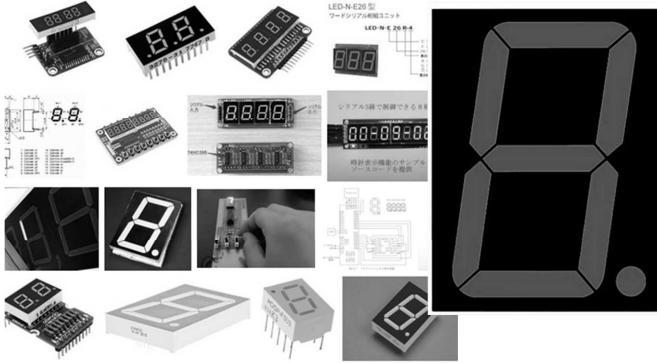
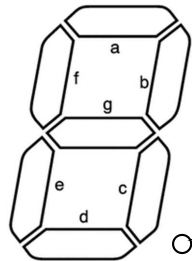


7セグメントLED (7セグ) のいろいろ



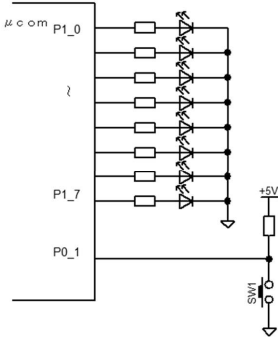
7セグメントLED (7セグ)



各セグメントにはa～gの記号が割り付けられている。  
小数点用のドットで8ビットを構成

小数点用のドット DP

8bitのLEDを表示する



図のようにLEDが8個接続されている  
新しい回路を提示されたら動作論理を確認する。

LEDは 正論理 端子に1が出力されると点灯  
SW1は負論理 ボタン を 押されると 0 入力

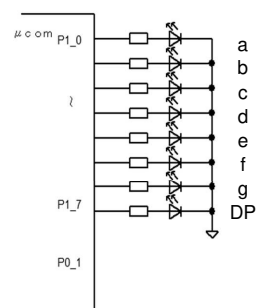
同じ素子の同極の端子が共通に接続されているものをコモン接続と言う。  
LEDのカソードが共通なのでカソードコモンと呼ばれる。

8bitのLEDを表示する

LEDを同時に点灯制御するので所定の1バイトをポートに出力すれば点灯制御できる。  
このように出力したり、入力したりする、1バイトの（まとまった）ビットの構成をビットパターンと呼ぶ。

所定の時間でビットパターンを切り替えれば動的なLED制御になる。また、モーターなどアクチュエータを取り付ければ順次動作を変更する制御もできる。

制御回路はこれが使える



表示のためのビットパターンは？

P1	dp	g	f	e	d	c	b	a	データ
0	0	0	1	1	1	1	1	1	3f
1	0	0	0	0	0	1	1	0	06
2	0	1	0	1	1	0	1	1	5b
3	0	1	0	0	1	1	1	1	4f
4	0	1	1	0	0	0	1	1	06
5	0	1	1	0	1	1	0	1	6d
6	0	1	1	1	1	1	0	1	7d
7	0	0	1	0	0	1	1	1	27
8	0	1	1	1	1	1	1	1	7f
9	0	1	1	0	1	1	1	1	6f

const char seg[]={63,6,91,79,102,109,125,39,127,111}  
0x3f,0x06,0x5b,0x4f,0x66,...,0x6f

const char seg[] とは

const char seg[]={63,6,91,79,102,109,125,39,127,111}

char 型配列変数

収まるデータ

constは定数のための接頭語

機械語に展開される時にR8ではROMセクションに割り付けられる。

固定値領域

P1 = seg[1]; とすると7segLEDに 1 が表示される

SW1が押された回数をカウントしてみよう

SW1は負論理で押されると0、離されると1になる、  
SW1は宣言済とする

```
void main(){
  int i=0;
  p1=seg[i];
  while(1){
    if(sw1==0){
      i++;
      p1=seg[i];
      while(sw1==0);
    }
  }
}
```

```
void main(){
  int i=0;
  p1=seg[i];
  while(1){
    while(sw1==1);
    i++;
    p1=seg[i];
    while(sw1==0);
  }
}
```

SW1が押された回数をカウントしてみよう

SW1は正論理で押されると1、離されると0になる、  
SW1は宣言済とする

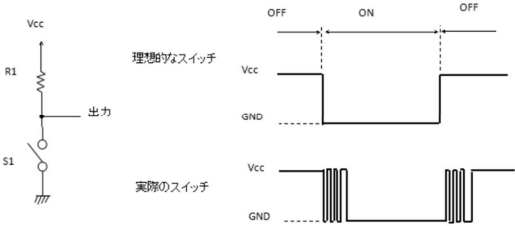
```
void main(){
  int i=0;
  p1=seg[i];
  while(1){
    if(sw1==1){
      i++;
      p1=seg[i];
      while(sw1==1);
    }
  }
}
```

```
void main(){
  int i=0;
  p1=seg[i];
  while(1){
    while(sw1==0);
    i++;
    p1=seg[i];
    while(sw1==1);
  }
}
```

現実には甘くない

接点の特性 --- チャタリング

現実のスイッチにはONまたはOFF直後に出力が短い時間 ON/OFFを繰り返す現象・「チャタリング」があります。



接点の特性 --- チャタリング

CR積分回路を使った実際例



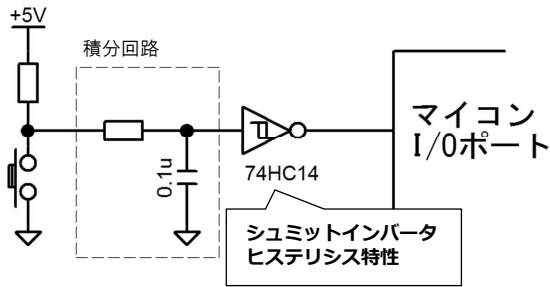
チャタリング防止 ～ 対処法 ～

装置が安定して動作するためには、チャタリング  
対策は重要で以下の方法について説明する

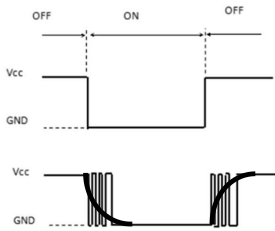
- 1.ハードウェアで対応する
- 2.ソフトウェアで対応する

ハードウェアで対応 ～ 割り込みなど2重処理が問題なる時 ～

積分回路とシュミットバッファの利用

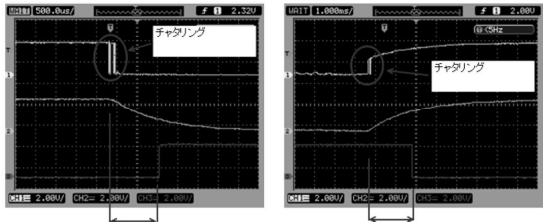


接点の特性 - - - チャタリング（対策）



コンデンサに充放電され波形が赤線のように  
なりますコンデンサが十分に大きければ  
滑らかに変化します。  
論理素子は中間電圧の入力禁止なので  
シュミットトリガ入力で受けます。

接点の特性 - - - チャタリング（対策）

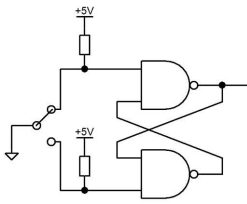


チャタリング防止

CR積分回路を使った実際例 R1,R2は一般的に数KΩ～100KΩの範囲います。  
遅れ時間は、正確にはICの種類、電源電圧、R1、R2,C1の値で決まり、  
目安としては以下の式で計算しても良いです。  
(スイッチ ON時の遅れ時間 Ton)  
 $Ton = R2 \times C1$  ----①  
(スイッチ off時の遅れ時間 Toff)  
 $Toff = (R1 + R2) \times C1$  ----②  
例えば、 R1 = R2 = 10K C1 = 0.1μF の場合、  
遅れ時間 スイッチON  $Ton = 10K \times 0.1\mu = 1msec$   
スイッチOFF  $Toff = (10K + 10K) \times 0.1\mu = 2msec$   
チャタリング時間に対して十分大きくなっています。

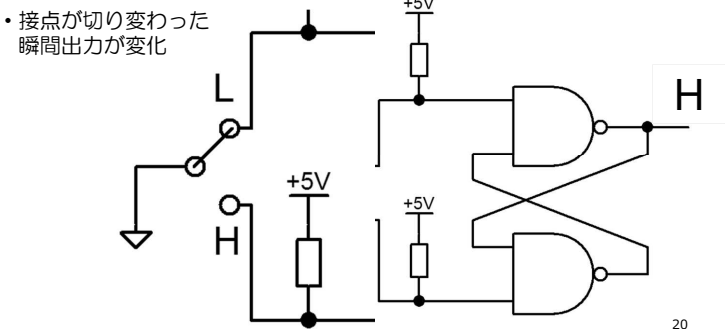
積分回路コストは安いが・・・  
パルスを訛らせた分遅れが生じる。  
と言う問題点がある。

完璧なチャタリング対策



R-Sフリップフロップの利用

スイッチが切り替わる瞬間、R-SどちらもH



チャタリング防止 ～ ソフトウェアで対策 ～

スイッチの変化を検出したらしばらく待って  
レベルを確認しON－OFFを確定する。  
この場合処理が遅れる  
スイッチの変化を検出したら対応した処理を行  
い、一定時間たってからスイッチを受け付ける。  
入力の間隔が十分長いことが前提  
受付禁止時間中は入力できない。

## チャタリングまとめ

- トグルスイッチ、押しボタンスイッチ、タクトスイッチなどの機械式スイッチ（リレー 接点も含む）にはチャタリングがある。
- チャタリングは接点の「バウンド」または「擦れ」などで発生する。
- チャタリングの発生度合、時間はスイッチの種類、操作方法により異なり、必ず発生すると思ったほうが良い。またノイズ（雑音）耐性も高まる。
- チャタリング時間は数100μsec～数10msec程度。スイッチの仕様による。
- チャタリングが問題となるシステムではチャタリング防止が必要。
- チャタリング対策には、ソフトウェアとハードウェア的な対策がある。  
高速な対応が必要な場合はハードウェア（FF回路使用）  
低コストなソフト処理、状況により使い分ける。

22

## SW1が押された回数をカウントしてみよう

SW1は正論理で押されると1、離されると0になる、  
SW1は宣言済とする

```
void main(){
    int i=0;
    p1=seg[i];
    while(1){
```

```
        while(sw1==0);
        i++;
        p1=seg[i];
        timer(5);
        while(sw1==1);
```

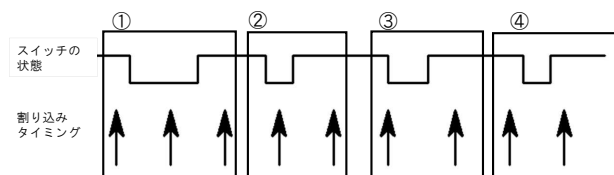
← チャタリングが収まるまで待つ

```
    }
```

23

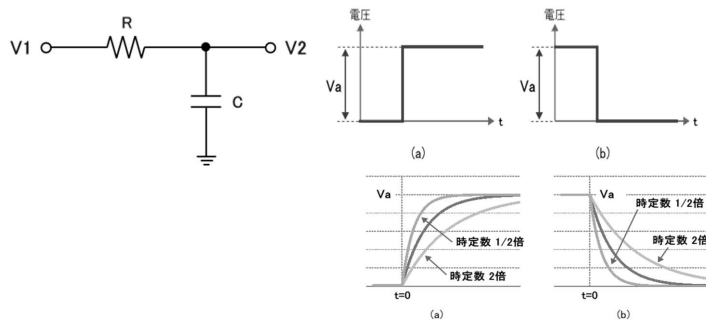
## チャタリングのもう一つの対策

- スイッチ処理を割り込みで行う。  
大前提{スイッチは少なくとも割り込み間隔未満でON-OFFされないこと。}



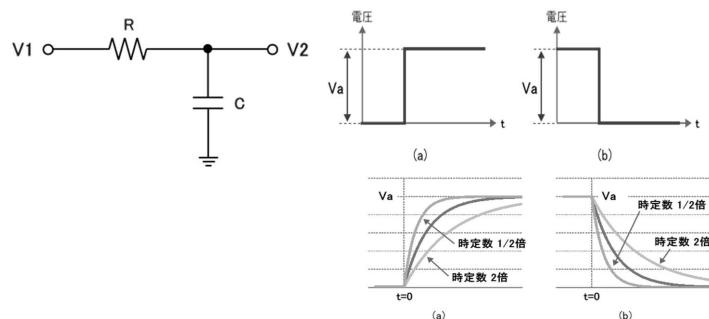
24

## 積分回路について



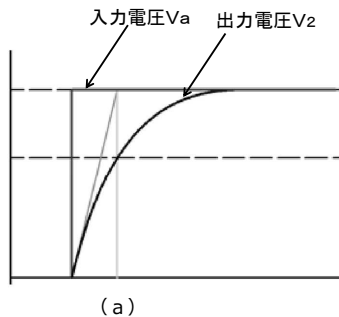
25

## 積分回路について



26

## 時定数について



$$V_2 = V_a \times e^{-\frac{1}{CR}t}$$

$$V_2 = V_a \times \left(1 - e^{-\frac{1}{CR}t}\right)$$

27