

RaspberryPiを使った定点カメラの作成

s1260242

Ryusei Takahashi

July,22

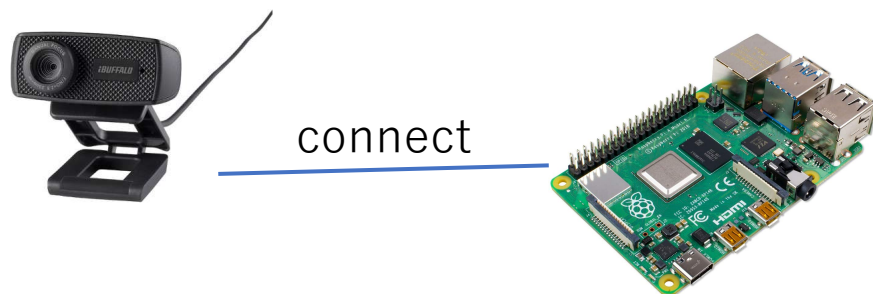
Contents

- Current Status before Today's class
- Motivation
- Purpose
- Functional spec
- Today's content
- What I did on Today's class

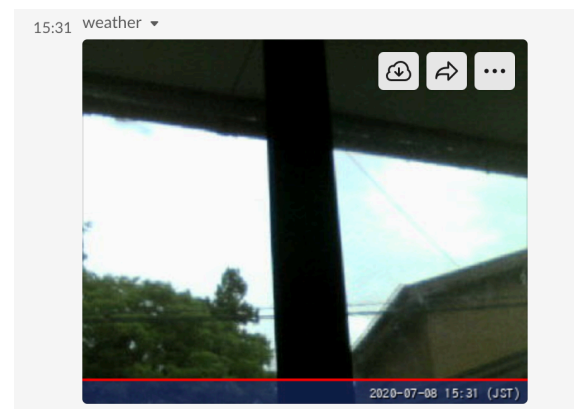
Current Status before Today's class

ラズパイに繋いだカメラで取得した画像をスラック にアップロードすることができる。

実装済みの機能



Send a picture by using
python code and Slack API



That's the weather right now.

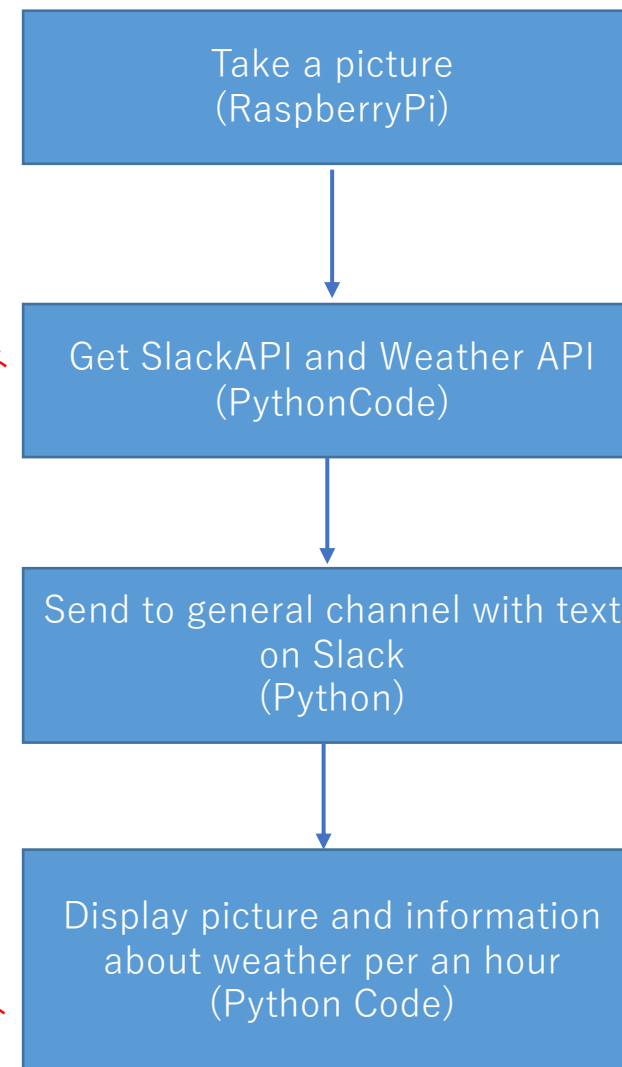
スラックAPIのみ実装済み

単位時間ごとに写真を撮影し、
Slackに投稿できるように実装済み

実装済み

実装済み

Operation Flow



Motivation

今日はOpenWeatherMapを用いて天気情報を取得できるようにしたい。

(6/24に提出した計画スライドには、LivedoorのWeatherHackを利用すると記載していましたが、このサービスが7月末に終了してしまいますのでOpenWeatherMapを利用します。)

Purpose

- 定期的に自分のアパートの天気を確認して、帰省時や旅行時に使いたい。雨なら傘の準備ができ、気温の変化によって着ていくものを考えることができるようにしたい。

Functional Spec

- カメラで写真を取得することができる。
- 取得した写真を保存し、Slackの指定したgeneralチャンネルに送信することができる。
- Slackに画像を送信する際、"That's the weather right now."と表示される。
- Crontab.textの設定によって単位時間あたりのプログラムの実行ができる。

Today's Content 2nd Period

WeatherHackに代わるお天気APIを探し、OpenWeatherMapと言うサービスを見つけました。
また、実装にはpytzを使うので“pip install pytz” を叩き環境を整えました。



Today's Content 3rd Period

実装しました。

参考：<https://qiita.com/marksard/items/472000594ca83b64f00c>

```
weather.py No Selection
1 # -*- coding: utf-8 -*-
2 import json
3 import datetime
4 import os
5 import requests
6 import sys
7
8 #print(sys.path)
9
10 from pytz import timezone
11
12 API_KEY = 'da1bc6b2497e5f663ff00c738ed049f0'
13 ZIP = '100-0001,JP'
14 API_URL = 'http://api.openweathermap.org/data/2.5/forecast?zip={0}&units=metric&lang=ja&APPID={1}'
15
16 def getWeatherForecast():
17     url = API_URL.format(ZIP, API_KEY)
18     response = requests.get(url)
19     forecastData = json.loads(response.text)
20
21     if not ('list' in forecastData):
22         print('error')
23         return
24
25     # print(forecastData)
26
27     for item in forecastData['list']:
28         forecastDatetime = timezone('Asia/Tokyo').localize(datetime.datetime.fromtimestamp(item['dt']))
29         weatherDescription = item['weather'][0]['description']
30         temperature = item['main']['temp']
31         rainfall = 0
32         if 'rain' in item and '3h' in item['rain']:
33             rainfall = item['rain']['3h']
34
35         print('日時:{0} 天気:{1} 気温(℃):{2} 雨量(mm):{3}'.format(forecastDatetime, weatherDescription, temperature, rainfall))
36
37 getWeatherForecast()
38
```

5日間の天気を三時間ごとに表示することができます。

ZIPには郵便番号を格納しているので黒塗りしました。

Today's Content 4th Period

3rdPeriodで作成したコードは情報があまりにも多いので一度の実行でひとつ分の天気情報を取得しなかったのですが、なかなかうまくいかずエラーの連続で終わらせることができませんでした。これから来週に向けて改善していこうと思います。

```
❏ < > 📄 rasWeather.py > No Selection
1 # -*- coding: utf-8 -*-
2 import json
3 import datetime
4 import os
5 import requests
6 import sys
7
8 from pytz import timezone
9
10 API_KEY = 'da1bc6b2497e5f663ff00c738ed049f0'
11 ZIP = ████████.JP'
12 API_URL = 'http://api.openweathermap.org/data/2.5/forecast?zip={0}&units=metric&lang=ja&APPID={1}'
13
14 def getWeatherForecast():
15     weatherDescription=list.weather.main
16     temperature = list.main.temp
17
18     print("weather:{0}, temp:{1}".format( weatherDescription, temperature))
19
20 getWeatherForecast()
21
```

ZIPには郵便番号を格納している
ので
黒塗りしました。

What I did on July 22

weatherHack by Livedoorが今月末でサービスを終了するのでOpenWeatherMapを使用して天気情報の取得を試みました。しかし、自分の思い通りの表示をすることができませんでした。

Schedule

- ~~7/8:カメラの動作確認とSlackのTokenの発行し、Pythonコードを作成して画像をアップロードできるようにする~~
- ~~7/15:crontabを用いて、1分おきに画像をアップロードできるようにする~~
- ~~7/22:画像だけになってしまうので、日付と天気情報(気温や湿度)を表示する~~
- 7/29:22日に同じ
- 8/4:プレゼンテーションの作成

この実装には二日分の時間を確保しているので、スケジュールに変更はありません。