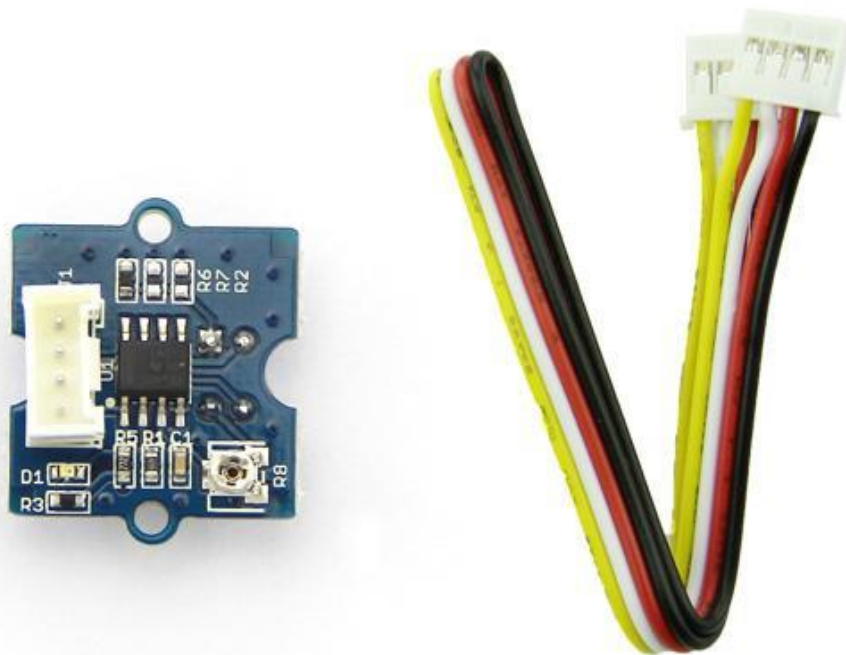


Grove - IR Distance Interrupter v1.2

Introduction

3.3V 5.0V Digital



Grove - IR Distance Interrupter is used to detect any object blocking the path of light. The module consists of an IR LED and a photosensor (phototransistor) pair. The light emitted by the IR LED gets reflected by any object placed in front of the sensor and this reflection is detected by the photosensor(phototransistor). Any white (or lighter) colored surface reflects more than black (or darker) colored surface.

When the reflected light is detected, it produces **Digital HIGH** (or Binary **1**) output on the **SIG** pin. The on-board LED indicator will also glow. If no reflection is detected or if the object is too far from the sensor, the output on the **SIG** pin stays at **Digital LOW** (Binary **0**). The on-board LED indicator will be off as well. The detectable range of this sensor is 7.5–40 cm. The module incorporates a Rail-to-Rail Operational Amplifier to amplify the output of phototransistor. There is a potentiometer which can be used to adjust the gain of the amplifier, that is, sensitivity of detection.

With this sensor, you can build the following (but not limited to) applications: **line following robots**, **optical encoders** and **object counting applications**.

Note

This product is mildly sensitive to non-IR radiations also and hence any bright light on photosensor impairs or disturbs IR light detection.

Tip

The instructions to use this product are same as **Grove - Infrared Reflective Sensor's**. You can use this product directly if you have used Grove - Infrared Reflective Sensor.

Version Tracker

Product version	Release date	Support status
Versions older than v1.2	June 2012	Not supported
Grove - IR Distance Interrupter v1.2(current version)	April 2016	Supported

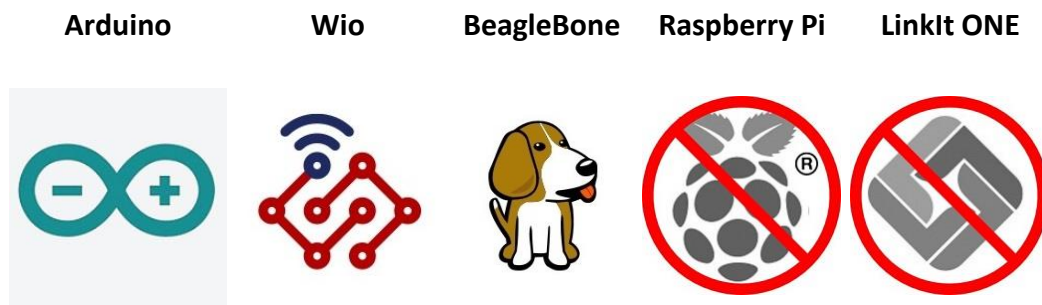
Features

- Grove compatible and easy to use
- Highly sensitive and reliable
- Longer detectable distance
- Adjustable sensitivity for various occasions
- More durable
-

Specifications

Parameter	Value
Operating voltage(V)	3.3 or 5 Volts
Operating current(mA)	Maximum: 20 mA
Effective detectable distance	7.5–40 cm
Reflective photosensor	datasheet
Output operational amplifiers	datasheet
Weight	2.5 g(for the module), 8.5 g(for all single package)

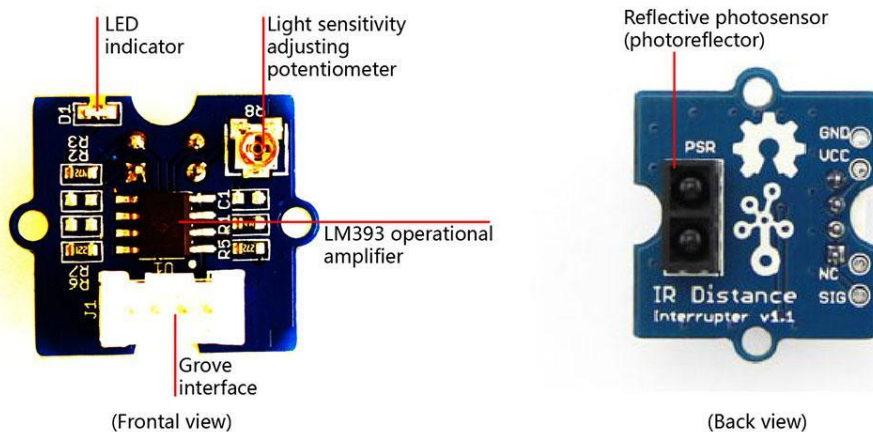
Platforms Supported



Caution

The platforms mentioned above as supported is/are an indication of the module's hardware or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Hardware Overview



- **RPR-359F Reflective photosensor**, Highly sensitive reflective photosensor.
- **LM393 operational amplifier**, rail-to-rail operational amplifier.
- **LED Indicator**, The LED will turn on when the received infrared light intensity exceeds a preset level.
- **Light sensitivity adjusting potentiometer**, adjust the sensitivity of reflective photosensor to light.

Package includes

Parts name	Quantity
Grove - Infrared Reflective Sensor	1 piece
Grove cable	1 piece

Getting Started

Let us see how to implement few basic applications with this module:

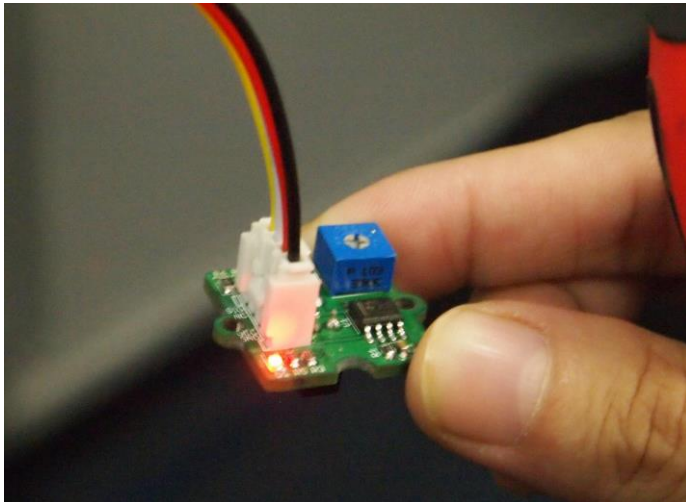
With Arduino

Material required

- Grove - IR Distance Interrupter v1.2 × 1
- Arduino UNO (other models also are fine) × 1
- [Grove cable](#) × 1
- [Grove - Base Shield](#) × 1

Connections

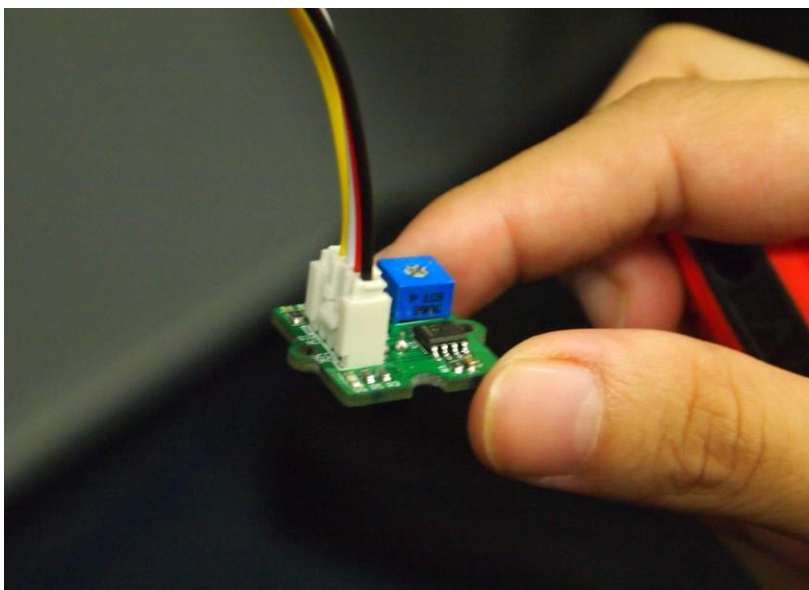
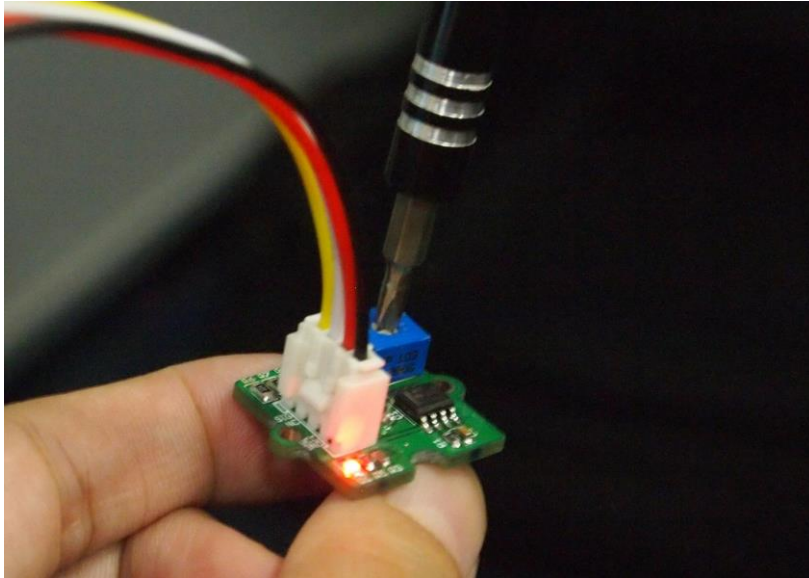
- 1.Connect Grove - IR Distance Interrupter v1.2 to Arduino UNO with Grove cable.
- 2.Place and hold the Reflective photosensor towards white(or light) colored surface.



- 3.Adjust the potentiometer with a screwdriver to change the sensitivity of reflective photosensor, until the LED indicator glows. As your rotate clockwise, the reflective photosensor will be more sensitive to light.

Note

Use a proper screw-driver to adjust the tiny potentiometer. Applying heavy pressure or frequent adjustments might damage the wiper of the potentiometer.



4. Create an Arduino sketch and copy the below code into it.

```
void setup() {  
  Serial.begin(9600);  
  pinMode(6, INPUT);  
}  
void loop() {  
  while(1) {  
    delay(500);  
    if(digitalRead(6) == LOW) {  
      Serial.println("Somebody is here.");  
    }  
  }  
}
```

```

    }
    else {
        Serial.println("Nobody.");
    }
}
}

```

5.Upload the code. If you do not know how to upload a Arduino sketch, please visit <https://www.arduino.cc/en/Guide/Windows> for Windows user or <https://www.arduino.cc/en/Guide/MacOSX> for Mac user. You can see the result as below.

6.When the path of light is blocked by some object, you would see “Somebody is here.” in Serial Terminal else you will see “Nobody.”

With Raspberry Pi

Material required

- Raspberry Pi (other models also are fine) × 1
- [GrovePi](#) or [Grovepi+](#) × 1
- [Grove cable](#) × 1

Hardware Connections and Software Work

1.You should have a Raspberry Pi and a GrovePi or GrovePi+. In this demo, we use GrovePi.

2.We assume you have built the development environment successful. If not, follow [this tutorial](#).

3.Connection:

Plug Grove - IR Distance Interrupter into port D4 on GrovePi with [Grove cable](#).

4.Navigate to the demos' directory, run the following command in a terminal.

```
cd yourpath/GrovePi/Software/Python/
```

Run the command in a terminal:

```
nano grove\_infrared\_distance\_interrupt.py
```

Copy and save the following code into it.

```
import time
import grovepi
```

```
# Connect the Grove Infrared Distance Interrupt Sensor to digital port D4
```

```
# SIG,NC,VCC,GND
sensor = 4

grovepi.pinMode(sensor,"INPUT")

while True:
    try:
        # Sensor returns LOW and onboard LED lights up when the
        # received infrared light intensity exceeds the calibrated level
        if grovepi.digitalRead(sensor) == 0:
            print "found something"
        else:
            print "nothing"

        time.sleep(.5)

    except IOError:
        print "Error"
```

5.Run the demo, run following command in terminal.

```
sudo python grove\_infrared\_distance\_interrupt.py
```