```c
/* Name: David (DongYun) Kim
 * SID: 200405213
 * Course: ENEL351
 * Description: ENEL351 Project - Smart Parking System
 * File name: i2c_lcd_driver.c
 */

/* Rev 2, 2024 */

#include <string.h>
#include "stm32f10x.h"
#include "i2c_lcd_driver.h"
#include "i2c.h"
#include "timer.h"

//Data payload

//D7 DB7
//D6 DB6
//D5 DB5
//D4 DB4
//D3 Backlight
//D2 EN
//D1 RW
//D0 RS


// Write an address and single byte to the i2c inteface

void lcd_i2c_write(uint8_t addr, uint8_t data)
{       i2c_sendbyte(addr, data);

}

//commands defined in *.h

//LCD_CLEARDISPLAY LCD_ENTRYRIGHT   LCD_BACKLIGHT
//LCD_RETURNHOME  LCD_ENTRYLEFT    LCD_NOBACKLIGHT
//LCD_ENTRYMODESET LCD_ENTRYSHIFTINCREMENT
//LCD_DISPLAYCONTROL LCD_ENTRYSHIFTDECREMENT
//LCD_CURSORSHIFT
//LCD_FUNCTIONSET LCD_DISPLAYON    LCD_ENA
//LCD_SETCGRAMADDR LCD_DISPLAYOFF    LCD_RW
//LCD_SETDDRAMADDR LCD_CURSORON    LCD_RS_DAT
//LCD_LN1   LCD_CURSOROFF    LCD_RS_CMD
//LCD_LN2   LCD_BLINKON
//    LCD_BLINKOFF
//
//LCD_DISPLAYMOVE LCD_8BITMODE 0x10
//LCD_CURSORMOVE  LCD_4BITMODE
//LCD_MOVERIGHT   LCD_2LINE
//LCD_MOVELEFT   LCD_1LINE
//    LCD_5x10DOTS
//    LCD_5x8DOTS

void lcd_init(uint8_t addr)
{
   //uint8_t addr = 0x27;
   delay_ms(20);
   lcd_write_cmd(addr, 0x03);
   delay_ms(10);
   lcd_write_cmd(addr, 0x03);
   delay_us(1000);
   lcd_write_cmd(addr, 0x03);
   delay_us(1000);
   lcd_write_cmd(addr, 0x02);
      delay_us(1000);
```

```
      lcd_write_cmd(addr, LCD_FUNCTIONSET | LCD_2LINE | LCD_5x8DOTS | LCD_4BITMODE);
      lcd_write_cmd(addr, LCD_DISPLAYCONTROL | LCD_DISPLAYON | LCD_CURSORON | LCD_BLINKON);
      lcd_write_cmd(addr, LCD_CLEARDISPLAY);
     delay_us(2000);
      lcd_write_cmd(addr, LCD_ENTRYMODESET | LCD_ENTRYLEFT);
   delay_ms(2000);

}

//   # clocks EN to latch command

void lcd_strobe(uint8_t addr, uint8_t data)
{
   lcd_i2c_write(addr , data | LCD_ENA | LCD_BACKLIGHT );
      delay_us(100);
      lcd_i2c_write(addr , ((data & ~LCD_ENA ) | LCD_BACKLIGHT));
      delay_us(100);
}
void lcd_write_four_bits(uint8_t addr, uint8_t data)
{     lcd_i2c_write(addr , (data | LCD_BACKLIGHT));
   delay_us(100);
      lcd_strobe(addr, data);
}
  // # write a command to lcd
void lcd_write_cmd(uint8_t addr, uint8_t data)
{
      lcd_write_four_bits(addr , (LCD_RS_CMD | (data & 0xF0)));
      lcd_write_four_bits(addr , (LCD_RS_CMD | ((data << 4) & 0xF0)));

}
   //# write a character to lcd

void lcd_write_dat(uint8_t addr, uint8_t data)
{
      lcd_write_four_bits(addr , (LCD_RS_DAT | (data & 0xF0)));
      lcd_write_four_bits(addr , (LCD_RS_DAT | ((data << 4) & 0xF0)));
}


void stringToLCD(uint8_t addr,char * message)
{
 int i=0;
 while(message[i] != 0)
  {
  lcd_write_dat(addr, message[i]);
  i++;
  delay_us(1000);
  }
}

//   # clear lcd and set to home
//   def lcd_clear(self):
void lcd_clear_home(uint8_t addr)
{
      lcd_write_cmd(addr, LCD_CLEARDISPLAY);
    delay_us(2000);
      lcd_write_cmd(addr, LCD_RETURNHOME);
    delay_us(2000);

}
```