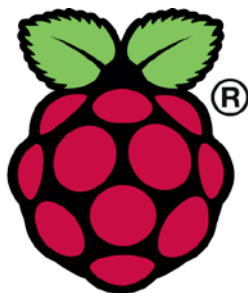


抜粋版

Lineと連携した雷センサIoT開発キット

～RaspberryPiを使い、ゴルフ場から農作業まで、落雷から人命を守ります～

実践編



Raspberry Pi



スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales@spectrum-tech.co.jp

IoT開発キット 目次

Pi運用マニュアル

1. RaspberryPiについて
2. Linux基本コマンド
3. 基本操作
4. 日常運用(ウイルススキャン、更新)

ページ

[3](#)
[4](#)
[5](#)
[6](#)

IoT開発

- ① メニュー
- ② 雷センサ 全体構成
- ③ 雷センサ接続、設定
 - AS3935仕様 [10](#)
 - AS3935雷センサとRaspberryPiを接続 [12](#)
 - AS3935の校正 [13](#)
- ④ 単体試験
 - AS3935の単体試験 [14](#)
- ⑤ IFTTT設定
 - IFTTTを使ってLineへの通知を設定 [15](#)
 - Lineグループへの通知設定 [26](#)
- ⑥ Line接続
 - Line連携試験 [30](#)
 - AS3935からLineまでの接続確認 [31](#)

ページ

[8](#)
[9](#)

[10](#)
[12](#)
[13](#)

[14](#)

[15](#)
[26](#)

[30](#)
[31](#)

抜粋版です。目次と
ページは一致しま
せん

参考

- 雷センサ活用事例

[33](#)



Pi運用マニュアル

1. Raspberry Piについて

既に全世界で1000万台以上販売された手のひらサイズのコンピュータです。
LinuxベースのRasbianOSで動作しております。

2. Linux基本コマンド

① システム関係

- 起動: 電源を入れると自動で起動します。
- 再起動: # reboot
又は、アプリケーション>ログアウト>再起動; 左上のメニューから
- 終了: # shutdown
又は、アプリケーション>ログアウト>シャットダウン; 左上のメニューから
- ログアウト # logout
又は、アプリケーション>ログアウト>ログアウト; 左上のメニューから
- **日本語／英語の入力切替**: キーボードのctl+jを同時に押します。又は右上のアイコン(右から7個目)からプルダウンで選択



Raspberry Pi

Pi運用マニュアル

2. Linux基本コマンド

② ディレクトリ操作、コピー、移動、削除

root@:~\$ cd /root/Documents ディレクトリの切り替え
 root@:/root/Documents# ls ファイルとディレクトリの表示(表示したら操作したいファイルを右クリックでコピーして操作します)
 root@:~# cp ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへコピー
 root@:~# mv ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへ移動
 root@:~# rm ファイル名 ファイルの削除
 便利な機能 rm -help コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべてのコマンド共通(マイナスを2個とhelp)

③ ユーザ権限、プロセス他

root@:~\$ su - スーパーユーザ(root)に切り替え、パスワードを入力
 root@:~# ps a 現状の動いているプロセスを表示
 root@:~# kill 特定のプロセスを強制終了
 root@:~# apt-get install pkg パッケージのインストールなどに使用
 root@:~# date 日付、時間の設定を行います。
 root@:~# leafpad /etc/network/interfaces インタフェースに記述している内容を変更します。Viよりも使いやすいです。

④ モジュール、usb、メモリ、HDDなどの表示

root@:~# lsmod linuxのモジュールリスト表示
 root@:~# lsusb usbのデバイス表示
 root@:~# free -mt メモリ使用状態表示
 root@:~# df -h HDD(マイクロSD)の使用状態表示

pi@raspberrypi:~\$ sudo
 も同様のコマンドになります

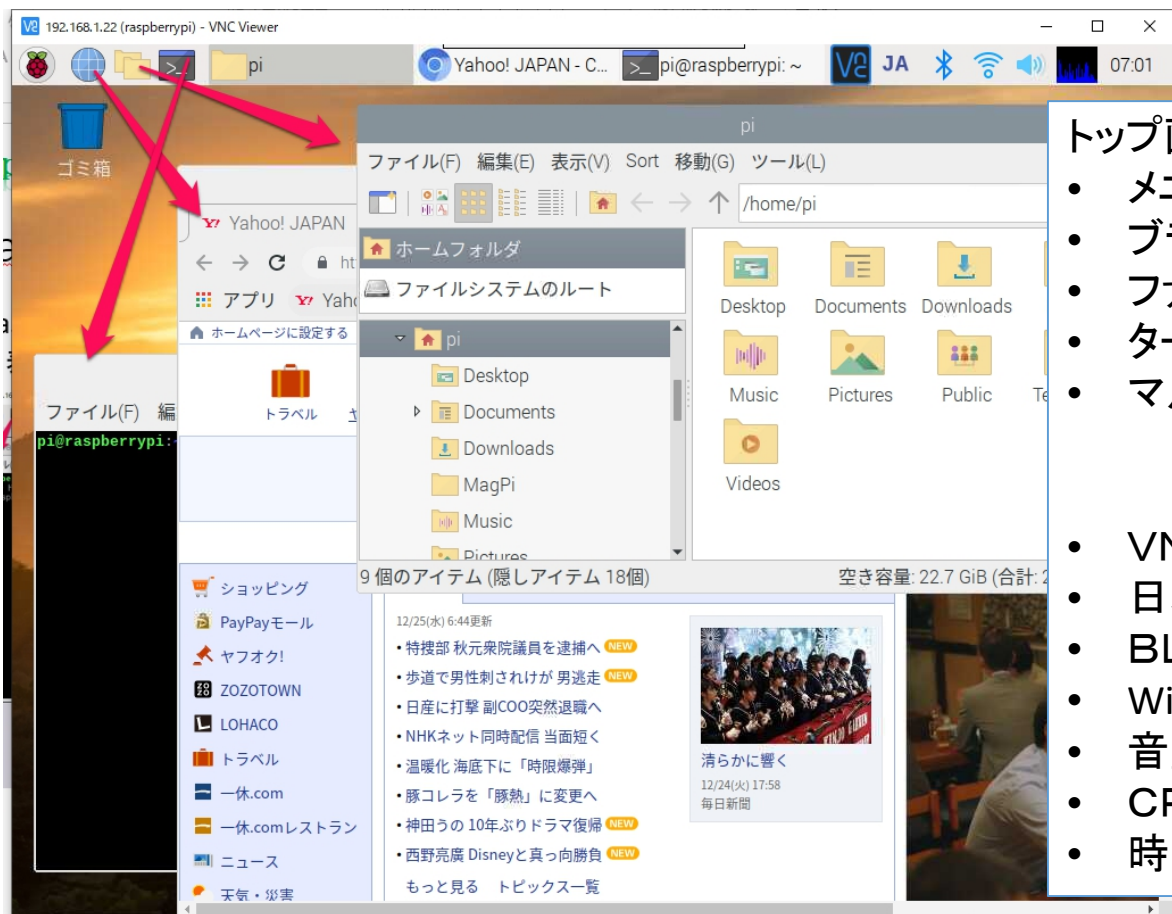


RaspberryPi運用マニュアル

3. Raspberry Piの基本操作

① 表示画面と内容

デスクトップ上によく使うコマンド.txtがあります。
コピーして使ってください



トップ画面(上段のタスクバーで選択)

- メニュー
- ブラウザ
- ファイルマネージャ
- ターミナル
- マルチ画面選択

- VNC
- 日本語入力
- BLE
- WiFi
- 音量
- CPU使用率
- 時刻



RaspberryPi運用マニュアル

4. 日常運用

① セキュリティ対策(アンチウイルス更新、スキャン)

- アンチウイルス対策として無料のclamAVをインストールしてます。
- 手動での運用を基本としています。

パターンファイル更新

手動スキャン時に更新されます

手動でスキャン

\$ sudo clamscan --infected --remove --recursive
自動化可能ですが、バックグラウンドで重くなる可能性大。コマンド入力後約5分位かかります。

```
pi@raspberrypi: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
ERROR: /var/log/clamav/freshclam.log is locked by another  
ERROR: Problem with internal logger (UpdateLogFile = /var/  
og).  
root@raspberrypi: ~# leafpad /etc/clamav/freshclam.conf  
root@raspberrypi: ~# freshclam  
ClamAV update process started at Fri J  
main.cvd is up to date (version: 57, sigs: 4218790, f-level: 60, builder: mishh  
ammer)  
daily.cvd is up to date (version: 21862, sigs: 394456, f-level: 63, builder: neo  
)  
bytecode.cvd is up to date (version: 283, sigs: 53, f-level: 63, builder: neo)  
root@raspberrypi: ~# clamscan --infected --remove --recursive  
SCAN SUMMARY  
Known viruses: 4607906  
Engine version: 0.99.2  
Scanned directories: 264  
Scanned files: 2063  
Infected files: 0  
Data scanned: 61.31 MB  
Data read: 49.02 MB (ratio 1.25:1)  
Time: 71.844 sec (1 m 11 s)  
root@raspberrypi: ~#
```

手動でスキャン

② インストール済パッケージの更新リスト、アップグレード

- Linuxの場合は、頻繁に更新が発生します。アップグレードを定期的 to 実施してください。
- 更新前には、バックアップを取ることをお勧めします。特にアップグレードはまれに動作不良、戻せない状態が発生します。自己責任で実施してください。

[illegible]

```
更新リスト取得
$ sudo apt-get update
アップグレード実施
$ sudo apt-get upgrade
```

必ず実施前にバックアップ

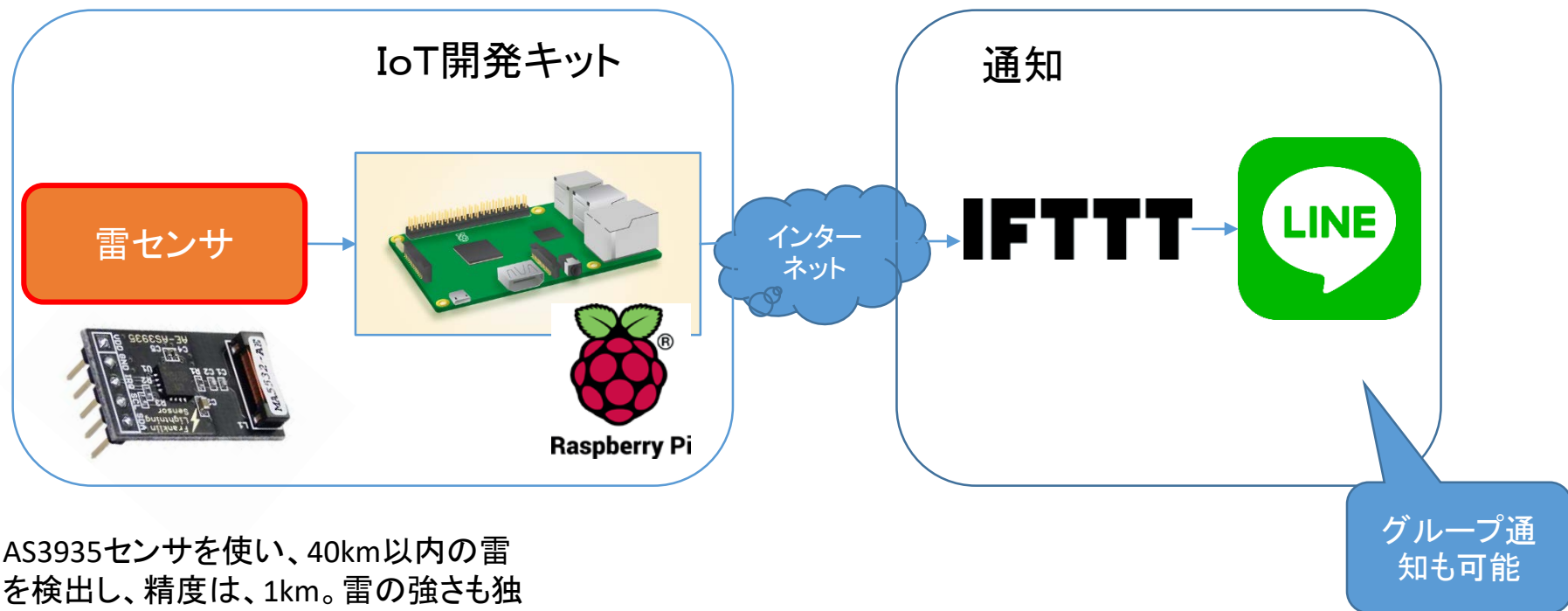
IoT開発

①. メニュー

- 雷センサ 全体構成
- 雷センサ接続、設定
 - AS3935仕様
 - AS3935雷センサとRaspberryPiを接続
 - AS3935の校正
- 単体試験
 - AS3935との単体試験
- IFTTT設定
 - IFTTTを使ってLineへの通知を設定
 - Lineグループへの通知設定
- Line接続
 - AS3935からLineまでの接続確認

IoT開発

②雷センサ 全体構成



AS3935センサを使い、40km以内の雷を検出し、精度は、1km。雷の強さも独自に判定します。500KHzの周波数で雷のノイズを測定します。

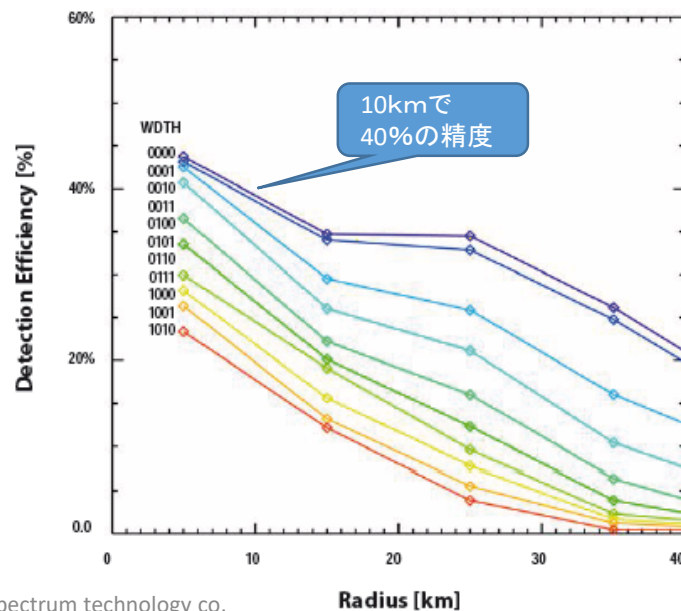
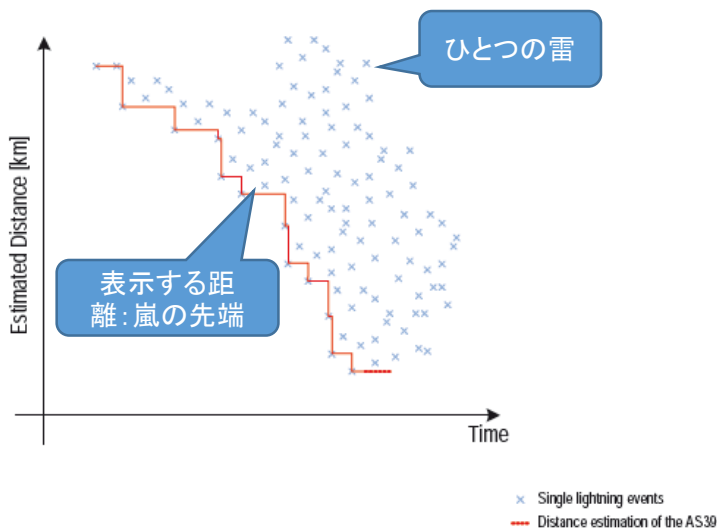
IoT開発

③雷センサ接続、設定

AS3935仕様

- <https://www.meteororange.fr/scripts/AS3935.pdf>
- 雷の検知は、左図；距離の遠いところから（40Km）から検知してゆく、雷の距離ではなく、嵐の最先端までの距離を自動測定。
- 検出の有効性、は右図。WDTHは信号の強さ。雷の信号が弱いと精度が下がるため、屋外の設置を推奨

Detection Efficiencies vs. Distance for Different Settings for WDTH, if SREJ=0000



IoT開発

③雷センサ接続、設定

AS3935仕様

- 雷の強さは、21ビット(max:2,097,152):0x04,0x05,0x06[4:0]で表示。相対値。
- 距離は、5ビット(最大;63):0x07[5:0]
- AFE:アンテナ前段利得:屋内の方が利得を多く入れるが、雑音も同時に利得する。基本は屋外設置にしている。屋内に設置する場合は、プログラム変更。また校正用のプログラムは屋内で設定。

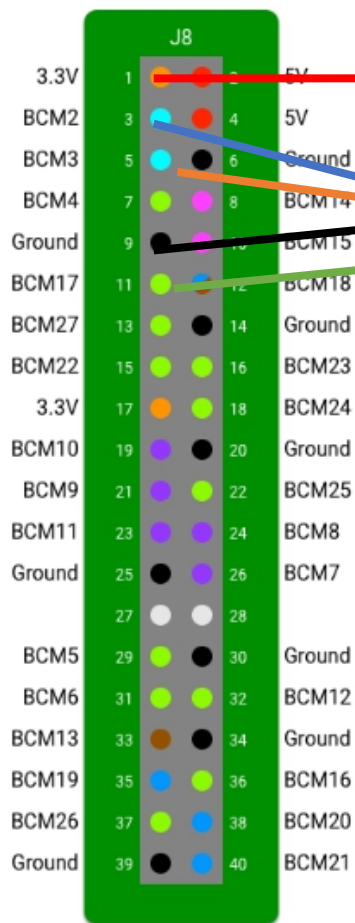
Register Table

Register #	7	6	5	4	3	2	1	0
0x00	Reserved		AFE_GB					PWD
0x01	Reserved	NF_LEV				WDTH		
0x02	Reserved	CL_STAT	MIN_NUM_LIGH			SREJ		
0x03	LCO_FDIV		MASK_DIST	Reserved	INT			
0x04	S_LIG_L							
0x05	S_LIG_M							
0x06	Reserved			S_LIG_MM				
0x07	Reserved		DISTANCE					
0x08	DISP_LCO	DISP_SRCO	DISP_TRCO	Reserved	TUN_CAP			
0x3A	TRCO_CALIB_DONE	TRCO_CALIB_NOK	Reserved					
0x3B	SRCO_CALIB_DONE	SRCO_CALIB_NOK	Reserved					

IoT開発

③雷センサ接続、設定

Raspberry PiとAS3935の接続



メスオスのジャンパで接続
Pi ブレッドボード AMG8833

1	+	VDD
3	-----	SDA
5	-----	SCL
9	-----	GND
11	-----	IRQ

ピンをハンダ付けしてください



ブレッドボード
は添付されて
いません

IoT開発

③雷センサ接続、設定

AS3935校正

- Piコンソールで指定のフォルダに移動

```
$ cd /home/pi/Documents/ar
```

```
$ python3 calibrate.py
```

- アンテナのチューニングを行います。500KHzに近い値に設定
- デフォルトは4:31.32kHz(デジタルマルチテスタで測定) $\times 16 = 501.12\text{KHz}$
- 急ぎの場合は、プログラムを動作させ、デフォルトのまま次へ。

```
$ cd /home/pi/Documents/ar  
$ python3 calibrate.py
```

全てpython3
で動作します



IRQとGND間に接
続しHz測定

IoT開発

④雷センサ単体試験

AS3935単体試験

- Piコンソールで指定のフォルダに移動

```
$ cd /home/pi/Documents/ar
```

```
$ python3 lightning.py
```

- 単体試験がスタート、雷センサが動作時に、画面及びlightningフォルダに出力
- 時刻、距離(km)、強さの順。

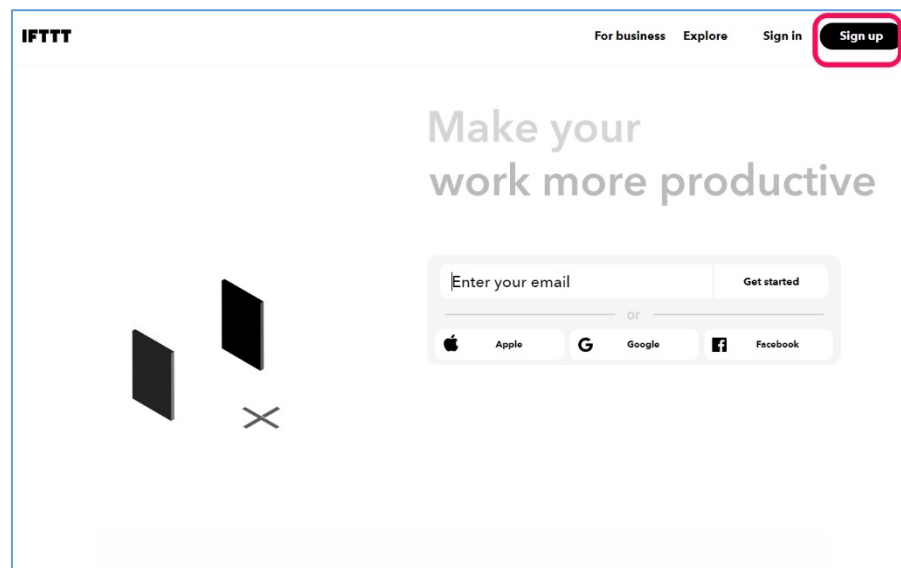
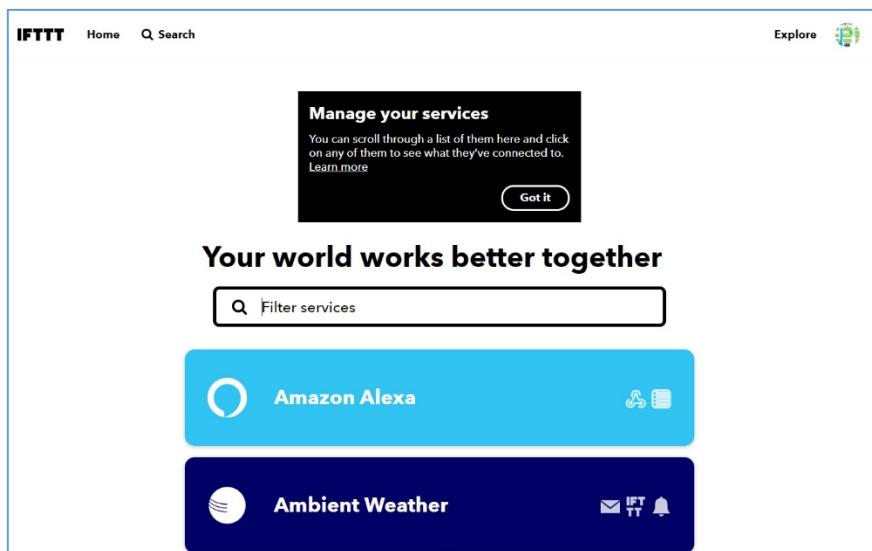
```
$ cd /home/pi/Documents/ar  
$ python3 lightning.py
```


IoT開発

⑤IFTTT設定

IFTTTとは

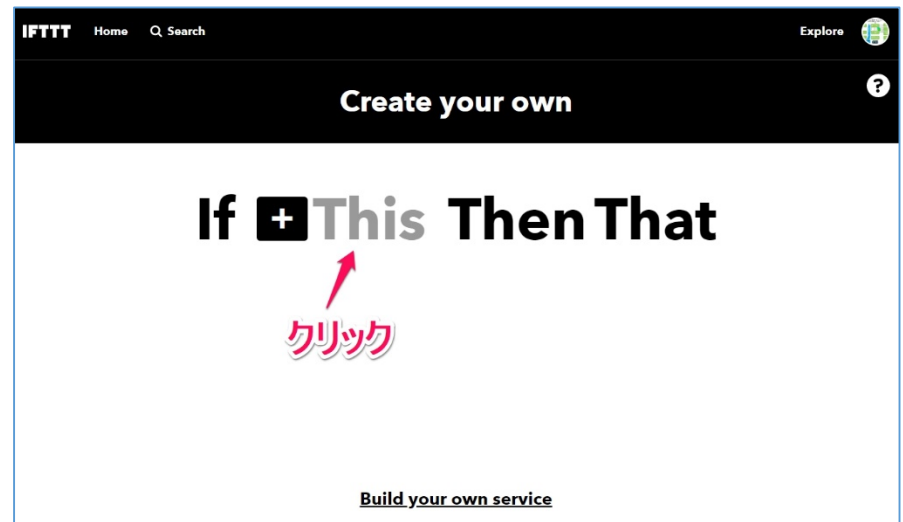
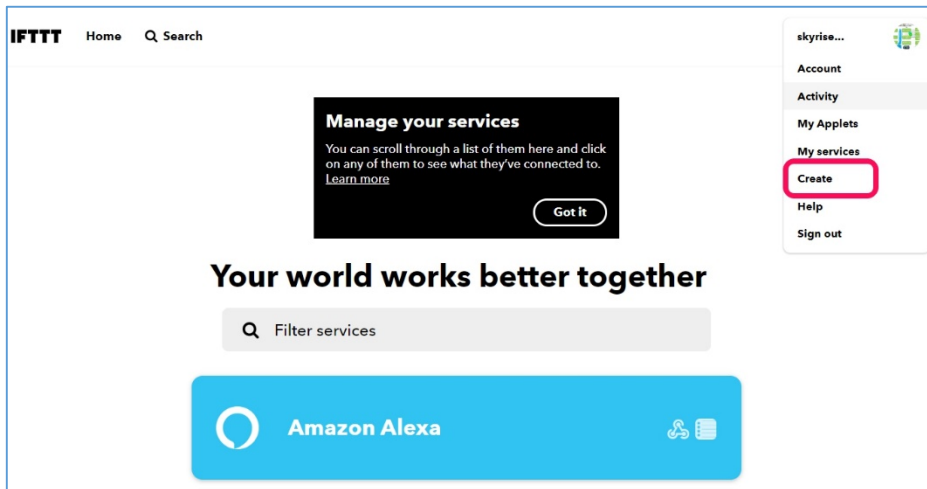
- 「レシピ」と呼ばれる個人作成もしくは公に共有しているプロフィールを使って数あるWebサービス（Facebook、Evernote、Weather、Dropboxなど）同士で連携することができるWebサービスである。
- <https://ifttt.com/>
- IFTTT登録
 - メールとパスワードで登録



IoT開発

⑤IFTTT設定

- IFTTT設定
 - レシピを作成して行きます。
 - Create
 - If thisをクリック



IoT開発

⑤IFTTT設定

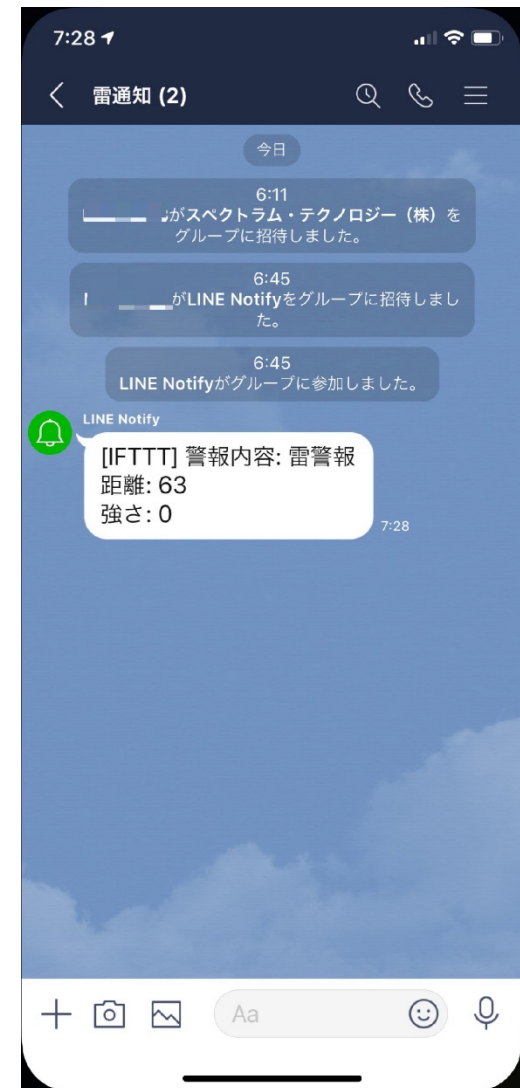
- Line設定:スマホ側
 - トーク>Line notifyを開く
 - 確認



IoT開発

⑤IFTTT設定

- Line設定:スマホ側
 - Line notifyに招待されているのを確認
 - 雷センサのプログラムを動作させ通知
 - グループ側に通知されているのを確認



IoT開発

⑤IFTTT設定

- Line連携試験

```
$ cd /home/pi/Documents/ar
```

```
$ python3 ifttt_as.py
```

- IFTTT連携開始、終了と出力

- Line側にもメッセージ受信、雷警報、0,0と出ればOK

```
$ cd /home/pi/Documents/ar
```

```
$ python3 ifttt_as.py
```

```
pi@raspberrypi: ~/Documents/ar
ファイル(F) 編集(E) タブ(T) ヘルプ(H)

run_cap= 14
run_cap= 15
register0x08= 0b1111
pi@raspberrypi:~/Documents/ar $ python3 lightning_inform.py
['2020/06/14 10:31:09', 'Waiting for lightning', '63', '0']
CTraceback (most recent call last):
  File "lightning_inform.py", line 69, in <module>
    time.sleep(1.0) #(10)
KeyboardInterrupt
pi@raspberrypi:~/Documents/ar $ ls
Pi_AS3935.py  calibrate.py  lightning  lightning_inform.py
__pycache__  ifttt_as.py  lightning.py
pi@raspberrypi:~/Documents/ar $ python3 ifttt_as.py
IFTTT連携開始
IFTTT連携終了
pi@raspberrypi:~/Documents/ar $ python3 calibrate.py
set indoors
set noise floor to 0
register0x08= 0b0
register0x08= 0b10000000
Please measure the frequency of the IRQ pin by a multi-meter
```



IoT開発

⑥Line接続

- AS3935からLine連携

\$ cd /home/pi/Documents/ar

必ず事前にcalibrate.pyを実施してください。

\$ python3 lightning_inform.py

- 雷センサからLine連携開始
- Line側にもメッセージ受信、雷警報、63.0と出ればOK

\$ cd /home/pi/Documents/ar
\$ python3 lightning_inform.py

```

pi@raspberrypi: ~/Documents/ar
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
register0x08= 0b0
register0x08= 0b10000000
Please measure the frequency of the IRQ pin by a multi-meter
tune_cap is changed every 10 seconds
tun_cap= 0
tun_cap= 1
tun_cap= 2
tun_cap= 3
tun_cap= 4
tun_cap= 5
tun_cap= 6
tun_cap= 7
tun_cap= 8
tun_cap= 9
tun_cap= 10
tun_cap= 11
tun_cap= 12
tun_cap= 13
tun_cap= 14
tun_cap= 15
register0x08= 0b1111
pi@raspberrypi:~/Documents/ar $ python3 lightning_inform.py
'2020/06/14 10:38:38', 'Waiting for lightning', '63', '0']
    
```

