

BDD BLUETOOTH CAR CHASSIS KIT

Step 1: Secure the motor

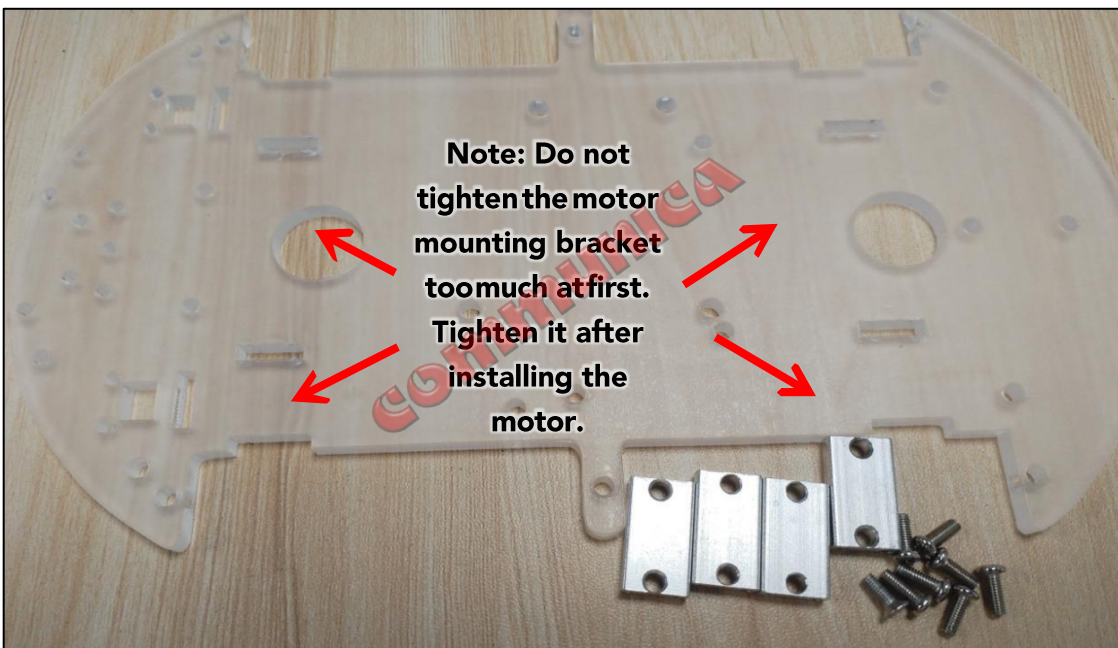
Step 1.1: Solder the positive and negative wires to the motor



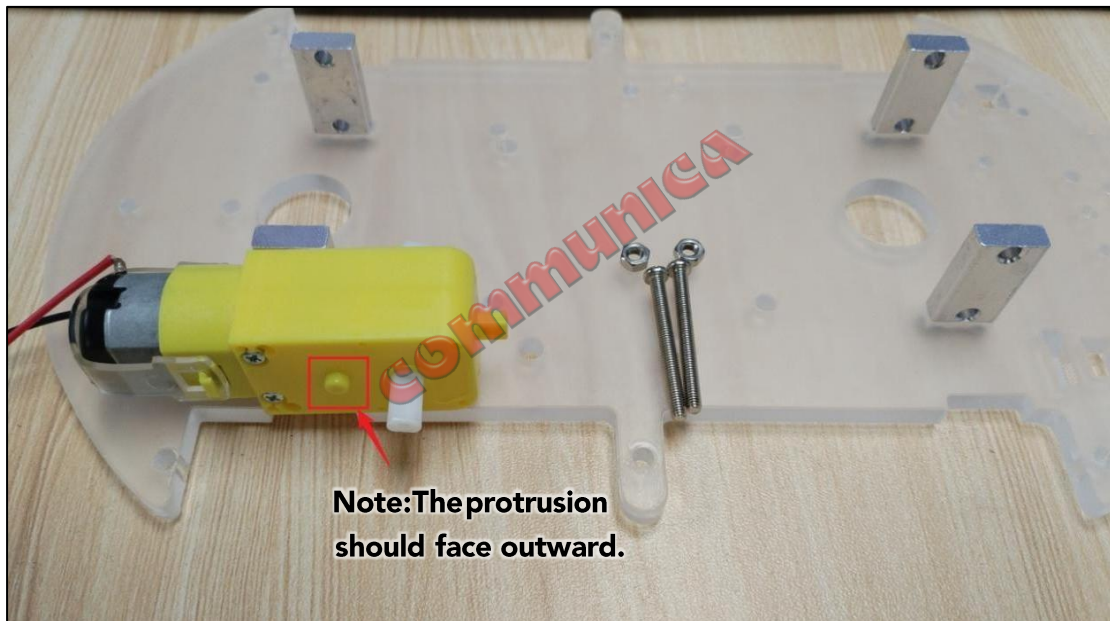
Solder the other three motors as well.



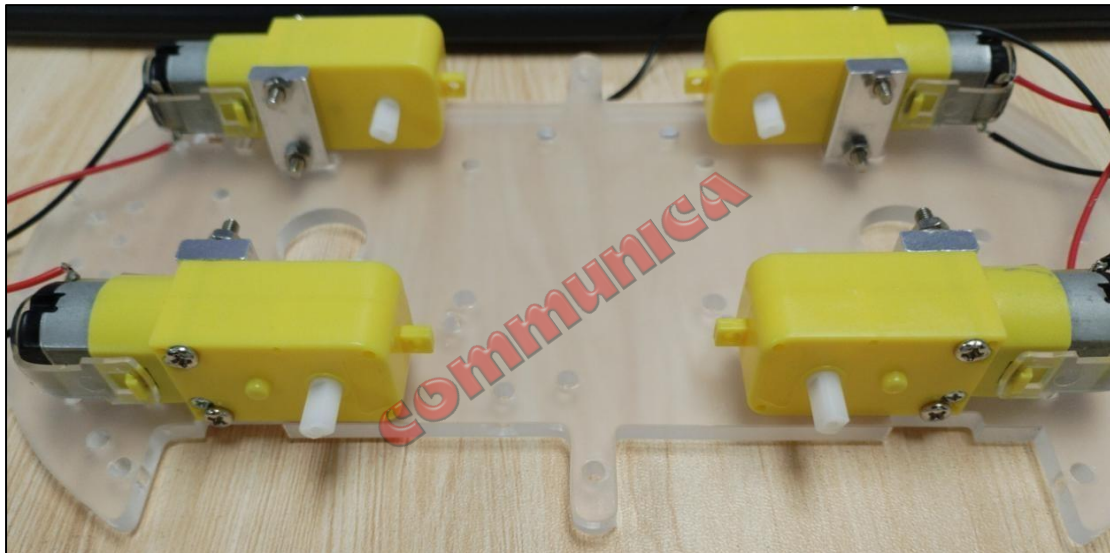
Step 1.2: Install the motor mounting bracket onto the acrylic plate.



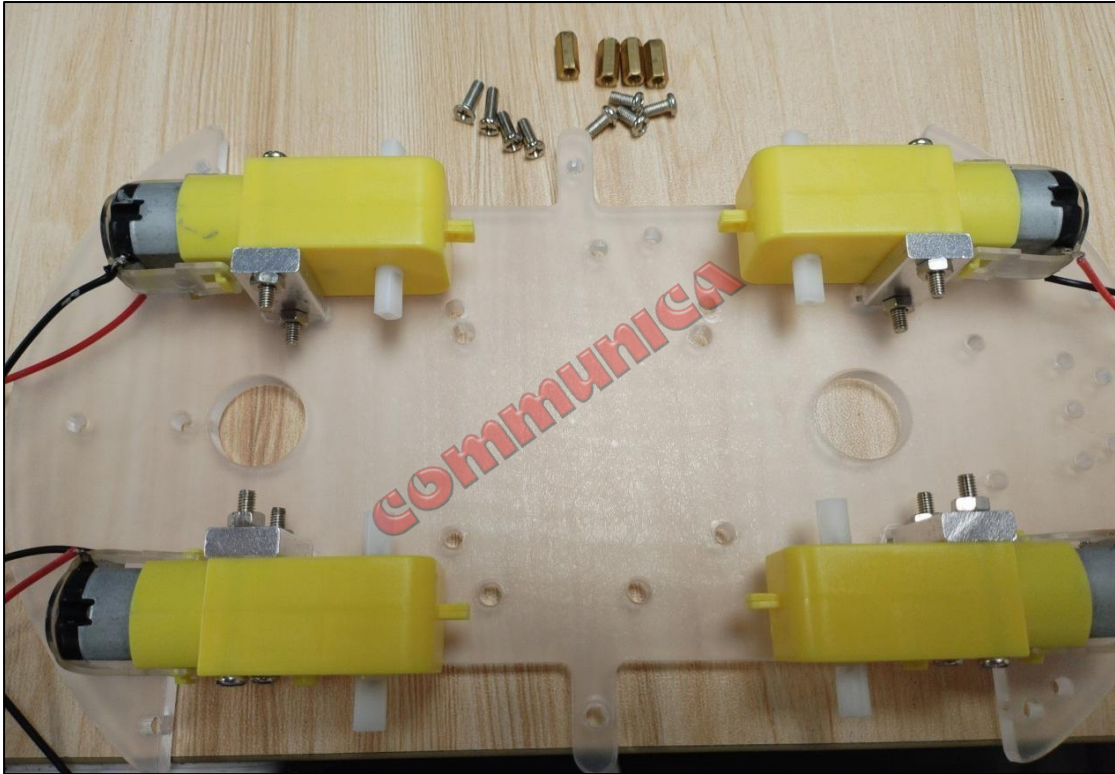
Step 1.3: Mount the motor onto the motor mounting bracket.



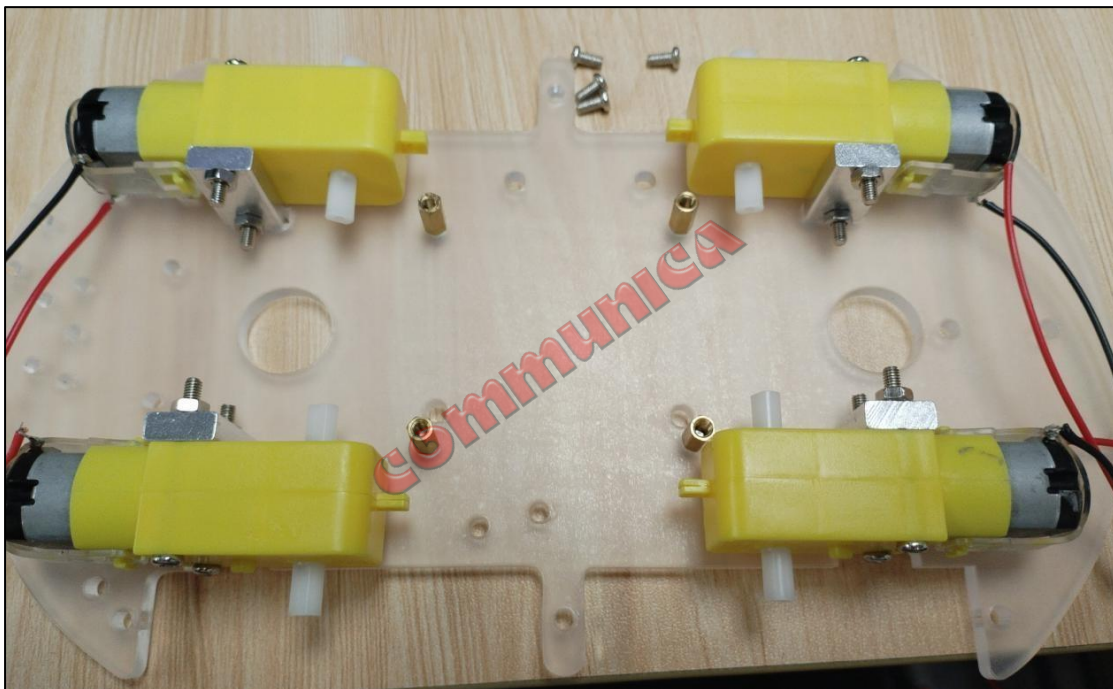
Next, install the remaining three motors onto the motor mounting brackets.



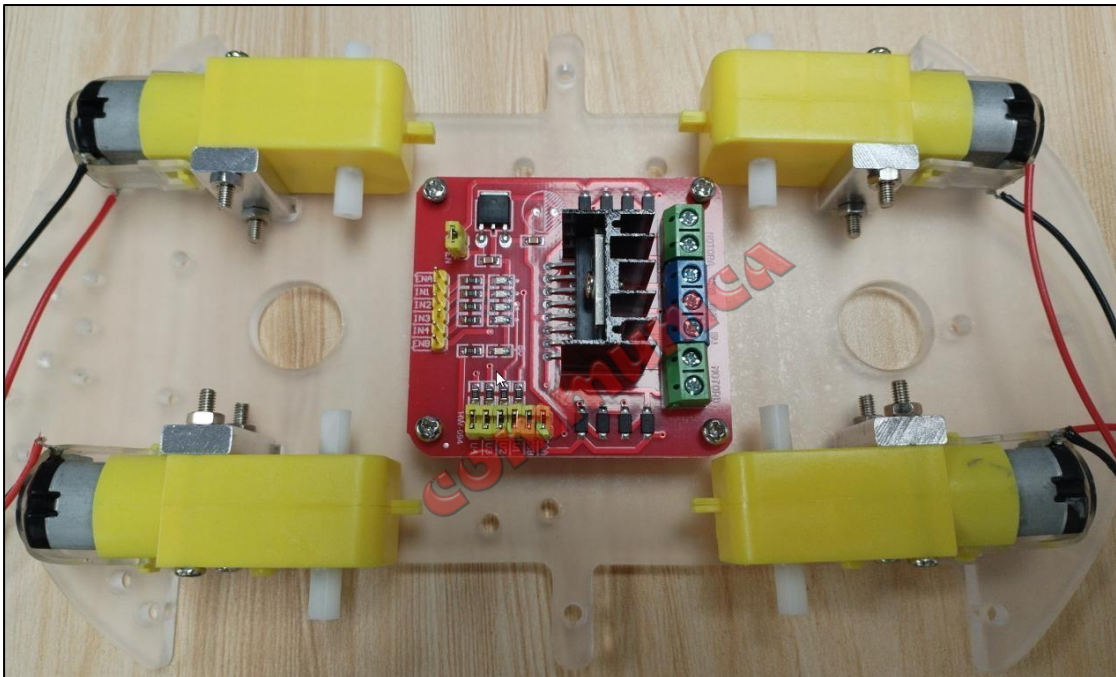
Step 2: Install the motor driver board



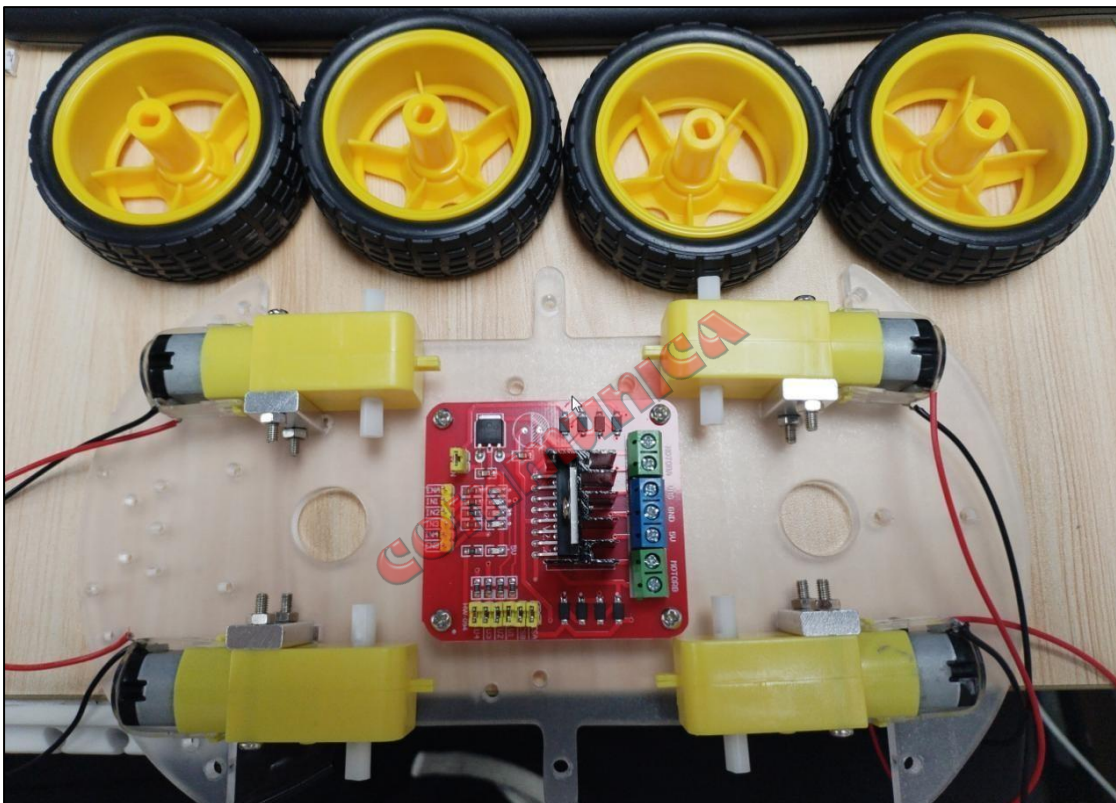
Step 2.1: Install the dual-pass copper standoffs onto the acrylic plate.

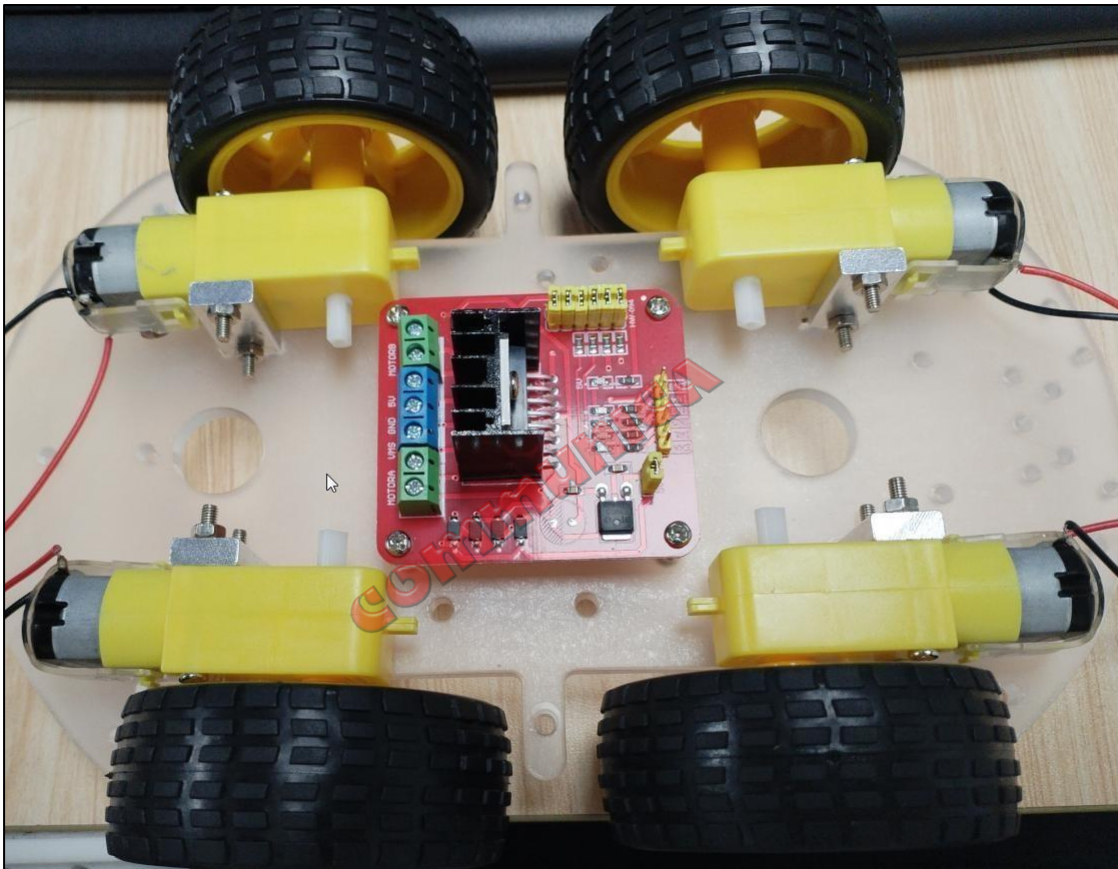


Step 2.2: Mount the motor driver board onto the dual-pass copper standoffs.

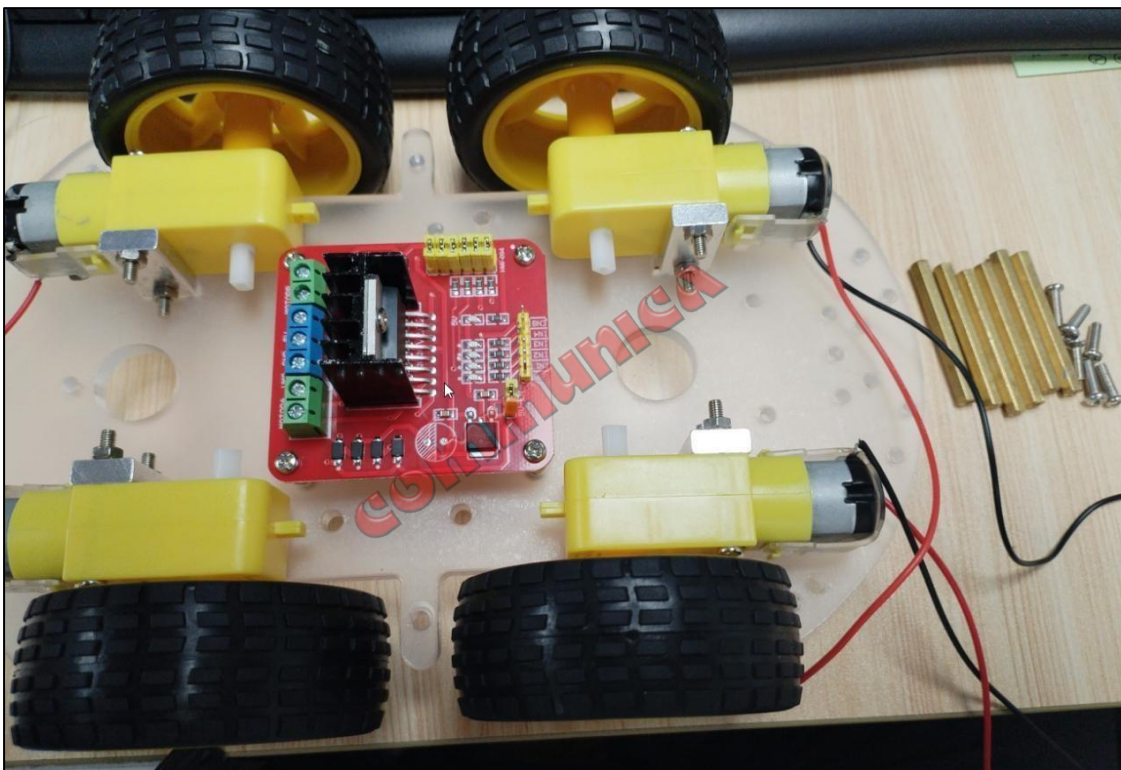


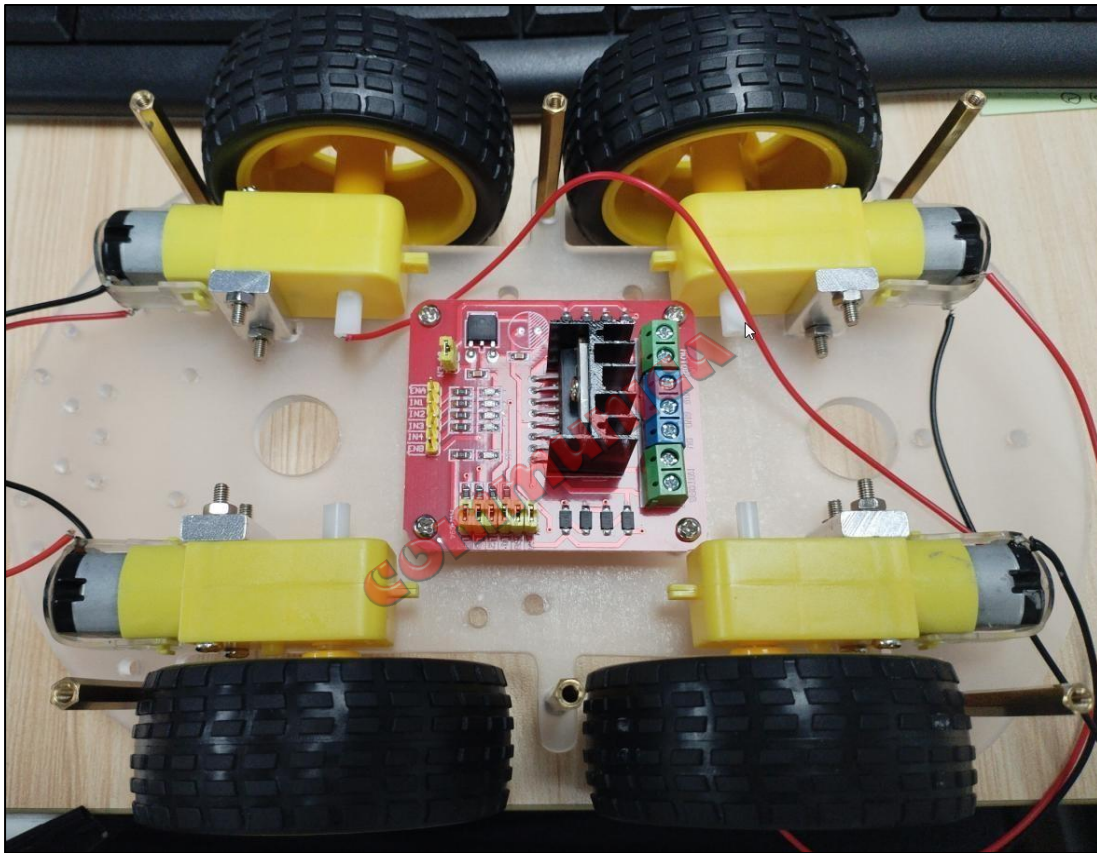
Step 3: Install the rubber wheels



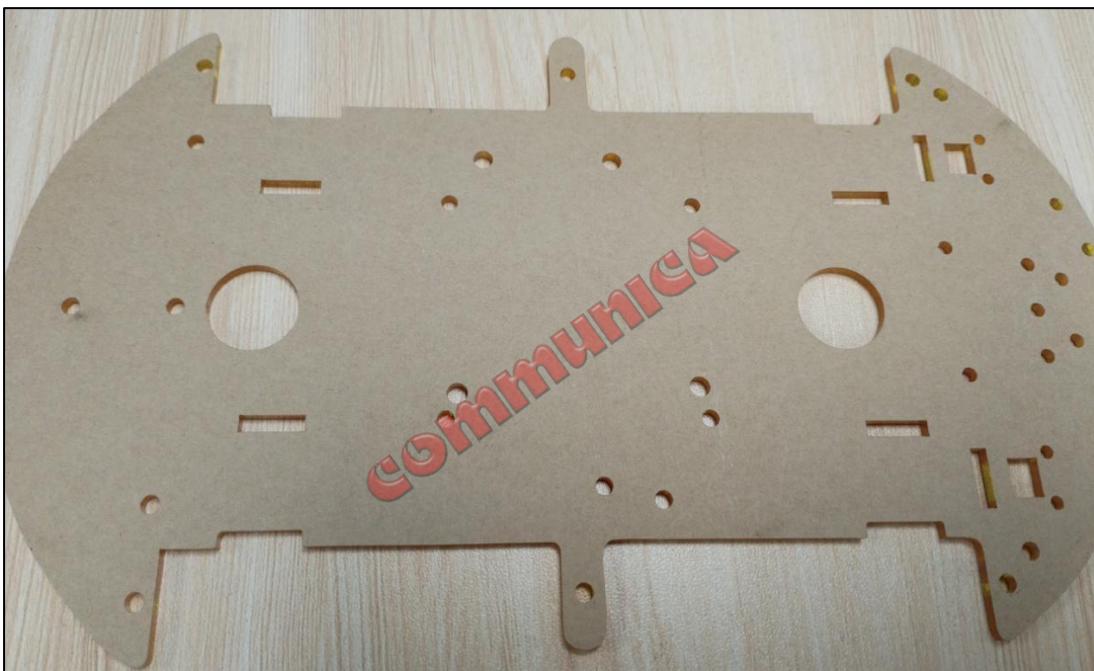


Step 4: Install support copper pillars





Step 5: Peel off the acrylic film

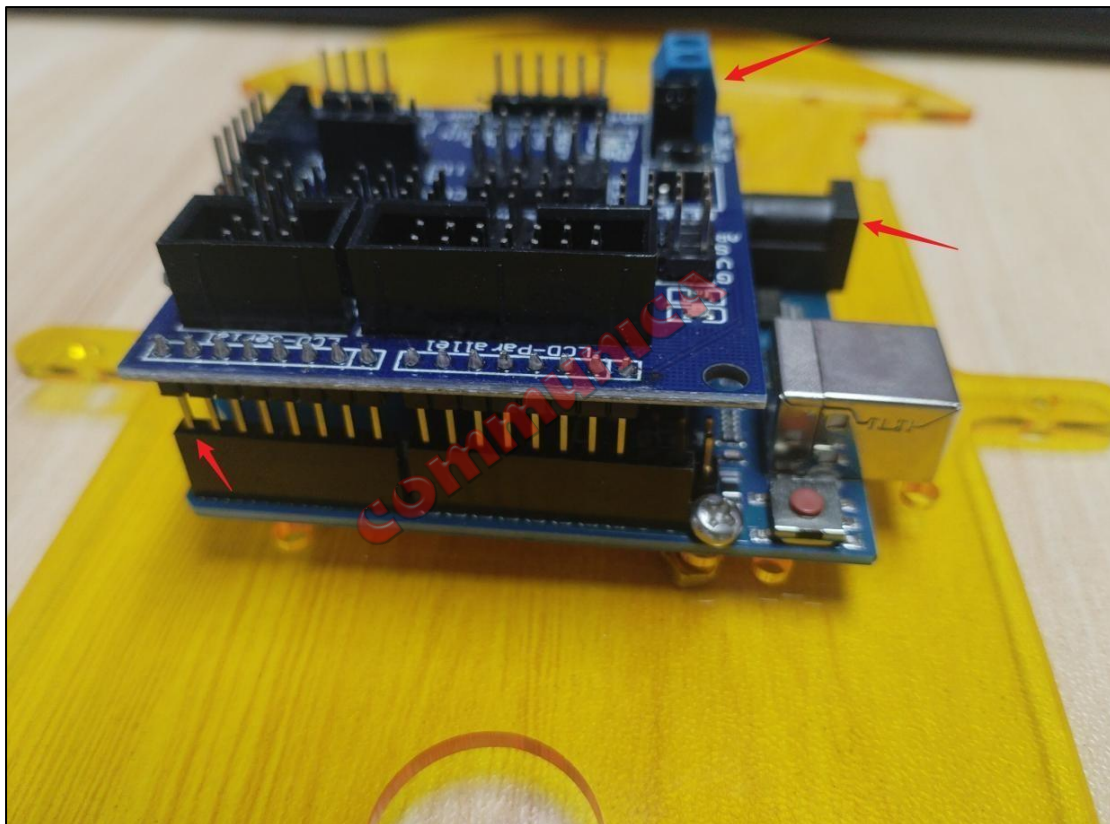
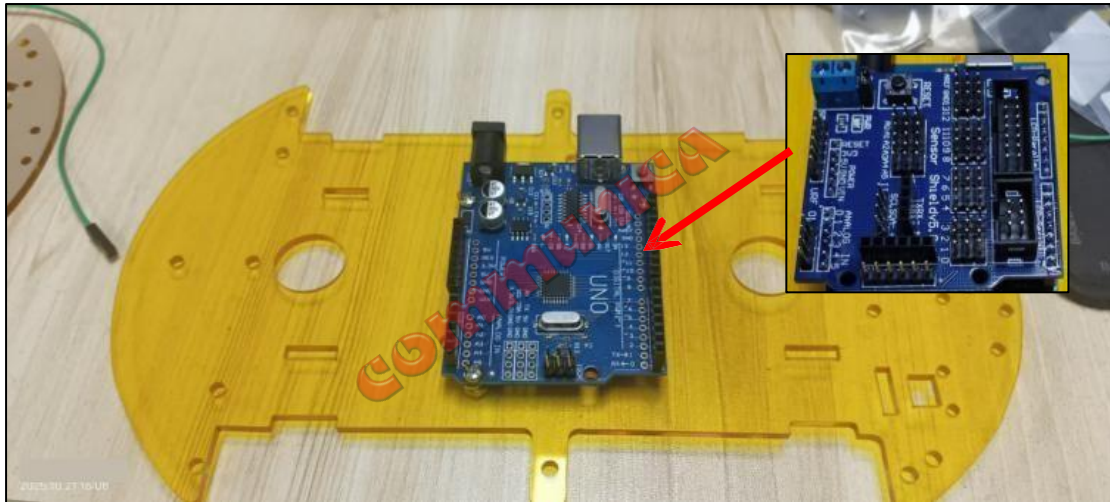




Step 6: Install the Arduino controller board



Step 7: Install the Arduino expansion board



Step 8: Install the servo pan-tilt



Step 8.1: Take out the cross-shaped servo horn.



Step 8.2: Cut the cross-shaped servo horn into the shape shown in the image.

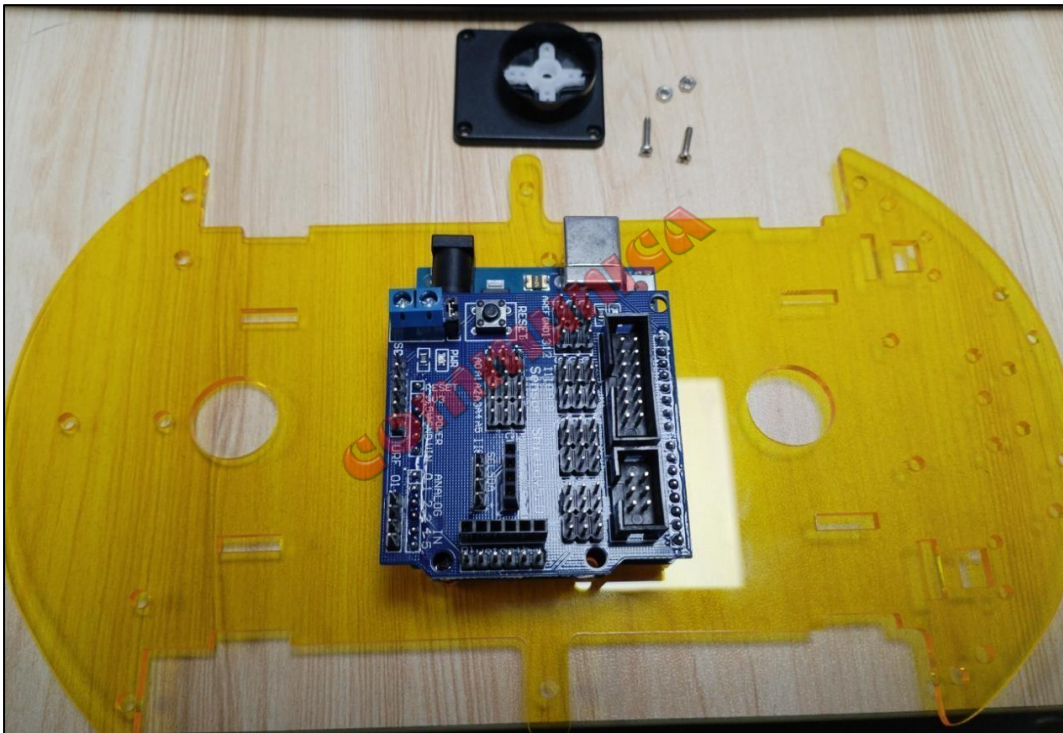


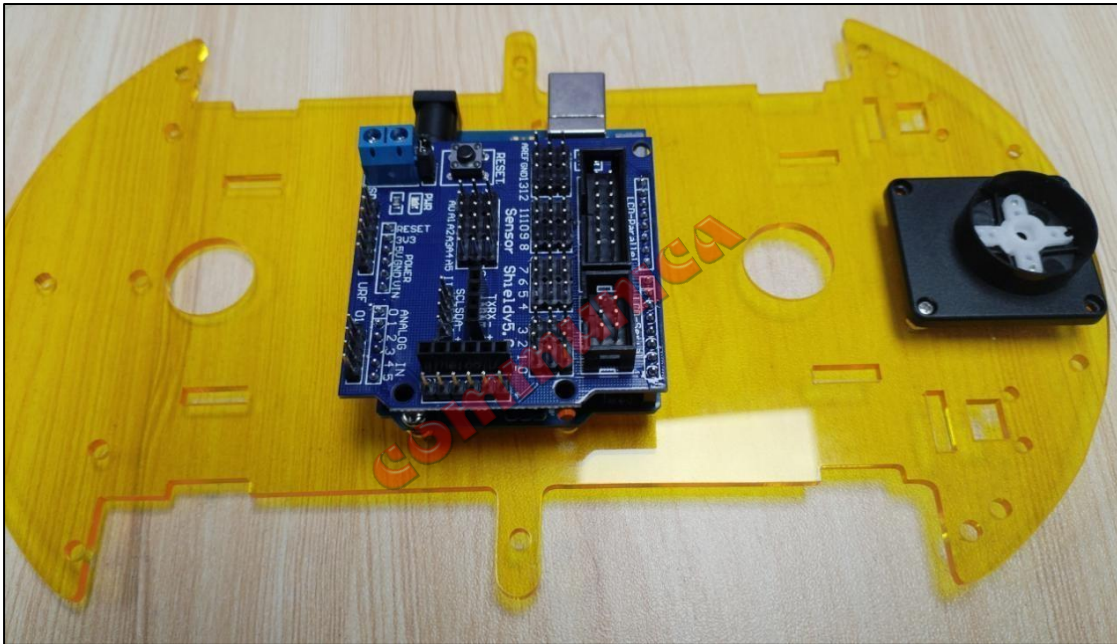
Step 8.3: Install the servo horn onto the pan-tilt base.





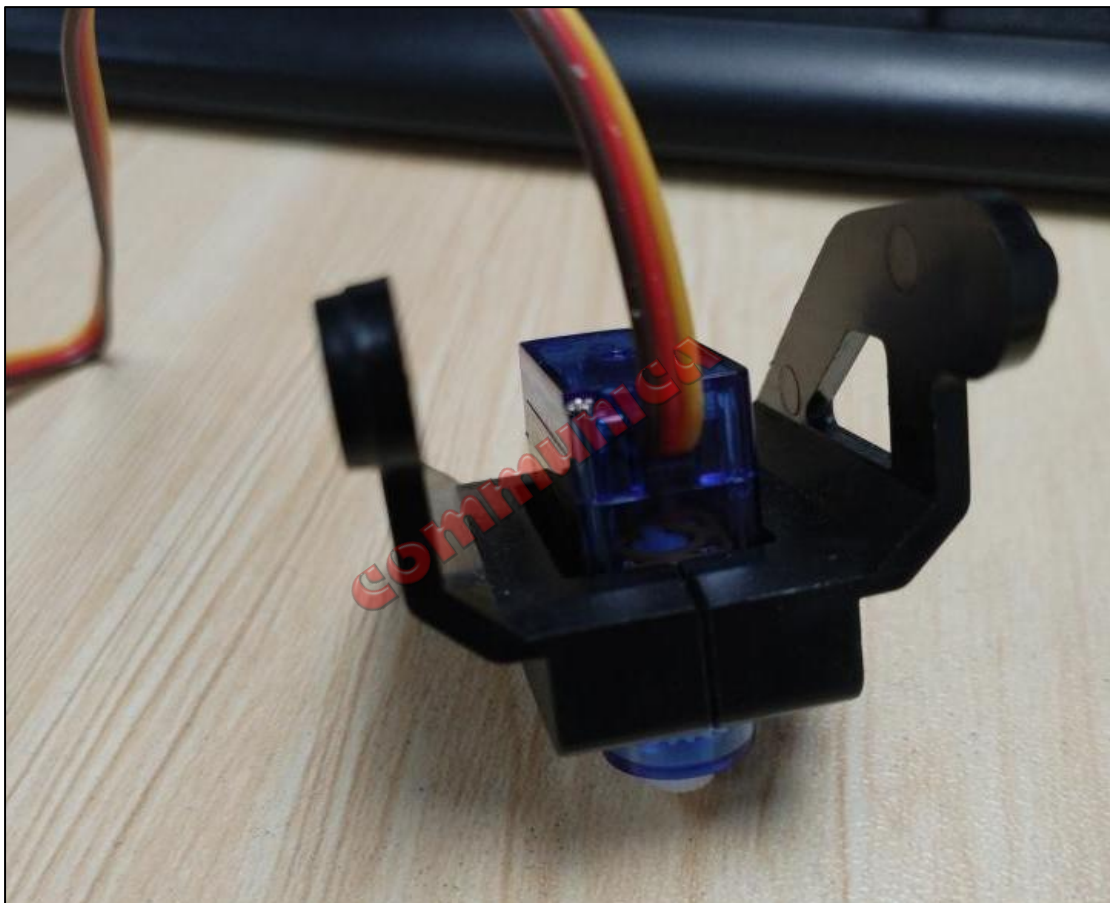
Step 8.4: Install the servo pan-tilt onto the acrylic board.



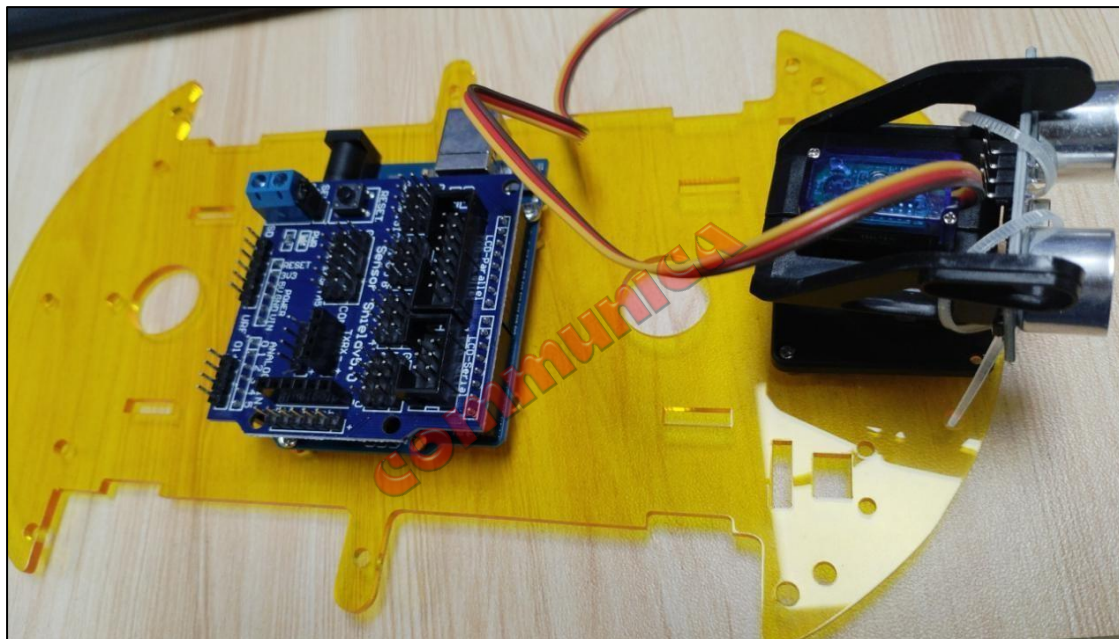
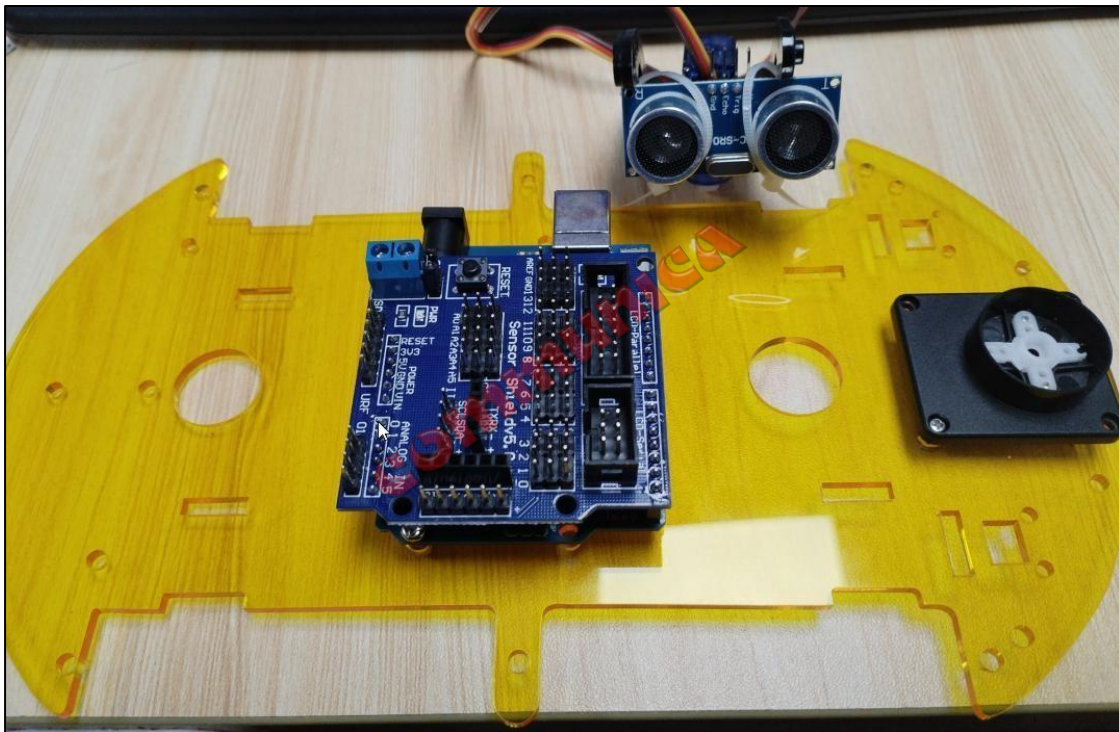


Step 8.5: Install the ultrasonic module onto the pan-tilt bracket.

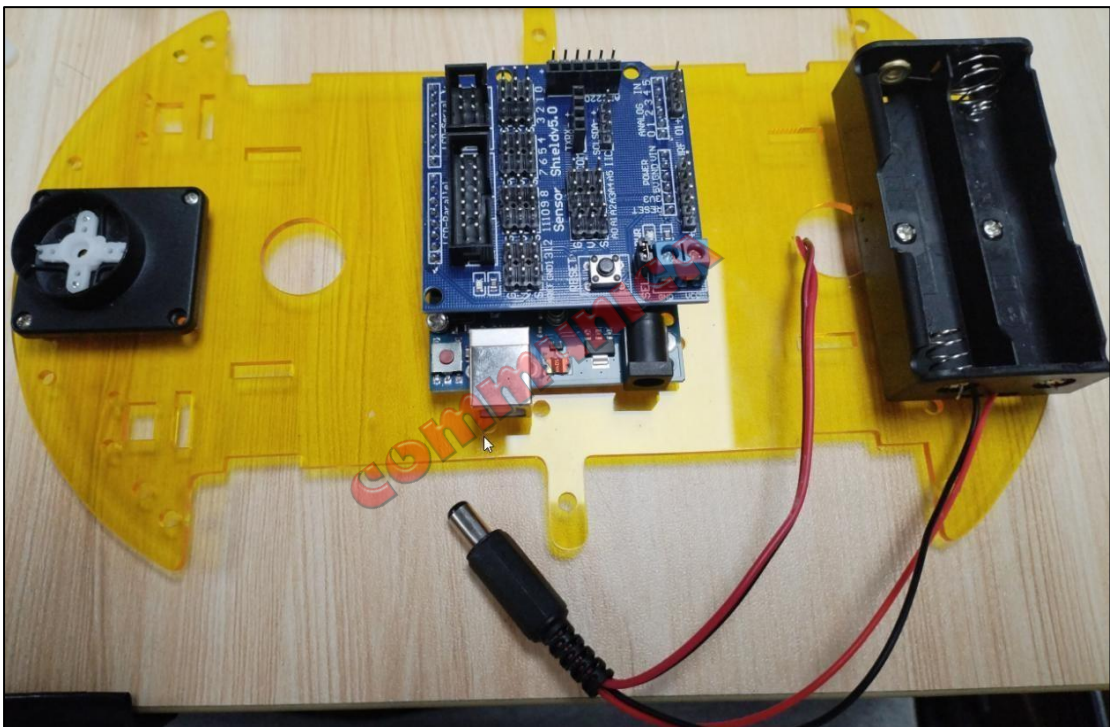
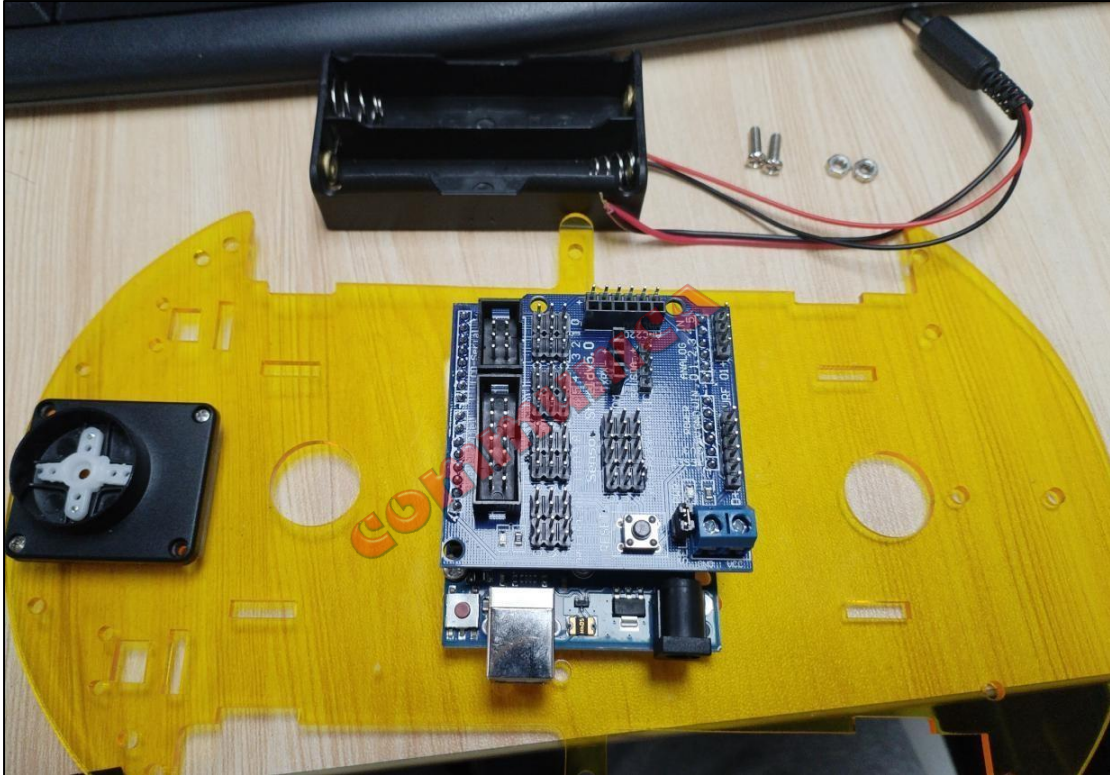




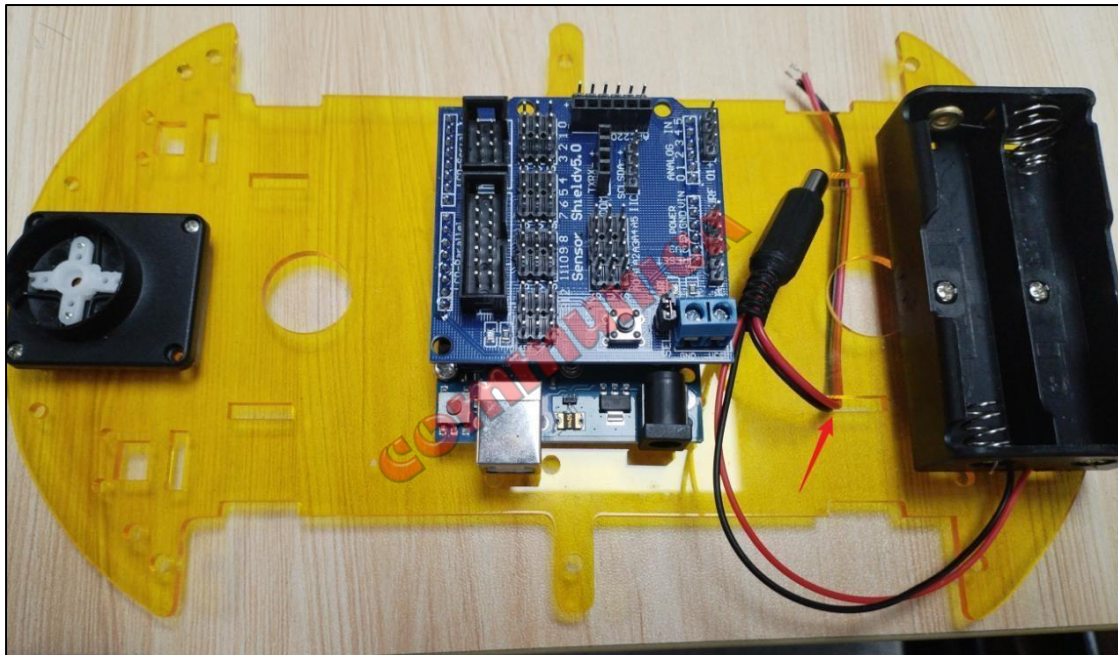




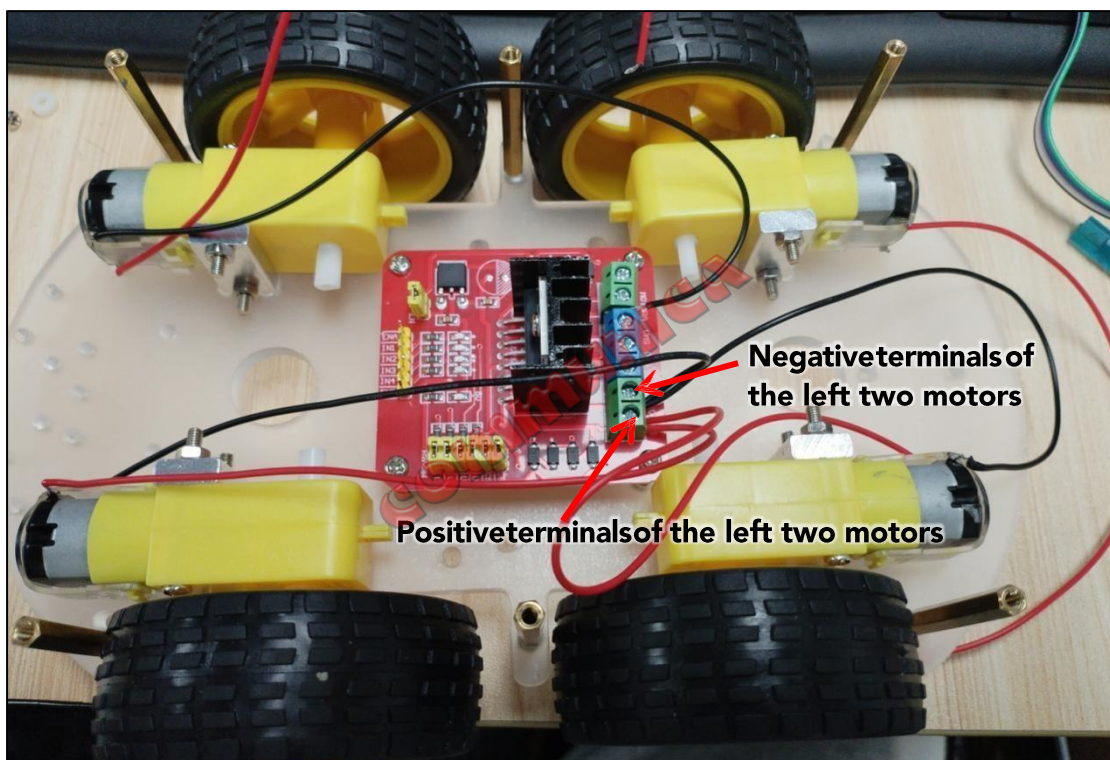
Step 9: Install the battery holder

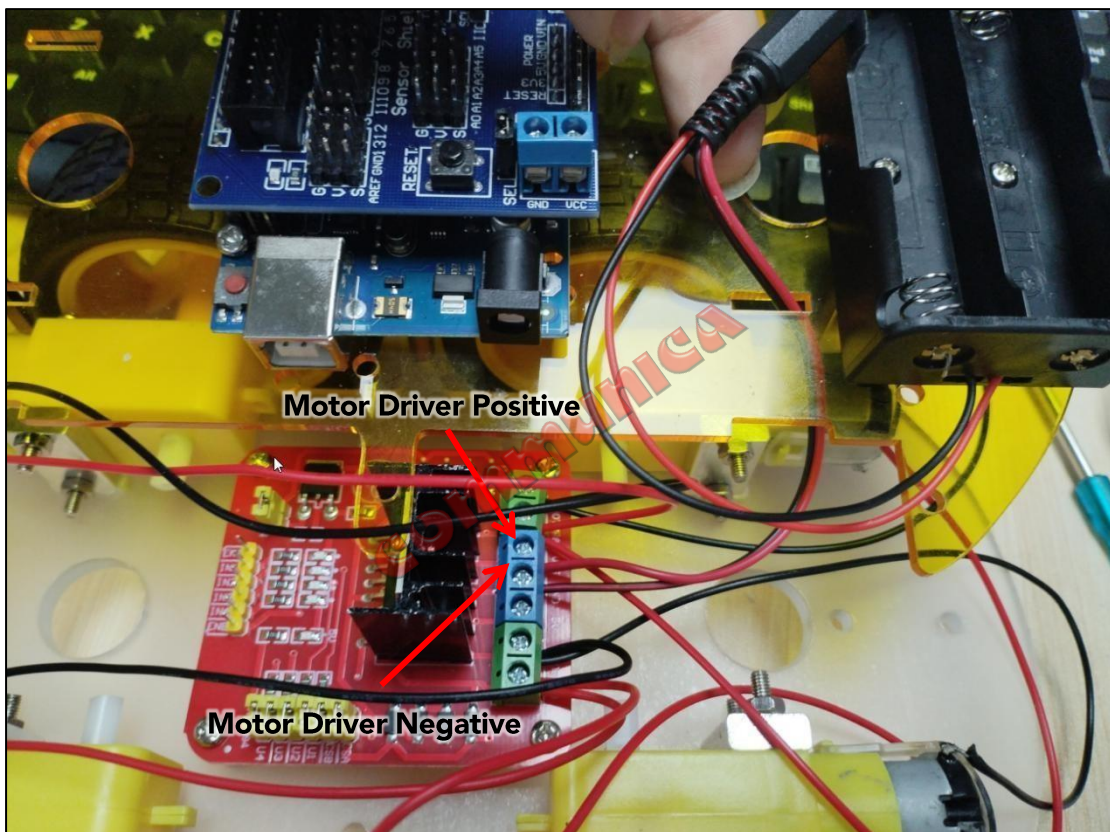
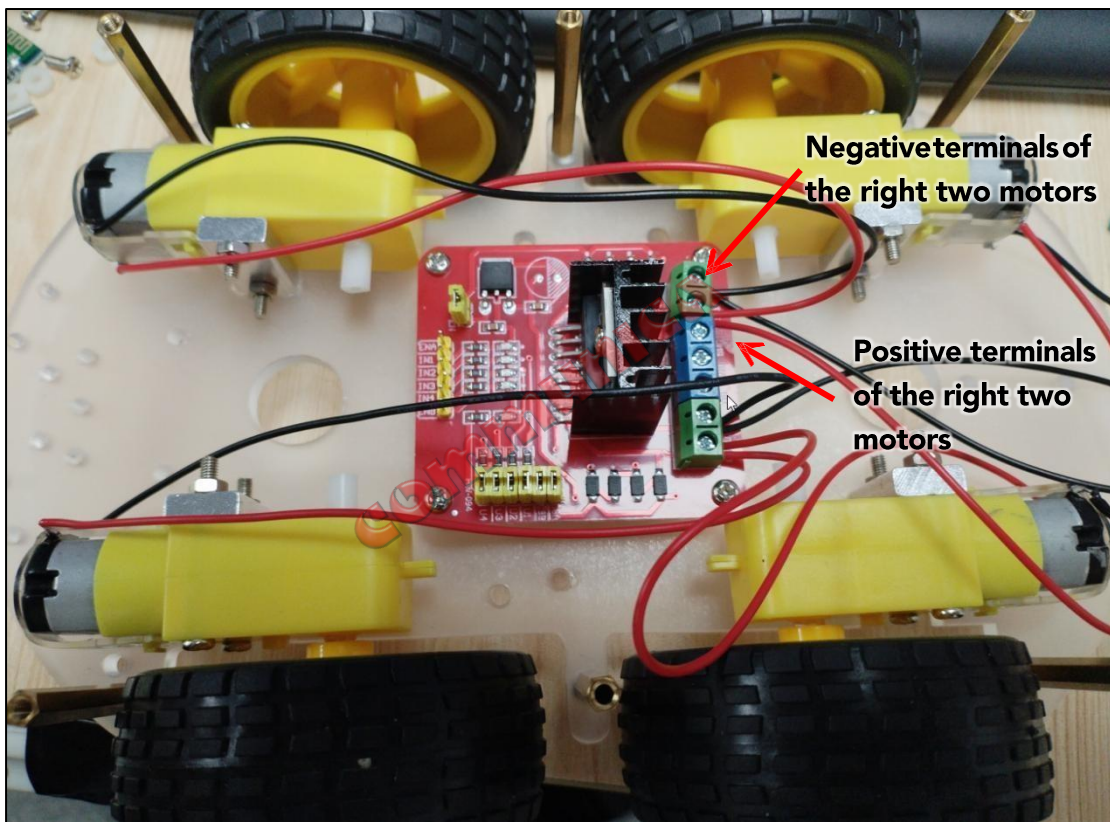


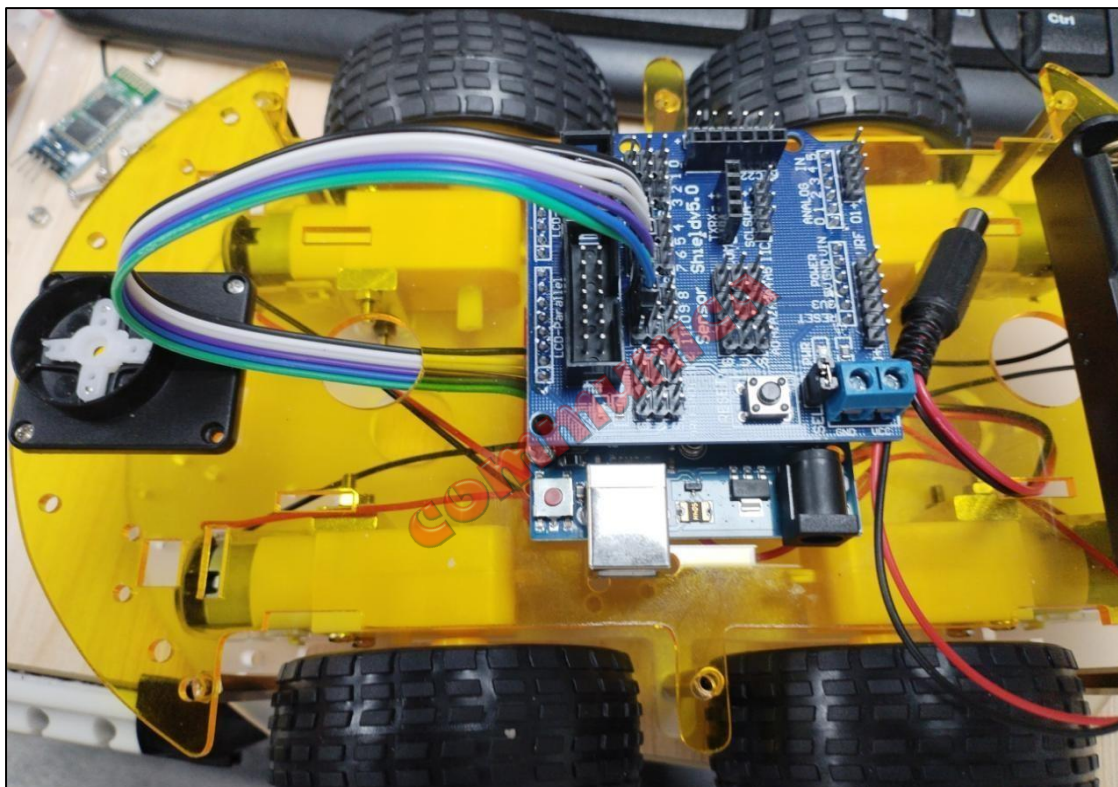
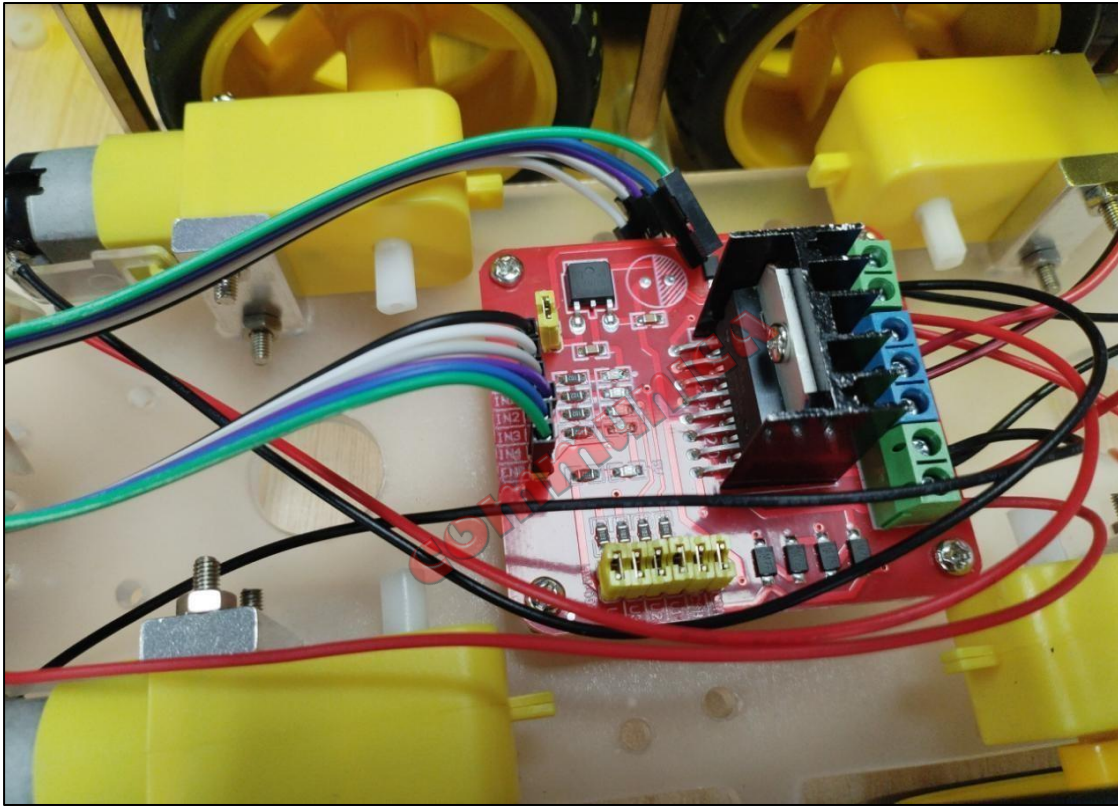
Pass the power cable of the motor driver board through the hole indicated by the arrow



Step 10: Connect the motor power cable



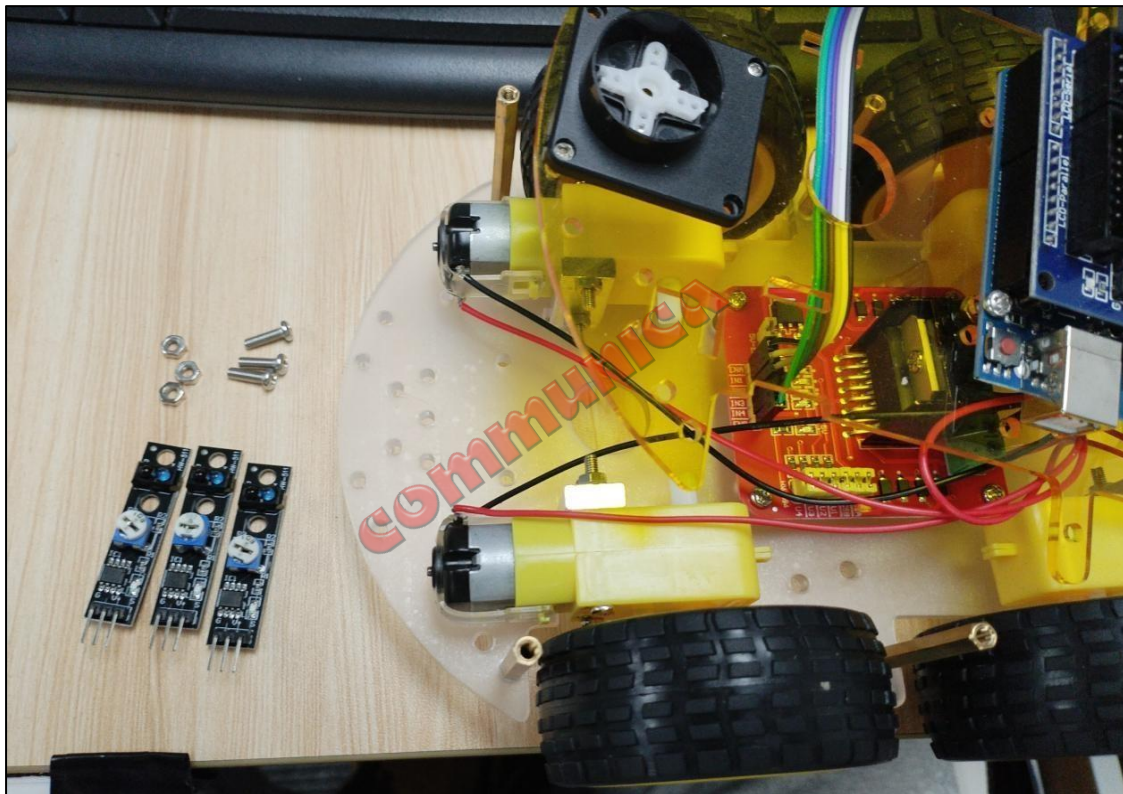


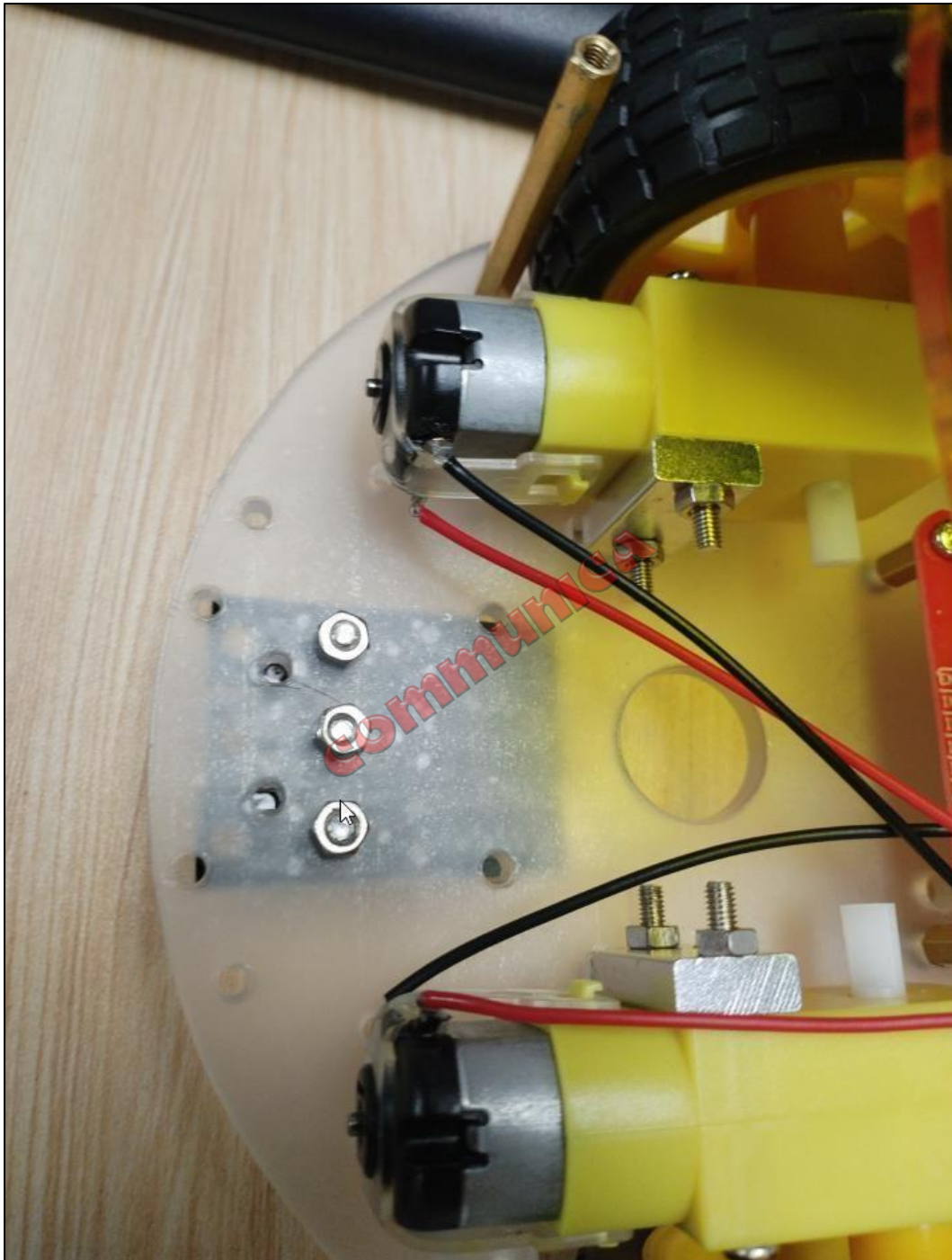


Pin connection table

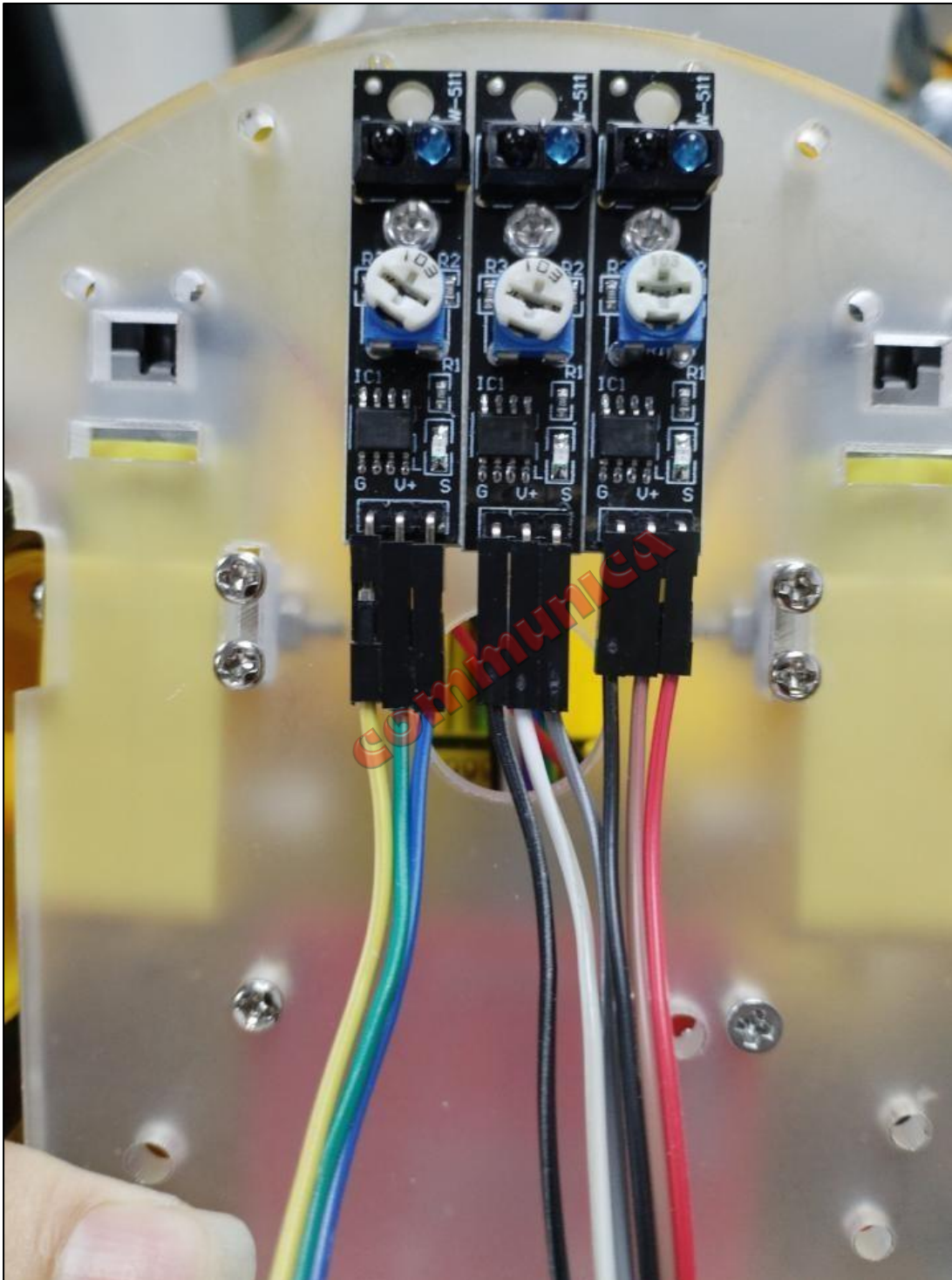
Arduino	L298N
5	ENA
6	IN1
7	IN2
8	IN3
9	IN4
10	ENB

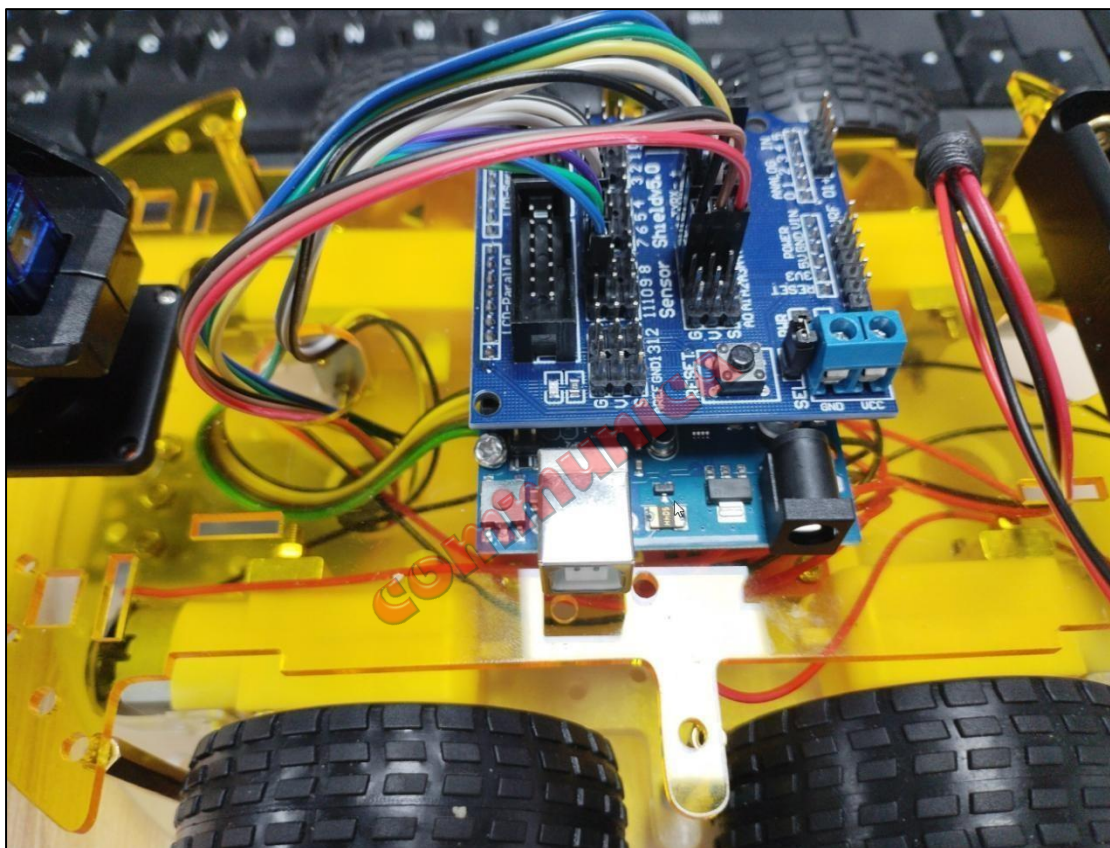
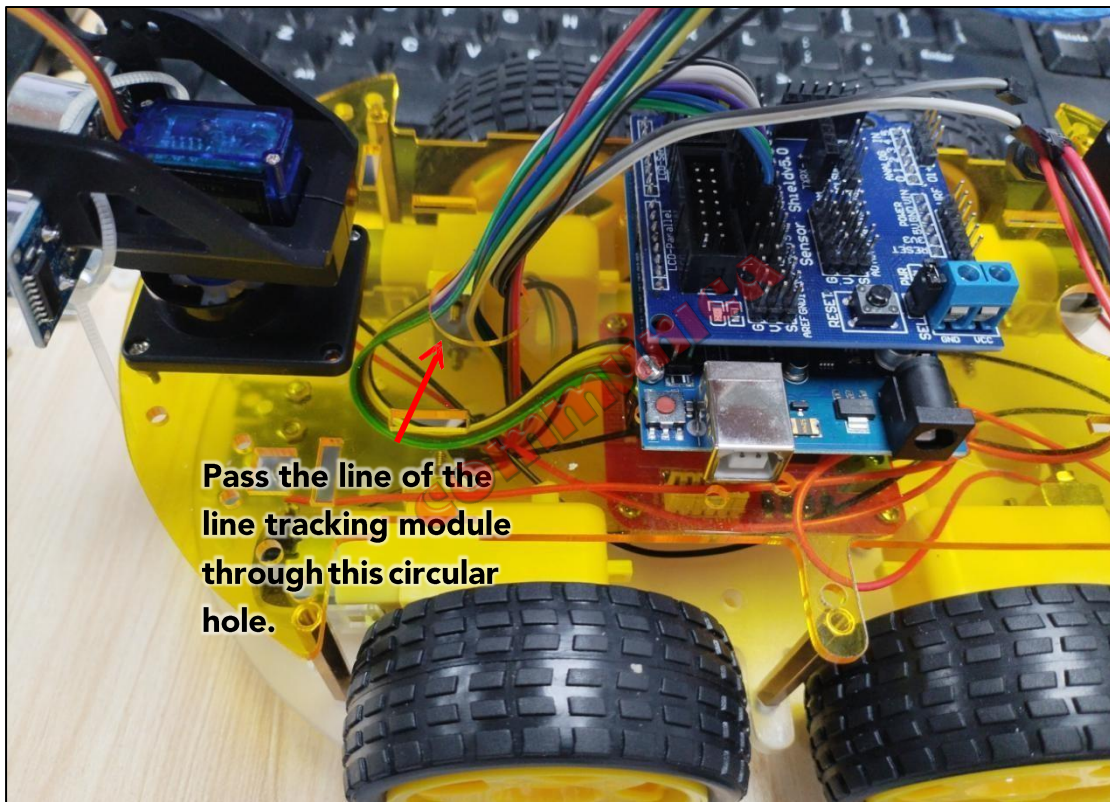
Step 11: Install the line tracking module







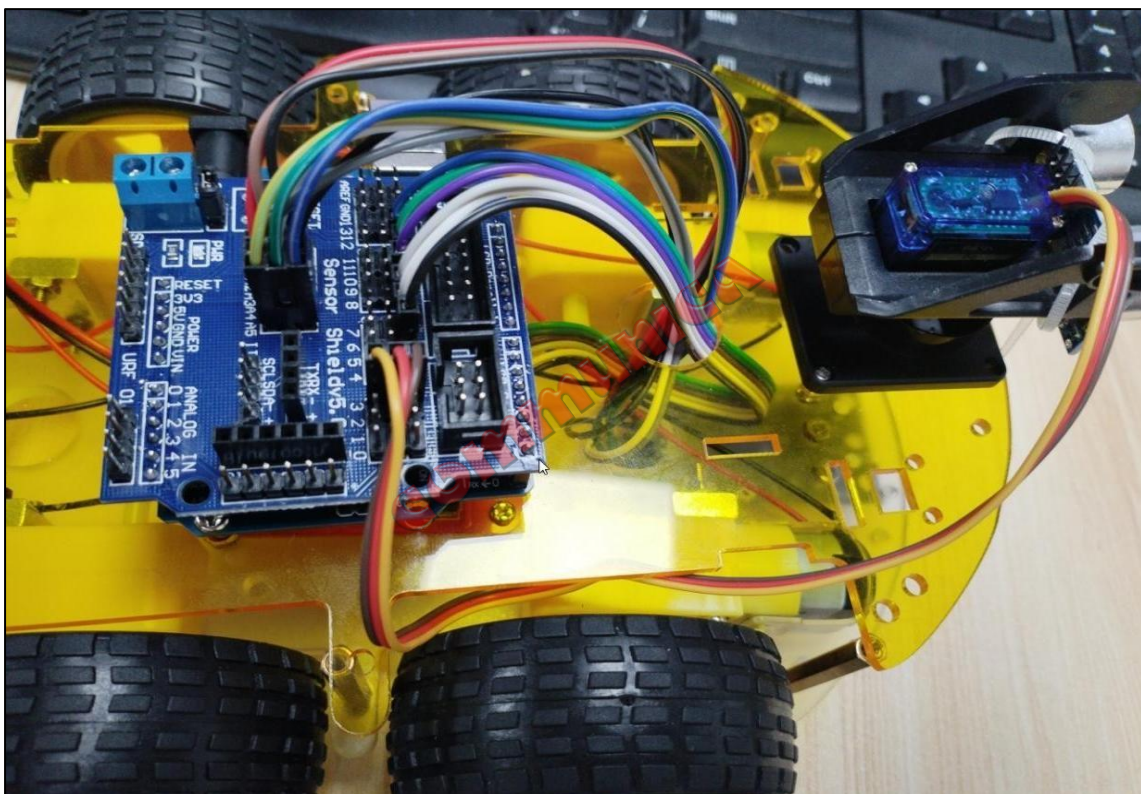


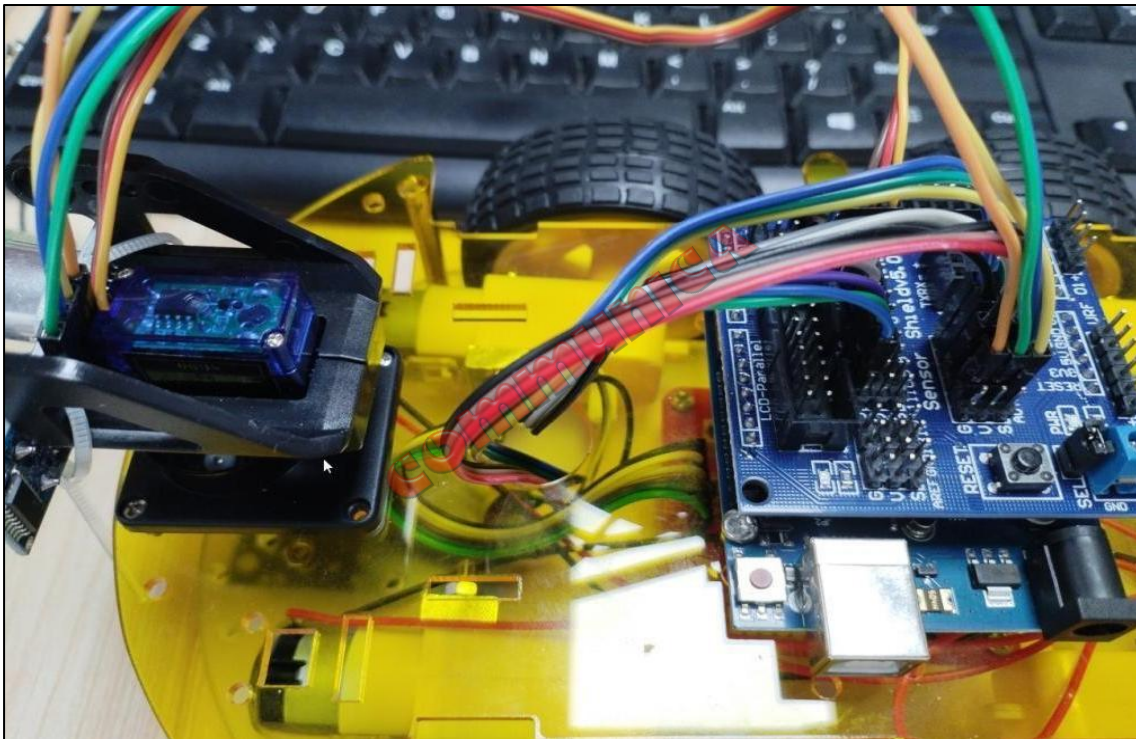


Pin connection table

Module	Tracking module	Arduino
Left tracking module	G	G
	V	V
	S	A3
Center tracking module	G	G
	V	V
	S	A4
Right tracking module	G	G
	V	V
	S	A5

Step 12: Connect the servo to the controller



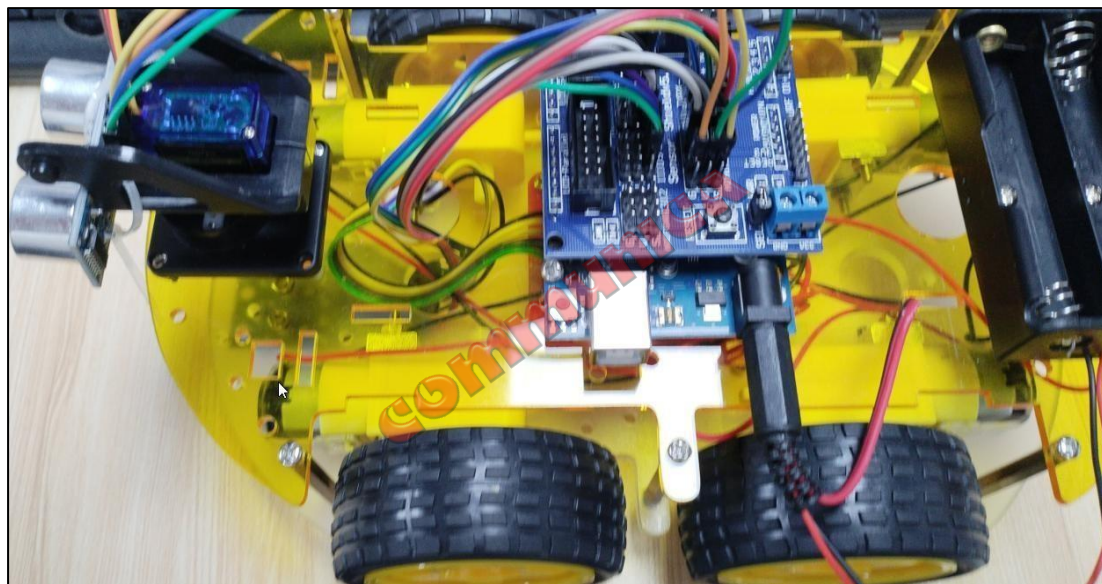
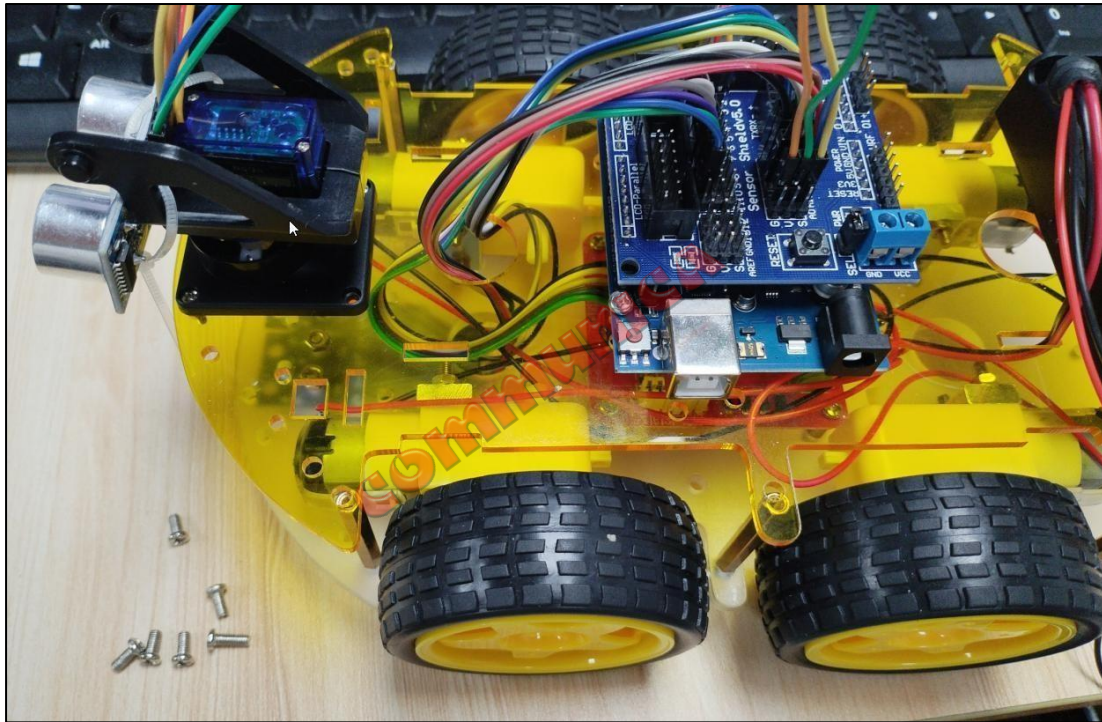


Pin connection table

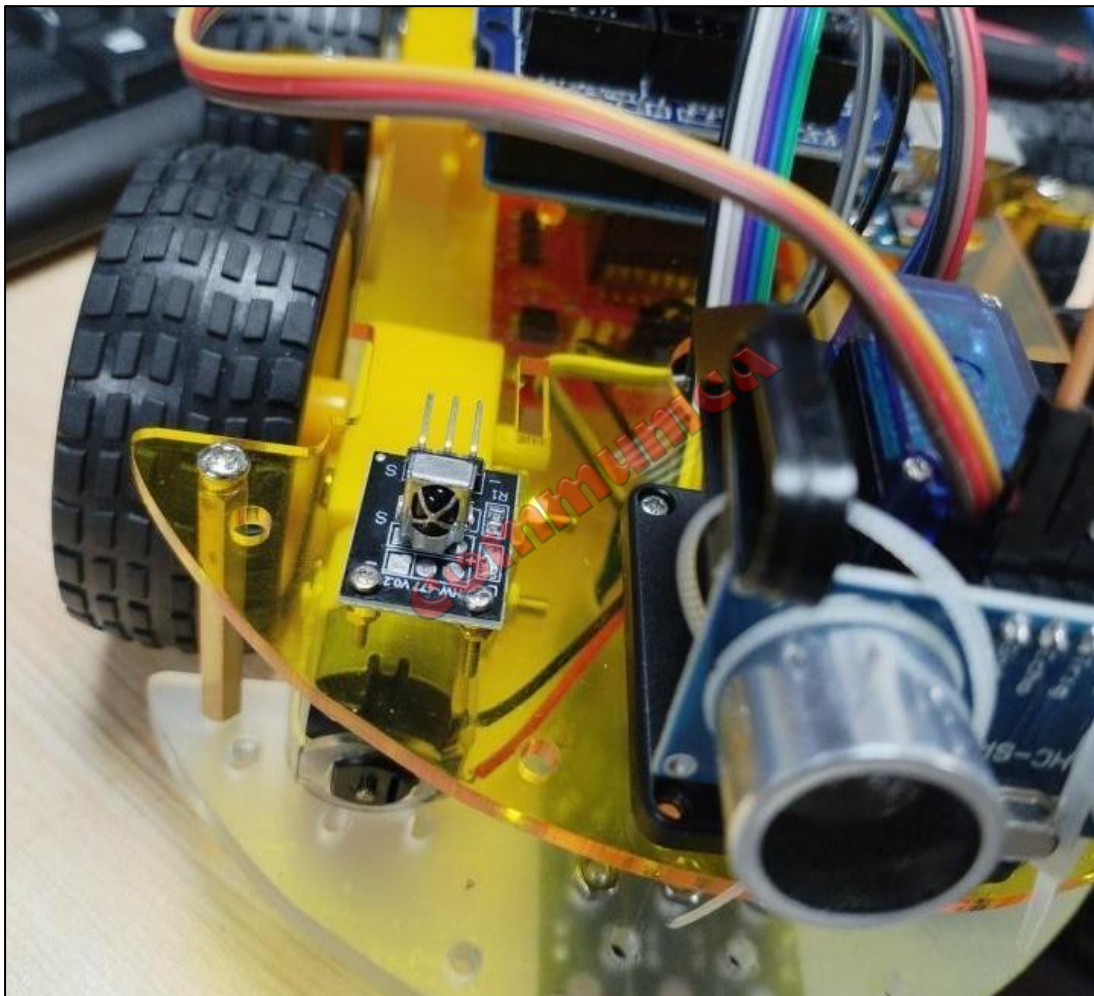
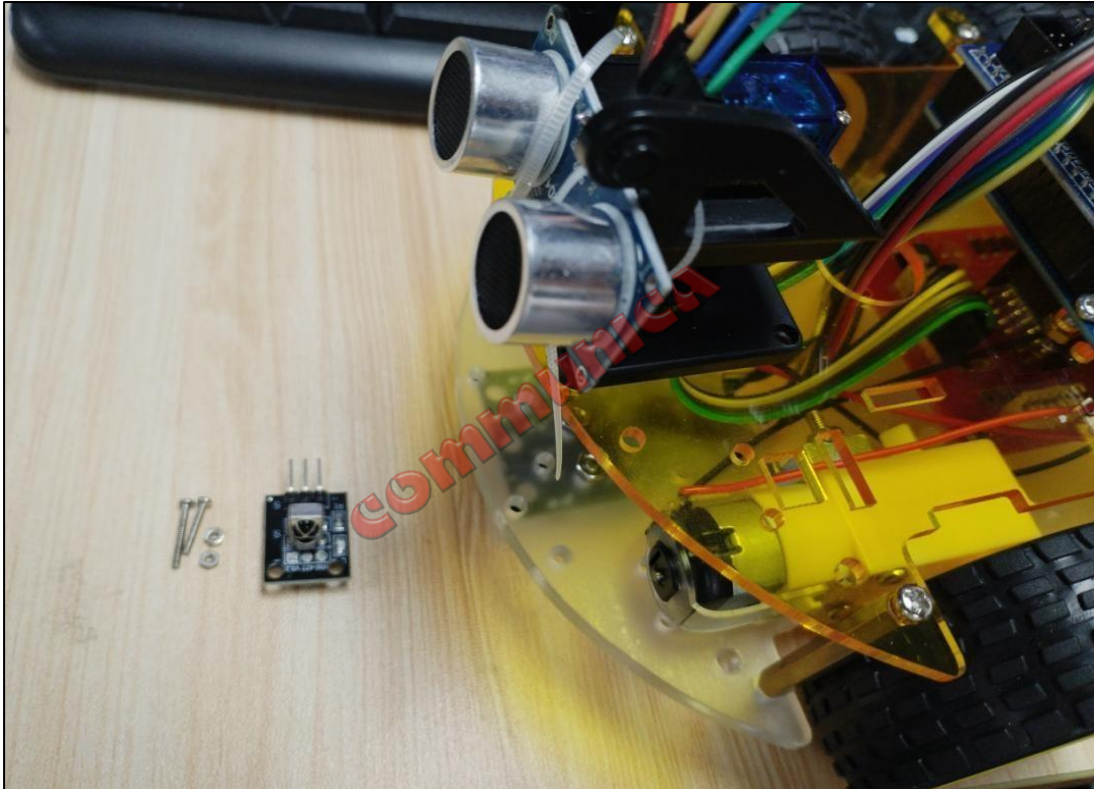
servo	Arduino
G(Brown)	G
V(Red)	V
S(Yellow)	6

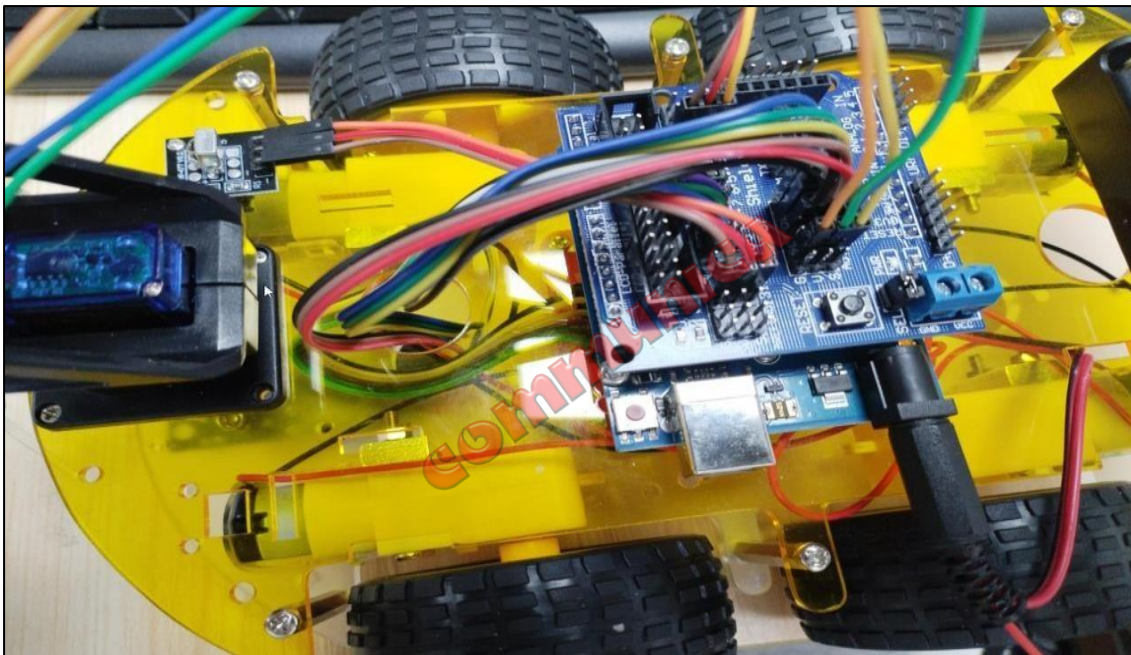
Ultrasonic module	Arduino
Gnd	G
Echo	A1
Trig	A2
Vcc	V

Step 13: Secure the acrylic plate



Step 14: Install the infrared receiver module

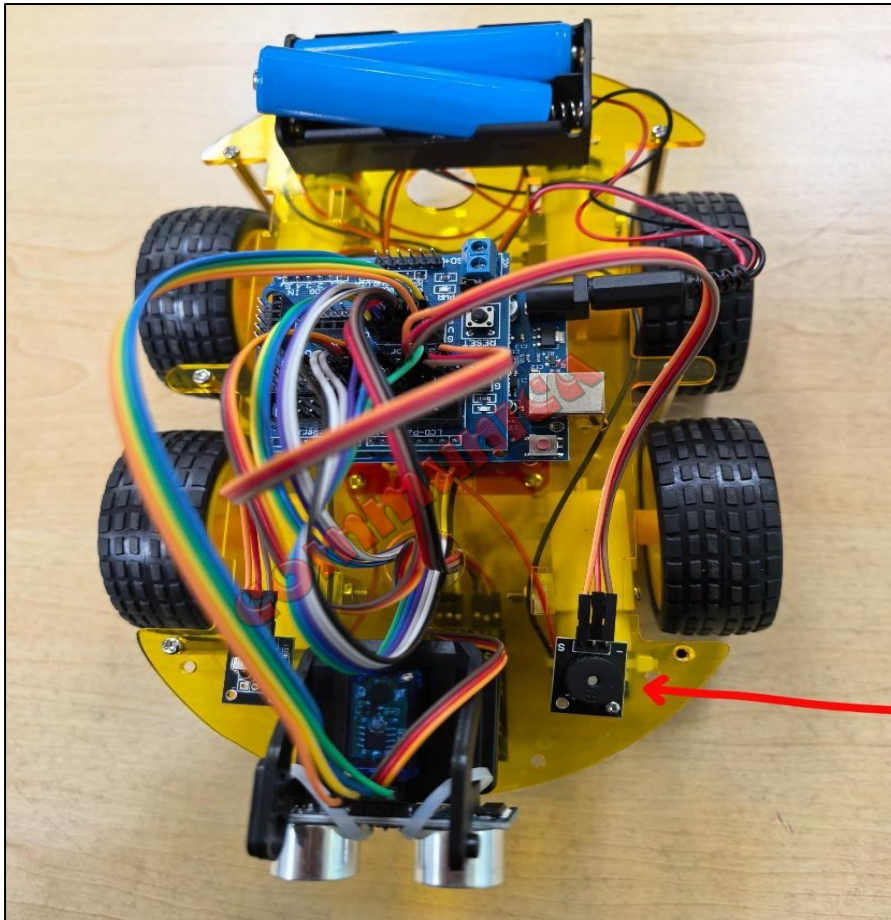




Pin connection table

Infrared receiver module	Arduino
G	G
V	V
S	12

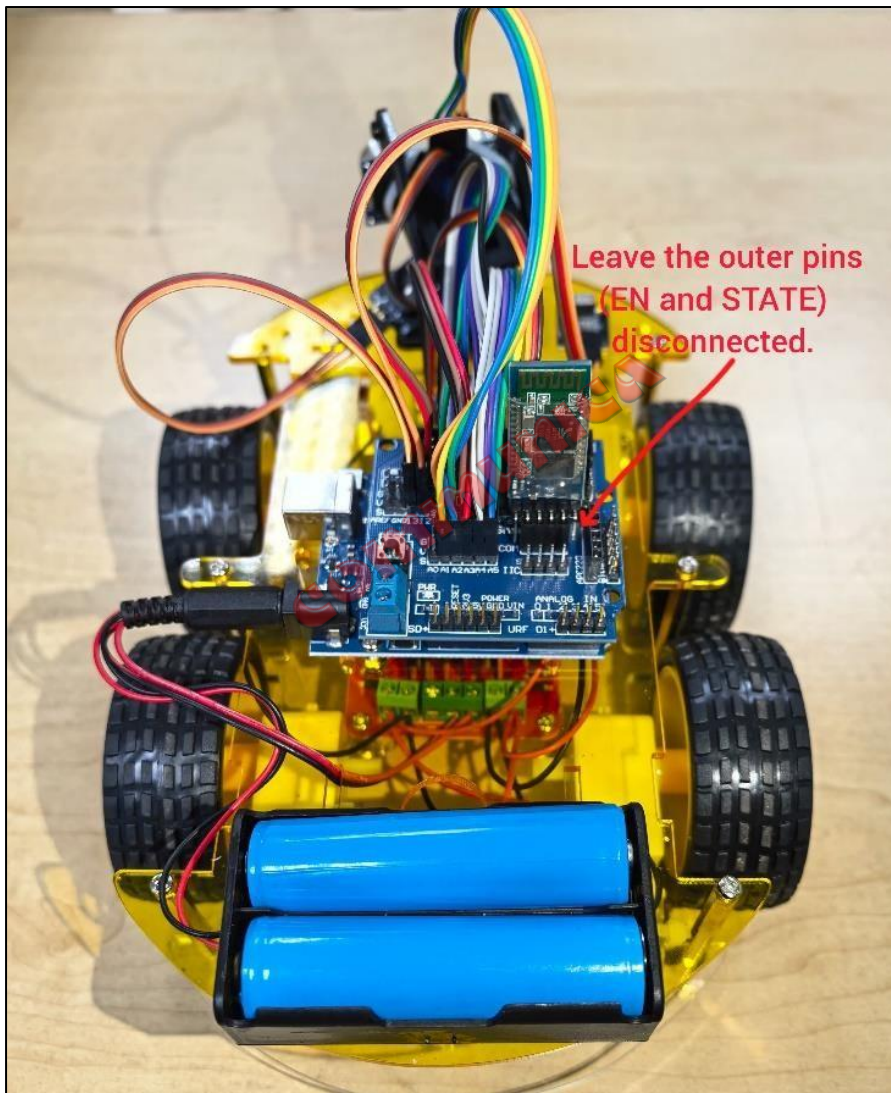
Step 14: Install the P-Buzzer module



Pin connection table

P-Buzzer module	Arduino
G	G
V	V
S	10

Please note that you have to remove the Bluetooth module (HC-05) on the Arduino UNO board when uploading the code.



Tracking module test code

```
void setup() {
  Serial.begin(9600);
  pinMode(A3, INPUT); // Left
  pinMode(A4, INPUT); // Center
  pinMode(A5, INPUT); // Right
}

void loop() {
  int left = digitalRead(A3);
  int center = digitalRead(A4);
  int right = digitalRead(A5);

  Serial.print("Left: ");
  Serial.print(left);
  Serial.print(" | Center: ");
  Serial.print(center);
  Serial.print(" | Right: ");
  Serial.println(right);

  delay(200);
}
```


Ultrasonic sensor test code

```
const int trigPin = A2;
const int echoPin = A1;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  long duration;
  int distance;

  // Send pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read echo
  duration = pulseIn(echoPin, HIGH);

  // Calculate distance in cm
  distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(500);
}
```

Servo test code

```
#include <Servo.h>

Servo myservo;

void setup() {
  myservo.attach(6); // Servo signal connected to pin 6
}

void loop() {
  myservo.write(0); // Rotate to 0°
  delay(1000);

  myservo.write(90); // Rotate to 90°
  delay(1000);

  myservo.write(180); // Rotate to 180°
  delay(1000);
}
```

Bluetooth control code

```
// Motor A pins
const int EN1 = 11;
const int IN1 = 8;
const int IN2 = 7;

// Motor B pins
const int EN2 = 3;
const int IN3 = 5;
const int IN4 = 4;

char command;

void setup() {
  // Set motor pins as outputs
  pinMode(EN1, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(EN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  // Start serial communication with HC-05
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()) {
    command = Serial.read();

    switch (command) {
      case 'F': // Forward
        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, HIGH);
        digitalWrite(IN4, LOW);
        analogWrite(EN1, 255);
        analogWrite(EN2, 255);
        break;

      case 'B': // Backward
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, HIGH);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, HIGH);
        analogWrite(EN1, 255);
        analogWrite(EN2, 255);
        break;

      case 'L': // Smooth Left Turn
        digitalWrite(IN1, HIGH); // Right motor forward
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW); // Left motor off
        digitalWrite(IN4, LOW);
        analogWrite(EN1, 180); // Full speed right motor
        analogWrite(EN2, 0); // Stop left motor
        delay(500);
        analogWrite(EN1, 0);
        break;

      case 'R': // Smooth Right Turn
        digitalWrite(IN1, LOW); // Right motor off
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, HIGH); // Left motor forward
        digitalWrite(IN4, LOW);
        analogWrite(EN1, 0); // Stop right motor
        analogWrite(EN2, 180); // Full speed left motor
        delay(500);
        analogWrite(EN2, 0);
        break;

      case 'S': // Stop
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, LOW);
        analogWrite(EN1, 0);
        analogWrite(EN2, 0);
        break;
    }
  }
}
```

Carkit test Code

```
#include <Servo.h>
#include <NewPing.h>

// ----- ULTRASONIC SENSOR -----
--
#define trig_pin A2
#define echo_pin A1

// ----- MOTOR PINS -----
// Right motors
const int RightMotorForward = 8; // RF
const int RightMotorBackward = 7; // RB
const int enableRight = 11; // enR (changed from 9
to avoid timer conflict)

// Left motors
const int LeftMotorForward = 5; // LF
const int LeftMotorBackward = 4; // LB
const int enableLeft = 3; // enL

// ----- Buzzer PIN -----
const int buzzerPin = 10;

// ----- GLOBAL SETTINGS -----
#define MOTOR_SPEED 130 // Adjust speed (0-255)
#define maximum_distance 200
int i = 0;
Servo myservo;
NewPing sonar(trig_pin, echo_pin, maximum_distance);

// ----- SETUP -----
void setup() {
  Serial.begin(9600);
  pinMode(enableRight, OUTPUT);
  pinMode(RightMotorForward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);
  pinMode(enableLeft, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(trig_pin, OUTPUT);
  pinMode(echo_pin, INPUT);
  pinMode(buzzerPin, OUTPUT);

  myservo.attach(6); // Servo signal pin (Timer1)
  myservo.write(115); // Center position
  delay(500);
  int cm = sonar.ping_cm();
  Serial.println(cm);
// ----- Basic MOTOR CONTROL -----
  moveForward();
  delay(1000);
  moveBackward();
  delay(1000);
  turnLeft();
  delay(1000);
  turnRight();
  delay(1000);
  stopMotors();
  Buzzer();
}

// ----- MAIN LOOP -----

void loop() {
  // Add your code here !!
}

// ----- MOTOR CONTROL Functions -----

void moveForward() {
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  analogWrite(enableRight, MOTOR_SPEED);
  analogWrite(enableLeft, MOTOR_SPEED);
}

void moveBackward() {
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(LeftMotorBackward, HIGH);
  analogWrite(enableRight, MOTOR_SPEED);
  analogWrite(enableLeft, MOTOR_SPEED);
}

void turnLeft() {
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(LeftMotorBackward, HIGH);
  analogWrite(enableRight, MOTOR_SPEED);
  analogWrite(enableLeft, MOTOR_SPEED);
}

void turnRight() {
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  analogWrite(enableRight, MOTOR_SPEED);
  analogWrite(enableLeft, MOTOR_SPEED);
}

void stopMotors() {
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(LeftMotorBackward, LOW);
}

void Buzzer() {
  for(i=0;i<1000;i++){
    digitalWrite(buzzerPin,HIGH);
    delay(1);//wait for 1ms
    digitalWrite(buzzerPin,LOW);
    delay(1);//wait for 1ms
  }
}
```