GOES-16 Satellite Receiver Final Presentation

SDmay20-03
Adviser/client: Nathan Neihart
Theodore Mathews IV, Jonathan Massner, Riley Stuart, Jordan Tillotson, Nicholas Butts, Yong Lim

Problem Statement

The goal of this project is to receive, decode, and display weather products from NOAA's GOES-16 weather satellite.

This project involves:

- Receiving and demodulating the analog signal (RF team)
- Digitizing the signal and transferring the data to Raspberry Pi 4 (ADC team)
- Decoding the data and image construction (software team)

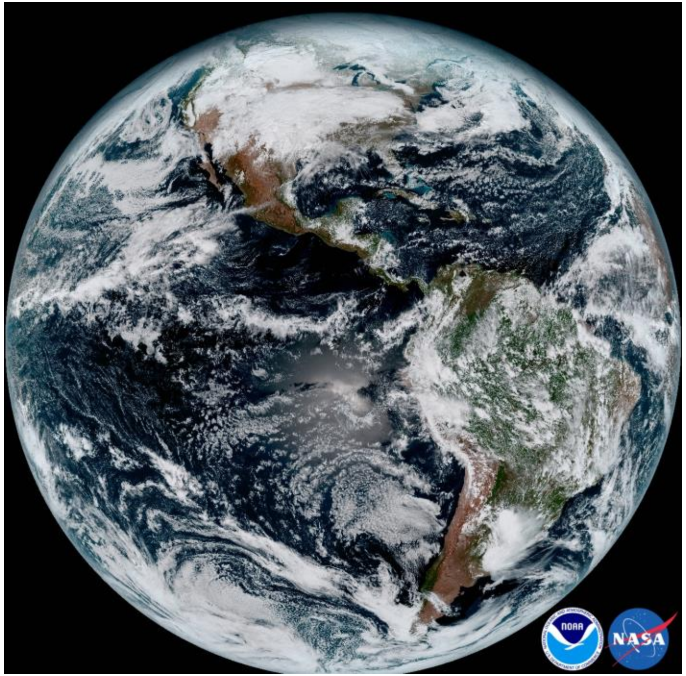
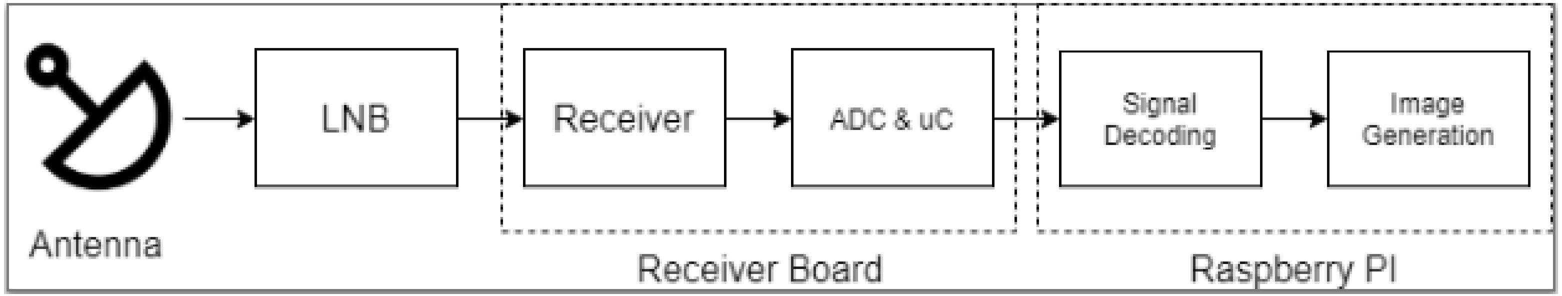
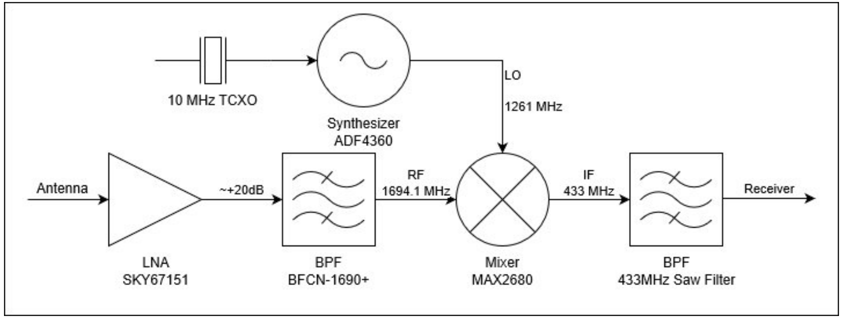


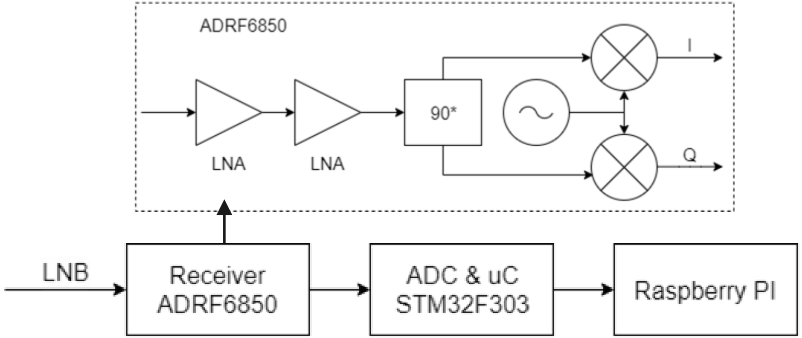
Image credit: NOAA

Conceptual Sketch



Detailed Design





LNB

- Input : 1694.1 MHz from antenna
- Output: 433 MHz downmixed and filtered signal

Receiver and Microcontroller

- Input: 433 MHz from LNB
- ADC: 4x 12-bit 5 MSPS inside µC
- µC: Integrated usb and small DSP
- RPI: Runs Open Satellite Project tools

Requirements

Functional

- The system must download a current Earth image from the GOES-16 satellite
- RPi4 will host a website to display information about the downloading process and full images

Non-functional

- RF section:
  - System must receive 1694.1MHz signal with at least 1.205MHz of bandwidth from NOAA's satellite and output a baseband signal

- ADC section:
  - System must digitize the baseband signal and output received binary to Raspberry Pi
- Software section:
  - Software must receive the binary information from the ADC, decode the information, and generate an image file

Technical Considerations

Receiver architecture takes advantage a mix of SDR and superheterodyne strengths

uC capabilities

- Data rate
- Resolution
- ADC sampling rate
- Communication

Software must compile and run on Raspbian

Computation limitations of RPi4

Market Survey

This project is geared more for hobbyists

- Information transmitted from GOES-16 is available for public consumption

Our project is not a competing product

- Components to build similar projects are available for purchase off-the-shelf
- Attempting to build a version different from similar existing projects

The RF implementation is the primary component of the project that is unique

- ADCs are standard for this type of project
- Software is open-source

Potential Risks and Mitigation

Completion Risks

- Risk to hardware exists by improper connection and powering of system components.
- Signal strength matching - ensuring that the boosted signal is within tolerance for other system components.

Physical and Environmental Risks

- Low - Physical risk present when improperly transporting equipment such as the antenna.

Risk Mitigation

- Meticulously integrate systems to prevent damage to components
- Through calculations, testing, and datasheet analysis, ensure signal strength levels are appropriate
- Properly store and transport the system within Coover.

Resource/Cost Estimate

- Most of the costs are for hardware
    - Antenna
    - RF Boards, Components, RF Cables, and Connectors
    - Raspberry Pi 4
- Software is open-source
    - OpenSat project (Teske)
- Total cost estimate is approximately
    - 700$

Milestones

RF team

- Layout, fabricate, assemble, and test first board prototypes
- Begin work on second revision and design LNB enclosure

ADC team

- Communicate to all of the necessary components - microcontroller, Raspberry Pi 4, IQ demodulator, PLL
- Digitize output of RF section

Software team

- Compile code from Open Satellite Project repository on Raspberry Pi 4
- Get RPi-hosted website operational

Test Plan

Functional

- Test each subsystem for proper operation
  - RF system output is an analog demodulated signal
  - ADC team output is raw binary data
  - Software team output is processed image

Non-functional

- Image quality is the best measure of how well the system functions

Hardware Platforms Used

Prototyping

- STM32F303 Nucleo (microcontroller development board)
- Raspberry Pi3/Pi4
- ADRF6850 I/Q Demodulator development board

Final Implementation

- ADRF6850 I/Q Demodulator
- LCOM 2.4GHz 24 dBi Dish Antenna
- SKY67151 LNA
- BFCN-1690 BPF
- MAX2680 Mixer
- ADF4360 Synthesizer
- STM32F303 Microcontroller (pending circuit design)
- Raspberry Pi4

LCOM 2.4GHz 24 dBi Dish

ADRF685
0

STM32F303 Nucleo

Raspberry Pi4

Software Platforms Used

STM32CubeMX IDE

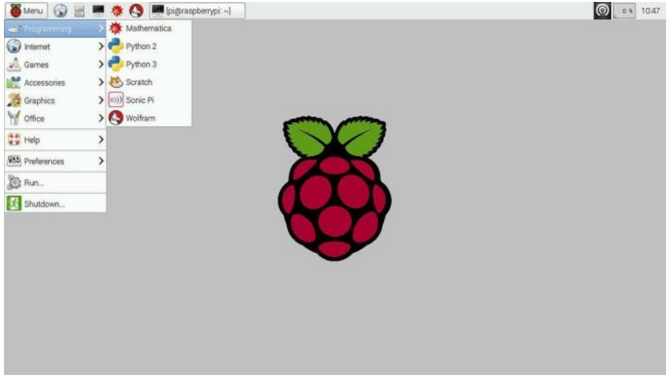
- Used to initialize the STM32F303 uC

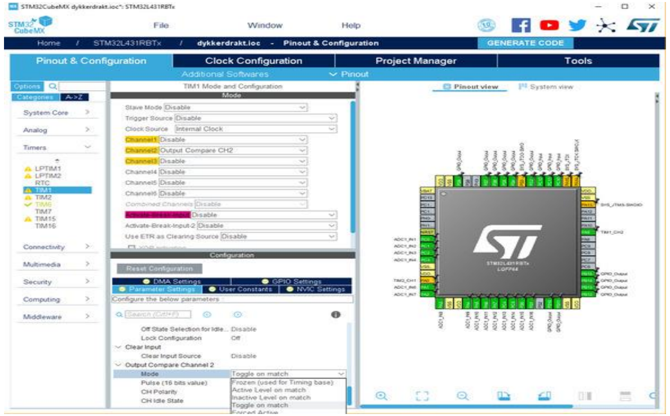
Mbed IDE

- Platform for creating and compiling firmware to be used on uC

Raspbian OS

- Operating system for Raspberry Pi4





Project Status

RF Team

- Antenna assembled and ready for modification
- Schematics made for RF boards (LNB, ADRF6850, and microcontroller interface)
- RF board layout in-progress

ADC Team

- STM32F303 uC configured for ADCs, and SPI communication,
- Interfacing uC to Raspberry Pi4

Software Team

- Working on compiling code from the OpenSatProject
- Trouble with executing linux commands as stated in compiling guide.

Team Member Contributions

Ted Mathews - RF Design and Testing

- SME, All system component recommendations, Antenna tripod build, Antenna assembly and modification, RF circuit design and layout

Jonathan Massner - RF Design and Testing, System Design

- Antenna assembly, RF circuit design and layout, system integration, managerial tasks

Riley Stuart - ADC Design and Testing

- STM32F303 Nucleo communication with I/Q demodulator, microcontroller initialization, ADC circuit design and layout (Spring 2020), system integration

Jordan Tillotson - ADC Design and Testing

- STM32F303 Nucleo communication with Raspberry Pi4, ADC circuit design and layout (Spring 2020)

Nick Butts - Software/DSP Development

- Setup of Raspberry Pi4 OS, implementation of OpenSat project material to our project

Yong Yi Lim - Software/DSP Development

- Setup of Raspberry Pi4 OS, implementation of OpenSat project material to our project

Plan for next semester

(subject to minor changes)

RF Team

- February 14 - Working RF hardware prototype including filtering, amplifying, and demodulation
- March 1 - RF circuit design completed

ADC Team

- January 19 - functioning ADC code (arbitrary signals similar to actual signals)
- February 25 - Tested and working ADC code
- March 1 - ADC circuit design completed

Software Team

- January 13 - OpenSat code compiled and working properly
- February 3 - Test demodulator and decoder for correct operation
- March 2 - Raspberry Pi4 hosted website operational

System

- March 10 - PCB designs complete and ordered for RF and ADC circuits
- April 1 - Operational system constructed with final prototype
- April 22 - Reviewed hardware and software documentation released

Works Cited

Teske, Lucas. “Open Satellite Project.” *GitHub*, github.com/opensatelliteproject.