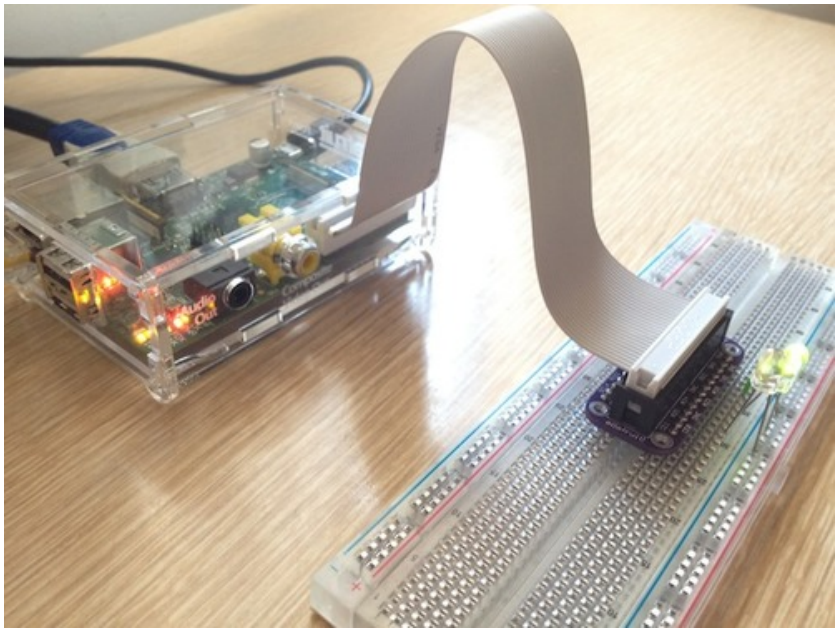
adafruit learning system

Raspberry Pi E-mail Notifier Using LEDs

Created by Mikey Sklar

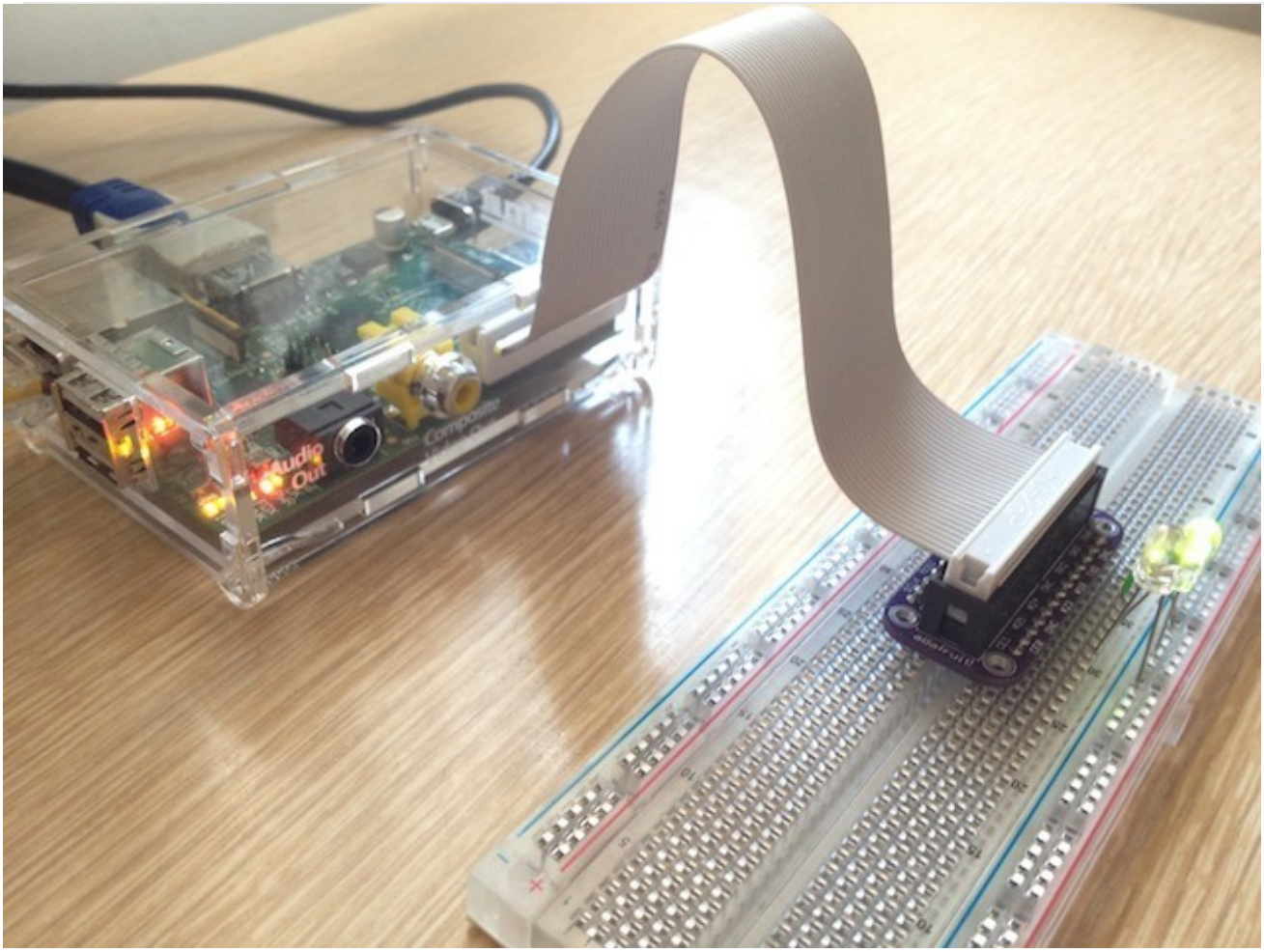


Last updated on 2015-04-09 03:48:49 PM EDT

Guide Contents

Guide Contents 2
Overview 3
Remote SSH 4
Prepare Python 5
Wire LEDs 8
Python Script 13

Overview



Raspberry Pi's popularity makes things so easy that it is almost scary. I set forth on a simple starter project of having the Raspberry Pi show me when new GMail messages arrive.

After some searching, it seems that lots of people are already talking about how to do this and there are some great examples. Michael over at MitchTech (http://adafru.it/aJG) had the most ready to go code which I pilfered from. Adafruit's Cobbler Breakout Kit (http://adafru.it/914) makes the bread board experience even easier with the clearly labeled pins for each of the raspi's GPIO pins.

These are the things I had to setup in order to get a working Raspberry Pi + GMail + Adafruit Cobbler. You can probably get yourself up and running with this same setup in less than 30 minutes!

Remote SSH

As a first step, you'll want a terminal on your Pi. There are a variety of options, explored in more detail in the Getting a Terminal on Your Raspberry Pi (http://adafru.it/eUN) section of our introduction to the command line (http://adafru.it/ek4):

1. Plug in an HDMI monitor and keyboard
2. Use a console cable (http://adafru.it/eUO)
3. Log in via **S**ecure **SH**ell (SSH) (http://adafru.it/dZL)

Since this project uses the GPIO pins for its own purposes, you'll want to choose between working directly on the Pi (this is fine if you're already set up!) or connecting via SSH (a good idea if you don't have an extra monitor / input devices laying around, and want to work from the comfort of your desktop or laptop).

Examples here will assume a working Raspbian installation with an SSH connection. Not sure how to get started?

For installating Raspbian, check out our guide to preparing an SD card for the Pi (http://adafru.it/evL). If you need help finding the Pi on your network and connecting to it, check out our Pi Finder utility.

Adafruit Pi Finder

http://adafru.it/enj

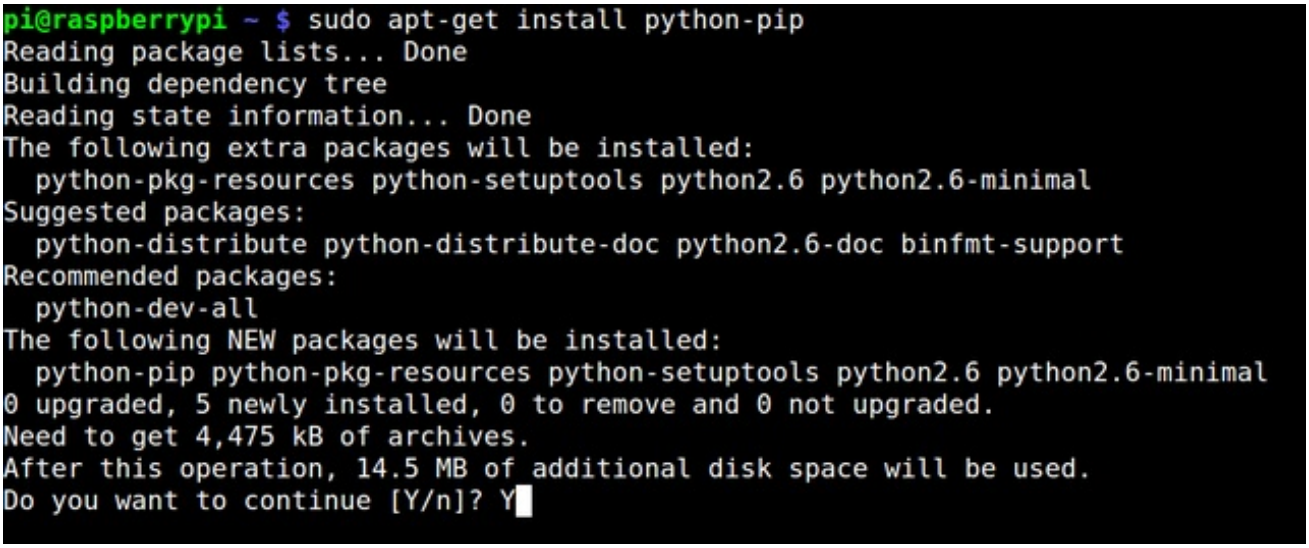
...and here's our guide to using SSH (http://adafru.it/dZL).

Prepare Python

In order for our Python code to work, we'll want to make sure a few libraries are installed.

First, from either the keyboard/monitor or SSH console type in:

```
sudo apt-get install python-pip
```



...you'll be asked if you want to continue. Type "Y" for yes, and hit enter.

```
pi@raspberrypi ~ $ sudo apt-get install python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  python-pkg-resources python-setuptools python2.6 python2.6-minimal
Suggested packages:
  python-distribute python-distribute-doc python2.6-doc binfmt-support
Recommended packages:
  python-dev-all
The following NEW packages will be installed:
  python-pip python-pkg-resources python-setuptools python2.6 python2.6-minimal
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,475 kB of archives.
After this operation, 14.5 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
0% [Working]
```

This part will probably take a little while.

Then you can install the IMAPClient (http://adafru.it/eUP) Python library, which lets Python talk to most e-mail services:

```
sudo pip install imapclient
```

```
Downloading/unpacking imapclient
  Downloading IMAPClient-0.12.tar.gz (100Kb): 100Kb downloaded
  Running setup.py egg_info for package imapclient

Installing collected packages: imapclient
  Running setup.py install for imapclient

Successfully installed imapclient
Cleaning up...
pi@raspberrypi ~ $
```

Older versions of this guide used a Python module called feedparser to read an Atom feed from GMail.

The current version of the script should actually work with any e-mail provider that provides IMAP (http://adafru.it/eUQ) access. A few useful links:

- Yahoo! Mail IMAP Settings (http://adafru.it/eUR)
- Hotmail / Outlook.com IMAP Settings (http://adafru.it/eUS)
- iCloud: Mail server settings for email clients (http://adafru.it/eUT)

I**f you have two-factor authentication enabled on your GMail account**, you'll need to generate an application-specific password to use IMAP. I followed Google's detailed instructions (http://adafru.it/eUU) and was up and running in no time.

Not sure if you have two-factor auth enabled? That probably means you don't, so don't worry about it for now.

Wire LEDs

Now we can use the Adafruit Pi Cobbler (http://adafru.it/914) (for a Raspberry Pi Model B) or the Pi Cobbler Plus (http://adafru.it/2029) (for the Pi Models A+/B+ and Pi 2) to help us wire up the LEDs.

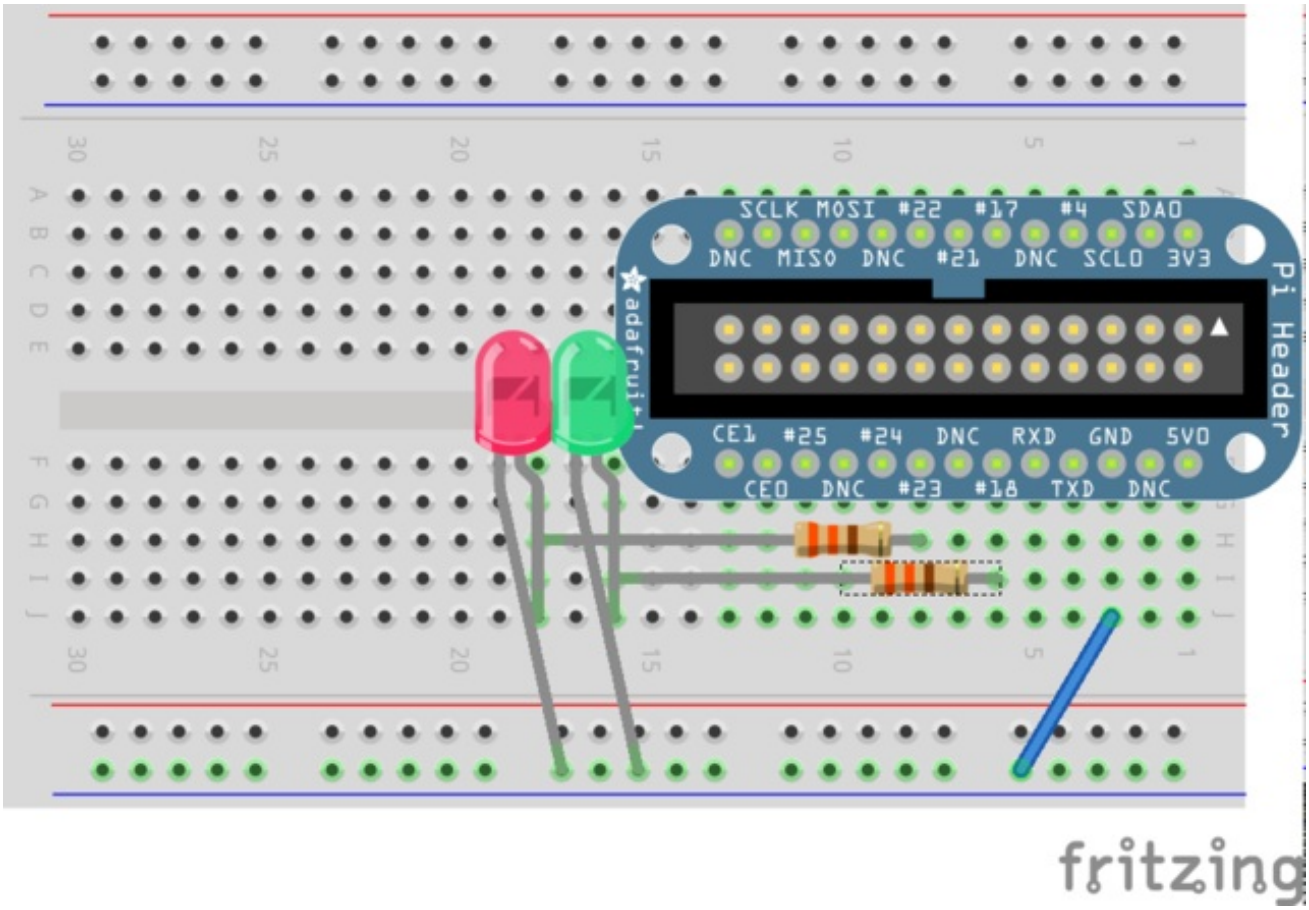
(If you need a refresher on the how and why of breadboards, check out this edition of Collin's Lab (http://adafru.it/eUV).)

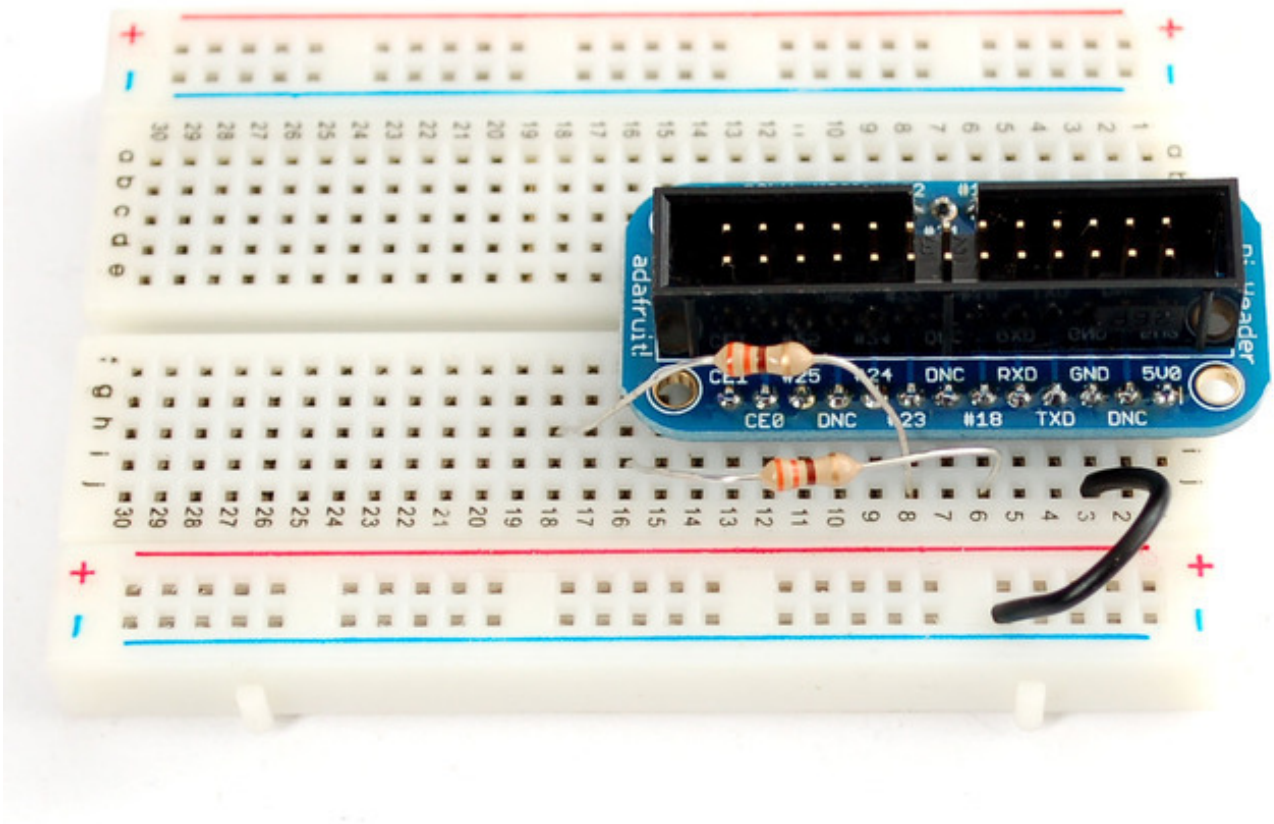
When connecting the GPIO cable, make sure that you note the red or white wire on the ribbon, that's pin #1 of the cable. That end goes at the side closest to the SD Card and is labeled **P1** on the Pi. The other side connects to the cobbler and can only be inserted one way due to the notch in the cable.

Place the cobbler onto the bread board straddling the center line. Connect the **GND** pin (ground) to the blue power rail on the side of the breadboard. You'll need two resistors (any values from 330 ohm up to 1000 ohm are fine).

Connect the first resistor to the cobbler row marked **#18**, and the other end to a row that isn't used by the cobbler.

Connect the second resistor to the cobbler row marked #**23** and the other end to another empty row.

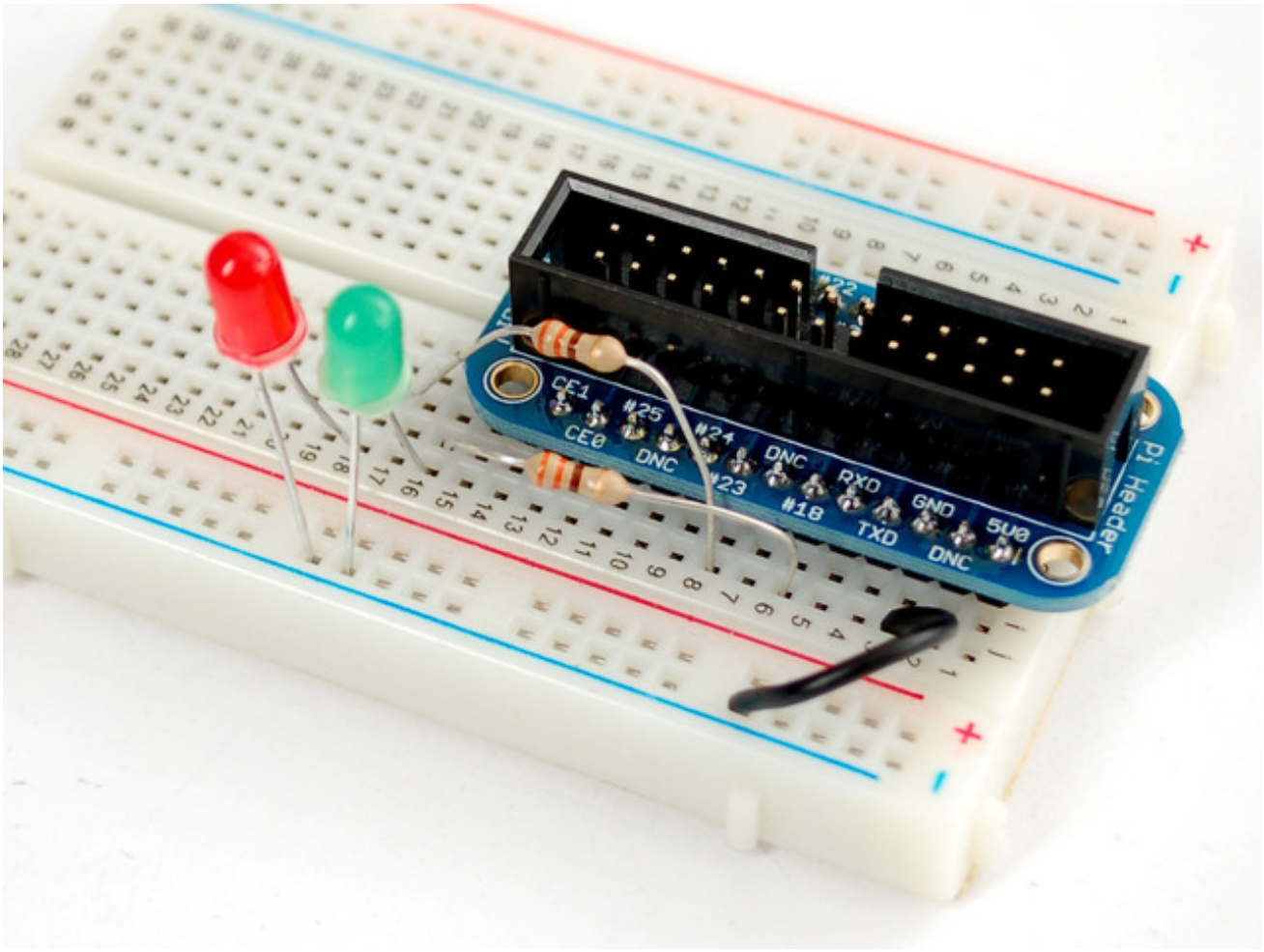

SCLK MOSI #22 #17 #4 SDA0
DNC MISO DNC #21 DNC SCL0 3V3
CE1 #25 #24 DNC RXD GND 5V0
CE0 DNC #23 #18 TXD DNC
Pi Header
adafruit
fritzing



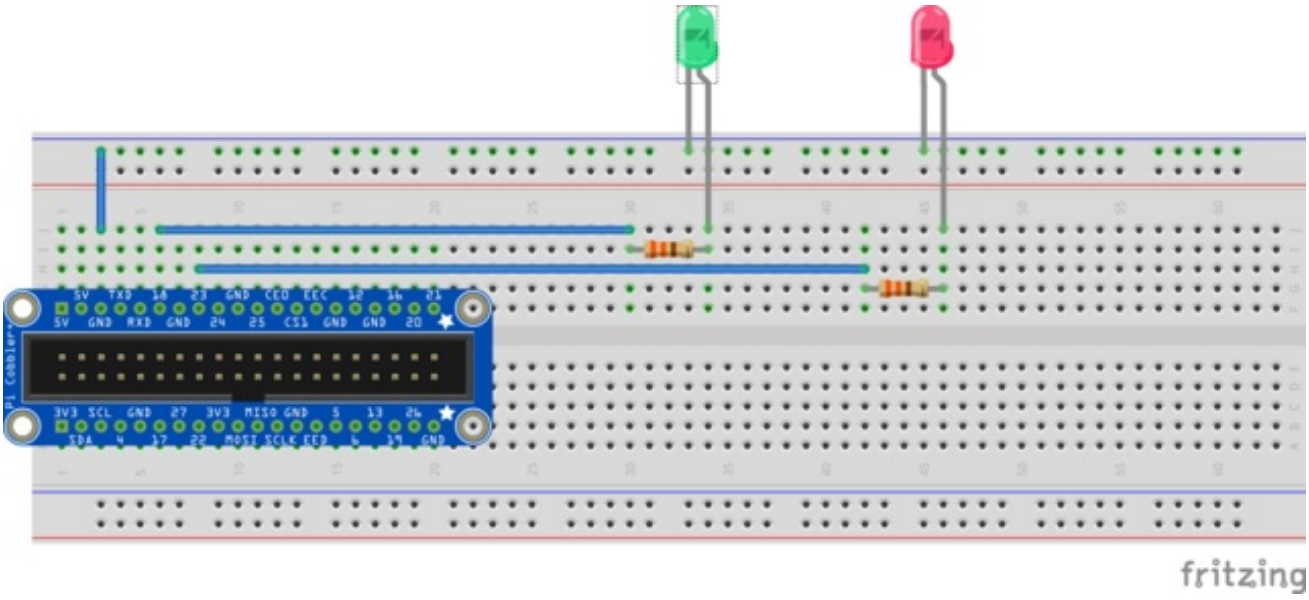
Now grab a red LED and a green LED. Look for the long pins on the LEDs; those are the positive (+) legs.

Connect the long (+) leg of the red LED to the resistor connected to **#23** (GPIO #23), and the long leg of the green LED to the resistor connected to **#18**.

The short legs plug into the blue striped rail on the side of the breadboard.



The above images are for an original Pi Cobbler. For newer, 40-pin models like the A+/B+/Pi 2, you'll probably want to do something more like this, using wires to connect the resistors to the LEDs (click for a larger image):



That's it! You've just wired two LEDs with current-limiting resistors to the GPIO pins of the Pi.

Python Script

Now you're ready to write the code to check your GMail IMAP email (using IMAPClient) and light up the red or green LEDs.

Open up a file called `checkmail.py` with the Nano editor:

```
nano checkmail.py
```

...and paste the following code:

```
#!/usr/bin/env python

from imapclient import IMAPClient
import time

import RPi.GPIO as GPIO

DEBUG = True

HOSTNAME = 'imap.gmail.com'
USERNAME = 'your username here'
PASSWORD = 'your password here'
MAILBOX = 'Inbox'

NEWMAIL_OFFSET = 1   # my unread messages never goes to zero, yours might
MAIL_CHECK_FREQ = 60 # check mail every 60 seconds

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GREEN_LED = 18
RED_LED = 23
GPIO.setup(GREEN_LED, GPIO.OUT)
GPIO.setup(RED_LED, GPIO.OUT)

def loop():
    server = IMAPClient(HOSTNAME, use_uid=True, ssl=True)
    server.login(USERNAME, PASSWORD)

    if DEBUG:
        print('Logging in as ' + USERNAME)
        select_info = server.select_folder(MAILBOX)
        print('%d messages in INBOX' % select_info['EXISTS'])

    folder_status = server.folder_status(MAILBOX, 'UNSEEN')
```

```
    folder_status = server.folder_status(MAILBOX, 'UNSEEN')
    newmails = int(folder_status['UNSEEN'])

    if DEBUG:
        print "You have", newmails, "new emails!"

    if newmails > NEWMAIL_OFFSET:
        GPIO.output(GREEN_LED, True)
        GPIO.output(RED_LED, False)
    else:
        GPIO.output(GREEN_LED, False)
        GPIO.output(RED_LED, True)

    time.sleep(MAIL_CHECK_FREQ)

if __name__ == '__main__':
    try:
        print 'Press Ctrl-C to quit.'
        while True:
            loop()
    finally:
        GPIO.cleanup()
```

Here's a Gist of checkmail.py (http://adafru.it/eUW).

Don't forget to set the `USERNAME` and `PASSWORD` to match your GMail account. (Remember, if you're using two-factor authentication under GMail, you'll need to generate an application-specific password (http://adafru.it/eUU) for this. If you're using a different e-mail provider, you may need to check their documentation for what `HOSTNAME` to use. It's usually something like `imap.youremailproviderhere.com`.)

Next up, we'll mark the file as executable, so that it can run as a standalone program:

```
chmod +x checkmail.py
```

Finally you can run the script! Type in:

```
sudo ./checkmail.py
```

```
pi@raspberrypi ~ $ sudo
```

Send yourself some emails to see the green LED light up!

You can stop the script at any time by pressing **Ctrl-C**.