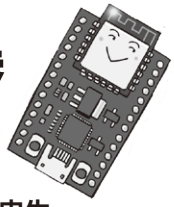
ご購入はこちら
ダウンロード・データあります

700円マイコンESP32ではじめる

逆引きMicroPythonプログラム集



第12回 性能改善①…プログラム実行の高速化

角 史生

MicroPythonは，リソースの少ないマイコン上でPython 3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．

● 処理時間やメモリ使用状況の把握が重要

MicroPythonで開発したシステムを動かしてみると，処理速度が遅かったり，長時間動作でメモリが枯渇してException（例外）などの異常が発生する場合があります．システムの性能改善や動作安定化には多くのノウハウが必要ですが，まずは速度の低下やメモリの大量使用がどの処理で発生しているのかを特定することが重要です．

Pythonにはプロファイラが用意されているので，処理時間やメモリ使用量を容易に把握できますが，MicroPythonには用意されていません．そのため，関数を工夫するなどして，自力で原因箇所を特定する必要があります．

今回は，処理速度に関する基本的な調査方法と性能改善のヒントを紹介します．性能改善に関する情報は，MicroPythonの公式ドキュメントである文献(1)も参考になります．メモリ使用量の把握と低減については次回解説します．

10-1 関数の実行時間を計測する

● 実行時間計測用の関数を用意する

実行時間計測用関数timed_function()を使えば，対象の関数の実行時間を計測できます．**リスト1**にtimed_function()を示します．

まず，計測対象の関数に対して，計測関数timed_function()をデコレータとして設定します．**リスト2**に示すのは，計測対象の関数をwait_test()にした場合の例です．

リスト1と**リスト2**に示す関数をREPL内で定義した後，調査対象の関数wait_test()を実行すると，実行時間が表示されます．**図1**にREPLでの実行結果を示します．wait_test関数の実行時間が1234.260msであることが分かりました．

● 実行時間を計測する仕組み

計測関数timed_function()がデコレータとして適用されることで，変数wait_testには計測対

リスト1(1) 関数の実行時間を計測する関数timed_function()の内容

```
import utime
def timed_function(f, *args, **kwargs):
    myname = str(f).split(' ')[1]
    def new_func(*args, **kwargs):
        # 計測開始時刻を取得
        t = utime.ticks_us()
        # 計測対象の関数を実行
        result = f(*args, **kwargs)
        # 実行時間を算出
        delta = utime.ticks_diff(utime.ticks_us(), t)
        # 計測結果を表示
        print('Function {} Time = {:6.3f}ms'.
                        format(myname, delta/1000))
        # 計測対象関数の実行結果を返却
        return result
    # 計測機能付きの関数（クロージャ）を返却
    return new_func
```

リスト2 timed_function()を使って関数の実行時間を計測するプログラム

計測対象の関数wait_test()に対して，timed_function()をデコレータとして設定することで実行時間を計測する

```
import utime

@timed_function  # 調査対象関数に対して，timed_functionを
def  wait_test(wait_time, count):  #デコレータとして設定
    for _ in range(count):
        utime.sleep(wait_time)
                                # インデントとして空白8文字必要
```


第1回 開発環境を整える（2021年4月号）
第2回 スイッチやボリューム，ロータリ・エンコーダによる入力検出（2021年5月号）
第3回 静電容量，磁気，赤外線（人感），温湿度／気圧の検出（2021年6月号）