



D1-H Tina Linux 以太网 开发指南

版本号: 1.1

发布日期: 2021.04.29

版本历史

版本号	日期	制/修订人	内容描述
1.0	2021.04.06	AWA1361	Initial Version
1.1	2021.04.29	AWA1381	修改一些符号错误



目 录

1 概述	1
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
2 相关术语介绍	2
3 模块介绍	3
3.1 模块功能介绍	3
3.1.1 以太网简介	3
3.1.2 网络设备框架	3
3.2 模块配置介绍	4
3.2.1 menuconfig 配置说明	4
3.2.2 device tree 配置说明	6
3.2.3 board.dts 配置说明	7
3.2.3.1 RGMII 接口配置	7
3.2.3.2 RMII 接口配置	8
3.3 GMAC 源码结构	8
4 以太网常用调试手段	9
4.1 以太网常用调试命令	9
4.2 以太网通用排查手段	9
4.2.1 常用软件排查手段	9
4.2.2 常用硬件排查手段	10
4.3 以太网常见问题排查流程	10
4.3.1 ifconfig 命令无 eth0 节点	10
4.3.2 ifconfig eth0 up 失败	10
4.3.3 网络不通或网络丢包严重	11
4.3.4 吞吐率异常	11

1 概述

1.1 编写目的

介绍以太网模块配置及调试方法，为以太网模块开发提供参考。

1.2 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
D1-H	Linux-5.4	drivers/net/ethernet/allwinner/*

1.3 相关人员

以太网模块开发/维护人员。

2 相关术语介绍

表 2-1：以太网相关术语介绍

术语	解释说明
SUNXI	Allwinner 一系列 SOC 硬件平台
MAC	Media Access Control，媒体访问控制协议
EMAC	百兆以太网控制器
GMAC	千兆以太网控制器
PHY	物理收发器
MII	Media Independent Interface，媒体独立接口，是 MAC 与 PHY 之间的接口
RMII	简化媒体独立接口
RGMII	简化千兆媒体独立接口

3 模块介绍

3.1 模块功能介绍

3.1.1 以太网简介

以太网是一种局域网通信技术，遵循 IEEE802.3 协议规范，包括 10M、100M、1000M 和 10G 等多种速率的以太网。以太网在 TCP/IP 协议族中的位置如下图所示：

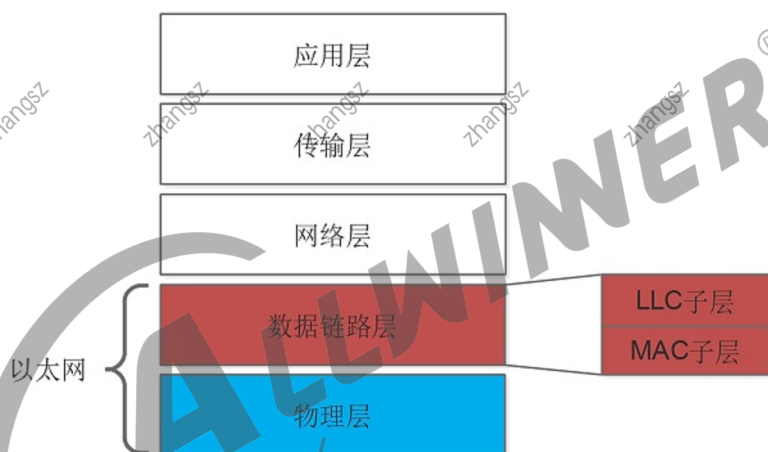


图 3-1: 以太网在 TCP/IP 协议族中的位置

以太网与 TCP/IP 协议族的物理层（L1）和数据链路层（L2）相关，其中数据链路层包括逻辑链路控制（LLC）和媒体访问控制（MAC）子层。

3.1.2 网络设备框架

Linux 内核中网络设备框架如下图所示：

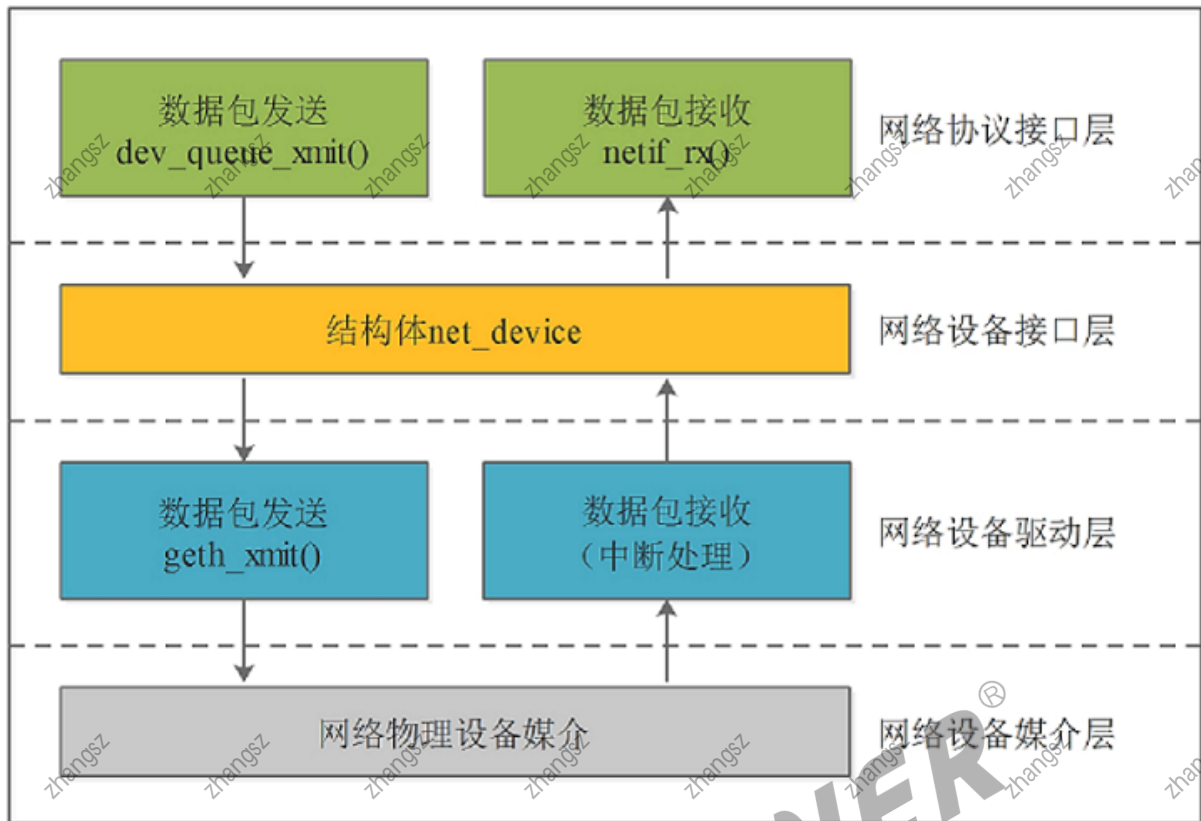


图 3-2: 网络设备框架

从上至下分为 4 层：

- (1) 网络协议接口层：向网络协议层提供统一的数据包收发接口，通过 `dev_queue_xmit()` 发送数据，并通过 `netif_rx()` 接收数据；
- (2) 网络设备接口层：向协议接口层提供统一的用于描述网络设备属性和操作的结构体 `net_device`，该结构体是设备驱动层中各函数的容器；
- (3) 网络设备驱动层：实现 `net_device` 中定义的操作函数指针（通常不是全部），驱动硬件完成相应动作；
- (4) 网络设备媒介层：完成数据包发送和接收的物理实体，包括网络适配器和具体的传输媒介。

3.2 模块配置介绍

3.2.1 menuconfig 配置说明

进入 SDK 根目录，执行 `make kernel_menuconfig` 进入配置界面，按以下步骤操作：

- (1) 配置网络协议栈，如下图所示：

```
.config - Linux/riscv 5.4.61 Kernel Configuration
> Networking support > Networking options

Networking options
Arrow keys navigate the menu. <Enter> selects submenus ---> (or enter
----). Highlighted letters are hotkeys. Pressing <Y> includes, <M>
modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] Packet socket
[*] Packet: sockets monitoring interface
[*] Unix domain sockets
[*] UNIX: socket monitoring interface
< > Transport Layer Security support
< > Transformation user configuration interface
< > PF_KEY sockets
[*] TCP/IP networking
[*] IP: multicasting
[*] IP: advanced router
[*] FIB TRIE statistics
[*] IP: policy routing
[ ] IP: equal cost multipath
[ ] IP: verbose route monitoring
[ ] IP: kernel level autoconfiguration
```

图 3-3: 网络协议栈配置

(2) 勾选 GMAC 驱动，如下图所示：

```
.config - Linux/riscv 5.4.61 Kernel Configuration
> Device Drivers > Network device support > Ethernet driver support

Ethernet driver support
Arrow keys navigate the menu. <Enter> selects submenu
----). Highlighted letters are hotkeys. Pressing <Y>
modularizes features. Press <Esc><Esc> to exit, <?> f
Legend: [*] built-in [ ] excluded <M> module < > mo

--- Ethernet driver support
[*] Alacritech devices
[*] Allwinner devices
< > Allwinner A10 EMAC support
[*] Allwinner GMAC support
[*] Use extern phy
< > Altera Triple-Speed Ethernet MAC support
[*] Amazon Devices
[*] aQuantia devices
[*] ARC devices
```

图 3-4: GMAC 驱动配置

3.2.2 device tree 配置说明

在 device tree 中对 GMAC 控制器进行配置, 一个 GMAC 控制器对应一个 GMAC 设备节点, 如下所示:

```
1  gmac0: eth@4500000 {
2      compatible = "allwinner,sunxi-gmac";
3      reg = <0x0 0x04500000 0x0 0x10000>,
4          <0x0 0x03000030 0x0 0x4>;
5      interrupts-extended = <&plic0 62 IRQ_TYPE_LEVEL_HIGH>;
6      interrupt-names = "gmacirq";
7      clocks = <&ccu CLK_BUS_EMAC0>, <&ccu CLK_EMAC0_25M>;
8      clock-names = "gmac", "ephy";
9      resets = <&ccu RST_BUS_EMAC0>;
10     device_type = "gmac0";
11     pinctrl-0 = <&gmac_pins_a>;
12     pinctrl-1 = <&gmac_pins_b>;
13     pinctrl-names = "default", "sleep";
14     phy-mode;
15     tx-delay = <7>;
16     rx-delay = <31>;
17     phy-rst;
18     gmac-power0;
19     gmac-power1;
20     gmac-power2;
21     status = "disabled";
22 }
```

- (1) "compatible" 表征具体的设备, 用于驱动和设备的绑定;
- (2) "reg" 设备使用的地址;
- (3) "interrupts-extended" 设备使用的中断;
- (4) "clocks" 设备使用的时钟;
- (5) "pinctrl-0" 设备 active 状态下的 GPIO 配置;
- (6) "pinctrl-1" 设备 suspend 状态下的 GPIO 配置;
- (7) "phy-mode" GMAC 与 PHY 之间的物理接口, 如 MII、RMII、RGMII 等;
- (8) "tx-delay" tx 时钟延迟, tx-delay 取值 0-7, 一档约 536ps (皮秒);
- (9) "rx-delay" rx 时钟延迟, rx-delay 取值 0-31, 一档约 186ps (皮秒);
- (10) "phy-rst" PHY 复位脚;
- (11) "gmac-powerX" gmac 电源脚, 根据实际情况配置;
- (12) "status" 是否使能该设备节点;

3.2.3 board.dts 配置说明

3.2.3.1 RGMII 接口配置

对于 RGMII 接口，外挂 RTL8211F PHY 的 EMAC，使用 SOC 内部 EPHY_25M 时钟，支持 10Mbps/100Mbps/1000Mbps 速率。

board.dts 配置范例如下：

路径：device/config/chips/{IC}/configs/{BOARD}/linux-5.4/board.dts

```
1 &gmac0 {  
2     phy-mode = "rgmii";  
3     use_ephy25m = <1>;  
4     pinctrl-0 = <&gmac_pins_a>;  
5     pinctrl-1 = <&gmac_pins_b>;  
6     pinctrl-names = "default", "sleep";  
7     phy-rst = <&pio PE 16 GPIO_ACTIVE_HIGH>;  
8     tx-delay = <4>; /* 2 ~ 4 */  
9     rx-delay = <0>;  
10    status = "okay";  
11 };
```

说明

use_ephy25m=1，代表 PHY 使用 SOC 内部 EPHY_25M 时钟；

use_ephy25m=0 或者不配置该参数，代表 PHY 不使用 SOC 内部 EPHY_25M 时钟，需外挂 25M 晶振为 PHY 提供时钟；

RGMII 接口对时钟和数据波形的相位要求比较严格，因此通常需要调整 **tx-delay** 和 **rx-delay** 参数保证数据传输的正确性。

GMAC0 pinctrl 配置如下：

```
1 gmac_pins_a: gmac@0 {  
2     pins = "PE0", "PE1", "PE2", "PE3",  
3           "PE4", "PE5", "PE6", "PE7",  
4           "PE8", "PE9", "PE10", "PE11",  
5           "PE12", "PE13", "PE14", "PE15";  
6     function = "gmac0";  
7     drive-strength = <10>;  
8 };  
9  
10 gmac_pins_b: gmac@1 {  
11     pins = "PE0", "PE1", "PE2", "PE3",  
12           "PE4", "PE5", "PE6", "PE7",  
13           "PE8", "PE9", "PE10", "PE11",  
14           "PE12", "PE13", "PE14", "PE15";  
15     function = "gpio_in";  
16 };
```

- (1) "pins" 表示使用的 GPIO 管脚；
- (2) "function" 表示复用功能；
- (3) "drive-strength" 表示 GPIO 管脚驱动能力；

3.2.3.2 RMII 接口配置

对于 RMII 接口，外挂 RTL8201F PHY 的 EMAC，使用外挂 25M 晶振，支持 10Mbps/100Mbps 速率。

board.dts 配置范例如下：

```
1 &gmac0 {
2     phy-mode = "rmii";
3     use_ephy25m = <1>;
4     pinctrl-0 = <&gmac_pins_a>;
5     pinctrl-1 = <&gmac_pins_b>;
6     pinctrl-names = "default", "sleep";
7     phy-rst = <&pio PE 16 GPIO_ACTIVE_HIGH>;
8     status = "okay";
9 }
```

GMAC0 pinctrl 配置如下：

```
1 gmac_pins_a: gmac@0 {
2     pins = "PE0", "PE1", "PE2", "PE3",
3           "PE4", "PE5", "PE6", "PE7",
4           "PE8", "PE9";
5     function = "gmac0";
6     drive-strength = <10>;
7 };
8
9 gmac_pins_b: gmac@1 {
10    pins = "PE0", "PE1", "PE2", "PE3",
11          "PE4", "PE5", "PE6", "PE7",
12          "PE8", "PE9";
13    function = "gpio_in";
14 };
```

3.3 GMAC 源码结构

GMAC 驱动的源代码位于内核 drivers/net/ethernet/allwinner 目录下：

```
1 drivers/net/ethernet/allwinner/
2
3 |— sunxi-gmac.h // Sunxi平台GMAC驱动头文件，里面定义了一些宏、数据结构及内部接口
4 |— sunxi-gmac.c // Sunxi平台GMAC驱动核心代码
5 |— sunxi_gmac_ops.c // Sunxi平台GMAC驱动各个内部接口具体实现
```

4 以太网常用调试手段

4.1 以太网常用调试命令

(1) 查看网络设备信息

```
查看网口状态: ifconfig eth0
查看收发包统计: cat /proc/net/dev
查看当前速率: cat /sys/class/net/eth0/speed
```

(2) 打开/关闭网络设备

```
打开网络设备: ifconfig eth0 up
关闭网络设备: ifconfig eth0 down
```

(3) 配置网络设备

```
配置静态IP地址: ifconfig eth0 192.168.1.100
配置MAC地址: ifconfig eth0 hw ether 00:11:22:aa:bb:cc
动态获取IP地址: udhcpc -i eth0
PHY强制模式: ethtool -s eth0 speed 100 duplex full autoneg on (设置100Mbps速率、全双工、开启自协商)
```

(4) 常用测试命令

```
测试设备连通性: ping 192.168.1.100

在吞吐测试前, 需确认勾选iperf工具:
make menuconfig -->Network --> <*> iperf

TCP吞吐测试:
Server端: iperf -s -i 1
Client端: iperf -c 192.168.1.100 -i 1 -t 60 -P 4

UDP吞吐测试:
Server端: iperf -s -u -i 1
Client端: iperf -c 192.168.1.100 -u -b 100M -i 1 -t 60 -P 4
```

4.2 以太网通用排查手段

4.2.1 常用软件排查手段

(1) 检查 phy mode 配置是否正确, 如 rgmii、rmii 等;

- (2) 检查 clk 配置是否正确，如 gmac clk、ephy_25m clk；
- (3) 检查 GPIO 配置是否正确，如 IO 复用功能、驱动能力等；
- (4) 检查 phy reset 配置是否正确；
- (5) 通过 cat /proc/net/dev 命令查看 eth0 收发包统计情况；

4.2.2 常用硬件排查手段

- (1) 检查 phy 供电（vcc-ephy）是否正常；
- (2) 检查 phy 时钟波形是否正常；

4.3 以太网常见问题排查流程

4.3.1 ifconfig 命令无 eth0 节点

问题现象：

执行 ifconfig eth0 无相关 log 信息

问题分析：

以太网模块配置未打开或存在 GPIO 冲突

排查步骤：

- (1) 抓取内核启动 log，检查 gmac 驱动 probe 是否成功；
- (2) 如果无 gmac 相关打印，请参考 3.2 节确认以太网基本配置是否打开；
- (3) 如果 gmac 驱动 probe 失败，请参考 4.2.1 节并结合 log 定位具体原因，常见原因是 GPIO 冲突导致；

4.3.2 ifconfig eth0 up 失败

问题现象：

执行 ifconfig eth0 up，出现 “Initialize hardware error” 或 “No phy found” 异常 log

问题分析：

常见原因是供给 phy 使用的 25M 时钟异常

排查步骤：

- (1) 检查软件 phy_mode 配置与板级情况一致；
- (2) 检查 phy 供电是否正常；
- (3) 若步骤 1 和步骤 2 正常，需重点检查 phy 使用的 25M 时钟（ephy25M 或外部晶振）是否正常；

4.3.3 网络不通或网络丢包严重

问题现象：

ping 不通对端设备、无法动态获取 ip 地址或有丢包现象

问题分析：

一般原因是 tx/rx 通路不通

排查步骤：

- (1) 检查 ifconfig eth0 up 是否正常；
- (2) 检查 eth0 能否动态获取 ip 地址；
- (3) 若步骤 1 正常，但步骤 2 异常，需首先确认 tx/rx 哪条通路不通；
- (4) 若无法动态获取 ip 地址，可配置静态 IP，和对端设备互相 ping；
- (5) 检查对端设备能否收到数据包，若能收到，则说明 tx 通路正常，否则 tx 通路异常；
- (6) 检查本地设备能否收到数据包，若能收到，则说明 rx 通路正常，否则 rx 通路异常；
- (7) 若 tx 通路异常，可调整 tx-delay 参数或对照原理图检查 tx 通路是否异常，如漏焊关键器件；
- (8) 若 rx 通路异常，可调整 rx-delay 参数或对照原理图检查 rx 通路是否异常，如漏焊关键器件；
- (9) 若经过上述排查步骤问题仍未解决，需检查 phy 供电与 GPIO 耐压是否匹配；

4.3.4 吞吐率异常

问题现象：

千兆网络吞吐率偏低，如小于 300Mbps

排查步骤：

- (1) 检查内核有无开启 CONFIG_SLUB_DEBUG_ON 宏，若有，则关闭此宏后再进行测试；
- (2) 如问题仍没有解决，请检查网络是否有丢包、错包现象，若有，参考 4.3.3 进行排查；



著作权声明

版权所有 © 2022 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。