



HAL_V2 HWSPINLOCK 开发指南

版本号: 1.2

发布日期: 2025.7.11

版本历史

版本号	日期	制/修订人	内容描述
1.0	2022.10.28	XAA0250	初始版本
1.1	2024.10.26	AW1942	基于新模板更新文档
1.2	2025.7.11	AW2327	基于 hal_v2 更新文档



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
1.4 文档约定	1
1.4.1 标志说明	1
1.4.2 地址与数据描述方法约定	2
1.4.3 数值单位约定	2
1.5 相关术语介绍	3
1.5.1 软件术语	3
2 模块介绍	4
2.1 模块功能	4
2.2 模块配置	4
2.2.1 menuconfig 配置说明	4
2.3 源码结构介绍	4
2.4 模块配置介绍	5
3 模块接口说明	6
3.1 hal_hwspinlock_put	6
3.2 hal_hwspinlock_get_noblock	6
3.3 hal_hwspinlock_check_taken	7
3.4 hal_hwspinlock_get_block	7
3.5 hal_hwspinlock_get_timeout	7
4 功能开发	8
4.1 开发流程	8
4.2 编程示例	8
5 调试方法	10
5.1 调试节点	10
5.1.1 hal_hwspinlock_get	10
5.1.2 hal_hwspinlock_put	10
6 FAQ	11
6.1 一号通 FAQ 列表	11

1 前言

1.1 文档简介

本文档介绍 AW 平台 RTOS 中 Hwspinlock 驱动接口及使用方法。

1.2 目标读者

- RTOS Hwspinlock 模块开发工程师
- RTOS Hwspinlock 模块使用工程师

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
T153	FreeRTOS	hal_spinlock.c
T536	FreeRTOS	hal_spinlock.c

1.4 文档约定

1 标志说明

注意

- 提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。

 说明

准确理解文中指令、正确实施操作而提供的补充或强调信息。

 技巧

一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

表 1-2: 地址与数据描述方法约定

符号	例子	说明
0x	0x0200, 0x79	地址或数据以 16 进制表示。
0b	0b010, 0b00 000 111	数据采用二进制表示 (寄存器描述除外)。
X	00X, XX1	数据描述中, X 代表 0 或 1。 例如, 00X 代表 000 或 001; XX1 代表 001, 011, 101 或 111。

3 数值单位约定

本文档在描述数据容量 (如 NAND 容量) 时, 单位词头代表的是 1024 的倍数; 描述频率、数据速率等时则代表的是 1000 的倍数。具体如下:

表 1-3: 数值单位约定

类型	符号	对应数值
数据容量 (如 NAND 容	1 K	1024
	1 M	1 048 576
	1 G	1 073 741 824
频率, 数据速率等	1 k	1000
	1 M	1 000 000
	1 G	1 000 000 000

相关术语介绍

1.5.1 软件术语

术语	解释说明
hwspinlock	在多核系统中，hwspinlock 提供一种硬件锁同步机制
HAL	Hardware Abstraction Layer，硬件抽象层



2 模块介绍

2.1 模块功能

hwspinlock 提供一种硬件同步机制，lock 操作可以防止多处理器同时处理共享数据。保证数据的一致性。

2.2 模块配置

2.2.1 menuconfig 配置说明

```
-> Drivers Options
-> soc related device drivers
-> Drivers V2 Config
-> Hwspinlock Driver
    [*] Enable Allwinner Hwspinlock Controller Unit Driver
    [*] Enable HWSPINLOCK hal APIs test command
```

2.3 源码结构介绍

1. 驱动源码

```
hal/source/hwspinlock/
├── hal_spinlock.c
├── Kconfig
├── Makefile
├── platform
│   └── hwspinlock-sun55iw6.h
```

platform-hwspinlock.h

2. 驱动 APIs 声明头文件

```
include/hal/
└── hal_hwspinlock.h
```

3. 驱动 APIs 测试代码

```
hal/examples/hwspinlock/  
├── hwspinlock_block_test.c  
├── hwspinlock_noblock_test.c  
├── hwspinlock_timeout_test.c  
└── Makefile
```

2.4 模块配置介绍

配置路径如下：

```
Symbol: DRIVERS_V2_HWSPINLOCK [=y]  
Type : boolean  
Prompt: Enable Allwinner Hwspinlock Controller Unit Driver  
  
-> Drivers Options  
  -> soc related device drivers  
    -> Drivers V2 Config  
      -> Hwspinlock Driver
```



3 模块接口说明

表 3-1: 模块接口列表

API	解释说明
hwspinlock_put	解锁（非阻塞）
hal_hwspinlock_get_noblock	hwspinlock 上锁（非阻塞）
hal_hwspinlock_check_taken	hwspinlock 获取上锁状态
hal_hwspinlock_get_block	hwspinlock 上锁（阻塞）
hal_hwspinlock_get_timeout	hwspinlock 带超时上锁

3.1 hal_hwspinlock_put

解锁（非阻塞）

- 参数：
 - num：锁序号
- 返回：
 - 非 0：失败
 - 0：成功

hal_hwspinlock_get_noblock

- 作用：hwspinlock 上锁（非阻塞），无论上锁是否成功都会立即返回
- 参数：
 - num：锁序号
- 返回：
 - 非 0：失败
 - 0：成功

hal_hwspinlock_check_taken

- 作用：hwspinlock 检查上锁状态
- 参数：
 - num：锁序号
- 返回：
 - 非 0：已上锁
 - 0：未上锁

hal_hwspinlock_get_block

- 作用：hwspinlock 上锁（阻塞），直到上锁成功才会返回
- 参数：
 - num：锁序号
- 返回：
 - 非 0：失败
 - 0：成功

3.5 hal_hwspinlock_get_timeout

- 作用：hwspinlock 上锁（阻塞），直到上锁成功或超时后才会返回
- 参数：
 - num：锁序号
 - ms_timeout：上锁超时时间
- 返回：
 - 非 0：超时

0：成功

4 功能开发

4.1 开发流程

步骤 1 根据需求调用其中一个接口上锁，int hal_hwspinlock_get_noblock(int num)（非阻塞上锁）/hal_hwspinlock_get_block（阻塞上锁）/hal_hwspinlock_get_timeout（超时上锁）

执行需要保护的代码。

步骤 3 执行 hal_hwspinlock_put 解锁。

4.2 编程示例

1. 测试 demo 路径

lichee/rtos/drivers/hal_v2/hal/examples/hwspinlock/

力调用示例

非阻塞上锁应用

```
void app_lock_unblock(void)
{
    /* hwspinlock0 lock unblock */
    if (hal_hwspinlock_get_noblock(0)) {
        /* hwspinlock0 lock fail */
        return;
    }

    /* code need to protect */
    ...

    /* hwspinlock0 unlock */
    hal_hwspinlock_put(0);
}
```

阻塞上锁应用

```
void app_lock_block(void)
{
    /* hwspinlock0 lock block */
}
```

```
hal_hwspinlock_get_block(0)

/* code need to protect */
...
...
...

/* hwspinlock0 unlock */
hal_hwspinlock_put(0);
}
```

超时上锁应用

```
void app_lock_timeout(void)
{
    /* hwspinlock0 lock 1s timeout */
    if (hal_hwspinlock_get_timeout(0, 1000)) {
        /* hwspinlock0 lock timeout */
        return;
    }

    /* code need to protect */
    ...
    ...
    ...

    /* hwspinlock0 unlock */
    hal_hwspinlock_put(0);
}
```

5 调试方法

5.1 调试节点

5.1.1 hal_hwspinlock_get

_hwspinlock_getXX

- 用途：hwspinlock 上锁测试

使用示例：

```
uart>hal_hwspinlock_get 0  
hwspinlock get success
```

5.1.2 hal_hwspinlock_put

_hwspinlock_putXX

- 用途：hwspinlock 解锁测试

使用示例：

```
uart>hal_hwspinlock_put 0  
hwspinlock put success
```

6 FAQ

6.1 一号通 FAQ 列表

暂无



声明

版权所有 © 2025 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



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。