COVERSHEET

THE UNIVERSITY OF WESTERN AUSTRALIA

Faculty of Engineering and Mathematical Sciences

Assignment, Report & Laboratory Coversheet for Individual & Group Assignment

SUBMITTING STUDENT		
SURNAME VO	GIVEN NAMES BA-NANG	STUDENT NUMBER 23376047
UNIT NAME ELECTRICAL AND ELECTRONIC ENGINEERING DESIGN PROJECT		UNIT CODE ELEC5552
TITLE/TOPIC OF ASSIGNMENT FINAL REPORT		NAME OF LECTURER/FACILITATORS JEGA GURUSAMY

FOR GROUP ASSIGNMENTS ONLY NAME	STUDENT NUMBER
1. Ba-Nang Vo	23376047
2. Josh Wong	23693193
3. Nehan Cripps	23417516
4. Daniel Papasergio	2306658
5. Josh Gorn	23350938
6. Johane Swanepoel	23168823

Unless other arrangements have been made it will be assumed that all group members have contributed equally to group assignments/laboratory reports

SIGN:	SIGN:
SIGN: Nehan Cripps	SIGN:
SIGN:	SIGN:
SIGN:	SIGN:

DECLARATION

We are aware of the University's policy of academic conduct (see over), and we each declare that this submission is our own group's work entirely; suitable acknowledgement has been made of any sources of information used in its preparation; we are able to provide evidence if needed of the individual contribution made by each member of the group; we have retained a hard copy for our own records.

NOTE: No assignment will be accepted without the declaration above being signed and dated

SEE OVER FOR INFORMATION ON REFERENCING & PLAGIARISM

UWA ELEC5552

Indoor Surveillance Drone
Team 2-02

Author(s)
Josh Wong (23693193)
Nehan Cripps (23417516)
Johane Swanepoel (23168823)
Ba-Nang Vo (23376047)
Daniel Papasergio (23066558)
Josh Gorn (23350938)

Final Report
Volume 1

Project Partners: UWA Aviation Lab

UWA Supervisor: Jega Gurusamy

Team Number: 2-02
Version 2.0

Group Meeting Day and Time: Wednesday 12 PM

Word Count: ***20,520 Words***

Word Limit: ***21,600 Words***

Revision History

Date	Version	Description	Author
16/07/25	1.0	- Creation of document based on template	Ba-Nang Vo
29/09/25	1.1	- Document Updated based on Final Report Template document	Nehan Cripps
02/10/2025	1.2	- EMS Cover Sheet Added - Started System Integration	Nehan Cripps Johane Swanepoel
07/10/2025	1.3	- Further separated sections - Stakeholder Engagement, Safety and Ethical Issues and List of Design Outputs	Josh Wong Ba-Nang Vo
08/10/2025	1.4	- Edited risk analysis section to better suit rubric and set up automatic reference table - Moved old risk analysis section to safety section	Daniel Papasergio Ba-Nang Vo
08/10/2025	1.5	- Word count limit updated - Section 2.3 updated	Nehan Cripps
13/10/2025	1.6	- Separated sectioning to be component and tech specific for 2.3 – 2.8.	Josh Wong
15/10/2025	1.7	- Added Master Version of report to prevent change conflicts	Josh Wong
17/10/2025	1.8	- Added PCB and Power - Added Navigation Sensors - Added Navigation Software	Josh Gorn Daniel Papasergio Ba-Nang Vo
20/10/2025	2.0 (CURRENT)	- Finished entire report	All

Executive Summary

This project delivers a compact, indoor drone platform designed for safe operation in the Lycopodium Lab. A lightweight 3D-printed frame and custom four-layer PCB centre an ESP32-WROOM flight controller, four brushed motors, and a focused sensor suite containing: VL53L1X time-of-flight sensors for altitude and obstacle ranging, and a PMW3901 optical-flow unit for planar odometry.

Control is structured around a straightforward hover loop: the system reads orientation and height, compares them to a “level at target height” goal, and applies small PID corrections blended across four motors. Above this, a deterministic navigation planner executes simple motion primitives (e.g., forward, rotate) while enforcing a 0.5 m stand-off from obstacles. If the commanded path encroaches on that buffer, the drone arrests, sidesteps or gently turns into free space, then resumes.

Electrically, a 2S battery feeds regulated 5 V/3.3 V rails, with motors on the battery path, I²C (with XSHUT) assigns unique addresses to the ToF sensors, and SPI links the optical-flow module. A lightweight Wi-Fi web interface provides arming/disarm, waypoint entry, live telemetry, and a kill switch. A link watchdog (~200 ms) and basic guards (battery, sensor validity) can pre-empt to hover, land, or disarm.

To de-risk integration, the firmware supports Flight (motors enabled) and Demo (motors inhibited). In Demo mode, the full perception-planning stack runs in real time while publishing virtual commands, allowing indoor walkthroughs that validate detection, planner logic, and safety behaviour before flight. Remaining work focuses on UI polish, logging, and incremental flight tuning.

Table of Contents

1. Introduction

This report presents the design, development and testing of a quadcopter indoor surveillance drone for the UWA Aviation Laboratory. This project delivers a lightweight aerial quadcopter vehicle capable of stable flight, avoidance and real-time video streaming within enclosed environments. This document encapsulates the complete engineering process and life cycle, from the original requirements through to design, fabrication, software integration, testing, verification and recommendations for future improvements. This document also records important lessons learned and the professional practices followed throughout the project life cycle.

1.1 Purpose

The purpose of this final report is to formally document how the project team met the objectives set out in the Requirements Analysis Report and subsequent design deliverables. This final report explains the rationale behind key design choices and changes made during the hardware fabrication and software development, highlighting how these decisions address not only safety, performance, compliance requirements but also project feasibility. This report also provides evidence that the final system satisfies user and regulatory needs, including Civil Aviation Safety Authority (CASA) guidelines, UWA Laboratory safety standards and general engineering ethics. It serves as a record of lessons learned and a reference for future development or maintenance for engineering projects.

1.2 Scope

The scope of this report encompasses the entire engineering life cycle undertaken during the 12-week project. It describes activities and results across all phases, including:

- **Requirements Analysis and Prioritization** – Consolidation of stakeholder needs, constraints and compliance standards.
- **System Architecture and Power Distribution Design** – Hardware layout, signal routing and supply regulation.
- **Component Selection** – Justification of motors, sensors, microcontroller, communications modules, camera and power electronics.
- **Hardware and Firmware Integration -** Pin mapping, motor driver circuits, sensor buses and shielding strategies.
- **Control Algorithm Design -** Altitude control, pre-flight check state machine and wireless shutdown logic.
- **Verification and Validation -** Bench testing, hover and flight trials, obstacle avoidance tests and compliance review.
- **Risk Assessment and Mitigation** – Integration challenges, electromagnetic interference and power stability.
- **Recommendations for Future** – Lessons learned and future opportunities.

1.3 Definitions, Acronyms, and Abbreviations

UWA – University of Western Australia
CASA – Civil Aviation Safety Authority
GPS – Global Positioning System
MCU – Microcontroller Unit
UWAAL – UWA Aviation Labs
ESC – Electronic Speed Controller: A device that regulates the speed of an electric motor, most found in radio-controlled vehicles and drones.
RoHS and CE – Restriction of Hazardous Substances and European Conformity: Set of laws in the EU that restrict the use of hazardous substances in electrical and electronic equipment for safety of users. Although, not directly applicable to Australian standards, compliance with RoHS is required for commercial exportation especially to countries in the European Union. Majority of electronics used in this project will already be RoHS and CE compliant.
IDE – Integrated Development Environment: software program to aid programming efficiently.
TQ – Technical Query
FOV – Field of View
UAV – Unidentified Aerial Vehicles
UI – User Interface
SPI – Serial Peripheral Interface
SDIO – Secure Digital Input/Output
RPA – Remotely Piloted Aircraft

1.4 Contributions of Team Members

Johane Swanepoel (Project Controller/Manager) – Executive Summary, System Architecture Overview, Obstacle Detection and Altitude, Final Integration of Software
Josh Wong (Client Liaison Manager) – Chassis Design, Motors and Hover Design, Assembly Build Process and Issues
Nehan Cripps (Quality Assurance Manager) – Server Control and Surveillance Design, Recommendations for Future Development
Josh Gorn (Treasurer/Secretary) – PCB and Power, Final Cost Table and Budget, Tending Considerations
Daniel Papasergio (Project Administrator) – Project Background, Obstacle Avoidance and Altitude, Navigation and Odometry, Risk Analysis, Safety & Ethical Issues, Operation Manual
Ba-Nang Vo (Configuration Manager) – Introduction, Final Requirements, Obstacle Avoidance and Altitude, Navigation and Odometry, Stakeholder Engagement, Safety & Ethical Issues, List of Design Outputs, Recommended Future Tests

1.5 Project Background

The project revolves around the main stakeholder UWA Aviation Labs (UWAAL) requiring an indoor drone capable of surveying a garage or room. The indoor aspect of these tasks means features commonly implemented for navigation and stabilisation such as GPS cannot be utilised due to the concrete environment which affects GPS [1]. The lack of GPS presents a challenge in navigating and stabilizing the drone, to retain these features a new avenue for navigation and stabilisation needs to be incorporated. The drone must also, in the worst-case scenario of using full power, be able to fly for at least 3 minutes in order to obtain sufficient data.

The project needs to consider safety with indoor operation; this is to be achieved by software and mechanical means. For software a “kill switch” will need to be implemented to ensure the drone shuts off or safely lands when communication with the drone to the controller is lost, another software safety consideration is including the ability of the drone to detect and avoid obstacles. Mechanical safety design requirements are also required to be implemented. Propellor shrouding/guards are needed as the propellor blades pose a cutting risk to people near when the drone is in operation. The drone must also be capable of following a pre-defined path around the room, detecting any obstacles from any directions, performing evasive manoeuvres to prevent any damage and have adjustable hovering heights.

This project is an integration of mechanical design, embedded control systems and autonomous navigation algorithms to produce a reliable indoor surveillance drone. The final design aims to incorporate a lightweight frame, onboard sensors for obstacle detection and altitude control and an MCU-based control system to process this sensor data and execute movement commands as directed by the navigation algorithm. The project seeks to demonstrate autonomous flight within an enclosed environment while maintaining positional stability, obstacle avoidance and user safety. Through the combination and integration of the aforementioned elements, the project team seeks to deliver a working prototype for an autonomous indoor surveillance drone capable of operating without the reliance on GPS for the UWA Aviation Lab.

1.6 Report Structure Overview

1. **Introduction** – Outlines context, purpose and scope of the project. This section also includes individual team member contributions, background to the project and definitions of key terms and acronyms to ensure clarity for all readers.
2. **Final Design and Design Process**
   2.1. Final Requirements - Finalised prioritised requirements summarised
   2.2. System Architecture Overview – Provides block-level view of the hardware and software systems and their interactions between major subsystems
   2.3. Chassis Design – Covers development of physical frame including material selection and structural justification
   2.4. Motors and Hovers System Design – Details thrust and lift mechanisms, motor configuration and structural justification
   2.5. Obstacle Detection and Altitude – Describes sensor choice, placements and signal interpretation for accurate positional and obstacle detection
   2.6. Navigation and Odometry – Describes movement logic and odometry
   2.7. PCB and Power Design – Documents electrical layout, component integration and power distribution network
   2.8. Communications System Design – Explains the wireless communication protocol and data-link interface between onboard and external systems.
   2.9. Stakeholder Engagement – Records interactions with supervisors, technical staff and end-users to ensure alignment with expectations
   2.10. Safety and Ethical Issues - Identifies safety hazards and ethical concerns, and describe mitigations or design decisions to address them
   2.11. Assembly/ Build Process and Issues - Documents on how the prototype was built, what problems occurred and how they were resolved.
   2.12. Final Integration of Software – Details the penultimate integration of the preliminary code
   2.13. Risk Analysis – Identifies the five biggest risks to safety, functionality and schedule along with their mitigation strategies.
   2.14. List of Design Outputs – A structured list of deliverables including hardware, software and documentation.
   2.15. Final Cost Table and Budget – Summarises expenses against the allocated budget and justifies financial decisions.
3. **Recommendations for Future Development**
   3.1. Next Steps for Prototype – Proposes short-term improvements for stability, endurance and useability.
   3.2. Required Approvals/Compliance - Identifies additional certifications, testing or academic approvals required for further development.
   3.3. Tending Considerations – Guidance on how prototype can be commercialised.
   3.4. Recommended Future Tests – Additional experiments/field trials suggested to further validate the project design.

2. Final Design and Design Process

2.1 Final Requirements

Below is a table summary of the group's original consensus top-ranked requirements.

Table 2-1: Original Top 20 Requirements

Req No.	Category	Requirement	Group Priority Score
REQ-002	2. Altitude Control/ Hover Stability	The drone shall stop and hover within 0.5 s when an obstacle is detected within 0.5 m and shall shut down propulsion if obstacle persists for more than 30 s	5
REQ-003	1. Indoor Navigation	The drone shall follow a pre-determined horizontal and vertical flight path	5
REQ-008	6. System Integration	The flight controller, power-distribution board, and ESC circuitry shall be implemented on custom-designed PCBs tailor to the drone layout	5
REQ-010	1. Indoor Navigation	The operator shall comply with Part 101.B of the CASR regulations during all drone use and testing	5
REQ-011	1. Indoor Navigation	The drone shall be able to fly for a minimum of 3 minutes	5
REQ-017	3. Obstacle Avoidance	The drone shall detect and avoid obstacles within +/- m in all directions including forward, backward, lateral and vertical axes	5
REQ-001	2. Altitude Control / Hover Stability	The drone shall maintain altitude within +/- 0.1 m of the target set-point when hovering	4.83
REQ-021	4. Manual Override & Safety	All onboard electronics shall comply with RoHS and CE standards	4.83
REQ-026	8. Drone Frame & Protection	The drone shall meet safety standards including battery protection, secure mounting and physical safeguards	4.83
REQ-032	7. Flight Stability & Control	The drone shall maintain lateral deviation of +/- 0.5 m of the target horizontal path	4.83
REQ-004	7. Flight Stability & Control	The drone shall store at least one pre-defined course which it can actively follow	4.67
REQ-013	8. Drone Frame & Protection	The drone design shall implement propellor guards for protection	4.67
REQ-014	6. System Integration	The drone shall use 1S batteries in a capacity of no more than 800 mAh	4.67
REQ-007	4. Manual Override & Safety	The operator shall be able to wirelessly shutdown the drone on command with latency ≤≤ 200 ms	4.5
REQ-009	1. Indoor Navigation	The drone shall not make any use of any GPS system	4.5
REQ-022	8. Drone Frame & Protection	The drone shall weight no more than 250g	4.5
REQ-012	6. System Integration	The drone batteries shall be chargeable using commercial USB-C ports	4.33
REQ-025	5. Communication	The drone shall be able to communicate using open-source non-specialized software easily accessible to the end user	4.33
REQ-030	5. Communication	The drone's course shall be able to be reprogrammed in less than 5 minutes	4.33
REQ-031	5. Communications	Flight logs including timestamps, battery %, commands, and sensor events shall be exported in csv file format post-flight	4.33

Table 2-2 below shows the Verification and Validation testing of the initial set of requirements including any updates or learnings from testing the requirements, including whether the team was able to or failed to meet the previous set of requirements.

Table 2-2: Verification and Validation of Original Set of Requirements

Req No.	Category	Old Requirement	Updated Requirement (only if updated)	Acceptance Criteria	Pass/Fail	Test	Learnings
REQ-002	2. Altitude Control/ Hover Stability	The drone shall stop and hover within 0.5 s when an obstacle is detected within 0.5 m and shall shut down propulsion if obstacle persists for more than 30 s	Regarding obstacle detection under the altitude control/stability section, this was altered to instead of turning off propulsion after object persists for 30s to attempt to find a safe way around the object	Drone halts <=0.5 s after obstacle insertion; propellers stop <=10 s if obstacle remains.	F	Requirement was changed regardless of testing	Shutting down propulsion entirely proved to be unsafe due to the drone dropping far too quickly
REQ-003	1. Indoor Navigation	The drone shall follow a pre-determined horizontal and vertical flight path		Drone halts <=0.5 s after obstacle insertion; propellers stop <=10 s if obstacle remains.	P	Drone has pre-programmed path in code	
REQ-008	6. System Integration	The flight controller, power-distribution board, and ESC circuitry shall be implemented on custom-designed PCBs tailor to the drone layout		Design files and assembled boards reviewed and approved against requirements.	P	Designed files and assembled boards reviewed and approved against requirements	
REQ-010	1. Indoor Navigation	The operator shall comply with Part 101.B of the CASR regulations during all drone use and testing			P	All testing was done in controlled environments and out of harm's way	
REQ-011	1. Indoor Navigation	The drone shall be able to fly for a minimum of 3 minutes			P	Drone battery was tested at full power with all components to theoretically last 3 minutes	

Req No.	Category	Old Requirement	Updated Requirement (only if updated)	Acceptance Criteria	Pass/Fail	Test	Learnings
REQ-017	3. Obstacle Avoidance	The drone shall detect and avoid obstacles within +/- m in all directions including forward, backward, lateral and vertical axes			P	Drone was able to detect obstacles within 0.5 m in all directions	
REQ-001	2. Altitude Control / Hover Stability	The drone shall maintain altitude within +/- 0.1 m of the target set-point when hovering		Stable altitude held for >=30 s in three consecutive trials.	F	Drone did NOT maintain altitude of 0.1m of target set point when hovering	
REQ-021	4. Manual Override & Safety	All onboard electronics shall comply with RoHS and CE standards			P	All the electronic components important are already RoHS and CE standards	
REQ-026	8. Drone Frame & Protection	The drone shall meet safety standards including battery protection, secure mounting and physical safeguards			P	Drone was designed with ALL safety standards implemented	
REQ-032	7. Flight Stability & Control	The drone shall maintain lateral deviation of +/- 0.5 m of the target horizontal path			F	Drone did NOT maintain lateral deviation of 0.5m of the target horizontal path	
REQ-004	7. Flight Stability & Control	The drone shall store at least one pre-defined course which it can actively follow		Patterns selected and executed as commanded during demo.	P	Patterns are pre-programmed in code	
REQ-013	8. Drone Frame & Protection	The drone design shall implement propellor guards for protection			P	Drone design implemented propellor guards for protection	

Req No.	Category	Old Requirement	Updated Requirement (only if updated)	Acceptance Criteria	Pass/Fail	Test	Learnings
REQ-014	6. System Integration	The drone shall use 1S batteries in a capacity of no more than 800 mAh	The drone 2S batteries because less voltage sag and no need to boost, just buck	The batteries the drone operates off in the final design are 1S 800MaH batteries	F	Drone ended up using 2S Batteries instead	2S Batteries give a more steady voltage and current
REQ-007	4. Manual Override & Safety	The operator shall be able to wirelessly shutdown the drone on command with latency ≤≤ 200 ms		Measured round-trip latency <=0.2 s for 90 % of 20 commands.	P	The code was tested for a latency of 200 ms and shutdown according to web server	
REQ-009	1. Indoor Navigation	The drone shall not make any use of any GPS system		The drone has no GPS integrated	P	The drone had no GPS components integrated, instead utilising sensors for navigation (e.g. ToF)	
REQ-022	8. Drone Frame & Protection	The drone shall weight no more than 250g		The drone is weighed and is less than 250g total	P	Weighed final design and the weight total	Case = 35g
REQ-012	6. System Integration	The drone batteries shall be chargeable using commercial USB-C ports	Drone ended up being charged with Micro-USB instead due to difficulties with USB-C connections		F	In a TQ asked it was noted the batteries are to be charged externally not internally for safety reasons, thus we could not implement this.	Micro-USB proved easier to use
REQ-025	5. Communication	The drone shall be able to communicate using open-source non-specialized software easily accessible to the end user		The drone is controllable via an easy to use webserver UI.	P	All code was accessible simple code on GitHub and drone controlled with easy-to-use web server UI	
REQ-030	5. Communication	The drone’s course shall be able to be reprogrammed in less than 5 minutes			F	Reprogramming drone was physically timed	

Req No.	Category	Old Requirement	Updated Requirement (only if updated)	Acceptance Criteria	Pass/Fail	Test	Learnings
REQ-031	5. Communication	Flight logs including timestamps, battery %, commands, and sensor events shall be exported in csv file format post-flight			F	Drone only outputs to serial monitor instead	

2.2 System Architecture Overview

2.2.1 Purpose and Scope

This section presents the end-to-end system architecture of the indoor drone. It explains how the mechanical frame, propulsion, electronics, sensing, control software, and communications integrate into a coherent system. How signals and power move through the platform; and how timing and safety mechanisms ensure predictable behaviour in indoor spaces. The aim is to give the reader a single mental model of the whole machine before later sections dive into design details. Figure 2-1 shows the drone operating in the Lycopodium Lab, showing floor surface, nearby obstacles, and key interactions with the indoor environment.

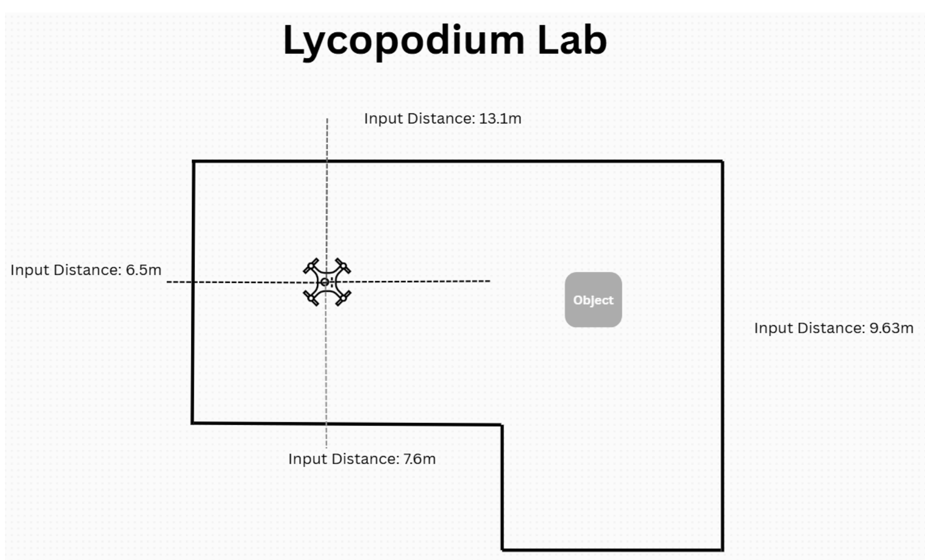


Figure 2.2.1-1: Operational context showing the drone, indoor environment (Lycopodium Lab), floor surface, and possible obstacles.

2.2.2 Architectural Concept

The drone follows a "light but sufficient" principle. A hybrid 3D-printed chassis provides just enough stiffness to push structural modes above the control bandwidth while staying inside the weight budget. A custom 4-layer PCB concentrates the ESP32-WROOM compute, power conversion, motor drivers and sensor interfaces at the mass centre. Propulsion is four brushed DC motors driven by 18 kHz, 8-bit PWM for quiet, smooth output.

A focused sensor suite delivers the three essentials for indoor autonomy: (i) absolute height via a downward VL53L1X ToF, (ii) sector ranging for obstacle avoidance via horizontal VL53L1X sensors around the perimeter, and (iii) planar velocity/odometry via a PMW3901 optical-flow sensor. On top of this hardware, a compact control stack on the ESP32 runs the attitude/hover loop at 250 Hz and a supervisory navigation planner that respects a configurable stand-off distance, handles avoidance, and executes queued motion primitives. A lightweight web UI over Wi-Fi provides arming, mode control, telemetry, and planner visualisation; the control link is deliberately separated from the video stream.

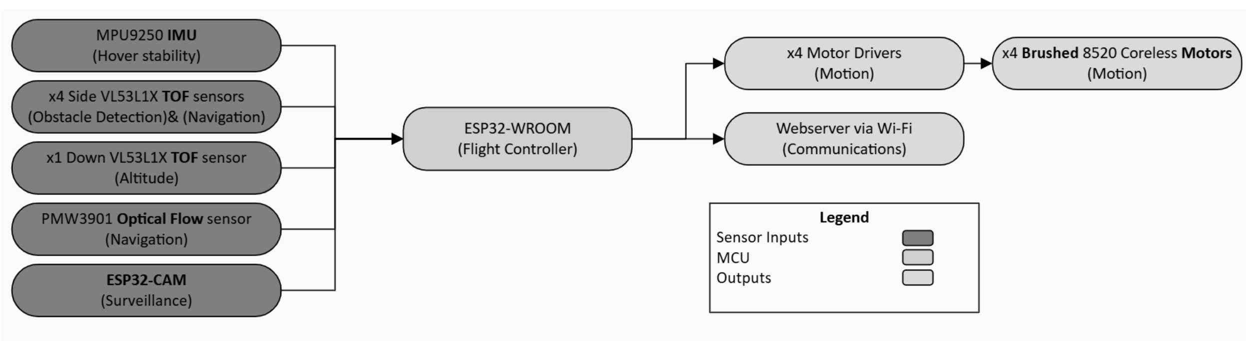


Figure 2.2.2-1: High-level connections between the ESP32 controller, sensors, motor drivers, and the Wi-Fi UI/video links.

2.2.3 Physical Integration

The chassis is a thin base plate with locally thickened arms that increase second moment of area at the motor tips, raising the first bending mode above the control loop frequency and reducing IMU contamination. Motors are retained in friction-fit collars to save mass and simplify service. The PCB mounts centrally and low to keep the centre of mass near the propeller plane as can be seen in Figure 2-3. The downward VL53L1X and PMW3901 sit close together on the centreline so their measurement footprints co-locate, reducing geometry error when the vehicle tilts. Horizontal VL53L1X modules face forward/left/right/rear with a slight outward cant to reduce self-reflections and widen near-field coverage.

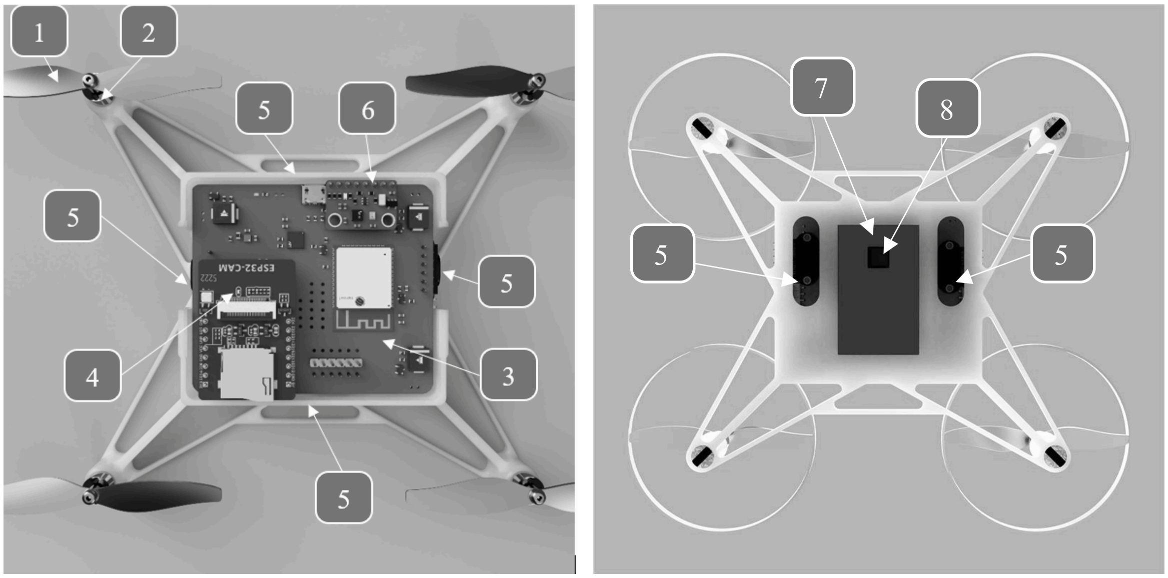


Figure 2.2.3-1: Top and bottom view of the design showing locations of components.

Table 2-3: Descriptions of labels on Figure 2-3

Label	Component Name
1	Propellor Blades
2	Motors
3	PCB
4	ESP32-Cam
5	VL53L1X ToF sensors
6	IMU
7	2S Battery
8	Optical Flow sensor

2.2.4 Electrical & Bus Architecture

Power enters from a 2S Li-ion/LiPo feed as can be seen in Figure 2-4. Two buck stages generate regulated 5 V and 3.3 V rails for logic and peripherals, the motor rail draws directly from the battery path through drivers sized for peak current. High-current motor returns are routed away from sensitive grounds; a continuous ground plane between signal layers shields long I²C/SPI/ADC runs.

All VL53L1X modules share the I²C bus and use per-sensor XSHUT lines so the ESP32 can hold them in reset at boot, bring them up one at a time, assign unique addresses (0x30–0x34), and start continuous ranging without collisions. The PMW3901 interfaces over SPI with a dedicated chip-select. Battery voltage is monitored on an ESP32 ADC channel via a resistor divider (scaled in firmware) for brown-out awareness and logging. Motor drivers accept 18 kHz PWM from the controller so thrust updates are smooth and outside audible bands.

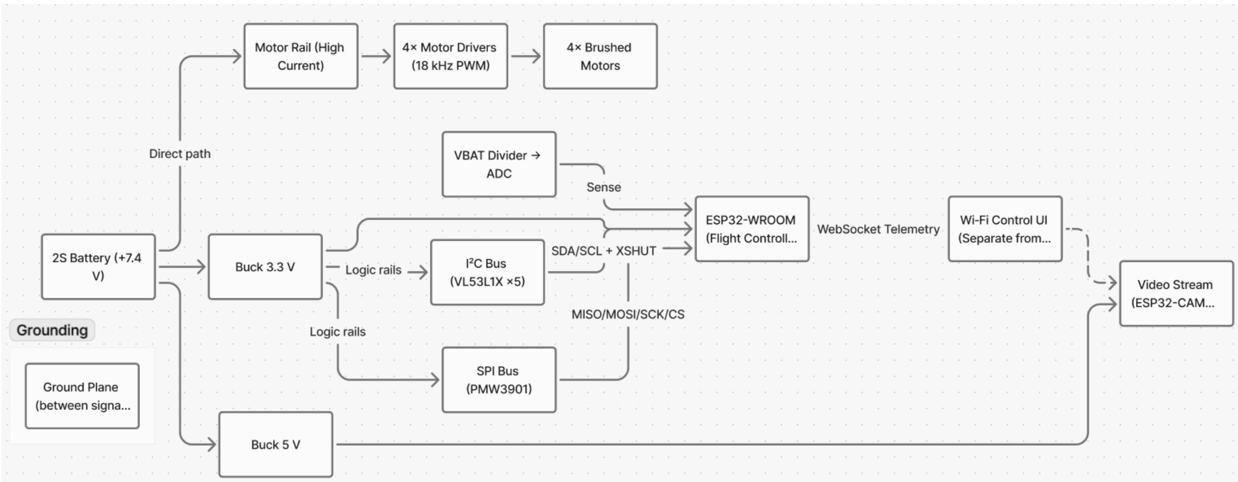


Figure 2.2.4-1: Battery, regulators, and motor rail routing with shared I²C (XSHUT addressing), SPI for optical flow, and ADC battery monitor.

2.2.5 Hover Control & Timing

The controller continuously senses orientation and height, compares them to a target "level at hover height," and turns any difference into small PID corrections. These are blended with a baseline lift and distributed across the four motors, so total thrust remains steady while attitude and height are quietly corrected. An arming ramp brings power up smoothly and basic guards (battery, sensor validity, link) are checked before closed-loop control takes over. During demos we run the same loop with motors inhibited ("demo mode") to visualise decisions and stability without spinning props. A network watchdog in the control link disarms if heartbeats stop or latency exceeds ~200 ms.

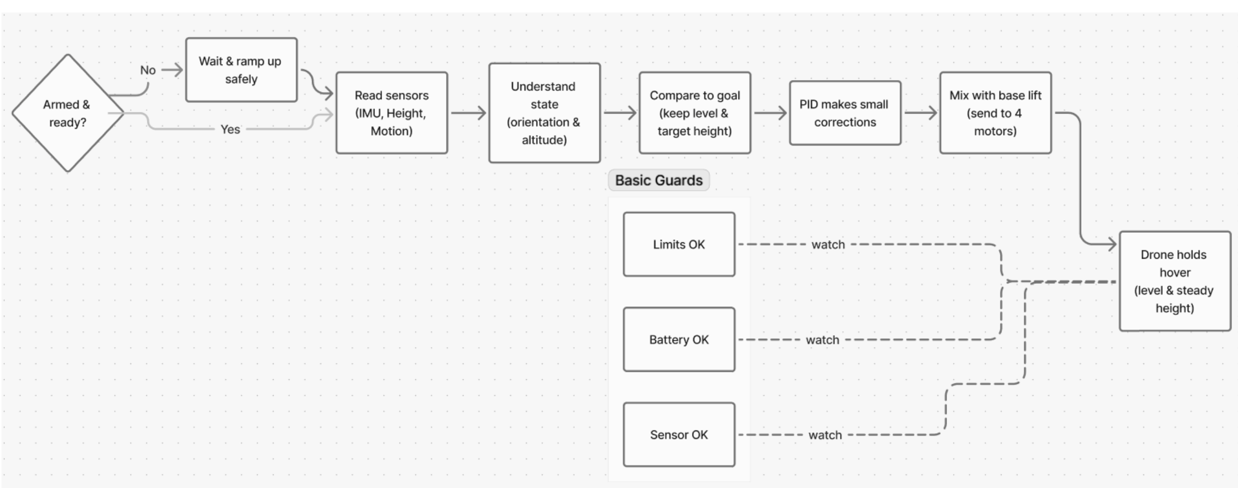


Figure 2.2.5-1: Conceptual loop: read orientation/height, compare to target, apply small PID corrections, and blend into four motor commands.

2.2.6 Obstacle Detection and Behaviour

Four horizontal VL53L1X sensors provide sector-based ranging. Each sensor runs in continuous mode. The planner enforces a 0.5 m stand-off: if the commanded direction reports < 0.5 m to an obstacle, motion is arrested and the vehicle holds; if there is space laterally, a short sidestep/turn away from the obstacle is commanded before resuming. Hysteresis and light debounce avoid chatter. The downward VL53L1X is handled separately as absolute altitude and does not contribute to obstacle flags.

In flight, avoidance suggestions become actual motor commands (lateral velocity and modest yaw). For demonstrations in confined spaces, an identical logic path can be run with motors inhibited. Suggested commands are still computed and published to the UI for transparent behaviour without propeller motion.

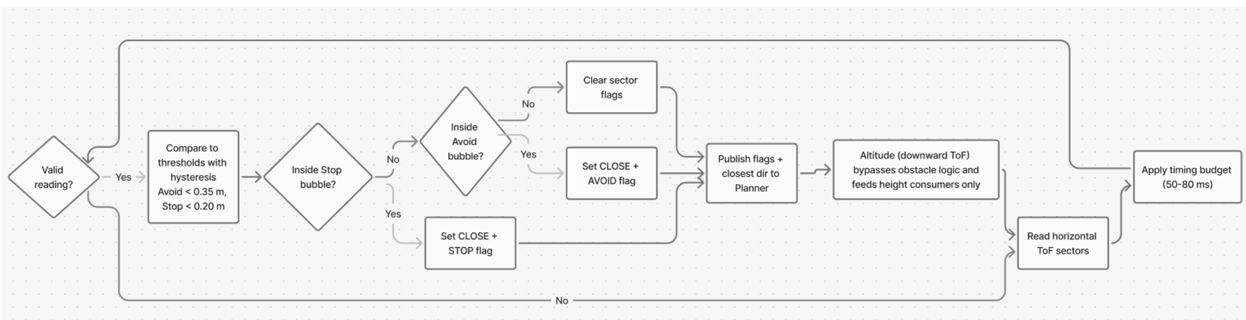


Figure 2.2.6-1: Sector ToF readings mapped to avoid/stop bubbles with hysteresis, publishing the closest-obstacle cue to the planner.

2.2.7 Odometry & Navigation Logic

Planar odometry uses the PMW3901 optical-flow sensor and the current altitude from the downward ToF. Pixel-flow counts are converted to angular rates using a calibrated FoV scale, multiplied by height to obtain ground-relative velocities, then lightly filtered and integrated to produce (x, y) displacement. A small deadband suppresses drift when stationary; height is clamped within sensible bounds to reject poor surfaces.

Navigation is structured as a deterministic stepper executed by a supervisory planner/state machine:

- **States:** IDLE to NAVIGATE to AVOID to HOVER_EMERGENCY (faults pre-empt to hover/land).
- **Commands (primitives):** FORWARD d, RIGHT d, ROTATE θ, HOLD t.
  Each executes with continuous guard checks (obstacles, link, altitude validity, VBAT).
- **Avoidance:** if a threat enters the avoid bubble along the commanded direction, the stepper arrests motion, performs a short sidestep into the clearer sector (or a gentle yaw), then resumes the original plan; entering the stop bubble forces hover and re-planning.

During flight, the planner's outputs drive the mixer as body-frame Vx/Vy and yaw-rate commands. During demo mode, the same outputs are published as virtual commands on the web UI while motors remain disabled, allowing the team to walk the prototype around the lab to show that perception, planning, and decision-making work as intended.

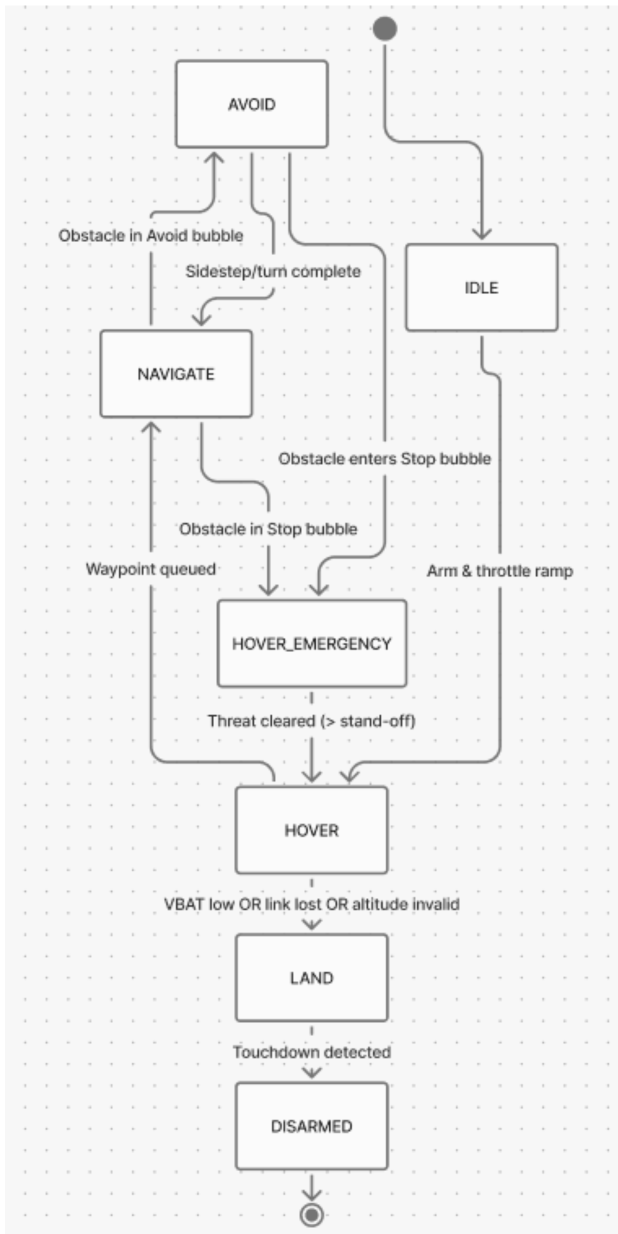


Figure 2.2.7-1: State machine executing queued motion primitives with guarded sidestep/hover responses to obstacles and faults.

2.2.8 Communications, Telemetry and Safety

The ESP32 hosts a lightweight Wi-Fi interface (AP fallback enabled) that serves the web UI and a WebSocket for live control/telemetry. Endpoints expose arming/disarming, mode control (Hover

/ Navigate), gain/trim edits, waypoint queue management, and a status string with motor outputs, VBAT, pose, and planner state. The UI also visualises closest obstacle, planner state, and suggested/actual commands (Vx, Vy, Yaw) so operators can validate logic in real time. The control link is separate from any video stream, so flight control never competes for bandwidth.

Safety is enforced in several places: (i) a client-presence guard disarms automatically if the last control client disconnects while armed; (ii) VBAT thresholds trigger warnings and controlled landing; (iii) altitude and sensor validity are monitored so stale/invalid readings near the ground do not propagate; (iv) all motor outputs pass through an arming ramp and persistent slew limiter; and (v) the planner's stop bubble always pre-empts to HOVER_EMERGENCY before any commanded translation.

The control webserver exposes three features used in both flight and demo modes: live sensor data, a path/command input panel, and a kill switch. It also emits periodic heartbeats that the controller's ~200 ms watchdog uses to disarm on link loss.

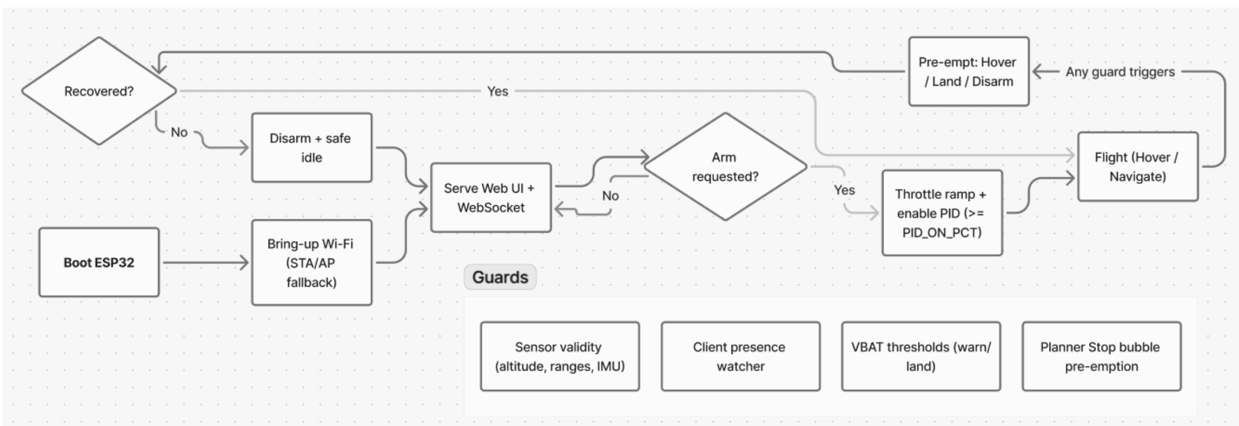


Figure 2.2.8-1: Web UI/WebSocket control path with client-presence, VBAT, and sensor-validity guards that can pre-empt to hover/land/disarm.

2.3 Chassis Design

The chassis philosophy evolved as test results clarified the most important trade-offs. Outlined below is the closed feedback loop style of iterating the engineering process for the chassis design. Predominantly this is separated into the 3 sections of Design Philosophy and Justification, Design Elements and lastly, Testing and Results. Each test from each ***Concept*** inspired the next ***Concept***.

2.3.1 Design Philosophy and Justification

Pre-Concept

The first design was inspired from basic research into what the shape of a drone looks like from inspiration like DJI Mavic 2 and DJI Tello [2]. Below are drawings that capture the inspiration that drove the very first steps in the design process.

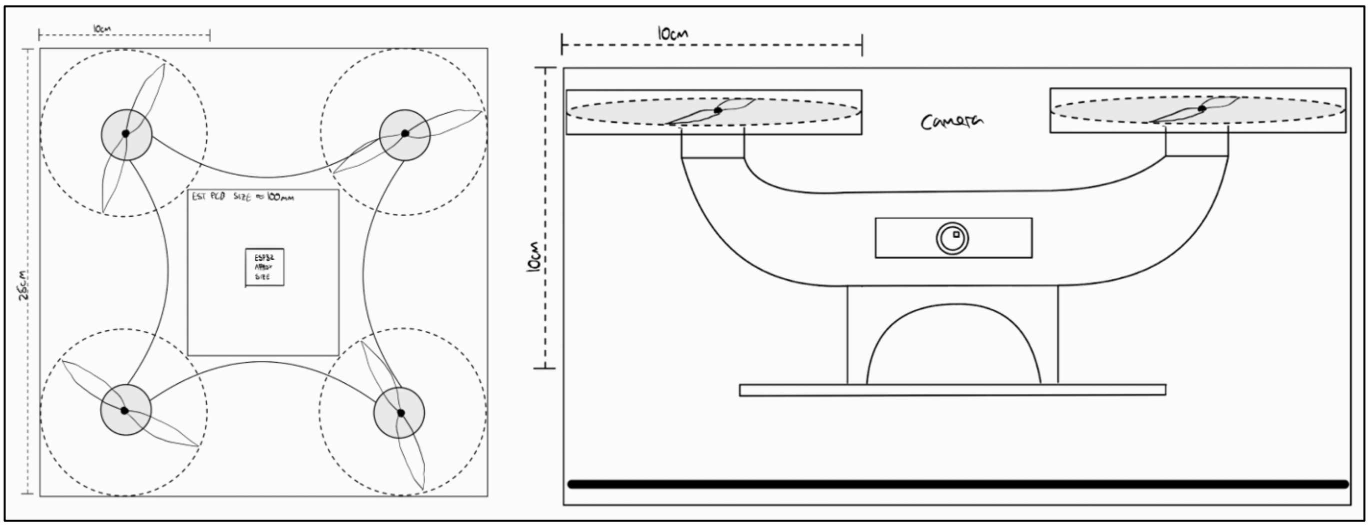


Figure 2.3.1-1 Concept art for a quadcopter design inspired from various DJI products

Although basic, this inspired the initial justification behind fundamental characteristics of the drone. Specifically, the symmetrical distribution of motors ideally leading to a simpler design with weight being placed centrally in the drone to reduce the complexity of PID tuning in the future.

Concept 1

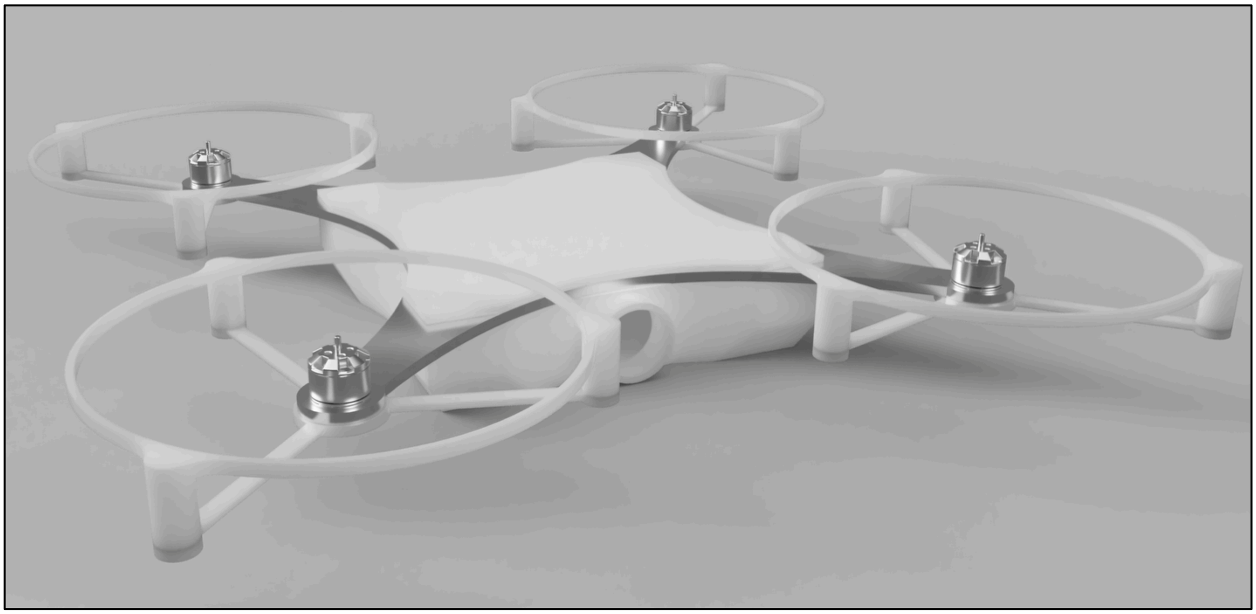


Figure 2.3.1-2 – Fusion360 render of Concept 1, using Aluminium X-Frame and ABS shell

Concept 1 was the practical realisation of the prior concept art. It was designed using brushless motors in mind, this theoretically allowed for a significant weight budget. This build had a estimated weight of about 200g, with immense structural rigidity. However, as stated in section 2.4, brushless motors were found to be infeasible due to cost and complexity of a custom electronic speed controller. Hence, following the change to brushed motors, this weight needed to be significantly reduced.

Concept 2

Concept 2 was the first attempt at building a lightweight frame for the drone, the carbon version of it weighed only 60 grams which was a large change in comparison to the previous design. The biggest fundamental change was swapping out one large bulky aluminium plate weighing 120g for 2 small carbon x frames totally only 12 grams. This design had 10% the weight while holding 70% of the tortional rigidity for a force applied in the z direction at one motor point when compared to the previous design (as simulated in fusion 360).

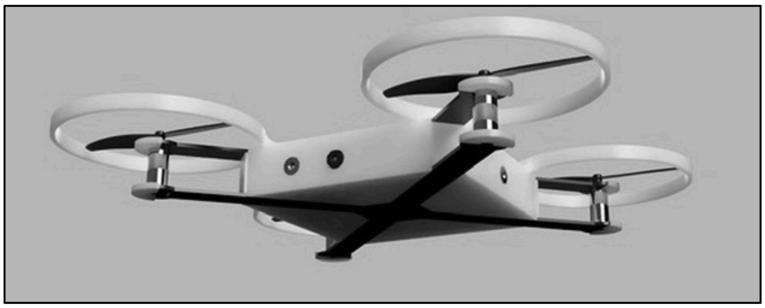


Figure 2.3.1-3 - Render of concept 2

This design proved a large conceptual struggle in determining the optimal mounting for brushed motors. The motors incorporated in the design, 8520 coreless brushed motors, fundamentally have no mounting points. The solution to this was using puck like motor plates, which would slip onto the edge of the drone's arms. This ideally would complete the tortional box using the motors as a fundamental pillar of the design and are visible in the image above as the white plates on the top and bottom of the carbon plates next to the motors (see figure above).

This design was fundamentally built around the components which were being prototyped at the time. As you can see in the left figure, the optical flow sensor was mounted due to the preliminary assumption that it would be reading the roof and similarly the team's preliminary obstacle detectors used ultrasonic sensors for obstacle detection (see figure below).

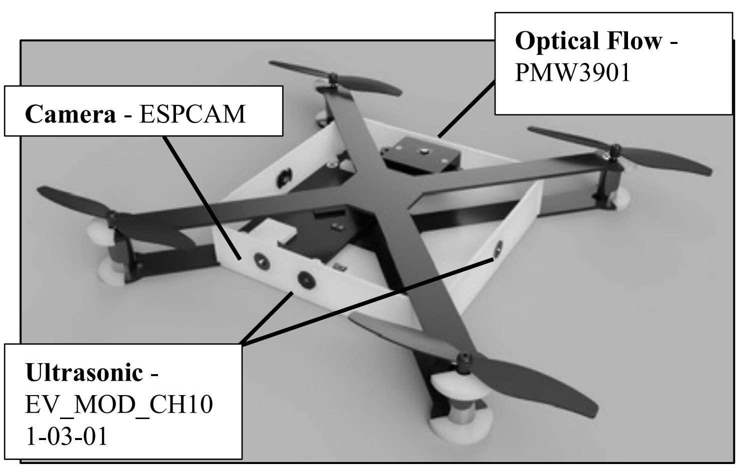


Figure 2.3.1-4 Fusion 360 render of Concept 2, using 2 carbon fibre X-Frames and ABS body

Upon assessment of feasibility for the manufacturing of this carbon X-Frame design it was found that with the resources available, it would not be possible to manufacture this concept, due to safety concerns over milling carbon fibre. It was decided to push the prototyping process forward; this would instead be cut out aluminium.

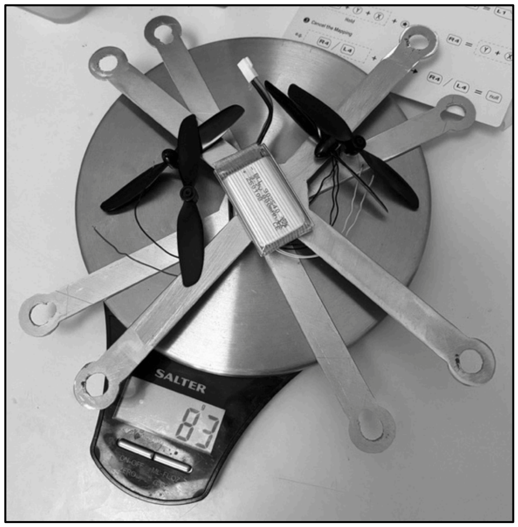


Figure 2.3.1-5 - Weighing of initial prototype frame

However, upon manufacturing of the aluminium alone it was found that the difference in densities of the materials and similarly the flaws in the design meant that this drone would be significantly over the weight budget, with an estimated final drone weight being over 140g. This led to another redesign with weight more in mind.

Concept 3

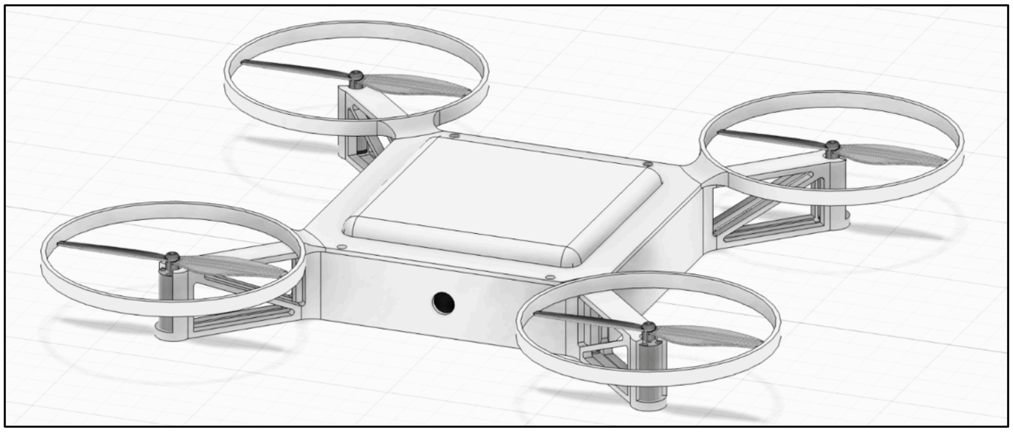


Figure 2.3.1-6 - Concept 3 fusion file

The above image shows the new drone with an 80g chassis. It also was redesigned to accommodate the protoboard motor driver which is discussed in 2.4. This was the first model of drone concepts which was fully realised. It featured 3D printed motor mounts which made a better solution to the prior motor mounts.

The first issue encountered in testing was to do with the manufacturing stage. To get the motors into the motor mounts, the design had to be significantly warped, this led to delamination and the need to remelt the ABS to get the motors back into the design. An image of the printed frame is seen below.

Figure 2.3.1-7 - Printed drone chassis with motors

The next problem was evident, when the motor driver system was integrated the drone skated along the ground, and could not get into the air. The first assumption was weight issues, which inspired the next concept.

Concept 4

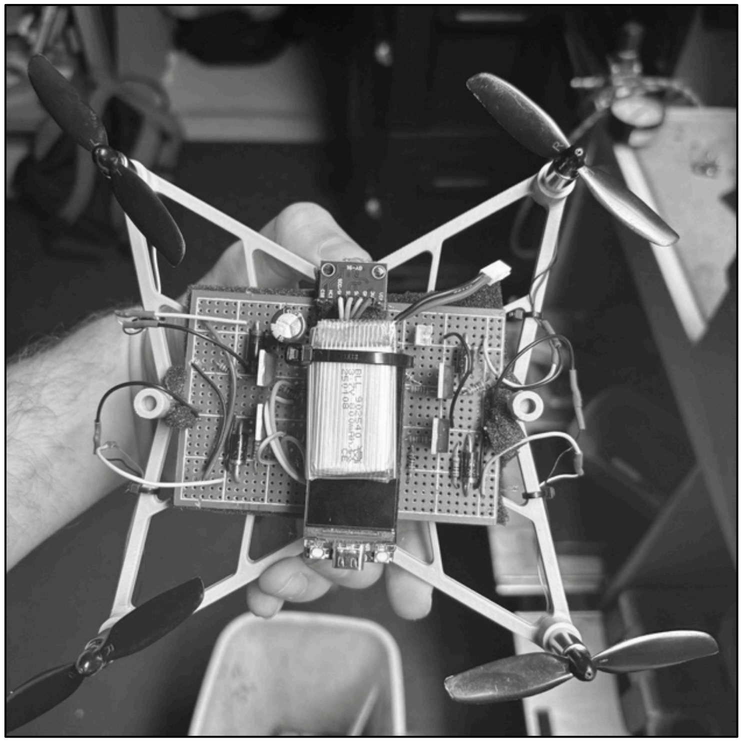


Figure 2.3.1-8 - Image of prototype drone with all flying components

This concept reflected the stage the team was at in the design process, which was to get this drone flying at all costs. It took the past failed idea of snap fitting the past drone to a new level with friction fit mounts for the motors. With the motors being measured to have exactly an 8.5mm diameter, the CAD file was designed to have exactly an 8.5mm diameter. When a 3D printer prints, it squishes the layers, so the hole is slightly less than 8.5mm by approximately 0.1mm. This leads to an extremely tight fit which did allow for the motors to separate from the chassis in any of durability tests.

All prop guards were removed, and a barebones design was created, this led to the drone being able comfortably lift off the ground at 60% PWM. However, it was not stable, flipping and constantly yawing. There were initial concerns with the design's centre of mass being off, the image above shows the solution to this was placing the ESP32 module and the 1S battery in the centre with a cable tie. Yet the platform still had further hover issues which will be mentioned in section 2.4.

Ultimately, the main issue with the chassis now was that the motor was uncontrollable due to the frame being extremely weak in the z-direction (perpendicular to the PCB plane). This meant the gyroscope's roll could not get any accurate reading and made hover impossible. This led to a new redesign

Concept 5

Taking in the learnings from the past 2 drones, the resulting hybrid design balances these lessons: a thin base with targeted arm thickening, friction/snap fits to remove metal hardware, and just enough stiffness to push structural modes out of the control bandwidth. This "light but sufficient" approach met the weight target while maintaining stable sensor behaviour and leaves room for continued refinement of local stiffening where testing indicates it is most effective. The figure below shows the enhancements in the design.

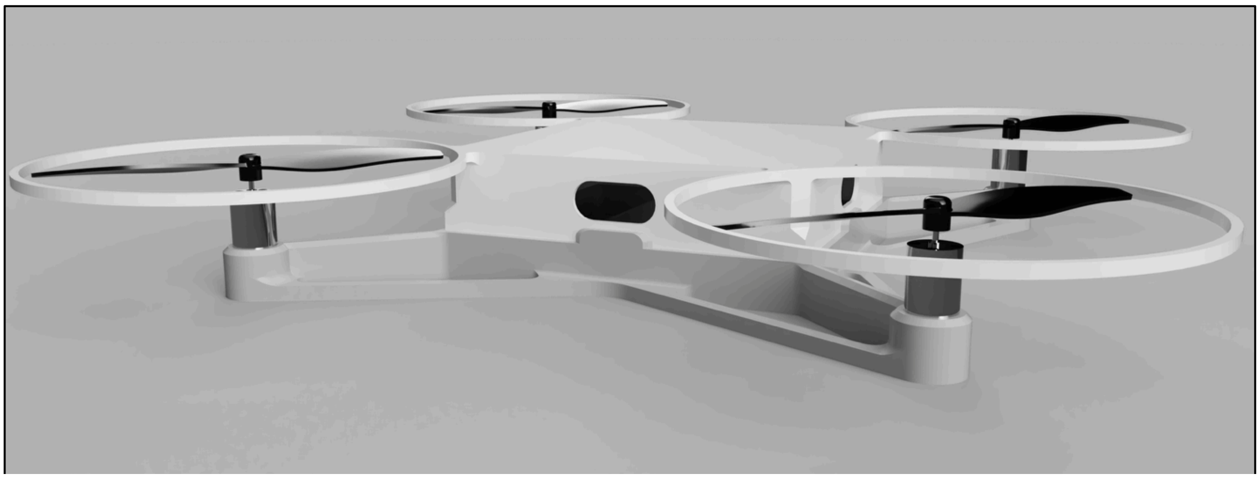


Figure 2.3.1-9 - Render of concept 5

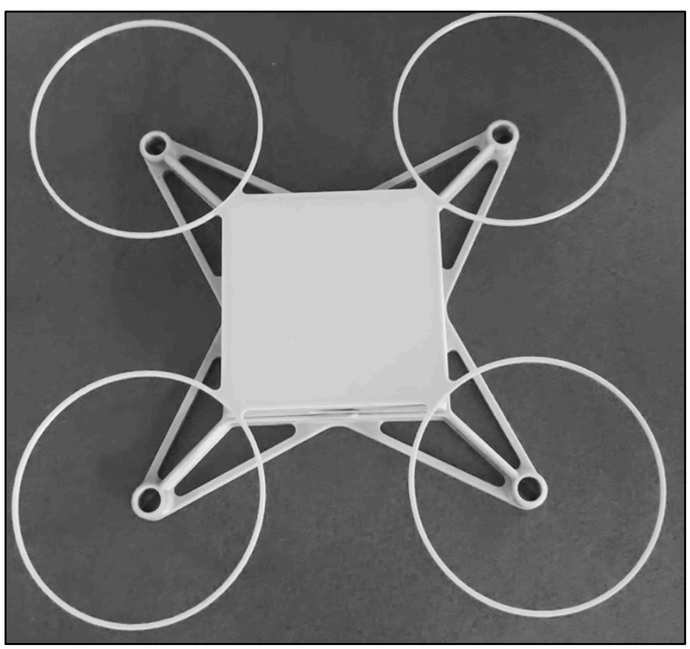


Figure 2.3.1-10 - Print of concept 5

Final Design

The final drone is the culmination of the best components from each drone and heavily inspired from the previous concept. It was mainly designed to accommodate the PCB and new parts that the group chose as other members refined their designs. The drone's height had to be significantly increased to accommodate all parts, this led to concerns with weight; to optimise this the design is a tight fit around the PCB's footprint. Leading to a design which was more rigid than the previous one and only 34g (see figure below).

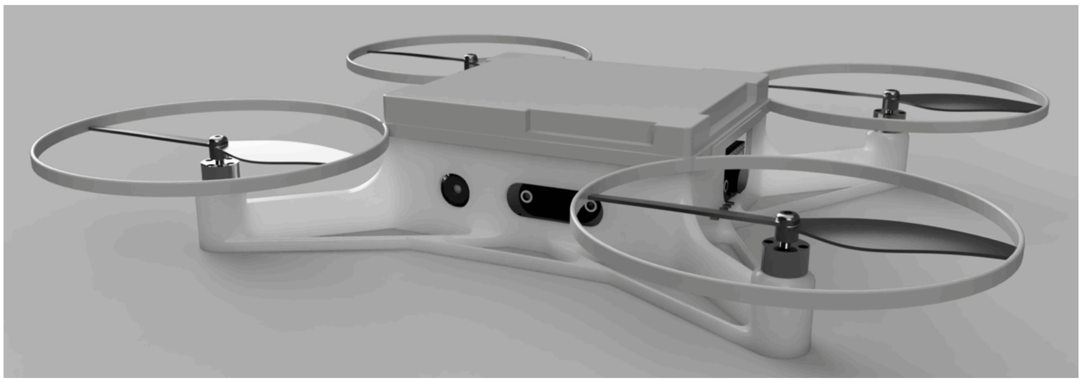


Figure 2.3.1-11 - Render of Final Design (front view)

The importance behind this design was to make it more rigid in the Z-direction as their where still significant issues with the vibration spanning through the frame. This was found during a test where the motor was held in different spots to see if mounting position mattered. What was found is that the closer to the propeller the motor is mounted, the less vibrations there were. Hence the extrusion at the motor mounts is significantly taller than the prior design.

The propellor guards were also made slightly taller due to the prior design not providing sufficient protection, this can be seen in the figure below.

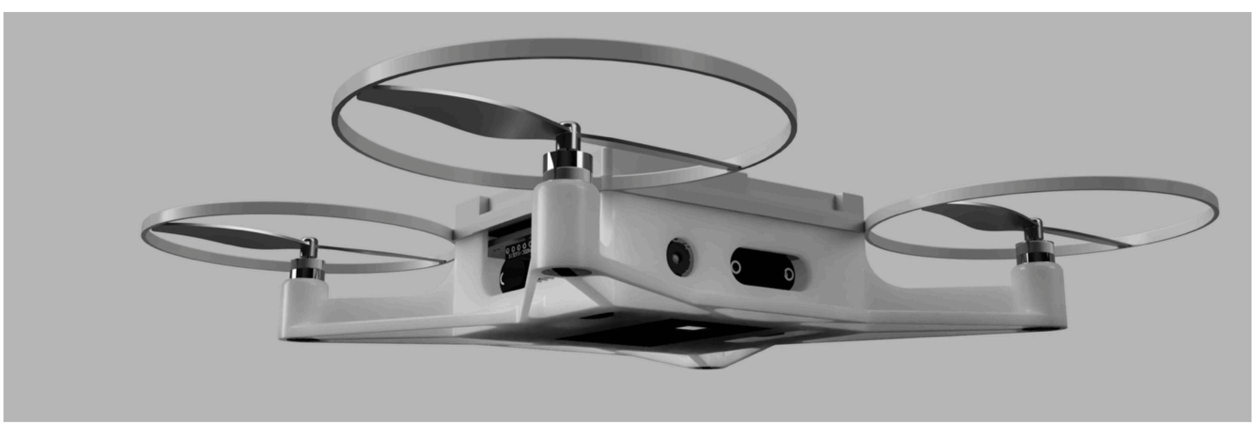


Figure 2.3.1-12 - Render of Final Design (low front view)

The snap fit design of the prior concept was reused, except with the snap fit location being higher on the box, this allows for sensors to be mounted into the walls of the design without having to worry how to open the box once sensors are mounted (a previous issue with concept 5).

There are large cut outs to allow for easy mounting and cooling in the design. The top view below shows how to PCB and sensors are incorporated into the shell.

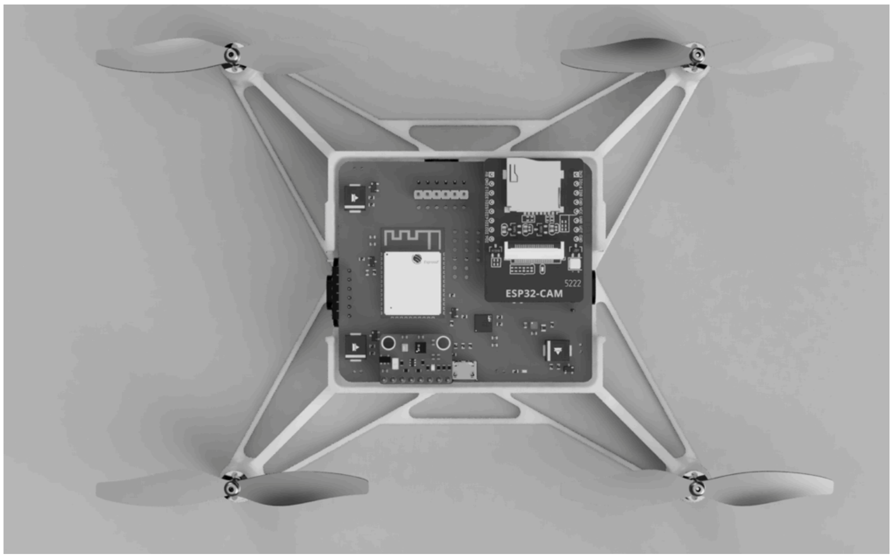


Figure 2.3.1-13 - Render of Final Design (top view)

2.3.2 Design Elements

Drone body

Main frame made from ABS, box component is 1mm ABS. Drone centre arm is 3mm thick and its height is 10mm. This is to minimise vibrations in the z direction (see figure below).

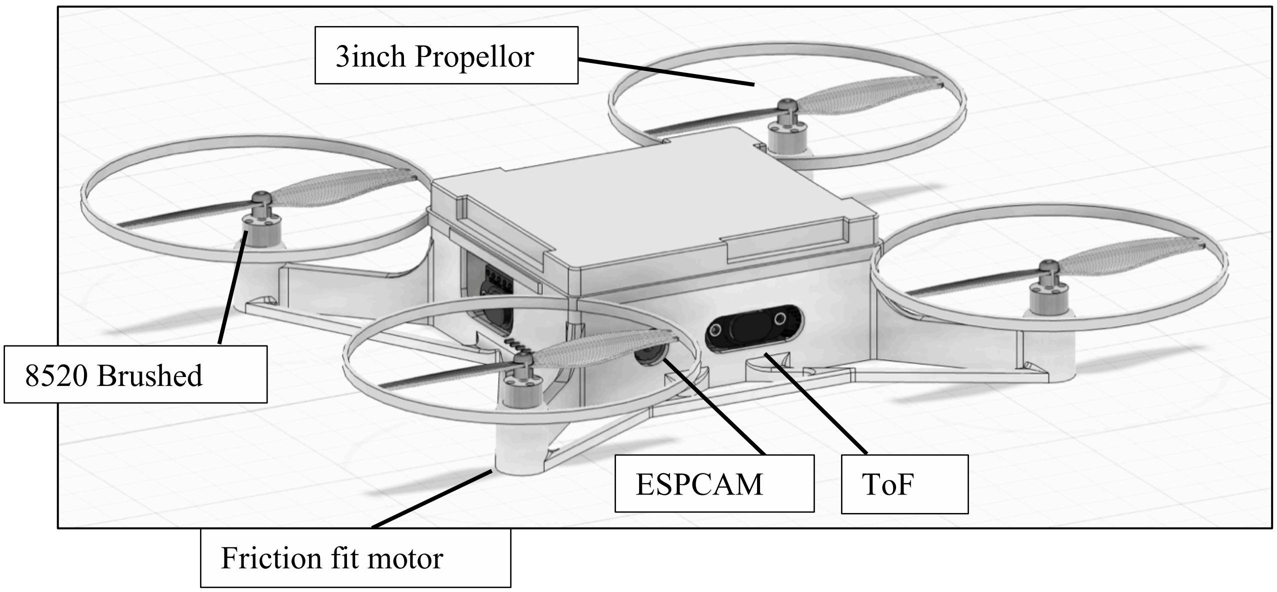


Figure 2.3.2-1 - Labelled Image of final design

Snap fit case

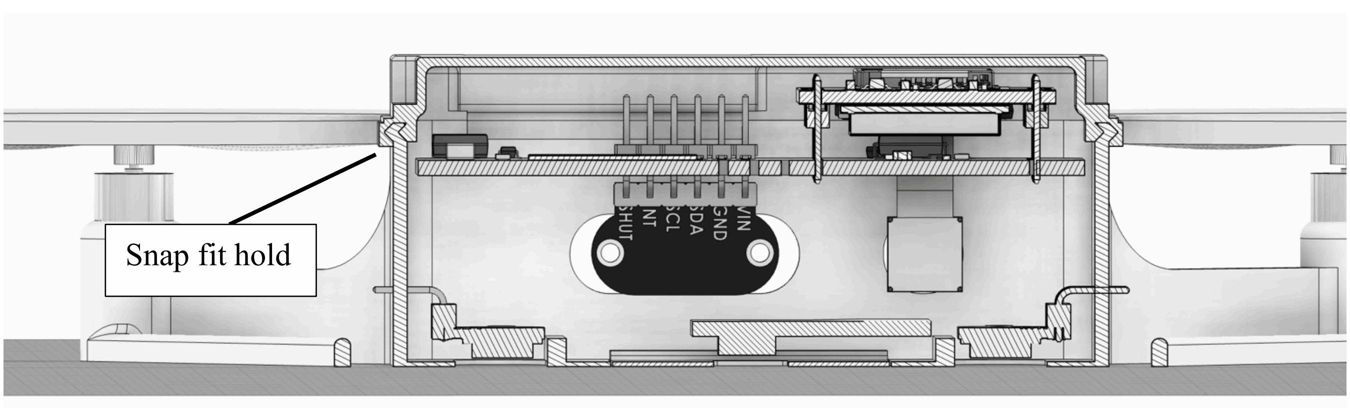


Figure 2.3.2-2 - Snap fit case labelled diagram

The drone uses a snap fit design, shown above, this allows the case to be screwless. This was done to minimise weight as the actual lid is not taking any structural loads, so it needs very minimal strength. However, previous designs were not strong enough. This design has a tight fit using a 45-degree extrusion on both sides to ideally hold the lid on tightly.

Drone Lid

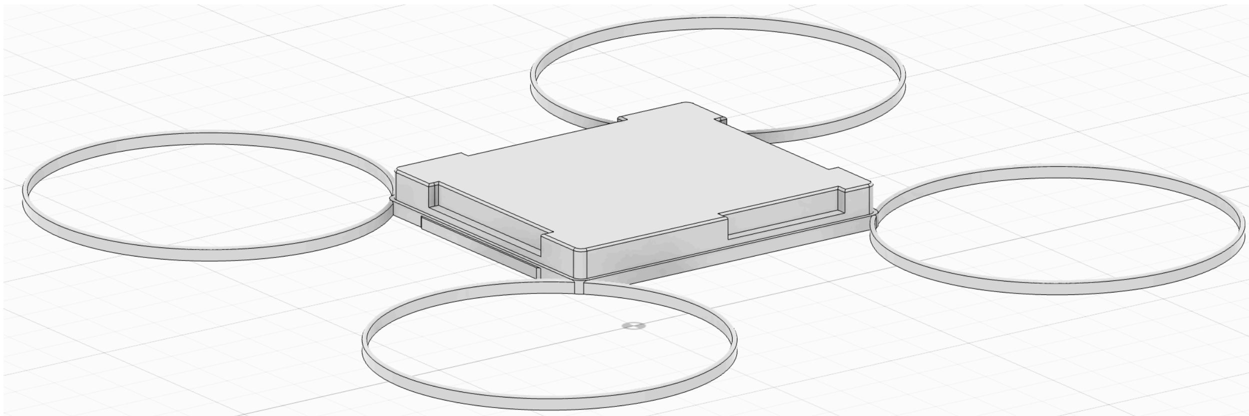


Figure 2.3.2-3 - Final design drone lid file

The drone lid, above, is designed to be an easily removable system to simply enclose and protect the propellors (and people from the propellors). It is extremely lightweight at 12g.

Battery Lid

Similarly, the battery lid is designed to safely enclose the battery in its compartment whilst also allowing for quick and easy removal for charging, see below.

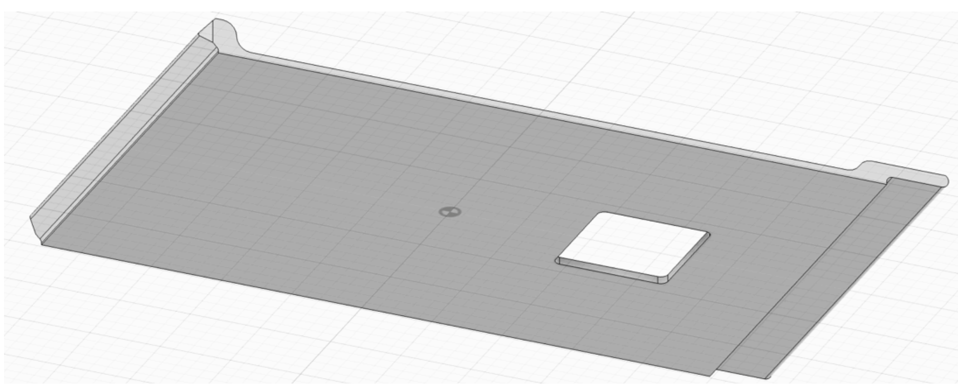


Figure 2.3.2-4 - final design battery lid

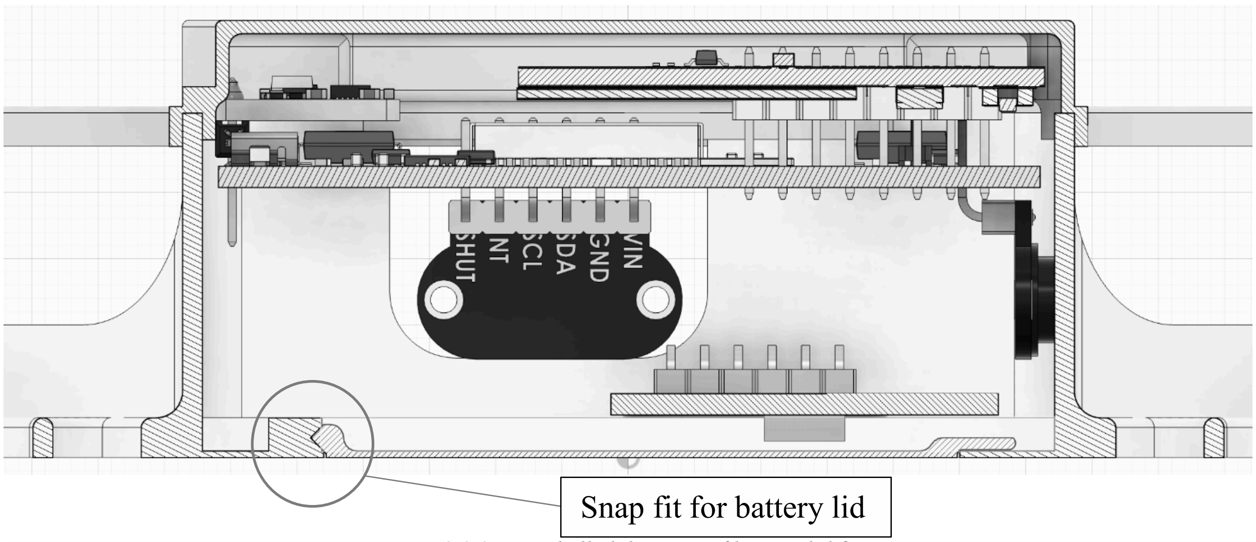


Figure 2.3.2-5 - Labelled diagram of battery lid fitting

The above image shows the battery lid the snap fit mechanism for the battery lid from side on. The reason for including a battery lid at the bottom is purely for accessibility, and it was deemed an appropriate spot for it.

2.3.3 Testing and Results

Concept 3 Testing

During the testing of Concept 3, the drone underwent its first full assembled flight trials. While the design allowed the system to lift off, the overall frame was found to be too heavy for stable sustained flight. This additional mass placed great strain on the motors and reduced the battery life significantly, to a point where the batteries struggled even at 50% PWM. Additionally, the propellor guards offered limited structural rigidity and failed to adequately protect the propellors from contact during lateral movements and crashes, proving to be insufficient in its protection. Overall, these two findings highlighted the need for a lighter and stronger frame material with more protective elements for the frame.

Concept 4 Testing

Concept four introduced a redesigned frame with lighter materials and thinner structural members to reduce weight. However, testing proved the frame experienced excessive vibration during motor operation. The material was also far too flexible, allowing high-frequency oscillations to transfer throughout the structure causing massive instability issues and noisy sensor readings. This iteration of the chassis proved that although weight reduction improved lifting performance, mechanical rigidity and vibration damping were critical for flight control and sensor reliability.

Concept 5 Testing

The fifth prototype chassis was 3D printed with reinforced geometry, resulting in a significantly more durable and rigid frame. This new design was a balance between weight and stiffness eliminating the flexibility seen in previous versions. However, due to time constraints, no elaborate flight testing was conducted for this iteration. Static vibration testing indicated an improvement in structural integrity, suggesting this design was a step in the right direction for the final model.

Final Testing

The final design combined learnings from all previous prototype designs, using the reinforced printed frame with additional damping elements at motor mounts and arm junctions. These improvements greatly reduced motor vibrations, resulting in smoother sensor readings and hover stability. The final design was also lighter and more compact, allowing for more successful flight testing with improved control and stability. Overall, the final design achieved a robust balance between strength, vibration resistance, weight supporting the drone’s safe and reliable operation.

2.4 Motors and Hover Systems Design

2.4.1 Design Philosophy and Justification

Motor and hover choices were driven by cost, lead time and integration simplicity. Given the PCB and schedule constraints, the team selected a motor solution that could be driven directly from the MCU with minimal external electronics and paired it with a conservative hover controller focused on safe take-off, stable level hold and smooth throttle authority.

Initially the most obvious option was to go for a brushless motor design over brushed with the BE-1104 being an ideal candidate. This was due to the brushless motor having a significantly more thrust per motor for similar power. The BE-1104 having ~82g of thrust per motor [3] vs the best brushed motor a 8520 coreless motor only having max ~42g of thrust per motor. [4]

However, an immediate hurdle was placed for the group, a technical query (TQ002) which revealed a premanufactured ESC was not allowed in this project. With the complexity of designing a custom ESC deemed out of scope for the project, it left brushed motors the only option. Severely limiting Chassis design due to the size of components needed.

This left the team with manually creating a motor driver circuit, to create PWM, the team used a simple MOSFET design with a Schottky flyback diode to prevent over current. The team then used a basic PWM signal form the ESP to actuate the MOSFET at the desired frequency.

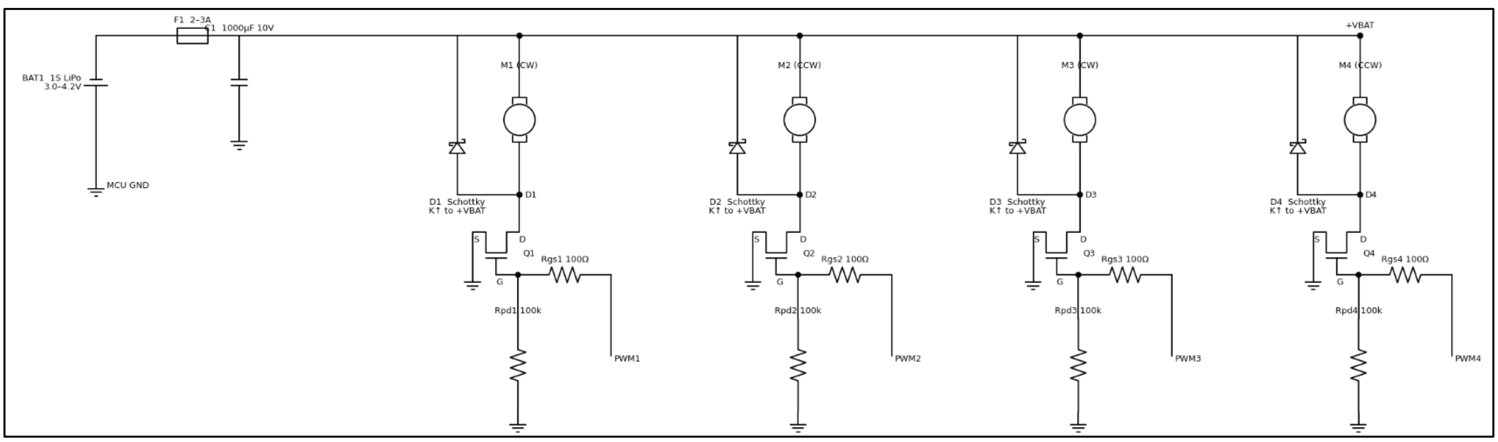


Figure 2.4.1-1 – Motor Driver Circuit

For the hover, the team used the MPU9250, this is a 9 DOF gyroscope which can confidently identify x,y,z and roll, pitch, yaw. With a basic PID loop conceptually this can get the drone to hover. Later in the design the team also found the necessity to trim the motor PWM manually to bring the centre of mass of the drone tighter so the PID could level the drone better.

2.4.2 Design Elements

The flight-control code implements PWM motor drive at 18 kHz with 8-bit resolution, per-motor slew-rate limiting (≈300%/s) for smoothness, and a mixer that holds total thrust constant while applying roll/pitch corrections. Arming/disarming gates outputs cleanly, and a short throttle ramp after arming avoids impulsive loads. Basic stabilise mode uses a complementary filter on an MPU-class IMU and PI-like terms on roll/pitch with integrator clamping. A simple web UI over Wi-Fi provides arm/disarm, gain/trim edits, status lines (VBAT, motor %, R/P/Y) and settings persistence in flash. Safety behaviour includes emergency disarm if the last client disconnects and watchdog-style state reporting. Battery voltage is sampled and filtered for brown-out awareness. These elements form the hover baseline while the altitude ToF and navigation stack manage vertical set-point and lateral motion limiting.

The final design used a 2S battery stepped down to the 4.2V that the 8520 brushed motors can used, this meant that the 1S batteries sag would not affect the drone and it could more consistently operate a heavier load.

2.4.3 Testing and Results

The very first motor driver protoboard was built as shown, in the figure below. This was used with the concept 3 flat plate drone to test hover but, its weight combined with that frame's vibration meant this design was undesirable and more improvement would have to be made.

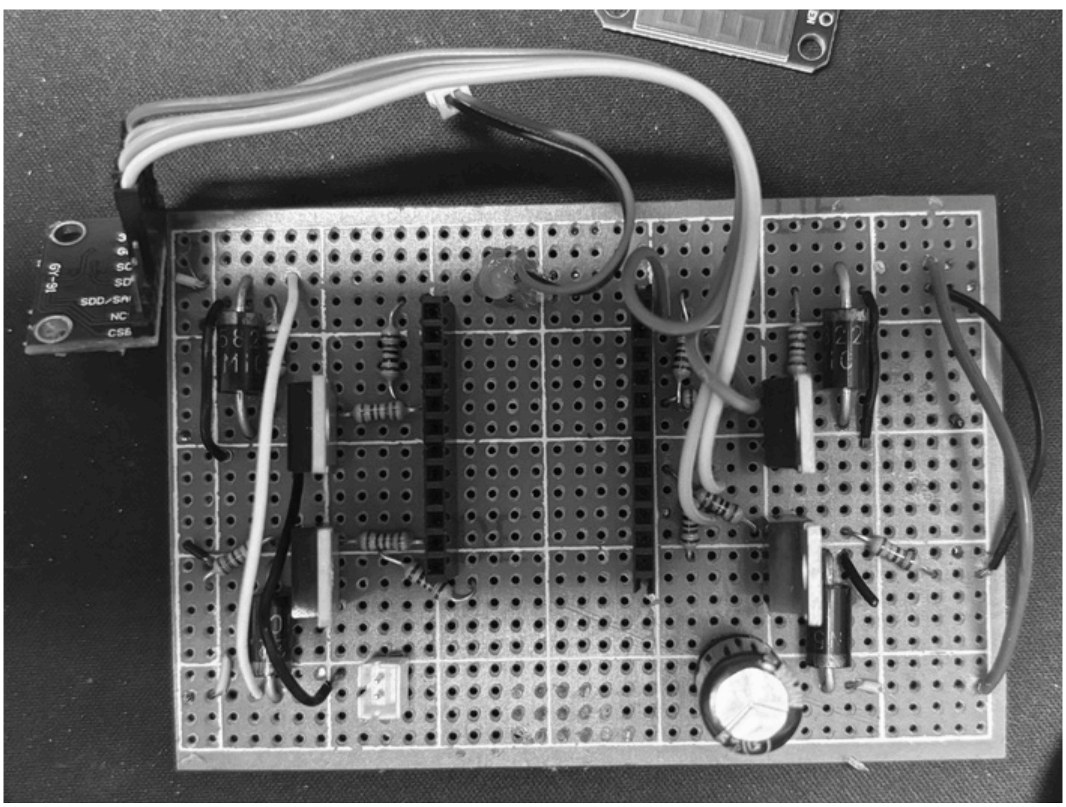


Figure 2.4.3-1 - Original protoboard

The next protoboard design is shown below. It featured thicker gauge wires that would handle the current draw better than the standard hobby wire. It also had more optimised rails that would allow for more even power distribution to the four motors.

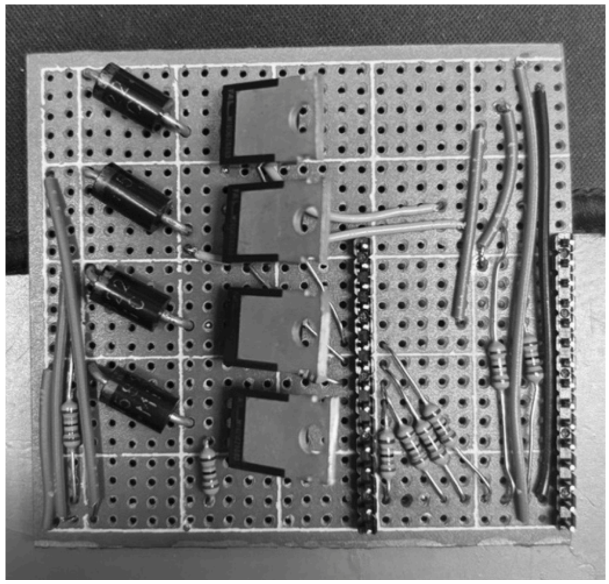


Figure 2.4.3-2 - MOSFET Protoboard Design

The newer protoboard design proved much better when utilised in the better chassis design concept 4 (see below), vibration was successfully minimised and power distribution was much more even, allowing the drone to "bunny hop". There was still some voltage sag which we attributed to the 1s battery not being capable of supplying the necessary voltage and current to the motors at maximum thrust.

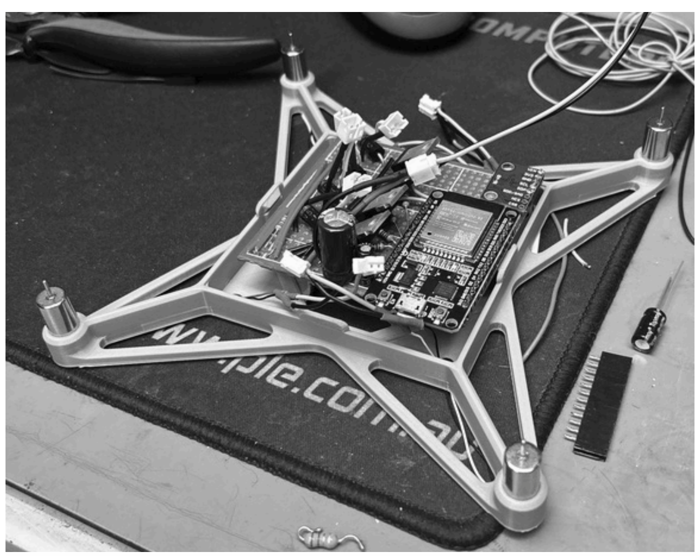


Figure 2.4.3-3 - Protoboard in Frame

To confirm that the voltage sag was due to the battery, the protoboard was connected to a bench power supply and the correct voltage was able to be supplied to new power rails and the motors where able to fly at full thrust (see below).

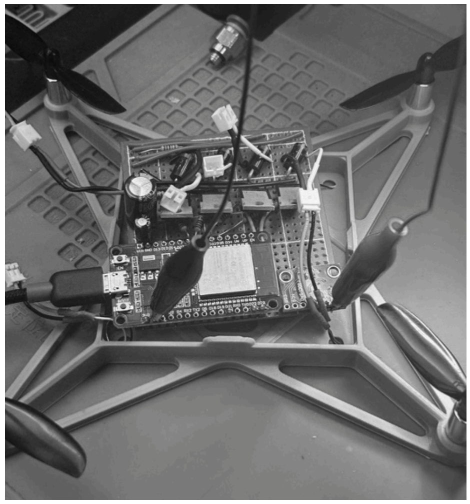


Figure 2.4.3-4 - Bench test

After trying the new PCB with the 2S battery it was found that the motor power circuit was not correctly configured (see PCB section). So, the testing of one motor was done using the bench power supply as seen below.

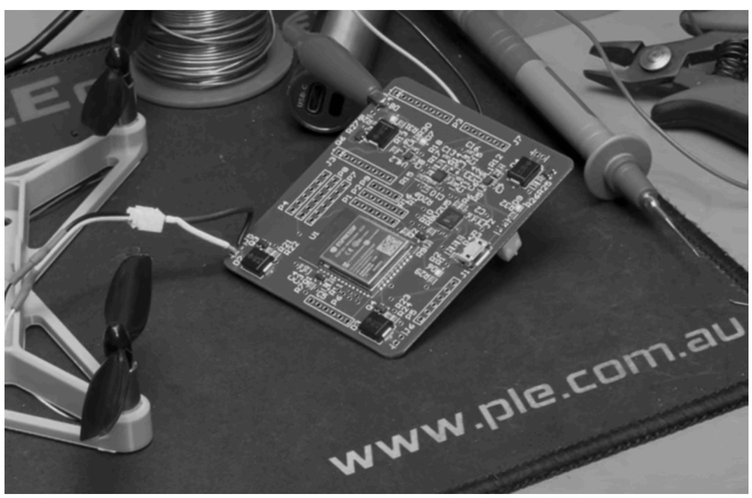


Figure 2.4.3-5 - Bench test on new PCB

Further corrections were made to improve reliability of the motor, including the use of external buck converters to handle current better than the custom PCB circuit.

2.5 Obstacle Detection and Altitude

2.5.1 Design Philosophy and Justification

The navigation algorithm forms the logical core of the indoor surveillance drone. The software sets out to manage how sensor data and flight commands interact to achieve stable and autonomous movement. The algorithm interprets input from the onboard sensors and follows a loop of motion commands ensuring the drone follows the pre-determined flight path while avoiding collisions in real time. The navigation software was implemented and written in C to maintain consistency, reliability and low-latency performance, following a pre-determined syntax and file layout by devised by the team to avoid any conflicts. The tasks were separated into distinct functional blocks i.e. movement control and obstacle avoidance. interacting with the data from the sensors and motor controller interface to ensure smooth command execution. The sensor data from 3 ToF sensors is continuously processed throughout the operation of the drone to monitor the drone's surrounding in all directions (REQ-017). If an obstacle is detected within a 0.5m threshold, the software is to stop moving and execute the corrective evasive manoeuvres before resuming on its original pre-defined path (REQ-002). Below in Figure 2.5.1-1 is a flow chart representation of this navigation algorithm, which is to be implemented in all directions.

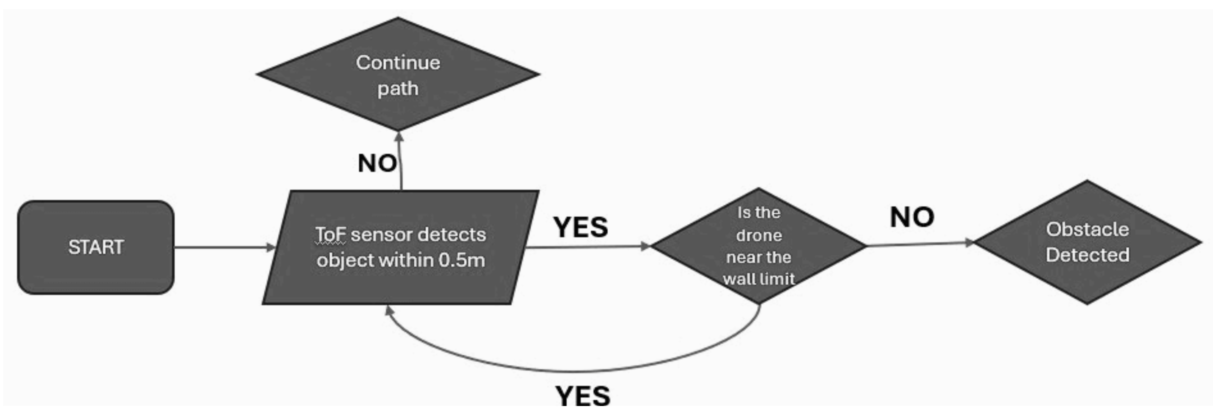


Figure 2.5.1-1 - Flowchart representation of Obstacle Avoidance algorithm

The navigation algorithm adapts its response depending on where the obstacle is detected (REQ-017). For an obstacle detected on the front sensor, if the distance between that front sensor to the obstacle is less than 0.5 metres, the system compares the left and right sensor readings to determine the clearer side. The drone will continuously move 0.1 m towards the clearer side while the distance between the obstacle and the front sensor is still less than 0.5 metres. Similarly, for an obstacle on the side of the drone, the algorithm compares both side sensors and then moves the drone in the opposite direction by 0.1 metres until the distance is more than 0.5 metres. As indicated, the algorithm also accounts for obstacles that are not placed perfectly in front or to the side of the drone. This approach enables smooth avoidance behaviour even for obstacles not perfectly placed in front or to the side of the drone. Overall, the navigation software acts as the connecting piece between the data from the navigation sensors and the flight control, enabling the drone to autonomously survey and fly in indoor spaces safely and efficiently.

2.5.2 Design Elements

There were 4 models of the ToF sensor considered these were HC-SR04, ev_mod_ch101-03-01, VL53L1X and VL53L1X. The HC-SR04 was initially considered due to the cheap cost, high availability, and ranging distance of 4m. The HC-SR04 was rejected due to the size of the component, at 45*20*15mm [5]. The HC-SR04 was deemed too large and heavy for the small profile of the drone as weight was a major consideration in determining if the drone would have enough thrust to lift off this needed to be minimised with every component. The HC-SR04 deemed unfit for the design alternative options were researched such as the ev_mod_ch101-03-01, VL53L0X and VL53L1X. The ev_mod_ch101-03-01 initially showed promise, the sensor boasted 8m ranging, with low power use and a large 180 FoV. These features were all contained within a 3.5*3.5*1.26mm package meeting the small form factor requirements for the drone size and weight [6]. The two issues identified with the ev_mod_ch101-03-01 were the high cost and library availability, use of this chip in the quantity desired being 5 sensors would consume too much of the budget causing issues where money needs to be spent in other areas, thus a cheaper option was required. The VL53L0X and VL53L1X were priced similarly, range up to 4m and have a small form factor of 25*10.7mm [7]. These sensors were easy to source from AliExpress and ordered for testing. After testing both the sensors the VL53L1X was found to provide more accurate readings than the VL5L0X and communicate with the esp more reliably, thus it was decided to move forward with the VL53L1X for both obstacle detection and altitude readings.

The ToF sensors were used for both obstacle detection and odometry and thus were the first components related to navigation that were tested. The ToF communicate over an I^2C interface requiring 3.3V to Vin, GND, SDA, and SCL. As multiple ToF will be in used and share the same SDA and SCL lines the XSHUT pin will also be required, this turns the sensor on and is active low allowing only 1 sensor to provide a reading at a time. An image of the sensor and the pinouts is shown in **Error! Reference source not found.**.

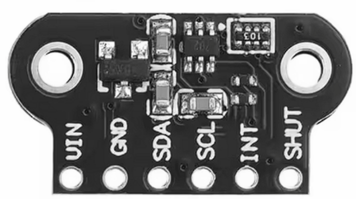


Figure 2.5.2-1 - VL53L1X Time of Flight Sensor

2.5.3 Testing and Results

For initial testing of the ToF sensors one sensor was purchased. The sensor was then wired to a ESP32-C3-mini as shown in Figure . Using the Adafruit VL53L1X library and Arduino IDE and some initial test code labelled “VL_Working_Code” this can be found in the GitHub repository along with a README. When the code is run the sensor provides data output to the serial monitor in the following format “Distance: 215 mm”. The sensor was then tested to see how much of the 4m the datasheet stated it was able to track could actually be measured, after testing in light and dim environments on light and dark surfaces it was found that the full range of 4m could be measured in a light environment of a white surface, for dimmer lighting or measuring of a darker surface closer to 2-3m was the achievable range. With the desired hovering height of the drone to be ~1.5m off the ground which is a dark carpeted surface and the obstacle detection to be a much closer distance of 0.5m as laid out by (REQ-017) the VL53L1X sensors were deemed ideal and an additional 4 were ordered for both obstacle detection and height sensing use cases.

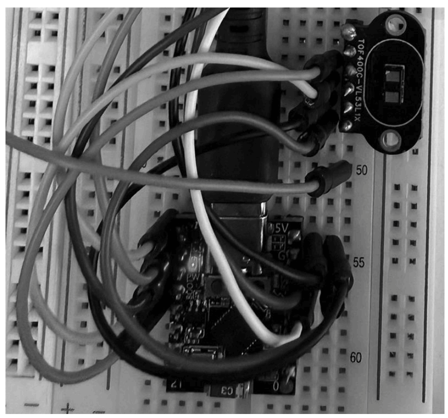


Figure 2.5.3-1 Initial test setup for VL3L1X

An issue noticed with the initial test was the code used blocking functions such as “delay()” this would stall other processes run on the ESP32 when called such as the drone motor PWM and communication processes. This led to more test code being written utilising non-blocking function setups such as “millis()” and can be found in the GitHub labelled as “VL_working_code_non_blocking” along with a README, this code operated exactly the same as the original sensor code without using blocking functions.

The next step in testing was to have multiple of the VL53L1X ToFs working at the same time, this was required as 1 sensor would be used for height readings and another 3 for obstacle detection on the front, left and right of the drone. To minimise pin usage on the ESP32 the VL53L1X could share the same SDA and SCL lines due to their I^2C interface and XSHUT pins where only 1 sensor would be activated when the sensors respective XSHUT was pulled low. Thus 4 more VL53L1X sensors were ordered from AliExpress, the sensors that arrived were not VL53L1X and instead VL53L0/1XV2, this used a slightly different library the “Adafruit VL53L0X” and came with a slightly smaller FoV by 2 degrees and from testing this sensor independently less capable in imperfect conditions, when this was discovered 4 more of the correct sensor were ordered from a more reputable vendor. Due to time constraints attempts to use the incorrect sensors with the correct sensor were made, ultimately the 4 incorrect sensors could work together although not when the correct VL53L1X sensor was in the mix. This issue was debugged to be an issue with being able to reassign addresses for the VL53L1X as it could not be reassigned form 0x29 and would interfere with the other sensors readings. The issue was eventually resolved by changing the VL53L1X sensor library to one from polulu which is called "VL53L1X" this library allowed for the VL53L1X sensors to have their addresses reassigned. Around the time this was discovered the new VL53L1X sensors arrived and to avoid issues with using different sensors and libraries the VL53L0/1XV2 were discontinued in favour of all VL53L1X. Code for testing all 5 VL53L1X together was written and can be found on the GitHub labelled as "5_working_VL53L1X_14_10_25" this code was using an ESP32-S3 for testing as this was going to be the chip used in the final design until it was changed to a ESP32-WROOM, the libraries used across both ESP are the same thus for testing this is fine as only minor pin changes need to be made. The "5_working_VL53L1X_14_10_25" allowed for 5 separate distance readings to be output to serial this was due to the use of the polulu library as a previous attempt had failed to work due to using the Adafruit library which cannot reassign addresses and can be found on the GitHub labelled "5_working_VL_XSHUt".

With all 5 desired ToF sensors now confirmed to be able to output accurate measurements using via I^2C sharing the same SDA and SCL pins with separate XSHUT pins driving each sensor high. The next step involved integrating an adjustable obstacle detection feature to 4 of the sensors left, right, front and back of the drone, while leaving the height sensor to read any value without obstacle detection. This was implemented into the code file "Obstacle_Detection_With_Height_15_10_25" and can be found on the GitHub, in this code there is a variable "OBSTACLE_LIMIT_MM" which is by default set to 400mm although can be adjusted to any value within the sensors range. Once an reading of below 400mm is read by any of the left, right, forward or back sensors a message will be printed to serial such as "Obstacle within 400mm Left". This serial print out will be replaced with sending a signal via the webserver to have the drone enter the "Hover" state once this is established.

As the altitude is simply determined by a reading from one of the ToF in which the obstacle detection has been neglected. This sensor faces to the ground and aids in satisfying (REQ-001) when hover is achieved. The obstacle detection code was finalised into "Obstacle_Dection_With_Height_15_10_25" which provides both obstacle detection at the same time as a height reading, this is useful for later implementation of the optical flow and is further discussed in section 2.6.

2.6 Navigation and Odometry Design

2.6.1 Design Philosophy and Justification

With the omission of a GPS (REQ-009) as per the project team's requirements, the navigation software was designed to enable autonomous indoor flight and obstacle avoidance, ensuring compliance with multiple requirements for indoor navigation. These requirements included a pre-determined horizontal and vertical flight path (REQ-003), with said path already stored for the drone to actively follow (REQ-004). The core design philosophy emphasises modularity, simplicity and safety. The navigation logic is to be implemented as a step-based control framework on the ESP32 MCU to ensure low latency, portability and consistency with other system components. Movement will be defined through discrete but simple commands such as "forward x metres" or "rotate y radians", a sequence of which can be coded to follow a pre-defined path. This deterministic structure greatly simplifies debugging, verification and traceability of system behaviour.

2.6.2 Design Elements

For accurate navigation of the drone, the drone position in x, y, and z coordinates are required. The x variable relates to lateral movement forward or backward, the y variable to lateral movement left or right and the z variable relates to the movement of the drone vertically (height). To achieve this feat an optical flow sensor (PMW3901) and a ToF (VL53L1X) were employed to operate using odometry – a method used to determine the change in position of an object by integrating changes in position over time. The optical flow sensor (PMW3901) was chosen as it is relatively cheap ~20$ and the data sheet specifically mentions the use is for "far field devices" and "indoor and outdoor X-Y positioning especially in GPS denied environments" has been reported [8]. The PMW3901 measures how ground texture moves across the camera field of view (FoV) and outputs the change in x and y motion counts from the previous frame to new frame, to relate these changes in the sensors view to actual displacement factors regarding the image sensor width, FoV and height must be known. The height data has a potential to change e.g. the drone flies over a desk, as such the height needs to be read live. The height data is provided by a ToF facing the ground and should be situated as close to the PMW3901 as possible. The other variables required for finding displacement are constants relating to the PMW3901, the FoV is 42 degrees and image sensor width is 35. An example of how the displacement calculation works is shown in *equation (1)*.

$$f_{pixels} = \frac{\frac{image\ sensor\ width}{2}}{\tan\left(\frac{FoV}{2}\right)} = \frac{17.5}{\tan\left(0.365\right)} = 45.6$$

$$\Delta X_{metres} = \frac{\Delta X_{count}}{f_{pixels}} * height$$

$$\Delta X_{metres} = \frac{5}{45.6} * 2 = 0.22m$$

With the optical flow and ToF an accurate way to find the displacement of the drone in x, y and z coordinates was found, this opened the door to potentially setting waypoints for the drone which could be used in navigation software.

The PMW3901 optical flow sensor can be viewed in Figure , this shows the available pin on the sensor. The sensor communicates with the ESP over an SPI interface, the minimum number of pins required to connect the PMW3901 are a 3.3V supply, GND, MOSI, CLK, MISO and CS. Additional pins can be connected such as MOT which is a pin relating to motion detection, however this pin will not be utilised.

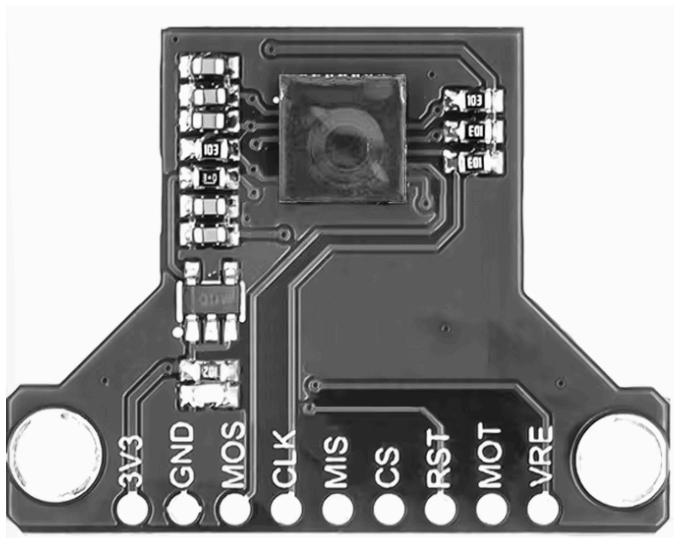


Figure 2.6.2-1 PMW3901 Optical Flow Sensor

2.6.3 Testing and Results

Testing of the navigation software was primarily carried out through Serial Monitor based simulation using fake sensor data to verify the core logic of the obstacle avoidance and navigation algorithms were working before any hardware integration was implemented. A series of pre-defined obstacle scenarios including a clear path, obstacle in front, obstacle to the left and obstacle to the right were used to validate the software correctly identified nearby obstacles and followed the correct evasive manoeuvre and resumed the planned path once clear. Upon connecting the ESP32 with the algorithm uploaded, the Serial Monitor outputs confirmed that the command sequence, obstacle detection and sidestep logic was operating as intended meeting the aforementioned requirements (REQ-002, REQ-017).

The navigation and obstacle algorithm serial logs were initially computed in the code "DroneNavTest". Each navigation command ("forward x metres" or "rotate y radians", etc.) was printed in real time to the serial monitor to validate proper logic and sequencing. Timing intervals and state progression were monitored to ensure that command updates remained stable under the 150-millisecond loop period. As shown below in Figure 2.6.3-1, the serial monitor logs show what the simulated drone will do in response to certain events like obstacle detection or latency issues. The navigation moves are sorted into cases such as case NAV_CMD_FORWARD which moves forward given a variable amount. The algorithm reads the ToF data from the readToF() function uses the avoidanceStep() function to determine its next move based on those readings.

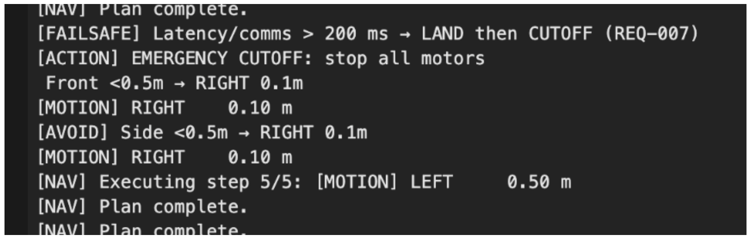


Figure 2.6.3-1 - Serial Monitor Logs

Now obstacle detection has been implemented and the drone's height reading available focus is shifted to odometry using the PMW3901 optical flow sensor. The PMW3901 was initially tested using a setup which involved an ESP32-C3-Mini, a VL53L1X ToF and the PMW3901, the setup is shown in Figure excluding the ToF.

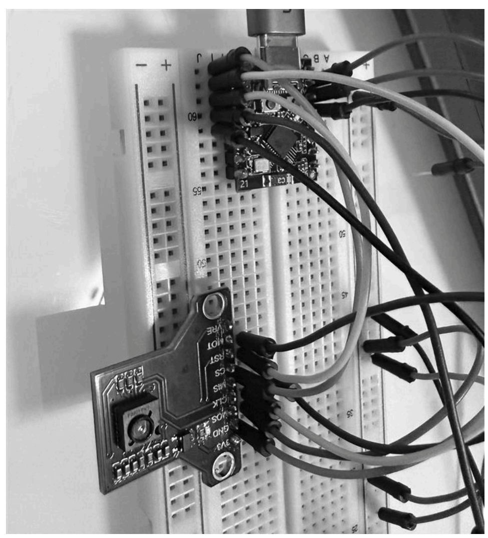


Figure 2.6.3-2 PMW3901 wired for testing to ESP32-C3-Mini

To ensure wiring to the PMW3901 was completed correctly and ensure the chips health example code "flow" from the "Bitcraze_PMW3901" library was used this simply output "dx(int), dy(int)" to the serial monitor showing x and y count shifts. Once wiring was confirmed initial test code began, "Odometrey_corrected_FoV" which can be found on the Github was uploaded and provided "dxdy(int,int) height (int)mm" as the ouput to serial, when the sensors were moved the counts for distance moved would increase, and decrease when moved back in the opposite direction. It was very apparent that scaling was incorrect which meant the distance readings were incorrect, another apparent issue was the height reading would jump in value +-20mm when at steady state likely due to the imperfect conditions of operation i.e. moderately reflective object and dim lighting. To combat these issues the next test would involve a filter for the height readings and calibration of the scaling by moving a known "X" and "Y" metres and then scaling the appropriate variable "K_pix_to_rad_x" or "K_pix_to_rad_y" respectively. To combat the height, issue the filter keeps a buffer of the last 4 values and picks the middle value thus rejecting outliers – the buffer size can be adjusted to better suit. Another filter on the height is the introduction of a slew-rate limiter this is adjustable and stops the sensor from jumping in height values by large amounts up or down as the drone is expected to be hovering at the same height. The final filter applied is an exponential moving average removing any remaining jitter in the readings, this can again be calibrated by adjusting variable alpha to find the desired spot between responsive and smooth readings. The exponential moving average follows equation (2) where $\Delta t = 50ms$.

$$\tau = \frac{\Delta t}{\ln{(1-\alpha)}} \; eq(2)$$

With the filter changes implemented a test was conducted where the optical flow was moved 1.2m at a set height of 1.1m – limited by the length of the ESP connection cord as battery power had not been established – the optical flow measured distance was then divided by the true 1.2m distance and multiplied by the initial scaling factor to provide a new accurate scaling factor. The test was then re-run and the measured distance was compared to the real distance, these readings matched within +-50mm. The relating code can be found in the GitHub as "ODOMETRY_1.2M_WORKED", limitations with this were noted as tested at a much lower height the readings did not appear as accurate. The potential solution for this is to find measured distance at different set heights and have multiple scaling factors or an adaptive scaling factor depending on the height being flown at.

With both obstacle detection and odometry operational the code was amalgamated into one code file so obstacle detection could work in unison with the odometry when PCB integration could be completed. This code file is "Obstacle_Height_Odometry" and can be found on the GitHub.

Way points are the next step for navigation as the drone needs to follow a defined pattern. The movement algorithm follows a step-based navigation plan consisting of commands such as "forward x metres", "rotate y radians" or "right x metres". The pre-determined flight path as such, will consist of a sequence of these commands (REQ-003), which is pre-emptively stored for the drone to follow (REQ-004). During operation, the software reads continuous distance measurements from the 4 ToF sensors positioned around the drone's front, left, right and back side. These readings are also processed with an obstacle avoidance routine that checks for nearby obstacles and issues corrective movement commands to follow before returning to the planned path. The drone is programmed to stop and hover when an obstacle is within 0.5 metres and will shut down propulsion if the obstacle persists for more than 30 seconds (REQ-002).

The code created for testing setting waypoints was "tuning_waypointing", in this code waypoints are accepted via serial text such as "move x 3m, move y 120cm" a comma separates the step of the process this example would first set a waypoint at +x 3000mm and then a second waypoint at +y 1200mm. The tolerance for completing a waypoint can be set by entering "t int" in serial where the int value relates to the mm +- to accept as the drone moving to the new waypoint – this has been implemented as some error is expected when moving. When testing this code success was variable the tolerance for completing a waypoint had to be made great as when only concerned about moving in the X direction, small changes in the Y direction would cause the waypoint step to not be seen as completed

2.7 PCB and Power Design

2.7.1 Design Philosophy and Justification

An iterative approach was taken in designing the internal circuitry of the drone, in order to meet requirements, and to ensure the individual design elements are compatible with one another (REQ-008 and REQ-021). Research was undertaken for the circuitry and components needed for various operations, and once each one of these operational blocks was designed, they were debugged using Altium Designer's validation tool, and by reading information from the data sheets.

The main requirements for the PCB were to: have a main processor that can receive code flashed onto it via USB from a laptop; have an array of sensors that will send data to the processor; have the capability to send sensor data via Wi-Fi; all circuitry on the PCB is powered by a portable battery (REQ-012, REQ-014 and REQ-026); motors are controlled by and that the PCB is as small and even as possible, to help with the weight budget for flying the drone (REQ-022).

2.7.2 Design Elements

Main Processor

The choice of main processor was changed a number of times throughout the design process, which resulted in many delays.

In initial team discussion, some form of ESP-32 architecture was made as the team's design choice, due to the team's familiarity with the system, through undertaking similar projects. However, this decision was later changed to using an STM-32 chip, specifically the STM32F030K6T6 [9]. The benefit the STM held over the ESP was that they have more dedicated GPIO pins, which was very useful to this project due to the large number of inputs and outputs for the motors, sensors, and camera. From researching other drone designs the team also found that STM-32s are the standard for drone processors.

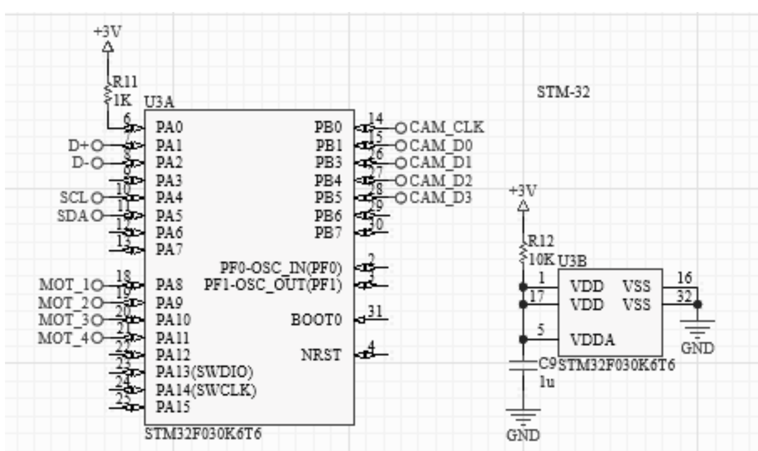


Figure 2-2.7.2-1 ***Processor Architecture for Drone Schematic v1.5***

The team then ended up reverting back to the initial plan of using a form of ESP, specifically the ESP32- S3R8 [10]. There were several motivations behind this decision: the pinouts of the ESP-32 were more familiar, and so there was no additional learning curve; the team already had access to ESP-32 dev kits, and so testing and prototyping could begin earlier, without having to wait on new parts; the STM-32 required its own specific IDE for flashing code, that not a single team member was familiar with. In the process of this design change, it was realised that a UART module was needed to convert the data from the USB port to be readable by the ESP. The FT232RNL-REEL [11] was chosen for the team's UART.

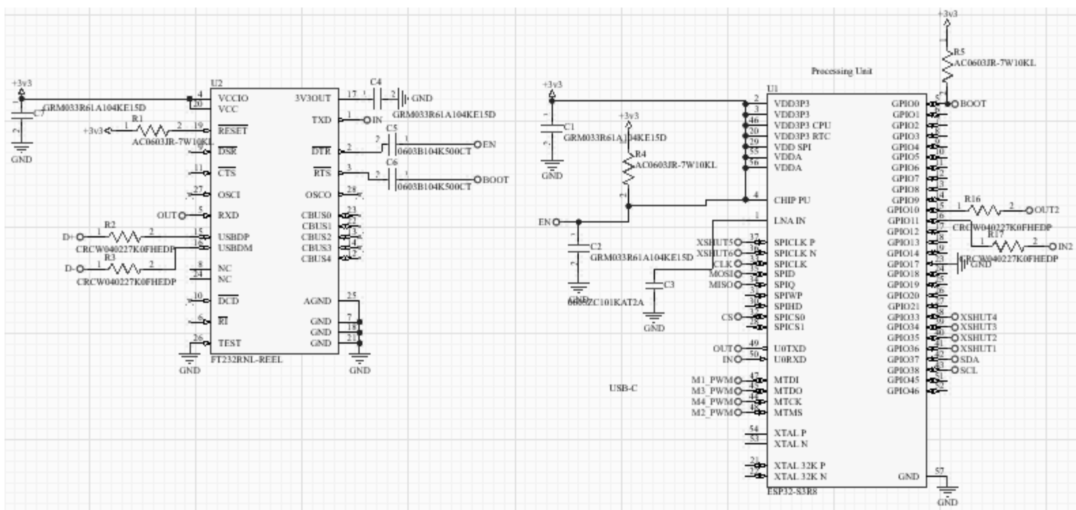


Figure 2.7.2-2 - Processor Architecture for Drone Schematic v1.8

In the final iteration of the PCB, v2.0, the ESP and UART were remodelled based off of a pre-existing ESP-32 devkit [12], due to issues with flashing from the USB. The S3 model had major issues during physical testing, as it required its own external clock, which was not accounted for in the design. So, the final ESP chosen was a WROOM model [13], as it has its own external clock. An added advantage is that the WROOM has an antenna, so the need to flash information outputs to the webserver is no longer allocated to the ESP-32 CAM (see section 2.8). The UART chip was also changed to a CP2102N model. [14]. More sophisticated enabling and booting was also added using MOSFET logic that is controlled from the UART.

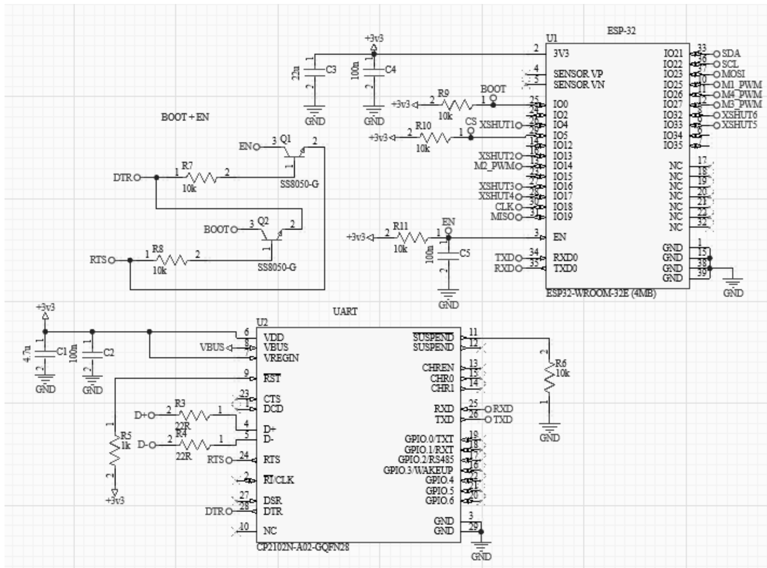


Figure 2.7.2-3 - Processor Architecture for Drone Schematic v2.0

Power Conversion

A number of different arrangements were made for the voltage and current distributed across the PCB. The power source for the PCB was decided to be a 1S Li-Po battery, due to its light weight, appropriate current draw, and 3.7V nominal voltage. For the first design the battery voltage directly powered the motors. A +3v3 buck converter, the MIC5219 [15] was used to step down the battery voltage for powering the ESP, UART, ESP CAM, and sensors.

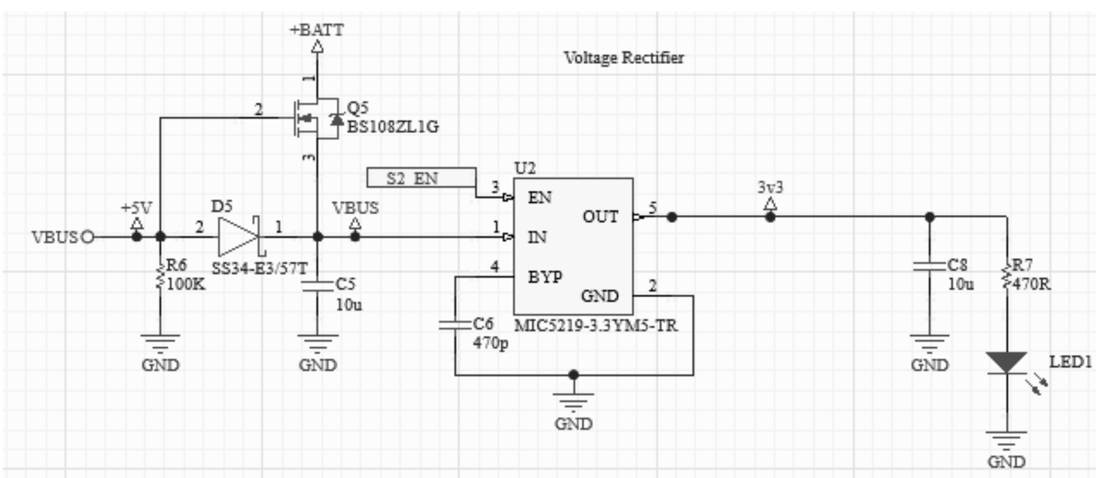


Figure 2.7.2-4 - Processor Architecture for Drone Schematic v2.0

The team also changed their requirement to have battery charging capability on the drone (REQ-012), during the process, as it was not required in the brief, and was deemed to conflict with safety precautions.

After testing with the ESP-32 cam, it was realised that the +3v3 voltage would not be sufficient for booting, so a +5V rail had to be added. This was achieved through a boost converter with direct input from the battery. These output rails were designed by using the +5V boost converter, which fed into the TPS62160 [16] +3v3 buck converter.

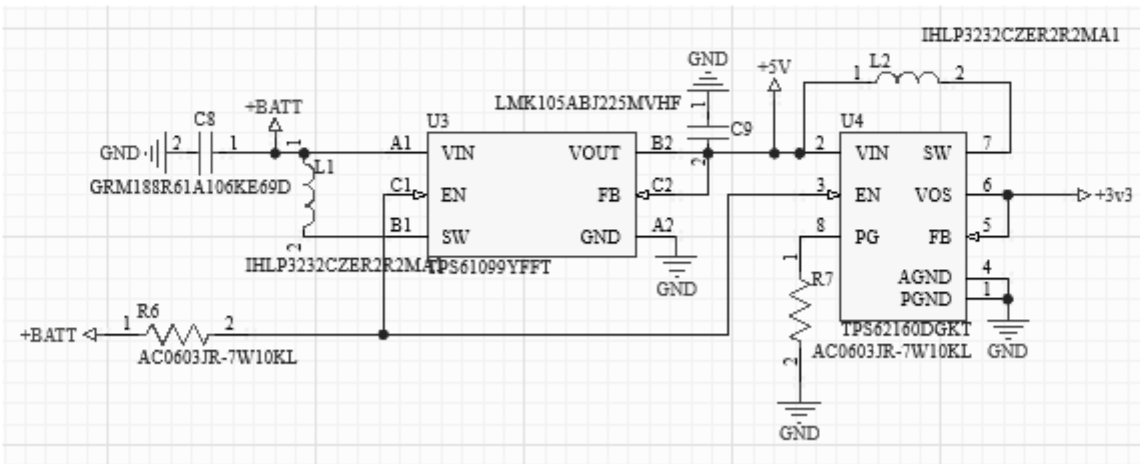


Figure 2.7.2-5 - Voltage Conversion for Drone Schematic v1.11

Through testing, it was discovered that there were major issues with the voltage conversion. Initially, the +5V rail was outputting the same voltage as the battery, and the +3v3 rail was not outputting any power at all. The +3v3 rail issue was diagnosed by the inductor output going to the incorrect pin on the buck converter. However, the output rail remained as the battery voltage. So, while the team could confirm power was distributing across the PCB, the buck and boost converter were not operating.

Therefore, the power operations had to be rethought. The first change made was to change the batteries from 1S to 2S, (i.e. +3.7V to +7.4V) so there was a cleaner power supply to the motors (see section 2.4), and buck converters, rather than boost, were used to power the integrated circuit chips, which will give a steadier voltage and current than boost conversion. This leaves us with three different power conversion circuits: a +3v3 for the ESP, UART, and sensors, a +5V for the ESP CAM, and a half step-down converter for the motors. The +3v3 and +5V buck convertors were chosen from the TPS616X chip family from Texas Instruments [17], as they were both designed specifically to output these voltages with minimal external components. The third buck converter was the TPS568230 [18], which was more complicated to implement due to the high current outputs required.

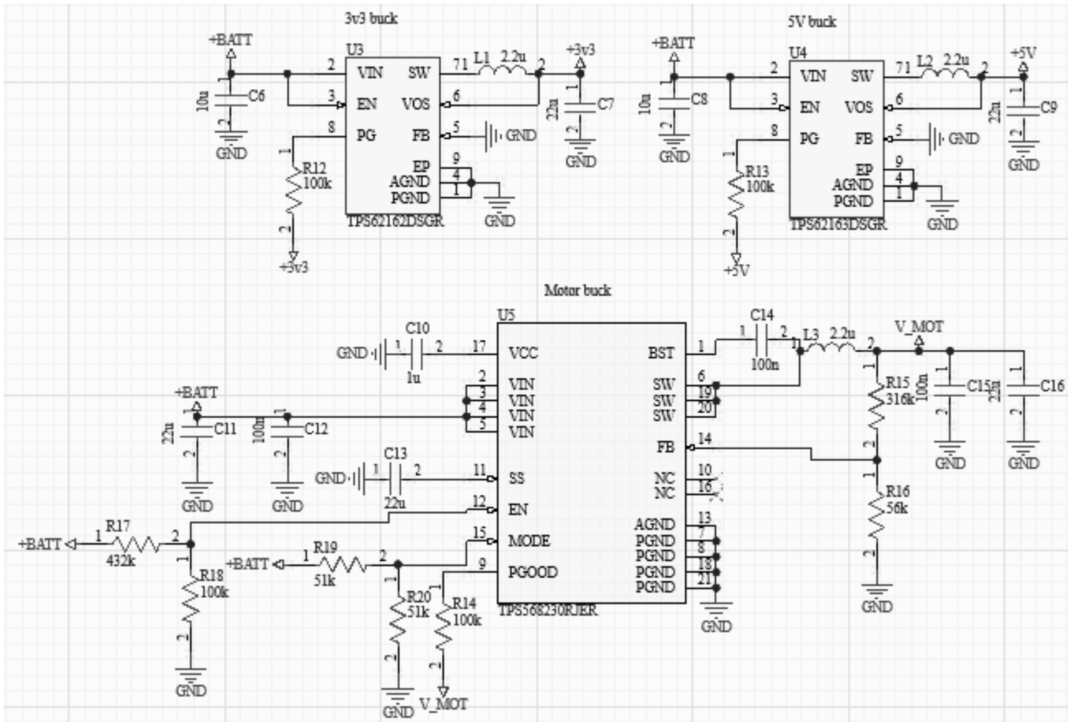


Figure 2.7.2-6 - Voltage Conversion for Drone Schematic v2.0

Motor Drivers

A couple of different design ideas were made for the motor circuitry (see section 2.4). Initial research was made for using brushless motors, however these required ESCs, which were deemed too expensive to buy commercially, and too complicated to make a custom design within the timeframe.

After finalising a brushed motor design, there was some debate as to whether a pre-existing motor encoder should be used, or a simple, custom design. The custom design was chosen, as it was easier to prototype and test for consistency with the ordered PCB.

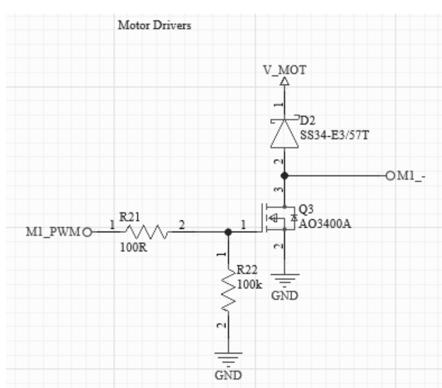


Figure 2.7.2-7 - Motor Driver for Drone Schematic v2.0

Ports and Peripherals

In the initial design of the PCB, the sensor and motor pins were traced together, for ease of routing architecture. This was done based on whether the sensors use I^2C or SPI pins. While this was convenient for a draft design, this required too much wiring to be practically implemented, particularly due to the fact that time of flight controllers needed to be on each face of the chassis, and motor drivers at each corner. This was major motivation to change the board from a 2-layer design to a 4-layer.

During the routing stage of PCB design, the USB-C port that was to be used for flashing onto the board, was changed to the USB-Micro. This was due to the USB-C being complicated to route, having two pairs of each data pin, which would have crossing traces, making the design more complicated.

Board Architecture

Between v1 and v2 of the PCB, the team decided to change the PCB from a 2-layer design to a 4-layer. This was motivated by a number of reasons, but mainly because it could remove traces for +3v3 and allow twice the amount of space for signal routes.

The initial 2-layer design dedicated the top layer for signals, and the bottom layer was a ground plane. The 4-layer design, however, had a signal layer as the top, followed by a ground plane as the second layer, a +3v3 power plane as the third, and then another signal layer at the bottom.

#	Name	Material	Type	Weight	Thickness	Dk	Df
	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask		0.01016mm	3.8	
1	Top Layer 1	CF-004	Signal	1oz	0.035mm		
	Dielectric 2	PP-022	Prepreg		0.2104mm	4.4	0.02
2	Top Layer		Plane	1oz	0.0152mm		
	Dielectric 1	Core-025	Core		1.065mm	4.6	0.02
3	Bottom Layer		Plane	1oz	0.0152mm		
	Dielectric 3	PP-022	Prepreg		0.2104mm	4.4	0.02
4	Bottom Layer 1	CF-004	Signal	1oz	0.035mm		
	Bottom Solder	Solder Resist	Solder Mask		0.01016mm	3.8	
	Bottom Overlay		Overlay				

Figure 2.7.2-8 - Board Layers for Drone PCB v2.0

Impedance characteristics and layer weight were determined based off of the standard manufacturing settings for JLC PCB.

2.7.3 Testing and Results

Testing

Constant testing was undertaken whilst designing the PCB schematic in Altium Designer, however there was only so much that simulations could reveal.

When the first iteration of the PCB was shipped in, a number of physical tests were made. Firstly, 3.7 volts was applied to the input terminals, to see if the board would draw current. Unfortunately, this first test did not yield promising results, as the power draw from the board was not significant, which seemed to imply the integrated circuit chips were not turning on.

This was further explored by checking the potential values for the different voltage rails. The team found that the battery rails were functional, however, the +5V rail was just outputting the battery voltage, and the +3v3 had no output. After checking the data sheet for the +3v3 buck converter, the zero output was easily diagnosed as an inductor going across the wrong pins, but the same issue remained of no voltage conversion.

On top of this, the port was not recognised by the IDE on the laptop, and code could not be flashed on the ESP, so no further tests could be made.

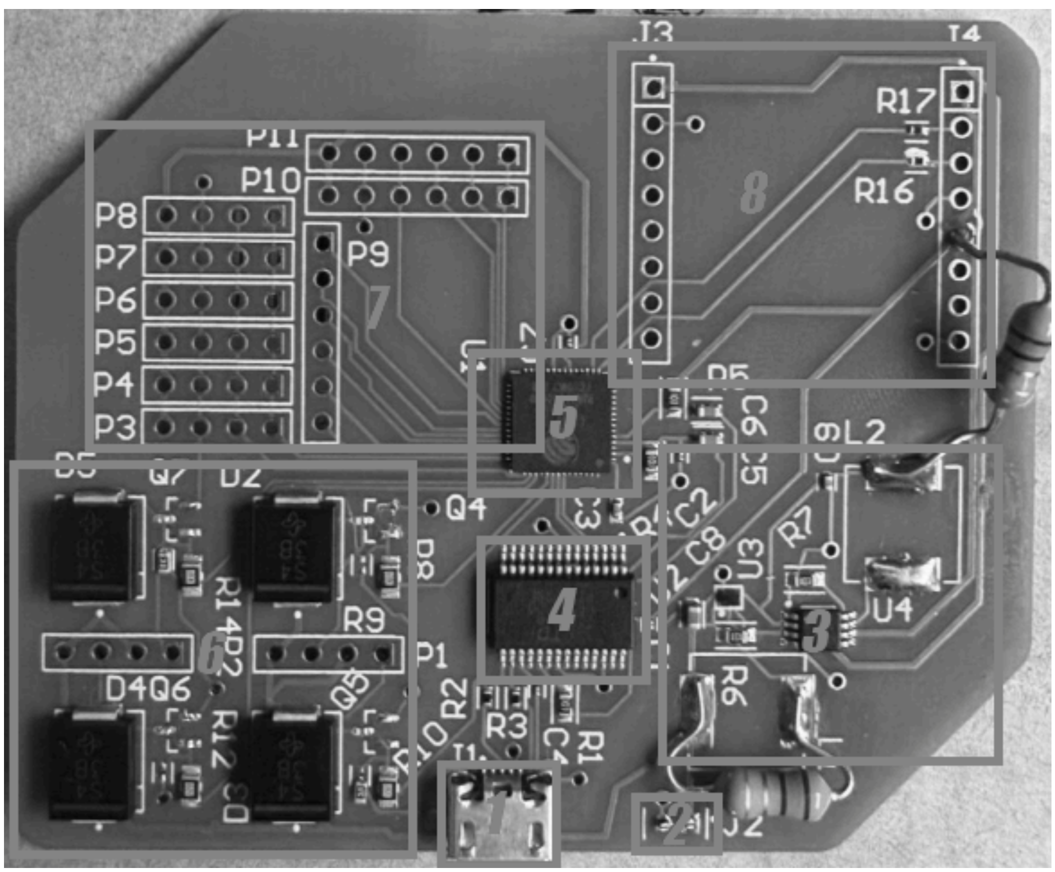


Figure 2.7.3-1 - PCB v1.12

Table 2-4 - PCB v1.12 Function Breakdown

Label	Name	Functionality
1	USB-Micro	Data input for flashing.
2	Power port	1S battery terminal input.
3	Voltage converter	Splits the +3.7V battery voltage into +3v3 and +5V rails.
4	UART	Converts parallel data into serial.
5	ESP32-S3	The main processing chip, controls circuit peripherals.
6	Motor	Controls motor speed, from a PWM signal.
7	Sensor ports	Inputs and outputs for time of flight, optical flow, and gyroscopic sensors.
8	ESP-CAM	Streams camera footage and sensor outputs.

For the second iteration of the PCB LEDs were added to the board so that the voltage rails could be checked instantly. This was a success for all of the voltage rails, except for the motor rail, which was fluctuating.

This time around flashing and webserver connection from the ESP-32 worked to full effect, and motors could be controlled wirelessly. The sensors could also be attached and sent data to the ESP reliably. Some rewiring was needed however to some inconsistencies with output pin order and the order the pins were on the sensors.

Also, the 7.4V to 3.7V converter posed issues, as some of the components could not handle the 8A that was needed to provide all four motors, which resulted in an inductor going past its rated heat value and rendered obsolete. This meant an external buck convert had to be used for motor driving.

Final Design

The final PCB design was accumulated throughout the team's design process and to meet the integration requirements for each of the other design elements. The team ended up with a 72.4mm x 62.3mm, four-layer drone control PCB that has wireless PWM control and sensor feedback.

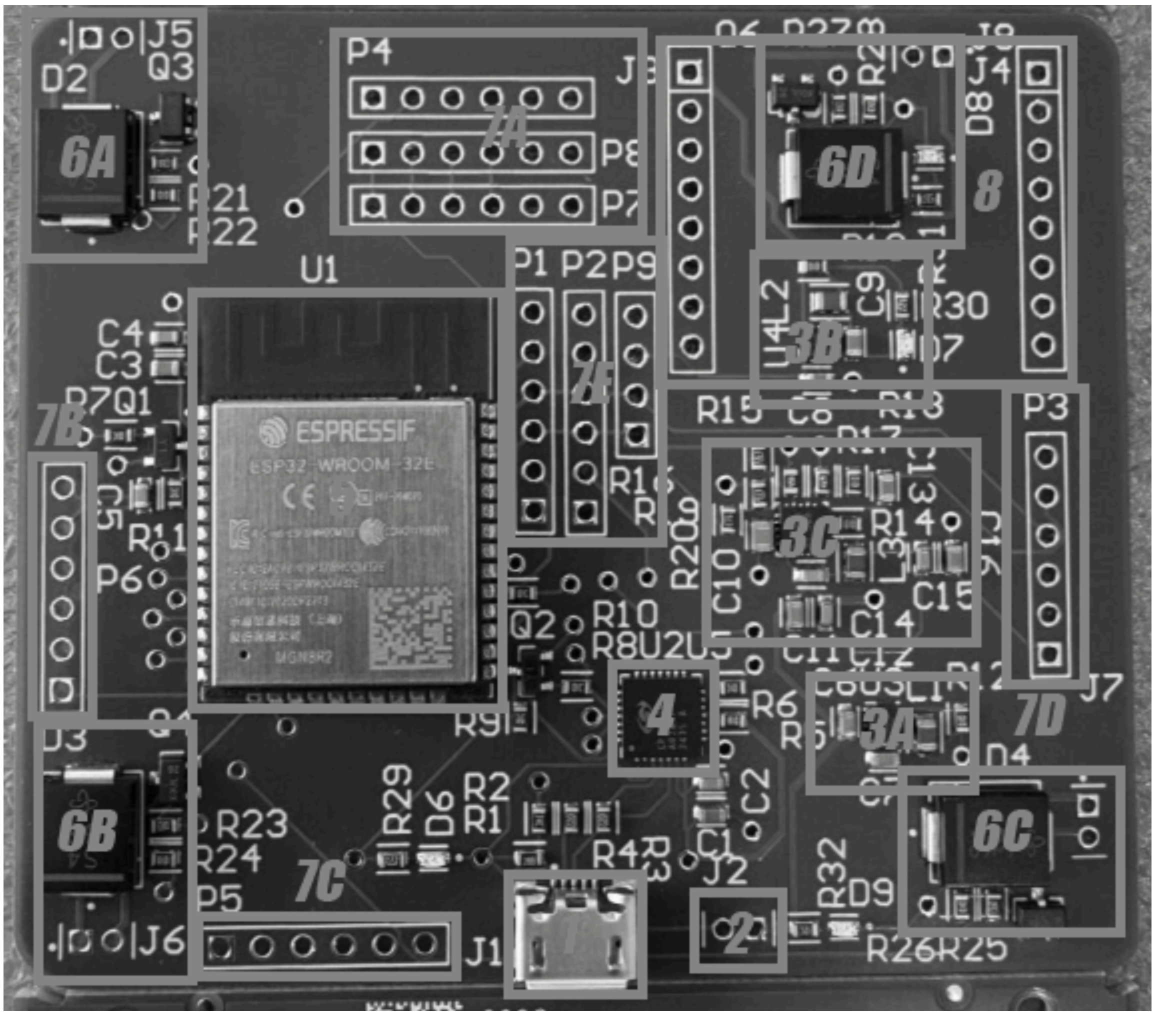


Figure 2.7.3-2 - PCB v2.0

Table 2-5 PCB v2.0 Function Breakdown

Label	Name	Functionality
1	USB-Micro	Data input for flashing.
2	Power port	2S battery terminal input.
3A	+3v3 converter	Converts the battery voltage to 3.3V.
3B	+5V converter	Converts the battery voltage to 5V.
3C	Motor voltage converter.	Converts the battery voltage to 3.7V.
4	UART	Converts parallel data into serial.
5	ESP32-WROOM	The main processing chip. Controls the motors and reads sensor data.
6A	Motor driver 1	Controls motor speed, from a PWM signal.
6B	Motor driver 2	"…"
6C	Motor driver 3	"…"
6D	Motor driver 4	"…"
7A	Front sensors	Time of flight and optical flow.
7B	Left sensor	Time of flight.
7C	Back sensor	Time of flight.
7D	Right sensor	Time of flight
7E	Top and bottom sensors	Time of flight and
8	ESP-CAM	Streams camera footage.

2.8 Server Control and Surveillance Design

2.8.1 Design Philosophy and Justification

Communication with the drone must be performed wirelessly. There are a number of options available when using the ESP32. The main two options are Wi-Fi or Bluetooth. The choice was made to use Wi-Fi as it would have higher bandwidth when streaming data and would allow for more reliable communication with the drone, this is particularly important when considering safety features such as the kill switch.

Having selected Wi-Fi as the medium for communication, the interface must be chosen. The standout choice was to use a webserver that can be set to allow for an input from the user regarding the path the drone should trace as well as producing flight logs and readouts with regards to the various on-board sensors.

As part of the project scope, the implementation also includes a watchdog timer and safer overrides to prevent uncontrolled flights in the event of sensor or communications failures. The design also integrates a watchdog and latency monitor to trigger a controlled landing or propulsion shutdown if communication exceeds a 200-millisecond latency or fails (REQ-007), in conjunction with the hover system design.

2.8.2 Design Elements

The communication system also involves the camera system. The communication system was initially chosen to utilise the built in Wi-Fi module of the ESP32CAM, this would allow for the control of the main ESP32 chip to be facilitated by a webserver on the CAM and then have instructions related to flight sent via a UART link.

It was later decided that the safest option for flight control and communication would be to instead utilise an ESP32 Wroom chip as the main controller. This would allow for the control of flight to be facilitated separate from the video feed. This was chosen due to the instability of the ESP32CAM to accept UART signals when also streaming video.

The choice of separating the flight control web interface with the video interface gives greater reliability and consistency to both flight communication (REQ-025, REQ-030, REQ-031) and can provide a better video feed.

Having separated the different interfaces, video and data/control, the next step is to enhance each web interface separately. The video stream web server must be kept simple so that transmission of video can be prioritised and allow for simple viewing. This would simply require that a Wi-Fi enabled connect to the ESP32CAM and load into the web server where the video stream would begin instantly.

As for the control, the webserver can be much more intricate. The webserver would need three main features: sensor data, path input, and the kill switch.

Sensor data is important because the fallback design (no flying or motors) would require the sensor to output an instruction to the webserver that can be interpreted in place of a motor action. E.g if the drone is close to an object in front of it, the output on the webserver would say “Object in front, drone stops movement”. This would allow the algorithm to be displayed during demonstration despite the lack of functioning motors.

Path input would consist of a terminal-like window in which the user can input a simple command sequence that would be interpreted by the drone as movement. This would allow on-the-fly path input commands, giving the drone the ability to quickly adapt to new requirements and to follow any path that the user may want with little delay.

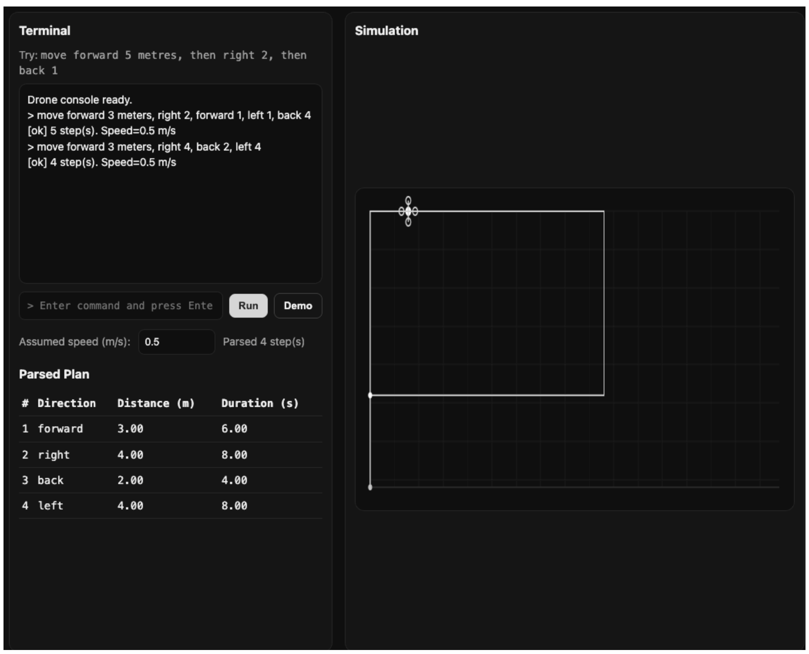


Figure 2.8.2-1 - Example drone webserver with simulation

Concretely, the ESP32 stores timestamps of the last heartbeat received from the browser (/hn HTTP endpoint or a WebSocket “hb” message). A background check compares millis() against this timestamp, i.e. if no heartbeat arrives for more than 200 milliseconds, the controller latches a failsafe and immediately cuts off propulsion. The Disarm function in the Flight Controller code enables the user to stop all motors from the web server. Manual re-arming is required after the trip. This mechanism satisfies REQ-007 by integrating with the hover control and guaranteeing a deterministic shutdown under link loss or excessive latency. The control web server provides the feature of a kill switch and emits the periodic heartbeat used by the watchdog timer. If the flight controller network stalls, web server is stopped or latency exceeds 200 ms, it executes the failsafe and disarms automatically.

2.8.3 Testing and Results

The ESPCAM has an example script in Arduino IDE that is very useful for testing the various resolutions available. This webserver script was initially used. It was found that the access point version of the Wi-Fi connection is the best for the intended application. Since the Unifi network has a more sophisticated login system, it is simpler the use the ESPCAM as an access point, using a router would be complicated in future applications.

Later the code was streamlined to only include the video on the webserver. The webpage was then changed to include different resolutions. The different resolutions understandably had varying framerates and hence vary levels of trade-off between quality and framerate.

The final webserver code settled on a resolution of 360p as can be seen in the figure below.

Figure 2.8.3-1 ESP-CAM webpage view

2.9 Stakeholder Engagement

Throughout the project life cycle, active engagement with stakeholders ensured that design, testing and safety requirements were aligned with academic expectations and regulatory standards. The involved stakeholders included, but were not limited to, the UWA Aviation Laboratory staff, academic supervisors and the team members acting as both developers and test operators.

Supervisor and Client Engagement

Regular meetings were held with academic supervisors to review design process, assess feasibility and ensure compliance with scope, UWA safety standards and CASA regulations.

Key feedback from supervisors included:

1. Simplifying the power distribution network and replacing brushless ESCs with brushed motor drivers to ease integration.
2. Implementing clear sensor calibration procedures before each test to ensure repeatable results.
3. Using a 4-layer PCB instead of a 2-layer to ease the congestion of circuit connections
4. Emphasising design with propellor guard fitment and emergency shutdown option for compliance with scope.
5. Using an ESP32 instead of an STM32 due to difficulty programming the STM32

This feedback from the academic supervisors was carefully considered, with these corresponding hardware/software changes being accepted and implemented into the final design.

Design Review and Feedback

2 formal Design Reviews were conducted throughout the project timeline

Design Review 1: Focused on defining subsystem interfaces, communication protocols and establishing software testing framework
Design Review 2: Addressed integration issues between optical flow sensor and the MCU, and changes made to components and design due to setbacks or feedback from 1st design review.

Feedback from these sessions guided hardware and firmware refinements, improved integration of sensing modules and reinforced safety compliance within the final prototype.

Technical Queries Integration

The Technical Queries register was used to record and track formal questions to stakeholders and their binding responses. Each entry captured: ID, description, resolution and date. These queries documented the team's communication with the stakeholders and clarification of project requirements, ensuring every design decision was validated through official feedback. The TQ process allowed the team to confirm compliance boundaries and clarify ambiguous technical details such as desired component specifications or operational constraints. An example of this process included the sensors being integrated into the PCB for a simple design. This tool enabled the team to maintain a clear record of stakeholder input and guarantee that feedback was properly addressed, documented and implemented in the final prototype.

As indicated below in Table 5, each TQ represented a targeted design uncertainty that required stakeholder validation before the team could progress. This table also documents how the stakeholder validation influenced the final drone design including components used, features explored/dropped or even clarifying project scope. This process ensured that upcoming design decisions ranging from component selection to integration strategy remained aligned with project objectives and client expectations.

Table 2-6 - Technical Queries between Team and Stakeholders

Query Title	Description	Response	Resolution	Date
Camera Orientation	Does the drone's camera have to focus on a particular point during flight or simply the path in front of the drone?	That should be determined by the team based on comparable products on the market	Team prioritised stability and coverage over complex image tracking	28/7/25
ESC Purchase	Can the team purchase an ESC pre-made or does this have to be made by the team?	It should be part of the custom PCB	ESCs proved to be expensive and difficult to produce so ESCs were dropped from the final design	28/7/25
Flashing Open-Source Software	Can the team flash open source (Betaflight) software to the STM32 or does it have to be full custom code in C++ written by the team?	It does not need to be fully custom code, but team must understand and explain codebase, so large and complicated software should be the default choice	Team ended up using own code for simplicity and understanding	28/7/25
Motors	Does the client prefer a specific motor type between brushed or brushless motors?	Motor type is team's choice	Brushless motors required ESCs, so brushed motors ended up being chosen.	28/7/25
Manual Control	Project brief states communications require manual control of the drone. Are there requirements for this to be a function of the hand-held controller or can they be controlled through a device?	Through a computer is okay.	The drone can connect to a Web Server via an iPhone or computer	28/7/25

Query Title	Description	Response	Resolution	Date
Integration	Is the team allowed to integrate off-the-shelf modules such as a PMW3901 or ICM-20948 into the system as individual components or is the team required to redesign and embed their circuitry directly into custom single PCB?	ICM-20948 should be integrated with PCB as the design is simple and does not require special PCB. Other modules should be assessed by the team	Sensors and other hardware components were chosen carefully so that only 1 central PCB was required rather than any special PCBs	28/7/25
ESPCAM Module	Can the team integrate an ESPCAM Module as the drone's security camera or would the client prefer an optical camera directly wired to the PCB?	The client prefers the camera be directly wired to the PCB. However, ESP-CAM is also acceptable if it has benefits over a camera directly wired to the PCB	ESP32CAM allowed for the control of ESP32 chip to be facilitated by a webserver with instructions sent by UART link. However, ESPWroom chip ended up being used due to instability with ESPCAM video feed	13/8/25
Wind Force Specification	On the Beaufort Wind Force Scale in the air at the drone's position, what wind rating does the drone need to maintain a stable hover in	Wind will be created by a large fan in the room. Use commercially available fans to calculate/measure wind speeds	Drone chassis was designed to be sturdy and survive normal wind speeds	13/8/25
Charging Infrastructure	Does the drone need to be rechargeable through a USB-C port, or does it need to be externally?	Battery should be charged with an external USB charger, where battery is disconnected from the drone and connected to charger	USB-C port was initially chosen. However, due to complications with routing, Micro-USB was used instead.	22/8/25

2.10 Safety and Ethical Issues

The following of safety and ethical standards were integral to the design and testing process of the indoor surveillance drone. The team recognised that although the system operated in a controlled laboratory environment, the drone still posed significant mechanical, electrical and privacy-related risks if operated without proper procedures. From a safety perspective, all testing closely followed UWA laboratory safety protocols. Hazard identification was finalized prior to each test with a Risk assessment and Method statement being completed before entering the lab room. The relevant safety hazards were meticulously ranked based on likelihood, consequences and exposure with mitigation and control measures being prepared in advance. Risk control measures were implemented in accordance with UWA's risk management framework to minimise danger to any persons. The top risks in the project were use of high voltage equipment (plasma cutters), loss of signal/control, slips and trips, shared workspace, Projectiles (propellor strike/component detachment), burns and flammability risk.

High voltage equipment was placed as a 300H due to the high consequence of the risk; a plasma cutter was used to cut the drone body out of aluminium for the initial frame design. To mitigate the possibility of injury from this equipment a strategy was placed in which high voltage equipment would need to be clearly labelled and not used without supervision. This mitigation strategy brought the risk down to a 15L. Loss of signal/control was placed as 300M due to a moderate consequence rating and a high likelihood. The risk with loss of signal/control is the drone moving in an unintended manor, this brings the possibility of collision with bystanders or dangerous equipment. To mitigate the risk propellor guards, clearing the test area and retaining a visual line of sight with the drone, and a watchdog timer system for automatic shutdown in the event of communication loss were implemented. This mitigation strategy brought the risk down to a 10L.

Slips and trips were placed as a 180M risk due to high likelihood and a low-moderate consequence. The workspace in which the drone is constructed and tested contains many tripping hazards, these range from tables and chairs to loose cables from equipment such as soldering irons. To mitigate this risk control measures were put in place, this included inspecting the workspace prior to entry to alert members of potential hazard locations (loose cables), if the workspace was messy the team was to clean up before commencing work removing hazards from the floor. Enclosed shoes were required and personal belongings to be stored off the ground. These control measures reduced the risk to a 15L. The shared workspace was another risk ranked at 180M due to high likelihood and low-moderate consequence. The risk with the shared workspace is other students and drone operators, poor communication with other teams could result in hazardous situations such as a drone being active as members of another team are entering the room. To mitigate this risk control measures were placed in which any potential hazards will be reported to the area supervisor, and testing will only be completed when other teams are not present. To stop other teams entering during a test one team member must pay attention to the entrance and inform other groups of the hazard. These control measures bring the potential risk down to 30L.

Procedures and aspects such as the use of propellor guards, visual line of sight operation and manual emergency shutdown functions were requirements devised by the team designed to prevent harm to personnel or property. Propellor guards and failsafe program were an initial stakeholder priority requirement to ensure users were protected from the sharp rotors and has a contingency for if the drone loses connection and or control. Appropriate protective equipment and clothing was worn during the manufacturing and testing of the drone to minimise likelihood and impact of possible injuries. The team was also limited to testing in an empty enclosed indoor space avoiding interference with other lab users.

From an ethical perspective, the team adhered to the Engineers Australia Code of Ethics [19], ensuring all design decisions and testing were undertaken with safety and ethical usage in mind. The project avoided any form of data misuse by restricting camera operation for demonstration and testing purposes only. This restriction ensured video data remained local and was not stored or transmitted externally, maintaining honesty and integrity [19, 19]. The drone was not tested or used in the presence of any unrelated bystanders, eliminating the ethical risk of privacy breaches or unnecessary involvement of other people, respecting the dignity of other persons [19]. The watchdog and failsafe systems were added as an important requirement for the team to adhere to ensure the safety of the clients using the drone.

Environmental considerations were also considered by using rechargeable batteries, re-using old components and minimising material waste where possible. These environmental concerns addressed by the team fostered the environmental, health and safety considerations into the engineering task [19]. The team also aimed to minimize power usage and noise pollution choosing components and designing the drone to be able to fly for at least 3 minutes without the need for full power and only flying the drone in an empty sound-proof room to avoid disturbing bystanders.

From a broader ethical standpoint, the team also recognised the importance of transparency, accountability and public trust in emerging autonomous systems. All software developed for this project was documented, version-controlled and shared with supervisors and clients on the GitHub repository with every script defined and justified with its purpose and features. This allowed the team to maintain traceability and allowed for peer verification of safety-critical code for the watchdog and failsafe systems. The team exercised professional judgement by clearly defining system limitations and ensuring the drone only operated in controlled environments under supervision [19]. These practices demonstrated the team's commitment to transparency and responsibility reflects the Engineers Australia principles of leadership, integrity and sustainability [19]. In doing so, the team not only prioritised user and bystander safety but also contributed to the ethical progression of autonomous technologies. By seamlessly integrating both ethical and safety considerations into each design stage, the team ensured the project not only met technical performance standards but also the professional practicum of responsibilities entrusted to project engineers.

2.11 Assembly/Build Process and Issues

For the manufacturing of the chassis the group used a CNC Plasma cutter for the parts due to one of the group members having free access to this. This was not necessary due to the later switch to a full ABS design.

This design was printed on a Voron 2.4 using ABS, originally issues were encountered with printing overhangs as pictured below with the stringing effect due to layer height mismatches.

Figure 2.8.3-1 - Printing process

With this fixed the final build print quality was extremely high as pictured below with clean seams.

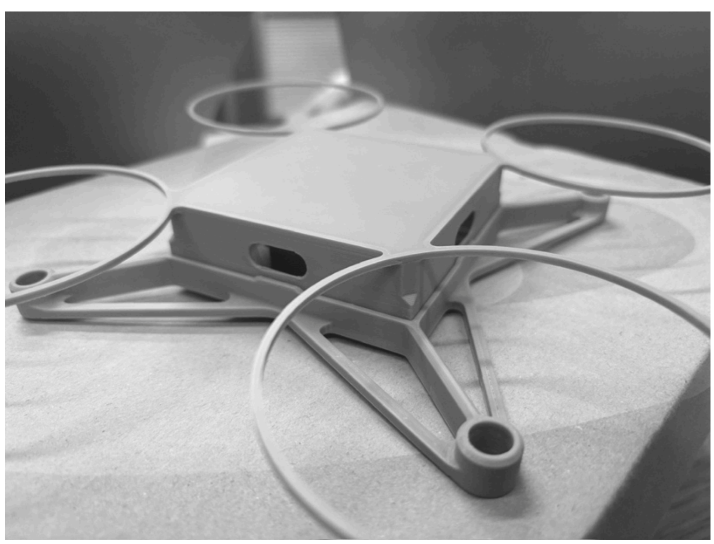


Figure 2.8.3-2 - Improved drone print

Prototyping was completed separately for each of the main systems, obstacle detection, optical flow navigation, hover/flying and chassis. The prototyping was completed via breadboarding and then protoboards after sensor responses were confirmed.

There were a few issues with the protoboard's PCBA missing resistors; to fix this was done through a microscope and a hot air gun. The lesson learned by the team was sticking to standard pad sizes when manufacturing the PCB, to avoid misplacements of parts.

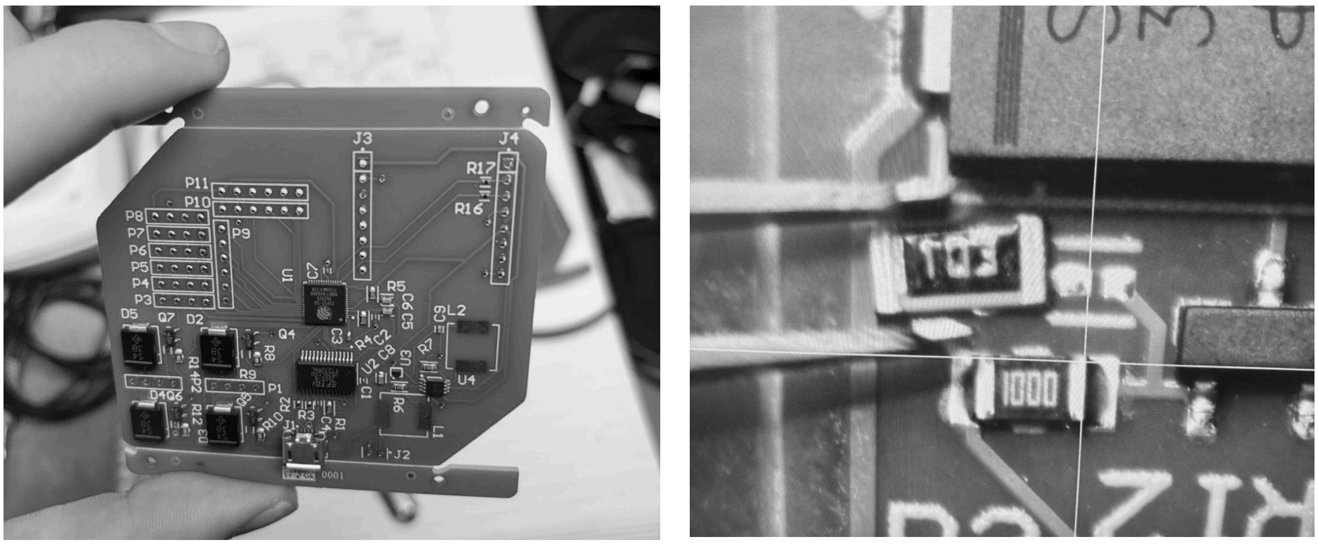


Figure 2.8.3-3 PCB Iteration 1 (Left) and Microscope view of resistor attachment (Right)

Another issue with manufacturing was the constant soldering and desoldering of boards, sometimes damaging traces and in extreme cases causing them to come off the PCB completely. Similarly, there were dangers the testing of the boards power system where the team blew an inductor. A solution to this was using a thermal camera during all tests as pictured below.

Figure 2.8.3-4 - Heat Camera view of PCB during testing

On arrival of the final PCB there was very careful testing of all sensors on a voltage and current controlled test bench. Every pin was tested for voltage levels under basic operations and the heat camera pictured above was used to identify hotspots. An issue identified which was mentioned in PCB & Power, was the issue with the voltage divider to the enable pin on the motor driver's buck circuit. This was identified as pictured below with the probe touching the resistor reading 029.

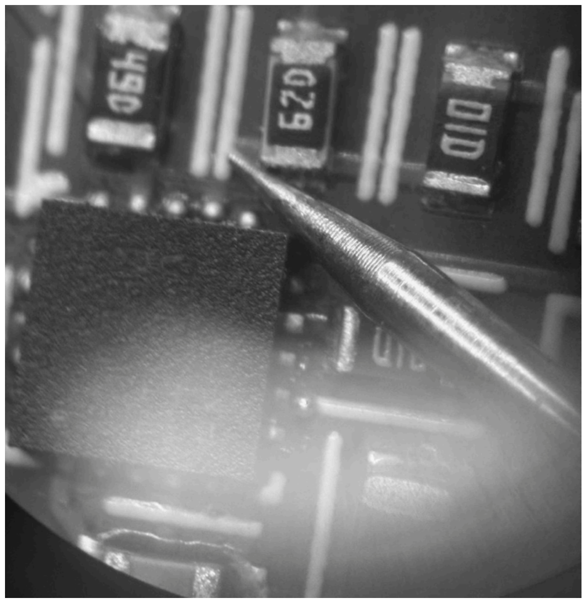


Figure 2.8.3-5 - Microscope view of PCB

Below is the final assembled PCB with all sensors attached, a few sensors had to have their pins twisted to fit in the port. Similarly, the gyroscope pictured in purple below has a red wire to fix the active high I2C enable.

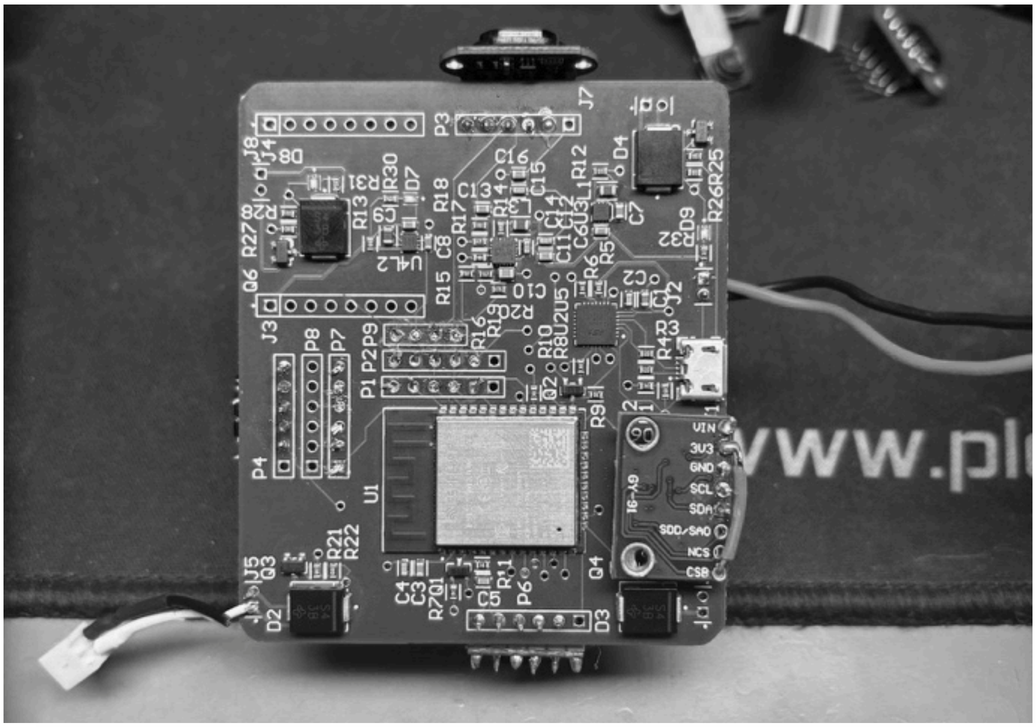


Figure 2.8.3-6 – Final PCB

The final PCB is a significant improvement on the first iteration. The UART works which was extremely important for debugging and problem solving. All of the sensors were able to be configured to work as intended, making the final PCB a success when it comes to navigational capability.

The remaining issues, however, are that the motor power circuitry is not sufficient for the high current draw experienced with the brushed motors. See section 3 for further improvements to the final product.

2.12 Final Integration of Software

The final firmware consolidates sensing, control, navigation and communications into a single ESP32-WROOM application with two runtime modes: **Flight** (motors enabled) and **Demo** (motors inhibited). All modules share a non-blocking, time-sliced main loop so that sensing and links remain responsive under load.

Sensing layer. Five VL53L1X ToF sensors share one I²C bus using XSHUT sequencing for per-device address assignment. One sensor is dedicated to altitude, four provide horizontal sectors for obstacle flags. This matches the verified five-sensor bring-up and address-reassignment approach developed during testing. The PMW3901 optical-flow sensor provides planar motion; its output is fused with altitude to estimate ground-relative velocity and odometry. Calibration and simple filtering (median/EMA/slew on height) from the odometry tests are retained to suppress jitter and improve scale.

Control layer. A conceptual hover loop runs continuously: read orientation/height, compare to target, compute small PID corrections, blend with baseline thrust, write four motor commands. The arming sequence applies a throttle ramp and checks guards (battery, sensor validity, link) before closing the loop. The same logic runs in **Demo Mode** with motor outputs suppressed so behaviour is still visible in telemetry/UI during lab walkthroughs.

Navigation & Avoidance. The navigation "stepper" executes queued motion primitives (e.g., forward d, rotate θ), while a guard layer enforces a **0.5 m** stand-off. If a sector reports < 0.5 m in the commanded direction, the stepper arrests motion, requests a short sidestep/turn into a clearer sector, and then resumes. This mirrors the REQ-017 test framing used during obstacle-avoidance development.

Communications & Safety. The flight controller hosts the control web UI (sensor data, path input, kill switch). It also monitors a lightweight heartbeat from the browser; failure to receive heartbeats or latency beyond ~200 ms triggers a deterministic disarm. This integrates the watchdog/kill behaviour proven in Section 2.8 into the unified firmware.

Scheduling & non-blocking operation. All sensor reads and comms use non-blocking patterns derived from earlier prototypes (e.g., replacing delay() with millis-based state machines), ensuring motor control and the web UI remain responsive.

Build and configuration. The codebase selects pins and addresses for ESP32-WROOM hardware (migrated from S3 test rigs with minimal changes) and includes a compile-time flag or runtime toggle for **Flight** vs **Demo** modes.

2.13 Risk Analysis

Risks for the project were identified by team discussion at the beginning of the project. The top 5 risks identified by the team were high lead times in deliveries, incompatibility of parts, PCB design, team availability/communication and software integration failure. Mitigation strategies were created for each risk in the case that it would eventuate.

High lead times for delivery of parts was viewed as a Top 5 risk due to the devastating impact it would have on overall progress. In this project due to the low budget of $350 [20], most of the parts would have to be ordered as cheaply as possible. This meant using sites such as AliExpress an online retail service owned and operated out of China as this site provided the best prices. The downside to the cheaper pricing by using the Chinese retailer is the delivery times, almost all the parts required from the project to be ordered from AliExpress had a lead time of ~2 weeks. In the event these high lead time components had any further delay this a cascading issue would be created, setting back the testing and report writing stages of the project. To mitigate the impact of this risk, backup components could be identified from local retailers such as Aptronics, although this would likely push the project overbudget. In the event a component arrived broken another ~2 weeks would be required for a replacement to arrive; to mitigate this issue, parts would be ordered with redundancy e.g. 4 sensors are required, order 5.

Incompatibility of parts was another high-risk aspect of the project identified. Due to the project being split amongst many team members parts selected could clash in terms of pins required for communication with the selected chipset. A lack of compatibility between components would mean each aspect may only work by itself rather than as a whole. The mitigation strategy to prevent this risk from eventuating was to find the required pins for the sensors required for the section the team member was working on. Once, all the information on components were found the team would together discuss and draw out a wiring diagram to ensure there was enough pins on the chosen chipset and that no clashes in pins required were occurring. The wiring diagram created for this can be found in Appendix A.

PCB design has been identified as a high risk to the project. None of the team members have past experience with PCB design, let alone on Altium and due to the small timeline to design a PCB the risk of errors in component sizing and traces being run correctly very possible. The PCB is important not just to tie all the different sensors together into one working system, but in terms of weight savings as well. A non-working PCB which can utilise onboard PCB components which are smaller and lighter in weight than components generally used with protoboards, would result in a much larger and heavier drone. The increased weight and size, increases the likelihood of issues with balancing, movement, flight authority and wind resistance. To mitigate the chances of a non-working PCB, multiple design reviews will be conducted where each member will check traces for their sensors are correctly laid out and placed in a logical area of the board. For example, motor drivers are to be placed in the 4 corners of the board. In the event the board arrives with issues that cannot be resolved by jumping traces or soldering extra components a protoboard and local offboard components will be used to create a working drone.

Team availability/communication has the potential to be very damaging to project completion as the task is too large for a group of less than 5-6 individuals, in the event a team member is unavailable one week, progress may be hindered as a smaller percentage of the team has to take on more than 8 hours of work to prevent progress being slowed or final product quality decreasing. This risk was considered likely due to team members having varying commitments to other units and work outside of study. To mitigate the effects of this risk and decrease the likelihood of occurrence regular meetings at the same time each week were determined by utilising the website “when2meet” this allowed the team to align when every member was free and find the best time to meet [21]. In the event that aspects of the task were seen to be neglected each team member had the responsibility to raise this issue at the next meeting or in the team’s chat when noticed so a group member with a best role aligning to that item could be appointed.

Finally, the last significant risk identified was software integration failure, particularly concerning communication between the various hardware components and the flight control algorithms. As the project relied on multiple custom firmware modules written by different team members, there was a huge potential risk of code incompatibility, unexpected bugs or timing conflicts when merging subsystems such as navigation, motor control and wireless communication. Such issues could have led to critical functionality failures or unstable flight behaviour. To mitigate this, the team implemented incremental testing, version control and clearly defined every common variable and function used between different pieces of code to verify each module independently before system integration. Regular team reviews and debugging were also adopted to identify issues early in development and prevent any delays in testing.

2.14 List of Design Outputs

List of all documentation and code created to support design as seen in the GitHub, including prototyping code. The GitHub can be accessed using the following link. With each script, the GitHub folder contains a readme file that thoroughly explains the function of the code.
https://github.com/Crypt16/ELEC5552-Team-2-02-Indoor-Surveillance-Drone

Table 2-7 - List of Design Outputs in the GitHub

Deliverable	Type	Purpose	Storage Location
Drone Manual	Document	Details how to use drone for non-technical audience	GitHub => Drone Manual
Drone Frame Design	Schematic	Defines the drone's structure, dimensions and assembly layout	GitHub => Drone Design
Printed Circuit Board Design	Schematic	Defines the PCB's design and connections	GitHub => PCB Teams => Schematics => Circuit Board => Schematics => V2.0
Firmware Source Code (ESP32 Flight Control and System Integration)	Code	Implements all the code for real-time sensor acquisition, motor control, navigation, obstacle avoidance and telemetry via Wi-Fi into one script	GitHub => Final Code Teams => PCBDroneDam
Final Report	Document	Provides complete summary of design process, compliance, testing and outcome of the project	GitHub => Final Report
Functional Prototype (Indoor Quadcopter)	Physical Prototype	Final working drone for testing stability, obstacle avoidance and streaming	UWA Aviation Lab
DroneNavTest	Code	Serial Monitor outputs with obstacle avoidance algorithm implementation	GitHub => Prototyping Code => DroneNavTest
Drone Web Server	Code	Implements Web Server connected from ESP to a phone or computer	GitHub => Prototyping Code => Drone Web Server
ESP Gyro	Code	Reads data from an MPU-6050 IMU module and displays roll, pitch, and yaw on the T-Display S3's TFT screen. Applies a complementary filter to fuse accelerometer and gyroscope data for stable roll and pitch reading	GitHub => Prototyping Code => ESP Gyro
Flight Controller	Code	Turns an ESP32 WROOM-32 dev board into a simple flight-controller core with web	GitHub => Prototyping Code => Flight Controller

Deliverable	Type	Purpose	Storage Location
		control. It can arm/disarm motors, read an MPU-6050 IMU, run a lightweight roll/pitch PID stabilizer, mix outputs to 4 motors, and expose a tiny web API + WebSocket for status and tuning.	
Navigation_Combined_No_Webserver_No_Motor	Code	Combines PMW3901 optical-flow odometry with multi-VL53L1X ToF ranging (via XSHUT pin control) to simulate waypoint navigation and basic obstacle detection. It does not drive motors—it prints state, pose, and events over Serial to validate sensing and logic before wiring actuator	GitHub => Prototyping Code => Navigation
16_10_2025_Odometry_with_s3		Computes planar odometry on an **ESP32-S3** (e.g., TTGO T-Display S3) using a **PMW3901** optical-flow sensor, with optional height from a **VL53L1X** ToF sensor. It converts flow counts → velocities using FoV & height, then integrates to (x,y)	GitHub => Prototyping Code => Odometry
ODOMETRY_1.2M_WORKED	Code	Performs planar odometry with a **PMW3901** optical-flow sensor (and optional **VL53L1X/L0X** height) on an **ESP32-S3** (e.g., TTGO T-Display S3). It is tuned for a test in which the rig successfully travelled **≈1.2 m**, validating scale and drift	GitHub => Prototyping Code => Odometry
Obstacle_Height_Odometry	Code	ESP32 sketch that fuses PMW3901 optical flow (SPI) for XY ground motion, multiple VL53L1X ToF sensors (I2C) for obstacles (L/R/F/B), one VL53L1X ToF dedicated to	GitHub => Prototyping Code => Odometry

Deliverable	Type	Purpose	Storage Location
		height-above-ground	
5_working_VL53L1X_14_10_25	Code	A proven setup that initialises and reads five VL53L1X ToF sensors on a single I²C bus using XSHUT sequencing and per-device addressing. Target board is an ESP32-S3 (e.g., TTGO T-Display S3)	GitHub => Prototyping Code => VL53L1X
VL_working_code_non_blocking	Code	Reads distance from a VL53L1X time-of-flight sensor on an ESP32-S3 (TTGO T-Display S3) using a fully non-blocking state machine. It repeatedly prints raw and moving-average distances to the Serial Monitor and automatically retries sensor bring-up if anything fails.	GitHub => Prototyping Code => VL53L1X
VL_working_code_non_blocking_av_5	Code	A variant of the non-blocking VL53L1X reader tailored around a **fixed 5-sample moving average** for steady, low-noise distance reporting. It uses a compact state machine so the loop never blocks while waiting for new measurements	GitHub => Prototyping Code => VL53L1X
working_way_points	Code	Fuses **PMW3901 optical flow** with a **VL53L1X ToF** height sensor to estimate 2D pose (x, y) and execute simple **queued waypoint moves** entered over Serial (e.g., `move x 3m, move y -0.5m`). Runs on ESP-32 C3 Mini	GitHub => Prototyping Code => Waypointing

2.15 Final Cost Table and Budget

The total cost to manufacture the drone had many differences to the team's initial budget plan. The total final costing table compared to the budget is shown below. Note that there are some discrepancies between totals and individual item costs due to exchange rates and rounding to the nearest cent, please see invoices for more accurate costings.

Table 2-8 - Final Cost Table

Area of Purchase	Allocated Budget (AUD)	Item	Quantity	Unit Price (AUD)	Total Cost (AUD)
PCB	$100	4-layer PCB	5	$2.13	$10.65
		Setup fee	2	$18.99	$37.98
		Stencil	2	$5.97	$11.94
		Components			$33.72
		Extended Components fee			$70.63
		SMT Assembly	2	$1.53	$3.06
		Packaging fee	2	$0.36	$0.72
		Special Components fee	2	$0.12	$0.24
		TOTAL Merchandise			**$168.91**
		Shipping	1	$37.11	$37.11
		Discount	1	-$23.28	-$23.28
		TOTAL			**$184.70**
Power circuitry	$50	Buck converter	2	$5.96	$11.92
		2S Li-Po batteries	2	$21.00	$42.00
		TOTAL			**$53.92**
	$75	Motors	4	$5.22	$20.88
		Dampeners	1	$10.79	$10.79
		TOTAL			**$31.68**
External components	$75	Time of Flight	5	$8.94	$44.70
		IMU	1	$8.99	$8.99
		Optical Flow	1	$1.99	$1.99
		ESP-CAM	1	$19.15	$19.15
		TOTAL			**$74.83**
GRAND TOTAL	**< $350**				**$345.13**

The biggest area of expense was the PCB manufacturing, and also the area that exceeded the budget the most. This can mainly be attributed to the extended components fee, which is a fee that JLC PCB charges for each component outside of their standard component library. Unfortunately, many of the components on the PCB were not a part of this library and did not have alternatives that would not incur this fee. Express shipping was also required to get the PCB delivered in enough time to allow sufficient testing and assembly allocation, which incurred an additional fee of $37.11.

The buck convertors for the motors were also nought for testing purposes; however, they ended up being crucial for the final design, due to the failure of the motor buck converter on the PCB. This added to the power circuitry budget; However, this was mostly mitigated by only buying two of the 2S Li-Po batteries.

Motor costs actually ended up being cheaper than expected, as commercial brushed motors are widely available and do not use expensive parts to manufacture. So, this meant this section in the budget well under what was allocated.

Overall, the main source of budget constraint was the PCB, which almost ended up being twice what was allocated. Luckily, there was enough leeway in the other sections of the budget, so that the project did not go over budget.

3. Recommendations for Further Development

3.1 Next Steps for Prototype

There were multiple of issues that would be solved with alternate design decisions. A future prototype would most likely have fewer changes from iteration two of the drone than iteration two had from iteration one. This is due to the number of changes that were made to iteration two that made it more likely to succeed at a commercial level.

3.1.1 Brushless Motors

One standout change would be to change the design to include brushless motors. Although not explicitly excluded from the design brief, the use of commercial ESCs was the limiting factor in the decision to use the brushed motors. Utilising brushless motors would significantly improve the product in multiple ways. Firstly, brushless motors draw much less power, this would extend the flying time to beyond the 3-minute target initially set. Secondly, brushless motors provide significantly more thrust than their brushed counterparts [22]. Thirdly, brushless motors do not get as hot as brushed motors which allows for less focus on airflow around the motor body and opens avenues to more integrated chassis design. Brushless motors also have a significantly longer operating lifetime than brushed motors.

In a commercial setting, these changes would make the product significantly more attractive than its competitors. Longer flight time would open the drone up to potentially new applications such as surveillance from a further distance where flying in a larger area is required or simply for surveying an area for a longer period, both are commercially advantageous. The higher thrust would allow for less restriction on weight; this could allow for more features on the drone or a longer flying time for the same weight. Further, the longer operational lifespan could also be particularly desirable in commercial applications as it would allow the drone to operate in harsher conditions and in areas where replacement parts are not easily accessible.

3.1.2 USB Type C

Another change that is minor in its effect on the design but large in its effect on the commercial viability of the product is to use a USB type C port. This is because that type of cable is the industry standard and, in some cases, must be used to ensure legality [23].

3.1.3 Improved Power Management Infrastructure

During testing of the second PCB iteration, it was discovered that the power circuitry for the motors was not properly designed. The motors draw approximately 2A of current at maximum draw, with a steady draw of at least 1.5A. This was an issue when testing only one motor (see Test M004) on the motor power circuit. During this test, one of the inductors on the circuit melted from the excessive current draw. The inductor was later found to be inappropriate for the current it was conducting, raising the question as to whether the rest of the power circuit is adequately designed. In future, the design should account for this excessive current. To implement these changes, heat sinks were used on the hotter components such as MOSFETs, capacitors and inductors. Thicker traces to also ensured the board itself can handle large current.

3.1.4 *Carbon fibre chassis*

A further weight reduction could be found in using carbon fibre for the chassis. This would again improve the thrust-to-weight ratio of the drone that could lead to significant improvements in flight time and versatility. There are, however, some manufacturing challenges related to the construction of a carbon fibre chassis; carbon fibre is hazardous to cut and work with so must be manufactured with appropriate caution.

3.1.5 *Vibration Dampening*

An issue that arose early with significant impact was the vibration of the brushed motors causing the gyro to drift in the roll measurement. This made hovering almost impossible as with incorrect yaw it is not possible to tune the PID correctly for stable flight. This can be improved using rubber O-rings on the motor mounts or silicon pads to isolate the vibration from the IMU.

3.2 Required Approvals / Compliance

There are several commercial and technical standards that need to be met to commercialise the drone.

3.2.1 *Compliance with CASA Airworthiness and Design Standards*

In order to successfully market the final product, it may need to pass CASA airworthiness. This only applies if the drone is to be sold as a specialist or professional drone device or if it is flown over people. This could be the case if the team wished to have the drone used in surveillance around people [24].

3.2.2 *Electrical and EMC Compliance*

All components that are sold in Australia must meet ACMA (Australian Communications and Media Authority). That would require that the drone be EMC compliant as per the Radiocommunications (Electromagnetic Compatibility) Standard 2017 [25], as well as using pre-approved RF modules that follow Low Interference Potential Devices (LIPD) Class Licence 2025 [26].Having these compliances would ensure that the drone is legal to operate as per Australian electronic safety guidelines.

3.2.3 *Battery and Dangerous Goods Standards*

Due to the drone utilising a LIPO battery, there are certain rules and regulations regarding the safe transport of this type of battery. The use of this battery must comply with IATA (International Air Transport Association) Dangerous Goods Regulations [27].

3.2.4 Product Safety and Consumer Law

The product must be safe for end consumer use. This is stated in the Competition and Consumer Act 2010. Ensuring this will make sure that the product meets Australian Consumer Law Product safety guidelines [28].

3.2.5 Import and Export Controls

All components must comply with the Australian Defence and Strategic Goods List (DSGL) and Customs (Prohibited Exports) Regulations 1958. These regulations stipulate that the components cannot be controlled military-grade electronics, such as encrypted radio [29].

A production model would have to ensure these regulations are met.

3.2.6 Software and Cybersecurity

The drone must follow some key rules, called Cybersecurity Principles. They include signed and verified firmware updates, using industry-standard protocols such as WPA2 with no external internet control, fail safes for loss of connection (already included), storing sensitive data such as telemetry and position.

3.3 Tendering Considerations

3.3.1 PCB Fabrication

JLC PCB was as the key manufacturer for the PCB. The benefit JLC holds over competitors in PCB manufacturing, is that they have: short turn-around times between ordering and completion for PCB-A; access to worldwide shipping ports from their base of operations in Shanghai; and the frequent discounts they provide.

However, JLC would not be viable to use for mass manufacturing, as they deal with multiple orders from different clients, so they do not have the capacity. JLC also charges additional fees for components outside of their default library, as well as assembly fees per component, which conflicts with what the team would need for it to financially viable to manufacture. So, alternatively, a dedicated plant would have to be used for drone manufacturing.

3.3.2 Shipping

DHL express was used for shipping the PCBs. DHL was a good choice for getting the PCBs in with short notice for testing; However, this incurred an additional fee, compared to other postal companies, which offered free shipping, but with longer wait times. To implement mass shipping for the design, the team would need to do a benefits analysis to find a company that minimises cost and shipping time.

3.4 Recommended Future Tests

Based on the verification and validation outcomes in Table 2 in Section 2.1, a number of requirements were failed or partially satisfied during testing. The following future tests are recommended to close outstanding verification gaps and improve robustness of the system.

REQ-001 – Altitude Control/Hover Stability
Future testing should focus on refining the altitude controller to maintain a consistent hover height within 0.1m of the target setpoint. This can be achieved by implementing a PID and logging sensor feedback at a higher frequency to tune gain values. Controlled indoor tests with motion-capture or ultrasonic range verification recommended to quantify steady-state error.

REQ-002 – Obstacle Avoidance Stability
Further validation is required to safely re-enable automatic motor cutoff logic without risking sudden descent. A gradual thrust reduction routine or controlled hover hold should be tested to comply with the 0.5 s reaction requirement while maintaining safe stability. Simulated obstacle trials at multiple speeds should also be tested.

REQ-032 – Flight Path Deviation and Stability
Lateral deviation requirement of 0.5 m was not met. Additional experiments could include motion-tracking to compared commanded versus actual path coordinates. Refinement of odometry scaling and closed-loop feedback from optical flow sensors should be validated in repeatable indoor test courses.

REQ-014 – Battery Integration and Power Testing
Future tests should re-evaluate onboard power integration using an isolated buck-converter or battery management module to allow safe charging via USB-C. Validation campaign should confirm battery discharge profiles, sag under load and over-current protection tests should be conducted before integration into mainframe.

REQ-012 – Charging Method
USB-C charging could not be validated, Micro-USB ended up being used instead. Once charging circuit and connector redesign are implemented, electrical-safety and over-current protection tests must be conducted before integration into the airframe.

REQ-030 – Reprogramming Speed
The time required to pre-program the drone's path should be retested using an optimised interface workflow. Automating parameter uploads through the Web Server UI will allow revalidation of the 5-minute requirement in future iterations.

REQ-031 – Post- Flight Logging
Currently, data is printed only to serial monitor. Future validation should test full CSV data excerpts of timestamp logging of flight variables (position, velocity, altitude, obstacle detection) for external and regulatory compliance.

4. References

[1] M. Vlasova, “Why is GPS Ineffective Inside Buildings? GPS Alternatives for Positioning and Tracking,” 30 01 2023. [Online]. Available: https://navigine.com/blog/why-is-gps-ineffective-inside-buildings/.

[2] DJI, “DJI Tello,” [Online]. Available: https://www.d1store.com.au/categories/tello-drone. [Accessed 20 October 2025].

[3] fishpepper, “Thrust-Test: DYS BE1104 6500KV,” fishpepper.de, 3 Dec 2017. [Online]. Available: https://fishpepper.de/2017/12/03/thrust-test-dys-be1104-6500kv/. [Accessed 18 Oct 2025].

[4] N. B. Magic, “Motor Test Stand (Results: Coreless 8520 with multiple props),” Not Black Magic, 24 September 2025. [Online]. Available: https://notblackmagic.com/projects/motor-test-stand/. [Accessed 18 October 2025].

[5] E. Freaks, “Ultrasonic Ranging Module HC - SR04,” [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.

[6] T. C. Microsystems, “MOD_CH101 Datasheet,” [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2019/10/DS-000333-MOD_CH101-Datasheet-v1.1.pdf.

[7] “A new generation, long distance ranging Time-of-Flight sensor based on ST's,” 2024. [Online]. Available: https://www.st.com/resource/en/datasheet/vl53l1x.pdf.

[8] PixArt, “PMW3901MB Product Datasheet,” [Online]. Available: https://wiki.bitcraze.io/_media/projects:crazyflie2:expansionboards:pot0189-pmw3901mb-txqt-ds-r1.00-200317_20170331160807_public.pdf.

[9] STM Electronics, “STM32F03 Data Sheet,” November 2021. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.st.com/resource/en/datasheet/stm32f030f4.pdf. [Accessed October 2025].

[10] espressif, “ESP32-S3R8 Datasheet,” [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf. [Accessed October 2025].

[11] FTDI Chip, “FT232R USB UART Datasheet,” [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf. [Accessed October 2025].

[12] espressif, “ESP32 Devkit Datasheet,” 2017. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://dl.espressif.com/dl/schematics/esp32_devkitc_v4-sch.pdf. [Accessed October 2025].

[13] espressif, “ESP32-WROOM Datasheet,” [Online]. [Accessed October 2025].

[14] Silicon Labs, “USB XPress Family CP2102N Datasheet,” [Online]. [Accessed October 2025].

[15] Microchip, “MIC5219 Datasheet,” [Online]. [Accessed October 2025].

[16] Texas Instruments, “TPS6216x Datasheet,” [Online]. [Accessed October 2025].

[17] Texas Instruments, “TPS6160x Datasheet,” [Online]. [Accessed October 2025].

[18] Texas Instruments, “TPS568230 Datasheet,” [Online]. [Accessed October 2025].

[19] Engineers Australia , “Engineers Australia Code of Ethics,” August 2022. [Online]. Available: https://www.engineersaustralia.org.au/sites/default/files/2022-08/code-ethics-guidelines-professional-conduct-

2022.pdf. [Accessed 7 October 2025].

[20] UWA Aviation Labs, 2025. [Online]. Available: https://learn-ap-southeast-2-prod-fleet01-xythos.content.blackboardcdn.com/5ddb108fe0c42/33874396?X-Blackboard-S3-Bucket=learn-ap-southeast-2-prod-fleet01-xythos&X-Blackboard-Expiration=1759903200000&X-Blackboard-Signature=TfhjVhy1Ww9tiC%2Fol1OgR5VPCPr3O0.

[21] "when2meet," [Online]. Available: https://www.when2meet.com/.

[22] Dunkermotoren GmbH, "Brushed vs Brushless DC Motors: Which is Best for your Application?," Dunkermotoren GmbH, [Online]. Available: https://www.dunkermotoren.com/en/knowledge/white-papers/brushed-vs-brushless-dc-motors. [Accessed 19 October 2025].

[23] Directorate-General for Communication, "EU common charger rules: Power all your devices with a single charger," European Union, 28 December 2024. [Online]. Available: https://commission.europa.eu/news-and-media/news/eu-common-charger-rules-power-all-your-devices-single-charger-2024-12-28_en. [Accessed 15 October 2025].

[24] CASA, "Part 101 of CASR Unmanned aircraft and rockets," 14 September 2022. [Online]. Available: https://www.casa.gov.au/rules/regulatory-framework/casr/part-101-casr-unmanned-aircraft-and-rockets#Rulestatus. [Accessed 20 October 2025].

[25] epartment of Infrastructure, Transport, Regional Development, Communications, Sport and the Arts, "Radiocommunications (Electromagnetic Compatibility) Standard 2017," 05 January 2018. [Online]. Available: https://www.legislation.gov.au/F2018L00024/asmade/text.

[26] Department of Infrastructure, Transport, Regional Development, Communications, Sport and the Arts, "Radiocommunications (Low Interference Potential Devices) Class Licence 2025," 05 September 2025. [Online]. Available: https://www.legislation.gov.au/F2025L01047/asmade/text. [Accessed 20 October 2025].

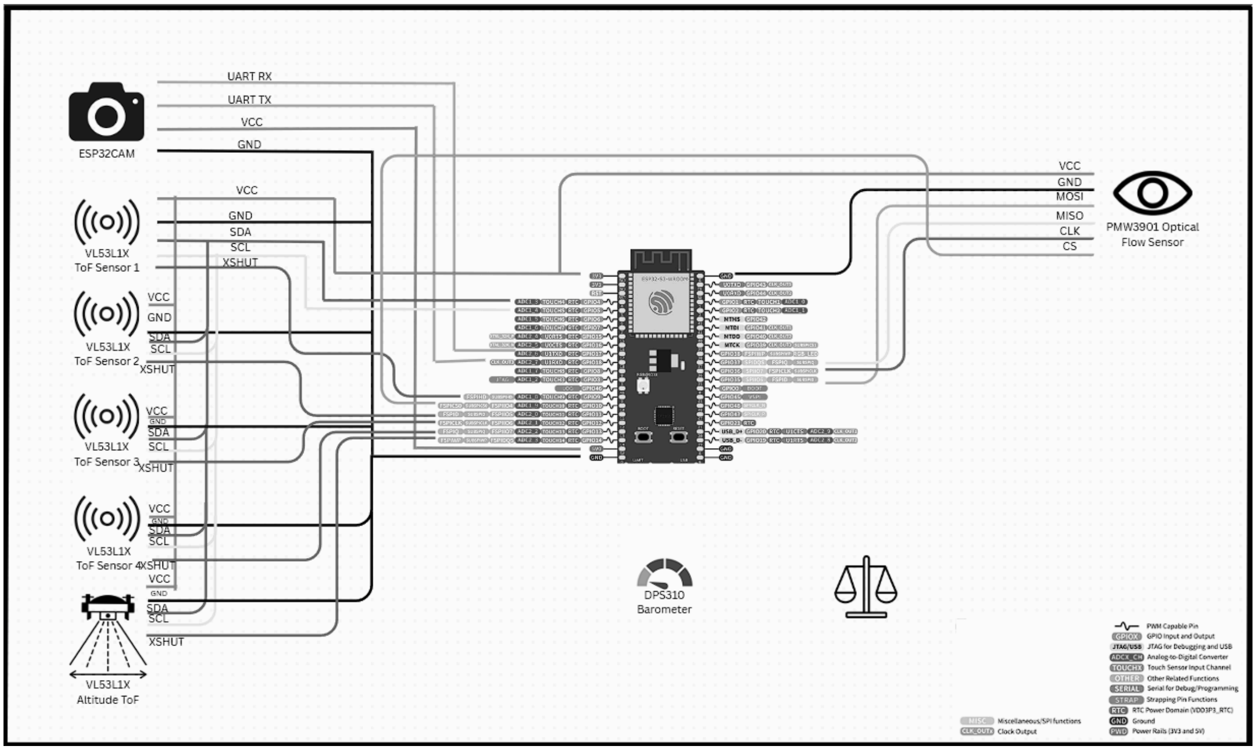
[27] IATA, "Batteries," [Online]. Available: https://www.iata.org/en/programs/cargo/dgr/lithium-batteries/. [Accessed 20 October 2025].

[28] ACCC, "Product Safety," [Online]. Available: https://www.productsafety.gov.au/. [Accessed 20 October 2025].

[29] Department of Defence, "Defence Strategic Goods List," [Online]. Available: https://www.defence.gov.au/business-industry/exporting/export-controls-framework/defence-strategic-goods-list. [Accessed 20 October 2025].

5. Appendices

5.1.1 Appendix A



Initial Wiring Diagram for Sensors and ESP32

5.1.2 Appendix B

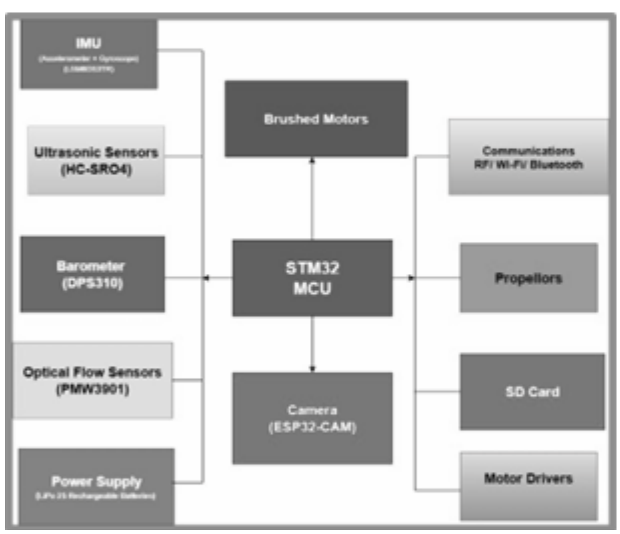


Figure 5.1.2-1: Chosen Components after Design review 1

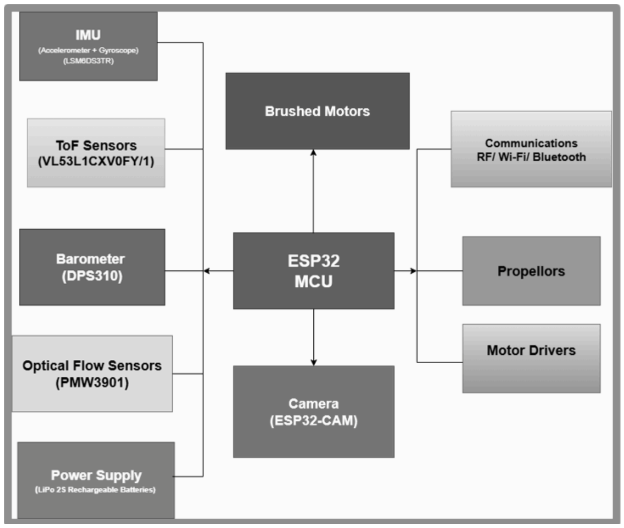


Figure 5.1.2-2: Components after Des

5.1.3 Appendix C – Testing Documentation

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Aluminium X-Frame (Torsion Box)	1	C001	20/08/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Validate structural rigidity and assess weight feasibility	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Bench torsion and weight feasibility test of the aluminium X-frame chassis. Static and dynamic load tests conducted to assess rigidity, deflection, and weight-to-thrust ratio.	Requires motor mount assembly, motor controller calibration, and power supply integration.	Indoor bench test, 25°C	Emergency cut off switch implemented

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Measure deflection under 500 g static load	20/08/25	< 5mm	3.8mm	Pass	Frame is rigid but heavy
2	Thrust-to-weight evaluation	21/08/25	Lift achievable at ≤ 60% throttle	Hover achieved, full throttle required	Fail	Excessive frame mass
3	Visual Inspection	21/08/25	No cracks or distortion	No cracks observed	Pass	Material integrity maintained

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
3D Printed ABS Box Chassis	2	C002	27/08/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Validate the suitability of ABS for the chassis	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Validation of 3D printed ABS box-style chassis for rigidity, vibration isolation, and weight balance	Functional 3D printer and ABS filament	Indoor bench and tethered hover test with full motor load	Power cutoff switch and tethered test zone

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Measure total frame mass	27/08/25	<100g	140g	Fail	Too heavy for flight
2	Hover thrust test	27/08/25	Hover at 60% throttle	Hover unstable, 85% throttle required	Fail	Weight exceeded safe thrust margin
3	Inspect motor mount	27/08/25	No delamination	Delamination at high thrust	Fail	Requires mount redesign

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Ultra-Light Flat Plate Frame	1	C003	02/09/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Test the suitability of a flat plate ABS frame	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Evaluation of ultra-light flat plate ABS chassis for weight optimization and vibration response	Mounted motor assembly, IMU sensor connection	Indoor hover test, 3.7V supply, temp 24°C	Tethered, clear perimeter maintained

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Record frame weight	02/09/25	≤20g	12g	Pass	Met weight requirements
2	Perform hover test	02/09/25	Stable hover ≤ 60% throttle	Hover at 50%, mild vibration	Pass	Thrust margin acceptable
3	IMU vibration analysis	02/09/2025	Peak vibration < 200Hz	150Hz resonance observed	Fail	Structural flex and resonance detected

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Pre-Final Integrated Fram	1	C004	19/09/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Validation of chassis	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Validation of final integrated chassis for rigidity, maintainability, and endurance under operating conditions.	Fully assembled prototype with all subsystems installed.	Indoor hover and endurance testing at room temperature	Safety perimeter, remote throttle control, visual observer.

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Hover stability test	19/09/25	Stable hover ≤ 60% throttle	No hover was recorded	Fail	Voltage sag prevented stable hover
2	Motor endurance run (3 min)	19/09/25	No detachment or vibration drift	No detachment, acceptable drift	Pass	Vibration mitigated, secure mounts
3	Drop test from 1m height	19/09/25	No cracks or deformation	No visible damage	Pass	Final design sturdy but needs power resolved

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
PID Hover Control	1	M001	N/A

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Test PID and sensor feedback	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Tested closed-loop altitude PID controller using ToF feedback to maintain hover height.	ToF altimeter and PWM motor drive active.	Indoor hover at 0.6 m; tethered.	Throttle limited to 70%; emergency cutoff switch.

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Set PID (P=1.2 I=0.4 D=0.1)	N/A	Stable ±30 mm	N/A	N/A	
2	Apply 10 cm disturbance	N/A	Recover < 1 s	N/A	N/A	

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Motor Thrust and Cut-off Validation	1	M002	12/10/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Test bench motor assessment	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Measured thrust curve and verified firmware cutoff on over-current or stall conditions.	MOSFET PWM drivers and regulated supply.	Single motor bench test @ 3.7 V.	Manual cutoff switch; scope current probe.

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	PWM ramp 0–100 %	12/10/25	Smooth Rise	Smooth	Pass	No spikes
2	Force Stall	12/10/25	Stop <200ms	150ms	Pass	Firmware cutoff ok

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
System Shutdown Failsafe	1	M003	15/09/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Calibrate Safety Shutdown	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Verified automatic motor shutdown on controller disconnect	Active watchdog and sensor status monitor.	Bench test	Web sever feedback

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Disconnect mobile device while motor running	15/09/25	Motor stops in <1s	Motor stopped as expected	Pass	Motor is safe

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
PCB 2nd iteration motor test	1	M004	16/10/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Single motor test with 2nd iteration pcb power circuit	Josh Wong	Josh Wong

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Run motor via integrated pcb motor controller circuit	PCB motor controller circuitry	Bench test using 2s battery	Safety shut-down switch, temperature imaging device to view hot spots

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Connect battery to pcb	16/10/25	All 4 power leds illuminate	Confirmed 4 leds	Pass	No load circuit works
2	Load pwm code to esp	16/10/25	UART serial connection successful, code loaded with no errors	Code upload successful	Pass	UART serial connection works
3	Initiate PWM	16/10/25	Motor runs successfully	Hot spot found in power circuitry, blown inductor	Fail	PCB motor power circuit not capable of motor load

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Optical Flow Module Verification	1	S001	09/10/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Optical flow module tested for accuracy	Daniel Papasergio	Daniel Papasergio

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Validated SPI communication and optical flow data stability of the PMW3901 sensor on the ESP32-S3 platform. Motion counts were recorded at rest and in translation to evaluate sensitivity to surface pattern and height.	Functional SPI configuration, power supply regulation, and stable surface with sufficient visual texture.	Indoor bench test, consistent lighting, reflective surface at ~200 mm height.	Serial data logging; 3.7 V USB supply; emergency stop available.

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Initialize SPI and verify device ID (0x49)	09/10/25	Device responds correctly	Response confirmed	Pass	Stable at 1 MHz SPI
2	Hold sensor stationary for 5 s	09/10/25	Δx, $\Delta y \approx 0$	±2 counts noise	Pass	Low-noise baseline
3	Translate board 100 mm laterally	09/10/25	Linear count increase	Counts proportional to motion	Pass	Consistent Response

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
ToF (VL53L0X) Single Sensor Verification	1	S002	10/10/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Test Reliability of ToF Sensor	Daniel Papasergio	Daniel Papasergio

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Confirmed reliable operation of one VL53L0X sensor for forward obstacle detection. Measured distance accuracy and response stability.	Configured I²C interface; address 0x29 active.	Indoor, 25 °C, flat white target at 200 mm.	

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Initialise Sensor	10/10/25	Init() works	Confirmed working	Pass	Stable Startup
2	Read distance to wall	10/10/25	200±10mm	197mm avg	Pass	Within tolerance
3	Hold for 60s	10/10/25	±20mm variance	Max measured variance was ±15mm	Pass	Consistent Reading

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Multiple VL53L0X Address Configuration	1	S003	11/10/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Test connection of multiple ToF sensors	Daniel Papasergio	Daniel Papasergio

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Validated simultaneous operation of three VL53L0X modules with unique I²C addresses (0x30/0x31/0x32) using manual re-addressing and round-robin polling.	Independent XSHUT lines wired to GPIO pins	ESP32-S3 at 3.3 V logic; sensors mounted front/left/right.	I²C clock 400 kHz after setup; error trap active.

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Sequential power-up and address set	11/10/25	0x29→0x30/31/32	All Assigned	Pass	No bus collision
2	Poll each sensor	11/10/25	All return valid range	Consistent ±10mm	Pass	Stable loop

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Odometry Integration (Flow + Altitude)	1	S004	11/10/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Combine ToF and Optical Flow sensors	Daniel Papasergio	Daniel Papasergio

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Combined optical-flow (Δx Δy) and ToF altitude data to estimate planar motion and velocity in real time.	PMW3901 and VL53L1X active simultaneously.	Hover 0.5 m over surface; yaw held constant.	Data logged to serial 100 Hz.

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Run odometry at 0.5m	11/10/25	Drift < 5mm/s	4.3mm/s	Pass	Stable Output
2	Yaw 45° rotation	11/10/25	Correct axis transform	Rotation matrix OK	Pass	Verified

Drone Component Test Document

GROUP 2-02

Optical Flow Calibration vs Height	2	S005	13/10/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY
Test and compare calibrations	Daniel Papasergio	Daniel Papasergio

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Calibrated PMW3901 pixel-to-radian scaling against measured height to improve velocity accuracy.	Flow sensor and ToF altitude operational.	Surface height 0.2–1.0 m in steps.	Serial logging at 100 Hz.

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Record Δx Δy vs distance	13/10/25	Linear Scaling	Linear	Pass	Good Fit
2	Apply Calibration	13-10-25	Error < 10 %	6%	Pass	Constants Stored

GROUP 2-02

Drone Component Test Document

TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
ESP-CAM Wifi Test	3	V001	10/09/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY

Confirm Wifi Test	Nehan Cripps	Nehan Cripps

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Initiate wifi as AP mode and test that mobile device connects	Wifi library enabled	Mobile device nearby with no clear signal interference	Mobile device as close and realisable

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Initiate wifi access point	10/09/25	Wifi initiates successfully, printed in serial monitor	Wifi initiated	Pass	Wifi reliable
2	Connect on mobile device	10/09/25	Device connects to wifi access point	Wifi connected successfully	Pass	Wifi connection good

GROUP 2-02

Drone Component Test Document

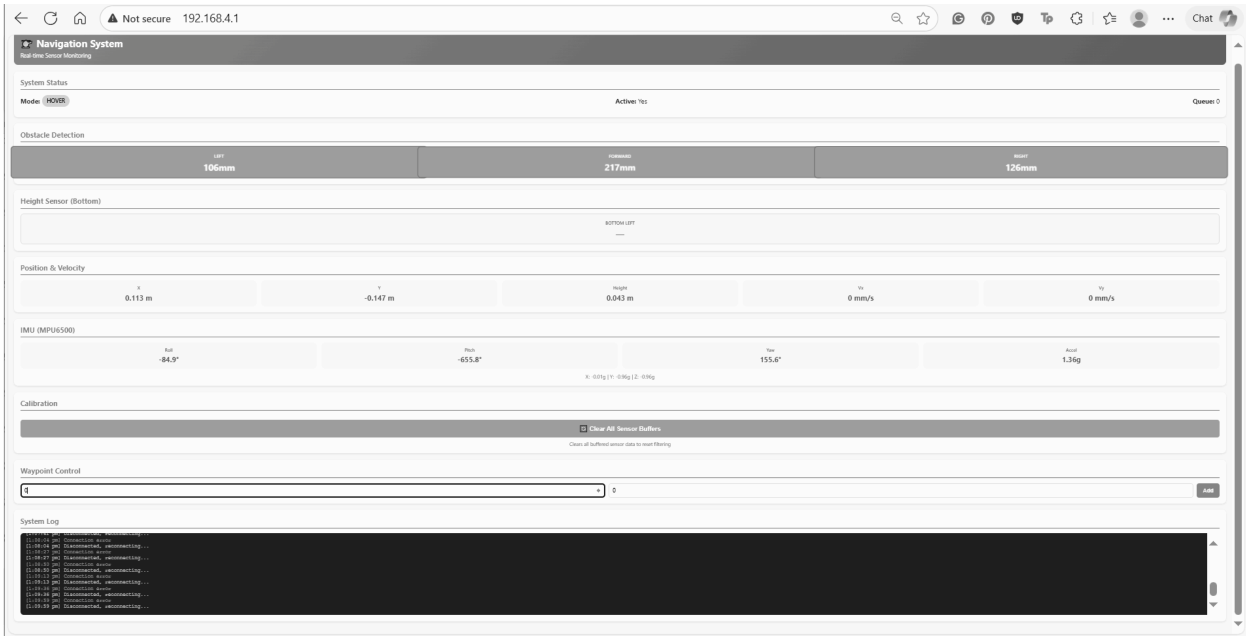
TEST TITLE	PRIORITY	TEST CASE ID	TEST DATE
Video Web server	3	V002	23/09/25

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY

Test the video web server at various resolutions	Nehan Cripps	Nehan Cripps

TEST PROCESS SUMMARY	TEST DEPENDENCIES	TEST CONDITIONS	TEST CONTROL
Test the video quality at various resolutions	Functioning wifi (see test V001)	OV3600 Camera module	Clean connection for web server

STEP ID	STEP DESCRIPTION	TEST DATE	EXPECTED RESULTS	ACTUAL RESULTS	PASS / FAIL	ADDITIONAL NOTES
1	Stream video link at 120p	23/09/25	>25 fps	30-40 fps	Pass	
2	Stream video link at 240p	23/09/25	>20fps	30 fps	Pass	
3	Stream video link at 360p	23/09/25	>15 fps	20 fps	Pass	
4	Stream video link at 480p	23/09/25	>10 fps	10-15 fps	Pass	Requires strong 5v power


Not secure 192.168.4.1
Chat
Navigation System
Real-time Sensor Monitoring
System Status
Mode: HOVER
Active: Yes
Queue: 0
Obstacle Detection
LEFT
106mm
FORWARD
217mm
RIGHT
126mm
Height Sensor (Bottom)
BOTTOM LEFT
—
Position & Velocity
X
0.113 m
Y
-0.147 m
Height
0.043 m
Vx
0 mm/s
Vy
0 mm/s
IMU (MPU6500)
Roll
-84.9°
Pitch
-655.8°
Yaw
155.6°
Accel
1.36g
X: -0.01g | Y: -0.96g | Z: -0.96g
Calibration
Clear All Sensor Buffers
Clears all buffered sensor data to reset filtering
Waypoint Control
0
0
Add
System Log
[1:08:04 pm] Connection error
[1:08:04 pm] Disconnected, reconnecting...
[1:08:27 pm] Connection error
[1:08:27 pm] Disconnected, reconnecting...
[1:08:50 pm] Connection error
[1:08:50 pm] Disconnected, reconnecting...
[1:09:13 pm] Connection error
[1:09:13 pm] Disconnected, reconnecting...
[1:09:36 pm] Connection error
[1:09:36 pm] Disconnected, reconnecting...
[1:09:59 pm] Connection error
[1:09:59 pm] Disconnected, reconnecting...