AKI-H８（HitachiH8/3048）の使い方

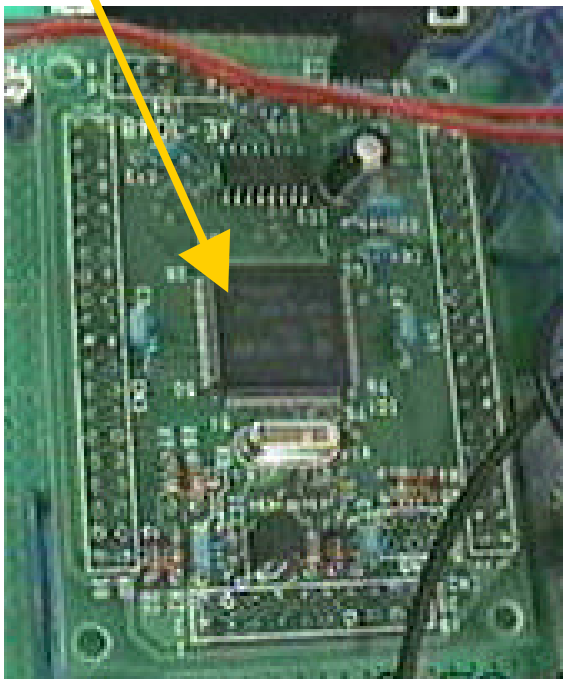
東京工業高等専門学校

情報工学科

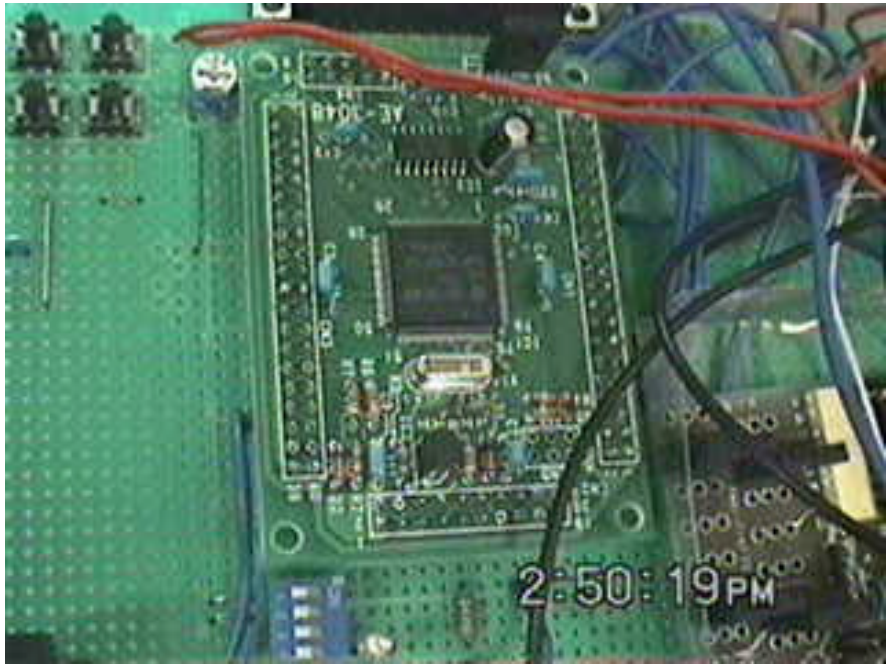
小坂敏文

AKI-H８（HitachiH8/3048）

CPUH8



AKI-H８

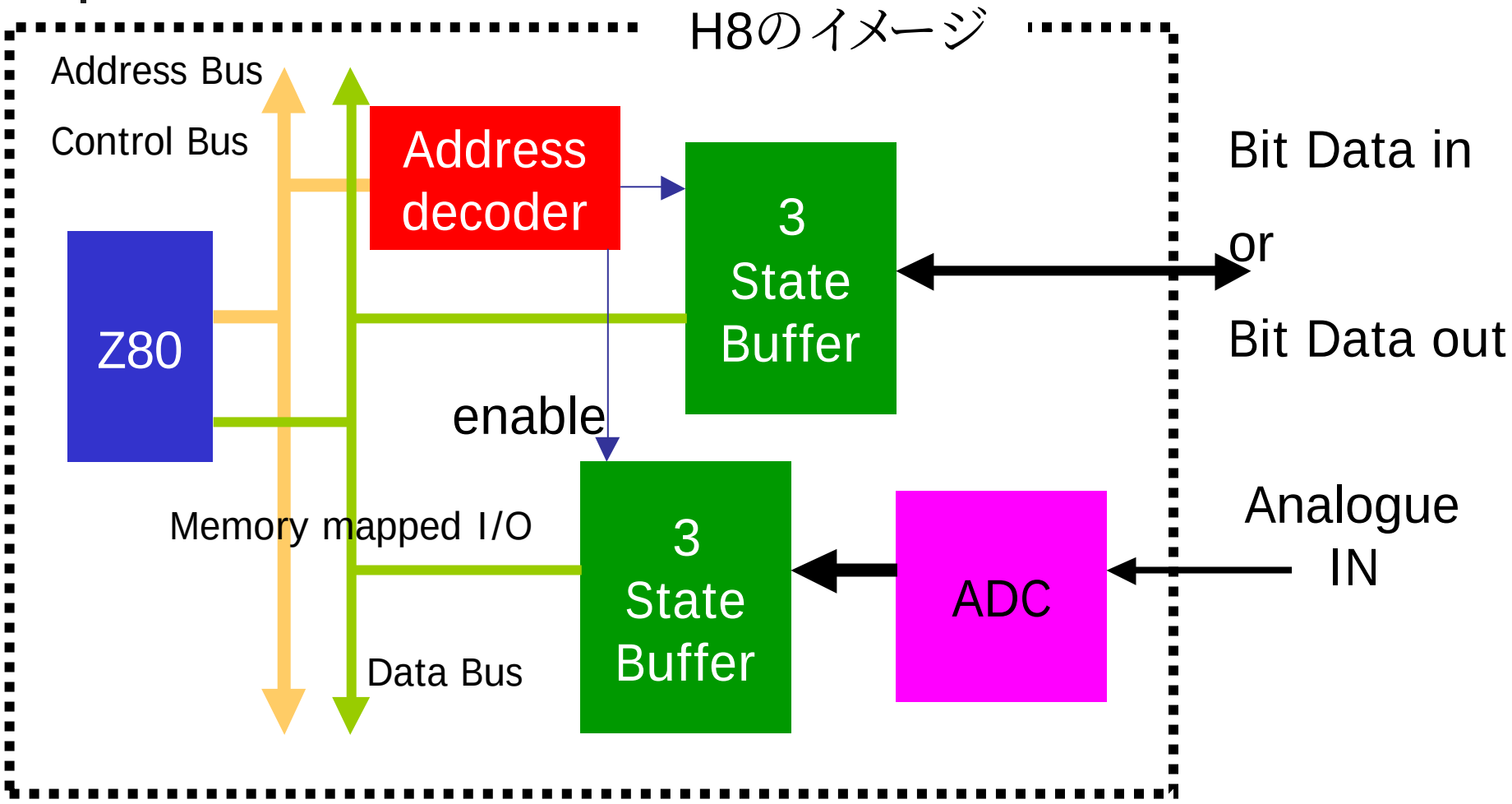


AKIマザーボード

1. 特徴 •用途

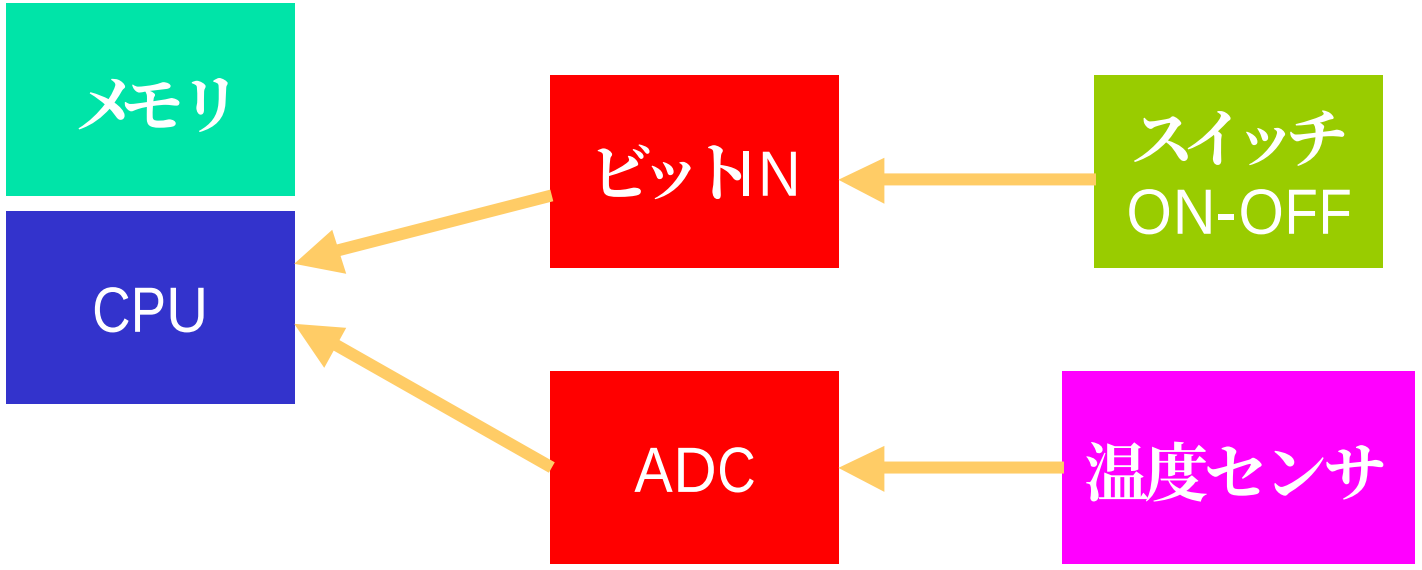
- 開発環境が便利　フラッシュROM内臓で WINパソコンでプログラム開発 •転送可能
- 入出力が簡単　ADC,DAC, カウンタ, RS232C等内蔵
- RS２３２Cでパソコンと通信可能
- 割り込み
- 制御に適している

おまけ Z80との入出力の違い



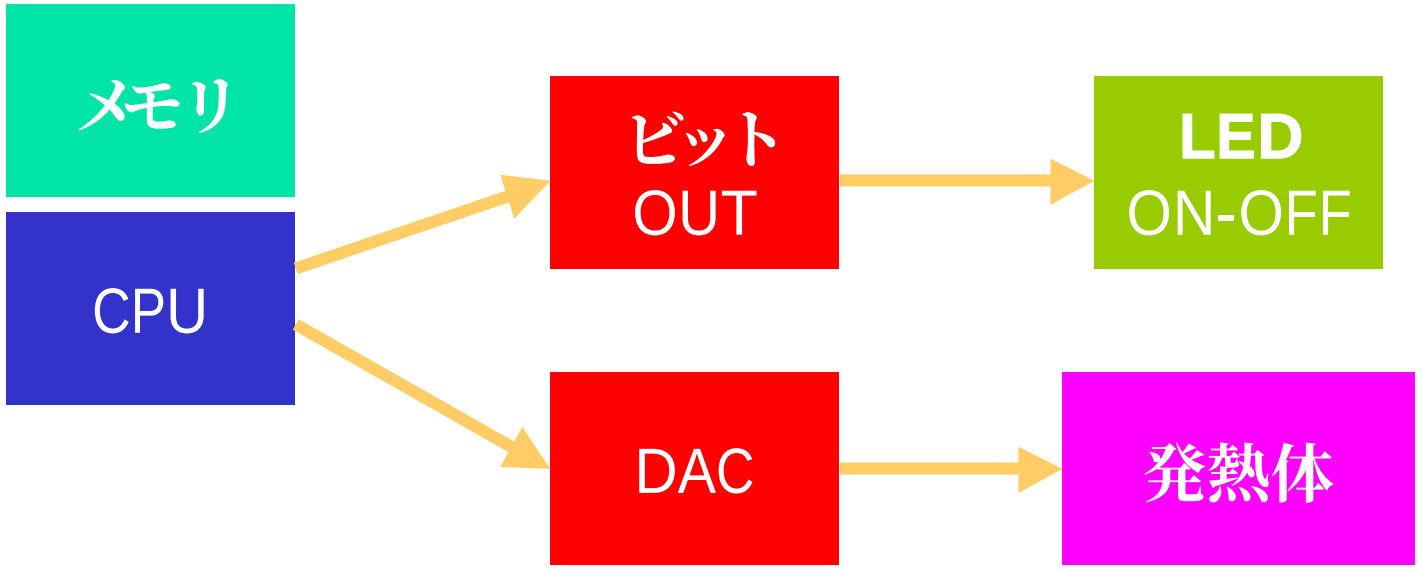
2. CPUが外部の状況を知る

- CPUはメモリ上に置かれた変数について計算を行うだけでなく，外部の状況をビット入力，ADC入力などで知ることができる



3. CPUが外部に働きかける

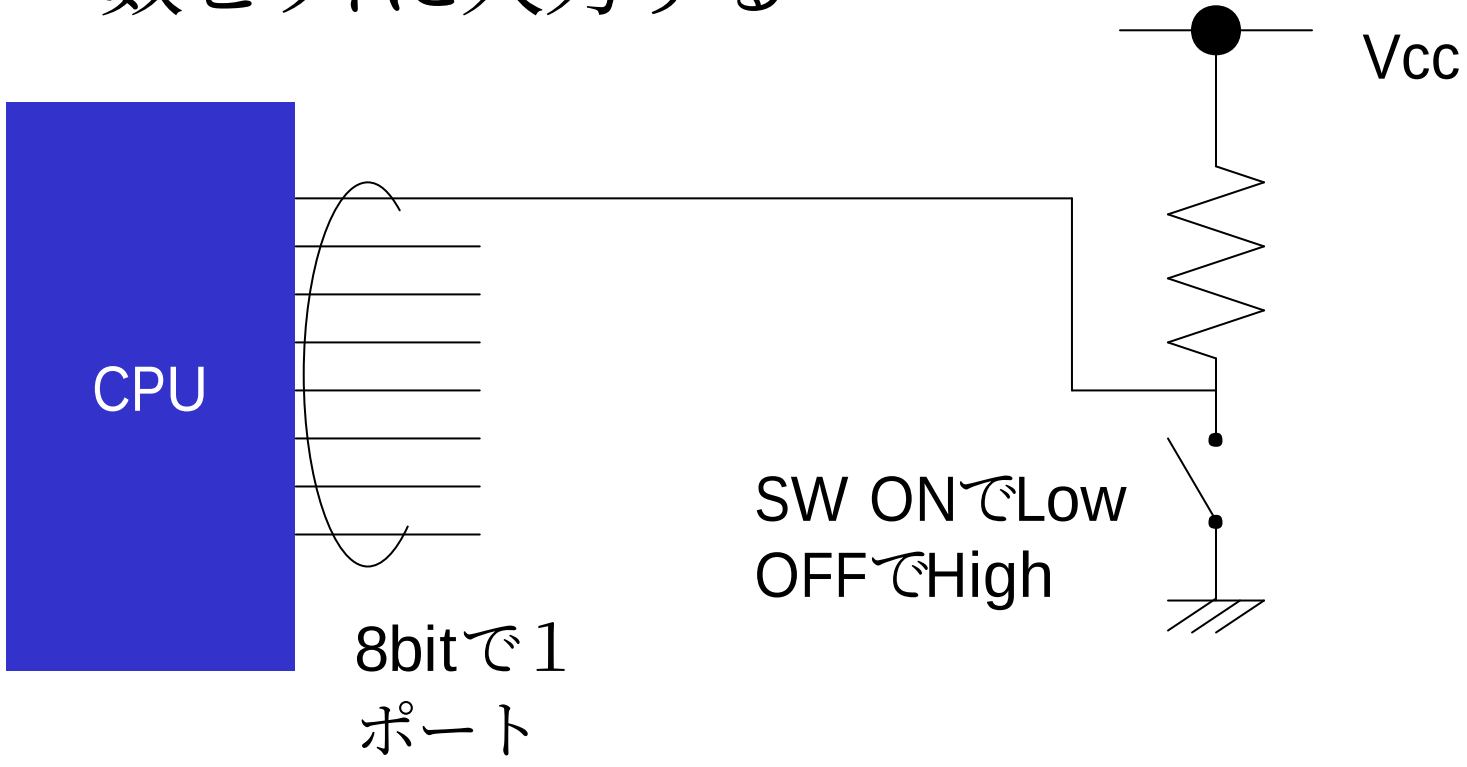
- CPUはメモリ上に置かれた変数について計算を行うだけでなく，外部にをビット出力，DAC出力などで働きかけることができる



4.　CPUが外部の状況を知る-1ビット入力

4．１ CPUが外部の状況を知る-1ビット入力 機械的スイッチ）

- 8ビット幅のポート中の1ビットあるいは数ビットに入力する



4．２ CPUが外部の状況を知る-1ビット入力 光センサ）

- 8ビット幅のポート中の1ビットあるいは数ビットに入力する

4．３ CPUが外部の状況を知る-1ビット入力

- H8CPUでは入力したいポートの特定のビットを入力に初期設定しておくと，このポートはある変数に見える。（memory mapped I/O）
- この変数名がxの時，第0ビットにつながれた入力信号は　x&1で入力にH/Lが確認される。(第1ビットにつながれた入力信号は　x&2)

4．３ CPUが外部の状況を知る-1ビット入力

- H8CPUでは入力したいポートの特定のビットを入力に初期設定しておくと，このポートはある変数に見える。（memory mapped I/O）
- この変数名がxの時，第0ビットにつながれた入力信号は　x&1で入力にH/Lが確認される。(第1ビットにつながれた入力信号は　x&2)

4．3 CPUが外部の状況を知る-1ビット入力プログラム（1）

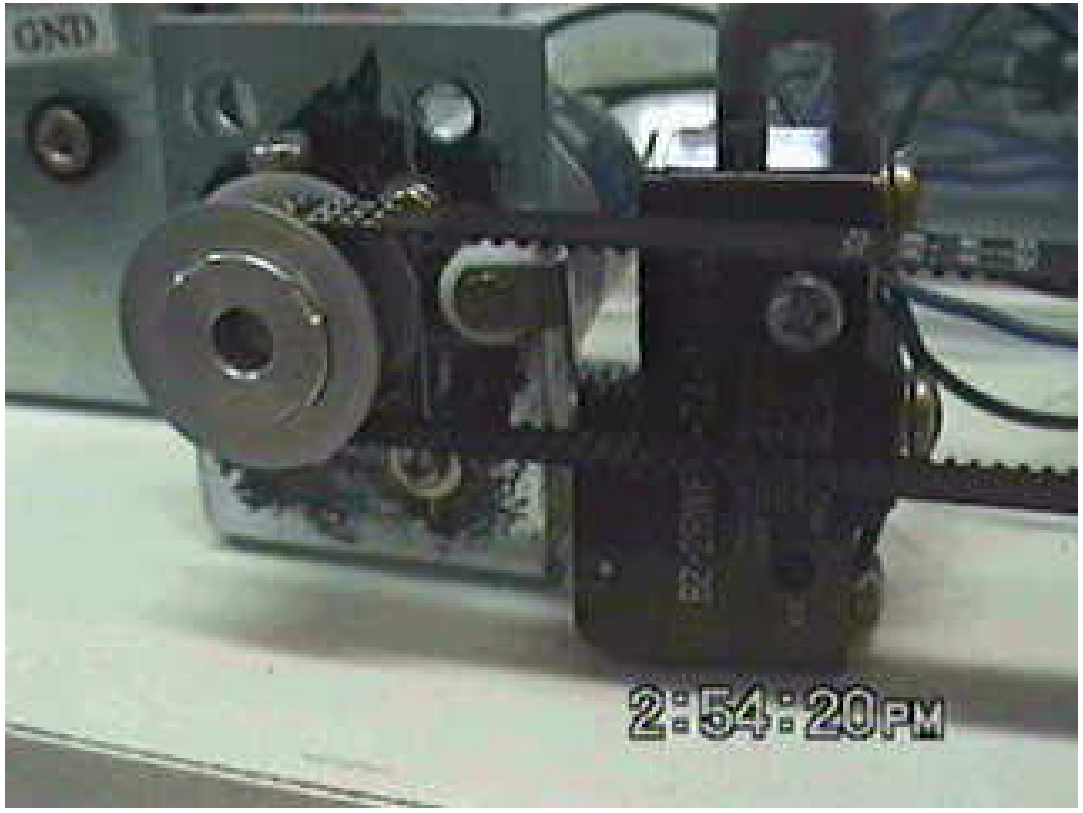
```
/*****************スイッチのON-OFFセンス************************/
#include "h8-00.h"
#include "rsw_p4b2.h"
main()
{
        int x;
        ddr_init();
        sci1_init();        /*シリアルポート初期化*/
        rsw_init();
        while(1) {
                x=rsw();
                sci1_putDecInt(x);   /*RS232Cを通じてホストへ値を送る*/
                sci1_rtlf();
        }
}
```

4．３ CPUが外部の状況を知る-1ビット入力プログラム（2）

```
void rsw_init(void)
{
        P4DDR &= 0xfb; /*P4-2は入力*/
        P4.DDR = P4DDR;
        P4.PCR.BYTE |= 0x04; /*P4-2はプルアップ*/
}

int rsw(void)
/*リミットsw の状態を調べる　押されていたら1、そうでなかったら0を返す*/
{
        int tmp,ret=0;
        tmp=P4.DR.BYTE;
        if (tmp&0x04) ret=1;
        return ret;
}
```

4．３ CPUが外部の状況を知る-1ビット入力プログラム（3）



- 機械的スイッチ

4．４　1ビット入力の用途

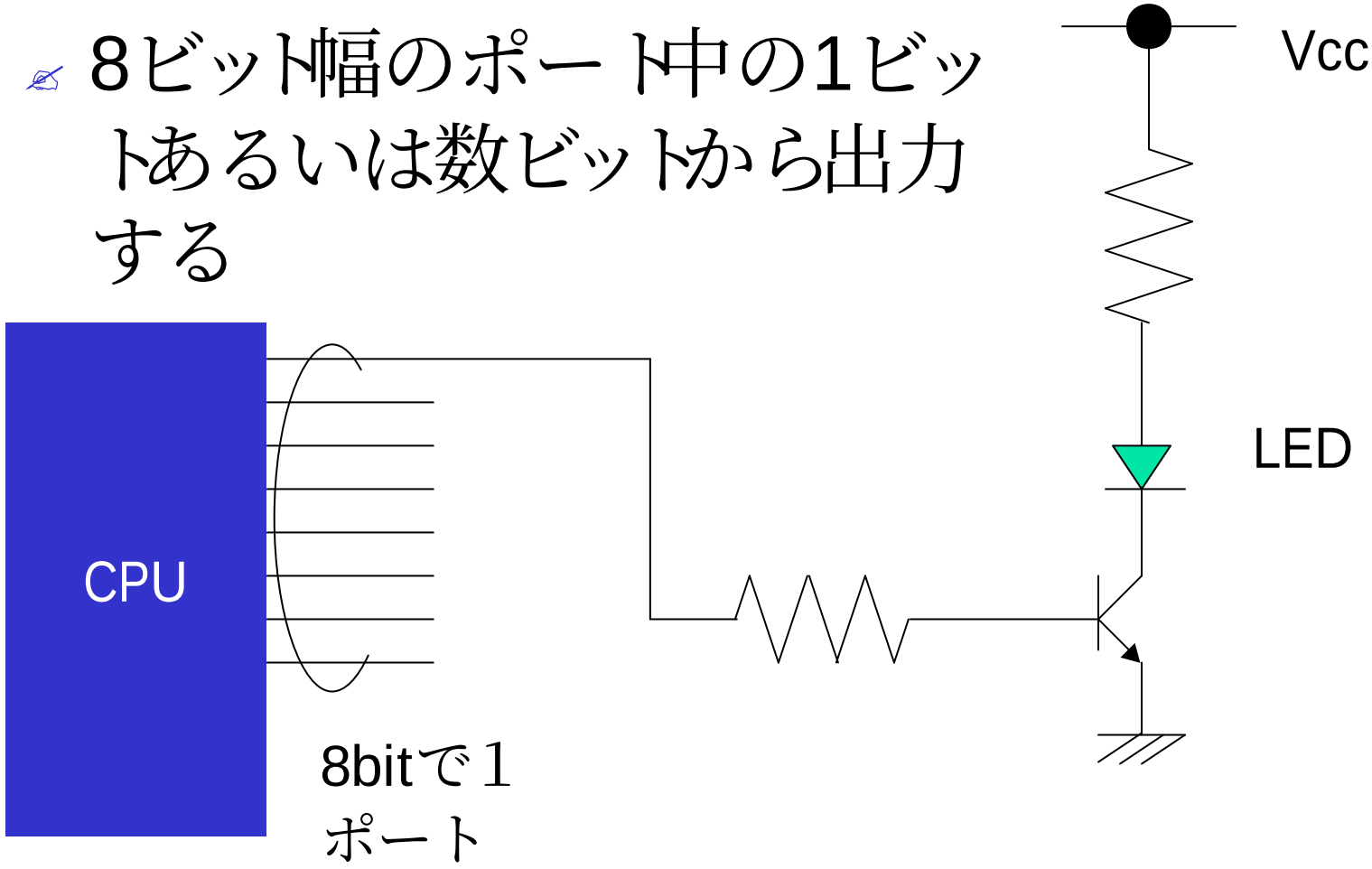
- 機械的スイッチのON-OFFの状態検査
- 光センサの受光状態が電圧High-Low出力の場合の状態検査
- 磁気センサの状態が電圧High-Low出力の場合の状態検査

光センサには透過型，反射型がある）

5.　CPUが外部に働きかける-1ビット出力

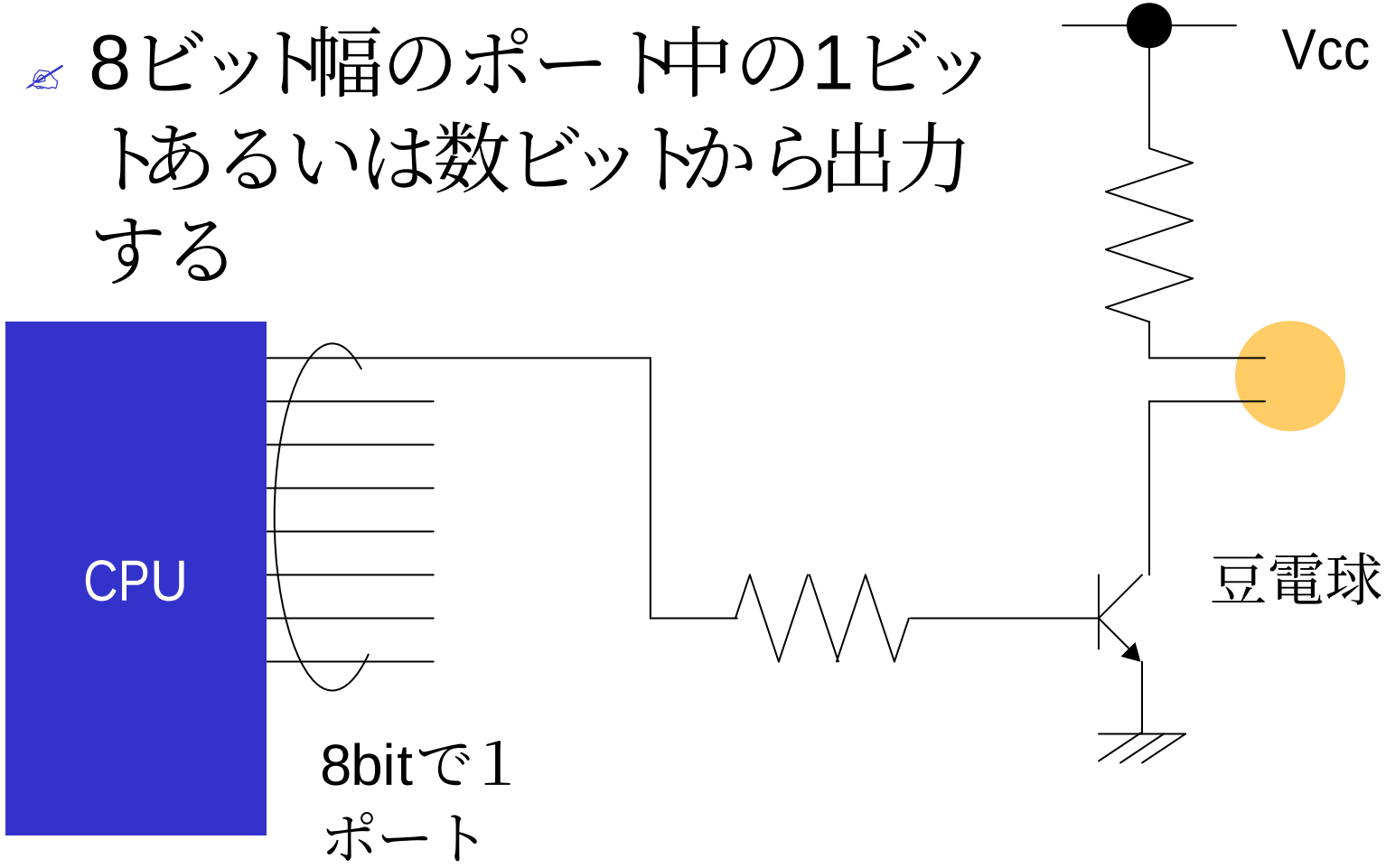
5．１ CPUが外部に働きかける-1ビット出力（LED）

- 8ビット幅のポート中の1ビットあるいは数ビットから出力する



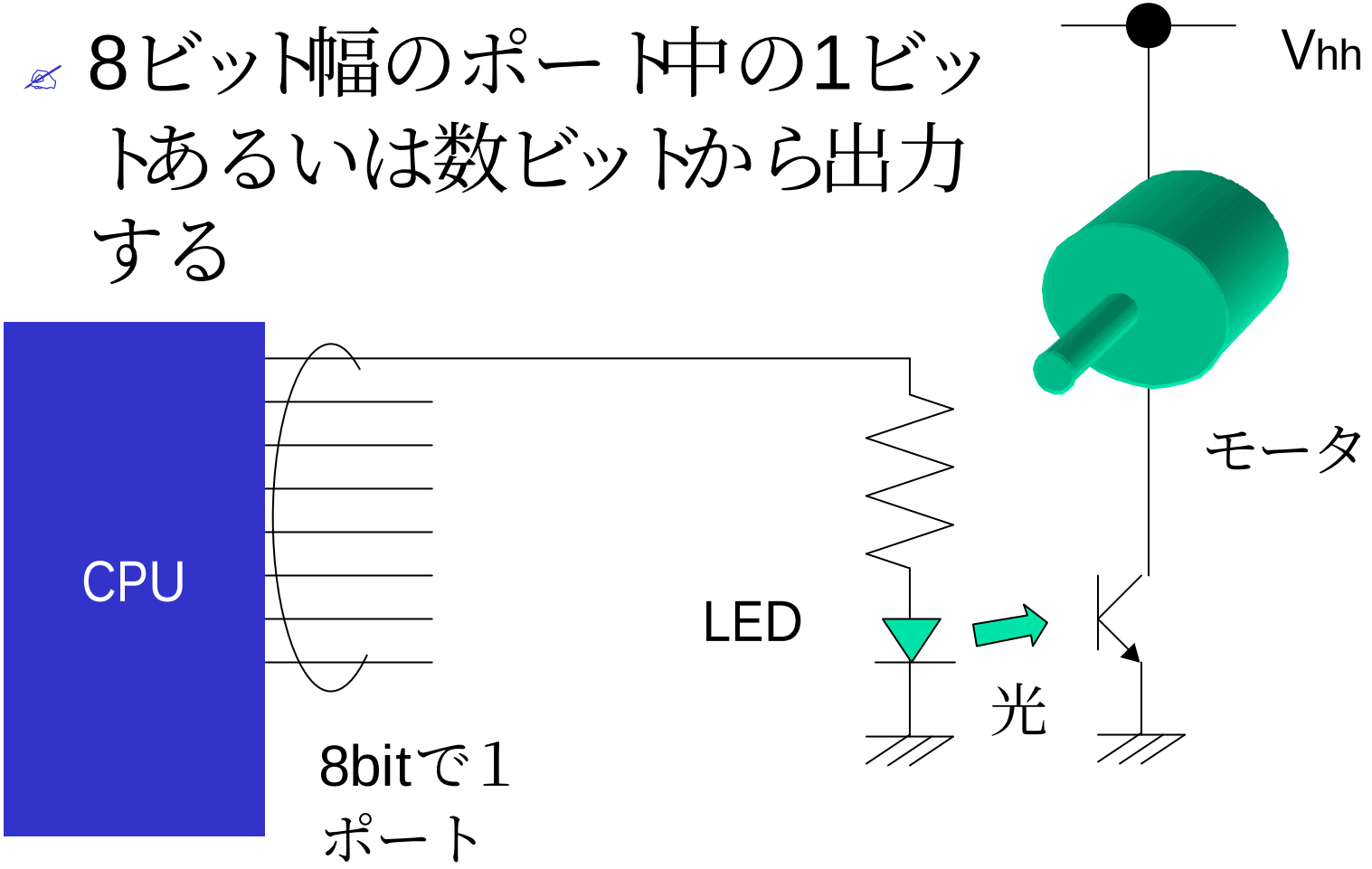
5．２ CPUが外部に働きかける-1ビット出力（豆電球）

- 8ビット幅のポート中の1ビットあるいは数ビットから出力する



5．３ CPUが外部に働きかける-1ビット出力（モータ）

- 8ビット幅のポート中の1ビットあるいは数ビットから出力する



5．４ CPUが外部に働きかける-1ビット出力

- H8CPUでは出力したいポートの特定ののビットを出力に初期設定しておくと，このポートはある変数に見える。（memory mapped I/O）
- この変数名がxの時，第0ビットにHを出したい時は x|=1, Lを出したい時は x&=0xfe とすればよい。(第1ビットにHを出したい時は x|=2, Lを出したい時は x&=0xfd とすればよい。)

5．４ CPUが外部に働きかける-1ビット出力 (1)

```
/**************12V単純出力テスト*****************/
#include "h8-00.h"
#include "led12_p5b2.h"
main()
{
    int x;
    ddr_init();
    led12_init();
    while(1) {
        int i,j;
        led12_on();
        for(i=0;i<30000;i++)for(j=0;j<30;j++);
        led12_off();
        for(i=0;i<30000;i++)for(j=0;j<30;j++);
    }
}
```

5．４ CPUが外部に働きかける-1ビット出力 (2)

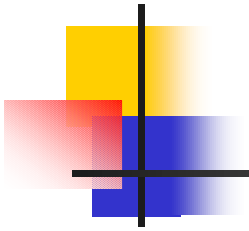
```
void led12_init()
{
    P5DDR |=  0x4;
    P5.DDR=P5DDR;
}

void led12_on(void)    /* LET LED ON */
{
    P5.DR.BYTE |= 4; /* on */
}

void led12_off(void) /* LET LED OFF */
{
    P5.DR.BYTE &= 0xfb; /* off */
}
```

5．４ CPUが外部に働きかける-1ビット出力 (3)

- 豆電球点灯

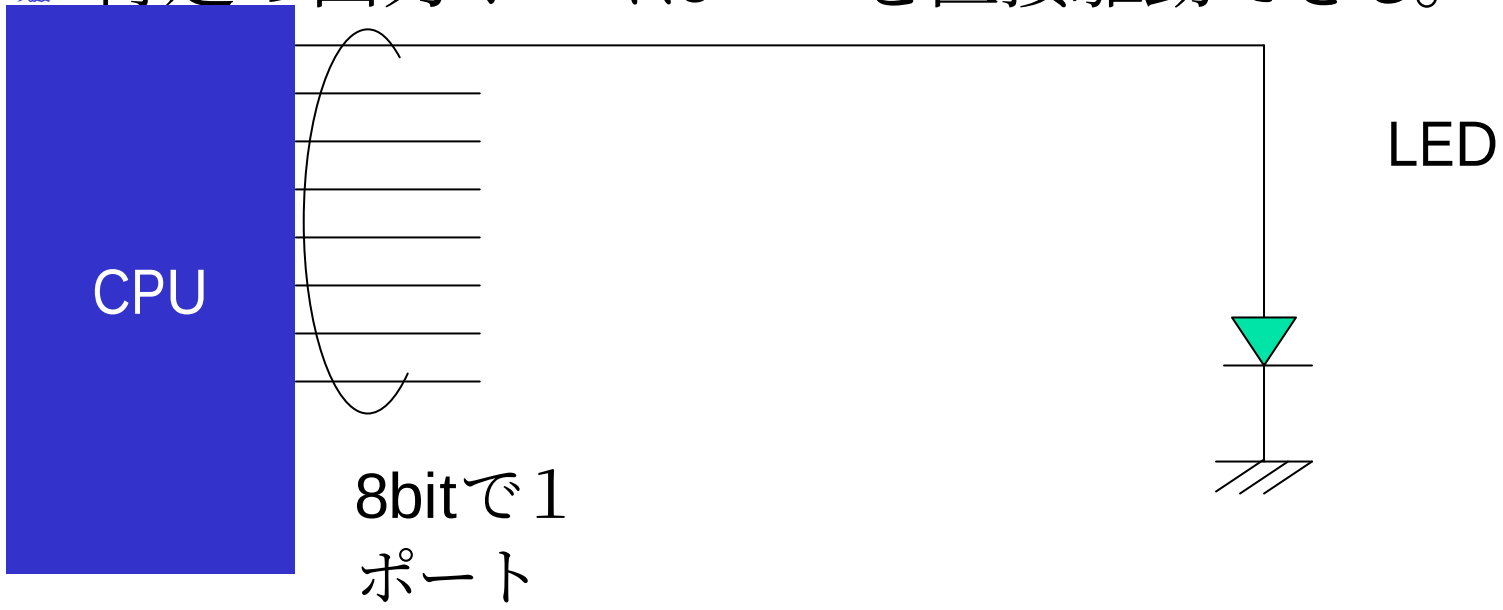


5．５　1ビット出力の用途

- LED点灯
- 豆電球点灯
- モータ駆動
- リレー駆動でかなり大電流のものも可
- FET駆動でかなり大電流のものも可
- H型回路でモータの正逆転

6. H8CPUのビット入出力

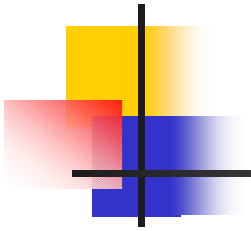
- H8CPUはあるポートを入出力に利用する場合，ビットごとに入力用途，出力用途に利用できる。初期設定することで可能
- 特定の出力ポートはLEDを直接駆動できる。



7.　CPUが外部の状況を知る-ADコンバータ

7．１ CPUが外部の状況を知る-ADコンバータ

- CPUのあるピン（足）にはアナログ入力を受け付けて，AD変換してくれるところがある。
- ４ch分（８ch分）のADCがある。
- 簡単なプログラムで設定後，特定の変数の変換結果がしまわれるようになる。利用したい時は，この変数内の値を使えばよい
- （0V - 5V）→（0 – 1024(分解能10bit)）



7. 2 ADコンバータの用途

- ポテンショメータ 角度-電圧変換器）
- 温度センサ電圧出力
- 圧力センサ電圧出力
- リニア出力赤外線センサ電圧出力

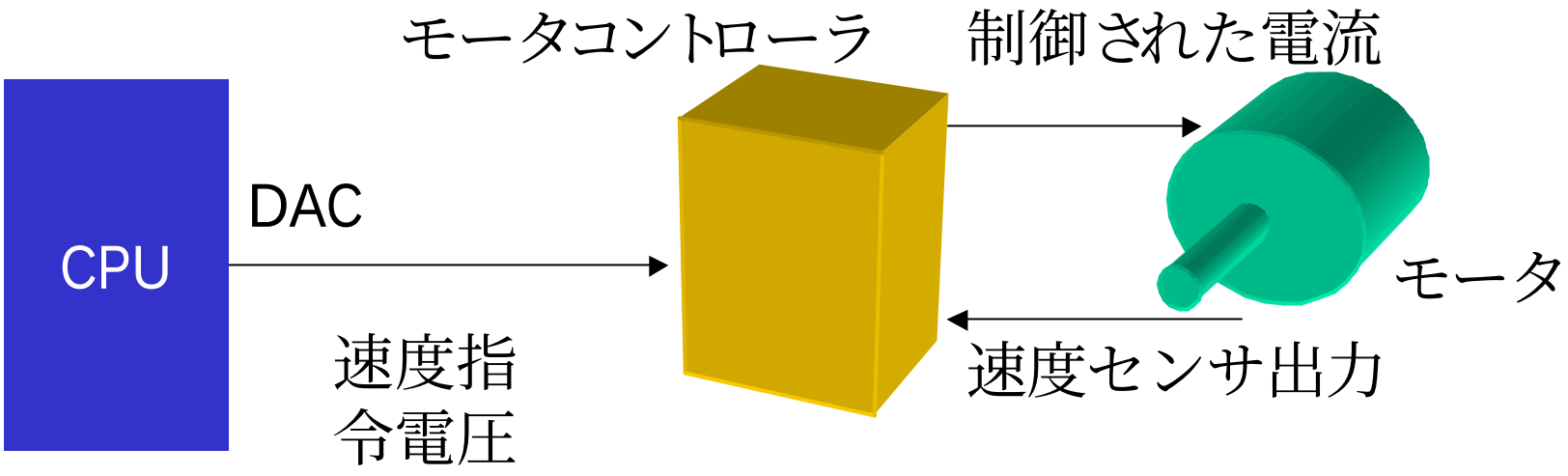
8.　CPUが外部の状況を知る-ADコンバータ

8．１ CPUが外部に働きかける-DAコンバータ

- 特定の変数値をDA変換し対応した電圧をCPUのあるピン（足）に出力する
- ４ch分のDACがある。
- 簡単なプログラムで設定後，特定の変数をDA変換するようになる。（0 – 255(分解能8bit)）→　（0V - 5V）

8. 2 DAコンバータの用途

- DCサーボモータのコントローラでは速度指令値を電圧で受け取るものがある。

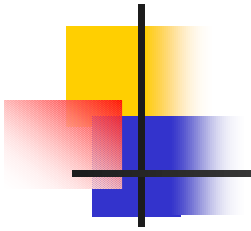


9. H8CPUのADコンバータとDAコンバータ

```
/*DACからの出力をADCの入力へ  */
#include "h8-00.h"
main()
{
    int x;
    unsigned int y;
    ddr_init();  sci1_init();
    da_init();
    while(1) {
        x=getint("DA value(0..255) =");
        da(0,x&0xff);
        y=ad(0);
        sci1_rtlf();
        sci1_putDecInt((int)y);
        sci1_rtlf();
    }
}
```

結果

DA値	出力電圧		AD値
0	0.08V	→	4
64	1.27V	→	256
128	2.53V	→	508
192	3.78V	→	765
255	5.02V	→	1017



10.　CPUが軸の回転角を知る

- H8CPUはロータリエンコーダのパルス出力を数えるカウンタ回路を内蔵し，直接軸の回転角度を知ることができる。

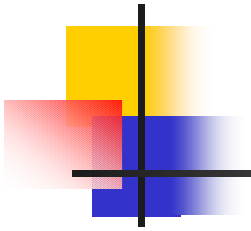
10．１ ロータリエンコーダのパルス出力を数える

```
/***** H8カウンタの取得とシリアルポートへの出力　*****/
#include "h8-00.h"
main()
{
    int x;
    ddr_init(); sci1_init();          /*シリアルポート初期化*/
    h8counter_init();   /*H8位相内部カウンタ初期化（TUchanel2使用）*/
    while(1) {
        x=geth8counter();   /*位相カウンタの読み出し*/
        sci1_putDecInt(x);  /*シリアルポートへ送信*/
        sci1_rtlf();        /*シリアルポートへ改行送信*/
        if (0<sci1_chkandget()) h8counter_init();
            /*シリアルポートから入力があったら、位相カウンタクリア*/
    }
}
```

10．２ ロータリエンコーダのパルス出力を数える

```
/*  H8 INNER COUNTER INITIALIZE AND CLEAR*/
void h8counter_init(void)
/*ITU chanel2を使用*/
{
    ITU.TMDR.BYTE |= 0x40; /*MDF->1 ITU2:counter mode*/
    ITU.TSTR.BIT.STR2=1;
    ITU2.TCNT=0;
}

/* READ H8 INNER COUNTER*/
int geth8counter(void)
{
    return ITU2.TCNT;
}
```



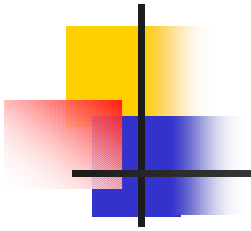
11.　CPUによるモータ駆動

- H8CPUはPWMパルス出力可能である。これによりモータ指令値出力が直接できる。

CPUによるモータ駆動

```
/**  モータへのPWM出力  **/
#include "h8-00.h"
#include "pwm3cntl.h"
main()
{
    int x;
    ddr_init();
    sci1_init();          /*シリアルポート初期化*/
    initPWM3(4096);       /*PWM出力初期化（TUchanel3使用）*/
    while(1) {
        x=getint("PWM value(-4096..4096 otherwize break:on) =");
        sci1_putString("¥n"); sci1_putDecInt(x);sci1_rtlf();
        if (-4096<=x&&x<=4096) outPWM3(x);
        else outPWM3_break(1);
    }
}
```

12.　プログラミング＋ダウンロード



- WINDOWS上でプログラミング
- クロスコンパイラでコンパイル＋リンク
- HEXフォーマットファイル作成
- 転送アプリでPCからAKIH 8へダウンロード
- AKIH8で実行