

Wildcat Rocketry Universal Flight Computer



Submitted for:

Wildcat Rocketry

Alan Levin Department of Mechanical Engineering

Kansas State University

Submitted By:

The Apogeeniuses

Tannyr Singleterry, Gannon Bird, Ethan Benne, Nathan Emerson

Department of Electrical and Computer Engineering

Kansas State University

12 December 2024

Executive Summary

The goal of this project is to develop a flight computer dedicated for Wildcat Rocketry. The design will be focused on functionality of the computer as well as future I/O and part expandability. While Wildcat Rocketry has explored flight computer development in the past, they did not have much success. Our solution aims to provide a working, dedicated flight computer that is easy to use and can be expanded later. It will also be able to offload data via Wi-Fi and USB-C.

In this report, you will find potential solutions and a development plan for our flight computer, task assignment, scheduling, and other work. We first discuss our project's background, including the problem statement and establishing customer and system requirements. Next, we will discuss our technical design with different conceptual designs as well as our ultimate design decision. Finally, we will include our technical work, financial plans, lessons learned, conclusions, references, and appendices. This document should serve as an overview of this project and a record of what we learned throughout the process.

Table of Contents

Executive Summary.....	2
Table of Contents.....	3
Project Description.....	5
Background and Motivation	5
Problem Statement	5
Project Requirements	5
Validation and Acceptance Tests	6
Verification and Validation Matrix	6
Data Needed to Support Analysis	7
Technical Design	7
Conceptual Design Descriptions	7
Design Alternative 1: Microcontroller	7
Design Alternative 2: Field Programmable Gate Array (FPGA)	7
Selection Process for Preliminary Design Solution	8
Global Issues	8
System-Level Overview	8
Applicable Standards	9
Module-Level Descriptions	10
Detailed Design/Solution Descriptions	10
Top-Level Design	10
ESP32-S3-WROMM Design	12
IMU Design	13
Power and USB Design	14
Validation and Acceptance Test Results	14
Assessment of Test Results	15
Work Plan	15
Work Breakdown Structure (WBS) & RACI Chart	15
Schedule Gantt Chart	16

Prototyping and Testing Protocol	16
Financial Plan	16
Proposed Budget	16
Comparison of Final Expenditures to Budget	17
Feasibility Assessment	18
Strengths	18
Risks	18
Lessons Learned	18
First Semester	18
Second Semester	19
Conclusions	19
First Semester	19
Second Semester	19
References	20
Appendices	21
Appendix A: Prototype Development Board.....	22

Discussion

Project Description

This project is a flight computer for Wildcat Rocketry. The goal of this project is to design and make a flight computer that will take in data from multiple sensors and produce accurate, reliable, and readable flight data.

Background and Motivation

Wildcat Rocketry has previously attempted electronics projects with some of these projects being flight computers. However, most of these flight computers never got past the designing phase. If they did get past the design phase, they never got more than one board revision. This year, Wildcat Rocketry is not attempting to design a flight computer but instead is working on a Long-Range tracking solution. The motivation for this senior design project is to design and construct a flight computer that Wildcat Rocketry can use as another way to get accurate flight data as well as expand upon later.

Problem Statement

Wildcat Rocketry desires a traditional flight computer that can gather data and transmit data in real time as well as export that data via USB-C. Also, that flight computer must be able to be expanded upon later for evolving needs.

Project Requirements

Below is a table that lists our requirements for this project. Each requirement is identified by an Identification Number (ID) with larger sections being broken into subsections. The first ID section details our power goals. ID 1.0 states the requirement that the flight computer needs to be powered by a 1S LiPo battery. The reason for this decision was that all the commercial flight computers currently used by Wildcat Rocketry implement this battery type. Subsection ID 1.a was made with regards to battery life driven by the fact that Wildcat Rocketry would like the minimum battery life to be 2 hours, while the ideal battery life would be 3-4 hours. ID 2 addresses the weight requirement of the flight computer. Weight matters a significant amount when it comes to rocket flight, so the lowest feasible weight is ideal. ID 3.0 addresses the size requirement. The flight computer must be constrained to a certain size because all the electronics for the intended rockets go inside of their avionics bay. Each avionics bay has a certain inner diameter that the flight computer

must fit inside of. This inner diameter is typically 3in-5in. ID 4.0 deals with cost. Since club funds will be needed to produce these and revise the design, a low cost is desirable. ID 5.0 is the top-level data requirement. Wildcat Rocketry wants the flight computers to store at least three flights worth of flight data. We came to this number due to the commercial flight computers on the market typically storing this amount. ID 5.a specifies the data requirement to be able to take the data from the sensors on the flight computer and turn them into readable flight data. ID 5.b specifies the data export requirement over USB-C or visualized over Wi-Fi.

ID	Requirement	Description
1.0	Input Voltage and Current	1S LiPo
1.a	Minimize Power Use	Design Goal: The longer we can power the board the better. Minimum of 2 hours of battery life. Would like 3-4 hours
2.0	Weight	Maximum 100 grams
3.0	Size	Maximum 3 in width, 4 in length
4.0	Cost	\$200 or less
5.0	Data Logging	Store at least three flights worth of data
5.a	Data Processing	Be able to process data from sensors into readable flight data
5.b	Data Export	Be able to export that data over serial or visualize over Wi-Fi

Table 1: Requirements Table.

Validation and Acceptance Tests

We will be able to verify and validate our final design by demonstrating all the project requirements. We can measure the power usage by putting the flight computer under load for a set amount of time and seeing how much voltage is left in the battery after. We can demonstrate the weight and size by measuring using a scale and ruler. The plan is to have two different board revisions, the first one being a testing unit with buttons and switches for debugging while the second revision would be one that flies and gathers data actively during the flight. We can demonstrate the data processing, storage, and exporting using the first revision testing board.

Verification and Validation Matrix

ID	Requirement	Analysis	Demonstration	Inspection	Test
1.0	Input Voltage and Current				X
1.a	Minimize Power Usage				X
2.0	Weight			X	
3.0	Size			X	
4.0	Cost	X			
5.0	Data Logging				X
5.a	Data Processing				X
5.b	Data Export		X		

Table 2: Verification Table.

Data Needed to Support Analysis

The type of data needed for this project is values produced from the sensors (High G accelerometer, IMU and barometer). We plan on collecting this data during flight, with the sensors talking to the Microcontroller and the Microcontroller will process this data.

Technical Design

Conceptual Design Descriptions

We had two alternative designs in mind for the flight computer. Our first alternative was to use a microcontroller to gather and process the sensor data, while our second was to use a Field Programmable Gate Array (FPGA). These options will now be explored:

Design Alternative 1: Microcontroller

Microcontrollers are powerful small computers used on a single integrated circuit. There is a wide variety of microcontrollers, differing with regards to processing power, I/O capabilities, and data storage. These are relatively easy to program because of the simplicity of serial execution and team experience in languages such as C/C++. Power consumption can be low, especially for a model optimized for a low power environment. Some disadvantages are the serial execution nature of microcontrollers which limits capabilities when multi-tasking, such as gathering data from multiple processors simultaneously and processing that data. The choice of microcontroller must consider the internal peripherals of the specific model such as General-Purpose Input/Output (GPIO) count.

Design Alternative 2: Field Programmable Gate Array (FPGA)

FPGAs are a configurable integrated circuit that can be repeatedly programmed after manufacturing. FPGAs contain logic blocks and allow a programmer to connect these blocks and configure them. The main advantage of using an FPGA is the logic level programming allows for design of a specialized multi-tasking circuit. This would mean that intake and processing of sensor data would be simultaneous and potentially increase the number of data points. It can also be reprogrammed to allow for quick iterating and bug fixing. There are two main disadvantages of an FPGA: The complexity of coding and the increase of power consumption. Programming at a logic level is less familiar to the team members as well as the Wildcat Rocketry members which decreases the ability for future iteration. The other disadvantage is that to utilize the increased speed of parallel processing, more power is consumed.

Selection Process for Preliminary Design Solution

We used an evaluation matrix to decide between implementing the flight computer with a microcontroller or an FPGA. Compactness and reliability were our two most weighted requirements because those are especially applicable for the size of the rocket and for consistent data. Concept 1 represents the microcontroller while concept 2 represents the FPGA.

Need	Engineering Requirement	Weight (1-10)	Base line	Concept 1	Concept 2
Lightweight	Weight < 100 grams	3	5	5	5
Lower Power Consumption	Minimum 2 hours battery life	5	5	7	3
Data storage	Can store multiple flights worth of data	7	5	7	5
Ease of use	Easy to use interface	7	5	9	3
Affordability	Unit production cost < \$200	3	5	7	3
Connectivity	Be able to output data via serial	5	5	5	5
Reliability	Produces data consistently	10	5	4	6
Compact	3 inches width, 4 inches length	10	5	7	5
Accuracy	Produces accurate data	7	5	4	7
Total			285	346	279

Table 3 Decision Matrix

Our final choice was the microcontroller. Some major factors over the FPGA were ease of use and power consumption.

Global Issues

This is to be used for K-State Rocketry Club so global issues do not pertain.

System-Level Overview

The flight computer will gather data from the flight and store that data for later analysis. It is meant to be easily expandable.

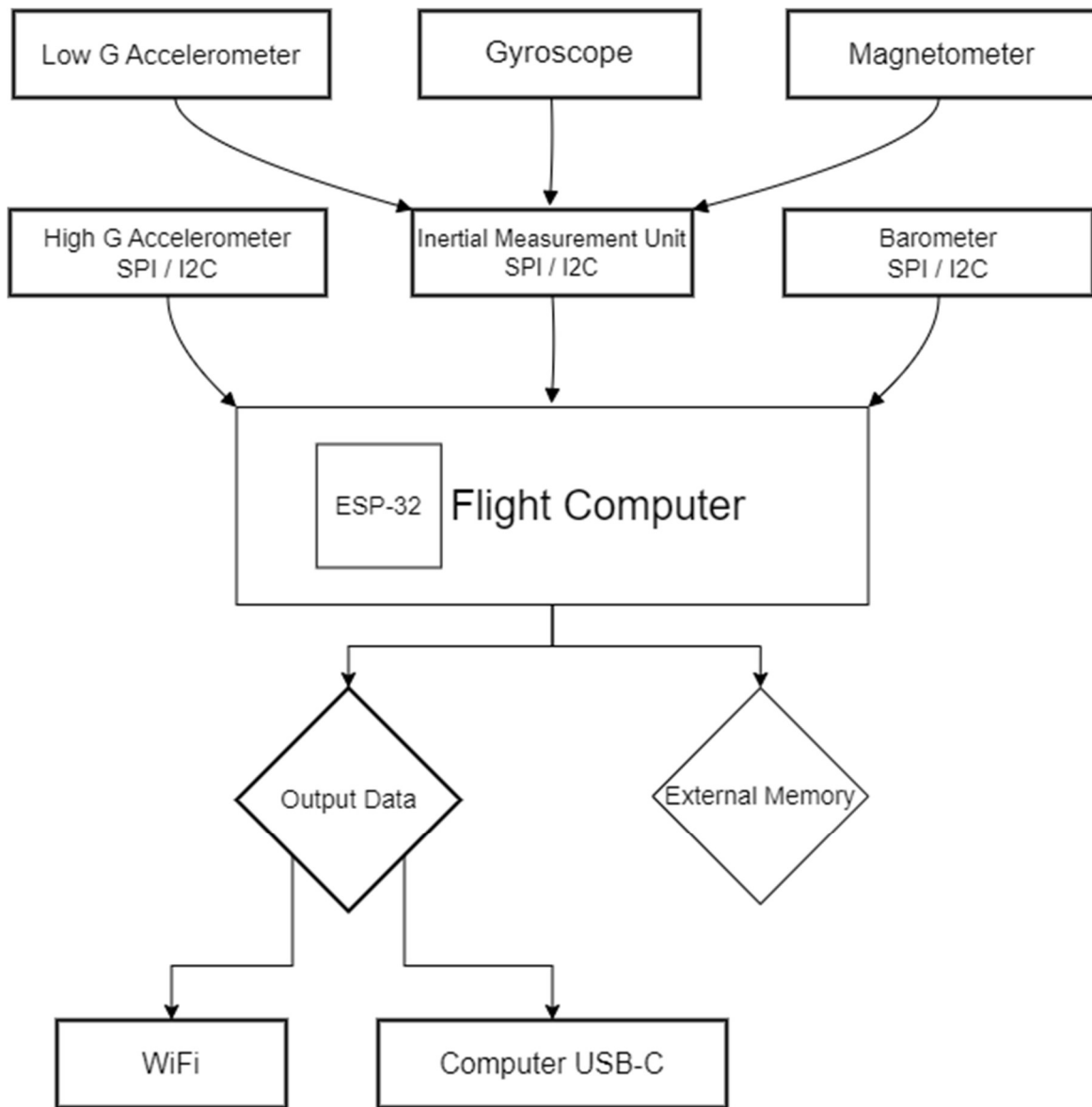


Figure 1: System Block Diagram

Applicable Standards

- I2C – We are using this standard for the barometer. I2C is controlled by NXP Semiconductors and is currently on version UM10204 [1]
- SPI – We are using this standard for the IMU, Flash, and High G Accelerometer. SPI is an uncontrolled standard.
- USB-C – We are using the USB-C standard for our flight data export bus. The USB standard is controlled by USB Implementers Forum [2], [3].
- ESP32-S3 [4]

- Wi-Fi – This standard is controlled by IEEE and the version being used is 802.11b/g/n [5].
- C++ - We are using C++ as our programming language. It is controlled by the International Organization for Standardization and is currently on version 14882:2020(E) [6].

Module-Level Descriptions

Sensors:

The sensors consist of a high-G accelerometer, barometer, and an Inertial Measurement Unit which itself consists of a low-G accelerometer, gyroscope, and magnetometer. These sensors are typical and required in a flight computer and will be used to gather data on pressure, acceleration, heading, and tilt.

Flight Computer PCB:

The Flight Computer PCB module will consist of a microprocessor on a custom-made PCB.

External Memory:

Multiple flights worth of data necessitates the use of external memory in the form of an SD card.

Output Data:

Data from each flight can be outputted via USB-C on the PCB, or by Wi-Fi access by the wireless chip on the board.

Detailed Design/Solution Descriptions

During the first semester, our group designed a prototype board to be used for various testing needs. This board is not meant to fly but rather to assess our hardware connections and to test flight code. Each one of our team members designed at least one schematic in KiCad. Below are a few of our important schematics, including our top-level design, ESP32, IMU, and power/USB. Also see Work Breakdown and RACI chart below for specifics on which member did what.

Top-Level Design

The first place to begin with this first board is the top-level design. As this project is designed with a bottom-up modular approach, the system design can be seen effectively through our top-level schematic below.

ESP32-S3-WROOM Design

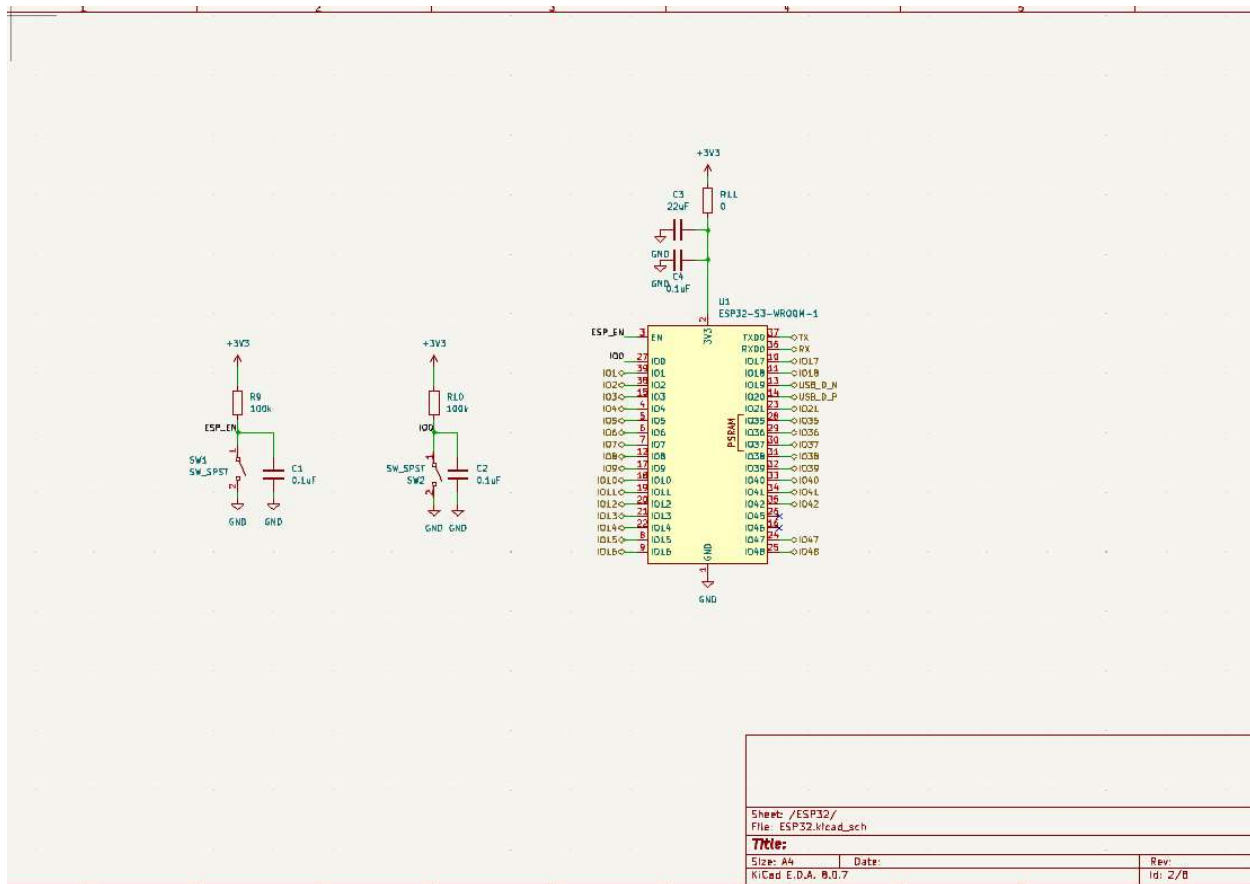


Figure 3: ESP32-S3-WROOM Schematic.

The ESP32 sub schematic, Figure 3, contains the ESP32-S3-WROOM-1 module as well as the reset and boot mode select buttons. Each sub schematic has a shared characteristic in the 0Ω resistor between 3.3V power and the power for that subsystem, R11 in ESP32. This is useful for debugging and testing allowing us to disconnect power to a whole section of the board for debugging connection issues.

Another note with the ESP32 sub schematic is the GPIO hierarchal label names. The ESP32-S3 has an open pin mux except for a couple of signals. It has two pins dedicated to UART0, labeled TX and RX, as well as the two GPIO used for the USB programming labeled USB_D_P and USB_D_N. This helps ensure these important signals are seen at the top level.

IMU Design

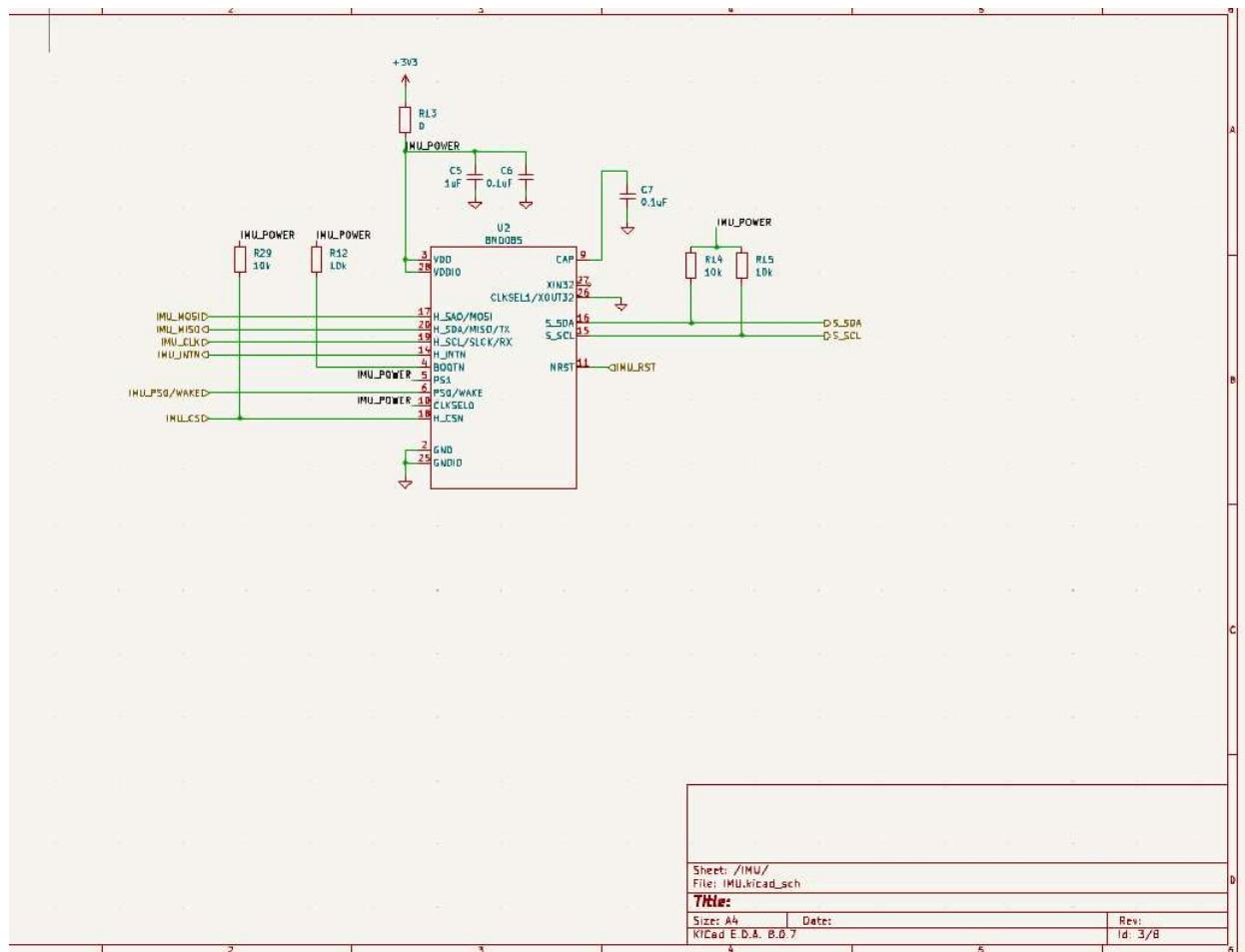


Figure 4: IMU Schematic.

Figure 4 shows the IMU schematic as an example of what each single IC schematics, like the Baro, High G Accel, and Flash, look like. They contain their IC focus in the center with their decouple cap and other required passive circuits.

The IMU was chosen as an example because of an irregularity with the secondary I2C on it. The IMU's contains a secondary I2C bus that it can connect to sensors as the master to better pool data to single points. We currently do not know if we want to use this feature or not, we have Do Not Populate 0 ohms in the Top-Level Schematic to allow for testing; however, the IMU requires these pins to be pulled up for proper startup sequencing. Since these are required, regardless of if this I2C bus is shared for not, the pullups are contained in this schematic.

Power and USB Design

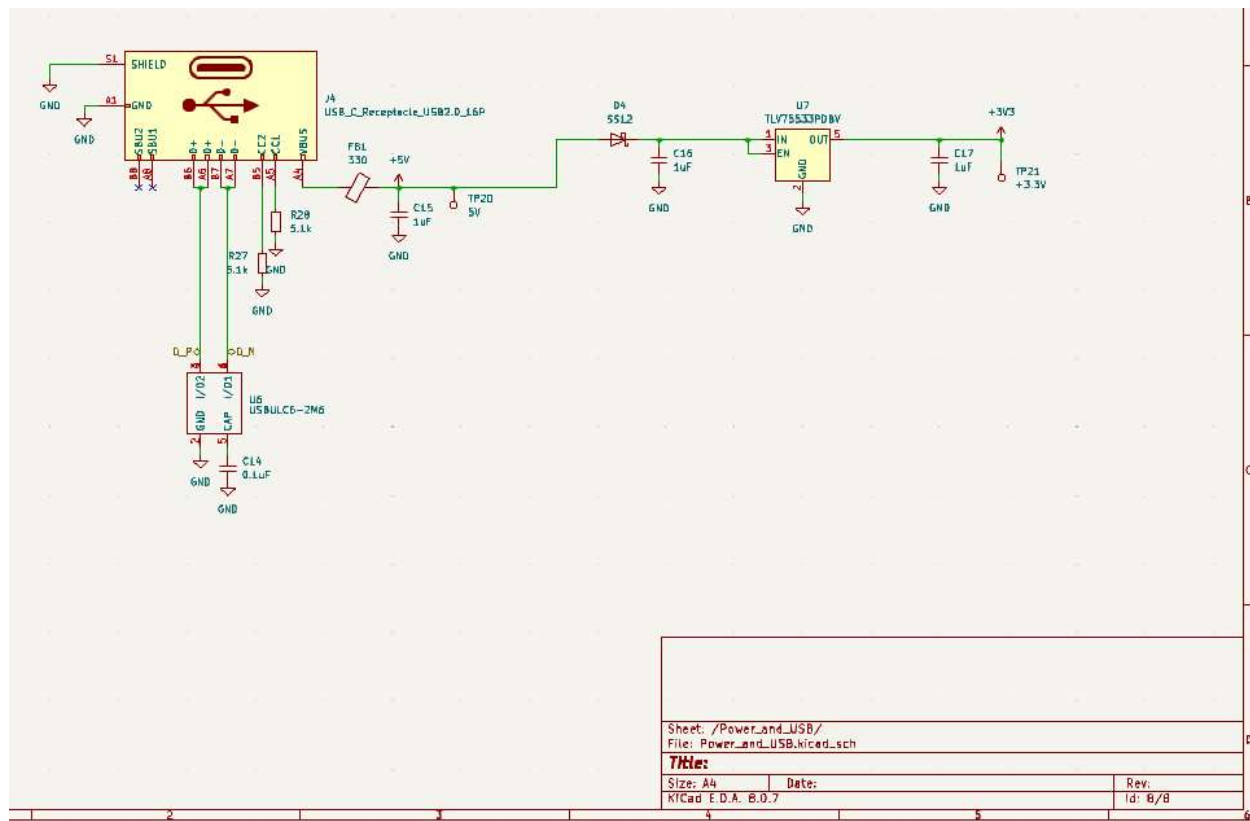


Figure 5: Power and USB.

Figure 5 contains the power and USB Schematic this includes: the USB-C port, USB TVS protection, Reverse Polarity Protection Diode, and linear voltage regulator.

The USB-C port breaks out the data lines to the TVS protection diode and the 5V power to the 3.3V LDO. Also, at the USB-C port are the Configuration Channel or CC pins, these are used to identify the port to the other side. To set our USB-C port to be an Upstream Facing Port, UFP, so that the other device knows to supply power to it, we place 20% 5.1K Ω pull downs to set it as a UFP without Detect Power capabilities.

We decided to use a set 3.3V linear regulator for our power supply. We picked a linear regulator because we believe at our lower current draw, we will see more efficiency than we would with a Switching Mode Power Supply, SMPS. Smaller SMPS topologies also require running at higher frequency's potential introducing switching noise that could harm the WI-FI connectivity of the system.

Validation and Acceptance Test Results

THIS IS SEMESTER TWO ONLY

Assessment of Test Results

THIS IS SEMESTER TWO ONLY

Work Plan

This section our group is still finalizing details and tasks.

Work Breakdown Structure & RACI Chart

Here is a rough RACI chart. To be fixed and change later.

Task #	Task	Tannyr	Gannon	Nathan	Ethan
1	Designing PCB	---	---	---	---
1a	Footprints		R		A
1b	Schematics	---	---	---	---
1bi	ESP		R		A
1bii	Power and USB		R		A
1biii	IMU		A		R
1biv	Barometer		A		R
1bv	High G Accelerometer		A		R
1bvi	Flash		A	R	A
1bvii	Debug	R	A		A
1c	PCB Layout		R		A
2	Design Software	---	---	---	---
2a	Flight Software	R		A	
2b	Data Visualization	A		R	
3	Communicating with Client	R			
4	Possible Board Assembly	A	A	A	R
5	Testing	A		R	

Table 4: RACI Chart.

R = Responsible, A = Assisting

Schedule Gantt Chart

Rocket Flight Computer

Select a period to highlight at right. A legend describing the charting follows.

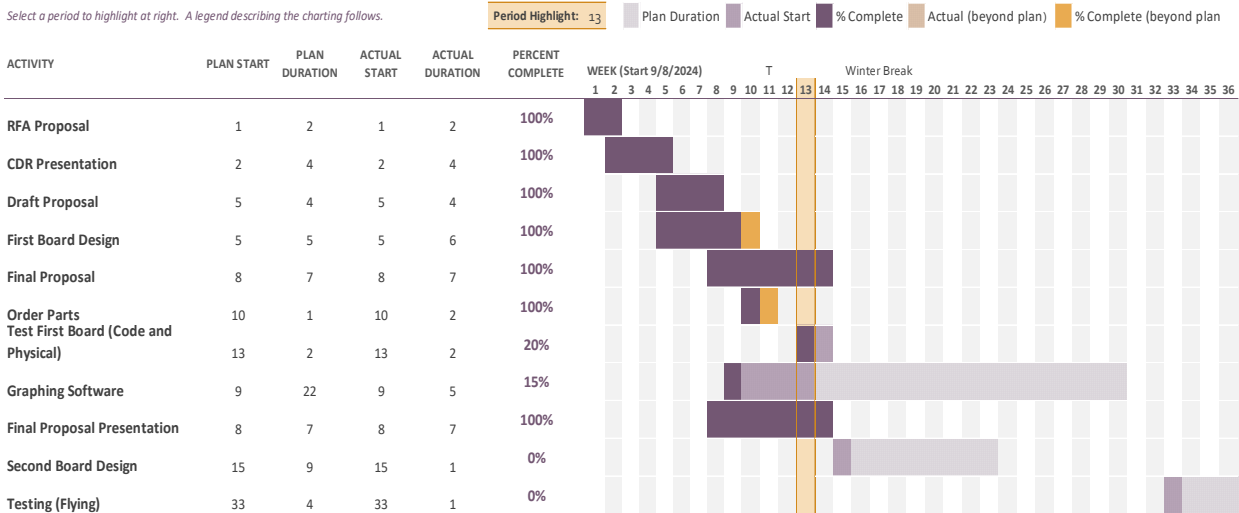


Figure 6: Gantt Chart.

Prototyping and Testing Protocol

The first revision board for this project will be the testing board. We will create artificial flight data to be sent to the flight computer for it to process. We will use either the Senior Design lab or Electronics Design Club room for lab space. Safety aspects to be considered is a potential fire hazard with using a battery. We will make sure to always have a fire extinguisher close by.

Financial Plan

This section covers the costs of the project.

Proposed Budget

Here is a preliminary bill of materials for the first board revision.

- First board Revision
 - PCB - \$4.00
 - Sensors - \$20.00

- Microcontroller - \$10.00
- IMU - \$30.00

For a total of \$64. This is bound to change based on components and PCB costs.

Our group did end up placing an order during the first semester. Here is the breakdown of the actual costs for our test board.

- Digi-Key Order: \$42.49
 - 2 IMU (BNO085) - \$29.46
 - 2 6-Pin Screw Terminal - \$3.98
 - 2 2-Pin Screw Terminal - \$2.06
 - Shipping - \$6.99
- JLCPCB & Assembly: \$124.82
- Parts on Hand
 - 2 ESP32-S3-WROOM-1-N8
 - 2 USB-C Connectors
- Price for 5 boards (2 assembled) ~ \$165

As seen, the price per board for the testing board is roughly \$80 per board. This is unfortunately above our proposed budget. We got two boards assembled so that our electrical group (Ethan and Gannon) can work on testing power and the physical board, and our computer engineering group (Nathan and Tannyr) can test out flight code. With the three unassembled boards, one was given to Tannyr, Ethan, and Nathan. For the completed boards, Tannyr and Gannon will keep the two at the conclusion of the course.

Priority 1) ESP32-S3: this ESP32 is the microcontroller we proposed to use for our code and data handling. In case we are not able to obtain funds to buy the ESP32-S3, this will require the following changes to our Technical Solution, Work Plan, and Financial Plan. For the Technical Solution this would require us to move to another microcontroller, possibly something like the STM32F042F6. However, this would mean we would not be able to use the ESP-DASH feature of the ESP32. Our Work Plan would have to change to either implement that in a different way, or not at all. Our Financial Plan would be less due to the STM32F042F6 being an overall less power hungry and smaller chip.

Comparison of Final Expenditures to Budget

THIS IS SEMESTER TWO ONLY

Feasibility Assessment

We believe that this project is feasible. Our group has a mix of Electrical and Computer Engineers. Most of our group have other project experience through classes or clubs. The biggest weakness with our team will be communication. We must make sure to always be communicating what needs to be done and what we plan to do as group members. All of us are a part of clubs and have jobs along with school. The best course to solve these is to use our given work time wisely.

Strengths

Each member of the project brings in their own unique strengths that fit well into this board's development. Gannon's best strength is their experience in KiCad and PCB design through earlier work/club projects. There are not many college students that have as much experience doing PCB design as Gannon. Ethan's strength is his background in circuit analysis and design. Having someone who has excelled in circuits classes is useful to make sure all our design is doing what we intend it to. Nathan's strength is coding experience, having implemented the code for the ESP controlled Solar Club speaker prototype. Nathan also shines bright like a diamond and is always eager and able to work on code. Finally, Tannyr is serving as Wildcat Rocketry's vice president. Tannyr has a good overall skillset from his background in work/clubs to help the entire team wherever the need arises. As a team with adequate Electrical and Computer Engineering skills, we believe that our team has what it takes to be successful throughout this project.

Risks

No matter what project, there will always be places for weakness. One example is communication within the team. Our team currently uses a text message group and Discord to communicate needs within the team. Another risk is there is only a limited number of launch dates to launch our board in a rocket. Given that time limit, we may have to aim for a sooner project completion date or must travel, which not all our members may be able to do.

Lessons Learned

First Semester

Early in the project's life cycle, there are not too many lessons that we have had to learn. However, one of them has been component selection. The IMU our group wanted to go with

was almost perfect. But during a final check before buy, we figured out that the IMU we wanted to buy (ICM-20948) had 1.65V logic, while everything else on our first board ran at 3.3V logic. As well the ICM-20948 is marked “Not recommended for new design” on DigiKey. Our group made the proper switch to a new IMU, the BNO085. This switch caused our estimated price of the IMU to double. Another issue we met was when we were placing our JLCPCB order, they were out of the specific ESP32 model we wanted. We did not check before placing the order to check if they were still in stock. We had to get the ESP32 from someone else who was willing to give theirs to us. Overall, the first semester has gone smoothly, but we have learned to always triple check to make sure parts are compatible with the rest of our board, and to check if parts are going to be in stock when ordering.

Second Semester

THIS IS SEMESTER TWO ONLY.

Conclusions

First Semester

Overall, this project so far has been a great success. Our group has gone through the design process for a prototype development board. This board has been designed in KiCad and ordered through JLCPCB with hand soldering from our group. We have started testing this development board. The software is still in an early stage with plans to tackle a sizable part over winter break. Our group works well together to accomplish goals, and we are looking forward to the second semester and everything we will accomplish.

Second Semester

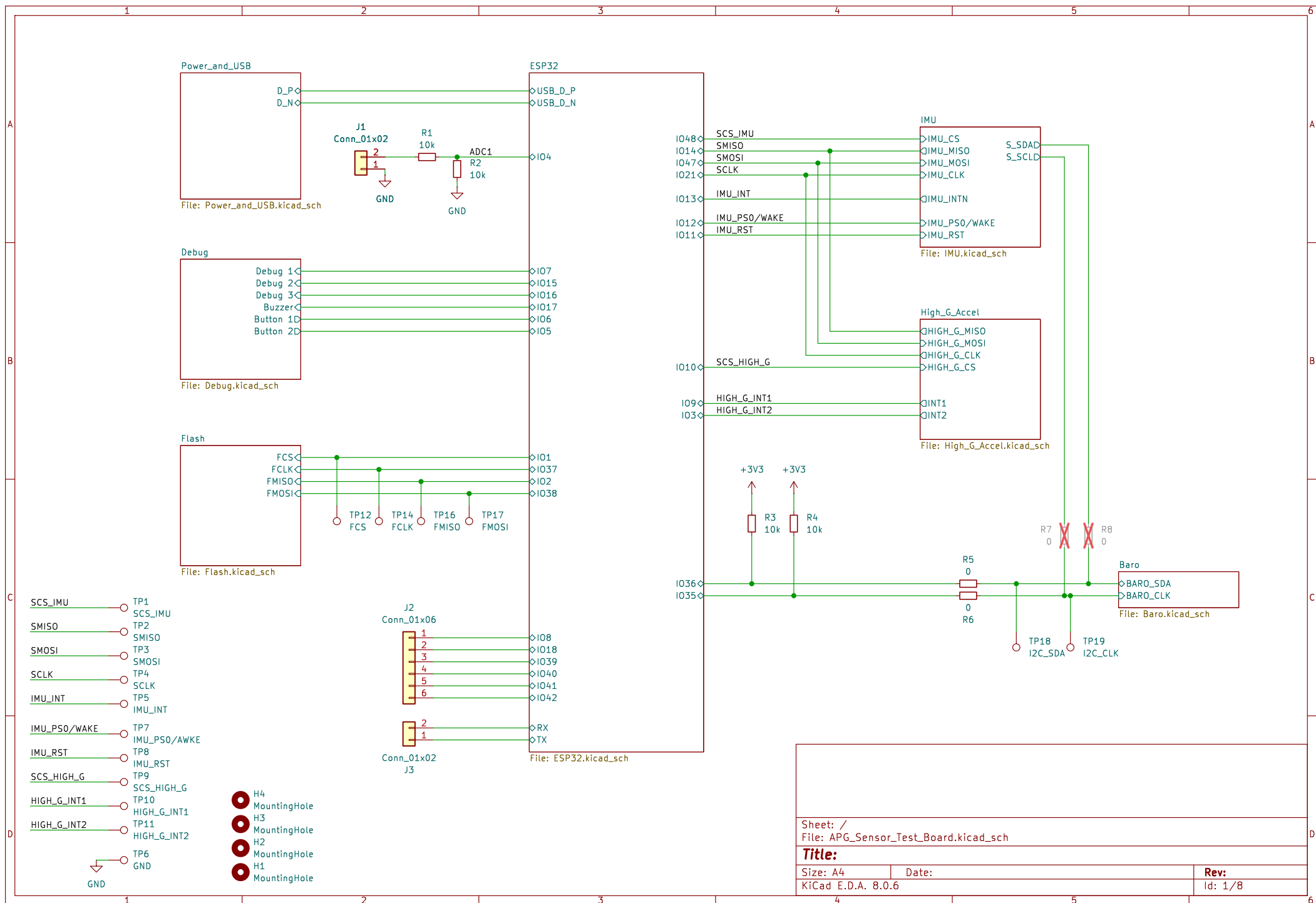
THIS IS SEMESTER TWO ONLY.

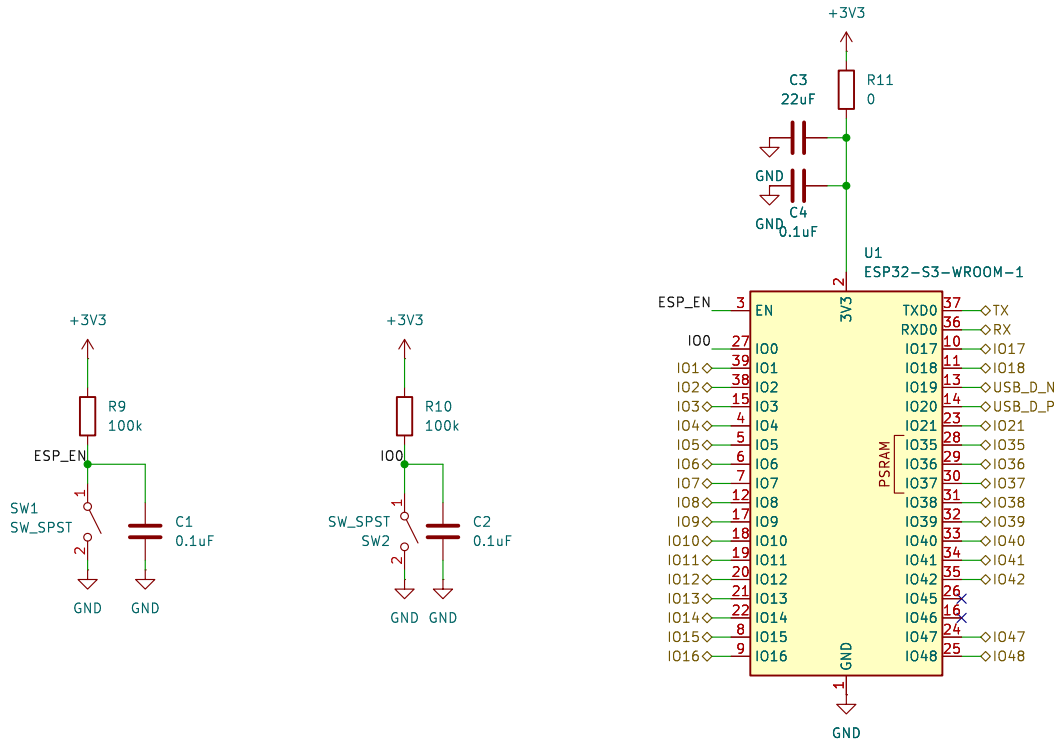
REFERENCES

- [1] "I2C-bus specification and user manual," vol. 2021, 2021.
- [2] "USB Type-C® Cable and Connector Specification | USB-IF." Accessed: Dec. 05, 2024. [Online]. Available: <https://www.usb.org/usb-type-cr-cable-and-connector-specification>
- [3] A. Rogers, "Introduction to USB Type-C™," 2015, Accessed: Dec. 12, 2024. [Online]. Available: <https://ww1.microchip.com/downloads/en/AppNotes/00001953A.pdf>
- [4] "ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U Datasheet Version 1.4." Accessed: Dec. 03, 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf
- [5] "IEEE Standards Association," IEEE Standards Association. Accessed: Dec. 03, 2024. [Online]. Available: <https://standards.ieee.org/ieee/802.11n/3952/>
- [6] "The Standard : Standard C++." Accessed: Dec. 05, 2024. [Online]. Available: <https://isocpp.org/std/the-standard>

APPENDEICES

Appendix A: Prototype Development Board





Sheet: /ESP32/
File: ESP32.kicad_sch

Title:

Size: A4

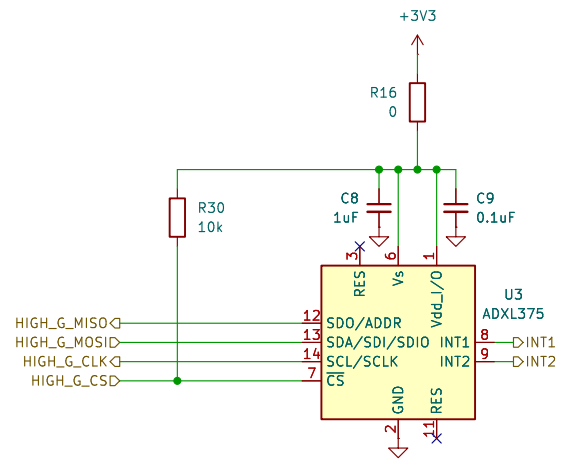
Date:

KiCad E.D.A. 8.0.6

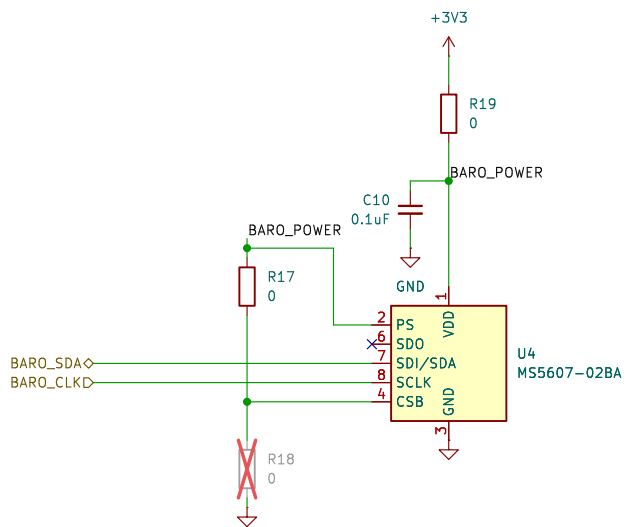
Rev:

Id: 2/8





Sheet: /High_G_Accel/ File: High_G_Accel.kicad_sch		
Title:		
Size: A4	Date:	Rev:
KiCad E.D.A. 8.0.6	Id: 4/8	



Sheet: /Baro/
File: Baro.kicad_sch

Title:

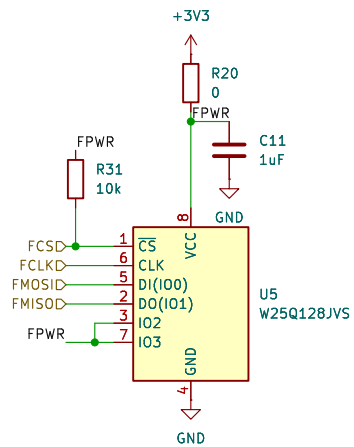
Size: A4

Date:

KiCad E.D.A. 8.0.6

Rev:

Id: 5/8



Sheet: /Flash/
File: Flash.kicad_sch

Title:

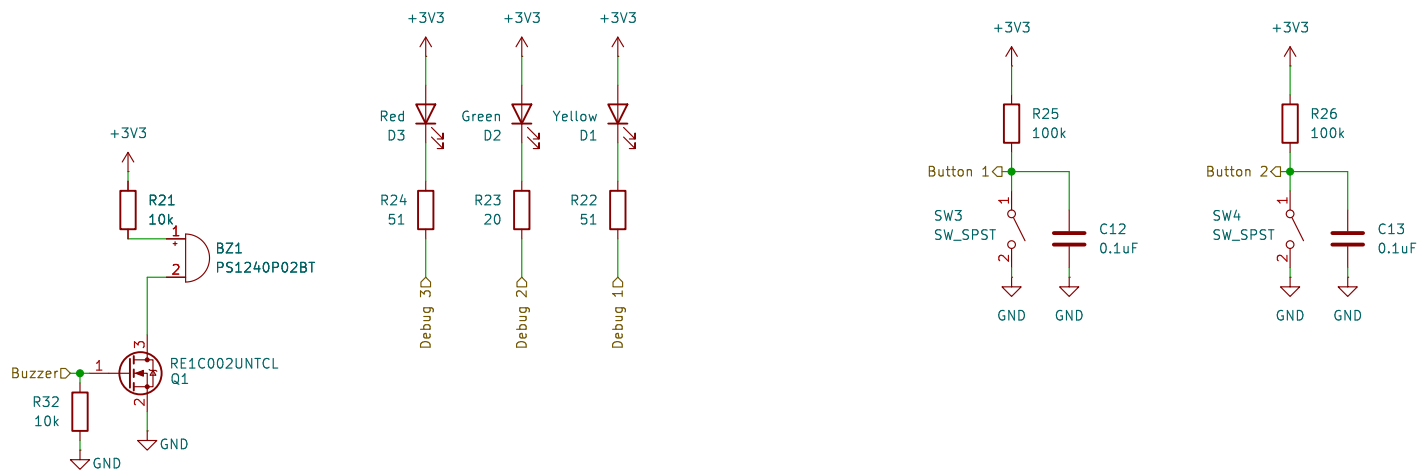
Size: A4

Date:

KiCad E.D.A. 8.0.6

Rev:

Id: 6/8



Sheet: /Debug/
File: Debug.kicad_sch

Title:

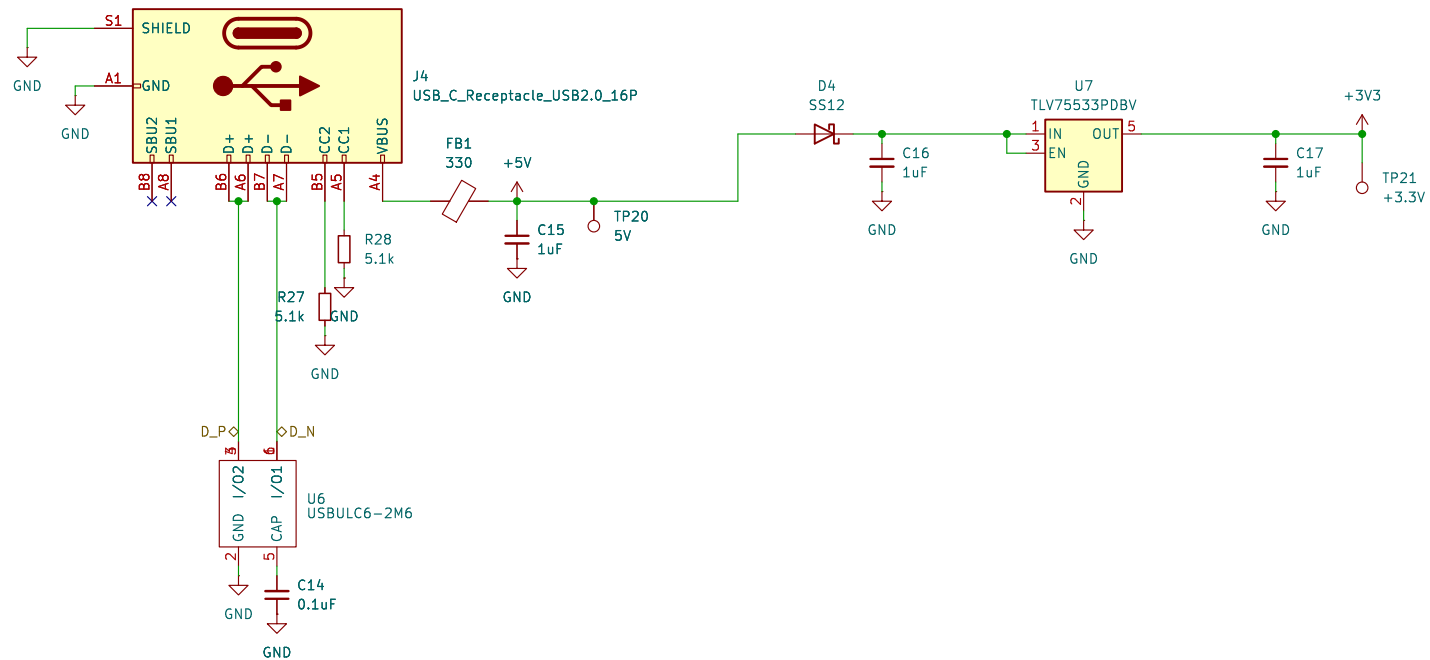
Size: A4

Date:

KiCad E.D.A. 8.0.6

Rev:

Id: 7/8



Sheet: /Power_and_USB/
File: Power_and_USB.kicad_sch

Title:

Size: A4

Date:

KiCad E.D.A. 8.0.6

Rev:

Id: 8/8

