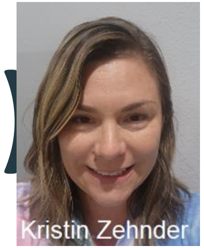
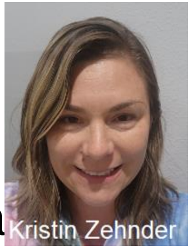
Road Surface Mapping (RSM)



Group 3

- Gregory Flynn - Computer Engineering
- Jason Gerlach- Photonic Engineering
- Ivan Pavlov - Computer Engineering
- Kenady Saus- Electrical Engineering
- Kristin Zehnder- Photonic Engineering

Motivation



❖ Project Description

*To safely, quickly, and accurately locate road damage a geo tag it for repair.

*To solve the problem of poorly maintained roads in isolated locations.

❖ Economic Benefits

*More efficient use of road repair crews

*Less damaged roads leads to less wear and tear on vehicles.

Overall Goal: To safely and effectively map the road’s surface and tag road damage with GPS tracking.

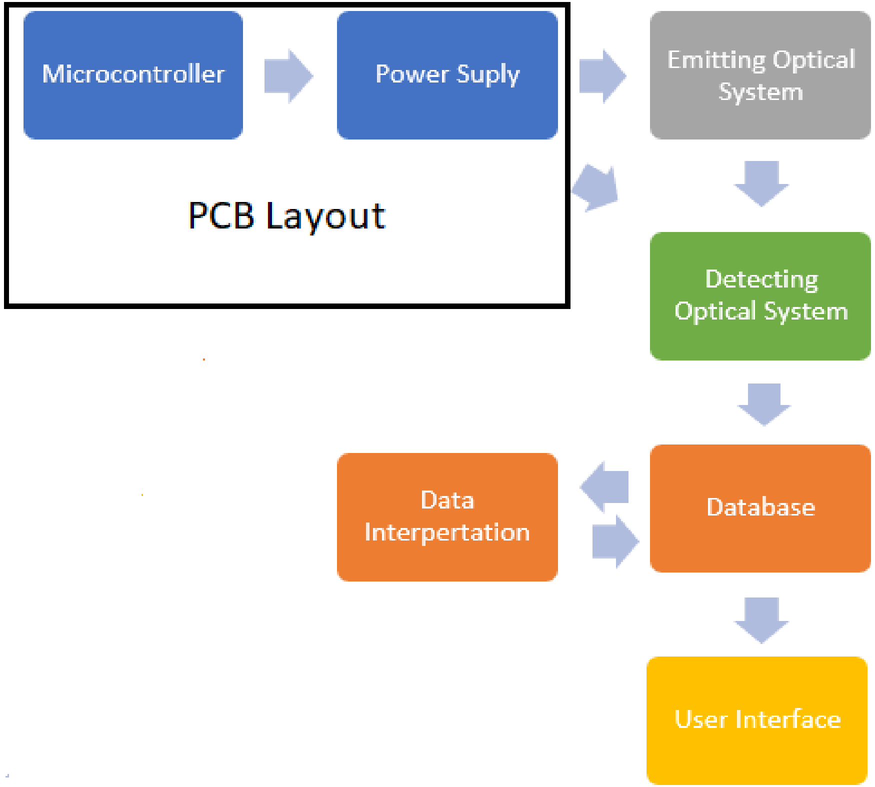
Goals and Objectives

Type of Goal	Goals	Objective
Basic	Longitudinal resolution of 4in X 4in	Camera fps at a min of 40 Hz and vehicle speed of 10 mp
	Damage depth resolution of at least 6 in	Powell lens is set at an angle large enough to create a sh the laser line when it encounters a pothole.
	Power the web camera and laser diode	Split power sources to provide 5V current to the camera as well as what is needed for the diode.
	Data processing	Use computer vision principles to find differences in elevation.
Advance	Longitudinal resolution of 2in X 2in	Camera fps at a min of 176 Hz and vehicle speed of 20 mph.
	Damage depth resolution of at least 4 in	Powell lens is set at an angle large enough to create a shift in the laser line when it encounters a pothole.
	Power any added components	Supply enough power for another device to be added or for a camera with higher capabilities .
	Website	Create custom 3D geometry that accurately represents the road damage’s dimensions.
Stretch	Longitudinal resolution of 1in X 1in	Camera fps at a min of 528 Hz and vehicle speed of 30 mph.
	Damage depth resolution of at least 2 in	Powell lens is set at an angle large enough to create a shift in the laser line when it encounters a pothole.
	Provide visual battery charge level	Display the current battery charge so the user can approximate how much longer the device can run.
	Website	Provide advanced calculations like “volume of concrete needed to fix damage”.

Specifications

Requirement	Specification	
Due to the difference in ambient light, during the day, a spatial filter aperture and bandpass filter is used which reduces the length of the line. At night, the aperture and bandpass filter is not needed allowing the line to be longer.	Day: 2 ft. Night: 6 ft.	High
The DOT classification for a pothole is 2 in depth or 8 in diameter. The margin of error must be acceptable between the calculated and actual values.	Less than 50% difference	High
Longitudinal resolution to detect a pothole is determined by the camera's frame rate and vehicle speed. The car will travel at a minimum speed to be safe to use on most roads. A minimum frame rate of 40 and a speed of 10mph must be used. As the frame rate increases, the faster the vehicle may go.	Maximum longitudinal resolution of 4 inches.	High
GPS tag locations of scans so the road damage can be located. Location is based on the center of the saved scan.	Less than 5 meters	High
Laser emits in visible spectrum as the camera can only process visible light.	400 nm to 700 nm	High
The design of the PCB allows for adequate power consumption to maintain operation off all components	Less than 6 volts	High
Google Lighthouse Metrics: Performance, Accessibility, Best Practices, SEO	> 80/100 all metrics	Low

Overall Block Diagram



Gregory Flynn
- Computer Engineer

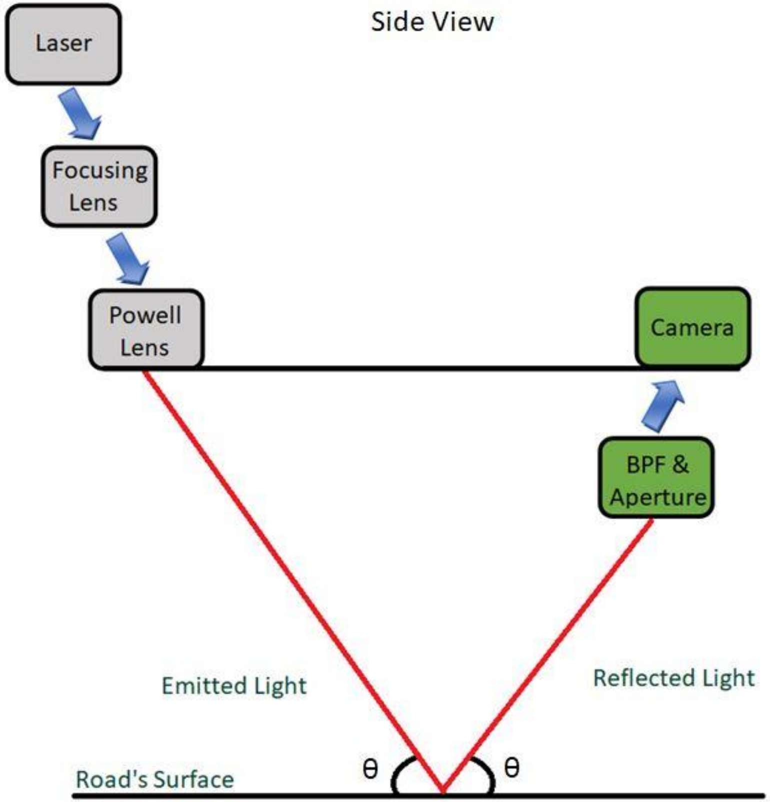
Jason Gerlach
- Photonic Engineer

Ivan Pavlov
- Computer Engineer

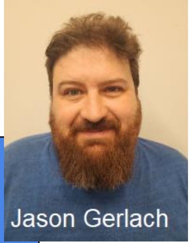
Kenady Saus
- Electrical Engineer

Kristin Zehnder
- Photonic Engineer

Optical Block Diagram



Cylindrical VS. Powell Lens

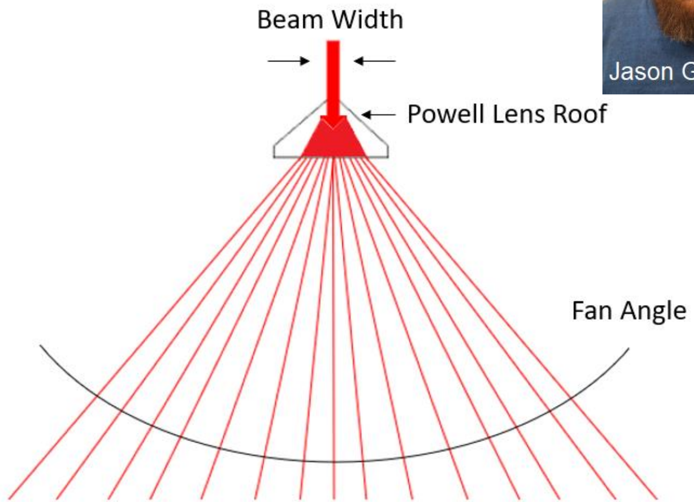


Decision Factors		Cylindrical Lens	Powell Lens
Criteria	Weight		
Cost	3.0	1.0	2.0
Uniform intensity	3.0	1.0	2.0
Weighted Total		6.0	12.0

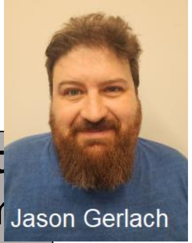
- Cylindrical lens costs at least twice as much
- Cylindrical Lens produces a nonuniform Guassian beam
- Powell lens produces a uniform beam.

Powell Lens Characteristics

- 110 Degree Fan Angle
- Collimated Beam
- Elliptical Beam Profile
  - 1 mm width
- End 10% of the generated line has diminished intensity



Laser Wavelength



Decision Factors		Red Light (630 - 670 nm)	GreenLight (520 -532 nm)	Blue Light (360-480 nm
Criteria	Weight			
Cost	2.0	4.25	3.5	0.5
Safety	2.0	3.0	1.0	2.0
Wavelength	1.0	1.0	2.0	1.0
Weighted Total		15.5	11	6

- Red Laser is the cheapest
- Green Light - Safety concerns to due IR radiation
- Blue Light - Photobiological effects

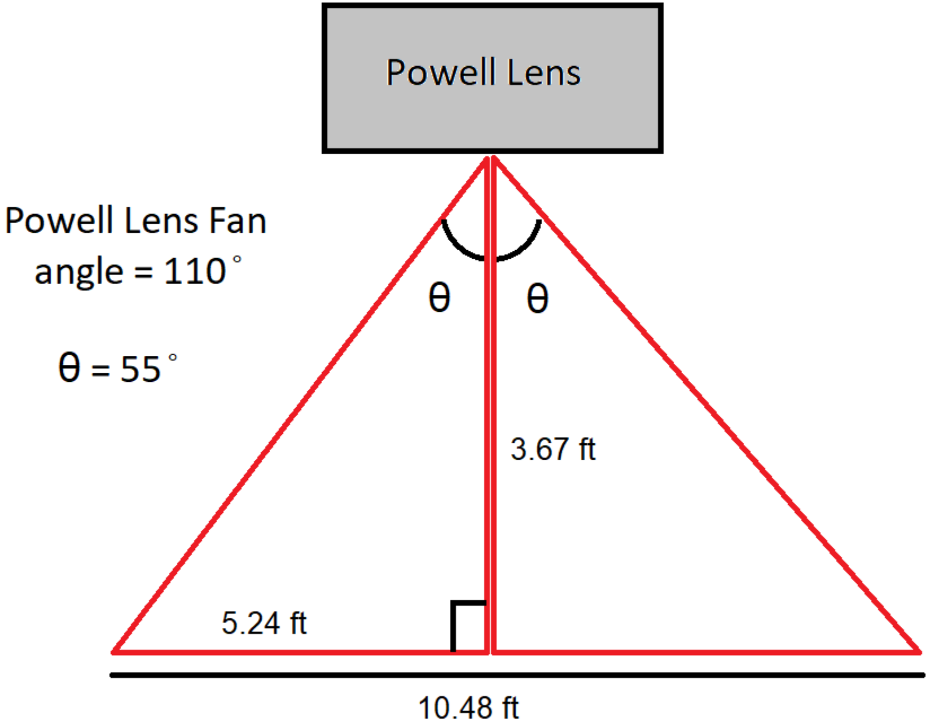
Laser Characteristics

- Wavelength: 635 nm
- Beam is collimated
- Beam Size: 4mm X 1mm
  - beam size when lens is 50.8 mm from the front of the laser housing.
- Max Power 5mW
- Laser Class 3R

- Reflected Light
  - Reflects 4.2% or 0.21mw and transmit 4.79 mW of power through lens

$$R = 100\left(\frac{n_{air} - n_{glass}}{n_{air} + n_{glass}}\right)^2$$

Emitting System Optical Design



- Generated line is 10.48 ft
- Detector will detect a 2 ft line
- Powell lens should be placed 3.67 ft above the road's surface

Detecting System Components

Camera Characteristics

- Frames per second
  - fps = (63360 in/mi x speed)/(3600 sec/hr x resolution)
- Resolution
  - More pixels the better the quality or crispness of an image
- Field of View
  - Horizontal FOV must be large enough to capture the entire projected line.

Specification	Frames per Second
4 in Resolution at 10mph	44
4 in Resolution at 20mph	88
4 in Resolution at 30mph	132
2 in Resolution at 10mph	88
2 in Resolution at 20mph	176
2 in Resolution at 30mph	264
1 in Resolution at 10mph	176
1 in Resolution at 20mph	352
1 in Resolution at 30mph	528

Camera Characteristics

Features of ELP-USBFHD085-MFV and ELP-USBFHD03AF-A10

Feature	ELP-USBFHD085-MFV	ELP-USBFHD03AF-A1
Pixel Size	2.0 um X 2.0 um	3.0 um x 3.0 um
Image Area	5440 um X 3072 um	5856 um X 3276 um
Frames per Second	260	100
Resolution	640 X 360	640 X480
FOV	100 degrees	100 degrees
Price	$77.00	$50.99

Spatial Filter Aperture

Black Body Radiation

$$L_{max} = 2.89 \cdot \frac{10^{-3}}{T}$$

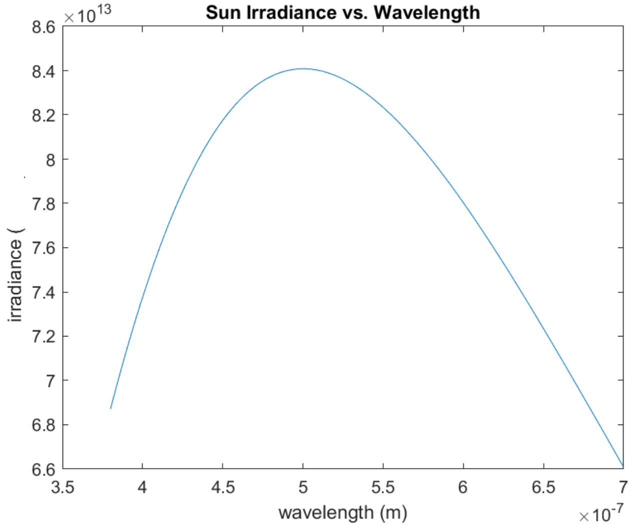
$$U_{max} = \frac{10^{-9} \cdot 2 \cdot h \cdot c^4 \cdot \pi \cdot \left(\exp\left(\frac{h \cdot c}{L_{max} \cdot T \cdot k}\right) - 1.0\right)^{-1}}{(L_{max})^5}$$

- ○ 498.3 nm @ 8.4280 W/m^2

Laser Power = 0.4174 W/m^2

❖ Low signal to noise ratio

- Aperture decreased noise, but not enough for the camera's sensor to capture the reflected light

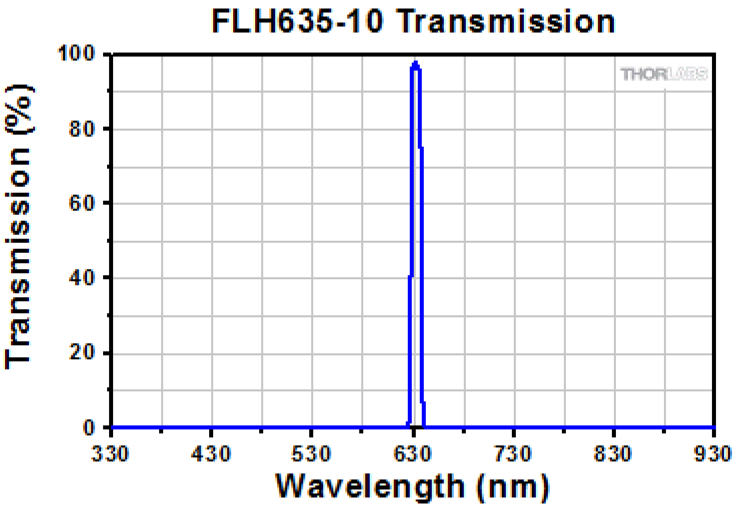


Bandpass Filter

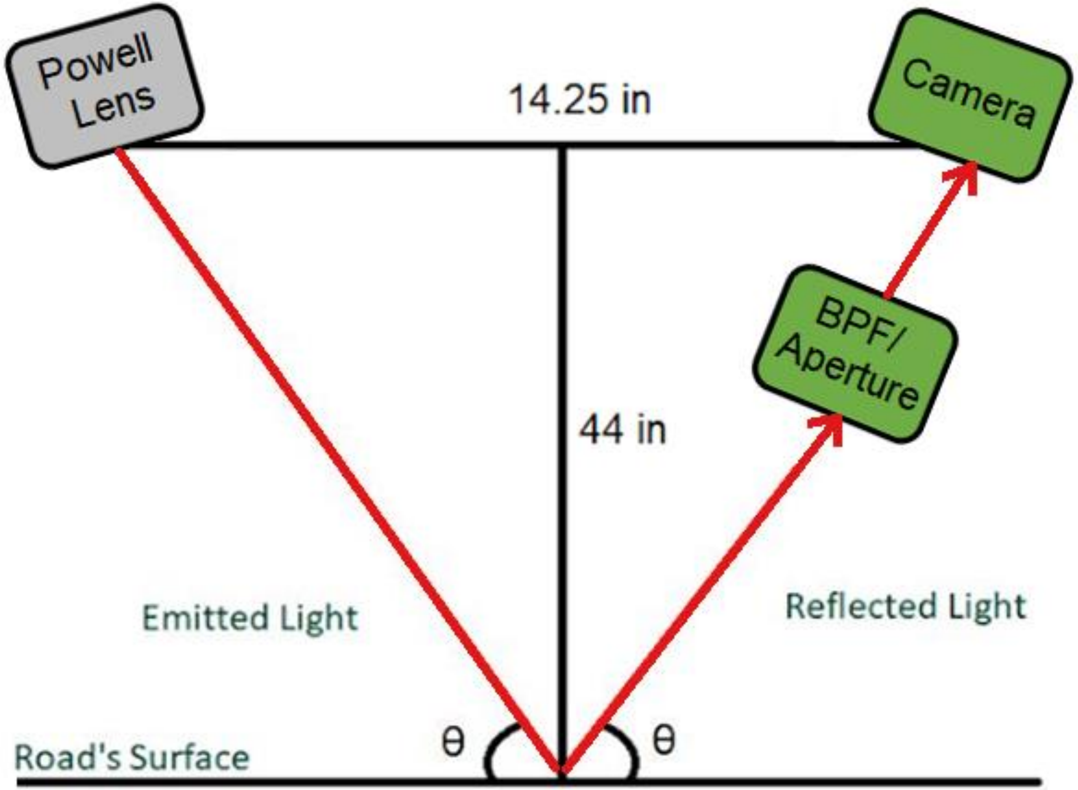
- 10nm filter centered on 635 nm allowing 90% transmission
- 630nm and 640 nm allow 45% transmission
- 0.01% transmission allowed outside the

- Post filtering 0.3757 W/m was allowed through the filter which is less than the laser power of 0.4174 W/m

- Future Plans
  - The BPF requires 0 angle of incidence
  - Decreases captured light to 2 ft
  - Collimator lens

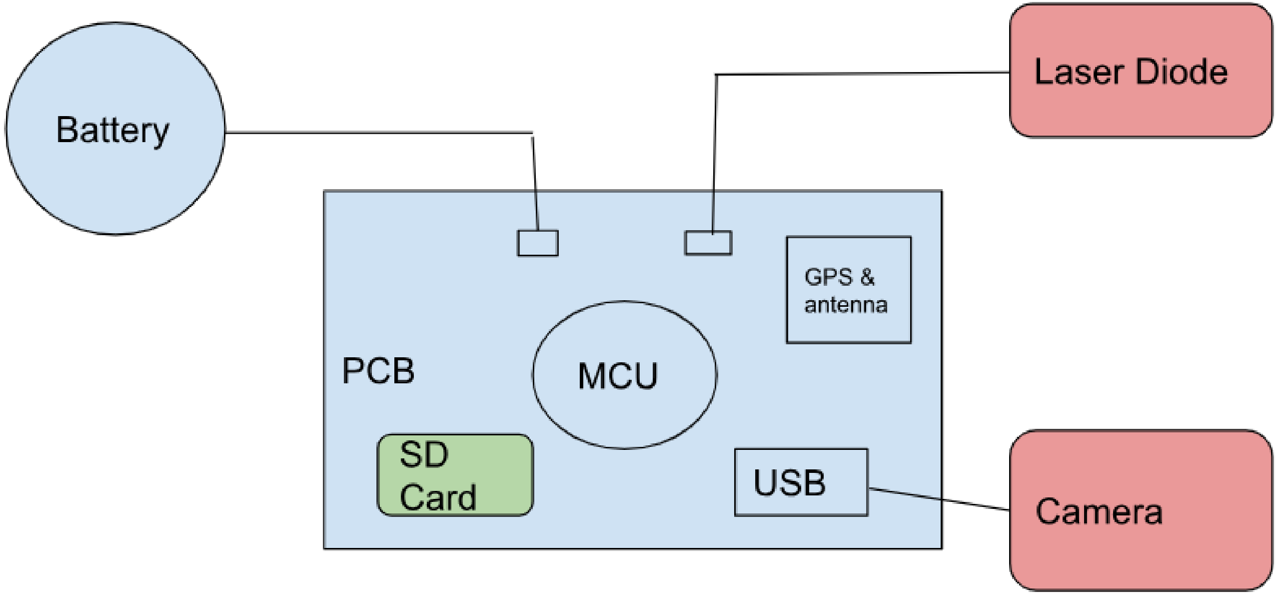


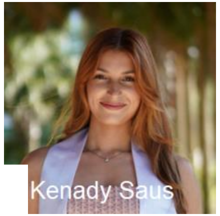
Overall Optical System Design



- θ = 80.8
- Light is reflected off the surface of the road at the same angle
- At a height of 44 inches, the powell lens and camera must be 14.25 inches apart.

Electrical Block Diagram

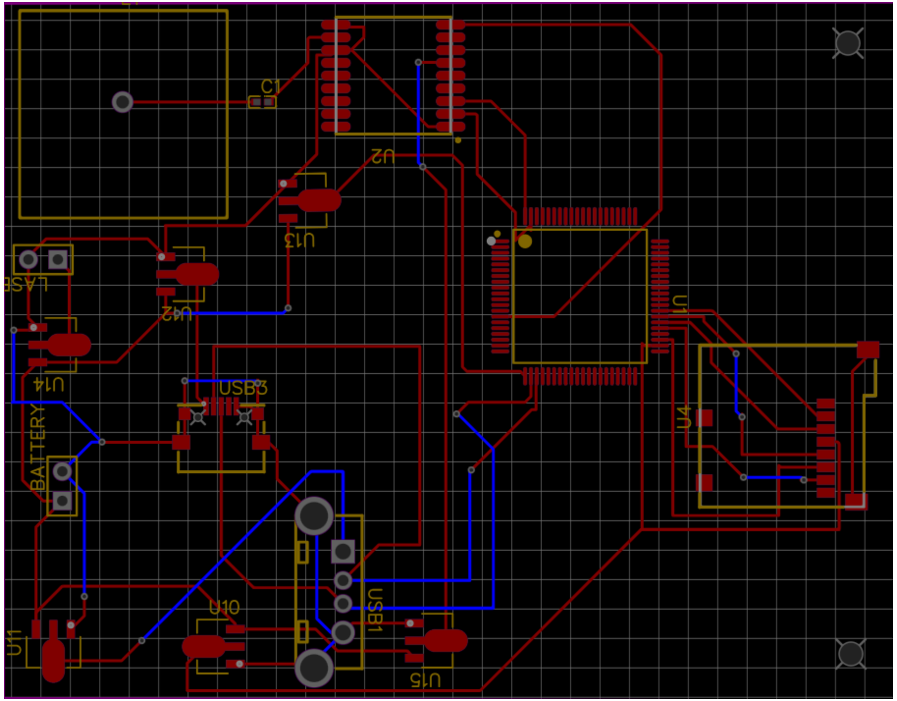




Electrical System Approach

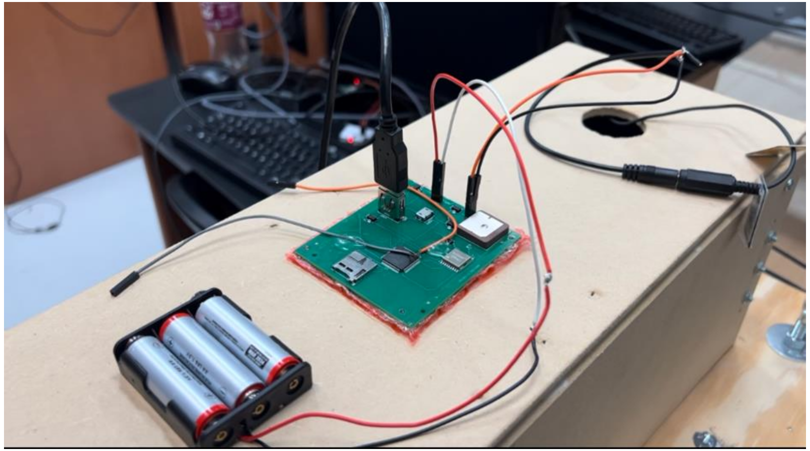
- The system required a microcontroller that could handle the capabilities of supporting the UART programming functions as well as supporting the USB and SD card slots
- PCB designed from MCU selection provided for assembly by JLC
  - Chosen MCU, LPC1752FBD80
  - Independently add micro SD slot, USB slot, etc. to the PCB
- Power supply designed with goal of reliability and efficiency to be sustained throughout the overall system use
- Once the PCB arrived we tested design functionality to determine is alterations were necessary
  - Multiple board ordered for alterations / soldering purposes
- Implement test cases to the system
  - Add new parts to test power capabilities when new power consumption is added

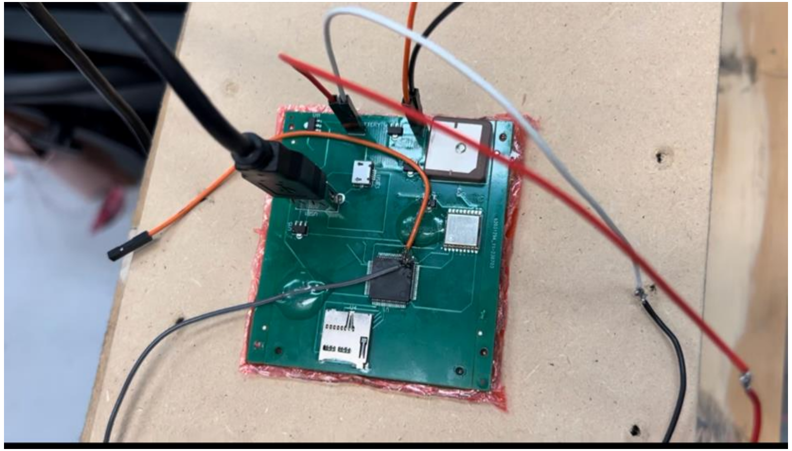
Current PCB Design



- PCB design used for final demo
- Method of voltage regulators used to step down the input voltage to the necessary value
- If less of a time constraint, we would have altered for easier programming techniques

Final PCB



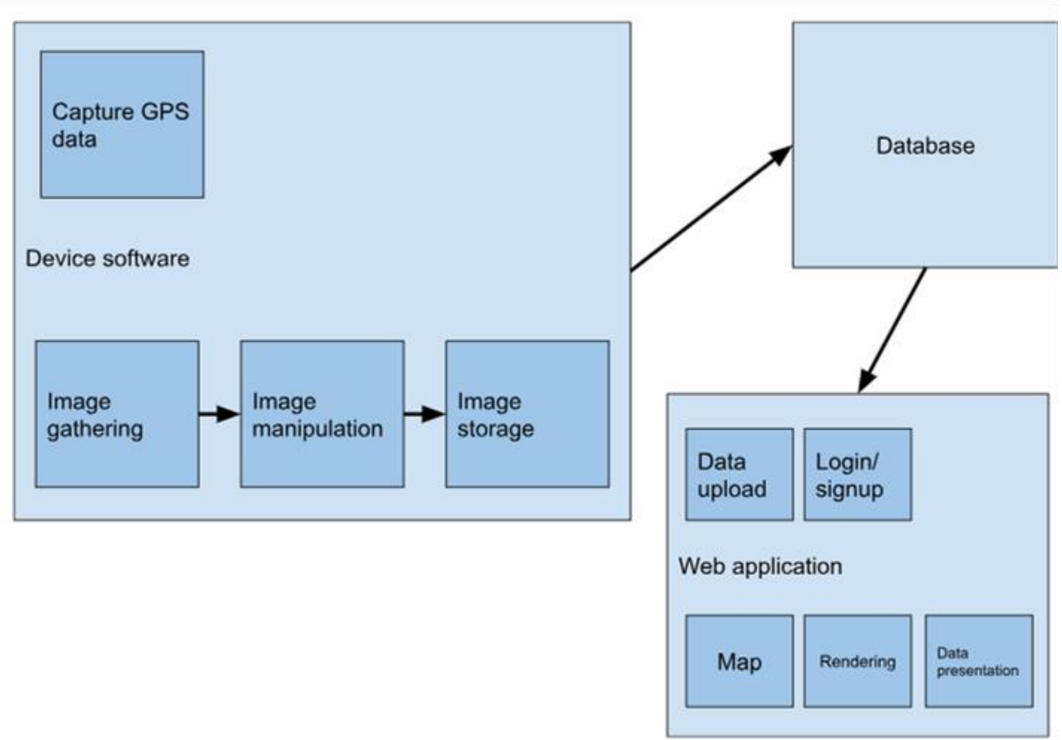


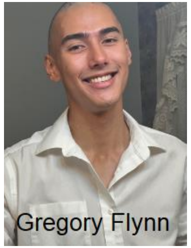
- Images represent the final functioning system involving the PCB within the housing design
- The image to the right shows a closer look at the adjustments made with the jumpers for programming

PCB / Electrical System Issues

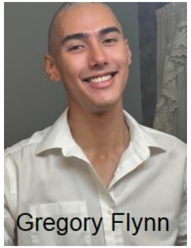
- Encountered issue with first MPU design when transferring to PCB
  - Pin package was too high for beginner level design and multiple voltage variations required
  - Search for new processor; encouraged incorporating only the necessary voltage regulation techniques for our board
- First board ordered experienced a delay in production
  - Selected GPS module was out of stock and had to wait for the turn-over time for restock
  - This caused future issues in re-design for the final board, testing of the design was pushed off until the first design delivered
- New MCU chosen
  - Defect in micro usb connect to the designated programming pins
  - Due to time constraints, we could not order a new PCB with a usb-c attachment for the programming ability, instead had to solder jumpers to the swd pins to create the ability of transferring the program from another device
  - Due to this defect most of the extra boards ordered were utilized to practice adjustment for the mistake

Software Block Diagram





Software Design Approach

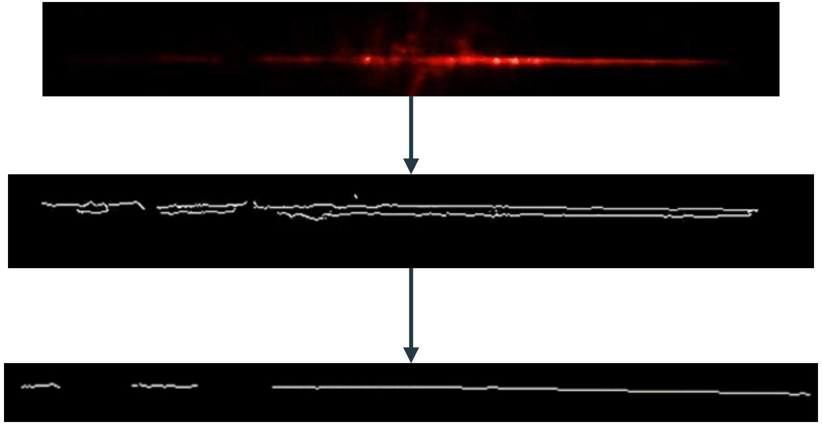


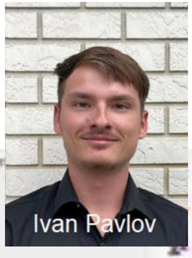
- Add google maps to react project
- Add 3D rendering to react project
- Connect google maps location with a rendering
- Connect map locations with database data
- Present database data on front end
- Create software for MCU
  - read data from the camera and gps
  - Store camera data on sd card
- Program MCU
- Create computer vision program which reads sd card data and stores a saved point cloud file

Computer vision program

- Inputs:
  - Name of video file
  - Name of gps file
  - Depth to remove off road
- Steps
  - Cropping video into a smaller format for quicker processing
  - Canny edge detection on red channel of the video with gaussian blur
  - Merge edges algorithm
  - Process edges video and remove a height from the road
    - Send edges to a preliminary point cloud file
  - Filter out statistical outliers
  - Divide the scan into ten equal partitions and ask the user if they want to save
  - Saved partitions find the closest GPS location in terms of time
- Output
  - Output of the file is named after the GPS coordinates of the pothole
    - Example: "28.601726_-81.196485.ply"

Edges detected





Point Cloud calculations

Y-value: Point in the direction of travel

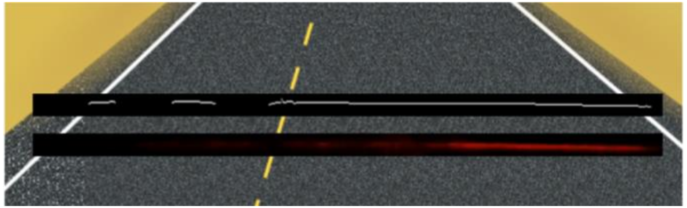
$$Y = \frac{speed}{FPS} * frame = \frac{176\ in/s}{60\ fps} * frame$$

X-value: Position along laser line

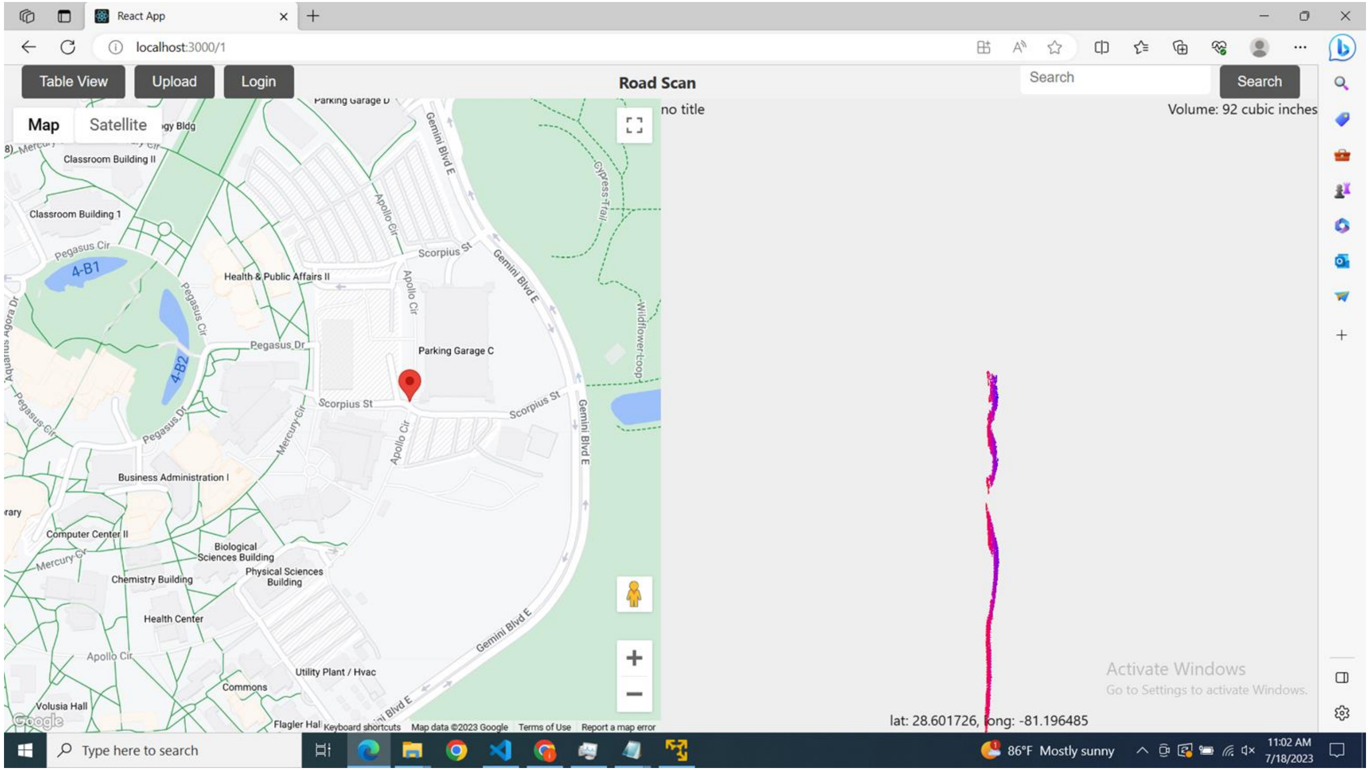
$$X = \frac{width}{resolution\ width} * pixel.x = \frac{57\ in}{640\ pixels} * pixel.x$$

Z-value: Depth of a pixel shift

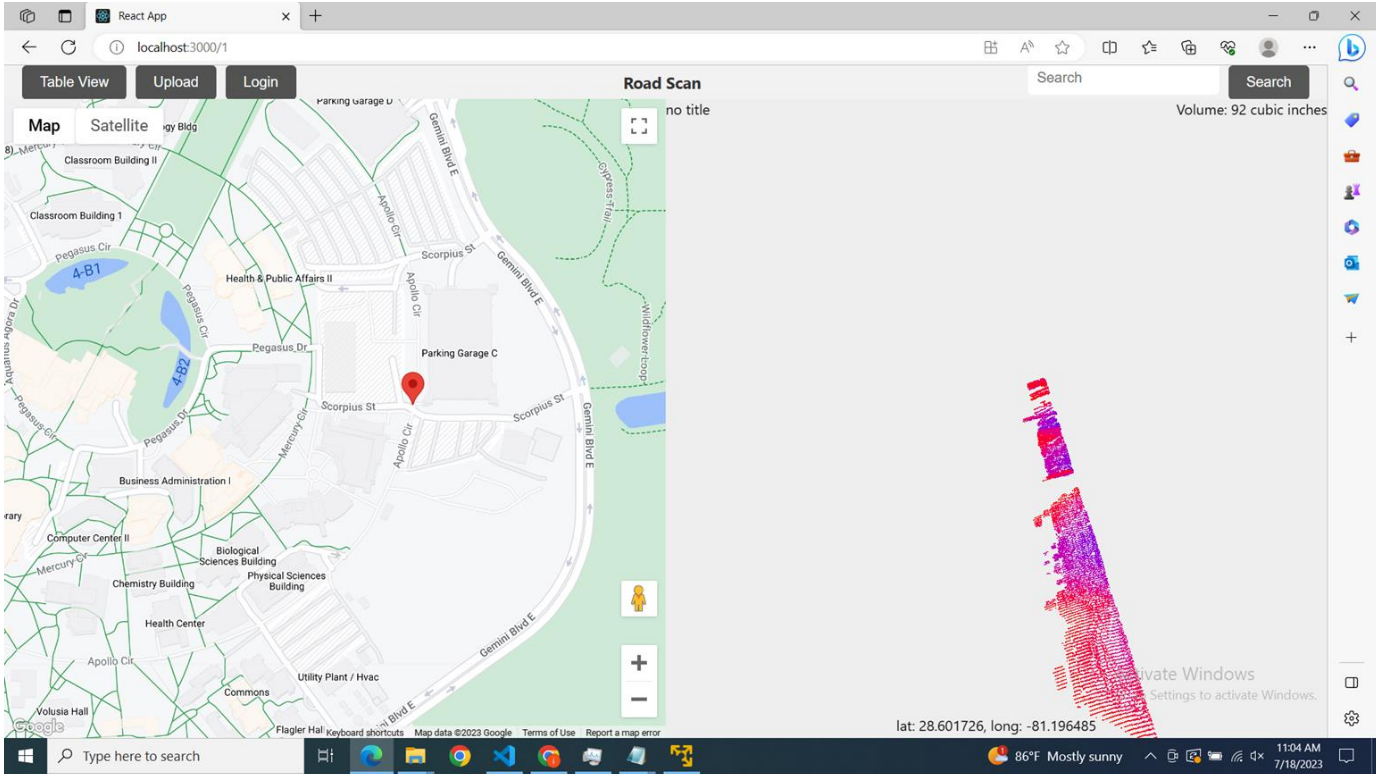
$$Z = -(-Initial.y + pixel.y) / 2$$

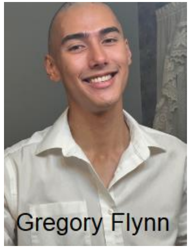


Website features



Website Features

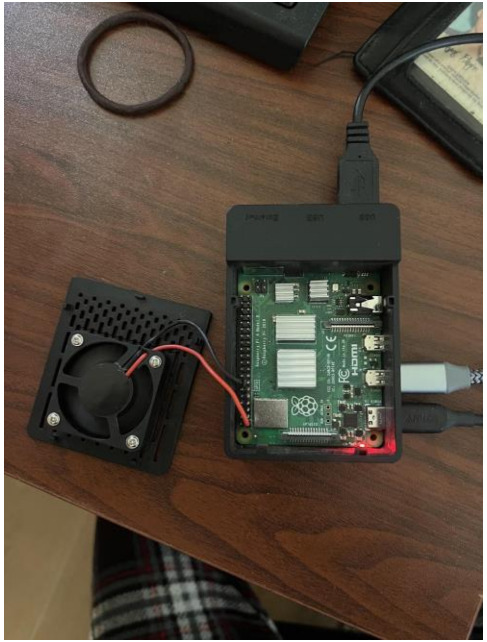




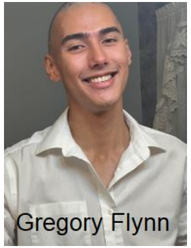
Back End Design: Do-it-yourself

How to host a website from the comfort of your room:

1. Define a GET response for the root route (“/”)
   a. res.send(“<h1>Hello World</h1>”);
2. Type “npm {yourJavaScriptFileName}.js” in the console
3. Download NGINX, edit the config file to proxy to “localhost:{yourPort}”
4. Port forward
5. Your website is now available to the World Wide Web!



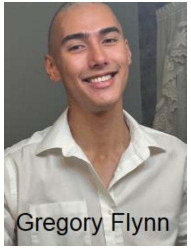
Lessons from do-it-yourself...



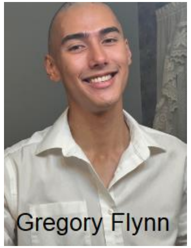
- The website ceases to exist once your Internet Service Provider changes your dynamic IP.
- To access your website people have to type in something like “204.310.19.321” which rolls right off the tongue.
- Nightmares from opening your home network to the world wide web… of hackers.

Database Options?

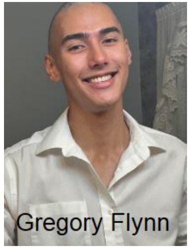

```
mysql> CREATE DATABASE test;
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> USE test;
Database changed
mysql> DROP TABLE IF EXISTS `t_mqtt_msg`;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CREATE TABLE `t_mqtt_msg` (
    ->   `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
    ->   `msgid` varchar(64) DEFAULT NULL,
    ->   `topic` varchar(255) NOT NULL,
    ->   `qos` tinyint(1) NOT NULL DEFAULT '0',
    ->   `payload` blob,
    ->   `arrived` datetime NOT NULL,
    ->   PRIMARY KEY (`id`),
    ->   INDEX topic_index(`id`, `topic`)
    -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8MB4;
Query OK, 0 rows affected (0.07 sec)

mysql> describe t_mqtt_msg;
+---------+------------------+------+-----+---------+----------------+
| Field   | Type             | Null | Key | Default | Extra          |
+---------+------------------+------+-----+---------+----------------+
| id      | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| msgid   | varchar(64)      | YES  |     | NULL    |                |
| topic   | varchar(255)     | NO   |     | NULL    |                |
| qos     | tinyint(1)       | NO   |     | 0       |                |
| payload | blob             | YES  |     | NULL    |                |
| arrived | datetime         | NO   |     | NULL    |                |
+---------+------------------+------+-----+---------+----------------+
6 rows in set (0.02 sec)
```



Back End: Endgame



“Hey there... ever wanted to `bypass capacity planning, configuration, management, maintenance, fault tolerance, or scaling of containers, VMs, or physical servers?...`”



Back End: Hybrid vs Serverless

Hybrid

Use RPi4 as database

Dockerize Next.js project

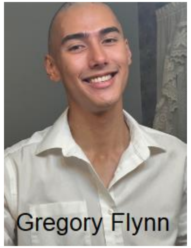
Deploy to Google Cloud Run

Serve content to client

Serverless

Serve content to client

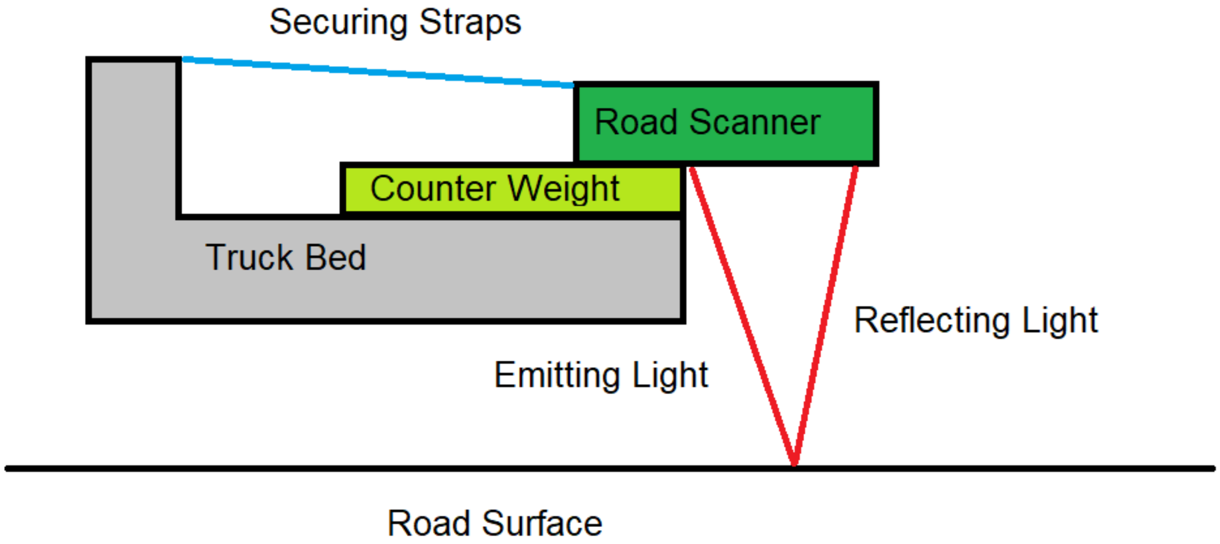
Back End Issues



- Raspberry Pi 4B Processor (ARM Cortex-A72) does not support vectorized instructions.
  - If database query/retrieval speed is an issue, can migrate database to another computer with a more capable CPU.
- The hybrid model nullifies the adjacency bonus of having the server and the database together, increasing the round trip time for the user, making them wait (bad!)
  - Solved by ~~succumbing~~ migrating to the serverless computing model.
  - Alternatively, Next.js provides the ability to add caching headers to data fetch requests.
- **Going Forward**: try serverless computing because connecting the front-end to the back-end needs to happen soon.

Vehicle Mounting System

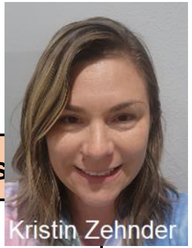
Truck Mounting Diagram



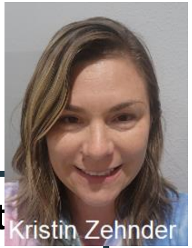
Budget and Financing

Item Description	Quantity	Total Cost
Red Laser Module	1	$123.87
Camera	1	$75.00
Powell Lens	1	$37.28
Bandpass FIlter	1	$185.60
Wiring	1	$20.00
Battery (5 volt)	1	$4.00
PCB USB port	5	$0.98
PCB Voltage Reg.	31	$1.34
PCB Micro SD Slot	5	$2.27
PCB board / assembly	5	$94.63
PLA filament	2	$41.17

Item Description	Quantity	Total Cos
Maps api	$200 monthly credit	$0.00
SD Card (256 GB)	1	$37.00
Mounting Bracket	1	$54.37
System Casing	1	$69.48
PCB Antenna	5	$4.14
PCB Male Connectors	10	$0.37
PCB GPS Module	5	$23.21
PCB MCU	5	$37.92
PCB Micro USB	5	$0.13
	Total	$812.76



Work Distribution



Name	Emitting System	Detection System	PCB	Data Processing	Device Software	Mount System
Gregory Flynn			S	S	P	
Jason Gerlach	P	S				P
Ivan Pavlov				P	S	
Kenady Saus			P			
Kristin Zehnder	S	P				S

Thank You!

Dr. Sonali Das ECE
Dr. Peter Delfyett CREOL
Dr. Arup Guha CPE
Dr. Lei Wei
Dr. Advanka Kar

QUESTIONS?