```
#include <Servo.h>
// IR sensor pins
const int IR_LEFT = A3;
const int IR_MIDDLE = A4;
const int IR_RIGHT = A5;
// Ultrasonic sensor pins
const int TRIGGER_PIN = 10;
const int ECHO_PIN = 9;
// Servo motor pin
const int SERVO_PIN = 11;
// Motor pins
const int MOTOR1_PIN1 = 4;
const int MOTOR1_PIN2 = 5;
const int MOTOR2_PIN1 = 6;
const int MOTOR2_PIN2 = 7;
int count=3; // classroom count

// Define the IR toggle sensor pin
const int IR_SENSOR_PIN = 2;
// Define the toggle state variable
bool toggleState = true;
// Variable to store the previous state of the IR sensor
bool previousState = true;

// Ultrasonic sensor variables
long duration;
int distance;
// Servo motor variables
Servo servo;
void setup() {
  // Initialize servo motor
servo.attach(SERVO_PIN);
servo.write(98);
// Initialize IR sensors
pinMode(IR_LEFT, INPUT);
pinMode(IR_MIDDLE, INPUT);
pinMode(IR_RIGHT, INPUT);

pinMode(IR_SENSOR_PIN, INPUT);
// Initialize ultrasonic sensor
pinMode(TRIGGER_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
// Initialize motor pins
pinMode(MOTOR1_PIN1, OUTPUT);
pinMode(MOTOR1_PIN2, OUTPUT);
pinMode(MOTOR2_PIN1, OUTPUT);
pinMode(MOTOR2_PIN2, OUTPUT);
Serial.begin(9600);
}

void obstacle_avoid(){
//move right
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, LOW);
digitalWrite(MOTOR2_PIN2, HIGH);
delay(2000);
```

```
//go straight
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
delay(2000);
//go left
digitalWrite(MOTOR1_PIN1, LOW);
digitalWrite(MOTOR1_PIN2, HIGH);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
delay(2000);
//go straight
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
delay(3000);
//go left
digitalWrite(MOTOR1_PIN1, LOW);
digitalWrite(MOTOR1_PIN2, HIGH);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
delay(2000);
//go straight
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
delay(1600);
//go right
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, LOW);
digitalWrite(MOTOR2_PIN2, HIGH);
delay(1500);

}

void line_following(){

  int irLeft=digitalRead(IR_LEFT);
  int irMiddle=digitalRead(IR_MIDDLE);
  int irRight=digitalRead(IR_RIGHT);

if ((irMiddle == HIGH && irLeft == LOW && irRight==LOW) || (irLeft==LOW && irMiddle==LOW
&& irRight==LOW)) {
// Go straight
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
delay(30);
} else if ((irLeft == HIGH && irMiddle==LOW && irRight==LOW) || (irMiddle==HIGH && irLeft ==
HIGH && irRight==LOW)) {
// Turn left
digitalWrite(MOTOR1_PIN1, LOW);
```

```
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
delay(50);
} else if ((irRight == HIGH && irMiddle==LOW && irLeft==LOW) || (irMiddle==HIGH && irLeft ==
LOW && irRight==HIGH)) {
// Turn right
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, LOW);
digitalWrite(MOTOR2_PIN2, LOW);
delay(50);
}
// Stop
else   {
digitalWrite(MOTOR1_PIN1, LOW);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, LOW);
digitalWrite(MOTOR2_PIN2, LOW);
 }
}

void loop() {
digitalWrite(TRIGGER_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIGGER_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH);
distance = duration / 58.2;
if(distance<15){        // check for obstacle
   //go reverse
   {digitalWrite(MOTOR1_PIN1, LOW);
   digitalWrite(MOTOR1_PIN2, HIGH);
   digitalWrite(MOTOR2_PIN1, LOW);
   digitalWrite(MOTOR2_PIN2, HIGH);
   delay(1000);
   //stop
   digitalWrite(MOTOR1_PIN1, LOW);
   digitalWrite(MOTOR1_PIN2, LOW);
   digitalWrite(MOTOR2_PIN1, LOW);
   digitalWrite(MOTOR2_PIN2, LOW);}
   servo.write(25);
   delay(1000);
   digitalWrite(TRIGGER_PIN, LOW);
   delayMicroseconds(2);
   digitalWrite(TRIGGER_PIN, HIGH);
   delayMicroseconds(10);
   digitalWrite(TRIGGER_PIN, LOW);
   duration = pulseIn(ECHO_PIN, HIGH);
   distance = duration / 58.2;
   if(distance<30){          // check for obstacle
      digitalWrite(MOTOR1_PIN1, LOW);
      digitalWrite(MOTOR1_PIN2, LOW);
      digitalWrite(MOTOR2_PIN1, LOW);
      digitalWrite(MOTOR2_PIN2, LOW);
   }
```

```
else{
      servo.write(98);
      delay(1000);
      obstacle_avoid();
   }
}
else{
   line_following();     // Follow line
   file_taken();          // Check if file is taken
}
}

void file_taken() {
   int irLeft=digitalRead(IR_LEFT);
  int irMiddle=digitalRead(IR_MIDDLE);
  int irRight=digitalRead(IR_RIGHT);
   // Read the current state of the IR sensor
bool currentState = digitalRead(IR_SENSOR_PIN);
// Check for a transition from HIGH to LOW
if (currentState == LOW && previousState == HIGH && irMiddle==HIGH && irLeft == HIGH &&
irRight==HIGH && (count>=0)) {
// Toggle the state
toggleState = !toggleState;
delay(10000);       // Wait 10 seconds after file is taken
digitalWrite(MOTOR1_PIN1, HIGH);
digitalWrite(MOTOR1_PIN2, LOW);
digitalWrite(MOTOR2_PIN1, HIGH);
digitalWrite(MOTOR2_PIN2, LOW);
   delay(400);
   count--;
   }
// Store the current state as the previous state for the next loop iteration
previousState = currentState;
}
```