

Giới Thiệu Về GRAPHIC LCD12864ZX

I. CÁC TÍNH NĂNG CHÍNH.

- Điện áp hoạt động: 4.5V – 5.5V.
- Điện năng tiêu thụ thấp:
 - Chế độ thường: 450uA.
 - Chế độ dự phòng: 30uA.
- Vout tối đa lên đến 7V.
- Tần số hoạt động của LCD có thể điều khiển được bằng 1 biến trở bên ngoài.
- Có thể RESET lại LCD bằng chân RESET ngoài (RST – chân 17).
- Hỗ trợ cho người dùng 3 chế độ giao tiếp với LCD là: 8bit song song, 4bit song song, nối tiếp.
- 64 x 16 bit bộ nhớ RAM dùng để hiển thị dữ liệu (DDRAM – Display Data Ram):
 - Hỗ trợ hiển thị 16 kí tự x 4 dòng.
 - Có thể hiển thị 16 kí tự x 2 dòng.
- 64 x 256 bit bộ nhớ RAM dùng để hiển thị đồ họa (GDRAM – Graphic Display Ram).
- 2M-bits phong kí tự (16x16) trong ROM (CGROM - Character Generation Rom) .
- 16K-bits phong kí tự có chiều rộng bằng ½ chuẩn (16x8) trong ROM (HCGROM – Half-Width Character Generation Rom). Hỗ trợ 126 kí tự.
- Nhiều lệnh điều khiển hiển thị LCD:
 - Display Clear
 - Return Home
 - Display ON/OFF
 - Cursor ON/OFF
 - Display Character Blin
 - Cursor Shift
 - Display Shift
 - Vertical Line Scroll
 - Reverse Display (by line)
 - Standby Mode

II. GIỚI THIỆU CHUNG VỀ CHỨC NĂNG CỦA LCD12864ZX

Mỗi Graphic LCD 12864ZX sử dụng 1 chip điều khiển ST7920. LCD có thể hiển thị bảng chữ cái, chữ số, các phong chữ mà người sử dụng tự thiết lập... Cung cấp cho người sử dụng tới 3 chế độ giao tiếp với LCD là 8bit *song song*, 4 bit *song song*, đồng bộ *nối tiếp*. Hỗ trợ đầy đủ các phong chữ lấy từ **CGROM**, **HCGROM**, **GDRAM**, **CGRAM** (không gian nhớ lưu trữ phong chữ 16x16 mà người sử dụng tự thiết lập). Mức điện áp hoạt động của ST7920 từ 2.7V – 5.5V, mức tiêu thụ điện năng thấp.

III. CẤU TẠO PHẦN CỨNG VÀ CHỨC NĂNG CÁC CHÂN CỦA LCD128x64ZX

Graphic LCD12864ZX có 20 chân, trong đó chân 19 và 20 là Anode và Cathode của LED nền. 18 chân còn lại, có 4 chân cung cấp nguồn và 14 chân điều khiển ... Chức năng các chân được nêu cụ thể ở bảng dưới đây:

GLCD

GRAPHIC LCD12864ZX																				
BLK	BLA	VOUT	RST	NC	PSB	D7	D6	D5	D4	D3	D2	D1	D0	E	R/W	RS	V0	VDD	VSS	
	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

NAME	NO.	I/O	CONNECTS TO	FUNCTION
VSS	1	I	POWER	Vss: 0V
VDD	2	I	POWER	Vdd: 4.5V – 5.5V.
V0	3	I	Biến Trở	Hiệu chỉnh độ tương phản cho LCD.
RS	4	I	MPU	Chế độ song song (4/8 bit) Dùng để lựa chọn thanh ghi: 0: chọn thanh ghi điều khiển . 1: chọn thanh ghi dữ liệu. Chế độ nối tiếp Dùng để chọn chip: 0: disable chip. 1: enable chip.
R\W	5	I	MPU	Chế độ song song (4/8 bit) Dùng để chọn chức năng ghi hay đọc dữ liệu vào chip: 0: ghi dữ liệu vào chip. 1: đọc dữ liệu vào chip. Chế độ nối tiếp Dùng làm chân dữ liệu đầu vào (Data Input).
E	6	I	MPU	Chế độ song song (4/8 bit): Dùng để kích hoạt LCD. Chế độ nối tiếp: Chân nhận xung clock.
D0 – D3	7 - 10	I/O	MPU	4 bit cao của thanh ghi dữ liệu 8 bit.
D4 – D7	11 - 14	I/O	MPU	4 bit thấp của thanh ghi dữ liệu 8 bit.
PSB	15	I	-	PSB = '0': Giao tiếp nối tiếp. PSB = '1': Giao tiếp 4/8 bit.
RST	17	O	-	Reset LCD, tích cực mức thấp.
Vout	18		Biến trở	Vout < 7V

IV. GIAO TIẾP LCD12864ZX VỚI VI ĐIỀU KHIỂN

Dưới đây MinhHagroup xin cung cấp cho các bạn những thiết lập khởi tạo, lệnh điều khiển, và các chế độ giao tiếp với LCD12864ZX cơ bản. Cụ thể sẽ là 2 chế độ giao tiếp 8bit và 4bit **song song** (chế độ **nối tiếp** Minh Hà sẽ giới thiệu với các bạn trong những bài tiếp theo).

Lưu ý khi sử dụng 2 chế độ này để điều khiển LCD là 2 chân PSB (chân 15) và chân RST (chân 17) phải được đặt ở mức '1' (hiểu đơn giản là kéo 2 chân này lên dương nguồn).

Giới Thiệu Về GRAPHIC LCD12864



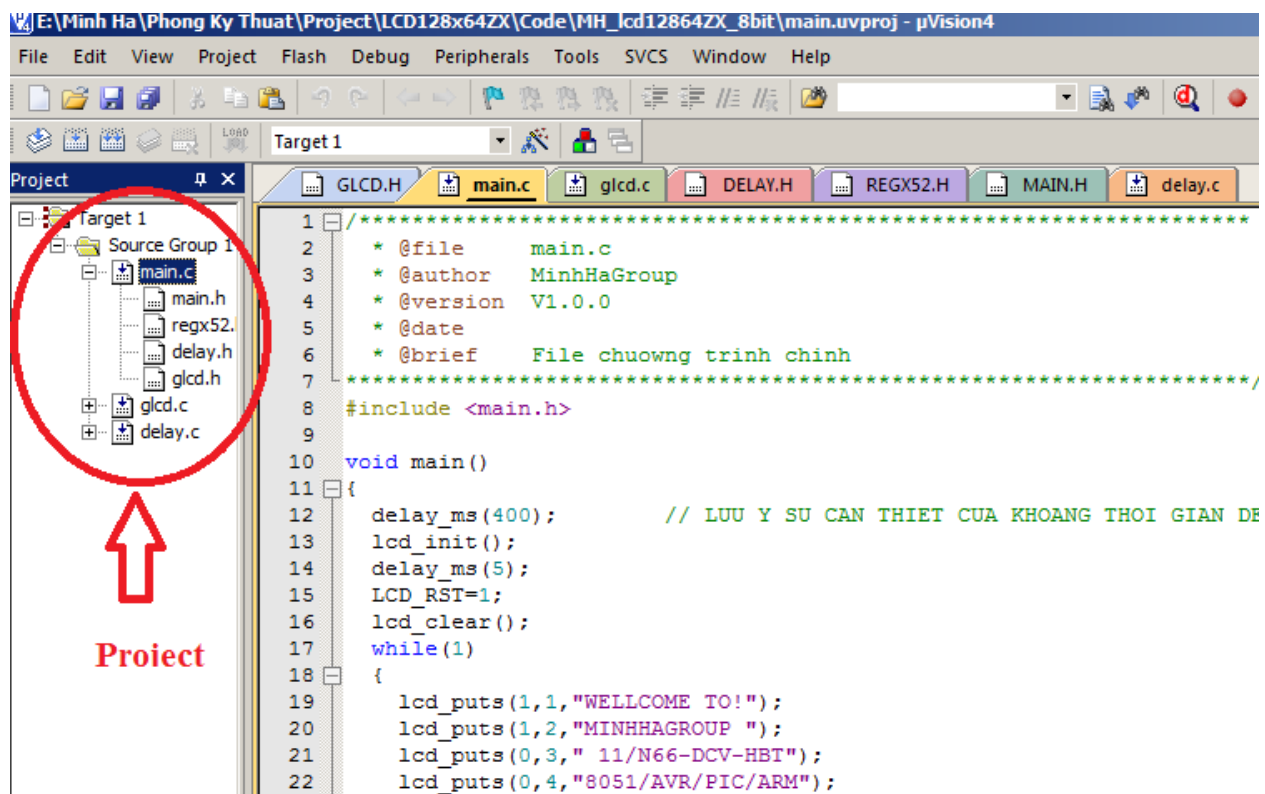
1. CHẾ ĐỘ GIAO TIẾP 8 BIT.

Những chân LCD được sử dụng:

- 3 chân điều khiển: RS, R/W, E.
- 8 chân data: D0 – D7.
- Các chân cung cấp nguồn cho LCD.

Code tham khảo :

Các hàm trong Project



glcd.h:

```
/*
 * @file      glcd.h
 * @author    MinhHaGroup
 * @version   V1.0.0
 * @date
 * @brief     File thu vien khai bao ham cho file glcd.c
 */
```

MinhHaGroup

```

#ifndef __GLCD_
#define __GLCD_

#include <main.h>

// Ham cho phep viet 1 byte du lieu vao LCD -----
void write_data_lcd(unsigned char WDLCD);

// Ham cho phep viet 1 byte lenh dieu khien vao LCD -----
void write_command_lcd(unsigned char WCLCD,unsigned char BuysC);

// Ham cho phep khoi tao de su dung LCD -----
void lcd_init(void);

// Ham cho phep xoa toan bo man hinh LCD -----
void lcd_clear(void);

// Ham cho phep ON/OFF hien thi ra man hinh LCD -----
void lcd_flash(void);

// Ham cho phep viet 1 ki tu ra man hinh LCD -----
void lcd_putchar(unsigned char X, unsigned char Y, unsigned char DData);

// Ham cho phep viet 1 chuoi ki tu ra man hinh LCD -----
void lcd_puts(unsigned char X, unsigned char Y, unsigned char *DData);

// Ham cho phep hien thi hinh anh ra man hinh LCD -----
void lcd_display_image (unsigned char *DData);

// Ham cho phep hien doc du lieu tu LCD ra ngoai -----
//unsigned char read_data_lcd(void);

#endif

```

```

/* -----END OF FILE----- */

```

glcd.c:

```

/*****
 *      @file      glcd.c
 *      @author    MinhHaGroup
 *      @version    V1.0.0
 *      @date
 *      @brief     File cac ham de dieu khien LCD 128x64
 *****/

```

```

*****/
#include "main.h"

void write_data_lcd(unsigned char WDLCD)
{
    delay_us(200);           // LUU Y SU CAN THIET CUA
    KHOANG THOI GIAN DELAY NAY
    LCD_RS = 1;
    LCD_RW = 0;
    LCD_Data = WDLCD;
    LCD_E = 1;
    LCD_E = 1;
    LCD_E = 1;
    LCD_E = 0;
}
//
=====
void write_command_lcd(unsigned char WCLCD,unsigned char BuysC)
{
    if (BuysC)
        delay_us(200);       // LUU Y SU CAN THIET CUA KHOANG THOI GIAN DELAY NAY
    LCD_RS = 0;
    LCD_RW = 0;
    LCD_Data = WCLCD;
    LCD_E = 1;
    LCD_E = 1;
    LCD_E = 1;
    LCD_E = 0;
}
//
=====
// HAM KHOI TAO LCD -----
// MODE 8 bit -----
void lcd_init(void)
{
    // Function set
    write_command_lcd(0x30,1);           // Chon che do truyen 8 bit
    delay_us(150);
    // delay > 100us
    write_command_lcd(0x30,1);           // Chon che do truyen 8 bit
    delay_us(50);
    // delay > 37us

```

```

// Display ON/OFF control
write_command_lcd(0x0C,1);
128x64
delay_us(120);
// delay > 100us

// Clear LCD
write_command_lcd(0x01,1);
delay_ms(40);
// delay > 10ms

// Entry mode set
write_command_lcd(0x06,1);
man hinh LCD 128x64 (Ki tu sau xuất hiện bên phải ki tu trước)
}
//
=====
=====
void lcd_clear(void)
{
    write_command_lcd(0x01,1);
    //write_command_lcd(0x34,1);
    write_command_lcd(0x30,1);
}
//
=====
=====
void lcd_flash(void)
{
    write_command_lcd(0x08,1);
    delay_ms(400);
    write_command_lcd(0x0c,1);
    delay_ms(400);
    write_command_lcd(0x08,1);
    delay_ms(400);
    write_command_lcd(0x0c,1);
    delay_ms(400);
    write_command_lcd(0x08,1);
    delay_ms(400);
}
//
=====
=====
void lcd_putchar(unsigned char X, unsigned char Y, unsigned char DData)
{
    if(Y<1)

```



```

        Y=1;
    if(Y>4)
        Y=4;
    X &= 0x0F;
    switch(Y)
    {
        case 1:X|=0X80;break;
        case 2:X|=0X90;break;
        case 3:X|=0X88;break;
        case 4:X|=0X98;break;
    }
    write_command_lcd(X, 0);
    write_data_lcd(DData);
}
//
=====
=====
void lcd_puts(unsigned char X, unsigned char Y, unsigned char *DData)
{
    unsigned char ListLength,X2;

    ListLength = 0;
    X2=X;
    if(Y<1)
        Y=1;
    if(Y>4)
        Y=4;
    X &= 0x0F;
    switch(Y)
    {
        case 1:X2|=0X80;break;
        case 2:X2|=0X90;break;
        case 3:X2|=0X88;break;
        case 4:X2|=0X98;break;
    }
    write_command_lcd(X2, 1);
    while (DData[ListLength]>=0x20)
    {
        if (X <= 0x0F)
        {
            write_data_lcd(DData[ListLength]);
            ListLength++;
            X++;
            delay_ms(5);
        }
    }
}

```

```

}
//
=====
=====
void lcd_display_image (unsigned char *DData)
{
    unsigned char x,y,i;
    unsigned int tmp=0;
    for(i=0;i<9;)
    {
        for(x=0;x<32;x++)
        {
            write_command_lcd(0x34,1);
            write_command_lcd((0x80+x),1);
            write_command_lcd((0x80+i),1);
            write_command_lcd(0x30,1);
            for(y=0;y<16;y++)
                write_data_lcd(DData[tmp+y]);
            tmp+=16;
        }
        i+=8;
    }
    write_command_lcd(0x36,1);
    write_command_lcd(0x30,1);
}
//
=====
=====
/**
unsigned char read_data_lcd(void)
{
    LCD_RS = 1;
    LCD_RW = 1;
    LCD_E = 0;
    LCD_E = 0;
    LCD_E = 1;
    return(LCD_Data);
}
**

unsigned char read_status_lcd (void)
{
    write_data_port(0xFF);
    LCD_RS = 0;
    LCD_RW = 1;
    LCD_E = 1;

```

```

        while (LCD_Data & BUSY);
        LCD_E = 0;
        return(LCD_Data);
    }*/
    //
    =====
    =====
    /* -----END OF FILE----- */

```

main.h:

```

/******
 *      @file      main.h
 *      @author    MinhHaGroup
 *      @version    V1.0.0
 *      @date
 *      @brief      File cau hinh su dung cho AT89S52
 *****/
#ifndef __MAIN_
#define __MAIN_

#include <regx52.h>
#include <DELAY\delay.h>
#include <GLCD\glcd.h>

/* =====CONFIGURATION===== */
// -----
// PORT0 -----
sbit LCD_RS=P0^0;
sbit LCD_RW=P0^1;
sbit LCD_E=P0^2;
sbit LCD_PSB =P0^3;
sbit LCD_RST=P0^5;

// PORT1 -----
// PORT2 -----
#define LCD_Data P2
#define Busy 0x80

// PORT3 -----

#endif
/* -----END OF FILE----- */

```

main.c:

```

/******

```

```

*      @file          main.c
* @author    MinhHaGroup
* @version   V1.0.0
* @date
* @brief     File chuowng trình chính
*****/
#include <main.h>

void main()
{
    delay_ms(400);          // LUU Y SU CAN THIET CUA KHOANG THOI GIAN DELAY NAY (>40ms)
    lcd_init();
    delay_ms(5);
    LCD_RST=1;
    lcd_clear();
    while(1)
    {
        lcd_puts(1,1,"WELLCOME TO!");
        lcd_puts(1,2,"MINHHAGROUP ");
        lcd_puts(0,3," 11/N66-DCV-HBT");
        lcd_puts(0,4,"8051/AVR/PIC/ARM");
    }
}

/* -----END OF FILE----- */

```

delay.h

```

#ifndef _DELAY_
#define _DELAY_

void delay_ms(unsigned int Time);
void delay_us(unsigned int Time);

#endif

```

delay.c

```

#include <main.h>
void delay_ms(unsigned int Time)
{
    unsigned int i,j,t;
    t=Time;
    for(i=0;i<t;i++)
    {
        for(j=0;j<125;j++);
    }
}
void delay_us(unsigned int Time)

```

```
{  
    unsigned int i,j;  
    for(i=0;i<Time;i++)  
    {  
        for(j=0;j<2;j++);  
    }  
}
```

Để tham khảo Full Project các bạn vui lòng truy cập:

<http://banlinhkien.vn/goods-612-lcd12864zx.html>

2. CHẾ ĐỘ GIAO TIẾP 4 BIT

Những chân LCD được sử dụng:

- 3 chân điều khiển: RS, R/W, E.
- 4 chân data: D4 – D7.
- Các chân cung cấp nguồn cho LCD.

Tương tự như chế độ giao tiếp 8 bit song song ở trên, các bạn có thể tham khảo thêm tài liệu:

Code mẫu chế độ 4 bit:

<http://banlinhkien.vn/goods-612-lcd12864zx.html>

WELLCOME TO MINHHA GROUP.

Thanks For Reading!