

```

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 DigitalOut dsr (DSR_PIN);
10 static DigitalOut dsl (DSL_PIN);
11
12 // mux stuff
13 static Ticker tick_4ms; // frecuencia de multiplexacion de 250Hz
14 static bool volatile t_4ms; //Variable para controlar el tiempo, frecuencia multiplexación
15
16 static void isr_4ms (void) { //Función para poner el contador de tiempos a true
17     t_4ms = true;
18 }
19
20 //timeout
21 static Timeout t_down_swr; //Variable para generar el Timeout para verificar que se ha
22 // pulsado el pulsador
23 static Timeout t_down_sw1; //Variable para generar el Timeout para verificar que se ha
24 // pulsado el pulsador
25 static Timeout t_down_swm; //Variable para generar el Timeout para verificar que se ha
26 // pulsado el pulsador
27
28 //switch
29 static InterruptIn swr (SWR_PIN);
30 static InterruptIn sw1 (SWL_PIN);
31 static InterruptIn swm (SWM_PIN);
32
33 //switch management
34
35 static bool volatile swr_fall_evnt;
36 static bool volatile sw1_fall_evnt;
37 static bool volatile swm_fall_evnt;
38
39 //SWR_PIN
40 static void swr_fall_isr(void){
41     swr_fall_evnt = true;
42 }
43
44 static void time_down_swr(void){
45     t_down_swr.attach_us(swr_fall_isr, 5000);
46 }
47
48 //SWL_PIN
49 static void sw1_fall_isr(void){
50     sw1_fall_evnt = true;
51 }
52
53 static void time_down_sw1(void){
54     t_down_sw1.attach_us(sw1_fall_isr, 5000);
55 }
56
57 //SWM_PIN
58 static void swm_fall_isr(void){
59     swm_fall_evnt = true;
60 }
61
62 static void time_down_swm(void){
63     t_down_swm.attach_us(swm_fall_isr, 5000);
64 }
65
66 int main(void) {
67     int8_t cuenta = 0;
68     bool sentido = false; // variable que pone a dsl o dsr encendido
69
70     tick_4ms.attach_us(isr_4ms, 4000); // 4000us = 250Hz
71     // //Esta función lo que hace es iniciar un timeout de
72     // 5ms para comprobar que se ha pulsado el
73     // pulsador, saltando a la función swr_fall_isr
74
75     //proceso de pulsar
76     swr.mode(PullUp);
77     swr.fall(time_down_swr);
78
79     sw1.mode(PullUp);
80     sw1.fall(time_down_sw1);
81
82     swm.mode(PullUp);
83     swm.fall(time_down_swm);
84
85     dsl = 1;

```

```

81     dsr = 1;
82     display = to_7seg(cuenta);
83
84
85     for (;;) {
86
87
88         if(t_4ms){ //Representación del contador en los displays con una frecuencia de
multiplexación de 4ms para
89             // poder ver cada display con un numero distinto
90             t_4ms = false;
91             sentido = !sentido;
92             dsl = sentido;
93             dsr = !sentido;
94
95             display = to_7seg(sentido ? cuenta/10 : cuenta%10); // to_7seg(cuenta/10) calculo
de las decenas del display 1/10 = 0, 12/10 = 1
96                                     // to_7seg(cuenta%10)
calcula de las unidades del display 8%10 = 8, 12%10 = 2 (resto de la division)
97         }
98
99         if(swr_fall_evnt){
100             swr_fall_evnt = false;
101
102             if(swr==0){
103                 cuenta++;
104
105                 if(cuenta > 99)
106                     cuenta = 0;
107             }
108         }
109
110         if(swl_fall_evnt){
111             swl_fall_evnt = false;
112
113             if(swl==0){
114                 cuenta--;
115                 if(cuenta < 0){
116                     cuenta = 99;
117                 }
118             }
119         }
120
121         if(swm_fall_evnt){
122             swm_fall_evnt = false;
123
124             if(swm==0)
125                 cuenta = 99 - cuenta;
126         }
127
128         __disable_irq();
129         if(!t_4ms && !swr_fall_evnt && !swl_fall_evnt && !swm_fall_evnt){ //Cada vez que no
pasen los 4ms o no se pulse el sistema estará dormido
130             __WFI();
131         }
132         __enable_irq();
133
134     } // forever
135 } // main()
136

```