第3章

マイコン用超軽量インタプリタ×Wi-Fiアルデュイーノで IoTまで10分！

BASIC風1行リターン・プログラミング MicroPython入門

コンパイラもデバッガも不要

白阪 一郎 Ichiro Shirasaka

Arduino IDEではC++ベースの専用言語「スケッチ」を使ってプログラムを記述しますが，ソースコードを書き換えるたびにコンパイルし，実行ファイルをESP32に書き込む必要があります．プログラム開発中は，頻繁にソースコードに手を加えるので，何度もコンパイルと書き込みを実行します．数行の小規模なプログラムなら問題ありませんが，大規模になるとコンパイルと書き込みの待ち時間が長くなり，開発効率が悪くなります．ソースコードを実行時に解読して動作するインタプリタを使えば，ほとんど待ち時間なしで修正がうまくいったかどうかを確認できます．

本稿では，インタプリタ型のプログラミング言語でありながら，Tiny BASICなどにはない文字列や小数点などを扱う言語機能を備えたPythonを使ってESP32を開発する方法を解説します． 〈編集部〉

インタプリタ型プログラミング言語 Pythonとは

● メリット

▶その1：コマンドで1行ずつ実行

Pythonはプログラミング言語の1つです．往年のBASICと同じインタプリタ型の言語なので，キーボードから**入力されたコマンドを実行時に解読して動作します**．コマンドを数行入力するだけで，LEDをON/OFFしたり，入力端子の電圧値の読み取りができます．

▶その2：専用機器なしでデバッグできる

ArduinoのスケッチやCプログラムのように，大規模なアプリケーションも作れます．コンパイルが不要で，1行ごとに動かせるので，**専用のデバッガは不要です**．初めて使う部品も動きを確かめながら少しずつ動かせます．

▶その3：格安Linuxコンピュータ ラズベリー・パイに標準装備されている

ラズベリー・パイの標準OS Raspbianには，最初からPythonがインストールされています．ラズベリー・パイでは周辺機器を制御するプログラムとしてよく使われています．

● マイコンでも使える軽量Pyhton登場！

Pythonを動かすには，ラズベリー・パイに搭載されているような1 GHz前後で動作する比較的高性能なプロセッサが必要です．OSの動いているボードは，起動やシャットダウンに時間がかかったり，消費電力が大きかったりするので，ロボット制御やセンサ・ネットワーク制御を行う機器のような小規模な組み込み機器には向いていません．

OSレスのマイコンでも動作するように設計されたPythonインタプリタ「MicroPython」が登場しました．

MicroPythonは，ARMマイコンを使った専用開発ボードpyboard向けに開発された小規模なPythonインタプリタです．現在ではさまざまなマイコン向けにポーティングされています．

ESP8266向けのMicroPythonは数年前からリリースされていましたが，2017年5月よりESP-WROOM-32向けのリリースも開始しました．Wi-Fiなどのネットワーク機能やA-Dコンバータ，I^2C，SPIなどの周辺機能を一通り使えます．

ESP-WROOM-32用のMicroPythonは，メーカ純正の開発環境であるESP-IDFでビルドされています．

IoT Express上で MicroPythonを使えるようにする

本誌付録のIoT ExpressでもMicroPythonを動かすことができます．ESP-WROOM-32は，Wi-Fi通信やデュアル・コアCPUなど本家pyboardよりも豊富な機能を備えているので，さまざまな応用が可能です．ここでは，IoT ExpressでMicroPythonを動かす方法を紹介します．

● ステップ1：MicroPythonの実行環境をインストール

▶手順1：ダウンロード

MicroPythonの公式Webページから，ESP-WROOM-32用のMicroPythonインタプリタを入手します．URLは次のとおりです．

https://micropython.org/download#esp32

▶手順2：ESP-WROOM-32への書き込み

手順1で入手したMicroPythonインタプリタをESP-WROOM-32の内蔵SPIフラッシュ・メモリに書き