

## AW9106 Android Driver

### 驱动信息

驱动文件	leds-aw9106.c, leds-aw9106.h
芯片支持	AW9106
I <sup>2</sup> C 器件地址	0x58/0x59/0x5a/0x5b
ADB 调试	支持
参考平台	mtk6737

### 驱动移植

#### 1. 修改 dtsi

打开/arch/arm/boot/dts/\*.dtsi 文件，添加 aw9106 的配置。（参考 aw9106.dtsi）

```
&i2c3 {  
    aw9106_led@5b {  
        compatible = "awinic,aw9106_led";  
        reg = <0x5b>;  
        reset-gpio = <&pio 63 0>;  
        status = "okay";  
        aw9106,led {  
            aw9106,name = "aw9106_led";  
            aw9106,imax = <3>;  
            aw9106,brightness = <32>;  
            aw9106,max_brightness = <255>;  
            aw9106,rise_time = <2>;  
            aw9106,on_time = <2>;  
            aw9106,fall_time = <2>;  
            aw9106,off_time = <2>;  
        };  
    };  
};
```

#### 2. 添加驱动文件

在 driver/leds/目录下添加 leds-aw9106.c 文件

在 include/linux 目录下添加 leds-aw9106.h 文件

### 3. 更新 Kconfig 和 Makefile

在 Kconfig 中添加

```
config LEDS_AW9106
    tristate "LED Support for AW9106"
    depends on LEDS_CLASS && I2C
    help
        this option enables support for the AW9106 RGB LED connected
        through I2C. Say Y to enable support for the AW9106 LED
```

在 Makefile 中添加

```
#for AW9106 LED Driver
obj-$(CONFIG_LEDS_AW9106) += leds-aw9106.o
```

### 4. 使能 AW9106 编译

在/arch/arm64/configs/projetc\_defconfig 中添加  
CONFIG\_LEDS\_AW9106=y

## 调试接口

AW9106 Driver 会创建 brightness/blink/hwen/reg/ 4 个设备节点文件，路径是 sys/class/leds/aw9106/。可以使用 adb 配置 brightness/blink/hwen/reg 参数，调试 AW9106 的呼吸效果。

### brightness

用于调节 LED 常亮时的亮度，当 LED 的 brightness=0 时，会关掉 LED。  
brightness 的范围为 0-255。

节点使用：

```
cat brightness (10 进制操作)
echo led_brightness > brightness (10 进制操作)
```

参考例程：

```
echo 255 > brightness
```

### blink

用于 LED 的呼吸使能控制，blink 的范围是 0/1。

节点使用

```
echo 0 > blink (LED 呼吸关闭)
echo 1 > blink (LED 呼吸使能)
```

## **hwen**

用于 AW9106 的芯片硬件控制，hwen 的范围是 0/1。

节点使用

echo 0 > hwen （AW9106 硬件关闭）

echo 1 > hwen （AW9106 硬件使能）

## **reg**

用于读写 AW9106 的所有寄存器。

节点使用：

读寄存器值：cat reg （16 进制操作）

写寄存器值：echo reg\_addr reg\_data > reg （16 进制操作）

参考例程：

echo 0x01 0x02 > reg （向 0x01 寄存器写值 0x02）