```
#include "range_finder.h"
#include "display.h"
#include "switch.h"
#include "to_7seg.h"
#include "control.h"

typedef enum {IDLE, COUNT, WAIT, MEAS} control_estado;
static control_estado estado;

bool volatile gb_control_can_sleep;

static bool volatile gb_control_initd;

uint8_t cnt;
uint16_t dist;

static BusOut *g_leds;
static AnalogIn *g_lit;

static Timeout to;

static bool volatile to_evnt;

static void to_isr (void){
  to_evnt = true;
}




void control_init(BusOut *leds, AnalogIn *lit){
  if(!gb_control_initd){
    gb_control_initd = true;

    estado = IDLE;

    cnt = 0;
    dist = 0;

    g_leds = leds;
    g_lit = lit;

    to_evnt = false;


  }
}

void control_fsm(void){
  if(gb_control_initd){

    switch(estado){

      case COUNT:

          to_evnt = false;

          if(gb_swm_long_msg){

            if(cnt == 0){
              gb_display_off_msg = true;
              g_display_segs = 0;
              estado = IDLE;

            }else{
              to.attach_us(to_isr,1000000);
              estado = WAIT;
            }

          }else if(gb_swm_msg){
            gb_swm_msg = false;
            cnt = (cnt < 5) ? (cnt+1) : 0;
           // gb_display_update_msg = true;
            g_display_segs = (0x54 << 8) | to_7seg(cnt);

          }else{
              //nothing
          }

      break;

      case  WAIT:

          gb_swm_msg = false;
```

```
85              gb_swm_long_msg = false;
86
87              if(to_evnt){
88                to_evnt = false;
89                to.detach();
90
91                if(cnt == 0){
92                  gb_display_update_msg = true;
93                  g_display_segs = 0x543F;
94                  *g_leds = 0;
95                  estado = COUNT;
96
97                }else{
98                  gb_rf_start_msg = true;
99                  estado = MEAS;
100               }
101
102             }
103         break;
104
105         case MEAS:
106
107              gb_swm_msg = false;
108              gb_swm_long_msg = false;
109              to_evnt = false;
110
111
112             if(gb_rf_done_msg){
113
114               to.attach_us(to_isr,1000000);
115
116               if(-1 == g_rf_range_cm){
117                 dist = 0x7950;
118                  g_display_segs = dist;
119                 *g_leds = 7;
120
121               }else if(g_rf_range_cm  > 99){
122
123                 dist = 0x4040;
124                 g_display_segs = dist;
125                 *g_leds = 5;
126
127               }else if(g_rf_range_cm  <= 33){
128                 *g_leds = 4;
129                 dist = g_rf_range_cm;
130                  g_display_segs = (to_7seg(dist/10) << 8) | to_7seg(dist%10);
131
132               }else if(g_rf_range_cm  <= 66 && g_rf_range_cm  >= 34 ){
133                 *g_leds = 2;
134                 dist = g_rf_range_cm;
135                  g_display_segs = (to_7seg(dist/10) << 8) | to_7seg(dist%10);
136
137               }else if(g_rf_range_cm  <= 99 && g_rf_range_cm  >= 67 ){
138                 *g_leds = 1;
139                 dist = g_rf_range_cm;
140                  g_display_segs = (to_7seg(dist/10) << 8) | to_7seg(dist%10);
141               }
142
143
144
145               // gb_display_update_msg = true;
146
147               cnt--;
148               estado = WAIT;
149
150             }
151           break;
152
153           default:    //IDLE
154               gb_swm_msg = false; //irrelevante
155               to_evnt = false;
156
157           if(gb_swm_long_msg){
158             gb_swm_long_msg = false;
159             gb_display_on_msg = true;
160             //gb_display_update_msg = true;
161             gb_display_brightness_msg = 100;
162             g_display_segs = 0x543F;
163             g_display_brightness = 100;
164             estado = COUNT;
165           }
166
167           break;
168
```

```
169       }
170       __disable_irq();
171       if(!to_evnt && !gb_swm_long_msg && !gb_swm_msg && !gb_display_update_msg &&
      !gb_rf_done_msg && !gb_rf_start_msg){
172         gb_control_can_sleep = true;
173       }
174       __enable_irq();
175     }
176   }
177
178
179
```