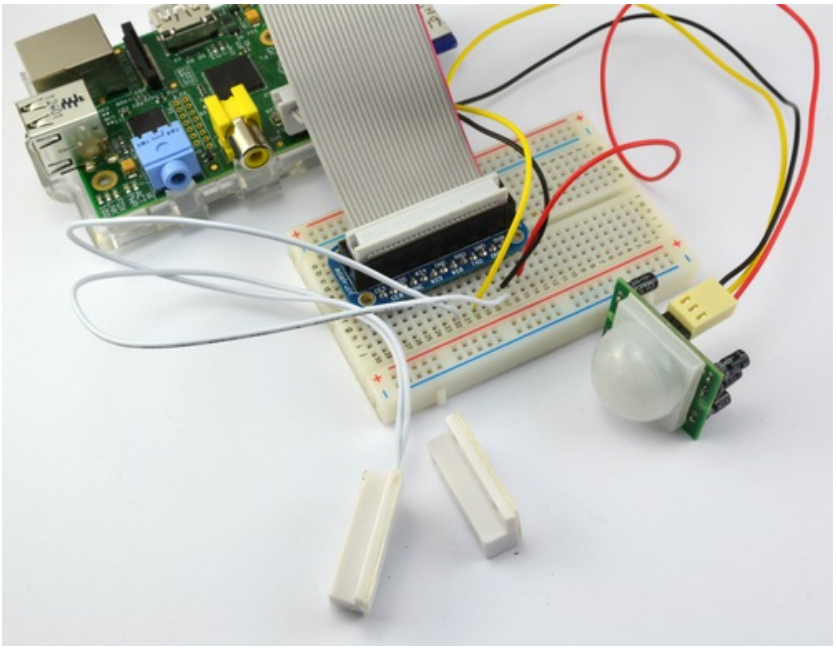
adafruit learning system

Adafruit's Raspberry Pi Lesson 12. Sensing Movement

Created by Simon Monk

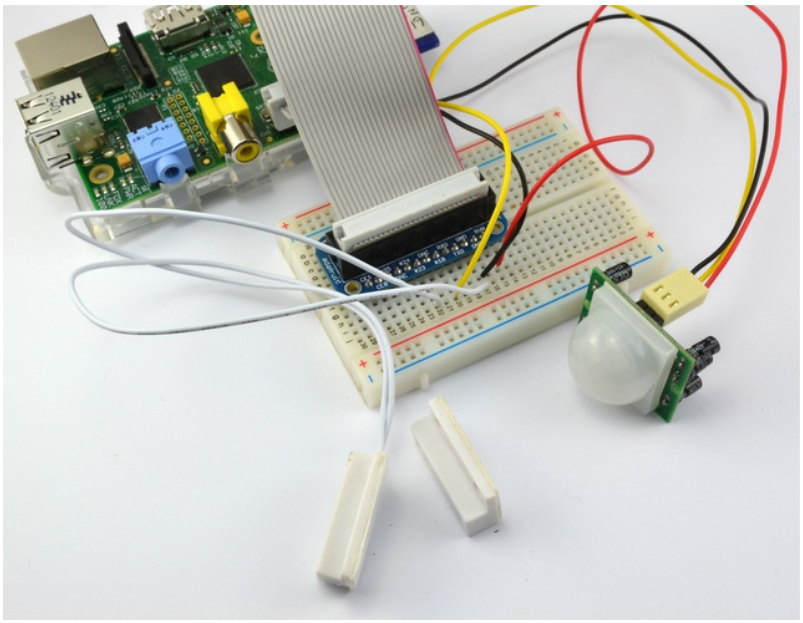


Last updated on 2013-09-11 02:00:54 PM EDT

Guide Contents

Overview

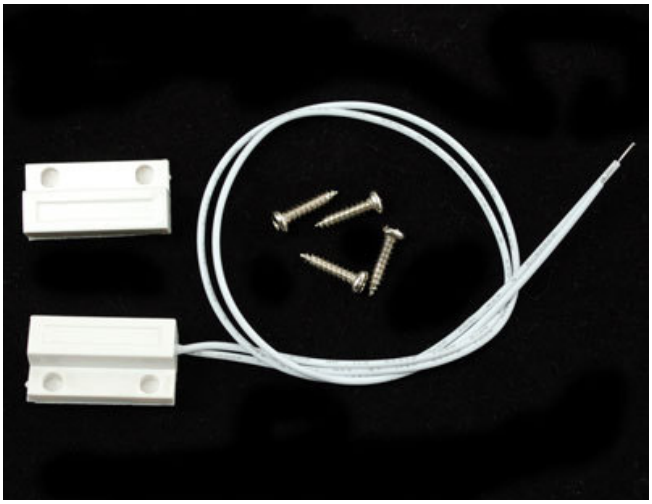
In this lesson, you will learn how to use the digital inputs on the GPIO connector with a door sensor and a PIR motion detector.

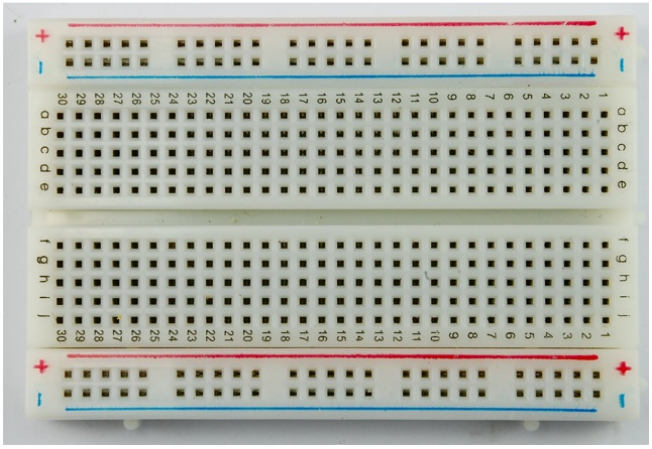
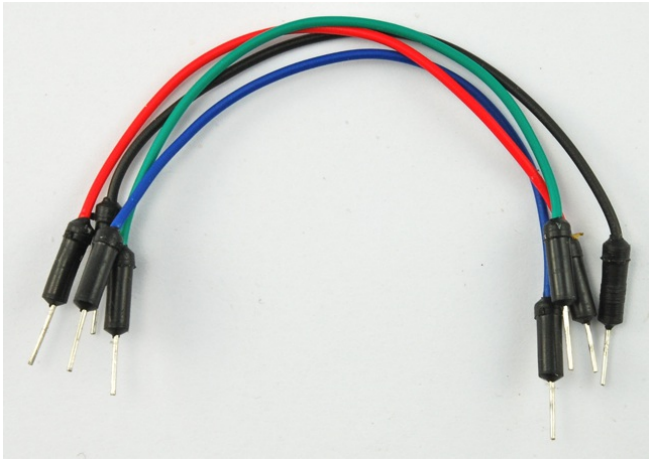
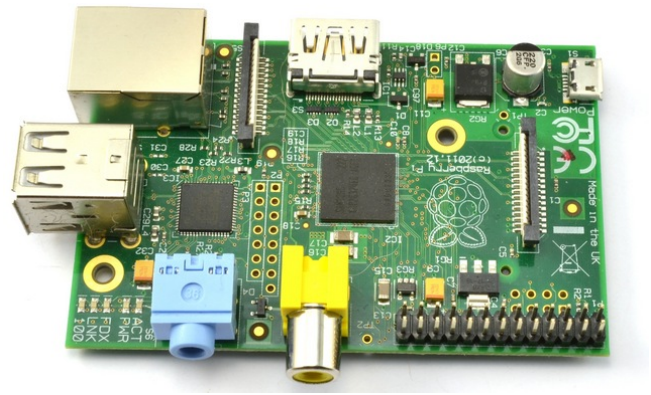


In this lesson, we will concentrate on sensing movement and activation of the door switch. In Lesson 13 we will build on this security sensing to have the Pi use a digital output to control the power to an electrical appliance when movement is detected.

Parts

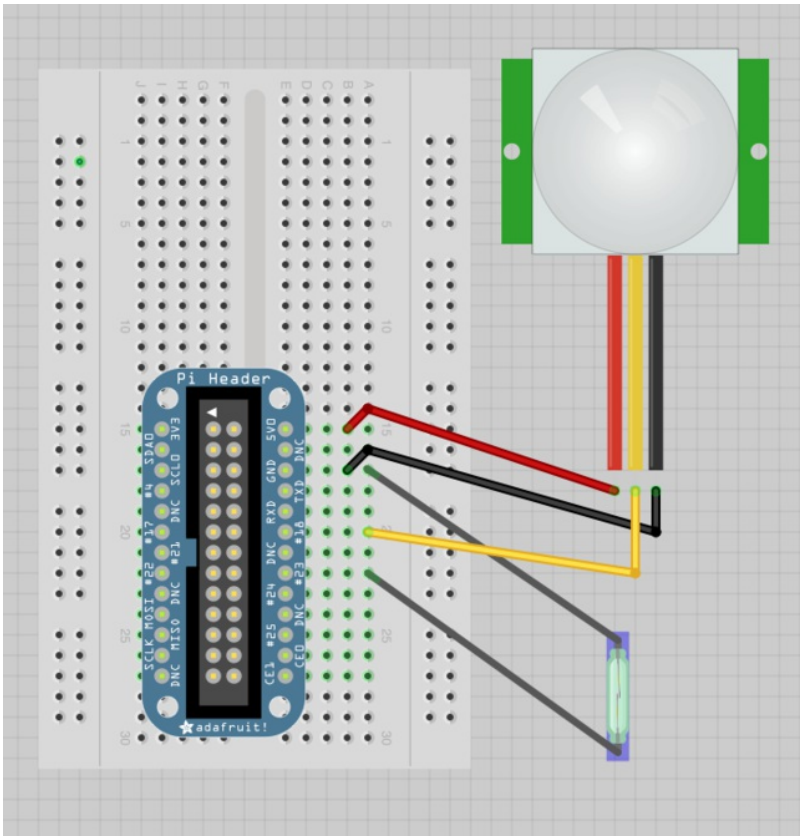
To build the project described in this lesson, you will need the following parts.

	PIR Sensor (Adafruit product 189)
	Magnetic Door Sensor (Adafruit product 375)
	Pi Cobbler

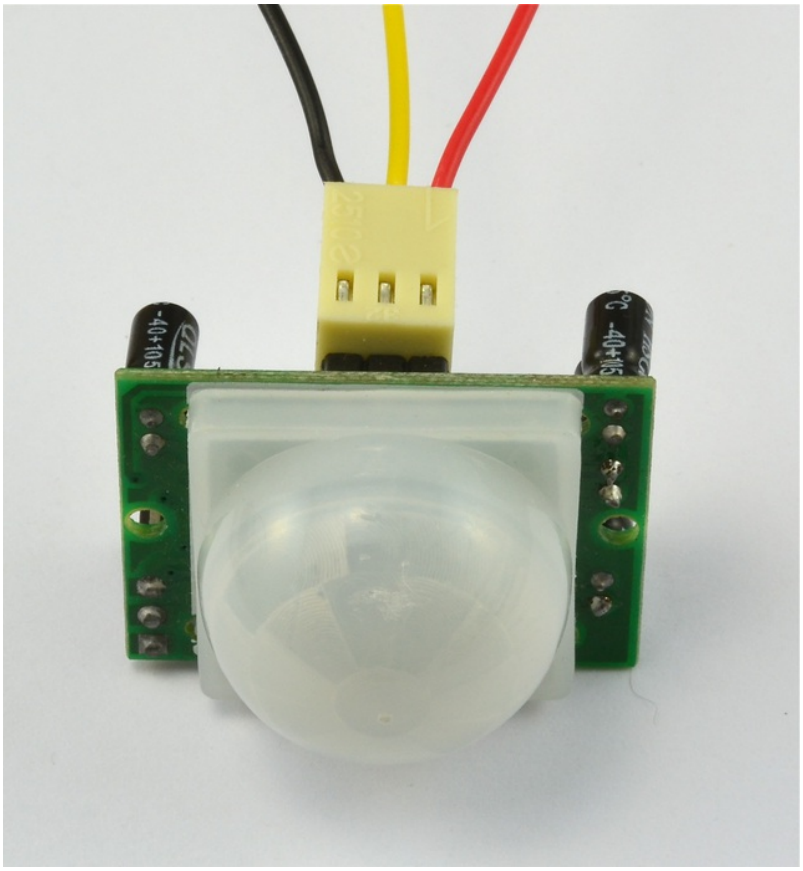
	Half-size Breadboard
	Jumper wire pack
	Raspberry Pi

Hardware

We are going to connect both sensors to the Raspberry Pi at the same time. Neither sensor requires any extra components.



The PIR sensor comes with a socket and lead. Make sure that the socket is the right way around (use the picture below) and that the red lead goes to 5V, the black to GND and the yellow to 18 on the Cobbler.



Although the PIR sensor requires a 5V supply, its output is a Pi-friendly 3.3V, so it can be connected directly to a GPIO input.

The Door Switch, uses what is called a reed switch. These are two contacts inside a glass tube, that is then encased in plastic. When a magnet (the other white block) is placed near the reed switch, the contacts are drawn together and the switch closes. Since this is just a switch, the leads can be connected either way around.

We will use the Pi's ability to create an internal pull-up resistor on the reed-switch pin, so we don't need an external pull-up resistor.

Software

The program for this project just loops round printing a message every time motion is detected, or the magnet is moved away from the door.

The program uses the Rpi.GPIO library. See Lesson 4 (http://adafru.it/aXR).

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

pir_pin = 18
door_pin = 23

io.setup(pir_pin, io.IN)         # activate input
io.setup(door_pin, io.IN, pull_up_down=io.PUD_UP)  # activate input with PullUp

while True:
    if io.input(pir_pin):
        print("PIR ALARM!")
    if io.input(door_pin):
        print("DOOR ALARM!")
    time.sleep(0.5)
```

The program sets the pir_pin to be just a plain old input. This is because the PIR sensor has a digital output of either 3.3V or 0V. By contrast, the door_pin, since it is a switch does not generate a voltage for a digital input. So, that input pin uses the extra argument (pull_up_down=io.PUD_UP). This activates an internal resistor that makes the input HIGH (pulled-up) unless something stronger (like a switch connecting it to GND) pulls it LOW.

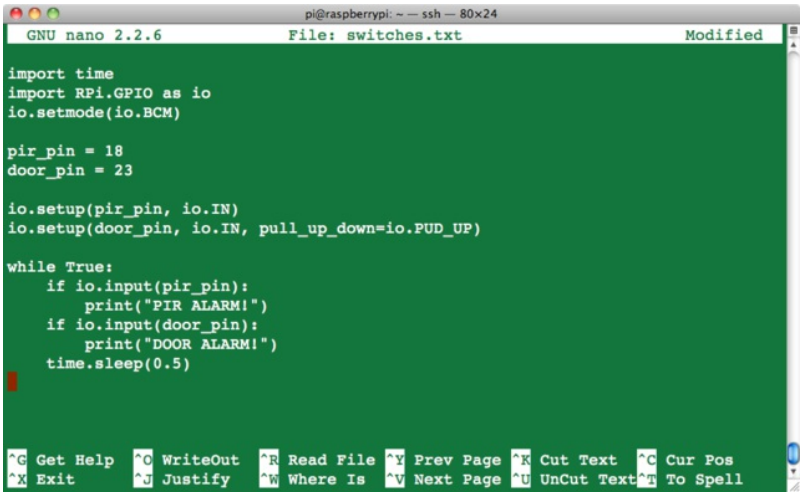
The loop then reads each of the inputs in turn and prints a message appropriately. Remember that the door switch warning will be activated when the magnet is removed from the sensor rather than the other way around.

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)) opening an editor using the command below:

```
nano switches.py
```

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



To start with, place the magnet next to the switch and cover the PIR sensor with something.

Run the program as superuser using the command:

```
sudo python switches.py
```

..and you should see some trace appear in the terminal, when you move the magnet, or take the cover off the PIR sensor.

A good exercise might be to place your kids in-front of the PIR sensor and see how long they can keep still!