Linux EMAC
开发指南

版本号: 2.9
发布日期: 2025.04.16

版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.6.20	AWA1637	建立初版
1.1	2021.5.10	AWA1637	添加 R528 说明
1.2	2021.8.16	AWA1730	添加 AC200 AC300 说明
1.3	2022.5.16	AWA1730	增加驱动适用范围
1.4	2022.10.28	XAA0250	1. 增加 BSP 独立仓库相关说明 2. 增加 emac 相关说明
1.41	2022.11.28	XAA0250	适用产品列表新增 t113-i
1.5	2023.2.7	XAA0250	新增常用调试工具章节
1.6	2023.10.24	XAA0250	更新 jumbo 帧测试方法
1.7	2023.11.07	AWA2088	增加 gmac-200 驱动使用说明
1.8	2023.11.28	XAA0250	1. 整理常见 FAQ 2. 增加千兆带宽标准测试步骤章节 3. 补充以太网常用调试工具
1.9	2023.12.22	XAA0250	1. 新增以太网 PHY 硬件排查方法 2. 补充内核打印 Link is Up 的条件 3. 按照文档 checklist，刷新一版文档
2.0	2024.2.23	XAA0250	1. 完善文档结构 2. 补充以太网硬件排查方法细节 3. 新增章节跳转
2.1	2024.07.19	AWA1979	1. 新增 MR536 支持 2. 根据驱动更新迭代文档使用方法
2.2	2024.08.29	AWA1979	新增 T536 支持
2.3	2024.10.23	XAA0191	新增 V821 支持
2.4	2024.11.08	AWA1979	新增 H136 支持
2.5	2024.11.13	AWA1979	新增 A733 支持
2.6	2025.2.25	XAA0249	新增 A537、F136、TV323 支持
2.7	2025.04.08	AWA1979	新增 T736 支持
2.8	2025.04.15	AWA1979	新增 H726 支持
2.9	2025.04.16	AWA1979	1. 新增 MR153 支持 2. 产品、内核版本及驱动文件表增加网口数量统计 3. 拆分产品、接口及 PHY 表格，EMAC/GMAC 一组，GMAC2XX 单独一组

目　录

插　图

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: 产品、内核版本及驱动文件

产品名称	内核版本	网口数量	驱动文件
T507	Linux-5.10	2	sunxi-gmac.c
A40I	Linux-5.10	2	sunxi-gmac.c、sunxi-emac.c
A523	Linux-5.15	1	sunxi-gmac.c
T527	Linux-5.15	2	sunxi-gmac.c、dwmac-sunxi.c
MR527	Linux-5.15	1	dwmac-sunxi.c
A527	Linux-5.15	2	sunxi-gmac.c、dwmac-sunxi.c
H728	Linux-5.15	1	dwmac-sunxi.c
T113-i	Linux-5.15	1	sunxi-gmac.c
H618	Linux-5.15	2	sunxi-gmac.c
MR536	Linux-5.15	1	dwmac-sunxi.c
T536	Linux-5.10	2	dwmac-sunxi.c
V821	Linux-5.4	1	sunxi-gmac.c、sunxi-ephy.c
H136	Linux-6.6	2	sunxi-gmac.c
A733	Linux-6.6/Linux-5.15	1	dwmac-sunxi.c
T736	Linux-5.15	2	dwmac-sunxi.c
A537	Linux-5.15	1	dwmac-sunxi.c
F136	Linux-6.6	2	sunxi-gmac.c
H726	Linux-5.15	1	sunxi-gmac.c
TV323	Linux-5.15	1	sunxi-gmac.c、sunxi-ephy.c
MR153	Linux-5.15	1	dwmac-sunxi.c

说明

- **dwmac-sunxi.c 对应 GMAC-2XX 模块**
- **sunxi-gmac.c 对应 GMAC 模块**
- **sunxi-emac.c 对应 EMAC 模块**
- **sunxi-ephy.c 对应内置百兆 EPHY**

表 1-2: EMAC/GMAC 产品、接口及 PHY

产品名称	MAC0	PHY0 接口类型	PHY0 类型	MAC1	PHY1 接口类型	PHY1 类型
T507	GMAC	RGMII	外置	GMAC	RMII	外置
A40I	GMAC	MII/RGMII	外置	EMAC	MII	外置
A523	GMAC	RMII/RGMII	外置	/	/	/
T527	GMAC	RMII/RGMII	外置	/	/	/
A527	GMAC	RMII/RGMII	外置	/	/	/
T113-i	GMAC	RMII/RGMII	外置	/	/	/
H618	GMAC	RGMII	外置	GMAC	RMII	内置
V821	GMAC	RMII	外置/内置	/	/	/
H136	GMAC	RMII/RGMII	外置	GMAC	RMII	外置
F136	GMAC	RMII/RGMII	外置	GMAC	RMII/RGMII	外置
H726	GMAC	RMII/RGMII	外置	/	/	/
TV323	GMAC	RMII/RGMII	内置	/	/	/

表 1-3: GMAC2XX 产品、接口及 PHY

产品名称	MAC0	PHY0 接口类型	MAC1	PHY1 接口类型	MAC2	PHY2 接口类型
T527	/	/	外置	RMII/RGMII	/	/
MR527	/	/	外置	RMII/RGMII	/	/
A527	/	/	外置	RMII/RGMII	/	/
H728	/	/	外置	RMII/RGMII	/	/
MR536	/	/	外置	RMII/RGMII	/	/
T536	外置	RMII/RGMII	外置	RMII/RGMII	/	/
A733	外置	RMII/RGMII	/	/	/	/
T736	外置	RMII/RGMII	外置	RMII/RGMII	/	/
A537	外置	RMII/RGMII	/	/	/	/
MR153	外置	RMII/RGMII	/	/	/	/

说明

- **百兆对应 RMII/MII，向下兼容 10Mbps，千兆对应 RGMII，向下兼容 100Mbps/10Mbps**
- **外置指需要在板级外挂以太网 PHY 芯片，如 RTL8021/RTL8211 等**
- **内置指 SoC 集成内部百兆 EPHY，无需在板级外挂**

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
GMAC/GMAC-2XX	千兆以太网控制器
EMAC	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 以太网介绍

3.1 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 10 Mbps、100 Mbps、1000 Mbps 和 10 Gbps 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。

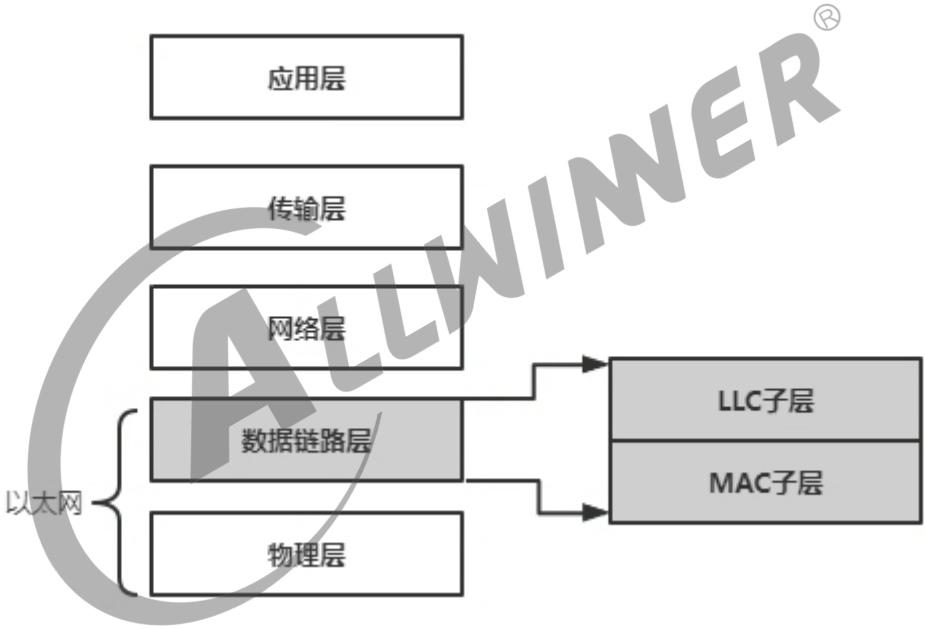


图 3-1: 以太网在 TCP/IP 协议族中的位置

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。

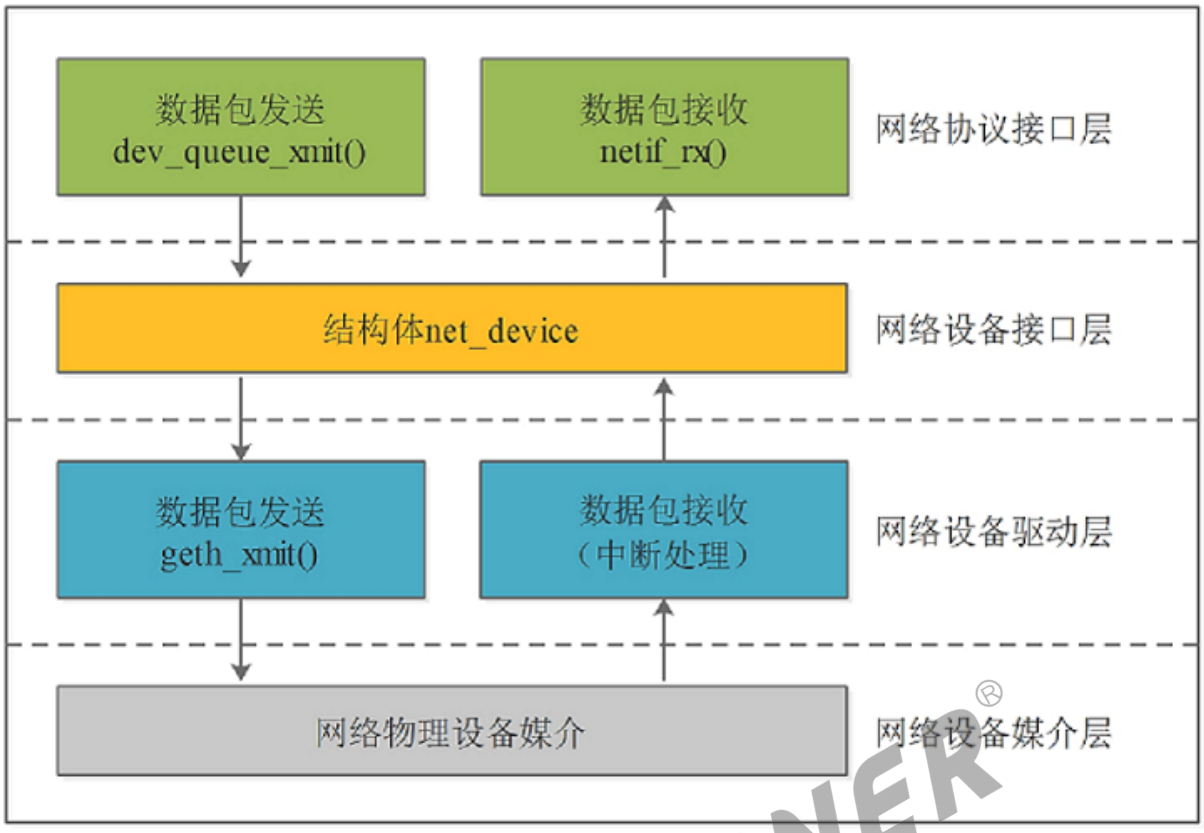


图 3-2: 网络设备框架

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 以太网配置说明

3.2.1 协议栈配置说明

对于 longan，可以直接在 longan 的根目录执行./build.sh menuconfig；

对于 Tina 的环境，可以在根目录执行 make kernel_menuconfig 进入 menuconfig 配置界面。

（1）配置网络协议栈

```
CONFIG_PACKET
CONFIG_UNIX
CONFIG_UNIX_DIAG
CONFIG_NET_KEY
CONFIG_INET
CONFIG_IP_MULTICAST
CONFIG_IP_ADVANCED_ROUTER
CONFIG_IP_FIB_TRIE_STATS
CONFIG_IP_MULTIPLE_TABLES
```

3.2.2 驱动配置说明

对于 longan，可以直接在 longan 的根目录执行./build.sh menuconfig；

对于 Tina 的环境，可以在根目录执行 make kernel_menuconfig 进入 menuconfig 配置界面。

3.2.2.1 GMAC 配置说明

勾选 GMAC 和 GMAC 对应的 MDIO 驱动

```
CONFIG_AW_GMAC
CONFIG_AW_GMAC_MDIO
```

3.2.2.2 EMAC 配置说明

勾选 EMAC 和 EMAC 对应的 MDIO 驱动

```
CONFIG_AW_EMAC
CONFIG_AW_EMAC_MDIO
```

3.2.2.3 GMAC2XX 配置说明

```
CONFIG_AW_STMMAC_ETH
```

3.2.3 设备树配置说明

3.2.3.1 GMAC 配置说明

soc 设备树配置

```
mdio0: mdio0@4500048 {
    compatible = "allwinner,sunxi-mdio";
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <0x0 0x04500048 0x0 0x8>;
    status = "disabled";
    gmac0_phy0: ethernet-phy@1 {
        /* RTL8211F (0x001cc916) */
        reg = <1>;
        max-speed = <1000>;  /* Max speed capability */
        reset-gpios = <&pio PH 19 GPIO_ACTIVE_LOW>;
        /* PHY datasheet rst time */
        reset-assert-us = <10000>;
        reset-deassert-us = <150000>;
    };
};

gmac0: gmac0@4500000 {
    compatible = "allwinner,sunxi-gmac";
    reg = <0x0 0x04500000 0x0 0x10000>,
          <0x0 0x03000030 0x0 0x4>;
    interrupts = <GIC_SPI 46 IRQ_TYPE_LEVEL_HIGH>;
    interrupt-names = "gmacirq";
    clocks = <&ccu CLK_GMAC0>, <&ccu CLK_GMAC0_25M>;
    clock-names = "gmac", "phy25m";
    resets = <&ccu RST_BUS_GMAC0>;
    phy-handle = <&gmac0_phy0>;
    status = "disabled";
};
```

（1）“compatible” 表征具体的设备, 用于驱动和设备的绑定；（**注：客户无需关注**）

（2）“reg” 设备使用的地址；（**注：客户无需关注**）

（3）“interrupts” 设备使用的中断；（**注：客户无需关注**）

（4）“clocks” 设备使用的时钟；（**注：客户无需关注**）

（5）“status” 是否使能该设备节点；（**注：客户需关注，使能接口需要修改 status = “okay”**）

（6）“phy-handle”phy 器件句柄；（**注：客户无需关注**）

（7）phy 子节点配置（**注：客户需关注，更换 PHY 器件需要更改此属性**）：“reg” 表征 phy 地址，“max-speed” 表征 phy 的最大速率，“reset-gpios” 表征 phy 硬件复位的引脚，“reset-assert-us” 硬件复位拉低时间，“reset-deassert-us” 硬件复位拉高时间。

phy 地址：根据 PHY 器件的 PHYAD[x] 实际高低电平配置

max-speed：千兆配置 1000、百兆配置 100

reset-gpios：根据板级实际使用的 gpio 配置<&pio Px xx GPIO_ACTIVE_LOW>

reset-assert-us：根据 phy 手册中介绍的上电时序中 phyrst 拉低时间配置，默认属性为通用配置，一般无需更改

reset-deassert-us：根据 phy 手册中介绍的上电时序中 phyrst 拉高时间配置，默认属性为通用配置，一般无需更改

注：更换 PHY 器件的其他问题请查看PHY 适配指导

板级设备树配置

板级设备树文件路径为SDK/device/config/chips/{IC}/configs/{BOARD}/board.dts。

配置内容如下：

```
&gmac0 {
    phy-mode = "rgmii";
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&gmac0_pins_default>;
    pinctrl-1 = <&gmac0_pins_sleep>;
    sunxi,phy-clk-type = <0>;
    tx-delay = <3>;
    rx-delay = <4>;

    gmac3v3-supply = <&reg_cldo3>;
    status = "okay";
};
```

（1）“phy-mode”GMAC 与 PHY 之间的物理接口，如 MII、RMII、RGMII 等；（**注：更换 PHY 器件需要更改此属性**）：百兆一般为 RMII，千兆为 RGMII

（2）“pinctrl-0” 设备使用状态下的 GPIO 配置，“pinctrl-1” 设备休眠状态下的 GPIO 配置；（**注：客户无需关注**）

（3）“sunxi,phy-clk-type” 配置 phy 使用的时钟，0 表示使用 soc 内置的 25 M 时钟；（**注：客户需关注**）：为外挂 phy 器件的时钟，由方案决定，若实际使用外挂 25M 晶振给 phy 供时钟，则该属性为 1，若使用 soc 的 25M 时钟给 phy 供时钟，则该属性为 0；关于时钟配置方式的硬件原理图，请参考时钟检查

（3）“tx-delay”tx 时钟延迟，tx-delay 取值 0-7，一档约 536 ps（皮秒）；（**注：客户需关注，具体配置方法请查看delay 参数配置**）

（4）“rx-delay”rx 时钟延迟，rx-delay 取值 0-31，一档约 186 ps（皮秒）；（**注：客户需关注，具体配置方法请查看delay 参数配置**）

（5）“gmac3v3-supply”gmac 电源脚；（**注：客户需关注**）根据原理图实际使用的电源进行配置，若使用的是 VCC-IO，则不需要配置；

（6）“status” 是否使能该设备节点。

3.2.3.2 EMAC 配置说明

```
mdio1: mdio1@1c0b080 {
   compatible = "allwinner,sun4i-mdio";
   #address-cells = <1>;
   #size-cells = <0>;
   reg = <0x0 0x01c0b080 0x0 0x14>;
   status = "okay";
   phy1: ethernet-phy@1 {
      reg = <1>;
   };
};

emac0: emac0@1c0b000 {
   compatible = "allwinner,sunxi-emac";
   reg = <0x0 0x01c0b000 0x0 0x0c000>;
   interrupts     = <GIC_SPI 55 IRQ_TYPE_LEVEL_HIGH>;
   interrupt-names = "emacirq";
   clocks   = <&ccu CLK_BUS_EMAC>;
   clock-names = "emac";
   resets = <&ccu RST_BUS_EMAC>;
   pinctrl-names = "default";
   pinctrl-0    = <&emac_pins_a>;
   phy-rst = <&pio PH 27 1 1 1 0>;
   status = "okay";
   phy-handle = <&phy1>;
   status = "disabled";
};
```

（1）”compatible” 表征具体的设备, 用于驱动和设备的绑定；（**注：客户无需关注**）

（2）”reg” 设备使用的地址；（**注：客户无需关注**）

（3）”interrupts” 设备使用的中断；（**注：客户无需关注**）

（4）”clocks” 设备使用的时钟；（**注：客户无需关注**）

（5）”pinctrl-0” 设备 active 状态下的 GPIO 配置；（**注：客户无需关注**）

（6）”phy-rst” （**注：客户需关注，更换 PHY 器件需要更改此属性**）PHY 复位脚，根据板级实际使用的 gpio 配置<&pio Px xx 1 1 1 0>

（7）“phy-handle”phy 器件句柄；（**注：客户无需关注**）

（8）phy 子节点配置（**注：客户需关注，更换 PHY 器件需要更改此属性**）：“reg” 表征 phy 地址。根据 PHY 器件的 PHYAD[x] 实际高低电平配置

注：更换 PHY 器件的其他问题请查看PHY 适配指导

3.2.3.3 GMAC2XX 配置说明

soc 设备树配置

```
gmac_stmmac_axi_setup: stmmac-axi-config {
  snps,lpi_en;
  snps,wr_osr_lmt = <0xf>;
  snps,rd_osr_lmt = <0xf>;
  snps,blen = <256 128 64 32 16 8 4>;
};

gmac_mtl_rx_setup: rx-queues-config {
  snps,rx-queues-to-use = <1>;
  queue0 {};
};

gmac_mtl_tx_setup: tx_queues-config {
  snps,tx-queues-to-use = <1>;
  queue0 {};
};

gmac0: ethernet@4500000 {
  compatible = "allwinner,sunxi-gmac-220", "snps,dwmac-5.10a";
  reg = <0x0 0x04500000 0x0 0x8000>,
      <0x0 0x04508000 0x0 0x8000>;
  interrupts = <GIC_SPI 46 IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI 92 IRQ_TYPE_LEVEL_HIGH>,
          <GIC_SPI 93 IRQ_TYPE_LEVEL_HIGH>, <GIC_SPI 94 IRQ_TYPE_LEVEL_HIGH>,
          <GIC_SPI 224 IRQ_TYPE_LEVEL_HIGH>;
  interrupt-names = "macirq", "eth_lpi", "tx0_irq", "rx0_irq", "mac_eccirq";
  clocks = <&ccu CLK_GMAC0>, <&ccu CLK_MBUS_GMAC0_GATE>, <&ccu CLK_GMAC0_PHY>,
      <&ccu CLK_GMAC0_PTP>, <&ccu CLK_GMAC_NSI>;
  clock-names = "stmmaceth", "pclk", "phy", "ptp_ref", "nsi";
  assigned-clocks = <&ccu CLK_GMAC_NSI>, <&ccu CLK_GMAC0_PHY>;
  assigned-clock-parents = <&ccu CLK_PLL_PERI0_480M>;
  assigned-clock-rates = <480000000>, <25000000>;
  resets = <&ccu RST_BUS_GMAC0_AXI>, <&ccu RST_BUS_GMAC0>;
  reset-names = "stmmaceth", "ahb";
  phy-handle = <&gmac0_phy0>;
  status = "disabled";

  aw,rgmii-clk-ext;
  snps,fixed-burst;
  snps,en-tx-lpi-clockgating;
  snps,axi-config = <&gmac_stmmac_axi_setup>;
  snps,mtl-rx-config = <&gmac_mtl_rx_setup>;
  snps,mtl-tx-config = <&gmac_mtl_tx_setup>;

  mdio0: mdio0@0 {
    compatible = "snps,dwmac-mdio";
    #address-cells = <1>;
    #size-cells = <0>;
    gmac0_phy0: ethernet-phy@1 {
      compatible = "ethernet-phy-ieee802.3-c22";
      reg = <0x1>;
      max-speed = <1000>;  /* Max speed capability */
      reset-gpios = <&pio PJ 27 GPIO_ACTIVE_LOW>;
      /* PHY datasheet rst time */
      reset-assert-us = <10000>;
      reset-deassert-us = <150000>;
    };
  };
};
```

（1） “compatible” 表征具体的设备, 用于驱动和设备的绑定；（**注：客户无需关注**）

（2）“reg” 设备使用的地址；（**注：客户无需关注**）

（3）“interrupts” 设备使用的中断；（**注：客户无需关注**）

（4）“clocks” 设备使用的时钟；（**注：客户无需关注**）

（5）“status” 是否使能该设备节点；（**注：客户需关注，使能接口需要修改 status = “okay”**）

（6）“phy-handle”phy 器件句柄；（**注：客户无需关注**）

（7）phy 子节点配置（**注：客户需关注，更换 PHY 器件需要更改此属性**）：“reg” 表征 phy 地址，“max-speed” 表征 phy 的最大速率，“reset-gpios” 表征 phy 硬件复位的引脚，“reset-assert-us” 硬件复位拉低时间，“reset-deassert-us” 硬件复位拉高时间；

phy 地址：根据 PHY 器件的 PHYAD[x] 实际高低电平配置

max-speed：千兆配置 1000、百兆配置 100

reset-gpios：根据板级实际使用的 gpio 配置<&pio Px xx GPIO_ACTIVE_LOW>

reset-assert-us：根据 phy 手册中介绍的上电时序中 phyrst 拉低时间配置，默认属性为通用配置，一般无需更改

reset-deassert-us：根据 phy 手册中介绍的上电时序中 phyrst 拉高时间配置，默认属性为通用配置，一般无需更改

注：更换 PHY 器件的其他问题请查看PHY 适配指导

（8）“snps,axi-config”AXI 总线模式参数；（**注：客户无需关注**）

（9）“snps,mtl-rx-config”rx 队列参数；（**注：客户无需关注**）

（10）“snps,mtl-tx-config”tx 队列参数。（**注：客户无需关注**）

板级设备树配置

```
&gmac0_phy0 {
    // compatible = "ethernet-phy-id001c.c916";
    // reset-gpios = <&pio PI 5 GPIO_ACTIVE_LOW>;
};

&gmac0 {
    phy-mode = "rgmii";
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&gmac0_pins_default>;
    pinctrl-1 = <&gmac0_pins_sleep>;
    aw,soc-phy25m;
    tx-delay = <3>;
    rx-delay = <4>;
    dwmac3v3-supply = <&reg_cldo3>;
    phy3v3-supply = <&reg_cldo4>;
    status = "okay";
};
```

（1）“phy-mode”GMAC 与 PHY 之间的物理接口，如 MII、RMII、RGMII 等；（**注：客户需关注，更换 PHY 器件需要更改此属性**）：百兆一般为 RMII/MII，千兆为 RGMII

（2）“pinctrl-0” 设备使用状态下的 GPIO 配置，“pinctrl-1” 设备休眠状态下的 GPIO 配置；（**注：客户无需关注**）

（3）“aw,soc-phy25m” 属性表示使用 SOC 供的 25M 时钟给 PHY；（**注：客户需关注**）：为外挂 phy 器件的时钟，由方案决定，若实际使用外挂 25M 晶振给 phy 供时钟，则不配置该属性，若使用 soc 的 25M 时钟给 phy 供时钟，则配置该属性；关于时钟配置方式的硬件原理图，请参考 6.3 时钟检查章节

（3）“tx-delay”tx 时钟延迟，tx-delay 取值 0-7，一档约 536 ps（皮秒）；（**注：客户需关注，具体配置方法请查delay 参数配置**）

（4）“rx-delay”rx 时钟延迟，rx-delay 取值 0-31，一档约 186 ps（皮秒）；（**注：客户需关注，具体配置方法请查看delay 参数配置**）

（5）“dwmac3v3-supply”gmac 控制器电源脚；（**注：客户需关注**）根据原理图实际使用的电源进行配置，若使用的是 VCC-IO，则不需要配置；

（6）“phy3v3-supply” 外挂 phy 电源脚；（**注：客户需关注**）根据原理图实际使用的电源进行配置，若使用的是 VCC-IO，则不需要配置；

（7）“status” 是否使能该设备节点。

（8）“compatible” 外挂 phy 设备树匹配名字；（**注：客户需关注**）若需固定 phyid 配置，则在 dts 中写死相关信息；

（9）“reset-gpios” 外挂 phy 复位 gpio 引脚（**注：客户需关注**）若修改了 phyrst 引脚，则在板级 dts 中覆盖 dtsi 的默认配置；

3.2.4 AW 定制 PHY

AW 部分 SOC 集成了 AC200 和 AC300，而 AC200 和 AC300 内部又集成了 EPHY。

3.2.4.1 AC200

ARM 通过 TWI 与 AC200 进行通讯, 把 EPHY 初始化，然后 MAC 通过 MDIO 总线是访问 EPHY，PWM 模块提供一个内部 25M 时钟给 EPHY。

AC200 整体框图如下。

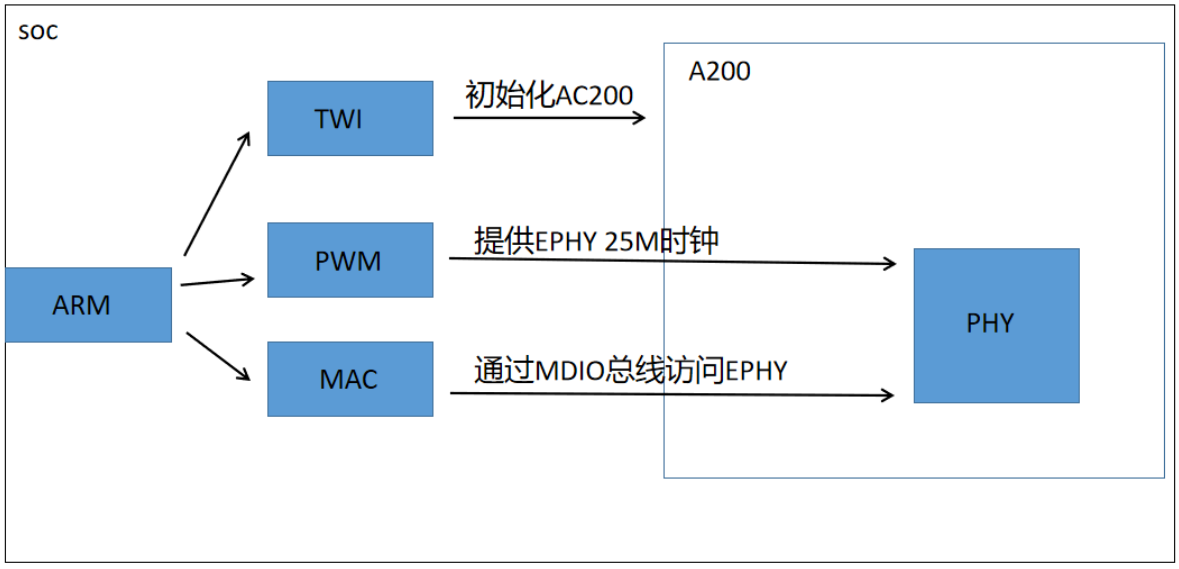


图 3-3: AC200 框图

3.2.4.2 AC300

ARM 通过 MDIO 总线与 AC300 进行通讯，把 EPHY 初始化，然后 MAC 通过 MDIO 总线是访问 EPHY，PWM 模块提供一个内部 25M 时钟给 EPHY。

AC300 整体框图如下。

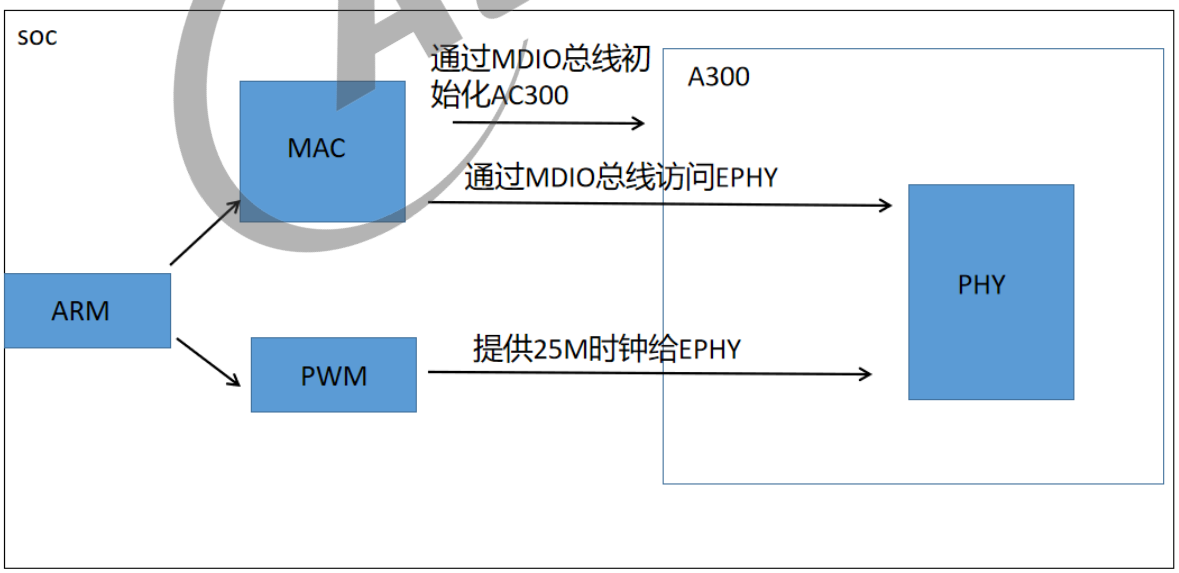


图 3-4: AC300 框图

说明

有些 SOC 内部 AC200 封装了 EPHY，通过 TWI 与 AC200 进行通讯，把 EPHY 初始化，然后 MAC 通过 MDIO 总线是访问 EPHY。有些 SOC 内部 AC300 封装了 EPHY，通过 MAC 控制器 MIDO 总线与 AC300 进行通信，把 EPHY 初始化，然后 MAC 同样利用 MDIO 总线去访问 EPHY。V821 中，内部 PHY 时钟不再由 PWM 给出，而是芯片内部 PLL 分频提供。

3.3 源码结构

GMAC 驱动源码位于bsp/drivers/gmac目录下：

```
├── Kconfig
├── Makefile
├── sunxi-gmac.c   //GMAC驱动代码
└── sunxi-mdio.c   //GMAC对应MDIO驱动代码
```

EMAC 驱动源码位于bsp/drivers/emac目录下：

```
├── Kconfig
├── Makefile
├── sun4i-emac.c   //EMAC驱动代码
└── sun4i-mdio.c   //EMAC对应MDIO驱动代码
```

GMAC-2XX 驱动源码位于bsp/drivers/stmmac目录下：

```
├── Kconfig
├── Makefile
├── dwmac-sunxi.c     /* AW GMAC平台配置 */
├── dwmac-sunxi.h
├── dwmac-sunxi-sysfs.c /* sysfs调试节点 */
└── dwmac-sunxi-sysfs.h
```

4 FAQ

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
```

打开节点，正常 log 如下：

```
GMAC log:
sunxi-gmac 4500000.gmac0 eth0: eth0: Type(9) PHY ID 001cc916 at 1 IRQ poll (4500048.mdio0-mii:01)

GMAC-2XX log:
dwmac-sunxi 4510000.ethernet eth1: PHY [stmmac-0:01] driver [Generic PHY] (irq=POLL)
dwmac-sunxi 4510000.ethernet eth1: Register MEM_TYPE_PAGE_POOL RxQ-0
dwmac4: Master AXI performs fixed burst length
dwmac-sunxi 4510000.ethernet eth1: No Safety Features support found
dwmac-sunxi 4510000.ethernet eth1: IEEE 1588-2008 Advanced Timestamp supported
dwmac-sunxi 4510000.ethernet eth1: configuring for phy/rgmii link mode
```

插上网线以后，正常 log 如下：

```
sunxi-gmac 4500000.gmac0 eth0: Link is Up - 1000Mbps/Full - flow control off
```

关闭节点，正常 log 如下：

```
sunxi-gmac 4500000.gmac0 eth0: Link is Down
```

（3）配置网络设备；

```
配置静态IP地址：ifconfig eth0 192.168.1.110
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 1000 duplex full autoneg on（设置1000 Mbps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对端ip地址，本机ip和对端ip的前三个网段需相同才能ping通。
ping 192.168.1.100

TCP吞吐测试：
```

```
Server端：iperf -s -i 1
Client端：iperf -c 192.168.1.100 -i 1 -t 60

UDP吞吐测试：
Server端：iperf -s -u -i 1
Client端：iperf -c 192.168.1.100 -u -b 1000M -i 1 -t 60
```

4.2 以太网常见问题排查流程

4.2.1 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，检查驱动 probe 是否成功；

（2）如果无驱动相关打印，请参考以太网配置说明确认以太网基本配置是否打开；

（3）如果驱动 probe 失败，请 log 定位具体原因，常见原因是 GPIO 冲突导致。

4.2.2 网络不通或网络丢包

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，硬件问题请参考以太网 PHY 硬件排查方法

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，如果和电脑对测，确保防火墙不会影响测试；

（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。

4.2.3 网卡 up 失败

网卡无法 up 的现象与PHY 适配指导出错的日志是一致的，一般是找不到 PHY 或者 DMA 复位失败，表现：

```
# GMAC
[ 25.506200] sunxi-gmac gmac0 eth0: Error：Could not connect to phy

# GMAC-2XX
[ 25.116200] dwmac-sunxi 45000000.ethernet eth0: stmmac_open: Cannot attach to PHY
```

```
# GMAC
[ 11.315214] sunxi-gmac gmac1 eth0: Error: Mac reset failed, please check phy and mac clk

# GMAC-2XX
[ 22.601790] dwmac-sunxi 45000000.ethernet eth0: stmmac_hw_setup: DMA engine initialization failed
[ 22.611644] dwmac-sunxi 45000000.ethernet eth0: stmmac_open: Hw setup failed
```

请按照PHY 适配指导、以太网 PHY 硬件排查方法排查顺序进行排查（先排查软件配置，再排查硬件）

4.3 jumbo 帧测试方法

前提：

准备两台均支持 jumbo 帧的板卡直连对测。

步骤：

以千兆网卡为例；

（1）板卡 A：ifconfig eth0 mtu 8100，板卡 B：ifconfig eth0 mtu 8100（设置 eth0 的 mtu 大小为 8100）；

（2）板卡 A：ifconfig eth0 192.168.200.10，板卡 B：ifconfig eth0 192.168.200.9；

（3）板卡 A：ping 192.168.200.9 -s 8000（传输 8000 长度的数据）。

注：

（1）百兆网卡不存在带宽不足情况，一般不需要使用 jumbo 帧；

（2）百兆网卡支持的 jumbo 帧范围为 1500 至 4000，千兆网卡支持的 jumbo 帧范围为 1500 至 8100；

（3）测试时两端 mtu 无需设置成相同大小；

（4）ping 通即表明测试通过；

（5）ping 传输 8000 长度的原因：mtu 为整个以太网帧的长度，ping 工具指定的长度 8000 为纯数据长度，不包括 ICMP 及以太网报头的长度。

4.4 以太网回环测试方法

问题背景：

重新设计 PCB 板，或者更换 PHY 器件后，导致以太网能正常 up，但是 ping 不通。

复现步骤：

ping 对端设备，例如ping 192.168.200.9。

问题分析：

- PHY 被设置成回环模式后，从 MAC 经过 RMII/RGMII 发送过来的数据不会再被发送到 MDI 上，而是在 PHY 内部 PCS 层进行回环，被回环到 RGMII/RMII 的接收通道，如下图所示；

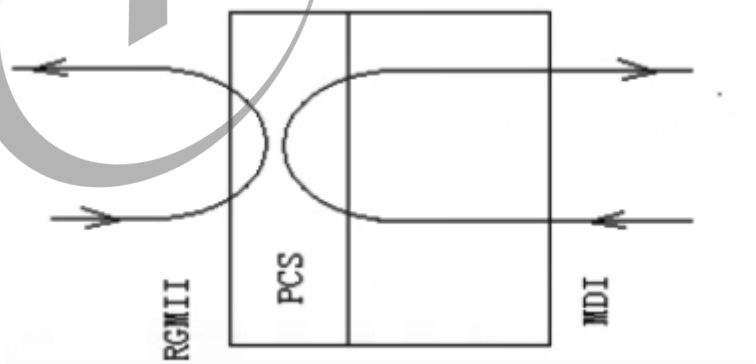


图 4-1: PHY 的 PCS 层

- 当设置了 PHY 的回环模式后，往外发包，如果小机端的 rx 能正常接收到数据，说明需要检查 PHY 到对端的硬件电路有无异常，反之需要检查 RMII/RGMII 接口的硬件电路有无异常；

测试办法：

- 通过 GMAC 驱动提供的节点，设置 PHY 的回环模式；

```
cd sys/devices/platform/soc/gmac/ #注：此处为大致路径，不同平台路径有轻微差异
echo 2 > loopback_test; cat loopback_test
```

- 往外 ping 包，观察 tx 和 rx 的增长数量是否一致，如下图所示；

```
/sys/devices/platform/soc/gmac0 # ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.102  netmask 255.255.255.0  broadcast 192.168.100.255
        ether ea:9d:32:f6:32:7f  txqueuelen 1000  (Ethernet)
        RX packets 93  bytes 8211 (8.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 882 (882.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 55

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1  (Local Loopback)
        RX packets 23  bytes 2576 (2.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 2576 (2.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

/sys/devices/platform/soc/gmac0 # echo 2 > loopback_test
/sys/devices/platform/soc/gmac0 # cd
/ # ping 192.168.100.101
PING 192.168.100.101 (192.168.100.101): 56 data bytes
^C
--- 192.168.100.101 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
/ # ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.102  netmask 255.255.255.0  broadcast 192.168.100.255
        ether ea:9d:32:f6:32:7f  txqueuelen 1000  (Ethernet)
        RX packets 99  bytes 8571 (8.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 1134 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 55

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1  (Local Loopback)
        RX packets 27  bytes 3024 (2.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 3024 (2.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

图 4-2: 回环测试

- 若 tx 和 rx 的增长数量一致，则需要重点排查 PHY 到对端的硬件电路有无异常；若不一致（例如 rx 收不到包或数量少于 tx），则需要重点排查 RMII/RGMII 接口的硬件电路有无异常。

4.5 内核打印 Link is Up 的条件

- 保持网线插入网口，并连接以太网功能正常的对端设备；
- ifconfig up 网卡后，软件会起一个 1s 的轮询机制；

- 轮询查询 phy 的 02 寄存器的 Link status 标志位，若为 1，则根据 phy 的速率/双工更改 mac 的速率/双工，并打印 Link is up - xxxMbps/Full ，若为 0，则打印 Link is Down；
- 只有 Link status 发生改变时才会打印，比如 0->1 1->0；
- phy 的 Link status 标志位是只读的，根据两端 phy 的自协商情况完成置 1/清 0，若遇到频繁 Up/Down 的情况，请参考以太网 PHY 硬件排查方法。

5 PHY 适配指导

问题背景：

重新设计 PCB 板，或者更换 PHY 器件后，导致以太网无法正常工作，硬件问题请参考以太网 PHY 硬件排查方法。

复现步骤：

up 网卡

```
ifconfig eth0 up
```

具体表现：

（1）找不到 PHY；

```
# GMAC
[ 25.506200] sunxi-gmac gmac0 eth0: Error：Could not connect to phy

# GMAC-2XX
[ 25.116200] dwmac-sunxi 45000000.ethernet eth0: stmmac_open: Cannot attach to PHY
```

（2）识别到 PHY 但 MAC 复位失败；

```
# GMAC
[ 11.315214] sunxi-gmac gmac1 eth0: Error: Mac reset failed, please check phy and mac clk

# GMAC-2XX
[ 22.601790] dwmac-sunxi 45000000.ethernet eth0: stmmac_hw_setup: DMA engine initialization failed
[ 22.611644] dwmac-sunxi 45000000.ethernet eth0: stmmac_open: Hw setup failed
```

问题分析：

找不到 PHY 的原因有如下几点：

- PHY 没有被正确初始化：没有给 PHY 供 25M 时钟、PINMUX 选择不对/PIN BANK 电压没开（概率较低）、UP 网卡之前没有给 PHY 做复位操作、PHY 供电异常，请参考以太网 PHY 硬件排查方法；
- 平台更换 PHY 之后，PHY 地址出现变更。

识别到 PHY 但 MAC 复位失败的原因有如下几点：

- 时钟引脚 I/O 方向不正确：RMII 接口：MAC 的 TXCK 引脚是输入，由 PHY 输出该信号；RGMII 接口：MAC 的 RGMII_CLKIN 是输入，由 PHY 输出该信号；

- 时钟引脚 I/O 方向正确，供给 PHY 的 25M 时钟偏差较大（合理范围一般是 25Mhz -+ 0.02Mhz，具体请参考 PHY 的 datasheet），导致 PHY 输出给 MAC 的时钟频率不稳定；
- 时钟引脚 I/O 方向正确，供给 PHY 的 25M 时钟在合理范围内，但回给 MAC 的时钟信号频率不稳定：RMII: 5Mhz（10M）/50Mhz（100M），RGMII：25Mhz(100M)/125Mhz（1000M）。

解决方法：

1、找不到 PHY 的解决方法：

- 示波器检查 PHY25M 时钟是否正常，万用表检查 PHY 供电（如 AVDD33、DVDDIO、AD-VDD10OUT、DVDD10OUT 等）是否正常；
- 是否在 dts 中配置 PHY 的复位引脚：请结合原理图并参考以太网配置说明，配置复位引脚到设备树的 phy 子节点的reset-gpios属性中；
- PHY 地址出现变更：请结合原理图上 PHYADD 相关引脚的电平，并参考以太网配置说明，配置 PHY 地址到设备树的 phy 子节点的reg属性中，若无法确定 PHY 地址，如下两种方法可以获取到 PHY 地址；
- 请使用 7.3 章节的 mii_reg 工具手动读取下对应地址的 PHY 寄存器，若能正确读到寄存器（比如 phyid 寄存器），则表示 phy 的地址为当前地址；
- 打开 mdio 驱动（GMAC 对应sunxi-mdio.c）开头的 DEBUG 宏，启动日志会打印 PHY 地址，log 如下，PHYADDR 即为 PHY 地址，将这个值写到设备树 phy 子节点的reg属性中属性中。

```
[   2.814160] sunxi-mdio 5020048.mdio0: PHYADDR = 1, PHYID = 0x1cc916
```

- 检查设备树中 phy 节点的 phyid 是否做修改：

```
gmac1_phy0: ethernet-phy@1 {
        compatible = "ethernet-phy-ieee802.3-c22";
        ...
}
```

修改为实际的 phyid：

```
gmac1_phy0: ethernet-phy@1 {
        compatible = "ethernet-phy-id001c.c916";
        ...
}
```

修改原因：内核读取流程不同导致使用compatible = "ethernet-phy-ieee802.3-c22";属性要求 phyrst 必须接上拉电阻，而compatible = "ethernet-phy-id001c.c916";则不需要

2、识别到 PHY 但 MAC 复位失败解决办法：

硬件问题，请与一名硬件同事协同排查；

- 时钟引脚 I/O 方向不正确：MAC 这边对时钟引脚的要求均是输入，请更改硬件让相应 RMII 的 TXCK，RGMII 的 CLKIN 引脚变成 PHY 输出，MAC 输入；

- 时钟引脚 I/O 方向正确，PHY25Mhz 不稳定或者回给 MAC 的时钟信号偏差较大：请检查 PCB 走线/电容电阻等器件是否符合 PHY 手册里的 PCB 设计要求；
- 复位信号及上电时序。

注：

一般情况下 PHY 不需要 PHY 驱动，走内核的通用驱动，但某些厂商的 PHY 需要特殊的 PHY 驱动支持，如果按照通用 PHY 的调试方法仍无法调通，请联系 PHY 厂介入。

6 以太网 PHY 硬件排查方法

本章介绍硬件通用排查方法，具体请以实际使用的 PHY 器件规格书为准。

6.1 检查供电电压

- 电压检查：用万用表测试电压，需满足规格书要求。

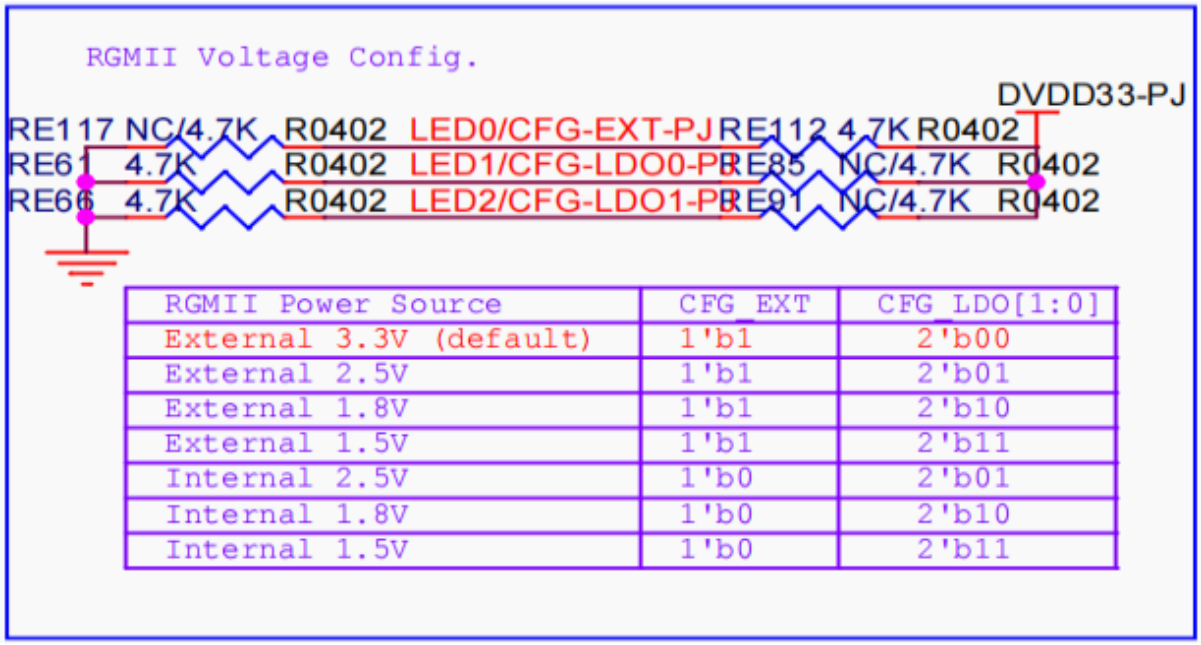
对于 PHY，需要检查 PHY 的电源电压和 IO 供电，以 RTL8211，为例，VDD 与 IO 供电要求如下：

Symbol	Description	Min	Max	Unit
VDD33, AVDD33	Supply Voltage 3.3V.	-0.3	3.63	V
AVDD09, DVDD09	Supply Voltage 0.9V.	-0.3	1.05	V
2.5V RGMII	Supply Voltage 2.5V.	-0.2	2.75	V
1.8V RGMII	Supply Voltage 1.8V.	-0.2	1.98	V

图 6-1: 电压范围

- PHY IO 电压与 SOC IO 电压要适配

如 RTL8211，IO 电压可以配置为 3.3V、2.5V、1.8V、1.5V 等，需要通过外围电路配置，若配置错误，可能引起工作异常；



RGMII Power Source	CFG_EXT	CFG_LDO[1:0]
External 3.3V (default)	1'b1	2'b00
External 2.5V	1'b1	2'b01
External 1.8V	1'b1	2'b10
External 1.5V	1'b1	2'b11
Internal 2.5V	1'b0	2'b01
Internal 1.8V	1'b0	2'b10
Internal 1.5V	1'b0	2'b11

图 6-2: 电压外围电路配置

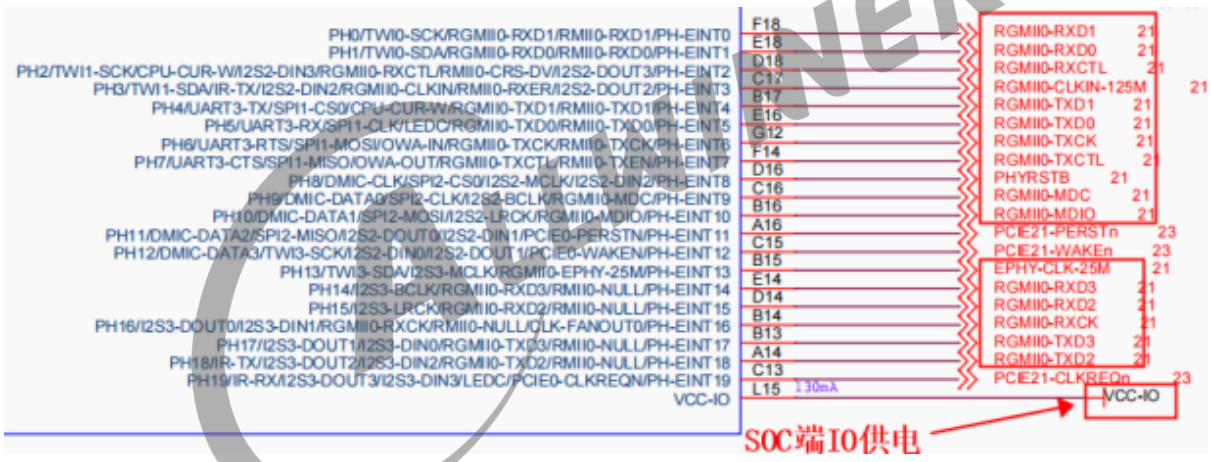


图 6-3: SOC 端 IO 供电

6.2 检查时钟源/晶振

要求满足 PHY 时钟频率的范围的要求。

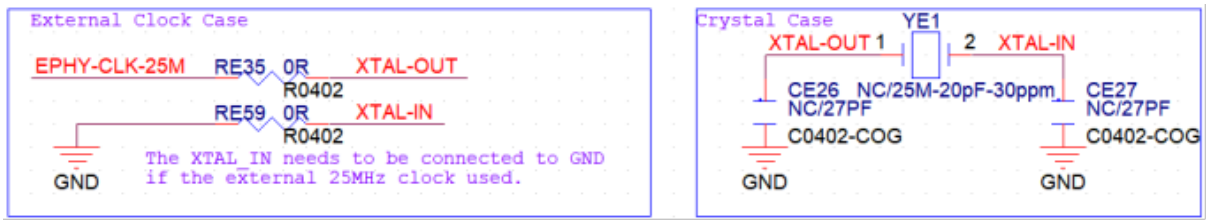


图 6-4: 时钟选择

Tips：PHY 的基准时钟可以选择晶振输入或 SOC 25M 输入；若使用 SOC 25M 时钟输入，XTAL-IN 引脚需要接 GND，且软件上要在设备树中配置相应参数

```
#GMAC
sunxi,phy-clk-type = <0>;
#GMAC2XX
aw,soc-phy-clk-en; # 原 aw,soc-phy25m
```

如 RTL8211 为例，检查 SOC 25M 时钟输入

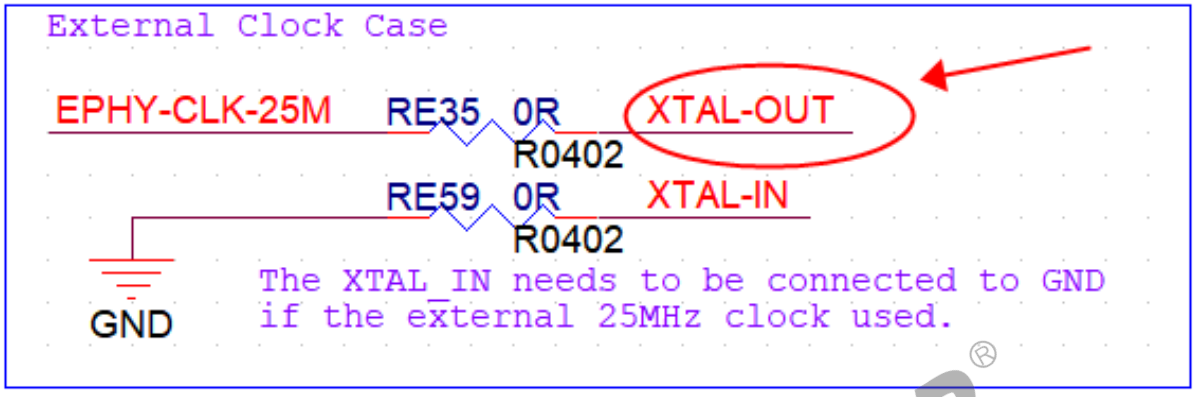


图 6-5: 25M 时钟输入

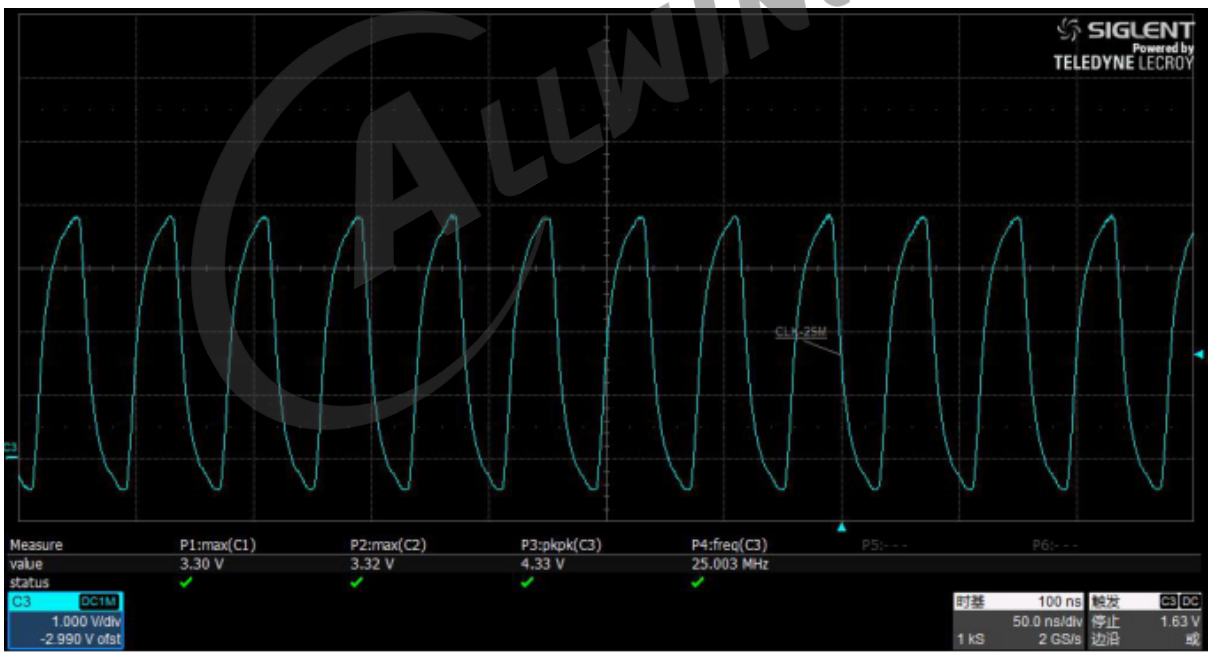


图 6-6: 时钟波形

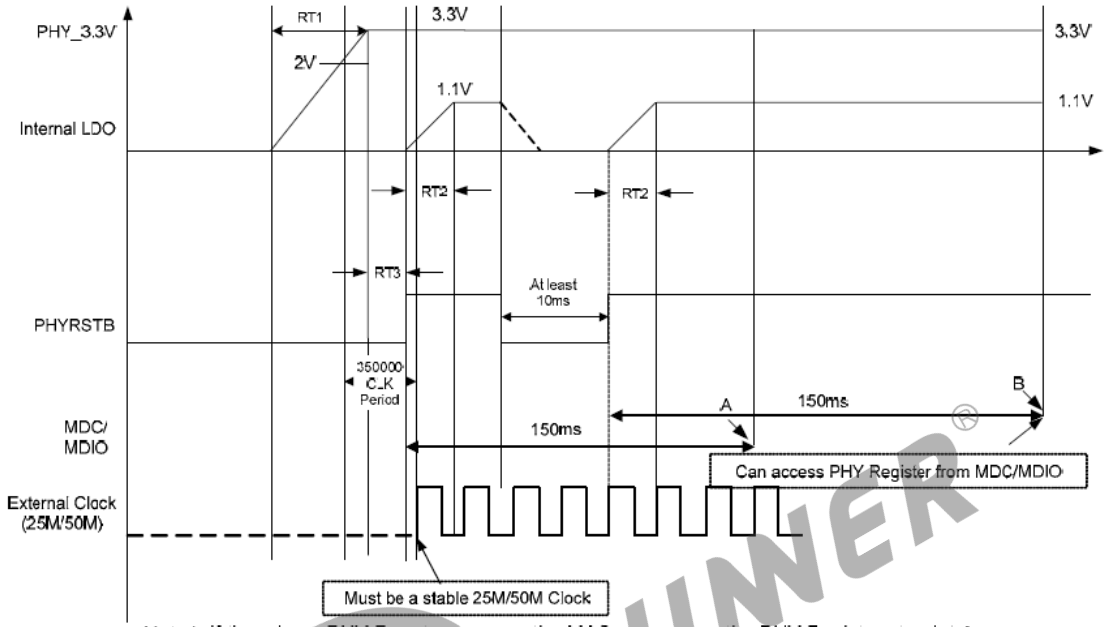
注意：示波器测试会受探头等效电容的影响，导致测试频率略有偏差；在要求高的情况下，确认晶振频率建议使用频谱仪器测试。

每款 PHY 对晶振的精确度要求不一致，PHY 的 datasheet 上会有相关说明

6.3 检查上电时序

要求：满足规格书的时序要求；

如 RTL8211，样机 PHY IO 使用外部供电，应满足规格书（如下图）的时序要求；



Note1: If there is no PHY Reset sequence, the MAC can access the PHY Register at point A.
Note2: If there is a PHY Reset sequence, the MAC can access the PHY Register at point B.
Note3: The Internal LDO (Linear Regulator) output voltage is 1.1V.

Figure 20. Power On and PHY Reset Sequence

Table 48. Power On and PHY Reset Sequence

Symbol	Description	Minimum	Maximum
Rt1	3.3V Rise Time@ Power On Sequence	100μs	-
Rt2	1.05V Rise Time@ Power On and PHY Reset Sequence	100μs	-
Rt3	PHYRSTB De-Assert after PHY_3.3V Stable	80μs	-

图 6-7: 上电时序规格书

上电时序检查：根据规格书，确定 PHY 的上电时序是否正常；

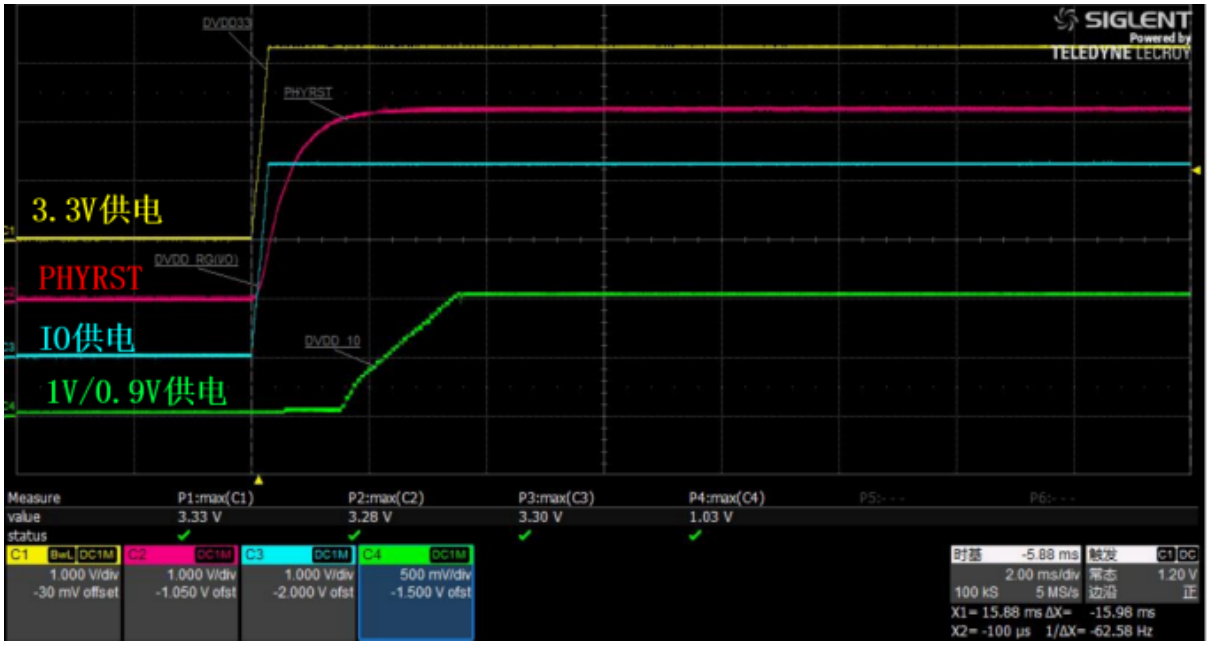


图 6-8: 上电时序

如上图，实测波形电源供电（黄色波形）和 IO 供电（蓝色波形）同时上电，满足 Rt6 0~5ms 的时序要求；

注意：不同 PHY 的要求不同，具体以 PHY 规格书要求为准。

6.4 检查 PHYRST

要求：满足规格书 Reset 时序

如 RTL8211，要求 PHY 启动时，PHYRSTB 下拉 10ms 以上；。

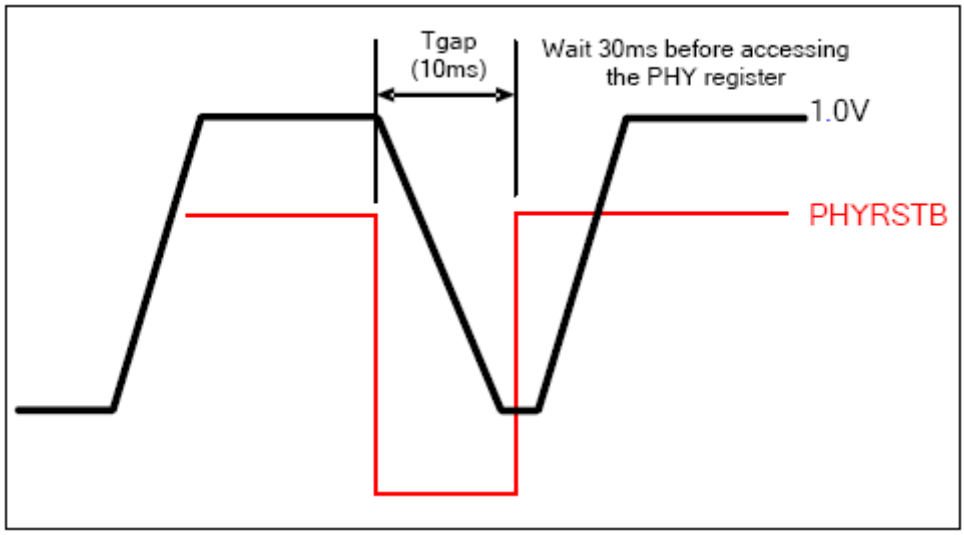


图 6-9: 复位时序

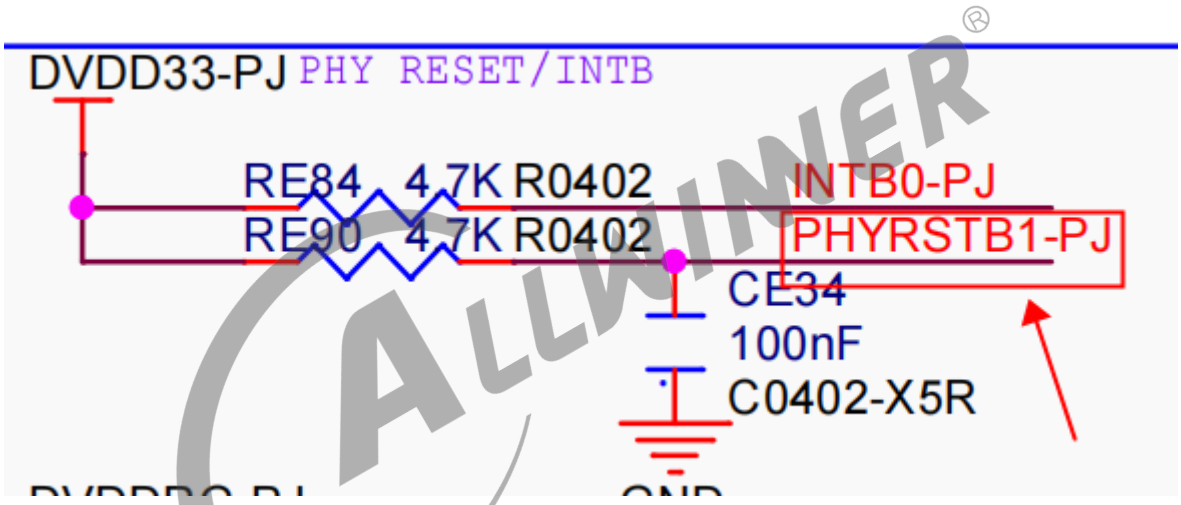


图 6-10: phyrst 原理图

示波器测量实际波形如下，实测 PHYRSTB 低电平为 20.1ms，满足时序要求：

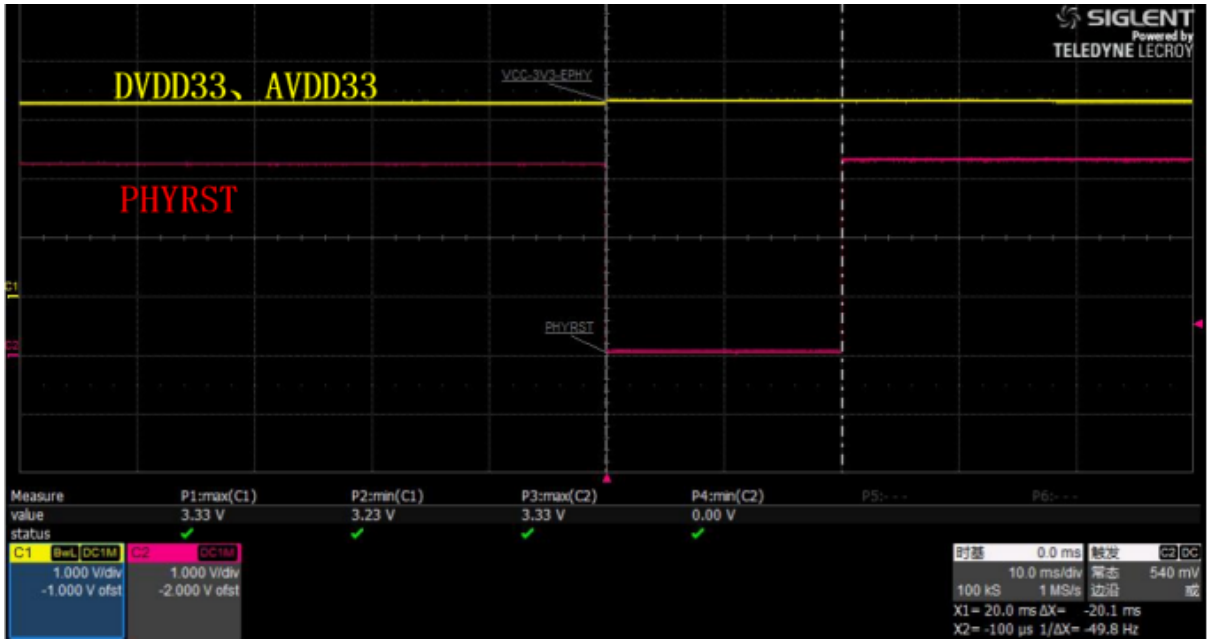


图 6-11: phyrst 波形图

6.5 检查 PHY 地址配置

确认硬件配置的 PHY 地址是否与软件设置的地址相匹配；

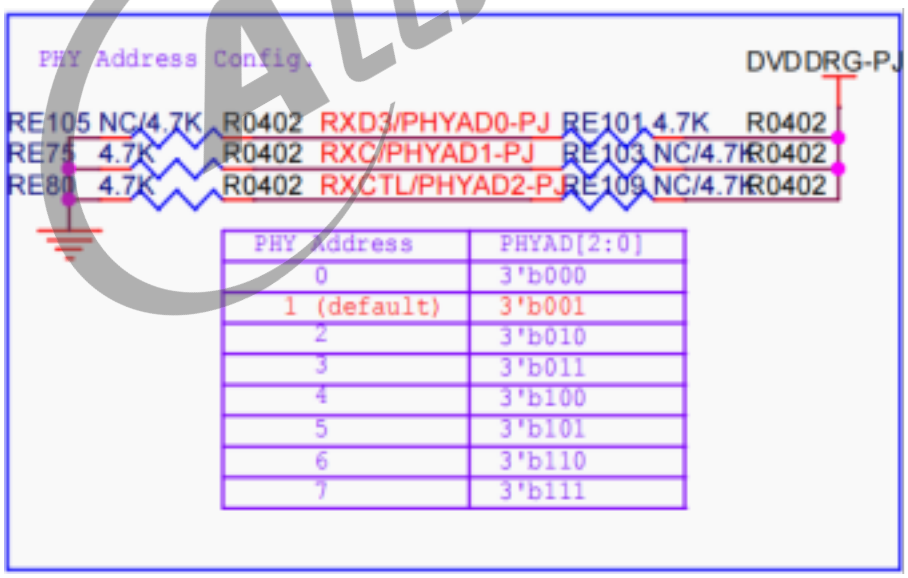


图 6-12: PHY 地址

6.6 检查 MDIO 总线

要求：MDC 频率要满足规格要求，MDC、MDIO 信号正常

异常案例：

异常板卡使用百兆 PHY RTL8201，启动网卡后，不断 link down，link up；

```
# ifconfig eth0 up
 146.914872] dwmac-sunxi 4510000.ethernet eth0: PHY [stmmac-0:01] driver [Generic PHY] (irq=POLL)
 146.925215] dwmac-sunxi 4510000.ethernet eth0: Register MEM_TYPE_PAGE_POOL RxQ-0
 146.946786] dwmac4: Master AXI performs fixed burst length
 146.952957] dwmac-sunxi 4510000.ethernet eth0: No Safety Features support found
 146.971175] dwmac-sunxi 4510000.ethernet eth0: IEEE 1588-2008 Advanced Timestamp supported
 146.980450] dwmac-sunxi 4510000.ethernet eth0: configuring for phy/rmii link mode
 146.989426] dwmac-sunxi 4510000.ethernet eth0: Link is Up - 10Mbps/Half - flow control off
 146.998718] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
#
# [  156.186963] dwmac-sunxi 4510000.ethernet eth0: Link is Down

# [  157.211000] dwmac-sunxi 4510000.ethernet eth0: Link is Up - 10Mbps/Half - flow control off
 165.402993] dwmac-sunxi 4510000.ethernet eth0: Link is Down
 166.427010] dwmac-sunxi 4510000.ethernet eth0: Link is Up - 10Mbps/Half - flow control off
 172.571001] dwmac-sunxi 4510000.ethernet eth0: Link is Down
 173.594967] dwmac-sunxi 4510000.ethernet eth0: Link is Up - 10Mbps/Half - flow control off
 181.786962] dwmac-sunxi 4510000.ethernet eth0: Link is Down
 182.810967] dwmac-sunxi 4510000.ethernet eth0: Link is Up - 10Mbps/Half - flow control off
 184.859001] dwmac-sunxi 4510000.ethernet eth0: Link is Down
 185.882968] dwmac-sunxi 4510000.ethernet eth0: Link is Up - 10Mbps/Half - flow control off
```

图 6-13: 异常 link

原因：PHY 规格书要求 MDC 频率 2.5M，实测 GMAC MDC 频率是 12.5M，导致 PHY 工作异常，在不接网线的情况下，不断打印出协商 10M 或 1GM 的速率；

GMAC 默认 MDC 频率配置低，该问题会在 GMAC2XX 上出现，请检查设备树中是否有如下配置：

```
csr_clk = <4>; # 不再需要，驱动代码里写死了
```

6.7 其他 PHY 配置

注意：不同 PHY 的配置和功能设置不同，需要按照具体 PHY 的规格书和参考电路设置外围电路；

模式配置：

如 RTL8201 可以配置 RMII 或 MII 两个 2 模式，RXDV 拉高，则配置为 RMII 模式；

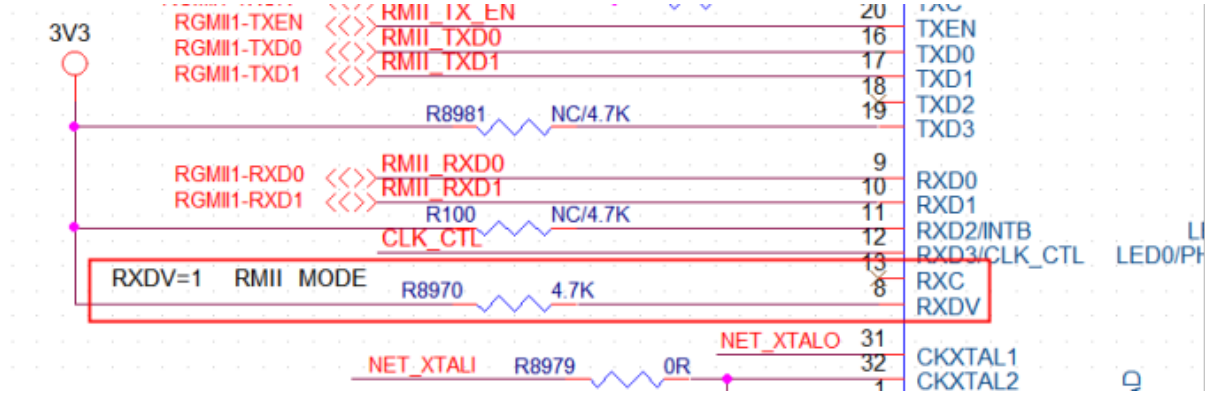


图 6-14: 模式设置

7 以太网带宽测试

平台性能可参考文档《{IC}_Ethernet_benchmark_CN》

7.1 测试步骤

在带宽测试时，请按照如下步骤测试：

- 对端准备一块经过验证的千兆带宽达标（>=900Mbps）或百兆带宽达标的（>=90Mbps）板子/PC；
- 保持测试板卡系统处于低负载状态，关闭无关 DEBUG 选项及不跑非原生应用；
- 将对端与测试板卡用网线直连；
- 指定双方 IP 地址为同一网段，例如：测试板卡 ifconfig eth0 192.168.200.10，对端：ifconfig eth0 192.168.200.9，PC 的 IP 地址修改请参考 7.2 章节；
- 发送测试：对端输入 iperf -s -i 1，测试板卡输入 iperf -c 192.168.200.9 -i 1 -t 60；
- 接收测试：测试板卡输入 iperf -s -i 1，对端输入 iperf -c 192.168.200.10 -i 1 -t 60。

若按照上述步骤测试后，千兆带宽不达标，如下是千兆带宽的影响因素：

- 单核 CPU 算力：其中 DDR 频率、CPU 自身算力、CPU 频率、CPU 的访存能力都会影响到 CPU 算力；
- IP 差异：GMAC 使用 AHB 总线比 GMAC-2XX 的 AXI 总线带宽少 10~40M 左右；
- 系统负载：以太网性能测试属于 CPU 消耗型任务，测试时系统需要保持在低负载的状态下，并且关闭无关的内核 DEBUG 选项；
- 硬件信号：硬件信号时序（保证建立保持时间最优，请使用 8.4 章节节点调试），硬件信号质量；
- 对端的发送与接收能力：以太网测试时确保对端是经过验证的千兆带宽达标的平台；
- 总线抢占：其他模块抢占 GMAC 的 AHB 总线及新 GMAC 的 AXI 总线带宽；
- 中断抢占及调度抢占：Android 固件建议在测试时把 GMAC 中断放在单独的核上（比如大核，命令例如：`echo f0 > /proc/irq/498/smp_affinity`，注：498 是指 GMAC 的中断号，具体请以实际平台为准），Linux 固件由于场景简单，一般无此需求。

注：

- 为什么要使用一块经过验证千兆带宽达标的板子直连对测；

- 千兆达标的板子：对端的带宽吞吐能力会直接影响到测试板卡的吞吐；
- 直连对测：避免引入无关因素，比如路由器的吞吐能力；
- 若测试千兆请使用**千兆网线；**
- 如果对端使用 PC 进行测试，则参考 7.2 配置正确的 PC 环境。

测试环境示意图：

图 7-1: 测试拓扑图

7.2 PC 环境配置流程

7.2.1 PC 如何关闭防火墙

右击状态栏网络图标，打开 “网络和Internet” 设置：

图 7-2: 网络和 Internet

点击Windows 防火墙，关闭网络防火墙，如下：

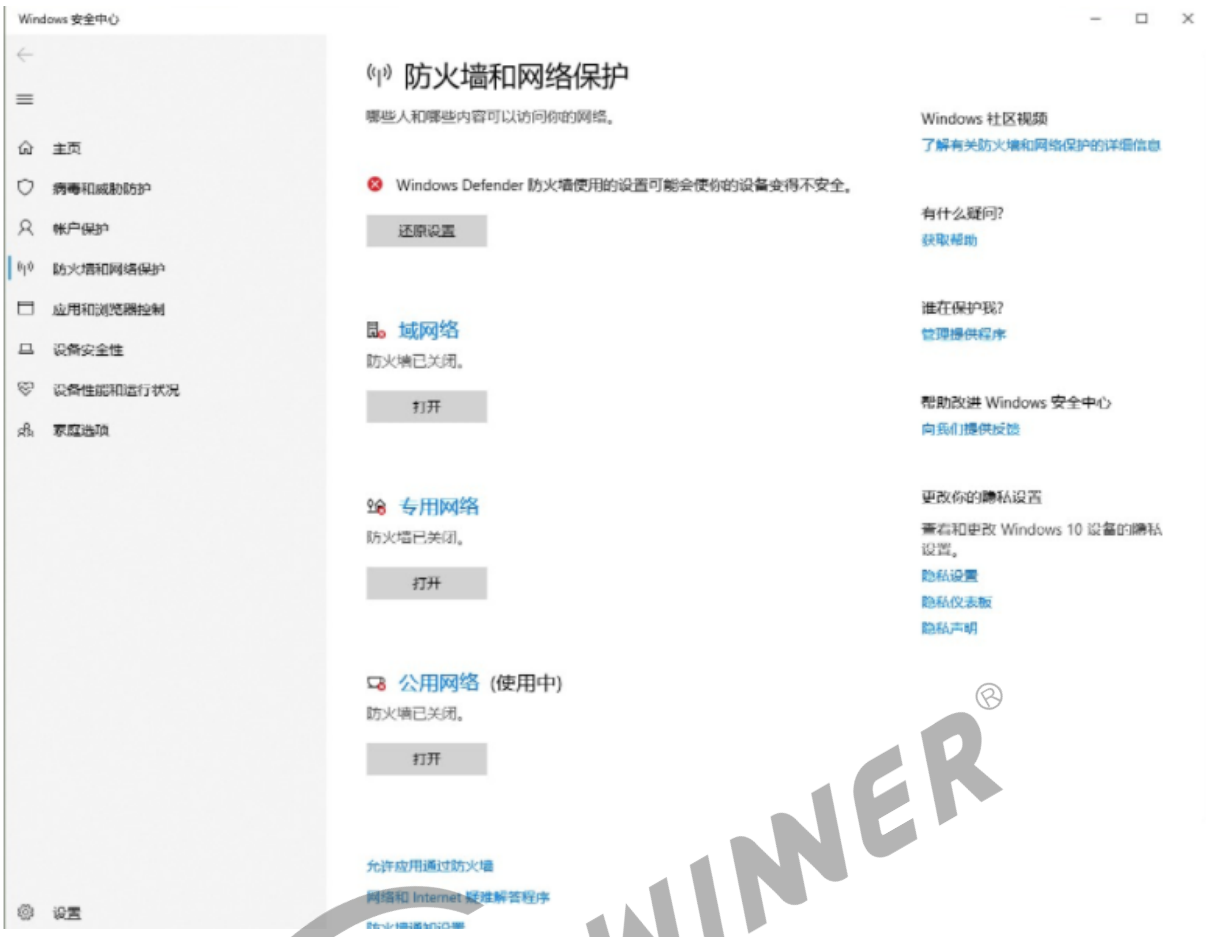


图 7-3: 防火墙

7.2.2 设置 PC 的 IP 地址

右击状态栏网络图标，打开 “网络和Internet” 设置：

图 7-4: 网络和 Internet

找到并选择更改适配器选项，会出现如下显示：

图 7-5: 更改适配器选项

右击属性，选择配置网络IP，如下：

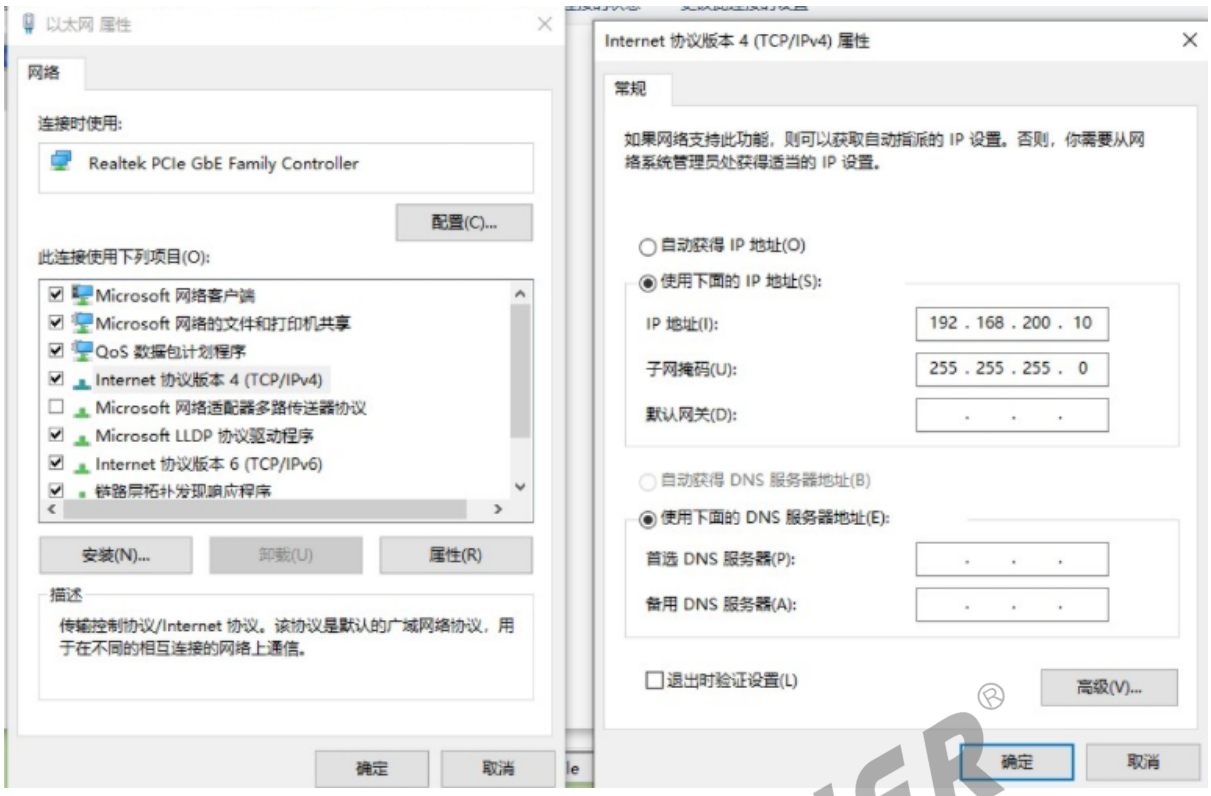


图 7-6: 配置 IP

使用Win + R组合键，输入cmd，进行Dos命令行；

输入ipconfig查看当前 PC 机 IP，如下图：

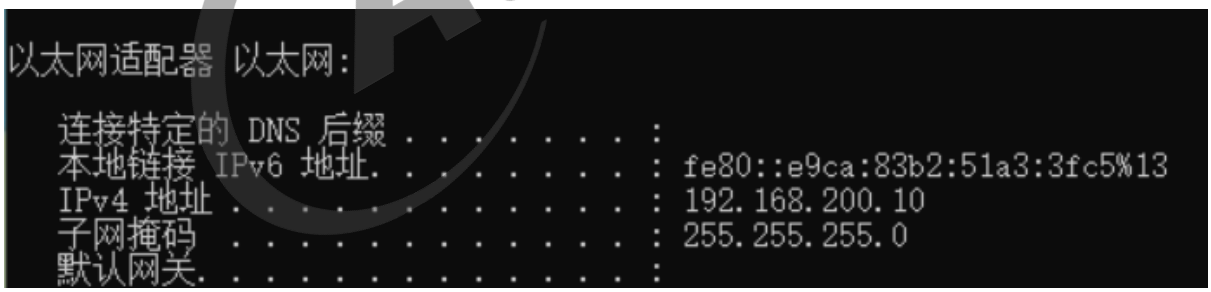


图 7-7: 查看当前 PC 机 IP

7.2.3 PC 如何禁用其他网卡

当PC机有多个网卡时，需要关闭其他网卡，只保留测试网卡。关闭网卡可参考如下步骤：

右击状态栏网络图标，打开 “网络和Internet” 设置，找到并选择更改适配器选项，会出现如下显示：

图 7-8: 更改适配器选项

右击，选择禁用选项：

图 7-9: 选择禁用选项

8 以太网常用调试工具

8.1 ifconfig

使用ifconfig -h查看支持的参数，如下所示：

```
/ # ifconfig -h
Usage:
  ifconfig [-a] [-v] [-s] <interface> [[<AF>] <address>]
  [add <address>[/<prefixlen>]]
  [del <address>[/<prefixlen>]]
  [[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
  [netmask <address>]  [dstaddr <address>]  [tunnel <address>]
  [outfill <NN>] [keepalive <NN>]
  [hw <HW> <address>]  [mtu <NN>]
  [[-]trailers]  [[-]arp]  [[-]allmulti]
  [multicast]  [[-]promisc]
  [mem_start <NN>]  [io_addr <NN>]  [irq <NN>]  [media <type>]
  [txqueuelen <NN>]
  [[-]dynamic]
  [up|down] ...

  <HW>=Hardware Type.
  List of possible hardware types:
    loop (Local Loopback) slip (Serial Line IP) cslip (VJ Serial Line IP)
    slip6 (6-bit Serial Line IP) cslip6 (VJ 6-bit Serial Line IP) adaptive (Adaptive Serial Line IP)
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) tunnel (IPIP Tunnel)
    ppp (Point-to-Point Protocol) hdlc ((Cisco)-HDLC) lapb (LAPB)
    arcnet (ARCnet) dlci (Frame Relay DLCI) frad (Frame Relay Access Device)
    sit (IPv6-in-IPv4) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) x25 (generic X.25) infiniband (InfiniBand)
    eui64 (Generic EUI-64)
  <AF>=Address family. Default: inet
  List of possible address families:
    unix (UNIX Domain) inet (DARPA Internet) inet6 (IPv6)
    ax25 (AMPR AX.25) netrom (AMPR NET/ROM) rose (AMPR ROSE)
    ipx (Novell IPX) ddp (Appletalk DDP) ash (Ash)
    x25 (CCITT X.25)
```

- -a：显示全部接口信息；
- -s：显示摘要信息；

```
/ # ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500       18      0      0 0           19      0      0      0 BMRU
lo       65536        0      0      0 0            0      0      0      0 LRU
/ # netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500       18      0      0 0           19      0      0      0 BMRU
lo       65536        0      0      0 0            0      0      0      0 LRU
```

图 8-1: 显示摘要信息

- add/del：添加对应网口的 ipv6 地址；

```
/ # ifconfig eth0 inet6 add 2001:250:250:250:250:250:250:222/64
/ # ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 2001:250:250:250:250:250:250:222  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::8809:46ff:fe7f:229  prefixlen 64  scopeid 0x20<link>
        ether 8a:09:46:7f:02:29  txqueuelen 1000  (Ethernet)
        RX packets 30  bytes 2348 (2.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1102 (1.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 60

eth1: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 4a:e1:76:1e:98:83  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
```

图 8-2: 添加 ipv6 地址

- netmask：设置子网掩码；

```
/ # ifconfig eth0 netmask 255.255.255.254
/ # ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.10  netmask 255.255.255.254  broadcast 192.168.200.255
        inet6 fe80::786a:2fff:fe06:915e  prefixlen 64  scopeid 0x20<link>
        ether 7a:6a:2f:06:91:5e  txqueuelen 1000  (Ethernet)
        RX packets 27  bytes 1894 (1.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22  bytes 1732 (1.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 60
```

图 8-3: 设置子网掩码

- hw：设置 mac 地址；

```
/ # ifconfig eth0 hw ether 12:34:56:78:90:00
/ # ifconfig -a
eth0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 12:34:56:78:90:00  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 60
```

图 8-4: 设置 mac 地址

- arp：打开或者关闭对应网口是否支持 arp 功能，如下图所示，关闭 arp 协议，ping 不通，打开之后就可以 ping 通；

```
/ #
/ # ifconfig eth0 -arp
/ # ifconfig eth0 192.168.200.10
[   65.624942] [sound  395][MACH simple_parse_of] simple_dai_link_of failed
[   65.788422] sunxi-gmac 5020000.gmac0 eth0: eth0: Type(8) PHY ID 001cc916 at 1 IRQ poll (5020048.mdio0-mii:01)
[   65.801404] sunxi-gmac 5020000.gmac0 eth0: Link is Up - 1Gbps/Full - flow control off
/ #
/ # ping 192.168.200.9
PING 192.168.200.9 (192.168.200.9): 56 data bytes
^C
--- 192.168.200.9 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
/ # ifconfig eth0 arp
/ # ping 192.168.200.9
PING 192.168.200.9 (192.168.200.9): 56 data bytes
64 bytes from 192.168.200.9: seq=0 ttl=64 time=0.867 ms
64 bytes from 192.168.200.9: seq=1 ttl=64 time=0.869 ms
^C
--- 192.168.200.9 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.867/0.868/0.869 ms
```

图 8-5: 关闭 arp

8.2 route

设置和查看路由表都可以用 route 命令，设置内核路由表的命令格式是：

```
# route  [add|del] [-net|-host] target [netmask Nm] [gw Gw] [[dev] If]
```

其中：

- add：添加一条路由规则；
- del：删除一条路由规则；
- -net：目的地址是一个网络；
- -host：目的地址是一个主机；
- target：目的网络或主机；
- netmask：目的地址的网络掩码；
- gw：路由数据包通过的网关；
- dev：为路由指定的网络接口。

示例：

```
/ #
/ # ifconfig eth0 -arp
/ # ifconfig eth0 192.168.200.10
[   65.624942] [sound  395][MACH simple_parse_of] simple_dai_link_of failed
[   65.788422] sunxi-gmac 5020000.gmac0 eth0: eth0: Type(8) PHY ID 001cc916 at 1 IRQ poll (5020048.mdio0-mii:01)
[   65.801404] sunxi-gmac 5020000.gmac0 eth0: Link is Up - 1Gbps/Full - flow control off
/ #
/ # ping 192.168.200.9
PING 192.168.200.9 (192.168.200.9): 56 data bytes
^C
--- 192.168.200.9 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
/ # ifconfig eth0 arp
/ # ping 192.168.200.9
PING 192.168.200.9 (192.168.200.9): 56 data bytes
64 bytes from 192.168.200.9: seq=0 ttl=64 time=0.867 ms
64 bytes from 192.168.200.9: seq=1 ttl=64 time=0.869 ms
^C
--- 192.168.200.9 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.867/0.868/0.869 ms
```

图 8-6: 添加路由表

```
route add -host 192.168.200.9 dev eth0 #添加eth0的目的地址为192.168.200.9
route add -host 192.168.200.9 gw 192.168.200.1 #添加到目的地址192.168.200.9经过网关192.168.200.1
```

8.3 mii_reg

mii_reg 工具是 GMAC 驱动中提供的调试节点，其主要作用是对外部 PHY 寄存器地址进行读写操作。

启动网卡：

```
ifconfig eth0 up
```

进入操作目录：

```
cd sys/devices/platform/soc@3000000/450000.eth
```

注：不同的目录可能会因板卡不同有差异，此处为大致路径。

读取对应的 phy 寄存器：

```
addr: PHY地址
reg: PHY寄存器
val: 数据
```

写操作：

```
echo addr reg val > mii_write; cat mii_write
eg:
echo 0x00 0x1f 0xa43 > mii_write; cat mii_write
```

读操作：

```
echo addr reg > mii_read; cat mii_read
eg:
echo 0x10 0x06 > mii_read; cat mii_read
```

一次读多位寄存器操作: 0x01~0x0a

```
cat mii_reg
```

8.4 delay 参数配置

delay 参数在板级硬件发生变动时，有可能出现千兆时序不匹配的情况，请按照如下步骤进行测试修改：

（1）delay 参数调试节点是 GMAC 驱动中提供的调整千兆 RGMII 接口时序的节点

进入操作目录：

```
cd /sys/devices/platform/soc/gmac/
```

注：不同的目录可能会因板卡不同有差异，此处为大致路径。

调整 rxdelay 时序：

```
#rx_delay：val - rxdelay参数，0~31共32挡，每档将采样时间推迟约130ps
echo val > rx_delay; cat rx_delay
```

调整 txdelay 时序：

```
#tx_delay：val - txdelay参数，0~7共8档，每档将采样时间推迟约536ps
echo val > tx_delay; cat tx_delay
```

（2）如何获取最优参数 - GMAC

一般通过窗口法获取最佳参数，窗口代表网络可以 ping 通时，delay 参数的值

1. 固定 txdelay 参数，调整 rxdelay 参数（通过（1）的节点）
2. 找到 rxdelay 参数的两端窗口，比如 delay 参数为 9 时无法 ping 通，为 10 时可以 ping 通；delay 参数为 20 时可以 ping 通，为 21 时无法 ping 通，则窗口值为 [10，20]，最优 delay 参数为窗口的中间值 15
3. 按照相同方法调整 txdelay 参数时序

（3）如何获取最优参数 - GMAC2XX

集成了自动校准工具，可自动扫描并选择最佳延时窗口，此方法需要连接千兆对端并处于 linkup 状态。

```
echo > /sys/devices/platform/*soc*/*ethernet*/calirate
[15082.594537] TX(0x00): XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[15083.278500] TX(0x01): ----------XXXXXXXXXXXXXXXXXXXX
[15083.958496] TX(0x02): ----------XXXXXXXXXXXXXXXXXXXX
[15084.634495] TX(0x03): ----------XXXXXXXXXXXXXXXXXXXX
[15085.314494] TX(0x04): ----------XXXXXXXXXXXXXXXXXXXX
[15086.002497] TX(0x05): ----------XXXXXXXXXXXXXXXXXXXX
[15086.782494] TX(0x06): XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[15087.562497] TX(0x07): XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
[15087.569250] sunxi:sunxi_stmmac-4510000.ethernet:[INFO]: Calibrate suitable delay tx:3 rx:5
```

（4）固定最优参数

当找到最优 delay 参数时，配置到设备树中，例如：

```
gmac0: eth@05020000 {
        ...
        tx-delay = <5>;
        rx-delay = <15>;
        ...
    };
```

著作权声明

商标声明

免责声明