



HAL_V2 MSGBOX 开发指南

版本号: 1.2

发布日期: 2025.07.16

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	2025.07.16	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
3 模块接口说明	6
3.1 模块配置介绍	6
4 模块接口说明	7
4.1 hal_msgbox_alloc_channel	7
4.2 hal_msgbox_free_channel	7
4.3 hal_msgbox_channel_send	8
5 功能开发	9
5.1 开发流程	9
5.2 编程示例	9
5.2.1 测试 demo 路径	9
5.2.2 驱动调用示例	9
6 测试方法	11
6.1 调试节点	11
6.1.1 hal_msgbox	11
7 FAQ	12
7.1 一号通 FAQ 列表	12

1 前言

1.1 文档简介

本文档介绍 AW 平台 RTOS 中 MSGBOX 驱动接口及使用方法。

1.2 目标读者

- RTOS MSGBOX 模块开发工程师
- RTOS MSGBOX 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
T536	FreeRTOS	msgbox.c
T153	FreeRTOS	msgbox.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
msgbox	在多核系统中，msgbox 提供一种中断通信方式
HAL	Hardware Abstraction Layer，硬件抽象层



2 模块介绍

2.1 模块功能

- msgbox 用来实现多 CPU 之间通讯，在一些 IC 内部可能同时存在多种核心用来实现多种不同功能，这些不同核心跑不同系统，通过 MSGBOX 用来实现这些不同系统间通讯。

msgbox 为一个双端 fifo 结构，cpu0 从一端写，cpu1 从一端读。

- msgbox 为片上处理器之间提供了中断通讯机制

2.2 模块配置

2.2.1 menuconfig 配置说明

> MSGBOX Driver

```
-> Drivers Options
-> soc related device drivers
-> Drivers V2 Config

[*] Enable Allwinner MSGBOX Controller Unit Driver
[*] Enable MSGBOX hal APIs test command
```

2.3 源码结构介绍

```
|—— Kconfig
|—— Makefile
|—— msgbox.c
|—— platform
|   |—— msgbox-sun5iw6.h
|   |—— msgbox-sun8iw22.h
```

platform.h

2. msgb 驱动 APIs 声明头文件

```
include/hal/
|—— hal_msgbox.h
```

3. 驱动 APIs 测试代码

```
hal_v2/hal/examples/msgbox
├── Makefile
├── msgbox_baremetal_alloc_channel
│   ├── Makefile
│   └── msgbox_baremetal_alloc_channel.c
├── msgbox_openamp_alloc_channel
│   ├── Makefile
│   └── msgbox_openamp_alloc_channel.c
```



3 模块接口说明

3.1 模块配置介绍

Symbol: DRIVERS_V2_MSGBOX [=y]

Type : boolean

Prompt: Enable Allwinner MSGBOX Controller Unit Driver

Location:

- > Drivers Options
- > soc related device drivers
- > Drivers V2 Config
- > MSGBOX Driver



4 模块接口说明

表 4-1: 模块接口列表

API	解释说明
msgbox_alloc_channel	申请通道 msgbox
hal_msgbox_channel_send	msgbox 发送
hal_msgbox_free_channel	msgbox 释放通道

4.1 hal_msgbox_alloc_channel

- 作用：msgbox 申请通道
- 参数：
节点，可设置接收回调和发送完成回调
 - remote：远端处理器 id
 - read：硬件接收通道
 - write 硬件发送通道
- 返回：
 - 非 0：失败
 - 0：成功

hal_msgbox_free_channel

- 作用：msgbox 释放通道
- 参数：
 - edp：msgbox 节点
- 返回：
 - void

hal_msgbox_channel_send

- 作用：msgbox 申请通道，可注册接收回调和发送完成回调
- 参数：
 - edp: msgbox 节点
 - bf: 发送 buff 指针
 - len: 发送 buff 长度
- 返回：
 - 非 0: 失败
 - 0: 成功



5 功能开发

5.1 开发流程

步骤 1 调用 `hal_msgbox_alloc_channel` 申请 msgbox 通道，需要指定远端处理器 id，发送通道编号，接收通道编号，设置接收回调函数，发送完成回调函数。

调用 `hal_msgbox_send_data` 发送数据，数据发送完成会触发发送回调函数。

步骤 3 接收到数据时，会触发接收回调函数。

步骤 4 调用 `hal_msgbox_free_channel` 释放 msgbox 通道（无需继续收发时才释放）。

5.2 编程示例

5.2.1 测试 demo 路径

/hal_v2/hal/examples/msgbox

5.2.2 驱动调用示例

```
static void app_msgbox_rcv_cb(uint32_t len, void *data)
{
    /* 接收处理 */
}

static msg_endpoint app_ept = {
    .rec = app_msgbox_rcv_cb,    //接收回调函数

static uint8_t app_buff[] = "hellow world";

void app_msgbox_send_and_rcv(void)
{
    /* 申请msgbox通道，远端处理器编号为1，接收通道为3，发送通道为3 */
    hal_msgbox_alloc_channel(&app_ept, 1, 3, 3);
    /* 发送测试数据 */
    hal_msgbox_channel_send(&app_ept, app_buff, sizeof(app_buff));
    /* 释放msgbox通道 */
    hal_msgbox_free_channel(&app_ept);
}
```

}

 ALLWINER®

6 调试方法

6.1 调试节点

6.1.1 hal_msgbox

1、rtos输入hal_msgbox

uart>hal_msgbox

local:2, remote:0, read_ch:3, write_ch:3, irq:200

local:表示local id为2;

remote: 表示remote id为0;

read_ch, write_ch: 表示读写通道为3, 要与linux dts保持一致

irq:中断号

2、linux往rtos发送消息

linux msgbox driver dts配置可参考:

```
mailbox_test: mailbox_test@0 {  
    compatible = "core0-mailbox-test";  
    mbox-names = "tx|rx0";  
    status = "okay";  
}; /*core与core之间通信*/
```

```
mailbox_test: mailbox_test@0 {  
    compatible = "mailbox-test";  
    mbox-names = "tx|rx";  
    status = "okay";  
}; /*arm与RV之间通信*/
```

3、rtos打印log如下:

uart>recv data:1

sending data:adadadad

7 FAQ

7.1 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



（不全列）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。