Project Document

Team 8: HALE Rocket Communications

Team Members:

Camden Robustelli **Jesse Fretz**

Sam Wagner **Kevin Kott**

Table of Contents

Section 1: Overview

1.1 - Executive Summary

This document is to inform the project partner and the HALE team about the current progress of the communications system. Our project will be combined with many other projects to create a fully functioning rocket. We are a part of a multi-year competition sponsored by NASA with the goal of launching a liquid bipropellant to the edge of space(100km). NASA spends about 20,000 dollars on the OSU HALE rocket, of which, about a thousand will go to our communication system.

The project for the ECE sub-team is to build an uplink and a downlink communication system for the rocket during its flight up to space. Our challenge will be to get this system to work up to at least 60,000 feet in the air. Future iterations of this project will build on what we have to get its communication range up to 100 kilometers.

At this point in the design process, our team has created a block diagram system and is working to finalize all the parts within the system. We currently meet with our project partner and HALE team every Monday at 7pm to check in to see where everyone is at and have discussions about the project. We plan on using ultra high frequency (UHF)/very high frequency (VHF) radio signals to communicate between the rocket and a ground station. We will need to use the amateur radio frequency bands which are between 420MHz to 450MHz (UHF) for the downlink and 144MHz to 148MHz (VHF) for the uplink. In order to transmit at these frequencies, we will need to get a HAM radio license.

The project will be split up into two discrete systems, a receiving and transmitting system. This allows us to simplify our protocol and focus on getting the information from both sides which will be a challenge of itself. All design features are subject to change until winter term.

1.2 - Team Communication Protocols and Standards

Table 1: Communication Information

Members Name	Email Address	Phone Number
Camden Robustelli	robustec@oregonstate.edu	(971)-332-6396
Kevin Kott	kottk@oregonstate.edu	(609)-346-1238
Sam Wagner	wagnsamu@oregonstate.edu	(503)-984-7907
Jesse Fretz	fretzj@oregonstate.edu	(541)-778-2576

Table 2: Team Roles

Role	Role Description	Role Owner
Video Producer	This team member splices the video clips together and produces the final video and submits it.	Sam Wagner
Team Communications Officer	This team member is responsible for enforcing the team to update their progress as well as coordinate weekly meetings.	Camden Robustelli
Treasurer	This team member makes sure we have enough funding and monitors are spending.	Jesse Fretz
Press Secretary	This team member communicates with the community as well as the larger HALE team.	Kevin Kott

Team Communication:

Our main form of communication is on Discord. We are constantly talking in a group chat daily about the project and breakthroughs. During the week we at least have one discord meeting using audio where we discuss bigger picture ideas and assignments. On Mondays we meet up in person in the library and later that night we meet with the whole HALE rocket team to discuss what we have done for the week. At that meeting, we sit down with the project partner and go over what we have done and ask him any questions if needed. We also contact him sometime during the week about project related information. If someone stops responding, try to contact them over discord and then meet them in class. If that doesn't work then we will talk to instructors for help.

Table 3: Team Protocols and Standards

Topic	**Protocol**	**Standard**
Team Communication	It is an open and free space where everyone can voice their opinions about the project at hand.	During team meetings everyone has a chance to talk about what they have done and their issues with the project.
Task Management	Everyone is given relatively an equal amount of work	No one person is given or doing nothing to help the project, work on the project is allocated during

		meetings
Respect	Each person in the group is treated with respect and equally.	If a conflict of interest arises and an argument breaks out the group members involved respect each other and the argument is fact driven not personal attacks.
Quality of Work	All team members are held at a high standard to provide an appropriate level of work for a senior undergraduate student.	Everyone's work will be compared to group members and class members to see if the work is up to par.
Conflict Resolution	Every team member will have an equal say when a conflict arises.	When we encounter a conflict if the team can solve it within ourselves with a compromise or a majority vote on a subject. If that cannot be done we will look for outside help from the professors in senior design to help with the issue.
Due Dates	Each person in the group needs to finish their work on the agreed timeline to ensure the project succeeds	The group has a timeline with dates that mark when specific project subsystems and assignments should be completed.

1.3 - Gap Analysis

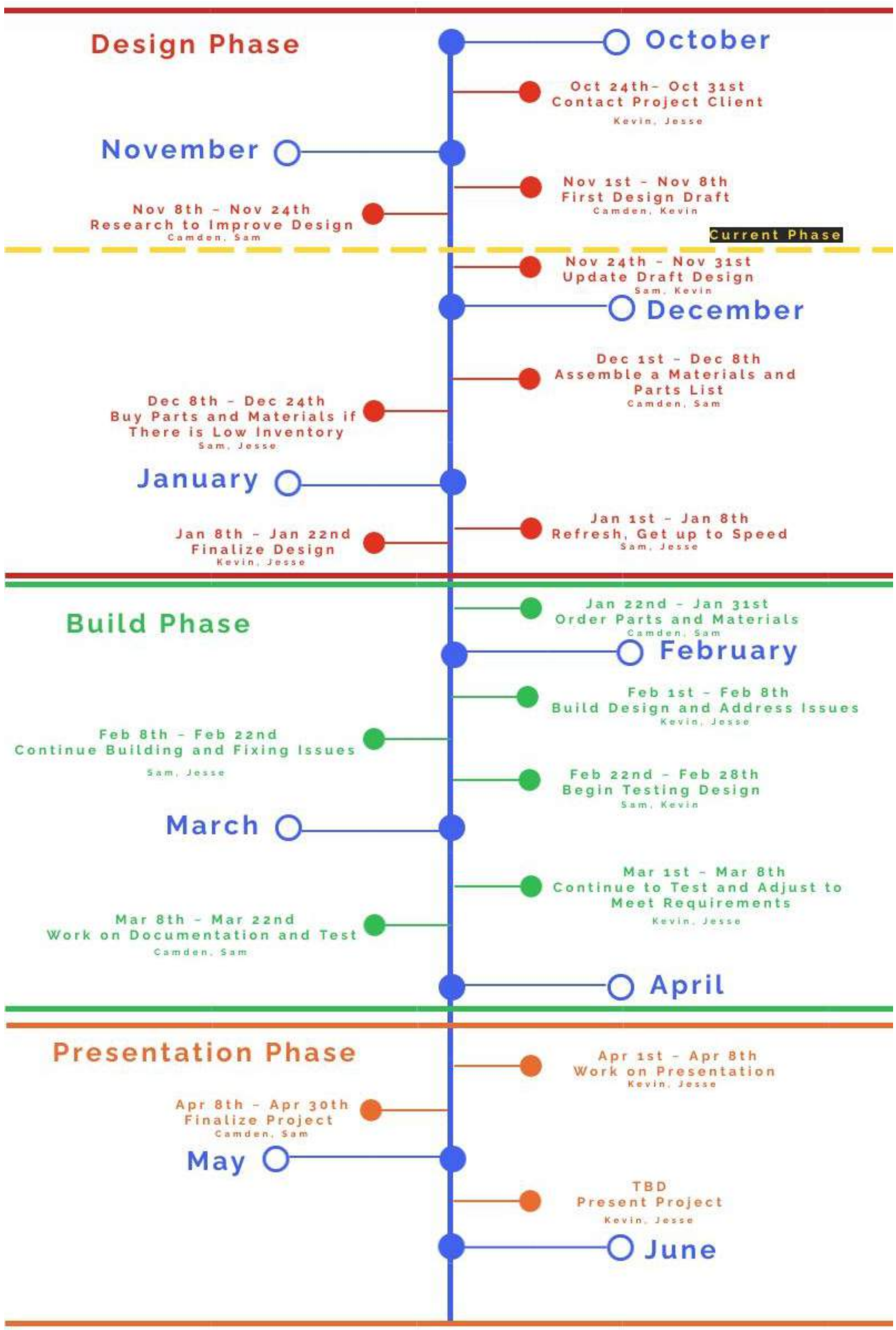
The reason our project exists is to deliver communication from the ground station to the rocket as well as send data from the rocket down to the ground station. As of right now there is no way for this team and many teams at the collegiate level to get live data from the rocket. Having live data will allow the rocket teams to monitor it in real time so if there is an issue they can terminate the rocket or redirect it mid-flight. Live data, such as atmospheric conditions, altitude, etc. can also be collected and used in studies by researchers of various fields (climate scientists, for example) later on.

The main stakeholder in this project is the HALE rocket team as they are the ones that have specifically requested this system be designed.

We have interviewed an individual who has industry experience developing RF communications systems for airplanes. The design we presented to him had one component receiving and transmitting on both the ground station and the rocket along with amplifiers on each antenna. He recommended having a dedicated microprocessor for receiving and transmitting so we wouldn't have to make a complex communications protocol. This will not only save us time but also simply designing the system and eliminating possible points of failure. By implementing the recommended changes so that the flow of data from one site to the other is more streamlined, the process of debugging the system becomes less complicated.

We can assume the rocket and ground station will have a clear path. We will also be using at least two microcontrollers and four transceivers for the ground station and rocket. Due to the rotation of the rocket there will be four antennas for both receiving and sending. We can also assume the rocket will be launched in clear weather and no other strong interfering signals (such as nearby HAM repeaters) will be present.

1.4 - Timeline/Proposed Timeline



1.5 - References and File Links

[1] J. Prades, A. Ghiotto, E. Kerhervé and K. Wu, "Broadband Sounding Rocket Antenna for Dual-Band Telemetric and Payload Data Transmission," in IEEE Antennas and Wireless Propagation Letters, vol. 15, pp. 540-543, 2016, doi: 10.1109/LAWP.2015.2457338.

[2] Wei, M., Hu, C., Estévez, D. *et al.* Design and flight results of the VHF/UHF communication system of Longjiang lunar microsatellites. *Nat Commun* 11, 3425 (2020). https://doi.org/10.1038/s41467-020-17272-8

[3] T. F. C. Leao, V. Mooney-Chopin, C. W. Trueman and S. Gleason, "Design and Implementation of a Diplexer and a Dual-Band VHF/UHF Antenna for Nanosatellites," in IEEE Antennas and Wireless Propagation Letters, vol. 12, pp. 1098-1101, 2013, doi: 10.1109/LAWP.2013.2280454.

[4] S. S. Alam, A. J. Islam, M. Mahmudul Hasan and M. Mehedi Farhad, "Design and Implementation of an Embedded System to Observe the Atmospheric Condition using a Helium Balloon," 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2018, pp. 242-246, doi: 10.1109/ICISET.2018.8745589.

[5] A. Y. Botero, J. S. Rodríguez, J. G. Serna, A. Gómez and M. J. García, "Design, construction and testing of a data transmission system for a mid-power rocket model," 2017 IEEE Aerospace Conference, 2017, pp. 1-14, doi: 10.1109/AERO.2017.7943739.

[6] V. Sokolov, E. Filimonov and V. Pyltsov, "The design method of low earth orbit (LEO) satellite communication systems for digital messages transmission with limited delivery time," Proceedings of The 2nd International Conference on Satellite Communications, 1996, pp. 186-191, doi: 10.1109/ICSC.1996.864277.

1.6 - Revision Table

Date	**Revisions Made**
10/19/2021	All team members, Initial rough draft
10/29/2021	All team members, Added more content to timeline and added communication protocols and team standards
11/11/2021	Camden, added and edited section 1.1 and cleaned up section 1.2 added table. Revised 1.3. Added a timeline to section 1.4 Kevin, Added more to gap analysis
11/19/2021	Sam, Added more to the executive summary to reach minimum word count of 300
12/1/2021	All team members - Edited the communication information table

Section 2: Requirements, Impacts, and Risks

2.1 - Requirements

1) **2.1.1**
   a) **Project Partner Requirement:** The communication system will be two-way, i.e. it will provide uplink and downlink data transfer. (Distance doesn't matter for this requirement.)
   b) **Engineering Requirement:** The ground station subsystem will send a test message to the rocket subsystem. Confirmation that the message was received on the rocket end will be done visually using arduino IDE serial monitor. Next, the rocket will send a test message to the ground station. Confirmation that the message was received by the ground station subsystem will be done visually, using arduino IDE serial monitor.
   c) **Verification Method:**
      i) The uplink will be verified first. This is communication from the ground station to the rocket. The rocket and ground station Arduinos will be connected to individual computers, and then a string of characters (such as the numbers 1 through 10) will be sent from the ground station to the rocket. If at least 75% of the characters are displayed within the Arduino console on the rocket's computer then the uplink has been verified.
      ii) The downlink will be verified second. This is communication from the rocket to the ground station. The same setup used for the first verification method will be used for this verification method. A series of characters (such as the numbers 11 through 20) will be sent from the rocket to the ground station and if the ground station computer displays at least 75% of the same characters within the arduino console then the downlink is verified.
      iii) Both sites should be able to transmit and receive messages independently, without altering hardware connections.
2) **2.1.2**
   a) **Project Partner Requirement:** The radio will be designed to work at a minimum of 60,000 ft (~11.4 miles) between the rocket and the ground station.
   b) **Engineering Requirement:** The rocket subsystem will be capable of transmitting RF signals to a distance of at least 60,000 ft. This will be confirmed by measuring the gain of each antenna and then estimating the effective link range of the system. A conservative link margin of 25dB will be used to account for any unforeseen losses within the system.
   c) **Verification Method:**
      i) To confirm this requirement, each site's transmitting and receiving antenna will have their gain measured. This is performed by using antennas with known gain as a baseline, taking a measurement from a distance, then replacing one of the known antennas with the antenna-under-test and taking a measurement from the same distance.
      ii) Once the gain has been measured, an online RF Link Range calculator will be used to estimate the ideal range of our system. The link to the calculator is: https://www.immersionrc.com/rf-calculators/

iii) An ideal RF link range of at least 1.5x the target distance of 60,000ft (18.288km) will be used to confirm this requirement to account for any non-idealities within the system. 1.5x the target distance is 90,000ft (27.432km).

3) **2.1.3**
   a) **Project Partner Requirement:** The module must fit inside the body of the rocket. (Try to design it to fit in as little space as possible)
   b) **Engineering Requirement:** The horizontal measurement of the enclosure (both length and width) must not exceed 6 inches. The vertical measurement is preferred to be as small as possible but a target is 12 inches. (Excluding antennas)
   c) **Verification Method:**
      i) The rocket communication module will be measured with a tape ruler. If the ruler measures the length and width to be less than eight inches. If the ruler measures the height of the enclosure (excluding coax cables) to be less than 12 inches, too, then this property is verified.
4) **2.1.4**
   a) **Project Partner Requirement:** The ground station subsystem must be able to communicate with an outside computer for user input/output
   b) **Engineering Requirement:** The Ground Station will be able to communicate with an outside user interface by means of input from the user and output to the user.
   c) **Verification Method:**
      i) This requirement can be tested on two Arduino serial monitors or an Arduino serial monitor on the transmitting end and a Python application (which was one of the constraints).
      ii) One will type in a string of characters (like numbers 1 through 10) on the Arduino serial monitor. Then it will reprint each of the numbers that were entered into the terminal, in order from first to last, as each of them are being transmitted out to the receiving side. This reprinting, which is the communication between the ground station and the Arduino Uno, will act as confirmation that each number was transmitted. This will be verified for both the uplink and the downlink.
      iii) The receiving end will print out numbers that were transmitted by the rocket module. It should print them out in the order they were sent. This will confirm that the Ground Station communicates with the Arduino Nano. The acceptable loss of 25% applies to this requirement for both uplink and downlink. That means at least 75% of the characters transmitted need to be received and printed out.
      iv) The receiving port of the ground station will also be tested with the Python script. The verification of the communication between the Python application and the ground station will be confirmed using step iii.

5) **2.1.5**
   a) **Project Partner Requirement:** There must be four antennas on the rocket. **Engineering Requirement:** Four antennas on the rocket. Two for receiving and two for sending data. Receiving antennas should be on opposite sides from the rocket. Transmitting antennas should also be opposite from each other.
   b) **Verification Method:**

i) Visual verification. If there are four antennas, with two of them on the receiving ports and two of them on the sending ports.
ii) If verification method 2.1.1 passes with the four antennas set up, then this shows that the four antennas are working.
iii) The two sets of antennas are controlled by switches. If further proof is needed, remove the transmitting antennas and hook up one of the sending ports to an oscilloscope reader. Make note of what port it is hooked to. Set the switch pin to HIGH or LOW, whichever one turns on the port. If there is a steady voltage wave on the oscilloscope, then this means that one port is working. Check the other port - it should not be working. Set the switch pin to the opposite of what it was and see if a voltage wave appears. If so, then that means the switch is working and this property is verified.

6) **2.1.6**
   a) **Project Partner Requirement:** The major electronic components will be stored neatly and safely while still being easily accessible.
   b) **Engineering Requirement:** An enclosure will be designed to house all of the electrical components. The components will be mounted neatly within the enclosure with no components moving around.
   c) **Verification Method:**
      i) The enclosure for both the ground station and the rocket module will entirely contain the Arduinos, HamShields, PCB, and Power module.
      ii) The enclosure will be inspected after mounting the internal components and checked for loose cables and mounts. All circuit boards will be either mounted with screws, secured with zip ties or velcro straps or taped to the surface of the inside of the enclosure. Other ways of mounting and securing are feasible, too.
      iii) After the internal components are mounted, the system's wiring will be disconnected between modules (the wiring in individual blocks won't be disconnected) and reconnected to confirm accessibility.

7) **2.1.7**
   a) **Project Partner Requirement:** Antenna monopole cannot protrude outside the body of the rocket.
   b) **Engineering Requirement:** The antenna must be fully contained within the eight inch diameter body of the rocket.
   c) **Verification Method:**
      i) The antennas will be connected with flexible coax cable. Measure with a foot ruler. If the maximum distance between the furthest two points on the coax cable or antennas is less than eight inches (with someone exerting pressure on the coax cables to make them as close as possible) then this property is verified.

8) **2.1.8**
   a) **Project Partner Requirement:** The rocket subsystem should be light enough so that it doesn't affect the trajectory of the rocket.
   b) **Engineering Requirement:** The subsystem within the rocket must be less than 10 pounds.
   c) **Verification Method:**
      i) The weight will be measured with a package delivery scale.

9) **2.1.9 (Insurance requirement)**
    a) **Project Partner Requirement:** The antennas/transceivers will be easily replaceable with new antennas.
    b) **Engineering Requirement:** The communication system must be able to swap out the necessary antennas to communicate from space (100 km) when needed for the rocket next year. Antennas must be detachable and swappable for different ones. Any external components or wiring leaving or entering the module must be detachable.
    c) **Verification Method:**
        i) All antennas, wires (connecting between blocks) and power supplies will be taken out of their connectors and put back in to verify that they are in fact replaceable.
        ii) Other components, such as different wires and antennas will replace the current components to see if the module still works.
        iii) Redo the previous eight verification methods with these new antennas or wires. If they are all verified then this last requirement will be verified.

2.2 - Impacts

Public Safety Impacts

The project could have impacted the public, if the wrong radio frequency is used it would interfere with other companies or people trying to use the same frequency. One specific example is commercial radio waves for commercial flights, in the most extreme of cases our radio waves could interfere with a commercial flights radio communication not allowing them to get into contact with air traffic controller's causing them to be unaware of other flights and they could collide with another plane. According to "Applications of Radio Frequency technology in Aviation," by Rajiv, this frequency for commercial flights is somewhere between 118 MHz and 137 MHz, given this information as a group we stayed outside of this frequency when communicating with the rocket. The best way to prevent this impact from happening is to make sure careful research is done on available radio frequencies [8].

Welfare Impacts

The HALE Rocket team will get funding from NASA, which itself is funded by taxpayers. With so many things society could be spending tax money on, would spending it on a 'fancy' rocket communication system be the best? Humanity mostly sends rockets up to space to send humans to conduct experiments on the International Space Station or explore other planets. While these are important to advance science and technology, they wouldn't be very useful to the average person and their local community. Usually, someone would rather spend the money to improve something in their immediate surroundings rather than on some far off organization that has nothing to do with them. Even if someone were willing to invest on a global scale, projects like developing medicine and vaccines, better ways to produce more food and so on, would directly benefit the general population. They would save many lives and improve the quality of life for many others.

One way to deal with this negative economic factor would be to turn it into a positive factor. The communication system will be forwarding data back to Earth and the data could benefit humanity somehow. Climate change is an issue that will hurt people in the future. Sending a rocket up into the atmosphere to collect climate data could help prepare for future impacts, in turn saving people. The communication module could be modified in a way such that satellites could use it too. Satellites are important tools in weather forecasting and tracking. They can predict deadly storms ahead of when they hit.

Cultural and Social Impacts

Within a status report article from NASA ASK Magazine in 2008 written by Steven J. Dick titled, "The Societal Impact of Space Flight," Dick recalls past milestones in space flight from the good things like the Moon Landing and Sputnik, as well as some tragic incidents like the space shuttle disasters. Rocket science is difficult, this is a reason not everyone can or even wants to do it, calculations can be made incorrectly, systems can fail, and this can lead to tragedies like the space shuttle incidents. These tragedies can take a toll on society, failure is never perceived well by people, especially when some of these failures are broadcast for everyone to watch. These terrible incidents can instill fear for space flight into societies hearts, which in turn may lead to less progression in the space flight fields as less and less people will pursue it. This can be an unfortunate loop, without more people working on the safety of space flight more accidents could happen, leading again to more fear and less people to pursue space flight. This fear has a large impact socially and culturally. The way our team could prevent this is by making sure our subsystem in the rocket is fully secured and enclosed in order to ensure that any malfunction in the rocket is not caused by our system [9].

Environmental Impacts

If launch of the rocket does not go as planned and the rocket is sent to a place where it is unrecoverable, it will add to the Earth's pollution as well as add to the already exceeding waste throughout the planet, whether it be in the ocean or in landfills waste leads to a lot of the pollution we see today. Another danger of our rocket being unrecoverable by our team could be if someone else stumbles onto it who is not careful with it, if they do not handle it with care it could cause problems because there are many highly flammable materials that could cause a fire easily. And according to Bob Bewyn in his article "How Wildfires Can Affect Climate Change (and Vice Versa)," he talks about a study from scientists saying that wildfires are responsible for almost 8 billion tons of CO2 in a year [10].

Some ways we can prevent or lessen these impacts is to ensure recovery the best we can, by putting multiple "eyes" on the rocket, one of the eyes being the group members watching to see where the rocket touches down. The other set of "eyes" is the communication system, when working properly in conjunction with an eventual avionics system, our system could relay the location of the rocket's whereabouts. Prevention of fires is something as a societal whole is being worked on, in our case we need to make sure to take precautionary measures to make sure that if a fire does start, we can put it out before it becomes too large for us to deal with. Another method is to make sure that the launch of the rocket is not done in a dry/flammable area. For the lithium polymer batteries, we plan to make an enclosure to hold the entirety of our communications, so in the worst-case scenario only our system and our enclosure is harmed rather than having the batteries cause a fire.

Economic Impacts

Right now, there is a large shortage of electronic parts across the world making it hard to purchase and receive those parts. Our project contributes to the shortage in the sense that we need the parts for our communication system in a rather small/short timeline to build our system. In the Forbes article “The Big Chip Shortage: Three Innovation Strategies Against The Lack Of Electronic Components,” by Marco Annunziata, it is said that the shortage is short-lived, though unfortunately so is our project. With this shortage of components, and our project's need for them comes the possibility of not being able to complete the system, or at the least having to use less optimal components due to the shortages. There also comes a small chance a project more important than ours is put on hold because of shortages, and the purchase of our parts go through before theirs [11].

Minimizing this impact was difficult, as we needed to get components to complete our project. We tried to use components from past projects that were in good shape if possible, but not too many past components were used.

2.3 - Risks

Table 4: Table of Risks and Action Plans

Risk ID	Risk Description	Risk Category	Risk Probability	Risk Impact	Performance Indicator	Responsible Party	Action Plan
R1	Part breaks	Technical	10%	(M) Depending on when it breaks could be huge, could be minimal	Stops producing the values/data needed	Camden	If a part breaks and we do not have a spare, a replacement part will be ordered. The broken part will then be removed and the new part will be transferred into the system. All part breakages will be reported to the rest of the team.
R2	Out of Stock [1]	Timeline	50%	(L) Small impact	Vendor or website says out of stock	Kevin	Wait for a week. If it doesn't return to stock, then email the vendor. Look into other components to reduce the amount of time lost due to waiting.
R3	Miscommunication of tasks	Communication /Time management	50%	(M) Medium impact	One task has been untouched and/or two people have done the same task	Sam	Communicate frequently on Discord and use the weekly presentations to keep track of who will do what. Retain records of who does what on an online calendar.

R4	Exams and other class assignments	Time management	75%	(H) Large impact if not dealt with in a timely manner	Peoples engagement with and work on the project slows down	Jesse	Plan ahead of time when everyone’s midterms are and make sure to make plenty of progress before they hit. Make an online calendar to keep track of everyone’s outside engagements. Transfer responsibilities from the busy team member to the other less busy team members if necessary.
R5	Hardware issues and debugging	Technical	99%	(M) Medium impact because an issue in program or hardware can break the whole system	Morale goes down. Building more of the design cannot be done until bugs are fixed.	Anyone experiencing the error.	Almost guaranteed to happen. Whoever experiences an error should avoid hiding it and reach out to the others for help. If the error persists for more than a week, we should reach out to any ECE faculty that can help.
R6	PCB Delay/Built Incorrectly	Technical/ Timeline	50%	(M) May be able to still build but no PCB makes loss of requirement and will damage our grade significantly.	The company doesn’t ship the PCB one week after requesting it. ETAs keep getting longer. The PCB doesn’t have the same traces	Jesse	Design PCB early and avoid ordering a PCB from a company far away to reduce shipping delay risks. Find a reliable business or website to make the PCB we design. Look at the online reviews and see if they get 95% positive reviews or more in order to avoid.

					and connection as our blueprint.		
R7	Insufficient/ Excess Power Provided to the System	Technical	25%	(H) The entire system will fail if power is not met sufficiently.	Voltage and current are not being supplied across the system.	Jesse	Avoid immediately plugging in any power supply without proper consultation. Test power supply outside of the system to make sure it supplies within the range needed. Communicate with experts for help if needed.
R8	COVID resurges and school shuts down	Health	5%	(H) Limits our ability to meet together and use university facilities	Case numbers in the news go up and the university closes campus	Sam	Retain connections with each other, but transfer them to a fully online medium. Meet sparingly to build up the module.
R9	Parts Supplier going out of business	Technical	100%	(H) Eliminates the possibility of purchasing replacement parts in the event of parts breakage	Parts supplier website is taken down	Kevin	In the event that a parts supplier goes out of business, there are only a couple actions that can be taken: Replace broken parts with alternatively sourced parts, Redesign system and find another supplier

2.4 - References and File Links

[1] Annunziata, Marco, *The Big Chip Shortage: Three Innovation Strategies Against The Lack Of Electronic Components*, Forbes, Sep, 2021. Available: https://www.forbes.com/sites/marcoannunziata/2021/09/15/the-big-shortage--3-innovation-strategies-against-the-lack-of-electronic-components/

[2] Oldenburg, Baron, *The Yagi-UDA Antenna: An Illustrated Primer*, DUO, Aug, 2019. Available: https://duo.com/labs/tech-notes/the-yagi-uda-antenna-an-illustrated-primer

[3] Frenzel, Louis E. *Welcome to Antennas 101*, ElectronicDesign, Jul, 2021. Available: https://www.electronicdesign.com/technologies/passives/article/21769333/welcome-to-antennas-101

[4] Halverson, Casey, *HamShield for Arduino (VHF/UHF transceiver)*, KickStarter, Oct, 2019. Available: https://www.kickstarter.com/projects/749835103/hamshield-for-arduino-vhf-uhf-transceiver

[5] Ghosh, Anindo, *How to know (or estimate) the range of a transceiver?* StackExchange, Nov, 2013. Available: https://electronics.stackexchange.com/questions/83052/how-to-know-or-estimate-the-range-of-a-transceiver

[6] *RF Calculators,* ImmersionRC, Nov, 2013. Available: https://electronics.stackexchange.com/questions/83052/how-to-know-or-estimate-the-range-of-a-transceiver

[7] Enhanced Radio Devices, *HAMSHIELD 1.0,* Inductive Twig, 2021. Available: https://inductivetwig.com/products/hamshield

- Design Impact Statement Citations

[8] Rajiv, Applications of Radio Frequency technology in Aviation, RF Page, Aug, 2021, Available: https://www.rfpage.com/applications-of-radio-frequency-technology-inaviation/

[9] Dick, Steven J, The Societal Impact of Space Flight, SpaceRef, Dec, 2008, Available: http://www.spaceref.com/news/viewsr.html?pid=30009

[10] Berwyn, Bob, How Wildfires Can Affect Climate Change (and Vice Versa), Inside Climate News, Aug, 2018, Available: https://insideclimatenews.org/news/23082018/extreme-wildfires-climate-change-globalwarming-air-pollution-fire-management-black-carbon-co2/

[11] Annunziata, Marco, The Big Chip Shortage: Three Innovation Strategies Against The Lack Of Electronic Components, Forbes, Sep, 2021. Available: https://www.forbes.com/sites/marcoannunziata/2021/09/15/the-big-shortage--3-innovation-strategies-against-the-lack-of-electronic-components/?sh=33a84fb345f0

2.5 - Revision Table

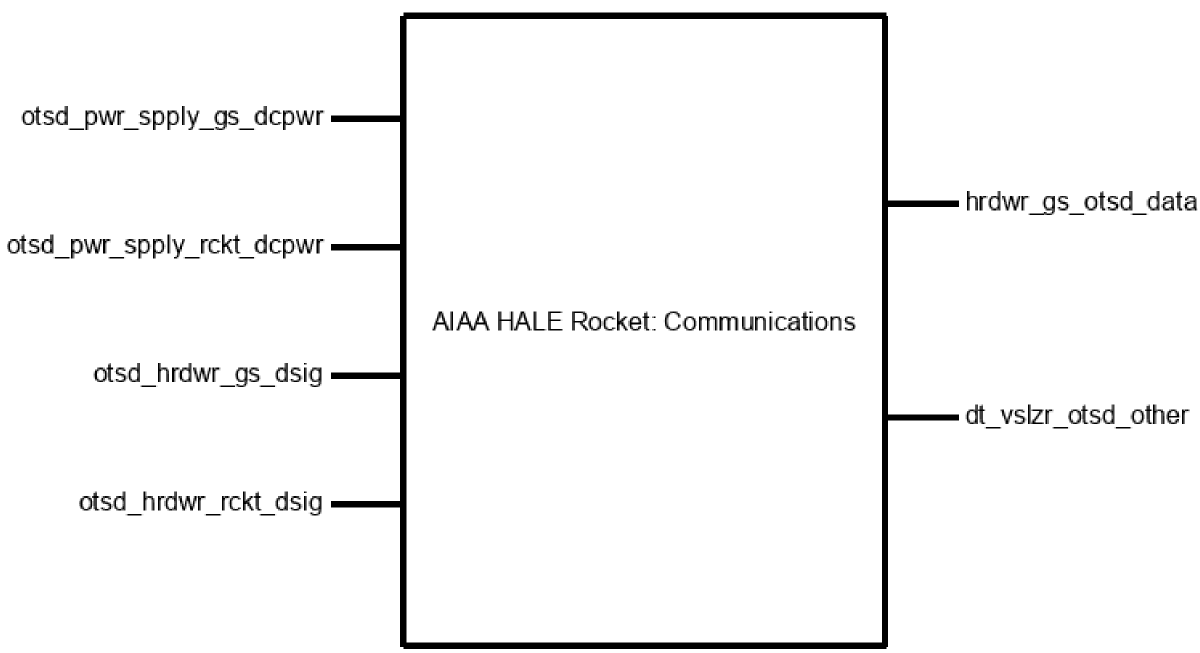
Date	Revisions Made
10/28/2021	All team members, added initial information for each section
11/11/2021	Sam & Jesse, Added more risks, and created more requirements and split them into Engineering and Design sub requirements.
11/12/2021	Camden, Put requirements into a table and reformatted all sections

	Sam, Added verification methods for the engineering requirements, replaced R3 with another risk. Jesse, Made sources in IEEE format.
11/19/2021	Sam, reformatted section 2.1, deleted two of the risks in section 2.3
12/1/2021	All team members, Revised the project requirements to satisfy the feedback given. Added one more risk and added key words to action plans.
5/4/2022	Sam, updated testing procedures in section 2.1 so that they match section 5
5/6/2022	Jesse - Added impacts and citations to 2.2 Sam - Added welfare impact
5/6/2022	Kevin - Added Risk and Action Plan

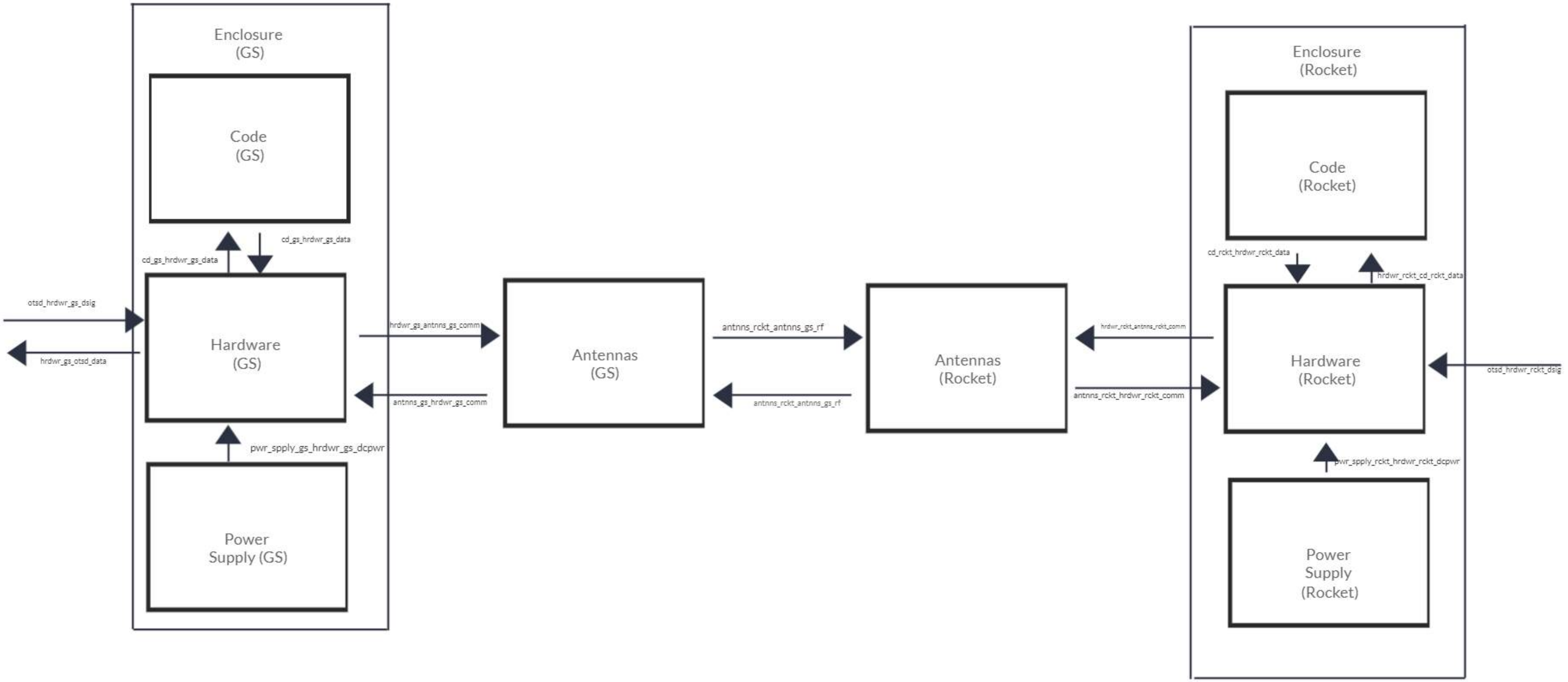
Section 3: Top-Level Architecture

3.1 - Block Diagram

Black Box view



White Box View



3.2 - Block Descriptions

Table 5: Table of Blocks and their descriptions

Name	Description
Power Supply (GS) Champion: Jesse Fretz	This block supplies some current and voltage to the components in the ground system, including the Arduinos, Hamshields, and the interface where the transmitted information and values will be displayed. This block may receive power from either an outlet or there is a chance that it will also be powered by batteries if the launch location does not have access to wall power.
Power Supply (Rocket) Champion: Jesse Fretz	Powers the system within the rocket will most likely be battery powered to supply the correct voltage and currents. This will also likely include some sort of converter in order to provide the right power to the components, like the Arduinos, Hamshields, and switches.
Code (Rocket) Champion: Sam Wagner	This will be the code for the rocket receiving side. It will be stored on the Arduino Uno/Mini hardware. It will take received signals from the antenna, through the Hamshield and coax cable, and interpret the signal. The interpretation will be output as a message (a string of ASCII characters). In future extensions of this project, this block might send commands to equipment on the rocket, like engines or pumps. Also, a protocol will be necessary for the SPDT switch, so that while switching between receiving antennas, the message from the ground station is not cut off. Another protocol will be developed to switch between the two transmitting antennas. This is to avoid the transmission getting cut off as the rocket is spinning along its vertical axis. The C language will be used as well as custom Hamshield Arduino libraries.
Enclosure (GS/Rocket) Champion: Camden Robustelli	This block is the enclosure for the ground system and rocket system. There will be two separate enclosures but they both will have both a Hamshield mini, Hamshield 1.0, power supply, Arduino Uno, Arduino Mini. The only difference is the rocket version of the enclosure will have two switches for the 4 antennas and a battery pack. The block will be 3D-printed and made of hard rigid plastic. The block will also have a mechanism to open and close it and will also have places for connectors to fit in.

Code (GS) Champion: Sam Wagner	This is a non-physical block since it is code. This code will be stored on the Arduino Uno/Mini hardware. It takes in a string of characters from an external source, like a computer, and processes it. It then outputs a signal to the HamShield so that it can turn it into an AFSK radio signal. The language used will be C. Custom Hamshield Arduino libraries will be used to create the output signal.
Antennas (Rocket) Champion: Kevin Kott	The Rocket Mounted Transmitting Antennas will be of a Rubber Ducky type antenna, tuned to operate at 435 MHz, the median frequency for our operating range. We will be using the UHF band for downlink communication from the rocket and the frequency range allocated for amateur use (420 to 450 MHz). Antennas will consist of a solid copper wire coiled into a helical pattern and impedance matched to 50 ohms using an in-series capacitor. Antennas will be encapsulated in heat shrink wire tubing and be tuned using a Vector Network Analyzer (VNA). Since rubber ducky antennas do not have a standard set of equations that define dimensions the antennas' dimensions have not been determined beforehand. This is because the rubber ducky antenna does not follow the fractional wavelength design characteristics of other types of antennas and instead prioritizes form factor size and minimizes spatial takeup. The antenna's dimensions will be ever evolving to achieve adequate efficacy based on operating conditions and gathered VNA data. The antenna's dimensions and design characteristics are subject to change.
Antennas (GS) Champion: Kevin Kott	The Ground Station Receiving Antenna will be a Quagi type antenna, tuned to operate at 435 MHz, the median frequency for our operating range. We will be using the UHF band for downlink communication from the rocket and the frequency range allocated for amateur use (420 to 450 MHz). Antenna will be composed of 8 copper elements, have a length a little less than 1.4 meters, and have an impedance of 50 ohms. The driven and reflector elements will use 14 AWG solid wire and the director elements will use 3mm rods. The antenna's design characteristics will be ever evolving to achieve adequate efficacy based data obtained from our VNA (Vector Network Analyzer). Antenna dimensions and design characteristics are subject to change.
Hardware (GS) Champion: Camden Robustelli	This block consists of two different modules. One for receiving data and one for sending. The receiving data is made up of an Arduino Mini and a Hamshield Mini. The sending data is made up of a Hamshield 1.0 and an Arduino Uno.

Hardware (Rocket) Champion: Camden Robustelli	This block consists of two different modules. One for receiving data and one for sending. The receiving data is made up of an Arduino Mini and a Hamshield Mini. The sending data is made up of a Hamshield 1.0 and an Arduino Uno. There will also be two switches, one to send the signal to the two sending antennas and the other one will be for the receiving antennas.

3.3 - Interface Definitions

Table 6: Table of Interfaces and their properties

Name	Properties
otsd_pwr_spply_gs_dcpwr	• **Inominal:** 5000mA • **Ipeak:** 5.2A • **Vmax:** 14.8 • **Vmin:** 10V
otsd_hrdwr_gs_dsig	• **Other:** 9600 baud • **Other:** Serial • **Vmax:** 5V
otsd_hrdwr_rckt_dsig	• **Logic-Level:** Active High • **Other:** Data from rocket sensors • **Vmax:** 5V
pwr_spply_gs_hrdwr_gs_dcpwr	• **Inominal:** 40mA • **Ipeak:** 500mA • **Vmax:** 12V • **Vmin:** 7V
pwr_spply_rckt_hrdwr_rckt_dcpwr	• **Inominal:** 134mA • **Ipeak:** 1.2A • **Vmax:** 12V • **Vmin:** 7V

cd_rckt_hrdwr_rckt_data	• **Datarate:** 9600 baud • **Messages:** Code for receiving data • **Messages:** Code for sending data
enclsr_gsrckt_pwr_spply_gs_other	• **Other:** In ground station enclosure • **Other:** Secured down into enclosure • **Other:** Has a designed spot for the part
enclsr_gsrckt_pwr_spply_rckt_other	• **Other:** In rocket enclosure • **Other:** Has a designed spot for the part • **Other:** Secured down into enclosure
enclsr_gsrckt_hrdwr_gs_other	• **Other:** In ground station enclosure • **Other:** Has a designed spot for the part • **Other:** Secured down into enclosure
enclsr_gsrckt_hrdwr_rckt_other	• **Other:** In rocket enclosure • **Other:** Has a designed spot for the part • **Other:** Secured down into enclosure
cd_gs_hrdwr_gs_data	• **Datarate:** 9600 baud • **Messages:** Sending code for ground station • **Messages:** Receiving code for ground station
antnns_rckt_antnns_gs_rf	• **Datarate:** UHF: 420MHz - 450MHz • **Other:** Rx Gmin: 1dBi • **Other:** Tx Gmin: 1dBi • **Other:** Rx Gnom: 10dBi • **Other:** Tx Gnom: >1dBi
antnns_rckt_hrdwr_rckt_comm	• **Other:** VSWRmax: 10 • **Other:** Zdev: +/-50% • **Other:** Znom: 50Ω
antnns_gs_antnns_rckt_rf	• **Datarate:** VHF: 144MHz - 148MHz • **Other:** Rx Gmin: 1dBi • **Other:** Tx Gnom: >=10dBi

	● **Other:** Rx Gnom: >1dBi ● **Other:** Tx Gmin: 1dBi
hrdwr_rckt_antnns_rckt_comm	● **Other:** Znom: 50Ω ● **Other:** Zdev: +/-50% ● **Other:** VSWRmax: 10 ● **Other:** Pnom: >445mW ● **Other:** Pmin: 165.3mW
antnns_gs_hrdwr_gs_comm	● **Other:** Znom: 50Ω ● **Other:** Zdev: +/-50% ● **Other:** VSWRmax: 10
hrdwr_gs_antnns_gs_comm	● **Other:** Pmin: 165.3mW ● **Other:** Pnom: >445mW ● **Other:** Znom: 50Ω ● **Other:** Zdev: +/-50% ● **Other:** VSWRmax: 10
hrdwr_gs_otsd_data	● **Datarate:** 9600 baud ● **Messages:** Displays message received on screen ● **Protocol:** Serial
hrdwr_gs_cd_gs_data	● **Datarate:** 9600 baud ● **Messages:** Sends received message to be decoded ● **Protocol:** Serial
hrdwr_rckt_cd_rckt_data	● **Datarate:** 9600 baud ● **Messages:** Message received to be decoded ● **Protocol:** Serial

3.4 - Reference and File Links

[1] *Data Sheet 74HC/HCT595*, Arduino, Jun, 1998. Available: https://www.arduino.cc/en/uploads/Tutorial/595datasheet.pdf
[2] Enhanced Radio Devices, *HAMSHIELD 1.0,* Inductive Twig, 2021. Available: https://inductivetwig.com/products/hamshield
[3] *Arduino Uno Rev3,* Arduino, 2021. Available: https://store-usa.arduino.cc/products/arduino-uno-rev3

3.5 - Revision Table

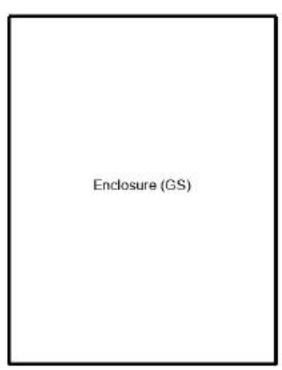
Date	Revisions Made
11/19/2021	All team members, created section and added initial content
12/3/2021	All team members, revised and improved interfaces, block descriptions, and block diagram
3/6/2022	Kevin Kott: Updated interface properties and definitions for the antenna blocks
5/6/2022	Kevin Kott: Updated interface properties and definitions for the antenna blocks

4.1 Ground Station Enclosure

4.1.1 Block Overview

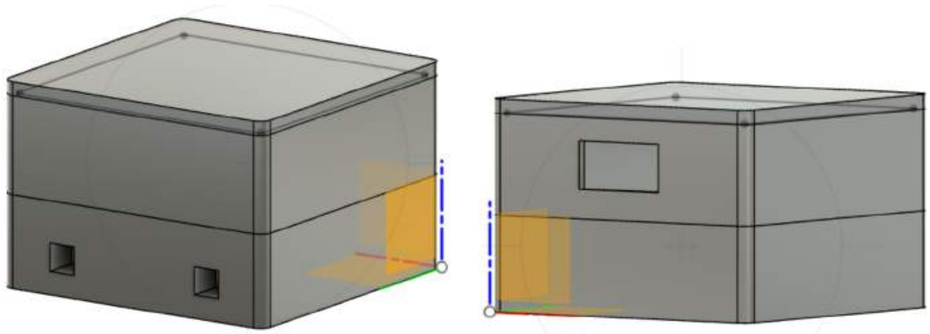
This block is the enclosure for the ground station system. It will house a transceiver, receiver, power supply and the microcontrollers. This enclosure will be two tiers that snap together the top tier will have the power supply with batteries as well as the external ports such as a USB . The bottom tier will hold the rest of the components and will have accessible ports to the SMA connectors coming off of the transceiver and receiver. It will be 3d printed and the lower tier components will be velcroed down.

4.1.2 Block Design



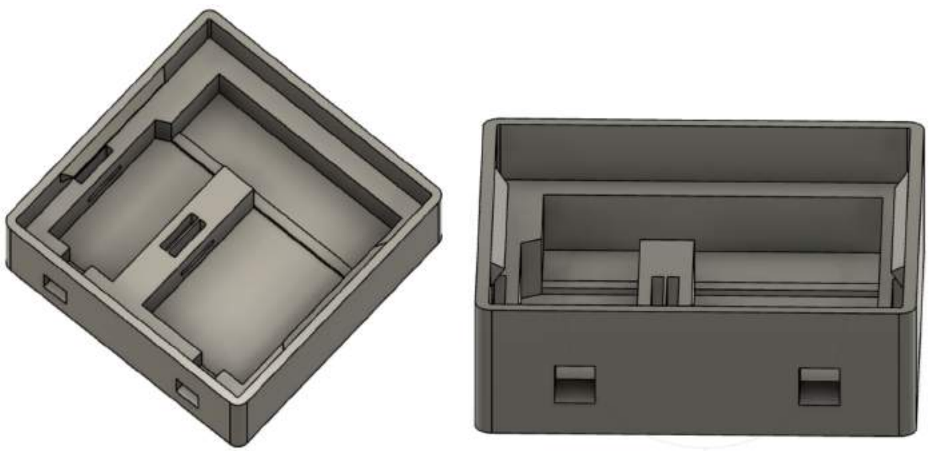
Fusion 360 3D Models for Enclosure:

Overview:



The model above shows the end result of the enclosure. It is a three piece enclosure that snaps together and can be taken apart. The front has two ports for the sma in the bottom piece and the middle piece on its back has a USB and a power port.

Bottom Section:



This model above shows the bottom section and as you can see there are two ports for the sma that are accessible. The Hamshield 1.0 stacks on the arduino uno in the right side bigger component holder. Velcro will be woven through the slots and over the stacked components to hold them down into the enclosure. Then in the left side smaller component holder there will be a Hamshield Mini stacked on a nano and held down by another piece of velcro strap through its corresponding slots. The back bay in the enclosure is where the wiring will run between the components as well as the ports for power and communication will be facing that way.

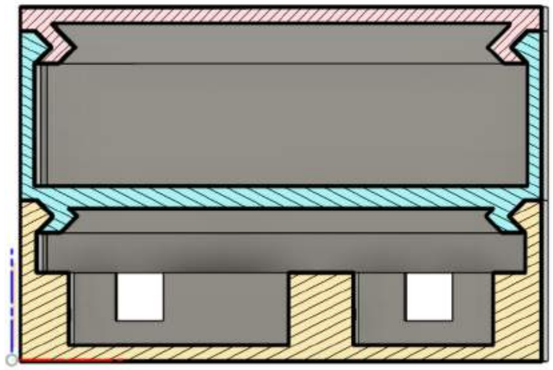
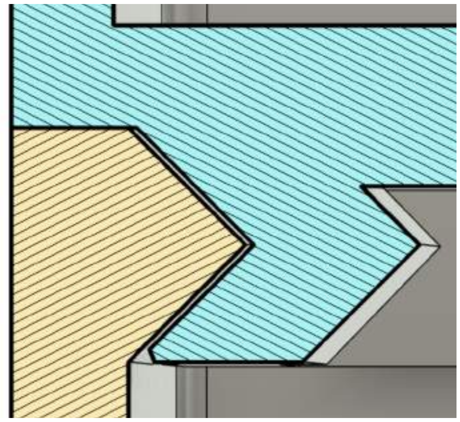
Middle Section:

Above is the model of the middle section which will house the power supply. There is a rectangle cut out in the bottom of this section that allows you to run power cables and communication cables to both sections. The backside cut outs will have external connections in it one will be a USB port that is connected to the microcontrollers and the other one is a power port that is connected to the power supply circuit.The middle section fits in the bottom section and snaps into place. Power supply parts will be secured down at a later date upon completion but will fit in this space.

Top Section:

The model above shows the top section of the enclosure. This is just a lid that snaps onto the top of the middle section and can be removed for quick access to the components. The groves are the opposite of what is on the middle section which allows them to lock together.

Cross Section Complete Enclosure:

The model above shows the complete enclosure together. From this cross section you can see how each section snaps together.

4.1.3 Block General Validation

This block is a crucial piece of the project itself. All the components need to be protected and secure so if any accidents occur like dropping the system. This also makes are project look professional with a clean container that holds all the components in a specific place.

4.1.4 Block Interface Validation

The USB port and the power supply port are to be installed on a later date but there is room for it to be installed. The power supply is subject to change and as of right now it will house two lipo batteries and not have a power port to connect to another outside source. There is enough room for the two batteries and the circuit for the power supply in the power supply area of the enclosure.

Section Dimensions:

Arduino Uno stacked on Hamshield Cutout:
Front wall offset- 12 mm
Side Wall offset- 8mm
Length- 70 mm
Width- 55 mm
Height- 18 mm
Coax Port from corner- 8mm
Width Coax port- 20mm
Back Ridge Height- 2mm

Bottom Section:
Height- 40 mm
Length- 130 mm
Width- 160 mm
Shell- 4 mm

Back Cutout:
Width- 100 mm
Height- 18 mm
Length- 65

Arduino Nano on Hamshield Mini Cutout:
Front wall offset- 12 mm
Side Wall offset- 8mm
Length- 67 mm
Width- 35 mm
Height- 18 mm
Coax Port in center of width measurement
Width Coax Port- 20 mm
Back Ridge Height- 2 mm

Snap Joints:
Length- 80 mm
Width- 8 mm
Extruded or Cut at a 45 degree angle
Flush with the top or bottom of each enclosure

Velcro Slots:
Centered on Cutouts
4 mm off the edge of Cutouts
Width- 3mm
Length- 20mm
Depth- 12mm mm

Middle Section:
Height- 60 mm
Length- 130 mm
Width- 160 mm
Shell- 4 mm

Top Section:
Height- 10 mm
Length- 130 mm
Width- 160 mm

4.1.5 Block Testing Process

1) Check if all specified parts are within the enclosure
2) Make sure all the specified parts are secure

4.1.6 References and File Links

6.1 References

Printing Options:
Corvallis 3D Printing
Tekbots 3D Printing

Modeling Software:
Fusion 360
Solidworks

Hardware Specifications:
Arduino Uno Data Sheet
Arduino Nano Data Sheet
Lipo Battery
Hamshield Documentation

4.1.7 Revision Table

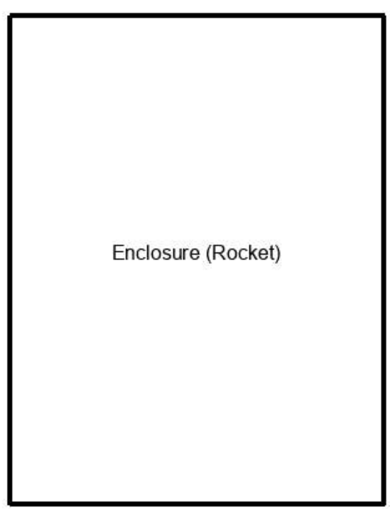
Date	Revisions Made
1/7/2022	Camden Robustelli: Put in initial information
1/18/2022	Camden Robustelli: Revised every section
1/21/2022	Camden Robustelli: Added and Revised all sections

4.2 Rocket Enclosure

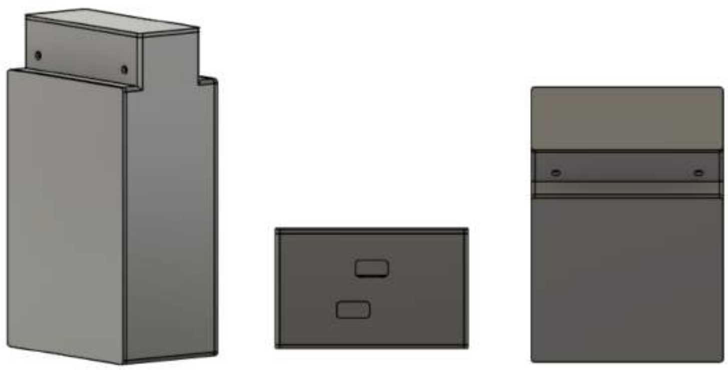
4.2.1 Block Overview

This block is the enclosure for the rocket sub system. It will house a transceiver, receiver, power supply and the microcontrollers. This enclosure will be one piece that has three different bays that are accessible by just removing the bays lid that snaps on and off. The top bay houses the power components. The first side bay houses the receiving components and the second side bay houses the sending components.The top bay will house the power supply and batteries. All the hardware components will have heat set inserts they will screw into. The enclosure will be 3D printed in PLA and be modeled on fusion 360.

4.2.2 Block Design



Fusion 360 3D Models for Enclosure:
Complete Enclosure:



Complete Enclosure Dimensions-

Height- 10.103 in
Width- 3.589 in
Length- 5.832 in
Height of Top Bay- 1.618 in
Width of Top Bay- 2.362 in
All fillet's- .125 in
Wall thickness- .118 in

Top Bay:

Above is the top bay of the enclosure that will house two lipo batteries held vertically in the two rectangle cubbies. A ¼ inch bolt will be ran through the holes above the battery to hold them in place with a piece of foam underneath the battery to remove the slop and add a little tension. Then the large flat area is where the power supply pcb will be and will be screwed into the enclosure using heat set inserts.The two rounded cutouts allow the power cables to go to both side bays of the enclosure to power all the hardware.

Top Bay Dimensions:

Battery Cubby Width- 1.959in
Battery Cubby Length- 2.126in
Power PCB Heat Set Inserts Spacing Length to Center- 1.5 in
Power PCB Heat Set Inserts Spacing Width to Center- 1.25 in
Rounded Cutouts- .5 in by .5 in
Rounded Cutouts Fillet- .125 in
Rounded Cut outs off the Battery Cubby- .125 in
Rounded Cutouts off the Wall - 1.5 in

Side Bay #1:

The first side bay will house the receiving hardware which is a arduino nano stacked on a hamshield mini. It will be screwed down into the heat inserts . Then below that there will be a switch that will be screwed down into heat inserts. Lastly at the bottom of the side bay there is a cutout for the two coax ports to be connected to cables that are ran to the two receiving antennas.

Dimensions Side Bay #1:

Everything is Centered
Everything Extruded to .375in
Top Rounded Cutout- 1.5 in by .5 in
Top Rounded Cutout is Shifted Down from Top- .125 in
Rounded Rectangle- 1.18 in by .539
Rounded Rectangle is shifted Down from Top- 2.757 in
Hamshield Mini Heat Set Inserts Center to Center- .85 in
Hamshield Mini Heat Set Inserts Shifted Down from Top- 4.825 in
Switch Heat Set Inserts Shifted Down from Top- 6.742 in
Switch Heat Set Inserts Center to Center Horizontal- 1 in
Switch Heat Set Inserts Center to Center Vertical- .813 in
Cutout for Coax From Switch Centered On Bottom Face- 1 in by .5 in with .125 in Fillet

Cutout for Coax Shifted .875 in from Center On Bottom Face- 1 in by .5 in with .125 in Fillet
Both Cutouts Shifted Vertically from the center Horizontal of the Bottom Face by .375 in to account for the extruded heat set inserts.

Side Bay #2:

The second bay houses the sending hardware which is a hamshield stacked on an Arduino Uno. They will be screwed down onto heat inserts that line up with the arduino uno pcb holes. Then below that will be the switch that is mounted on heat inserts and a cutout at the bottom for the two coax ports to be connected to cables that run to the sending antennas. Also there is a cutout at the very top of the side bay that goes to the other side bay to run wires between the two arduinos for sending and receiving.

Dimensions Side Bay #2:

Everything is Centered
Everything Extruded to .375in
Top Rounded Cutout is Shifted Down from Top- .125 in
Uno heat set inserts are shifted down from top- 2.5 in
Diagrams later show heat set insert spacing for uno
Switch heat set inserts are shifted down from top- 6.882 in
Switch heat set inserts are shifted right - .55 in
Diagrams later show heat set insert spacing for switch

Side Lids:

It will snap on so everything is contained but also is easily accessible. There will be two of these lids for the side and they will be identical

Side Lids Dimensions:

Length- 8.41 in
Width- 5.83 in
Thickness- .118 in
Offset snap part- .157 in
Offset thickness - .314 in
Cut extrude on long side is at a -45 degree angle centered 3.14 in by .314 in
Bottom lip of extrusion chamfered by .039 in
Shelled offset- .118 in

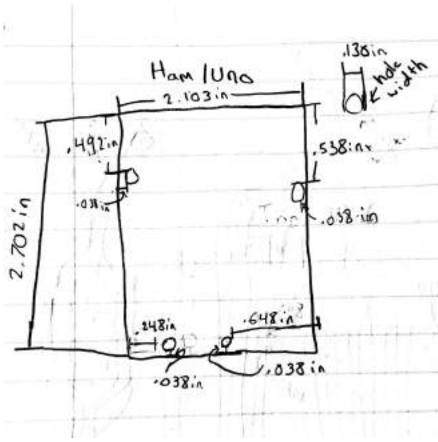
Top Lid:

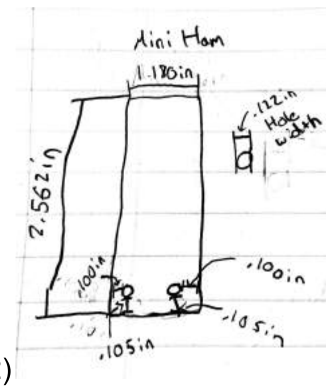
It will snap on the top so everything is contained but also is easily accessible.

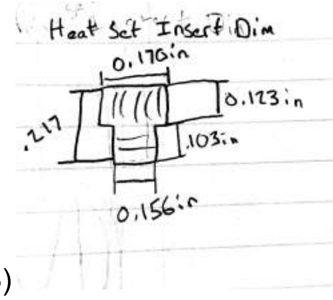
Top Lid Dimensions:

Length- 5.83 in
Width- 2.36 in
Thickness- .118 in
Offset snap part- .157 in
Offset thickness - .314 in
Cut extrude on long side is at a -45 degree angle centered 3.14 in by .314 in
Bottom lip of extrusion chamfered by .039 in
Shelled offset- .118 in

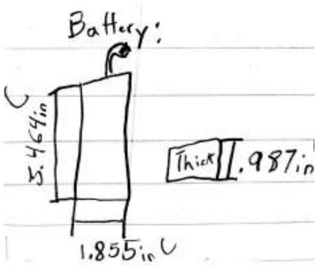
Dimensions for hardware inside of enclosure:

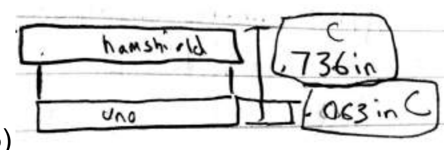
1) 

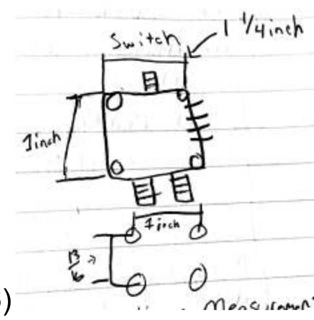
2) 

3) 

1) Location of holes on Uno and Ham and overall size
2) Location for holes on Mini Ham and overall size
3) Heat set insert size

4) 

5) 

6) 

4) Lipo battery size
5) Uno on Hamshield thickness
6) Switch size and hole location

4.2.3 Block General Validation

This block is a crucial piece of the project itself. All the components need to be protected and secure so if any accidents occur like dropping the system. This also makes the project look professional with a clean container that holds all the components in a specific place.

4.2.4 Block Interface Validation

There will be data wires coming from the sensors from the rocket. That hasn't been specified yet so it will be put in later. This port will be located at the bottom of the enclosure next to the cutout for the coax cables. The cutout for the coax cables is large enough for those wires to pass through as well so that's another option for when that needs to be implemented. The rest of the system will function internally with no other external ports except the ports for the antennas.

4.2.5 Block Testing Process

1) Check if all specified parts are within the enclosure
2) Make sure all the specified parts are secure

4.2.6 References and File Links

6.1 References

Printing Options:
Corvallis 3D Printing
Tekbots 3D Printing

Modeling Software:
Fusion 360
Solidworks

Hardware Specifications:
Arduino Uno Data Sheet
Arduino Nano Data Sheet
Lipo Battery
Hamshield Documentation
Bolt Size Chart

4.2.7 Revision Table

Date	Revisions Made
2/3/2022	Camden Robustelli: Put in initial information
2/10/2022	Camden Robustelli: Updated model pictures and made adjustments to explanations on sections as well as adding to references.
2/13/2022	Camden Robustelli: Added dimensions for all parts of the enclosure and the hardware that is going inside the enclosure.
2/18/2022	Camden Robustelli: Finalized document to submit

4.3 Data Visualizer

4.3.1

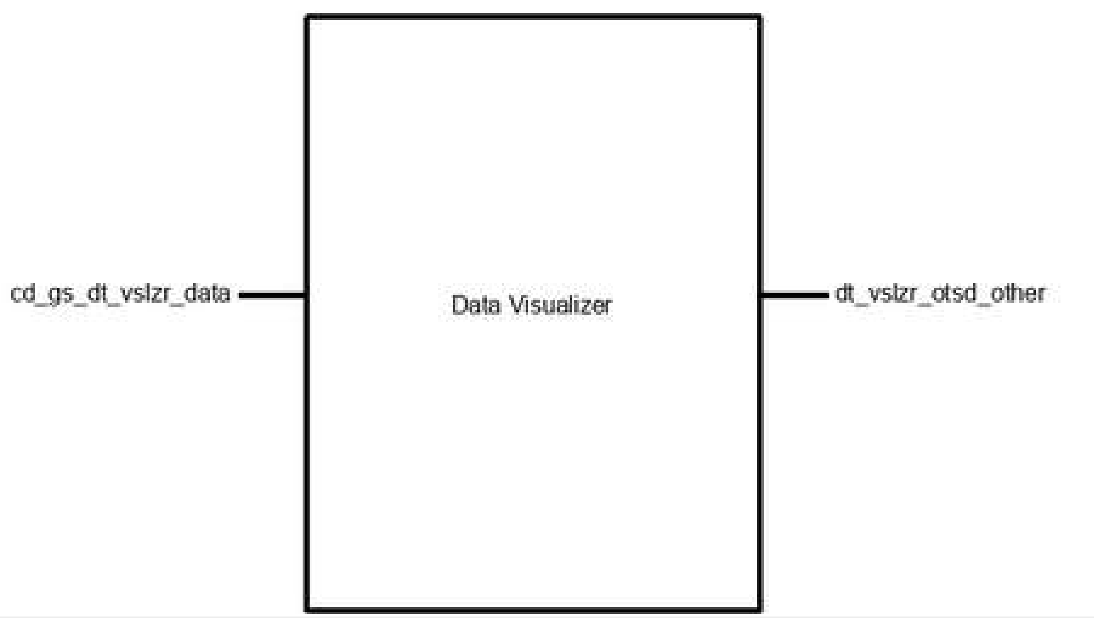
This project is a block in the HALE Communications project. This project aims to put a remote communication module that will operate with radio frequencies and send data from its sensors, such as altitude measurements from an altimeter or fuel capacity from the fuel sensor. The communication system will relay this data down to the ground station where the HALE team members will be monitoring the rocket data on an interface. This interface will be a Python script based PC app on which team members can look at the data being sent in on a graph in real time. It is an advantageous part of the system as it allows HALE team members to easily interpret data, since it is displayed in a visual way. This interface will also fulfill one of the universal constraints: ***“the final system must contain both of the following: a student designed PCB and a custom Android/PC/Cloud application.”***

This block will be installed on a computer that is connected to the ground station via a USB cable. Data will be transferred serially at a baud rate of 9600 between the Arduino that is receiving data from the rocket. The rate of numbers coming through is far less. Only 1 data point per second will be transmitted from the rocket as that will be a large enough data rate for the HALE team to get consistent updates on their rockets. The HALE team never specified any specific data rate (i.e. how many points of data they want per second), so 1 data point per second is assumed. If the HALE team wants to change the data rate, it can be done easily.

4.3.2

For this test, an Arudino script that generates one random number between 0 and 300 per second will be loaded onto an Arduino Uno. The random number generator will only be used for this test and will simulate the data stream being received by radio, as the receiving and transmitting parts of the system are not done yet. This Arduino Uno will be connected to the computer via USB and a Python script will run and collect these numbers and graph them.

Block image:



Pseudocode -

Random Number Generator (for only verifying block 1, not in the actual system)-
Setup:
Initialize baud rate of 9600
Seed random number generator
Loop:
Generate random number
Print random number
Send it to computer via USB (with Serial.println print and send are the same command)
delay 1 second

Data visualization interface (graphing application)-
Setup:
Initialize baud rate of 9600
Set up serial connection with computer
Clear the output file
Loop:
Receive integer
Convert it to character form
Append it to the output file

Loop – While end of file not reached:
Read in line of file
Convert the char to integer
Place the integer on the graph

Graph updates with all the data
Delay 1 second

4.3.3

This graphing application will be made in the Python scripting language. Specifically, the matplotlib and the pyserial libraries were installed and used. Pyserial is a python library that allows python to access a serial data stream and perform operations. In this case, the code will take the numbers (in integer form) and place them in an output text file. Matplotlib is a graphing tool in Python that allows data to be plotted live. That means the more data is being plotted on the graph and you can witness any changes in the data while the rocket is flying and not after the flight. It takes values out of the text file and graphs them as they are being entered in and saved.

These code libraries are a good choice for this task as they are specifically designed to do this. The code itself is simple. The test random number generator is essentially taken care of in one line of code. This command is random(x), and the number generated is between 0 and x. The Matplotlib has already been tested and is shown to graph all numbers in an output text file. Getting this to run automatically should only take another command or two as Python functions are very powerful and a lot can be done with them. Also, the documentation on Python is extensive, so if any other issues come up, they should be easy to fix with a quick internet search.

The main reason the random number generator is used is to simulate data coming into the ground station from the rocket module. This is a good simulation as the data being sent from the sensors on the rocket will be in integer form and will be all sorts of values. Although, technically the messages sent on the radio waves will be in character form. However, extra code in the Ground Station code block can convert these characters back into integers. So the input into this block will still be the same.

4.3.4

Interface: cd_gs_dt_vslzr_data

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Baud Rate: 9600 baud	The rocket will rotate along its vertical axis approximately 20 times a second, leaving only 1/40th of a second for an antenna on one side of the rocket exposed to communication. The message must be received by the antenna within this time. At this bit rate, the antenna can receive 240 bits or 30 chars or 7 full 32 bit integers in this time frame. At this point, there is no specific reason as to why this baud rate is used, other than it is a very common and reliable baud rate for communication. Other baud rates could be used, as well.	In the program, the baud rate can be set with a single command. The Arduino will interpret this command and set the baud rate appropriately.
Other: Connection with a USB	In order to collect data from the rocket, the HALE team will need some place to store and display it. That place will be on a computer. The most common type of connection port on the computer is a USB, so this block will use a USB connection.	From past experience (in Junior Design) data can be streamed from any Arduino microcontroller to a computer via the proper USB cable and adapter. All arduinos come with this cable. The Arduino Uno specifically comes with a USB 2.0 cable. So it is assumed that it will use the USB 2.0 protocol.

Messages: Receives integers, which represent the data sent by the rocket.	The data will be numerical. Whether it is represented with floating point or integers will depend on the sensor that it is collecting data from. Since none of the rocket hardware has been made by the HALE rocket team, the document assumes 32-bit unsigned integers. This is type int in C, which is the language used by Arduino. This can easily be switched to another data-type, such as a signed 32 bit int or otherwise.	In both Python and C, which the Arduino uses, one can declare an integer with a keyword. So it is quite simple to create integers in any piece of code and send them somewhere else.

Interface: dt_vslzr_otsd_other

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Other: Produces Graph	The main output of the block is a scatter plot graph, with lines connecting each of the points. The point of the graph is to give the HALE team members a visual representation of data they are collecting. It will display the graph on the computer screen to which the Arduino is connected to.	This property will be verified visually, so it should be easy enough to verify. The graph should have two labeled axes, a title and should have thick enough data points and lines that users can easily see that data from at least 5 feet away from the computer screen.
Other: Updates graph with at least one data point per second.	This is not the same as baud rate. The number of data points sent from the rocket will likely be limited to a few numbers per second. In this case, at least one per second is being tested with. This seems like enough data to collect over time. The HALE team did not request a specific data rate, so one point per second is to be	In the Arduino and Python programs used, a certain delay factor can be added in so that the Arduino sends one number every second over USB. The delay(1000) command can be used to delay the program for 1 second.

	assumed.	
Produces a .txt file with data	This text file will place incoming data into it. It will also be used by the graph to pull data from. This data will be displayed on the graph.	Python has built in functions that deposit data into text files. It will be known if it is implemented properly when the data points on the graph match the integer values in the .txt sheet and if the values on the graph are in the same order from left to right as the order from top to bottom in the text file.

4.3.5

Interface: cd_gs_dt_vslzr_data

Property – Baud Rate: 9600 baud

1. In order to verify that the baud rate is actually at 9600 bps, first connect the Arduino Uno to the computer via a USB cable. Flash the test code (random number generator) onto the Uno. Make sure the baud rate in the Arduino Uno code is set to 9600 bps.
2. Open an application on the computer which has a terminal that can vary the baud rate. Set the baud rate to a different baud rate (not 9600 bps).
3. Start test code on the Uno (there should be an option on the Arduino IDE to start it) and watch the terminal on the computer. The characters streaming onto the terminal should look like gibberish. This is to be expected since the baud rates do not match. Do this step as many times with as many different baud rates if needed.
4. Now set the baud rate on the computer terminal to 9600 bps and run the code on the Arduino once more. The data stream being displayed on the terminal should match the numbers being randomly generated in the Arduino. This will confirm that the Arduino has the baud rate set to 9600 bps.

Property – Other: Connection with USB 2.0

1. Connect the USB port of the Arduino with the USB port of the computer with a USB 2.0 cable.
2. Begin the Arduino code and the Python script.
3. Stop both after about 10 seconds.
4. Look at and compare the numbers printed out in the Arduino IDE and in the output text file on the computer.
5. If all the numbers in the Arduino IDE and the output text are the same and in the same order then this confirms that transmission did occur properly and that the USB connection is valid and verified.
6. If one wants to do in depth verification that the connection is USB, they may download a packet sniffer that can be used to look at USB protocol packets. In which case, activate the packet sniffer while running the code. If the packet sniffer detects the USB 2.0 protocol packets and the data running through, then the property is verified. However,

the first 5 steps should be enough confirmation.

Property – Messages: Receives 32 bit integers

1. Set the number in the random(x) command in the Arduino random number generator to a number less than 10.
2. Generate two numbers in sequence. For example 9 and 6.
3. Have the Python script add the two numbers before converting them and appending them to the file.
4. Print the result. It should be 15 in this example. If so, then this property is verified. If not, then the program is sending another data type over.

Interface: dt_vslzr_otsd_other

Property – Other: Produces Graph

1. Flash the test code onto the Arduino. Open up the Python script on the computer which is connected to the Arduino.
2. Make sure the Arduino and computer are connected via USB.
3. Run the code on the Arduino and start the Python script.
4. If a graph is produced on the computer automatically, a figure with two axes and a title, then this property is verified. Data doesn't necessarily have to show up on the graph as that will be verified as another property.

Property – Other: Updates graph with at least one data point per second

1. Three things need to be started within as small a time frame as possible. First the Arduino code needs to be started. Then the Python script needs to be started.
2. Make sure the graph is displaying values in real time.
3. Next, turn on a timer (either on the computer or another timer, such as a watch or smartphone timer). If possible, have someone else start the timer at the same time as the Python script.
4. Run Arduino script for 10 seconds. Then end it by clicking stop in the Arduino IDE or pulling the USB cable out.
5. Count how many integers have been plotted on the graph or pull up the output text doc and count how many integers are in there.
6. If there are more than 10 integers, then this implies that the data rate exceeded 1 number per second and so the property is verified.

Property – Other: Sends data into a .txt file

1. The Python script will have an output file name of 'example.txt'.
2. Run the Python script and the Arduino script.
3. After 10 seconds stop the Arduino script.
4. Compare the numbers printed out in the Arduino IDE to the numbers printed out in the text document. If all the numbers in the text document are the same as the numbers printed out by the random number generator, then this property is verified.

4.3.6: References and Links

[1] “Hamshield,” *HamShield - Arduino Reference*. [Online]. Available: https://www.arduino.cc/reference/en/libraries/hamshield/. [Accessed: 08-Jan-2022].

[2] “Pyserial,” *PyPI*. [Online]. Available: https://pypi.org/project/pyserial/. [Accessed: 21-Jan-2022].

[3] “Visualization with Python,” *Matplotlib*. [Online]. Available: https://matplotlib.org/. [Accessed: 21-Jan-2022].

4.3.7

Revisions Table

1/7/22	Added initial content – Author: Sam Wagner
1/17/22	Changed block to something easier and revised section 4.1.1, 4.1.2 and 4.1.3 -Author: Sam Wagner
1/19/22	Filled in more of 4.1.4 -Author: Sam Wagner
1/21/22	Finished 4.1.2, 4.1.3, 4.1.4 and 4.1.5, Addressed all the feedback: such as adding image of block diagram, showing where baud rate is set in the pseudocode, added the missing interface name, explained reason for baud rate, etc. 4.1.6: Added more citations -Author: Sam Wagner (for all these edits)

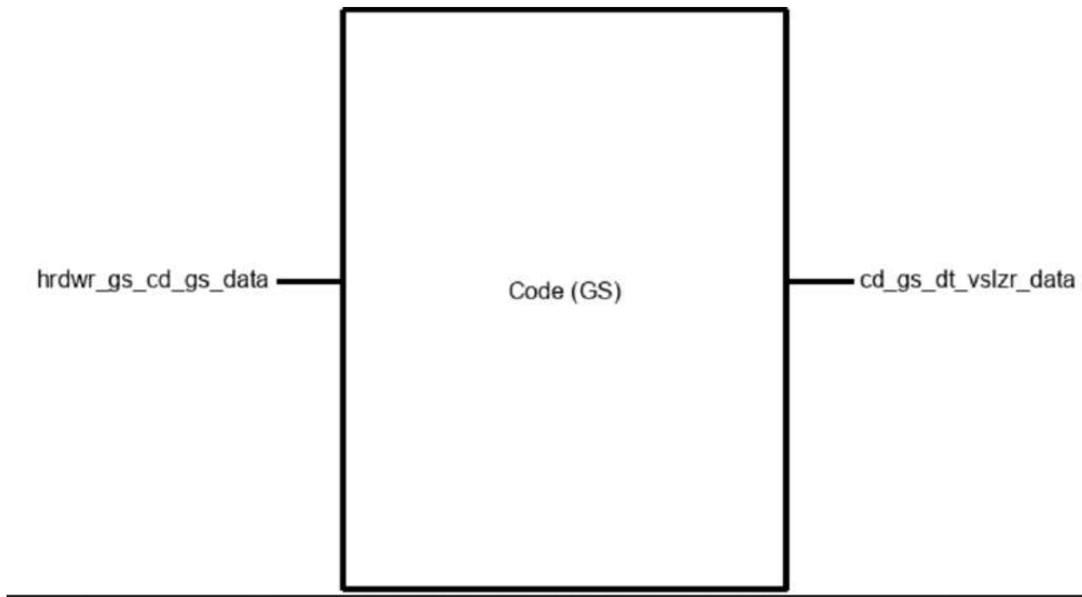
4.4 Code for Ground Station

4.4.1

The code for the Ground Station should operate the transceiver HamShield. The Hamshield is a pre-made board which transmits and receives messages on the amateur radio bands. It is also able to connect to an Arduino, which has the hardware to support the code. The Hamshield and code should have to act as a downlink, receiving information from the rocket. This block will feed into the data visualizer block.

The Hamshield board and the Arduino Uno will be attached together via pins and the Uno will be attached to the computer via a USB 2.0 cable. This will allow the user to see what messages they get on the serial monitor or a similar tool. In summary, the basic purpose of this block is to capture and print out any messages that the Hamshield/Arduino pair receive from the radio waves.

4.4.2



Pseudocode -

```
Set up:
    Initialize sensitivity and transmission frequency
    Set baud rate to 9600
    Set mode to receive

Loop continuously:
        If an ASCII character arrives
            Interpret what ASCII character was received
            If no string exists
                Create new string
            Else
                If string reaches 10 chars long
                    Print string
                Else
                    Append character to a string
        Else
            Do nothing
```

4.4.3

The block is supposed to operate the Hamshield. The block will allow the Hamshield board to detect and receive AFSK waves that have been sent by the rocket. A small rubber ducky antenna is attached to the coax input of the Hamshield. The Arduino Hamshield pair is connected via USB to a computer where a serial monitor will be running and printing out the characters it receives. On the other end, a Hamshield Arduino pair set to transmit will be sending out characters for the block to receive. The data transmitted will be in the form of ASCII characters. ASCII will be used in this case because it is easy to print and also a wide variety of symbols, such as alphabetical and numerical symbols, can be sent in this format. ASCII is also used in C strings, which are easy to take apart and put back together again. This feature is very necessary as data is sent one character at a time.

The code will be written with functions from the special HamShield C library. There is some documentation on these functions; however, less than what most people are comfortable with. In addition to the little documentation, there is a piece of sample code that is actually very well suited for our project. This sample code will be used to base our own Ground Station and Rocket code off of.

Testing will be conducted in the following general manner. The code for the ground station will be loaded onto the receiving HamShield Arduino pair and another edited piece of code for the transmitting HamShield will be uploaded to the HamShield Arduino pair on the other end. This setup will simulate what the interaction between the Ground Station and the Rocket will be like. Two computers will be set up with the Arduino IDE and serial monitors. These monitors will keep track of the incoming and outgoing streams of data from each side. The serial monitors allow for an easy way to look and confirm if the message received matches the message sent. The USB interface between the computer and the receiving HamShield will be the cd_gs_dt_vslzr_data interface, while the wireless transmission between the two HamShields will in effect serve as the hrdwr_gs_cd_gs_data (this is because the hardware passes the wireless signal onto the code block). In order to test if the block is functioning correctly, a known message, such as a string of integers, is typed out into the Arduino IDE terminal and sent off. If the message sent matches the message received on the other end, then the block is working.

4.4.4

Interface: hrdwr_gs_cd_gs_data

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Baud Rate: 9600 baud	The rocket will rotate along its vertical axis approximately 20 times a second, leaving only 1/40th of a second for an antenna on one side of the rocket exposed to communication. The message must be received by the antenna	In the program, the baud rate can be set with a single command. The Arduino will interpret this command and set the baud rate appropriately.

	within this time. At this bit rate, the antenna can receive 30 ASCII characters without interruption. That is more characters than what will be contained within a single data packet.	
Other: Uses coax input to connect to the antenna	Transmission lines are used to send analog signals. Coax lines are commonly used as transmission lines. The coax input is used to connect the coax lines to the Hamshield. The AFSK waveform that the	The Hamshield already comes with the standard female SMA coax input. It has a radius of about 1mm.
Messages: Received characters from the transmitting rocket module.	The data coming from the rocket will be on two bands of radio waves, either VHF or UHF. The code needs to set the Hamshield to listen in on these frequencies and pick up the ASCII characters it hears on these lines.	The Hamshield library has a command to set the frequency it should listen to.

Interface: cd_gs_dt_vslzr_data

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Baud Rate: 9600 baud	The rocket will rotate along its vertical axis approximately 20 times a second, leaving only 1/40th of a second for an antenna on one side of the rocket exposed to communication. The message must be received by the antenna within this time. At this bit rate, the antenna can receive 30 ASCII characters without interruption. That is more characters than what will be contained within a single	In the program, the baud rate can be set with a single command. The Arduino will interpret this command and set the baud rate appropriately.

	data packet.	
Other: Connection with a USB	In order to collect data from the rocket, the HALE team will need some place to store and display it. That place will be on a computer. The most common type of connection port on the computer is a USB, so this block will use a USB connection.	From past experience (in Junior Design) data can be streamed from any Arduino microcontroller to a computer via the proper USB cable and adapter. All arduinos come with this cable. The Arduino Uno specifically comes with a USB 2.0 cable. So it is assumed that it will use the USB 2.0 protocol.
Messages: Sends integers, but in ASCII form, which represent the data sent by the rocket.	The data will be numerical, but represented with ASCII chars.	The command Serial.println is the easiest way to send data down the serial pipeline. It produces ASCII chars instead of actual numbers which is why chars are used instead.

4.4.5
Interface: hrdwr_gs_cd_gs_data

Property – Baud Rate: 9600 baud

1. In order to verify that the baud rate is actually at 9600 bps, first connect the Arduino Uno to the computer via a USB cable. Flash the test code (random number generator) onto the Uno. Make sure the baud rate in the Arduino Uno code is set to 9600 bps.
2. Open an application on the computer which has a terminal that can vary the baud rate. Set the baud rate to a different baud rate (not 9600 bps).
3. Start test code on the Uno (there should be an option on the Arduino IDE to start it) and watch the terminal on the computer. The characters streaming onto the terminal should look like gibberish. This is to be expected since the baud rates do not match. Do this step as many times with as many different baud rates if needed.
4. Now set the baud rate on the computer terminal to 9600 bps and run the code on the Arduino once more. The data stream being displayed on the terminal should match the numbers being randomly generated in the Arduino. This will confirm that the Arduino has the baud rate set to 9600 bps.

Property – Uses coax input to connect to the antenna

1. Get a coax line with an antenna attached at one end.
2. Fit it into the input on the Hamshield board
3. If the connector fits and the Hamshield is able to receive messages off the air from the antenna, then this property is verified. What the chars are doesn’t matter.

Property – Received characters from the transmitting rocket module.

1. First set the transmitting Hamshield pair to a frequency between 144 MHz and 148 MHz.
2. Have it send out a consistent repeating message.
3. Set the receiving Hamshield to a frequency outside of 144 MHz and 148 MHz. Expect to

see gibberish on the serial terminal.

4. Now switch the frequency on the receiving Hamshield to match the frequency of the transmitting Hamshield.
5. If the characters received on the terminal match those being sent out, then this interface property is confirmed.

Interface: cd_gs_dt_vslzr_data

Property – Baud Rate: 9600 baud

1. In order to verify that the baud rate is actually at 9600 bps, first connect the Arduino Uno to the computer via a USB cable. Flash the test code (random number generator) onto the Uno. Make sure the baud rate in the Arduino Uno code is set to 9600 bps.
2. Open an application on the computer which has a terminal that can vary the baud rate. Set the baud rate to a different baud rate (not 9600 bps).
3. Start test code on the Uno (there should be an option on the Arduino IDE to start it) and watch the terminal on the computer. The characters streaming onto the terminal should look like gibberish. This is to be expected since the baud rates do not match. Do this step as many times with as many different baud rates if needed.
4. Now set the baud rate on the computer terminal to 9600 bps and run the code on the Arduino once more. The data stream being displayed on the terminal should match the numbers being randomly generated in the Arduino. This will confirm that the Arduino has the baud rate set to 9600 bps.

Property – Other: Connection with USB

1. Connect the USB port of the Arduino with the USB port of the computer with a USB 2.0 cable.
2. Begin the Arduino code and the Python script.
3. Stop both after about 10 seconds.
4. Look at and compare the numbers printed out in the Arduino IDE and in the output text file on the computer.
5. If all the numbers in the Arduino IDE and the output text are the same and in the same order then this confirms that transmission did occur properly and that the USB connection is valid and verified.
6. If one wants to do in depth verification that the connection is USB, they may download a packet sniffer that can be used to look at USB protocol packets. In which case, activate the packet sniffer while running the code. If the packet sniffer detects the USB 2.0 protocol packets and the data running through, then the property is verified. However, the first 5 steps should be enough confirmation.

Property – Sends integers, but in ASCII form, which represent the data sent by the rocket.

1. One can visually verify that there are integers being sent. If one sees only whole numbers being sent and the data type in the code is char, then this property is confirmed. If one wants to do something more thorough, follow the next steps.
2. Print out a string as a row of characters.
3. Next print it out as a row of integers.
4. If the integers match the ASCII code for each character, this will confirm that ASCII chars

are being used.

4.4.6: References and Links

[1] "Hamshield," *HamShield - Arduino Reference*. [Online]. Available: https://www.arduino.cc/reference/en/libraries/hamshield/. [Accessed: 08-Jan- 2022].

4.4.7:

Revisions Table

1/7/22	Added initial content. -Author: Sam Wagner
2/3/22	Reusing the original block draft as this was supposed to be my first block but I changed to another block. Now this will be my second block. -Author: Sam Wagner
2/4/22	Filling in details, albeit a bit messy still -Author: Sam Wagner
2/18/22	Changed block diagram and adding a few more details to each section -Author: Sam Wagner

4.5 Ground Station Antenna System

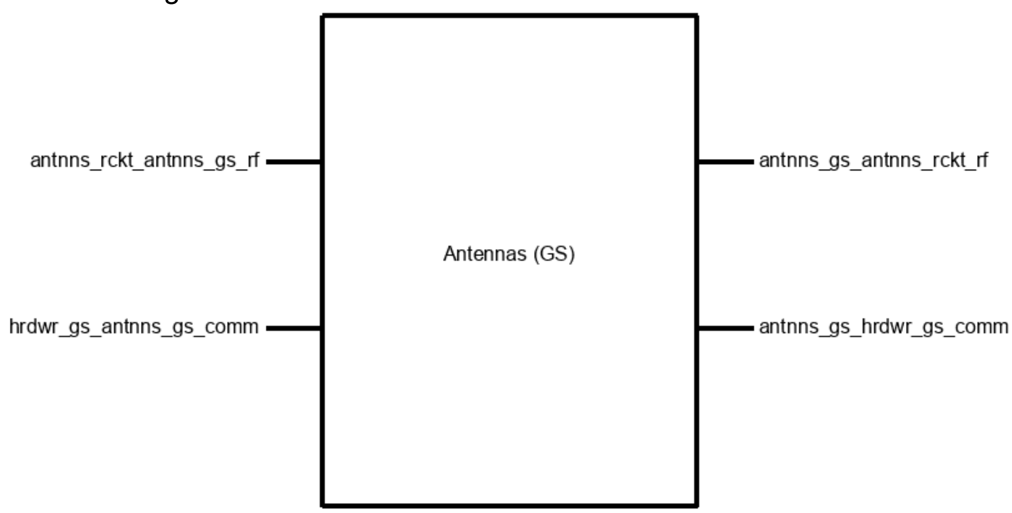
4.5.1 Block Overview

This block entails the design and manufacture of a Quagi antenna system for the Ground Station. The antenna system will consist of two single-band Quagi antennas, one for Ultra High Frequency (UHF), and one for Very High Frequency (VHF). Each antenna will be connected to its own individual HamShield transceiver, with the VHF Quagi antenna being used for dedicated uplink communication and the UHF Quagi antenna being used for dedicated downlink communication.

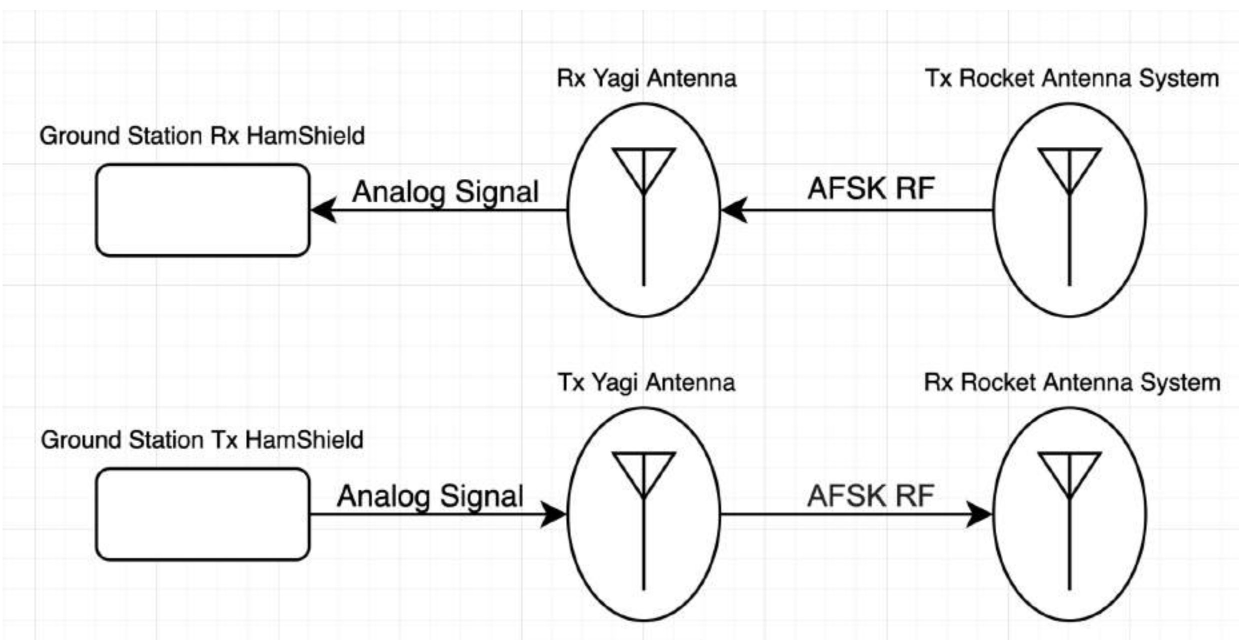
Each antenna will be composed of copper wires and rods with a non-conductive central boom. Both antennas will have a reflector and a driven element, however, the VHF antenna will have 3 directors with a boom length of 209cm, while the UHF antenna will have 6 directors with a boom length of 137.5cm. The VHF antenna will use 12 AWG (2mm) copper wire for the reflector and driven elements and 3/16-inch (5mm) copper rods for the directors. The UHF antenna will use 14 AWG (1.6mm) copper wire for the reflector and driven elements and 1/8-inch (3mm) copper rods for the directors. The antennas will be connected to the HamShields using a 50Ω coaxial cable transmission line. Completion of the block will result in a fully functional, dual-band ground station communication system. This block will be designed and built by Kevin Kott.

4.5.2 Block Design

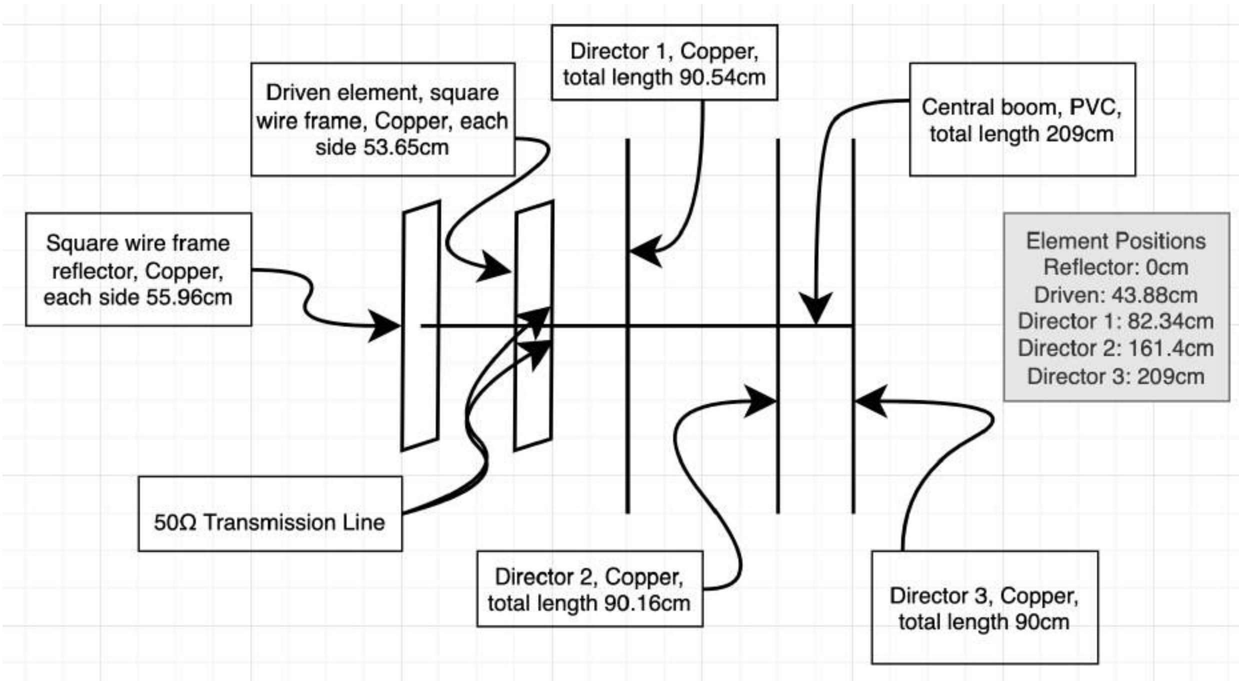
Black Box Diagram



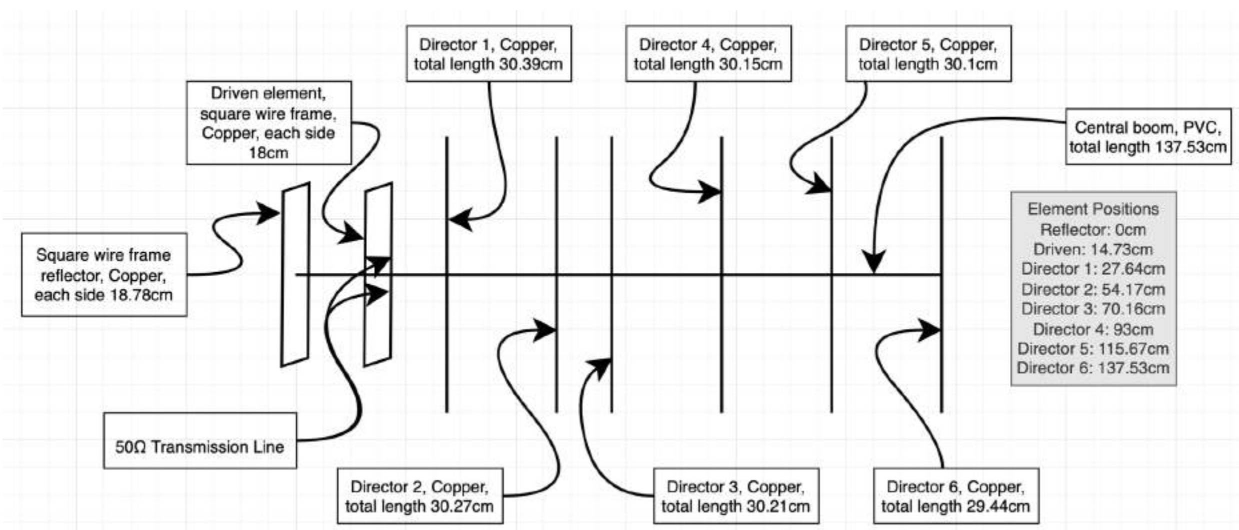
Data Flow chart



VHF Quagi Antenna



UHF Quagi Antenna



4.5.3 Block General Validation

The communications system will be operating over a large distance meaning antenna gain becomes the most important design characteristic. Based on gathered information from conducted research, the Yagi-Uda antenna provides the greatest gain out of all the other antenna types, at the expense of size. Since this block pertains to the ground station antenna system, size and spatial constraints are limited, meaning the antennas can be built to fully optimize gain over compatibility. One characteristic of a Yagi antenna is that the bandwidth is narrow, however, this is of no concern since the ground station antennas will be operating within a single band over narrow frequency ranges. Another bonus to this type of antenna design is that the cost of materials and the difficulty of manufacture are low. However, there are some drawbacks to using a Yagi antenna. The Yagi antenna inherently has a low input impedance, which means that an antenna with a simple Yagi design will not be efficient at transmitting or receiving a radio signal. The antenna will have a high standing wave ratio and lead to poor performance. There are solutions to this issue, like bending the director downward in a V-like configuration, however, this reduces the antenna's gain. Another solution would be to utilize a balun, however, these are expensive and must be tuned properly. Gamma matching seems to be the technique that most use to impedance match Yagi antennas, however, this is a tedious and difficult process that is prone to failure and doesn't result in an adequate gain. The solution to the issue at hand is to combine the desired characteristics of two different antenna types; the impedance of a quad antenna, and the gain of a Yagi antenna. The result is what is known as a Quagi antenna. Quagi antennas are known for being extremely efficient at UHF and VHF operating frequencies, as well as having input impedances around 50Ω. This will help to keep the standing wave ratio of the antennas close to 1, limit power loss to the antenna, and maintain adequate gain for long-distance radio communication. However, there is an issue with this antenna design, which is that there is not a general formula for the size and spacing of the antenna's elements. To address this issue, known working antenna designs will be built, and the antenna's operating characteristics will be assessed using a vector network analyzer and changed accordingly.

4.5.4 Block Interface Validation

Hrdwr_GS_Antnns_GS_Comm

Pmin: 165.3mW	The minimum power was chosen based on the maximum standing wave ratio which was selected to achieve a distance of double that of the Karman line	**HamShield 1.0 Specifications** Power Output: 0.5W

Pnom: >445mW	The nominal power was chosen based on the HamShield's listed power output and a standing wave ratio of 2	**HamShield 1.0 Specifications** Power Output: 0.5W
Znom: 50Ω	The nominal impedance was chosen based on the HamShield's output impedance	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
Zdev: +/-50%	The impedance deviation was chosen based on the resulting standing wave ratio. Antennas with an impedance deviation of greater than 50% are poor performers	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
VSWRmax: 10	The maximum standing wave ratio was chosen based on a worst-case scenario for Tx and Rx gain, and a reserved link budget of 25dB which results in a max range of greater than twice than the distance to the Karman line	**SWR Calculator** VSWR: 10 Reflected Power: 66.94% Mismatch Loss: 4.807dB **RF Calculator** Tx Frequency: 146MHz Adj. Transmitted Power: 165.3mW Min Tx Gain: 1dBi Tx Cable Loss: 0.5dB Min Rx Gain: 1dBi Rx Cable Loss: 0.5dB Link Budget: 25dB Free Space Path Loss: 122.2dB Max Range: 208.8km

Antnns_GS_Antnns_Rckt_RF
(Quagi Tx Antenna)

Tx Gmin: 1dBi	The minimum Tx antenna gain was chosen based on minimum Rx antenna gain, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 363.2km
Tx Gnom: >=5dBi	The nominal Tx antenna gain was chosen based on typical gain for selected antenna type	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Nom Tx Gain: 5dBi Tx Cable loss: 0.5dB Nom Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 131.0dB Max Range: 575.6km

Rx Gmin: 1dBi	The minimum Rx antenna gain was chosen based on minimum Tx antenna gain, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 363.2km
Rx Gnom: >=1dBi	The nominal Rx antenna gain was chosen based on typical gain for selected antenna type	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Nom Tx Gain: 5dBi Tx Cable loss: 0.5dB Nom Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 131.0dB Max Range: 575.6km

Frequency: 144-148 MHz	This frequency was chosen based on the available transmitting frequencies of the HamShield and the limited frequency ranges allowed for amateur Ham radio Licensees	**HamShield 1.0 Specifications** Amateur Radio Bands: 2m, 1.25m, 70cm Frequency RX/TX: 134-174MHz, 200-260MHz, 400-520MHz **US Frequency Allocation** Band: Very High Frequency Source: Amateur Frequencies (MHz): 50-54, 144-148, 216-220, 222-225

Antnns_GS_Hrdwr_GS_Comm

Znom: 50Ω	The nominal impedance was chosen based on the HamShield's output impedance	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
Zdev: +/-50%	The impedance deviation was chosen based on the resulting standing wave ratio. Antennas with an impedance deviation of greater than 50% are poor performers	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
VSWRmax: 3.3	The maximum standing wave ratio was chosen based on a worst-case scenario for Tx and Rx gain, and a reserved link budget of 25dB which results in a max range of 3% greater than the distance to the Karman line	**SWR Calculator** VSWR: 3.3 Reflected Power: 28.61% Mismatch Loss: 1.4636dB **RF Calculator** Tx Frequency: 435MHz Adj. Tx Power: 356.98mW Min Tx Gain: 1dBi Tx Cable Loss: 0.5dB Min Rx Gain: 1dBi Rx Cable Loss: 0.5dB Link Budget: 25dB Free Space Path Loss: 125.5dB Max Range: 103.0km

Antnns_Rckt_Antnns_GS_RF
(Quagi Rx Antenna)

Rx Gmin: 1dBi	The minimum Rx antenna gain was chosen based on minimum Tx antenna gain, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 121.9km
Rx Gnom: >=7dBi	The nominal Rx antenna gain was chosen based on typical gain for antenna type	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Nom Tx Gain: 1dBi Tx Cable loss: 0.5dB Nom Rx Gain: 7dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 133.0dB Max Range: 243.2km

Tx Gmin: 1dBi	The minimum Tx antenna gain was chosen based on minimum Rx antenna gain, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 121.9km
Tx Gnom: >=1dBi	The nominal Tx antenna gain was chosen based on typical gain for selected antenna type	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Nom Tx Gain: 1dBi Tx Cable loss: 0.5dB Nom Rx Gain: 7dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 133.0dB Max Range: 243.2km

Frequency: 420-450 MHz	This frequency was chosen based on the available transmitting frequencies of the HamShield and the limited frequency ranges allowed for amateur Ham radio Licensees	**HamShield 1.0 Specifications** Amateur Radio Bands: 2m, 1.25m, 70cm Frequency RX/TX: 134-174MHz, 200-260MHz, 400-520MHz **US Frequency Allocation** Band: Ultra High Frequency Source: Amateur Frequency (MHz): 420-450, 902-928, 1240-1300, 2300-2310, 2390-2450

4.5.5 Block Testing Process

Hrdwr_GS_Antnns_GS_Comm
For this interface, there are four properties.

Power
To verify the transmitted power, the antenna will be hooked up to a vector network analyzer and the standing wave ratio will be documented. The standing wave ratio will determine how much power is being transmitted through the transmission line to the antenna. The target power transmission of 445mW or greater will be considered nominal, with a minimum power transfer of 165.3mW considered a passed test. The power transmitted will be calculated using the antenna's standing wave ratio and a theoretical input power of 500mW.

Steps
1. The transmitting antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antennas via an S11 test
3. Test results will be reviewed and antenna standing wave ratio will be recorded
4. The antenna's standing wave ratio will be inputted into the VSWR calculator listed in the references section
5. A nominal standing wave ratio of 2 correlates to a transmitted power of 445mW
6. The test will be considered passed if the standing wave ratio is less than 10, correlating to a transmitted power of 165.3mW

Impedance and deviation
The nominal impedance for the antenna is 50Ω and will be verified using a vector network analyzer. Verification will be considered complete when the vector analyzer shows an impedance deviating less than 50% of the target impedance.

Steps
1. The transmitting antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's impedance will be recorded
4. The test will be considered passed if the impedance does not deviate more than 50%

Standing Wave Ratio
The max standing wave ratio (SWR) for the antenna is 10 and will be verified using a vector network analyzer. The antenna will be hooked up to the testing port of the analyzer, a test signal will be sent through the antenna from and to the analyzer, and the SWR will be calculated with the analyzer. Verification will be confirmed if the SWR is less than 10.

Steps
1. The transmitting antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed and antenna standing wave ratio will be recorded
4. The test will be considered passed if the standing wave ratio is less than 10

Antnns_GS_Antnns_Rckt_RF
For this interface, there are three properties.

Tx Gain
The minimum gain of the antenna must be no less than 1 dBi. The gain of the antenna will be verified using a vector network analyzer (VNR) by connecting the antenna to the VNR and sending a signal at the operating frequency. The gain will be listed in the test results and have a gain of no less than 1dBi. The nominal gain of the antenna will be greater than or equal to 5dBi. The gain of the antenna will be verified using a vector network analyzer (VNR) by connecting the antenna to the VNR and sending a signal at the operating frequency. The gain will be listed in the test results and considered nominal if no less than 5dBi.

Steps
1. Two antennas with operating frequencies between 136MHz and 176MHz will be connected to the VNA at ports S11 and S21

2. The antennas will be placed a certain distance away from each other and the return loss will be recorded from the S21 port
3. The antenna under test (AUT) will then be connected to the Vector Network Analyzer at the S21 port, replacing the known antenna, using a coaxial cable
4. The gain of the antenna under test will then be recorded
5. The AUT's testing gain will then be subtracted from the known antenna's testing gain giving us a reference gain
6. The reference gain will be added to the known antennas known gain giving us the dBi of the AUT
7. The test will be considered passed if the gain is greater than 1dB with a nominal gain of 5dBi

Rx Gain

The Rx gain of the VHF channel cannot be tested or verified until the rocket antenna system is completed. For the time being, the worst case scenario gain for the VHF Rx Antenna system will be used.

Operating Frequency

The operating frequency of the antenna will be set to 146MHz but may vary between 144MHz and 148MHz. To verify this property, the antenna will be connected to a VNA and swept through a frequency range of 1Hz-1GHz, and the resulting frequency response will be generated. Verification will be considered complete when the frequency response shows a dip between 144-148MHz.

Steps

1. The receiving antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's frequency response will be noted
4. The test will be considered passed if the frequency response shows a significant extreme at the intended operating frequency between 144-148MHz
5. The test will be conducted again for the second receiving antenna
6. The test will be considered passed if the frequency response shows a significant extreme at the intended operating frequency between 144-148MHz

Antnns_GS_Hrdwr_GS_Comm

For this interface, there are three properties.

Impedance and deviation

The nominal impedance for the antenna is 50Ω and will be verified using a vector network analyzer. Verification will be considered complete when the vector analyzer shows an impedance deviating less than 50% of the target impedance.

Steps

1. The receiving antenna will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's impedance will be recorded
4. The test will be considered passed if the impedance does not deviate more than 50%

Standing Wave Ratio

The max standing wave ratio (SWR) for the antenna is 3.3 and will be verified using a vector network analyzer. The antenna will be hooked up to the testing port of the analyzer, a test signal will be sent through the antenna from and to the analyzer, and the SWR will be calculated with the analyzer. Verification will be confirmed if the SWR is less than 3.3.

Steps

1. The receiving antenna will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed and antenna standing wave ratio will be recorded
4. The test will be considered passed if the standing wave ratio is less than 3.3

Antnns_Rckt_Antnns_GS_RF

For this interface, there are three properties.

Rx Gain

The minimum gain of the antenna must be no less than 1 dBi. The gain of the antenna will be verified using a vector network analyzer (VNR) by connecting the antenna to the VNR and sending a signal at the operating frequency. The gain will be listed in the test results and have a gain of no less than 1dBi. The nominal gain of the antenna will be greater than or equal to 7dBi. The gain of the antenna will be verified using a vector network analyzer (VNR) by connecting the antenna to the VNR and sending a signal at the operating frequency. The gain will be listed in the test results and considered nominal if no less than 7dBi.

Steps

1. Two antennas with operating frequencies between 400MHz and 470MHz will be connected to the VNA at ports S11 and S21
2. The antennas will be placed a certain distance away from each other and the return loss will be recorded from the S21 port
3. The antenna under test (AUT) will then be connected to the Vector Network Analyzer at the S21 port, replacing the known antenna, using a coaxial cable
4. The gain of the antenna under test will then be recorded

5. The AUT's testing gain will then be subtracted from the known antenna's testing gain giving us a reference gain
6. The reference gain will be added to the known antennas known gain giving us the dBi of the AUT
7. The test will be considered passed if the gain is greater than 1dBi and a nominal gain of 7dBi

Tx Gain
The Tx gain of the UHF channel cannot be tested or verified until the rocket antenna system is completed. For the time being, the worst case scenario gain for the UHF Tx Antenna system will be used.

Operating Frequency
The operating frequency of the antenna will be set to 435MHz but may vary between 420MHz and 450MHz. To verify this property, the antenna will be connected to a VNA and swept through a frequency range of 1Hz-1GHz, and the resulting frequency response will be generated. Verification will be considered complete when the frequency response shows a dip between 420-450MHz.

Steps
1. The receiving antenna will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's frequency response will be noted
4. The test will be considered passed if the frequency response shows a significant extreme at the intended operating frequency between 420-450MHz

4.5.6 References and File Links

M. Raman, "VSWR calculator (voltage standing wave ratio calculator)," *VSWR Calculator - Find the Voltage Standing Wave Ratio*, 14-May-2021. [Online]. Available: https://www.omnicalculator.com/physics/vswr-voltage-standing-wave-ratio. [Accessed: 06-Mar-2022].

"RF Link Budget calculator," *Go to .*, 08-Nov-2020. [Online]. Available: https://afar.net/rf-link-budget-calculator/. [Accessed: 06-Mar-2022].

"RF Calculators," *ImmersionRC*. [Online]. Available: https://www.immersionrc.com/rf-calculators/. [Accessed: 06-Mar-2022].

"SINGLE CHIP TRANSCEIVER FOR WALKIE TALKIE - RDA1846." RDA Microelectronics, 02-Dec-2009.

K. B. R. F. Cafe, "U.S. radio frequency allocations - 30 MHz to 300 GHz," *U.S. Radio Frequency Allocations from 30 MHz to 300 GHz - RF Cafe*. [Online]. Available: https://www.rfcafe.com/references/electrical/rf-freq-allo-us.htm. [Accessed: 06-Mar-2022].

Richard, C. Astiz, Owen, Gary, K. Kruger, and Bill, "Quagi antenna: Everything you need to know," *Amateur Radio Wiki*, 02-Nov-2021. [Online]. Available: https://www.amateur-radio-wiki.net/quagi/. [Accessed: 06-Mar-2022].

4.5.7 Revision Table

Author	Description	Date
Kevin Kott	Composed the initial block 1 validation document. Added sections 4.1.1 – 4.1.6	1/8/2022
Kevin Kott	Updated draft for block 1 validation. Included section 4.1.7 revisions table. Rewrote section 4.1.5 to include individual steps. Updated 4.1.2 to include black box diagram and list antenna ports	1/21/2022
Kevin Kott	Removed protocol interface from document as the antennas are passive and will handle any protocol that is input into them. Also, added impedance deviation to interface properties	3/6/2022
Kevin Kott	Corrected grammatical and arithmetic errors within the section. Recalculated values and edited where necessary	5/6/2022

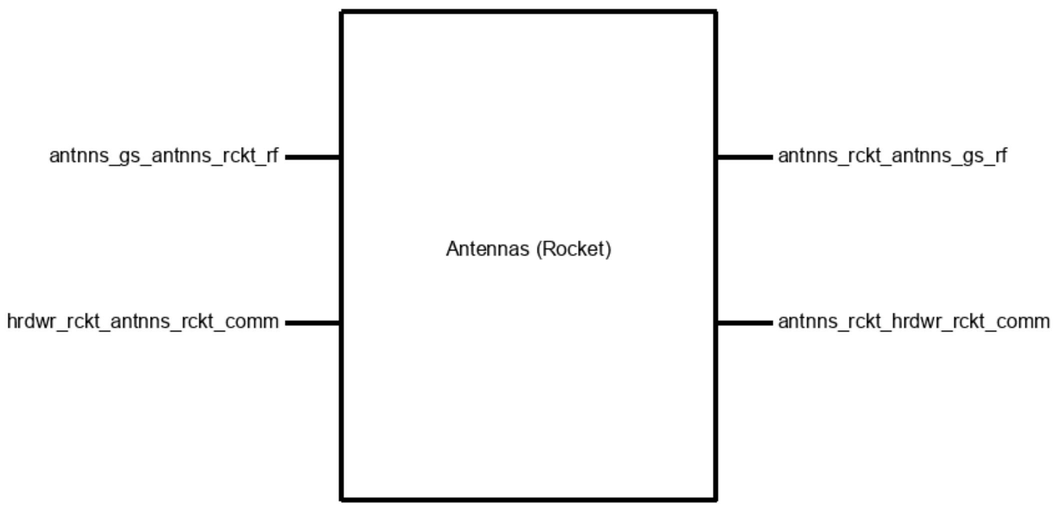
4.6 Rocket Antenna System

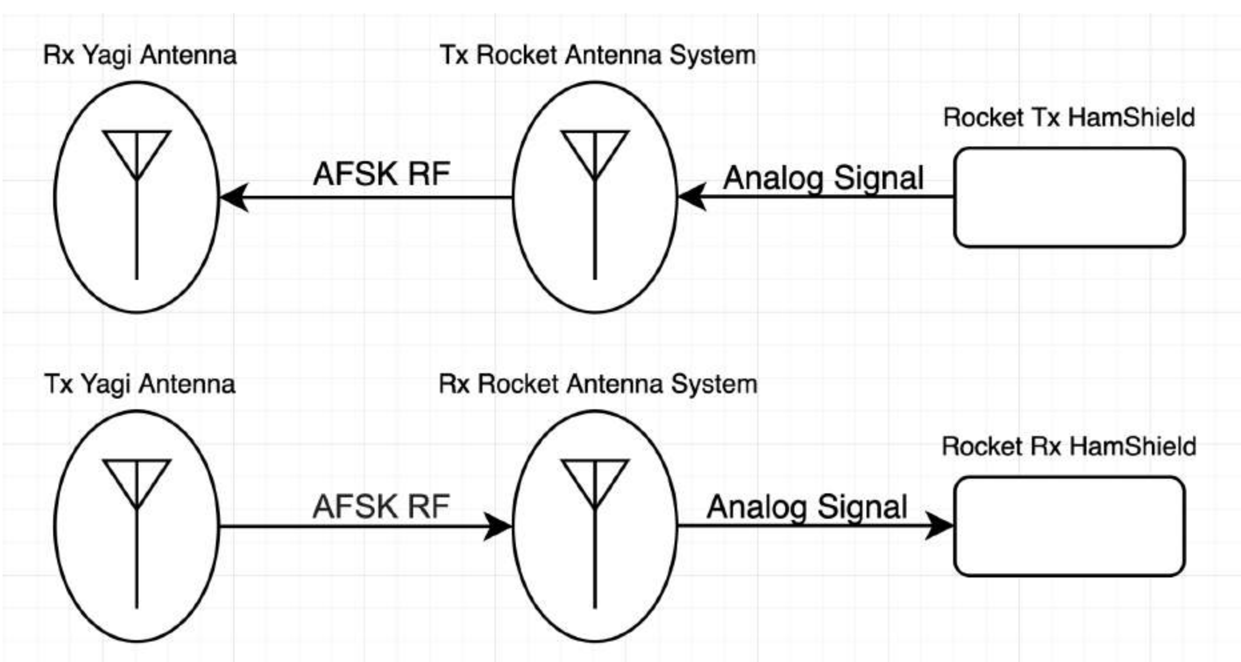
4.6.1 Block Overview

This block entails the design and manufacture of a rubber ducky antenna system for the in-flight vehicle. The antenna system will consist of two pairs of single-band rubber ducky antennas, one pair for Ultra High Frequency (UHF), and one pair for Very High Frequency (VHF). Each antenna pair will be connected to a single pole double throw (SPDT) switch, which itself is connected to its own individual HamShield transceiver. The VHF antenna pair will be used for dedicated uplink communication to the rocket and the UHF antenna pair will be used for dedicated downlink communication.

Each antenna will be composed of copper magnet wire and shielded with heat shrink tubing. Both antenna pairs will be limited to a total overall size of 8 inches and contain a helical section providing electrical length to the antenna while shortening the overall length. The antenna pairs will be connected to their own SPDT switches, which will itself be connected to its own HamShield, with all connections using a 50Ω coaxial cable transmission line. Completion of the block will result in a fully functional, dual-band rocket-mounted communication system. Because of the nature of rubber ducky antenna design, the antennas will not have physical characteristics pre-designed since the antennas will be electrically lengthened to ¼ wavelength, therefore the antennas will be built using a construct and tune technique. This block will be designed and built by Kevin Kott.

4.6.2 Block Design





4.6.3 Block General Validation

The communications system will be operating over a large distance meaning antenna gain becomes one of the most important design characteristics. However, because of the nature of space constraints within a flight vehicle, the rocket-mounted antenna system will be designed to minimize the overall size profile. Based on gathered information from conducted research, the rubber ducky antenna provides the greatest performance per size ratio out of all the other antenna types, at the expense of gain. Since this block pertains to the rocket antenna system, and since gain was prioritized for the ground station antenna system, it is our team's opinion that it is acceptable to focus on size over gain for this antenna system. While there is not a set series of equations that can model a rubber ducky antenna to be able to design prior to construction, there are some general rules when constructing such antennas. First, it has been recorded that the diameter of the coil of a rubber ducky antenna is related to the operating bandwidth of the antenna. Since each antenna pair will be operating within a narrow range of frequencies, according to which band it will be operating in, the diameter of the antenna's coil can be set to optimize performance over bandwidth. The second general rule of rubber ducky antennas states that if the antenna is resonant, that is the antenna is operating within the frequency range it has been tuned for, the larger the diameter of the coil the more the antennas impedance tends towards 0Ω. Conversely the smaller the diameter of the coil the more the impedance tends towards 70Ω. To make sure that the antennas are operating as close to 50Ω as possible, multiple rubber ducky models will be constructed with varying physical dimensions and characteristics in order to achieve the best-operating properties. Each antenna model will be tested using a vector network analyzer (VNA) and tuned accordingly, with the best-performing antenna being used as the final implemented model. The receiving antennas will be tuned to operate in the VHF band centered at 146MHZ and the transmitting antennas will be tuned to operate in the UHF band centered at 435MHZ.

4.6.4 Block Interface Validation

Hrdwr_Rckt_Antnns_Rckt_Comm

Pmin: 356.98mW	The minimum power was chosen based on the maximum standing wave ratio which was selected to achieve a max range of 3% greater than the distance to the Karman line	**HamShield 1.0 Specifications** Power Output: 0.5W
Pnom: >445mW	The nominal power was chosen based on the HamShield's listed power output and a standing wave ratio of 2	**HamShield 1.0 Specifications** Power Output: 0.5W

Znom: 50Ω	The nominal impedance was chosen based on the HamShield's output impedance	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
Zdev: +/-50%	The impedance deviation was chosen based on the resulting standing wave ratio. Antennas with an impedance deviation of greater than 50% are poor performers	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
VSWRmax: 3.3	The maximum standing wave ratio was chosen based on a worst-case scenario for Tx and Rx gain, and a reserved link budget of 25dB which results in a max range of 3% greater than the distance to the Karman line	**SWR Calculator** VSWR: 3.3 Reflected Power: 28.61% Mismatch Loss: 1.4636dB **RF Calculator** Tx Frequency: 435MHz Adj. Tx Power: 356.98mW Min Tx Gain: 1dBi Tx Cable Loss: 0.5dB Min Rx Gain: 1dBi Rx Cable Loss: 0.5dB Link Budget: 25dB Free Space Path Loss: 125.5dB Max Range: 103.0km

Antnns_GS_Antnns_Rckt_RF
(Rubber Ducky Rx Antenna)

Tx Gmin: 1dBi	The minimum Tx antenna gain was chosen based on minimum gain for Rx antenna, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 363.2km
Tx Gnom: >=5dBi	The nominal Tx antenna gain was chosen based on typical gain for selected antenna type	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Nom Tx Gain: 5dBi Tx Cable loss: 0.5dB Nom Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 131.0dB Max Range: 575.6km

Rx Gmin: 1dBi	The minimum Rx gain was chosen based on minimum gain for Tx antenna, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 363.2km
Rx Gnom: >=1dBi	The nominal Rx antenna gain was chosen based on typical gain for selected antenna type	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 146MHz Tx Power: 500mW Nom Tx Gain: 5dBi Tx Cable loss: 0.5dB Nom Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 131.0dB Max Range: 575.6km

Frequency: 144-148 MHz	This frequency was chosen based on the available operating frequencies of the HamShield and the limited frequency ranges allowed for amateur Ham radio Licensees	**HamShield 1.0 Specifications** Amateur Radio Bands: 2m, 1.25m, 70cm Frequency RX/TX: 134-174MHz, 200-260MHz, 400-520MHz **US Frequency Allocation** Band: Very High Frequency Source: Amateur Frequencies (MHz): 50-54, 144-148, 216-220, 222-225

Antnns_Rckt_Hrdwr_Rckt_Comm

Znom: 50Ω	The nominal impedance was chosen based on the HamShield's output impedance	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
Zdev: +/-50%	The impedance deviation was chosen based on the resulting standing wave ratio. Antennas with an impedance deviation of greater than 50% are poor performers	**HamShield 1.0 Specifications** Antenna Connector: SMA, 50Ω
VSWRmax: 10	The maximum standing wave ratio was chosen based on a worst-case scenario for Tx and Rx gain, and a reserved link budget of 25dB which results in a max range of 30% greater than the distance to the Karman line	**SWR Calculator** VSWR: 10 Reflected Power: 66.94% Mismatch Loss: 4.807dB **RF Calculator** Tx Frequency: 146MHz Adj. Transmitted Power: 165.3mW Min Tx Gain: 1dBi Tx Cable Loss: 0.5dB Min Rx Gain: 1dBi Rx Cable Loss: 0.5dB Link Budget: 25dB

		Free Space Path Loss: 122.2dB Max Range: 208.8km

Antnns_Rckt_Antnns_GS_RF
(Rubber Ducky Tx antenna)

Tx Gmin: 1dBi	The minimum Tx antenna gain was chosen based on minimum gain for Rx antenna, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 121.9km
Tx Gnom: >=1dBi	The nominal Tx antenna gain was chosen based on a desire to achieve greater than the minimum gain possible	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Nom Tx Gain: 1dBi Tx Cable loss: 0.5dB Nom Rx Gain: 7dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 133.0dB Max Range: 243.2km

Rx Gmin: 1dBi	The minimum Rx antenna gain was chosen based on minimum gain for Tx antenna, estimated path loss, estimated Tx and Rx loss, and a greater than average miscellaneous loss, all using an RF calculator	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Min Tx Gain: 1dBi Tx Cable loss: 0.5dB Min Rx Gain: 1dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 127.0dB Max Range: 121.9km
Rx Gnom: >=7dBi	The nominal Rx antenna gain was chosen based on typical gain for antenna type	**HamShield 1.0 Specifications** Receiver: RDA1846 Receiver Sensitivity: -124dB **RF Calculator** Tx/Rx Frequency: 435MHz Tx Power: 500mW Nom Tx Gain: 1dBi Tx Cable loss: 0.5dB Nom Rx Gain: 7dBi Rx loss: 0.5dB Link Budget: 25dB Free Space Loss: 133.0dB Max Range: 243.2km

Frequency: 420-450 MHz	This frequency was chosen based on the available operating frequencies of the HamShield and the limited frequency ranges allowed for amateur Ham radio Licensees	**HamShield 1.0 Specifications** Amateur Radio Bands: 2m, 1.25m, 70cm Frequency RX/TX: 134-174MHz, 200-260MHz, 400-520MHz **US Frequency Allocation** Band: Ultra High Frequency Source: Amateur Frequency (MHz): 420-450, 902-928, 1240-1300, 2300-2310, 2390-2450

4.6.5 Block Testing Process

Hrdwr_Rckt_Antnns_Rckt_Comm
For this interface, there are four properties.

Power
To verify the transmitted power, the antenna will be hooked up to a vector network analyzer and the standing wave ratio will be documented. The standing wave ratio will determine how much power is being transmitted through the transmission line to the antenna. The target power transmission of 445mW or greater will be considered nominal, with a minimum power transfer of 356.98mW considered a passed test. The power transmitted will be calculated using the antenna's standing wave ratio and a theoretical input power of 500mW.

<u>Steps</u>

1. The transmitting antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antennas via an S11 test
3. Test results will be reviewed and antenna standing wave ratio will be recorded
4. The antenna's standing wave ratio will be inputted into the VSWR calculator listed in the references section
5. A nominal standing wave ratio of 2 correlates to a transmitted power of 445mW
6. The test will be considered passed if the standing wave ratio is less than 3.3, correlating to a transmitted power of 356.98mW

Impedance and deviation

The nominal impedance for the antenna is 50Ω and will be verified using a vector network analyzer. Verification will be considered complete when the vector analyzer shows an impedance deviating less than 50% of the target impedance.

Steps

1. The transmitting antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's impedance will be recorded
4. The test will be considered passed if the impedance does not deviate more than 50%

Standing Wave Ratio

The max standing wave ratio (SWR) for the antenna is 3.3 and will be verified using a vector network analyzer. The antenna will be hooked up to the testing port of the analyzer, a test signal will be sent through the antenna from and to the analyzer, and the SWR will be calculated with the analyzer. Verification will be confirmed if the SWR is less than 3.3.

Steps

1. The transmitting antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed and antenna standing wave ratio will be recorded
4. The test will be considered passed if the standing wave ratio is less than 3.3

Antnns_GS_Antnns_Rckt_RF

For this interface, there are three properties.

Rx Gain

The minimum gain of the Rx antennas must be no less than 1 dBi. The gain of the Rx antennas will be verified using a vector network analyzer (VNA). Reference antennas will be used to determine the return loss of a known antenna's gain. The receiving antennas will then be connected to the VNA and the return loss at the operating frequency recorded. The difference between the known antennas return loss and the tested antennas return loss will be known as the reference gain. The antennas known gain plus the reference gain will be the tested antennas gain and be no less than 1dBi. The nominal gain of the antenna will be greater than 1dBi. The gain will be considered nominal if no less than 1dBi.

Steps

1. Two antennas with operating frequencies between 136MHz and 176MHz will be connected to the VNA at ports S11 and S21

2. The antennas will be placed a certain distance away from each other and the return loss will be recorded from the S21 port
3. The antenna under test (AUT) will then be connected to the Vector Network Analyzer at the S21 port, replacing the known antenna, using a coaxial cable
4. The gain of the antenna under test will then be recorded
5. The AUT's testing gain will then be subtracted from the known antenna's testing gain giving us a reference gain
6. The reference gain will be added to the known antennas known gain giving us the dBi of the AUT
7. The test will be considered passed if the gain is greater than 1dBi

Tx Gain

The Tx gain of this interface was tested and verified in the previous block. The VHF Tx Antenna gain is greater than or equal to 5dBi and will be implemented into the calculations of this block.

Operating Frequency

The operating frequency of the antenna will be set to 146MHz but may vary between 144MHz and 148MHz. To verify this property, the antenna will be connected to a VNA and swept through a frequency range of 1Hz-1GHz, and the resulting frequency response will be generated. Verification will be considered complete when the frequency response shows a dip between 144-148MHz.

<u>Steps</u>

1. The receiving antennas will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's frequency response will be noted
4. The test will be considered passed if the frequency response shows a significant extreme at the intended operating frequency between 144-148MHz
5. The test will be conducted again for the second receiving antenna
6. The test will be considered passed if the frequency response shows a significant extreme at the intended operating frequency between 144-148MHz

Antnns_Rckt_Hrdwr_Rckt_Comm

For this interface, there are three properties.

Impedance and deviation

The nominal impedance for the antenna is 50Ω and will be verified using a vector network analyzer. Verification will be considered complete when the vector analyzer shows an impedance deviating less than 50% of the target impedance.

Steps

1. The receiving antenna will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's impedance will be recorded
4. The test will be considered passed if the impedance does not deviate more than 50%

Standing Wave Ratio

The max standing wave ratio (SWR) for the antenna is 10 and will be verified using a vector network analyzer. The antenna will be hooked up to the testing port of the analyzer, a test signal will be sent through the antenna from and to the analyzer, and the SWR will be calculated with the analyzer. Verification will be confirmed if the SWR is less than 10.

Steps

1. The receiving antenna will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed and antenna standing wave ratio will be recorded
4. The test will be considered passed if the standing wave ratio is less than 10

Antnns_Rckt_Antnns_GS_RF

For this interface, there are three properties.

Tx Gain

The minimum gain of the Tx antennas must be no less than 1 dBi. The gain of the Rx antennas will be verified using a vector network analyzer (VNA). Reference antennas will be used to determine the return loss of a known antenna's gain. The transmitting antennas will then be connected to the VNA and the return loss at the operating frequency recorded. The difference between the known antennas return loss and the tested antennas return loss will be known as the reference gain. The antennas known gain plus the reference gain will be the tested antennas gain and be no less than 1dBi. The nominal gain of the antenna will be greater than 1dBi. The gain will be considered nominal if greater than 1dBi.

Steps

1. Two antennas with operating frequencies between 400MHz and 470MHz will be connected to the VNA at ports S11 and S21
2. The antennas will be placed a certain distance away from each other and the return loss will be recorded from the S21 port
3. The antenna under test (AUT) will then be connected to the Vector Network Analyzer at the S21 port, replacing the known antenna, using a coaxial cable
4. The gain of the antenna under test will then be recorded

5. The AUT's testing gain will then be subtracted from the known antenna's testing gain giving us a reference gain
6. The reference gain will be added to the known antennas known gain giving us the dBi of the AUT
7. The test will be considered passed if the gain is no less than 1dBi

Rx Gain

The Rx gain of this interface was tested and verified in the previous block. The UHF Rx Antenna gain is greater than or equal to 7dBi and will be implemented into the calculations of this block.

Operating Frequency

The operating frequency of the antenna will be set to 435MHz but may vary between 420MHz and 450MHz. To verify this property, the antenna will be connected to a VNA and swept through a frequency range of 1Hz-1GHz, and the resulting frequency response will be generated. Verification will be considered complete when the frequency response shows a dip between 420-450MHz.

Steps

1. The receiving antenna will be connected to the Vector Network Analyzer using the systems coaxial cable
2. The Vector Network Analyzer will run an analysis on the antenna via an S11 test
3. Test results will be reviewed, and the antenna's frequency response will be noted
4. The test will be considered passed if the frequency response shows a significant extreme at the intended operating frequency between 420-450MHz

4.6.6 References and File Links

M. Raman, "VSWR calculator (voltage standing wave ratio calculator)," *VSWR Calculator - Find the Voltage Standing Wave Ratio*, 14-May-2021. [Online]. Available: https://www.omnicalculator.com/physics/vswr-voltage-standing-wave-ratio. [Accessed: 06-Mar-2022].

"RF Link Budget calculator," *Go to .*, 08-Nov-2020. [Online]. Available: https://afar.net/rf-link-budget-calculator/. [Accessed: 06-Mar-2022].

"RF Calculators," *ImmersionRC*. [Online]. Available: https://www.immersionrc.com/rf-calculators/. [Accessed: 06-Mar-2022].

"SINGLE CHIP TRANSCEIVER FOR WALKIE TALKIE - RDA1846." RDA Microelectronics, 02-Dec-2009.

K. B. R. F. Cafe, "U.S. radio frequency allocations - 30 MHz to 300 GHz," *U.S. Radio Frequency Allocations from 30 MHz to 300 GHz - RF Cafe*. [Online]. Available: https://www.rfcafe.com/references/electrical/rf-freq-allo-us.htm. [Accessed: 06-Mar-2022].

Richard, C. Astiz, Owen, Gary, K. Kruger, and Bill, "Quagi antenna: Everything you need to know," *Amateur Radio Wiki*, 02-Nov-2021. [Online]. Available: https://www.amateur-radio-wiki.net/quagi/. [Accessed: 06-Mar-2022].

4.6.7 Revision Table

Author	Description	Date
Kevin Kott	Composed the initial block 2 validation document. Added sections 4.1.1 – 4.1.7	2/4/2022
Kevin Kott	Minor grammatical edits. Removed all reference to communication protocol as it is not relevant to antenna system	2/18/2022
Kevin Kott	A impedance deviation to interface properties	3/6/2022
Kevin Kott	Corrected grammatical and arithmetic errors within the section. Recalculated values and edited where necessary	5/6/2022

4.7.Power Supply (Rocket) Block Validation

4.7.1. Description

This block is responsible for powering the subsystem within the rocket while in launch. This power is required for the subsystem to be able to transmit desired information back to the ground station. The primary function of the block is to take input voltage from batteries and step voltage down to 9.5 volts to power the Arduino Uno in the subsystem. This block's Champion is Jesse Fretz.

4.7.2. Design

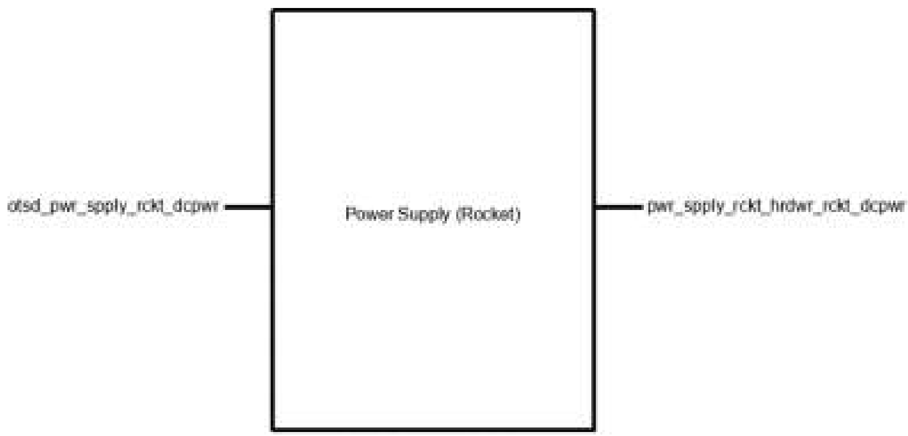


Figure 1: Power Supply (Rocket) Black Box

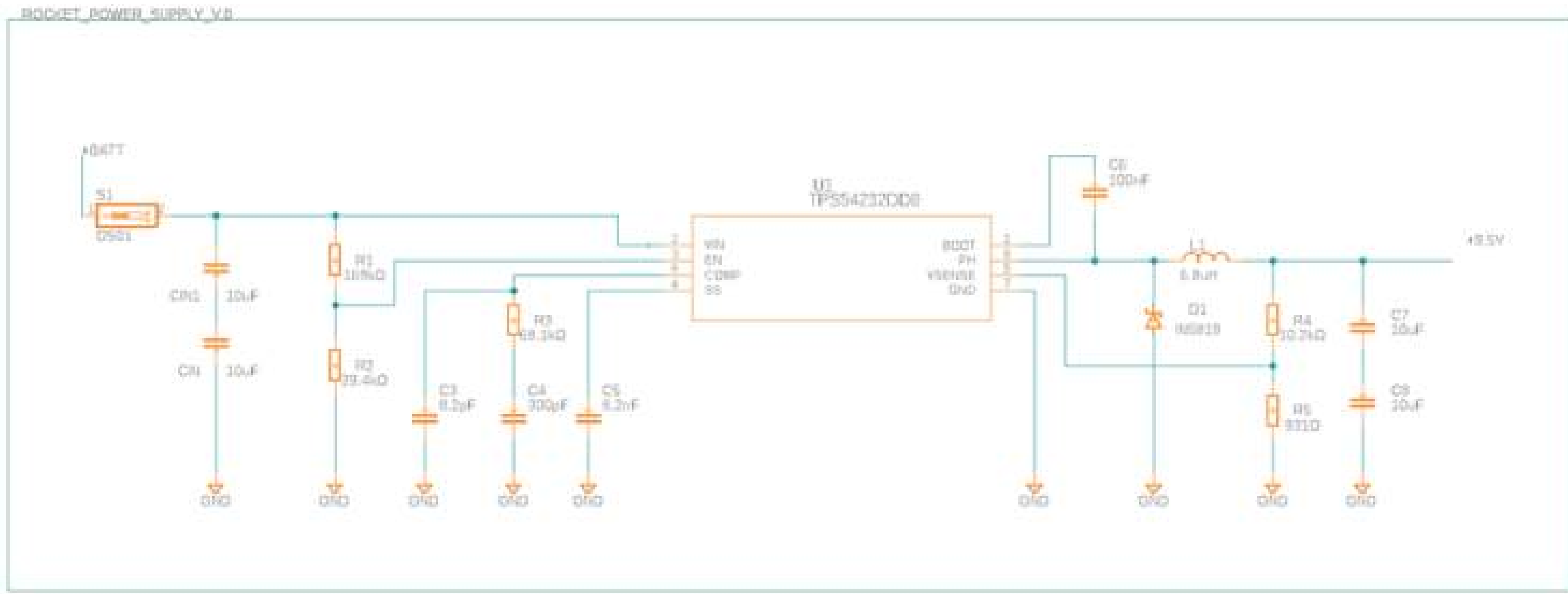


Figure 2: Power Supply (Rocket) Schematic

Notes:

+Batt is the input for this block, otsd_pwr_spply_rckt_dcpwr

+9.5V is the block output, pwr_spply_rckt_hrdwr_rckt_dcpw

4.7.3. General Validation

This block is based on a typical application schematic from the TPS54232 datasheet, with a few changes in order to get the desired voltage output. The chip is a standard voltage step down chip and is rated to take some voltage between 3.5V-28V, and can supply a continuous output of 2A [5]. The schematic seen above is an adaptation of a design given by Texas Instruments WEBENCH when giving them my desired input and output voltages [2]

4.7.4. Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?

otsd_pwr_spply_rckt_dcpwr : Input

Inominal: 5200mA	This was derived by looking at the 2 batteries which will be put in series with one another.	While the battery is working properly it will output this much current at a given time [3].
Ipeak: 260A	This was derived by looking at the 2 batteries which will be put in series with one another. This current will not be reached.	The current peak amount of the batteries comes from the fact that if every cell in the LiPo battery discharges at once it will output this much [3].
Vmax: 14.8V	This was derived by looking at the 2 batteries which will be put in series with one another.	With the two 7.4V LiPo batteries put in series it will create a maximum voltage of 14.8V. LiPo batteries hold their desired output power well but once they reach the end of their charge they almost immediately fall to 0V [3].
Vmin: 14V	This was derived by looking at the 2 batteries which will be put in series with one another	Likely this voltage won’t be seen as LiPo batteries have a steep fall off so they go from their max voltage to 0 volts once they have low

		power [3].

pwr_spply_rckt_hrdwr_rckt_dcpwr : Output

Inominal: 40mA	This is likely higher than needed, it is based on the minimum amount of current Arduino Unos need to operate at.	According to Darshil Patel of Maker Pro an Arduino Uno can operate at a current as low as 15mA. Since the block will consistently have 2A of power when the batteries are plugged in [4].
Ipeak: 2A	This is based on the voltage needs of the Arduino Uno.	The Arduino Uno will draw as much current as it needs to operate at, and the value of 2A exceeds the minimum it will need to operate at [4].
Vmax: 10.5V	This is based on the voltage needs of the Arduino Uno.	According to the Tech specs table on the Arduino website the Arduino should take a recommended voltage between 7-12V. And this block is expected to output around 9.5V, so this Vmax and Vmin allow for 1V ripple [1].
Vmin: 8.5V	This is based on the voltage needed by the Arduino Uno	According to the Tech specs table on the Arduino website the Arduino should take a recommended voltage between 7-12V. And this block is expected to output around 9.5V, so this Vmax and Vmin allow for 1V ripple [1].

4.7.5. Verification Plan

1. First, measurements of the LiPo Batteries in series will be taken to ensure that the voltage is 14V - 14.8V. Current will also be measured to show the batteries current is around 5.4A.

2. Next, measurements at the output of the block will be taken. This will show the Vout is between 8.5V and 10.5V. Current will also be measured to show the current is at the lowest 40mA and at the highest 2A.

3. Lastly, to prove the system can power an Arduino Uno, a wire will be attached to the Vout pin of the block and plugged into the Vin pin on the Arduino board. Then the light on the board should be lit to prove it is powered.

4.7.6. References

[1] *Arduino Uno Rev3,* Arduino, 2021. Available: https://store-usa.arduino.cc/products/arduino-uno-rev3

[2] *Customize TPS54232DR - 14V-14.8V to 9.50V @ 2A,* WEBENCH POWER DESIGNER, Jan, 2022. Available: https://webench.ti.com/power-designer/switching-regulator/customize/1?noparams=0

[3] *Ovonic 2S 7.4V 5200mAh 50C Lipo Battery,* Battery Equivalents. Available: https://www.batteryequivalents.com/lithium-polymer-batteries/ovonic-2s-7-4v-5200mah-50c-lipo battery.html#:~:text=As%20its%20label%20suggests%2C%20Ovonic,store%20~38%20Wh%20of%20energy.

[4] Patel, Darshil, *How to Reduce Arduino Power Consumption,* Maker Pro, Nov, 2019. Available: https://maker.pro/arduino/tutorial/how-to-reduce-arduino-power-consumption#:~:text=Arduino%20Uno%20and%20Pro%20Mini,certain%20situations%20quickly%20adds%20up.

[5] *TPS54232 2-A, 28-V, 1-MHz, Step-Down DC-DC Converter With Eco-Mode™,* Texas Instruments, Nov, 2014. Available: https://www.ti.com/lit/ds/symlink/tps54232.pdf

4.7.7. Revision Table

Date	Revisions Made
1/08/2022	Jesse Fretz: Power Supply (Rocket) initial draft created
1/21/2022	Jesse Fretz: Document was completed, changes were made to the interfaces, as noted by Donald Heer. Schematic was also completed which was also noted by him.

4.8 Power Supply (Ground Station) Block Validation

4.8.1. Description

This block is responsible for powering the subsystem in the ground station while the rocket is launched. This power is required for the subsystem to be able to transmit desired information to and from the rocket. The primary function of the block is to take input voltage from batteries and step voltage down to 9 volts to power the Arduino Uno in the subsystem. This block's Champion is Jesse Fretz.

4.8.2. Design

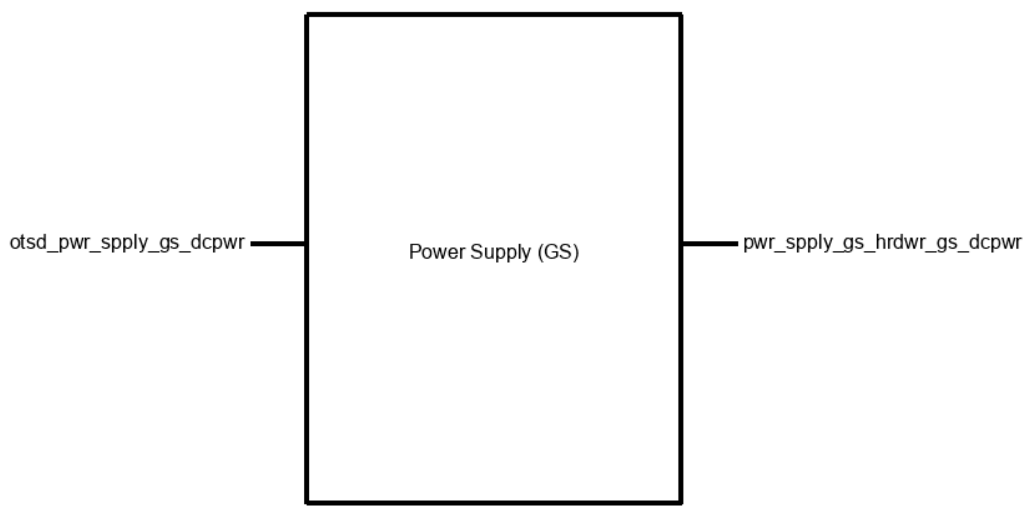


Figure 1: Power Supply (GS) Black Box

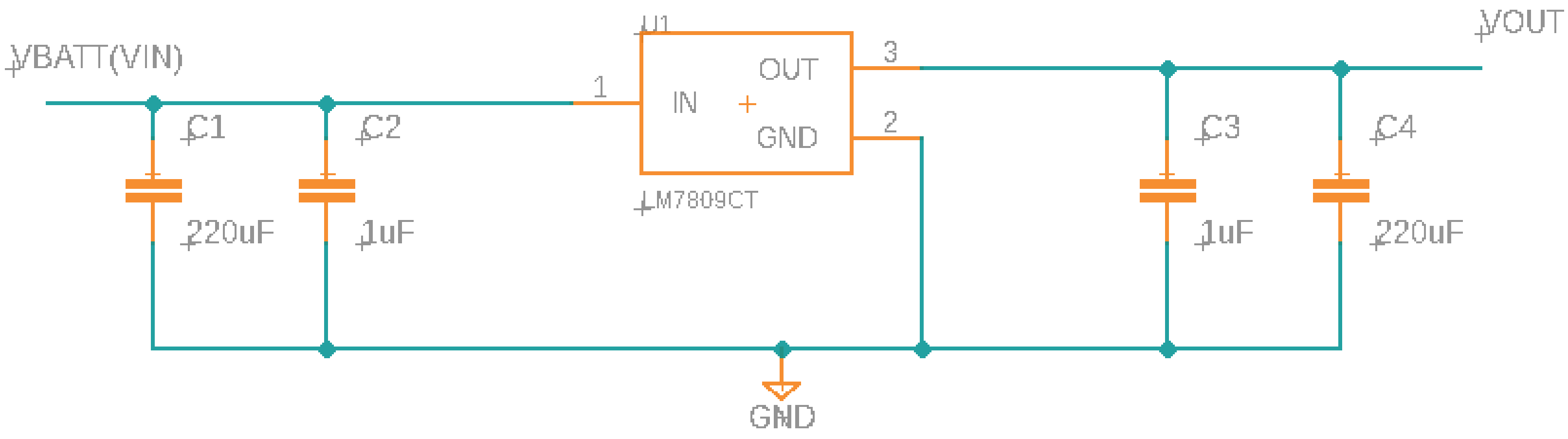


Figure 2: Power Supply (Ground Station) Schematic

Notes:

+VBatt(VIN) is the input for this block, otsd_pwr_spply_gs_dcpwr

+Vout is the block output about 9V, pwr_spply_gs_hrdwr_gs_dcp

4.8.3. General Validation

This block is based on a typical application schematic of a 9V regulator, specifically the LM7809CT. The regulator can take an input of a lot of voltage and step it down to 9 volts as long as the voltage input is about 2 volts greater than 9V. The block will see an input of 14.8V and after the power flows through the block it will output around 9V.

4.8.4. Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?

otsd_pwr_spply_gs_dcpwr : Input

Inominal: 5000mA	This was derived by looking at the 2 batteries which will be put in series with one another.	While the battery is working properly it will output this much current at a given time
Ipeak: 5.2A	This was derived by looking at the 2 batteries which will be put in series with one another. This current will not be reached.	The current peak amount of the batteries comes from the fact that if every cell in the LiPo battery discharges at once it will output this much
Vmax: 14.8V	This was derived by looking at the 2 batteries which will be put in series with one another.	With the two 7.4V LiPo batteries put in series it will create a maximum voltage of 14.8V. LiPo batteries hold their desired output power well but once they reach the end of their charge they almost immediately fall to 0V [3].
Vmin: 10V	This was derived by looking at the 2 batteries which will be put in series with one another.	Likely this voltage won't be seen as LiPo batteries have a steep fall off so they go from

		their max voltage to 0 volts once they have low power [3].

pwr_spply_gs_hrdwr_gs_dcpwr : Output

Inominal: 1A	This is based on the voltage needs of the Arduino Uno.	The Arduino Uno will draw as much current as it needs to operate at, and the value of 2A exceeds the minimum it will need to operate at [3].
Ipeak: 2A	This is based on the voltage needs of the Arduino Uno.	The Arduino Uno will draw as much current as it needs to operate at, and the value of 1.2A exceeds the minimum it will need to operate at [3].
Vmax: 9.09V	This is based on the voltage needs of the Arduino Uno.	According to the Tech specs table on the Arduino website the Arduino should take a recommended voltage between 7-12V. And this block is expected to output around 9.0V, so this Vmax and Vmin allow for 1V ripple [1].
Vmin: 7.5V	This is based on the voltage needed by the Arduino Uno	According to the Tech specs table on the Arduino website the Arduino should take a recommended voltage between 7-12V. And this block is expected to output around 9.0V, so this Vmax and Vmin allow for 1V ripple [1].

4.8.5. Verification Plan

1. First, measurements of the LiPo Batteries in series will be taken to ensure that the voltage is 14V - 14.8V. Current will also be measured to show the batteries current is around 5.4A.

2. Next, measurements at the output of the block will be taken. This will show the Vout is between 7V and 12V. Current will also be measured to show the current is at the lowest 1A and at the highest 2A.

3. Lastly, to prove the system can power an Arduino Uno, a wire will be attached to the Vout pin of the block and plugged into the Vin pin on the Arduino board. Then the light on the board should be lit to prove it is powered.

4.8.6. References

[1] *Arduino Uno Rev3,* Arduino, 2021. Available: https://store-usa.arduino.cc/products/arduino-uno-rev3

[2] *Ovonic 2S 7.4V 5200mAh 50C Lipo Battery,* Battery Equivalents. Available: https://www.batteryequivalents.com/lithium-polymer-batteries/ovonic-2s-7-4v-5200mah-50c-lipo-battery.html#:~:text=As%20its%20label%20suggests%2C%20Ovonic,store%20~38%20Wh%20of%20energy.

[3] Patel, Darshil, *How to Reduce Arduino Power Consumption,* Maker Pro, Nov, 2019. Available: https://maker.pro/arduino/tutorial/how-to-reduce-arduino-power-consumption#:~:text=Arduino%2

4.8.7. Revision Table

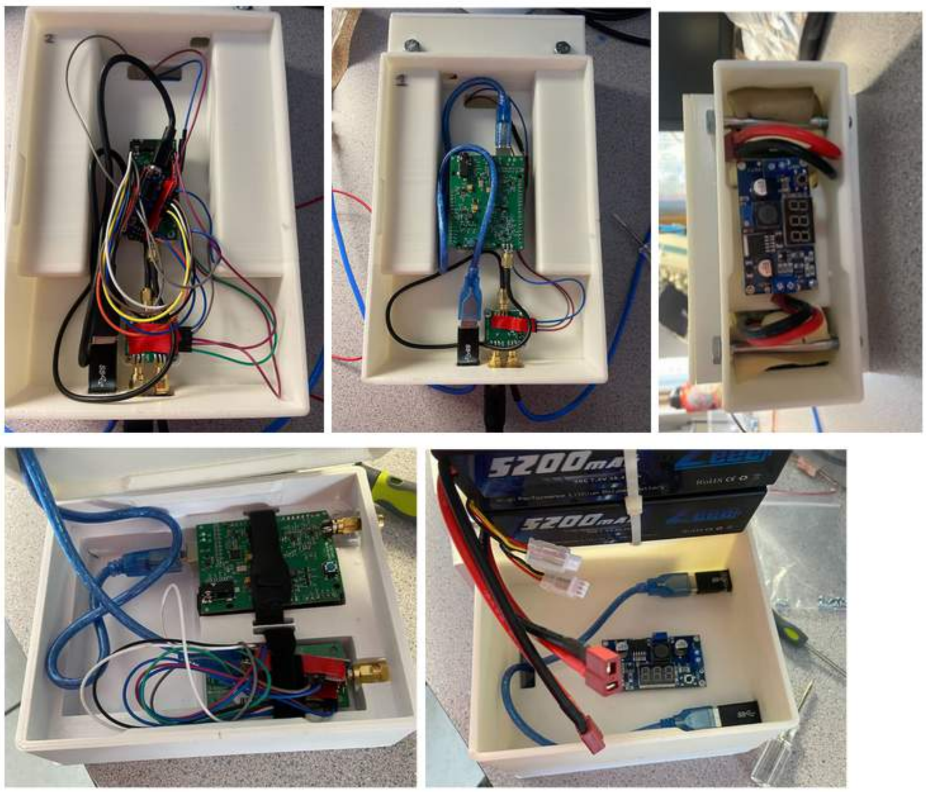
Date	**Revisions Made**
2/04/2022	Jesse Fretz: Power Supply (Ground Station) initial draft created
2/18/2022	Jesse Fretz: Added the schematic and finished final draft of document

Section 5: System Verification Evidence

5.1 - Universal Constraints

5.1.1 The system may not include a breadboard

No breadboard to be found. The power supplies and Hamshield-Arduino pairs use PCBs. The antennas, enclosure and code don’t need a breadboard.



A look into the innards of the system, for both the rocket and the ground station, shows only Arduino, switch and power supply printed circuit boards. No breadboards found.

5.1.2 The final system must contain both of the following: a student designed PCB and a custom Android/PC/Cloud application

Our team received an exemption from the PCB constraint, there was not a good way to fit 50% of our blocks on a PCB as we needed separation for the antennas to not interfere with one another. There was a PCB created for the power supply, though the power supply ended up not meeting the efficiency requirement, so it was decided to go for a more efficient supply over a PCB. There will still be a small PCB used as a connector to a barrel jack for the Arduino Uno.

The data visualizer PC application is a Python based script which takes in the received data and graphs it on the computer screen.

GUI_Test_Python.MOV

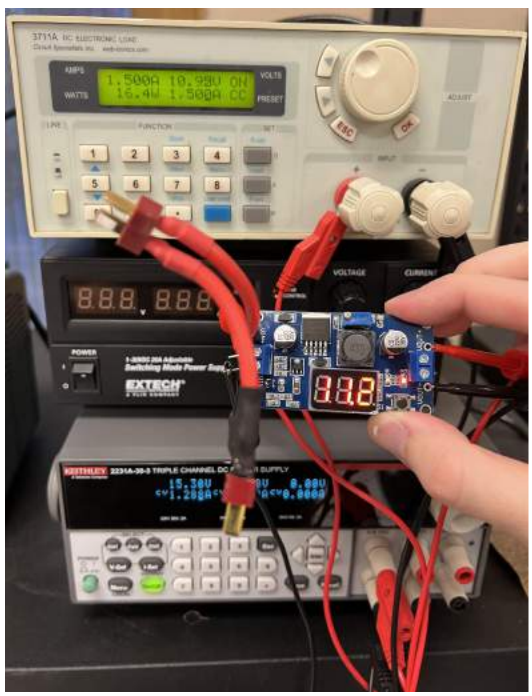
5.1.3 If an enclosure is present, the contents must be ruggedly enclosed/mounted as evaluated by the course instructor

All the hardware in both enclosures are mounted down and secured. See images in 5.1.1.

5.1.4 If present, all wire connections to PCBs and going through an enclosure (entering or leaving) must use connectors

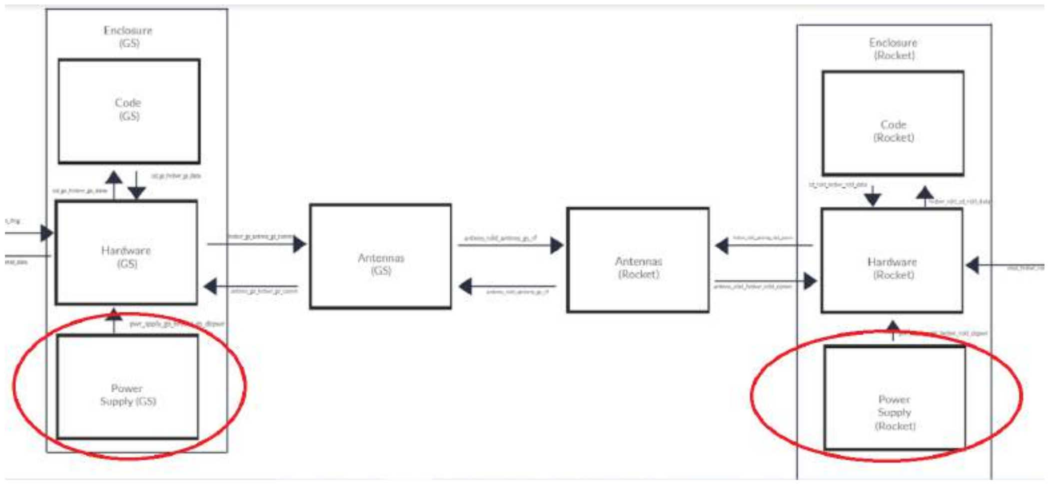
On both enclosures any wires leaving the enclosures have connectors that easily connect and disconnect. See images in 5.1.1.

5.1.5 All power supplies in the system must be at least 65% efficient



With the new power supplies, according to the calculations at maximum load, the power out over the power in is about 84%.Pin = 15.3 * 1.288 = 19.7 W; Pout = 16.4 W; So efficiency is Pout/Pin = 0.8325 or 83.25%.

5.1.6 The system may be no more than 50% built from purchased modules



This is our block diagram shown above. The blocks circled in red are purchased modules.The other six blocks: code, enclosure, antennas, hardware were built by us. Only the power supplies were bought which means only 20% of the system is made up of purchased modules.

5.2 - Requirement 1

5.2.1 The ground station subsystem will send a test message to the rocket subsystem. Confirmation that the message was received on the rocket end will be done visually using arduino IDE serial monitor. Next, the rocket will send a test message to the ground station. Confirmation that the message was received by the ground station subsystem will be done visually, using arduino IDE serial monitor.

5.2.2 Testing Process:

i) The uplink will be verified first. This is communication from the ground station to the rocket. The rocket and ground station Arduinos will be connected to individual computers, and then a string of characters (such as the numbers 1 through 10) will be sent from the ground station to the rocket. If at least 75% of the characters are displayed within the Arduino console on the rocket's computer then the uplink has been verified.
ii) The downlink will be verified second. This is communication from the rocket to the ground station. The same setup used for the first verification method will be used for this verification method. A series of characters (such as the numbers 11 through 20) will be sent from the rocket to the ground station and if the ground station computer displays at least 75% of the same characters within the arduino console then the downlink is verified.
iii) Both sites should be able to transmit and receive messages independently, without altering hardware connections.

5.2.3 Testing Evidence:

Uplink_Downlink_Test.MOV

5.3 - Requirement 2

5.3.1 The rocket subsystem will be capable of transmitting RF signals to a distance of at least 60,000 ft. This will be confirmed by measuring the gain of each antenna and then estimating the effective link range of the system. A conservative link margin of 25dB will be used to account for any unforeseen losses within the system.

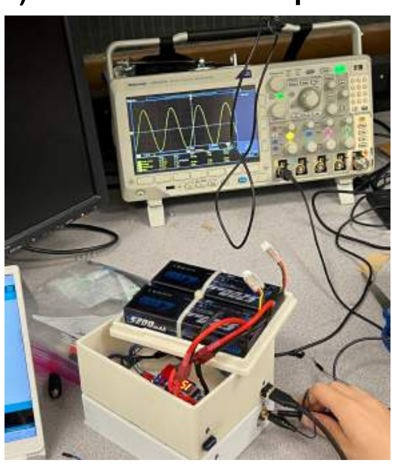
5.3.2 Testing Process:

i) To confirm this requirement, each site's transmitting and receiving antenna will have their gain measured. This is performed by using antennas with known gain as a baseline, taking a measurement from a distance, then replacing one of the known antennas with the antenna-under-test and taking a measurement from the same distance.
ii) Once the gain has been measured, an online RF Link Range calculator will be used to estimate the ideal range of our system. The link to the calculator is: https://www.immersionrc.com/rf-calculators/
iii) An ideal RF link range of at least 1.5x the target distance of 60,000ft (18.288km) will be used to confirm this requirement to account for any non-idealities within the system. 1.5x the target distance is 90,000ft (27.432km).

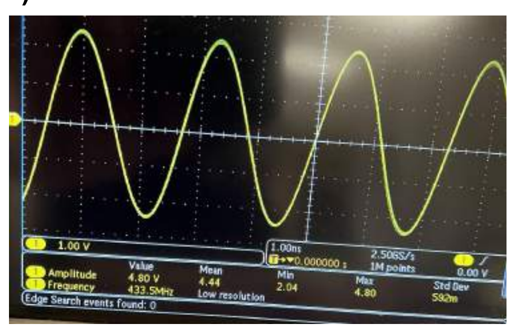
5.3.3 Testing Evidence:

Step 1: Calculate the transmitting power
Below is our system powered on and the hamshield transmitter being measured by an oscilloscope.

i) Measurement setup



ii) Power measurement zoomed in



Calculated Power

Amplitude: 4.80V
Average Voltage: 3.394Vrms
Impedance: 50Ω
Transmitted Power: 0.2304W (230mW)

Step 2a: Gathering baseline measurements with 3dBi antennas at 146MHz

Step 2b: Gathering measurements with VHF Quagi Antenna at 146MHz

Step 2c: Calculating the antenna gain for VHF Quagi antenna from reference antennas

S21 3dBi baseline antenna received signal strength at 146MHz: -29.45dB
S21 AUT Quagi antenna received signal strength at 146MHz: -27.37dB
VHF Quagi antenna gain at 146MHz: 3dBi + [(-27.37dB) - (-29.45dB)] = 5.08 dBi

Step 3a: Gathering baseline measurements with 3dBi Antennas at 146MHz

Step 3b: Gathering measurements with VHF Rubber Ducky antenna at 146MHz

Step 3c: Calculating the antenna gain for VHF Rubber Ducky antenna from reference antennas

S21 3dBi baseline antenna received signal strength at 146MHz: -34.22dB
S21 AUT Rubber Ducky antenna received signal strength at 146MHz: -26.42dB
VHF Rubber Ducky antenna gain at 146MHz: 3dBi + [(-26.42dB) - (-34.22dB)] = 10.8dBi

Step 4a: Gathering baseline measurements with 3dBi antennas at 435MHz

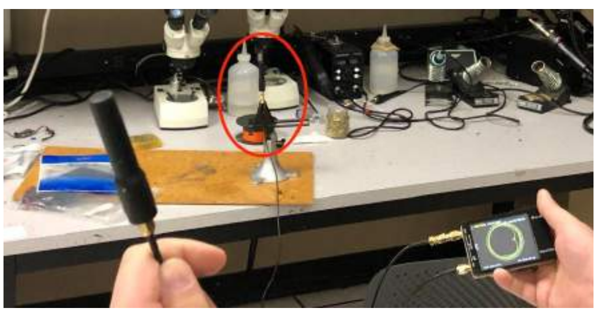
Step 4b: Gathering measurements with UHF Quagi antenna at 435MHz

Step 4c: Calculating the antenna gain for UHF Quagi antenna from reference antennas

S21 3dBi baseline antenna received signal strength at 435MHz: -43.80dB
S21 AUT Quagi antenna received signal strength at 435MHz: -39.39dB
UHF Quagi antenna gain at 435MHz: 3dBi + [(-39.39dB) - (-43.80dB)] = 7.41 dBi

Step 5a: Gathering baseline measurements with 3dBi antennas at 435MHz

Step 5b: Gathering measurements with UHF Rubber Ducky antenna at 435MHz



Step 5c: Calculating the antenna gain for UHF Rubber Ducky antenna from reference antennas

S21 3dBi baseline antenna received signal strength at 435MHz: -50.46dB
S21 AUT Rubber Ducky antenna received signal strength at 435MHz: -39.82dB
UHF Rubber Ducky antenna gain at 435MHz: 3dBi + [(-39.82dB) - (-50.46dB)] = 13.64dBi

Step 6: Calculating RF link Range using calculated antenna gains in previous steps

UHF - Downlink
Receiver Sensitivity: -124dB
RF Calculator Tx Frequency: 435MHz
Tx Power: 230mW, 23.62dB
Tx Gain: 13.64dBi
Rx Gain: 7.41dBi
Link Budget: 25dB
Free Space Loss: 143.67dB
Max Range: 836.59km

VHF - Uplink
Receiver Sensitivity: -124dB
RF Calculator Tx Frequency: 146MHz
Tx Power: 230mW, 23.62dB
Tx Gain: 5.08dBi
Rx Gain: 10.8dBi
Link Budget: 25dB
Free Space Loss: 138.50dB
Max Range: 1374.51km

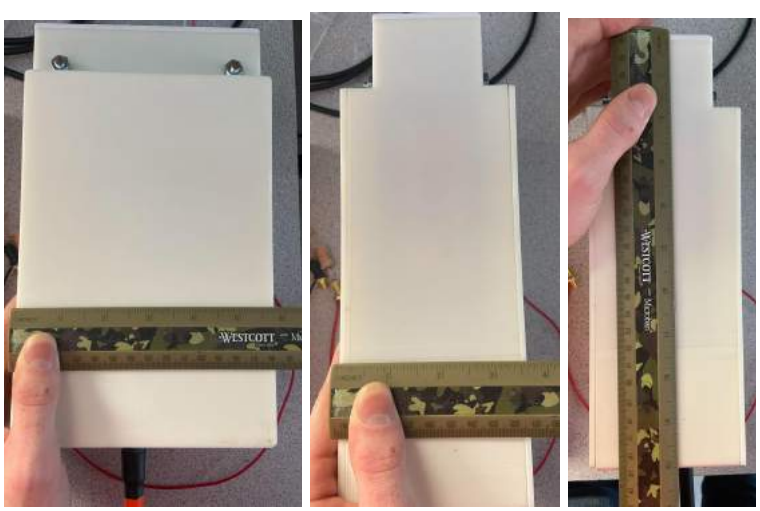
5.4 - Requirement 3

5.4.1 The horizontal measurement of the enclosure (both length and width) must not exceed 6 inches. The vertical measurement is preferred to be as small as possible but a target is 12 inches. (Excluding antennas)

5.4.2 Testing Process:

i) The rocket communication module will be measured with a tape ruler. If the ruler measures the length and width to be less than eight inches. If the ruler measures the height of the enclosure (excluding coax cables) to be less than 12 inches, too, then this property is verified.

5.4.3 Testing Evidence:



From the three pictures above you can see that the rocket enclosure is 5.75 inches long by 3.5 inches wide by 10 inches in height. The width and the length are under 8 inches so it will fit within the rocket body.

5.5 - Requirement 4

5.5.1 The Ground Station will be able to communicate with an outside user interface by means of input from the user and output to the user.

5.5.2 Testing Process:

i) This requirement can be tested on two Arduino serial monitors or an Arduino serial monitor on the transmitting end and a Python application (which was one of the constraints).
ii) One will type in a string of characters (like numbers 1 through 10) on the Arduino serial monitor. Then it will reprint each of the numbers that were entered into the terminal, in order from first to last, as each of them are being transmitted out to the receiving side. This reprinting, which is the communication between the ground station and the Arduino Uno, will act as confirmation that each number was transmitted. This will be verified for both the uplink and the downlink.
iii) The receiving end will print out numbers that were transmitted by the rocket module. It should print them out in the order they were sent. This will confirm that the Ground Station communicates with the Arduino Nano. The acceptable loss of 25% applies to this requirement for both uplink and downlink. That means at least 75% of the characters transmitted need to be received and printed out.
iv) The receiving port of the ground station will also be tested with the Python script. The verification of the communication between the Python application and the ground station will be confirmed using step iii.

5.5.3 Testing Evidence:

GUI_Test_Python.MOV

5.6 - Requirement 5

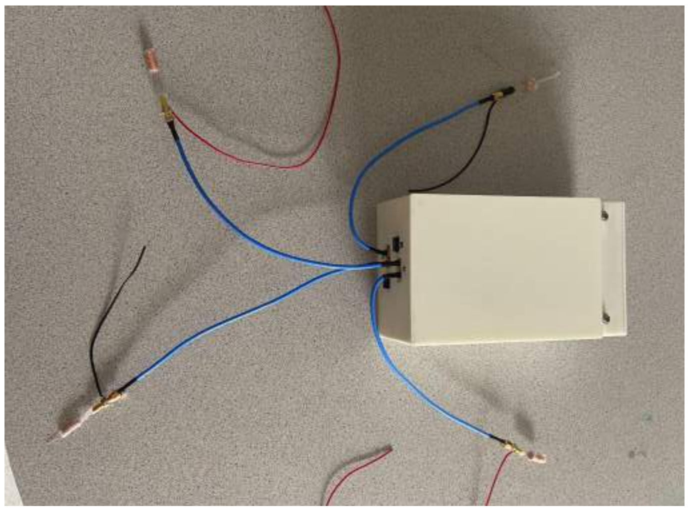
5.6.1 Four antennas on the rocket. Two for receiving and two for sending data. Receiving antennas should be on opposite sides from the rocket. Transmitting antennas should also be opposite from each other.

5.6.2 Testing Process:

i) Visual verification. If there are four antennas, with two of them on the receiving ports and two of them on the sending ports.
ii) If verification method 2.1.1 passes with the four antennas set up, then this shows that the four antennas are working.
iii) The two sets of antennas are controlled by switches. If further proof is needed,

remove the transmitting antennas and hook up one of the sending ports to an oscilloscope reader. Make note of what port it is hooked to. Set the switch pin to HIGH or LOW, whichever one turns on the port. If there is a steady voltage wave on the oscilloscope, then this means that one port is working. Check the other port - it should not be working. Set the switch pin to the opposite of what it was and see if a voltage wave appears. If so, then that means the switch is working and this property is verified.

5.6.3 Testing Evidence:



As you can see there are two receiving antennas and two transmitting antennas and the same antennas are pointed in opposite directions.

5.7 - Requirement 6

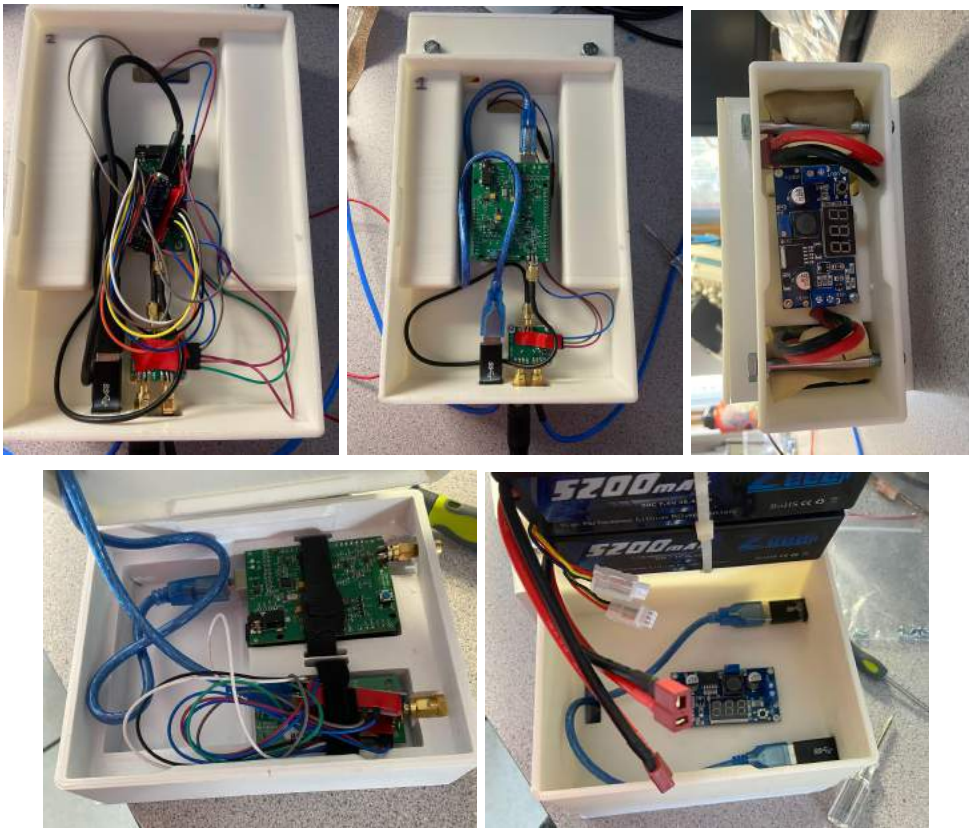
5.7.1 An enclosure will be designed to house all of the electrical components. The components will be mounted neatly within the enclosure with no components moving around.

5.7.2 Testing Process:

i) The enclosure for both the ground station and the rocket module will entirely contain the Arduinos, HamShields, PCB, and Power module.
ii) The enclosure will be inspected after mounting the internal components and checked for loose cables and mounts. All circuit boards will be either mounted with screws, secured with zip ties or velcro straps or taped to the surface of the inside of the enclosure. Other ways of mounting and securing are feasible, too.

iii) After the internal components are mounted, the system's wiring will be disconnected between modules (the wiring in individual blocks won't be disconnected) and reconnected to confirm accessibility.

5.7.3 Testing Evidence:



The pictures above show all the components in our system housed within the enclosure.

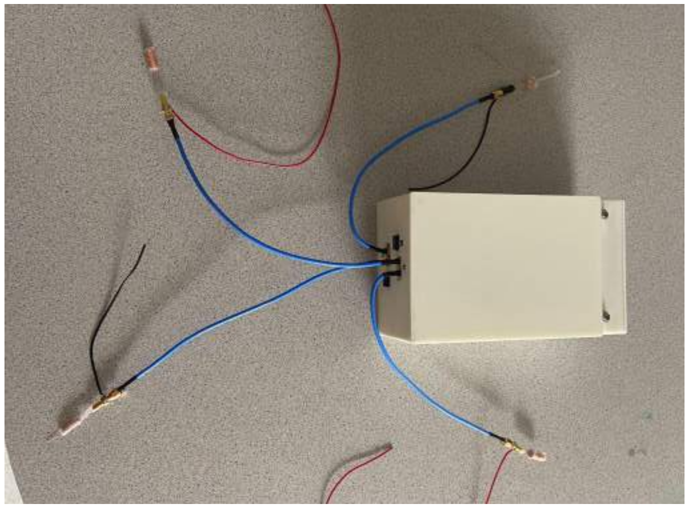
5.8 - Requirement 7

5.8.1 The antenna must be fully contained within the eight inch diameter body of the rocket.

5.8.2 Testing Process:

i) The antennas will be connected with flexible coax cable. Measure with a foot ruler. If the maximum distance between the furthest two points on the coax cable or antennas is less than eight inches (with someone exerting pressure on the coax cables to make them as close as possible) then this property is verified.

5.8.3 Testing Evidence:



The picture above shows the rocket enclosure and the 4 rocket antennas connected to the switches with coax cables; they can be flexed and moved to fit inside the 8 inch rocket body.

5.9 - Requirement 8

5.9.1 The subsystem within the rocket must be less than 10 pounds.

5.9.2 Testing Process:

i) The weight will be measured with a package delivery scale.

5.9.3 Testing Evidence:

The scale shows that the rocket enclosure is 3 lb and 5.6 oz.

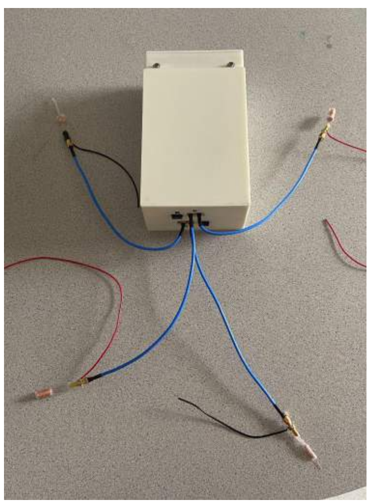
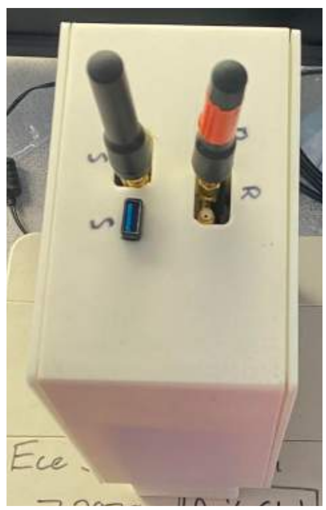
5.10 - Requirement 9 (Insurance requirement)

5.10.1 The communication system must be able to swap out the necessary antennas to communicate from space (100 km) when needed for the rocket next year. Antennas must be detachable and swappable for different ones. Any external components or wiring leaving or entering the module must be detachable.

5.10.2 Testing Process:

i) All antennas, wires (connecting between blocks) and power supplies will be taken out of their connectors and put back in to verify that they are in fact replaceable.
ii) Other components, such as different wires and antennas will replace the current components to see if the module still works.
iii) Redo the previous eight verification methods with these new antennas or wires. If they are all verified then this last requirement will be verified.

5.10.3 Testing Evidence:


From the first two pictures above you can see that you can easily change antennas by unscrewing them. The last picture shows the power supply bay and the power supply is held in by screws so you can easily unscrew it and screw in a new power supply.

5.4 - References and File Links

5.5 - Revision Table

Date	Revisions Made

3/5/2022	Sam Wagner: Added Initial Content
3/6/2022	Camden Robustelli: Added System Requirements
4/22/2022	Jesse Fretz: Updated PCB and Efficiency constraints
4/29/2022	Camden Robustelli: Added pictures, videos and comments in the testing section of the requirements.
5/3/2022	Sam Wagner: Updated verification methods added evidence pictures to the universal constraints section. Kevin Kott: Updated verification requirements, methods, and calculations
5/5/2022	Camden Robustelli: Reformatted and added extra details to all requirements but primarily requirement 2. Fixed indents for clarity and added professor recommendation adjustments.

Section 6: Project Closing

6.1 Future Recommendations

6.1.1 Technical Recommendations

1. Do not use Hamshield code base

   The Hamshield code base for the Arduino microcontroller is not very good. One cannot make slight (legitimate and compilable) adjustments in the code without breaking the whole system. For instance, the volume coming from the jack on the Hamshield was too loud and when the volume was adjusted to be lower, the entire system stopped working for some reason. Volume was not related to transmission or receiving characteristics, so seeing that break the whole system was odd. It seemed that a number of the custom functions in the code base were broken, such as the setRFpower command. It didn't increase power to the necessary specification. The datasheet said the max power output was 500 mW; however, the maximum that we observed was only 230 mW. Increasing the power output broke the system sometimes, as well.

   While testing the system, some non-numerical messages were put through. Most words worked well. However, for reasons we never found out, a few words didn't work out. This was likely another unintended fault with the Hamshield code base.

   Finding information about these functions was very difficult as well. There was only a single GitHub page which only gave a brief description of some of the custom functions[1]. Some important functions, like the threshold functions which controlled the sensitivity of the Hamshield, did not have any documentation whatsoever. Overall, there was very poor documentation for the Hamshield.

Also, Hamshield is now a deprecated piece of hardware. It is no longer sold. The company that was producing them, Inductive Twig, shut down on December 31st, 2021. Likewise, its custom firmware is deprecated, too.

The solution to this issue would be to design custom firmware for this device. The Hamshield board uses easily sourced SMDs that have datasheets online. As a part of the HamShield documentation, there is a GitHub repository with all of the design documents and CAD files[2].

Another solution would be to not use the Hamshield at all, although a new system would need to be designed and built from scratch. Telemegas are one such option. This would be vastly more expensive, however.

2. Rework the wiring for the system.

   In the rocket module the power station is connected to two 7.4V LiPo batteries in series. This gives a voltage of 14.8 volts. The power supply steps that down to 12 volts and powers an Arduino Uno and Nano. Both the uno and the nano were hooked up to the computer via a USB. The USB supplied some power, too. When the battery pack was disconnected, there was a power surge and it killed the computer connected to it. The computer was unable to be turned on for several minutes. The Arduinos didn't seem to be damaged. So for the current design, make sure to disconnect the computer from the system before unhooking the power supply. There is an article that delves into the possibilities of computer damage from arduino operation listed in the sources section[3].

   In order to solve the problem, one might get a surge protector, like an ESD protection diode, and place it between the power supply and the input to the system. Although it is not entirely known what is causing the problem, it is suspected that it has something to do with wiring issues. Perhaps the grounds are not all hooked up together. So redoing the wiring in the rocket module power supply would be good.

   Regarding the HamShield/Arduino interface, we used female and male headers to mount the systems together across all pin points. In the end, this seemed to hinder our efforts to optimize the system. This is because not every pin is used between the two modules, and when we ran out of pins on the main microcontroller, we had to resort to using the secondary microcontroller for minute operations. This is non-ideal as we would have to pass the states of certain subsystems into the main microcontroller from the secondary.

3. Consider other options rather than linear voltage regulators for power supply.

   The original power supply design was a simple 9 volt regulator circuit that would take the 14.8V input and drop it down to 9V and 1A for power. There were several problems with this, to begin with when using regulators it is hard to meet efficiency requirements, as regulators are inefficient to begin with. Another issue with the regulator power supply designed this year is without a heat sink the supply would shut down and stop supplying anything anywhere close to necessary voltage and would only stay on for about 30 seconds. Even after adding a heatsink it still only lasted 2 minutes 30 seconds. This may have had something to do with the amount of power the LiPo batteries were supplying, but it is still unclear exactly why it shut off so quickly. Another consideration for why it may not have worked is the PCB designed for the power supply as the regulator did not have the shut off problem on the breadboard but did once moved to the PCB. I think the best idea for a power supply would be a buck-boost converter that has a duty cycle/frequency “knob” which makes it so you can vary frequency or duty cycle, or what was used in the system which was a nonlinear adjustable voltage regulator.

4. Use a function generator to obtain a more accurate measurement for gain of the Antennas, instead of a NanoVNA.

   The issue with the NanoVNA is that testing is limited to the length of the coax cables plugged into it. This is because the NanoVNA only works as a closed system. After multiple attempts to use just the S21 port with an outside signal generator, the NanoVNA would not measure received RF signals. Instead, a good alternative would be to use a portable function generator that has the ability to operate at both UHF and VHF frequencies and have a tunable power output, and then use either a portable oscilloscope or a digital multimeter to measure the power of the received signal. There may be other alternatives to these suggested devices but the main takeaway is to utilize tools that have the ability to operate together over long distances.

   With one of our requirements being to send data at a distance of at least 11.4 miles, we had to find a way to prove that our system was able to accomplish this. Unfortunately, the tools that we had purchased were not the best at completing this task. In the end we had concluded that the best way to prove our system was capable of communicating over 11.4 miles was to gather baseline measurements and calculate the effective range of the system. Being that an entry level vector network analyzer comes in at several hundred dollars, with professional systems going into the tens of thousands, purchasing a nano vector network analyzer seemed like a good alternative; however, as has been mentioned previously, it did not have the full functionality that we needed. With the nano vector network analyzer we were limited to closed loop signal analysis which requires a coax cable connection and limits testing range.

6.1.2 Global Impact Recommendations

1. Everyone on the team NEEDS to get a HAM license to transmit over the RF bands used by the communications system.

   It is very important to get a HAM license when operating the communications system. This is because it is illegal to transmit over HAM radio frequencies without a license and doing so can result in stiff fines, seizure of equipment, and civil and/or criminal penalties. In addition to this, transmitting without finding an open frequency can cause co-channel interference which could jam the channel. There are many amateur projects that use the frequency ranges our system operates on, such as the Earth-Moon-Earth and amateur satellite communication[4]. Jamming a channel could have consequences ranging from disrupting an amateur call to possibly jamming signals for emergency services.

   Finding time to study for the license can be challenging, as capstone and other classwork tends to be pretty demanding. Also tests are only given once a month which is another difficulty. It would be best to take the test in the fall since the block building doesn't start until winter. It gives you multiple shots at the test just in case you fail the first time. There is a HAM radio club on campus which can be a great resource to use if there are any HAM related questions. Making a plan with all team members would be good. Communicate with the capstone instructor to see what kind of tips and help they could give you.

2. It would be a good idea to develop a protocol to make sure the communication system does not mistake outside interference as a user inputted command which might trigger a malfunction on the rocket and could endanger people on the ground.

   The rocket will be traveling at least 60,000 feet into the air. At these heights, an explosion could send debris raining down that could harm on onlookers below. What must be taken into consideration is that the flight vehicle is composed of many different subsystems, designed by many different teams. If the communications team was responsible for creating an in-flight disaster that harmed somebody, the entire HALE team would be liable.

   As the IEEE code of conduct states, we are "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment [5]".

6.1.3 Teamwork Recommendations

1. Planning out the work you want to accomplish before meeting up can be very helpful and save a lot of time.

   A lot of meetings were counterproductive - trying to solve the same issues over and over again. This could be solved by writing down the methods that were used last time and the solutions that were found. Sometimes a lot of time was wasted trying to figure out what we should test and what we shouldn't. Also, we brought testing tools with us to our respective homes. Communicating beforehand what tests we should do will allow us to know what tools to bring.

2. Assigning tasks that are reasonable to do in a week's worth of time for each teammate.

   Being accountable when the work you split up is due don't let your teammates slide or else it might keep occurring and put your group in a bad spot. Delegate tasks early on. By assigning each member an individual part to work on early, this leaves more time for system integration and helps weed out hidden problems within the system that might not be apparent initially.

3. Make sure to order parts as soon as possible.

   Ordering parts early will make your project a lot easier. It allows you to have extra time to debug your system. No plan is ever perfect so there are going to be problems you stumble upon when building your system. Having that extra time can save you in the long run and allow you to perfect your system. Also choose components that are easy and fast to receive if possible. Amazon is a great source for getting parts the next day and the components are good enough for the task at hand. Using an electrical components company can work too but they usually have delays or slow shipping so take that into consideration.

4. Create a project work time

   A lot is going on for each of your team members from other classes to just life in general. To be successful in this project you need to allocate at least twice a week time to specifically work on the project. Choose times when everyone is free and meet up and work on the project. Getting into a routine of working piece by piece will save you in the long run. Try to avoid cramming at the last minute. The deadlines sneak up fast and a lot of the things you have to do for class are very stressful if you wait until the due date [6]. The product isn't going to be very polished and this project is trying to showcase what you can do when you graduate so give it your best effort. Our group had bi weekly meetings with one of the professors pretty much the whole year and it was a great resource and gave us another opportunity to assess and see where we are at.

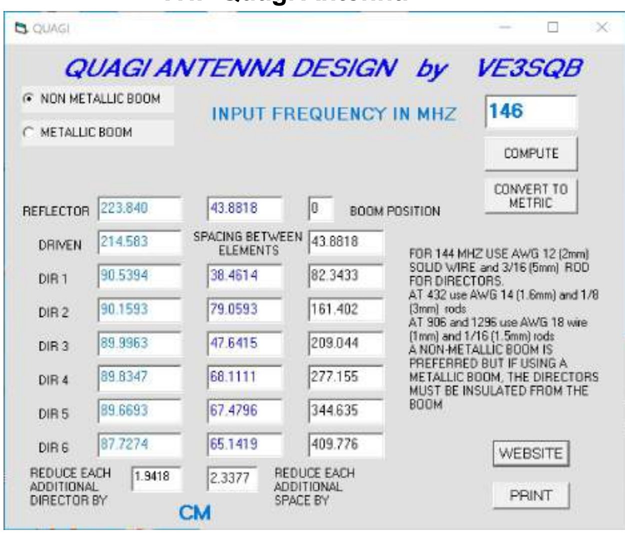
6.2 Project Artifact Summary

6.2.1 Antenna Design Artifacts

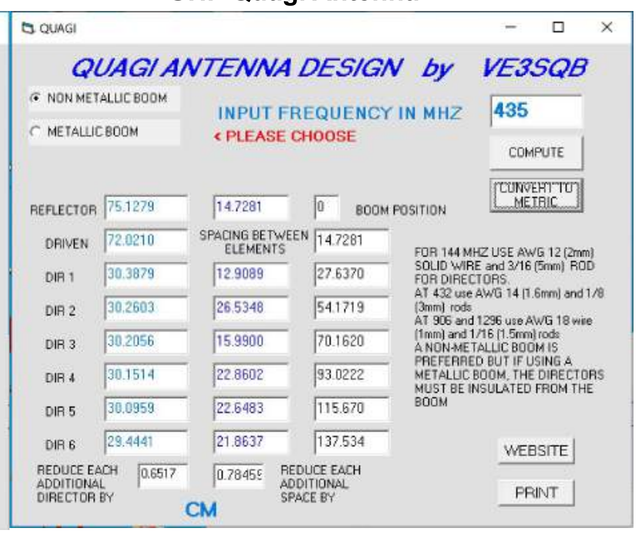
Antenna Documentation

Quagi Antenna design was completed using a Quagi dimensions calculator [7]. Being that there is no standard documentation nor calculations for designing a quagi type antenna, a program that extrapolates data from successfully implemented quagi antennas and outputs proper dimensions was used. This program will be cited in section 6.4

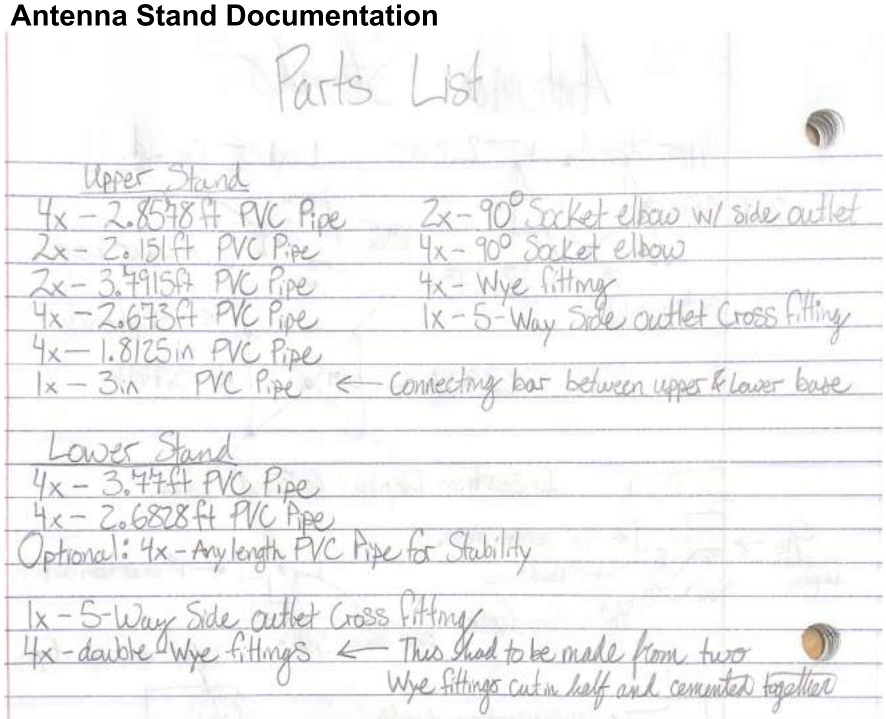
VHF Quagi Antenna



UHF Quagi Antenna



Antenna Stand Documentation


Parts List

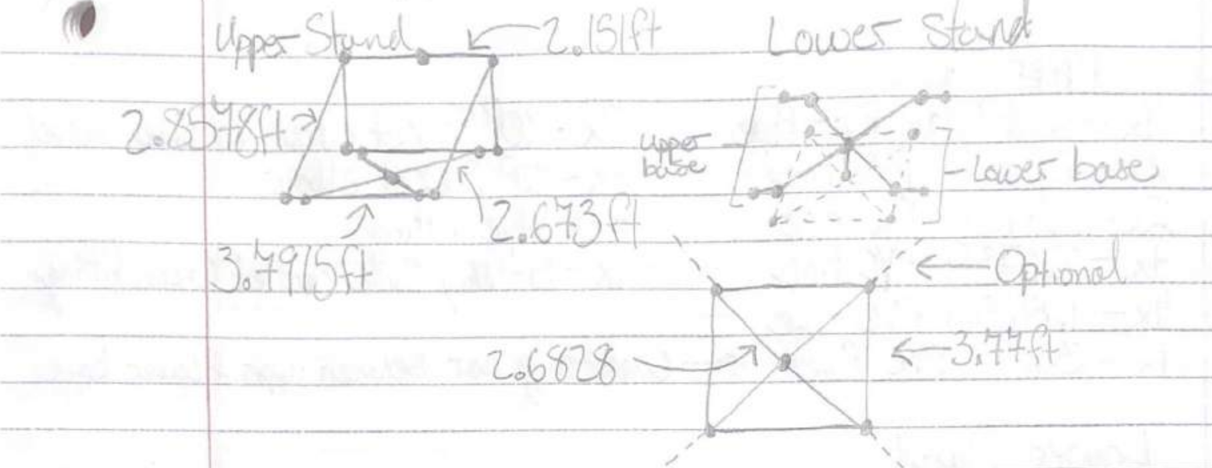
Upper Stand
4x - 2.8548 ft PVC Pipe
2x - 2.151 ft PVC Pipe
2x - 3.7915 ft PVC Pipe
4x - 2.673 ft PVC Pipe
4x - 1.8125 in PVC Pipe
1x - 3in PVC Pipe ← Connecting bar between upper & Lower base

2x - 90° Socket elbow w/ side outlet
4x - 90° Socket elbow
4x - Wye fitting
1x - 5-Way Side outlet Cross fitting

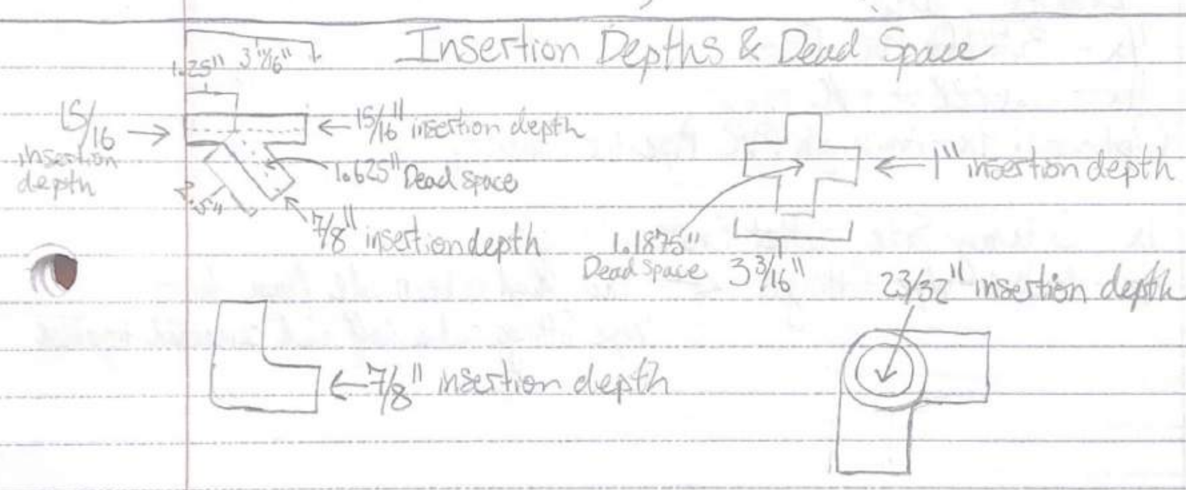
Lower Stand
4x - 3.77 ft PVC Pipe
4x - 2.6828 ft PVC Pipe
Optional: 4x - Any length PVC Pipe for Stability

1x - 5-Way Side outlet Cross fitting
4x - double Wye fittings ← This had to be made from two Wye fittings cut in half and cemented together

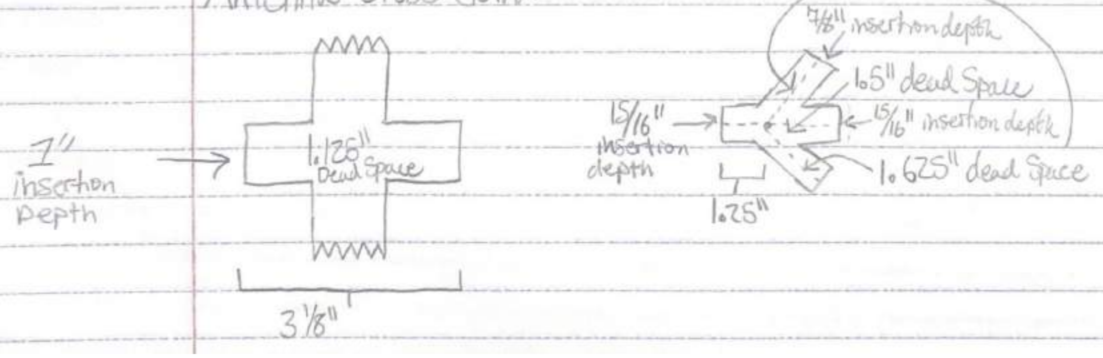
Antenna Stands



Insertion Depths & Dead Space

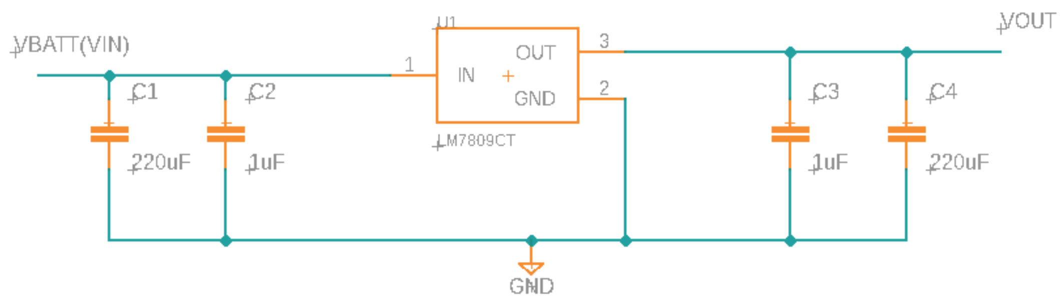


Antenna Cross Joint



6.2.2 Power Supply Artifacts

Power supply design was a simple 9V regulator circuit which was eventually deemed unusable due to efficiency and how long it would supply power issues. The link referenced here is the power supply that ended up being used. This is an adjustable voltage regulator meaning it has variable output voltage which was especially important as the output voltage changed depending on if it was the rocket subsystem load or the ground subsystem load [Adjustable V Regulator].



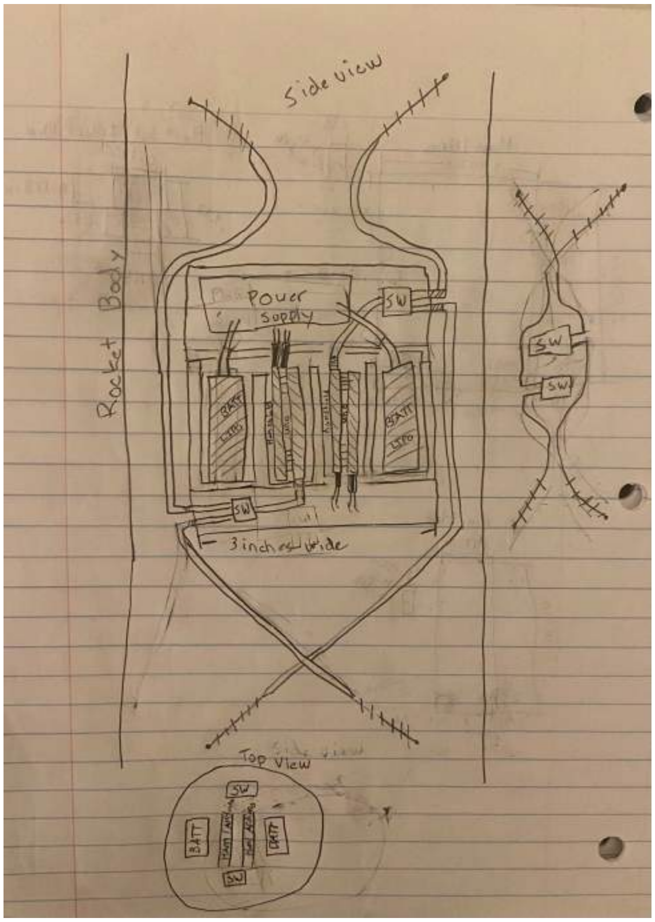
6.2.3 Code Artifacts

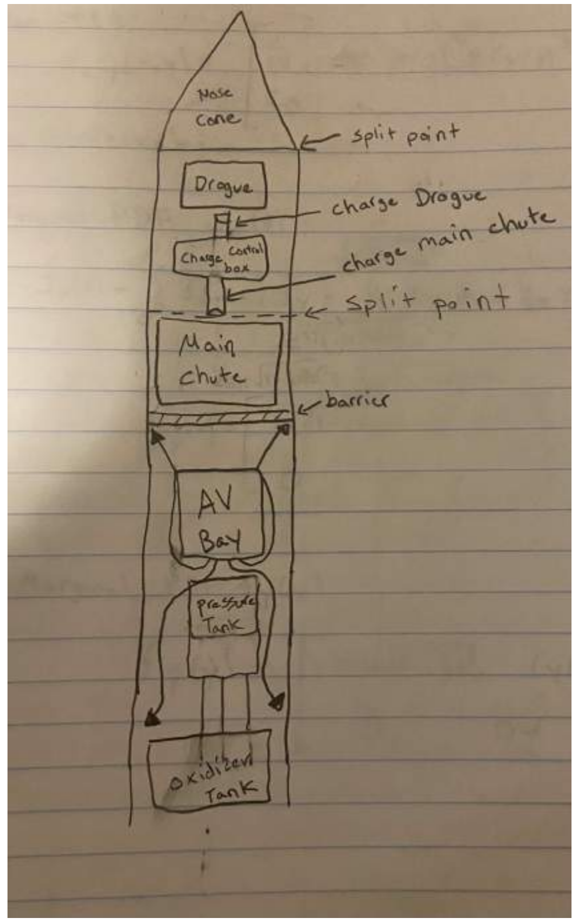
The code was originally sourced from the Hamshield example code found on their GitHub page. The example sketch that was used was AFSK_SerialMessanger.ino [8]. It was modified in four different ways for the four individual microcontrollers. The rate of transmission was increased and the data entry procedure was changed among other things. Instead of using the ‘ ` ‘ character after every string you want to transmit, spaces are included after every character. Spacing out characters in a string will rapidly send each character, one at a time. The code for the project can be found here in our code base. (Log into your OSU account to access it.) Lastly, the python script, which can graph and save the data received is also in the folder. It is named block1.py.

The GS_receiving_code.ino is meant to be flashed onto the Arduino Nano in the ground station, while GS_transmitting_code.ino is meant to be flashed onto the Arduino Uno in the ground station. The Rocket_Receiving_Code.ino will go on the Nano in the rocket module, while the Rocket_Transmitting_Code.ino will go on the Uno.

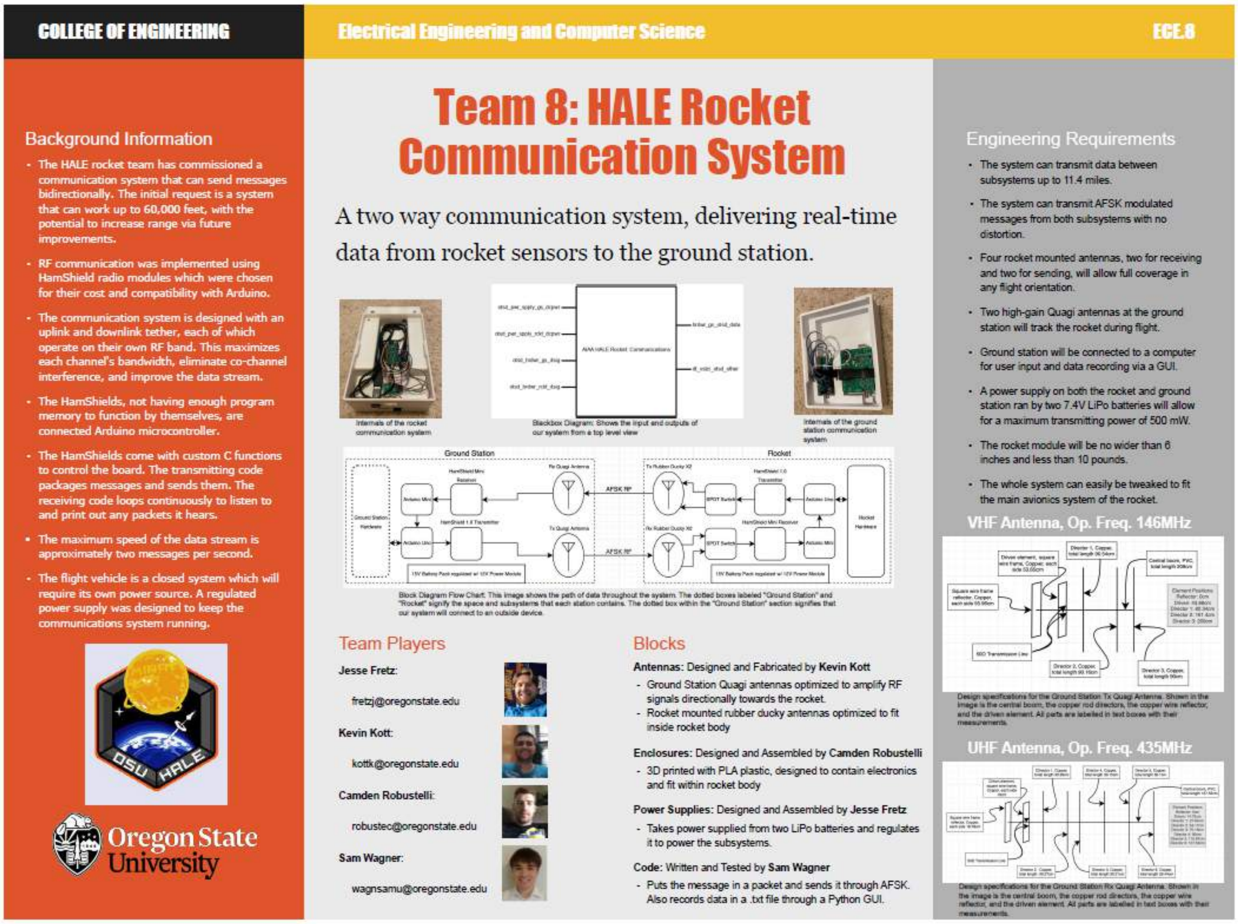
6.2.4 Enclosure Artifacts

The photos below show the configuration of how the enclosure is going to be with respect to the rocket and other subsystems. At the time of posting this document this is an accurate representation of where it's going to be located and how the antennas are going to mount within the rocket. There will be two antennas facing up at an angle, a sending and a receiving as well as two more facing downward. The left picture is an old enclosure version; it ended up being too wide when it was made in fusion 360 which led to the current enclosure version.





6.3 Presentation Materials



6.4 Citations

[1] EnhancedRadioDevices, “HamShield library reference · enhancedradiodevices/Hamshield Wiki,” *GitHub*. [Online]. Available: https://github.com/EnhancedRadioDevices/HamShield/wiki/HamShield-Library-Reference. [Accessed: 06-May-2022].

[2] "spaceneedle" and M. "mogar" Redfield, “EnhancedRadioDevices/Hamshield1: HamShield 1.0 hardware,” *GitHub*, 07-Aug-2019. [Online]. Available: https://github.com/EnhancedRadioDevices/HamShield1. [Accessed: 05-May-2022].

[3] Chris, “Can arduino damage (or kill) your computer?,” *Chip Wired*, 20-Mar-2022. [Online]. Available: https://chipwired.com/can-arduino-damage-your-computer/. [Accessed: 06-May-2022].

[4]“ARRL,” *Band Plan*. [Online]. Available: http://www.arrl.org/band-plan. [Accessed: 06-May-2022].

[5] “IEEE code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 06-May-2022].

[6] “Steps to building an effective team,” Steps to Building an Effective Team | People & Culture. [Online]. Available: https://hr.berkeley.edu/hr-network/central-guide-managing-hr/managing-hr/interaction/team-building/steps. [Accessed: 06-May-2022].

[7] "VE3SQB", “Quagi Antenna Design,” *VE3SQB ANTENNA DESIGN PROGRAMS*. Available: https://www.ve3sqb.com/. Software Available: https://www.ve3sqb.com/quagi.exe. [Accessed: 05-May-2022].

[8] EnhancedRadioDevices, “HamShield/AFSK_SERIALMESSENGER.Ino at master · enhancedradiodevices/Hamshield,” *GitHub*, 23-Dec-2019. [Online]. Available: https://github.com/EnhancedRadioDevices/HamShield/blob/master/examples/AFSK_SerialMessenger/AFSK_SerialMessenger.ino. [Accessed: 06-May-2022].

6.5 Revision Table

Date	Revisions Made
5/5/2022	Sam Wagner - Added content to technical and global impact recommendations Camden Robustelli - Added content to teamwork recommendations
5/5/2022	Kevin Kott - Added future recommendations, global impact recommendations, teamwork recommendations, and citations
5/6/2022	Jesse Fretz - Added future recommendations and design documentation
5/6/2022	Sam Wagner - Added code documentation, edited recommendations and added citations
5/6/2022	Kevin Kott - Edited future recommendations, global impact recommendations, teamwork recommendations, and added citations