

# Automatic Door Opening and Closing System

**1. Introduction:** The automatic door opening and closing system using IR sensor and servo motor is designed to automate the operation of doors by detecting the presence of objects or individuals using Infrared sensors. This system provides convenience, security, and efficiency by automatically opening and closing the door without manual intervention.

## 2. Components:

- Arduino Uno
- IR sensor module
- Servo motor
- Door/door mechanism
- Power supply
- Connecting wires

**Arduino Uno:** It is a microcontroller board based on the ATmega328P chip. The Uno acts as the brain of the system, receiving input from the IR sensor and controlling the servo motor to operate the door.

**IR Sensor Module:** The Infrared (IR) sensor module consists of an IR transmitter and receiver pair. When an object comes within its sensing range, it detects the reflected IR radiation and sends a signal to the Arduino, indicating the presence of an object.

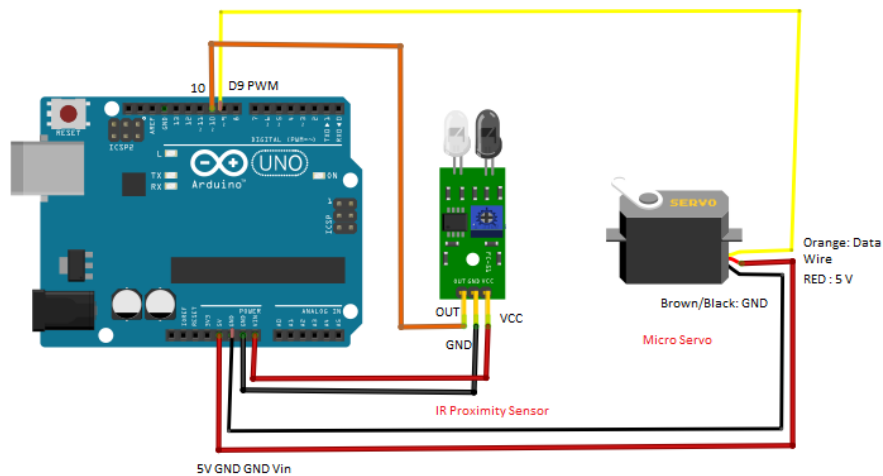
**Servo Motor:** A servo motor is a rotary actuator that allows precise control of angular position. It consists of a motor coupled with a feedback system and is capable of rotating to a specific angle. In this project, the servo motor is used to physically open and close the door by adjusting its angle.

**Door Mechanism:** This includes the physical components required to construct the door system, such as doors, wheels, tracks, and supporting structures. The servo motor is connected to this mechanism to facilitate the movement of the door for opening and closing.

**Power Supply:** It provides the necessary electrical power to the Arduino, servo motor, and other components in the circuit. Depending on the voltage and current requirements of the components, a suitable power supply is selected to ensure proper functionality.

**Connecting Wires:** These are used to establish connections between various components in the circuit. Different types of wires, such as jumper wires or specific connectors, are used to ensure secure and reliable connections.

## 3. Circuit Diagram:



#### 4. Code:

```
#include <Servo.h> // servo library to run servo

Servo doorServo; //declaration of servo variable

// IR Sensor pin connected to Arduino at pin no 10
int IRsensor = 10;

void setup() {
    doorServo.attach(9); // Attach servo to pin 9
    pinMode(IRsensor, INPUT); //using ir sensor as input
    Serial.begin(9600);
}

void loop() {
    int IRValue = digitalRead(IRsensor);
    if (IRValue == HIGH) {
        // If IR sensor detects an object
        Serial.println("Object Detected! Door Opening...");
        doorServo.write(90); // Open door (servo angle adjustment may vary)
        delay(5000); // Delay for door opening time
    } else if (IRValue == LOW) {
        //if ir sensor don't find any object
        doorServo.write(0); // Close door (servo angle adjustment may vary)
        Serial.println("Door Closed.");
    }
}
```

## **5. Working:**

- The IR sensor detects the presence of an object.
- Upon detecting an object, the Arduino receives the signal from the IR sensor.
- The code processes this signal and triggers the servo motor to open the door by adjusting its angle.
- After a delay for the door to open fully, the system closes the door by adjusting the servo motor's angle again.
- The door remains closed for a set duration before the system resets, ready to detect the next object.

## **6. Procedure Step-wise:**

- Assemble the circuit as per the provided diagram.
- Upload the provided Arduino code to the Arduino Uno.
- Connect the door mechanism to the servo motor.
- Power up the system and test the IR sensor by placing an object in front of it.
- Observe the serial monitor for the system's response and the door's actions.

## **7. Applications:**

- Residential complexes
- Parking lots
- Commercial buildings
- Security checkpoints

## **8. Future Scope:**

- Integration with IoT for remote operation and monitoring
- Adding additional sensors for more accurate object detection
- Implementing a feedback system for door status monitoring