SOURCE CODE

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
// SoftwareSerial Bluetooth (RX = 2, TX = 3)
SoftwareSerial BT(2, 3);
const int trigPin = 7;
const int echoPin = 6;
long duration;
int d;
// Motor pins for first drive (A0, A1, A2, A3)
int motor1A = A0;
int motor1B = A1;
int motor1C = A2;
int motor1D = A3;
#define buttonPin 5   // SHIFTED (because 2 is used for RX)
#define Buzzer 4
bool obstacleDetected = false;
void setup() {
  BT.begin(9600);         // Bluetooth
  lcd.begin(16, 2);
  pinMode(motor1A, OUTPUT);
```

```
  pinMode(motor1B, OUTPUT);
  pinMode(motor1C, OUTPUT);
  pinMode(motor1D, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buttonPin, INPUT);
  pinMode(Buzzer, OUTPUT);
  stopMotors();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Obstacle Detect");
  lcd.setCursor(0, 1);
  lcd.print("Bluetooth Robot.");
  delay(3000);
  lcd.clear();
}
void loop() {
  // Ultrasonic
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
duration = pulseIn(echoPin, HIGH);
d = duration * 0.034 / 2;
lcd.setCursor(0, 0);
lcd.print("D:");
lcd.print(d);
lcd.print("cm");
lcd.print("  ");
// Obstacle detection
if (d <= 10 && d > 0) {
  obstacleDetected = true;
  stopMotors();
  digitalWrite(Buzzer, HIGH);
  lcd.setCursor(9,0);
  lcd.print("ObsStop");
} else {
  obstacleDetected = false;
  digitalWrite(Buzzer, LOW);
  lcd.setCursor(9,0);
  lcd.print("PathClr ");
}
// Bluetooth command
if (BT.available()) {
  char command = BT.read();
```

```
if (obstacleDetected && command == 'F') {
  lcd.setCursor(0, 1);
  lcd.print("Forward Blocked!");
  stopMotors();
  return; }
if (command == 'F') {
  forward();
  lcd.setCursor(0, 1);
  lcd.print("    Forward     ");
}
else if (command == 'B') {
  backward();
  lcd.setCursor(0,1);
  lcd.print("    Backward    ");
}
else if (command == 'S') {
  stopMotors();
  lcd.setCursor(0,1);
  lcd.print("     Stopped    ");
}
else if (command == 'R') {
  lcd.setCursor(0,1);
  lcd.print("   TurnRight    ");
```

```
      turnRight();
      delay(1000);
      stopMotors();
    }
    else if (command == 'L') {
      lcd.setCursor(0,1);
      lcd.print("    TurnLeft    ");
      turnLeft();
      delay(1000);
      stopMotors();}
  }
  delay(100);
}
// Motor Functions
void forward() {
  digitalWrite(motor1A, HIGH);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor1C, HIGH);
  digitalWrite(motor1D, LOW);
}
void backward() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, HIGH);
```

```
  digitalWrite(motor1C, LOW);
  digitalWrite(motor1D, HIGH);
}
void stopMotors() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor1C, LOW);
  digitalWrite(motor1D, LOW);
}
void turnRight() {
  digitalWrite(motor1A, HIGH);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor1C, LOW);
  digitalWrite(motor1D, HIGH);
}
void turnLeft() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, HIGH);
  digitalWrite(motor1C, HIGH);
  digitalWrite(motor1D, LOW);
}
```