

ミニ温度計(7セグ表示)V2

概要

7セグメントLEDは、よく使用される表示モジュールの1つです。しかし、7セグメントLEDには、「カソードコモン」と「アノードコモン」の二つのタイプが存在するので、通常はどちらかのタイプで回路設計を行うと、他方のタイプを接続するためには、回路設計をやり直す必要があります。

そこで今回は、同一の回路で、両方のタイプが使用できるように、以前に製作した、ミニ温度計(7セグ表示)の機能強化を図ってみました。

<仕様>

- 接続7セグメントLED=カソードコモンおよびアノードコモンの両方に対応します。
- 温度測定範囲=-30 ~100
- 表示桁数=3桁表示(小数点以下表示)、2桁表示(小数点以下非表示)

動作原理

温度の測定については、ミニ温度計(7セグ表示)を参照してください。ダイナミック点灯の基本的な原理については、ミニ周波数カウンタ(kHz表示)を参照してください。電源電圧の設定については、ミニ電圧&電流計(7セグ表示)を参照してください。

動作原理(ソフトウェア)

割り込み処理の中で、7セグメントLEDをダイナミック点灯させています。そこでこの処理の中で、カソードコモンとアノードコモンの二つのタイプに対応するために、ポートの反転機能を追加しました。

回路図


```
// マクロ定義
#define BYTE          unsigned short
#define WORD          unsigned int
#define DWORD         unsigned long
//
#define DATA0        0b01111011
#define DATA1        0b00011000
#define DATA2        0b11010011
#define DATA3        0b11011010
#define DATA4        0b10111000
#define DATA5        0b11101010
#define DATA6        0b11101011
#define DATA7        0b01111000
#define DATA8        0b11111011
#define DATA9        0b11111010
#define DATA_SPACE   0b00000000

#define SPACE         10

#define NON_SEG       0b00000000
sbit SEG1             at RA0_bit;
sbit SEG2             at RA7_bit;
sbit SEG3             at RA6_bit;

#define ON             0
#define OFF            1

sbit LED              at RA1_bit;
sbit SW_DISP_MODE    at RA4_bit;
sbit SW_7SEG_COMMON  at RA5_bit;

//*****
*

short seg_flg, data1, data2, data3, dot;
short tbl[11] = {DATA0, DATA1, DATA2, DATA3, DATA4, DATA5, DATA6,
DATA7, DATA8, DATA9, DATA_SPACE};

void interrupt()
{
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //□□□□点灯処理
        switch (seg_flg) {
        case 0:
            seg_flg = 1;
            SEG3 = OFF;
            if (SW_7SEG_COMMON == 0) {
                SEG3 = ~SEG3;
            }
            PORTB = (dot == 1) ? tbl[data1] | 0b00000100 :
```

```
tbl[data1];
    SEG1 = ON;
    if (SW_7SEG_COMMON == 0) {
        PORTB = ~PORTB;
        SEG1 = ~SEG1;
    }
    break;
case 1:
    seg_flg = 2;
    SEG1 = OFF;
    if (SW_7SEG_COMMON == 0) {
        SEG1 = ~SEG1;
    }
    PORTB = (dot == 2) ? tbl[data2] | 0b00000100 :

tbl[data2];
    SEG2 = ON;
    if (SW_7SEG_COMMON == 0) {
        PORTB = ~PORTB;
        SEG2 = ~SEG2;
    }
    break;
case 2:
    seg_flg = 0;
    SEG2 = OFF;
    if (SW_7SEG_COMMON == 0) {
        SEG2 = ~SEG2;
    }
    PORTB = (dot == 3) ? tbl[data3] | 0b00000100 :

tbl[data3];
    SEG3 = ON;
    if (SW_7SEG_COMMON == 0) {
        PORTB = ~PORTB;
        SEG3 = ~SEG3;
    }
    break;
}
}
}

//*****
*

int measurement()
{
    static double dat;
    static int cnt;
    //
    dat = 0.0;
    for (cnt = 0; cnt < 1000; cnt++) {
        dat += ADC_Get_Sample(2);
    }
}
```

```
    }
    //
    dat = (dat / 1000.0) * 2.4365234375;
//2.495V/1024...TL431
    dat = (((dat - 300.0) / 10.0) - 30.0) * 10.0;
    return (dat);
}

//*****
*

void    init_timer()
{
    // CCPの設定
    PIE1.CCP1IE = 1;
    PIR1.CCP1IF = 0;
    CCP1CON.CCP1M3 = 1;
    CCP1CON.CCP1M2 = 0;
    CCP1CON.CCP1M1 = 1;
    CCP1CON.CCP1M0 = 1;
    CCPR1L = 0xE8;          // 1msec...(1÷4000000)*4*1000
    CCPR1H = 0x03;        //
    // TIMER1の設定
    PIE1.TMR1IE = 0;
    PIR1.TMR1IF = 0;
    T1CON.T1CKPS0 = 0;
    T1CON.T1CKPS1 = 0;
    TMR1L = 0;
    TMR1H = 0;
    T1CON.TMR1ON = 1;
}

//*****
*

void    init_port()
{
    TRISA = 0b00111100;
    TRISB = 0b00000000;
    //
    SEG1 = OFF;
    SEG2 = OFF;
    SEG3 = OFF;
    data1 = SPACE;
    data2 = SPACE;
    data3 = SPACE;
    seg_flg = 0;
    dot = 0;
}

//*****
*
```

```
*  
  
void    init_adc()  
{  
    ANSEL    = 0b00001100;  
    ADC_Init();  
    ADCON1.VCFG1 = 1;  
    ADCON1.VCFG0 = 0;  
}  
  
//*****  
*  
  
void    main()  
{  
    static char    buf[16];  
    static int     dat, tmp;  
    //  
    OSCCON = 0b01100000;  
    //  
    init_port();  
    init_adc();  
    init_timer();  
    // 割り込みを許可する。  
    INTCON.PEIE = 1;  
    INTCON.GIE = 1;  
    //  
    while (1) {  
        // 温度を測定します。  
        dat = measurement();  
        // 小数点以下を表示有無を決定します。  
        tmp = abs(dat);  
        if (SW_DISP_MODE == 1) {  
            dot = 2;  
        } else {  
            // 小数点以下を四捨五入します。  
            if ((tmp % 10) >= 5) {  
                tmp = ((tmp / 10) + 1) * 10;  
            } else {  
                tmp = (tmp / 10) * 10;  
            }  
            dot = 0;  
        }  
        // データを表示します。  
        IntToStr(tmp, buf);  
        data1 = (buf[3] == ' ') ? SPACE : buf[3] - '0';  
        if (SW_DISP_MODE == 1) {  
            data2 = (buf[4] == ' ') ? SPACE : buf[4] - '0';  
            data3 = (buf[5] == ' ') ? SPACE : buf[5] - '0';  
        } else {  

```

```

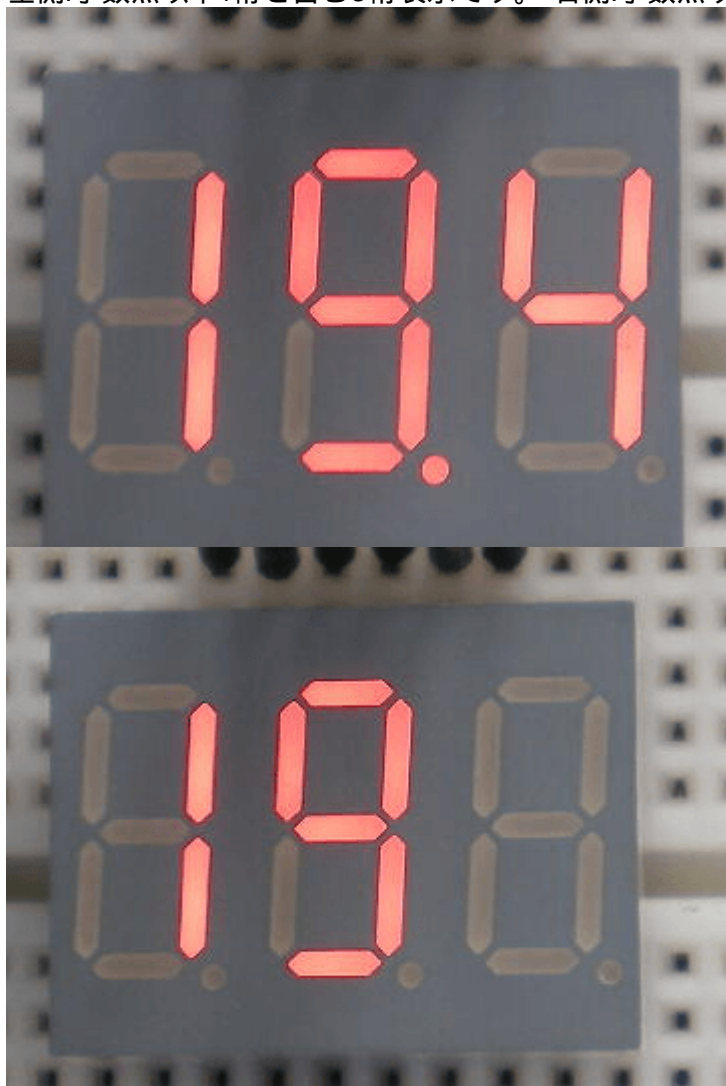
        data2 = (buf[4] == ' ') ? 0 : buf[4] - '0';
        data3 = SPACE;
    }
    // マイナスの温度であれば□□□□を点灯させます。
    if (dat < 0) {
        LED = ON;
    } else {
        LED = OFF;
    }
}

//*****
*
```

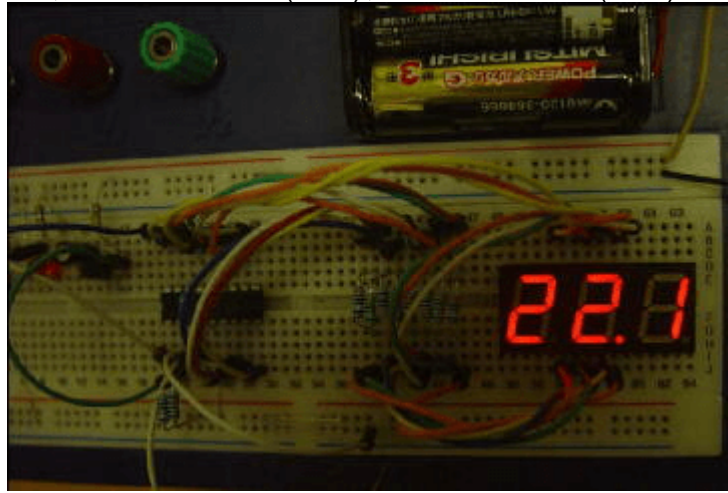
動作確認

左側から□LM61□TL431□PIC16F88□抵抗アレイ、3桁7セグ(カソードコモン)です。

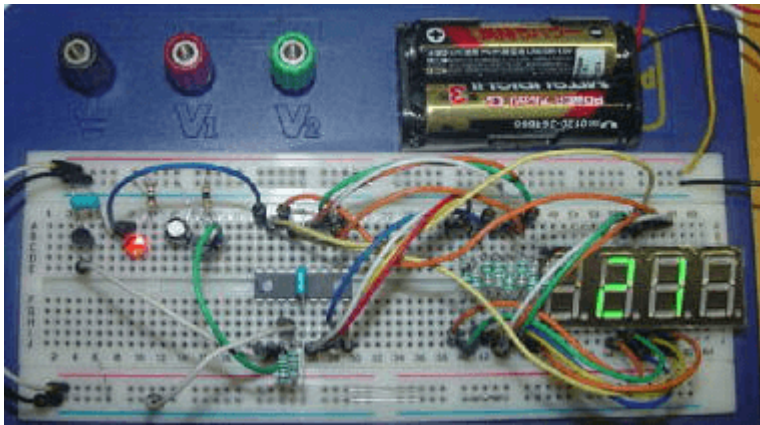
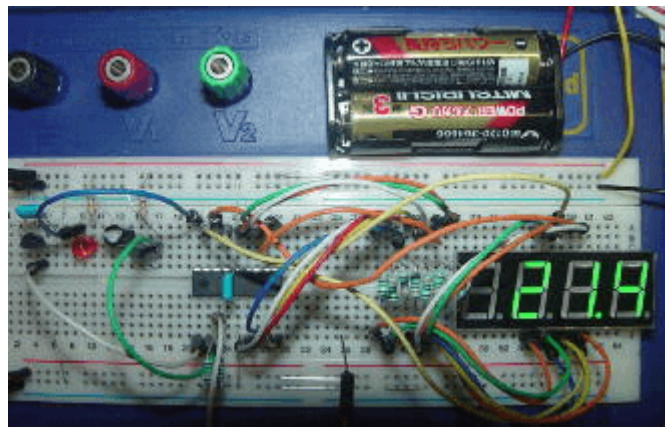
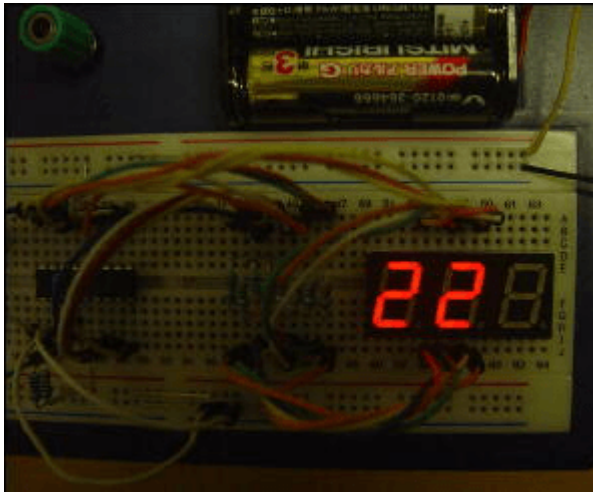
左側:小数点以下1桁を含む3桁表示です。 右側:小数点以下を四捨五入した2桁表示です。



青森県弘前市在住の清藤さんに協力して頂き、カソードコモン(上側)、アノードコモン(下側)の動作確



認をして頂きました。感謝です! 😊



From: <http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link: <http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:146&rev=1588226819>

Last update: 2025/10/17 14:27

