

PCとつないで楽しむ電子工作。 IoTへの第一歩

その2

モウフカブール
大澤文孝



Who Am I?

大澤文孝

テクニカルライター／システムエンジニア／プログラマ／インフラエンジニア。
著書はもうすぐ100冊。100冊記念パーティーやりたい(けど反対されている)。

主な著書



もうすぐ100冊
パーティー
やりたい！





新刊①



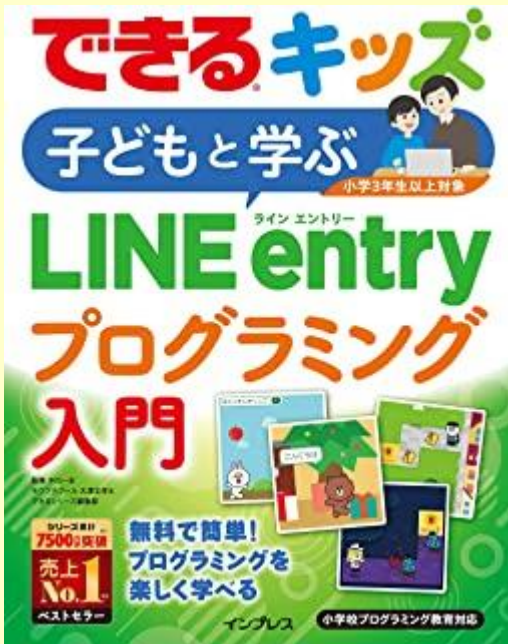
こちらは
もうすぐ出ます

Google
Cloud本





新刊②



こども向けプロ
グラミング入門





近刊



清水美樹さんと一緒に手がけた
翻訳書



学術書。
翻訳モノです。

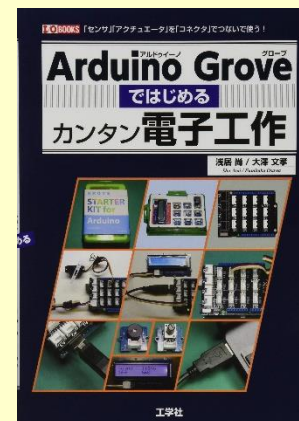
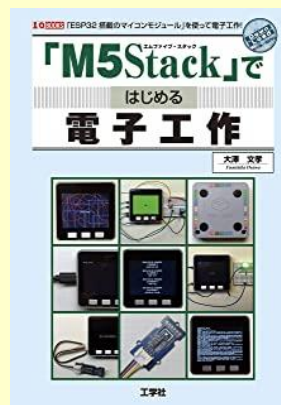




今日のお話は電子工作



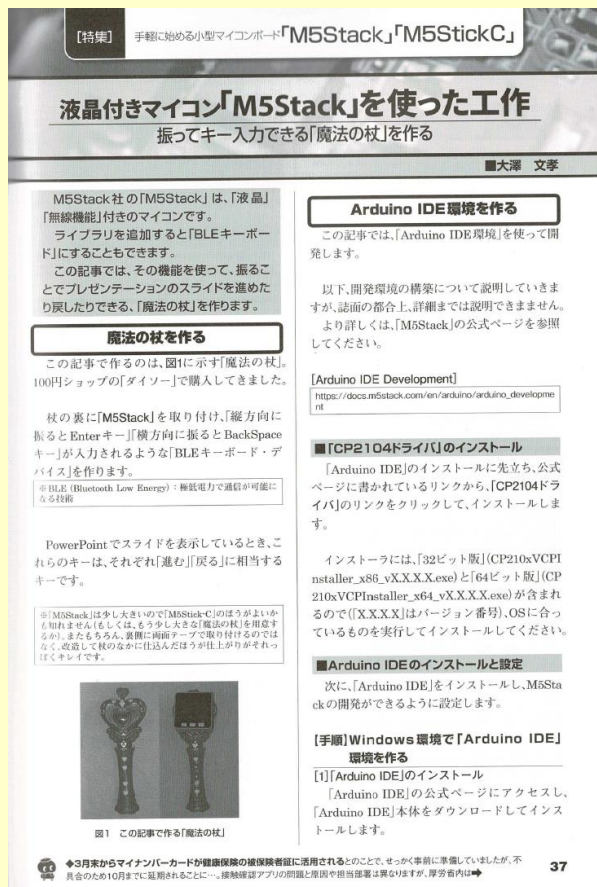
毎月18日
発売





月刊I/O 6月号

振るとプレゼン
テーションが進む





GitHubで ライブラリが公開されているおかげ

こうした短いコードで書けるのは、ライブラリ作者さんのおかげ。

BLE、Wi-Fi、各種センサー、グラフィック処理、さまざまなライブラリが、オープンソースで提供されている。

皆様に感謝！



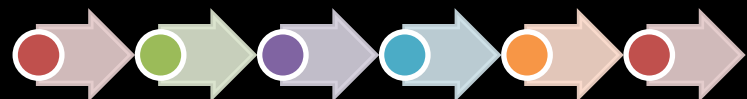
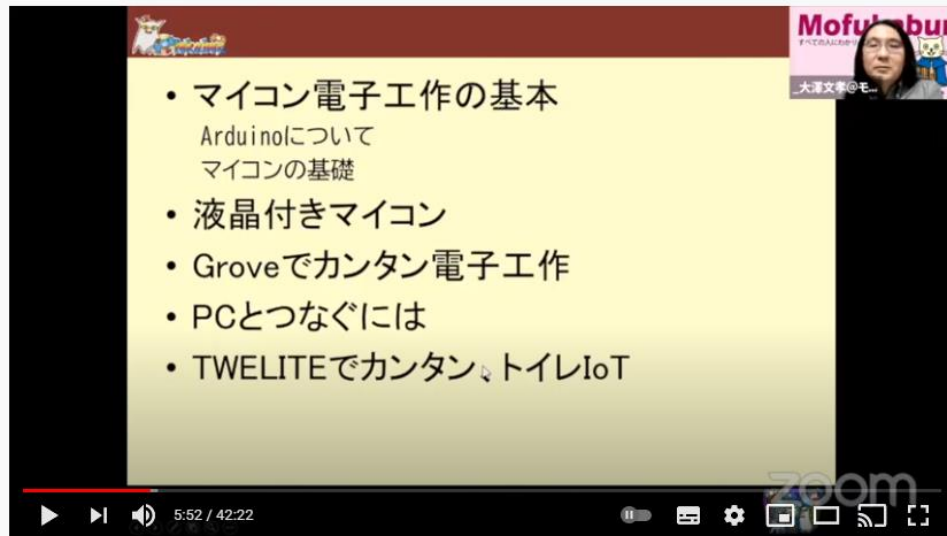


前回までのお話

<https://youtu.be/bnECjIC0mUo>

OSC2021
OnlineNagoya
IoT

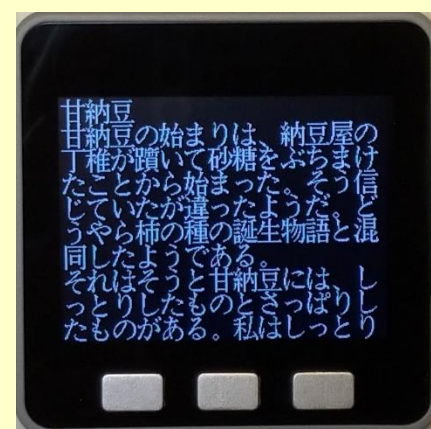
で検索



Presented by
Mofukabur.いんく



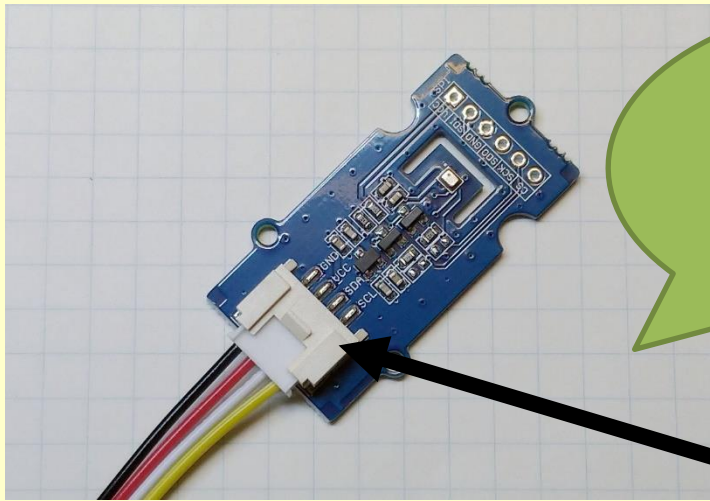
- ・液晶付きマイコンのおかげで、何か外につながなくても済むのでカンタン





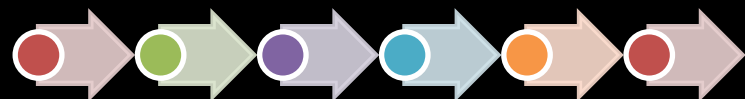
• Groveでセンサーなどの利用もカンタン

- コネクタでつなぐだけ
- ライブラリがオープンソースで公開されている



ハンダ付けからの
解放！
オープンソースの
ライブラリ

4ピンの専用コ
ネクタで接続





IoTへの
第一歩

• Wi-Fi対応

マイコンをWebサーバーにできる。

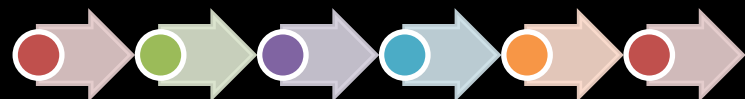
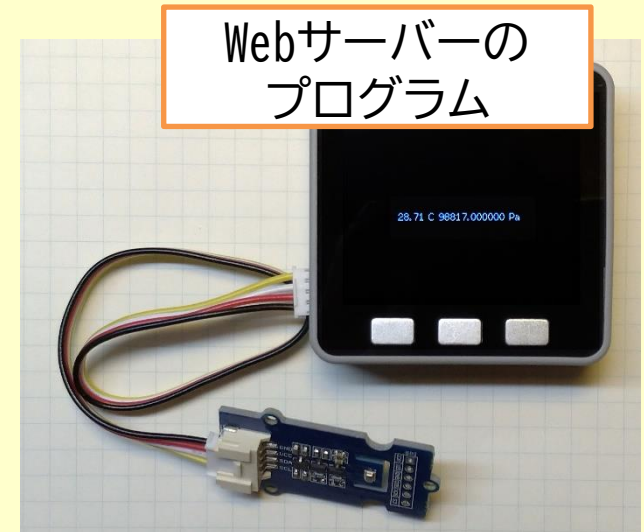
ブラウザでアクセスして、センサーの値を取得できる。



ブラウザでアクセス
(<http://XXX.XXX.XXX.XXX>)



温度を含むHTMLページ
を返す



Presented by
Mofukabur.いんく





HTMLを返すコード

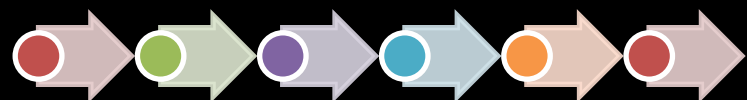
```
server.on("/", HTTP_GET, []() {  
    // 温度・気圧の取得  
    float t = bmp280.getTemperature();  
    float p = bmp280.getPressure();  
  
    char msg[256];  
    sprintf(msg, "%2.2f C %f Pa", t, p);  
    server.send(200, "text/html",  
                "<html><head><meta charset='utf-8'></head>"  
                "<body><h1>温度</h1>" +  
                String(msg) +  
                "</body></html>"  
    );  
});
```

センサの値を埋め
込んでHTMLを返
すように作る





アジェンダ

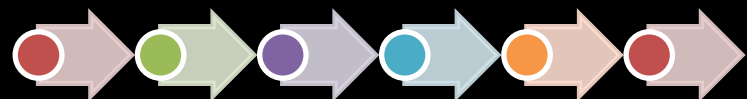


Presented by
Mofukabur.いんく





- 電源のお話
- インターネットのお話





参考書



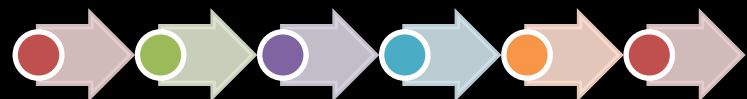
TWELITE PALではじめるクラウド電子工作

工学社

TWELITE PALという無線センサを使って、データ収集するというお話。

送信側はラズパイ。クラウドにはAWSを利用。DynamoDBやサーバレスの話まで。

プレゼントで
当たるかも！？

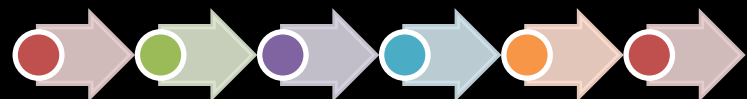


Presented by
Mofukabur.いんく





電源のお話



Presented by
Mofukabur.いんく





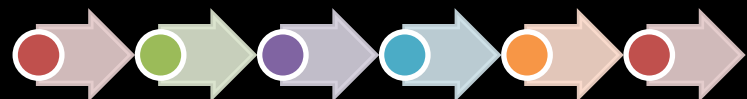
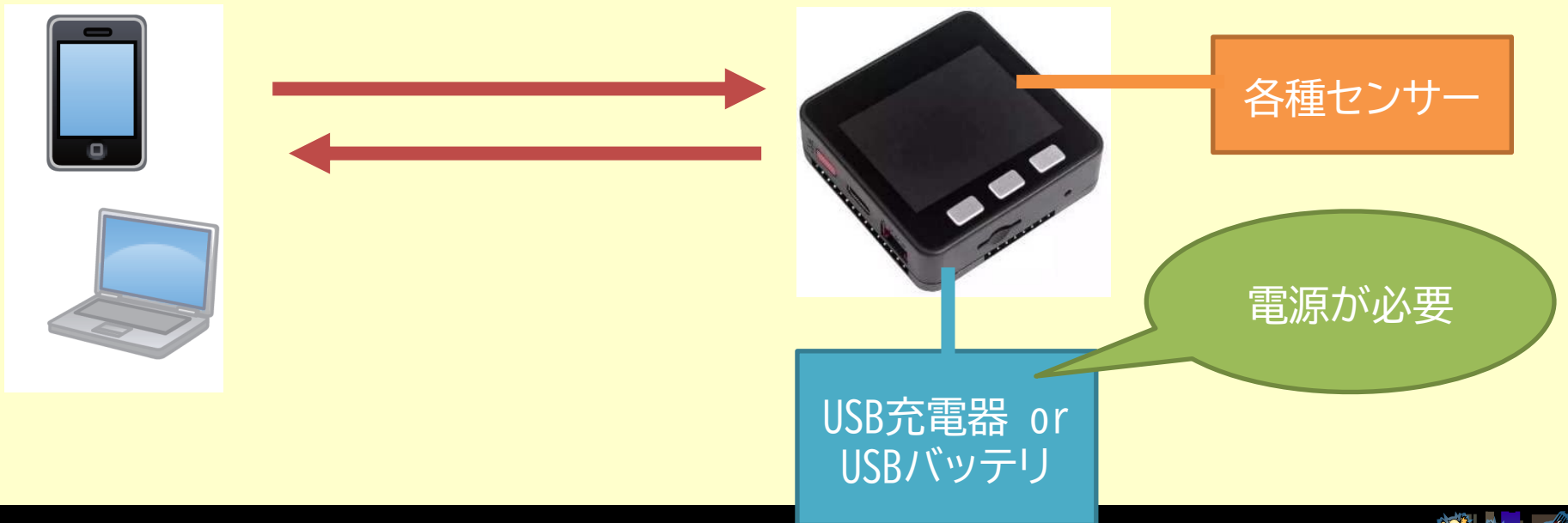
・マイコンの電源

USB給電の5Vのマイコンが多い。

スマホの充電器で動かす？ USBバッテリーで動かす？

センサを置きたいところに、電源があるとは限らない

→乾電池で動くマイコンが嬉しい





・省電力センサの活用

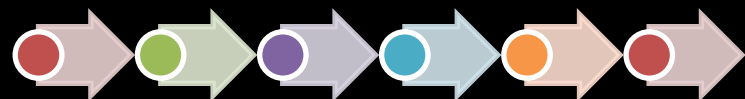
・BLEセンサ



コイン電池で
動く

TIのSensorTag

<https://www.tij.co.jp/tool/jp/CC2650STK>

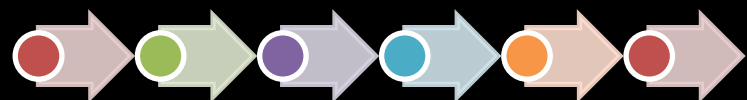
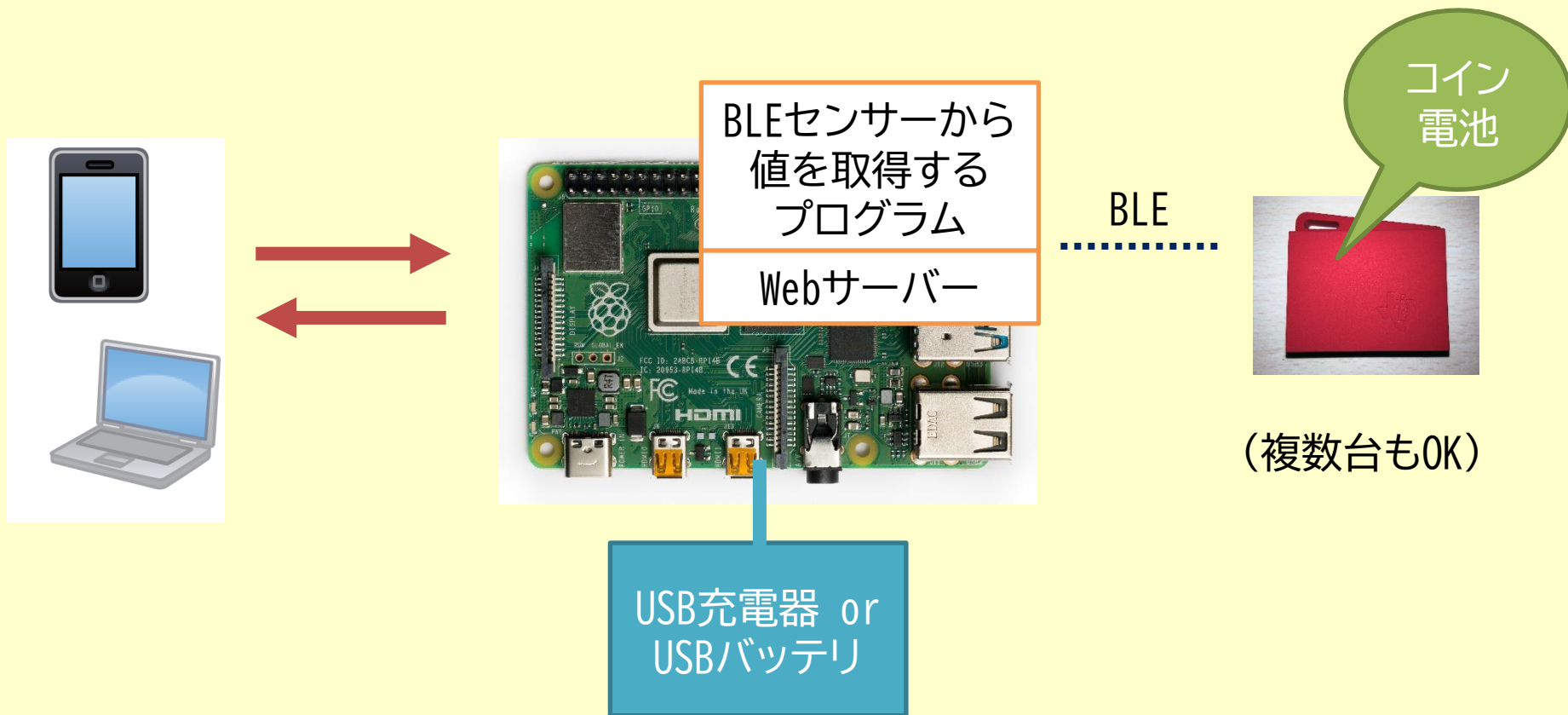


Presented by
Mofukabur.いんく





ラズパイなどでBLEセンサを使う





- ちょっとお高め

SensorTagの場合、6000円ぐらい。
(SwitchBotの温湿度計は2000円ぐらい。
仕様非公開だが解析されてるぽい)



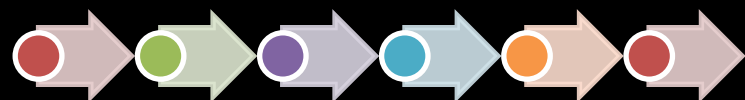
有志によって
解析されて
いる

- プログラムが少し複雑

BLEですし…。

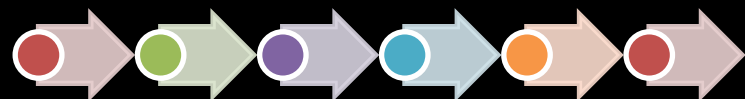
- 種類がそんなにない

いろんなセンサーがあるわけではない





別の無線規格はどう？



Presented by
Mofukabur.いんく

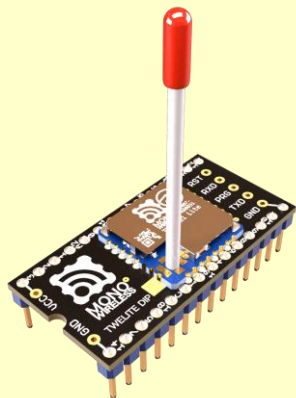




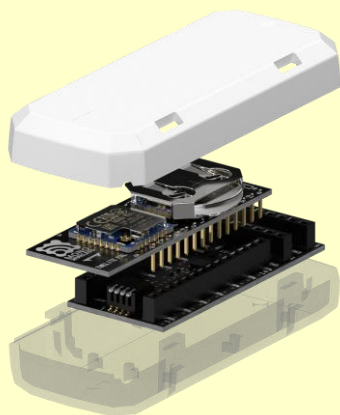
TWELITE ?

モノワイヤレス社の無線マイコン。DIP型(ICのかたち)のTWELITE DIPやセンサーのPAL、小型のCUE、パソコンと接続するMONOSTICKなどがある。

乾電池・コイン電池で動作OK！



DIP



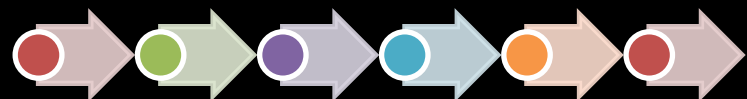
PAL



CUE

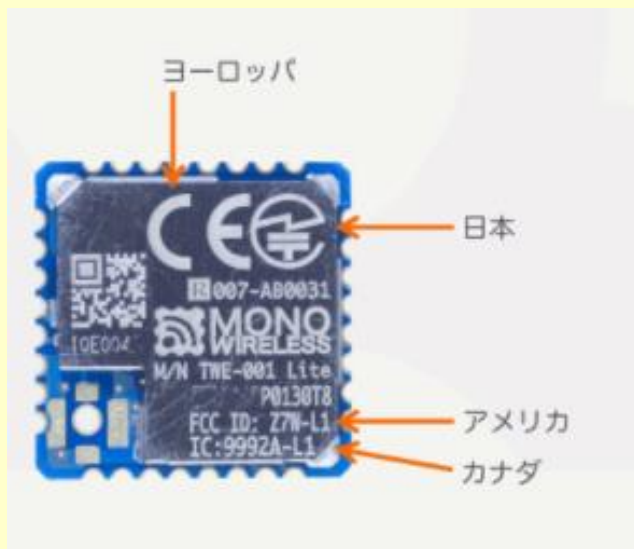


MONOSTICK





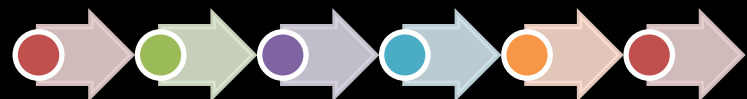
技適OK。2.4GHz帯。802.15.4



対応国一覧

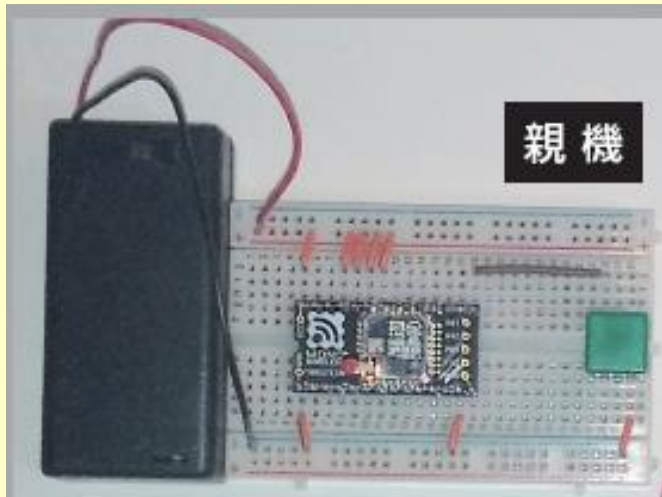
オーストリア	ベルギー	ブルガリア
クロアチア	キプロス	チェコ
デンマーク	エストニア	フィンランド
フランス	ドイツ	ギリシャ
ハンガリー	アイルランド	イタリア
ラトビア	リトアニア	ルクセンブルク
マルタ	オランダ	ポーランド
ポルトガル	ルーマニア	スロバキア
スロベニア	スペイン	スウェーデン
イギリス	アイスランド	ノルウェー
リヒテンシュタイン	スイス	トルコ
アメリカ	カナダ	中国
ベトナム	日本	

※ 対応国は法令の改定等で変わる場合がございますので、ご使用時に最新情報をご確認ください。

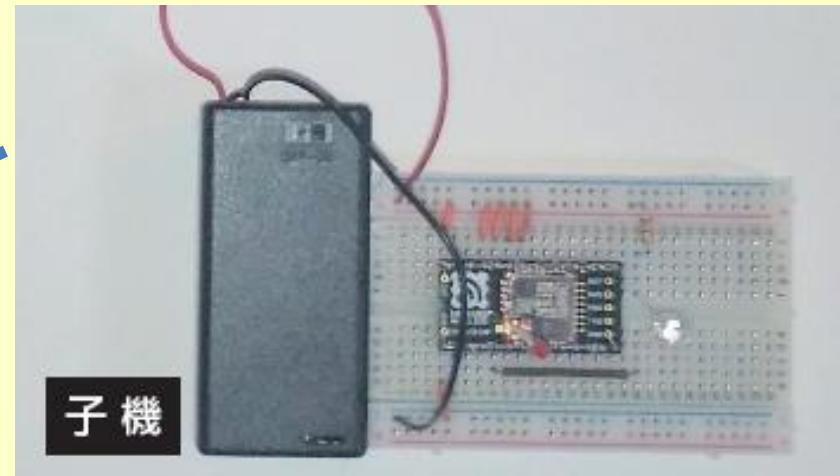


基本機能

2台買ってきて、片方にスイッチ付ける、もう片方にLED付ける。
電池をつなげば、片方をオンにしたとき、LEDが光る。
到達距離は最大1kmとされている。



スイッチオン



LED光る

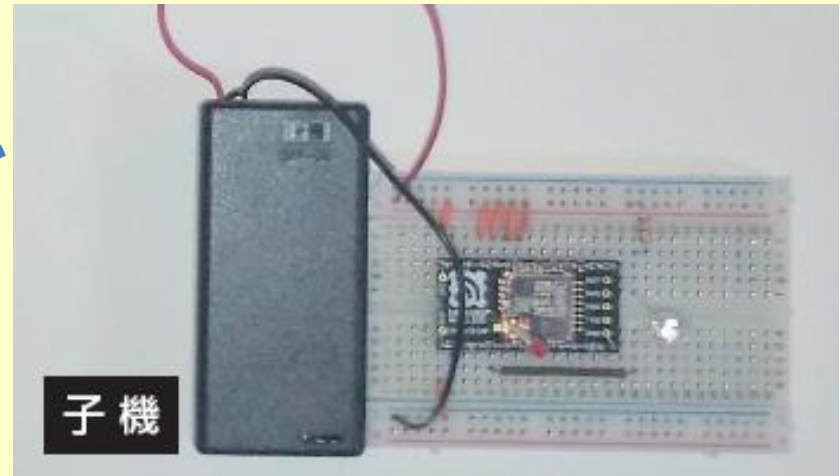


MONOSTICKでPCから制御

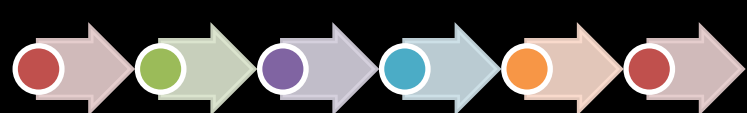
PCにMONOSTICKを使う。COMポート(シリアル)として見える。
シリアルに文字列を送信すると、無線信号を出して、LEDとか
光らせる。



コマンド送信



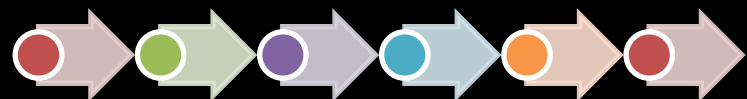
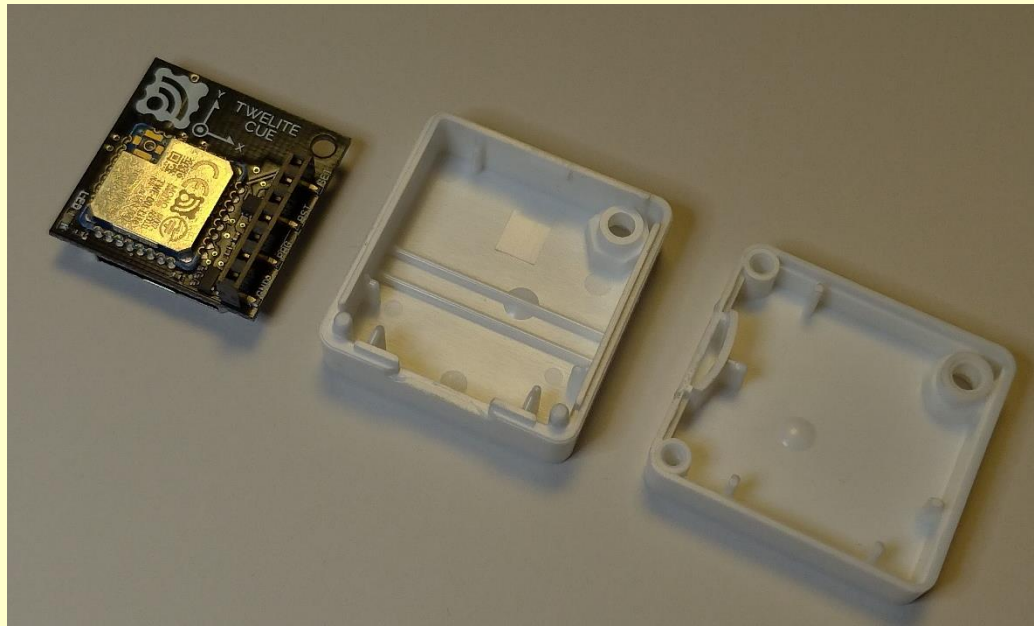
LED光る





TWELITE CUE

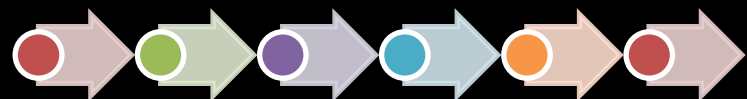
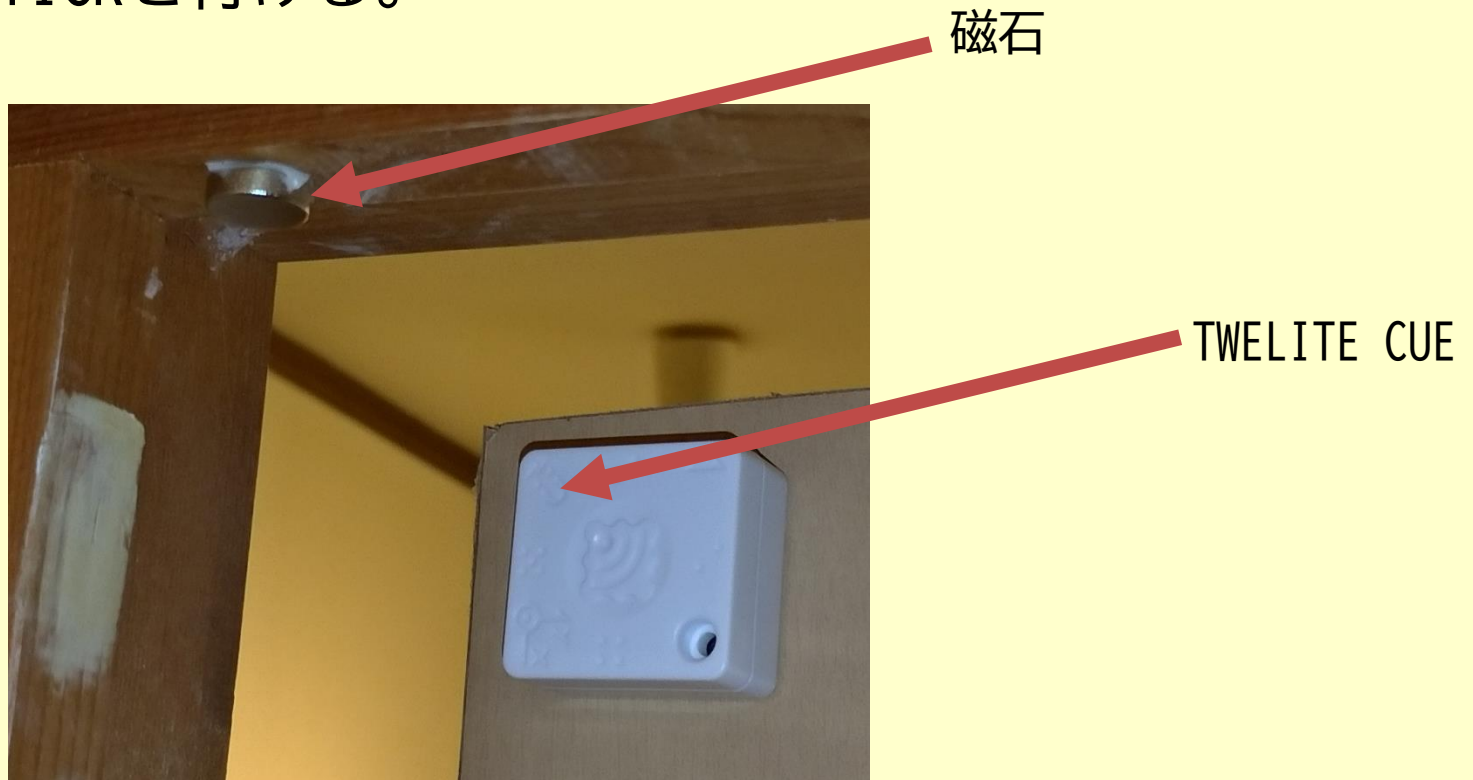
マグネットスイッチと加速度センサを搭載したTWELITE。
MONOSTICKでデータを読み取れる。





トイレIoTすぐ作れる

ドアにTWELITE CUEとマグネットを取り付ける。パソコンにはMONOSTICKを付ける。





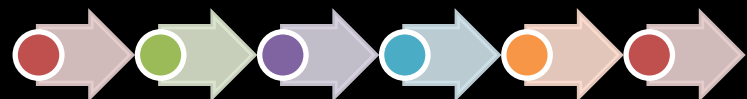
こんなプログラムでOK

```
import sys
sys.path.append('./MNLib/')
from apppal import AppPAL

port = 'COM3' // 環境に合わせる
PAL = AppPAL(port = port)

# データの取得
while True:
    if PAL.ReadSensorData():
        # 届いているなら、値を取得して表示
        data = PAL.GetDataDict()
        if data['HALLIC'] < 10:
            print("開閉発生");

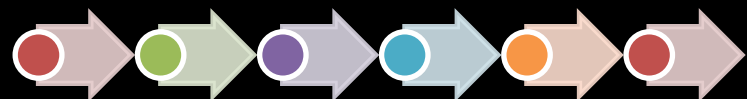
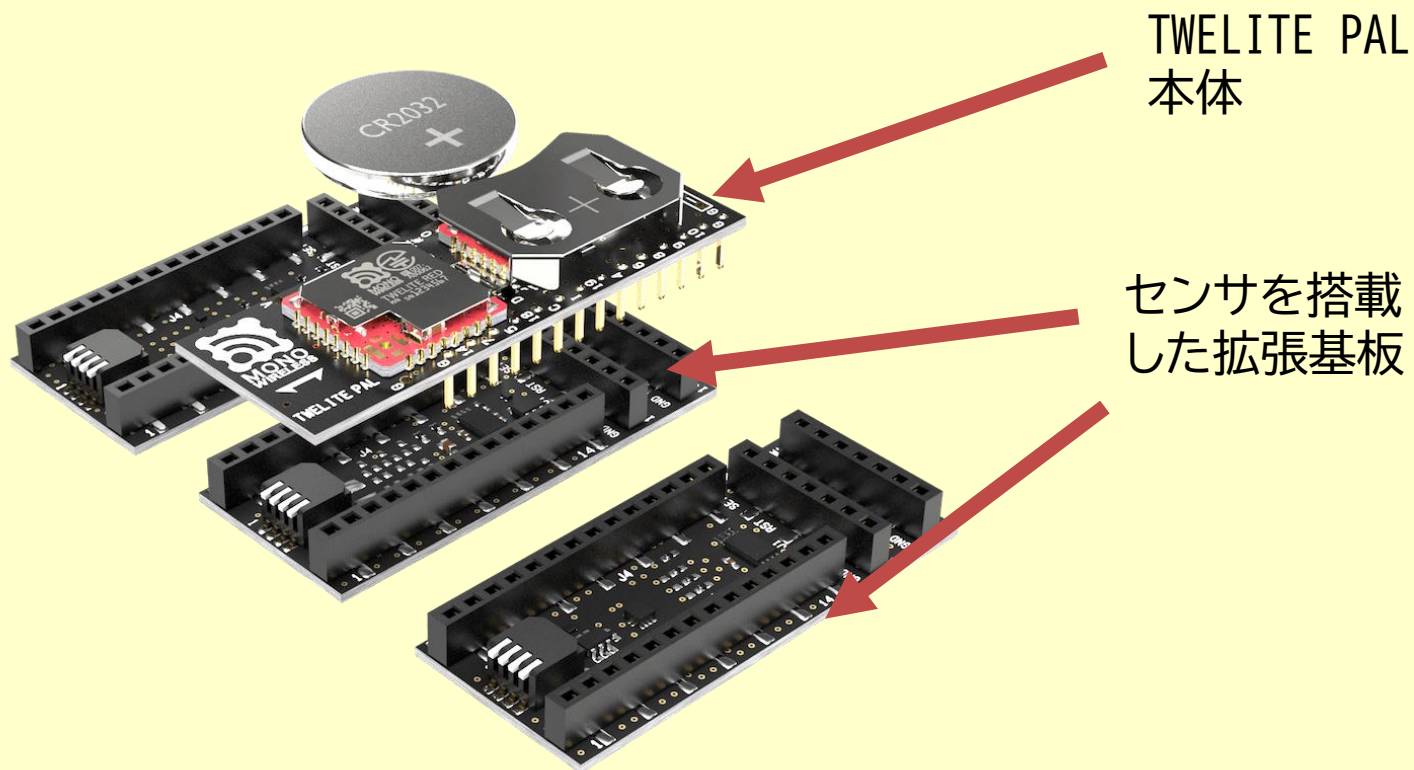
del PAL
```





TWELITE PAL

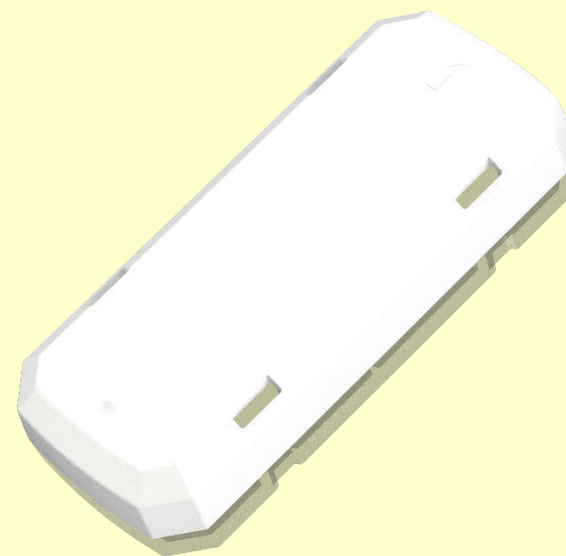
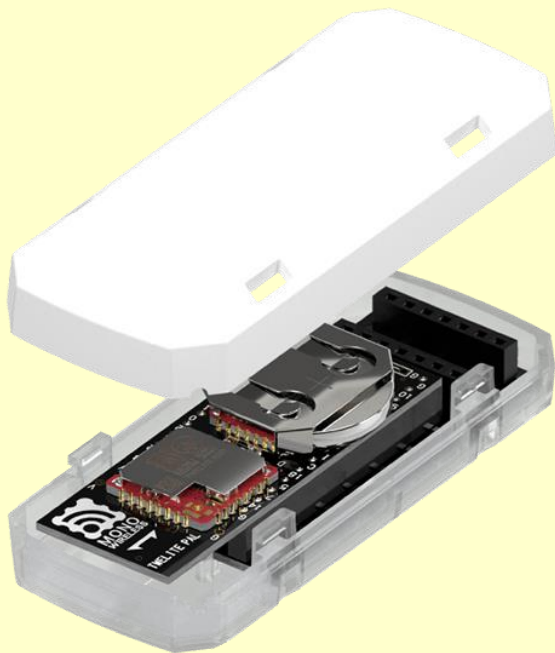
センサーを亀の子のように取り付けられるTWELITE。コイン電池で動作。



Presented by
Mofukabur.いんく



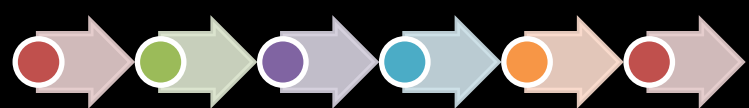
• 別売ケース





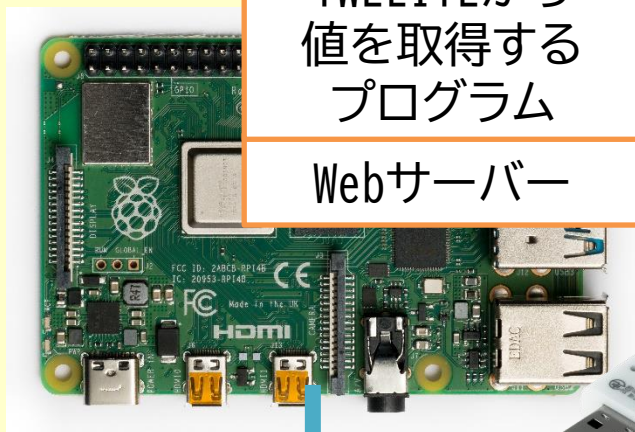
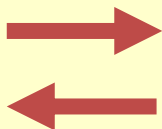
TWELITE PALの種類

- 開閉センサーパル
マグネットセンサー
- 環境センサーパル
温度・湿度・照度センサー
- 動作センサーパル
加速度センサー(向きや動きがわかる)
- 通知パル
高輝度カラーLED、加速度





TWELITE PALを使った 温湿度・照度センサー



TWELITEから
値を取得する
プログラム
Webサーバー

USB充電器 or
USBバッテリー



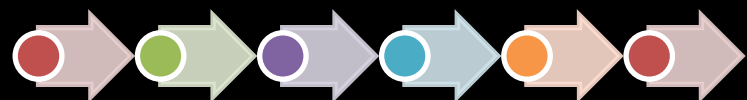
MONOSTICK

独自無線

コイン
電池



環境センサ
パル
(複数台もOK)



Presented by
Mofukabur.いんく

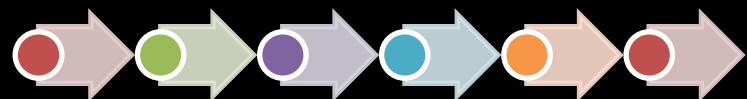




• お値段はそこそこ

MONOSTICK	3000円ぐらい
環境センサーパル	1600円ぐらい
TWELITE PAL	1600円ぐらい
ケース	500円ぐらい

※TWELITEやMONOSTICKには、標準の「BLUE」と電波がよく飛ぶ「赤」のモデルがあります。上記価格は標準BLUEの場合



Presented by
Mofukabur.いんく





シリアルなのでプログラムはカンタン

【開閉センサーパル】 ←

```
:80000000C3000082012BAE05808103113008020C1C1130010202D6000000018  
00840←
```

←
【環境センサーパル】 ←

```
:80000000CF0001820120D302808205113008020C0811300102043D050100020  
B8E0102000210DF02030004000000185F38←
```

←
【動作センサーパル】 ←

```
:80000000A80001810B0FBFE01808312113008020BFE1130010204DF15040006F  
D08FE98037015040106FDB0FE98040015040206FE98FF50040015040306FEE0F  
EF8042015040406FEB0FF2003F015040506FEF0FF28035815040606FEA8FFA00  
3B815040706FF10FF70032815040806FE88FF60038015040906FEA8FF6003881  
5040A06FE80FF1003D015040B06FE60FEA003A815040C06FE68FF58035815040  
D06FEC0FF8803A815040E06FEC8FF5003E815040F06FF38FEF803E8A7B0←
```



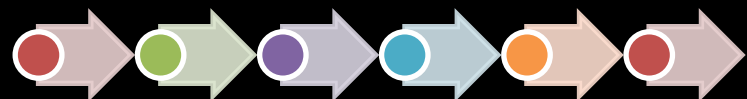


値を読み取るPythonプログラムの例

```
import sys
sys.path.append('./MNLlib/')
from apppal import AppPAL

port = 'COM3'
PAL = AppPAL(port = port)

while True:
    # データが届いているか
    if PAL.ReadSensorData():
        # 届いているなら、値を取得して表示
        data = PAL.GetDataDict()
        print(data)
```



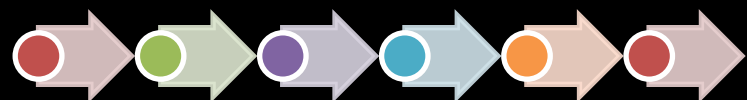


読み込んだ値

←

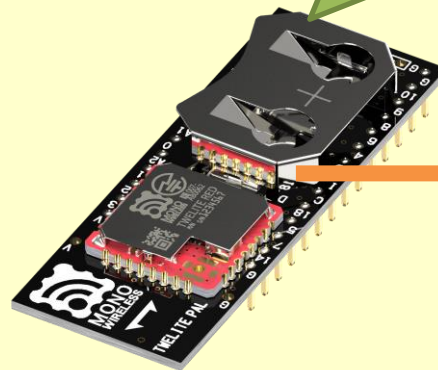
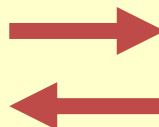
```
{'ArriveTime': -datetime.datetime(2020, -2, -21, -4, -31, -18, -945334),  
  -'LogicalID': -1, -'EndDeviceSID': -'82012BAE', -'RouterSID': -'800000  
00', -'LQI': -189, -'SequenceNumber': -36, -'Sensor': -128, -'PALID': -1,  
  -'PALVersion': -1, -'Power': -2990, -'ADC1': -1087, -'HALLIC': -0}←
```

開閉センサパルの場合。環境センサパルの場合は、Temperature、Humidity、Illuminanceの値が、この構造に含まれる。





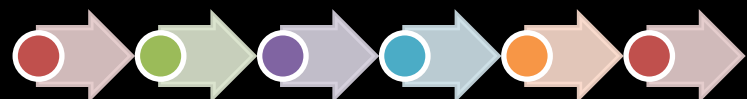
独自のセンサーが作れる



センサを制御するプログラムを自作し、書き込んでおく

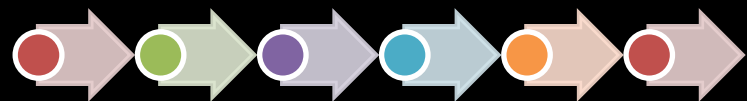
独自のセンサー

TWELITEの標準ROMは、I2Cの制御ができるため、I2Cセンサなら、ROM書き換えなしでPythonなどのプログラムだけでも対応可能





実際に作る場合・・・



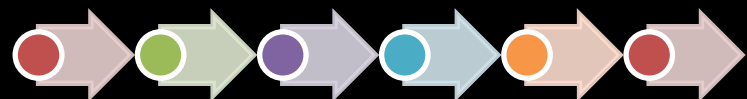
Presented by
Mofukabur.いんく





Webサーバー化する Pythonのプログラム

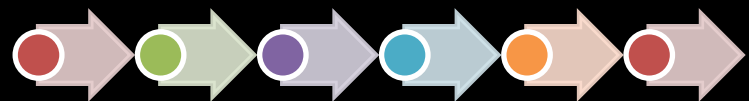
```
from wsgiref.simple_server import make_server
# サーバーの立ち上げ
with make_server('', 80, process) as httpd:
    print("started")
    httpd.serve_forever()
```





HTMLコンテンツを返す

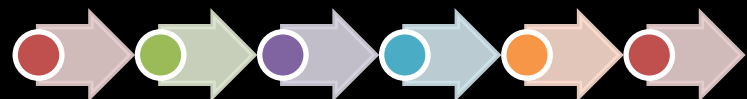
```
def process(env, response):  
    global temperature, humidity, illuminance  
    headers = [('Content-Type', 'text/html; charset=utf-8')]  
    response('200 OK', headers)  
    # Jinja2のテンプレートを読み込む  
    env = Environment(loader=FileSystemLoader('.'))  
    template = env.get_template('index.tpl')  
    # 値を差し込んでHTMLにする  
    html = template.render({  
        'Temperature' : temperature.value,  
        'Humidity' : humidity.value,  
        'Illuminance' : illuminance.value  
    })  
    return [html.encode('utf-8')]
```





HTMLテンプレート

```
<html>
<body>
<h1>環境センサーパル</h1>
<p>温度: {{Temperature}} </p>
<p>湿度: {{Humidity}}%</p>
<p>照度: {{Illuminance}}ルクス</p>
</body>
</html>
```



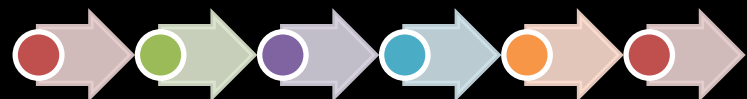


センサからの値取得はマルチプロセスで①

```
from multiprocessing import Value, Process
```

```
port = '/dev/ttyUSB0'
```

```
def sensing(t, h, i):  
    PAL = AppPAL(port = port)  
    while True:  
        if PAL.ReadSensorData():  
            data = PAL.GetDataDict()  
            t.value = data['Temperature']  
            h.value = data['Humidity']  
            i.value = data['Illuminance']
```





センサからの値取得はマルチプロセスで②

共有メモリを作る

```
temperature = Value('d')
```

```
humidity = Value('d')
```

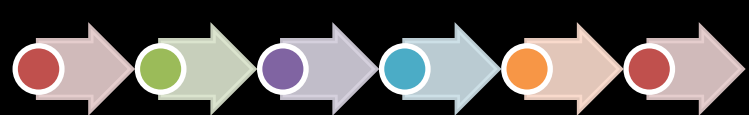
```
illuminance = Value('i')
```

プロセスとして実行

```
process1 = Process(target=sensing,  
    args=[temperature, humidity, illuminance])  
process1.start()
```

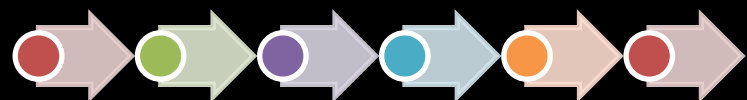
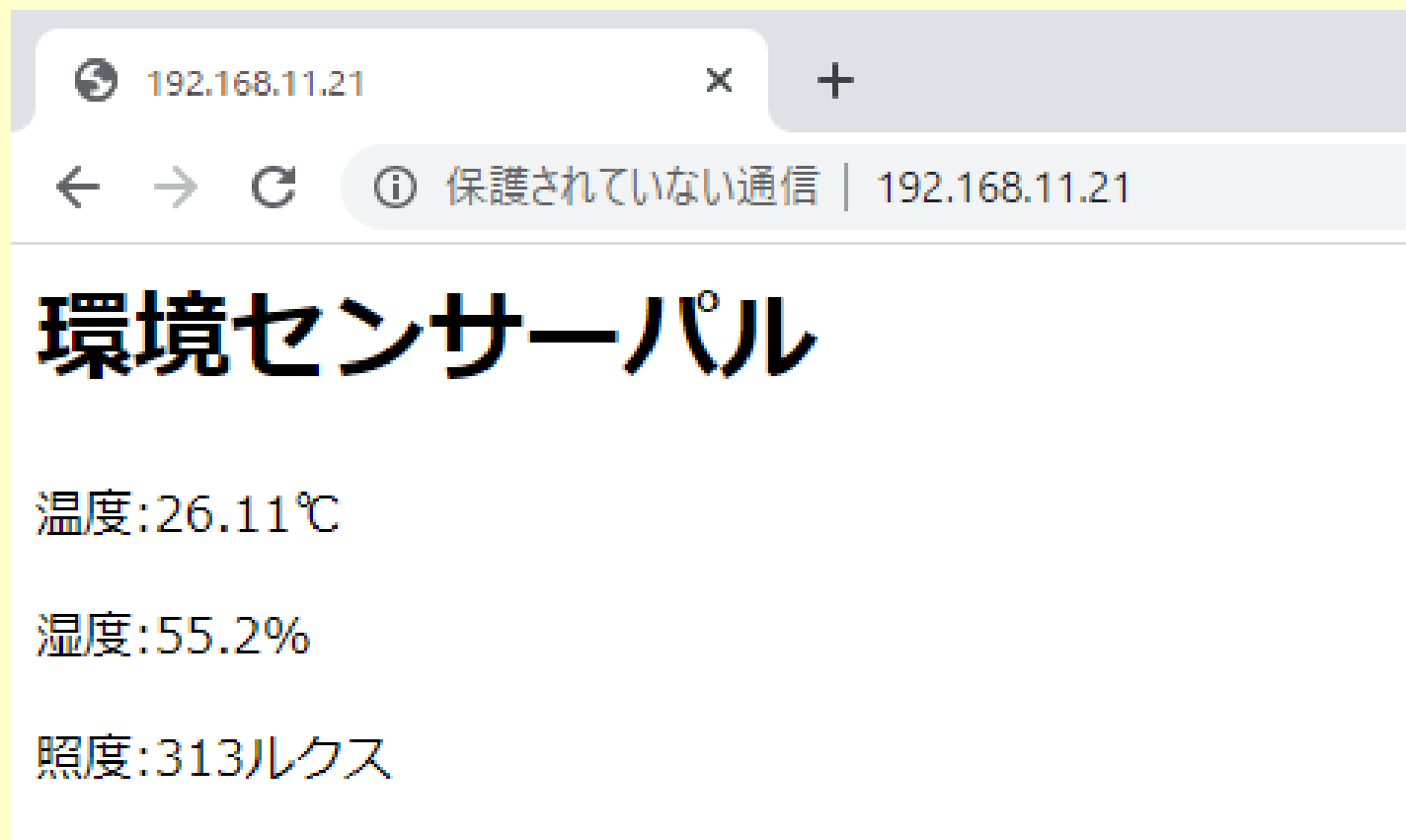


詳しくは書籍で



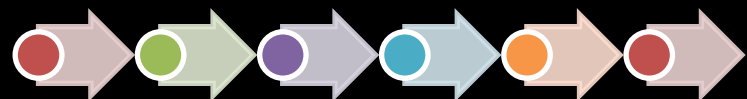


ブラウザから接続すると…





インターネットのお話

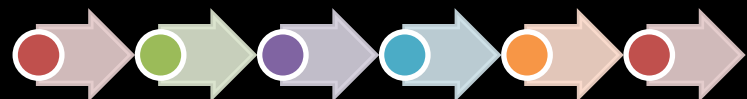
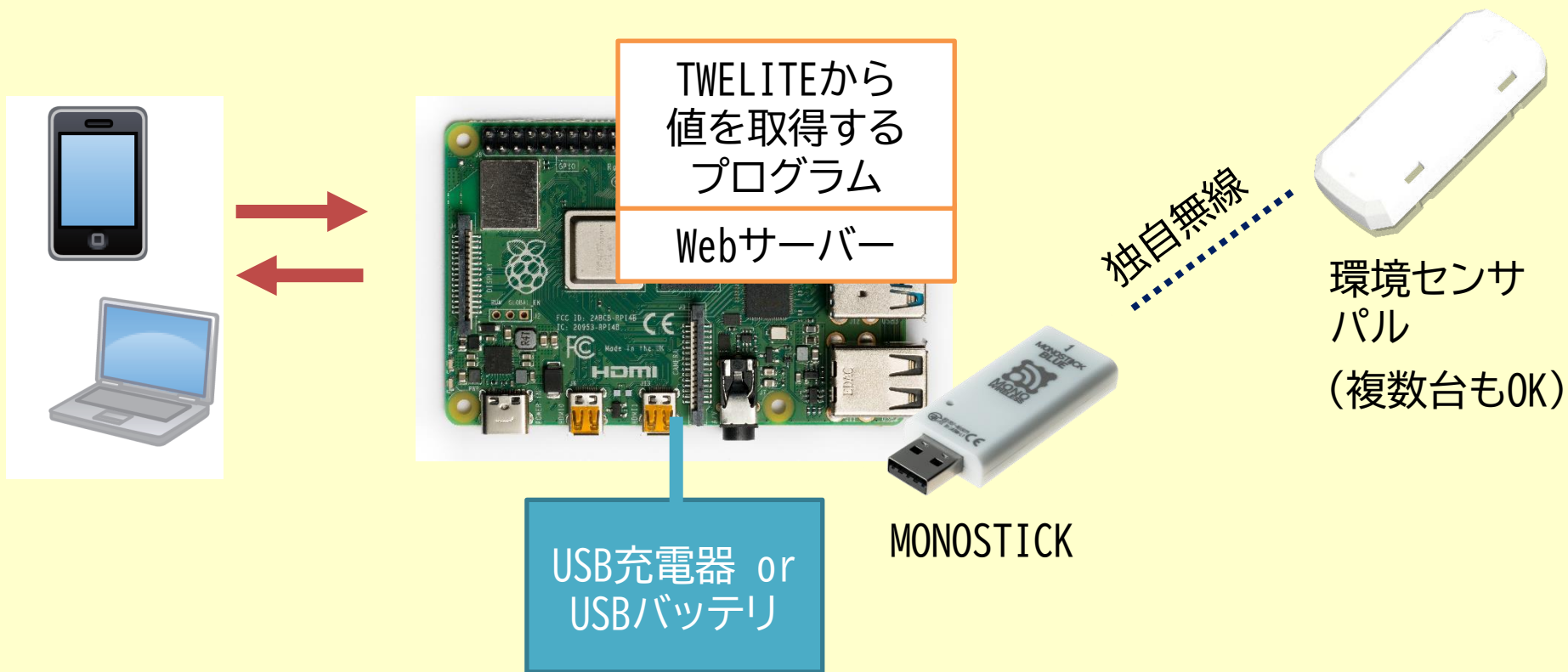


Presented by
Mofukabur.いんく



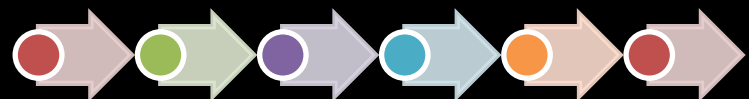
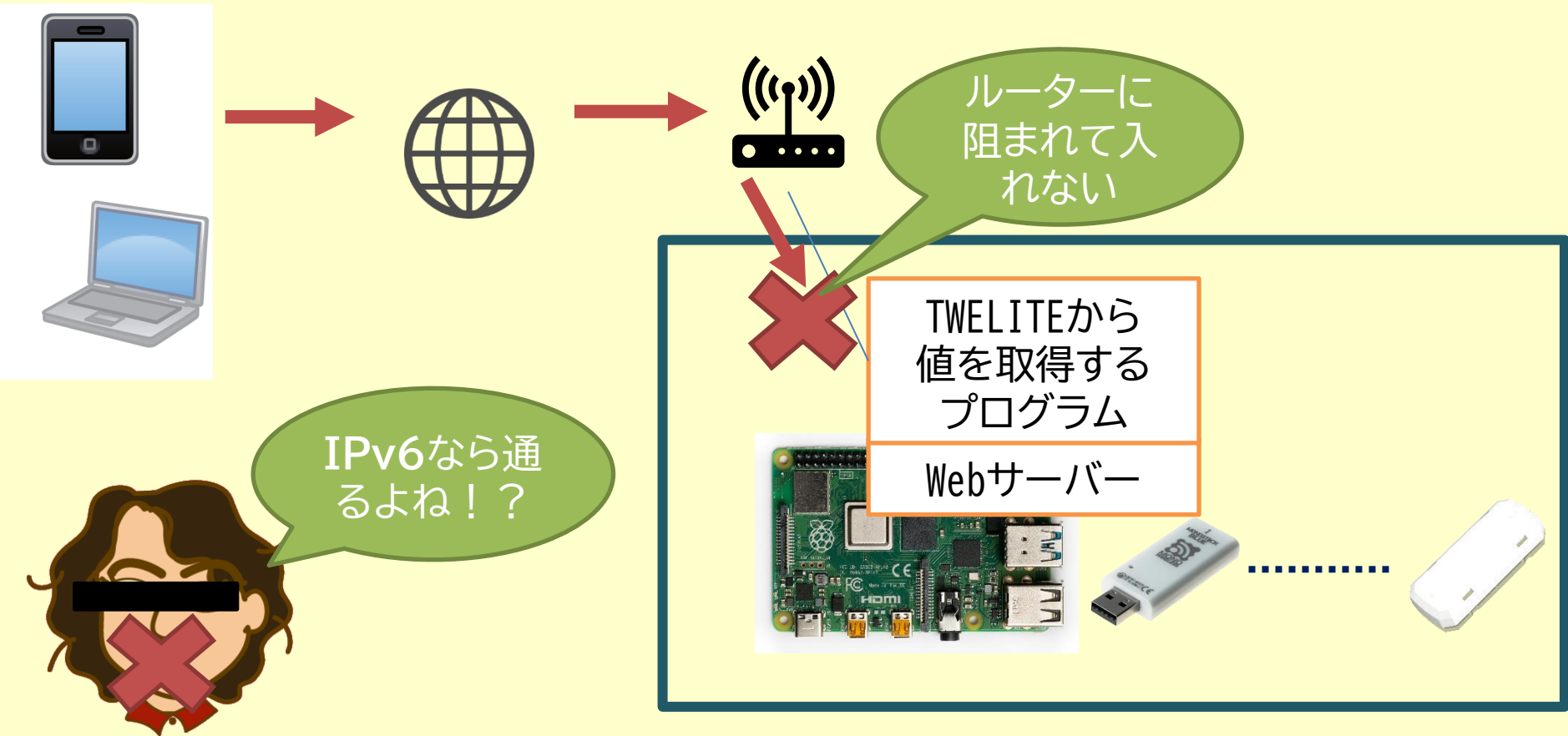


ここまでで実現できたもの



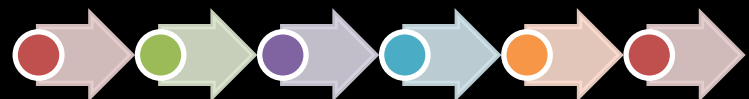
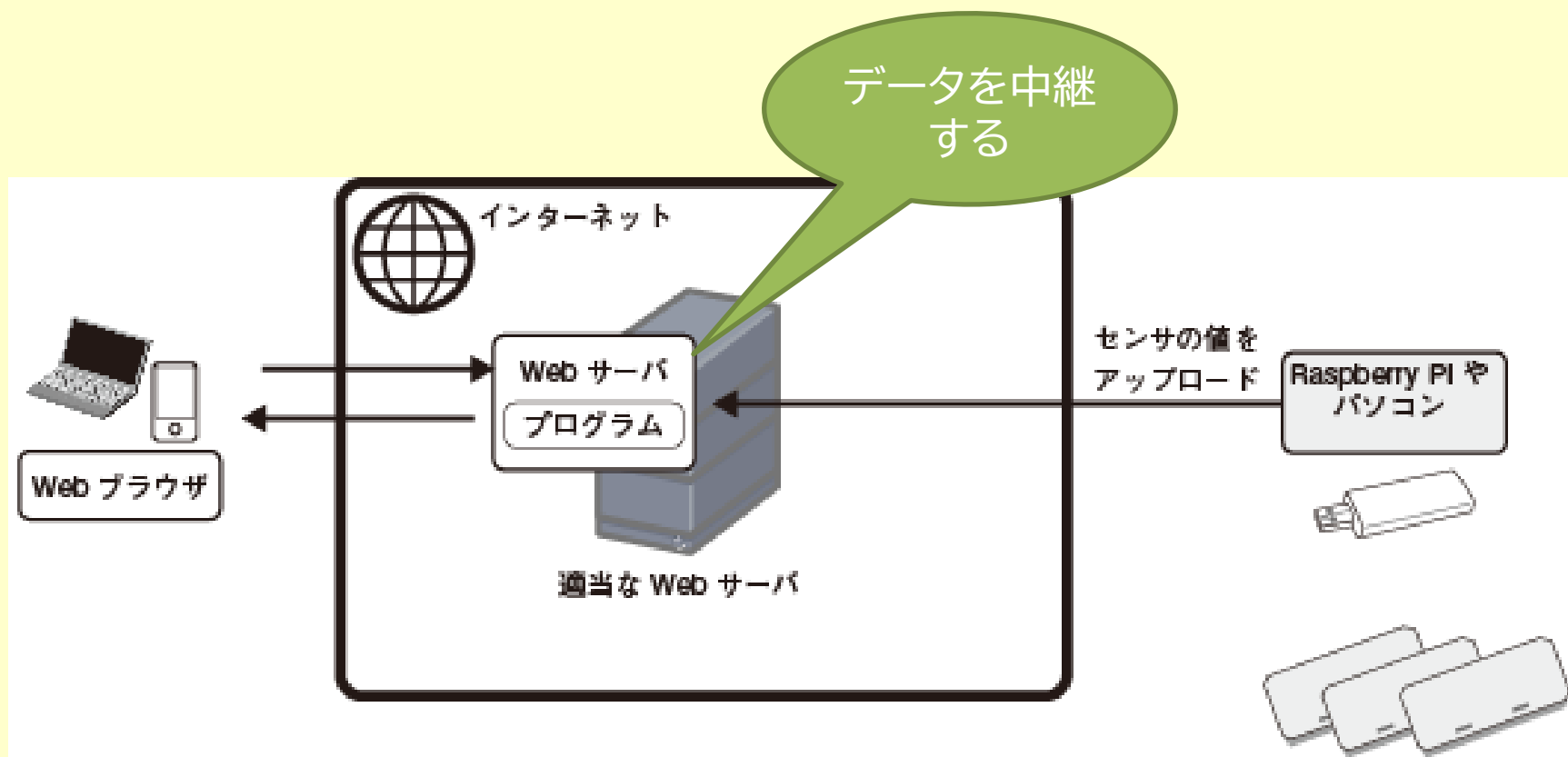


インターネットからはつながらない



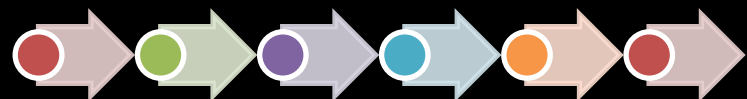


サーバを作って転送する





プログラムを作るのはめんどくさい



Presented by
Mofukabur.いんく



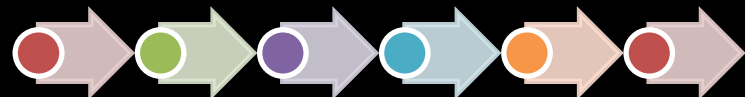
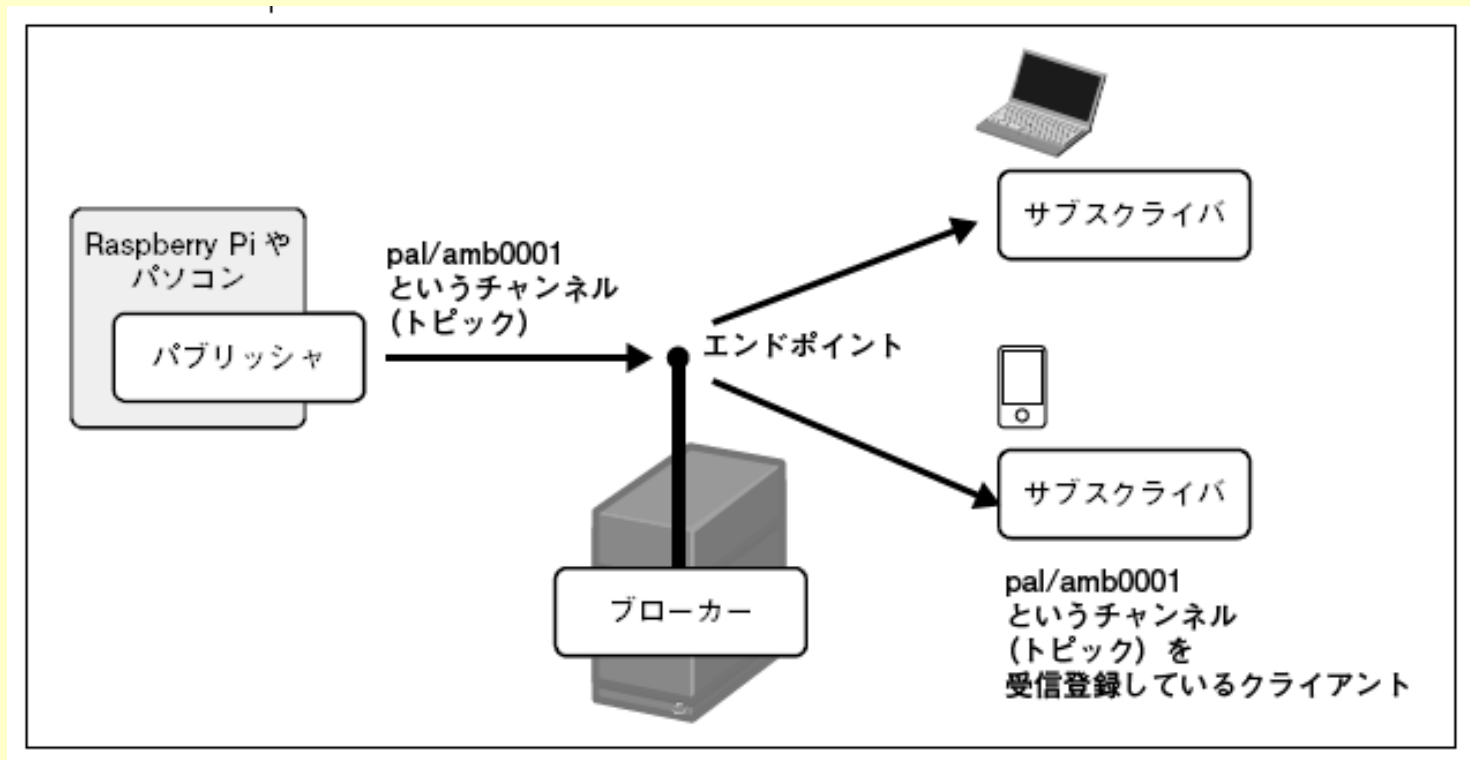


MQTT

送信する側=publisher

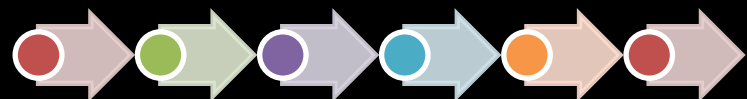
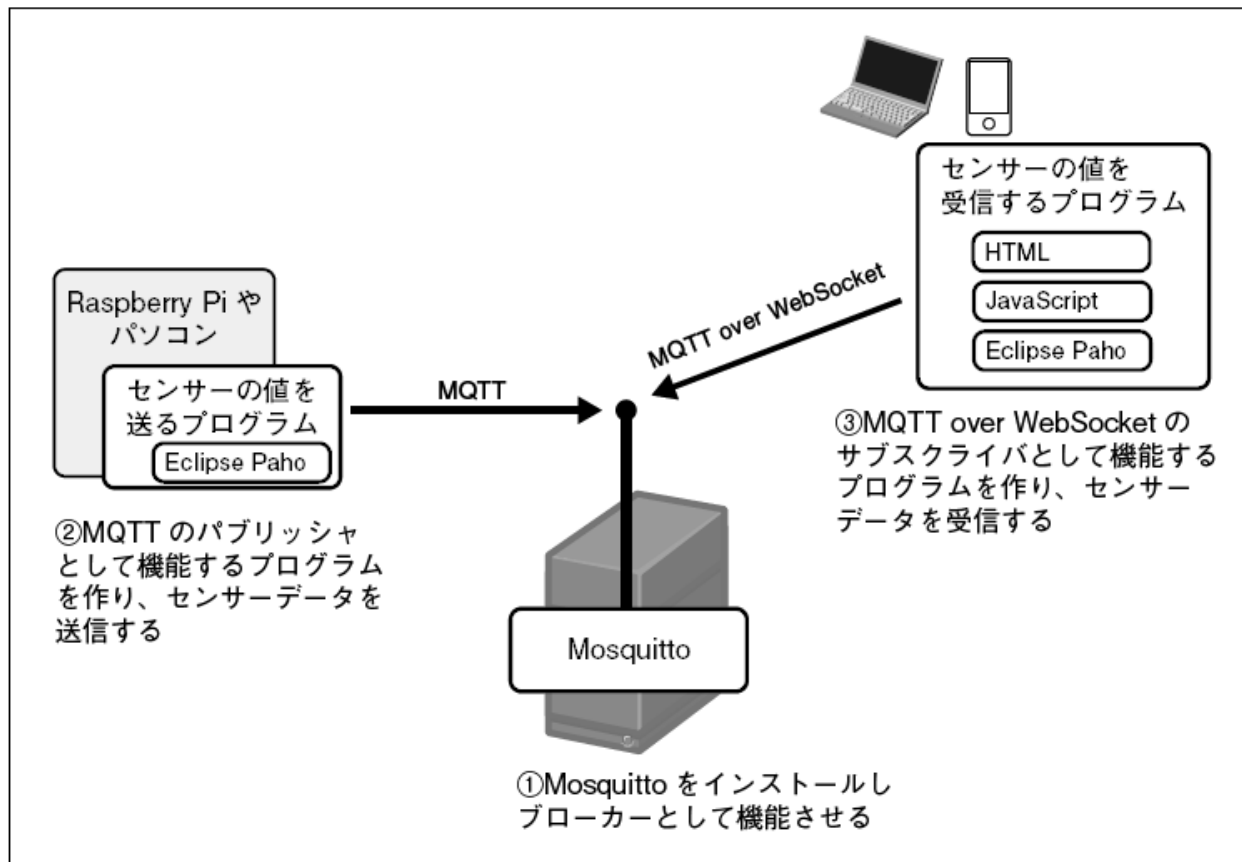
受信する側=subscriber

中継するサーバ=broker





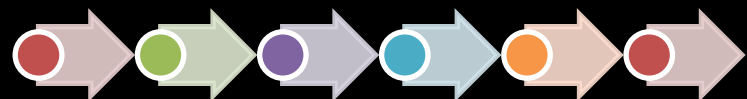
Mosquitto





- ブローカー(中継するサーバー)

Mosquittoをインストールする。



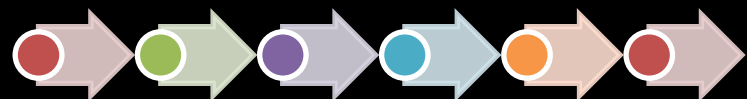
Presented by
Mofukabur.いんく





• パブリッシャ

MQTTでデータを送信するプログラムを書く。
「Eclipse Pahoライブラリ」を使うのがカンタン。



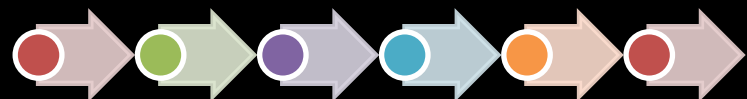


• サブスクライバ

エンドユーザーがアクセスする部分。ブラウザでアクセスするのであれば、Webサーバを起動する必要があるが…。

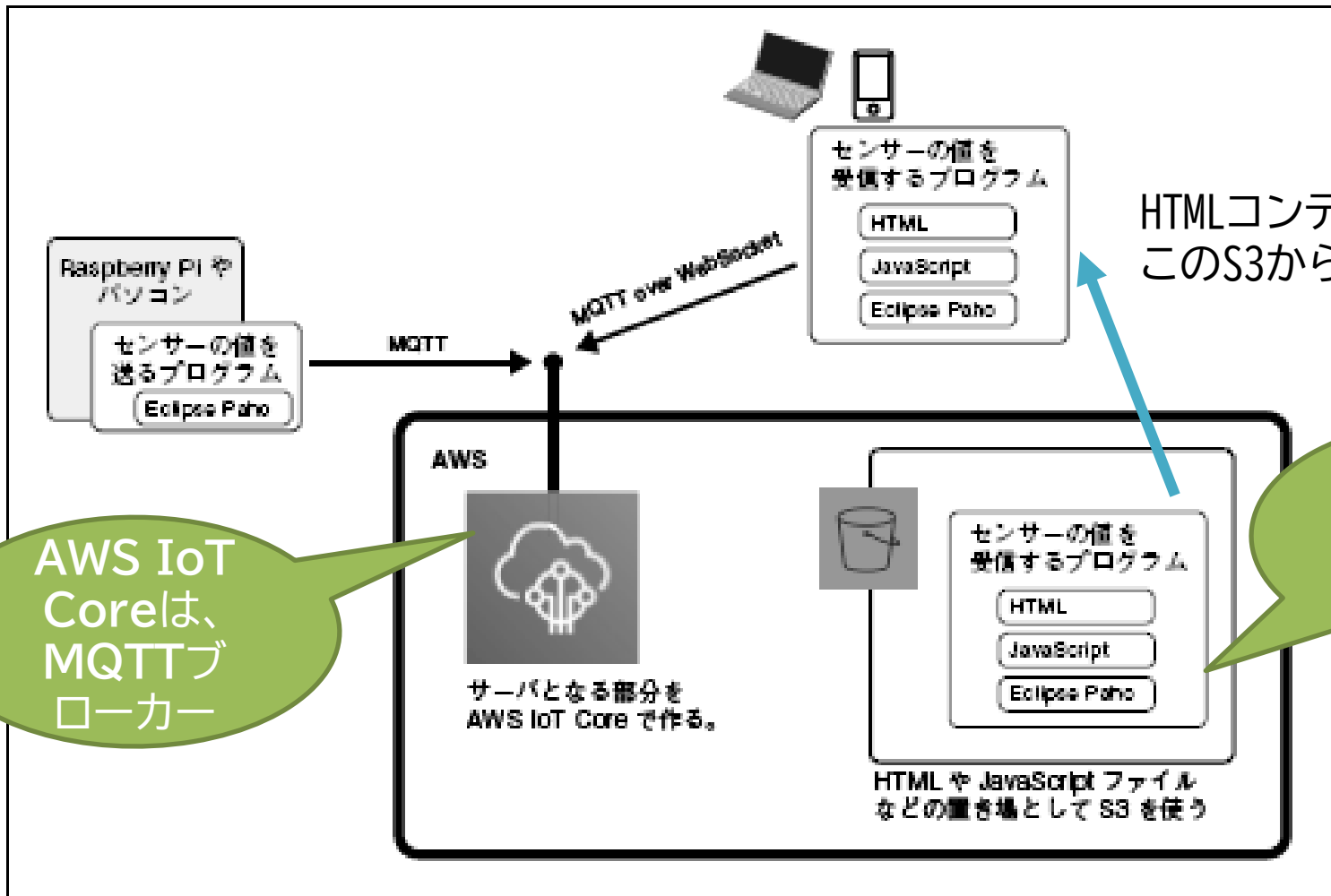
「MQTT over WebSocket」を使うと、ブラウザがJavaScriptのWebSocketという仕組みを使ってMQTTと直接通信できる。

この実装にも、Eclipse Pahoライブラリを使う





クラウドを使う



by

Mofukabur.いんく





データの送信(サブスクライブ)

```
endpoint = 'XXXXXXXXXX-ats.iot.ap-northeast-1.amazonaws.com'  
port = 8883
```

...証明書の設定、接続完了コールバックなどは略...

```
client.connect(endpoint, port=port)
```

```
client.loop_start()
```

...送信温度データを設定...

```
topic = 'pal/amb0001'
```

```
senddata = {
```

```
  'temperature' : data.get('Temperature'),
```

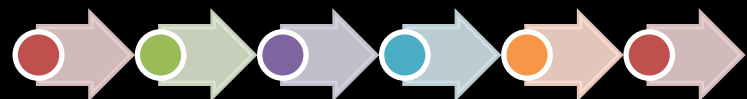
```
}
```

```
info = client.publish(topic, json.dumps(senddata))
```

```
info.wait_for_publish()
```

温度データ
を構造体に
設定

JSON文字
列としてパブ
リッシュ





データの取得(MQTT over HTTP)

```
function onMessage(message) {  
    var div = document.getElementById('msg');  
    div.innerHTML += '<p>' + message.payloadString + '</p>';  
}  
  
var client = new Paho.MQTT.Client(requestUrl, clientId);  
var connectOptions = {  
    onSuccess: function(){  
        client.subscribe(TOPIC);  
    },  
};  
  
client.onMessageArrived = onMessage;  
client.connect(connectOptions);
```

受信した
データ
(JSON)

コールバック
関数を設定





あとはお好きにグラフ化するなど

Smoothie Chartsを使ってみました

<http://smoothiecharts.org/>

WebSocketデスト

開始



```
{"temperature": 20.0, "humidity": 70.0, "illuminance": 100}
```

```
{"temperature": 20.0, "humidity": 70.0, "illuminance": 100}
```

```
{"temperature": 20.0, "humidity": 70.0, "illuminance": 100}
```

```
{"temperature": 20.0, "humidity": 70.0, "illuminance": 100}
```

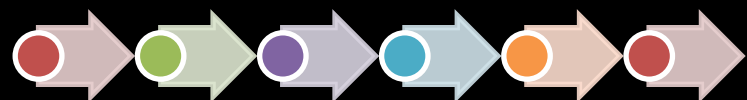
```
{"temperature": 20.0, "humidity": 70.0, "illuminance": 100}
```

```
{"temperature": 20.0, "humidity": 70.0, "illuminance": 100}
```





まとめ

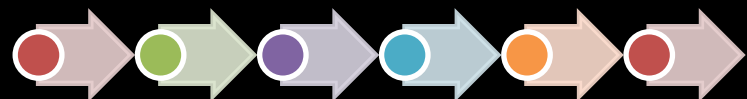


Presented by
Mofukabur.いんく



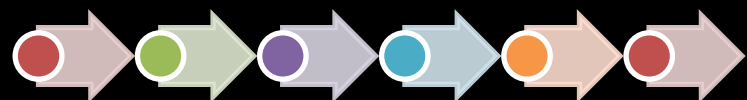


- 省電力センサを使う
- TWELITEは便利
- インターネットとのやりとりにはMQTT





今後の活動



Presented by
Mofukabur.いんく



モウフカブール



小笠原種高



モウフカブール

<http://mofukabur.com/>



大澤文孝



同人活動

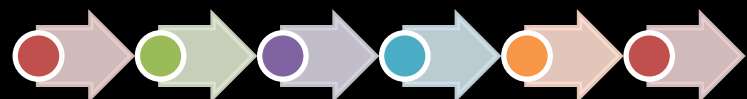
次回は技術書典11
7月10日～7月25日
(トライアルオフラインには
不参加)

モウフカブール
(<http://mofukabur.com/>)

技術書典、技術書博覧会(技書博)、コミケなどで活動中。
技術同人のほかの音楽CD、写真集などを頒布しています。



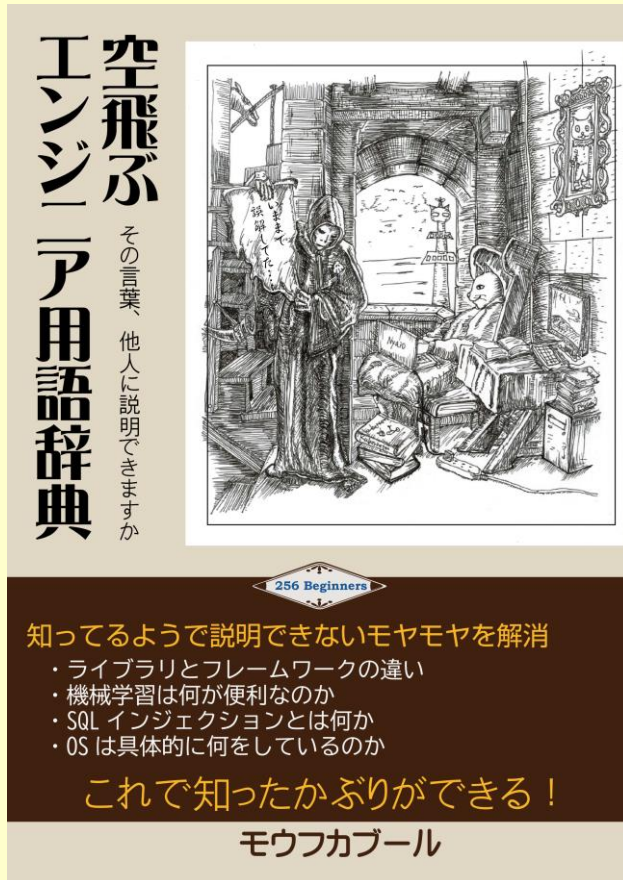
技書博5は、無事、終了
しました。皆様のご参加、
ありがとうございました！



Presented by
Mofukabur.いんく



同人誌新刊

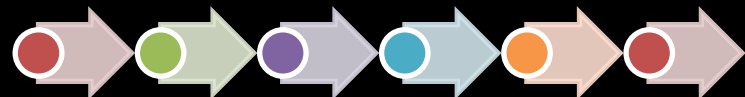


空飛ぶエンジニア用語辞典

いまさら聞けない用語を解説。
これで知ったかぶりができる、そんな用語集。

フレームワークとライブラリの違いは？
CDNって何？
機械学習で何ができるの？

などなど、いまさら聞けない疑問をまとめています





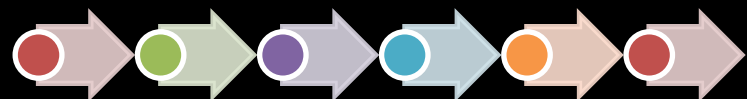
July Tech Festa 2021

7月18日(日)
オンライン開催

<https://techfesta.connpas.com/event/213069/>

今さら聞けないIT技術

踏み台とSSH、ポートフォワードについて、語ります



Presented by
Mofukabur.いんく



アンケートのご協力 お願いいたします

https://bit.ly/OSC2021Hokkaido_Form



Follow me



https://bit.ly/OSC2021Hokkaido_Form



Follow me

 @sour23



https://bit.ly/OSC2021Hokkaido_Form



Follow me

 @sour23



AWSで学ぶ
docker入門書

