

# PIR Motion Sensor Tutorial

by [adafruit](#) on August 23, 2009

## Table of Contents

PIR Motion Sensor Tutorial .....	1
Intro: PIR Motion Sensor Tutorial .....	2
Step 1: How does it work? .....	3
Step 2: Lenses .....	4
Step 3: Connecting to your PIR .....	6
Step 4: Testing your PIR .....	6
Step 5: Retriggering .....	7
Step 6: Changing pulse time and timeout length .....	8
Step 7: Project examples .....	8
Step 8: Reading PIR sensors .....	9
Related Instructables .....	10
Comments .....	11



Author:adafruit [Adafruit Industries](#)

All-original DIY electronics kits - Adafruit Industries is a New York City based company that sells kits and parts for original, open source hardware electronics projects featured on [www.adafruit.com](http://www.adafruit.com) as well as other cool open source tronix' that we think are interesting and well-made.

## Intro: PIR Motion Sensor Tutorial

### Pyroelectric ("Passive") InfraRed sensors:

"What is a PIR sensor?"

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

PIRs are basically made of a [pyroelectric sensor](#) (which you can see above as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

Along with the pyroelectric sensor is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the [BISS0001 \("Micro Power PIR Motion Detector IC"\)](#), undoubtedly a very inexpensive chip. This chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor.

For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. Note that PIRs won't tell you how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance (although it can be hacked somewhere) and they are also sometimes set off by house pets. Experimentation is key!

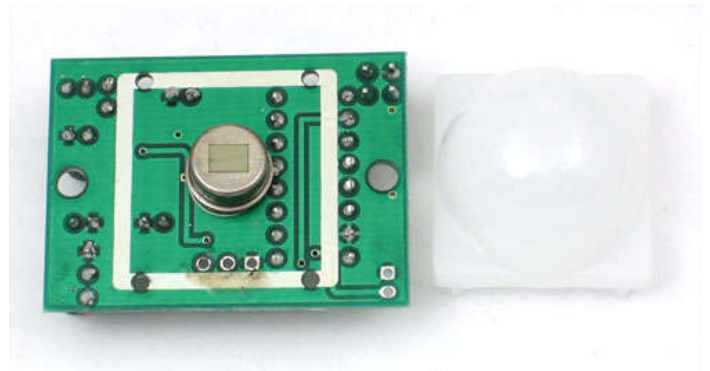
### Some basic stats

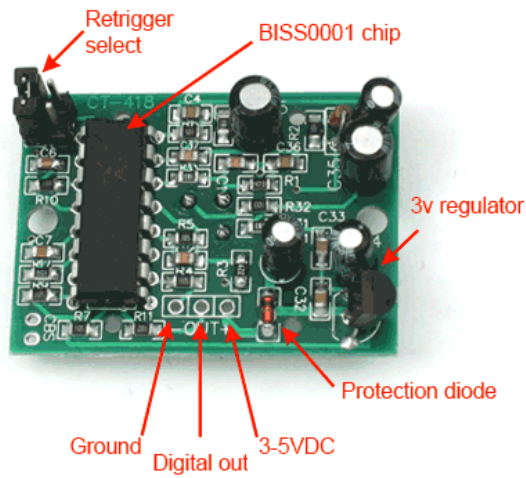
These stats are for the PIR sensor in the Adafruit shop which is very much [like the Parallax one](#). Nearly all PIRs will have slightly different specifications, although they all pretty much work the same. If there's a datasheet, you'll want to refer to it

- **Size:** Rectangular
- **Price:** [\\$10.00 at the Adafruit shop](#)
- **Output:** Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor.
- **Sensitivity range:** up to 20 feet (6 meters) 110 degrees x 70 degrees detection range
- **Power supply:** 3.3V - 5V input voltage,
- [BIS0001 Datasheet](#) (the decoder chip used)
- [RE200B datasheet](#) (most likely the PIR sensing element used)
- [NL11NH datasheet](#) (equivalent lens used)
- [Parallax Datasheet on their version of the sensor](#)

More links!

- [A great page on PIR sensors from GLOLAB](#)
- [NYU sensor report](#)





### Step 1: How does it work?

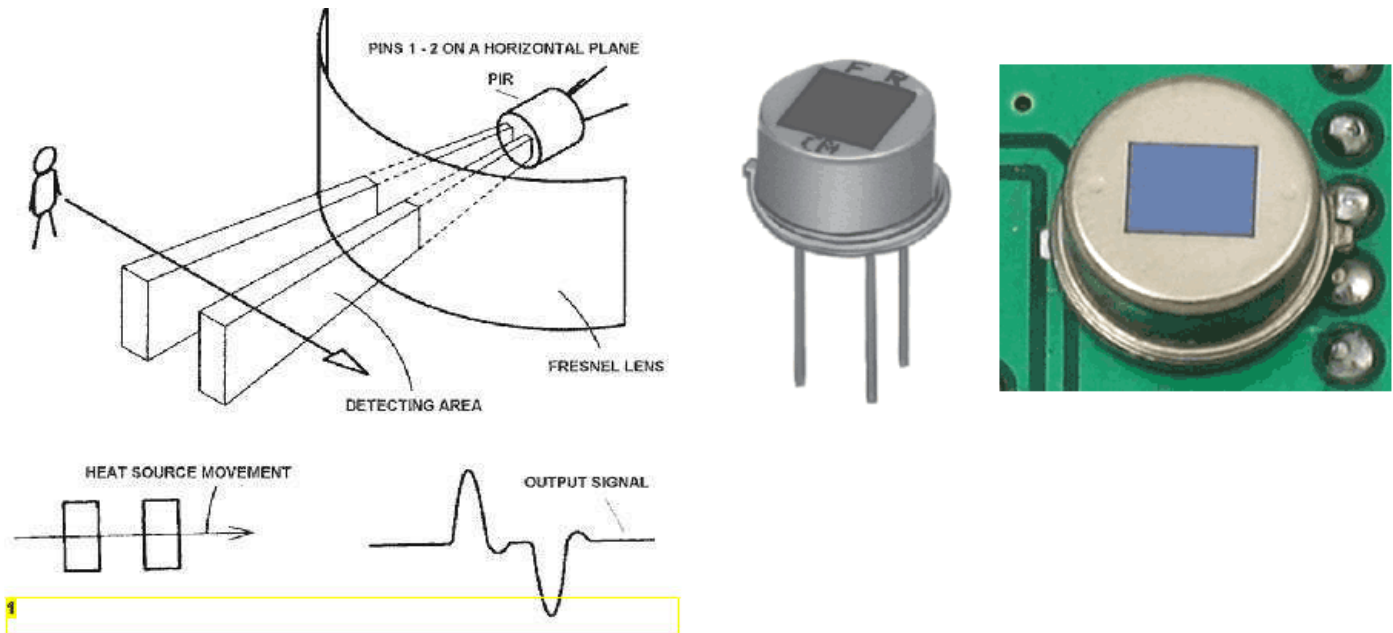
PIR sensors are more complicated than many of the other sensors explained in these tutorials (like photocells, FSRs and tilt switches) because there are multiple variables that affect the sensors input and output. To begin explaining how a basic sensor works, we'll use the rather nice diagram below (if anyone knows where it originates plz let me know).

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

#### The PIR sensor itself

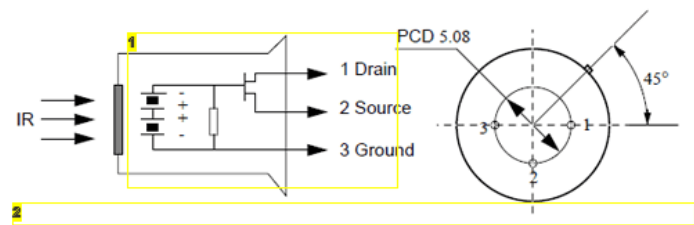
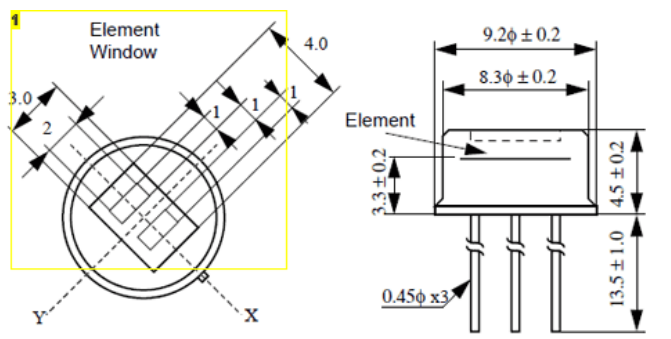
The IR sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element. Behind the window are the two balanced sensors.

Check out the images for more details:



#### Image Notes

- [Citation needed]



**Image Notes**

1. In this diagram you can see the element window, the two pieces of sensing material.
2. Image from RE200B datasheet: <http://www.ladyada.net/media/sensors/RE200B.pdf>

**Image Notes**

1. This image shows the internal schematic. There is actually a JFET inside (a type of transistor) which is very low-noise and buffers the extremely high impedance of the sensors into something a low-cost chip (like the BIS0001) can sense.
2. Image from RE200B datasheet: <http://www.ladyada.net/media/sensors/RE200B.pdf>

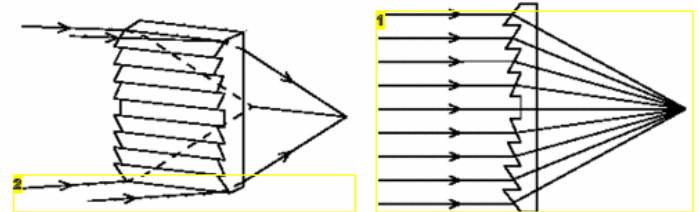
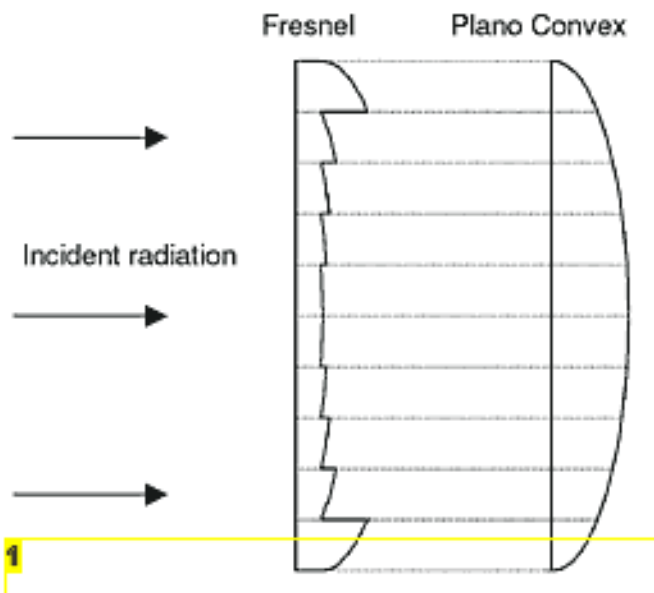
**Step 2: Lenses**

PIR sensors are rather generic and for the most part vary only in price and sensitivity. Most of the real magic happens with the optics. This is a pretty good idea for manufacturing: the PIR sensor and circuitry is fixed and costs a few dollars. The lens costs only a few cents and can change the breadth, range, sensing pattern, very easily.

In the diagram above, the lens is just a piece of plastic, but that means that the detection area is just two rectangles. Usually we'd like to have a detection area that is much larger. To do that, we use a simple lens such as those found in a camera: they condense a large area (such as a landscape) into a small one (on film or a CCD sensor). For reasons that will be apparent soon, we would like to make the PIR lenses small and thin and moldable from cheap plastic, even though it may add distortion. For this reason the sensors are actually Fresnel lenses (see image below).

OK, so now we have a much larger range. However, remember that we actually have two sensors, and more importantly we don't want two really big sensing-area rectangles, but rather a scattering of multiple small areas. So what we do is split up the lens into multiple sections, each section of which is a fresnel lens.

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

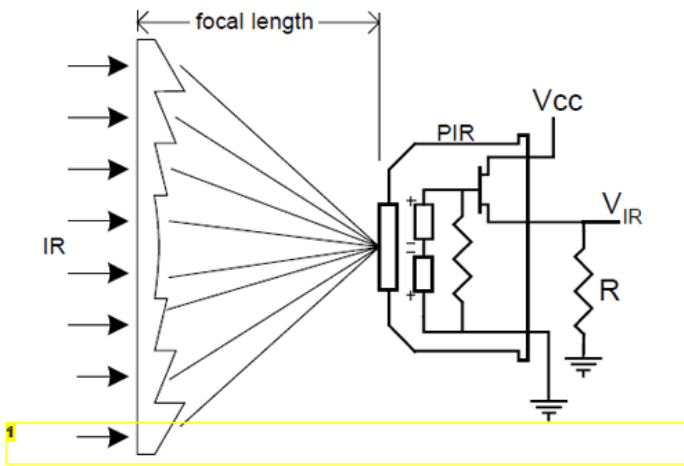


**Image Notes**

1. The Fresnel lens condenses light, providing a larger range of IR to the sensor.
2. Image from BHlens.com: [http://www.bhlens.com/linear\\_fresnel\\_lens.aspx](http://www.bhlens.com/linear_fresnel_lens.aspx)

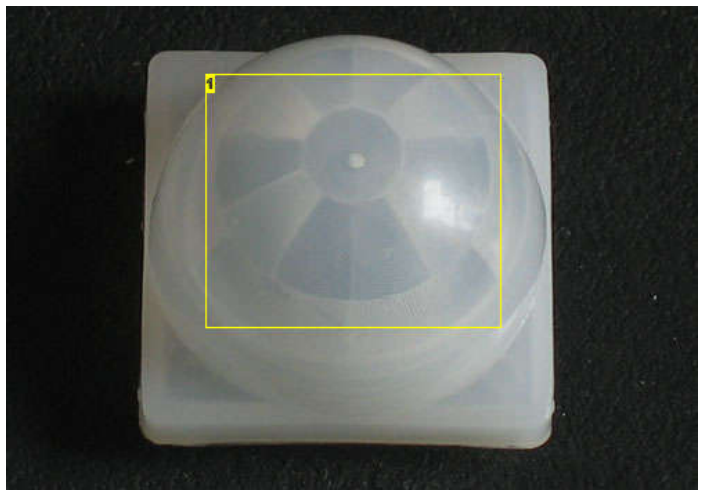
**Image Notes**

1. Image from Sensorsmag: <http://www.sensorsmag.com/articles/0403/35/main.shtml>



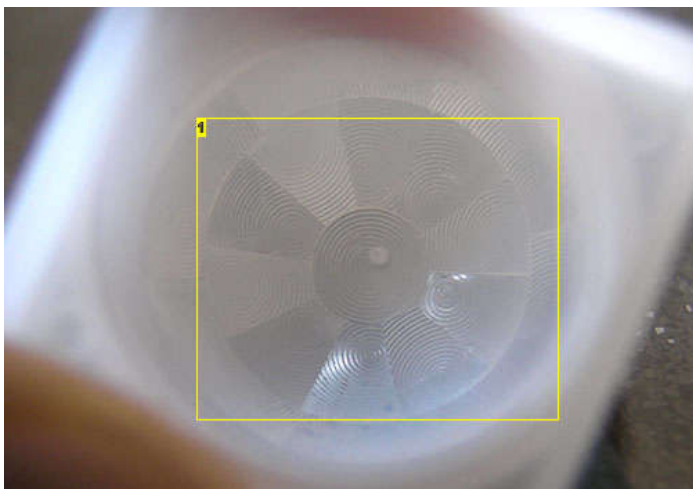
**Image Notes**

1. Image from Cypress appnote 2105:  
<http://www.ladyada.net/media/sensors/an2105.pdf>



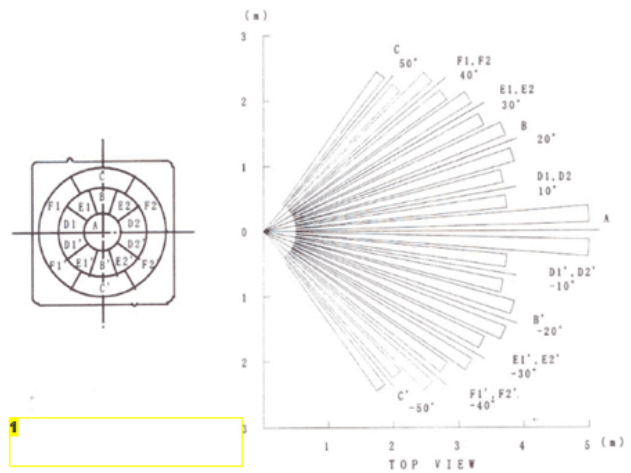
**Image Notes**

1. Here you can see the multiple facet-sections



**Image Notes**

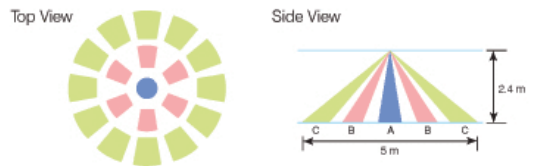
1. This macro shot shows the different Fresnel lenses in each facet!



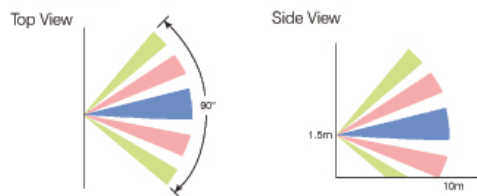
**Image Notes**

1. Images from NL11NH datasheet:  
<http://www.ladyada.net/media/sensors/NL11NH.pdf>

**2 Ceiling Mount**



**1 Wall Mount**



**Image Notes**

1. Images from NL11NH datasheet:  
<http://www.ladyada.net/media/sensors/NL11NH.pdf>

**Image Notes**

1. Here is another image, more qualitative but not as quantitative. (Note that the sensor in the Adafruit shop is 110° not 90°)  
 2. Image from IR-TEC: <http://www.irtec.com/ms-360.htm>

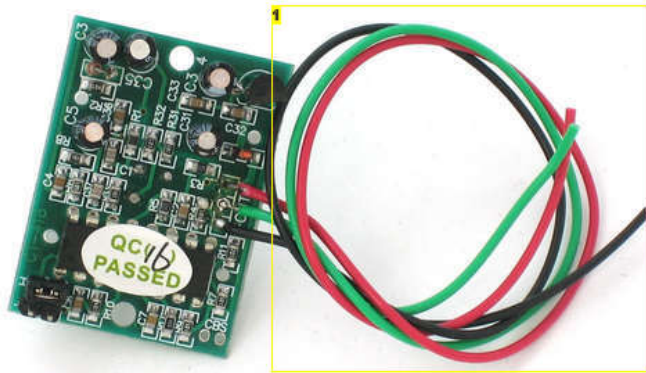
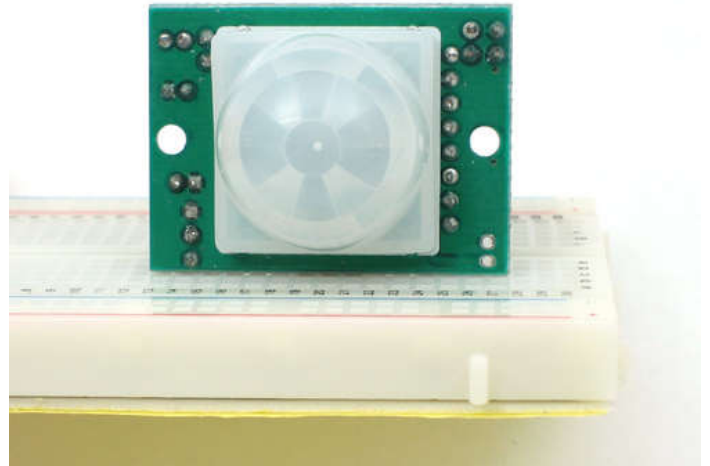
### Step 3: Connecting to your PIR

Most PIR modules have a 3-pin connection at the side or bottom. The pinout may vary between modules so triple-check the pinout! It's often silkscreened on right next to the connection. One pin will be ground, another will be signal and the final one will be power. Power is usually 3-5VDC input but may be as high as 12V. Sometimes larger modules don't have direct output and instead just operate a relay in which case there is ground, power and the two switch connections.

The output of some relays may be 'open collector' - that means it requires a pullup resistor. If you're not getting a variable output be sure to try attaching a 10K pullup between the signal and power pins.

An easy way of prototyping with PIR sensors is to connect it to a breadboard since the connection port is 0.1" spacing. Some PIRs come with header on them already, the ones from Adafruit don't as usually the header is useless to plug into a breadboard.

By soldering in 0.1" right angle header, a PIR is easily installed into a breadboard!



#### Image Notes

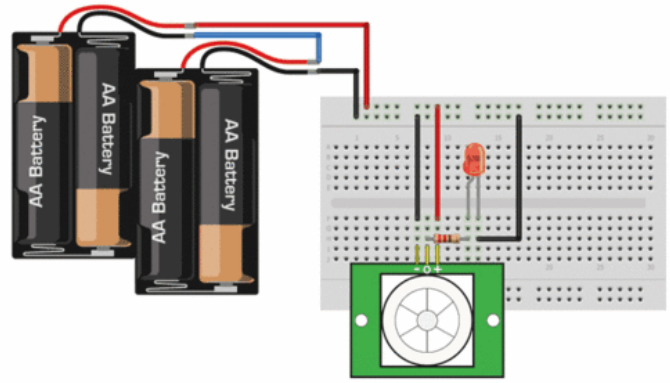
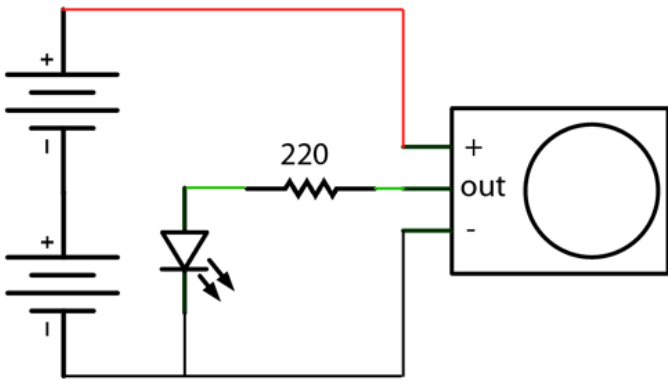
1. Most people want to position PIRs in a particular location and often times that's far from the other electronics, in which case wires will work just fine.

### Step 4: Testing your PIR

Once you have your PIR wired up its a good idea to do a simple test to verify that it works the way you expect. This test is also good for range testing. Simply connect 3-4 alkaline batteries (make sure you have more than 3.5VDC out but less than 6V by checking with your multimeter!) and connect **ground** to the - pin on your PIR. **Power** goes to the + pin. Then connect a basic red LED (red LEDs have lower forward voltages than green or blue so they work better with only the 3.3v output) and a 220 ohm resistor (any value from 100 ohm to 1.0K ohm will do fine) to the **out** pin as shown. Of course, the LED and resistor can swap locations as long as the LED is oriented connection and connects between **out** and **ground**

Now when the PIR detects motion, the output pin will go "high" to 3.3V and light up the LED!

Once you have the breadboard wired up, insert batteries and wait 30-60 seconds for the PIR to 'stabilize'. During that time the LED may blink a little. Wait until the LED is off and then move around in front of it, waving a hand, etc. to see the LED light up!



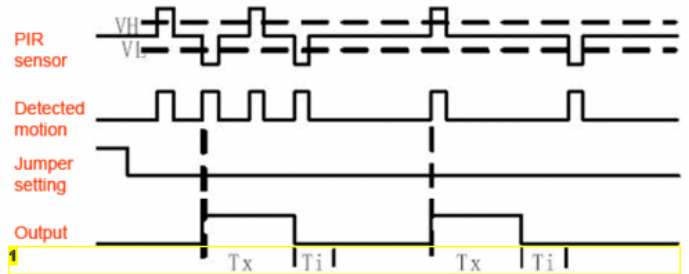
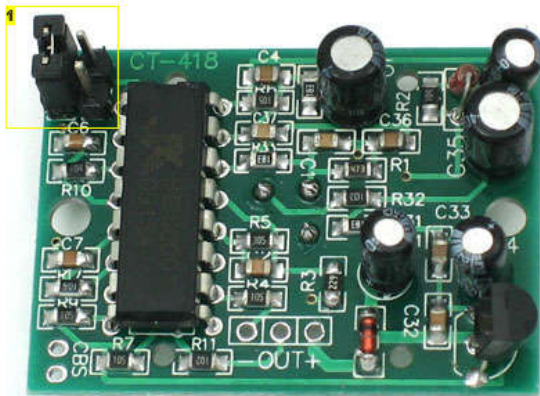
### Step 5: Retriggering

Once you have the LED blinking, look on the back of the PIR sensor and make sure that the jumper is placed in the L position as shown below.

Now set up the testing board again. You may notice that when connecting up the PIR sensor as above, the LED does not stay on when moving in front of it but actually turns on and off every second or so. That is called "non-retriggering".

Now change the jumper so that it is in the H position. If you set up the test, you will notice that now the LED does stay on the entire time that something is moving. That is called "retriggering".

For most applications, "retriggering" (jumper in H position) mode is a little nicer. If you need to connect the sensor to something edge-triggered, you'll want to set it to "non-retriggering" (jumper in L position).

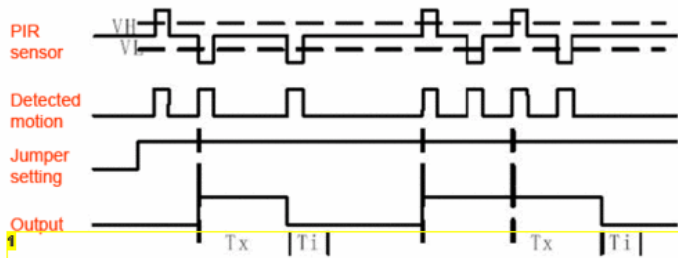


#### Image Notes

1. (The graphs above are from the BISS0001 datasheet, they kinda suck)

#### Image Notes

1. Once you have the LED blinking, look on the back of the PIR sensor and make sure that the jumper is placed in the L position.



#### Image Notes

1. (The graphs above are from the BISS0001 datasheet, they kinda suck)

## Step 6: Changing pulse time and timeout length

There are two 'timeouts' associated with the PIR sensor. One is the "Tx" timeout: how long the LED is lit after it detects movement. The second is the "Ti" timeout which is how long the LED is guaranteed to be off when there is no movement. These are not easily changed but if you're handy with a soldering iron it is within reason.

First, lets take a look at the BISS datasheet again (see image below)

Determining R10 and R9 isnt too tough. Unfortunately this PIR sensor below is mislabeled (it looks like they swapped R9 R17). You can trace the pins by looking at the BISS001 datasheet and figuring out what pins they are - R10 connects to pin 3 and R9 connects to pin 7. the capacitors are a little tougher to determine, but you can 'reverse engineer' them from timing the sensor and solving!

For the sensor in the Adafruit shop:

**Tx is =  $24576 * R10 * C6 = \sim 1.2$  seconds**  
**R10 = 4.7K and C6 = 10nF**

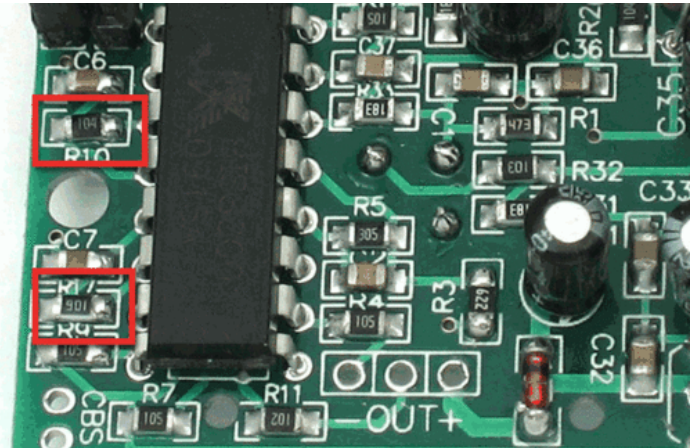
Likewise,

**Ti =  $24 * R9 * C7 = \sim 1.2$  seconds**  
**R9 = 470K and C7 = 0.1uF**

You can change the timing by swapping different resistors or capacitors. For a nice tutorial on this, see Keith's PIR hacking page

Tx = The time duration during which the output pin (Vo) remains high after triggering.  
Ti = During this time period, triggering is inhibited. See timing charts for details.

**Tx**  $\approx 24576 \times R10 \times C6$ ;    **Ti**  $\approx 24 \times R9 \times C7$ . (ref to schematic)



## Step 7: Project examples

A USB-powered singing and blinking Mario mushroom (there's a video on the site!)



Rain Umbrellas



Testing a PIR sensor for interfacing to Max/MSP for an interactive garden

<http://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/>



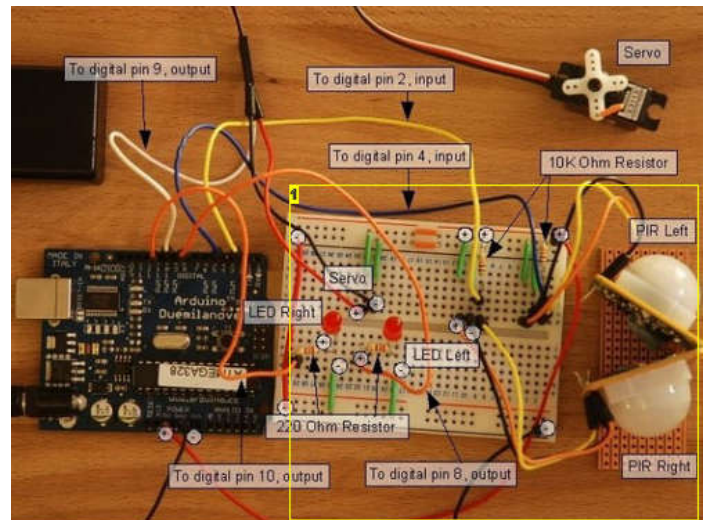
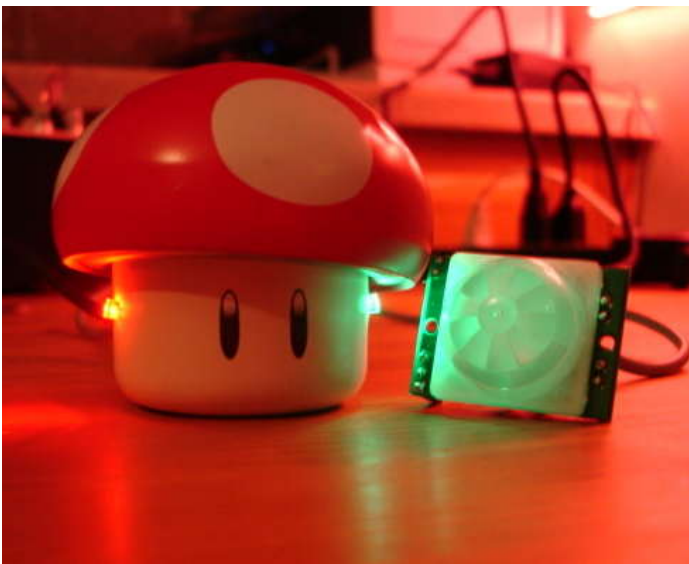


A home-made security system using PIR sensors (which is built into a Start Trek panel!)



PIR sensor + Arduino + Servo = automatic cat door!

A PIR-based remote camera trigger (also by Lucky Larry!)



#### Image Notes

1. A 2-PIR motion tracker by Lucky Larry: <http://luckylarry.co.uk/2009/07/arduino-very-basic-motion-tracking-with-2-pir-sensors/>

### Step 8: Reading PIR sensors

Connecting PIR sensors to a microcontroller is really simple. The PIR acts as a digital output so all you need to do is listen for the pin to flip high (detected) or low (not detected).

Its likely that you'll want rerigging, so be sure to put the jumper in the **H** position!

Power the PIR with 5V and connect ground to ground. Then connect the output to a digital pin. In this example we'll use pin 2.

The code is very simple, and is basically just keeps track of whether the input to pin 2 is high or low. It also tracks the state of the pin, so that it prints out a message when motion has started and stopped:

```
/* PIR sensor tester*/
```

```
int ledPin = 13; // choose the pin for the LED
int inputPin = 2; // choose the input pin (for PIR sensor)
int pirState = LOW; // we start, assuming no motion detected
int val = 0; // variable for reading the pin status
```

<http://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/>

```

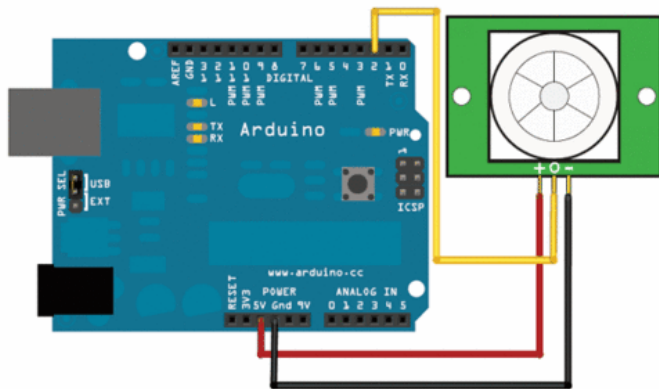
void setup() {
pinMode(ledPin, OUTPUT); // declare LED as output
pinMode(inputPin, INPUT); // declare sensor as input

Serial.begin(9600);
}

void loop(){
val = digitalRead(inputPin); // read input value
if (val == HIGH) { // check if the input is HIGH
digitalWrite(ledPin, HIGH); // turn LED ON
if (pirState == LOW) {
// we have just turned on
Serial.println("Motion detected!");
// We only want to print on the output change, not state
pirState = HIGH;
}
} else {
digitalWrite(ledPin, LOW); // turn LED OFF
if (pirState == HIGH){
// we have just turned of
Serial.println("Motion ended!");
// We only want to print on the output change, not state
pirState = LOW;
}
}
}
}

```

Don't forget that there are some times when you don't need a microcontroller. A PIR sensor can be connected to a relay (perhaps with a transistor buffer) without a micro!



### Related Instructables



**Arduino Controlled Motion Sensor**  
by LemonSlice



**Arduino Basics: PIR Sensors** by frenzy



**Weekly Project: Follow Me--the Mobile Beverage Cooler** by PSdp



**Prepare some surplus PIR sensors for robotics** by frickelkram



**Getting started with the Maxbotix sonar sensor - quick start guide** by adafruit



**Super loud security alarm** by mortaldoom780

## Comments

7 comments [Add Comment](#)



**Vick Jr** says:

Jan 15, 2011. 6:44 AM [REPLY](#)

I'm going to use one of these to detect opponents in air-soft wars. Is there any material that i can cover the sensor with to protect it from BBs but will still allow IR light through?



**Auger Duval** says:

Jan 7, 2011. 12:32 PM [REPLY](#)

I have just the sensor (TO9 can package) itself. A PIR D203S. The datasheet tells me how to hook it up: 5V, Gnd and Signal (it is an analog device). But it tells me very little about the signal I will be getting. Short of dragging out my O-scope and doing the testing; do you know what type of signal I should expect? This module from LA you use is nice, but I'm using my own uController so I wouldn't get the clean digital signal you show. BTW Wonderful writeup! you would make my old college profs proud.



**Auger Duval** says:

Jan 7, 2011. 12:41 PM [REPLY](#)

opps, it a TO-5 package not a TO-9



**kyle brinkerhoff** says:

Oct 29, 2010. 6:34 PM [REPLY](#)

huh, got 10 of these super cheap from my school used, now i actually know what they do!



**Brennn10** says:

Feb 4, 2010. 9:49 AM [REPLY](#)

Are you still selling the sensors on your website?



**robmiester** says:

Aug 24, 2009. 11:30 AM [REPLY](#)

Great info. I always wondered if you could take the lens/globe off. (looks like not a good idea)



**XxShadowxX14** says:

Aug 24, 2009. 9:52 AM [REPLY](#)

love it. there are a great number of instructables with just schematics to their projects using different components like motion sensors without going into detail of how exactly they work, and their wiring, so if you ever wanted to improve on, or create your own project using the component, you wouldnt know what your doing. this on the other hand gives a great amount of information and knowledge on using the motion sensor. maybe in the near future you could do another instructable similar to this, but using RFID chips?? -Shadow