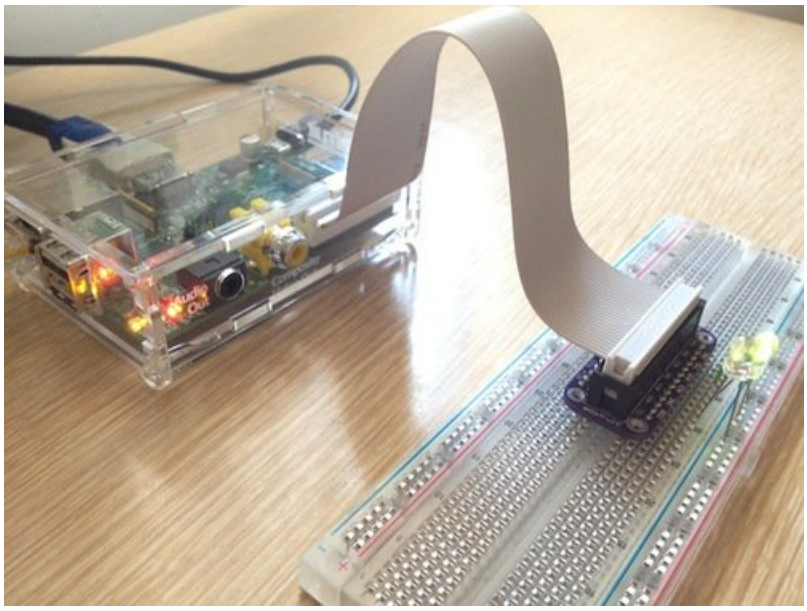
adafruit learning system

Raspberry Pi E-mail Notifier Using LEDs

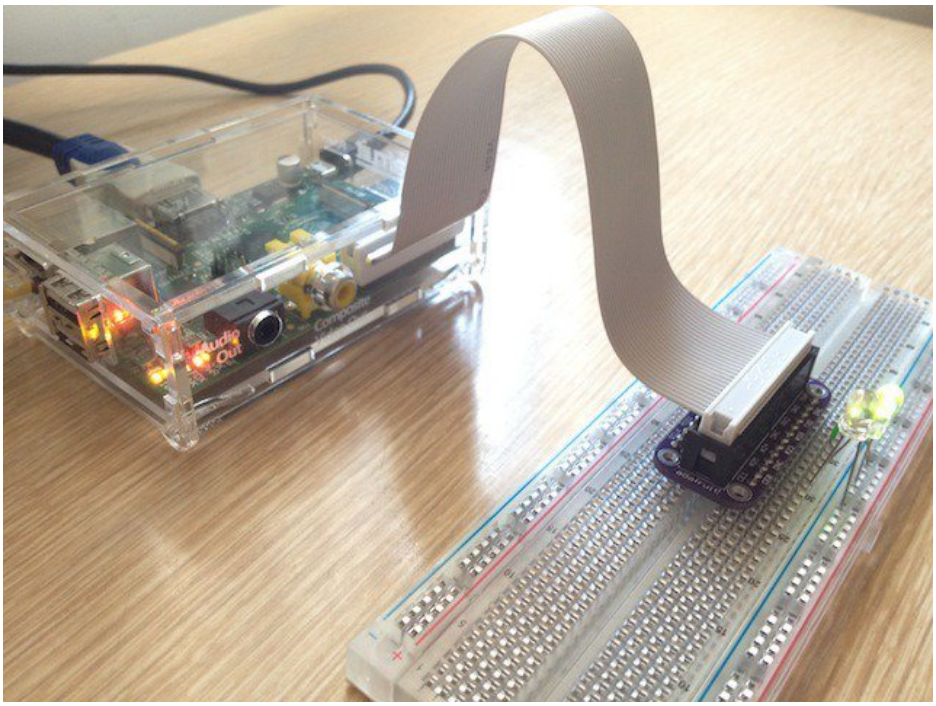
Created by Michael Sklar



Last updated on 2019-02-05 11:21:20 PM UTC

Guide Contents

Overview



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 (https://adafru.it/aJG) had the most ready to go code which I pilfered from.

This project uses two LEDs one green and one red. When a new mail message has arrived the green LED will illuminate. When there are no unread messages the red LED will illuminate.

To Follow This Tutorial You Will Need

- Pi T-Cobbler Plus, (https://adafru.it/xgf) Pi Cobbler Plus for Model B+ / Pi 2 (http://adafru.it/2029) or the original Pi Cobbler (https://adafru.it/DBe)
- (1) Full sized breadboard (https://adafru.it/Bgk)
- Hook-up Wire (https://adafru.it/aM5)
- A Raspberry Pi (https://adafru.it/DBf) (compatible with all 26pin and 40pin Pi releases to date)

HINT: We suggest purchasing the Raspberry Pi 3 Model B Starter Pack (https://adafru.it/DOO) along with a Raspberry Pi (https://adafru.it/Bi1) which includes all the necessary materials for this tutorial.

Wire the Cobbler to the LEDs

Modern Raspberry Pi's use a 40-pin header. The first generation of Pi's used a 26-pin header. This tutorial works with all of these versions using the same code and GPIO pins. We have provided wiring examples of both types using the same GPIO pins (#18 and #23).

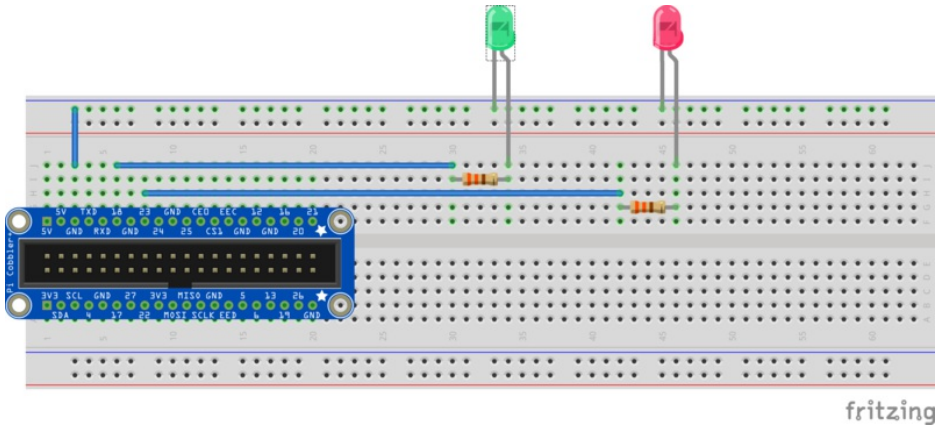
When connecting the GPIO cable, make sure that you note the red or white wire on the ribbon, that's pin #1 of the cable. That end goes at the side closest to the SD Card and is labeled **P1** on the Pi. The other side connects to the cobbler and can only be inserted one way due to the notch in the cable.

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).

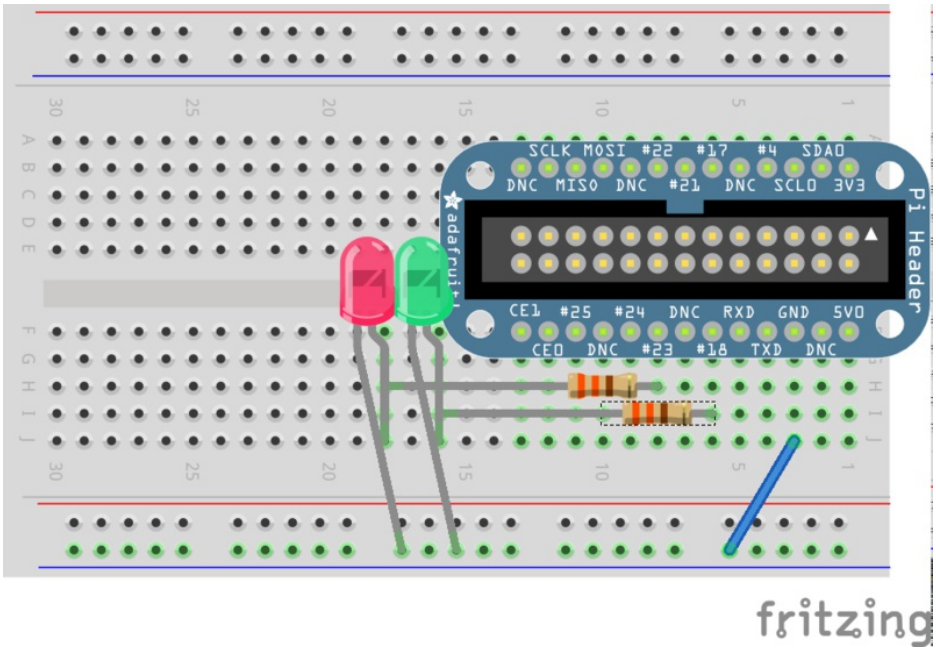
Connect the first resistor to the cobbler row marked **#18**, and the other end to a row that isn't used by the cobbler.

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

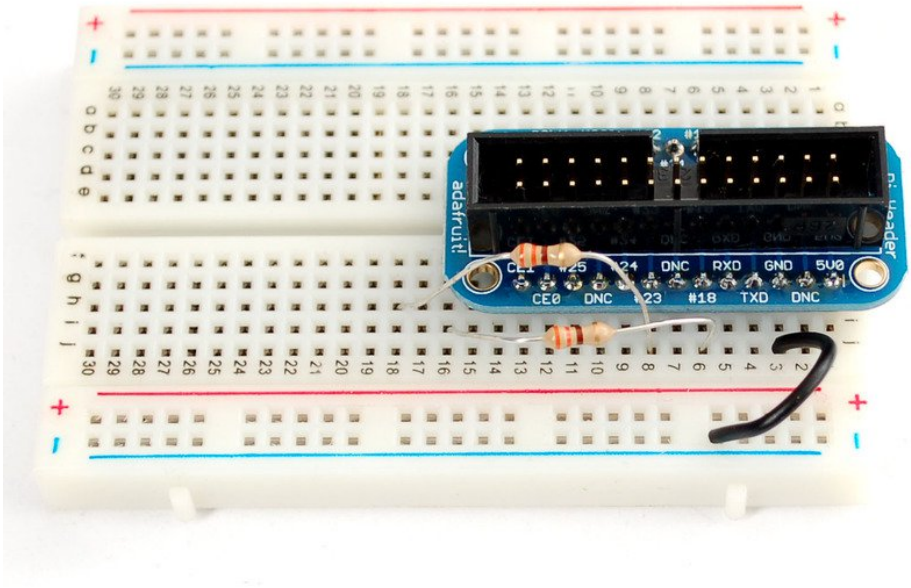
Raspberry Pi Cobbler Plus - 40-pin for Pi v2, v3, Zero



Raspberry Pi Cobber - 26-pin - for Pi v1 only



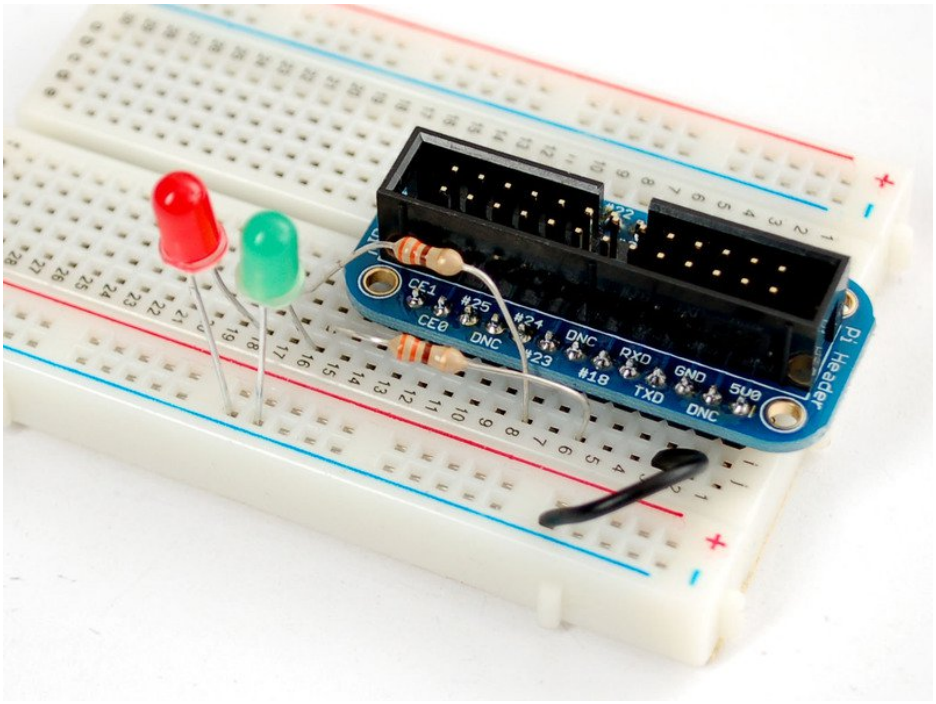
Step-by-Step Hookup on a 26-pin Pi Cobbler



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.

Necessary Packages

Update Your Pi to the Latest Raspbian

Your Pi will need to be running the latest version of Raspbian. This tutorial was written using Raspbian Stretch (Nov. 2018). Checkout our guide for Preparing an SD Card for your Raspberry Pi (https://adafru.it/dDL) if you have not done so already. After the installation is complete be sure and run the following commands to make sure your packages are up to date.

```
$ sudo apt-get install update -y
$ sudo apt-get install upgrade -y
```

Install pip3

pip3 is already installed with a full Raspbian installation, but the Raspbian Lite does not include pip3 so it needs to be installed as shown below.

```
$ sudo apt-get install python3-pip
```

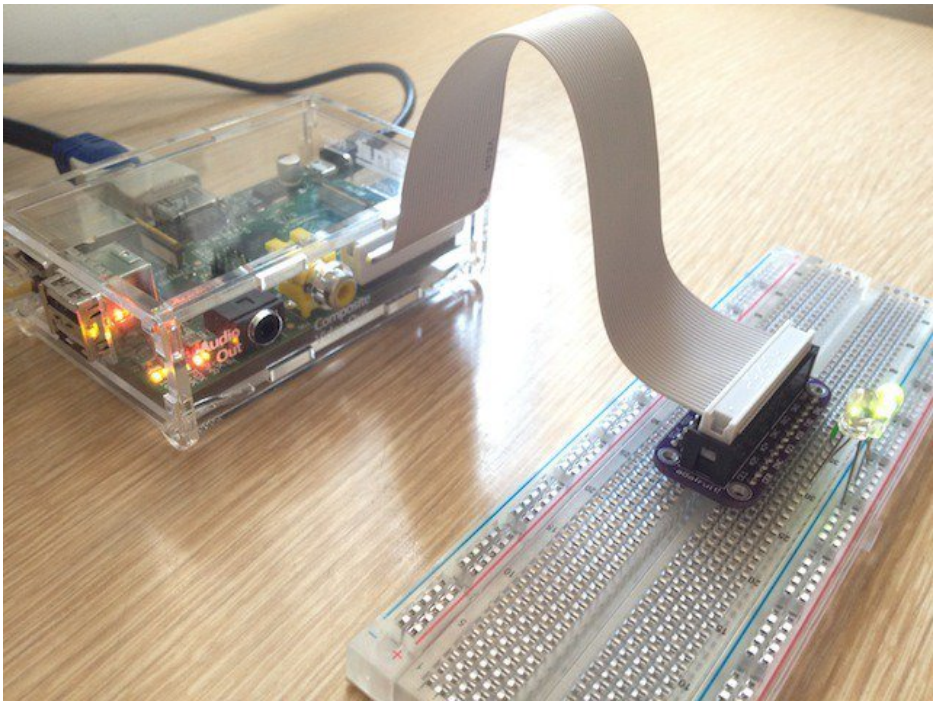
Install adafruit-blinka

```
$ sudo pip3 install adafruit-blinka
```

Install imapclient

```
sudo pip3 install imapclient
```

Python Script



Now you're ready to download some Python code and check your Gmail account. The following python script can be downloaded directly onto your raspberry pi, customized with your email settings and executed to illuminate the LEDs.

```
import time
import board
from imapclient import IMAPClient
from digitalio import DigitalInOut, Direction

HOSTNAME = 'imap.gmail.com'
MAILBOX = 'Inbox'
MAIL_CHECK_FREQ = 60        # check mail every 60 seconds

# The following three variables must be customized for this
# script to work
USERNAME = 'your username here'
PASSWORD = 'your password here'
NEWMAIL_OFFSET = 1          # my unread messages never goes to zero, use this to override

# setup Pi pins as output for LEDs
green_led = DigitalInOut(board.D18)
red_led = DigitalInOut(board.D23)
green_led.direction = Direction.OUTPUT
red_led.direction = Direction.OUTPUT

def mail_check():
    # login to mailserver
    server = IMAPClient(HOSTNAME, use_uid=True, ssl=True)
    server.login(USERNAME, PASSWORD)

    # select our MAILBOX and looked for unread messages
    unseen = server.folder_status(MAILBOX, ['UNSEEN'])

    # number of unread messages
    # print to console to determine NEWMAIL_OFFSET
    newmail_count = (unseen[b'UNSEEN'])
    print('%d unseen messages' % newmail_count)

    if newmail_count > NEWMAIL_OFFSET:
        green_led.value = True
        red_led.value = False
    else:
        green_led.value = False
        red_led.value = True

    time.sleep(MAIL_CHECK_FREQ)

while True:
    mail_check()
```

Download the Code

Let's put this file right in your home directory for simplicity. The wget command makes this easy.

```
$ cd
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Raspberry_Pi_E-m
```

Customize Mail Variables

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 (https://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`.)

Finally, my INBOX never goes to zero unread messages. We have a variable called `NEWMAIL_OFFSET` which you can customize to whatever your current unread message count is. When running this python script there will be a number output to the console showing the current number of unseen messages.

Running the Code

```
$ python3 ./Raspberry_Pi_E-mail_Notifier_Using_LEDs.py
```

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

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