

Pico W アプリの開発方法③

…MicroPython

関本 健太郎

リスト1 MSYS2のシェルからMicroPythonのビルドに必要なパッケージをインストールする

```
pacman -Syu
pacman -S base-devel mingw-w64-x86_64-toolchain mingw-w64-x86_64-arm-none-eabi-toolchain autoconf automake gcc
libtool zlib git mingw-w64-x86_64-cmake mingw-w64-x86_64-doxygen mingw-w64-x86_64-ninja
```

(a) MSYS2-Mingw64環境の場合

```
pacman -Syu
pacman -S base-devel mingw-w64-i686-toolchain mingw-w64-i686-arm-none-eabi-toolchain autoconf automake gcc
libtool zlib git mingw-w64-i686-cmake mingw-w64-i686-doxygen mingw-w64-i686-ninja
```

(b) MSYS2-Mingw32環境の場合

この章では、マイコン向けの開発言語MicroPythonを使ってBluetoothアプリケーションを開発します。MicroPythonはPythonのサブセットなので、Pythonを知っていれば容易にマイコン・プログラミングができます。また、C言語でBTstackを使って実装したときとサイズを比較すると、同じプログラムが1/3以下の行数で実装できます。

ここでは、第2章でも作成したGATTサーバをMicroPythonで作成し、Bluetooth経由でLチカを行います。(編集部)

Pico W向けのファームウェア

●現時点ではNightly builds版が必要

本稿執筆時点の最新MicroPython v1.20では、ラズベリー・パイPico W(以降、Pico W)向けのBluetooth機能はサポートされていませんでした。v1.20.0(2023-04-26)の追加修正で、Pico W向けのBluetooth機能がサポートされました。ただし、公開されているPico W向けのファームウェアには、まだ追加修正が含まれていません^{注1}。次節のファームウェアのビルド手順を行わない場合には、Nightly builds版を下記MicroPythonのウェブ・ページからダウンロードしてください。

注1: 本誌発売直前にMicroPython v1.21.0(2023-10-05版)が公開されました。第1部 Appendix 2でv1.21.0を取り上げています。

https://micropython.org/download/RPI_PICO_W/

ファームウェアのビルド手順

この節では、最新の追加修正を含むv1.20.0を独自ビルドする方法を紹介します。ビルドには、BTstackのサンプルのビルドと同じようにWindows 10のMSYS2-Mingw64(64ビット版)またはMSYS2-Mingw32(32ビット版)の環境を使います。

以下の手順で行います。

- MSYS2-Mingw64またはMSYS2-Mingw32をインストールする
<https://www.msys2.org/>
- MicroPython(最新版またはv1.20)をGitクローンする
- mpy-crossをビルドする
- rp2ポートのPICO_Wボードを指定してビルドを実行する

●MSYS2のインストール

MSYS2のインストーラ(本稿執筆時msys2-x86_64-20230718.exe)を実行します⁽²⁾。

インストール後、MSYS2 MINGW64 shellを起動し、リスト1のようにMicroPythonのビルドに必要なMSYS2-Mingw64またはMSYS2-Mingw32パッケージをインストールします。