```
/*****************************************************************************/
/*                                                                           */
/*                                 MAINLOOP                                  */
/*                             Main Program Loop                             */
/*                         Digital Oscilloscope Project                      */
/*                                 EE/CS 52                                  */
/*                                                                           */
/*****************************************************************************/

/*
   This file contains the main processing loop (background) for the Digital
   Oscilloscope project.  The only global function included is:
      main - background processing loop

   The local functions included are:
      key_lookup - get a key and look up its keycode

   The locally global variable definitions included are:
      none


   Revision History
      3/8/94   Glen George       Initial revision.
      3/9/94   Glen George       Changed initialized const arrays to static
                 (in addition to const).
      3/9/94   Glen George       Moved the position of the const keyword in
                 declarations of arrays of pointers.
      3/13/94  Glen George       Updated comments.
      3/13/94  Glen George       Removed display_menu call after plot_trace,
                 the plot function takes care of the menu.
      3/17/97  Glen George       Updated comments.
      3/17/97  Glen George       Made key_lookup function static to make it
                 truly local.
      3/17/97  Glen George       Removed KEY_UNUSED and KEYCODE_UNUSED
                 references (no longer used).
      5/27/08  Glen George       Changed code to only check for sample done if
                 it is currently sampling.
      6/03/14  Santiago Navonne  Added initialization code.
      6/11/14  Santiago Navonne  Added sleep time between draws.
*/



/* library include files */
#include  "unistd.h"

/* local include files */
#include  "interfac.h"
#include  "scopedef.h"
#include  "keyproc.h"
#include  "menu.h"
#include  "tracutil.h"




/* local function declarations */
static enum keycode  key_lookup(void);      /* translate key values into keycodes */




/*
   main

   Description:      This procedure is the main program loop for the Digital
                     Oscilloscope.  It loops getting keys from the keypad,
                     processing those keys as is appropriate.  It also handles
                     starting scope sample collection and updating the LCD
                     screen. Additionally, it initializes the triggering logic
                     and key interface.

   Arguments:        None.
   Return Value:     (int) - return code, always 0 (never returns).

```

```
   Input:            Keys from the keypad.
   Output:           Traces and menus to the display.

   Error Handling:   Invalid input is ignored.

   Algorithms:       The function is table-driven.  The processing routines
                     for each input are given in tables which are selected
                     based on the context (state) the program is operating in.
   Data Structures:  Array (process_key) to associate keys with actions
             (functions to call).

   Global Variables: None.

   Author:           Glen George
   Last Modified:    June 11, 2014

*/

int  main()
{
    /* initialize keys, triggering */
      keys_init();
      trigger_init();

    /* variables */
    enum keycode        key;            /* an input key */

    enum status
    state = MENU_ON;    /* current program state */

    unsigned char *sample;          /* a captured trace */

    /* key processing functions (one for each system state type and key) */
    static enum status  (* const process_key[NUM_KEYCODES][NUM_STATES])(enum status) =
       /*   Current System State                        */
       /*  MENU_ON      MENU_OFF             Input Key  */
      { {  menu_key,    menu_key    },   /* <Menu>      */
        {  menu_up,     no_action   },   /* <Up>        */
        {  menu_down,   no_action   },   /* <Down>      */
        {  menu_left,   no_action   },   /* <Left>      */
        {  menu_right,  no_action   },   /* <Right>     */
        {  no_action,   no_action   } }; /* illegal key */



    /* first initialize everything */
    clear_display();        /* clear the display */

    init_trace();       /* initialize the trace routines */
    init_menu();        /* initialize the menu system */


    /* infinite loop processing input */
    while(TRUE)  {

        /* check if ready to do a trace */
    if (trace_rdy())
        /* ready for a trace - do it */
        do_trace();


    /* check if have a trace to display */
    if (is_sampling() && ((sample = sample_done()) != NULL))  {

        /* have a trace - output it */
        plot_trace(sample);

        /* sleep for some time to reduce blinking of display */
        /*usleep(DRAW_INTERVAL);

        /* done processing this trace */
        trace_done();
    }
```

```
    /* now check for keypad input */
    if (key_available())  {

        /* have keypad input - get the key */
        key = key_lookup();

        /* execute processing routine for that key */
        state = process_key[key][state](state);
    }
    }


    /* done with main (never should get here), return 0 */
    return  0;

}




/*
   key_lookup

   Description:      This function gets a key from the keypad and translates
                     the raw keycode to an enumerated keycode for the main
                     loop.

   Arguments:        None.
   Return Value:     (enum keycode) - type of the key input on keypad.

   Input:            Keys from the keypad.
   Output:           None.

   Error Handling:   Invalid keys are returned as KEYCODE_ILLEGAL.

   Algorithms:       The function uses an array to lookup the key types.
   Data Structures:  Array of key types versus key codes.

   Global Variables: None.

   Author:           Glen George
   Last Modified:    Mar. 17, 1997

*/

static  enum keycode  key_lookup()
{
    /* variables */

    const static enum keycode  keycodes[] = /* array of keycodes */
        {                           /* order must match keys array exactly */
           KEYCODE_MENU,       /* <Menu>     */   /* also need an extra element */
       KEYCODE_UP,         /* <Up>       */   /* for unknown key codes */
       KEYCODE_DOWN,       /* <Down>     */
       KEYCODE_LEFT,       /* <Left>     */
       KEYCODE_RIGHT,      /* <Right>    */
       KEYCODE_ILLEGAL     /* other keys */
        };

    const static int  keys[] =   /* array of key values */
        {            /* order must match keycodes array exactly */
           KEY_MENU,    /* <Menu>     */
       KEY_UP,      /* <Up>       */
       KEY_DOWN,    /* <Down>     */
       KEY_LEFT,    /* <Left>     */
       KEY_RIGHT,   /* <Right>    */
        };

    int  key;       /* an input key */

    int  i;             /* general loop index */



    /* get a key */
```

```
    key = getkey();


    /* lookup key in keys array */
    for (i = 0; ((i < (sizeof(keys)/sizeof(int))) && (key != keys[i])); i++);


    /* return the appropriate key type */
    return  keycodes[i];

}

```