

```

1  #include "mbed.h"
2  #include "pinout.h"
3  #include "to_7seg.h"
4
5  // seven segment display anodes
6  static BusOut display ( SGA_PIN, SGB_PIN, SGC_PIN, SGD_PIN, SGE_PIN, SGF_PIN, SGG_PIN);
7
8  // display cathodes
9  static PwmOut dsr (DSR_PIN);
10 static PwmOut dsl (DSL_PIN);
11
12 //BRILLO LIT
13 static AnalogIn LIT(LIT_PIN);
14
15 // tiempo de multiplexacion
16 static Ticker tick_4ms; // frecuencia de multiplexacion de 250Hz
17 static bool volatile t_4ms; //Variable para contar el tiempo, frecuencia multiplexación
18
19 static void isr_4ms (void) { //Función para poner el contador de tiempos a true
20     t_4ms = true;
21 }
22
23 //tiempo de intensidad de los displays
24 static Ticker tick_333ms;
25 static bool volatile t_333ms;
26
27 static void isr_333ms (void){
28     t_333ms = true;
29 }
30
31 //switch
32 static InterruptIn swr (SWR_PIN);
33 static InterruptIn swl (SWL_PIN);
34 static InterruptIn swm (SWM_PIN);
35
36 //switch management
37
38 //SWR_PIN
39 static bool volatile swr_fall_evnt; //interrupcion
40 static void swr_fall_isr(void){
41     swr_fall_evnt = true;
42 }
43
44 //TIMEOUT SWR
45 static Timeout time_down_swr; //Variable para generar el Timeout para verificar que se ha
46 // pulsado el pulsador
47 static bool volatile t_down_swr;
48 static void down_swr_isr(void){
49     t_down_swr = true;
50 }
51
52 //SWL_PIN
53 static bool volatile swl_fall_evnt;
54 static void swl_fall_isr(void){
55     swl_fall_evnt = true;
56 }
57
58 //TIMEOUT SWL
59 static Timeout time_down_swl; //Variable para generar el Timeout para verificar que
60 // se ha pulsado el pulsador
61 static bool volatile t_down_swl;
62 static void down_swl_isr(void){
63     t_down_swl = true;
64 }
65
66 //SWM_PIN
67 static bool volatile swm_fall_evnt;
68 static void swm_fall_isr(void){
69     swm_fall_evnt = true;
70 }
71
72 //TIMEOUT SWM
73 static Timeout time_down_swm;
74 static bool volatile t_down_swm;
75 static void down_swm_isr(void){
76     t_down_swm = true;
77 }
78
79
80 int main(void) {
81     int8_t cuenta = 0;
82     bool sentido = false; // variable que pone a dsl o dsr encendido

```

```

83
84     tick_4ms.attach_us(isr_4ms, 4000); // 4000us = 250Hz
85                                     // //Esta función lo que hace es iniciar un timeout de
86                                     // 5ms para comprobar que se ha pulsado el
87 pulsador, saltando a la función swr_fall_isr
88     tick_333ms.attach_us(isr_333ms, 333333); //tiempo para refrescar la intensidad de los
displays
89
90     //BRILLO DEL DISPLAY PROPORCIONAL Vlit
91
92     uint16_t luz = LIT.read_ul6()/656;
93     uint16_t t_on_disp = 0.39*luz+1;
94
95     //periodo de la señal
96     dsr.period_us(40);
97     dsl.period_us(40);
98
99 //proceso de pulsar
100 swr.mode(PullUp);
101 swr.fall(swr_fall_isr);
102
103 swl.mode(PullUp);
104 swl.fall(swl_fall_isr);
105
106 swm.mode(PullUp);
107 swm.fall(swm_fall_isr);
108
109 dsl = 1;
110 dsr = 1;
111 display = to_7seg(cuenta);
112
113
114 for (;;) {
115
116     if(swr_fall_evnt){
117         swr_fall_evnt = false;
118         time_down_swr.attach_us(down_swr_isr,4000);
119     }
120
121     if(swl_fall_evnt){
122         swl_fall_evnt = false;
123         time_down_swl.attach_us(down_swl_isr,4000);
124     }
125
126     if(swm_fall_evnt){
127         swm_fall_evnt = false;
128         time_down_swm.attach_us(down_swm_isr,4000);
129     }
130
131     if(t_4ms){ //Representación del contador en los displays con una frecuencia de
multiplexación de 4ms para
// poder ver cada display con un numero distinto
132         t_4ms = false;
133         sentido = !sentido;
134         if(sentido){
135             dsr = 0;
136             dsl = 1;
137             display = to_7seg(cuenta/10);
138             dsl.pulsewidth_us(t_on_disp);
139
140         }else{
141             dsr = 1;
142             dsl = 0;
143             display = to_7seg(cuenta%10);
144             dsr.pulsewidth_us(t_on_disp);
145         }
146
147         // to_7seg(cuenta/10) calculo de las decenas del display 1/10 =
0, 12/10 = 1
148         // to_7seg(cuenta%10) calculo de las unidades del display 8%10 = 8,
12%10 = 2 (resto de la division)
149
150
151         if(t_333ms){
152             t_333ms = false;
153             luz=LIT.read_ul6()/656;
154             t_on_disp = 0.39*luz+1;
155         }
156
157         if(t_down_swr){
158             t_down_swr = false;
159
160             if(swr==0){
161                 cuenta++;

```

```

162
163         if(cuenta > 99)
164             cuenta = 0;
165     }
166 }
167
168     if(t_down_sw1){
169         t_down_sw1 = false;
170
171         if(sw1==0){
172             cuenta--;
173             if(cuenta < 0){
174                 cuenta = 99;
175             }
176         }
177     }
178
179     if(t_down_swm){
180         t_down_swm = false;
181
182         if(swm==0)
183             cuenta = 99 - cuenta;
184     }
185
186     __disable_irq();
187     if(!t_4ms && !t_down_swr && !t_down_sw1 && !t_down_swm && !swr_fall_evnt &&
!swl_fall_evnt && !swm_fall_evnt){ //Cada vez que no pasen los 4ms o no se pulse el
sistema estará dormido
        __WFI();
    }
    __enable_irq();
191
192 } // forever
193 } // main()
194

```