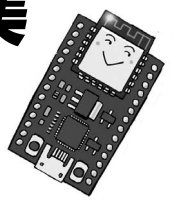
700円マイコンESP32ではじめる

逆引きMicroPythonプログラム集

新連載 第1回

開発環境を整える



角 史生

MicroPythonは，プログラミング言語Pythonの実装の1つで，マイコン上での動作に最適化された言語処理系です．Python3と高い互換性を持っているので，マイコンに慣れていない初心者でも開発しやすいという特徴を持ちます．

連載では，スイッチをつなぎたい，センサ値を読みたい，LCDに表示したいなどの目的別にMicroPythonプログラムを紹介します（**表1**）．Wi-Fi/Bluetooth接続機能を持ち，700円で購入できる無線マイコン・モジュールESP32-WROOM-32D（Espressif Systems）を実際に動かしながら解説します．

MicroPythonがプロトタイプ開発に向いている理由

MicroPythonはリソースの少ないマイコン上でPython3と同じようにプログラミングできる環境の実現を目指して開発されました．MicroPythonの特徴が活かせる開発用途としてプロトタイプ開発が挙げられます．プロトタイプ開発では，試作，テスト，修正を繰り返しながら開発が進みますが，MicroPythonを用いることで得られるメリットを次に整理します．

● 理由1…対話インタープリタ・モードが使える

Python/MicroPythonはインタープリタ型言語です．インタープリタは，マイコンやコンピュータで解釈できるように変換し実行する機能があります．このため，コンパイル不要で実行したいコードや確認したい変数をコンソールから入力すると，すぐに実行され結果が得られます．この機能は対話インタープリタ・モードREPLと呼ばれます（Read，Evaluate，Print，Loop）．対話インタープリタ・モードを活用することでプロトタイプ開発を効率良く進めることができます．

CMOSカメラ・モジュールなどの周辺機器を制御するソフトウェアを開発する際に，コンパイル型言語と，インタープリタ型言語による難易度の違いを比較してみましょう．

まず仕様書やサンプル・コードを参照して，制御レジスタの操作方法や設定パラメータを理解し，テスト・プログラムを作成します．テストを繰り返しながら希望した動作になるまでソフトウェアを修正します．

▶コンパイル型言語での開発の場合

C言語やArduinoなどのコンパイル型言語でのソフトウェア開発を**図1**に示します．

1. PC上でプログラムを作成
2. コンパイラによりマイコンで実行可能なファイルに変換（コンパイル）
3. 変換したバイナリ・ファイルをマイコンのフラッシュ・メモリに書き込む
4. プログラムを実行
5. 実行結果を確認し，問題があれば1に戻る．希望した動きになるまでプログラム修正，コンパイル，テストを繰り返す．

▶インタープリタ型言語による開発の場合

インタープリタ型言語による開発の例を**図2**に示します．

1. プログラムをREPLに入力，または，コピー＆ペースト．
2. インタープリタによりプログラムを実行．
3. 実行結果を確認し，問題があれば1に戻る．希望した動きをするまでプログラムを修正，実行を繰り返す．

表1　逆引きMicroPythonプログラム集で扱うテーマ

連載［回］	テーマ
1	開発環境の準備
2	スイッチ入力/ボリューム入力
3	センサ入力
4	PWM出力で光る/動く/鳴る
5	ディスプレイ出力
6	シリアル通信
7	ファイル操作
8	クラウド通信/サーバ機能
9	スリープ機能
10	タイマ機能
11	コード改善とメモリ管理