```
#include "mbed.h"
#include "pinout.h"
#include "to_7seg.h"

// seven segment display anodes
static BusOut display ( SGA_PIN, SGB_PIN, SGC_PIN, SGD_PIN, SGE_PIN, SGF_PIN, SGG_PIN);

// display cathodes
static DigitalOut dsr (DSR_PIN);
static DigitalOut dsl (DSL_PIN);

//HC
static DigitalOut trigger (TRG_PIN);

static InterruptIn eco(ECH_PIN);

static bool volatile eco_start_evnt;
static void eco_start_isr (void){
  eco_start_evnt = true;
}

static bool volatile eco_stop_evnt;
static void eco_stop_isr (void){
  eco_stop_evnt = true;
}

// tiempo de multiplexacion
static Ticker tick_4ms;// frecuencia de multiplexacion de 250Hz
static bool volatile tick_4ms_evnt; //Variable para contar el tiempo, frecuencia multiplexación
static void tick_4ms_isr (void) {          //Función para poner el contador de tiempos a true
  tick_4ms_evnt = true;
}

//tiempo para poner trigger nivel alto 10 veces por segundo 0.1s
static Ticker tick_trg;
static bool volatile tick_trg_evnt;
static void tick_trg_isr (void){
  tick_trg_evnt = true;
}

//bajar a cero al trigger 1ms despues
static Timeout tout_trg;
static bool volatile tout_trg_evnt;
static void tout_trg_isr (void){
  tout_trg_evnt = true;
}

//anchura pulso en echo
static Timer tm_eco;
static bool volatile tm_eco_evnt;
static void tm_eco_isr (void){
  tm_eco_evnt = true;
}


int main(void) {

  uint8_t dist=0; //Distancia a la que estará el objeto
  bool sentido = false; // varible que pone a dsl o dsr encendido

  tick_4ms.attach_us(tick_4ms_isr, 4000); //multiplexacion

  tick_trg.attach_us(tick_trg_isr, 100000);

  eco.rise(eco_start_isr);
  eco.fall(eco_stop_isr);

  for (;;) {

    if(tick_trg_evnt){
      tick_trg_evnt = false;
      trigger = 1;
      tout_trg.attach_us(tout_trg_isr, 1000);
    }

    if(tout_trg_evnt){
      tout_trg_evnt = true;
      trigger = 0;
    }

    if(eco_start_evnt){
      eco_start_evnt = false;
```

```
83            tm_eco.reset();
84            tm_eco.start();
85        }
86
87        if(eco_stop_evnt){
88          eco_stop_evnt = false;
89          tm_eco.stop();
90          dist = tm_eco.read_us()/58; //Se calcula la distancia mediante esta división
91        }
92
93
94        if(tick_4ms_evnt){
95
96          tick_4ms_evnt = false;
97          sentido = !sentido;
98
99         if(sentido){
100             dsr = 0;
101             dsl = 1;
102           display = (dist > 99) ? 0x40 : to_7seg(dist/10);
103           }else{
104             dsr = 1;
105             dsl = 0;
106             display = (dist > 99) ? 0x40 : to_7seg(dist%10);
107           }
108        }
109
110       __disable_irq();
111        if(!tick_4ms_evnt && !tick_trg_evnt && !tout_trg_evnt && !eco_start_evnt &&
      !eco_stop_evnt){ //Cada vez que no pasen los 4ms o no se pulse el sistema estará dormido
112          __WFI();
113        }
114        __enable_irq();
115
116     }// forever
117   }  // main()
118
```