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

/// extended state ----------------------------------------------------
  // "basic" state
  typedef enum {CT_IDLE, CT_MENU, CT_Li, CT_di, CT_LE, CT_OFF} control_state_t;
  static control_state_t g_control_state;

  // output messages
  bool volatile gb_control_can_sleep;  //this FSM can sleep

  // hardware resources
 static BusOut *g_leds;
 static AnalogIn *g_LIT;

static Ticker         tick_100ms;     // tiempo para refrescar la luz del LIT
static Timeout        tout_50ms_ldr;
static Timeout        tout_50ms_ldm;
static Timeout        tout_50ms_ldl;

 // internal objects
static bool volatile  gb_control_initd;            // true after call to dp_init()
uint8_t contador;


uint16_t luz;
static bool volatile tick_100ms_evnt;
static bool volatile tout_ldl_evnt;
static bool volatile tout_ldr_evnt;
static bool volatile tout_ldm_evnt;
 // end of extended state  -------------------------------------------

// ISRs ---------------------------------------------------------------
static void tick_100ms_isr(void){
  tick_100ms_evnt = true;
}

static void tout_ldr_isr(void){
  tout_ldr_evnt = true;
}

static void tout_ldl_isr(void){
  tout_ldl_evnt = true;
}

static void tout_ldm_isr(void){
  tout_ldm_evnt = true;
}

 // FSM  ---------------------------------------------------------------
void control_fsm(void){

  if(gb_control_initd){

    switch (g_control_state){

      case CT_MENU:

          if(tout_ldl_evnt){
                tout_ldl_evnt = false;
                *g_leds = 0;
           }

          if(gb_swm_long_msg){
             gb_swm_long_msg = false;
             *g_leds = 4;
               tout_50ms_ldl.attach_us(tout_ldl_isr, 50000);

             if(contador == 1){
               g_control_state = CT_Li;

             }else if(contador == 2){
               g_control_state = CT_di;

             }else if(contador == 3){
               g_control_state = CT_LE;

             }else if(contador == 4){
               g_control_state = CT_OFF;
             }else{
             }
```

```
85
86              }
87
88            if(gb_swm_msg){
89              gb_swm_msg = false;
90              gb_display_on_msg = true;
91              g_display_brightness = 40;
92              *g_leds = 1;
93              tout_50ms_ldr.attach_us(tout_ldr_isr, 50000);
94              contador = (contador >= 4) ? 1 : (contador+1);
95
96               if(contador == 1){
97                   g_display_segs = 0x3810;
98
99               }
100              if(contador == 2){
101                  g_display_segs = 0x5E10;
102
103              }
104              if(contador == 3){
105                 g_display_segs = 0x3879;
106
107              }
108              if(contador == 4){
109                  g_display_segs = 0x3F71;
110              }else{
111              }
112
113           }
114
115           if(tout_ldr_evnt){
116                tout_ldr_evnt = false;
117                *g_leds = 0;
118           }
119
120       break;
121
122
123       case CT_Li:
124
125                if(tout_ldl_evnt){
126                  tout_ldl_evnt = false;
127                  *g_leds = 0;
128                }
129
130              if(tick_100ms_evnt){
131                tick_100ms_evnt = false;
132                gb_display_on_msg = true;
133                luz = g_LIT -> read_u16()/656;
134                g_display_segs = ( to_7seg(luz/10)<< 8 ) | to_7seg(luz%10);
135                g_display_brightness = 0.39*luz+1;
136
137              }
138
139              if(gb_swm_long_msg){
140                gb_swm_long_msg = false;
141                gb_display_on_msg = true;
142                *g_leds = 4;
143                 tout_50ms_ldl.attach_us(tout_ldl_isr, 50000);
144
145                g_display_segs = 0x3810;       // display Li
146                g_display_brightness = 40;
147                g_control_state = CT_MENU;
148              }
149
150              if(gb_swm_msg){
151                  gb_swm_msg = false;
152                  *g_leds = 1;
153                  tout_50ms_ldr.attach_us(tout_ldr_isr, 50000);
154
155             }
156              if(tout_ldr_evnt){
157                     tout_ldr_evnt = false;
158                     *g_leds = 0;
159             }
160
161       break;
162
163
164       case CT_di:
165
166             if(tout_ldl_evnt){
167                  tout_ldl_evnt = false;
168                  *g_leds = 0;
```

```
169               }
170
171               if(tick_100ms_evnt){
172                 tick_100ms_evnt = false;
173                 gb_rf_start_msg = true;
174
175               }else if (gb_rf_done_msg) {
176                 gb_rf_done_msg = false;
177                 gb_display_on_msg = true;
178                 g_display_segs = (to_7seg(g_rf_range_cm/10) << 8)| to_7seg(g_rf_range_cm%10);
179                 g_display_brightness = 0.4*(g_rf_range_cm) + 0.01*(99 - g_rf_range_cm);
180
181
182               }
183
184               if(gb_swm_long_msg){
185                 gb_swm_long_msg = false;
186                 gb_display_on_msg = true;
187                 *g_leds = 4;
188                 tout_50ms_ldl.attach_us(tout_ldl_isr, 50000);
189
190                 g_display_segs = 0x5E10;        // display di
191                 g_display_brightness = 40;
192                 g_control_state = CT_MENU;
193               }
194
195               if(gb_swm_msg){
196                 gb_swm_msg = false;
197                 *g_leds = 1;
198                 tout_50ms_ldr.attach_us(tout_ldr_isr, 50000);
199
200               }
201                 if(tout_ldr_evnt){
202                  tout_ldr_evnt = false;
203                  *g_leds = 0;
204
205                 }
206
207           break;
208
209         case CT_LE:
210
211                  if(tout_ldl_evnt){
212                    tout_ldl_evnt = false;
213                    *g_leds = 0;
214                  }
215
216                  g_display_segs = 0x4040;
217                  g_display_brightness = 40;
218
219                  if(tick_100ms_evnt){
220                     tick_100ms_evnt = false;
221                     tout_50ms_ldm.attach_us(tout_ldm_isr,50000);
222                     *g_leds = 2;
223
224                   }
225
226                   if(tout_ldm_evnt){
227                     tout_ldm_evnt = false;
228                     *g_leds = 0;
229
230                    }
231
232                   if(gb_swm_long_msg){
233                      gb_swm_long_msg = false;
234                      gb_display_on_msg = true;
235                      *g_leds = 4;
236                      tout_50ms_ldl.attach_us(tout_ldl_isr, 50000);
237
238                      g_display_segs = 0x3879;        // display LE
239                      g_display_brightness = 40;
240                      g_control_state = CT_MENU;
241                   }
242
243                   if(gb_swm_msg){
244                     gb_swm_msg = false;
245                     *g_leds = 1;
246                     tout_50ms_ldr.attach_us(tout_ldr_isr, 50000);
247
248                   }
249
250                   if(tout_ldr_evnt){
251                      tout_ldr_evnt = false;
252                      *g_leds = 0;
```

```
253
254                 }
255
256
257             break;
258
259         case CT_OFF:
260
261                 gb_display_off_msg = true;
262                 tout_50ms_ldr.detach();
263                 tout_50ms_ldl.detach();
264                 tout_50ms_ldm.detach();
265                 tick_100ms.detach();
266                 *g_leds = 0;
267                 contador = 0;
268                g_control_state = CT_IDLE;
269
270               break;
271
272         default:                // CT_IDLE
273             tick_100ms_evnt = false;
274
275             if(gb_swm_long_msg){
276               gb_swm_long_msg = false;
277               gb_display_on_msg = true;
278               *g_leds = 4;
279
280               g_display_segs = 0x3810;           // Li
281               g_display_brightness = 40;        // maximo  brillo
282               tout_50ms_ldl.attach_us(tout_ldl_isr, 50000);
283               tick_100ms.attach_us(tick_100ms_isr,100000);
284
285              contador++;
286              g_control_state = CT_MENU;
287             }
288
289
290       } // switch
291
292       __disable_irq();
293       if( !tout_ldm_evnt && !tout_ldl_evnt && !tout_ldr_evnt && !tick_100ms_evnt ) {
294         gb_control_can_sleep = true;
295       }
296       __enable_irq();
297     } // if (gb_sf_initd)
298   }
299   // end of FSM ----------------------------------------------------------
300
301   // initialize FSM machinery  -------------------------------------------
302   void control_init(BusOut *leds, AnalogIn *lit){
303     if (!gb_control_initd) {
304       gb_control_initd = true;    // protect against multiple calls to sf_init
305
306       // initialize state
307       g_control_state = CT_IDLE;
308
309       // initial actions
310       contador = 0;
311
312       g_leds = leds;
313       g_LIT = lit;
314
315       tick_100ms_evnt = false;
316       tout_ldl_evnt = false;
317       tout_ldr_evnt = false;
318       tout_ldm_evnt = false;
319
320       luz = 0;
321     }
322   }
323   // end of FSM initialization  ------------------------------------------
324
```