屏幕的背光亮度可以进行调节，LCD通过PWM进行控制，oled则可以通过命令进行控制。Linux的backlight子系统已经对背光的控制进行了抽象。

# 数据结构

背光设备由backlight\_device结构进行描述，

struct backlight\_device {

struct backlight\_properties props; //背光属性

struct mutex update\_lock;

struct mutex ops\_lock;

const struct backlight\_ops \*ops;//背光的操作函数集

struct notifier\_block fb\_notif;

struct list\_head entry;

struct device dev;

bool fb\_bl\_on[FB\_MAX];

int use\_count;

};

背光属性结构如下：

struct backlight\_properties {

int brightness; //当前背光值

int max\_brightness; //最大背光值

int power;//电源模式 0：全开 1~3：省电 4：关闭

int fb\_blank;//弃用

enum backlight\_type type;//背光控制类型

unsigned int state;//当前状态

};

背光的操作函数集如下：

struct backlight\_ops {

unsigned int options;

#define BL\_CORE\_SUSPENDRESUME (1 << 0)

int (\*update\_status)(struct backlight\_device \*);//更改背光状态

int (\*get\_brightness)(struct backlight\_device \*);//获取背光值

int (\*check\_fb)(struct backlight\_device \*, struct fb\_info \*);//检查是否与指定fb设备绑定

};

# 背光设备的注册

struct backlight\_device \*backlight\_device\_register(const char \*name, struct device \*parent, void \*devdata, const struct backlight\_ops \*ops, const struct backlight\_properties \*props)

name: 在/sys/class/backlight/下创建的背光设备目录名称

parent：父设备

devdata:私有数据

Ops：背光操作函数

Props：背光属性

# 示例说明

Oled可以通过命令进行背光强度的设置，示例中更改背光的update\_status函数即是通过命令对背光的设置。

## 设备树节点

&i2c2 {

pinctrl-names = "default";

pinctrl-0 = <&i2c2\_pins>;

status = "okay";

clock-frequency = <100000>;

backlight:backlight@3c {

compatible = "backlight-test";

reg = <0x3c>;

};

};

## 操作说明

insmod backlight-drv.ko

加载模块后出现/sys/class/backlight/oled\_bl/目录

cat /sys/class/backlight/oled\_bl/brightness

当前背光值

cat /sys/class/backlight/oled\_bl/max\_brightness

背光最大值

echo 255 > /sys/class/backlight/oled\_bl/brightness

设置背光值