

RL78/G22-FPB(048)で遊んでみた(その 3)

今回は、RL78/G22-FPB に Multi function シールドを接続することにします。しかし、7SEGLEED の場合のように、使用するピンを変更する対応だけでは不十分です。ハードウェアがそのままでは、絶対に Multi function シールドを動かすことはできません。

Multi function シールドは Arduino コネクタの 8pin を MOSI 信号として使用していますが、RL78/G22-FPB はそのままでは Arduino コネクタの 8pin から信号を出力することはできません(RL78/G22-FPB には、そのためのショートパッドがあるのですが、**絶対**、それを**ショートしてはいけません**)。

対策として、RL78/G22-FPB にジャンパー線を追加します。写真が分かりにくいですが、図 1 に示すように MCU コネクタ J2 にある P51 の信号を Arduino コネクタの 8pin と接続します。

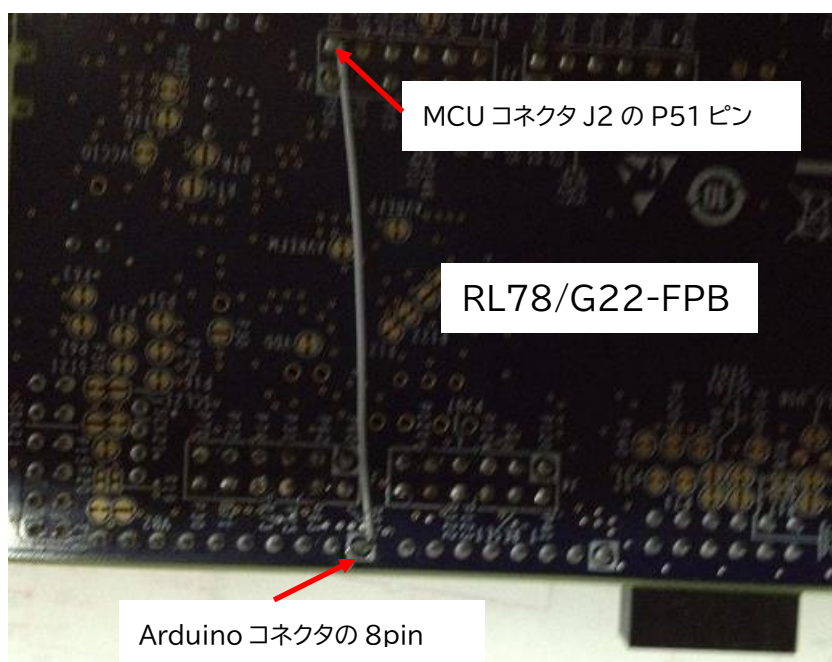


図 1. 追加するジャンパー線

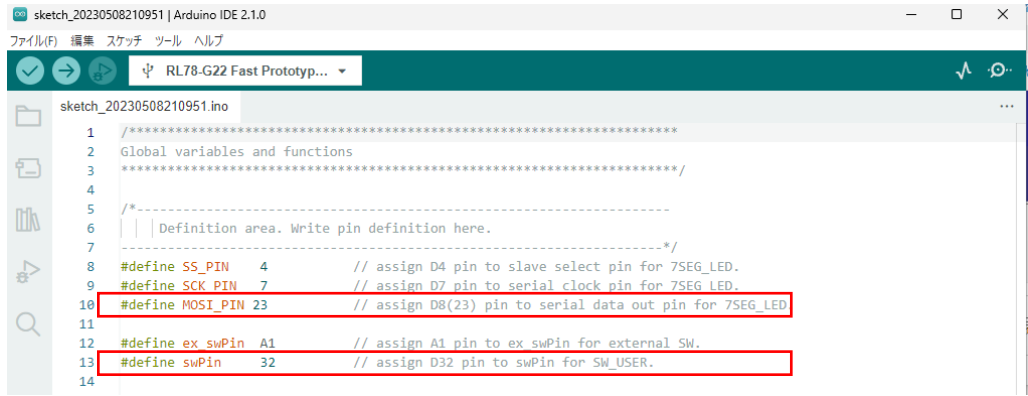
ここでは、今後の拡張に備えて、J1～J4 の MCU コネクタに 12pin のメスコネクタを半田付けしており、そのコネクタの P51 用のコネクタの足を利用してジャンパー配線しています。

P51 は Pmod2 の 7 ピンの INT 信号として準備されているだけなので、使う可能性が低いと考えて、この信号を選択しました。

スケッチは「RL78G23-64PPFB で遊ぶ(2)」で使用したものを流用します。当然ながら、そのままでは使えないので、ArduinoIDE の V2.0.0 以降を使います。そのうえで、使用するボードを RL78-G22Fast Prototype Board に変更します。

図 2.に示すように、赤い四角で囲んだオンボードの信号 pin を変更します。13 行目の swPin はオンボードの SW が接続されている P137(Arduino コネクタの 32pin)に変更するだけです。

問題は、MOSI_PIN です。物理的には 8pin なのですが、ジャンパーで P51 に接続してあるので、図 3. で示してあるように 23 とします。



```
1  /*****
2  Global variables and functions
3  *****/
4
5  /*-----
6  | | Definition area. Write pin definition here.
7  | |-----*/
8  #define SS_PIN 4 // assign D4 pin to slave select pin for 7SEG_LED.
9  #define SCK_PIN 7 // assign D7 pin to serial clock pin for 7SEG_LED.
10 #define MOSI_PIN 23 // assign D8(23) pin to serial data out pin for 7SEG_LED
11
12 #define ex_swPin A1 // assign A1 pin to ex_swPin for external SW.
13 #define swPin 32 // assign D32 pin to swPin for SW_USER.
14
```

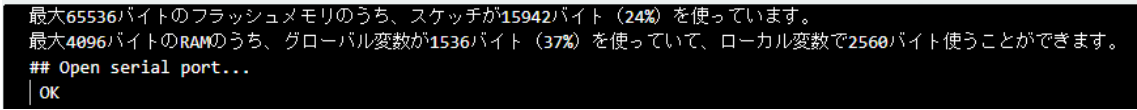
図 2. スケッチの変更点

表 5-6: MCU ヘッダ ピンアサイン(J2)

回路図 部品番号	Arduino® 信号名 ¹⁾	RL78/G22			その他の機能
		ピン	電源	ポート/周辺機能	
J2-1	22	13	-	P50/TS00/INTP1/SI11/SDA11	-
J2-2	23	14	-	P51/TS28/INTP2/SO11	-
J2-3	~9	15	-	P17/TI02/TO02/TS18/(TxD0)	-

図 3. 使用する P51

この変更を行ったスケッチは名前を変更して保存しておきます。これをコンパイルすると、図 4. に示すような結果が得られます。



```
最大65536バイトのフラッシュメモリのうち、スケッチが15942バイト（24%）を使っています。
最大4096バイトのRAMのうち、グローバル変数が1536バイト（37%）を使っていて、ローカル変数で2560バイト使うことができます。
## Open serial port...
| OK
```

図 4. コンパイル結果

これで、RL78/G22-FPB を USB で接続して、書き込みを行います。

書き込みが完了すると、図 5.に示すように表示は 000.0 で停止しています。

そこで、Multi function シールドの左端のスイッチを押すと、カウントを開始します。

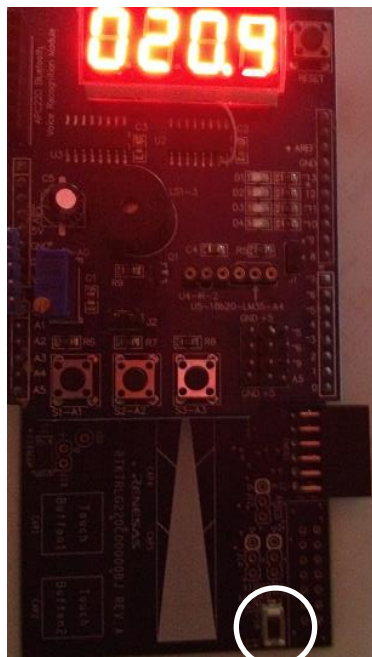


図 6. 動作中表示例

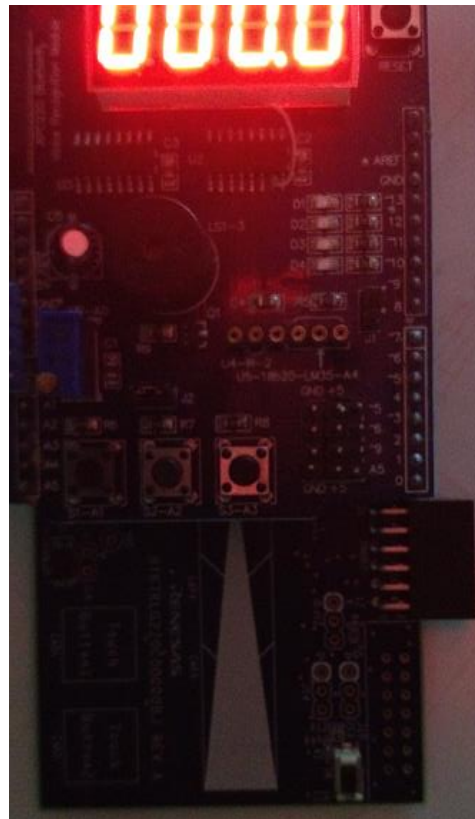


図 5. 起動時の表示

後は、Multi function シールドの左端のスイッチか、上の白い○で囲んだ RL78/G22-FPB の SW を押すたびにカウントしたり、停止したりを繰り返します。

このように、Arduino IDE ベースのセットでは、簡単にスケッチを作成したり、移植したりすることが可能なのですが、ちょっとしたスケッチでもサイズが 10K バイトを超えています。それでも、RL78/G22-FPB のように 64K バイトのコードフラッシュがあれば、問題はないですが。もっとも、メモリが気になるような場合には、CS+のベアメタル環境にすることになります。

(個人的には、そちらの方が好きですが。)

以上