

## 第6章

ESP32 向けに書いた MicroPython プログラムを Pico 向けに  
CircuitPython で書いてみた

# CircuitPython の 特徴と互換性

角 史生

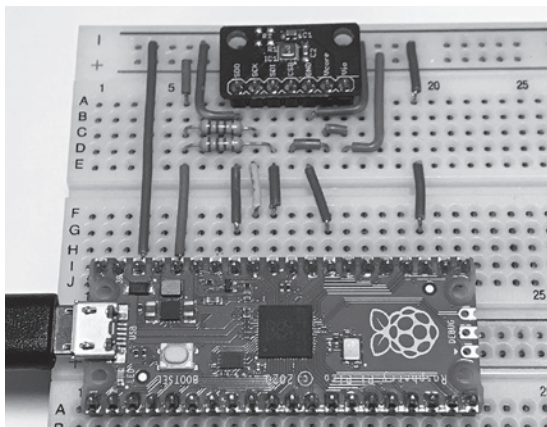


写真1 ESP32 向けに制作した MicroPython プログラムを Pico  
に CircuitPython で移植した

ラズベリー・パイ Pico (以降, Pico) 上で動作する Python 互換のプログラミング言語として, MicroPython と CircuitPython が挙げられます。

MicroPython はハードウェア・リソースが少ないマイコン上でも, Python と同等のプログラミング環境を実現することを目的として, Damien George 氏によって開発されました<sup>(1)</sup>。

CircuitPython は Adafruit 社によって開発が進められ, より使いやすくなるように作り込まれている点が特徴です。

これら2つの言語の特徴を整理し, 2つの言語の互換性について, 実際にプログラミングを行って調べました。また, MicroPython で記述した ESP32 用のソースコードが Pico 版 CircuitPython でも利用可能であるか, どのような修正が発生するか, 連載「逆引き MicroPython プログラム集」<sup>(17)</sup>に掲載したソースコードを Pico に移植して確認しました(写真1)。

## 2つの言語の使いやすさ

CircuitPython は MicroPython をベースに開発されており, コアとなる言語仕様はほぼ同じです。一方,

表1 各言語の特徴

比較項目	MicroPython	CircuitPython
[1] ドキュメント類の充実	○	○
[2] プログラミングの始めやすさ	△注1	○
[3] ライブラリの充実度	○	◎
[4] PIO プログラミング	○	△注2
[5] USB プログラミング	×注3	○
[6] 割り込みプログラミング	○	×

注1: ソース・プログラムを Pico に転送するには adafruit-ampy などのツールが必要

注2: 割り込みプログラミングがサポートされないためプログラマブル I/O からの IRQ を受け取れない (コールバック関数が利用できない)

注3: USB デバイス・サポートについて議論されている状況

提供されるライブラリや開発環境が異なります。MicroPython, CircuitPython の特徴を以下の項目で比較しました。表1は筆者の主観です。

### ● 1, ドキュメント類の充実

MicroPython, CircuitPython とも, 公式のドキュメント・サイトが存在します<sup>(3)(4)</sup>。言語仕様や使いたいモジュールを探し出してモジュールの利用方法を調べるのに役立ちます。GitHub には各言語のソースコードが登録されており参照できます<sup>(5)(6)</sup>。サンプル・コードも掲載されています。

### ● 2, プログラミングの始めやすさ

MicroPython, CircuitPython は, インタープリタ型言語でコンパイル作業が不要なため, 簡単にプログラミングを始められることが特徴です。CircuitPython は Pico に内蔵された USB コントローラをサポートしており, Pico のフラッシュ・メモリを, USB ストレージとして PC に接続できます。これによって REPL (Read-Eval-Print Loop) でプログラムをテストしながら, ソース・ファイルを PC からドラッグ&ドロップで Pico のフラッシュ・メモリに書き込むことが可能です。

このように CircuitPython では, ファイル転送機能を持つ開発用ツールが不要となり, 最低限エディタと