

## I/O ポート制御と2進数演算

小型の制御用プロセッサを使う場合、入出力ポート (I/O ポート) の処理では、2進数演算を多用する。ここで、制御プログラミングでよく利用する、演算をまとめる。

演算	意味	例	論理演算子
&	論理積 (AND)	0xAA & 0x0F = 0x0A	&&
	論理和 (OR)	0xAA   0x0F = 0xAF	
^	排他的論理和 (EXOR)	0xAA ^ 0x0F = 0xA5	(ない)
~	否定 (NOT)	~0xAA = 0x55	!

### 1 bit 積・不要な bit を OFF にする

I/O ポートからデータを入出力する場合、byte 単位 (8bit) や word 単位 (16bit) のことが多い。H8 でも 8bit 単位の入出力となる。例えばポート 1 であれば、P10 ~ P17 のデータをまとめて入出力することになる。

例えば、H8 の P11 を 1[bit] のセンサーにつなげてあるとする。

この場合、P11=ON の時に何らかの処理をするのであれば、ポートからの入力レジスタ PDR1 から値を読み取ると、P10,P12 ~ P17 の状態も一緒に読み取ってしまう。

こういった場合、目的外の P10,P12 ~ P17 の値を強制的に 0 にしてから、比較すれば良い。よって bit 積 (&) を使うと便利。

PDR1のP11がセンサーにつながっているとしたら...

× if ( PDR1 == 0x02 )  
P11がONの処理;

P10,P12-P17の値によつては動かない。

```

PDR1 = 0 1 0 1 0 1 0 1 1
&) 0 0 0 0 0 0 0 1 0 = 0x02
-----
0 0 0 0 0 0 0 1 0 --- P11
    
```

```

if ( (PDR1 & 0x02) == 0x02 )
    P11がONの処理;
    
```

### 2 bit 和・目的の bit を ON にする

ポートからのデータ入力とは反対に、I/O ポートにデータを出力する時には、別の注意が必要となる。

例えば、H8 が複数の出力ポートに使われている時、特定の bit だけを ON/OFF したいとする。具体的には、P10 ~ P17 が LED に接続されているとして、P11 だけを ON/OFF と値を変更したいとする。

P11 を OFF にしたい場合は、前説明の bit 積を使えば良い。P11 を ON にしたい場合は、bit 和が便利である。

PDR1が複数の出力ポートにつながっていたら...

目的のポートだけOFFにしたい

$$\begin{array}{r} \text{P17 P16 P15 P14 P13 P12 P11 P10} \\ \text{PDR1} = 10101010 \\ \text{\&) } 11111101 = 0xFD \\ \hline 10101000 \text{ --- P11=0} \end{array}$$

PDR1 = PDR1 & 0xFD ;

C言語流なら

PDR1 &= 0xFD ;

ビット反転を使って書くと

PDR1 &= ~0x02 ;

目的のポートだけONにしたい

$$\begin{array}{r} \text{P17 P16 P15 P14 P13 P12 P11 P10} \\ \text{PDR1} = 01010101 \\ \text{)|} 00000010 = 0x02 \\ \hline 01010111 \text{ --- P11=1} \end{array}$$

PDR1 |= 0x02 ;

目的のポートだけ反転したい

$$\begin{array}{r} \text{P17 P16 P15 P14 P13 P12 P11 P10} \\ \text{PDR1} = 01010101 \\ \text{^)} 00000010 = 0x02 \\ \hline 01010111 \text{ --- P11=NOT(P11)} \end{array}$$

PDR1 ^= 0x02 ;

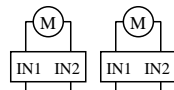
### 3 特定の bit 列を入れ換える

ここまでの制御では、ポートの信号線が 1bit に対応していた。しかし、複数の bit が組み合わせて使われていた場合には、注意が必要である。

例えば、2bit の ON/OFF でモータの回転を制御できる IC を 2 つ使う場合を考える。この場合、一方のモータはそのまま、片方のモータの制御を変更する場合は、「ON にする /OFF にする」という操作が混在することになる。

よって、bit 演算も and,or を取り混ぜた計算式を書く必要がある。

PDR1 に 2 つのモータ制御ICがつながっている場合



IN1	IN2	動き
0	0	停止
0	1	正転
1	0	逆転
1	1	停止

PDR1 = P17 P16 P15 P14 P13 P12 P11 P10

```
#define STOP 0
#define FORWARD 1
#define BACK 2
#define BREAK 3
```

/\* 前進 \*/

PDR1 = (FORWARD << 2) | FORWARD ;

/\* 左モータを逆転 \*/

PDR1 = (PDR1 & 0xF3) | (BACK << 2) ;  
P13, P12を消す P13=1, P12=0

```
PDR1= ----01-- FORWARD<<2
| -----01 FORWARD
= ----0101
```

```
----0101 PDR1
11110011 & 0xF3
----0001 P13,P12を消した
----10-- BACK<<2
----1001 P13,P12をBACKにした
```