**SOURCE CODE**

// Motor Driver Connections

const int motorPin1 = 2; // Motor 1: Input 1

const int motorPin2 = 3; // Motor 1: Input 2

const int motorPin3 = 4; // Motor 2: Input 1

const int motorPin4 = 5; // Motor 2: Input 2

// Ultrasonic Sensor Connections

const int trigPin = 8;

const int echoPin = 9;

// IR Sensor Connection

const int irSensorPin = A0;

// UV Light Control Pin

const int uvLightPin = 11;

// Variables to store sensor readings

long duration;

int distance;

int irValue;

void setup() {

  // Motor driver pins as outputs

  pinMode(motorPin1, OUTPUT);

  pinMode(motorPin2, OUTPUT);

  pinMode(motorPin3, OUTPUT);

  pinMode(motorPin4, OUTPUT);

  // Ultrasonic sensor pins

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  // IR sensor pin

  pinMode(irSensorPin, INPUT);

  // UV light control pin

  pinMode(uvLightPin, OUTPUT);

  // Initialize Serial Monitor

  Serial.begin(9600);

}

void loop() {

  // Measure distance using the ultrasonic sensor

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distance = duration \* 0.0343 / 2; // Calculate distance in centimeters

  // Read IR sensor value

  irValue = analogRead(irSensorPin);

  // Print sensor readings to Serial Monitor

  Serial.print(&quot;Distance: &quot;);

  Serial.print(distance);

  Serial.println(&quot; cm&quot;);

  Serial.print(&quot;IR Value: &quot;);

  Serial.println(irValue);

  // Control the motors based on sensor readings

  if (distance &lt; 25 || irValue &lt; 700) {

    // Stop motors and turn off UV light if obstacle is detected

    stopMotors();

    digitalWrite(uvLightPin, LOW); // Turn off UV light

  } else {

    // Move motors forward and turn on UV light if path is clear

    moveForward();

    digitalWrite(uvLightPin, HIGH); // Turn on UV light

  }

  // Add a delay before the next iteration

  delay(100);

}

void moveForward() {

  digitalWrite(motorPin1, HIGH);

  digitalWrite(motorPin2, LOW);

  digitalWrite(motorPin3, HIGH);

  digitalWrite(motorPin4, LOW);

}

void stopMotors() {

  digitalWrite(motorPin1, LOW);

  digitalWrite(motorPin2, LOW);

  digitalWrite(motorPin3, LOW);

  digitalWrite(motorPin4, LOW);

}