```
/************************************************************************
	lcd_drv.c

		April  28,1998 Kenji Arai Porting from Gigo-san's program
		May    28,1998	Fist debuging is finished
		January 6,1999	Move LongToStr to utlty.c
		July   11,2004	CPU is H8/Tiny 3664F (port5)
		July   13,2004	Modify for 4bit interface
		July   22,2004

	Original source file came from;
		mailto:gigo@yk.rim.or.jp
		http://www1.yk.rim.or.jp/~gigo/
************************************************************************/

/************************************************************************/
/* Interface between H8/300 series to HD44780 LCD                       */
/* based on Hitachi A/N APPS/059/1.0(this code can't reset...)          */
/************************************************************************/

/* -----< Include Files >--------------------------------------------- */
#include "3664f.h"
#include "lcd.h"

/* -----< Definition >------------------------------------------------ */
#define WAIT15mS	30	// 15mS
#define WAIT5mS		10	//  5mS
#define WAIT1mS		2	//  1mS

/* -----< Function Prototype >---------------------------------------- */
/*	(just show static function, others are in lcd.h file */
void clk_wait(unsigned);

/* -----< RAM assign >------------------------------------------------ */
char *lcdtxp;
char lcdbuf[82];		// (40+1)*2 + 0
char anthr_lcdbuf[82];		// (40+1)*2 + 0

extern tim_usr0;	// use for wait control,
				// timer0 is drived by RTM (2mS tick)

/* -----< Control program >------------------------------------------- */
void write_lcd(unsigned char data, unsigned char select)
{
	if (select){
		LCD_RS = HIGH;/* set RS signal */
	} else {
		LCD_RS =LOW;
	}
	LCD_RW = WR;		/* set R/W low */
	LCD_DDR = LCD_DDR_OUT;	/* set data port to output */
	LCD_DAT = data/16;	/* write high 4bit data to LCDII */
	LCD_E = HIGH;		/* set E high */
	no_op();
	LCD_E = LOW;		/* set E low */
}

void write2_lcd(unsigned char data, unsigned char select)
```

```
{
	if (select){
		LCD_RS = HIGH;/* set RS signal */
	} else {
		LCD_RS =LOW;
	}
	LCD_RW = WR;		/* set R/W low */
	LCD_DDR = LCD_DDR_OUT;	/* set data port to output */
	LCD_DAT = data/16;	/* write high 4bit data to LCDII */
	LCD_E = HIGH;		/* set E high */
	no_op();
	LCD_E = LOW;		/* set E low */
	LCD_DAT = data;		/* write high 4bit data to LCDII */
	LCD_E = HIGH;		/* set E high */
	no_op();
	LCD_E = LOW;		/* set E low */
}

void wr_lcd(unsigned char data, unsigned char select)
{
	while (rd_lcd(C) & BSYBIT);/* busy wait, busy signal it bit4 */
	if (select){
		LCD_RS = HIGH;/* set RS signal */
	} else {
		LCD_RS =LOW;
	}
	LCD_RW = WR;		/* set R/W low */
	LCD_DDR = LCD_DDR_OUT;	/* set data port to output */
	LCD_DAT = data/16;	/* write high 4bit data to LCDII */
	LCD_E = HIGH;		/* set E high */
	no_op();
	LCD_E = LOW;		/* set E low */
	LCD_DAT = data;		/* write low 4bit data to LCDII */
	LCD_E = HIGH;		/* set E high */
	no_op();
	LCD_E = LOW;		/* set E low */
}

unsigned char rd_lcd (unsigned char select)
{
	unsigned char i;

	LCD_DDR = LCD_DDR_IN;	/* set data port to input */
	if (select){
		LCD_RS = HIGH;/* set RS signal */
	} else {
		LCD_RS =LOW;
	}
	LCD_RW = RD;		/* set R/W high */
	LCD_E = HIGH;		/* set E high */
	no_op();
	i = LCD_DAT;		/* read one byte from lcd */
	LCD_E = LOW;		/* set E low */
	no_op();
	// Dumy read for low 4bit
	LCD_E = HIGH;		/* set E high */
	no_op();
	LCD_E = LOW;		/* set E low */
```

```
        return (i);
}

void reset_lcd(void)
{
        LCD_E = LOW;
        LCD_DDR = LOWBYT;     /* set output */
        LCD_DAT = 0;          /* write 0 */
        LCD_DDR = 0;          /* set input */
        clk_wait(WAIT15mS);
        write_lcd(F_SET,C);
        clk_wait(WAIT5mS);
        write_lcd(F_SET,C);
        clk_wait(WAIT5mS);
        write_lcd(F_SET,C);
        clk_wait(WAIT5mS);
        write_lcd(FUNCTION_SET,C);
        clk_wait(WAIT1mS);
        write2_lcd(FUNCTION_SET,C);
        clk_wait(WAIT1mS);
        write2_lcd(DISP_OFF,C);
        clk_wait(WAIT1mS);
        write2_lcd(DISP_ON,C);
        clk_wait(WAIT1mS);
        write2_lcd(DISP_CLR,C);
        clk_wait(WAIT5mS);
        write2_lcd(ENTRY_MODE,C);
        clk_wait(WAIT1mS);
        return;
}

void put_lcd(const char *p)
{
        while (*p) {
                while (rd_lcd(C) & BSYBIT); /* busy wait */
                if (*p < ' ')
                        write2_lcd(*p++,C);
                else
                        write2_lcd(*p++,D);
        }
}

void set_pos_lcd(unsigned int line,unsigned int pointer)
                                                /* Set cursor position */
{
        unsigned int i;

        while (rd_lcd(C) & BSYBIT);  /* busy wait */
        write_lcd( RETURN_HOME, C );
        if( pointer >= 40)
                pointer = 0;
        if( line >= 2)
                line = 2;
        i = pointer + line*40;
        for( ; i; --i){
                while (rd_lcd(C) & BSYBIT);/* busy wait */
                write2_lcd( CUR_PLS_ONE, C);
        }
```

```
}

// Wait n*500uSec
void clk_wait(unsigned t)
{
	tim_usr0 = t;
	while(tim_usr0) ;		// wait time-up by RTM
}

/* No operation function */
void no_op(void)
{
	;	/* no operation */
}

/*
	Original source
	Copyright (C) 1998, Office SoftBone
	SoftBone Repesentative Tanaka, Toshihide
	ttanaka@cb3.so-net.or.jp
	http://www02.so-net.or.jp/~ttanaka
*/
```