

簡易砂時計

概要

圧電スピーカは、電極に信号電圧を加えることにより圧電体(圧電素子)が歪み、その振動を音(空気の振動)として聞くものです。逆に、圧電体に加えられた力(音圧など)を電圧に変換することも出来ます。そこでこの特性を利用して、“開始スイッチ”や“停止スイッチ”を使用しない簡易な砂時計を製作し



ました。



<仕様>

- 時計の“開始”は、圧電スピーカを、“指で軽く叩く”だけとする。
- 時計のカウント値は、1分~8分までの8種類とする。(1分間隔)
- 時計のカウント中の“停止”は、再度、圧電スピーカを、“指で軽く叩く”だけとする。
- 時計のカウント中は、1秒毎に“ピッ”という音(約3kHz)を発生させる。
- 時計がタイムアウトすると、ピンポンパンポン、ピンポンパンポンと音を発生させる。

動作原理

<時計の開始の判断> 圧電スピーカからの信号をA/D変換で取り込み、ある値(V1)以上の電圧が得られる

と開始する。電源電圧(3V時)□V1=300mV 電源電圧(5V時)□V1=490mV

<時計のカウント>

1. PICのCCPモジュールをキャプチャモードで使用し、0.1秒の周期割り込みを発生させる。
2. 1秒毎に“ ピッ ” という音(約3kHz)を約50msec間発生させ□LEDを点滅させる。
3. 圧電スピーカからの信号をA/D変換で取り込み、ある値(V1)以上の電圧が得られると停止する。
これは、カウントを途中で停止させるための処理です。
4. カウント時間が、ディップスイッチ(SW1□SW2□SW3)で指定された時間(1分~8分)経過するとタイムアウトする。

<時計のタイムアウト> 時計がタイムアウトすると、ピンポンパンポン、ピンポンパンポンと音を発生させる。

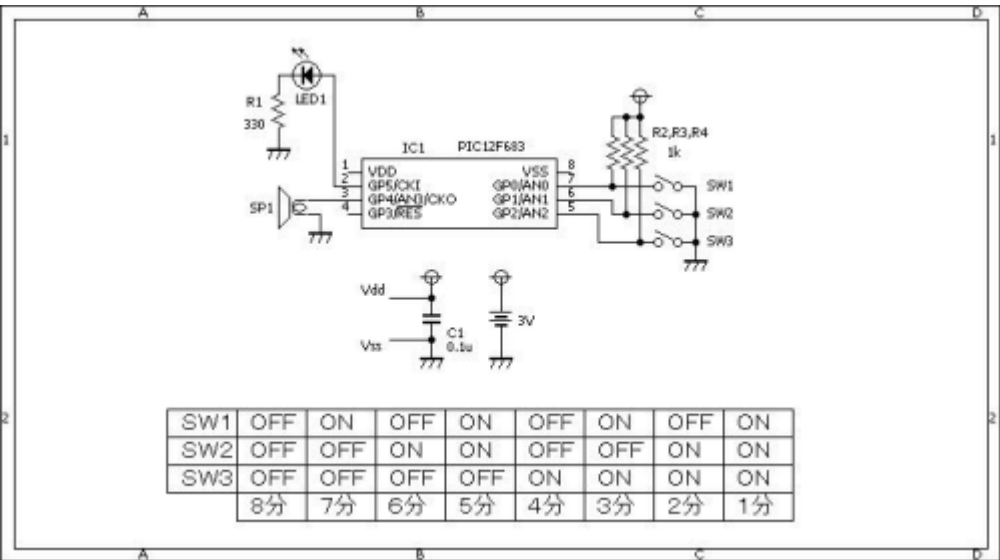
- ピン= “ ド”(523Hz/250msec)
- ポン= “ ミ”(659Hz/250msec)
- パン= “ ソ”(784Hz/250msec)
- ポ〜ン= “ ド”(1047Hz/500msec)

休止符(500msec)

- ピン= “ ド”(1047Hz/250msec)
- ポン= “ ソ”(784Hz/250msec)
- パン= “ ミ”(659Hz/250msec)
- ポーン= “ ド”(523Hz/500msec)

<時計の停止中> LEDを1秒間隔で点滅させる。

回路図



ソースコード

[sandTimerV1.c](#)

```
//*****
*****
/*
< 簡易砂時計 (スイッチ不要) >
*/
//*****
*****

#define      SW1          GPIO.F0
#define      SW2          GPIO.F1
#define      SW3          GPIO.F2

#define      SP           TRISIO.F4

#define      LED          GPIO.F5

#define      INPUT_MODE    1
#define      OUTPUT_MODE   0

#define      ON            0
#define      OFF           -1

//*****
*****

static int      clock, blink;

void  interrupt(){
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //
        if (blink != OFF) {
            blink++;
            if (blink == 5) {
                LED = ~LED;
                blink = ON;
            }
        }
        //
        clock++;
    }
}

//*****
*****

short  Delay_sec(unsigned int sec)
{
    clock = 0;
    blink = OFF;
    sec *= 10;
}
```

```
while (clock < sec) {
    if ((clock % 10) == 0) {
        LED = 1;
        Sound_Play(3000, 50);    //3kHz×50msec
        LED = 0;
    }
    //
    SP = INPUT_MODE;
    if (Adc_Read(3) > 100) {
        SP = OUTPUT_MODE;
        blink = ON;
        return (-1);
    }
    SP = OUTPUT_MODE;
}
blink = ON;
//
return (0);
}

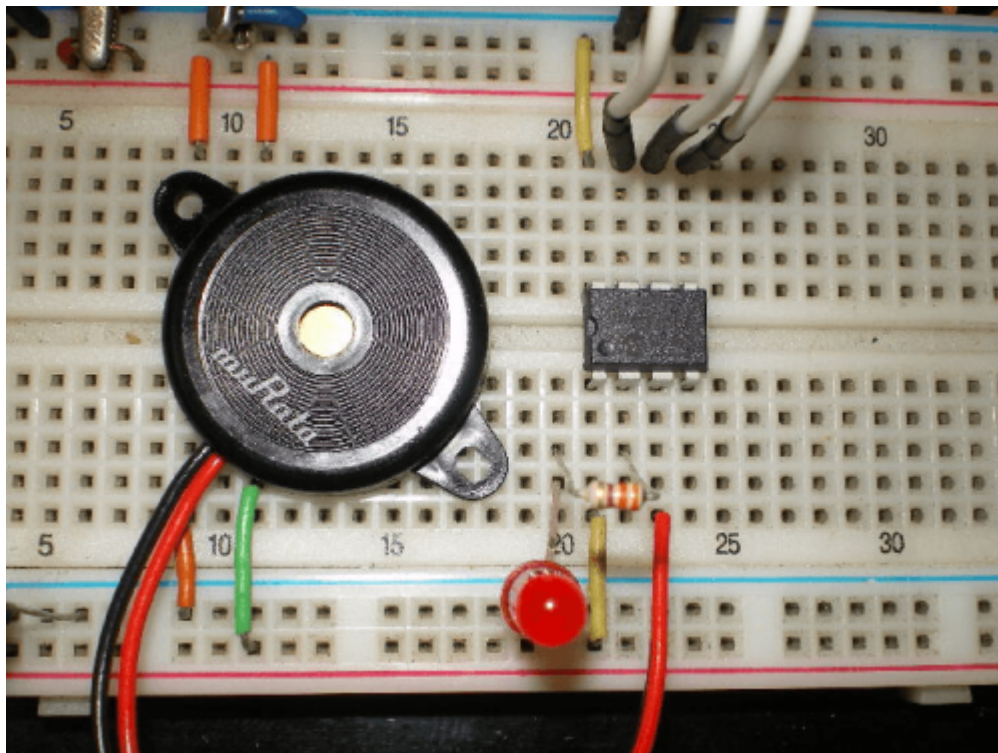
//*****
*****

void main()
{
    //
    OSCCON = 0b01010000;    //クロックを2Mhzに設定
    CMCON0 = 0b00000111;    //コンパレータは使用しない。
    ANSEL  = 0b00001000;    //A/D変換はAN3を使用する。
    TRISIO = 0b00001111;    //入出力ポートを設定する。
    // CCPの設定
    PIE1.CCP1IE = 1;
    PIR1.CCP1IF = 0;
    CCP1CON = 0b00001011;
    CCPR1L = 0x6A;    // 0.1sec...10hz...クロックが2Mhzの時
    CCPR1H = 0x18;    // 0.1sec...(1÷8000000)*4*8*6250
    // TIMER1の設定
    PIE1.TMR1IE = 0;
    PIR1.TMR1IF = 0;
    TMR1L = 0;
    TMR1H = 0;
    T1CON.T1CKPS0 = 1;
    T1CON.T1CKPS1 = 1;
    T1CON.TMR1ON = 1;
    //
    Sound_Init(&GPIO, 4);
    //
    clock = 0;
    blink = ON;
    // 割り込みの設定
```

```
INTCON.PEIE = 1;
INTCON.GIE = 1;
while (1) {
    SP = INPUT_MODE;
    while (Adc_Read(3) < 100)
        ;
    SP = OUTPUT_MODE;
    //
    if (Delay_sec(60 + (60 * (GPIO & 0x07))) == 0) {
        LED = 1;
        Sound_Play(523, 250);    //ド(ピン)
        LED = 0;
        Sound_Play(659, 250);    //ミ(ボン)
        LED = 1;
        Sound_Play(784, 250);    //ソ(パン)
        LED = 0;
        Sound_Play(1047, 500);    //ド(ポーン)
        LED = 1;
        Delay_ms(500);
        LED = 0;
        Sound_Play(1047, 250);    //ド(ピン)
        LED = 1;
        Sound_Play(784, 250);    //ソ(ボン)
        LED = 0;
        Sound_Play(659, 250);    //ミ(パン)
        LED = 1;
        Sound_Play(523, 500);    //ド(ポーン)
        LED = 0;
    }
    //
    Delay_ms(500);
}

//*****
*****
```

動作確認



如何ですか? 圧電スピーカを指で軽く叩くだけで開始するので、台所等で使用する場合にはとても重宝します。 “ 操作性抜群 ”

基本的な考え方は、“ 一つの素子を入力用と出力用 ” に使い分けるといことです。 発想の転換ですね! 😊! 実は、同じような考え方をしたものが、受光&発光(1個のLED)です。

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic12f683:22&rev=1588126419>

Last update: 2025/10/17 14:27

