```
#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;
uint16_t luz;

static BusOut *g_leds;
static AnalogIn *g_lit;

static Timeout to;
static Ticker tick;

static bool volatile to_evnt;
static bool volatile tick_evnt;

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

static void tick_isr(void){
  tick_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;
              tick.detach();
              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 if(tick_evnt){
```

```
85                tick_evnt = false;
86                luz = g_lit -> read_u16()/656;
87                g_display_brightness = 0.39*luz+1;
88
89              }else{
90                //nothing
91              }
92
93          break;
94
95          case  WAIT:
96
97              gb_swm_msg = false;
98              gb_swm_long_msg = false;
99
100             if(to_evnt){
101               to_evnt = false;
102               to.detach();
103
104               if(cnt == 0){
105                 gb_display_update_msg = true;
106                 g_display_segs = 0x543F;
107                 *g_leds = 0;
108                 estado = COUNT;
109
110               }else{
111                 gb_rf_start_msg = true;
112                 estado = MEAS;
113               }
114
115             }else if(tick_evnt){
116                tick_evnt = false;
117                luz = g_lit -> read_u16()/656;
118                g_display_brightness = 0.39*luz+1;
119
120             }else{
121               //nothing
122             }
123         break;
124
125         case MEAS:
126
127              gb_swm_msg = false;
128              gb_swm_long_msg = false;
129              to_evnt = false;
130
131
132             if(gb_rf_done_msg){
133
134               to.attach_us(to_isr,1000000);
135
136               if(-1 == g_rf_range_cm){
137                 dist = 0x7950;
138                  g_display_segs = dist;
139                 *g_leds = 7;
140
141               }else if(g_rf_range_cm  > 99){
142
143                 dist = 0x4040;
144                 g_display_segs = dist;
145                 *g_leds = 5;
146
147               }else if(g_rf_range_cm  <= 33){
148                 *g_leds = 4;
149                 dist = g_rf_range_cm;
150                  g_display_segs = (to_7seg(dist/10) << 8) | to_7seg(dist%10);
151
152               }else if(g_rf_range_cm  <= 66 && g_rf_range_cm  >= 34 ){
153                 *g_leds = 2;
154                 dist = g_rf_range_cm;
155                  g_display_segs = (to_7seg(dist/10) << 8) | to_7seg(dist%10);
156
157               }else if(g_rf_range_cm  <= 99 && g_rf_range_cm  >= 67 ){
158                 *g_leds = 1;
159                 dist = g_rf_range_cm;
160                  g_display_segs = (to_7seg(dist/10) << 8) | to_7seg(dist%10);
161               }
162
163
164
165               // gb_display_update_msg = true;
166
167               cnt--;
168               estado = WAIT;
```

```
169
170             }else if(tick_evnt){
171                tick_evnt = false;
172                luz = g_lit -> read_u16()/656;
173                g_display_brightness = 0.39*luz+1;
174
175             }else{
176               //nothing
177             }
178
179           break;
180
181           default:    //IDLE
182               gb_swm_msg = false; //irrelevante
183               to_evnt = false;
184
185             if(gb_swm_long_msg){
186               gb_swm_long_msg = false;
187               gb_display_on_msg = true;
188               //gb_display_update_msg = true;
189               tick.attach_us(tick_isr,10000);
190               gb_display_brightness_msg = 100;
191               g_display_segs = 0x543F;
192               g_display_brightness = 100;
193               estado = COUNT;
194             }
195
196           break;
197
198       }
199       __disable_irq();
200       if(!to_evnt && !gb_swm_long_msg && !gb_swm_msg && !gb_display_update_msg &&
    !gb_rf_done_msg && !gb_rf_start_msg){
201         gb_control_can_sleep = true;
202       }
203       __enable_irq();
204     }
205   }
206
207
208
```