ENGR 301 Project 12 Project Proposal and Requirements Document

Mohammad Al-Rubayee, Ciaran King, Nicholas Lauder, Ikram Singh, Yee Young Angus Tan, Angus Weich, Jesse Wood

Table of Contents:

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Product Overview
        - 1.3.1 Product Perspective
        - 1.3.2 Product Functions
        - 1.3.3 User Characteristics
        - 1.3.4 Limitations
2. References
3. Specific Requirements
    - 3.1 External Interfaces
    - 3.2 Functions
    - 3.3 Usability Requirements
    - 3.4 Performance Requirements
    - 3.5 Domain Model Requirements
    - 3.6 Design Constraints
    - 3.7 Software System Attributes
    - 3.8 Physical and Environmental Requirements
    - 3.9 Supporting Information
4. Verification
    - 4.1 External Interfaces
    - 4.2 Functions
    - 4.3 Usability Requirements
    - 4.4 Performance Requirements
    - 4.5 Domain Model Requirements
    - 4.6 Design Constraints
    - 4.7 Software System Attributes
    - 4.8 Physical and Environmental Requirements
5. Development Schedule
    - 5.1 Schedule
    - 5.2 Budget
    - 5.3 Risks
6. Appendices
    - 6.1 Assumptions and Dependencies
    - 6.2 Acronyms and Abbreviations
7. Contributions

\newpage

1. Introduction

Amateur rockets are flown regularly worldwide. These rockets are typically flown using off the shelf rocket motors with widely available propellant reloads. These rockets often exceed the speed of sound and altitudes above 30 km are not unheard of. These rockets are almost never controlled, they are stable due to passive aerodynamic features.

While passively stable rockets are reasonably simple and reliable if well designed, they are susceptible to a variety of disturbances, particularly early in flight. Unexpected winds can cause the rocket to weathercock; flexibility in the launch tower/rail can cause railwhip, imparting a random launch angle to the rocket; the thrust from the rocket motor is also never perfectly symmetrical.

The stabilizing response afforded by active control provided changes depending on the measured state of the rocket. While it requires a large overhead of weight and complexity; active control can provide course correction that passive stabilization is not able to achieve.

Client

Andre Geldenhuis is the client for this project. He is a rocket enthusiast and is a member of the New Zealand Rocketry Association. He has had previous experience with building and launching rockets. Contact email: Andre.Geldenhuis@vuw.ac.nz

1.1 Purpose

The purpose of the avionics package is to provide active control for aerodynamic launch vehicles to launch without the aid of a launch rail.

1.2 Scope

This avionics package

- Shall provide telemetry broadcasting over a wireless channel.
- Shall provide location broadcasting over a wireless channel.
- Shall provide data logging for diagnostics and analysis.
- Is to be used on rocket vehicles which use gimballed solid fuel motors to provide active control.

1.3 Product overview

The following subsections will explain the perspective, functions, user characteristics and limitations of the product.

1.3.1 Product perspective

The project aims at solving the problem of getting a rocket to an aerodynamically stable speed without the need for a launch rail. Without a launch rail, a rocket could take off from a much more diverse set of locations such as a high altitude balloon or boat providing the model rocket community with new options for launches previously not viable. By presenting this project as open source it will further open new possibilities for the

model rocket community, this will allow any members of the community to tweak the design for different hardware or new concepts not yet imagined.

As an open source project, the final designs and software will be freely available for anyone to view and edit as they wish. This adds longevity to the project as it can be improved upon for as long as there is interest from the community.

1.3.2 Product functions

To meet the requirements of the minimum viable product the avionics package will:

- Record all sensor data from the launch vehicle as it performs an aerodynamically stable launch from a launch rail.
- In the last second of the motor's burn the avionics package will turn the gimbal to full extension one direction to perform a kickover. This will demonstrate that the control system is capable of changing the rocket's path.
- After the motor has completed its burn, the avionics package will broadcast telemetry including location over a wireless channel to be received by a base station on the ground.

The final avionics package will function beyond the minimum viable product and will:

- Provide control for aerodynamically stable launches without a launch rail.
- Provide support for parachute deployment.

1.3.3 User characteristics

Per client request, the avionics package willl be open source which would imply that the wider model rocket community can use the product. The user should have an understanding of the methodology used in applying the package to their specific needs. This may vary but for the general case, a rocket is the main application. As this is an open source product, educational level is not confined to a certain range since using the package is based on understanding the package operations as well as its application. However, technical expertise is needed when installing the different components and understanding how they function within the system. Technical expertise is also needed with the telemetry as the systems telemetry characteristics vary for different applications. Furthermore, users need to be familiar with the way the software interacts with the hardware.

Experience is important as the Avionics package is primarily applied to aircraft systems such as rockets. The experience required includes but is not limited to:

- Dealing with Civil Aviation Authority (CAA) regulations
- Operating aircraft applications
- Knowledge of telemetry operations

Users operating the package physically, should not have any physical and/or mental disabilities due to health and safety issues.

1.3.4 Limitations

While inflight the rocket will send telemetry data to the base station. This will be limited in terms of signal range and strength by steep hills and terrain will limit the signal range and may lead to errors in the satellite navigation system readings due to data loss and/or interference from other signals.

The timing between the sensors and the microprocessors is also a limitation since the microprocessor and sensors at times may not be operating at an appropriate frequency. The microprocessors speed at the time may be too fast for computing the data that the sensors are acquiring.

2. References

[1] Civil Aviation Authority of New Zealand, Civil Aviation Rules Part 101 Gyrogliders and Parasails, Unmanned Aircraft (including Balloons), Kites, and Rockets – Operating Rules, 10 March 2017. [Online]. https://www.caa.govt.nz/rules/Rule_Consolidations/Part_101_Consolidation.pdf. [Accessed April. 4, 2018].

[2] New Zealand Legislation, Radiocommunications Regulations 2001, 10 September 2001. [Online]. http://www.legislation.govt.nz/regulation/public/2001/0240/latest/DLM71513.html. [Accessed April. 7, 2018].

[3] Quora, What is thrust vectoring, and how is it helpful?, 10 April 2017. [Online]. https://www.quora.com/What-is-thrust-vectoring-and-how-is-it-helpful [Accessed April. 8, 2018]

[5] GNU Coding Standards: Writing C. [Online]. https://www.gnu.org/prep/standards/html_node/Writing-C.html [Accessed April. 8, 2018]

3. Specific requirements

This section will describe the external interfaces, functions, usability requirements, performance requirements, domain model requirements, design constraints, physical and environmental requirements and software system attributes of the avionics package.

3.1 External interfaces

Radio Antenna

The radio antenna will provide small amounts of data to be transmitted from the avionics package to a nearby computer for locating the rocket and minimal debugging. The antenna will transmit the satellite navigation system location of the rocket to allow retrieval should the rocket get lost. Per client requirements, small amounts of data about the state of the avionics package will also be sent including the state of the gimbal and output of the Inertial Measurement Unit. The information will be sent as a set of numerical values quantizing the longitude and latitude as given by the satellite positioning system and a value for the power given to each servo controlling the gimbal so that it's exact position can be determined.

SD Card

An SD card will be present as part of the onboard avionics package. As requested by the clients, the SD card will store in a text or CSV file the outputs of the IMU and satellite navigation system, as well as the position of the gimbal for each control loop. The information will be available for later diagnosis and analysis. The write rate to the SD card must be capable of matching the rocket guidance control loop without missing any data or negatively affecting the processing speed of the microcontroller.

Inertial Measurement Unit

The inertial measurement unit (IMU) will measure changes in the linear acceleration and rotational rate of the avionics package. Output data from the IMU will be stored on the SD card. The IMU data will be used to

control the positioning of the gimbal.

Gimbal

It is required that the avionics package os capable of self correction while in flight. Therefore, the gimbal on the rocket engine nozzles shall act to change the position of the motor nozzle which in turn will change the direction of thrust relative to the center of mass of the rocket.[3] This is the definition of thrust vectoring. Servos will alter the thrust direction by altering the pitch or yaw of the motor. The on-board Teensy shall provide specific computational instructions on the required changes to the positioning of the nozzle.

Servos

Gimbal servos will be receiving computational instructions from the Teensy to alter the rocket's course by changing its pitch or yaw during its boost phase. One servo acts to change the pitch of the nozzle; the other acts to alter its yaw. Inputs to the gimbal interface specify the amount the servo must rotate in degrees. Changes in the positioning of the nozzle and thus the orientation of the rocket are the output.

3.2 Functions

Stakeholder requirements

The avionics package requirements are as follows:

- Avionics package shall fit entirely inside a 29mm airframe rocket vehicle.
- Avionics package shall interface with the gimbal.
- Avionics package shall guide the rocket on course.
- Avionics package shall stabilize the rocket so that it stands in an upright position while preparing to launch.
- Avionics package blueprints and information shall be published on the client's GitHub as open-source.
- The avionics package logical database will be human readable in the form of a text file.

Use Cases

The purpose of the avionics package is to guide the rocket vehicle and for this to occur, it needs to be installed, configured, and calibrated before it is launched. The user(s) may wish to look at diagnostic information that is produced from the flight. When the client is delivered the avionics package, the typical usage of the product would be: Installation, Calibration, Launch, and Diagnosis.

Installation

The avionics package shall be easy to install and uninstall. This requires that

- Hardware shall be packaged into a minimal number of pieces for spatial efficiency.
- Hardware pieces shall have minimal connections for seamless installation and minimal interference.

Configuration

- The avionics package should be easy to configure in terms of setting up the software and/or hardware required. It should take 10 minutes to set up the avionics package as well as preparing the necessary

software.
- The avionics package shall support calibration data pre-loading (i.e data is loaded into the microprocessor prior to a launch).

Calibration

The avionics package should be easy to calibrate and provide feedback on how it is calibrated based on in-flight data. The SD card contains all of the information required for the calibration process which is easily accessible. Calibration in the simple sense is using the data acquired to readjust the sensors/device's ability to make precise measurements. Sensors or any device do become imbalanced over time which will make the data produced unreliable, hence calibration is a crucial part of making these measurements accurate and precise.

Launch

Once launched the avionics package will control the gimbal to maintain an upward trajectory while under powered flight. When the rocket has reached its apogee the avionics package will take any steps needed to avoid damaging the rocket upon landing such as deploying a parachute.

Diagnosis

The avionics package shall store all available data from each flight for later diagnosis.

- Avionics package shall broadcast basic positional data via the wireless antenna.
- Avionics package shall write all measurements, instructions and gimbal positions to an SD card in a readable format.
- Avionics package shall receive any debugging data sent from the user.

3.3 Usability Requirements

Goal

The goal is to create an avionics package that can be applied to any aircraft application which will be easily accessible to the wider avionics community due to it being open source. In this context, the aircraft application is a rocket launch vehicle.

Scenarios

Launch and diagnostics are the two main scenarios for the avionics package usability.

Launch

It is required that the rocket (which is implementing the avionics package) is ready within 10 minutes of the system being activated.

Diagnostics

During a flight, the data from the gyroscope, inertial measurement unit, etc. are constantly being written to the onboard SD card. This information can be used to diagnose any issues or factors regarding the rocket.

Performance

The avionics package should have an efficient means of in-flight correction through the means of correcting the gimbal positioning which is to be done on a frequent basis.

3.4 Performance requirements

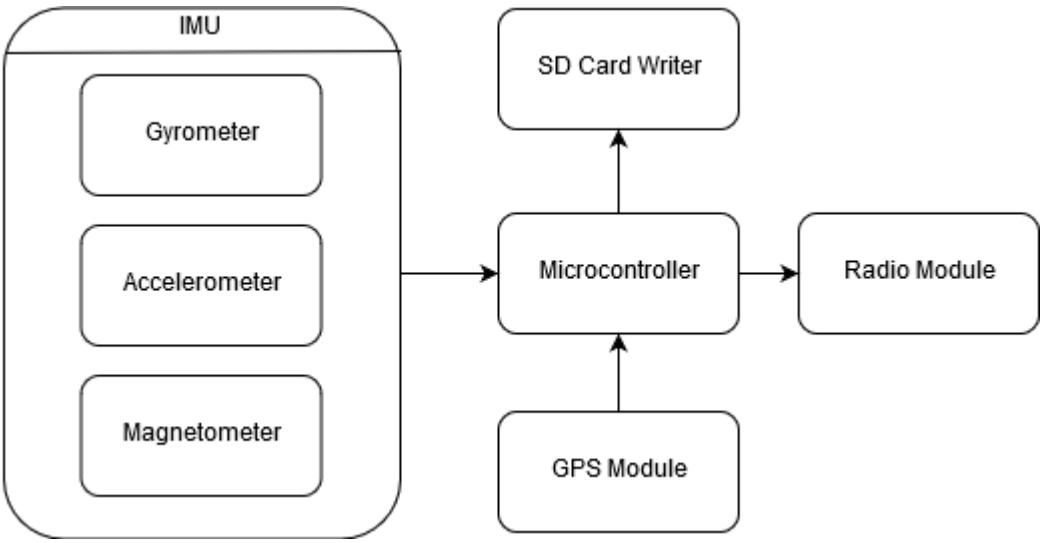
Performance requirements define the base expectations of the rocket and what a user can expect from the package.

Due to the speed at which the rocket is expected to travel, the avionics package needs to be able to take measurements from the inertial measurement unit at least once per meter traveled to allow the avionics package to properly control the rocket. This allows for a sufficient reaction speed of the gimbal to counter any exterior forces such as wind, as well as general balance while flying. Any slower than once per meter traveled will result in the software reacting to old sensor information and not effectively balancing the rocket. The satellite navigation system needs to be accurate to within ten meters in order to make locating the rocket possible once it has landed. The radio needs to have a range of at least the 80 meters without line of sight. 80 meters is the maximum expected range the rocket should travel horizontally from launch base but this could be behind the natural landscape. Even when landed the rocket must still be able to communicate with the base station and report its location.

In order to function adequately for the intended purpose, the avionics package must be durable enough to maintain structural integrity throughout the strain of high-velocity travel as well as the impact of landing multiple times. This is to ensure that the package is cost effective for any potential rocket enthusiasts. The components used will need to survive an acceleration in excess of 200 meters per second per second for the duration of the flight. These components both electrical and structural, must maintain their position and shape on the rocket. If the components do not survive the flight, they pose a risk of damaging components and drastically affecting the stability and predictability of the rocket. It is expected that the structural integrity of the rocket will not deteriorate to a point where the rocket is unsafe to fly for at least ten flights.

3.5 Domain Model Requirements

The domain model of the software system describes the connection of different software components.



Microcontroller Driver

Handles the general initialization of the system. Responsible for communicating with each component, as well as storing data before it gets used.

Inertial Measurement Unit Driver

Reads from the Inertial Measurement Units. This allows the Microcontroller to access data about the accelerometer, gyro, and magnetometer. The data these modules provide is vital to the operation of the control system of the rocket.

Satellite Navigation System Driver

Connects to the satellite navigation module. Feeds location data to the microcontroller to be used as required. These coordinates are required to allow the rocket to be located once it is grounded.

SD Card Writer

Logs the data generated by the various sensors on the rocket. This stores all the sensor data onto the onboard SD card allowing for further analysis when the rocket is located. The data logged includes timestamps, accelerometer values, gyroscope values, magnetometer values, the satellite navigation system coordinates, velocity, height, gimbal positions.

Radio Driver

The radio is used to communicate wirelessly from the rocket to the base station. The main purpose of the radio module is to relay the satellite navigation system coordinates to the users. Without this, the users would not have any location data to help with locating the rocket after it has landed. The radio module is also vital in sending commands from the base station to the rocket. Being able to execute these commands allows the users to execute safety actions in case of any unforeseen circumstances.

3.6 Design Constraints

The design constraints define the restrictions placed upon the design of the software and hardware systems by either government regulations or client requirement.

Radio Antenna and the Satellite Navigation System Unit

Use of the radio antenna and the satellite navigation system must conform to all notices pursuant to Regulation 9 of the Radiocommunications Regulations 2001[2]. Which specifies the available frequencies for use with particular devices and purposes. By conforming to these regulations, the package will not be acting illegally and will be used in an efficient way. The radio antenna must also use frequencies with least use to avoid interference with nearby devices. Interference with other devices is likely to result in corrupted data which could affect the satellite navigation system location of the rocket, resulting in a prolonged search effort. The satellite navigation system unit used must use the legally available satellite navigation frequencies for New Zealand as laid out in Regulation 9 of the Radiocommunications Regulations 2001[2].

Open Source

In order for the final product to qualify as open source, it must be developed using only open source resources. No closed source software or code bases may be used unless unavoidable. When completed all

source code from the project must be released with the final avionics package in accordance with the open source definition.

Physical Construction

In accordance with the New Zealand Civil Aviation Authority, the rocket at its maximum size must satisfy the following criteria [1]:

- The rocket cannot use more than 125g of propellant.
- The rocket cannot produce more than 320 Newton-seconds of impulse.
- The rocket must use a slow burning propellant.
- The rocket is made from lightweight materials such as plastic, rubber, wood, etc.
- The rocket must not have any part of its body fabricated from metal.
- The rocket must not exceed a weight 1.5kg.
- The rocket must not use an aerial firework as an ingredient to create its own jet

3.7 Software system attributes

Software system attributes describe the most important attributes that will be present in software aspects of the project.

Open Source

As this project will become an open source avionics package, it is vital that any rocket enthusiast could pick it up and apply it to their own needs. Being open source is the most important aspect of this project. In order to achieve this, all aspects of development must also make use of open source technologies. Such as Eclipse and open source tool-chains such as avr-gcc for software development, KiCad for circuit board design, Onshape for 3D Modelling and Open Rocket for rocket simulation. By using open source technologies, we can allow anyone, with any amount of resources to make effective use of our avionics package.

Deployment

Ease of deployment is a large aspect of our project as we require the simplest possible process for building the software, programming the microcontroller and running the finished project. Having this process as smooth as possible minimizes downtime and allows issue diagnosis be as easy as possible. Having a smooth deployment cycle allows others users to pick up the avionics package and apply it to their own system by following a clearly laid out deployment process. This will give the package a reputation for being beginner friendly and easy to use.

Fault tolerance and Reliability

As we are dealing with rockets, fault tolerance and reliability are crucial factors in the operation of the system. These attributes have the ability to affect the safety of people, property and the rocket itself. Having the software system tolerant to faults allows the rocket to continue to operate successfully. A typical fault that may take place includes an incomplete/corrupted data between components that could potentially throw off the control software. This could result in a case where the controllers on board are drastically changed by an extreme value that, in terms of a rocket, take too long to correct themselves. By writing fault-tolerant controllers, the rocket can withstand unexpected circumstances and still operate as expected and safely.

Reliability ties in with fault tolerance as we should never be relying on our fault tolerance to ensure the correct operation of the package. We must design the software in such a way that data can be reliably transferred and stored throughout the system. Problems are likely to arise when data is being stored, as there is a potential for variables to be overwritten before they have finished being used. Following typical good software practices can minimize this risk. Regardless, fault tolerance will only be used as a backup and will never be implemented in such a way that is relied upon. When available, error checking will be taking place when data is being transferred. But performance is a key aspect and will be taken into consideration.

Safety

Safety is of the utmost importance when dealing with the avionics package. Other than typical fault tolerance, there a number of unforeseen circumstances that can take place outside of the software's control. Physical faults of the rocket cannot be controlled by the software but can be mitigated in order to pose the least threat to people and property. With the range of sensors available to the package, physical defect needs to be detectable and actions need to be taken accordingly. Actions that can be taken include, but not limited to, deploying the parachute, locking motor gimbal and with an increased focus on broadcasting satellite navigation coordinates. These actions have potential to be actioned automatically, when the package deems appropriate, or by an external command sent to the rocket by the base station. Having these safety aspects built into the package allows for confidence that if there are unforeseen events, steps can and will be made to ensure others safety.

Performance

Dealing with such high speeds with a rocket, a motor gimbal will need to be operating as fast as possible in order to ensure the reliability and predictability of the rocket. If the software cannot react fast enough to rapid changes in direction and speed, then the rocket is likely to spiral out of control. When building the software system, the efficiency and speed of the processes are a vital aspect. This can be difficult in such safety reliant systems as typically ensuring reliable communications requires time and resources. Having the most efficient code can also sacrifice readability, but this can be mitigated with effective documentation of the avionics package.

Programming Language

All the software in the avionics package will conform to GNU Coding Standard[5]. As the software system will be developed by a team, it is important to have a consistent formatting style. It should not be possible to distinguish who wrote which section of code by the style used. The language used for this software system will be C. C is a low-level language which is extensively used in embedded systems. Because of this language's extensive use in embedded systems, there is a significant amount of community support and development around the type of systems we will be using. Allowing for a potential user of the avionics package to have a well-documented environment to work in. Using the GNU Coding Standard[5] allows the code base to be very familiar to a wide range of users as it is an industry-accepted coding standard.

Testing

As avionics package is a safety-critical system, the team will be using a test-driven development strategy. Using this strategy encourages the principle of developing unit tests first, then writing modules that conform to these tests after. Unit tests are very basic tests that check small components/modules of the code. For example, checking that modules handle input/outputs correctly or that a module handles invalid data

correctly. For a more overall test security, acceptance testing will be used to test the functionality of entire modules in expected and unexpected behavior. By using this strategy the team can ensure that changes made to any one component and not propagating through the software system. Allowing the team to focus on the development cycle rather than on checking their own code. Having unit and acceptance tests included in the avionics package sets a precedent for clean/functional code and allows potential users to make changes as they require, but also know that their changes are functioning properly.

Other key software aspects include:

- Adaptability
    - Writing software that allows for modification and additions without causing significant refactoring
- Documentation
    - Having the system well-documented lets others pick up and understand the processes taken.
- Maintainability
    - The code is tolerant to updates and changes. Does not have any major reliances on any other sections of code.
- Quality
    - High-quality code means the code sets out clear processes, has well-defined modularity between components and lets others pick up and package and modify it with reasonable ease.
- Reporting
    - The system has support for extensive data reporting/logging. Used for diagnosis.
- Throughput
    - There is a low latency from the sensors to the physical modules used to either send or store data.

3.8 Physical and Environmental Requirements

The following criteria in regards to the weight, volume and dimensions:

- Weight: 60g excluding the C-motor; 79g including the C-motor and framework surrounding it.
- Dimensions: The design of MVP 1 is of a broad cone shape with r = 9cm, h = 9cm, where r is the cone's radius and h is the cone's height.
- Inside the cone, there is the holder which encapsulates the C-motor and is similar to that of a cylinder with r = 0.1cm and h = 7cm.
- Volume: MVP Stage 1 includes only the rockets first stage which has an approximate volume of 763.41 cm^3^.

Construction of the rocket and its individual parts will take place within the laboratories at Victoria University of Wellington. The physical components required are already in existence (from previous trials) however further testing will be required for validatory purposes as well as further purchasing of new components if the current ones are not up to par. The physical components involved with the rocket are as follows:

- C-motor: Low power rocket motor capable of delivering 14.1 Newtons of thrust.
- Teensy 3.2: Microcontroller on board the rocket. Used for controlling the gimbal.
- MPU 9520 IMU: Located within the rocket, connected to Teensy.
- RFM96W Radio: Located at the base station for receiving any information sent by the rocket.
- Teensy 3.6: Microcontroller used at the base station.
- A2200A GPS: Located on the Teensy 3.6.

Beyond the physical construction of the rocket, the avionics package is also required to collect data from the satellite navigation system and transmit it with a radio antenna via wireless channels to be used for research and recovery purposes. This comprises of a custom made PCB that holds the Teensy 3.6 microcontroller, A2200A GPS and RFM96W Radio which is accessible using a computer via micro USB cable.

The rocket is required to be operable under all of the environmental conditions listed under 9.4.11 of the IEEE standards [3], as well as complying with the CAA weather regulations (specifically subpart D of CAA Part 101 rules) [1]. The regulations state that:

- A rocket shall not operate on or within 4 km of an aerodrome boundary.
- A rocket shall not be flown if cloud coverage is extensive (implies that rocket should not be flown into a clouded area) as well as if visibility is less than 8km.

The rocket will have a control system which pivots the motor using a gimbal to stabilize the rocket, therefore it is ideal to have non-windy conditions for operation. However, if the average wind speed is less than 30km/h then the rocket is still operable due to the control system set in place. If stronger winds persist, then the rocket should not be flown.

Client Requirements:

Faculty of Engineering and Computer Science

The faculty requires that all aspects of the project are properly licensed.

Critical Success Factor

It is vitally important to the faculty that the project also follows all guidelines set out by them.

Residents of Wellington Region

Critical Success Factor

It is vital for the residents of Wellington that the avionics package is reliable and conforms to all relevant safety regulations and guidelines from applicable organizations and ruling bodies. This is because without proper safety features the avionics package could pose a threat to nearby residents.

Wider Rocketry Community

In order to benefit the wider rocketry community, the project is required to be open source so that it can be used and edited by anyone under the open source license. If the package is not open source the potential uses by the community are drastically reduced as it cannot be changed to suit different projects.

Local Council

The local council (Wellington City Council) requires that the rocket can operate without violating any laws or regulations governing the use of model rockets in the area. It is also required that during the operation of the rocket there is no damage to property or persons linked to the use of the rocket.

Civil Aviation Authority

Critical Success Factor

The CAA requires that any rockets conform to all regulations set out in CAA Regulations Part 101 [1].

Operational Environment

- The rocket shall not launch in high-speed winds (30 kmph).
- The rocket shall not launch at night.
- The rocket shall not launch if cloud coverage is greater than 50%.

3.9 Supporting information

The avionics package will add to the body of knowledge within the model rocket community. The project aims at solving the problem of getting a high powered rocket to an aerodynamically stable speed without the need for a launch rail. Without a launch rail, a rocket could take off from a much more diverse set of locations such as a high altitude balloon or boat providing the model rocket community with new options for launches previously not viable. By presenting this project as an open source product, it will further open new possibilities for the model rocket community. This will allow any members of the community to tweak the design for different hardware or new concepts not yet imagined.

As an open source project, the final designs and software will be freely available for anyone to view and edit as they wish. This adds longevity to the project as it can be improved upon for as long as there is interest from the community.

4. Verification

4.1 External Interfaces

The radio antenna can be considered verified so long as it can transmit the satellite navigation system location and have it received by a computer at the base station of the launch. Once the satellite navigation system coordinates have been received and confirmed to be accurate to within 10 meters the radio antenna will have met the requirements.

Once the SD card has been retrieved from the avionics package after a launch and the IMU and control loop information has been written as a human-readable format such as .txt or .csv, it will have met the requirements for data logging. However, if the data is found to be missing due to insufficient write speeds or a flaw in the measurement and control loop the SD card will not have provided useful diagnostic data and will have failed the requirements.

4.2 Functions

The avionics package must be packaged into a single piece excluding the battery pack. This will ensure that the package meets the requirement of being easy to install as only a single part needs installation and the battery pack is provided.

When configurations can be saved as a separate file and be used for multiple flights the avionics package will be considered easy to configure and will have met the requirement for configuration.

Calibration must be performed by the avionics package upon startup. All results of the calibration must be sent via the radio antenna to the base station as well as being written to the SD card for later diagnosis. If

values and calibration data needs to be sent to the avionics package for each launch it will have failed this requirement as it needs to be easy to calibrate without user input.

Once launched, the rocket must maintain an upward trajectory while underpowered flight until it has reached the apogee. It must then land in minimal to no damages to fly again by simply replacing the engine. If the rocket does not maintain an upwards trajectory or breaks upon landing, the avionics package will have failed the launch requirements.

Diagnosis of each launch must be possible given only the data provided by the avionics package to the user. If debugging is not possible due to a lack of relevant data stored on the SD card and sent to the base station via radio frequencies, the avionics package will have failed the requirement.

4.3 Usability Requirements

In order to achieve the usability requirements, the avionics package must guide the rocket upwards despite external factors such as wind. This can be tested by launching the rocket under a variety of circumstances such as varied wind speeds and direction as well as humidity and temperature. To do this the rocket will be launched on multiple days; if the rocket can maintain an upward trajectory despite the external forces, it will have passed the requirement.

4.4 Performance Requirements

The avionics package will have met the performance requirements if it is able to keep to an upward trajectory at high speeds. In order to achieve this, it will need to process inputs from the IMU and output instructions to the gimbal at an appropriate speed. If the performance achieved by the avionics package is not suitable for maintaining the trajectory then it is not fast enough and has not met the requirements.

The physical attributes of the avionics package will have met the performance requirements when the rocket can complete a launch and return to the ground in a state capable of launching again. If it takes more than an hour to repair the avionics package to a state where it is ready to launch, then the physical attributes of the package have not met the performance requirements.

4.5 Domain Model Requirements

Because the domain for the software aspects of the project is fairly simple, there is not a large requirement for their verification. The domain has been designed in accordance with typical software practices and will meet the design constraints if it shows proper modulation and design patterns, as set out in the requirements section.

4.6 Design Constraints

The radio antenna will have met the design constraints given that they operate at a frequency allocated for general use by the Radiocommunications Regulations 2001 document and any relevant notices issued by the relevant authority and/or acting government.[2] The satellite navigation system unit must do the same within the satellite navigation system's allocated frequencies. If the components do not operate within the allocated frequencies they will not have met the requirement.

If any software is used throughout the design process when an open source alternative was available the project will have failed the open source design constraint. To ensure it has met this a list of all software used

during development will be used and their open source licenses will be checked before any contributions from the software will be added to the project.

4.7 Software System Attributes

As this is a safety critical system, a large amount of testing will need to take place in order to verify that the system meets the requirements.

The open source aspects of the avionics package will be verified by ensuring all software packages, and toolchains used are also open source. To specifically verify this, someone should be able to download the package, edit all aspects, build and then use the package, all without using any proprietary software.

Testing will include a large amount of data collection from frequent launches, recording as much data as possible. This data will primarily be used for development of the control system to allow for smooth operation of a rocket. Witnessing the smooth operation of the rocket, with continuous physical iterations will verify that the software system is performing reliably and at sufficient speeds.

Testing the safety and fault tolerance will be done in a simulated environment. This is because, by nature, it is very difficult to test for unforeseen circumstances. What can be done is to simulate situations that can be predicted to go wrong. The user can create situations such as an unbalanced rocket and passively dropping from a building. While in a testing state, the rocket will be in a more verbose reporting mode to allow the user to analyze how the rocket reacts to different scenarios. Once a test plan has been developed, this will be used to verify the fault tolerance and safety aspects of the rocket.

4.8 Physical and Environmental Requirements

The avionics package will have met the physical requirements when the final package weighs less than 60 grams excluding the motor, and the package fits within the rocket body which is a cone 9cm tall and 13cm wide; fitting in a 29mm airframe.

The rocket needs to be able to operate in up to 30km/h winds with no more than 50% cloud coverage and/or at night while maintaining vertical travel in order to achieve the environmental requirements.

To fulfill the physical and environmental requirements the avionics package will also need to meet the requirements of the Faculty of Engineering and Computer Science. To ensure these requirements are met the project will be checked against the guidelines laid out by the Faculty at all times.

The requirements of the residents of the Wellington region and local council will be achieved and verified when the avionics package can complete all other requirements consistently without any potentially damaging consequences.

The performance of the final package will be checked to ensure that it abides by the guidelines laid out by the CAA in the Civil Aviation Rules Part 101 [1]; if the avionics package does not meet all of these guidelines it will have failed the requirements.

5. Development schedule

5.1 Schedule

Architectural Prototype

The architectural prototype will be completed by 14 May 2018.

Minimum Viable Product

The minimum viable product as defined in Part 1.3.2 of this document will be completed by 15 June 2018.

Further Releases

The final prototype will be completed by 19 October 2018.

5.2 Budget

At the current state, all of the required equipment and tools required to complete the tasks already exist (as well as an ample amount of spare parts). Testing and further analysis will need to be done to validate the state of the equipment and tools. Therefore at the current stage, no further costs will need to be spent unless the analysis results is a negative outcome. If the outcome is negative, then the possible costs for the parts are as follows:

Part	Purpose	Cost per unit
Teensy 3.6 Microcontroller	Microcontroller used at the base station which carries out the computations	As of 5th April 2018, the cost is $40.02 NZD based on a 1USD: 1.37NZD conversion
RFM9x LoRa Radio	433MHz Radio Transceiver	As of 5th April 2018, the cost is $27.30 NZD based on a 1USD: 1.37NZD conversion
Teensy 3.2 Microcontroller	Microcontroller used on board the rocket	As of 5th April 2018, the cost is $27.09 NZD based on a 1USD: 1.37NZD conversion
L7805 Voltage Regulator	A 5V, 1A voltage regulator	As of 5th April 2018, the cost is $0.69 NZD
3D printing	3D printing the rocket models	Cost dependent on design changes
PCB	Creating a PCB for the necessary circuitry	Cost dependent on PCB design and quotation from a company
MPA-154-ANT L1 GPS	High performance embedded GPS	10.12NZD
HKM-282A servo	Servo to the 1st stage. Compact 0.12kg servo.	As of 7th April 2018, the cost is $5.60 NZD based on a 1USD: 1.38NZD conversion
HC2422T 7.5kg servo	High torque, high voltage servo.	As of 7th April 2018, the cost is $33.25 NZD based on a 1USD: 1.38NZD conversion
AAA/AA Battery	Batteries to supply voltage to the required components	Cost dependent on the company where the batteries are chosen from

Part	Purpose	Cost per unit
FUNcube Dongle	Signal Receiver	As of 7th April 2018, the cost is $290.83 NZD based on a 1GBp: 1.94NZD conversion

5.3 Risks

Risk	Risk Type	Likelihood	Impact	Mitigation Strategies
Rocket falls on a person causing injury	Health, and Safety	4	5	All participants will remain five meters from the launch site for the entirety of the flight. All passers-by will be warned of the launch and if the park is too busy (more than ten people) the launch will be delayed until the number of people has decreased or the launch will be canceled.
Engine ignites prematurely and exhaust hits participants causing burns or injury	Health and Safety	3	5	The engine will be kept away from any heat source to reduce the chance of premature ignition. Two safety interlocks will also be used to reduce the chance of a premature ignition.
Rocket engine lands on a flammable surface causing fires	Environmental	3	5	The chosen site is a wide open field with very little flammable material within the range of the rocket. The fire risk for the area will be checked before the launch and the launch will be canceled if the risk is too high (above "high" from Fire Danger Levels). A fire extinguisher will be present for all launches and participants will have working phones in case emergency services need to be contacted.

Risk	Risk Type	Likelihood	Impact	Mitigation Strategies
High wind moves the rocket away from the launch site and the rocket falls on people or property causing damage	Environmental	4	5	The wind levels will be measured before launches and the launch will be canceled if the average wind speed is above 30km/h the flight will be canceled.
Rocket chassis comes apart while in the air and debris falls down causing damage to property or injury to persons	Health and Safety/ Environmental	2	5	The frame will be made from a single 3D printed piece, minimizing the chance of it coming apart. All components will be securely attached to the frame using screws to prevent them from falling off.
Heat, light or small particles ejected from the engine strike participants causing injury	Health, and Safety	3	5	All participants will stand five meters from the rocket launch site which is the recommended safe distance advised by the NZ Model Rocketry Association when using D motors or lower.
Rocket enters aircraft flight path and effects flight causing damage to aircraft or passengers	Legal/Health and Safety	1	5	When using a C motor the maximum apogee is around 60 meters which places the rocket well below the minimum flying height for aircraft. If a larger motor is used the site will be moved beyond the safe range of the nearest aerodrome and a NOTAM will be issued informing all aircraft pilots of the launch in accordance with CAA regulations part 101.
Rocket falls over before it launches and flies into bushes or trees	Health and Safety/Environmental	2	5	A launch rail attached securely to a launch platform will be used to ensure the rocket remains upright until it reaches an aerodynamically stable speed.

6. Appendices

6.1 Assumptions and dependencies

Below is a list of Assumptions and dependencies the avionics package.

- The hardware components are used for broadcasting status information and the satellite navigation system positioning can affect the software dramatically. If they are not soldered properly or they are positioned in the incorrect polar direction, the user will not be able to receive required data values.
- The avionics package depends on the servos to achieve their functions properly. The servos could have a defect in their system that denies them access to the information sent by the Teensy. In this case, it would affect the corrections made by the gimble and could cause the rocket to go off course.
- The avionics package depends on the Teensy to calculate the status information of the IMU and send it through to the servos. If the Teensy fails to do this, the servos will not have the data needed to guide the rocket on its course.
- The avionics package depends on the software to have correct computations. If these are incorrect, incorrect data would be sent throughout the system and cause malfunctions in the rocket.
- The avionics package depends on the physical characteristics of the application. All parts must fit and cooperate with the data given to the gimbal; if they fail to cooperate, the launch requirements will immediately fail.
- The avionics package depends on the radio signal to function properly with no defects. If this radio is not fully functional, then the user will not be able to receive the data information needed. In this case, they will not be able to send any signal to initialize a safety system if and when necessary.

6.2 Acronyms and abbreviations

CAA (Civil Aviation Authority): Responsible for the rules that govern managing the risk of aviation systems.

IMU (Inertial Measurement Unit): Electronic device that is used to measure and report the body's force, angular rates.

NOTAM (Notice to Airmen): Notice issued by Airways NZ to alert aircraft of an event within the airspace

7. Contributions

Name	Contributions/Sections
Mohammad Al-Rubayee	table of contents, 1.1, 1.3, 3, 3.1, 4.*, 6.1, 6.2, spelling and grammar
Ciaran King	1, 1.2, 1.3.1, 1.3.2, 6.2,
Nicholas Lauder	3.4, 3.5, 3.6, 3.7, 4.5, 4.7, git fixing
Ikram Singh	1.3.1, 1.3.3, 1.3.4, 3.2, 3.3, 3.6, 3.8, 3.9, 4.1, 5.2
Yee Young Angus Tan	1, 1.2, 1.3.2 1.3.4, 2, 3.1, 3.2, 3.3, 3.8, grammar, formatting
Angus Weich	3.2, 3.4, 3.6, 3.9, 4.1, 4.2, 4.3, 4.4, 4.6, 4.8, 5.1, 5.3
Jesse Wood	3.7, spelling and grammar