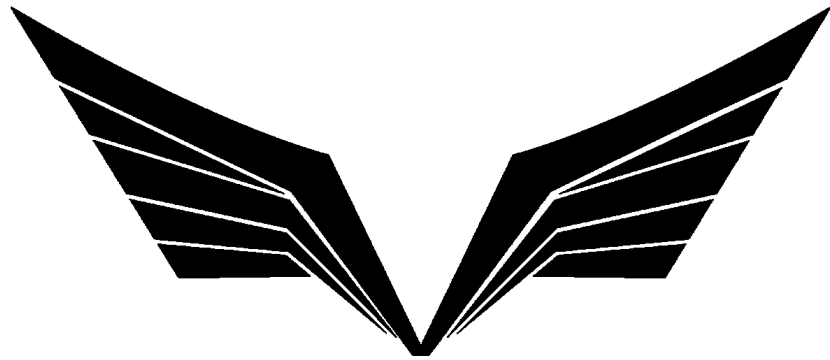




S U N D E V I L

R O C K E T R Y

Avionics Firmware Development Team

Flight Computer (Rev. 2)
System Requirements Document

Current Release	2.7.1
Project Engineer(s)	Nicholas Armistead
Firmware Sublead(s)	Nicholas Armistead
Avionics Lead(s)	Eli Sells

TABLE OF CONTENTS

1. OVERVIEW:	3
2. HARDWARE REQUIREMENTS:	4
3: FIRMWARE REQUIREMENTS	5
4. REVISION HISTORY:	6

1. OVERVIEW:

The All Purpose Primary Avionics (a.k.a. “APPA”) project seeks to improve the reliability and maintainability of Sun Devil Rocketry’s Avionics firmware. In its current form, the codebase contains a large amount of repeated code between different applications. This causes a number of problems:

1. Any new features in one application typically require a port to another.
2. Any existing features that require updates require similar updates to any clones.
3. Any existing features that are modified across applications require each implementation to be tested separately, increasing the club’s final verification burden before every launch.
4. Implementations of the same feature may vary between applications, making it more difficult to see when just *one* implementation has issues.
5. Interfaces with common modules need to be able to accommodate every implementation. This has, for example, caused SDEC’s complexity to increase significantly as more firmware projects have unique needs.

The first primary goal of APPA is to mitigate these problems at the source; rather than attacking increasingly common bugs individually, the intention is to fundamentally alter the architecture of the flight firmware to make these bugs common by ceasing support for multiple applications with cloned functions in favor of a common firmware application.

The second primary goal of APPA is to decentralize the application. In much of the flight firmware, the bulk of the application sits in the same file as the `main()` function. This was acceptable when the firmware was a far smaller project. But today, SDR flight computers are directly responsible for flight dynamics/control, serve as the primary data loggers for some flights, and are otherwise *significantly* more complex than they were previously. This complexity has revealed the need for more exhaustive testing, both at the unit level and above. In its current form, the avionics team is unable to perform unit testing on any function in the `main.c` file. By moving the flight and terminal portions of the application to their own functions and files, the maximum cyclomatic complexity of the application has been significantly reduced, allowing the team to tackle unit testing on the app layer rather than just the mod layer like before. It also makes the code significantly more readable, as the highest level operation of the application is able to be abstracted significantly more than in previous iterations.

2. HARDWARE REQUIREMENTS:

- 2.1: The system shall be designed for operation on an STM32-based microcontroller.
- 2.2: The system shall have a method to report its angular and linear acceleration.
- 2.3: The system shall have a method to report its GPS coordinates.
- 2.4: The system shall have a method to report barometric readings.
- 2.5: The system shall have a method to connect to an outside computer.
- 2.6: The system shall have a method to record data to non-volatile memory.
- 2.7: The system shall have an external switch to facilitate arming the system.
- 2.8: The system shall have a method to provide basic operational information to the user, including:
 - 2.8.1: A visual method.
 - 2.8.2: An auditory method.
- 2.9: The system shall have a method to control four servos.
- 2.10: The system shall have a method to activate ejection charges.
- 2.11: The system shall have a method to accept power over USB.
- 2.12: The system shall have a method to accept power from a battery.
- 2.13: The system shall have a method to send wireless radio signals.
- 2.14: The system shall have a method to restart the application.
- 2.15: The system shall have a method to upload new firmware.

3: FIRMWARE REQUIREMENTS

- 3.1: The system shall utilize the Sun Devil Rocketry “lib”, “mod”, and “driver” libraries to interface with flight computer hardware.
- 3.2: The system shall have an option to log flight data.
 - 3.2.1: The data that is logged shall be configurable.
 - 3.2.2: The data logging shall begin when launch has been detected and end when flash is full.
 - 3.2.3: The data logging shall have a method to reduce the sampling rate and extend the period that data is logged.
- 3.3: The system shall have a methods to detect if launch has occurred, including:
 - 3.3.1: A method based on barometric pressure.
 - 3.3.2: A method based on acceleration.
- 3.4: The system shall calibrate the IMU and barometer before launch.
- 3.5: The system shall have an interface to communicate with SDEC.
- 3.6: The system shall change modes from the SDEC interface to the pre-launch mode when the external switch is activated.
- ~~3.7: [PLANNED/POSTPONED] The system shall have an option to send commands to servos that change fin angles in order to stabilize the vehicle in flight.~~
 - ~~3.7.1: [PLANNED/POSTPONED] The axes of control shall be the following configurable options:~~
 - ~~3.7.1.1: [PLANNED/POSTPONED] Roll~~
 - ~~3.7.1.2: [PLANNED/POSTPONED] Pitch/Yaw~~
 - ~~3.7.2: [PLANNED/POSTPONED] The maximum deflection angle of the fins shall be configurable.~~
 - ~~3.7.3: [PLANNED/POSTPONED] The system shall be capable of roll control with two or four servos.~~
 - ~~3.7.4: [PLANNED/POSTPONED] The system shall be capable of pitch and yaw control with four servos.~~
 - ~~3.7.5: [PLANNED/POSTPONED] The system shall support two and four servo modes without requiring configuration changes.~~
- ~~3.8: The system shall have an option to deploy a drogue and main parachute at apogee.~~
 - ~~3.8.1: [POSTPONED] The terminals used for dual deploy shall be configurable.~~
- 3.9: The system shall have a number of configurable parameters that can be changed without recompiling the application.

-
- ~~3.10: [POSTPONED] The system shall be able to send a signal to onboard cameras using the dual deploy terminals.~~
- ~~3.10.1: [POSTPONED] The system shall operate the cameras under one of the following modes based on configuration:~~
- ~~3.10.1.1: [POSTPONED] When a wireless telemetry signal is received.~~
 - ~~3.10.1.2: [POSTPONED] When launch is detected.~~
 - ~~3.10.1.3: [POSTPONED] When calibration has completed.~~
- ~~3.10.2: [POSTPONED] Full dual-deploy and the camera signaling system shall be mutually exclusive.~~
- ~~3.10.3: [POSTPONED] The terminals used by the camera signaling system shall be configurable.~~

There is an open hardware problem report for the following functionality, which is not able to be verified due to this issue. It is thus currently not supported in v2.7.1. We will do a branched release once the hardware is fixed if any software changes are needed, or otherwise we will verify the functionality after release.

- ~~3.11: The system shall be able to wirelessly broadcast vehicle data to observers on the ground.~~
- ~~3.11.1: The wireless system shall be configurable at runtime.~~
 - ~~3.11.2: The wireless system shall run without blocking the main flight loop.~~
 - ~~3.11.3: The wireless system shall broadcast messages with the following data:~~
 - ~~3.11.3.1: Messages containing vehicle state data, including position and orientation.~~
 - ~~3.11.3.2: Messages containing vehicle identification data, such as the firmware and hardware IDs and a configurable flight identifier.~~
 - ~~3.11.3.3: Messages containing software calibration data, such as the IMU offsets and barometric calibration offsets.~~
- 3.12: The system shall calculate an orientation estimate based on sensor data.
- 3.12.1: The orientation shall be represented by a Hamilton unit quaternion.
 - 3.12.2: The system shall fuse data from the following sensors to determine orientation:
 - 3.12.2.1: Gyroscope
 - 3.12.2.2: Accelerometer
 - ~~3.12.2.3: [POSTPONED] Magnetometer~~

4. REVISION HISTORY:

0.1 - Initial Revision (For APPA PDR).

2.5 - Initial Release of APPA.

2.6 - Updates for new release.

2.7 - Updates for new release.

2.7.1 - Update for branched release.