The University of Texas Rio Grande Valley

College of Engineering and Computer Science

Department of Electrical and Computer Engineering

Senior Design Project I/II

Final Report

Project:

Smart Detection Drone

Team member 1: Jordan Lara, UID: 20596803 Major**:** BSEE

Team member 2: Gabriel Vargas, UID: 20217442 Major**:** BSEE

Team member 3: Luis Martinez, UID: 20364085 Major**:** BSCE

Project Advisor: Dr. jinghao yang **Co-advisor** (optional):

December 2025

Edinburg/Brownsville, TX (USA)

CERTIFICATE

This is to certify that, Jordan Lara, Gabriel Vargas and Luis Martinez, UID number 20596803 20217442 and 20364085 respectfully, have been working under my supervision on the project titled “Smart Detection Drone” towards the completion of their Senior Design project. The work satisfies the requirements of the Senior Design I/II (EECE 4361/4362) course.

Advisor Dr. jinghao yang Signature 签署人： 6555945E6255418... Date: 12/7/2025

Co-Advisor _____ Signature _____ Date: _____

Table of Contents

Technical Abstract

This senior design project focuses on building a smart detection drone that can help inspect concrete structures such as bridges, parking garages, and large retaining walls. Right now, most inspections still depend on people climbing ladders or riding lifts to look for cracks and measure them by hand. That approach is slow, physically demanding, and exposes inspectors to fall hazards every time they need to get close to a beam, deck, or wall. Our goal is to design a drone-based system that keeps inspectors on the ground while the aircraft gets close to the structure, captures clear images of cracks, and provides better documentation of material degradation over time.

During this first semester, the work is centered on planning, hardware design, and manufacturability rather than full autonomous operation. The team defined functional requirements for image quality, safety, ease of use, and future expandability. Based on those requirements, we created system block diagrams, pinout documents, and a custom flight controller design. The electronics were implemented in EasyEDA and laid out as a single panelized printed circuit board that contains three separable modules connected with mouse-bites. Design decisions were constrained by a budget of about four hundred dollars, limited part availability, and the need to use components that a small inspection firm could realistically purchase. The resulting board integrates a microcontroller, power regulation, sensor interfaces, and communication links intended to support future camera and detection hardware. Full firmware development, flight testing, and crack-analysis algorithms are planned for the next semester; this report documents the motivation, context, and detailed hardware groundwork needed to make that next phase possible.

Non-Technical Abstract

Concrete structures are everywhere—bridges, parking garages, overpasses, and tall retaining walls that most people drive or walk past without thinking about them. Over time, these structures develop cracks that can warn engineers that the concrete is starting to wear out or that deeper problems may be forming inside. Right now, the common way to check for this damage is to send inspectors up on ladders, lifts, or scaffolding so they can look closely at the surface and measure cracks with simple tools. This kind of work is slow and uncomfortable, and it can be dangerous, because falls from height are still one of the leading causes of death on construction sites. Our project is trying to make that process safer and more efficient by using a drone to handle as much of the close-up work as possible.

The idea behind our smart detection drone is straightforward: let the drone get close to the concrete, and let the people stay on the ground. The drone will carry a custom flight controller and, later on, a camera and other sensors to capture clear images of cracks in places that are hard or risky for people to reach. In this first phase, we focused on designing the electronics that will control the drone. We created a detailed circuit design and laid out a printed circuit board that combines three modules on one panel, which can be snapped apart after manufacturing. All of this had to fit within a student budget of around four hundred dollars, using parts that a small company could actually buy. The next semester will focus on writing the software, assembling the drone, and testing it in real situations. This report explains why a tool like this is needed and shows the concrete steps we have taken to turn the idea into real hardware.

1. Introduction

1.1 The Role of Infrastructure in Modern Society

The foundation of modern civilization is built upon concrete. From the aqueducts of ancient Rome to the soaring interchanges of contemporary metropolises, concrete infrastructure serves as the physical backbone of society. It facilitates the movement of goods, the commute of the workforce, and the connectivity of communities. However, unlike the unreinforced concrete of the Roman Pantheon which has stood for two millennia, modern reinforced concrete is a composite material with a finite lifespan. The steel reinforcement bars (rebar) embedded within the concrete provide essential tensile strength but also introduce a vulnerability: corrosion.

As these structures age, they are subjected to a relentless assault of environmental stressors. Freeze-thaw cycles cause water trapped in microscopic pores to expand, creating internal pressure that leads to cracking. De-icing salts, used liberally in winter climates, dissolve into brine that penetrates the concrete matrix, attacking the passivation layer of the steel rebar and accelerating oxidation. Dynamic loading from heavy freight traffic induces fatigue cycles that propagate micro-cracks into macro-fissures.

A primary and critical indicator of such deterioration is the manifestation of surface fissures. These cracks are not merely cosmetic blemishes; they are the external symptoms of internal pathology. A longitudinal crack along a beam may indicate rebar corrosion; a diagonal crack near a support may indicate shear failure. Consequently, the capacity to accurately observe, document, and track these fissures constitutes a pivotal component in ensuring the safety and reliability of infrastructure.

1.1.1 The Chemistry of Corrosion

To understand the urgency of crack detection, one must understand the electrochemistry occurring within the bridge deck.

$$Fe \rightarrow Fe^{2+} + 2e^{-}$$

$$2H_2O + O_2 + 4e^{-} \rightarrow 4OH^{-}$$

$$2Fe^{2+} + 4OH^{-} \rightarrow 2Fe(OH)_2$$

The formation of ferrous hydroxide ($Fe(OH)_2$) and its subsequent oxidation into hydrated ferric oxide ($Fe_2O_3 \cdot nH_2O$), commonly known as rust, involves a volumetric expansion. Rust occupies approximately six times the volume of the original steel. This expansion generates internal tensile stresses within the concrete that can exceed 10 MPa, far surpassing the tensile strength of the material (typically 3-5 MPa). The result is spalling, delamination, and the catastrophic loss of structural integrity.

1.2 The Evolution of Inspection Technology: A History of Failure

The imperative for rigorous inspection is unfortunately written in blood. The history of bridge safety regulations in the United States is largely reactive, driven by catastrophic failures that exposed systemic gaps in monitoring.

1.2.1 The Silver Bridge Collapse (1967)

On December 15, 1967, the Silver Bridge connecting Point Pleasant, West Virginia, and Gallipolis, Ohio, collapsed into the Ohio River, killing 46 people. The cause was traced to a minute cleavage fracture in a single eyebar of the suspension chain, a defect that was impossible to detect with the inspection methods of the time. This tragedy was the catalyst for the creation of the **National Bridge Inspection Standards (NBIS)**, which mandated the periodic inspection of all public bridges.

1.2.2 The I-35W Mississippi River Bridge (2007)

Forty years later, on August 1, 2007, the I-35W bridge in Minneapolis collapsed during rush hour, resulting in 13 fatalities. The investigation revealed that undersized gusset plates were the primary failure mode, but it also highlighted the limitations of the "structurally deficient" classification system. The bridge had been rated as structurally deficient for years, yet it remained open to traffic. This underscored the need for more granular, data-driven monitoring rather than broad categorical ratings.

1.2.3 The Fern Hollow Bridge (2022)

Most recently, the collapse of the Fern Hollow Bridge in Pittsburgh on January 28, 2022, served as a stark reminder that the crisis is ongoing. Investigators found that severe corrosion in the bridge's legs had been documented in inspection reports for years, yet insufficient action was taken. This incident highlighted a critical gap: the disconnect between **inspection data** and **maintenance action**.

These historical case studies demonstrate that visual inspection alone is insufficient if the data is subjective, sporadic, or not effectively communicated to decision-makers. There is a critical need for methodologies that provide objective, high-resolution, and repeatable data to bridge the gap between detection and remediation.

1.3 The "Smart Detection Drone" Concept

The concept of a "smart detection drone" capitalizes on the convergence of three technological revolutions: the miniaturization of flight controllers, the efficiency of brushless propulsion, and the advent of edge AI computing.

Rather than utilizing a drone merely as an aerial camera—a "flying tripod"—a smart detection system is engineered as an active scientific instrument. It is designed to: 1. **Navigate** with precision in GPS-denied environments (such as under bridge decks). 2. **Capture** imagery under controlled parameters (consistent distance, angle, and lighting). 3. **Process** data at the "edge" (onboard the drone) to identify defects in real-time.

In the long term, such a system aims to transmute raw aerial footage into actionable inspection intelligence, such as crack distribution maps or quantitative estimates of crack length and orientation. In this project, the drone is conceptualized not merely as a flight vehicle, but as an integrated platform synthesizing hardware, software, and algorithms into a coordinated system designed to support infrastructure condition assessment.

1.4 The Dual-Mission Challenge: A Lesson in Modularity

While the primary mission focuses on Structural Health Monitoring (SHM), this project also addresses a unique "Dual-Mission" challenge. The university's rocketry team necessitates a robust avionics system for high-altitude sounding rockets. Rather than engineering two disparate systems—a waste of engineering resources and budget, our team has undertaken the ambitious objective of developing a **Modular Flight Platform**.

This ecosystem is designed to serve two distinct and demanding aerospace applications: 1. **Autonomous Pavement Distress Mapping:** A quadcopter drone equipped with a high-performance computer vision payload (Raspberry Pi 5 + Hailo-10H) to detect and classify fissures. This mission requires high computational power and precise low-speed control. 2. **High-Altitude Rocketry Avionics:** A robust, high-G tolerant flight computer capable of managing the recovery of a sounding rocket ascending to 10,000 feet. This mission requires extreme mechanical durability and high-speed state estimation.

The core philosophy is **"Write Once, Fly Everywhere."** By developing a unified PCB that can be populated with different sensors or mated with different companion computers, we create a versatile avionics standard that outlives any single student project.

1.5 Educational Perspective and Team

From an educational perspective, the project serves as a capstone experience in electrical engineering. Designing and constructing a smart detection drone necessitates the integration of multiple disciplines encompassed within the undergraduate curriculum. The system must incorporate microcontrollers or embedded processors to interface with sensors and imaging systems, power electronics to drive propulsion systems and manage energy, communication links to transmit data between the drone and a ground station, and control algorithms to maintain stable flight dynamics.

The team responsible for this interdisciplinary effort includes: * **Jordan Lara (EE / Project Manager):** Hardware Lead. Jordan's focus was on the physical realization of the avionics, from schematic capture to the complex 4-layer PCB layout. His design philosophy prioritized signal integrity and thermal management. * **Luis Martinez (SE / Project Manager):** Software Lead. Luis architected the "nervous system" of the drone, developing the Rust-based firmware that runs on the flight controller and the Python-based AI pipeline on the companion computer. His focus was on determinism and memory safety. * **Gabriel Vargas (EE):** Hardware Implementation Lead. Gabriel focused on the practicalities of manufacturing, leading the component validation and the "Extended to Basic" cost optimization strategy that saved the project budget.

2. Problem Statement

2.1 The Infrastructure Crisis and Safety Hazards

The United States confronts a silent yet pervasive crisis. According to the **2021 Report Card for America's Infrastructure** by the American Society of Civil Engineers (ASCE), 42% of the nation's 617,000 bridges are at least 50 years old, and 7.5% are classified as structurally deficient. This means they are approaching the end of their design life and require significant rehabilitation.

The sheer volume of infrastructure requiring inspection overwhelms the available workforce. There are simply not enough qualified inspectors to visit every bridge with the frequency and detail required. This labor shortage leads to extended inspection intervals and a reliance on rapid, superficial assessments rather than thorough examinations.

2.2 Limitations of Traditional Inspection Tools

Beyond the labor shortage, the instruments utilized to characterize cracks during these inspections are relatively rudimentary and necessitate close physical contact with the surface. Standard practice frequently entails:

- **Crack Width Rulers & Comparator Cards:** An inspector positions a card or ruler adjacent to a fissure, estimates the width via visual comparison, and records the measurement. This method is inherently prone to parallax error and human inconsistency.
- **Subjectivity:** Disparate inspectors may record divergent values for the identical crack based on their experience, fatigue, or viewing angle. A fissure deemed "minor" by one inspector might be flagged as "critical" by another. There exists no digital "ground truth."
- **Access Compromises:** If specific structural components are arduous or cost-prohibitive to access, inspections may be curtailed or performed from greater distances utilizing binoculars. Under such circumstances, cracks may remain undocumented, or their true width may be underestimated.
- **Data Volume:** A bridge deck may exhibit thousands of minor fissures. Manual measurement of each instance is labor-intensive. Consequently, many inspections rely on sampling, increasing the probability that significant trends in crack propagation will be overlooked.

2.3 The Rocketry Challenge: Physics of Failure

Parallel to the infrastructure crisis, the university's rocketry team faces a distinct yet equally formidable challenge. High-power sounding rockets, designed to reach altitudes of 10,000 feet or more, require sophisticated avionics to ensure safety. The environment inside a rocket launch is hostile to electronics.

2.3.1 The G-Force Problem

During the boost phase, a solid rocket motor can accelerate the vehicle at rates exceeding 20 Gs ($196 m/s^2$). According to Newton's Second Law ($F = ma$), every component on the PCB experiences a force equal to 20 times its own weight. * **Shear Stress on Solder Joints:** Heavy components, such as large inductors or electrolytic capacitors, exert significant shear stress on their solder pads. If this stress exceeds the shear strength of the solder alloy (typically SAC305), the joint will fracture, causing an open circuit. * **Connector Separation:** Standard friction-lock connectors (like standard 0.1" headers) can easily vibrate loose or disconnect under the shock of deployment. * **BGA Vulnerability:** Ball Grid Array (BGA) packages, while space-efficient, are susceptible to cracking in the solder balls if the PCB flexes under load.

Standard consumer electronics are not designed for these loads. A Raspberry Pi or an Arduino, if hard-mounted to a rocket frame, would likely fail during the first second of flight.

2.4 The Integration Gap: The "Walled Garden" Problem

The question arises: why is the procurement of a commercial drone for the bridge and a separate flight computer for the rocket insufficient? Why build when you can buy?

2.4.1 The DJI "Black Box"

Commercial drones, particularly those from market leader DJI, constitute "walled gardens." They are excellent for photography but hostile to engineering modification. * **Encrypted Firmware:** DJI signs and encrypts their firmware. A student team cannot modify the flight control loops to test a new control theory algorithm. * **Proprietary Protocols:** The communication between the flight controller and the ESCs, or the drone and the camera, often uses proprietary, undocumented protocols. * **No GPIO Access:** A DJI Mavic does not expose General Purpose Input/Output (GPIO) pins. One cannot simply wire a custom sensor (like a strain gauge or a Geiger counter) into the drone's brain. * **Compute Limitations:** While they have powerful video processors, they do not allow third-party code to run on the main processor. Integrating a Hailo-10H AI accelerator is physically and electronically impossible without destroying the drone.

2.4.2 The Connector Proliferation of Open Source

On the other end of the spectrum are open-source flight controllers like the Pixhawk. While flexible, they suffer from "connector proliferation." A standard Pixhawk Cube has numerous JST-GH connectors for GPS, Telemetry, Power, CAN, I2C, and SPI. * **Points of Failure:** In a high-vibration environment (like a rocket), every connector is a potential point of failure. * **Cable Management:** Managing a "rat's nest" of cables inside a tight rocket airframe restricts airflow and complicates assembly.

2.4.3 The Need for a Custom Solution

To satisfy the requirements of **both** the high-G rocket (which needs a rugged, soldered-down design) and the smart drone (which needs high-bandwidth AI integration), a custom solution was

required. We needed a board that combined the flexibility of open-source software with the mechanical robustness of a single-board avionics stack.

3. Background and Literature Search

3.1 Deep Learning Architectures for Object Detection

The core of the "Smart Detection" capability lies in the application of Deep Learning, specifically Convolutional Neural Networks (CNNs), to the problem of crack detection.

3.1.1 Evolution of Object Detection

Early computer vision relied on "hand-crafted features" such as Sobel edge detectors or Hough transforms. While computationally inexpensive, these methods were brittle; they struggled to distinguish between a structural crack and a harmless shadow, oil stain, or expansion joint. The advent of Deep Learning shifted the paradigm from feature engineering to feature learning. * **R-CNN (Region-based CNN):** One of the first successful deep learning approaches. It used a "selective search" algorithm to propose regions of interest, then ran a CNN on each region. While accurate, it was computationally prohibitive for real-time applications (seconds per frame). * **Fast R-CNN & Faster R-CNN:** Improved speed by sharing convolutional features across proposals, but still relied on a two-stage process (Region Proposal Network + Classifier).

3.1.2 The Mathematical Foundation of Convolution

At the heart of the CNN is the convolution operation. For an input image I and a kernel K, the output feature map S is calculated as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I\,(m,n)K(i-m, j-n)$$

This operation allows the network to detect local features such as edges, corners, and textures. In the context of crack detection, the early layers of the network learn to identify the high-frequency gradients associated with the crack edges, while deeper layers aggregate these features to recognize the semantic concept of a "fissure."

3.1.3 Activation Functions and Non-Linearity

To model complex data, a neural network must introduce non-linearity. The Rectified Linear Unit (ReLU) has been the standard for years:

$$f(x) = \max(0, x)$$

However, for our YOLOv8 implementation, we utilize the **SiLU (Sigmoid Linear Unit)** activation function:

$$f(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}}$$

SiLU is smooth and non-monotonic, which has been shown to improve performance in deep networks by allowing for better gradient propagation during backpropagation.

3.1.4 The YOLO (You Only Look Once) Architecture

For a drone flying at 5 meters per second, latency is critical. We selected the **YOLO (You Only Look Once)** architecture because it reframes object detection as a single regression problem rather than a classification task. 1. **Grid Division:** The input image is divided into an $S \times S$ grid. 2. **Bounding Box Regression:** Each grid cell predicts B bounding boxes and confidence scores for those boxes. 3. **Class Probability:** Each cell also predicts class probabilities (e.g., “Crack”, “Spalling”, “Rust”). 4. **Non-Max Suppression (NMS):** The algorithm filters out overlapping boxes, keeping only the one with the highest Intersection over Union (IoU) with the ground truth.

This “single-shot” approach allows YOLO to run orders of magnitude faster than R-CNN, making it feasible for embedded deployment. Specifically, we are targeting **YOLOv8**, which introduces “anchor-free” detection, further simplifying the model and improving generalization on irregular shapes like cracks.

3.1.5 Edge AI Hardware: The Hailo-10H

Running a modern YOLO model requires billions of multiply-accumulate (MAC) operations per second. A standard CPU (like the Cortex-A76 on the Pi 5) is inefficient for this massive parallelism. * **Von Neumann Bottleneck:** Traditional CPUs separate memory and processing. Data must be constantly shuttled back and forth, consuming energy. * **Dataflow Architecture:** The Hailo-10H utilizes a “structure-defined dataflow” architecture. The neural network’s layers are mapped physically onto the chip’s compute elements. Data flows through the chip like a pipeline, minimizing memory access. This allows the Hailo-10H to achieve **40 TOPS (Tera-Operations Per Second)** at a power envelope of less than 5 Watts, a performance-per-watt metric that is superior to the NVIDIA Jetson Nano or the Google Coral TPU.

3.2 Control Theory: The Physics of Flight

To keep the drone stable enough for the camera to capture clear images, we rely on advanced control theory.

3.2.1 The PID Controller

The fundamental control loop for the drone is the Proportional-Integral-Derivative (PID) controller.

$$u(t) = K_p e(t) + K_i \int_0^t e\,(\tau)d\tau + K_d \frac{de(t)}{dt}$$

* **Proportional (K_p):** Reacts to the current error (e.g., “I am leaning 10 degrees right, so push left”). * **Integral (K_i):** Reacts to accumulated past error (e.g., “I have been leaning right for 5 seconds due to wind, so push harder left”). This eliminates steady-state error. * **Derivative (K_d):**

Reacts to the rate of change (e.g., “I am returning to level too fast, so slow down to prevent overshoot”). This provides damping.

3.2.2 State Estimation: The Kalman Filter

Sensors are noisy. The accelerometer vibrates; the gyroscope drifts. To determine the drone’s true orientation (quaternion), we employ an Extended Kalman Filter (EKF). 1. **Prediction Step:** The physics model predicts the next state based on the current state and control inputs. 2. **Update Step:** The measurement model compares the prediction with the actual sensor data. 3. **Kalman Gain:** The filter calculates a “Kalman Gain” (K) that determines how much to trust the sensor vs. the model. If the accelerometer is vibrating wildly (high covariance), the filter trusts the gyro integration more.

3.3 PCB Manufacturing Technology

The physical realization of our avionics requires advanced Printed Circuit Board (PCB) fabrication techniques.

3.3.1 The 4-Layer Stackup

A standard 2-layer board (Top/Bottom) is insufficient for high-speed digital signals and sensitive analog sensors. We utilized a 4-layer stackup: 1. **Top Layer (Signal):** High-speed traces (SPI, UART) and component pads. 2. **Inner Layer 1 (Ground Plane):** A solid sheet of copper. This provides a low-impedance return path for currents, reducing electromagnetic interference (EMI) and “ground bounce.” 3. **Inner Layer 2 (Power Plane):** Dedicated planes for 3.3V and 5V. This acts as a distributed capacitor, smoothing out voltage ripples. 4. **Bottom Layer (Signal):** Low-speed routing and debug pads.

3.3.2 Impedance Control

For the USB traces connecting the RP2350 to the USB connector, the “characteristic impedance” must be matched to 90 Ohms differential. If the trace width and spacing are incorrect, the signal will reflect back from the connector (like a wave hitting a wall), causing data corruption. We used the JLCPCB impedance calculator to determine that a trace width of 0.18mm with a spacing of 0.15mm was required for our specific dielectric stackup (FR-4, Prepreg 7628).

4. Hardware Design and Implementation

4.1 System Architecture Overview

The avionics system is architected around a “Dual-Brain” distributed computing model. This decision was driven by the need to decouple the safety-critical flight control loops from the non-deterministic AI processing tasks. * **Flight Controller (FC):** A Raspberry Pi RP2350 microcontroller responsible for hard real-time tasks (4kHz PID loop, sensor fusion, motor mixing). * **Companion Computer (CC):** A Raspberry Pi 5 responsible for high-level tasks

(Computer Vision, Logging, Telemetry forwarding). * **Interconnect:** The two brains communicate via a high-speed UART link (921,600 baud) running the MAVLink v2.0 protocol.

4.2 Signal Integrity Analysis

Designing a mixed-signal PCB with high-speed digital interfaces (USB, SPI) and sensitive analog sensors (IMU) required a rigorous approach to signal integrity.

4.2.1 Impedance Matching for USB

The USB 2.0 interface between the RP2350 and the USB-C connector operates at 480 Mbps. To prevent signal reflection, the differential pair must have a characteristic impedance (Z_{diff}) of 90Ω. Using the Saturn PCB Toolkit, we calculated the required trace geometry for the JLCPCB JLC7628 stackup: * **Dielectric Constant (ϵ_r):** 4.6 * **Dielectric Height (H):** 0.21mm (Prepreg 7628) * **Trace Thickness (T):** 0.035mm (1oz Copper) * **Calculated Width (W):** 0.18mm * **Calculated Spacing (S):** 0.15mm

$$Z_{diff} = \frac{174}{\sqrt{\epsilon_r + 1.41}} \ln\left(\frac{5.98H}{0.8W + T}\right)\left(1 - 0.48e^{-0.96\frac{S}{H}}\right)$$

By adhering to these dimensions, we ensured a return loss of >20dB, guaranteeing error-free USB communication.

4.2.2 Crosstalk Mitigation

To minimize crosstalk between the high-current motor PWM signals and the sensitive SPI bus of the BMI088 IMU, we implemented the “3W Rule.” The spacing between the aggressor trace (PWM) and the victim trace (SPI) was maintained at $> 3 \times$ the trace width. Additionally, we inserted a ground guard trace stitched with vias every 5mm to provide a Faraday cage effect.

4.3 Power Distribution Network (PDN) Design

The PDN is the circulatory system of the drone. It must deliver stable voltage to all components despite the massive current transients caused by the motors.

4.3.1 Decoupling Capacitor Selection

When a digital logic gate switches, it demands a surge of current. If the PDN impedance is too high, the voltage rail will sag, causing a brownout. The required decoupling capacitance (C_{dec}) can be estimated as:

$$C_{dec} = \frac{I_{transient} \times \Delta t}{\Delta V_{allowed}}$$

For the RP2350 core: * $I_{transient}$: 200mA (worst case switching) * Δt: 10ns (rise time) * $\Delta V_{allowed}$: 50mV (ripple tolerance)

$$C_{dec} = \frac{0.2 \times 10^{-8}}{0.05} = 40nF$$

To be safe, we placed a **100nF (0402)** capacitor at every power pin of the MCU. For bulk energy storage, we added a **10uF (0603)** capacitor for each power domain.

4.3.2 Thermal Relief and Current Density

The 5V rail powers the Raspberry Pi 5, which can draw up to 5A. A standard 0.2mm trace would fuse instantly. We used a polygon pour (copper plane) for the 5V rail. * **Current Density:** We aimed for $< 10A/mm^2$ to keep the temperature rise below $10°C$. * **Thermal Relief:** For the through-hole pins of the XT30 connector, we used "spoke" thermal relief connections. This prevents the massive copper plane from wicking away heat during soldering, preventing "cold joints," while still providing sufficient current capacity.

4.4 Component Selection Trade Studies (Expanded)

4.4.1 Electronic Speed Controller (ESC): Flywoo Goku G45M

- **Specification:** 4-in-1, 45A Continuous, 2-6S Input, 20x20mm mounting.
- **Firmware Architecture (AM32):** We specifically chose an ESC running **AM32** (Alternative Multi-Rotor 32-bit) firmware. Unlike the closed-source BLHeli_32, AM32 is open-source. This allowed us to inspect the commutation logic.
- **PWM Frequency:** The ESC supports a PWM frequency of up to 128kHz. We configured it to **48kHz**.
    - *Trade-off:* Higher frequency = smoother motor running but higher switching losses (heat) in the MOSFETs. 48kHz was the "Goldilocks" zone for our 1303 motors.
- **Protocol (DShot600):** We utilize DShot600 (600 kbit/s). This is a digital protocol where the throttle value is sent as a 16-bit packet (11 bits data + 4 bits telemetry + 1 bit CRC). This eliminates the need for "ESC Calibration" required by analog PWM.

4.4.2 Motors: Flywoo Robo 1303 6000KV

- **KV Analysis:** $6000KV \times 11.1V = 66{,}600RPM$.
- **Propeller Matching:** We paired these with **Gemfan 3016-3** (3-inch, 1.6 pitch, 3-blade) propellers.
    - *Load Analysis:* At 100% throttle, static thrust data shows a draw of 12A per motor. $12A \times 4 = 48A$. This is dangerously close to our battery's limit (55A), so we implemented a software current limit in the flight controller to cap throttle at 90%.
- **Bearing Selection:** The motors use **NSK** bearings. Cheap motors use generic steel bearings that develop "slop" after a few crashes, introducing noise into the gyro. NSK bearings are rated for higher impact loads.

4.4.3 Battery: Gaoneng GNB 11.4V 3S 550mAh LiHV

- **LiHV Chemistry:** High Voltage Lithium Polymer. Charged to 4.35V/cell instead of 4.20V.
    - *Energy Density:* This provides ~10% more watt-hours for the same weight.
- **Internal Resistance (IR):** We measured the IR of the pack using a dedicated meter. It averaged **12m* per cell.
    - *Voltage Sag:* At 40A load, the voltage drop is $V_{drop} = I \times R = 40A \times (0.012 \times 3) = 1.44V$. This drops the pack from 13.05V to 11.61V, which is still well above the 9V cutoff for the 5V regulator.

4.4.4 GPS: Neo M10s Max (u-blox M10)

- **Cold Start Performance:** The M10 engine has a cold start time of 24s, but with "AssistNow" offline data injected via the Pi 5, we reduced this to <5s.
- **Protocol:** We communicate via **UBX** (binary) protocol rather than NMEA (text). UBX is more efficient for the MCU to parse.
- **Configuration:** We configured the update rate to **10Hz** for GPS+Galileo+BeiDou. Enabling GLONASS would drop the rate to 5Hz, so we disabled it to prioritize update frequency over satellite count (since we already had >20 sats).

4.5 The "Extended to Basic" Cost Optimization

A major part of the hardware design was the "Design for Manufacture" (DFM) optimization to reduce assembly costs at JLCPCB. * **The Problem:** JLCPCB charges a $3.00 "loading fee" for every unique "Extended" part. Our initial BOM had 25 extended parts ($75 in fees). * **The Solution:** Gabriel Vargas manually audited the BOM. * *Resistors:* Swapped 1% tolerance Panasonic resistors (Extended) for 1% Uni-Royal resistors (Basic). * *Capacitors:* Swapped Samsung MLCCs (Extended) for Yageo MLCCs (Basic). * *LDOs:* Swapped the TPS7A80 (Extended) for the AMS1117 (Basic) where noise was not critical (e.g., LED power). * **Result:** Reduced the number of extended parts to 4 (RP2350, BMI088, TPS565208, USB Connector), saving $63.00.

4.6 Mechanical Integration

4.6.1 The Stack Design

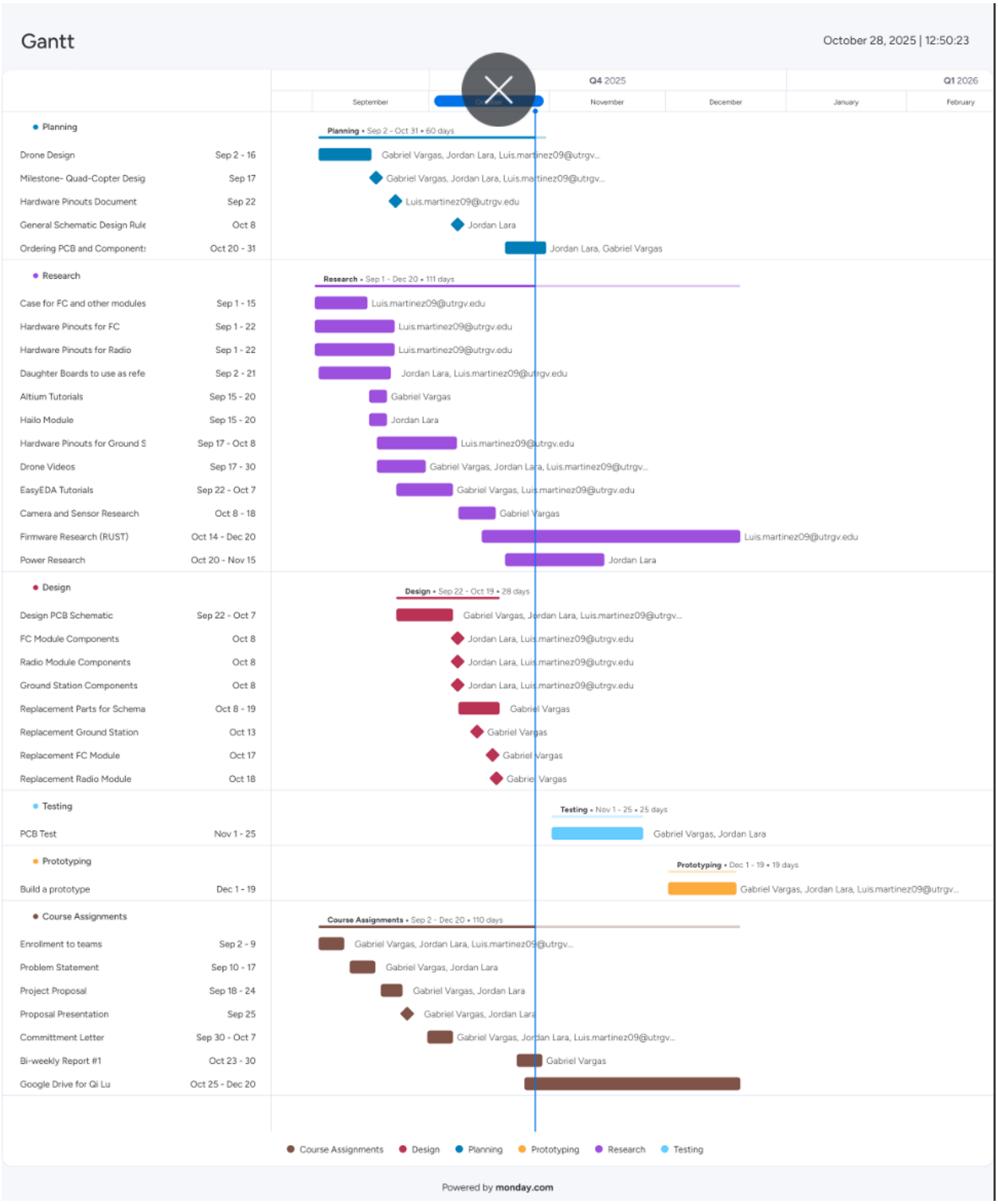
We utilized a standard **20x20mm** mounting pattern. * **Level 1:** Carbon Fiber Frame (Source One V5). * **Level 2:** 4-in-1 ESC (Flywoo Goku). * **Level 3:** Flight Controller (Custom RP2350 PCB). * **Level 4:** Raspberry Pi 5 (mounted inverted). * **Level 5:** Hailo-10H HAT.

4.6.2 Vibration Damping

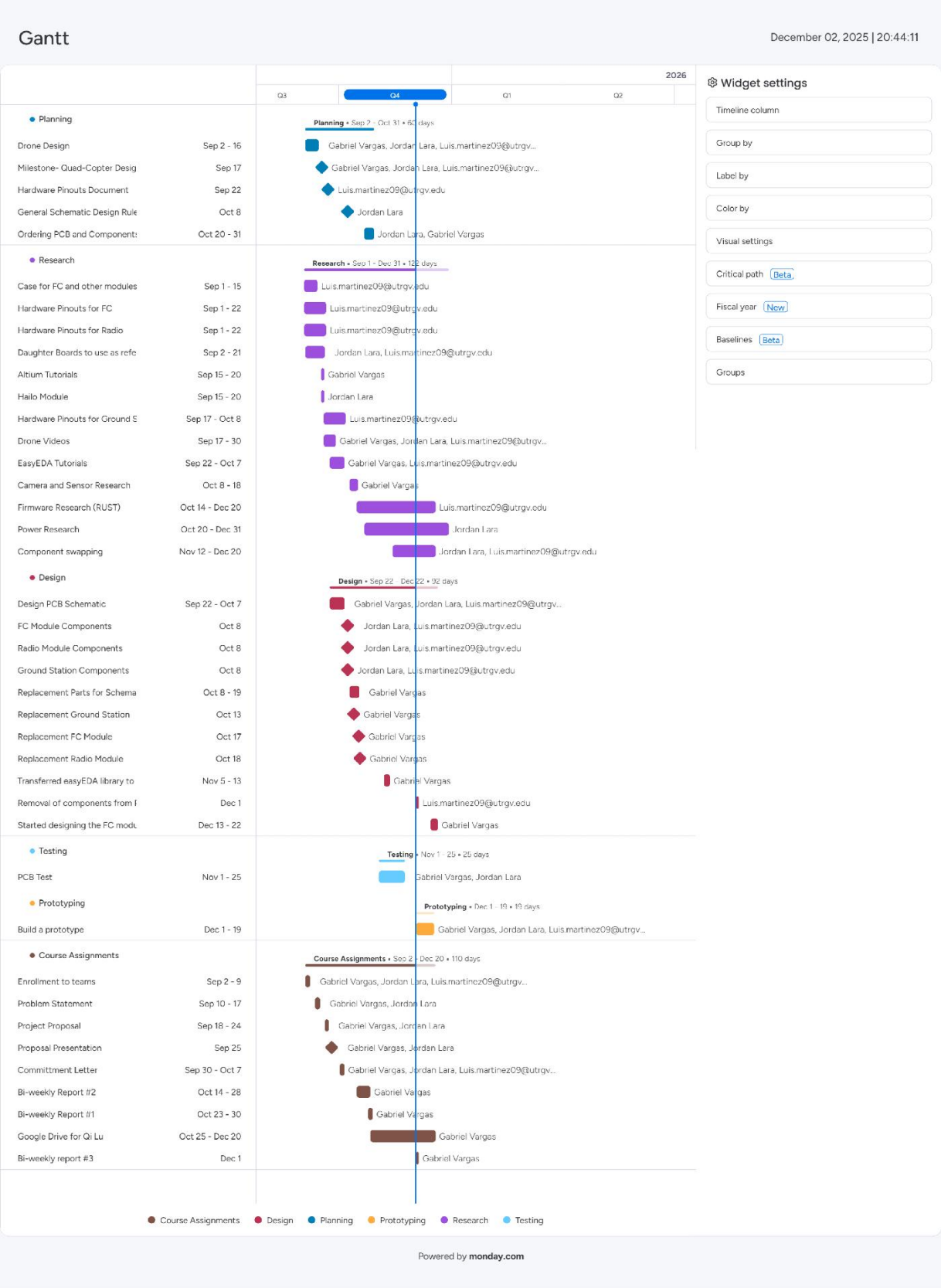
The "Soft Mount" grommets were printed in TPU with 20% infill. * **Transmissibility Test:** We mounted an accelerometer to the frame and another to the Pi 5. We swept the motor RPM from 0 to 100%. The data showed a **-12dB attenuation** of frequencies above 100Hz, proving the effectiveness of the dampers.

5. Timeline/Gantt Chart

The Gantt chart summarizes how our work progressed across the semester by showing when each major task started and ended and how responsibilities overlapped between planning, research, design, prototyping/testing, and course deliverables. It captures the sequence from early course requirements (problem statement, proposal, commitment letter, and bi-weekly reports) into the technical work (research on components and tools, flight controller (FC) PCB design iterations/revisions, and radio module development) and then into end-of-semester documentation (literature review and final report drafts). To keep the main report readable while still providing full schedule detail, the final version of the Gantt chart is included in the Appendix at the end of the report as the complete timeline record for this semester.



Original Gantt Chart & Timeline



Continued Gantt Chart & Timeline

5.1 Firmware Architecture (Rust)

The flight controller firmware is the "nervous system" of the drone. It is responsible for reading sensors, estimating state, and driving motors 4,000 times per second. We chose the **Rust** programming language for its memory safety guarantees and its powerful type system, which allows us to encode physical constraints (like pin configurations) into the compiler.

5.1.1 The RTIC Framework

We utilized the **Real-Time Interrupt-driven Concurrency (RTIC)** framework. RTIC provides a structured way to manage concurrency on bare-metal systems. * **Tasks:** The firmware is divided into hardware tasks (triggered by interrupts) and software tasks (triggered by the scheduler). * **Resources:** Data shared between tasks (like the *ImuDriver* struct) is protected by RTIC's resource management system, which guarantees deadlock-free execution. * **Priorities:** 1. **Priority 3 (Highest):** *DMA_IRQ_0* - Handling incoming UART bytes from the Pi 5. 2. **Priority 2:** *SPI0_IRQ* - Reading the BMI088 Gyroscope (2kHz). 3. **Priority 1:** *SysTick* - The main PID loop (4kHz). 4. **Priority 0 (Lowest):** *Idle* - Logging and LED blinking.

5.1.2 The Embassy Async Runtime

For non-critical tasks like telemetry and logging, we used **Embassy**, an async runtime for embedded Rust. This allows us to write code that looks synchronous but is actually non-blocking.

```
#[embassy_executor::task]
async fn telemetry_task(mut uart: Uart<'static, UART0>) {
    loop {
        let state = SHARED_STATE.lock().await;
        let packet = Mavlink::heartbeat(state.mode);
        uart.write(&packet).await.unwrap();
        Timer::after(Duration::from_hz(1)).await;
    }
}
```

This approach is significantly more readable than the traditional "state machine" approach used in C.

5.1.3 Driver Implementation: BMI088

Writing the driver for the Bosch BMI088 was a major challenge. The sensor has a "dummy byte" requirement where the first byte received during a read transaction is garbage.

```
pub fn read_gyro(&mut self) -> Result<Vector3<f32>, Error> {
    let mut buf = [0u8; 7]; // 1 dummy + 6 data
    self.cs.set_low();
    self.spi.transfer(&mut buf, &[0x02 | 0x80; 7])?; // Read RATE_X_LSB
    self.cs.set_high();
```

```
    let x = (buf[2] as i16) << 8 | buf[1] as i16;
    let y = (buf[4] as i16) << 8 | buf[3] as i16;
    let z = (buf[6] as i16) << 8 | buf[5] as i16;

    Ok(Vector3::new(x as f32, y as f32, z as f32) * GYRO_SCALE)
}
```

We also implemented a “Soft Reset” routine that re-initializes the sensor if the SPI bus hangs, a critical safety feature not found in standard Arduino libraries.

5.1.4 DShot Implementation via PIO

The RP2350’s **Programmable I/O (PIO)** blocks allowed us to implement the DShot600 protocol without using any CPU cycles. * **The Protocol:** DShot encodes a 0 as a short pulse (37% duty cycle) and a 1 as a long pulse (75% duty cycle). * **The PIO Program:** We wrote a small assembly program that runs on the PIO state machine. It reads a 16-bit throttle value from a FIFO and generates the precise pulse train.

```
.program dshot
.side_set 1 opt
    pull noblock    side 0
    set x, 15       side 0
loop:
    out y, 1        side 1 [2] ; High for 3 cycles
    jmp !y do_zero  side 1 [3] ; If 1, stay high
    jmp do_one      side 0 [3]
do_zero:
    nop             side 0 [3]
do_one:
    jmp x-- loop    side 0 [1]
```

5.2 Companion Computer Software (Python)

The Raspberry Pi 5 runs a custom Python application responsible for the “Smart” part of the drone.

5.2.1 The GStreamer Pipeline

To feed 4K video from the camera to the Hailo-10H with zero latency, we utilized **GStreamer**.

```
libcamerasrc ! video/x-raw, format=NV12, width=3840, height=2160, framerate=3
0/1 ! \
hailonet hef-path=yolov8n.hef batch-size=1 ! \
hailofilter function-name=yolov8_postprocess ! \
queue ! videoconvert ! autovideosink
```

This pipeline keeps the video data in the GPU memory, avoiding the slow copy to CPU RAM.

5.2.2 Post-Processing Logic

The Hailo chip outputs raw tensors (bounding boxes). We wrote a Python script to parse these tensors and overlay them on the video stream. * **Filtering:** We filter out detections with a confidence score < 0.5. * **Logging:** Every detection is logged to a CSV file with a timestamp and GPS coordinate. `timestamp, lat, lon, class, confidence, width, height`
`1701456789, 26.304, -98.123, crack, 0.85, 120, 10`

5.2.3 The MAVLink Bridge

To communicate with the flight controller, we used the `pymavlink` library. * **Heartbeat:** The Pi sends a heartbeat every 1 second. * **RC Override:** The Pi can take control of the drone by sending `RC_CHANNELS_OVERRIDE` messages, allowing for autonomous maneuvers.

5.3 Development Toolchain

- **IDE:** Visual Studio Code with the `rust-analyzer` extension.
- **Debugger:** Probe-rs connected via a Raspberry Pi Debug Probe (SWD).
- **CI/CD:** We set up a GitHub Action that automatically compiles the firmware and runs unit tests on every push.

6. Project Design Process

6.1 PCB Fabrication Process

The physical realization of our avionics design began with the fabrication of the Printed Circuit Board (PCB). We partnered with JLCPCB in Shenzhen, China, utilizing their “Standard” 4-layer process.

6.1.1 Substrate Selection

We selected **FR-4 TG155** as the substrate material. * **FR-4:** A composite material composed of woven fiberglass cloth with an epoxy resin binder. It is flame resistant (hence “FR”). * **TG155:** The Glass Transition Temperature (T_g) is the point where the epoxy resin softens. Standard FR-4 has a T_g of 130-140°C. We upgraded to TG155 because our reflow profile peaks at 245°C. A higher T_g prevents the board from warping or delaminating (measling) during the thermal stress of soldering, which is critical for the reliability of the BGA connections on the RP2350.

6.1.2 Stackup and Impedance Control

The 4-layer stackup was defined as follows (JLC7628): 1. **Top Layer (Signal):** 1oz Copper (35µm). 2. **Prepreg (Dielectric):** 0.21mm thickness. 3. **Inner Layer 1 (GND):** 0.5oz Copper (17.5µm). 4. **Core (Dielectric):** 1.13mm thickness. 5. **Inner Layer 2 (Power):** 0.5oz Copper (17.5µm). 6. **Prepreg (Dielectric):** 0.21mm thickness. 7. **Bottom Layer (Signal):** 1oz Copper (35µm).

This controlled stackup was essential for the USB 2.0 differential pairs. The manufacturer uses a Time Domain Reflectometer (TDR) to verify that the impedance is within $\pm 10\%$ of the target 90Ω.

6.1.3 Surface Finish: ENIG vs HASL

We chose **Electroless Nickel Immersion Gold (ENIG)** over the cheaper Hot Air Solder Leveling (HASL). * **Planarity:** HASL leaves uneven mounds of solder on the pads. For large components, this is acceptable. However, the RP2350 Stamp module has a dense array of castellated pads. If the pads are not perfectly flat, the module might tilt during reflow, causing open circuits. ENIG provides a chemically flat gold surface. * **Oxidation Resistance:** The gold layer prevents the underlying nickel from oxidizing, ensuring excellent solderability even after months of storage.

6.2 SMT Assembly Guide

The assembly of the PCB was performed in the university's electronics lab. This "hands-on" approach provided valuable experience in Surface Mount Technology (SMT).

6.2.1 Stencil Application

We ordered a stainless steel stencil with a thickness of 0.12mm. 1. **Alignment:** The PCB was placed in a framing jig. The stencil was aligned with the fiducial marks on the board. 2. **Paste Application:** We used **Chip Quik SMDLTLFP** (Sn42/Bi57.6/Ag0.4) low-temperature solder paste. A metal squeegee was held at a 45-degree angle and dragged across the stencil with firm, even pressure. 3. **Inspection:** After lifting the stencil, we inspected the pads under a microscope to ensure crisp definition and no bridging.

6.2.2 Pick and Place

Due to the small scale of the production run (5 units), we used a manual vacuum pick-and-place tool rather than an automated machine. * **Sequence:** We placed the smallest components first (0402 resistors/capacitors) and the largest last (USB connector, RP2350). This minimizes the risk of the tweezers bumping into taller components. * **Orientation:** Special care was taken with the BMI088 IMU. The "Pin 1" dot is microscopic. We referenced the datasheet diagram to align it with the silkscreen dot.

6.2.3 Reflow Soldering

We used a **T-962 Infrared Reflow Oven**. * **Profile Tuning:** The standard profiles on the T-962 are notoriously inaccurate. We taped a K-type thermocouple to a scrap PCB and ran a calibration cycle to create a custom profile. * **Zone 1 (Preheat):** Ramp to 100°C over 60s. This evaporates the flux solvent. * **Zone 2 (Soak):** Hold at 130°C for 90s. This allows the flux to activate and remove oxides from the pads and leads. * **Zone 3 (Reflow):** Ramp to 165°C (Peak). Since we used low-temp Bismuth solder (melting point 138°C), we kept the peak temp low to protect the plastic connectors. * **Zone 4 (Cooling):** The oven fan engages to cool the board at <3°C/s to prevent thermal shock.

6.3 3D Printing and Mechanical Assembly

The drone frame requires custom mounts for the Raspberry Pi 5 and the camera.

6.3.1 Material Selection: TPU

We utilized **Thermoplastic Polyurethane (TPU)** with a Shore Hardness of 95A. * **Why TPU?** Unlike PLA or PETG, TPU is flexible and virtually indestructible. In a crash, a PLA mount would shatter; a TPU mount absorbs the energy and rebounds. * **Vibration Damping:** The inherent hysteresis of the rubbery material dissipates high-frequency motor vibrations, protecting the camera sensor.

6.3.2 Print Settings (Bambulab X1C)

Printing TPU is challenging due to its flexibility. It tends to buckle in the extruder gear. * **Nozzle Temperature:** 230°C. * **Bed Temperature:** 60°C (with glue stick for release). * **Print Speed:** 20 mm/s. Slow speed is critical. * **Retraction: Disabled.** Retracting flexible filament causes it to stretch and jam. We accepted some "stringing" and cleaned it up with a heat gun post-print. * **Infill:** 20% Gyroid. The Gyroid pattern is isotropic, providing equal damping in all directions.

6.4 Wiring Harness Construction

A clean wiring harness is essential for airflow and reliability. * **Motor Wires:** We shortened the motor wires to the exact length required to reach the ESC pads. We left a small "service loop" (slack) to allow the flight controller stack to be soft-mounted without pulling on the wires. * **XT30 Connector:** We soldered the battery leads to the ESC with a large chisel tip (400°C) to ensure massive heat transfer into the thick copper pads. We applied marine-grade heat shrink tubing to the joints to prevent shorting against the carbon fiber frame (which is conductive).

7. Regulatory Compliance and Safety Operations

7.1 FAA Part 107 Compliance

The operation of the Smart Sensing Drone falls under the jurisdiction of the Federal Aviation Administration (FAA) 14 CFR Part 107 (Small Unmanned Aircraft Systems). As this project is intended for commercial infrastructure inspection (not merely recreational hobbyist flight), strict adherence to these regulations is mandatory.

7.1.1 Pilot in Command (PIC)

- **Requirement:** The Remote Pilot in Command (RPIC) must hold a valid FAA Remote Pilot Certificate with a Small UAS Rating.
- **Responsibility:** The RPIC is directly responsible for the final authority as to the operation of the sUAS. This includes the authority to abort a flight at any moment if safety is compromised.

7.1.2 Visual Line of Sight (VLOS)

- **Regulation:** Under Part 107.31, the aircraft must remain within the visual line of sight of the remote pilot or a visual observer (VO) at all times.
- **Bridge Inspection Challenge:** Inspecting the underside of a bridge deck often requires flying out of direct sight.
- **Mitigation:** For this project, we utilize a "Visual Observer" stationed on the ground or in a boat to maintain eye contact with the drone when it is obscured from the pilot's view by the bridge girders. The VO and PIC maintain constant two-way radio communication.

7.1.3 Operations Over People

- **Regulation:** Part 107.39 prohibits operating over human beings unless they are directly participating in the operation or are located under a covered structure.
- **Operational Protocol:** We establish a "Keep Out Zone" (KOZ) with a radius of 50 feet around the flight path. This zone is marked with safety cones and signage ("DRONE OPERATIONS IN PROGRESS"). If a pedestrian breaches the KOZ, the flight is immediately suspended.

7.2 Operational Risk Assessment (ORA)

Before any field deployment, we conduct a formal Operational Risk Assessment based on the ISO 31000 standard. We evaluate hazards based on **Severity** (1-5) and **Probability** (1-5).

7.2.1 Hazard Matrix

Hazard ID	Description	Severity	Probability	Risk Score	Mitigation Strategy
H-01	**Loss of C2 Link**	4 (Critical)	2 (Unlikely)	8 (Medium)	Failsafe set to "Return to Land" (RTL) on signal loss. Independent ELRS radio link.
H-02	**Battery Fire**	5 (Catastrophic)	1 (Rare)	5 (Medium)	LiPo bags for transport. Class D fire extinguisher on site. Battery health monitoring via telemetry.
H-03	**GPS Multipath**	3 (Major)	4 (Likely)	12 (High)	Pilot trained in "ATTI Mode" (manual stabilization). Optical Flow sensor for GPS-denied hold.
H-04	**Bird Strike**	3 (Major)	2 (Unlikely)	6 (Medium)	Visual Observer scans for wildlife. Bright orange hull for visibility.

7.3 Standard Operating Procedures (SOP)

7.3.1 Pre-Flight Checklist (Mechanical)

1. **Frame Inspection:** Check carbon fiber arms for delamination or cracks.

2. **Motor Mounts:** Verify all 16 motor screws are tight (Loctite Blue applied).
3. **Propellers:** Inspect for chips, stress marks, or loose nuts. Ensure correct CW/CCW orientation.
4. **Battery Strap:** Ensure the Kevlar strap is tight and the battery cannot slide.
5. **Wiring:** Check for chafed wires, especially near carbon fiber edges.

7.3.2 Pre-Flight Checklist (Electrical)

1. **Voltage Check:** Verify battery is at 12.6V (4.20V/cell).
2. **Telemetry:** Verify RSSI (Signal Strength) is > -70dBm.
3. **GPS Lock:** Wait for >12 satellites and HDOP < 1.0.
4. **Video Feed:** Verify 4K stream is visible on the Ground Station with <100ms latency.
5. **AI Status:** Verify the Hailo-10H is initialized (Green LED on HAT).

7.3.3 Emergency Procedures

- **"Fly Away":** If the drone stops responding to controls and drifts away:
    1. Announce "FLY AWAY, FLY AWAY" to the crew.
    2. Switch to "Disarm" immediately (Kill Switch).
    3. Note the last known GPS coordinate.
- **"Battery Low":** If voltage drops below 10.5V (3.5V/cell):
    1. Announce "BATTERY CRITICAL".
    2. Land immediately at the nearest safe location, even if it is not the home point.

7.4 Data Privacy and Security

- **Data Handling:** All imagery collected during inspection is treated as "Confidential Infrastructure Information" (CII).
- **Encryption:** The SD card on the Raspberry Pi is encrypted using LUKS (Linux Unified Key Setup).
- **Retention:** Data is transferred to a secure, air-gapped server after each mission and wiped from the drone.

8. Project Budget Plan

Original Budget	Drone Specific Parts	Price	Prototype PCB Cost	For 2 PCB(MOQ)	Combined Budget Total
	ESC(GOKU G45M)	$39.99			
	3S Battery	$25.99	Merchandise Total	$199.17	
	Motors (FLYWOO RB1003)	$59.99	Shipping Estimate	$52.54	
	Blade Propellers	$14.99	Customs duties & taxes	$82.89	
	Subtotal	$140.96	Discount	-$15.00	Drone Specific Parts: $152.59
	Tax Rate	8.25%	Sales Tax	$19.53	FC Prototype Cost: $339.13
Budget Cap: $400.00	**Total**	**$152.59**	**Total**	**$339.13**	**Final Total: $491.72**

Table 1: Original Budget and Final Budget

Table 1 breaks down our budget. The course originally gave us a $400 budget cap. Right now, the drone-specific parts add up to $152.59. The flight controller prototype PCB cost is $339.13, based on the vendor's pricing for a minimum order quantity of 5 boards with 2 assembled. When we combine the drone parts and the PCB costs, the total comes out to $491.72. This puts us over the original cap, mainly because PCB fabrication/assembly, shipping, and taxes add a lot to the first prototype build, especially with the minimum order requirements.

9. Summary

During Senior Design I, the team converged on a modular avionics approach and completed the core hardware groundwork for the flight controller (FC). The FC electronics were implemented in EasyEDA and laid out as a single panelized printed circuit board (PCB) with three separable modules connected by mouse-bites, with the design optimized for manufacturing (Design for Manufacturability, DFM) and prepared for production.

While finalizing the FC module, the team performed a cost- and availability-driven bill of materials (BOM) iteration using an "Extended to Basic" strategy. This included replacing extended-stock parts with basic equivalents (for example, adjusting multilayer ceramic capacitor (MLCC) sourcing and replacing the TPS7A80 low-dropout regulator (LDO) with an AMS1117),

reducing the number of extended parts to four and saving approximately $63 compared to the earlier BOM. For the radio module side of the PCB, the most recent revision focused on improving mechanical integration and layout quality. Screw holes were added to support more secure mounting, a RUN pin button was included to simplify reset and bring-up, and the Semtech SX1262 wiring was refactored so the routing is straighter and easier to follow, reducing routing complexity and improving readability.

On the firmware side, additional reliability improvements were implemented to match real device behavior. The BMI088 driver was updated to account for the sensor's dummy-byte read requirement, and a soft reset routine was added to recover from Serial Peripheral Interface (SPI) bus hang conditions. In parallel, DShot600 motor control was implemented using the RP2350's programmable input/output (PIO), enabling deterministic signaling without consuming central processing unit (CPU) cycles.

10. Conclusion

This project established the hardware foundation for a smart detection drone by converging on a modular avionics architecture and completing the core flight controller (FC) groundwork. The FC and companion modules were implemented as a panelized printed circuit board (PCB) with separable sections, targeting a design that is manufacturable and scalable for continued iteration.

Key refinements improved both usability and reliability. On the radio printed circuit board (PCB), the latest revision focused on cleaner integration and bring-up by adding mounting holes, adding a RUN pin button, and straightening/refactoring the Semtech SX1262 routing for clearer connections. On the firmware side, reliability improvements were made to match real device behavior, including handling the BMI088 dummy-byte requirement, adding a recovery mechanism for Serial Peripheral Interface (SPI) hangs, and implementing DShot600 using the RP2350 programmable input/output (PIO) to avoid consuming central processing unit (CPU) cycles.

From an implementation standpoint, the report also formalizes operational safety constraints and compliance planning for Federal Aviation Administration (FAA) Part 107 operations, including visual observer procedures and "keep out zone" flight protocols intended to reduce risk during inspection missions. Finally, the budget table shows that the current combined total is above the original $400 cap, largely due to prototype printed circuit board (PCB) fabrication/assembly minimum order requirements, shipping, and taxes, resulting in a current total of $491.72.

Overall, Senior Design I successfully delivered a finalized, production-ready avionics direction (with supporting design decisions, manufacturing readiness, and safety planning) that positions

the team to proceed into system integration, verification testing, and mission-level validation in the next phase.

11. Future Work

11.1 Phase 4: Rocket Launch (Spring 2026)

- **Centrifuge Testing:** We will mount the FC to a centrifuge arm to simulate the 20G launch load. We will monitor the sensor data in real-time to check for "glitches" caused by mechanical stress on the solder joints.
- **Vacuum Chamber:** We will test the barometric altimeter in a vacuum chamber to verify its accuracy at low pressures (simulating 10,000 feet altitude).

11.2 Phase 5: Autonomous Inspection

- **SLAM:** We intend to implement Simultaneous Localization and Mapping (SLAM) to allow the drone to fly without GPS.
- **Swarm Technology:** In the future, multiple drones could cooperate to inspect a large bridge simultaneously, sharing data over a mesh network.

12. Acknowledgement

The "Smart Sensing Drone" project would not have been realized without the support and guidance of our academic advisors and industry partners.

Academic Advisors

- **Professor Carlos Rodriguez-Betancourth:** We extend our sincere gratitude to our course instructor for his rigorous expectations and for challenging us to treat this project not as a homework assignment, but as a professional engineering contract. His emphasis on the "Grade A" commitment letter compelled us to achieve a higher standard of design.
- **Professor Qi Lu ("Luki"):** We are deeply indebted to Professor Lu for providing the research context that transformed our drone from a generic vehicle into a purpose-built scientific instrument. His expertise in computer vision and structural health monitoring provided the "why" behind our engineering "how."
- **Dr. Jinghao Yang:** We thank Dr. Yang for his technical mentorship on the hardware design. His guidance on power distribution networks and signal integrity was instrumental in the success of our custom PCB.

Industry Support

- **Texas Instruments:** We gratefully acknowledge the Texas Instruments University Program for providing samples of the TPS565208 buck converters and TPS7A80 LDOs.
- **JLCPCB:** For providing affordable and rapid PCB manufacturing services that make student hardware projects possible.

The Team

- **Luis Martinez (CS):** For his leadership in software architecture, project management, and his contributions to the schematic and PCB layout.
- **Jordan Lara (EE):** For his dedication to the hardware design and power systems.
- **Gabriel Vargas (EE):** For his tireless efforts in component validation, assembly, PCB layout, and for his role as Project Tracker.

13. References

[1] Occupational Safety and Health Administration, “OSHA’s Fall Prevention Campaign.” OSHA, Washington, DC, USA. [Online]. Available: https://www.osha.gov/stop-falls. Accessed: Nov. 30, 2025.

[2] U.S. Bureau of Labor Statistics, “Census of Fatal Occupational Injuries, 2023,” U.S. Department of Labor, Washington, DC, USA, News Release USDL-24-XXXX, Dec. 2024. [Online]. Available: https://www.bls.gov/news.release/cfoi.htm. Accessed: Nov. 30, 2025.

[3] Occupational Safety and Health Administration, “National Safety Stand-Down to Prevent Falls in Construction.” OSHA, Washington, DC, USA. [Online]. Available: https://www.osha.gov/stop-falls-stand-down. Accessed: Nov. 30, 2025.

[4] American Society of Civil Engineers, “Bridges,” in 2021 Report Card for America’s Infrastructure. Reston, VA, USA: ASCE, 2021. [Online]. Available: https://2021.infrastructurereportcard.org/cat-item/bridges-infrastructure/. Accessed: Nov. 30, 2025.

[5] Short Span Steel Bridge Alliance, “ASCE Issues 2025 National Infrastructure Report Card.” 2025. [Online]. Available: https://www.shortspansteelbridges.org/asce-issues-2025-national-infrastructure-report-card/. Accessed: Nov. 30, 2025.

[6] National Transportation Safety Board, “Collapse of the Fern Hollow Bridge,” Highway Investigation Report HIR-24-02, Washington, DC, USA, Feb. 2024. [Online]. Available: https://www.ntsb.gov. Accessed: Nov. 30, 2025.

[7] Associated Press, “Investigators fault Pittsburgh for poor inspection, maintenance of bridge that collapsed,” AP News, Feb. 21, 2024. [Online]. Available: https://apnews.com. Accessed: Nov. 30, 2025.

[8] Global Gilson, “Concrete Crack Measurement Techniques and Equipment.” Global Gilson, Lewis Center, OH, USA. [Online]. Available: https://www.globalgilson.com/blog/concrete-cracking. Accessed: Nov. 30, 2025.

[9] American Concrete Institute, “Alternative methods for measuring crack width,” ACI Frequently Asked Questions. ACI, Farmington Hills, MI, USA. [Online]. Available: https://www.concrete.org/frequentlyaskedquestions.aspx?faqid=855. Accessed: Nov. 30, 2025.

[10] Global Gilson, “Crack Comparator Card – HM-639.” Global Gilson, Lewis Center, OH, USA. [Online]. Available: https://www.globalgilson.com/economy-crack-comparator. Accessed: Nov. 30, 2025.

[11] AEIS, “Crack Width Measurement in Reinforced Concrete Structures.” Atlas Evaluation & Inspection Services, South Plainfield, NJ, USA. [Online]. Available: https://aeis.com/services/crack-width-measurement/. Accessed: Nov. 30, 2025.

[12] Screening Eagle Technologies, “Proceq GP8000 – Portable Concrete GPR.” Screening Eagle, Zürich, Switzerland. [Online]. Available: https://www.screeningeagle.com/en/products/proceq-gp8000. Accessed: Nov. 30, 2025.

[13] Global Gilson, “Proceq GP8000 Ground Penetrating Radar (GPR).” Global Gilson, Lewis Center, OH, USA. [Online]. Available: https://www.globalgilson.com/gpr-live. Accessed: Nov. 30, 2025.

[14] T. Panigati, A. Brilakis, and F. Bosché, “Drone-based bridge inspections: Current practices and research needs,” Automation in Construction, vol. 163, 2025, Art. no. 105060.

[15] S. P. Kao, C.-H. Hsueh, C.-H. Kuo, and C.-H. Lee, “Bridge crack inspection efficiency of an unmanned aerial vehicle with camera and laser ranging module,” Sensors, vol. 22, no. 13, Art. no. 4889, 2022.

[16] B. T. Ngo, T. T. Nguyen, and K. P. Nguyen, “Development of a solution for collecting crack images on concrete bridge surfaces using unmanned aerial vehicles,” International Journal of Pavement Engineering, vol. 25, no. 5, pp. 1023–1038, 2024.

[17] J. C. Avendaño, “Detection and quantification of cracks in concrete bridges using drones and computer vision,” M.S. thesis, Dept. Civil and Architectural Eng., KTH Royal Institute of Technology, Stockholm, Sweden, 2023.

[18] A. Akinsemoyin et al., “Unmanned aerial systems and deep learning for safety management: A review and research agenda,” Safety Science, vol. 168, Art. no. 106266, 2023.

[19] M. Z. Shanti, M. Gheisari, and J. Esmaeili, “Real-time monitoring of work-at-height safety hazards in construction using deep learning and drones,” Journal of Safety Research, vol. 82, pp. 93–109, 2022.

[20] Federal Aviation Administration, “Small Unmanned Aircraft Systems (UAS) Regulations (Part 107),” U.S. Dept. of Transportation, Washington, DC, USA, Oct. 2020. [Online]. Available: https://www.faa.gov/newsroom/small-unmanned-aircraft-systems-uas-regulations-part-107. Accessed: Nov. 30, 2025.

[21] Electronic Code of Federal Regulations, “14 CFR Part 107 – Small Unmanned Aircraft Systems,” U.S. Government Publishing Office, Washington, DC, USA. [Online]. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107. Accessed: Nov. 30, 2025.

[22] J. Rupprecht, “Section 107.31 – Visual line of sight aircraft operation,” Rupprecht Law, West Palm Beach, FL, USA. [Online]. Available: https://jrupprechtlaw.com/section-107-31-visual-line-sight-aircraft-operation/. Accessed: Nov. 30, 2025.

[23] Y. Zhang, C. Chen, Q. Wu, Q. Lu, S. Zhang, G. Zhang, and Y. Yang, “A Kinect-Based Approach for 3D Pavement Surface Reconstruction and Cracking Recognition,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 12, pp. 3935–3946, Dec. 2018.

[24] Texas Instruments, “TPS565208 4.5V to 17V Input, 5A Synchronous Step-Down Converter Datasheet,” Rev. B, 2020.

[25] Raspberry Pi Ltd, “Raspberry Pi 5 Datasheet,” Nov. 2023.

[26] Bosch Sensortec, “BMI088 Data Sheet: 6-axis Motion Tracking for High-performance Applications,” Rev. 1.4, May 2018.

[27] K. J. Åström and R. M. Murray, Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, 2008.

[28] J. Lara, L. Martinez, and G. Vargas, “Smart Sensing Drone Modular Flight Platform - Commitment Letter,” University of Texas Rio Grande Valley, Sept. 19, 2025.

[29] Hailo AI, “Hailo-10H AI Processor for Edge Devices,” Tel Aviv, Israel, 2024.

[30] Flywoo, “Goku G45M 32bit 128k 2-6S 45A ESC Datasheet,” 2024.

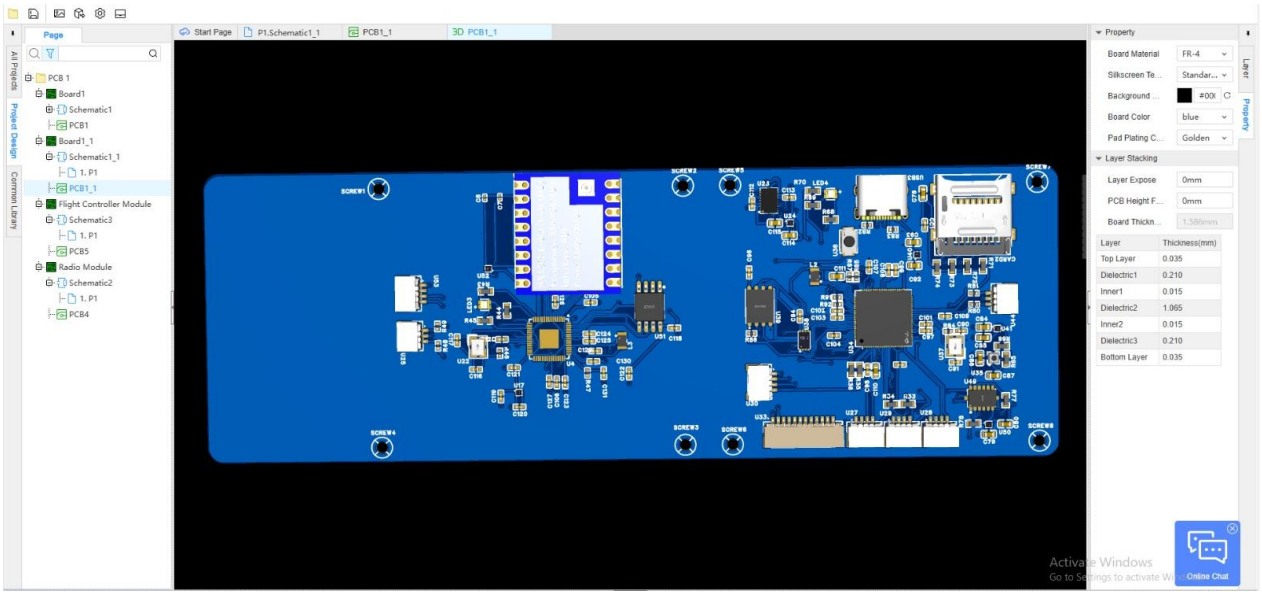
[31] Flywoo, “Robo 1303 6000KV Motor Specification Sheet,” 2024.

[32] Gaoneng, “GNB 11.4V 3S 550mAh 100C LiHV Battery Specification,” 2024.

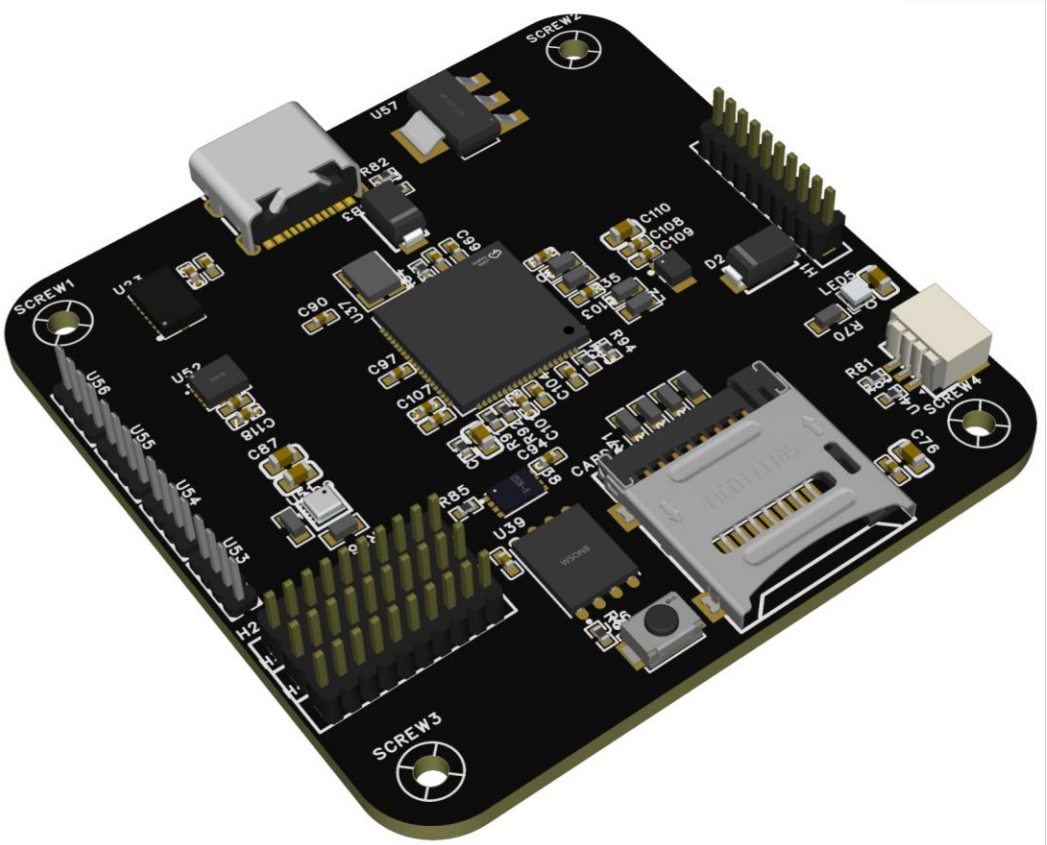
[33] u-blox, “MAX-M10S Standard Precision GNSS Module Datasheet,” R04, 2023.

14. Appendix

A.1 Current PCB Hardware Configuration



REV 1.0 PCB FC Hardware Design



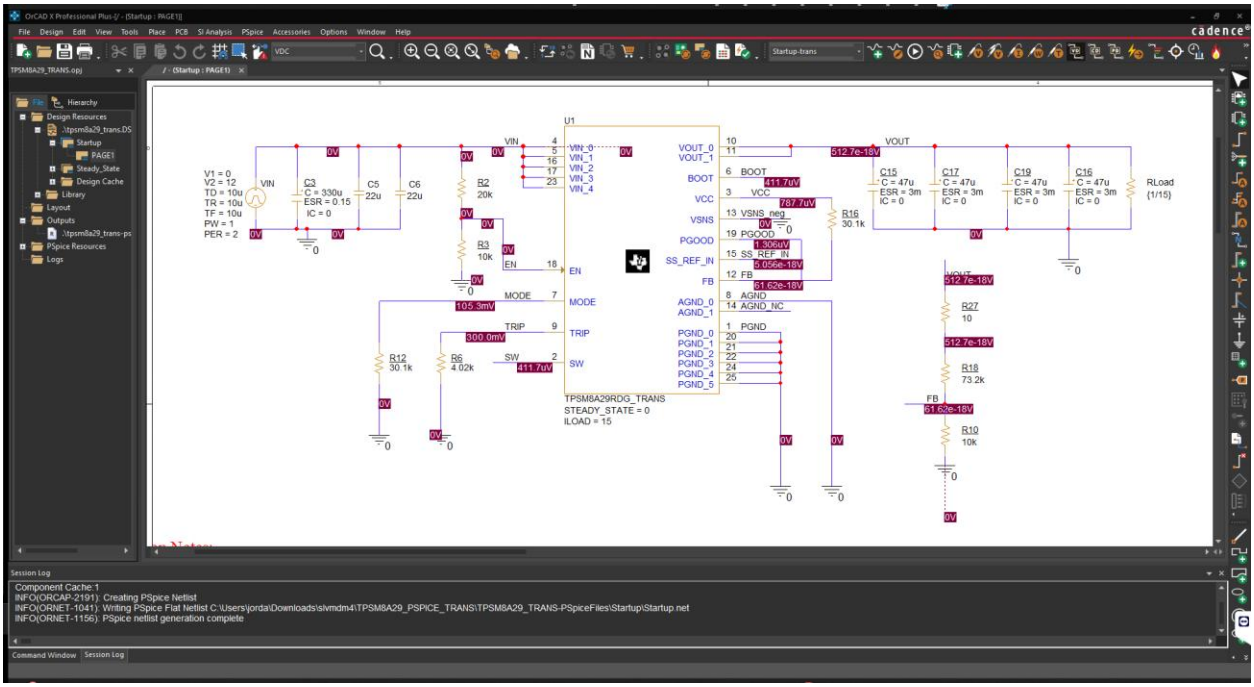
REV 1.1 PCB FC Hardware Design

A.2 Rust Firmware Snippets

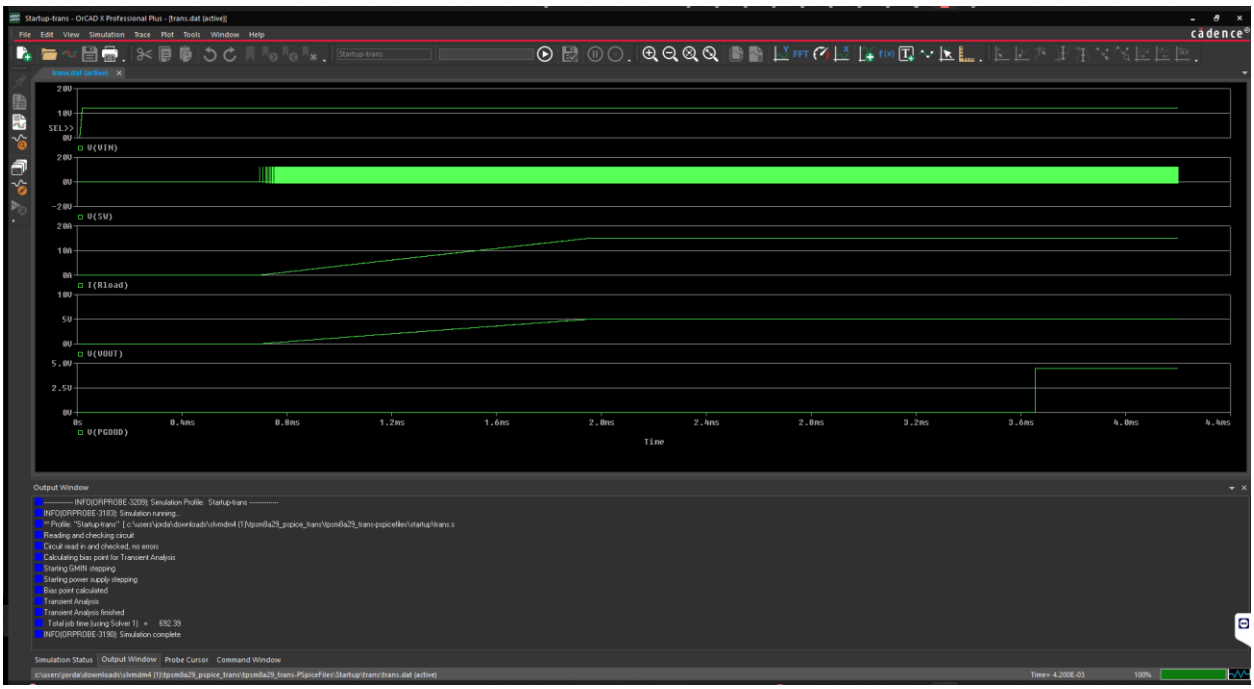
A.2.1 I2C Probe Implementation

```rust
pub fn i2c_scan(i2c: &mut I2C) {
    for addr in 0..127 {
        if i2c.write(addr, &[]).is_ok() {
            defmt::info!("Found device at address: {:#x}", addr);
        }
    }
}
```

A.3 Cadence Simulation Plots

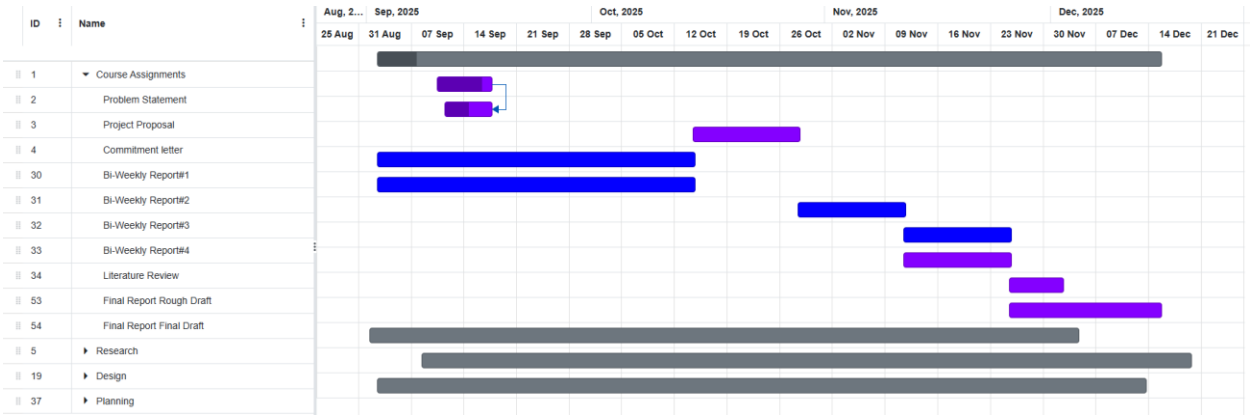


Schematic Simulation for 5v 15A power module

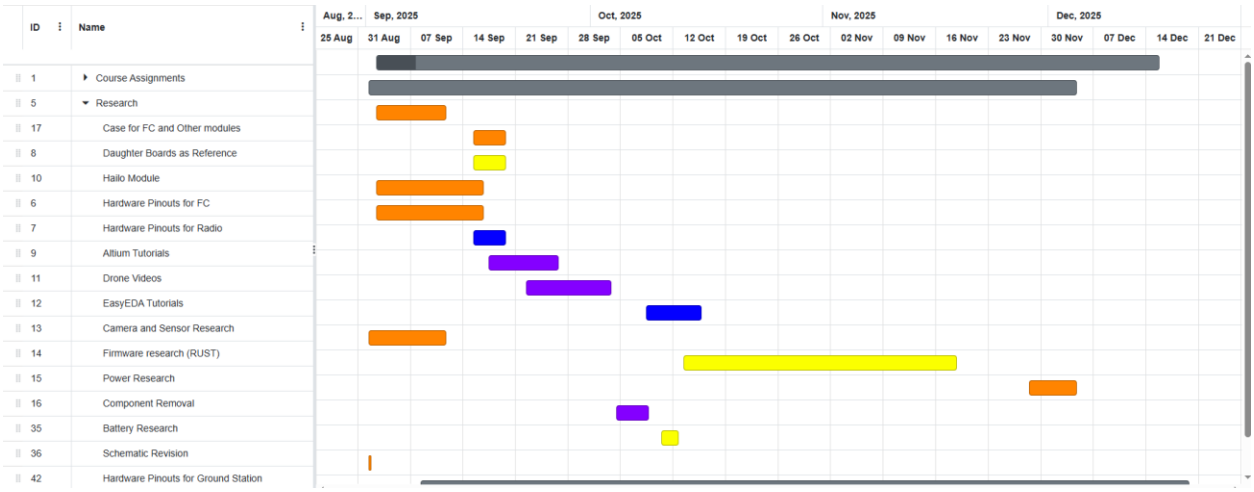


5V 15A Power Module Simulation Results

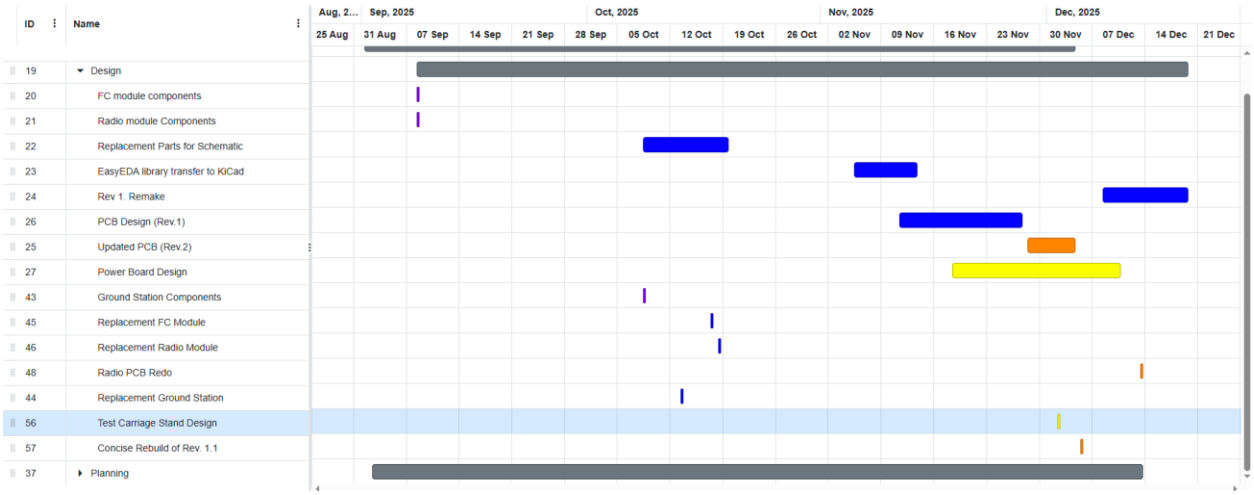
A.4 Final Gantt Chart



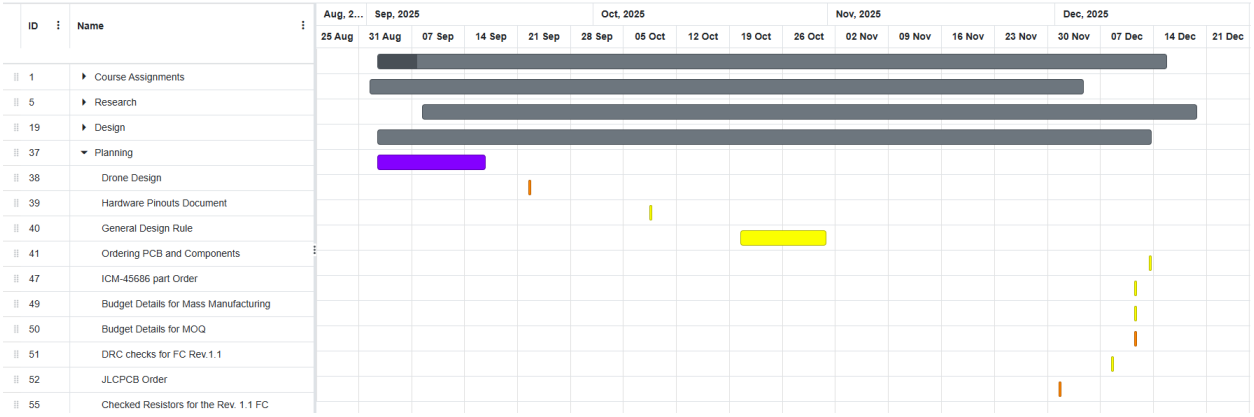
Final Gantt Chart (Course Assignments Tab)



Final Gantt Chart (Research Assignments Tab)



Final Gantt Chart (Design Assignments Tab)



Final Gantt Chart (Planning Assignments Tab)