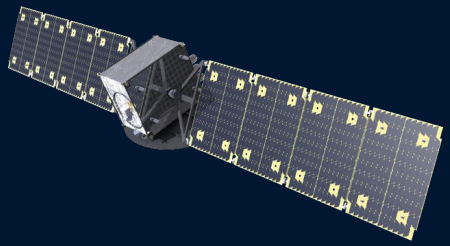
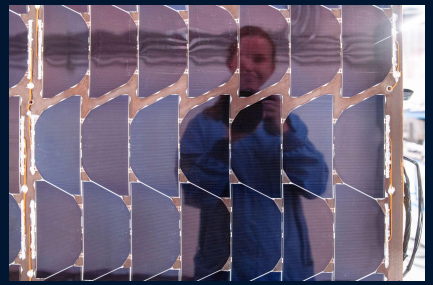
Low-Cost Software-Defined Radio

Alex Willis & Anya Svintsitski

Background

- We interned at a place that makes satellites with Software-Defined Radios (SDRs)
  - Alex did SDR hardware, Anya did avionics
- SDRs have revolutionized many fields of wireless communications
- Typically very high budget, complexity and performance
- Some existing cheap products exist but are pretty niche
  - Nothing <$300 that works for networks

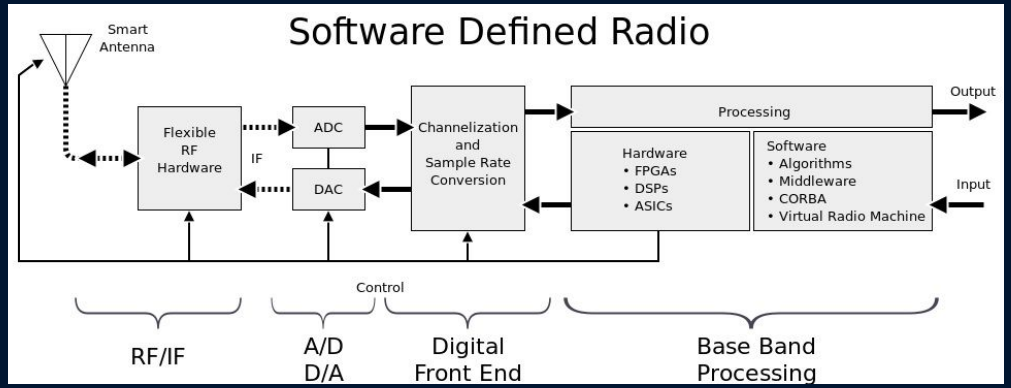




ASTRANIS

What is an SDR?

- RF communication system with software performing signal-processing tasks
- First made in 1987 by the air force, made one module to replace its several radio architecture which performed integrated communications, navigation, and identification
- In 1991, the military made SpeakEasy 1, which allowed 2MHz to 2GHz communication



[4]

Benefits of SDRs

- Adaptability for changing standards
- Can communicate over multiple channels simultaneously
- Software development has lower associated costs
- Don’t have to design different sets of hardware for the same product line with varying features like frequency bands or modulation schemes

Existing SDRs: High End

- Capable, complex, and expensive
- Based around high-performance FPGAs and data converters
- USRP (Universal Software Radio Peripheral)
  - Common R&D Tool
  - Expensive ($2,900 - $20,000+)
  - DC - 6 GHz range
  - 25 MHz - 400 MHz usable bandwidth
- Many other devices restricted to defense and telecommunications companies

[1]

Existing SDRs: Low End

- HackRF One
  - Mid range price ($340)
    - Too expensive for mass deployment
  - Tx and Rx
  - 10 MHz usable bandwidth
  - 1 MHz to 6 GHz carrier

- RTL-SDR
  - Entry level price ($30)
  - 1.6 MHz usable bandwidth
  - Rx only
  - Exploits TV tuner chip
  - Receive over the air AM/FM radio and TV

[2] [3]

Market Void

- Existing SDRs are too expensive to be implemented on large scales
- They require too much power to be used in low-power applications
- Offer much more performance than is necessary to support many applications
  - Many wireless sensors don't need to transmit much data very often
- Some are limited in capability
  - Rx only
  - Limited frequency bands
    - See RTL-SDR

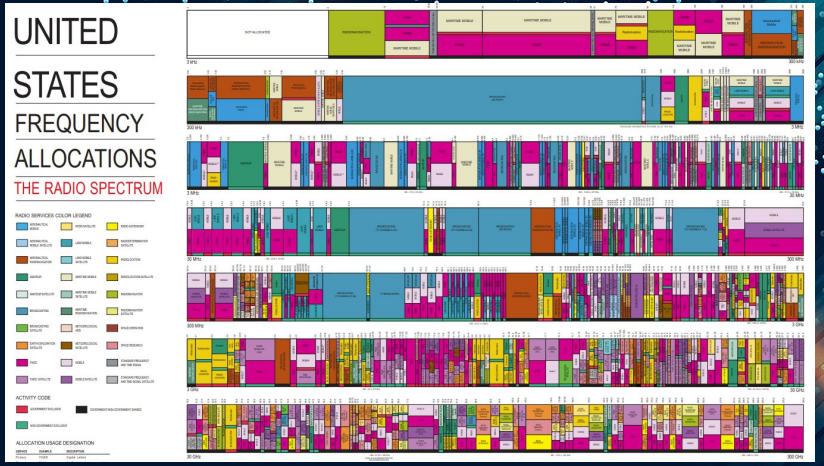
- Non-SDR, low cost, low performance radios are commonplace
  - Do not provide same flexibility as SDR
  - Locked in to specific protocol, modulation scheme, etc

Project Design Objectives

- Create a barebones SDR using smallest number of specialty components
  - Simplified RF Front End (RFFE)
- Keep cost low
  - Fill aforementioned market void
- Bidirectional communication at 1 kbps
  - Modest performance target
    - Suitable for applications such as sensor networks
    - Again, fit market niche
- Target 915 MHz ISM band
- Use commodity microcontroller's onboard DAC and ADC
  - Simulate components already deployed in embedded applications
  - STM32 32-bit Arm Cortex-M4
  - No need for high performance FPGA

Design Challenges

- Microcontrollers have limited RAM
  - Dictates the length of signal buffers
  - Varies between chips
    - Can alleviate with external memory
- Data converters have limited sample rates
  - Determines usable bandwidth
  - Cannot do Direct Digital Synthesis (DDS)
- SDRs require significant computational resources
  - Target limited bitrate
- Government frequency allocations
  - Extremely limited unlicensed bands
  - 902 MHz - 928 MHz ISM band selected
    - Still has power restrictions

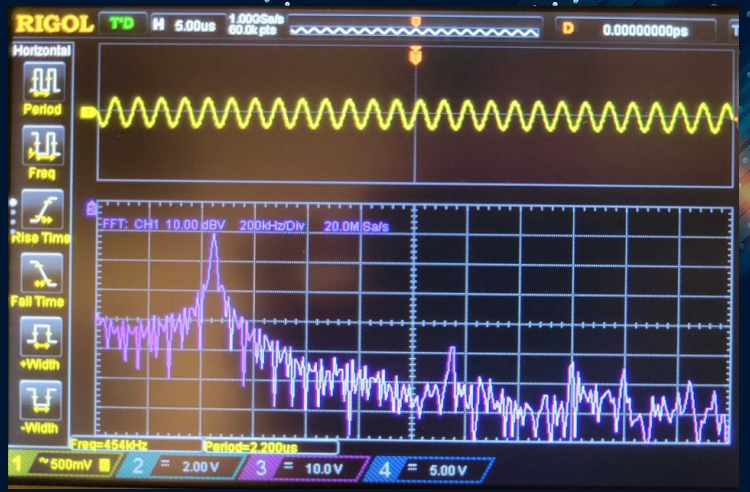


[5]

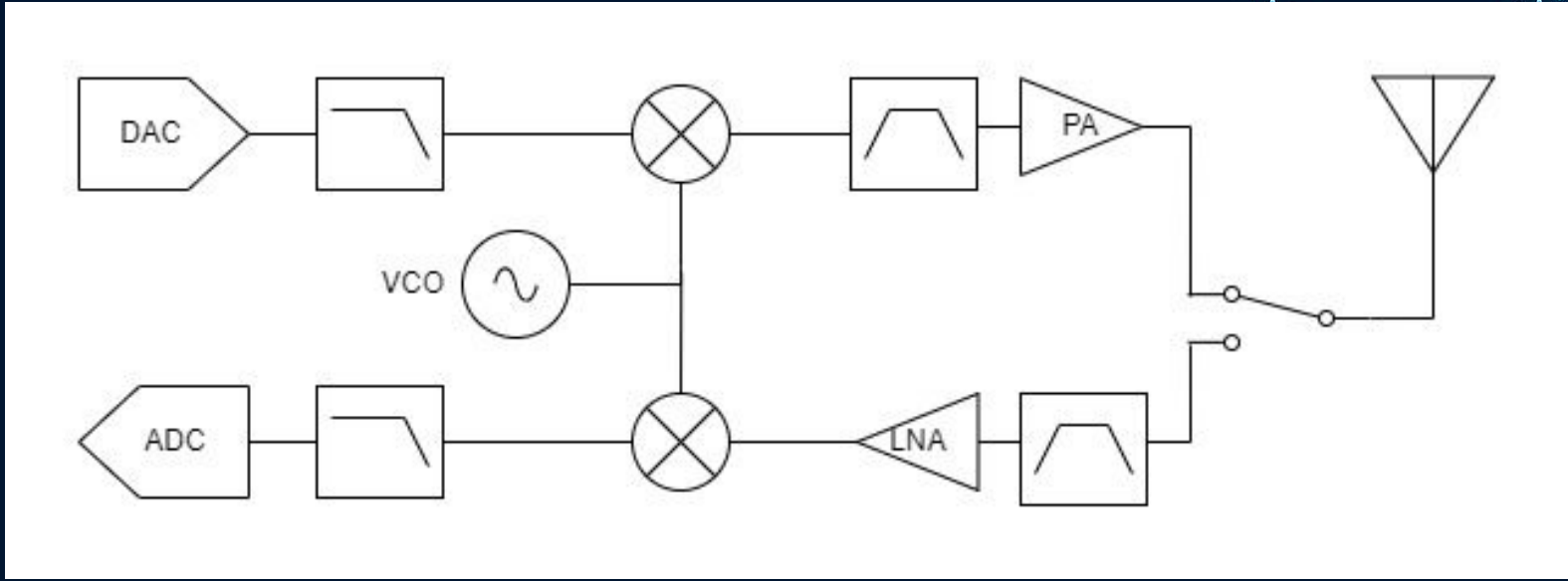
Data Converter Characterization

- How much useful bandwidth can we get out of the ADC/DAC?
- Limited by samples/second and Nyquist Theorem
- Software-triggered conversion is too slow
  - 900 kHz max
  - Occupies main processor for duration of transmission
  - Subject to interruption

- Using Direct Memory Access (DMA)
  - Leaves main processor free to encode/decode data
    - Or perform other tasks
  - Not subject to same interrupts
  - Can precisely set sample rate using timer
  - 5 MSPS max ADC, 6 MSPS max DAC
    - Using 4.5 MSPS for both
      - Yields 2.25 MHz usable bandwidth

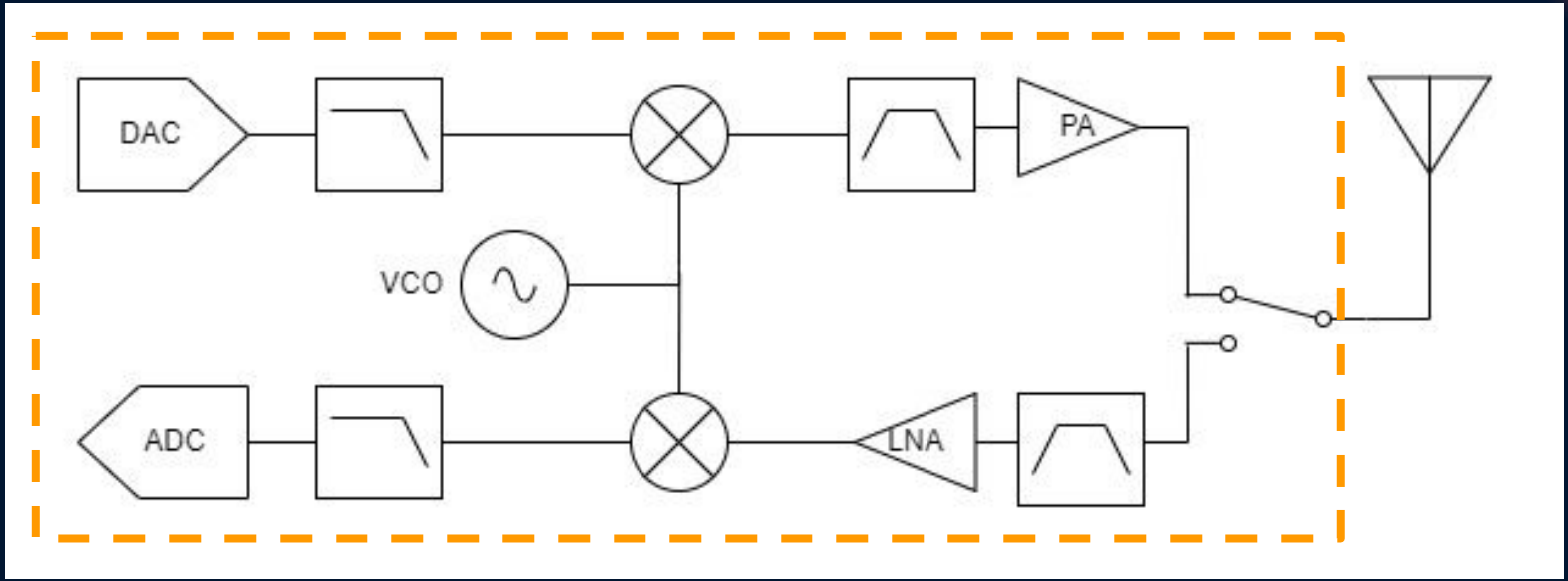


Hardware Overview



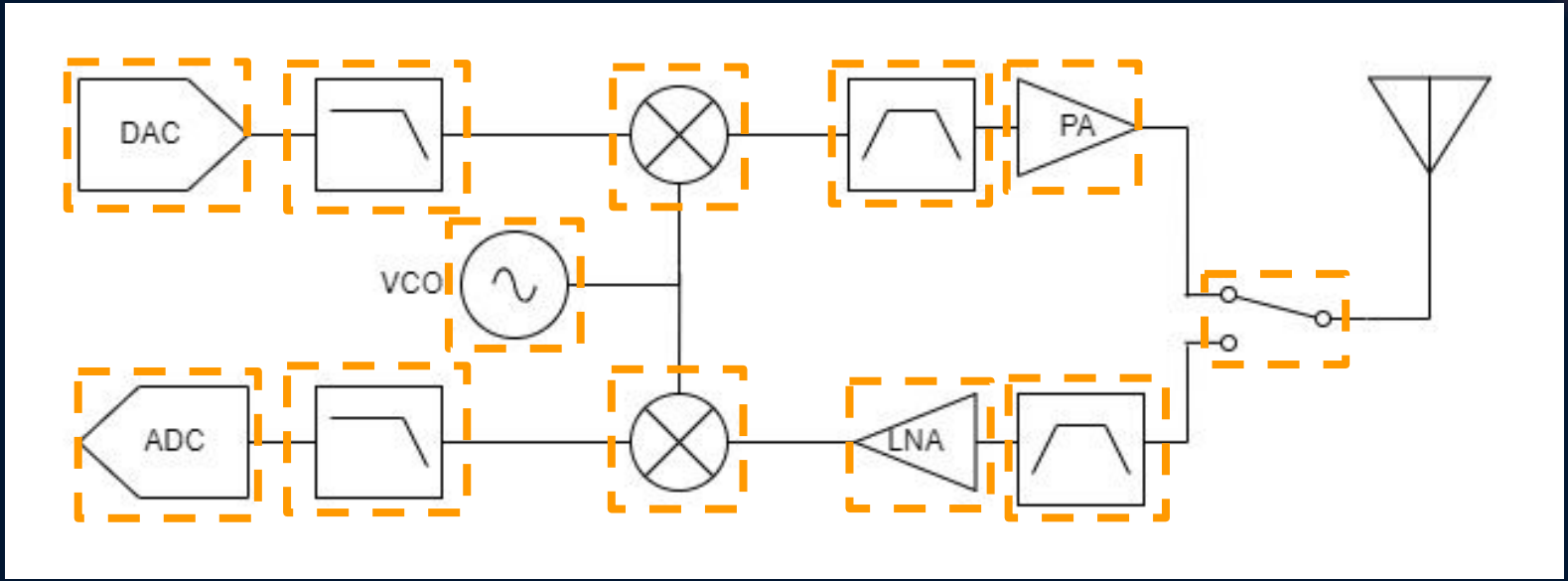
Hardware Trade-off: Level of Integration

- RF components can be highly integrated (one ASIC)
  - Pros: Lower power, smaller area, less board-level complexity
  - Cons: Little flexibility



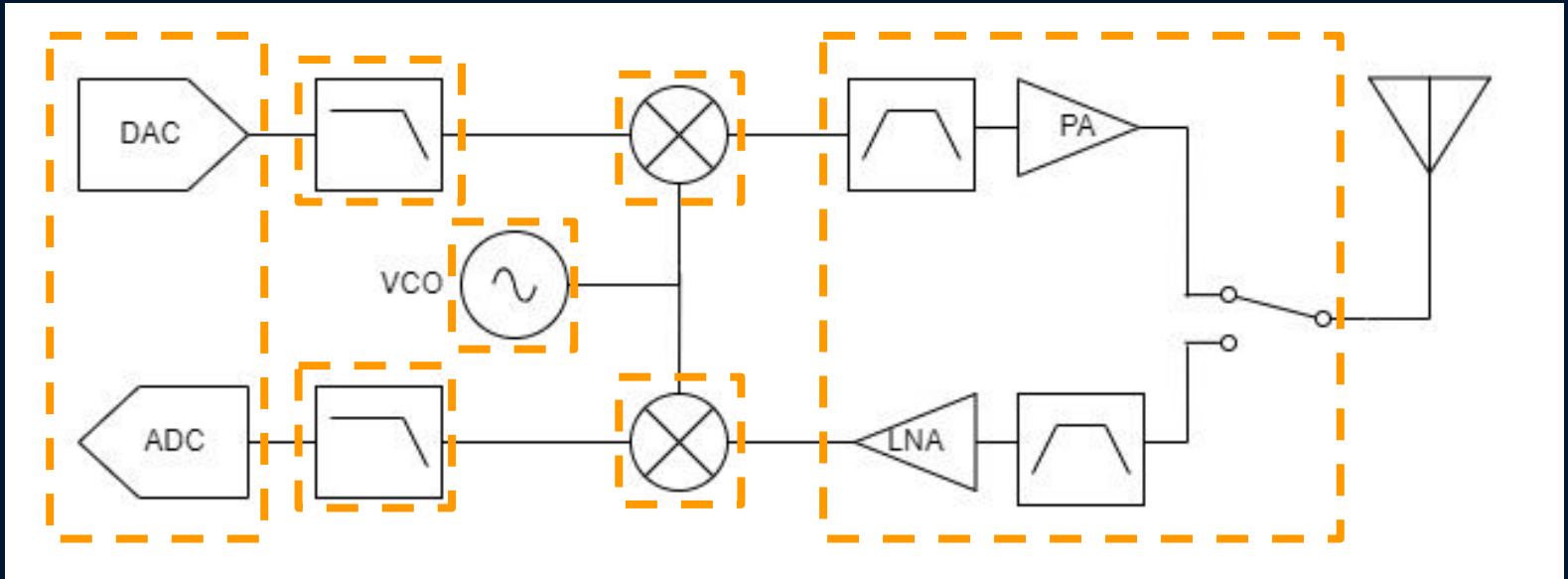
Hardware Trade-off: Level of Integration

- Very loosely integrated (discrete components)
  - Pros: Maximum flexibility
  - Cons: Higher power, more board area, higher board-level complexity

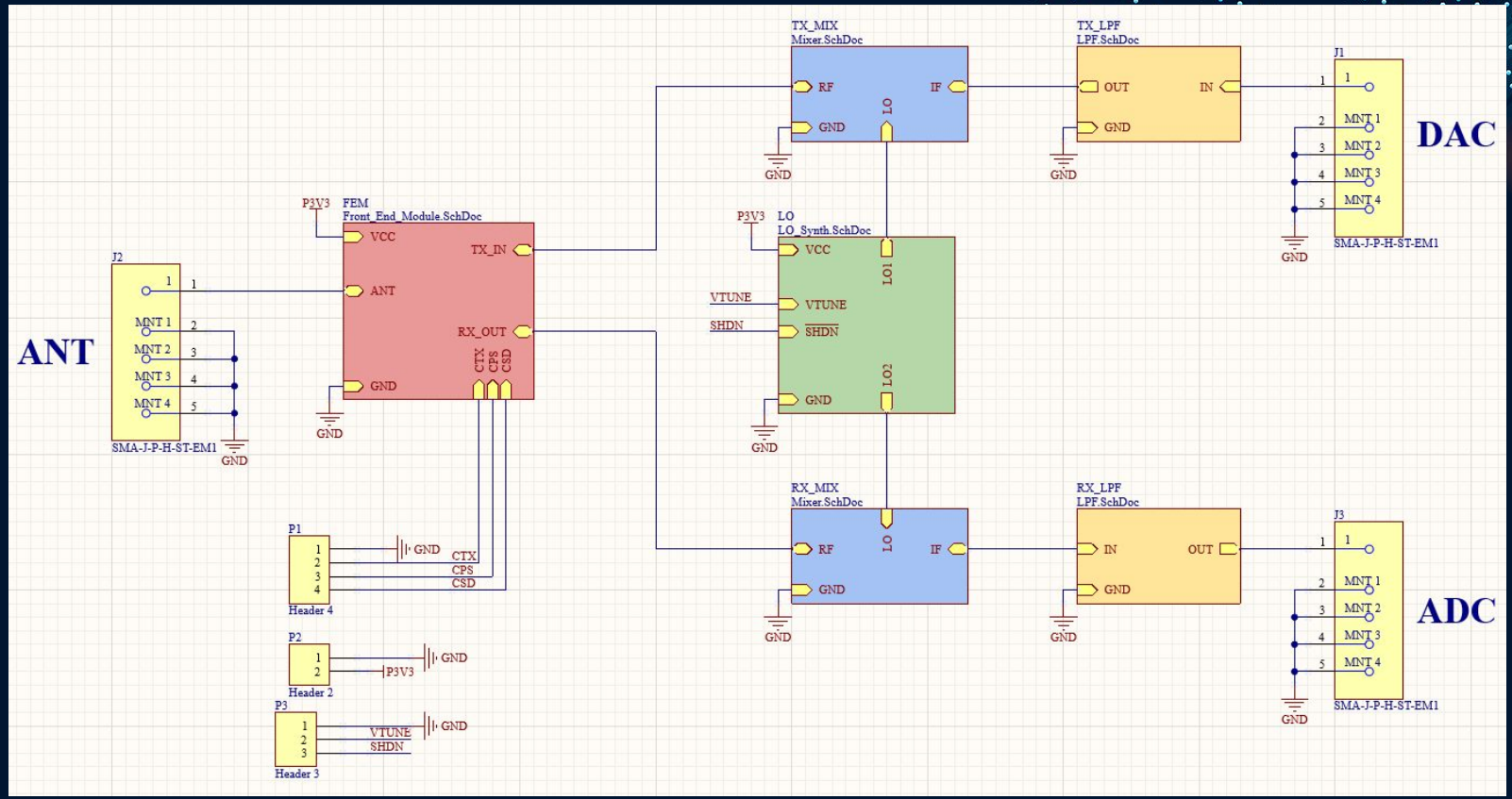


Hardware Trade-off: Level of Integration

- Or somewhere in between (Components grouped into multiple devices)
  - Pros: Moderate flexibility, power consumption, complexity, board area
  - Cons: Moderate flexibility, power consumption, complexity, board area
- The approach we pursued

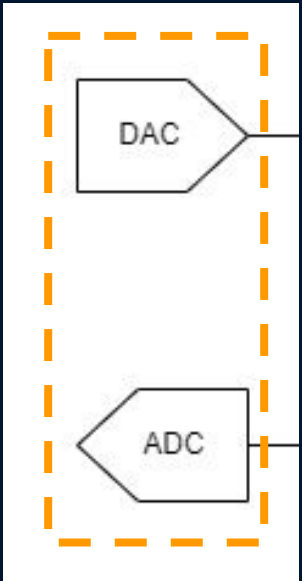


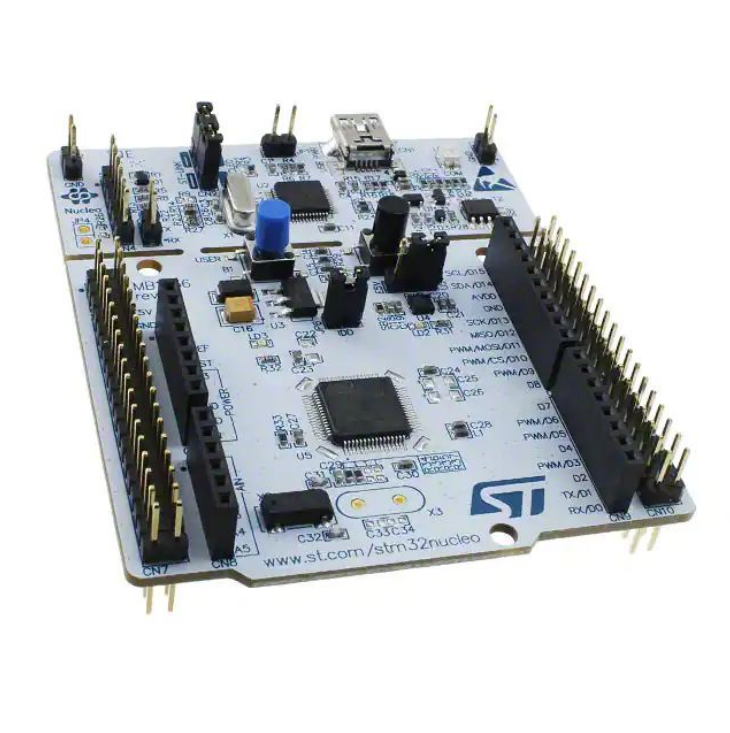
Schematic Block Diagram



Data Converters

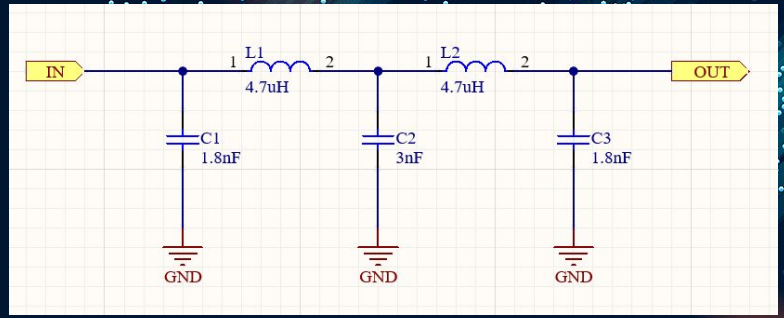
- STM32F334 - Arm Cortex-M4 32 bit dev board
  - Onboard 5 MSPS 8-bit ADC
  - Onboard 8-bit DAC
  - Max core clock 72 MHz
  - Supports DMA

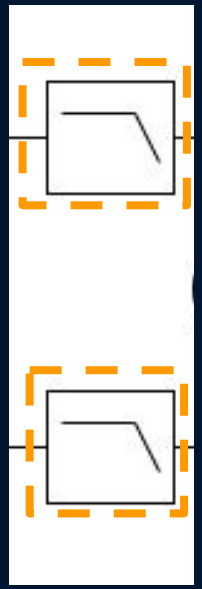


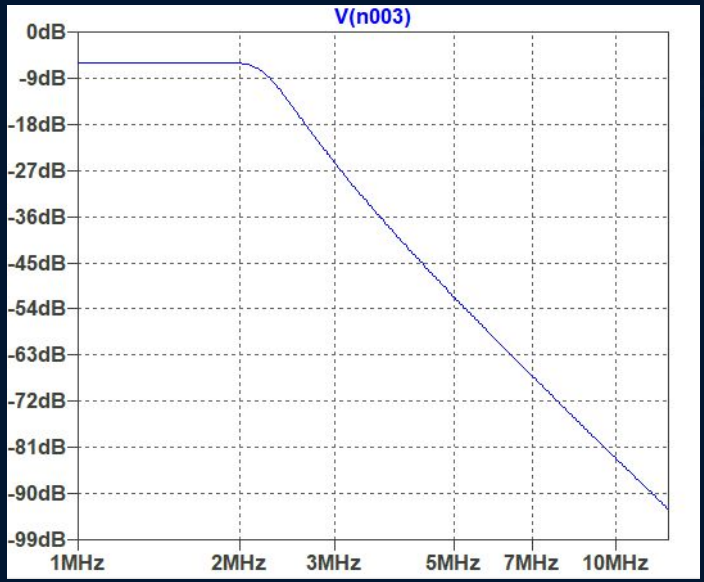


Filtering

- 5th order LC Chebyshev low pass filter
- Cutoff frequency at 2 MHz
  - Obey Nyquist limit of data converters

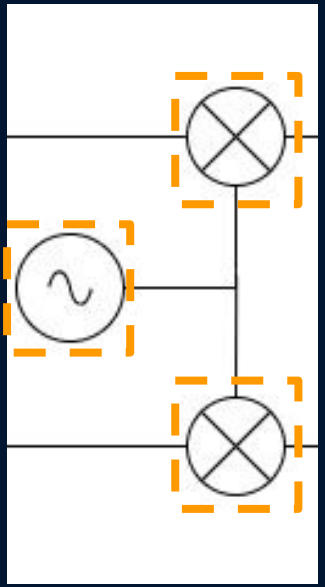




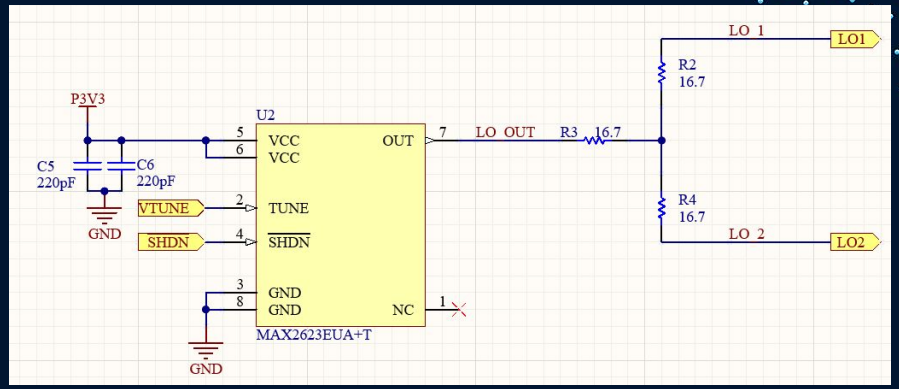


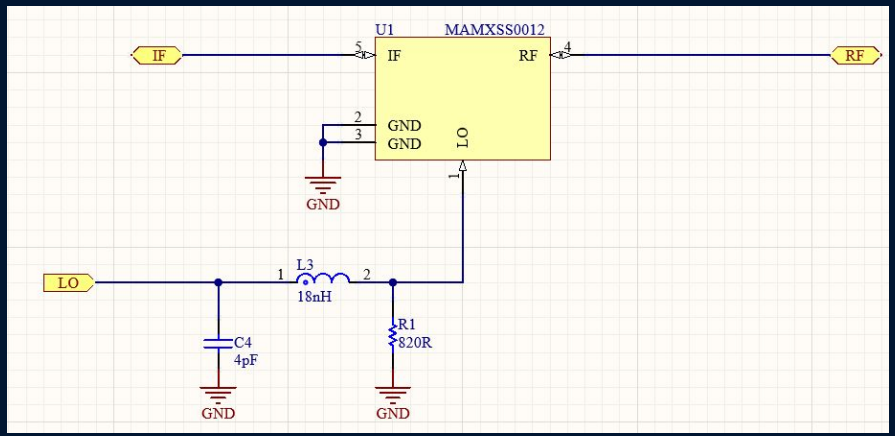
Frequency Conversion

- MAMXSS0012 Passive Mixer
  - 200 MHz - 1 GHz RF and LO
  - DC(!) - 200 MHz IF
- MAX2623 Voltage Controlled Oscillator
  - Tuning range 885 MHz to 950 MHz which covers entire 900 MHz ISM band



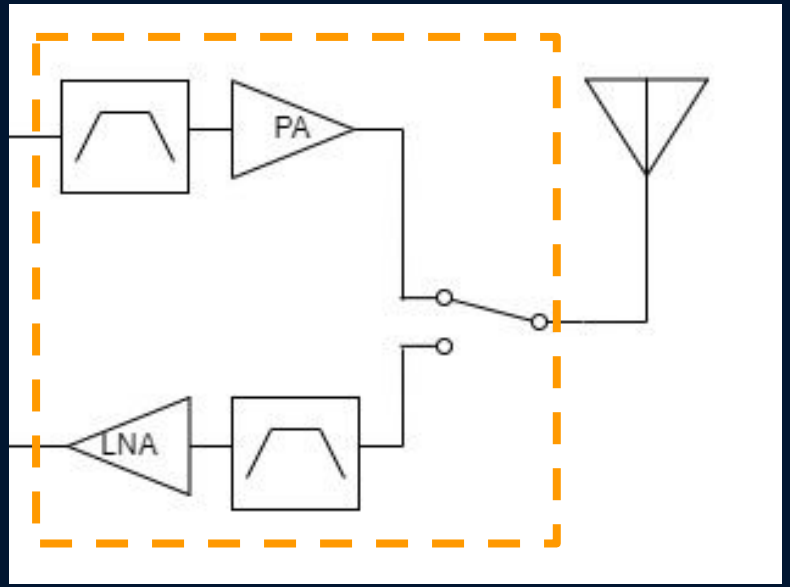
Frequency Conversion



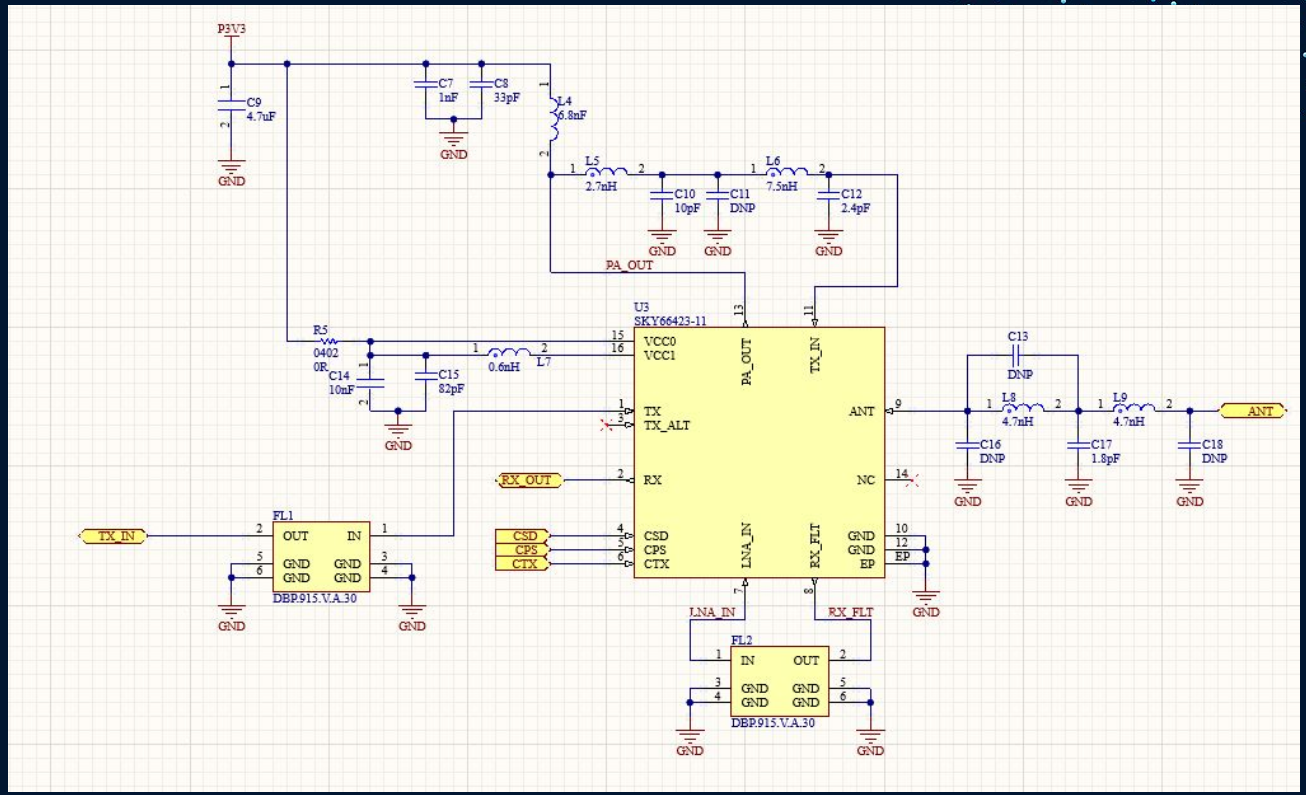


Front End

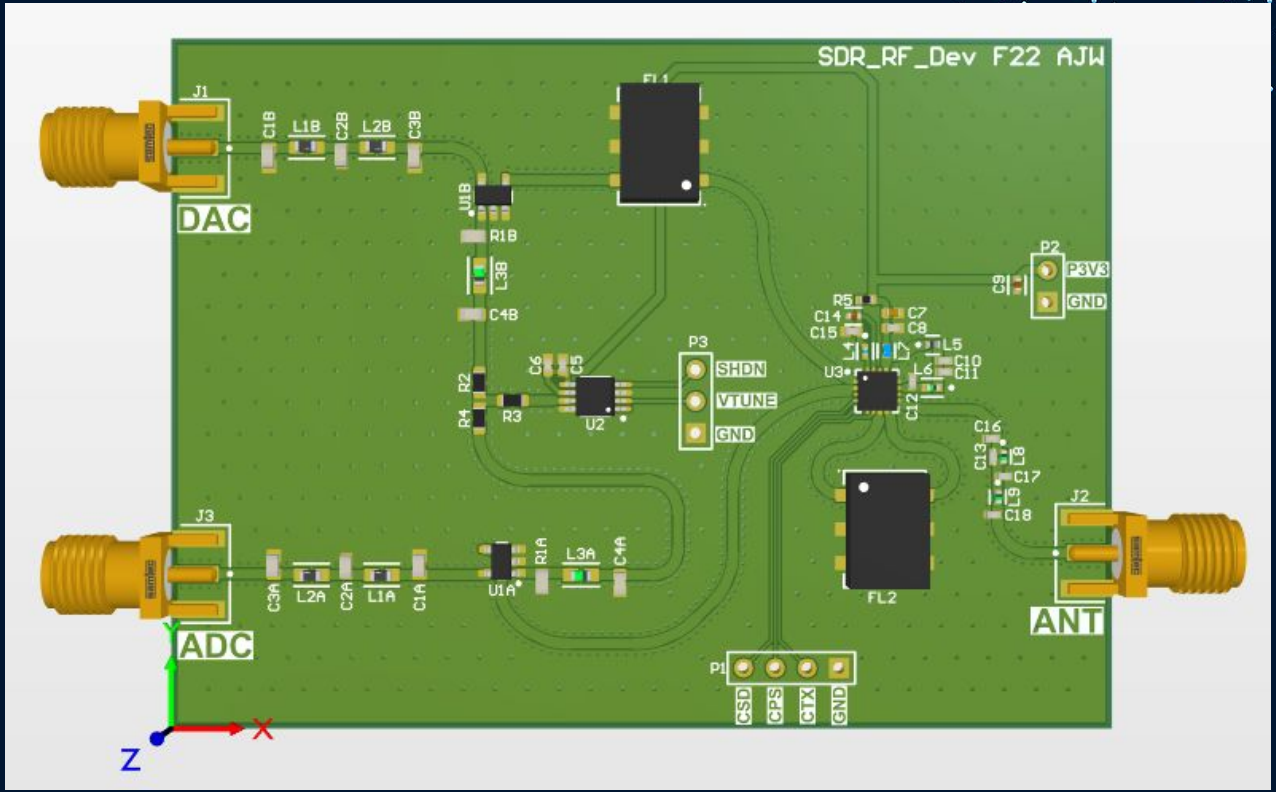
- Skyworks SKY66423-11 Front End Module
  - Includes TX/RX Switching
  - Power Amplifier with 29 dB gain
  - Low Noise Amplifier with 18 dB gain
- Taoglas 915 MHz Bandpass filters
  - +/- 5 MHz 3 dB Bandwidth



Front End

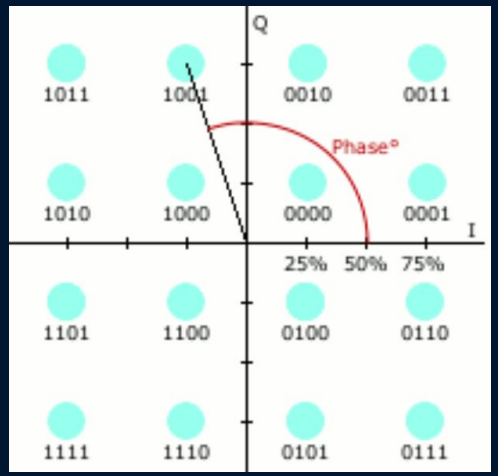

P3V3
C9
4.7uF
GND
C7
1nF
C8
33pF
L4
6.8nF
L5
2.7nH
C10
10pF
C11
DNP
L6
7.5nH
C12
2.4pF
PA_OUT
U3
SKY66423-11
R5
0402
0R
C14
10nF
C15
82pF
L7
0.6nH
VCC0
VCC1
TX
TX_ALT
RX
CSD
CPS
CTX
PA_OUT
TX_IN
LNA_IN
RX_FLT
ANT
NC
GND
EP
C13
DNP
L8
4.7nH
L9
4.7nH
C16
DNP
C17
1.8pF
C18
DNP
ANT
RX_OUT
CSD
CPS
CTX
TX_IN
FL1
OUT
IN
GND
DBP.915.V.A.30
FL2
IN
OUT
GND
DBP.915.V.A.30
LNA_IN
RX_FLT

Hardware Layout



Digital Modulation Basics

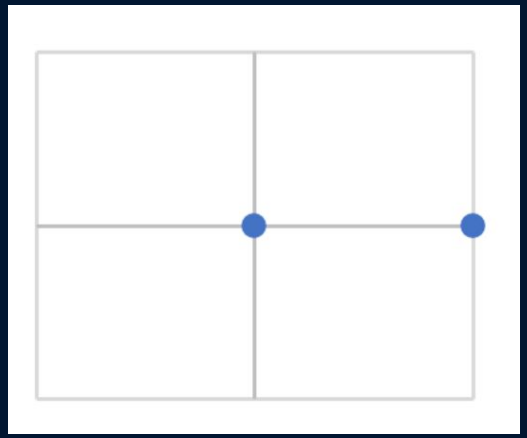
- Encode a sequence of bits on a carrier wave
- Number of states can vary between 2 and 4096+
  - Higher numbers require very high performance equipment
- Amplitude (ASK), Phase (PSK), and Frequency (FSK) keying possible
  - As well as combinations such as QAM (Quadrature Amplitude Modulation)

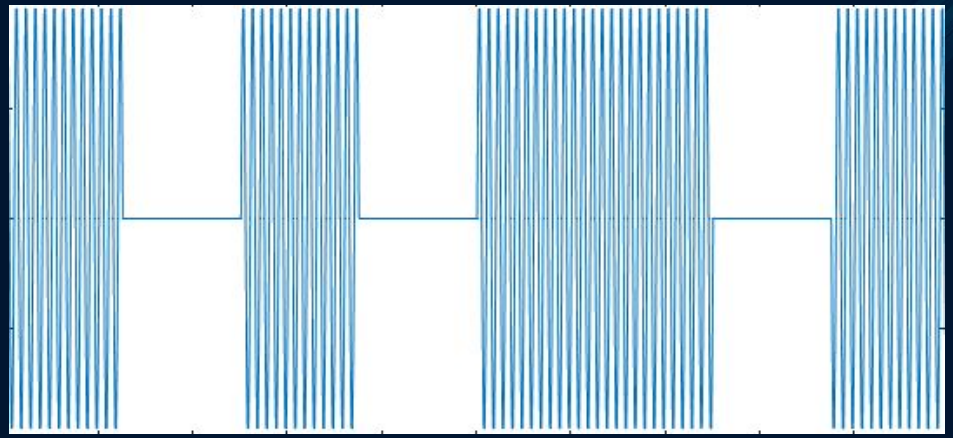


[6]

Basic ASK

- Amplitude Shift Keying (ASK) is the simplest form of digital modulation
  - Digital counterpart to AM
- States represented by the amplitude of the carrier wave at a given time
- Chosen as a starting point due to its simplicity
  - Proof of concept

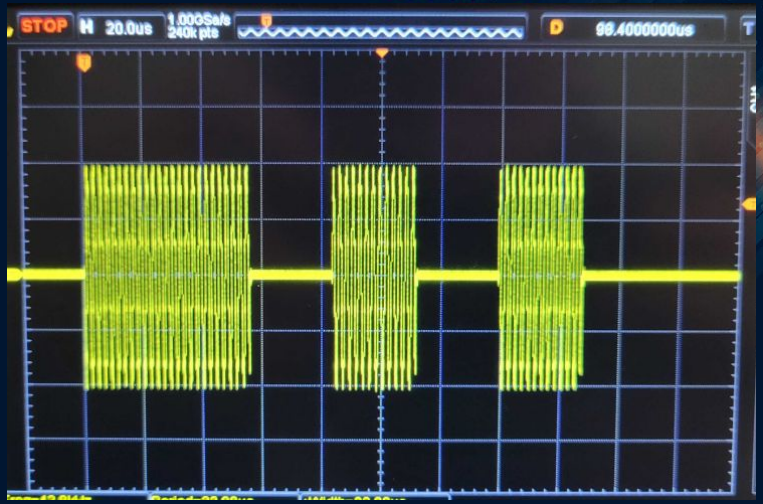




AFSK Implementation - TX (DAC)

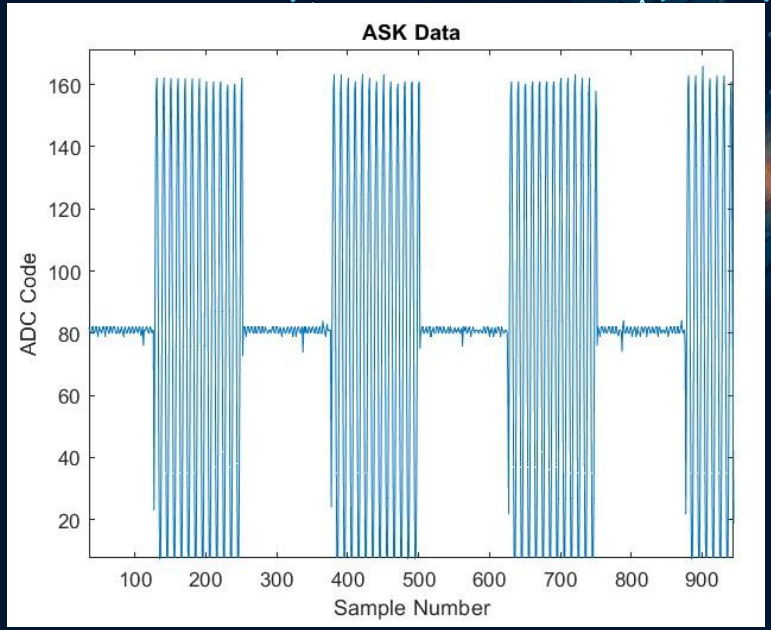
- Generate carrier signal on system start
  - Re-use computation
  - Cost of RAM usage

- For each transmission:
  - Read in desired 8-bit message
  - Mask carrier signal using target data to create transmittable waveform
  - Initiate DMA transfer
    - Outputs buffer to DAC at 4.5 MSPS



AFSK Implementation - RX (ADC)

- For each transmission:
  - Initiate ADC conversion
  - Fill buffer with samples using DMA
    - 4.5 MSPS
  - Find moving windowed amplitude
    - Effectively the signal envelope
  - Sample decoded envelope to get encoded data
  - Return received 8b data



Performance

- Transmit Side
  - Modulating 8 bits of data takes 169 us
  - Theoretical max of 47.34 kbps
- Receive Side
  - Demodulating 8 bits of data takes 705 us
  - Theoretical limit of **11.35 kbps**
- Currently limited by receive signal processing and demodulation
- 11.35 kbps exceeds baseline design target of 1 kbps
- Room for improvement with more sophisticated modulation and demodulation
  - Increase density of symbols in time
  - Increase number of symbol states
    - 4+ ASK
    - PSK/FSK

Power Consumption

- Transmit (Microcontroller) - 105 mW
- Receive (Microcontroller) - 89 mW
- RF Components
  - Not yet tested, these are estimates
  - Front End
    - 924 mW Tx
    - 17 mW Rx
  - VCO - 26 mW

- Total Transmit - 1055 mW = 93 uW/bit*
- Total Receive - 132 mW = 12 uW/bit*

- All active components have sleep states drawing < 1 mW

* Assuming best-case 11.35 kbps data rate

Cost Breakdown

- Microcontroller and PCB - $0
  - Presumed to already exist in target application
    - Assuming MCU has capable ADC and DAC
- Front End RFIC - $2.94
- Mixer (x2) - $5.46
- VCO - $6.84
- Bandpass Filters (x2) - $3.08
- Passives - ~ $1

- Total of $19.32
- All costs given in single quantity, will significantly reduce at volume
- Can likely optimize components further for cost
  - Remove bidirectional communication capability

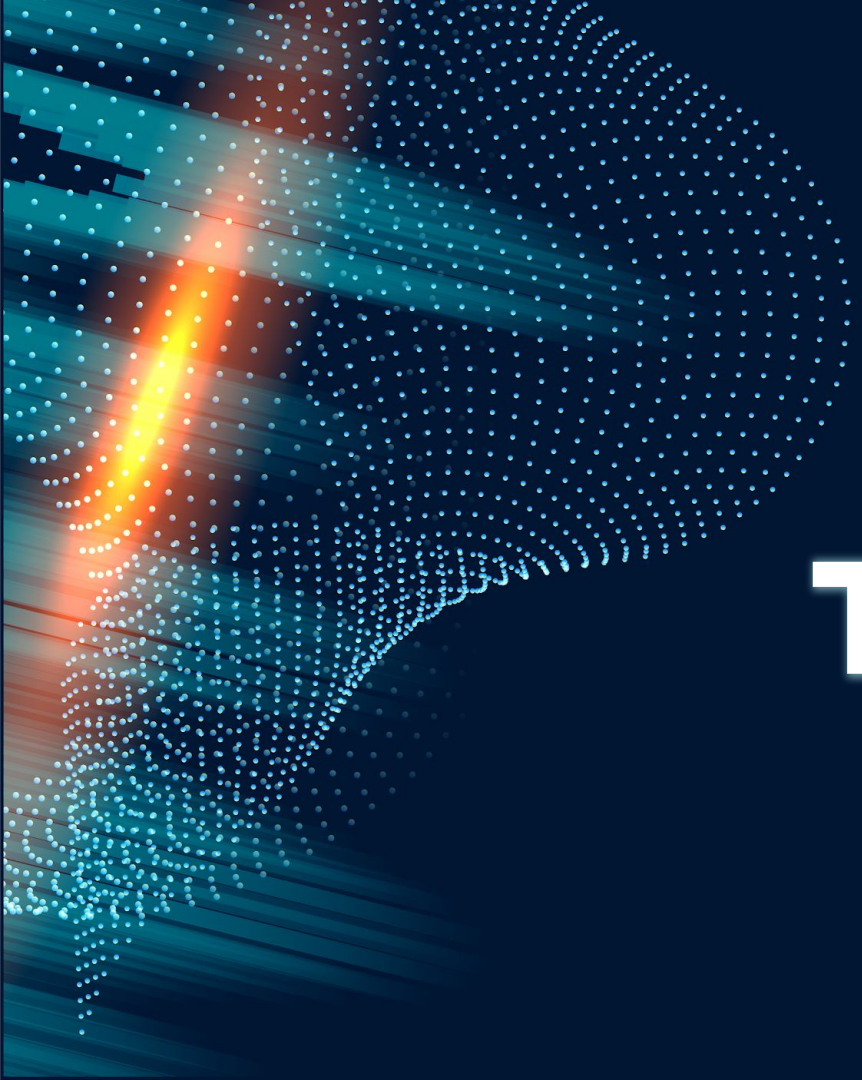
Ongoing and Future Work

- Ongoing this semester:
  - Connect RF hardware to microcontroller and test
  - Range testing
  - Validate power consumption estimates
  - Perform additional testing
    - Throughput
    - Error rate

- Future development efforts
  - Explore additional modulation schemes
    - Definite room for performance improvement
  - Integrate microcontroller and RF hardware into single board
  - Add support for standard SDR software toolchains
    - GNU Radio

Conclusion

- SDRs offer a lot of flexibility
- Not many consumer products exist, those that do are not affordable and often have niche targeted use cases
- Have tested ADC/DAC, need to test with RF hardware integration



Thank you!

Any questions?

References

[1] USRP:
https://www.ettus.com/all-products/un200-kit/

[2] HackRF One:
https://greatscottgadgets.com/hackrf/one/

[3] RTL SDR:
https://www.rtl-sdr.com/about-rtl-sdr/

[4] SDR Architecture:
http://www.analogictips.com/wp-content/uploads/2017/12/WHTH_FAQ-Receiver-Architectures_Pt2_Fig2.jpg

[5] Frequency Allocation:
https://www.ntia.doc.gov/files/ntia/publications/2003-allochrt.pdf

[6] QAM graphic
https://en.wikipedia.org/wiki/Quadrature_amplitude_modulation