「M5Core2を使って気象情報を可視化」

センサ：ENVⅡSENSER BMP280(温湿度、気圧センサ)

アクチュエータ：M5Core2LCD

**1：センサから値を取得して、PCにデータを送信する**

Arduinoソースコード

BMP280.ino

|  |
| --- |
| void loop() {    pressure = bme.readPressure();    if(sht30.get()==0){      tmp = sht30.cTemp;      hum = sht30.humidity;    }      M5.Lcd.printf("Temp: %2.1fC  \r\nHumi: %2.0f%%  \r\nPressure:%2.0fPa\r\n", tmp, hum, pressure);    float data1[]={tmp,hum,pressure};    writeSerial(data1);      delay(60000); // 1分ごとにデータを送信 60件/h  }  // Arduino から PC へセンサデータを送信する関数  void writeSerial(float sensorData[]) {    Serial.write("START\n");    for(int i=0;i<3;i++){    char buf[256];    String str = String(sensorData[i]) + "\n";    str.toCharArray(buf, str.length()+1);    Serial.write(buf);    }    Serial.write("END\n");  } |

参考：<https://sample.msr-r.net/m5stack-env2/>

持つ, テーブル, 座る, 男 が含まれている画像

自動的に生成された説明

図 1　M5Core2LCDセンサデータ表示

**2：PCへ送られてきたセンサデータをAmbientへ送信する**

data\_send\_ambient.py

|  |
| --- |
| # Arduinoの接続ポート      # PORT = '各自の接続ポート'      PORT = 'COM4'      ser = serial.Serial(PORT, 9800)      # チャネルID      # CHANNEL\_ID = 自分のチャネルID      CHANNEL\_ID = 30677      # ライトキー      # WRITE\_KEY = "自分のライトキー"      WRITE\_KEY = "hogehoge"      data = []      flag = False      while True:          message = ser.readline().decode('utf-8', errors='ignore').replace('\n', '')          if(message == 'START'):              flag = True          elif(message == 'END'):              flag = False              ambi = ambient.Ambient(CHANNEL\_ID, WRITE\_KEY)              # データが1つのとき              # r = ambi.send({"d1": data[0]})              # データが複数のとき              # print(data)              r = ambi.send({"d1": data[0], "d2": data[1], "d3": data[2]})              print(r)              data = []          else:              if(flag):                  print(message)                  data.append(message) |

テキスト

自動的に生成された説明

図 cmdにて$python data\_send\_ambient.pyを実行

グラフ, 折れ線グラフ

自動的に生成された説明

図3　Ambient温湿気圧グラフ

**3：Ambientにて取得した値をcsvファイルとして保存して、JupyterLabにて解析。**

bme280.ipynb

|  |
| --- |
| import pandas as pd #for import csv |
| bme\_data = pd.read\_csv("data1\_4\_night.csv") #before 'upload files' |
| import matplotlib.pyplot as plt |
| x = bme\_data['d4']  y\_temp = bme\_data['d1']  y\_humi = bme\_data['d2']  y\_press = bme\_data['d3'] |
| fig, ax1 = plt.subplots()  p1 = ax1.plot(x,y\_temp,label='pop',marker="o",color="blue")  ax1.set\_title("WeatherData\_1/4\_night")  ax1.set\_xlabel('min')  ax1.set\_ylabel('°C')  ax2 = ax1.twinx()  # 2つのプロットを関連付ける  ax3 = ax1.twinx()  # 2つのプロットを関連付ける  rspine = ax3.spines['right'] # 3軸対応  rspine.set\_position(('axes', 1.2)) # y軸の位置をずらす  ax3.get\_yaxis().get\_major\_formatter().set\_useOffset(False) # 気圧の数字を省略形から本来の形へ  p2 = ax2.plot(x,y\_humi,label='pop',marker="o",color="red")  ax2.set\_ylabel('%')  p3 = ax3.plot(x,y\_press,label='pop',marker="o",color="green")  ax3.set\_ylabel('Pa')  ax1.legend((p1[0], p2[0],p3[0]), ("Temp", "Humi","Press"), loc=2) |

参考：<https://qiita.com/doskoicoffee/items/281496e542ce3c8e2b34>

　　：<https://note.com/sakawapc/n/n8bc72b4c4ebc>

折れ線グラフ が含まれている画像

自動的に生成された説明

図 3　室内　朝の変化　7:30~8:00

グラフ, 折れ線グラフ

自動的に生成された説明

図 4　室内　昼の変化 13:00~13:20

グラフ, 折れ線グラフ

自動的に生成された説明

図 5　室内　夜の変化 19:00~19:30