

急遽、タイトル変更

今回は、当初、「062 RX220 マイコンにて I2Cの通信機能を ソフトで実現」を、タイトルにする予定でしたが、途中から、不可解なトラブルに見舞われて、解決策が 思い付かない状態で日付が過ぎて行き、挫折しそうになってました。

最初の きっかけは、RX220マイコンと SPIデバイス基板との間の、連結ケーブルに I2Cのケーブルを追加した事でした。今まで動いていた SPIのデバイスが、動くものと、動かないものが出てきました。具体的には、SRAM 23LC512 が、Write Read コンペアテストで、全アドレスで エラーになったり、D/A コンバータ MCP4922にてアナログ信号が、全く出ない現象が出ました。ところが、I/Oエクspander MCP23S17 は、動くのです。何故 ???

ケーブルの半田付け不良とか、有るのだろうかとかコネクタ両端での、導通テスト、隣同志のピンの ショートの確認等行いましたが、不良個所はありませんでした。

SRAMと DAC が ダメで I/Oエクspanderが動くという事は、SS スレーブセレクト信号に問題があると思い、3つの デバイスの SS信号をオシロにて観測しました。すると、SRAMと DACの SS信号が、全く出ていませんでした。I/Oエクspanderの SS信号は、出ています。

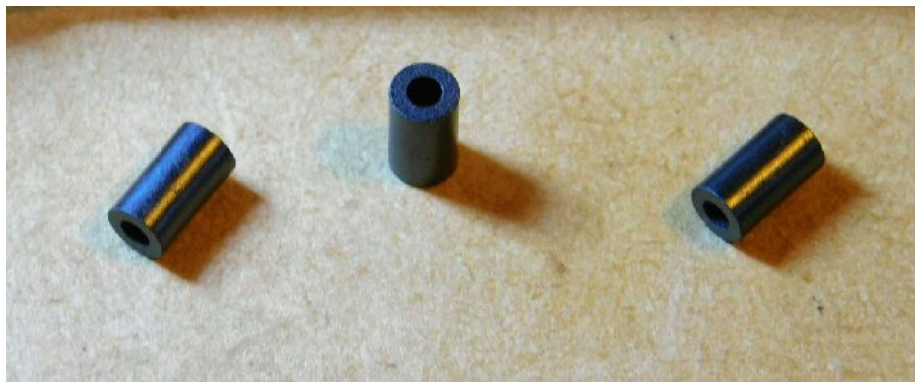
では、何が違うかというと SRAMと DACの SS信号は、ポートAから出ており、I/Oエクspanderの SS信号は、ポートCから出ているのです。ポートAが、途中から機能しなくなっている。

ソフトは、全く触っていません。何故、？
非常に、気味の悪い現象で 暗礁に乗り上げた感じでした。

で、冷静に考えてみると、連結ケーブルに
I2Cのケーブルを 追加接続した後から という
事で、現象からすると考えにくいのですが SPI
の クロックや データの信号が、SSの信号に
ノイズとして、多少混入しては いました。

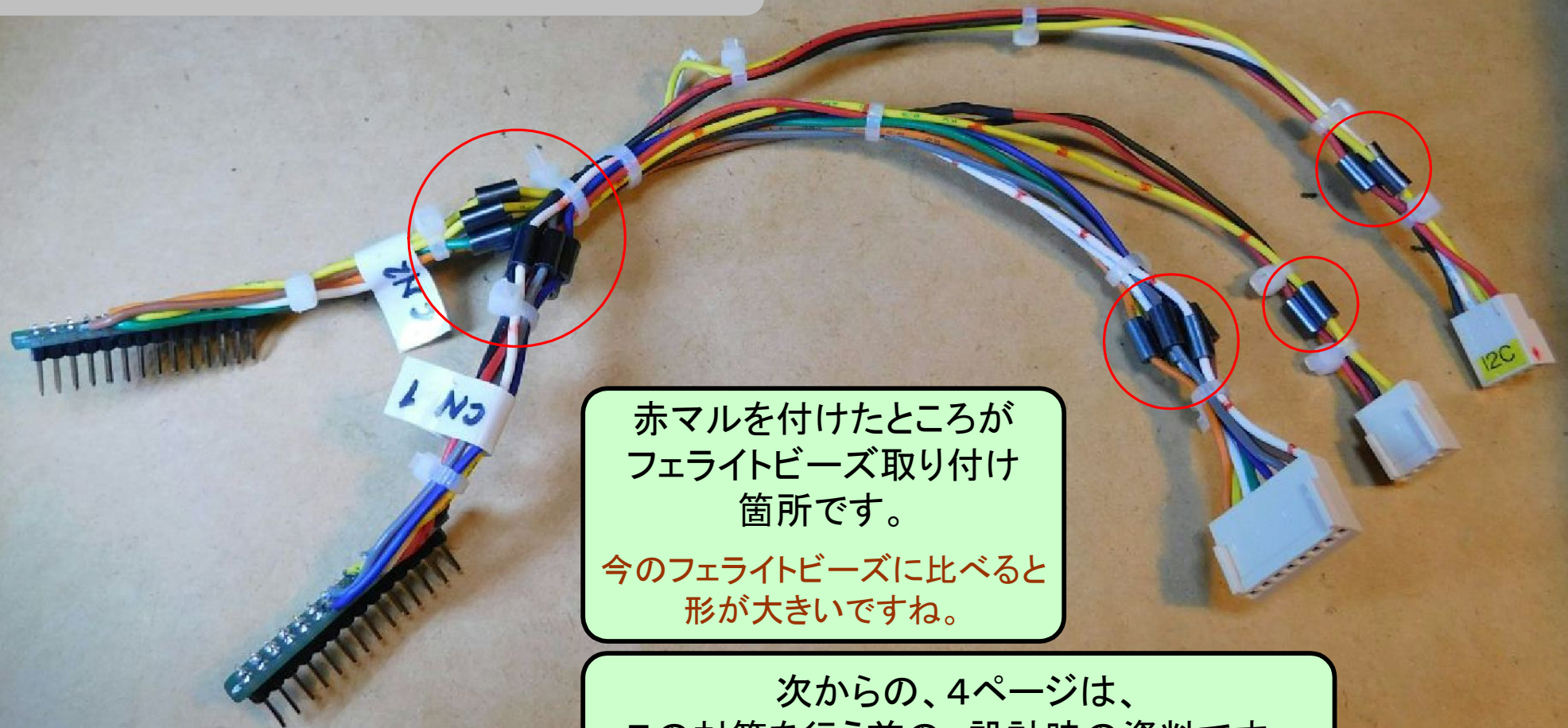
面倒ですが、他に手段が思いつかないので
連結ケーブルを 一旦分解して、信号線の両端
に **フェライトビーズを 嵌め込む事**にしました。

今回、使用したフェライトビーズの画像を
貼り付けておきます。 この処置を行った事で
何と 正常に動き出しました。 良かったです。



ちなみに、このフェライトビーズは 10年ぐらい前に、
秋月電子で 購入したものです。 現在は、このタイプの
物は 売られてないです。

フェライトビーズを付けた連結ケーブル



赤マルを付けたところが
フェライトビーズ取り付け
箇所です。

今のフェライトビーズに比べると
形が大きいですね。

次からの、4ページは、
この対策を行う前の 設計時の資料です。

RX220マイコン ソフト制御で I2C実現

RX220には、周辺回路で **RSPI** は **SPI** を実現し、**RIIC** は **I2C** を実現する回路が実装されています。但し、秋月電子の RX220基板は 64ピンパッケージの RX220 CPUチップを使用しており、100 ピンパッケージと比べると、ピンが 少ない分、ピンアサインの自由度が、やや制限されています。このピンアサインの問題で 64ピンの RX220 CPU は、**RSPI** と **RIIC** を 同時に使用しようとすると、一部のピンの機能が重なってしまいます。具体的には、I/O ポート **P17** 端子が **RSPI/MISOA** と **RIIC/SDA** で 重なります。

よって、ソフトで I2Cの SCL と SDA を I/O ポートの空いている任意のポートに割り当て、I/Oポートとしてソフトウェアで、アクセスする事により、I2C の機能を実現します。

ソフトで、クロック同期のシリアル通信を行うのは、遅くなる。 というイメージが、あります。

SPIの場合は、**RSPI**周辺回路で、RX220では 20MHz水晶で、最大 5Mbps、32MHzの内部発振器では、最大 8Mbpsの設定が出来ます。

このような、数Mbpsのビットレートを、ソフトで実現するのは無理です。しかし、**RIIC**の場合 **I2C**の標準速度である、400Kbps が 最高速度になっているので、ソフトでやっても同等の速度が出せると思います。で、**I2C**の **SCL** と **SDA** を 任意のI/Oポートに割り当てられるはずなのですが、一つ条件があります。オープンドレインの出力ポートに出来る必要があります。特に SDA は、入出力の方向を反転させるため、切り替える瞬間 マイコンと デバイスの出力同士を 喧嘩させないために、オープンドレインの 必要性があります。

RX220の I/Oポートで、オープンドレインに設定できるポートと、出来ないポートがあるようです。オープンドレインに出来るポートは、P1、P2、P3、PA、PB、PC、PEの 7個です。

I2Cで ポートを使用する場合、1つのポートで2ビット だけのポートが望ましいです。

そのポートを I2C だけで占有したいのです。

I2Cで使用するポートに、他の用途のビットが混じっていると、ビット単位の I/O命令を使わないとアクセス出来ません。ビット単位で、I/O命令を行うという事は 遅くなります。過去の 024の動画に 遅くなる理由を説明しています。

という事で、オープンドレイン設定可能な I/Oポートで、2ビットだけのポートという事で 当初 P2 を選択しました。 使用可能なビットは、P26 と P27 の 2つです。

ところが、出力ポートとして P26 と P27 を設定して、方形波を出してみると、P27 は、出ますが、P26 が 出ません。？

他の機能として P26 が、使われて無いか確認したところ、プログラム書き込みに 使用するシリアル通信の TxD1 に 割り当ててあったのです。TxD1 は 外す事が出来ないので、P2 を使う事は諦めます。

オープンドレインのポートで、ピンが 2個空いているポートがないかを見たところ、P3 が 使えそうです。P3 は、I/Oピンが、4ビット有りますが、P30 が、またも通信の RxD1 で 使用されており、P35は NMIとして使用する端子でもあり、入力端子としてしか使えません。

P3の 残りは、P31 と P32 で、どこにも使用してないようです。という事で、P31 と P32 を 使い I2Cを 実現しようと思います。

I/Oポートレジスタ表 改修 4版

2022年 9月 21日 改修

P0	b7	
	b6	
	b5	CN1/19 : RSPI
	b4	
	b3	CN1/18 : RSPI
	b2	
	b1	
	b0	

P3 OD	b7	
	b6	
	b5	CN2/ 28 : NMI
	b4	
	b3	
	b2	CN2/27 : I2C
	b1	CN2/26 : I2C
	b0	CN2/25 : RxD1

PA OD	b7	
	b6	CN1/17
	b5	
	b4	CN1/16
	b3	CN1/15
	b2	
	b1	CN1/14:SPI
	b0	CN1/13:SPI

PE OD	b7	
	b6	
	b5	CN1/12
	b4	CN1/11
	b3	CN1/10
	b2	CN1/ 9
	b1	CN1/ 8
	b0	CN1/ 7

P1 OD	b7	CN2/22 : RSPI
	b6	CN2/21
	b5	CN2/20
	b4	CN2/19
	b3	
	b2	
	b1	
	b0	

P4	b7	
	b6	CN1/ 6
	b5	
	b4	CN1/ 5
	b3	CN1/ 4 : ADC
	b2	CN1/ 3 : ADC
	b1	CN1/ 2 : ADC
	b0	CN1/ 1 : ADC

PB OD	b7	CN2/ 6
	b6	CN2/ 5
	b5	CN2/ 4
	b4	
	b3	CN2/ 3
	b2	
	b1	CN2/ 2
	b0	CN2/ 1

PH	b7	
	b6	
	b5	
	b4	
	b3	CN2/18
	b2	CN2/17
	b1	CN2/16:ベース基板/LED緑
	b0	CN2/15:ベース基板/LED赤

P2 OD	b7	CN2/24
	b6	CN2/23 : TxD1
	b5	
	b4	
	b3	
	b2	
	b1	
	b0	

P5	b7	
	b6	
	b5	CN2/14
	b4	CN2/13
	b3	
	b2	
	b1	
	b0	

PC OD	b7	使用不可
	b6	CN2/11:RSPI
	b5	CN2/10:RSPI
	b4	CN2/ 9:RSPI
	b3	CN2/ 8
	b2	CN2/ 7:RSPI
	b1	
	b0	

ポート番号 P0 ~ PH 下に OD の文字が、付いたポートがありますがオープンドレインの設定が可能なポートです。

SPI & I2C 接続 基板間ケーブルの コネクタ配線

RX220側

CN1	SSLA1	PA0	13
	SSLA2	PA2	14
	IO_RESET	P03	18
	LDAC	P05	19
	Vcc		21、22
	GND		27、28

CN2	SSLA3	PC2	7
	SSLA0	PC4	9
	RSPCKA	PC5	10
	MOSIA	PC6	11
	MISOA	P17	22
	I2C/SCL	P31	26
	I2C/SDA	P32	27

デバイス基板側

SPI/CN1	1	RSPCKA
	2	MOSIA
	3	MISOA
	4	SSLA0
	5	SSLA1
	6	SSLA2
	7	SSLA3
	8	LDAC

SPI/CN2	1	Vcc
	2	GND
	3	IO_RESET
	4	予備

I2C	1	Vcc
	2	SCL
	3	SDA
	4	GND

