

```
#include <htc.h>
```

```
/*SSR-1 段タイマー、リセットボタン付き、タイマー時間 T1,T2 はソフトによる*/
```

```
/*R0:キッチンタイマープログラム 2012-07-03 作成を再編 2012-08-08 by Takehiko Inoue*/
```

```
/*R1:2012-08-12 : SW1 でタイマーT1start : 15min 後出力 off で T2 ブザーstart 5sec 後ブザー停止*/
```

```
/*R2:EEPROM で off タイム書き込み*/
```

```
/*R3:2013-03-26:時間設定に改良*/
```

```
__CONFIG(FCMDIS & IESODIS & BOREN & UNPROTECT & MCLREN & PWRTEN & WDTDIS  
& INTIO);
```

```
__IDLOC(0001);
```

```
__EEPROM_DATA(6,0,0,0,0,0,0,0);
```

```
#define _XTAL_FREQ 1000000
```

```
#define JIKAN 500
```

```
#define beat 10
```

```
ioport0;
```

```
unsigned char State;
```

```
unsigned char Flag1,Sec1,Min1,Hr1,Flag2,Sec2,Min2;
```

```
void Buzzer(unsigned cont);
```

```
void Process4(void);
```

```
void Process0(void);
```

```
main()
```

```
{
```

```
    OSCCON=0x40;                /* 1MHz */
```

```
    GPIO=0;                    /* GPIO ピンの出力データを 0 にする */
```

```
    ioport0;
```

```
    GPIF=0;
```

```
    CMCON0=0x07;                /*コンパレータ停止 */
```

```
    /*EEPROM に切り時間 min 書き込み準備時間*/
```

```

Buzzer(2); // 3kHz ブザーオンオフ*/
__delay_ms(beat);
Buzzer(0);
__delay_ms(JIKAN);

/*EEPROM に切り時間 min 書き込み*/

unsigned char i,sw_cnt0=0;

while(GPIO4==0){ //GP4 スイッチ(SW1)が押された状態で
*/

    if(GPIO0==0) continue; //GPO スイッチ(SW3)が押されたら*/
    __delay_ms(50);
    if(GPIO0==0) continue;

    eeprom_write(0,++sw_cnt0); //sw_cnt0 をインクリメントしてデータ EEPROM
の 0 番地に書き込む*/
    while(!(GPIO0==0)); //スイッチ(SW3)チが離されるのを待つ*/
}
    __delay_ms(JIKAN);

/*EEPROM の切り時間 min 書き込み値確認*/

for(i=0;i<(eeprom_read(0)*2);i++){ //Hr1 設定の点滅*/
    __delay_ms(JIKAN);
    GPIO5^=1; // LED1 点滅 motor 停止 */
    GPIO1^=1;
}

/* 変数初期化 */

State= 0;

Flag1= 0; //T1*/
Sec1 = 1;
Min1 = 1;

```

```

Hr1 = eeprom_read(0);

Flag2= 0;                                /*T2*/
Sec2 = 10;
Min2 = 0;

GPIO=0;                                   /*出力準備*/

Buzzer(1);                                // 4kHz ブザーオン*/
__delay_ms(beat);                         /*点灯準備時間*/
Buzzer(0);
__delay_ms(JIKAN);
Buzzer(2);                                // 3kHz ブザーオン*/
__delay_ms(beat);                         /*点灯準備時間*/
Buzzer(0);
__delay_ms(JIKAN);

/* Timer1 初期設定 1秒周期 **/
T1CON = 0x30;                             // 内部クロック、1/8 4usec
TMR1H = 0x85;                             // TMR1 上位セット
TMR1L = 0xF0;                             // TMR1 下位セット
TMR1IE = 1;                               // 割り込み許可
TMR1ON = 1;                               // タイマ1スタート*/

/* 割り込み許可 */
PEIE  = 1;
GIE   = 1;

GPIO1=1;                                   /* LED2 on stand by*/
GPIO5=1;                                   /* LED1 on motor stop*/

/***** メインループ *****/
while(1){
    if(GPIO0==0) Process0(); /*GP0 スイッチ(SW3)が押されたらm o t o r 正転または
逆転*/
    if(GPIO4==0) Process4(); /*GP4 スイッチ(SW1)が押されたらタイマー処理*/
}
}

```

```

ioport0
{
    TRISIO=0x19;          /* d00011001:GP0,GP3,GP4 ピンは入力ピン、GP1~GP2,GP5 ピ
ンは出力ピン*/
    ANSEL=0x00;          /* GP0,GP1,GP2,GP3,GP4,GP5 ピンをデジタル I/O モードにする
*/

    WPU=0x19;           /* GP0,GP3,GP4 pull up*/
    GPPU=0;              /* pull up enable*/
}

/*****
* Timer1 割り込み処理
* Timer1 : 1 秒周期
*****/
void interrupt T1ISR(void){

    /*** 秒タイマ Timer1 割り込み処理 ***/
    TMR1H = 0x85;        // TMR1 上位セット
    TMR1L = 0xF0;        // TMR1 下位セット
    TMR1IF = 0;         // 割り込みフ
ラグクリア
    GPIF=0;             /* リセット
*/

    GPIO2 = 1;          /* チック音
on*/

    /*** タイマ ***/
    if(Flag1){
        if(Sec1 == 0){  /*Sec1=0 ならば*/
            if(Min1 == 0){ /*Min1=0 ならば*/
                if(Hr1 > 0){ /* Hr が残っていれば*/
                    Hr1--;    /*Hr 減し
*/
                    Min1 = 59; /*Min1=59 に
セット*/
                }
            }
        }
    }
}
else{

```

```

GPIO5 = 0; /* LED1 off
n5=1 */
GPIO1 = 1; /* LED2 on
n6=0 motor 正転*/
Buzzer(2); // 3kHz ブザー
オン*/
Flag1 = 0; /* T1 motor
正転カウントアップ off*/
Flag2 = 1; /* T2 motor
ブレーキカウントアップ&ブザーカウント on*/
}
}
else
Min1--; /* 分減 */
Sec1 = 59; /* 59sec にセ
ット*/
}
else
Sec1--; // 秒減
}
if(Flag2){
if(Sec2 == 0){ // 秒終了か?
if(Min2 > 0){ // 分が残っていれば?
Sec2 = 59; // 59 秒に再セ
ット
Min2--; // 分減
}
else{
GPIO5 = 0; /* LED2 off
n6=1 & LED1 off n5=1 & motor Brake*/
GPIO1 = 0;
Buzzer(0);
Flag2 = 0;
TMR1ON = 0; /*Timer1 停止
*/
}
}
else

```

```

                Sec2--;                                // 秒減
            }
            GPIO2 = 0;                                /* チ ッ ク 音
off*/
        }

/* *****
/* ブザー制御
/* CCP1 の PWM モードで出力 デューティは 50%
/* *****/
void Buzzer(unsigned cont){

    switch(cont){
        case(0):
            /* Timer2 停止 */
                T2CON = 0;                            // Timer2 停止
                CCP1CON = 0;                          // PWM モード停止
            break;

        case(1):
            /* Timer2 初期設定 4kHz */
                PR2 = 61;                             // 4usec*62=0.25msec
                TMR2 = 0;
                T2CON = 0x04;                          // Timer2 スタート
                CCP1CON = 0x0C;                        // PWM モード
                CCPR1L = 31;                           // Duty 50%
            break;

        case(2):
            /* Timer2 初期設定 3kHz */
                PR2 = 81;                             // 4usec*82=0.33msec
                TMR2 = 0;
                T2CON = 0x04;                          // Timer2 スタート
                CCP1CON = 0x0C;                        // PWM モード
                CCPR1L = 41;                           // Duty 50%
            break;

        default:break;
    }
}

```

```
    }  
}
```

```
/*Process0*/
```

```
void Process0(void){
```

```
    /*Pleriminal operation */
```

```
    switch(State){
```

```
        case(0):
```

```
/*リセット後正転*/
```

```
            if(GPIO0==0){
```

```
                __delay_ms(JIKAN);
```

```
                /*Buzzer(1);*/
```

```
// 4kHz ブザー
```

```
オンオフ*/
```

```
                GPIO5 = 0;
```

```
/* LED1 off
```

```
n5=1 */
```

```
                GPIO1 = 1;
```

```
/* LED2 on
```

```
n6=0 motor 正転*/
```

```
            }
```

```
            else if(!(GPIO0==0)){
```

```
                __delay_ms(JIKAN);
```

```
                /*Buzzer(0);*/
```

```
                GPIO0 = 0;
```

```
            }
```

```
            State++;
```

```
            break;
```

```
        case(1):
```

```
/*リセット後逆転*/
```

```
            if(GPIO0==0){
```

```
                __delay_ms(JIKAN);
```

```
                /*Buzzer(2);*/
```

```
// 3kHz ブザー
```

```
オンオフ*/
```

```
                GPIO5 = 1;
```

```
/* LED1 on
```

```
n5=0 */
```

```
                GPIO1 = 0;
```

```
/* LED2 off
```

```
n6=1 motor 逆転*/
```

```
            }
```

```
            else if(!(GPIO0==0)){
```

```
                __delay_ms(JIKAN);
```

```

        /*Buzzer(0);*/
        GPIO5 = 0;
    }
    State++;
    break;

case(2): /*motor 停止*/
    if(GPIO0==0){
        __delay_ms(JIKAN);
        Buzzer(1); // 4kHz 3kHz
        /*ブザーオンオフ*/
        __delay_ms(beat);
        Buzzer(2);
        __delay_ms(beat);
        Buzzer(0);
        GPIO5 = 0;
        GPIO1 = 0;
    }
    else if(!(GPIO0==0)){
        __delay_ms(JIKAN);
        Buzzer(0);
    }
    State = 0;
    break;

default:break;
}
}

/*Process4*/
void Process4(void){
    Flag1=1; /* T1 motor 正転カウントアップ on*/
    Buzzer(1); /*ブザー音 on*/
    __delay_ms(beat);
    Buzzer(0); /*ブザー音 off*/
    GPIO5=0; /* LED1 off n5=1*/
    GPIO1=0; /* LED2 off n6=1 motor brake

```



\*/

}