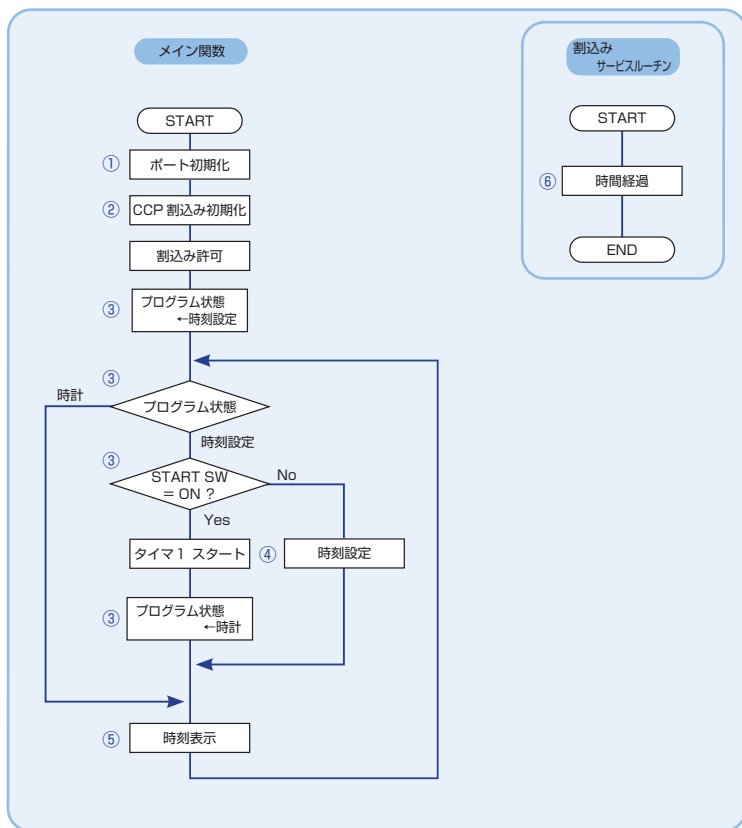




## 04-3

## 概略フローチャート



## 04-4 概略フローチャートの解説

## ① 使用する I/O ポートの設定

PORTB : 出力ポート  
RA0 ~ RA2 : 入力ポート

② CCP1 割り込みを使い 1m 秒ごとに割り込み発生  
<設定内容>

CCP モジュール

- ・モード : CCP 割り込み
- ・周期 : 1000  $\mu$ 秒

タイマ 1

- ・クロック源 : 内部クロック
- ・プリスケアラ : 1

③ プログラム状態を記憶しておき状態ごとに処理を分岐  
<状態>

- ・時刻設定 : 時刻あわせ時  
START SW を押すと "時計" に遷移
- ・時計 : 通常時計時  
CCP 割り込みにより 1m 秒ごとに時間をプラス

## ④ 時刻の設定を行う

"分" あわせ用 SW を押すと, "分" + 1 (59  $\rightarrow$  0)  
"時" あわせ用 SW を押すと, "時" + 1 (23  $\rightarrow$  0)  
詳細は後述

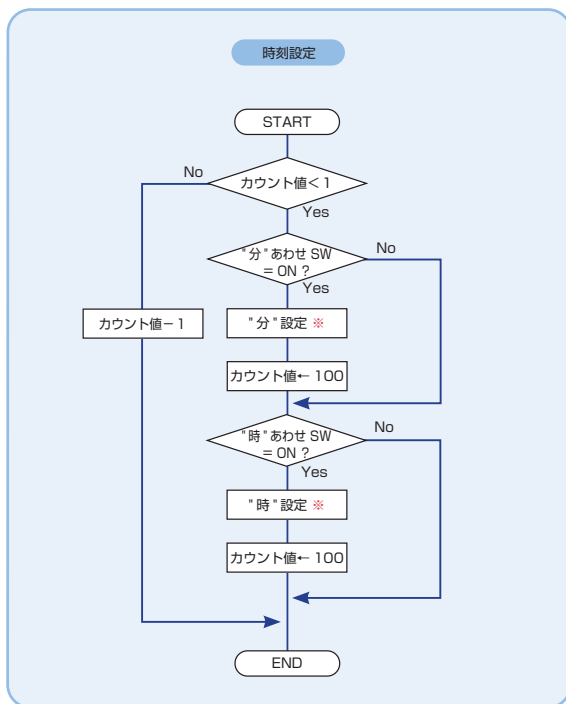
## ⑤ ダイナミック点灯を利用し, 4 つの 7SEGMENT LED に残り時間を表示する

## ⑥ 1m 秒ごとに割り込み, 時を刻む

## 04-5 時刻設定

マイコンは処理速度が速いため、人間が一瞬 SW を押しただけのつもりでも、マイコンには押されっぱなしの状態と判断される。そのため、プログラムに調整を入れないと 1 分ずつ、1 時間ずつの時間調整は困難である。

(一回 SW を押しただけで何分、何時間も増えてしまう)



※ "分" が 60 になった場合 0 に戻す

"時" が 24 になった場合 0 に戻す

## 04-6 ソースコード

掲載ソースは弊社 HP で公開中のエレモサンプルソースからの抜粋です。  
(URL : [https://www.adwin.com/image/support/ADM\\_SampleSource.zip](https://www.adwin.com/image/support/ADM_SampleSource.zip))

## ■ 時刻設定

```
/*
 時刻設定
*/
void setTime( void ){

static int cnt = 0; // 連続して時刻変更が行わないように調整するための変数

// 連続して時刻変更が行われないように変数 cnt で調整する
if( cnt < 1 ){

// "分" あわせ用 SW が押された場合
if( SW_MIN == SW_ON ){
g_min++; // "分" + 1

if( g_min > 59 ){
g_min=0; // "60分" の場合は "0分" に戻す
}
cnt = 100;
}

// "時" あわせ用 SW が押された場合
if( SW_HOUR == SW_ON ){
g_hour++; // "時" + 1

if( g_hour > 23 ){
g_hour = 0; // "24時" の場合は "0時" に戻す
}
cnt = 100;
}
}
}
else{
cnt--; // 連続して時刻変更が行われないように変数 cnt で調整する
}
}
```

時刻表示

```

/*
 1つ1つの7SEGに数字セット
 <引数>
   num : セットする数字 ( 数値 )
   id  : 0 -> "時" 2桁目
         1 -> "時" 1桁目
         2 -> "分" 2桁目
         3 -> "分" 1桁目
*/
void set7Seg( unsigned int num, unsigned int id ){

    PORTB = ( ( 0b00001000 >> id ) | ( num << 4 ) );

}

/*
 時刻表示
*/
void dispTime( void ){

    set7Seg( g_hour / 10, 0 ); // "時" 2桁目 表示
    set7Seg( g_hour % 10, 1 ); // "時" 1桁目 表示

    set7Seg( g_min / 10, 2 ); // "分" 2桁目 表示
    set7Seg( g_min % 10, 3 ); // "分" 1桁目 表示
}

```