

```

1
2  /*****
3  * C source with CCS
4  *   File Name: 20120405_MK314_fsr_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 fsr
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
21 // settings
22 #fuses INTRC IO, NOWDT, NOPUT, NOPROTECT, NOMCLR
23 #use delay(CLOCK = 4000000)
24 //
25 long value_setting=512;
26 long value_sensor=512;
27 long value_s[17];
28 int hyste = 0;
29 int i;
30 int m;
31 float log data;
32 float rt_temp;
33
34 //
35 // LOG calculation table
36 const float r value[23] = {23635.92567, 15161.78685, 10000.0, 6764.289855, 4682.171292,
37 3309.912751, 2385.430628, 1749.917183, 1304.847807, 987.7513058, 758.2094321, 589.5776639,
38 463.9827971, 369.239262,
39 269.9123, 241.0803, 197.53, 163.2266, 135.9523, 114.0936, 96.42013, 82.02448, 70.21388};
40 const float log value[23] = {10.07052311, 9.626533518, 9.210340372, 8.819412562,
41 8.451517233, 8.104677109, 7.777134944, 7.467323742, 7.17384169, 6.895430951, 6.630959643,
42 6.379406457, 6.139847476,
43 5.911444841, 5.693437, 5.48513, 5.285891, 5.095139, 4.912345, 4.73702, 4.568715, 4.407018,
44 4.251546};
45
46 //
47 // prototyping
48 #separate
49 void get_sensor();
50 #separate
51 void get_setting();
52 #separate
53 void initializing();
54 //
55 //
56 // main
57 //
58 void main()
59 {
60     //
61     initializing(); //ADC port initialize
62     output_low(PIN_A4); //active high, Relay off
63     output_high(PIN_A5); //active low, LED off
64     //
65     // main loop
66     while(1)
67     {
68         if(input(PIN_A3)==1)
69         {
70             delay_ms(50); //check it again for chattering
71             if(input(PIN_A3)==1)
72             {
73                 //
74                 get_setting();
75                 get_sensor();
76                 //
77                 if(value_sensor > value_setting*0.9)
78                 {
79                     output_low(PIN_A5); // LED on
80                 }
81                 else

```

```

78         {
79             output_high(PIN_A5);    // LED off
80         }
81         if(value_sensor > value_setting)
82         {
83
84             if(hyste == 0)
85             {
86                 output_high(PIN_A4);    // Relay on
87                 hyste = 50;
88             }
89         }
90         //else
91         else if(hyste == 0)
92         {
93             output_low(PIN_A4); // Relay off
94             hyste = 50;
95         }
96     }
97 }
98 }
99 }
100 //
101 //
102 #separate
103 void get_setting()
104 {
105     set_adc_channel(1); //must wait 65u
106     delay_us(100);
107     value_setting=read_adc();    // 0 to 1024, 0.0V to 5.0V
108 }
109 //
110 #separate
111 void get_sensor()
112 {
113     // moving average
114     int i;
115     for (i=15; i>0; i--)
116     {
117         value_s[i] = value_s[i-1];
118     }
119     set_adc_channel(2); //must wait 65u
120     delay_us(100);
121     value_s[0] = read_adc();    //0 to 1024, 0.3V to 4.6V
122     //
123     value_sensor = 0;
124     for (i=0; i<16; i++)
125     {
126         value_sensor = value_sensor + value_s[i];
127     }
128     //
129     rt_temp = value_sensor; // 16 times of actual value, almost twice of real register
130     //rt_temp = value_sensor / 16;
131     if(rt_temp > 16384.0 ) rt_temp = 16384.0;
132     //
133     m=0;
134     //
135     //
136     while(rt_temp < r_value[m])
137     {
138         m++;
139     }
140     //
141     log_data =
142     ((rt_temp-r_value[m])/(r_value[m-1]-r_value[m]))*(log_value[m-1]-log_value[m])+log_value[m];
143     if(log_data < 4.2515) log_data = 4.2515;
144     if(log_data > 10.0705) log_data = 10.0705;
145     //
146     log_data = (log_data-4.2515)*1024/5.82;
147     value_sensor = 1024-log_data;
148     //
149 }
150 //
151 // system intializing
152 //
153 #separate
154 void initializing()
155 {
156     //

```

```

157     SET_TRIS_A(0x0F);    //A0 to 3 are input, 4, 5 are output
158     //
159     // A/D converter initialize AN1 and AN2
160     setup_adc_ports(sAN1 | sAN2 | VSS_VDD); // gnd to 5v
161     setup_adc(ADC_CLOCK_DIV_8);           // 2usec
162     //
163     // timer 1 initialize for hysteresis count down
164     //
165     setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
166     set_timer1(0xF4F0);           // timer 1 setting almost 5msec
167     enable_interrupts(INT_TIMER1); // timer 1 enable
168     enable_interrupts(GLOBAL);    // global enable
169     //
170     //
171 }
172 //
173 // timer 1 inturruption
174 // hysteresis count down
175 //
176 #int_timer1
177 void isr1(void)
178 {
179     set_timer1(0xF4F0);           // timer 1 almost 5msec
180     //
181     if(hyste > 1) hyste = hyste - 1;
182     else hyste = 0;
183     //
184 }
185 //
186

```