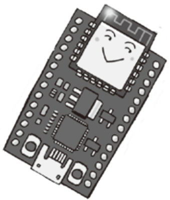
700円マイコンESP32ではじめる

逆引きMicroPythonプログラム集



第13回
最終回

性能改善②…メモリ使用量の可視化と軽量化

角 史生

```
# gcモジュールをインポート
>>> import gc↵
# ヒープ・メモリの空き容量を取得
>>> gc.mem_free()↵
109648  # 空き容量は109648バイト
```

図1[注1] **ヒープ・メモリの空き容量をREPLで確認している様子**
gcモジュールの関数mem_free()を使えば確認できる

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た言語処理系で，プロトタイプ開発に向いています．

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を進めるので，MicroPythonを用いることでトライ&エラーが容易になります．本連載ではESP32-WROOM-32（Espressif Systems）を搭載する開発ボードESP32-DevKitC（Espressif Systems，以降はESP32と表記）を使って，用途別にMicroPythonの使用例を紹介します．

10-4　ヒープ・メモリ使用量の把握と改善

● プログラム実行時に確保する「ヒープ・メモリ」

MicroPythonでは，メモリ管理をMicroPython側で行います．メモリ割り当てや不要になったメモリの回収は，MicroPythonにより自動で行われます．この不要メモリの回収は，ガベージ・コレクションと呼ばれます．プログラムの実行に連動して割り当て，解放，回収されるメモリをヒープ・メモリと呼びます．

ヒープ・メモリに十分余裕のある状態で実行できるプログラムであればメモリ使用量の分析は不要です．

```
# micropythonモジュールをインポート
>>> import micropython↵
# メモリ情報を表示
>>> micropython.mem_info()↵
stack: 704 out of 15360
GC: total: 111168, used: 1440, free: 109728
 No. of 1-blocks: 17, 2-blocks: 10, max blk sz: 18,
                                   max free sz: 6847
# スタック15360バイト中，704バイト使用
# ヒープ・メモリ 111168バイト中，1440バイト使用
# 空き容量は109728バイト
```

図2　ヒープ・メモリの総容量をREPLで確認している様子
micropythonモジュールの関数mem_info()を使えば確認できる

テスト開始時点では安定動作していたのに，長時間動作させると不安定になる場合は，ヒープ・メモリが枯渇している可能性があります．このような場合，実行中のヒープ・メモリの使用量を把握して，メモリを大量に使用している処理の特定と対策が必要です．

● ヒープ・メモリの空き容量を計測する

ヒープ・メモリの空き容量は，gcモジュールの関数mem_free()を使えば確認できます．**図1**にREPL（Read-Eval-Print Loop）による確認操作の様子を示します．

ヒープ・メモリの総容量を知りたい場合は，micropythonモジュールの関数mem_info()を使います．関数mem_info()は，ヒープ・メモリ以外にスタックの情報も表示します．**図2**にREPLによる確認操作の様子を示します．この操作により，ヒープ・メモリの容量は約100Kバイトであることが分かります．

micropythonモジュールの関数mem_info()は，詳細な情報が得られるメリットがありますが，表示専用の関数なので戻り値がありません．集計などの目的で空き容量を数値で取得したい場合は，gcモジュールの関数mem_free()を使います．

調べたい箇所に次のようなコードを追加してプログラムを実行し，REPL画面で確認することで，ヒープ・メモリの空き容量の推移が分かります．

`print(gc.mem_free())`

注1：今回の動作例では，gcモジュールをインポートしていますが，MicroPythonのバージョンによっては起動直後から使える場合があるので1行目のimport文が不要になります．インポート済みかどうかは，定義済みの名前一覧を返す関数dir()，またはglobals()を使えば確認できます．

第4回　PWM出力機能を使ったLED調光，モータ制御，電子音再生（2021年7月号）
第5回　小型液晶ディスプレイに文字や線を表示する（2021年10月号）
第6回　定番SPI/I^{2}C/UARTによるシリアル通信（2021年12月号）