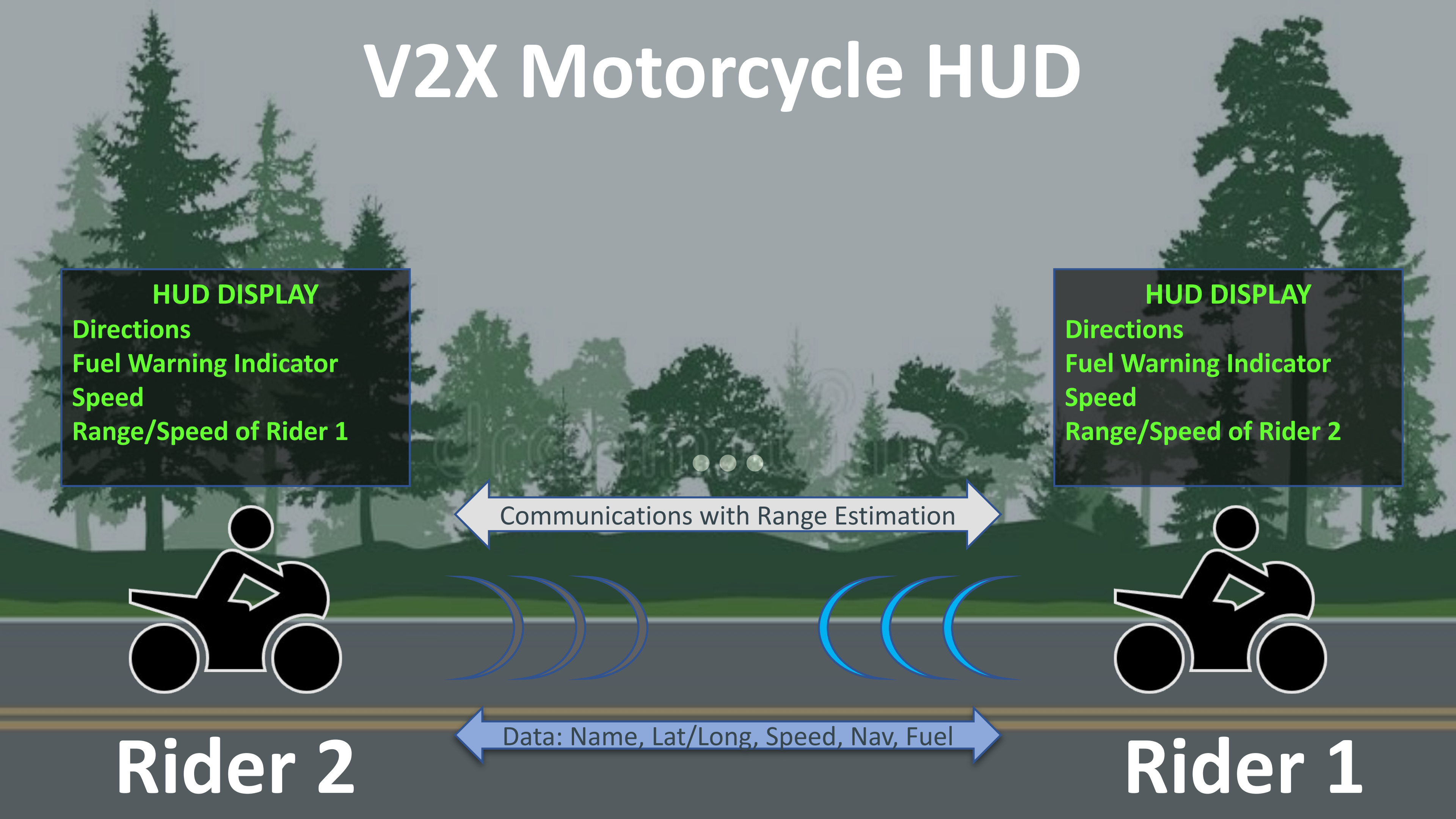
V2X Motorcycle HUD Weekly Updates

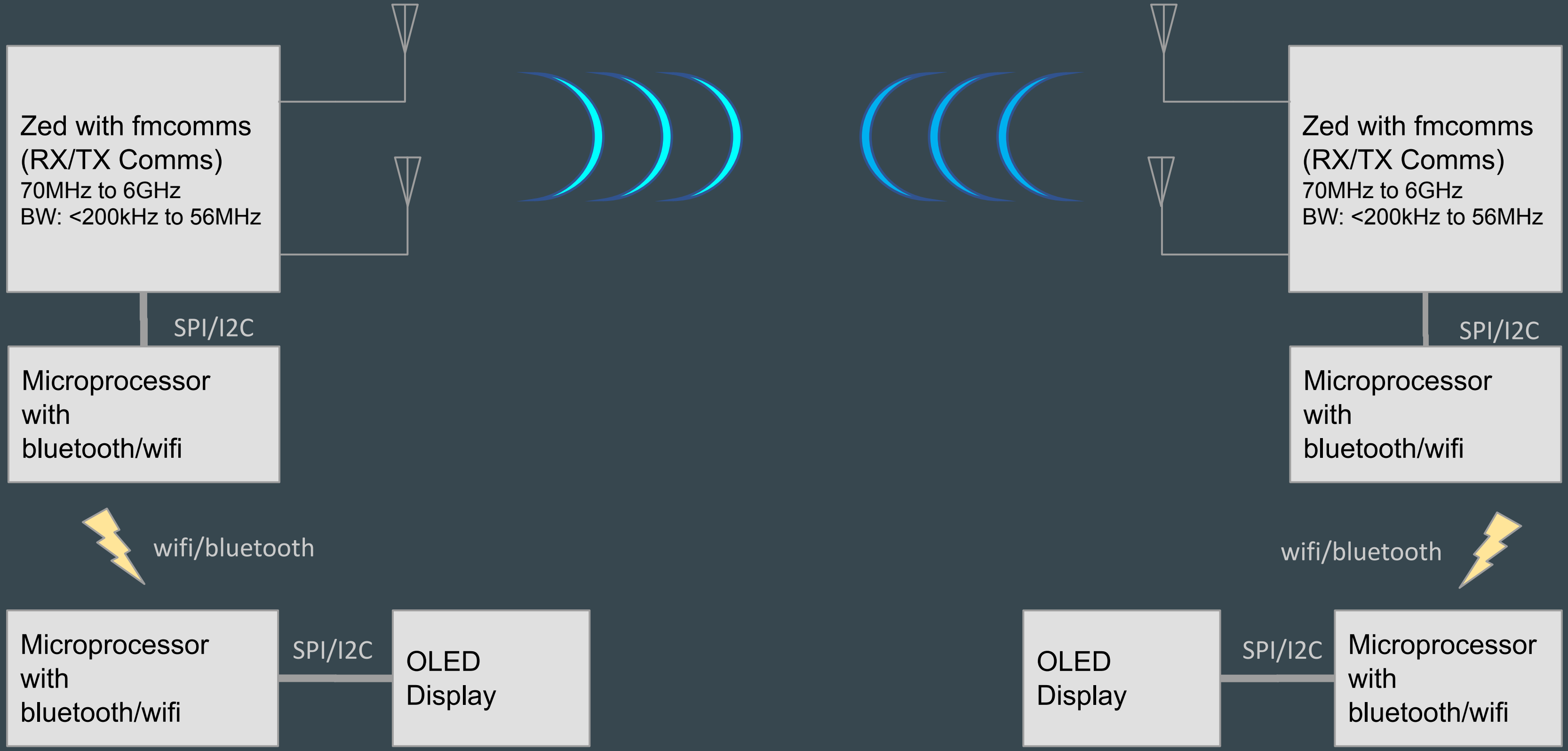
Ryan Hiser, Jacob Nguyen, Jorge Pacheco
UCSD MAS WES
1 April 2022


V2X Motorcycle HUD
HUD DISPLAY
Directions
Fuel Warning Indicator
Speed
Range/Speed of Rider 1
HUD DISPLAY
Directions
Fuel Warning Indicator
Speed
Range/Speed of Rider 2
Communications with Range Estimation
Data: Name, Lat/Long, Speed, Nav, Fuel
Rider 2
Rider 1

Motivation

- Motorcyclists often ride in groups. To effectively and safely ride it is often necessary to communicate. This takes many forms:
  - Hand gestures (hands off of handlebars)
  - High speed maneuvers
  - Intercoms (difficult with multiple riders)
  - Refuel periods (gas tanks size/consumption speed vary)
  - Modern cruise control is handled as a throttle lock
    - Riders at rear cannot easily keep consistent following distance
- A HUD system would allow for fewer distractions from the driving experience by displaying this information in a digestible format:
  - Driving Directions
  - Speed
  - Simple Messages
    - Others need to refuel
    - Following distance

Apparatus / System Diagram



Previous Quarter Accomplishments (Current Status)

- Waveform Description defined
- Simulink Model of Waveform (compatible with code generation)
- OLED and wireless link using arduino
- FMComms4 + Zedboard integration with PYNQ

Milestones (this quarter)

- TX Modulator
- RX Demodulator
- Complete Heads-up-display (HUD)
- TDMA Channel
- Two-way link
- Comms system integrated with HUD

Deliverables

- Git Repo: https://github.com/jtn017/capstone
- Final demonstration/video
  - HUD Demonstration
  - Data-link demonstration
- Project Report
  - Progress
  - Problems we faced
  - Solutions to problems
  - Future work

Gantt (timeline)

V2X Motorcycle HUD Project Plan

Periods are 1 Week Intervals

Period Highlight: 1

Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE	PERIODS 1	2	3	4	5	6	7	8	9	10
HUD	1	6	0	0	0%										
Mount OLED	1	6	0	0	0%										
Screen Interface/ICD	3	1	0	0	0%										
Integrate HUD with Zed	1	2	0	0	0%										
HUD Demo (PYNQ to HUD Data Display)	7	1	0	0	0%							DEMO 14th			
TX	1	2	0	0	0%										
Connect TX output (baseband + modulator) to DAC	1	2	0	0	0%										
Modulator DEMO	3	1	0	0	0%			DEMO 16th							
RX	1	6	0	0	0%										
AGC	1	2	0	0	0%										
Matched Filter	1	2	0	0	0%										
Coarse Frequency Correction (FLL)	1	2	0	0	0%										
Timing/Symbol Recovery	3	4	0	0	0%										
Fine Frequency Correction (Optional)	3	3	0	0	0%										
PL to PS interface	3	3	0	0	0%										
Port Baseband RX processing (simulink code) to ARM	1	2	0	0	0%										
RX Demodulator Demo	7	1	0	0	0%							DEMO 14th			
TDMA MAC Layer	5	4	0	0	0%										
Verification and Validation	7	3	0	0	0%										
TWO-WAY LINK DEMO	9	1	0	0	0%									DEMO 27th	
Audio Input/Output	7	3	0	0	0%										
Final Demonstration (COMMS with HUD Integrated)	10	1	0	0	0%										FINAL DEMO

	1	2	3	4	5	6	7	8	9	10
Start Date	4/1/2022	4/8/2022	4/15/2022	4/22/2022	4/29/2022	5/6/2022	5/13/2022	5/20/2022	5/27/2022	6/3/2022
End Date	4/7/2022	4/14/2022	4/21/2022	4/28/2022	5/5/2022	5/12/2022	5/19/2022	5/26/2022	6/2/2022	
CLASS DATE	4/1/2022		4/16/2022		4/29/2022		5/14/2022		5/27/2022	6/3/2022

Next Sprint

- TX
  - Generate data and send from PS to DAC
- RX
  - Modulator
    - Implement AGC (digital and/or fmcomms method)
    - Implement Matched Filter
    - Implement Coarse Frequency Correction
    - Test PL to PS using dummy data
  - Baseband
    - Create test vectors (IO)
      - Setup Simulink model with actual data
      - Process RX Baseband data and display on HUD
- HUD
  - Integrate HUD with PYNQ
  - Start prototype mounts

ACTIVITY	PERIODS 1	2
HUD		
Mount OLED		
Screen Interface/ICD		
Integrate HUD with Zed		
HUD Demo (PYNQ to HUD Data Display)		
TX		
Connect TX output (baseband + modulator) to DAC		
Modulator DEMO		
RX		
AGC		
Matched Filter		
Coarse Frequency Correction (FLL)		
Timing/Symbol Recovery		
Fine Frequency Correction (Optional)		
PL to PS interface		
Port Baseband RX processing (simulink code) to ARM		
RX Demodulator Demo		
TDMA MAC Layer		
Verification and Validation		
TWO-WAY LINK DEMO		
Audio Input/Output		
Final Demonstration (COMMS with HUD Integrated)		
	4/1/2022 4/7/2022	4/8/2022 4/14/2022

Additional Slides

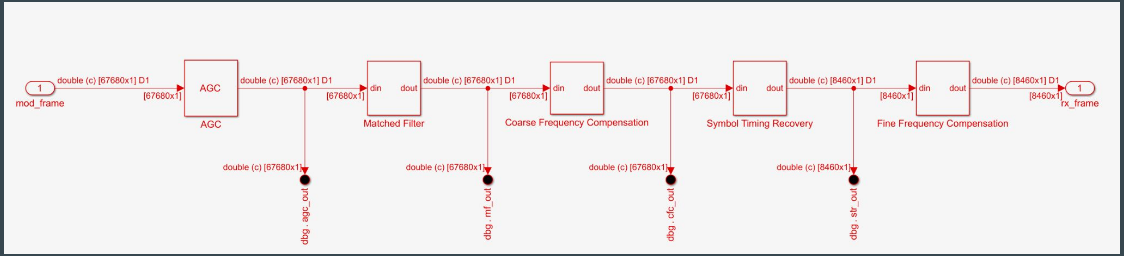
Current state of affairs

Waveform Description

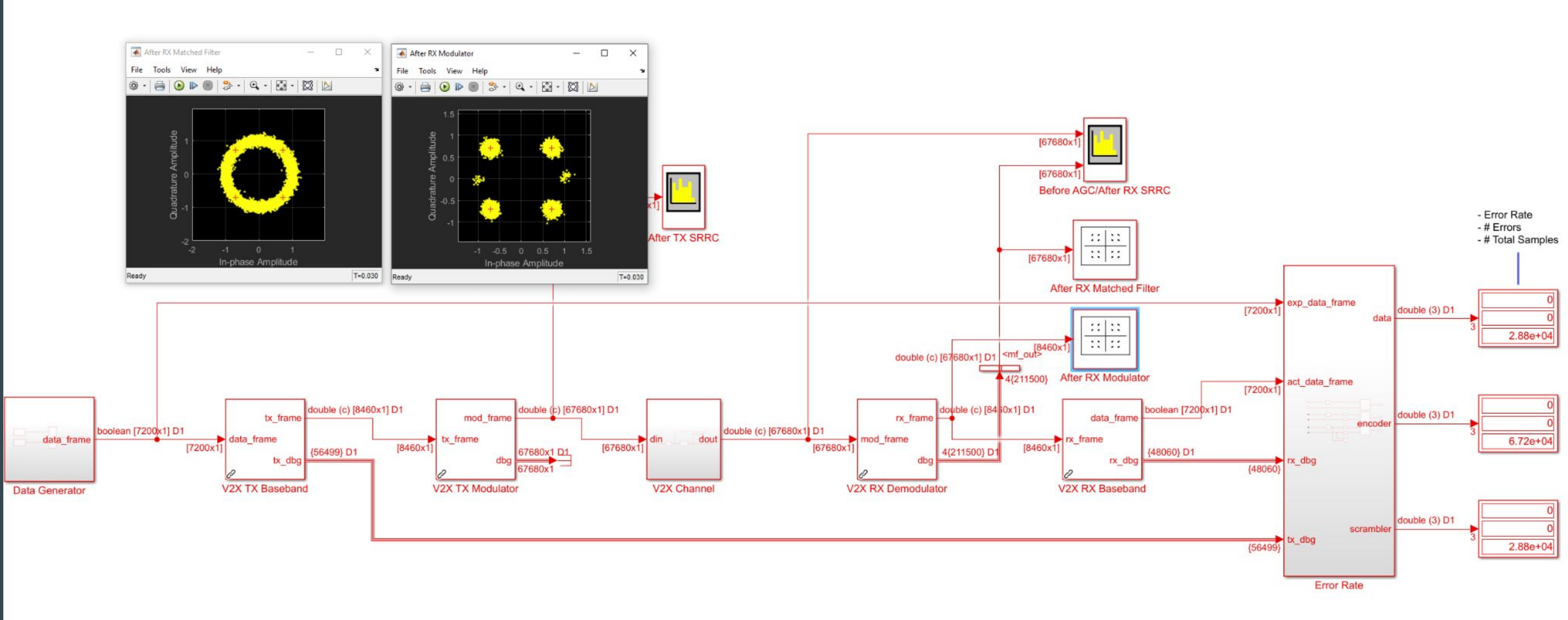
- Datalength: 7200 bits (Data + 0.01s audio).
  - Throughput: 7200 bits * (1/0.01s) = 720 kbps
- Minimum Required Sampling Frequency (before carrier mixer)
  - After TX processing/modulation: 67200 sym
  - Min Fs = 67200 sym * (1/0.01s) * 2 = 13.44 Msps
- Max Doppler Frequency:
  - Fc = 2.4 GHz (WiFi)
  - Max speed (x2): 200 MPH = 89.41m/s
  - Fd = 2,400,715 Hz (offset = 715 Hz)
- Max fe < 1/2T
  - 1/2T = fs/2L
  - fs/2L = 100 MHz/(2*64) = 781.25 KHz

Waveform Description (cont)

RX Demod

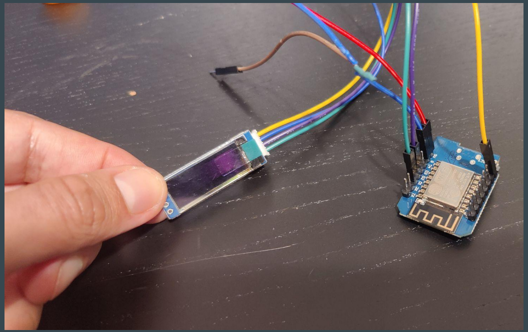


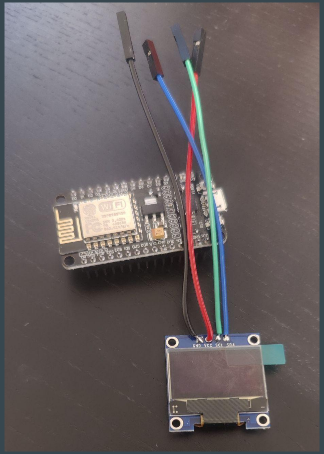
QPSK sim with phase + freq offset and AWGN



HUD Progress

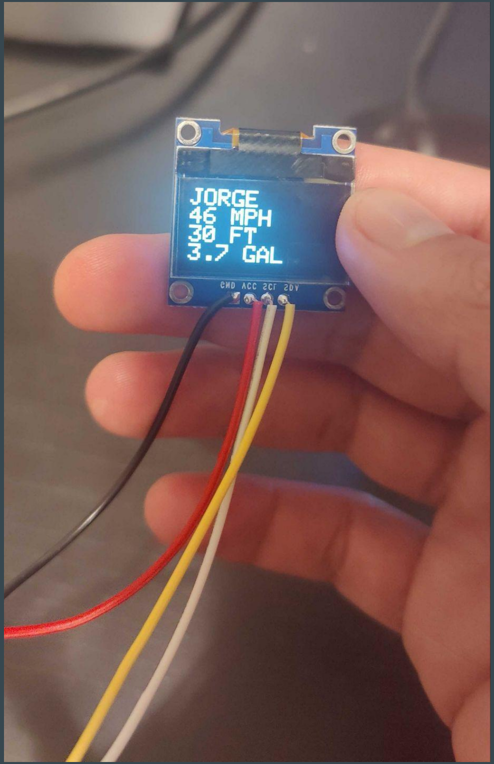
- Encountered issues with first batch of components
  - Assembled all parts and soldered components
  - Created test programs on Arduino IDE to configure ESP8266 (WIFi) and SSD1306 (OLED)
  - Couldn’t control OLED with SPI or I2C
  - Ordered a new set or parts with same controllers





HUD Progress (cont)

- Configured OLED to receive subset of data we plan to transmit
  - Created test programs to fiddle with WIFi server functionality
  - Connected WIFi module to local network and was able to toggle individual GPIO pins through a client connected to the network
- Next steps
  - Create a server to listen to our incoming traffic based on our requirements
  - Display text in real time on OLED
  - Work on enclosure



FMComms+Zed Board Integration

- Working SD Card PYNQ image
  - ADI provided base HDL project for FPGA.
  - Used Petalinux to generate boot files with desired FMComms4 device tree.
  - Integrated Petalinux and PYNQ Root Filesystem
  - Installed LibIIO library
  - Created a Transmit and Receive Demonstration.
    - Sends to Tx DMA → DAC
    - Reads from Tx DMA←ADC and stores to file.
    - On our github.