

第1章

MicroPythonと相性抜群のThonny^{ソニー}で始める

開発環境を構築する

宮田 賢一

ラズベリー・パイ Pico (以降, Pico) の開発環境として, C/C++ SDK と MicroPython が公式に提供されています。マイコンとしての機能をフルに使うには, C/C++ SDK が優れていますが, フルセットの Python に近い記述ができ, インタープリタとしての利便性もある MicroPython も, 用途によっては便利に使えます。

ここでは, Pico で MicroPython を使えるようになる方法と, MicroPython に適した統合開発環境を紹介합니다。

実行環境を Pico に書き込む

購入直後の Pico は C/C++ でのプログラミング・モードになっているため, まず MicroPython のファームウェアを Pico にインストールするところから始めます。

● インストール方法

▶ MicroPython のファームウェアをダウンロード

下記の公式サイトから, 最新の MicroPython ファームウェアを PC 上にダウンロードします。

<https://micropython.org/download/rp2-pico/>

本稿執筆時点での安定版最新バージョン (V1.19.1) のファイル名は rp2-pico-20220618-v1.19.1.uf2 です。

▶ 書き込みモードで PC に接続

Pico ボード上の [BOOTSEL] ボタンを押しながら Pico と PC を USB で接続して [BOOTSEL] ボタンを放すと, Pico がファームウェア書き込みモードでブートします。すると Pico は RPI-RP2 という名称の外付けドライブとして PC から見えるようになります。

▶ ファームウェアを Pico に書き込む

ダウンロードした MicroPython のファームウェアを, RPI-RP2 にコピーします。コピーは通常のファイルと同じように Explorer (Windows) や Finder (macOS) を使って行います。

● MicroPython が起動することを確認する

ファームウェアの書き込みが終了すると自動的に Pico が再起動し, MicroPython が起動します。同時に PC からドライブがアンマウントされ, ドライブとして認識されなくなります。

開発環境 Thonny を導入する

この時点で既に PC のシリアル端末から MicroPython のプログラミングが可能ですが, グラフィカルな開発環境 Thonny^{注1} も導入しておきます。

Thonny はシンプルな操作で MicroPython に加え, その派生言語の CircuitPython, PC などでも一般的に使われている CPython のプログラミングができる統合開発環境です。また Windows, macOS, Linux (Raspberry Pi OS を含む) で動作するのも便利です。

● Thonny を使う準備

下記のサイトから, 開発環境の OS に対応するインストーラをダウンロードし, 実行します。

<https://thonny.org/>

▶ Thonny を起動

インストールが完了したら, MicroPython を使えるようにした Pico を, PC に接続した状態で Thonny を起動します。すると Thonny が MicroPython を実行している Pico に接続し, プログラムの入力が可能となります。

▶ Thonny の初期設定

MicroPython でプログラムする場合は, 特設の設定は不要ですが, 本稿では以下の設定がなされていることを前提とします。

・メニューの日本語化

メニューの [Toos]-[Options] でオプション設定画面を開きます。「Language」項目で「日本語 [ALPHA]」を選び Thonny を再起動します。画面上部にメニューが表示

注1: 「ソニー」と発音するようです。Python の「tho」と同じ発音です。

開発環境

I/O
プログラミング

USB

OS
リアルタイム

人工知能

活用事例

実験
RP2040基礎知識
MicroPython拡張モジュール
MicroPython活用事例
PICO W