



# **RTOS RTC 开发指南**

**版本号: 1.0**  
**发布日期: 2021.04.11**

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2021.04.11	AWA1637	1. 初版



# 目 录

<b>1 前言</b>	<b>1</b>
1.1 文档简介 . . . . .	1
1.2 目标读者 . . . . .	1
1.3 适用范围 . . . . .	1
<b>2 模块介绍</b>	<b>2</b>
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
<b>3 模块接口说明</b>	<b>4</b>
3.1 接口列表 . . . . .	4
3.2 接口使用说明 . . . . .	4
3.2.1 Timer 初始化接口 . . . . .	4
3.2.2 Timer 失能接口 . . . . .	4
3.2.3 Timer 周期性定时功能 . . . . .	5
3.2.4 Timer 单次定时功能 . . . . .	5
<b>4 模块使用范例</b>	<b>6</b>
<b>5 FAQ</b>	<b>7</b>

# 1 前言

## 1.1 文档简介

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

## 1.2 目标读者

Timer 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_timer.c
F133	Melis	hal_timer.c
R328	FreeRTOS	hal_timer.c

## 2 模块介绍

### 2.1 模块功能介绍

Timer 模块可以用于实现计时，计数功能。其具体规格如下所示

- 计数时钟可配置:LOSC 和 OSC24M
- 可配置 8 中预分频系统
- 可编程 32bit 减法定时器
- 两种工作模式：循环模式和单次计数模式
- 当计数值减到 0 时可以产生中断

### 2.2 相关术语介绍

术语	解释说明
Timer	定时器

### 2.3 模块配置介绍

RTC 模块寄存器的基本配置位于文件 timer\_{platform}.h 里面，包括每个 timer 的寄存器地址和中断号，如 timer\_sun20iw1.h, 配置如下：

```
#define SUNXI_TMR_PBASE 0x02050000

#if defined(CONFIG_ARCH_SUN8IW20) /* ARM */
#define SUNXI_GIC_START 32
#elif defined(CONFIG_SOC_SUN20IW1) /* RISC-V */
#define SUNXI_GIC_START 16
#endif

#define SUNXI_IRQ_TMR(id) (SUNXI_GIC_START + 59 + (id))
```

## 2.4 模块配置介绍

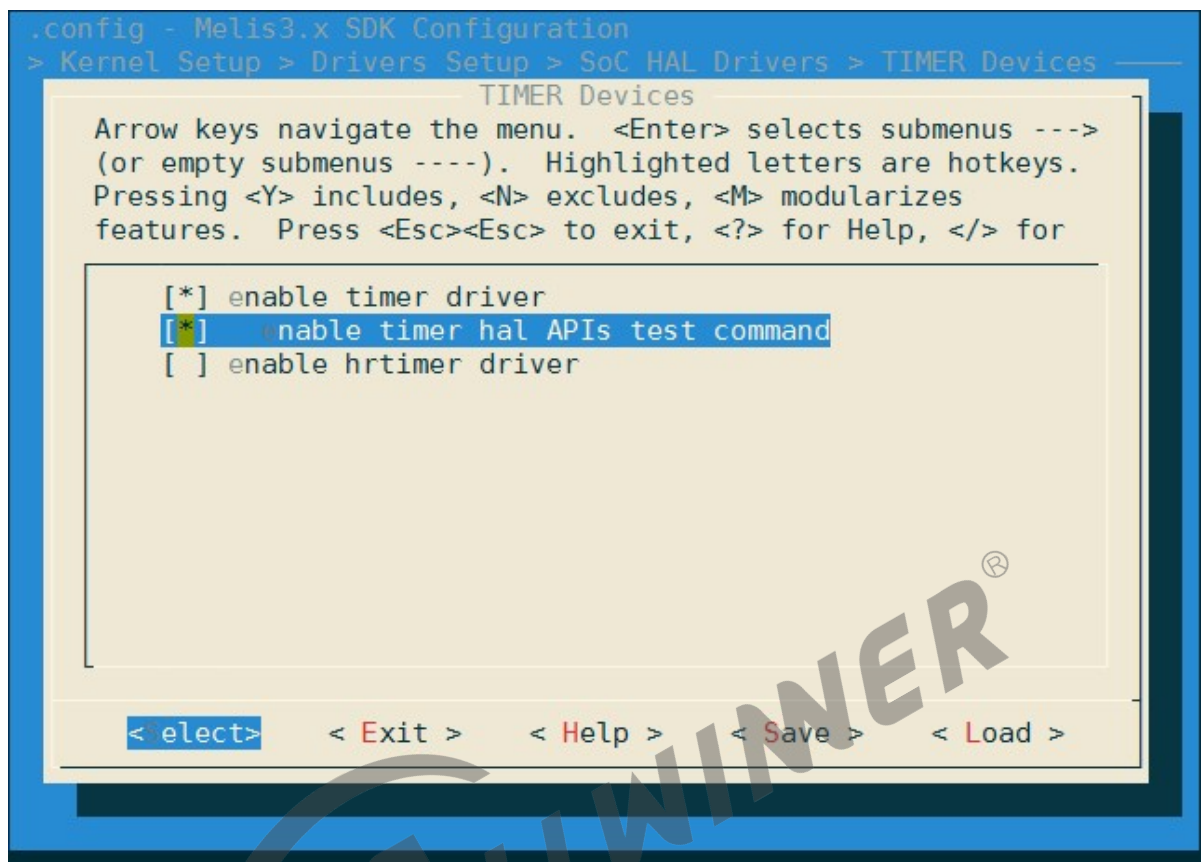


图 2-1: RTC menuconfig

## 2.5 模块源码结构

Timer 模块源码结构如下所示：

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

## 3 模块接口说明

### 3.1 接口列表

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

```
void hal_timer_init(hal_timer_id_t timer);
void hal_timer_uninit(hal_timer_id_t timer);
void hal_timer_stop(hal_timer_id_t timer);
void hal_timer_start(hal_timer_id_t timer, bool periodic);
hal_timer_status_t hal_timer_set_oneshot(hal_timer_id_t timer, uint32_t delay_us,
    timer_callback callback, void *callback_param);
hal_timer_status_t hal_timer_set_periodic(hal_timer_id_t timer, uint32_t delay_us,
    timer_callback callback, void *callback_param);
```

### 3.2 接口使用说明

#### 3.2.1 Timer 初始化接口

- 原型：int hal\_timer\_init(void);
- 功能：Timer 模块初始化，主要初始化一些工作模式，时钟等等
- 参数：无
- 返回值：无

#### 3.2.2 Timer 失能接口

- 原型：int hal\_timer\_uninit(void);
- 功能：Timer 模块失能，主要失能一些工作模式，时钟等等
- 参数：无
- 返回值：无

### 3.2.3 Timer 周期性定时功能

- 原型：hal\_timer\_status\_t hal\_timer\_set\_periodic(hal\_timer\_id\_t timer, uint32\_t delay\_us, timer\_callback callback, void \*callback\_param);
- 功能：设置定时功能，延时 delay\_us us 后响应中断，周期性触发
- 参数：
  - hal\_timer\_id\_t timer: timer 的 id 值，表示使用哪个 timer
  - uint32\_t delay\_us: 延时时间，us
  - timer\_callback callback: 中断回调函数
  - void \*callback\_param: 中断函数传入的参数
- 返回值：无

### 3.2.4 Timer 单次定时功能

- 原型：hal\_timer\_status\_t hal\_timer\_set\_oneshot(hal\_timer\_id\_t timer, uint32\_t delay\_us, timer\_callback callback, void \*callback\_param);
- 功能：设置定时功能，延时 delay\_us us 后响应中断，只触发一次
- 参数：
  - hal\_timer\_id\_t timer: timer 的 id 值，表示使用哪个 timer
  - uint32\_t delay\_us: 延时时间，us
  - timer\_callback callback: 中断回调函数
  - void \*callback\_param: 中断函数传入的参数
- 返回值：无



## 4 模块使用范例

---

可参考驱动 APIs 测试代码（hal/test/timer/test\_timer.c）。



## 5 FAQ

---

无



## 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。