



HAL_V2 GMAC 开发指南

版本号: 1.1

发布日期: 2025.8.2

版本历史

版本号	日期	制/修订人	内容描述
1.0	2025.7.15	AWA2351	初始版本
1.1	2025.8.2	AWA2351	修改格式错误



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
1.5 相关术语介绍	2
1.5.1 硬件术语	2
1.5.2 软件术语	2
2 模块介绍	3
2.1 模块功能介绍	3
2.2 模块配置介绍	3
2.3 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 hal_phy_init	5
3.2.2 hal_phy_start	6
3.2.3 hal_dwmac_start	6
3.2.4 hal_dwmac_get_cfg	6
3.2.5 hal_dwmac_set_cfg	6
3.2.6 hal_phy_get_linkstate	7
3.2.7 hal_phy_loopback	7
3.2.8 hal_dwmac_send	7
3.2.9 hal_dwmac_recv	8
4 模块使用范例	9
5 FAQ	11

1 前言

1.1 文档简介

介绍 HAL_V2 中 GMAC 驱动的接口及使用方法，为 GMAC 使用者提供参考。

1.2 目标读者

GMAC 驱动的开发/维护人员。

1.3 适用范围

Table: 适用产品列表

表 1-1: 适用产品列表

产品名称	内核版本	驱动文件
T536	FreeRTOS	hal_dwmac.c、hal_phy.c
T153	FreeRTOS	hal_dwmac.c、hal_phy.c

1.4 文档约定

1.4.1 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。

1.5 相关术语介绍

1.5.1 硬件术语

表 1-2: 硬件术语

术语	解释说明
GMAC	千兆以太网控制器
RGMII	简化的吉比特媒体独立接口
RMII	简化的媒体独立接口
PHY	物理收发器

1.5.2 软件术语

表 1-3: 软件术语

术语	解释说明
HAL	Hardware Abstraction Layer，硬件抽象层

2 模块介绍

2.1 模块功能介绍

GMAC 模块可以用于实现网络数据收发

2.2 模块配置介绍

menuconfig 配置项

```
DRIVERS_V2_ETH
DRIVERS_V2_ETH_DWMAC
HAL_TEST_ETHERNET
```

2.3 模块源码结构

```
hal_v2/hal/source/ethernet/
├── dwmac
│   ├── hal_dwmac.c
│   ├── hal_dwmac.h
│   ├── platform_dwmac.h
│   └── platform
│       ├── dwmac_sun8iw22.h
│       └── dwmac_sun5iw6.h
├── phy
│   ├── hal_phy.c
│   └── hal_phy.h
└── Kconfig
```

2. 驱动 APIs 声明头文件

```
hal_v2/include/hal/
└── hal_ethernet.h
```

3. 驱动 APIs 测试代码

```
hal_v2/hal/examples/ethernet/  
├── ethernet_loopback  
│   ├── ethernet_loopback.c  
│   └── Makefile  
├── ethernet_receive  
│   ├── ethernet_receive.c  
│   └── Makefile  
└── ethernet_transmit  
    ├── ethernet_transmit.c  
    └── Makefile
```



3 模块接口说明

3.1 接口列表

表 3-1: 模块接口列表

API	解释说明
hal_phy_init	PHY 模块初始化
hal_phy_start	PHY 模块启动
hal_dwmac_start	MAC 控制器启动
hal_dwmac_get_cfg	获取 MAC 控制器配置
hal_dwmac_set_cfg	设置 MAC 控制器配置
hal_phy_get_linkstate	PHY 模块 link 状态
hal_phy_loopback	PHY 模块回环模式
hal_dwmac_send	MAC 控制器发送数据
hal_dwmac_recv	MAC 控制器接收数据

3.2 接口使用说明

3.2.1 hal_phy_init

- 作用：PHY 模块初始化
- 参数：
 - phy 代表 PHY 模块句柄
- 返回值：
 - 0 代表成功
 - 非 0 代表失败

3.2.2 hal_phy_start

- 作用：PHY 模块启动
- 参数：
 - phy 代表 PHY 模块句柄
- 返回值：
 - 0 代表成功
 - 非 0 代表失败

3 hal_dwmac_start

- 作用：MAC 控制器启动
- 参数：
 - dwmac 代表 MAC 模块句柄
- 返回值：
 - 0 代表成功
 - 非 0 代表失败

4 hal_dwmac_get_cfg

- 作用：获取 MAC 控制器配置
- 参数：
 - dwmac 代表 MAC 模块句柄
 - dwmac_cfg_type 代表 MAC 模块寄存器类型
 - cfg 代表 MAC 模块配置
- 返回值：
 - 0 代表成功
 - 非 0 代表失败

3.2.5 hal_dwmac_set_cfg

- 作用：设置 MAC 控制器配置
- 参数：
 - dwmac 代表 MAC 模块句柄

- dwmac_cfg_type 代表 MAC 模块寄存器类型

置cfg 代表 MAC

- 返回值：
 - 0 代表成功
 - 非 0 代表失败

3.2.6 hal_phy_get_linkstate

- 作用：获取 PHY 模块 link 状态
- 参数：

模块句柄代表

- 返回值：状态

3.2.7 hal_phy_loopback

- 作用：PHY 模块回环模式
- 参数：
 - phy 代表 PHY 模块句柄

否回环模式PHY

- 返回值：
 - 0 代表成功
 - 非 0 代表失败

3.2.8 hal_dwmac_send

- 作用：MAC 控制器发送数据
- 参数：

MAC 模块句柄

- txcfg 代表发送报文
- timeout 代表发送超时时间

- 返回值：
 - 0 代表成功
 - 非 0 代表失败

3.2.9 hal_dwmac_recv

- 作用：MAC 控制器接收数据
- 参数：
 - dwmac 代表 MAC 模块句柄
 - pbuf 代表接收报文
 - timeout 代表接收超时时间
- 返回值：
 - 0 代表成功
 - 非 0 代表失败



4 模块使用范例

可参考驱动 APIs 测试代码 (hal_v2/hal/examples/ethernet/ethernet_loopback/ethernet_loopback.c) , 具体实现如下:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

de <hal_ethernet.h>

#if defined(CONFIG_KERNEL_FREERTOS)
#include <hal_log.h>
#include <hal_cmd.h>
#include <hal_thread.h>
#include <hal_time.h>
#include <hal_timer.h>
#elif defined(CONFIG_KERNEL_BAREMETAL)
#include "shell.h"
#endif

#define ETH0          0
#define ETH1          1
#define ETH_DMA_TRANSMIT_TIMEOUT 1000000

void *eth_get_initstruct(int port);
extern int board_ethernet_init(int port);
extern void *eth_tx_buf_alloc(struct hal_ethernet *ethernet, int len);

static void ethernet_loopback_task(int port)
{
    struct hal_ethernet *ethernet = eth_get_initstruct(port);
    struct dwmac_instance *dwmac = &ethernet->dwmac;
    struct phy_instance *phy = &ethernet->phy;
    struct dwmac_tx_pkt_cfg tx_pkt_cfg = {0};
    struct dwmac_buffer_list txbuf = {0};
    int cnt = 0;
    int tx_len = 1100;
    int rx_len = 0;
    u8 *rxbuf;
    int i, ret;

    if (phy->state != PHY_UP) {
        printf("ethernet-%d phy is not up\n", port);
        return;
    }

    hal_phy_loopback(phy, 1);
    udelay(1000*1000);

    printf("ethernet-%d loopback task start!!!\n", port);

    while (cnt < 10) {
```

```

txbuf.buf = eth_tx_buf_alloc(ethernet, tx_len);
if (txbuf.buf != NULL)
{
    for (i = 0; i < tx_len; i++)
        txbuf.buf[i] = i;
    txbuf.len = tx_len;
    txbuf.next = NULL;
    tx_pkt_cfg.len = tx_len;
    tx_pkt_cfg.txbuf = &txbuf;

    ret = hal_dwmac_send(dwmac, &tx_pkt_cfg, ETH_DMA_TRANSMIT_TIMEOUT);
    if (ret < 0) {
        printf("ethernet-%d send txbuf timeout %d\n", port, ret);
    }
    printf("ethernet-%d dwmac send buf len %d\n", port, tx_pkt_cfg.len);
}

```

```

rx_len = hal_dwmac_rcv(dwmac, &rxbuf, ETH_DMA_TRANSMIT_TIMEOUT);
if (rx_len < 0) {
    printf("ethernet-%d rcv rxbuf timeout %d\n", port, rx_len);
}
printf("ethernet-%d dwmac rcv buf len %d\n", port, rx_len);

if (rx_len > 0)
{
    if (!memcmp(tx_pkt_cfg.txbuf->buf, rxbuf, rx_len))
        printf("ethernet-%d tx/rx compare same, pass %d\n", port, cnt);
    else
        printf("ethernet-%d tx/rx compare diff, fail %d\n", port, cnt);
}
udelay(1000*1000);
cnt++;
}

```

```

void hal_v2_ethernet_loopback(void)
{
    board_ethernet_init(ETH0);
    ethernet_loopback_task(ETH0);
}

```

```

#if defined(CONFIG_KERNEL_FREERTOS)
FINSH_FUNCTION_EXPORT_CMD(hal_v2_ethernet_loopback, ethernet_loopback_test, ethernet hal_v2 APIs tests)
#elif defined(CONFIG_KERNEL_BAREMETAL)
SHELL_EXPORT_CMD(
SHELL_CMD_PERMISSION(0)|SHELL_CMD_TYPE(SHELL_TYPE_CMD_FUNC)|SHELL_CMD_DISABLE_RETURN,
ethernet_loopback_test, hal_v2_ethernet_loopback, test for ethernet loopback);
#endif

```

5 FAQ

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。