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

// mux stuff
static Ticker         g_mux_tick;
static bool volatile  gb_mux_evnt;

static void mux_isr (void) {
  gb_mux_evnt = true;
}

// start range_finder stuff
static Ticker         g_meas_tick;
static bool volatile  gb_meas_evnt;

static void meas_isr (void) {
  gb_meas_evnt = true;
}

int main (void) {
  // right display on?
  bool      b_right = false;
  // the 4 MSB of this variable hold the symbol to be displayed at the
  //  left display, the 4 LSB that to be displayed at the right one
  uint8_t   disp = 0;

  hw_init();

  g_seven_seg = to_7seg(disp);
  g_dsr = b_right;
  g_dsl = !b_right;
  g_mux_tick.attach_us(mux_isr, 4000);      // 250 Hz

  // initialize the range finder FSM
  rf_init(&g_trg, &g_ech);

  g_meas_tick.attach_us(meas_isr, 100000);  // 10 Hz

  for (;;) {
    // the range finder FSM
    rf_fsm();

    // start a new range measurement every 100 ms
    if (gb_meas_evnt) {
      gb_meas_evnt = false;
      gb_rf_start_msg = true;
    }

    // when the measurement is complete, update variable disp
    if (gb_rf_done_msg) {
      gb_rf_done_msg = false;
      if (g_rf_range_cm > 99) {
        disp = 0x40;// --
      } else {
        disp = (g_rf_range_cm / 10) << 4 | (g_rf_range_cm % 10);
      }
    }

    // display multiplex
    if (gb_mux_evnt) {
      // complete this code for the display multiplex ++++++++++++++++++
      gb_mux_evnt = false;
      b_right = !b_right;
      if(b_right){
          g_dsr = 1;
          g_dsl= 0;
        g_seven_seg = (g_rf_range_cm > 99 ) ? disp : to_7seg(disp%10);

        }else{
          g_dsr = 0;
          g_dsl= 1;
          g_seven_seg = (g_rf_range_cm > 99 ) ? disp : to_7seg(disp/10);

        }
      }

      // ----------------------------------------------------------------

    // sleep
    __disable_irq();
    if (!gb_meas_evnt && !gb_rf_done_msg && !gb_mux_evnt && !gb_rf_start_msg &&
```

```
    gb_rf_can_sleep) {
85          __WFI();
86        }
87        __enable_irq();
88      } // forever
89    } // main()
90
91
```