



RTOS RTC 开发指南

版本号: 1.1
发布日期: 2021.4.27

版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.7.16	AWA1637	1. 初版
1.1	2021.4.27	AWA1637	1. 增加 F133 平台说明



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	2
2.3 模块配置介绍	2
2.4 模块配置介绍	3
2.5 模块源码结构	3
2.6 模块结构说明	3
2.6.1 RTC 时间结构体	4
2.6.2 RTC 闹钟结构体	4
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 RTC 初始化接口	5
3.2.2 获取时间接口	5
3.2.3 设置时间接口	6
3.2.4 获取闹钟接口	6
3.2.5 设置闹钟接口	6
3.2.6 使能闹钟中断接口	6
3.2.7 注册闹钟回调接口	7
4 模块使用范例	8
5 FAQ	9

1 前言

1.1 文档简介

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

1.2 目标读者

RTC 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_RTC.c
F133	Melis	hal_RTC.c
R328	FreeRTOS	hal_RTC.c

2 模块介绍

2.1 模块功能介绍

实时时钟 RTC 用于实现时钟计数和定时唤醒功能，能够实时显示日，时，分，秒功能。该模块电源独立，可以在系统掉电后工作，RTC 具有以下特点。

- 内部具有一个 16bit 的日计数器，5bit 的小时计数器，6bit 的分计数器，6bit 的秒计数器
- 可外接 32768Hz 低频晶振作为计数时钟
- 可随时软件配置初始值
- 具有定时闹钟功能，可产生中断及唤醒外围设备
- 14 个用户寄存器可存放掉电信息
- 多个特殊寄存器记录 BROM 相关信息

2.2 相关术语介绍

术语	解释说明
RTC	实时时钟

2.3 模块配置介绍

RTC 模块寄存器的基本配置位于文件 `rtc_{platform}.h` 里面，包括每个 RTC 的寄存器地址和中断号，部分配置如下：

```
#寄存器基址
#define SUNXI_RTC_BASE      0x07000000
#define SUNXI_RTC_DATA_BASE  (SUNXI_RTC_BASE+0x100)

#中断
#define SUNXI_GIC_START 32
#define SUNXI_IRQ_RTC (SUNXI_GIC_START + 105)
```

2.4 模块配置介绍

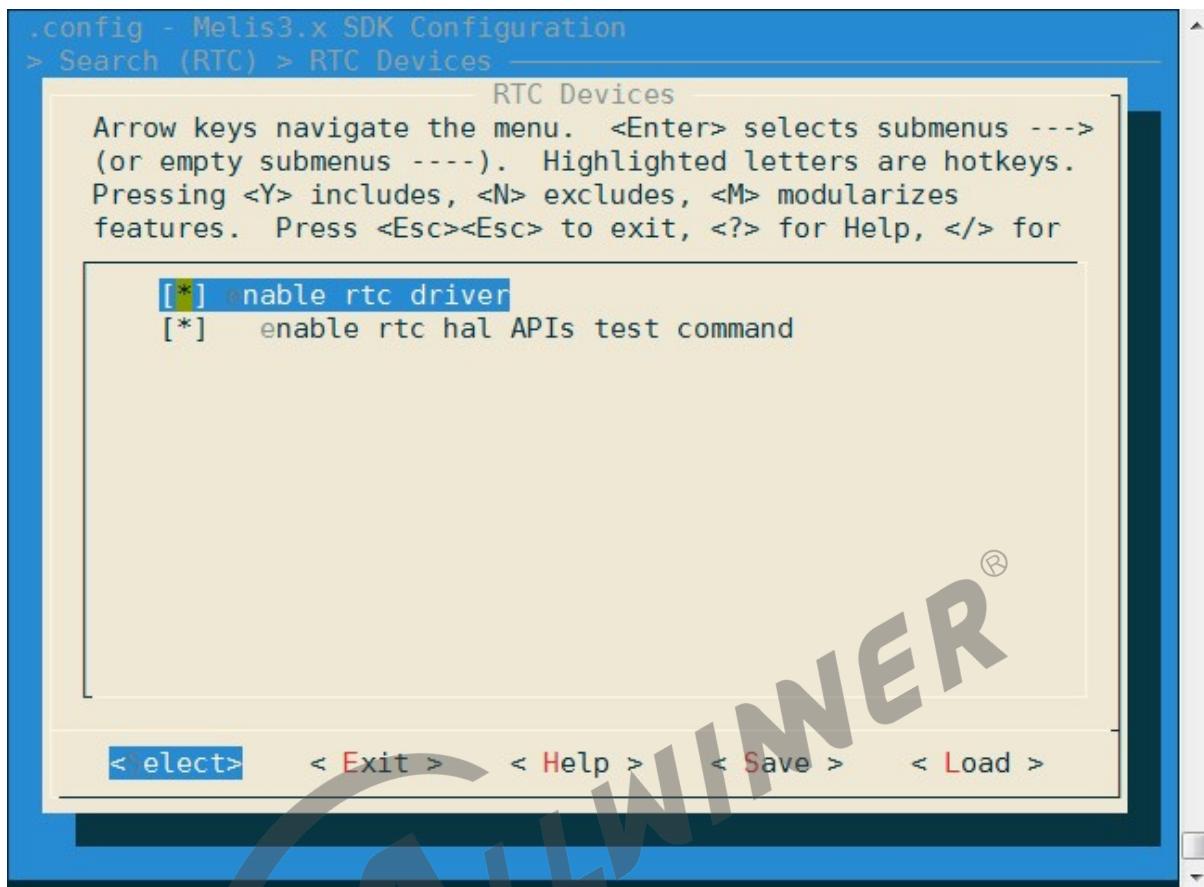


图 2-1: RTC menuconfig

2.5 模块源码结构

RTC 模块源码结构如下所示：

```
rtos-hal/
|--hal/source/rtc/hal_rtc.c //hal层接口代码
|--include/hal/sunxi_hal_rtc.h //头文件
```

2.6 模块结构说明

RTC 模块提供给用户使用的主要包括设置/读取时间，设置/读取闹钟，以及设置闹钟回调接口等等，所以主要讲解一下这几个接口。想要了解更多的数据结构可以查看 sunxi_hal_rtc.h 文件

2.6.1 RTC 时间结构体

该结构体用来保存 RTC 模块的时间，具体如下所示：

```
struct rtc_time
{
    int tm_sec; //秒
    int tm_min; //分
    int tm_hour; //时
    int tm_mday; //天
    int tm_mon; //月
    int tm_year; //年
    int tm_wday;
    int tm_yday;
    int tm_isdst;
};
```

2.6.2 RTC 闹钟结构体

该结构体用来保存 RTC 模块的闹钟时间，具体如下所示：

```
struct rtc_wkalrm
{
    unsigned char enabled; /* 0 = alarm disabled, 1 = alarm enabled */
    unsigned char pending; /* 0 = alarm not pending, 1 = alarm pending */
    struct rtc_time time; /* time the alarm is set to */
};
```

3 模块接口说明

3.1 接口列表

RTC 提供的主要接口列表如下，其余接口可查看 sunxi_hal_rtc.h 文件，使用这些接口需要引入 sunxi_hal_rtc 文件：

```
int hal_rtc_init(void);
int hal_rtc_gettime(struct rtc_time *rtc_tm);
int hal_rtc_settime(struct rtc_time *rtc_tm);
int hal_rtc_getalarm(struct rtc_wkalrm *wkalrm);
int hal_rtc_setalarm(struct rtc_wkalrm *wkalrm);
int hal_rtc_alarm_irq_enable(unsigned int enabled);
int hal_rtc_register_callback(rtc_callback_t user_callback);
```

3.2 接口使用说明

3.2.1 RTC 初始化接口

- 原型：int hal_rtc_init(void);
- 功能：RTC 模块初始化，主要初始化一些工作模式，中断等等
- 参数：无
- 返回值：
 - 0 代表成功
 - 负数代表失败

3.2.2 获取时间接口

- 原型：int hal_rtc_gettime(struct rtc_time *rtc_tm);
- 功能：获取时间值，保存在 rtc_tm 结构体里面
- 参数：
 - rtc_tm，保存时间的结构体参数
- 返回值：
 - 0 代表成功
 - 负数代表失败

3.2.3 设置时间接口

- 原型: int hal_rtc_settime(struct rtc_time *rtc_tm);
- 功能: 设置时间值
- 参数:
 - rtc_tm, 设置时间的结构体参数
- 返回值:
 - 0 代表成功
 - 负数代表失败

3.2.4 获取闹钟接口

- 原型: int hal_rtc_getalarm(struct rtc_wkalrm *wkalrm);
- 功能: 获取闹钟数据
- 参数:
 - wkalrm, 保存闹钟数据的结构体
- 返回值:
 - 0 代表成功
 - 负数代表失败

3.2.5 设置闹钟接口

- 原型: int hal_rtc_setalarm(struct rtc_wkalrm *wkalrm);
- 功能: 设置闹钟数据
- 参数:
 - wkalrm, 保存闹钟数据的结构体
- 返回值:
 - 0 代表成功
 - 负数代表失败

3.2.6 使能闹钟中断接口

- 原型: int hal_rtc_alarm_irq_enable(unsigned int enabled);
- 功能: 使能闹钟中断
- 参数:

- enable:
 - 1, 使能中断
 - 0, 失能中断
- 返回值:
 - 0 代表成功
 - 负数代表失败

3.2.7 注册闹钟回调接口

- 原型: `int hal_rtc_register_callback(rtc_callback_t user_callback:);`
- 功能: 注册闹钟回调接口
- 参数:
 - `user_callback: int callback(void)` 的函数指针
- 返回值:
 - 0 代表成功
 - 负数代表失败

4 模块使用范例

可参考驱动 APIs 测试代码 (hal/test/rtc/test_rtc.c) 。



5 FAQ

RTC 模块记录的时间范围为 1900-2097 年。



著作权声明

版权所有 © 2021 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。