

# Raspberry Pi Pico W BaseBoard

## サンプルプログラム ユーザーガイド

Version 1.1

2024/1/31



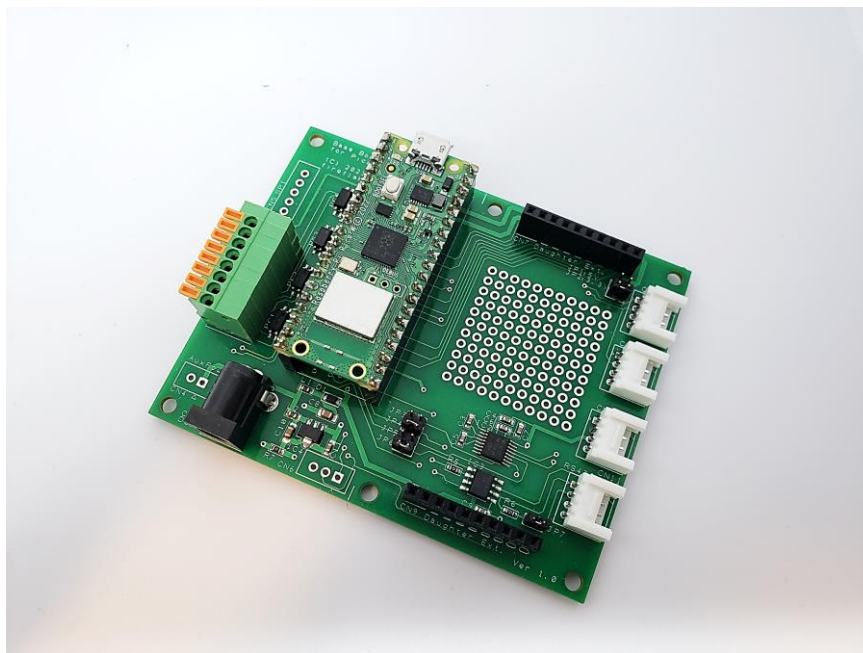
## 目次

1. 始めに .....	3
2. サンプルプログラムの概要 .....	3
3. プログラムのインストール方法 .....	4
3.1. PC から Pico W に接続してプログラムの読み書きが出来るようにする .....	4
3.2. Thonny などの IDE から Pico W にプログラムを書き込む.....	6
4. プログラムの実行方法 .....	9
5. 各プログラムの説明 .....	10
5.1. I2C で通信する.....	12
5.2. SPI で通信する.....	13
5.3. RS232C で通信する .....	13
5.4. RS485 で通信する .....	14
5.5. フォトカプラへの電圧入力を検知する .....	14
5.6. フォトカプラから電圧出力を行う .....	16
5.7. ADC で電圧測定をする.....	16
6. ライセンスについて .....	17
7. おわりに.....	17

## 1. 始めに

この資料は、fireflake 製ベースボードのためのサンプルプログラム、の説明資料です。

サンプルプログラムを使う際には、お手元に、Raspberry Pi Pico W, fireflake 製ベースボードの 2 点があり、下の画像のように組み立ててあるものとします。



基本的に本資料の情報が正となりますが、サンプルプログラムのバージョンによっては、本資料の説明と動作が異なるかもしれません。その場合にはサンプルプログラムそれ自体を正とします。

## 2. サンプルプログラムの概要

ベースボード用サンプルプログラムは MicroPython で書かれています。MicroPython は組込み向けに最適化された Python です。Pico W では C/C++ も動作しますが、MicroPython の方が手軽に使う事が出来るため、これを採用しました。お好みに応じて C, C++ を使うことも、もちろん可能です。

サンプルプログラムには、単機能のプログラムのみが含まれます。

単機能のプログラムとは、ベースボード上のフォトカプラから電圧出力を行う、あるいはベースボードに接続したセンサーのデータを取得する、といったシンプルなプログラムです。

サンプルプログラムは、プログラム本体と関連ライブラリーを Pico W にインストールするだけで動作します。

### 3. プログラムのインストール方法

ここでは、サンプルプログラムを Pico W にインストールする方法を説明します。

#### 3.1. PC から Pico W に接続してプログラムの読み書きが出来るようにする

この方法については、Raspberry Pi 公式サイトの説明が、（英語ですが）分かりやすくまとまっています。

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

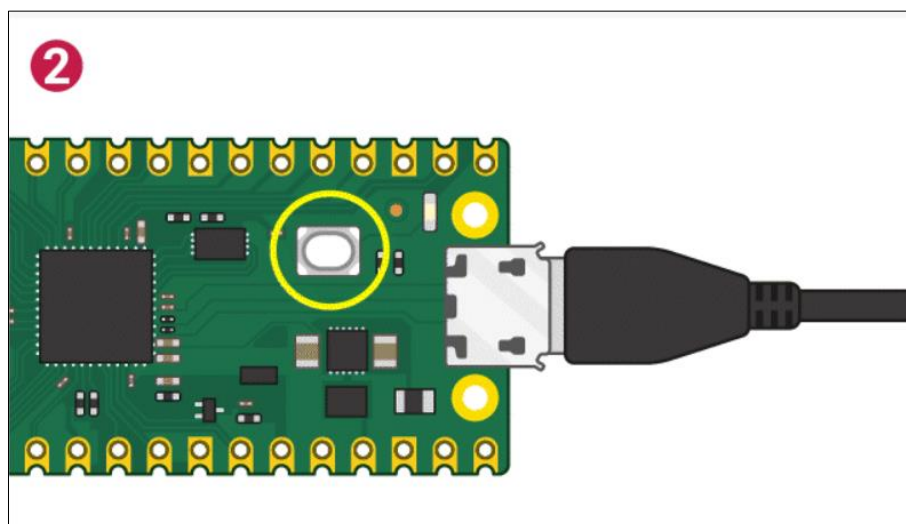
以下は、公式サイトの説明を抜粋したものになります。

まず、公式サイトから UF2 ファイルをダウンロードしてください。「Download the correct MicroPython UF2 file for your board:」と書いてある箇所で「Raspberry Pi Pico W」のダウンロードリンクをクリックしてください。

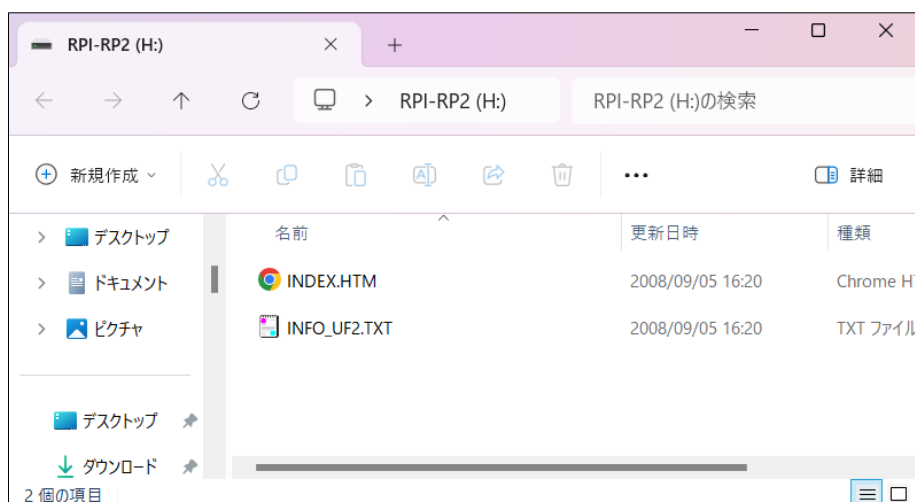
Download the correct MicroPython UF2 file for your board:

- Raspberry Pi Pico
- Raspberry Pi Pico W with Wi-Fi and Bluetooth LE support

Pico W の BOOTSEL ボタンを押したままにして、PC の USB ポートに接続します。接続出来たら、BOOTSEL ボタンを  
はなします。



Pico W が、RPI-RP2 という大容量ストレージデバイスとして認識されます。Windows の場合、RPI-RP2 のエクスプローラー画面が開きます。



さきほどダウンロードした MicroPython UF2 ファイルを RPI-RP2 のエクスプローラー画面にドラッグ&ドロップします。すると Pico W が再起動して、内部で MicroPython が実行され、USB シリアル経由でプログラムの読み書きと実行ができるようになります。

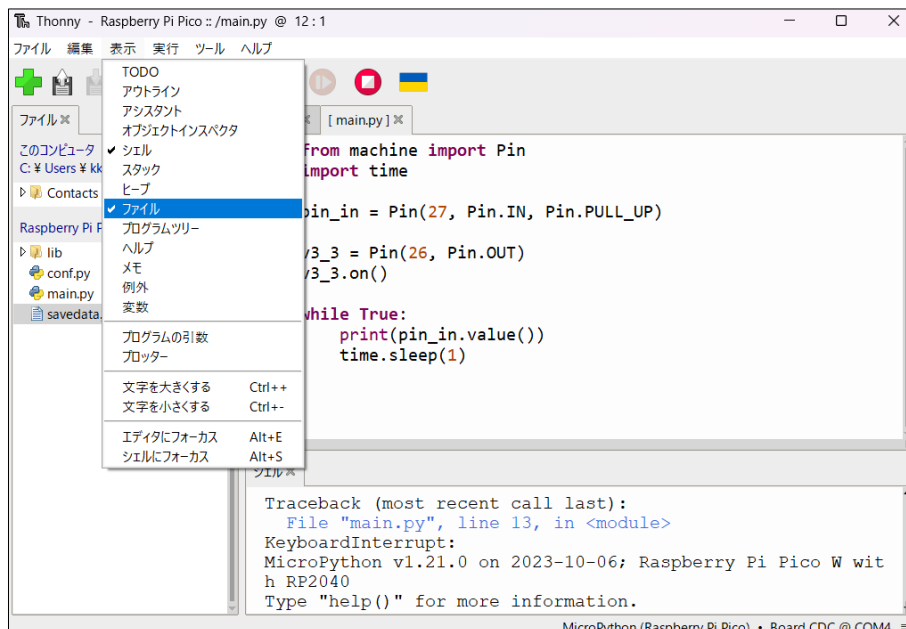
### 3.2. Thonny などの IDE から Pico W にプログラムを書き込む

サンプルプログラムを Pico W に書き込むには、MicroPython 向けの IDE を使うと便利です。ここでは Thonny を例にして説明します。

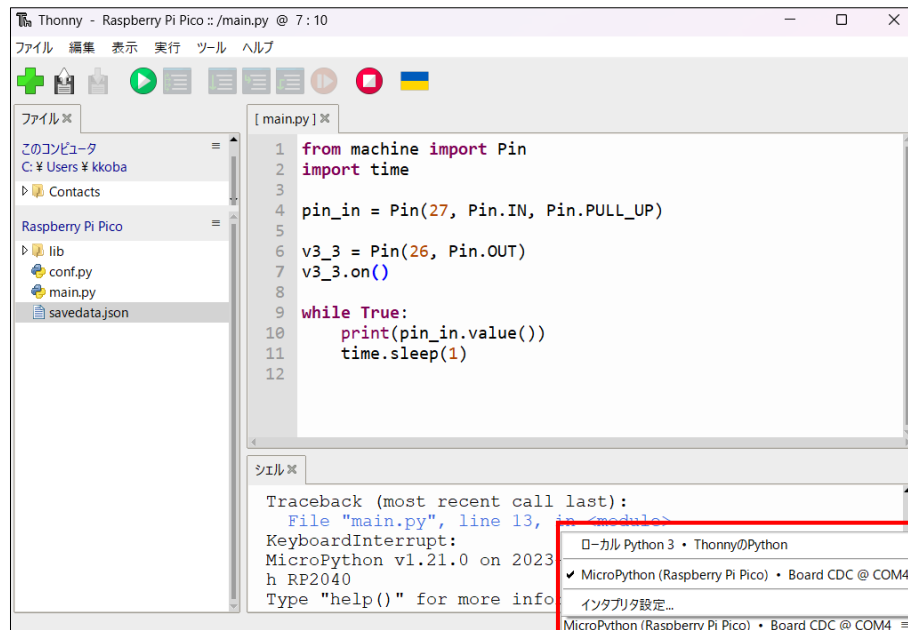
Thonny 公式サイト(<https://thonny.org/>)

Thonny については Web 上に使い方やインストール方法の情報が多くあるので、ここではその説明は行いません。ここでは Thonny のインストールと起動が済んでいるものとして説明します。

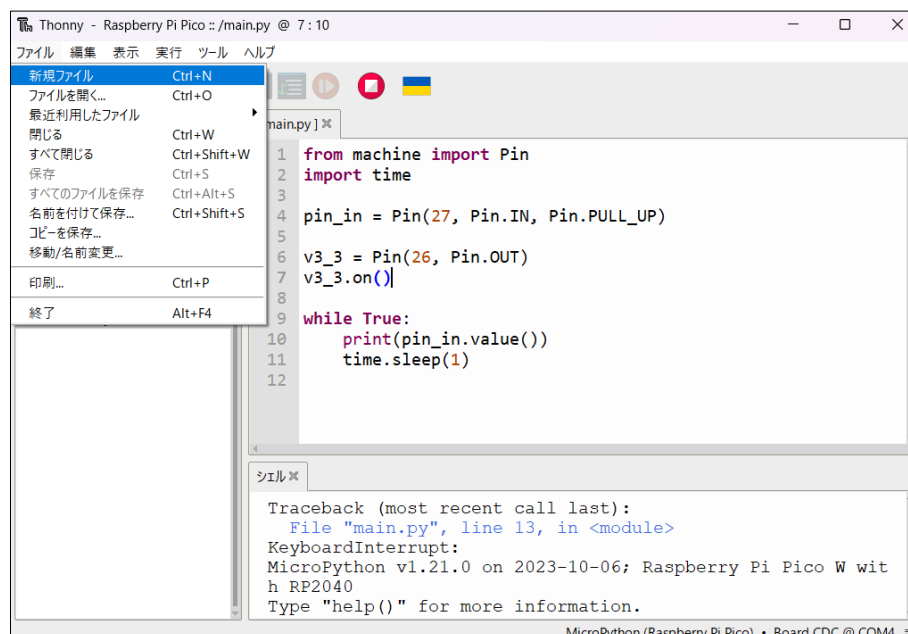
Thonny の表示メニューを開いて、ファイル表示にチェックを入れてください。すると左側にファイルツリーが表示されます。ここでファイルツリーの中に、先ほど PC に接続した Pico W が「Raspberry Pi Pico」として表示されているので、それを確認してください。



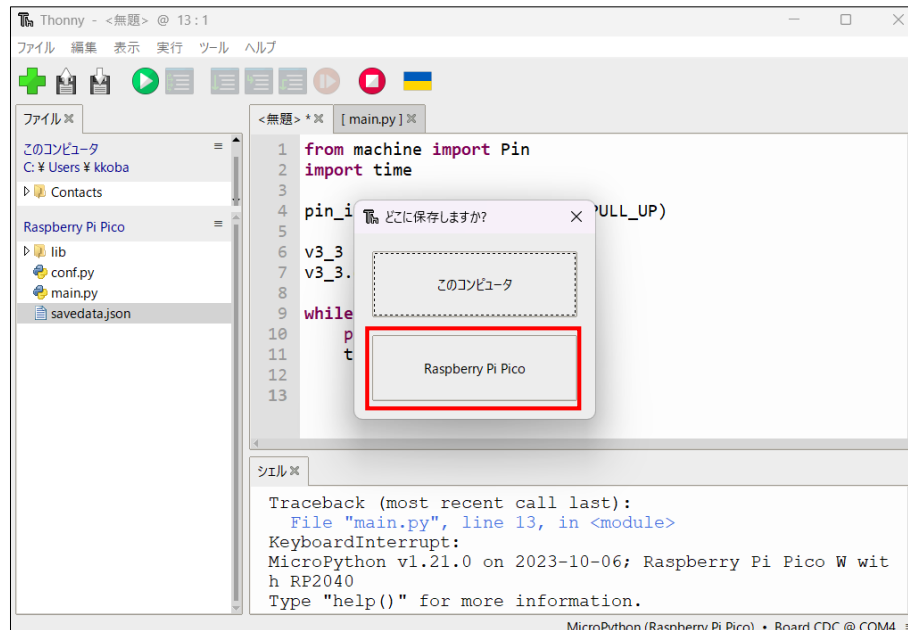
もしもない場合は、Thonny 右下のシリアルポート割り当てメニューを開いて、Pico W が接続されているシリアルポートを選択してください。



Pico W がある事を確認したら、ファイルメニューから「新規ファイル」を選びます。無題のファイルが開くので、インストールしたいサンプルプログラムのコードをそこにコピーします。



無題のファイルを、サンプルプログラムのファイルと同じ名前を付けて Pico W に保存します。これを繰り返して、動かしたいサンプルプログラムのコードをすべて Pico W に保存してください。lib フォルダの作成が必要なプログラムの場合は、ファイルツリーの中で右クリックして「新しいディレクトリ」メニューを選び、lib と名付けて OK ボタンを押すと lib フォルダが出来ます。サンプルプログラムをすべて Pico W に保存したらインストールは完了です。



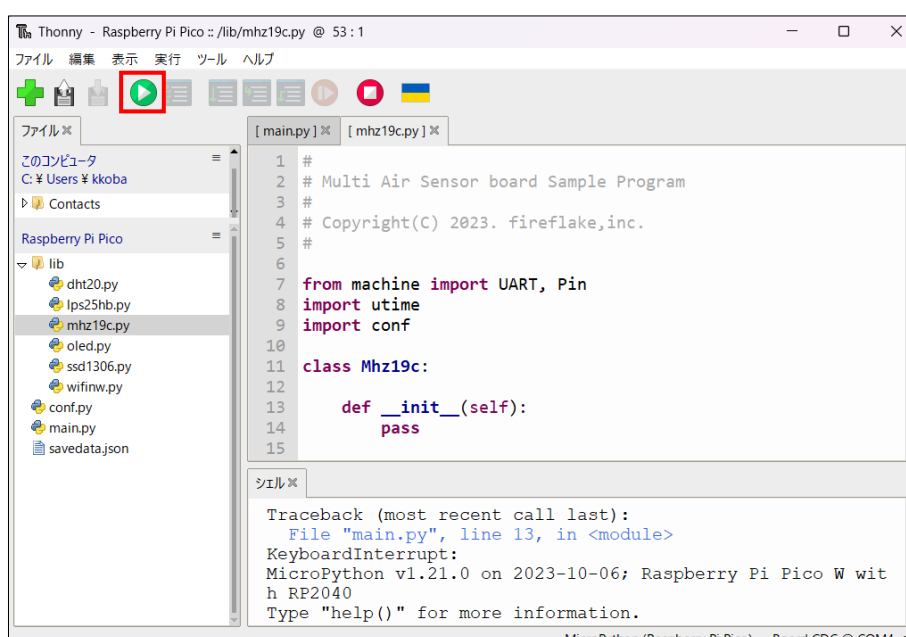


## 4. プログラムの実行方法

ここでは、サンプルプログラムを Pico W 上で実行する方法を説明します。

プログラムの実行には 2 種類の方法があります。ひとつは Pico W と Thonny を接続して、Pico W に入っているプログラムを Thonny で開いて編集して保存し※、その状態で実行ボタンを押す方法です。これを行うと Pico W 上でそのプログラムが実行されます。

※：Pico W に入っているプログラムを Thonny で開いて編集し、保存せずに実行ボタンを押すと、そのプログラムが Pico W のフラッシュメモリに書き込まれたうえで実行されます。ただし複数のファイルを編集してひとつも保存せずに実行ボタンを押すと、フロントに表示されているプログラムだけがフラッシュメモリに書き込まれて実行されます。



もうひとつは、Pico W にただ電源投入する方法です。分かりやすいイメージとしては、Pico W が載ったベースボードに AC アダプタから電源供給して、独立電源で動作させるというものです。これを行うと、自動的に Pico W にインストールされたプログラムが実行されます。この場合、プログラムは必ず main.py が実行されるという制約がある※ので注意してください。またこの方法で動かすと、エラーやログはどこにも表示されません。エラーやログを確認するには、それらを何らかのデバイス（例えば OLED）に表示するなどの仕組みを組んだうえでプログラムを実行する必要があります。

※：boot.py という名前のファイルが Pico W にインストールされていれば、それは main.py より先に動作しますが、べつの目的があって用意されている仕組みなので、通常は main.py から実行してください。先に動作します。

## 5. 各プログラムの説明

ここからは、各サンプルプログラムの説明をします。

サンプルプログラムは、

ff-pico-examples/baseboard/src/以下に、下記の構成で置いてあります。

(工事中と記載してあるものは今後置かれる予定です。)

src

├adc.py

├gpio\_in.py

├gpio\_out.py

├i2c.py

├rs232c\_receive.py

├rs232c\_send.py

├rs485\_receive.py

├rs485\_send.py

└spi.py

サンプルプログラムの一覧は下記になります。

- ADC で電圧測定をする
- フォトカプラへの電圧入力を検知する
- フォトカプラから電圧出力を行う
- I2C で通信する
- RS232C で送受信する
- RS485 で送受信する
- SPI で通信する

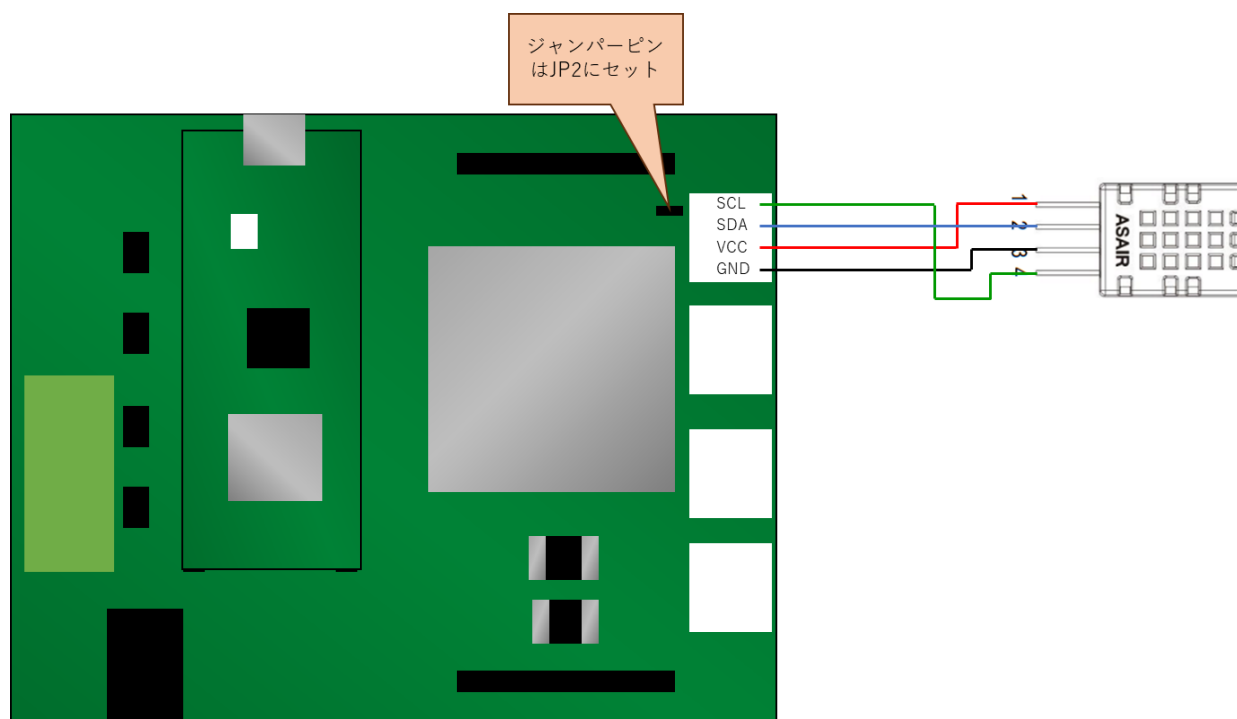
#### ※注意

ベースボードで開発やテストを行う際に、配線を間違えたり過大な電圧をかけると故障に繋がるリスクがあります。

実際にベースボードにセンサーモジュールなどを接続してサンプルプログラムを動作させる前に、あらかじめベースボード取扱説明書とベースボード仕様書をよく読んで、安全に使っていただけるようお願いします。

## 5.1. I2C で通信する

- このプログラムは、ベースボードの I2C 通信ブロックにあるコネクタから I2C デバイスと通信するプログラムです。I2C ブロックの詳細な説明は、ベースボード取扱説明書の「I2C 通信ブロック」のチャプターを参照してください。
- I2C による具体的な通信相手として、DHT20 を接続して以下の配線を行っているものとします。



- Thonny から動作させる場合、下記の構成でサンプルプログラムを Pico W にインストールし、i2c.py を開いている画面で、Thonny の実行ボタンを押してください。

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└dht20.py (※)

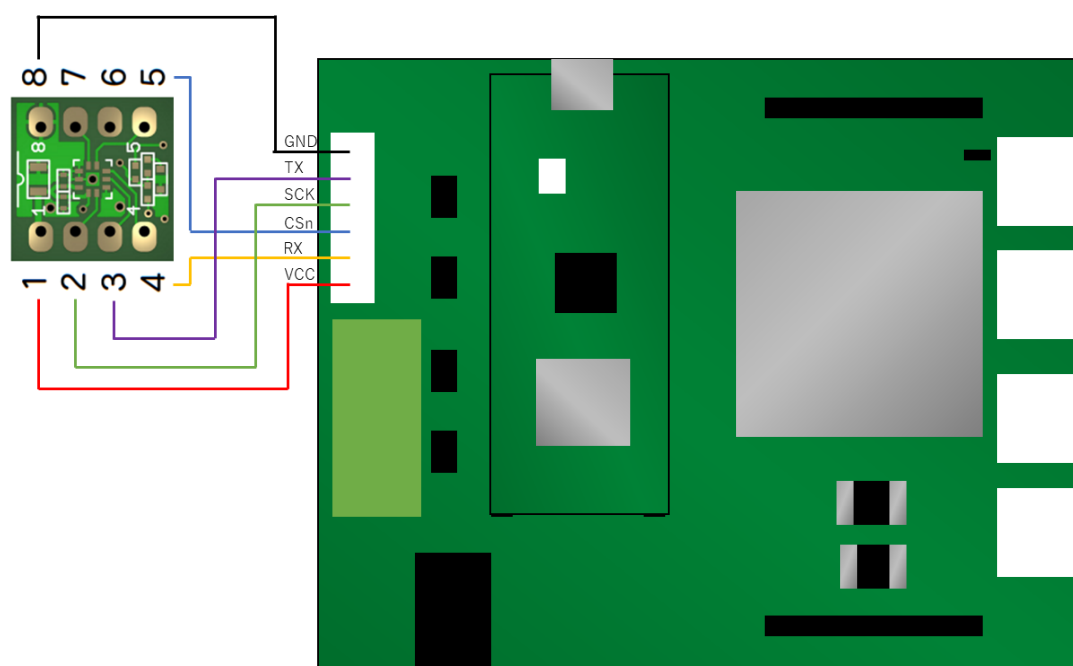
└i2c.py

※: dht20.py については、ff-pico-examples/multiairsensors/src/lib/dht20.py を使ってください。  
また、i2c.py の動作のために dht20.py を使う際には、9 行目の「import conf」をコメントアウトしてください。

- 独立電源で動作させる場合は、上記の構成でサンプルプログラムを Pico W にインストールし、i2c.py のファイル名を main.py に変えたうえで、Pico W に電源投入してください。ただこれをして、通信結果を何らかのディスプレイに表示する仕組みなどがなければ、見た目上は何も起きません。

## 5.2. SPI で通信する

- このプログラムは、ベースボードの SPI 通信ブロックにあるコネクタから SPI デバイスと通信するプログラムです。SPI ブロックの詳細な説明は、ベースボード取扱説明書の「SPI 通信ブロック」のチャプターを参照してください。
- SPI による具体的な通信相手として、秋月電子の LPS25HB 使用 気圧センサーモジュール DIP 化キットを接続して以下の配線を行っているものとします。



- Thonny から動作させる場合、下記の構成でサンプルプログラムを Pico W にインストールし、i2c.py を開いている画面で、Thonny の実行ボタンを押してください。

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└spi.py

- 独立電源で動作させる場合は、上記の構成でサンプルプログラムを Pico W にインストールし、spi.py のファイル名を main.py に変えたうえで、Pico W に電源投入してください。ただこれをして、通信結果を何らかのディスプレイに表示する仕組みなどがなければ、見た目上は何も起きません。

## 5.3. RS232C で通信する

- このプログラムは、ベースボードの RS-232C/RS422 ブロックにある RS-232C I/F で送信または受信を行うプログラムです。具体的なピン配置などについては、ベースボード取扱説明書の「RS-232C/RS422 ブロック」のチャプターを参照してください。

- Thonny から動作させる場合、下記の構成でサンプルプログラムを Pico W にインストールし、rs232c\_receive.py または rs232c\_send.py を開いている画面で、Thonny の実行ボタンを押してください。

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└rs232c\_receive.py

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└rs232c\_send.py

- 独立電源で動作させる場合は、上記の構成でサンプルプログラムを Pico W にインストールし、rs232c\_receive.py または rs232c\_send.py のファイル名を main.py に変えたうえで、Pico W に電源投入してください。ただこれをして、出力した事を何らかのディスプレイに表示する仕組みなどがなければ、見た目上は何も起きません。

#### 5.4. RS485 で通信する

- このプログラムは、ベースボードの RS-232C/RS422 ブロックにある RS-422/485 I/F で送信または受信を行うプログラムです。具体的なピン配置などについては、ベースボード取扱説明書の「RS-232C/RS422 ブロック」の章を参照してください。
- Thonny から動作させる場合、下記の構成でサンプルプログラムを Pico W にインストールし、rs485\_receive.py または rs485\_send.py を開いている画面で、Thonny の実行ボタンを押してください。

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└rs485\_receive.py

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└rs485\_send.py

- 独立電源で動作させる場合は、上記の構成でサンプルプログラムを Pico W にインストールし、rs485\_receive.py または rs485\_send.py のファイル名を main.py に変えたうえで、Pico W に電源投入してください。ただこれをして、出力した事を何らかのディスプレイに表示する仕組みなどがなければ、見た目上は何も起きません。

#### 5.5. フォトカプラへの電圧入力を検知する

- このプログラムは、ベースボードの絶縁型の入出力ブロックにあるフォトカプラへの電圧入力を検知するプログラムです。絶縁型の入出力ブロックの詳細な説明と、具体的な配線例は、ベースボード取扱説明書の「絶縁型の入出力ブロック」の章を参照してください。

- Thonny から動作させる場合、下記の構成でサンプルプログラムを Pico W にインストールし、gpio\_in.py を開いている画面で、Thonny の実行ボタンを押してください。

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└gpio\_in.py

- 独立電源で動作させる場合は、上記の構成でサンプルプログラムを Pico W にインストールし、gpio\_out.py のファイル名を main.py に変えたうえで、Pico W に電源投入してください。ただこれをして、出力した事を何らかのディスプレイに表示する仕組みなどがなければ、見た目上は何も起きません。

## 5.6. フォトカプラから電圧出力を行う

- このプログラムは、ベースボードの絶縁型の入出力ブロックにあるフォトカプラから電圧出力を行うプログラムです。絶縁型の入出力ブロックの詳細な説明と、具体的な配線例は、ベースボード取扱説明書の「絶縁型の入出力ブロック」のチャプターを参照してください。
- Thonny から動作させる場合、下記の構成でサンプルプログラムを Pico W にインストールし、gpio\_out.py を開いている画面で、Thonny の実行ボタンを押してください。

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└gpio\_out.py

- 独立電源で動作させる場合は、上記の構成でサンプルプログラムを Pico W にインストールし、gpio\_in.py のファイル名を main.py に変えたうえで、Pico W に電源投入してください。ただこれをして、入力した事を何らかのディスプレイに表示する仕組みなどがなければ、見た目上は何も起きません。

## 5.7. ADC で電圧測定をする

- このプログラムは、ベースボードのアナログ入力ブロックにある ADC 入力用コネクタに入力された電圧を測定するプログラムです。コネクタはハンダ付けをして取り付ける必要があります。アナログ入力ブロックの詳細な説明は、ベースボード取扱説明書の「アナログ入力ブロック」のチャプターを参照してください。
- Thonny から動作させる場合、下記の構成でサンプルプログラムを Pico W にインストールし、adc.py を開いている画面で、Thonny の実行ボタンを押してください。

.Raspberry Pi Pico (Pico W は Thonny 上ではこのように表記されます)

└adc.py

- 独立電源で動作させる場合は、上記の構成でサンプルプログラムを Pico W にインストールし、gpio\_out.py のファイル名を main.py に変えたうえで、Pico W に電源投入してください。ただこれをして、出力した事を何らかのディスプレイに表示する仕組みなどがなければ、見た目上は何も起きません。



## 6. ライセンスについて

サンプルプログラムのうち、fireflake 制作のプログラムは MIT ライセンスで配布されます。それ以外のプログラムは、それぞれのプログラムが配布されているライセンスに従います。fireflake 制作のプログラムは下記になります。

adc.py  
gpio\_in.py  
gpio\_out.py  
i2c.py  
rs232c\_receive.py  
rs232c\_send.py  
rs485\_receive.py  
rs485\_send.py  
spi.py

## 7. おわりに

本資料が、皆さんがもの作りを楽しめる一助になれば幸いです。Happy Hacking!!