AVDASI2 2024-25
A2 Technical Report

BlueBird Company
Team 18 - Avionics

1. Contents

Contents

Contents

1. Executive Summary

During the project, the BlueBird company was contracted to design, build and test a UAV with maximised flight endurance and the ability to execute landings on short distances. Within this context, Team 18 was tasked with the design and manufacture of the electrical subsystems of the drone, developing the software and overseeing the installation and calibration of sensors and control surface actuators.

Firstly, the team undertook a rigorous down-selection process to aid with selection appropriate software framework and accurate sensing solutions for the flaps. To do so, a myriad of down-selection tools were deployed, aided by the extensive prior experience of the team members. The process is detailed in the scope of the sensor down selection, which yielded magnetic angle sensors as the best choice for this application, a novel approach in the history of the project and the first to ever be successfully implemented in this unit.

Because of the space limitations and request from wing teams, the team has designed and manufactured several bespoke PCBs that handled flap angle sensing tasks. The sensor circuits were built with industry best practice methods and offered dependable and accurate data as well as making integration relatively easy. The Harness Distributor ensured the safety of critical core electronics and reduced the risk of systems being connected in the wrong order.

The software developed by the team performed flawlessly during testing, requiring no intervention while in the wind tunnel. The UAV was able to seamlessly transition from ground station control to radio control seamlessly and the flight stability assist was effective at ensuring level flight.

During the benchtop tests, sensor performance was better than expected, allowing the accurate deflection angle of the flaps to be measured to one degree. Wind tunnel tests shown deterioration of these results due to the harsher operating conditions. Several future improvements have been discussed that could improve the design in following iterations.

The document also details the aerodynamic and structural performance of the UAV and discusses reasons as to why the system performed as such.

Additionally, team 18’s approach to project and risk management are laid out in full. These sections offer deep and critical analysis into the thought process behind critical decisions and how time was managed across such a vast project.

Finally, a series of appendices are appended in the appendix for the readers’ enjoyment. This aims to offer further insights into all the sections above and can be easily accessed by scrolling down when an appendix is mentioned.

2. Group Information

Company B Avionics

Name	Intra-team roles	Report Sections Overseen
Bryn Fergusson (PM)	Hardware Manager	-3.3 Manufacture -4 Structures -6.1 Hardware project management
Ethan Sheehan	Avionics Test Lead	-3.4 Integration -5 Aerodynamics
Lucas Dick	Executive Software Director	-3.2 Sub assembly design -6.2 Software project management
Dohyoon Kwon	Quality Assurance Supervisor	-7 Project risk assessment
Sohaib Zubair	Calibration Coordinator	-2 Group information -BOM
Máté (DTM/TD)	Angle Sensor Lead Systems Architect	-3.1 Down selection -3.2 Sub assembly design

The Team Was Responsible For:

- Designing and Developing the UAV's Avionics System – Creating a robust avionics system to support flight operations and mission objectives.
- Ensuring Compliance with Avionics Requirements – Met all technical and functional requirements, including power management, sensor integration, and data acquisition.
- Testing and Validation – Conducted thorough testing of avionics components, including sensor calibration, power regulation, and system reliability, to ensure safe and efficient operation.

The Team Contributed By:

- Avionics System Integration – Designed and implemented a functional avionics system, ensuring compatibility with the UAV's overall architecture.
- Graphical User Interface – Programmed custom user interface for real-time telemetry display, Wi-Fi based arming and seamless UAV integration
- Sensor Data Acquisition and Logging – Integrated Hall effect angle sensor for flap angle measurement and pressure acquisition system, with data processing and storage handled by the Cubepilot.
- Custom PCB Design and Power Management – Developed a custom PCBs for Hall effect sensor and Power Distributor.
- Calibration and Troubleshooting – Conducted servo calibration and angle sensor accuracy testing, refining performance for improved UAV stability and control.
- Collaboration Across Teams – Worked closely with other sub-teams to ensure avionics integration aligned with the UAV's design and mission requirements.

Report responsibilities	
Responsible person	Section
Bryn	-3.5 Manufacture -4 Structures -6.1 Hardware project management
Ethan	-3.7 Integration -5 Aerodynamics -1 Executive Summary
Lucas	-3.2 Sub assembly design -6.2 Software project management
Dohyoon	-7 Project risk assessment
Sohaib	-2 Group information -BOM
Máté	-3.1 Down selection -3.2 Sub assembly design -1 Executive Summary

3. Design and Manufacturing Review

This section aims to introduce the design, manufacturing, and integration efforts, highlighting and justifying specific technical decisions that have been made through the course of the project. Additionally, this section will provided a review of the sub-assembly performance and commenting on possible future improvements.

3.1. Down selection

To satisfy the requirements of the customers, the Avionics team of BlueBird carried out a detailed down-selection process to identify a sensor solution that would be able to measure flap angle deflection independently from servo output on both wings of the UAV.

3.1.1. Angle Sensor: preliminary-Selection

Before beginning the down selection process, the team looked at industrial standards for similar applications and identified a set of possible solutions. This preliminary pool included *rotary encoders (both absolute and incremental), potentiometers, hall effect sensors and gyroscopes.* Afterwards, each member was given at least two of the above-mentioned implementations to research further.

3.1.2. Angle Sensor: Identifying the down selection criteria

While sensor research was underway, the team met to determine the criteria that was to be used in the later part of the procedure. The way this was facilitated was a two-part discussion.

After a small roundtable, each idea was voted on with a simple majority basis. This way the team managed to reduce the number down selection criteria to just six: *resolution, durability, cost, ease of integration, refresh rate and accuracy.*

3.1.3. Angle Sensor: Applying down selection methods learned

In the following part the usage of the provided down-selection methods that help with quantifying the strengths and weaknesses of the proposed designs will be discussed.

Having found the criteria to be used in the process, the team needed to determine how they would be weighed. For this reason, two rounds (simple and numerical) of Pairwise comparisons were carried out. The Simple Pairwise method aimed to determine the relative importance of each criterion. Afterwards the Numerical version was used to quantify the said relative importance. Figure 18 shows the results yielded by the Numerical Pairwise comparison. It can be seen that the most important benchmarks have been found to be *accuracy, resolution and refresh rate*.

The team then went ahead with a Controlled Convergence test. In this part of the process each proposed sensor was compared to a chosen ‘basis’ sensor. The basis was a potentiometer due to its simplicity and large variety of uses. This test was found not to be really useful as it only compares the basis to everything else.

The final test method deployed was a multi-criteria decision analysis (MCDA). Every sensor was placed into the matrix and given a 1 to 5 score (5 being best) for each criterion. The benchmarks were weighted using the values found in the Numerical Pairwise comparison, as seen in Figure 18. After evaluating the performance of each potential sensor design the MCDA matrix outputted the numerical score of each sensor, as can be seen in Figure 19 below.

The magnetic Hall effect sensor had a best score, and after a brief final discussion, the team proceeded to develop it.

3.1.4. Angle Sensor: Comments on the down selection method

This part of the document aims to comment on the methods used during process of the down selection itself.

While the team did its best to ensure an objective and smooth down selection, as with everything, there were challenges. Starting with the preliminary research of the chosen sensors. Given that each member was responsible for studying some of the sensors, it is not unexpected for them to develop some sort of bias towards/against their own design. Fortunately, the team was able to step past this and evaluate each proposed design with high level of objectivity.

The team had members that were very experienced is some of the topics encountered during the project. This came with the natural tendency for some members to be more confident in their way, inadvertently influencing others to be more susceptible to their choices. It is hard to quantify the extent the team encountered this issue, but in retrospect the final choice was successful, and the quality of the team’s work was not compromised.

Finally, many of the criteria that were used in the down selection were hard to quantify. As such, these tools can help the down selection, but blindly trusting them could lead to issues. They are also easily influenced to validate a predetermined outcome. To combat this, the team made sure to discuss every result that was achieved with the tools, but one can't outright reject the idea that some of the inputs were subconsciously skewed to favour one output over the others. However, the end results, to some extent, validate the down selection. As can be seen in the later parts of this document the design performed exceptionally and as such the process is considered a stunning success.

3.1.5. Software: Down Selection

As with the angle sensor, the process of designing the software began with down-selecting the framework. Pymavlink was the only choice considered for the TMS due to its comprehensive documentation. The choice then had to be made to either interface with the TMS in a separate UI program, or choose the simpler, yet less configurable option of using a Python UI library. After a brief discussion, creating a separate user interface in React/TypeScript was chosen, primarily due to prior experience of the team members. Further, as it is a type-safe language, it helps to avoid bugs at compile time [14], mitigating R25. This choice allowed for a more configurable and aesthetically pleasing design, as well as better performance of complex elements such as the artificial horizon. However, this came at the cost of spending more time integrating the two telemetry and UI systems, via WebSockets. With hindsight, whilst the full stack developed was successful, the team could have experimented with different Mavlink libraries and checked their feasibility. Python was easy to use, but libraries in other languages such as Java, Rust, or C++ may have had better performance and would be easier to integrate with a performant and customisable UI, without requiring a web browser to run.

3.2. Sub-Assembly Initial Design

This section of the technical report will outline and reflect upon the avionics sub assembly, highlighting specific design choices that were taken and providing an overview of the performance of the final assembly.

As avionics engineers, the team's role was to design all the hardware and software operating on the UAV. Thus the sub-assembly design was essentially split into two parts; hardware design and software design, with the two designs integrating with one another through the Flight Control System, a CubePilot Orange Plus.

3.2.1. Software sub-assembly

Requirement 2.2.1. identified four main software systems: a flight control system (FCS), a radio control system (RCS), a telemetry system (TMS) and a ground control station (GCS).

The flight control system's primary function is to command control inputs onto the control surfaces, and collect data from the whole UAV. The control inputs could be either manually input (via the TMS or RCS), or the UAV could enter "stabilise mode" and automatically stabilise the itself under turbulence. This was handled by a CubePilot Orange Plus, using the Ardupilot autopilot system.

The radio control system's function was to interface with the FCS using a hand-held radio controller. Its primary role was to provide manual control of the flight surfaces, as well as being a back-up in case of TMS failure.

The telemetry system handles all non-radio controlled communication between the FCU and the GCS. The FCU had a telemetry system integrated into the CubePilot, the MAVLINK protocol. Using an ESP32 WiFi chip, the FCU would periodically send out data messages to the ground control station, through a bidirectional UDP (WiFi) connection. The ground station could then send commands back the FCU. On the ground station, this telemetry system was implemented through a python script running the Pymavlink library.

The ground control system provided a way of interacting with the UAV in real time. The main component of this was a user interface (UI). The UI allows an operator to view critical information pertaining the UAV's current state, as well as send customised commands to the UAV. The UI was written with TypeScript and the React framework. It interfaced with the TMS through the WebSocket protocol.

The Figure 20 demonstrates how each of the systems work and interface with each other.

3.2.2. Angle Sensor sub-assembly

The Angle Sensor sub-assembly consisted of two parts: a stationary magnet and a Hall effect sensor.

The Hall effect sensor, as its name suggests, utilizes the Hall effect: a conductor will experience a potential difference across its sides when it is transverse to an electric field and perpendicular to a magnetic field. As such the sensor, by

measuring this potential difference in multiple directions can determine the direction of the magnetic field (since the electrical field is kept stationary).

3.2.3. Harness Distributor sub-assembly

The Harness Distributor was a PCB with no active components, taking in power from the power modules, and data to and from the cube and splitting it out into separate whole harnesses for each servo and sensor. The initial driver behind this was to ensure that strain on the wires would not damage the locking GH headers on the cube pilot. It also provided the benefits of segmentation between the avionics bay and the rest of the UAV, with a single place to repatch harnesses, as well as external power routing to servos which would protect the cube from a current overload in the case of large power draw from the servos or sensors.

3.3. Sub-Assembly Final Design

3.3.1. Software Final Design

As in section 3.3.1, the software was split into a Telemetry system, written in Python and using the Pymavlink library, and a user interface, written in React and TypeScript. Extra RC functionality and angle sensor telemetry was designed using scripts written in Lua running onboard the CubePilot. Source control was carried out using a GitHub repository.

Throughout the software development, there were three main principles in mind: type checking, configurability and readability. Readability was the most important aspect of design. Since there were multiple people working on the code simultaneously, it was important that everybody could understand each other's code. In the TMS, it made sense to use object orientated programming, and in the UI, functional components were used, following React best practices.

Type checking ensures that each datatype, whether it be floats representing pitch and roll, or strings containing logging messages, is passed accurately between the different systems, particularly the TMS and UI. TypeScript was a particularly good choice for this, as it natively implements type checking. Python on the other hand only has type hinting, so mistakes could be caught during development, but weren't caught whilst running. This helps obviate bugs [14] , and was a good mitigation as per R25. Configurability was primarily done using enumerated types; similar to variables, but easily changeable and clearly readable.

3.3.2. Telemetry System (TMS) Final Design

The telemetry system connects to two separate systems; the CubePilot over UDP, and the UI over WebSockets. The main feature of the TMS design was that it could connect to each of the systems independently (R19). That is to say, if the UI crashes, telemetry is maintained with the CubePilot, and if telemetry is lost with the CubePilot, the UI maintained its functionality. The code was split into four separate classes:

1) CubeConnection class – the main class, handled telemetry with the CubePilot, and the UI, passing data between the two systems. Contained the main program loop (implemented asynchronously) that checked for new messages from the CubePilot.
2) Logger class – output telemetry data to a CSV file (2.2.1.g, 3.3.3.a, 3.3.3.b)
3) CubeOrangeServo – a servo object with certain attributes such as min and max pulse width modulation (PWM) values, and methods such as `pwmToAngle()`, converting PWM input into an output angle as per the servo calibration.
4) ServoConfiguration – for each servo, create an instance of CubeOrangeServo. When the UI tries to move a servo, this class had a `sendAngle()` method that would send the new angle request to the Cube, as well as automatically write servo parameters to the FCU on load.

3.3.3. User Interface (UI) Final Design

In order to maximise readability, each component of the UI, and each type of request had its own corresponding file. HTML Canvas elements were used for gauges and the artificial horizon, due to ease of development. Live data plotting was carried out using the Recharts.js library. Buttons and sliders used were designed with Material UI.

On program load, an object containing the FCU's latest outputs, `latestData` was initialised. Every time a WebSocket message was received from the TMS, the `latestData` object would be updated. The UI would pass the `latestData` object into the components at $5\ Hz$ (3.3.3.b), and the components would then reload with this new data. The UI had the following gauges and controls, as per Table 1.

Table 1: UI Features

Gauges	Controls
Pitch and roll (3.3.3.a)	Autopilot mode switching (3.3.4.b)
Artificial horizon	Arming and safety switch buttons (R21)
Displaying and graphing the live servo outputs (3.3.3.a)	Manual servo deflection angles (2.4.3, 3.3.1)
CubePilot message logs	Flap angle outputs (2.4.3) with specified configurations
Flap sensor position (3.3.3.a)	

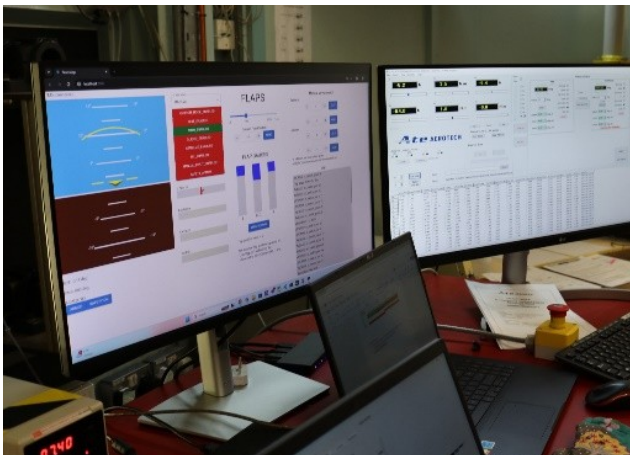


Figure 1, UI in operation

3.3.4. Sensor Software Final Design

The sensor chips used the I2C protocol to communicate with the Cube. The real challenge involved setting up the software that would control the sensor, receive the angle data and transmit it over mavlink to the GCS. Due to the way the Cube architecture is built, there were only two ways of achieving this: writing a custom sensor firmware, or use onboard scripting features built into the flight controller.

The custom firmware allows for greater customization and more robust code but would have been overkill for the comparatively simple task the code had to fulfil. On the other hand, the custom scripting is more accessible however, it also came with several downsides. The coding language the Cube uses is the LUA language, a lightweight, high-level programming language that is usually used in similar embedded applications in industry. This was entirely new for everyone on the team, so the people working with it had to learn a completely new coding language. To make matters worse the LUA environment in the Cube was a severely restricted version of LUA, meaning that many, otherwise core, functionalities were inaccessible. Debugging and installing code was also challenging as every change required a full hard reset of the cube, making the development slow. This also meant that if the code goes offline, the sensor is lost for the duration of the flight, until it can be serviced.

The final code managed to circumvent these issues and was able to offer extensive functionality: sensor and logging state control (both the sensor and the onboard logging could be turned on/off separately from the UI), zeroing (again, from the UI) and dynamic log file naming. The code would periodically request the raw angle from the chip using the I2C bus, perform bitwise operations to extract the necessary bytes from the data and translate that into actual angle measurements. It would then send the data to the GCS via mavlink in a `named_value_float` type message. The code was highly robust, with appropriate error handling to ensure it would not crash in extreme scenarios.

3.3.5. FCU Code Final Design

The FCU had inbuilt functionality to meet all the control system requirements, via its AUTOTUNE flight mode. Several Lua scripts also ran onboard the FCU, allowing for autopilot mode switching to be carried out using RC buttons (3.3.4.b).

3.3.6. Angle Sensor Final Design

The final version of the sensor design incorporated the A1335 chip, which was found to be much more accurate than the earlier AS5600. The new integrated circuit chip also had several quality-of-life functions, like changeable I2C address, magnetic field strength measurement and programmable linearisation methods.

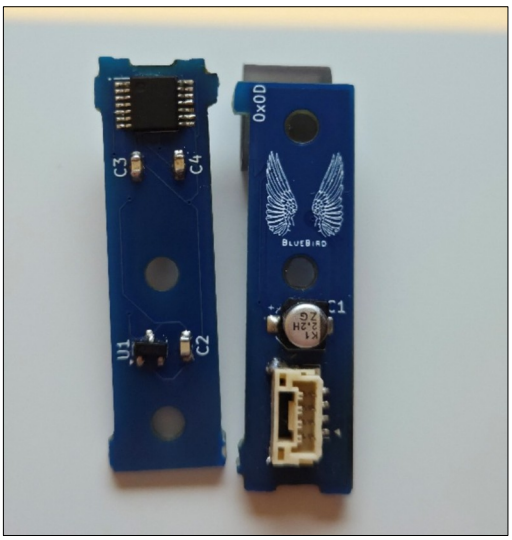


Figure 2 shows the final sensor PCB. A bespoke circuit board allowed the team to work with the limited space available in the wings. The design team has made some specific choices during the development of the sensor that will be elaborated here.

Firstly, the inclusion of a *Low-Dropout Voltage (LDO) Linear Voltage Regulator* (labelled U1 in Figure 2). The sensor chip is a delicate instrument, requiring a stable DC input voltage of between 4.5 and 5 volts. The Voltage Regulator on the sensor ensures the sensor will always operate in favourable electrical conditions and allows for a large range of input voltages, making the sensor more versatile.

Figure 2 Picture of the sensor PCB

Secondly, the usage of *Decoupling Capacitors*. These (labelled C2 to C4 in Figure 2), drawing upon the behaviour of capacitors resisting the change of voltage, add further protection to against fluctuations in voltage. (Appendix 1)

Thirdly, a Ground Plane. This is a large section of neutral copper on the underside of the PCB. This has several positive effects on the operation of the sensor. It makes routing easier making the copper traces less cluttered; it also helps signal integrity within the clock and data lines of the sensor; and finally, it helps heat dissipation.

3.3.7. Harness Distributor Final Design

The final Avionics Bay assembly featured another bespoke PCB: the Harness Distributor. This component was designed to manage outbound wires from the cube, ensuring seamless interfacing with the subsystems and the AvBay. Its implementation brought significant advantages to the avionics systems, including reduced mechanical stress on the Cube and minimized risk of damage. Power was routed to the Additionally, PWM signals for the servos were routed through the distributor, offering the Cube protection against overcurrent. The board also provided multiple I2C output lines from a singular input, enabling multiple sensors to connect to the data bus efficiently. Servo outputs were clearly labelled on the silkscreen, making it easy to connect each servo to its designated location which sped up assembly times. Diagrams can be seen in Appendix 1.

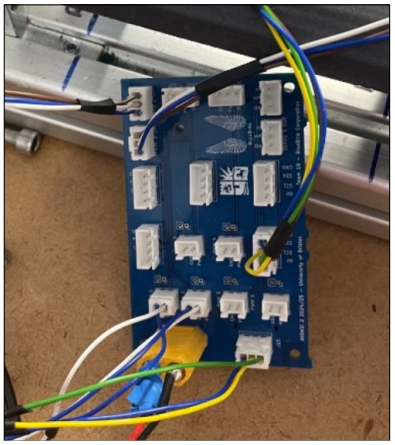


Figure 3 Picture of the Harness Distributor

3.4. Review of sub-assembly performance & opportunities of improvement

3.4.1. Software Performance

During the wind tunnel test, the software performed just as expected, meeting the requirements without any major errors. After the first two wind tunnel tests proved incredibly successful, the UI was updated with extra logging and graphing capabilities. This update did not impact the FCS in any way. That said, some extra small graphics changes (e.g. changing the colour of a gauge) were made during the test.

Other than this, there were a few areas of improvement that were known of before the test that would have been ideal to have fixed. Firstly, the UI was prone to lagging after a few minutes. This bug was not strictly reproducible. Time was spent trying to minimise memory usage, the usual cause of this sort of lag. Refreshing the page would usually fix the issue. During the wind tunnel test, this bug did appear, yet as was the case during development, the page was simply refreshed once the lag became noticeable. The best fix would have been not using a web app for the UI, as web applications in React are notoriously prone to poor memory management [14].

A further area of improvement was implementing logic onboard the FCS in case of simultaneous GCS and RCS failure. Whilst it was presumed ArduPilot would handle this automatically, and it wasn't noticed as a potential issue during development. In retrospect, the FCS' logic should have been double checked in case of these systems failing. This may not have explicitly been a requirement, but was something potentially overlooked in the risk assessment.

As well as this, the autopilot mode display on the UI was not functional. Whilst not an explicit requirement, and the autopilot modes were still controllable, it was certainly a useful indicator of the FCS' performance, which could have been used during the auto-tune phases of testing. Ultimately, the OTS Mission Planner software was used for this purpose.

During testing and full integration, there were two novel errors discovered during the test. Firstly, during the final test, the GCS, TMS and PAS code was ran on the same laptop using Visual Studio Code (VSCode). Yet, the team was not aware that pressing the 'run' button on VSCode would kill whatever process was already running in the VSCode terminal. This inadvertently stopped the UI process. The easy fix, and a feature that should have been implemented before the test, would have been to create a script that automatically starts all the processes together, at once. Such "container" would have made operation of the GCS far easier for people unfamiliar with the system.

Once the whole UAV was assembled, and connected to the CubePilot, the aileron output wasn't differential. This problem was swiftly fixed by changing parameters onboard the FCS. Whilst it was assumed the FCS would automatically configure itself for differential aileron output, this error introduced no risk to the wind tunnel test, as the servo was manually deflected to each angle through the UI, and differential aileron output would not have made a difference.

Finally, some minor aesthetic changes were made during the wind tunnel test; such as changing the colour of the gauge. Whilst it posed no risk to the UAV or tunnel, in retrospect such tweaks should not be made during testing in principle.

3.4.2. Angle Sensor Performance

The flap angle sensor performance was compared to several requirements set by the customers. Requirement *2.4.3.e* demanded flap position sensing independent of servo motors. Whilst there was no explicit requirement for the sensor's accuracy, a target of 1 degree was imposed due to requirement ***3.3.2.c***.

The first requirement was met by the sensor by design. The magnet was placed on the flap spars in both wings and the sensor therefore measured the true rotation of the flap, independent from the servo.

Regarding accuracy, during benchtop testing and in Gate 4 the sensors generally achieved the demanded 1-degree accuracy. Unfortunately, accuracy was degraded to 1-3 degrees in the wind tunnel test. This was attributed to a few factors:

Firstly, the wing experienced excessive vibrations during test. This affected the Hall effect chip by pushing the magnet off-axis. Incorporating damping into the wing or by using stronger magnets, could have mitigated this so that even if the magnet becomes off-axis the impact is negligible.

Secondly, the magnet used in the final build was not the one originally planned. Neodymium magnets have been ordered for the wind-tunnel test, as these were specifically designed for sensing applications. However, due to shipping delays, regular ferrous magnets with weaker strength needed to be installed because of the incoming deadlines. These made the effects of the vibration worse (as mentioned in the earlier paragraph). In the future proper time must be allocated for ordering components to ensure everything arrives before the final deadline.

3.5. Manufacture

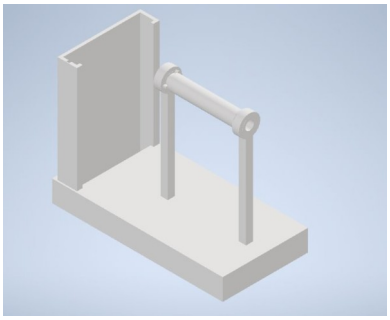
The manufacturing of hardware saw three main divisions: PCBs, Harnesses and Fittings.

PCBs were ordered unassembled from JLCPCB halfway through the manufacturing phase. Whilst this was originally planned earlier to be ready for assembly at that time, a combination of extra design validation time and communication. The PCBs were then hand soldered using irons and personal hot air rework stations with electronics components from Farnell. In the assembly of the first PCBs, issues were encountered with the heat charring and warping of the plastic header sockets. However, those manufacturing found ways to mitigate these issues in follow up attempts on the spare PCBs, primarily through reduction in exposure time possible due to developed skill. Ultimately whilst the delayed orders did introduce some danger to the on-time completion of the subassembly, it was primarily within the company designated overflow times and they were delivered working before their post delay critical dates.

To provide strain relief harnesses were zip tied to structural elements. Segmentation, originally planned via failed socket PCBs, was enabled by epoxy bonded JST XH male to female connectors which although an improvisation unsuitable for commercial deployment, improved the electrical connectivity whilst maintaining suitable mechanical strength. Possible alternatives to zip ties could consist of cable trays and screwed rubber grips for a commercial deployment iteration or, more suitably for the vehicles prototyping purpose: harness lacing, as advised by the Australian Civil Aviation Safety Authority [3], would be a more reliable and maintainable solution.

Fittings, again while planned in advance had to be somewhat improvised due to the lack of requested mounting systems provided to the avionics. For the sensors these required bonding of nylon bolt standoffs to plywood and carving of foam. The most significant fitting, the avionics bay was designed and 3D printed after initial integration attempts, in which the Pod's provided avionics bay consisting of fixed aluminium ceiling deck was found to be impractical to mount avionics in due to the enclosed space reducing dexterity. After those attempts the Pod's requirements were re-examined and the new avionics bay was made to bolt and clamp into the main payload decking.

3.6. Angle Sensor testing rig



In order to investigate the accuracy of the hall effect sensor, a testing rig was designed. The setup included a breadboard-mounted sensor prototype and an axle that held a spinning magnet at an adjustable distance from the sensor. This flexibility enabled an examination into the ideal sensing range. A protractor was attached to the rear of the axle, allowing for exact measurements of the sensor's accuracy at various magnet-to-sensor distances. Collars were also added in the rig to prevent the axle from moving along its support.

3.7. Integration

Integration of the Avionics with the rest of the UAV proved to be the most challenging aspect of the project. Due to slippage in other parts of the company, many parts of the integration did not occur as planned. Nevertheless, it all came together for a flawless wind tunnel test campaign. This analysis highlights how the company could have benefitted from more communication and advanced planning for the integration phase.

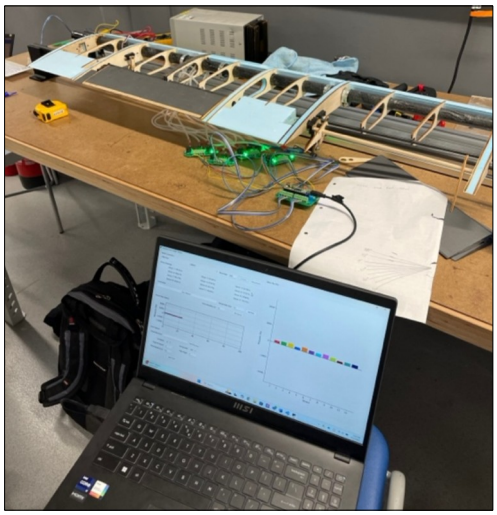


Figure 4, PAS testing

3.7.1. PAS setup

The first step during integration was setting up the Pressure Acquisition System and calibrating it with the wings in order to fulfil the pressure sensing part of requirement 3.3.3.a. Due to the simplicity of the system, it was ready well in advance so as soon as the wings installed their pressure tapping wings, decoupled testing was able to take place. This played perfectly into the planned parallel assembly and testing approach. The flap and aileron teams were able to continue working on installing the aileron while each pressure tube was tested to ensure there were no blockages or tears.

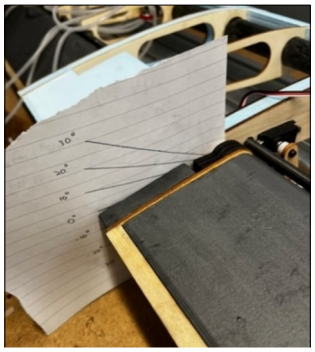


Figure 5, servo calibration setup. Piece of paper with angles traced from protractor

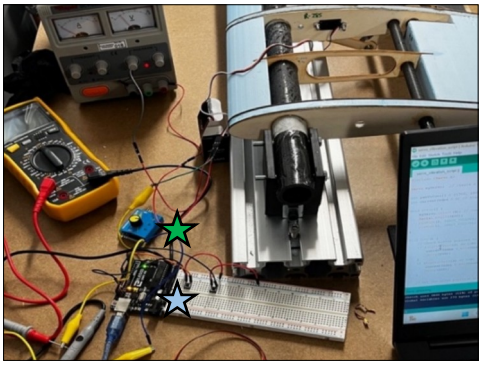


Figure 6, servo calibration circuit (blue start) and servo tester (green star)

3.7.2. Servo Calibration

A 3 step system was developed to properly calibrate the servos as per requirement 2.3.3.j and to ensure the accuracy of control surface deflections. The first step was using the servo tester as per requirement 2.3.3.l. This was done as soon as final mechanisms were built and could be tested. Was used as a prove the servos could reach the desired maximum angles and weed out any servo issues early on. Second step was using the custom servo calibration circuit to map control surface deflection angles to PWM inputs. This was done by finding the maximum and minimum deflection angles, recording their respective PWM inputs, then mapping a few other angles in between. This data was then used to create line of best fit equations between the points, fully calibrating the mechanism. This step provided a fast and reliable way to get these servo equations and could be done every time a mechanism team had to make changes to something in their subsystem. The final step was implementing the equations in the TMS code, and doing a final fine tuning using an electronic level sensor. This approach worked extremely well, allowed for adaption to the needs of the mechanism teams while still giving enough time to meet the 1 degree of accuracy requirement.

3.7.3. Wing floor boards

Figure 7, Integrated P-wing

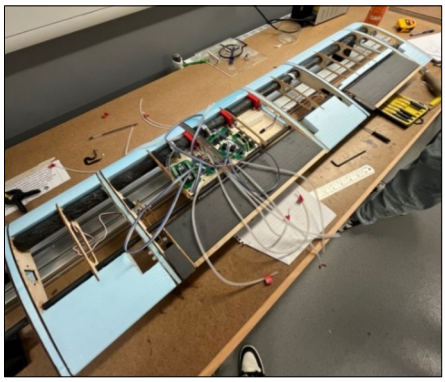


Figure 8, ~~Integrated~~semi-integrated S-wing

The design for the wing floors changed multiple times throughout the journey of the project. These boards were designed to house the Pressure Sensor boards and the pressure manifold. Originally, the PAS was designed to be self-contained in a box since it offered an easy way to integrate into the wings: simply placing the box inside. Unfortunately, once the wings moved further along in their development it became apparent these boxes were too large to fit inside and there was simply no way to minimize them further. This led to a redesign where the PAS was installed to wooden 'floor' panels secured into the wing. Initially, these floors were designed by the Avionics team, integration, the design was handed off to the wing teams since they had the most knowledge of how to fit it inside and had the manufacture experience of already having built the wing.

Figure 7 & 8 show the integrated wings and avionics. The S-wing floor boards were screwed in from the bottom and then skinned over while the P-wing floor boards had access panel hatches that screwed in from the top. Both these system were relatively easy to integrate. The electronics could be screwed in before being placed inside the wings and then they were wired to each other. Then the pressure tubes, that had been numbered previously, were connected to their corresponding sensor.

3.7.4. Pod avionics bay

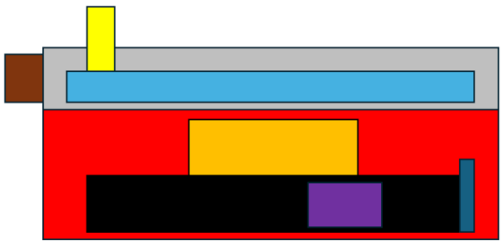


Figure 10, AvBay block diagram, legend in appendix 1

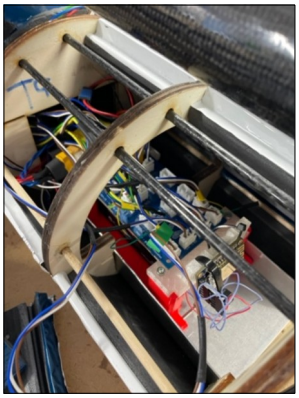


Figure 11, AvBay in pod

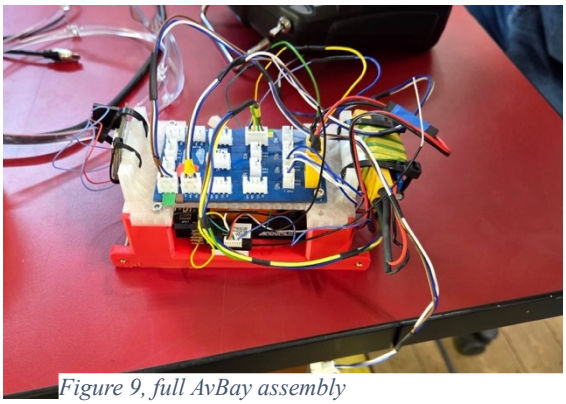
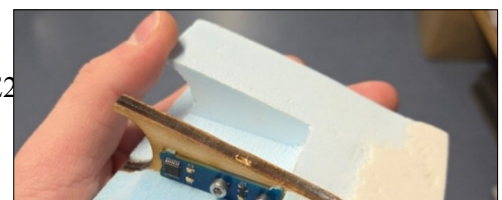


Figure 9, full AvBay assembly

As with the floor boards, plans for the Avionics bay changed drastically between design and integration. Originally the pod team were given a cubic geometric envelope to provide the space for and design mounting holes. Halfway through the manufacturing phase The pod team decided this was not a workable solution as the payload decking had to remain clear and instead gave 2 aluminium plates mounted in the roof between formers with holes for bolts and zip ties. During assembly it was found to be an unreasonably difficult task to fit components in, especially considering the numerous times it would have to occur as only had 1 complete cube set (fully integrated avionics module) was available so this solution did not fulfil requirement 2.3.3.a's "accessibility." Upon observing other teams mounting in the payload decking the mounting solution was reassessed and a new an agreement with the pod team was made to place a removeable avionics bay in the payload deck.

This avionics bay consisted of two 3D printed parts: the lower portion mounted the Cube module and the transceiver modules, the upper portion mounted the harness distributor PCB as well as the 2 power modules (power brick and BEC). The AvBay was then screwed into the pod floor from below to ensure it didn't slide or move around during testing.



3.7.5. Angle sensor Integration

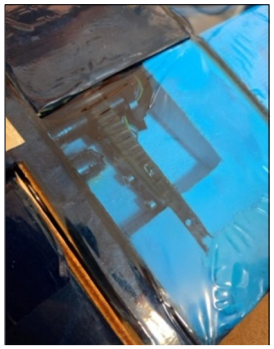


Figure 12, integrated angle sensor under skin

Figure 13, test sensor integration in mock-up wing

As laid out in the previous section, the flap angle sensors were mounted to a rib inside the wing as seen in figure 12. The flap spar had a magnet epoxied to it that needed to align with the sensor. The final magnet used was not the high quality Neodymium magnet as this had not arrived on time, instead a regular button magnet was used which could have had a detrimental effect on the sensor's accuracy. The spar also ended up being slightly off axis to the sensor which could have also affected the performance. Ultimately, the accuracy and sensitivity were far better than anticipated and its believed these imperfections did not cause any major repercussions but for future applications these issues should be highlighted and sorted.

This design was surprisingly easy to integrate and implement, only requiring a small amount of foam to be trimmed with a Stanley knife in order to fit. This was one of the rare instances where pre-planning actually came through to integration without any major changes being needed. The only downside of this design was neither the sensor or magnet were easily removeable so if something stopped functioning they would be quite difficult to replace. The skin would need to be cut which would have been a major setback for the wing team if this was ever needed so another access panel or something of this sort should have been considered for this system.

4. Structures

Harness strain relief ensured the continued electrical connections across the UAV, this was achieved via zip ties connecting harnesses clusters to various structural elements like spars and ribs. Zip ties are an unideal solution with a risk of pinching on small, non-armoured wires however there use is widespread enough in similar applications for them to be satisfactory in prototyping. Originally extra strain relief was planned with socket PCBs but were cancelled after manufacturing issues. Strain relief to the cube was provided by the Harness distributor PCB to provide specific protection to it.

Ultimately avionics fixings loads were of little relevance, none acted as load transmission elements and the avionics components they held had negligible masses, creating negligible loads. The heaviest elements: the PAS and the Cube pilot were mounted on large, robust parts which provided plenty of strength, if lacking in mass savings. One of the few structural concerns of note with the implementation of the avionics bay was its co-existence with 3kg of payload masses, which could cause a crushing hazard if improperly secured, the avionics bay was thus designed with robust walls and columns as well as redundant mounting solutions to protect the cube pilot and other OTS electronics.

Accommodation of avionics systems have greatly shaped wing structural design. Following discussions with Port Wing personnel, a design change to their ribs have been requested by the avionics team to allow for easier wiring. This meant a different shape of the main cutout in the rib. The new shape introduced a vertical stiffener in the cutout that helped secure wires. Because of this, the design requests have had a positive impact on the axial load bearing capability of the wings.

On the Starboard Wing, the sensor mounts have caused issues. Due to the stressful and rushed nature of the manufacturing process, the rib that would have housed the sensor was prematurely glued onto the spar before the proper mounts were machined. This meant that some of the foam of the trailing edge mass needed to be cut from the wing and holes in the rib had to be drilled with a hand drill. This was done under the careful watch of the Starboard structural specialists to ensure minimal adverse effect on structural integrity of the wing.

In accordance with design requirement 3.3.4.a, the team had to develop a stability assist system using the built-in sensors of the flight controller. For this to function as expected, it was important for the Cube to be at the centre of gravity of the UAV; this was predicted to be near the main wing joint. As such, a full mass report was carried out by members of the team to quantitatively find the location of the mass centre for the drone. This report can be found in Section 5.1 of the BlueBird company data repository.

The team has identified several ways to improve future iterations. First, using more robust wire securing methods, like strain relief clamps or brackets instead of zip ties would offer support without pinching the wires. Secondly, better planning would have allowed for the originally planned socket PCBs to be manufactured, that would have ensured proper connection between parts of the UAV. Thirdly, given the delicate electronics in the POD, padding could be introduced in the future to protect these devices. Additionally, drawing on the results from part 3.4.2, introducing damping in the wings could have improved sensor and control surface performance. Finally, to account for the shift in the CG during flight, more advanced flight controller calibration could be done in the future to give better stability assist performance.

5. Aerodynamics

There were 2 main requirements for the BlueBird corporation wings. First was to maximise endurance at a given C_L a to enable it to loiter for long periods of time while it records the whales. Second was to achieve a slow landing speed in order to land on the short floating airstrip.

5.1. Predicted vs Experimental

The data for S Wing was chosen for analysis as it performed better than P Wing. The data collected was used to compare and analyse the predicted lift coefficients, drag coefficients and drag polar against the given angle of attacks. The NACA 2415 airfoil profile was chosen because matched the requirements for L/D ratio and Cl max better than the other airfoils considered. All data collected from the Company B Team 12 S-wing Aerodynamic Overview document in the repository.

5.1.1. C_L *vs* α

The

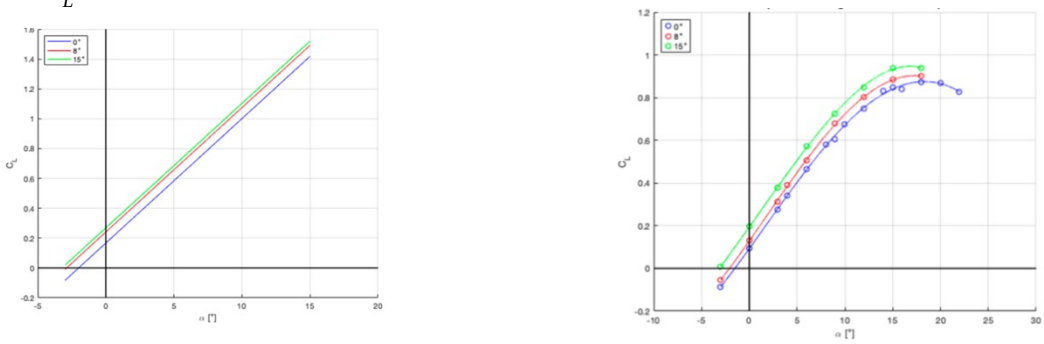


Figure 14, Predicted (left) and experimental (right) lift coefficient for varying angle of attack and flap deflection angles

above graph shows the predicted lift coefficient plotted against the angle of attack for different flap deflections on the left and the experimental values on the right. As seen in Figure 14, the Cl rises linearly at lower angles of attack and begins curving off around 15 degrees. This curving off represents flow separation occurring at the tailing edge of the wing at higher angles of attack~~ing~~, eventually leading to stall at almost exactly the angle predicted.

The behaviour of the flap was also as expected, causing an increase in lift at a given angle of attack. Looking at the predicted data, the 8 degree flap deflection is noticeably higher than 0 degree prediction but the 15 degree prediction follows the 8 degree much closer. On the other hand, this relationship seems to be reversed for the experimental data, especially at lower angles of attack the 0 degree results follow the 8 degree much closer. This discrepancy could be due to slack in the flap mechanism allowing the air to push the flap more closed at lower flap deflections, causing a decrease in lift. More flap deflections would need to be prepared in order to properly diagnose this discrepancy.

Overall, the experimental Cl was found to be lower than predicted. This is most likely due to imperfections introduced in the wing profile during manufacturing which is to be expected as, realistically, the wing would never be manufactured to the perfect standards assumed in the predicted model.

5.1.1. *Drag vs* α

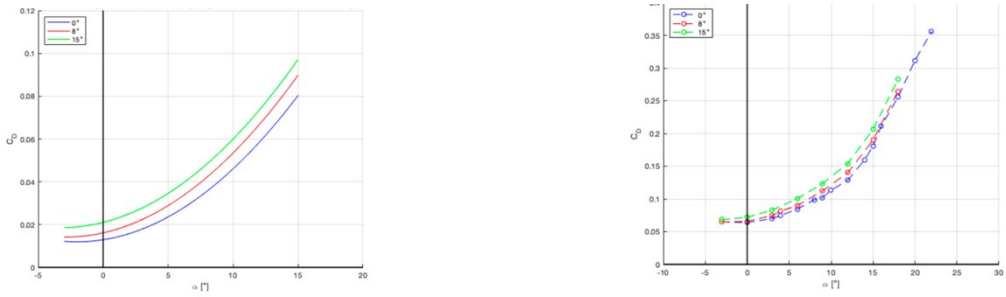


Figure 15, Predicted (left) and experimental (right) drag coefficient for varying angle of attack and flap deflection angles

The above graph shows the predicted drag coefficient for varying angle of attack and flap deflection angles on the left and the experimental values on the right. The experimental data follows the overall shape of the predicted data, with drag increasing with angle of attack. Overall, the drag is much higher than predicted, but, as with the Cl being lower, this is most likely due to defects introduced in manufacture such as surface inconsistencies and holes and cavities allowing flow to penetrate into the skin, creating more air resistance. Another major factor causing this increase in drag is the flap mechanism protruding from the underside of the wing. It's quite a large system and would be causing quite a large interference with the flow, increasing the drag.

The drag for each flap is closer together that predicted, probably due to flap mechanism causing a high interference with the flow. This interference is probably offsetting some of the increased drag caused by higher flap angles.

5.1.1. C_L vs C_D

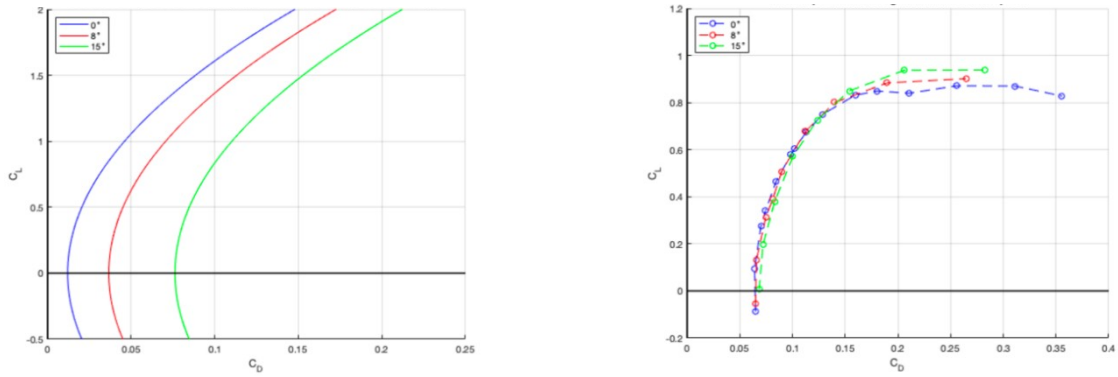


Figure 16, Predicted (left) and experimental (right) coefficient of lift vs coefficient of drag for varying flap angles

The above graph shows the predicted drag polars on the left and the experimental polars on the right. The experimental drag polar indicates that the drag data for varying flap angles is more closely clustered than predicted for a given C_L. While drag increases with flap deflection, the effect is significantly lower than expected. The zero-lift drag coefficient, C_{D0}, is also higher than anticipated, likely due to the same factors discussed in Section 3.2.

To mitigate these discrepancies, design modifications should focus on improving flap integration to minimize profile inconsistencies and eliminate cavities. Extending the chordwise leading edge could help maintain a consistent aerofoil shape and ensure proper skin tension. Additionally, addressing errors in skin application, particularly on the underside of the leading edge by eliminating plastic film folds and improving tack adhesion would enhance surface uniformity and reduce unintended drag sources. [21]

5.2. Conclusions and Recommendations

Overall, the wing underperformed compared to expected data by around 20-40% in most metrics. This was to be expected since real results never exactly match reality but changes could have been made to make these closer. Manufacturing discrepancies aside and assuming the wing was built to the best standards it physically could have, the largest contributor to the underperformance is assumed to be the drag caused by the flap mechanism. It protruded for more than it needed to and was also extremely bulky. More care could have gone into minimizing, streamlining, and considering the aerodynamic performance of this mechanism. This would of course come with trade-offs against it's rigidity and reliability so this would needed to be accounted for. Additionally, had more data points been recorded, more in depth observations could have been made. Had more flap deflections or greater angle of attacks been tested, the flap effects and stall characteristics could have been better defined and better conclusions drawn.

6. Project Management

At the start of the design phase, the team naturally fell along the lines of hardware and software subgroups covering the sensors, circuitry and fittings and the ground station and cube configuration respectively. This was largely due to the skills and knowledge of team members from previous projects. Initially the hardware subgroup was broadly over manned and the software slightly under manned, partially due to the interests of team members and partially misunderstanding of the avionics course components software weighted balance. This imbalance levelled and became hardware weighted in the manufacturing phase, with the introduction of integration and calibration tasks.

Early communications with other teams across the company consisted of primarily hardware drivers back and forth. Going into the manufacturing phase these communications expanded to cover integration, experimental aerodynamics and assembly timelines. Some issues arose in the lack of consideration for requested avionics accommodations which left members of the team in the position of shaving and drilling away at half assembled wings for mounting and harness solutions. Some of these were understandable mistakes with manufacturing, the ultimate takeaway is to push harder and more timely for such accommodations when integration issues are expected.

Going into the manufacture phase the single point of contact through project managers became a more fluid system of direct communication between relevant persons and other teams e.g. the Avionics Aerodynamics Lead to companywide Experimental Avionics Leads).

This was a product of increasing independence of members throughout both phases, with initial weekly meetings to discuss progress and consult with each other on top of ADVASI workshops and lab hours, reducing to just the lab hours and workshops are the manufacturing phase progressed. Whilst this independence allowed for more efficient working it did introduce the danger of knowledge loss if members became unavailable, however never to a critical degree, there was always at least 2 people who had the skills necessary, if not to the optimal standard, to complete critical tasks between both software and hardware.

The company assigned a number of broad internal deadlines, such as components design deadline for the end of November and a company wide dry assembly (requiring near finished manufacture) in the 3rd week of the manufacturing phase. Whilst these deadlines were useful to drive the design and manufacturing work, they were not practically applicable to avionics especially, with many designs waiting on parameters provided across the company, some of which such as the pod were not received until late in manufacture. This was of little relevance due to the low manufacturing effort of only OTS and small 3D printed fixings. The dry assembly was planned to figure out any integration problems and size harnesses before the final ‘wet’ assembly but due to its cancellation after company wide slippage the assembly of harnesses was further delayed, meaning the company carried out integration tests well into the ‘fixes’ designated reading week, nevertheless full readiness was achieved for the wind tunnel tests, with the first wing in the tunnel.

6.1. Hardware Project Management

The management of hardware was largely simple, consisting of many small tasks that could be quickly completed in blocks independently, Moreso as the year passed and trust in the team developed. Initial communications with other teams concerned both the available and require geometric footprints for various avionics components, requesting their provision of mounting points for the core avionics and sensors and PAS. Also discussed were the requirements for harnessways, the scope of avionics throughout the UAV and the possibility of extensions such as angle sensors beyond the wing flaps. Additionally consultation was provided for the mechanisms teams on servo selection based on avionics members previous experience as well as requests for a single coordinate working voltage to simplify power electronics.

These communications evolved into each team member interfacing with the appropriate members of other teams to ensure mountings would be workable with support through project management. Whilst this partially worked many issues were found during integration such as lack of harnessways through ribs, an impractical avionics bay in the pod, whilst some of these were understandable errors in manufacturing as occurred occasionally with avionic’s harnesses they suggest a need to push harder for integration requirements and do so more timely in future projects.

6.2. Software Project Management

Throughout the project, the project management style for software development changed numerous times. As the whole CubePilot ecosystem was completely new to all of us, learning how to use the CubePilot and write Pymavlink scripts took up most of the time. Until January, time was mostly spent in a waterfall approach, spending lots of time on each new feature and learning more about the CubePilot frameworks.

The first task was to achieve telemetry with the CubePilot over UDP. As with many software engineering tasks, the solution took many weeks to find despite being remarkably simple. This was followed by autopilot mode switching through the TMS, and then configuring and moving the servos. With each new feature added, the team gained further experience with the ecosystem, and development was much faster. Eventually, once all the key features were working, the development style switched to Agile development

Agile software methodology, an approach widely used by software engineers [18], is characterised by an iterative approach, working on new features in parallel, testing out this new feature then moving on to the next. Once each new feature had been achieved, there would still be areas to improve on; performance, simplicity of code etc. This allowed for the development a vast number of new features in a short period of time. Further; it meant that areas could be left areas completed on the UI side, which had not still been completed on the telemetry system. Adopting this agile methodology allowed the development all the features possible, before freezing a week before the wind tunnel test to ensure there was a fully working version.

For example, one such Agile task was implementing a gauge on the UI recording the output PWM by the CubePilot. A functional gauge component would be created on the UI. A function showing a gauge could be written by the UI, and tested purely on the UI side until it was complete. Then, amendments could be made separately on the backend, and once learning how to actually receive servo output data was achieved, the two components could be connected, and both the TMS and UI side tasks were complete.

With hindsight, ideally less time would have been spent working on innovating unnecessary features that exceeded the requirements. For example, lots of time was spent working on making the PAS feed data into the user interface, rather than, say, making the autopilot mode display show the correct value. Such tasks, whilst the outcome would be incredibly ideal, turned out to require lots of detailed

Many of the total features were implemented very close to the deadline. As the servo management code relied a lot on the work of other teams, much of it was left until the other teams had finished their mechanisms and were ready for testing. To mitigate this, lots of development time was spent on providing mock functionality. For instance, the ServoConfigure and CubeOrangeServo classes were developed remotely before reading week, using placeholder functions for the servo equations. Developing such placeholders was a task typical of Agile software development; the new feature was developed early before it was assumed to be a problem.

In retrospect, however, the team should have been stricter with what tasks to carry out next. As soon as a new feature was conceived, the team would instantly begin hacking away at a solution and try to get it done as quickly as possible, at times ignoring more important features that were yet to be fully completed.

As per Martin Fowler's Agile manifesto [19], Agile methodologies often rely on detailed requirements, and the detailed requirements Agile thrives upon were loosely defined in the requirements specification. The software side would have thrived more if the team had constantly defined and rewritten the requirements for the software, exceeding those set out in the specification.

The Agile manifesto also places an emphasis on customer feedback where the team pretended to be the 'customer' in order to evluate the work. As a result, at times the mantra "working software is the primary measure of progress" was not followed; whilst the software did work in the test, at times during development there were too many features that didn't work. This is likely a byproduct of only having two people focusing on the TMS/GCS code within the team; there simply weren't enough developers to effectively cover each new feature, and rather than reverting to a waterfall approach, features were developed in parralell. In most cases, seven, plus or minus two members, are ideal. [18]

7. Project Risk Assessment

Risk assessment followed recognised process which the team have innovated or improved. Risks were initially divided into three types – flight operations, manufacturing and the overall project. Then each type of risks was identified realistically, sensibly and appropriately to the project. A customised scale system was developed and implemented to evaluate the severity of each risk, applying structured approaches suggested in ISO (2018). Risks and further mitigations were also properly captured and implemented in general. Since some of team members already had remarkable prior knowledge about avionics, the team were able to maximise the benefits that risk assessment can bring. Additionally, every team member complied with what lab staff have instructed to prevent safety incident. No significant risks were missing. Even if they were missing, team members discussed in the review.

One of the important risks identified is damage to cubepilot. The project could have faced significant setbacks if further mitigations had not been put in place in an appropriate manner. To reduce the likelihood of damaging the cubepilot, it was decided to install it at the final stage of the assembly process meaning that it did not need to be dealt with during earlier stages. The cubepilot must be connected via the provided USB adaptor to ensure electrical safety. It was also noted that the servos must not allowed to be powered through the cubepilot as it is highly likely to exceed the cubepilot's power handling capability and may lead to voltage droop or burnout of the internal circuit.

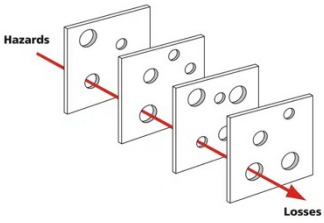


Figure 17.Reason's Swiss cheese model

The main point to reflect on is the prior knowledge. A high level of prior knowledge was undoubtedly one of the biggest strengths, but at the same time it could also pose a limitation which made the team rely on what the team already know and overlooking potential hazards. According to Reason (1990), prior knowledge can lead to rule-based or knowledge-based mistakes, where existing models are applied inappropriately to new contexts and results in errors that stem not from inexperience, but from overconfidence in familiar patterns. For instance, the familiarity led to an overconfidence in the designing process of the plugging system. The team were too focused on implementing an innovative plugging system which connects every part of UAV, but failed to recognise the structural stability concerns which could lead to a failure of avionics integration. This case can be viewed applying Reason's Swiss Cheese model (Figure 17). Overconfidence due to prior knowledge can be understood as an active failure. When combined with latent conditions, these blind spots can get through multiple layers of defence and probably result in system-level failure if not critically reviewed and re-evaluated.

Secondly, the risk assessment documentation had been updated occasionally, but not as frequently as it was supposed to be. As the team went through the project, there was tendency to simply implement the new changes rather than assessing and evaluating any additional potential risks that could arise from changes in the plan. To improve this, scheduling dedicated update sessions, which the team could systematically review and update risk assessment for corresponding plan changes, would be a good strategy. Regularly reviewing risks in an explicit way could have allowed the team to identify emerging issues earlier and made smoother progress.

A risk that was not foreseen at the early stage is supply chain management. Since students only dealt with relatively not complex SRAD components, supply chain management was at only a minimal risk for the team. However, there would be potential to become a critical risk in a larger scale of projects especially where more complex logistics and dependencies interface. For a higher level of SRAD projects, it would become more important for students that they should identify issues regarding supply chain management taking account for lead time. This risk can be mitigated by diversifying the supply chain for example.

One Area for improvement would be strengthening stakeholder engagement. A lack of stakeholder engagement led to communication issues between avionics team and other teams. Some sub-teams did not fully understand how long avionics integration would take. If the risks involving with stakeholders at the whole company level had been properly identified, every sub-team could have maintained clearer and more consistent communication without confusion the schedule.

8. Gate 3 - Design Sign-off

Gate 3: Assessment Proforma

Gate 3 Assessment result

Team number: **18**
Sub-assembly: **Avionics**
Company: **Bluebird**
Assessment date: **15/01/2025**
Overall grading given for Gate 3 (0-100%): **80%**
Assessor: **MG & SB**
Present/absent: **All present**

Gate 3 Comments & observations

Comments recorded by the assessor(s).

Strength areas & positive outcomes from the review:
- Drawings looking good
- Brought along the angle sensing prototype
- Managed the discussions and questions very well
- Overall excellent level of preparedness for the gate/build

Areas of concern or opportunities/actions for improvement:
- Need to see a detailed test process for the wind tunnel
- No tinkering/software adjustments in the tunnel – ensure all code and assemblies tested prior

9. Gate 4 – Quality Inspection

Gate 4: Assessment Proforma

Gate 4 Assessment result

Team number: **18**
Sub-assembly: **Avionics**
Company: **Bluebird**
Assessment date: **19/02/2025**
Assessment time and location: 17:30 M005
Overall grading given for Gate 4 (0-100%): 78
Assessor(s): MG/SB
Team members present/absent: all

Gate 4 Comments & observations

Comments recorded by the assessor(s). Please include photos/evidence where relevant.

Strength areas & positive outcomes from the review:
- Innovative and effective sensing solution, tested and calibrated pre-integration. Custom PCB developed using out-of-unit skills.
- Good positional accuracy flap +/-1deg SW+PW, ail +/-3deg SW, +/-2deg ail, rudder sub 1deg. Minimal hysteresis.
- Nice interface, robust and usable, great use of prior skillset.

Areas of concern or opportunities/actions for improvement:
- Significant gate slip, root cause at least partially company-wide integration slip. Consider how to manage upwards, possible mitigation via staged out-of-aircraft demonstrators – could you meet most gate objectives with these?

10. Reference List

1. Allegro Microsystems, *A1335 Datasheet*, Rev. 11, 2023. Available: https://www.allegromicro.com/~/media/files/datasheets/a1335-datasheet.pdf
2. Allegro Microsystems, *A1335 Programming Reference*, Rev. 4, 2021. Available: https://www.allegromicro.com/-/media/files/programming-manuals/a1335-programming-manual.pdf
3. CASA, *AC 21-99(1) Aircraft Wiring & Bonding,* Section 2 Chapter 8, Ed. 1, 2013, Available ; Advisory Circular: September 2013 Aircraft Wiring and Bonding | PDF | Electrical Connector | Electrical Wiring
4. ISO, *ISO 31000 : 2018, Risk Management - guidelines.* Vernier, Geneva International Organization for Standardization, 2018.
5. J. Reason, *Human Error*. Cambridge: Cambridge University Press, 1990.
6. Michael Doumpos, Constantin Zopounidis, A multicriteria classification approach based on pairwise comparisons, European Journal of Operational Research, Volume 158, Issue 2, 2004, Pages 378-389, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2003.06.011.
7. Montibeller, G., Franco, A., Multi-Criteria Decision Analysis for Strategic Decision Making. In: Zopounidis, C., Pardalos, P. (eds) Handbook of Multicriteria Analysis. Applied Optimization, vol 103., 2010, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-92828-7_2
8. Marco Cinelli, Miłosz Kadziński, Michael Gonzalez, Roman Słowiński, How to support the application of multiple criteria decision analysis? Let us start with a comprehensive taxonomy, Omega, Volume 96, 2020, 102261, ISSN 0305-0483, https://doi.org/10.1016/j.omega.2020.102261.
9. Beaudrie C, Corbett CJ, Lewandowski TA, Malloy T, Zhou X. Evaluating the Application of Decision Analysis Methods in Simulated Alternatives Assessment Case Studies: Potential Benefits and Challenges of Using MCDA. Integr Environ Assess Manag. 2021 Jan;17(1):27-41. doi: 10.1002/ieam.4316. PMID: 32681741.
10. Gasperi, M., Hurbain, P.“. Potentiometer Sensors. In: Extreme NXT: Extending the LEGO MINDSTORMS NXT to the Next Level., 2009., Apress. https://doi.org/10.1007/978-1-4302-2454-9_6
11. Ellin, A. and Dolsak, G., "The design and application of rotary encoders", Sensor Review, Vol. 28 No. 2, 2008, pp. 150-158. https://doi.org/10.1108/02602280810856723
12. D. Piyabongkarn, R. Rajamani and M. Greminger, "The development of a MEMS gyroscope for absolute angle measurement," in IEEE Transactions on Control Systems Technology, vol. 13, no. 2, pp. 185-195, March 2005, doi: 10.1109/TCST.2004.839568
13. Y. Y. Lee, R. -H. Wu and S. T. Xu, "Applications of linear Hall-effect sensors on angular measurement," 2011 IEEE International Conference on Control Applications (CCA), Denver, CO, USA, 2011, pp. 479-482, doi: 10.1109/CCA.2011.6044465 .
14. Barklund, M. (2024) *React in depth*. Shelter Island, NY: Manning Publications. Available at: https://ieeexplore.ieee.org/book/10745334 (Accessed: April 1, 2025).
15. Slatkin, B. (2020) *Effective Python : 90 specific ways to write better Python*. Second edition. Addison-Wesley: Pearson Education. Available at: https://proquest.safaribooksonline.com/9780134854717 (Accessed: March 31, 2025).
16. Williams, T., ‘The ground plane: Lord of the Board’, The EMC Journal, (72), 2007, pp. 26–31.
17. T. Charania, A. Opal and M. Sachdev, "Analysis and Design of On-Chip Decoupling Capacitors," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 4, pp. 648-658, April 2013, doi: 10.1109/TVLSI.2012.2198501.
18. J. Goodbrand, A. Shrivastav, “Team 12 – AVDASI 2 Aerodynamics Overview.,” Mar. 2025.
19. Bass, J.M. (2023) Agile software engineering skills. Cham, Switzerland: Springer. Available at: https://doi.org/10.1007/978-3-031-05469-3.
20. Shore, J. et al. (2021) *The art of agile development*. Second edition. Sebastopol, CA: O’Reilly Media, Inc. Available at: https://go.oreilly.com/stanford-university/library/view/-/9781492080688/?ar (Accessed: March 31, 2025).
21. I. Abbott and A. Von Doenhoff, “THEORY OF WING SECTIONS Including a Summary of Airfoil Data,” 1959. Available: https://aeroknowledge77.wordpress.com/wp-content/uploads/2011/09/58986488-theory-of-wing-sections-including-a-summary-of-airfoil-data.pdf

11. Appendix 1 – Design and Manufacturing Review

11.1. Down-selection

Criteria		Resolutio	Durability	Cost	Ease of Integration	Refresh Rate	Accuracy	0		
	X	1	2	3	4	5	6	7	Total	%
Resolution	1	X	9	9	9	3	0.333333333		30.33	30
Durability	2	0.1111	X	3	3	0.11111111	0.111111111		6.33	6
Cost	3	0.1111	0.333333	X	0.3333333	0.11111111	0.111111111		1.00	1
Ease of Integration	4	0.1111	0.333333	3	X	0.11111111	0.111111111		3.67	4
Refresh Rate	5	0.3333	9	9	9	X	0.333333333		27.67	27
Accuracy	6	3	9	9	9	3	X		33.00	32
	7							X	0.00	0
		3.67	27.67	33.00	30.33	6.33	1.00	0.00	102.00	100
		3.6	27.1	32.4	29.7	6.2	1.0	0.0	100	102.00

Figure 18 Pairwise comparison table

MCDA matrix

Criteria	Weightings	Designs →	Potentiometer		Incremental Rotary Encoder		Absolute Rotary Encoder		Gyroscope		Magnetic Angle Sensor		0		0	
	↓ Must add to 100% ↓		score	weighted score	score	weighted score	score	weighted score	score	weighted score	score	weighted score	score	weighted score	score	weighted score
Resolution	0.300	1	5	1.5	4	1.2	1	0.3	3	0.9	4	1.2		0.0		0.0
Durability	0.060	2	1	0.1	3	0.2	3	0.2	4	0.2	5	0.3		0.0		0.0
Cost	0.010	3	5	0.1	4	0.0	2	0.0	1	0.0	4	0.0		0.0		0.0
Ease of integration	0.040	4	1	0.0	3	0.1	4	0.2	2	0.1	5	0.2		0.0		0.0
Refresh rate	0.270	5	5	1.4	4	1.1	4	1.1	4	1.1	4	1.1		0.0		0.0
Accuracy	0.320	6	2	0.6	3	1.0	4	1.3	1	0.3	3	1.0		0.0		0.0
0		7		0.0		0.0		0.0		0.0		0.0		0.0		0.0
	1															
Total score for design →			3.6		3.6		3.0		2.6		3.8		0.0		0.0	

Figure 19 MCDA matrix

11.2. Sub-assembly design

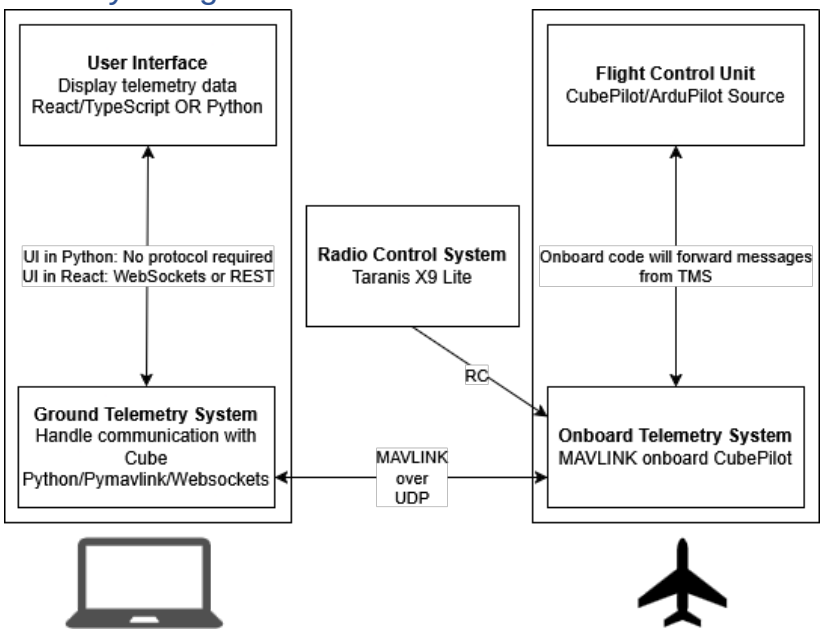


Figure 20, software interface diagram

11.3. Gate 2 outcomes

Gate 2a – Prototype Inspection

Gate status: Fail

Positive points (things to keep): -Functions mapped to reqs

-Tests somewhat mapped to functions

Things to consider improving/changing: - Function/risk table feels a little half-hearted - some missing risks, risks seem post-justified (how did you anticipate some of those obtuse ones and miss other obvious ones? ;)

-I suggest removing placeholder/example text in future submissions.

-It's not obvious how test table maps to function table - categories vaguely match but many of the tests don't assure against stated risks (also note that tunnel test is only free in pitch - no roll!)

-No actual test/inspection results and no comment on why/timetable for addressing. Use these docs in a way that is relevant to your context - if you're behind, recognise it and plan/show us how you're going to get back on track!

All software code used in the project: https://github.com/lucas-m-d/AVDASI2-GROUP-B

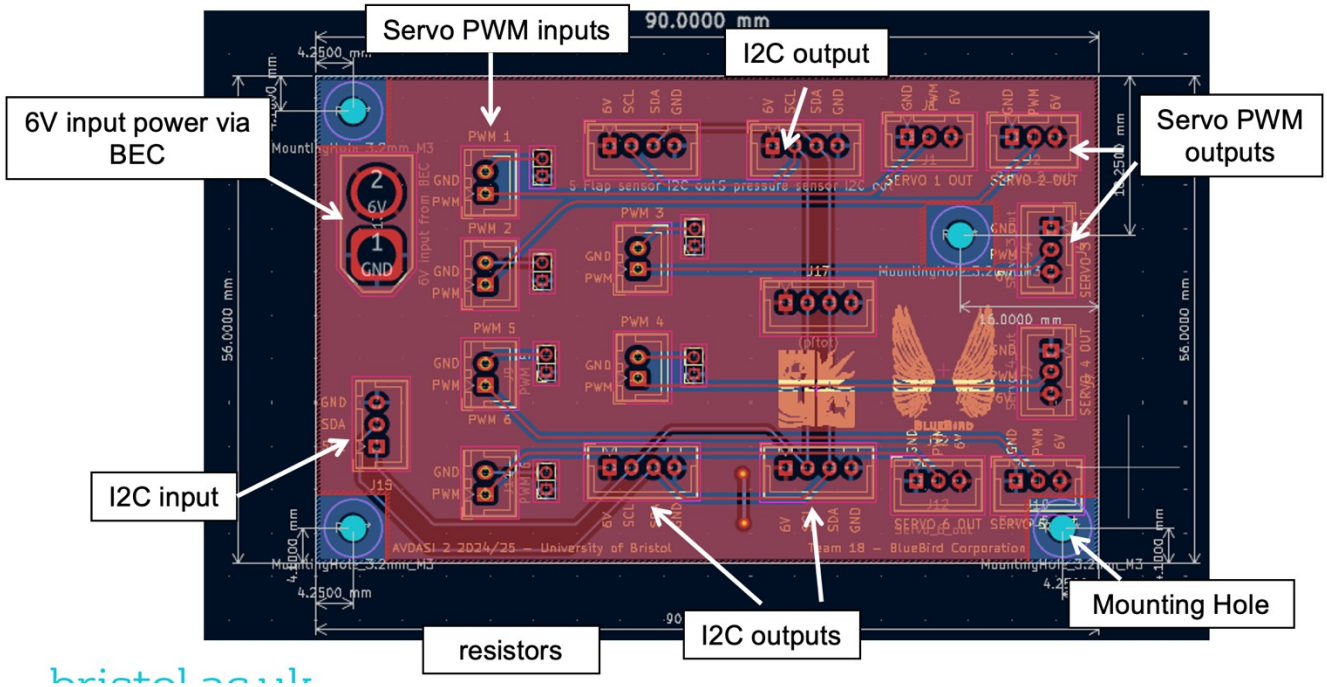


Figure 21, Power distributor schematic

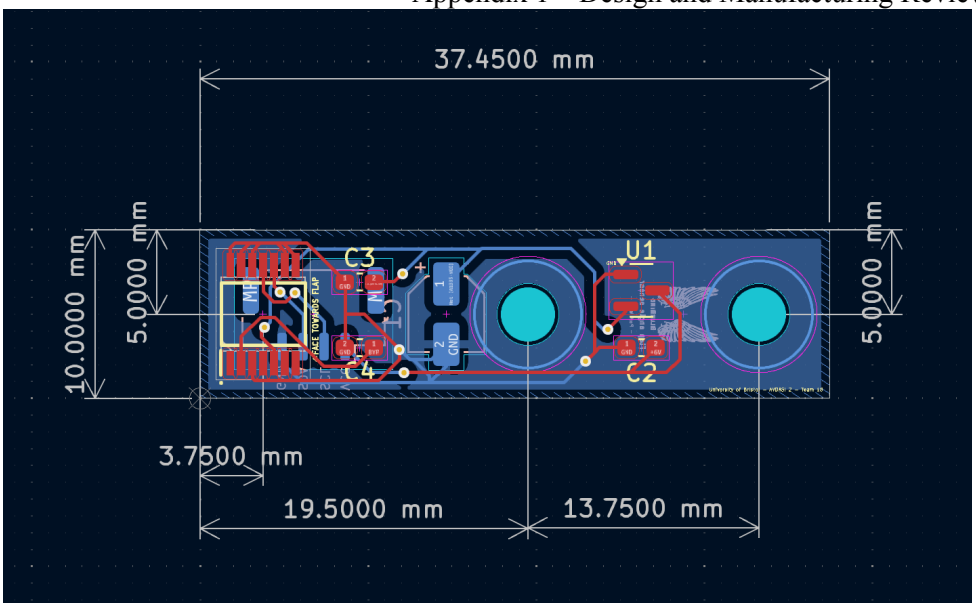


Figure 22, angle sensor PCB schematic

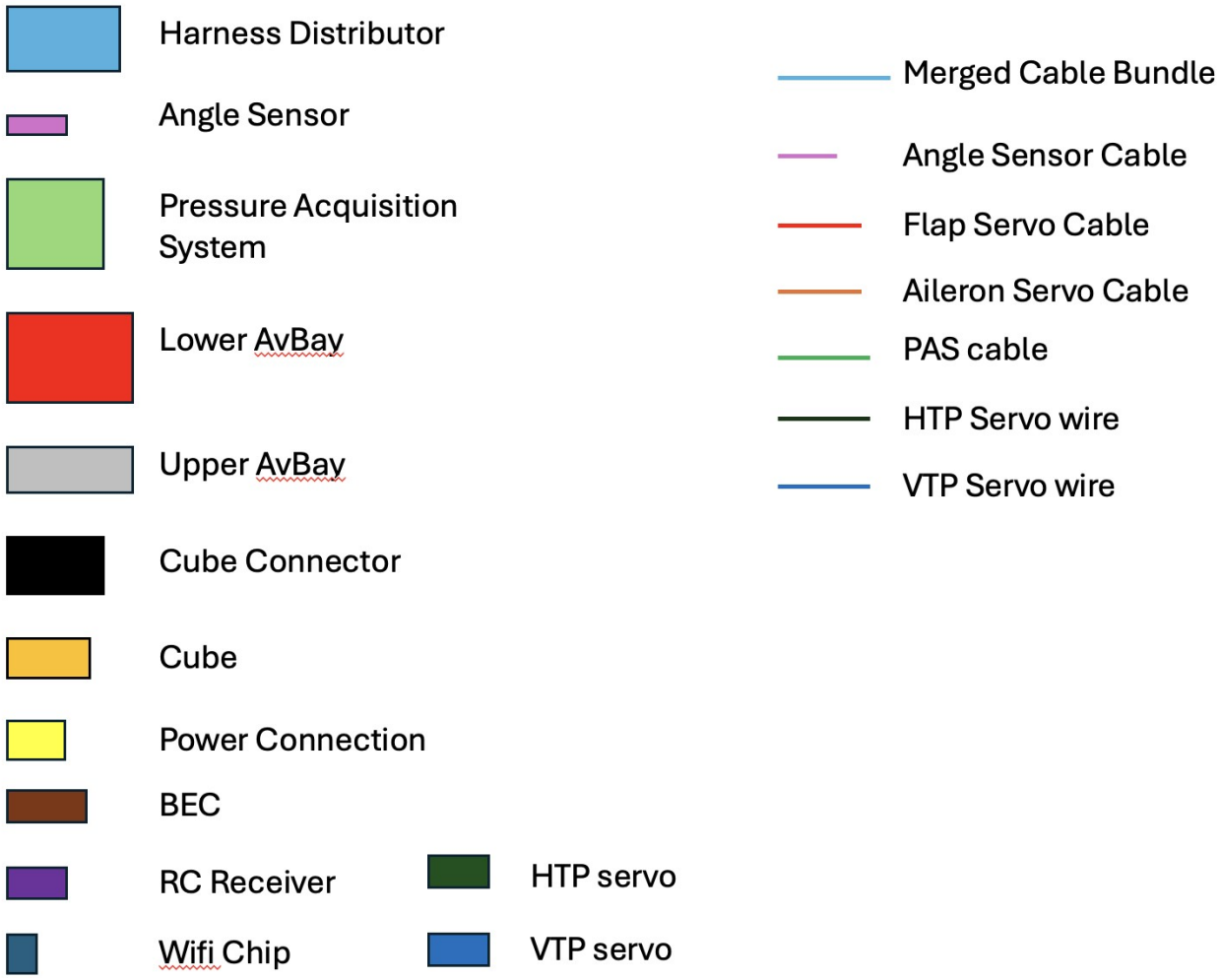
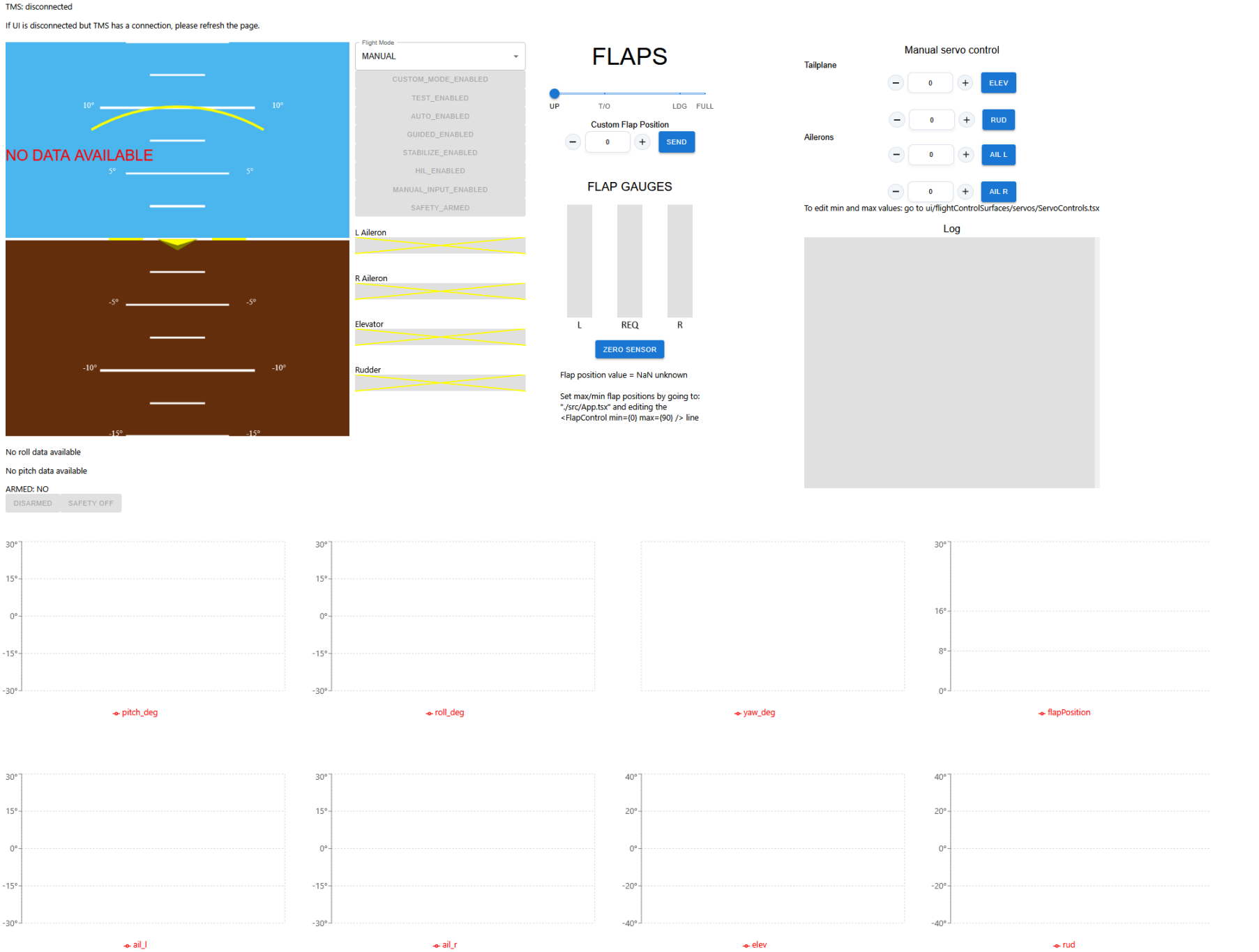


Figure 23, legend for physical diagrams

Final UI Design



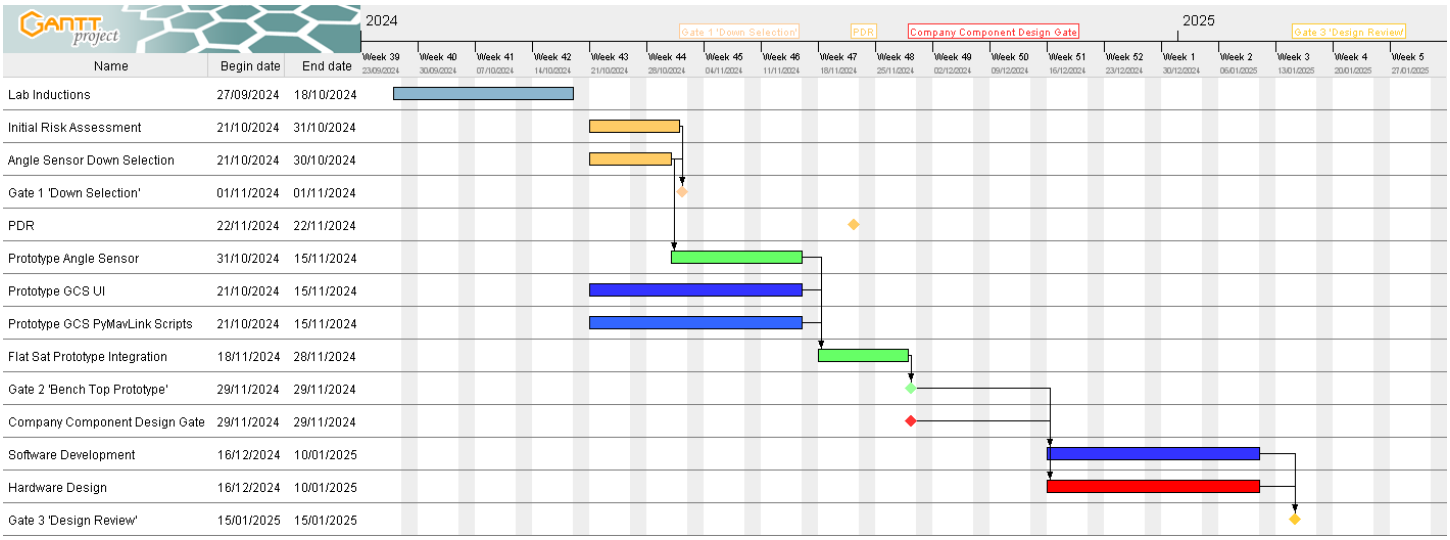
12. Appendix 2 – Structures

Any structures appendices go here.

13. Appendix 3 – Aerodynamics

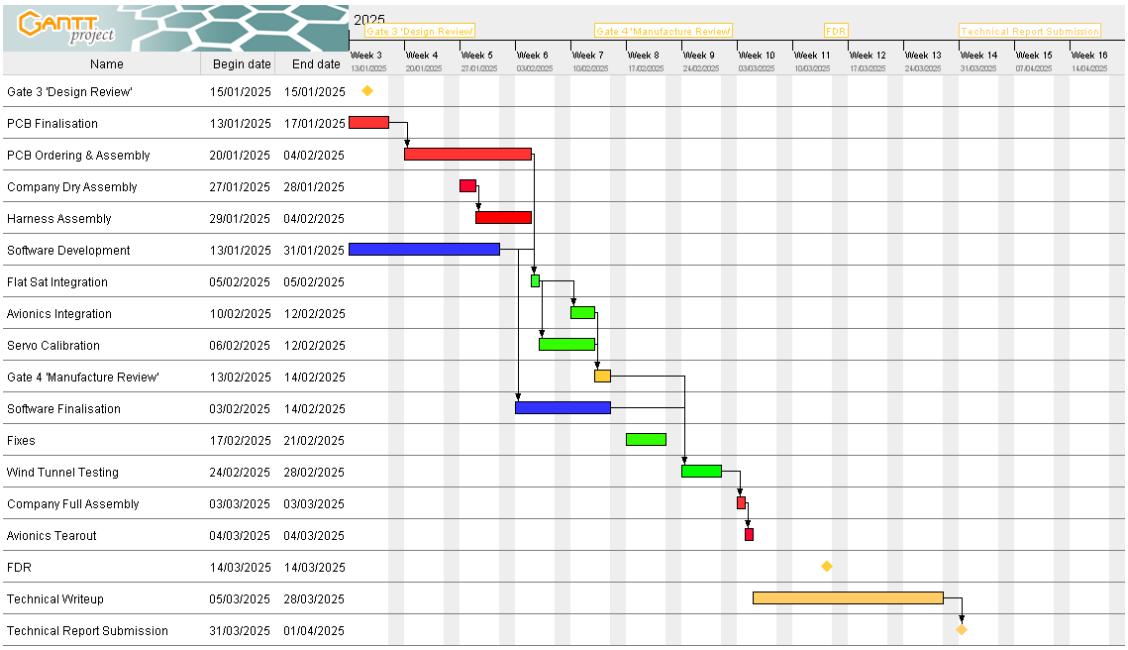
Any aerodynamics appendices go here.

14. Appendix 4 – Project Management



Name	Begin date	End date
Lab Inductions	27/09/2024	18/10/2024
Initial Risk Assessment	21/10/2024	31/10/2024
Angle Sensor Down Selection	21/10/2024	30/10/2024
Gate 1 'Down Selection'	01/11/2024	01/11/2024
PDR	22/11/2024	22/11/2024
Prototype Angle Sensor	31/10/2024	15/11/2024
Prototype GCS UI	21/10/2024	15/11/2024
Prototype GCS PyMavLink Scripts	21/10/2024	15/11/2024
Flat Sat Prototype Integration	18/11/2024	28/11/2024
Gate 2 'Bench Top Prototype'	29/11/2024	29/11/2024
Company Component Design Gate	29/11/2024	29/11/2024
Software Development	16/12/2024	10/01/2025
Hardware Design	16/12/2024	10/01/2025
Gate 3 'Design Review'	15/01/2025	15/01/2025

Figure 24, Design Gantt Chart



Name	Begin date	End date
Gate 3 'Design Review'	15/01/2025	15/01/2025
PCB Finalisation	13/01/2025	17/01/2025
PCB Ordering & Assembly	20/01/2025	04/02/2025
Company Dry Assembly	27/01/2025	28/01/2025
Harness Assembly	29/01/2025	04/02/2025
Software Development	13/01/2025	31/01/2025
Flat Sat Integration	05/02/2025	05/02/2025
Avionics Integration	10/02/2025	12/02/2025
Servo Calibration	06/02/2025	12/02/2025
Gate 4 'Manufacture Review'	13/02/2025	14/02/2025
Software Finalisation	03/02/2025	14/02/2025
Fixes	17/02/2025	21/02/2025
Wind Tunnel Testing	24/02/2025	28/02/2025
Company Full Assembly	03/03/2025	03/03/2025
Avionics Tearout	04/03/2025	04/03/2025
FDR	14/03/2025	14/03/2025
Technical Writeup	05/03/2025	28/03/2025
Technical Report Submission	31/03/2025	01/04/2025

Figure 25, Manufacture Gantt Chart

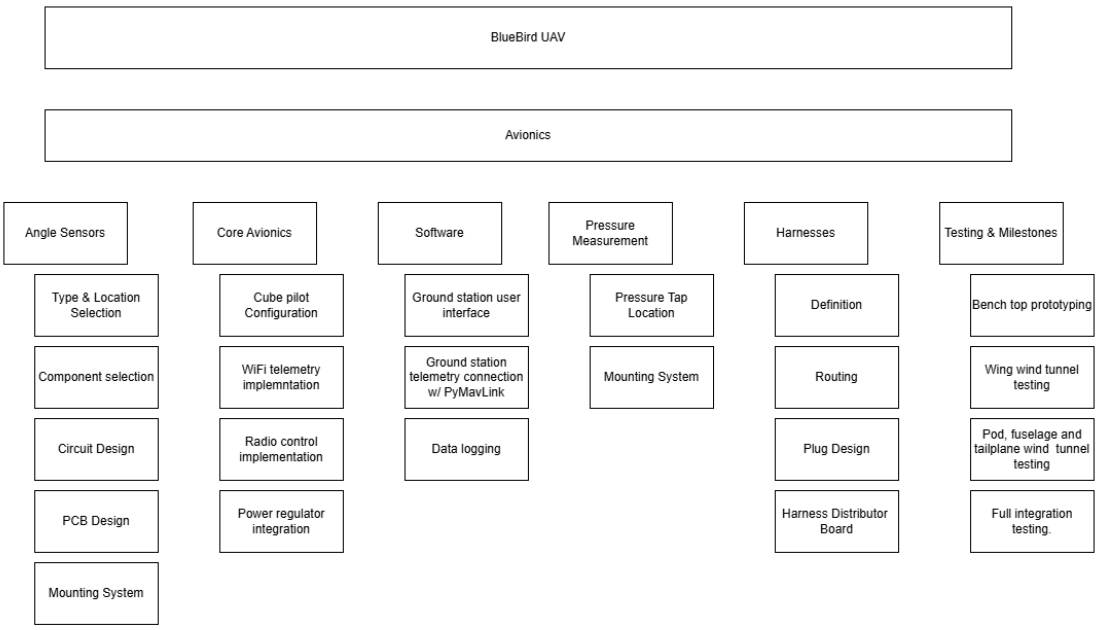
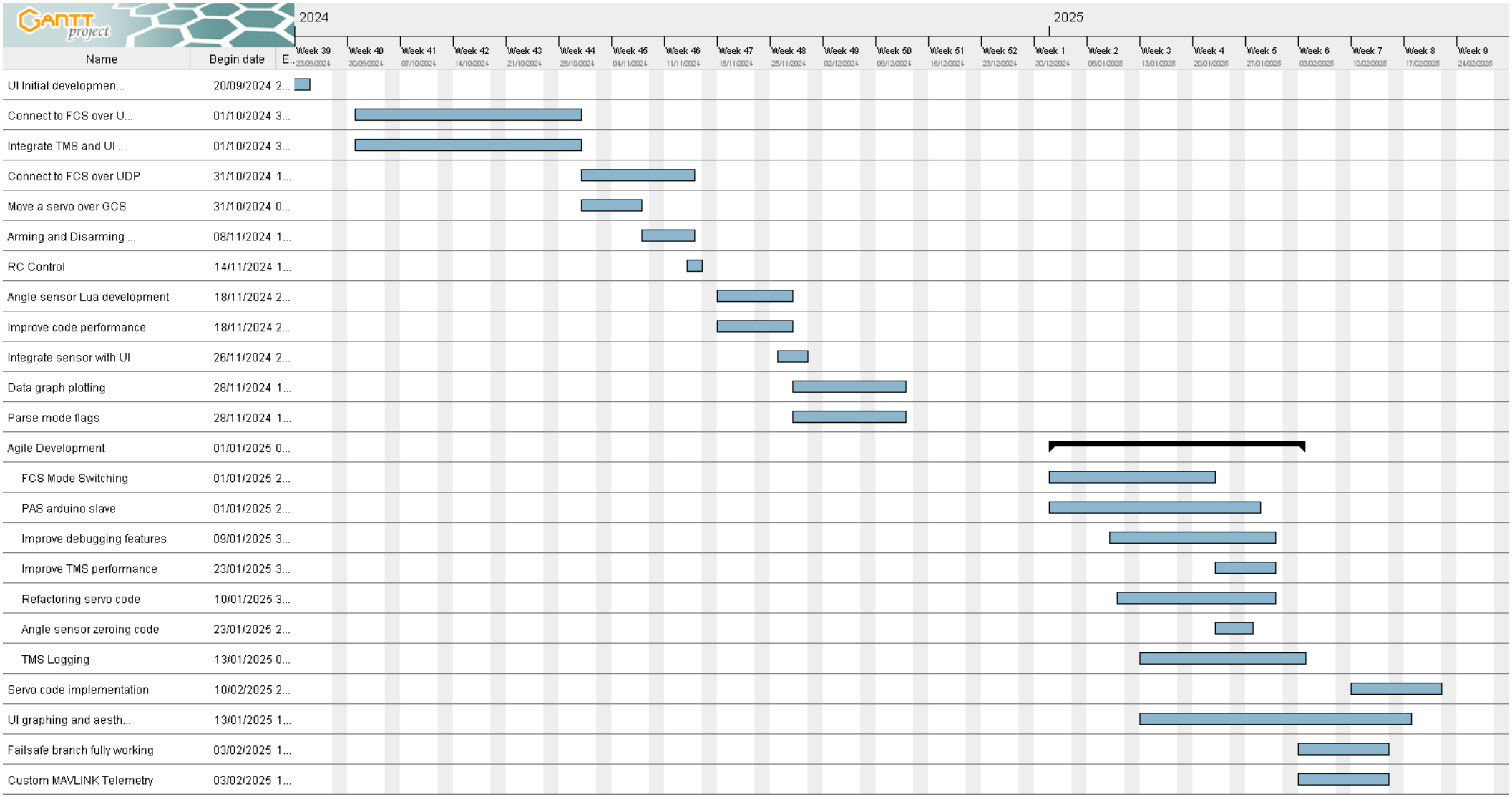


Figure 26, WBS

Software – retrospective Gantt chart



15. Appendix 5 – Risk Assessment

Your risk assessment documentation goes here.

Risk Matrix			Consequence				
			1	2	3	4	5
			Minimal	Minor	Moderate	Major	Catastrophic
Frequency	5	Almost certain	5	10	15	20	25
	4	Likely	4	8	12	16	20
	3	Possible	3	6	9	12	15
	2	Unlikely	2	4	6	8	10
	1	Highly unlikely	1	2	3	4	5

1-5 Negligible risk: Existing control measures are likely appropriate.

6-10 Tolerable risk: Additional control measures and mitigations required. Written action plan required.

11-25 Intolerable risk: Activity must halt. Resumption of activity requires approval from the safety board.

Figure 27, Risk Assesment quantification criterion

Risk ID (number)	Type of Risk	Description of Hazard	Description of Consequence or Effect	mitigation			Mitigations (identify existing and additional)	mitigation			Actions (with owners and date)
				Frequeny	Consequ	Risk number		Frequeny	Consequ	Risk number	
001	Flight Operations	**Damage to cubepilot:** Electrical damage and burnout, Mechanical damage, during assembly or testing.	Partial or total disabling of cubepilot. Replacements may be unavailable and have significant costs and lead time.	4	5	20	Keep cube away from hand tools. Integratec cube into pod avionics assembly last. Make connections via provided USB adapter. Never power servos through cube. Audit and approve all test circuits. Keep cube supervised at all times when not in	1	4	4	
002	Flight Operations	Damage to radio controller during assembly or testing	High cost, high lead time, easier to replace than cube.	2	4	8	Keep radiocontroller away from manufacturing areas when not needed.	1	3	3	
003	Manufacturing	Damage to other OTS and SRAD avionics components.	Medium to low cost and lead time for replacements.	3	2	6	Order and manufacture spare components where viable.	2	2	4	
004	Project	Component inventory shrinkage.	Replacement components required if not found.	4	2	8	Keep an up to date stock spreadsheet. Keep close track of high value components such as the cube and radio controller.	2	2	4	
005	Manufacturing	Misdesign of SRAD components	May show up in manufacture, requiring redesign, may cause assembly failures.	3	3	9	Conduct team reviews of all final designs. all components should be tested in advance of assembly deadlines.	1	3	3	
006	Manufacturing	Mismanufacture of SRAD components	Components may not function, may cascade into misassembly, remanufacture and assembly required.	4	2	8	Inspect all manufactured components for abnormalites, test components where appropriate.	2	2	4	
007	Manufacturing	Misassembly of Avionics systems	Reassembly required. May cause damage to components if mistakes make it to integration.	4	2	8	Produce assembly plans listing all avionics components, where they should go and in any required orders. Inspect all asemblies before integration testing.	2	2	4	
008	Manufacturing	Damage to 'graphite goose' structural components during avionics assembly.	Replacements likely not available, catastrophic damage could cause programme failure.	2	5	10	Notify other teams of tool work in the vicinity of graphite components and take additional care.	1	5	5	
009	Manufacturing	Damage to other structural or mechanisms components during avionics assembly.	Replacements available but may require complex manufacture or assembly.	2	4	8	Notify other teams of tool work in the vicinity of their components and take additional care.	1	4	4	Training
010	Manufacturing	Damage to avionics assemblies and harnesses.	Loss of components, reassembly required.	2	3	6	Ensure teams working near avionics assemblies have taken guidance from an avionics team member.	1	3	3	
011	Flight Operations	Servos improperly aligned to mechanisms.	Failure to achieve target control surface angle. Reassembly required.	4	3	12	Design and test sensor assemblies early in prototyping phase, test alignment. Ensure attachments are rigid.	2	3	6	Dicuss servo assembly with control surface teams.
012	Flight Operations	Servo angle limits not aligned with mechanisms.	Failure to achieve target control surface angle. Reassembly required.	3	3	9	Zero servos to a standard angle before integration.	1	3	3	Dicuss servo assembly with control surface teams.
013	Flight Operations	Angle sensors improperly aligned to mechanisms.	Realignment required, may require redesign of avionics structure components.	4	3	12	Design and test sensor assemblies early in prototyping phase, test alignment. Ensure attachments are rigid.	1	3	3	Discuss angle sensor assembly with control surface teams.
014	Manufacturing	Metal and graphite shavings or dust shorting exposed electronics.	Damage to components, power and data loss during operations.	4	4	16	All electronics should be covered in final assembly. Exposed components should be kept away from tool works. All components should be inspected and cleaned if neccesary before power is connected.	2	4	8	
015	Manufacturing	Cutting, heat and irritation hazards to personnel from tools and materials as described in technical service risk assessments.		3	4	12	Follow any risk assessments for equipment and materials provided for AVDASI 2 by FENG Technical services.	2	3	6	

Risk ID (number)	Type of Risk	Description of Hazard	Description of Consequence or Effect	Pre-			Mitigations (identify existing and additional)	Post-			Actions (with owners and date)
				Frequeny	Consequence	Risk number (f x c)		Frequeny	Consequence	Risk number (f x c)	
16	Flight Operations	Severed Wiring	Render key features to the project unusable Fire hazard Can break other electronics	3	3	9	Ensure we test wires before placing them Soldering joints	2	3	6	
17	Flight Operations	Battery fires	Could lead to catastrophic damage of uav	1	5	5	Only use certified battery parts/power sources	1	5	5	
18	Flight Operations	Power disconnection	Loss of power mid flight Render features necessary for the project unusable	3	4	12	Check all cables before flight	1	4	4	
19	Flight Operations	Data disconnection	Loss of data transmission to the software	3	3	9	Test the connections Write fail-safe code	2	3	6	
20	Flight Operations	Collateral damage from failed structure	Poses risk to the wind tunnel and the project	3	5	15	Ensure all structures will work Avionics in particular properly attached to the UAV	2	5	10	
21	Flight Operations	Fingers trapped in servo	Danger to the person	1	5	5	Require flight control surfaces to be armed before they move Ensure the code moving the servos is	1	4	4	
22	Flight Operations	Loss of telemetry	Losing connection enabling the UAV to be controlled	3	4	12	Ensure the code is entirely fail-safe on both ends Make sure that the data received from				
23	Flight Operations	Radio interference	Interference preventing 2-way communication	1	4	4	Ensure the correct radio bands are used and don't interfere with other teams	1	4	4	
24	Flight Operations	Faulty sensor data	Systems return faulty data, impacting aircraft performance	3	4	12	Adequately test all of the sensors Create a mock up electrical system disconnected from the AC	2	4	8	
25	Flight Operations	Software bug (generic)	Could render key software features unusable and risk not delivering a UAV suitable for the submission	3	3	9	Write fail-safe code in languages with appropriate error handling	2	3	6	

Risk ID (number)	Type of Risk	Description of Hazard	Description of Consequence or Effect	Pre- mitigation			Mitigations (identify existing and additional)	Post-mitigation			Actions (with owners and date)
				Frequeny	Consequence	Risk number (f x c)		Frequeny	Consequence	Risk number (f x c)	
26	Project	Loss of internal communication	Possible slowdowns of project; faults in implementation and manufacturing; possible health risks from miscommunication	2	4	8	Regular team meetings; encouraging social environment within the team	1	3	3	
27	Project	Loss of cross-team communication	Severe slowdown of project implementation, suboptimal design and manufacturing (possibly heavily increasing complexity or outright making project completion impossible)	3	5	15	Regular correspondence with other PMs and DTMs, possibly help from PDs and TDs	2	3	6	
28	Project	Unequal workload	Detrimental effects on team mental health and moral, leading to future shortcomings of team efforts	2	3	6	Keeping an environment where people feel safe to speak up; members and PM making sure to divide work equally	1	3	3	
29	Project	Disagreements	Slowing team efforts; could lead to lasting animosity between members reducing productivity	4	3	12	Providing platform for dialogue; use of respectful language; PM intervention	3	1	3	
30	Project	Slacking	Could lead to unequal workload (see above); lower moral among other team members; issues with Aerobucks	2	5	10	Regular status updates to PM and/or DTM; regular team meetings; PM oversight; worst case: Programme Director (Mark) involvement	1	3	3	
31	Project	"Lone geniuses"	Could lead to loss of internal communication, disagreements and unequal workload (see respective cases above); errors in design, calculations and manufacturing	1	5	5	Multiple people on specialisations, specialists working together (also corresponding with fellow specialists in other teams in company)	1	1	1	
32	Project	Poor data keeping	Risk of losing prior work; misunderstandings on team and company level; hard to find documents	3	4	12	Company level standardised data repository system overseen by DTMs and TDs, enforced by PMs; keeping copies of documents on teams as well	1	1	1	
33	Project	Wind-tunnel damage during testing	Further testing of project assembly becomes impossible, severe financial damage to UNI infrastructure	2	5	10	Structural integrity checked by UNI and company staff; Robust design; Avionics cables well secured and generally not running on the outside of drone; Careful wind-tunnel operation	1	5	5	