

```

1  #include "mbed.h"
2  #include "pinout.h"
3  #include "to_7seg.h"
4
5  static DigitalOut dsr (DSR_PIN);
6  static DigitalOut dsl (DSL_PIN);
7
8  static BusOut display ( SGA_PIN, SGB_PIN, SGC_PIN, SGD_PIN, SGE_PIN, SGF_PIN, SGG_PIN);
9  static AnalogIn LIT(LIT_PIN);
10
11  static Ticker tick_4ms; // frecuencia de multiplexacion de 250Hz
12  static Ticker tick_333ms; // incremento de cuenta a una frecuencia de 3Hz
13
14  static bool volatile t_4ms; //Variable para contar el tiempo, frecuencia multiplexación
15  static bool volatile t_333ms; //Variable para contar el tiempo, contador
16
17
18  static void isr_4ms (void) { //Función para poner el contador de tiempos a true
19      t_4ms = true;
20  }
21
22  static void isr_333ms (void) { //Función para que se incremente la cuenta cada T=1/3
23      t_333ms = true;
24  }
25  int main(void) {
26      bool mux = 0;
27
28      uint16_t luz=LIT.read_u16()/656; //Mediante esta division lo que hacemos es ajustar el
29      //rango de valores de 0 a 99
30      tick_4ms.attach_us(isr_4ms, 4000); //Mediante este método, se llamará cada 4ms a la
31      //función isr_4ms
32      tick_333ms.attach_us(isr_333ms, 333333); //Mediante este método, se llamará cada 3ms a la
33      //función isr_333ms
34
35      for (;;) {
36
37          if(t_333ms){ //Cada 1/3 ms se incrementará la cuenta y al llegar a 99 pasará a 0
38              t_333ms = false;
39
40              luz=LIT.read_u16()/656;
41          }
42
43          if(t_4ms){ //Representación del contador en los displays con una frecuencia de
44          //multiplexación de 4ms
45              t_4ms = false;
46              mux = mux ? 0 : 1;
47              dsl = mux ? 1 : 0;
48              dsr = mux ? 0 : 1;
49
50              display = mux ? to_7seg(luz/10) : to_7seg(luz%10);
51          }
52
53          if(!t_4ms && !t_333ms){ //Cada vez que no pasen los 4ms o 3ms el sistema estará dormido
54              __WFI();
55          }
56      } // forever
57  } // main()

```