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


// range finder automatic start
static Ticker         g_meas_tick;
static bool volatile  gb_meas_evnt;

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

//variable en curso
bool volatile en_curso;;

int main (void) {
  //variable en curso
   en_curso = false;

  hw_init();

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

  // initialize the middle switch FSM
  swm_init(&g_swm);

  // initialize the display FSM
  display_init(&g_dsl,&g_dsr,&g_seven_seg);

  //  ------------------------------------------------------------------
  for (;;) {

    // the range finder FSM
    rf_fsm();

    // the middle switch FSM
    swm_fsm();

    // the display FSM
    display_fsm();

    //codigo
    if(gb_swm_long_msg){
      gb_swm_long_msg = false;

      if(!en_curso){
           en_curso = true;
           g_meas_tick.attach_us(meas_isr, 100000);

      }else{
          en_curso = false;
            g_meas_tick.detach();
      }
    } else if(gb_swm_msg && !en_curso){
      gb_swm_msg = false;
      en_curso = false;
       gb_meas_evnt = true;

      }else{
      }

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

    // 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;
      gb_display_on_msg = true;
    }

    // display multiplex


```

```
85        // sleep
86        __disable_irq();
87        if (!gb_meas_evnt && !gb_rf_done_msg  && !gb_rf_start_msg && gb_rf_can_sleep
88            && !gb_swm_msg && !gb_swm_long_msg && gb_swm_can_sleep && gb_display_can_sleep) {
89          __WFI();
90        }
91        __enable_irq();
92      } // forever
93    } // main()
94
95
```