

# ePaper 取扱説明. V1.1 2024/12/09

ePaper 表示器のラズパイ専用基板です。

この取扱説明書は、WeAct Studio 2.9 インチ 白黒 ePaper ディスプレイ用の基板、Python ライブラリの使用方法について説明します。このライブラリは GNU General Public License v3.0 のもとで配布されています。

ePaper は書き換え時に電力を少し消費しますが、その後は電源を切っても表示し続けます。この特性を活かした活用ができれば、非常に便利です。

また原理上焼き付けを起こさないなので、長期間の表示が可能です。

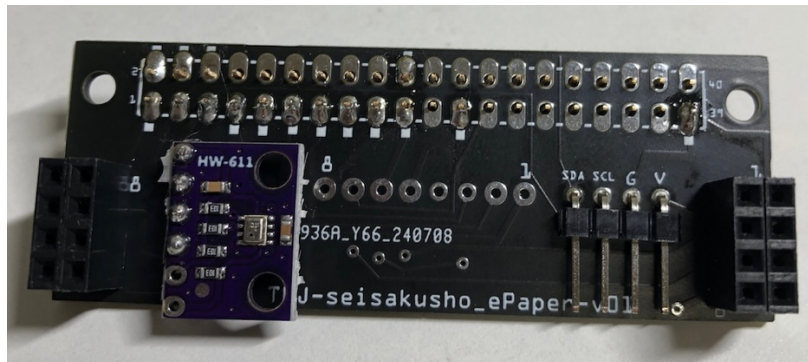
本ライブラリは元々jairoshさんの raspberry-pi-ssd1680 を基にしています。

## 使用方法

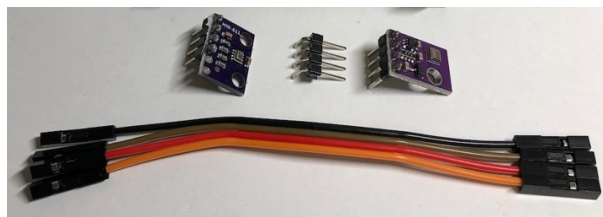
### 1. 基板への装着はコネクタ式です。

(ア) 基板にはセンサー付きとセンサーなしがあります。以下の説明ではセンサー付きで説明しますが、センサーなしを購入された場合は読み飛ばして下さい。

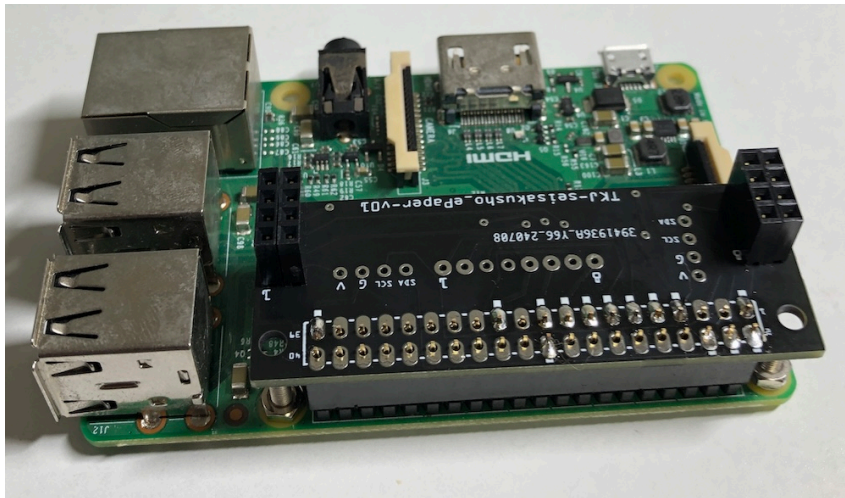
(イ) 左右どちらでも接続ができプログラムの変更なしに表示上下を変えられます。



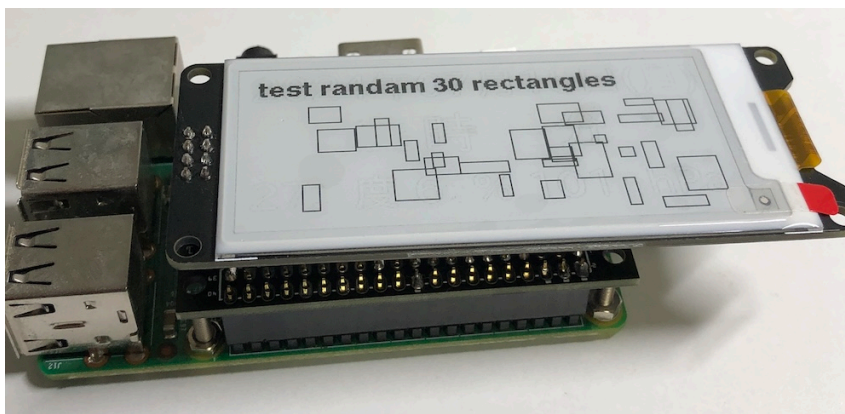
気圧センサーBMP280 は基板に直付けとなりますが、気温・湿度センサーAHT30 はラズパイの熱の影響を避けるためリード線接続としています。



AHT30 の接続は間違わないように基板の表示を確認して行なって下さい。逆に接続すると壊れます。



ラズパイに基板を接続した状態



左右どちらでも接続して使用可能です。

装着方法を逆にしないように注意して下さい、壊れる恐れがあります。

(ウ) リード線接続も可能で、いろんな使い方に対応可能です。ただし L 型ピンヘッダーはオプションですので、ご自身で調達装着をお願いします。リード線は ePaper

に付属しています。

(エ) コネクタから外しても表示し続けるので、通常の表示器とは違った使い方が可能です。

## 2. ePaper の信号配線

(ア) ePaper は SPI インターフェースを使いますので、ラズパイで SPI を使用にする必要があります。

(イ) ピン配置は以下の通りです。

- ① # BUSY      GPIO4
- ② # RES        GPIO17
- ③ # D/C        GPIO27
- ④ # CS         GPIO22
- ⑤ # SCK        GPIO11 (SPI0 SCK)
- ⑥ # SDATA      GPIO10 (SPI0 MOSI)
- ⑦ # GND

## 3. インストールとセットアップ

spi を有効にする

必要なライブラリをインストールします。

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install python3-pip
```

```
pip install RPi.GPIO numpy bdfparser pillow
```

```
pip3 install adafruit-blinka
```

# 日本語 TT フォントをインストール

```
sudo apt install python3-pil
```

```
sudo apt-get install fonts-dejavu
```

```
sudo apt-get install fonts-ipafont
```

プログラムをインストール

```
git clone https://github.com/momorara/ePaperPi
```

AHT,BMP を使う場合はまず、システムを再起動して以下をインストール  
まず、システム設定で i2c を有効にしてください。

>>i2c デバイスとして認識されているか確認

i2cdetect -y 1

AHT 38 /BMP 76 です。

>>BMP280 ライブラリのインストール(2023/9 以降)

sudo pip3 install bmp280

git clone https://github.com/pimoroni/bmp280-python

cd bmp280-python/examples/

sudo python3 temperature-and-pressure.py

(例)

temperature-and-pressure.py - Displays the temperature and pressure.

Press Ctrl+C to exit!

29.58\*C 1005.60hPa

29.57\*C 1005.60hPa

....

>>AHT10,AHT20,AHT30 ライブラリのインストール

cd

pip3 install adafruit-circuitpython-ahtx0

cd ePaperPi/

python3 AHTx0\_dataSave03.py

として、温度と湿度が表示されればOK

(例)

測定値 Temperature: 25.1 C    Humidity: 31.6 %

補正值            温度: 0                    湿度: 0

補正後 Temperature: 25.1 C    Humidity: 31.0 %

自動起動する場合は

cron の設定をしてください

#### 4. 使用方法

ePaper の使用方法としての基本は、メモリ上に画面に対応したドット数のバッファを設け、そのバッファに白黒のドットを書き込み、最後にバッファのデータを ePaper に転送することで表示します。こうするのは、ePaper は通常の表示器と違い表示更新に時間がかかるためです。いちいちデータを転送していると表示が遅くなってしまいます。

サンプルプログラムを用意しています。

ePaperPi にて以下のサンプルを実行してみてください。

ep\_bmpFile.py

ep\_circles.py

ep\_jptext.py

ep\_line.py

ep\_pixel.py

ep\_rectangle.py

ep\_text.py

今回なるべく python で使いやすいようにライブラリを作成しました、  
ep\_lib.py を import して使います。

a. 画面を白くクリア

```
ep_lib.clear_w()
```

b. 画面を黒くクリア

```
ep_lib.clear_b()
```

c. テキストの表示

テキストを表示するには、まずフォントを設定し、次に text() 関数を使用します。

```
ep_lib.set_font(1) # フォントを設定 (1, 2, 3 のいずれか)
```

1. helvB14.bdf

2. luBS14.bdf

3. spleen-8x16.bdf となっています。

```
ep_lib.text("表示するテキスト", x 座標, y 座標, バッファに書き込むかどうか(1/0))
```

d. 図形の描画

点を書く

ep\_lib.pixel(x座標, y座標, 色('B' または 'W'), バッファに書き込むかどうか(1/0))  
線を書く

ep\_lib.ina(x1, y1, x2, y2, 色('B' または 'W'), バッファに書き込むかどうか(1/0))  
円を書く

ep\_lib.circle(x座標, y座標, 半径, 色('B' または 'W'), バッファに書き込むかどうか(1/0))

四角を書く

ep\_lib.rectangle(x座標, y座標, 幅, 高さ, 色('B' または 'W'), バッファに書き込むかどうか(1/0))

e. ビットマップ画像の表示

ビットマップ画像を表示するには、bmp() または bmp\_img() 関数を使用します。

- ・ビットマップファイルを指定して表示

ep\_lib.bmp(x座標, y座標, 画像のパス, 色(0 または 1), バッファに書き込むかどうか(1/0))

ep\_lib.bmp\_img(x座標, y座標, 画像イメージ, 色(0 または 1), バッファに書き込むかどうか(1/0))

f. 日本語フォントの表示

日本語の表示のみ上記の使い方と違ってしています。

日本語表示用のビットマップイメージを別に作り、そこに日本語フォントを書き込み、最後にそのビットマップイメージを ePaper に転送する形をとっています。

使い方については、ep\_jptext.py を確認して下さい。

トゥルータイプフォントを使えば、大きさも表示位置も自由に設定可能です。

g. ライブラリの終了

ep\_lib.close()

※ 文字では使用方法がわかりにくいと思います。ep\_lib.py やサンプルプログラムを動かしてみても使い方を理解していただくのが早いと思います。

7. 注意事項

ライブラリを使用する前に、GPIO と SPI の設定が正しく行われていることを確認してください。(専用基板を使用する場合は、接続方向を注意して下さい。)

ディスプレイに書き込む際は、適切なバッファ操作を行うようにしてください。

プログラム終了時には必ず close() 関数を呼び出して、ライブラリを終了してください。

表示の際にバッファからはみ出すようなデータを使うと、表示が無視されるか、最悪エラーになります。

以上