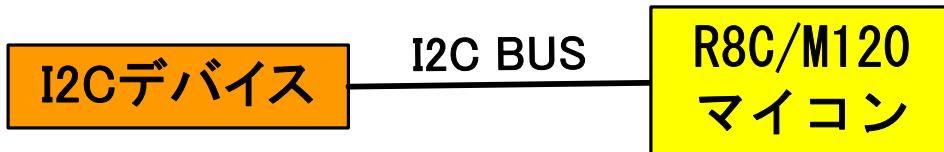


20pinの R8C/M120Aにて
400Kbpsの I2C機能を実装する



I2C バス: 通常 マイコン基板内のデバイス
との通信用途の 2線式インターフェース

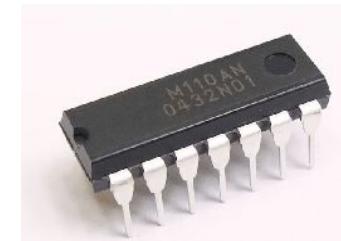
I2Cデバイス: 各種センサ(温度、湿度、気圧、
加速度、角速度、地磁気、他)
小型表示器(LCD、OLED) 等が ある。

以前、14Pinの R8C/M110Aを使用して、
USB-I2C変換器を作り 16x2文字表示の
OLEDや、温度、湿度、気圧センサをアクセスし
ました。

その後、このアダプタハード([R8C/M110A](#))を、
利用して、超小型の BitMap式のOLEDに、半角
文字を 表示する実験を行いました。

しばらくの間、14pinの M110Aばかり使用して
いたので、[20Pinの M120A](#)にて、I2Cを実装して
いろいろ実験出来るようにしようと考えました。

R8C/M110A



R8C/M120A



6Pin増えるだけですが、R8C/M110Aでは、
厳しい I/O足ピン本数が、だいぶ楽になると
思います。

また、足ピン数が少ないマイコンにとって I2Cの
2線式バスは、センサや表示器等 いろいろな物
を、接続出来る 救世主とも いえます。

R8C/M110A 主な仕様

- ・電源電圧 : 1.8~5.5V
- ・コア : R8C
- ・コアサイズ : 16bit
- ・クロック : 20MHz
- ・プログラムメモリ : 2kB
- ・EEPROM : 2kB
- ・RAM : 256B
- ・GPIO : 11pin
- ・ADC : 5 ch
- ・UART/USART : 1 ch
- ・タイマ : 3 ch
- ・オシレータ : 内蔵/外付
- ・パッケージ : DIP 14

R8C/M120A 主な仕様

- ・電源電圧 : 1.8~5.5V
- ・コア : R8C
- ・コアサイズ : 16bit
- ・クロック : 20MHz
- ・プログラムメモリ : 2kB
- ・EEPROM : 2kB
- ・RAM : 256B
- ・GPIO : 17pin
- ・ADC : 6 ch
- ・UART/USART : 1 ch
- ・タイマ : 3 ch
- ・オシレータ : 内蔵/外付
- ・パッケージ : DIP 20

上記 メモリ容量は、メーカーの カタログ表記の値です。

(実際のメモリ容量は、まだ大きいです。 次にその件について説明します。)

R8C/Mシリーズのメモリ容量について

この話題に関してはあまり触れたくなかったのですが、ルネサスとしては、競合他社の百円ローエンドマイコンとスペックを合わせたのではないかと思います。

百円マイコンは、他に出しているメーカーは、PICマイコンが、8pinシリーズを多数出しておられます。その他 アトメル社の百円マイコンがありましたが、アトメル社が、PICのMicroChip社に吸収合併されたせいか、ディスコン扱いになっています。

私は、R8Cマイコンは 第二世代の R8C/27 から使用していました。R8C/Mシリーズが発売されるようになった時、ROM : 2K、RAM : 256Bでは、使い物にならないと思っていました。

HEWで、Project生成時、ROM容量を 4K、8Kも設定出来るので 試しに 8Kでやったら ちゃんと動く。どうなってるの。？ と思ったのが

メモリ容量に対する疑問の始まりでした。

その後、ネット上で別の方も、それに気付き自分で、デバッガのようなプログラムを作成して、R8C/MシリーズCPUのメモリをダンプされて、ROM : 32KB、RAM : 1280B である事を確認されました。私も ROM : 32KB、RAM : 1280Bでもう 10年ぐらい前から使っています。

最近 Qiitaで、最強の百円マイコンの記事を出しておられる方がいて、ROM : 64KB、RAM : 1366B、実際に使えると書いておられます。

上記の記事に興味はありますか、ROMの64KBとか 実際にアクセスした事は まだないです。

私としては、当面 ROM : 32KBで不自由ないので、現状のまま使うと思います。

Fullにメモリを 使用する場合は、開発環境をgcc に切り替える必要があります。

400KbpsのI2Cを ソフトで実装

I2Cの 通信シーケンスの詳細は、024の動画にて説明していますので、そちらを参照して下さい。

前にも話したと思いますが、10年ぐらい前にR8Cマイコンで、I2Cバスのアクセス処理を、C言語にて作成した事があります。その時はビット転送速度が、100Kbpsぐらいでした。

C言語では、ビット操作が、簡単に出来ます。一見、特定の 1bitだけ、書き込んでいる様に見えますが、ハード的には、メモリや、I/Oポートを ビット単位でアクセスする事は出来ません。 バイト単位でしかアクセス出来ません。

よって、1bit変更する時は、

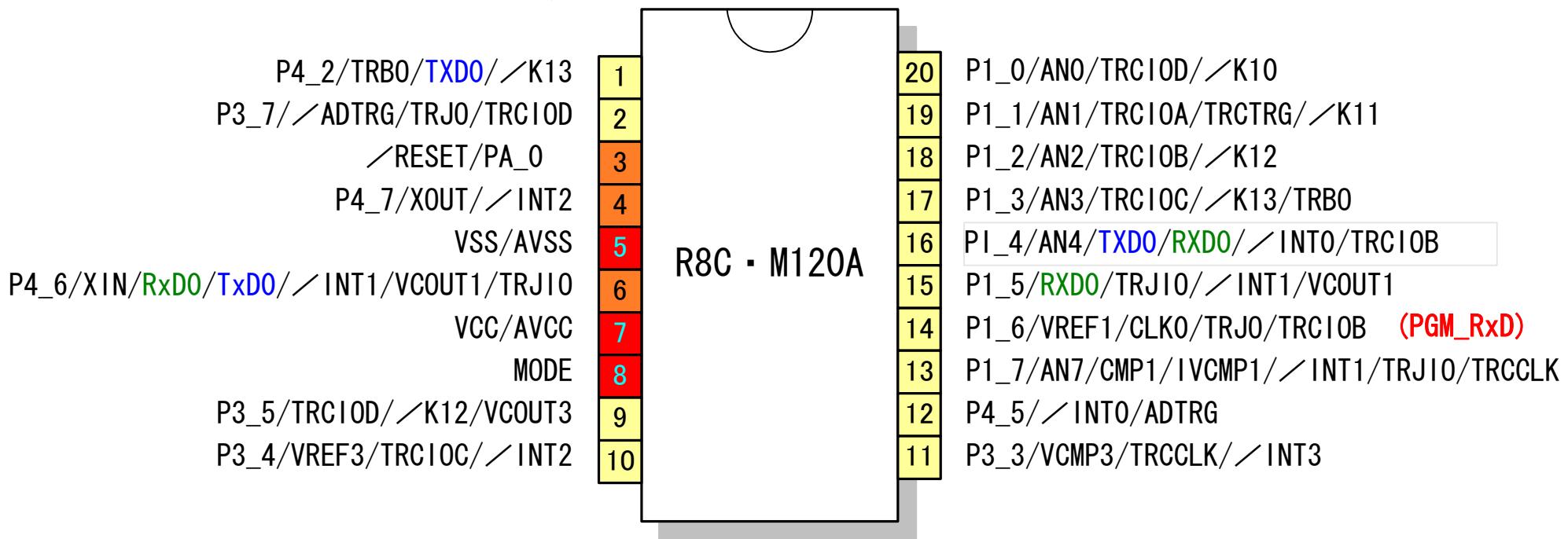
① 変更したいビットを含む バイト単位のデータを 演算レジスタ等に 読み出す。

- ② 演算レジスタ内の変更したいビットを 他のビットを壊さないようにして、1または、0にします。(具体的には AND, OR 演算)
- ③ 演算レジスタ内の データを 元あった場所に バイト単位で、書き込む。

このような、3ステップの操作を行いビットデータを更新します。そして、ROMや RAMをアクセスする際は、ノーウェイトですが、I/Oポートをアクセスする場合は、読み 書き 両方で、ウェイトサイクルが付加され 更に遅くなります。

という事で、I/Oポートを頻繁に、ビット操作でアクセスする事は、速度面で、明らかに遅くなります。よってビット操作を行う時は、I/Oポートに出力したデータの ミラーイメージを RAM上に配置して、そのRAM上で ビット操作を行い、I/Oポートに、バイト単位で出力するようにして下さい。

R8C/M120A CPUの Pinアサイン



- 1 **赤のPin** (5, 7, 8) は、他の用途では使えない。
- 2 (3, 4, 6) は、通常は、RESET、XOUT、XIN に使用するが、
どうしても Pinが 足りない時は、条件付きで 別の用途にも使える。
- 3 プログラム書き込み時は、MODE=L にして、14Pin=RxD, 16Pin=TxDとして
使用する。実行時は、TxDは 同じ16ピンで使えるが、RxDは 15Pin になるので、注意する事。

R8C/M120Aのポートレジスタ

Port. 1のレジスタは、b7~b0まで全て P1_7~P1_0にアサインされる。水色は A/D入力と重なるPin。

当然、A/D入力として使用するPinは、I/O Pinとしては 使えない。

Port. 3のレジスタは、b7, b5, b4, b3 の 4bitが P3_7, P3_5, P3_4, P3_3 に アサインされる。

Port. 4のレジスタは、通常 b5, b2 の 2bitが P4_5, P4_2 にアサインされる。

Port. Aのレジスタは、3Pin 端子を RESET信号入力として使わない場合に限り B0(PA_0) として使用出来る。 柿色の部分を使用しない場合 I/Oポートは、最大 14本使用できる事になる。

Port. 1							
b7	b6	b5	b4	b3	b2	b1	b0
P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0
13	14	15	16	17	18	19	20

Port. 3							
b7	b6	b5	b4	b3	b2	b1	b0
P3_7		P3_5	P3_4	P3_3			
2		9	10	11			

Port. 4							
b7	b6	b5	b4	b3	b2	b1	b0
P4_7	P4_6	P4_5			P4_2		
4	6	12			1		

Port. A							
b7	b6	b5	b4	b3	b2	b1	b0
							PA_0
							3

I2C(SPI)のピン割り付け

R8C/M120Aの I2Cのピン割り付けですが、Port4の **SCL=P4_5**、**SDA=P4_2** に 固定的に割り付けます。

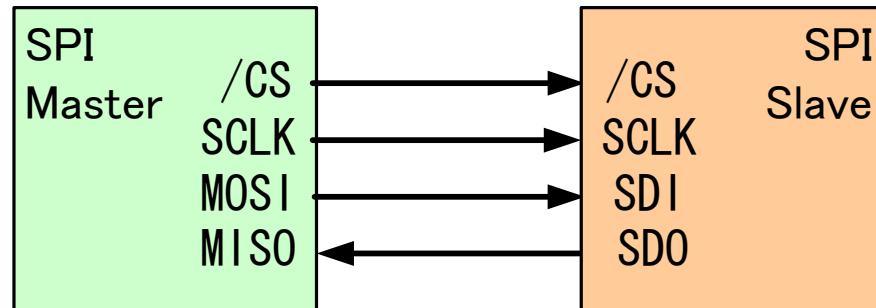
後々 SPIの接続を行う事も想定してSPIの場合は、**SCLK=P4_5**、**MOSI=P4_2**に 固定的に割り付けます。

MISOと /CS に関しては、外部発振子を使わない場合、**MISO=P4_6**、**/CS=P4_7** をデフォルトにしておきます。

外部発振子を使用する場合は、
MISO=P3_5、**/CS=P3_7** にします。

今まで、SPIインターフェースに関してはあまり説明して無かったので、ここで信号線の種類等、簡単に説明します。

SPIの接続概要



- ・チップセレクト (CS)
- ・クロック (SPICLK、SCLK)
- ・マスタ出力／スレーブ入力 (MOSI)
- ・マスタ入力／スレーブ出力 (MISO)

信号名とポート割付けの 対応表

I2C信号名	SPI信号名	水晶無し	水晶有り
SCL	SCLK	P4_5(12)	P4_5(12)
SDA	MOSI	P4_2(1)	P4_2(1)
()内の数字は CPU 足ピン 番号	/CS	P4_7(4)	P3_7(2)
	MISO	P4_6(6)	P3_5(9)

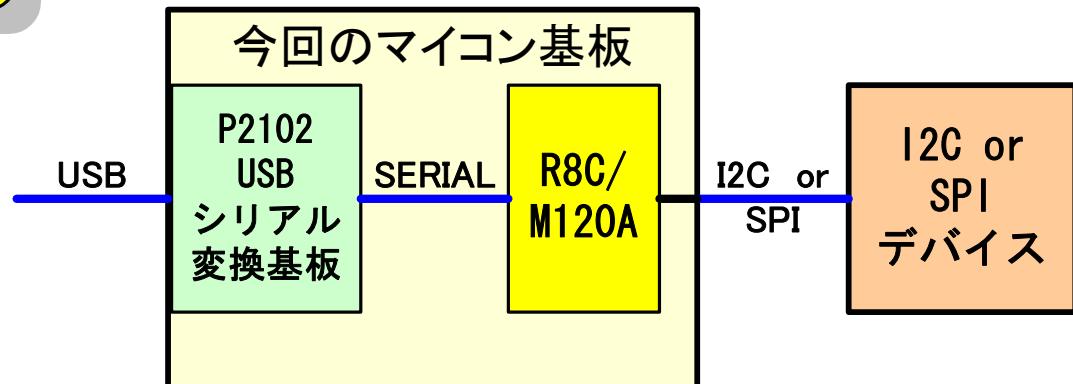
今回のマイコン基板の構成

信号名とポート割付けの 対応表

I2C信号名	SPI信号名	水晶無し
SCL	SCLK	P4_5(12)
SDA	MOSI	P4_2(1)
()内の数字は CPU 足ピン 番号	/CS	P4_7(4)
	MISO	P4_6(6)

今回は、R8C/M120Aの内蔵 20MHzオシレータを使う事にします。よって水晶を接続する端子に SPIの信号（ /CS、MISO ）を接続できます。

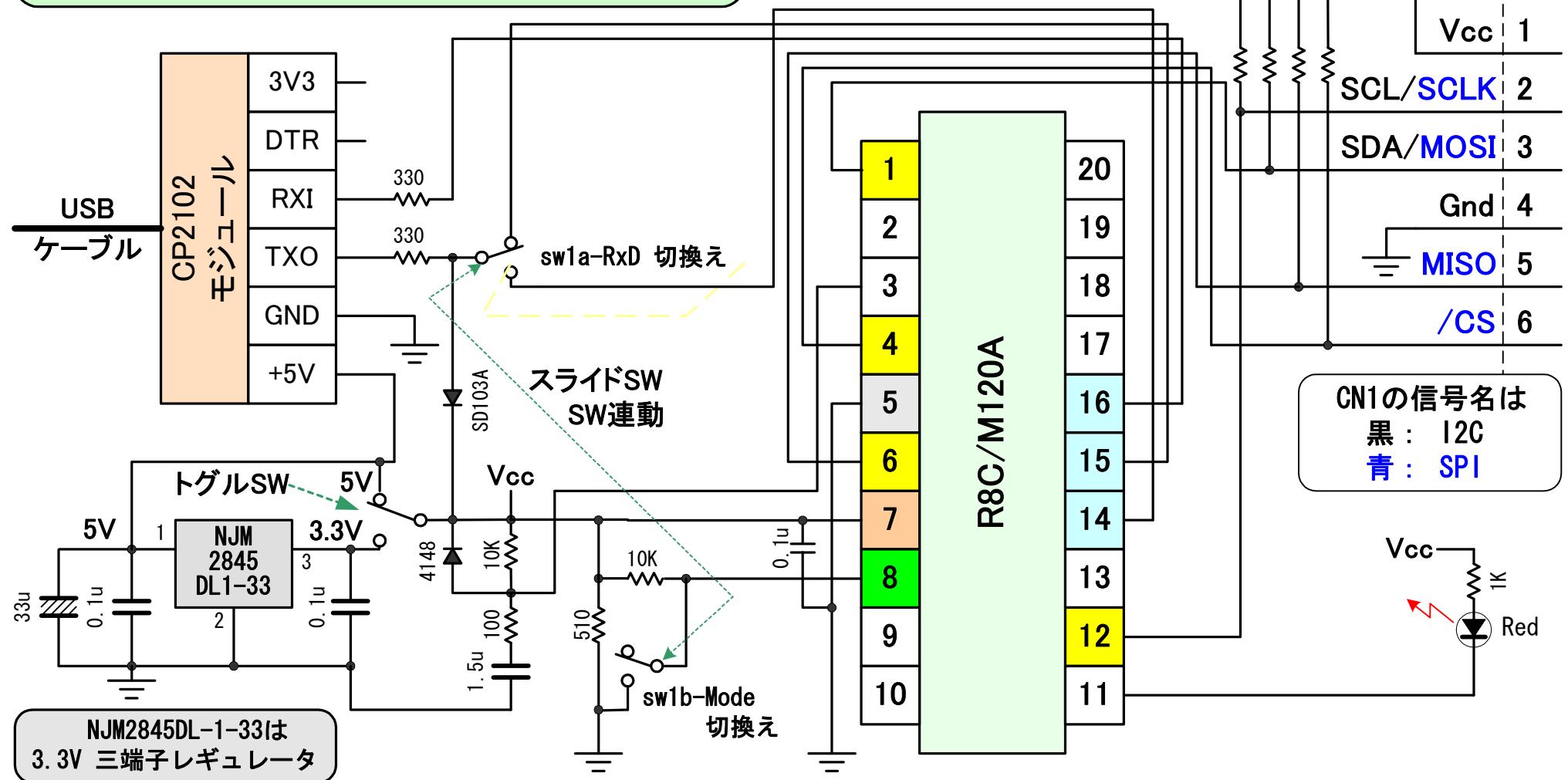
そして、計 4 本の SPI（ I2C含む ）信号線は全て Port. 4 の 信号です。元々 Port. 4には、4本しか外部端子がつながってないので、Port. 4は、I2C、SPI専用のポートという事になります。



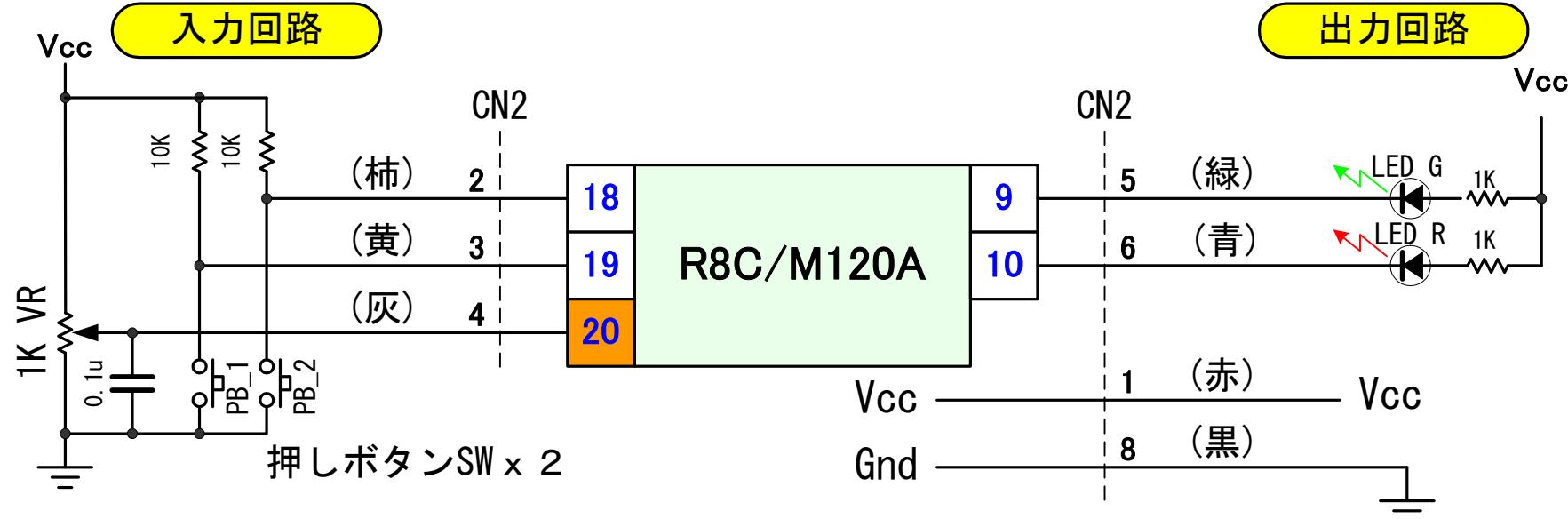
その他の回路は、基本 以前作成した USB-I2C変換アダプタと同様の構成に します。

このような構成にしておくと、R8C/M120Aのプログラム変更も USB-シリアルで、そのまま書き込みますし、プログラム実行時も、パソコンからコマンドを送って柔軟にテストする事が出来ます。

R8C/M120Aによる USB-I2C接続Unit 基本回路図

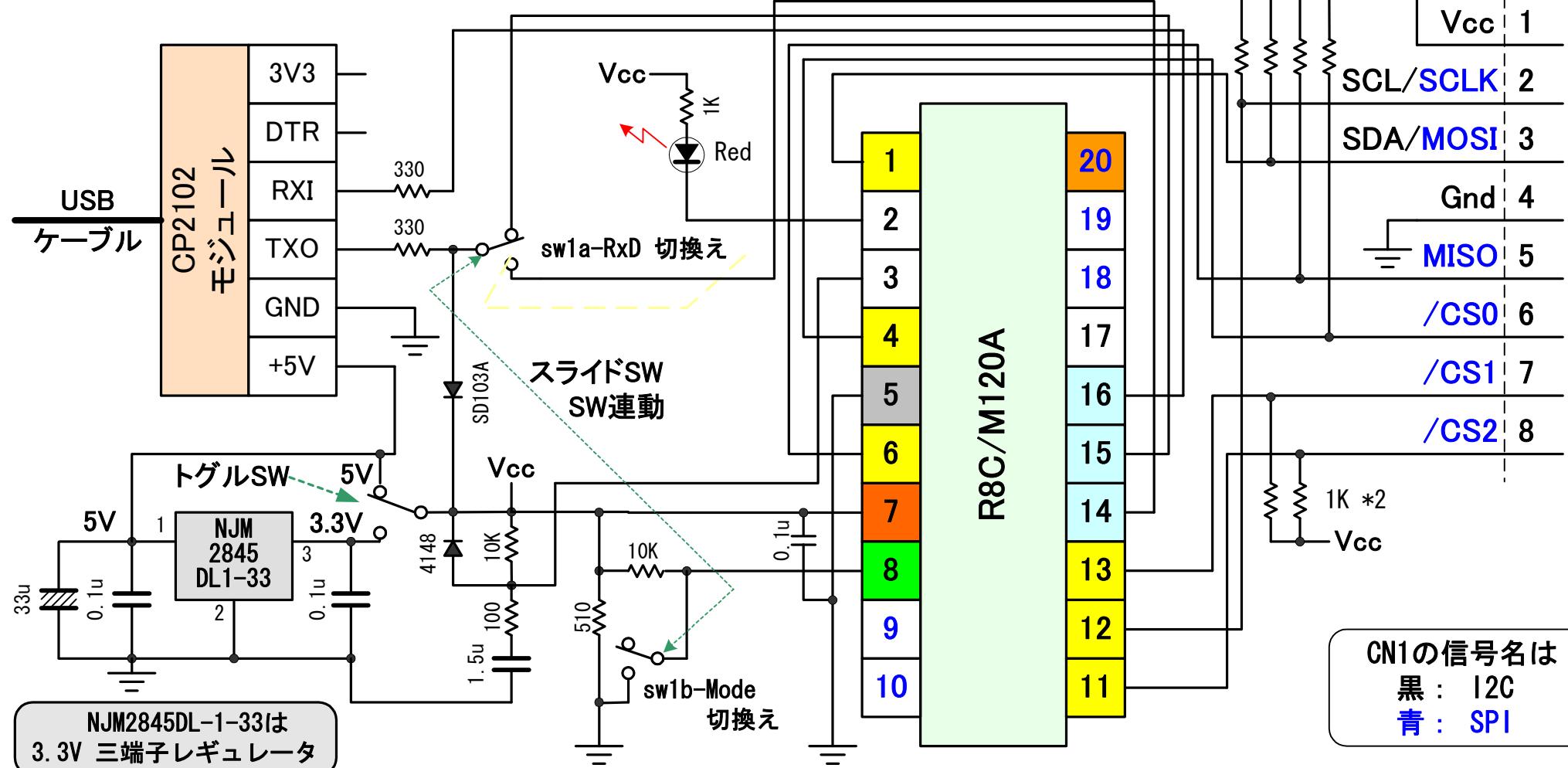


追加回路 回路図



CPUの空きピンに、若干の回路を追加する事にします。
この回路は 基板上ではなく、基板を入れるケース(多分 百均の器)に
付ける予定です。
コネクタCN2 は、8ピンのコネクタを使う予定です。7pinは、NCになります。

R8C/M120Aによる
USB-I2C接続Unit 基本回路図(変更)



R8C/M120Aにて 400Kbpsの I2C機能を実現する

No.	部品名	型式名	値	個数
1	ユニバーサル基板(小)			1
2	マイコンIC	R8C/M120A		1
3	20P DIP ICソケット			1
4	USBシリアル変換基板	CP2102		1
5	3.3V 三端子電源IC	NJM2845DL-1-33		1
6	電解コンデンサ		33uF	1
7	積層セラミックコン		1.5uF	1
8	積層セラミックコン		0.1uF	4
9	1/4W カーボン抵抗		100Ω	1
10	1/4W カーボン抵抗		330Ω	2
11	1/4W カーボン抵抗		510Ω	1
12	1/4W カーボン抵抗		1KΩ	7
13	1/4W カーボン抵抗		10KΩ	4
14	可変抵抗 (ボリウム)		1KΩB	1
15	ダイオード	1S4148		1
16	ショットキーダイオード	SD103A		1
17	基板用小型3Pトグルスイッチ	フレーム有り 5ピン		1
18	超小型スライドスイッチ	IS-2235		1
19	φ3mmLED (赤)			1
20	φ5mmLED (赤)			1
21	φ5mmLED (緑)			1
22	押しボタンスイッチ(小)			2
23	モレックス 2.5ピッチコネクタ	6ピン		1
24	モレックス 2.5ピッチコネクタ	8ピン		1
25	配線材+ハンダ	少々		
26	ケース (プラスチック)	百均で購入		1
27	φ3mm/L12~15mmネジ+ナット			4
28	樹脂スペーサ 高さ 5mm			4

今回のパーツリストです。
Excelで作成した物を、画像として貼り付けたら文字が小さくて見にくくなりました。

(回路を変更したので、1KΩは、9本です。)

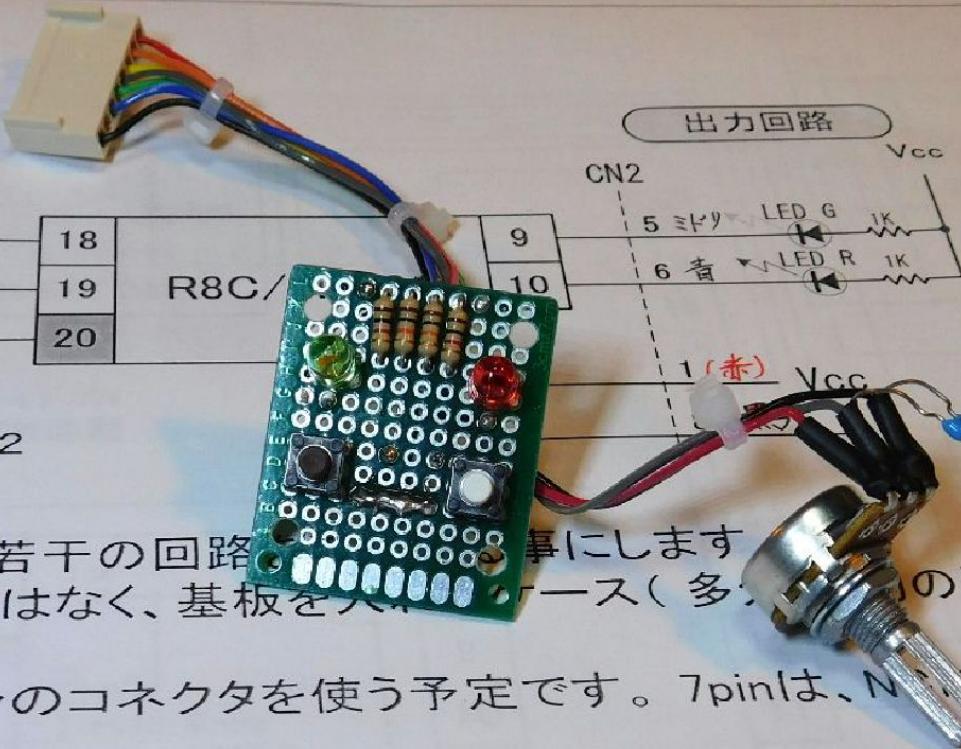
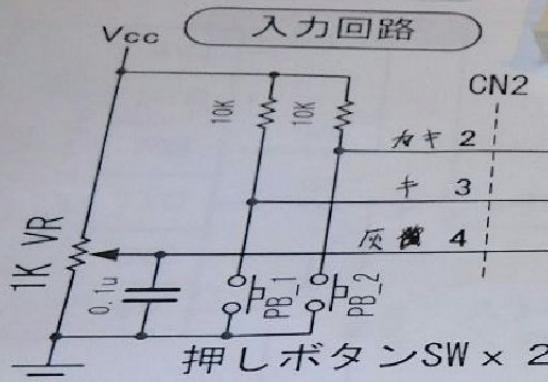
必要な方には、またダウンロード出来るようにしておきます。

では、基板の制作に入ります。





追加回路 回路図



CPUの空きピンに、若干の回路を接続する事にします。
この回路は 基板上ではなく、基板を入れ替える事で、マザーボード(多機能の器)に
付ける予定です。
コネクタCN2 は、8ピンのコネクタを使う予定です。7pinは、N/Aになります。

