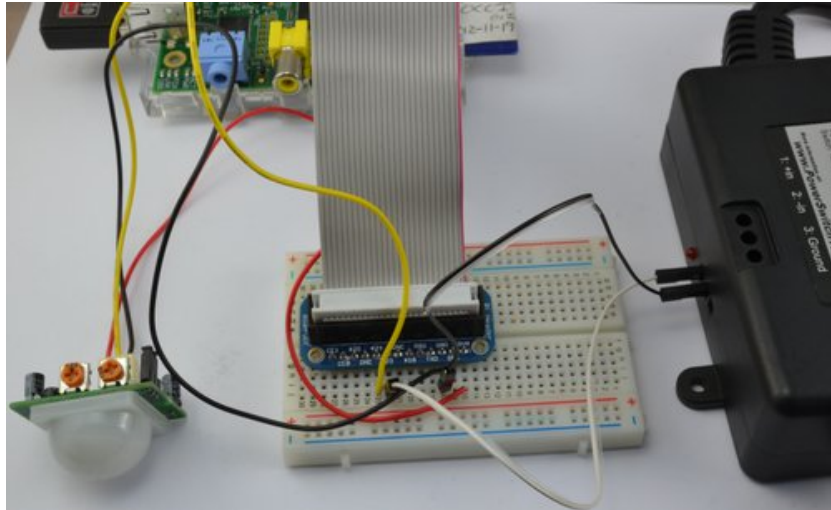




## Adafruit's Raspberry Pi Lesson 13. Power Control

Created by Simon Monk



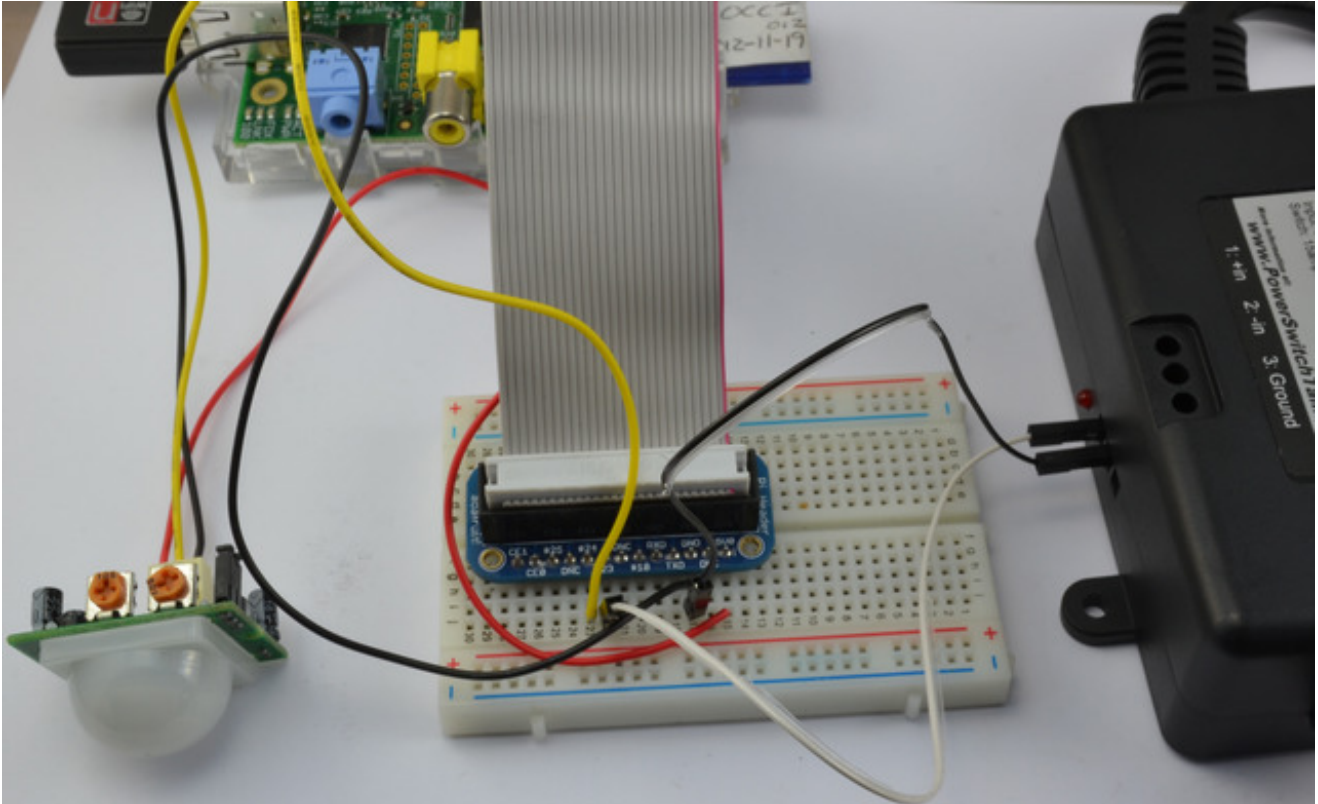
Last updated on 2016-06-23 10:53:54 PM EDT

## Guide Contents

|                    |   |
|--------------------|---|
| Guide Contents     | 2 |
| Overview           | 3 |
| Parts              | 4 |
| Hardware           | 5 |
| Software           | 7 |
| Configure and Test | 8 |

# Overview

The Raspberry Pi can make a great home automation controller. But to be really useful, it needs to be able to turn electrical appliances and lights on and off. That means it needs to be able to control 110V safely.



In this lesson, you will combine the PIR sensor from [Lesson 12 \(http://adafru.it/aZ3\)](http://adafru.it/aZ3) with the Powerswitch Tail 2 module from Adafruit, to automatically switch something on when movement is detected.

# Parts

To build the project described in this lesson, you will need the following parts.

PIR Sensor

Power Switch Tail 2

Pi Cobbler

Half-size Breadboard

Raspberry Pi

Jumper wire pack

# Hardware

The Powerswitch Tail 2 from Adafruit, looks like a 110V electrical lead with a box in the middle.

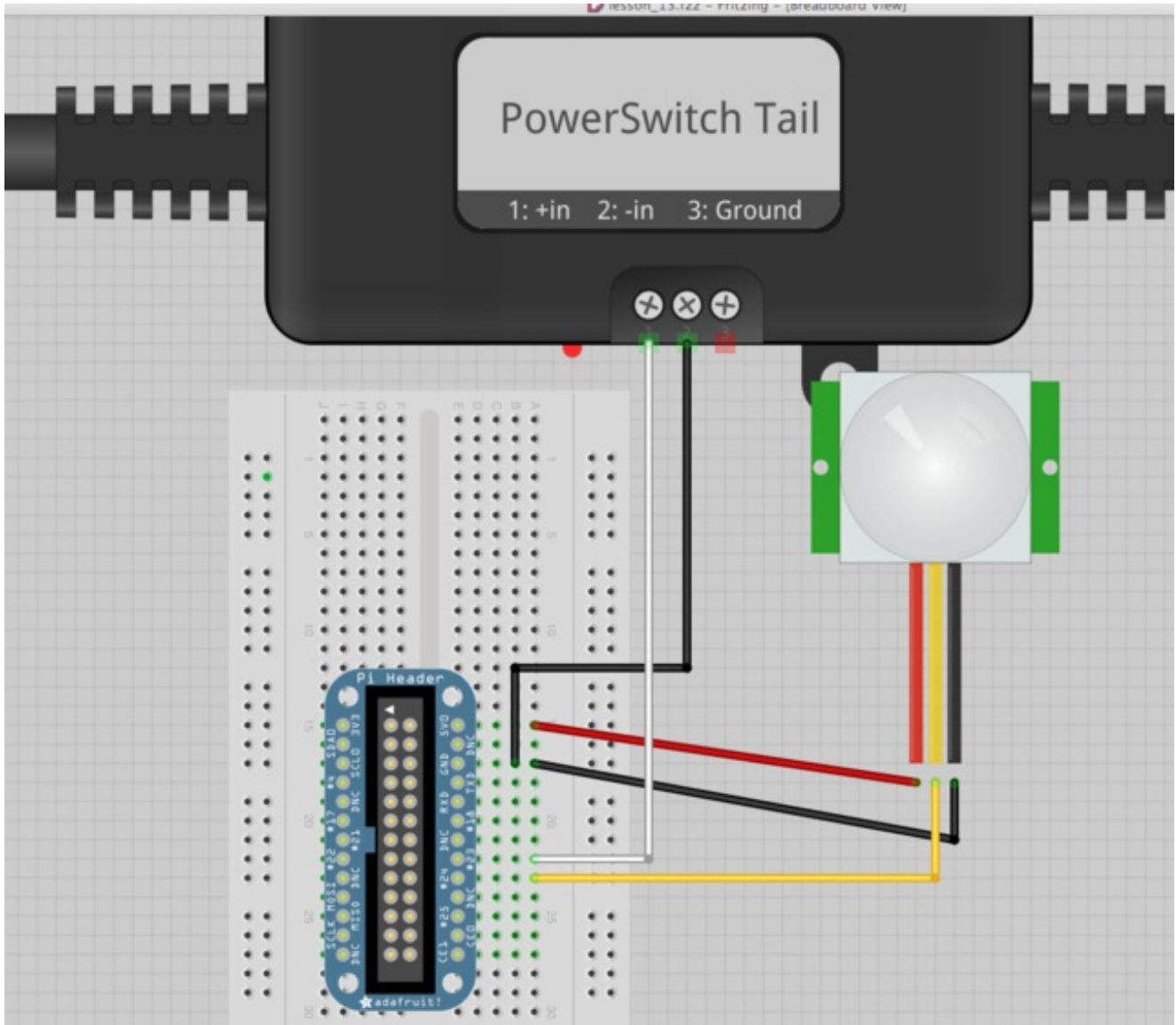
If you have 220V mains, this tutorial wont work for you - If we knew of an equivalent 220V controller like the PST we would have a link to it here



The box is actually an opto-isolated solid-state relay. The opto-ilsolation bit means that there is no actual electrical connection between the low voltage switching circuit and the 100V mains. This makes it extremely safe and removes any chance of accidentally frying your Pi.

What is more the control input consumes just 3 mA of current at 3.3V, which means we can control it directly from one of the Pi's output pins.

The PIR sensor is connected to a different pin from lesson 12, so be sure to move the yellow lead.



The Powerswitch Tail comes with a little indicator LED that shows when it is on, so you do not need to connect anything high voltage to it yet.

# Software

This project is possible the most over-engineered automatic light switch ever created. You really do not need a Raspberry Pi to turn on the Powerswitch, but the example can easily be adapted for other purposes. For example, you could use a combination of temperature, humidity, light, and perhaps some internet data on weather forecasts to turn on or off a heater, fan or humidifier.

```
import time
import RPi.GPIO as io
io.setmode(io.BCM)

pir_pin = 24
power_pin = 23

io.setup(pir_pin, io.IN)
io.setup(power_pin, io.OUT)
io.output(power_pin, False)

while True:
    if io.input(pir_pin):
        print('POWER ON!')
        io.output(power_pin, True)
        time.sleep(20)
        print('POWER OFF!')
        io.output(power_pin, False)
        time.sleep(5)
    time.sleep(1)
```

The program first sets up the two GPIO pins that are used, one as an input to the PIR sensor and one as an output for the Powerswitch Tail.

The main loop then just waits for the PIR sensor to sense movement and then prints a message and turns on the Powerswitch tail, waits 20 seconds then turns it off again.

If the output was turned on, it waits for 5 seconds to prevent immediate retriggering.

There is also always a 1 seconds delay each time around the loop.

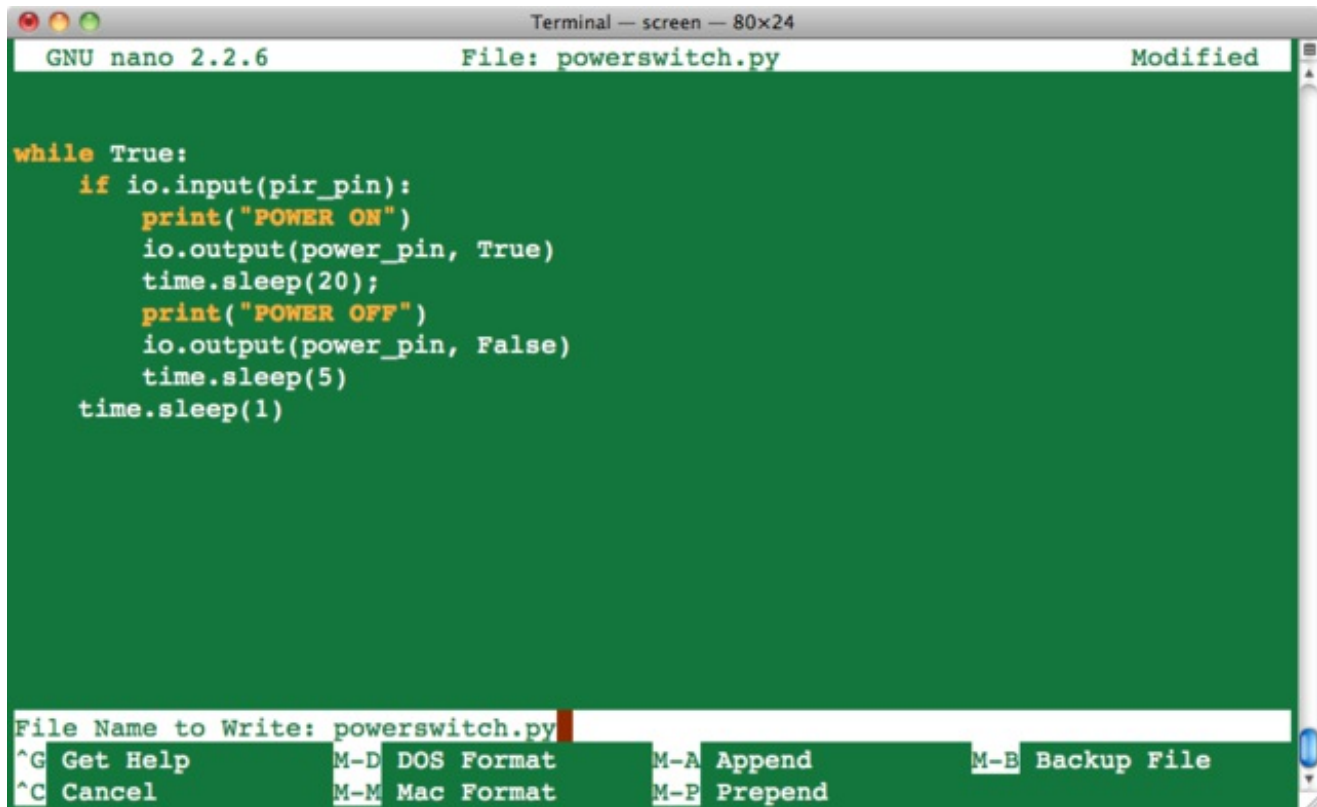
PIR sensors and the Pi 3 sometimes don't get along - if you're having false trigger reports, make sure the PIR Sensor is far away from the Pi 3

# Configure and Test

There are lots of ways to get the sketch from the listing below onto your Raspberry Pi. Perhaps the easiest is to connect to your Pi using SSH (See [Lesson 6 \(http://adafru.it/aWc\)](http://adafru.it/aWc)) opening an editor using the command below:

```
sudo nano powerswitch.py
```

and then pasting in the code , before saving the files using CTRL-x.



```

GNU nano 2.2.6      File: powerswitch.py      Modified
while True:
    if io.input(pir_pin):
        print("POWER ON")
        io.output(power_pin, True)
        time.sleep(20);
        print("POWER OFF")
        io.output(power_pin, False)
        time.sleep(5)
    time.sleep(1)

File Name to Write: powerswitch.py
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend

```

To start with, cover the PIR sensor with something, so that it is not activated until you are ready.

Run the program as superuser using the command:

```
sudo python powerswitch.py
```

Uncover the PIR sensor and you should see a message saying 'POWER ON' and the little LED on the Powerswitch Tail should light, then turn itself off after 20 seconds.

Now this is all working, you can try attaching a lamp or some electrical appliance to the Powerswitch Tail.

Do not write code that turns the power on and off very rapidly. Many kinds of appliance, including



lamps will be damaged by high frequency switching. This type of setup should turn things on for a few seconds minimum.