# SERVO MOTOR BRAKING SYSTEM WITH MULTI IR SENSOR

## 1. Project Description

### Objective

This project uses five IR sensors, a single-channel relay, and a servo motor to control the operation of a DC motor. The motor halts when an obstacle is detected by any IR sensor, the servo motor applies a brake by moving to 90°, and resumes operation when all sensors are clear.

### System Overview

• Obstacle Detection: Five IR sensors continuously monitor for obstacles. If any sensor detects an obstacle, the system halts the motor and engages the brake.  
• Motor Control: A 5V single-channel relay switches the motor on and off based on the sensor inputs.  
• Braking Mechanism: The servo motor is responsible for braking by moving to a specified angle (90°) when an obstacle is detected. After a delay, it returns to its initial position (0°) to release the brake.

### Applications

• Autonomous vehicles.  
• Conveyor belt systems with obstacle detection.  
• Automated machinery requiring controlled stopping mechanisms.

## 2. Components and Pin Connections

### Components Required

|  |  |
| --- | --- |
| Component | Quantity |
| Arduino Uno | 1 |
| IR Sensors | 5 |
| Single-Channel Relay | 1 |
| Servo Motor | 1 |
| DC Motor | 1 |
| External Power Supply | 1 |
| Connecting Wires | As needed |
| Breadboard | 1 |

### Pin Connections

|  |  |
| --- | --- |
| Component | Arduino Pin |
| IR Sensor 1 | Pin 2 |
| IR Sensor 2 | Pin 3 |
| IR Sensor 3 | Pin 4 |
| IR Sensor 4 | Pin 5 |
| IR Sensor 5 | Pin 6 |
| Relay Control Pin | Pin 7 |
| Servo Control Pin | Pin 8 |

Relay Connection:  
• Input: Connected to Arduino Pin 7.  
• VCC: Connected to 5V on Arduino.  
• GND: Connected to GND on Arduino.  
• Output: Connected in series with the motor and power supply.

Servo Motor Connection:  
• Signal: Connected to Arduino Pin 8.  
• VCC: Connected to 5V on Arduino.  
• GND: Connected to GND on Arduino.

## 3. Code

#include <Servo.h>  
// Define pins for IR sensors, relay, and servo  
#define IR\_SENSOR1\_PIN 2  
#define IR\_SENSOR2\_PIN 3  
#define IR\_SENSOR3\_PIN 4  
#define IR\_SENSOR4\_PIN 5  
#define IR\_SENSOR5\_PIN 6  
#define RELAY\_PIN 7 // Pin for controlling the relay  
#define SERVO\_PIN 8 // Pin for controlling the servo  
Servo brakeServo; // Servo object for braking  
void setup() {  
 // IR sensor pins  
 pinMode(IR\_SENSOR1\_PIN, INPUT);  
 pinMode(IR\_SENSOR2\_PIN, INPUT);  
 pinMode(IR\_SENSOR3\_PIN, INPUT);  
 pinMode(IR\_SENSOR4\_PIN, INPUT);  
 pinMode(IR\_SENSOR5\_PIN, INPUT);  
 // Relay pin  
 pinMode(RELAY\_PIN, OUTPUT);  
 // Attach servo to pin  
 brakeServo.attach(SERVO\_PIN);  
 // Initialize servo and relay  
 brakeServo.write(0); // Servo in "no brake" position (0 degrees)  
 digitalWrite(RELAY\_PIN, LOW); // Relay off initially (motor is off)  
}  
void loop() {  
 // Read IR sensor values  
 int irValue1 = digitalRead(IR\_SENSOR1\_PIN);  
 int irValue2 = digitalRead(IR\_SENSOR2\_PIN);  
 int irValue3 = digitalRead(IR\_SENSOR3\_PIN);  
 int irValue4 = digitalRead(IR\_SENSOR4\_PIN);  
 int irValue5 = digitalRead(IR\_SENSOR5\_PIN);  
 // Check if any IR sensor detects an obstacle  
 if (irValue1 == LOW || irValue2 == LOW || irValue3 == LOW || irValue4 == LOW || irValue5 == LOW) {  
 // Stop the motor (turn off relay)  
 digitalWrite(RELAY\_PIN, HIGH); // Relay off (motor stops)  
 // Apply brake using servo at 90 degrees  
 brakeServo.write(90); // Move servo to 90 degrees (brake position)  
 delay(2000); // Wait for 2 seconds at 90 degrees  
 // Return servo to 0 degrees (initial position)  
 brakeServo.write(0); // Move servo back to 0 degrees (brake released)  
 delay(1000); // Wait for 1 second at position 0  
 } else { // No obstacle detected by any sensor  
 // Release brake  
 brakeServo.write(0); // Move servo to "no brake" position  
 delay(500); // Wait for servo to move  
 // Turn on the relay to power the motor (motor moves forward)  
 digitalWrite(RELAY\_PIN, LOW); // Relay on (motor starts)  
 }  
}

#include <Servo.h>  
// Define pins for IR sensors, relay, and servo  
#define IR\_SENSOR1\_PIN 2  
#define IR\_SENSOR2\_PIN 3  
#define IR\_SENSOR3\_PIN 4  
#define IR\_SENSOR4\_PIN 5  
#define IR\_SENSOR5\_PIN 6  
#define RELAY\_PIN 7 // Pin for controlling the relay  
#define SERVO\_PIN 8 // Pin for controlling the servo  
Servo brakeServo; // Servo object for braking  
void setup() {  
 // IR sensor pins  
 pinMode(IR\_SENSOR1\_PIN, INPUT);  
 pinMode(IR\_SENSOR2\_PIN, INPUT);  
 pinMode(IR\_SENSOR3\_PIN, INPUT);  
 pinMode(IR\_SENSOR4\_PIN, INPUT);  
 pinMode(IR\_SENSOR5\_PIN, INPUT);  
 // Relay pin  
 pinMode(RELAY\_PIN, OUTPUT);  
 // Attach servo to pin  
 brakeServo.attach(SERVO\_PIN);  
 // Initialize servo and relay  
 brakeServo.write(0); // Servo in "no brake" position (0 degrees)  
 digitalWrite(RELAY\_PIN, LOW); // Relay off initially (motor is off)  
}  
void loop() {  
 // Read IR sensor values  
 int irValue1 = digitalRead(IR\_SENSOR1\_PIN);  
 int irValue2 = digitalRead(IR\_SENSOR2\_PIN);  
 int irValue3 = digitalRead(IR\_SENSOR3\_PIN);  
 int irValue4 = digitalRead(IR\_SENSOR4\_PIN);  
 int irValue5 = digitalRead(IR\_SENSOR5\_PIN);  
 // Check if any IR sensor detects an obstacle  
 if (irValue1 == LOW || irValue2 == LOW || irValue3 == LOW || irValue4 == LOW || irValue5 == LOW) {  
 // Stop the motor (turn off relay)  
 digitalWrite(RELAY\_PIN, HIGH); // Relay off (motor stops)  
 // Apply brake using servo at 90 degrees  
 brakeServo.write(90); // Move servo to 90 degrees (brake position)  
 delay(2000); // Wait for 2 seconds at 90 degrees  
 // Return servo to 0 degrees (initial position)  
 brakeServo.write(0); // Move servo back to 0 degrees (brake released)  
 delay(1000); // Wait for 1 second at position 0  
 } else { // No obstacle detected by any sensor  
 // Release brake  
 brakeServo.write(0); // Move servo to "no brake" position  
 delay(500); // Wait for servo to move  
 // Turn on the relay to power the motor (motor moves forward)  
 digitalWrite(RELAY\_PIN, LOW); // Relay on (motor starts)  
 }  
}