

## 9. I/Oポートの入出力(プロジェクト:io)

### 9.1 概要

本章では、マイコンのクロックの切り替え、I/O ポートの入出力設定、データの出力、データの入力方法を説明します。

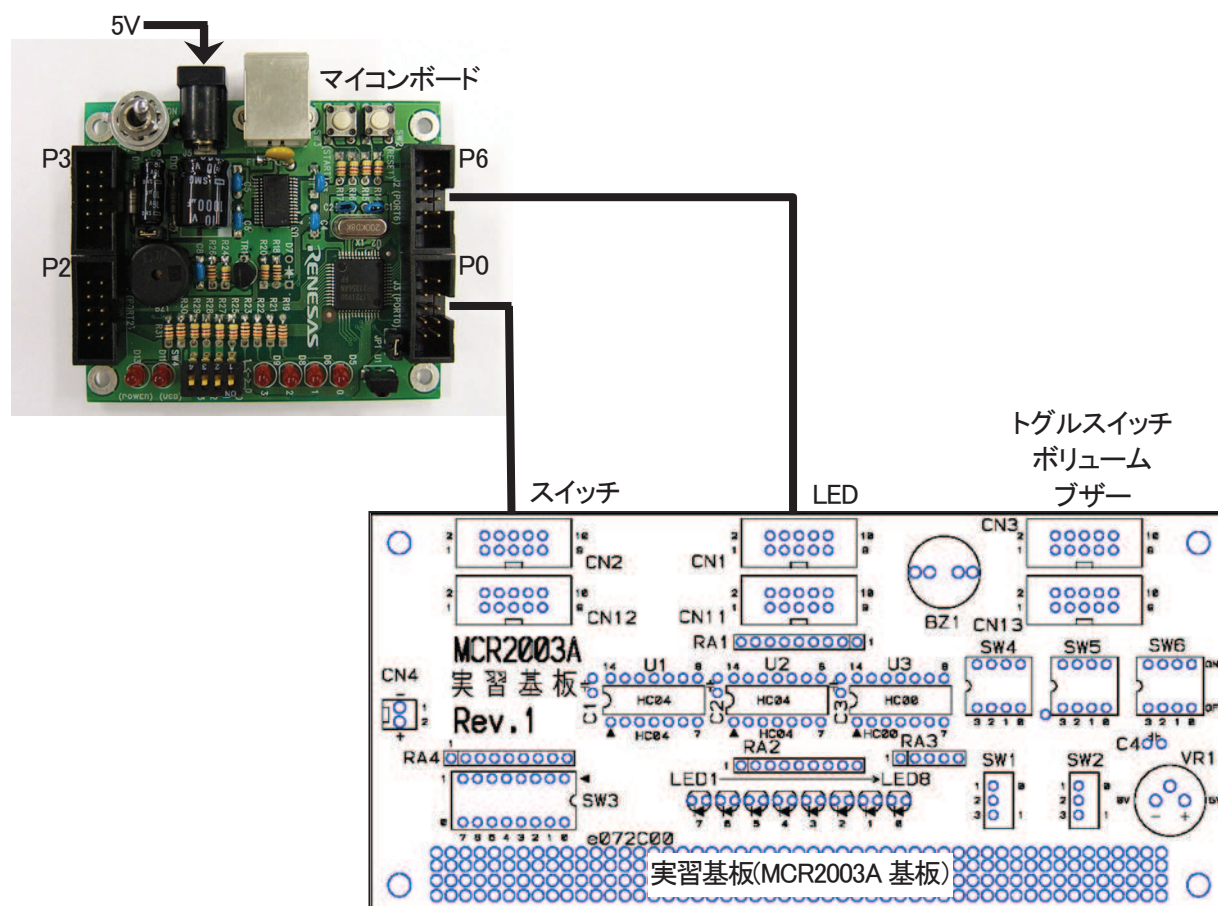
### 9.2 接続

#### ■使用ポート

マイコンのポート	接続内容
P0 (J3)	実習基板のディップスイッチ部など、入力機器を接続します。マイコンカーラリーのセンサ基板 Ver.4 も可能です。
P6 (J2)	実習基板の LED 部など、出力機器を接続します。

#### ■接続例

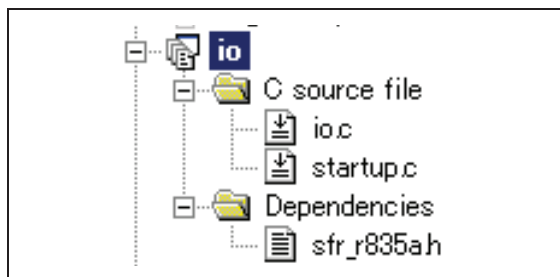
実習基板を使ったときの接続例を次に示します。



## ■操作方法

実習基板のディップスイッチ(SW3)を ON/OFF すると、それに合わせて LED(LED1～8)が ON/OFF します。入力したデータを出力する、制御の基本中の基本です。

## 9.3 プロジェクトの構成



	ファイル名	内容
1	startup.c	固定割り込みベクタアドレスの設定、スタートアッププログラム、RAM の初期化(初期値のないグローバル変数、初期値のあるグローバル変数の設定)などを行います。このファイルは共通で、どのプロジェクトもこのファイルから実行されます。
2	io.c	実際に制御するプログラムが書かれています。R8C/35A の内蔵周辺機能(SFR)の初期化も行います。
3	sfr_r835a.h	R8C/35A マイコンの内蔵周辺機能を制御するためのレジスタ (Special Function Registers)を定義したファイルです。

## 9.4 プログラム「io.c」

```

1 :  /******
2 :  /* 対象マイコン  R8C/35A
3 :  /* ファイル内容  データの入力、出力
4 :  /* バージョン   Ver. 1.20
5 :  /* Date         2010.04.19
6 :  /* Copyright    ルネサスマイコンカーラリー事務局
7 :  /*              日立インターメディックス株式会社
8 :  /******
9 :  /*
10 :  入力 : P0_7-P0_0(ディップスイッチなど)
11 :  出力 : P6_7-P6_0(LEDなど)
12 :
13 :  ポート0に入力した状態を、ポート6に出力します。
14 :  ポート0にセンサ基板、ポート6に実習基板のLED部分を接続すれば
15 :  センサ基板のチェックになります。
16 :  */
17 :
18 :  /*=====*/
19 :  /* インクルード
20 :  /*=====*/
21 :  #include "sfr_r835a.h" /* R8C/35A SFRの定義ファイル */
22 :
23 :  /*=====*/
24 :  /* シンボル定義
25 :  /*=====*/
26 :
27 :  /*=====*/
28 :  /* プロトタイプ宣言
29 :  /*=====*/
30 :  void init( void );
31 :

```

```

32 : /*****
33 : /* メインプログラム */
34 : /*****/
35 : void main( void )
36 : {
37 :     unsigned char d;
38 :
39 :     init();                /* 初期化 */
40 :
41 :     while( 1 ) {
42 :         d = p0;
43 :         p6 = d;
44 :     }
45 : }
46 :
47 : /*****
48 : /* R8C/35A スペシャルファンクションレジスタ (SFR) の初期化 */
49 : /*****/
50 : void init( void )
51 : {
52 :     int i;
53 :
54 :     /* クロックをXINクロック (20MHz)に変更 */
55 :     prc0 = 1;              /* プロテクト解除 */
56 :     cm13 = 1;              /* P4_6, P4_7をXIN-XOUT端子にする*/
57 :     cm05 = 0;              /* XINクロック発振 */
58 :     for(i=0; i<50; i++ );  /* 安定するまで少し待つ(約10ms) */
59 :     ocd2 = 0;              /* システムクロックをXINにする */
60 :     prc0 = 0;              /* プロテクトON */
61 :
62 :     /* ポートの入出力設定 */
63 :     prc2 = 1;              /* PD0のプロテクト解除 */
64 :     pd0 = 0x00;            /* スイッチなど入力 */
65 :     p1 = 0x0f;             /* 3-0:LEDは消灯 */
66 :     pd1 = 0xdf;            /* 5:RXD0 4:TXD0 3-0:LED */
67 :     pd2 = 0xfe;            /* 0:PushSW */
68 :     pd3 = 0xfb;            /* 4:Buzzer 2:IR */
69 :     pd4 = 0x80;            /* 7:XOUT 6:XIN 5-3:DIP SW 2:VREF*/
70 :     pd5 = 0x40;            /* 7:DIP SW */
71 :     pd6 = 0xff;            /* LEDなど出力 */
72 : }
73 :
74 : /*****
75 : /* end of file */
76 : /*****/

```

## 9.5 プログラムの解説

### 9.5.1 ヘッダファイルの取り込み

```

21 : #include "sfr_r835a.h"          /* R8C/35A SFR の定義ファイル */

```

「#include」はインクルードと読み、外部のファイルを取り込む命令です。取り込むファイルは「< >」、または「" ”」で囲い、その中に記述します。違いを下表に示します。

< >	通常の include フォルダから取り込むファイルを探します。 標準ライブラリとしてルネサス統合開発環境が用意しているファイルは、こちらを使います。 例えば、stdio.h , stdlib.h , math.h などです。ちなみに、これらのファイルは、 C:\¥Program Files¥Renesas¥Hew¥Tools¥Renesas¥nc30wa¥v545r00¥inc30 フォルダにあります。 <u>波線部分</u> は、ルネサス統合開発環境のバージョンにより異なります。
“ ”	まず、カレント・フォルダ (Cプログラムファイルがあるフォルダ) を探して、そこに無ければ通常の include フォルダから取り込むファイルを探します。自分で作ったファイルは、こちらを使います。

ここでは、「sfr\_r835a.h」を取り込んでいます。このヘッダファイルは、R8C/35A マイコンの内蔵周辺機能を制御するためのレジスタ (Special Function Registers) を定義したファイルです。ちなみにこのファイルは、次のフォルダにあります。

C:\¥Workspace¥common\_r8c35a

## 9.5.2 init関数(クロックの選択)

init 関数で、R8C/35A マイコンに内蔵されている機能の初期化を行います。「init」とは、「initialize (イニシャライズ)」の略で、初期化の意味です。io.c では、init 関数内でクロックの選択とI/Oポートの入出力設定を行います。ここでは、クロックの選択について説明します。

```

50 : void init( void )
51 : {
52 :     int i;
53 :
54 :     /* クロックをXINクロック (20MHz)に変更 */
55 :     prc0 = 1;                /* プロテクト解除          */
56 :     cm13 = 1;                /* P4_6, P4_7をXIN-XOUT端子にする*/
57 :     cm05 = 0;                /* XINクロック発振          */
58 :     for(i=0; i<50; i++ );    /* 安定するまで少し待つ(約10ms) */
59 :     ocd2 = 0;                /* システムクロックをXINにする */
60 :     prc0 = 0;                /* プロテクトON            */

```

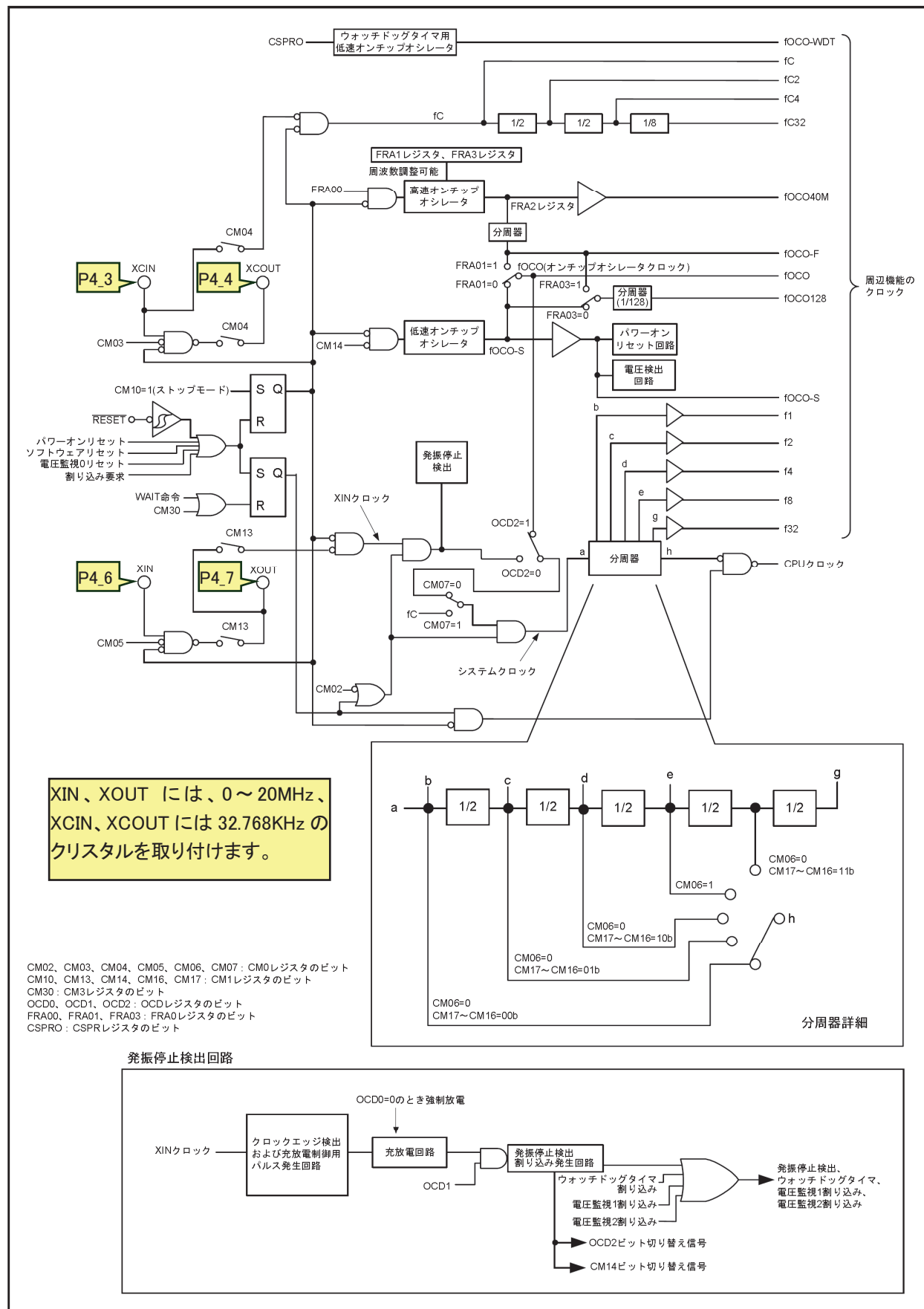
## (1) R8C/35Aのクロック

R8C/35A は、2 種類のオシレータ(発振器)が内蔵されています。また、2 種類のクリスタルを外付けすることができます。どのクロックを使うかは、プログラムで設定します。下表にまとめます。

	XIN クロック 発信回路	XCIN クロック 発信回路	高速オンチップ オシレータ (内蔵クロック)	低速オンチップ オシレータ (内蔵クロック)	ウォッチドッグタイマ用 低速オンチップオシレータ (内蔵クロック)
接続端子	XIN(P4_6)、 XOUT(P4_7) に接続	XCIN(P4_3)、 XCOUT(P4_4) に接続	マイコン内蔵	マイコン内蔵	マイコン内蔵
クロック周 波数	0～20MHz	32.768KHz	約 40MHz	約 125KHz	約 125KHz
リセット後 の状態	停止	停止	停止	<b>発振</b>	停止、発振はROMの 特定の番地書き込 む値で決めます
備考	<b>ミニマイコンカー Ver.2 では 20MHz のクリスタルを実 装しています。リ セット後、プログラ ムでこのクロック に切り替えます。</b>	取り付けて いません。	R8C/35A には搭 載されていませ ん(R8C/35C に は搭載されてい ます)。	リセット直後のみ 使用します。すぐ に、XIN に切り替 えます。	使用しません。

## (2) ブロック図

R8C/35A のクロック発生回路を下記に示します。図の状態は、リセット直後の状態です。



### (3) クロックの設定

R8C/35A マイコンは起動時、低速オンチップオシレータ(約 125KHz)で動作しています。init 関数内のプログラムで XIN クロック発振回路に切り替えて、20MHz で動作させます。レジスタの設定手順を下記に示します。



## ①プロテクトレジスタ(PCR:Protect register)の設定(書き込み許可)

R8C/35A マイコンには、プログラムが暴走したときに備え重要なレジスタは簡単に書き替えられないように保護するレジスタがあります。プロテクトレジスタ(PCR)は、プロテクトがかかっているレジスタの書き込みを禁止、または許可するレジスタです。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～4		"0000"を設定	変更 なし
bit3	プロテクトビット 3 prc3	OCVREFCR、VCA2、VD1LS、VW0C、VW1C、VW2C レジスタへの書き込み許可 0:書き込み禁止 1:書き込み許可	変更 なし
bit2	プロテクトビット 2 prc2	PD0 レジスタへの書き込み許可 0:書き込み禁止 1:書き込み許可(注 1)	変更 なし
bit1	プロテクトビット 1 prc1	PM0、PM1 レジスタへの書き込み許可 0:書き込み禁止 1:書き込み許可	変更 なし
bit0	プロテクトビット 0 prc0	CM0、CM1、CM3、OCD、FRA0、FRA1、FRA2、FRA3 レジスタへの書き込み許可 0:書き込み禁止 1:書き込み許可	1

注 1. PRC2 ビットは"1"を書いた後、任意の番地に書き込みを実行すると、"0"になります。他のビットは"0"になりませんので、プログラムで"0"にしてください。

これからCM1、CM0、OCDの設定をします。これらのレジスタはプロテクトレジスタ(PCR)で保護されており、値を変更するには保護を解除します。今回、これらのレジスタを書き込み許可するために bit0 を"1"にします。bit0 のみ OR 演算で"1"にします。下記にプログラムを示します。

```
prcr |= 0x01;
```

もう1つ、設定方法があります。上表の左から2列目に「下:シンボル」とありますが、各レジスタのビットに名前を付けています。プロテクトレジスタ(PCR)の bit0 は、表より「**prc0**」という名前が付いています。よって下記のようにプログラムすることもできます。今回のプログラムでは、「prc0」として bit0 を"1"にしています。

```
55 :      prc0  = 1;                      /* プロテクト解除          */
```





## ③システムクロック制御レジスタ 0(CM0: System clock control register0)の設定

XIN クロックを発振させます。CM0 レジスタは、PRCRレジスタの PRC0 ビットを“1”(書き込み許可)にした後で書き換えてください。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	XIN、XCIN クロック選択ビット (注 7) cm07	0:XIN クロック 1:XCIN クロック	変更 なし
bit6	CPUクロック分周比選択ビット 0(注 4) cm06	0:CM1 レジスタの CM16、CM17 ビット有効 1:8 分周モード	変更 なし
bit5	XIN クロック(XIN-XOUT)停止 ビット(注 1、3) cm05	0:発振 1:停止(注 2)	0
bit4	ポート、XCIN-XCOUT 切り替 えビット(注 5) cm04	0:入出力ポート P4_3、P4_4 1:XCIN、XCOUT 端子(注 6)	変更 なし
bit3	XCIN クロック停止ビット cm03	0:発振 1:停止	変更 なし
bit2	ウェイトモード時周辺機能クロ ック停止ビット cm02	0:ウェイトモード時、周辺機能クロックを停止しない 1:ウェイトモード時、周辺機能クロックを停止する	変更 なし
bit1,0		“00”を設定	変更 なし

注 1. CM05 ビットは高速オンチップオシレータモード、低速オンチップオシレータモードにすると XIN クロックを停止させるビットです。XIN クロックが停止したかどうかの検出には使えません。XIN クロックを停止させる場合、次のようにしてください。

- (1) OCD レジスタの OCD1～OCD0 ビットを“00b”にする。
- (2) OCD2 ビットを“1”(オンチップオシレータクロック選択)にする。

注 2. 外部クロック入力時には、クロック発振バッファだけ停止し、クロック入力は受け付けられます。

注 3. CM05 ビットが“1”(XIN クロック停止) かつ CM1 レジスタの CM13 ビットが“0”(P4\_6、P4\_7) の場合のみ、P4\_6、P4\_7 は入出力ポートとして使用できます。

注 4. ストップモードへの移行時、CM06 ビットは“1”(8 分周モード)になります。

注 5. CM04 ビットはプログラムで“1”にできますが、“0”にできません。

注 6. XCIN クロックを使用する場合、CM04 ビットを“1”、PINSR レジスタの XCSEL ビットを“1”にしてください。  
また、ポート P4\_3、P4\_4 は入力ポートで、プルアップなしにしてください。

注 7. CM04 ビットを“1”(XCIN-XCOUT 端子)にし、XCIN クロックの発振が安定した後に、CM07 ビットを“0”から“1”(XCIN クロック)にしてください。

XIN クロック(XIN-XOUT)停止ビット(CM05)を“0”にして、XIN クロックを発振させます。CM0 の bit5 を“0”にします。今回は、ビット指定で CM05 を“0”にします。

57 :	cm05 = 0;	/* XIN クロック発振	*/
------	-----------	---------------	----

## ④ウェイト

③で XIN クロックを発振させました。発振が安定するまで 10ms 程度、待ちます。  
プログラムでは、for 文を使って、「50 回分何もせずに繰り返す」命令を実行します。実測で約 10ms です。

```
58 :      for(i=0; i<50; i++ );          /* 安定するまで少し待つ(約 10ms) */
```

プログラムの実行時間は、机上での計算はできないので、実測する必要があります。

実測の方法は、プログラム実行前に空いているポートの端子を"1"にして、プログラム実行後にその端子を"0"にします。端子の"1"の時間を、オシロスコープなどで測定すると実行時間が分かります。

ポート 6 を使って実行時間を計測するプログラム例を、下記に示します。ポート 6 はすべて空き端子とします。

```
p6 = 0x00;
```

```
pd6 = 0xff;
```

**最初に実行 ポート 6 の値をすべて"0"にする**

**最初に実行 ポート 6 の入出力設定 すべて出力**

~~~~~

```
p6 = 0xff;
```

```
for(i=0; i<50; i++ );
```

```
p6 = 0x00;
```

**// ポート 6 の値をすべて"1"にする**

**/\* 安定するまで少し待つ(約 10ms) \*/**

**// ポート 6 の値をすべて"0"にする**

## ⑤発振停止検出レジスタ(OCD:Oscillation stop detection register)の設定

システムクロックを低速オンチップオシレータ(約 125KHz)から、XIN クロックに切り替えます。OCD レジスタは、PRCR レジスタの PRC0 ビットを“1”(書き込み許可)にした後、書き換えてください。

| 設定 bit | 上:ビット名<br>下:シンボル            | 内容                                           | 今回の<br>内容 |
|--------|-----------------------------|----------------------------------------------|-----------|
| bit7～4 |                             | “0000”を設定                                    | 変更<br>なし  |
| bit3   | クロックモニタビット(注 4、5)<br>ocd3   | 0:XIN クロック発振<br>1:XIN クロック停止                 | 変更<br>なし  |
| bit2   | システムクロック選択ビット(注 3)<br>ocd2  | 0:XIN クロック選択(注 6)<br>1:オンチップオシレータクロック選択(注 2) | 0         |
| bit1   | 発振停止検出割り込み許可<br>ビット<br>ocd1 | 0:禁止(注 1)<br>1:許可                            | 変更<br>なし  |
| bit0   | 発振停止検出有効ビット(注 6)<br>ocd0    | 0:発振停止検出機能無効(注 1)<br>1:発振停止検出機能有効            | 変更<br>なし  |

注 1. ストップモード、高速オンチップオシレータモード、低速オンチップオシレータモード(XIN クロック停止) に移行する前に OCD1～OCD0 ビットを“00b”に設定してください。

注 2. OCD2 ビットを“1”(オンチップオシレータクロック選択)にすると、CM14 ビットは“0”(低速オンチップオシレータ発振)になります。

注 3. OCD2 ビットは、OCD1～OCD0 ビットが“11b”のときに XIN クロック発振停止を検出すると、自動的に“1”(オンチップオシレータクロック選択)に切り替わります。また、OCD3 ビットが“1”(XIN クロック停止)のとき、OCD2 ビットに“0”(XIN クロック選択)を書いても変化しません。

注 4. OCD3 ビットは OCD0 ビットが“1”(発振停止検出機能有効)のとき有効です。

注 5. OCD1～OCD0 ビットが“00b”のとき OCD3 ビットは“0”(XIN クロック発振)になり、変化しません。

注 6. 発振停止検出後、XIN クロックが再発振した場合の切り替え手順は、R8C/35A ハードウェアマニュアルの「図 9.10 低速オンチップオシレータから XIN クロックへの切り替え手順」を参照してください。

先の設定で、XIN クロック(XIN-XOUT)停止ビット(CM05)を“0”にして、XIN クロックを発振させました。システムクロックを低速オンチップオシレータ(約 125KHz)から、XIN クロックに切り替えます。ここから、動作が 20MHz になります。OCD の bit2 を“0”にします。今回は、OCD2 を“0”にします。

```
59 :      ocd2  = 0;                      /* システムクロックを XIN にする */
```

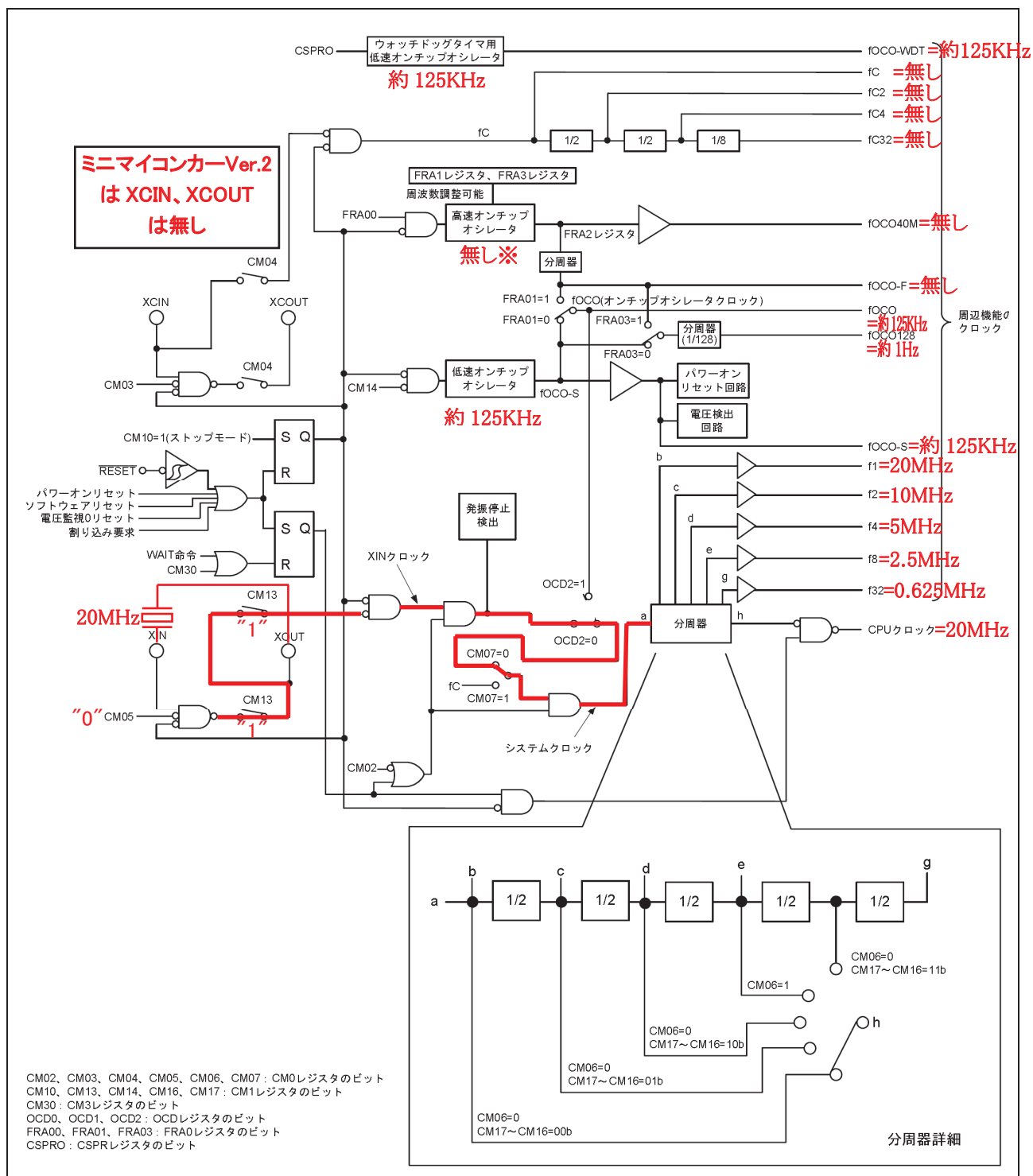
## ⑥プロテクトレジスタ(PRCR:Protect register)の設定(書き込み禁止)

①で書き込みを許可しました。それを、元に戻します(書き込み禁止にします)。

```
60 :      prc0  = 0;                      /* プロテクト ON */
```

## (4) 設定後のクロックの状態

今回の設定をしたときの、クロック発生回路を下記に示します。



### 9.5.3 init関数(I/Oポートの入出力設定)

init 関数内でクロックの選択と、I/O ポートの入出力設定を行っています。ここでは I/O ポートの入出力について説明します。プログラムを下記に示します。

|      |                 |                                     |    |
|------|-----------------|-------------------------------------|----|
| 62 : | /* ポートの入出力設定 */ |                                     |    |
| 63 : | prc2 = 1;       | /* PD0のプロテクト解除                      | */ |
| 64 : | pd0 = 0x00;     | /* スイッチなど入力                         | */ |
| 65 : | p1 = 0x0f;      | /* 3-0:LEDは消灯                       | */ |
| 66 : | pd1 = 0xdf;     | /* 5:RXD0 4:TXD0 3-0:LED            | */ |
| 67 : | pd2 = 0xfe;     | /* 0:PushSW                         | */ |
| 68 : | pd3 = 0xfb;     | /* 4:Buzzer 2:IR                    | */ |
| 69 : | pd4 = 0x80;     | /* 7:XOUT 6:XIN 5-3:DIP SW 2:VREF*/ |    |
| 70 : | pd5 = 0x40;     | /* 7:DIP SW                         | */ |
| 71 : | pd6 = 0xff;     | /* LEDなど出力                          | */ |
| 72 : | }               |                                     |    |

#### (1) I/Oポートとは

I/O ポートは、Input／Output Port の略で、直訳すると「**入力や出力を行う港**」という意味です。今回は「入力、出力をまとめて行う場所」というような意味合いです。I/O ポートの入出力設定は、プログラムの初めに行います。

#### (2) ポートの入出力設定

R8C/35A には、ポート 0～ポート 6 まで 7 本のポートがあります。基本的には 1 ポート 8 ビットですが、8 ビットないポートもあります。それぞれの端子を入力用として使用するか、出力用として使用するか、表にまとめてみます。

考え方は、次のとおりです。

- ディップスイッチなど、外部からの信号をマイコンへ入力する場合、端子を「入力」にします。
- LED など、マイコンから外部へ信号を出力する場合、端子を「出力」にします。

今回の実習でのポートの一覧を下記に示します。ゴシック体部分は、実習基板を接続した場合の内容です。

| ポート | bit7                    | bit6               | bit5                    | bit4                    | bit3                     | bit2                            | bit1                     | bit0                       |
|-----|-------------------------|--------------------|-------------------------|-------------------------|--------------------------|---------------------------------|--------------------------|----------------------------|
| 0   | ディップ<br>スイッチ<br>入力      | ディップ<br>スイッチ<br>入力 | ディップ<br>スイッチ<br>入力      | ディップ<br>スイッチ<br>入力      | ディップ<br>スイッチ<br>入力       | ディップ<br>スイッチ<br>入力              | ディップ<br>スイッチ<br>入力       | ディップ<br>スイッチ<br>入力         |
| 1   | 未接続                     | 未接続                | RxD0<br>入力              | TxD0<br>出力              | マイコンボード<br>上の LED3<br>出力 | マイコンボード<br>上の LED2<br>出力        | マイコンボード<br>上の LED1<br>出力 | マイコンボード<br>上の LED0<br>出力   |
| 2   | 未接続                     | 未接続                | 未接続                     | 未接続                     | 未接続                      | 未接続                             | 未接続                      | マイコンボード<br>上のタクトスイッ<br>チ入力 |
| 3   | 未接続                     | 未接続                | 未接続                     | マイコンボー<br>ド上のブザー<br>出力  | 未接続                      | マイコンボー<br>ド上の赤外線<br>受光 IC<br>入力 | 未接続                      | 未接続                        |
| 4   | クリスタル<br>出力             | クリスタル<br>入力        | マイコンボード<br>上の SW2<br>入力 | マイコンボード<br>上の SW1<br>入力 | マイコンボード<br>上の SW0<br>入力  | Vcc<br>入力                       |                          |                            |
| 5   | マイコンボード<br>上の SW3<br>入力 | 未接続                |                         |                         |                          |                                 |                          |                            |
| 6   | LED<br>出力               | LED<br>出力          | LED<br>出力               | LED<br>出力               | LED<br>出力                | LED<br>出力                       | LED<br>出力                | LED<br>出力                  |

※表の斜線の bit は、端子がない bit です。

※リセット後は、全て入力ポートです。

ポートの入出力方向の設定は、「ポート Pi 方向レジスタ」(i は 0～6 の数字が入ります)で行い、レジスタ名は「pd0～pd6」の 7 個あります。最後の数字がポート番号です。例えばポート 0 は「PD0」となります。

ポート Pi 方向レジスタの設定方法を下記に示します。

- ①外部へ信号を出力する端子は、“1”を設定する
- ②外部から信号を入力する端子は、“0”を設定する
- ③未接続の端子は、プルアップ抵抗、またはプルダウン抵抗を接続して入力(“0”)にするか、何も接続せずに出力(“1”)にする。今回は、後者で設定する
- ④端子が無い場合(表の斜線部分)は、“0”にする

①～④の考え方を基に、“1”、“0”で書き換えた表を下記に示します。

| ポート | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 | 16進数 |
|-----|------|------|------|------|------|------|------|------|------|
| 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0x00 |
| 1   | 1    | 1    | 0    | 1    | 1    | 1    | 1    | 1    | 0xdf |
| 2   | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0xfe |
| 3   | 1    | 1    | 1    | 1    | 1    | 0    | 1    | 1    | 0xfb |
| 4   | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0x80 |
| 5   | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0x40 |
| 6   | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0xff |

C言語は2進数表記はできないので、10進数か16進数に変換します。通常は、2進数を16進数に変換する方が簡単なため、16進数に変換します。

表より、ポート P0 方向レジスタ(PD0)～ポート P6 方向レジスタ(PD6)に設定する値を、下記に示します。

| ポート | 方向レジスタ名 | 設定値  |
|-----|---------|------|
| 0   | PD0     | 0x00 |
| 1   | PD1     | 0xdf |
| 2   | PD2     | 0xfe |
| 3   | PD3     | 0xfb |
| 4   | PD4     | 0x80 |
| 5   | PD5     | 0x40 |
| 6   | PD6     | 0xff |

プログラムを下記に示します。

|      |             |                                     |    |
|------|-------------|-------------------------------------|----|
| 63 : | prc2 = 1;   | /* PD0のプロテクト解除                      | */ |
| 64 : | pd0 = 0x00; | /* スイッチなど入力                         | */ |
| 65 : | p1 = 0x0f;  | /* 3-0:LEDは消灯                       | */ |
| 66 : | pd1 = 0xdf; | /* 5:RXD0 4:TXD0 3-0:LED            | */ |
| 67 : | pd2 = 0xfe; | /* 0:PushSW                         | */ |
| 68 : | pd3 = 0xfb; | /* 4:Buzzer 2:IR                    | */ |
| 69 : | pd4 = 0x80; | /* 7:XOUT 6:XIN 5-3:DIP SW 2:VREF*/ |    |
| 70 : | pd5 = 0x40; | /* 7:DIP SW                         | */ |
| 71 : | pd6 = 0xff; | /* LEDなど出力                          | */ |

63行は、次のページで説明します。

## ※PD0を設定するときの注意点

R8C/35A マイコンには、プログラムが暴走したときに備え重要なレジスタは簡単に書き換えられないように保護するレジスタがあります。

PD0～PD6 の中で、PD0 だけは保護(プロテクト)されており、保護を解除しないと書き換えることができません。PD0 を書き換える前に、プロテクトレジスタ(PCR)の bit2 を"1"にします。このビットは「sfr\_r835a.h (R8C/35A のレジスタ定義ファイル)」で「PRC2」と定義されており、「PRC2」と記述すると PCR の bit2 の意味になります。プログラムを下記に示します。

```
prc2 = 1;      ←PD0の書き換えを許可 (PRC2=PCRのbit2の意味です)
pd0 = 0x00;    ←その後、書き換える
```

PRC2 を"1"にした後、次に何かの命令を実行すると PRC2 は自動的に"0"(書き換え不可)になります。そのため、**必ず PRC2 を"1"にした次の行で PD0 を設定してください。**プログラムで PRC2 を"0"に戻す必要はありません。

## (3) データの出力

出力に設定した端子からデータを出力するときの方法を説明します。

データを出力するときは、「**ポート Pi レジスタ**」(i は 0～6 の数字が入ります)で行い、**レジスタ名は「P0～P6」の 7 個あります。**最後の数字がポート番号です。例えばポート 0 は「P0」となります。

ポート Pi レジスタの設定値は、次のようになります。

- ①端子から"1"(5V)を出力したい場合は、"1"を設定する
- ②端子から"0"(0V)を出力したい場合は、"0"を設定する
- ③入力端子は、"0"、"1"のどちらを設定しても構わないが、"1"だと意味があるように思われるため"0"を設定しておく
- ④端子が無い場合は、"0"にする

例えば、ポート 3 の出力端子から次のように電圧を出力したいとします。

| ポート | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|-----|------|------|------|------|------|------|------|------|
| 3   | 5V   | 0V   | 0V   | 0V   | 5V   | 入力   | 0V   | 5V   |

5V 部分を"1"に、0V 部分を"0"に、入力部分を"0"にすれば、ポート 3 に設定する値になります。

| ポート | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|-----|------|------|------|------|------|------|------|------|
| 3   | 1    | 0    | 0    | 0    | 1    | 0    | 0    | 1    |

2 進数で「1000 1001」、16 進数に直すと「0x89」になります。プログラムは、次のように記述します。

```
p3 = 0x89;
```



#### (4) データの入力

入力に設定した端子からデータを入力するときの方法を説明します。

データを入力するときは、「**ポート Pi レジスタ**」(i は 0～6 の数字が入ります)で行い、**レジスタ名は「P0～P6」の 7 個あります**。最後の数字がポート番号です。例えばポート 0 は「P0」となります(レジスタ名は出力と同じです)。

ポートPiレジスタの値を読み込めば、端子の状態が分かります。端子の状態とポートPiレジスタ関係は、次のようになります。

- ①端子に 5V が入力されている場合は、“1”が読み込まれる
- ②端子に 0V が入力されている場合は、“0”が読み込まれる
- ③出力端子を読み込むと、現在出力している状態が読み込まれる
- ④端子が無い場合は、不定(“0”か“1”か分からない状態)が読み込まれる

例えば、ポート 0 には次の電圧が入力されているとします(例なので今回の演習とは関係ありません)。

| ポート | bit7            | bit6            | bit5            | bit4            | bit3 | bit2 | bit1 | bit0 |
|-----|-----------------|-----------------|-----------------|-----------------|------|------|------|------|
| 0   | 出力端子<br>(0V 出力) | 出力端子<br>(0V 出力) | 出力端子<br>(5V 出力) | 出力端子<br>(5V 出力) | 5V   | 5V   | 0V   | 0V   |

ポート0を読み込んだときに入力される値は、5V 部分は“1”が、0V 部分は“0”が読み込まれます。また、出力端子は現在出力している状態が読み込まれます。よって次のようになります。

| ポート | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|-----|------|------|------|------|------|------|------|------|
| 0   | 0    | 0    | 1    | 1    | 1    | 1    | 0    | 0    |

2 進数で「0011 1100」、16 進数に直すと「0x3c」になります。次のようにプログラムすると、変数 c には、0x3c が代入されます。

```
c = p0;    // 変数cには0x3cが代入される
```

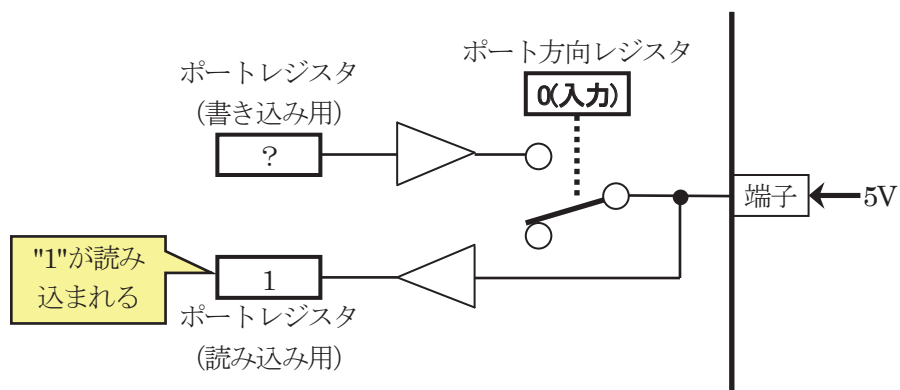
#### (5) 出力設定にするときのポート方向レジスタレジスタ(PD0～PD6)とポートレジスタ(P0～P6)の設定手順

ポート方向レジスタ(PD0～PD6)とポートレジスタ(P0～P6)を設定するとき、どちらを先に設定すればよいのか説明します。マイコンリセット後の初期値を下記に示します。

| レジスタ               | リセット後の初期値        |
|--------------------|------------------|
| ポート方向レジスタ(PD0～PD6) | 0x00(入力設定)       |
| ポートレジスタ(P0～P6)     | 不定(どのような値か分からない) |

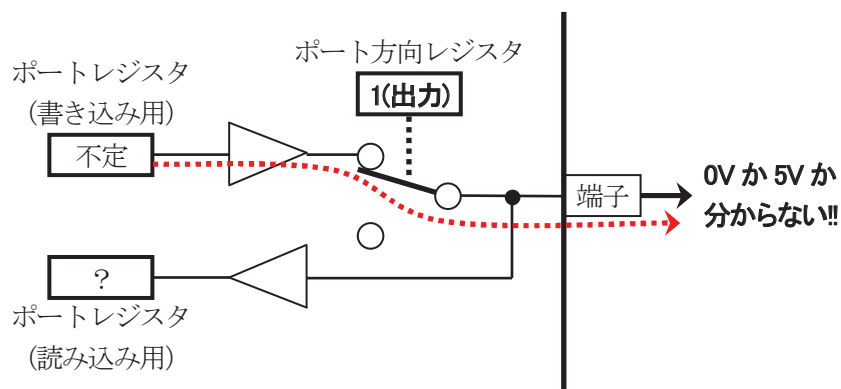
## ●入力設定の場合

リセット後は入力設定なので、ポートレジスタの値を読み込むと端子に入力されている状態が読み込まれます。

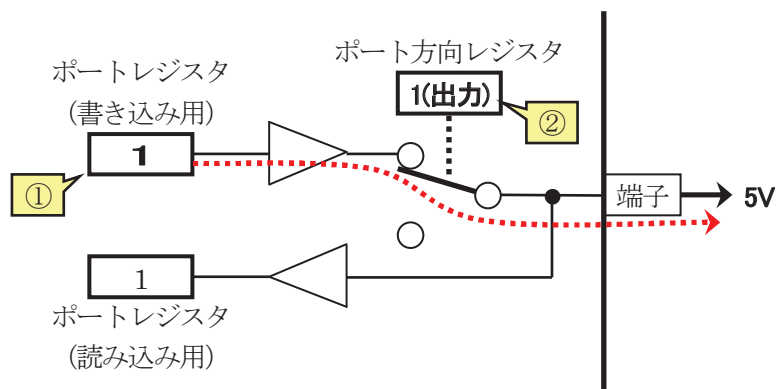


## ●出力設定の場合

リセット後、ポート方向レジスタを"1"にすると、出力設定になり、ポートレジスタの値が出力されます。ただし、**ポートレジスタの値はリセット後、不定です。そのため、出力設定にすると"0"が出力されるか"1"が出力されるか分かりません。**



そのため、出力設定するときは、①ポートレジスタに出力したい値を設定してから、②ポート方向レジスタで出力に設定します。



例えば、マイコンボード上の LED(ポート 1 の bit3~0)は、“0”で点灯、“1”で消灯します。ポート方向レジスタを設定する前に、ポートレジスタに“1”を設定してからポート方向レジスタを設定します。

|             |                          |    |
|-------------|--------------------------|----|
| p1 = 0x0f;  | /* 3-0:LEDは消灯            | */ |
| pd1 = 0xdf; | /* 5:RXD0 4:TXD0 3-0:LED | */ |

## (6) 入力時の電圧について

入力ポートの場合、“1”と判断する電圧(下表の  $V_{IH}$ )と“0”と判断する電圧(下表の  $V_{IL}$ )を、下表に示します。

| 記号                                | 項目       |          |                         |                               | 測定条件                          | 規格値                 |                    |                     | 単位                 |   |
|-----------------------------------|----------|----------|-------------------------|-------------------------------|-------------------------------|---------------------|--------------------|---------------------|--------------------|---|
|                                   |          |          |                         |                               |                               | 最小                  | 標準                 | 最大                  |                    |   |
| V <sub>CC</sub> /AV <sub>CC</sub> | 電源電圧     |          |                         |                               |                               | 1.8                 | —                  | 5.5                 | V                  |   |
| V <sub>SS</sub> /AV <sub>SS</sub> | 電源電圧     |          |                         |                               |                               | —                   | 0                  | —                   | V                  |   |
| V <sub>IH</sub>                   | “H” 入力電圧 | CMOS入力以外 |                         |                               |                               |                     | 0.8V <sub>CC</sub> | —                   | V <sub>CC</sub>    | V |
|                                   |          | CMOS入力   | 入力レベル切り替え機能<br>(I/Oポート) | 入力レベル選択 : 0.35V <sub>CC</sub> | 4.0V ≤ V <sub>CC</sub> ≤ 5.5V | 0.5V <sub>CC</sub>  | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         |                               | 2.7V ≤ V <sub>CC</sub> < 4.0V | 0.55V <sub>CC</sub> | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         |                               | 1.8V ≤ V <sub>CC</sub> < 2.7V | 0.65V <sub>CC</sub> | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         | 入力レベル選択 : 0.5V <sub>CC</sub>  | 4.0V ≤ V <sub>CC</sub> ≤ 5.5V | 0.65V <sub>CC</sub> | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         |                               | 2.7V ≤ V <sub>CC</sub> < 4.0V | 0.7V <sub>CC</sub>  | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         |                               | 1.8V ≤ V <sub>CC</sub> < 2.7V | 0.8V <sub>CC</sub>  | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         | 入力レベル選択 : 0.7V <sub>CC</sub>  | 4.0V ≤ V <sub>CC</sub> ≤ 5.5V | 0.85V <sub>CC</sub> | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         |                               | 2.7V ≤ V <sub>CC</sub> < 4.0V | 0.85V <sub>CC</sub> | —                  | V <sub>CC</sub>     | V                  |   |
|                                   |          |          |                         |                               | 1.8V ≤ V <sub>CC</sub> < 2.7V | 0.85V <sub>CC</sub> | —                  | V <sub>CC</sub>     | V                  |   |
| V <sub>IL</sub>                   | “L” 入力電圧 | CMOS入力以外 |                         |                               |                               |                     | 0                  | —                   | 0.2V <sub>CC</sub> | V |
|                                   |          | CMOS入力   | 入力レベル切り替え機能<br>(I/Oポート) | 入力レベル選択 : 0.35V <sub>CC</sub> | 4.0V ≤ V <sub>CC</sub> ≤ 5.5V | 0                   | —                  | 0.2V <sub>CC</sub>  | V                  |   |
|                                   |          |          |                         |                               | 2.7V ≤ V <sub>CC</sub> < 4.0V | 0                   | —                  | 0.2V <sub>CC</sub>  | V                  |   |
|                                   |          |          |                         |                               | 1.8V ≤ V <sub>CC</sub> < 2.7V | 0                   | —                  | 0.2V <sub>CC</sub>  | V                  |   |
|                                   |          |          |                         | 入力レベル選択 : 0.5V <sub>CC</sub>  | 4.0V ≤ V <sub>CC</sub> ≤ 5.5V | 0                   | —                  | 0.4V <sub>CC</sub>  | V                  |   |
|                                   |          |          |                         |                               | 2.7V ≤ V <sub>CC</sub> < 4.0V | 0                   | —                  | 0.3V <sub>CC</sub>  | V                  |   |
|                                   |          |          |                         |                               | 1.8V ≤ V <sub>CC</sub> < 2.7V | 0                   | —                  | 0.2V <sub>CC</sub>  | V                  |   |
|                                   |          |          |                         | 入力レベル選択 : 0.7V <sub>CC</sub>  | 4.0V ≤ V <sub>CC</sub> ≤ 5.5V | 0                   | —                  | 0.55V <sub>CC</sub> | V                  |   |
|                                   |          |          |                         |                               | 2.7V ≤ V <sub>CC</sub> < 4.0V | 0                   | —                  | 0.45V <sub>CC</sub> | V                  |   |
|                                   |          |          |                         |                               | 1.8V ≤ V <sub>CC</sub> < 2.7V | 0                   | —                  | 0.35V <sub>CC</sub> | V                  |   |

電源電圧(V<sub>CC</sub>)は 5.0V、入力レベル選択は 0.5V<sub>CC</sub> です。よって、下記のように判断します。

“1”と判断する電圧は表より

$$0.65V_{CC} = 0.65 \times 5.0 = 3.25V$$

∴ 3.25V～5.0V の電圧なら“1”と判断

“0”と判断する電圧は表より

$$0.4V_{CC} = 0.4 \times 5.0 = 2.0V$$

∴ 0～2.0V の電圧なら“0”と判断

では、上記の範囲に入っていない 2.0～3.25V の電圧が入力された場合はどうなるのでしょうか。この場合は、“0”と判断するか“1”と判断するか分かりません。ある時は“0”と判断し、ある時は“1”と判断するかもしれません。そのため、この範囲の電圧は加えないようにします。

## (7) 出力時の電圧、電流について

出力ポートの場合、流せる電流を下表に示します。

|                |                              |                              |    |        |       |     |
|----------------|------------------------------|------------------------------|----|--------|-------|-----|
| $I_{OH(sum)}$  | “H” 尖頭総出力電流                  | 全端子の $I_{OH(peak)}$ の総和      | —  | —      | — 160 | mA  |
| $I_{OH(sum)}$  | “H” 平均総出力電流                  | 全端子の $I_{OH(avg)}$ の総和       | —  | —      | — 80  | mA  |
| $I_{OH(peak)}$ | “H” 尖頭出力電流                   | 駆動能力 Low 時                   | —  | —      | — 10  | mA  |
|                |                              | 駆動能力 High 時                  | —  | —      | — 40  | mA  |
| $I_{OH(avg)}$  | “H” 平均出力電流                   | 駆動能力 Low 時                   | —  | —      | — 5   | mA  |
|                |                              | 駆動能力 High 時                  | —  | —      | — 20  | mA  |
| $I_{OL(sum)}$  | “L” 尖頭総出力電流                  | 全端子の $I_{OL(peak)}$ の総和      | —  | —      | 160   | mA  |
| $I_{OL(sum)}$  | “L” 平均総出力電流                  | 全端子の $I_{OL(avg)}$ の総和       | —  | —      | 80    | mA  |
| $I_{OL(peak)}$ | “L” 尖頭出力電流                   | 駆動能力 Low 時                   | —  | —      | 10    | mA  |
|                |                              | 駆動能力 High 時                  | —  | —      | 40    | mA  |
| $I_{OL(avg)}$  | “L” 平均出力電流                   | 駆動能力 Low 時                   | —  | —      | 5     | mA  |
|                |                              | 駆動能力 High 時                  | —  | —      | 20    | mA  |
| $f(XIN)$       | XIN クロック入力発振周波数              | $2.7V \leq V_{CC} \leq 5.5V$ | —  | —      | 20    | MHz |
|                |                              | $1.8V \leq V_{CC} < 2.7V$    | —  | —      | 5     | MHz |
| $f(XCIN)$      | XCIN クロック入力発振周波数             | $1.8V \leq V_{CC} \leq 5.5V$ | —  | 32.768 | 50    | kHz |
| $fOCO40M$      | タイマ RC、タイマ RD のカウントソース (注 3) | $2.7V \leq V_{CC} \leq 5.5V$ | 32 | —      | 40    | MHz |
| $fOCO-F$       | fOCO-F 周波数                   | $2.7V \leq V_{CC} \leq 5.5V$ | —  | —      | 20    | MHz |
|                |                              | $1.8V \leq V_{CC} < 2.7V$    | —  | —      | 5     | MHz |
| —              | システムクロック周波数                  | $2.7V \leq V_{CC} \leq 5.5V$ | —  | —      | 20    | MHz |
|                |                              | $1.8V \leq V_{CC} < 2.7V$    | —  | —      | 5     | MHz |
| $f(BCLK)$      | CPU クロック周波数                  | $2.7V \leq V_{CC} \leq 5.5V$ | —  | —      | 20    | MHz |
|                |                              | $1.8V \leq V_{CC} < 2.7V$    | —  | —      | 5     | MHz |

注1. 指定のない場合は、 $V_{CC} = 1.8V \sim 5.5V$ 、 $Topr = -20^{\circ}C \sim 85^{\circ}C$  (Nバージョン) /  $-40^{\circ}C \sim 85^{\circ}C$  (Dバージョン) です。

注2. 平均出力電流は 100 ms の期間内での平均値です。

注3.  $fOCO40M$  は  $V_{CC} = 2.7V \sim 5.5V$  の範囲で、タイマ RC、タイマ RD のカウントソースとして使用することができます。

“1”(5V)にした端子から流すことのできる電流は、5mA です ( $I_{OH(avg)}$ )。また、マイコン全体では、“1”(5V)にした端子から流すことのできる電流は 80mA です ( $I_{OH(sum)}$ )。

“0”(0V)にした端子から流すことのできる電流は、5mA です ( $I_{OL(avg)}$ )。また、マイコン全体では、“0”(0V)にした端子から流すことのできる電流は 80mA です ( $I_{OL(sum)}$ )。

注1. 指定のない場合は、 $4.2V \leq V_{CC} \leq 5.5V$ 、 $T_{opr} = -20^{\circ}C \sim 85^{\circ}C$ (Nパッケージ)/ $-40^{\circ}C \sim 85^{\circ}C$ (Dパッケージ)、 $f(XIN) = 20MHz$ です。

“0”出力は、0～2V が出力されます。マイコンに接続されている回路は、この電圧で動作するように回路設計する必要があります。

```

35 : void main( void )
36 : {
37 :     unsigned char d;
38 :
39 :     init();                /* 初期化 */
40 :
41 :     while( 1 ) {
42 :         d = p0;
43 :         p6 = d;
44 :     }
45 : }

```