

CircuitPythonの 特徴と互換性

角 史生

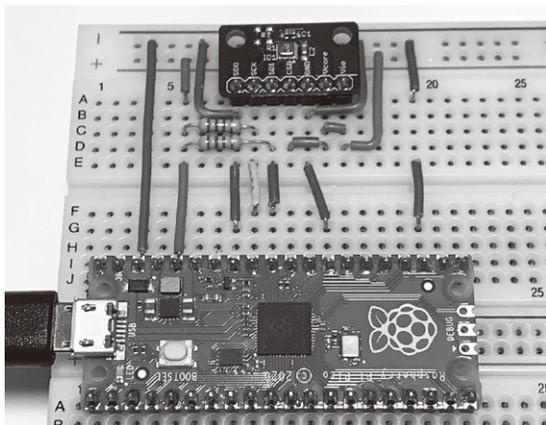


写真1 ESP32向けに制作したMicroPythonプログラムをPicoにCircuitPythonで移植した

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⁽¹⁾。

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」⁽¹⁷⁾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⁽³⁾⁽⁴⁾。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⁽⁵⁾⁽⁶⁾。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