TinaLinux

LEDC 开发指南

1.0

2019.03.13

文档履历

版本号	日期	制/修订人	内容描述
1.0	2019.03.13	AWA1526	

目录

1. 概述

1.1 编写目的

介绍全志 LEDC 驱动的使用方法，方便 LEDC 驱动维护和应用开发。

1.2 适用范围

Tina Linux-4.9 平台。

1.3 相关人员

LED 驱动和应用开发人员。

2. 模块介绍

2.1 相关术语说明

LED：Light Emitting Diode LEDC：Light Emitting Diode Controller

2.2 源码结构说明

本模块借助于标准 Linux LED 子系统。其代码路径为：

```
tina/lichee/linux4.9/drivers/leds/
```

主要包含以下部分代码：

```
led-core.c：为led子系统的核心文件。

ledtrigger-xxx.c：为trigger相关的文件。

led-sunxi.c：LEDC驱动实现代码。

leds-sunxi.h：定义全志LEDC驱动数据结构。
```

2.3 模块配置说明

1、内核配置

在 tina 根目录下，执行 make menuconfig，配置路径如下：

```
Device Drivers
└─>LED_Support
    └─>LED support for Allwinner platforms
```

└─>trigger(需要用到trigger的时候才需要选择)

操作图示：

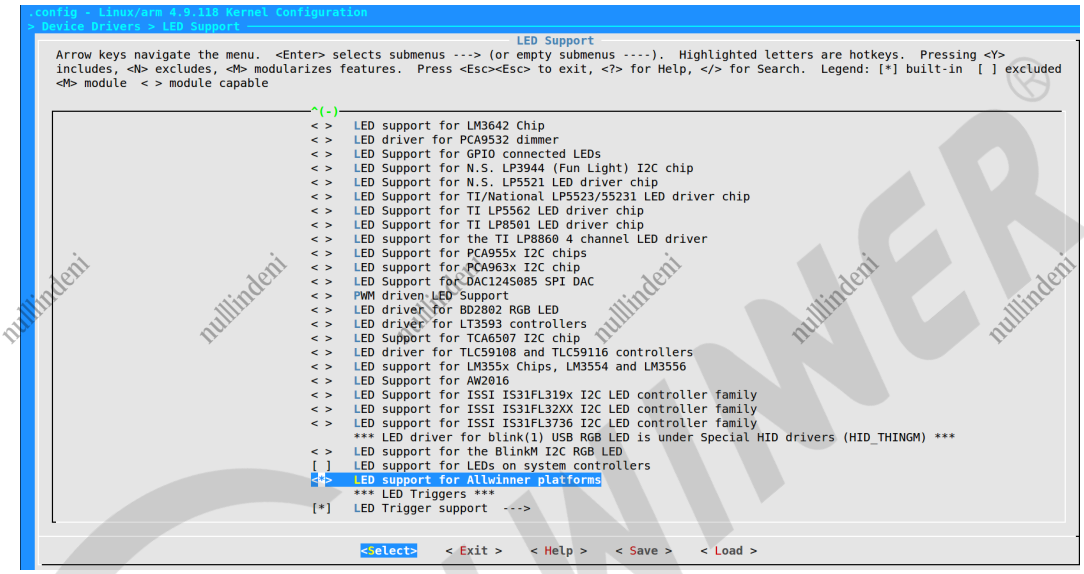

图 1: ledc 配置界面

2、DTS 配置

dts 路径：

tina/lichee/linux-4.9/arch/arm/boot/dts/<平台代号.dts>

```
ledc@06700000 {
        compatible = "allwinner,sunxi-leds";
        reg = <0x0 0x06700000 0x0 0x50>;
        interrupts = <GIC_SPI 60 IRQ_TYPE_LEVEL_HIGH>;
        interrupt-names = "ledcirq";
        clocks = <&clk_ledc>, <&clk_cpuapb>;
        clock-names = "clk_ledc", "clk_cpuapb";
        pinctrl-names = "default";
        pinctrl-0 = <&ledc_pins_a>;
        led_count    = <32>; -> LED的数量
```

```
        output_mode   = "GRB"; -> 输出模式，默认为GRB
        trans_mode    = "DMA"; -> 数据传输模式，默认为DMA
        reset_ns      = <42>; -> LED reset时间
        t1h_ns        = <42>; -> 1码高电平时间
        t1l_ns        = <42>; -> 1码低电平时间
        t0h_ns        = <42>; -> 0码高电平时间
        t0l_ns        = <42>; -> 0码低电平时间
        wait_time0_ns  = <42>; -> 相邻两个LED数据之间等待的时间
        wait_time1_ns  = <42>; -> 相邻两帧数据之间等待的时间
        wait_data_time_ns = <20000000>; -> LEDC内部FIFO等待数据的时间容忍度
    };
```

3、sys_config.fex 配置

sys_config.fex 路径：

```
tina/target/allwinner/<board>/configs/sys_config.fex
```

```
[ledc]
ledc_used = 1 ->是否使用LEDC
ledc = port:PE02<2><default><default><default> ->LEDC对应的引脚
led_count = 1 ->LED数量
output_mode = "RGB" -> 输出模式，默认为GRB
t1h_ns = 800 -> 1码高电平时间
t1l_ns = 450 -> 1码低电平时间
t0h_ns = 400 -> 0码高电平时间
t0l_ns = 850 -> 0码低电平时间
```

3. 接口描述

3.1 内部接口

LEDC 驱动主要的内部接口如下：

1、sunxi_ledc_set_length：

设置 LED 的数量。

2、sunxi_ledc_set_output_mode：

设置 LEDC 的输出模式（R、G、B 的排布顺序）。

3、sunxi_ledc_set_cpu_mode：

设置 CPU 的传输模式。

4、sunxi_ledc_set_cpu_mode：

设置 DMA 的传输模式。

5、sunxi_ledc_enable：

启动 LEDC。

5、sunxi_ledc_trans_data：

设置 LEDC 相关寄存器；将 RGB 数据搬到 LEDC FIFO 中；启动 LEDC。

6、sunxi_ledc_set_time：

模块初始化时设置 reset_ns、t1h_ns、t1l_ns 等的时间。

7、sunxi_ledc_reset：

将 transmitted_data 置为 0；释放系统资源；对 LEDC 做 soft reset 操作。

8、sunxi_set_led_brightness：

设置 LED 亮度。

9、sunxi_register_led_classdev：

模块初始化时注册 led_classdev 设备。

10、sunxi_unregister_led_classdev：

模块卸载时注销 led_classdev 设备。

3.2 外部接口

1、brightness 调节说明

每个 RGB LED 在/sys/class/leds 目录下对应有 3 个 led_classdev 设备目录，分别如下：

/sys/class/leds/sunxi_led[n]r

/sys/class/leds/sunxi_led[n]g

/sys/class/leds/sunxi_led[n]b

其中 n 表示 LED 的编号，n 最小值为 0。

注意：从 LEDC PIN 端开始数，第一个 LED 的编号为 0，沿着远离 PIN 端的方向 LED 的编号依次递增。

例如，调节第 0 个 LED 的颜色为白光且最亮，操作如下：

echo 255 > /sys/class/leds/sunxi_led0r

echo 255 > /sys/class/leds/sunxi_led0g

echo 255 > /sys/class/leds/sunxi_led0b

2、led trigger 使用说明

通过 ``/sys/class/leds/[device]/trigger" 来设置 trigger 类型。

Trigger 类型有：backlight、camera、cpu、default-on、disk、gpio、heartbeat、mtd、oneshot、panic、timer、transient。

例如设置 trigger 类型为 timer，操作如下：

echo timer > /sys/class/leds/sunxi_led0r/trigger

注意：trigger 类型为 timer 时，默认亮 500ms，灭 500ms。可通过以下节点设置亮和灭持续的时间。

/sys/class/leds/[device]/delay_on

/sys/class/leds/[device]/delay_off

3、debugfs 使用说明

LEDC 相关的 debugfs 文件节点所在目录为/sys/kernel/debug/sunxi_leds，节点说明如下：

reset_ns：通过该节点可设置和读取LED的reset时间，范围为80ns-327us。

t1h_ns：通过该节点可设置和读取1码高电平时间，范围为80ns-2560ns。

t1l_ns：通过该节点可设置和读取1码低电平时间，范围为80ns-1280ns。

t0h_ns：通过该节点可设置和读取0码高电平时间，范围为80ns-1280ns。

t0l_ns：通过该节点可设置和读取1码低电平时间，范围为80ns-2560ns。

wait_time0_ns：通过该节点可设置和读取相邻两个LED数据之间等待的时间，范围为80ns-10us。

wait_time1_ns：通过该节点可设置和读取相邻两帧数据之间等待的时间，范围为80ns-85s。

wait_data_time_ns：通过该节点可设置和读取LEDC内部FIFO等待数据的时间容忍度，范围为80ns-655us。

data：通过该节点可读取data buffer中的数据，即所有LED对应的数据。

output_mode：通过该节点可设置和读取当前输出的模式，输出模式有GRB、GBR、RGB、RBG、BGR和BRG。

trans_mode：通过该节点可设置和读取当前的数据传输模式（CPU或DMA）。

hwversion：通过该节点可查看当前LEDC的硬件版本。

注意：

（1）设置的时间必须在所说明的时间范围内，否则不会做任何操作。

（2）最终设置寄存器之后得到的时间均为 42ns 的整数倍，若通过节点设置的时间不遵循 42ns 的整数倍，则实际所设置的时间为小于该值的最大能够被 42ns 整除的数。例如通过 reset_ns 设置 90ns，则设置成功之后的 LED reset 时间为 84ns。

debugfs 使用举例如下：

```
echo 84 > /sys/kernel/debug/sunxi_leds/reset_ns
cat /sys/kernel/debug/sunxi_leds/data
echo RGB > /sys/kernel/debug/sunxi_leds/output_mode
cat /sys/kernel/debug/sunxi_leds/hwversion
```

4. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.tates nor implies warranty of any kind, including fitness for any particular application.