Team 13: Traffic to Vehicle Communication



Daniel Ackuaku, Hamilton Mutschler, Drew Smits, and Kevin Um

# 

© 2020, Calvin University and Daniel Ackuaku, Hamilton Mutschler, Drew Smits,

and Kevin Um.

# Abstract

This report details the Senior Design project proposal of Team 13, T2V, composed of Daniel Ackuaku, Hamilton Mutschler, Drew Smits, and Kevin Um. The proposed project is the design of a working prototype for a communication system between vehicle and traffic infrastructure for the improvement of traffic throughput and safety. The design consists of an interface for both vehicle and traffic infrastructure that would allow communication over Bluetooth. The location of each vehicle is determined by the traffic infrastructure via RFID readers and compatible RFID tags so that traffic flow can be controlled intelligently.

Due to the unprecedented circumstances regarding the COVID-19 pandemic, the second half of our spring semester was restricted in which our team could not meet in person and had no access to Calvin University’s Engineering Building. Because of this, the scope of the project had to be reduced. However, our team successfully displayed a working prototype of a fully autonomous figure-8 intersection through numerous online meetings and remote work.

# Table of Contents

[**Abstract**](#_fbxj14riyuaz) **3**

[**Table of Figures**](#_h2n05el77olb) **6**

[**1. Introduction**](#_qwr23p2arrj7) **7**

[1.1 Problem Definition](#_vxggwe21rbpb) 7

[**2. Project Management**](#_mu2tcn6pgix7) **8**

[2.1 Project Structure](#_5t3btxr6qqpr) 8

[2.2 Budget](#_lk6tcdegynku) 8

[2.3 Work Breakdown Schedule](#_bfxgnfhzhcsm) 9

[2.4 Team Organization](#_sw3j8p5oubnb) 10

[**3. Design Process**](#_u5invzmkg8eo) **11**

[3.1 Design Norms](#_qv49fgtrbchv) 11

[3.2 Research](#_x807pll3p943) 12

[3.3 Design Decisions](#_hyfb182os7xz) 14

[3.3.1 Design Criteria](#_ad67obtx191r) 14

[3.3.2 Design Alternatives](#_yos1fgn222eb) 14

[3.3.3 Design Choices](#_nfmcbsc9grm4) 15

[3.4 Requirements](#_3lkikblzpjt) 16

[3.5 Task Specifications and Schedule](#_t8gg75h1v06e) 17

[3.6 System Architecture](#_cbb0ggh35tf7) 18

[3.7 Integration, Test, Debug](#_naf1t4oqdgbp) 21

[3.7.1 Collision Avoidance](#_dt8s5ycr5kpr) 21

[3.7.2 Track Thickness and Corner Curvature.](#_axyco5xiy478) 22

[3.7.3 Traffic Control System](#_8om61my1xz4) 23

[3.7.4. Throughput Testing](#_tzqhz8g5k1nh) 24

[3.8 Production](#_xbcepsdegm0o) 25

[3.8.1 Prototype Production Budget](#_3i6fhdtwzwe7) 25

[3.8.1.1 Phase 0.5](#_1jws4mq549l6) 25

[3.8.1.2 Phase 1](#_u620j45a92jf) 26

[3.8.1.3 Phase 1.5](#_5ya8rp5su05b) 27

[3.8.2. Estimated Market Value](#_g2x4f1zgahwj) 27

[3.8.3. Per Unit Annual Cost](#_wg5hrnocw3d2) 27

[**4. Conclusion**](#_57z8r2typagd) **29**

[**5. References/Bibliography**](#_6pc59pfaysql) **30**

[**6. Appendix**](#_bhu8rjfoft9r) **31**

# Table of Figures

Tables:

1. Table 1. Project Prototype Budget
2. Table 2. Project Work Breakdown Schedule
3. Table 3. Team Organization
4. Table 4. Specification comparison between Bluetooth 5 and ZigBee (Collotta).
5. Table 5. Decision Matrix for Communication Medium
6. Table 6. Requirements for Phase 1.0 Prototype
7. Table 7. Task Specifications and Schedule for Project Phases 0.5 and 1.0
8. Table 8. Bill of Materials for Phase 0.5 of the Prototype
9. Table 9. Bill of Materials for Phase 1.0 of the Prototype
10. Table 10. Bill of Materials for Phase 1.5 of the Prototype

Figures:

1. Figure 1. ZigBee Network Topology (Malaric)
2. Figure 2. Bluetooth Mesh Network Topology (Baert)
3. Figure 3. T2V Final Product System Architecture Block Diagram
4. Figure 4. T2V Phase 1 Prototype System Architecture Block Diagram
5. Figure 5. T2V Phase 1 Prototype Software Architecture Block Diagram
6. Figure 6. Explanation of States in Prototype Traffic Control Software
7. Figure 7. 4WD Line-Following Car with Attached Reflective Paper
8. Figure 8. Final Figure-8 Track with Specified Dimensions
9. Figure 9. Hardware Diagram for the Traffic Control System
10. Figure 10. Final Prototype of the Traffic Control System

# 

# 1. Introduction

This report describes the project conducted by T2V, a group of four Electrical & Computer Engineering students enrolled in the Senior Design Project course, ENGR 340, at Calvin University.

This project seeks to create an improved communication system, which consists of a network thereby allowing for a transfer of information between the traffic light and vehicle (T2V) and vehicle-to-vehicle (V2V). The status of the traffic light is transmitted to vehicles both waiting or approaching the light. Having this system could improve flow in low-traffic situations, prevent side-impact collisions, reduce the need to accelerate and decelerate excessively in between intersections, and shorten the delay caused by drivers reacting to the traffic light.

## 1.1 Problem Definition

In most metropolitan areas, traffic density varies significantly throughout the day and cannot be efficiently managed by existing traffic light infrastructures. Algorithms controlling the infrastructures typically consist of either a time-based system or inductive-loop system. Inductive-loop systems can only detect the current state of an intersection, and cannot predict future traffic flow. Even worse, time-based systems do not use any real-time traffic data and often increase congestion (How).

Secondly, we plan to reduce the risk of having a side-impact collision in an intersection. According to the National Transportation Highway Safety Administration, roughly 8,000 fatalities occur annually in the US due to side-impact/T-bone collisions (T-Bone). To decrease this number, we plan to design and implement an interface for each automobile. The traffic light would detect an imminent collision, and send an alert to the automobile interface to avoid the collision.

# 

# 2. Project Management

## 2.1 Project Structure

This multiple phase structure used for project organization has been designed based on input given by Omar Waller, an industrial consultant. It is intended to break the project down into smaller more achievable stages while also keeping an eventual final product in mind.

**Phase 0.5:** Propose a feasible project and design a feasible prototype. Outline future phases of the project. This phase is intended to act as a planning phase for the scope of the project.

**Phase 1.0:** Develop and build a fully operational prototype that demonstrates the feasibility of the final product. This phase is intended to act as an achievable goal for the scope of this project.

**Phase 1.5:** Scale-up the prototype to include more advanced functionality. This phase is intended to act as a stretch goal for the scope of this project.

**Phase 2.0:** Develop and build a fully operational final product. This phase is intended to act as an eventual final stage of development for a final product for production. Realistically this could not be achieved within the scope of this project but is important to outline for the sake of developing project management skills.

## 2.2 Budget

Based on the desired prototyping tools required to assess the feasibility of the project the following provisional budget was proposed and subsequently approved. This budget covered the basic costs for Raspberry Pi, ZigBee modules, explorer boards for the ZigBee modules, Arduinos, among other devices. The full cost breakdown can be found in Table 1 below.

Table 1. Project Prototype Budget

|  |  |  |
| --- | --- | --- |
| Description | Debit | Balance |
| Beginning Balance |  | $500.00 |
| RC Car Kits | 398.00 | $102.00 |
| Zigbee S2C Module | 20.00 | $82.00 |
| Infrared Obstacle Module | 11.00 | $71.00 |
| RFID Scanner Module | 5.00 | $66.00 |

## 

## 

## 2.3 Work Breakdown Schedule

This work breakdown schedule in Table 2, below, outlines the work to be done within Phase 0.5 and Phase 1.0 of this project. Any time spent working on subsequent phases is not planned.

Table 2. Project Work Breakdown Schedule

|  |  |  |
| --- | --- | --- |
| Level 1: | | |
| 1 | Traffic IoT System | 100% |

|  |  |  |
| --- | --- | --- |
| Level 2: | | |
| 2 | Traffic IoT System | 100% |
| 2.1 | PPFS | 25% |
| 2.2 | Traffic Light Device | 20% |
| 2.3 | In-Car Device | 20% |
| 2.4 | Integration | 20% |
| 2.5 | Final Report | 10% |
| 2.6 | Project Management | 5% |

|  |  |  |  |
| --- | --- | --- | --- |
| Level 3: | | | |
| **3** | **Traffic IoT System** | **100%** |  |
| **3.1** | **PPFS** | **25%** | Completed |
| **3.2** | **Traffic Light Device** | **20%** | Completed |
| 3.2.1 | Configure Bluetooth on RPi | 5% |  |
| 3.2.2 | Connect RPi to line-following cars | 5% |  |
| 3.2.3 | Control movement of cars with RPi | 10% |  |
| **3.3** | **In-Car Device** | **20%** | Completed |
| 3.3.1 | Find and order line-following cars | 5% |  |
| 3.3.2 | Build cars and figure-8 track | 5% |  |
| 3.3.3 | Debug line-following functionality | 10% |  |
| **3.4** | **Integration** | **20%** | Completed |
| 3.4.1 | Write state machine application to control traffic light and line-following cars | 20% |  |
| **3.5** | **Final Report** | **10%** |  |
| 3.5.1 | Write final report draft | 8% | Completed |
| 3.5.2 | Edit report for final submission | 2% | In-Progress |
| **3.6** | **Project Management** | **5%** | Meetings and planning. |

## 

## 2.4 Team Organization

Table 3. Team Organization

|  |  |
| --- | --- |
| **Team Member** | **Responsibilities** |
| Drew Smits | Systemwide architecture development. Integration and debugging of the line following cars and the traffic light device. Testing of the final prototype. |
| Hamilton Mutschler | Research of various communication technologies. Traffic light device architecture development. Preliminary development of Bluetooth connection and data transfer. |
| Kevin Um | Organization of team meetings and deadlines. Line-following car architecture development. Preliminary debugging of line-following capabilities. |
| Daniel Ackuaku | Budgeting and resource management. Prototype design and development. Built line-following cars and figure-8 track. |
| **Advising/Consulting** | **Role** |
| Eric Walstra | Industrial Consultant |
| Omar Waller | Industrial Consultant |
| Mark Michmerhuizen | Team Advisor |

This team organization for this project worked well for us. Rather than electing a project manager, we divided both the project management and the tasks up early in the second semester once we had finalized our design for the prototype and restructured our project. Some members had more of a technical role and some had more of a project management role based on personal strengths and, unfortunately, COVID-19. These roles are elaborated on in Table 3, above.

# 

# 3. Design Process

## 3.1 Design Norms

**Openness and Communication:**

The aim of this project is to design a communication network that would improve the traffic throughput of intersections and the safety of motorists by communicating over a ZigBee communication protocol and relaying information that can be used by the driver and traffic system infrastructure to alter traffic flow.

With respect to the openness portion of this project, the safety and security of this communication protocol in integral devices is of paramount importance. The basic functionality of the project would be made readily available to the general public. However, the technical particulars of the project would be guarded more closely for security and safety reasons.

**Trust:**

In addition, the scope of the project includes creating and maintaining communication between vehicles and traffic infrastructure. The project does not encompass actual traffic flow and the algorithms that would be required, but rather develops a tool that can be paired with a traffic flow control algorithm to improve traffic throughput and safety.

Another important aspect of this design project is gaining the trust of the public. The project prototype would be twofold, first a full computer model of a 2x2 traffic grid, highlighting the gains in traffic throughput and second with RC cars and a traffic infrastructure in a controlled environment showcasing the safety aspect of the project. The prototype would showcase the traffic throughput improvements as well as the safety aspect of the system by preventing a head-on collision of two vehicles. This coupled with open communication with the public would be the key to developing their trust in the systems being proposed.

**Cultural Appropriateness:**

The proposed solution is one that is culturally appropriate for a couple of reasons. Firstly, the interfaces being designed for both the traffic infrastructure and cars are being designed for aftermarket additions to existing cars and traffic infrastructure. This eradicates the need for new devices to be developed with this addition only.

The project design will be engineered such that it will be identical to existing traffic infrastructure in functionality and aesthetics. The communication modules would fit inside the traffic lights and cars inconspicuously.

**Stewardship:**

The final design norm considered was stewardship. This was tied to the team Bible verse Ephesians 2:10 which reads as follows “ For we are his workmanship, created in Christ Jesus for good works, which God prepared beforehand that we should walk in them”. This scripture implores us as God's workmanship, to follow the calling of performing good works. In this case, the good works are planning and creating this communication protocol.

By improving traffic efficiency, the project also ensures that carbon emissions from vehicles will be reduced because vehicles would spend less time at stoplights and signs and there would be a lower carbon footprint for areas where this module is in use.

## 3.2 Research

The critical design decision for this project is choosing a communication technology. This project falls into the category of the Internet of Things, specifically a sensor network. In the eventual final product, the in-car end nodes will be communicating GPS data to the traffic infrastructure coordinator nodes directly or via a repeater node. The requirements for the devices that apply to the communication technology are as follows: the devices need to be low-power so that they do not draw excess current from the car or the traffic infrastructure, the devices need to have a low connection latency so that the devices can connect and transfer data before a car passes through the intersection, the devices need to be configurable in a mesh topology so information from the traffic infrastructure can be relayed to in-car devices that may not be directly connected to the infrastructure. Specifications for these requirements can be seen in Table 6.

In the sensor network space, there are four popular communication technologies: Bluetooth, ZigBee, Wi-Fi, and cellular. Wi-Fi can be eliminated right away because its connection latency is far too large as fifteen percent of established Wi-Fi connections have a connection set-up time of over five seconds (Meng). Cellular can also be eliminated immediately due to its high cost. Implementing cellular communication would require a data plan with a cellular provider which is not feasible for the budget of this project. Other emerging communication technologies, such as SigFox and LoRaWAN, were not highly researched because these technologies are aimed more towards long-range data transfer on the order of kilometers while this project focused on shorter ranges on the order of hundreds of meters.

That leaves Bluetooth and ZigBee as the two options for the communication of data in this project. Both protocols implement a mesh network topology, but the implementation for each is slightly different. ZigBee uses a coordinator, repeater, and end-device system as seen in Figure 1 (Marlaric).

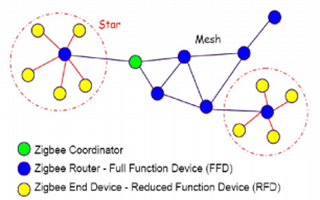


Figure 1. ZigBee Network Topology (Malaric)

Bluetooth follows a publisher and subscriber system. Nodes can subscribe to more than one address and only publish to one address, as seen in Figure 2, below. Communication in this mesh network is achieved by flooding. This means that each node repeats incoming messages to increase the effective range until the destination node is reached (Baert).

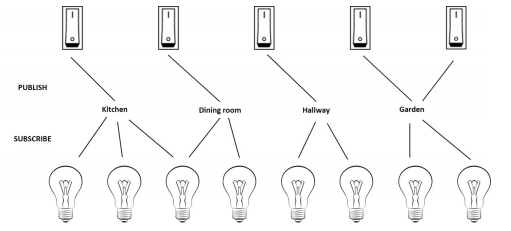


Figure 2. Bluetooth Mesh Network Topology (Baert)

Table 4. Specification comparison between Bluetooth 5 and ZigBee (Collotta).

|  |  |  |
| --- | --- | --- |
| Feature | Bluetooth 5 | ZigBee |
| Radio Frequency (MHz) | 2400-2483.5 | 868.3, 902 – 928, 2400-2483.5 |
| Range (meters) | Up to 200 | Up to 150 |
| Latency (ms) | < 3 | < 4 |
| Network Topology | Star-Bus, Mesh | Mesh |

According to the data in Table 4, Bluetooth 5 has slightly better specifications than ZigBee. What is not listed is the current draw for each as this will depend on the actual device used in the project, but there are devices available for both protocols that meet the specified requirement. According to this data, both Bluetooth and ZigBee meet the requirements set for the communication technology used in this project and the decision between the two will be elaborated upon in Section 3.3.

## 

## 3.3 Design Decisions

### 3.3.1 Design Criteria

The communication technology implemented in this system should meet the requirements set in Section 2.2. The communication between devices shall be effective up to 150 meters with a mesh network topology and the communication latency shall be less than 50 ms. These values were based on the research done by Collotta, Pau, Talty, and Tonguz at Carnegie Mellon University. These requirements are based on the design norm of openness and communication. Designing these devices with these requirements will ensure that the communication between them is reliable and accurate. With reliable and accurate communication the public will be able to trust that this system will indeed improve traffic flow and safety. The other requirement for the communication technology is the requirement for low power, drawing less than 55 mA during transmit and 35 mA while idle will help it comply with the design norm of stewardship. These values were determined by comparing modules for both Bluetooth and ZigBee. Designing low-power devices will allow them to consume less electricity which will help automakers and municipalities implement these devices without concern.

### 3.3.2 Design Alternatives

There are four main communication technologies that could be used to implement this proposed system: Wi-Fi, cellular, Bluetooth, and ZigBee. As stated in Section 3.2, Wi-Fi and cellular are not feasible alternatives for this project, but Bluetooth and ZigBee are both viable options.

The decision matrix in Table 5 was used to determine between Bluetooth and ZigBee. Ease of use is weighted as the most important factor. This category is based on the topologies of the mesh networks: ZigBee’s coordinator, router, and end device structure and Bluetooth’s publisher and subscriber device structure. Cost and range are tied in second. Cost is weighted highly due to the amount of interfaces necessary for the system to work. If the cost of an interface is too high, it would be very difficult to convince automobile manufacturers and municipalities to implement the T2V system. Range is weighted equally high because of its importance in allowing time for transmitting the cars data to the traffic light before it passes through the intersection. The next highest weighted criterion is connection latency. This is important for the same reason as range but is slightly less crucial. The number of connected nodes and network speed are also important for a similar reason, but they have a lower priority due to the small amount of data expected from each connection.

Table 5. Decision Matrix for Communication Medium

|  |  |  |  |
| --- | --- | --- | --- |
|  | Weight | ZigBee | Bluetooth |
| Cost | 8 | 4 | 6 |
| Latency | 7 | 4 | 5 |
| Network Speed | 3 | 3 | 4 |
| Ease of Use | 10 | 5 | 3 |
| Range | 8 | 3 | 4 |
| Number of Connected Nodes | 5 | 5 | 5 |
| Total |  | 168 | 182 |

The results of the decision matrix showed that Bluetooth was a more feasible option for the T2V communication system. While the differences in performance between Bluetooth and Zigbee were insignificant, the cost to attain each module for the prototype were not. The line-following car kits purchased for our project used Bluetooth modules for vehicle to traffic infrastructure communication. We decided to proceed with Bluetooth to reduce the overall cost of the prototype as Zigbee modules were not supplied. The prices for actual production devices may be more competitive between Bluetooth and ZigBee. This would make the choice more dependent on the other factors for a final production device.

### 3.3.3 Design Choices

Phase 0.5 contained the majority of the planning for the project along with some initial building and development work. The first major decision made was to model the T2V system using four-wheel-drive line-following cars. This idea was presented to us by our industrial consultant Omar Waller. This was chosen over the original idea of using remote-controlled cars to make the prototype’s core functionality, control of traffic flow, easier to demonstrate and highlight the impact this system could have for fully-autonomous vehicles in the future.

Once the line-following cars were purchased, the decision to use Bluetooth communication to issue movement commands to the cars was made. This was due to the fact that the line-following car kits were Arduino-based and came with Bluetooth modules. Choosing Bluetooth over ZigBee for the prototype reduced the both cost and development time significantly as no additional ZigBee modules were necessary to integrate into both the cars and the Raspberry Pi.

For Phase 1, we needed to constrain the project into a prototype that was physically able to demonstrate the functionality of the T2V system and be possible to complete within the time frame of this project. The decision was made to build a Figure-8 shaped track for the line-following cars to travel around and interact with the traffic light at the intersection in the middle. The original 2x2 grid track idea was moved to Phase 1.5 to function as a stretch goal for the project, but once all classes were moved online, it became clear that this stretch goal would not be attainable.

In the final production system for Phase 2, ZigBee will most-likely be used due to its mesh networking architecture. The repeater modules in the ZigBee architecture that will allow the range for T2V to detect incoming traffic to be extended easily. Bluetooth’s architecture does not include a repeater-like node. Though when determining the final prototype, the decision matrix between the two will have different weighting based on the circumstances for implementing a real-world product. It is expected that the cost will be more competitive between Bluetooth and ZigBee chips. The cost of integrating these modules into cars and infrastructure will also play a large role in determining the final communication medium. For the scope of this project, ZigBee will be assumed for the communication medium of the final product.

## 

## 3.4 Requirements

Table 6, below, outlines the requirements for the planned prototype at the end of Phase 1.

Table 6. Requirements for Phase 1.0 Prototype

|  |  |
| --- | --- |
| Requirement: | Test Process: |
| Stoplight interface shall operate in temperatures between -40°F to +85°F (automotive standard). | The stoplight interface will receive and transfer information at -40°F to +85°F. |
| Prototype devices shall be smaller than 1000 cm3 | Verify that devices are installable in their respective locations. |
| Communication modules shall have a range greater than 150 m | Verify, using the datasheet of the given Bluetooth module, that data can be exchanged at that range. |
| Communication technology shall have a latency of less than 50 ms | Verify with device datasheets that latency value is not exceeded. |
| Communication modules shall draw less than 50 mA while transmitting and less than 35 mA while receiving/idle | Verify, using a current meter, that these values are not exceeded during operation. |
| Prototype line-following cars shall operate by following the black line in a white background or white line in black background | Verify that all line-following cars do not deviate from the line at all times. |
| Figure 8 Loop | Create a figure-8 course to allow line-following functionality. Implement traffic light and vehicle interface with RFID to show increased throughput compared to a time-based system. |

## 3.5 Task Specifications and Schedule

The task specifications and schedule for this project outlines the work to be done within Phase 0.5 and Phase 1.0 of this project. Any time spent working on subsequent phases is not planned.

Table 7. Task Specifications and Schedule for Project Phases 0.5 and 1.0

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Description** | **Percentage Complete** | **Estimated Completion Date** | **Actual Completion Date** |
| Raspberry Pi Bluetooth Functionality | 100% | 2/1/20 | 2/1/20 |
| Find and Order Arduino-based Line-Following Cars | 100% | 2/1/20 | 2/1/20 |
| Build Line-Following Cars | 100% | 2/7/20 | 2/7/20 |
| Connect RPi to LF Cars via Bluetooth | 100% | 2/7/20 | 2/7/20 |
| Issue Movement Commands to LF Cars from RPi | 100% | 2/7/20 | 2/12/20 |
| Build Figure-8 Track | 100% | 2/7/20 | 2/10/20 |
| Build Traffic Light Assembly | 100% | 3/20/20 | 3/20/20 |
| Write Application Code to Control Intersection | 100% | 3/20/20 | 3/20/20 |
| Debug and Refine Traffic Light Application | 100% | 4/20/20 | 4/17/20 |
| Debug and Refine Line-Following Functionality | 100% | 4/20/20 | 4/17/20 |
| Full System Test and Debug | 100% | 5/1/20 | 4/30/20 |

## 

## 

## 3.6 System Architecture

The scope of this project calls for multiple different systems: an overall system for the prototype for Phase 1, a software system for the operation of the traffic light, and an overall system for the theoretical final product for Phase 2. Although there are no plans to reach Phase 2 within the scope of this project, it is still valuable to keep a final product in mind while working on this project. Because there are multiple systems involved in this project, there are multiple different architectures that define them.

Starting with the final product for the T2V system, the potential system architecture can be seen in Figure 3, below.



Figure 3. T2V Final Product System Architecture Block Diagram

The final product for T2V will feature ZigBee communication modules, as stated in Section 3.3.3, and GPS modules for determining the positions of cars in an intersection. GPS was not feasible for the small-scale prototype for Phase 1 due to its inability to be accurate on a small scale. The GPS location of cars will be transmitted from the In-Car device to the Traffic Infrastructure directly or indirectly through the ZigBee Repeater module. The Traffic Infrastructure device will be receiving these GPS coordinates from all the different cars within its radius and change the state of the traffic light accordingly.

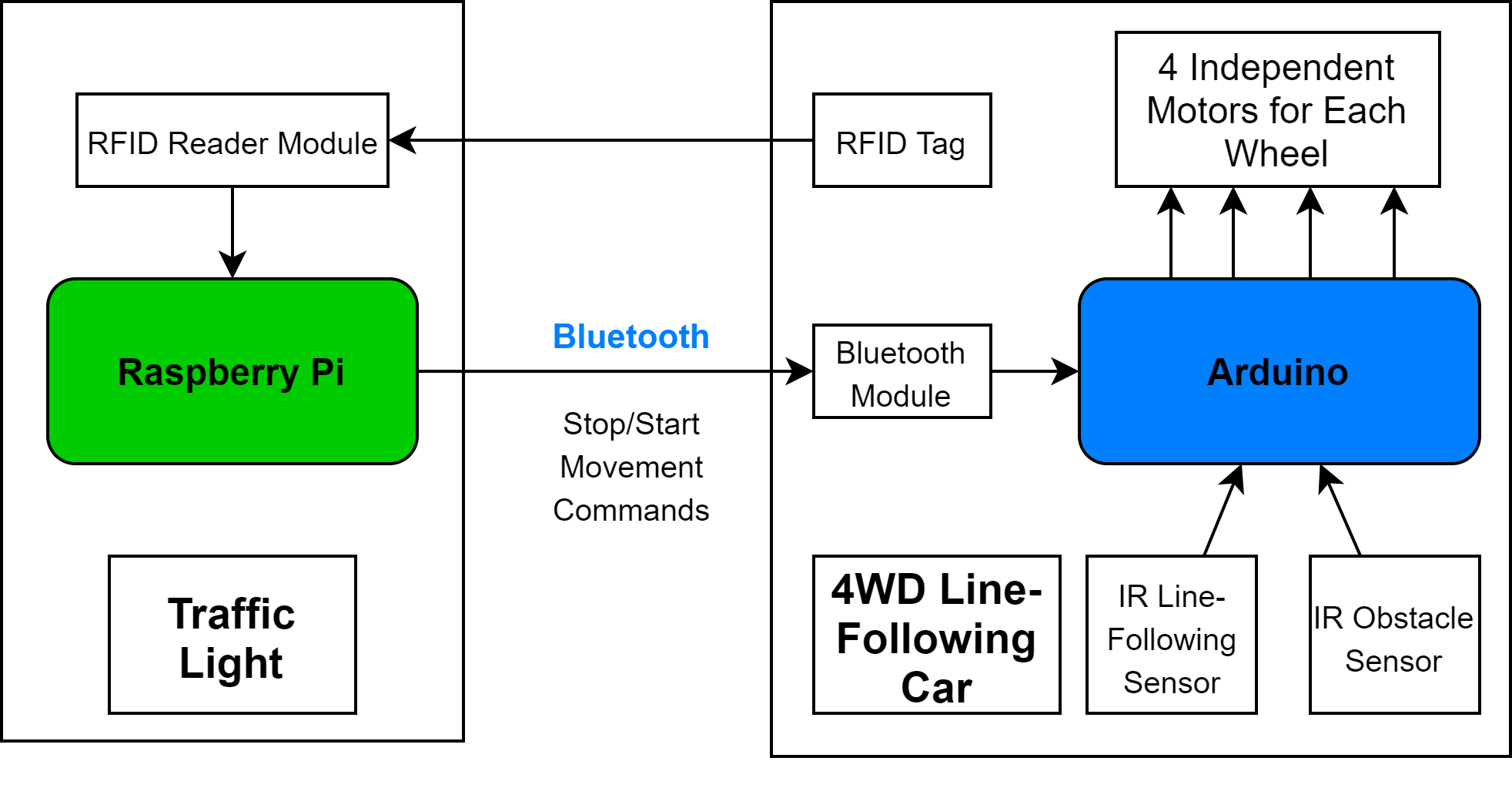


Figure 4. T2V Phase 1 Prototype System Architecture Block Diagram

The architecture of the prototype for T2V can be seen in Figure 4, above. The prototype features a central Traffic Infrastructure node controlled by a Raspberry Pi 3B+ with Bluetooth built in and RFID reader modules connected via jumper wires to the Raspberry Pi’s SPI bus. The RFID modules determine where the line-following cars are in the intersection by sending signals to the Raspberry Pi when a car is underneath the reader. Depending on the readings from the RFID modules, the Raspberry Pi will issue stop and go commands to the cars to control the flow of traffic through the intersection. The 4WD Line-Following Cars are controlled by an Arduino that receives input from four infrared sensors and an external Bluetooth module. The Arduino receives the stop and go commands issued by the Raspberry Pi through Bluetooth and then drives the four independent wheels based on the input from the infrared sensors to follow the line on the track and avoid collisions. The state machine that controls the changing of the traffic light in the intersection can be seen in Figure 5 and Figure 6, below.

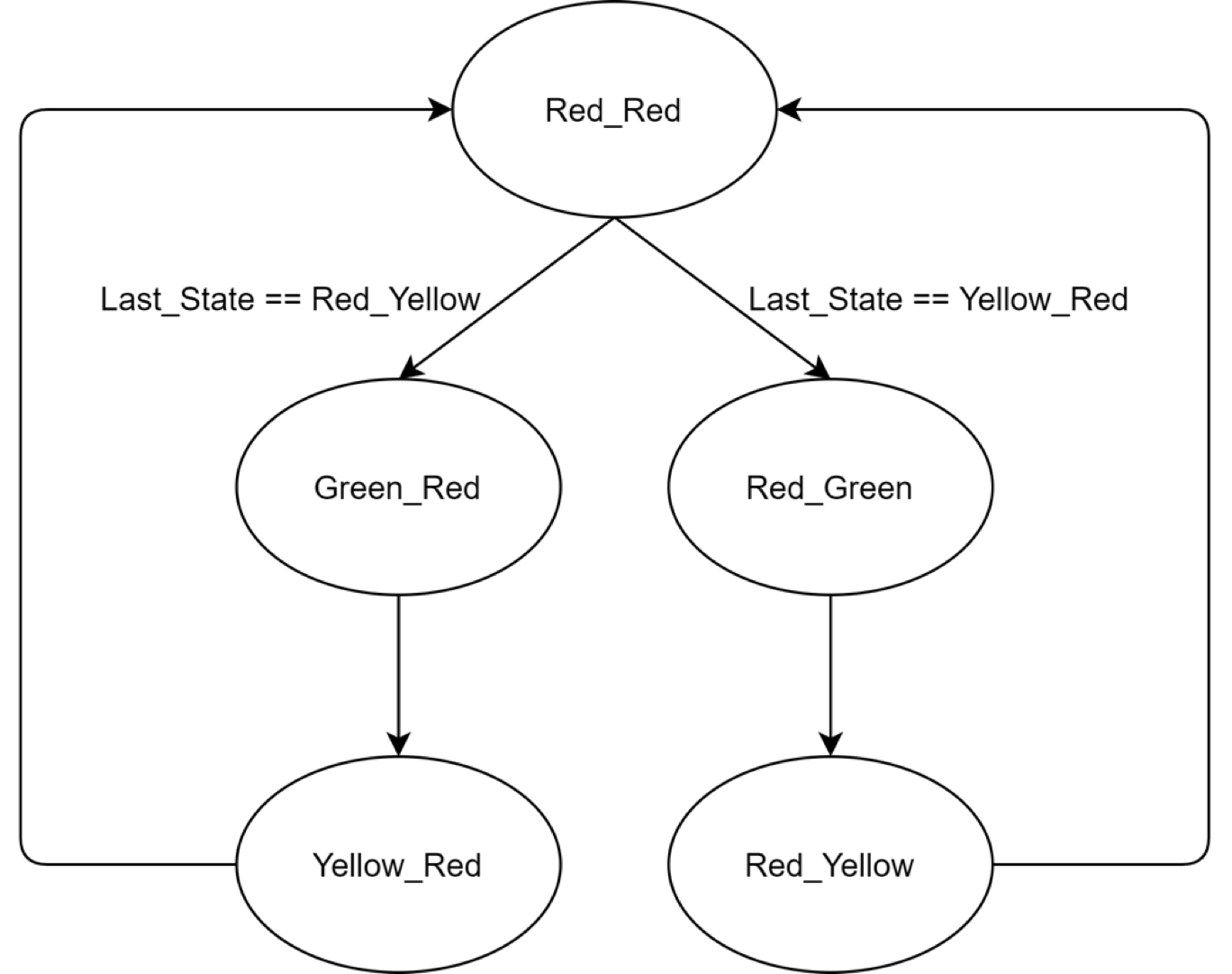


Figure 5. T2V Phase 1 Prototype Software Architecture Block Diagram

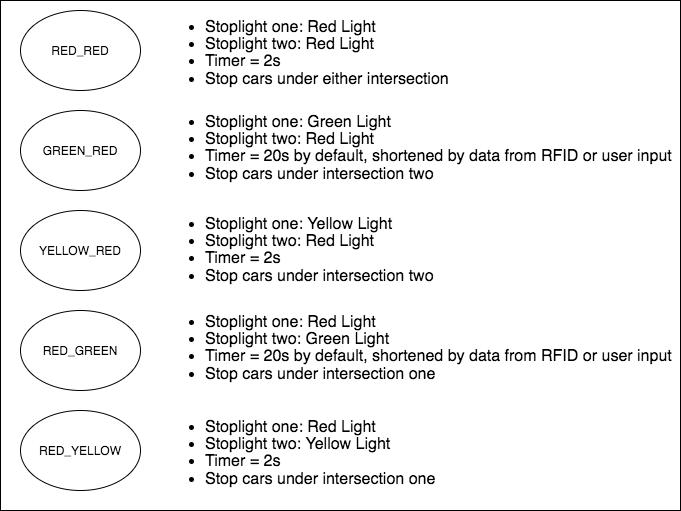


Figure 6. Explanation of States in Traffic Control Software

The state diagram shown above describes the way our stoplights operate. The changing of states is triggered by a combination of timers, RFID readers that can sense when a car is at the light, and by user input sent through MQTT from another Raspberry Pi. The RED\_RED, YELLOW\_RED, and RED\_YELLOW states are all controlled by a timer, changing states after two seconds, while the GREEN\_RED and RED\_GREEN states operate on a timer as well, but for twenty seconds and this time can be shortened by a user or by input from the RFID readers depending on the location of the cars on the track.

## 3.7 Integration, Test, Debug

Owing to the implementation model adopted by the team, some of the tests spelled out in the initial PPFS document that were meant to be carried out on the fully fleshed-out version of the product could not be carried out on Phase 1.5 prototypes. As a result, the testing scheme of this particular portion of the project was altered in order to ensure that the prototype for the traffic system was an adequate representation of the actual model. For the tests that could be carried out on this prototype here are the results.

### 3.7.1 Collision Avoidance

The initial test in this phase of the project was the collision avoidance test. Two RC cars equipped with ultrasonic sensors were set on a crash course. The expected outcome of the programmed system was that the car that sensed the impending collision would have halted, allowing the other RC vehicle right of passage.

It was discovered that the cross-sectional area of the RC car was an important factor in these tests. With the car being constructed of mainly a thin PCB and wheels the ultrasonic sensor was unsuccessful in detecting the oncoming vehicle.

To mitigate this aspect of the test, a reflective shield was constructed out of cardboard and attached to the sides and rear end of the RC car. This reflective shield would offer a suitable surface for the ultrasonic sensor to bounce its detection rays off. This improvement to the design of the cars made it possible for the vehicle to pass the collision avoidance test. This test was conducted using the RC cars equipped with ultrasonic sensors, bluetooth receivers, a bluetooth transmitter and a laptop.

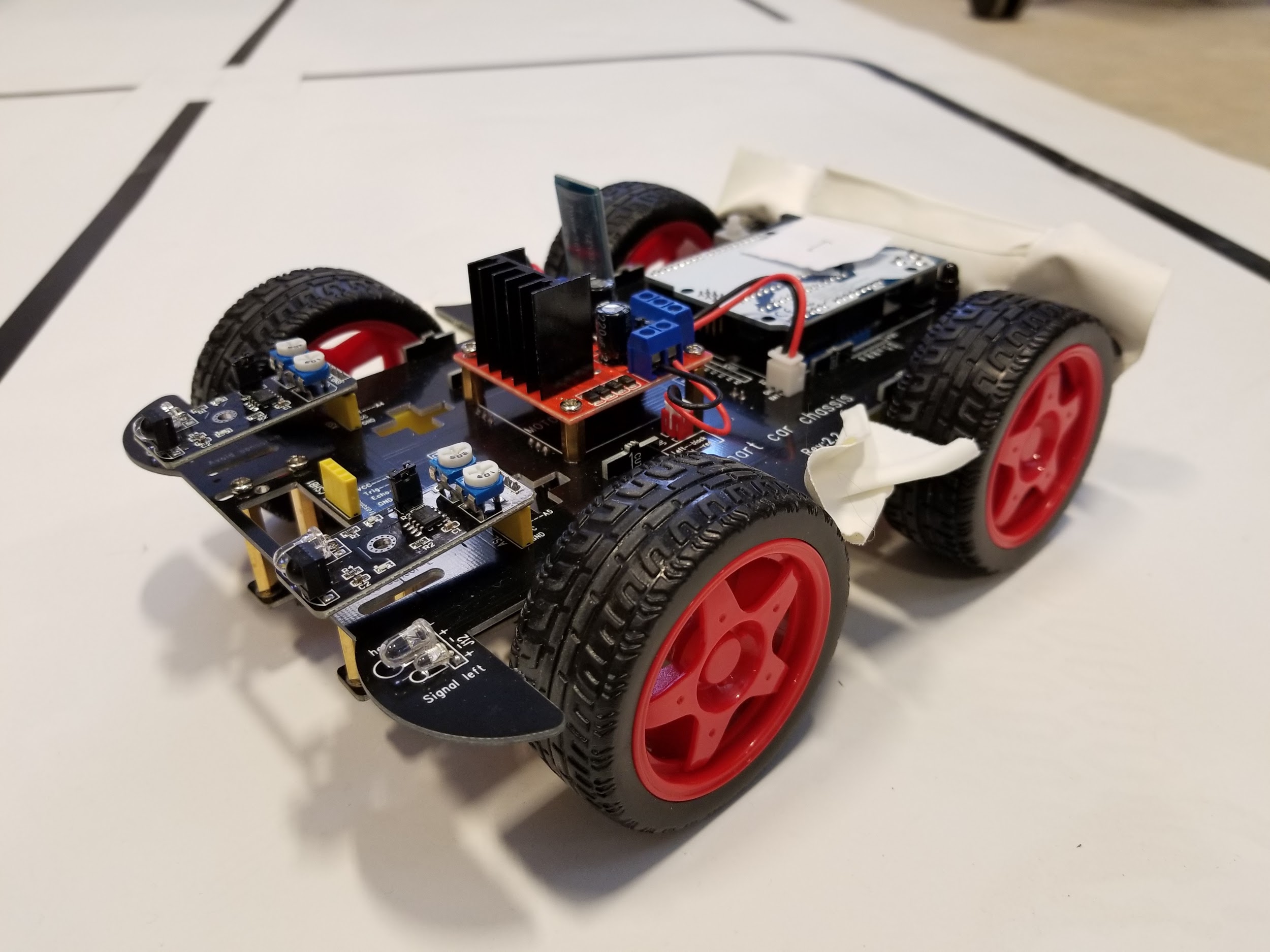


Figure 7. 4WD Line-Following Car with Attached Reflective Paper

### 3.7.2 Track Thickness and Corner Curvature.

The track used to guide these cars was a line following circuit. Each vehicle was programmed to move forward if the two sets of infrared sensors on the underside of the car did not detect a black line. Once a black line was detected the vehicle would turn in the opposite direction to compensate for the presence of the line. The reason for this test was, if the thickness of the black line was less than 0.5 cm the car would move quickly across the track without detecting any of the lines. On the other hand a line thickness of greater than 2.5 cm would prevent the car from moving forward at all because both of the infrared sensors would be reading black lines. A range of line thicknesses were tested and the thickness that maximized both straight line speed and the smoothness of the ride was determined to be 2 cm. With respect to the cornering of the cars on the track, it was determined that a 90 degree turn was too hard for the RC car to execute successfully considering the speed the car was travelling at. It was experimentally determined that the most ideal radius of curvature for the RC cars was an outer radius of 10 cm and an inner radius of 8cm.

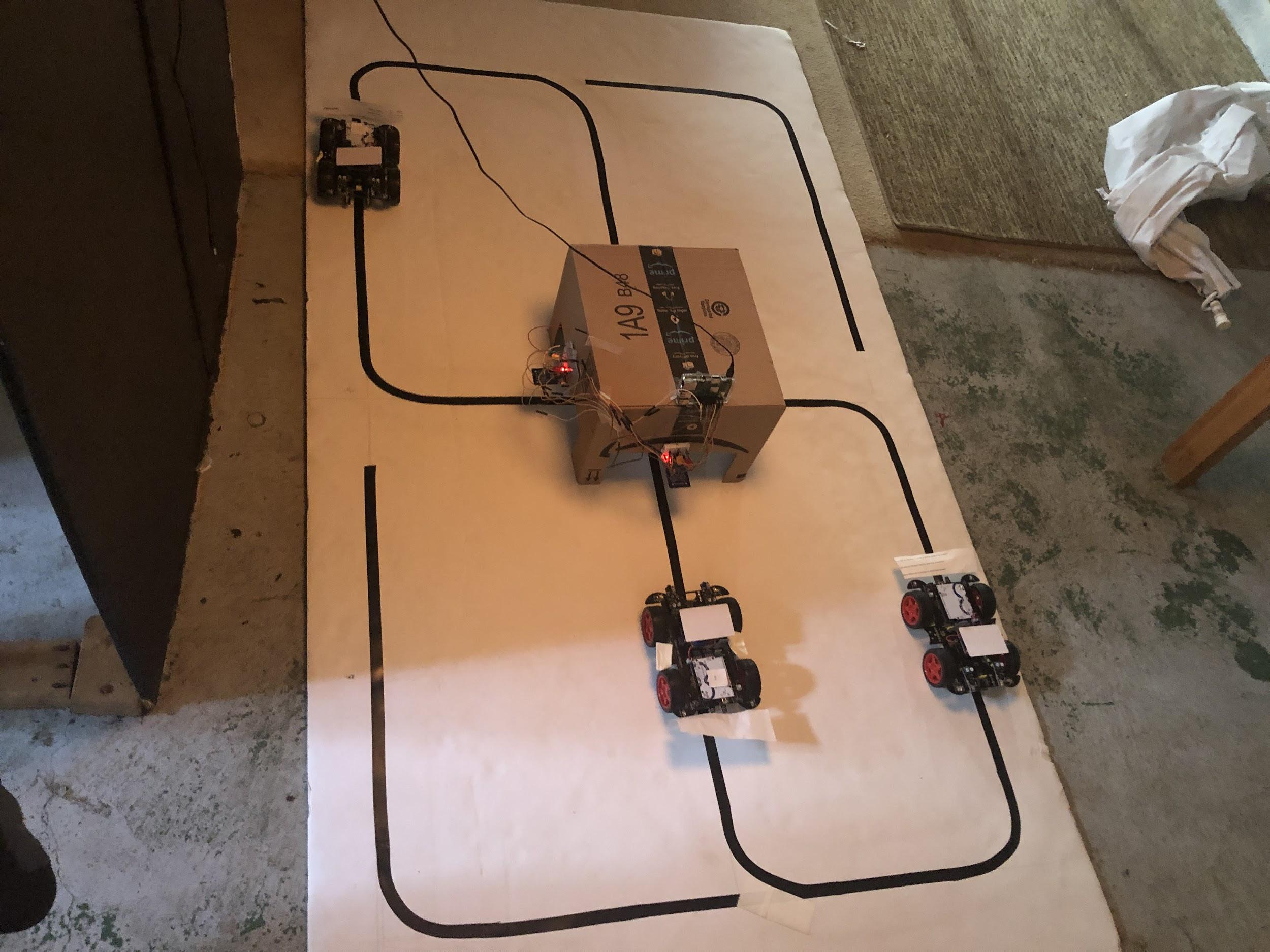


Figure 8. Final Figure-8 Track with Specified Dimensions

### 3.7.3 Traffic Control System

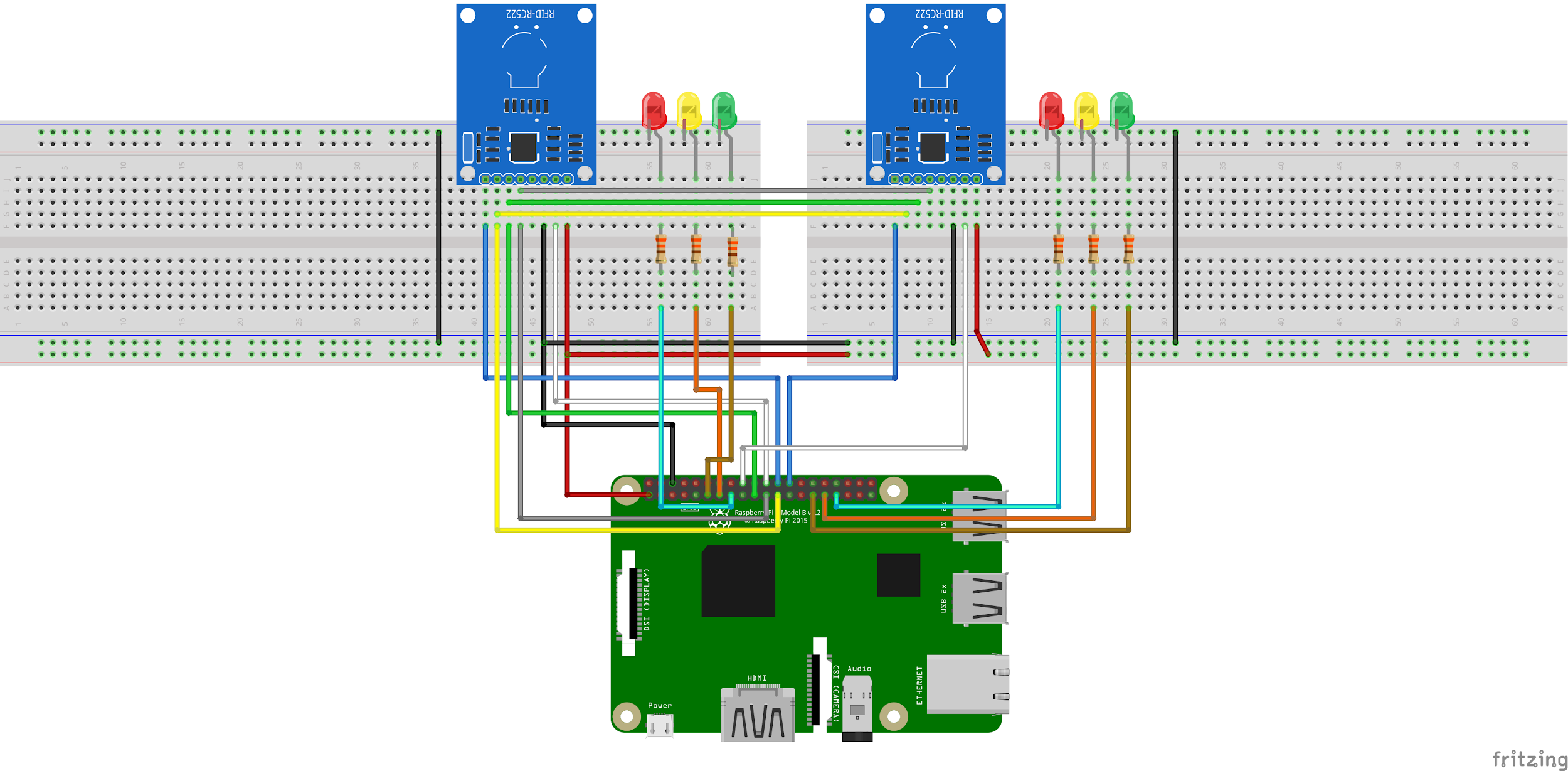


Figure 9. Hardware Diagram for the Traffic Control System

After all the unit tests for different aspects of the system were complete the overall system was assembled and subjected to testing. Every aspect of the traffic flow system of the project was built on test driven implementation. Each additional piece of functionality was only added once the previous piece had passed the test.

* A simple timed stop light was created with three LED lights and ( red, yellow, green) along with a Raspberry pi. A simple timing program was created that caused the lights to turn on and off based on the timing instructions obtained from the raspberry pi.
* The next stage of the test was to ensure that the RFID modules obtained would be read by the RFID tags when the RFID reader was connected to the Raspberry pi.
* The RFID tags were placed on the track at positions where we would term “stoplights” to determine when the RC car passed over the RFID tag it would be stopped. This test was successful.
* The next iteration involved combining the functionality of the RFID reader and the traffic lights with instructions being sent to the cars. The RFID tag embedded in the track delivered a stop signal whereas the Green led light was programmed to deliver a go signal to the cars. This functionality was added to the system with one car and then successfully tested with an additional car.

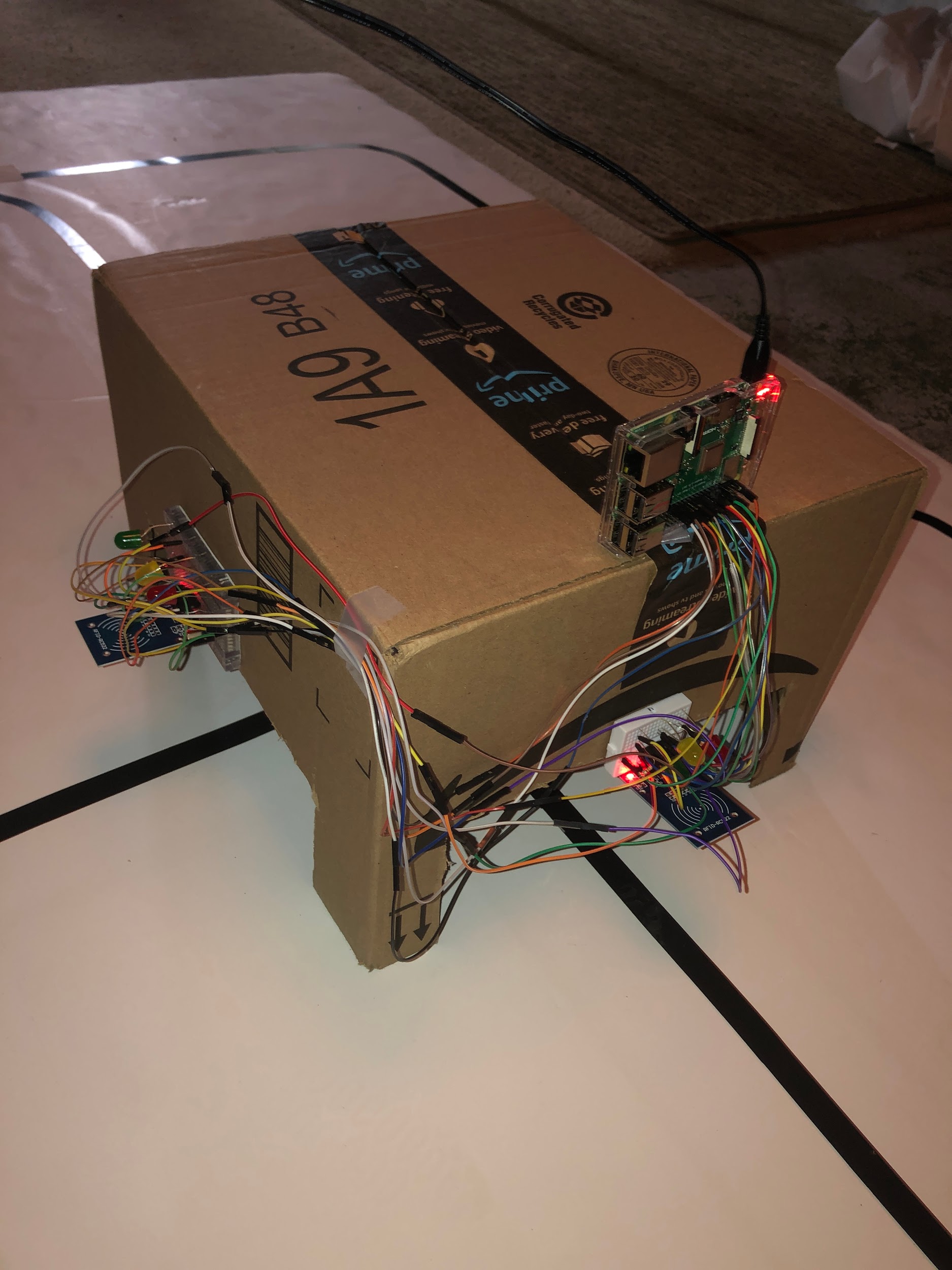


Figure 10. Final Prototype of the Traffic Control System

### 3.7.4. Throughput Testing

We used the prototype model of the intersection to test the throughput of the system in three different modes. The first modelled the behavior of a simple timed stoplight, each green light lasted for ten seconds, and each yellow light and wait period between green lights lasted for two seconds. The second mode modelled the behavior of a timed stoplight with an inductive loop sensor used to change the state of the light. When the RFID readers read that a car was stopped below them at a red light, the program would trigger a change of states to allow that car to pass through the intersection as quickly as possible. The last mode looked at the possible improvements to these conventional intersections if there were a system of sharing GPS and intersection data to make decisions regarding the state of the light. In this mode a user controlled the changing of green lights, acting as a more advanced light controlling algorithm with complete location data to see how the throughput would improve. This mode increased throughput of the intersection the most by reducing the amount of time spent waiting at red lights, but also avoiding excessive changing of states that increased the number of times cars would be stopped at a red light.

## 3.8 Production

As stated initially, the phased implementation approach was adopted for this project, that being said the Production costs for some aspects of the prototype system in Phase 1.5 do not exactly translate into the production costs for the full device. As a result the production costs would be split into two sections, the prototype production cost, and the expected device production cost. Both of these sections would detail the bill of materials required to assemble the complete smart traffic system as prescribed in the project brief. Some of the items listed in the bill of materials were already owned by members of the team and as such even, though they would be considered as a part of the prototype production costs, they did not have any effect on the overall project budget.

### 3.8.1 Prototype Production Budget [[1]](#footnote-0)

#### 3.8.1.1 Phase 0.5

The aim of this part of the project was to test the overall feasibility of the idea. To accomplish this successfully, there needed to be an autonomous vehicle that would simulate the driver in a real-world scenario and there had to be proof that GPS coordinates could be transferred over the Zigbee network. In order for this to be accomplished the following materials were procured.

Table 8. Bill of Materials for Phase 0.5 of the Prototype

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Unit Cost | Quantity | Identification ( Vendor/ URL) |
| Zigbee modules | $ 20.06 | 3 | XB24CDMWIT-001 |
| Zigbee explorer board | $ 25.95 | 2 | 474-WRL-11812 |
| Raspberry Pi (RPi) \* | $ 35.00 | 1 | Adafruit : 3775 |
| RPi camera \* | $ 29.95 | 1 | Adafruit : 3099 |
| RC car \* | $ 29.95 | 1 | B07QLX1854 |
| 9v battery \* | $ 2.49 | 2 | B002UGVWA4 |
| 3500 mAhr Portable Battery \* | $ 15.41 | 1 | B07D48V1Y7 |
| RPi Ultrasonic sensor \* | $ 3.95 | 2 | Adafruit : 4007 |
| GPS Module | $ 11.99 | 1 | B07P8YMVNT |
| Man hours | $ 100 | 40 | T2V |

The Zigbee modules were the chosen hardware devices for data transfer because of their use in existing IoT production that transmitted data over large distances and the promise of it’s mesh network capabilities. For the pilot test two Raspberry pi 3 B’s were selected because of the performance overhead of the 2.4 Ghz microprocessor, easily accessible GPIO pinouts as well as the wifi capabilities which allowed the team to effortlessly connect to the Pi of ssh. For similar reasons an additional pi was used to simulate the first version of the autonomous vehicle tried out in phase one. This vehicle had the proprietary Raspberry Pi camera that allowed a live camera feed to be viewed from the hood of the pi with little latency and an ultrasonic sensor that measured the distance to the object.

The approximate amount of time spent by the team on the design, and development of the phase 0.5 prototype was 40 man-hours which amounts to a direct labor cost of $4000. The overall estimated cost of production for the phase 0.5 prototype was $ 4.247.26.

#### 3.8.1.2 Phase 1

Phase 1.0 was focused on creating a line following robot that could mimic the mannerisms of a human driver, following traffic regulations. The cost of prototype production materials and man-hours can be found in the table below.

Table 9. Bill of Materials for Phase 1.0 of the Prototype

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Unit Cost | Quantity | Identification ( Vendor/ URL) |
| Track Development | $ 20 | 1 | T2V |
| Optical sensors | $ 1.10 | 10 | B01I57HIJ0 |
| Arduino RC car | $ 65.99 | 1 | UA060 |
| Man-hours | $ 100 | 200 | T2V |

For this part of the project, the hardware choices were enforced by the ease of use and availability of off-the-shelf solutions to some of the engineering issues the team was facing. The arduino RC car kit along with the optical sensors were chosen because they formed the best pair that adequately simulated the human driving aspect of the prototype being designed. The code for the Arduino was written in C ( a language most of the team was familiar with ) and thus allowed the team to get the cars assembled and move on to creating the traffic monitoring system.

The approximate amount of time spent by the team on the design, and development of the phase 1.0 prototype was 200 man-hours which amounts to a direct labor cost of $20000. The overall production cost for this phase amounts to $20,097.09

#### 3.8.1.3 Phase 1.5

This portion of the project focused on scaling up in size and number of the prototypes with the following items added to the bill of materials.

Table 10. Bill of Materials for Phase 1.5 of the Prototype

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Unit Cost | Quantity | Identification ( Vendor/ URL) |
| RFID Module | $ 3.99 | 3 | RC522 |
| Arduino RC car | $ 65.99 | 3 | UA060 |
| Man hours | $ 100 | 350 | T2V |

In addition to the hardware used in Phase 1, the team procured two RFID readers and four RFID chips. The combination of the readers and chips were chosen because they worked well with both the Arduino platforms and that of the Raspberry Pi. The RFID readers had a sensing distance of five centimeters which made them perfect for use as a part of the traffic system to detect the passing car.

The approximate amount of time spent by the team on the design, and development of the phase 1.5 prototype was 100 man-hours which amounts to a direct labor cost of $35,209.94.

### 3.8.2. Estimated Market Value

Seeing as though the full-size prototype has not yet been developed it would be difficult to adequately estimate the price range within which the device in question would fall within. That notwithstanding, looking at market trends for similar aftermarket GPS systems with integrated infotainment, there seems to be a market equilibrium at the price of $200. Some car manufacturers offer systems with more features and higher prices but for the purposes of our prototype design and costing analysis the team decided the best current market comparison to our product is the aftermarket GPS system.

### 3.8.3. Per Unit Annual Cost

As our project focused on developing a functional prototype which demonstrated traffic efficiency and safety, a full-size production unit was not in the scope of the project’s timeline due to unforeseen delays. Therefore, we do not have a per unit annual cost figure for a production unit as our project did not create one. That being said, based on a raw tally of the cost of the bill of materials and the engineering design and development, the team came up with the following market projection:

Assuming the device being sold is the current Phase 1.5 prototype that has been developed this past semester.

Per the production cost calculations for Phase 1.5 outlined in Table 4.6, the total engineering and development cost stands at an estimated value of $20,000. Considering the benefits of economies of scale that the team would encounter once production and sales pick up, the estimated competitive price for the production was deemed to be $75.00. In 2018 there were an estimated 250,000,000 registered vehicles in the United States (Statista Research Department). Supposing one percent of the car market falls in the bracket of early adopters, this prospective price would allow the team to devote more resources to research and development while maintaining a quality product.

# 

# 4. Conclusion

The project undertaken by Team 13, T2V, in determining the feasibility of a vehicle to vehicle and vehicle to infrastructure traffic communication network was largely deemed a success. Through research and testing, we have determined that Bluetooth was the best communication technology for making a working prototype for our project. The system was implemented following this four phase plan:

**Phase 0.5:** Propose a feasible project and design a feasible prototype. Outline future phases of the project. This phase is intended to act as a planning phase for the scope of the project.

**Phase 1.0:** Develop and build a fully operational prototype that demonstrates the feasibility of the final product. This phase is intended to act as an achievable goal for the scope of this project.

**Phase 1.5:** Scale-up the prototype to include more advanced functionality. This phase is intended to act as a stretch goal for the scope of this project.

**Phase 2.0:** Develop and build a fully operational final product. This phase is intended to act as an eventual final stage of development for a final product for production. Realistically this could not be achieved within the scope of this project but is important to outline for the sake of developing project management skills.

Phases 0.5 and 1.0 were successfully completed during the span of this project. A complete traffic flow system was designed with autonomous vehicles that represented the driver, and a traffic system consisting of LED lights, a RFID reader and tag, and a Raspberry Pi that controlled the timing aspects of the system. This system successfully demonstrated increased traffic throughput through user control of the state of the lights. This was intended to be implemented through additional RFID monitoring throughout the track but due to the difficulty of acquiring parts, the user control was used to demonstrate this instead.

Phase 1.5 and the 2x2 track implementation was originally intended to be a stretch goal for the project, but was ultimately cancelled due to the transition to online work.

These accomplishments prove that we have successfully completed our senior design project by May 2020.

# 

# 5. References/Bibliography

Baert, Mathias, et al. “The Bluetooth Mesh Standard: An Overview and Experimental

Evaluation.” *Sensors*, vol. 18, no. 8, ser. 2409, 25 July 2018. *2409*.

Collotta, M., et al. “Bluetooth 5: A Concrete Step Forward toward the IoT.” *IEEE*

*Communications Magazine*, July 2018, pp. 125–131.

*How Does a Traffic Light Detect That a Car Has Pulled up and Is Waiting for the Light to*

*Change?* HowStuffWorks, 1 Apr. 2000,

https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/how-does-a-traffic-light-detect-that-a-car-has-pulled-up-and-is-waiting-for-the-light-to-change.htm.

Malaric, Kresimir, et al. “ZigBee wireless standard.” *48th International Symposium*

*ELMAR-2006*, Zadar, Croatia, June 2006, pp. 259–262.

Meng Y. *et al*., "Why it takes so long to connect to a WiFi access point," *IEEE INFOCOM 2017*

*- IEEE Conference on Computer Communications*, Atlanta, GA, 2017, pp. 1-9.

doi: 10.1109/INFOCOM.2017.8057164

Sisiopiku, Virginia. “Assessment of Red Light Running Camera Enforcement Technologies.”

*DOT ITS Knowledge Resources*, US DOT, Jan. 2002, www.itscosts.its.dot.gov/ITS/benecost.nsf/ID/2B209AD2C5AD2AB985256DB10045892B?OpenDocument&Query=CApp.

“T-Bone Accident Statistics And Other Legal Information.” *Davis Law Group, P.S.*,

https://www.injurytriallawyer.com/faqs/how-dangerous-are-t-bone-accidents-.cfm.

“Unit Cost Entries for Inductive Loop Surveillance at Intersection.” *Inductive Loop Surveillance*

*at Intersection Unit Cost Entries*, US Department of Transportation, https://www.itscosts.its.dot.gov/its/benecost.nsf/DisplayRUCByUnitCostElementUnadjusted?ReadForm&UnitCostElement=Inductive+Loop+Surveillance+at+Intersection+&Subsystem=Roadside+Detection+(RS-D).

“Number of motor vehicles registered in the United States from 1990 to 2018.” Statista Research   
 Department, https://www.statista.com/statistics/183505/number-of-vehicles-in-the-

united-states-since-1990/

# 6. Appendix

All code for all code used in this project can be found at the following link:

<https://github.com/hmutschler15/rpi_git/tree/master/engr/340/Final_Software>

1. \* These are the parts of the project already owned by team 13. [↑](#footnote-ref-0)