University of North Dakota Advanced Rocketry Club

Avionics Team

Preliminary Design Review



Contents

[Overall System Architecture 3](#_Toc4108342)

[Electronics and Altitude Monitoring 3](#_Toc4108343)

[Avionics System Overview 3](#_Toc4108344)

[Position Monitoring 3](#_Toc4108345)

[Global Positioning System (GPS) 4](#_Toc4108346)

[Barometric Altimeter 4](#_Toc4108347)

[Doppler Effect Speed Measurement 6](#_Toc4108348)

[Inertial Navigation System 6](#_Toc4108349)

[Telemetry 7](#_Toc4108350)

[Recovery 9](#_Toc4108351)

[Engine Control and Monitoring 10](#_Toc4108352)

[Software 11](#_Toc4108353)

[Ground Control 12](#_Toc4108354)

[Tracking & Receiving 13](#_Toc4108355)

[Remote Abort System 13](#_Toc4108356)

[Hardware Options 14](#_Toc4108357)

[Computers 14](#_Toc4108358)

[Controllers 15](#_Toc4108359)

[Peripherals 15](#_Toc4108360)

# Overall System Architecture

ARC’s avionics system will be a low-budget, yet capable and safe platform. Since the exposure to harsh environment time is very short, it is reasonable to 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.

The avionics responsibilities include active management of the engine and its systems, data acquisition, telemetry, abort systems, and recovery device deployment. The vehicle will not be using any form of guidance, navigation, and control (GNC) systems due to the complexity of designing such a system and the expected lack of a need for active guidance. Each of these requirements will be achieved with appropriate hardware as described in upcoming sections.

# Electronics and Altitude Monitoring

While still in the research phase, there are several options being considered for data acquisition and position determination. Some of the solutions being considering include the Global Positioning System (GPS), using the Doppler effect to measure speed, Inertial Measurement Units (IMU), and more. What follows is a detailing of how each system will be used to help provide complete situational awareness of and on the vehicle.

## Avionics System Overview

There are many parts that combine to form one avionics system. As outlined in the responsibilities above, there will be 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 the team to be certain that the 100kilometer goal has been met and 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. Therefore, a sensor-fusion approach to take data from multiple sensors/subsystems will be used.

|  |  |  |
| --- | --- | --- |
| **Type of device** | **What it provides** | **When it’s useful** |
| GPS receiver | Position (3D), velocity (3D) | Within CoCom or receiver 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 |
| Ground-based radar | Position (3D), 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 the propulsion team indicate that the 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 (apparently) 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, this method will not be used.

What GPS can be used for is augmentation while within the CoCom limits. The GPS receiver will be used to cross-check the INS unit and provide any necessary calibrations on-the-fly. Since it provides 3-dimensional position, acceleration and velocity can both be derived (although the latter may take a few seconds to gather enough data). This provides augmentation and cross-referencing for INS and barometric subsystems up to the limit of 18km. From here, the data can be used to determine errors between subsystems, store, and apply those differences 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, the vehicle’s altitude above mean sea level can be determined as a direct function of atmospheric pressure. These functions have been well documented as they’re critical to aviation, and therefore can be implemented with very little work. Barometric sensors are cheap and readily available; the MS5803-14BA unit is rated all the way down to and including 0 mbar of pressure, and there are breakout boards available for both Serial Peripheral Interface (SPI) and Inter-Integrated Circuits (I2C) protocols. 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. Current estimates suggest that it will be approximately the lower third of the atmosphere, however, complete vacuum chamber tests will be run 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 54 kilometers: its resolution is 0.2 mbar and the expected pressure at 56km is approximately 0.74mbar. Below that altitude, the derivative of pressure with respect to altitude begins to drop below the minimum resolution of 0.2mbar.

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.

Comparing the barometric altimeter to using GPS altitude, certain restrictions apply to both and when they an be used. The GPS altitude tends to be less precise but will be accurate to a higher altitude. Barometric altitude, on the other hand, will be more precise but less accurate with higher altitudes. By varying the precision read from the altimeter as altitude increases, it should be possible to read accurate and precise pressures up to 185,000ft, or almost 57 kilometers. A snippet from the ISA data table shows the critical part of the atmosphere where the altimeter is no longer precise enough:

|  |  |  |  |
| --- | --- | --- | --- |
| Altitude (ft) | Altitude  (km) | Pressure (millibar) | Δ (pressure@alt / sea-level-pressure) |
| |  | | --- | | 175,000 | | 180,000 | | 185,000 | | 190,000 | | 195,000 | | 200,000 | | 205,000 | | |  | | --- | | 53.34 | | 54.864 | | 56.388 | | 57.912 | | 59.436 | | 60.96 | | 62.484 | | |  | | --- | | 1.098000 | | 0.903700 | | 0.741700 | | 0.606900 | | 0.494900 | | 0.402300 | | 0.325800 | | |  | | --- | | 0.000518700 | | 0.000427000 | | 0.000350500 | | 0.000286800 | | 0.000233900 | | 0.000190100 | | 0.000154000 | |

Since the barometric altimeter is accurate up to these altitudes, it can be used alongside the GPS to calibrate and correct the INS in the same manner.

Table 2: Comparison of GPS vs. barometer

|  |  |  |
| --- | --- | --- |
| **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) |

#### Doppler Effect Speed Measurement

The Doppler Effect is a well-documented phenomenon which causes the frequency of a radio transmission to shift as a function of velocity. For example, if a receiver in the path of a vehicle travelling at high speed is expecting to get a frequency of, for example, 433 MHz, it may in fact experience a frequency 65 Hz higher than that. The formula for calculating velocity given the other variables (initial frequency and received frequency) is available online and easily implementable in code as it requires no integration or derivation and is a constant-time function.

Even better, there likely won’t need to be any additional hardware to take advantage of the Doppler effect to measure speed. The vehicle will be using a radio frequency (RF) link to send telemetry data back, and the Doppler effect speed measurement system can simply run on top of that physical hardware. It will likely be entirely a software product, simply running on the ground control machine to compute vehicle velocity. Since the vehicle does not have any need to know its velocity, there is no need for a return transmission to send that information back to the vehicle, further simplifying equipment requirements.

#### Inertial Navigation System

Inertial navigation systems have been used in high-fidelity aircraft for quite some time. The system relies on a set of accelerometers, preferably 6- or 9-axis. The angular accelerometers are used for determining orientation, by integrating the angular acceleration twice with respect to time to get current orientation. The same idea is applied to the linear accelerometers to determine current position. Since the starting position, orientation, and velocity (rotational and linear) are known, there are no unknown constants left to be solved for.

Further performance increases are given by precomputing the integration formulas and simply plugging in constants. This reduces the operation to constant-time functions, which are easy to compute and can be done in a fixed amount of time no matter the values (within reason). By using a 9-axis unit such as the BNO055, triaxis geomagnetic data can also be acquired, effectively giving the vehicle a 3D compass to use for orientation during the entire flight.

Unfortunately, INS units sometimes suffer from drift errors. Since there are a certain set of constants only known at one point (launch), any error in the accelerometers is further magnified through the double integration. Most units are unique in their errors and therefore additional testing will be required to determine the best corrections to make.

There are several different accelerometer sensor-in-package (SiP) types available for use. The 9-axis BNO055 discussed earlier has the advantage of the geomagnetic sensors, but is a bit more expensive and an external unit. On the other hand, the 6-axis MPU-6050 unit lacks the magnetometer but is much smaller. Another advantage to this unit is that the BBE computer comes pre-packaged with an MPU-6050 unit on the board, linked to the processor already. There are some differences between the two, outlined below.

|  |  |  |
| --- | --- | --- |
| **Metric** | **MPU-6050** | **BNO055** |
| Size | 4mm x 4mm x 0.9mm | 3.8mm x 5.2mm x 1.13mm |
| Weight |  |  |
| Breakout board interfaces | I2C | SPI, I2C |
| Available on-board SBC | Yes | No |
| Sensor package | Triaxis linear accelerometer  Triaxis gyroscope | Triaxis linear accelerometer  Triaxis gyroscope  Triaxis geomagnetic sensor  Sensor fusion |
| Linear accelerometer type | MEMS proof mass |  |
| Gyroscope type | MEMS CVG |  |
| Geomagnetic sensor type | N/A |  |
| Linear accelerometer accuracy | At ±16g, 2048 LSB/g | All modes, 1 LSB/mg |
| Gyroscope accuracy | At ±250 °/s, 131 LSB/(°/s) | All modes, 16 LSB/(°/s) |
| Geomagnetic sensor accuracy | N/A | ±0.3 µT |
| Linear accelerometer sampling rate |  | 100 Hz |
| Gyroscope sampling rate | 8 kHz | 100 Hz |
| Geomagnetic sensor sampling rate | N/A | 20 Hz |
| Power consumption | 3.9 mA | 12.3 mA |
| Code available for use? | Yes | Yes |

## Telemetry

Having a data transfer link to the rocket is of utmost importance for making informed safety decisions on the ground. Engineers need data about engine performance, vehicle position, velocity, and acceleration; all sensor data should be transmitted to ensure that the vehicle is processing the data correctly in real-time. One key difference to understand is that telemetry is one-way: vehicle-to-ground. Ground-to-vehicle, by definition, are not telemetry data but rather a communications link. Since the only signal that may need to be sent to the rocket is an abort command, the communications link can have a much lower bandwidth than the vehicle-to-ground telemetry link, which must be able to transmit all the necessary data.

Since all data is being recorded on the rocket in case the telemetry link fails, it makes sense to simply insert the telemetry output process on the same hardware responsible for recording the data. This simplifies program design, since the telemetry process can listen to the data channels being recorded and pipe it to the radio transmission process. From there, the data is sent through an amplifier and finally to the antenna. The ground control station picks up the signal, processes and records the data as well as displaying it for the ground controllers to see.

Another potential safety feature that can be implemented easily is running a mirrored version of the vehicle abort determination software on the ground. This allows controllers to see immediately if the vehicle is going to/has initiated an abort sequence, and if so for what reason. It may also allow controllers to see a situation where an abort will be necessary in the near future and make that decision earlier than the automated systems would have determined, further improving safety.

As for the radio hardware required, estimates of atmospheric attenuation by distance determine the best frequency band (visible laser, IR, radio) to use. The search range is narrowed down to the 1MHz-1GHz range by finding a large window of atmospheric and ionosphere transparent ranges.

From this point, determining frequencies gets a bit harder. Atmospheric opacity charts for the desired range are very hard to find. Charts for the 1GHz to the 1THz range are common, showing that the limit as frequency goes to 1GHz from the right approaching very low numbers of dBm/km of approximately 0.01. From this point, the best option given the total lack of available data is to perform trials with the appropriate distance.

Since there are a handful of private pilots on the avionics team, one of the tests being planned is a combination test of both RF-link effective distance and tracking & receiving hardware. While that particular element is left to be described later in the telemetry section, it will likely be small enough to mount in the back of a pickup truck. The pilot will take up a passenger with an omnidirectional radio transmitter to be used in the test and fly in a random (but safe) pattern around an unpopulated area. The job of the ground personnel is to track and follow the aircraft, demonstrating the capability of the tracking turret. Once that test has been completed, the ground personnel will remain stationary while the pilot flies directly away until contact is lost. This procedure will be repeated as necessary until a suitable frequency and associated hardware are found.

There are still some bounds on what frequencies can be used. To summarize the 177-page FCC frequency band allocation, there are several interesting bands that may be available to use. Several industrial, scientific, and medical (ISM) bands that are available for anyone to use. Since there are multiple ISM bands, one can be picked that best suits the desired ease of transmission and minimal atmospheric attenuation characteristics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Frequency range (MHz)** | | **Center (MHz)** | **Type** | **Common uses** |
| 6.765 | 6.795 | 6.78 | A | Fixed & mobile service |
| 13.553 | 13.567 | 13.56 | B | Fixed & mobile service, except automobile |
| 26.957 | 27.283 | 27.12 | B | Fixed, mobile, except automobile & CB |
| 40.66 | 40.7 | 40.68 | B | Fixed, mobile, Earth exploration-satellite |
| 433.05 | 434.79 | 433.92 | Region 1 | Amateur service & radiolocation |
| 902 | 928 | 915 | Region 2 | Fixed, mobile except aeronautical & radiolocation |

**Type A:** Requires authorization by concerning administration (usually FCC)  
**Type B:** Must accept harmful interference in the ISM band

Left to right: Regions 2, 1, 3

## Recovery

The main avionics bay will be located near the top of the main body segment. Locating the unit here provides a short link to the sensitive sky-facing GPS antennae while allowing adequate reach to the communications antennae near the rear of the rocket. Running communications links down service tunnels primarily for fuel will allow for a hard-wired communications channel to the engine control unit.

A nominal flight would include burning the engine at full thrust until fuel depletion, then initiating an abort procedure. The abort procedure at any altitude consists of (in order) an immediate depressurization of the fuel tanks, coasting until vertical velocity reaches 0 (apogee), and starting the recovery device deployment loop. The recovery device deployment loop is the program responsible for monitoring the barometric pressure sensor and determining whether to deploy one or both parachutes.

Since the abort procedure is the same as the nominal recovery procedure, the abort signal can simply be used to trigger both an unexpected abort and a normal recovery. This further simplifies program design by reducing overall code size and complexity. By implementing a simple velocity check (or even simpler, an altitude-changing-at-minimum-rate check), on-the-pad aborts can be performed with the same signal. During an on-the-pad abort, the only things to do are to safely shut down the engine (if it’s been turned on) and depressurize the tanks. Ejecting the nosecone and deploying parachutes would not be productive.

Deciding when to abort based on an avionics unit failure of any type is discussed later, in the risk assessment section. However, it is imperative that the recovery procedure does *not* depend on having a functioning communications link – that is, it can be triggered by an on-board abort signal. To do this, a dedicated program will run on both the vehicle and ground control to factor in available information and determine an automatic abort signal.

Because data such as current GPS position and integrated INS solutions are not available to the vehicle but rather on the ground, there are scenarios possible where an abort can be made by ground controllers before the automated system would have determined it. While it hasn’t been set in stone yet, a separate SBC for the abort-decision algorithm is being considered. Since that process could be resource-intensive, having its own dedicated hardware would not only be a safety enhancement but also provide a platform capable of running a more in-depth analysis of all variables and make a more informed, and therefore safer, abort decision.

## Engine Control and Monitoring

The propulsion team is still working to finalize engine design. However, writing general code for the higher-level functions is still possible. An Arduino Nano will control the necessary servos and pumps, communicating with the brain via a hardwired interface, likely UART.

The engine controller would be responsible for exposing high-level function calls to the applicable interface. These calls would be something like setting engine thrust to a certain value or percentage; initiating or shutting down the engine or reading from sensors. Communication would use a serial protocol with the brain unit sending commands to the Arduino, which would respond appropriately both acknowledging the commands and notifying the brain when the actions have completed.

Safety can be improved by implementing a heartbeat protocol between the two devices. If the engine controller stops responding, it can be assumed that it has failed, and the abort procedure should be initiated. The engine controller can also monitor the brain – if the brain stops responding, then the engine should be shut down. This is another argument for dedicated recovery hardware: if the brain fails, the recovery unit would be aware of that and able to initiate the abort sequence safely.

Some initial calculations show the vehicle exceeding the target altitude by a large margin, potentially endangering other space traffic such as satellites. To deal with this, the brain controller will monitor velocity and altitude to determine if the vehicle has enough velocity to overshoot target altitude by a certain margin. If this margin is exceeded, the engine controller will be sent a shutdown or throttle-down command to prevent this from happening.

Flight Dynamics is still determining if the vehicle will encounter a maximum dynamic pressure (Max-Q) requiring the engine to be throttled down. If so, since there’s no way of measuring dynamic pressure on the vehicle (i.e. no pitot tube), timing control must be used to determine the throttle-down point and duration. To satisfy this requirement, the engine control unit (ECU) must be able to control the pumps and servos with enough accuracy and precision to throttle the engine safely.

# Software

In order to get the best and safest process, custom software is essential. It can be tailored to exactly what the engineering teams want/need, allowing faster and safer work. Custom software will be found everywhere from running on the brain of the vehicle to the ground control telemetry processor. It can be made more efficient than already existing software, which does not do exactly what is needed – code that does exactly what it’s designed for is often better that code that can do many things including ARC-specific requirements.

The software can be broken down into 4 main parts: telemetry, engine control, abort decision-making, and recovery. Each of the 4 processes will have their own hardware in one form or another and are described below. Since each of these elements are described in more detail above (per the Base 11 PDR template), only a brief summary of responsibilities & requirements is given below.

* Telemetry: Responsible for sending data back to the ground control station. It must be able to process all the available sensor data onboard the vehicle and format it in a manner that takes up the minimum bandwidth for transmission. While the exact amount of expected data isn’t yet set in stone, packetization and compression are being investigated to weigh the transmission efficiency versus the additional computing power required.
* Full-authority digital engine control (FADEC): Responsible for processing the requests from the brain controller and turning those requests into reality, the FADEC software will be required to run in real-time. The first choice for hardware is the Arduino Nano, which provides a small, lightweight, and familiar environment meeting the real-time requirements.
* Recovery: As discussed earlier, the recovery process will have dedicated hardware to facilitate safe recovery of the vehicle. Primarily altimeter-based and single-threaded, the recovery software must also be run in real-time, making the Arduino Nano again a good option.
* Abort decision-making: There are multiple reasons to have an on-board abort decision. For example, if the telemetry link fails but FADEC and recovery remain, there’s no reason to abort the flight prematurely. A counterexample where abort would be necessary could be if the vehicle begins to rotate about the wrong axis – vehicle destruction could happen very quickly and therefore ground controllers may not have time to send an abort command back. An on-board abort controller could process the rotational acceleration real-time and initiate an abort much faster than a ground-based controller could.

# Ground Control

Ground control software is where most of the calculations will take place. The vehicle itself only needs to know altitude, orientation and engine parameters to facilitate a safe abort in the event of a communications link failure, so the rest of the calculations can be performed by much higher-performance computers on the ground. These on-site calculations will provide a live data feed for the launch team to monitor and make a decision about whether they want to abort the flight or not.

This also allows the vehicle avionics to offload unnecessary stress. The ground control unit (GCU) will consist of a high-powered desktop computer, necessary radio receivers and antennae, a cellular phone modem (unless somehow Internet access is provided via a different method), and a suitably stocked snack drawer. The software running on the computer will be custom-built and designed to give the engineers all available data and system statuses in an easy-to-understand manner.

Current thoughts are that the most efficient system is to set up everything needed in a gutted van. This allows the setup to be mobile, tested at home and known to work anywhere. It can be used for multiple launches, including smaller outreach launches for hardware/software testing, and provides an easy method to transport any necessary electronics.

Certain functionality can be enhanced by having umbilical cables running to the rocket. These cables, although it is not yet known exactly what interface (Ethernet, serial, a custom implementation) is best, would allow a direct line to the brain of the rocket pre-launch and enable safer on-the-pad aborts, launch control, and a much faster telemetry line for engine parameters. It also provides a much easier method of starting the avionics processes before the radio link is established without having someone physically at the vehicle.

## Tracking & Receiving

The vehicle’s omnidirectional antenna won’t be transmitting with enough power to be received by an isotropic antenna on the ground. Therefore, a more directional high-gain antenna such as a Yagi-Uda antenna is required. To keep the antenna oriented correctly, a “turret” will be built. This turret would include a rotational servo, for controlling azimuth, and another servo for controlling elevation. Between these two servos the turret can be pointed at any point in the sky.

Mounted on the turret will be a high-resolution, high-zoom video camera for a visual feed of the rocket. This provides a very simple manner for visual target tracking.

Target tracking will be accomplished with a differential radio antenna system. Mounting 4 directional Yagi-Uda antennas in a diamond pattern, slightly angled away from the center, will give the capability to track the telemetry signal from the vehicle. Comparing the receive volume from the two left-right antennas yields the direction to rotate in, and comparing the volume from the two vertically-aligned antennas gives the direction to elevate in.

Since radio processing is a complex task requiring sufficient hardware, each of the radio receivers will be connected to a dedicated controller responsible for determining the appropriate action for the turret to take. Many such radio receivers are designed to output to a USB port, so something with a 4-port USB hub is required. Ideally, the device would also have a multi-core processor to allow for one process to determine necessary actions and another to order their execution.

Some smoothing will be required to keep the unit from oscillating violently based on small signal variations. It will likely be easier to keep the target slightly above the vertical center, so that the entire system continually and smoothly increases in elevation rather than attempting to constantly focus on the vehicle and then immediately falling behind. The optical camera can be angled up slightly to compensate for this, but more experimenting will be required to determine optimal setup and smoothing.

While implementation can be difficult, mounting a compact radar on the turret will also give distance data. Once a range value is acquired, a simple coordinate transformation can be done from spherical coordinates to Cartesian coordinates to acquire a 3-dimensional position of the vehicle in terms of latitude, longitude, and altitude. A benefit to this system is that it allows ground controllers to know the position of the vehicle even if every onboard system fails, since it requires no response from the vehicle. While an active-interrogation setup would provide more accurate and precise data, passive radar will still be capable of returning a range. Further, since the coordinate transformation is a handful of trigonometric functions, it can be performed in-place on the turret controller and handed back to the main ground control unit.

### Remote Abort System

As discussed above, it is vital to have a functioning abort system that can be activated from the ground control station. For security reasons, it is imperative that the signal be securely transmitted to prevent an erroneous abort that could be triggered unintentionally. To allow this, a radio receiver will be connected to the telemetry or abort hardware on the vehicle for signal processing and interpretation.

On the ground, however, it is advantageous to mount a high-power transmitter to the tracking turret. Since the turret will by design always be pointing at the vehicle, the transmitter antenna could be highly directional and minimize the power required to get an abort signal to the vehicle.

Since such an abort signal could be triggered by several things including radio interference, an unintentional transmission, or even a malicious third party, it is important that the signal be distinguishable from the background noise. Several options are being investigated to allow this, including the transmission of a cryptographically secure message, or a simpler text command. Each option could be subject to transmission degradation or interference, but that could be resolved with higher transmission power and active responses.

# Hardware Options

What follows is a comparison of computing hardware for multiple goals. Full-fledged computers are categorized under “computers”, single-task-oriented microprocessors and their associated boards are categorized under “controllers”, and peripherals such as sensors and radio equipment are categorized appropriately under the “peripherals” section.

## Computers

There are multiple options for computing hardware on the vehicle. Each one has it’s benefits and downfalls, but a quantitative comparison of each option is given below. It must be kept in mind that the BeagleBone Enhanced is currently the team favorite because the manufacturer, SanCloud LTD, provided 2 units free-of-charge.

|  |  |  |
| --- | --- | --- |
| Single-board computer | Raspberry Pi 3B+ | BeagleBone Enhanced |
| Manufacturer | Raspberry Pi Foundation | SanCloud LTD |
| Unit price ($) | 35.00 | 76.00 |
| Size (mm) | 85.6 x 56.5 x 17 | 86.4 x 53.3 |
| Weight (gram) | 45 | 39.68 |
| Connectivity | 1 Gb/s Ethernet  4 USB2.0  1 HDMI  1 3.5mm audio barrel  1 DSI  1 CSI  1 microUSB (power supply only)  1 40-pin header | 1 Gb/s Ethernet  2 USB2.0  2 USB2.0 (solder required)  1 microHDMI  1 miniUSB (power supply, USB bus)  1 5V DC barrel  1 debug serial header  2 46-pin headers |
| # GPIO pins | 17 | 19 |
| # Analog/PWM pins | 1x PWM | 7x Ain  8x PWM |
| # Bus-level interfaces | 1x UART  1x SPI, 2CS  1x I2C | 4x UART, 1 for debug  1x SPI, 2CS  2x I2C |
| CPU | Broadcom BCM2837 SoC  x64-bit  4xCoretex-A53 1.4GHz | TI AM3358BZCZ100 Sitara  x32-bit  Coretex-A8 1.0GHz  2x PRU-ICSS |
| Main memory | 1GB DDR3 (shared w/GPU) | 1GB DDR3 |
| Onboard storage | 1x microSD card slot | 1x microSD card slot  1x 4GB eMMC flash |
| Power consumption (max) | 980 mA @ 400% CPU load | Not documented |
| Sensors | N/A | 1x MPU6050  1x LPS331AP |
| Temperature range (C) | Certified: 0°C - 70°C  Tested: -110°C - 112°C | Certified: 0°C - 60°C |

## Controllers

|  |  |
| --- | --- |
| Microcontroller | Arduino Nano |
| Manufacturer | Arduino |
| Unit price ($) | 22.00 |
| Size (mm) | 18 x 45 |
| Weight (g) | 7 |
| Connectivity | 1x miniUSB for programming |
| Pinout | 6x PWM output  22x digital IO (includes PWM)  8x analog-in |
| Processor | 1x ATmega328  8-bit AVR  16MHz |
| Storage capacity | 1KB EEPROM (on processor die)  32KB flash (2KB used by bootloader) |
| Power consumption (max) | 19 mA |
| Operating voltage (V) | 5 |

## Peripherals

|  |  |  |  |
| --- | --- | --- | --- |
| Inertial Measurement unit | MPU6050 | MPU9250 | BNO055 |
| Manufacturer | InvenSense | InvenSense | Bosch Sensortech |
| Unit price ($) |  |  |  |
| Size (mm) | 4 x 4 x 0.9 | 3 x 3 x 1 | 5.2 x 3.8 x 1.1 |
| Weight (g) |  |  |  |
| Interfaces | 1xI2C | 1xSPI or 1xI2C | 1xHID-I2C  1xI2C  1xUART |
| Sensors | 1x3-axis accelerometer  1x3-axis gyroscope | 1x3-axis accelerometer  1x3-axis gyroscope  1x3-axis magnetometer | 1x3-axis accelerometer  1x3-axis gyroscope  1x3-axis magnetometer  Onboard sensor fusion |
| Accelerometer accuracy | 16384 | 8196 | 4096 | 2048 LSB/g | ±4800 LSB/g | 1 LSB/mg |
| Accelerometer range | ±2 | ±4 | ±8 | ±16 g | ±2 | ±4 | ±8 | ±16 g | ±2 | ±4 | ±8 | ±16 g |
| Gyroscope accuracy | 131 | 65.5 | 32.8 | 16.4 LSB/(°/s) | 131 | 65.5 | 32.8 | 16.4 LSB/(°/s) | 16 LSB/(°/s) |
| Gyroscope range | ±250 | ±500 | ±1000 | ±2000 °/s | ±250 | ±500 | ±1000 | ±2000 °/sec | ±125 | ±2000 °/s |
| Magnetometer accuracy | N/A | ±500 LSB | 0.3 µT |
| Magnetometer range | N/A | ±4800 µT | xy-axis ±1300 µT  z-axis ±2500 µT |
| Power consumption |  |  |  |

|  |  |  |
| --- | --- | --- |
| Barometer | MS5803-14BA | LPS331AP |
| Manufacturer | Measurement specialties | STMicroelectronics |
| Unit price ($) | 59.95 | 3.95 |
| Size (mm) | 6.2 x 6.4 x 2.88 | 3 x 3 x 1 |
| Weight (g) |  |  |
| Interfaces | 1xI2C  1xSPI | 1xI2C  1xSPI |
| Sensors | 1xbarometer  1xthermometer | 1xbarometer |
| Barometer accuracy | 0°C - 40°C: ±20mbar  -40°C - 85°C: ±40mbar | 0.020 mbar |
| Barometer range | 0 – 14 bar | 260 – 1260 mbar |
| Thermometer accuracy | ±0.8 °C | N/A |
| Thermometer range | -40 – 85 °C | N/A |
| Power consumption |  |  |