Mobile LiDAR Scanner

Group 2 – Final Presentation

Mateo Cuesta - Photonic Science Engineering

Dannah Dolorfino - Computer Engineering

Arturo Martin Jimenez - Photonic Science Engineering

Troy Morgan - Computer Engineering

Recap and changes to design

- For our Senior Design project, we decided to create a remote-controlled mobile LiDAR scanner that:
  - Creates a ToF image using custom optics
  - Renders a 3D point-cloud of that image
- Some changes to first design
  - Instead of a room scan, convert on ToF image to 3D point cloud
  - No user interface; had to focus on implementing hardware instead

Requirement specifications

Specification	Description	Unit	Tested
Horizontal field of view	The LiDAR system is able to scan horizontally across an area	360°	360°
Vertical field of view	The LiDAR system is able to scan vertically across an area	10° - 45°	-
Detection range for LiDAR	How far the LiDAR system is able to read distances.	1 - 50 m	-
Operation lifetime	Vehicle must have enough power for movement and LiDAR scans for this amount of time.	30 mins	54 mins
Speed of System	This would determine the operating speed thar we intended to use on our system.	< 5 mph	0.975 mph (calculated)
Bluetooth Range	The range at which the Bluetooth module can receive data to transmit to our LiDAR system	< 20 m	10 m

Detection system

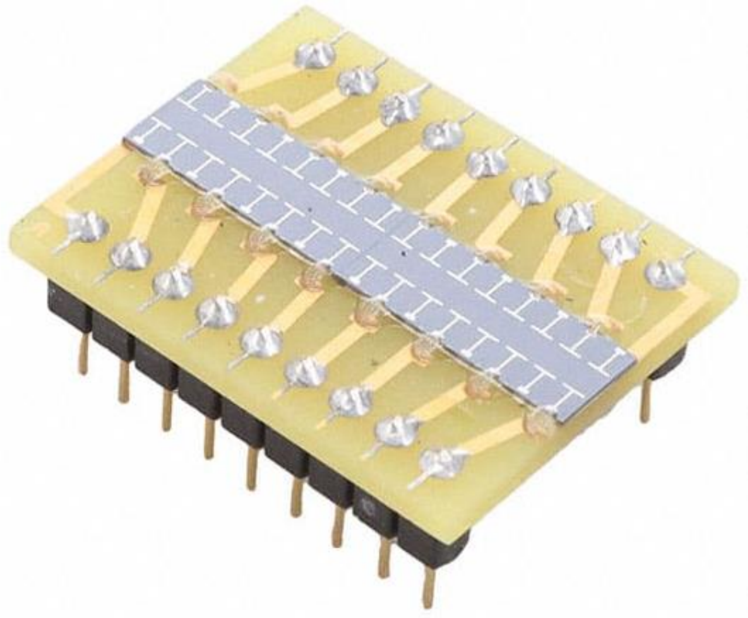
Mateo Cuesta: PSE

Detection System: CCD sensor

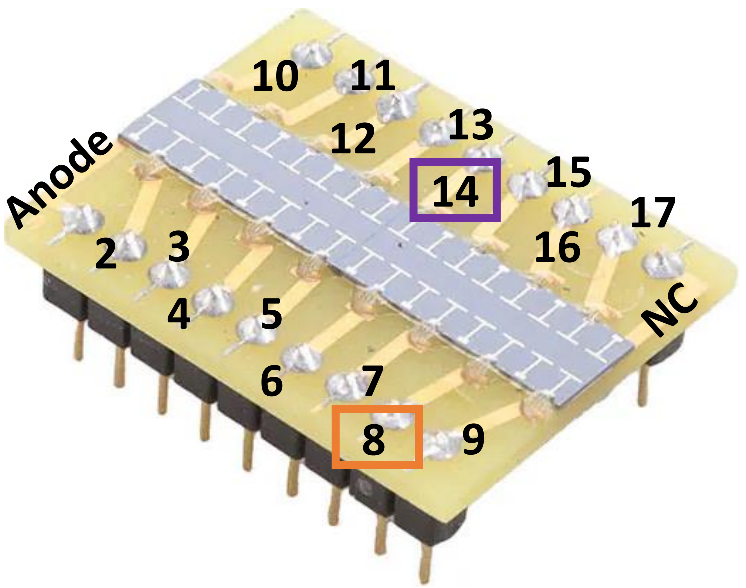
- The selected CCD sensor is the EPC635 3D TOF Imager
  - 3.2 mm × 1.2 mm
  - 160 × 60-pixel active region
  - 20 × 20 µm pixel size
  - 70% QE at 940 nm wavelength
  - This board is equipped with a TOF distance calculation
    - Uses internal and external modulators to modulate the light impinging on target.
      - Internal Modulator: reference
      - External Modulator: Modulates laser diode
    - Distance calculation is based on TOF modulation theory, uses phase offset to calculate the distance.

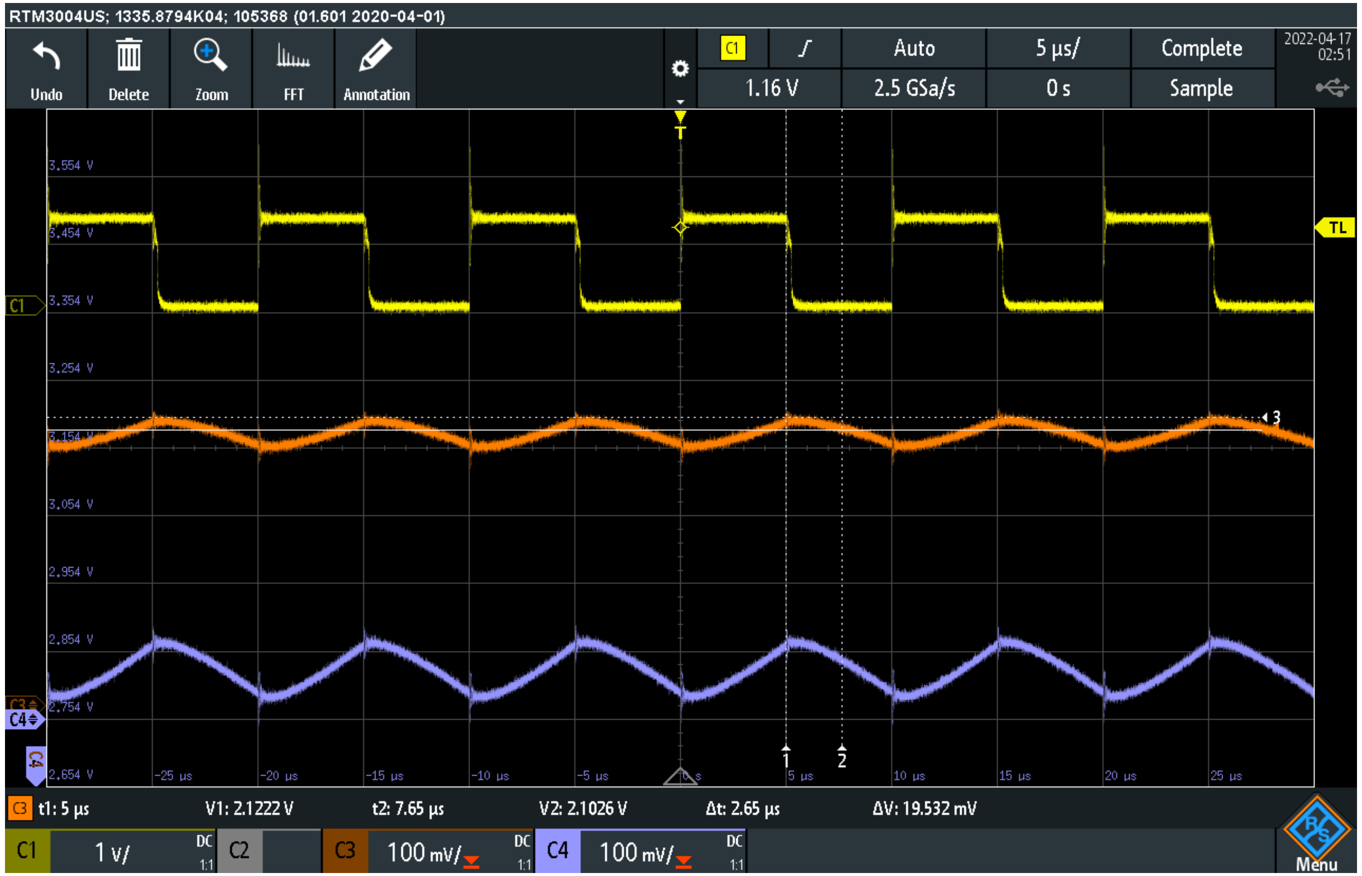
Detection System: Photodiode Array

- Integrated sensor into the Mobile LiDAR Scanner
  - 1.92 mm^2 active region
  - 0.60 A/W responsivity at operating wavelength
  - Can be used to calculate distance by measuring the time difference between detected pulse with respect to a reference pulse.



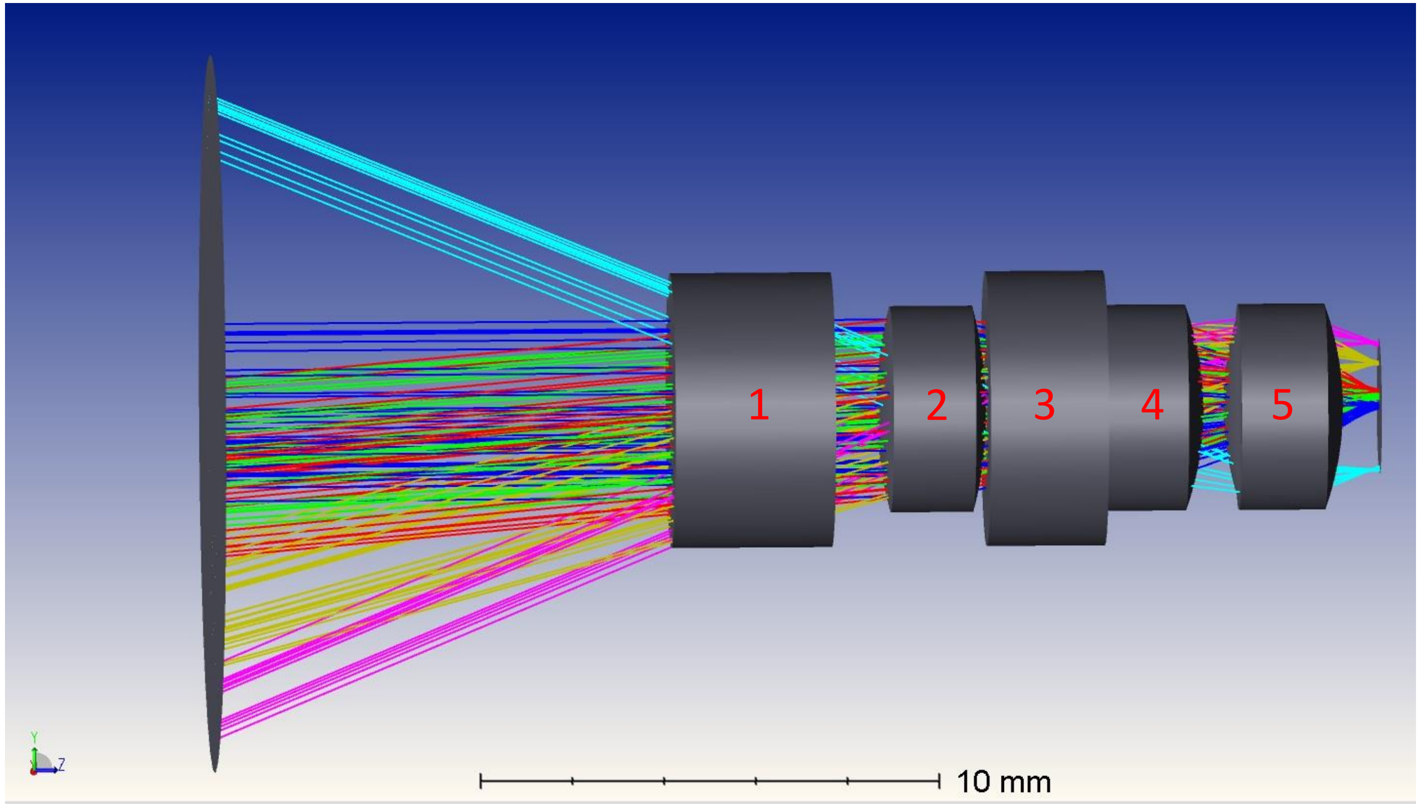
Detection System: Photodiode Array





Detection System: Optical Design

- Goal of detection optics:
  - Collect light from FOV of 0° to 45°.
  - Maximize SNR
  - Focus light onto CCD sensor active region
- The detection system is composed of:
  - Optical BPF centered at 940 nm (1)
  - Three convex lenses (2,4,5)
  - One concave lens (3)
- Approach:
  - 940 nm Optical BPF to eliminate background light
  - Focusing optical system to create smallest spot size



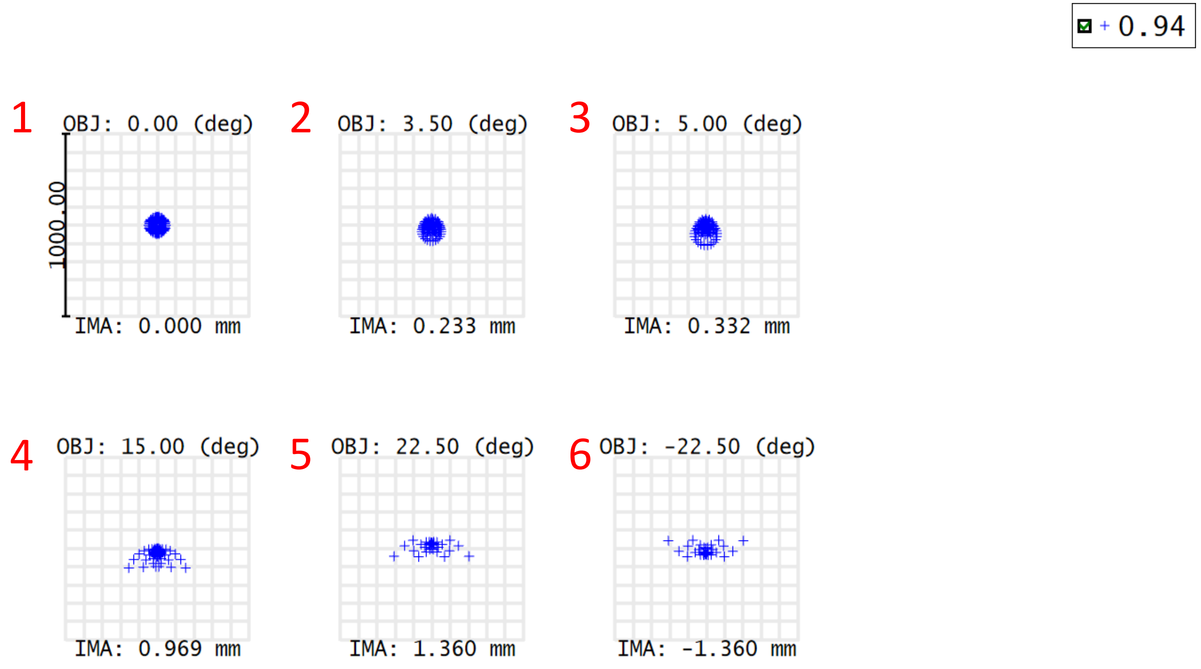
Detection System: Optical Design

- Image height radius is 1.48 mm
    - Focused light is fully incident on CCD sensor and the photodiode array

	Surface Type		Comment	Radius	Thickness	Material	Coating	Clear Semi-Dia	
0	OBJECT	Standard		Infinity	Infinity			Infinity	
1		Standard		Infinity	10.000			7.844	
2	(aper)	Standard		Infinity	3.500	N-BK7		3.000	U
3		Standard		Infinity	1.000			2.784	
4	STOP (aper)	Standard	67595	8.880	2.380	N-BK7	EO_NIRII_517	2.025	U
5	(aper)	Standard		-8.880	0.000		EO_NIRII_517	2.025	U
6	(aper)	Standard	67982	Infinity	1.500	N-SF11	EO_NIRII_785	2.700	U
7	(aper)	Standard		4.710	0.650		EO_NIRII_785	2.700	U
8	(aper)	Standard	67594	6.580	2.600	N-LASF...		2.025	U
9	(aper)	Standard		-6.580	0.439			2.025	U
12	(aper)	Standard	67594	6.580	2.600	N-LASF...		2.025	U
13	(aper)	Standard	0.8 is dist...	-6.580	0.800			2.025	U
14	IMAGE	Standard		Infinity	-			1.480	

Detection System: Optical Design

- Spot size incident on CCD sensor from incidence angle (angles are half angle)
    - Field 1 ⟶ 0°
    - Field 2 ⟶ 3.5°
    - Field 3 ⟶ 5°
    - Field 4 ⟶ 15°
    - Field 5 ⟶ 22.5°
    - Field 6 ⟶ -22.5°
- Our current design yields a 4-pixel blur.
- RMS spot size radius in µm.



Surface: IMA

Spot Diagram

67594, 1/28/2022
Units are µm. Legend items refer to Wavelengths

Field :	1	2	3	4	5	6
RMS radius :	38.937	43.584	48.610	62.474	75.516	75.516
GEO radius :	47.637	86.227	110.311	188.245	209.709	209.709

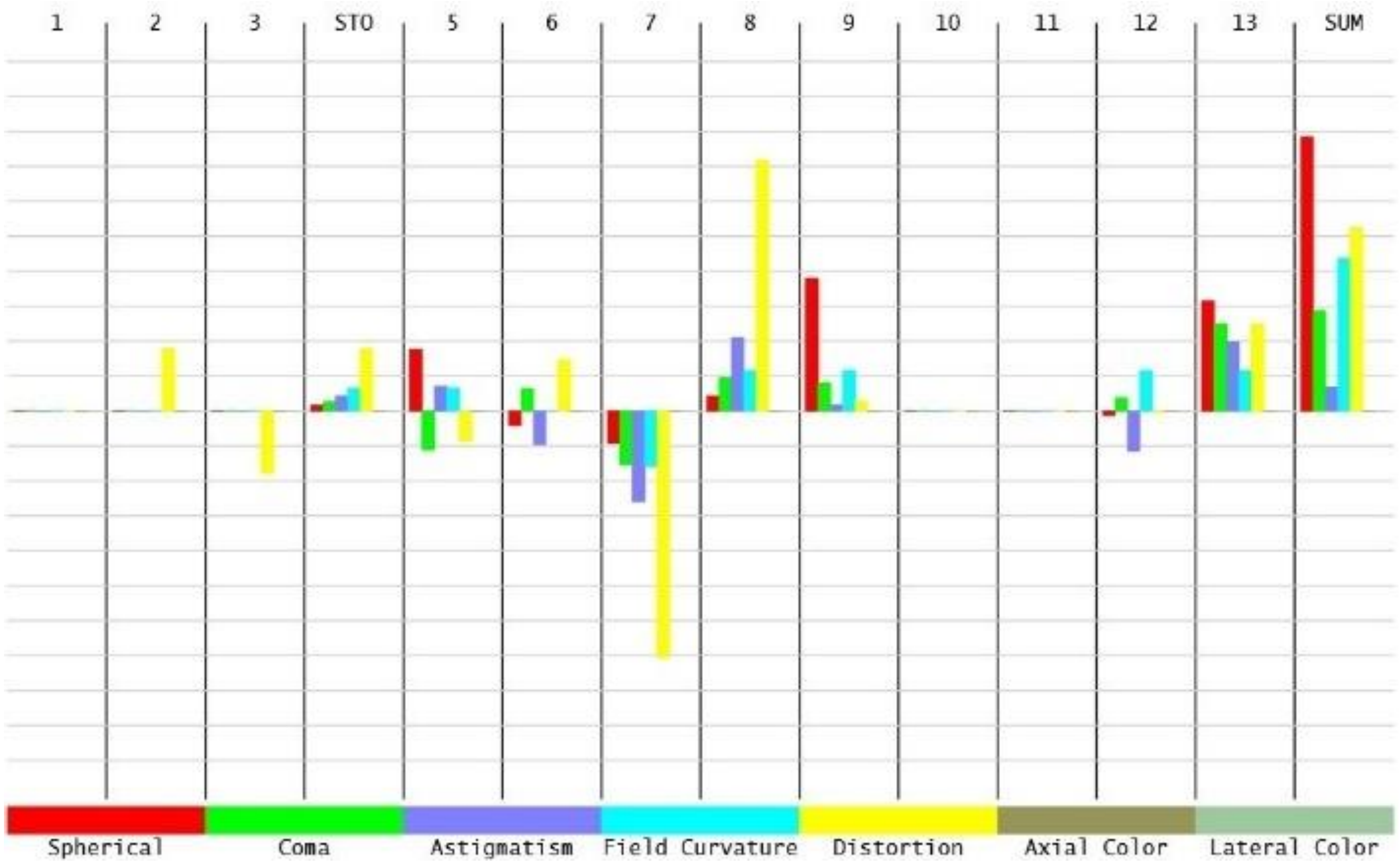
Scale bar : 1000 Reference : Chief Ray

Zemax
Zemax OpticStudio 21.3.2

3lens_4.5d_vex9_caveneg6_vex4.5_V1.zos
Configuration 1 of 1

Detection System: Optical Aberrations

- Detection system designed to focus a single reflection point in object space to an individual pixel on the sensor
- Large FOV resulted in a large field curvature
    - Intentional defocus from paraxial limit from on axis rays to compensate for field curvature
    - Resulted in higher spherical aberrations
- Distortion can be neglected in our system because we are not capturing an image, but rather distances to create a point cloud data set.



Detection System: Filter

- BPF needs to be centered at 940 nm
- Optical filter shop filter has a larger FWHM
    - More background light will enter the sensor
- Diameter comparable to collection optics size
- Less expensive

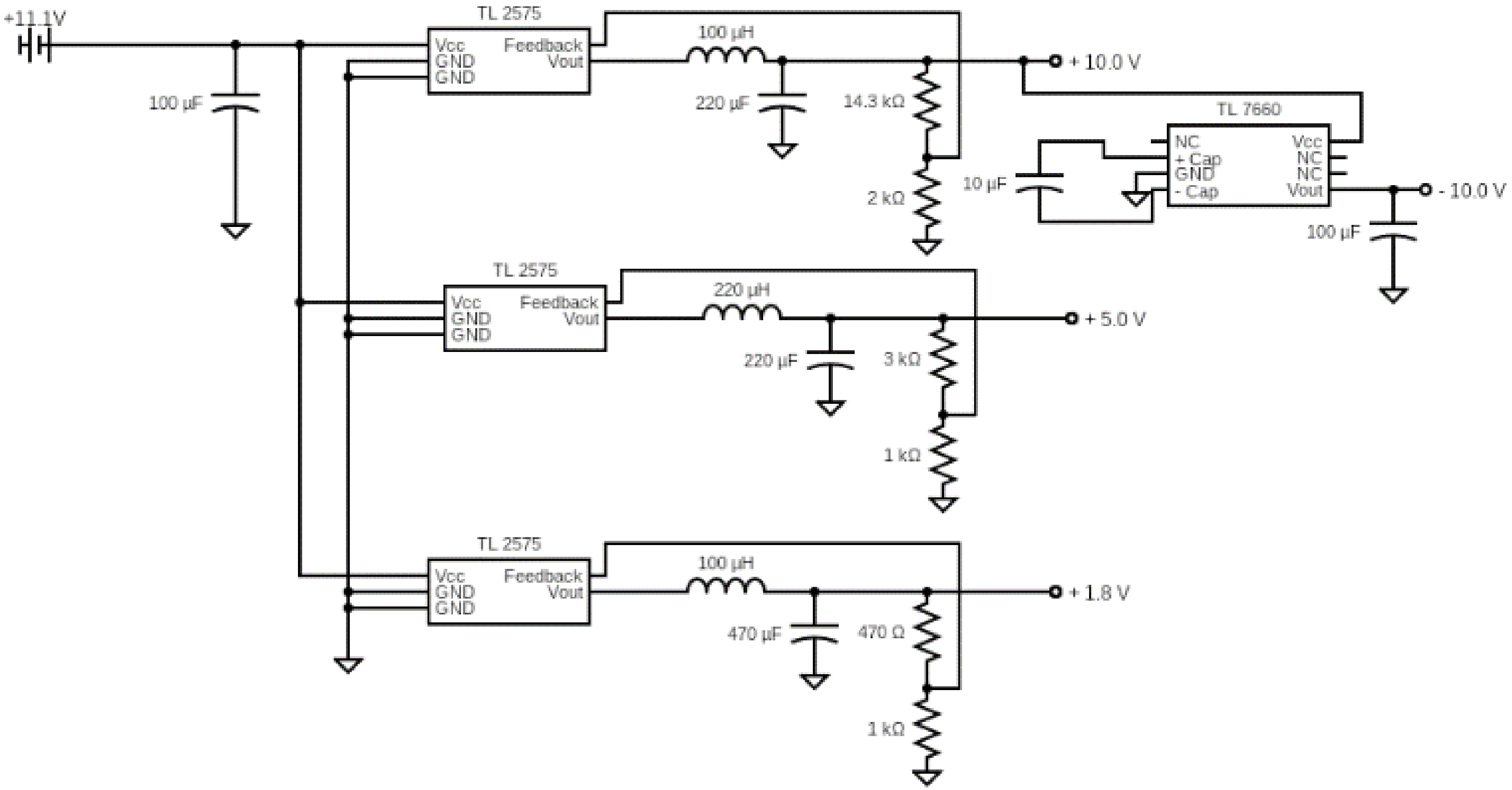
Filter	Optical Filter Shop	Edmund Optics
Wavelength	940 nm	940 nm
FWHM	± 20 nm	± 10 nm
diameter	6 mm	12.5 mm
Price	$73.00	$168.00

Power Board for CCD Sensor

- Requires 5 distinct voltages to be applied.
- Distribute the voltages from a single 11 V 800 mAh battery.
- Voltages must be applied in specific order from top to bottom from the table.
- V_{DDA} will bias the photodiode circuit.

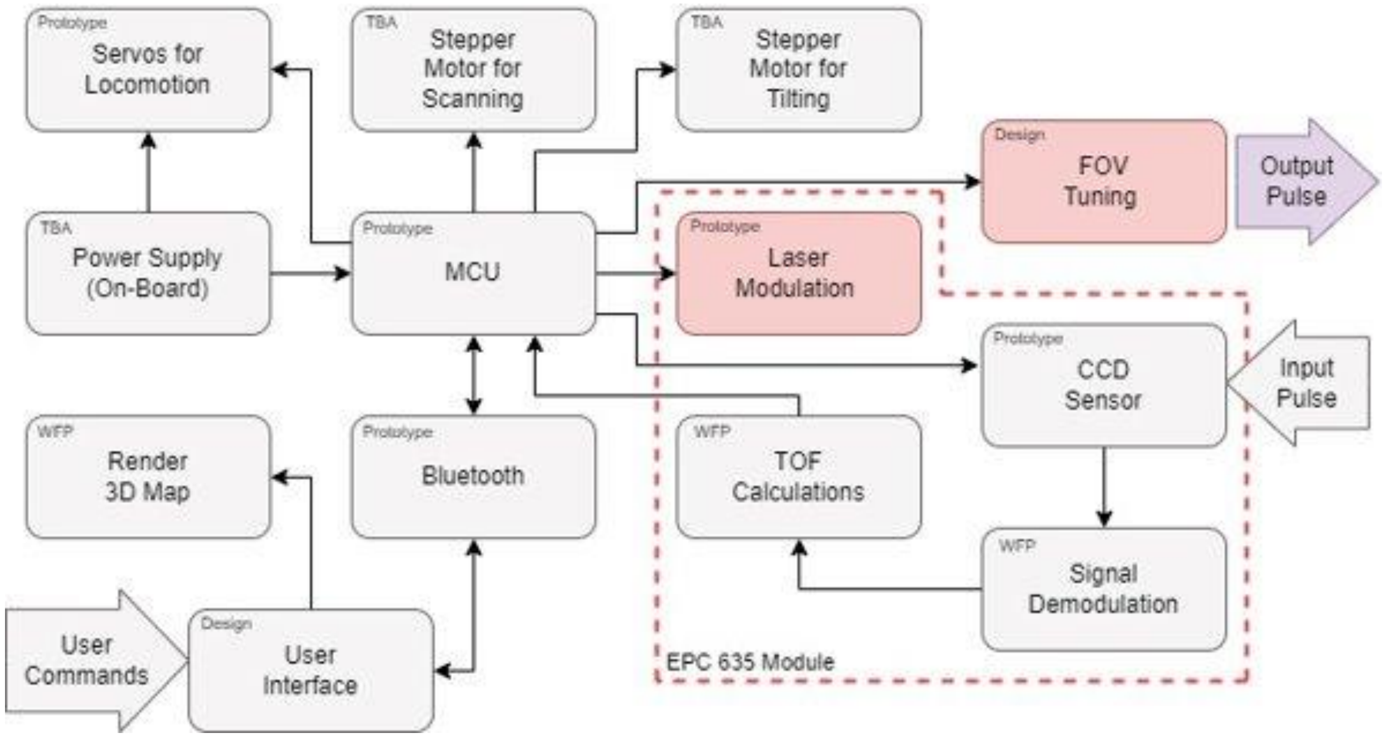
Pin	Voltage	Ripple
V_{DDIO}	+ 2.5 / + 3.3 V	± 50 mV
V_{DD}/V_{DDPLL}	+ 1.8 V	± 20 mV
V_{DDA}	+ 5.0 V	± 20 mV
V_{DDPXH}	+ 10.0 V	± 20 mV
V_{BS}	- 10.0 V	± 50 mV

Circuit Schematic for Detector Power



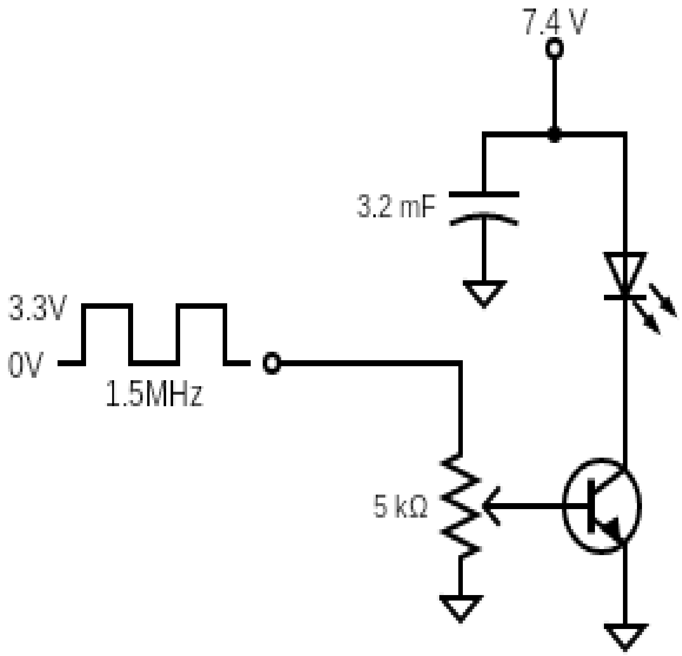
Illumination system

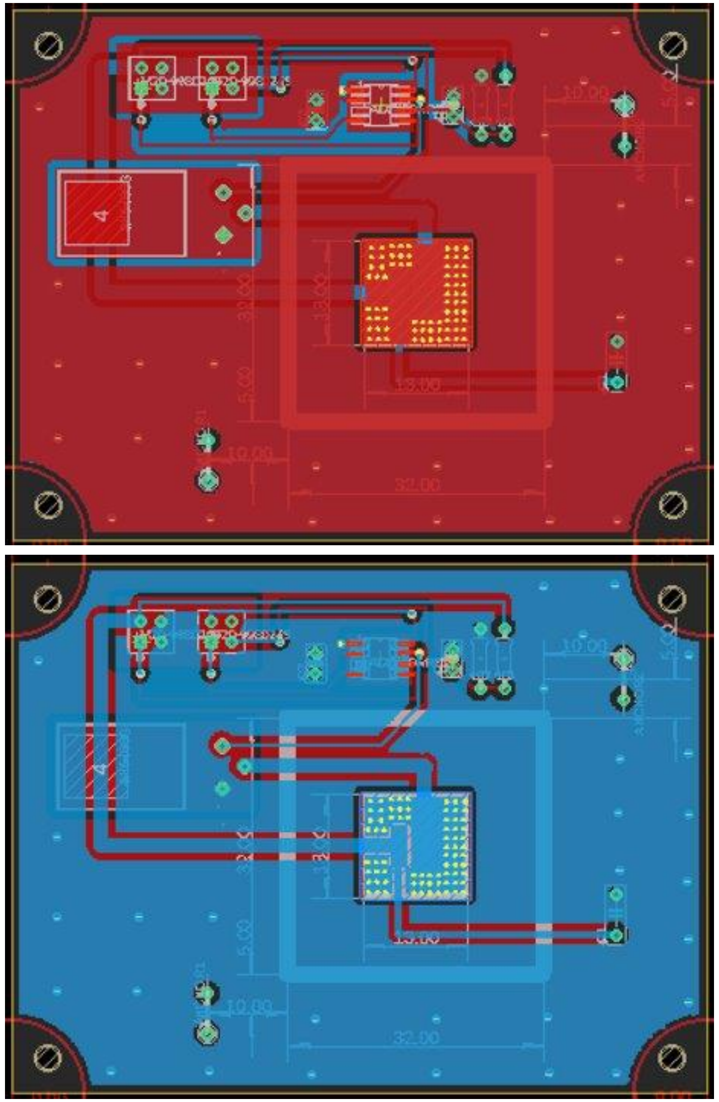
Arturo Martin Jimenez: PSE



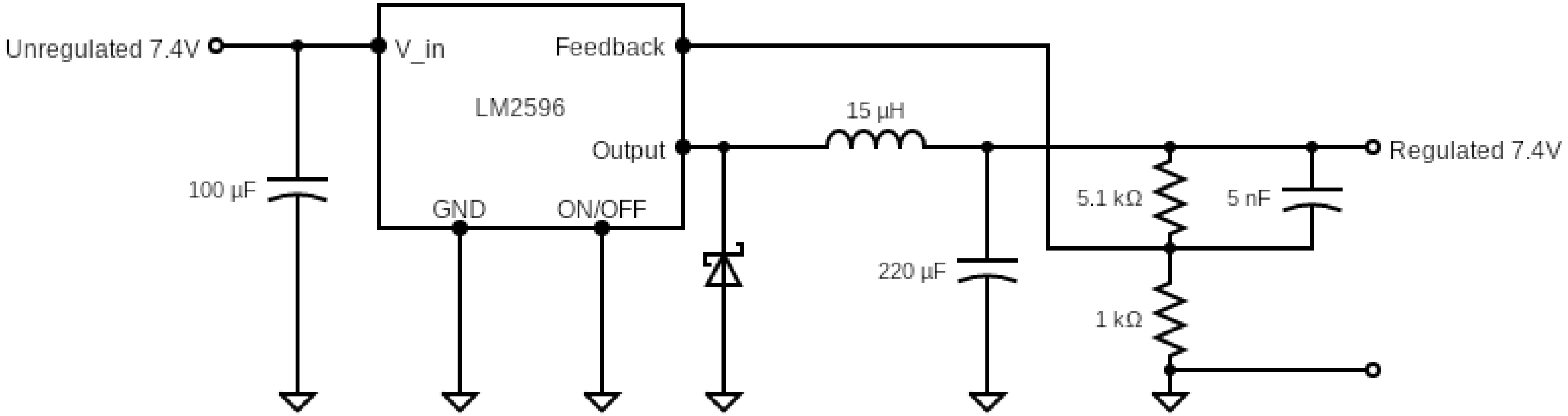
VCSEL Driver Design

- Minimum required modulation frequency: 1.5 MHz.
- Using high power NPN transistor.
- Potentiometer controls amplitude of switching voltage to decrease emission intensity.
- Temperature control using heat sink and cathode plane on PCB to dissipate heat.

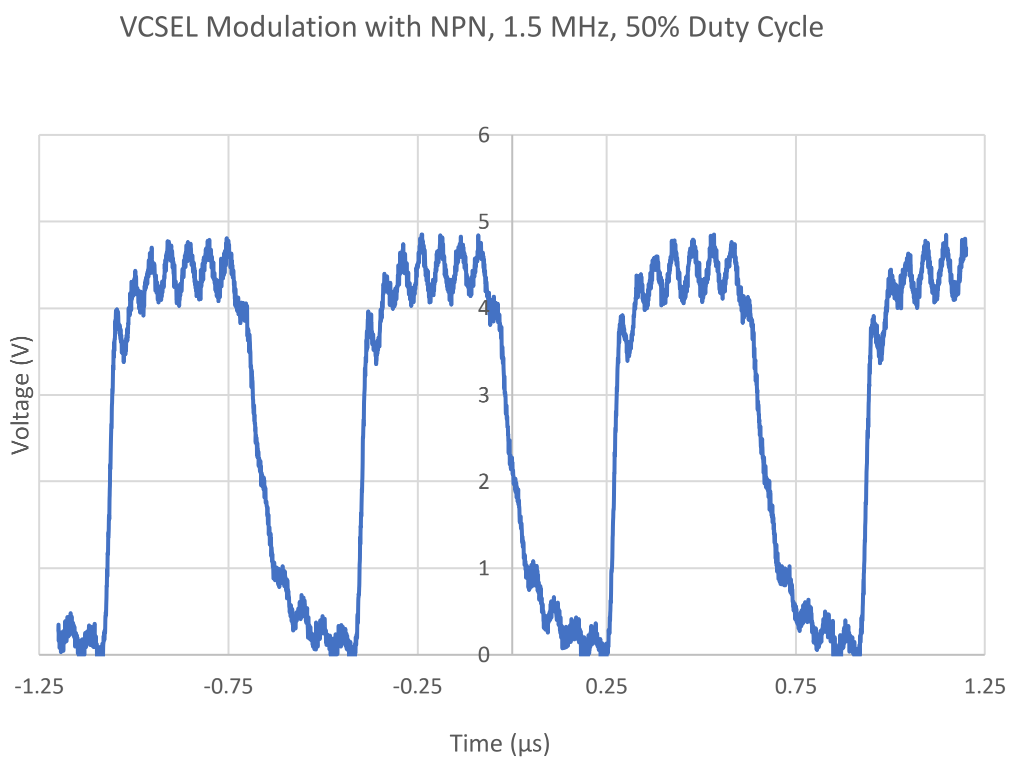


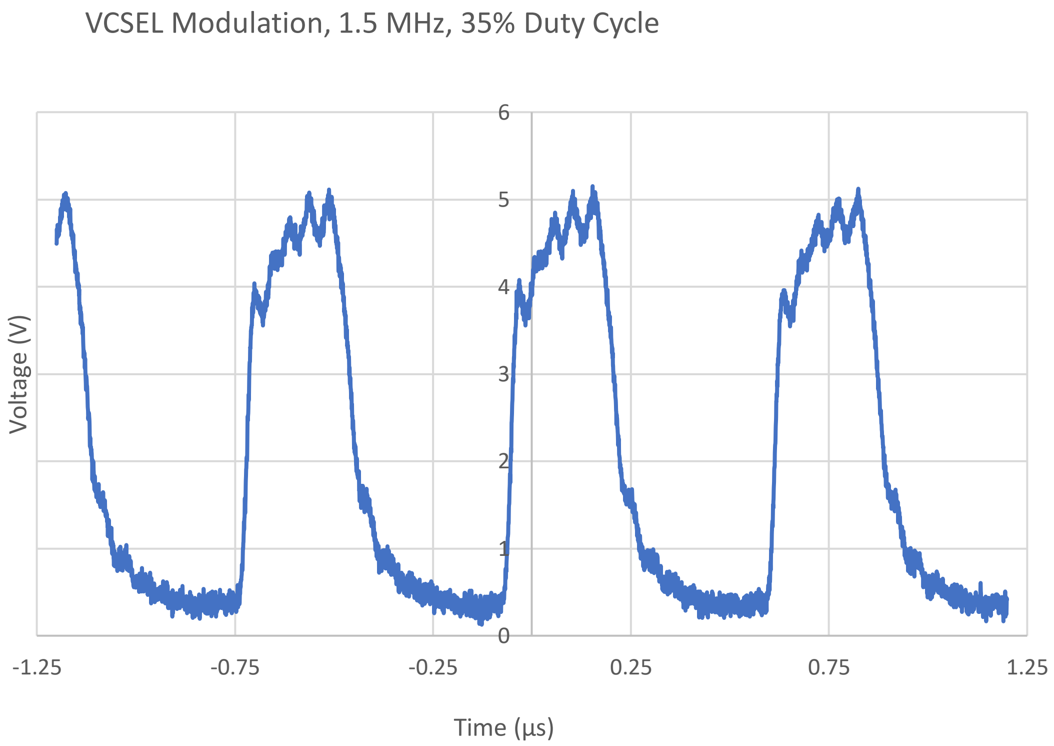


Supply Voltage Regulator for VCSEL Driver

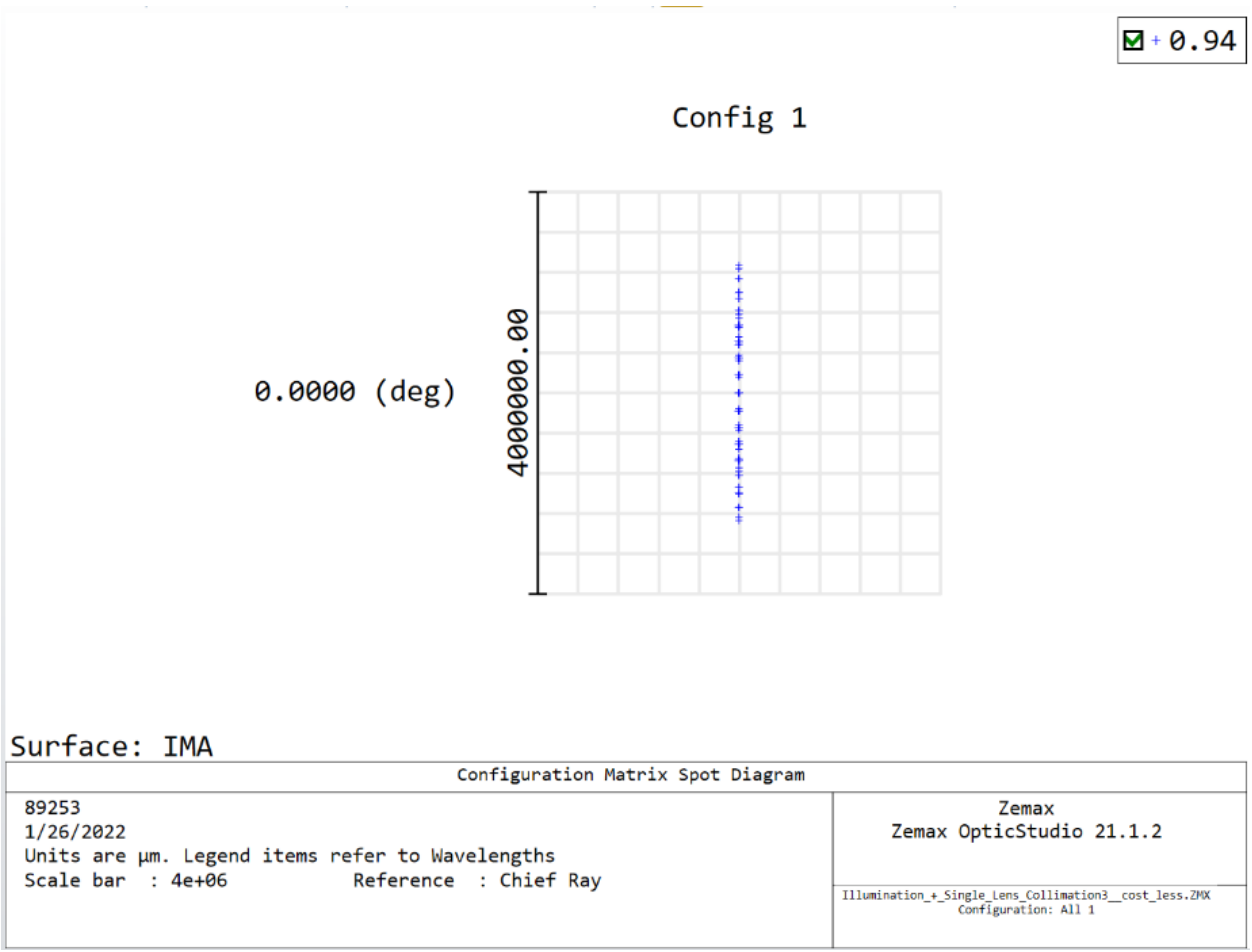


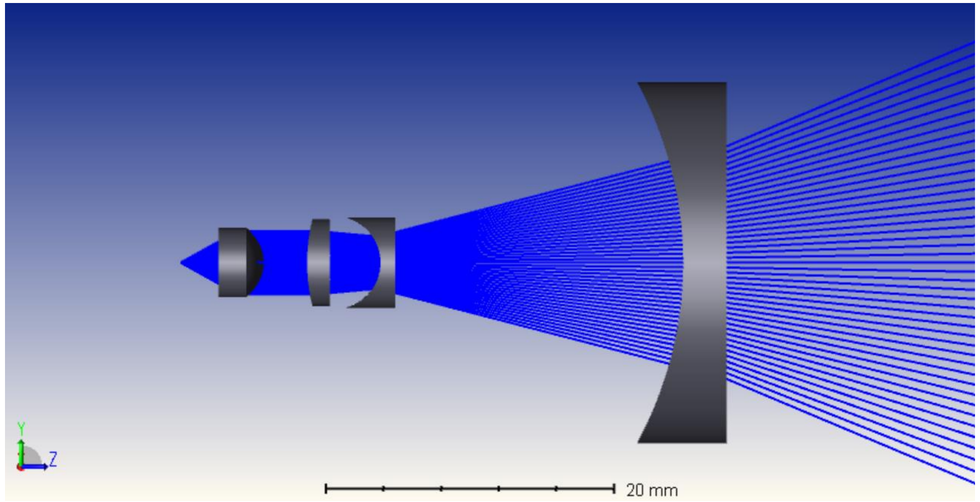
VCSEL Modulation

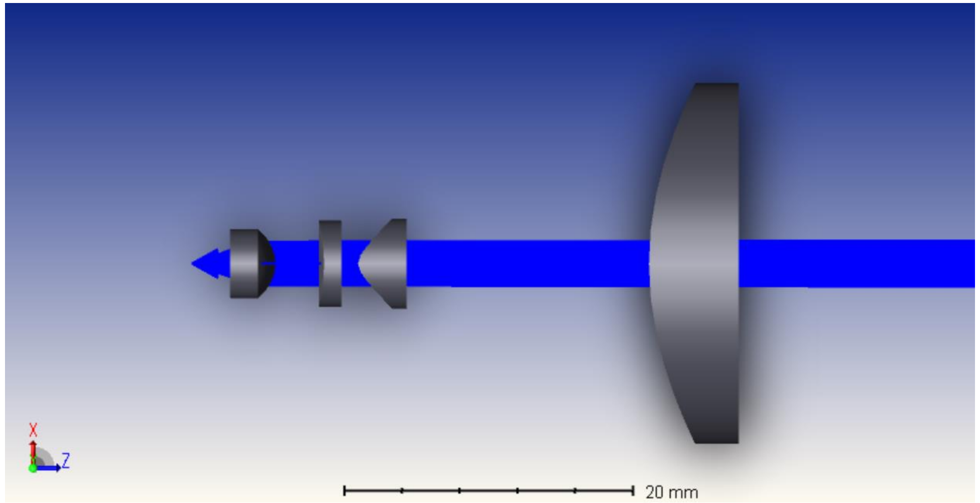




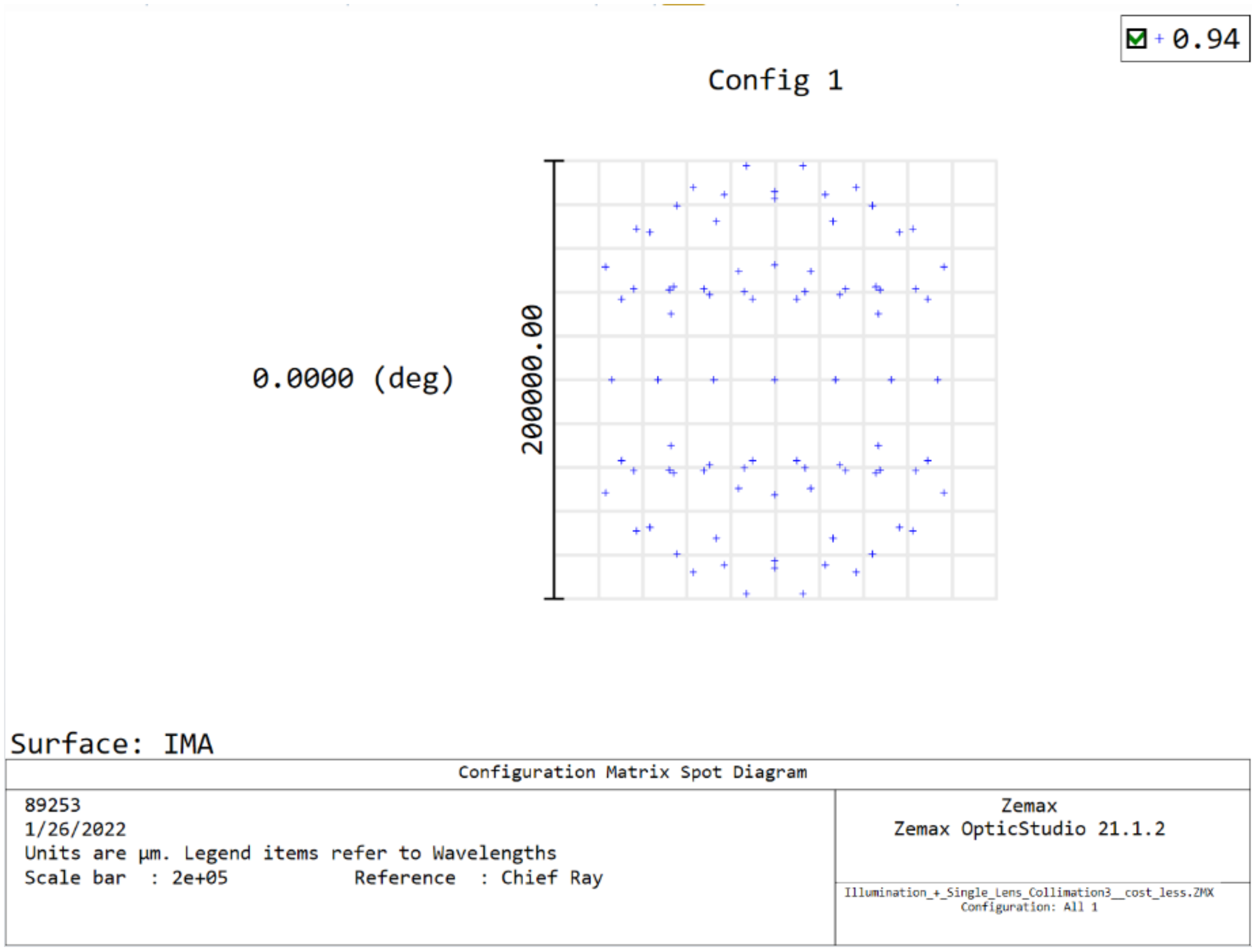
Illumination: Optical Design (Max FOV)

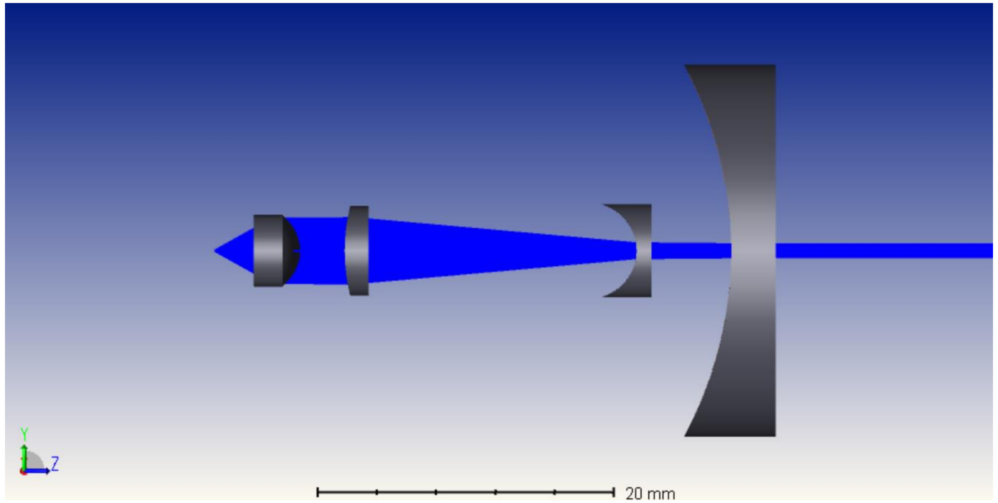


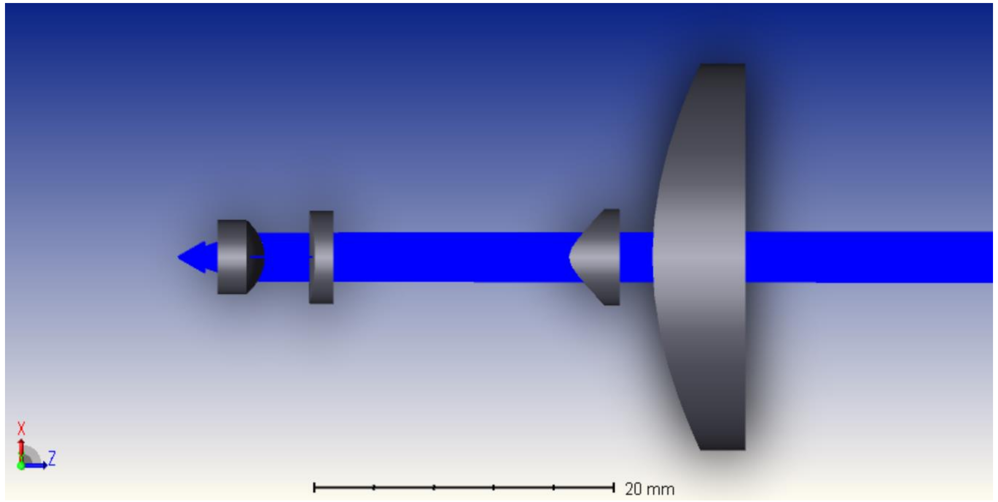




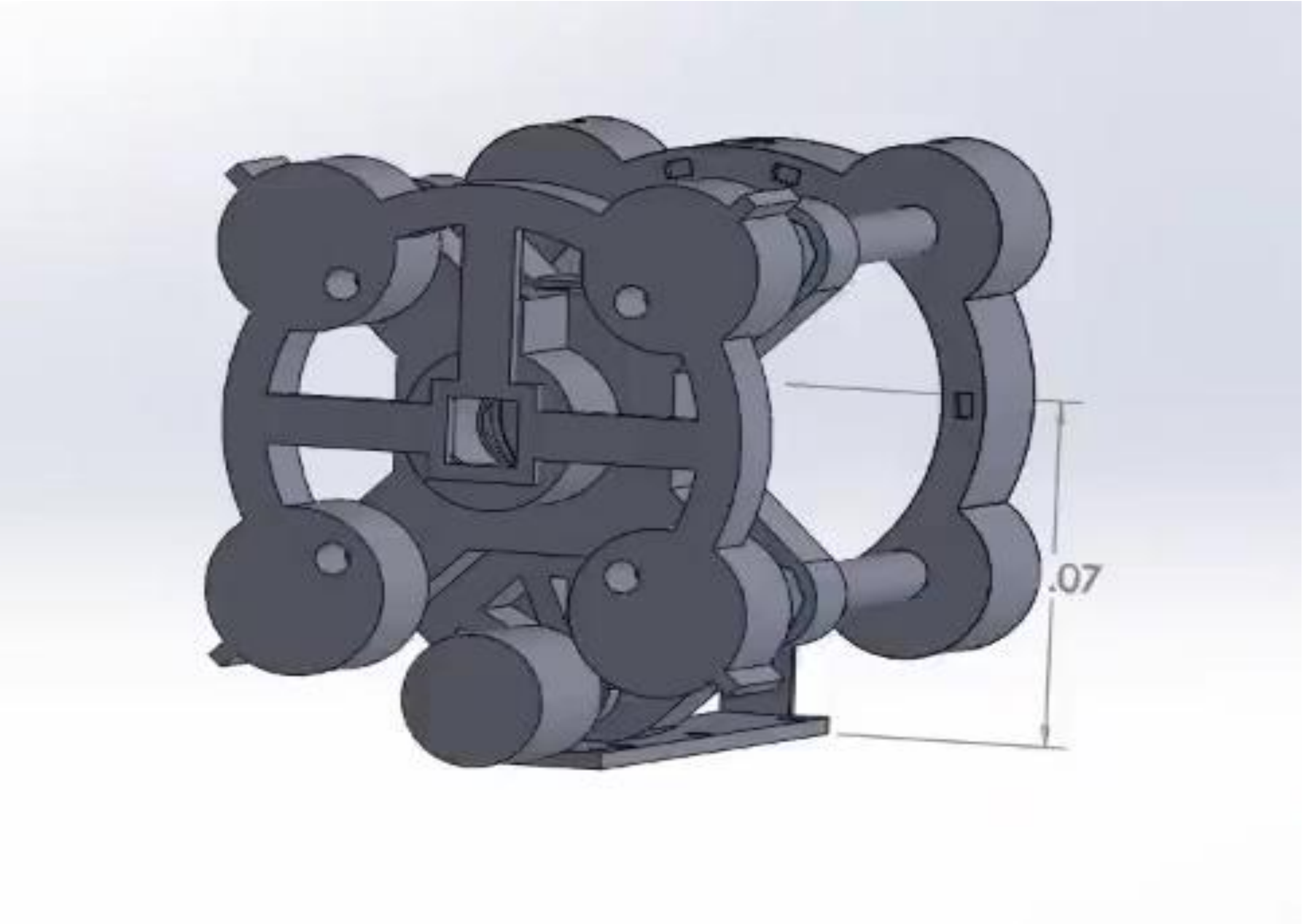
Illumination: Optical Design (Min FOV)



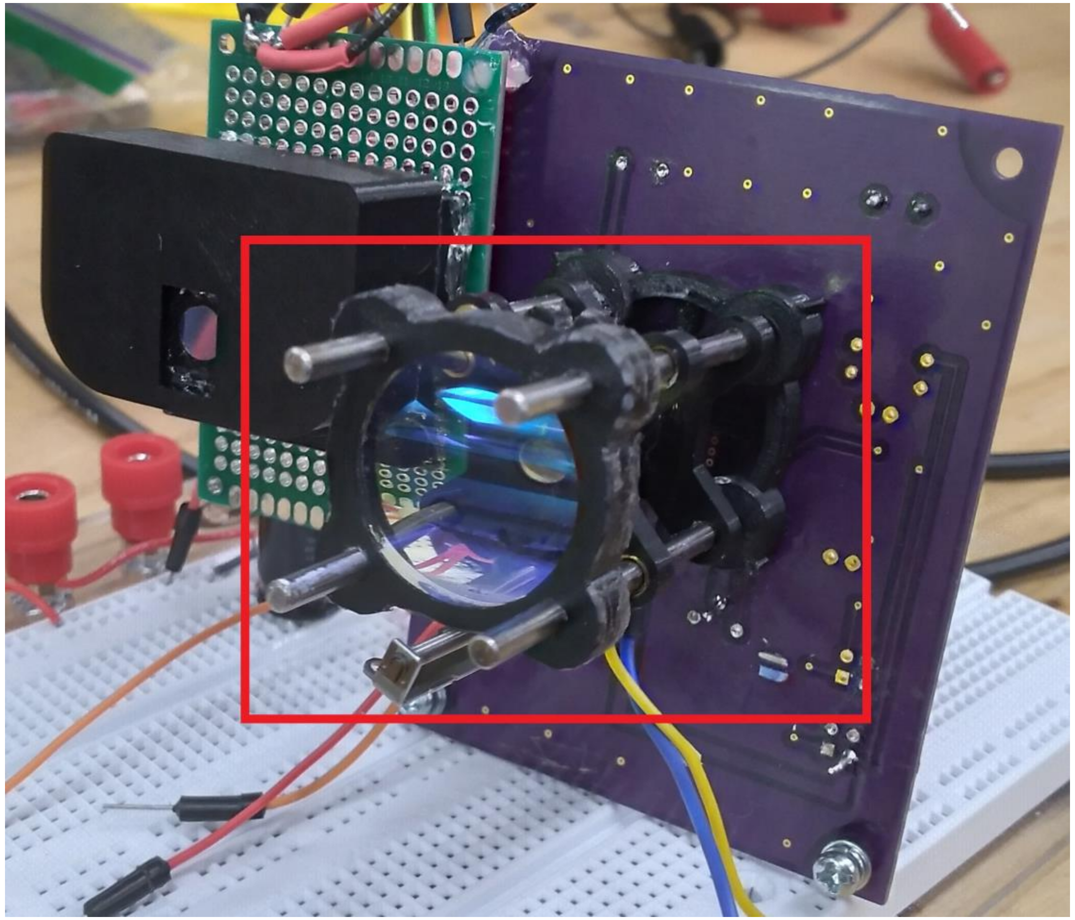




SolidWorks Design for Lens Housing



FOV Tuning Optics

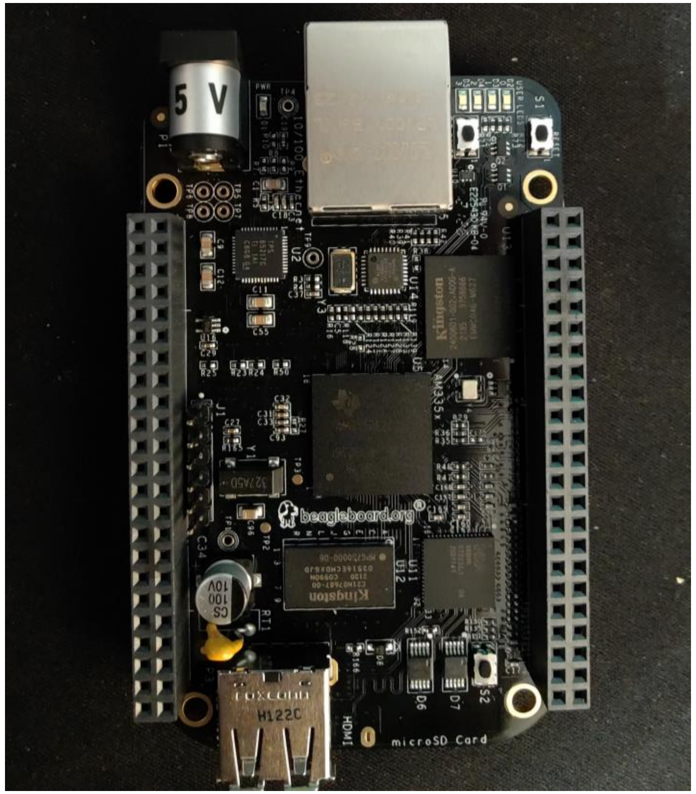


Data transmission, data processing, and motors

Dannah Dolorfino: CpE

BeagleBone Black (BBB)

- The BeagleBone Black is a community supported and low-cost system on chip, developmental platform
  - uses a AM335x 1 GHz ARM Cortex-A8 processor with 512 MB of DDR3 RAM, 4GB 8-bit eMMC flash storage and 2 PRU 32-bit microcontrollers
- BBB runs on Linux and has the latest Debian image flashed on to it.
  - Peripheral associated: USB, ethernet, 92 pins connectors that are compatible with different Communication Protocols (UART, I2C, SPI)



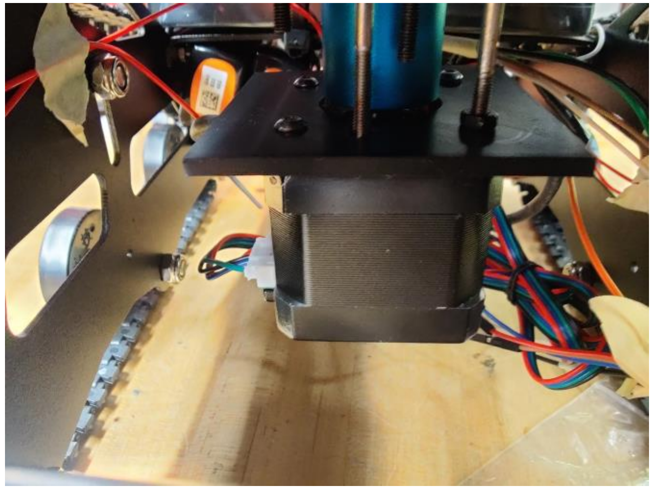
BeagleBone Black

Bluetooth communication

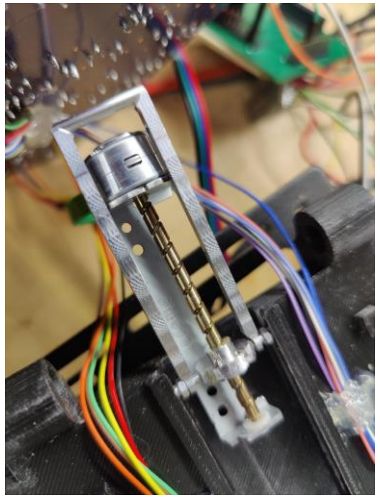
- Used to connect host computer to the Mobile LiDAR Scanner
  - During movement mode: controls locomotion motion and rotation
  - During configuration mode: controls tilting and FoV tuning
  - During scanning mode: start/stop ToF scans; DCS frame pixel data would also be sent over to the computer for calculation.
- Used HC-05 Bluetooth module
  - Can act as master or slave
  - Connected to UART channel 4 on BBB
  - To computer, sees module as Bluetooth device
  - To BBB, sees it as a UART Device

Interfacing stepper motors

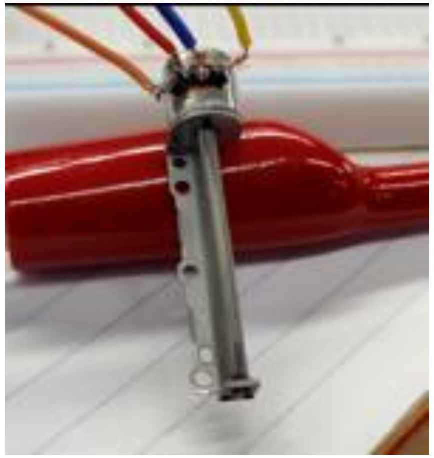
- Rotation motor to rotate the LiDAR system allowing for horizontal FoV of 360 degrees
- Tilting motor (allow us to move the camera), FoV tuning motors
  - Salvaged from old computer (no datasheets)
- Had to write custom software motor drivers due to no ready libraries
- Used A4988 stepper motor driver used to interface each motor in final design



Rotation motor attached



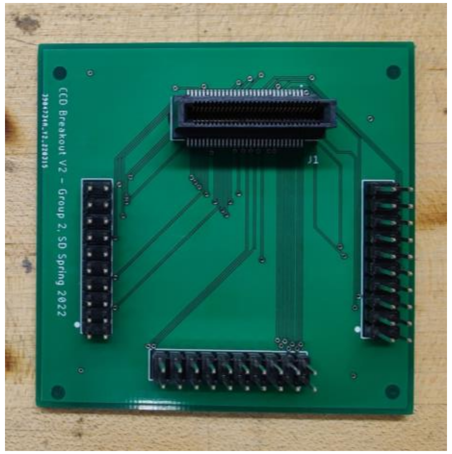
Tilting motor



FoV turning motor

Getting CCD sensor data with current implementation

1. Turn on epc635 using power circuit.
2. Check using 'i2cdetect' to confirm that the CCD sensor was on and detected by BBB.
3. Run the sequencer code provided by datasheet.
4. Set configurations writing to I2C registers.
5. Use the HW shutter to start image acquisition.
6. Read DATA[7:0] at rising edge of DCLK (2.5 MHz).
7. Print to output.txt file.



CCD sensor breakout board

Setting configurations using I2C interface

Address	Register	Value	Description
0x20	0xCC	0x01	All CLKs active low
.	0xCB	0x1D	8-bit, labeling, gated CLK
.	0x89	0x1F	DCLK = 2.5MHz
.	0x80	0x3F	Enable internal clock
.	0x91	0x43	Read out rollover for slow DCLK
.	0x7D	0x04	Turns on all clocks
0x20	0xA4	0x01	SW shutter signal (optional)

```
root@beaglebone:/var/lib/cloud9/Project# i2cdump -y 2 0x20
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: 00 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00    ..?.............
10: 00 02 13 00 00 00 00 32 44 13 0a 00 1e e1 aa 55    .??....2D??.???U
20: 00 0f 30 00 00 35 00 00 3a 00 00 3f 00 00 50 00    .?0..5..:..?..P.
30: 10 60 00 10 90 00 00 a0 01 00 10 00 26 00 00 00    ?`.??..??.?.&...
40: 9c bd 0e b0 06 50 00 00 00 01 00 01 00 01 00 28    ?????P...?.?.?.(
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 03 1f 00    .............??.
60: 35 af 00 00 00 00 00 00 0f ff 00 00 02 00 00 00    5?......?...?...
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 04 03 00    .............??.
80: 3f 00 09 00 03 01 04 00 1f 1f 01 01 08 05 7f ff    ?.?.???.???????.
90: c4 43 34 00 00 03 00 04 00 a3 06 41 00 50 07 ff    ?C4..?.?.??A.P?.
a0: 00 01 07 ff 00 07 03 1f 09 5f 00 04 e1 ff 01 0d    .??..????_.??.??
b0: 07 ff 03 1f 09 5f 00 18 00 80 00 0a 00 00 00 00    ?.???_.?.?.?....
c0: 01 8f 01 8f 01 8f 01 8f 01 8f 20 1d 00 13 45 00    ?????????? ?.?E.
d0: 26 0b 0c 12 42 36 1f 00 00 00 00 00 00 00 37 00    &???B6?.......7.
e0: b5 46 00 00 00 00 00 00 79 6f 00 00 00 00 00 00    ?F......yo......
f0: 8d fe 00 c7 07 00 00 00 00 00 00 00 00 00 00 00    ??.??...........
root@beaglebone:/var/lib/cloud9/Project#
```

Interfacing the CCD sensor

Problems

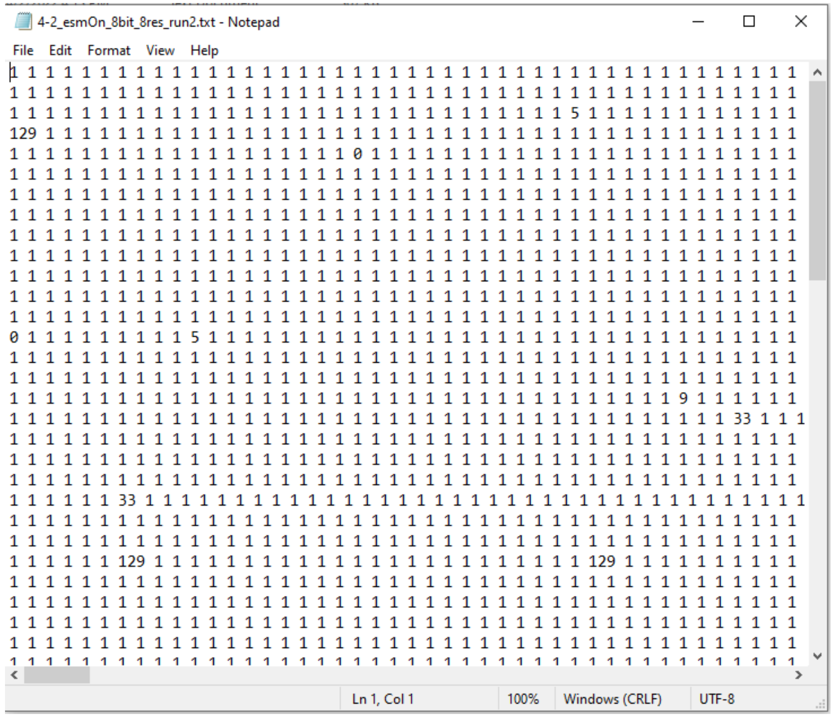
- Powering the CCD sensor
- Main issue was getting the BBB to capture the DCS frame data back from the CCD sensor
  - BBB could not properly read the data being sent back
  - Due to BBB not capturing the data fast enough
- Considered the following solutions and advice:
  - Reduce amount of data coming back (1 ToF measurement)
  - DCLK was set to lowest speed (2.5 MHz)
  - Polling the DCLK pin using while loops
  - Use GPIO library with direct pin references
  - Use the on-board PRUs to collect data and store into BBB RAM
  - Get data using MSP430FR6989 then sending data back to BBB via SPI

Expected data for one ToF image:

- Each intensity on each pixel is an 8-bit number (0-255)
- 4 DCS frames (160 X 160 pixels images)
  - $9{,}600\ bytes\ \times 4 = 38{,}400\ bytes$
- 4-byte labels for frame starts/ends and line starts/ends
  - 32 frame starts/ends
  - ~480 lines starts/ends
- Total: 38,912 bytes

Data captured back from the CCD Sensor

Initial received data



```
4-2_esmOn_8bit_8res_run2.txt - Notepad
File Edit Format View Help
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1 1
129 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 9 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 33 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 33 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 129 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 129 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

Latest data coming from CCD sensor

(1) 1 146 134 176 40 33 10 99 205 58 231 130 182 242 149 210 105 54 35 185 21 99 197 139 11 21 217 89 19 141 23 133 12 113 178 85 48 23 151 163 100 17 148 75 206 45 146 199 33 67 177 90 21 147 147 229 178 213 168 100 179 144 25 18 4 84 151 197 195 1 133 (uncovered, lab light on)

(2) 249 3 29 229 229 82 20 212 17 211 99 86 128 98 200 221 75 109 25 65 150 140 37 32 144 32 79 81 11 98 89 205 41 59 161 117 159 63 34 106 195 0 196 81 52 167 143 208 54 182 11 68 68 212 42 160 83 143 229 112 162 38 139 181 193 133 161 216 (covered, lab light off)

(3) 121 173 77 19 216 71 38 6 153 123 226 81 225 121 151 213 200 21 37 156 37 111 200 151 21 57 33 141 21 179 168 141 18 5 70 195 34 210 134 149 21 50 75 211 1 132 150 64 245 145 201 4 244 86 29 3 108 209 50 13 11 78 74 1 113 105 81 184 194 146 195 (uncovered, lab light off)

(4) 17 36 229 17 219 179 81 219 113 188 177 202 155 143 133 98 84 205 201 180 11 101 20 19 148 206 201 229 50 25 25 222 166 9 53 20 41 144 112 205 220 211 13 50 33 129 67 5 178 156 163 1 93 17 51 99 14 163 159 97 141 0 57 131 148 97 211 (uncovered, lab light)

(5) 3 9 100 150 203 67 224 41 104 143 130 115 39 31 227 93 36 6 91 97 137 41 35 199 93 145 169 203 17 34 103 166 29 138 105 2 173 115 145 34 133 15 4 56 36 129 40 76 40 23 144 81 99 149 2 40 154 201 52 48 19 24 3 3 9 8 199 68 5 (uncovered, lab light on)

Data captured back from CCD Sensor

Initial received data

- Continuously reads data from the sensor.
- Reads blanking values (0x01), meaning the sensor is idling
- Shutter signal was not even sent

Latest data coming out of the CCD sensor

- Actual valid data coming out (no blanking)
- About 70 bytes of data
  - Not the 38,192 bytes we are looking for
- No 4-byte data labels

Locomotion and image rendering

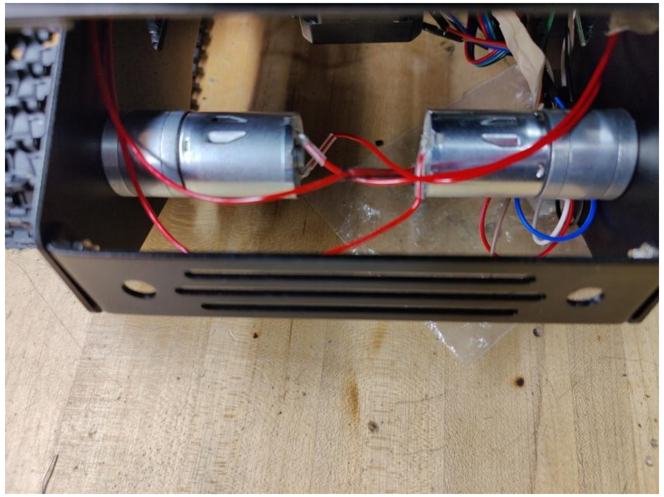
Troy Morgan: CpE

Locomotion/Transportation

- RC Platform comes with enough room to fit our components
  - LiDAR scanner
  - Multiple MCUs
  - PCBs for the wiring
- Offers a higher and lower platform for flexibility of placements for our system
- Chose a sprocket wheel design for maneuverability on different terrains

Locomotion: Maneuverability

- Motors present in the RC Car are GM25-370 with a rotational speed of 3700 rpm.
    - The motors power the sprocket, depending on the speed of a particular motor will determine the direction it's movement will be
- Turning radius
    - 0 degrees for immediate turning
- How we plan to control it
    - Controlled via UI design, using Bluetooth module
- Using L298N for operating the motors
    - high current motor drive chip
    - Takes in voltage ranging from 5V - 35V



Locomotion: Communication

- Movement controlled by the Bluetooth module with communication protocol UART
  - Interfaced with *pyserial*
  - Sends commands from a host computer to BBB
    - Can be controlled from a separate host computer
- Travels at a speed of less than 5 mph
  - Travels in all directions

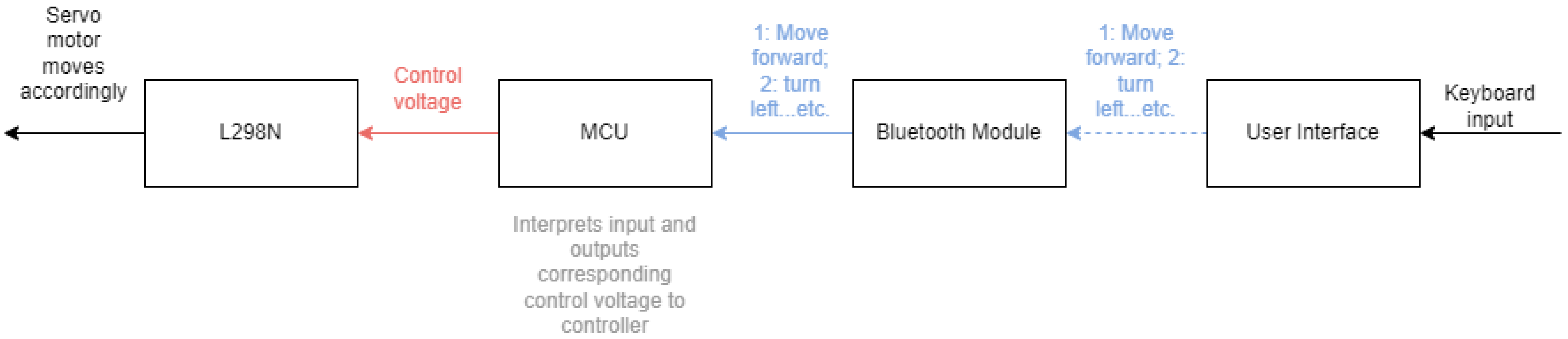
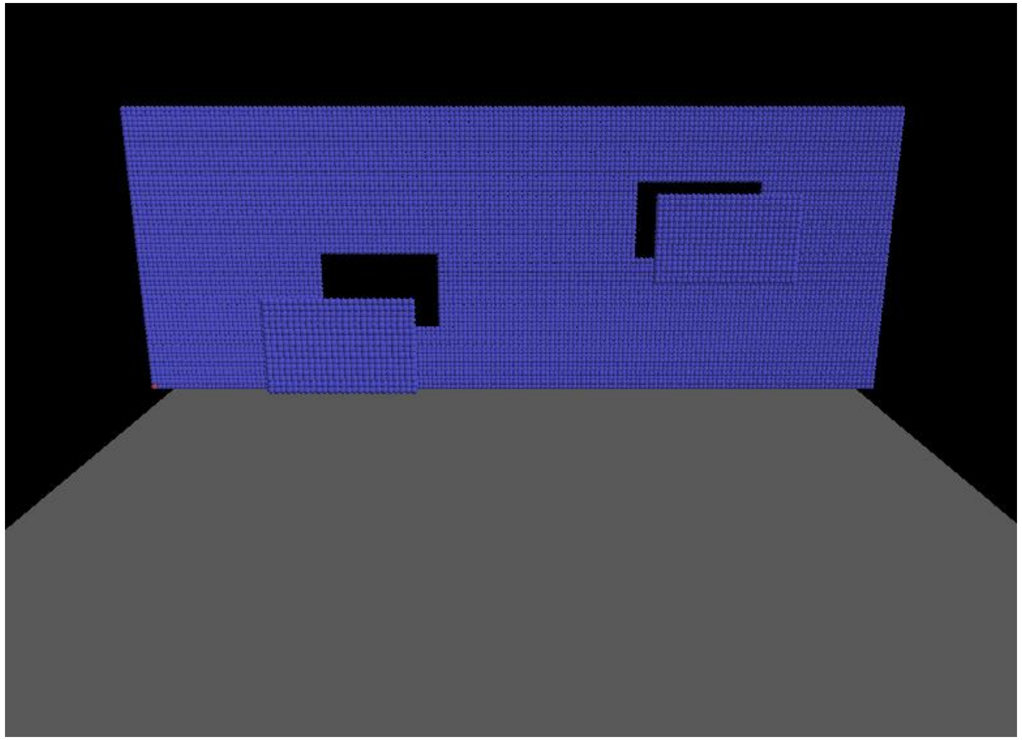


Image Rendering: Data Collection

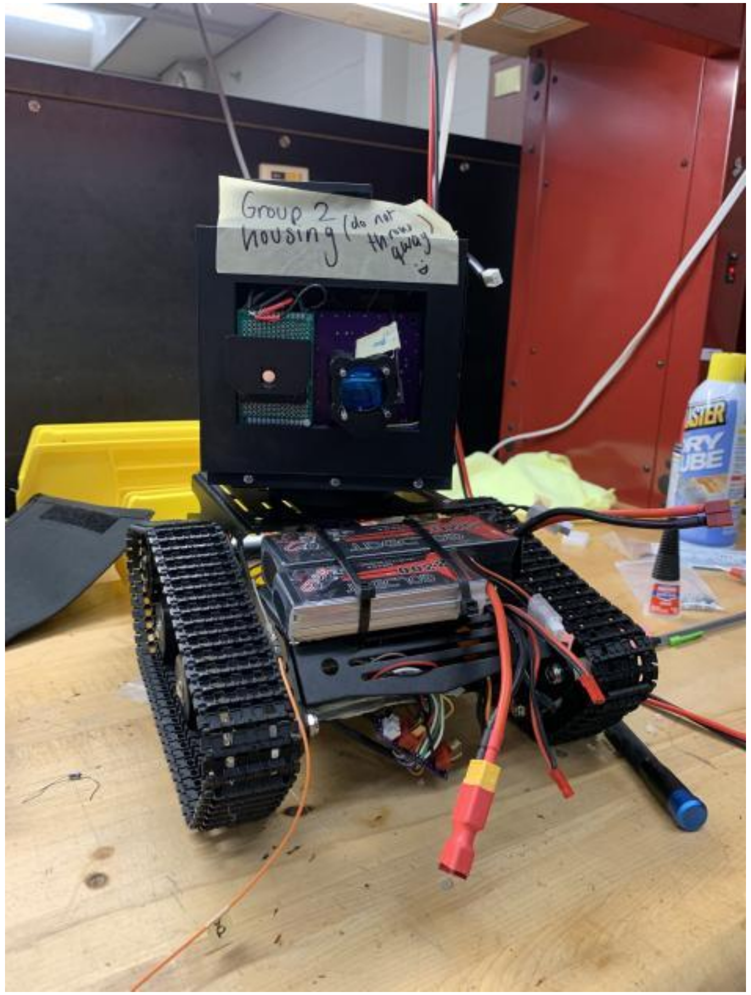
- When RC Platform is positioned in the desired location, the sensor scans and processes the incoming data
  - Incoming data is collected and written to a file to image rendering
- Due to the intensity of the object presented in front of the sensor, that data is transferred to distance data
  - Data is processed in the files bytes by bytes to gather and display it correctly

Image Rendering

- Data collected from the sensor transmits to distance data that is then used to plot into the 3D space
  - Each point collected transmits directly to its own point in the 3D map
  - Calculates the points associated with the sensor
- Image shows two boxes at a different height and depth, displays correctly during image rendering



Lessons learned and Final Thoughts



Thank you for watching.

- Group 2