```
/************************************************************************
       7 segments LED
              led_drv.c

              October 1,2005       H8/3694F
              October23,2005

       Copyright (C) 2005  Kenji Arai/JH1PJL
       All rights reserved. Permission is granted to use, modify,or redistribute
       this software so long as it is not sold or exploited for profit.

       THIS SOFTWARE IS PROVIDED AS IS AND WITHOUT WARRANTY OF ANY KIND,
       EITHER EXPRESSED OR IMPLIED.
************************************************************************/
/*  -----< Include Files >--------------------------------------------- */
#include      "iodefine.h"
#include      "led.h"
#include      "task.h"
#include      "const.h"


/* -----< RAM assign >------------------------------------------------- */
unsigned char led_disp_buf[6];      // display segment data buffer
unsigned char action_flag;          // flag for 1st and 2nd period
unsigned char act_no;
unsigned int  tim_drv0, tim_drv1;   // 700uSec 1st(drive) time and 2nd(non-drive) time

extern unsigned int tim_usr0;               // user timer no.0
//extern unsigned char buzzer_flg;  // buzzer on/off flag
extern unsigned char mode_led_flg;  // Mode LED flag
extern unsigned char lap_ptr_reach_end;

/* -----< Constant >--------------------------------------------------- */
//                                                  0   1  2   3   4   5   6   7   8   9
static char const svnsg_dat0[] = {252,96,218,242,102,182,190,224,254,230};
//                                                  a  b   c   d   e   f
static char const svnsg_dat1[] = {250,62,156,122,158,142};

/*  -----< Control program >------------------------------------------- */
void reset_led ( void )
{
       action_flag =0;
       tim_drv0 = 1000;
       tim_drv1 = INTVL_1MS  - tim_drv0;
       act_no = 0;
       led_disp_buf[0] = svnsg_dat0[0];
       led_disp_buf[1] = svnsg_dat0[1];
       led_disp_buf[2] = svnsg_dat0[2];
       led_disp_buf[3] = svnsg_dat0[3];
       led_disp_buf[4] = svnsg_dat0[4];
       led_disp_buf[5] = svnsg_dat0[5];
       TW.GRC = TW.TCNT + INTVL_1MS;
}

void tim_dsp_to_led(unsigned long tim, unsigned char dsp_mode, unsigned char dsp_zero)
{
       unsigned char i, zero;
       unsigned long dt;
```

```
if (dsp_zero){
	dt = (tim & 0x07ffffff);                          // output Tx:xx:xx.xx
	dt = dt  - ((dt / 72000000) * 72000000);    // output xT:xx:xx.xx
	dt = dt  - ((dt / 7200000 ) * 7200000 );    // output xx:Tx:xx.xx
	i = (char)(dt  / 1200000 );
	if (i == 0){
		led_disp_buf[5] = 0;
		zero = 0;
	} else {
		led_disp_buf[5] = svnsg_dat0[i];
		zero = 1;
	}
	dt = dt  - ((dt / 1200000 ) * 1200000);     // output xx:xT:xx.xx
	i = (char)(dt  / 120000  );
	if ((i == 0) & (zero == 0)){
		led_disp_buf[4] = 0;
		zero = 0;
	} else {
		led_disp_buf[4] = svnsg_dat0[i] + LED_DOT; // Add dot (+1)
		zero = 1;
	}
	dt = dt  - ((dt / 120000  ) * 120000);      // output xx:xx:Tx.xx
	i = (char)(dt  / 20000   );
	if ((i == 0) & (zero == 0)){
		led_disp_buf[3] = 0;
		zero = 0;
	} else {
		led_disp_buf[3] = svnsg_dat0[i];    // Add dot (+1)
		zero = 1;
	}
	dt = dt  - ((dt / 20000   ) * 20000);       // output xx:xx:xT.xx
	i = (char)(dt  / 2000    );
	if ((i == 0) & (zero == 0)){
		led_disp_buf[2] = 0;
		zero = 0;
	} else {
		led_disp_buf[2] = svnsg_dat0[i] + LED_DOT; // Add dot (+1)
		zero = 1;
	}
	dt = dt  - ((dt / 2000    ) * 2000);        // output xx:xx:xx.Tx
	i = (char)(dt  / 200     );
	if ((i == 0) & (zero == 0)){
		led_disp_buf[1] = 0;
		zero = 0;
	} else {
		led_disp_buf[1] = svnsg_dat0[i];    // Add dot (+1)
		zero = 1;
	}
	dt = dt  - ((dt / 200     ) * 200);         // output xx:xx:xx.xT
	i = (char)(dt  / 20     );
	if (dsp_mode == 1) {
		led_disp_buf[0] = svnsg_dat0[i];
	} else {
		led_disp_buf[0] = LED_MINUS;
	}
} else {
	dt = (tim & 0x07ffffff);                          // output Tx:xx:xx.xx
```

```
                dt = dt  - ((dt / 72000000) * 72000000);    // output xT:xx:xx.xx
                dt = dt  - ((dt / 7200000 ) * 7200000 );    // output xx:Tx:xx.xx
                i = (char)(dt  / 1200000 );
                led_disp_buf[5] = svnsg_dat0[i];
                dt = dt  - ((dt / 1200000 ) * 1200000);     // output xx:xT:xx.xx
                i = (char)(dt  / 120000  );
                led_disp_buf[4] = svnsg_dat0[i] + LED_DOT; // Add dot (+1)
                dt = dt  - ((dt / 120000  ) * 120000);      // output xx:xx:Tx.xx
                i = (char)(dt  / 20000   );
                led_disp_buf[3] = svnsg_dat0[i];
                dt = dt  - ((dt / 20000   ) * 20000);       // output xx:xx:xT.xx
                i = (char)(dt  / 2000    );
                led_disp_buf[2] = svnsg_dat0[i] + LED_DOT; // Add dot (+1)
                dt = dt  - ((dt / 2000    ) * 2000);        // output xx:xx:xx.Tx
                i = (char)(dt  / 200     );
                led_disp_buf[1] = svnsg_dat0[i];
                dt = dt  - ((dt / 200     ) * 200);         // output xx:xx:xx.xT
                i = (char)(dt  / 20     );
                if (dsp_mode == 1) {
                        led_disp_buf[0] = svnsg_dat0[i];
                } else {
                        led_disp_buf[0] = LED_MINUS;
                }
        }
}

void id_dsp_to_led( unsigned char dt )
{
        int j;
        unsigned char i, zero;

        if (dt == 255){
                lap_ptr_reach_end = 1;
                end_id_dsp_to_led();
        } else {
                lap_ptr_reach_end = 0;
                j = 5;
                i = dt/100;
                if (i == 0){
                        zero = 0;
                } else {
                        led_disp_buf[j --] = svnsg_dat0[i];  // output Dxx
                        zero = 1;
                }
                dt = dt  - (i * 100);
                i = dt/10;
                if ((i == 0) & (zero == 0)){
                        ;
                } else {
                        led_disp_buf[j --] = svnsg_dat0[i];  // output xDx
                }
                i = dt  - (i * 10);
                led_disp_buf[j --] = svnsg_dat0[i];         // output xxD
                for ( ; j>= 0; j --){
                        led_disp_buf[j] = LED_BLANK;
                }
        }
}
```

```
// 7 segments LED display rutine
// 6 digit x 8 digits(7seg+dot)
// every timebase(1mS) change one digit with bright control
//             ____tim_drv0(on)            ____
//             |      |                          |      |
//             |      |                          |      |
//      -----  ---tim_drv1(off) ----- --------
void led_drive ( void )
{
       unsigned int cnt;

       if (action_flag) {    // LED off action
              cnt = TW.TCNT + tim_drv1;
              if (cnt > B_CLK_20MS ) {
                     TW.GRC = cnt  - B_CLK_20MS;
              } else {
                     TW.GRC = cnt;
              }
              action_flag = 0;
              switch(act_no){
                     case 0:
                            L7SEG_NO1 = 0;
                            break;
                     case 1:
                            L7SEG_NO2 = 0;
                            break;
                     case 2:
                            L7SEG_NO3 = 0;
                            break;
                     case 3:
                            L7SEG_NO4 = 0;
                            break;
                     case 4:
                            L7SEG_NO5 = 0;
                            break;
                     case 5:
                            L7SEG_NO6 = 0;
                            act_no = 0xff;
                            break;
                     default:
                            break;
              }
              act_no++;
       } else {                     // LED on action
              cnt = TW.TCNT + tim_drv0;
              if (cnt > B_CLK_20MS ) {
                     TW.GRC = cnt  - B_CLK_20MS;
              } else {
                     TW.GRC = cnt;
              }
              action_flag = 1;
              L7SEG_P0 = (L7SEG_P0 & 0x0f)  | (led_disp_buf[act_no] & 0xf0);
              L7SEG_P1 = led_disp_buf[act_no] & 0x0f;
              switch(act_no){
                     case 0:
                            L7SEG_NO1 = 1;
                            break;
```

```
                        case 1:
                                L7SEG_NO2 = 1;
                                break;
                        case 2:
                                L7SEG_NO3 = 1;
                                break;
                        case 3:
                                L7SEG_NO4 = 1;
                                break;
                        case 4:
                                L7SEG_NO5 = 1;
                                break;
                        case 5:
                                L7SEG_NO6 = 1;
                                break;
                        default:
                                break;
                }
        }
}

void delete_ok_dsp_to_led( void )
{
        // show CLEAr
        led_disp_buf[5] = LED_BLANK + LED_DOT;
        led_disp_buf[4] = 156;
        led_disp_buf[3] = 28;
        led_disp_buf[2] = 158;
        led_disp_buf[1] = 238;
        led_disp_buf[0] = 10;
}

void hold_dsp_to_led( void )
{
        led_disp_buf[0] = LED_MINUS;
        led_disp_buf[1] = LED_MINUS;
        led_disp_buf[2] = LED_MINUS;
        led_disp_buf[3] = LED_MINUS;
        led_disp_buf[4] = LED_MINUS;
        led_disp_buf[5] = LED_MINUS;
}

void allzero_dsp_to_led( void )
{
        led_disp_buf[0] = LED_ZERO;
        led_disp_buf[1] = LED_ZERO;
        led_disp_buf[2] = LED_ZERO;
        led_disp_buf[3] = LED_ZERO;
        led_disp_buf[4] = LED_ZERO;
        led_disp_buf[5] = LED_ZERO;
}

void end_id_dsp_to_led( void )
{
        // show  -End -
        led_disp_buf[5] = LED_MINUS;
        led_disp_buf[4] = 158;
        led_disp_buf[3] = 42;
```

```
        led_disp_buf[2] = 122;
        led_disp_buf[1] = LED_MINUS;
        led_disp_buf[0] = LED_BLANK;
}

void led_bright ( void )
{
        unsigned long ad;

        ad = (unsigned long)AD.ADDRA/64;
        if (ad > 1000){
                ad = INTVL_1MS  - 100;
        } else if (ad > 900){
                ad = INTVL_1MS  - 200;
        } else if (ad > 800){
                ad = INTVL_1MS  - 300;
        } else if ( ad < 3){
                ad = 5;
        } else {
                ad = (ad * 0xc971) / 0x10000 + 5;
        }
        tim_drv0 = (unsigned int)ad;
        tim_drv1 = INTVL_1MS  - tim_drv0;
}

//
//      ONLY FOR TEST PURPOSE
//

void test_led_bzr( void )
{
        LED_MODE = ON_DR_HS;  // Mode LED ON
        tim_usr0 = 1000;
        while (tim_usr0) ;
        LED_MODE = OFF_DR_HS; // Mode LED OFF
        ;
        OUT_LED2 = 1;
        tim_usr0 = 1000;
        while (tim_usr0) ;
        OUT_LED2 = 0;
        ;
        BUZZER = ON_DR_HS;           // Buzzer ON
        tim_usr0 = 1000;
        while (tim_usr0) ;
        BUZZER = OFF_DR_HS;          // Buzzer OFF
        ;
}

void test_7seg_ledxx ( void )
{
        unsigned int i,j;

        for (j=0;j<5;j++){
                for (i=0;i<300;i++){
                        // one action #1
                        L7SEG_P0 = (L7SEG_P0 & 0x0f)  | (svnsg_dat0[j+0] & 0xf0);
                        L7SEG_P1 = svnsg_dat0[j+0] & 0x0f;
                        L7SEG_NO1 = 1;
```

```
                        tim_usr0 = 1;
                        while (tim_usr0) ;
                        L7SEG_NO1 = 0;
                        // ---------------
                        // one action #2
                        L7SEG_P0 = (L7SEG_P0 & 0x0f) |(svnsg_dat0[j+1] & 0xf0);
                        L7SEG_P1 = svnsg_dat0[j+1] & 0x0f;
                        L7SEG_NO2 = 1;
                        tim_usr0 = 1;
                        while (tim_usr0) ;
                        L7SEG_NO2 = 0;
                        // ---------------
                        // one action #3
                        L7SEG_P0 = (L7SEG_P0 & 0x0f) |(svnsg_dat0[j+2] & 0xf0);
                        L7SEG_P1 = svnsg_dat0[j+2] & 0x0f;
                        L7SEG_NO3 = 1;
                        L7SEG_DOT = ON;
                        tim_usr0 = 1;
                        while (tim_usr0) ;
                        L7SEG_NO3 = 0;
                        // ---------------
                        // one action #4
                        L7SEG_P0 = (L7SEG_P0 & 0x0f) |(svnsg_dat0[j+3] & 0xf0);
                        L7SEG_P1 = svnsg_dat0[j+3] & 0x0f;
                        L7SEG_NO4 = 1;
                        tim_usr0 = 1;
                        while (tim_usr0) ;
                        L7SEG_NO4 = 0;
                        // ---------------
                        // one action #5
                        L7SEG_P0 = (L7SEG_P0 & 0x0f) |(svnsg_dat0[j+4] & 0xf0);
                        L7SEG_P1 = svnsg_dat0[j+4] & 0x0f;
                        L7SEG_NO5 = 1;
                        L7SEG_DOT = ON;
                        tim_usr0 = 1;
                        while (tim_usr0) ;
                        L7SEG_NO5 = 0;
                        // ---------------
                        // one action #6
                        L7SEG_P0 = (L7SEG_P0 & 0x0f) |(svnsg_dat0[j+5] & 0xf0);
                        L7SEG_P1 = svnsg_dat0[j+5] & 0x0f;
                        L7SEG_NO6 = 1;
                        tim_usr0 = 1;
                        while (tim_usr0) ;
                        L7SEG_NO6 = 0;
                        // ---------------
                }
        }
        L7SEG_P0 = 0x00;
        L7SEG_P1 = 0x00;
}

void test_7seg_led ( void )
{
        // ----------------- SEGMENT a -----------------
        // one action
        L7SEG_P0 = 0x80;
        L7SEG_NO1 = 1;
```

```
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO1 = 0;
// ---------------
// one action
L7SEG_P0 = 0x80;
L7SEG_NO2 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO2 = 0;
// ---------------
// one action
L7SEG_P0 = 0x80;
L7SEG_NO3 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO3 = 0;
// ---------------
// one action
L7SEG_P0 = 0x80;
L7SEG_NO4 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO4 = 0;
// ---------------
// one action
L7SEG_P0 = 0x80;
L7SEG_NO5 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO5 = 0;
// ---------------
// one action
L7SEG_P0 = 0x80;
L7SEG_NO6 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO6 = 0;
// ---------------
// ------------------ SEGMENT b ------------------
// one action
L7SEG_P0 = 0x40;
L7SEG_NO1 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO1 = 0;
// ---------------
// one action
L7SEG_P0 = 0x40;
L7SEG_NO2 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO2 = 0;
// ---------------
// one action
L7SEG_P0 = 0x40;
L7SEG_NO3 = 1;
tim_usr0 = 30;
```

```
while (tim_usr0) ;
L7SEG_NO3 = 0;
// ---------------
// one action
L7SEG_P0 = 0x40;
L7SEG_NO4 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO4 = 0;
// ---------------
// one action
L7SEG_P0 = 0x40;
L7SEG_NO5 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO5 = 0;
// ---------------
// one action
L7SEG_P0 = 0x40;
L7SEG_NO6 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO6 = 0;
// ---------------
// ------------------ SEGMENT c ------------------
// one action
L7SEG_P0 = 0x20;
L7SEG_NO1 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO1 = 0;
// ---------------
// one action
L7SEG_P0 = 0x20;
L7SEG_NO2 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO2 = 0;
// ---------------
// one action
L7SEG_P0 = 0x20;
L7SEG_NO3 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO3 = 0;
// ---------------
// one action
L7SEG_P0 = 0x20;
L7SEG_NO4 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO4 = 0;
// ---------------
// one action
L7SEG_P0 = 0x20;
L7SEG_NO5 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
```

```
L7SEG_NO5 = 0;
// ---------------
// one action
L7SEG_P0 = 0x20;
L7SEG_NO6 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO6 = 0;
// ---------------
// ------------------ SEGMENT d ------------------
// one action
L7SEG_P0 = 0x10;
L7SEG_NO1 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO1 = 0;
// ---------------
// one action
L7SEG_P0 = 0x10;
L7SEG_NO2 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO2 = 0;
// ---------------
// one action
L7SEG_P0 = 0x10;
L7SEG_NO3 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO3 = 0;
// ---------------
// one action
L7SEG_P0 = 0x10;
L7SEG_NO4 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO4 = 0;
// ---------------
// one action
L7SEG_P0 = 0x10;
L7SEG_NO5 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO5 = 0;
// ---------------
// one action
L7SEG_P0 = 0x10;
L7SEG_NO6 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO6 = 0;
// ---------------
L7SEG_P0 = 0x00;
// ------------------ SEGMENT e ------------------
// one action
L7SEG_P1 = 0x08;
L7SEG_NO1 = 1;
tim_usr0 = 30;
```

```
while (tim_usr0) ;
L7SEG_NO1 = 0;
// ---------------
// one action
L7SEG_P1 = 0x08;
L7SEG_NO2 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO2 = 0;
// ---------------
// one action
L7SEG_P1 = 0x08;
L7SEG_NO3 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO3 = 0;
// ---------------
// one action
L7SEG_P1 = 0x08;
L7SEG_NO4 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO4 = 0;
// ---------------
// one action
L7SEG_P1 = 0x08;
L7SEG_NO5 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO5 = 0;
// ---------------
// one action
L7SEG_P1 = 0x08;
L7SEG_NO6 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO6 = 0;
// ---------------
// ------------------ SEGMENT f ------------------
// one action
L7SEG_P1 = 0x04;
L7SEG_NO1 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO1 = 0;
// ---------------
// one action
L7SEG_P1 = 0x04;
L7SEG_NO2 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO2 = 0;
// ---------------
// one action
L7SEG_P1 = 0x04;
L7SEG_NO3 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
```

```
L7SEG_NO3 = 0;
// ---------------
// one action
L7SEG_P1 = 0x04;
L7SEG_NO4 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO4 = 0;
// ---------------
// one action
L7SEG_P1 = 0x04;
L7SEG_NO5 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO5 = 0;
// ---------------
// one action
L7SEG_P1 = 0x04;
L7SEG_NO6 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO6 = 0;
// ---------------
// ------------------ SEGMENT g ------------------
// one action
L7SEG_P1 = 0x02;
L7SEG_NO1 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO1 = 0;
// ---------------
// one action
L7SEG_P1 = 0x02;
L7SEG_NO2 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO2 = 0;
// ---------------
// one action
L7SEG_P1 = 0x02;
L7SEG_NO3 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO3 = 0;
// ---------------
// one action
L7SEG_P1 = 0x02;
L7SEG_NO4 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO4 = 0;
// ---------------
// one action
L7SEG_P1 = 0x02;
L7SEG_NO5 = 1;
tim_usr0 = 30;
while (tim_usr0) ;
L7SEG_NO5 = 0;
```

```
        // ---------------
        // one action
        L7SEG_P1 = 0x02;
        L7SEG_NO6 = 1;
        tim_usr0 = 30;
        while (tim_usr0) ;
        L7SEG_NO6 = 0;
        // ---------------

        // ------------------ SEGMENT dot ------------------
        // Caution!!! I don't know exact reason but using following comman
        // then freeze.
        //      L7SEG_P1 = 0x01;  // = L7SEG_DOT = ON;
        // one action
        L7SEG_DOT = ON;
        L7SEG_NO1 = 1;
        tim_usr0 = 30;
        while (tim_usr0) ;
        L7SEG_NO1 = 0;
        // ---------------
        // one action
        L7SEG_DOT = ON;
        L7SEG_NO2 = 1;
        tim_usr0 = 30;
        while (tim_usr0) ;
        L7SEG_NO2 = 0;
        // ---------------
        // one action
        L7SEG_DOT = ON;
        L7SEG_NO3 = 1;
        tim_usr0 = 30;
        while (tim_usr0){;}
        L7SEG_NO3 = 0;
        // ---------------
        // one action
        L7SEG_DOT = ON;
        L7SEG_NO4 = 1;
        tim_usr0 = 30;
        while (tim_usr0) ;
        L7SEG_NO4 = 0;
        // ---------------
        // one action
        L7SEG_DOT = ON;
        L7SEG_NO5 = 1;
        tim_usr0 = 30;
        while (tim_usr0) ;
        L7SEG_NO5 = 0;
        // ---------------
        // one action
        L7SEG_DOT = ON;
        L7SEG_NO6 = 1;
        tim_usr0 = 30;
        while (tim_usr0) ;
        L7SEG_NO6 = 0;
        // ---------------

        L7SEG_P1 = 0x00;
}
```