



RTOS LEDC

开发指南

版本号: 1.0
发布日期: 2020-10-22

版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.7.16	Allwinner	1. 初版



目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 模块介绍	2
2.1 模块功能介绍	2
2.2 相关术语介绍	3
2.3 模块配置介绍	3
2.4 模块源码结构	3
3 模块接口说明	5
3.1 接口列表	5
3.2 接口使用说明	5
3.2.1 LEDC 初始化接口	5
3.2.2 LEDC 数据传输接口	5
3.2.3 LEDC 清除中断接口	6
3.2.4 LEDC 获取中断状态接口	6
3.2.5 LEDC DMA 回调接口	6
3.2.6 LEDC 复位接口	6
3.2.7 LEDC 去初始化接口	6
4 模块使用范例	7
5 FAQ	8

1 前言

1.1 文档简介

介绍 RTOS 中 LEDC 驱动的接口及使用方法，为 LEDC 使用者提供参考。

1.2 目标读者

LEDC 驱动层/应用层开发/使用/维护人员。

1.3 适用范围

表 1-1: 适用产品列表

产品名称	内核版本	驱动文件
V459	Melis	hal_ledc.c
R328	FreeRTOS	hal_ledc.c

2 模块介绍

2.1 模块功能介绍

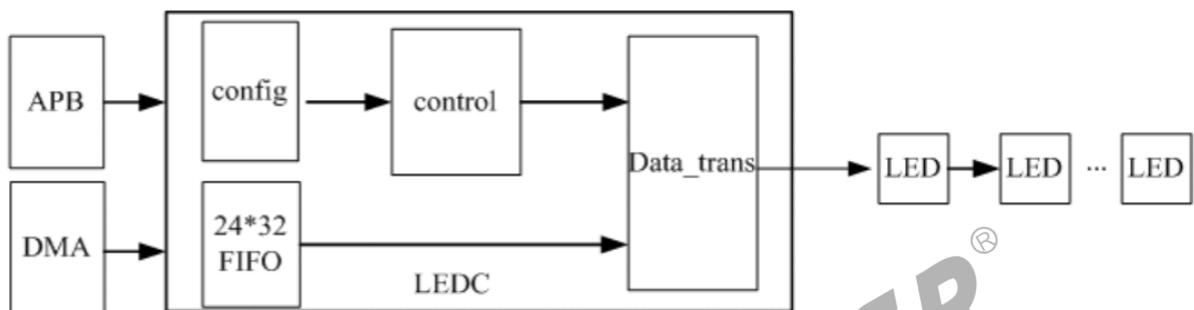
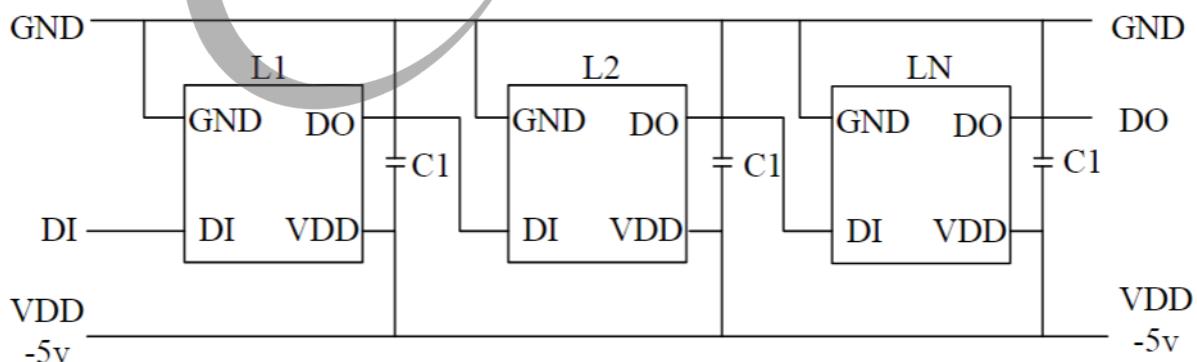


图 2-1: LEDC 硬件方框图

LEDC 硬件方框图如上图所示，CPU 通过 APB 总线操作 LEDC 寄存器来控制 LEDC；当 CPU 配置好 LEDC 的相关寄存器之后，通过 CPU 或 DMA 将 R、G、B 数据从 DRAM 搬到 LEDC FIFO 中，启动 LEDC 之后就可以通过 PIN 脚向外部的 LED 发送数据了。

目前市场上已有成熟的智能外控 LED，每个元件即为一个 LED，每个 LED 的三基色均可实现 256 级亮度显示，因此整个 LED 可完成 256^3 (即 16777216) 种颜色的全真色彩显示。



C1为LED灯的滤波电容，一般取值100NF

图 2-2: LEDC 典型电路

LED 典型电路如上图所示，其中 DI 表示控制数据输入脚，DO 表示控制数据输出脚。DI 端接收从控制器传过来的数据，每个 LED 内部的数据锁存器会存储 24bit 数据，剩余的数据经过内

部整形处理电路整形放大后通过 DO 端口开始转发输出给下一个级联的 LED。因此，每经过一个 LED，数据减少 24bit。

LED 数据传输方式如下图所示：



图 2-3: LEDC 数据传输方式

注意，如果在单次直接设置第 n 个 LED 的亮度和色彩的时候，前面 n-1 个 LED 的亮度数据会在第 n 个 LED 的数据前发送，不过这些数据将会是原来 n-1 个 LED 的亮度数据。

2.2 相关术语介绍

术语	解释说明
Sunxi	指 Allwinner 的一系列 SOC 硬件平台
LED	Light-emitting diode, 发光二极管

2.3 模块配置介绍

```
.config - Melis3.x SDK Configuration
> Kernel Setup > Drivers Setup > SoC HAL Drivers > LEDC Devices -
    LEDC Devices
    Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus
    ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
    <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
    Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
[*] enable ledc driver
```

图 2-4: LEDC menuconfig

2.4 模块源码结构

LEDC 模块源码结构如下所示：

```
rtos-hal/  
|--hal/source/ledc/hal_ledc.c //hal层接口代码  
|--include/hal/sunxi_hal_ledc.h //头文件
```



3 模块接口说明

3.1 接口列表

LEDC 提供的接口列表如下：

```
void hal_ledc_init(void);
void hal_ledc_deinit(void);
void hal_ledc_trans_data(struct ledc_config *ledc);
void hal_ledc_clear_all_irq(void);
unsigned int hal_ledc_get_irq_status(void);
void hal_ledc_dma_callback(void *para);
void hal_ledc_reset(void);
```

3.2 接口使用说明

3.2.1 LEDC 初始化接口

- 原型：int hal_gpadc_init(void)
- 功能：LEDC 模块初始化，主要初始化时钟，GPIO 以及 DMAC 通道等
- 参数：无
- 返回值：无

3.2.2 LEDC 数据传输接口

- 原型：void hal_ledc_trans_data(struct ledc_config *ledc)
- 功能：发送 RGB 数据
- 参数：
 - ledc：配置参数信息，包括待发送数据、数据长度、发送方式（CPU/DMA）及各时间参数设置
- 返回值：无

3.2.3 LEDC 清除中断接口

- 原型: void hal_ledc_clear_all_irq(void)
- 功能: 清除 LEDC 中断
- 参数: 无
- 返回值: 无

3.2.4 LEDC 获取中断状态接口

- 原型: unsigned int hal_ledc_get_irq_status(void)
- 功能: 获取 LEDC 中断状态
- 参数: 无
- 返回值: LEDC 中断状态寄存器值

3.2.5 LEDC DMAC 回调接口

- 原型: void hal_ledc_dma_callback(void *para)
- 功能: 获取 DMAC 状态信息（目前无具体实现）
- 参数: DMAC 状态信息
- 返回值: 无

3.2.6 LEDC 复位接口

- 原型: void hal_ledc_reset(void)
- 功能: 复位 LEDC
- 参数: 无
- 返回值: 无

3.2.7 LEDC 去初始化接口

- 原型: void hal_ledc_deinit(void)
- 功能: LEDC 模块去初始化
- 参数: 无
- 返回值: 无

4 模块使用范例

可参考驱动 APIs 测试代码 (hal/test/ledc/) 。



5 FAQ



著作权声明

版权所有 © 2020 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

  **全志科技**  (不完全列举) 均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。