Long-Range Transmission System for Sounding Rockets

Mei Qi Tang

260742835, ECSE 458D1

Jake Peters

260708582, ECSE 458D1

DP 19

Supervisor

Prof. Benoit Champagne

Sponsor

McGill Rocket Team

McGill University

May 1, 2020

Abstract

The McGill Rocket Team (MRT) has been having difficulty maintaining reliable communication links and recovering their 30 000 ft sounding rockets at the annual Spaceport America Cup (SAC). With the goal of helping the MRT solve its communications issues, this project aims to design a long-range transmission system for sounding rocket flights going up to 10 km. The first part of this project involves the design of a low-power long-range radio using a commercial transceiver chip and the second part consists of designing a beamforming phased array antenna. The MRT will be providing all software licenses and budget needed for the hardware components. During the Fall 2019 semester, progress consisted mostly of preliminary and first iteration design work. More specifically, we were able to choose the hardware components needed for the radio, complete the schematic of the radio's Printed Circuit Board (PCB), explore various beamforming antenna designs, and analyze each antenna design's performance through simulations. Altium Designer was used for PCB design and computational simulation software like MATLAB and Ansys HFSS was used for the beamforming antenna design. In the Winter 2020 semester, we finalized and manufactured our PCB design for the radio and completed scripts to operate the transceiver chip as both a transmitter and receiver. Additionally, we implemented our own beamforming simulator in Python to plot the radiation pattern of an antenna array, ultimately confirming with the MATLAB simulations done in Fall 2019.

Table of contents

i. Abbreviations

BER	Bit Error Rate
BPF	Band Pass Filter
COTS	Commercial-Off-the-Shelf
GCPW	Grounded Coplanar Waveguide
CRC	Cyclic Redundancy Check
CSS	Chirp Spread Spectrum
EM	Electromagnetic
FSPL	Free Space Path Loss
MRT	McGill Rocket Team
MISO	Master In Slave Out
MOSI	Master Out Slave In
PCB	Printed Circuit Board
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SAC	Spaceport America Cup
SCK	Serial Clock
SMD	Surface Mount Device
SPI	Serial Peripheral Interface
SRAD	Student-Researched-and-Developed
SS	Slave Select
TCXO	Temperature Compensated Crystal Oscillator
ULA	Uniform Linear Array

I. Introduction

Driven by 100+ passionate students, the McGill Rocket Team (MRT) is a university design team that designs and manufactures sounding rockets. Alongside more than 100 other teams across the world, the MRT participates at annual sounding rocket competitions like the Spaceport America Cup (SAC). The MRT has flown three 30 000 ft sounding rockets for the last three SACs, but has yet to successfully maintain a communication link for the entirety of flight and recover them. Recoverability is an important feature of sounding rockets since they are meant for carrying a scientific payload to perform sub-orbital experiments. Losing communications with the rocket implicates the loss of not only the payload, but also an expensive airframe that could be flown again.

This project aims to design a long-range transmission system that is able to cover up to 15 km for the MRT's sounding rockets. This transmission system will include a low-power long-range transmitting radio, designed using Semtech's transceiver chips, and a beamforming phased array antenna with beamforming circuitry. The ultimate goal is to design, prototype, test, integrate, and fly this transmission system with the rest of the MRT's Avionics for future SACs. This project should be well documented enough to be a project extending beyond the scope of this course, potentially spanning multiple years.

The design of our radio will be based on Semtech's SX1262 transceiver chip which will require the use of Altium Designer, a printed circuit board (PCB) design software, to design the PCB layout for the integrated radio. With the design reviewed and finalized, we will receive our manufactured PCB from a manufacturer, and we can then use our reflow oven to solder on the components. Once fully assembled a microcontroller will be used to run software programs to operate the radio as a transmitter or receiver to communicate between devices.

The preliminary design of single antenna elements as well as beamforming antenna arrays will be done using the Antenna Toolbox^{TM} and the Phased Array System Toolbox^{TM} in MATLAB. MATLAB is used due to its simplicity of use and its built-in antenna models. Once the preliminary designs are set, Ansys HFSS, a 3D electromagnetic (EM) simulation software, will be used to more detailed antenna modelling and tuning for performance optimization. The radiation pattern designed using HFSS can be then exported to MATLAB for further beamforming simulations. Then, to better understand the mathematical model of the beamformer, Python on Jupyter Notebook will be used to implement the gain pattern equation and compare with MATLAB's functions.

The MATLAB software license is provided by McGill but both Altium Designer and Ansys HFSS licenses are provided by the MRT through sponsorships. Prototyping tools such as the spectrum analyzer are made available by a collaboration with the Factory, a student-run hardware design lab for students in the department of Electrical, Computer, and Software Engineering at McGill.

This report first presents the background necessary about the MRT's rocket design for SAC 2020, the rocket's flight dynamics with respect to the communications system, the SAC 2019 transmission system's inefficiencies, and some theory on radio and beamforming antenna design. The report then presents the requirements, design, and results for both the radio and beamforming antenna subsystems. Lastly, the societal and environmental impact of rockets and this project, as well as the teamwork involved in this project are discussed further.

II. Background

A. Spaceport America Cup and Rocket Overview

The yearly SAC happens in Las Cruces, New Mexico, USA, near the newly built Spaceport America for commercial space launch vehicles. The SAC involves 100+ teams participating in one of the 6 categories for either 10 000 ft or 30 000 ft altitude, either Commercial-Off-the-Shelf (COTS) or Student-Researched-and-Developed (SRAD) motor, and either solid or hybrid/liquid motor. Rockets reaching the 10 000 ft or 30 000 ft altitude are categorized as sounding rockets since they are designed to be suborbital, delivering a payload to a high altitude for scientific purposes. Sounding rockets are generally single stage and should be fully recoverable and reusable for future flights, which provides much lower overall launch costs than orbital launches.

For the SAC 2020 happening this upcoming June, the MRT will again be participating in the 30 000 ft solid COTS motor category for the fourth consecutive year. The MRT's SAC 2020 rocket is composed of 3 main separable sections, notably the Nose Cone, the Top Body Tube, and the Bottom Body Tube, as shown in Figure 1. The flight electronics, which we generalize as the Avionics, will be sharing the Top Body Tube space with the Recovery system, which contains recovery gear such as parachutes. The total allocated length for the Avionics is 16 inches in a 5-inch inner diameter tube.

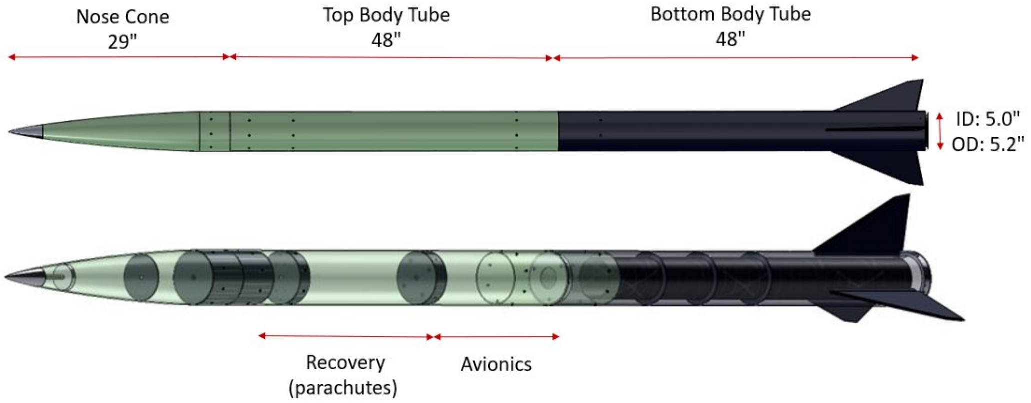


Figure 1. SAC 2020 rocket main sections overview.

An overview of the general flight profile is shown in Figure 2, separated in three main phases by four events. The first event is Launch, happening at 1km from the ground-station. The second event is the Drogue Parachute Deployment, happening after having apogee at around 9.1 km. The third event is the Main Parachute Deployment, happening after having reached 0.5 km. Lastly, the rocket lands at touch down. Each flight phase can be approximated as a different linear relationship between the altitude and the range from the ground-station. This linear relationship will be used for the beamformer to map real-time altitude readings to the estimated steering angles.

Using the OpenRocket flight simulator (see Appendix A), the rocket is predicted to have a max velocity of Mach 1.7 (1.7 times the speed of sound) and a max acceleration of 14.9 g experienced at the peak thrust, 5 seconds after liftoff. These flight dynamics will be taken into account while designing our transmission system to be robust enough against the mechanical loads and vibrations.

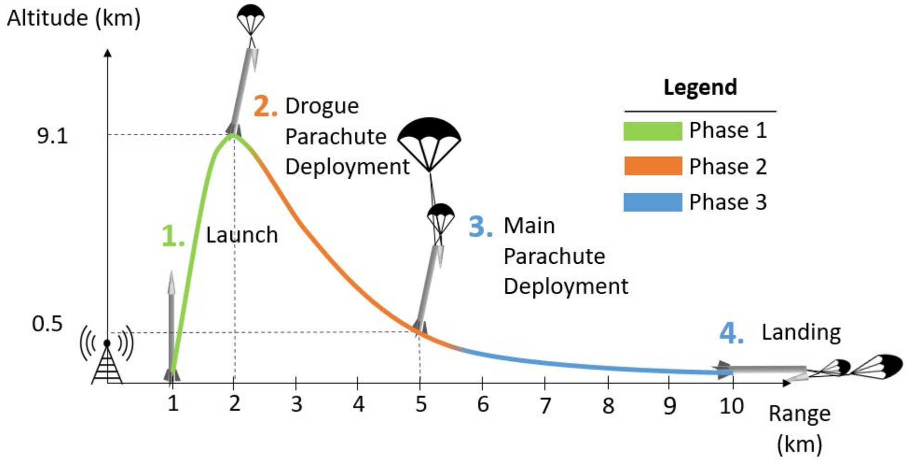


Figure 2. Flight events and phases overview.

C. SAC 2019 Design

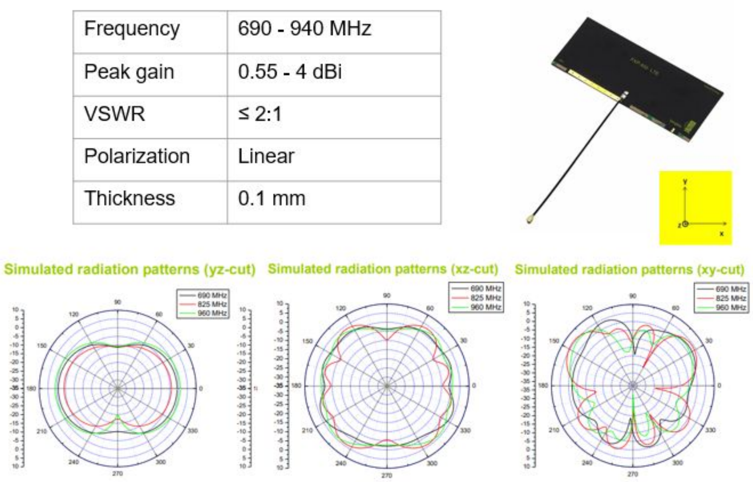
SAC 2019's transmission system used the XTend vB, a commercial radio from Digi International operating in the 900 MHz ISM band with the highest transmit power allowed of 1W (see Table 1). The radio was configured to hop with a random sequence between 50 channels within the frequency range to avoid interference with other transmission sources also using that band. This radio was chosen due to its ease of usage as a commercial solution, its high transmit power, and its operation in the license-free ISM band. The main issue with this radio is its inability to configure a specific transmission channel as well as its high unit cost of 300$.

Operating frequency range	902 MHz - 928 MHz
Transmit power	30 dBm (1W)
Receiver sensitivity	-110dBm at 10kb/s of data rate
Modulation	Frequency Shift Keying - Frequency Hopping Spread Spectrum
Channels	Hopping between 50 channels

Table 1. Digi's XTend vB radio specifications.

The type of antenna chosen were Taoglas's High Efficiency Ultra Wide-Band FXP.400 antenna. This antenna is a microstrip half-wave folded dipole, therefore it has a somewhat toroidal radiation pattern, as shown in Figure 3. The integrated Avionics Bay with all the antennas is shown in Figure 4. We can see that the biggest design flaw for SAC 2019 was that each telemetry antenna was occupying the same vertical space as not only the other telemetry antenna, but also some conductive elements in the

Avionics bay, such as PCBs. Being mounted in the same vertical space as other conductive elements will highly interfere with the radiation performance of the antenna, decreasing drastically its efficiency. Indeed, the much lower Received Signal Strength Indicator (RSSI) read after having the Avionics Bay assembled showed much more attenuation in the signal.



Frequency	690 - 940 MHz
Peak gain	0.55 - 4 dBi
VSWR	≤ 2:1
Polarization	Linear
Thickness	0.1 mm

Figure 3. SAC 2019 commercial telemetry antenna.

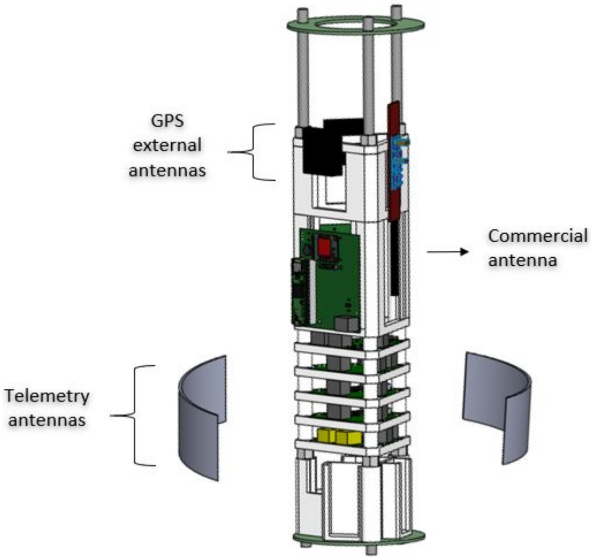


Figure 4. SAC 2019 Avionics Bay - Antenna overview.

D. Radios

In our communication system, the radio has two main functions, the first is to encode collected sensor data into EM waves to be transmitted from the rocket, and the second is to decode these received EM waves into comprehensible data on the ground. These sensors include: an environmental sensor which outputs the temperature and pressure of the rocket, a 9 Degrees of Freedom sensor that will output the acceleration and orientation of the rocket, and a GPS sensor to record the rocket's current latitude and longitude.

However, once these EM waves have been transmitted, there becomes many reasons for the degradation of the strength of these signals. The first reason for the loss of power in our radio signal that we must consider is loss due to travelling through the medium, or specifically the equation of free-space path loss (FSPL) which relates the attenuation of a radio signal's energy to the distance travelled and the wavelength of the signal, Equation 1.

$$FSPL = \left(\frac{4\pi d}{\lambda}\right)^2 \tag{1}$$

With this formula, we find that the loss increases with the square of the distance between antennas. This loss of energy is due to the spreading of the radio wave as it propagates through the air. We can visualize this idea, Figure 5, by showing how the power density, red arrows, of the radio wave decreases as it spreads over areas proportional to the square of the distance from the transmitting antenna. We can also note that the loss decreases with the square of the signal's wavelength which emphasizes the desire for a lower carrier frequency.

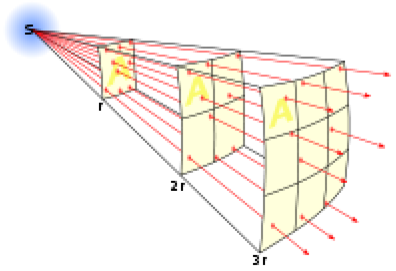


Figure 5. Visualization of FSPL

Another reason for degradation of signal strength is from collisions and absorption from obstructions in the path of transmission. This leads to another theory, called the Fresnel zone, Figure 6, which defines the necessary region of free space needed between antennas so obstructions, such as the ground, mountains or trees, do not weaken the radio wave. From this theorem, we conclude a general rule of thumb that 60% of the first Fresnel zone should be clear of any obstructions in order for the radio signal to behave as if it is in "free space" [1]. If not, reflected signals will interfere with the direct signal and cause the free space path loss to increase with the fourth power of the distance travelled instead of the square.

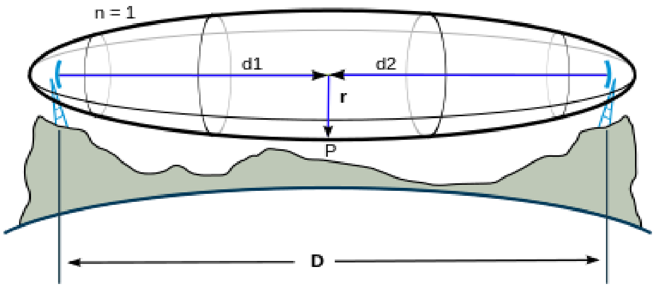


Figure 6. The Fresnel zone

The final background information needed for the EM wave portion of the radio is the modulation technique employed by the transceiver chip. There are various modulation methods used to encode data into EM waves but we will focus on chirp spread spectrum (CSS) modulation. CSS was developed in the 1940's and has been traditionally used in military and radar applications however, recently this modulation technique has seen increased adoption in communication application due to its relatively low power requirements and inherent robustness from channel degradation [2]. LoRa modulation, is a proprietary variation of CSS developed by Semtech. In LoRa and CSS, information is transmitted using up and down chirps, seen in Figure 7. Due to its large bandwidth time product, the signal has a narrow autocorrelation function and can be correlated easily. This property allows the signal to be more resilient to multipath propagation and the doppler effect, which is important with rockets as they move quickly.

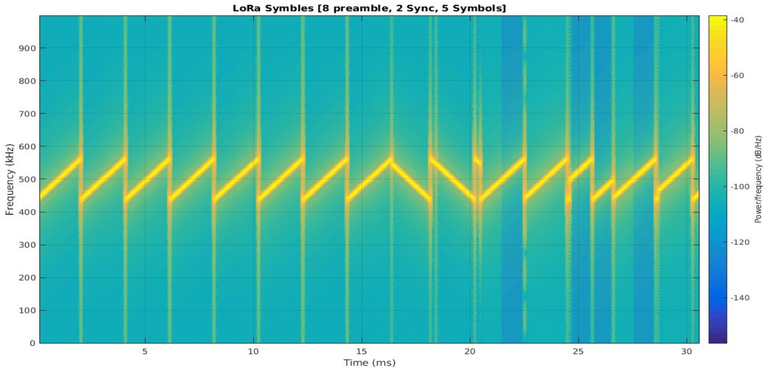


Figure 7. LoRa modulation signal, Frequency vs. Time

The last key detail that should be known is for the hardware portion of the radio. As we will be creating PCBs with high speed analog signals, we must also consider characteristic impedance control throughout these transmission lines. When designing the transmission lines for the PCB, we must ensure impedance matching throughout the design to minimize power reflection, which causes loss of power in the signal. Due to most components input and output ports having 50 Ω impedance, this means the transmission lines also need to have a characteristic impedance of 50 Ω as well. The impedance of the transmission line is calculated from various attributes such as the dielectric constant of the PCB material, the substrate height, the trace width and trace thickness. We will be considering two common transmission line designs for our PCB, the microstrip and coplanar waveguide transmission lines, Figure 8.

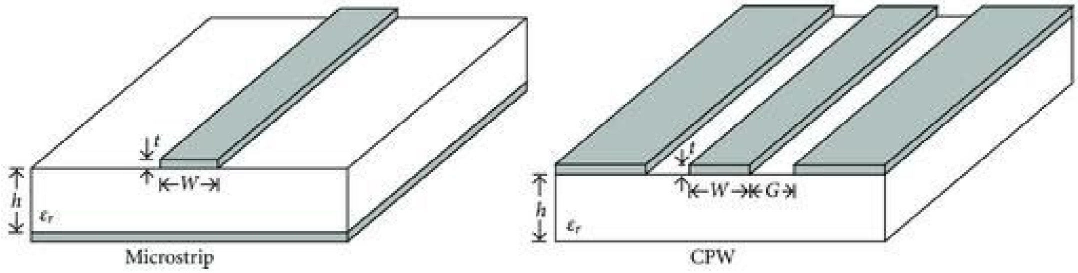


Figure 8. Mircrostrip transmission line (left) and Coplanar waveguide transmission line (right)

E. Antennas

1. Operating frequency and wavelength

Every antenna is designed specifically for a range of operating frequency since its dimensions will be directly proportional to the wavelength, such as half-wavelength. For antenna arrays, the element spacing will also be determined as a factor of the wavelength, such as quarter-wavelength. The wavelength is inversely proportional to the operating frequency, given by Equation 2.

$$\lambda = \frac{v}{f} , \tag{2}$$

where

λ = wavelength
f = operating frequency
v = speed of propagation.

Due to this inverse relationship, the antenna dimensions and spacing will be greater for lower operating frequencies. Therefore, it is important to choose the operating frequency carefully otherwise the antenna might not fit in the rocket.

2. Impedance matching: Z-parameter and Smith Chart

Impedance matching refers to designing the antenna and all transmission lines to have the same characteristic impedance. An impedance mismatch will result in power being reflected back to the transmitting source, decreasing the overall signal strength through transmission. Characteristic impedance will be denoted by Z_0 and the normalized impedance will be denoted by z_0 .

3. Reflection coefficient: S-parameter

The reflection coefficient Γ indicates how much the EM wave is reflected back to the source by an impedance discontinuity. This parameter is a direct result of how well the impedance matches as per the previous requirement. Modelling the antenna as a load to a circuit, the reflection coefficient is described by Equation 3.

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} , \tag{3}$$

where

Γ = reflection coefficient
Z_L = load impedance
Z_0 = characteristic impedance

In HFSS, we can design for optimal reflection coefficient with S-parameters. The S11 parameter is equivalent to the reflection coefficient at the input port.

4. Radiation pattern and beamwidth

The radiation pattern describes the radiation behavior of an antenna with respect to space, usually in a polar coordinate system. It illustrates the relative signal power compared to an isotropic antenna at specific positions (azimuth or elevation angles) in space. The efficiency of the radiation pattern is highly dependent on the application. In fact, it is essential to consider the flight trajectory when determining whether an antenna's radiation pattern is efficient.

5. Antenna arrays

Antenna arrays are when multiple antenna elements transmit the same signal together, to increase the overall transmission efficiency in a specific direction. Antenna arrays are usually made of multiple identical individual antenna elements. Despite each having their own radiation behaviour, when placed at specific element spacing between each other, the antenna array will radiate according to its own radiation pattern. Radio waves will interfere with each other, whether constructively or destructively, to create the new radiation pattern.

5. Beamforming and beam steering

Beamforming is a signal processing technique used in antenna arrays for directional signal transmission if implemented on the transmitter or for signal reception if implemented on the receiver. Beamformers can also have a beam steering capability, electronically steering the radiation beam towards a desired direction. The 2 and 3-Dimensional geometry of a plane wave incident on an array of two identical antennas, at an angle θ, is illustrated in Figure 9.

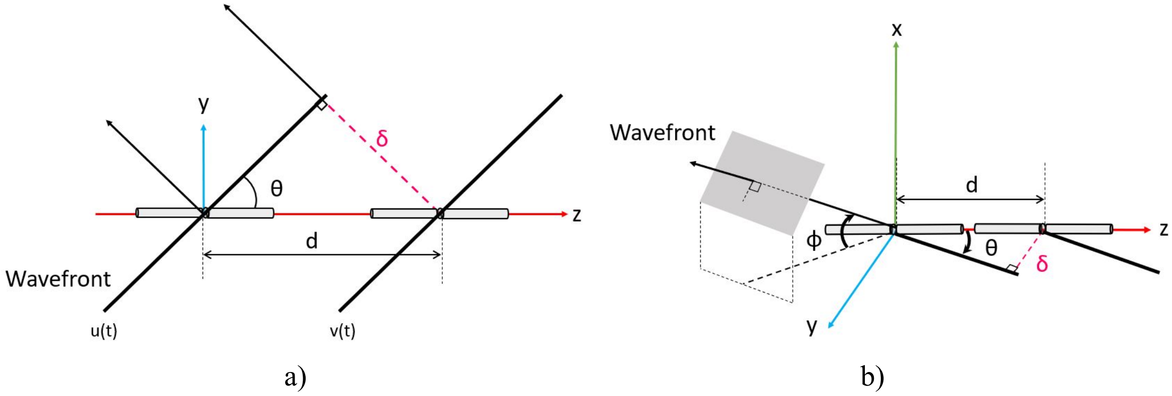


Figure 9. a) 2D and b) 3D geometry of plane wave incident on a 2-element antenna array.

We have defined the antennas to be positioned along the z-axis, in this specific configuration to stay consistent throughout the derivations and implementations. Since the antenna array in our design lies in this configuration, we will be deriving the equations in this configuration as well.

Let $u(t)$ denote the signal representing a plane wave transmitted at the reference element:

$$u(t) = e^{j\omega t} . \tag{4}$$

Let $v(t)$ denote the signal transmitted at the second sensor element, which is delayed by τ, the time required for the plane wave to propagate through δ:

$$v(t) = u(t-\tau)\,, \tag{5}$$

where $\delta = dsin\theta$ by geometry demonstrated in Figure 9. Given that the two sensor elements are separated by a distance of d, the time delay to propagate through δ with speed of of light $c = 2.998 \cdot 10^{8} m/s$ can be calculated as:

$$\tau = \frac{dsin\theta sin\phi}{c}\,. \tag{6}$$

Hence, the signal can be then expressed as:

$$\begin{aligned} v(t) = u(t-\tau) = u(t)e^{\frac{-j(\omega_0 dsin\theta sin\phi)}{c}} &= u(t)e^{\frac{-j2\pi f(dsin\theta sin\phi)}{c}} \\ &= u(t)e^{\frac{-j2\pi(dsin\theta sin\phi)}{\lambda}}\,, \end{aligned} \tag{7}$$

where $\lambda = \frac{c}{f}$ is the wavelength Since we have the sensor elements separated by a distance of $d = \frac{\lambda}{2}$ to avoid forming sidelobes, we can further simplify the equations as

$$v(t) = u(t)e^{-j\pi(sin\theta cos\phi)}\,. \tag{8}$$

If we add more elements to the array, we can then generalize the equation as

$$v(t) = u(t)e^{-jm\pi(sin\theta cos\phi)}\,, \tag{9}$$

where N is the number of elements and $m = 0, 1, ..., N$. Now, ignoring noise, we can express the signal transmitted at each element with a generalize expression

$$x_i(t) = u(t)e^{-jm\pi(sin\theta cos\phi)}\,. \tag{10}$$

Furthermore, we can introduce some weight w_i to each signal using phase delays

$$w_i = \frac{1}{\sqrt{N}}e^{-jm\pi(cos\theta_0 cos\phi)}\,,\ m = 0, 1, ..., N\,, \tag{11}$$

where θ_0 is the steering angle. Since each of the individual signals will add up with each other through superposition once transmitted in the air, we have a total transmitted output signal of

$$y(t) = \sum_{m=1}^{N} w_i x_i(t) = \sum_{m=1}^{N} \frac{1}{\sqrt{N}}u(t)e^{jm\pi(cos\theta_0 - sin\theta)cos\phi} = \frac{1}{\sqrt{N}}\sum_{m=1}^{N} u(t)e^{jm\pi(cos\theta_0 - sin\theta)cos\phi}\,. \tag{12}$$

A beamforming array of 4 elements is depicted in Figure 10.

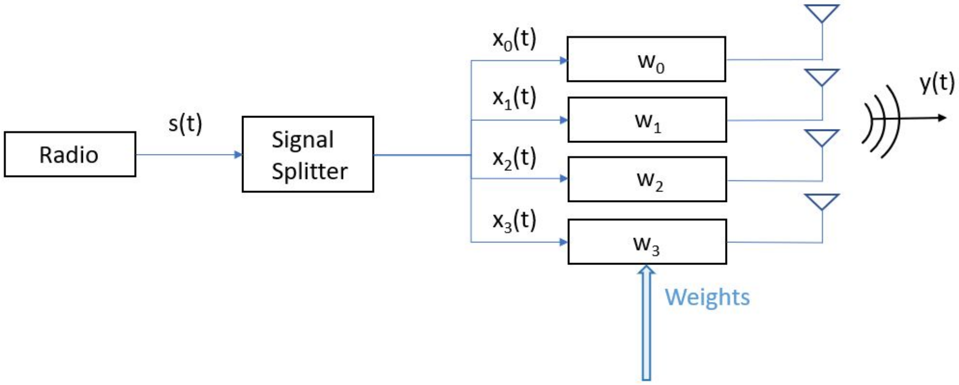


Figure 10. High-level block diagram of a 4-element beamforming transmitter.

The array factor and be expressed as

$$F(\theta_0, \theta, \phi) = \sum_{m=1}^{N} e^{jm\pi(cos\theta_0 - sin\theta)cos\phi} \,, \tag{13}$$

and the normalized gain pattern of the beamforming phased array can be expressed as

$$\begin{aligned} G(\theta_0, \theta, \phi) &= \left|y(t, \theta_0, \theta, \phi)\right| \\ &= \frac{1}{N}\left|\sum_{m=1}^{N} w_m x_m(t)\right|^2 = \frac{1}{N}\left|F(\theta_0, \theta, \phi)\right|^2 \\ &= \frac{1}{N}\left|\sum_{m=1}^{N} e^{jm\pi(cos\theta_0 - sin\theta)cos\phi}\right|^2, \; m = 0, 1, ..., N \end{aligned} \tag{14}$$

By adjusting the set of weights $\{w_m\}$, it is possible to beam steering and direct the maximum of the main beam of the array factor in any desired steering angle, in terms of azimuth and elevation angles (θ, φ).

III. Problem and Requirements

A. Problem Overview

The main problem we wish to address with our project is to design a communication system specifically for sounding rockets. In the past, we have had to use commercial products as there are no products that perfectly fulfill the requirements for our desired use. Hence, our goal is to design a radio and antenna pair that satisfy the requirements set by a sounding rocketry, listed below.

B. Radio Requirements

The design considerations for the radio and its operation fall under 5 main requirements. These requirements help in the decisions of which components to use for the radio, the construction of the printed circuit board (PCB), the transmission power of the radio, the operating frequency and many more.

1. Maintain communication link for a minimum of 15 km

The first requirement that the radios must meet is that they need to communicate for a minimum distance of 15 km, with line of sight. The minimum communication link distance is the expected maximum distance the rocket may be from the ground station in a nominal flight. While this distance could be larger in the case of a ballistic flight, we chose not to base our design on this scenario as recovering the rocket would likely not be possible. However, we hope that our design can operate above our minimal distance in case of signal loss due to obstructions between the two radios and other interfering signals.

2. Reliable communication link

The second requirement is that the communication link should be reliable and fairly resilient to other interfering signals. This is important as the information received from the rocket contains the GPS coordinates which are very important and should not be altered during transmission. As a low bit error rate is difficult in wireless communication, especially when many other radios will be transmitting around the same frequency, some packets are expected to contain errors. However, we need these erroneous packets not affect the overall localization capabilities of the telemetry system.

3. Minimal interference with other electronics

With other radios and electromagnetic radiation sensitive components on board the rocket, it is of high importance that the designed radio does not affect these electronics. This means that we wish to avoid overlapping in transmitting frequency with the other radios on the rocket, which are centered at 435 MHz and 920 MHz. As well, to ensure that no unwanted electromagnetic interfering signals are outputted we can simulate our design and use shielding to avoid interfering with other components and to protect our system from other EM interference.

4. Sustain environmental effect of rocket

A hardware requirement that must be met is to ensure that the transmitter, which will be on the rocket, will be able to sustain the forces that will be acted on it, as well as the high temperatures of the desert sun. This requirement entails secure soldering of all components onto the PCB, a low g-sensitivity crystal oscillator to minimize frequency drift during high acceleration periods, and heat resistant components that can endure being in an enclosed rocket in the desert for a few hours, which from experience when operating all the avionics electronics could reach above 55 °C. As the receiver on the ground will not be transmitting, heat dissipation is less of an issue, however, sand from the desert could be a problem so an enclosure for the receiver will be needed.

5. Operate in accordance to the law

The final requirement for the radios is that our design must be in accordance with the laws of Canada's Innovation, Science and Economic Development Canada department as well as the United States' Federal Communications Commision, because our competition will take place in New Mexico. This means that we must transmit within the legal allotted amateur frequency bands, as well as the maximum output power into the antenna feed must be 1 Watt (30 dBm) or less.

C. Antenna Requirements

The antenna design will be mainly constrained by the chosen operating frequency, the available space of where to mount the antenna, and the surroundings of the potential mounting location. Furthermore, the antenna will have some standard performance requirements to meet with respect to its impedance matching, return loss, radiation pattern, and beamwidth.

1. Operating frequency and wavelength

Since the antenna's dimensions are inversely proportional to the frequency, and directly proportional to the wavelength, it is important to choose the operating frequency carefully since the antenna will need to be physically integrated with the rocket. If the frequency is too low, the antenna might be too big to fit in the rocket.

2. Space and surroundings

The rocket for SAC 2020 will be in total 10 feet tall with 5/5.2 inches inner/outer diameter. However, the rocket detaches its nose cone after reaching apogee to deploy its recovery parachutes and the bottom section of the airframe is not RF transparent. Hence, there is only 48 inches of effective available length to mount the beamforming array antenna.

3. Impedance matching: Z-parameter and Smith Chart

It is important to match the impedance for all RF components such as the ports, the transmission lines, and antennas. Following the industry standard, we will be matching our designs as close to $Z_0 = (50 + 0i)\,\Omega$ as possible. The Smith Chart normalizes the impedance so we should be aiming for $z_0 = 1 + 0$i.

4. Reflection coefficient: S-parameter

Since the reflection coefficient Γ indicates the amount of signal power being reflected back to the source due to an impedance discontinuity, it is desired to minimize the reflection coefficient as much as possible. However, since it is most of the time not possible or practical to exactly match the impedance, we set a minimum requirement for the reflection coefficient to be -10 dB for the operating frequency range.

5. Radiation pattern and beamwidth

The radiation pattern should be omnidirectional in horizontal planes due to the unpredictable spinning nature of the rocket. However, the beamforming radiation pattern should show the radiation beam being directed towards a specific steering angle. Depending on how precise the beamsterring and direct the main lobe, the beamwidth could be as narrow as 30°.

IV. Design and Results

A. Operating Frequency

The operating frequency for this project was chosen to be centered at 915 MHz. The 440 MHz amateur band would have been a more suitable choice in terms of a greater link budget due to its lower path loss. However, since the antenna dimensions are inversely proportional to frequency, we have not yet found a potential design of a beamforming antenna array that could fit inside the at 440 MHz.

B. Radio Design

Design Decisions:

1. Choice of transceiver chip

During the radio design process, several decisions had to be made. The first decision was finding the right transceiver chip to base the radio around. We ended up deciding on choosing Semtech's new SX1262 transceiver chip. The chip has one of the largest link budgets, sum of transmitting power and receiver sensitivity, on the market at 170 dB, however this number is only achievable at the lowest data rate. The reason for its spectacular receiver sensitivity, at -148 dBm, comes from the LoRa modulation used when transmitting, which is a proprietary variation of chirp spread spectrum. As well with a large link budget it covers 3 major amateur bands in North America, the 1.25 m band (222 MHz to 225 MHz), the 70 cm band (430 MHz to 450 MHz) and the 33 cm band (902 MHz to 928 MHz). While it does suffer from a lower data rate compared to other options, this is not an issue as we do not require a large throughput. Finally, we were able to acquire a sponsorship with Semtech, and we received 20 SX1262 chips, so the price did not become a hindrance.

2. Filtering

With the number of interfering signals expected at the competition, we decided it would be intelligent to include filtering to minimize the effects of the noisy environment. However, we needed to make a choice between a higher insertion loss narrow band band pass filter (BPF) or a lower insertion loss wide band BPF. We concluded that the narrow band BPF would suit us best as we have a large link budget from the transceiver chip and minimizing the interfering signals will play a large role in our signal reception as many radios will be transmitting close to our operating frequency. We decided to choose a surface acoustic filter (SAW) from Golledge, the TA0521A. Thankfully, we were able to acquire the filter as a sample, so no purchasing was required.

Radio Printed Circuit Board:

Progress from Previous Term:

With the initial design decisions completed for the radio, we could now begin working on the PCB to create the radio. To design the PCB we used the software Altium Designer. The licence we received is part of a sponsorship that the McGill Rocket Team has with Altium Designer. This software costs several thousand dollars per license and has allowed us to design our radio with much more configurability compared to other free PCB designing software.

The first step when designing a PCB is creating the component schematics and PCB footprints. The component schematics are used to represent the component in the schematic document, Figure 11, to show what net each component is connected to. This allows readers to easily examine the design without trying to figure out what components are connected from the PCB layout. In the schematic document of the radio, directly above the SX1262 chip is a crystal oscillator with a frequency of 32 MHz. This chosen oscillator serves as a reference frequency for the chip. It has a g-sensitivity of 2 ppb/G, which means it drifts $2x10^{-9}$*32MHz per G applied to the oscillator. While this is not the lowest possible g-sensitivity oscillator, the acceleration on the radio will be short lived, with a max of around 5 gs, and the most important period of the radio communication, during the descent, the rocket will experience very little acceleration.

On the rightmost side of the schematic in Figure 11, we can find the SMA port which connects to an antenna, or signal splitter for the transmitter. Moving left from the connector, we find an RF switch. This component allows the SX1262 to switch between ports 1 and 3 when transmitting and receiving, allowing us to use one antenna for both directions of communication. On the left side of the RF switch, the upper trace is connected to the power amplifier and output of the transceiver. The lower port is connected to the differential low noise amplifiers of the chip used when receiving incoming packets. We can also observe that before the received signal reaches the transceiver, it passes through a BPF to attenuate any unwanted frequency components of the incoming signal.

To communicate with the SX1262 chip, we use Serial Peripheral Interface (SPI) communication protocol. This is accomplished by using the MISO (Master Input Slave Output), MOSI (Master Output Slave Input), SCK (Serial Clock) and NSS (Slave Select) pins and connecting them to a microcontroller to send the transceiver chip commands. The SX1262 also uses a BUSY line to indicate when the chip cannot receive commands and three interrupt lines, which can be mapped to several interrupts available, such as indicating when a packet transmission is completed or if a wrong CRC is received.

Another important remark about our design is the component values we used in our schematic. As we were unable to find a reference design for our exact desired frequency, we considered completing our own impedance matching test. However, to create a circuit that would allow us to vary the impedance at the transceiver ports, to determine the optimal presented impedance for maximum power transfer, would have delayed our design significantly. So we decided to use a reference design based on a slightly higher operating frequency, knowing that some power may be reflected back, but allowing us to skip the load pull process and saving us a notable amount of time.

The final remark about our initial design are the capacitors and inductors at the transmitter output of the SX1262, seen on the upper trace to the left of the RF switch in Figure 11. These components are used to filter the outgoing signal by first passing through a notch filter centered at the second harmonic of the transmitted signal, then through a low pass filter. In most radios, all the filtering is done at the receiver end, however, because we have GPS antennas operating at higher frequencies than our radio, the filtering of those higher frequency components becomes beneficial.

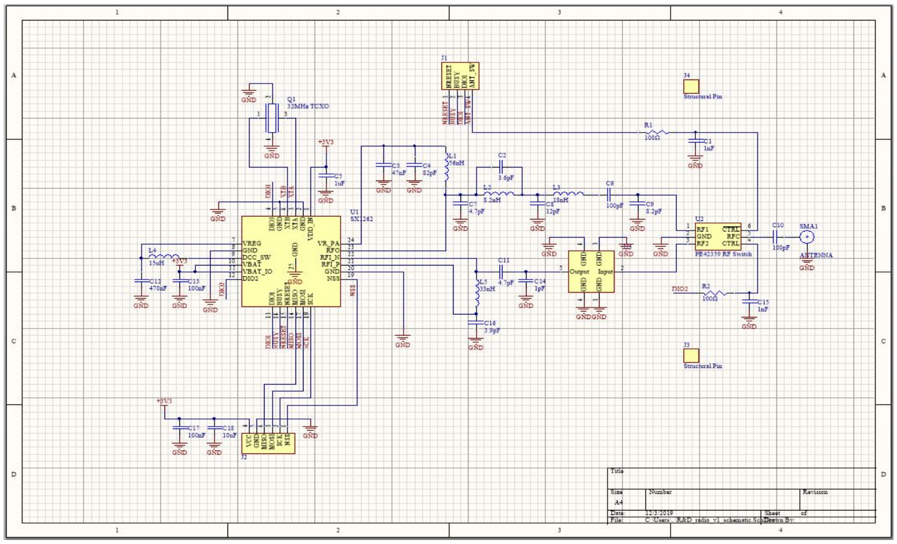


Figure 11. Radio Transceiver PCB Schematic

After all the components were placed on the schematic diagram, we were then able to move on to the PCB layout, which is a physical representation of each layer for your PCB.

The first notable feature in the PCB layout is the polygon cutout around the crystal oscillator, this minimizes any transferring of heat created from the SX1262 to the oscillator, reducing frequency drift. Another feature is the large polygon pour beginning at the leftmost pin of the lower header pins, the +3.3V power pin. The reason for making the net large is to avoid any problems supplying the chip with enough current during its peak power consumption. We also used vias to dissipate heat and grounded vias to isolate our transmission lines from EM interference.

We also calculated that the trace width needed to be 0.16 mm to achieve a characteristic impedance of 50 Ω for our microstrip transmission lines. This trance width was found by an online calculator which calculates to required trace width when given the properties of the circuit board.

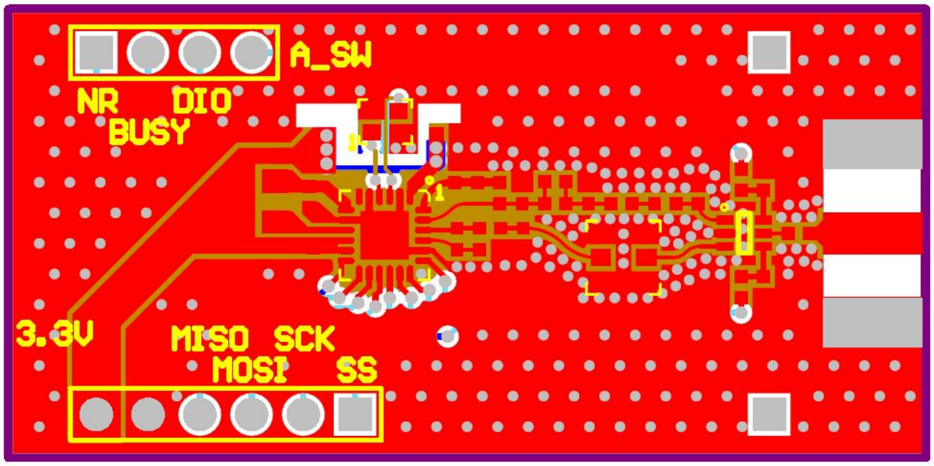


Figure 12. Top layer of PCB layout

Current Term Progress:

During this current term, we were able to receive some critical feedback on our initial design from an engineer at Semtech. After many messages between each other we were able to enhance our initial design to improve the RF characteristic and performance of our design.

The first change we made was to replace the crystal oscillator with a Temperature Compensated Crystal Oscillator (TCXO) to minimize frequency drift. In previous years data, we found that the avionics bay can reach temperatures up to 55 ℃, which can negatively affect the operating frequency and cause the radio to drift from its intended transmitting frequency. To minimize this unwanted frequency drift, we switched to a TCXO to allow us to vary the input voltage to slightly alter the frequency of the oscillator to compensate for the frequency drift of the crystal due to changes in temperature. These changes can be viewed in the schematic and PCB layout in Figure 14 and 15.

The next few changes that were made were to improve the RF characteristics of the PCB traces. The first modification made was to change the transmission line from a microstripe transmission line to a grounded coplanar waveguide (GCPW). GCPW allows the transmission line to radiate less and improve the isolation compared to microstripe transmission lines as seen in Figure 13. This and a few other key properties of GCPW permit us to achieve lower loss in our transmission line and increase our overall electromagnetic performance.

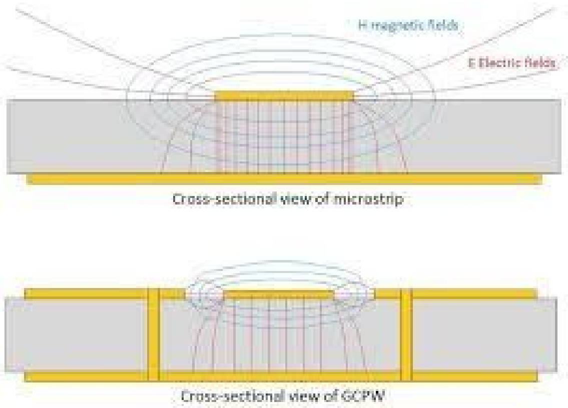

H magnetic fields
E Electric fields
Cross-sectional view of microstrip
Cross-sectional view of GCPW

Figure 13. Electromagnetic field properties of microstrip transmission line vs. GCPW

We then changed the width transitions between transmission line and component pads. As we learned, each width transition negatively impacts the RF performance and so we carefully adjusted the trace widths without severely veering away from our desired characteristic impedance of 50 **Ω**.

Finally, the last modification that was made was to the via placements. We added better via grounding around the pads for the SMA connector, reorganized the vias surrounding the transmission lines and added ground vias along the perimeter of the board to better isolate the PCB from external EM interference, as seen in Figure 15.

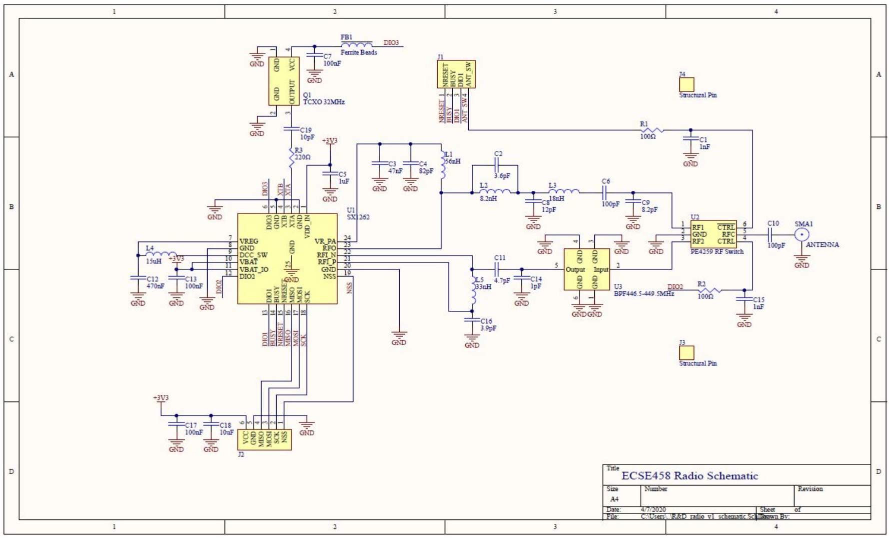


Figure 14. Improved Radio Transceiver Schematic

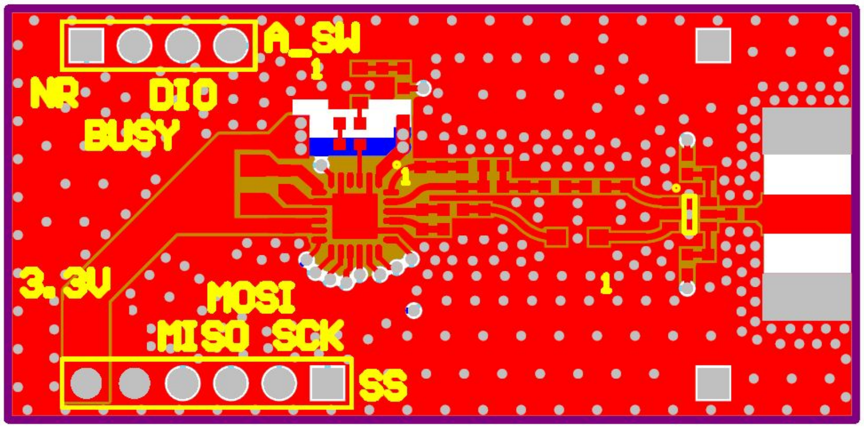


Figure 15. Top layer of improved PCB layout

With the board design complete, we were able to then manufacture the board. Due to complications with COVID-19, our initial manufacturer choice of JLCPCB was unable to produce the board, therefore we chose the US manufacturer OSHPark. However, with the closure of McGill, we were unable to assemble the PCB. When school reopens and we have access to the components and our equipment again, we hope to construct and test our designed radios. A 3D model of the fully assembled board can be seen below in Figure 16.

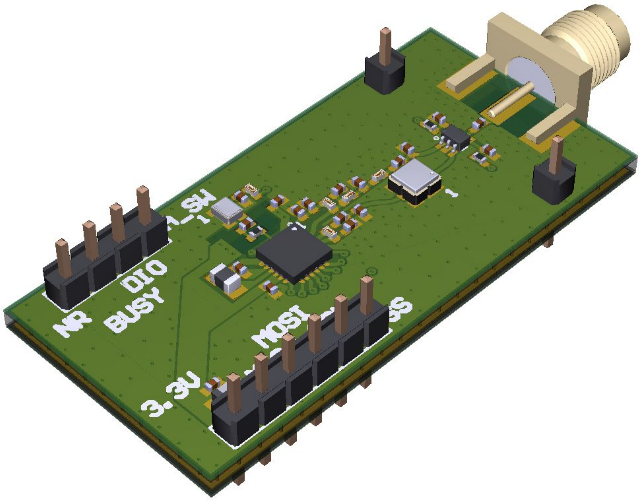


Figure 16. 3D view of the improved radio PCB

Radio Software:

In addition to modifications made to the PCB design, we developed the software scripts to enable us to operate the radio. To communicate with the radio chip, we use a series opcode (operation code) instructions and their corresponding arguments to send to the transceiver chip. The opcode is sent to the chip using the Serial Peripheral Interface (SPI) communication protocol

SPI consists of four lines: MISO, MOSI, SCK and SS. When we wish to begin sending instructions to the chip, we first need to bring the SS line low to notify the chip we want to communicate with it. Once the SS line is low and the chip is not busy, which can be checked from its BUSY line, we can send the instruction (opcode) followed by its required arguments. For the detailed series of instructions to enter the transmitter mode of the radio see Appendix D and for the receiver mode, see Appendix E.

The device can also be viewed as a state machine with 6 main operation modes. Once the chip is powered on or reset, the chip will need to enter the STDBY (standby) state. In the STDBY state, we may change the protocol (FSK or LoRa), location to write to in the buffer and many other commands. In STDBY we can transition to the Tx (transmit) state used to transmit packets, the Rx (receive) state used to receive packets, the Sleep state to conserve power and the FS (Frequency Synthesis) state used to debug the internal circuitry of the chip. These operational modes can be visualized by a finite-state machine and the actions through which each mode transitions in Figure 17.

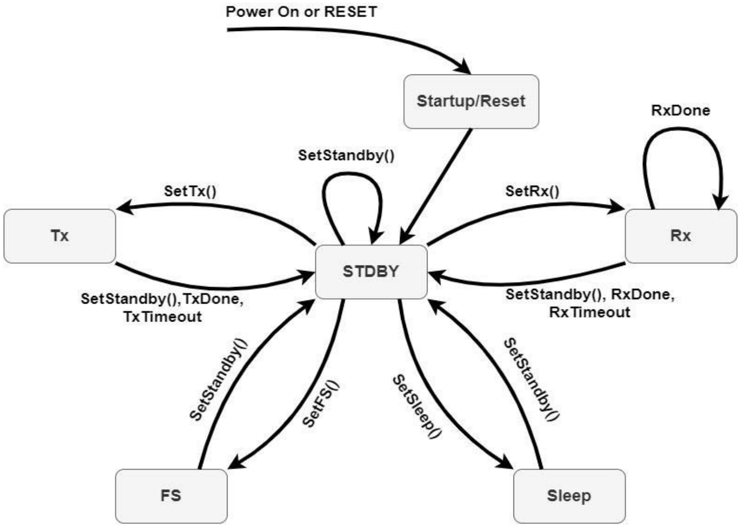


Figure 17 - FSM of Transceiver States

B. Antenna Design

1. The single antenna element

The half-wave dipole antenna was chosen to be the single antenna element due to its omni-directionality and simplicity of design. The half-wave dipole length (L) and width (W) were chosen to be as described in Equation (15).

$$L = \frac{\lambda}{2} = \frac{v}{2f} = \frac{2.998 \cdot 10^{8} m/s}{2 \times 915 \cdot 10^{6} s^{-1}} = 0.164\ m\ = 16.4\ cm \qquad (15)$$

$W\ = 0.002m\ = 2\ mm$ (using a conventional copper wire diameter value)

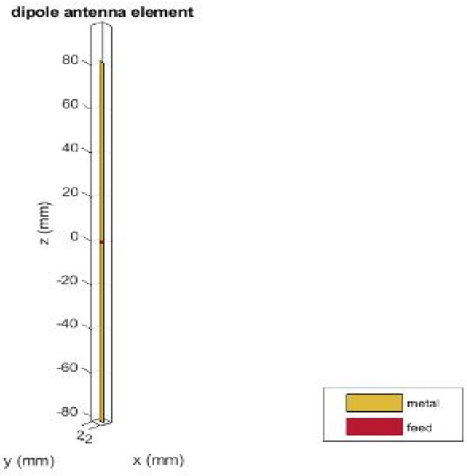


Figure 18. MATLAB dipole model.

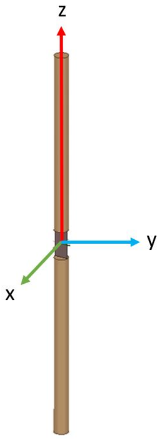


Figure 19. HFSS dipole model.

The dipole with these dimensions was first modeled and simulated in MATLAB (Figure 18) and then modeled and simulated in Ansys HFSS (Figure 19). The parametric sweep feature was used to tune the dipole for optimal performance at 915MHz. The tuned antenna dimensions are the following:

$$\begin{aligned} L &= 15.1\ cm \\ W &= 2\ mm\,. \end{aligned} \tag{16}$$

2. HFSS simulation results

The following HFSS simulation results plots show multiple curves because the optimization parametric sweep tool was used to simulate for multiple variations of the antenna's length. More precisely, the antenna was simulated with a length ranging from [15, 15.2] cm in steps of 0.5 mm. The following sections will only abstract the results from the most tuned antenna design, which is with a length of 15.1 cm.

i. S-parameter

For the operating frequency of 915 MHz, the tuned antenna has a reflection coefficient (S11) of -17.8 dB (see Figure 18), which meets the minimum requirement of -10 dB.

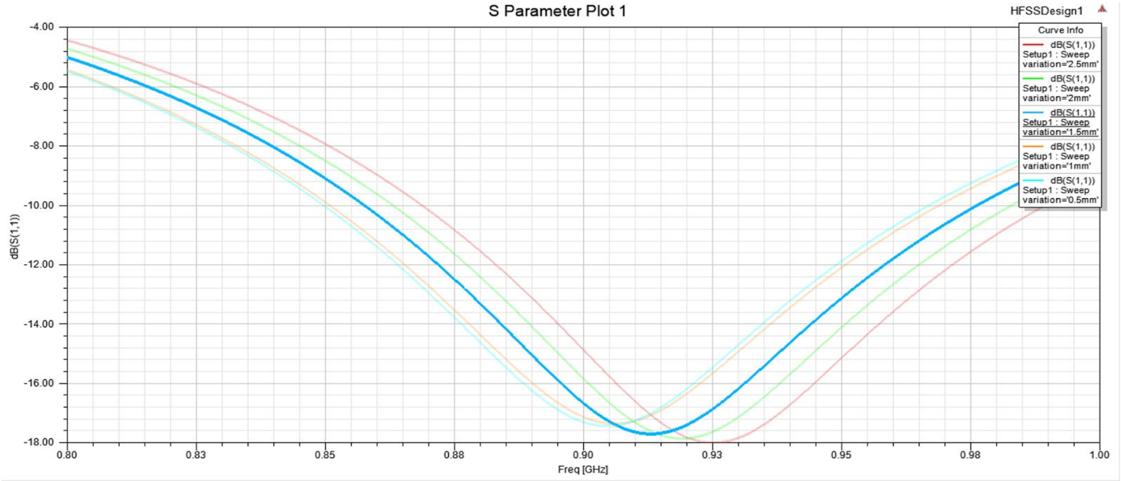


Figure 20. HFSS simulation results - S11 plot.

ii. Impedance

For the operating frequency of 915 MHz, the tuned antenna has an impedance of $Z_L = (64.5 + 3i)\,\Omega$ (see Figure 19), which is close enough to the desired characteristic impedance of $Z_0 = (50 + 0i)\,\Omega$. This impedance is evaluated as close enough since the reflection coefficient meets its requirement.

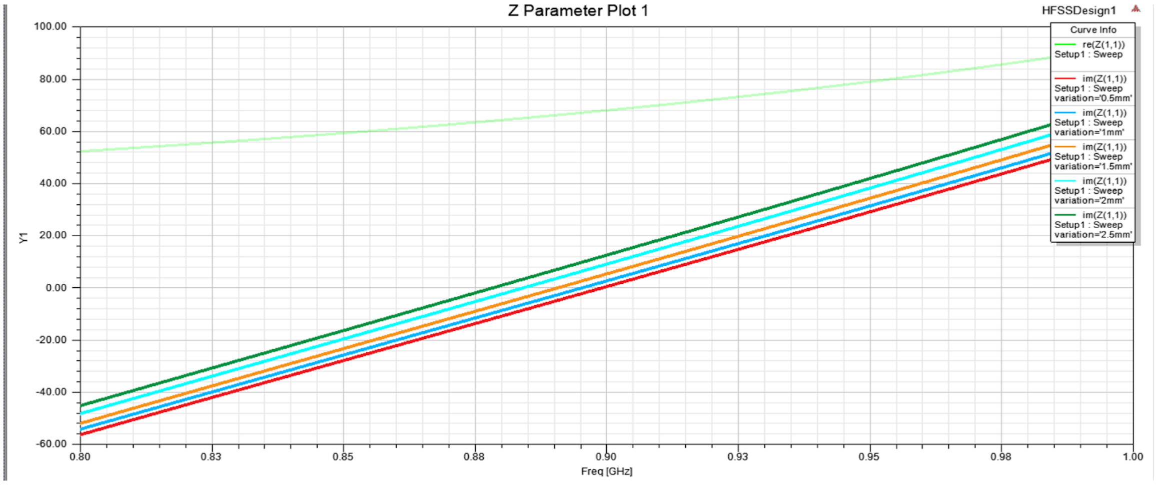
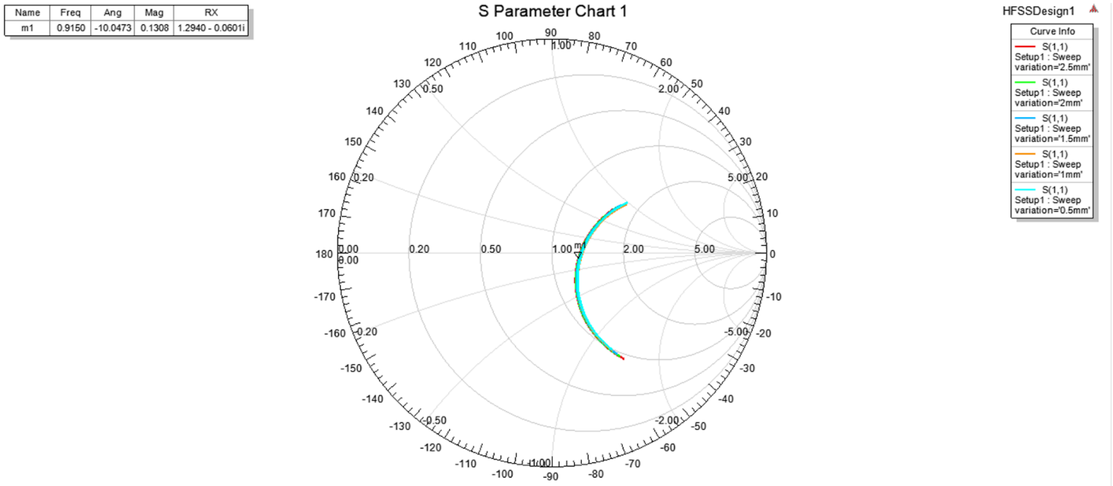


Figure 21. HFSS simulation results - Impedance plot.

iii. Smith Chart

For the operating frequency of 915 MHz, the tuned antenna has a normalized impedance of $1.3 - 0.06i$, calculated from Equation 9 and shown in Figure 22. This normalized impedance is also close enough to the ideal value of $1 + 0i$, which is determined by the reflection coefficient shown previously.

$$z_0 = \frac{Z_L}{Z_0} = \frac{(64.5 + 3i)\ \Omega}{50\ \Omega} = 1.3 - 0.06i \tag{9}$$



Name	Freq	Ang	Mag	RX
m1	0.9150	-10.0473	0.1308	1.2940 - 0.0601i

Figure 22. HFSS simulation results - Smith Chart plot.

iv. 3D Radiation pattern

For the operating frequency of 915 MHz, the tuned antenna has a toroidal radiation pattern, as expected from a half-wave dipole antenna. Peak gain is $2.9\ dBi$ (see Figure 23), which is also close to the theoretical value of $3\ dBi$.

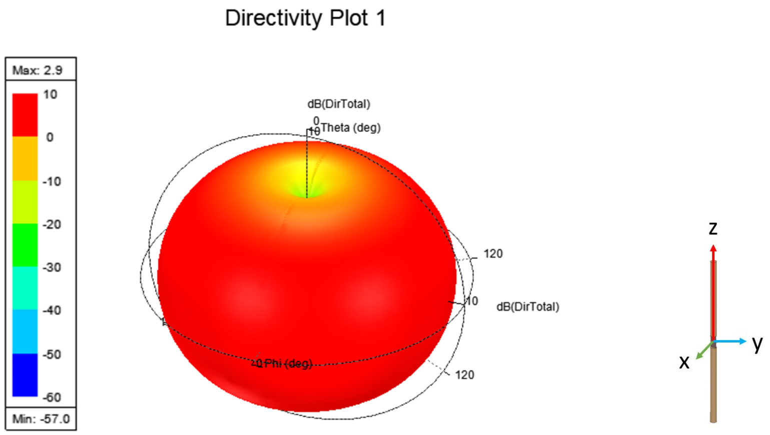
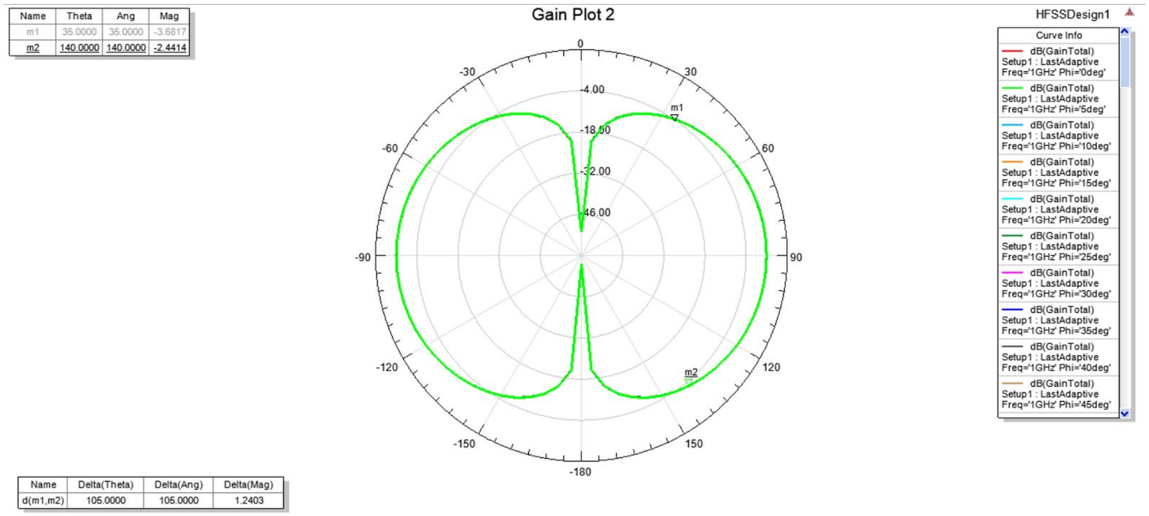


Figure 23. HFSS simulation results - 3D radiation pattern plot.

v. 2D Gain plot and beamwidth

As a cross-section of the 3D radiation pattern, the 2D gain plot shows a beamwidth of 100° (see Figure 24), which is better than the theoretical half-wave dipole beamwidth of 78°. This increase in beamwidth explains the decrease in peak gain from the previous section.



Name	Theta	Ang	Mag
m1	35.0000	35.0000	-3.6817
m2	140.0000	140.0000	-2.4414

Name	Delta(Theta)	Delta(Ang)	Delta(Mag)
d(m1,m2)	105.0000	105.0000	1.2403

Gain Plot 2
HFSSDesign1
0
-30
30
-60
60
-90
90
-120
120
-150
150
-180
-4.00
-18.00
-32.00
-46.00
m1
m2
Curve Info
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='0deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='5deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='10deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='15deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='20deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='25deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='30deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='35deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='40deg'
dB(GainTotal) Setup1 : LastAdaptive Freq='1GHz' Phi='45deg'

Figure 24. HFSS simulation results - 2D gain plot.

3. Beamforming Phased Array Design

The beamforming phased array was chosen to have a 4-element configuration, where each antenna element is a half-wave dipole of the dimensions specified by Equation (16). The dipole has been chosen for its simplicity of design, low cost, and horizontal omni-directionality as shown by the simulations in Figures 23 and 24. Moreover, we have chosen to implement the beamformer on the transmitter side, instead of the receiver side, because we will be using the rocket's on-board pressure sensor to read real-time altitude values and beam steer accordingly. This design decision is simple and realistic as the rocket knows best its current position and can therefore beam steer towards the ground much easier than the ground can scan for the rocket.

Hence, each dipole is positioned along the rocket's body tube, on the z-axis, spaced by a half-wavelength distance of 16.4 cm (see Figure 25 a)). The antenna array was chosen to be 4-element because 2-element arrays resulted in too much power lost in the sides lobes, but more than 4 elements resulted in an unsuitable narrow beamwidth. Since we will not be able to accurately align the transmitter and receiver antennas, a narrow beamwidth (< 10°) is more likely to result in a communication loss. In order to validate our design decisions, the total gain pattern of the antenna array was simulated in MATLAB (Figure 25 b)) and is shown to satisfy the performance requirements, such as horizontal omni-directionality.

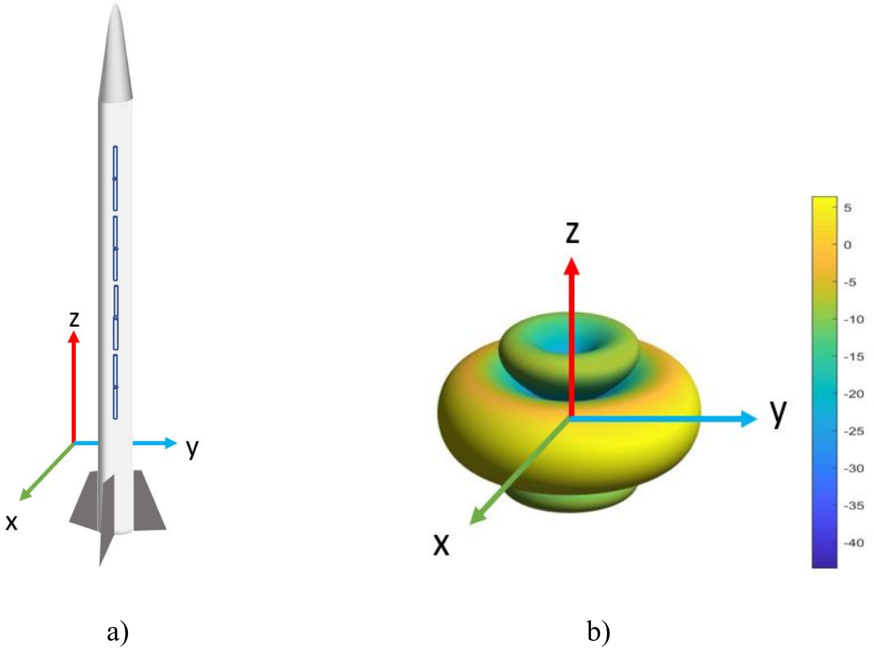


Figure 25. 4-Element phased array a) configuration on rocket b) simulated radiation pattern in MATLAB.

As described above, the beamformer will need to compute steering angles after reading real-time altitude values. The complete system is shown in Figure 26.

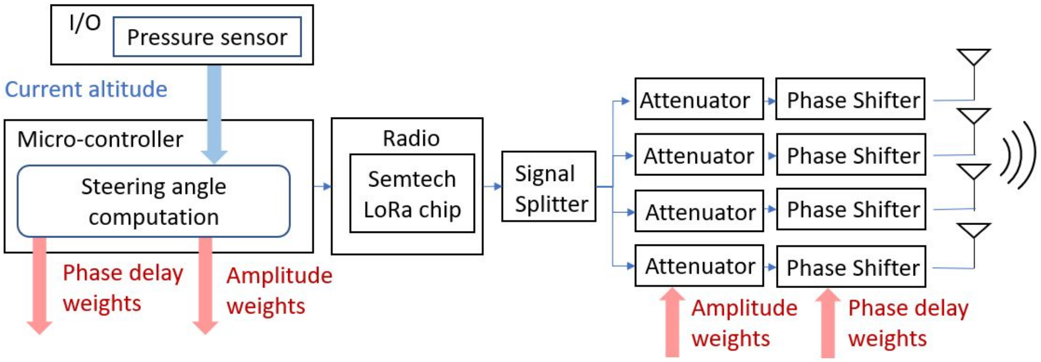
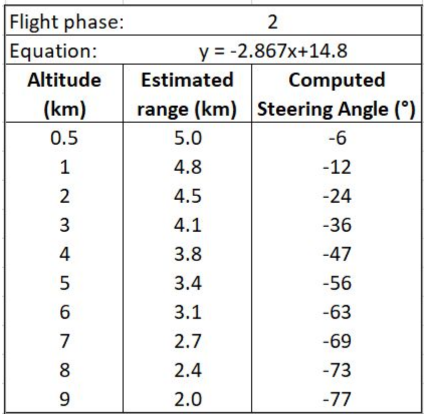


Figure 26. High-level block diagram of the beamformer transceiver system .

Starting with the pressure sensor providing altitude data to the beamformer, it will then map the altitude value to a steering angle depending on its current phase in flight (Figure 2). Since each phase has a different linear relationship between the altitude and range, the steering angles also need to be mapped with different equations. Table 2 shows an example of steering angle mapping for one of the phases.

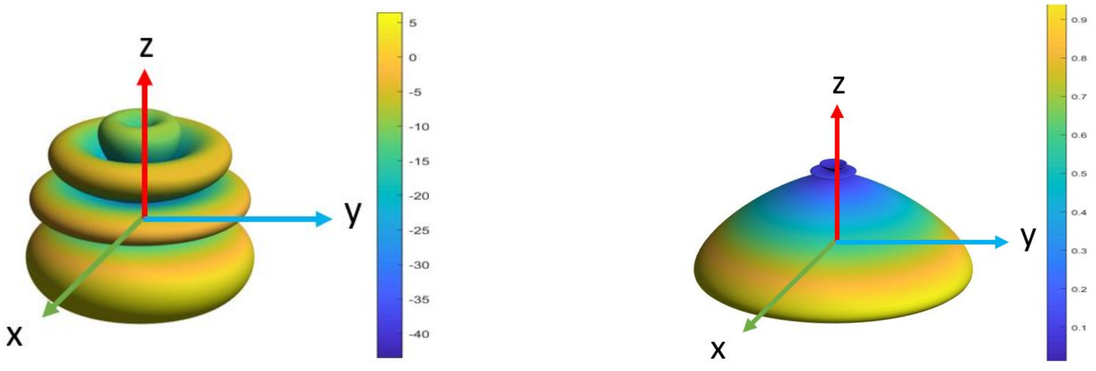


Flight phase:	2	
Equation:	y = -2.867x+14.8	
Altitude (km)	**Estimated range (km)**	**Computed Steering Angle (°)**
0.5	5.0	-6
1	4.8	-12
2	4.5	-24
3	4.1	-36
4	3.8	-47
5	3.4	-56
6	3.1	-63
7	2.7	-69
8	2.4	-73
9	2.0	-77

Table 2. Example of steering angle mapping at flight phase 2.

Once the steering angle is computed, the beamformer can apply the corresponding weights, implemented as phase delays applied by phase shifters. Amplitude weights can also be used, which would be then inputted to attenuators. As a high-level overview, the microcontroller builds the telemetry packet by reading the respective sensors and I/O peripherals needed and writes to the serial port of the radio. The radio then generates an RF signal, which goes through a signal splitter to get fed into each antenna of the 4-element array. Each of these split signals will be weighted and phase shifted by different values depending on the steering angle.

4. MATLAB Beamforming and Beam Steering Simulation



a) b)

Figure 27. 3D radiation pattern in a) dB and in b) linear scale of the 4-element array beam steered at -45° elevation angle.

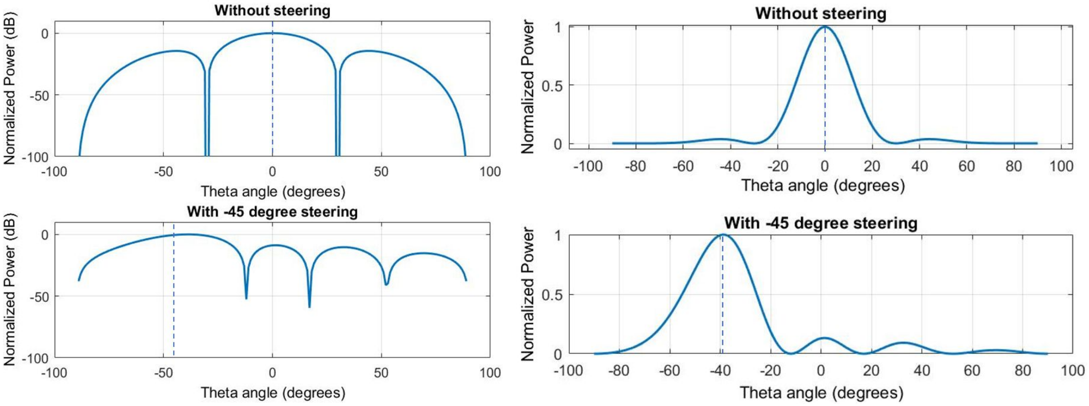


Figure 28. 2D linear radiation pattern in a) dB and in b) linear scale of the 4-element array beam steered at -45° elevation angle.

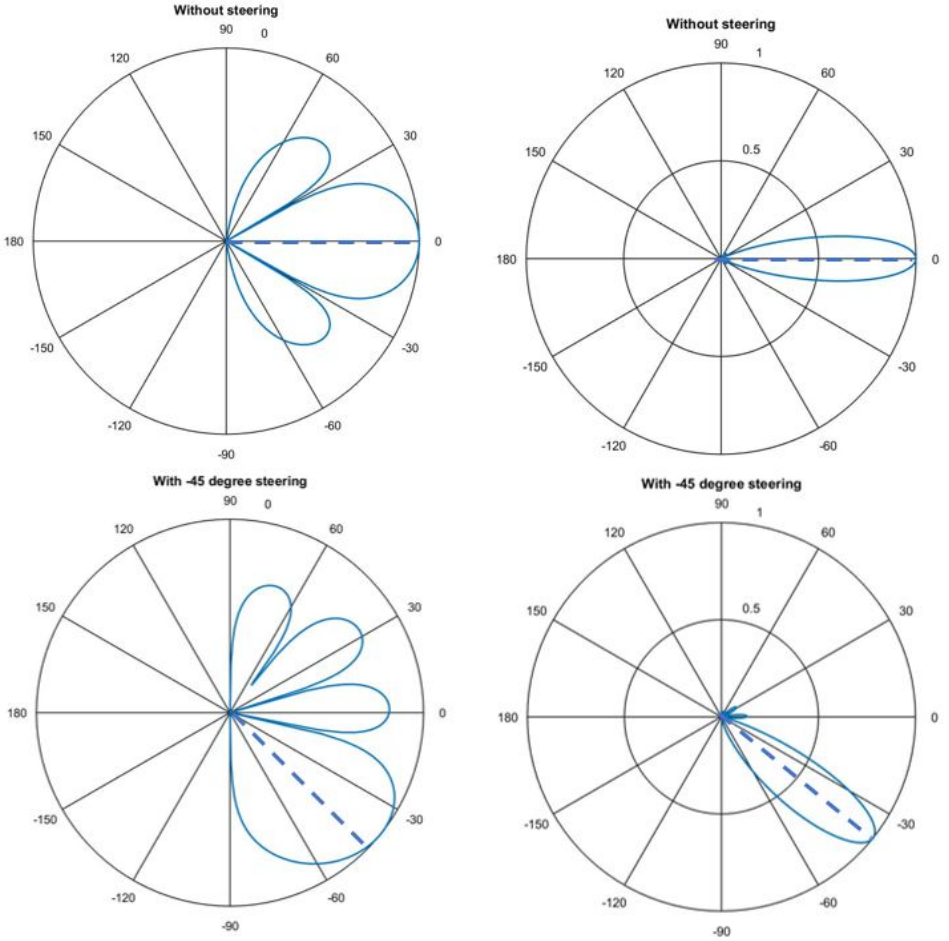


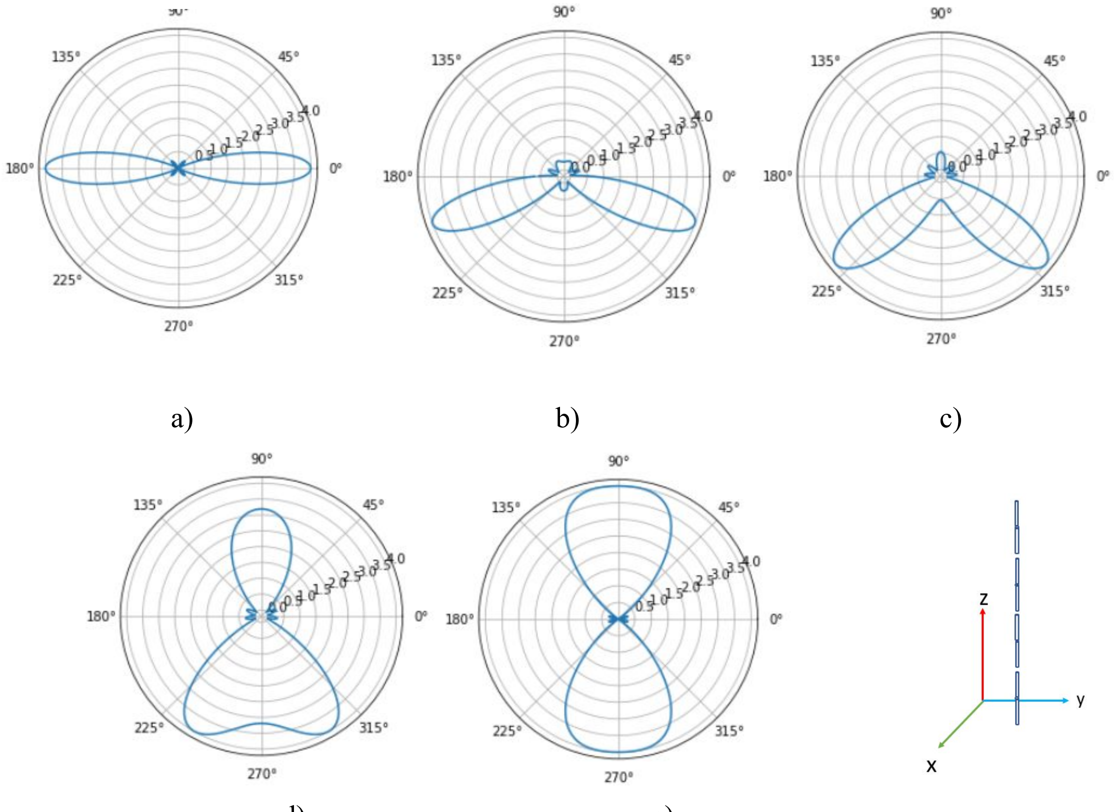
Figure 29. 2D polar radiation pattern in a) dB and in b) linear scale of the 4-element array beam steered at -45° elevation angle.

Figures 27-29 show the radiation patterns in different plotting scales. Figures 28-29 specifically show the patterns of beam steering at -45° and without beam steering. We want to steer towards the -90°

as the rocket increases in altitude and beam back to 0° as the rocket decreases in altitude (refer to Table 2 for the specific angles). The negative angle is purely depending on our initial definition of the reference frame. These simulations were achieved using the MATLAB Phased Array System Toolbox's methods *phased.SteeringVector*. From these plots, we can see that our beamformer's design would efficiently help improve the MRT's transmission. Even though there are sidelobes, which we would want to minimize as much as possible, most of the power is still concentrated at the main lobe, which has an acceptable beamwidth of approximately 20°. A beamwidth over 10° gives us some threshold for the steering angle estimation inaccuracies. The rocket will not be able to accurately predict its real-time distance from the ground-station because GPS gets locked during most of the flight since its speed exceeds the allowed limit. Depending on wind conditions, the rocket might drift at varying ranges from the ground-station. Hence, the beamformer should not have a narrow beamwidth.

5. Implementing the Beamforming Simulation with Python

As a learning exercise, we have decided to implement our own beamforming simulator in Python to compare with the simulation results from MATLAB. This was because we could not see the MATLAB source file of the Phased Array Toolbox library. The complete code is in Appendix F. The algorithm was implemented following a naive approach to directly translate from Equation (14). Hence, it is not very computationally efficient, though is enough for plotting in 2 dimensions. More efficient algorithms could be explored in the future to improve the current $O(N^2)$ runtime.



a)

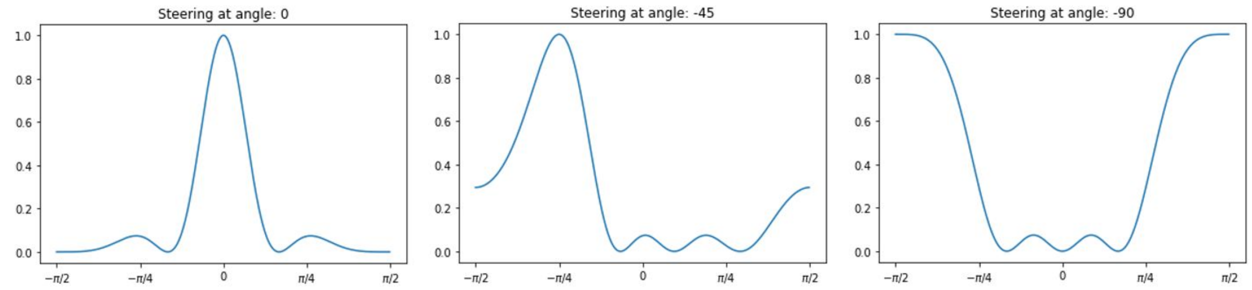
b)

c)

d)

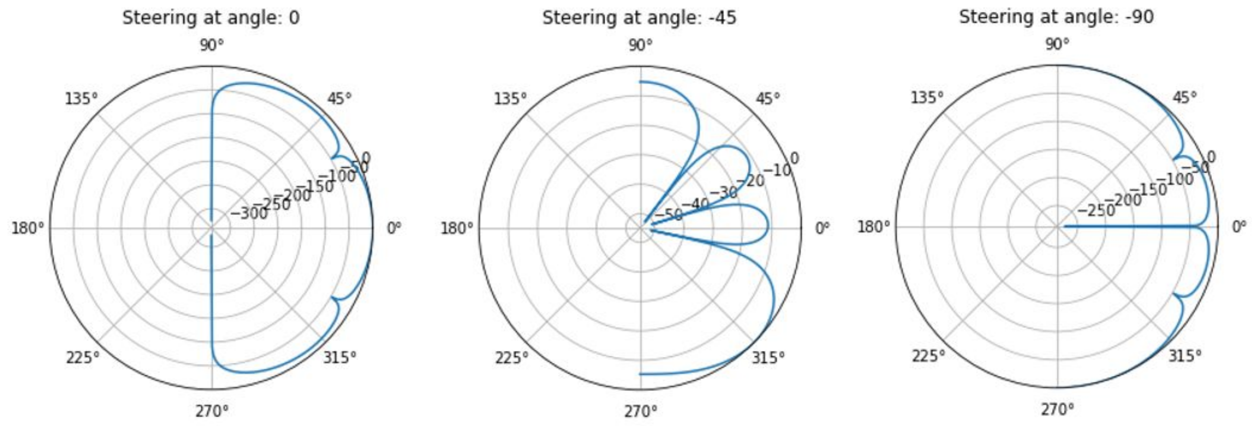
e)

Figure 30. 2D polar radiation pattern in the linear scale of the 4-element array, beam steering at a) 0°, b) -20°, c) -40°, d) -60°, e) -90°



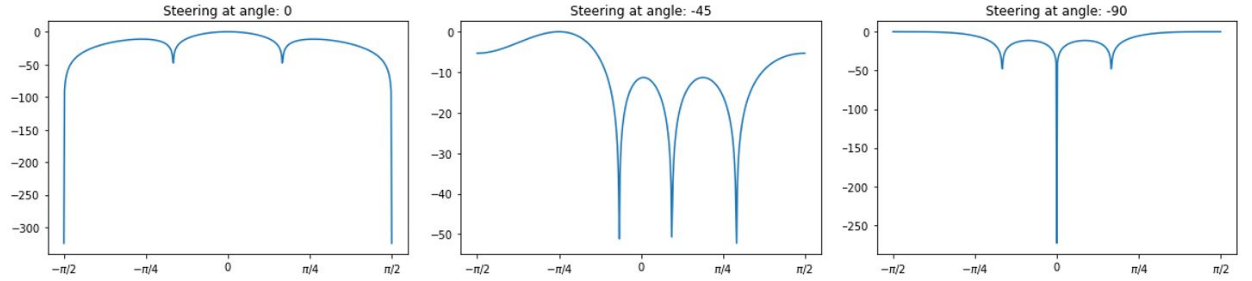
a) b) c)

Figure 31. 2D linear radiation pattern in a linear scale of the 4-element array steered at a) 0°, b) -45°, c) -90°



a) b) c)

Figure 32. 2D polar radiation pattern in a dB scale of the 4-element array steered at a) 0°, b) -45°, c) -90°



a) b) c)

Figure 33. 2D linear radiation pattern in a dB scale of the 4-element array steered at a) 0°, b) -45°, c) -90°

The linear power scaled plots on Figures 30-31 were the most similar to the MATLAB plots. The dB power scaled plots had some large negative number at a few points that offset the scale of the plot. These inconsistencies might be due to the small number of data points, or numerical computation errors due to data imprecisions, etc. However, the general behavior is still consistent with the MATLAB simulation results.

V. Impact on Society and the Environment

The design of our communication system so far mostly involves software applications, so there is not a large impact on non-renewable resources. However for the manufacturing of our radios and antennas, we require the use of PCBs, which does have an impact on non-renewable resources such as copper, lead and other metals used in the manufacturing process, chemical compounds for baths and resistive materials, and the disposal of toxic material used in the manufacturing process. We will also be using various SMD components on our PCBs, that all require their own resources to produce.

As far as rocketry, it is not considered very beneficial to our environment, due to the burning of the solid rocket fuel. However, one benefit with our project is the significant decrease in power consumption for our communication system compared to previous years systems. With our transceiver chip and more efficient antenna, we require a much lower power consumption for the same return in communication ability. As well, with a working communication link during the rocket's flight, we would be able to recover and reuse a significant portion of the sounding rocket for following launches.

As rocketry is quite dangerous and can be used in harmful manners potential users may use the product for something mischievous. As well, with the relatively small market for sounding rocket communication systems, we don't expect many people to use our product but, there may be a few hobbyists that are interested in this product which could help them in their sounding rockets.

VI. Report on Teamwork

For the entirety of the semester, we organized two weekly meetings to discuss our progress and any design decisions. The first weekly meeting with our project supervisor, Prof. Benoit Champagne, was used to guide us on the correct path by discussing the more complicated aspects of the project, such as the theory behind beamforming and chirp spread spectrum. These meetings we also used to validate our design decision by allowing another perspective to analyze our choices in this project. The second weekly meeting only included us and was used primarily to decide on system decisions that affect both of us, such as the operating frequency and the type of modulation used by the radios. With only two of us working on this capstone project, the team dynamic between us is very easy to maintain. Thus, no difficulties have been faced regarding teamwork on this project. We found that an open and strong line of communication between us allows for good collaboration.

The contribution of work on this project has been evenly worked on by both of us and we have both added much progress to the overall end goal of the transmission system. With the workload being separated into the radio section and antenna section, confusing over the work has not been a problem

VII. Conclusion

With our capstone project coming to a close, we have been fortunate to progress significantly to our end goal of a long-range transmission system. Regarding the radio portion of the system, we manufactured our revised PCB design and were able to obtain all of our necessary components to assemble the radio. Our proposed radio will come with a narrow band bandpass filter to attenuate unwanted frequency components, and a TCXO to compensate for the frequency drift caused by the extreme heat inside the rocket. With LoRa modulation, the transmitted signal will have a high resilience to the doppler effect and allow for a very low receiver sensitivity, ensuring a large link budget in case of unexpected interfering signals. Finally, we were able to complete various software programs to operate the transceiver chip as both a transmitter and a receiver to allow us to communicate between the radios.

For the antenna side of the system, we have been able to successfully design the single half-wave dipole antenna in HFSS, simulate some beamforming antenna array with beam steering capability in MATLAB, and implement our own beamforming simulation in Python to confirm with the MATLAB results. Since the beamforming part is a theoretical design project, we were only able to confirm its success through simulations and not through actual testing.

With much of our progress being delayed due to COVID-19 we hope to continue with our development of the system once the confinements have been lifted. As for the radio design we wish to solder on the components with the MRT's reflow oven to allow us to test our PCB design. After testing we hope to continue creating new iterations of an improved radio design, integration with the beam steering hardware for the antenna and creating localization techniques based on RSSI values from the receiver.

In conclusion, we have made significant progress towards our overall goal of a long-range transmission system. We have both learned a great deal of knowledge on radio and antenna designing and were able to apply theory learned in class for real world applications.

References

[1] Afar Communications, Inc. Santa Barbara, California, USA. [Online]. http://afar.net/fresnel-zone-calculator/

[2] LoRa Modulation Basics, Revision 2, May 2015, Semtech Corporation.
Retrieved from:
(https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001OJu/xvKUc5w9yjG1q5Pb2IIkpolW54YYqGb.frOZ7HQBcRc)

[3] Balanis, C. A. (2016). Antenna Theory - Analysis and Design (4th Edition). In: John Wiley & Sons.

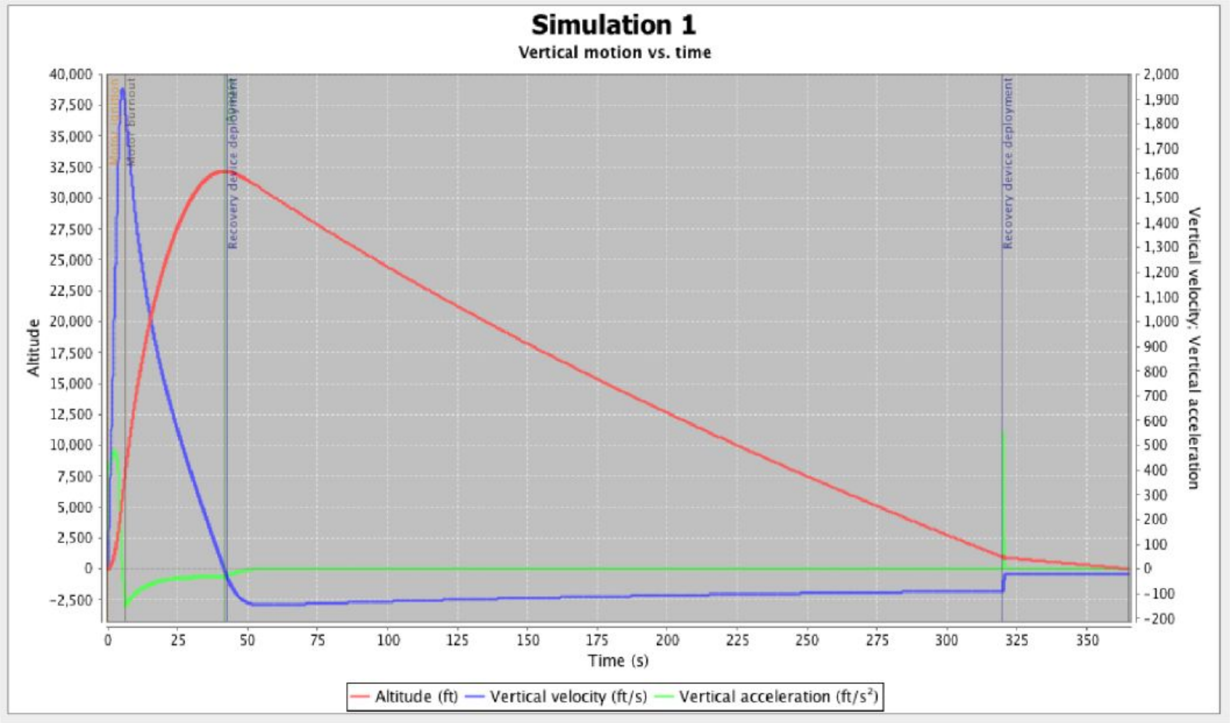
[4] Chen, W.-K. (2005). Antenna Elements. In *The Electrical Engineering Handbook* (pp. 569-578). Retrieved from https://doi.org/10.1016/B978-012170960-0/50043-8

[5] Liberti, J. C., & Rappaport, T. S. (1999). *Smart antennas for wireless communications : IS-95 and third generation CDMA applications*. Upper Saddle River, NJ: Prentice Hall PTR.

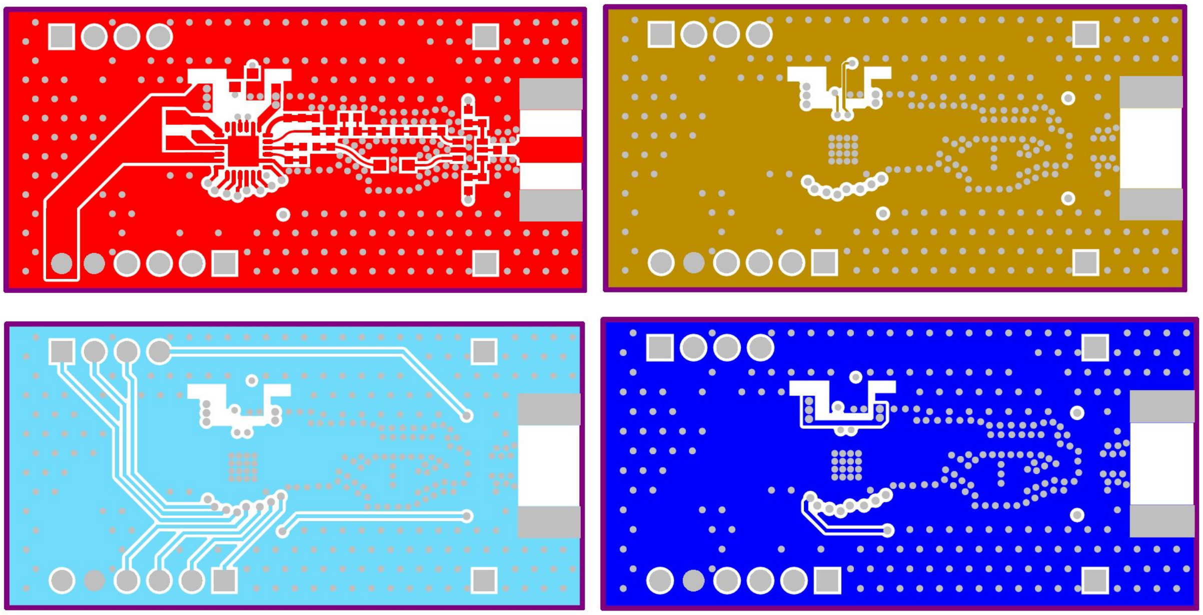
[6] SX1262 Datasheet, Revision 1.2, June 2019, Semtech Corporation.
Retrieved from:
(https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000HT7B/4cQ1B3JG0iKRo9DGRkjVuxclfwB.3tfSUcGr.S_dPd4)

Appendices

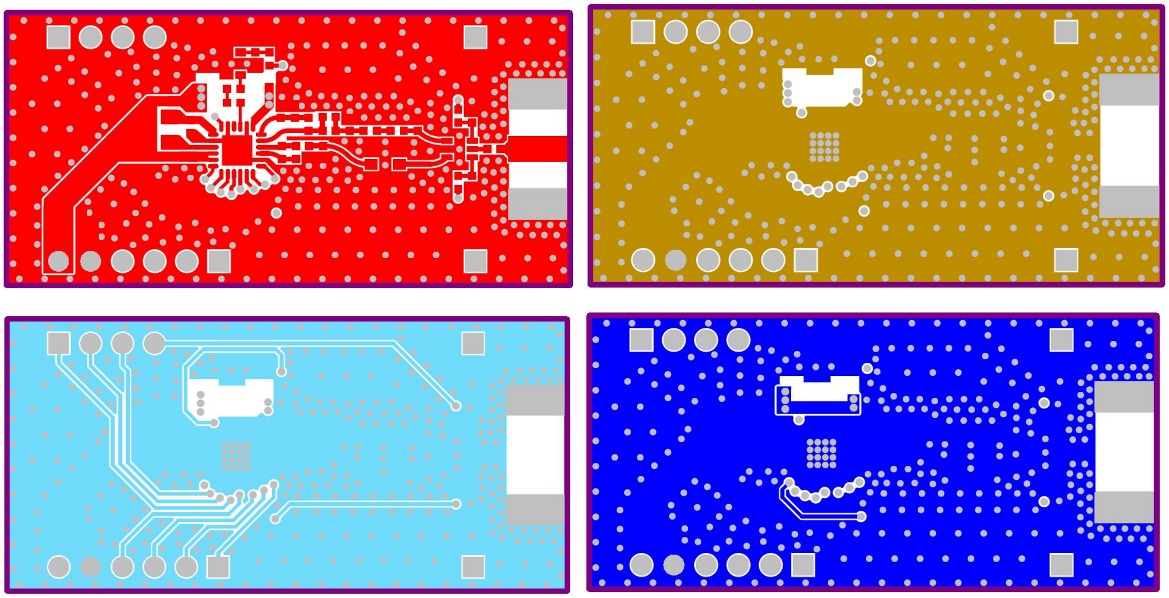
Appendix A - OpenRocket Simulation of Flight Dynamics



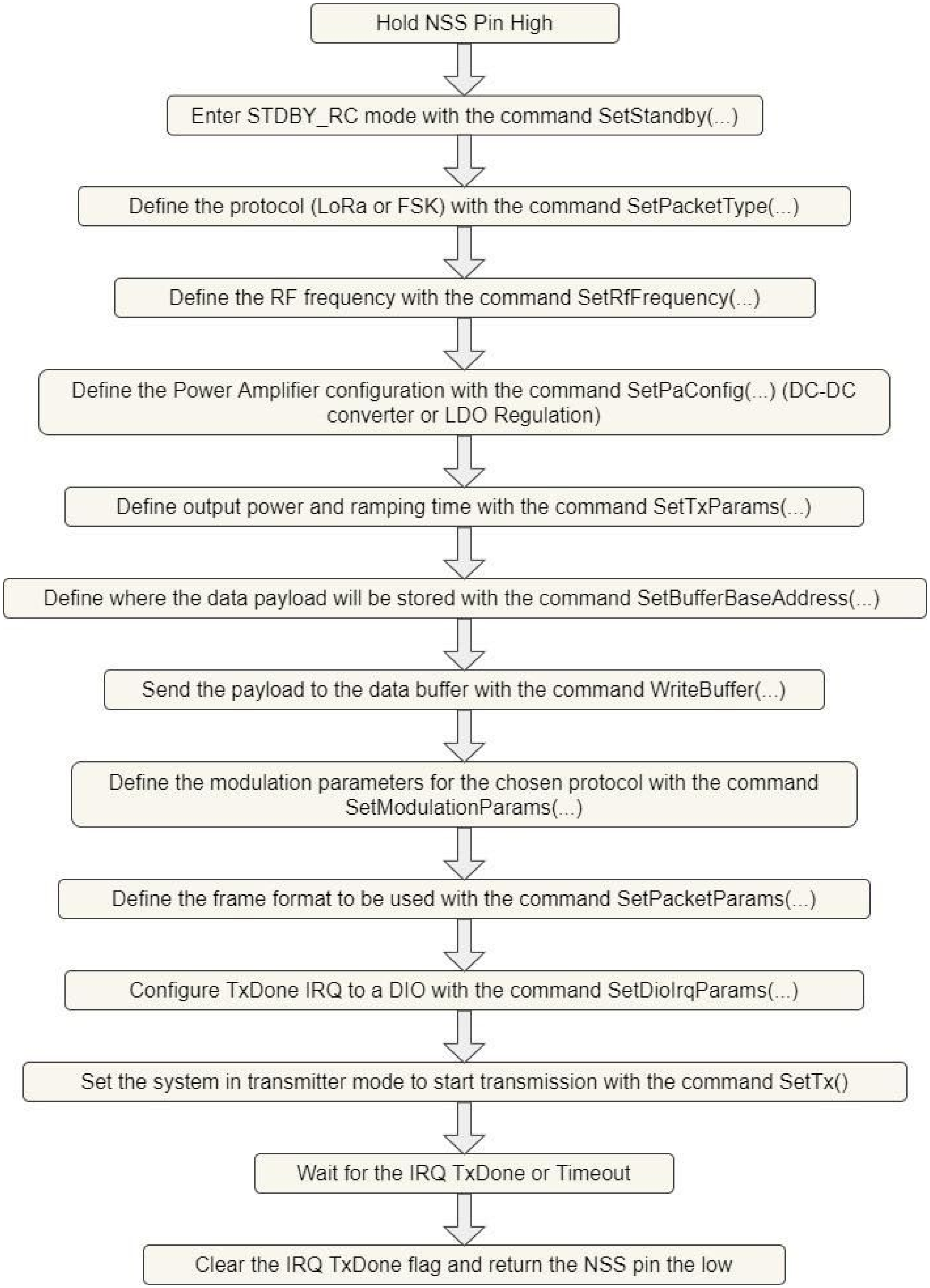
Appendix B - All Layers of Old PCB Layout



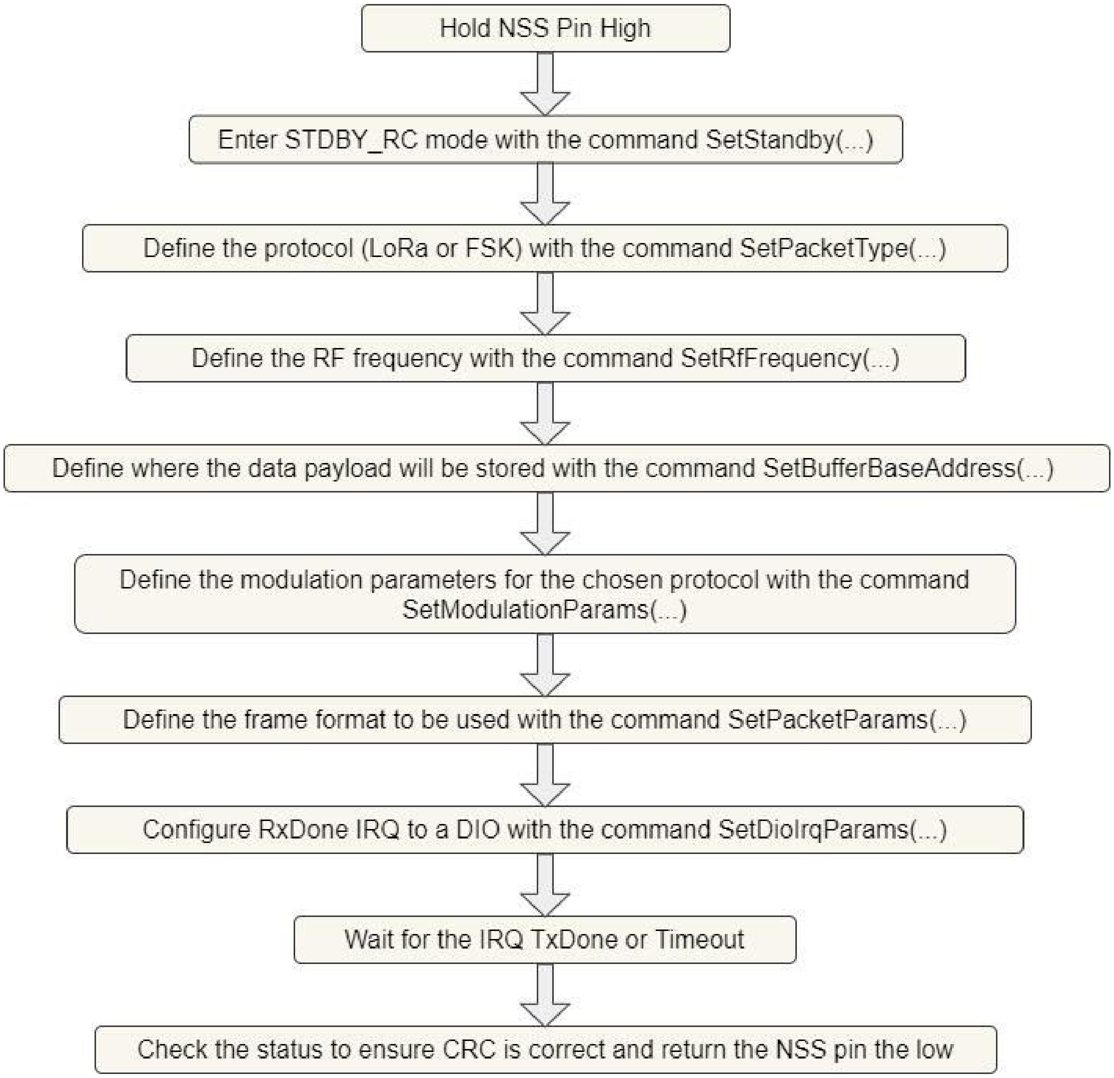
Appendix C - All Layers of the Revised PCB Layout



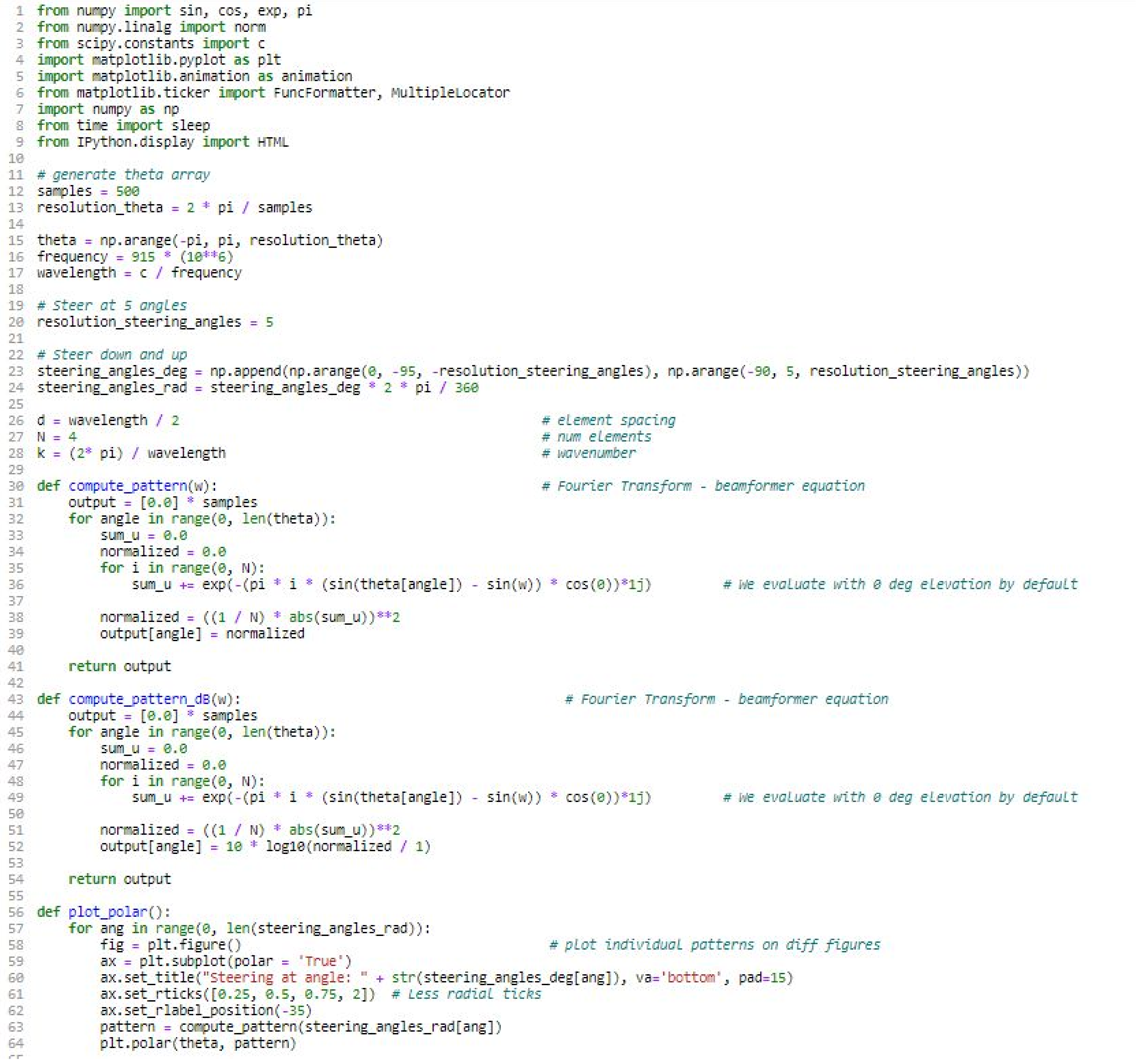
Appendix D - Flowchart to Configure Transmitter Operation


Hold NSS Pin High
Enter STDBY_RC mode with the command SetStandby(...)
Define the protocol (LoRa or FSK) with the command SetPacketType(...)
Define the RF frequency with the command SetRfFrequency(...)
Define the Power Amplifier configuration with the command SetPaConfig(...) (DC-DC converter or LDO Regulation)
Define output power and ramping time with the command SetTxParams(...)
Define where the data payload will be stored with the command SetBufferBaseAddress(...)
Send the payload to the data buffer with the command WriteBuffer(...)
Define the modulation parameters for the chosen protocol with the command SetModulationParams(...)
Define the frame format to be used with the command SetPacketParams(...)
Configure TxDone IRQ to a DIO with the command SetDioIrqParams(...)
Set the system in transmitter mode to start transmission with the command SetTx()
Wait for the IRQ TxDone or Timeout
Clear the IRQ TxDone flag and return the NSS pin the low

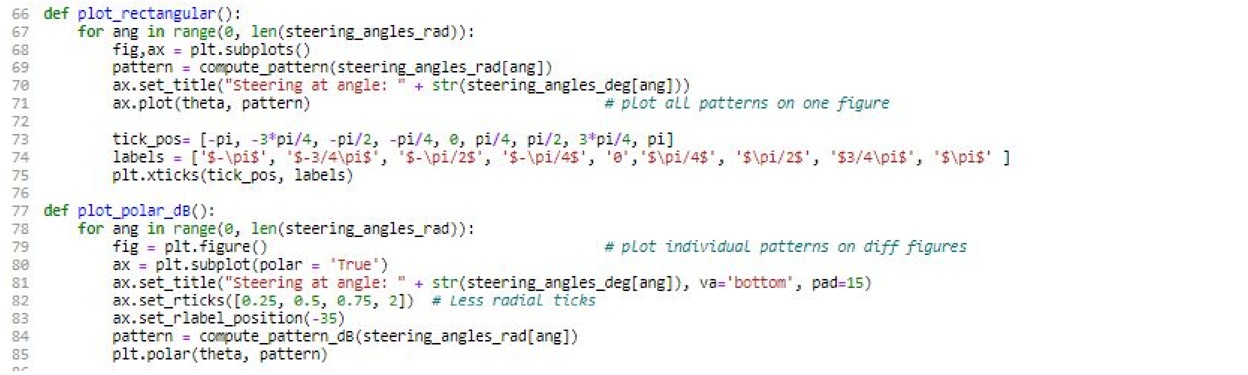
Appendix E - Flowchart to Configure Receiver Operation


Hold NSS Pin High
Enter STDBY_RC mode with the command SetStandby(...)
Define the protocol (LoRa or FSK) with the command SetPacketType(...)
Define the RF frequency with the command SetRfFrequency(...)
Define where the data payload will be stored with the command SetBufferBaseAddress(...)
Define the modulation parameters for the chosen protocol with the command SetModulationParams(...)
Define the frame format to be used with the command SetPacketParams(...)
Configure RxDone IRQ to a DIO with the command SetDioIrqParams(...)
Wait for the IRQ TxDone or Timeout
Check the status to ensure CRC is correct and return the NSS pin the low

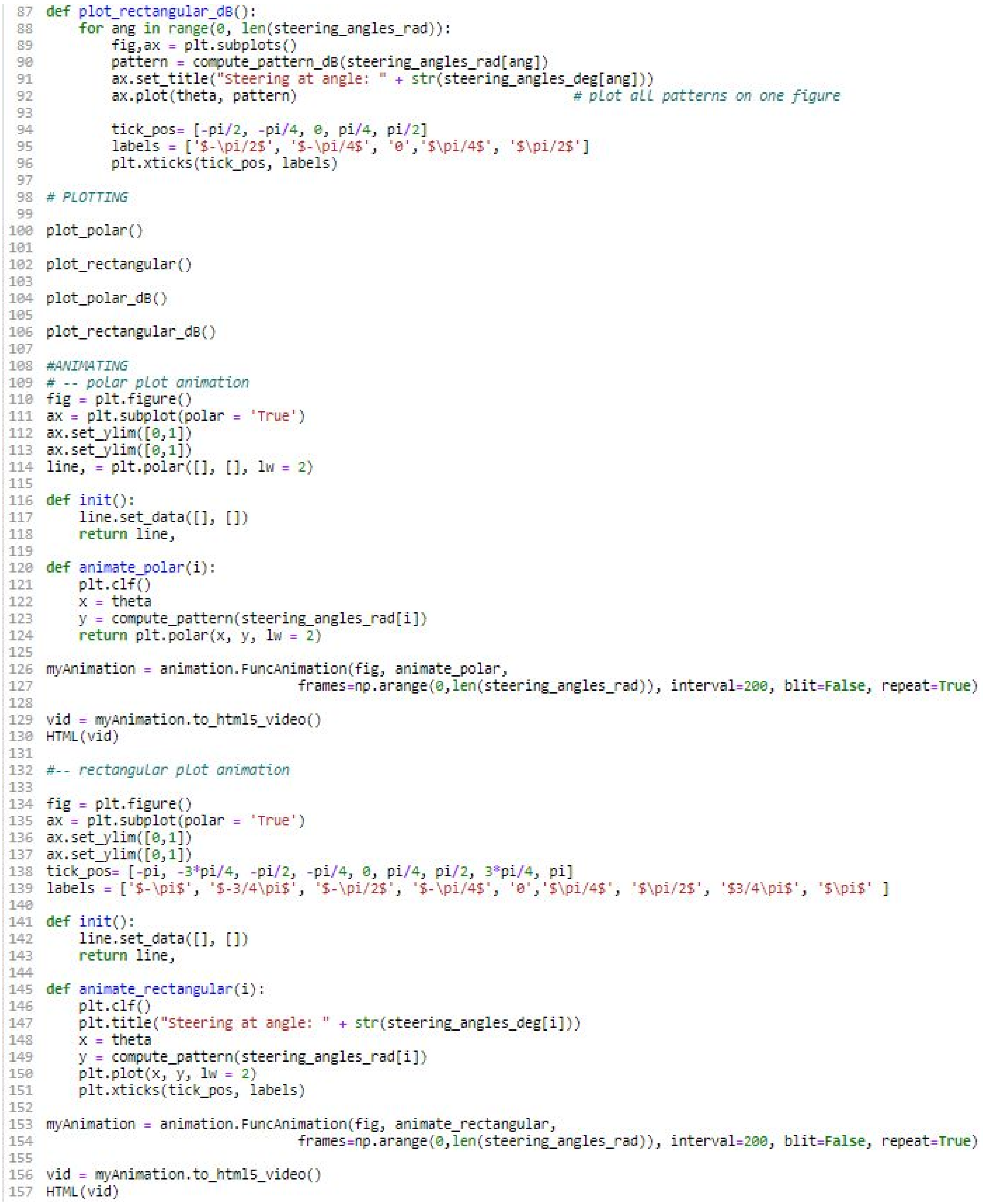
Appendix F - Beamforming Simulation Algorithm in Python



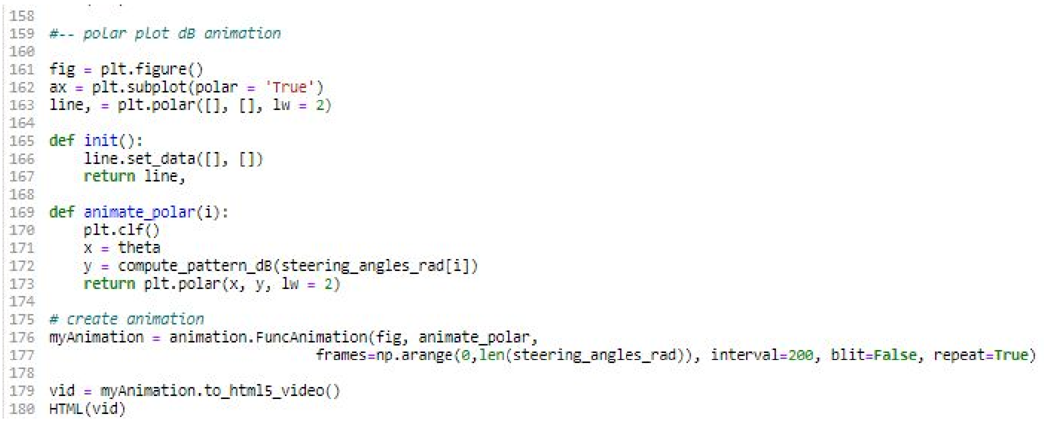
```python
from numpy import sin, cos, exp, pi
from numpy.linalg import norm
from scipy.constants import c
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from matplotlib.ticker import FuncFormatter, MultipleLocator
import numpy as np
from time import sleep
from IPython.display import HTML

# generate theta array
samples = 500
resolution_theta = 2 * pi / samples

theta = np.arange(-pi, pi, resolution_theta)
frequency = 915 * (10**6)
wavelength = c / frequency

# Steer at 5 angles
resolution_steering_angles = 5

# Steer down and up
steering_angles_deg = np.append(np.arange(0, -95, -resolution_steering_angles), np.arange(-90, 5, resolution_steering_angles))
steering_angles_rad = steering_angles_deg * 2 * pi / 360

d = wavelength / 2                                              # element spacing
N = 4                                                           # num elements
k = (2* pi) / wavelength                                        # wavenumber

def compute_pattern(w):                                         # Fourier Transform - beamformer equation
    output = [0.0] * samples
    for angle in range(0, len(theta)):
        sum_u = 0.0
        normalized = 0.0
        for i in range(0, N):
            sum_u += exp(-(pi * i * (sin(theta[angle]) - sin(w)) * cos(0))*1j)         # We evaluate with 0 deg elevation by default

        normalized = ((1 / N) * abs(sum_u))**2
        output[angle] = normalized

    return output

def compute_pattern_dB(w):                                         # Fourier Transform - beamformer equation
    output = [0.0] * samples
    for angle in range(0, len(theta)):
        sum_u = 0.0
        normalized = 0.0
        for i in range(0, N):
            sum_u += exp(-(pi * i * (sin(theta[angle]) - sin(w)) * cos(0))*1j)         # We evaluate with 0 deg elevation by default

        normalized = ((1 / N) * abs(sum_u))**2
        output[angle] = 10 * log10(normalized / 1)

    return output

def plot_polar():
    for ang in range(0, len(steering_angles_rad)):
        fig = plt.figure()                                       # plot individual patterns on diff figures
        ax = plt.subplot(polar = 'True')
        ax.set_title("Steering at angle: " + str(steering_angles_deg[ang]), va='bottom', pad=15)
        ax.set_rticks([0.25, 0.5, 0.75, 2])  # Less radial ticks
        ax.set_rlabel_position(-35)
        pattern = compute_pattern(steering_angles_rad[ang])
        plt.polar(theta, pattern)
```



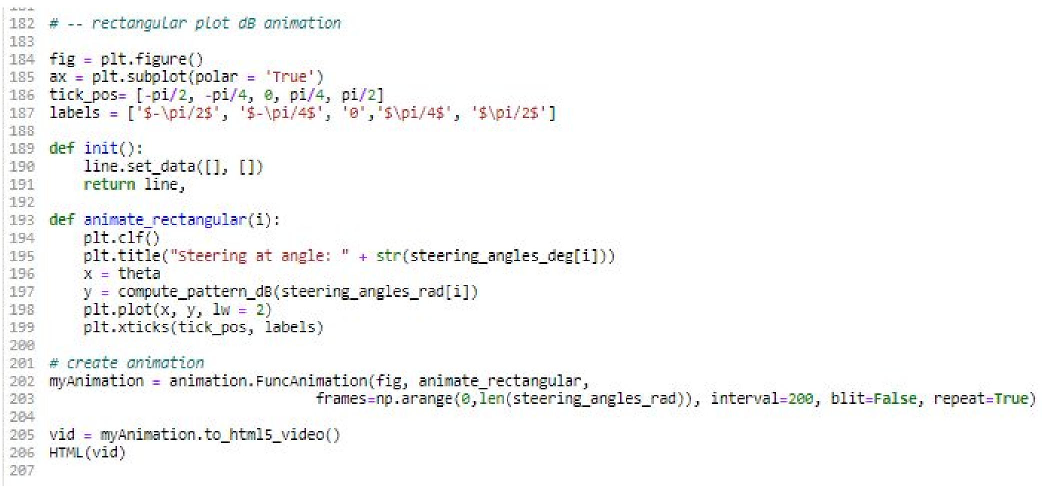
```python
def plot_rectangular():
    for ang in range(0, len(steering_angles_rad)):
        fig,ax = plt.subplots()
        pattern = compute_pattern(steering_angles_rad[ang])
        ax.set_title("Steering at angle: " + str(steering_angles_deg[ang]))
        ax.plot(theta, pattern)                                  # plot all patterns on one figure

        tick_pos= [-pi, -3*pi/4, -pi/2, -pi/4, 0, pi/4, pi/2, 3*pi/4, pi]
        labels = ['$-\pi$', '$-3/4\pi$', '$-\pi/2$', '$-\pi/4$', '0','$\pi/4$', '$\pi/2$', '$3/4\pi$', '$\pi$' ]
        plt.xticks(tick_pos, labels)

def plot_polar_dB():
    for ang in range(0, len(steering_angles_rad)):
        fig = plt.figure()                                       # plot individual patterns on diff figures
        ax = plt.subplot(polar = 'True')
        ax.set_title("Steering at angle: " + str(steering_angles_deg[ang]), va='bottom', pad=15)
        ax.set_rticks([0.25, 0.5, 0.75, 2])  # Less radial ticks
        ax.set_rlabel_position(-35)
        pattern = compute_pattern_dB(steering_angles_rad[ang])
        plt.polar(theta, pattern)
```



```python
def plot_rectangular_dB():
    for ang in range(0, len(steering_angles_rad)):
        fig,ax = plt.subplots()
        pattern = compute_pattern_dB(steering_angles_rad[ang])
        ax.set_title("Steering at angle: " + str(steering_angles_deg[ang]))
        ax.plot(theta, pattern)                                 # plot all patterns on one figure

        tick_pos= [-pi/2, -pi/4, 0, pi/4, pi/2]
        labels = ['$-\pi/2$', '$-\pi/4$', '0','$\pi/4$', '$\pi/2$']
        plt.xticks(tick_pos, labels)

# PLOTTING

plot_polar()

plot_rectangular()

plot_polar_dB()

plot_rectangular_dB()

#ANIMATING
# -- polar plot animation
fig = plt.figure()
ax = plt.subplot(polar = 'True')
ax.set_ylim([0,1])
ax.set_ylim([0,1])
line, = plt.polar([], [], lw = 2)

def init():
    line.set_data([], [])
    return line,

def animate_polar(i):
    plt.clf()
    x = theta
    y = compute_pattern(steering_angles_rad[i])
    return plt.polar(x, y, lw = 2)

myAnimation = animation.FuncAnimation(fig, animate_polar,
                               frames=np.arange(0,len(steering_angles_rad)), interval=200, blit=False, repeat=True)

vid = myAnimation.to_html5_video()
HTML(vid)

#-- rectangular plot animation

fig = plt.figure()
ax = plt.subplot(polar = 'True')
ax.set_ylim([0,1])
ax.set_ylim([0,1])
tick_pos= [-pi, -3*pi/4, -pi/2, -pi/4, 0, pi/4, pi/2, 3*pi/4, pi]
labels = ['$-\pi$', '$-3/4\pi$', '$-\pi/2$', '$-\pi/4$', '0','$\pi/4$', '$\pi/2$', '$3/4\pi$', '$\pi$' ]

def init():
    line.set_data([], [])
    return line,

def animate_rectangular(i):
    plt.clf()
    plt.title("Steering at angle: " + str(steering_angles_deg[i]))
    x = theta
    y = compute_pattern(steering_angles_rad[i])
    plt.plot(x, y, lw = 2)
    plt.xticks(tick_pos, labels)

myAnimation = animation.FuncAnimation(fig, animate_rectangular,
                               frames=np.arange(0,len(steering_angles_rad)), interval=200, blit=False, repeat=True)

vid = myAnimation.to_html5_video()
HTML(vid)
```



```python

#-- polar plot dB animation

fig = plt.figure()
ax = plt.subplot(polar = 'True')
line, = plt.polar([], [], lw = 2)

def init():
    line.set_data([], [])
    return line,

def animate_polar(i):
    plt.clf()
    x = theta
    y = compute_pattern_dB(steering_angles_rad[i])
    return plt.polar(x, y, lw = 2)

# create animation
myAnimation = animation.FuncAnimation(fig, animate_polar,
                               frames=np.arange(0,len(steering_angles_rad)), interval=200, blit=False, repeat=True)

vid = myAnimation.to_html5_video()
HTML(vid)
```



```python
# -- rectangular plot dB animation

fig = plt.figure()
ax = plt.subplot(polar = 'True')
tick_pos= [-pi/2, -pi/4, 0, pi/4, pi/2]
labels = ['$-\pi/2$', '$-\pi/4$', '0','$\pi/4$', '$\pi/2$']

def init():
    line.set_data([], [])
    return line,

def animate_rectangular(i):
    plt.clf()
    plt.title("Steering at angle: " + str(steering_angles_deg[i]))
    x = theta
    y = compute_pattern_dB(steering_angles_rad[i])
    plt.plot(x, y, lw = 2)
    plt.xticks(tick_pos, labels)

# create animation
myAnimation = animation.FuncAnimation(fig, animate_rectangular,
                               frames=np.arange(0,len(steering_angles_rad)), interval=200, blit=False, repeat=True)

vid = myAnimation.to_html5_video()
HTML(vid)
```