

ラズパイは不得意な世界

第4章

ESP32リアルタイム処理

井田 健太

研究すること…ESP32マイコンの「正確な周期処理」

センサからデータを読み取ったり、モータなどを制御する場合、十分な正確さで周期的に処理を実行する必要があります。こういった周期処理を実行するために、大抵のマイクロ・コントローラはハードウェア・タイマによるタイマ割り込み機能を持っており、ESP32にも同様の機能があります。

一方、ESP32の標準開発環境であるESP-IDFには、リアルタイムOSであるFreeRTOSが組み込まれており、開発したプログラムでは、ユーザが記述した処理以外にもESP-IDFの処理が実行されています。このため、前述のセンサやモータに関する処理を行うために十分な正確さで周期的な処理を実行するためには、幾つか注意する点があります。

特に、ESP32の無線通信処理と並行して、十分な正確さで周期的な処理を実行するためには、幾つか注意点があります。これらの注意点を押さえておくことにより、ESP32でセンサやモータの処理を実装できるようになります。

● 環境

実験環境や機材は以下の通りです。

OS : Ubuntu 18.04 LTS

開発環境 : ESP-IDF v3.3^{注1}

デバイス : M5StickC (ESP32-PICO) × 2

タイム・スタンプ関数の時間精度を調べる

タイマの正確性を測定する前に、ESP32の機能を使って、どれくらい正確に時間を調べられるのか検討しなければなりません。ESP32の時間取得関数を使ってタイマ精度の測定を行うからです。この時間取得関数の精度が悪ければ、測定には別の方法を使わなければなりません。

ESP-IDFには、タイム・スタンプを取得するAPIとして、`esp_timer_get_time`が用意されています。このAPIはESP32が起動してからの経過時間を μs (マイクロ秒)単位で返します。 μs 単位のタイム・スタンプを返すという仕様だけを見れば、タイマの精度の測定に十分使えるように見えます。しかし、処理時間を計る場合、次の2つの要素を考慮する必要があります。

- `esp_timer_get_time`を呼び出してからAPIの内部で実際のタイム・スタンプの取得までにどれくらいの時間がかかるか
- `esp_timer_get_time`の処理にかかる時間はどれくらいか

この2点について、それぞれ十分に短い時間である必要があるので、`esp_timer_get_time`がどのように実装されているのかを確認し、その精度を見てみます。

● タイム・スタンプ取得関数の実装をしてみる

`esp_timer_get_time`はESP-IDFの`components/esp32/esp_timer.c`で定義されています。内部では単に`esp_timer_impl_get_time`を呼び出しています。

```
int64_t IRAM_ATTR esp_timer_get_time()
{
    return (int64_t) esp_timer_impl_get_time();
}
```

`esp_timer_impl_get_time`はESP-IDFの`components/esp32/esp_timer_esp32.c`で定義されています(リスト1)。この関数の内部では一定間隔でカウントし続けるフリー・ランニング・カウンタのカウント値やフラグを読み取っています。このカウンタは、ESP32のペリフェラル・バス (APB) のクロックである80MHzでカウントし続けているので、分解能は $1/80\mu\text{s}$ (12.5ns)です。

`esp_timer_impl_get_time`内で呼び出されている`timer_overflow_happened`関数は、上記カウンタのレジスタを幾つか読んだ上で、条件を満たすかどうかを返しています。

注1: ESP-IDF v4.4.3でビルドして動作することも確認しています。