

## AW9523B Android Driver

### 驱动信息

驱动文件	aw9523b.c aw9523b.h
芯片支持	aw9523b
I2C 器件地址	0x58/0x59/0x5a/0x5b
ADB 调试	支持
参考平台	mt6765
驱动版本	V1.0.0

### 驱动移植

1. 根据应用配置 aw9523b.dtsi 的内容。

参考的 aw9523b.dtsi 文件是以应用场景为 P0\_0-P0\_3 作为 LED、P0\_4-P0\_7 作为通用 GPIO、P1\_0-P1\_3 口作为矩阵按键的输入（一般会接上拉电阻），P1\_4-P1\_7 作为矩阵按键的扫描端口来进行配置的。用户可以根据实际需要配置设备树的相关属性使 AW9523B 工作在不同的应用场景中。配置规则如下：

1. 配置以下三个属性去开启和关闭矩阵按键、LED、GPIO 的功能。当属性值为 1 时启用功能，为 0 时关闭功能。如果没有使能相关功能，相关的节点可不配置。

**aw9523,key\_enable = <1>;** 该属性关联的节点为 **aw9523,key**

**aw9523,led\_enable = <1>;** 该属性关联的节点为 **aw9523,led**

**aw9523,gpio\_enable = <1>;** 该属性关联的节点为 **aw9523,gpio**

2. 配置独立按键应用的相关属性

**aw9523,input\_port\_mask**

该属性为无符号的 32bit 整数，整数的[bit15-bit0]的比特位分别对应 aw9523b 的 P1\_7-P1\_0-P0\_7-P0\_0 端口，在矩阵按键中用于标识用于触发中断的端口。例如使用 P1\_3-P1\_0 口作为触发中断的输入口，则配置

**aw9523,input\_port\_mask = <0x0F00>;**

**aw9523,output\_port\_mask**

该属性为无符号的 32bit 整数，整数的[bit15-bit0]的比特位分别对应 aw9523b 的 P1\_7-P1\_0-P0\_7-P0\_0 端口，在矩阵按键中用于标识用于扫描的端口。例如使用 P1\_3-P1\_7 口作为扫描的输出口，则配置

**aw9523,output\_port\_mask = <0xF000>;**

3. 配置 LED 应用的节点和属性

驱动会为 **aw9523,led** 节点下的每一个子节点在创建独立的 LED 设备，使每一个 LED 都独立可控。

**aw9523,default\_imax** 该属性值配置全局的 LED 最大输出电流，值为 0x00-0x03

**aw9523,name** 该属性是字符串类型，为 LED 设备的名字

**aw9523,idx\_count** 该属性决定当前节点控制的 LED 数量，数值最大值由 AW9523B 的端口数量决定，最大为 16

**aw9523,idx** 该属性是类型为 32bit 的数组，数组的元素个数由 **aw9523,idx\_count** 决定，数组的每一个元素值标识 AW9523B 的端口，每一个元素值的范围均为 0x0F-0x00，分别对应 aw9523b 的 P1\_7-P1\_0-P0\_7-P0\_0 端口。比如 **aw9523,idx\_count = <2>**；使用的端口为 P0\_0 和 P1\_0。则 aw9523,idx 配置为 **aw9523,idx = <0x00 0x08>**；

**aw9523,default\_brightness** 该属性值决定初始化后 LED 的亮度

**aw9523,max\_brightness** 该属性值决定最大的亮度值

#### 4. 配置 GPIO 应用的节点和属性

当使用 GPIO 功能时，每一个需要用的 GPIO 都需要在 **aw9523,gpio** 节点下创建子节点。

**aw9523,gpio\_mode** 该属性值为 1 时配置 P0 口初始化后为推挽输出，为 0 时配置 P0 口初始化后为开漏输出

**aw9523,gpio\_idx** 该属性的类型为无符号整数，取值为 15-0，分别对应 aw9523B 端口的 P1\_7-P0\_0

**aw9523,gpio\_dir** 该属性表明端口的方向，1 为输出、0 为输入

**aw9523,gpio\_default\_val** 该属性值表明初始化后端口的状态，只在端口设置为输出有效，取值为 0 或 1

#### 5. 除以上和应用相关的属性值需要配置外，中断、复位、上电、器件地址等也需要根据实际的硬件连接进行配置。

#### 2. 当配置好设备树后，需要将配置好的设备节点信息加载到功能设备树中 打开/arch/arm/boot/dts/\*.dtb 文件，添加 AW9523B 的配置

#### 3. 添加驱动文件

在内核的 driver/leds 目录中添加 aw9523b.c 文件

在内核的 include/linux 目录下添加 aw9523b.h 文件

#### 4. 更新 Kconfig 和 Makefile

在 Kconfig 中添加

config AW9523B

tristate "Support for AW9523B"

depends on I2C && LEDS\_CLASS

在 Makefile 中添加

#for AW9523B Driver

obj-\$(CONFIG\_AW9523B) += aw9523b.o

## 调试及功能接口

在驱动移植完成后，会在系统上创建一些功能节点，主要的节点如下：

### reg

该节点为调试节点，可以使用 **cat reg** 获取可读寄存器的值，也可以通过 **echo** 指令操作可写寄存器的值，Echo 指令的格式为：

**echo addr val > reg**

其中 addr 和 val 都是 16 进制的数值，比如要想向地址为 0x02 寄存器写入 0xff，可以执行的指令为：

**echo 0x02 0xff > reg**

## imax

如果使能 led 应用，会在系统上创建 imax 节点。该节点的主要作用是修改 led 驱动的可调节最大电流值，使用格式为：

***echo imax\_val > imax***

其中 imax\_val 的值为 0x00-0x03，值的意义如下：

0x00 0-Imax

0x01 0-3/4\*Imax

0x02 0-2/4\*Imax

0x03 0-1/4\*Imax

Imax 的典型值为 37mA

## aw9523\_gpio

如果使能 gpio 的应用，会在系统上创建 aw9523\_gpio 节点，该节点主要作用是修改 gpio 的状态，使用格式为：

***echo gpio\_idx direction state > aw9523\_gpio***

**gpio\_idx** 标识需要操作的 gpio 标号，取值为 15-0，对应 P1\_7-P0\_0.和设备树上的 aw9523\_gpio\_idx 属性值对应

**direction** 标识 gpio 的方向，取值为 0 或 1. 0 表输入, 1 表输出

**State** 标识 gpio 的状态，当 direction 为输入时，该值无效。取值为 0 或 1。1 表输出低, 1 表输出高。

比如需要让 P1\_2 输出高电平，则执行：

***echo 10 1 1 > aw9523\_gpio***

## Sys/class/leds/<led-name>/brightness

如果使能 LED 应用，会创建 LED 设备节点，可通过 ***cat brightness*** 和 ***echo brightness\_val > brightness*** 查看和修改 LED 的亮度值。