

leds-gpio配置说明(linux 4.4内核)

要使用leds-gpio驱动控制led灯,需要配置相应的dts

节点的compatible属性需要设置为"gpio-leds"。

每个LED灯都是以子节点存在的，子节点的名字就是这个LED灯的名字。

LED灯子节点支持一些属性:

- gpios: 表示是哪个GPIO的引脚控制这个LED
- linux,default-trigger: 配置的触发器，触发器可以触发LED灯的状态
- default-state: 初始化默认的状态，有on、off和keep三种设置，on表示开机点亮，off表示开机点灭，keep是保持不操作
- retain-state-suspended: 二级休眠的时候不把LED灯关闭

举例说明:

leds { compatible = "gpio-leds";

```
power-green {
    gpios = <&rk805 0 GPIO_ACTIVE_LOW>;
    linux,default-trigger = "none";
    default-state = "on";
    retain-state-suspended;
};

net-green {
    gpios = <&rk805 1 GPIO_ACTIVE_HIGH>;
    linux,default-trigger = "none";
    default-state = "off";
    retain-state-suspended;
};

ir {
    gpios = <&gpio2 GPIO_C2 GPIO_ACTIVE_LOW>;
    linux,default-trigger = "none";
    default-state = "off";
    retain-state-suspended;
};
```

};

在Rockchip平台中，dts对GPIO的引用分三个部分 第一个部分是表示GPIO的控制器，第二个部分是GPIO的编号，第三部分表示是使能的时候是高有效还是低有效

上面的例子中有3个LED灯，power-green、net-green和ir。power-green是电源灯，开机需要亮灯，显示关闭(一级休眠)的时候需要灭灯 net-green是网络灯，开机需要灭灯，显示关闭(一级休眠)的时候需要灭灯。ir是红外灯，开机需要灭灯，在按下遥控器按键的时候亮灯，弹起遥控器按键的时候灭灯 上面的操作需要通过上层APk在相应的时候控制LED灯，驱动本身不会去控制LED。

power-green的属性

gpios = <&rk805 0 GPIO_ACTIVE_LOW>; rk805是电源管理芯片一些GPIO引脚弥补SOC上引脚的不足，不是SOC上的引脚，rk805本身是通过 I2C进行通信。rk805有2个GPIO引脚，已经注册到系统的GPIO控制器，我们用0和1标识它们，GPIO_ACTIVE_LOW表示是告诉驱动这个GPIO 是低电平点亮LED。

linux,default-trigger = "none"; 不设置触发器，意思是内核没有触发器去控制LED灯。

default-state = "on"; 开机默认点亮LED灯。

retain-state-suspended; 二级休眠的时候，系统不要自作主张把LED灯关闭。

net-green的属性

gpios = <&rk805 1 GPIO_ACTIVE_HIGH>; 这边也使用到了rk805的另外一个GPIO引脚和power-green不同的地方是，这边配置为高电平点亮LED，具体要根据原理图进行修改。

linux,default-trigger = "none"; 不设置触发器，意思是内核没有触发器去控制LED灯。 default-state = "off"; 开机默认点灭LED灯，驱动初始化会关闭LED灯。

retain-state-suspended; 二级休眠的时候，系统不要自作主张把LED灯关闭。

ir的属性

gpios = <&gpio2 GPIO_C2 GPIO_ACTIVE_LOW>; gpio2是SOC的引脚，引脚标号是GPIO_C2, 低电平点亮LED

linux,default-trigger = "none"; 不设置触发器，意思是内核没有触发器去控制LED灯

default-state = "off"; 开机默认点灭LED灯，驱动初始化会关闭LED灯。

retain-state-suspended; 二级休眠的时候，系统不要自作主张把LED灯关闭。

leds-gpio导出到sysfs的接口

通过上面的配置，将会导出sysfs接口到/sys/class/leds目录下，会在这个目录下创建LED的子节点，目录的名字即为子节点的名字。

如下创建了ir、net-green和power-green三个目录 rk_box:/sys/class/leds # ls ir net-green power-green

brightness接口

每个子目录下都有一个brightness文件 rk_box:/sys/class/leds/power-green # ls brightness device max_brightness power subsystem trigger uevent

写1可以点亮led，写0可以点灭led。在shell中可以用echo命令测试，echo 1 > brightness或者echo 0 > brightness 默认的权限是0644 root root, 需要在ueventd.rc文件添加修改权限，否则上层无法操作该文件。

trigger接口

相同的每个子目录下都有一个trigger文件，可以在运行时改变默认的trigger

查看当前trigger rk_box:/sys/class/leds/power-green # cat trigger
[none] mmc0 mmc1 mmc2 backlight default-on

修改trigger为backlight rk_box:/sys/class/leds/power-green # echo backlight > trigger

查看修改后的trigger rk_box:/sys/class/leds/power-green # cat trigger
none mmc0 mmc1 mmc2 [backlight] default-on

查看leds-gpio gpio输出状态

cat /d/gpio可以查看gpio的高低电平状态，结合brightness可以分析电平状态是否和操作一致

rk_box:/sys/class/leds/power-green # cat /d/gpio GPIOs 0-31, platform/pinctrl.27, gpio0:

GPIOs 32-63, platform/pinctrl.27, gpio1:

GPIOs 64-95, platform/pinctrl.27, gpio2: gpio-82 (ir) out lo

GPIOs 96-127, platform/pinctrl.27, gpio3:

GPIOs 254-255, platform/rk8xx-gpio, rk8xx-gpio, can sleep: gpio-254 (power-green) out hi gpio-255 (net-green) out lo