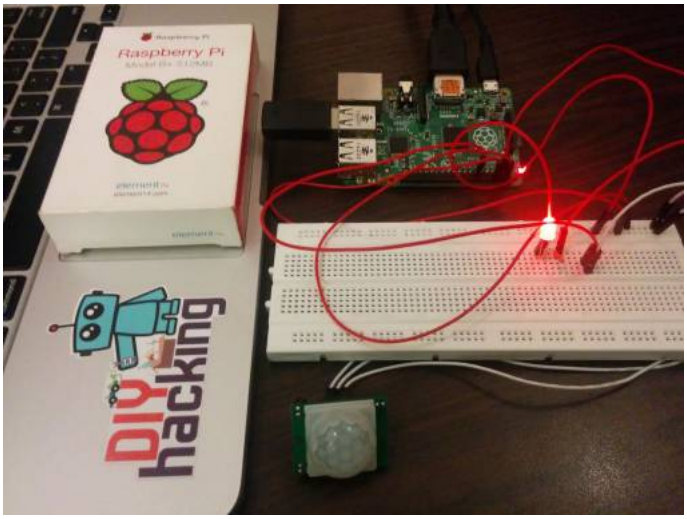
How to Interface a PIR Motion Sensor With Raspberry Pi GPIO



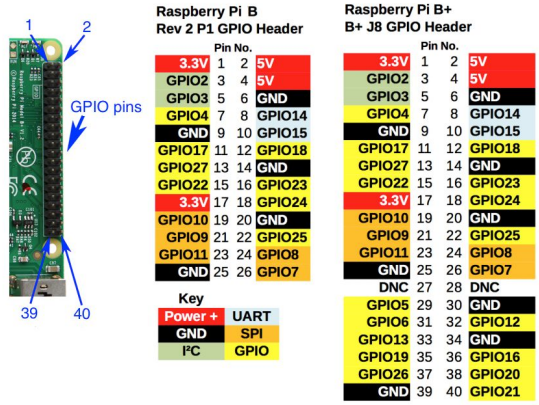
This tutorial will show you to interface a PIR motion sensor with the Raspberry Pi and how to use the GPIO pins on it. The GPIO pins on the Raspberry Pi are critical when it comes to making a hardware project. May it be a robot, home automation system, or whatever crazy creation you come up with. In all these cases you will have to use the GPIO (General Purpose Input/Output) pins on the Raspberry Pi. With this simple tutorial, you will be able to learn how to control the output on the GPIO pins. and read inputs through them. Moreover, you will get to read the output from a PIR motion sensor and write a simple code to blink an LED. If you're not familiar with the Rasbperry Pi terminal, check out this tutorial on Basic Linux Commands. If you are a true beginner, you can always use our free eBook on Raspberry Pi and Arduino to get started from step 0. So gear up and get ready to have some fun with the Raspberry Pi GPIOs!

Required Materials

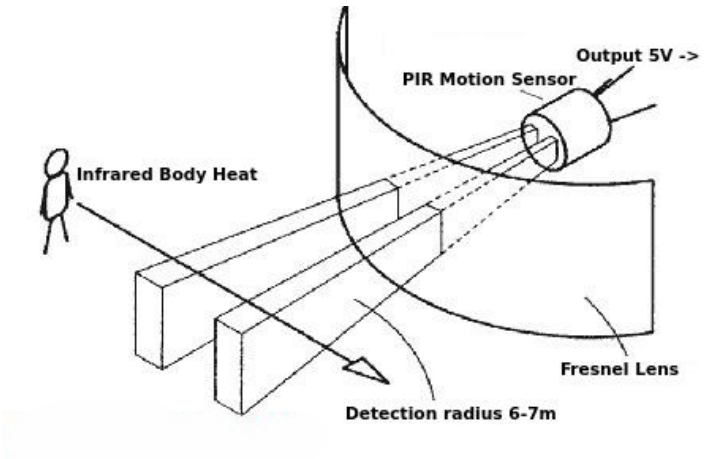
1. Raspberry Pi B/B+ or 2 and basic peripherals: SD card, keyboard, mouse, etc. (This should work on Raspberry Pi 3 as well)
2. An LED and 220Ohm resistor.
3. PIR motion sensor.
4. Breadboard.
5. Male to male and Female to Male jumper wires.

How Does it Work?

The Raspberry Pi GPIO can be accessed through a python program. You will get to know how to access these pins and the commands required to do that later in this tutorial. Each pin on the Raspberry Pi is named based on its order (1,2,3,..) as shown in the diagram below:



Here, we are using a PIR motion sensor. PIR stands for Passive InfraRed. This motion sensor consists of a fresnel lens, an infrared detector, and supporting detection circuitry. The lens on the sensor focuses any infrared radiation present around it toward the infrared detector. Our bodies generate infrared heat, and as a result, this heat is picked up by the motion sensor. The sensor outputs a 5V signal for a period of one minute as soon as it detects the presence of a person. It offers a tentative range of detection of about 6-7 meters and is highly sensitive. When the PIR motion sensor detects a person, it outputs a 5V signal to the Raspberry Pi through its GPIO and we define what the Raspberry Pi should do as it detects an intruder through the python coding. Here we are just printing: “Intruder detected”.



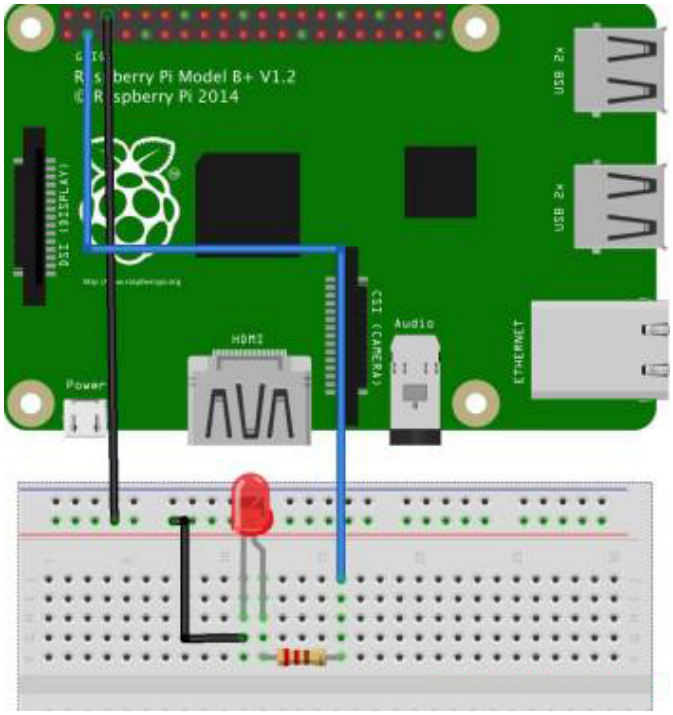
How the PIR Motion Sensor works

Blinking an LED Using the Raspberry Pi GPIO- Output GPIO Control

After you have set up your Raspberry Pi, we can now start messing around with its GPIO pins. Here, we will try to blink an LED using a python script. Copy and paste the following code into your Raspberry Pi. You can do this by opening the text editor: “leafpad” on your Raspberry Pi and copying this code into it and save this as a python file: **ledblink.py** :

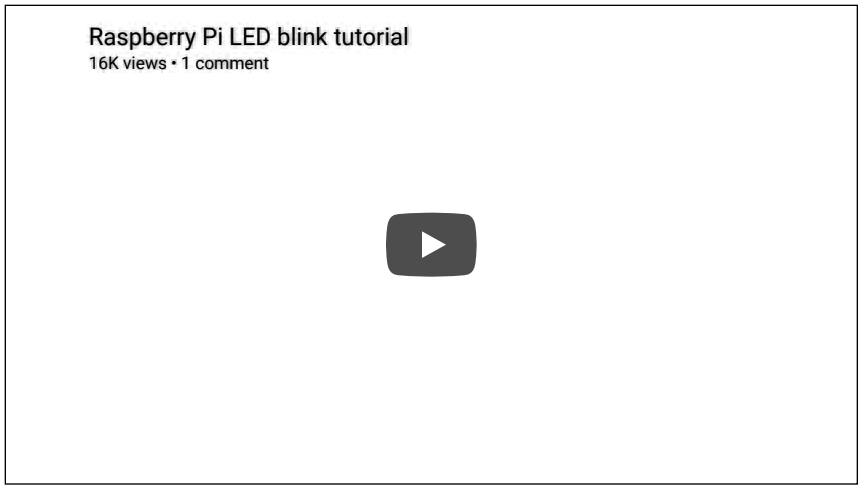
```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(3,GPIO.OUT)     #Define pin 3 as an output pin

while True:
        GPIO.output(3,1)   #Outputs digital HIGH signal (5V) on pin 3
        time.sleep(1)      #Time delay of 1 second

        GPIO.output(3,0)   #Outputs digital LOW signal (0V) on pin 3
        time.sleep(1)      #Time delay of 1 second
```

Next, we need to connect the LED to pin 3 on the Raspberry Pi GPIO. You can check out the connection diagram below to do that:



Raspberry Pi GPIO LED connection diagram

You should notice that the LED starts blinking after you execute the python program. Using this command: **sudo python ledblink.py**. The LED blinks because it receives a HIGH (5V) signal and a LOW (0V) signal from the Raspberry Pi GPIO at a delay of one second. You can check out the video below for a demo:

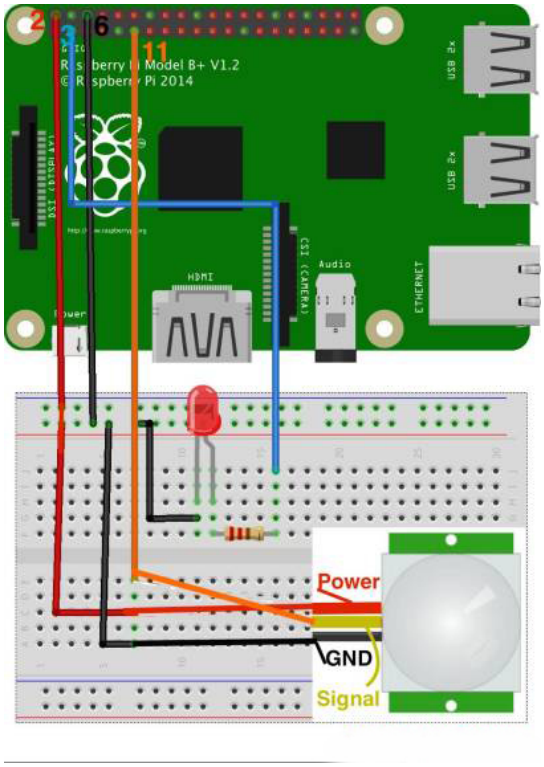


Interfacing the PIR Motion Sensor to the Raspberry Pi's Input GPIO Read

Now, we can try reading the output from the PIR motion sensor. The sensor outputs a digital HIGH (5V) signal when it detects a person. Copy and paste the following code into your Raspberry Pi and save it as a python file: **pirtest.py**:

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.IN)         #Read output from PIR motion sensor
GPIO.setup(3, GPIO.OUT)         #LED output pin
while True:
       i=GPIO.input(11)
       if i==0:                 #When output from motion sensor is LOW
             print "No intruders",i
             GPIO.output(3, 0)  #Turn OFF LED
             time.sleep(0.1)
       elif i==1:               #When output from motion sensor is HIGH
             print "Intruder detected",i
             GPIO.output(3, 1)  #Turn ON LED
             time.sleep(0.1)
```

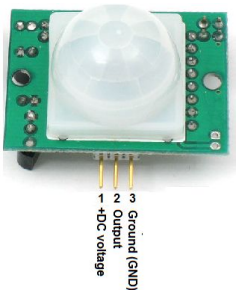
Now, connect your Raspberry Pi GPIO to the PIR motion sensor as per the connection diagram below:



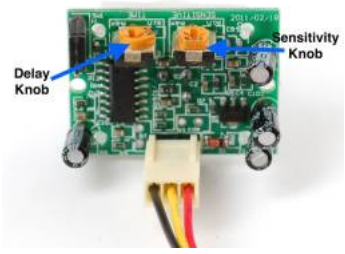
Raspberry Pi PIR motion sensor connection

You will notice that this code prints: “Intruder detected” when you place your hand over the sensor. After removing your hand and waiting some time, it prints: “No intruders”.

In certain PIR motion sensors, you can even adjust the delay at which the sensor outputs a HIGH signal at the expense of compromising the accuracy. You just need to turn the two knobs on the sensor counterclockwise using a screwdriver.



PIR Motion Sensor pin out



PIR motion sensor adjustment knobs

You can also extend the display of your laptop to the Raspberry Pi via a VNC server and a LAN cable just like how I did in the below video.

You can see this sensor in action, in the video below: