# SafeSight

## Group 25

Matthew Carvajal
Computer Engineer

Albert Lougedo
Electrical Engineer

Mahyar Mahtabfar
Electrical Engineer

# Project Motivation & Background


Rush hour traffic - Miami, FL

Car Dependency in the U.S.

- About 92% of U.S. households have access to a car but only 55% of Americans have access to public transit. Additionally, only 3.5% of that 55% take advantage.

Roadway  Construction vs. Vehicle and Population Growth

- SInce 1980, U.S. population has grown by 94 million or 41%, but highway capacity increased a mere 10.3%. With vehicle miles of travel more than doubling (102.3%)

Miami's Crippling Infrastructure

- From 2021 to 2022, Miami's traffic has seen an increase of 30%. The city now ranks as the 8th most congested city on the planet. The city's failure to invest in public transportation, roadway development, and the population surge from post pandemic newcomers have all contributed to this current situation.

# Distracted Driving Vs. Traffic/Safety

A steady rise in Digital Distraction

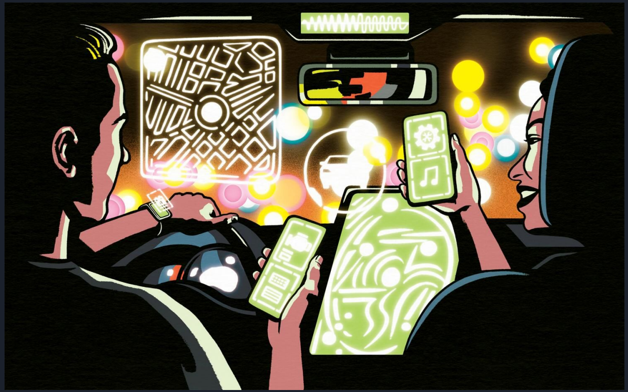
- On average, a person who owns a cellphone will check it around 96 times a day
- Addictive cell phone applications and improving technology in vehicles are to blame for an increase in distracted driving and elevated risks.

Safety Concerns

- Distracted driving leads to reduced safety and traffic flow.
- According to NHTSA, distracted driving claimed the lives of 3,275 in 2023 and claims that most distracted driving incidents are failed to be reported.

Proposition for the Problem at Hand

- A fully autonomous device that is compatible with any motor vehicle on the road and actively works to stop distracted driving, while collecting valuable driving data.



How technology has made your car a "candy store of distraction"

- *Los Angeles Times*

# Introduction of Safe Sight

Safe Sight

- Create a product that collects data to expose attention infractions of drivers, while making real-time efforts to maintain driver attention on the road.

The Modern Horn Vs. Safe Sight

- Universally, a blaring horn is used as communication between vehicles and to alert other drivers of danger. Distracted drivers rely on other drivers to correct their attention on the roadway.
- Safe Sight will actively monitor the drivers posture in relation to their speed, acceleration, and reaction to changing traffic lights to catch and eliminate distracted behavior as soon as it starts.

Possible Markets and Applications

- Insurance, Company Vehicles, Long Distance Driving

# Safe Sight Goals

Basic

- Maintain driver attention to the road and environment.
- Analyze common traffic lights and patterns
- Establish communication between components through the ESP32.
- Train CV to analyze environment at ~30FPS.

Advanced

- Develop the application that transmits driver data to driver device
- Take images and video to record attention disturbances.
- Implement interrupts to have MCU on low power mode.

Stretch

- Pre-collision alarm for imminent danger.

# Safe Sight Objectives

Benefits

- Ensure drivers remain attentive to the road at all times, including traffic stops
- Minimize distracted driving incidents, reducing the risk of accidents
- Improve overall traffic flow and roadway safety through proactive driver monitoring.

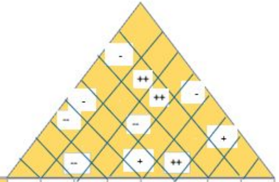
Capabilities

- Monitor driver posture to detect signs of distraction or drowsiness
- Measure acceleration and distance between vehicle ahead in relation to traffic lights in certain traffic stop scenarios
- Operate autonomously with no input from the driver, driver will be made aware of attention infractions from real-time notifications and alarms, and given a drivers report after driving
- Collect valuable driving data that can be displayed on a mobile application

# Engineering Specifications

| Component(s) | Parameter | Specification |
| --- | --- | --- |
| Application | Distraction tracking / Photo View | Tracks number of distractions and shows photos taken |
| Cameras, raspberry Pi, sensors | Pedestrian, light and motion detection | CV algorithms for color thresholding, size change analysis, and face orientation recognition. |
| Piezoelectric Buzzer | Beep sound/Action Delay | 102dB +/- 3dB , 600 Hz to 10kHz |
| Power | Cigarette lighter adapter | 12V |
| Microcontroller ESP 32 | Performs calculations and communicates with Pi. Transmits data to the application host device. | Data transfer >= 5Mbs |
| IC (integrated circuit) | Non-inverting Operational Amplifier to perform message signal amplification. | Gain of range from 50-100mV |

# House of Quality



House Of Quality

| | Minimize or Maximize | | Max | Min | Min | Min | Min | Min | Min | Min |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Technical Specifications (How)** | | Attention Infraction Accuracy | Weight | Cost | Latency | Power Consumption | Temperature | Size | Start-up Time |
| | **Customer Requirements (What)** | **Importance** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | Size | 5 | 1 | 9 | 3 | 1 | 5 | 1 | 9 | 1 |
| 2 | Weight | 2 | 1 | 9 | 3 | 1 | 5 | 1 | 9 | 1 |
| 3 | Cost | 3 | 1 | 2 | 9 | 1 | 2 | 1 | 5 | 2 |
| 4 | Ease of Use | 5 | 9 | 1 | 1 | 9 | 1 | 1 | 3 | 9 |
| 5 | Power Consumption | 1 | 3 | 1 | 1 | 1 | 9 | 9 | 1 | 3 |
| 6 | Latency | 5 | 9 | 1 | 1 | 9 | 1 | 1 | 1 | 4 |
| | **Target** | | < %90 | < 32 Ounces | < $300 | < 5ms | < 7.2 Watts per Hour | < 45 Celcius | < 8.5 in X 7 in | < 10 Seconds |
| | **Importance** | | 24 | 23 | 18 | 22 | 23 | 14 | 28 | 20 |

Correlations:
- ++ Strong Positive
- \+ Positive
- \- - Strong Negative
- \- Negative

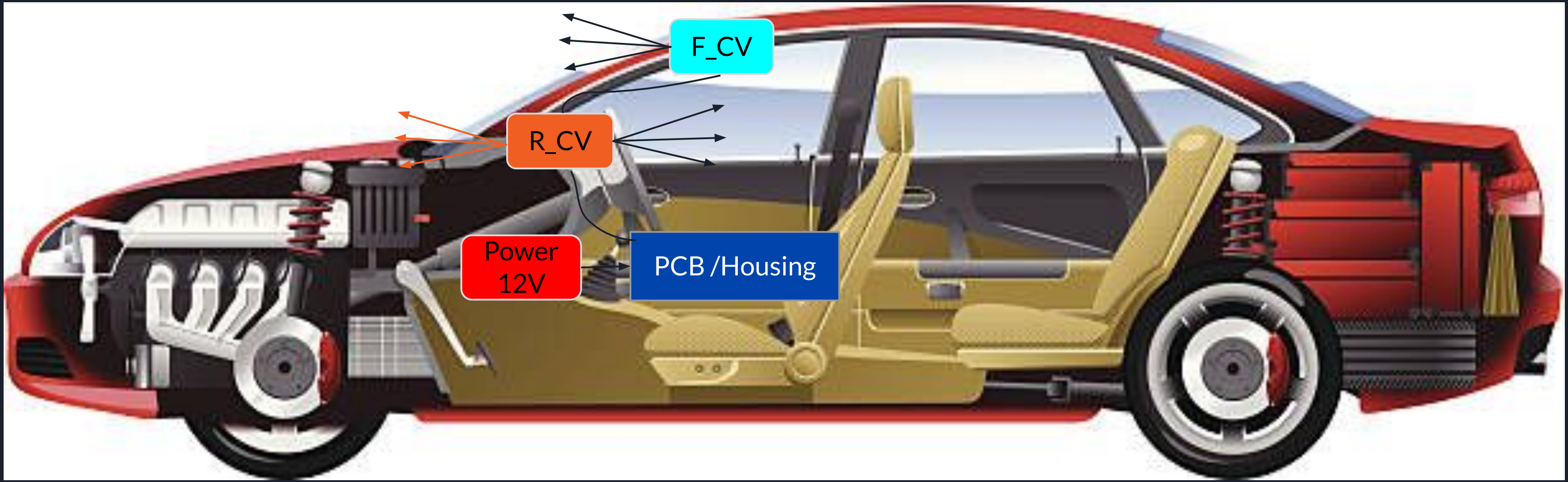
Relationships:
Strong= 9
Medium= 3
Weak= 1

# Safe Sight Model - Concept Overview

F_CV - Forward Facing CV Camera

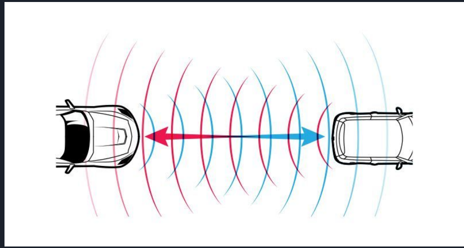
R_CV - Rear Facing CV Camera
- Radar

PCB /Housing
- Raspberry Pi
- Accelerometer
- Piezo Buzzer
- PCB



# Safe Sight Prototype


# Safe Sight Technology



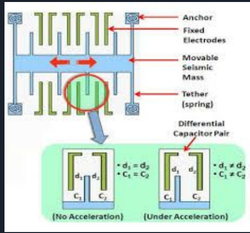
- mm Wave Radar Technology

## Computer Vision

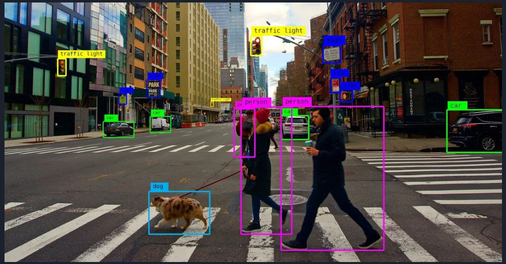
- Analyze traffic lights
- Gather information on the driver’s head posture

## Radar Distance and Accelerometer Motion Sensing

- MEMS Accelerometer technology to measure acceleration and speed relative to the vehicle
- Distance between vehicle ahead and drivers vehicle  in traffic stop scenarios
- Once data on driver speed, acceleration, traffic scenery, and head posture is collected, we can start making real-time decisions on whether or not a driver is distracted in a multitude of scenarios

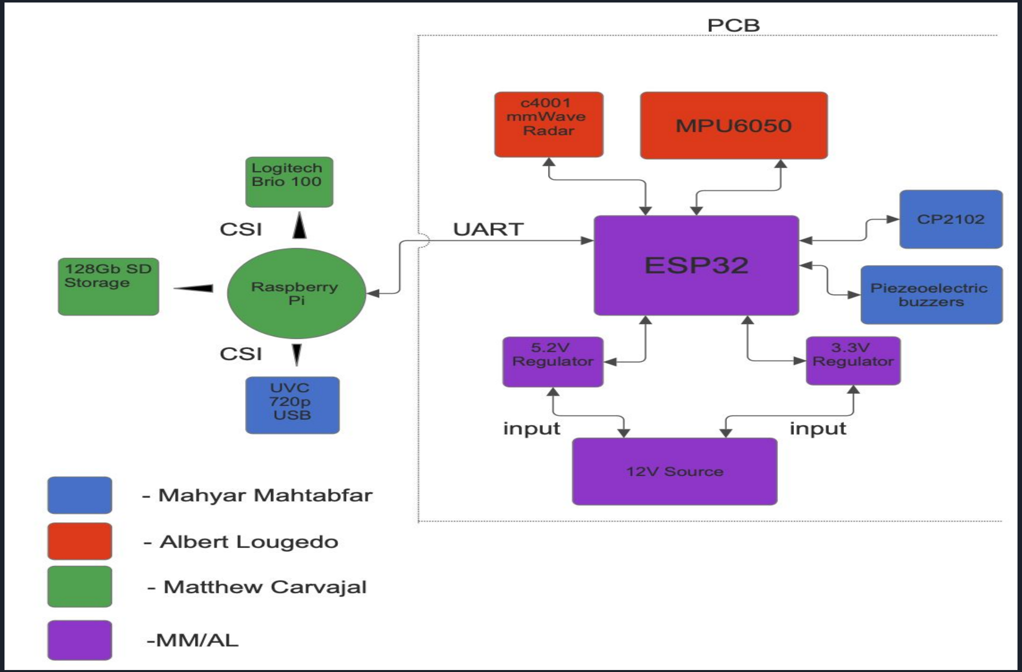



- MEMS Accelerometer Technology



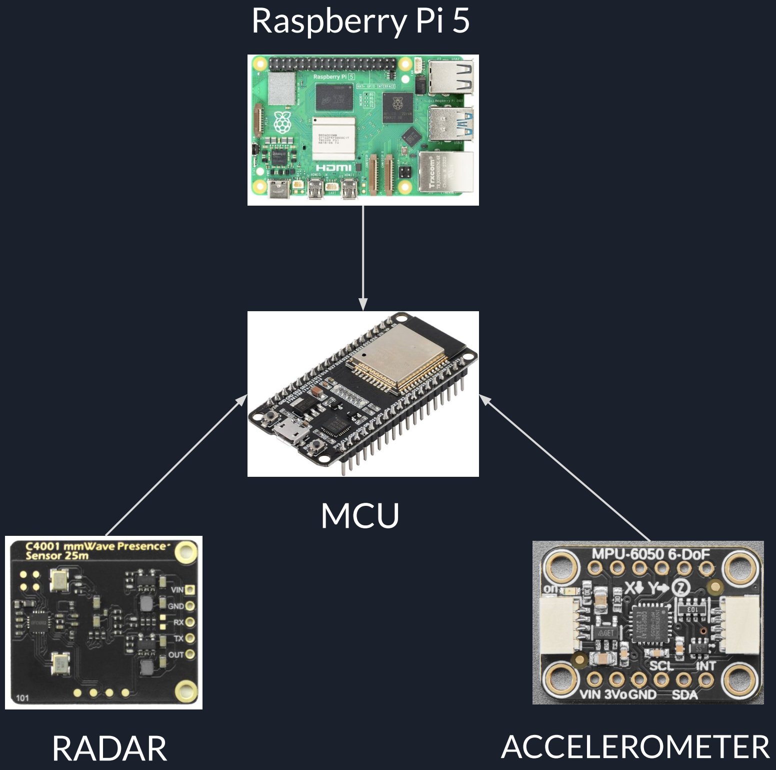

- Computer Vision Technology

# Hardware Diagram



# Major Component Selection Overview

- C4001 mmWave Presence Radar Sensor
  - Operates using millimeter wave radar technology to permeate driver windshield
- Raspberry Pi 5 8GB RAM
  - Serves as the Powerhouse to run the Computer Vision program and communicate to ESP 32 MCU
- ESP32
  - MCU that connects to all major and peripheral components that reads and writes information
- MPU-6050  Accelerometer
  - Provides Acceleration readings for MCU to dictate whether or not the vehicle is in motion
- USB Cameras
  - The “eyes” of safesight - collect real-time traffic and driver posture data to be analyzed by CV algorithms.



# Accelerometer System

MPU-6050

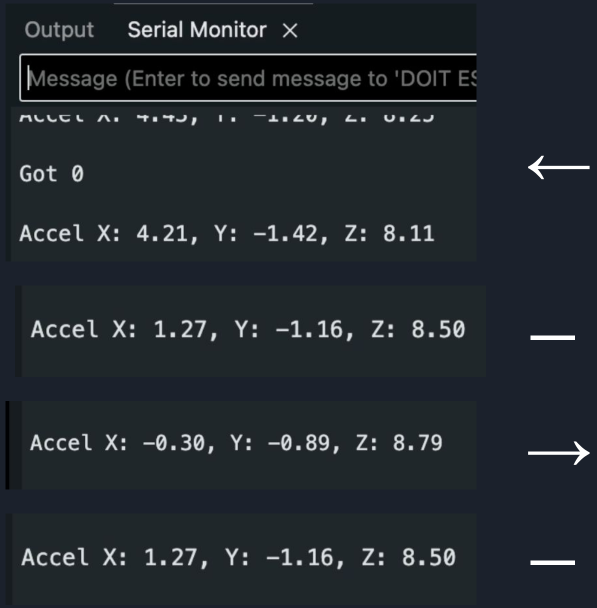
- MEMS device that gives off raw sensor data
- Sensor data is processed at ESP-32 using open-source libraries to provide accurate acceleration readings in real-time given changes in acceleration

Application - 4 States

- Vehicle Acceleration →
- Vehicle Deceleration ←
- Vehicle Idle (Motion) –
- Vehicle Idle (Stopped –

Logic

- Vehicle will be in a constant loop of acceleration and deceleration with given time periods of idle
- Idle state type can be referenced using Vehicle's previous state



# Face Angle Computer Vision

Driver Posture CV Model

- OpenCV face detection and face angle library
- Face detection model plots points and calculates angles
- Face angle values vary depending on head posture

When is it Applied?

- Vehicle in motion
- Traffic light switches from red to green

If Improper Head Posture Detected?

- Make record of the traffic infraction
- Communicate via UART to MCU for logic handling
- Set off Piezo alarm until attention is returned to the roadway

*Attention Infraction Recorded Incident

# Traffic Light Computer Vision



Model Overview

- Detects traffic lights and their states ( Red, Green, Yellow, Off)
- Provides confidence scoring along with light state

Application

- Traffic light detection and state detection
- Communicate via UART to MCU for logic handling

If Red Light Detected and Vehicle Idle?

- Allow Driver to break attentive driving posture

If Green Light or No Traffic light Present?

- Rely on head posture CV to make decision on driver’s attention

# Radar Detection System

- Utilized for human presence detection and pedestrian collision alarm.
    - All weather conditions
    - Measures motion, distance and speed
    - Near field coverage for crosswalk/ bumper-zone safety.
- Mounting away from the main PCB board with cables for better range detection
- Used to trigger audio signal to drive via piezoelectric buzzer.
    - Working with accelerometer

# Hardware Comparison

## Other Considerations

## Our Decision

| | | | | |
|---|---|---|---|---|
| MPU 6050 Accelerometer | ADXL345<br>- 3 axes, no gyroscope, no magnetometer, no fusion algorithm.<br>- $15<br>- More EMI Resistant | BNO055<br>- 9 axes, Gyroscope, magnetometer, onboard fusion algorithm.<br>- ~$30<br>- Higher accuracy, requires calibration. | BMI160<br>- 6-axis IMU, I2C/SPI, low power, high performance | - Integrated 6-axis sensing<br>- Onboard Digital Motion Processor<br>- Cost Effective<br>- Low power consumption<br>- Wide SW support |
| C4001 mmWAVE Presence Radar Sensor | Seed Studio mmWave Sensor<br>- 24GHz operation, motion detection, I2C/UART<br>- Only 9-15 meters | HIK-RD02/03D<br>- Human tracking, presence, and velocity detection<br>- Can reach 12-16 meters<br>- Not robust for noise, weather, traffic | Other DFRobot Models<br>- Shorter range, less suitable for outdoor large-zone traffic sensing | - Long detection range with 25 meters<br>- Wide beam angle, anti-interface, detection of static/ animate objects<br>- FMCW modulation |
| ESP32 | Arduino UNO<br>- Single core 8-bit, 16MHz<br>- 2KB SRAM<br>- 32KB<br>- No wifi/ bluetooth<br>- 14-22 GPIO<br>- 10 bit, 6-8 channels | STM32<br>- single/ dual core 32/72 bit<br>- 20-96 KB SRAM<br>- **Some** wifi. Bluetooth models<br>- 37-80 GPIO, 10 channels<br>- Variable power options | Raspberry Pi Pico<br>- Dual core 32-bit 133MHz<br>- 264 KB SRAM<br>- No wifi/ bluetooth<br>- 26 GPIO, 3 ADC, 0 DAC<br>- No built in sensors, low power consumption | - Dual core 32-bit 240MHz<br>- 520 KB SRAM, 4-16MB flash<br>- Wifi, bluetooth, 22-34 GPIO<br>- 18 ADC channels, 2 DAC,<br>- I2C,SPI,UART,(ultra-low power consumption) |
| Raspberry Pi 8GB | Orange Pi 5 Pro<br>- 8 core, 2.4GHz CPU<br>- 32GB LPDDR4<br>- Single camera support<br>- Multiple USB port<br>- High Cost | Radxa Rock<br>- Quad/hex core, 4/8/16 GB options<br>- Camera support<br>- Multiple USB ports<br>- Small community/ docs | Nvidia Jetson Nano<br>- Quad-core ARM<br>- 4GB DDR4<br>- NVidia maxwell 128-core<br>- CSI, superior for vision<br>- Cost is highest $$$ | - Quad core 2.4GHz, 8GB RAM DDR4<br>- Multiple cameras supported<br>- Largest SBC ecosystem<br>- Moderate price for full functionality. |

# Hardware Comparison (Cont.)

| | Other Considerations | | | Our Decision |
|---|---|---|---|---|
| Camera | - Raspberry pi camera Module 3<br>- 12MP Sony IMX708<br>- Native CSI interface<br>- Autofocus | - Arducam Mini USB Camera<br>- 1080p, UVC compliant<br>- Small form factor<br>- Good low-light performance | - Logitech C270<br>- Basic 720p webcam<br>- Very cheap and reliable | - UVC 720p USB Camera & Logitech brio 100 1080p |
| Storage Device | - USB 3.0 Flash drive<br>- Faster sustained write speed<br>- Easier hot-swap<br>- Works directly on Pi 5 | - External SSD<br>- Very high performance<br>- Useful for constant video recording<br>- Higher power consumption | - 32Gb microSD<br>- Lower capacity<br>- Designed for continuous recording<br>- Great longevity for logging applications | - Raspberry pi 128Gb micro SD |
| Speaker | - Magnetic buzzer<br>- Louder at low frequency<br>- Simple drive requirements<br>- Limited tonal range | - 8ohm dynamic speaker<br>- Higher acoustic quality<br>- Allows voice/ alert output<br>- Needs amplifier & more power | - Active Pizeo Buzzer<br>- Self-driven, requires no DAC output<br>- One tone, very loud | - Piezoelectric speaker |

| Component | Supply Voltage | Current | Wattage |
| --- | --- | --- | --- |
| MPU6050 | 3.3V | ~3.5mA | 0.012 W |
| c4001 mmWave Sensor | 5V | 90-100mA | 0.50 W |
| ESP32 | 3.3V | 80-240mA | 0.26-0.8W |
| Raspberry Pi | 5V | 2.5-4A | 12.5-20W |
| Cameras | 5V | 250-400mA | 1.25-2W |
| Piezoelectric speaker (1) | 3.3V | 10mA | 0.033W |
| Piezoelectric speaker (1) | 12V | 20mA | 0.24W |
| | | Total | ~21.8W |

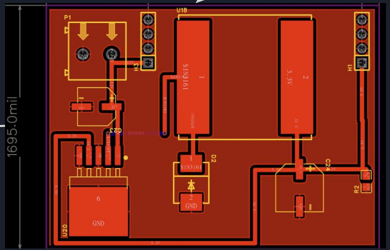
# Power Source Decision

| Battery | AC Source | DC Car Source |
|---|---|---|
| - Not sustainable<br>- Lack of reliability/ robustness<br>- Can lose charge mid journey | - Entire system is DC based<br>- Rapid polarity reversals<br>- Components see negative voltage<br>- Switching converters won’t operate at all | - Stable values, GNd reference<br>- No “life” operates with vehicle start<br>- Improves robustness and effectiveness |

# Allocating Voltage

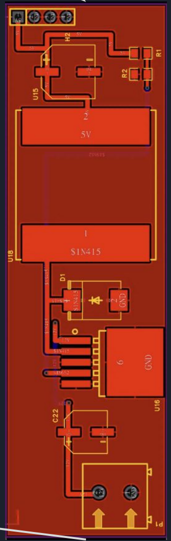
12V

- Voltages must be distributed to avoid component failure
- 3.3V components
  - MPU6050
  - ESP32
  - Piezoelectric speaker
- 5V Components
  - C4001 mmWave Radar
  - Raspberry Pi
  - Cameras
- Done through regulator breakout boards



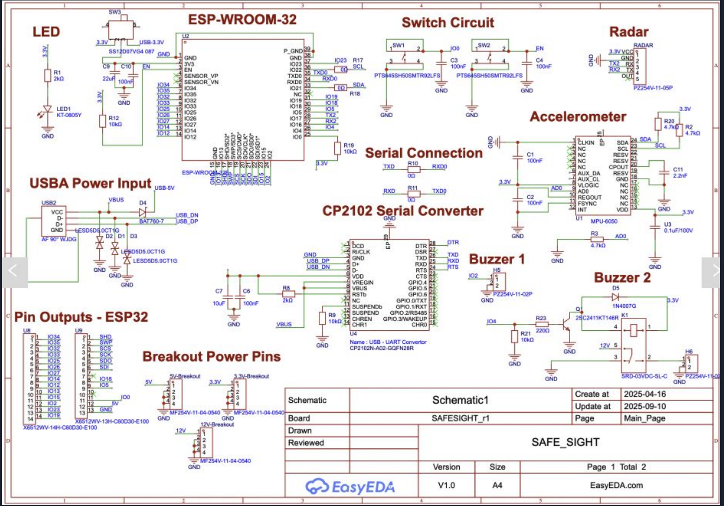
3.3V

5V

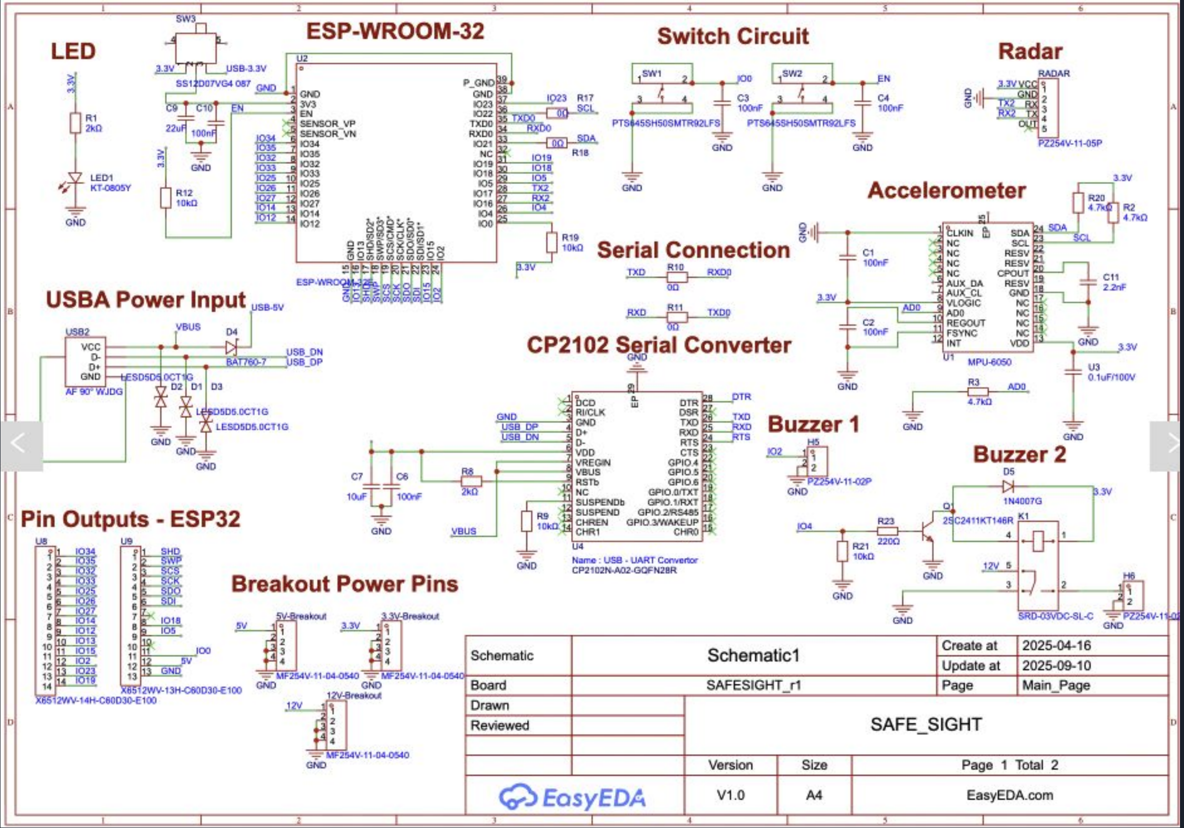


# Main Board PCB

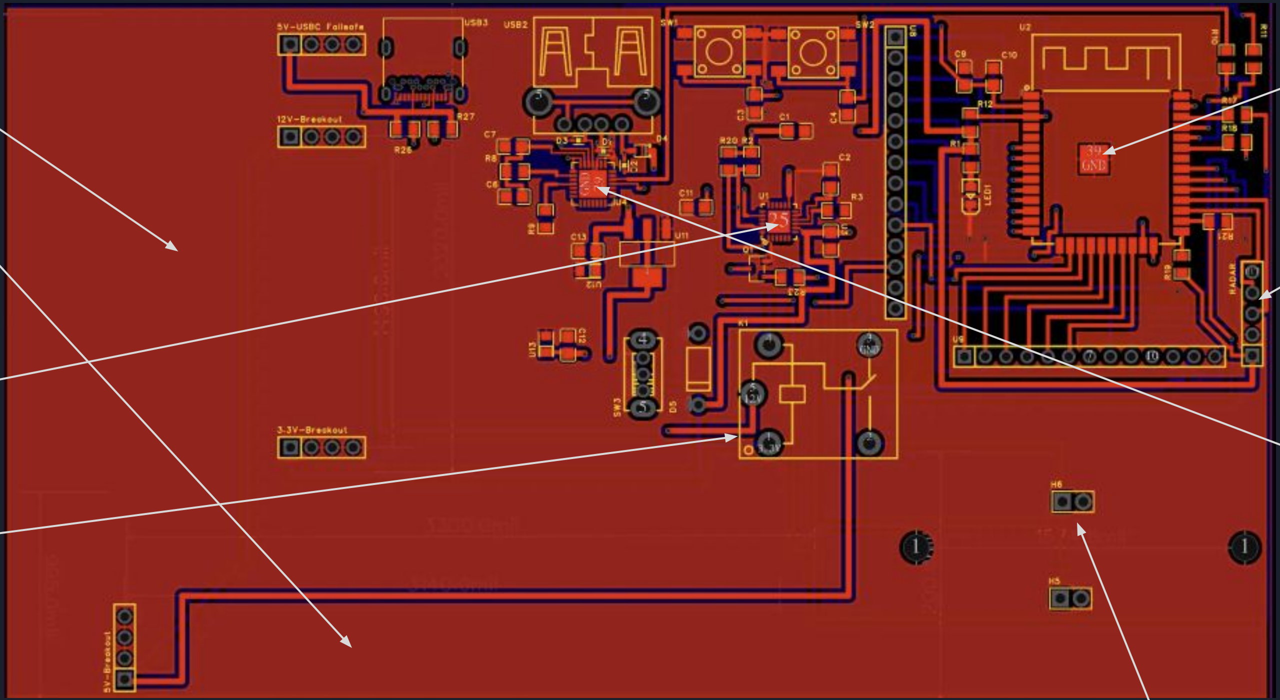
- Responsible for the communication and power delivery of all boards and components in this system.
- DC supply clines are thicker than the rest of the clines in order to reduce current density, minimize voltage drop, and increase power integrity.
- Power LED for visual check to ensure that the board is receiving ample power for each component.
- Expanded to fit the plug-in power regulator boards, radar, and raspberry pi.
- Ensured signal integrity for all traces with a copper pour on both the top and bottom layers.
    - This acts as a shield to the traces, confining electric fields and reducing emissions.



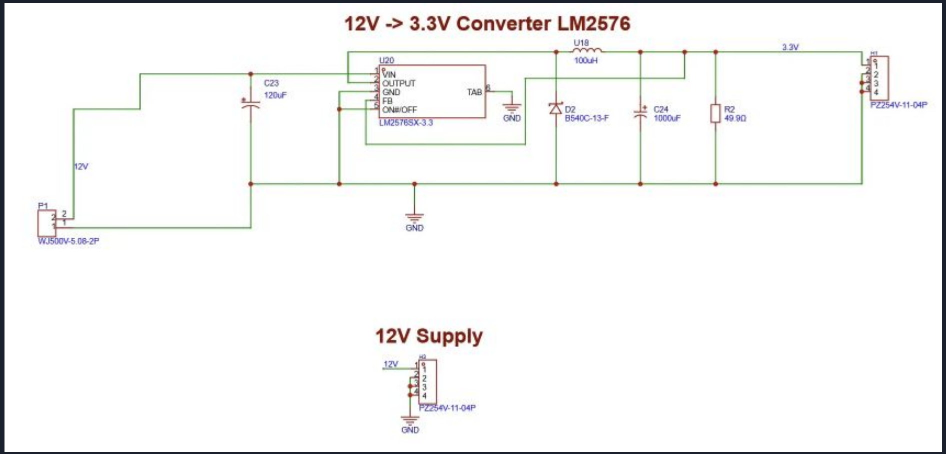
# Main Board Schematic

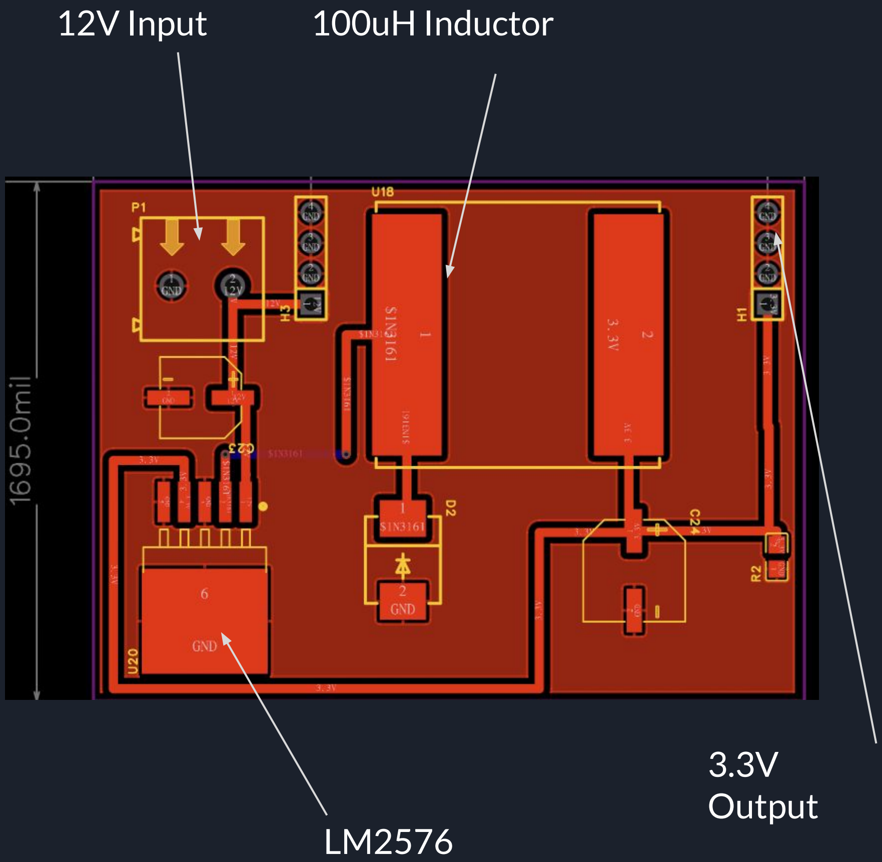


# Main Board PCB Layout

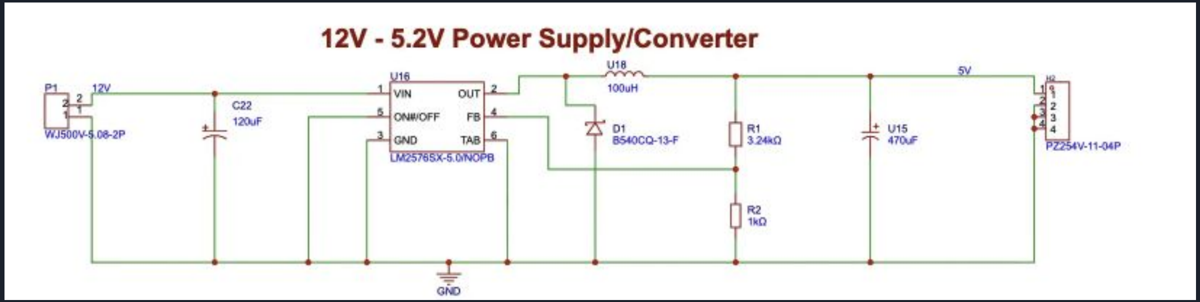


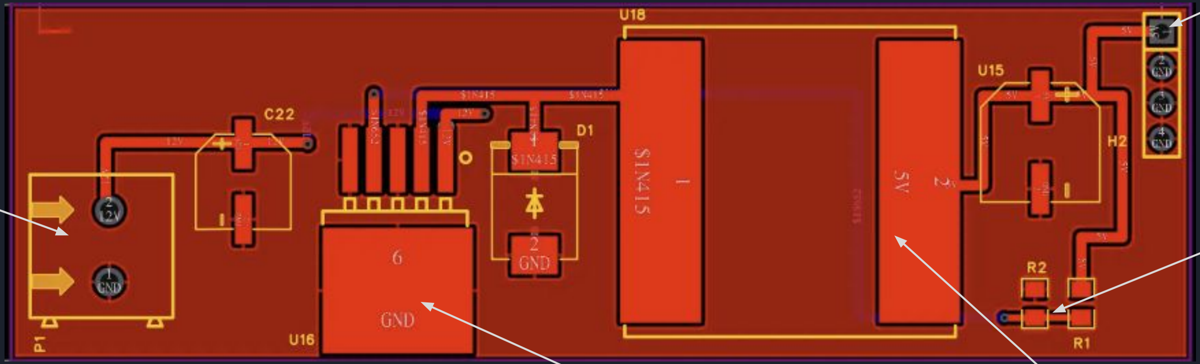
# 3.3 V Converter



# 5.2V Converter




12V Input

5V Output

Voltage Divider

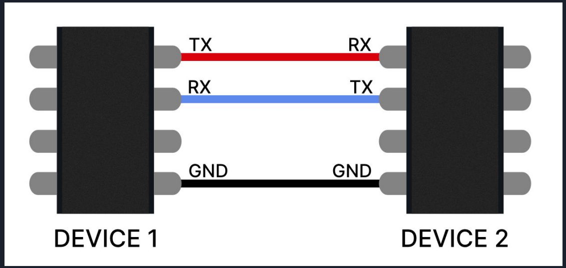
LM25765

100uH Inductor

# Communication Protocols

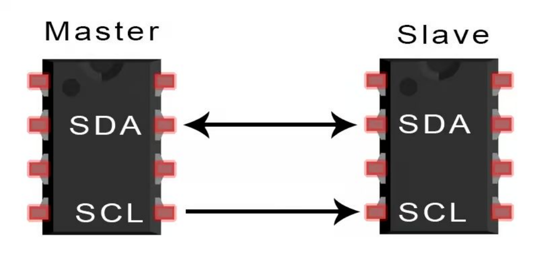
## UART

- Raspberry Pi
  - Sends face posture and traffic status bytes along bus
  - Receives bytes to document and save current photo
- ESP 32
  - Sends bytes to Pi and also receives them
- C4001 Radar Sensor
  - Sends and receives data to and from the ESP32




## I2C

- MPU-6050 Accelerometer
  - ESP32 -> Master
  - MPU-6050 -> Slave



# Software Comparison: Embedded Programming

| Technology | Pros | Cons |
| --- | --- | --- |
| **C/C++** | - High performance and widely used in embedded systems | - Steep learning curve<br>- Manual memory management<br>- Error-prone and verbose |
| **Assembly** | - Super fast signal-processing<br>- Full control over hardware | - Very complex and error prone<br>- Not familiar with it |
| **Java** | - Strong OOP support<br>- Cross platform | - Verbose syntax<br>- Slower than C/C++ |
| **Python** | - Easy to learn<br>- Massive library support | - Slower performance<br>- Limited for low-level hardware interaction |

**Our Choice: Picked C/C++ due to familiarity and reliability with an ESP32.**

# Software Comparison: App Development (Frontend)

| Technology | Pros | Cons |
|---|---|---|
| **React Native** | - Cross-platform<br>- Huge community for resources<br>- Fast reload speed | - Performance lower than SwiftUI<br>- Heavy reliance on third-party packages |
| **Swift/SwiftUI** | - Modern and safe language<br>- Great for iOS/macOS development<br>- Fast and swift UI design | - Limited to Apple ecosystems<br>- Small community and less resources available |
| **Android Native** | - Best performance on Android<br>- Modern, declarative language<br>- Fast previews, live edit | - Android only<br>- Missing some edge cases<br>- Huge learning curve |
| **Flutter** | - Cross platform<br>- Smooth performance<br>- Reloads super fast | - Larger app size<br>- Requires learning Dart<br>- Large overhead |

**Our Choice: We chose Swift/SwiftUI due to familiarity and willingness to learn iOS Development. SwiftUI makes UI design easy with its built in libraries.**

# Software Comparison: App Development (Backend)

| Technology | Pros | Cons |
|---|---|---|
| **Node.js** | - Fast for I/O<br>- Huge ecosystem<br>- Great for real-time systems (WebSockets) | - Single-threaded<br>- Lots of third-party dependencies<br>- Not ideal for data heavy apps |
| **Python/Flask API** | - Easiest and fastest the write<br>- Native support for Raspberry Pi<br>- Large community, many libraries | - Slower than other backend tech<br>- Not ideal for high concurrency<br>- Can struggle under heavy traffic |
| **C#/ .NET Core** | - Very fast and scalable<br>- Cross-platform<br>- Very reliable | - Requires more setup<br>- Learning curve with C#<br>- Smaller community |
| **Rust** | - Highest performance<br>- Extremely safe | - Steep learning curve<br>- Small ecosystem for resources |

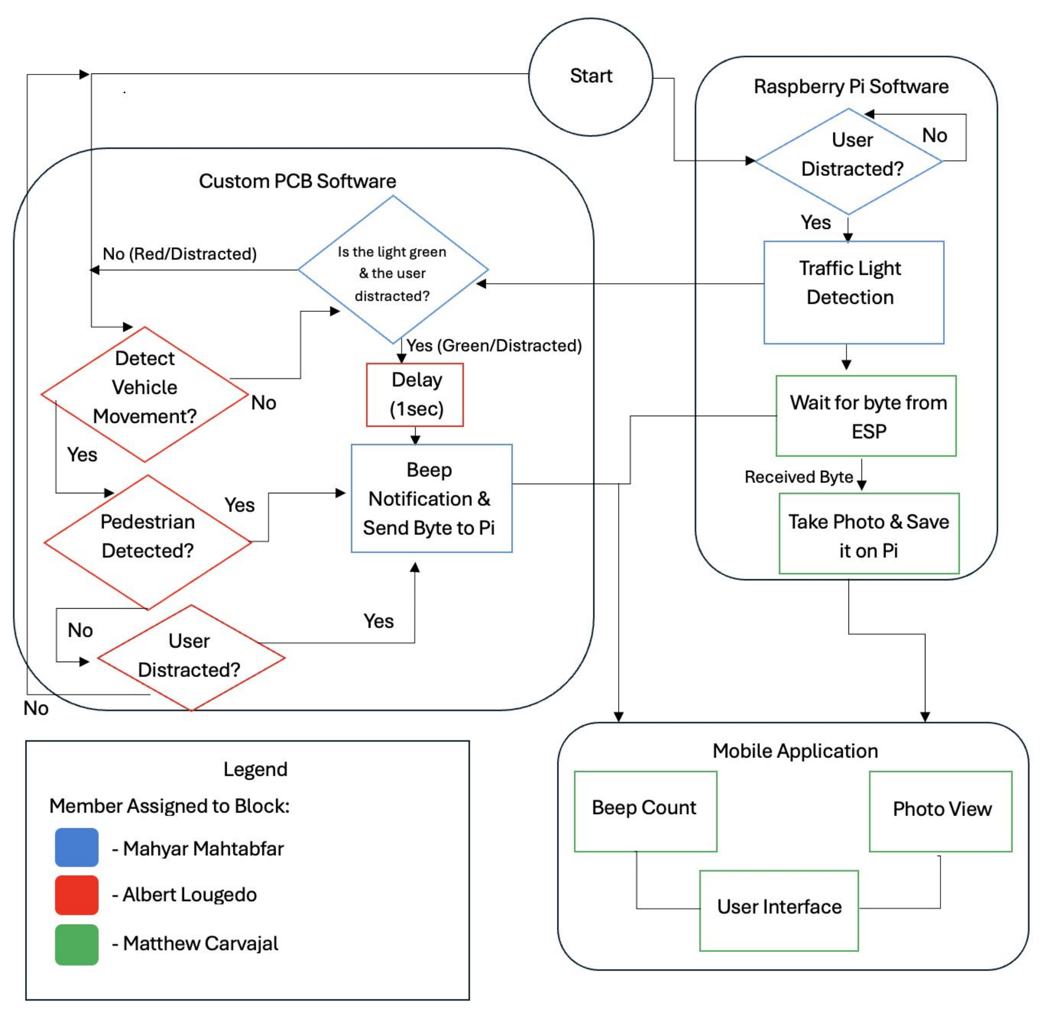
Our Choice: We chose **Python/Flask API** **for its native support on the Raspberry Pi. It is also the easiest to write out of all of the other technologies that we have considering that it is Python.**

# Computer Vision Selection

| Framework | Capabilities | Ease of Use | Speed | Accuracy | Best Use Cases |
|---|---|---|---|---|---|
| OpenCV | Comprehensive library for image/video processing, object detection, tracking, facial recognition, and machine learning integration | High (simple APIs) | Fast | High | Real-time applications involving color/shape labeling |
| TensorFlow/TensorFlow Lite | Advanced deep learning framework supporting neural networks, TPUs, dataflow graphs, and distributed computing. TensorFlow Lite enables deployment on mobile/edge devices | Moderate | Moderate | Very High | Deep-learning based object detection, facial sentiment analysis |
| Keras | High-level API for building neural networks with TensorFlow backend, simplifies model creation, training, and deployment. Supports integration with datasets like NumPy arrays and Pandas DataFrames. | Moderate | Moderate | Very High | Custom object detection tasks that require high precision |
| Scikit-Image | Filtering, segmentation, feature extraction, morphological operations, and geometric transformations. | High | Moderate | Moderate | Simple image processing tasks including basic object/face detection |

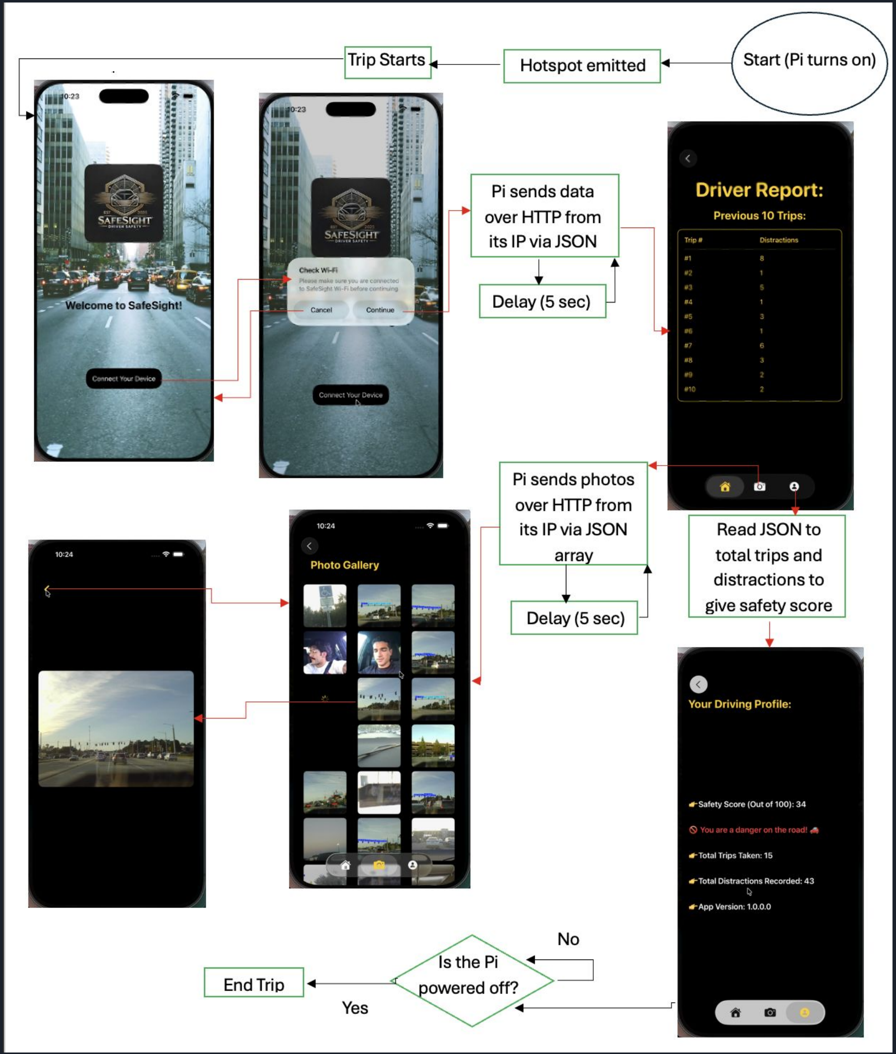
# Software Diagram 1

- SafeSight will take many factors into account when determining distracted driving
- Factors will include (shown in chart):
  - Driver head angle
  - Traffic light color
  - Speed of car
  - Pedestrian detected



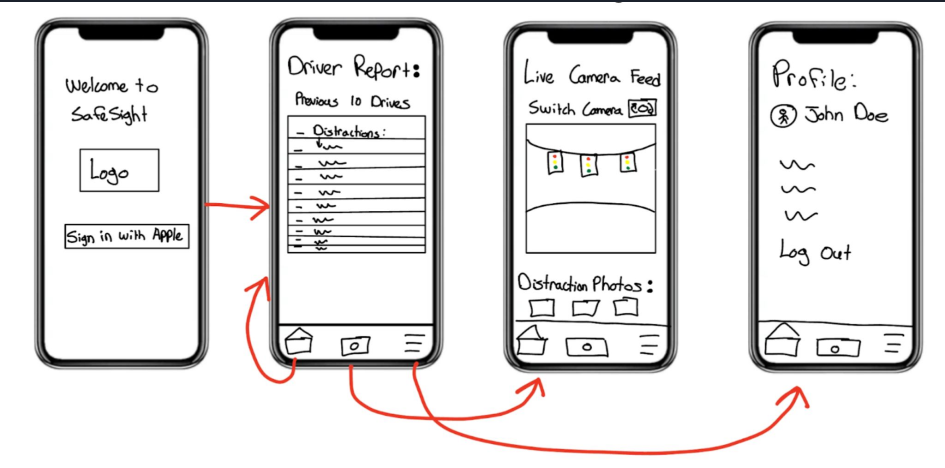
# Software Diagram 2: App

- The application will have 4 views:
  - Welcome view
  - Home view
  - Photo gallery
  - Profile view
- Pi creates a local hotspot to send information



# App Development: Idea

- **Original idea:**
  - Have 4 Views:
    - Have the user login using Apple ID at launch
    - Driver Report View
    - Live Camera & Gallery View
    - Profile Page View
- **Problem:**
  - Login with Apple ID feature requires Apple Developer Account ($100/yr)
- **Solution:**
  - Have the app tell the user to connect the SafeSight hotspot network upon startup instead of logging in

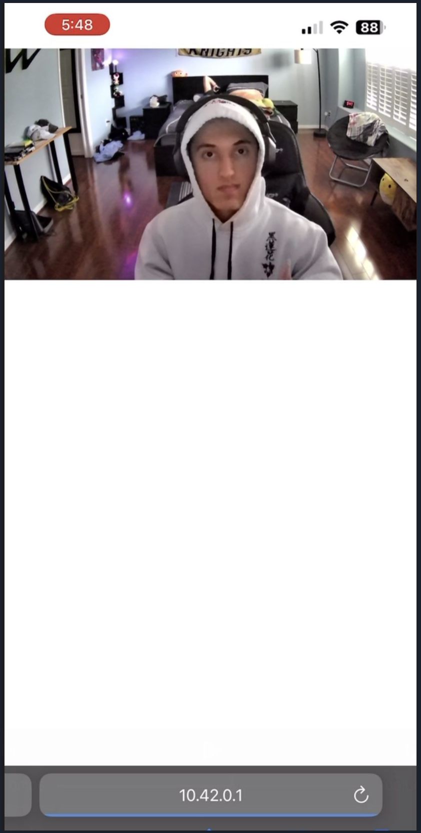


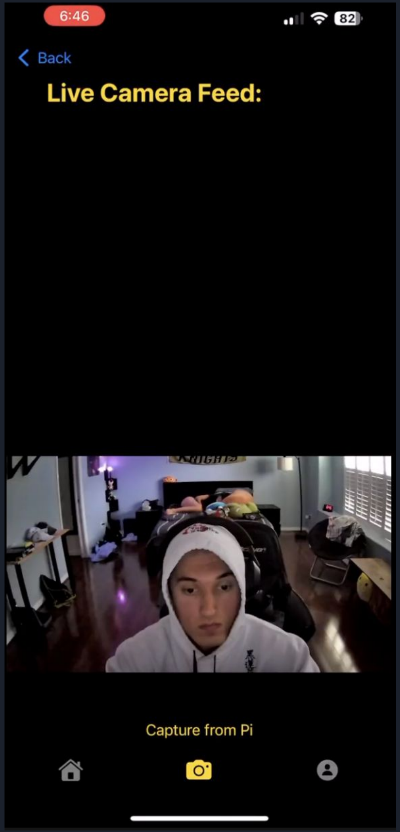
# The Problem at Hand

- **Issue:**
  - Needed to figure out a way to send the pictures of distracted drivers over from the Pi to our iOS app
  - **Came across one issue:** iOS applications do not allow for bluetooth file transfer
- **Solution:**
  - Have the Pi put out a signal, act as a host with an IP address
  - Connect to that signal on the iPhone
  - Have the Pi point to a folder on the system
  - Send the Pictures over HTTP to the iPhone app and display them
  - Similar to a INSTA 360 or GOPRO camera

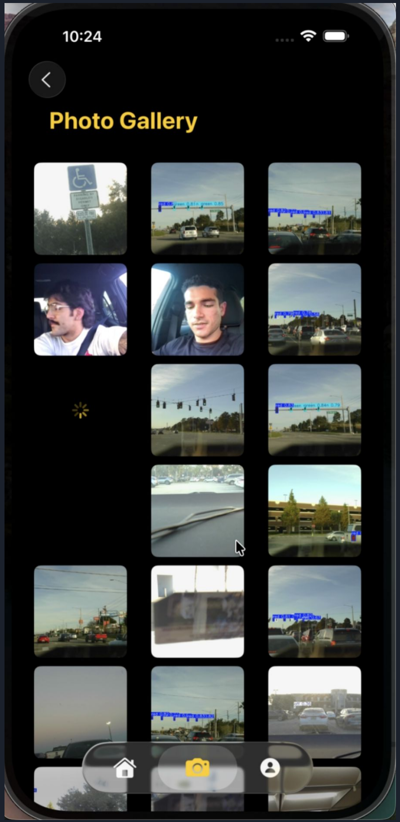
# Solution: Host Over HTTP

- Pi connection established via Pi's Hotspot IP address
- App accepts pictures in JSON format



# Displaying Photos From Pi




- The photos of the distracted driving are shown from the Pi to the app in a gallery style
- Allows the user to see what they looked like when caught driving distracted
- Stores front picture for surroundings while distracted
- Essential for insurance companies and other companies like Uber, Lyft, etc.

# Driver Posture Computer Vision

Driver's Posture CV Model

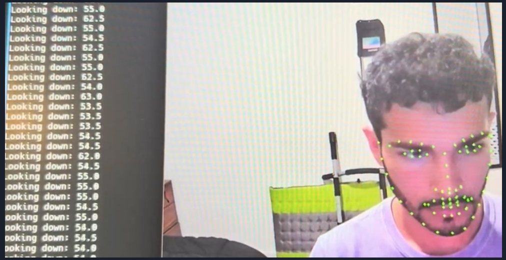
- Utilized Teachable Machine that uses tensorflow to create a model and export it as a Keras file to be used in a python script.

Problem

- Models from teachable machine were in need of much more training in order to obtain a working algorithm that was universal to all facial structures.
- Testing of Keras file generated from teachable machine. Was so selective in its training model that a simple change in color of a T-shirt would disrupt its operation of face posture.
- Decided that we needed much more raw data collection and it was unreliable since we did not have the ability to properly teach the computer vision model in the given time frame.

Solution

- OpenCV face detection and face angle library proved to
- Proposed solution was to use OpenCV face angle detection that would pickup on any person's facial structure and assume their head posture relative to the road by calculating angles in real-time based on their face posture.




Face Angle CV

# Functionality Testing - C4001 Radar

# Test Results

| Trial | Outcome |
| --- | --- |
| 1 | PASS |
| 2 | PASS |
| 3 | FAIL |
| 4 | PASS |
| 5 | PASS |
| 6 | PASS |
| 7 | PASS |
| 8 | PASS |
| 9 | PASS |
| 10 | FAIL |
| Average | 80% PASS |

# Functionality Testing - Face Angle CV

# Test Results

| Trial | Outcome |
|---|---|
| 1 | Pass |
| 2 | Pass |
| 3 | Fail |
| 4 | Pass |
| 5 | Pass |
| 6 | Pass |
| 7 | Pass |
| 8 | Pass |
| 9 | Pass |
| 10 | Pass |
| Average | 90% Accuracy |

# Functionality Testing - Traffic Light CV





- We tested 10 different traffic lights in our area to see the accuracy of our CV
- Each traffic light has a color label as well as the confidence level next to it (0.00 - 1.00)

# Test Results

| Trial | Outcome |
|---|---|
| 1 | Fail (missed green arrow) |
| 2 | Pass |
| 3 | Pass |
| 4 | Pass |
| 5 | Pass |
| 6 | Pass |
| 7 | Pass |
| 8 | Fail (missed 2/4 red lights) |
| 9 | Pass |
| 10 | Pass |
| Average | 80% |

# Functionality Testing - MPU-6050 Accelerometer

# Test Results Accelerometer

| Trial | Outcome (Y Axis) |
|---|---|
| 1 | PASS |
| 2 | PASS |
| 3 | PASS |
| 4 | FAIL |
| 5 | PASS |
| 6 | PASS |
| 7 | PASS |
| 8 | PASS |
| 9 | PASS |
| 10 | PASS |
| Average | PASS |

*Consistency test based on accelerometer readings of acceleration/ deceleration

# Major Component Integration

# BOM

| Selection | Cost |
|---|---|
| Raspberry pi 5 8GB RAM + Cooler Kit | $140 |
| Accelerometer Dev Board (Testing) | $12.95 |
| C4001 Radar | $27.90 |
| JLCPCB Order | $159.58 |
| Digikey + LCSC Component Order | $92.37 |
| Total Cost | $432.80 |

| Line Item | Ordered | Available Qty | Backordered Qty | Item Number/ Description | Unit Price USD $ | Amount USD $ |
|---|---|---|---|---|---|---|
| 1 | 3 | 3 | 0 | **PART:** 565-EMVY6R3ARA102MHA0GCT-ND **DESC:** CAP ALUM 1000UF 20% 6.3V SMD<br>**MFG** : UNITED CHEMI-CON (VA) / EMVY6R3ARA102MHA0G<br>**COO** : JAPAN **ECCN:** EAR99 **HTSUS:** 8532.22.0020<br>ROHS3 COMP REACH UNAFFECTED 21-JAN-2025<br>**Mercury:** Cert on File. For more information contact Environmental@DigiKey.com | 0.77000 | 2.31 |

| | |
|---|---|
| Sales Amount | 2.31 |
| Shipping charges applied | 4.99 |
| Sales Tax | 0.15 |
| Total | 7.45 |
| | USD $ |

**Summary**

| | |
|---|---|
| **Merchandise Total** | **$76.33** |
| **Shipping Fee** | **$18.19** |
| **Total Discounts** | **- $9.60** |
| Merchandise Discount | - $1.60 |
| Shipping Discount | - $8.00 |
| **Total** | **$84.92** |
| **Paid Amount** | **$84.92** |

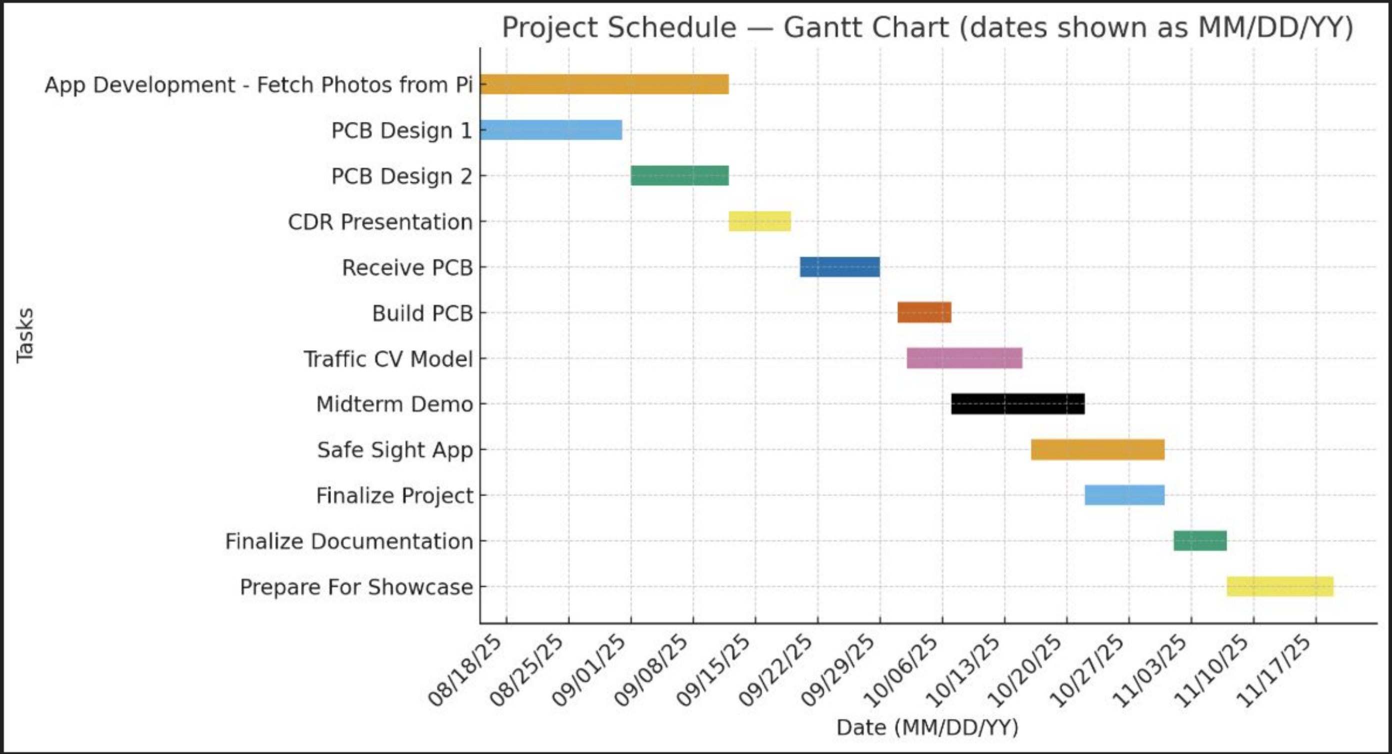
| | Product | File Name | Order Number | QTY | Unit Price | Ext.Price |
|---|---|---|---|---|---|---|
| 1 | Solder Paste Stencil | SAFE_SIGHT_PCB7_2 0250912144136 | SO12509126559 | 1 | USD $3.00 | USD $3.00 |
| 2 | Solder Paste Stencil | SAFE_SIGHT_PCB8_2 0250912143444 | SO12509126558 | 1 | USD $7.00 | USD $7.00 |
| 3 | Solder Paste Stencil | SAFE_SIGHT_PCB4_2 0250912144313 | SO12509126561 | 1 | USD $3.00 | USD $3.00 |
| 4 | 2-layer Bare Rigid Printed circuit board | SAFE_SIGHT_PCB4_2 0250912144313 | Y4 | 5 | USD $0.80 | USD $4.00 |
| 5 | 2-layer Bare Rigid Printed circuit board | SAFE_SIGHT_PCB8_2 0250912143444 | Y2 | 5 | USD $11.85 | USD $59.25 |
| 6 | 2-layer Bare Rigid Printed circuit board | SAFE_SIGHT_PCB7_2 0250912144136 | Y3 | 5 | USD $0.42 | USD $2.10 |

| | |
|---|---|
| Merchandise Total: | USD $78.35 |
| Shipping: | USD $34.31 |
| Shipping Discount: | -USD $10.00 |
| Subtotal: | USD $102.66 |
| States Sales&Use Tax: | USD $6.67 |
| Import Taxes : | USD $50.25 |
| Grand Total: | USD $159.58 |

Tax rates vary for different products; please refer to the tax details .

*USB Cameras and ESP-32 Dev board were pre-owned contributions to the project and are not included in the BOM

# Project Gantt Chart



# Challenges Faced

- Accelerometer integration
  - -X axis and Y axis orientation depends on system placement
- CV confidence for face and light recognition
  - Adjusted PASS/FAIL values
- Raspberry Pi bit transfer to ESP32
  - Only sent bit on a change in environment
- Face Angle CV Model Selection
  - Used open source face detection and angle model in OpenCV instead of teaching our own AI model using teachable machine and a Keras file.

# Future Challenges and Proposed Solutions

1. Accelerometer calibration
    a. Solution: Gather accelerometer data based on different device placement and height from the ground. Create preset profiles that the user can select according to their device placement relative to the ground.
2. Improved Radar Component
    a. Solution: Higher grade radar component will be more efficient in detecting animate and inanimate objects. Can also be sourced outside of the vehicle.
3. Face Angle Computer Vision calibration
    a. Solution: Tweak for all different camera angles so it can work wherever the camera is placed
4. Application: Being able to download pictures from the app to the phone's camera roll
    a. Solution: Make a share function in the app that can share info and allow for JSON data download into iOS camera roll format

# Work Distribution


| Name: | PCB | Firmware/Logic | CV | Application: |
|---|---|---|---|---|
| Mahyar Mahtabfar | X | X | X | |
| Albert Lougedo | X | X | X | |
| Matthew Carvajal | | X | X | X |