# Cover Page

**Smart Robot Car Kit**

Braylon Morgan

Salatu Bukari

Marcos Chavez

Andrew Medrano

Javier Ramirez

Andy Arce

CS 3319: Internet of Things

Professor Ting Zhang

May 3, 2022

# Abstract

We have decided to build an arduino car that will and can line track, uses infrared control, obstacles avoidance and is also bluetooth enabled. The big idea of our project was to build something fun, interesting, slightly complicated and educational. The multifunctional smart car kit came as a do it yourself kit, with instructions to put it together for every screw and part (i.e. sensors. Installing the IDE has already been set up during class time for Arduino and that was one step we did not need to implement. Mainly the biggest part and most fun part was building the kit. It comes with an assortment of parts. (uno R3, V5 Expansion Board 1PCS, IR Receiver Module 1 PCS, Line Tracking Module 3PCS, Ultrasonic sensor 1PC, Bluetooth Module 1 pc, Black tape 1PC, Acrylic Chassis 2 PC, Servo Motor(SG90) 1PCS, Cell Box 1PCs, F-F Dupont Wire 1PC, Remote Control 1PCS, Ultrasonic Holder 1PCS, Tire 4 PCS, DC Motor 4PCS, L298N Motor Board, Screws, Screwdriver and bundling belts.

Installation process was fairly simple, the project came with directions and there was no need to search and peruse the internet to assemble the Smart car. After completing the build servo motors were the driving mechanisms for the car to move around. They are devices that can turn in specific positions. Using the Arduino we are able to have the car move from one position to another and it will do exactly as it's instructed to do.

# Introduction

**Who are we?**

* Braylon Morgan, CS major hoping to land in the field of gaming as a software engineer or security like NSA.
* Salatu Bukari, CS major and fitness enthusiast. Bridging the gaps from computer science and exercise science.
* Andrew Medrano, CS major still figuring things out but having fun along the way.
* Marcos Chavez, CS major dedicated to coding and hopes to program for a videogame company.
* Javier Ramirez. CS major learning new information day by day and interested in the Artificial Intelligence field.

**Why did we choose this?**

* It's a fun project that incorporates all that we have learned about IOT and more.
* It was and is a great learning experience.

**What are the benefits?**

* Learning how motors work and the connections to the arduino
* Combining mechanical engineering with computer science

**The trial and error?**

* Initially we installed some of the parts incorrectly
* We overcame these errors looking at videos and perusing documents.
  + One of the videos did not have certain connections
    - For instance the ir receiver was not installed on some of the other similar projects for the smart car

**Constructive criticism?**

* The steering should be more advanced
  + The steering is based on isolating one wheel so another can run.
  + A steering mechanism would be more economical considering its size and weight.

# Sensor Determination

As mentioned earlier our project needs several sensors in order to function properly.

**Servo motors**

Servo motors are typically found in remote control cars and RC planes, now we can find them in robots (like our car) and automation. Unfortunately, our smart car can only move two servos at a time. In order to control more than one motor, an external power would be needed. Our project just doesn't offer this type of output, although future projects like drones and moving arms on robots would be exciting to try building. Considering the motors are small and compact they are quite strong for their size and do not require much energy.

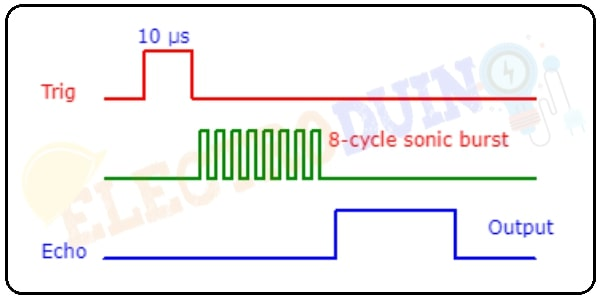
**Ultrasonic sensor**

The exact type used in our class is used in our project and it's meant to give echo feedback in order to detect objects it encounters. The Ultrasonic sensor is awesome for all kinds of projects that require echo location for distance measurements. It requires four outputs (i.e. VCC[power], Tring[sends ultrasonic waves from transmitter as input],Echo[reflection pulse], ground). For the Ultrasonic sensor module HC-SR04 that provides a non contact measurement function ranging from 2cm-13 ft. although, the ranging accuracy is merely 3mm; extreme difference to say the least.

**Basic operation principles**

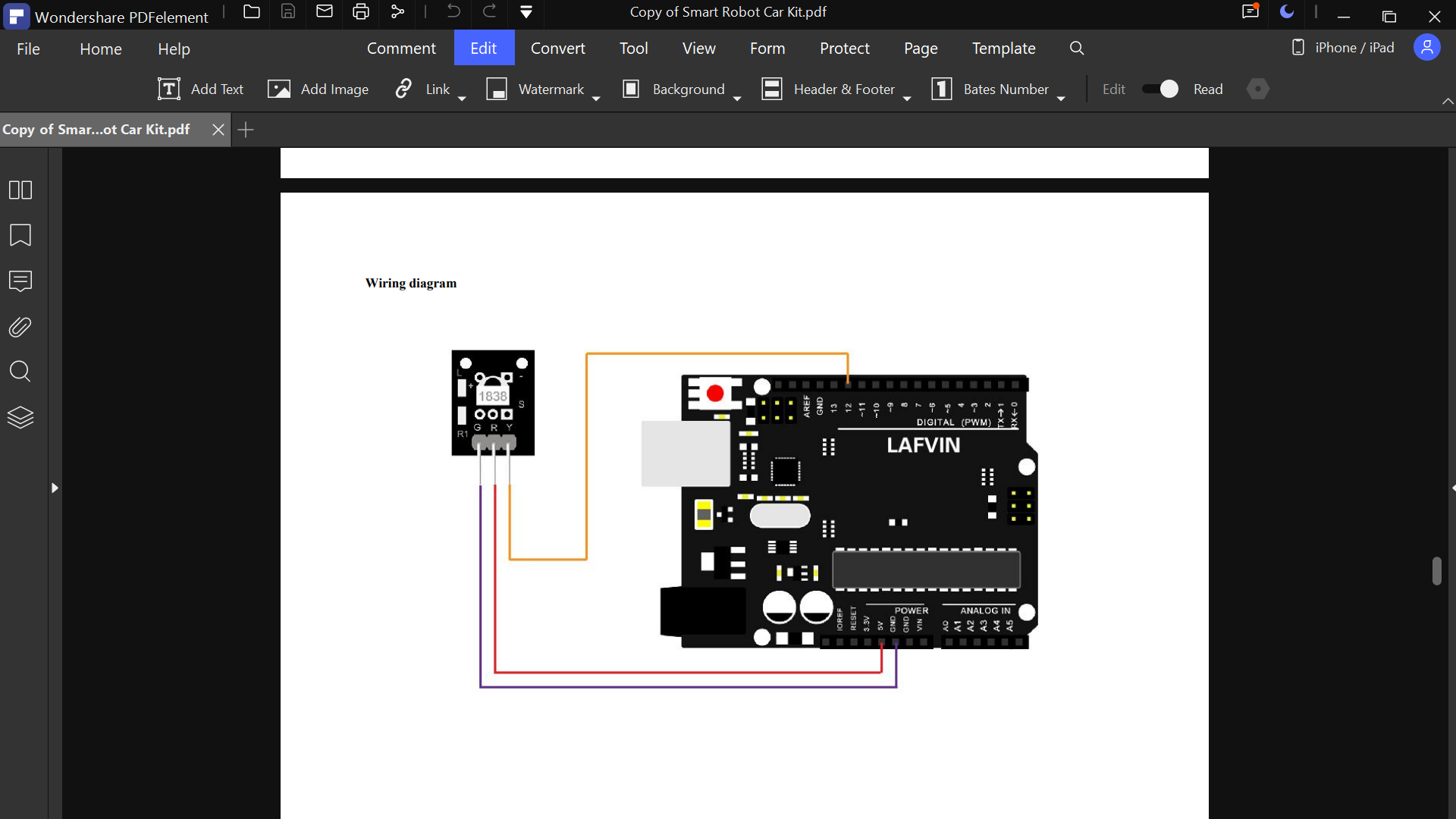
1. IO trigger for at least 10µs(micro seconds) high level signaling
2. Automatic feed sends eight pulses 40 khz and detect whether there is a pulse signal returned
3. If the signal is received, the time of the high output IO duration will determine its distance. Formula: Test distance = (high level time x velocity of sound(340m/s)/2

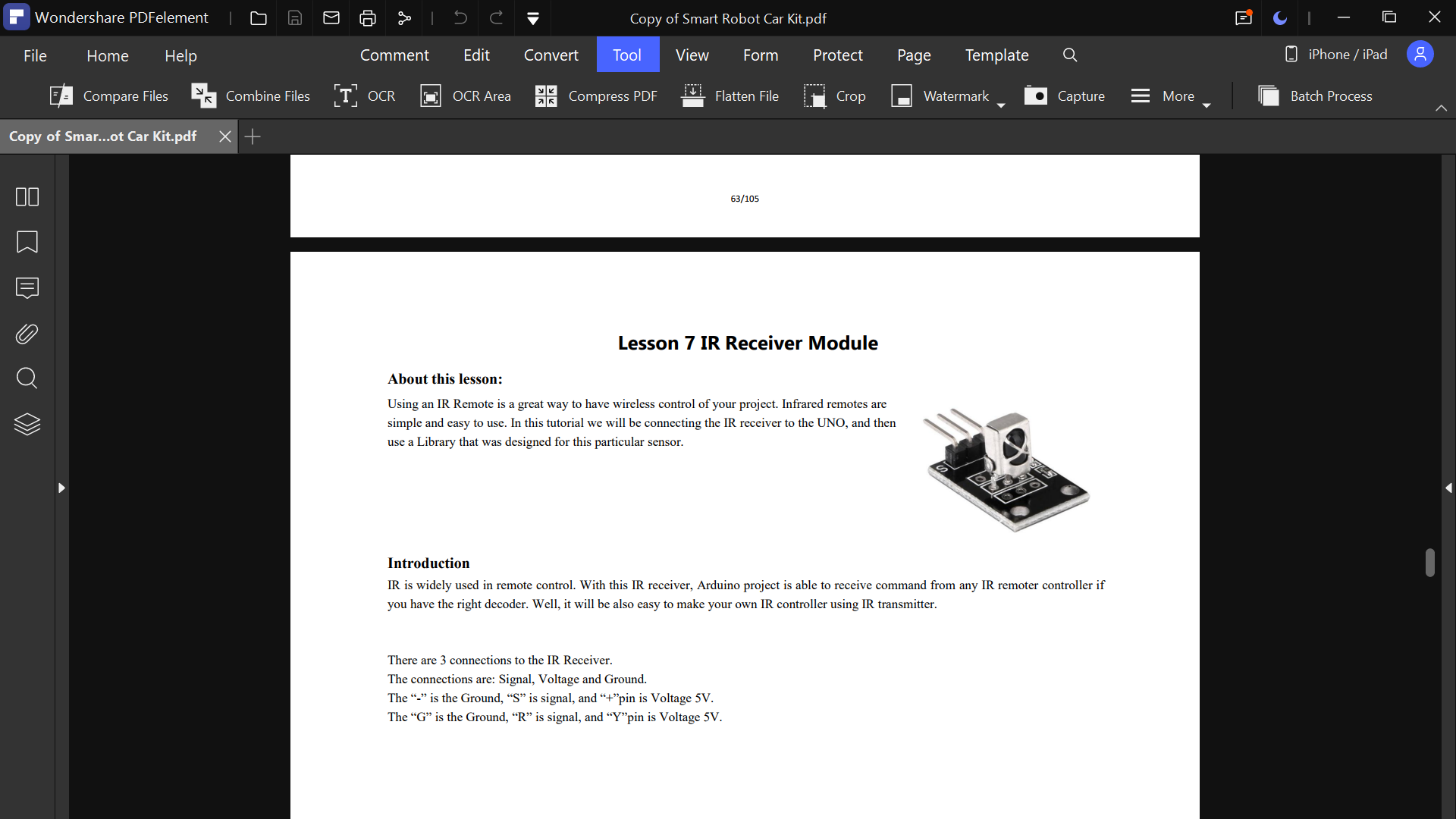
**Wire diagram** **ultrasonic sensor**



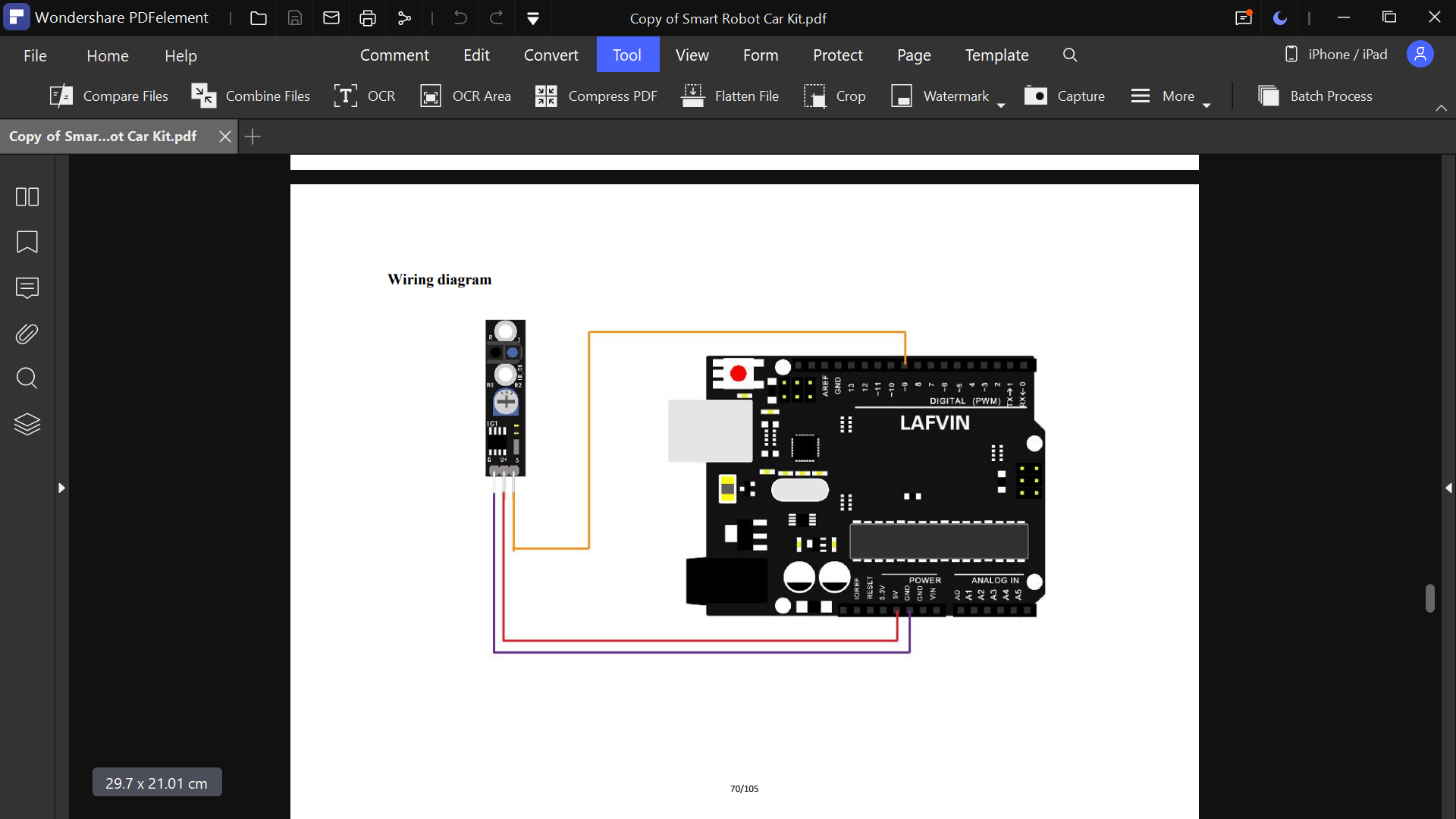
**IR Receiver Module**

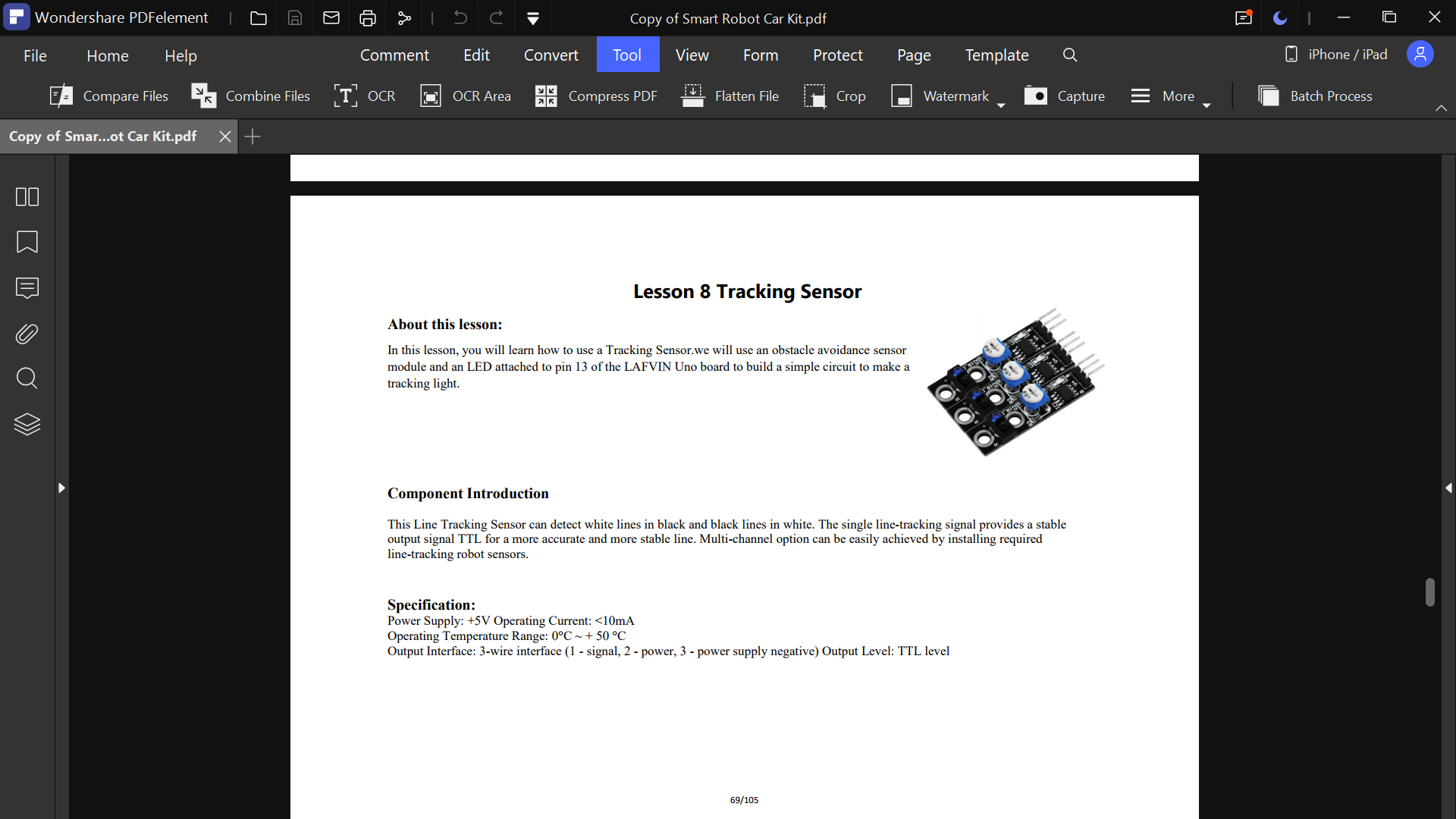
Our project has IR remote capability, which gives the smart car the ability to move without being wired to any source. IR receiver module connects to the UNO, it is necessary to use a library that was designed for this particular sensor. Which is downloaded .

There are three male adapters to the IR receiver. The signal, the voltage and the ground. The ground connector is indicated by “-” the signal indicated by “S” and the voltage is indicated by “+”.

**IR Receiver Wire Diagram**  **IR Receiver Module**

**Tracking Sensor**

The line tracking sensor is designed to track black lines in white and white lines in black. Its simplicity makes for a cool road map for the smart car to traverse . To acquire multiple lines to be tracked more than one sensor is needed therefore we have used three. The sensor returns an infrared signal from the reflection of the line (the energy is sensed) consistently pulsating the energy similar to the ultrasonic sensor but in the form of infrared energy. “ When this line tracking sensor is on a black surface then all of the radiation that’s been transmitted gets absorbed by the surface and nothing is reflected onto the sensor and so we get a zero output signal and when it is on a white surface the opposite happens”. 

**Line tracking sensor** **Wiring diagram**

# Data Collection

For our data collection for each sensor that we are utilizing we have feedback from the ultrasonic sensor, the infrared sensor, bluetooth module and the line tracking sensor

With the ultrasonic sensor we are getting the distance to communicate where things are in the obstacle avoidance mode where it will be using the distance to determine if it should continue or start avoiding the object that is placed.

With the Infrared sensor it is used to change the mode and as well as control the car manually utilizing the infrared sensor to communicate from the remote to the sensor to the arduino

With the line tracking sensor there is getting the color of the ground using the sensors to detect black and white and directing the car in that direction.

# Communication Methods

The communication methods that were given to us gives a clear understanding of how to utilize the aspects of the materials that operate the robot. Given from what we experience with this assignment we can interpret the feedback from the robot, examine its limitations, and use this knowledge for our future endeavors.

**UltraSonic Sensor** 

This sensor is used to detect its whereabouts using echolocation. Depending on the environment the robot is in, the measurement cycle could trigger an echo signal when being used.

**Bluetooth module and Remote Control**

Both the bluetooth module and remote control work in unison to give and receive feedback. Commands can be given to it without the controller by downloading an application on one's phone.

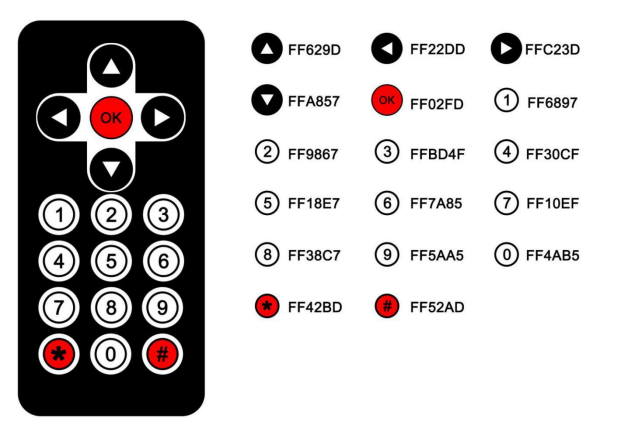
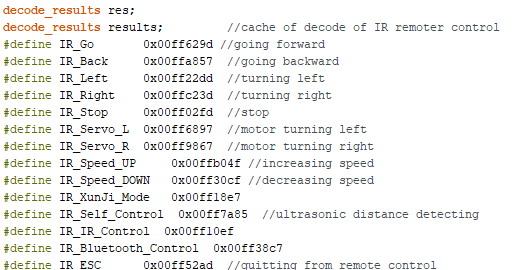
**Line Tracking Module** 

This sensor can detect white lines and black lines in white. The single line-tracking signal provides a stable output for a more accurate and stable line.

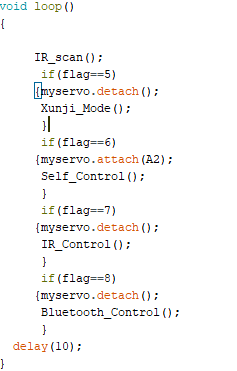
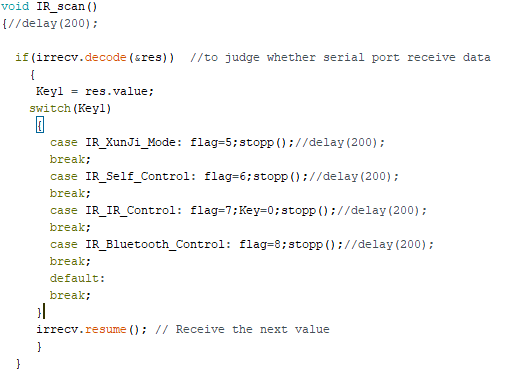
# Communication Implementation

**IR Remote Communication - Mode Selection**

With the necessary wirings (utilizing the wiring diagram), the libraries needed to utilize the functions of the IR Receiver and Controller ( #include IRremote.h ), and the definition of where the receiver pin is located ( for this example, pin 12 [#define RECV\_PIN 12 and IRrecv irrecv(RECV\_PIN)] ) we can now begin to implement the communication system between the remote and the receiver to control our robot car. In order for the user to tell the the car what mode they are using, a loop function was utilized that will update us every 10 milliseconds as to which mode the car is in based on the input from the IR controller (this input from the IR controller is decoded so that it makes the process neater).



The inputs in our loop are: press 5 for line tracking mode, press 6 for obstacle avoidance mode (using the ultrasonic sensor) , press 7 for the IR control mode, and press 8 for the Bluetooth control mode. Of course, since the decoded inputs are not numbers that the loop function can check, another function is created to check the decoded inputs and set a global variable to the intended number that was pressed on the remote.



Once the mode is selected, it will begin to run the function of the selected mode.

**Line Tracking Mode**

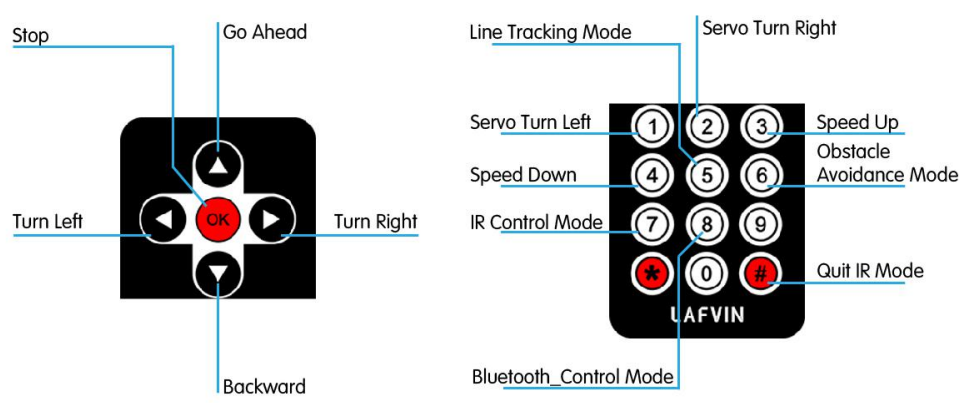
The Line Tracking Mode is a bit more simple in its functions and variable definitions. After connecting the Line Tracking Module to the arduino, defining the pins it is connected to in the IDE (in this case left, middle and right are 9, 10, and 11 respectively) , initializing the sensors, digitally reading the sensors’ values, and setting those values to variables (one for the Left, Right and Middle sensors), all that is needed is a simple function that will facilitate what movements are necessary based on the variables. If the middle sensor reads the area as HIGH (LOW means White and HIGH means Black) then the system can continue with the rest of the options (either the turn left, turn right, or advance forward functions based on what the left and right sensors read) otherwise the system will stop moving (facilitating that we have reached an area that is all white/the end of the line) .

**Obstacle Avoidance Mode**

In Obstacle Avoidance Mode, the Ultrasonic sensor is used to detect if we are near an obstacle so that the car avoids colliding with random objects. After connecting the pins of the sensor to the arduino ( View the wiring diagram), one of the ports of the Ultrasonic sensor should emit a high level of more than 10 US. Once the level is outputted, the potentiometer is open to time. This function will have to read the current value if the port emits a lower level (in this meantime we calculate distance). Using all of this, the program can now detect the distance between an obstacle and the car. If the distance between the car and obstacle is less than 15 cm, the car will go backwards; if it is no less than 35 cm, the car goes forwards; if it is less than 30 cm, the car will now detect the distance between the left and right obstacle by moving the Ultrasonic sensor left and right while measuring. If both left and right measurements are less than 35 cm, the car will move backwards. If the measurement from either side is greater than the other the car will turn towards the direction that is greater.

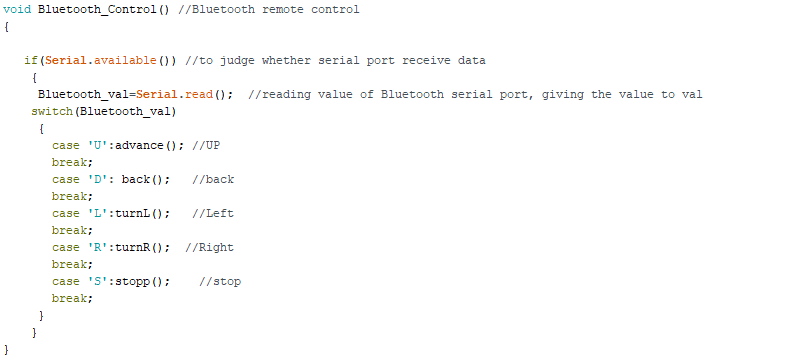
**IR Remote Communication - IR Control Mode**

The process used in the Mode Selection portion of the IR Remote Communication is the same when it comes to the Control mode. The difference being is that this part is meant to control the movement of the Car. In this function the program will cycle a switch case statement based on the decoded input of the user.



**Bluetooth Control Mode**

For the Bluetooth Control Mode, all the program has to do is receive data from the serial port on the Bluetooth Module that the Bluetooth Device had sent to it. Once the Bluetooth Module is connected to the arduino, all that is needed to do is read the value from the Serial port and use a use case statement to see which car movement function is needed. The application that comes with the car is meant to send values to the serial port, (those being U , D, L, R, and S). These values are the ones that determine whether the car goes up, back, left, right or stop.



# Data Analytics

During our time working with the robot, a lot of “what if’s” were asked in the group and because of this it would be best to focus on the prescriptive data analytics. Running tests and courses with the robot gave us a better understanding of trial and error.

In the beginning of its ultrasonic obstacle course, the robot was having a difficult time using its echolocation. The robot had a delayed response when faced with an object in front of it, so to optimize its response to an obstacle, we as a team decided to change the measurement cycle from the default 40ms to 60ms. This made it less likely for the robot to miscalculate its echo signal.

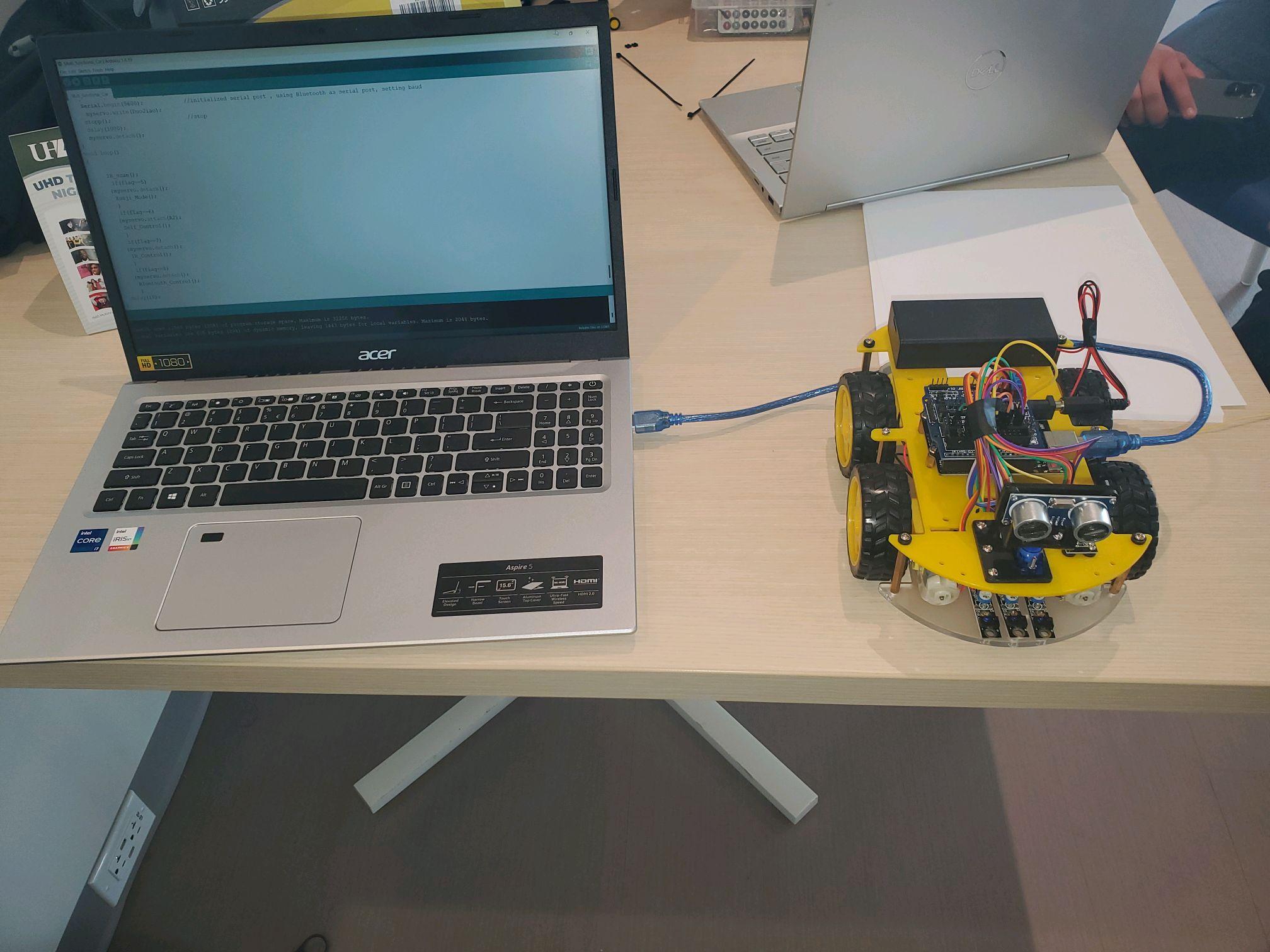
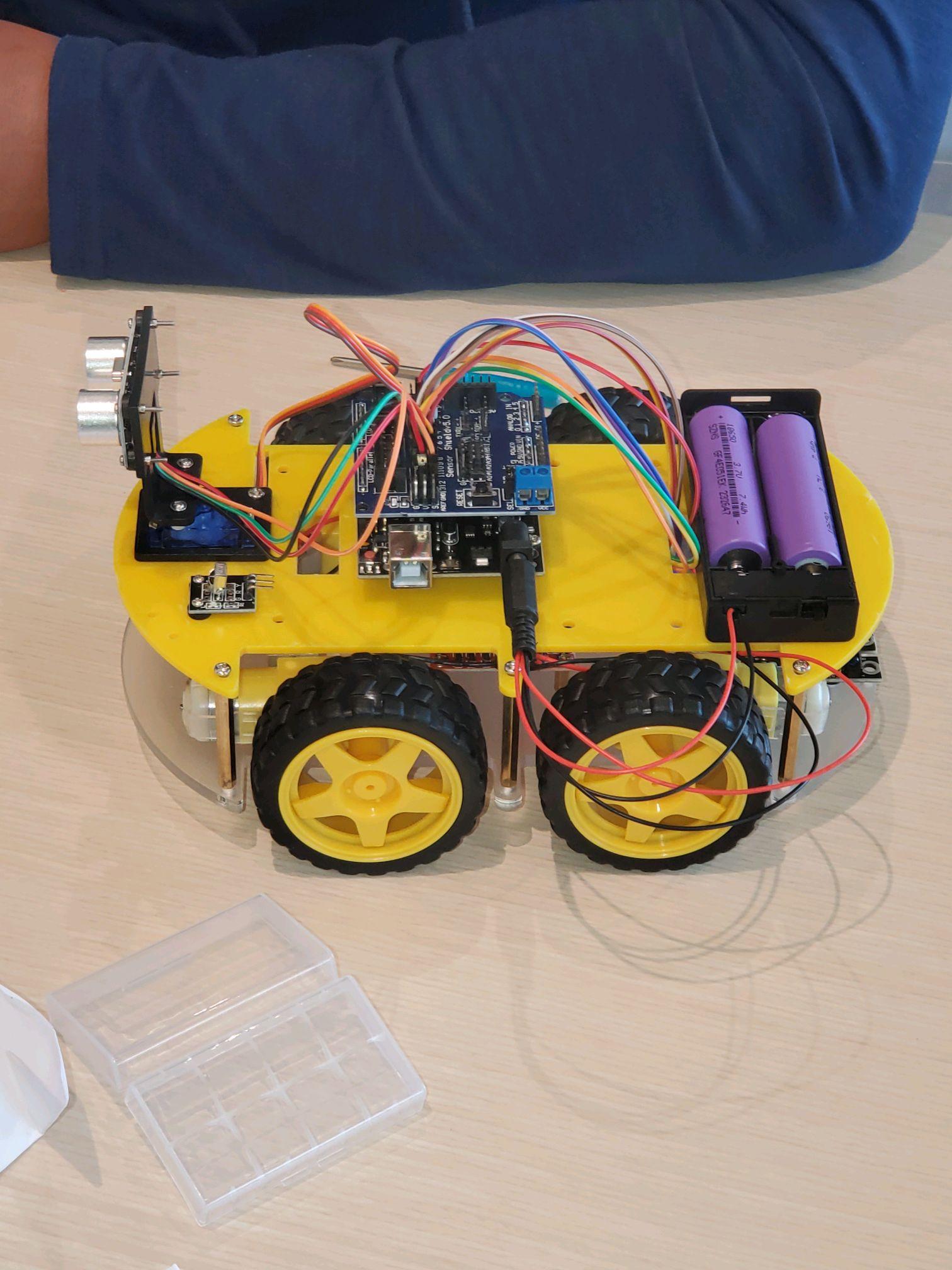
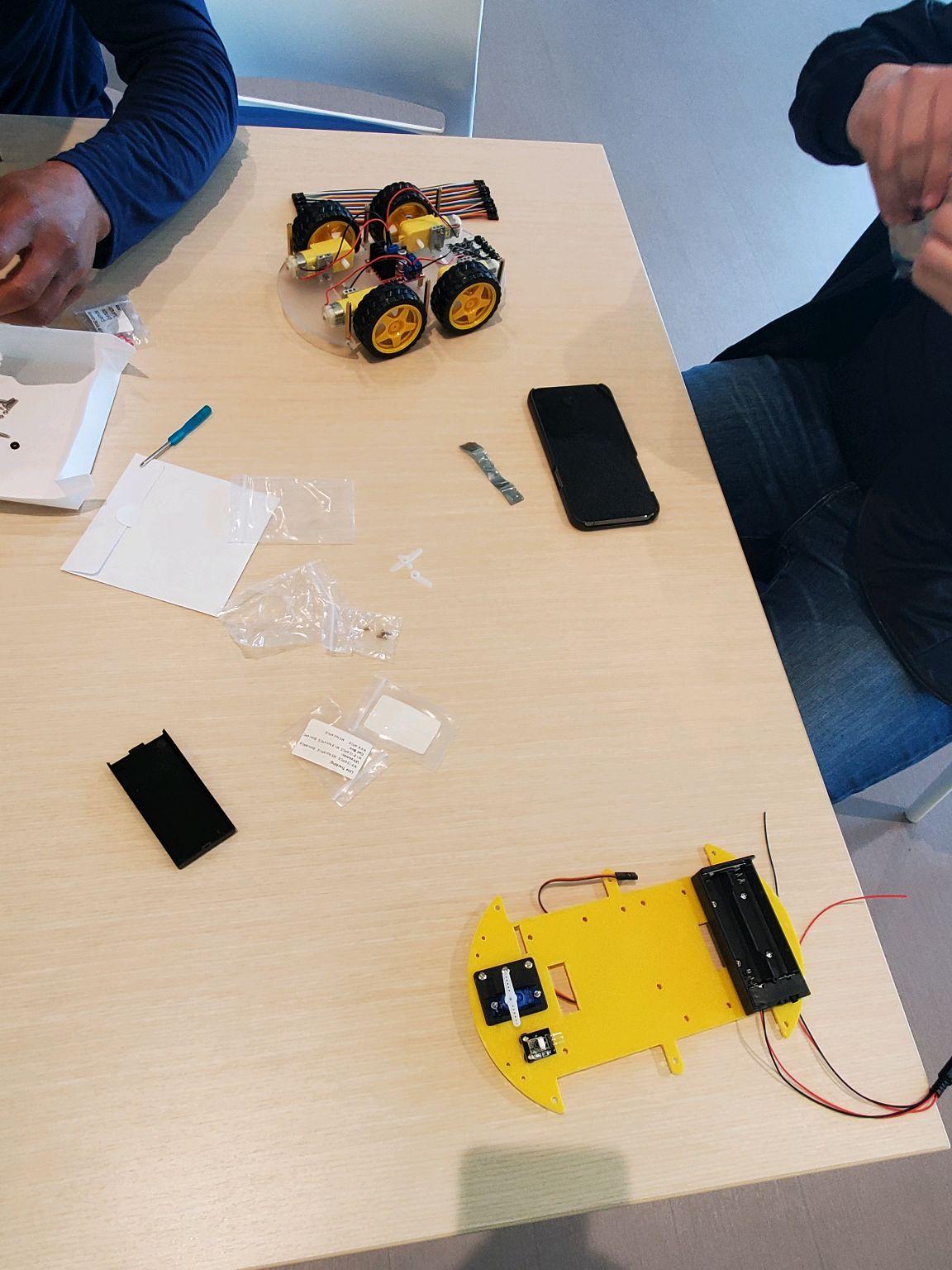
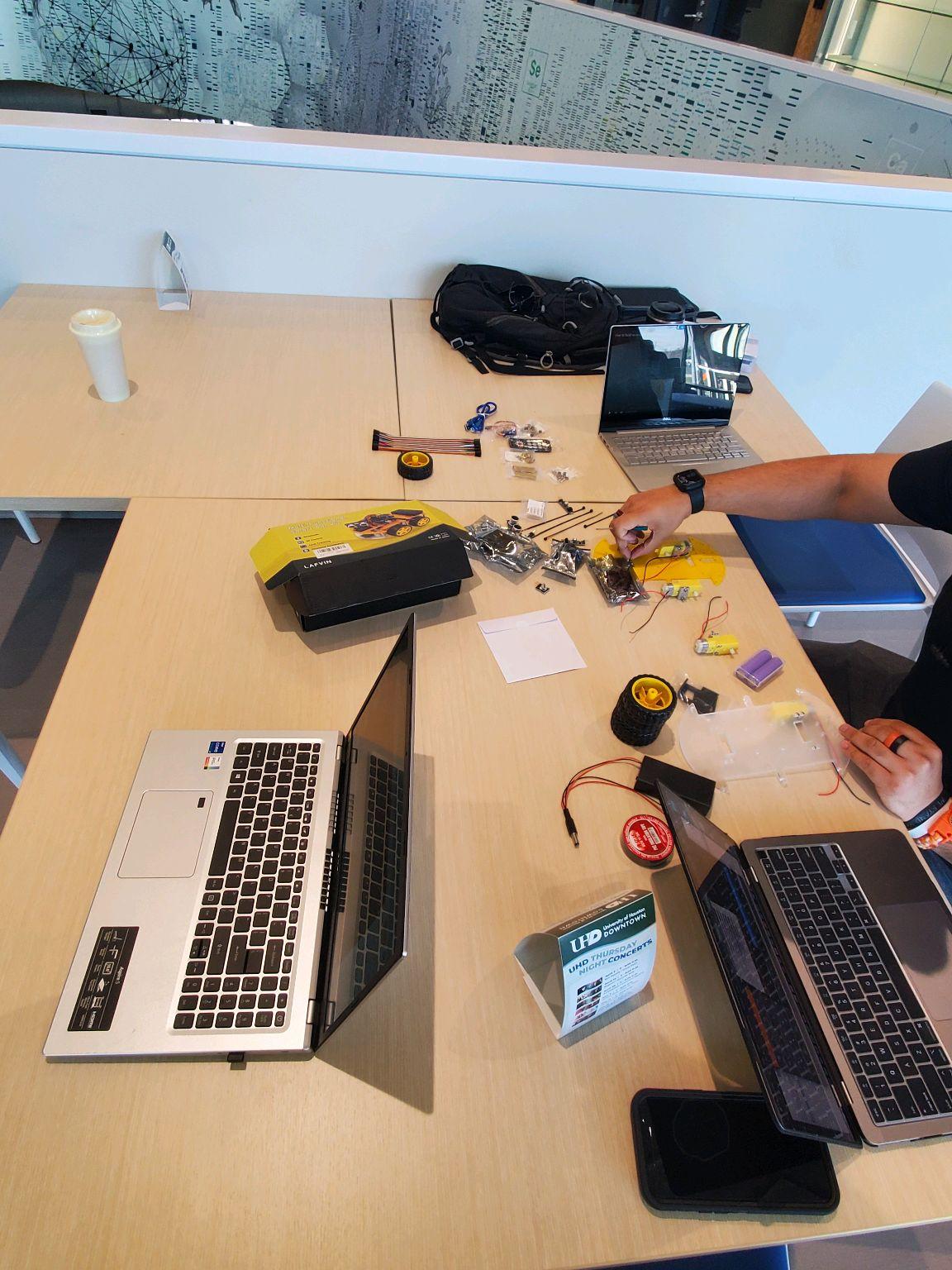
During the line tracking course, the robot was showing irregular movements when attempting to align itself with the line or when discovering a new line to track. The team decided to use black tape to mark the floor and represent it as a line, however it kept repeating the same process. We decided that the line tracking course could use some extra coding to fix this issue, and we would place this under the future works section.

During the IR remote control course, the robot was operating exactly as intended. We had to fix the wheels a few times due to the robot having turning issues, but after the quick fix the robot was moving well. We had to remember to press # on the remote to change back to the other course of the car.

# Feedback Generated~

The feedback generated is completed in different ways, as the IR infrared remote went and switched between all modes and created movement based on if the car was in manual mode. For the other two modes we had the obstacle avoidance mode and the line tracking mode made the car move forward on its own toward. While the car moved forward on its own in the obstacle avoidance mode it had used the servo motor to rotate the ultrasonic sensor to detect distance from anything near so that it would make sure to avoid it. Utilizing a delay in the code and checking its surroundings for any obstacles and moving accordingly.

# Demo

* Pictures: 

# Conclusion/Future Work

In conclusion of our overall build of the arduino car we gained more hands on knowledge that helped us visualize how the different sensors worked on the car such as the ultrasonic sensor, and line tracking sensor on how the arduino car operated as well as how it was able to move on the separate modes, For example the obstacle avoidance mode made it possible for us to visually see the ultrasonic sensor perform as it looked left and right and avoided the obstacles ahead of it and after slight tweaking of the code by increasing the delay we had initially started we were able to have it avoid something as simple as our foot standing in front of it or even have it travel between our legs on it’s own as shown in the video. Furthermore this arduino car gave us an idea as to how smart cars and electric cars are able to use such sensors like the ultrasonic sensor for their self-parking and anti-collision safety systems that are being used to this day. In continuation this arduino car had a mode that used line tracking sensors. We could visually see the sensor at work and the Line Tracking Sensor is most useful for detecting a contrast between dark and light objects. This makes it particularly well suited for tracking dark lines on light surfaces such as in our demo video when using insulating tape as the line, the robot would be able to distinguish it and its sensor would change to a bright red color.

Lastly we gained the necessary experience that we will use for any future work related to arduino, with extensive projects and resources available to us online. A simple example is we can potentially create a working flying drone, a boat that works with remote control and many projects available to explore on the web. The key takeaways that we gained from this project is being able to successfully work as a team writing out our goals and our parts then implementing it making any adjustments to have our project car working to the best of our abilities

# Reference

“Here’s How IoT Data Collection Works [Complete Guide].” *Ubidots Blog*, 30 Dec. 2021, [ubidots.com/blog/iot-data-collection/#:~:text=What%20Is%20IoT%20Data%20Collection,at%20any%20moment%20in%20time](https://ubidots.com/blog/iot-data-collection/#:~:text=What%20Is%20IoT%20Data%20Collection,at%20any%20moment%20in%20time). Accessed 3 May 2022.

“How to Build an Arduino Smart Car (LAFVIN).” [www.youtube.com/watch?v=MF-FqZn5e5M](https://www.youtube.com/watch?v=MF-FqZn5e5M). Accessed 3 May 2022.

“KY-033 Line Tracking Sensor Arduino Circuit and Programming.” *Electronic Clinic*, 29 May 2021, [www.electroniclinic.com/ky-033-line-tracking-sensor-arduino-circuit-and-programming/](https://www.electroniclinic.com/ky-033-line-tracking-sensor-arduino-circuit-and-programming/). Accessed 3 May 2022.

“Servo Motor Basics with Arduino | Arduino Documentation.” [docs.arduino.cc/learn/electronics/servo-motors](https://docs.arduino.cc/learn/electronics/servo-motors). Accessed 3 May 2022.

“Smart Robot Car Kit.” *Dropbox*, [www.dropbox.com/sh/bgcqwry4p3zmx8t/AAAzYjY3kz3VueQyvEzXHoSga?dl=0](https://www.dropbox.com/sh/bgcqwry4p3zmx8t/AAAzYjY3kz3VueQyvEzXHoSga?dl=0). Accessed 3 May 2022.

“Ultrasonic Sensor Module HC-SR04 Datasheet | How It Works» ElectroDuino.” *ElectroDuino*, 16 June 2020, [www.electroduino.com/ultrasonic-sensor-how-ultrasonic-sensor-works/](https://www.electroduino.com/ultrasonic-sensor-how-ultrasonic-sensor-works/). Accessed 3 May 2022.

‌

‌

‌

‌

‌