# How HC-SR501 PIR Sensor Works & Interface It With Arduino

Whether you want to build a home burglar alarm or a trail camera, or perhaps you want to wake up animated Halloween props when trick-or-treaters come to your door, then you should definitely consider getting an HC-SR501 Passive Infrared (PIR) sensor for yourself.

The PIR sensor allows you to detect when a person or animal moves in or out of sensor range. This sensor is what you’ll find in most modern security systems, automatic light switches, garage door openers and similar applications where we want to react to motion.the best part is that it consumes less than 2mA of current and can detect motion up to 7 meters (21 ft) with sensitivity control.

Before getting into the nitty-gritty, let’s first understand how a PIR sensor actually works.

## **How does a PIR sensor work?**

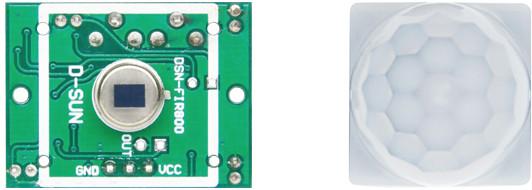
All objects, including the human body, at temperatures above absolute zero (0 Kelvin / -273.15 °C) emit heat energy in the form of infrared radiation. The hotter an object is, the more radiation it emits. This radiation is not visible to the human eye because it is emitted at infrared wavelengths. The PIR sensor is specifically designed to detect such levels of infrared radiation.

#### **Technical Specifications**

| **Specification** | **Value** |
| --- | --- |
| Sensor Type | PIR (Passive Infrared) |
| Detection Range | Up to 6 meters |
| Detection Angle | 120 degrees |
| Sensing Element | Dual-element pyroelectric |
| Output Type | Digital (High/Low) |
| Operating Voltage | 3.3V - 5V |
| Operating Temperature | -20°C to +60°C |

A PIR sensor consists of two main parts:

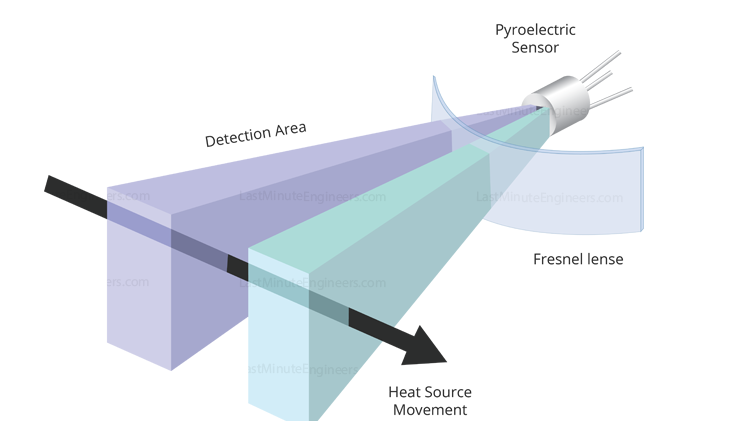
1. A pyroelectric sensor, which you can see in the image below as a round metal with a rectangular crystal in the center.
2. A special lens called a fresnel lens which Focuses the infrared signals on the pyroelectric sensor.



### **The Pyroelectric Sensor**

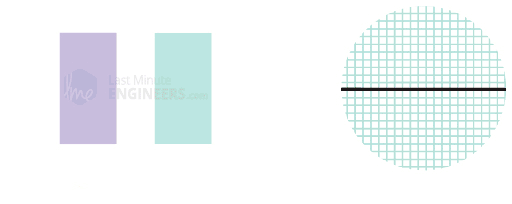
A pyroelectric sensor consists of a window with two rectangular slots and is made of a material (typically coated silicon) that allows infrared radiation to pass through. Behind the window, there are two separate infrared sensor electrodes, one responsible for producing the positive output and the other for producing the negative output.

The two electrodes are wired such that they cancel each other out. This is because we are looking for changes in IR levels and not ambient IR levels. That’s why when one half sees more or less IR radiation than the other, we get the output.



When there is no movement around the sensor, both slots detect the same amount of infrared radiation, resulting in a zero output signal.

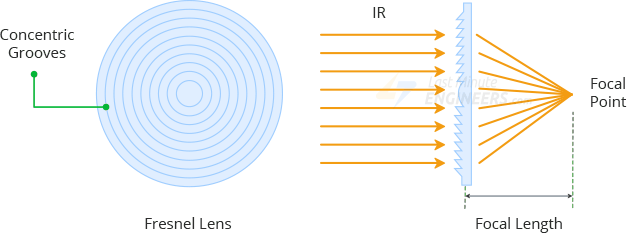
But when a warm body like a human or an animal passes by, it first intercepts half of the sensor. This causes a positive differential change between the two halves. When the warm body intercepts the other half of the sensor (leaves the sensing region), the opposite happens, and the sensor produces a negative differential change. By reading this change in voltage, motion is detected.



### **The Fresnel Lens**

You may feel that the Fresnel lens used here is not really doing anything. In fact, this is what increases the range and field of view of the PIR sensor. Its slim, lightweight construction and excellent light gathering capability make it extremely useful for making PIRs small in size yet powerful.

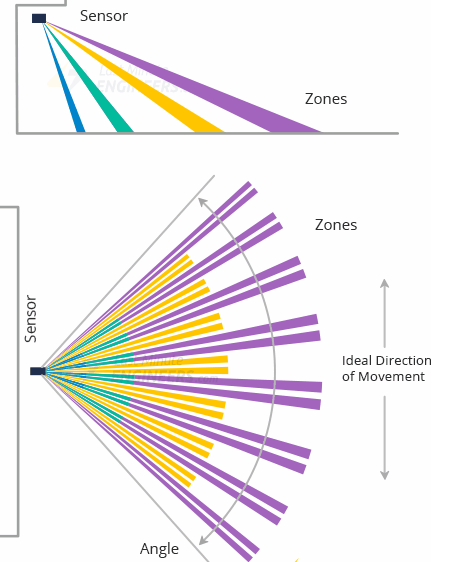
A Fresnel lens consists of a series of concentric grooves carved into the plastic. These contours act as individual refracting surfaces, gathering parallel light rays at a focal point. As a result a Fresnel lens, although smaller in size, is able to focus light similarly to a conventional optical lens.



In reality, to increase the range and field of view of the PIR sensor, the lens is divided into several facet-sections, each section of which is a separate Fresnel lens.

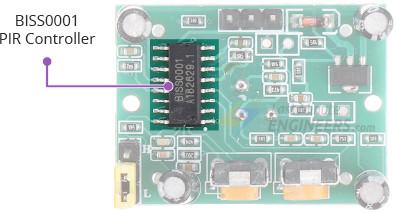


The different faceting and sub-lenses create a range of detection areas/zones, interleaved with each other. That’s why the centers of the lenses are ‘inconsistent’ in the image above – every other one points to a different half of the PIR sensing element.



### **BISS0001 PIR Controller:**

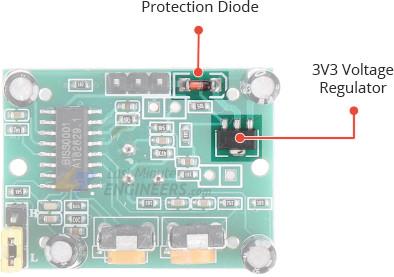
This chip takes the output from the Pyroelectric sensor and does some minor processing on it to emit a digital output pulse.



### **Power**

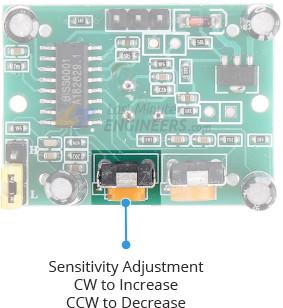
The module comes with a 3.3V precision voltage regulator, so it can be powered by any DC voltage from 4.5 to 12 volts, although 5V is commonly used.

The module comes with a protection diode (also known as a safety diode) to protect the module from reverse voltage and current. So even if you accidentally connect the power with incorrect polarity, your module will not be damaged.



### **Sensitivity Adjustment**

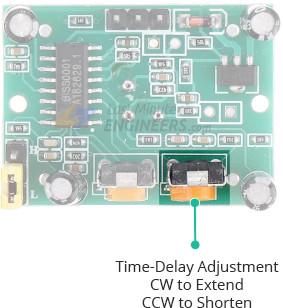
The PIR sensor has a potentiometer on the back to adjust the sensitivity.



This potentiometer sets the maximum detection range. Sensitivity can be adjusted over a range of approximately 3 meters to 7 meters (9 to 21 feet). However the topology of your room can affect the actual range you get. Rotating the pot clockwise will increase the sensitivity and thus the range, and vice versa.

### **Time-Delay Adjustment**

There is another potentiometer on the back of the PIR sensor to adjust the Time-Delay.



This potentiometer sets how long the output will remain HIGH after motion is detected. It can be adjusted from 1 second to about 3 minutes. Turning the potentiometer clockwise increases the delay, while turning the potentiometer counter-clockwise decreases the delay.

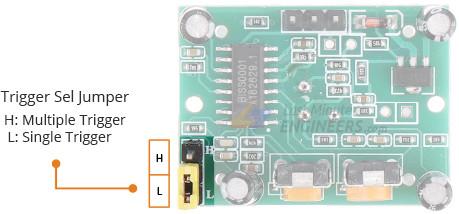
### **Trigger Selection Jumper**

There are two trigger modes that determine how the sensor will react when motion is detected.

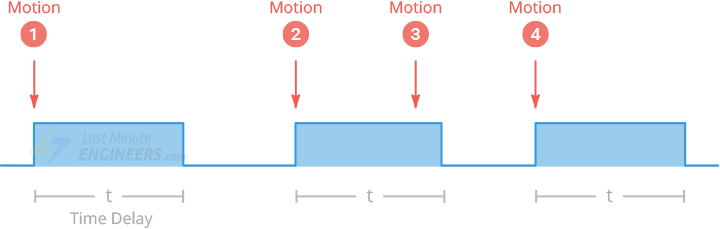
Single Trigger Mode: The constant motion will cause a single trigger.

Multiple Trigger Mode: The constant motion will cause a series of triggers.

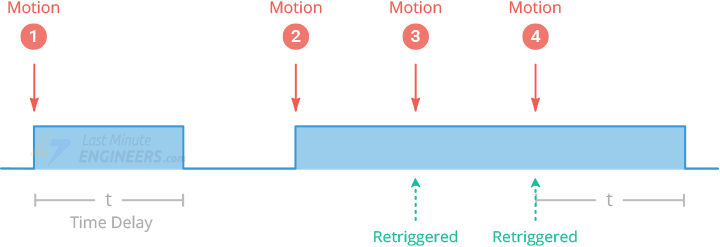
The board comes with a berg jumper (some modules have a solder bridge jumper) allowing you to choose one of two modes:



**L** – Selecting this will set the single trigger mode. In this mode the output goes HIGH as soon as motion is detected and remains HIGH for a period determined by the Time-Delay potentiometer. Further detection is blocked until the output returns to LOW at the end of the time delay. If there is still motion, the output will go HIGH again. As you can see in the image below, Motion #3 is completely ignored.

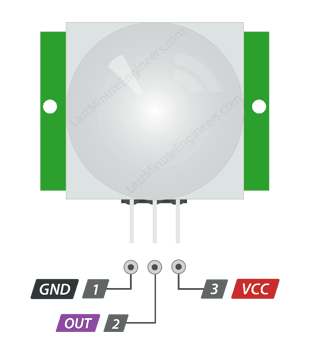


H – Selecting this will set the multiple trigger mode. In this mode the output goes HIGH as soon as motion is detected and remains HIGH for a period determined by the Time-Delay potentiometer. Unlike single trigger mode, further detection is not blocked, so the time delay is reset each time motion is detected. Once the motion stops, the output returns to LOW only after a time delay. Hence the name multiple trigger mode.



## HC-SR501 PIR Sensor Pinout

The HC-SR501 has a 3-pin connector. The markings are hidden by the Fresnel lens, so refer to the following image for pinout.



VCC is the power supply for the sensor. You can connect an input voltage anywhere between 5 to 12V to this pin, although 5V is commonly used.

Output pin is the 3.3V TTL logic output. It goes HIGH when motion is detected and goes LOW when idle (no motion detected).

GND is the ground pin.

### Connection

The steps to set up the connection are listed below:

* Connect the Vcc terminal of the PIR sensor to the 5V pin of the Arduino board.
* Connect the Output terminal of the PIR sensor to pin 8 of the Arduino board.
* Connect the GND terminal of the PIR sensor to the Ground pin of the Arduino board.
* Connect the positive leg of the LED in series with 220 Ohm resistor to pin 13 of the Arduino board.
* Connect the negative terminal of the LED to the Ground pin of the Arduino board.

Consider the below code:

**int** LEDpin = 13; // LED pin

**int** PIRpin = 8; // The pin of Arduino connected to the PIR output

**int** PIRvalue = 0; // It specifies the status of PIR sensor

**void** setup() {

pinMode(LEDpin, OUTPUT);

pinMode(PIRpin, INPUT);

// the output from the sensor is considered as input for Arduino

Serial.begin(9600);

}

**void** loop()

{

PIRvalue = digitalRead(PIRpin);

**if** (PIRvalue == HIGH)

{

digitalWrite(LEDpin, HIGH);

// turn ON LED if the motion is detected

Serial.println("hello, I found you...heyyy..");

}

**else**

{

digitalWrite(LEDpin, LOW);

// LED will turn OFF if we have no motion

Serial.println("I cannot find you");

delay(1000);

}

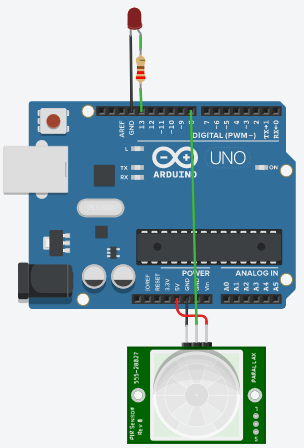
}

### Steps to upload the code to the project

The steps are listed below:

* Open the Arduino IDE.
* Select the type of board from Tools -> Board -> Arduino UNO.
* Select the port from Tools -> Port -> COM..
* Upload the sketch to the connection diagram.

### Connection Diagram



Where are assign the PIRpin 8(digital pin) as INPUT and the LEDpin 13 as OUTPUT.if value of pin 8 is HIGH then the LED will turn on and the LED will turn off when the value pin 8 is low.