

## RX220 RSPIを使った SPIデバイスアクセス

RX220マイコンの 周辺回路 RSPIを使用して、SPIの デバイスをアクセスするソフトを作成します。

今まで、RX220で、周辺回路 CMT0 を 使い インターバルタイマ機能を作成したり、SCI1を使って 調歩同期のシリアル通信機能を作成したり、S12ADを使い 12bit A/D入力の ソフトを作成してきました。そして今回 RSPI0 を 使って SPI周辺デバイスを アクセスするソフトを作成した訳ですが、今回の RSPI0 が、ソフト開発で、一番手間のかかる開発作業でした。

メーカーのデータシートに 初期化のフローチャートが、載っているのですが、省略されている部分が 結構あり、分かりにくいのです。

MPCや、PORTn. PMR設定や、IEN 割り込みコントローラの設定の確認で、1200ページほどある、データシートのそれぞれの周辺回路の説明を書いてあるページに飛んで確認する必要があります。そして作成した初期化、1byte転送処理は、全く動作しませんでした。で、動くサンプルソースが、ないかと探して 書籍のRX600シリーズの RSPIのソースがあったので入力して、その後 RX600シリーズと RX220は多少周辺回路が異なるので、コンパイルエラーを取り除き、やってみると動かない。で、下手な鉄砲も数打ちゃ当たるで、色々試行錯誤して、やっと動き出しました。今回は、必要 最低限の 機能だけ実現しました。

RSPIの機能は、かなり高機能で、柔軟性があるので SPI以外でも使えそうな気がします。但し、高機能ゆえに理解するのが難しいです。

## SPI信号波形の確認 1

右のオシログラフは、上側の信号が **SCK**  
下側の信号が、**MOSI** マスター出力データで  
す。

データは、1バイト分のデータ転送時の波形  
です。データは、**0xAA** を転送してます。

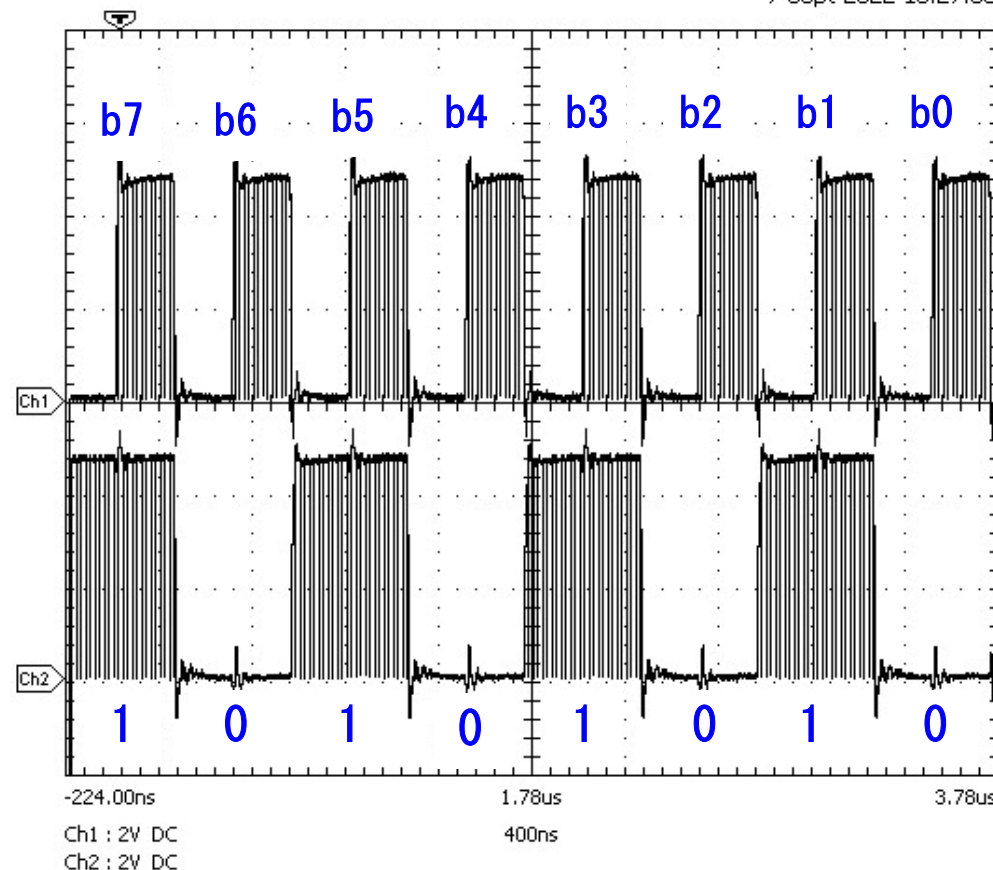
データは、正論理で 左から **MSB b7** です。  
このデータは、マスタが送信していますので  
スレーブ側では、上側の **SCK**の **立ち上がり  
エッジ**で、下側の **MOSI信号の 取り込みを  
行う事**になってます。

そして、**SCK信号の 立ち下がリエッジ**で  
**データの信号を、次の信号に入れ替えて  
います。**

**b0**のデータ信号を出した後は、**しばらくの間、  
b0のデータ信号レベルを 維持**しているよう  
です。

データは、**0xAA**

7-Sept-2022 15:27:33



## SPI信号波形の確認 2

右のオシログラフは、上側の信号が **SCK**  
下側の信号が、**MOSI** マスター出力データで  
す。このオシログラフでは 4バイトのデータを送  
信しています。

データは、**0x00**、**0x03**、**0x0F**、**0x3F**の順に  
送信しています。右のグラフで、色を付けてい  
る四角は、青が 0 の領域、赤が 1 の領域で  
す。

SPI信号の 伝送時の波形のイメージが多少  
なりと掴めていただければ幸いです。

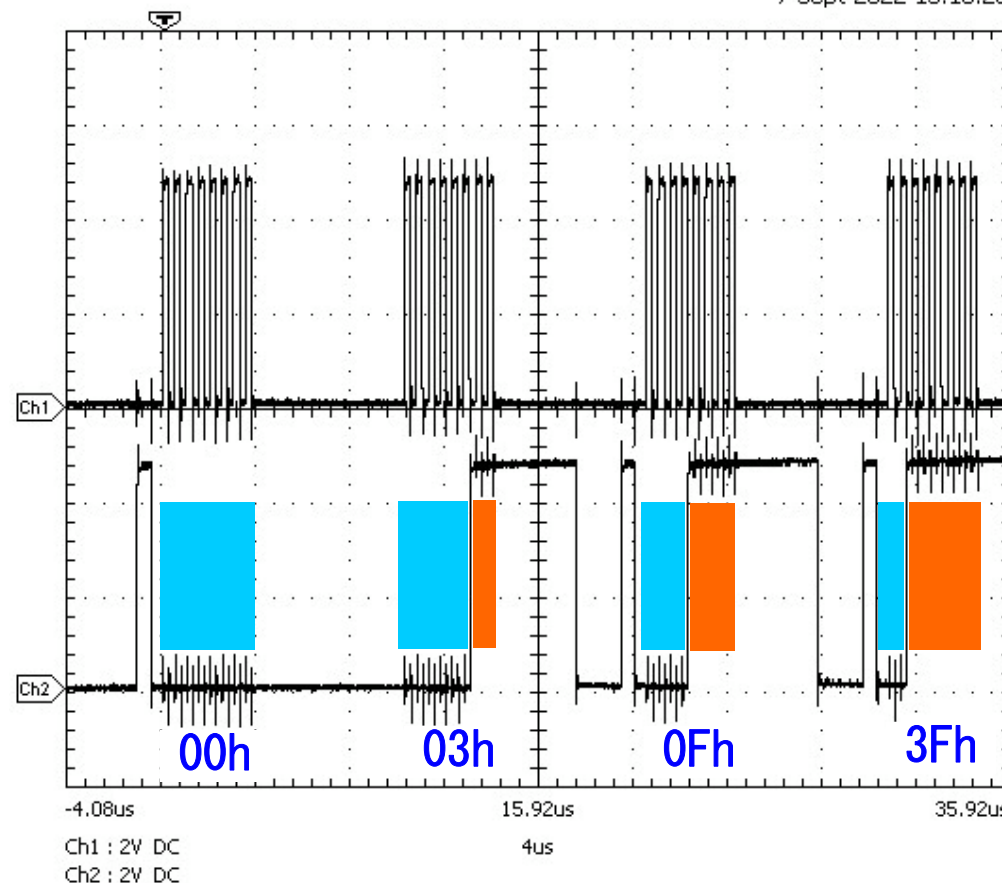
次は、3つのデバイス

- ① I/Oエクspander **MCP23S17**
- ② D/Aコンバータ **MCP4922**
- ③ 512Kbit SPI SRAM **23LC512**

の、コマンド電文の出し方を 説明します。

データ: **00 -> 03 -> 0F -> 3F**

7-Sept-2022 15:18:26



## I/O エクスパンダ MCP23S17

MCP23S17は、内部に 22個のレジスタを持っている 16bitの I/Oエクスパンダです。16bitというよりは、8bitの GPIOAと GPIOBの 2本の I/O ポートレジスタを持っています。

他にも設定のレジスタが、多数あります。マイコンの I/Oポートと同様に 入出力方向を指定する IODIRAと IODIRB があります。まだ理解してない機能が、多々あります。今回は、PortAを 入力、PortBを 出力に設定します。そして PortBに 0x35を出力する電文を示します。尚、このデバイスは 元々 I2Cのデバイスとして作られたようで、説明が I2Cのコマンドイメージで作られています。その関係で デバイスに 3本のアドレスを設定するピンが存在します。そしてその I2Cの アドレス選択機能は、SPIにおいても生きています。

先頭の、コマンドバイトは、I2Cのコマンドそのもので、b7 ~ b1は デバイスアドレスで、b0 が 1=Read／0=Write になっています。

でコマンドバイトは 0,1,0,0, A2,A1,A0 R/W で今回、デバイスアドレスは 0 です。よってコマンドバイトは、Write時は、0x41、Read時は、0x40 になります。

第2バイトは、内部レジスタアドレスです。

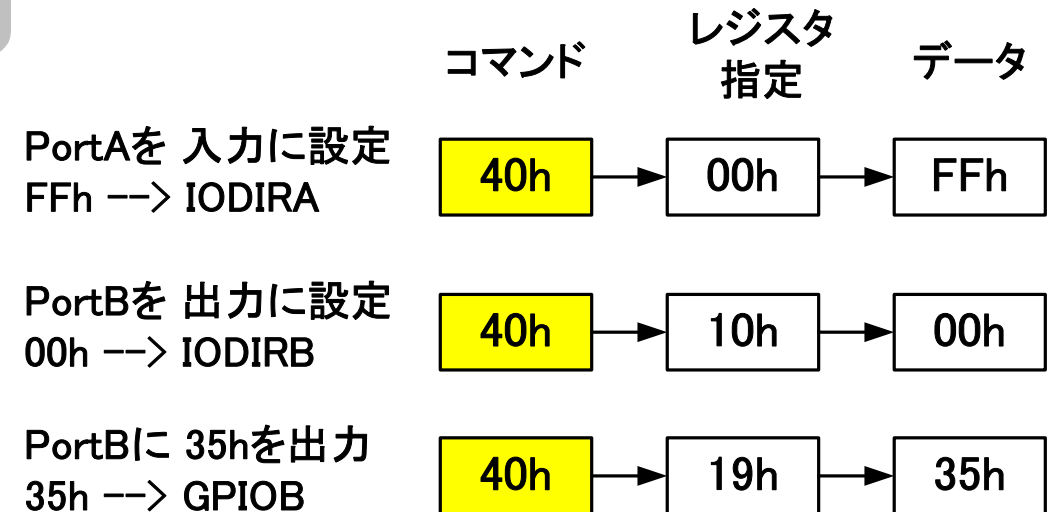
第3バイトは、書き込む、あるいは読み出すデータになります。

言葉で、説明すると分かりにくいので次のページで図で説明します。

## I/O エクスパンダ MCP23S17 電文

MCP23S17 内部レジスタ、アドレス一覧表

| BANK=1<br>アドレス | アクセス先    | BANK=1<br>アドレス | アクセス先    |
|----------------|----------|----------------|----------|
| 00h            | IODIRA   | 10h            | IODIRB   |
| 01h            | IPOLA    | 11h            | IPOLB    |
| 02h            | GPINTENA | 12h            | GPINTENB |
| 03h            | DEFVALA  | 13h            | DEFVALB  |
| 04h            | INTCONA  | 14h            | INTCONB  |
| 05h            | IOCON    | 15h            | IOCON    |
| 06h            | GPPUA    | 16h            | GPPUB    |
| 07h            | INTFA    | 17h            | INTFB    |
| 08h            | INTCAPA  | 18h            | INTCAPB  |
| 09h            | GPIOA    | 19h            | GPIOB    |
| 0Ah            | OLATA    | 1Ah            | OLATB    |



電文は、基本上記のように、3バイトとなります。

今回、原因が分かりませんが、PortB( GPIOB )に接続した 8個の LEDに、任意のデータを表示する事が出来ませんでした。

最初に、このデバイスから やり始めた関係で、RX220の SPI信号の問題か、MCP23S17のコマンドが、間違っているのか分かりませんでした。

切り分けのため、DAコンバータのMCP4922とSRAMのアクセステストを行ってみました。DACとSRAMは、正常にアクセス出来ました。

MCP23S17のコマンドの出し方に問題が、ありそうです。 検討中...

## 12bit D/A コンバータ MCP4922 電文

MCP4922のコマンドは、2Byteしかありません。  
極めて単純です。

先頭バイトの 上位 4bit コマンドで、下位 4bitが  
12bit D/A量子化数の 上位 4bit( b11 ~ b8 )と  
なります、2バイト目に 12bit D/A量子化数の残り  
8bit( b7 ~ b0 )が、並びます。

| 第1バイト |     |     |      |     |     |    |    |
|-------|-----|-----|------|-----|-----|----|----|
| A/B   | BUF | G/A | SHDN | b11 | b10 | b9 | b8 |

| 第2バイト |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|
| b7    | b6 | b5 | b4 | b3 | b2 | b1 | b0 |

第1バイト上位 4bitですが、A/Bは、0=VoutA、  
1=VoutB です。出力チャネルの選択です。

あとは、BUF=0、G/A=1、SHDN=1 で 問題ないよう  
です。

チャネルAを 選択の場合: 0011

チャネルBを 選択の場合: 1011 に、なります。

それと、MCP4922から、アナログデータを 出力する  
場合は、忘れやすいのが、電文を出して、SS 信号  
を Hi に戻した後に、MCP4922の独自信号である、  
LDAC 信号を Hi から Lowに する必要があります。  
LDAC 信号の ダウンエッジで、アナログ信号が  
更新されます。

今回は、スイッチング電源を使用しましたが、12bit  
の D/Aコンバータなので、出来ればノイズの少ない  
電源を使用した方が、綺麗なアナログ出力が  
得られると思います。



## SRAM 23LC512 の通信電文

23LC512に、送る電文は、バイトモードと、ページモードと、シーケンシャルモードの 3つがあります。

今回は、単純な **バイトモード** で通信を行います。  
その場合は、**電文長は、4バイト固定**になります。

まず、**先頭に Read と Write を選択するコマンド**を送ります。23LC512は、64KByte のアドレスレンジがあるので、**アドレス情報を送るのに、2byte必要**になります。その後 **1バイトのデータ転送**を行います。

今回は、バイトモードで、転送する実験を行いました。が、実使用では、**ページモード**や、**シーケンシャルモード**の方が、**転送効率が上がる**と思います。

一つ、**RX220側の処理で、説明を忘れましたが** RSPIにて、転送を始めるまえに、複数スレーブが接続されている場合、SPIでは、選択するデバイスのSS信号をアクティブ Low にする必要があります。転送処理が、終了したら、SS信号を Hi に戻します。

今回は、SS信号出力は 通常の I/Oポートとしてアクセスしています。

