

## PC、ESP32間のパラメータのやり取り

今回は、前回の続きで、パソコン、ESP32マイコン間でのパラメータファイルのやり取りを行うプログラム開発の続きです。今回は、すぐプログラム開発に着手したので説明用のドキュメント作成が後回しになっていました。

目標にしていた、ESP32マイコンとパソコン間で双方向のパラメータ転送の実現とESP32側でSDカードにパラメータファイルを書き込み、読み出しを行う事は一応完成しました。

今回は、SDカードにパラメータファイルを書き込んだり、読み出したりする処理は、あっさり出来ましたが、通信処理はパソコン側とマイコン側とで、通信を行わせる訳ですが、パソコン、マイコン両方のプログラムが出来てないと動作確認が出来ないです。

そして、通信プログラムで障害が発生するとパソコン側の問題かマイコン側の問題か判別するのが、難しい場合があります。

今回、データ転送するプログラムを久々開発して、Windows環境でシリアル通信を行う時にパソコン側で送信を行う動作は、作りやすいのですが、パソコン側で受信する時が厄介な場合があります。それは、パソコン側は外部からの事象を受け取る時に、イベントドリブンという方式を採用しています。シリアル通信で受信する場合、タイミング的な問題でイベントドリブンと相性が悪い場合があります。

どういう事かというと、例えばパソコン側で42byteの電文を送信する場合はすんなり出来ます。そして相手側からACKコードのような1byte受ける場合も、1回のイベント呼び出しがありACKコードを受け取れます。

逆に、相手側のマイコンで 42byteの電文を送信してくる場合、パソコン側では **ある程度受信した状態で シリアル通信の受信文字のイベントが発生**します。で、厄介なのは **ある程度受信した状態で** という事で 送られて来た 42byteの電文を **全て受信した状態で イベントが発生する訳ではありません。**

42byte中の 20byteほど受信した状態で、受信イベントが発生します。で、**イベントハンドラで 電文を読み出すと前半の 20byteほどしか取り出せない。**という事です。で、残りの電文後半は イベントが発生しない場合もあります。確実に 後半のデータを 取り出すために 再度イベントが発生すれば、前半と 後半を連結して 1本のデータを 受け取れます。が、**再度のイベントが発生する場合と、発生しない場合があるので、厄介**という事です。

何故、そうなるのかは分かりません。分かっている事は、次々に電文が送られて来る場合は トコロテン式に 時間差をおいて受信イベントが発生して受信した順に 電文が取り出せます。

中途半端に短い電文が残留してる場合 Windows 側で 受信のイベントが発生しない場合がある様です。

で、受信バッファ内の残留しているデータを取り出す方法もあります。**受信イベント内にて 0.1秒の タイマーをスタートして、0.1秒経過で タイマーイベントが呼び出された時に、受信バッファ内に何バイトデータが残っているかを 調べて、その残留データを 取り出す事は出来ます。**

ちょっと面倒ですね。ちなみに私が使用している PCアプリ開発環境は 古い Delphiです。

新しい開発環境は 便利な機能が 多々ついてますが 高いので なかなか買えません。

試行錯誤した後、一応 PCから ESP32マイコン、及び ESP32マイコンから PCの双方向のパラメータのやり取りが出来るようになりました。

それと、今回の通信テストを行う上で、**最初の1回目のパラメータ転送にて時々転送に失敗する事**を確認しました。2回目以降は全く問題なくパラメータ転送出来ます。

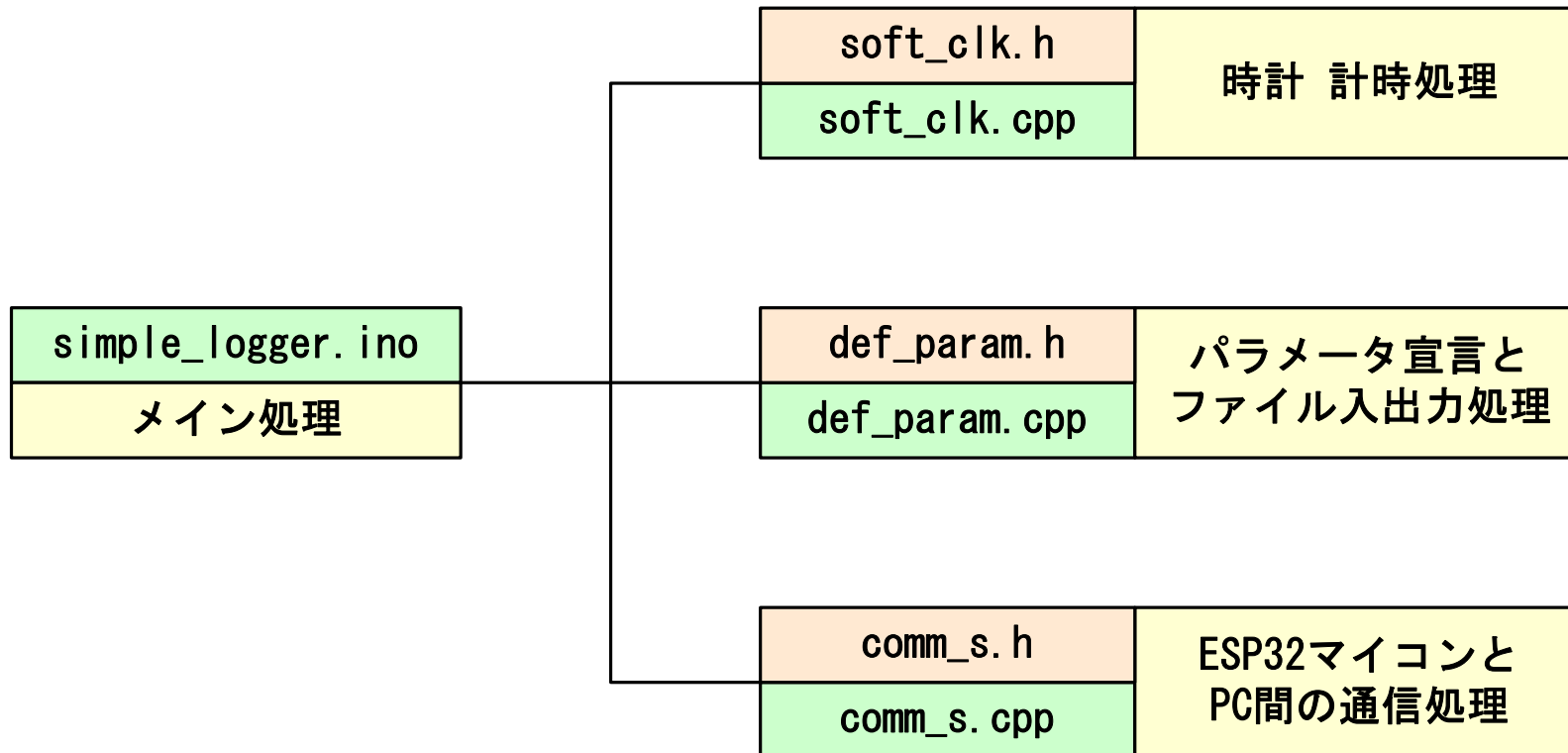
最初、考えにくいな。何が影響しているのだろうと思ってましたが、分かりました。これは、**ESP32マイコン特有の事**ですが、**ESP32マイコン**は ESP32マイコン内に常駐している ESP TOOLが 電源ON時、またはリセット直後に **ステータス情報を USBシリアルポートに出力**します。これを **パソコン側アプリが受信**して 伝送障害を 起こしていたという事です。

よって、PC側アプリにて COMポート オープン直後に 0.2秒ほどの タイマーを起動させて 0.2秒後のタイマーイベント内にて シリアルポートの受信バッファに 残留しているデータの byte 数を確認して 残留データがあれば 空読みして受信バッファを 空にするようにしました。

その後、初回アクセスに失敗する事は、無くなりました。

以上が 今回確認した障害とその対応でした。

## ESP32側 ソフト モジュール構成



```
:S
:H10000067452301CDAB34125469746C6520737437
:H10001072696E672E0000000000000000000000003C
:H100020000000000000000000000000000000000030
:H100030000000000000000000000000000000000020
:H100040000000000000000000436F6D6D656E742023
:H100050737472696E672E000000000000000000007B
:H100060000000000000000000000000000000000070
:H100070000000000000000000000000000000000060
:H100080000000000000000000000000000000000090
:H1000900000000000000000000046400180A0B00F9
:H1000A000001900000003005802000000000300F3
:H1000B058020000000003005802000000000300A0
:H1000C058020000B4000000000000000000000003E
:H1000D0000000000000000000000000000000000C0
:H1000E0000000000000000000000000000000000F0
:H1000F0000000000000000000000000000000000E0
:E
* Normal end.
```

パソコンから、ESP32マイコンに送信する  
パラメータデータ転送電文の ダンプリスト  
です。

:S が パラメータデータ転送の開始コマン  
ドです。

:H が パラメータの部分データ 16byteを  
先頭に":H"と バイト数と先頭から  
の相対番地を付けて 40文字の  
16進文字列にして 16行 転送して  
ます。

16行の電文で 256byteの  
パラメータデータを 転送した事に  
なります。

:E が データ転送終了コマンドです。

```
:s
:h10000067452301CDAB34125469746C6520737437
:h10001072696E672E0000000000000000000000003C
:h100020000000000000000000000000000000000030
:h100030000000000000000000000000000000000020
:h1000400000000000000000000436F6D6D656E742023
:h100050737472696E672E000000000000000000007B
:h100060000000000000000000000000000000000070
:h100070000000000000000000000000000000000060
:h100080000000000000000000000000000000000090
:h10009000000000000000000000046400180A0B00F9
:h1000A0000D0700000003005802000000000300E0
:h1000B058020000000003005802000000000300A0
:h1000C058020000B4000000000000000000000003E
:h1000D0000000000000000000000000000000000C0
:h1000E0000000000000000000000000000000000F0
:h1000F0000000000000000000000000000000000E0
```

:e

\* Normal ned.

ESP32マイコンから パソコンに 送信する  
パラメータデータ転送電文の ダンプリスト  
です。

:s が パラメータデータ転送の開始コマン  
ドです。

:h が パラメータの部分データ 16byteを  
先頭に":h"と バイト数と先頭からの  
相対番地を付けて 40文字の16進  
文字列にして 16行 転送してます。  
16行の電文で 256byteの  
パラメータデータを 転送した事に  
なります。

:e が データ転送終了コマンドです。