

```

1  #include "mbed.h"
2  #include "pinout.h"
3  #include "to_7seg.h"
4
5  #if 1
6  #define VERBOSE
7  #endif
8
9  // seven segment display anodes
10 // when in a int8_t, they are 0b-GFEDCBA
11 BusOut      g_seven_seg(SGA_PIN, SGB_PIN, SGC_PIN, SGD_PIN,
12                        SGE_PIN, SGF_PIN, SGG_PIN);
13
14 // display cathodes
15 DigitalOut   g_dsr(DSR_PIN);
16 DigitalOut   g_dsl(DSL_PIN);
17
18 // leds
19 BusOut      g_leds(LDR_PIN, LDM_PIN, LDL_PIN);
20
21
22 //DISPLAY
23 //CAMBIO DE CUENTA cada 1.2s
24 static Ticker tick_1200ms;
25 static bool volatile tick_1200ms_evnt;
26 static void tick_1200ms_isr (void){
27     tick_1200ms_evnt = true;
28 }
29
30 //Tiempo de multiplexacion
31 static Ticker tick_4ms;
32 static bool volatile tick_4ms_evnt;
33 static void tick_4ms_isr (void){
34     tick_4ms_evnt = true;
35 }
36
37
38 //LED
39 //PULSADOR
40 static InterruptIn swr(SWR_PIN);
41 static bool volatile swr_fall_evnt;
42 static void swr_fall_isr(void){
43     swr_fall_evnt = true;
44 }
45
46
47 //REBOTES PULSADOR
48 static Timeout tout_4ms;
49 static bool volatile tout_4ms_evnt;
50 static void tout_4ms_isr(void){
51     tout_4ms_evnt = true;
52 }
53
54 static int8_t cnt_sw = 0;
55
56 //tick 10s
57 static Ticker tick_10s;
58 static bool volatile tick_10s_evnt;
59 static void tick_10s_isr (void){
60     tick_10s_evnt = true;
61 }
62
63 int main (void) {
64     g_dsl = 0;
65     g_dsr = 1;
66     uint8_t cnt_display = 0;
67     g_seven_seg = to_7seg(cnt_display);
68
69     bool mux = false;
70
71     tick_1200ms.attach_us(tick_1200ms_isr, 1200000);
72     tick_4ms.attach_us(tick_4ms_isr, 4000);
73     tick_10s.attach_us(tick_10s_isr, 10000000);
74
75     swr.mode(PullUp);
76     swr.fall(swr_fall_isr);
77
78     for (;;) {
79
80         if(tick_10s_evnt){
81             tick_10s_evnt = false;
82             cnt_display = cnt_display + cnt_sw;
83
84             cnt_display = (cnt_display >= 99) ? (cnt_display%100) : cnt_display;

```

```

85
86     #ifdef VERBOSE
87     printf("n%d -- n%d", cnt_sw, cnt_display);
88     #endif
89
90 }
91
92 if(tick_4ms_evnt){
93     tick_4ms_evnt = false;
94     mux = !mux;
95
96     if(mux){
97         g_dsl = 1;
98         g_dsr = 0;
99         g_seven_seg = to_7seg(cnt_display/10);
100
101     }else{
102         g_dsl = 0;
103         g_dsr = 1;
104         g_seven_seg = to_7seg(cnt_display%10);
105     }
106
107 }
108
109 if(tick_1200ms_evnt){
110     tick_1200ms_evnt = false;
111     cnt_display = cnt_display == 99 ? 0 : cnt_display+1;
112     g_seven_seg = to_7seg(cnt_display);
113 }
114
115 if(swr_fall_evnt){
116     swr_fall_evnt = false;
117     tout_4ms.attach_us(tout_4ms_isr, 4000);
118 }
119
120 if(tout_4ms_evnt){
121     tout_4ms_evnt = false;
122
123     if (swr == 0){
124         cnt_sw++;
125
126         if( cnt_sw%2 == 0){
127             g_leds = 0;
128
129         }else{
130             g_leds = 2;
131         }
132     }
133 }
134
135 __disable_irq();
136 if(!tick_1200ms_evnt && swr_fall_evnt && tout_4ms_evnt ){
137     __WFI();
138 }
139 __enable_irq();
140
141
142
143 } // for (;;)
144 } // main()
145

```