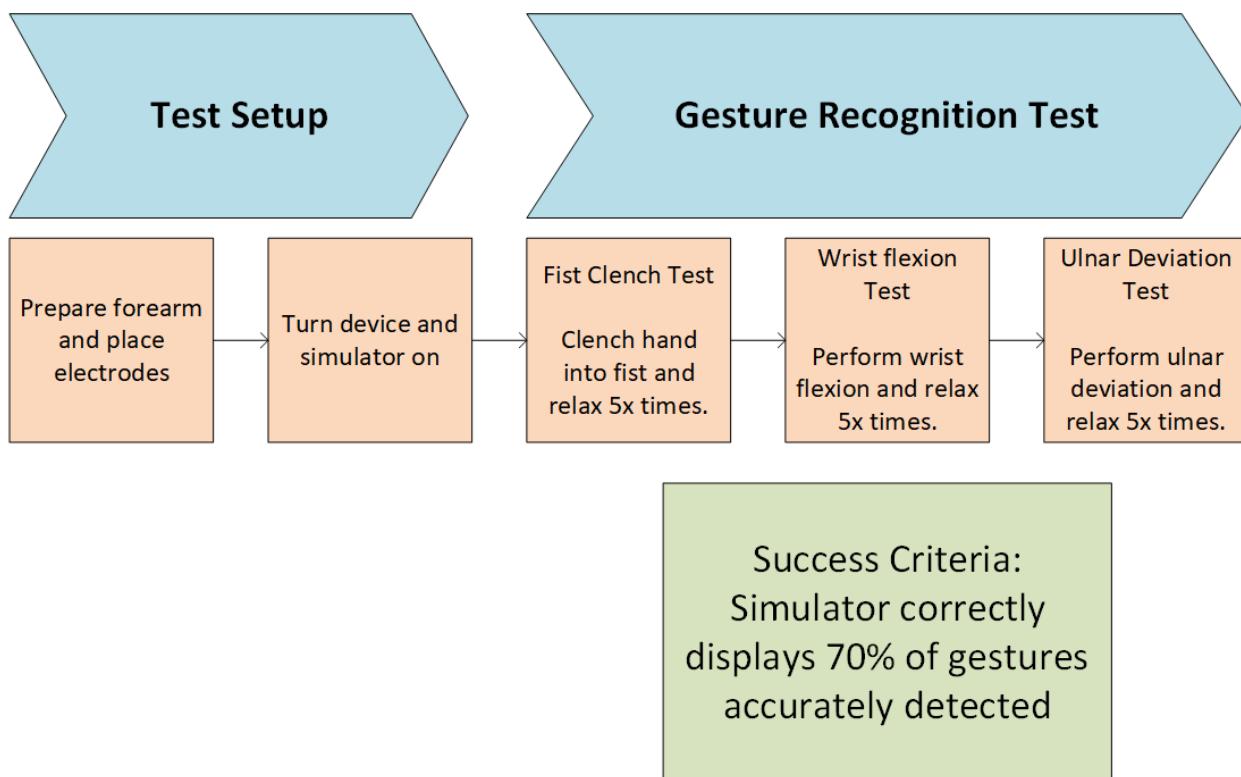


Figure 43: Total Integration Test/Success Criteria

Figure 43 above shows the total integration test that will be performed ultimately on the device as a definition of done.