```
/*************************************************************************/
/*                                                                       */
/*                                  MENU                                 */
/*                              Menu Functions                           */
/*                        Digital Oscilloscope Project                   */
/*                                 EE/CS 52                              */
/*                                                                       */
/*************************************************************************/

/*
   This file contains the functions for processing menu entries for the
   Digital Oscilloscope project.  These functions take care of maintaining the
   menus and handling menu updates for the system.  The functions included
   are:
      clear_menu       - remove the menu from the display
      display_menu     - display the menu
      init_menu        - initialize menus
      menu_entry_left  - take care of <Left> key for a menu entry
      menu_entry_right - take care of <Right> key for a menu entry
      next_entry       - next menu entry
      previous_entry   - previous menu entry
      refresh_menu     - re-display the menu if currently being displayed
      reset_menu       - reset the current selection to the top of the menu

   The local functions included are:
      display_entry    - display a menu entry (including option setting)

   The locally global variable definitions included are:
      menu             - the menu
      menu_display     - whether or not the menu is currently displayed
      menu_entry       - the currently selected menu entry


   Revision History
      3/8/94   Glen George       Initial revision.
      3/9/94   Glen George       Changed position of const keyword in array
                 declarations involving pointers.
      3/13/94  Glen George       Updated comments.
      3/13/94  Glen George       Added display_entry function to output a menu
                 entry and option setting to the LCD (affects
                 many functions).
      3/13/94  Glen George       Changed calls to set_status due to changing
                      enum scale_status definition.
      3/13/94  Glen George       No longer clear the menu area before
                 restoring the trace in clear_menu() (not
                 needed).
      3/17/97  Glen George       Updated comments.
      3/17/97  Glen George       Fixed minor bug in reset_menu().
      3/17/97  Glen George       When initializing the menu in init_menu(),
                 set the delay to MIN_DELAY instead of 0 and
                 trigger to a middle value instead of
                 MIN_TRG_LEVEL_SET.
      5/3/06   Glen George       Changed to a more appropriate constant in
                                 display_entry().
      5/3/06   Glen George       Updated comments.
      5/9/06   Glen George       Changed menus to handle a list for mode and
                              scale (move up and down list), instead of
                      toggling values.
*/



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

/* local include files */
#include  "scopedef.h"
#include  "lcdout.h"
#include  "menu.h"
#include  "menuact.h"
#include  "tracutil.h"
```

```
/* local function declarations */
static void  display_entry(int, int);      /* display a menu entry and its setting */




/* locally global variables */
static int  menu_display;           /* TRUE if menu is currently displayed */

const static struct menu_item  menu[] =     /* the menu */
    { { "Mode",    0, 4, display_mode      },
      { "Scale",   0, 5, display_scale     },
      { "Sweep",   0, 5, display_sweep     },
      { "Trigger", 0, 7, no_display        },
      { "Level",   2, 7, display_trg_level },
      { "Slope",   2, 7, display_trg_slope },
      { "Delay",   2, 7, display_trg_delay },
    };

static int  menu_entry;         /* currently selected menu entry */




/*
   init_menu

   Description:      This function initializes the menu routines.  It sets
                     the current menu entry to the first entry, indicates the
             display is off, and initializes the options (and
             hardware) to normal trigger mode, scale displayed, the
             fastest sweep rate, a middle trigger level, positive
             trigger slope, and minimum delay.  Finally, it displays
             the menu.

   Arguments:        None.
   Return Value:     None.

   Input:            None.
   Output:           The menu is displayed.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: menu_display - reset to FALSE.
             menu_entry   - reset to first entry (0).

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

*/

void  init_menu(void)
{
    /* variables */
      /* none */



    /* set the menu parameters */
    menu_entry = 0;     /* first menu entry */
    menu_display = FALSE;   /* menu is not currently displayed (but it will be shortly) */


    /* set the scope (option) parameters */
    set_trigger_mode(NORMAL_TRIGGER);   /* normal triggering */
    set_scale(SCALE_AXES);      /* scale is axes */
    set_sweep(0);           /* first sweep rate */
    set_trg_level((MIN_TRG_LEVEL_SET + MAX_TRG_LEVEL_SET) / 2); /* middle trigger level */
    set_trg_slope(SLOPE_POSITIVE);  /* positive slope */
    set_trg_delay(MIN_DELAY);       /* minimum delay */
```

```
    /* now display the menu */
    display_menu();


    /* done initializing, return */
    return;

}



/*
   clear_menu

   Description:      This function removes the menu from the display.  The
                     trace under the menu is restored.  The flag menu_display,
             is cleared, indicating the menu is no longer being
             displayed.  Note: if the menu is not currently being
             displayed this function does nothing.

   Arguments:        None.
   Return Value:     None.

   Input:            None.
   Output:           The menu if displayed, is removed and the trace under it
             is rewritten.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: menu_display - checked and set to FALSE.

   Author:           Glen George
   Last Modified:    Mar. 13, 1994

*/

void  clear_menu(void)
{
    /* variables */
      /* none */



    /* check if the menu is currently being displayed */
    if (menu_display)  {

        /* menu is being displayed - turn it off and restore the trace in that area */
    restore_menu_trace();
    }


    /* no longer displaying the menu */
    menu_display = FALSE;


    /* all done, return */
    return;

}



/*
   display_menu

   Description:      This function displays the menu.  The trace under the
                     menu is overwritten (but it was saved).  The flag
             menu_display, is also set, indicating the menu is
             currently being displayed.  Note: if the menu is already
             being displayed this function does not redisplay it.
```

```
226
227     Arguments:          None.
228     Return Value:       None.
229
230     Input:              None.
231     Output:             The menu is displayed.
232
233     Error Handling:     None.
234
235     Algorithms:         None.
236     Data Structures:    None.
237
238     Global Variables: menu_display - set to TRUE.
239                menu_entry   - used to highlight currently selected entry.
240
241     Author:             Glen George
242     Last Modified:      Mar. 13, 1994
243
244 */
245
246 void  display_menu(void)
247 {
248     /* variables */
249     int  i;     /* loop index */
250
251
252
253     /* check if the menu is currently being displayed */
254     if (!menu_display)  {
255
256         /* menu is not being displayed - turn it on */
257     /* display it entry by entry */
258     for (i = 0; i < NO_MENU_ENTRIES; i++)  {
259
260         /* display this entry - check if it should be highlighted */
261         if (i == menu_entry)
262             /* currently selected entry - highlight it */
263             display_entry(i, TRUE);
264         else
265             /* not the currently selected entry - "normal video" */
266             display_entry(i, FALSE);
267         }
268     }
269
270
271     /* now are displaying the menu */
272     menu_display = TRUE;
273
274
275     /* all done, return */
276     return;
277
278 }
279
280
281
282
283 /*
284     refresh_menu
285
286     Description:        This function displays the menu if it is currently being
287                displayed.  The trace under the menu is overwritten (but
288                it was already saved).
289
290     Arguments:          None.
291     Return Value:       None.
292
293     Input:              None.
294     Output:             The menu is displayed.
295
296     Error Handling:     None.
297
298     Algorithms:         None.
299     Data Structures:    None.
300
```

```
   Global Variables: menu_display - determines if menu should be displayed.

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

*/

void  refresh_menu(void)
{
    /* variables */
      /* none */



    /* check if the menu is currently being displayed */
    if (menu_display)  {

        /* menu is currently being displayed - need to refresh it */
    /* do this by turning off the display, then forcing it back on */
    menu_display = FALSE;
    display_menu();
    }


    /* refreshed the menu if it was displayed, now return */
    return;

}




/*
   reset_menu

   Description:      This function resets the current menu selection to the
                     first menu entry.  If the menu is currently being
             displayed the display is updated.

   Arguments:        None.
   Return Value:     None.

   Input:            None.
   Output:           The menu display is updated if it is being displayed.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: menu_display - checked to see if menu is displayed.
             menu_entry   - reset to 0 (first entry).

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

*/

void  reset_menu(void)
{
    /* variables */
      /* none */



    /* check if the menu is currently being displayed */
    if (menu_display)  {

        /* menu is being displayed */
    /* remove highlight from currently selected entry */
    display_entry(menu_entry, FALSE);
    }


    /* reset the currently selected entry */
```

```
    menu_entry = 0;


    /* finally, highlight the first entry if the menu is being displayed */
    if (menu_display)
    display_entry(menu_entry, TRUE);



    /* all done, return */
    return;

}




/*
   next_entry

   Description:      This function changes the current menu selection to the
                     next menu entry.  If the current selection is the last
             entry in the menu, it is not changed.  If the menu is
             currently being displayed, the display is updated.

   Arguments:        None.
   Return Value:     None.

   Input:            None.
   Output:           The menu display is updated if it is being displayed and
             the entry selected changes.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: menu_display - checked to see if menu is displayed.
             menu_entry   - updated to a new entry (if not at end).

   Author:           Glen George
   Last Modified:    Mar. 13, 1994

*/

void  next_entry(void)
{
    /* variables */
      /* none */



    /* only update if not at end of the menu */
    if (menu_entry < (NO_MENU_ENTRIES - 1))  {

        /* not at the end of the menu */

    /* turn off current entry if displaying */
    if (menu_display)
            /* displaying menu - turn off currently selected entry */
        display_entry(menu_entry, FALSE);

    /* update the menu entry to the next one */
    menu_entry++;

    /* now highlight this entry if displaying the menu */
    if (menu_display)
            /* displaying menu - highlight newly selected entry */
        display_entry(menu_entry, TRUE);
    }


    /* all done, return */
    return;

```

```
}



/*
   previous_entry

   Description:      This function changes the current menu selection to the
                     previous menu entry.  If the current selection is the
             first entry in the menu, it is not changed.  If the menu
             is currently being displayed, the display is updated.

   Arguments:        None.
   Return Value:     None.

   Input:            None.
   Output:           The menu display is updated if it is being displayed and
             the currently selected entry changes.

   Error Handling:   None.

   Algorithms:       None.
   Data Structures:  None.

   Global Variables: menu_display - checked to see if menu is displayed.
             menu_entry   - updated to a new entry (if not at start).

   Author:           Glen George
   Last Modified:    Mar. 13, 1994

*/

void  previous_entry(void)
{
    /* variables */
      /* none */



    /* only update if not at the start of the menu */
    if (menu_entry > 0)  {

        /* not at the start of the menu */

    /* turn off current entry if displaying */
    if (menu_display)
            /* displaying menu - turn off currently selected entry */
        display_entry(menu_entry, FALSE);

    /* update the menu entry to the previous one */
    menu_entry--;

    /* now highlight this entry if displaying the menu */
    if (menu_display)
            /* displaying menu - highlight newly selected entry */
        display_entry(menu_entry, TRUE);

    }


    /* all done, return */
    return;

}



/*
   menu_entry_left

   Description:      This function handles the <Left> key for the current menu
                     selection.  It does this by doing a table lookup on the
             current menu selection.
```

```
526
527     Arguments:        None.
528     Return Value:     None.
529
530     Input:            None.
531     Output:           The menu display is updated if it is being displayed and
532               the <Left> key causes a change to the display.
533
534     Error Handling:   None.
535
536     Algorithms:       Table lookup is used to determine what to do for the
537               input key.
538     Data Structures:  An array holds the table of key processing routines.
539
540     Global Variables: menu_entry - used to select the processing function.
541
542     Author:           Glen George
543     Last Modified:    May 9, 2006
544
545 */
546
547 void  menu_entry_left(void)
548 {
549     /* variables */
550
551     /* key processing functions */
552     static void  (* const process[])(void) =
553        /*  Mode            Scale             Sweep           Trigger      */
554         {  mode_down,      scale_down,       sweep_down,     trace_rearm,
555            trg_level_down, trg_slope_toggle, trg_delay_down               };
556        /*  Level           Slope             Delay                        */
557
558
559
560     /* invoke the appropriate <Left> key function */
561     process[menu_entry]();
562
563     /* if displaying menu entries, display the new value */
564     /* note: since it is being changed - know this option is selected */
565     if (menu_display)  {
566         menu[menu_entry].display((MENU_X + menu[menu_entry].opt_off),
567                          (MENU_Y + menu_entry), OPTION_SELECTED);
568     }
569
570
571     /* all done, return */
572     return;
573
574 }
575
576
577
578
579 /*
580    menu_entry_right
581
582    Description:      This function handles the <Right> key for the current
583                      menu selection.  It does this by doing a table lookup on
584              the current menu selection.
585
586    Arguments:        None.
587    Return Value:     None.
588
589    Input:            None.
590    Output:           The menu display is updated if it is being displayed and
591              the <Right> key causes a change to the display.
592
593    Error Handling:   None.
594
595    Algorithms:       Table lookup is used to determine what to do for the
596              input key.
597    Data Structures:  An array holds the table of key processing routines.
598
599    Global Variables: menu       - used to display the new menu value.
600              menu_entry - used to select the processing function.
```

```
601
602     Author:           Glen George
603     Last Modified:    May 9, 2006
604
605 */
606
607 void  menu_entry_right(void)
608 {
609     /* variables */
610
611     /* key processing functions */
612     static void  (* const process[])(void) =
613        /*  Mode          Scale             Sweep           Trigger      */
614         {  mode_up    ,  scale_up,     sweep_up,       trace_rearm,
615            trg_level_up, trg_slope_toggle, trg_delay_up                 };
616        /*  Level         Slope             Delay                        */
617
618
619
620     /* invoke the appropriate <Right> key function */
621     process[menu_entry]();
622
623     /* if displaying menu entries, display the new value */
624     /* note: since it is being changed - know this option is selected */
625     if (menu_display)  {
626         menu[menu_entry].display((MENU_X + menu[menu_entry].opt_off),
627                         (MENU_Y + menu_entry), OPTION_SELECTED);
628     }
629
630
631     /* all done, return */
632     return;
633
634 }
635
636
637
638
639 /*
640    display_entry
641
642    Description:      This function displays the passed menu entry and its
643              current option setting.  If the second argument is TRUE
644              it displays them with color SELECTED and OPTION_SELECTED
645              respectively.  If the second argument is FALSE it
646              displays the menu entry with color NORMAL and the option
647              setting with color OPTION_NORMAL.
648
649    Arguments:        entry (int)    - menu entry to be displayed.
650              selected (int) - whether or not the menu entry is
651                               currently selected (determines the color
652                        with which the entry is output).
653    Return Value:     None.
654
655    Input:            None.
656    Output:           The menu entry is output to the LCD.
657
658    Error Handling:   None.
659
660    Algorithms:       None.
661    Data Structures:  None.
662
663    Global Variables: menu - used to display the menu entry.
664
665    Author:           Glen George
666    Last Modified:    Aug. 13, 2004
667
668 */
669
670 static void  display_entry(int entry, int selected)
671 {
672     /* variables */
673       /* none */
674
675
```

```
676
677     /* output the menu entry with the appropriate color */
678     plot_string((MENU_X + menu[entry].h_off), (MENU_Y + entry), menu[entry].s,
679             (selected ? SELECTED : NORMAL));
680     /* also output the menu option with the appropriate color */
681     menu[entry].display((MENU_X + menu[entry].opt_off), (MENU_Y + entry),
682                 (selected ? OPTION_SELECTED : OPTION_NORMAL));
683
684
685     /* all done outputting this menu entry - return */
686     return;
687
688 }
689
```