

```

// Includes
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(0, 1);

// Variables
// UltraSonic Distance Detector
const int triggerPin = 9;
const int echoPin = 10;
long duration;
int distance;
// Motor
const int fmotor1 = 8;
const int fmotor2 = 3;
const int smotor3 = 2;
const int smotor4 = 11;
// MotorController Pin
const int fenablePin = 12;
const int senablePin = 13;
// Commands
bool forwardMotor = false;
bool backMotor = false;
bool turnR = false;
bool turnL = false;
bool Stop = false;
bool Lights = false;
bool BlinkLights = false;
//Led
const int ledR = 4;
const int ledL = 6;
bool isLed = true;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  bluetooth.begin(9600);

  // UltraSonic Distance
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Motor
  pinMode(fmotor2, OUTPUT);
  pinMode(fmotor1, OUTPUT);
  pinMode(smotor3, OUTPUT);
  pinMode(smotor4, OUTPUT);
  pinMode(fenablePin, OUTPUT);
  pinMode(senablePin, OUTPUT);
  // Assigning Speed to the Motors.
  analogWrite(fenablePin, 255);
  analogWrite(senablePin, 255);

  //Led
  pinMode(ledR, OUTPUT);
  pinMode(ledL, OUTPUT);
}

void loop() {
  // Measuring the distance using Ultra Sonic Distance Sense
  MeasureDistance();

  // Getting Command From Bluetooth
  if (bluetooth.available()) {
    // Getting Command from bluetooth Serial Device

```

```

String command = bluetooth.readStringUntil('\n');

// Print the Command we get
for (int i = 0; i < command.length(); i++) {
  command[i] = tolower(command[i]);
}

// Removing Extra Spaces from "Command"
while (command.startsWith(" ")) {
  command = command.substring(1);
}
while (command.endsWith(" ")) {
  command = command.substring(0, command.length() - 1);
}

// Print the command
Serial.println("Command: " + command);

// Main bool Logic
if (command == "move forward") {
  ForwardMotor();
}
if ((command == "move back") || (command == "move backward")) {
  BackMove();
}
if (command == "turn right") {
  RightTurn();
}
if (command == "turn left") {
  LeftTurn();
}
if ((command == "light up the car") || (command == "turn on the lights")) {
  LightUpTheCar();
}
if ((command == "blink the lights") || (command == "make the lights blinking")) {
  BlinkTheLight();
}
if (command == "stop") {
  StopMotor();
}
}

// Motor Controlling
// BoolLogic();
if (forwardMotor == true) {
  if (distance > 20) {
    MoveForward();
  } else {
    StopCar();
  }
}
if (turnR == true) {
  TurnRight();
}
if (turnL == true) {
  TurnLeft();
}
if (Stop == true) {
  StopCar();
}
if (Lights == true) {
  Light();
}
}

```

```

if (BlinkLights == true) {
    BlinkLight();
}
if (backMotor == true) {
    MoveBack();
}
}

void MeasureDistance() {
    // UltraSonic Distance
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    duration = pulseIn(echoPin, HIGH, 30000);
    distance = duration * 0.034 / 2;
    if (distance == 0) {
        distance = 50;
    }

    // Serial.print("Distance: ");
    // Serial.println(distance);
    delay(100);
}

// Motor Logic
void MoveForward() {
    AntiClockwiseM1();
    AntiClockwiseM2();
    Serial.println("Motor is Running Forward.....");
}
void MoveBack() {
    ClockWiseM1();
    ClockWiseM2();
}
void ClockWiseM1() {
    digitalWrite(fmotor2, LOW);
    digitalWrite(fmotor1, HIGH);
}
void ClockWiseM2() {
    digitalWrite(smotor3, LOW);
    digitalWrite(smotor4, HIGH);
}
void AntiClockwiseM1() {
    digitalWrite(fmotor2, HIGH);
    digitalWrite(fmotor1, LOW);
}
void AntiClockwiseM2() {
    digitalWrite(smotor3, HIGH);
    digitalWrite(smotor4, LOW);
}
void TurnRight() {
    ClockWiseM1();
    AntiClockwiseM2();
    digitalWrite(ledL, LOW);
    digitalWrite(ledR, HIGH);
    delay(1000);
    digitalWrite(ledR, LOW);
    delay(800);
}
void TurnLeft() {
    AntiClockwiseM1();

```

```

ClockWiseM2();
digitalWrite(ledR, LOW);
digitalWrite(ledL, HIGH);
delay(1000);
digitalWrite(ledL, LOW);
delay(800);
}

void StopCar() {
    digitalWrite(fmotor2, LOW);
    digitalWrite(fmotor1, LOW);
    digitalWrite(smotor3, LOW);
    digitalWrite(smotor4, LOW);
}

void Light() {
    digitalWrite(ledR, HIGH);
    digitalWrite(ledL, HIGH);
}

void BlinkLight() {
    digitalWrite(ledR, HIGH);
    digitalWrite(ledL, HIGH);
    delay(1000);
    digitalWrite(ledR, LOW);
    digitalWrite(ledL, LOW);
    delay(800);
}

// bool Logic
void ForwardMotor() {
    forwardMotor = true; turnL = false; turnR = false; Stop = false; backMotor = false;
}
void LeftTurn() {
    forwardMotor = false; turnL = true; turnR = false; Stop = false; backMotor = false;
}
void RightTurn() {
    forwardMotor = false; turnL = false; turnR = true; Stop = false; backMotor = false;
}
void StopMotor() {
    forwardMotor = false; turnL = false; turnR = false; Stop = true; backMotor = false; BlinkLights = false; Lights = false;
}
void BlinkTheLight() {
    BlinkLights = true; Lights = false;
}
void LightUpTheCar() {
    BlinkLights = false; Lights = true;
}
void BackMove() {
    forwardMotor = false; turnL = false; turnR = false; Stop = false; backMotor = true;
}

```