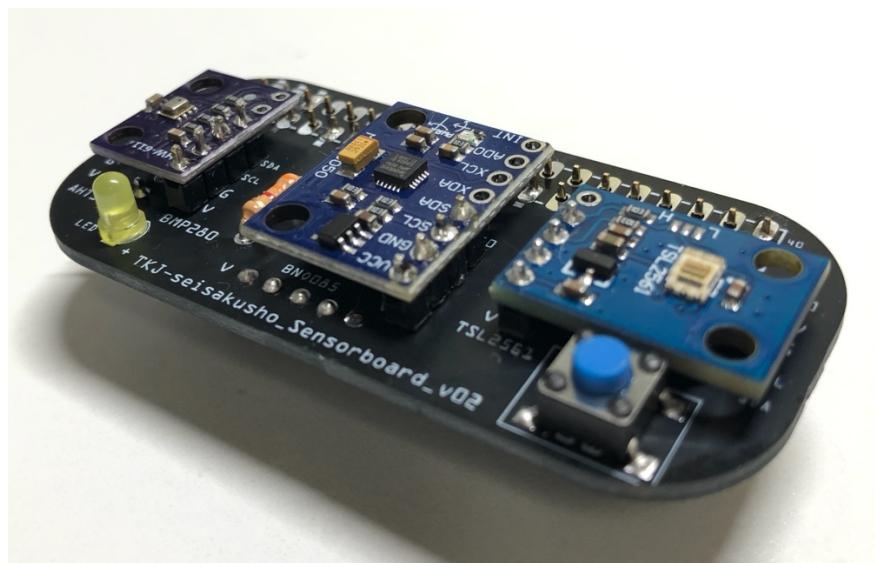


# EnvSensor 取扱説明書 v1.0

2026/2/4

## 1. 基板説明

### (ア) 搭載センサーについて



- |            |         |
|------------|---------|
| ① AHT30    | 気温、湿度   |
| ② BMP280   | 気圧      |
| ③ TSL-2561 | 照度      |
| ④ MPI6050  | 6 軸センサー |
- センサーの接続はすべて i2c プロトコル

### (イ) その他

- |       |   |
|-------|---|
| ① LED | 1 |
| ② SW  | 1 |

## 2. プログラム概要

本基板はセンサー類の評価用であり、具体的なアプリケーションは用意していませんが、下記のライブラリ、サンプルプログラムがあり、センサー類のテストが可能であり、アプリケーションへの応用も簡単になっています。

### (ア) センサーライブライ

- ① lib\_AHTx0.py
- ② lib\_BMP280.py
- ③ lib\_TSL2561.py

それぞれのライブラリは `def xxx` で読み出すと、1 回センサーの値を返します。`def main` で起動するとデータを連続して表示します。

## (イ) サンプルプログラム

- ① AHTx0\_dataSave.py
  - ② BMP280\_dataSave.py
  - ③ TSL2561\_dataSave.py
- ①～③の実行例

```
pi@raspberrypi: ~ env.sh
(V_env) pi@raspberrypi:~/envsensor $ ls
AHTx0_dataSave.py  lib_MPU6050.py      press_data.txt
BMP280_dataSave.py  lib_TSL2561.py      __pycache__
humdy_data_last.txt lux_data_last.txt  temp_data_last.txt
humdy_data.txt      lux_data.txt      temp_data.txt
lib_AHTx0.py        MPU6050_Demo.py   test_LED.py
lib_BMP280.py       old              test_sw.py
lib_MPU6050_log.csv press_data_last.txt TSL2561_dataSave.py
(V_env) pi@raspberrypi:~/envsensor $ python AHTx0_dataSave.py
測定値 Temperature: 24.0 C  Humidity: 42 %
補正値      温度: 0      湿度: 0
補正後 Temperature: 24.0 C  Humidity: 42 %
(V_env) pi@raspberrypi:~/envsensor $ python BMP280_dataSave.py
Temperature: 30.62 °C, Pressure: 1019.07 hPa
測定値 Pressure = 1019.07 hPa
補正値 0.0
補正後 Pressure = 1019.07hPa
(V_env) pi@raspberrypi:~/envsensor $ python TSL2561_dataSave.py
lux: 1213
測定値 lux = 1212.84
補正値 0.0
補正後 lux = 1212.84
補正後 int lux = 1213
(V_env) pi@raspberrypi:~/envsensor $
```

それぞれのサンプルプログラムは対応するライブラリを1回読み出し  
センサー値を取得して、データを保存します。

このサンプルプログラムを1分周期もしくは10分周期に起動すること  
により定期的にセンサー値を記録することが可能となります。

④ lib\_MPU6050.py

6 軸のデータを取得して表示します。

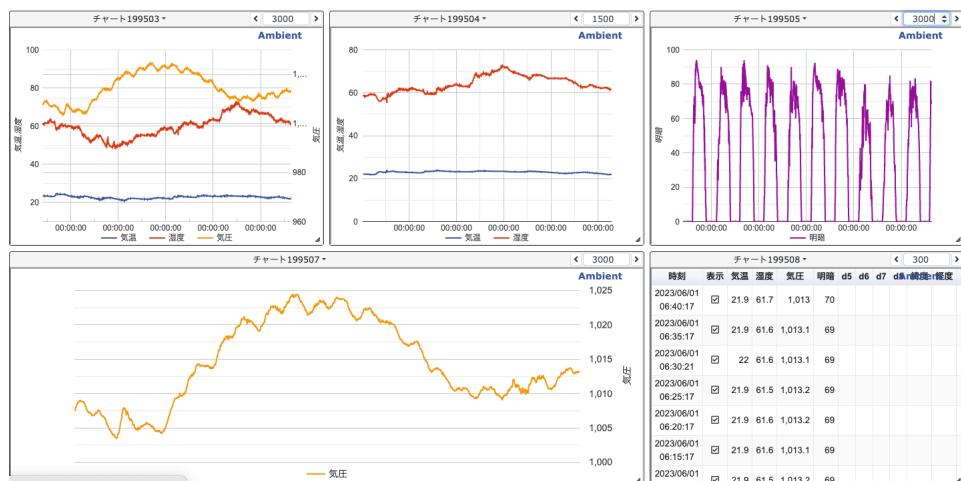
```
(V_env) pi@raspberrypi:~/envsensor $ python lib_MPU6050.py
GYRO [deg/s] X: 1.2 Y: 6.1 Z: 0.8 | ACCEL[m/s^2] X: -0.20 Y: 0.31 Z: 9.69
GYRO [deg/s] X: 0.5 Y: 6.7 Z: 0.8 | ACCEL[m/s^2] X: -0.23 Y: 0.23 Z: 9.52
GYRO [deg/s] X: 0.9 Y: 6.4 Z: 0.7 | ACCEL[m/s^2] X: -0.23 Y: 0.26 Z: 9.62
GYRO [deg/s] X: 0.4 Y: 6.6 Z: 0.7 | ACCEL[m/s^2] X: -0.22 Y: 0.18 Z: 9.40
GYRO [deg/s] X: 0.7 Y: 6.6 Z: 0.5 | ACCEL[m/s^2] X: -0.21 Y: 0.24 Z: 9.53
GYRO [deg/s] X: 0.8 Y: 6.8 Z: 0.4 | ACCEL[m/s^2] X: -0.24 Y: 0.32 Z: 9.48
GYRO [deg/s] X: -49.5 Y: -8.2 Z: 4.0 | ACCEL[m/s^2] X: 0.64 Y: -3.09 Z: 8.65
GYRO [deg/s] X: 1.9 Y: 7.1 Z: 0.7 | ACCEL[m/s^2] X: 1.03 Y: -4.28 Z: 8.52
GYRO [deg/s] X: -1.8 Y: 5.3 Z: 1.0 | ACCEL[m/s^2] X: 1.04 Y: -4.17 Z: 8.52
GYRO [deg/s] X: 109.3 Y: 32.8 Z: -6.9 | ACCEL[m/s^2] X: 0.21 Y: -1.33 Z: 8.92
GYRO [deg/s] X: 0.1 Y: 4.0 Z: -0.8 | ACCEL[m/s^2] X: 0.11 Y: 3.18 Z: 8.98
GYRO [deg/s] X: 39.6 Y: -17.2 Z: -2.6 | ACCEL[m/s^2] X: 0.05 Y: 0.97 Z: 9.08
GYRO [deg/s] X: 7.9 Y: 12.4 Z: 7.0 | ACCEL[m/s^2] X: 1.00 Y: 1.56 Z: 9.61
GYRO [deg/s] X: -0.1 Y: 5.7 Z: 0.8 | ACCEL[m/s^2] X: -0.71 Y: 0.21 Z: 9.41
GYRO [deg/s] X: 0.6 Y: 6.6 Z: 0.5 | ACCEL[m/s^2] X: -0.26 Y: 0.54 Z: 9.45
GYRO [deg/s] X: 0.5 Y: 6.5 Z: 0.6 | ACCEL[m/s^2] X: -0.26 Y: 0.46 Z: 9.63
```

ジャイロと加速度センサーの値を表示します。センサーを動かすと数値も変わります。生データを物理量に変換して表示しています。揺らぎや誤差対策はしていません。

(ウ) Ambient で時系列にグラフ表示します。

① 定期的にサンプルプログラムを起動した上で、データを ambient に送信することでインターネット上でデータを確認することが可能になります。

その際には、ambient にサインインして各種設定をする必要があります。例として以下ののようなダッシュボードを作ることが可能です。



② Ambient は無料では 3000 点のデータを表示できます。毎分データを送信すると約 2 日、5 分毎に送信すると 10 日分表示する事ができます。

- ③ Ambient の公式 github を参考に python でデータを投げることが可能です。

<https://github.com/AmbientDataInc/ambient-python-lib>

### 3. インストール

#### (ア) ■仮想環境を作る

① `python3 -m venv V_env --system-site-packages`

#### (イ) ■TSL2561 用ライブラリ

① `pip install adafruit-circuitpython-tsl2561`

### 4. Ambient の設定

#### (ア) Ambient の HP にて、アカウントを取得して、設定を行ってください。

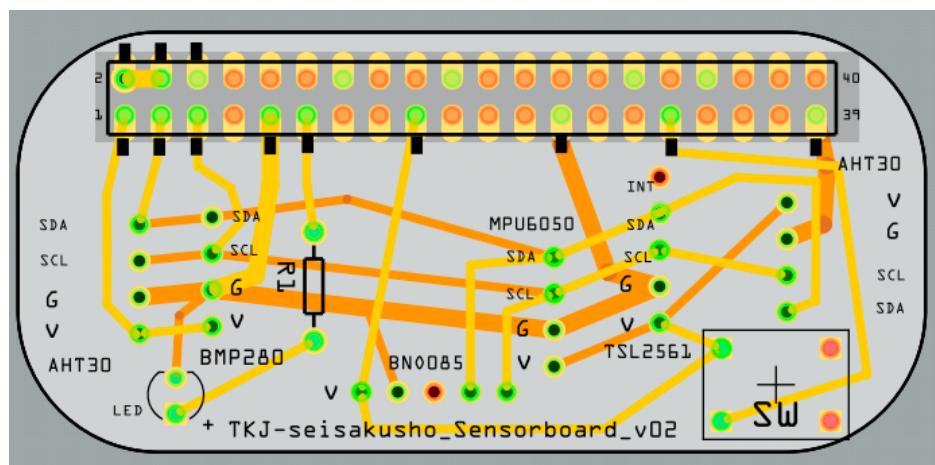
<https://ambidata.io/docs/gettingstarted/>

#### (イ) あと、「qiita ambient」などと検索すると丁寧に説明しているサイトがあるので、そちらを参照してみてください。

(ウ) d1:気温、d2:湿度、d3:気圧、d4:明暗となっています。

(エ) pico へのプログラムアップロード、config.py の設定、ambient の設定が正常であれば、pico を電源に接続すると自動起動し、ambient にデータが記録されます。

### 5. LED とスイッチ



#### (ア) 接続 GPIO #

① LED GPIO #17 が 1 で点灯

② タクトスイッチ GPIO # 6 スイッチ ON で GPIO が 1 となる

1. タクトスイッチのプルアップはソフト処理

### (イ) サンプルプログラム

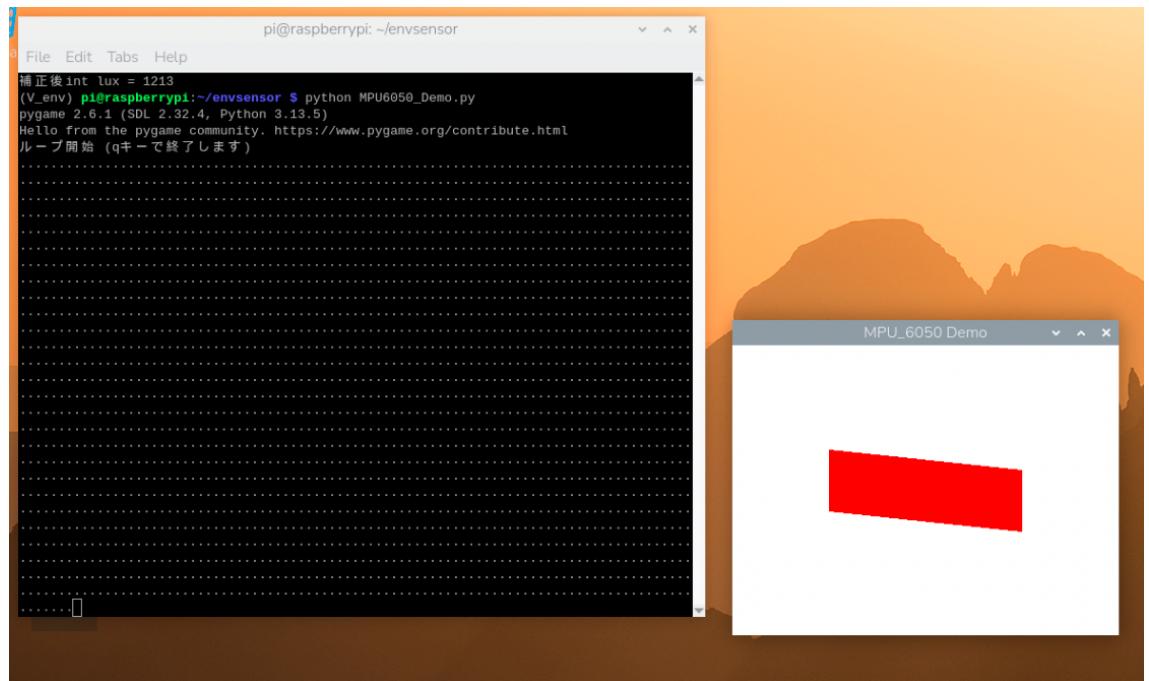
#### ① スイッチとLEDのデモプログラム

1. test\_LED.py
2. test\_sw.py

## 6. MPU6050 デモについて

### (ア) MPU6050\_Demo.py

- ① 起動すると、画面に平らな板を表示
- ② センサーを傾けるとそれに応じて画面の板が動きます。



q キーの入力でプログラム終了

以上