Senior Design Team 302:

Design/Prototype a Multi-Platform Broadband Communication Payload for a Search and Rescue Operation

Sponsor – Northrop Grumman

Introductions

Theodore Houck~
Test Engineer

Nicholas Crenshaw ~
Lead CPE

Jarrod Love ~
Lead EE

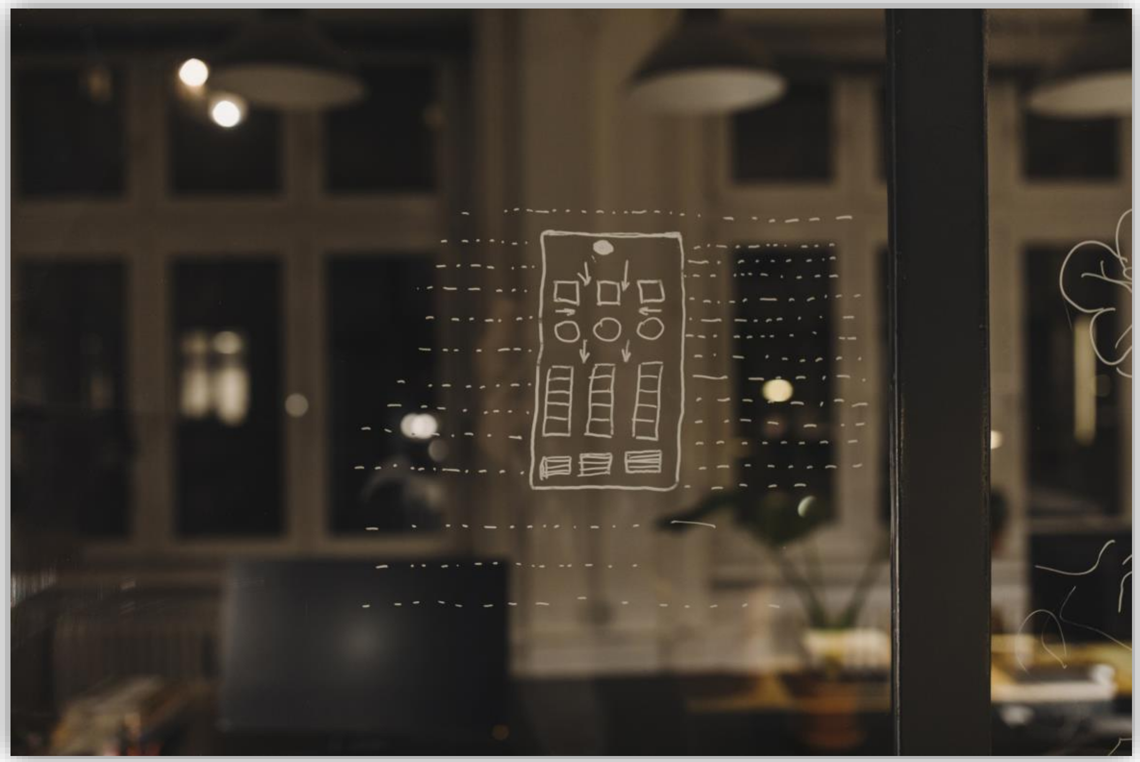
Matthew Brown ~
Team Leader

Sponsor Contact – Lin, Anny
Advisor – Dr. Rajendra Arora

NORTHROP GRUMMAN

Outline

- ↓ Project Overview
- ↓ Part Generation/Selection
- ↓ Interfacing of Parts
- ↓ Testing of Prototype
- ↓ Summary

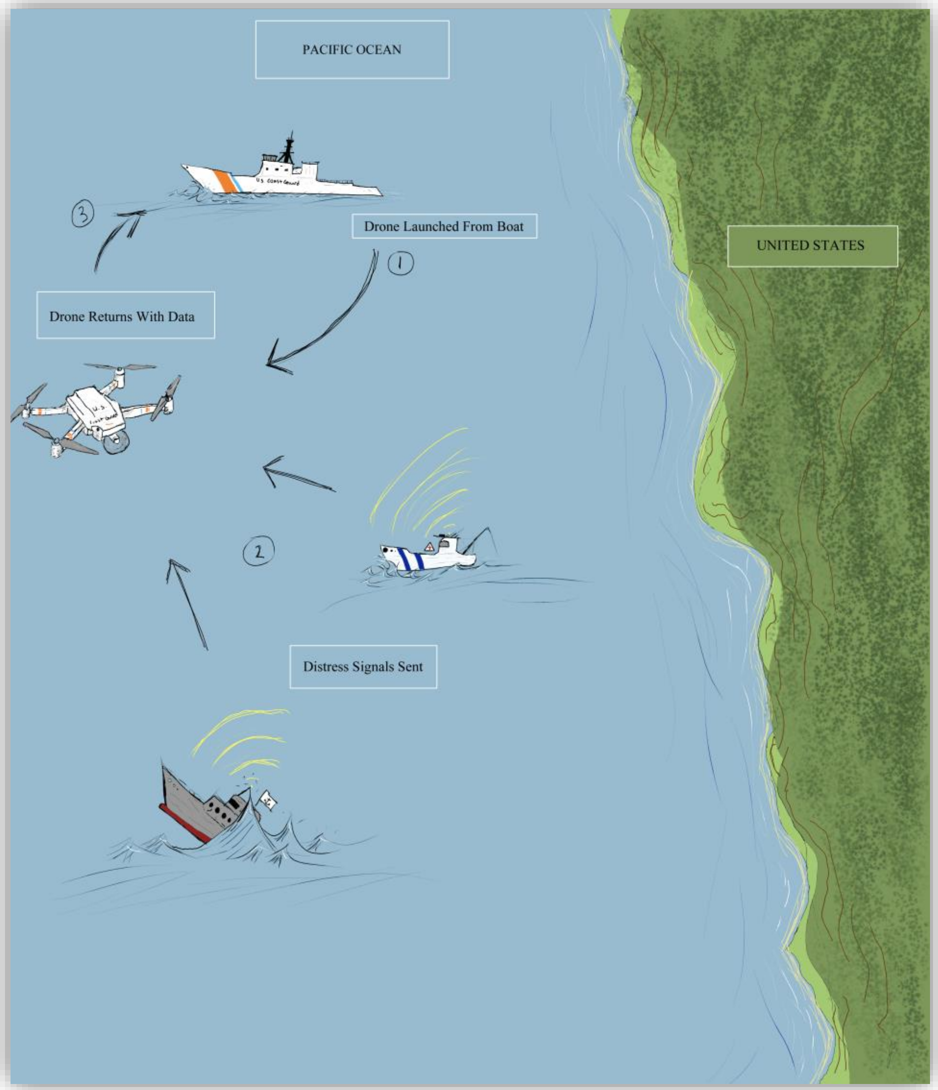


Presenter: Jarrod Love

Background of Project

- A natural disaster has occurred off the coast. Several ships have been severely damaged in such a way that they are unable to return to shore.

- These ships and yachts must be located so that search and rescue (SAR) operations such as emergency helicopters and lifeboats can be deployed to support stranded people and vessels



Presenter: Jarrod Love

Scope of Project

Key Goals

1. Payload can detect emergency signals found in typical emergency radar beacon signal
2. Once detected a timestamp of the detection and current GPS location will be stored
3. The payload should be able to attach to a drone without majorly hindering the function of the drone

Assumptions

- Team will be given a data sheet/information about the drone
- Detection algorithm needed for payload is provided by sponsor and is feasible

Markets

- Primary Market - U.S. Coast Guard
- Secondary Market - Various emergency responders or U.S. Military

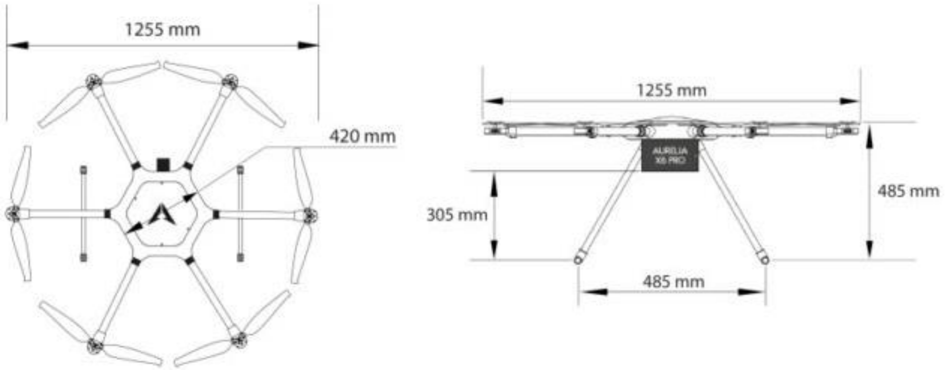
Stakeholders

- Northrop Grumman
- United States Coast Guard
- United States Military

Presenter: Jarrod Love

NORTHROP GRUMMAN

Targets



- Emergency signal detection
- Support of FPGA code (can support algorithm given by Northrop Grumman)
- Independent battery source (Payload will have a battery life that will outlast the drone, approximately 1 hr+)
- Internal storage memory
- Modularity (Should be able to easily mount and dismount)
- Weight limit of 5 kg
- Must be able to detect various emergency frequencies
  - Current target frequency: 156.8 MHz (RF International Distress Frequency)

Presenter: Jarrod Love

Concept Generation

- Concepts were generated using brainstorming and a morphological chart
- Chart to the right is ½ of the whole concept generation chart

FPGA	Zynq UltraScale+ RFSoC ZCU111	Zybo Z7: Zynq-7000 ARM/FPGA SoC	Nexys A7: FPGA Trainer Board	ZedBoard Zynq-7000 ARM/FPGA SoC Development Board	
FPGA Cont.	Alchitry Au+ FPGA Development Board (Xilinx Artix 7)	Arria 10 SoC @1.4GHz , 2GB RAM for HPS, 4GB FPGA	Arria 10 SoC @1.4GHz , 1GB RAM for HPS, 256Mbit QSPI Flash, 8GB MicroSD	Arria 10 @1.0GHz SoC with 2 speed (Dual ARM Cortex A9 + 480K LE)	
FPGA Cont.	Digilent USRP B205mini-i: Software	Arria 10 SoC with 3 speed			
FPGA Alternatives	Microcontroller	Microprocessors	Arduino		
Microprocessors	IC MPU 0MAP-L1X 375MHZ TI	IC MPU I.MX6QP 1GHZ NXP	IC MCU 3BIT EXT MEM 289LFBGA Mic Tech	IC MPU Z180 20MHZ 80QFP Zilog	IC MPU SITARA 375MHZ 176HLQFP TI
Microcontroller	Arduino Uno R3 USB Microcontroller	Arduino Mega 2560 Microcontroller Rev3	Arduino Leonardo Microcontroller	Arduino Nano 33 IoT Microcontroller with Headers	Bluno Arduino Uno BLE Bluetooth 4.0 Microcontroller
Power Source	Lithium-Ion Battery Pack	Solar Power	Rechargeable Battery Pack	Drone Battery	
Power Source Cont.	D-Cells	Dry Batteries	Organic Battery Cell		
Detection on Methods	Radar Detection	Thermal Imaging	Lidar	Standard Camera ID Software	
Drone Mounting System	Inside storage compartment	Attached to Gimbal mounted on drone	Attached apparatus on top of drone	Integrated directly with drone	
External Storage Methodology	Micro SD card	SD card	SSD	HDD	

Presenter: Jarrod Love

Concept Selection

- Concepts were selected based on several categories
- Primarily, these criteria consisted of:
  - Weight of the component
  - Ease of integration of the component
  - Sponsor requirements
  - Cost of component

Presenter: Jarrod Love

NORTHROP GRUMMAN

Part Selection: FPGA

	WEIGHT	Zynq UltraScale+ RFSoC ZCU111	Xilinx Zynq UltraScale+ MPSoC ZCU102	Abaco VP431 RFSoC Board	Genesys 2 Kintex-7 FPGA Development Board	Xilinx Kintex UltraScale+ FPGA KCU116
Size (1 = Smaller)	5	-	1	1	1	1
Has ADC? (1 = Yes)	4	-	-1	1	1	-1
Power Required (1 = less power)	2	-	0	0	0	0
High Heritage COT? (1 = yes)	3	-	1	1	-1	1
Cost (1 = less cost)	3	-	1	-1	1	1
SCORE		-	7	9	9	7
Continue?		Use	No	Yes	Yes	No

Presenter: Theodore Houck

Part Selection: Battery

	WEIGHT	Alkaline Battery Pack	Solar Power	Rechargeable Battery Pack	Drone Battery	D-Cells	Dry Batteries
Battery Life	5	-	1	-1	-1	-1	0
Power Ouput	2	-	-1	0	0	0	0
Longevity	1	-	0	0	0	-1	0
Weight	4	-	-1	0	1	1	-1
SCORE		-	-1	-5	-1	-2	-4
Continue?		Use	No	No	No	No	No

Presenter: Theodore Houck

Part Selection: Transceiver and Antenna

	WEIGHT	AD-FMCOMMS4-EBZ Wideband Software Defined Radio Board	CBX 1200-6000 for Ettus USRP N210	WBX 50-2200 MHz for Ettus USRP N210	CBX 1000-5500 for Ettus USRP N200	Pmod 5088 GPS Receiver	Detection on an FPGA	WBX 25-1500 MHz for Ettus
Low Power Consumption	3	-	-1	-1	1	-1	1	1
Weight	3	-	1	0	0	-1	1	0
Detection Range	4	-	-1	-1	-1	-1	-1	-1
Cost	2	-	1	1	-1	-1	-1	-1
SCORE		-	-2	-5	-4	-12	0	-4
Continue?		Best Option	No	No	No	No	No	No

- There wasn't a plethora of antennas for our desired frequency range
- No concept selection chart was used to pick antenna

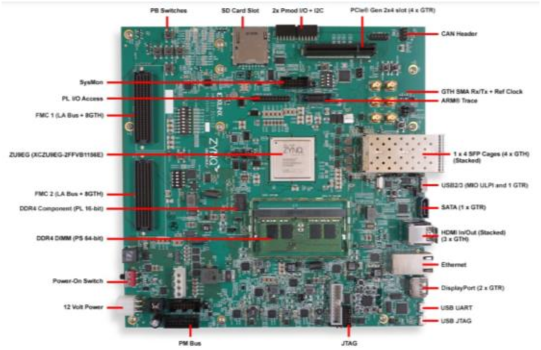
Presenter: Theodore Houck

Part Selection: Misc

- A 32 GB SD card was selected to provide a safe amount of storage.
- A pack of 6 Pin PCIE connection cables were purchased to allow the battery to connect to the FPGA
- A Digilent GPS Receiver was purchased as it uses PMOD to easily connect to our FPGA

Presenter: Theodore Houck

Core Components



FPGA
Xilinx Zynq UltraScale+ MPSoC ZCU102

Transceiver
AD-FMCOMMS4-EBZ Wideband Software Defined Radio Board

Power Source (2)
BatteryGuy 12V 2200 mAh Alkaline Door Lock Battery

Antenna (2)
VHF stout and long antenna SMA connector 150-160MHz

Presenter: Matthew Brown

Additional Components

SD CARD
SanDisk 32GB Ultra SDHC UHS-I Memory Card - 120MB/s

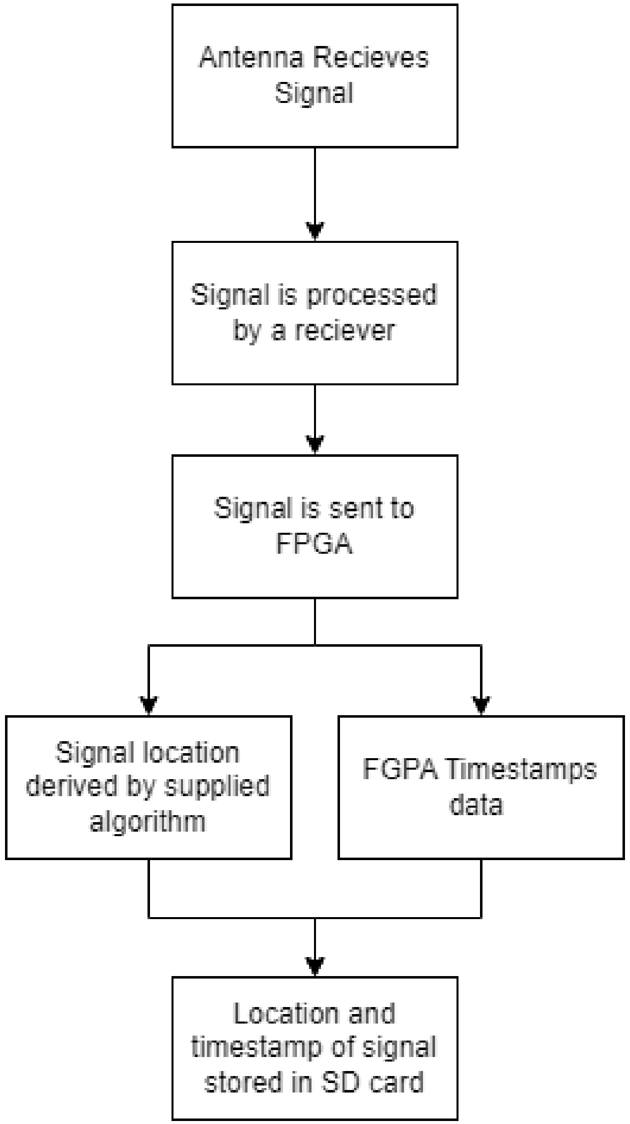
Battery Connectors
Cable Matters 2-Pack 6 Pin PCIe Extension Cable 10 Inches

PMOD GPS
Digilent GPS Receiver

Presenter: Matthew Brown

Planned Solution

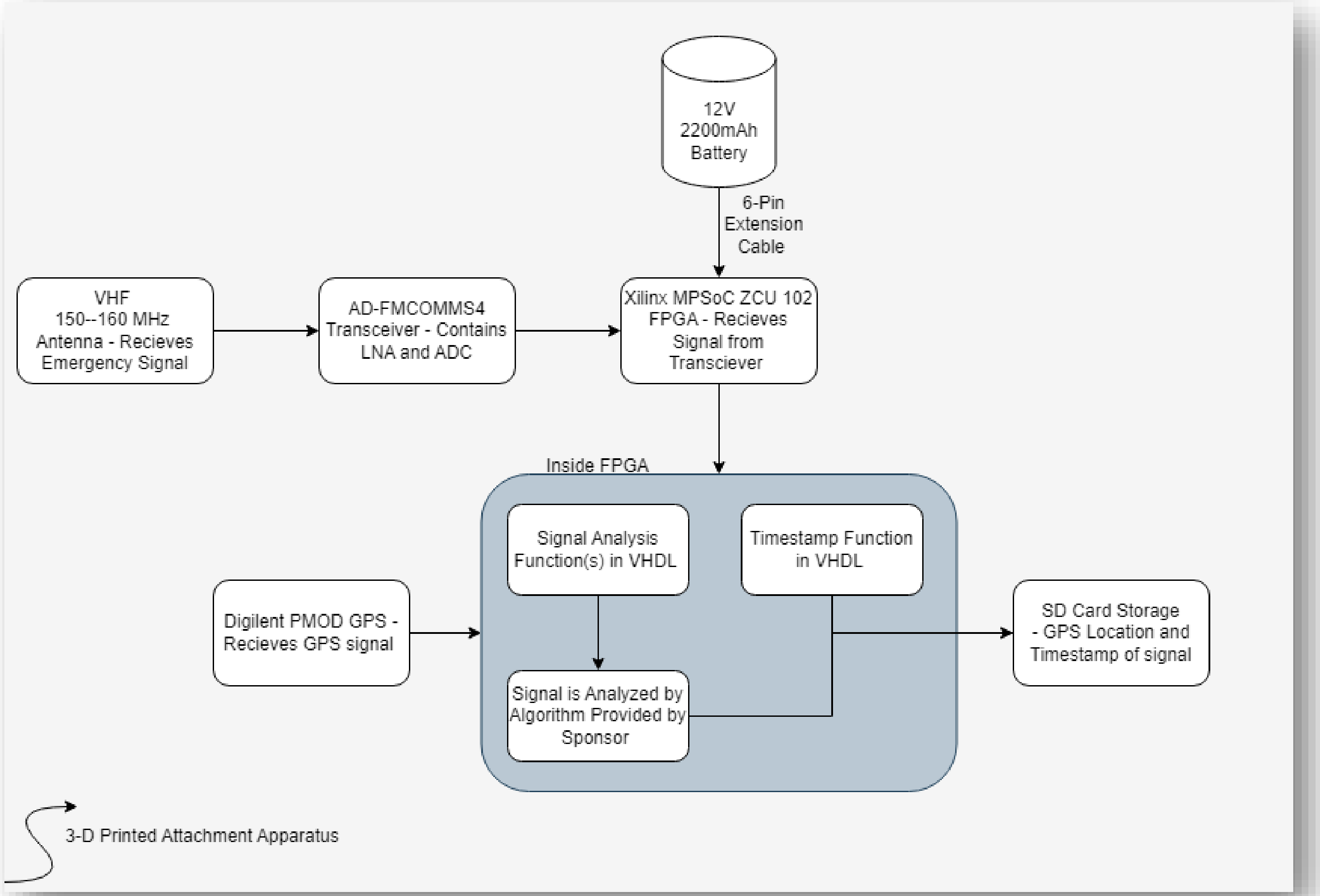
- This diagram follows the path of the signal
- Whole solution is powered by a remote power source



Presenter: Matthew Brown

Final Design Block Diagram

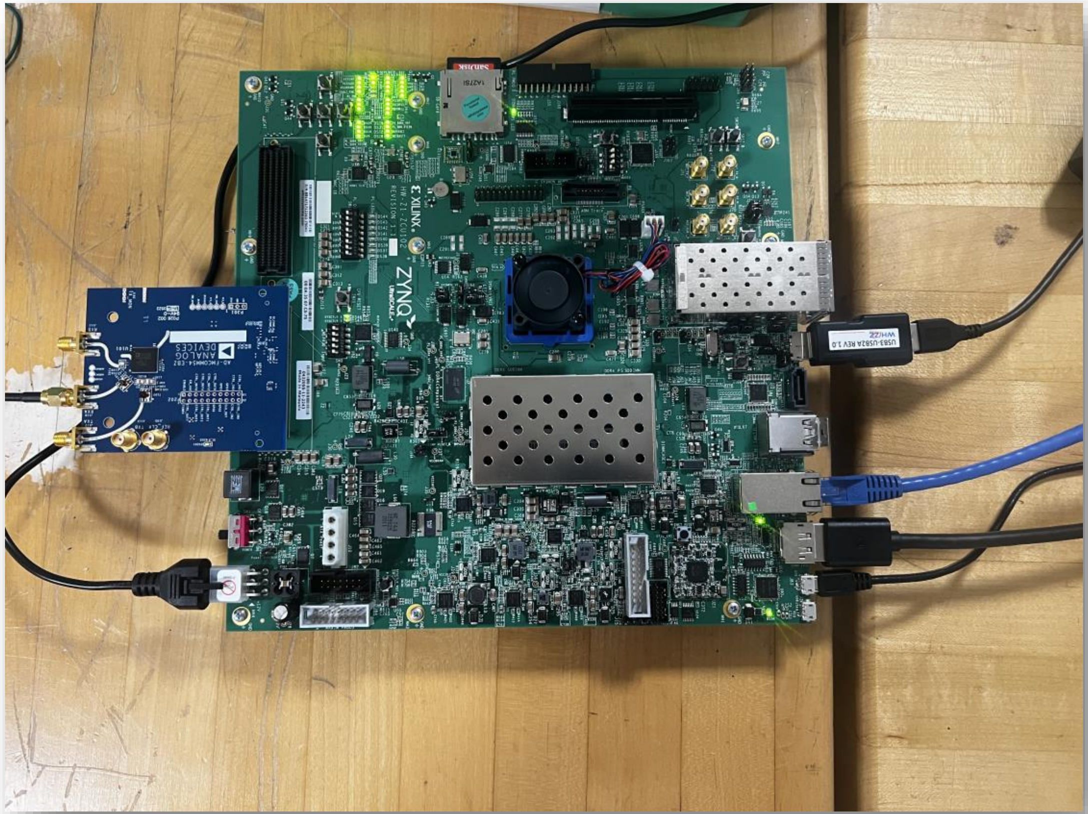
NORTHROP GRUMMAN



Presenter: Matthew Brown

Interfacing: Transceiver Module

- Original idea was to plug into FPGA and code to it
- Documentation for the receiver was unclear as to what pins are needed to program/read data to/from the transceiver
- The suggested method from the dealer is to setup up their local Linux environment on the FPGA



Presenter: Matthew Brown

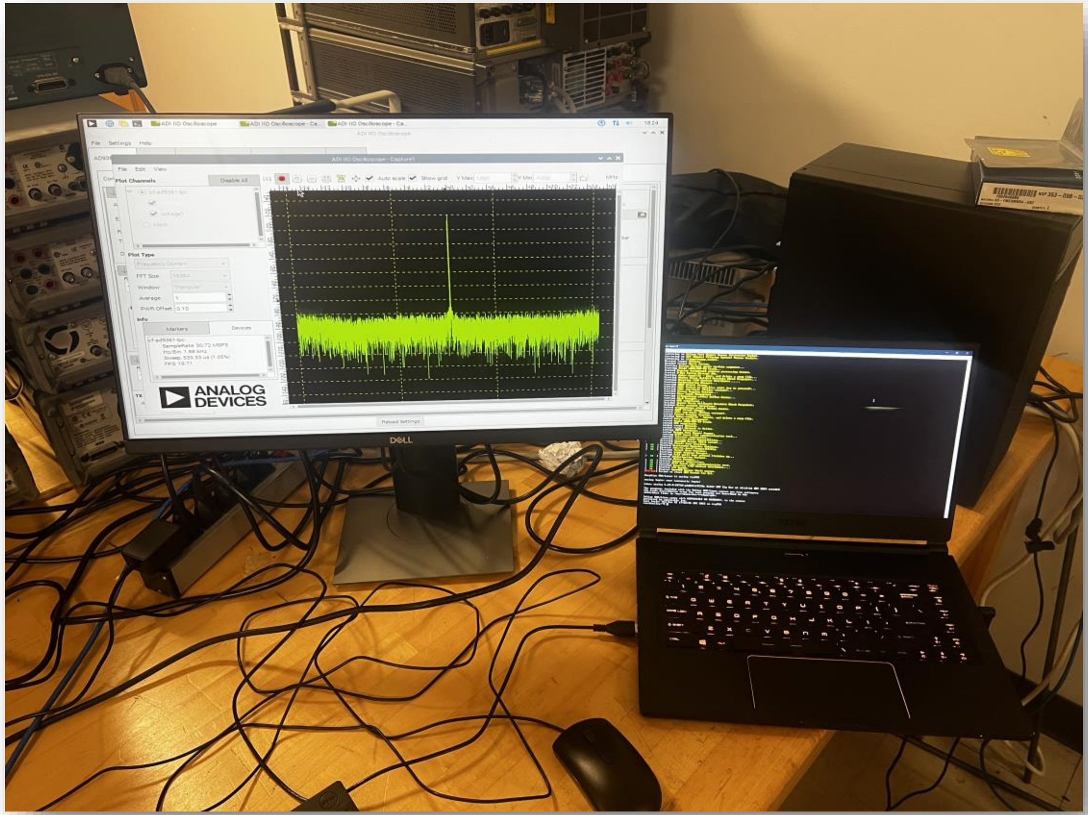
Interfacing: Transceiver Module

- Setup of Analog Devices boot image
  - Original Linux environment did not work, had to recreate on own
  - Rewrote device tree on Linux image to allow USB devices for interfacing
  - Change of voltage rail rating for FMC connector through FPGA system controller
  - Additional miscellaneous fixes to use boot image

Presenter: Matthew Brown

Interfacing: Transceiver Module

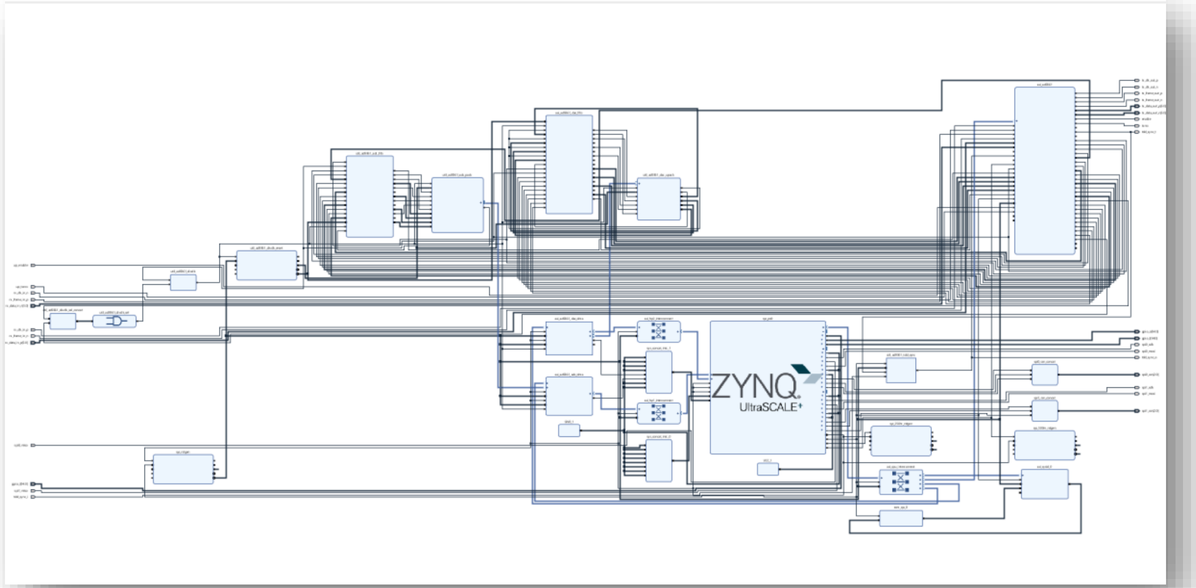
- Successfully able to detect signals coming from the signal generator via a direct SMA cable
- Was able to pick up signal(s) wirelessly with our antennas
- The current FPGA local Linux environment/program setup allows to change numerous settings on FPGA



Presenter: Nicholas Crenshaw

Interfacing: Transceiver Module

- Goal is to use VHDL based project not Linux
- A no-OS HDL project was created
- This allows implementing other VHDL designs with the transceiver project
- After finishing this HDL project exporting it as an image and booting off the SD card will be required



Presenter: Nicholas Crenshaw

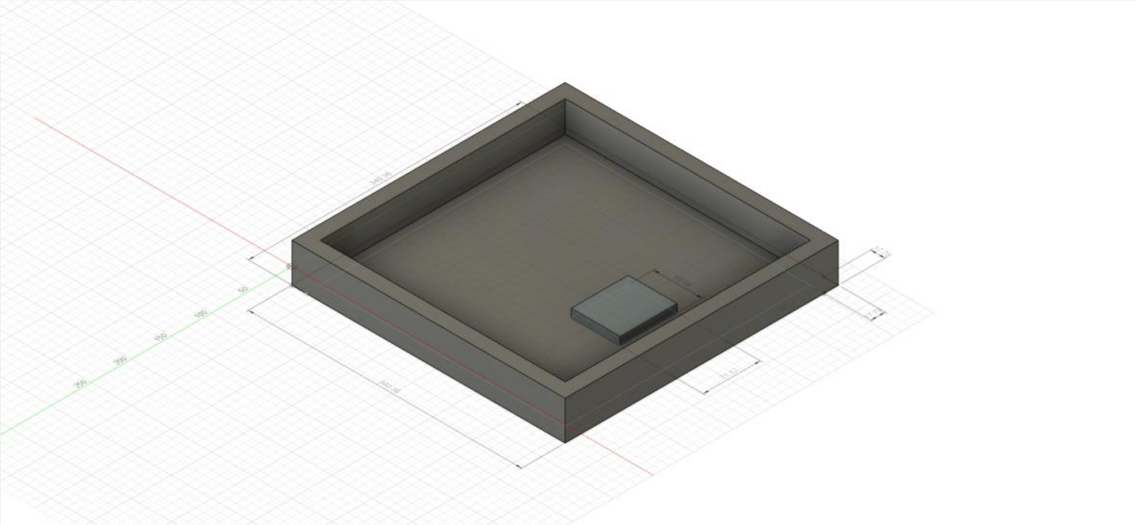
Interfacing: PMOD GPS

- GPS Module has not been successfully interfaced with FPGA
- Multiple IP blocks are available from Digilent for implementation
- Implementation will be done with I2C

Presenter: Nicholas Crenshaw

Housing Design

- The 3-D printed housing for the design is currently in production
- Several redesigns occurred because FSU does not have any 3D printers that print the entire housing at once

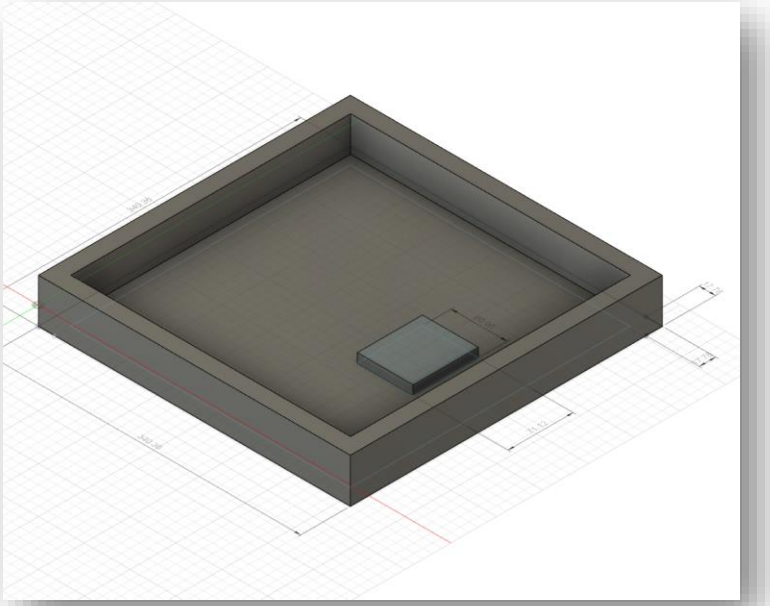


Presenter: Theodore Houck

NORTHROP GRUMMAN

Housing Design

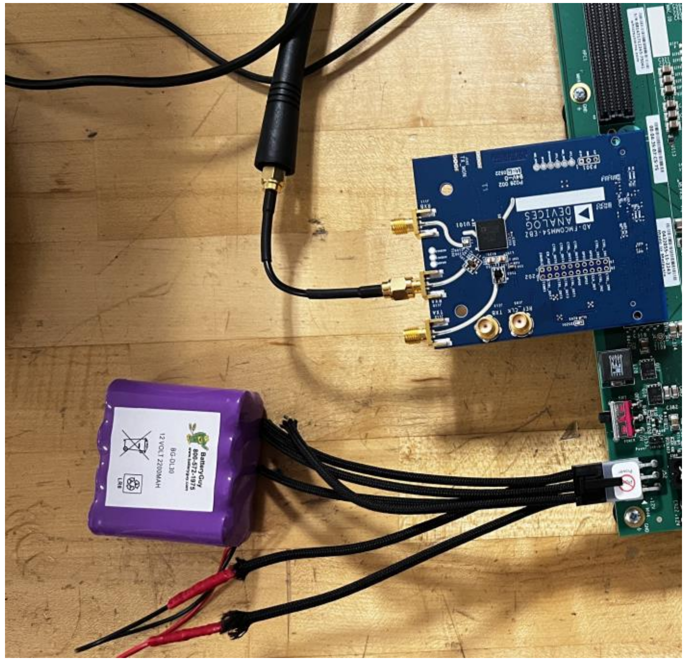
- The chief concern of the design is the weight
- 'Dead space' was reduced as much as possible
- Resulting design is not form fit to FPGA but has a ridge for battery and transceiver
- Housing will be 3D printed in two sections and will be joined via a friction fit

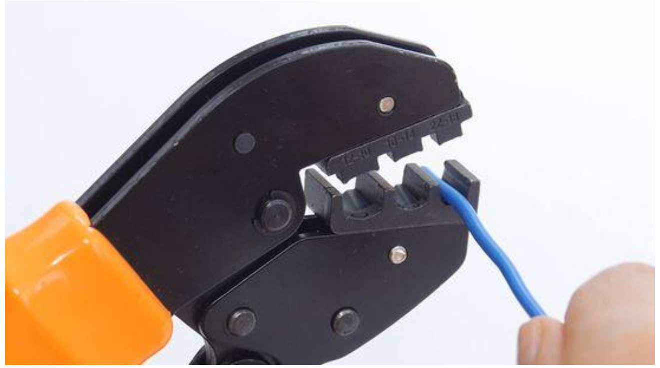


Presenter: Theodore Houck

Implementation of Batteries

- In order to have the batteries directly connect to the FPGA the 6-Pin PCIE connecter needed to be attached to the battery instead of the default end cables
- A cord crimp was used to accomplish this easily and allows for the cords to be separated if need be



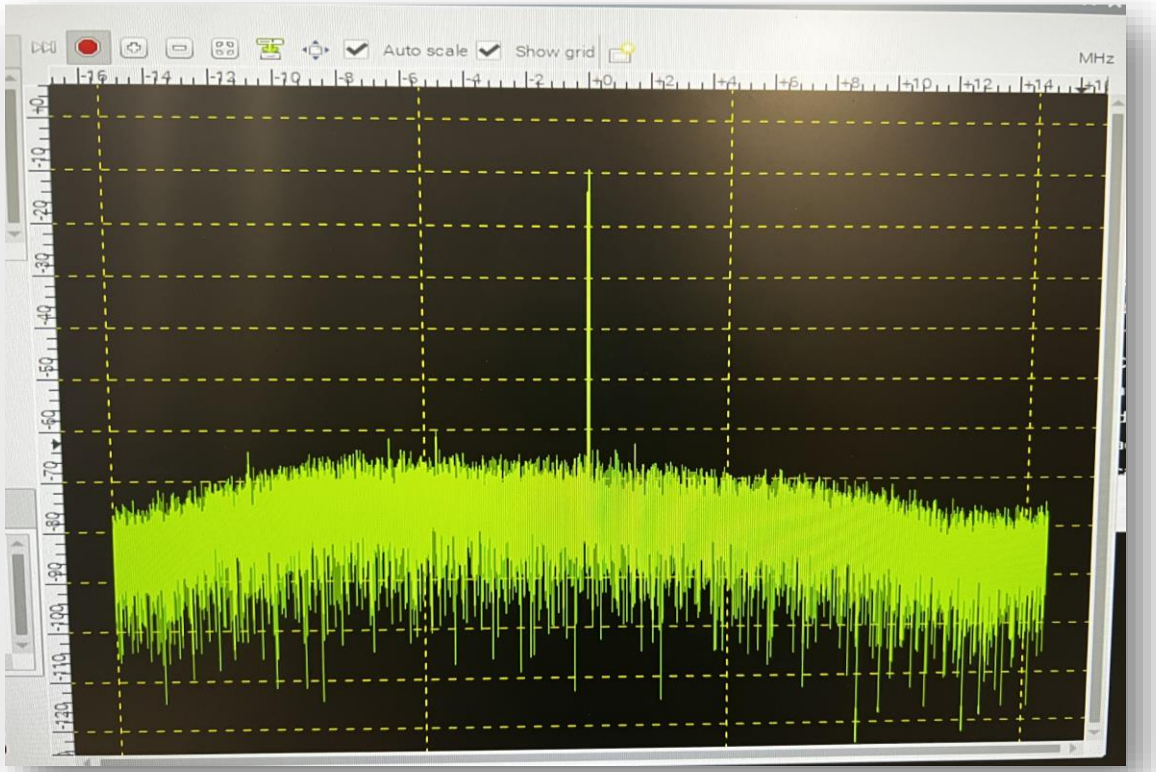


Presenter: Jarrod Love

Testing: Antenna (Freq)

- Desired frequencies were produced with a signal generator
- A frequency in the range of 150MHz to 1GHz was received

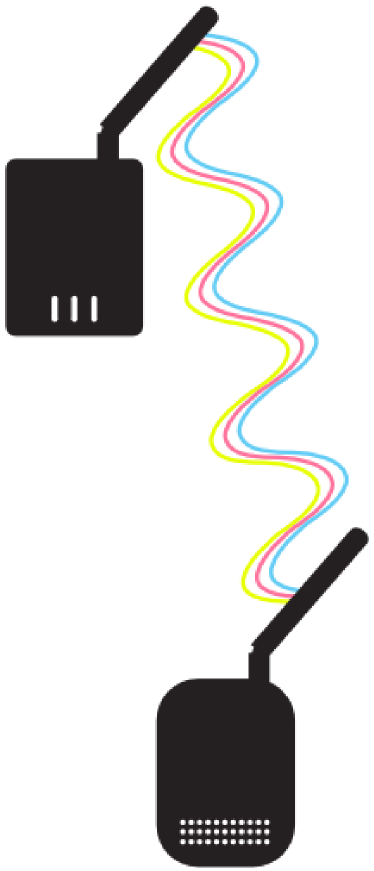
- Center Freq: 155 MHz
- Testing distance: ~5 ft
- Peak of desired signal: -10dB
- Amplitude of noise: -64dB



Presenter: Nicholas Crenshaw

Testing: Range of Design

- To test range, signal was transmitted from 3rd floor and receiver was placed on 1st floor
- Signal was not received
- Possibly due to:
  - Signal generator is not transmitting at high enough power
  - Noise issues



Presenter: Nicholas Crenshaw

Current Prototype

- The I and Q data produced was saved as both a CSV and MAT file onto the SD card
- The SD card is inserted into a computer
- A MATLAB script analyzes the I and Q data and plots a periodogram
- Detection of the emergency signal is stored into a .txt file and is time stamped

Presenter: Nicholas Crenshaw

Current Challenges

- Production of 3D printed housing
  - There are no 3D printers at FSU that allow the housing to be printed as a single piece
  - The housing would have to be designed as multiple pieces
- Automation of the signal detection and processing
  - Automation of data analysis via VHDL

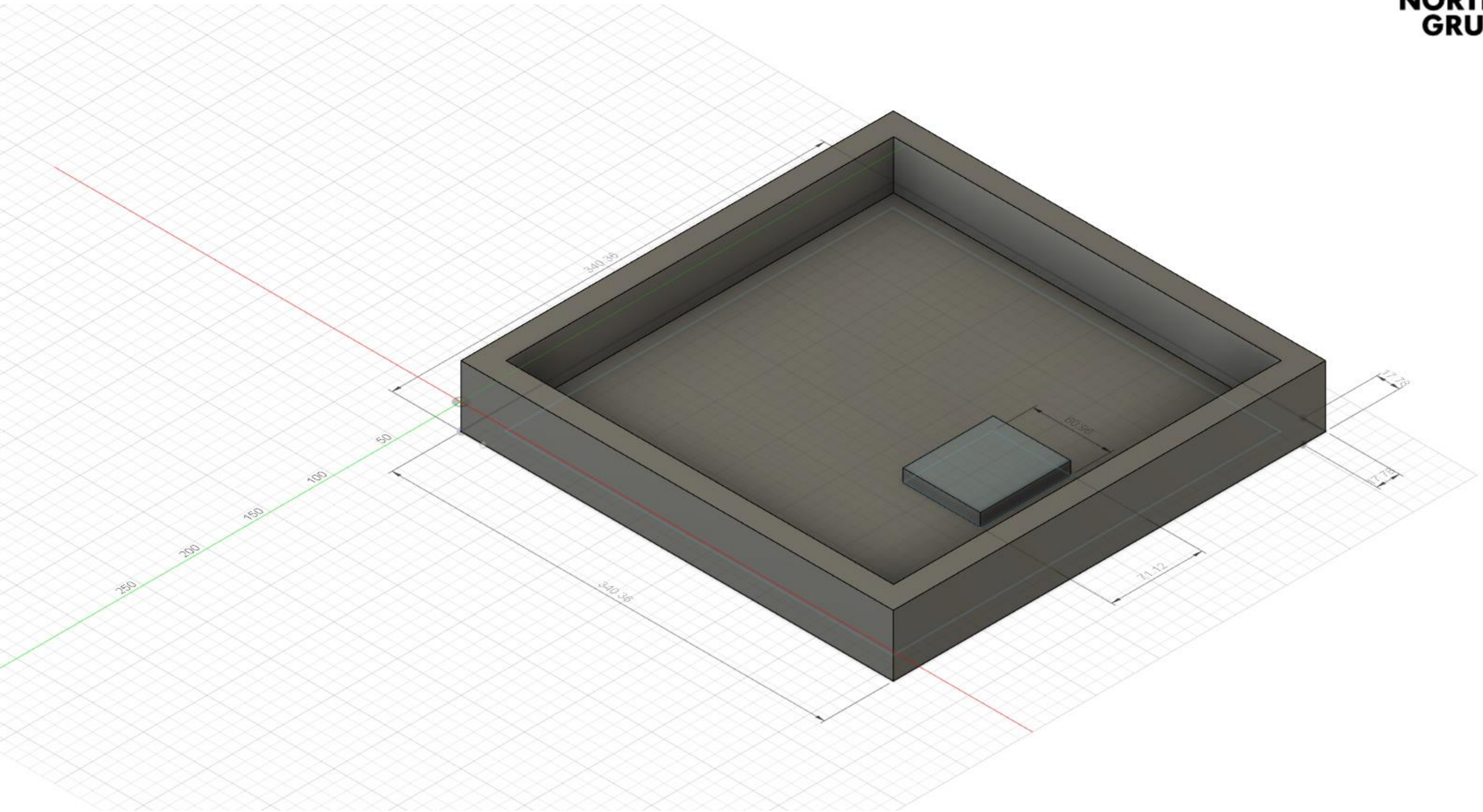
Presenter: Nicholas Crenshaw

Summary

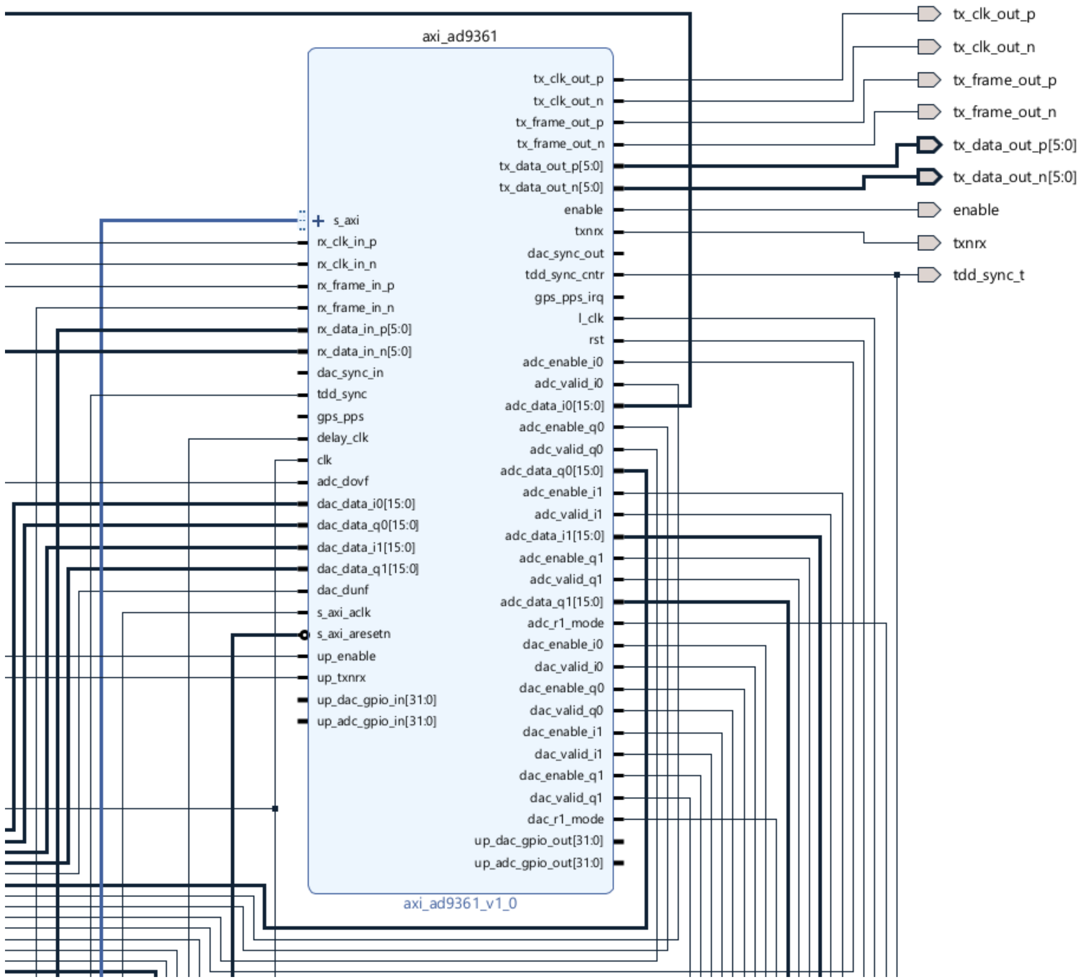
- A working prototype now needs to be translated to a fieldable product using HDL code to add on to the transceiver VHDL project
- The payload housing is nearing completion
- Once printed, the payload will be fully assembled and placed into its housing
- The design while being more challenging than originally anticipated, has made significant progress

Presenter: Theodore Houck

Thanks for Listening! Questions?


340.36
340.36
50
100
150
200
250
17.78
60.96
71.12
17.78

AD9361Block



Power Calculations

- In order to determine the estimated power consumption of the FPGA, the following formula was used: $P_{avg} = \frac{1}{2} \sum_{n \in nets} C_n \cdot f_n \cdot V^2$
- Where C_n = the capacitance of a net on the FPGA, V is the supply voltage, and f_n = the expected frequency a net will be operating at.
- Using the values from the FPGA data sheet yields an estimated average power consumption of approximately 45WH.
- The selected battery is rated at 12V and 2200mAH, therefore this = 26,400 mWH = 26.4 WH
- Therefore, two batteries were purchased to provide the full power necessary to the load.

MATLAB

plotIQ.m

```
sampleRate = 30e6;
centerFreq = 155e6;
bandWidth = 15e6;
load('test6_BW_15.mat')

fin_i = fft(cf_ad9361_lpc_voltage0);
fin_q = fft(cf_ad9361_lpc_voltage1);

comp_data = (cf_ad9361_lpc_voltage0 + i*(cf_ad9361_lpc_voltage1));
fin_com = fft(10*log10(abs(comp_data)));
IQData = periodogram(comp_data,hamming(length(comp_data)),[],sampleRate,"centered");

tempSampleSize = size(IQData);
sampleSize = tempSampleSize(1);
periodogram(comp_data,hamming(length(comp_data)),[],sampleRate,"centered");

ampIQData=10*log10(IQData);
peakAmp = max(ampIQData);
maxIndex = find(IQData==max(IQData));
maxFreq = ((sampleRate/sampleSize)*maxIndex)+(centerFreq-bandWidth);
maxFreqMHz = maxFreq/1e6;

if(maxFreq>=(155e6*.9) && maxFreq<=(155e6*1.1))
    fileID = fopen('data.txt','a');
    fprintf(fileID,'Emergency Signal of %dMHz was detected on %s\n', maxFreqMHz, datestr(now,'mmmm dd, yyyy HH:MM:SS PM'));
    fclose(fileID);
end
```