

# R 8-マイコン実習（4）

## PWM実習

### R 8-マイコン実習（PWM）

#### 目的

マイコン制御の基本技術であるPWM制御について理解を深める。

## R8-マイコン実習（PWMとは）

### 関係知識

**PWM(Pulse Width Modulation)**とは、半導体を使った電力を制御する方式の1つです。オンとオフの繰り返しスイッチングを行い、出力される電力を制御します。

一定電圧の入力から、パルス列のオンとオフの一定周期を作り、オンの時間幅を変化させる電力制御方式を **PWM** と呼びます。

## R8-マイコン実習（そもそも電力制御）

### 関係知識

電力とは、電圧・電流・時間の積なので

$$\mathbf{P = V \cdot I \cdot t} \quad [\text{w} \cdot \text{s}] \text{で表される}$$

つまり

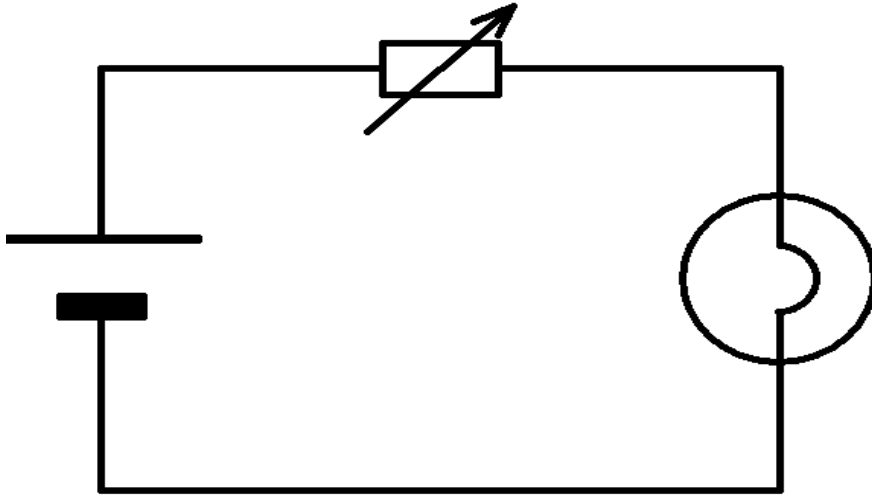
- ・ 電圧・電流を制御する
- ・ 時間を制御する

この二つの方法に分けられる

## R8-マイコン実習(そもそも電力制御)

### 関係知識

- ・電圧・電流を制御する



## R8-マイコン実習(そもそも電力制御)

### 関係知識

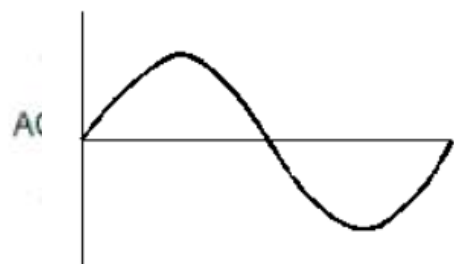
- ・通電時間を制御する
  - ・位相角制御
  - ・PWM

# R8-マイコン実習(そもそも電力制御)

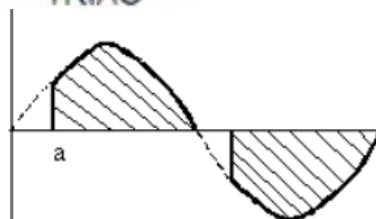
## 関係知識

トリアック制御

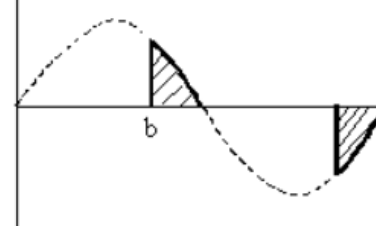
(a)元の正弦波



TRIAC



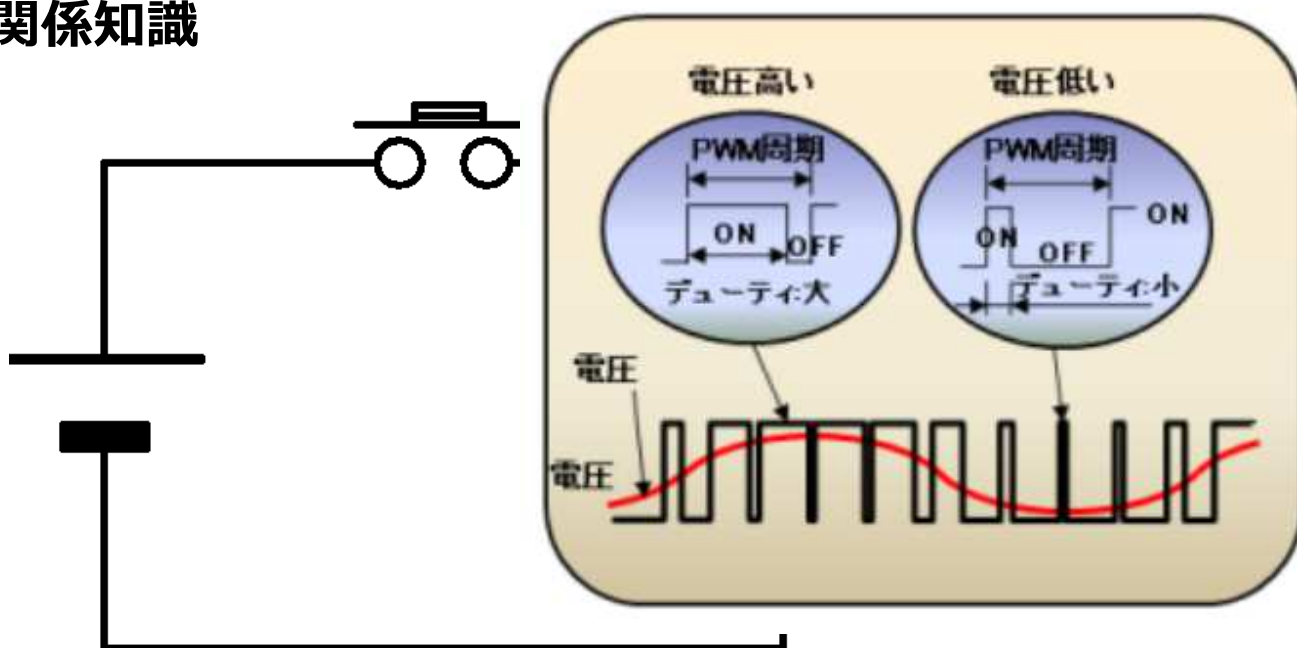
(c)遅めのOn(電力小)



AMP

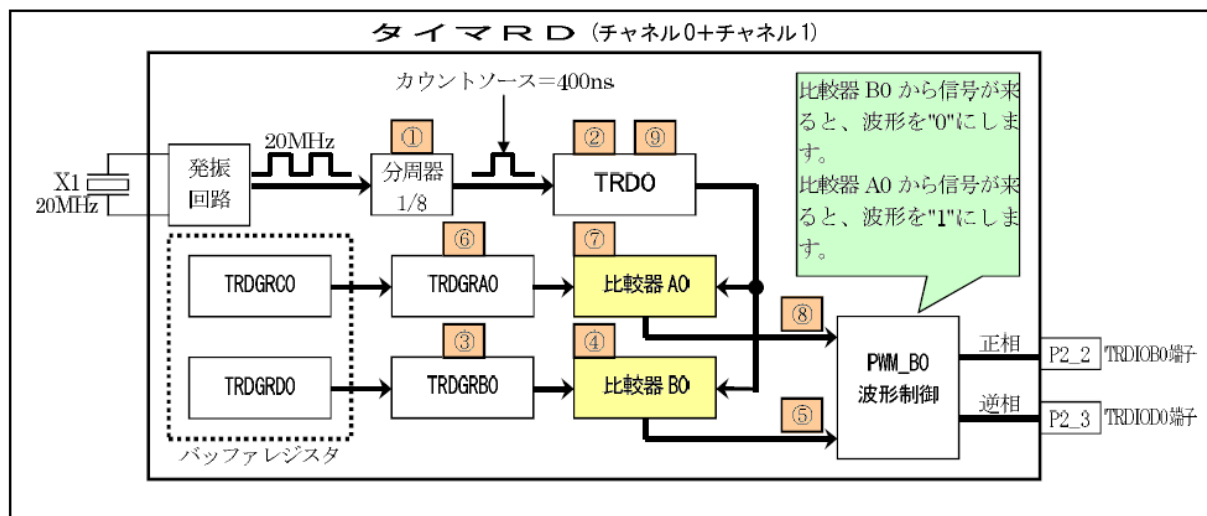
# R8-マイコン実習(PWMとは)

## 関係知識



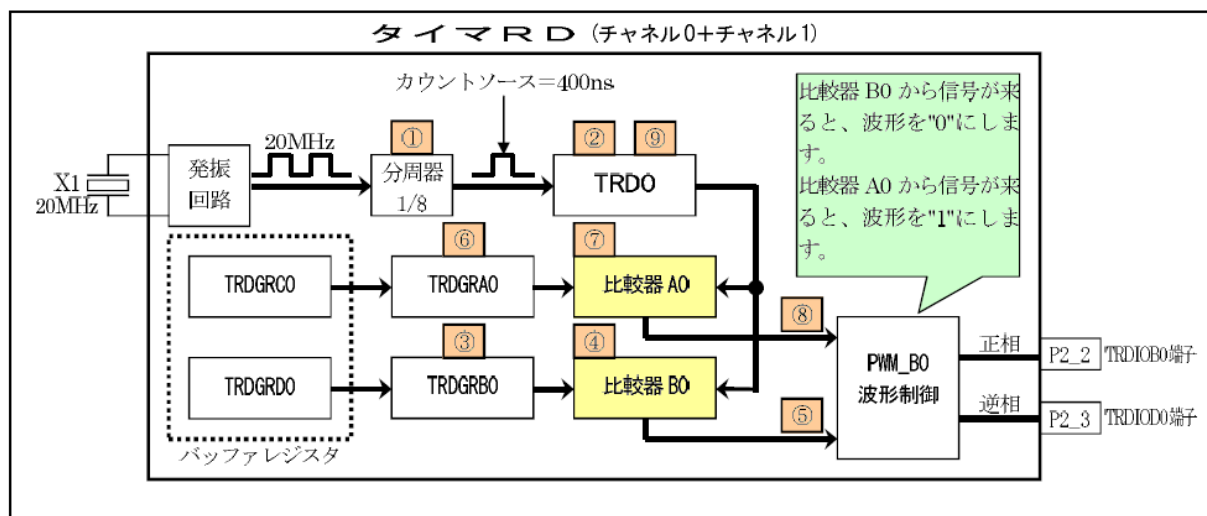
# R8-マイコン実習(タイマーRDの利用)

## 関係知識



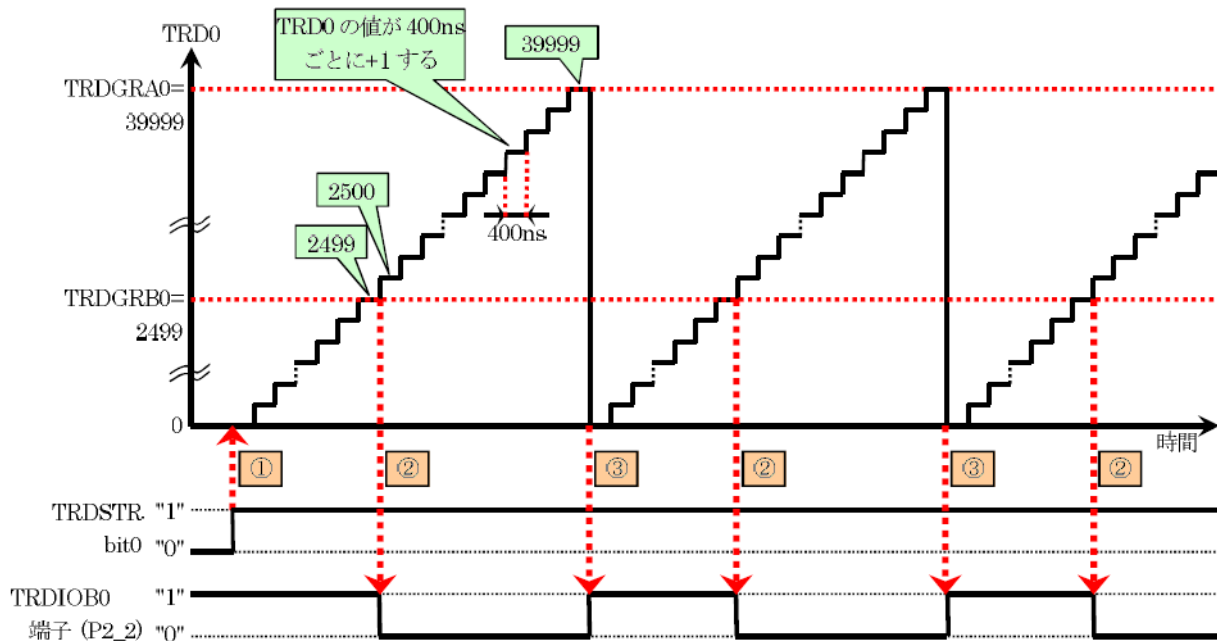
# R8-マイコン実習(タイマーRDの利用)

## 関係知識



# R8-マイコン実習(PWMとは)

## 関係知識



# R8-マイコン実習(R8Cで実現)

## 関係知識

### ・ R8Cの関係レジスタについて

/\* タイマRD リセット同期PWMモードの設定\*/

trdfcr = 0x01;

/\* リセット同期PWMモードに設定 \*/

trdmr = 0xf0;

/\* バッファレジスタ設定 \*/

trdoer1 = 0x01;

/\* 出力端子の選択 \*/

trdpsr0 = 0x68;

/\* TRDIOB0,C0,D0端子設定 \*/

trdpsr1 = 0x55;

/\* TRDIOA1,B1,C1,D1端子設定 \*/

trdcr0 = 0x23;

/\* ソースカウンタの選択:f8 \*/

trdgra0 = trdgrc0 = 39999;

/\* 周期 \*/

trdgrb0 = trdgrd0 = 0;

/\* P2\_2端子のON幅設定 \*/

trdgra1 = trdgrc1 = 0;

/\* P2\_4端子のON幅設定 \*/

trdgrb1 = trdgrd1 = 0;

/\* P2\_5端子のON幅設定 \*/

trdstr = 0x0d;

/\* TRD0カウンタ開始 \*/

## R8-マイコン実習(R8Cで実現)

### 関係知識

- trdfcr = 0x01;
- 

## R8-マイコン実習(R8Cで実現)

### 関係知識

- trdmr = 0xf0; バッファレジスタ設定
-

# R8-マイコン実習(R8Cで実現)

## 関係知識

- ・ `trdoer1 = 0x01;`      出力端子の選択  
`trdoer1`

# R8-マイコン実習(R8Cで実現)

## 関係知識

- ・ `trdpsr x;`    **TRDIOBx , Cx , Dx** 端子設定  
`trdpsr0 = 0x68;`      `/* TRDIOB0,C0,D0端子設定      */`

`trdpsr1 = 0x55;`      `/* TRDIOA1,B1,C1,D1端子設定      */`



# R8-マイコン実習(R8Cで実現)

## 関係知識

- ・ **trdgra0 = trdgrc0 = 39999; 周期設定**

```
trdgra0 = trdgrc0 = 39999;          /* 周期          */
```

# R8-マイコン実習(R8Cで実現)

## 関係知識

- ・ **出力端子のON幅設定**

```
trdgrb0 = trdgrd0 = 0;          /* P2_2端子のON幅設定  */
trdgra1 = trdgrc1 = 0;          /* P2_4端子のON幅設定  */
trdgrb1 = trdgrd1 = 0;          /* P2_5端子のON幅設定  */

trdstr = 0x0d;                  /* TRD0カウント開始    */
```

# R8-マイコン実習(R8Cで実現)

## 関係知識

### ・TRD x カウント開始力

```
trdstr = 0x0d;                /* TRD0カウント開始 */
```

# R8-マイコン実習(R8Cで実現)

## 関係知識

### ・メインルーチンのお仕事

```
void main( void )
{
    init();                /* 初期化 */

    while( 1 ) {
        trdgrd0 = 39998 * dipsw_get() / 15;
    }
}
```

dipsw\_get()はDIPスイッチの0～15の値を読み出す。  
DIPスイッチの切り替えで16段階に分ける事ができる  
39998に正規化していると言う

## 本日の課題

1. 周期が出力される端子(P2\_1)のPWM 出力を禁止して、通常のI/O ポートにしてください。このとき、この端子は出力端子として"0"を出力してください。
2. 逆相の端子(P2\_3、P2\_6、P2\_7 の3 つとも)のPWM 出力を禁止して、通常のI/O ポートにしてください。このとき、この端子は出力端子として"0"を出力してください。

※プログラムは、(1)の状態から改造するものとする。

3. P2\_4 端子もP2\_2 端子と同様に、ディップスイッチに合わせて PWM 信号が出力されるようにしてください。出力の仕方は、サンプルプログラム39 行目の「`39998 * dipsw_get() / 15`」を使用することとする。

※プログラムは、(2)の状態から改造するものとする。

## 本日の課題

4. P2\_5 端子もP2\_2 端子と同様に、ディップスイッチに合わせて PWM 信号が出力されるようにしてください。出力の仕方は、サンプルプログラム39 行目の「`39998 * dipsw_get() / 15`」を使用することとする。

※プログラムは、(3)の状態から改造するものとする。

5. P2\_4 端子、P2\_5 端子のPWM 出力を禁止して、通常のI/O ポートにしてください。このとき、この端子は出力端子として"0"を出力してください。

※プログラムは、(4)の状態から改造するものとする

6. A/D変換のプログラムを併用して正相出力のみを半固定VR の回転に合わせて出力を変化させなさい。

※半固定VR 右一杯で最大の明るさ左一杯で消灯するようにする。