# S-Band Radar Altimeter

ECE 445

Elliot Rubin, Rayan Nehme, Bobby Sommers

April 30, 2024

# Team Members

Elliot Rubin

Bobby Sommers

Rayan Nehme

# Objectives and Goals

Consumer drones rely on GPS or IR sensing for navigation or terrain avoidance

- GPS and IR are not the most efficient or reliable (poor performance in urban environments/indoors)

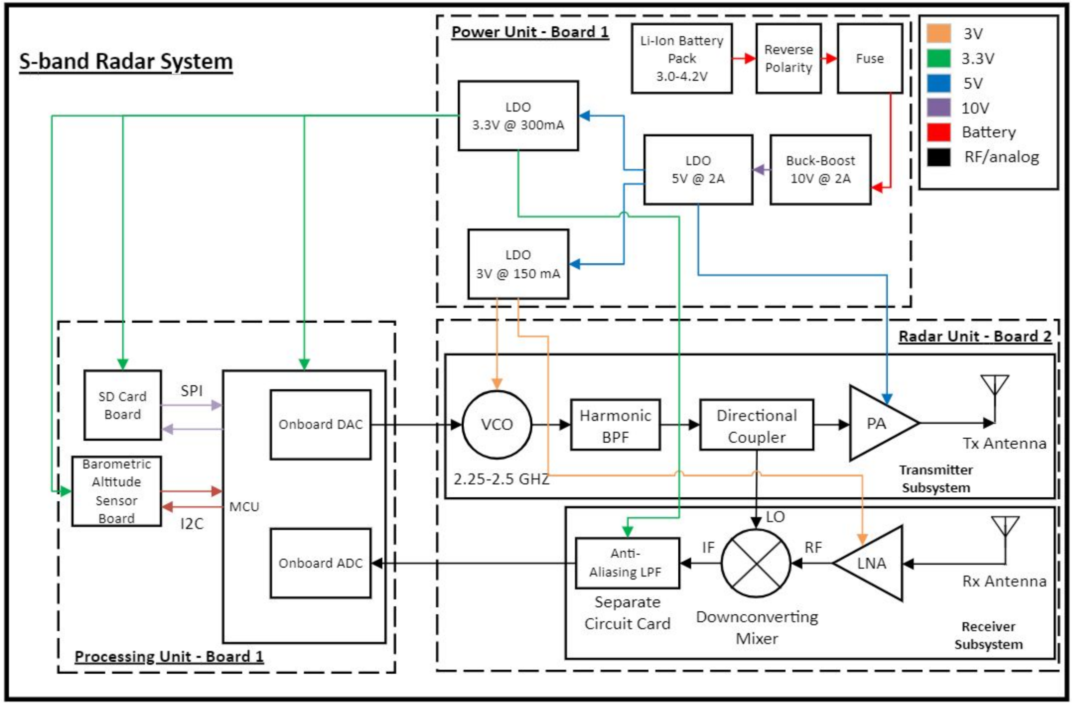
Solution is a cost effective S band radar altimeter which can be mounted on consumer drones

- FMCW radar architecture is well-documented and inexpensive to build
- Distance and velocity/Doppler measurements

High-level requirements:

- Minimum range > 20m
- Receiver noise figure < 10 dB
- Range resolution better than 1.5m

# Block Design



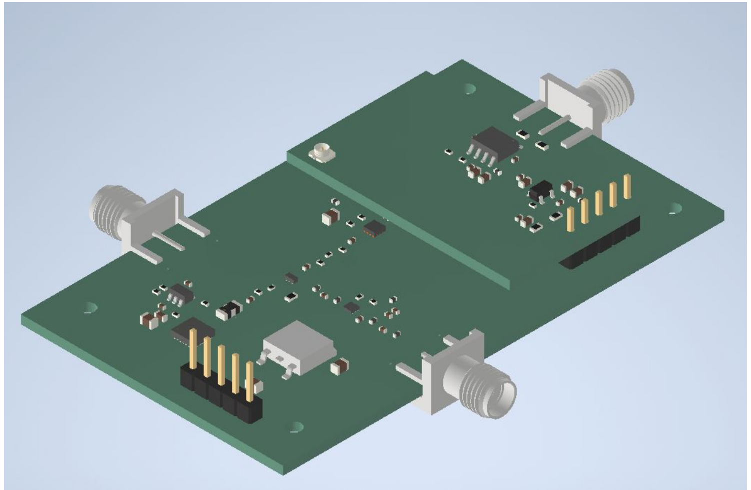

## Whole system partitioned into 3 subsystems

Radar unit
- All analog/RF xceiver hardware, Tx/Rx antennas

Power unit
- Power protection, power bus generation

Processing unit
- Digital processing and data conversion hardware
- MCU, ADC/DAC, storage

# Block Overview: Radar Unit



Radar transceiver with Tx/Rx chains and homemade antennas

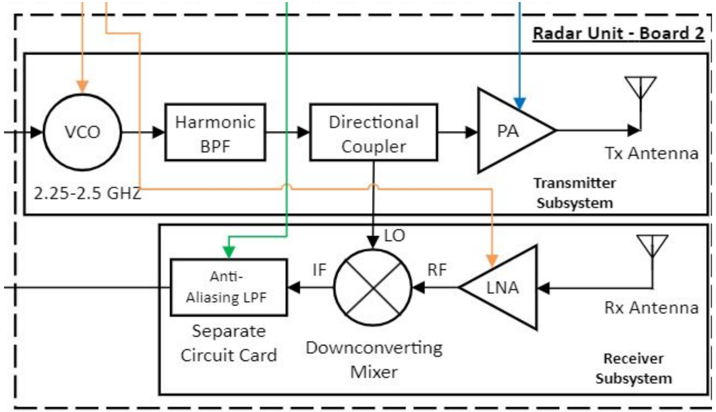
LO, amplifiers, mixers integrated onto one board

Baseband filtering achieved by second circuit card

Requirements:

- < 2W power consumption
- VCO second harmonic < -20 dBc
- PA/LNA stable ($\mu$ > 1) across operating band, 2250 - 2500 MHz

# Changes: Radar Unit




## Changes to radar unit were motivated by issues finding parts and poor RF performance/ease of testing

LNA/PA oscillation necessitated redesign

- Destruction of parts -> extra $

Directional coupler was expensive/difficult to solder

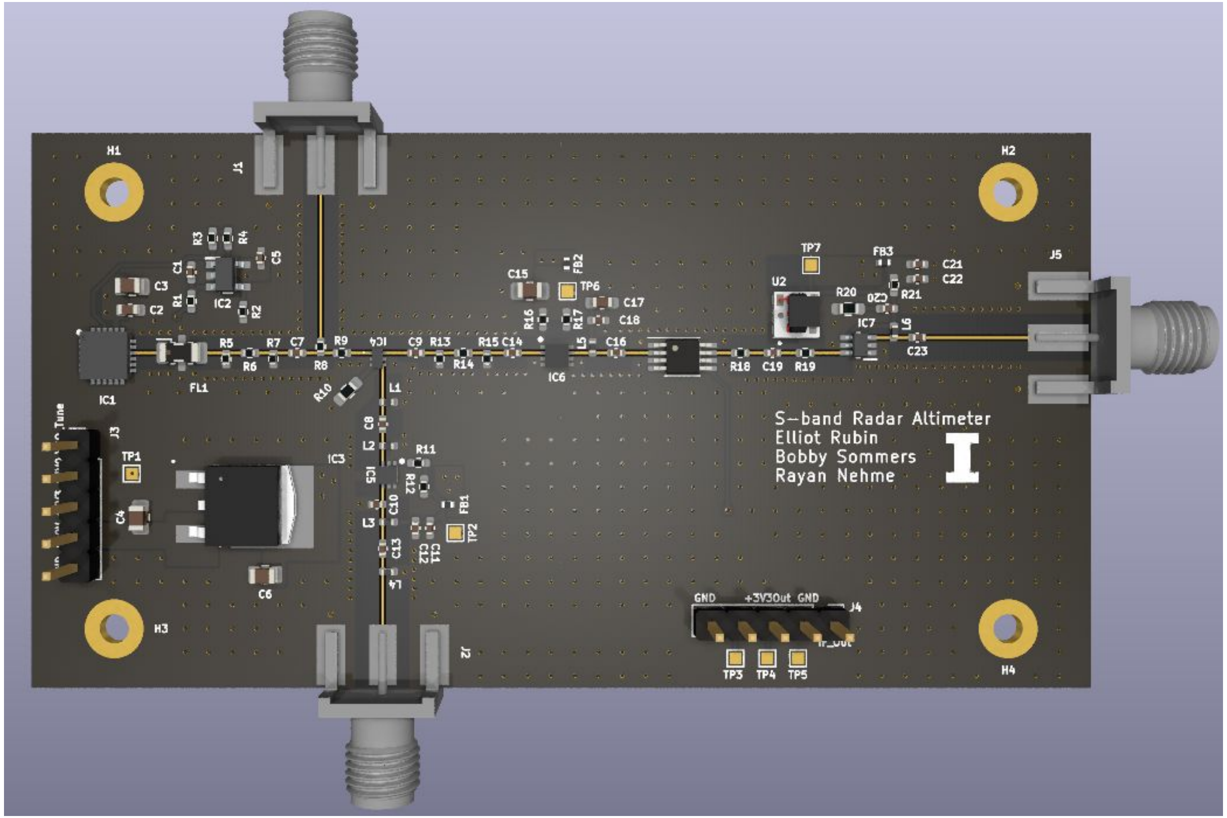
- Replaced with COTS power splitter

Anti aliasing filter was expensive and hard to test

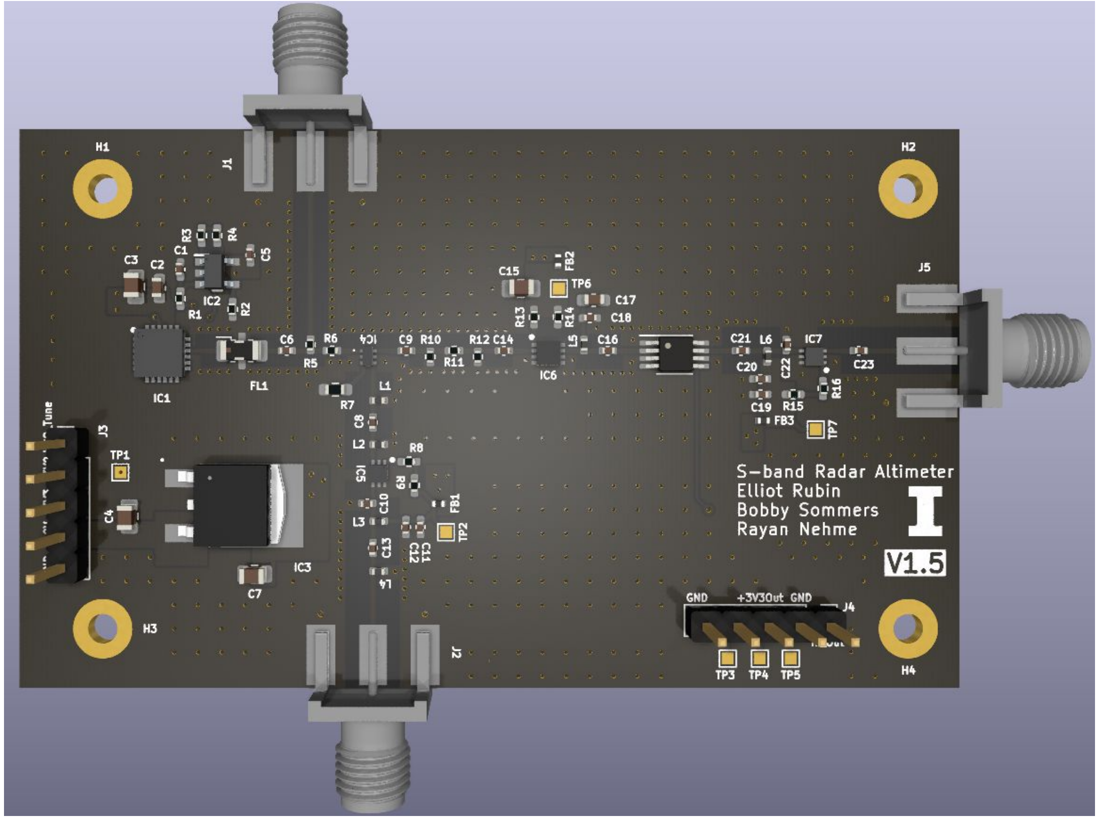
- Expensive components
- Required DC offset, limited dynamic range
- Used breadboard-based LPF instead

5V LDO not used on final board spin

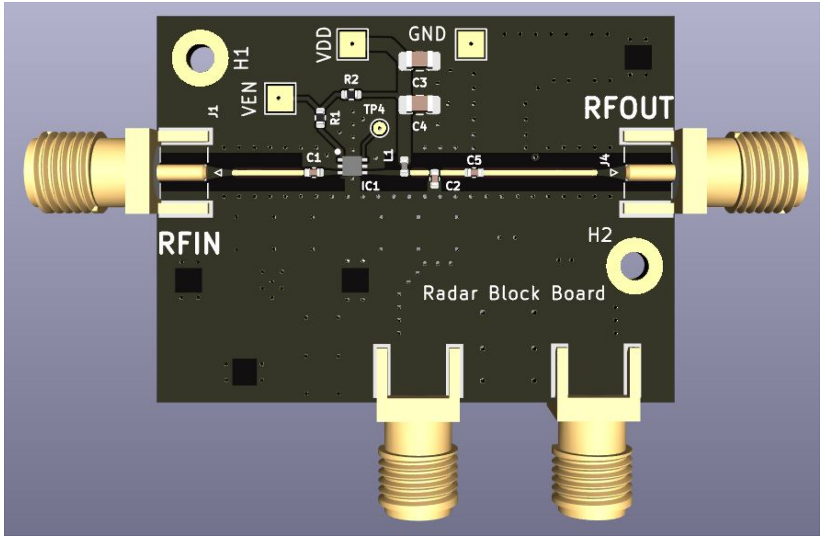
# Radar Unit V1

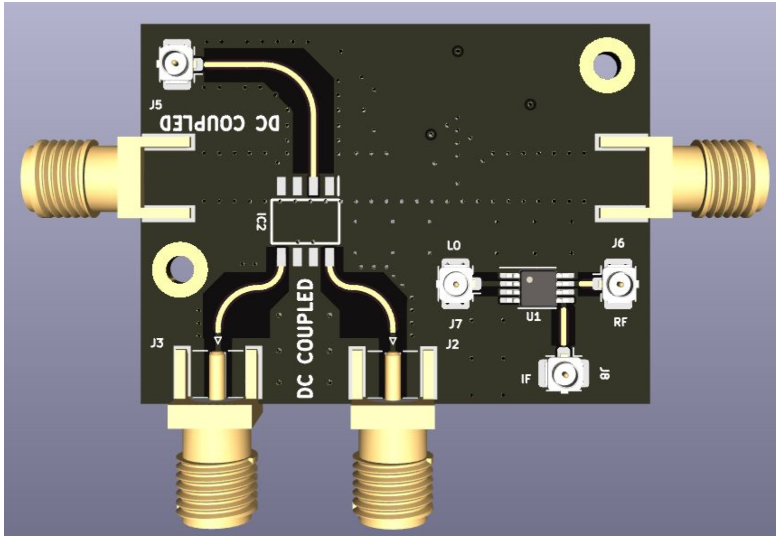


# Radar Unit V1.5 (V2)



# Radar Unit V3



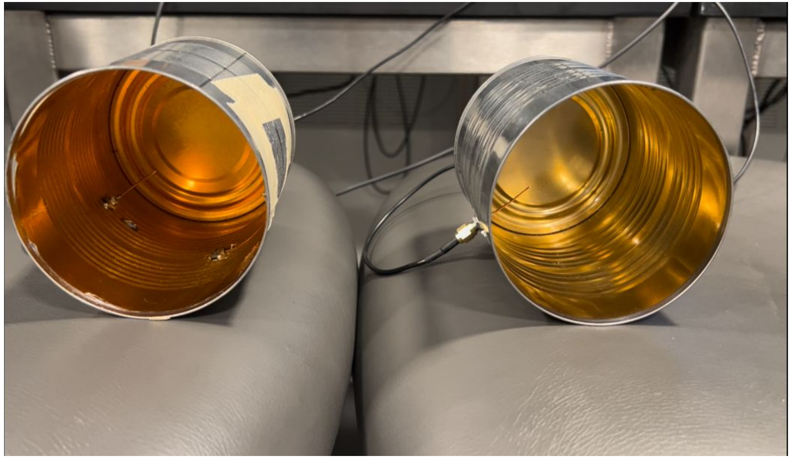



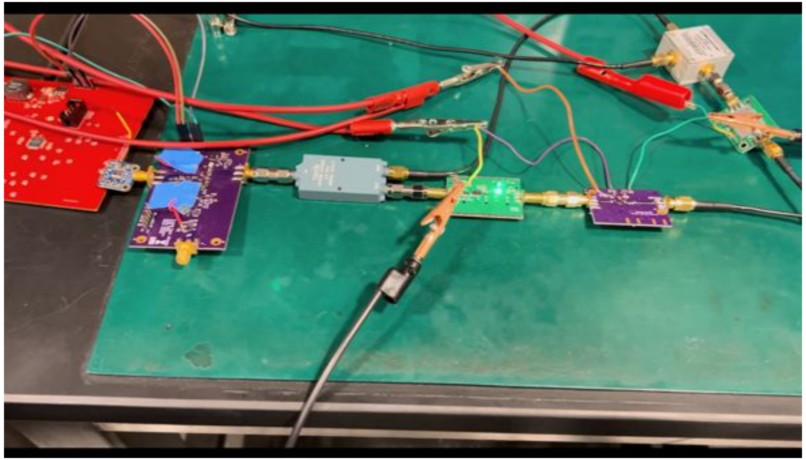

Block design at its most extreme

Modular - populate only what is necessary

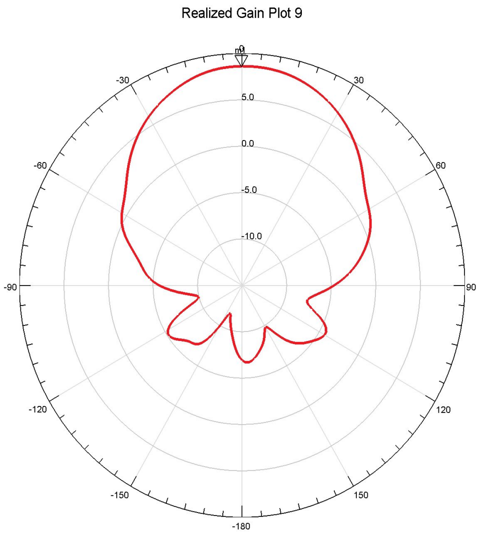
Third time’s the charm - 3rd and final revision was the only fully performant RF board

# Radar Setup

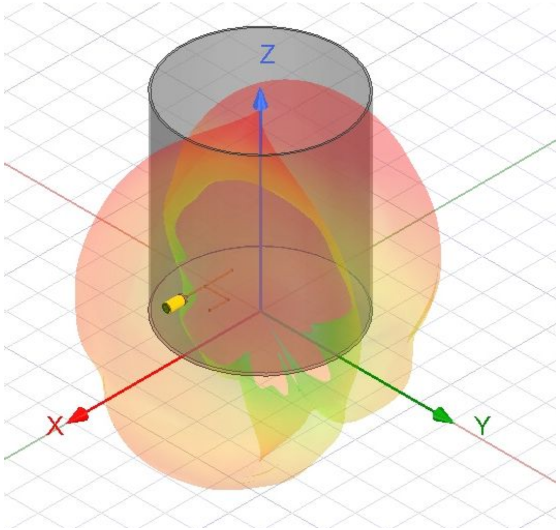




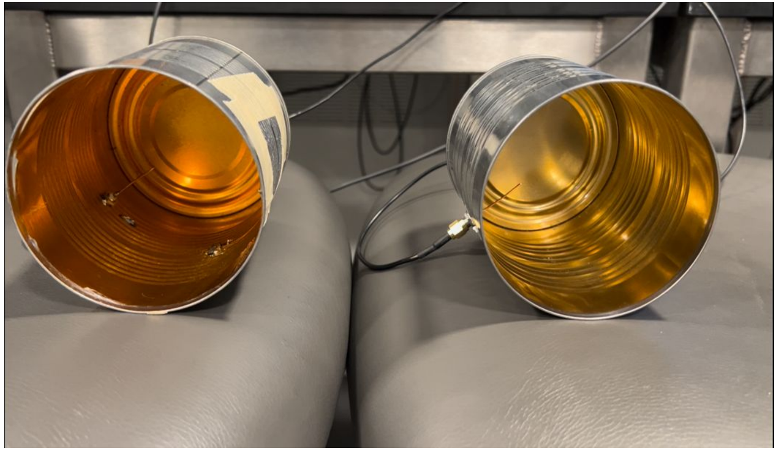
# Cantennas



Realized Gain Plot 9
0
m1
-30
30
-60
60
-90
90
-120
120
-150
150
-180
5.0
0.0
-5.0
-10.0

Z
X
Y

# Radar Unit Verification

Power consumption verified to be ~2W using latest version of amplifier boards along with a COTS amplifier

- custom amplifier boards were far more efficient!

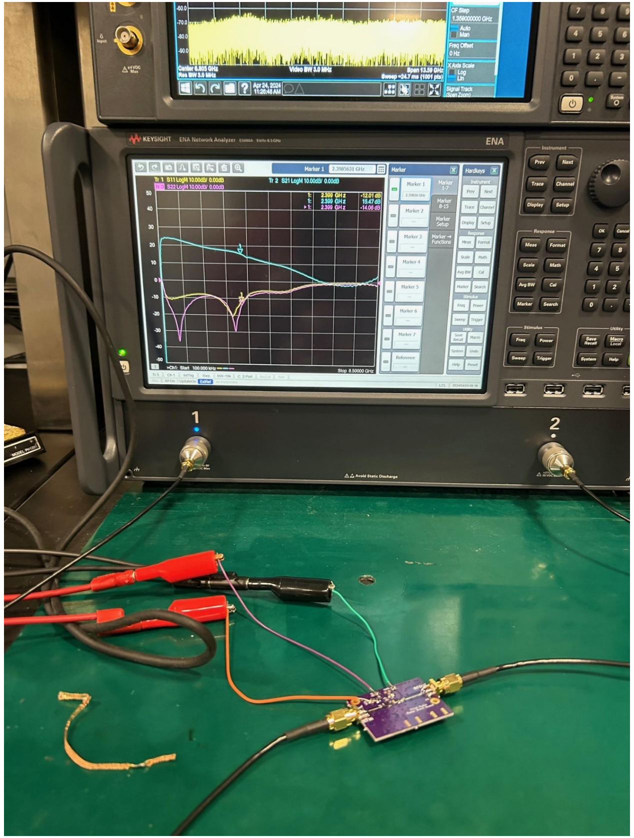
VCO second harmonic found to be -39.4 dBc after low pass filter

- almost 20 dB better than original spec of -20 dBc

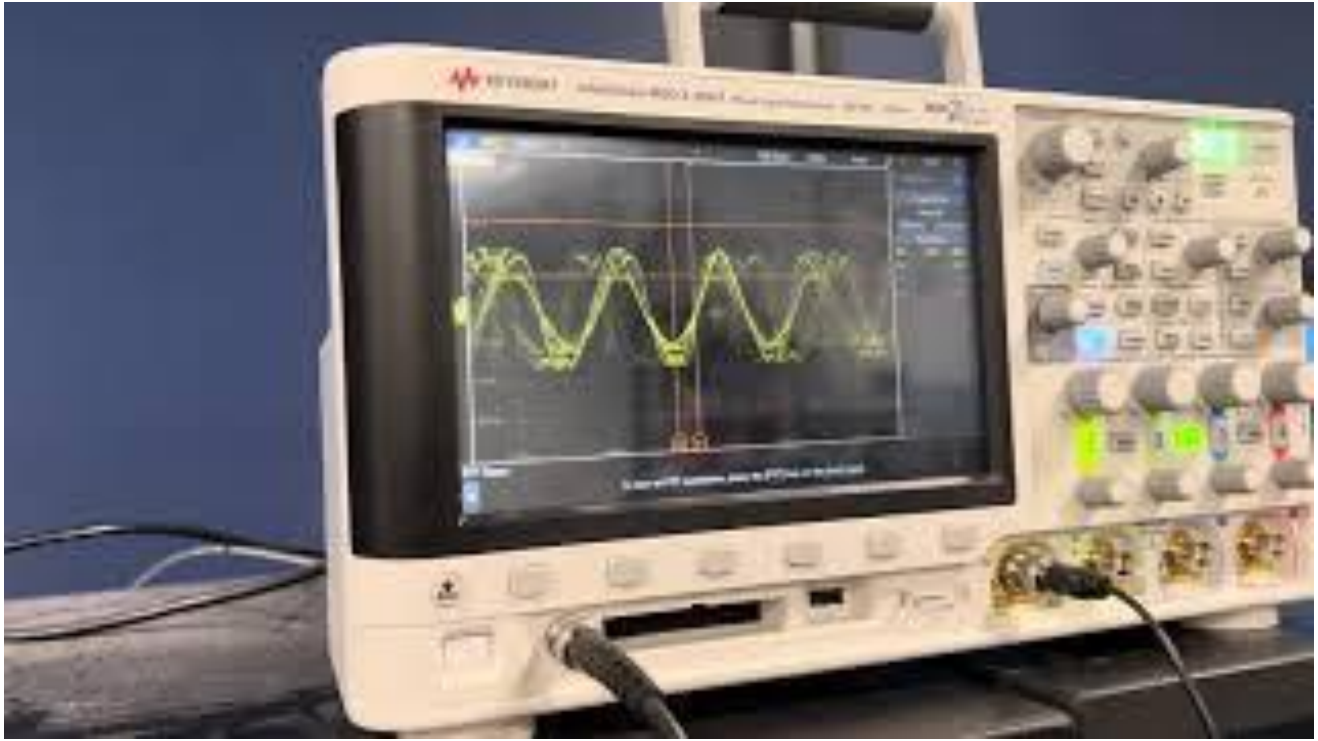
Stability factor for both PA and LNA uniformly greater than 1 for 100 kHz - 3.8 GHz

$$\mu_{ES} = \frac{1 - |S_{11}|^2}{|S_{22} - S_{11}^* D| + |S_{12} S_{21}|} > 1$$

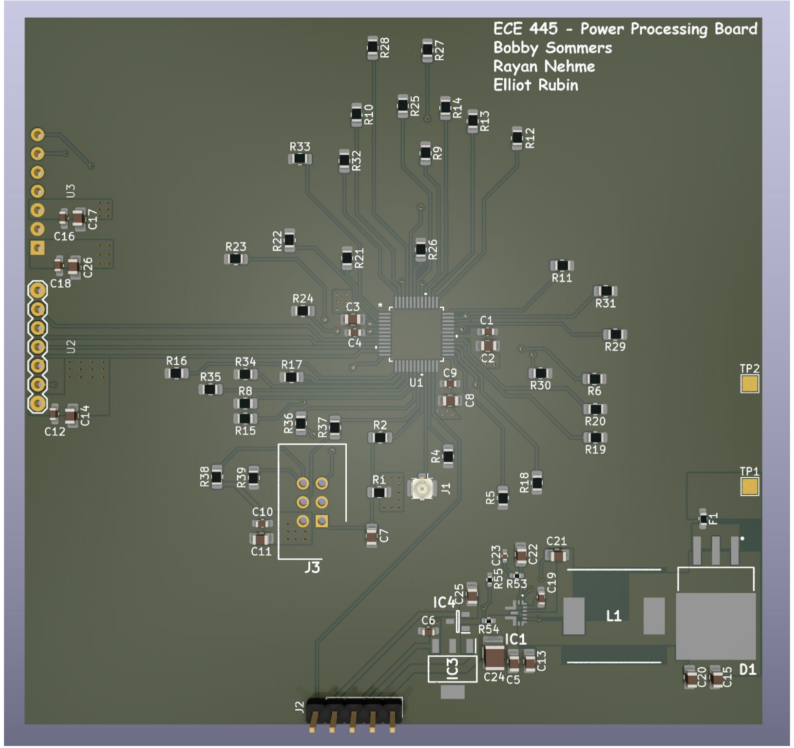
$$D = S_{11} S_{22} - S_{12} S_{21}$$



# Functionality



# Block Overview: Power Processing Unit




Power and processing unit consolidated onto one board

## Power Requirements

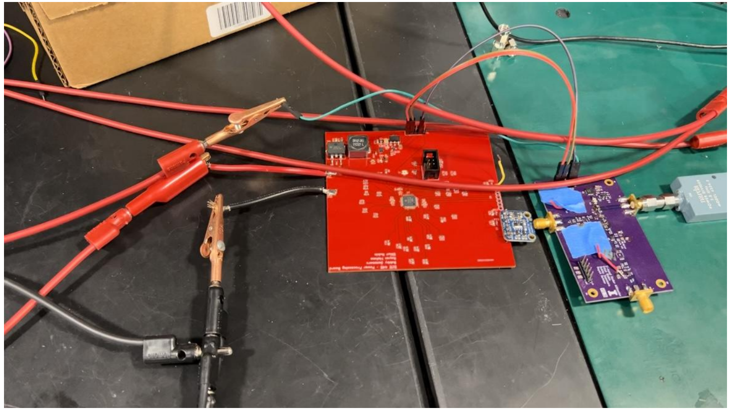
- Reverse polarity protection
- Must contain fuse that will open when >2.5A flows through it
- Under voltage protection
- Vpp ripple for 10V supply must be <0.3V
- Vpp ripple for 3V3, 3V0, 5V supplies must be <0.1V

## Processing Requirements

- Distance error rate must be <10%

# Power Unit Verification

- Reverse polarity was tested by connecting 3.7V across the smart diode in reverse to check that it acted as an open circuit
- Fuse was tested to open when injecting more than 2.5A through it
- Undervoltage protected was integrated into the Li-Ion battery, however this battery was scrapped due to inability to safely charge it
- Ripple voltage was ensured by probing each voltage supply and measuring the peak to peak voltage on the oscilloscope



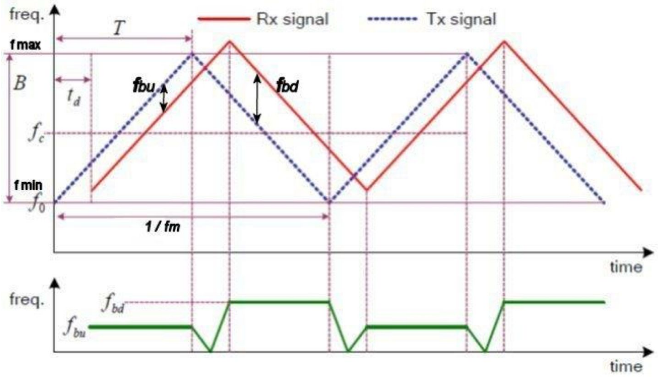
# Processing Unit Verification

- First test with Scopy: Used Scopy to check whether SPI was sending data
- Second test with Scopy: Test to see whether SPI and ADC work simultaneously
- Test with Arduino: Examined the terminal to see if digital values were being outputted
- While still using MPlab we were hoping to use “printf” statements to debug

# Processing Unit Verification Cont.

- Distance Error Rate < 10%
- Used equation to calculate distance
- Compared to value from Barometric sensor
- Not able to get data

$$R = \frac{Tcf_r}{2(f_{max} - f_{min})}$$



# Conclusion/Future Work

- Learned a lot while attempting to complete this very challenging project
- The use of a ESP or STM32  microcontroller as opposed to a PIC
- Better way to power our PCBs instead of the Li-Ion battery
- Implement PLL rather than VCO for the proper beat frequency

# Thank you

Questions?