

IoT プログラミング講座（6）

益田プログラミング研究会

Masuda Programming Research Society (MPRS)

<https://maro-v.jp/~nagata-iics/>

永田 武

#7 温度・湿度・気圧の収集（IoT プログラミング）

1. TWELITE の基礎

(1) 基本事項

BME280 で計測した温度と湿度、気圧を親機に設定した間隔で送信する。子機に BME280 を接続する場合、インターラクティブモードにて、センサ種別(m)を 0x39 に設定する。

https://mono-wireless.com/jp/products/TWE-APPS/App_Tag/mode_BME280.html

::rc=80000000:1q=162:ct=0016:ed=810043C1:id=0:ba=3080:a1=1097:a2=0586:tm=2600:hu=2736:at=1010

このデータは以下のような情報が記述されている。
rc: 中継機の SID(中継していない場合は 0x80000000)
lq: LQI
ct: 続き番号
ed: 子機 SID
id: 子機論理デバイス ID
ba: 子機の電源電圧(mV)
a1: AI1(mV)
a2: AI3(mV)
tm: 温度(℃)×100
hu: 湿度(%)×100
at: 気圧(hPa)

2. プログラム

(2.1) BME2080 フォーマットの受信

```
# bme_tw0.py
import serial # serial モジュールのインポート

def main():
    ser = serial.Serial("/dev/ttyUSB0", 115200)
    try:
        while True:
            line = ser.readline().rstrip()
            # 1行読込み（末尾空白削除）
            line = line.strip().decode('utf-8')
            # 空白削除・文字コード変換

            if len(line) > 0 and line[0] == ':': print("%s" % line)
            else: continue
            # : で始まっている場合は表示、そうでなければ以下スキップ

    except KeyboardInterrupt:
        pass

    ser.close() # ser を停止

if __name__ == "__main__": # プログラムの起点
    main()
```

動作確認 **p bme_tw0.py**

```
::ts=1697
::ts=1698
::rc=80000000:1q=162:ct=014E:ed=810B3545:id=0:ba=2970:a1=1116:a2=0603:tm=1811:hu=5887:at=1014
```

(2.2) 温度・湿度・気圧データの取出し

```
# bme_tw1.py
import serial # serial モジュールのインポート

def main():
    ser = serial.Serial("/dev/ttyUSB0", 115200)
    try:
        while True:
            line = ser.readline().rstrip()
            line = line.strip().decode('utf-8')
            line1 = line.split(':')
            if len(line1) > 3:
                print("¥n%s" % line1)
            else: continue
            print(line1[10]) # tm
            print(line1[11]) # hu
            print(line1[12]) # at

    except KeyboardInterrupt:
        pass

    ser.close()

if __name__ == "__main__": # プログラムの起点
    main()
```

動作確認 **p bme_tw1.py**

```
[',', 'rc=80000000', '1q=168', 'ct=015B',
'ed=810B3545', 'id=0', 'ba=2980', 'a1=1114', 'a2=0598',
'tm=1815', 'hu=5882', 'at=1014']
tm=1815
hu=5882
at=1014
```

(2.3) 温度・湿度・気圧の算出

```
# bme_tw2.py
import serial

def main():
    ser = serial.Serial("/dev/ttyUSB0", 115200)
    try:
        while True:
            line = ser.readline().rstrip()
            line = line.strip().decode('utf-8')
            line1 = line.split(':')

            if len(line1) <= 3: continue

            print(line1[10], line1[11], line1[12])
            tmlist = line1[10].split('=')
            hulist = line1[11].split('=')
            atlist = line1[12].split('=')
            print(tmlist, hulist, atlist)

            tm = int(tmlist[1])/100
            hu = int(hulist[1])/100
            at = int(atlist[1])
            print('tm = ', tm, 'hu = ', hu, 'at = ', at, '¥n')

    except KeyboardInterrupt:
        pass

    ser.close()

if __name__ == "__main__": # プログラムの起点
    main()
```

動作確認 **p bme_tw2.py**

```
tm=1818 hu=5859 at=1014
[tm', '1818'] [hu', '5859'] [at', '1014']
tm = 18.18 hu = 58.59 at = 1014
```

(2.4) 日時の追加

```
# bme_tw3.py
import serial # serial モジュールのインポート
from datetime import datetime

def main():
    ser = serial.Serial("/dev/ttyUSB0", 115200)
```

```

try:
    while True:
        line = ser.readline().rstrip()
        line = line.strip().decode('utf-8')
        line1 = line.split(':')

        if(len(line1) <= 3): continue

        now = datetime.now()
        date = '{:%Y/%m/%d %H:%M:%S}'.format(now)
        print(date, ' ', end="")

        #print(line1[10],line1[11],line1[12])
        tmList = line1[10].split('=')
        huList = line1[11].split('=')
        atList = line1[12].split('=')
        #print(tmList,huList,atList)

        tm = int(tmList[1])/100
        hu = int(huList[1])/100
        at = int(atList[1])
        print('tm =',tm,'hu=',hu,'at=',at)

except KeyboardInterrupt:
    pass

ser.close()

if __name__ == "__main__":
    main()

```

動作確認 **p bme_tw3.py**

2025/12/02 08:07:21 tm = 15.13 hu= 60.47 at= 1013

(2.5) ファイルへの書き出し

```

# bme_tw4.py
import serial
from datetime import datetime

def writeFile(fname,data):
    f = open(fname,'a')
    f.write(data)
    f.close()

...
    s = date+','+str(id)+','+
        str('{:<4.1f}'.format(tm))+','+
        str('{:<4.1f}'.format(hu))+','+
        str('{:4d}'.format(at))+'\n'
    #print(s)
    writeFile('tha.csv',s)

```

動作確認 **p bme_tw4.py**

2025/12/02 08:28 0 15.3 59.4 1013
2025/12/02 08:28 0 15.3 59.5 1013

ファイルの確認

cat .csv
2025/12/02 08:28 0 15.3 59.4 1013
2025/12/02 08:28 0 15.3 59.5 1013

3. 改造

%%%%%%%%%%%%%

(1) 1分毎にファイル保存

→ 秒が'00'になつたら保存する

(2) 1時間毎にファイル保存

→ 分が'00'になつたら保存する

(3) ファイル名を'yyyy_mm_dd.csv'として保存

→ ファイル名:

fname='{:%Y_%m_%d}'.format(now)+'.csv'

(4) 日替わり処理でファイル名を更新

→ 時が'00'になつたら、ファイル名を作成する

4. まとめ

- IoT プログラミングは、ハードウェアとソフトウェアの両方の技術が必要になります。
- ラズパイのような入出力ピンを有するコンピュータは、安価でありながら実用的なシステム開発が可能です。
- この講座では、IoT プログラミングで重要なと思われる内容について解説してきましたが、いかがでしたでしょうか。
- 今後の拡張としては、データベースの利用、GUI を備えたグラフの表示、インターネットへの収集データの更新などがあります。

参考資料(2)

第10章 データ保存

第11章 データ表示

第12章 データ公開

参考資料

(1) Pythonによるアルゴリズムとデータ構造の基礎

<https://www.coronasha.co.jp/np/isbn/9784339029079/>

(2) Raspberry PiによるIoTシステム開発実習

<https://www.morikita.co.jp/books/mid/085571>

連絡先 t.nagata.wp@gmail.com 永田 武

<http://www.maro-v.jp/~nagata-iics/>