# **Raspberry Pi Twitter Monitor** a

## learn.sparkfun.com tutorial

#### Available online at: <u>http://sfe.io/t160</u>

#### Contents

- The Twitterverse
- <u>Required Components</u>
- Hardware Hookup
- <u>Register Your Twitter App</u>
- Install Required Packages
- <u>Code</u>
- Dissecting the Code
- <u>Run!</u>
- Going Further

#### The Twitterverse

Like it or not, <u>Twitter</u> is a force to be reckoned with. Millions of people use Twitter to talk about general happenings in the world and their lives. Large events like conventions, elections, and uprisings have a habit of <u>crashing the Twitter servers</u>.

You, an avid DIY hacker, want to know how to use that massive stream of digital social media. There are plenty of fun projects built around <u>Tweeting something that happens</u>. Today, we offer you an alternative: make something happen when a specific Tweet occurs.

This is a basic project that will walk you through the steps to get an LED flashing when a specific hashtag appears in the Twitter stream. While a blinking LED might not be the most exciting of outcomes, attaching Twitter to the physical world will get you started on creating that Twitter controlled robot.



Blink an LED whenever specific hashtag appears in a Tweet

#### **Suggested Reading**

We need to build a rather simple LED circuit to connect to the Raspberry Pi's GPIO header. Be sure you are familiar with LEDs and resistors.

- Light-emitting Diodes (LEDs)
- <u>Resistors</u>
- How to Use a Breadboard

# **Required Components**

You will need to have a Raspberry Pi running Linux of some sort. We recommend the atest version of Raspbian, as it comes pre-loaded with Python and GPIO libraries. Additionally, the Pi will need to be connected to the Internet so that you can monitor the Twitter stream. This can be accomplished Page 2 of 14



If you are not planning to use <u>SSH to access your Pi</u>, you will also need a keyboard and a monitor.

### Hardware Hookup

Connect an LED to your Raspberry Pi following this diagram:



Additionally, you will want to plug in the rest of your peripherals (keyboard, monitor, SD card, Ethernet/WiFi) to your Pi. If you are not using a keyboard and monitor, then you will need to find some way to access your Pi, such as <u>SSH</u>.



The GPIO pins connected to an LED

# **Register Your Twitter App**

In order to interact with Twitter (including searching, monitoring, posting, etc.), you will need to register your application. This is accomplished by going to <u>https://dev.twitter.com/</u>.

Click the "Sign In" button at the top-right of the screen and enter your Twitter credentials (you can also sign up if you don't have a Twitter account).

. C	Developers API Health Blog Discussions Do	anetator (hern 4) Signin	L Ditter is
	More downloads	Att 31/1	
	for your app		
	with Twitter Cards		
	Twitter Cards offer a foot and early way to grow your user base for motion apps: timpy and some new manayor to your pages when users twisting to your origonation. Cards will be of their users viewing motion. Twester to download and texnich your app across a mumber of motile pathoms. User May		
	Trease Ended Treases	Tweet +	
	Recent poets from Twitter Developer Blog	Create applications that integrate Twitter	
	everyone everyone	Get started with the platform	
	sac Streaming SSL certificate apdates	Discuss	
	Test accounts with user IDs greater than 30 bits	Get in touch with the @twitterapi team and the community of developers	
	readines tel the story terms the Tweet	Explore Twitter Certified Products	

Once you have signed in, click on your user icon, and select "My Applications."

E Appe Co Projecto Co Bing Co Deval	Peneros 🔯 Eanes 🗁 Masic 🕐 Penerol 💭 Protography 🏠 Spolffun 🔯 Nortop 🖆 Syle 🗋 Rastion	W Migeda	Collect bookman
	Developers API Health Blog Discussions Do	currentiation (leven 4)	
		My systems	
	More downloads	A20 58/01	
	for your app		
	with Twitter Cards		
	Terms Come offer a first and enter one for the state and these		
	tweet cansis over a loss and overy way to give your over open for mobile apps, temps, and some new manual to your pages where seek hand tables to your develop. Cande all let other		
	users viewing these. Tweets to download and isunch your app		
	acrise a number or notive percents.		
		Tweet +	
	Tertiler Cands Embedded Timelines Embedde	d Tureets Tureet Bullion Patien Bullion	
	Recent posts from Twitter Developer Blog	Create applications that integrate Twitter	
	NY RESTARTICL Centrole updates	Get started with the platform	
	whoducing custom trivelines, create timelines of Tweets for	Explore the documentation and manage your appo	
	<ul> <li>Beenvine</li> <li>Beenvine bits certificate aptates</li> </ul>	Discuss Get in touch with the diseiterapi team and the community of developers	
	1	Explore Twitter Certified Products	
	31 car and an and on home and 7 po	Valuable products and services built on Teiliter	

Click the "Create a new application" button, and you will be presented with a form to fill out about your application. Fill out the name and description fields. For "Website," you can put something random, like http://github.com or something else that looks legitimate. At the bottom of the page, click the checkbox to agree to the rules, fill out the Captcha, and click the "Create your Twitter application" button.

← → C @ https://dev.twitter.com/apps/ho III. Apps @ Projects @ Bog @ Enal @ Promos	IV 🗅 Samer 🔿 Mass 🗅 Process 🕞 Processpaphy 🕒 Specifier 🕒 States 💭 Spite 🗋 Sections 🖤 Millipsila	😰 🧮
	🎔 Developers AltiHealth Blag Discussions Documentation (Immin 9) 📓-	
	nome Up approximes	
	Create an application	
	Application Details	
	Rame: 1	
	Tereditivity	
	The application takes The is used to ethnological entropy of a travel and in user-from automotion ecoluties. So characteris new,	
	Description *	
	Binis an LED when a specific hashing appears in Tutter	
	Vise agitudas descriptos, selval-sellas abases o ano facingandestados notesta distante 18 aed 200 charactere ena.	
	Website: 1	
	https://github.com/	
	You's additioning patitive accessed in concessed, where areas we are an obstantiation and work of the food many information advances application. This fight-qualifies (0), a seek for the accessed and the food accessed and the food accessed and the food accessed accesses and the food accesses and the food accesses accesses and the food accesses accesses and the food accesses	
	Caliback DRL	
	When stands on these after accurately administrately for glogostnee marketing, why be donors specified as the states of the work. Each 1d acquisition state printing specify the accurs, nullicars VII, as the respectively in gradient of the radie pion into Townsid para specific theorem, utilized, how this field later.	
	Developer Rules Of The Road	
	Land Update: July 2, 2013.	
	Rules of the Road	
	Traffer marketing on goin platform that suggeds the millions of people answell the weblik sites are sharing and discovering what's happening more. We work to ensure our experiment platform to table of valuate the second test sharing and the share test, and are to share a better to a site our experiment platform and are set of tables and assess of the state and the share test. The same test, we are to share a better to a site out of the state and and tables to better tables. The same test, we are to share a better to a site out of the state and and and tables to be tables.	
	Bit, only come up with a set of Developer Rules of the Road ("Balack") that describes the policies and philosophy around what type of inexcelors as percelled with the context and information shared on Twitter.	
	The finite self-scale along with the accounting as depletence confined to improve the first time, constant ways to see the Turbus MR as discuss	8

Once completed, you will be presented with a dashboard for your registered application. You need to create an access token, so click the "Create my access token" button at the bottom of the screen.

💙 Developers	API Health Blog Discussions Documentation (Immin 4)	
Deteils Zettings	Cifuli had @Republick Benatitis Reset Reys Didde	
Dividu an LED viten a specific heating appears in Teliter. Integri Stylinka scenifi?		
Organization		
reformation about the organization or company associated with your application. This information is optional		
Organization	Bone	
Organization website	Sare	
GAuth settings		
Your application's DNuth settings. Keep the 'Consumer secret' a secret. This key should never be human-eachable in your application.		
Access level	Read-only About the application permission model	
Consumer key	62K117Ed#TuxKEPyINg	
Consumer secont	0x168339v02yr1a6x0xyx68ey68ey68ex68ex682332	
Request token URL	https://wpi.teltter.com/wwwt/request_totes	
Authorize URL	https://wpi.tvinter.com/neuth/exthorine	
Access taken URL	https://wpi.thttts/.com/wwith/worker_times	
Calibach URL	Rea	
Sign is with Twitter		
Your access token		
It lucis like you haven't authorizenthis application for your own Twitter account yet. For your convenience, we give you the opportunity to create your GMuth accoust tables have, so you can rate signing your requests sight away. The accoust striken generated will reflect your application's cannot permission level.		

Wait a few seconds and then refresh the page. You should see a section titled "Your access token" at the bottom. Copy down the "Consumer key," "Consumer secret," "Access token," and "Access token secret." We will need these for our application. Note that we set up this Twitter application as an example. It will have been deleted by the time you read this, so don't be trying to use our tokens!

<ul> <li>✓ Twostliniy (Twiter No. +</li> <li>← → C</li> <li>↓ https://devtwiter.com/apps/520</li> </ul>	Teaching Team to: *		± ₹	
II Appe Co Projecto Co Biog Co Ernal Co Pinances C	Barres (2) Marie (2) Personal (2) Protography (	Spailfur Disatup District Providence Villageda	(mm	Diffee bookman
	Amagination	Perman day produces coordinator		
	international dead the organization or company associated with year application. This information is optional			
	Consciention	Borne .		
	Creatization website	Sone		
	OAuth settings			
	Your application's C4uth setting	p. Keep the "Consumer secret" a secret. This key should never be ho	man-nadiable in your application.	
	Access level	Read-enty Aland the application permission model		
	Consumar key	erkslybet/bettfyme		
	Consumer secont	To SATE Project of a Chynallogic Assarbs at EConstREE (2)		
	Request token URL	https://api.twinter.com/owuth/request_token		
	Authorita+ URL	https://api.twikter.com/seath/authorizer		
	Access taken URL	https://api.twinter.com/deuth/access_tomen		
	Calback URL	lieve .		
	Sign is with Twitter	10		
	Your access token			
	Use the access token telep as your "occos, coast," and the access token secret as your "occos, socce," to sign requests with your own Twitter account. Or not share your awarts, coast, access, and anyone.			
	Access takes	1989000°17-ExtRpVP1ac385aApix7EndW2h76ne8U28Us	CMWV	
	Access taken secret	UVIVERATE SHARE BEEN WITH YOUR DATE OF		
	Access level	Anderly		
	Recedency access lokes			

#### **Install Required Packages**

If you are using a relatively new release of Raspbian, it should already contain Python and the necessary modules to talk to the Pi's GPIO pins. We will still need to install modules to search Twitter with Python. Several exist, but I recommend <u>Twython</u> if you are following this tutorial. Boot up your Pi and connect to the Internet (Ethernet, WiFi, etc.). If you use the X Windows interface, open up a console window and install Twython:

sudo apt-get update sudo apt-get install python-pip sudo pip install twython

**[UPDATE: 2/6/14]:** From <u>Mark A. Yoder</u> - to get this working on the <u>BeagleBone Black</u>, install the following packages in addition to the ones above:

sudo apt-get install python-dev pip install Adafruit\_BBIO

#### Code

With our packages installed, we can write our program. Open up a text editor and make a new Python script. For example:

language:python nano TweetBlinky.py

In the new file, enter the code below.

language:python import time

import RPi.GPIO as GPIO from twython import TwythonStreamer

# Search terms TERMS = '#yes'

# GPIO pin number of LED LED = 22

# Twitter application authentication APP\_KEY = 'erRilYZd8UzsXEFycmg' APP\_SECRET = 'Yt0fGINvCyr1sFaC6ymdNhphHchaWbz0ECdotEXIQQ' OAUTH\_TOKEN = '1969690717-6a2RgVPXanSBaAjuie7EmUWZh78me8UZ6UxcM8V' OAUTH\_TOKEN\_SECRET = 'UIrYV2XbYZC3vHzer6ZxIDwqVa0VvynQLDJYnSQV0R3xt'

# Setup callbacks from Twython Streamer class BlinkyStreamer(TwythonStreamer): def on\_success(self, data): if 'text' in data: print data['text'].encode('utf-8') print GPIO.output(LED, GPIO.HIGH) time.sleep(0.5) GPIO.output(LED, GPIO.LOW) # Setup GPIO as output GPIO.setmode(GPIO.BOARD) GPIO.setup(LED, GPIO.OUT) GPIO.output(LED, GPIO.LOW) # Create streamer try: stream = BlinkyStreamer(APP KEY, APP SECRET, OAUTH TOKEN, OAUTH TOKEN SECRET) stream.statuses.filter(track=TERMS) except KeyboardInterrupt: GPIO.cleanup()

Alternatively, you can download a zip file of the scripthere.

**IMPORTANT**: You will need to change the Twitter authentication tokens! Right now, they are set to my example application. Copy the strings from the your dev.twitter.com page (the single quotes are needed):

```
language:python
APP_KEY = '<Your Consumer Key>'
APP_SECRET = '<Your Consumer Secret>'
OAUTH_TOKEN = '<Your Access Token>'
OAUTH_TOKEN_SECRET = '<Your Access Token Secret>''
```

Save and exit (ctrl + X and 'y' if you are using nano).

#### **Dissecting the Code**

We *could* just give you the code to copy-and-paste into your Pi (which we did). However, the Page 9 of 14

Python script is a bit more complicated than just blinking an LED, so we should take a moment to discuss what is going on.

At the top of the code, we import our necessary Python modules. We need "time" to call our 1/2 second delay when we blink our LED. "RPi.GPIO" is a pre-build Python package for the Raspberry Pi that gives us access to the GPIO pins. This is why we recommend the Raspbian build of Linux. Finally, "twython" is the package that connects us to Twitter allowing us to monitor the stream, search, and post.

language:python import time import RPi.GPIO as GPIO from twython import TwythonStreamer

Next, we add our global constants. TERMS holds a string (or strings) of things we want to search for on Twitter. LED contains the pin number for the GPIO header. Note that this is the *pin number* of the header and not the *GPIO number*. The Twitter authentication constants hold the token strings that we copied from the <u>dev.twitter.com</u> page. Remember: you need to change this section to your specific Twitter token strings.

language:python # Search terms TERMS = '#yes'

# GPIO pin number of LED LED = 22

# Twitter application authentication APP\_KEY = '<Your Consumer Key>' APP\_SECRET = '<Your Consumer Secret>' OAUTH\_TOKEN = '<Your Access Token>' OAUTH\_TOKEN\_SECRET = '<Your Access Token Secret>''

The next section is the crux of the Twitter monitor. If you have never dealt with<u>class inheritance</u> or <u>callbacks</u> before, we recommend brushing up on those concepts in order to really understand this section of code.

We create a new class that inherits TwythonStreamer (a class within Twython). While TwythonStreamer handles a bunch of stuff in the background (including communicating with the Twitter API), we are only concerned with the callback on\_success(). This is a specially named method that gets called whenever a Tweet appears matching our search terms criteria. The important thing to understand is that we *never* call this method from within our script. An outside force (e.g. thread, program) calls on\_success() for us. We just have to define what happens when that method is called. To use it, we create a BlinkyStreamer object and Twython handles the rest.

Inside of on\_success() is what we want to happen whenever a successful Twitter hit occurs. In this case, we make sure there is a 'text' field within the Tweet (meaning that the Tweet contains actual text from a user), and we print the Tweet's text to the console and flash the LED for 1/2 second.

language:python

# Setup callbacks from Twython Streamer class BlinkyStreamer(TwythonStreamer): def on\_success(self, data): if 'text' in data: print data['text'].encode('utf-8') print GPIO.output(LED, GPIO.HIGH) time.sleep(0.5) GPIO.output(LED, GPIO.LOW)

For those of you that are code-savvy, you might say, "Hey, wait a minute! That sleep function is blocking! You can't receive tweets while flashing the LED." And you would be correct. This is just a plain simple way to monitor Tweets. To "properly" do this, you would want to set up a separate thread in Python that monitors Tweets and puts them into a queue. Another thread would read that queue and flash the LED as quickly as Tweets are put into the queue. Threading is a bit too much for this tutorial, but if you would like to learn how, feel free to check out <u>Threading in Python</u>.

The GPIO setup section should look familiar if you have ever played with an Arduino or other microcontrollers. We need to tell the Raspberry Pi that we will be referencing its GPIO headers by board number (pin 1, pin 2, pin 3, etc.), that we want our LED pin to be an output, and initialize the LED to off (logic LOW).

language:python # Setup GPIO as output GPIO.setmode(GPIO.BOARD) GPIO.setup(LED, GPIO.OUT) GPIO.output(LED, GPIO.LOW)

Finally, we create an instance of our BlinkyStreamer class and let it do its thing. When we create a TwythonStreamer object (remember: our BlinkyStreamer inherits TwythonStreamer), we need to feed it our Twitter authentication information that we set earlier. Twython uses this information to connect to the Twitter servers. We provide the filter() method within TwythonStreamer some search terms and the streamer begins to monitor Twitter. Whenever one of the terms appears in the Twitter stream, the on\_success() method (defined above) is called.

```
language:python
# Create streamer
try:
    stream = BlinkyStreamer(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
    stream.statuses.filter(track=TERMS)
except KeyboardInterrupt:
    GPIO.cleanup()
```

Because the filter() method is <u>blocking</u>, the program will run forever until we manually kill it. We can terminate the program by pressing ctrl+c keys. We wrap the streamer calls in a <u>try/catch</u> statement so that the GPIO pins can be released when we tell Python to exit. If we do not do this, Python will issue a stern warning that the GPIO pins are in use by another program.

### Run!

This is *important*. In order to use the GPIO pins from a Python script, you need to run the script with superuser privileges. So, enter the following command:

sudo python TweetBlinky.py

Sit back and watch the fun! As people issue the ubiquitous #lol hashtag, you will see the Tweets appear in your console.

**DISCLAIMER:** This is Twitter we are talking about. We can make no guarantee regarding the appropriateness of comments that might appear on your screen.



Now, look over at your Raspberry Pi. That LED that you hooked up should flash with every incoming Tweet that contains the hashtag #lol.



The LED flashes for 1/2 second on an incoming Tweet with the appropriate hashtag

When you want to end the program, press *ctrl+c*.

### **Going Further**

So, you got something that flashed to notify you of Tweets. Great! The LED is just a start. Just about anything can be hooked up to the Raspberry Pi to notify you of Twitter activity. Let your imagination run wild...



A <u>car horn</u> to alert you of incoming Tweets? Why not?!

If you want to dig deeper into the Twython package, check out the Github repository here: <u>https://github.com/ryanmcgrath/twython</u>

The Raspberry Pi may not be as good at connecting to low-level peripherals as the Arduino, but it does offer a number of options on its 26-pin header. If you are looking to hook up additional hardware to the Pi, take a look at the <u>RPi's Low-Level Peripherals</u> page. Remember: the Pi uses 3.3V logic!

learn.sparkfun.com | CC BY-SA 3.0 | SparkFun Electronics | Niwot, Colorado