```
#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                  estado = COUNT;
95
96                }else{
97                  gb_rf_start_msg = true;
98                  estado = MEAS;
99                }
100
101             }
102         break;
103
104         case MEAS:
105
106              gb_swm_msg = false;
107              gb_swm_long_msg = false;
108              to_evnt = false;
109
110
111             if(gb_rf_done_msg){
112
113               to.attach_us(to_isr,1000000);
114
115               if(-1 == g_rf_range_cm){
116                 dist = 0x7950;
117                  g_display_segs = dist;
118
119               }else if(g_rf_range_cm > 99){
120                 dist = 0x4040;
121                 g_display_segs = dist;
122
123               }else{
124                 dist = g_rf_range_cm;
125                  g_display_segs = (to_7seg(dist/10) << 8) | to_7seg(dist%10);
126               }
127
128               // gb_display_update_msg = true;
129
130               cnt--;
131               estado = WAIT;
132
133             }
134           break;
135
136           default:    //IDLE
137               gb_swm_msg = false; //irrelevante
138               to_evnt = false;
139
140             if(gb_swm_long_msg){
141               gb_swm_long_msg = false;
142               gb_display_on_msg = true;
143               //gb_display_update_msg = true;
144               gb_display_brightness_msg = 100;
145               g_display_segs = 0x543F;
146               g_display_brightness = 100;
147               estado = COUNT;
148             }
149
150           break;
151
152       }
153       __disable_irq();
154       if(!to_evnt && !gb_swm_long_msg && !gb_swm_msg && !gb_display_update_msg &&
      !gb_rf_done_msg && !gb_rf_start_msg){
155         gb_control_can_sleep = true;
156       }
157       __enable_irq();
158     }
159   }
160
161
162
```