**Conclusion**

* The proposed solution: autonomous system that enforces traffic rules w/out driver intervention
* Current solutions are not effective, need driver intervention
* After making changes (due to money restrictions) to the original design, all tests passed 100% of the time.
* In the future
  + Use BLE instead of regular BT, closer to real life situation
  + BT chip that can receive data from multiple BT devices.

**Engineering Goal**

“The design and construction of an automotive system that will automatically enforce special driving zone requirements like speed, noise, and light levels.”

* Places near areas like hospitals might require certain traffic rules to take effect near them (no headlights, reduced volume, no honking, etc.)
* Currently, there is no way to enforce these requirements should the driver fail to adhere to the stated limits.

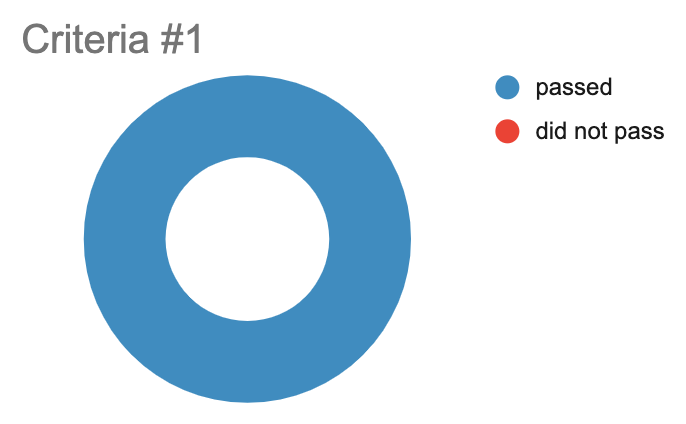
**Background Research**

* **BLE**
  + Uses less power than regular BT
    - Remains asleep between connections, connects for only a few seconds to transfer data
  + Ideal for transferring small amounts of data
    - Some use 100x less power than regular BT
    - However, does not have many other features BT has
* **RFID**
  + Transmitter uses radio waves to identify unique tag
  + Active RFID
    - Tag has power source, such as battery
  + Passive RFID
    - Electromagnetic wave wakes up antenna on tag, which wakes up circuit on tag
    - Tag sends back coded message
  + RFID technology used by FasTrak
    - FasTrak - used to electronically pay tolls on bridges and express lanes
    - Uses RFID technology to identify car
      * Uses tag # to charge money to that car

**References**

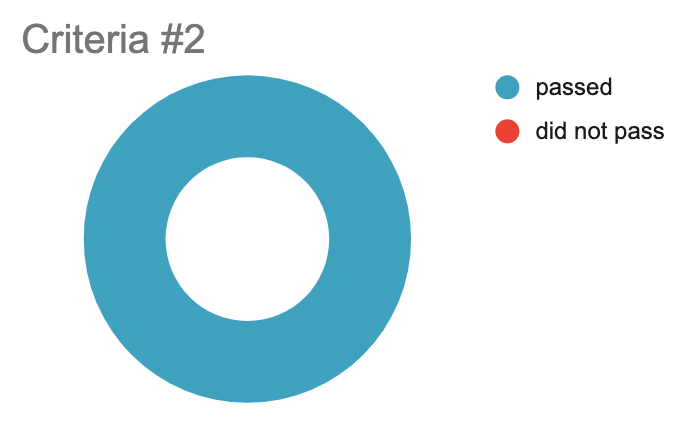
* Amsler, Sarah, and Sharon Shea. “What Is RFID and How Does It Work?” *IoT Agenda*, TechTarget, 31 Mar. 2021, <https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification> Accessed December 16, 2022.
* Currey, Martyn. “Connecting 2 Arduinos by Bluetooth Using a HC-05 and a HC-06: Pair, Bind, and Link.” *Martyn Currey*, Martyn Currey, 25 Nov. 2014, <http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc-06-pair-bind-and-link/>. Accessed January 11, 2022.
* Electronics Tech, director. *Make a Communication between HC-05 and HC-06 Bluetooth Modules with Arduino*. *YouTube*, YouTube, 10 Aug. 2018, <https://www.youtube.com/watch?v=Y2uwyUNt9ZM>. Accessed January 3, 2023.
* Nesbo, Elliot. “What Is Ble (Bluetooth Low Energy) and How Does It Work?” *MUO*, MUO, 27 Dec. 2021, <https://www.makeuseof.com/what-is-ble-bluetooth-low-energy/>. Accessed December 16, 2022
* Santos, Rui. “MFRC522 RFID Reader with Arduino Tutorial.” *Random Nerd Tutorials*, Random Nerd Tutorials, 2 Apr. 2019, <https://randomnerdtutorials.com/security-access-using-mfrc522-rfid-reader-with-arduino/>. Accessed January 5, 2023

**Criteria #1:** Receive instructions in given distance

****

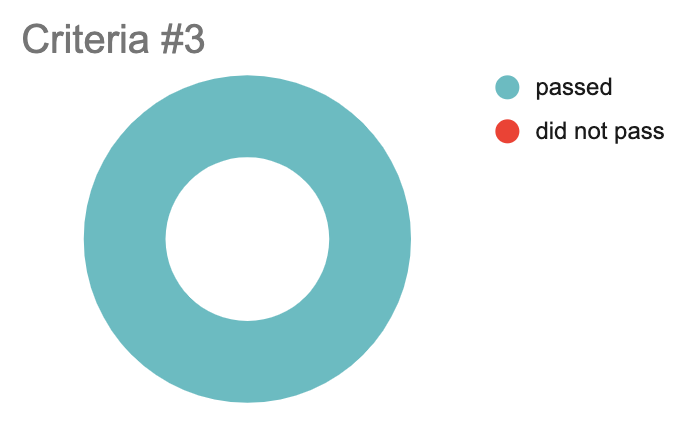
Out of 10 trials, the car could receive instructions 100% of the time.

**Criteria #2:** Use instructions to correctly disable features

****

Out of 10 trials, the car could disable features 100% of the time.

**Criteria #3:** Correctly re-enable previously disabled features

****

Out of 10 trials, the car could enable features 100% of the time.

**Results / Findings**

**Criteria #1:** Must be able to receive instructions from the given distance 90% of the time

The prototype car was able to receive instructions from the given distance (2 cm) in all **10 out of the 10 trials** conducted

\*Due to unforeseen money restrictions, the range had to be reduced from the original 30 cm to 2 cm for testing.

**Criteria #2:** Must be able to use instructions to correctly disable feature(s) 90% of the time

The prototype car was able to correctly disable features based off of the instructions received in **all 10 out of the 10 trials** conducted.

**Criteria #3:** When the car is out of a special zone, the system must be able to correctly enable all previously disabled feature(s) 90% of the time.

The prototype car was able to correctly enable previously disabled features in **all 10 out of the 10 trials** conducted.

**Testing Procedures**

Criteria #1 - Must be able to receive instructions in the given distance 90% of the time

1. Run program on master arduino. The program on master arduino will send out information.
2. Run program on slave arduino (on prototype car). The program on slave arduino will recieve information and display by turning on an LED.
3. Put slave arduino on a steady surface.
4. Put master arduino completely over the slave arduino, so that they are touching.
5. Slowly raise master arduino above the slave arduino, until the given distance (30 centimeters) is reached - measure with ruler
6. Record results - If LED is still on after the given distance (30 cm) is reached, trial is a success.
7. Stop program on master arduino and slave arduino
8. Repeat steps 1-6 ten times (10 trials)

Criteria #2 - Must be able to use instructions to correctly disable feature(s) 90% of the time

1. Put slave arduino on a steady surface, and put master arduino 25 cm directly above from the slave arduino
2. Run program on master arduino. The program on master arduino will send out information.
3. Run program on slave arduino (on prototype car). The program on slave arduino will recieve information and display by turning **off** a turned **on** LED.
4. If slave turns **off** the LED (on the prototype car) within 2 seconds, the trial is a success
5. Record results
6. Stop program on master arduino and slave arduino
7. Repeat steps 2-6 ten times (10 trials)
8. Put slave arduino on a steady surface, and put master arduino 25 cm directly above from the slave arduino
9. Run program on master arduino. The program on master arduino will send out information.
10. Run program on slave arduino (on prototype car). The program on slave arduino will receive information and display by turning **off** a sound playing on a speaker (speaker is mounted on prototype car).
11. If slave turns off speaker **on** the prototype car within 2 seconds, the trial is a success
12. Record results
13. Stop program on master arduino and slave arduino
14. Repeat steps 8-12 ten times (10 trials)

Criteria #3 - When the car is out of a special zone, the system must be able to correctly enable all previously disabled features 90% of the time.

1. Put slave arduino on a steady surface, and put master arduino 25 cm directly above from the slave arduino
2. Run program on master arduino. The program on master arduino will send out information.
3. Run program on slave arduino (on prototype car). The program on slave arduino will receive information and display by turning **on** a turned **off** LED.
4. If slave turns **on** the LED (on the prototype car) within 2 seconds, the trial is a success
5. Record results
6. Stop program on master arduino and slave arduino
7. Repeat steps 2-6 ten times (10 trials)
8. Put slave arduino on a steady surface, and put master arduino 25 cm directly above from the slave arduino
9. Run program on master arduino. The program on master arduino will send out information.
10. Run program on slave arduino (on prototype car). The program on slave arduino will receive information and display by playing a sound **on** a previously turned off speaker (speaker is mounted on prototype car).
11. If slave turns **on** speaker on the prototype car within 2 seconds, the trial is a success
12. Record results
13. Stop program on master arduino and slave arduino
14. Repeat steps 8-12 ten times (10 trials)

**Materials**

For each car (2 cars)

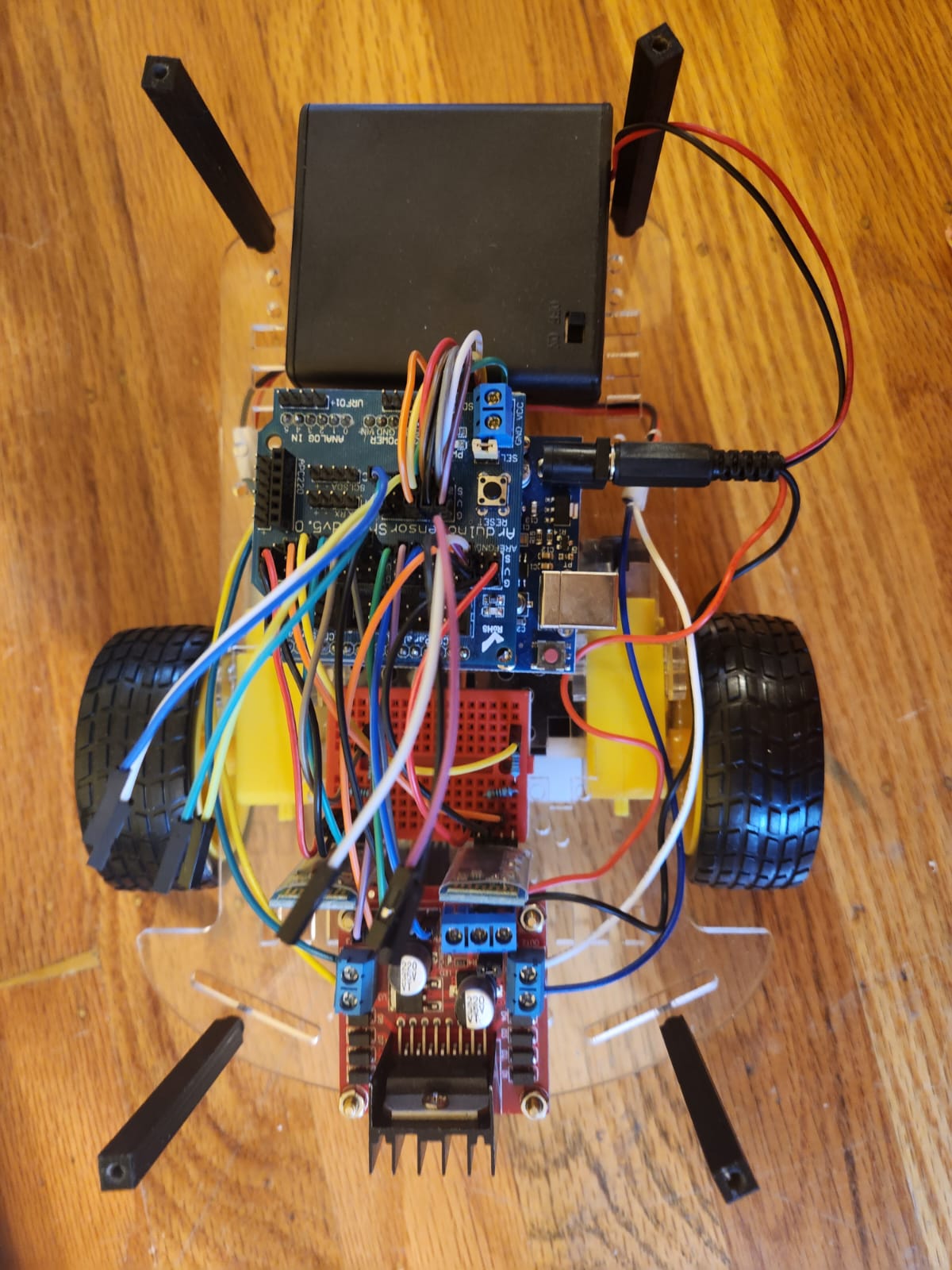
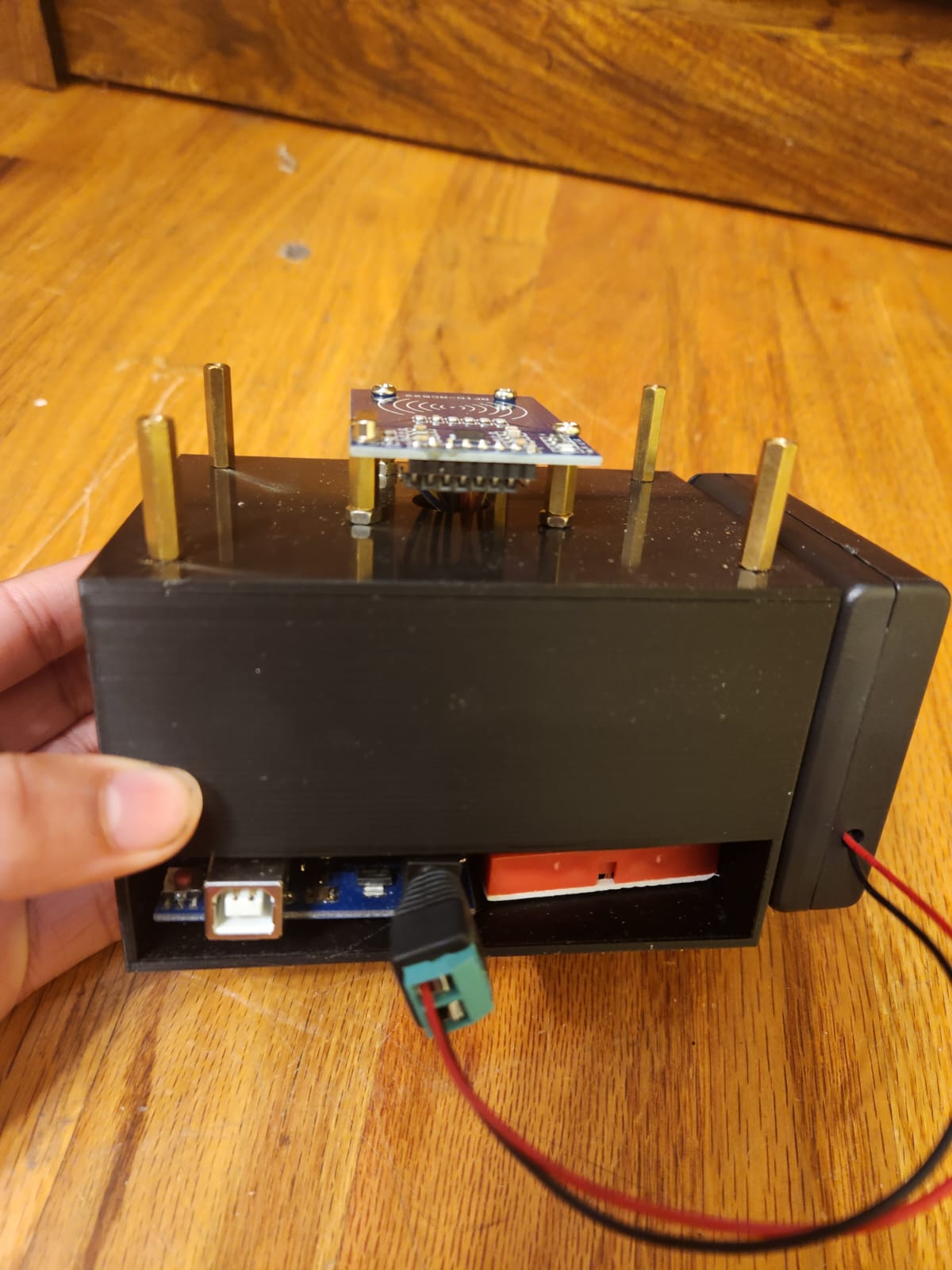
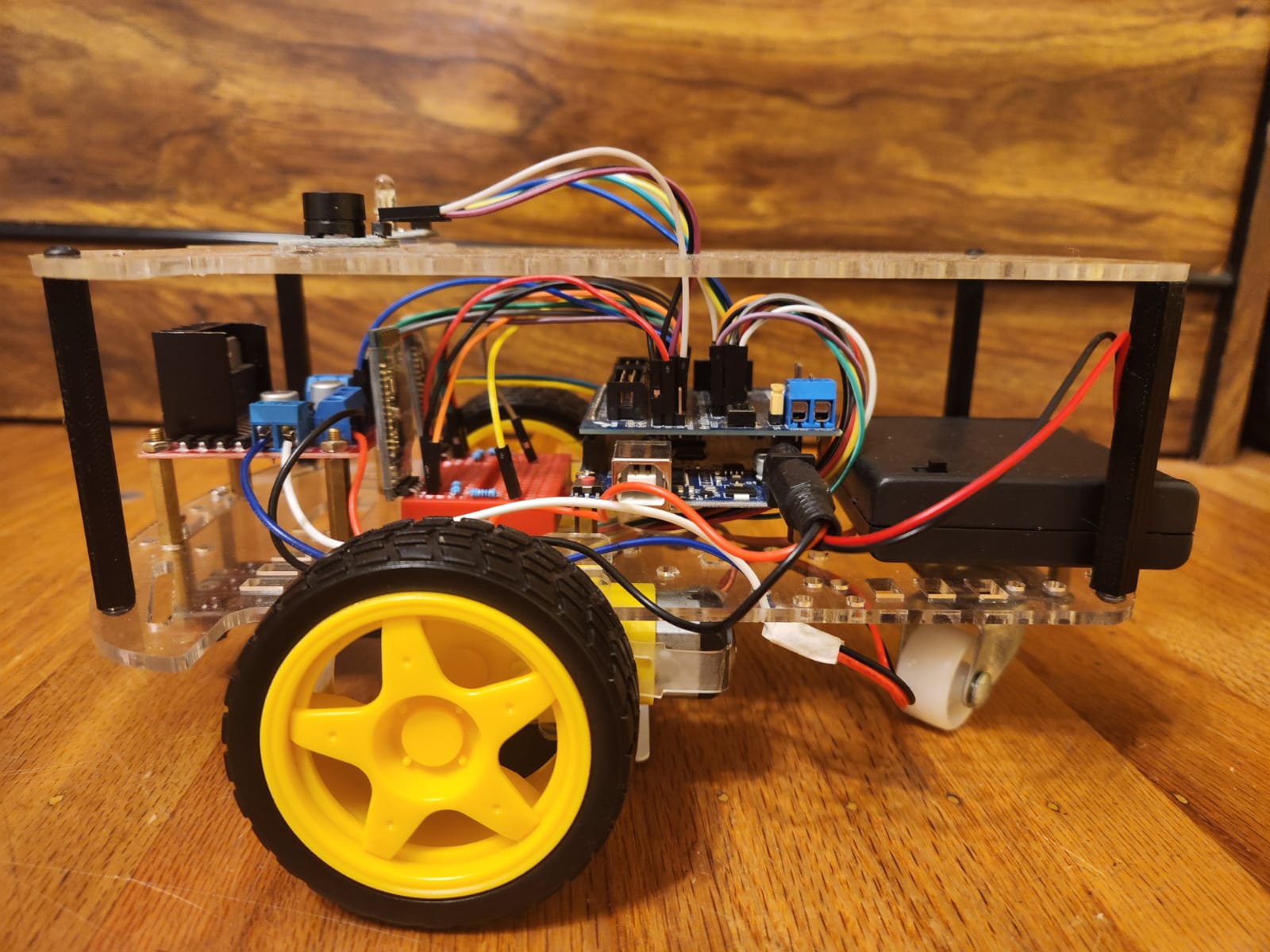
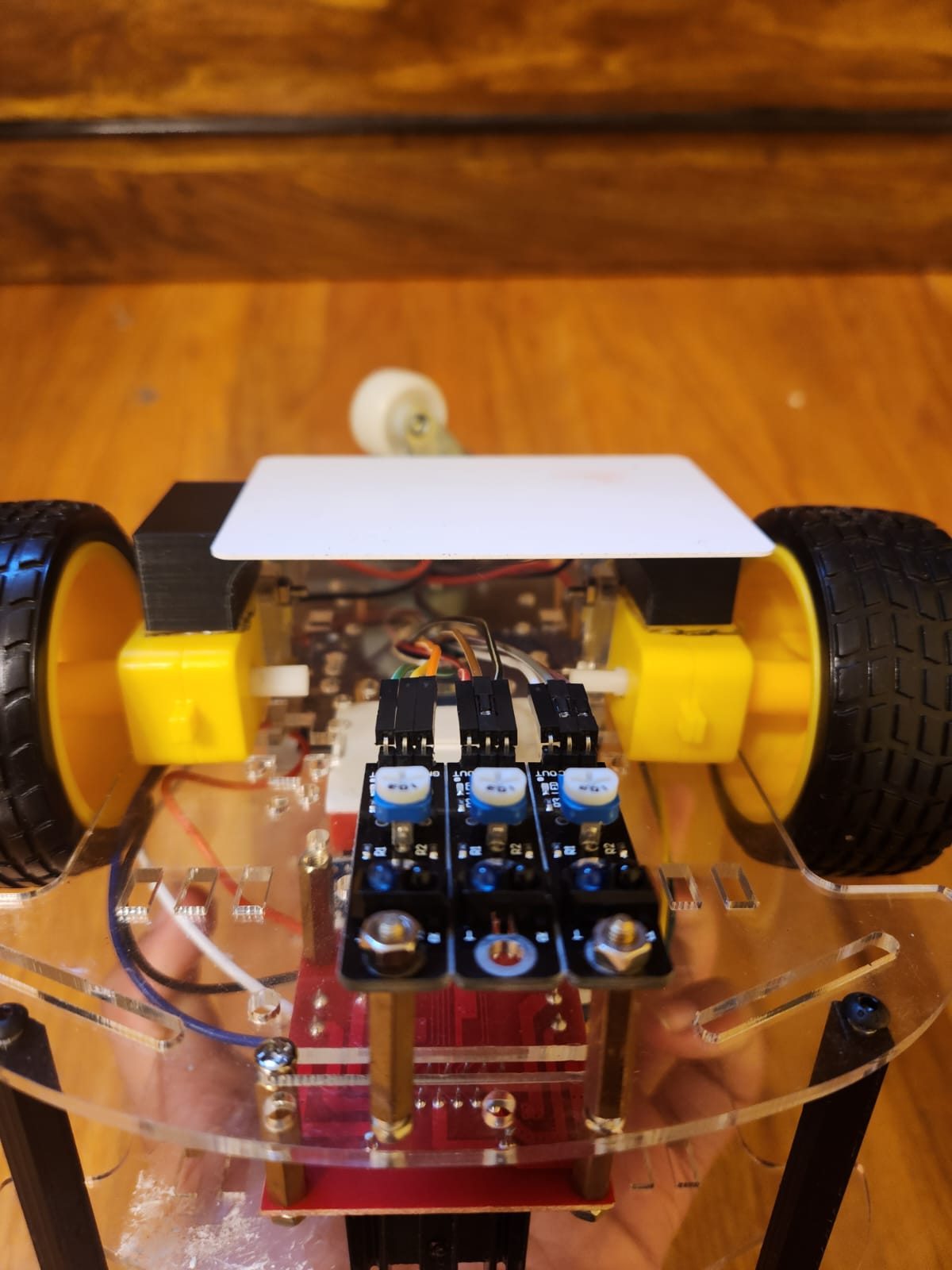
| - 2 acrylic car frames | - 4 AA Batteries + Battery pack |
| --- | --- |
| - Standoffs & screws | - Jumper wires & resistors |
| - 1 small breadboard | - Arduino Uno + Daughter shield |
| - RFID Tag | - 2 Wheels, 2 DC motors, 1 caster wheel |
| - Line follower | - LED and Speaker |
| - Motor driver | - 2 HC-06 BT chips |

For each scanner (2 scanners)

| - 3D-printed box | - Arduino UNO + Daughter Shield |
| --- | --- |
| - Standoffs & screws | - 4 AA Batteries + Battery pack |
| - 1 small breadboard | - RFID Scanner |
| - 2 HC-05 BT Chips | - Jumper wires & resistors |

Track + Building

| - Spray paint | - Large piece of ¼ inch plywood |
| --- | --- |
| - 2 x 4 wood piece | - Tape, screwdrivers, soldering iron, table saw |
| - USB cables | - Laptop w/ Arduino IDE installed |



**Introduction**

* In places like hospitals, patients and doctors might want reduced noise levels at certain times. Currently, there is no way of effectively communicating that to a car passing by.
* Even if drivers are aware of these rules, some may forget or be distracted
* The proposed solution is an automotive system that will automatically enforce these rules without any intervention from the driver