

OpenHarmonyOS 基础外设开发--LCD



目 录

CONTENTS

- [01] 什么是LCD
- [02] LCD工作原理
- [03] LCD相关接口
- [04] 如何控制LCD

01

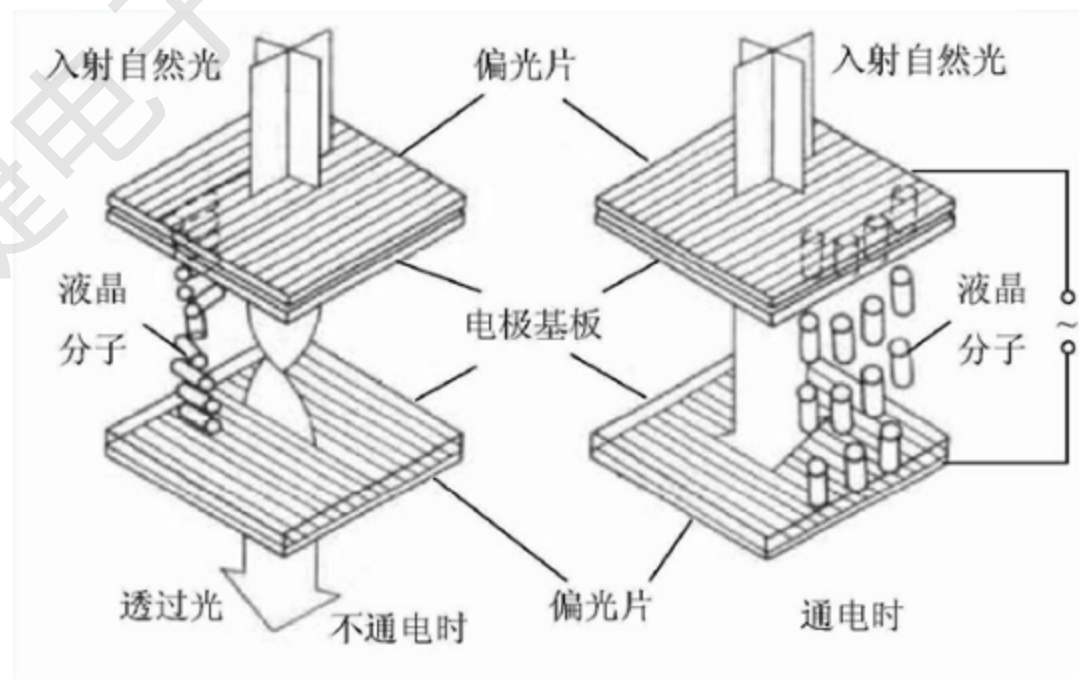
什么是LCD

液晶显示屏，英文简称为LCD，全称是Liquid Crystal Display，是属于平面显示器的一种。现在广泛应用于物联网设备、手机、电视机及计算机的屏幕显示。该显示屏的优点是耗电量低、体积小、辐射低。

02

LCD工作原理

LCD液晶屏显示的基本原理，是将液晶置于两片导电玻璃基板之间，在上下玻璃基板的两个电极作用下，引起液晶分子扭曲变形，改变通过液晶盒光束的偏振状态，实现对背光源光束的开关控制。



03

LCD相关接口

任务接口的头文件

`/vendor/lockzhiner/rk2206/samples/b4_lcd/include/lcd.h`

OpenHarmonyOS内核开发中，任务接口有很多，主要分为几大类：

- (1) 初始化、释放LCD；
- (2) LCD填充；
- (3) LCD绘制画；
- (4) LCD绘制字符；
- (5) LCD绘制图片。

03

任务的接口

功能分类	接口名	功能描述
初始化、释放LCD	lcd_init	lcd液晶屏设备初始化
	lcd_deinit	lcd液晶屏设备注销
LCD填充	lcd_fill	lcd液晶屏指定区域填充颜色
LCD绘制画	lcd_draw_point	lcd液晶屏指定位置画一个点
	lcd_draw_line	lcd液晶屏指定位置画一条线
	lcd_draw_rectangle	lcd液晶屏指定位置画矩形
	lcd_draw_circle	lcd液晶屏指定位置画圆
LCD绘制字符	lcd_show_chinese	lcd液晶屏显示汉字串
	lcd_show_char	lcd液晶屏显示一个字符
	lcd_show_string	lcd液晶屏显示字符串
	lcd_show_int_num	lcd液晶屏显示一个整数
	lcd_show_float_num1	lcd液晶屏显示两位小数变量
LCD绘制图片	lcd_show_picture	lcd液晶屏显示图片

03

LCD相关接口

```
unsigned int lcd_init();
```

该函数主要功能是lcd液晶屏设备初始化。

返回0为成功，反之为失败。

```
unsigned int lcd_deinit();
```

该函数主要功能是lcd液晶屏设备注销。

返回0为成功，反之为失败。

03

LCD相关接口

```
void lcd_fill(uint16_t xsta, uint16_t ysta, uint16_t xend, uint16_t yend, uint16_t color);
```

该函数主要功能是lcd液晶屏指定区域填充颜色。

- 参数xsta: 指定区域的起始点X坐标
- 参数ysta: 指定区域的起始点Y坐标
- 参数xend: 指定区域的结束点X坐标
- 参数yend: 指定区域的结束点Y坐标
- 参数color: 指定区域的颜色

无返回值。

03

LCD相关接口

```
void lcd_draw_point(uint16_t x, uint16_t y, uint16_t color);
```

该函数主要功能是lcd液晶屏设备注销。

- 参数x：指定点的X坐标
- 参数y：指定点的Y坐标
- 参数color：指定点的颜色

无返回值。

03

LCD相关接口

```
void lcd_draw_line(uint16_t x1, uint16_t y1,  
                   uint16_t x2, uint16_t y2,  
                   uint16_t color);
```

该函数主要功能是lcd液晶屏指定位置画一条线。

- 参数x1：指定线的起始点X坐标
- 参数y1：指定线的起始点Y坐标
- 参数x2：指定线的结束点X坐标
- 参数y2：指定线的结束点Y坐标
- 参数color：指定线的颜色

无返回值。

03

LCD相关接口

```
void lcd_draw_rectangle(uint16_t x1, uint16_t y1,  
                        uint16_t x2, uint16_t y2,  
                        uint16_t color);
```

该函数主要功能是lcd液晶屏指定位置画矩形。

- 参数x1：指定矩形的起始点X坐标
- 参数y1：指定矩形的起始点Y坐标
- 参数x2：指定矩形的结束点X坐标
- 参数y2：指定矩形的结束点Y坐标
- 参数color：指定矩形的颜色

无返回值。

03

LCD相关接口

```
void lcd_draw_circle(uint16_t x0, uint16_t y0, uint8_t r, uint16_t color);
```

该函数主要功能是lcd液晶屏指定位置画圆。

- 参数x0：指定圆的中心点X坐标
- 参数y0：指定圆的中心点Y坐标
- 参数r：指定圆的半径
- 参数color：指定圆的颜色

无返回值。

04

如何控制LCD

1、打开sdk下面路径的文件

```
vendor/lockzhiner/rk2206/samples/b4_lcd/lcd_example.c
```

2、创建任务

在lcd_example函数中，通过LOS_TaskCreate函数创建lcd_process任务。

```
task.pfnTaskEntry = (TSK_ENTRY_FUNC)lcd_process;
```

```
task.uwStackSize = 20480;
```

```
task.pcName = "lcd process";
```

```
task.usTaskPrio = 24;
```

```
ret = LOS_TaskCreate(&thread_id, &task);
```

04

如何控制LCD

lcd_process函数初始化LCD设备

```
ret = lcd_init();
```

lcd_process函数每1秒往LCD显示屏填充图片、字符串、整数、浮点数等。

```
while (1)
```

```
{
```

```
    printf("*****Lcd Example*****\n");
```

```
    lcd_show_picture(15, 0, 210, 62, &glimage_lingzhi[0]);
```

```
    lcd_show_string(0, 100, "Welcome to XiaoLingPai!", LCD_RED, LCD_WHITE, 16, 0);
```

```
    lcd_show_string(0, 130, "URL: http://www.fzljda.com", LCD_RED, LCD_WHITE, 16, 0);
```

```
    lcd_show_string(0, 160, "LCD_W:", LCD_BLUE, LCD_WHITE, 16, 0);
```

04

如何控制LCD

```
lcd_show_int_num(128, 160, LCD_H, 3, LCD_BLUE, LCD_WHITE, 16);  
lcd_show_string(80, 160, "LCD_H:", LCD_BLUE, LCD_WHITE, 16, 0);  
lcd_show_string(0, 190, "Increaseing Num:", LCD_BLACK, LCD_WHITE, 16, 0);  
lcd_show_float_num1(128, 190, t, 4, LCD_BLACK, LCD_WHITE, 16);  
t += 0.11;  
lcd_fill(0, 220, LCD_W, LCD_H, LCD_WHITE);  
lcd_show_chinese(0, 220, chinese_string, LCD_RED, LCD_WHITE, cur_sizey, 0);
```

3、修改编译脚本

修改 `vendor/lockzhiner/rk2206/sample` 路径下 `BUILD.gn` 文件，指定 `lcd_example` 参与编译。

```
"/b4_lcd:lcd_example",
```

修改 `device/lockzhiner/rk2206/sdk_liteos` 路径下 `Makefile` 文件，添加 `-llcd_example` 参与编译。

```
hardware_LIBS = -lhal_iohardware -lhardware -llcd_example
```

4、编译固件

```
hb set -root .
```

```
hb set
```

```
hb build -f
```


5、烧写固件

6、通过串口查看结果

运行结果

```
*****Lcd Example*****
```

```
*****Lcd Example*****
```

```
.....
```

谢谢聆听

单击此处添加副标题内容