```
1
2   /************************************************
3   * C source with CCS
4   *  File Name: 20120405_MK315_temperature_sensor.c
5   *  Description: check potentiometer and NTC sensor out
6   *  , and turn on yellow LED and relay
7   *  PIN A0 na
8   *  PIN A1 for potentiometer
9   *  PIN A2 for NTC
10  *  PIN A3 for start
11  *  PIN A4 for Relay
12  *  PIN A5 for LED
13  *  COPYRIGHT 2012 MYCOMKITS.COM, owned by CNET LIMITED
14  *  当プログラムの著作権は、製作者「マイコンキットドットコム運営  有限会社クネット」に帰属します。
15  *  著作権を放棄していませんが、当プログラムを使った学習の中でプログラムを自由に変更してお使いください。
16  ************************************************/
17  // include header file
18  #include   <12f683.h>
19  #DEVICE ADC=10
20  //#include <stdlib.h>
21
22  // settings
23  #fuses   INTRC_IO, NOWDT, NOPUT, NOPROTECT, NOMCLR
24  #use delay(CLOCK = 4000000)
25  //
26  long value_setting=512;
27  long value sensor=512;
28  long value s[17];
29  int hyste = 0;
30  int i;
31  int m;
32  int n;
33  float b const = 3452;
34  float Rvalue;
35  float log_data2;
36  float actual temp;
37  float sensor f;
38  float log_data;
39  float rt_temp;
40
41  //
42  // LOG calclation table
43  const float r value[20] = {109121.5436, 63185.33369, 37994.24658, 23635.92567, 15161.78685,
    10000, 6764.289855, 4682.171292,
44     3309.912751, 2385.430628, 1749.917183, 1304.847807, 987.7513058, 758.2094321, 589.5776639,
    463.9827971,
45      369.239262, 269.9123, 241.0803, 197.53};
46  const float log_value[20] = {11.60021762, 11.05382749, 10.54519002, 10.07052311, 9.626533518,
    9.210340372, 8.819412562,
47     8.451517233, 8.104677109, 7.777134944, 7.467323742, 7.17384169, 6.895430951, 6.630959643,
    6.379406457, 6.139847476,
48     5.911444841, 5.693437, 5.48513, 5.285891};
49  //
50  // prototyping
51  #separate
52  void get_sensor();
53  #separate
54  void get setting();
55  #separate
56  void initializing();
57  #separate
58  float log_calc();
59  //
60  //
61  // main
62  //
63  void main()
64  {
65     //
66     initializing();   //ADC port initialize
67     output_low(PIN_A4);   //active high, Relay off
68     output_high(PIN_A5);   //active low, LED off
69     //
70     // main loop
71     while(1)
72     {
73        if(input(PIN_A3)==1)
74        {
75           delay ms(50);   //check it again for chattaling
76           if(input(PIN_A3)==1)
```

```
            {
               //
               get_setting();
               get_sensor();
               //
               if(value_sensor > value_setting*0.9)
               {
                  output_low(PIN_A5);   // LED on
               }
               else
               {
                  output_high(PIN_A5);   // LED off
               }
               if(value_sensor > value_setting)
               {

                  if(hyste == 0)
                  {
                     output_high(PIN_A4);   // Relay on
                     hyste = 50;
                  }
               }
               //
               else if(hyste == 0)
               {
                  output_low(PIN_A4);   // Relay off
                  hyste = 50;
               }
            }
         }
      }
   }
   //
   //
   #separate
   void get_setting()
   {
      set_adc_channel(1);   //must wait 65u
      delay_us(100);
      value_setting=read_adc();   // 0 to 1024, 0.3V to 3.3V
   }
   //
   #separate
   void get_sensor()
   {
      // moving average
      int i;
      for (i=0; i<15; i++)
      {
         value_s[15-i] = value_s[14-i];
      }
      set_adc_channel(2);   //must wait 65u
      delay_us(100);
      value_s[0] = read_adc();   //0 to 1024, 0.3V to 5V
      //
      value_sensor = 0;
      for (i=0; i<16; i++)
      {
         value_sensor = value_sensor + value_s[i];
      }
      //
      rt_temp = value_sensor;   // 16 times of actual value, almost real register
      m=0;
      //
      while(rt_temp < r_value[m])
      {
         m++;
      }
      //
      log_data =
   ((rt_temp-r_value[m])/(r_value[m-1]-r_value[m]))*(log_value[m-1]-log_value[m])+log_value[m];
      //
      actual_temp = (1/((1/b_const)*(log_data-9.2103403)+0.003354))-273.15;
      value_sensor = (actual_temp-10)*10.72;   // adjust into 10 to 140 degree
      //
   }
   //
   // system intializing
   //
   #separate
```

```
156 void initializing()
157 {
158    //
159    SET_TRIS_A(0x0F);   //A0 to 3 are input, 4, 5 are output
160    //
161    // A/D converter initialize AN1 and AN2
162    setup_adc_ports(sAN1 | sAN2 | VSS_VDD);   // gnd to 5v
163    setup_adc(ADC_CLOCK_DIV_8);          // 2usec
164    //
165    // timer 1 intialize for hysteresis count down
166    //
167    setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
168    set_timer1(0xF4F0);               // timer 1 setting almost 5msec
169    enable_interrupts(INT_TIMER1);    // timer 1 enable
170    enable_interrupts(GLOBAL);        // global enable
171    //
172    //
173 }
174 //
175 // timer 1 inturruption
176 // hysteresis count down
177 //
178 #int_timer1
179 void isr1(void)
180 {
181    set_timer1(0xF4F0);              // timer 1 almost 5msec
182    //
183    if(hyste > 1) hyste = hyste - 1;
184    else hyste = 0;
185    //
186 }
187 //
188 // supply LOG data by table
189 #separate
190 void log_calc()
191 {
192    int m=0;
193    int n=0;
194    //float log_data;
195    //
196    while(rt_temp < r_value[m])
197    {
198       m++;
199       n=m;
200    }
201    log_data =
    ((rt_temp-r_value[m])/(r_value[m-1]-r_value[m]))*(log_value[m-1]-log_value[m])+log_value[m];
202    //
203    //return log_data;
204 }
205 //
206
```