

# Assessment Task 1: Completion Plan

HIVE AURC AVIONICS (Data Analytics and Verification)

RMIT UNIVERSITY  
SEMESTER 2 – 2024

Supervisor: Dr. Glenn Matthews

Name	Student No.	Contribution
Matthew Ricci	s3785111	50%
Jeremy Timotius	s3779178	50%

<b>Contents</b>	<b>1</b>
1 Introduction . . . . .	2
2 Completed Tasks . . . . .	2
3 Remaining Tasks . . . . .	3
3.1 Task 1: Finish Communications Tasks . . . . .	3
3.2 Task 2: Implement UART Tasks . . . . .	3
3.3 Task 3: Verify Data from Flash . . . . .	4
3.4 Task 4: Verify RTOS Task Execution . . . . .	4
3.5 Task 5: Implement GPS . . . . .	4
3.6 Task 6: Sensor Calibration . . . . .	4
3.7 Task 7: Implement Communications with Payload . . . . .	5
3.8 Task 8: Health Monitoring Task . . . . .	5
3.9 Task 9: Analyse Impact of Sample Loss . . . . .	5
3.10 Task 10: Verify Kalman Gain as an Indicator . . . . .	5
3.11 Task 11: Design Error Determination for Redundancy . . . . .	5
3.12 Task 12: Dissimilar Sensor Configuration for A5 . . . . .	5
3.13 Task 13: Testing and Validation of Sensors and System Testing . . . . .	5
3.14 Task 14: Redundancy Testing . . . . .	6
3.15 Task 15: Documentation of Firmware . . . . .	6
4 Challenges . . . . .	8
4.1 Time Management and Early Launch Dates . . . . .	8
4.2 Communications and Collaboration with Avionics, Ground Communica- tions, and Redundant Systems Teams . . . . .	8
4.3 Lead Times for Accessing Laboratory Equipment . . . . .	8
4.4 Code Bugs and Unexpected RTOS Behaviour . . . . .	8

## 1 Introduction

The purpose of this Completion Plan is to outline a detailed timeline for the remaining tasks required to finalise the Engineering Capstone Project, ensuring all milestones and deliverables are met within the remainder of the semester 2 2024. This plan revisits the project's deliverables and scope, taking into account recent changes and outlining strategies to overcome any logistical or technical barriers.

The original scope of the project aimed for the Aurora V rocket to reach an apogee of 10,000 feet as part of the Australian Universities Rocket Competition (AURC). However, due to unforeseen circumstances, the AURC has been canceled. This cancellation does not significantly impact our capstone project. The main changes include the use of a different rocket motor, resulting in a new expected apogee of 7,000 feet and an updated launch dates timeline which are detailed in the Gantt chart included in this plan.

Despite these adjustments, the project's core objectives and deliverables remain focused on the successful development and deployment of the rocket's avionics firmware, data collection, analysis tools and data verification. The detailed tasks and milestones have been adjusted accordingly to reflect these changes in scope and deliverables.

## 2 Completed Tasks

The following is a summary of the milestones achieved in the project so far:

- **Successful Launch of Aurora I:**
  - **Avionics Data Logging:** The custom avionics system successfully logged flight data, including sensor readings from accelerometers, gyroscopes, and magnetometers.
  - **Firmware Development:** Custom firmware and libraries were created for the Arduino Nano, facilitating clean data collection and storage.
  - **Data Retrieval:** Data was successfully extracted from the onboard flash storage upon recovery of the rocket.
  - **Preliminary Data Validation:** Initial comparison of sensor data with commercial flight computers (Blue Raven) indicated general agreement in acceleration, velocity, and attitude trends.
- **Design and Implementation of Data frame Specification:**
  - **Structured Data Formats:** Developed structured formats for the collection and storage of sensor data, ensuring consistency when data logging to flash storage, and facilitating ease of post-flight data analysis for comparison against Blue Raven data.
- **Implementation and Verification of State Estimation Algorithms:**
  - **Attitude Estimation:** Implemented attitude estimation using unit quaternion rotation, which provided accurate real-time orientation data of the rocket.
  - **Velocity Estimation:** Developed a Kalman Filter model for velocity estimation, which enhanced the precision of velocity measurements by fusing data from multiple sensors.
- **Design of A3 Firmware Structure:**
  - **RTOS Task Planning:** Designed a high level RTOS architecture to efficiently manage time-critical tasks and handle the increased complexity of the system. This architecture ensures precise time management between different subsystems, allowing for the prioritisation of tasks and seamless integration of additional features as the firmware evolves.
- **Aurora II Launch Outcomes:**

- **Barometric Data Collection:** Adjusted avionics A1 firmware to ensure correct collection and storage of barometric data, addressing issues identified during the Aurora I launch.
- **Telemetry Testing:** Conducted preliminary telemetry tests using ground communication hardware for real-time data transmission in future launches.
- **RTOS Implementation for Aurora III :**
  - **High-Resolution Task:** Implemented a high-resolution task for capturing and processing sensor data at a frequency of 500Hz.
  - **Low-Resolution Task:** Developed a low-resolution task for periodic data logging and communication.
  - **State Update Task and Event Logging:** Created a state machine tasks for updating the rocket's state and logging significant flight events to trigger events based on the rockets state during flight.
  - **Memory Flashing:** Implemented memory flashing routines to ensure data storage during flight similar to previous flights.
  - **CAN Communications with Aerobrakes:** Established CAN communication protocols with the Aerobrakes subsystem, enabling coordinated control during flight.
- **Kalman Filter Implementation: State Estimation Enhancement:** Improved state estimation algorithms for accurate velocity and altitude measurements, integrating barometric and inertial sensor data through the Kalman Filter.

These completed tasks has formed the foundation for the project's next stages, ensuring that the avionics system is well-prepared for future launches and real-time data processing.

### 3 Remaining Tasks

This section details the remaining tasks required to complete the capstone project, with a focus on finalising all RTOS coding-related activities by Week 5 in preparation of upcoming Aurora launches. Due to the early launch dates and time constraints, most formal testing will be conducted post-flight. While these tests will occur after the launch, they are designed to provide valuable insights and guidance for future students undertaking similar projects, helping them understand the key considerations when testing firmware at the system level. Additionally, post-flight testing will address any edge cases and fill in code gaps to ensure a well-rounded code base.

#### 3.1 Task 1: Finish Communications Tasks

**Activity:** This task involves finalising the communication systems, specifically managing the LoRa and CAN protocols. The activities include adding and receiving messages from buffer queues to ensure data transmission and reception between the rocket's avionics system and ground stations or other subsystems.

**Timeline/Expected Completion:** Week 4-5

#### 3.2 Task 2: Implement UART Tasks

**Activity:** Implement UART tasks to enable command and control functionalities. This includes developing UART command implementations for testing purposes and retrieving data from flash storage. Additionally, it involves manually setting states for debugging to ensure accurate system behaviour during various flight conditions.

**Timeline/Expected Completion:** Week 4-5

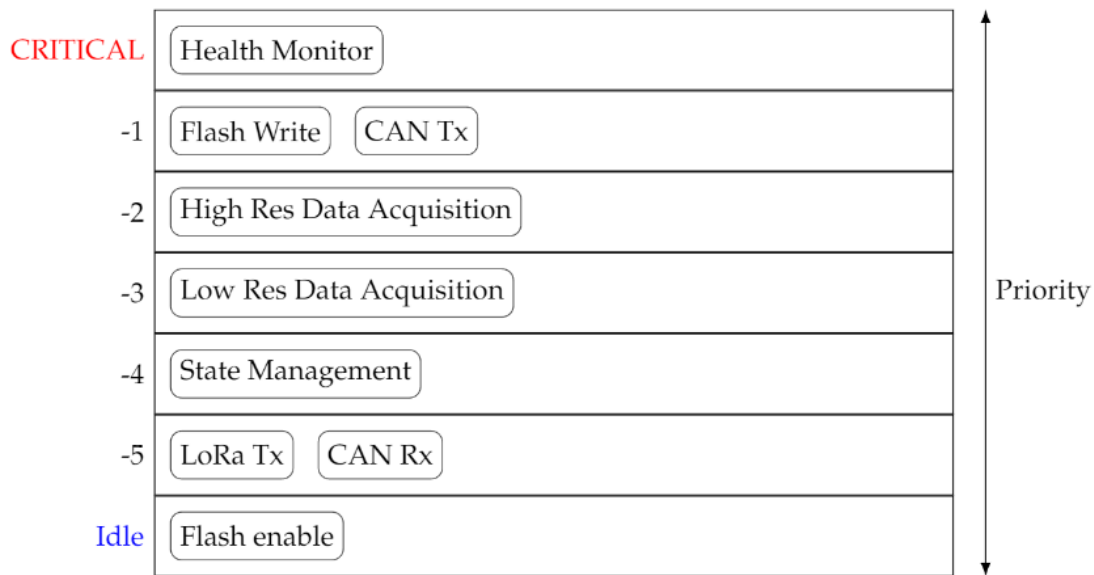


Figure 1: Enter Caption

### 3.3 Task 3: Verify Data from Flash

**Activity:** Ensure that the message buffer does not interfere with the written data. This task involves verifying that the data readout from flash storage is valid and is not corrupted due to buffering issues or context switching from the RTOS. The aim is to confirm the integrity and sensibility of the stored data.

**Timeline/Expected Completion:** Week 4-5

### 3.4 Task 4: Verify RTOS Task Execution

**Activity:** Conduct thorough verification to ensure that all RTOS tasks are executing as scheduled. This involves monitoring the execution times and sequences of tasks to confirm that they adhere to the predefined schedule and priorities.

**Timeline/Expected Completion:** Week 5-6

### 3.5 Task 5: Implement GPS

**Activity:** Implement the GPS task, utilising drivers already prepared by the Avionics, Ground Communications and Redundant Systems team. This task includes handling GPS data and ensuring it is correctly processed and sent to the LoRa communication system. Additionally, it involves setting up the GPS to operate in a non-blocking state, as it outputs a significant amount of data that is not required.

**Timeline/Expected Completion:** Week 4-5

### 3.6 Task 6: Sensor Calibration

**Activity:** Perform calibration of all sensors used in the rocket's avionics system. This ensures accurate readings and data integrity during flight. Calibration activities include setting up the sensors in a controlled environment recording any offsets that any of the sensors may output.

**Timeline/Expected Completion:** Week 5

### 3.7 Task 7: Implement Communications with Payload

**Activity:** Develop, implement and test communication protocols between the avionics system and the payload subsystem. This ensures that critical data is shared and processed accurately between these systems during flight.

**Timeline/Expected Completion:** Week 5

### 3.8 Task 8: Health Monitoring Task

**Activity:** Develop a high-priority health monitoring task designed to ensure system health by actively monitoring the status of various avionics components. This task will wait for flags triggered by other tasks, such as sensor malfunctions or microcontroller failures. If a flag is set, the health monitoring task will immediately enter a RTOS critical section with top priority, executing predefined protocols to mitigate the issue. These protocols include switching to redundant sensors or microcontrollers to maintain system functionality. Additionally, the task will manage timeouts on polling to prevent unnecessary delays and will monitor conditions such as the flash buffer reaching capacity.

**Timeline/Expected Completion:** Week 5

### 3.9 Task 9: Analyse Impact of Sample Loss

**Activity:** Analyse the effect of sensor data loss during errors and determine if the calculated state can recover from such loss. This involves creating scenarios where data samples are lost and observing the system's ability to maintain state estimation.

**Timeline/Expected Completion:** Week 5-6

### 3.10 Task 10: Verify Kalman Gain as an Indicator

**Activity:** Validate the use of Kalman gain as an indicator for system performance. This involves checking the accuracy of the Kalman gain values during different flight conditions and validating they provide meaningful insights.

**Timeline/Expected Completion:** Week 5-6

### 3.11 Task 11: Design Error Determination for Redundancy

**Activity:** Design logic for error determination to support redundancy. This involves creating algorithms that detect errors and switch to redundant sensors and microcontrollers to maintain functionality.

**Timeline/Expected Completion:** Week 5-6

### 3.12 Task 12: Dissimilar Sensor Configuration for A5

**Activity:** Configure and test dissimilar sensors to ensure they provide accurate data as per the primary suite of sensors. This involves configuring sensors with different specifications and integrating them into the system.

**Timeline/Expected Completion:** Week 5

### 3.13 Task 13: Testing and Validation of Sensors and System Testing

**Activity:** This task involves conducting comprehensive testing and validation of all sensors ensuring that each component performs accurately under expected flight conditions. The testing procedures will include the use of laboratory equipment such as shake tables and a vacuum chamber for barometric sensors. By doing so, conditions can be simulated to verify RTOS tasks.

#### Sensor Functionality and Edge Case Testing

- **Accelerometer Testing:** The accelerometers will be subjected to high and low G-force tests using a shake table. This will verify that the sensor outputs are proportional to the applied vibrations across the expected operating range.

- **Gyroscope Testing:** Gyroscopes will be tested using a driving simulator to validate their outputs are proportional to known rotation rates. A bias stability test will measure the drift over time, ensuring it remains within acceptable limits.
- **Barometer Testing:** Barometric sensors will undergo pressure validation in a vacuum chamber to simulate the pressure changes expected at an apogee of 7,000 feet. This includes verifying the accuracy of pressure readings against a Blue Raven flight computer. Additionally, temperature validation will be conducted in a temperature-controlled chamber to ensure the sensor compensates for temperature variations.
- **Magnetometer Testing:** Magnetometers will be calibrated outdoors, with readings compared against known magnetic field values to ensure accuracy.
- **GPS Testing:** The GPS will undergo stationary and dynamic accuracy tests. The stationary test will verify the GPS maintains stable position data in a fixed location, while the dynamic test will ensure it accurately tracks movement, such as on a moving vehicle.

#### **Mission Simulation Testing**

- **State Estimation Simulation Testing:** Simulated flight conditions will be used to validate state estimation algorithms, ensuring they accurately calculate position, velocity, and orientation during flight. This includes testing the apogee detection system to ensure it triggers recovery deployment at the correct altitude.
- **Aerobrakes Testing:** Aerobrakes will be tested under simulated flight conditions to verify their deployment.
- **Recovery Deployment:** The recovery system will be tested to ensure they deploy correctly at apogee. Indicators like LEDs will be used as a test to confirm that the recovery is triggered at the correct time before recovery is connected to avionics.

**Timeline/Expected Completion:** Week 6-11

### **3.14 Task 14: Redundancy Testing**

**Activity:** Conduct hardware-level testing to verify the redundancy mechanisms. This involves working closely with Avionics, Ground Communications and Redundant Systems team to design and implement tests that test the redundant systems function correctly in case of primary system failures.

**Timeline/Expected Completion:** Week 5-6

### **3.15 Task 15: Documentation of Firmware**

**Activity:** Document all firmware developed for the project, including detailed descriptions of functionalities, configurations, and usage instructions through Doxygen. This ensures that future teams or stakeholders can understand and utilise the firmware effectively.

**Timeline/Expected Completion:** Week 1-11

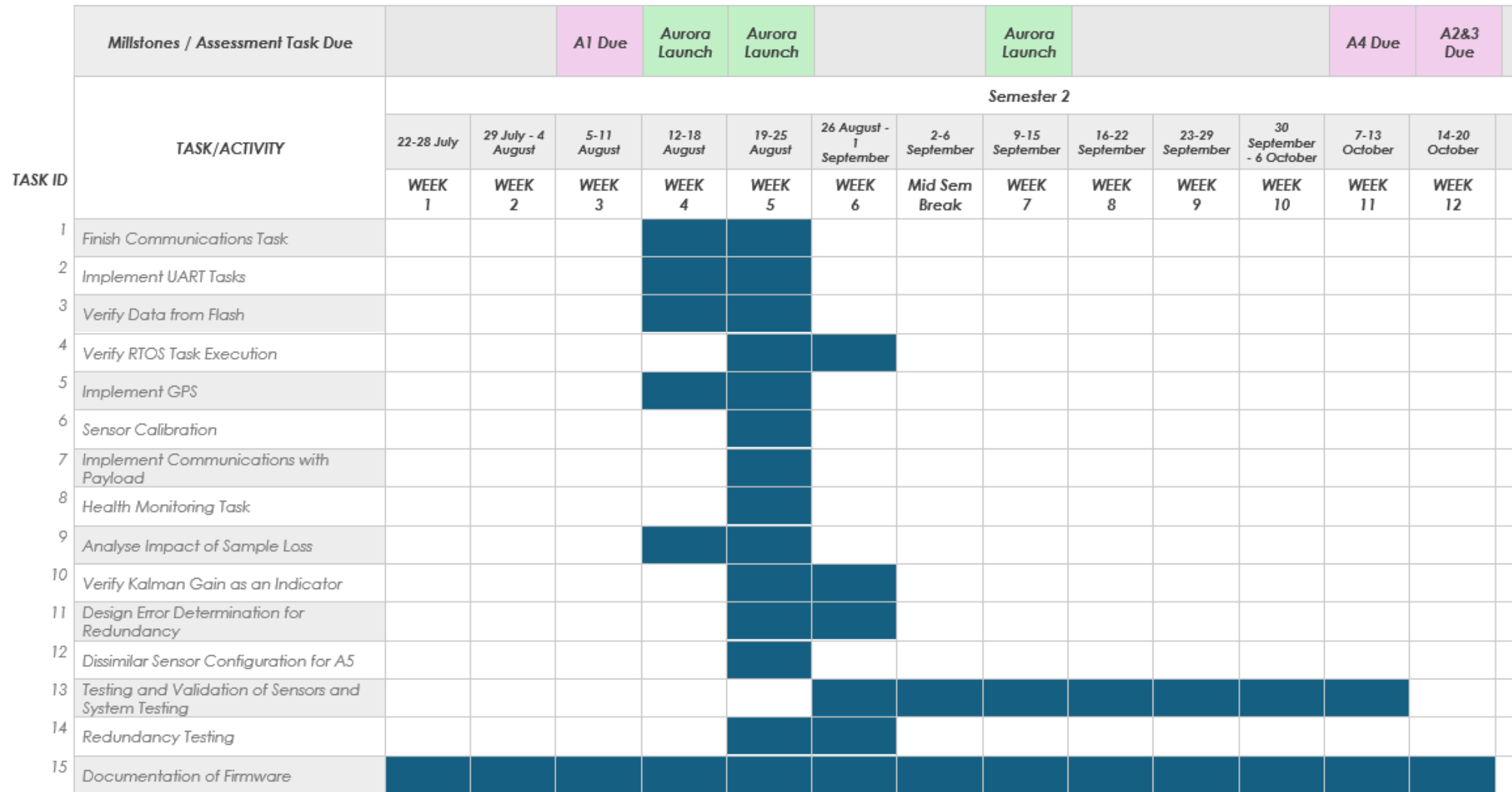


Figure 2: Gantt Chart - Project Timeline

## **4 Challenges**

Below are several challenges that have been identified that could potentially impact the timely and successful completion of the remaining tasks. This section outlines these challenges, and the strategies developed to overcome them.

### **4.1 Time Management and Early Launch Dates**

The launch dates scheduled for this semester present a significant time management challenge as they are primarily within the first half of the semester. With limited time to complete all coding tasks and prepare for the launch, there is a risk of falling behind schedule or incomplete tasks. To address this, certain tasks will be prioritised which mainly focus on completing all RTOS coding-related tasks by Week 6 before the 24th of August launch. In addition to prioritisation, scheduling practices will be utilised, including the use of Gantt charts to map out the timeline for each task and weekly capstone meetings to monitor progress.

### **4.2 Communications and Collaboration with Avionics, Ground Communications, and Redundant Systems Teams**

Effective communication and collaboration with the Avionics, Ground Communications, and Redundant Systems teams is key for the successful integration aspect of the rocket. Design conflicts between the hardware and firmware may pose problems. To address this, the team has established regular coordination meetings to discuss progress, align on requirements, and address any dependencies. Clear and detailed documentation of all requirements, progress, and changes is maintained to keep everyone on the same page.

### **4.3 Lead Times for Accessing Laboratory Equipment**

Access to specialised laboratory equipment, such as shake tables and vacuum chambers, is critical for testing and validation. However, the process of securing this equipment involves navigating approval processes, including consultations with RMIT technical staff and completing detailed activity and equipment risk assessments. To address this, the team has initiated the approval process well in advance, working closely with technical staff to ensure all safety and procedural requirements are met. Furthermore, a flexible testing schedule has been developed to accommodate any delays in equipment availability.

### **4.4 Code Bugs and Unexpected RTOS Behaviour**

Potential code bugs and unexpected RTOS behaviour pose significant risks, especially close to launch dates. Early testing procedures of the RTOS code will be conducted to identify and resolve issues before they become critical. Regular consultations with project supervisor Dr. Glenn Matthews provide expert guidance on complex issues. Debugging protocols, including RTOS live debugging tools, systematic code reviews, and peer testing, are in place to address any problems, reducing the likelihood of significant delays.