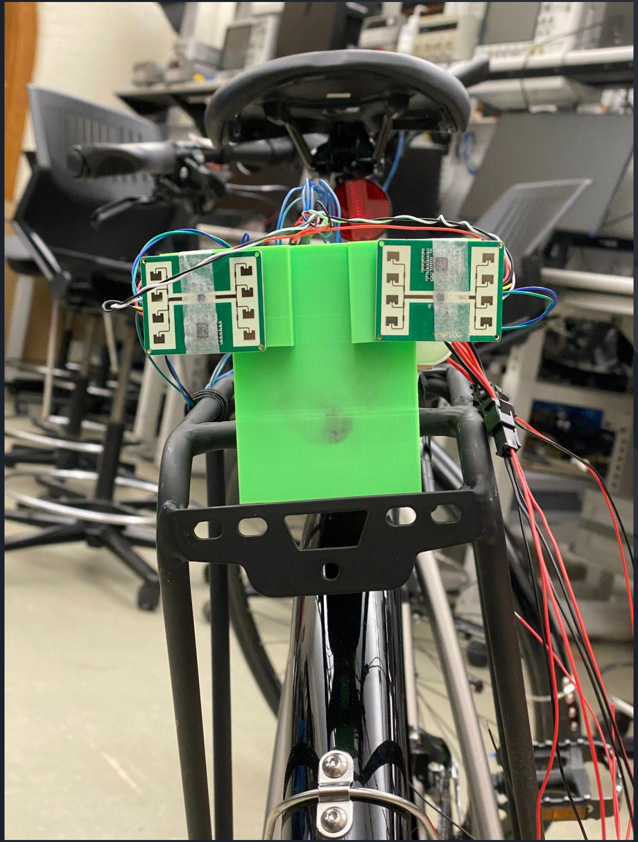
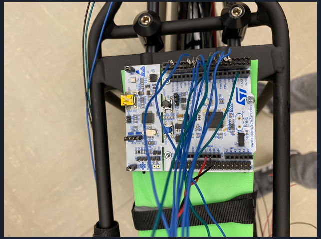
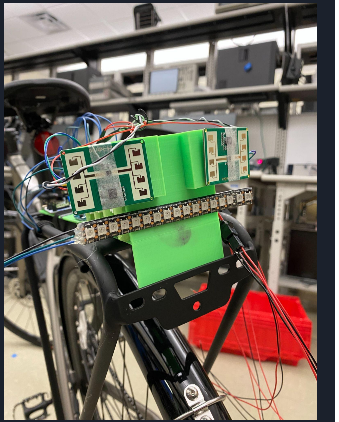
Bikewards View Final Presentation

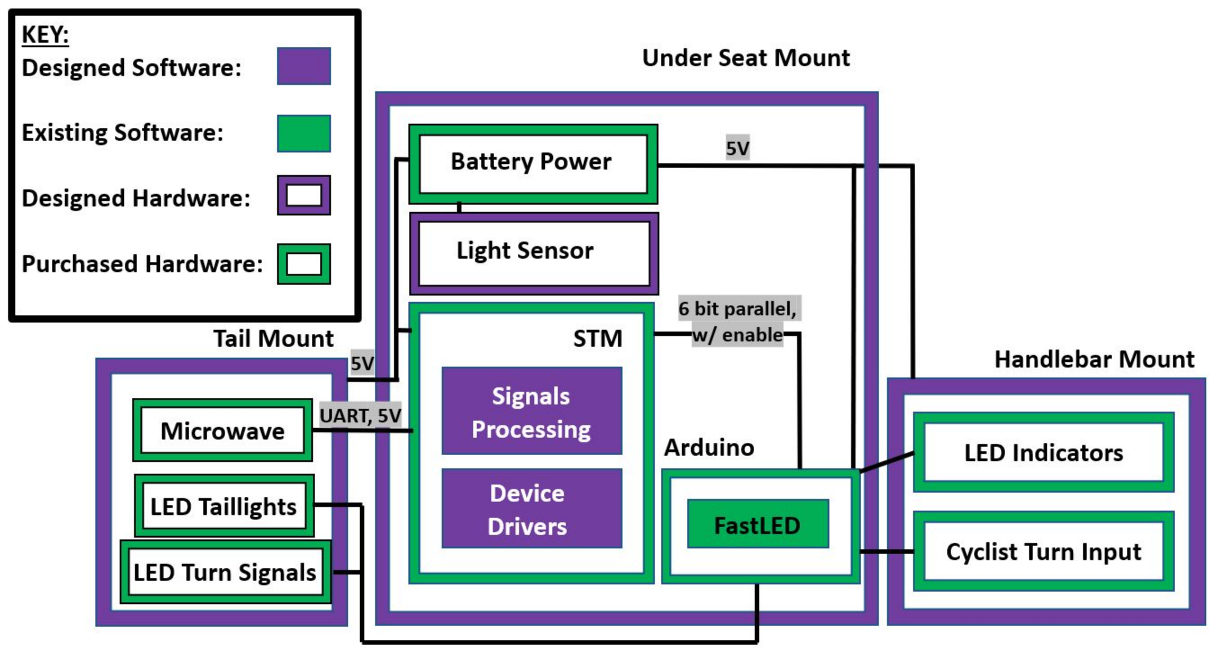
Albany Bloor, Emily Clayton, & Jason Xu

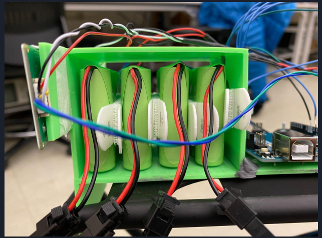
Introduction

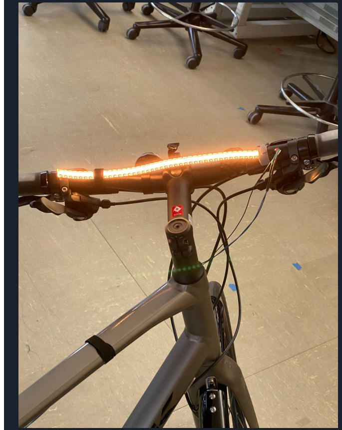












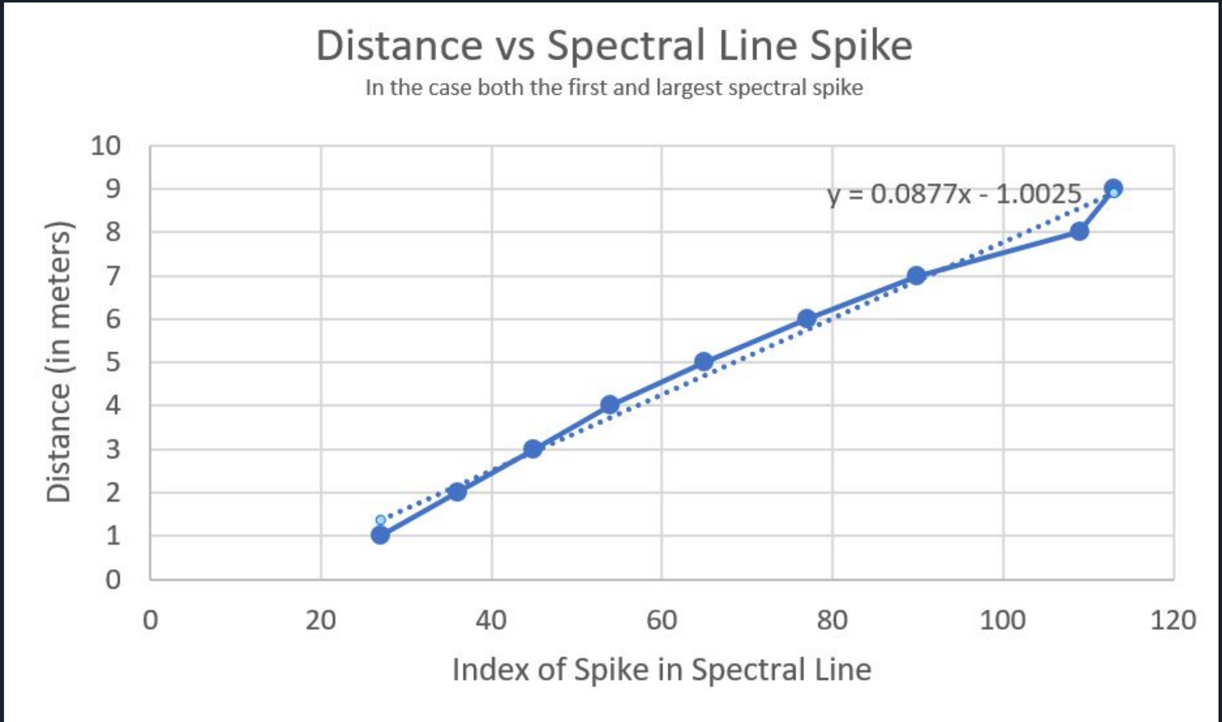
Final Design

UI

- LED output:
  - Red, Orange, Yellow
  - Front flashes in zone when initially transitioning to close detection
  - Green ON light on handlebars

  - Rear lights flash intermittently when object close
  - Light sensor modulates brightness of lights
  - Tail light and turn signals ON when dark outside, off when daytime (safety over power consumption at night)

Single Sensor Range Testing

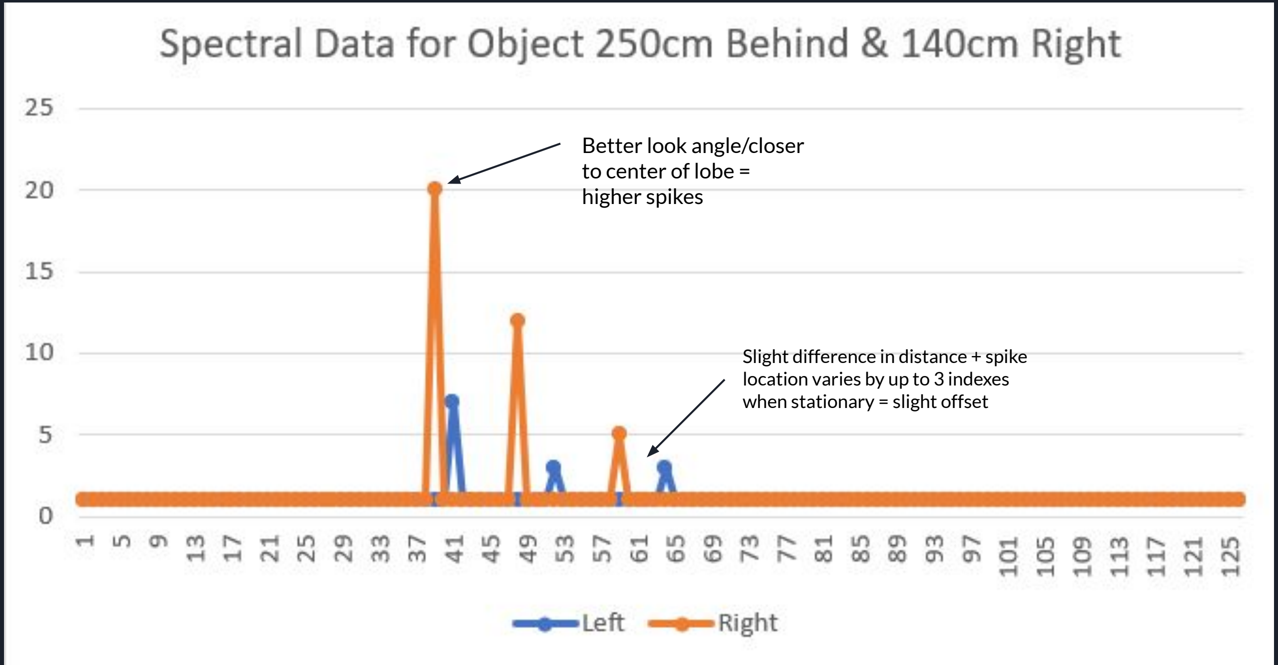


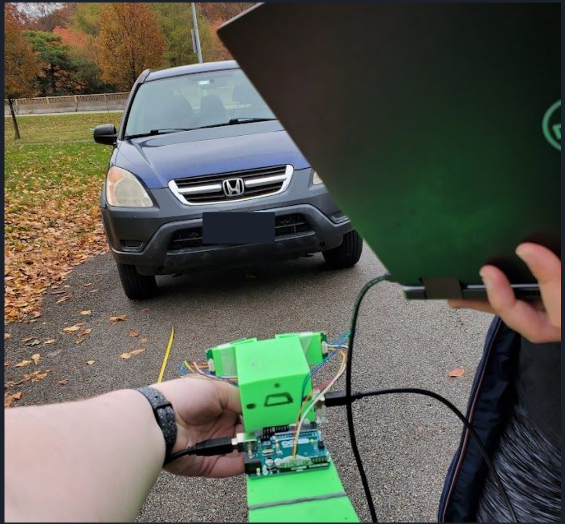
With Linear trendline and max spectral index of 126, range is only just over 10 meters.

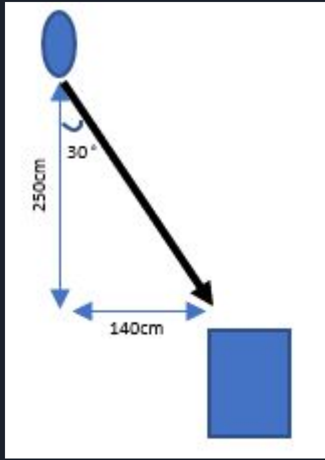
For both sensors we saw spectral line spike fall off end of data between 10 and 11 meters.

Dual Sensor Testing

Interference and Angles







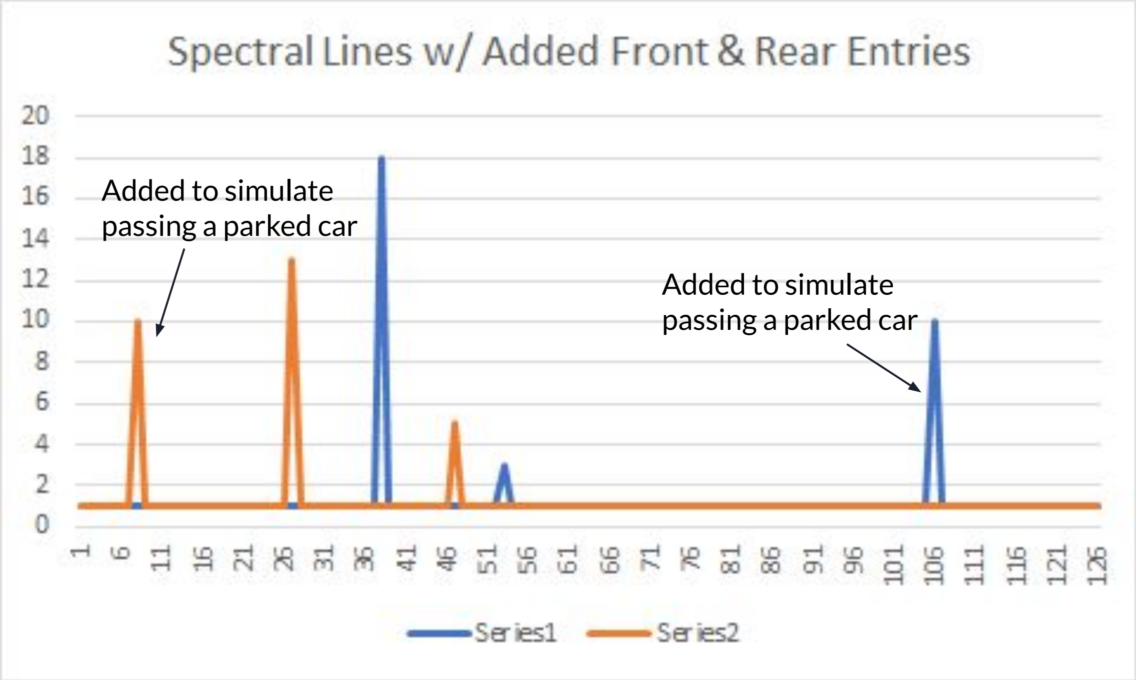
Objects visible to both sensors for most of FOV

At edge of FOV ~40 degrees and above only visible to closer sensor

Sensors angled to look 12 degrees apart

3dB Beamwidth of 78 degrees

Sensor Code Testing



Distances Returned by Code:
Blue - 37, 52,105
Orange - 7,26, 46

Velocities from code
(37 -26)= 11
(52-46) = 6

Display Levels
4.230769 = 10*11/26
1.304348 = 10*6/46
Display level in Hz in the inverse of the time a biker would have to respond before collision

Less than 0.001 secs to run

Combination for two sensors done with left display vals (4.230769, 1.66667) and right display vals (1.66667, 1.304348) correctly attributed 1.6667 to the center (overlap) section for display on the handle bars.

Requirements

Range	10.42 min	10 max
FOV	90 Degrees	90 Degrees
Battery Capacity	1 hr active, 8hr standby	TBD
Latency	250ms	~300ms (bottlenecked by sensors)
Accuracy	Never Fail within 10.42m range, allow up to 10 percent false positives	TBD

Trade-Offs

- **Using newer sensor technology (vs. LiDAR/Ultrasonic)**
  - Higher advertised performance at cheaper price
  - Limited documentation
  - Doesn’t work as intended, but not inadequately

- **Using Arduino as LED driver (vs. implementing FastLED on STM32)**
  - Slower performance - negligible due to low update rate requirements
  - Easier for us to implement with existing drivers
  - Problems with inter-device serial communication

- **Using STM32-F401RE (vs. STM32F0 series)**
  - Powerful processor - no risk of needing mid-development switch
  - More capability than needed - simplification of code
  - Higher power consumption than lower spec model

Problems & Solutions

- **Sensor doesn't appear to meet specifications from the manufacturer**
  - Detects 10m away
  - Refresh rate falls drastically when polled in pairs
  - Modified LED update rules to reflect deficiency of range and amount of data
- **Problems communicating with Arduino from STM**
  - Using GPIO interrupt-based parallel communication instead
  - Faster and less unknown behavior
  - More wires - not outrageous in our use case
- **Switched to 5V power rather than 12V**
  - Realized we didn't need the 12V to power the microwave sensors
  - Same battery capacity
  - Don't have to worry about stepping the voltage down

Schedule & Tasks

20-Sep	27-Sep	4-Oct	11-Oct	18-Oct	25-Oct	1-Nov	8-Nov	15-Nov	22-Nov	29-Nov	6-Dec
Ultrasonic & LIDAR Research	Microwave Research										
LED Research	LED & Power Parts Research	Memomory Management SP Code									
	STM Research	Basic Multi-target Selection	Distance Debuggin	Distance Testing							
		Parts Ordering & Confirmation	Overall Schematic	Speed Code Basic	Speed Debugging	Combine Multi Sensor					
		Github & STM Envo Setup	Arduino Driver	Mount Design	Mount Corrections	Microwave Testing					
				Microwave Arduino Control	LED Control Code	Mount Creation	Remove Mem management				
					Arduino Control Debug	LED soldering	Power Control	Hardware Assembly			
						STM Control Switch	Code Integration	Code Hardware Integ	Testing	Refinement	Testing
Albany											
Emily											
Jason											
Albany & Emily											
Albany & Jason											
Emily & Jason											
All											

To Do

- **Testing with entire system**
  - Signals code correctness on STM32
  - Running entire battery cycle for stress testing
- **Real world testing**
  - Riding on bike and following with car
  - Final latency and delay benchmarking
  - Accuracy measurements and code optimization
  - 1 hr active power consumption
- **Video**