CS-227 PROJECT

MULTI FUNCTIONAL BLUETOOTH CONTROLLED ROBO-CAR USING ARDUINO-UNO

MADE BY

Harshvardhan Singh 2201CS92
Kushal Agarwal 2201MC22

CONTENTS

INTRODUCTION

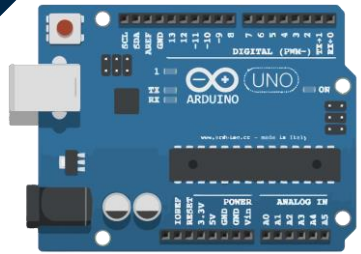
We built a Bluetooth-controlled Robo-Car model using Arduino and the HC 05 Bluetooth module that used an Android app as its remote controller.

We added functionalities like Crash avoidance , LED ,driver motor speed control, obstacle distance measure.

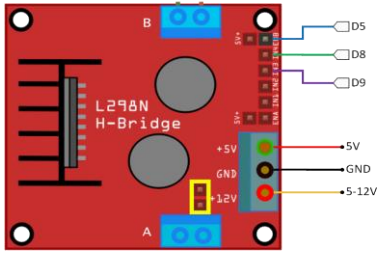
This was a group project of , Harshvardhan Singh (2201CS92)

Kushal Agarwal (2201MC22)

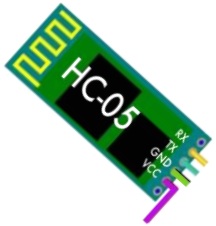
COMPONENTS REQUIRED



Arduino Uno

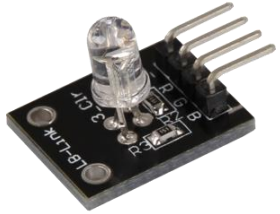


L298 Motor Driver

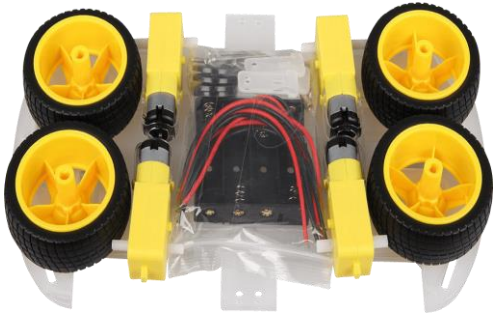


HC-05 Buetooth Module

Jumper Wires



RGB led

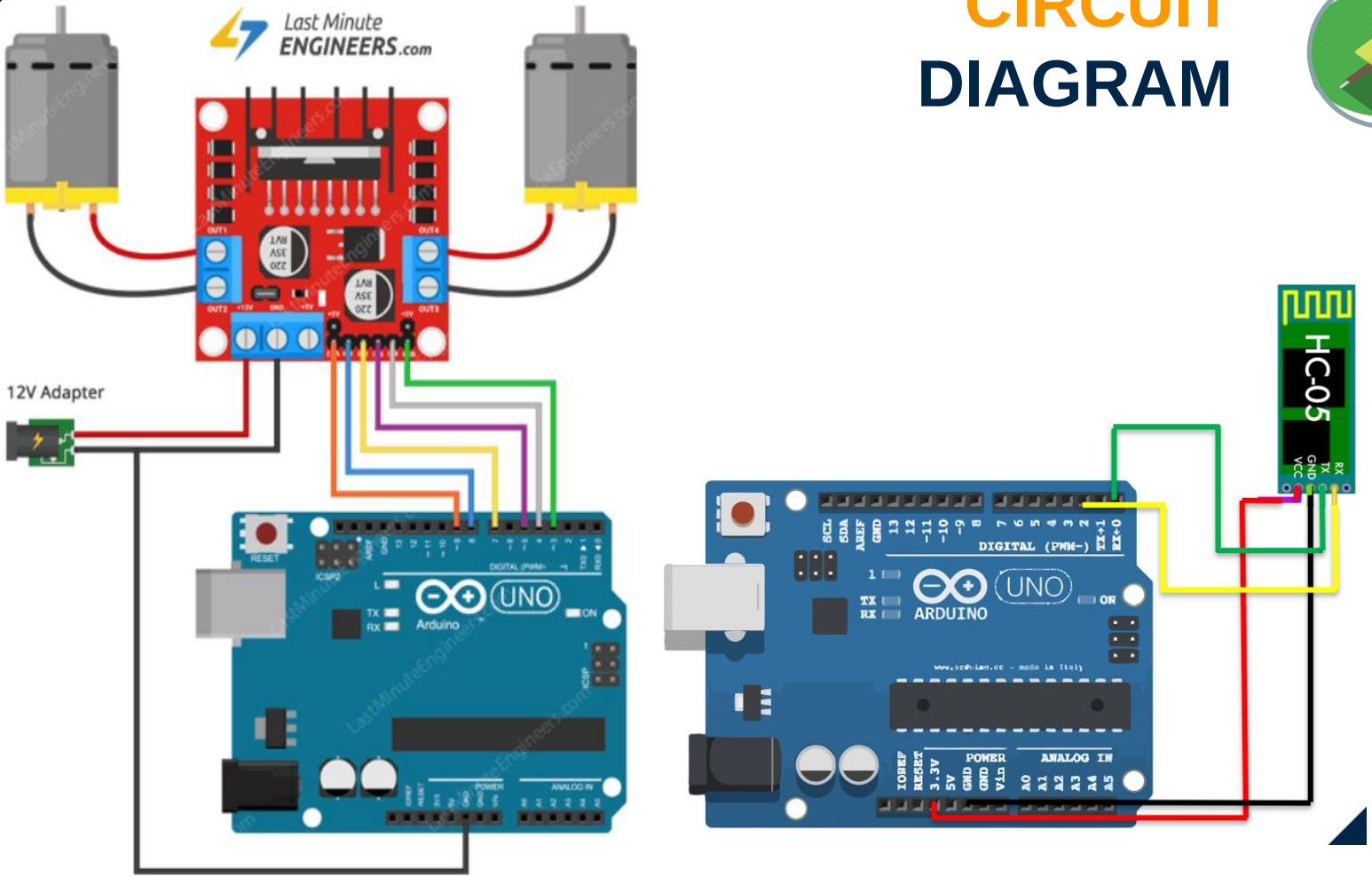


Car Chasis with 4 motors and wheels

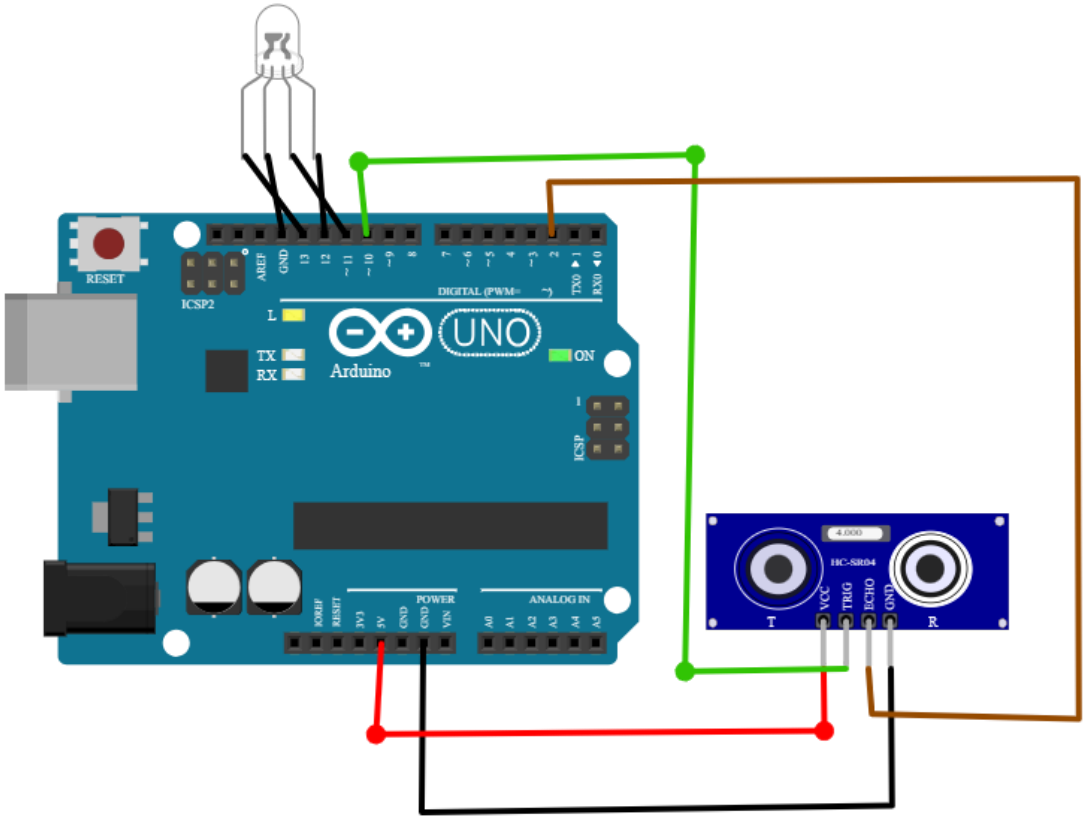
Lipo Battery 12V

Ultrasonic Distance sensor

CIRCUIT DIAGRAM

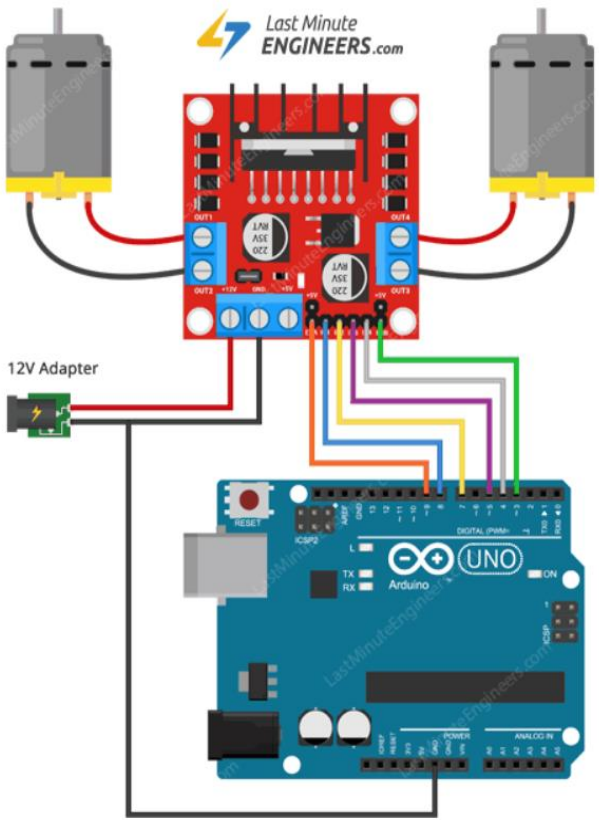


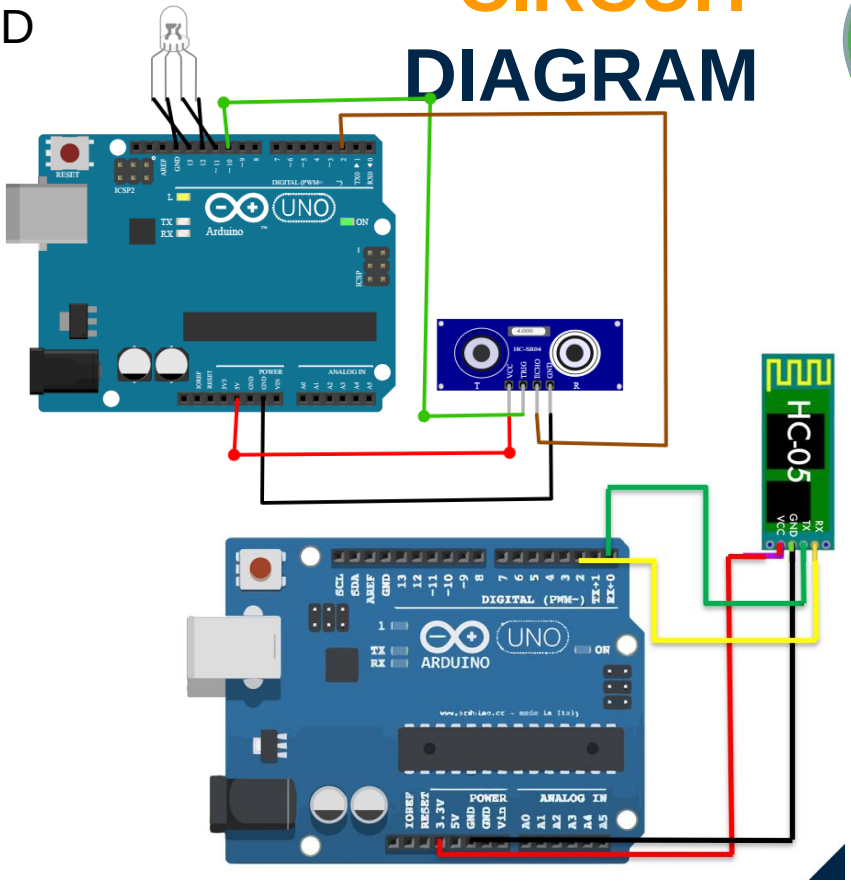
CIRCUIT DIAGRAM



FINALIZED

CIRCUIT DIAGRAM





ARDUINO CODE

```
#include <SoftwareSerial.h>

// Bluetooth module connection
SoftwareSerial bluetooth(0, 1); // RX, TX

// Motor driver connections
const int EN1 = 9;
const int IN1 = 8;
const int IN2 = 7;
const int IN3 = 6;
const int IN4 = 5;
const int EN2 = 3;
int MotorSpeed=70;
const int trigPin = 2;
const int echoPin = 10;

// Variables
long duration;
int distance;
int alertDistance = 7;// 7 cm
char crashavoiderstate='1';
char Command;

// LED connections
const int movingLedPin = 12; // Green LED
const int stoppedLedPin = 13; // Red LED

char lastcommand='S';


void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);

  pinMode(EN1, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(EN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  pinMode(movingLedPin, OUTPUT);
  pinMode(stoppedLedPin, OUTPUT);

  stopCar();
  // Ensure the car starts in a stopped state
}

void loop() {

  digitalWrite(trigPin, LOW);         // initializing the distance sensor
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
```

```
56    delayMicroseconds(10);
57    digitalWrite(trigPin, LOW);
58    duration = pulseIn(echoPin, HIGH);
59    distance = duration * 0.034 / 2;
60    bluetooth.println(distance);       // for transmitting to bluetooth
61    Serial.println(distance);
62
63
64
65    if (bluetooth.available() > 0) {
66      Command = bluetooth.read();
67    }
68
69    switch (Command)  // for motor speeds
70    {
71      case '2':
72            MotorSpeed=70;
73            break;
74      case '3':
75            MotorSpeed=100;
76            break;
77      case '4':
78            MotorSpeed=150;
79            break;
80    }
81    setMotorSpeed(MotorSpeed);
82
83    if(Command=='F'|| Command=='B'|| Command=='L'|| Command=='R'|| Command=='S')
84    {
```

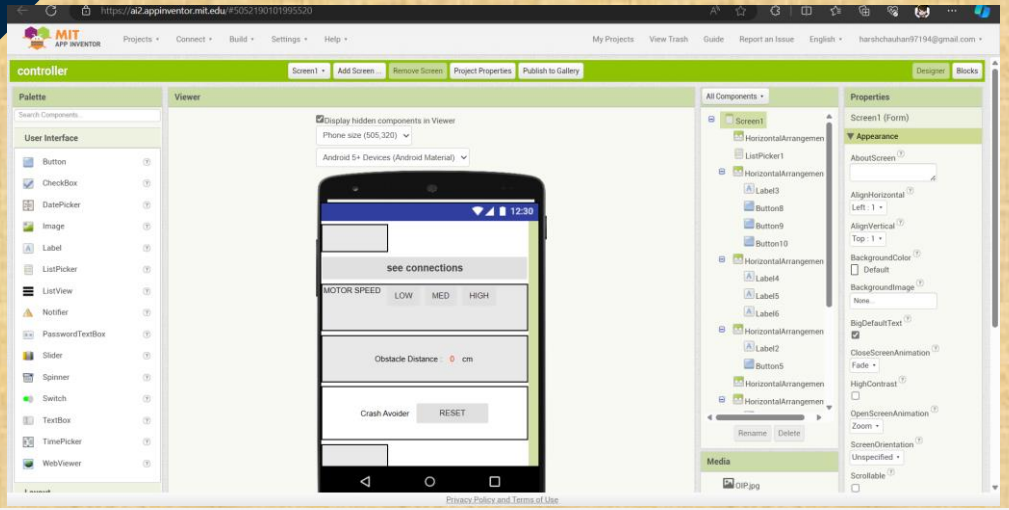
```
    lastcommand=Command;
  }
  controlCar(lastcommand);  // move car


   // Update LEDs based on car movement
  if (lastcommand == 'F' || lastcommand == 'B' || lastcommand == 'L' || lastcommand == 'R') {
    digitalWrite(movingLedPin, HIGH);   // Green LED ON
    digitalWrite(stoppedLedPin, LOW);  // Red LED OFF
  } else {
    digitalWrite(movingLedPin, LOW);    // Green LED OFF
    digitalWrite(stoppedLedPin, HIGH);  // Red LED ON
  }

if(Command=='1' || Command=='0')
{
  crashavoiderstate=Command;
}

 if(crashavoiderstate=='1' )
 {
  // Check if the distance is within the alert range(less than 5 cm)
  if(distance < 5) {
```

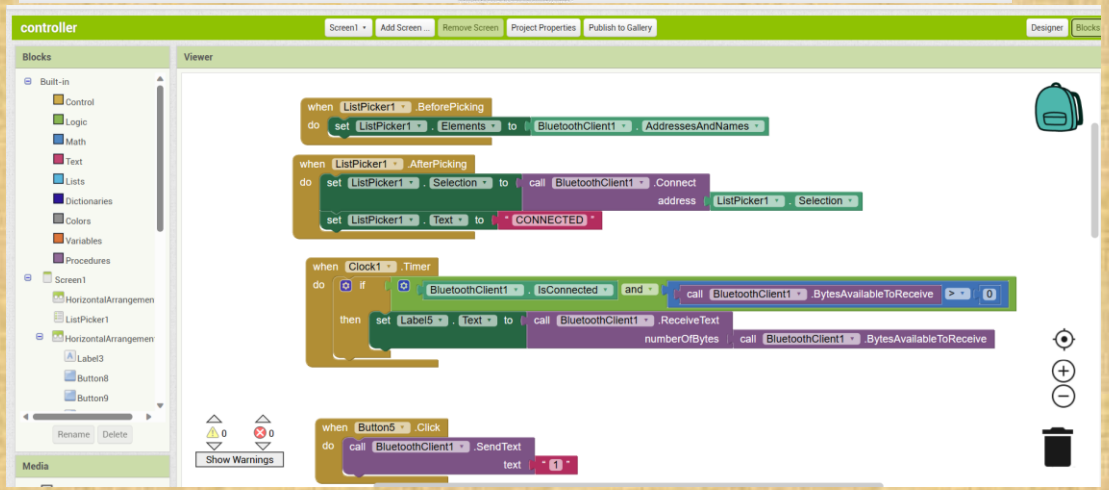
```
    stopCar();
    lastcommand='S';
    crashavoiderstate='0';
  }
 }
  // Store the last received command

  delay(100);
}

void controlCar(char speedCommand) {
  switch (speedCommand) {
    case 'F':
      moveForward();
      break;
    case 'B':
      moveBackward();
      break;
    case 'L':
      moveLeft();
      break;
    case 'R':
      moveRight();
      break;
    case 'S':
      stopCar();
  }
```

```
}

}

void setMotorSpeed(int speed) {
  analogWrite(EN1, speed);
  analogWrite(EN2, speed);
}

void moveForward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void moveBackward() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void moveLeft() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
```

```
}

void moveRight() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}

void stopCar() {
  setMotorSpeed(0); // Set speed to 0
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);

  digitalWrite(movingLedPin, LOW);    // Green LED OFF
  digitalWrite(stoppedLedPin, HIGH);  // Red LED ON
}
```

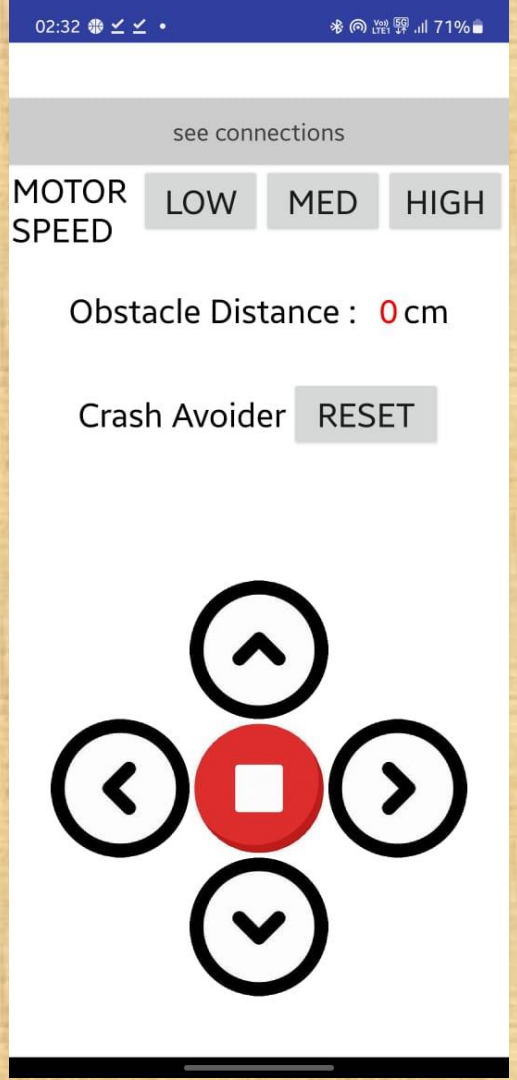
ANDROID APP





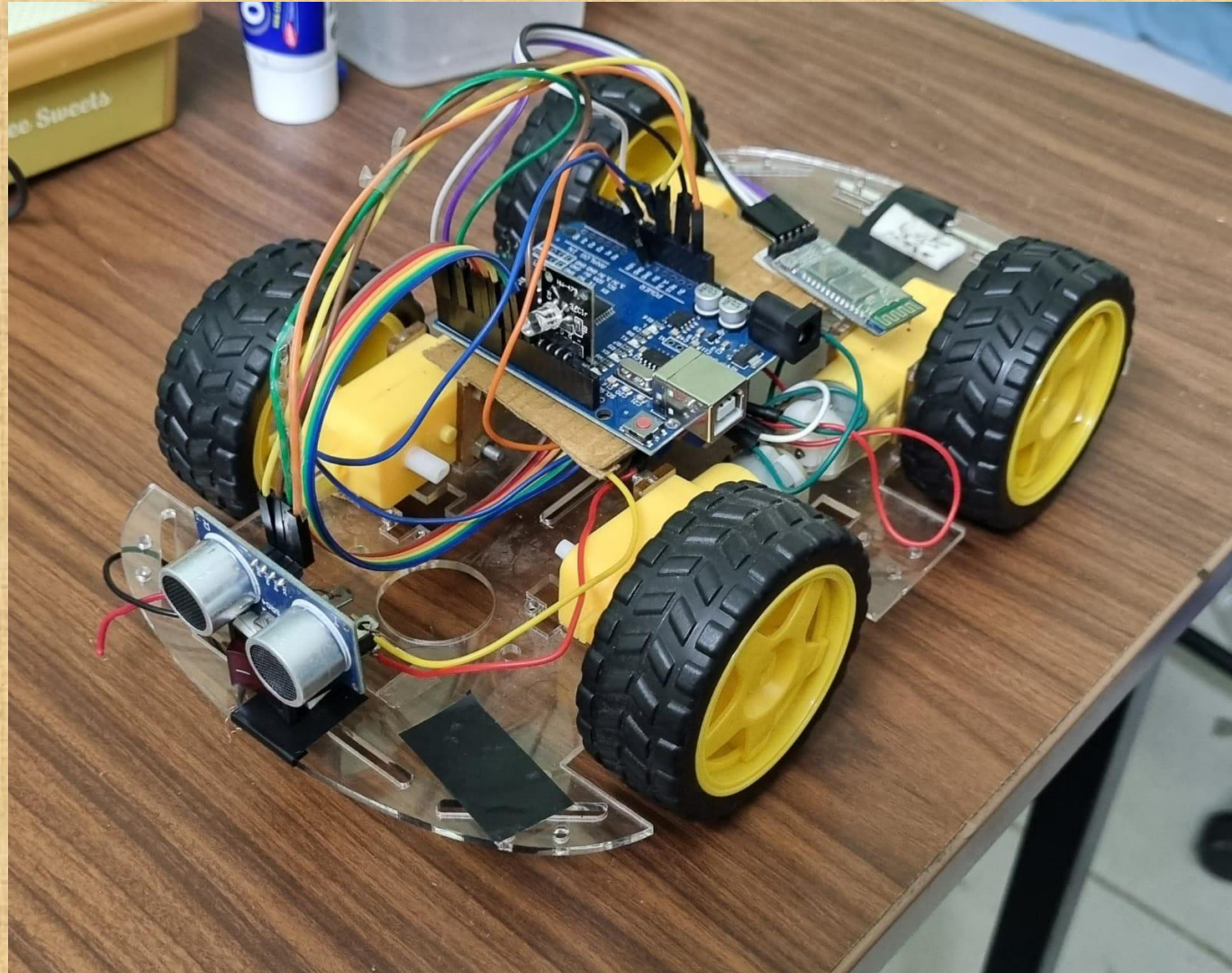
DEVELOPMENT PHASE

FINAL APP



ANDROID APP

FINAL MODEL



METHODOLOGY

DIVIDING TASK : we divided our task as I was handling more or Software works like app, Arduino code.

while Kushal Agarwal was building the hardware of RC car.

methodology in steps—

1. **Assembling all the components required and building the Car chassis , assembling wheels and motors.**
2. **Connecting Motors to motor driver with right pin planning.**
3. **Connection motor driver with Arduino using Ena,in1,in2,in3,in4,Enb, which controls the speed of right motor, right motor forward, RM back, LM forward, LM back and speed control of left motors respectively.**
4. **Powering Arduino from motor driver 5v output and ground, which take 12v input from battery.**
5. **Connecting Bluetooth module and first making a basic rc car code in arduino.**

Developing app : we developed app using MIT app inventor. Gave a basic frontend design. Integrating module like Bluetooth, clock(for refreshing screen when obstacle distance is received). Gave a backend logic for all of these. building app and exporting.

6. uploaded the code and checked if basic functions like forward ,backward,stop are implemented or not.
7. **Adding crash avoidance :** Integrating ultrasonic distance sensor and appending its working arduino code snippet in base code.Its basic function was to detect distance if it is less than 7-8 cm it will stop the motors. It will only provide one instance of crash avoidance after which we have to push reset button to add one more instance of crash avoidance in it.

8. **Adding led:** appending using appropriate logic in base code, green led is high only when car is moving , Red will high when car is in stopped state.
9. **Adding Manual motor speed :** utilizing the ENa and ENb pins to give values to motor speed in 3 ranges ,low medium and high.
10. Transmitting the value of obstacle distance from arduino to app via Bluetooth and properly showing in app.
11. Reviewing the final code and hardware of project and its all set.

Problems faced and Conclusion

Problem uploading code into arduino: Many times we found out that arduino had trouble while uploading code showing error messages like unable to access port, programmer failed to respond. Mostly these were solved either by reconnecting the arduino, or in some unfortunate cases, replacing the entire arduino. In one case we faced this error because we were using the ports 0, 1 while uploading, and when we disconnected those ports it seemed to work again.

Problem receiving and reading bluetooth signals: A lot time of the debugging was spent trying to fix the process of receiving the signals from bluetooth. Although the issue was mostly due to the circuit, we also had to fix the code as we did some errors in implementing the proper use of softwareserial library.

Problem with battery: We tried to drive the car using 9V batteries, but it failed to provide enough power to run the car. Even a combination of 9x4 = 36V was not enough to get the car rolling. We fixed it by using a 12V lippo battery which provided enough power to run the car at full speed.

And A LOT OF DEBUGGING PROBLEMS!!!!

Problems faced and Conclusion

This hands-on Arduino project was a great learning experience for both of us. It helped us learn how to write Arduino code and make projects like this. It also made us realize how different dealing with real hardware is how things might not go as planned always, and we need to adapt to the various situations we face.

We also familiarized with app development and integration of various modules(bt, clock) in it.

We would like to thank Jimson Matthew, sir, for allowing us to work on such an exciting project.

T H A N K S