```c
/****************************************************************************/
/*                                                                          */
/*                                  KEYPROC                                 */
/*                          Key Processing Functions                        */
/*                        Digital Oscilloscope Project                      */
/*                                  EE/CS 52                                */
/*                                                                          */
/****************************************************************************/

/*
   This file contains the key processing functions for the Digital
   Oscilloscope project.  These functions are called by the main loop of the
   system.  The functions included are:
      menu_down  - process the <Down> key while in a menu
      menu_key   - process the <Menu> key
      menu_left  - process the <Left> key while in a menu
      menu_right - process the <Right> key while in a menu
      menu_up    - process the <Up> key while in a menu
      no_action  - nothing to do

   The local functions included are:
      none

   The locally global variable definitions included are:
      none


   Revision History
      3/8/94   Glen George       Initial revision.
      3/13/94  Glen George       Updated comments.
*/



/* library include files */
  /* none */

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




/*
   no_action

   Description:      This function handles a key when there is nothing to be
                     done.  It just returns.

   Arguments:        cur_state (enum status) - the current system state.
   Return Value:     (enum status) - the new system state (same as current
		     state).

   Input:            None.
   Output:           None.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: None.

   Author:           Glen George
   Last Modified:    Mar. 8, 1994

*/

enum status  no_action(enum status cur_state)
{
    /* variables */
      /* none */



    /* return the current state */
    return  cur_state;

}




/*
   menu_key

   Description:      This function handles the <Menu> key.  If the passed
                     state is MENU_ON, the menu is turned off.  If the passed
		     state is MENU_OFF, the menu is turned on.  The returned
		     state is the "opposite" of the passed state.

   Arguments:        cur_state (enum status) - the current system state.
   Return Value:     (enum status) - the new system state ("opposite" of the
		     as current state).

   Input:            None.
   Output:           The menu is either turned on or off.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: None.

   Author:           Glen George
   Last Modified:    Mar. 8, 1994

*/

enum status  menu_key(enum status cur_state)
{
    /* variables */
      /* none */



    /* check if need to turn the menu on or off */
    if (cur_state == MENU_ON)
        /* currently the menu is on, turn it off */
    clear_menu();
    else
        /* currently the menu is off, turn it on */
    display_menu();


    /* all done, return the "opposite" of the current state */
    if (cur_state == MENU_ON)
        /* state was MENU_ON, change it to MENU_OFF */
        return  MENU_OFF;
    else
        /* state was MENU_OFF, change it to MENU_ON */
        return  MENU_ON;

}




/*
   menu_up

   Description:      This function handles the <Up> key when in a menu.  It
                     goes to the previous menu entry and leaves the system
		     state unchanged.

   Arguments:        cur_state (enum status) - the current system state.
   Return Value:     (enum status) - the new system state (same as current
             state).

   Input:            None.
   Output:           The menu display is updated.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: None.

   Author:           Glen George
   Last Modified:    Mar. 8, 1994

*/

enum status  menu_up(enum status cur_state)
{
    /* variables */
      /* none */



    /* go to the previous menu entry */
    previous_entry();


    /* return the current state */
    return  cur_state;

}




/*
   menu_down

   Description:      This function handles the <Down> key when in a menu.  It
                     goes to the next menu entry and leaves the system state
             unchanged.

   Arguments:        cur_state (enum status) - the current system state.
   Return Value:     (enum status) - the new system state (same as current
             state).

   Input:            None.
   Output:           The menu display is updated.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: None.

   Author:           Glen George
   Last Modified:    Mar. 8, 1994

*/

enum status  menu_down(enum status cur_state)
{
    /* variables */
      /* none */



    /* go to the next menu entry */
    next_entry();


    /* return the current state */
    return  cur_state;
```

```c

}




/*
   menu_left

   Description:      This function handles the <Left> key when in a menu.  It
                     invokes the left function for the current menu entry and
             leaves the system state unchanged.

   Arguments:        cur_state (enum status) - the current system state.
   Return Value:     (enum status) - the new system state (same as current
             state).

   Input:            None.
   Output:           The menu display may be updated.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: None.

   Author:           Glen George
   Last Modified:    Mar. 8, 1994

*/

enum status  menu_left(enum status cur_state)
{
    /* variables */
      /* none */



    /* invoke the <Left> key function for the current menu entry */
    menu_entry_left();


    /* return the current state */
    return  cur_state;

}




/*
   menu_right

   Description:      This function handles the <Right> key when in a menu.  It
                     invokes the right function for the current menu entry and
             leaves the system state unchanged.

   Arguments:        cur_state (enum status) - the current system state.
   Return Value:     (enum status) - the new system state (same as current
             state).

   Input:            None.
   Output:           The menu display may be updated.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: None.

   Author:           Glen George
   Last Modified:    Mar. 8, 1994

```

```c
*/

enum status  menu_right(enum status cur_state)
{
    /* variables */
      /* none */



    /* invoke the <Right> key function for the current menu entry */
    menu_entry_right();


    /* return the current state */
    return  cur_state;

}

```