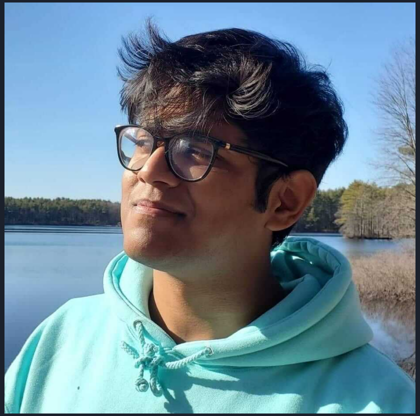
SafeX

Helping Riders Stay Safe

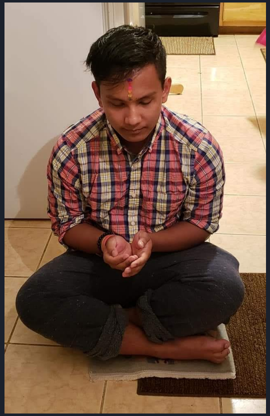
The Team



Dhruv Vikram Krishna
Computer Engineer

Jeffrey Wang
Computer Engineer

Paige Wadas
Electrical Engineering



Siddhanta Shrestha
Computer Engineer

Team Responsibilities

Dhruv:

- Blindspot Detection Software Lead:
  - Responsible for training machine learning algorithm to detect vehicles in motion and classify them
  - Developing algorithms for object detection and classification for hazards in blindspots

Sidd:

- Team Coordinator:
  - Coordinating tasks and direction to ensure the project is developing cohesively.
- Fall Detection Software Lead:
  - Responsible for developing robust Fall Detection algorithm
- Subsystem Coordinator:
  - Ensuring the functionality of the whole system by conjoining the subsystems cohesively

Jeff:

- Budget Management:
  - Ensuring the SDP budget is spent appropriately
- Bluetooth Lead:
  - Communication for the helmet and motorcycle subsystems
- User Interface Lead:
  - Developing mobile app, for communicating with the system via Bluetooth
  - Incorporating GPS data and cellular module into crash protocol

Paige:

- Team Communication Coordinator:
  - Coordinating team members and organizing tasks to ensure the project is progressing efficiently via deadlines.
- Hardware and PCB Design Lead:
  - Finite hardware research for system design
  - Design the hardware system and ensure the components can interface with each other successfully
  - Breadboard design using breakout boards and PCB design

Quick Review of SafeX

- There are alot of safety features available for cars that are not accessible to motorcycles.
  - Additionally, there are also missing features that should be available to motorcyclists.
- We hope to bring some of those safety features to motorcycles in order to ensure the safety of riders.

Project Goal

To create a two part device that is attachable to a motorcycle and helmet. The system should notify riders of oncoming vehicles within their blindspots to increase rider safety, and alert emergency services if the rider gets into a crash.

MDR Deliverables

Initial Deliverables

- Blindspot Subsystem
    - Detect moving vehicles by classifying objects within the blindspots
    - Transmit object detection data from the motorcycle to the helmet over Bluetooth
    - Configure lighting system to indicate direction and proximity using the detection data
- Fall Detection Subsystem
    - Compare both data transmitted from the motorcycle's gyroscope/accelerometer with the helmet's accelerometer for fall/accident
    - Implement fall detection protocol by testing visual alert notification with cut off button in the helmet

Updated Deliverables

- Blindspot Detection System
  - Detect and classify vehicles within incoming video
  - Generate blindspot region within the video frame and flag vehicles that enter this region
  - Provide visual notification to indicate the direction of incoming vehicle relative to the riders blindspots.
- Fall Detection System
  - Determine if the crash protocol needs to be initialized by comparing the data transmitted from the motorcycle's accelerometer with the helmet's accelerometer
  - Implement fall detection protocol and visual alert when crash is detected
- Mobile Interface
  - Prototype interface with rudimentary login/signup, verification and user profile pages

Challenges Faced

Blindspot Detection

- Low framerates on OpenCV & Tensorflow Lite on Raspberry Pi 3
  - Switched to NVIDIA Jetson Nano, with the 128-core GPU giving an increased speed on ML inference; lower than required frame rate due to OpenCV bottleneck
    - Switching to YOLOV5s and PyTorch for better performance and frame rate
- Raspberry Pi HQ Camera undetected by Jetson
  - Required using 3rd party library to read camera input
- Tensorflow crashing while loading YOLOV3 model into GPU memory
  - Unable to fix this error for MDR; demoing using output videos from pre-recorded test footage
- No bluetooth module on NVIDIA Jetson Nano, caused a delay in development of the bluetooth and hardware design of the Jetson so Raspberry Pi 3 A+ became the substitute
  - Purchased USB Bluetooth adaptor to allow NVIDIA Jetson tp communicate via bluetooth

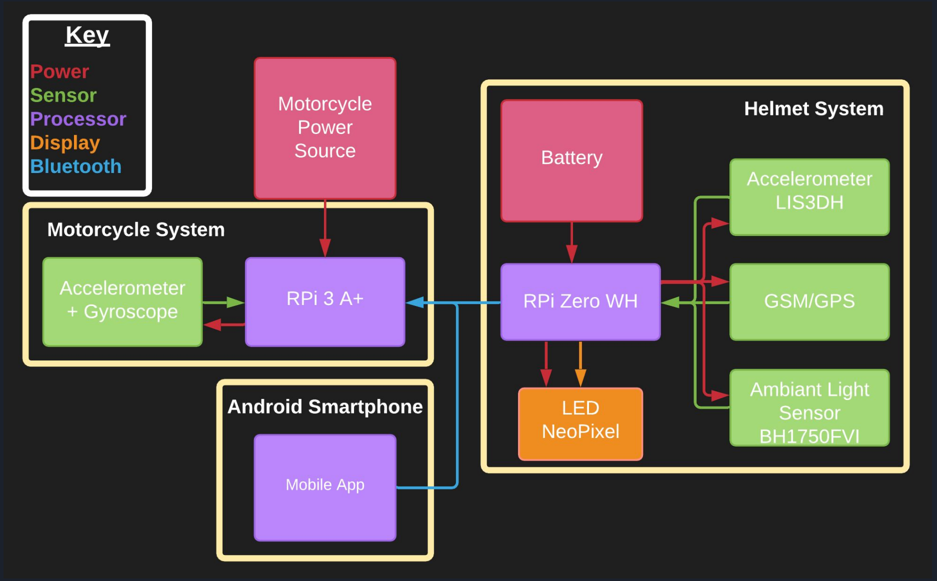
Fall Detection

- Issues with bluetooth disconnection between host and client when running into errors
  - Added delays to ensure that motorcycle's accelerometer data isn't being sent within the same clock cycle
    - Either storing the data being received on the helmet so that it can be used sequentially, implementing real time parsing of data to differentiate between multiple data points, or using bash to run multiple scripts concurrently in the background
  - Tampered with error catching to ensure program doesn't crash
    - Implementing better error catching to efficiently ensure program runs smoothly and does not crash at every edge case

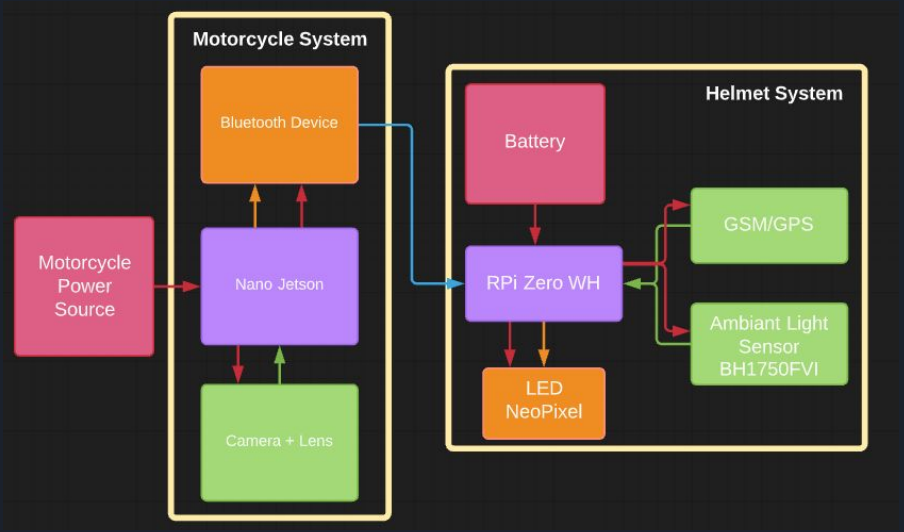
System Design

Hardware Flowchart

Fall Detection



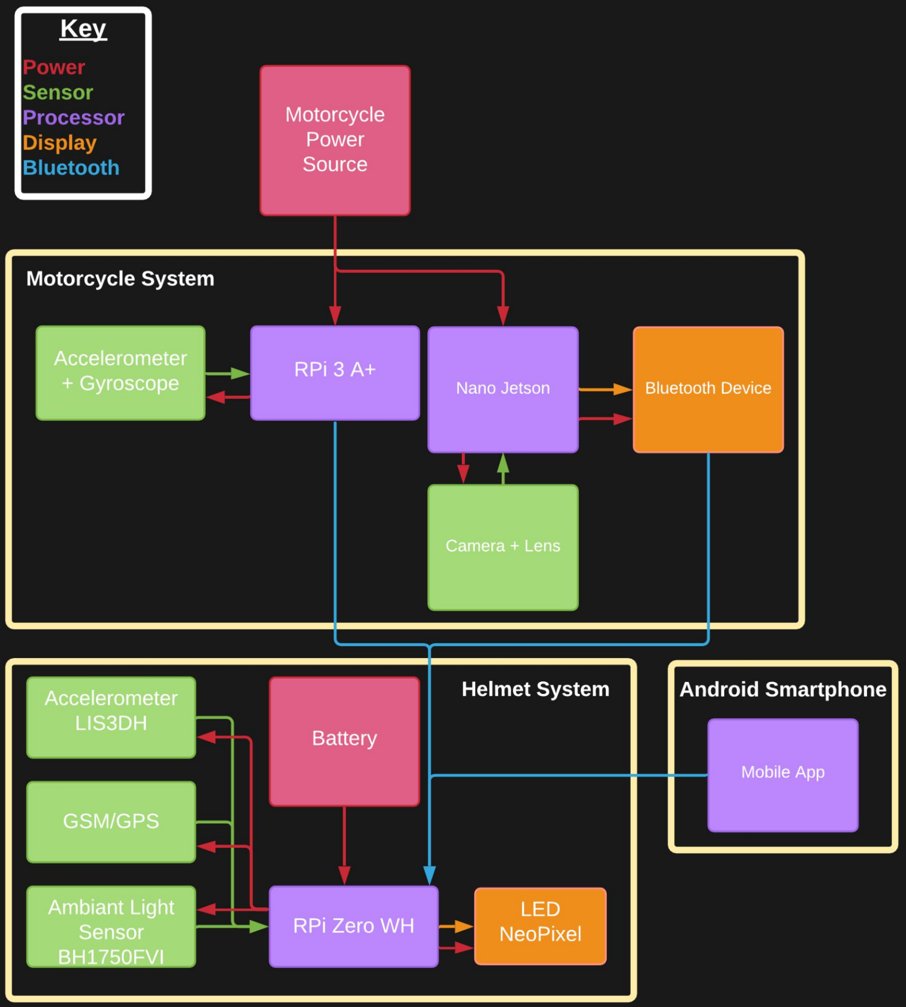
Blindspot Detection



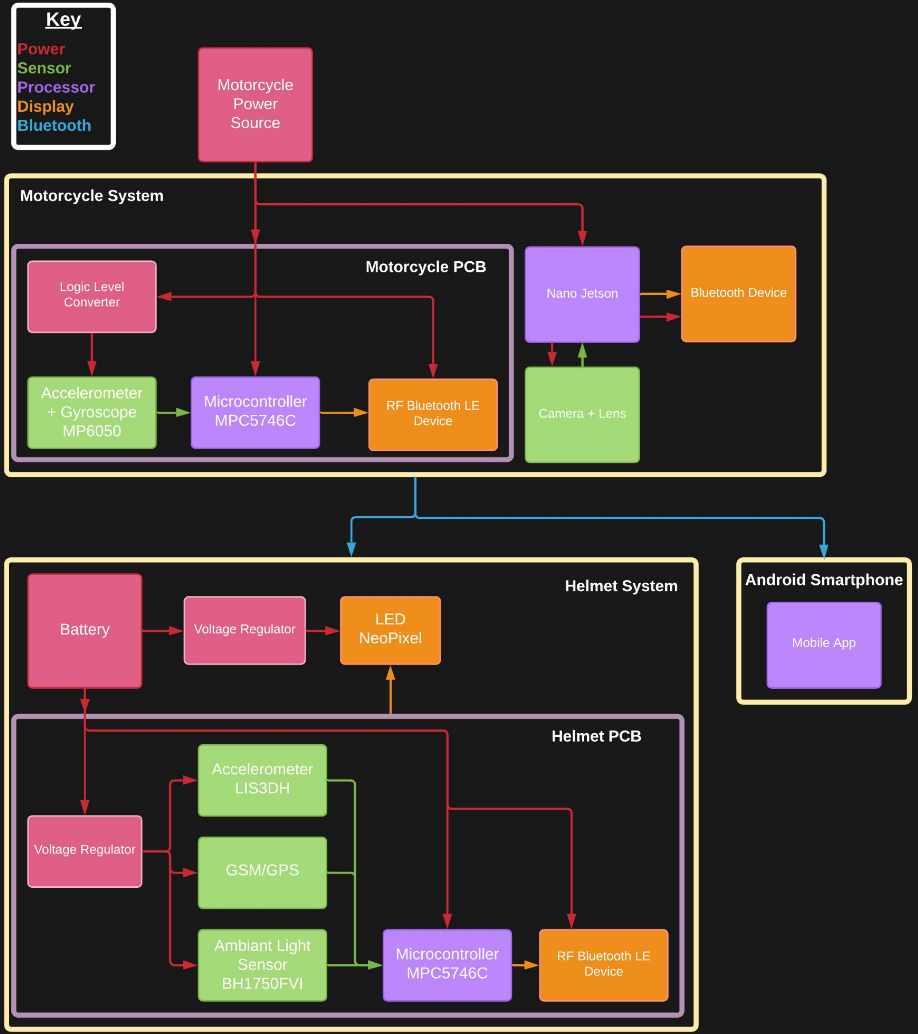
Hardware Used

- Helmet:
  - Raspberry Pi Zero W H
  - Adafruit LIS3DH Triple-Axis Accelerometer IC
  - Adafruit NeoPixel LED Side Light Strip
  - microSDHC Card 32GB
  - Battery power bank
    - Lithium Polymer, output 5V/3.4A(max) → 12Wh Capacity
- Motorcycle
  - Raspberry Pi 3 A+
  - NVIDIA Jetson Nano
  - Adafruit MPU6050 6-DoF Accel and Gyro Sensor - STEMMA QT Qwiic
  - 6mm 3MP Wide Angle Lens for Raspberry Pi HQ Camera
  - Raspberry Pi high quality camera 12MP

Updated Hardware Diagram

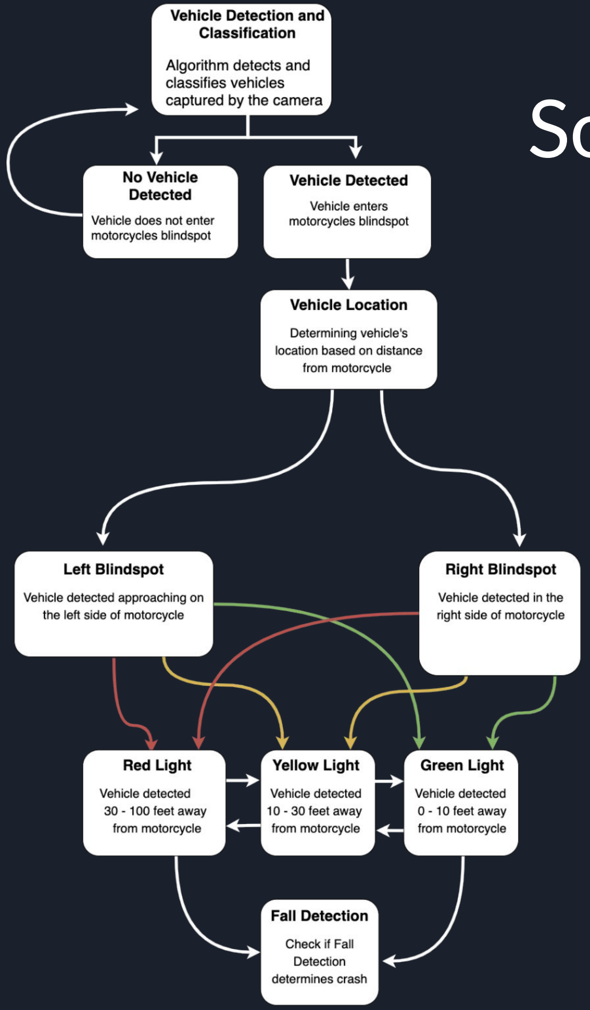


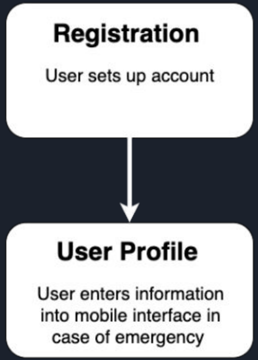
PCB Design

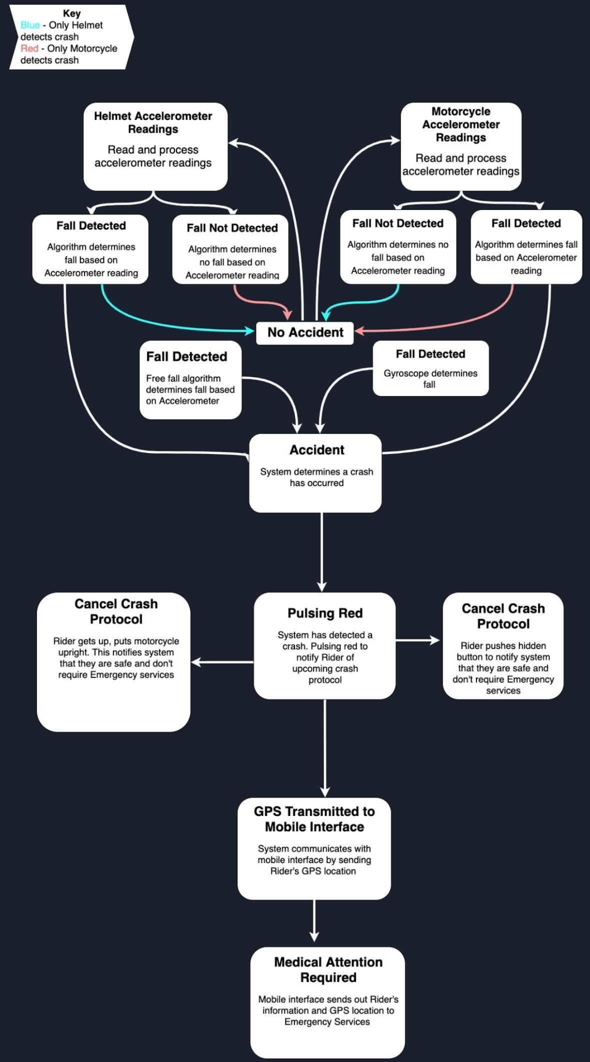


Key
Power
Sensor
Processor
Display
Bluetooth
Motorcycle Power Source
Motorcycle System
Motorcycle PCB
Logic Level Converter
Accelerometer + Gyroscope MP6050
Microcontroller MPC5746C
RF Bluetooth LE Device
Nano Jetson
Bluetooth Device
Camera + Lens
Helmet System
Battery
Voltage Regulator
LED NeoPixel
Helmet PCB
Voltage Regulator
Accelerometer LIS3DH
GSM/GPS
Ambiant Light Sensor BH1750FVI
Microcontroller MPC5746C
RF Bluetooth LE Device
Android Smartphone
Mobile App

Software Flowchart

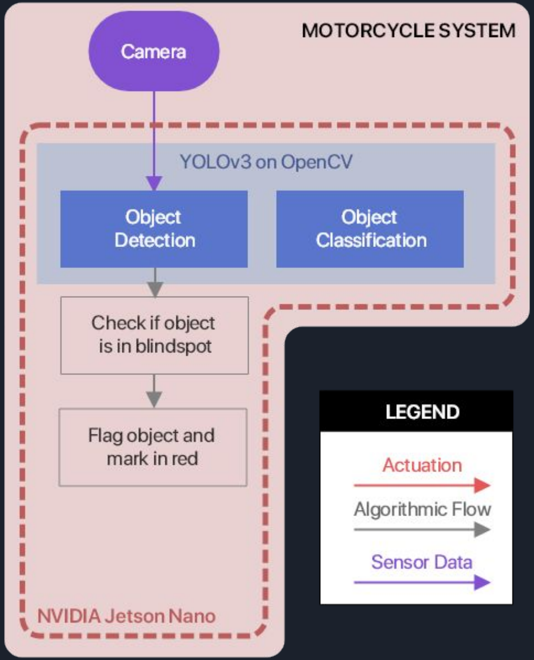






Blindspot Flowchart

Flowchart



Blindspot Region Mask

Power Considerations

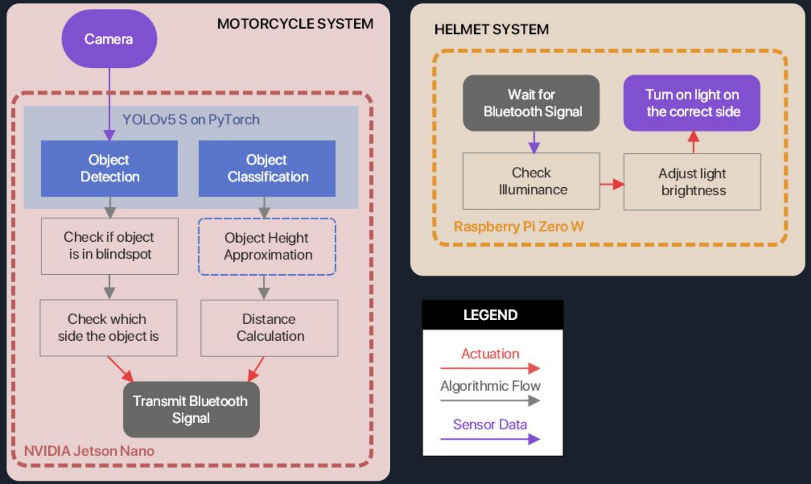
- Helmet System Power Consumptions
  - MDR: Total ~ 23W/Hr
    - Raspberry Pi Zero W H: 5V/2A → 10W
    - Accelerometer: 3.3V/75uA → 0.2475mW
    - NeoPixel LED Strip: 5V/2.5A → 12.5W
  - Post MDR: Total ~ 17W/Hr
    - Microcontroller: 5V/246mA → 1.23W
    - Light Sensor: 3.3V/7mA → 23.1mW
    - Accelerometer: 3.3V/75uA → 0.2475mW
    - NeoPixel: 5V/2.5A → 12.5W
    - Level Shifter(max) → 0.36W
    - GSM/GPS 4V/0.6A → 2.4W
    - Bluetooth 5V/10.7mA → 53.5mW
- LiFePO4 battery: output 5V/2.5A with minimum of 50Wh Capacity

- Motorcycle System Power Considerations
  - MDR: Total ~ 33W/Hr
    - Nano Jetson: 5V/2-4A → 10-20W
    - Raspberry Pi 3 A+: 5V/2.5A → 12.5W
    - Accelerometer/Gyroscope: 3.3V/1.2mA → 3.977W
  - Post MDR: Total ~ 23W/Hr
    - Nano Jetson: 5V/2-4A → 10-20W
    - Microcontroller: 5V/246mA → 1.23W
    - Accelerometer/Gyroscope: 3.3V/1.2mA → 3.977mW
    - Bluetooth 5V10.7mA →53.5mW
    - Level Shifter(max) → 0.36W
- Power will be supplied from the motorcycle via USB cable

Updated System Specifications

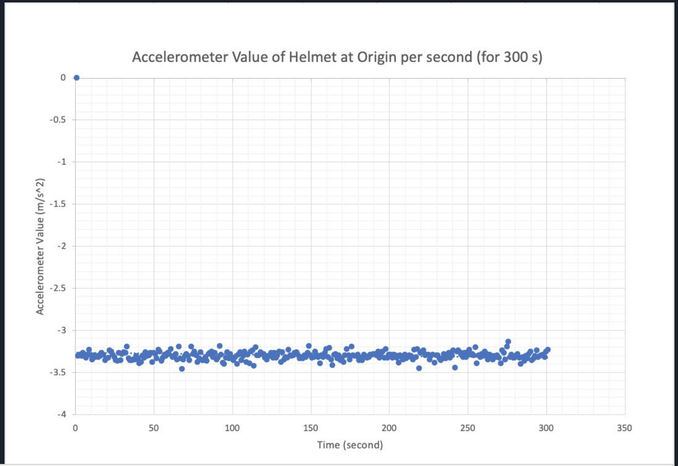
Software Specifications - Blindspot Detection

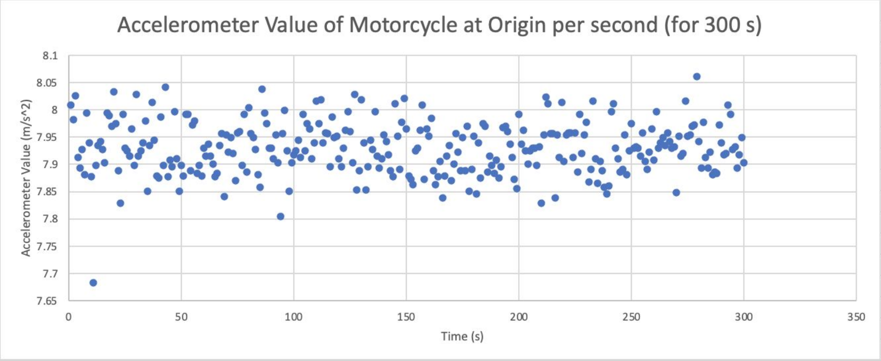
- System must execute algorithm at at least 10 frames per second
- System must detect and classify all vehicles within 120 ft of the system
- System must detect all vehicles in the blindspot region of the video frame
- System must detect and classify all vehicles within 80 ft of the system in low-light environment
- System must determine the proximity of vehicles within the blindspot region and classify into one of three categories: very near (0-10 feet), near (10-30 feet), far (30-100 feet)
- Send a bluetooth signal to helmet system with side and proximity information, once vehicle enters blindspot



Software Specifications - Fall Detection

- System must be able to accurately initialize crash protocol
  - Detected 100% of the time by comparing the accelerometer readings from the helmet and motorcycle
  - Communicate to the user via LED strip by pulsing the lights as a means of alerting the rider that a fall has been detected
- System must be able to terminate the crash protocol through user input
  - Detect when at least one accelerometer returns to origin orientation, the crash protocol is terminated
    - Communicate to the user via LED strip lighting that the crash protocol has terminated

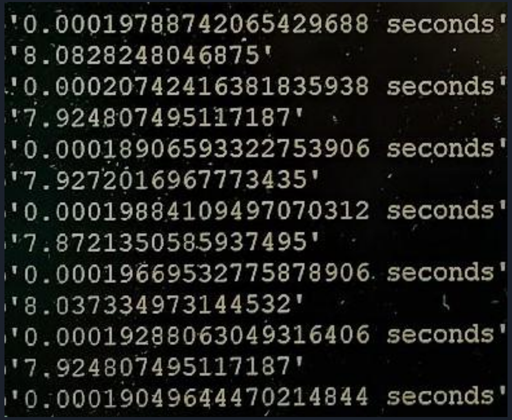


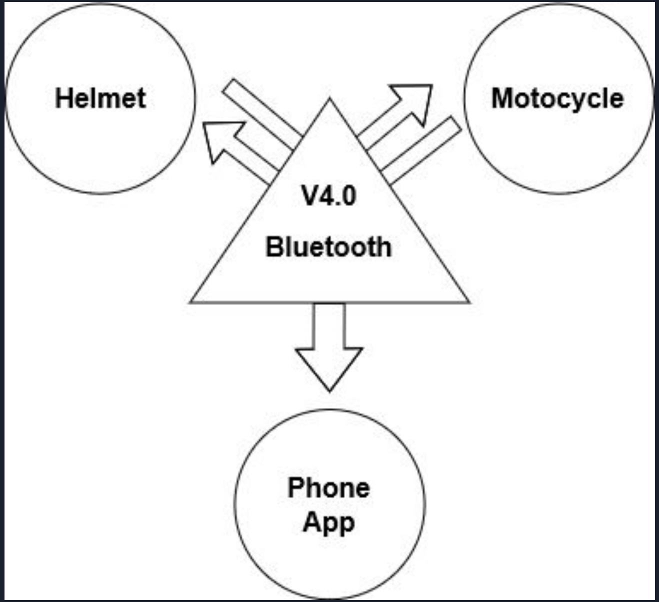


Bluetooth Specifications

- Raspberry Pi 3 A+: 4.2 BLE
- Raspberry Pi Zero W H: 4.1 BLE
  - 100 meters
  - 2.402 – 2.481 GHz
  - 1 Mbit/s Transfer Rate
- Future for CDR:
  - Jetson Nano USB Bluetooth Adapter 4.0 BLE Setup
  - Phone App Bluetooth Setup
  - Proximity between the bluetooth connected devices
  - Further latency testing between the bluetooth connected devices e.g. creating a normal distribution

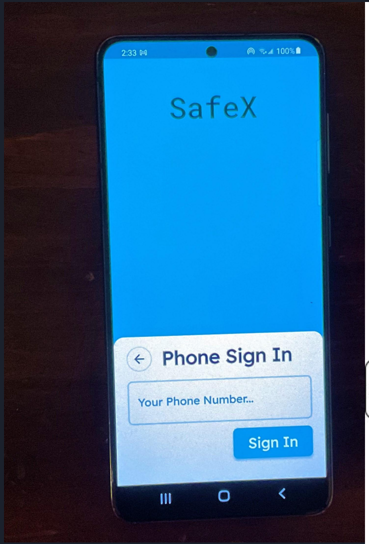
Testing Bluetooth with latency test

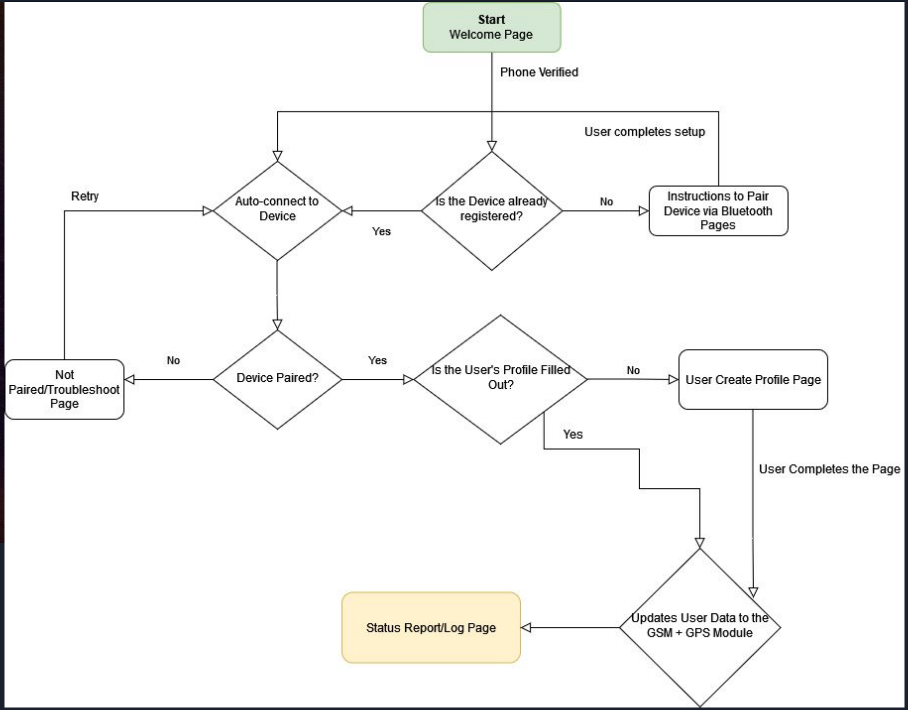




Mobile App Specifications

- Compatible with iOS and Android
- Phone Verification
- Profile Creation
- Future Goals:
    - Smoother Interface
    - Bluetooth Implementation
    - Upload Profile via Bluetooth to the GSM Module





Verifications and Demos

Blindspot Detection Verification

- Setup road simulation for test footage
  - 18 ft lane width
  - 160 ft road length
  - 8 scenarios with 26 blindspot passes
- Results
  - 90.96% detection rate in full lane (160 ft)
  - 100% detection rate within 120 ft mark
  - 87.51% classification rate as car
    - Misidentifying large car as truck
  - 100%; successfully flags the cars passing through the blindspot 26/26 times

Blindspot Detection Verification Demo Video

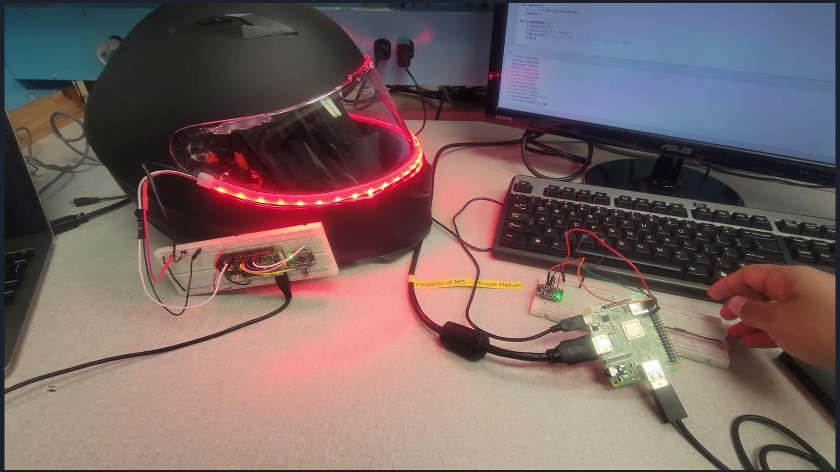
Blindspot Direction Demo Video

- Left or right LED Strips light up according to the location of vehicle in the blindspot
    - **Verification Type:** Demonstration

Fall Detection Verification & Demo

- Establishes communication from the motorcycle to helmet system, sending 100% of the data when not in crash mode
  - **Verification Type:** Demonstration

- LED Strip lights up red to indicate a fall has been detected 100% of the time;
  - **Verification Type:** Demonstration
- LED Strip lights up green to indicate idle state of system
  - **Verification Type:** Demonstration



CDR Deliverables

CDR Deliverables

- Blindspot Detection System
  - Detect & classify objects within 120 ft at 10 fps, with proximity approximation
  - Transmit vehicle detection data from the motorcycle to the helmet via Bluetooth
  - Configure lighting system to indicate direction and proximity using the detection data
- Fall Detection System
  - Comparable the data transmitted from the motorcycle's gyroscope/accelerometer with the helmet's accelerometer to determine if the crash protocol should be initialized
  - Implement crash protocol cutoff sequence using push button in the helmet, with LED notification
  - Implement GPS module to detect the rider's location for crash protocol
  - Implement GSM cellular module to send out ES distress message
- Mobile Interface
  - Functioning interface with user profile using input fields that store the rider's information
  - Establish Bluetooth communication with Fall Detection System for crash protocol
  - Implement a digital cut off button to alert Fall Detection System of rider safety

Budget

Item	Cost	Quantity	Total		Item	Cost	Quantity	Total
Motorcycle:					**Helmet**			
HQ Camera	$50.00	1	$50.00		Light Sensor	$4.50	1	$4.50
Wide Angle Lens	$25.00	1	$25.00		Accelerometer	$4.95	1	$4.95
Accelerometer & Gyroscope (MEMS)	$6.95	1	$6.95		Raspberry Pi Zero W	$14.00	1	$14.00
Raspberry Pi 3 A+	$25.00	1	$25.00		SD Card	$9.99	1	$9.99
USB V4.0 Bluetooth Adapter	$13.95	1	$13.95		LED RGB Strip	$17.95	1	$17.95
					Level Shifter	$3.95	3	$11.85
		M Subtotal	$120.90		GSM + GPS	$0.00	1	$0.00
					Helmet	$69.99	1	$69.99
Shipping:								
Adafruit	$11.67		**Subtotal**	$254.13			H Subtotal	$133.23
Amazon	$32		**Total**	$298.00				

Item	Cost	Quantity	Total
Future:			
PCB	$100.00	TBH	$100
Cellular Model w/ GPS	$50.00	1	$50.00
Batteries	$5	4	$20
Subtotal	$120		

Gantt Chart

Legends
Dhruv = DVK
Jeff = JW
Paige = PW
Sidd = SS

Task	Team Member(s)	Start — End Dates	2021 Finals 1/13 — 1/17 (M T W TH F)	Winter Break 12/20 — 1/21 (1 2 3 4 5)	Week 1 1/24 — 1/28 (M T W TH F)	Week 2 1/31 — 2/4 (M T W TH F)	Week 3 2/7 — 2/11 (M T W TH F)	Week 4 2/14 — 2/18 (M T W TH F)	Week 5 2/21 — 2/25 (M T W TH F)	Week 6 2/28 — 3/4 (M T W TH F)	Week 7 (CDR) 3/7 — 3/11 (M T W TH F)
Overall System											
Website Design	SS	12/18 — 01/28									
Battery power monitoring with low power alert	PW, SS	1/25 — 2/18									
PCB Design for helmet and motorcycle	PW	1/25 — 2/4									
Bluetooth transmision for motorcycle and helmet	JW	1/27 — 3/8									
Phone App with Bluetooth implementation	JW	1/31 — 3/9									
Blindspot Detection											
Object detection and tracking using extra wide angle camera	DVK	1/25 — 2/11									
Night vision object detection and tracking	DVK, JW	1/31 — 2/17									
Left vs right side blindspot notification	SS	1/26 — 2/04									
Light sensor to determine and alter intesity of the lights	PW, DVK	2/16 — 3/4									
Fall/Crash Detection											
ES alert using cellular GSM	JW, SS	2/17 — 3/4									
GPS tracking for ES alert	JW	1/27 — 2/18									
Fall detection and notification 1/2 sem	PW, SS	1/31 — 2/25									
ES alert cutoff via button 1wk	PW, SS	2/11 — 3/7									

Q & A

Thank you!