

# mimamoriPico インストール手順書

## V1.2

2024/2/6

## 1. RaspberryPi pico W

(ア) メーカーの正式な文書は

<https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>

にありますので、そちらを参照ください。

## 2. picoW に micropython 環境を作る

(ア) 本文書では、母艦をパソコン(Mac)として、IDE に Thonny を使用します。

(イ) 開発言語は、MicroPython とします。

(ウ) ファームウェアのダウンロード

- ① <https://www.raspberrypi.com/documentation/microcontrollers/micropython.html>

上記 URL にアクセスし、picoW 用のファームウェアをダウンロードします。

RPI\_PICO\_W-20240105-v1.22.1.uf2 (2024/1/25 現在)

(エ) picoW を Boot ボタンを押しながら PC に USB 接続し、マウントしたディスクダウンロードしたをファームウェアコピーする。

(オ) するとマウントされていたデバイスがリセットされます。

- ① この行為が picoW をリセットすることになります。今後エラーが発生し、どうしようもなくなった時は、ここへ戻ると良いと思われます。

## 3. Thonny 開発環境を PC に作る

(ア) Thonny のダウンロード

- ① <https://thonny.org/>
- ② 上記 URL にアクセスし、thonny をダウンロードします。
- ③ インストールします。
- ④ Thonny のバージョンや win,Mac により画面が違う場合がありますが、基本的には同様だと思われます。

(イ) picoW を USB 接続をしておいて、USB ポートを picoW とすると thonny の画面で picoW を認識します。

- ① thonny の使い方については、ネットに多数アップされているので、そちらを参照ください。

### ◆pico との接続

Thonny メニューを選択 > Setting を選択

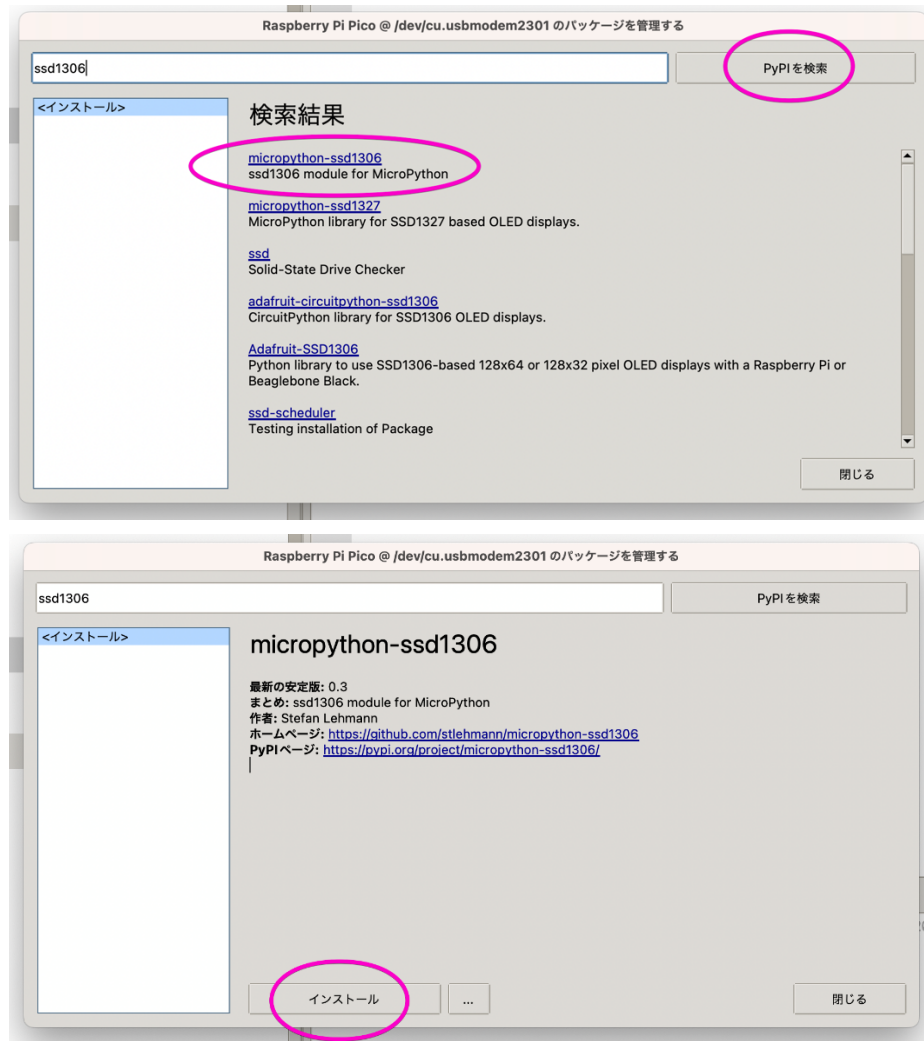
Thonny オプションが開くので、インタプリタを選択

MicroPython(Raspberry Pi Pico)を選択

ポートを Board in FS mode とする

#### 4. プログラムのダウンロード

- (ア) 今回のプログラムは git にアップしています。
- (イ) git clone <https://github.com/momorara/mimamoriPico>
- (ウ) で、pc にダウンロードしてください。
- (エ) Thonny でダウンロードしたディレクトリを見えるようにした状態で、プログラムを選択して、アップロードを実行してください。
- (オ) すると、picoW の方にプログラムがアップロードされます。
- (カ) OLED 用のライブラリをインストールする。
- ① メニューのツールを選択し、更にパッケージを選択
  - ② 入力エリアに「SSD1306」と入力、PyPI を検索ボタンをクリック
  - ③ SSD1306 MicroPython のインストールを行なってください。



## 5. 各種設定を行う。

(ア) Config.py に各種設定を入れていますので、こちらを編集してください。

(イ) Wifi の SSID やパスワード、測定周期などの設定ができます。

## 6. テスト

(ア) Thonny で、picoW 上の lib\_LED.py を選択して、実行ボタンを押してください。LED が点滅すれば、OK です。

(イ) Wifi\_onoff.py を選択して実行ボタンを押してください、wifi 接続ができれば、OK です。

(ウ) 複数のプログラムがあるので、できれば、一つひとつ試してみてください。ほとんどのプログラムは単体でも動作するようにしていますので、どのプログラムがどんな動作をするのかを理解しておくとも全体も理解しやすいと思います。

(エ) 次に ambient のアカウントを取って設定を行ってください。

- ① 取扱説明書に簡単に記載していますが、ambient の HP(<https://ambidata.io/docs/gettingstarted/>)を見て設定していただくか、「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。

(オ) 最後にメインプログラムである mimamori\_04.py を動作させてください。

- ① 設定が正しければ、センサーからデータを取得し、画面表示するとともに ambient にデータが表示・蓄積されます。

## 7. 実運用

(ア) 実際にスタンドアローンで動作させる場合は、main.py をアップロードして、pc と接続せずに、USB に給電すれば、自動的に main.py が動作します。動作自体は mimamori\_04.py とほぼ同じです。

以上