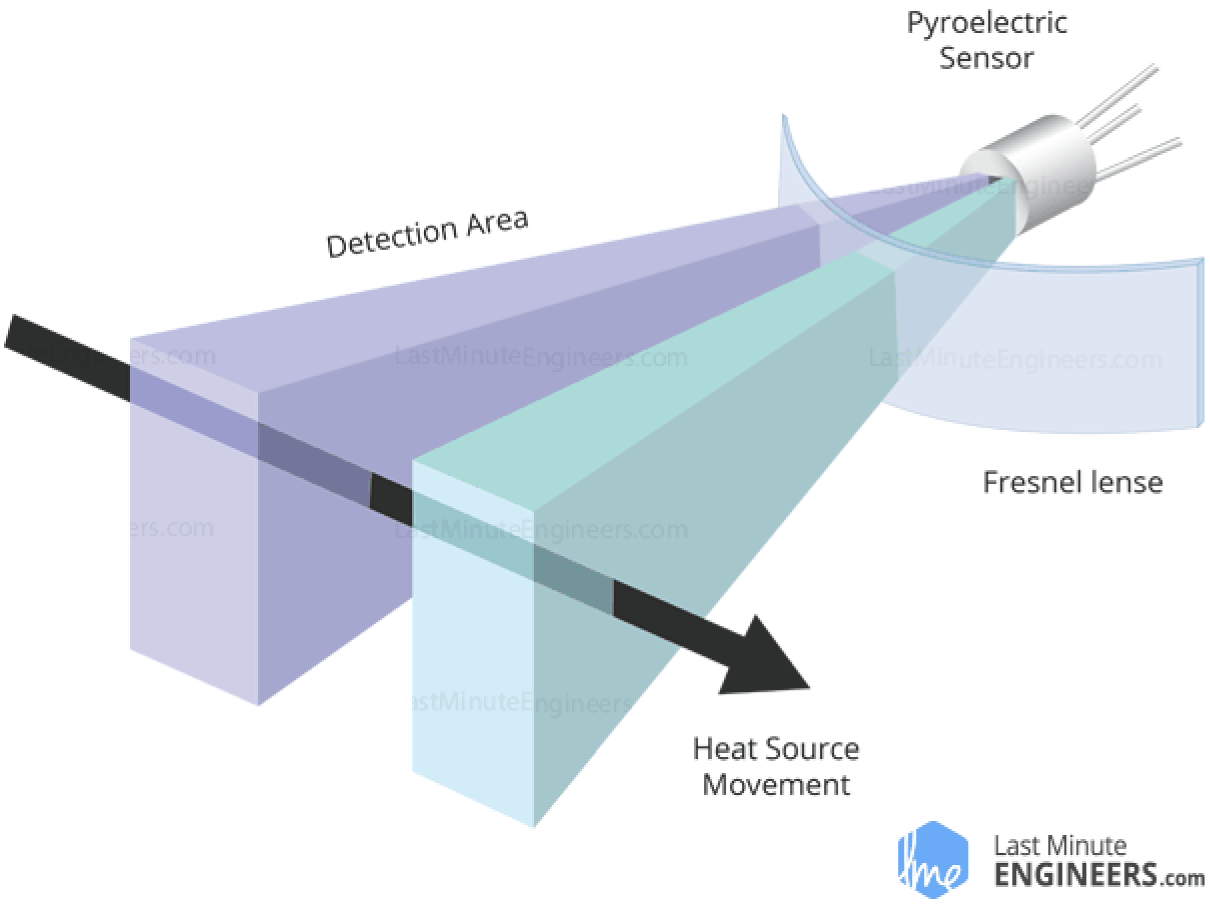
PROJECT 16: PIR SENSOR

How PIR Motion Sensor Works?

If you didn’t know, all objects with a temperature above Absolute Zero (0 Kelvin / -273.15 °C) emit heat energy in the form of infrared radiation, including human bodies. The hotter an object is, the more radiation it emits.

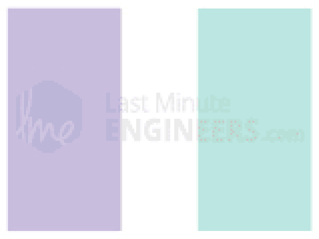
PIR sensor is specially designed to detect such levels of infrared radiation. It basically consists of two main parts: A Pyroelectric Sensor and A special lens called Fresnel lens which focuses the infrared signals onto the pyroelectric sensor.

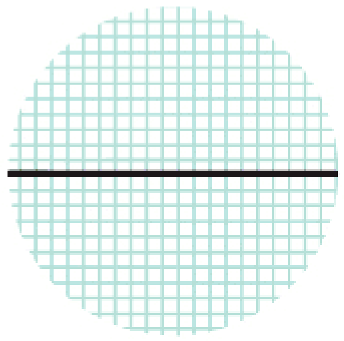


A ***Pyroelectric Sensor*** has two rectangular slots in it made of a material that allows the infrared radiation to pass. Behind these, are two separate infrared sensor electrodes, one responsible for producing a positive output and the other a negative output. The reason for that is that we are looking for a change in IR levels and not ambient IR levels. The two electrodes 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.

When the sensor is idle, i.e. 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 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. The corresponding pulse of signals results in the sensor setting its output pin high.



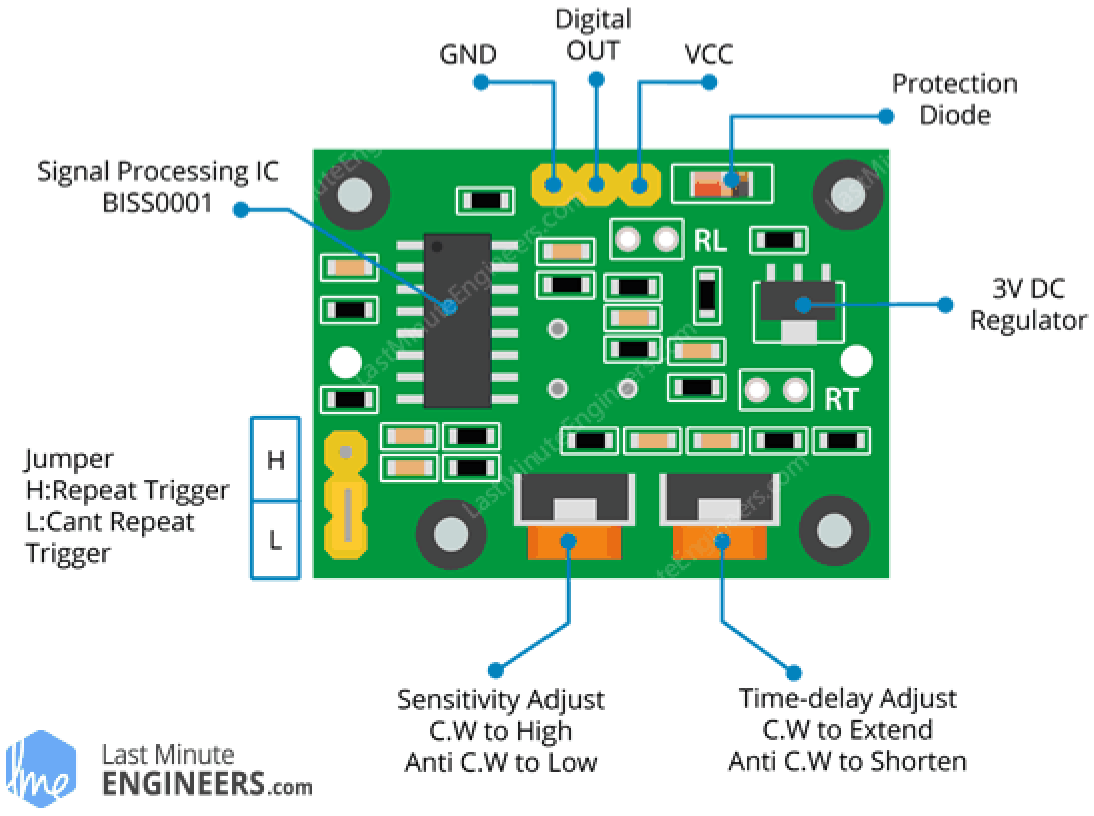


https://lastminuteengineers.com/pir-sensor-arduino-tutorial/

HC-SR501 PIR Motion Detector

For most of our Arduino projects that need to detect when a person has left or entered the area, or has approached, HC-SR501 PIR sensors are a great choice. They are low power and low cost, pretty rugged, have a wide lens range, easy to interface with and are insanely popular among hobbyists.

HC-SR501 PIR sensor has three output pins VCC, Output and Ground as shown in the diagram below. It has a built-in voltage regulator so it can be powered by any DC voltage from 4.5 to 12 volts, typically 5V is used. Other than this, there are a couple options you have with your PIR. Let’s check them out.


Digital OUT
GND
VCC
Protection Diode
Signal Processing IC BISS0001
RL
3V DC Regulator
RT
Jumper H:Repeat Trigger L:Cant Repeat Trigger
H
L
Sensitivity Adjust C.W to High Anti C.W to Low
Time-delay Adjust C.W to Extend Anti C.W to Shorten
Last Minute ENGINEERS.com

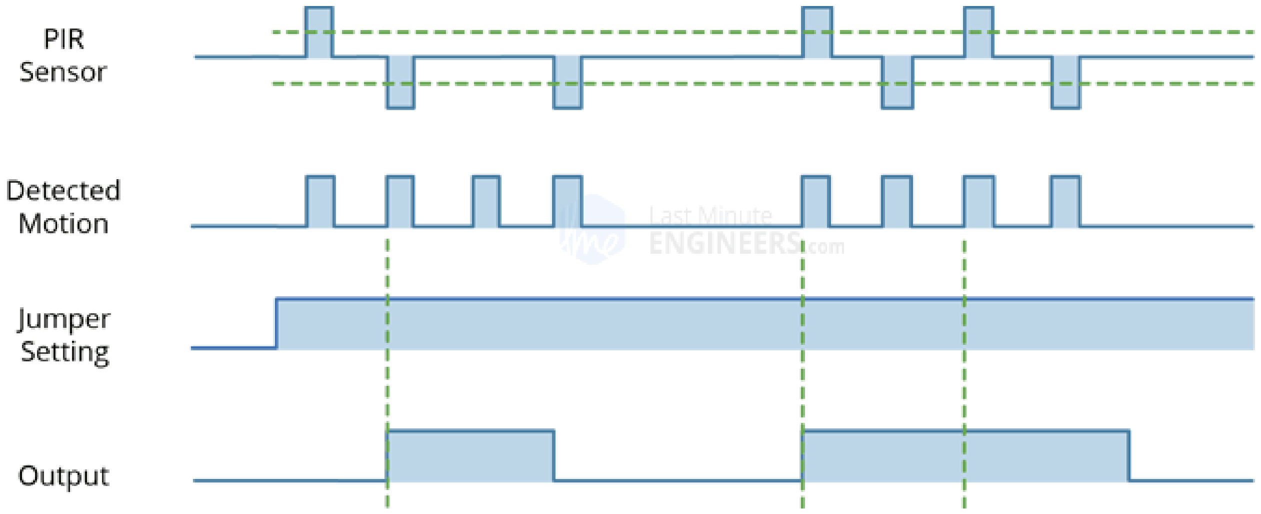
There are two potentiometers on the board to adjust a couple of parameters:

- Sensitivity– This sets the maximum distance that motion can be detected. It ranges from 3 meters to approximately 7 meters. The topology of your room can affect the actual range you achieve.

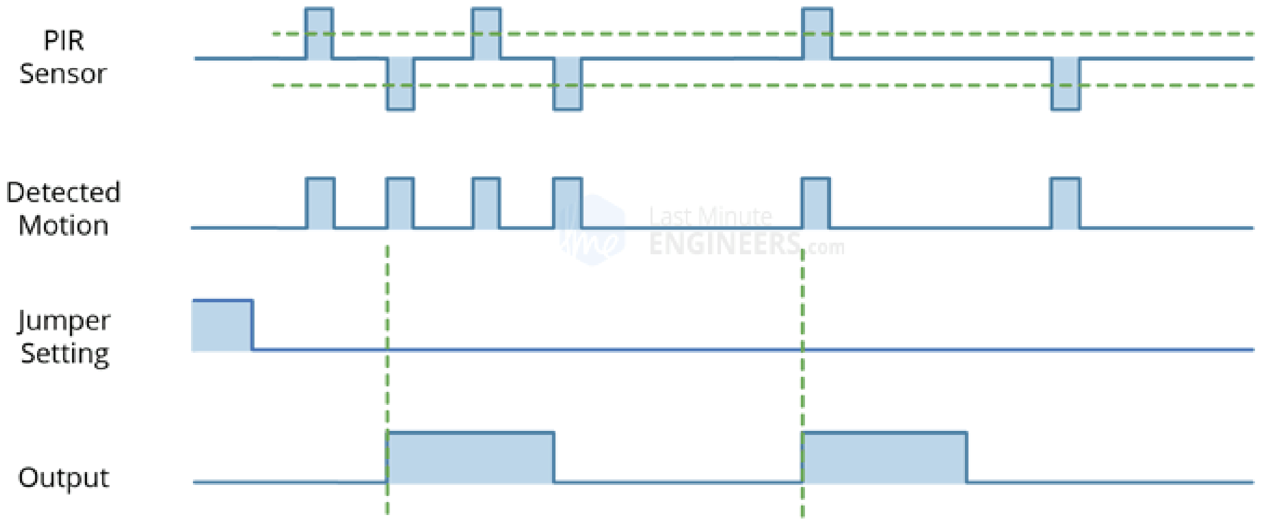
- Time– This sets how long that the output will remain HIGH after detection. At minimum it is 3 seconds, at maximum it is 300 seconds or 5 minutes.

Finally the board has a jumper (on some models the jumper is not soldered in). It has two settings:

- H– This is the Hold/Repeat/Retriggering In this position the HC-SR501 will continue to output a HIGH signal as long as it continues to detect movement.

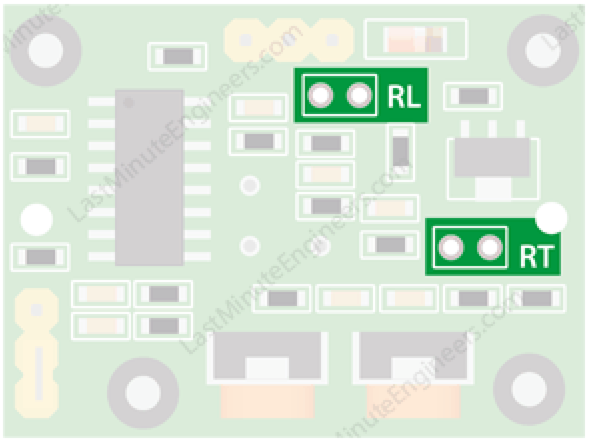


L– This is the Intermittent or No-Repeat/Non-Retriggering In this position the output will stay HIGH for the period set by the TIME potentiometer adjustment.



Making HC-SR501 PIR Sensor more versatile

The HC-SR501 circuit board has solder pads for two additional components. These are usually labeled as ‘RT’ and ‘RL’. Note that on some boards the labels may be covered by the “dome” lens on the side opposite the components.

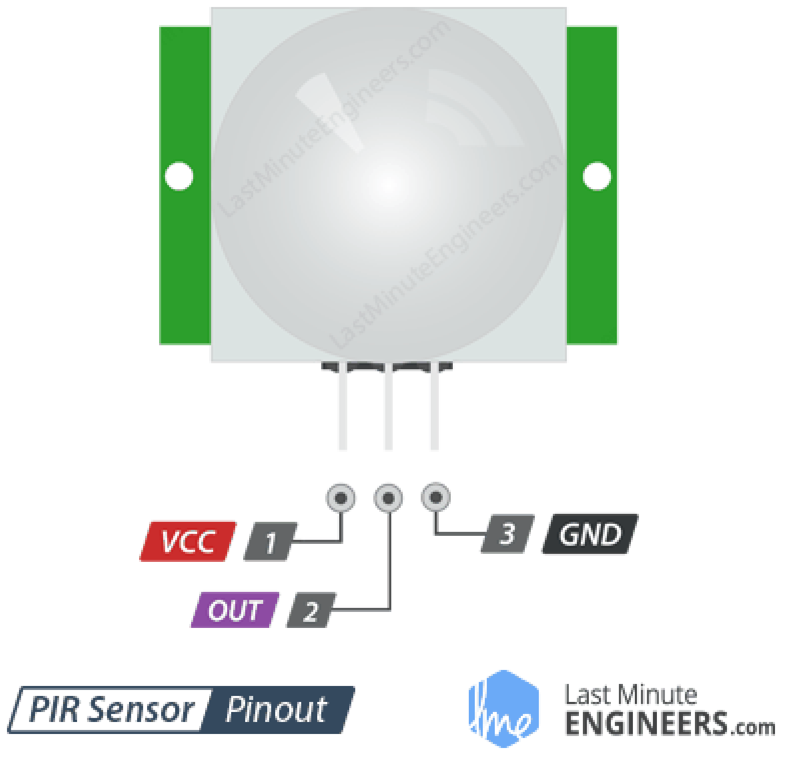


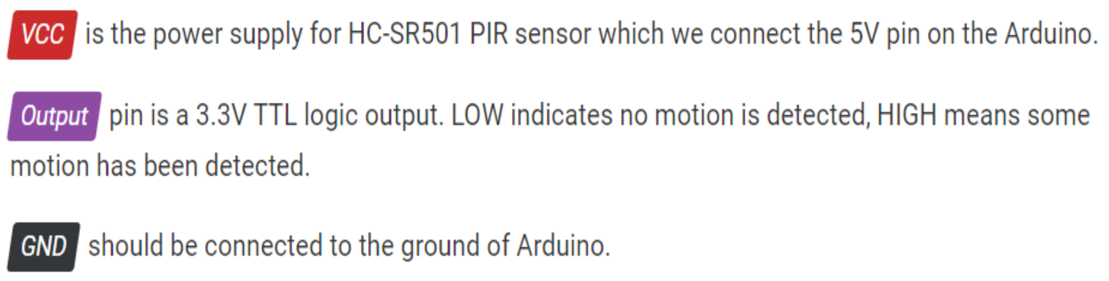
- RT– This is meant for a thermistor or temperature-sensitive resistor. Adding this allows the HC-SR501 to be used in extreme temperatures, it also increases the accuracy of the detector to some degree.
- RL– This connection is for a Light Dependant Resistor (LDR) or Photoresistor. By adding this component the HC-SR501 will only operate in darkness, a common application for motion-sensitive lighting systems.

The additional components can be soldered directly to the board or extended to remote locations using wires and connectors.

HC-SR501 PIR Sensor Pinout

The HC-SR501 has a 3-pin connector that interfaces it to the outside world. The connections are as follows:




VCC is the power supply for HC-SR501 PIR sensor which we connect the 5V pin on the Arduino.

Output pin is a 3.3V TTL logic output. LOW indicates no motion is detected, HIGH means some motion has been detected.

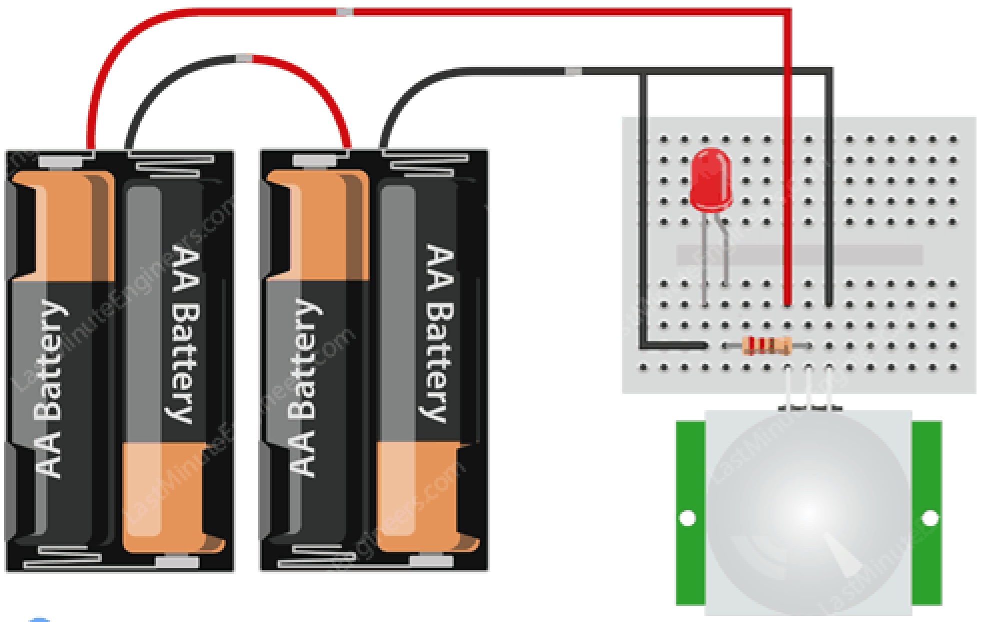
GND should be connected to the ground of Arduino.

Using PIR Sensor as a standalone unit

One of the reasons of HC-SR501 PIR sensor being extremely popular is the fact that HC-SR501 is a very versatile sensor that is pretty capable all on its own. And by interfacing it to some microcontrollers like an Arduino you can expand upon its versatility even further. For our first experiment we will use the HC-SR501 on its own to illustrate how useful it is by itself.

The wiring for this experiment is very simple. Batteries are connected across VCC and GND of the sensor and a small Red LED is connected to the output pin through a 220Ω current limiting resistor.

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



Last Minute ENGINEERS.com

Remember once you power up the circuit you need to wait 30-60 seconds for the PIR to acclimatize to the infrared energy in the room. 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!

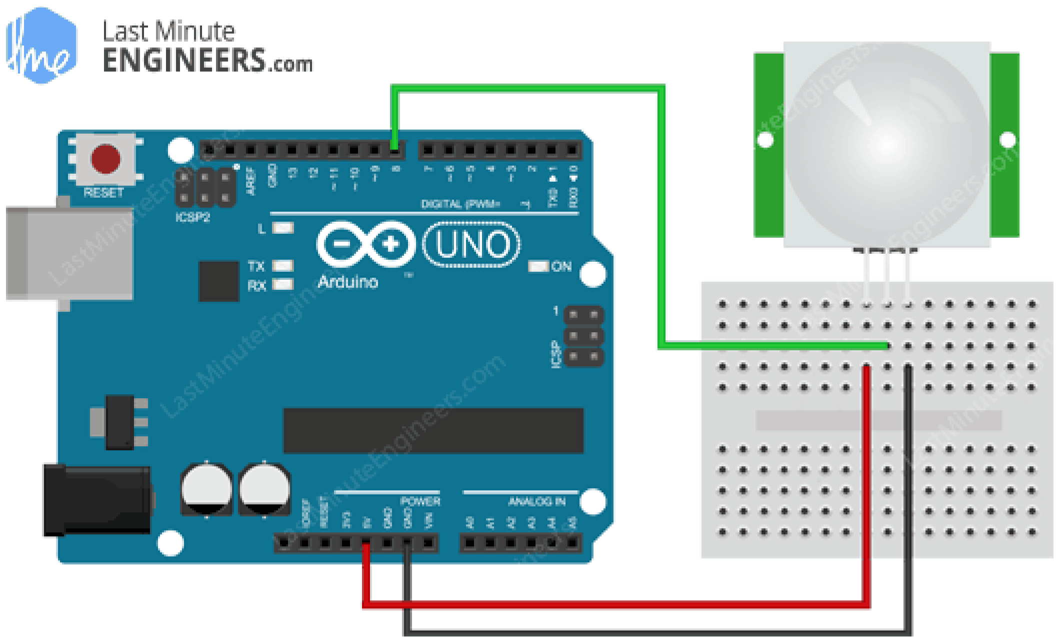
Wiring – Connecting PIR Sensor to Arduino UNO

Now that we have a complete understanding of how PIR sensor works, we can begin hooking it up to our Arduino.

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 output pin to flip HIGH (Motion Detected) or LOW (Not Detected). Power the PIR with 5V and connect ground to ground. Then connect the output to a digital pin #2.

You will want to set the jumper on the HC-SR501 to the H (Retriggering) position for this to work correctly. You'll also need to set the TIME to the minimum of 3 seconds, turn the TIME potentiometer as far counterclockwise as it will go. Set the sensitivity anywhere you like, set it to midpoint if you are not sure.

With that, you're now ready to upload some code and get the PIR working.



Arduino Code

The code is very simple, and is basically just keeps track of whether the input to pin#2 is HIGH or LOW.

```
//project16.ino
//Full Name:"Creator"
//COURSE SECTION:
```

```
int ledPin = 13;                // choose the pin for the LED
int inputPin = 8;               // 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

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)
            {
      Serial.println("Motion detected!");          // print on output change
      pirState = HIGH;
    }
  }
  else
  {
    digitalWrite(ledPin, LOW); // turn LED OFF

    if (pirState == HIGH)
            {
      Serial.println("Motion ended!");  // print on output change
      pirState = LOW;
    }
  }
}
```