

```

1  #include "mbed.h"
2  #include "pinout.h"
3  #include "to_7seg.h"
4
5  static PwmOut dsr (DSR_PIN);
6  static PwmOut dsl (DSL_PIN);
7
8  static BusOut display ( SGA_PIN, SGB_PIN, SGC_PIN, SGD_PIN, SGE_PIN, SGF_PIN, SGG_PIN);
9  static BusOut leds (LDR_PIN, LDM_PIN, LDL_PIN);
10
11  static AnalogIn LIT(LIT_PIN);
12
13  static Ticker tick_4ms; // frecuencia de multiplexacion de 250Hz
14  static Ticker tick_333ms; // frecuencia de multiplexacion de 3Hz
15  static Ticker led_on;
16  static Timeout led_off;
17
18  static bool volatile t_4ms; //Variable para contar el tiempo, frecuencia multiplexación
19  static bool volatile t_333ms; //Variable para contar el tiempo, contador
20  static bool volatile tick_on; //Variable para encender el led
21  static bool volatile tick_off; //Variable para apagar el led
22
23  static void isr_4ms (void) { //Función para poner el contador de tiempos a true
24      t_4ms = true;
25  }
26
27  static void isr_333ms (void) { //Función para que se incremente la cuenta cada T=1/3
28      t_333ms = true;
29  }
30
31  static void led_on_isr (void){
32      tick_on = true;
33  }
34
35  static void led_off_isr (void){
36      tick_off = true;
37  }
38
39  int main(void) {
40      bool mux = 0;
41
42      dsr.period_us(40); //Fija el periodo de la señal a 40us
43      dsl.period_us(40); //Fija el periodo de la señal a 40us
44
45      uint16_t luz=LIT.read_ul6()/656; //Mediante esta division lo que hacemos es ajustar el
46      //rango de valores de 0 a 99
47      uint16_t t_on_led = 200.95*luz+5; //Ecuación para que varíe el tiempo de encendido del
48      //con respecto a la luz variable tiempo encendido leds
49      uint16_t t_on_disp = 0.39*luz+1; //Ecuación de la recta que se utiliza para variar el
50      //brillo de los displays variable tiempo encendido displays
51
52      tick_4ms.attach_us(isr_4ms, 4000); //frecuencia de multiplexacion de 250 Hz
53      tick_333ms.attach_us(isr_333ms, 333333); //frecuencia incremento de cuenta de 3Hz
54      led_on.attach_us(led_on_isr, 20000); // frecuencia led de 100 Hz
55
56      for (;;) {
57
58          if(tick_on){
59              tick_on=false;
60              leds=1;
61              led_off.attach_us(led_off_isr, t_on_led); //Cada ton llamará a f_off y apagará el led
62          }
63
64          if(tick_off){
65              tick_off=false;
66              leds=0;
67          }
68
69          if(t_333ms){ //Cada 1/3 ms se incrementará la cuenta y al llegar a 99 pasará a 0
70              t_333ms = false;
71              luz=LIT.read_ul6()/656;
72              t_on_led =100.95*luz+5;
73              t_on_disp = 0.39*luz+1;
74          }
75
76          if(t_4ms){ //Representación del contador en los displays con una frecuencia de
77              //multiplexación de 4ms
78              t_4ms = false;
79              mux = mux ? 0 : 1;
80              if(mux){
81                  dsr = 0;
82                  dsl = 1;
83                  display = to_7seg(luz/10);
84                  dsl.pulsewidth_us(t_on_disp);
85              }
86          }
87      }
88  }

```

```
81         }else{
82             dsr = 1;
83             dsl = 0;
84             display = to_7seg(luz%10);
85             dsr.pulsewidth_us(t_on_disp);
86         }
87     }
88     if(!t_4ms && !t_333ms && !tick_on){ //Cada vez que no pasen los 4ms o 3ms el sistema
estará dormido
89         __WFI();
90     }
91 } // forever
92 } // main()
93
```