

RaspberryPiを使った定点カメラの作成

s1260242

Ryusei Takahashi

July,15

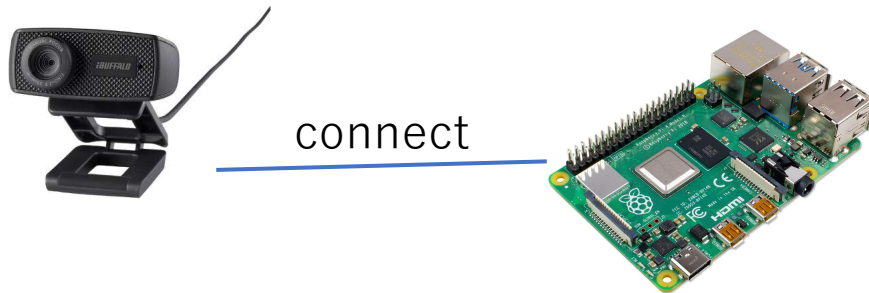
Contents

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

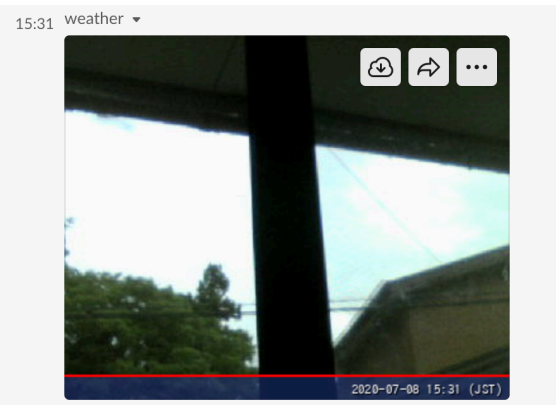
Current Status before Today's class

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.

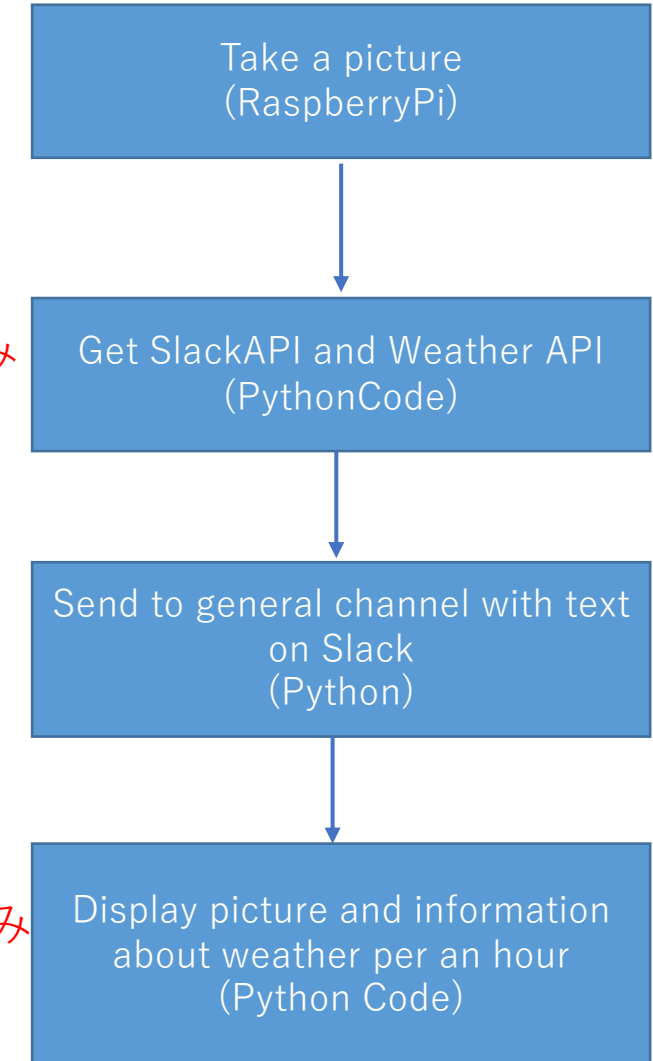
実装済み

スラックAPIのみ実装済み

実装済み

写真をアップロードすることのみ
実装済み

Operation Flow



Motivation

今日はcrontabを用いて、定期的に写真をSlackにアップロードできるようにしたい。最終的には1時間おきだが、今回はテスト実装のために1分おきの実装となっている。

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。
- 最終的には1時間おきだが、今はテスト実装のために1分おきに確認できるように実装している。

Today's Content 2nd Period

スライドやインターネットでcrontabについて調べ、実装した。

```
ryuseitakahashi@Amaterasu progress % cat crontab.txt  
*/1 * * * * python /home/pi/exercises/capture.py
```

crontabについて、左から実行時間を分、時、日、月、曜日として指定する。

テスト実装としては1分毎に実行したかったので1としている。

1時間毎にしたいなら、これを60にする。

さらにその右からは実行コマンドを書き記す。

python /home/pi/exercises/capture.py

は1分毎に実行するコマンド

Today's Content 3rd Period

crontabを実現するために、/etc/rsyslog.copnf と /etc/default/cronの設定をし、crontab.txtとcapture.pyのパフォーマンスを設定した。

/etc/rsyslog.copnfでは、63行目の #cron.* /var/log/cron.log のコメントアウトを解除し、
sudo /etc/rsyslog.copnf restart を実行した。

```
cron.* /var/log/cron.log
```

/etc/default/cronでは、一番最後の行を、# EXTRA_OPTS=""からEXTRA_OPTS="-L 15"に
コメントアウトを解除し、変更を加え。その後、sudo /etc/default/cron restartを実行した。

```
EXTRA_OPTS="-L 15"
```


Today's Content 3rd Period

```
chmod 777 crontab.txt
```

```
chmod 777 capture.py
```

を実行し、全てのユーザーが読み書き、実行をできるようにパーミッショを設定した。

Today's Content 4th Period

capture.pyの設定と、変更後の動作確認。

3行目のos.systemを絶対パスに変更

左側は変更前（前回）、右側が変更後（今回）

os.system("fswebcam /dev/video0 weather.jpg") から

os.system("fswebcam /dev/video0 /home/pi/exercises/weather.jpg")

```
import os

os.system("fswebcam /dev/video0 weather.jpg")

from slacker import Slacker

#AuthToken
token = "xoxp-1234430567092-1213507232231-1228269505394-0a816f3d41448d3ff85fdcecb14c7e00"

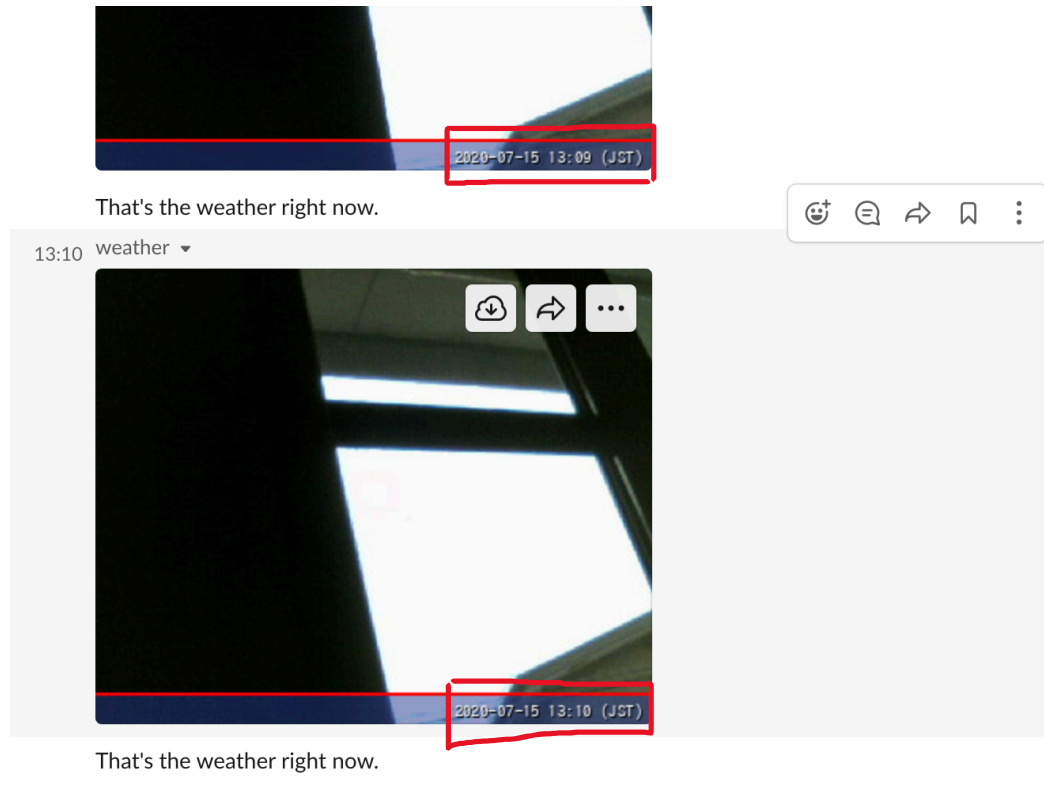
slacker = Slacker(token)
channel_name = "#" + "general"
result = slacker.files.upload("/home/pi/ex8/weather.jpg", channels=["C016WF9UWN8"])
slacker.chat.post_message("C016WF9UWN8", "That's the weather right now.", as_user=True)
slacker.pins.add(channel="C016WF9UWN8", file=result.body["file"]["id"])
```



```
1 import os
2
3 os.system("fswebcam /dev/video0 /home/pi/exercises/weather.jpg")
4
5 from slacker import Slacker
6
7 #AuthToken
8 token = "xoxp-1234430567092-1213507232231-1228269505394-0a816f3d41448d3ff85fdcecb14c7e00"
9
10 slacker = Slacker(token)
11 channel_name = "#" + "general"
12 result = slacker.files.upload("/home/pi/exercises/weather.jpg", channels=["C016WF9UWN8"])
13 slacker.chat.post_message("C016WF9UWN8", "That's the weather right now.", as_user=True)
14 slacker.pins.add(channel="C016WF9UWN8", file=result.body["file"]["id"])
15
```

What I did on July 15

前回までは、"python capture.py"を実行しなければ、Slackに写真をアップロードできなかったが、今回はcrontabを用いて、raspberryPiに電源がついている限り1分毎に、繰り返し実行できるようにした。



上の写真：2020-07-15 13:09 (JST)
下の写真：2020-07-15 13:10 (JST)
と、表示されている。

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

今日の目標が予定通りに終わったので、スケジュールに変更はありません。