

まずは、I2Cと SPIの 特長

今回は、SPI(シリアルペリフェラルインタフェース)を取り上げます。SPIは、I2Cと同様に いろいろなデバイスを接続するための、同期式シリアル通信規格です。この動画では I2Cによるデバイス接続は 何度か扱って来ましたが、SPIは 今回が、初めてです。同じような用途のインタフェースでは、ありませんが、I2Cと SPIは 異なる特徴があります。今回は、そのあたりから説明を行います。

まずは、I2Cは 2線式のインタフェースですが、SPIは 3線式のインタフェースです。

I2Cは SCL:シリアルクロック信号、SDA: 双方向データ信号の 2本の信号線で構成されます。標準的な転送速度は 400Kbpsです。デバイスによっては、更に早い物(1Mbps)もあります。また、Read/Writeコマンドに 7bitのアドレスも付くので、2線に アドレスの異なる複数のデバイスを接続する事が出来ます。

それに対し、SPI は 3線式で、SCK:シリアルクロック信号、SDO: マスタのデータ出力信号、SDI: マスタのデータ入力信号の 3本の信号線が基本です。

基本です。というのは、SPI は、1台のデバイスを接続する場合は、3本の信号線とグランド線で 用が足りませんが、複数台のデバイスを接続する場合は、各デバイス毎に SS信号(スレーブセレクト信号)が、必要になります。SPIは、転送速度等は、あまり明確に書いてありません。初期の頃の、SPIデバイスは 2Mbpsぐらいの物が、多かったと思います。最近は、非常に早いデバイスでは 20Mbpsの物もあるようです。ROHMのサイトを、参照しました。RXマイコン600シリーズでは 最大 10Mbpsで、RX220マイコンでは、PCLK=32MHzで 8Mbps の 設定が、最大速度のようです。転送速度は I2Cに比べ SPIは 圧倒的に早いですが どうしてだろうと、思われる方も おられるでしょう。まず、2線式と、3線式の違い、それと I2Cは、SCKと SDAの位相差で Start、Stop Condition検出している関係と思います。

I2Cと SPIの 長所と 短所

I 2 C	
長所	① 信号線2本+GNDでデバイスと通信可能。 ② 複数デバイスを接続する事が可能。 複数デバイスを接続しても、信号線は2本のままで、OK。 但し 各デバイスのアドレスは、重複させてはならない。
短所	① SPI と比べると データ転送速度が遅い。

という事で、データ転送速度が あまり問題にならなければ、I2Cが、気がるに使いそうです。

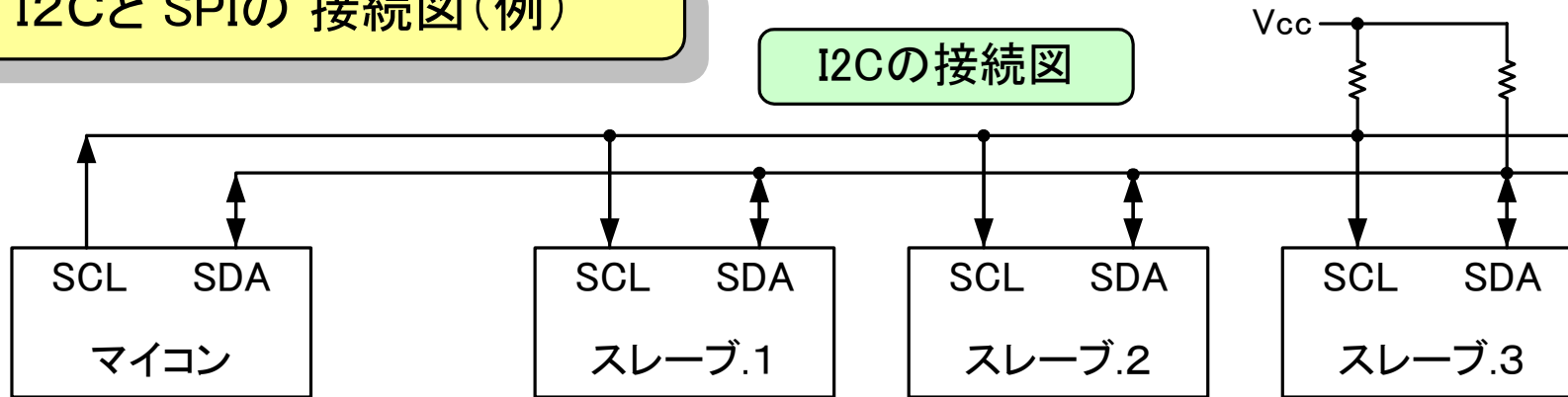
それに対し、SPIは、高速にデータ転送したい用途に使う事になります。 SPIに 複数デバイスを接続する時は、その**デバイスの個数分、SS信号を用意する必要がありますので、足ピン数の少ないマイコンでは、厳しい状況になるでしょう。**

S P I	
長所	① I2Cに比べて、データ転送速度が速い。 ② やろうと思えば、全二重通信も出来る。 ③ デバイス1台接続時は、信号線 3本+GNDで デバイスと 通信出来る。 その際、デバイスの SS信号は、GNDに接続する。
短所	① 複数デバイスを接続する事は、可能であるが、別途 SS信号を 各デバイスに独立して配線する必要が生じる。 デバイスを複数接続時、信号線の本数が増えるのが難点。

余談ですが、I2Cのデバイスも SPIのデバイスも高速にアクセスしなければならない。 というものではないです。 **デバイスは、CPUが出力するシリアルクロックに合わせて動きます。** よって ソフトで、I/Oポートを ビット単位で ON、OFFして アクセスする場合も、当然あります。 問題なくアクセス出来ます。

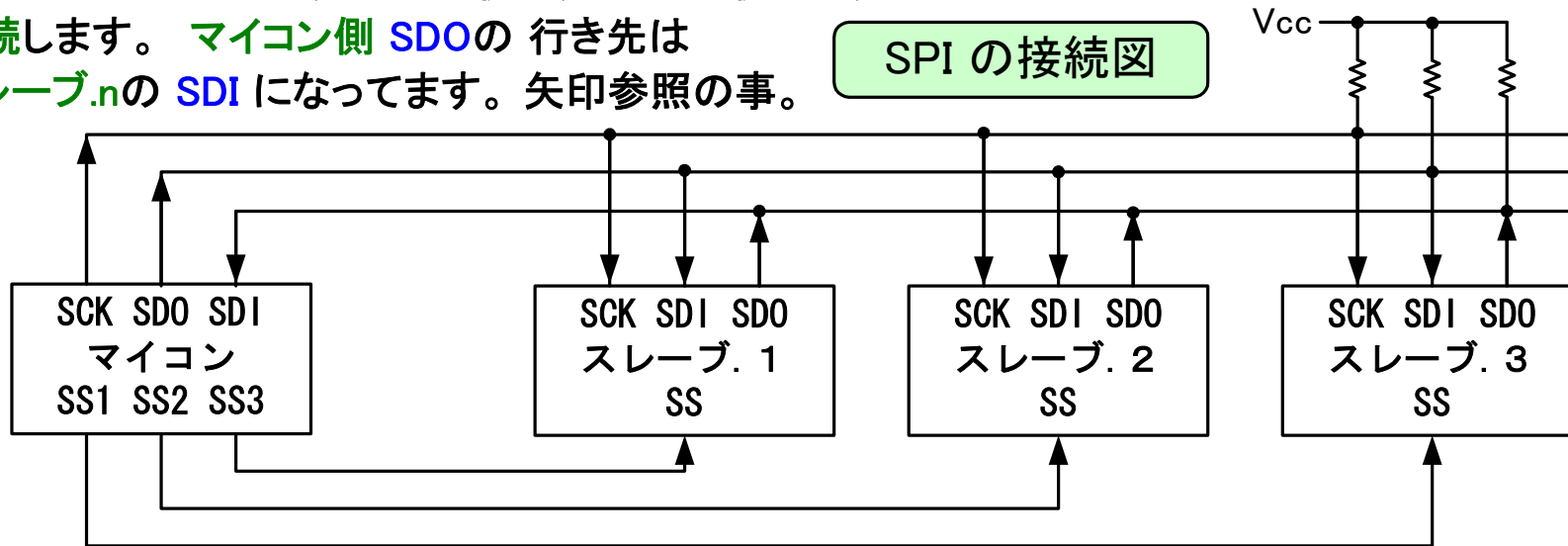
I2Cと SPIの 接続図(例)

I2Cの接続図



SPIの SDOと SDI は、マイコン側と、スレーブ側とで、**クロスした形で**接続します。マイコン側 SDOの 行き先はスレーブ.nの SDI になってます。矢印参照の事。

SPI の接続図



SPIの SDOと SDI は、信号の方向が紛らわしいので、最近では **MOSI、MISO** と表現します。**MOSI** は **マスタ Out、スレーブ In** です。**MISO** は **マスタ In、スレーブ Out** です。

RX220から、SPI信号の引き出し

前ページの説明で、SPI信号の引き回しは、理解して頂けたでしょうか？

RX220には、RSPIという SPI機能を有する周辺回路が、1つ実装されてます。RXシリーズ上位CPUには2チャンネル実装されてます。ちなみに、SS信号は、4本まで引き出せます。つまり、最大 4つのデバイスを接続可能な SPI という事になります。

前回の A/Dコンバータの時は、使用する信号名 AN000～AN003が、秋月電子CPU基板の CN1コネクタのピンアサイン表の 1～4 ピンに割り付けられていたので、引き出す線が すぐ分かりました。

これは、AN000～AN003は アナログ入力で CPUの信号としては特別で、固定的にピンを割り付けられていたものと思います。

RSPIの 入出力される信号名は、RSPCKA、MOSIA、MISOA、SSLA0、SSLA1、SSLA2、SSLA3 計 7本です。

ところが、RSPIの 入出力される信号名は、秋月電子CPU基板の CN1、CN2コネクタ端子一覧表には、出ていません。これは、どういう事かということ RSPI周辺回路と 入出力ポートとの間に、マルチファンクションピンコントローラ(MPC)なる回路が入っており、RSPIの入出力信号を どのポート端子に割り当てるかを設定する仕様になっているのです。

ちょっと面倒な機能で、初心者の方が引かかる可能性もあるので、細かく書いておきます。

ルネサス RX220グループ ユーザーズマニュアル ハードウェア編には、380ページから 7ページに渡り「マルチプル端子の割り当て端子一覧」というのがあります。その端子一覧の 6/7 に RSPI0 の欄があります。そこに RSPIの RSPCKA、MOSIA、MISOA、SSLA0、SSLA1、SSLA2、SSLA3 に 割り当て可能なポート名の候補が、記述されてますので その中から、使用するポートを1つ選択する必要があります。使用するポートは、他の機能と重ならないように注意する必要があります。

マルチプル端子の割り当て端子一覧

パッケージの 項目は削除しました。

100ピン、48ピン

パッケージの 項目は削除しました。				64ピン	秋月基板CN
シリアル ペリフェラル インタフェース	RSPI0	RSPCKA (入出力)	PA5	x	
			PB0	○	C N 2
			PC5	○	C N 2
		MOSIA (入出力)	P16	○	C N 2
			PA6	○	C N 1
			PC6	○	C N 2
		MISOA (入出力)	P17	○	C N 2
			PA7	x	
			PC7	○	C N 2
		SSLA0 (入出力)	PA4	○	C N 1
			PC4	○	C N 2
		SSLA1 (出力)	PA0	○	C N 1
			PC0	x	
		SSLA2 (出力)	PA1	○	C N 1
			PC1	x	
		SSLA3 (出力)	PA2	x	
			PC2	○	C N 2

左のマルチプル端子の割り当て端子一覧から、使用する端子を選択しました。

薄い黄緑色の端子と、薄赤い端子が、選択した端子です。出来れば、使用する端子は秋月電子RX220CPU基板コネクタのCN1側か、CN2側のどちらか片側だけにしたいのですが、ポート端子選択子が、1個しか無い信号が、CN1とCN2に分かれているのでSSLA1とSSLA2のポート端子をCN1側から取りました。よって、選択した端子の割り当ては、以下の通りに決めました。

- ① RSPCKA (SCK) = PC5 (CN2/10)
- ② MOSIA (SD0) = PC6 (CN2/11)
- ③ MISOA (SD1) = P17 (CN2/22)
- ④ SSLA0 (SS0) = PC4 (CN2/ 9)
- ⑤ SSLA1 (SS1) = PA0 (CN1/13)
- ⑥ SSLA2 (SS2) = PA1 (CN1/14)
- ⑦ SSLA3 (SS3) = PC2 (CN2/ 7)

まずは、ハードの配線に関わる事を先に行います。

SPIを使うための MPCの初期設定

前ページの マルチプル端子の割り当て端子一覧を参照して使用する端子を選択しました。

- ① RSPCKA (SCK) = PC5 (CN2/10)
- ② MOSIA (SD0) = PC6 (CN2/11)
- ③ MISOA (SD1) = P17 (CN2/22)
- ④ SS1A0 (SS0) = PC4 (CN2/ 9)
- ⑤ SS1A1 (SS1) = PA0 (CN1/13)
- ⑥ SS1A2 (SS2) = PA1 (CN1/14)
- ⑦ SS1A3 (SS3) = PC2 (CN2/ 7)

この SPIの信号線と 端子を 割り付ける初期設定は、RX220データシートの 494ページ以降を参照しレジスタの設定を行います。

PWPR: 書き込みプロテクトレジスタ

直接的に 必要な物では無いですが、
不用意に書き込まないように するための
カギのような物です。

設定の必要なレジスタのみ、記載します。

P17PFS: P17端子機能設定レジスタ (MISOA用)

コーディング: MPC.P17PFS.BIT.PSEL = 0x0D;

PA0PFS: PA0端子機能設定レジスタ (SS1A1用)

コーディング: MPC.PA0PFS.BIT.PSEL = 0x0D;

PA1PFS: PA1端子機能設定レジスタ (SS1A2用)

コーディング: MPC.PA1PFS.BIT.PSEL = 0x0D;

PC2PFS: PC2端子機能設定レジスタ (SS1A3用)

コーディング: MPC.PC2PFS.BIT.PSEL = 0x0D;

PC4PFS: PC4端子機能設定レジスタ (SS1A0用)

コーディング: MPC.PC4PFS.BIT.PSEL = 0x0D;

PC5PFS: PC5端子機能設定レジスタ (RSPCKA用)

コーディング: MPC.PC5PFS.BIT.PSEL = 0x0D;

PC6PFS: PC6端子機能設定レジスタ (MOSIA用)

コーディング: MPC.PC6PFS.BIT.PSEL = 0x0D;

0x0D は、RSPIと接続する事を意味するようです。
詳細は、RX220データシートを参照して下さい。

SPIを使うための MPCの初期化ルーチン

```
void spi_uses_mpc_init( void )
{
    MPC.PWPR.BIT.BOWI = 0;           // BOWI の 0 は、数字のゼロです。
    MPC.PWPR.BIT.PFSWE = 1;          // カギを開ける

    MPC.PC5PFS.BIT.PSEL = 0x0D;      // PC5 = RSPCKA
    MPC.PC6PFS.BIT.PSEL = 0x0D;      // PC6 = MOSIA
    MPC.P17PFS.BIT.PSEL = 0x0D;      // P17 = MIS0A
    MPC.PC4PFS.BIT.PSEL = 0x0D;      // PC4 = SSLA0
    MPC.PA0PFS.BIT.PSEL = 0x0D;      // PA0 = SSLA1
    MPC.PA1PFS.BIT.PSEL = 0x0D;      // PA1 = SSLA2
    MPC.PC2PFS.BIT.PSEL = 0x0D;      // PC2 = SSLA3

    MPC.PWPR.BIT.PFSWE = 0;          // カギを閉じる
    MPC.PWPR.BIT.BOWI = 1;
}
```

多分、これで よかろうと思いますが、まだ動かして無いので（ コンパイルは 通ります ） なんともいえません。

I/Oポートレジスタ表 改修 3rd 版

2022年 8月 17日 改修

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

P3	b7	
	b6	
	b5	CN2/ 28
	b4	
	b3	
	b2	CN2/27
	b1	CN2/26
	b0	CN2/25

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

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

P1	b7	CN2/22
	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
	b2	CN1/ 3
	b1	CN1/ 2
	b0	CN1/ 1

PB	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
	b0	CN2/15

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

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

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

今回 057のコンテンツ作成中に、以前作成したポートレジスタ表にて、2ヶ所使用出来る bitが、抜けている事を発見しました。

追加した箇所は、右側に緑色の○が付いている箇所 P26 と P30 です。

ポート3の b5 は 入力専用ポートになります。
ポートCの b7 は E1で使用する場合がありますので 使用しない事。

通常のポートとして 入出力出来るのは 45 bit あります。