

```

1
2  /*****
3  * C source with CCS
4  *   File Name: 20120405_MK313_distance_sensor.c
5  *   Description: check potentiometer and distance sensor out
6  *   , and turn on yellow LED and relay
7  *   PIN A0 na
8  *   PIN A1 for potentiometer
9  *   PIN A2 for sensor out
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 float log_data;
29 float rt_temp;
30 int hyste = 0;
31 int i, m;
32 //
33 // LOG calculation table
34 const float r_value[23] = {23635.92567, 15161.78685, 10000.0, 6764.289855, 4682.171292,
35 3309.912751, 2385.430628, 1749.917183, 1304.847807, 987.7513058, 758.2094321, 589.5776639,
36 463.9827971, 369.239262,
37 269.9123, 241.0803, 197.53, 163.2266, 135.9523, 114.0936, 96.42013, 82.02448, 70.21388};
38 const float log_value[23] = {10.07052311, 9.626533518, 9.210340372, 8.819412562,
39 8.451517233, 8.104677109, 7.777134944, 7.467323742, 7.17384169, 6.895430951, 6.630959643,
40 6.379406457, 6.139847476,
41 5.911444841, 5.693437, 5.48513, 5.285891, 5.095139, 4.912345, 4.73702, 4.568715, 4.407018,
42 4.251546};
43 //
44 // prototyping
45 #separate
46 void get_sensor();
47 #separate
48 void get_setting();
49 #separate
50 void initializing();
51 //
52 // main
53 void main()
54 {
55     //
56     initializing(); //ADC port initialize
57     output_low(PIN_A4); //active high, Relay off
58     output_high(PIN_A5); //active low, LED off
59     //
60     // main loop
61     while(1)
62     {
63         if(input(PIN_A3)==1)
64         {
65             delay_ms(50); //check it again for chattering
66             if(input(PIN_A3)==1)
67             {
68                 //
69                 get_setting();
70                 get_sensor();
71                 //
72                 if(value_sensor < value_setting*1.1)
73                 {
74                     output_low(PIN_A5); // LED on
75                 }
76                 else
77                 {
78                     output_high(PIN_A5); // LED off
79                 }
80             }
81         }
82     }
83 }

```

```

78         if(value_sensor < value_setting)
79         {
80
81             if(hyste == 0)
82             {
83                 output_high(PIN_A4);    // Relay on
84                 hyste = 50;
85             }
86         }
87         //else
88         else if(hyste == 0)
89         {
90             output_low(PIN_A4); // Relay off
91             hyste = 50;
92         }
93     }
94 }
95
96 //
97 //
98 //
99 #separate
100 void get_setting()
101 {
102     set_adc_channel(1); //must wait 65u
103     delay_us(100);
104     value_setting=read_adc();    // 0 to 1024, 0V to 5V
105 }
106 //
107 #separate
108 void get_sensor()
109 {
110     // moving average
111     int i;
112     for (i=15; i>0; i--)
113     {
114         value_s[i] = value_s[i-1];
115     }
116     set_adc_channel(2); //must wait 65u
117     delay_us(100);
118     value_s[0] = read_adc();    //0 to 1024, 0.3V to 3.3V
119     //
120     value_sensor = 0;
121     for (i=0; i<16; i++)
122     {
123         value_sensor = value_sensor + value_s[i];    // max 16384
124     }
125     //value_sensor = value_sensor / 16; // actual value
126     //
127     rt_temp = value_sensor; // 16 times of actual value, almost twice of real register
128     //rt_temp = 1.57 * rt_temp; // adjust for IR sensor
129     if(rt_temp > 16384.0 ) rt_temp = 16384.0;
130     //
131     m=0;
132     //
133     while(rt_temp < r_value[m])
134     {
135         m++;
136     }
137     //
138     log_data =
139     ((rt_temp-r_value[m])/(r_value[m-1]-r_value[m]))*(log_value[m-1]-log_value[m])+log_value[m];
140     //if(log_data < 6.8954) log_data = 6.8954;
141     //if(log_data > 9.6265) log_data = 9.6265;
142     //
143     log_data = 1024-((log_data-6.8954)*1024/2.71);
144     if(log_data < 0) value_sensor = 0;
145     else if(log_data >1024) value_sensor = 1024;
146     else value_sensor = log_data;
147 }
148 //
149 //
150 #separate
151 void initializing()
152 {
153     //
154     SET_TRIS_A(0x0F);    //A0 to 3 are input, 4, 5 are output
155     //
156     // A/D converter initialize AN1 and AN2

```

```

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

```