# Object Detection with Relay Triggering

## 1. Project Overview

The system detects obstacles using the IR sensor and triggers a relay when an obstacle is detected, which can be used to control external devices. The LCD screen displays the status of the system, indicating whether an obstacle is present and whether the relay is activated.

## 2. Components Required

- Arduino UNO
- Relay Module
- IR Sensor
- 16x2 LCD Screen with I2C interface
- Jumper Wires
- Breadboard(Optional)

## 3. Brief Overview of each of the components' uses

- Arduino Uno, a microcontroller that executes the code.
- IR sensor, used to detect obstacles.
- Relay module, used to switch on or off external devices.
- LCD screen, to display the output.

## 4. Pin Diagram

- **Arduino UNO and IR**
  VCC to 5V of Arduino
  GND to GND of Arduino
  OUT to D13 of Arduino

- **Arduino UNO and Relay**
  VCC to 5V on Arduino
  GND to GND of Arduino
  IN to digital pin 3 on Arduino

- **Arduino UNO and LCD**
  SDA to A4 of Arduino
  SCL to A5 of Arduino
  VCC to 5V of Arduino
  GND to GND of Arduino

## 5. Code Explanation

*#include <Wire.h>*
*#include <LiquidCrystal_I2C.h>*

This includes the library 'Wire.h' and 'LiquidCrystal_I2C.h' library to your Code which are required to simplify the use of the LCD display.

*LiquidCrystal_I2C lcd(0x27, 16, 2);*

This initializes and sets the address of the I2C to 0x27(Common address used in I2C modules). This tells the Arduino that the I2C module has the address 0x27. 16, and 2 refer to the columns and rows of the display.

*const int irPin = 13;*
*const int relayPin = 3;*

Since we connected OUT of IR sensor to D13, and IN to D3 on Arduino, we declare it here. If you connected to any other pins, you can change them here. Note that the Digital Pin you connected to, and the number you are declaring here are same.

*int irState = 0;*
*int relayState = 0;*

This sets the state of the IR sensor and Relay. 0 refers to "off" state.

*void setup() {}*
*void loop() {}*

This is the default function for all arduino codes. In *setup()*, we put codes that we need to execute only once. For example, to setup input and output devices.
In *loop()*, we put codes that we need to run repeatedly, as long as power is supplied to it.

*pinMode(irPin, INPUT_PULLUP);*
*pinMode(relayPin, OUTPUT);*
*relayState = LOW;*
*digitalWrite(relayPin, relayState);*
*lcd.begin();*
*lcd.setCursor(0, 0);*
*lcd.print("Obstacle");*
*lcd.setCursor(0, 1);*
*lcd.print("Detection");*
*delay(3000);*

These codes are inside the *setup()* function, meaning, on supplying power, it will execute once. In *pinmode()*, we let the arduino know if it is an input or an output device, we do so by first mentioning the pins(irPin and relayPin) followed by input or output.
Next, we set up relayState as LOW because at first we need the relay to be in "off" state. Next up, as we can see we use *digitalWrite()* to set the state of the relayPin. Alternatively, we can write this code in a single line. "*digitalWrite(relayPin, LOW);*"
Next, we have the codes that send commands to the lcd screen. We set it to start displaying texts. *lcd.setCursor()* is used to tell the lcd where the cursor should

start. It is a 16x2 Matrix starting from Column index starting from 0 to 15, and row index 0 and 1.
We use *lcd.print()* to display the required texts on the display.
*delay()* is used to tell the arduino to wait for a particular amount of time before moving to the next line of code. It is in milliseconds, so therefore 3000 means 3 seconds.

```
irState = digitalRead(irPin);

  if (irState == HIGH) {
    relayState = LOW;
    digitalWrite(relayPin, relayState);
    lcd.clear();
    lcd.print("No Obstacle");
    lcd.setCursor(0, 1);
    lcd.print("Relay OFF");
    delay(500);
  } else {
    relayState = HIGH;
    digitalWrite(relayPin, relayState);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Obstacle");
    lcd.setCursor(0, 1);
    lcd.print("Detected");
    delay(1000);
    lcd.clear();
    lcd.print("Relay ON");
    Serial.println("Obstacle Detected");
    delay(500);
  }
```

This set of code is in the *loop()* function, meaning it will execute continuously until power is turned off.
First up, we read the state of the IR sensor using *digitalRead()*.
If the IR state is HIGH, it means no objects are detected and the relay must be off, and if IR state is LOW, it means objects are detected and we must turn on the relay.