

Step 1: ラズベリーパイピコ W のセットアップ

1.1. はじめに

ラズベリーパイピコ W の初期セットアップ、MicroPython のインストール、Wi-Fi への接続、そして Web ブラウザを通じて LED を制御し、動作確認を行います。

1.2. 必要なもの

- ラズベリーパイピコ W
- Micro-USB ケーブル
- Wi-Fi ネットワークの SSID とパスワード
- ラズベリーパイまたは PC にインストールされた Thonny IDE

1.3. MicroPython のインストール手順

1. ラズベリーパイピコ W をブートローダーモードで接続:

- ピコ W の BOOTSEL ボタンを押しながら、Micro-USB ケーブルでラズベリーパイ若しくは PC に接続します。
- BOOTSEL ボタンを離すと、ラズベリーパイピコ W は USB ドライブとして認識されます。

2. MicroPython のダウンロード:

- Raspberry Pi 公式サイト ([MicroPython - Raspberry Pi Documentation](https://micropython.org/docs/en/latest/raspberry-pi/getting-started/quickstart.html)) から最新の MicroPython UF2 ファイルをダウンロードします。

3. MicroPython のインストール:

- ダウンロードした UF2 ファイルをピコ W の USB ドライブにドラッグ&ドロップします。
- ファイルをコピーすると、ピコ W が自動的に再起動し、MicroPython がインストールされます。

1.4. Wi-Fi 設定と LED 制御のテスト

1. Thonny IDE のインストール:

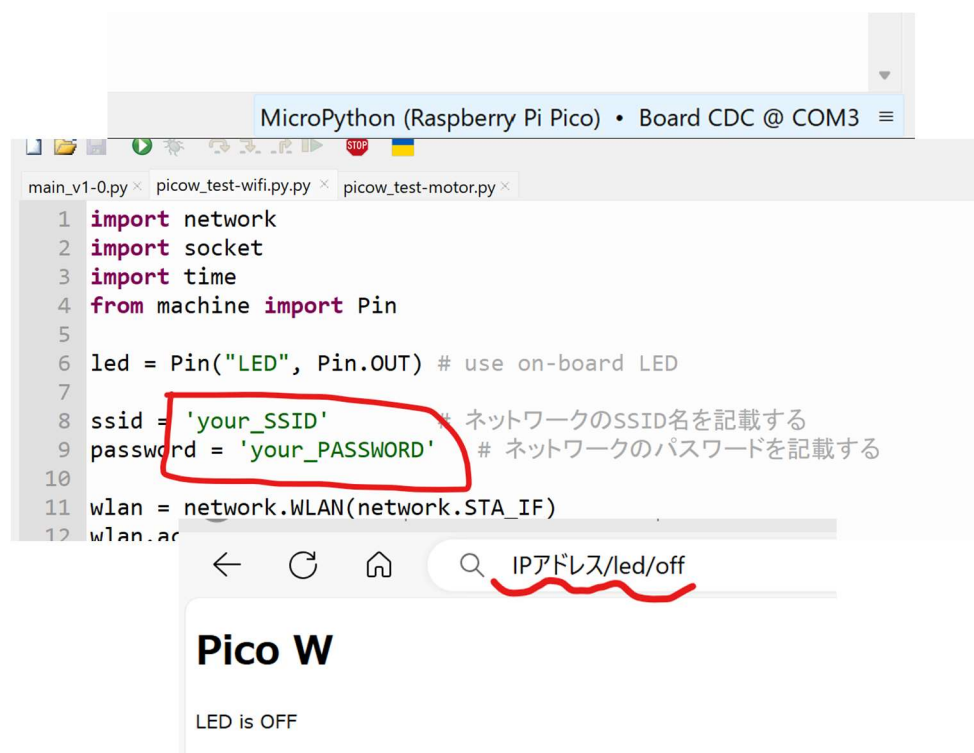
まだ PC に Thonny をインストールしていない場合は、[Thonny 公式サイト](#) からダウンロードしてインストールしてください。

2. Thonny を開き、ピコ W のインタプリタを選択:

Thonny を起動し、ツール > オプション に進み、インタプリタ タブを選択します。インタプリタとして MicroPython (Raspberry Pi Pico) を選択します（選択すると Thonny 右下の表示が下図のようになります。）。

3. Wi-Fi 設定と LED 制御のテスト:

- GitHub リポジトリから `picow_test-wifi.py` ファイルをダウンロードします。
- Thonny でファイルを開きます。
- ファイル内の 8 行目と 9 行目：SSID とパスワードをお使いの Wi-Fi ルータのものに変更します。
- 変更を保存し、スクリプトを実行します。
- スクリプトは Wi-Fi ネットワークに接続し、Web サーバーを起動します。
- 同じネットワークに接続された任意のデバイスの Web ブラウザを開き、ピコ W の IP アドレスを入力します。
- url に "IP アドレス/led/on" と入力するとピコ W の led が光り、"IP アドレス/led/off" と入力すると led が消えます。



Step 2: モータードライバーとモーターのセットアップ

2.1. はじめに

このステップでは、モータードライバーとモーターをラズベリーパイピコ W に接続し、モーターの回転をテストします。

2.2. 必要なもの

- ラズベリーパイピコ W
- モータキット
- モーター用の電源（例: バッテリーパック）
- ラズベリーパイまたは PC にインストールされた Thonny IDE

2.3. ハードウェアの接続

1. 配線図に従う:

GitHub リポジトリにある `connection_v2-0.jpeg` 画像を参照し、ピン接続を行います。

- モータ用電源を Micro USB でブレッドボードに接続します。
- テストの為に PC もしくはラズベリーパイとピコ W を USB で接続するので、モータ用電源からラズベリーパイピコ W への電源供給用ジャンパワイヤは接続不要です。

2. モーターの回転をテストする:

- GitHub リポジトリから `picow_test-motor.py` ファイルをダウンロードします。
- Thonny でファイルを開きます。
- スクリプトを実行します。次の動作が確認できるはずです:
 - 右輪が前進方向に 1 秒回転し、1 秒静止した後、後進方向に 1 秒回転します。
 - 1 秒の静止後に左輪が同じ動作を行います。
- モーターが期待通りに回転しない場合は、モータとモータドライバとのピン接続を変更します。

2.4. トラブルシューティング

- モーターが反応しない、または逆方向に回転する場合は、モータとモータドライバの接続を確認します。
 - ソースファイルの 4~7 行目に記載している PIN とモータドライバとが接続されていることを確認します
 - コンデンサが正しく接続されていることを確認します。
-

Step 3: 加速度センサのセットアップ

3.1. はじめに

このステップでは、加速度センサの計測値を取得して Web ブラウザで確認するためのプログラムを実行します。

3.2. 必要なもの

- Step2 で組み立てたラジコンキット
- ラズベリーパイまたは PC にインストールされた Thonny IDE

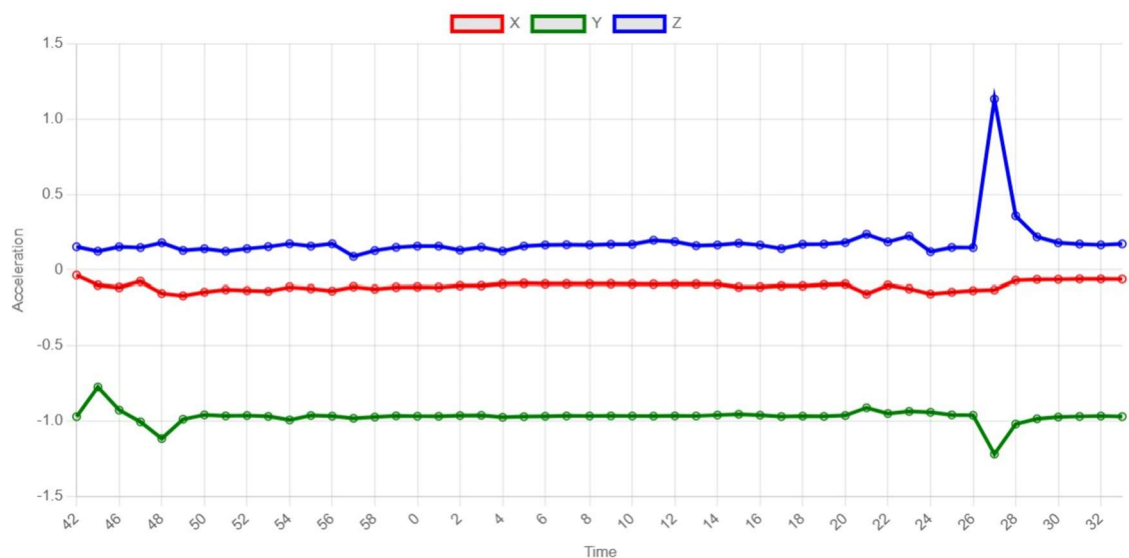
3.3. Web 制御プログラムの実行

1. プログラムのダウンロードと設定:

- GitHub リポジトリから `pico_test-accsensor.py` ファイルをダウンロードします。
- Thonny でファイルを開きます。
- ファイル内の 8 行目と 9 行目: SSID とパスワードをお使いの Wi-Fi ルータのものに変更します。
- 変更を保存し、スクリプトを実行します。

2. 加速度センサーをテストする:

- スクリプトを実行すると、ピコ W が Web サーバーを起動します。
- Web ブラウザを開き、ピコ W の IP アドレスにアクセスします。
- グラフにセンサーで取得した加速度の値がリアルタイムで表示されます。加速度センサの動きに合わせてグラフの値が変わることを確認します。グラフの値更新の間隔はファイル内の 170 行目の数字（デフォルトは 100 ）を変更します。



Step 4: Web ブラウザを介したモーター制御

4.1. はじめに

このステップでは、Web ブラウザを介してモーターを制御するためのプログラムを実行します。

4.2. 必要なもの

- ラズベリーパイピコ W
- Step2 で組み立てたラジコンキット
- モーター用の電源（例: バッテリーパック）
- ラズベリーパイまたは PC にインストールされた Thonny IDE

4.3. Web 制御プログラムの実行

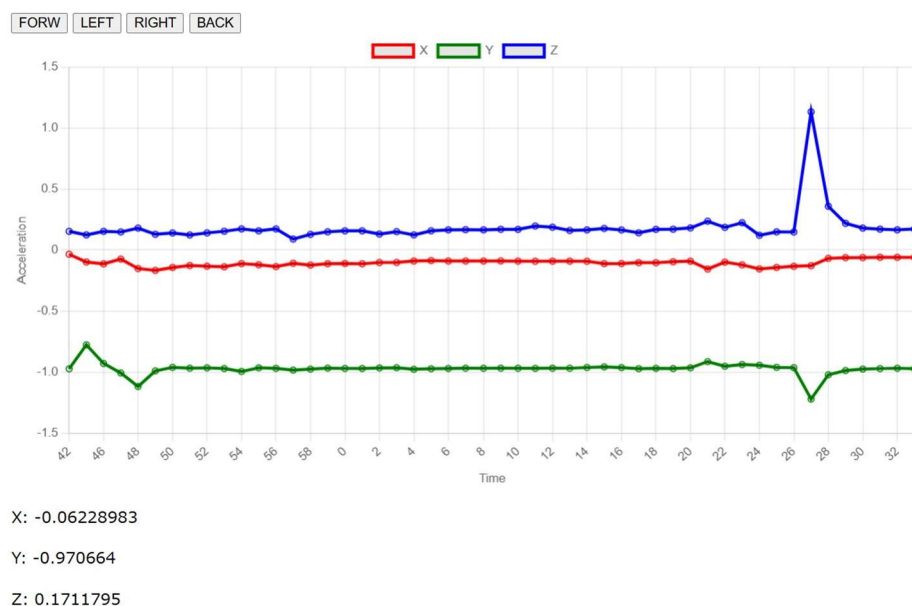
1. プログラムのダウンロードと設定:

- GitHub リポジトリから main_v2-0.py ファイルをダウンロードします。
- Thonny でファイルを開きます。
- ファイル内の 175 行目と 176 行目: SSID とパスワードをお使いの Wi-Fi ルーターのものに変更します。
- 変更を保存し、スクリプトを実行します。

2. Web 制御のテスト:

- スクリプトを実行すると、ピコ W が Web サーバーを起動します。
- Web ブラウザを開き、ピコ W の IP アドレスにアクセスします。
- RC カーを前進、後退、左折、右折させるためのコントローラ（下図）が表示されます。マウスを「FORW」～「BACK」の上に配置するとモータが動きます。

Control by Button



4.4. プログラムの自動実行設定

- 次回以降、ピコ W の電源を入れるとプログラムが自動的に実行されるようにするには：
 - Thonny で main_v2-0.py を main.py にリネームします。
 - main.py ファイルをピコ W にアップロードします。
 - モータ用電源の VCC とピコ W の VSYS をジャンパワイヤで接続します。
 - これで、ピコ W の電源が入るたびに自動的に Web サーバーが起動し、手動でスクリプトを実行することなく RC カーを制御できるようになります。