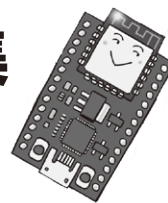


700円マイコンESP32ではじめる

逆引きMicroPythonプログラム集



第11回 時間管理…日時取得/ウェイト/タイマ

角 史生

リスト1 localtime関数を使用して現在時刻を表示するプログラム
UTC (協定世界時) を取得する。取得したUTCに9時間を加算することでJST (日本標準時) を計算している

```
import ntptime
import utime

ntptime.settime()          # NTPによる時刻同期
JST_OFFSET = 9 * 60 * 60   # JST = UTC + 9H(32400秒)

# 時刻を取得、UTCで表示する
(year, month, day, hour, min, sec, wd, yd) =
    utime.localtime()
print(f"{year:4d}-{month:02d}-{day:02d}
      {hour:02d}:{min:02d}:{sec:02d} (UTC)")

# 時刻を取得、JSTで表示する
(year, month, day, hour, min, sec, wd, yd) =
    utime.localtime(utime.time() + JST_OFFSET)
print(f"{year:4d}-{month:02d}-{day:02d}
      {hour:02d}:{min:02d}:{sec:02d} (JST)")
```

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。

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。

今回はMicroPythonで日時の取得、時間管理をする方法を解説します。

9-1 NTPサーバと同期して時刻を表示する

● サーバと同期して正しい時刻を表示する

標準ライブラリのntptimeモジュールを用いることで、時刻管理を行うNTP (Network Time Protocol) サーバとの同期を行えます。NTPを利用するにはESP32がインターネットに接続されている必要があります。接続方法は本連載の第9回 (2022年5月号) か、文献(2)のネットワークングの章を参照してください。

● プログラム

settime関数によりNTPサーバと同期し、

リスト2 sleep関数を使用したウェイト処理の例

```
import utime
utime.sleep(3)          # 3s待つ
utime.sleep_ms(100)    # 100ms待つ
utime.sleep_us(100)   # 100μs待つ

print("wait for 1sec...")
utime.sleep(1)
print("done")
```

utimeモジュールのlocaltime関数を実行することでUTC (協定世界時) を取得できます。しかし、この段階では、日本の標準時であるJST (日本標準時) とは9時間の時差があります。この時差を加算することでJSTにします。

MicroPythonには地域情報 (ロケール機構) のライブラリが実装されていません。そこで、utime.time関数を用い、取得したUTCの時間に32400秒 [60 (秒) × 60 (分) × 9 (時間)] を加算することでJSTに変換します。

本プログラムをリスト1^{注1}に示します。実行した結果は次のようになります。上段がUTCで下段がJSTです。

```
2022-04-29 00:09:36 (UTC)
```

```
2022-04-29 09:09:36 (JST)
```

9-2 ウェイト処理をする

● 処理の実行を待機したいときに使う

utimeモジュールのsleep関数を使うことで、s単位、ms単位、μs単位で待機します。

● プログラム

プログラムをリスト2に示します。utime.sleep、utime.sleep_ms、utime.sleep_usのそれぞれの引数に数字を入力することで、s単位、ms単位、μs

注1: リスト1は文字列の整形にformat関数を使わず、フォーマット済み文字列リテラル (f-string) を使っています。f-stringを使うことで文字列の整形指示が簡潔に行えます。2022年4月時点の最新版ファームウェア、v1.18 (2022-01-17) で動作確認しています。

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/I²C/UARTによるシリアル通信 (2021年12月号)

第7回 UART接続例…①ESP32同士②音声合成ICとEPS32 (2022年3月号)