

第 3 章

まだ Reaction で消耗してるの？

hideo54

3.1 :about_me:

こんにちは、SunPro メンバーの 1 人、hideo54 です。灘高校で高校 2 年生をしながら、いろいろ遊んでいます。最近だと iOS や Web で動くようなプログラミングを好んでいます。

あと、この記事の内容も該当しますが、工作とプログラミングを絡めたい時には Raspberry Pi を愛用しています。(そういえば、前回の SunPro 会誌でも RPi を使って遊ぶ記事を書きました^{*1}。無料公開していますのでぜひ。)

■コラム: 自己紹介リンクいろいろ

- Twitter: [@hideo54](#)
- Homepage: <https://hideo54.com>
- Blog: <https://blog.hideo54.com>
- Wishlist: <https://wishlist.hideo54.com>
- E-mail: contact@hideo54.com

3.2 :introduction:

さて、本題に入りたいと思います。

皆さんご存知の通り、現在 Slack には、Reaction という、各投稿に対して絵文字で反応をつけられる機能があります^{*2}。この機能はめっちゃ便利だし楽しいのですが、付けるのが少し面倒という側面があります。たとえば、+:+1:絵文字を Reaction として付けたい時には、直前の投稿に付けるならば+:+1:と入力するか、そうでない場合は“Add reaction”を押して、表示された絵文字パレットから Reaction として付けたい絵文字を選択しなければいけません。

^{*1} <https://sunpro.io/c89/pub/hideo/ch01>

^{*2} <https://get.slack.help/hc/en-us/articles/206870317-Emoji-reactions>

個人的に、簡単な反応であればなるべく Reaction で済ませたいという気持ちがありますので、頻繁に使うこの面倒な Reaction をなんとか楽にできないかと考えていた結果、**Reaction 専用キーボードを作る**という発想に至りました。

3.3 :plan:

「キーボード 自作」とかでググればやたら Arduino でしてやったぜ記事が出てくるのですが、Arduino に疎く Raspberry Pi に慣れている身なので、Raspberry Pi でなんとかできないかと考えました。(そのうちちゃんと Arduino も触っておきたい…。)

やり方として、マイコンチップを使ってどやこやして USB キーボードを作り特定マシンに接続して文字入力、という手も考えはしたのですが、設計が非常に面倒くさい上、接続先マシンで Slack を選択状態にしておく必要があります。これではメリットが薄くなりますので、マシンに接続して文字入力をするような USB キーボードではなく、ボタンを押したら Slack API を叩いて Reaction を送信、という形をとることにしました。

というわけで、仕様をまとめるとこんな感じです:

- ボタンやらなんやらを置いた基盤を用意し、Raspberry Pi の GPIO ピンと接続する
- 特定のボタンが押されたら、そのボタンに割り当てられた Reaction を送信

Slack メッセージの送信には、Real Time Messaging API³を使います。

3.4 :design:

Reaction ボタンを作ります。今回使用するものは以下のとおりです:

- Raspberry Pi
- SD カード
- ブレッドボード
- 押しボタンスイッチ (1 回路 1 接点) 4 つ
- 抵抗

まず、SD カードに Raspbian Jessie をインストール⁴し、Raspberry Pi に入れて起動します。Jessie ベースの Raspbian は、Wheezy と違って GPIO の制御に root 権限が必要ないという長所もあるのです。

ブレッドボードに各スイッチを刺し、抵抗を通すようにして適当に配線します。

図 3.1 は配線例です。狭いブレッドボードを使ったので、配線がややわかりにくい感がありますが、許してください。

³ <https://api.slack.com/rtm>

⁴ <https://www.raspberrypi.org/downloads/raspbian/> に DL リンクがあります。インストール方法は適当にぐぐってください。

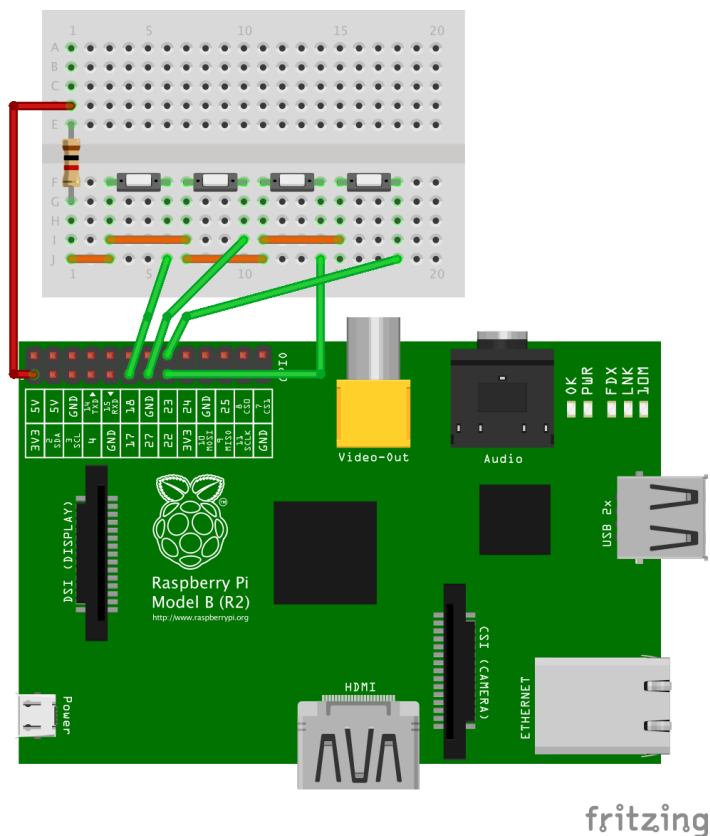


図 3.1 配線例

図 3.1 では、11, 13, 15, 16 番目のピンをそれぞれのスイッチに割り当てています。次章のソースコードでも、この配線を前提としたものを掲載しています。

実物の写真も用意しました。こんな感じです。

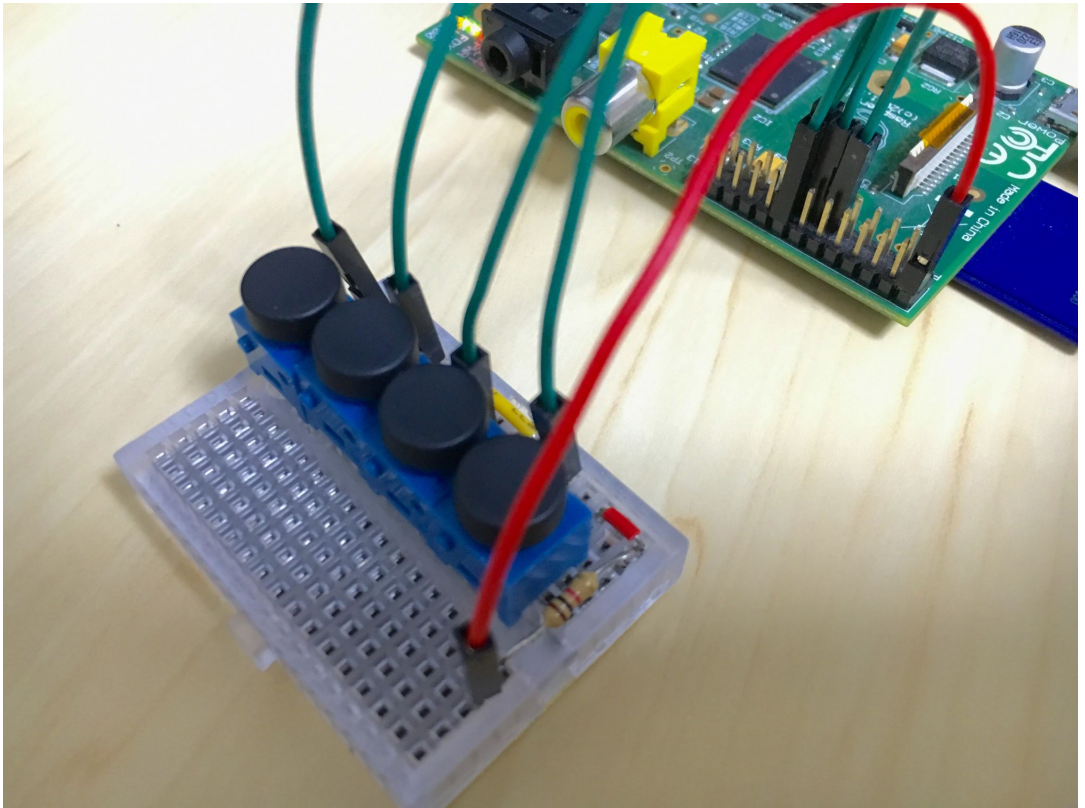


図 3.2 実物写真

(ボタンの様子が見えやすいように、配線図と上下が逆になるように撮りました。)

3.5 :code:

実装めんどくさいな〜、と思うところですが、ありがたいことに有志が作ったライブラリがたくさんあります^{*5}。(感謝)

今回は、僕が一番手馴れている Python で書きたいと思ったので、python-slackclient^{*6}を使うことにしました。pip などインストールすると良いでしょう。

というわけで書いたコードがこちらです。

ボタンを押すと Reaction が送信されるコード (Python3)

```
1 import time
2 import RPi.GPIO as GPIO
3 from slackclient import SlackClient
4
5 # Slack
6 token = 'xoxp-1234567890a-1234567890a-1234567890a-1234567890'
7 sc = SlackClient(token)
8
9 # Buttons
10 reactions = {11:'+1', 13:'weary', 15:'wakaru', 16:'uke'}
11 GPIO.setmode(GPIO.BOARD)
12 GPIO.setup(list(reactions.keys()), GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

^{*5} <https://api.slack.com/community>

^{*6} <https://github.com/slackhq/python-slackclient>

```

13 def add_selected_reactions(sc, channel_id, ts):
14     for button in reactions.keys():
15         if GPIO.input(button):
16             reaction = reactions[button]
17             sc.api_call('reactions.add', name=reaction, channel=channel_id, timestamp=ts)
18
19 if sc.rtm_connect():
20     latest_message = None
21     while True:
22         res = sc.rtm_read()
23         for item in res:
24             item_type = item.get('type')
25             if item_type == 'hello':
26                 print('Connected successfully.')
27             elif item_type == 'message':
28                 if latest_message != item:
29                     latest_message = item
30             else:
31                 pass
32         if latest_message is not None:
33             add_selected_reactions(sc, latest_message['channel'], latest_message['ts'])
34         time.sleep(1)
35 else:
36     print('Connection failed.')

```

4行目の token は、OAuth Test Tokens⁷から取得します。10行目の reactions は、key に使用するピン番号、value に対応する Emoji の名前をとる辞書です。ここでは、僕が所属している灘校パソコン研究部⁸の Slack にある Reaction のうち、この4つがあればなんとかなるかな、というものを選んでいきます。

Reaction の追加は、python-slackclient の api_call メソッドを使用しまんま該当する Slack API⁹を叩いています (17行目)。

これを Raspberry Pi 上で実行しておけば、ボタンが押されていると、そのボタンに対応した Reaction を付加するようになります！ (ただしインターネット接続が必要)

ボタンが押されているかどうかを1秒毎に見ているので、最大で1秒押し続ける必要がありますが、叩いているのは追加メソッドなので、複数命令が送られても Reaction が取り消されることはないようになっています。

3.6 :postscript:

と、まあ、以上です。実際は、さらに RPi にディスプレイ付けて、受信したメッセージを選択して Reaction を付けるなど、より実用的なものを作りたかったのですが、記事の締め切り直前に作業を始めたため、さすがにそこまでは行きませんでした…。(大げさなタイトルの割に単純なものに仕上がってしまった…)

曰く「かなり許されない部類」のギリギリ提出になってしまったものの記事を待ち続けてくれた編集長の hakatashi には深く感謝しています…！

余談ですが、hakatashi に Slack で「hideo の記事、機械学習で Slack のメッセージに自動で Reaction つけるみたいな記事かと思ったら全然違った」と言われました。それも一瞬考えていたので、またそのうちやりたいですね。

何かあれば、冒頭で書いた Twitter アカウントやメールアドレスまでご連絡ください。ここまで読んでくださってありがとうございます！

⁷ <https://api.slack.com/docs/oauth-test-tokens>

⁸ <http://www.npca.jp/>

⁹ <https://api.slack.com/methods/reactions.add>