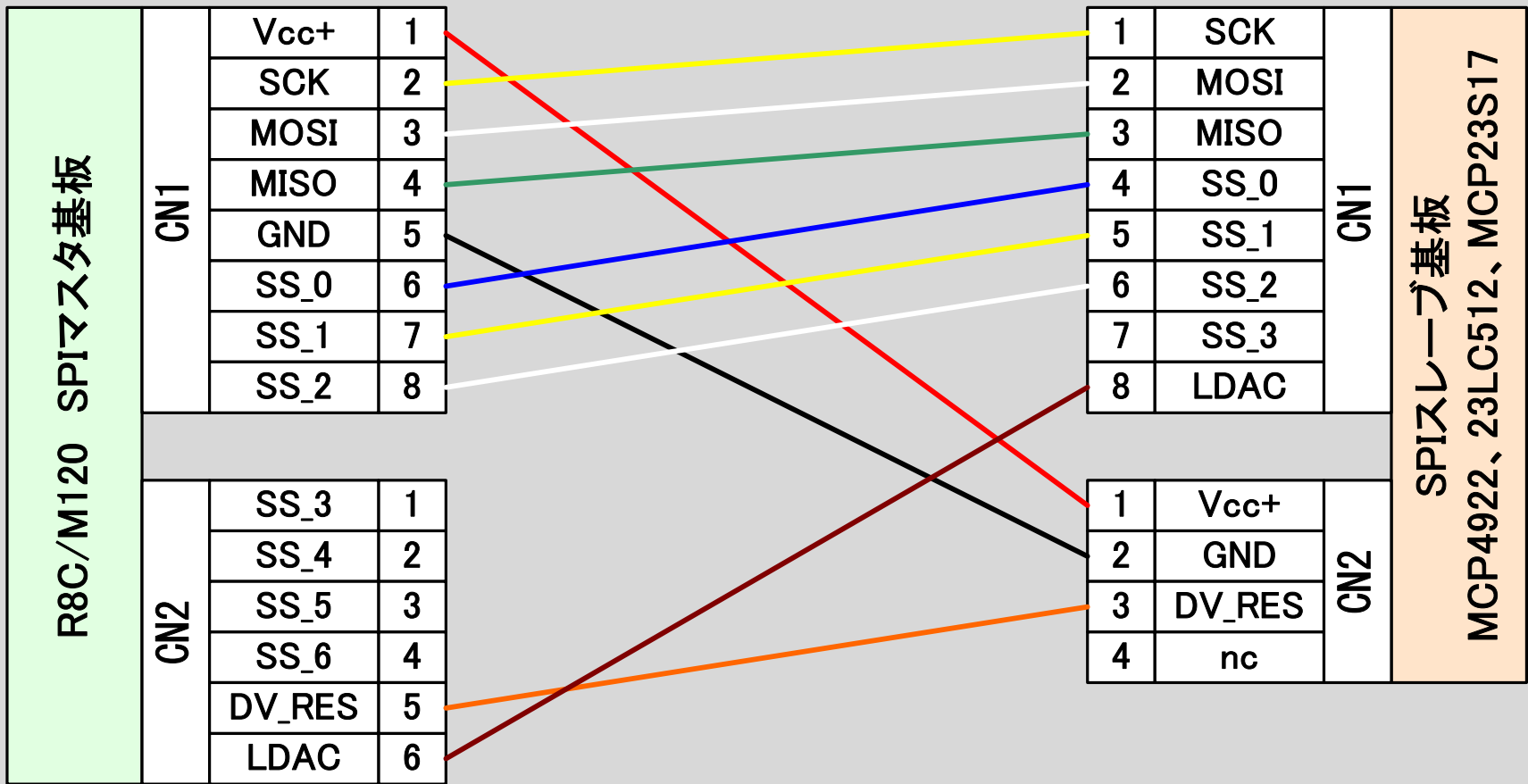
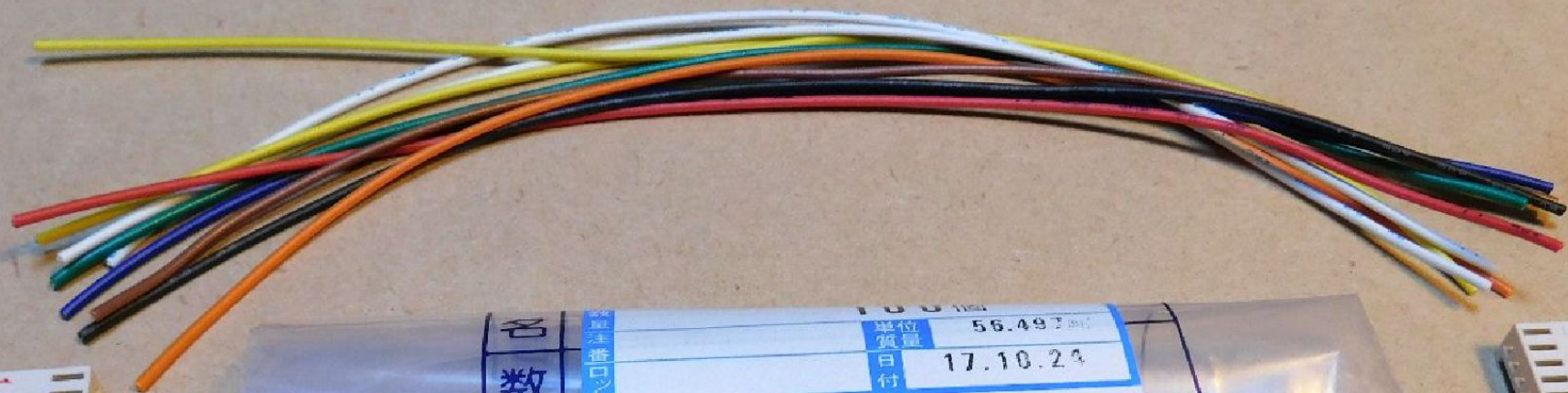


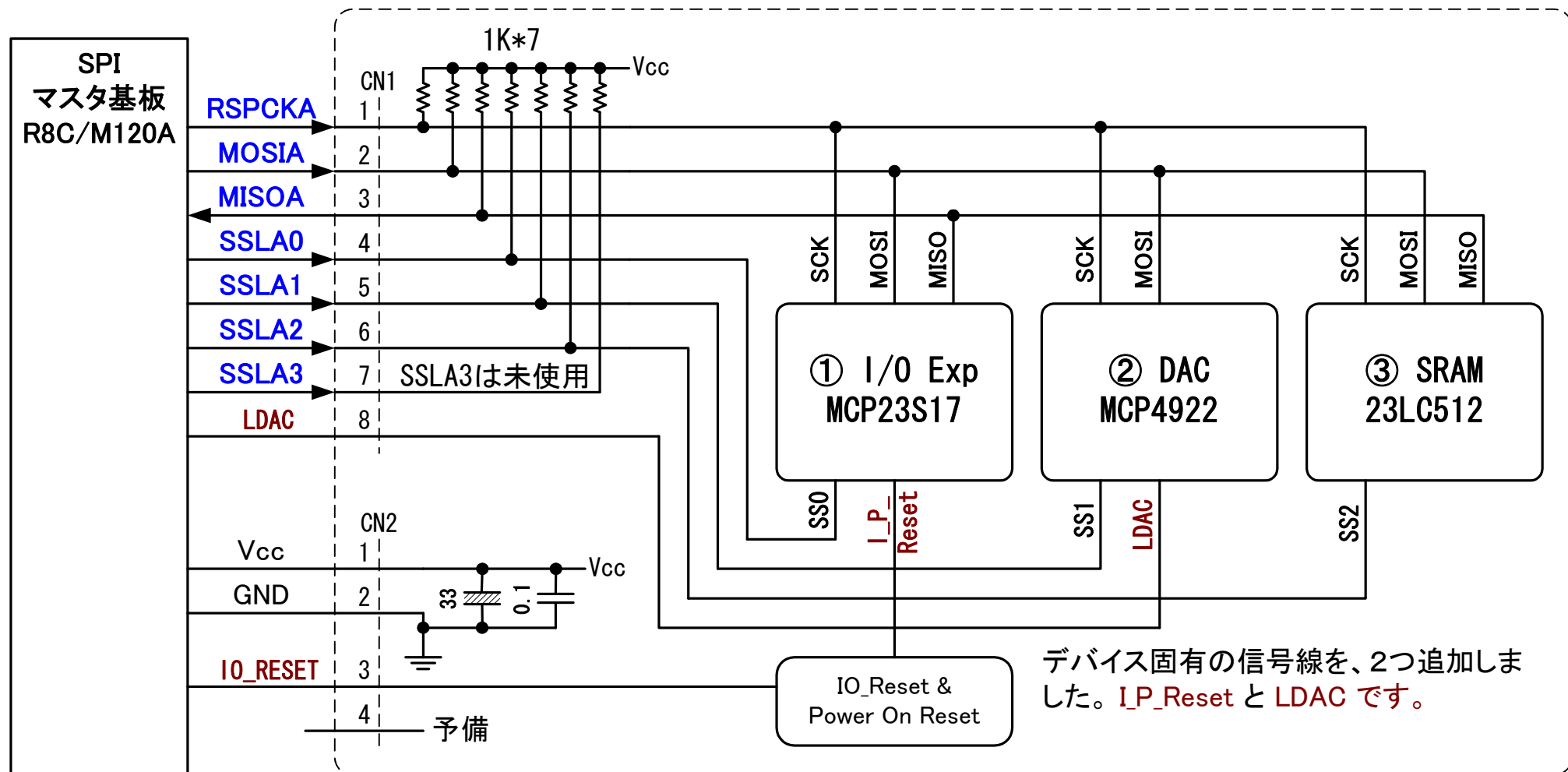
SPI マスタ、スレーブ基板間 接続ケーブルの配線表

この図を元に  
接続ケーブルを作成します。





# SPI デバイス基板 全体のブロック図

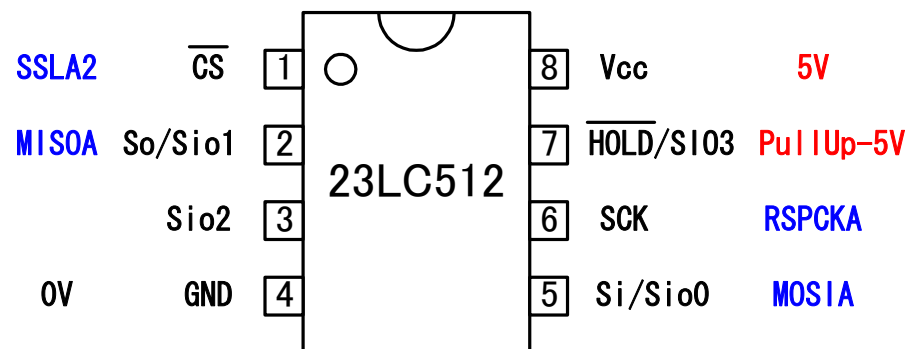


## 512Kbit SPI シリアル SRAM／23LC512

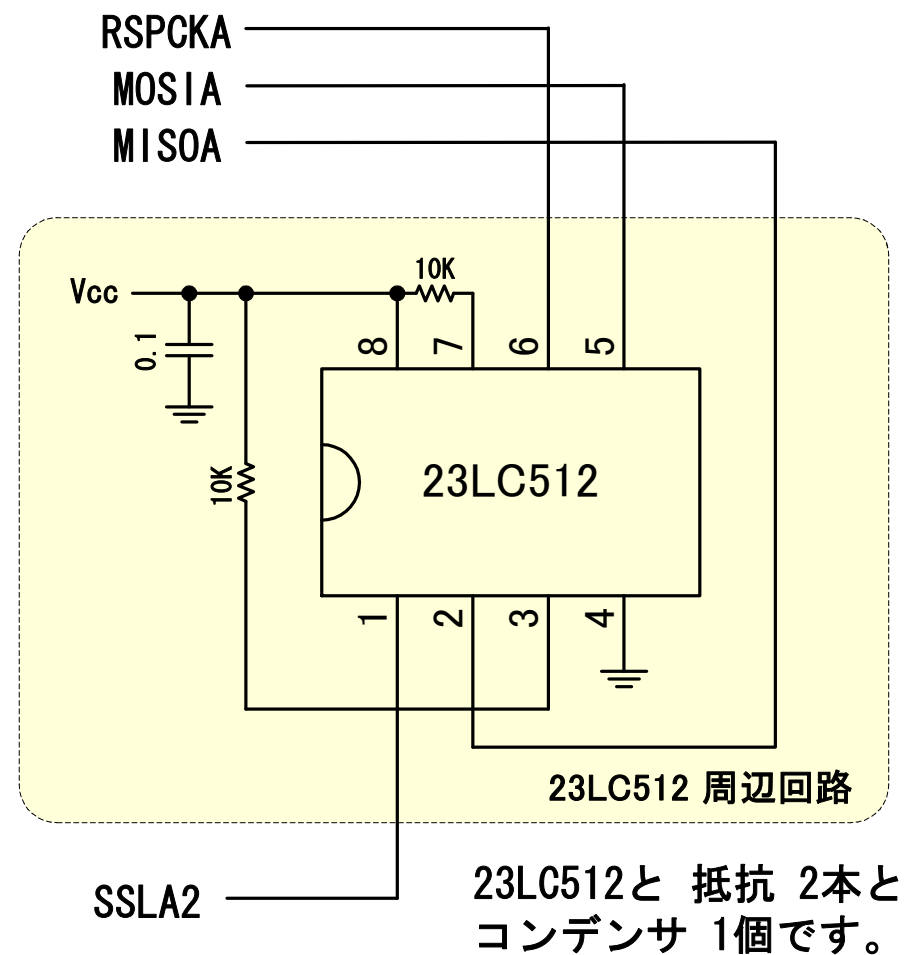
概要:

- ・ 記憶容量: 64K x 8bit / 32byte page
- ・ インタフェース: SPI ( 最大クロック: 20MHz )
- ・ 動作電圧: 2.7 ~ 5.5V
- ・ DIP 8 ピン パッケージ

23LC512のデータシートには、内部ブロック図は入ってませんでした。



## SRAM／23LC512 周辺回路



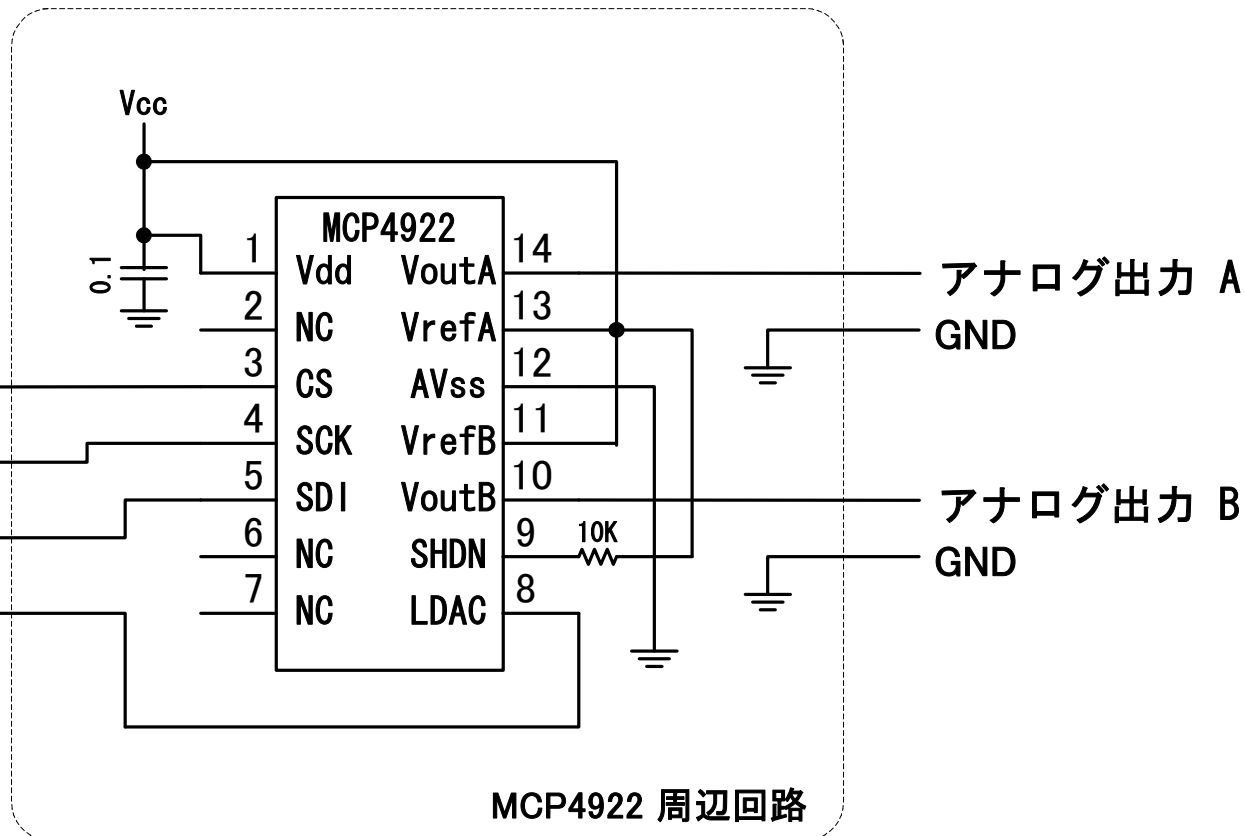
## DAC／MCP4922 周辺回路

**LDAC** この信号は、通常 **Hi** で、  
D/A変換値を 転送後、**CS**信号を  
**Hi** にした後に、**LDAC** 信号を  
**Low** に する事で、ダウンエッジで  
D/A変換したアナログ信号を出力  
するとの事です。

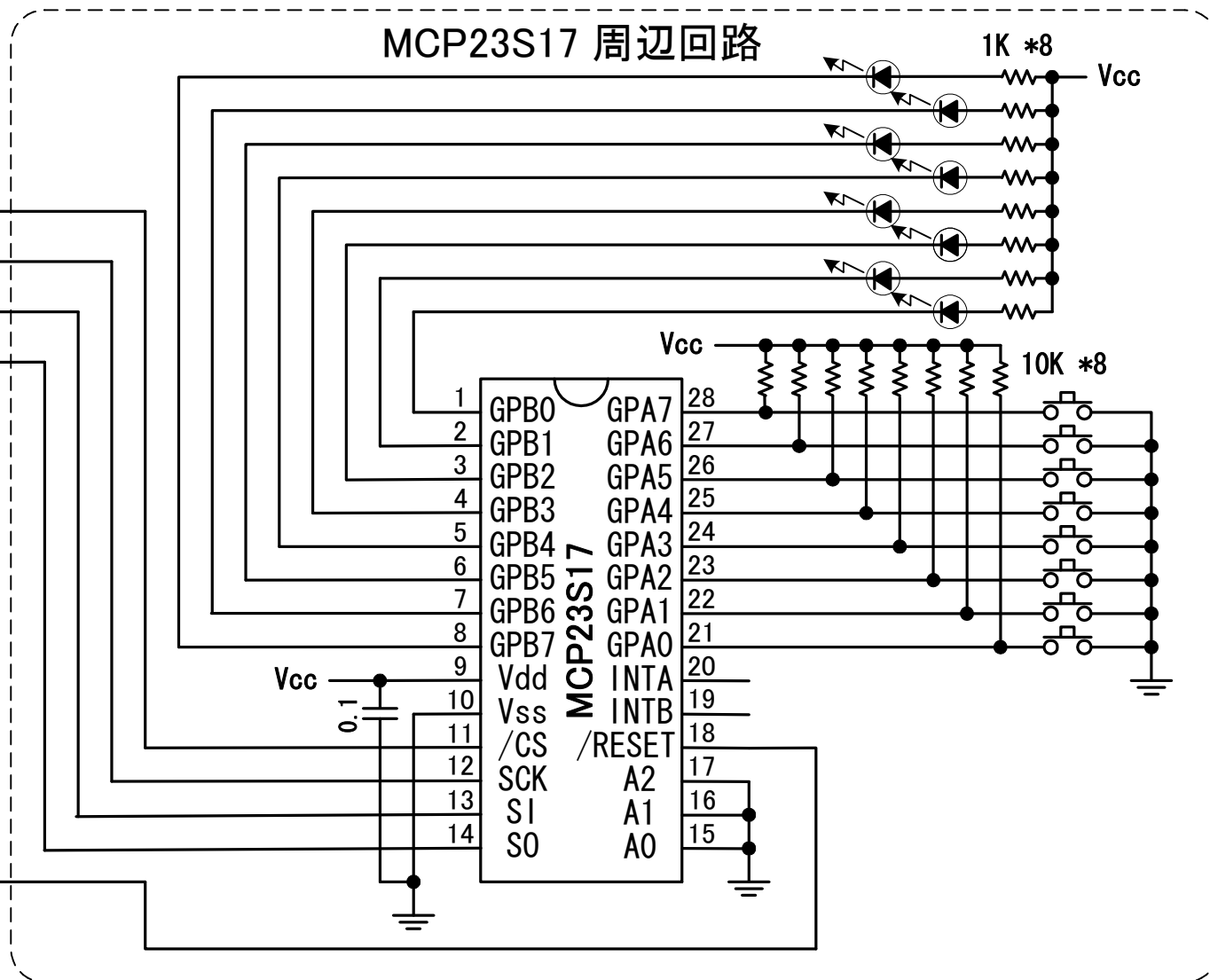
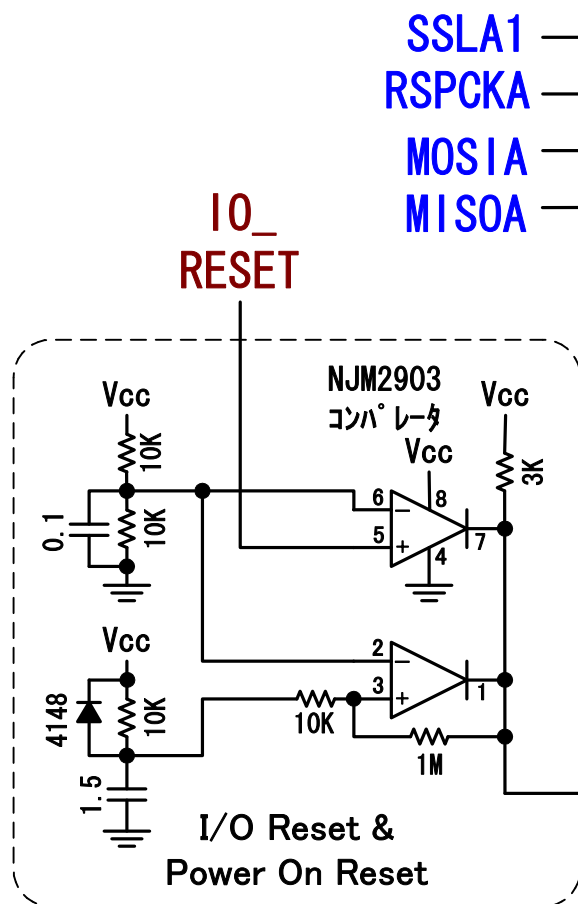
ちなみに **MISO** が、  
無い事に気付かれた  
でしょうか。？

DACは 出力デバイスで  
デバイスから、何らかの  
データを、返す事が無い  
からです。

**SSLA1**  
**RSPCKA**  
**MOSIA**  
**LDAC**



# I/O Exp MCP23S17 周辺回路





## R8CM12\_IOCS.h内の SPI関連関数プロトタイプ宣言 1

```
// -----  
// ★★★ R8CM1_IOCS_SPI_tport.a30      ★★★ ( SPI 下位層 )  
#pragma    PARAMETER    init_spi_port( R1L )  
void init_spi_port( BYTE sw ); // Soft SPI 初期化 sw=0 : HC138無し 、 sw=1 : HC138有り  
#pragma    PARAMETER    spi_byte_trans( R1L )  
BYTE spi_byte_trans( BYTE dt ); // SPI 1byte 送受信  
#pragma    PARAMETER    spi_ss_out( R1L )  
void spi_ss_out( BYTE n );      // SPI SSn 信号 有効化処理 ( n=0 から始まる デバイス番号 )  
void spi_ss_off( void );       // SPI SSn 信号 解除処理  
  
// -----  
// ★★★ R8CM1_IOCS_SPI_packet.c      ★★★ ( SPI 汎用 上位層 )  
void set_ssn( BYTE ssn );      // アクセスする スレーブ番号の設定  
void spi_w2( BYTE d1, BYTE d2 ); // 2byte 書込み (送信) のみ SS信号 制御無し  
void spi_ss_w2( BYTE d1, BYTE d2 ); // 2byte 書込み (送信) SS信号 制御有り  
BYTE spi_r1( void );           // 1 Byte SPI からデータ受信  
void spi_wn( BYTE dat[], int nn ); // 複数バイトの送信処理  
void spi_rn( BYTE dat[], int nn ); // 複数バイトの 受信処理
```

## R8CM12\_IOCS.h内の SPI関連関数プロトタイプ宣言 2

```
// ★★★ R8M_SPI_mcp23s17_access.c    ★★★( I/Oエクспанダ SPI 上位層 )
void init_mcp23s17( void );           // MCP23S17 初期化
BYTE ioexp_byte_dt_rd( BYTE adr );    // MCP23S17／指定レジスタから 1Byte 読み出し
void ioexp_byte_dt_wr( BYTE adr, BYTE dat ); // MCP23S17／指定レジスタに 1Byte 書き込み

// ★★★ R8M_SPI_23LC512_access.c    ★★★( シリアルS_RAM SPI 上位層 )
void sram_lc512write( WORD adr, BYTE dat ); // SRAM／23LC512へ 1byte 書き込み
BYTE sram_lc512read( WORD adr );           // SRAM／23LC512から 1byte 読み出し

// ★★★ R8M_SPI_MCP4922_access.c    ★★★( 12bit D/A コンバータSPI 上位層 )
void mcp4922_outA( WORD data );          // MCP4922／チャンネル A 出力
void mcp4922_outB( WORD data );          // MCP4922／チャンネル B 出力
```

これらの 関数を使用したサンプルは、main関数のある 143\_R8C\_soft\_SPI.c内の  
void [test\\_MCP4922](#)( void ); // SPI 12bit D/Aコンバータ 出力テスト  
void [test\\_SRAM\\_23LC512](#)( void ); // SPI S\_RAM アクセステスト  
void [test\\_mcp23S17](#)( void ); // I/Oエクспанダ MCP23S17 アクセステスト  
関数を 参照して下さい。ソースは ダウンロード出来るように しておきます。