

OpenHarmonyOS 基础外设开发--OLED



目 录

CONTENTS

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

01

什么是OLED

OLED显示屏是一种由有机分子薄片组成的固态设备，施加电力之后就能发光。OLED能让电子设备产生更明亮、更清晰的图像，其耗电量小于传统的LED显示屏。OLED相较于LCD，有明显的不同：

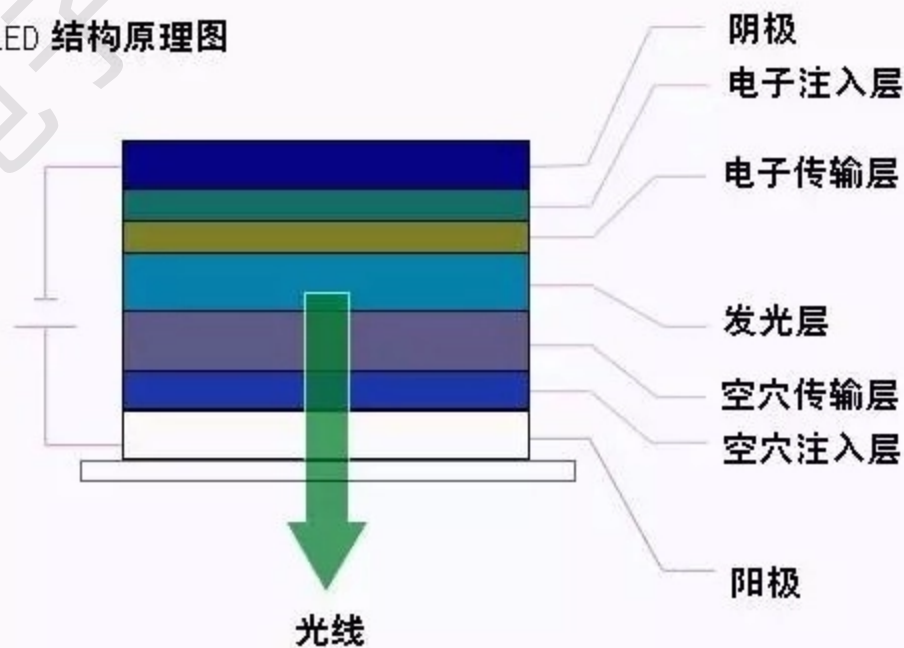
- (1) 相较于LCD，OLED更薄更轻；
- (2) OLED比LED更亮；
- (3) OLED并不需要采用LCD中的背光系统；
- (4) OLED制造起来更加容易，还可制成较大的尺寸；
- (5) OLED的视野范围很广，可达170度左右。

02

OLED工作原理

OLED工作原理是用ITO透明电极和金属电极分别作为器件的阳极和阴极，在一定电压驱动下，电子和空穴分别从阴极和阳极注入到电子和空穴传输层，电子和空穴分别经过电子和空穴传输层迁移到发光层，并在发光层中相遇，形成激子并使发光分子激发，后者经过辐射弛豫而发出可见光。

OLED 结构原理图



03

OLED相关接口

任务接口的头文件

`/vendor/lockzhiner/rk2206/samples/b5_oled/include/oled.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OLED；
- (2) 控制OLED显示。

03

OLED相关接口

功能分类	接口名	功能描述
初始化、释放OLED	oled_init	初始化oled设备
	oled_deinit	释放oled设备
控制OLED的显示	oled_display_on	oled显示开启
	oled_display_off	oled显示关闭
	oled_clear	oled清空屏幕
	oled_show_char	oled显示单个英文字符
	oled_show_num	oled显示数字
	oled_show_string	oled显示英文字符串
	oled_show_bmp	oled显示图片

03

OLED相关接口

```
unsigned int oled_init();
```

该函数主要功能是初始化oled设备。

返回0为成功，反之为失败。

```
unsigned int oled_deinit();
```

该函数主要功能是释放oled设备。

返回0为成功，反之为失败。

03

OLED相关接口

```
void oled_display_on(void);
```

该函数主要功能是oled显示开启。

无返回值。

```
void oled_display_off(void);
```

该函数主要功能是oled显示关闭。

无返回值。

03

OLED相关接口

```
void oled_clear(void);
```

该函数主要功能是oled清空屏幕。

无返回值。

03

OLED相关接口

```
void oled_show_char(uint8_t x, uint8_t y, uint8_t chr, uint8_t chr_size);
```

该函数主要功能是oled显示单个英文字符。

- 参数x：字符的X轴坐标
- 参数y：字符的Y轴坐标
- 参数chr：字符
- 参数chr_size：字符的字体，包括12/16两种字体

无返回值。

03

OLED相关接口

```
void oled_show_num(uint8_t x, uint8_t y, uint32_t num, uint8_t len, uint8_t size);
```

该函数主要功能是oled显示整数。

- 参数x：整数数的X轴坐标
- 参数y：整数数的Y轴坐标
- 参数num：整数
- 参数len：整数的位数
- 参数size：整数字的字体，包括12/16两种字体

无返回值。

03

OLED相关接口

```
void oled_show_string(uint8_t x, uint8_t y, uint8_t *p, uint8_t chr_size);
```

该函数主要功能是oled显示英文字符串。

- 参数x：字符串的X轴坐标
- 参数y：字符串的Y轴坐标
- 参数p：字符串
- 参数chr_size：字符的字体，包括12/16两种字体

无返回值。

03

OLED相关接口

```
void oled_draw_bmp(unsigned char x0, unsigned char y0,  
                   unsigned char x1, unsigned char y1,  
                   unsigned char bmp[]);
```

该函数主要功能是oled显示图片。

- 参数x0: 图片的起始点X轴坐标, 取值为0~127
- 参数y0: 图片的起始点Y轴坐标, 取值为0~63
- 参数x1: 图片的结束点X轴坐标, 取值为0~127
- 参数y1: 图片的结束点Y轴坐标, 取值为0~63
- 参数bmp: 图片

无返回值。

04

如何控制OLED

1、打开sdk下面路径的文件

```
vendor/lockzhiner/rk2206/samples/b5_oled/oled_example.c
```

2、创建任务

在oled_example函数中，通过LOS_TaskCreate函数创建oled_process任务。

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

```
task.uwStackSize = 2048;
```

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

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

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

04

如何控制OLED

oled_process函数初始化oled设备和清空oled屏幕。

```
oled_init();
```

```
oled_clear();
```

oled_process函数每1秒让oled显示屏显示英文字符。

```
while (1)
```

```
{
```

```
    printf("==== Oled Process =====\n");
```

```
    oled_show_string(6, 0, "0.96' OLED TEST", 16);
```

```
    oled_show_string(0, 3, "ASCII:", 16);
```

```
    oled_show_string(64, 3, "CODE:", 16);
```

04

如何控制OLED

```
snprintf(buffer, sizeof(buffer), "%d Sec!", i++);
```

```
oled_show_string(40, 6, buffer, 16);
```

```
printf("\n\n");
```

```
LOS_Msleep(1000);
```

```
}
```


3、修改编译脚本

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

```
"/b5_oled:oled_example",
```

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

```
hardware_LIBS = -lhal_iohardware -lhardware -loled_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

```
===== Oled Process =====
```

```
===== Oled Process =====
```

```
.....
```

谢谢聆听

单击此处添加副标题内容