

# サーバ情報に従いRaspberryPiのGPIOを制御する方法

2018/01/05

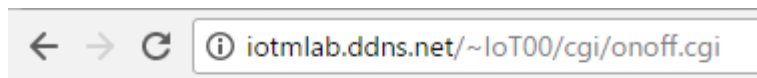
石山隼行

ドキュメント、【IoTサーバで簡易データ保持とスマートフォン連携】

[https://docs.google.com/document/d/1wETYjULJmXvk-fqhl6lHwRjl8DaLloarfzjtTV\\_1rg/edit?usp=sharing](https://docs.google.com/document/d/1wETYjULJmXvk-fqhl6lHwRjl8DaLloarfzjtTV_1rg/edit?usp=sharing)

を参考にし、

Webサーバ（仮に） <http://iotmlab.ddns.net/~IoT00/cgi/onoff.cgi>



## ON

の表示内容がONの時（ <http://iotmlab.ddns.net/~IoT00/cgi/onoff.cgi?ON> でONが記憶される）に RaspberryPiからGPIOシーケンスを開始するサンプルプログラムを作成しました。

Requests の使い方 (Python Library)

<https://qiita.com/sqrtxx/items/49beaa3795925e7de666>

および

Raspberry PiでPythonのRPi.GPIOを使ってLチカする

<https://qiita.com/masato/items/715e28e0c0c945a54297>

を参考にして下記のように作成した

## HTMLスクレイピング部

Requests の使い方 (Python Library)

<https://qiita.com/sqrtxx/items/49beaa3795925e7de666> 参考

まずRasPiにrequestsをインストール

```
$ sudo pip install requests
```

以下python3でテスト

```
$python
```

```
>>> import requests
```

```
>>> r=requests.get('http://iotmlab.ddns.net/~IoT00/cgi/onoff.cgi')
```

```
>>> r.text
```

```
'<!DOCTYPE html\n\tPUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"\n\t"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">\n<html\nxmlns="http://www.w3.org/1999/xhtml" lang="en-US"\nxml:lang="en-US">\n<head>\n<title>部屋の電灯</title>\n<meta\nhttp-equiv="Content-Type" content="text/html; charset=utf-8"\n/>\n</head>\n<body>\n<h1>ON</h1>\n</body>\n</html>'
```

このHTML内容にONがあるかどうかを判定するにはfindで行います。

※WebスクレイピングにBeautifulSoup等を利用しても良いですが、処理が重くなるので単にテキストから情報を検索してみます。

```
>>> print(r.text.find('<h1>ON</h1>'))  
312
```

↑321文字（バイト）目にONがある。なければ-1が帰ります。

この状態でGPIO制御するプログラムを書けばよいわけです。

## GPIO制御

Raspberry PiでPythonのRPi.GPIOを使ってLチカする

<https://qiita.com/masato/items/715e28e0c0c945a54297>

を参考に、Lチカプログラムをdef L\_Brink で関数にし、下記のようなプログラムを作成しました

```
$ sudo apt-get install python-rpi.gpio
```

test.py

```
# -*- coding: utf-8 -*-  
  
import requests  
import RPi.GPIO as GPIO  
import time  
  
def L_Brink(count,pin):  
    #COUNT = 3  
    #PIN = 11  
    GPIO.setmode(GPIO.BOARD)  
    GPIO.setup(pin,GPIO.OUT)  
  
    for _ in xrange(count):  
        GPIO.output(pin,True)
```

```

        time.sleep(1.0)
        GPIO.output(pin,False)
        time.sleep(1.0)

    GPIO.cleanup()

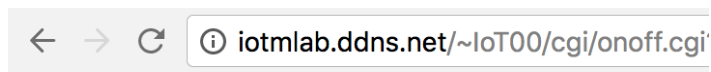
#main loop
while 1: #永久ループ。終了はCtrl+Cで
    r = requests.get('http://iotmlab.ddns.net/~IoT00/cgi/onoff.cgi')
    if(0<r.text.find('<h1>ON</h1>')):
        print('OK Go')
        L_Brink(3,11) # 3回 11番PINをTrue->False変化=Lチカさせる
        print('Ture End')
        break #一度でも実行したら終了

    print('NG - Loop Again')

```

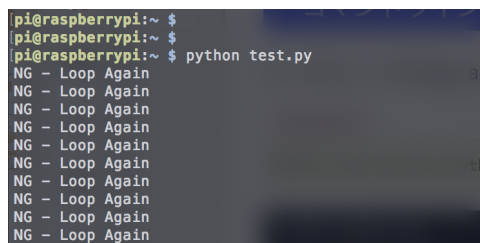
サーバの表示がONならば、L\_Brink(3,11)を実行。そうでなければ（サーバ内がOFFならば）ループし続けるプログラムです。

実行結果  
まず、



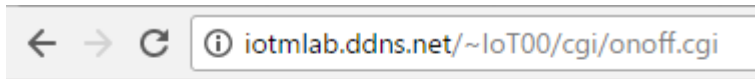
# OFF

OFF状態でスクリプトを実行すると、



NG - Loop Againと表示されつづけます。

ここで<http://iotmlab.ddns.net/~IoT00/cgi/onoff.cgi?ON> でON状態にすると



# ON

```
NG - Loop Again  
NG - Loop Again  
NG - Loop Again  
OK Go  
True End
```

OK Go と表示され、GPIOシーケンスがスタートし、True End となりループから終了します。

※ループさせ続けたい場合はbreakをコメントアウトしてください。

このプログラムのL\_Brinkを作り替え、各自のGPIO制御シーケンスを入れてみてください。

ー以上ー