Department of Electrical Engineering and Computer Science

Howard University

Washington, DC 20059

EECE 404-01 Senior Design

Spring 2024

10/10

Final Report Project: Deliveroid Enhancement (D2)

By

Mohammed Akinbayo

Anthony Berry

LaDelwyn Mealey

Faculty Advisor: Dr. Charles Kim **Date Submitted: 4/23/2024**

Abstract:

The transport of physical documents and other various items in an office space can still be automated. Our solution design for this problem, the Deliveroid, is an autonomous robot that seeks to carry out this transport as easily and as efficiently as possible. Using knowledge of microcontrollers as well as the engineering process in general, our group of three electrical and computer engineers attempted to assemble this solution design but found that we were unable to complete our short-term objective due to time and material constraints. Overall, the experience offered by this project will allow the three engineers to assess any engineering task much more realistically in the future.

Problem Statement:

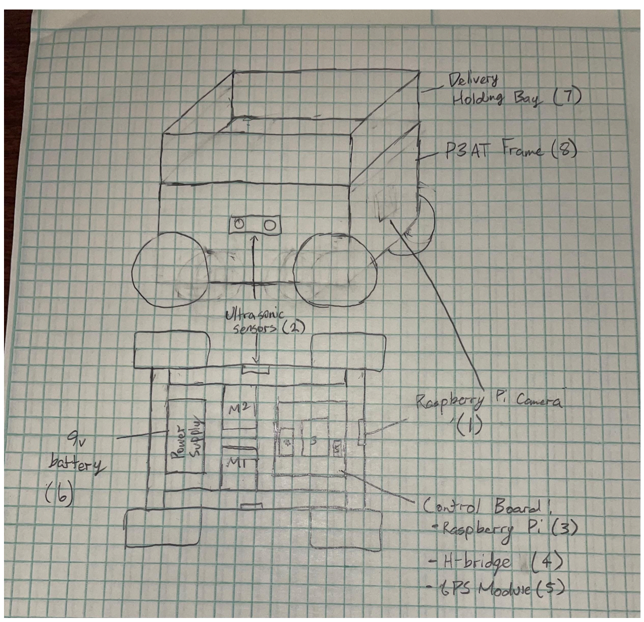
The current robot is not able to drive automatically or detect/avoid obstacles which could lead to damage to the robot. The approach to this problem is to equip the robot with cameras and sensors, and program in obstacle avoidance algorithms. This will allow the robot to drive on its own smoothly and avoid obstacles. The competitor to this approach would be to use motion sensors instead.

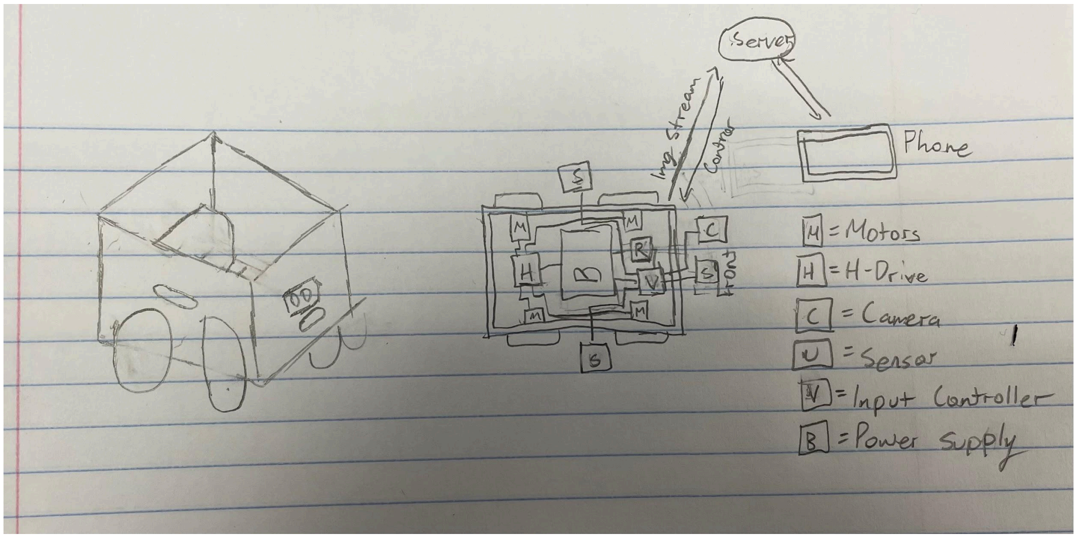
Design Requirement:

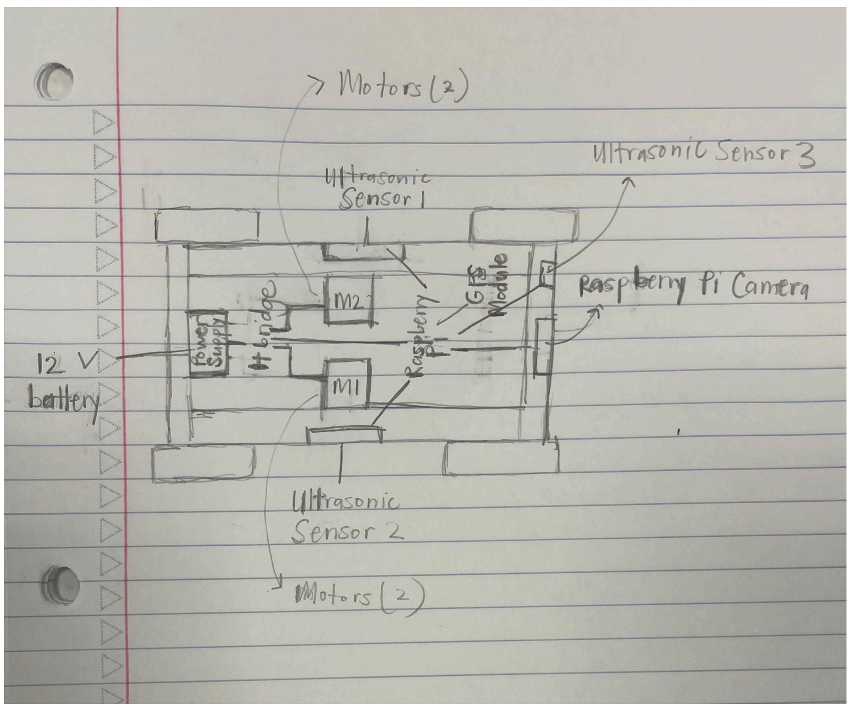
For the deliveroid enhancement, the items needed are the deliveroid robot, two cameras and one ultrasonic sensor (later updated to 1 camera and 3 sensors). The deliveroid must be compliant to the university rules of media capture. The socio - cultural constraints include the robot being able to move around varying sizes of groups of students without interrupting them and images captured by the robot may not be used for any other purposes other than obstacle avoidance. The environmental constraints for the robot are that it must be powered by electricity and must fall into the power constraints set by the power supply already built into the deliveroid.

Solution Design:

The deliveroid will use a P3AT frame [9] which was supplied. All of the electronics on board will be powered by a 12V central power supply [6]. There will be 3 ultrasonic sensors (40kHz) [2] and a Raspberry Pi camera module [1] attached. These, along with a GPS module [5] will help the robot to accomplish the primary goal of driving autonomously without human input (with the only input being where to go). The sensors will provide data to the Raspberry Pi, which will then use an AI pathing algorithm to figure out how and where to travel to the provided location in real time. A dashboard will connect to the robot via http and this will serve as the main point of human to robot connection. Once the robot is turned on, it will remain in standby mode until it is given direction to go to a certain location via coordinates. Once it reaches the location, food/items can be held in the holding bay [7].







Agile Workflow and Weekly Plan:

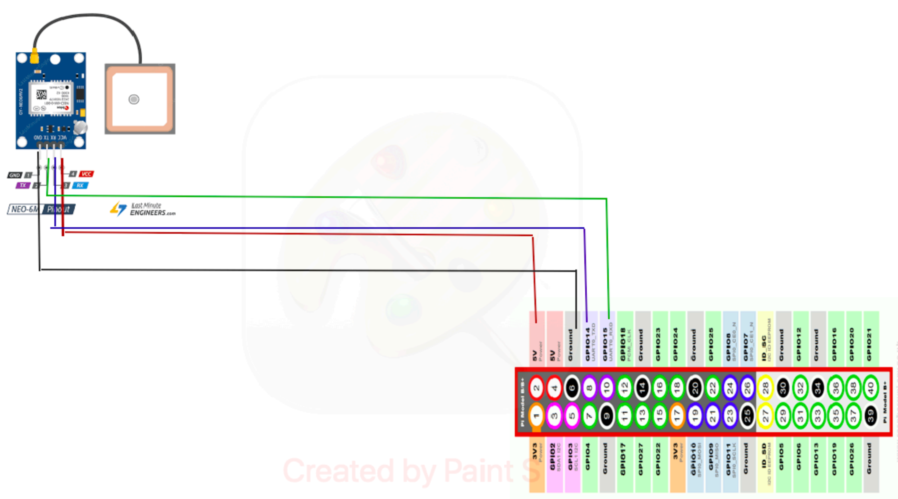
At the beginning of the semester, an agile workflow plan was drafted in order to plan in increments what would be completed. For the first sprint increment, the goal was to make sure that the deliveroid turned on and functioned properly by moving back and forth. This required a few weekly tasks such as gathering parts for the deliveroid (raspberry pi, battery, sensors, etc).

Once the wiring was completed for connecting the battery, motors, and raspberry to the h bridge, a test would be conducted with LEDs and python code in order to test movement and functionality.

For the second sprint increments, the goal was to have the robot able to move from one point to another from coordinates via a web app. The tasks for this increment included coding the web app and server connection on raspberry pi, attaching the sensors and camera, and testing the connection between the robot and web app. The third sprint increment goal was to make the deliveroid drive autonomously and avoid obstacles. In order for this to be done, the tasks included coding in AI for autonomous operation based on sensor data. This increment also required coding in path finding algorithm and obstacle avoidance. Once these tasks were complete, the final task would be to place obstacles in the path of the deliveroid in order to test obstacle avoidance.

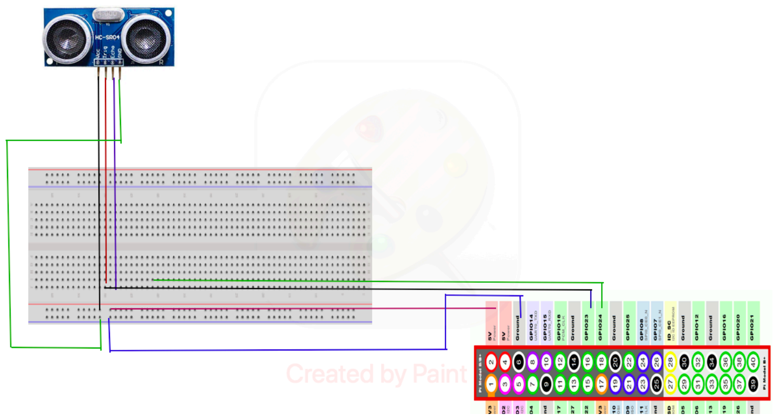
Project Implementation Process:

Due to delays with getting all of the required components, this increment was prolonged. Once the components arrived, the three ultrasonic sensors, GPS module, and camera were wired to the raspberry pi. The camera was connected to the raspberry pi by simply inserting the camera cable into the camera port. The GPS module was connected to the raspberry pi by connecting the VCC, RX, TX, and GND pins to the correct raspberry pi pins. Due to constraints and technical issues, both the camera and GPS module were scrapped from the final design. Instead, we went with a new solution design that would work with the last minute technological constraints (expanded upon in later section).

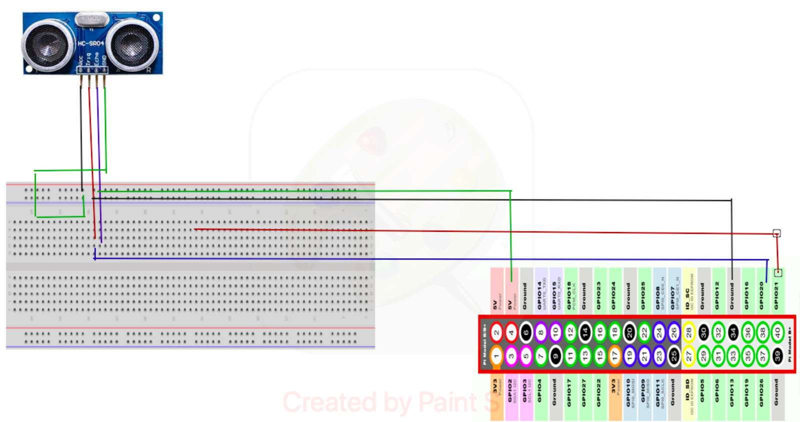


GPS Module Schematic

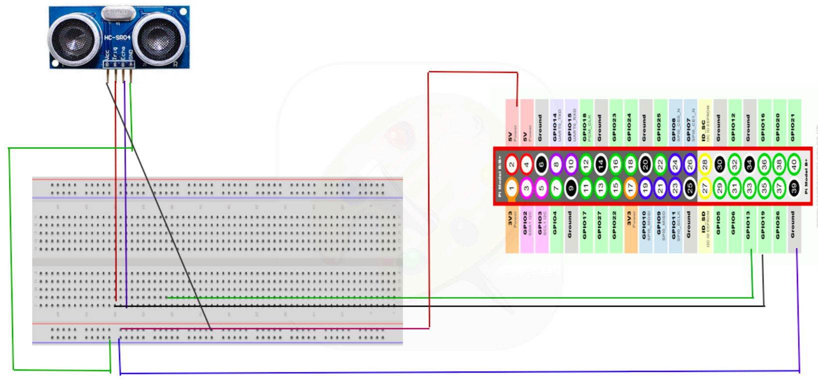
For the connections of the ultrasonic sensors, the VCC, TRIG, ECHO, and GRND pins were connected to positive/negative rails of the breadboard or in a blank rail. From the breadboard these connections were completed by connecting to the GPIO pins of the raspberry pi. Both 360 and 430 ohm resistors were used to create a voltage divider circuit and power the sensors.



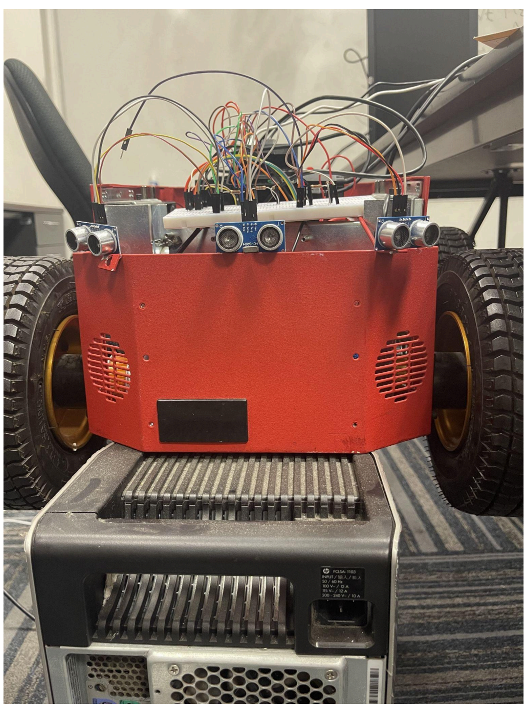
Sensor 1 Schematic

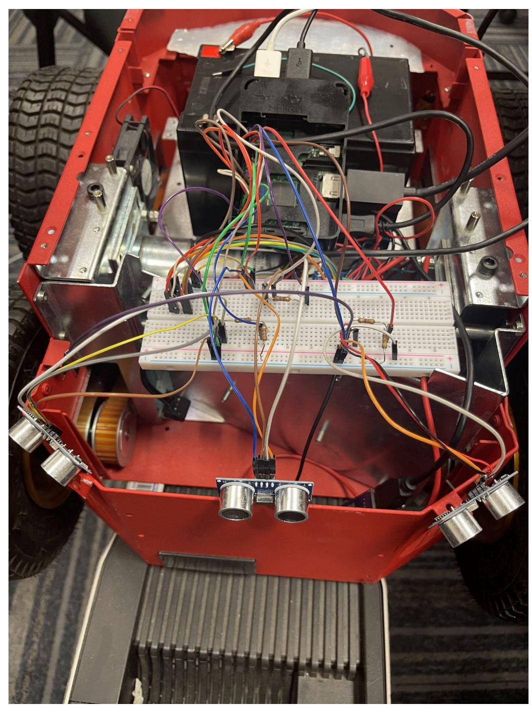


Sensor 2 Schematic



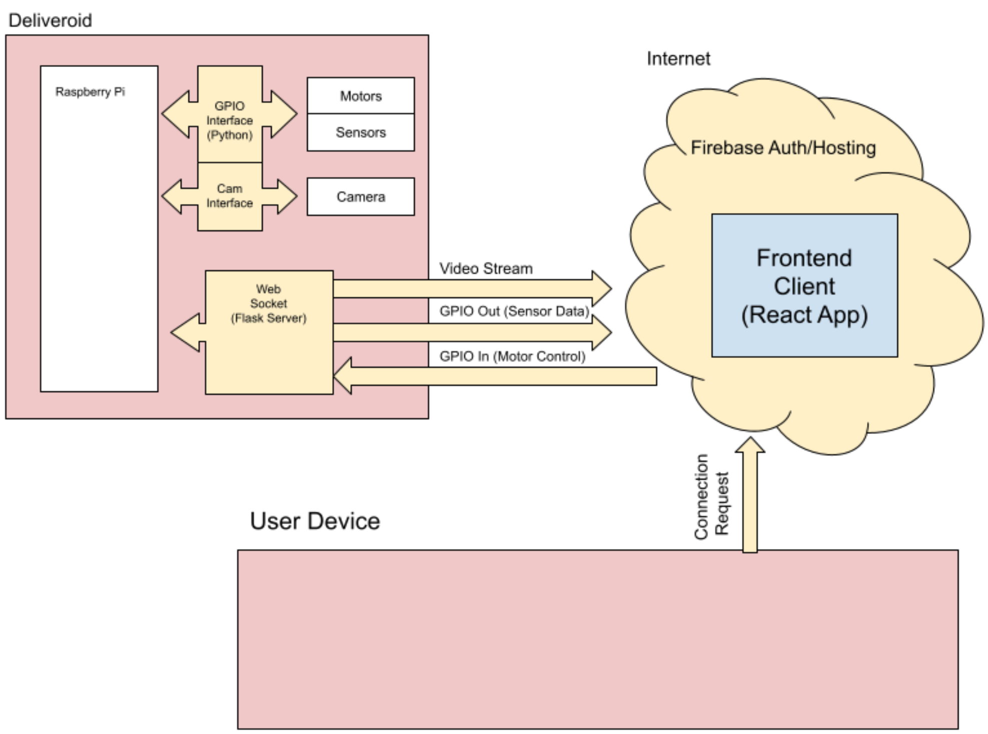
Sensor 3 Schematic





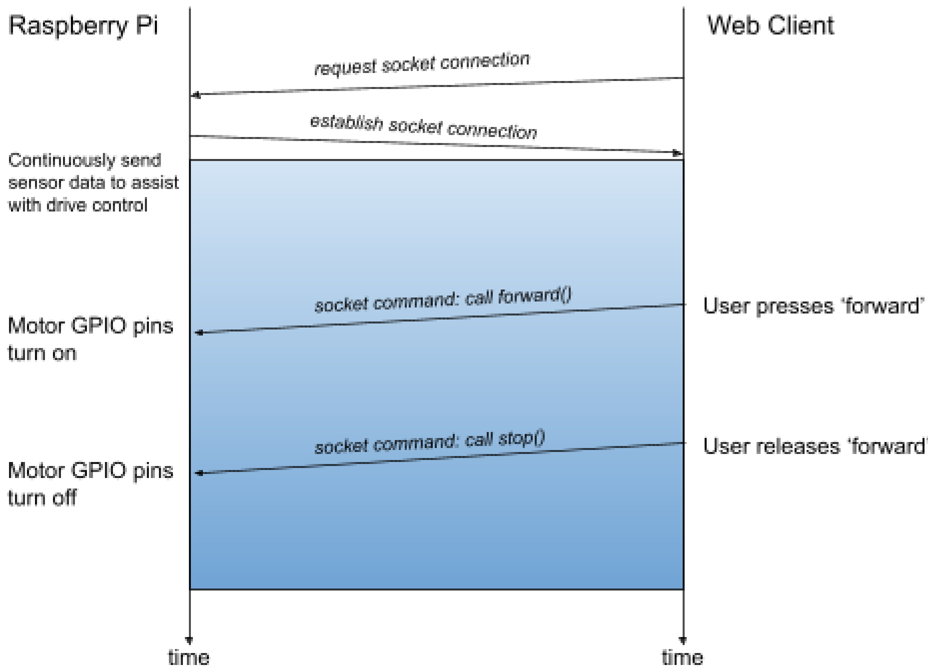
Updated Solution Design

As discussed earlier, due to technological constraints, the initial solution design had to be changed. The realization of further needed items (accelerometer, odometer, etc) as well as the malfunction of the camera meant that we had to take out the autonomous feature of the deliveroid. Instead, the robot will be controlled remotely via the Internet, allowing for users to still get snacks and other items without having to leave their residence. This required an updated approach to the backend and data communication.



Deliveroid Communication Diagram

In order to establish communication between the raspberry pi and web app, a Flask server was created that used the WebSocket protocol (Socket.IO) to open bidirectional, real-time communications.



Communication time graph

This allows a web client (ie. our React App) to communicate with/receive data from the deliveroid.

Conclusion

Though we had to change our scope and solution design, we still came up with a viable design that still achieved the overall goal of remote delivery of items. This project showed the imperativeness of proper research and readiness, as well as on-the-fly problem solving. There is still room for improvement. As discussed upon before, some more materials were needed that would make the objective of autonomous operation possible. With these materials, and more time, the initial solution design would indeed be very successful.