

今回の概要

今回のロガープログラムに関しては、大変お待たせして 申し訳ありません。

やっと、完成しました。それと、もう一つ

*****。

実は、前回の動画を Upした夜、久々に早く寝ました。そしたら未明に見た夢で トリガ判定は正常に動いていて、予想外のところに原因があるのではないか？ というお告げ？ が、ありました。だいたい 朝起きた時には、寝ている時に見た夢の事は 殆ど忘れていますが、今回は覚えてました。で、その通りでした。

トリガの判定は、割り込み処理の中で 正常に動いてました。そしてトリガ条件が成立し、**取り込んだデータを SDカードに書き込むことが出来ていません**でした。

割り込み処理から、メインループ処理に 書込み指示を伝えるのは、フラグ変数で 行っています。

フラグ変数に 収録開始の フラグが立っている事も確認しました。が、SDカードに 書込む処理が呼び出されませんでした。

原因は **コマンド待ちのシリアル通信の受信処理** でした。コマンドも いつPCから送ってくるか分からないので、待ち続けるのが 通常です。しかしシリアル受信で 待ち続けていたため、データの書き込み処理が 呼び出されずにいた。という事です。運転モードでは 受け付けるコマンドは 停止コマンドだけです。よって、待ち状態を発生させないで 停止コマンドだけを受け付ける 関数を作成しました。これで、うまく行くようになりました。

簡易ロガーのモジュール構成

177の 動画の時より、2本モジュールが増えています。 赤で囲った 下の2つです。

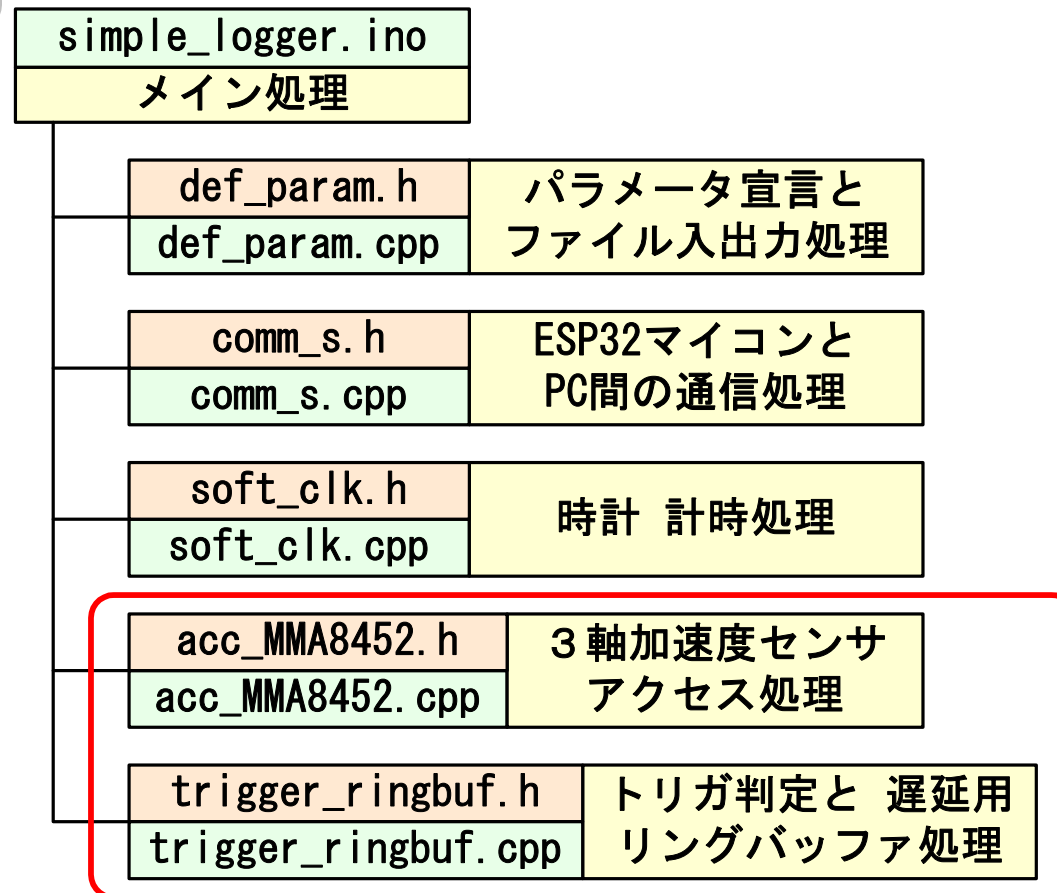
今回、1本 関数を追加したのは `comm_s` モジュール（ESPマイコンとPC間の通信処理）です。

次のページに 待ち状態を発生させないで 停止コマンドだけを受け付ける 関数 を お見せします。 その前に 一応 停止コマンドの説明をしておきます。 停止コマンドは 4byte です。

:	B	Cr	Lf
---	---	----	----

 ASCII文字の ":B" と 制御コードの Cr と Lf です。

ESP32側では このコマンドを受信すると 運転モードから 停止モードに 移行します。



停止コマンド専用の受信関数 ソース

```
//*****  
/** ":B"+Cr+Lf ( 4byte受信確認 ) **  
/** 関数値 : = 0 受信 無し **  
/**           = 1 停止コマンド受信 **  
//*****  
int  comm_s::recv_B_cmd( void )  
{  
    int  sts;  
    byte  a, b;  
  
    sts = 0;  
    if( Serial.available() == 4 )  
    {  
        a = Serial.read();  
        b = Serial.read();  
        Serial.read(); Serial.read(); // 2回空読み  
        if((a == ':' ) && (b == 'B'))    sts = 1;  
    }  
    return  sts;  
}
```

左は 今回作成した `recv_B_cmd()` 関数のソースです。`Serial` が、Arduino環境にて使用するシリアル通信オブジェクトのクラスです。`Serial.read` 関数は 1文字 受信する関数で受信した文字は 関数値として取り出せます。

但し、システムの受信バッファ内に 受信文字が 入っていないと 文字を受信するまで、この `read` 関数内にて 待ち状態になるようです。他に、何もやってない場合だと、受信待ちにしても いいかも しれませんが、**他にやる事がある場合は、障害になります。**

よって `Serial.available` 関数を用いて 何文字受信しているか確認します。停止コマンドは 4byteなので、`Serial.available`の関数値が **4でない場合は、0** を `recv_B_cmd()` 関数は 返します。4byte受信していたら **4byte取り出して、先頭 2文字が ":B"であれば 1** を返します。**1 であれば停止コマンド受信**です。

という事で、停止コマンド受信と、トリガONによる、遅延データの SDカード書き込み処理を2つパラって実行する事が 出来るようになりました。 確認も含め 1日で方が付いたので良かったです。

そのおかげで、3軸加速度データファイルの波形表示ビューアを作る事も 出来ました。

では、実験の動画で 出来あがった物をお見せします。

