Title: Voice Control Car with Obstacle Avoiding.

Objective:

1. To make a driver-free car for the new generation.
2. To operate the car with voice commands.
3. To work with this type of car for a research project on Mars or any other planet.
4. To give the touch of technology in the modern supercars for more efficiency.
5. To use this car as a restaurant servant.

Model Design:

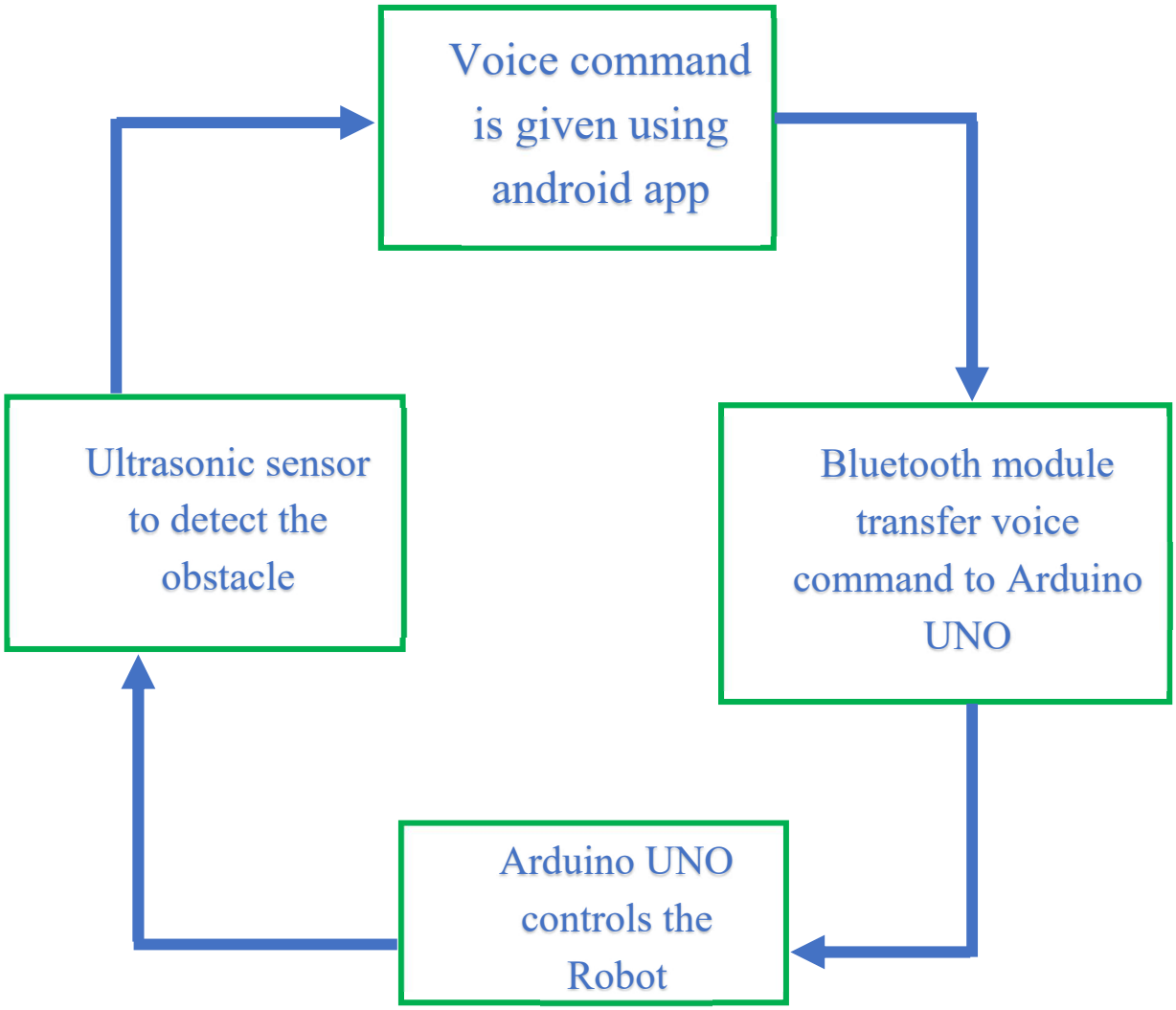


Figure: Block diagram of Robot.

Hardware Requirement:

I. Arduino UNO board x 1
II. L293D motor driver x 1
III. Ultrasonic sensor x 1
IV. Bluetooth module x 1
V. Servo motor x 1
VI. Gear motor x 4
VII. Robot wheel x 4
VIII. Li-ion battery holder x 1
IX. Li-ion battery x 2
X. Jumper wires
XI. Foam board or cardboard

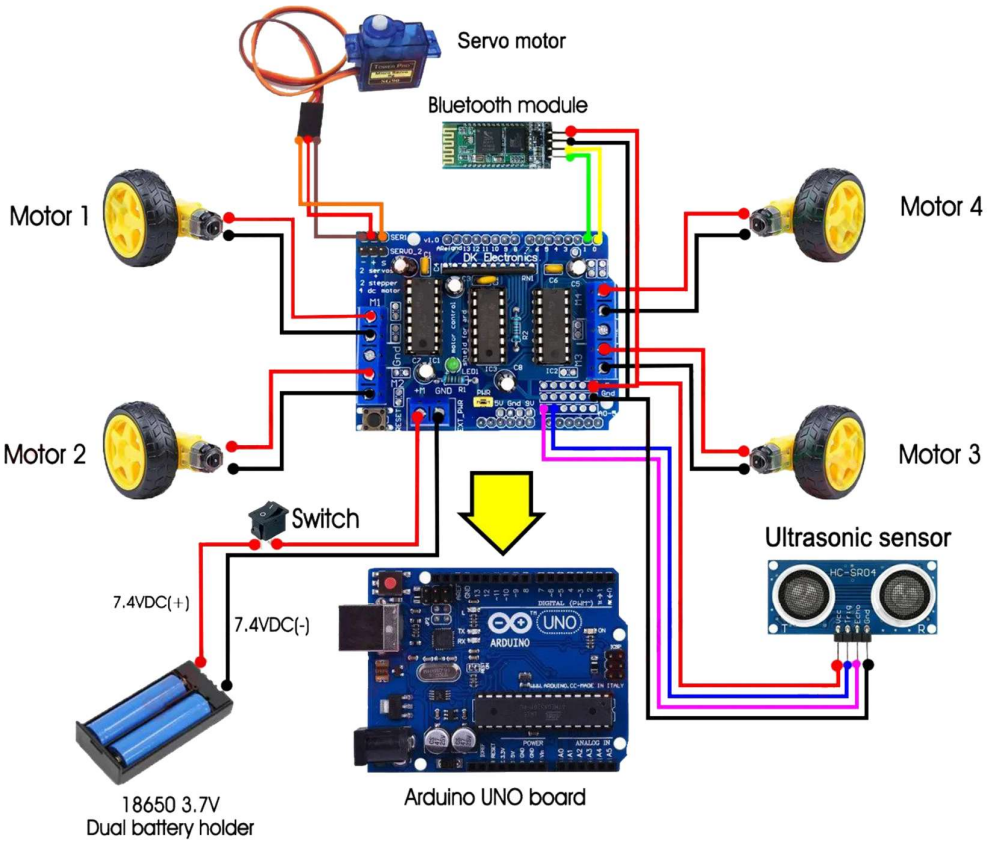
Software Requirement:

1. Arduino IDE 2.0.3
2. Windows
3. MS Office

Budget Proposal:

Arduino UNO board x 1	870tk.
L293D motor driver x 1	255tk.
Ultrasonic sensor x 1	105tk.
Bluetooth module x 1	345tk.
Servo motor x 1	155tk.
Gear motor x 4 = 110 *4	440tk.
Robot wheel x 4 = 70 * 4	280tk.
Li-ion battery holder x 1	50tk.
Li-ion battery x 2	220tk
Jumper wires	50tk
Foam board or cardboard	80tk
Gear motor x 4 = 110 *4	440tk
Total cost	**= 2850tk.**

Circuit Diagram:



The process of this robot:

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

Obstacle avoidance: In this case, the robot car moves along using obstacle avoidance. The ultrasonic sensor is mainly used for this purpose. Study the previous articles for more information.

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

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

Implementation Details :

Step-1: First, we create a structure of a car for our project.

Step-2: Then, we connect all parts of our hardware according to the circuit diagram we are given in our previous report.

Step-3: After, completing the connection of hardware, we write the required code for Voice Control, Obstacle Avoiding, and also Control the car manually.

Step-4: Completing the code part we upload it in Arduino UNO.

Step-5: After that, we install a app name Arduino Bluetooth RC Car to control it manually with the help of Bluetooth drive and apps.

Step-6: For voice control, we use an app name Arduino Bluetooth Control. Primarily we set up 5 voice commands to control the car.

1. Go
2. Back
3. Left
4. Right
5. Stop

Code explanation:

Firstly, libraries are included.

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

Secondly, ultrasonic sensor pins, servo motor pins, motor speed, and servo motor starting point are defined.

```
#define Echo A0
#define Trig A1
#define motor 10
#define Speed 170
#define spoint 103
```

Thirdly, some variables have been created to help the program.

```
char value;
int distance;
int Left;
int Right;
int L = 0;
int R = 0;
int L1 = 0;
int R1 = 0;
```

Then, objects are created for the Servo Library and the AFMotor Library.

```
Servo servo;
AF_DCMotor M1(1);
AF_DCMotor M2(2);
AF_DCMotor M3(3);
AF_DCMotor M4(4);
```

In the setup function, Ultrasonic pins are set to INPUT and OUTPUT. Also, the gear motor speeds have been included.

```
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);
}
```

In the loop function, the three main functions are included. we can run these functions one by one. These are described below.

```
void loop() {
  //Obstacle();
  //Bluetoothcontrol();
  //voicecontrol();
}
```
This function includes the Bluetooth control code. The code lines are described one by one in the code

```
void Bluetoothcontrol() {
//gets the serial communication values and puts them into the char variable.
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
  }
//Next, these values are checked using the IF condition.
//Then, if the char value is 'F', the car moves forward.
  if (value == 'F') {
    forward();
//If the char value is "B", the car moves backward.
  } else if (value == 'B') {
    backward();
//If the char value is "L", the car moves left.
  } else if (value == 'L') {
    left();
//If the char value is "R", the car moves right.
  } else if (value == 'R') {
    right();
//If the char value is "S", the car is stopped.
  } else if (value == 'S') {
    Stop();
  }
}
```

This function includes the obstacle-avoiding code. The code lines are described one by one in the code.

```
void Obstacle() {

//gets the ultrasonic sensor reading and puts it into the variable.
  distance = ultrasonic();

//then, these values are checked using the IF condition.
//If the value is less than or equal to 12,
//the robot is stopped and the servo motor rotate left and right.
// Also, gets both side distance.
  if (distance <= 12) {
    Stop();
    backward();
    delay(100);
    Stop();
    L = leftsee();
    servo.write(spoint);
    delay(800);
    R = rightsee();
    servo.write(spoint);

//After, if the left side distance less than the right side distance. The robot turns right.
    if (L < R) {
      right();
      delay(500);
      Stop();
      delay(200);

//After, if the left side distance more than the right side distance. The robot turns left.
    } else if (L > R) {
      left();
      delay(500);
      Stop();
```

```
    delay(200);
   }

//Otherwise, the robot moves forward.
  } else {
   forward();
  }
}
```

This function includes the voice control code. The code lines are described one by one in the code.

```
void voicecontrol() {

//gets the serial communication values and puts them into the char variable.
  if (Serial.available() > 0) {
   value = Serial.read();
   Serial.println(value);

//If the char value is "^", the car moves forward.
   if (value == '^') {
    forward();

//If the char value is "-", the car moves backward.
   } else if (value == '-') {
    backward();

//If the char value is "<", the car moves left.
   } else if (value == '<') {
    L = leftsee();
    servo.write(spoint);
    if (L >= 10 ) {
     left();
     delay(500);
     Stop();
    } else if (L < 10) {
     Stop();
    }
```

```
//If the char value is ">", the car moves right.
    } else if (value == '>') {
      R = rightsee();
      servo.write(spoint);
      if (R >= 10 ) {
        right();
        delay(500);
        Stop();
      } else if (R < 10) {
        Stop();
      }

//If the char value is "*", the car is stopped.
    } else if (value == '*') {
      Stop();
    }
  }
}
```

Obstacle avoidance program

OK, now connect this robot car to the computer. Then, remove the two forward slash in front of the “obstacle” function. Next, remove the RX and TX jumper wires connected to the Bluetooth module.

Now, select board and port. After, upload this code to the robot and reconnect the RX and TX jumper wires.

Conclusion :

We have collected all the equipment. After that, we connect all parts and we do our required soldering in our circuit according to our designed Circuit diagram. After that, we put the circuit in our structure. Then we write and upload the code in Arduino UNO. After some tests, we solve some problems. We run our car and successfully create our project. Now we are almost done with our project few of the work is left, we will do it before the final Demonstration.