

## 10. I/Oポートの入出力 2(プロジェクト:io2)

### 10.1 概要

本章では、マイコンボード上にあるディップスイッチ(4bit)と LED(4bit)を使って、I/O ポートの入出力を行う方法を説明します。ポートが分かれているため、ビット操作を行って入出力を行います。

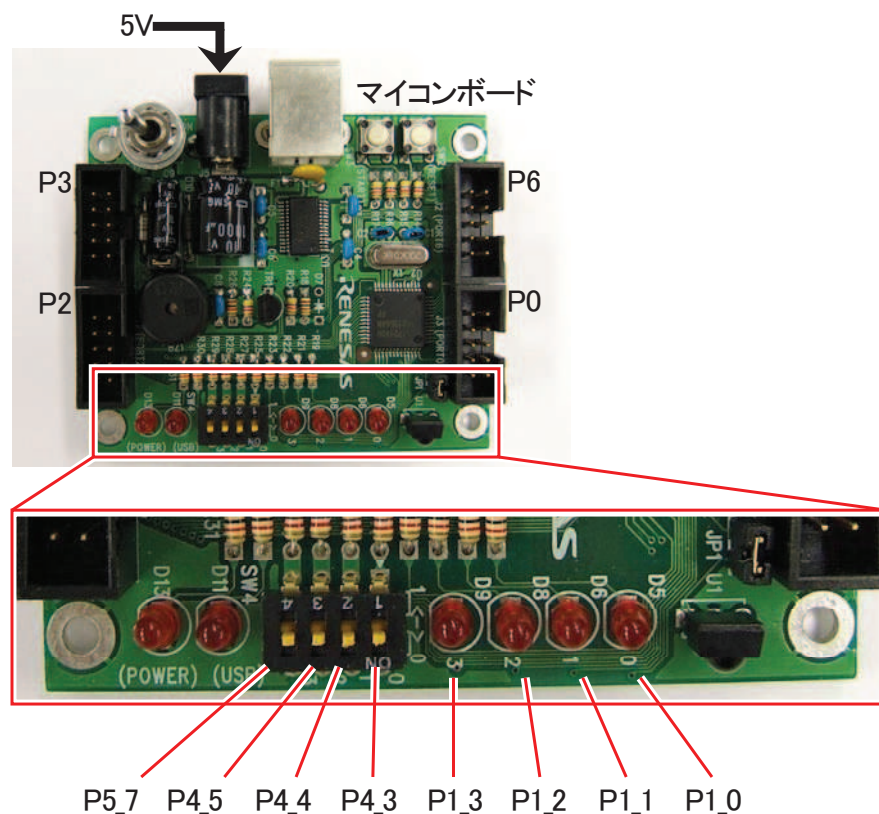
### 10.2 接続

#### ■使用ポート

| マイコンのポート            | 接続内容                 |
|---------------------|----------------------|
| P5_7、P4_5、P4_4、P4_3 | マイコンボード上のディップスイッチです。 |
| P1_3、P1_2、P1_1、P1_0 | マイコンボード上の LED です。    |

#### ■接続例

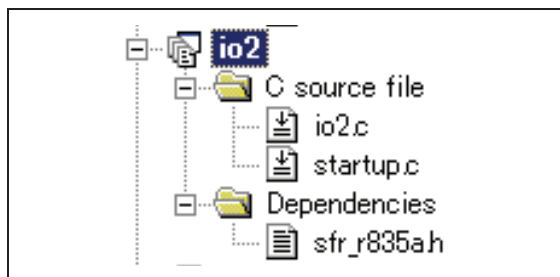
マイコンボードだけで実習できます。



## ■操作方法

マイコンボードのディップスイッチ(SW4)を ON/OFF すると、それに合わせて LED(D9,D8,D6,D5)が点灯／消灯します。

## 10.3 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 | io2.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。  |

## 10.4 プログラム「io2.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アックス株式会社             */
8 :  /******************************************************************/
9 :  /*
10 :  入力：マイコンボードのディップスイッチ(4bit)
11 :  出力：マイコンボードのLED(4bit)
12 :
13 :  マイコンボードのディップスイッチ(4bit)から入力した状態を、
14 :  マイコンボードのLED(4bit)に出力します。
15 :  */
16 :
17 :  /*=====*/
18 :  /* インクルード                                             */
19 :  /*=====*/
20 :  #include "sfr_r835a.h" /* R8C/35A SFRの定義ファイル */
21 :
22 :  /*=====*/
23 :  /* シンボル定義                                             */
24 :  /*=====*/
25 :
26 :  /*=====*/
27 :  /* プロトタイプ宣言                                         */
28 :  /*=====*/
29 :  void init( void );
30 :  unsigned char dipsw_get( void );
31 :  void led_out( unsigned char led );
32 :

```

```

33 : /*****
34 : /* メインプログラム */
35 : /*****
36 : void main( void )
37 : {
38 :     unsigned char d;
39 :
40 :     init();                      /* 初期化 */
41 :
42 :     while( 1 ) {
43 :         d = dipsw_get();
44 :         led_out( d );
45 :     }
46 : }
47 :
48 : /*****
49 : /* R8C/35A スペシャルファンクションレジスタ (SFR) の初期化 */
50 : /*****
51 : void init( void )
52 : {
53 :     int i;
54 :
55 :     /* クロックをXINクロック (20MHz)に変更 */
56 :     prc0 = 1;                    /* プロテクト解除 */
57 :     cm13 = 1;                    /* P4_6, P4_7をXIN-XOUT端子にする*/
58 :     cm05 = 0;                    /* XINクロック発振 */
59 :     for(i=0; i<50; i++ );        /* 安定するまで少し待つ(約10ms) */
60 :     ocd2 = 0;                    /* システムクロックをXINにする */
61 :     prc0 = 0;                    /* プロテクトON */
62 :
63 :     /* ポートの入出力設定 */
64 :     prc2 = 1;                    /* PD0のプロテクト解除 */
65 :     pd0 = 0xe0;                  /* 7-5:LED 4:MicroSW 3-0:Sensor */
66 :     p1 = 0x0f;                   /* 3-0:LEDは消灯 */
67 :     pd1 = 0xdf;                  /* 5:RXD0 4:TXD0 3-0:LED */
68 :     pd2 = 0xfe;                  /* 0:PushSW */
69 :     pd3 = 0xfb;                  /* 4:Buzzer 2:IR */
70 :     pd4 = 0x83;                  /* 7:XOUT 6:XIN 5-3:DIP SW 2:VREF*/
71 :     pd5 = 0x40;                  /* 7:DIP SW */
72 :     pd6 = 0xff;
73 : }
74 :
75 : /*****
76 : /* ディップスイッチ値読み込み */
77 : /* 戻り値 スイッチ値 0~15 */
78 : /*****
79 : unsigned char dipsw_get( void )
80 : {
81 :     unsigned char sw, sw1, sw2;
82 :
83 :     sw1 = (p5>>4) & 0x08;        /* ディップスイッチ読み込み3 */
84 :     sw2 = (p4>>3) & 0x07;        /* ディップスイッチ読み込み2, 1, 0*/
85 :     sw = sw1 | sw2;              /* P5とP4の値を合わせる */
86 :
87 :     return sw;
88 : }
89 :
90 : /*****
91 : /* マイコン部のLED出力 */
92 : /* 引数 スイッチ値 0~15 */
93 : /*****
94 : void led_out( unsigned char led )
95 : {
96 :     unsigned char data;
97 :
98 :     led = ~led;
99 :     led &= 0x0f;
100 :     data = p1 & 0xf0;
101 :     p1 = data | led;
102 : }
103 :
104 : /*****
105 : /* end of file */
106 : /*****

```

## 10.5 プログラムの解説

### 10.5.1 dipsw\_get関数

dipsw\_get 関数は、マイコンボードの 4bit のディップスイッチの値を読み込む関数です。

```

75 :  /***/
76 :  /* ディップスイッチ値読み込み */
77 :  /* 戻り値 スイッチ値 0～15 */
78 :  /***/
79 :  unsigned char dipsw_get( void )
80 :  {
81 :      unsigned char sw, sw1, sw2;
82 :
83 :      sw1 = (p5>>4) & 0x08;          /* ディップスイッチ読み込み3 */
84 :      sw2 = (p4>>3) & 0x07;          /* ディップスイッチ読み込み2, 1, 0*/
85 :      sw = sw1 | sw2;                /* P5とP4の値を合わせる */
86 :
87 :      return sw;
88 :  }
```

マイコンボードには 4bit のディップスイッチが搭載されています。左から順にマイコンの P5\_7、P4\_5、P4\_4、P4\_3 の各 bit に接続されています。

ポートがばらばらなので、ビット演算を使って、次の図のようにビットを移動させて bit3～0 になるようにします。これを行うのが、dipsw\_get 関数です。dipsw\_get 関数を呼ぶと、0～15 の値が返ってきます。ディップスイッチを上にする”1”、下(ON と書いてある側)にすると”0”です。

まず、変数 sw1 にポート 5(P5)の値を読み込みます。

```

83 :      sw1 = (p5>>4) & 0x08;          /* ディップスイッチ読み込み3 */
               ①      ②
```

① ポート 5(P5)の値を 4bit 右シフトします。

② ①の値を 0x08 でマスクします。0x08 は、「0000 1000」なので bit3 のみ有効に、他は強制的に”0”にします。

次に、変数 sw2 にポート 4(P4)の値を読み込みます。

```

84 :      sw2 = (p4>>3) & 0x07;          /* ディップスイッチ読み込み2, 1, 0*/
               ①      ②
```

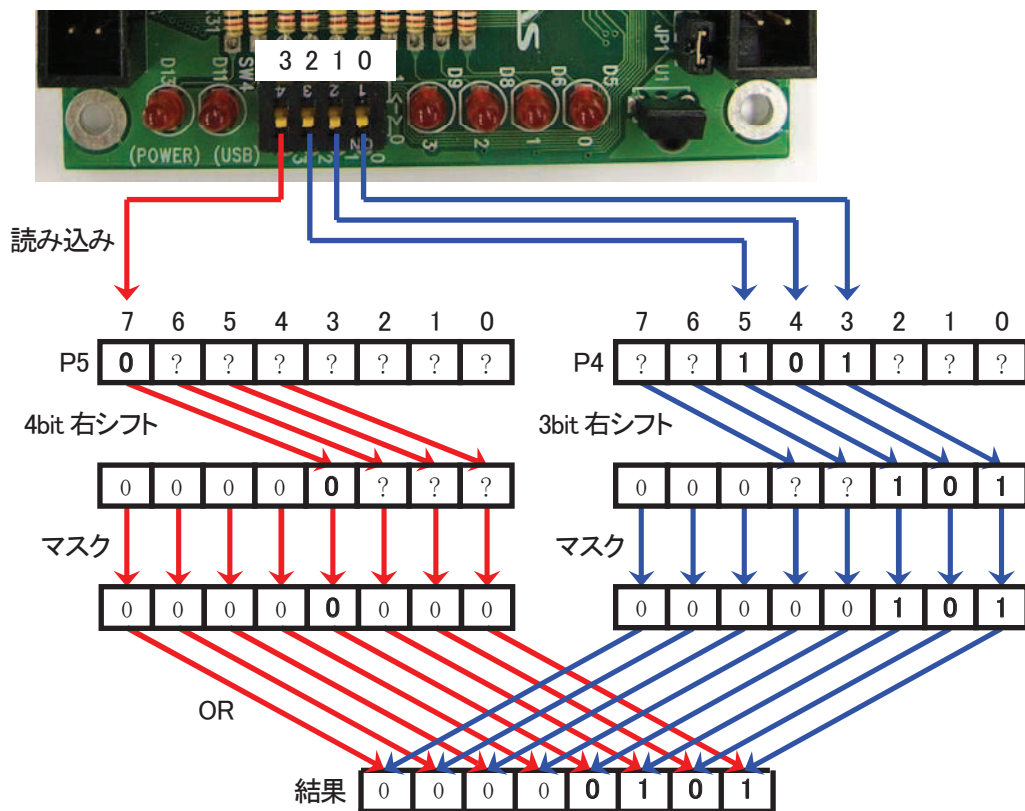
① ポート 4(P4)の値を 3bit 右シフトします。

② ①の値を 0x07 でマスクします。0x07 は、「0000 0111」なので bit2～0 のみ有効に、他は強制的に”0”にします。

最後に、sw1 と sw2 の値を OR 演算で合わせます。

```
85 :      sw = sw1 | sw2;                      /* P5とP4の値を合わせる */
```

ディップスイッチが"0101"(下上下上)のときの、dipsw\_get 関数の動きを下図に示します。



## 10.5.2 led\_out関数

led\_out 関数は、マイコンボードの 4 個の LED を点灯／消灯させる関数です。

```

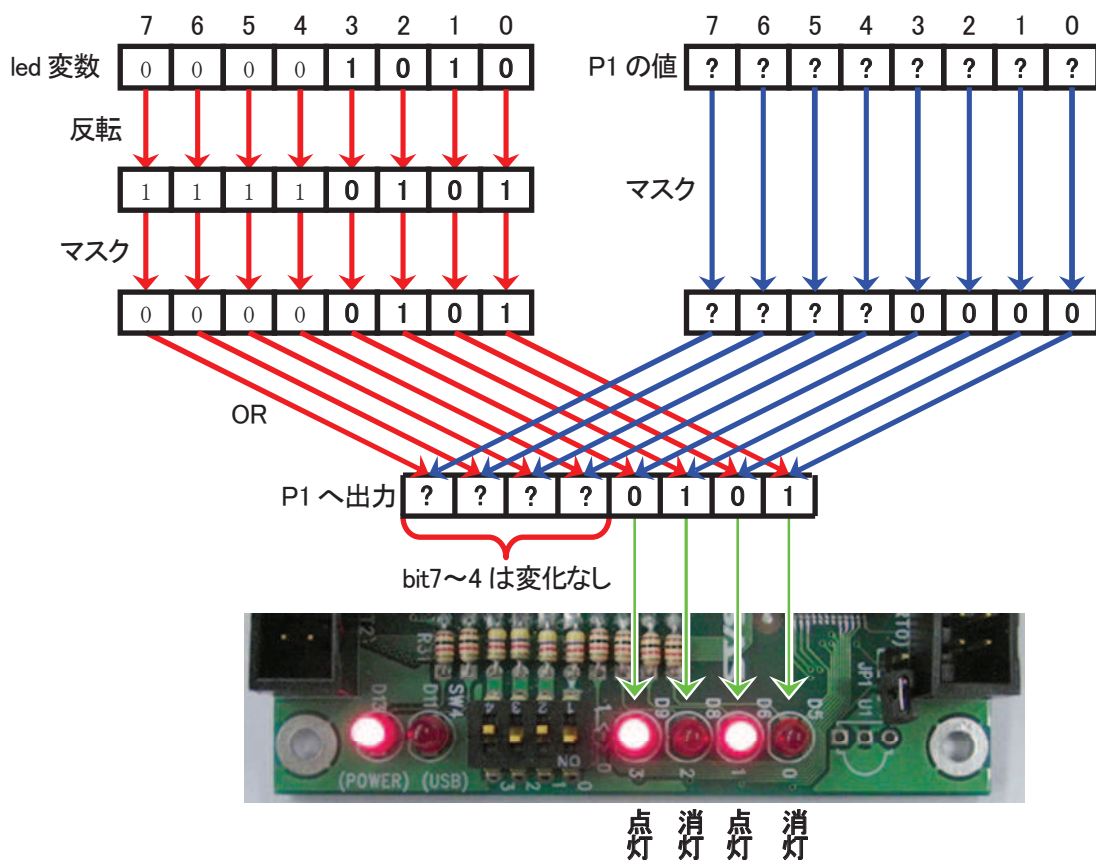
90 :  /***/
91 :  /* マイコン部のLED出力 */
92 :  /* 引数 スイッチ値 0~15 */
93 :  /***/
94 :  void led_out( unsigned char led )
95 :  {
96 :      unsigned char data;
97 :
98 :      led = ~led;
99 :      led &= 0x0f;
100 :      data = p1 & 0xf0;
101 :      p1 = data | led;
102 :  }

```

マイコンボードには 4 個の LED が搭載されています。左から順にマイコンの P1\_3、P1\_2、P1\_1、P1\_0 の各ビットに接続されています。

LED はポート 1 の bit3~0 に接続されていますので、ポート 1 の bit7~4 には影響がないようにします。また、LED は"1"で消灯、"0"で点灯なので、反転させます。これを行うのが、led\_out 関数です。

led\_out 関数に引数 10 を代入したときの動きを、下図に示します。



led 変数に入力されている値は、“0”で消灯、“1”で点灯です。実際の LED は“1”で消灯、“0”で点灯なので、引数の led 変数を反転させます。

```
98 :      led = ~led;
```

LED は bit3～0 に接続されています。関係ない bit7～4 を“0”にしておきます。

```
99 :      led &= 0x0f;
```

次に、LED が繋がっているポート 1(P1)の値を読み込みます。このとき、LED のある bit3～0 は“0”にしておきます。bit7～4 の値は現在のポート 1(P1)の値のままにしておきます。この値を変数 data に代入します。

```
100 :     data = p1 & 0xf0;
```

最後に、data と led の値を OR 演算で合わせ、ポート 1(P1)へ出力します。

```
101 :     p1 = data | led;
```

### 10.5.3 main関数

```
36 : void main( void )
37 : {
38 :     unsigned char d;
39 :
40 :     init();                /* 初期化                */
41 :
42 :     while( 1 ) {
43 :         d = dipsw_get();
44 :         led_out( d );
45 :     }
46 : }
```

main 関数は次のような動作をします。

|      |                                   |
|------|-----------------------------------|
| 43 行 | 変数 d にマイコンボード上のディップスイッチの値を読み込みます。 |
| 44 行 | マイコンボード上の LED に変数 d の値を出力します。     |

結果、マイコンボード上のディップスイッチの値を、マイコンボード上の LED へ出力します。

## 10.6 演習

本演習では、  
ディップスイッチ…マイコンボード上のディップスイッチ  
LED…マイコンボード上の LED  
とする。

- (1) ディップスイッチの値を反転させて、LED へ出力しなさい。
- (2) ディップスイッチの bit0 が"1"なら LED へ"1010"、"0"なら LED へ"0101"を出力するようにしなさい。ただし、bit0 以外は"0"か"1"かは分からないものとする。