University of North Dakota Advanced Rocketry Club

Avionics Team

Preliminary Design Review



Contents

[Overall System Architecture 3](#_Toc1653101)

[Electronics and Altitude Monitoring 3](#_Toc1653102)

[Avionics System Overview 4](#_Toc1653103)

[Position Monitoring 4](#_Toc1653104)

# Overall System Architecture

Our avionics system will be a low-budget, yet capable and safe platform. Since our exposure to harsh environment time is very short, we can reasonably deploy commercial off-the-shelf (COTS) hardware at a very low cost and low weight. For example, the Raspberry Pi (RPi) and BeagleBone Enhanced (BBE) single-board computers (SBCs) are both about the size of a credit card and weigh less than a typical smartphone. Both are fully featured Linux computers, and the BBE boasts a gigahertz processor with 2 embedded programmable real-time units (PRUs) along with an onboard barometer and 6-axis accelerometer.

Our responsibilities include active management of the engine and its systems, data acquisition, telemetry, abort systems, and recovery device deployment. Our rocket will not be using any form of guidance, navigation, and control (GNC) systems due to competition restrictions. We will achieve each of these requirements with appropriate hardware as described in upcoming sections.

# Electronics and Altitude Monitoring

While we’re still in the research phase of our design, we have several options being considered for data acquisition and position determination. Some of the solutions we’re considering include the Global Positioning System (GPS), using the Doppler effect to measure speed, Inertial Measurement Units (IMU), and more. Since we’re still so new to the design requirements and have a relatively tight budget we haven’t yet settled on a finalized design; however, we have a reasonably structured draft at this time and it will be presented below.

## Avionics System Overview

There are many parts that combine to form one avionics system. As outlined in our responsibilities above, we will have one or more subsystems to fulfill each of those requirements.

### Position Monitoring

Keeping track of the rocket is vital for multiple reasons, primarily safety. Knowing where the rocket is laterally and vertically allows mission controllers to quantify any risks that may be present and making an educated abort decision at any time. It also allows us to be certain that we’ve met our goal of reaching 100 kilometers, collect data for research purposes. However, determining where a rocket is can be quite difficult at times since it is not an often-solved problem. We plan to use a sensor-fusion approach to take data from multiple sensors/subsystems and use that data to provide an overall navigational picture.

|  |  |  |
| --- | --- | --- |
| **Type of device** | **What it provides** | **When it’s useful** |
| GPS receiver | Position (3D), velocity (3D) | Within CoCom limits |
| Barometer | Altitude, vertical speed | Approximately bottom third of the atmosphere |
| IMU/Inertial Navigation System (INS) | Acceleration (3D), can be integrated to provide velocity (3D) and position (3D) | Entire flight |
| Doppler effect velocity measurements | Velocity (3D) | Entire flight |

Table 1: Position determination data sources

#### Global Positioning System (GPS)

The obvious solution to target tracking, altitude and velocity monitoring is to use the GPS constellation. There are, however, some problems with this. The United States places export regulations on GPS receiver hardware to prevent foreign powers from using them to build inter-continental ballistic missiles (ICBMs). The result of these export regulations (called CoCom or COCOM, for Coordinating Committee for Multilateral Export Controls) is that the receivers automatically disable themselves outside of certain conditions. Most of the firmware manufacturers disable above 1000 knots OR 18 kilometers, however certain vendors will use an AND gate instead of OR. Unfortunately, any vehicle capable of reaching space (not even orbit, just 100km altitude) will *easily* break these conditions. 1000 knots is slightly under Mach 1.5 at sea level, and 18 kilometers is well short of the desired 100. Even with an AND-gated device, preliminary calculations from our propulsion team indicate that our vehicle will be crossing 18km at approximately Mach 1.57 (1770 ft/sec), disabling the GPS receiver.

There are some reports of devices that can easily have their CoCom limits disabled by a firmware “hack,” or modification, allowing them to continue functioning well outside of those limits. However, since this is a violation of international law, we will not be using this method.

What we can use GPS for is augmentation while within the CoCom limits. We plan on using the GPS receiver to cross-check the INS unit and provide any necessary calibrations on-the-fly. Since it provides 3-dimensional position, we can derive velocity and acceleration (although the latter may take a few seconds to gather enough data). This data provides augmentation and cross-referencing for INS and barometric subsystems up to the limit of 18km. We can use this data to determine errors between the systems, store those differences, and apply them over the rest of the flight.

#### Barometric Altimeter

A basic barometric altimeter has been used in aviation since 1928. Since the atmosphere’s pressure decreases in a predictable manner, we can determine the vehicle’s altitude above mean sea level as a direct function of atmospheric pressure. These functions have been well documented as they’re critical to aviation, and therefore we can implement them directly with very little work. Barometric sensors are cheap and readily available; the MS5803-14BA unit is rated all the way to and including 0 mbar of pressure, and there are breakout boards available for both Serial Peripheral Interface (SPI) and Inter-Integrated Circuits (I2C). In fact, the code to interpret the altimeter data is already available.

There are, unfortunately, restrictions on the altimeter’s use. Since the barometer requires atmospheric pressure to function, it will only be reliable during a portion of the flight. We estimate that it will be approximately the lower third of the atmosphere, however, we also plan on running complete vacuum chamber tests to determine exactly where the barometer becomes inaccurate. Looking at the International Standard Atmosphere (ISA) data that can be found online, it appears reasonable to expect the altimeter to stop functioning accurately at a little above 30 kilometers: its resolution is 0.2 mbar and the expected pressure at 32km is approximately 0.0085bar.

Using a barometer to aid GPS altitude calculations is quite common in aviation; a process called baro-aided receiver autonomous integrity monitoring (RAIM). In this procedure, corrections are made during the flight based on known pressure settings at different airports. From these corrections, a barometer can be used to accurately determine altitude above mean sea level with impressive precision & accuracy. For pilots, this is the primary means of determining altitude, almost to the exclusion of GPS altitude reports.

|  |  |  |
| --- | --- | --- |
| **Metric** | **Barometer (MS5803-14BA)** | **GPS/GNSS (MKT3339)** |
| Size | 20 mm / 18 mm / | 25.5mm / 35mm / 6.5mm |
| Weight |  | 8.5g |
| Breakout board interfaces | 1xI2C; 1xSPI | 1xUART |
| Lateral position measurement accuracy | No lateral position provided | Dependent on available GNSS constellation. Optimum: 0.715m Worst-case: 7.8m |
| Vertical position measurement accuracy | 1mbar in 0.5millisec  0.6mbar in 1.1millisec  0.4mbar in 2.1millisec  0.2mbar in 8.22millisec | Typically, worse than lateral position accuracy. VDOP values are higher due to all measurement satellites being above the receiver. Near impossible to quantify. |
| Altitude performance degradation? | Yes – as altitude increases, barometric pressure derivative steepens | No – as altitude increases, more satellites come into view |
| Altitude Limits | Until lack of precision from pressure gradient | CoCom limits – reportedly OR-gated |
| Power draw |  | Navigation mode: 20 mA |
| Code available for use? | Yes | Yes |
| Data sampling frequency | SPI: 20 MHz  I2C: 400 kHz | UART: 10 Hz (not accounting for NMEA string processing) |
|  |  |  |