Project Title

Obstacle avoiding, voice control and Bluetooth control robot car and smart car Parking System using Arduino.

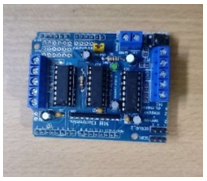
Project overview

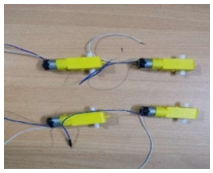
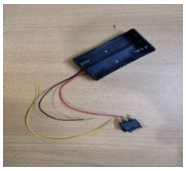
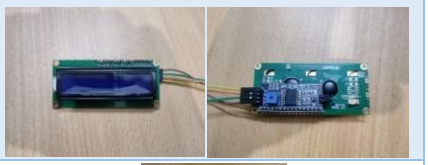
This project involves creating a multifunctional robot car with obstacle avoidance, voice control, Bluetooth control, and a parking system using Arduino. The robot car navigates autonomously, using Ultrasonic sensors to detect and avoid obstacles, with real-time adjustments managed by the Arduino microcontroller. Voice control is implemented via a recognition module, while Bluetooth control allows remote operation through smartphones.

Objective

To design and implement a versatile robotic car system using Arduino, featuring obstacle avoidance, voice control, and Bluetooth connectivity. Additionally, develop a smart parking system for conventional vehicles, integrating Arduino technology to enhance efficiency, user convenience and safely parking.

Components List:

Serial No.	Components	Requirements	Image
1	Arduino Uno	2	
2	Motor driver L293D	1	
3	Ultrasonic Sensor	1	
4	Bluetooth module	1	

5	Servo motor	2	
6	Gear motors	4	
7	Robot wheels	4	
8	Battery Holder	1	
9	Battery	2	
10	Mini breadboard	1	
11	LCD display and I2C module		
12	IR Sensor	2	
13	Glue Gun	1	
			

Procedure

Robot Car:

We can control this robot car using three methods. That is,

1.Obstacle avoidance

In this case, the robot car moves along using the obstacle avoiding. The ultrasonic sensor is mainly used for this purpose.

2.Bluetooth control

In this case, we can control the robot through an app on the smartphone. The Bluetooth module is used for this.

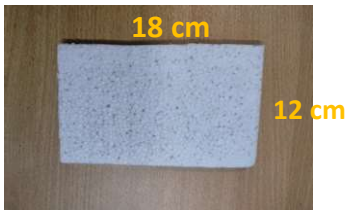
3.Voice control

In this case, we can control this robot using several voice commands. This also requires a Bluetooth module and mobile app.

Now steps to make this project is mentioned below:

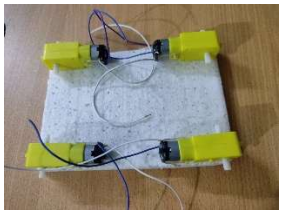
STEP 1:

We need a foam. The size of this foam is height 18cm and width 12cm.



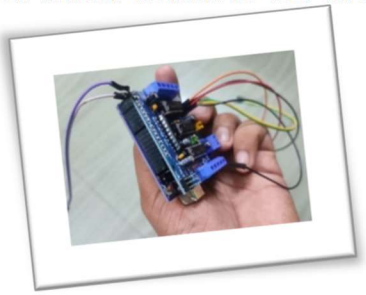
STEP 2:

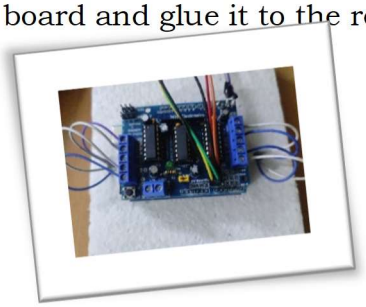
glue the four gear motors to the foam board piece.



STEP 3:

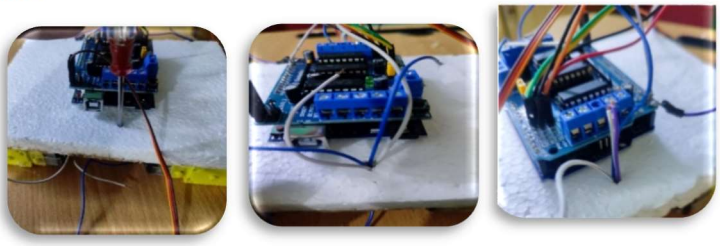
Then, attach the motor shield to the Arduino board and glue it to the robot chassis.





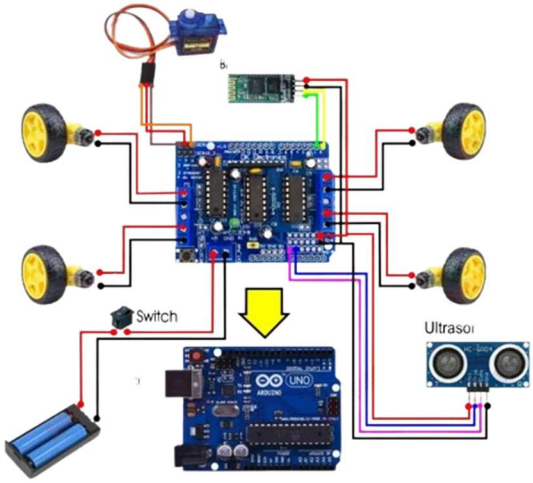
STEP 4:

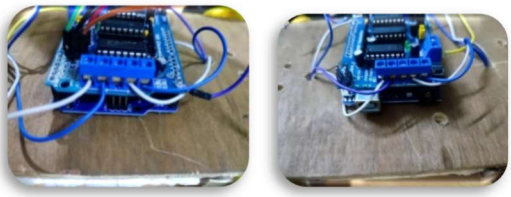
Next, dig two holes on either side of the Arduino board and insert the gear motor wire through these holes.



STEP 5:

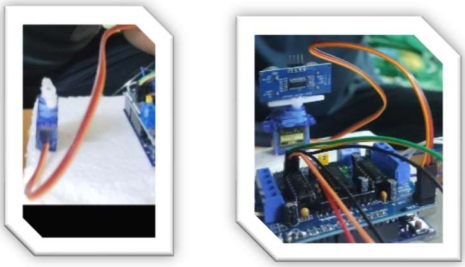
Then, connect the motors to the motor driver shield. To do this, use the circuit diagram below.





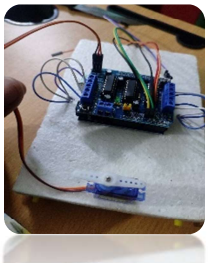
STEP 6:

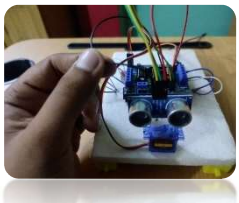
Afterward, attach the servo motor and ultrasonic sensor as follows.



STEP 7:

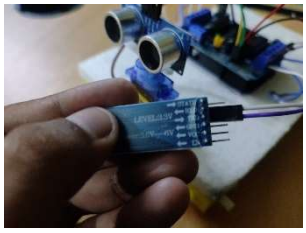
Now, connect the servo motor and the ultrasonic sensor using the circuit diagram above.

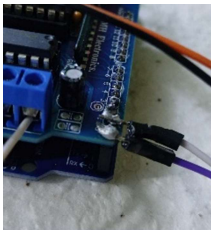




STEP 8:

Then, connect the Bluetooth module to the motor driver shield and glue it to the robot chassis.

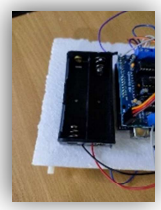


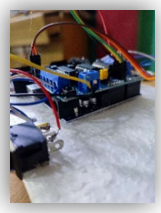


RXD -> D1
TXD -> D0

STEP 9:

After, glue the battery holder and connect it to the driver shield.





STEP 10:

Now, attach the robot wheels and put the batteries to the battery holder.

STEP 11:

So, let's create the program for this project. This program includes all three functions. We can run these separately. It is as follows.

AF motor library — Download

The complete program of this project – Download

Code:

```
/* obstacle avoiding, Bluetooth control, voice control robot car */
```

```
#include <Servo.h>
#include <AFMotor.h>// to control gear motors via motor sheild L293D
#define Echo A0 //Echo connect in A0 off L293D
#define Trig A1 //Trig connect in A0 off L293D
#define motor 10  // for connect 10 number pin in L293D
#define Speed 170 // Set Gear motor speed 170
#define spoint 103
char value;
int distance;

//Variables used for measuring distances and decision-making.
int Left;
int Right;
int L = 0;
int R = 0;
int L1 = 0;
int R1 = 0;
Servo servo;
AF_DCMotor M1(1);
AF_DCMotor M2(2);
AF_DCMotor M3(3);
AF_DCMotor M4(4);
void setup() {
  Serial.begin(9600);
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  servo.attach(motor);
  M1.setSpeed(Speed);
  M2.setSpeed(Speed);
  M3.setSpeed(Speed);
  M4.setSpeed(Speed);
}
void loop() {
  //Obstacle();
  Bluetoothcontrol();
 // voicecontrol();
}
```

```
void Bluetoothcontrol() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
  }
  if (value == 'D'){
    forward();
  } else if (value == 'U') {
    backward();
  } else if (value == 'R') {
    left();
  } else if (value == 'L') {
    right();
  } else if (value == 'S') {
    Stop();
  }
}
void Obstacle() {
  distance = ultrasonic();
  if (distance <= 12) {
    Stop();
    backward();
    delay(100);
    Stop();
    L = leftsee();
    servo.write(spoint);
    delay(800);
    R = rightsee();
    servo.write(spoint);
    if (L < R) {
      right();
      delay(500);
      Stop();
      delay(200);
    } else if (L > R) {
      left();
      delay(500);
```

```
      Stop();
      delay(200);
    }
  } else {
    forward();
  }
}
void voicecontrol() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
    if (value == '-') {
      forward();
    } else if (value == '^') {
      backward();
    } else if (value == '>') {
      R = rightsee();
      servo.write(spoint);
      if (L >= 10 ) {
        left();
        delay(500);
        Stop();
      } else if (L < 10) {
        Stop();
      }
    } else if (value == '<') {
      L = leftsee();
      servo.write(spoint);
      if (R >= 10 ) {
        right();
        delay(500);
        Stop();
      } else if (R < 10) {
        Stop();
      }
    } else if (value == '*') {
      Stop();
```

```
    }
  }
}
// Ultrasonic sensor distance reading function
int ultrasonic() {
  digitalWrite(Trig, LOW);
  delayMicroseconds(4);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  long t = pulseIn(Echo, HIGH);
  long cm = t / 29 / 2; //time convert distance
  return cm;
}
void forward() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void backward() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
}
void right() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void left() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
```

```
}
void Stop() {
  M1.run(RELEASE);
  M2.run(RELEASE);
  M3.run(RELEASE);
  M4.run(RELEASE);
}
int rightsee() {
  servo.write(20);
  delay(800);
  Left = ultrasonic();
  return Left;
}
int leftsee() {
  servo.write(180);
  delay(800);
  Right = ultrasonic();
  return Right;
}
```

Output Picture:



1. Video for Obstacle avoidance

obstacle_avoidance.mp4

2.Video for Bluetooth control:

Bluetooth_Control.mp4 (Command Line)

3.Video for Voice control:

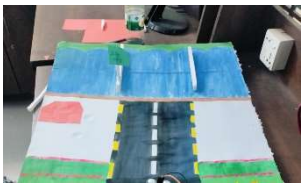
ddf6d250-04af-4dfe-b4ba-bc95c318a810 (1).mp4 (Command Line)

cacd4688-f2a9-4c01-9f61-b22d6f9f6a1b.mp4 (Command Line)

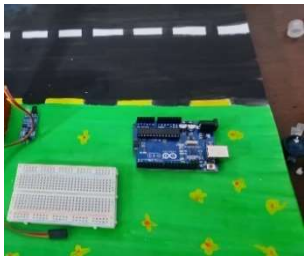
CAR PARKING SYSTEM:

We create car parking system by those following steps-

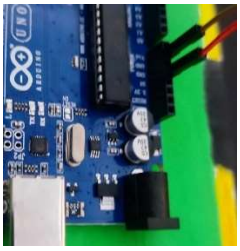
STEP-1: Firstly, we took e foam and draw it for our project.

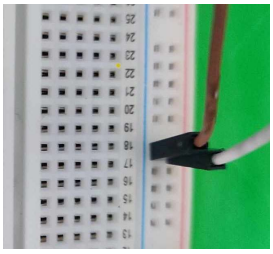


STEP-2: Glue breadboard and Arduino to the foam.



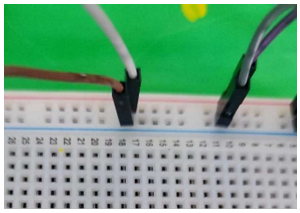
STEP-3: Now, connect breadboard +(VCC) and –(GND) hole with Arduino +5v and Gnd pin which work as a common vcc and gnd.

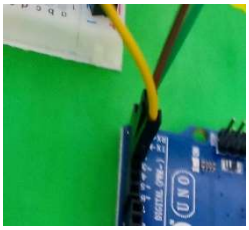




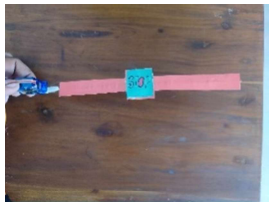
STEP-4: After that, we put 2 IR sensor in the foam which we draw.

STEP-5: Connect IR1 and IR2 Vcc and Gnd pin in breadboard and output pin with Arduino 2 and 3 pin.



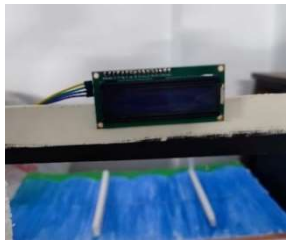


STEP-6: Then we also put Servo motor between 2 IR sensor. And set a stoper in the Servo motor.



STEP-7: Connect servo motor Vcc and Gnd pin in breadboard and output pin connect in 4 number pin of Arduino.

STEP-8: In this step, we glue a ghat and attach LCD Display & I2C in the gate. Where Vcc and Gnd pin connect with breadboard and SDA and SCL connect with Arduino A4 and A5 pin.



Video:

449340363_8038417129556789_9096620847632906051_n.mp4 (Command Line)

Code:

```
#include <Wire.h> // Library use for I2C communiation

#include <LiquidCrystal_I2C.h> // Library for interfacing with LCD Display

LiquidCrystal_I2C lcd(0x27,16,2);   // Address iniatialize for LCD Display

#include <Servo.h>

Servo myservo;

int IR1 = 2;

int IR2 = 3;
```

```
int Slot = 3
int flag1 = 0;
int flag2 = 0;
void setup() {
  Serial.begin(9600); // Iniatialize serial communiation with 9600 buads rate
    lcd.init();
    lcd.backlight();
pinMode(IR1, INPUT);
pinMode(IR2, INPUT);

myservo.attach(4);
myservo.write(140);  // normal position

lcd.setCursor (0,0);
lcd.print("     Team-2 ");
lcd.setCursor (0,1);
lcd.print("  Topic name: ");
delay (5000); // 5000 ms = 5s

lcd.setCursor (0,0);
lcd.print("    SMART CAR ");
lcd.setCursor (0,1);
lcd.print(" PARKING SYSTEM ");

delay (5000);
lcd.setCursor (0,0);
lcd.print("  Team members");
lcd.setCursor (0,1);
lcd.print("Siam,Afroza,Emon ");
delay (5000);

lcd.setCursor (0,0);
lcd.print("  STUDENTS OF ");
lcd.setCursor (0,1);
lcd.print("  R.P.SHAHA UV  ");
```

```
delay (5000);
lcd.setCursor (0,0);
lcd.print(" COURSE TEACHER ");
lcd.setCursor (0,1);
lcd.print(" TANJINA AKTER  ");
delay (5000);
lcd.clear();
}
void loop(){
if(digitalRead (IR1) == LOW && flag1==0){
if(Slot>0){
flag1=1;
if(flag2==0){
myservo.write(0); /// Gate on
Slot = Slot-1;
}
}else{
lcd.setCursor (0,0);
lcd.print("    SORRY :(    ");
lcd.setCursor (0,1);
lcd.print("  Parking Full  ");
delay (3000);
lcd.clear();
}
}
if(digitalRead (IR2) == LOW && flag2==0) {
flag2=1;
if(flag1==0) {
myservo.write(0);
Slot = Slot+1;
   }
}
if(flag1==1 && flag2==1){
delay (1000);
myservo.write(140);  // Normal position
```

```
flag1=0, flag2=0;
}
lcd.setCursor (0,0);
lcd.print("    WELCOME!    ");
lcd.setCursor (0,1);
lcd.print("Slot Left: ");
lcd.print(Slot);
}
```

OUTPUT PICTURE:

OUTPUT VIDEO:

449215083_8496083400420647_8300916734611682077_n.mp4 (Command Line)