```
/*
  ******************************************************************************
  * @file    lcd_function.c
  * @author  Yoshihito Shimada
  * @version V1.0
  * @date    2011.09.17
  ******************************************************************************
*/

#include "stm32f10x.h"
#include "lcd_function.h"
#include "delay.h"

void LCD_Config(void)                          // LCDをイニシャライズする
{
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA , ENABLE);//| RCC_APB2Periph_GPIOB

  GPIO_InitTypeDef GPIO_InitStructure;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_InitStructure.GPIO_Pin = LCD_DB4 | LCD_DB5 | LCD_DB6 | LCD_DB7;
  GPIO_Init(GPIOA, &GPIO_InitStructure);
  GPIO_InitStructure.GPIO_Pin = LCD_RS | LCD_E;
  GPIO_Init(GPIOA, &GPIO_InitStructure);//B

  GPIO_ResetBits(GPIOA, LCD_RS);           // B LCD RS --->Low (LCD コマンドモードに設定)
  GPIO_ResetBits(GPIOA, LCD_E);            // B LCD E  --->Lowレベル

  delay_ms(15);                            // 約15ms程度のウエイト

  GPIO_SetBits(GPIOA, LCD_E);              // B LCD E  --->Highレベル
  LCD_DB_SetBits(LCD_INIT8B);              // LCD ファンクションセット(データ長8ビット)
  GPIO_ResetBits(GPIOA, LCD_E);            // B LCD E  --->Lowレベル

  delay_ms(5);                             // 約5ms程度のウエイト

  GPIO_SetBits(GPIOA, LCD_E);              // B LCD E  --->Highレベル
  LCD_DB_SetBits(LCD_INIT8B);              // LCD ファンクションセット(データ長8ビット)
  GPIO_ResetBits(GPIOA, LCD_E);            // B LCD E  --->Lowレベル

  delay_ms(1);                             // 約1ms程度のウエイト

  GPIO_SetBits(GPIOA, LCD_E);              // B LCD E  --->Highレベル
  LCD_DB_SetBits(LCD_INIT8B);              // LCD ファンクションセット(データ長8ビット)
  GPIO_ResetBits(GPIOA, LCD_E);            // B LCD E  --->Lowレベル

  delay_ms(1);                             // 約1ms程度のウエイト

  GPIO_SetBits(GPIOA, LCD_E);              // B LCD E  --->Highレベル
  LCD_DB_SetBits(LCD_INIT4B);              // LCD ファンクションセット(データ長4ビット)
  GPIO_ResetBits(GPIOA, LCD_E);            // B LCD E  --->Lowレベル

  write_lcd_data(LCD_FCSET4B, LCD_CMD);          // LCD ファンクションセット(データ長4ビット)
                                           //        N=1(1/16デューティ), F=0(5*8ドット)
  write_lcd_data(LCD_DISP_OFF, LCD_CMD);   // LCD 表示オン／オフコントロールの設定
                                           //        D=0(表示オフ), C=0(カーソルなし), B=0(ブリンクなし)
  write_lcd_data(LCD_CLAR, LCD_CMD);           // LCD 表示クリアの設定
  write_lcd_data(LCD_ENTSET, LCD_CMD);         // LCD エントリモードの設定
                                           //        I/D=1(インクリメント), S=0(表示シフトしない)
  write_lcd_data(LCD_DISP_CUR, LCD_CMD);   // LCD 表示オン／オフコントロールの設定
                                           //        D=1(表示オン), C=1(カーソルあり), B=0(ブリンクなし)
}

void write_lcd_data(uint8_t data, uint8_t rs)
{                                           // LCDコマンド／データ書き込み関数
  uint8_t  upper_data;                     // データコード上位4ビットの格納変数
  uint8_t  lower_data;                     // データコード下位4ビットの格納変数

  upper_data = data >> 4;                  // データコード上位4ビットを抽出
  lower_data = data & 0x0f;                   // データコード下位4ビットを抽出

  delay_ms(100);//500                         // 約500ms程度のウエイト

  if (rs==1) {
      GPIO_SetBits(GPIOA, LCD_RS);         // B LCD DB  --->Highレベル(LCD_DAT)
  } else {
      GPIO_ResetBits(GPIOA, LCD_RS);       // B LCD DB  --->Lowレベル(LCD_CMD)
  }

  GPIO_SetBits(GPIOA, LCD_E);              // B LCD E  --->Highレベル
  LCD_DB_SetBits(upper_data);              // データコード上位4ビットをLCDへ書き込む
  GPIO_ResetBits(GPIOA, LCD_E);            // B LCD E  --->Lowレベル

  delay_ms(1);                             // 約1ms程度のウエイト

  GPIO_SetBits(GPIOA, LCD_E);              // B LCD E  --->Highレベル
  LCD_DB_SetBits(lower_data);              // データコード下位4ビットをLCDへ書き込む
  GPIO_ResetBits(GPIOA, LCD_E);            // B LCD E  --->Lowレベル
}

void LCD_DB_SetBits(uint8_t data_4bits)   // 4ビットデータをDBポートに出力する関数
{
  uint8_t i;
  uint16_t LCD_DB_Pin[]= {LCD_DB4, LCD_DB5, LCD_DB6, LCD_DB7};

  for (i=0; i<=3; i++)
  {
    if ((data_4bits >> i) & 0x01) {
      GPIO_SetBits(GPIOA, LCD_DB_Pin[i]);   // LCD DB  --->Highレベル
    } else {
      GPIO_ResetBits(GPIOA, LCD_DB_Pin[i]); // LCD DB  --->Lowレベル
    }
  }
}

void lcd_puts(uint8_t *str)                  // 文字列表示関数
{                                         //  (文字列をLCDに表示させる)
  while(*str) {                              //
    write_lcd_data(*str, LCD_DAT);          // 文字を順次表示
```

```
    str++;
  }
}

void lcd_xy(uint8_t x, uint8_t y)          // LCD 表示位置を設定する関数
{                                          //   桁(x=1～16), 行(y=1,2) の範囲
  uint8_t  adr;                            //
  adr=((x-1)+(y-1)*0x40) | 0x80;           // アドレスの算出
  write_lcd_data(adr, LCD_CMD);                  // DDRAMにアドレスデータを書き込む
}

void lcd_dataout(uint32_t data)            // 数値データをLCDに表示する関数
{                                          //
  uint8_t temp;                            //
  int8_t strtemp[16];                      //     文字コードの格納変数を定義
  int16_t i, k;                            //
  i=0;                                     //
  do {                                     //
    temp = data % 10;                      //     下位の桁から数字を抽出
    strtemp[i++] = temp + '0';             //     数字を文字コードに変換
  } while (( data /= 10 ) != 0);           //     数値の桁数までループ
  i--;                                     //
  for( k=i; k>=0; k-- ) {                  //     上位の桁から順次
    write_lcd_data( strtemp[k], LCD_DAT); //     数値文字を表示
  }                                        //
}
```