EXTENDING IMU DEGREES OF FREEDOM FOR POSE ESTIMATION USING AI ON CHIP

By

Chirag Rastogi (chiragr2)

Lukas Zscherpel (lukasez2)

Final Report for ECE 445, Senior Design, Spring 2023

TA: Yixuan Wang

3, May, 2023

Project No. 71

Abstract

Inertial Measurement Units (IMUs), vital in various applications like navigation, robotics, aerospace, and the medical field, require high accuracy to ensure the reliability of the data they provide. However, the cost of high-precision IMUs often poses limitations for projects necessitating multiple devices. To address this challenge, our research project explores methods to enhance the accuracy of affordable IMUs and develops a high-accuracy pose estimation device.

Our PCB Development Board facilitates easy testing of different accuracy enhancement methods, allowing various IMUs to be connected, calibrated, and their outputs read on any connected computer for accuracy testing. Traditional algorithms known for increasing IMU accuracy, were employed to minimize errors in the data.

The project introduced an innovative approach to improve IMU performance by leveraging a Long Short-Term Memory (LSTM) model. This project paves the way for a cost-effective, high-accuracy IMU solution, potentially helping numerous applications reliant on precise motion data.

Contents

1. Introduction

1.1 The Problem

IMU stands for Inertial Measurement Unit, which is a sensor module used to measure the orientation, velocity, and acceleration of an object. High levels of accuracy are crucial in IMUs because they provide precise and reliable data about the object's motion, which is essential in many applications such as navigation, robotics, aerospace, and the medical field. IMU's typically contain an accelerometer, a gyroscope, and sometimes a magnetometer. When an IMU contains only an accelerometer and gyroscope it is labeled as a 6 Degree of Freedom (DOF) IMU, an IMU with all three sensors is labeled a 9 DOF IMU. This is because each sensor offers 3 degrees of freedom, one for each of the X, Y and Z axis.

The IMU sensors are:

- Accelerometers: Used to measure linear acceleration in three dimensions. This information can be used to estimate the velocity and position of the object over time.
- Gyroscopes: Used to measure angular velocity in three dimensions. This information can be used to estimate the orientation of the object over time.
- Magnetometers: Used to measure the direction of the Earth's magnetic field. This information can be used to determine the orientation of the object with respect to the Earth's magnetic field, which can be used to correct errors in the orientation estimate obtained from the gyroscopes.

IMU sensors that contain all 3 sensors can be found for as low as $10-$20 for basic models, but these sensors have lower accuracy. For projects that require greater accuracy, the cost can go up to $300 (https://x-io.co.uk/ngimu/) and this limits projects that require multiple such devices. High levels of accuracy in IMUs are essential to ensure the reliability and precision of the data they provide. Even small errors in the IMU measurements can accumulate over time and lead to significant errors in the estimated position, velocity, or orientation of the object, which can cause problems in the applications that rely on this data.

1.2 The Solution

Our project is formatted as a research project where we explore different methods of increasing the accuracy of the IMU data with the intent to use the most efficient method as the basis to create an affordable high accuracy pose estimation device.

Our PCB Dev Board offers a simple procedure to easily test different methods of accuracy enhancement. The PCB we have created allows for different IMU's to be connected and calibrated with simple instructions. Once an IMU is connected and calibrated the user can easily read the output on any connected computer to test accuracy enhancement methods that they design in software.

The standard way of increasing accuracy of IMU sensor readings is through three traditional algorithms. These algorithms are the Kalman Filter, Madgwick Filter, and Mahony Filter and they can increase the accuracy of IMU sensor readings in a cheap and efficient way by combining measurements from multiple sensors and minimizing errors and noise in the sensor data.

The Kalman filter is a widely used algorithm for sensor fusion and data smoothing. It is based on a mathematical model that estimates the state of a system based on its previous state and the measurements obtained from different sensors. The Kalman filter can effectively filter out noise and errors in the sensor data, resulting in more accurate readings.

The Madgwick filter and Mahony filter are two algorithms that are commonly used for sensor fusion in low-cost IMUs. These filters are based on an adaptive algorithm that uses gradient descent to estimate the orientation of the sensor based on the measurements from the accelerometer, gyroscope, and magnetometer sensors. They are computationally efficient and require less processing power compared to the Kalman filter.

By combining the measurements from multiple sensors, these algorithms can compensate for the weaknesses and limitations of individual sensors and provide more accurate and reliable data. Additionally, these algorithms can be implemented on low-cost microcontrollers, making them a cost-effective solution for improving the accuracy of IMU sensor readings.

An AI on chip solution may have the potential to reduce the cost of 9DOF IMU sensors by enabling the integration of multiple sensors and processing functions onto a single chip, which can simplify the design, reduce the bill of materials, and lower the manufacturing costs. By leveraging AI algorithms among others, an AI on chip can enable 9DOF IMU sensors to perform advanced sensing and processing tasks on-device, reducing the data transmission requirements and minimizing the need for external computing resources.

In this project we tested multiple AI models but eventually eventually settled on an LSTM implementation for the best results. An LSTM (Long Short-Term Memory) model is a type of recurrent neural network (RNN) that is particularly useful for processing and predicting sequences of data, such as time series sensor data from an IMU. An LSTM model can learn the patterns and relationships in the sensor data over time, and use this knowledge to make more accurate predictions of the sensor readings. One of the advantages of using an LSTM model is that it can learn from large datasets of sensor data without requiring a lot of manual tuning or feature engineering. This can save time and cost in the development process, as there is no need to manually select or engineer features from the sensor data. Additionally, an LSTM model can be trained on low-cost hardware such as GPUs or even on cloud-based platforms. Another advantage of using an LSTM model is that it can adapt to changing conditions and new data over time. As more data is collected from the IMU sensors, the LSTM model can continue to learn and improve its accuracy.

2 Design

For this project we split up the design of the components into four sections. There is the power supply, the IMU, the control unit, and the data processing unit. The power supply is responsible for regulating and delivering power to the other components. The IMU is responsible for creating measurements on linear and angular acceleration. The control unit is responsible for driving the IMU and providing an interface between the user, the IMU, and the connected computer. The data processing unit is responsible for processing the outputs of the IMU using different algorithms as well as displaying the processed data in real time through attitude estimation.

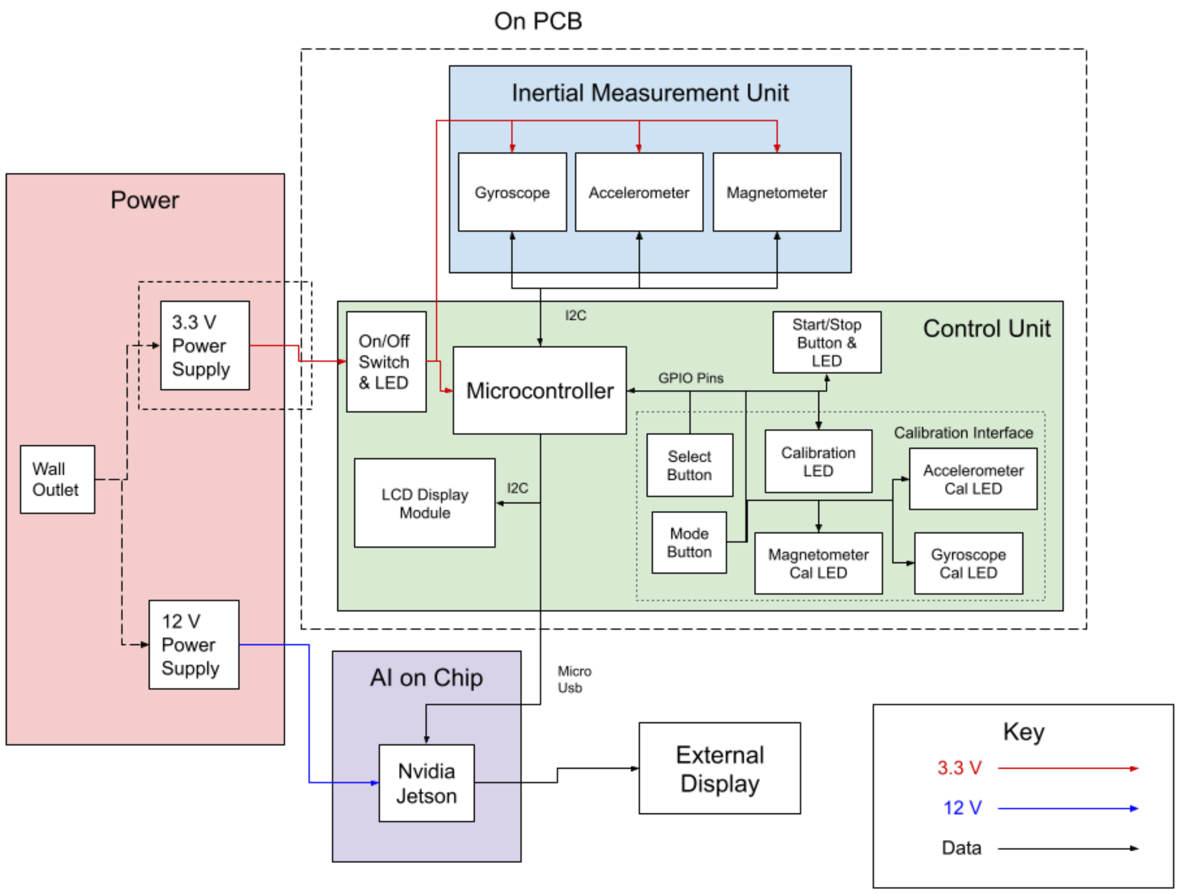

Figure 1: Block Diagram

2.1 Inertial Measurement Unit Design

This subsystem consists of only a 6 DOF or 9 DOF IMU that we acquire from a third party distributor. We chose to test the MPU-6050 and MPU-9250 for this project as they are some of the most commonly found IMU's. The MPU-6050 and MPU-9250 are relatively similar IMU's with a couple key differences. They both have onboard temperature sensors which can be used when calibrating the data output. They also both contain a Digital Motion Processor (DMP) which correlates the data from the sensors. The differences between the two IMU's is that the MPU-9250 has 9 DOF while the MPU-6050 has 6 DOF. The MPU-9250 also claims to have more accuracy within its sensors when compared to the MPU-6050. While both IMU's are relatively low cost, inside the United States the MPU-9250 can be found for a price ranging from around $7 to $20 while the MPU-6050 can be found for a price ranging from around $4 to $10.

Both IMU's require very little power. They both run on 3.3 volts, but they accept an input voltage in a range between 3 and 5 volts as they contain internal voltage regulator systems. Both IMU's also draw very little current and require no more than 5mA of current which can be easily supplied by the power supply subsystem.

We connected the IMU to the PCB using a 8 pin female connector. This allowed us to quickly and easily swap between IMU's when testing. The female pin header is connected to the microcontroller allowing it to get values from the accelerometer, gyroscope and the magnetometer if available. The communication will be done through the I2C protocol as it is supported by both IMU's as well as the microcontroller. The IMU's don't have a built in way of calibrating their sensors, so the calibration and application of offsets is done by the microcontroller. The IMU simply passes on the raw data it collects from its sensors. The IMU's do have multiple different power modes depending on the sensors operating speeds which we tested while programming our microcontroller. We ended up reading data from the IMU at about 450 Hz.

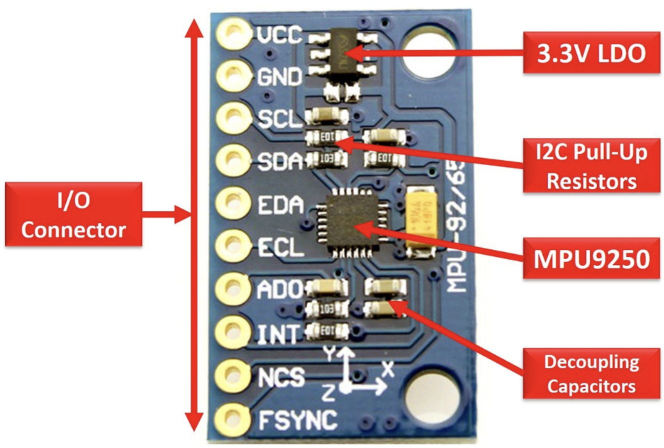

Figure 2: MPU 9250 Breakout Board

2.2 Control Unit Design

The Control Unit is a critical component of our design, tasked with interfacing with the IMU and directing the raw data it outputs. Communication between the Control Unit and IMU is established using the I2C data protocol, and the Control Unit is also responsible for calibrating the IMU. Additionally, the microcontroller is connected to a USB port for interfacing with the NVidia Jetson or whatever computer is connected. The Control Unit, as the central entity for data flow management, enables us to incorporate user interface components such as a button for activating and deactivating data flow, buttons for calibrating the IMU to determine the error size, and a small OLED display screen to convey instructions and information to the user.

The control unit subsystem consists of an ESP32 as the microcontroller, a 128x64 OLED display screen, 6 LED's of various colors, 3 push buttons, one 2 position toggle switch, a MCP2200 UART to USB chip, and a USB interface. The ESP32 has two I2C bus modules, one of which will be used to communicate with the IMU subsystem and the other to control the OLED display. The ESP32 also has 36 general purpose I/O pins (GPIO) which is plenty to connect with any switches, buttons or LED's that we will use. The ESP32 has 520kB of SRAM and 2MB of on board flash memory which is plenty for storing the code for the program and data that we collect temporarily during the calibration process. The RP2040 also runs at 160MHz which is fast enough to collect the data from the IMU without any losses.

The 128x64 OLED display screen is a small cheap monochrome OLED screen that can display graphics and text. The 128x64 OLED display has a built in I2C module which makes it simple to connect to the ESP32, it is also supported by many open source code libraries which makes programming the ESP32 to use the display straightforward. We used the display to communicate with the user what mode the mode the microcontroller is in, to display the data when the microcontroller is transmitting data from the IMU, and to give instructions to the user when the microcontroller is calibrating the IMU.

To program the microcontroller we used the arduino ide. The program we developed has three main stages. There is the transmit data stage, the idle stage and the calibration stage. To cycle between the different states the user presses different buttons on the PCB. The specific states and button interactions can be seen in the control flow diagram below (Figure 3). If at any point in time the user wants to reset the calibration process they just need to press the Cal_Mode button. The labels on the buttons are different than their intentions because we changed the specific functionality of the buttons over time, but they still serve purposes similar to their names.

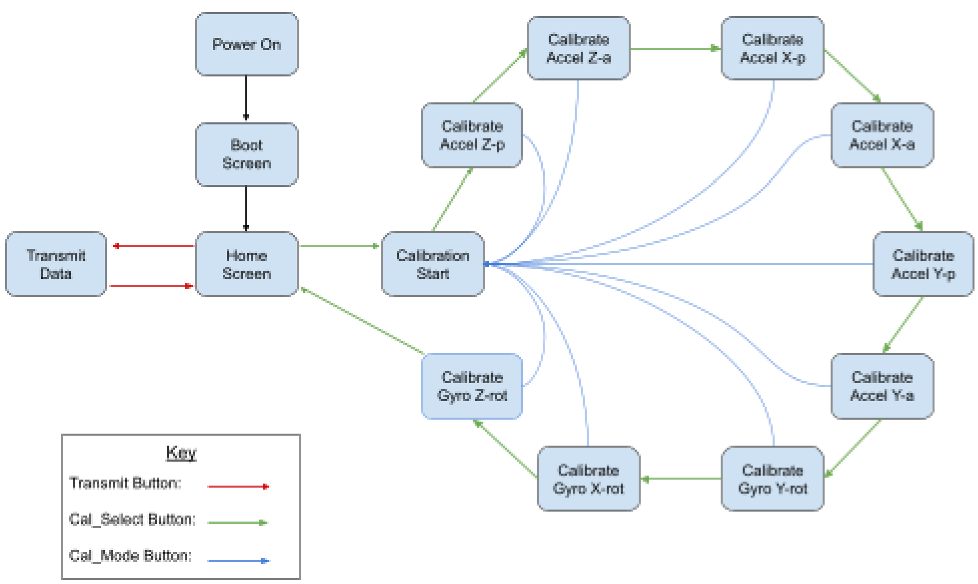

Figure 3: Control Flow

Connected to one of the ESP32's GPIO pins is a power on/off switch which controls whether the control unit and IMU subsystems will be powered on or off. This switch will be connected to the 3V3_EN port of the ESP32. In conjunction with the power switch is an LED which is turned on when power to the rest of the PCB is enabled. This helps give visual information to the user letting them know that the ESP32 has finished booting up and is ready to run.

In our original design for the control unit, the calibration of the control unit would be done on the microcontroller directly, but after advice from the professor we shifted the calibration data processing onto an external computer. This allowed us to use a standard calibration library to calibrate each of the IMU sensors. The calibration library we used is the IMUcal library available in python.

We implement a simple calibration routine for the accelerometer using gravity. We instruct the user to place the PCB equipped with the IMU on a level surface for all 6 faces and wait a few seconds and read each Degree of Freedom (DOF). Once all data has been recorded, the microcontroller calculates the average deviation of the data from the expected value at each different rotation to create an offset matrix. The results of the accelerometer calibration can be seen below.

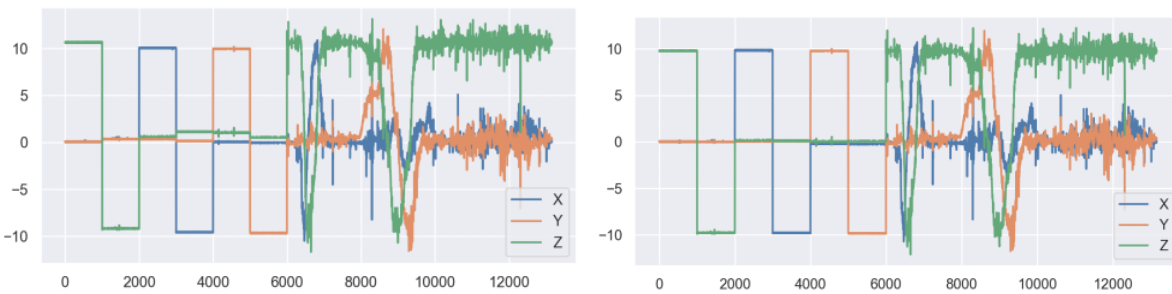

Figure 4: Accelerometer Calibration

We implement a simple calibration routine for the gyroscope. We instruct the user to place the Dev Board up against a vertical flat surface like the edge of a desk so that the gyroscope will read zero for each axis. Then we instruct the user to rotate the Dev Board in a clockwise direction 360 degrees around each of the axis one by one then place the Dev Board back up against the flat surface. This allows us to collect data from the gyroscope while having a ground truth to compare it against because we know it should only read that it was rotated 360 degrees. Using this knowledge the IMUcal library calculates and stores it within a calibration matrix to calibrate future data. An example of the gyroscope calibration can be seen in figure 4 below.

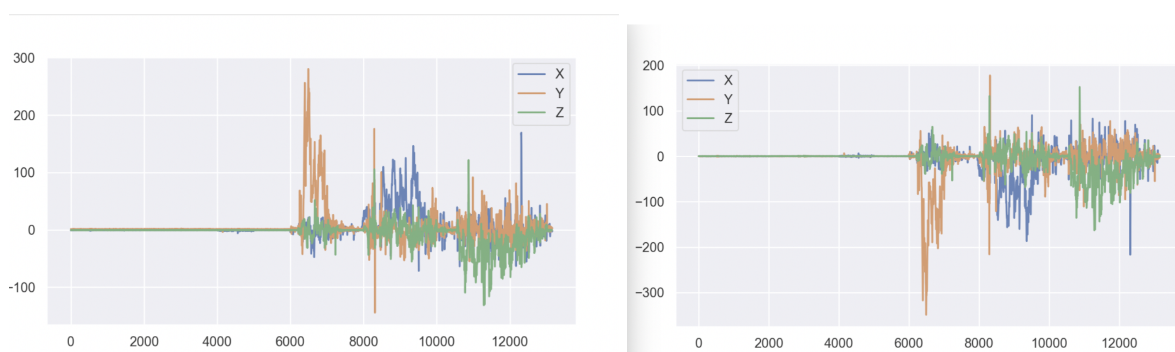

Figure 4: Gyroscope Calibration

2.3 Position Estimation using AI on Chip Design

Our project integrates an AI system for accurate pose estimation. This system is designed to process the output data from an Inertial Measurement Unit (IMU), subsequently predicting the device's orientation. The architecture of this system employs an LSTM model, and we explored several other algorithms, including Extended Kalman Filtering, Digital Motion Processing, Madgwick Filter, and Mahony Filter.

We observed that the Kalman Filter, Madgwick Filter, and Mahony Filter performed better than the other classical algorithms and used it to establish a baseline.

The data utilized in this project, as outlined in the IEEE Dataport study linked https://ieee-dataport.org/open-access/estimating-relative-angle-between-two-6-axis-inertial-measurement-units-imus, is harvested from two IMUs, specifically the MPU-6050 models.

In our initial analysis, we used the quaternion values, provided by the author of the dataset and computed by the Digital Motion Processor, to derive the pitch angle. We observed that the error exponentially increases after 10 mins.
Therefore we used other standard algorithms and Deep Learning algorithms that could offer a more accurate estimation.

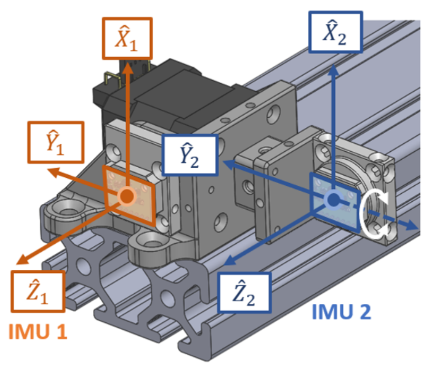

Figure 5: IMU setup for data collection

In the experimental design, we used data carried out on nine trials, each with a duration of 25 minutes.In this section, we will provide a more in-depth analysis of our results in conjunction with the algorithms employed in the project. All experiments were performed on a CPU, which read and processed IMU readings from the serial port at approximately 400Hz. As anticipated, the rate decreased with additional computation.

Following calibration, the rate dropped to 330Hz. We utilized the standard AHRS library for all the filters.

Mahony Filter:

- The Mahony Filter achieved the highest rate at 160Hz and an RMSE of 0.65 Rads or 37.2423 degrees. Despite being the fastest algorithm, it exhibited the highest error rate.

Madgwick Filter:

- The Madgwick Filter reached a rate of 125Hz and an RMSE of 0.612 Rads or 35.065 degrees. Among the standard algorithms, this filter offered the best balance between a relatively high rate and a comparable error.

Extended Kalman Filter:

- The Kalman Filter had the lowest rate at 40Hz and an RMSE of 0.5115 Rads or 29.3068 degrees. The error rate was as low as 3 degrees for the initial 10 minutes; however, it significantly increased afterward. Although this filter demonstrated the best performance among the three standard algorithms, the error rate remains notably high.

Deep Learning Architecture

- Our deep learning approach involved deploying an LSTM model consisting of 64 layers and a single dense layer. We experimented with various architectures, including Bidirectional LSTMs. However, we observed that models with more than 64 layers consistently overfit the data, leading to unsatisfactory results in other experiments.
- The implemented 64-layer LSTM model achieved a rate of 110Hz, which completed our high-level requirement, and demonstrated an impressively low RMSE of 0.1218 degrees. This exceptional performance indicated that our deep learning architecture effectively addressed the orientation estimation problem and delivered superior results compared to the standard algorithms.

2.4 Power Supply Design

The purpose of the power supply subsystem is to deliver power to the other components. In our original design we had planned for power to come from a DC power supply wall adapter and be controlled by a voltage regulator on the PCB. After deliberation and some design changes we decided to utilize the 5 volts that get supplied by the micro-usb port and use the original voltage regulator to reduce the 5 volt input to 3.3 volts. We decided on regulating the voltage down to 3.3 volts because that is the optimal operating voltage for all of the components on the PCB, even though some were able to operate on the 5 volt input directly, not all of the parts could.

The micro-usb port that we used was the MOLEX 1050170001. It is a standard micro-usb port with VCC, D-, D+, and GND connections to the PCB. When connected to a standard computer or power supply, the VCC supplies a voltage of 5V. We connected the VCC of the micro-usb port directly to the VIN of the voltage regulator. The D- and D+ connections transmit data to and from the PCB using standard USB protocol. They are connected to the D- and D+ pins of the USB to UART chip respectively. The GND pin of the micro-usb port is connected to the ground of every other component on the PCB, it grounds each component to the ground terminal of the connected computer or power supply. The circuit schematic of the micro-usb port can be seen in figure x in appendix B.

The 5V to 3.3V voltage regulator we used was the AMS 1117-3.3. It is a very common voltage regulator that reduces our input voltage of 5V down to 3.3V. The AMS 1117-3.3 has a maximum input voltage of

15V and a maximum output voltage of 0.8 amps. When operating in an ambient temperature of 25 degrees celsius the voltage regulator can dissipate up to 1.5 watts of power. The input to the voltage regulator will be the 5V output voltage of the micro-usb connection. The output voltage will then go to power the ESP32 microcontroller, the USB to UART IC, the OLED display, and the connected IMU. The circuit schematic of the voltage regulator can be seen in figure x in appendix B.

A USB 2.0 port can supply 500mA of current at 5V. This is enough to power everything on the PCB, including the ESP32, the IMU, and the OLED display. In standard mode the ESP32 consumes about 260mA of current which leaves plenty for the other components. The IMU does not consume more than 5mA of current and the OLED display does not draw more than 50mA of current when the display is fully illuminated. These values allow us to power the entirety of the PCB without requiring us to draw more power from other sources.

3. Design Verification

3.1 Inertial Measurement Unit Verification

The important criterias that the inertial measurement unit must stand up to are that it is able to read positional data, that the sensors have an accuracy similar to the manufacturers specifications, and that it is able to transmit data without error and transmit data fast enough for us to display a real time position estimation of the device.

Reading data from the IMU is relatively straightforward. There are multiple libraries available for both micropython and arduino that are able to streamline the process of communicating with the connected IMU. For the MPU 9250 we went with the MPU9250 by hideakitai library available on github: https://github.com/hideakitai/MPU9250. For the MPU 6050 we used adafruit's standard IMU communication library. To test that these libraries worked and we were able to read sensor data we simply displayed the data on the OLED display using a test program.

Verification of the accuracy of the sensors wasn't as straightforward. In the datasheets the manufacturer lists different rates of error for each sensor, and each sensor is prone to different rates of errors depending on many factors such as input voltage or even ambient temperature. To test the accuracy of each sensor we first calibrate the IMU, then we put the IMU in a stable position and then collect data from the IMU for 30 minutes. We then took the raw sensor sensor readings and used them to calculate the position and orientation of the IMU. We calculate the error of each sensor by seeing how much our estimation of the position of the IMU changed its orientation and position over time. Based on our tests the accelerometer on the MPU 9250 had an error rate of about 5.1 degrees per hour, the accelerometer on the MPU 6050 had an error rate of about 5.7 degrees per hour, the gyroscope on the MPU 9250 had an error rate of about 5.3 degrees per hour, and the gyroscope on the MPU 6050 had an error rate of about 5.8 degrees per hour.

Ensuring proper communication between the IMU and microcontroller is important because without a stable connection, unforeseen errors or problems can cause massive repercussions in our position estimation. To test the connection between the IMU and microcontroller we sampled the IMU at a rate of 100 Hz for 20 minutes and we recorded an error whenever the readings of the IMU sensors failed to update. After those twenty minutes we did not see even one error while reading the gyroscope or accelerometer of both the MPU 6050 and MPU 9250. This showed that not only can the IMU operate at a minimum speed of 100 Hz, but also that the communication with the IMU was stable.

3.2 Control Unit Verification

The control unit is not only the microcontroller that dictates what all the other components are supposed to do, but it is also the user interface components such as the LEDs and OLED display. As such verifying that the control unit performs to our expectations is an important part of the project. The control unit must be able to transmit data quickly and accurately from the IMU to the connected USB port, it must guide the user through the calibration process, and it must be intuitive to use.

We tested the processing and communication abilities of the microcontroller by testing how fast it could read from the IMU and how fast it could transmit data out through the USB port. Simply by recording the data collected by the microcontroller over a 10 minute period of time we verified that the ESP32 could read data from the connected IMU at a rate of at least 450 Hz. We believe that the ESP32 could communicate even faster over the I2C protocol, but the IMU would not report data at speeds faster than that so we didn't verify that belief. Faster speeds are unnecessary since the ability to read data faster than the sensors can record the data is pointless. To test the USB communication capabilities of the microcontroller and USB to UART IC we selected the maximum Baud rate that the arduino serial library would allow and we sent 10,000 lines of data over the connection. The maximum baud rate was 250000, using this baud rate we successfully were able to read the 10,000 lines of data with no error.

We tested the calibration process by calibrating the same IMU multiple times to ensure that the calibration offsets were consistent over multiple attempts. The output calibration offsets were then compared to calculate the difference, if any, between the calibration attempts. After testing each IMU five times we saw very consistent results of the calibration process with the difference between the calibration attempts being no more than 0.1 degrees. Through these tests we were also able to verify that the calibration process centered the sensor reading around 0 within a tolerance range of +/- 0.1 degrees.

The last aspect of the control unit that we verified was the user interface. We verified that it was intuitive and simple to use by letting multiple of our classmates and friends test the system and let us know any ways that the user experience could be enhanced. Because we did not have the time to redesign the layout of the PCB we were limited to enhancing the UI through the visuals on the OLED display and the instructions we gave. After letting 10 different people test our Dev Board we got approval from each of them saying it was simple and straightforward to use, and they liked the visuals of the arrows on the OLED display to help during the calibration process.

3.3 Position Estimation using AI on Chip Verification

Our deep learning model distinctly surpassed the standard models in performance. The established 64-layer LSTM model clocked a rate of 110Hz, showcasing an impressively low RMSE of 0.1218 degrees - a notable improvement from the 6 degrees recorded by other models.

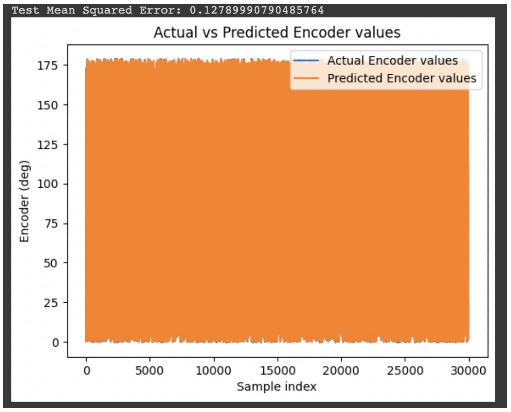

Figure 6: RMSE obtained from LSTM with the predicted values

Moreover, the calculated Mean Absolute Percentage Error was below 5% over a 20-minute data collection period. The output produced by the model was not only comprehensible but also held up when subjected to unseen test data from other trials. Our model was trained on medium-rate encoders (10˚/s), and we tested its prediction capabilities on both fast (200˚/s) and slow (20˚/s) encoder rate experiments.

We presented the final output in a 3D model displaying the Pitch, Yaw, and Roll, which significantly enhanced understanding. It was quite straightforward to visualize the IMU's movement in this manner.

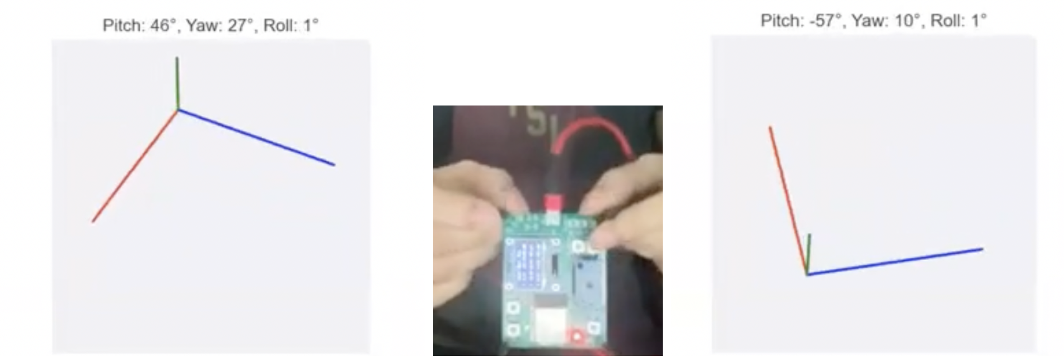

Figure 7: 3D pose model with live updates from IMU

In addition, we designed the IMU to showcase the Euler angles on the OLED screen mounted on the PCB, adding another layer of accessible data representation.

3.4 Power Supply Verification

The power supply must meet strict requirements to ensure the safety of the people interacting with the PCB and the safety of the parts soldered to the PCB. To verify the safety and functionality of the power supply system we ran multiple vigorous tests even taking the voltage regulator beyond its breaking point on a few occasions.

To verify that the output voltage of the voltage regulator remained consistent at 3.3V we ran the PCB under load with the microcontroller looping an animation over the OLED display while reading values from the IMU at the same time. We performed this test for 10 minutes to make sure that the output voltage is also consistent over time. We measured the output current of the voltage regulator to be stable at about 410 mA and the output voltage to be stable at 3.29 V.

Another important requirement for the power supply system is that it is able to handle voltage spikes or input voltages above or below 5V. We tested this by connecting the voltage regulator to a breadboard and using an accurate power supply and multimeter to test the voltage going in and out of the voltage regulator. After testing we found the maximum input voltage to the voltage regulator to be just under 14 volts, the minimum input voltage to be 4.8 volts for a consistent 3.3V output, and as long as the input voltage was between those values a sudden voltage spike would not affect the output voltage of the voltage regulator.

Our last requirement of the voltage regulator is that it does not heat up or consume power beyond reason. By measuring the input voltage and current as well as the output voltage and current we can calculate the efficiency of the voltage regulator. The PCB drew about 410 mA with an input voltage to the regulator being a constant 5V and the output voltage being a constant 3.29V. At these values the efficiency of the voltage regulator was 66% which was within our expectations.

4. Costs

4.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Retail Cost ($)	Number Needed	Bulk/Total Cost ($)
Capacitor 10uF	Samsung Electro-Mechanics	0.0112	2	0.56
Capacitor 100nF	Samsung Electro-Mechanics	0.0010	9	0.10
Capacitor 4.7uF	Samsung Electro-Mechanics	0.0072	1	0.72
Capacitor 22uF	Samsung Electro-Mechanics	0.0258	1	0.52
BiDirectional Diode	Leshan Radio	0.0157	3	0.79
Micro USB Adapter	MOLEX	0.2739	1	1.37
2 Pin Male	DEALON(德艺隆)	0.0132	1	0.66
8 Pin Female	ZHOURI(洲日)	0.0918	1	0.46
LED Green	FOSHAN	0.0166	1	0.83
LED Red	NATIONSTAR	0.0109	5	0.55
OLED Display	HiLetgo	5.00	1	15.00
Small Diode	DIODES	0.0267	2	0.27
Big Diode	CJ(江苏长电/长晶)	0.0197	2	0.39
Resistor 5.1kΩ	UniOhm	0.0005	1	0.05
Resistor 560Ω	UniOhm	0.0006	5	0.06
Resistor 1kΩ	UniOhm	0.0005	2	0.05
Resistor 0Ω	UniOhm	0.0005	2	0.05
Resistor 10kΩ	UniOhm	0.0006	8	0.06
Resistor 22.1kΩ	UniOhm	0.0005	1	0.05
Resistor 47.5kΩ	UniOhm	0.0006	1	0.06
Resistor 22Ω	UniOhm	0.0005	2	0.05
PushButton	C&K	0.1290	5	0.65
Voltage Regulator: 5v-3.3v	Youtai Semiconductor Co., Ltd.	0.0380	1	0.38
USB to UART	SILICON LABS(芯科)	2.50	1	2.50
ESP32-WROOM-32E-N8	Espressif Systems	3.2519	1	6.50
Custom PCB	PCBWay	1.00	1	5.00
Total		**$12.44**	**$13.10**	**$37.68**

4.2 Labor

If we assume that we both earn $40 an hour, that we work 8 hours a week, that we have been working for the past 7 weeks, we multiply that by 2.5, and that there are 2 people working on this project then the labor cost would be $40 * 8 * 7 * 2.5 * 2 = \$10,200$.

If we take the total labor cost of $10,200 and add it to the total parts cost from table 1 of $37.68 we get the total cost for this project at $10,237.68.

5. Conclusion

5.1 Accomplishments

We developed a pcb with efficient power delivery to all components. To accomplish this we decided to leverage the 5 volts supplied by the micro-USB port and use the voltage regulator to step this down to 3.3 volts. In terms of calibration, we have successfully implemented straightforward and effective methods for both the accelerometer and the gyroscope on the PCB itself. We have also successfully developed an efficient AI system that provides accurate pose estimation. Our system, using a 64-layer LSTM model, has demonstrated significant improvements over standard algorithms. We managed to achieve a robust processing rate of 110Hz and an impressively low Root Mean Square Error (RMSE) of 0.1218 degrees, far surpassing the performance of conventional models. Our PCB with the IMU maintained a Mean Absolute Percentage Error below 5% over a 20-minute period, showcasing its consistency and reliability. We also introduced user-friendly data visualization through a 3D model and an OLED screen on the PCB, making the system's output easy to interpret.

These accomplishments demonstrate our ability to optimize both hardware and software aspects of our system, resulting in a robust and efficient pose estimation on an IMU-based AI system.

5.2 Uncertainties

The deep learning model's real-world performance is a considerable area of uncertainty. While our LSTM model demonstrated impressive results on our test data, its performance in more varied or challenging environments is yet to be determined. Furthermore, the system's robustness in handling noise, outliers, or corrupted data remains an open question. From a hardware perspective, the durability and reliability of the micro-USB port and the voltage regulator under different usage conditions, including extreme temperatures or power fluctuations, are not entirely known.

5.3 Ethical and safety considerations

In accordance with the IEEE Code of Ethics, it is of utmost importance to maintain originality and integrity in the project ideas and research process. Any sources used during the research must be properly cited and credited to avoid plagiarism (IEEE Code of Ethics II.5). Our project is aligned with the ongoing efforts in improving the accuracy of IMU sensors, and while referencing relevant research papers, all sources used will be properly cited and credited. Our project aims to differentiate itself from existing technologies by utilizing unique methods of implementation. Furthermore, the IEEE Code of Ethics I.5 requires that all claims and estimates be honest and realistic. In the context of our project, we strive to enhance the precision of IMU data output to the best of our abilities, which includes verifying the reliability of raw data from various IMU sensors.

In terms of safety, our team is committed to following the laboratory safety regulations set by the Division of Research Safety in the Office of the Vice Chancellor for Research and Innovation (ECE 445 p.3). To minimize any potential risks, our team will work in pairs during laboratory sessions, promptly report any broken equipment, maintain cleanliness after each session, and avoid consuming food within the lab.The main concern of safety within out project is that of a short circuit within our PCB or power supply that cause elements of the electrical circuit to overheat and burn the users of the project. To mitigate this risk we will exercise caution to avoid skin contact with electrical circuits and separate electrical systems with high voltage requirements from human interaction. We will also thoroughly test our circuits against varying voltages and currents to make sure that a power surge will not cause any potential damage or safety hazards within our project.

5.4 Future work

Further research and development can be conducted to enhance the performance, accuracy, and cost-effectiveness of the IMU pose estimation device. Some potential avenues for future work include:

Developing custom hardware solutions or System-on-Chip (SoC) designs to optimize the processing capabilities and power efficiency of the IMU system, allowing for smaller form factors and reduced power consumption.

Conducting extensive testing and validation of the IMU system in various application scenarios, such as autonomous vehicles, robotics, virtual reality, or medical devices, to evaluate its performance, robustness, and reliability under diverse conditions.

Implementing real-time calibration techniques with the LSTM model to continuously improve the accuracy and reliability of the IMU system during its operation

References

[1] X-IO Technologies. "NGIMU: The Complete Sensor Fusion Solution." X-IO Technologies, www.x-io.co.uk/ngimu/.

[2] X-IO Technologies. "NGIMU: Sensor Fusion with IMU & AHRS." YouTube, uploaded by X-IO Technologies, 9 Apr. 2018, www.youtube.com/watch?v=a2wOjxRf_xg&list=PLJePd8QU_LYKZwJnByZ8FHDg5l1rXtcIq&index=2.

[3] "IEEE Code of Ethics." IEEE, www.ieee.org/about/corporate/governance/p7-8.html.

[4] Association for Computing Machinery. "ACM Code of Ethics." Association for Computing Machinery, www.acm.org/code-of-ethics.

[5] Kim, Hyunseok, et al. "Evaluating the Accuracy of Inertial Measurement Units: A Comparative Study on Real-World Data." arXiv, Cornell University Library, 8 May 2020, arxiv.org/pdf/2005.06897.pdf.

[6] Brossard, Marc. "Denoise-Imu-Gyro." GitHub, github.com/mbrossar/denoise-imu-gyro.

[7] Adafruit Industries, LLC. "Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055." Adafruit Industries, LLC, www.adafruit.com/product/4502.

[8] HiLetgo. "HiLetgo MPU-6050 3 Axis Accelerometer Gyroscope Module 6 DOF IMU for Arduino." Amazon, www.amazon.com/HiLetgo-MPU-6050-Accelerometer-Gyroscope-Converter/dp/B01DK83ZYQ?th=1.

[9] Hrisko, J. (2019). Accelerometer, Gyroscope, and Magnetometer Analysis with Raspberry Pi Part I: Basic Readings. Maker Portal. https://makersportal.com/blog/2019/11/11/raspberry-pi-python-accelerometer-gyroscope-magnetometer

[10] Shimmer Sensing. "Wireless Sensor Networks Videos." Shimmer Sensing, shimmersensing.com/support/wireless-sensor-networks-videos/.

[11] Adafruit. "Magnetometer Calibration." Adafruit Learning System, learn.adafruit.com/ahrs-for-adafruits-9-dof-10-dof-breakout/magnetometer-calibration.

[12] Hackster.io. "MPU9250 Datasheet." TDK InvenSense, 2015, invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf.

[13] Christopher Barnatt. "Raspberry Pi Pico Tutorial #5: Pico and Accelerometers." YouTube, uploaded by ExplainingComputers, 2 Feb. 2021, www.youtube.com/watch?v=XCyRXMvVSCw.

[14] Christopher Barnatt. "Raspberry Pi Pico Tutorial #6: Pico and Gyroscopes." YouTube, uploaded by ExplainingComputers, 6 Feb. 2021, www.youtube.com/watch?v=mzwovYcozvI.

[15] IMU Filters https://nitinjsanket.github.io/tutorials/attitudeest/mahony

Appendix ARequirement and Verification Table

Table 2 High Level Requirements

Requirement	Completion Status
1. The IMU interfacing system must be able to accurately measure and monitor the orientation and movement of a device to within a maximum error of 3 degrees and 0.5 cm/s.	Yes
2. The system must be able to perform calibration on the data outputted by the IMU to within a maximum error of 0.5 degrees and 0.2 cm/s to ensure accuracy and consistency.	Yes
3. For a given IMU the system must be able to determine the parameters for each algorithm, out of a set of predefined algorithms, that are effective at minimizing the error and noise outputted by the IMU. This would ensure a Mean absolute percentage error of less than 5% over 20 mins of data collection.	Yes

Table 3 System Requirements and Verifications

Requirement	Verification	Complete Status (Y or N)
4. The IMU must accurately measure orientation and motion data with a minimal error rate within the manufacturer's tolerance range.	1. The error rate of the IMU for measuring orientation data must be less than 3 degrees within the manufacturer's tolerance range of ±2 degrees.	1. Y
5. The IMU must operate at the same voltage level as the rest of the PCB and communicate using the I2C protocol with a low error rate.	1. The IMU must operate at 3.3V with a tolerance of 0.2V 2. The I2C communication between the IMU and the microcontroller must have an error rate of less than 1% for reliable data transfer.	1. Y 2. Y
6. The IMU must have a stable and consistent output rate of at least 100Hz	1. The output rate of the IMU must be within ±5Hz of the specified rate of 100Hz. 2. The output rate of the IMU must not fluctuate by more than ±2Hz over a 10-minute testing period.	1. Y 2. Y
7. The Control Unit must be able to process and transmit data quickly and accurately from the IMU to the device connected to the USB port.	1. The Control Unit must be able to process IMU data at a rate of at least 100Hz. 2. The Control Unit must be able to transmit data to the device connected to the usb port at a rate of at least 10Mbps. 3. The I2C communication link between the IMU and the ESP32 must have a maximum data transmission error rate of 5%.	1. Y 2. Y 3. Y
8. The control unit must perform accurate	1. The calibrated IMU data must remain stable with less than less than 1% drift (pose and position) when the IMU	1. Y

calibration of the raw IMU output data to account for any manufacturing bias and environmental factors.	is placed on a flat surface with no rotation, for a duration of 20 minutes 2. Verify that the calibrated data has a mean value of zero within a tolerance range of +/- 0.1 degrees 3. Verify that the calibration process can be repeated with consistent results within a tolerance range of +/- 0.1 degrees. 4. The calibration process must be able to calibrate each of the 3 different sensors and the data that they output. 5. Verify that the calibration process does not introduce any additional errors or noise to the IMU's output data, and that the calibrated data accurately represents the physical orientation and motion of the sensor.	2. Y 3. Y 4. N 5. Y
9. The user interface of the Control Unit must be designed in such a way that it is easy for a user to perform calibration and data collection tasks without the need for extensive technical knowledge.	1. The user interface must have clearly labeled buttons or controls that correspond to each functionality, such as "Calibrate" and "Collect Data". Each button or control must be labeled with clear and concise text or symbols that are easily understood by the user. 2. The user interface must provide clear instructions or prompts on how to initiate the calibration and data collection processes. These instructions or prompts should be displayed prominently on the interface and should be written in simple and easy-to-understand language. The user should be able to follow these instructions without any confusion or difficulty.	1. Y 2. Y
10. The Algorithm used must be able to minimize RMSE on the dataset	1. As we already know that standard models can perform with an RMSE of <6 degrees within 25 mins, the deep learning model needs to outperform these algorithms 2. The Output of the model should be easy to understand and be subjected to unseen test data taken from other trials	1. Y 2. Y
11. The Algorithm used should output the 3D rotation of the IMU	1. The Algorithm will provide us with quaternions values that we need to use to display a 3D model on a screen 2. The IMU can display the Euler angles on the LCD or OLED screen on the PCB.	1. Y 2. Y
12. The Algorithm used can minimize the error over the data	1. The calculated Mean absolute percentage error should be less than 5% over 20 mins of data collection	1. Y
13. The power supply must provide a stable output voltage of 3.3 volts to the PCB and a stable output voltage of 12 volts to the NVIDIA Jetson.	1. Measure the output voltage of the PCB power supply using a multimeter and verify that the value is 3.3 volts with a tolerance range of +/- 0.1 volts. 2. Measure the output voltage of the Jetson power supply using a multimeter and verify that the value is 12 volts with a tolerance range of +/- 0.5 volts 3. Place the PCB under varying load conditions and measure the output voltage to ensure it remains stable within the tolerance range.	1. Y 2. Y 3. Y

14. The power supply must be able to handle a maximum input voltage of 24 volts with a tolerance range of +/- 0.5 volts.	1. Increase the input voltage of the power supply in small increments while measuring the output voltage until the maximum voltage is reached 2. Verify that the power supply can handle the maximum input voltage without damaging any components or causing any safety hazards.	1. Y 2. Y
15. The power supply must have overvoltage protection to prevent any voltage spikes from damaging the components.	1. Introduce a voltage spike to the input voltage of the power supply and measure the output voltage to ensure it remains within the tolerance range. 2. Verify that the overvoltage protection activates and prevents any voltage spikes from passing through to the components.	1. Y 2. Y
16. The power supply must have short-circuit protection to prevent any damage to the components in case of a short circuit.	1. Introduce a short circuit to the output of the power supply and verify that it shuts down and prevents any current from flowing. 2. Verify that the short-circuit protection activates and prevents any damage to the components.	1. Y 2. Y
17. The power supply must have an efficiency of at least 60% to minimize power loss and heat generation.	1. Measure the input and output power of the power supply and calculate the efficiency. 2. Verify that the power supply meets the efficiency requirement under varying load conditions.	1. Y 2. Y

The only verification we did not meet was requirement 5 verification 4. We did not meet this requirement because we changed our system to use a standard IMU calibration library, but that library did not offer calibration methods for the magnetometer. This was okay though because the magnetometer did not have a big influence on the outputted eulerian angles.

Appendix B PCB Circuit Design

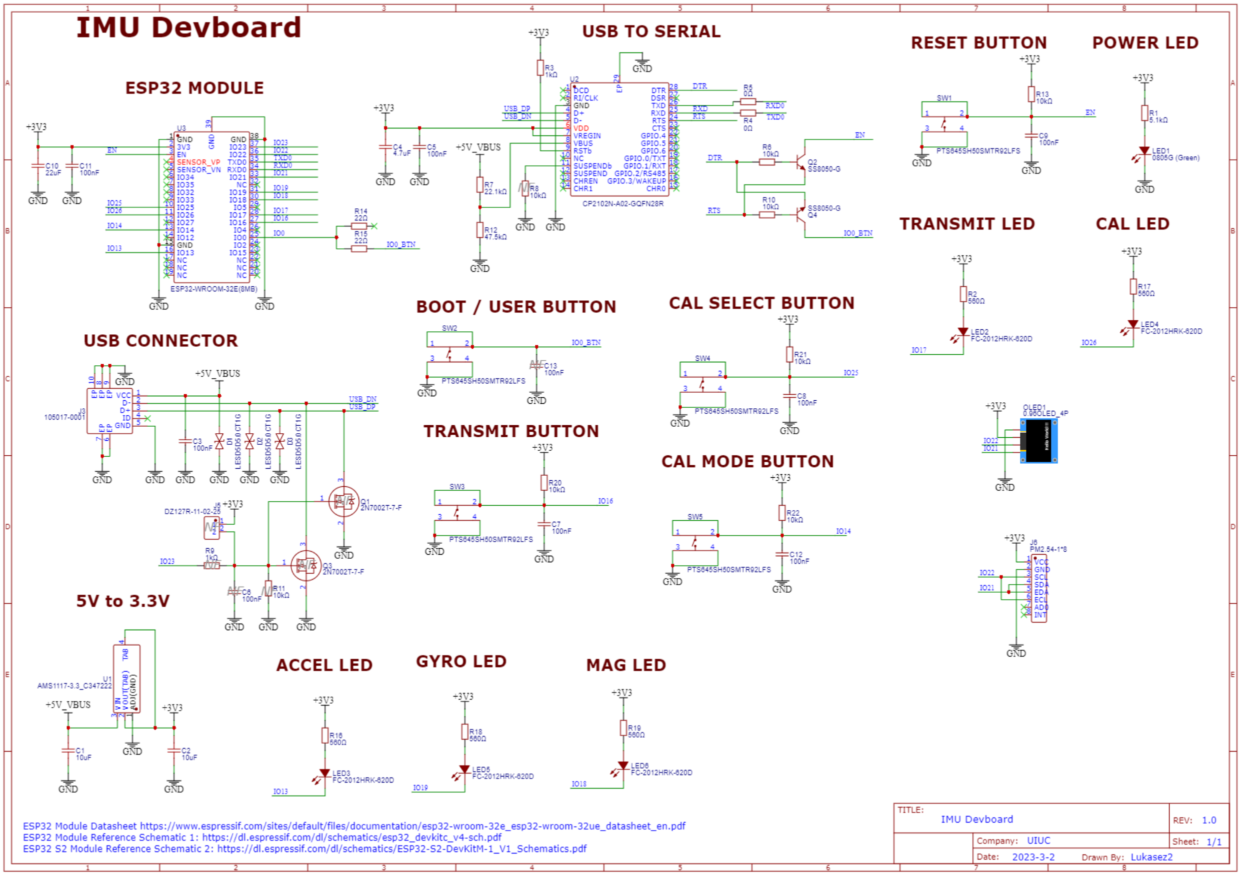

Figure 8: Complete PCB Schematic

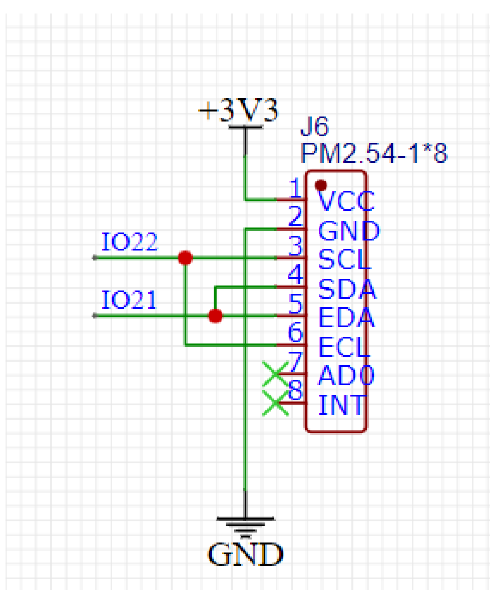

Figure 9: IMU Connection

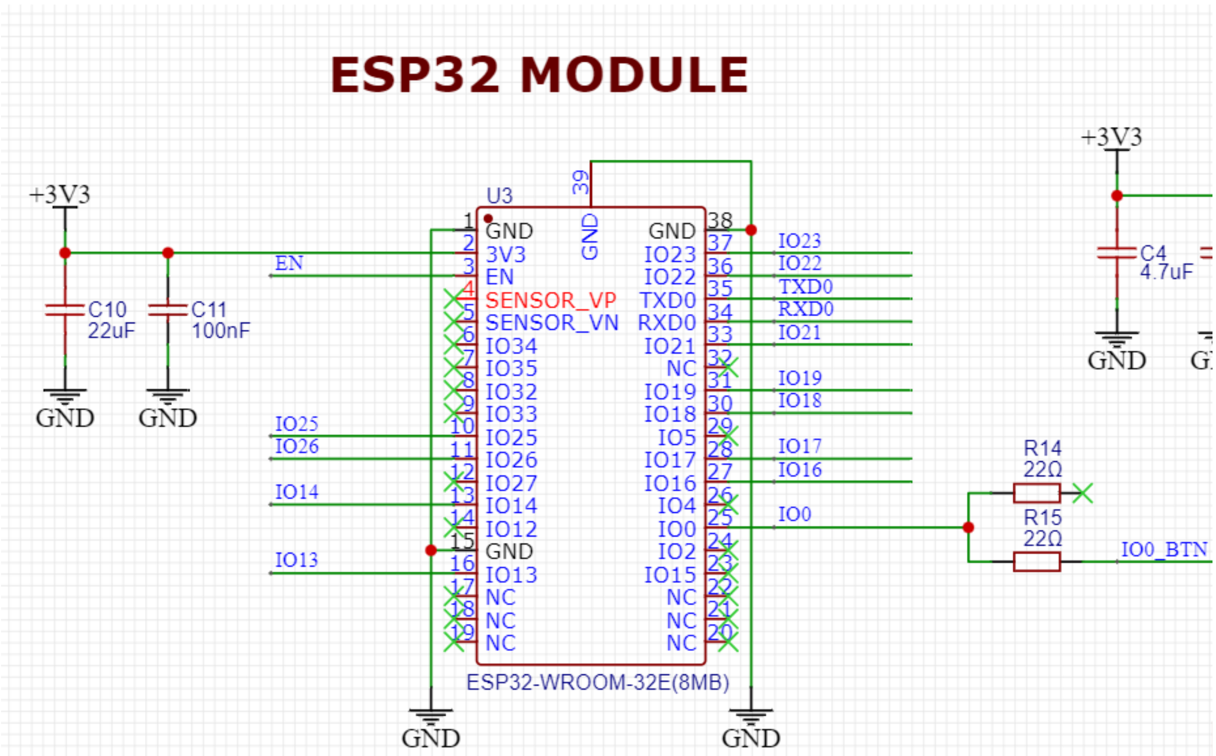

Figure 10: ESP32-WROOM-32E Connection

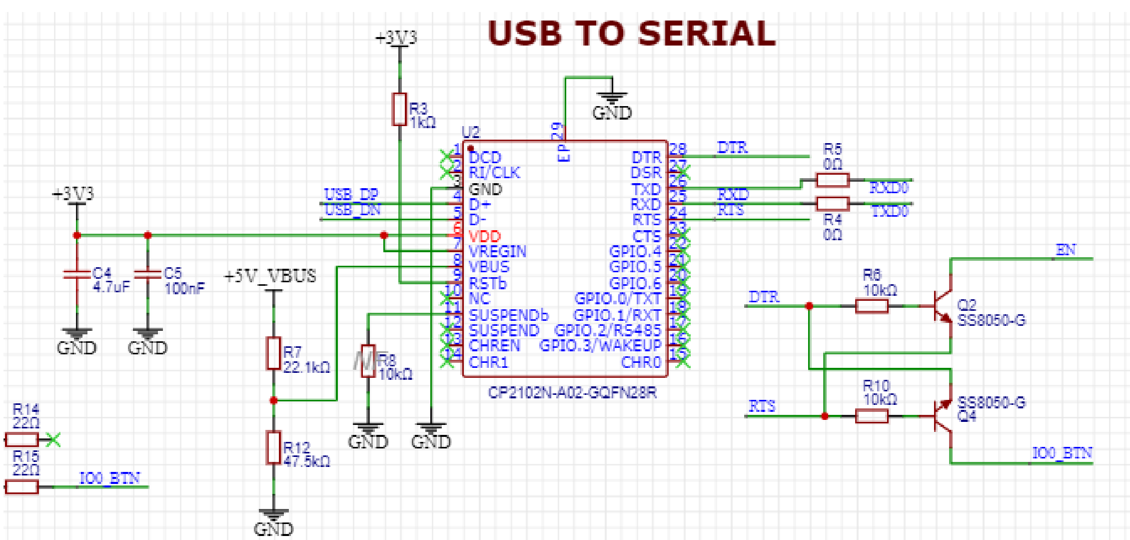

Figure 11: USB to UART Surface Mount IC Connections

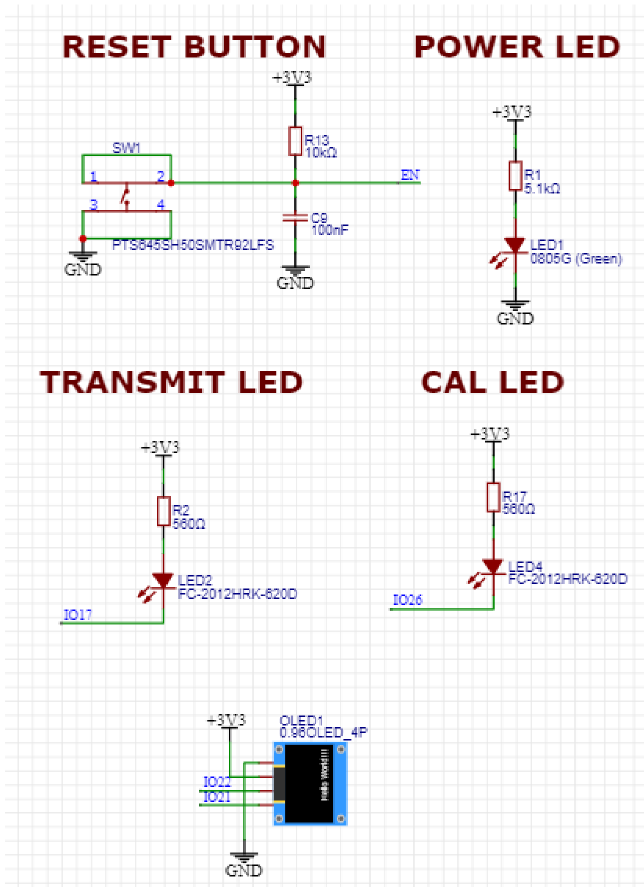

Figure 12: OLED and User IO connections

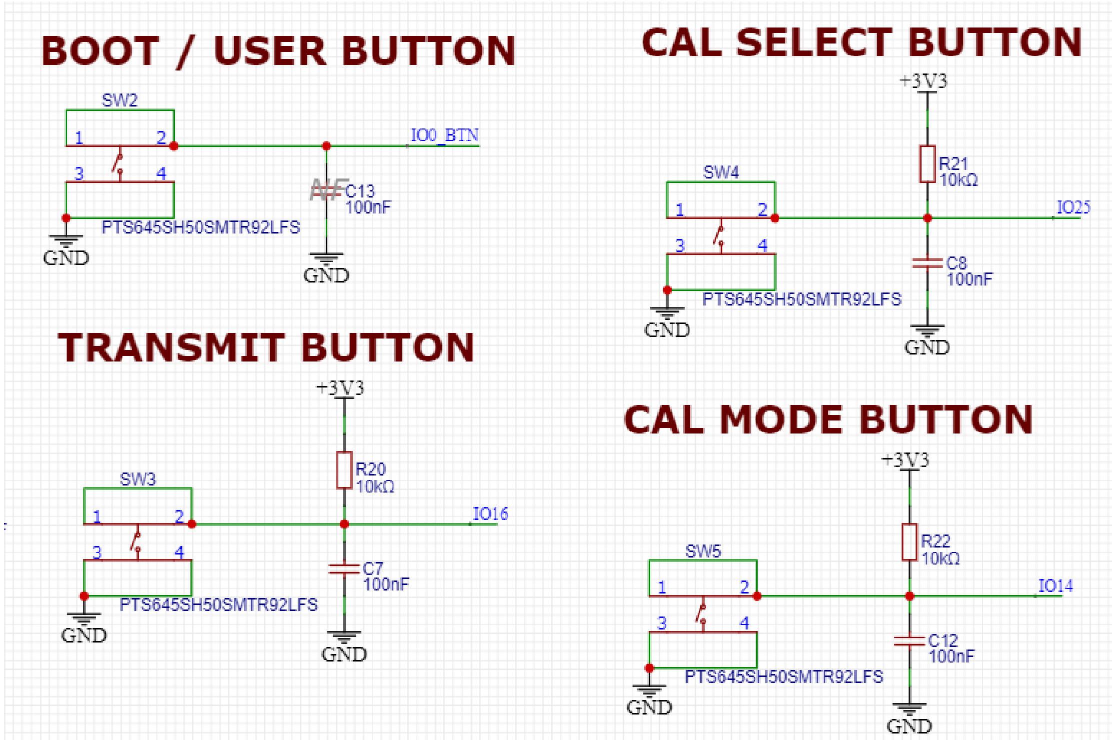

Figure 13: More User IO Connections

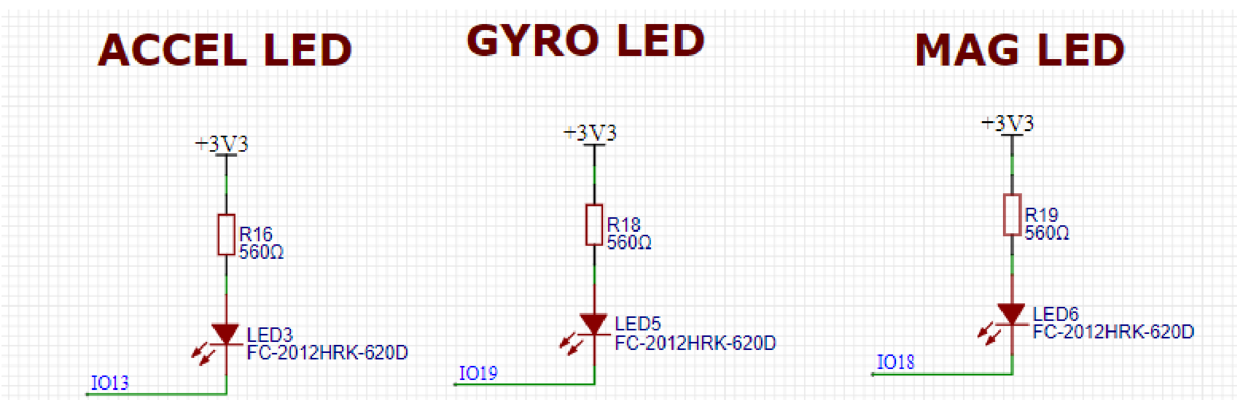

Figure 14: Sensor Calibration LED's

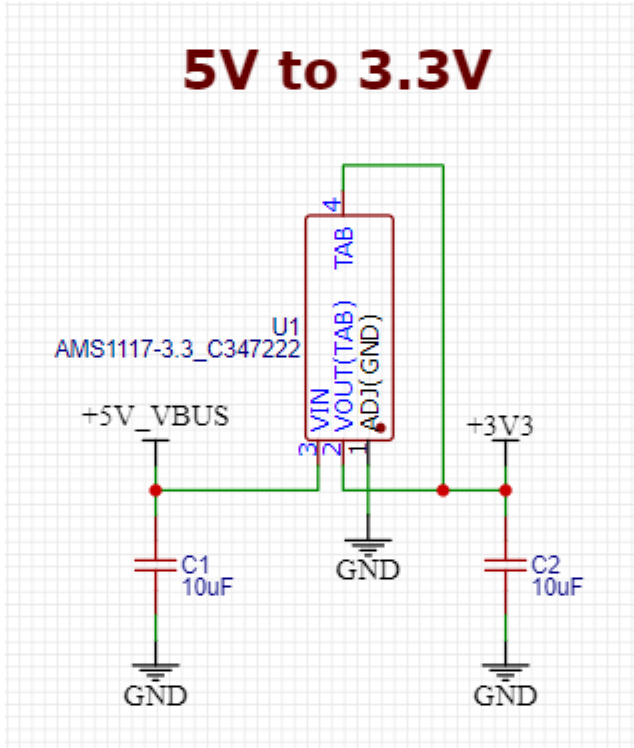

Figure 15: 5v to 3.3v Voltage Regulator Connection

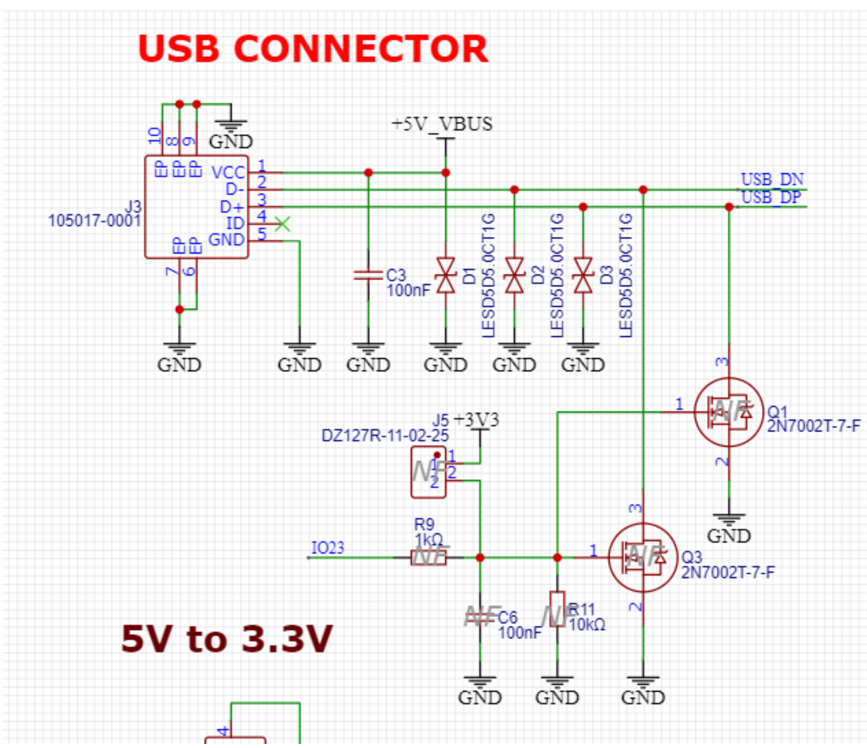

Figure 16: Micro USB Connection

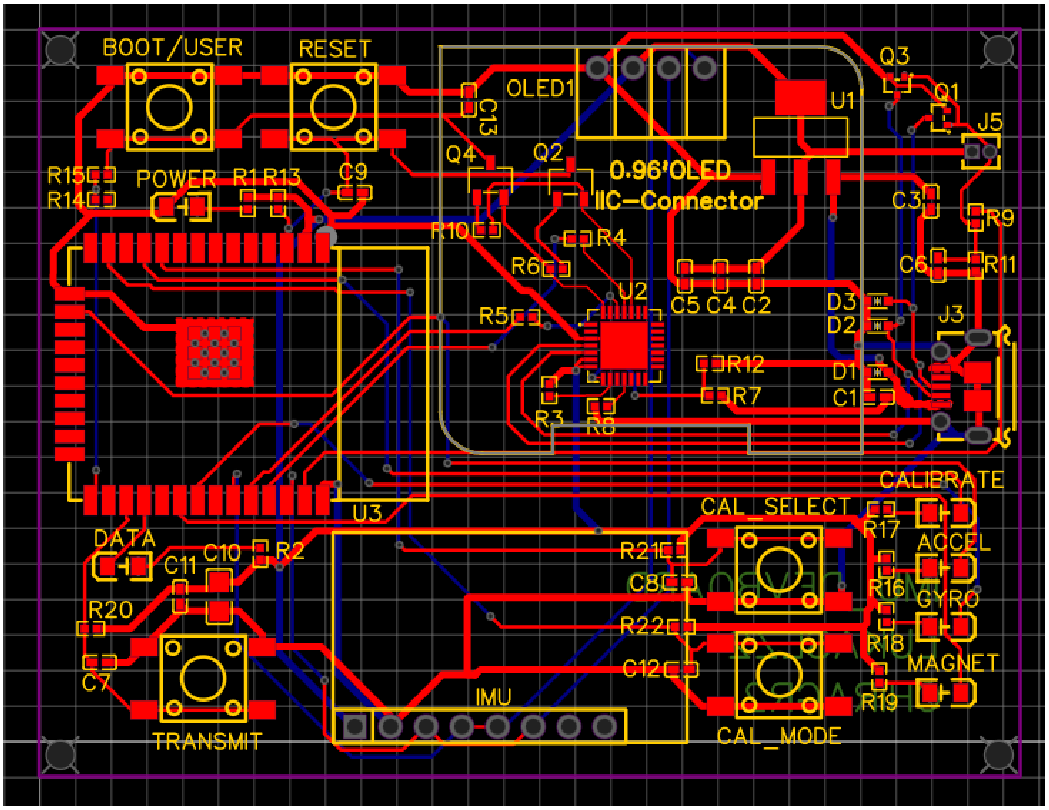

Figure 18: PCB Wire Layout