SOLUTION DESIGN DESCRIPTION

Automoe

PAWAN GAIRE | SAMANTHA-JO CUNNINGHAM
SATCHIN CAMPBELL | SAVANNAH MCCOY

November 6, 2018

<u>**Section 1: Individual Solution Designs**</u>

Design A

Communication + Data Flow

- Server, low-latency
    - Record{car_id, timestamp, gps coordinates, }
    - Use timestamp and gps coordinates as important secondary keys to make the api call faster since we have to query those then pick based off those.
    - Server-side processing so only relevant info is returned to car, which maneuver should be taken, what deviation from normal.
    - Clears server of data older than 20 seconds to minimize space usage.
- Client, RPi on Car
    - Serves location data periodically to the server creating new record entry
    - Requests periodically for special instructions
- Rpi to Arduino
    - Sends controls for real time movement.
    - Default autonomous movement from the trained model.
    - Two non-default behaviours
        - Special Instructions from server to do one of several predefined maneuvers e.g. swerve left, swerve right, speed up, slow down
        - Special response to priority stimulus mainly front ultrasonic sensor
            - IF distance is critical i.e speed*distance_to_object < minimum_time_to_come_to_halt(speed) + buffer_time_in_seconds THEN pull brakes which would also need to account for vehicles behind it.
            - Recovery from both should be accounted for in training model.

Layout

- 4 Ultrasonic sensors placed facing each classic direction, front back left and righ
    - Optional : 2 more, one for each blind spot
- Infrared at front
- Camera on top towards front for clearest view of signs and potential street markers.
- Choose car with slimmer body to minimize width to account for
- Choose car with high top for more space to place sensors etc.

Table of Inputs and Outputs between pairs of devices

Input/Output	Output to RC Car	Output to Arduino	Output to Rpi	Output to Server
RC Car	-	-	-	-
Arduino (Combined with Ultrasonic Sensor, Accelerometer)	Control instructions: Direction and Speed	-	Ultrasonic Sensor Reading, Last Controls sent to Arduino, Accelerometer Reading	
RPi (with camera)	-	Camera Data, Ultrasonic Sensor Data, Accelerometer Sensor Data	-	GPS data, Camera Information, Ultrasonic Sensor Information
Server (combining communication and a	-	-	1) Default Autonomous Instructions 2) Non-default Behavior	-

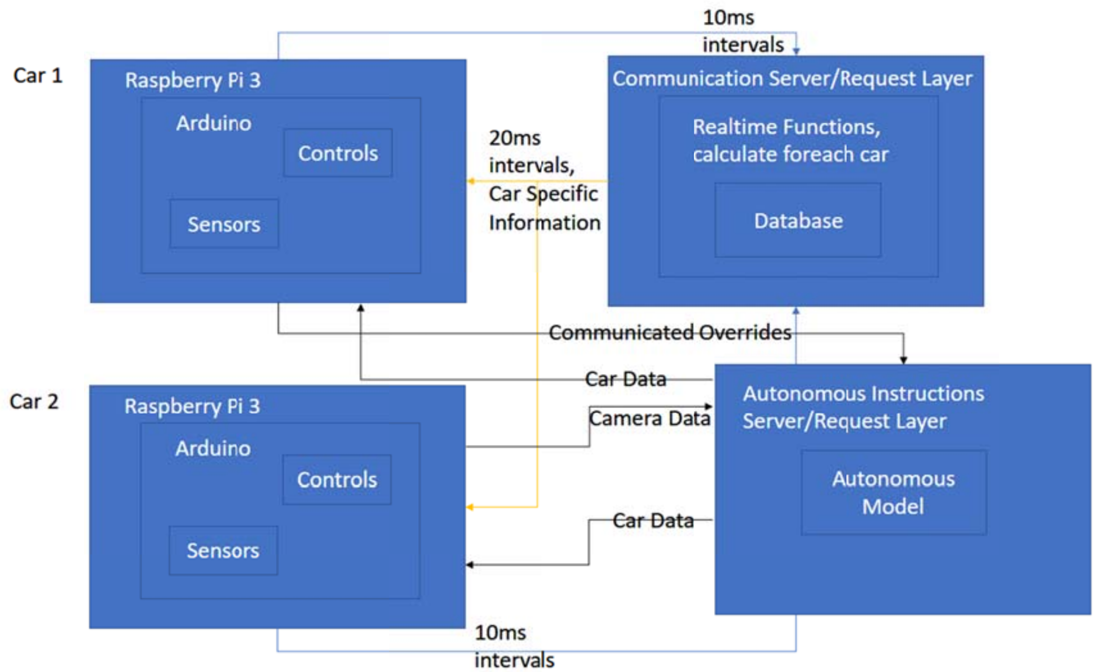
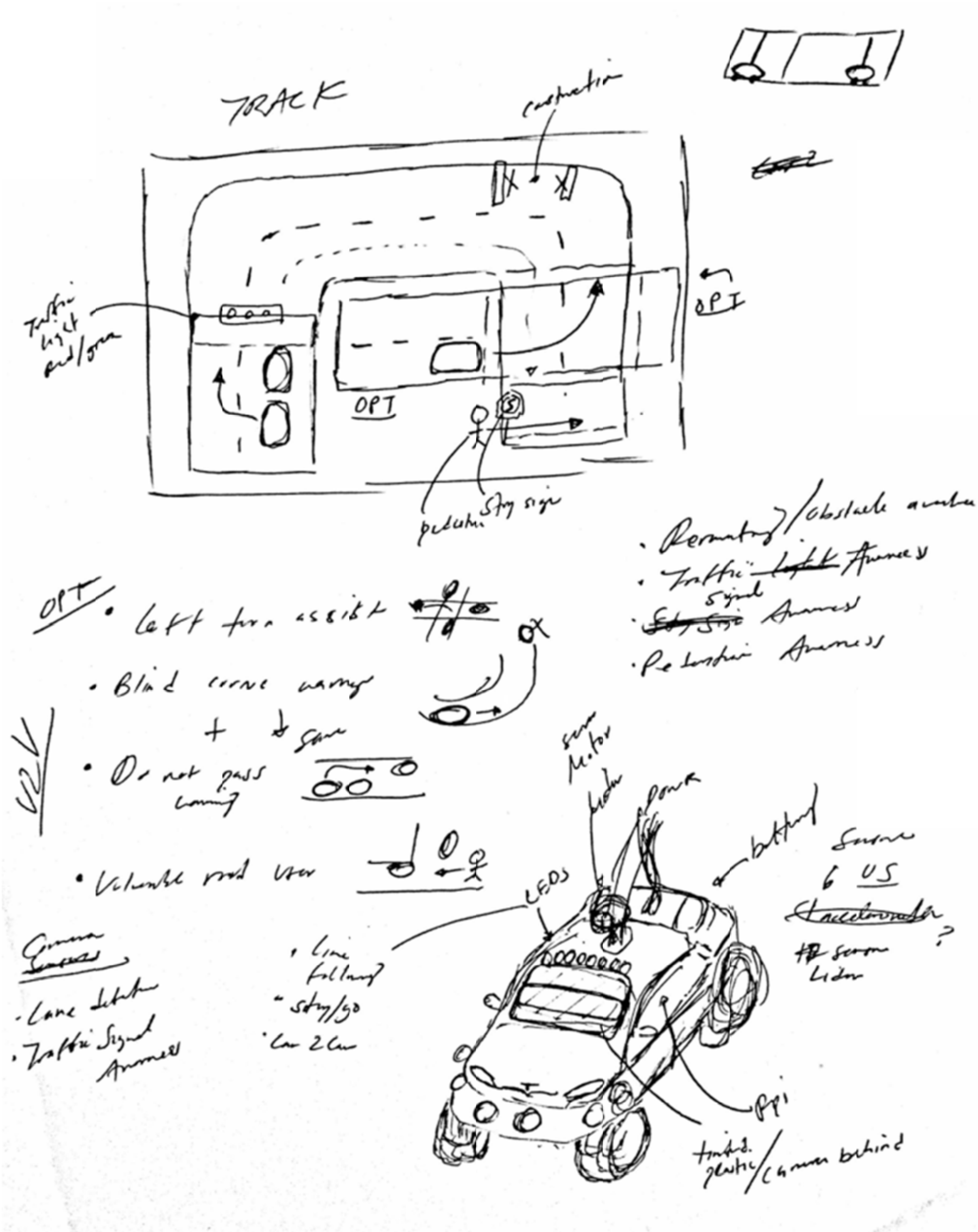


Figure 1 : Design 1 Flow of Data and Sharing of Responsibilities

Design B



2 cars travel through custom track equipped with traffic signals and obstacles. Cars use vehicle-to-vehicle communication to share data and make road decisions.

Track Features:

- Red/Green Traffic light
- Construction Zone (Obstacle)
- Crosswalk with Stop Sign
- Optional: Pedestrian at crosswalk
- Constant charging for vehicles

Smart-Car Abilities:

- Lane Detection
- Traffic Signal Awareness
- Obstacle Avoidance
- V2V Communication

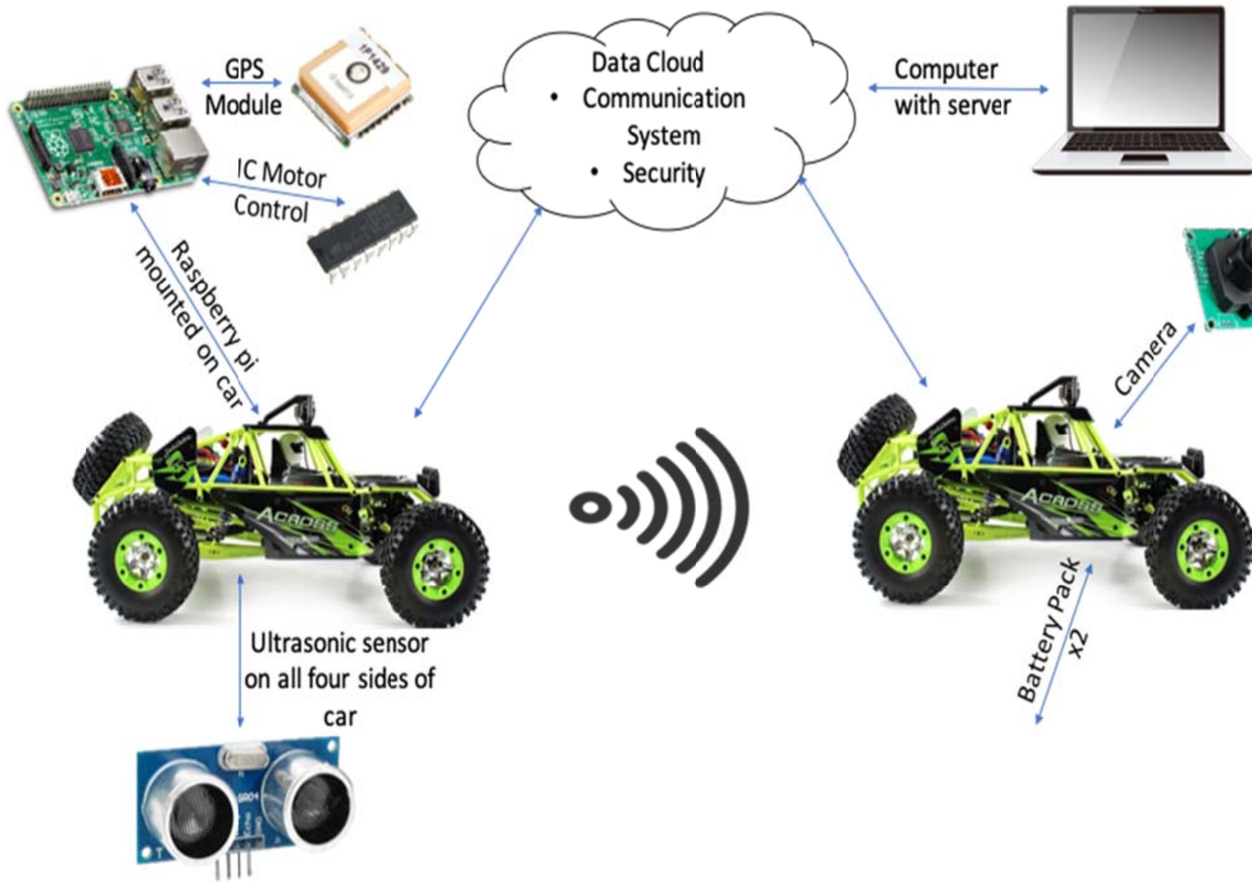
Equipment:

- Raspberry Pi 3+
- Arduino Mega
- 1 Lidar sensor
- 8 Ultrasonic sensor
- Camera (behind windshield)
- Leds mounted atop roof above windshield
- Custom car frame/body to enclose/mount other equipment

Optional V2V Signals:

- Vulnerable Road User (Pedestrian/Biker)
- Blind Curve Warning
- Do not Pass warning

Design C



Hardware Components:

- Ultrasonic sensors will be attached to all four sides of the RC car (front, back, left, right). The purpose of the ultrasonic sensor is detect how close the car is to other objects on the road.
- The camera will be mounted on the front of the car to collect surroundings data, which will be used to train the model for different road signs.
- The raspberry pi will be mounted on top of the car and will serve as the processor for the car.
- IC motor control will be used to control the speed of the motors on the IC car.
- 2 battery packs will be used to power the raspberry pi, the car and the motor control.
- GPS module will be used to collect the real time data of the car's location.
- Bluetooth module used for car to car communication.
- Wifi Module used for the communication between car and server.

Optional Hardware

- LCD mounted on RC car to display vehicle status information.

Software Components:

- All data collected will go to a cloud or dedicated server on a laptop.
- Raspberry Pi will be programmed to react to different environments (prevent collisions)

- A stop sign
- Car slows down in front
- When it is appropriate to switch lanes

Design D

This project aims to develop lightweight cybersecurity schemes, privacy aware communications, adaptive speed control, automatic braking, rerouting, information sharing using wireless access technologies and display vehicle's status information.

Display info on LCD or Android Phone

Must haves:
Self driving on artificial track
Street sign detections (stop sign, traffic light)
Collision avoidance

Input: we can use raspberry pi camera module and ultrasonic sensor to get the surrounding data. Ultrasonic sensor used to measure distance between the obstacle. We can send the sensor and camera data to the computer using wifi connection.

Mode of communication: use bluetooth module / built in wifi module to transmit data to the server. Use gps module to read location.

Processor: receive data from raspberry pi; train neural network; drive the car using prediction from neural network by detecting street sign and making decision.

Setup a server on computer that receives sensor and camera data from autonomous car. We can train neural network to make predictions on street signs. First need to train the model by manually driving the car. The model should also be trained to recognize stop signs (we might need to train different model for that purpose).

DC motor control: We need a dc motor control IC L293D to control the speed of motor. With this we will be able to control forward/backward direction of motor and also speed of motor.

Power supply: Power source is required to power the motors, L293D IC and raspberry pi.

Section 2: Description of Top 2 Designs with Pros and Cons

Design 1:

Introduction: This is a merge of design A and B incorporating the lidar usage along with the commonalities in the network setup among the RPi, Arduino and server. They both focus on the system's breadth more than design C and D.

Equipment : GPS Module, 8 ultrasonic sensors, camera module, RPi3, arduino, lidar sensor

Layout : 8 ultrasonic sensors laid around vehicle, three at the front, another three at the back and one on each side. Camera inside the car, "looking" through the windshield section.

Test Cases: Car must respond appropriately to walking pedestrian, stop signs, blind curve detection.

Power Solutions: Having overhead cables that connect to the back of the car to provide power constantly.

Data Flow: Ultrasonic sensors feed data to the arduino, which will pass the data onto the rpi3 and then onto a laptop server along with camera feed. The laptop will provide instruction on what to do based on the camera, along with other cars data in order for each car to decide what to do.

Data validation: The server uses the lidar, gps, and ultrasonic sensors to validate car to car communication by cross-checking the data using gps to know which cars' data is relevant.

Pros and Cons of Design 1

<table>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>

Modularity within the components

Arduino controls movement + ultrasonic sensor data collection
RPI3 communicates with server and captures camera data
Development is more manageable this way.
Each person can own a significant part

Many sources of information for cross-validation (lidar, ultrasonic, camera, gps)

This can improve security by requiring the ill-intended to compromise each source of data in order to cause true harm
This also improves the structure allowing for easier improvement of the smaller features

8 ultrasonic sensors provide more detection data.

</td><td>

Server is a single point of failure.

If the server malfunctions, the entire system fails

Raspberry and Arduino combination adds cost probably unnecessarily
The GPS is dependent on external sources which can be compromised or interfered with or just inaccurate.
Also as we plan to use this model indoors with small vehicles GPS is impractical
Many variables with no clear way to rank the data

</td></tr>
</table>

Design 2:

Introduction: This is a merge of designs C and D which both take the same approach to the flow of information using the RPi without the Arduino for logical and vehicular functions using an IC to control the cars. Though all designs were similar, it was deemed best to group A with B and C with D in order to be comprehensive in our final analysis and decision making.

Equipment: 4 Ultrasonic sensors, Raspberry Pi, IC Motor Control, LCD screen, camera module, GPS module

Layout: 4 Ultrasonic sensors, one on each side, camera module and LCD screen on the roof of the RC car

Data Flow: Ultrasonic sensors feed data to the raspberry pi, and then onto the server with the camera data at fixed periodic interval. The server provides instructions for the raspberry pi which controls the car's IC motor and controls the movement of the car.

Data validation: The server uses the lidar, gps, and ultrasonic sensors to validate car to car communication by cross-checking the data using gps to know which cars' data is relevant.

Power Solutions: Two battery packs per car (to power raspberry pi and car), switch as needed throughout testing.

Test Cases: Car must respond appropriately to traffic signals, a decelerating car ahead, switch lanes as needed.

Pros and Cons of Design 2

<table>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>

Not so pricy, minimalistic approach to choosing parts and number of parts.

The cost makes the system as a whole less expensive :
Less Cost
Less Data to secure
Less data to use

Removable battery packs provide more car options to choose from.

Charge/Usage time ratio of 2 hours/15 min norm is not ideal for testing

Few things for the car to support

The less things to put onto the car means the more efficiently they can be placed and the easier it will be for the Mech E’s to design a body for everything to fit into/onto/

LCD displays car data

This can display relevant test data and real time data to the user of the vehicle

</td><td>

Raspberry Pi expected to handle alot : sensor data collection, control movement of car, communicate with server.
Displaying information about the system on the LCD is useful for developing but a source of compromise in the final product.
Solely depends on 4 ultrasonic sensor for obstacle detection.

</td></tr>
</table>

Section 3: Decision Matrix for Top Design Selection [1-10(best)]

Design	Cost(no tax)	Complexity, Amount of data	Security/ Validation	Ease of development	Overall Score
1	7($547)	7	10	8	32
2	9($331)	10	7	6	32

In order to determine the best model we decided to use the following fields to value each design:

Cost: The cheaper the system, the more accessible the system and device becomes which is ideal for our development but also helps us to assess the tradeoff between cost and performance.

Complexity, Amount of Data: The more data there is, the more encryption of levels of security will be required and larger amounts of data that will be needed to be transmitted among each component. The result is that in a sensitive real-time system such as this, these will affect latency which is a very bad thing and affects the driving experience itself.

Security/Validation: Is there enough security when it comes to the data? Is there validation of external stimulus using multiple sources or some other means?

Ease of Development: It is important to finish this project in the scope and timeline of this class and if development is simpler then we are more likely complete this project and deliver our solution.

Section 4: Solution Design Description for selected design

Introduction:

Pulling the best from design one and design two we removed unnecessarily costly items such as the lidar sensor along which added unnecessary complexity. The GPS module did not persist to the final design due to its impracticality of indoor usage. Design 2 addressed necessary details in regards to controlling the car while Design 1 addressed details of the network implementation, both of which are important to the success of the whole system.

Equipment:

Part Name	Link	Model	Cost	Quantity	Price*
Car	Link	Feiyue FY03 Eagle-3 1/12 2.4G 4WD	77	2	154
Raspberry Pi	Link	Pi3 B+	40	2	80
Ultrasonic Sensor (pack of 5)	Link	Elegoo HC-SR04	10	4	40
Jumper Wires	Link	Generic	13	1	13
RPI Camera Module	Link	Module V2-8	24.9	2	49.8
Arduino	Link	Mega 2560 REV3	35	2	70
Rechargeable Batteries (pack of 2)	Link	7.4V 1500mAh 15C T Connector	25	1	25
IC Motor Control	Link	L293D	10	4	40
Total					471.8
Upscaled total plus 50% (Tax + Shipping)					697.7

*Current up to Nov. 5, 2018

Input Data:

8 Ultrasonic sensors

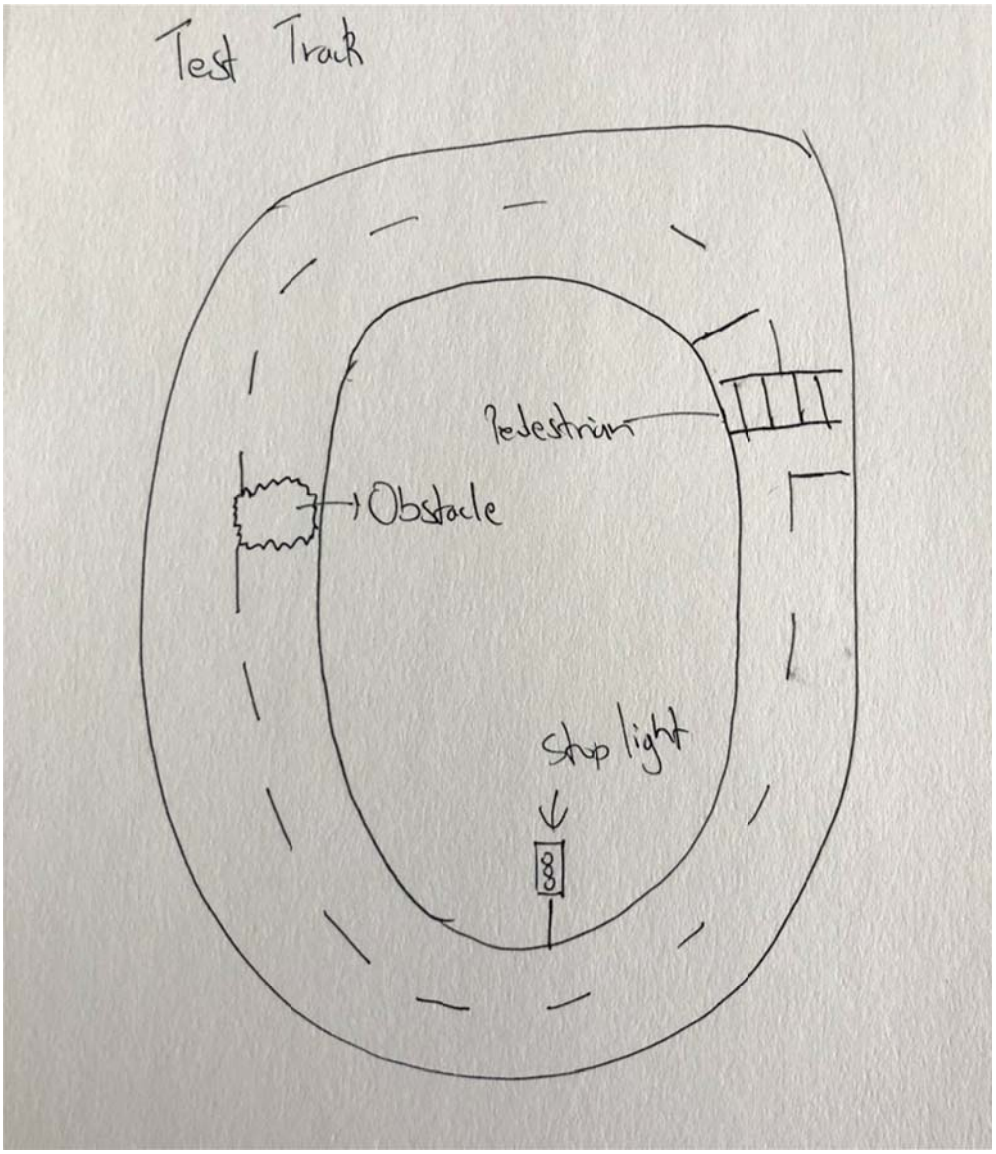
3 placed at the front of the car, 3 at the back, one on each side. The three on the front and back should be evenly distributed to maximize the readable area on boths sides.

Camera module

This is used to capture multiple frames per second to determine the response of the RC car using the ML trained model.

Demo Map:

The track will test lane detection, obstacle avoidance and response to traffic stimulus (stop light, stop sign, pedestrian). A rough sketch is shown below



Controllers:

Raspberry Pi 3 (B+ Model)

Used for the sensors, camera module. Consolidates all sensor data at timepoints and sends to server over network.

The RPI also acts as a client for the autonomous functions, several frames are sent per second to the server and the next functions i.e. movements for the cars are determined on the server, passed back to the RPi which then issues the logical commands to the arduino for the movement.

Motor control IC L293D

Used to control speed of the motors of the RC car. This removes the additional level of logic needed to control complex movements of the RC car. This is controlled by the Arduino and outputs to the cars motors.

Arduino Mega

Control car movement (releasing these functions from the RPi and moving them to a dedicated device) Also collects sensor data serving it to the raspberry pi

Analysis: The reason for using multiple devices is to share the load among each device reducing the likelihood of any component being unable to handle the load. The Raspberry Pi is effectively a client level, control unit of the entire car, while the arduino acts as a controller of the car taking instruction from the Pi and also collecting sensor data and sending it to the Raspberry Pi. The IC allows us to control the cars motors properly

Communication:

Autonomous Movement:
- Initially hosted on laptop
- Finally loaded on RPi, if possible.

Data Sharing:
- Laptop hosted server (Initial)
- Finally moved to one of the following options (to remove a single point of compromise)
    - Radio Transceiver
    - Bluetooth
    - Wifi

Neural Network:

- Host on both laptop and board, laptop first then implement on board in order to remove the server as a single point of failure allowing the car to function on its own

Validation Information:
- Send sensor information from one car to another to validate sensed data.
*This works for two cars but no more.

Power Options:
Rechargeable battery x2
Battery charger x2

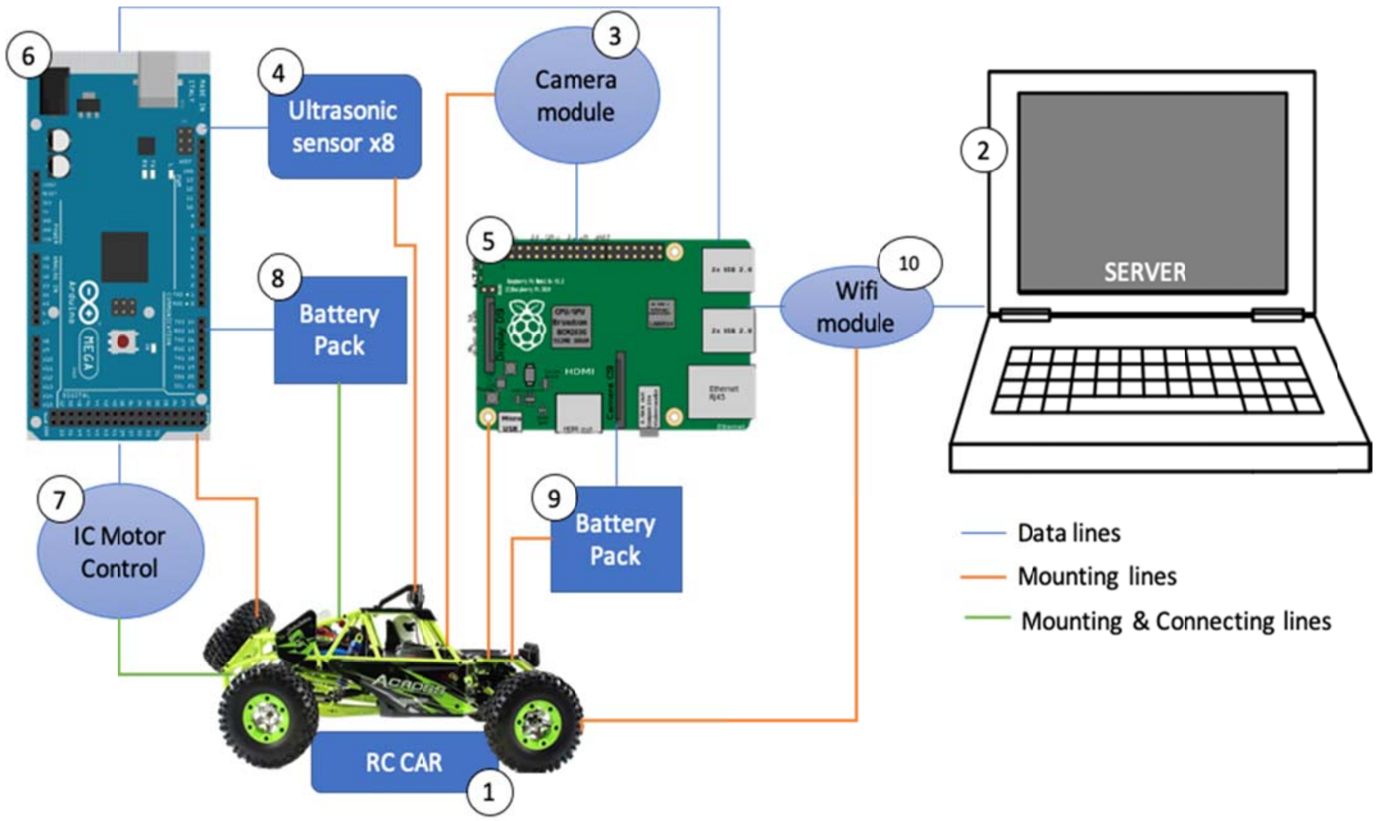
Definite preference for removable rechargeable batteries that we can buy multiple of as we don't have to switch cars if the battery fails and we can swap batteries instead of waiting for ludicrous charge times.

Layout:

Test Cases:

- Switch lanes as needed
- Slow down/stop behind vehicle in-front
- Reroute around obstacle (change lane and come back)
- Respond to Red/Green traffic light
- Stop at stop sign
- Avoid Pedestrian
- Secure vehicle-to-vehicle Communication

Design Schematic



Flow of Data:

The Arduino Mega (6) collects all the Ultrasonic sensors (4) that are connected to the RC car (1) and sends it to the raspberry pi (5). The camera module (3) captures several frames per second, then the sensor data and frames are sent via the raspberry pi (5) to the server (2) via the wifi module (10). The server (10) then process the data and sends instructions for the RC car back to the to the raspberry pi (5) which issues the logical commands to arduino (6). The arduino (6) along with the IC Motor Control (7) is used to control the movement of the car. The battery packs (8) and (9) are used to power the arduino (6) and the RC car (1) and the raspberry pi (5) respectively. Together this system will allow the system to respond to several scenarios, for example if a car in front begins to decelerate rapidly, the sensors should detect the change in this motion and after it is passed to the RPi then to the server, this will be interpreted. The server will in turn issue instructions to slow down which the RPi will send to the Arduino which converts the instructions and sends the control signals to the IC motor controls to the RC car's motors to effect the slow down instructions to avoid a collision.

Note :

Additional circuit elements will be required i.e. resistors, capacitors etc. but these will be determined during development in the iterative process.