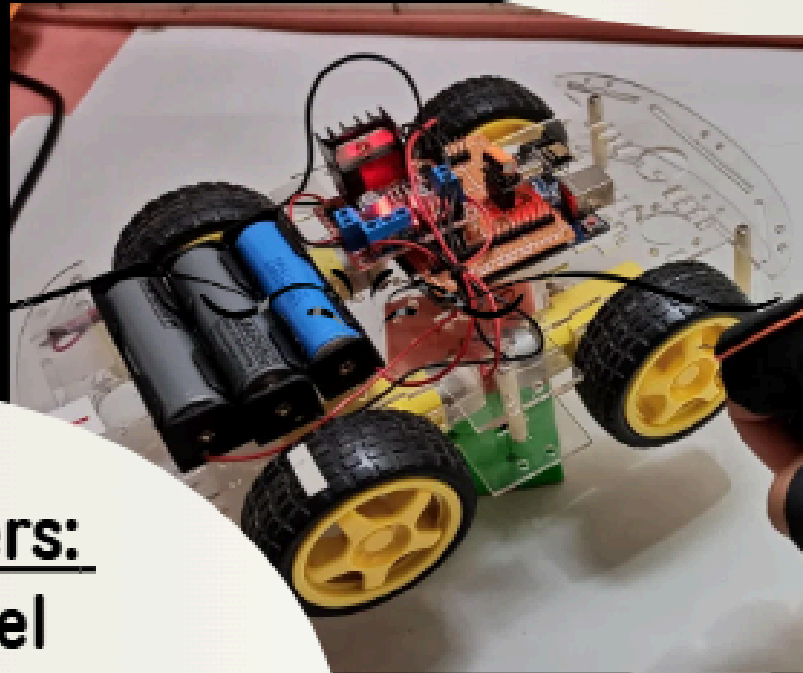




Hand Gesture Controlled Car (HGCC)



Team members:

Kashish Patel

Ayush Jaiswal

Abhay Sahu

Mehul Vishwakarma

Project Incharge :

Prof. Supriya
Shrivastava

Project Guide :

Prof. Rahul Chelani



Outline

AIM

COMPONENTS

CIRCUIT DAIGRAM

CODE

WORKING



Hand Gesture Controlled Car (HGCC)





COMPONENTS

1. Arduino NANO

2. Arduino UNO

3. L298 Motor Driver

4. Car Chassis

5. RF Module

6. ROOFER INR
18650 25000
ma(Battery)



COMPONENTS

7. Wheels

8. Gear Motor

9. Connectors

10. PCB

11. Adopter

12. Shouldering wire

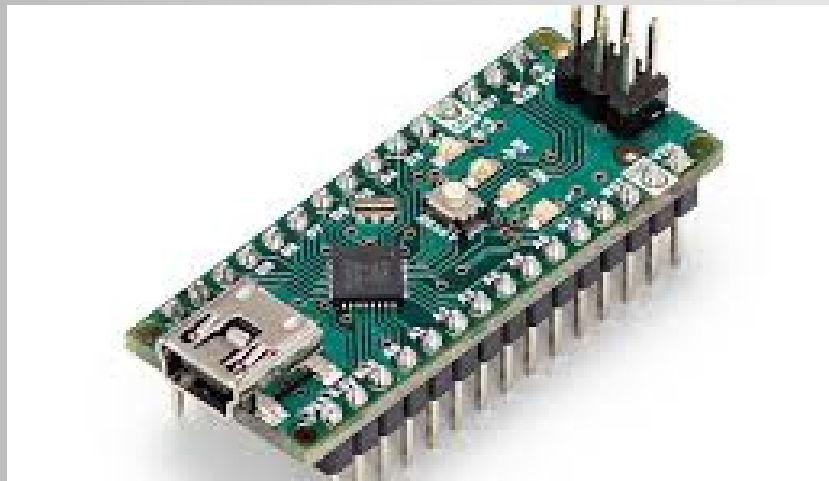
13. Glove

14. Velcro tape



Arduino NANO

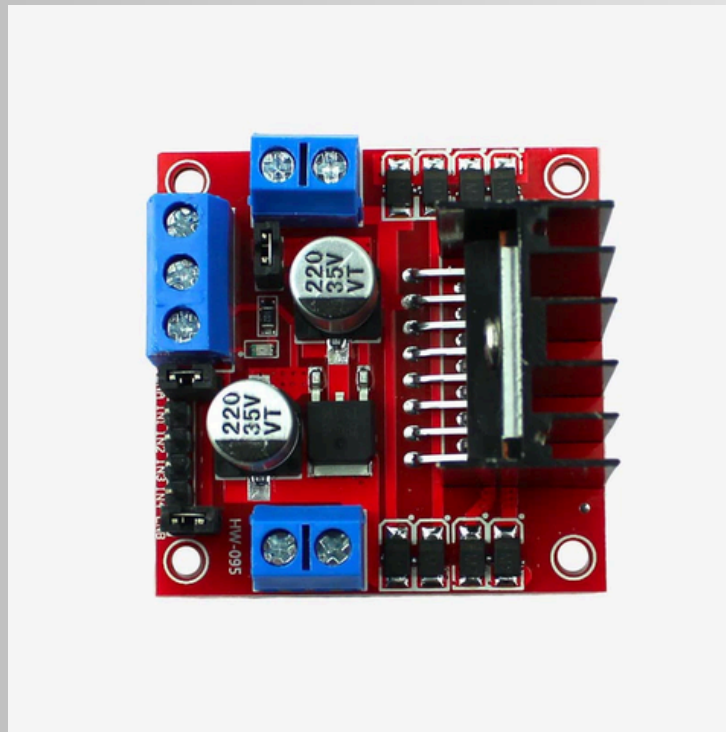
The Arduino Nano is an open-source breadboard-friendly microcontroller board based on the Microchip ATmega328P microcontroller (MCU) and developed by Arduino.cc and initially released in 2008. It offers the same connectivity and specs of the Arduino Uno board in a smaller form factor





L298N Motor Driver

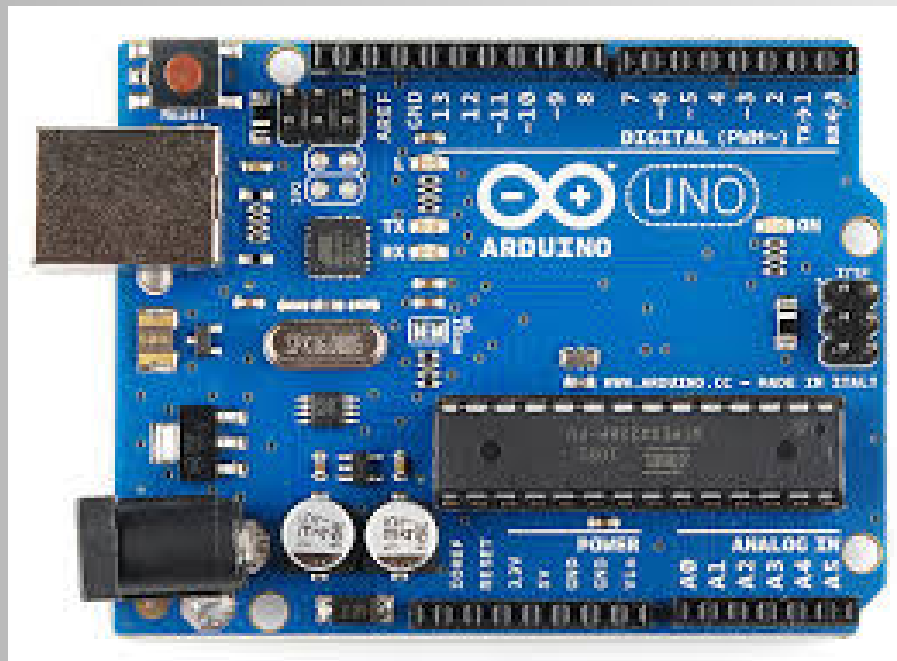
L298N motor driver controls both the speed and direction of rotation of a DC electric motor. It uses a L298N PWM system, which can control voltage using square wave pulses. The wider the pulses, the faster the motor will rotate





Arduino UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller (MCU) and developed by Arduino.cc and initially released in 2010





MPU6050(Accelerometer)

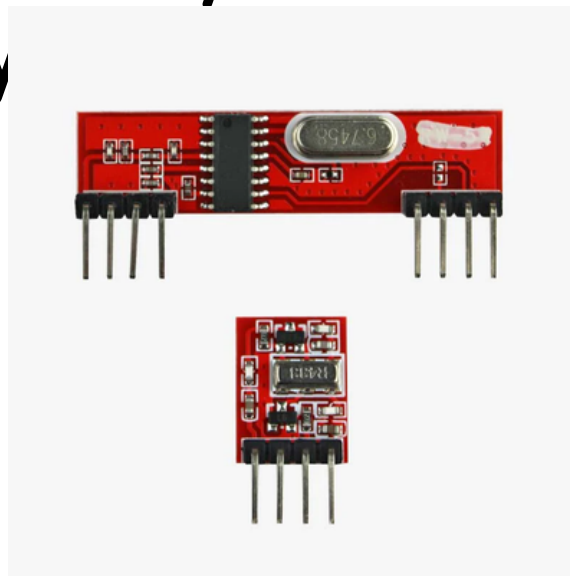
- The MPU-6050 is a 6-axis (combines 3-axis Gyroscope, 3-axis Accelerometer) motion tracking device. Changes in motion, acceleration and rotation can be detected. It is commonly used in robotics, gaming controllers, and other electronic devices that require motion detection.





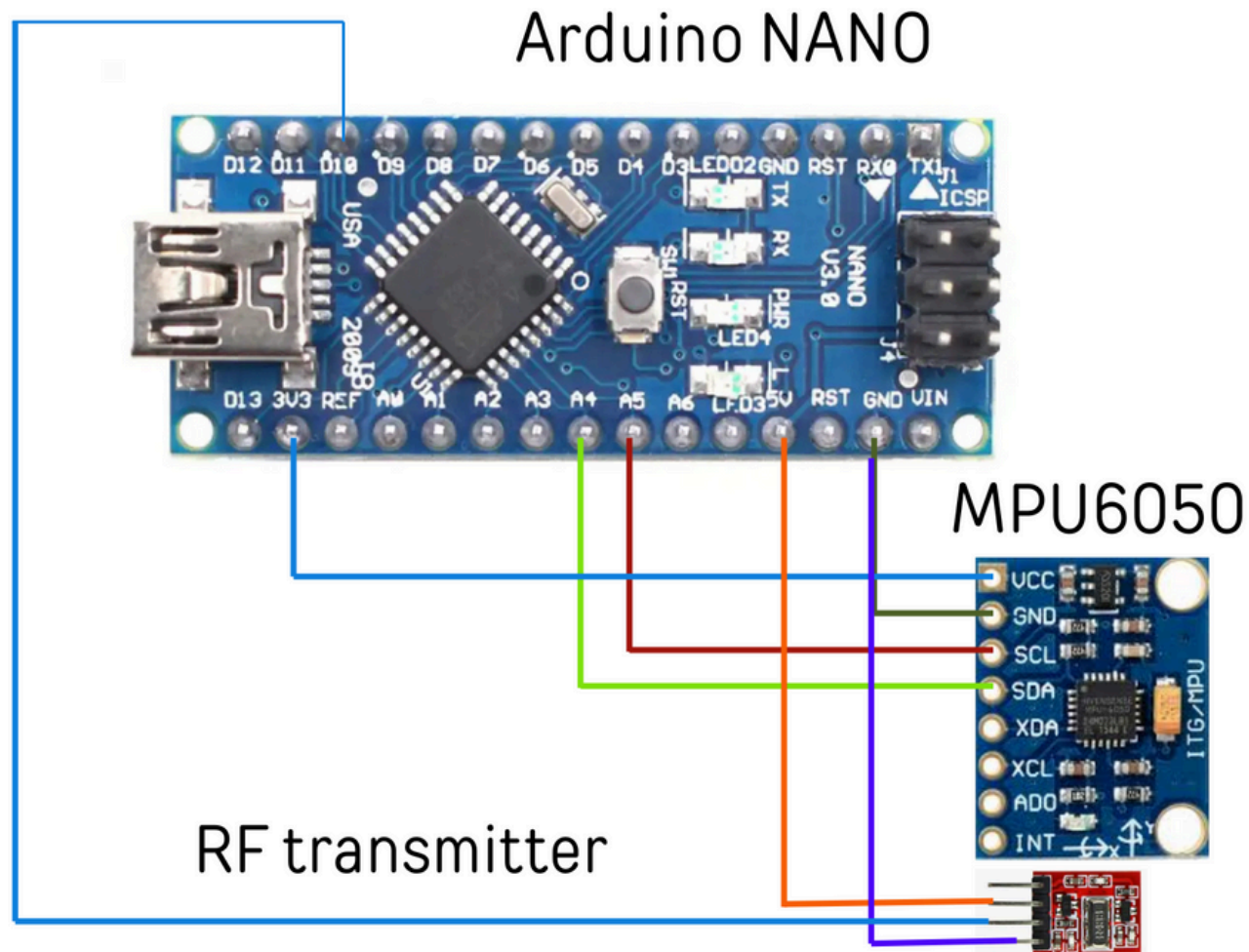
RF Module (, RF receiver)

RF transmitters and receiver is the backbone of wireless communication system, enabling devices to transmit voice, data, and multimedia content over the airwaves. They are vital for mobile phones, two-way connections. Wireless Internet



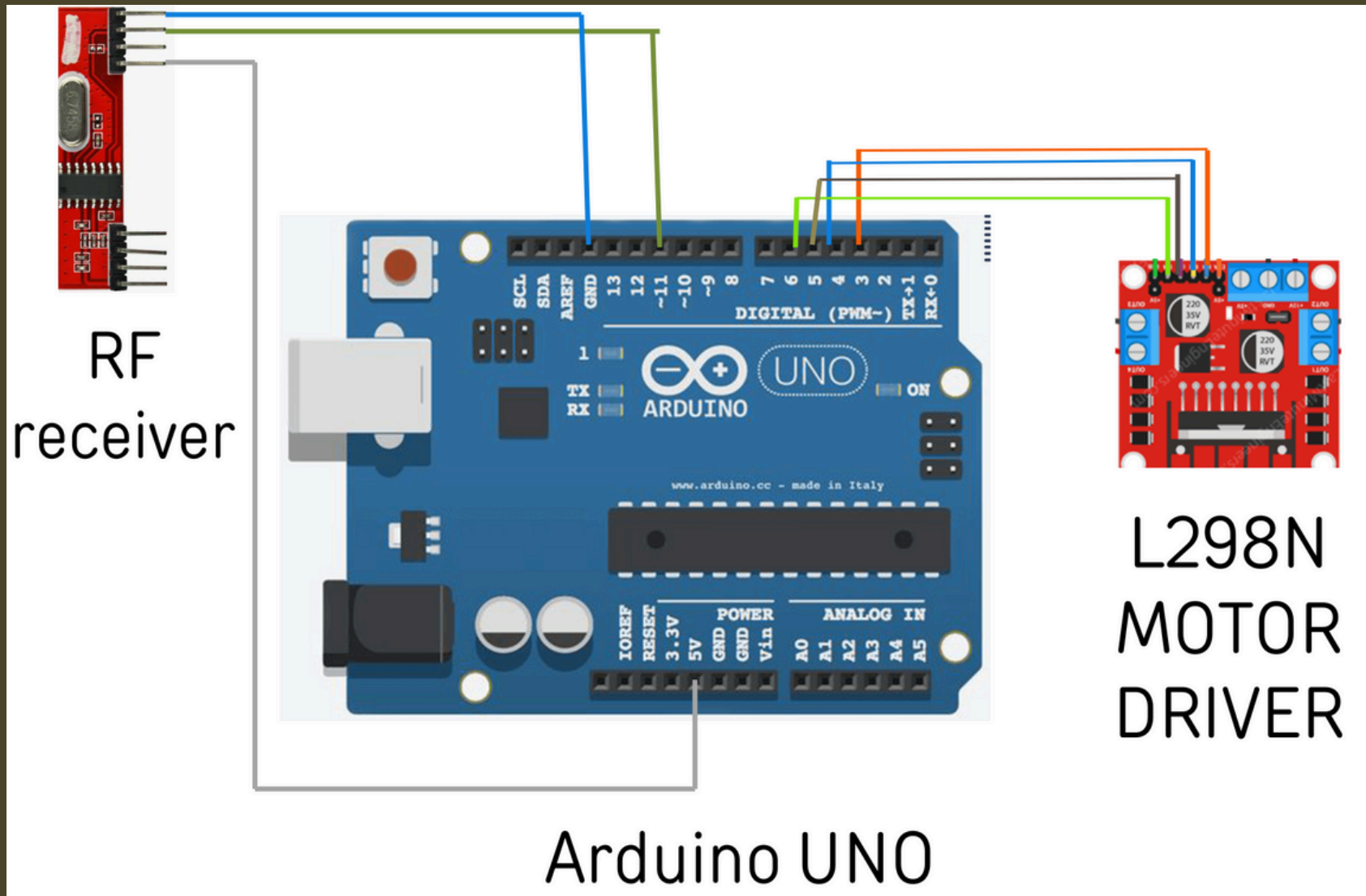


Circuit Diagram(TRANSMITTER)





Circuit Diagram(RECEIVER)





RECEIVER CODE

```
clude <VirtualWire.h>
#define IN1 3
#define IN2 4
#define IN3 5
#define IN4 6
void setup() {
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  vw_set_rx_pin(11);
  vw_setup(2000);
  vw_rx_start();
  Serial.println("Setup complete, waiting for commands...");
}
```

```
void loop() {
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
  if (vw_get_message(buf, &buflen)) {
    String command = "";
    for (int i = 0; i < buflen; i++) {
      command += (char)buf[i];
    }
    Serial.println("Command received: " + command);
    if (command == "FORWARD") {
      moveForward();
    } else if (command == "BACKWARD") {
      moveBackward();
    } else if (command == "LEFT") {
      turnLeft();
    } else if (command == "RIGHT") {
      turnRight();
    } else {
      stopMotors();
    }
  }
}
```



RECEIVER CODE

```
void moveForward() {  
  digitalWrite(IN1, HIGH);  
  digitalWrite(IN2, LOW);  
  digitalWrite(IN3, HIGH);  
  digitalWrite(IN4, LOW);  
  Serial.println("Moving forward");  
}  
  
void moveBackward() {  
  digitalWrite(IN1, LOW);  
  digitalWrite(IN2, HIGH);  
  digitalWrite(IN3, LOW);  
  digitalWrite(IN4, HIGH);  
  Serial.println("Moving backward");  
}
```

```
void turnLeft() {  
  digitalWrite(IN1, LOW);  
  digitalWrite(IN2, HIGH);  
  digitalWrite(IN3, HIGH);  
  digitalWrite(IN4, LOW);  
  Serial.println("Turning left");  
}
```

```
void turnRight() {  
  digitalWrite(IN1, HIGH);  
  digitalWrite(IN2, LOW);  
  digitalWrite(IN3, LOW);  
  digitalWrite(IN4, HIGH);  
  Serial.println("Turning right");  
}
```

```
void stopMotors() {  
  digitalWrite(IN1, LOW);  
  digitalWrite(IN2, LOW);  
  digitalWrite(IN3, LOW);  
  digitalWrite(IN4, LOW);  
  Serial.println("Motors stopped");  
}
```



TRANSMITTER CODE

```
#include <Wire.h>
#include <MPU6050.h>
#include <VirtualWire.h>
MPU6050 mpu;
void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection
failed!");
    while (1);
  }
  Serial.println("MPU6050 initialized.");
  vw_set_tx_pin(10);
  vw_setup(2000);
}
```

```
void loop() {
  int16_t ax, ay, az;
  mpu.getAcceleration(&ax, &ay, &az);
  String command = "";
  if (ay > 15000) command = "FORWARD";
  else if (ay < -15000) command =
    "BACKWARD";
  else if (ax > 15000) command = "LEFT";
  else if (ax < -15000) command = "RIGHT";
  else command = "STOP";
  vw_send((uint8_t *)command.c_str(),
    command.length());
  vw_wait_tx();
  Serial.println("Command sent: " +
    command);
  delay(500);
}
```




WORKING

- Utilizes an MPU6050 accelerometer to detect hand gestures.
- Transmitter converts hand tilts into commands like "FORWARD," "BACKWARD," "LEFT," or "RIGHT."
- Commands are sent wirelessly to the receiver via an RF module.
- The receiver processes commands to control the car's motors using a motor driver.
- Hand gestures direct the car's movements, and neutral position stops it.