温度計V2(温度制御機能付き)

概要

以前に製作した、温度計(温度制御機能付き)は、次のような課題が残ってしまいました。

- 温度センサーに、LM35DZを使用したので、測定温度範囲が、0~100℃となり、0℃以下の測定が不可。
- 閾値の上限制御機能は実装、下限制御機能は未実装(mikroCの試用版の制限のため)

そこで今回は、これらの課題に対して、次のような施策を講じました。

- 温度センサーに、 LM60を使用することにより、-25~+125℃までを測定温度範囲とする。
- 閾値の上限制御機能、下限制御機能共に実装(mikroCの正式版を使用)

<LM60の主な仕様>

- 従来のLM35ではマイナスの温度を出力するには負電源が必要でしたがLM60は単電源で

マイナス/プラスの温度が測れます。

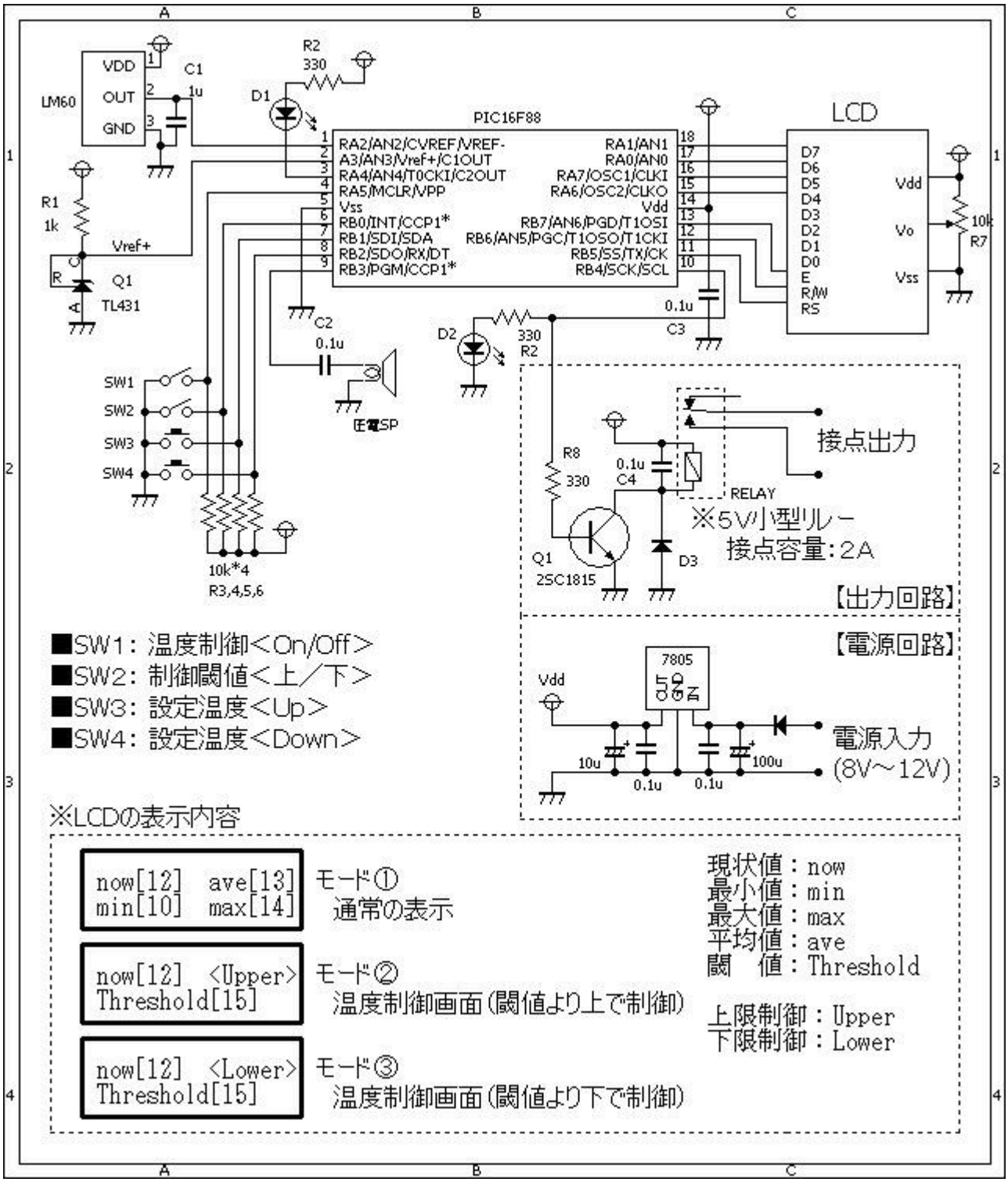
- 出力形式:アナログ
  - -40℃:174mV
  - -25℃:268mV
  - 0℃:424mV
  - +25℃:580mV
  - +100℃:1049mV
  - +125℃:1205mV
- 動作電圧:DC2.7V~10V
- 測定範囲:-25~+125℃1℃当たり6.25mV
- 誤差:±2℃(@25℃)

動作原理

- 温度センサーにはLM60を使用します。
- その出力を増幅せずにPIC入力とします。
- PICのA/D変換のVref+には、シャント・レギュレータTL431を使用し、+2.495とします。
- これで精度が2.44mV(2.495V÷1024)になります。
- LM60の出力は、温度係数が6.25mV/℃ なので約0.39℃単位の測定が出来ます。
- 温度表示画面では、現在温度、最小温度、最大温度、平均温度を計算し1秒周期で表示します。
- 温度制御画面では、閾値(threshold)の変更や、温度制御をするか否かの設定が出来ます。
- 上限値制御
  - 閾値以上になると、ブザー音(0.1秒)LED(点滅周期が短くなる)、リレーONを出力します。
  - 閾値-1℃未満になると、ブザー音(0.1秒)LED(点滅周期が遅くなる)、リレーOFFを出力します。
    ヒステリシス制御します。例えば、閾値が22℃である場合、
    リレーがONとなるのは、現在温度が22℃以上になったときです。
    リレーがOFFとなるのは、現在温度が21℃未満になったときです。
- 下限値制御

    - 閾値以下になると、ブザー音(0.1秒)□LED(点滅周期が短くなる)、リレ□ONを出力します。
    - 閾値+1℃より大きくなると、ブザー音(0.1秒)□LED(点滅周期が遅くなる)、リレ□OFFを出力します。
      ヒステリシス制御します。例えば、閾値が22℃である場合、
      リレーがONとなるのは、現在温度が22℃以下になったときです。
      リレーがOFFとなるのは、現在温度が23℃より大きくなったときです。
- 閾値は、PICのEEPROMに記憶しますので電源をOFFにしても、次回ON時には閾値を再設定する必要はありません。

回路図



ソースコード

※このプログラムは、mikroCの試用版の制限を越えるため正式版でなければコンパイルできません。

ThermoMeterV2.c

```
//*****************************************************************************
*
/*
      温度計□□□温度制御機能付き）
  ■高精度□□温度センサ □□□□を使用します。
     ・測定範囲：-25～+125℃　1℃当たり□□□□□□
□■□□変換の+□□□□に、可変シャントレギュレータ□□□□□□□□を使用します。
□□□□□□□□(min)→□□□□□□
  ■現在温度（－４０℃～９９℃）を表示します。
  ■起動時から現在までの最大温度と最小温度と平均温度を表示します。
  ■温度制御機能を実装します。（ヒステリシス制御）
     ・上限制御
       ：閾値以上で、ブザー音（0.1秒□□LED□点滅周期が短くなる）、リレ□ONを出力します。
       ：閾値－１℃以下で、ブザー音（0.1秒□□LED□点滅周期が遅くなる）、リレ□OFFを出力
します。
     ・下限制御
       ：閾値以下で、ブザー音（0.1秒□□LED□点滅周期が短くなる）、リレ□ONを出力します。
       ：閾値＋１℃以上で、ブザー音（0.1秒□□LED□点滅周期が遅くなる）、リレ□OFFを出力
します。
*/
//*****************************************************************************
*

#define       RELAY        PORTB.F4
#define       LED            PORTA.F4

#define       SW1            PORTA.F5
#define       SW2            PORTB.F0
#define       SW3            PORTB.F1
#define       SW4            PORTB.F2

#define       ON            1
#define       OFF            0

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

static int            nowData, maxData, minData, aveData,
thresholdData;
static unsigned    char    buf[8], sw3, sw4;

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

void   interrupt()
```

```
{
    if (PIR1.TMR1IF == 1) {
        PIR1.TMR1IF = 0;
        //LEDを点滅させる。
        LED = ~LED;
        //
        if (sw3 == OFF)
            sw3 = (SW3 == 0) ? ON : OFF;
        //
        if (sw4 == OFF)
            sw4 = (SW4 == 0) ? ON : OFF;
    }
}

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

void   Pwm_Change_DutyEx(unsigned int duty_ratio)
{
    CCPR1L = duty_ratio >> 2;
    CCP1CON.F6 = duty_ratio & 0b00000001;
    CCP1CON.F7 = (duty_ratio & 0b00000010) >> 1;
}

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

void   buzzer()
{
    Pwm_Start();
    Delay_ms(100);
    Pwm_Stop();
}

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

void   ThermoCntl()
{
    if (SW2 == 0) {         //閾値以下で温度制御する。
        Lcd_Custom_Out(1, 10, "<Lower>");
        //制御出力をONにするかを判断する。
        if ((nowData <= thresholdData) && (RELAY == OFF)) {
            buzzer();
            RELAY = ON;
            T1CON.T1CKPS1 = 0;
            T1CON.T1CKPS0 = 1;
        }
        //制御出力をOFFにするかを判断する。ヒステリシス制御！
        if ((nowData > (thresholdData + 1)) && (RELAY == ON)) {
```

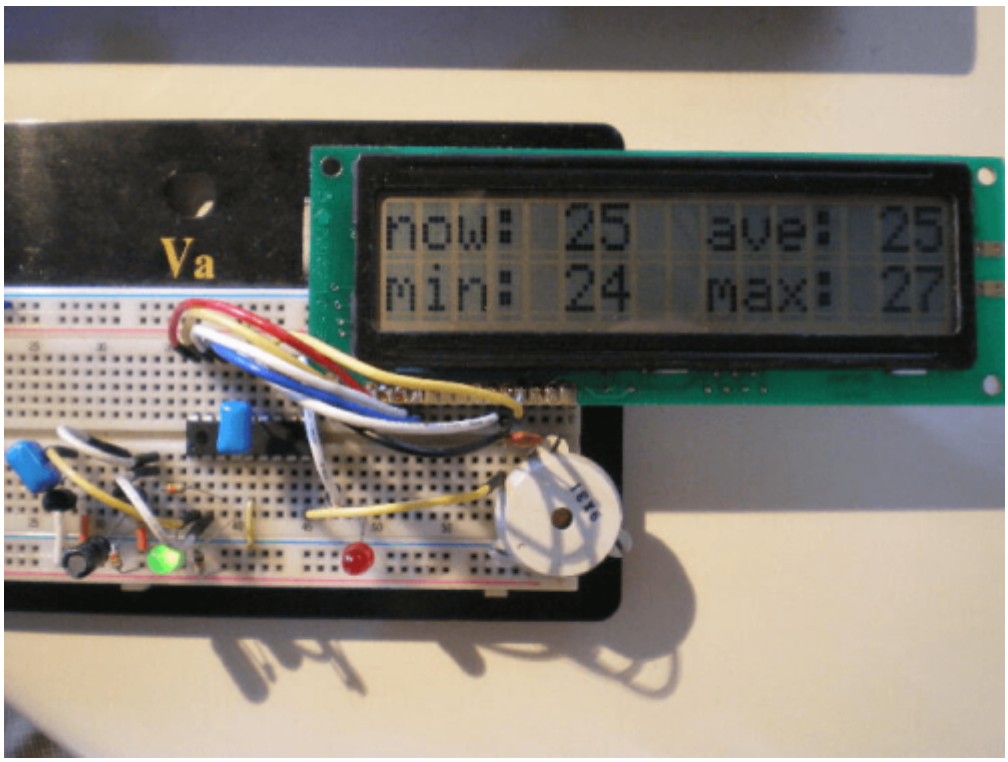
```
            buzzer();
            RELAY = OFF;
            T1CON.T1CKPS1 = 1;
            T1CON.T1CKPS0 = 1;
        }
    } else {            //閾値以上で温度制御する。
        Lcd_Custom_Out(1, 10, "<Upper>");
        //制御出力をONにするかを判断する。
        if ((nowData >= thresholdData) && (RELAY == OFF)) {
            buzzer();
            RELAY = ON;
            T1CON.T1CKPS1 = 0;
            T1CON.T1CKPS0 = 1;
        }
        //制御出力をOFFにするかを判断する。ヒステリシス制御！
        if ((nowData < (thresholdData - 1)) && (RELAY == ON)) {
            buzzer();
            RELAY = OFF;
            T1CON.T1CKPS1 = 1;
            T1CON.T1CKPS0 = 1;
        }
    }
    //
    if (sw3 == ON) {    //閾値を＋1する。
        sw3 = OFF;
        if (thresholdData < 99)
            thresholdData++;
        Eeprom_Write(0, thresholdData & 0xFF);
        Eeprom_Write(1, ((thresholdData >> 8) & 0xFF));
    }
    //
    if (sw4 == ON) {    //閾値をー1する。
        sw4 = OFF;
        if (thresholdData > -40)
            thresholdData--;
        Eeprom_Write(0, thresholdData & 0xFF);
        Eeprom_Write(1, ((thresholdData >> 8) & 0xFF));
    }
    //閾値温度を表示する。
    IntToStr(thresholdData, buf);
    Lcd_Custom_Out(2, 1, "threshold:");
    Lcd_Custom_Out(2, 11, &buf[3]);
}

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

void   main()
{
    unsigned    int        ad2, tmp;
    unsigned    char    cnt, flg;
```

```
    //
    OSCCON = 0b01110000;     //内臓クロックを8Mhzに設定する。
    CMCON = 0b00000111;         //コンパレータは使用しない。
    ANSEL = 0b00000100;         // A/D変換はAN2を使用する。
    TRISA = 0b00101100;
    TRISB = 0b00000111;
    OPTION_REG.NOT_RBPU = 0;
    //タイマー1を設定する。
    PIE1.TMR1IE = 1;
    PIR1.TMR1IF = 0;
    T1CON.T1CKPS1 = 1;
    T1CON.T1CKPS0 = 1;
    T1CON.TMR1ON = 1;
    //
    ADCON1.VCFG1 = 1;
    ADCON1.VCFG0 = 0;
    //
    Lcd_Custom_Config(&PORTA, 1, 0, 7, 6, &PORTB, 5, 6, 7);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    for (cnt = 0; cnt < 16; cnt++) {
        Lcd_Custom_Chr(1, cnt + 1, 0xFF);
        Delay_ms(100);
    }
    for (cnt = 0; cnt < 16; cnt++) {
        Lcd_Custom_Chr(2, cnt + 1, 0xFF);
        Delay_ms(100);
    }
    Delay_ms(100);
    Lcd_Custom_Cmd(LCD_CLEAR);
    //
    Pwm_Init(3000);    // 3Khz
    Pwm_Change_DutyEx(1024 / 2);
    buzzer();
    //
    LED = OFF;
    RELAY = OFF;
    nowData = 0;
    maxData = 0;         // 0
    minData = 1000;         // 100
    aveData = 0;         // 0
    cnt = 0;
    sw3 = sw4 = OFF;
    flg = 0;
    //閾値のEEPROMからの取り込み
    thresholdData = Eeprom_Read(1);
    thresholdData = thresholdData << 8;
    thresholdData = thresholdData | Eeprom_Read(0);
    if ((thresholdData < -40) || (thresholdData > 99))
        thresholdData = 22;
    //
```

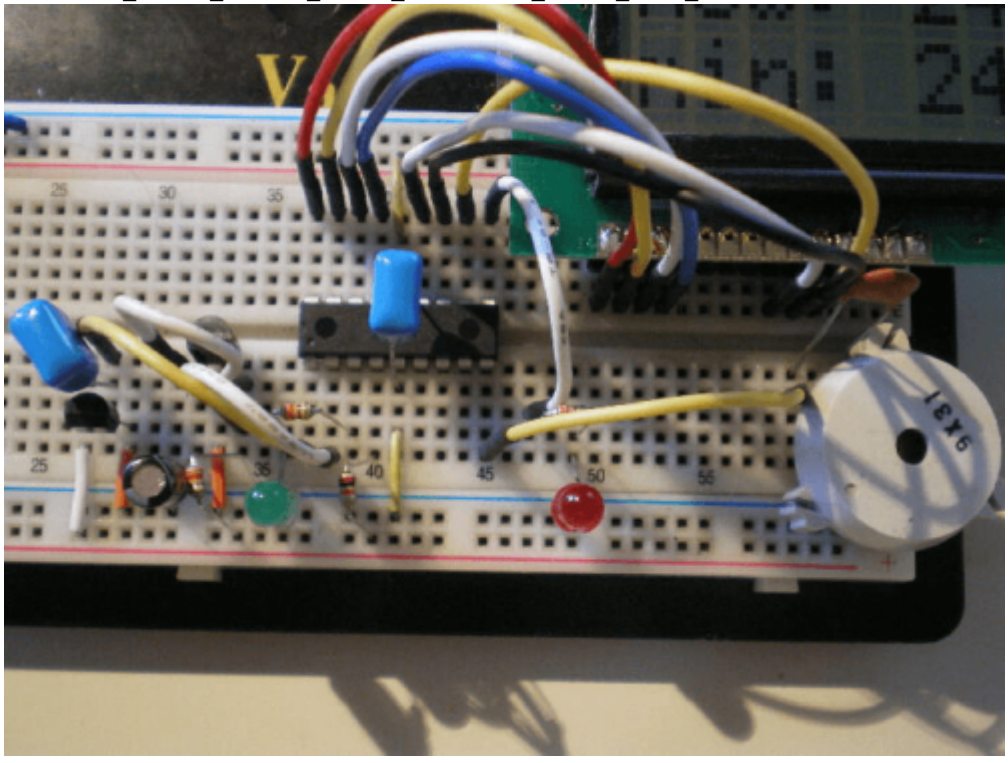
```
    INTCON.PEIE = 1;     //これ以降の処理で割り込みを許可する。
    INTCON.GIE = 1;          //これ以降の処理で割り込みを許可する。
    //
    while (1) {
        for (cnt = 0; cnt < 60; cnt++) {
            ad2 = Adc_Read(2);
            nowData = ((((double)ad2 * 2.4365234375) - 174.0) / 6.25) -
40.0;
            maxData = maxData < nowData ? nowData : maxData;
            minData = minData > nowData ? nowData : minData;
            aveData += nowData;
            //画面をクリアする。
            Lcd_Custom_Cmd(Lcd_Clear);
            //現在温度を表示する。
            IntToStr(nowData, buf);
            Lcd_Custom_Out(1, 1, "now:");
            Lcd_Custom_Out(1, 5, &buf[3]);
            //温度制御画面に切り替えるかをチェックする。
            if (SW1 == 0) {
                ThermoCntl();
                //
                Delay_ms(500);
                continue;
            } else {
                RELAY = OFF;
                T1CON.T1CKPS1 = 1;
                T1CON.T1CKPS0 = 1;
            }
            //最大温度を表示する。
            IntToStr(maxData, buf);
            Lcd_Custom_Out(2, 10, "max:");
            Lcd_Custom_Out(2, 14, &buf[3]);
            //最小温度を表示する。
            IntToStr(minData, buf);
            Lcd_Custom_Out(2, 1, "min:");
            Lcd_Custom_Out(2, 5, &buf[3]);
            //平均温度を表示する。
            if (flg == 0)
                tmp = aveData / (cnt + 1);
            else
                tmp = aveData / (cnt + 2);          //前回の平均温度を考慮する。
            IntToStr(tmp, buf);
            Lcd_Custom_Out(1, 10, "ave:");
            Lcd_Custom_Out(1, 14, &buf[3]);
            //
            Delay_ms(1000);
        }
        aveData = aveData / 60;          //前回の平均温度を考慮する。（引き継ぐ）
        flg = 1;
    }
}
```

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

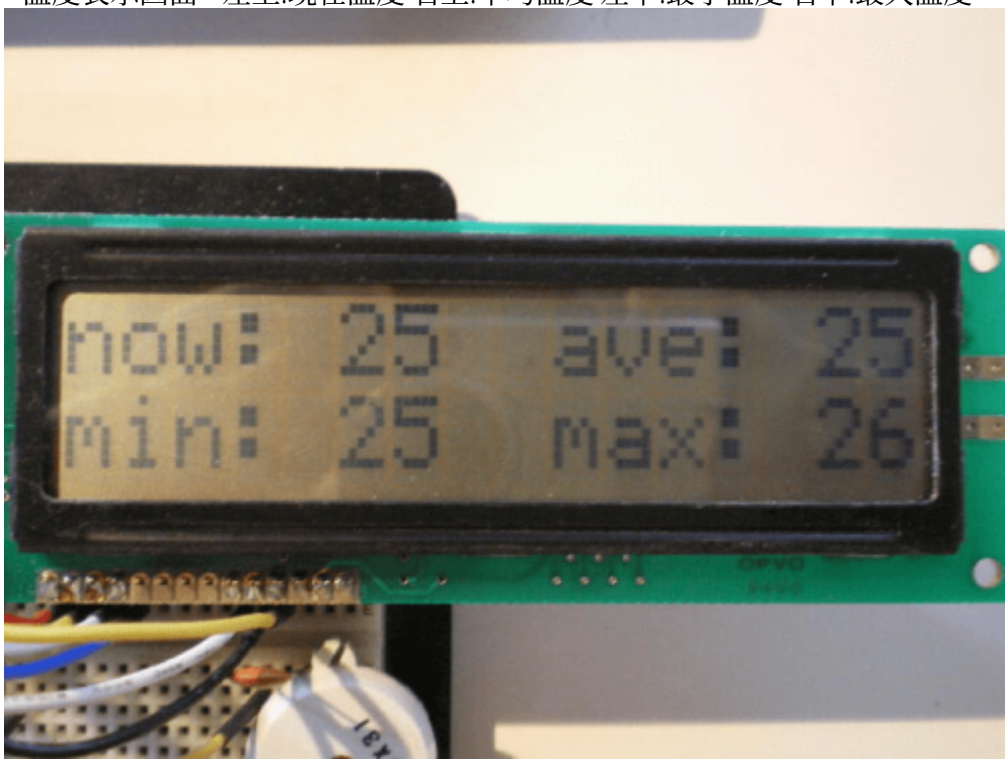
動作確認



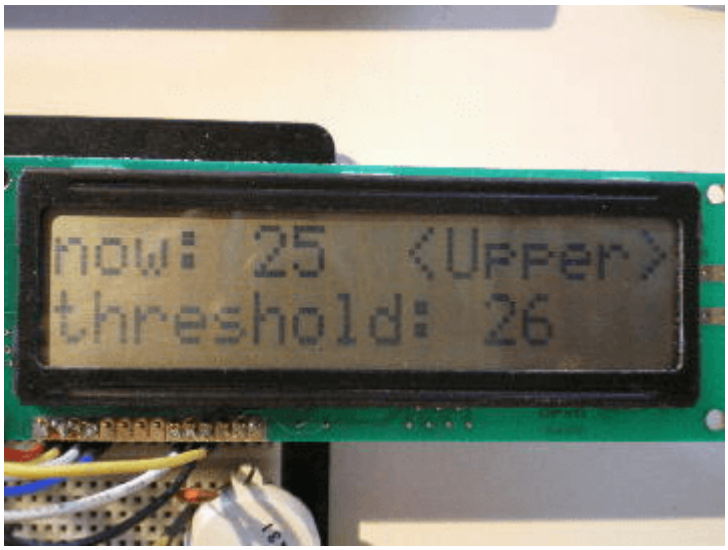
左側から□LM60□TL431□LED1□PIC16F88□LED2□LCD□圧電スピーカ

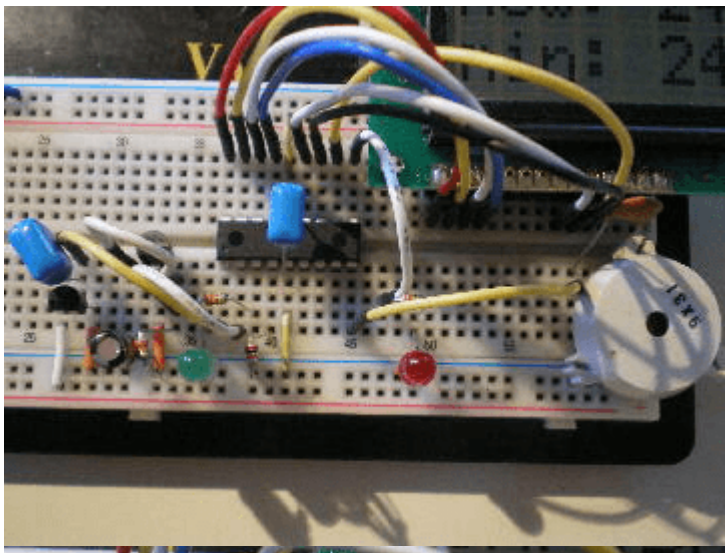


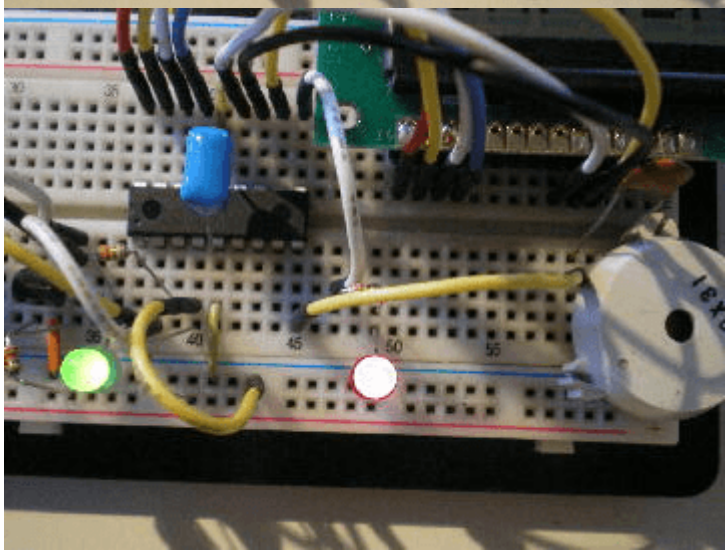
<温度表示画面> 左上:現在温度 右上:平均温度 左下:最小温度 右下:最大温度



<上限値制御> 左上:現在温度が25℃、閾値が26℃なので、出力はOFFになります。 左下:出力OFF□赤色LEDは消灯 右上:現在温度が26℃、閾値が26℃なので、出力はONになります。 右下:出力ON□赤色LEDは点灯







<下限値制御> 左上:現在温度が25℃、閾値が24℃なので、出力はOFFになります。 左下:出力OFF□赤色LEDは消灯 右上:現在温度が25℃、閾値が25℃なので、出力はONになります。 右下:出力ON□赤色LEDは点灯