NOAA GOES-R Satellite Receiver

Dr. Nathan Neihart
Nick Butts - Software Development
Yong Lim - Software Development
Jonathan Massner - Systems and RF Design
Theodore Mathews IV - RF Design
Riley Stuart - ADC Design
Jordan Tillotson - ADC Design

Team Email and Website:
sdmay20-03@iastate.edu
sdmay20-03.sd.ece.iastate.edu

April 26, 2020

Executive Summary

Engineering Standards and Design Practices Used

- Agile Development
- IEEE 145-2013: IEEE Standard for Definitions of Terms for Antennas
- IEEE 149-1979: IEEE Standard Test Procedures for Antennas
- IEEE Standard 211-2018: IEEE Standard Definitions of Terms for Radio Wave Propagation

Summary of Requirements

Hardware

- **Antenna:** WiFi grid antenna with a design frequency of 2.4 GHz.
  - Size Requirement: 1.1 meters in diameter
  - Frequency: 1694.1 MHz. The antenna will be modified for this frequency.
  - Connector: Female N
  - Polarization: Vertically Polarized
  - Aiming Accuracy: ±5° on all axis
- **Filtering & Amplification:** Following the Antenna, the signal needs to be filtered and amplified.
  - Gain: > 30 dB
  - Bandwidth: 2MHz (Signal is 1.205MHz)

Environmental & Physical

- **Line of Sight:**
  - The antenna needs a clear line of sight to the satellite, limiting possible setup locations.
- **Semi-Portable:**
  - Unable to permanently mount the antenna on campus due to cost and planning requirements
  - Long reception times complicate testing
- **Clouds & Weather:**
  - Cloud cover should not impact reception.
  - Equipment will need to be weather resistant if a permanent installation is desired.
- **Budget:**
  - $\approx$ $1,000
- **Components**
  - Use COTS (Commercial off-the-shelf) where possible for larger components (antenna, preamps, computers).
  - Radio Frequency (RF) and Analog-to-Digital (ADC) sections are the primary hardware design foci.
- **User Interface (UI):**
  - The Raspberry Pi 4 (RPi4) we are using will host a website that displays information about the download process and completed images.

Applicable Courses from Iowa State University Curriculum

- EE 414: Microwave Engineering
- EE 230: Electronic Circuits and Systems
- EE 333: Electronic Systems Design
- EE 224: Signals and Systems I
- EE 324: Signals and Systems II
- EE 321: Communication Systems I

New Skills/Knowledge Acquired

- All members of our team are Electrical Engineering majors, so we do not have the training that Software Engineering majors would have. We still have a strong programming background varying from individual experiences.
- The use of Git for code management and coding best practices
- Design and implementation of RF systems
- Advanced topics in Applied Digital Communications
- Real-life engineering applications; group communication, planning, problem solving, execution

Contents

List of Figures

List of Tables

Definitions

- μC: Microcontroller
- RPi4: Raspberry Pi 4
- IDE: Integrated Development Environment
- LNA: Low Noise Amplifier
- UART: Universal Asynchronous Receiver-Transmitter
- DSP: Digital Signal Processing
- NOAA: National Oceanic and Atmospheric Administration
- OS: Operating System
- PLL: Phase Locked Loop
- IF: Intermediate Frequency
- GOES: Geostationary Operational Environmental Satellite
- ADC: Analog to Digital Converter
- DADC: Discrete Analog to Digital Converter

- SFTP: SSH File Transfer Protocol
- SSH: Secure Shell
- SPI: Serial Peripheral Interface
- SDR: Software Defined Radio
- RF: Radio Frequency
- ETG: Electronics Technology Group
- COTS: Commercial Off The Shelf
- UI: User Interface
- PCB: Printed Circuit Board
- FPGA: Field Programmable Gate Array
- LNB: Low Noise Block down converter
- LO: Local Oscillator
- IEEE: Institute of Electrical and Electronics Engineers
- USB: Universal Serial Bus
- ISM: Industrial Scientific and Medical
- AC: Alternating Current
- PC: Personal Computer
- GPIO: General Purpose Input/Output
- RAM: Random-access Memory
- VCID: Virtual Channel ID

1 Introduction

1.1 Acknowledgement

Dr. Nathan Neihart, an Associate Professor in Electrical Engineering at Iowa State, is the faculty advisor for this project. He has provided technical support and taught important fundamental principles needed for the project. The ETG, located at Iowa State University's College of Electrical and Computer Engineering, has also provided support for the hardware portion of this project.

1.2 Problem and Project Statement

The goal of this project is to design and build a satellite receiver that will receive image products from the Geostationary Operational Environmental Satellite - 16 (GOES-16). The satellite is in geostationary orbit, meaning it is always above the same location on the earth, and sends image and weather data that is in a published format. These images consist of full disk, and zoomed in sections of the Earth in both visible, Ultra-Violet, and Infra-Red light spectrum. In order to successfully receive and decode the data, the end user must have a system to receive and process the data from the satellite. The focus of this project is to design a system that enables the reception and decoding of the data from the satellite. The problem and a more concise project statement can be found below.

1.2.1 Problem

This project was formed around the desire to construct a system that would enable the reception of the signal from the GOES-16 satellite. There are existing methods of accomplishing this goal, such as commercial systems or utilizing a software defined radio, however this would defeat the purpose of designing the hardware at a lower level. The ability to receive this signal would allow for multi-spectral images of the Earth along with emergency weather information to be received anywhere a line of sight with the satellite could be established. This is also not dependant on internet connectivity, and could therefor be used in remote locations.

1.2.2 Project Statement

In an effort to both learn more about RF systems design, antennas, software design, and to satisfy the desire to receive and generate the images from the GOES-16 satellite ourselves, this project will design a system that enables this ability. To accomplish this ever component from antenna to the final image generation needs to be considered and designed for. In general a antenna with sufficient size must be found to receive the signal which must then be amplified and converted from a RF signal to its initial baseband form so it can be digitized. This digitized signal can then be decoded and processed to obtain the transmitted weather products.
With respect to the changing circumstances due to COVID-19, the project goal of receiving image products from the GOES-16 satellite remain the same, yet the method of approach will vary slightly. Further discussion into the changes of deliverables will be seen throughout this document, including the switch to over the counter components and further use of software applications.

1.3 Operational Environment

There are two environments that the different components of this project will need to survive in. The antenna and low-noise block down-converter (LNB) will need to be able to survive an outdoor environment while the receiver and accompanying hardware will be operated indoors.

1.4 Requirements

- **Antenna:** The antenna for this receiver will be a linearly polarized high gain Wi-Fi grid antenna with a design frequency of 2.4 GHz.
  - Size Requirement: 1.1 meters in diameter

  - Center Frequency: 1694.1 MHz. The feedhorn must be modified for the desired frequency.
  - Connector: N
  - Polarization: Linear, vertical orientation
  - Aiming: ± 2° in both rotation and elevation to ensure proper aiming

- **Filtering & Amplification:** Following the Antenna, the signal needs to be filtered and amplified.
  - Gain: > 40 dB
  - Bandwidth: 2MHz centred at 1694.1 MHz
  - Filtering:
    - Antenna to LNB: Minimal, no high power interference sources in the frequency range
    - LNB, Receiver, and ADC: Filtering needed to reject mixing spurs, local oscillators, and anti aliasing for the ADC

- **Line of Sight:**
  - The antenna needs a clear line of sight to the satellite, limiting possible setup locations.

- **Semi-Portable:**
  - Antenna may not be permanently mounted on university property
  - Antenna must be movable so that it can be put away with ease when not in use

- **Clouds & Weather:**
  - Cloud cover should not impact reception
  - Sun outage, loss of signal due to the Sun being inline with the satellite, will prevent reception at certain times of the year.
  - Equipment will need to be weather resistant if a permanent installation is desired

- **Budget:**
  - Roughly $1,000 USD for all materials and equipment

- **Components**
  - Use of COTS (Commercial off-the-shelf) components where possible for larger components
  - RF and ADC sections will be purpose designed for this system
  - Components used in the RF sections will be either pre-matched to 50 Ohms or provide a standard matching network where possible.

- **User Interface:**
  - The Raspberry Pi will provide the ability to obtain the received and processed images using sftp (SSH File Transfer Protocol)

1.5 Intended Users and Uses

This project is intended to provide the ability to receive weather products from NOAA via the GOES-16 weather satellite without needing a connection to the internet.

There are three intended user groups for the project.

1. **Dr. Neihart:** This project was proposed by Dr. Neihart to expand his ability to receive different weather products from space.

2. **Hobbyists:** There is a large community of hobbyists around the world that seek to receive images and information from weather satellites. This project would provide them with an option that is tailored for the GOES-16 satellite.

3. **Emergency Responders:** This system could be set up in a disaster area to provide real time weather information without requiring access to the internet.

1.6 Assumptions and Limitations

1.6.1 Assumptions:

- The receiver will not be designed to be permanently mounted.
- End users have working knowledge and tools to correctly aim the antenna.
- End users have access to computer along with knowledge of Linux and its manipulation via SSH (Secure Shell)
- The system will be used in an area where the GOES-16 satellite is visible.
- The project development will be continued in the following year.

1.6.2 Limitations:

- The signals the receiver can process will be limited to the GOES-16 and GOES-17 satellites.
- The receiver cannot be placed in inclement weather situations.
- The system requires access to 12V and 5V DC power for the RPi4 and Receiver.
- The cost of the final product should not exceed $1,000.
- Due to COVID-19, the development scope of the project is limited.

1.7 Expected End Product and Deliverables

The end deliverable will be a complete receiver and data processing system. The finished system will be capable of receiving data from the GOES-16 or GOES-17 satellites, process the data, and uploading the image to an easily accessible folder. The system will not require any user input once it has been setup and activated. To achieve this goal, there are a variety of sub-systems that must be delivered, which are described below.

1.7.1 Antenna and RF Systems

The antenna consists of the following components, all of which will be delivered as an assembled unit:

- Antenna dish with modified feedhorn
- Antenna tripod
- Wooden tripod base

The antenna and tripod are commercial products that were purchased for use with this project. The modification to the feedhorn consists of a small block of wood cut to a specific length to move the reflector further from the dipole. The wooden base is a T shaped assembly constructed from two 2x4's to serve as a mounting platform for the tripod. This is necessary as the tripod is intended for mounting to a roof or other structure.

The RF systems consists of the LNB and the receiver. The deliverables associated with each of these are detailed below:

- KiCad project of the LNB uploaded to the Git repository.
- KiCad project of the receiver uploaded to the Git repository.
- Errata list for the receiver project.
- Assembled receiver PCB along with spare un-populated PCBs and components.

The design files for both boards will be provided on the project's Git repository enabling the continuation and modification of the designs. This consists of the KiCad project, schematics, bill of materials, and layout for both the LNB and receiver. Additionally, a list of known errata for the receiver will be included that can be used to update the design for a new revesion. The LNB was not able to be fabricated due to the evolving pandemic disrupting the supply chains for both components and PCB fabrication. The receiver was fabricated in early February allowing for both its fabrication and assembly, however testing was interrupted by the closure of all labs on campus.

1.7.2 Signal Processing

The ADC board consists of a Sallen-Key filter designed around the AD8137 op-amp, power section with dual outputs, and an ADC with an output driver. The final deliverable for this portion of the project is all KiCAD project files uploaded to the Git repository. The ADC board was not able to be fabricated and assembled due to the COVID-19 pandemic. Furthermore, we would have been unable to properly test the functionality of the ADC board while coursework moved to strictly online.

1.7.3 Software

The last sub-system is the software deliverable, which will generate an image file from the binary data. This system will include an executable program and a user interface for specifying which products to process. However, with the advent of the COVID-19 pandemic, we were forced to change our final deliverables to generating an image from the sample test data taken from Lucas Teske's blog. Unfortunately, we were unable to successfully generate an image from the decoded sample data.

1.7.4 Documentation

All documentation can be found on the team's GitLab repository. A setup and troubleshooting guide will be provided for users to refer to. Any additional documentation, such as source code, diagrams, and reports, can be found on the GitLab as well.

2 Specifications and Analysis

2.1 Proposed Design

The proposed design consists of 4 major components: the antenna, LNB and receiver, ADC, and software. A description of each of these components can be found below.

2.1.1 Antenna

The implementation of the antenna is constrained by both cost and gain requirements. Large dish antennas quickly exceed the budget of the project while also requiring permanent mounting. With this in mind, a small dish antenna normally used for long range Wi-Fi transmission can be used with a slight modification and additional amplification of the received signal. The antenna is small enough to be moved to storage when not in use and is not effected by weather conditions. The modification to the antenna consists of extending the distance of the reflector from the feedhorn to adjust for the lower frequency that the antenna will be used for. This can be fabricated from wood or any other non-conductive material.

2.1.2 LNB and Receiver

There are two high level approaches to the RF sections of the project. The first is to utilize a software defined radio to directly receive the signal from the satellite. This approach was not chosen as the goal of the project is to develop a lower level RF system to receive the signals. The second approach is to downmix the signal and demodulate it from its carrier using an architecture more inline with a super-heterodyne receiver. This approach was broken down into two separate board, the LNB and the receiver. This was done as the LNB is traditionally located as close to the feedhorn as possible for satellite systems so that the received signal can be down-converted as soon as possible in the signal path so it can be sent over a longer distance to the receiver with lower losses. The LNB will consist of low noise amplifiers, saw filters, a down-mixer, and synthesizer for local oscillator (LO) generation. The receiver section will consist of a down-mixer and demodulator, providing a baseband output that can then be digitized using an ADC. Through the use of components that are either internally matched to 50 Ohms, or provide a recommended matching network in their datasheets, the RF calculation and simulation is reduced to ensuring that the correct trace width is used and good layout practices are observed.

2.1.3 ADC

Processing the signal and extracting the binary data is the second stage of the final design. The first approach the team investigated was using an FPGA to do all of the desired digital signal processing. This method would require many hours of test and development, and given the lack of team experience with this implementation, it was decided that this option would not be the best. The second approach was to use a μC with a built in analog to digital converter (ADC) to read the demodulated signal and perform computations on the data with to determine the resulting binary waveform. The development of this approach was the focus of the ADC team during the first semester, but due to speed interface problems between the μC and the RPi4, the approach was scrapped. The third approach adopted was the use of a discrete ADC within a new PCB board that the team designed. The ADC board consists of a Sallen-Key filter, power section, input and output drivers, and a discrete ADC chip. A Sallen-Key filter was chosen because it is a simple way to implement a low-pass filter that can be cascaded into developing a higher order filter. Simulation suggested we only needed a second order filter mirrored across a fully differential op-amp. A higher order filter may improve results and can be further implemented and tested during the next phase of this project. Using a fully differential op-amp reduces the need for additional components (PLL) to re-synchronize the incoming signals. The ADC PCB design process can be further referenced in section 5.2.

2.1.4 Software

The final portion needed to implement the project design is software for processing the raw 8 bit samples and generating the image file. The team investigated a variety of ways to implement the software processes by researching different languages, operating systems, and software libraries. The initial approach to implementing the software was to write mostly original software in C programming language and use our software to process digital image data delivered by the ADC team. Writing the software in C programming language would provide fast and efficient data processing. Through further research, the software team determined that the complexity level of C programming necessary for this project was too high for the time allotted. The decision to use external libraries greatly minimized development time and allowed more time to focus on processing image data. The initial design included processing the digital data on a PC but evolved into using a RPi4. Most of the code for the project was adapted and developed for testing and final implementation, all of the way up until the final image construction. More time will be needed to finalize the last parts of the code.

2.2 Design Analysis

2.2.1 RF Section

The antenna was known to work as hobbyist projects using software defined radios, thus going with the Wi-Fi antenna had a high level of assured success. This was later verified by using the antenna along with a software defined radio to receive the signal from the satellite.

The LNB and receiver's performance was estimated using the ADISimRF tool from Analog Devices. This tool enables the signal path to be simulated based upon information from the different component's datasheets. This enabled the simulation of different configurations of the board with respect to component order and choice and was used to assist in final component selection. The other factor in component decision was the ease of integrating it into a 50 Ohm environment and the cost. As higher end components are sought out, or components that integrate a large number of RF blocks into one, the price increases substantially. This led to the LNB having a discrete mixer and synthesizer as opposed to a single integrated unit. This however had the oposite effect for the receiver board, where it was more cost effective to use an integrated IC from Analog Devices, the ADRF6850, which integrates a synthesizer, LNA's, and a demodulator into a single device. This allowed the receiver board to be a single chip solution with respect to the RF components. A deeper explanation of the LNB and Receiver board can be found in sections 5.1 and 5.2.

2.2.2 Signal Processing

The three approaches described in the above section had different strengths and weaknesses. First, although the FPGA method would give many benefits for signal processing, it required many hours of research and development. It was decided that this option would not be the best. The second approach, utilizing a μC, was the focus of the ADC team during the first semester. The team focused on learning about the μC, it's IDE, and how to configure and test the system. The μC was configured and tested for SPI communication as well as ADC functionality. Due to speed interface problems between the μC output and the RPi4, the approach was scrapped. The team moved to a discrete ADC approach within the first few weeks of the second semester. After researching the best ADC for the project, the team focused on two tasks: creating the new required PCB and testing the data transfer into the RPi4. The testing and development of both tasks were halted by the impact of labs closing down on campus.

2.2.3 Software

The main difficulty with writing the software was a lack of understanding of the structure of the data that the satellite sends. There are many documents from NOAA online that are free to view, but finding the necessary information is extremely difficult because they are lengthy documents and are very dense. Fortunately for us, we discovered the blog by Lucas Teske that walked through the data processing step-by-step and briefly explained a little bit about each of the necessary steps to decode and sort through all of the transmitted data. The blog did not include every detail regarding the full implementation, but did list every step that would need to be performed, so this served as a very useful guide to inform us of where exactly we needed to be investing time to understand how to go about the design.

2.2.4 Observations

The initial difficulty with this project came from the group's lack of experience in the area of satellite communications. The majority of the team did not have any prior knowledge of the GOES-16 satellite and the requirements for interfacing with the satellite. Due to this, we completed all research from the beginning of the engineering process. Through this research, multiple design ideas and examples were found. Overall, the progress of the group moved steadily throughout the year. There were some key learning experiences found within the multiple failed design approaches: switching to the OpenSat library, removing the μC and adding a discrete ADC, COVID-19 complications, etc.

The proposed design for the project as stated in *Section 2.1* holds both strengths and weaknesses. The RF front-end is one of the project's strengths, using both COTS components as well as a PCB to receive the signal and down-convert it to IF. A weakness of the initial design was the multiple failed approaches found within the ADC section; this eventually lead to the successful new ADC PCB design. Another weakness of the proposed design was within the software implementation. It was observed that the difficulty in writing a decoding library was too great, and a revised approach has been implemented. This has led to a strength in the use of the OpenSat library, which allows for faster implementation and development, as well as using previously tested functions.

2.3 Development Process

This project used a combination of an agile workflow and the waterfall process. Overall project organization and tracking was accomplished using a GitLab repository. This combination of development techniques was chosen due to the team being split into three sub groups which can be seen below:

- RF Team: Antenna, LNB, and Receiver boards
- ADC Team: ADC board and data acquisition
- Software Team: Data processing, image generation, user interface

Each of these teams consisted of two members and if necessary members could switch teams or assist another team for a period of time. During weekly meetings, progress from the specific teams would be reported and any future needs from other teams would be stated. Outside of the weekly meetings, the individual teams worked to complete any necessary tasks and reached out to other teams through either e-mail or the team's Slack chat system. Issues could be created on the Git repository and other teams or individual members could be tagged to obtain focus on a specific issue.

2.4 Conceptual Sketch

The overall sketch for the project is shown in Figure 1. This overviews the major components that are found in the project. A description and explanation of the sub block can be found below.

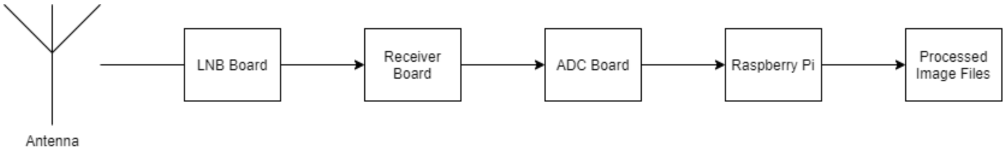


Figure 1: High Level System Overview

2.4.1 LNB

The block diagram for the LNB can be found in figure 2, which shows the signal path through the LNB along with the gain provided and the frequencies before and after the mixer. This section takes the signal from the antenna, amplifies it and filters it, then down-mixes the signal where it is further filtered and amplified before being sent to the receiver.

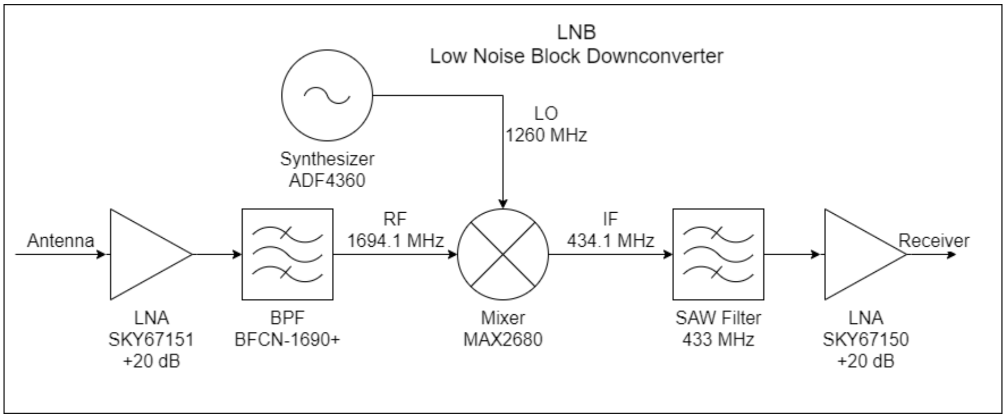


Figure 2: Block diagram for LNB RF path

2.4.2 Receiver

The receiver is comprised of a single chip as it is a fully integrated component from Analog Devices. The internal workings of this component can be seen in figure 3 below. This section takes the down-mixed signal from the LNB, further amplifies it, then down-mixes it to baseband and extracts the I and Q branches that can then be digitized.

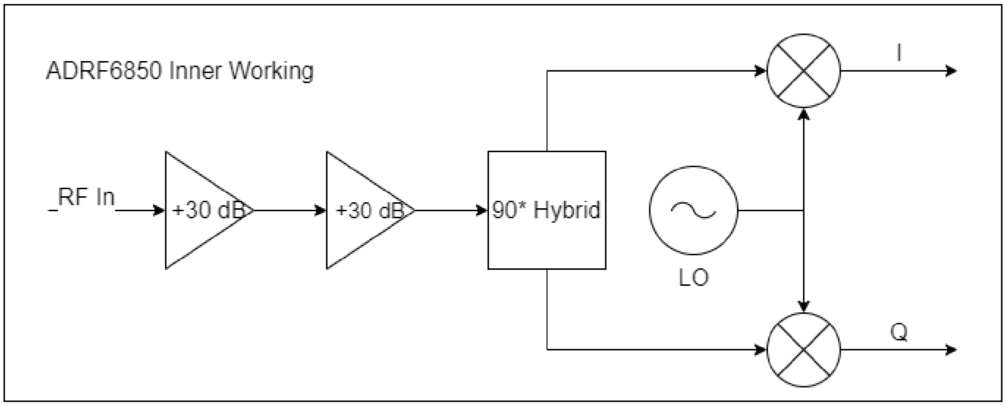
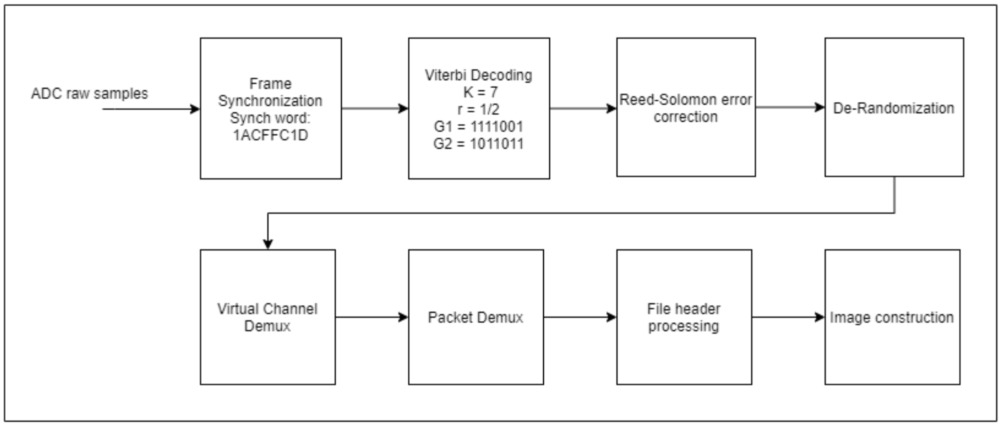


Figure 3: Internal block diagram of the ADRF6850 used in the receiver

2.4.3 Software

The data is processed with software in the order shown in figure 4. The software is written such that each of these blocks is a single function that is easy to call so that testing and debugging is easy.



ADC raw samples
Frame Synchronization Synch word: 1ACFFC1D
Viterbi Decoding K = 7 r = 1/2 G1 = 1111001 G2 = 1011011
Reed-Solomon error correction
De-Randomization
Virtual Channel Demux
Packet Demux
File header processing
Image construction

Figure 4: Processing flow for the software.

3 Statement of Work

3.1 Previous work and Literature

There has already been extensive work on this type of receiver by both engineers and hobbyists. There is hardware and software available for purchase to easily implement the entire system; our team attempted to design and implement our own system that does the same thing.

One project that we referenced (Roberson et al.) used a combination of RF hardware and an SDR to extract the information from the satellite signal. The antenna for the design was a 1.0 meter antenna which was connected to a Low Noise L-Band Block (LNB) downconverter. The IF signal from the LNB was then connected to a circuit board containing an ADC and FPGA. The ADC was used to digitize the IF signal, and the FPGA was programmed to handle all digital signal processing duties. The digital signal from the FPGA was then sent through USB to an SDR, which handled all of the signal receiving and checking. This particular system is close to what we want to do and we will use this as a guide.

In general, we will be modifying this system in a few ways. We will use more components in the RF signal manipulation stage and also try to use software to do everything the FPGA is doing. Using software to do this will simplify what the system is doing as a whole. However, the antenna used in the model system will likely be the same antenna we use to implement the system.

There is an open-source Github library, called libcorrect, made primarily by Brian Armstrong (Brettbolt) that has the same goal as our team. We used the functions from this library for performing Viterbi decoding and Reed Solomon error correction. The Github repository can be found at the following link:

https://github.com/quiet/libcorrect

With regards to background and literature, we researched the necessary concepts through the use of books such as Digital Communications (Sklar) as well as RF Microelectronics (Razavi).

3.2 Technology Considerations

3.2.1 RF Section

- A less integrated design was chosen for the LNB than what would be possible with what is available on the market. This was done deliberately both for cost considerations and for the experience of a lower level implementation.
- As the carrier frequency is specialized, there was not much available for input filtering, with 2 out of the three found no longer being in production.
- A 4 layer using a FR408 substrate was chosen for the PCB to ensure a continuous ground plane for the micro strips and enable fast prototyping.

3.2.2 ADC Section

- Capabilities of the RPi4 allow for sufficient data transfer speed
- ADC board uses commonly available components (ADC, op-amps, filters, etc.)
- Data format will be consistent with the rest of the group

3.2.3 Software Section

- Software must compile on Raspbian OS on a RPi4
- Program to process the raw data must be optimized sufficiently well, otherwise the runtime may be unacceptably long

3.3 Task Decomposition

The work is distributed among the teams we have assigned.

- The RF team is responsible for receiving the signal and demodulating it to allow for digitization.
- The ADC team will digitize the signal and transfer the data onto a RPi4 for image processing.
- The Software team will write the code to interpret the raw binary data, generate an image, and upload it to a website.

3.3.1 RF Team Tasks

The following tasks need to be completed for the RF portion of the project to be complete:

1. Antenna
   - Research, find, and purchase appropriate antenna
   - Research, find, and purchase appropriate tripod
   - Assemble antenna and attach to tripod
   - Design and build a base for the tripod
   - Aim antenna at the satellite
   - Determine and execute necessary feedhorn modification
2. LNB
   - Determine received signals specifications (frequency, bandwidth, and power level)
   - Determine availability of and choose filters for both the RF and IF sections
   - Determine gain necessary by examining existing products
   - Select components and begin RF signal path simulation
   - Design schematic for both RF and Power sections of the LNB
   - Layout the PCB
   - Fabricate and assemble the PCB
   - Test the LNB, this involves power and RF testing
3. Receiver
   - Determine the power level and frequency coming from the LNB
   - Determine best approach to down-convert to baseband
   - RF path simulation
   - Component selection
   - Design schematic for both RF and Power sections of the receiver
   - Layout the PCB
   - Fabricate and assemble the PCB
   - Test the receiver, this involves power and RF testing

3.3.2 ADC Team Tasks

The following tasks need to be completed for the ADC portion of the project to be complete:

1. Board Design
   - Determine received signal requirements (frequency, voltage, resolution, etc.)
   - Research and select appropriate components
   - design schematic for Salley-Key filter, power section, and drivers
   - Layout the PCB
   - Fabricate and assemble the PCB
   - Test the ADC board
2. Data Testing
   - Define data requirements from incoming IF signal (data rate, amplitude and offset, etc.)
   - Determine test plan for both incoming and outgoing data signal
   - Write C code on RPi4 to receive data and write as a binary file
   - Test components and their max capabilities

3.3.3 Software Team Tasks

The following tasks need to be completed for the software portion of the project to be complete:

1. Software
   - Set up the operating system for the RPi4.
   - Install and set up file paths needed for resources from OpenSatProject to work.
   - Write functions to carry out Frame Decoding
   - Use available scripts in OpenSatProject to carry out Channel Demuxing.
   - Use available resources from Lucas Teske's blog to complete the file assembler.
2. Software Testing
   - Find the sync word within the sample data using the Frame Decoding and Channel Demuxing functions.
   - Generate an image from the sample data available.
   - Test with multiple instances of data received from the GOES satellites.

3.4 Possible Risks and Risk Management

3.4.1 Project Completion Risks:

- Understanding how to process data packets into images requires a substantial amount of background knowledge in digital signal processing. Prior to this project, the team's combined educational and professional experience with DSP was very limited.
- The inability for hardware and software matching to meet the speed of data transfer would prove detrimental to project completion.
- Signal strength must be boosted to meet processing requirements. If the signal strength is not strong enough, the rest of the system will have trouble processing the signal and will not produce an image.
- Risk to hardware exists by improper connection and powering of system components.

3.4.2 Physical and Environmental Risks:

- Injury during antenna assembly or during its movement.
- Injury during PCB assembly
- Damage to PCB's from ESD events.

3.4.3 Risk Mitigation:

- The software team tirelessly researched DSP techniques and practices to gain a general understanding for moving forward with the project.
- Thorough analysis of datasheets was performed to determine if system components abilities were sufficient.
- A second LNA was added to the LNB on the IF side to ensure that strong enough signal could be sent to the receiver board.
- All system integration was performed meticulously to ensure no damage to components.
- Large hardware components were moved and assembled as a team to lessen physical strain.

3.5 Project Milestones and Evaluation Criteria

There are a number of major milestones that have been completed throughout the course of the project. Each of our sub-teams have milestones of their own, and the final system itself also has milestones.

3.5.1 RF Team

The first major milestone is the design of the receiver. This was chosen to be the first RF board designed as it is necessary to be able to successfully test the ADC board. The sub-milestones for this board were as follows:

- Completion of ADISimRF simulation and component selection.
- Completion of the schematic and first design review.
- Completion of the layout and first design review.
- Revision of the layout and send for fabrication.
- Order BOM and assemble PCBs
- Preform initial testing and fix any found errors
- Preform full testing

This board is considered completed when it can take a 434.1 MHz signal with 2 MHz of bandwidth and down-mix it to baseband and output it as an I and Q branch. This was not completed as testing was halted due to the pandemic shutting down access to all labs on campus.

The LNB follows an identical milestone process, but actual implementation was not started until the receiver board was shipped for fabrication. Due to the university shutting down, it was decided to not continue the fabrication of this board at this time. Progress was halted just before the layout design review.

3.5.2 ADC Team

The first major milestone that the ADC team accomplished was to successfully design the ADC board. This includes selecting the ADC, designing the Salley-Key filter, power section, and drivers. The second milestone for the ADC group was to be able to successfully receive and digitize, and output the incoming signal. Using an oscilloscope and C code on the RPi4, the max speed of the GPIO pins was found. This allows for preliminary testing of the output side of the signal digitization task. To finalize testing, the final boards would need to have been finished, thus making this milestone unachievable.

3.5.3 Software Team

The Software team used a guide provided by Lucas Teske on his blog to aide in development. Mr. Teske also provides sample data files so that the software team can test each component as it is developed. The milestones for the software team, then, are developing each smaller algorithm and testing it for correct behavior one at a time. For example, the first major step is to sort the data into frames by performing frame synchronization. We tested this algorithm for correctness by writing a MATLAB file that displays the frame data as bits - that is, a 1 is black and a 0 is white - to observe that there is a coherent structure to the synchronized frames in the way we expect.

3.6 Project Tracking Procedures

This project has been tracked through Git. The Wiki feature built into GitLab has tracked the information pertaining to different portions of the project, and has allowed us to explain and document research about the project for others to view. The project repository holds all of the software files, with multiple people having contributed to the software throughout the year. Non-software files, such as images or diagrams, have been uploaded to the repository. The last major part of our project tracking procedures was using issues within Git to organize and keep track of past, present, and future tasks. Team members were responsible for creating stories deemed necessary, and writing updates within the issue as they progressed.

In addition to Git, the team had weekly meetings with Dr. Neihart where the group's progress was discussed. Additional meetings were held as a team to discuss how the different groups could come together at the boundaries of their respective responsibilities. As in person meetings became impossible, the weekly meeting was transitioned to Cisco's WebEx.

3.7 Expected Results and Validation

The high-level end result for this project is to successfully acquire images of the Earth from data received from the satellite. Each of the subsystems have known inputs and outputs, so they can been tested individually.

3.7.1 RF Team

The RF boards can be tested individually for their performance criteria, that begin gain, signal to noise ratio, and bandwidth. This can be compared to the results of the simulations from ADISimRF to determine how effective the design was. The expected result, if the board were to have been fabricated and tested to completion would be results that were within 10% of the simulations while allowing a successful reception of the signal from the satellite.

3.7.2 ADC Team

The ADC board is ready for fabrication. All necessary design files are uploaded to the Git repository. It is expected that the follow-on project team be able to order and assemble the necessary components to complete the ADC hardware section of the project and complete functionality tests. We expect the ADC board to output the correct bit stream to the RPi4 with minimal error resulting from the ADC board.

3.7.3 Software Team

The program so far works well for the test data file for the parts that we have completed. This test file has a reasonable SNR, so it made developing and debugging the program relatively smooth. To really test our system, we would need more data taken ourselves, but we were not able to record any.

4 Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

The Gantt chart shown in Figure 5 outlines the general progression of the project throughout the course of the two semesters. The project was broken down into four time periods of development: research, prototyping, testing, and finalizing the project. The tasks shown in the chart correlate with the requirements previously discussed. Overall, the team schedule accounts for individual sub-team progress, along with the overall progress for the team.

The project schedule was created to help complete the project in two semesters. To do this, the team worked to have the project researched thoroughly and the requirements defined by the end of October of the first semester. The initial prototypes for each of the sub-teams were aimed to be completed early in the second semester to allow time for testing and debugging. The second semester was set aside for troubleshooting and testing of the system, with time scheduled in the second half of the semester to develop the final product. The project fell behind schedule due to multiple changing approaches, but was still aimed to finalize the project by the end of the semester. Due to the labs closing, all testing and assembly had to be stopped, thus making the current project schedule impractical.

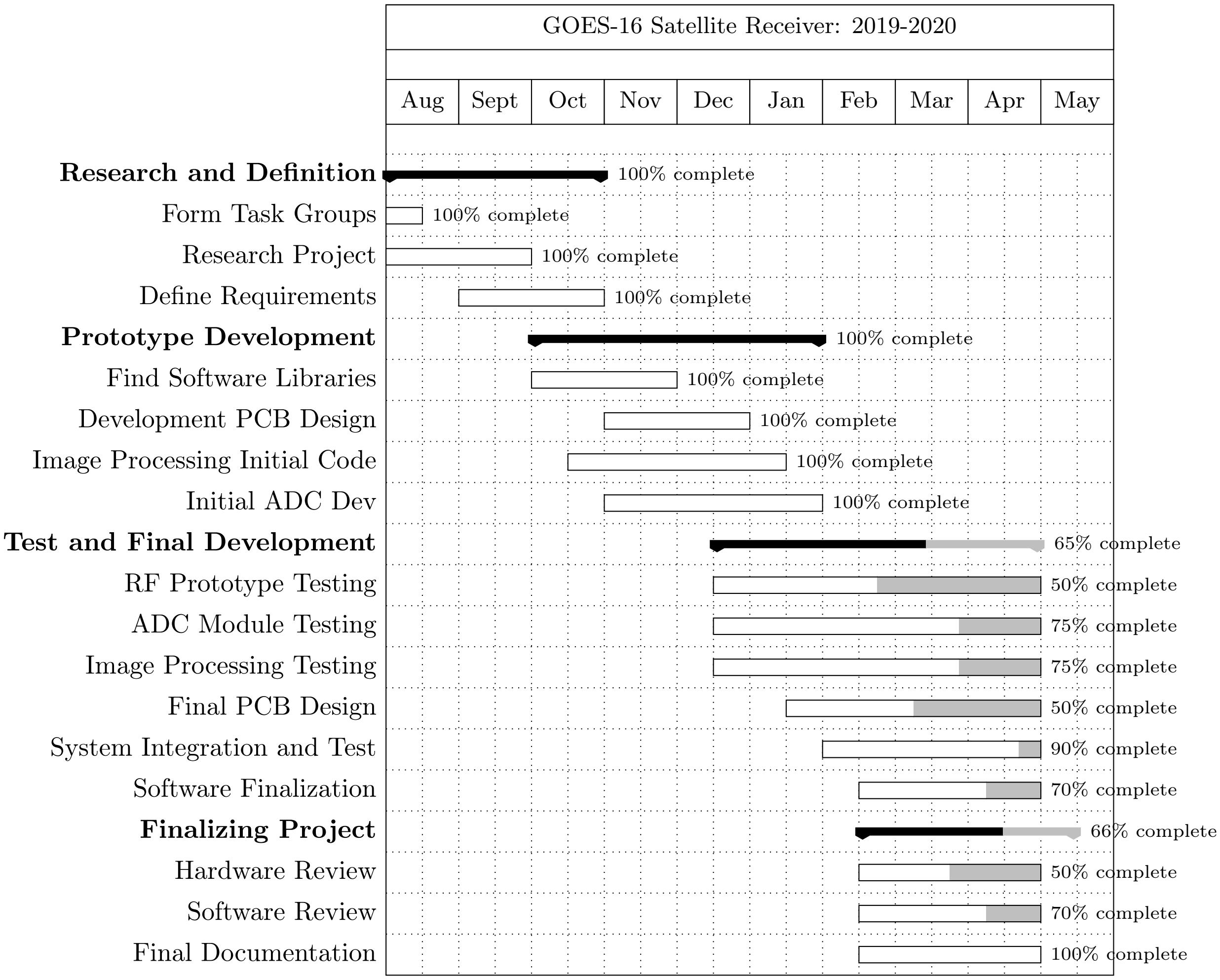


Figure 5: Project Gantt Chart

Along with the Gantt chart shown in Figure 5, the team has defined some dates at which specific deliverables were completed. They are as follows:

- RF Team
  - November 19 - Antenna placed in courtyard of Coover Hall
  - December 20 - Receiver board design completed, review scheduled for after break.
  - January 15 - Additional revision completed.
  - January 17 - LNB board design started
  - January 29 - Design review for receiver board layout completed
  - January 31 - Receiver board ordered from Advanced Circuits
  - February 14 - Board received from Advanced Circuits
  - February 21 - Assembly completed initial testing began.
  - February 23 - Error found with loop filter, components ordered
  - March 1 - Components received from DigiKey, repairs completed
  - March 5 - Testing shows issue with script used for testing receiver, rewriting driver.
  - March 10 - Continuing design of LNB board.
  - End of March - Access to labs lost, moving to refining the design of the LNB board
- ADC Team
  - November 11 - Configuration and testing of μC
  - January 19 - Revised ADC approach
  - February 1 - Finalized new test plan
  - April 1 - ADC circuit design completed
- Software Team
  - March 2 - Frame synchronization algorithm functional and tested
  - March 10 - RPi4-hosted website operational
  - April 10 - Decoding and channel demuxer functional and tested

4.2 Feasibility Assessment

Overall, the project required an extensive amount of work and knowledge in RF, signal processing, and software systems. As a result, this project has been difficult for the team to carry out successfully. However, the project was completely feasible given the amount of documentation available online, along with the professional advice of the advisor and other professors at Iowa State University. By separating the team into groups focused on specific sub-systems, team members were able to become experts in the field they are worked on. Due to COVID-19, the project was no longer feasible as the labs were closed and testing had to be halted.

Initially, the goal of the project was to create a completely weather proof system that can automatically align the antenna, receive the signal, process the signal, and output a high resolution image of Earth. Upon further investigation, a few of the requirements were removed or altered to increase the likelihood of the project being completed successfully. First, it was decided the antenna would be manually aimed to reduce the amount of hardware and software required for the implementation. Next, the team chose to remove the weatherproof requirement, as this requires a significant amount of work to protect hardware from temperature and moisture. With these requirements removed, the focus of the project could be shifted to the RF circuitry, signal processing, and decoding software. The group was able to complete all circuit designs, test the feasibility of the data transfer process, and complete most of the software.

There were a few foreseen challenges that hindered the development of the final product. One issue that came up initially was the background experience provided by the team. With all electrical engineering majors, there is not a strong background in software, which has proven to require many hours of research. In addition to that, the level of complexity in regards to the signal processing and data decoding is much higher than initially expected. It took much longer to troubleshoot the system due to lack of expertise in the field. The last thing that hindered progress is lead-times when ordering hardware to prototype. Development was slowed because of long wait-times for ordered parts.

4.3 Personnel Effort Requirements

Task	Est. Time	Description
Define System Requirements	2 weeks	Initial system and functional requirements were explored and defined so that the team could begin project research.
Research Domain Topics	4 weeks	Meetings with both the group and Dr. Neihart directed individual groups into specific research areas. Background knowledge on satellite communications, digital communications, signal processing, etc. were explored.
RF Component Selection	3 Weeks	Searched manufacture offerings to find components that would fill the need of the various RF subsystems.
OpenSat Research	1 weeks	Researched the necessary background for implementing the OpenSat functional library.
RF Simulation	2 Weeks	Run simulations of the primary components, (LNA, Mixer, Synthesizer) to determine the passive components required for their proper operation in the systems design. RF Sumulation was unable to be completed due to the COVID-19 pandemic.
RF Schematic Creation	5 Weeks	Created the schematics for the different RF boards that were required, ensuring that all of the footprints for the components were correct. Verified all connections with the datasheets for the components.
ADC Selection and Design	4 weeks	Selected an ADC platform that covers necessary data and communication requirements. Designed schematic and layout for ADC board.
RF Board Layout	5 Weeks	Layout the RF boards for the initial prototypes. This required meetings with Dr. Neihart to verify layout choices.
Data transfer testing	2 weeks	Configuration and of RPi4 and ADC. This included testing speed capabilities, communication, etc.
RF Board Assembly and Testing	3 Weeks	Assembled the prototype board and performed testing for requirement compliance. This required the assistance of Dr. Neihart for access to the equipment in the research laboratory. Full testing was unable to be completed due to the COVID-19 pandemic.
Compile OpenSat on RPi4	2-3 weeks	Cloned source code from the repository and built for use on the Rpi. Ensured that all of the different components of OpenSat were functional and caused no compilation errors.
System Integration	3 weeks	System Integration was unable to be completed due to the COVID-19 pandemic.
System Testing	2 weeks	This would have involved recording enough data for an image and seeing if the system can take that data and generate an image. Unfortunately, we were unable to test our system due to the COVID-19 pandemic.

Table 1: Personnel Effort Requirements

Description	Cost ($)
LCOM 2.4GHz 24 dBi Dish Antenna	130
ADRF6850 I/Q Demodulator	42
Various RF Boards Components	300
Various ADC Board Components	100
STM32F303 Nucleo	22
Raspberry Pi 4	47
Total	**650**

Table 2: Bill of Materials

4.4 Other Resource Requirements

Aside from financial resources, the team used lab equipment available in Coover such as RF testing equipment, oscilloscopes, and multimeters to test the functionality of the project.

On top of that, the team used the services of Electronics and Technology Group (ETG) to order parts needed for the project and to borrow additional equipment for development and testing.

Furthermore, the rooms of Coover are also important for holding weekly team meetings with the team advisor to discuss progress and direction of the project.

4.5 Financial Requirements

This project was provided with a budget of $1,000 USD. The primary financial requirement of the project was to stay within that budget. The following table shows the expenses of the project.

5 Hardware Design and System Integration

5.1 LNB Design and Layout

The LNB was designed to amplify, filter, and down-mix the incoming signal from the antenna so that it could be sent a longer distance over coaxial cable to the receiver. With this in mind, the goal was to ensure a high signal to noise ratio be maintained throughout the device as well as a reasonable amount of gain. To accomplish this the following components were chosen

- LNAs: SKY67151 & SKY67150
- SAW Filters: BFCN-1690 & AFS43453
- Mixer: MAX2680
- Synthesizer: ADF4360

The LNAs from Skyworks provided some of the lowest noise figures on the market while also provided over 20 dB of gain each. Additionally, they provided matching networks that worked over a wide range of frequencies. The saw filters were chosen because they were the only devices still being produced that covered the required frequency bands. The BFCN-1690 from Mini-Circuits covered the exact input frequency range for the satellite while the AFS43453 by Abracon had the required bandwidth in its passband to allow the downmixing of the signal to the 434 MHz ISM band. Finally the mixer from Maxim was both low in cost and provided a low noise and easy implementation to down-mix the signal. Since these components were either internally matched to 50 Ohms or provided a matching network, they could be easily chained together. From this point Analog Devices ADISimRF tool was used to estimate the performance of the LNB.

The layout of the

5.2 ADC Design and Layout

5.2.1 Sallen-Key Filter Design and Simulation

The ADC hardware design process began with designing a Sallen-Key Low-Pass Filter capable of providing unity gain and a passband large enough to allow the high-frequency RF signal at 1694.1MHz to pass while attenuating all frequencies above a -3dB cutoff of roughly 2MHz. This was accomplished using an LTSpice filter designer provided by Analog Devices (https://tools.analog.com/en/adcdriver).

The Sallen-Key filter for this design uses a fully differential op-amp in order to provide a common signal path for both signal outputs of the ADRF. This also reduces the risk of the ADC inputs and outputs from losing synchronization.

See Appendix B for filter simulation details and results.

5.2.2 Component Choices

- AD8137 Fully-Differential Op-amp
- TPS7A88 Low Noise Voltage Regulator
- MAX1426 10-bit, 10 Msps ADC
- SN74LVC827A 10-Bit Buffer/Driver
- ECS-TXO-2520 SMD Crystal Clock Oscillator

See Appendix B for the ADC BOM.

5.2.3 KiCAD Schematic and Layout Design Process

The MAX1426 ADC chip needed both an input driver and output driver. The AD8137 was the recommended driver given specifications during the Sallen-key filter design process.

The ADC PCB schematic was built by section. Sections include: power, ADRF input, input driver, and ADC and output driver. Each section was built using common practices and techniques for high speed/RF design.

The power section includes a common barrel jack, a ferrite bead for noise reduction, a fuse for circuit protection, and decoupling circuitry before entering the TPS7A88 Low Noise Voltage Regulator that supplies the two needed voltages of 3.3 and 5 Volts. A simple voltage divider was used to provide 2.5 Volts as Vcom. Vcom is needed as a reference voltage for the ADC and ADC input driver.

The ADC input driver is supplied 5 Volts from the Voltage regulator and follows the Sallen-Key filter schematic from Analog Devices. The input driver is fed two inputs from the ADRF and outputs to the MAX1426 ADC.

The MAX1426 ADC chip was chosen because of its bit depth and sampling capability. The minimum desired bit depth was 8 bits and minimum desired sampling rate was 4 to 5 Msps. The MAX1426 has more than enough capability for the projects needs. The schematic for the ADC and ADC output driver follow recommendations in both the MAX1426 datasheet and the MAX1425/MAX1426 evaluation board documentation. The MAX1426 uses the ECS-TXO-2520 SMD Crystal Clock Oscillator to reference a maximum clock frequency of 10 MHz.

For the PCB Layout, sections were organized in reference to the schematic. Once the sections were organized with high speed practices followed, the sections were integrated and organized for minimum noise and loss, neatness, and usability. Top and bottom ground planes are split between analog and digital sections of the ADC chip and output driver to allow isolation and lower noise. The split planes are tied together in small sections away from the ADC. Vias connect top and bottom ground planes.

See Appendix B for Schematic and Layout images.

6 Testing and Implementation

6.1 Interface Specifications

The team has investigated methods of carrying out testing on both the sub-systems and the final product. In order to do this, the team established hardware and software connections between the systems and test equipment. The RF team worked on a way to easily test various characteristics of the signal. That required RF cables and connectors, along with some RF test equipment. Further testing using the methods found for sub-systems and the final project was limited due to the COVID-19 pandemic.

In regards to the ADC and RPi4, strictly online coursework and the closing of campus limited the team's ability to fabricate and assemble the ADC board. Therefore, testing was unable to be completed due to the COVID-19 pandemic. The PCB schematic and layout have been completed for use by the next project team. The future team should be able to use the provided files for fabrication and assembly, and then test the sub-system for proper functionality. They will then be able to interface the ADC board with the RPi4 and test functionality of the ADC board to output arbitrary signal data to a binary file written to the RPi4. Following successful testing would be integrating the ADC board with the full system.

In regards to the Software, each block from figure 4 was a separate function and was tested individually before being integrated together, with the exception of frame synchronization and Viterbi decoding because it was easier to verify correct operation of the frame synchronization after the decoding was done. Each function generated a new binary file, so as development progressed we always had the binary file from the previous block to use for testing, so that made testing less faster because the frame synchronization - the first step - is by the most computationally expensive.

6.2 Hardware and Software

The following hardware and software was used for the testing of the components in the project
Raspian OS - The OS that the team is using for the RPi4. It includes a terminal where the team will be running the code on. The terminal is also used to install the necessary personal package archives needed to compile the code.
Oscilloscope - The oscilloscope is an instrument used to measure and analyze waveforms of signals. The oscilloscope was used to verify the functionality of the μC's SPI through the MOSI, MISO, CLK and SS pins. Furthermore, with the team's decision to use a DADC instead of the μC, The oscilloscope would have been used to test functionality of the ADC board. Due to unforeseen circumstances, we were unable to proceed this far in testing.
SDR - The SDR will be used as a way to test the signal reception of the antenna. It will act as a simple spectrum analyzer and the team will use it to confirm the antenna is aimed in the correct position, and that the desired signal is being received with the antenna.
Spectrum Analyzer - A spectrum analyzer will be used to test properties of the RF circuit. It will be used as a way to estimate the total attenuation in the circuit, along with confirming that all filtering and down-mixing is successful.
Serial Monitor - The team will use a serial communications monitor software, such as PuTTy, to ensure signal processing taking place in the ADC is working correctly. The serial monitor will allow the team to troubleshoot and test the code in the project during run-time.

6.3 Functional Testing

With the occurrence of COVID-19, the team was unsuccessful in developing a working prototype of the satellite receiver. However, each team has made significant progress in the testing and development of each individual section compared to last semester.

6.3.1 RF Team

The functional testing of the RF hardware is detailed in the process section below. The primary function of this testing will be to ensure that the components function close to that of the simulations and datasheet figures.

6.3.2 ADC Team

For the ADC team, there were two main testing areas. The first focus was on the transfer of data to and from the ADC board. The first semester consisted of tests focusing on SPI communication, data rate limitations, and reading or writing arbitrary data. These tests were performed on the μC and RPi4, and were viewed using an oscilloscope and PC. The max GPIO speed of the RPi4 was found using C code. The second focus was on the ADC board. Testing during the second semester was limited to simulations as access to labs was no longer available. No further tests were able to be completed on the board.

6.3.3 Software Team

The software team tested the validity of the frame synchronization and Viterbi algorithms by using the sample data available on Lucas Teske's blog and was successfully able to identify the sync word in the test data. The test processes that were conducted are listed in detail in the next section of this document.

6.4 Non-Functional Testing

6.4.1 RF

For the RF portion of the project the non-function testing is closely tied into that of the functional testing as these systems either work or do not work.

6.4.2 ADC

There were no major non-functional tests performed within the ADC team. Potential tests that were prevented due to COVID-19 closing labs down include the quality of the data stream, overall board quality, and integration testing.

6.4.3 Software

The end product is images of Earth. Since the software team was not able to finish the program to fully produce images from raw data, we could not do any non-functional testing.

6.5 Process

6.5.1 Hardware

Hardware testing follows the following initial steps:

1. During Assembly
   - Populate all passives, connectors, and power sections.
   - Connect an oscilloscope to the output voltage rail(s) and a power supply to the power input and slowly raise the voltage.
   - Measure the LDO voltage if applicable, voltage ripple, current draw, and output voltage accuracy.

- Populate major ICs one at a time and verify IC power up using the above methods, paying close attention to the current draw.

2. After Assembly

   - Measure output frequency of any oscillators or crystals to both ensure they at the correct frequency.
   - Attempt to communicate with ICs providing digital I/O's, eg SPI or I2C busses.

3. LNB Testing

   - Measure output for DC and ensure AC coupling of both the input and output
   - Set the synthesizer to generate a LO of 1260 MHz and measure its output with a spectrum analyzer
   - Input a generated signal similar in bandwidth and frequency to that of the GOES-16's HRIT signal
   - Measure the spectrum and power level of the resulting 434.1 MHz output using a spectrum analyzer.

4. Receiver Testing

   - Measure output for DC and ensure AC coupling of both the input and output
   - Set the synthesizer to generate a LO of 434.1 MHz and measure its output with a spectrum analyzer
   - Input a baseband BPSK signal with a symbol rate of 2 mbps
   - Observe both the I and Q branch of the output on an oscilloscope

5. Complete Chain Testing

   - Connect the LNB and Receiver, power both systems
   - Input a 1694.1 MHz BPSK modulated signal to the LNB
   - View and measure the I and Q branch results from the receiver on an oscilloscope.

6.5.2 Software

Software testing was carried out as follows:

1. Frame Synchronization + Viterbi decoding

   - The frame synchronization and Viterbi decoding were tested together because before the signal is decoded we are still working with 8 bit soft samples instead of digitized hard samples. To visualize this part, we printed the bits to a text file and wrote a MATLAB script to display the bits in grayscale. Figure 6 below shows the bit display. Each row in the figure is one frame (only the first 180 bits out of 1020 - this image is zoomed to show the detail of the synch words). The vertical bars on the left, then, are the synch words of each frame, followed by the headers and the more random-looking payload on the right of the image.

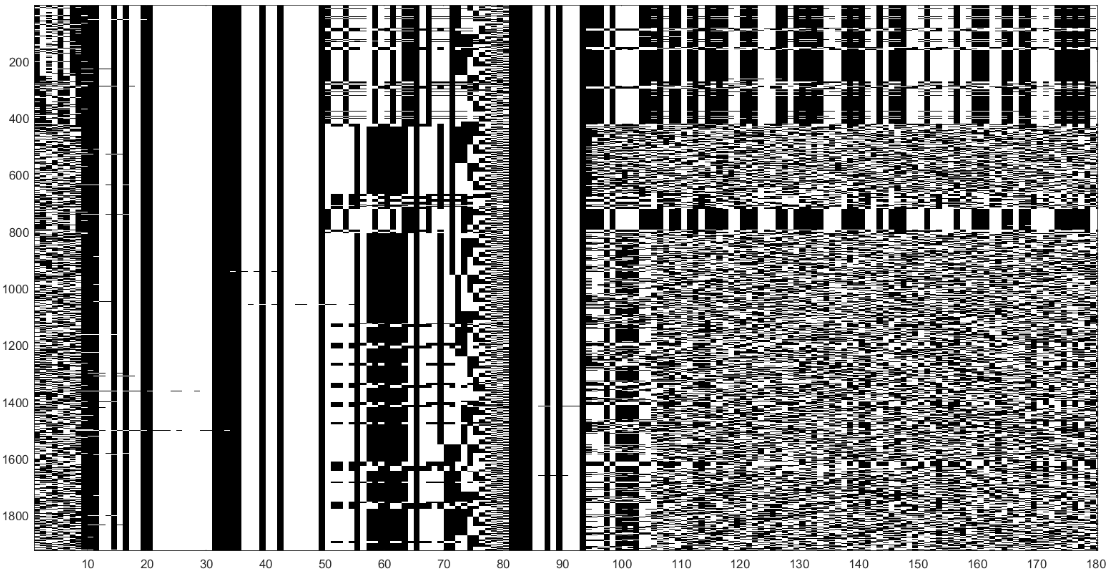


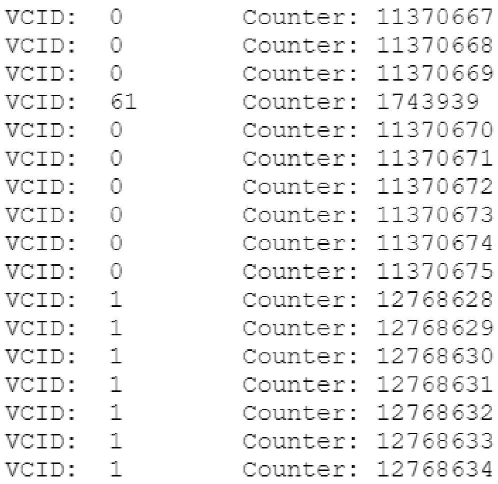
Figure 6: Bit display of synchronized and decoded frames.

2. Reed Solomon error correction

   - The Reed Solomon error correction algorithm is an optional step because it does not alter the data structure in any way; it uses 128 bits at the end of the frame to do parity checking to attempt to correct bits that may have been inverted in the transmission process. As such, we skipped this part to put our focus on more important sections of the software and since we did not finish everything we did not go back and implement the Reed Solomon error correction.

3. De-randomization

   - Before the frames are de-randomized, all of the data will not be coherent if you try looking for information contained within it except for the synch words because the synch words are not randomized. To test whether the de-randomization was done correctly, we read the virtual channel IDs and counter number for each frame and printed it out to check that it makes sense. Figure 7 shows an example of this. What we expect is that the VCID's are of a subset of valid numbers, and that the counter for a given frame for each VCID is always one greater than the last one. The figure shows that this is what we observe, so we know the de-randomization works correctly.



```
VCID:  0        Counter: 11370667
VCID:  0        Counter: 11370668
VCID:  0        Counter: 11370669
VCID:  61       Counter: 1743939
VCID:  0        Counter: 11370670
VCID:  0        Counter: 11370671
VCID:  0        Counter: 11370672
VCID:  0        Counter: 11370673
VCID:  0        Counter: 11370674
VCID:  0        Counter: 11370675
VCID:  1        Counter: 12768628
VCID:  1        Counter: 12768629
VCID:  1        Counter: 12768630
VCID:  1        Counter: 12768631
VCID:  1        Counter: 12768632
VCID:  1        Counter: 12768633
VCID:  1        Counter: 12768634
```

Figure 7: Bit display of synchronized and decoded frames.

4. Virtual Channel Demuxer

   - The virtual channel demuxer did not need testing because once we had verified that the derandomization works correctly, we already have the virtual channel ID's and we can just save the frames for a given VCID.

5. Everything else

   - We were not able to get the rest of the program tested. We tried using a Python scripts found at the following link that should do the rest of the processing:

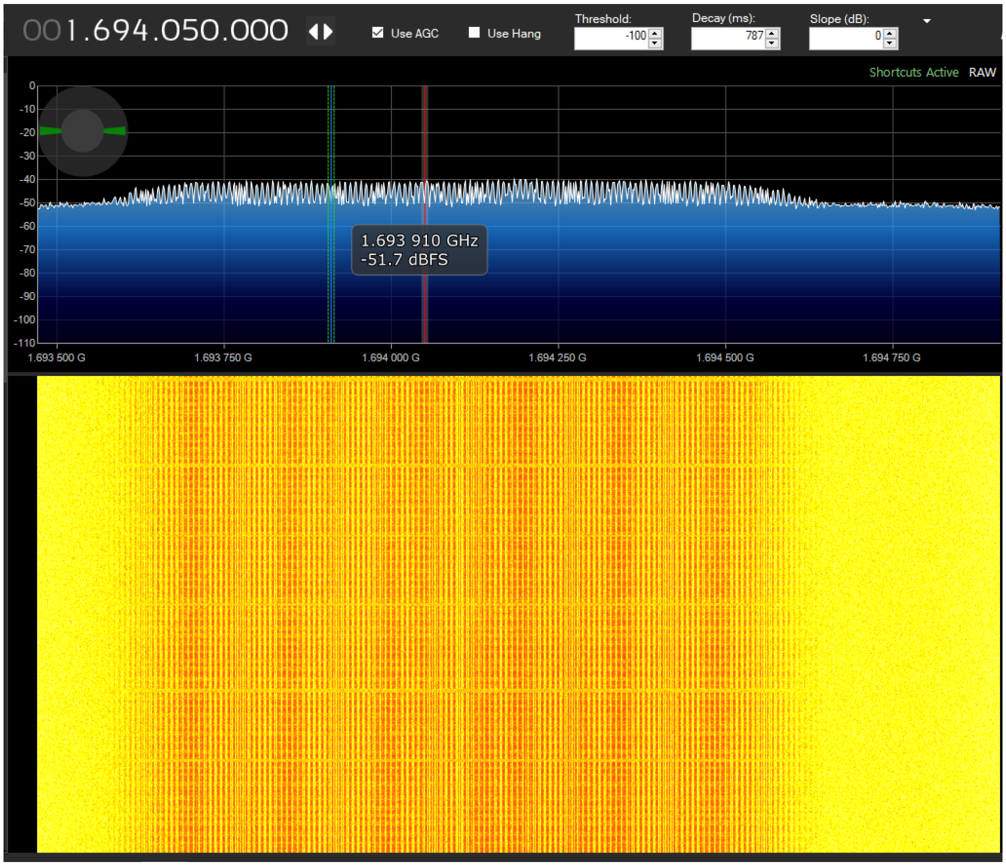
     https://github.com/racerxdl/open-satellite-project/tree/225a36d4144c0fe0704eb50a8fbc428914f654c0/GOES/standalone

     The scripts we tried using are the channeldecoder.py and packetmanager.py which is a helper script for channeldecoder.py. The expected input for these scripts is exactly the output of the rest of the program we have written up to this point. We were not able to get the scripts to work correctly and we ran out of time to figure out why it was not working.

6.6 Results

6.6.1 RF Testing

The antenna was tested using a personally owned portable spectrum analyzer, however the frequency resolution and the dependence on AC power for the spectrum analyzer limited testing to within the senior design lab. Results from this testing was poor and an alternate approach of connecting the antenna to a personally owned software defined radio and attempting to receive the satellite's signal was adopted. The results of this can be seen in figure 8 which shows the reception of the satellite's signal using the antenna.



001.694.050.000

☑ Use AGC

□ Use Hang

Threshold: -100

Decay (ms): 787

Slope (dB): 0

1.693 910 GHz
-51.7 dBFS

Figure 8: Antenna reception test using SDR radio and LNA

This image shows that the antenna is able to receive the signal with enough strength to make it out from the noise floor of the SDR. Testing of the receiver was only partially completed due to the finding of errata with the initial board design, and the replacement parts showing up just before spring break and the closure of the labs that followed. The receiver board was found to have been assembled with incorrect component values for the loop filter, this result in it being unable to produce the correct LO frequency. This can be seen in figures 9 and 10 showing the test setup and results.

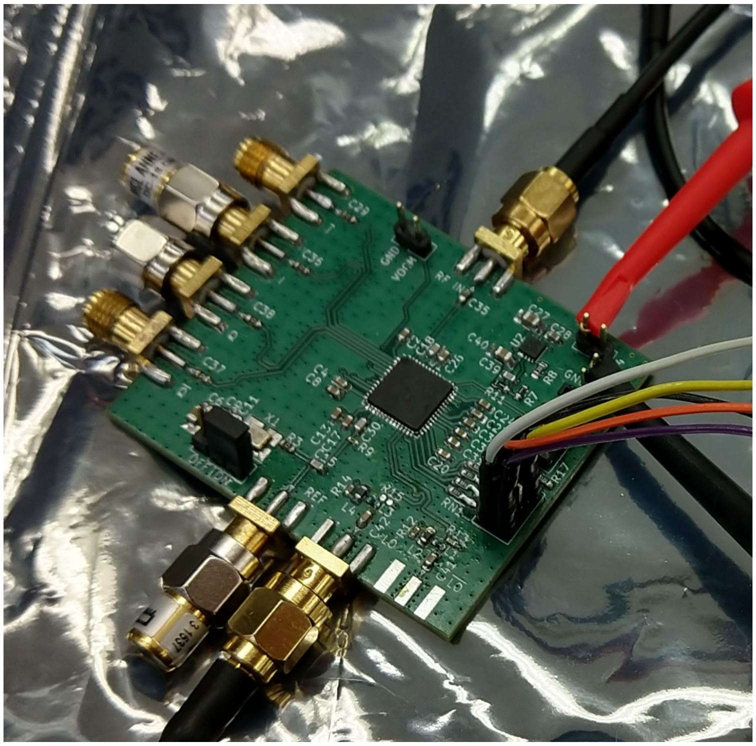


Figure 9: Receiver board and LO test connections

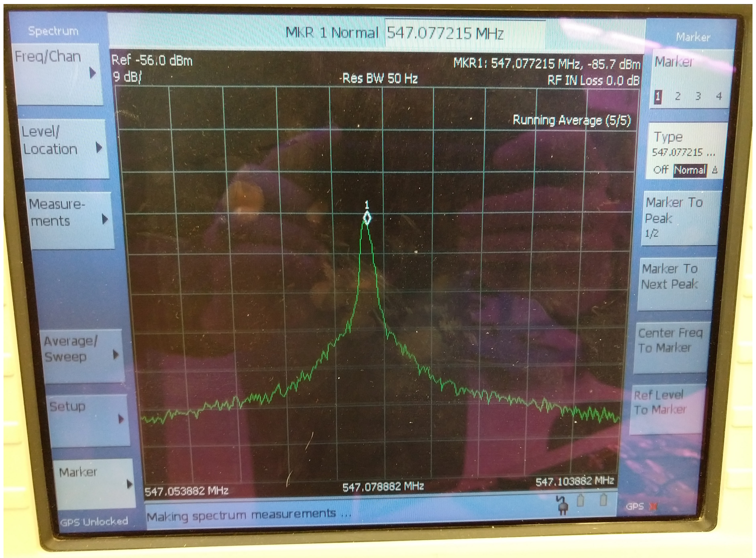


Figure 10: Spectrum of the receiver's LO output

This image shows the results of the receiver being programmed to output a frequency of 434.1 MHz but being unable to due to the correct loop filter components. Upon discovery of this, the correct components were ordered based upon simulations in ADISimPLL and the modifications were made to the board. At that time however spring break began, then all access to the labs was cutoff. Furthermore, this halted the sending of the LNB to be fabricated and as such no additional testing could be preformed.

6.6.2 ADC Testing

The various testing methods as described in Section 6.3 had the following results. Figures 11 and 12 show the oscilloscope outputs from the SPI μC test and RPi4 GPIO max toggle speed test. Testing for the ADC board was limited to simulation of the Sallen-Key filter as described in section 5.2.1.

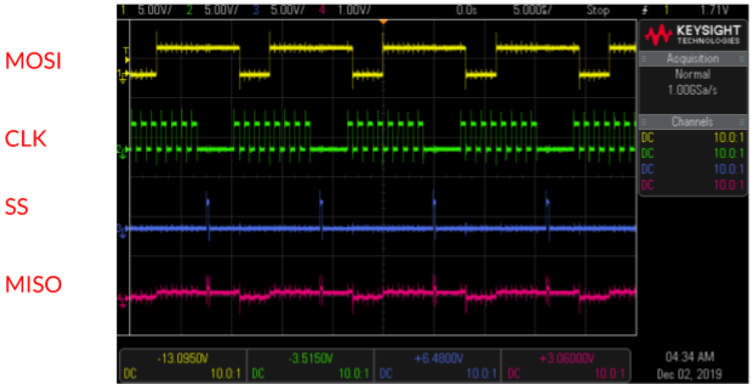


Figure 11: Example SPI output from 1st Semester

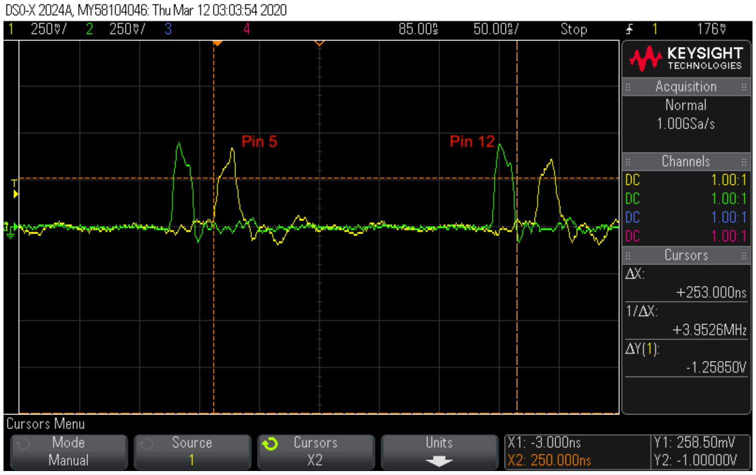


Figure 12: Max RPi4 GPIO toggle speed

7 Closing Material

7.1 Conclusion

As the project comes to a close, the team realizes that the project is not completed as was originally intended. The COVID-19 epidemic almost halted all hardware development, and made communication on software portions of the project much more difficult. As a result of this unfortunate circumstance, the team had to quickly find alternatives and change the scope of the project. Despite this large hurdle, the team still was able to complete a large portion of the project. Most of the hardware design was completed, the antenna was aimed and receiving a signal, and the software is nearing functionality. The team did not achieve the original goals laid out in the first semester, but performed well given the circumstances.

Despite the obstacles, the team was able to receive an image from the satellite through short-cuts, and has set up the framework for future team to come on and finish the project in an upcoming semester. To finish the project, the next team will need to perform tests on the hardware and software. Next, they'll have to integrate all the different system components and troubleshoot where necessary. The team believes this is very manageable to complete this project with one more senior design team.

7.2 References

Works Cited

Brettbolt, Brian Armstrong; Lucas Teske; "Libcorrect". *GitHub repository* (2019). https://github.com/quiet/libcorrect.

Razavi, B. *RF Microelectronics*. Prentice Hall, 2012. Web. Prentice Hall communications engineering and emerging technologies series.

Roberson, Jeremy, et al. "A Flexible Software Based EMWIN/HRIT Prototype Solution for the GOES-R Transition". *ATR-2010 Aerospace report* 2 (2009). Print.

Sklar, B. *Digital communications: fundamentals and applications*. Prentice-Hall PTR, 2001. Web. Prentice Hall Communications Engineering and Emerging Technologies Series.

Appendices

A Operation Manual

The following instructions describe how to set up the hardware and software for the complete system. Due to the change in the scope of the project, the instructions below describe how to set up a make-shift receiver to receive and image from the GOES-16 satellite.

1. Aim antenna at GOES-16 satellite to receive signal
   - From Coover Hall, the antenna can be aimed at 152.8° with a compass.
   - Elevation angle should be 37.9°.
   - Antenna skew angle is -19.3°.
2. Check signal strength with SDR
   - Connect an SDR to the antenna with an RF cable and connectors.
   - Use SDR on PC to confirm reception of signal at 1694.1 MHz.
   - Signal should look similar to that shown in figure 8
3. Install goesrecv software on RPi
   - goesrecv can be found in the goestools repository in GitHub.
   - Documentation can be found in the repository as well.
4. Connect antenna to RPi through SDR and run demodulation and decoding software
   - Run goesrecv command to begin demodulation and Viterbi decoding.
   - Run goesproc to begin processing of received data to transform into image.
   - Received image should show up in file path specified with goesproc command.

The instructions above were used to make a quick working prototype and receive the image from the GOES-16 satellite. The instructions above do not make use of most of our designed hardware and software, as we were not able to complete them properly.

B ADC

ADC BOM

Design Reference	**Customer Value**	**Footprint**	**DigiKey Part Number**	**Description**	**Quantity**	**Cost**
C1, C27	1μF	Capacitor_SMD:C_0805_2012Metric	490-10494-1-ND	Unpolarized capacitor	2	\$0.62
C2, C7, C11	10nF	Capacitor_SMD:C_0603_1608Metric	490-9666-1-ND	Unpolarized capacitor	3	\$0.66
C3, C8, C12, C14, C16, C18, C23, C24, C25, C26	0.1μF	Capacitor_SMD:C_0603_1608Metric	490-3285-1-ND	Unpolarized capacitor	10	\$2.08
C4, C5, C6, C9, C10, C13, C15, C17	10μF	Capacitor_SMD:C_0805_2012Metric	490-14381-1-ND	Unpolarized capacitor	8	\$2.00
C19, C20, C21, C22	100pF	Capacitor_SMD:C_0805_2012Metric	490-5542-1-ND	Unpolarized capacitor	4	\$1.44
FB1	Ferrite_Bead	Resistor_SMD:R_0805_2012Metric	587-1932-1-ND	Ferrite bead	1	\$0.10
F1	Polyfuse	Fuse:Fuse_Bourns_MF-SM_7.98x5.44mm	SMD200F/24-2920-2CT-ND	Resettable fuse, polymeric positive temperature coef	1	\$0.65
J1	Barrel_Jack	Connector_BarrelJack:BarrelJack_CUI_PJ-102AH_Horizontal	CP-102B-ND	DC Barrel Jack	1	\$0.64
J2	ADC_Out	Connector_PinHeader_2.54mm:PinHeader_2x10_P2.54mm_Vertical	SAM90020ND	GEneric connector, double row, 02x10	1	\$1.35
J3, J5	Conn_Coaxial	Connector_Coaxial:SMA_Samtec_SMA-J-P-X-ST-EM1_EdgeMount	SAM8857-ND	coaxial connector (BNC, SMA, SMB, SMC, Cinch/RCA, ...)	2	\$8.66
R1	2Ω	Resistor_SMD:R_0805_2012Metric	2019-RK73H2ATTD2R00FCT-ND	Resistor	1	\$0.10
R2	3.57kΩ	Resistor_SMD:R_0603_1608Metric	RNCF0603DTE3K57CT-ND	Resistor	1	\$0.14
R3	1.15kΩ	Resistor_SMD:R_0402_1005Metric	RNCF0603DTE1K15CT-ND	Resistor	1	\$0.14
R4	10.5kΩ	Resistor_SMD:R_0603_1608Metric	RNCF0603DTE10K5CT-ND	Resistor	1	\$0.14

Continuation of ADC BOM

Design Reference	**Customer Value**	**Footprint**	**DigiKey Part Number**	**Description**	**Quantity**	**Cost**
R5	2kΩ	Resistor_SMD:R_0603_1608Metric	RNCF0603DTE2K00CT-ND	Resistor	1	$0.14
R6, R7, R8, R9, R12, R13	1kΩ	Resistor_SMD:R_0402_1005Metric	RNCF0402DTE1K00	Resistor	6	$0.78
RN1, RN2, RN3	330Ω	Resistor_SMD:R_Array_Convex_4x0603	Y9331CT-ND	4 resistor network, parallel topology, DIP package	3	$0.30
U1	AD8137	Package_CSP:LFCSP-8-1EP_3x3mm_P0.5mm_EP1.45x1.74mm	AD8137WYCPZ-R7CT-ND	Differential Amplifier 1 Circuit Differential	1	$3.05
U2	TPS7A88	Package_DFN_QFN:QFN-20-1EP_4x4mm_P0.5mm_EP2.7x2.7mm	296-43312-1-ND	Linear Voltage Regulator IC 2 Output 1A 20-QFN (4x4)	1	$3.51
U3	SN74LVC8427A	Package_SO:SSOP-24_5.3x8.2mm_P0.65mm	296-8548-1-ND	Buffer, Non-Inverting	1	$1.19
U4	MAX1426	Package_SO:SSOP-28_5.3x10.2mm_P0.65mm	MAX1426EAI+-ND	10 Bit Analog to Digital Converter	1	$4.91
X1	10 MHz	Oscillator:Oscillator_SMD_Abracon_ASE-4Pin_3.2x2.5mm	XC2253CT-ND	3.3V CMOS SMD Crystal Clock Oscillator, Abracon	1	$3.10
	PWR SUPPLY		2368-57-5D-2000-5-ND	AC-DC 5V 2A 2.5MM PLUG	1	$15.80
	OSH Park PCB			Given cost for 3 boards	3	$18.20
				Approximate cost per board	**1**	**$57.57**

Table 3: ADC board Bill of Materials (BOM)

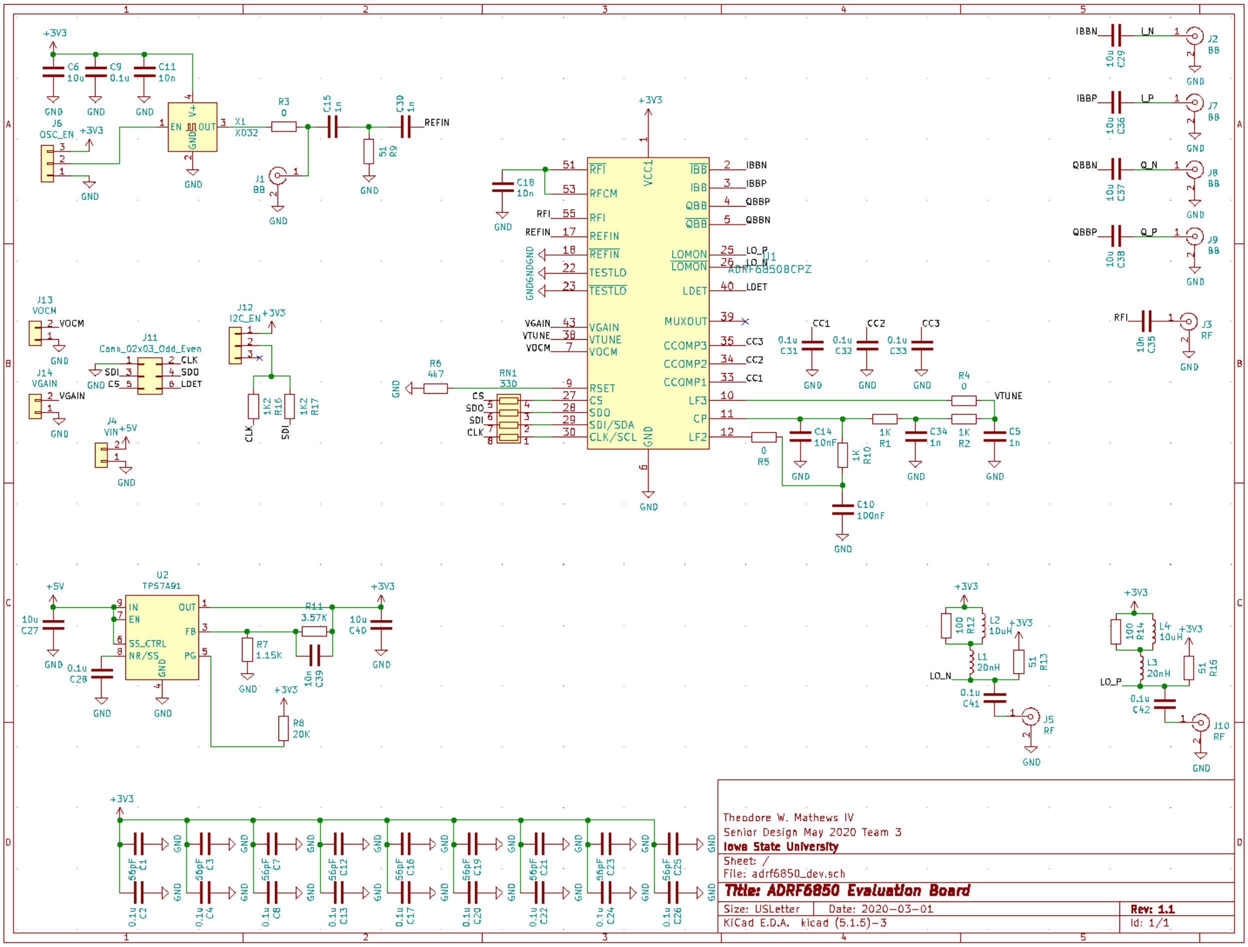


Figure 13: Schematic for the Receiver board

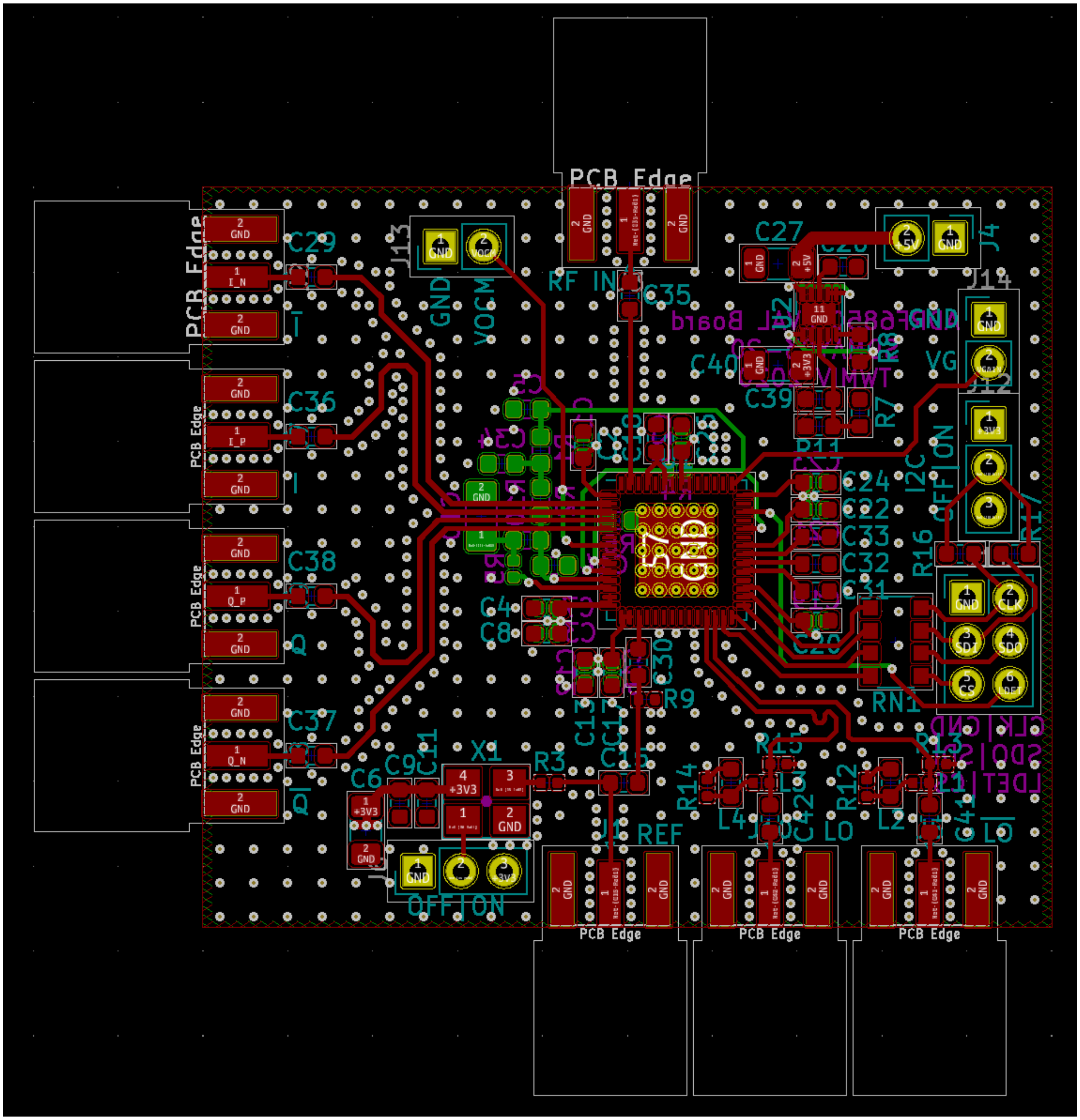


Figure 14: Layout for the Receiver board

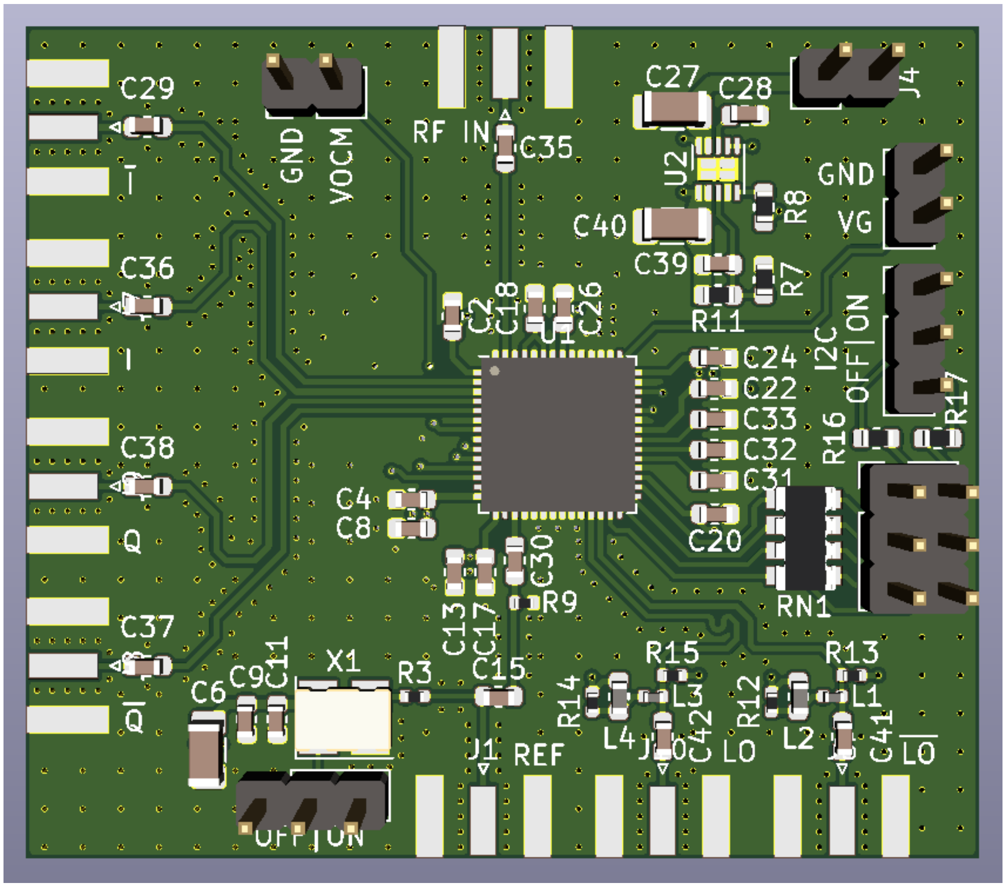


Figure 15: 3D render of the front of the Receiver board

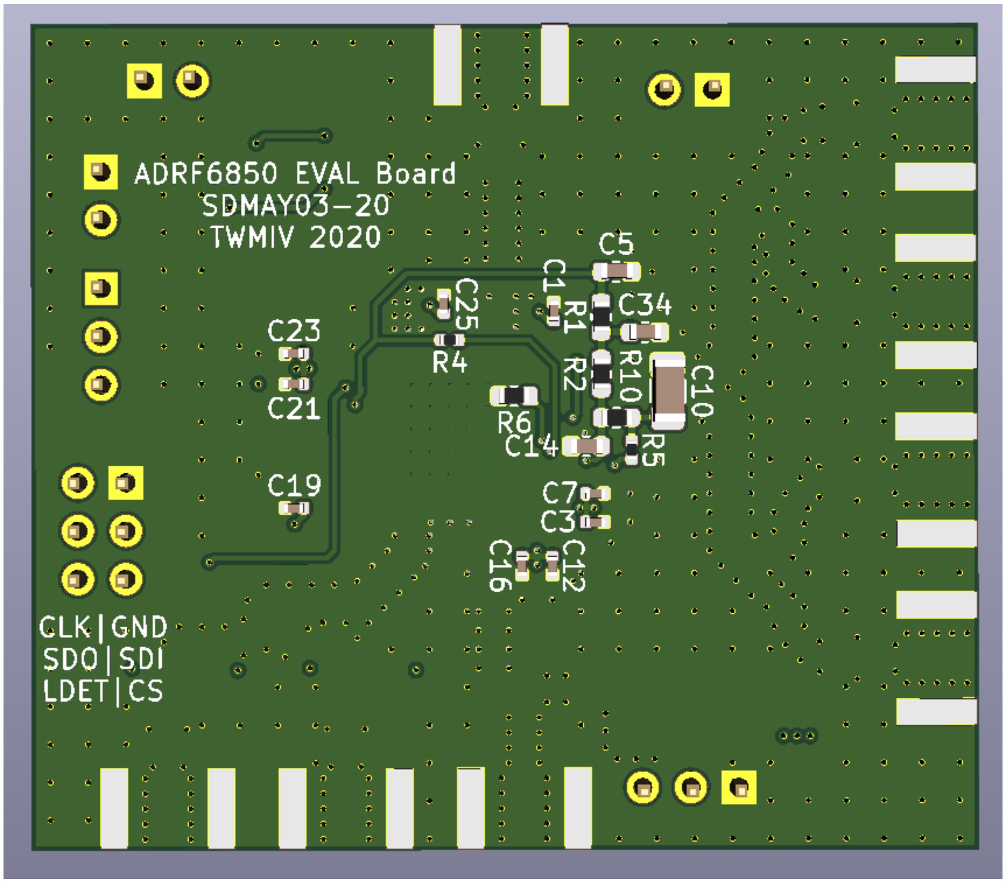


Figure 16: 3D render of the back of the Receiver board

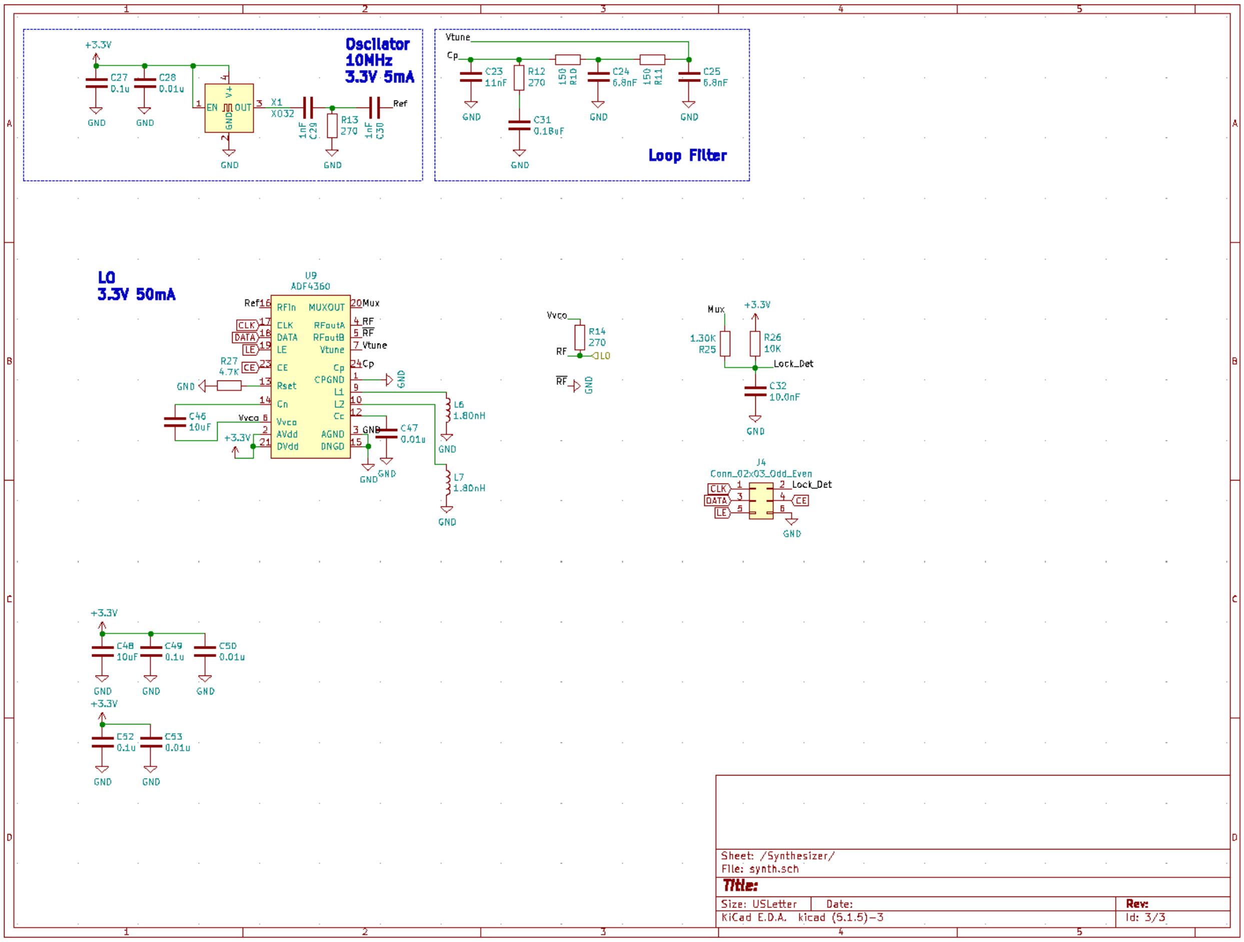


Figure 17: Schematic for the synthesizer section of the LNB

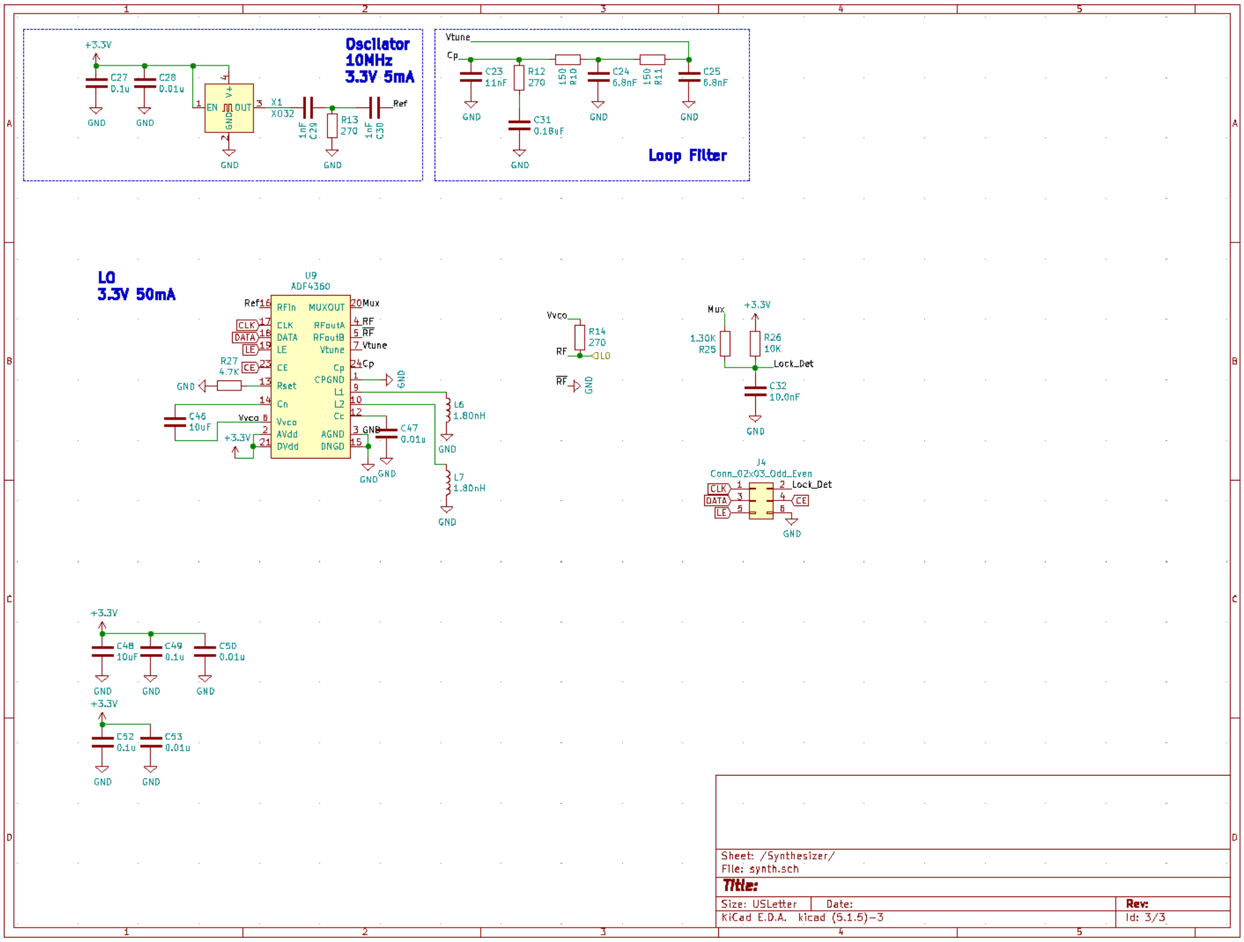


Figure 18: Schematic for the synthesizer section of the LNB

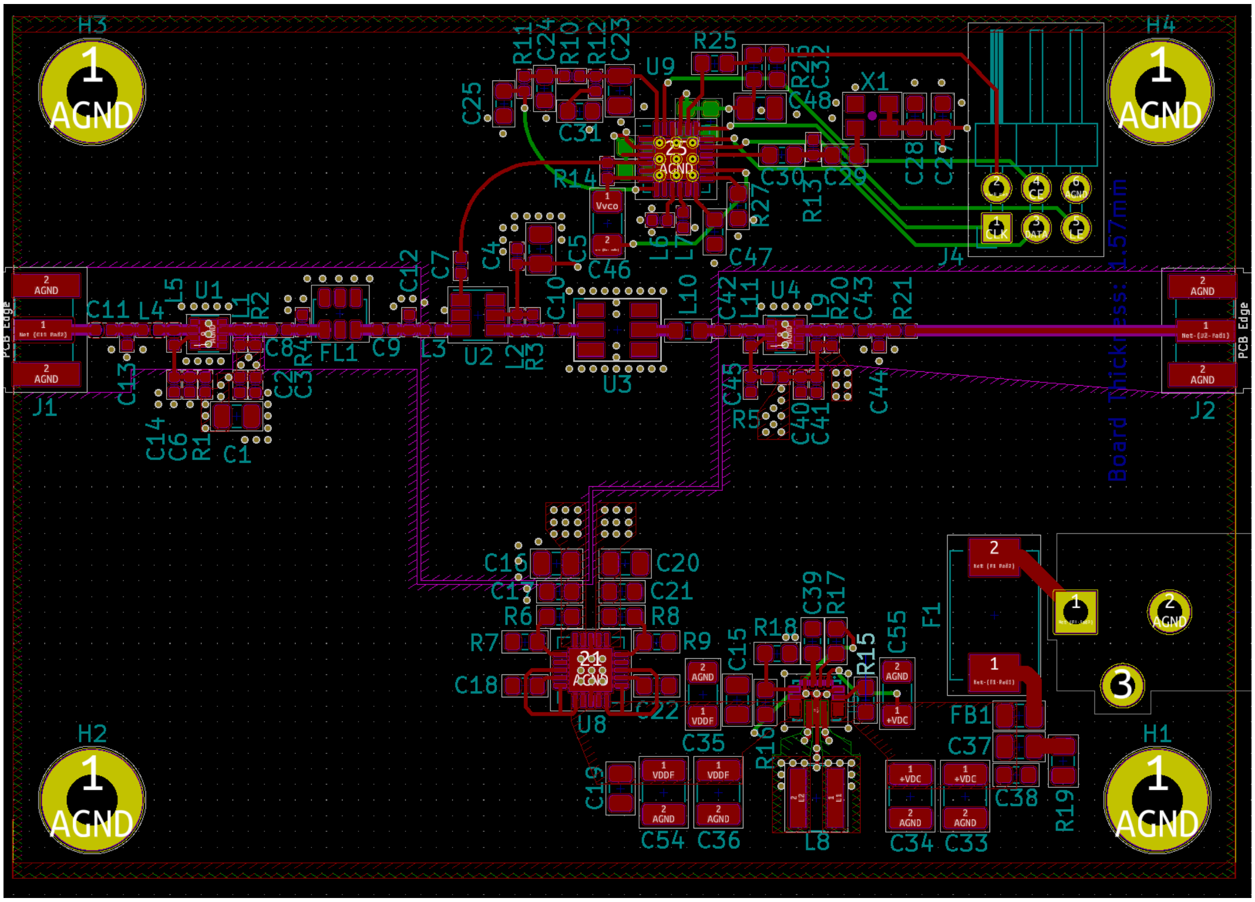


Figure 19: Layout for the LNB board

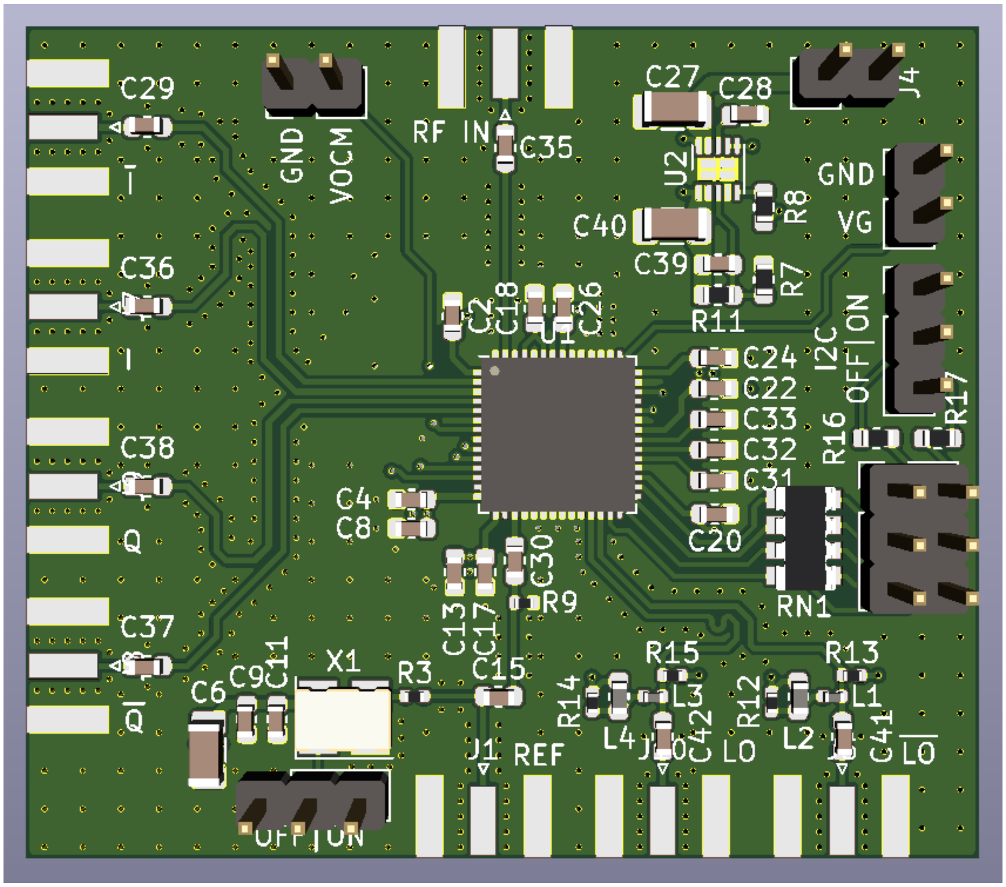


Figure 20: 3D render of the front of the Receiver board

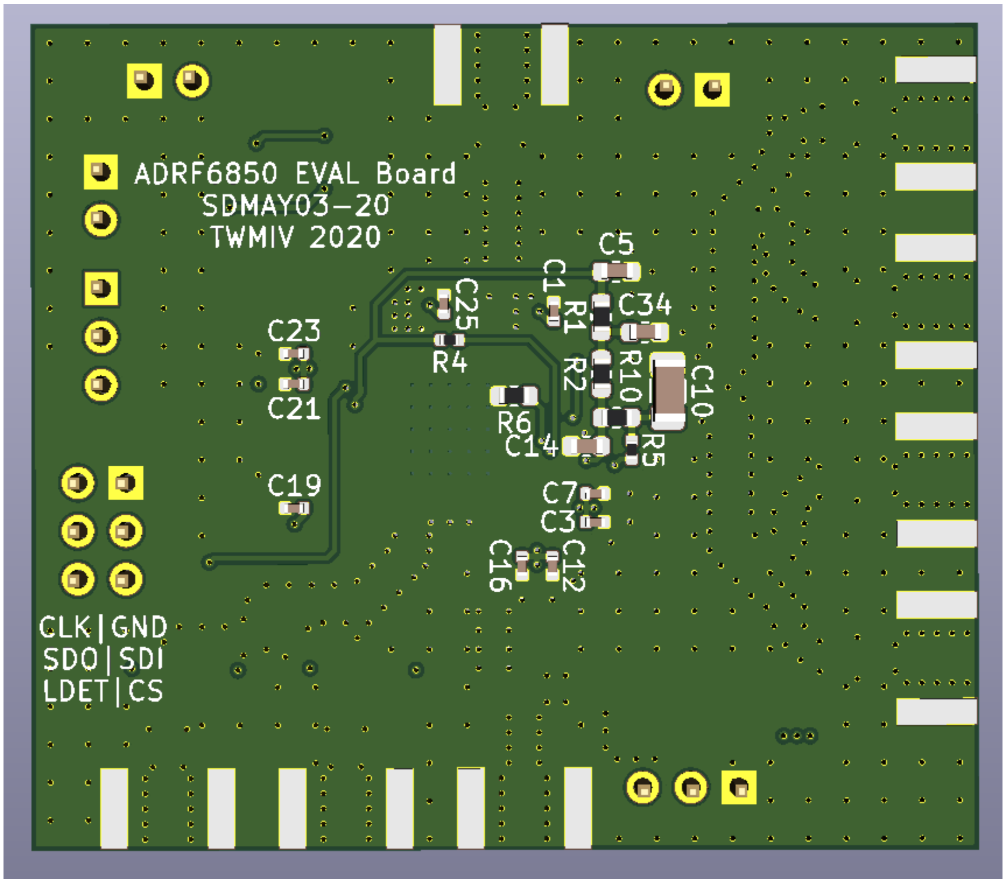
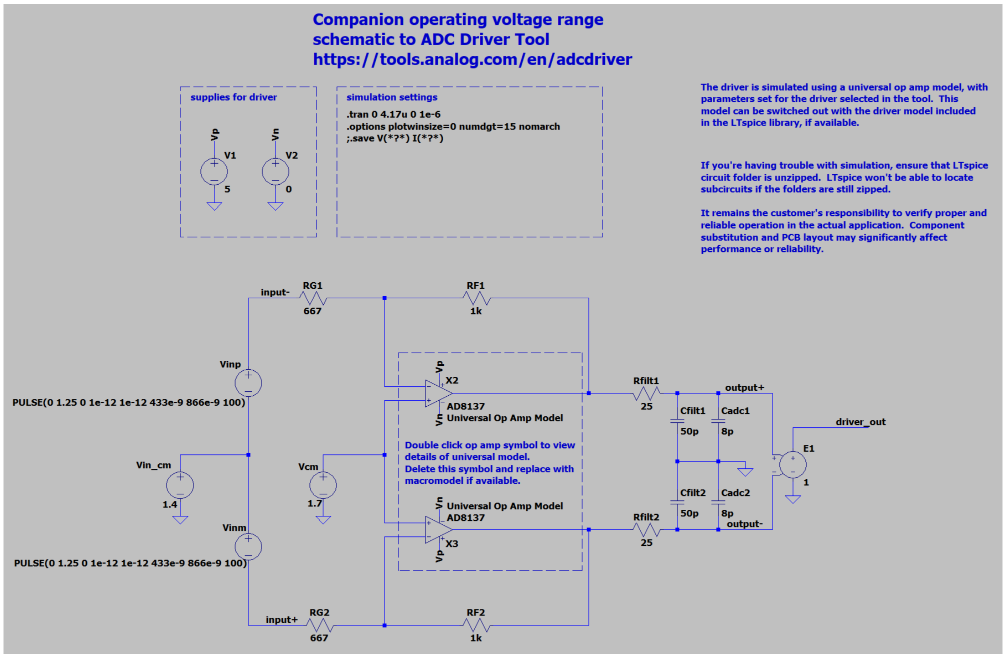


Figure 21: 3D render of the back of the Receiver board


Companion operating voltage range
schematic to ADC Driver Tool
https://tools.analog.com/en/adcdriver

supplies for driver

simulation settings

```
.tran 0 4.17u 0 1e-6
.options plotwinsize=0 numdgt=15 nomarch
;.save V(*?*) I(*?*)
```

The driver is simulated using a universal op amp model, with parameters set for the driver selected in the tool. This model can be switched out with the driver model included in the LTspice library, if available.

If you're having trouble with simulation, ensure that LTspice circuit folder is unzipped. LTspice won't be able to locate subcircuits if the folders are still zipped.

It remains the customer's responsibility to verify proper and reliable operation in the actual application. Component substitution and PCB layout may significantly affect performance or reliability.

Double click op amp symbol to view details of universal model.
Delete this symbol and replace with macromodel if available.

Figure 22: Sallen-Key filter generated using Analog Devices filter tool

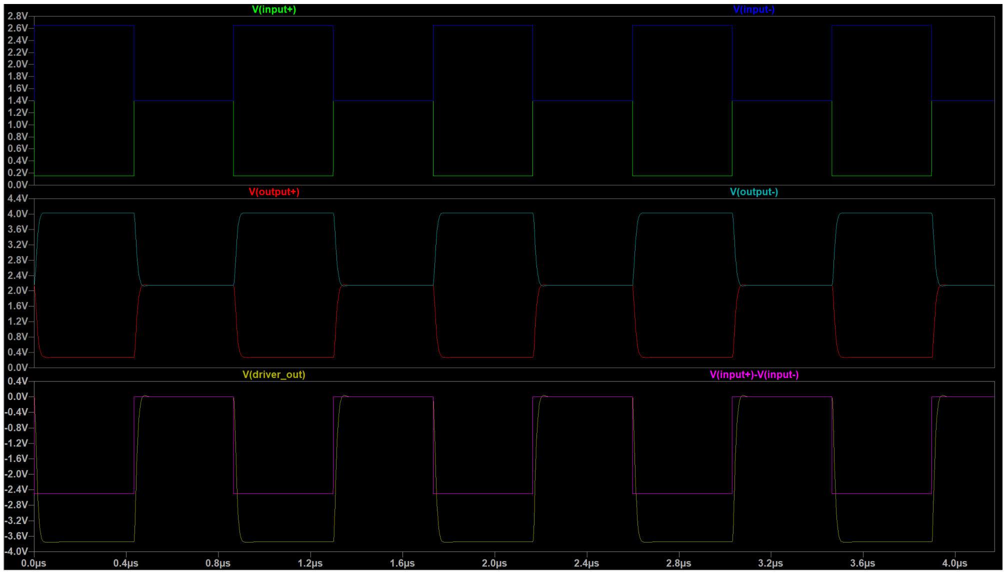


Figure 23: Spice simulation results of the Sallen-Key filter

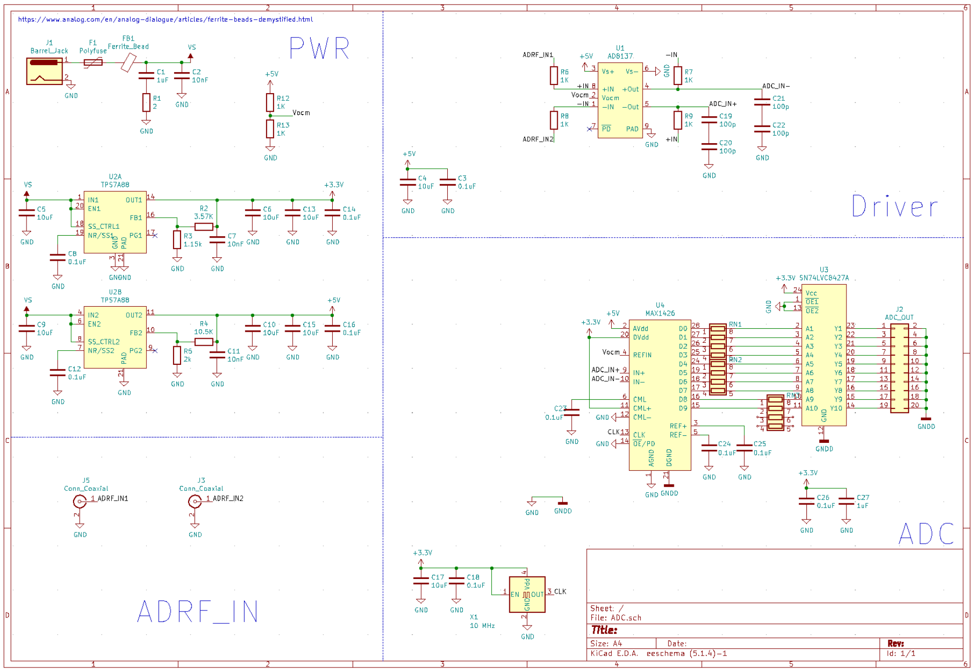


Figure 24: ADC board schematic

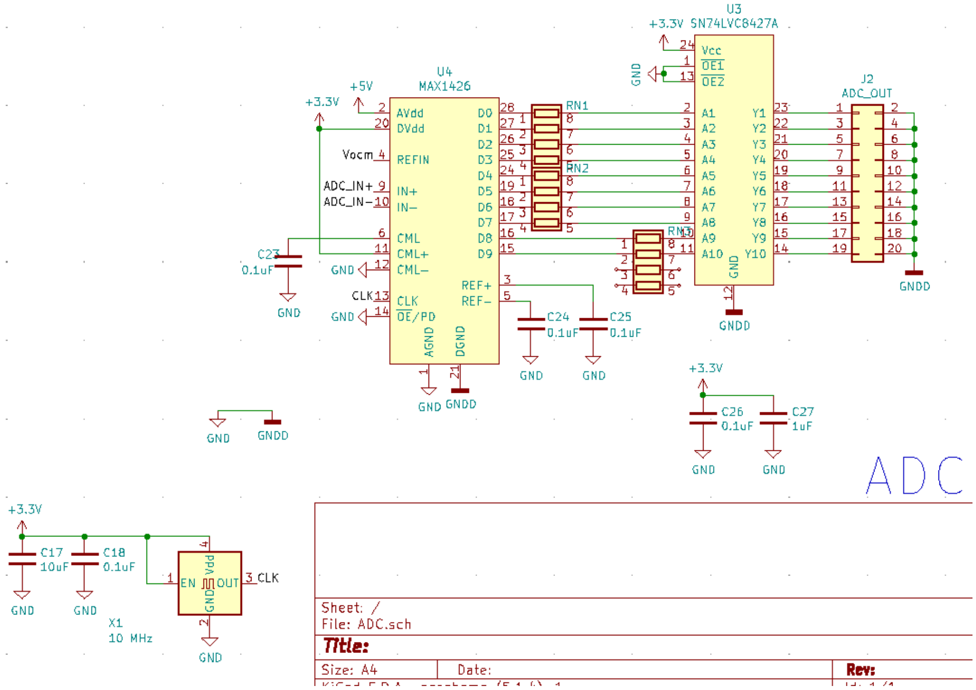


Figure 25: ADC and output driver schematic

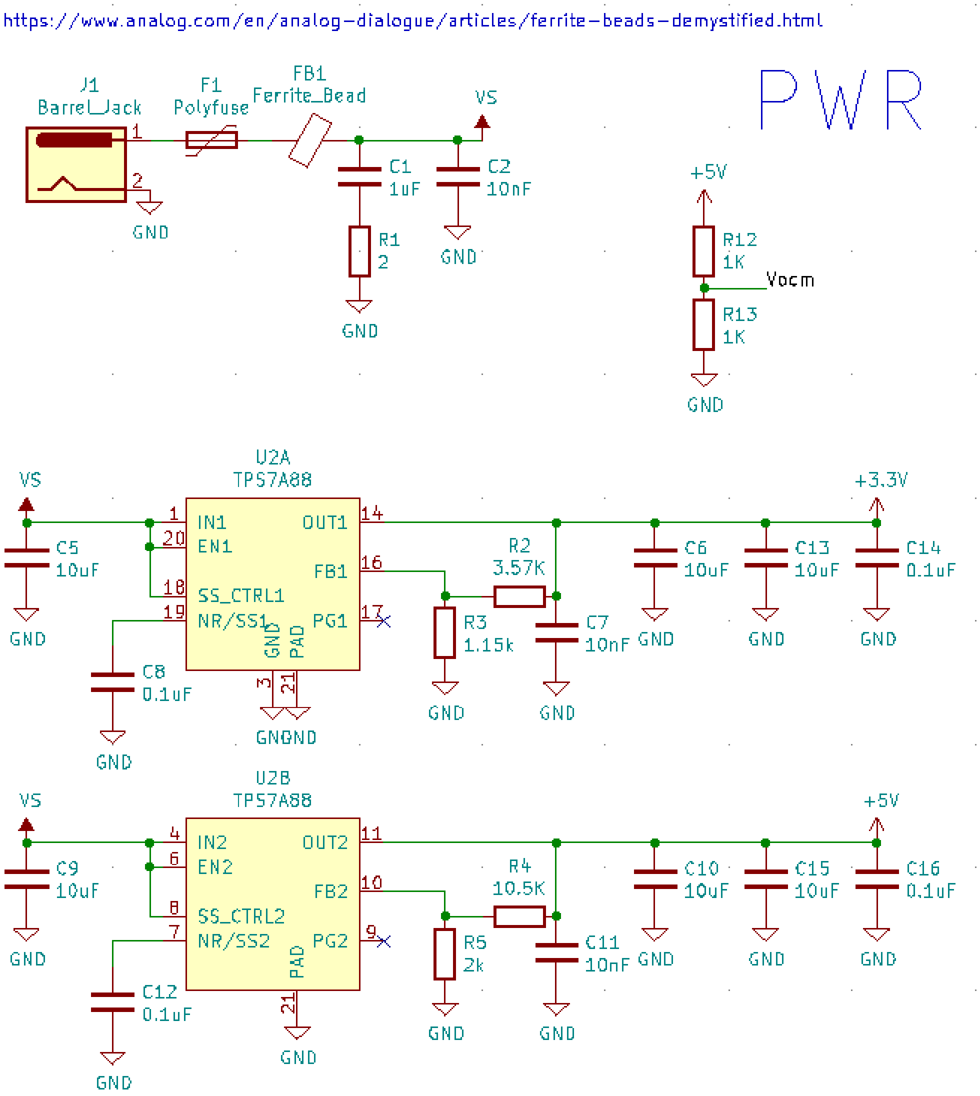


Figure 26: Schematic of the power section for the ADC

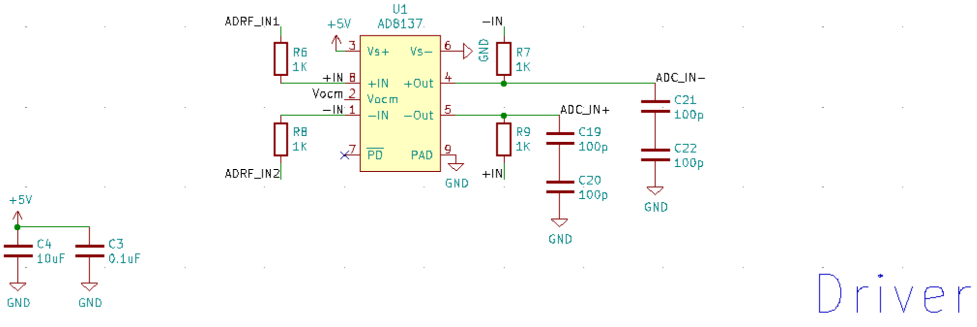


Figure 27: ADC input driver schematic

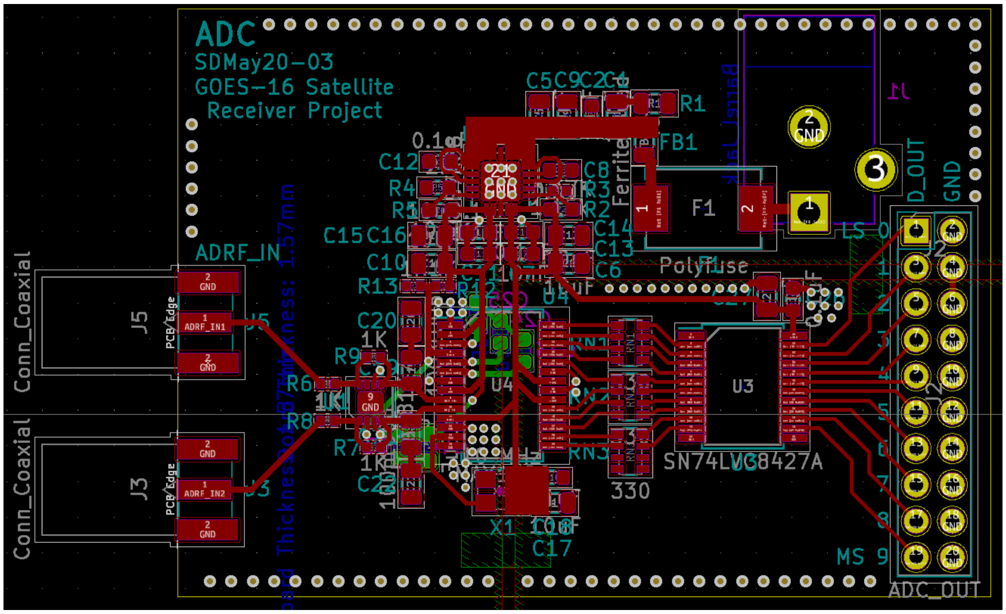


Figure 28: ADC layout

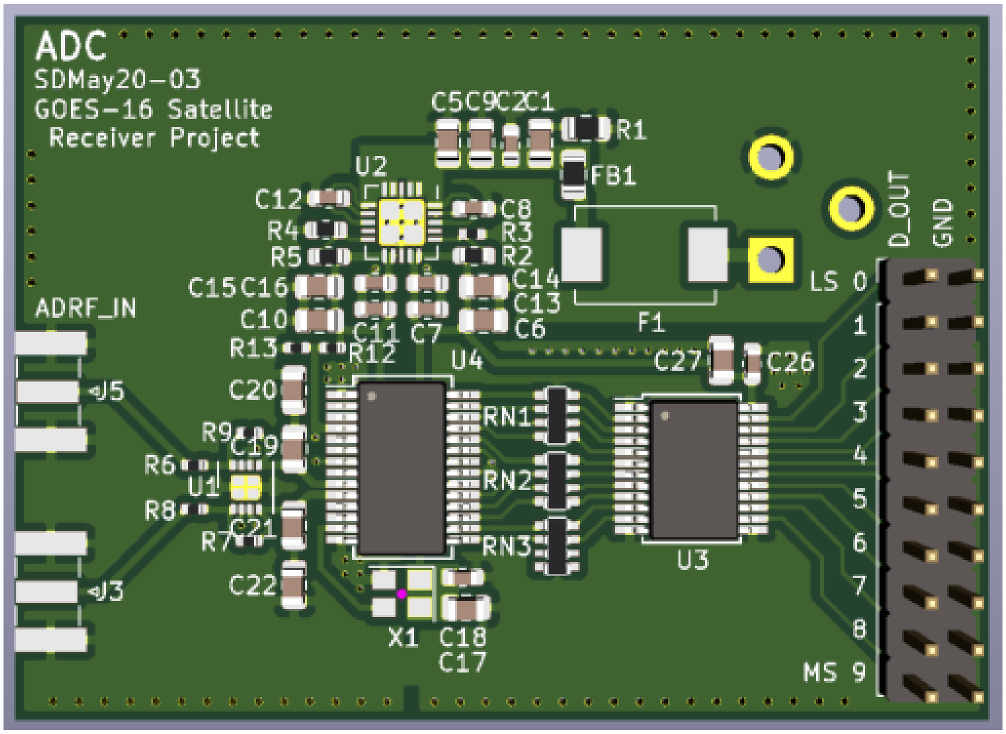


Figure 29: Rendered 3D view of the front of the ADC board

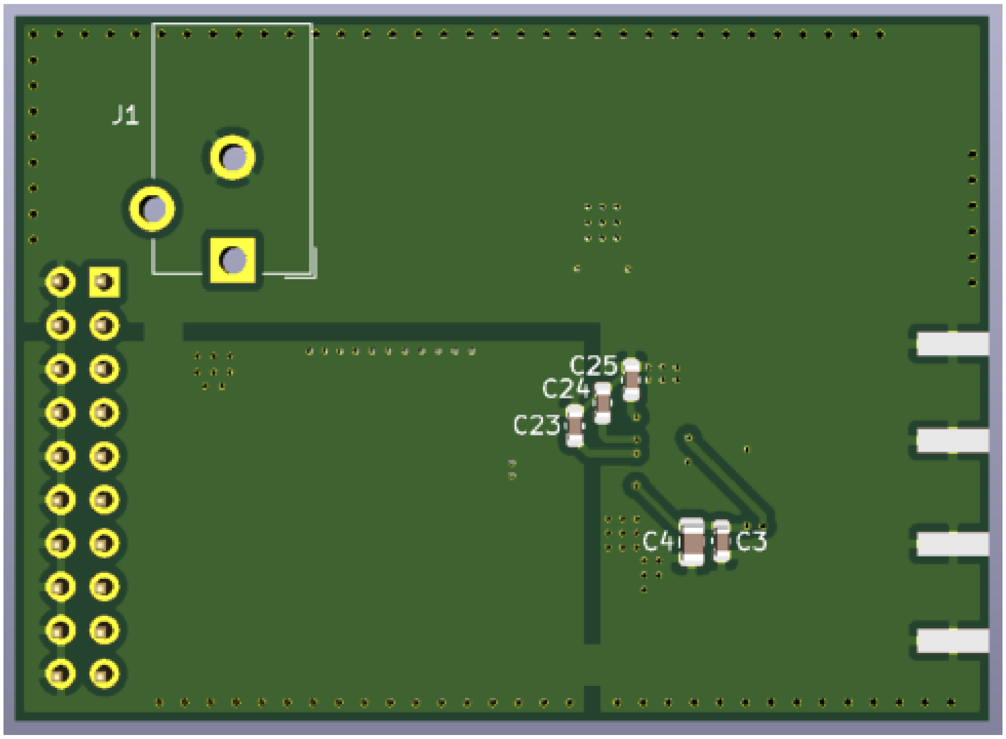


Figure 30: Rendered 3D view of the back of the ADC board