

# PIR MOTION SENSOR

Wenwen & Yang



# PIR MOTION SENSOR

- This is a simple to use motion sensor. Power it up and wait 1-2 seconds for the sensor to get a snapshot of the still room. If anything moves after that period, the 'alarm' pin will go low.



<https://www.sparkfun.com/products/8630>



# PIR MOTION SENSOR

- This unit works great from 5 to 12V (datasheet shows 12V). You can also install a jumper wire past the 5V regulator on board to make this unit work at 3.3V. Sensor uses 1.6mA@3.3V.



<https://www.sparkfun.com/products/8630>

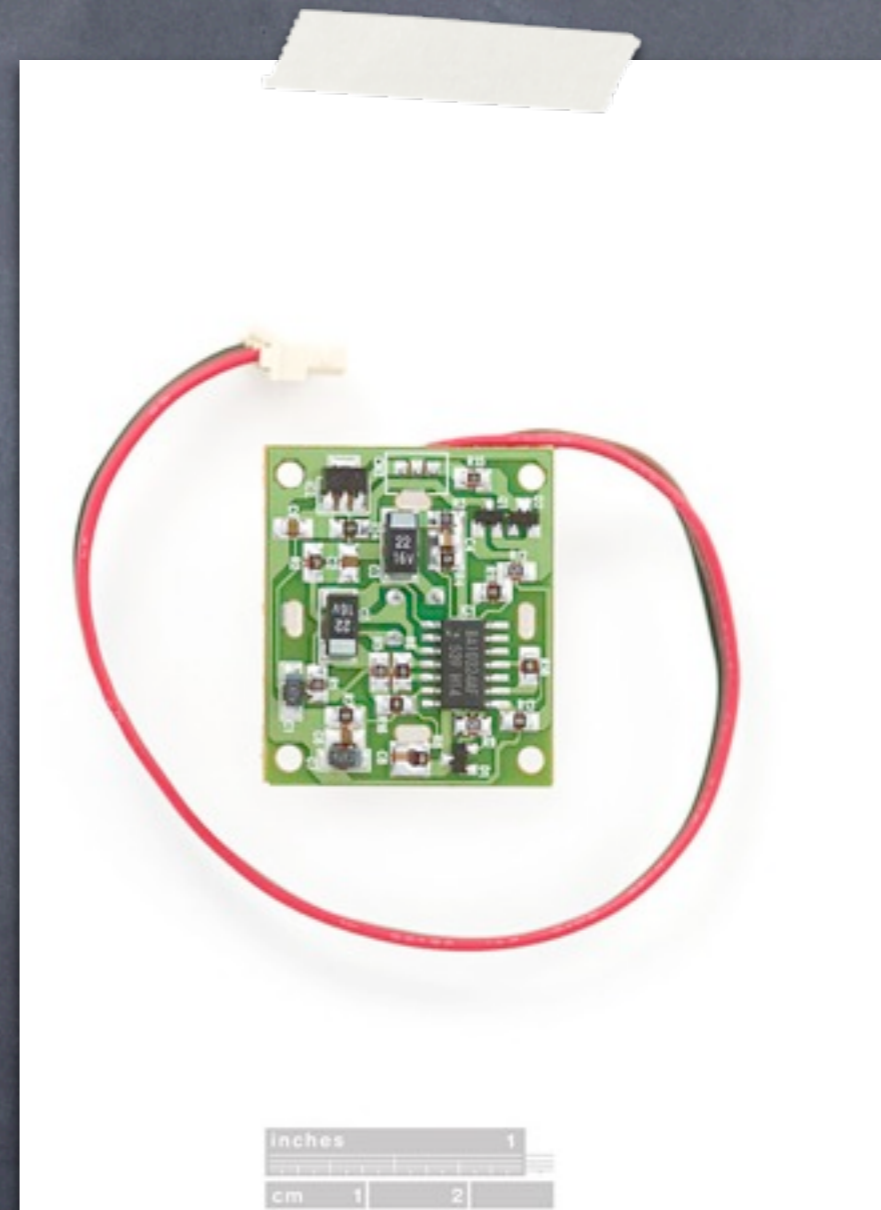


# PIR MOTION SENSOR

- The alarm pin is an open collector meaning you will need a pull up resistor on the alarm pin. The open drain setup allows multiple motion sensors to be connected on a single input pin. If any of the motion sensors go off, the input pin will be pulled low.

The connector is slightly odd but has a 0.1" pitch female connector making it compatible with jumper wires and 0.1" male headers.

<https://www.sparkfun.com/products/8630>





# PIR MOTION SENSOR

- **Note:** The colors of the wires can vary greatly. Looking at the unit from the top, the middle wire is ground, the left-most wire is the alarm, and the right-most is the power.

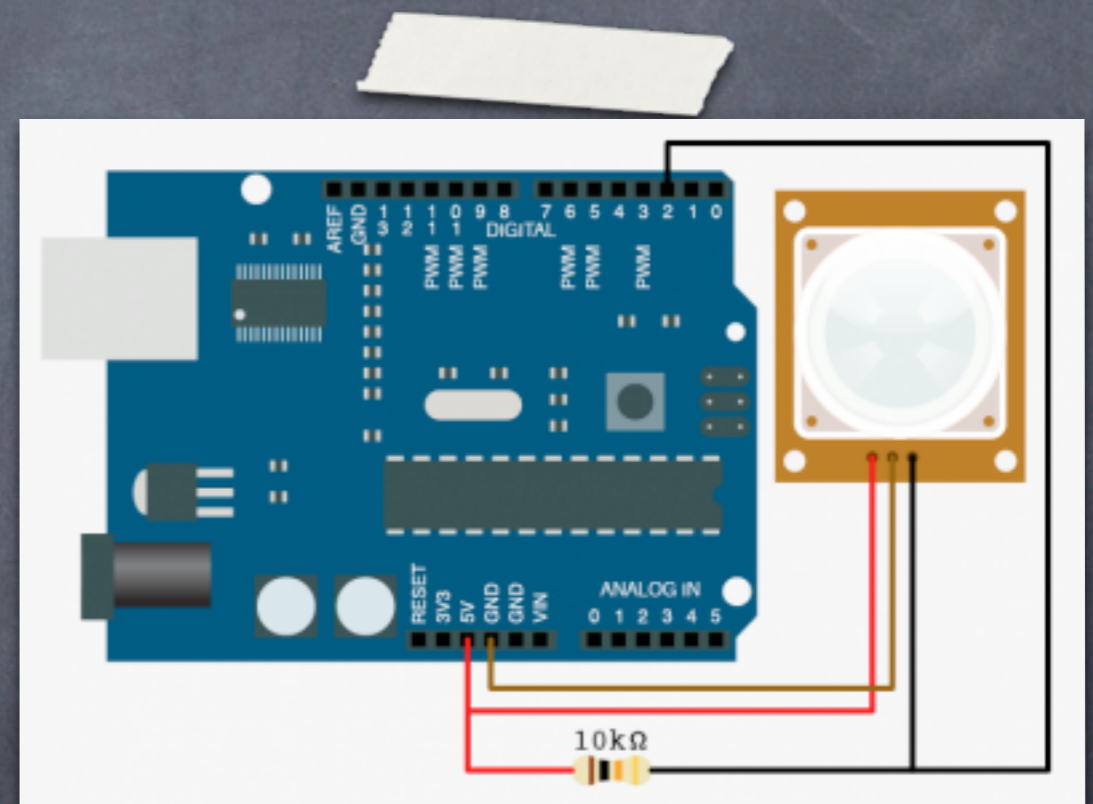
<https://www.sparkfun.com/products/8630>



# PIR MOTION SENSOR

## Tutorial

- Hooking one up to your Arduino is pretty simple, but you need to make sure not to fall for the manufacturer's trap. The red wire is V+ as you would think, the brown wire is **ground**, NOT the black wire. The black wire is the signal wire. To make matters even worse, the manufacturer uses a B/W image to show you what wire is what.



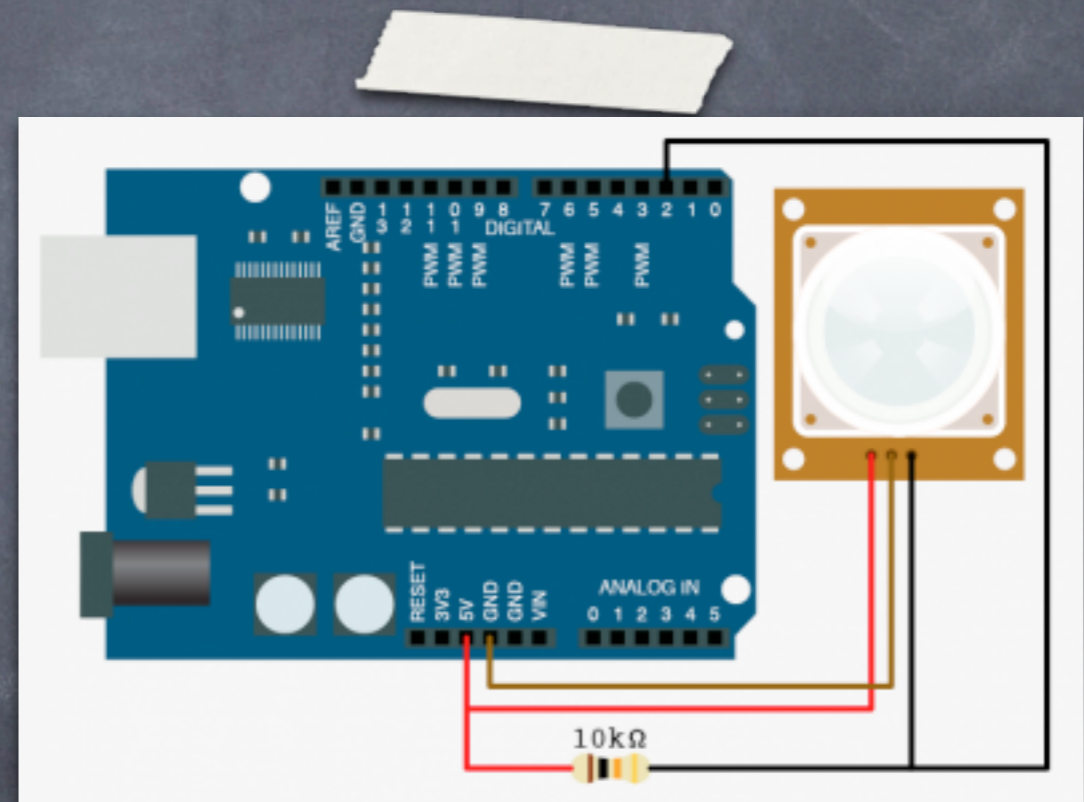
[http://bildr.org/2011/06/pir\\_arduino/](http://bildr.org/2011/06/pir_arduino/)



# PIR MOTION SENSOR

## Tutorial

- Aside from power, the signal pin connects to any digital pin on your Arduino but also needs a 10k pullup resistor between the signal and 5V.



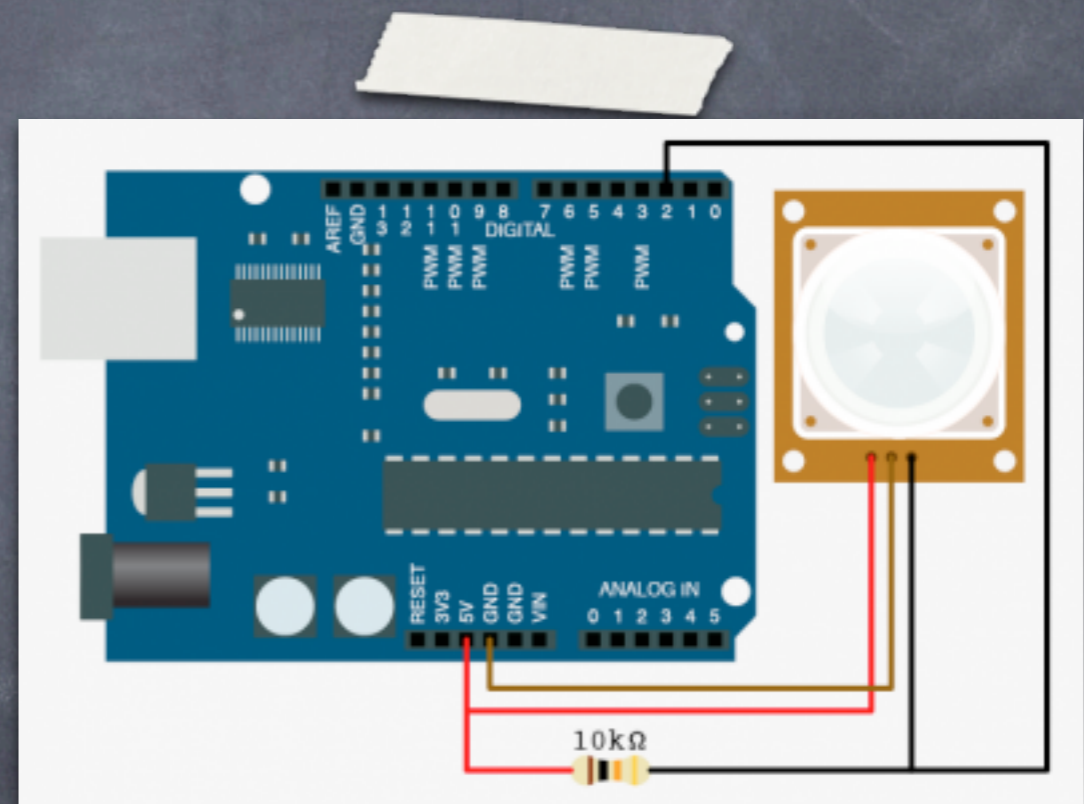
[http://bildr.org/2011/06/pir\\_arduino/](http://bildr.org/2011/06/pir_arduino/)



# PIR MOTION SENSOR

## Tutorial

- The reason for the **resistor** is that the signal pin is something called an **open-collector**, meaning it is as if it was not connected to anything at all when nothing is signaled. When motion is sensed, it connects the pin to **ground**. So we use the **resistor** so when we read the value from that pin when no motion is detected it looks HIGH. Without the resistor the pin would be floating around and the read value would randomly float between HIGH and LOW, and we don't want that.



[http://bildr.org/2011/06/pir\\_arduino/](http://bildr.org/2011/06/pir_arduino/)



# PIR MOTION SENSOR

- The code for this is pretty simple. When it starts up, it needs 2 seconds to take an image to compare to. When we see the signal pin go low, we print some text to the serial terminal (you can replace that with any code you want) and wait 2 seconds again before checking.
- You could make the code a little more sophisticated to do away with the 2sec delay after motion is sensed, but I found that the signal pin triggered on and off for a few seconds after it first sensed motion, so that delay is just to take care of that.



# PIR MOTION SENSOR

```
int pirPin = 2; //digital 2
```

```
void setup(){  
  Serial.begin(9600);  
  pinMode(pirPin, INPUT);  
}
```

```
void loop(){  
  int pirVal = digitalRead(pirPin);  
  
  if(pirVal == LOW){ //was motion  
    detected  
    Serial.println("Motion Detected");  
    delay(2000);  
  }  
}
```



Thank You!