

# スクリプト環境 MicroPythonの使い方

ご購入はこちら

西本 卓也

本稿では秋月電子通商から販売されているESP-WROOM-32開発ボードESP32-DevKitC(以後「ESP32ボード」と呼ぶ)を使用します。

作業にはWindows Subsystem for Linux (WSL) のUbuntu (bash) 環境を使用します。WSLの正式版はWindows 10 (64ビット版) のVersion 1709以降で利用できます。WSLベータ版ではシリアル・ポートが利用できませんでしたが、正式版では例えば/dev/ttyS1でCOM1ポートを利用できるようになり、オープンソースを活用したマイコン開発作業が容易になりました。Ubuntu 18.04でも確認しています。

なおWindowsコマンド・プロンプトやmacOSで作業する場合については、必要に応じて注記します。

## MicroPython公式バイナリを書き込む

ESP32用MicroPythonのバイナリは下記で公開されています。

<http://micropython.org/download#esp32>

ここには例えば以下のような2種類のファイルがあります。

- esp32-20180514-v1.9.4-8-g749b1617.bin
- esp32-20180511-v1.9.4.bin

安定版1.9.4として公開されているのは後者のファイルなので、こちらをダウンロードします。

システムのPython 3環境にesptool 2.3.1 (<https://pypi.org/project/esptool/>) をインストールします。

```
$ sudo apt-get update
$ sudo apt-get install python3-pip
$ sudo /usr/bin/python3 -m pip install-U pip
$ sudo /usr/bin/python3 -m pip install esptool
$ which esptool.py
/usr/local/bin/esptool.py
```

ESP32ボードのMicro-USBポートとPCのUSBポートを接続します。「Windows+X」→「デバイス マネージャー」→「ポート (COMとLPT)」から認識されたポート番号を調べます(図1)。

筆者の環境では「Silicon Labs CP210x USB to UART Bridge (COM4)」が表示されました。

ポート番号4のCOMデバイスはWSLのデバイス/dev/ttyS4としてアクセスできます。デバイスはあとで繰り返し使うので環境変数に入れておきます。

```
$ export PORT=/dev/ttyS4
```

最後の数字はCOMポートの番号に合わせてください。

転送速度も環境変数に入れておきます。WSLの場合はこの値を使うと失敗しにくいようです。

```
$ export BAUD=115200
```

一般ユーザがWSLでシリアル・ポートに書き込めるようにします。この操作はPCとESP32ボードを接続し直すたびに必要です。

```
$ sudo chmod 777 $PORT
```

まずESP32のフラッシュ・メモリを消去します。

```
$ esptool.py --chip esp32 -p $PORT -b $BAUD erase_flash
```

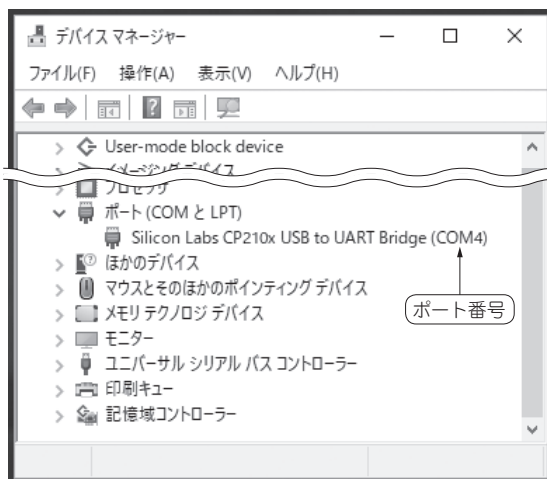


図1 デバイス・マネージャでポート番号を調べる