Rockchip Developer Guide Linux GMAC

文件标识：RK-KF-YF-130

发布版本：V1.0.0

日期：2021-01-16

文件密级：□绝密 □秘密 □内部资料 ■公开

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址： 福建省福州市铜盘路软件园A区18号

网址： www.rock-chips.com

客户服务电话： +86-4007-700-590

客户服务传真： +86-591-83951833

客户服务邮箱： fae@rock-chips.com

前言

概述

本文提供 Rockchip 平台以太网 GMAC 接口的使用文档，用于解决大部分以太网问题。

产品版本

芯片名称	内核版本
ROCKCHIP 芯片	3.10/4.4/4.19

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	吴达超	2021-01-16	初始版本

目录

Rockchip Developer Guide Linux GMAC

1. 代码位置
2. DTS
3. PHY 寄存器读写调试
    3.1 Linux 3.10
    3.2 其它版本
4. MAC 地址
5. 回环测试
6. RGMII Delayline
7. LED 灯
8. WOL
9. MAC To MAC 直连
10. Jumbo Frame
11. PTP1588
    11.1 PC master and RK slave
    11.2 RK master and PC slave
12. 硬件信号测试
13. 问题分析
    13.1 DMA Initialization Failed
    13.2 PHY 初始化失败
    13.3 Link 问题
    13.4 数据不通
        13.4.1 TX
        13.4.2 RX
    13.5 TX queue0 timeout

1. 代码位置

以太网模块的硬件相关的驱动代码主要包括 GMAC 和 PHY。其中 PHY 驱动一般使用通用 PHY 驱动，如果有需要修改特殊寄存器，请使用对应的 PHY 驱动，代码都在 drivers/net/phy。另外，RK322x/RK3328 自带有一个百兆的 PHY 芯片。

- Linux3.10 GMAC 驱动代码 `driver/net/ethernet/rockchip/gmac/*`
- 其它内核 GMAC 驱动代码，高于3.10 的内核版本，GMAC 驱动代码位置 `drivers/net/ethernet/stmicro/stmmac/*`
- RK 内部 EPHY 驱动代码 `drivers/net/phy/rockchip.c`

2. DTS

DTS 的配置参考 Documentation/devicetree/bindings/net/rockchip-dwmac.txt

```
gmac: ethernet@ff290000 {
        compatible = "rockchip,rk3288-gmac";
        reg = <0xff290000 0x10000>;
        interrupts = <GIC_SPI 27 IRQ_TYPE_LEVEL_HIGH>;
        interrupt-names = "macirq";
        rockchip,grf = <&grf>;
        clocks = <&cru SCLK_MAC>,
                 <&cru SCLK_MAC_RX>, <&cru SCLK_MAC_TX>,
                 <&cru SCLK_MACREF>, <&cru SCLK_MACREF_OUT>,
                 <&cru ACLK_GMAC>, <&cru PCLK_GMAC>;
        clock-names = "stmmaceth",
                 "mac_clk_rx", "mac_clk_tx",
                 "clk_mac_ref", "clk_mac_refout",
                 "aclk_mac", "pclk_mac";
        phy-mode = "rgmii";
        pinctrl-names = "default";
        pinctrl-0 = <&rgmii_pins /*&rmii_pins*/>;

        clock_in_out = "input";
        snps,reset-gpio = <&gpio4 7 0>;
        snps,reset-active-low;
        snps,reset-delays-us = <0 10000 1000000>;

        assigned-clocks = <&cru SCLK_MAC>;
        assigned-clock-parents = <&ext_gmac>;
        tx_delay = <0x30>;
        rx_delay = <0x10>;

        status = "ok";
};
```

板级配置需要关注的部分有以下几部分：

- phy-mode：主要分为 RMII 和 RGMII 模式
- snps,reset-gpio：PHY 的硬件复位脚
- snps,reset-delays-us：PHY 的复位时序，三个时间分别表示 PHY 的不同阶段的复位时序，不同的 PHY 的复位时序是不一样的，如果是 snps,reset-active-low 属性，则表示三个时间分别表示 Reset pin 脚拉高，拉低，再拉高的时间；如果是 snps,reset-active-high 属性，则反之
- phy-supply：如果 PHY 的电源是常供方式，可以不用配置；否则，需要配置对应的 regulator
- 时钟配置：请参考本文的第三章节
- pinctrl：RGMII 和 RMII 模式下配置不一样，另外对于时钟方式，如果是输出时钟的 pin 脚，该 pin 脚驱动强度一般也是不一样的，例如 RMII 模式下 ref_clock pin 脚输出时钟时，驱动强度也会配置更大
- tx_delay/rx_delay：RGMII 模式下需要配置该属性，请参考本文的 RGMII Delayline 第八章节

因为不同芯片下的不同模式配置比较多，请参考另外一份文档
《Rockchip_Developer_Guide_Linux_GMAC_Mode_Configuration_CN.pdf》

3. PHY 寄存器读写调试

驱动提供了读写寄存器的接口，目前在不同内核版本上面有两套接口。
路径： /sys/bus/mdio_bus/devices/stmmac-0:00，其中 stmmac-0:00 表示 PHY 地址是 0。

3.1 Linux 3.10

```
/sys/bus/mdio_bus/devices/stmmac-0:00/phy_reg
/sys/bus/mdio_bus/devices/stmmac-0:00/phy_regValue
```

- 写

例如，往 Reg0 写入 0xabcd

```
echo 0x00 > /sys/bus/mdio_bus/devices/stmmac-0:00/phy_reg
echo 0xabcd > /sys/bus/mdio_bus/devices/stmmac-0:00/phy_regValue
```

- 读

例如，读取 Reg0 值

```
echo 0x00 > /sys/bus/mdio_bus/devices/stmmac-0:00/phy_reg
cat /sys/bus/mdio_bus/devices/stmmac-0:00/phy_regValue
```

3.2 其它版本

```
/sys/bus/mdio_bus/devices/stmmac-0:00/phy_registers
```

- 写

例如， 往 Reg0 写入 0xabcd

```
echo 0x00 0xabcd > /sys/bus/mdio_bus/devices/stmmac-0:00/phy_registers
```

- 读

```
cat /sys/bus/mdio_bus/devices/stmmac-0:00/phy_registers
```

该命令会读取 0~31 的所有寄存器，所以可以查看对应的寄存器值。

4. MAC 地址

目前对 MAC 地址的读取策略是，优先使用 DTB 里面的 MAC 地址（uboot 也会写入），之后是烧写在 IDB 中的 MAC 地址，若该地址符合规范，则使用，若不符合或没有烧写，则使用随机生成的地址（重启开机 MAC 地址会变化）。在 RK3399、RK3328/RK3228H 及以后的版本中，对策略进行了完善：优先使用烧写在 IDB 或 vendor Storage 中的 MAC 地址，若该地址符合规范，则使用，若不符合或没有烧写，则随机生成 MAC 地址保存到 Vendor 分区中并使用，重启或恢复出厂设置不会丢失。

MAC 地址烧写工具参考文档《Rockchip_User_Guide_RKDevInfoWriteTool_CN.pdf》。

5. 回环测试

回环测试主要有 MAC 和 PHY 两种回环，具体可参考《Rockchip_Developer_Guide_Linux_GMAC_RGMII_Delayline_CN.pdf》文档里面，对 phy_lb 和 mac_lb 节点的说明。

6. RGMII Delayline

RGMII 接口提供了 tx 和 rx 的 delayline，用于调整 RGMII 时序，如何获取合适的 RGMII Delayline，请参考文档 《Rockchip_Developer_Guide_Linux_GMAC_RGMII_Delayline_CN.pdf》。

7. LED 灯

PHY 有各自的 LED 控制，下面是 RK3228 和 RK3328 里面的 macphy，其它外部 PHY 请参考其 datasheet。下面是 RK3228 和 RK3328 LED 配置：

- RK3228：需要打上补丁 `kernel_4.4_rk322x_phy_led_control.patch`。
- RK3328：通过配置 dts 上的 iomux，例如通过 rx 和 link 控制 led，则配置上对应的 pinctrl。

```
phy: phy@0 {
        compatible = "ethernet-phy-id1234.d400", "ethernet-phy-ieee802.3-c22";
        reg = <0>;
        clocks = <&cru SCLK_MAC2PHY_OUT>;
        resets = <&cru SRST_MACPHY>;
        pinctrl-names = "default";
        pinctrl-0 = <&fephyled_rxm1 &fephyled_linkm1>;
        phy-is-integrated;
};
```

8. WOL

Wake On Lan 功能，对于每个 PHY 来说配置的寄存器是不一样的。目前收录的补丁，包含了 RTL8211E/F，RTL8201F。

9. MAC To MAC 直连

参考文档《Rockchip_Developer_Guide_Linux_MAC_TO_MAC_CN.pdf》。

10. Jumbo Frame

从 RV1126/1109 芯片开始支持 Junbro Frame 9K，需要将测试网络 MTU 配置成 9000，以下是测试结果：

```
<pre>[root@Puma:/]# ping -s 9000 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 9000(9028) bytes of data.
9008 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=0.784 ms
9008 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=0.675 ms
9008 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=0.666 ms
9008 bytes from 192.168.1.100: icmp_seq=4 ttl=64 time=0.656 ms
9008 bytes from 192.168.1.100: icmp_seq=5 ttl=64 time=0.677 ms
9008 bytes from 192.168.1.100: icmp_seq=6 ttl=64 time=0.637 ms
9008 bytes from 192.168.1.100: icmp_seq=7 ttl=64 time=0.641 ms
9008 bytes from 192.168.1.100: icmp_seq=8 ttl=64 time=0.692 ms
9008 bytes from 192.168.1.100: icmp_seq=9 ttl=64 time=0.656 ms
```

11. PTP1588

从 RV1126/1109 芯片开始支持 PTP1588，以下是测试结果：

11.1 PC master and RK slave

```
ubuntu@thinkpad: sudo ptp4l -i enp0s31f6 -m -H
ptp4l[1790161.443]: selected /dev/ptp0 as PTP clock
ptp4l[1790161.443]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1790161.443]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1790168.489]: port 1: LISTENING to MASTER on
ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES
ptp4l[1790168.489]: selected local clock 54e1ad.fffe.dfa454 as best master
ptp4l[1790168.490]: assuming the grand master role
```

```
[root@Puma:/]# ptp4l -i eth0 -m -H -s
ptp4l[39.868]: selected /dev/ptp0 as PTP clock
[   39.871092] rk_gmac-dwmac ffc40000.ethernet eth0: stmmac_hwtstamp_set config
flags:0x0, tx_type:0x1, rx_filter:0xc
[   39.872029] stmmac_hwtstamp_set, value: 0x17e03
ptp4l[39.870]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[39.871]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[41.251]: port 1: new foreign master 54e1ad.fffe.dfa454-1
[   43.817340] rk_gmac-dwmac ffc40000.ethernet eth0: stmmac_hwtstamp_set config
flags:0x0, tx_type:0x1, rx_filter:0xc
[   43.818262] stmmac_hwtstamp_set, value: 0x17e03
ptp4l[45.251]: selected best master clock 54e1ad.fffe.dfa454
ptp4l[45.251]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[49.251]: master offset      -1608 s0 freq      +0 path delay      5691
ptp4l[50.251]: master offset      -5579 s0 freq      +0 path delay      9435
ptp4l[51.251]: master offset      -4831 s2 freq    +748 path delay      9435
ptp4l[51.251]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[52.251]: master offset      12189 s2 freq  +12937 path delay      7563
ptp4l[53.251]: master offset      14413 s2 freq  +18818 path delay      8287
ptp4l[54.251]: master offset      10712 s2 freq  +19441 path delay      8861
ptp4l[55.251]: master offset       7185 s2 freq  +19127 path delay      8861
ptp4l[56.251]: master offset       3234 s2 freq  +17332 path delay      9435
ptp4l[57.251]: master offset       1787 s2 freq  +16855 path delay      9454
ptp4l[58.251]: master offset        785 s2 freq  +16389 path delay      9454
ptp4l[59.251]: master offset         89 s2 freq  +15928 path delay      9473
ptp4l[60.251]: master offset         31 s2 freq  +15897 path delay      9454
ptp4l[61.251]: master offset        -71 s2 freq  +15804 path delay      9454
ptp4l[62.251]: master offset       -100 s2 freq  +15754 path delay      9406
ptp4l[63.251]: master offset        -27 s2 freq  +15797 path delay      9406
ptp4l[64.251]: master offset        -69 s2 freq  +15747 path delay      9395
ptp4l[65.251]: master offset         29 s2 freq  +15824 path delay      9395
ptp4l[66.251]: master offset        -73 s2 freq  +15731 path delay      9395
ptp4l[67.251]: master offset         32 s2 freq  +15814 path delay      9388
ptp4l[68.251]: master offset        -20 s2 freq  +15772 path delay      9388
ptp4l[69.251]: master offset       -104 s2 freq  +15682 path delay      9395
ptp4l[70.251]: master offset        -56 s2 freq  +15699 path delay      9395
ptp4l[71.251]: master offset         24 s2 freq  +15762 path delay      9388
ptp4l[72.251]: master offset         11 s2 freq  +15756 path delay      9395
```

11.2 RK master and PC slave

```
[root@Puma:/]# ptp4l -i eth0 -m -H
ptp4l[15.668]: selected /dev/ptp0 as PTP clock
ptp4l[15.670]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[15.670]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[22.120]: port 1: LISTENING to MASTER on ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES
ptp4l[22.120]: selected local clock aadc46.fffe.5da6d9 as best master
ptp4l[22.121]: assuming the grand master role
```

```
ubuntu@thinkpad: sudo ptp4l -i enp0s31f6 -m -H -s
ptp4l[1879661.603]: selected /dev/ptp0 as PTP clock
ptp4l[1879661.603]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1879661.603]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1879662.249]: port 1: new foreign master aadc46.fffe.5da6d9-1
ptp4l[1879665.849]: selected best master clock aadc46.fffe.5da6d9
ptp4l[1879665.849]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[1879667.649]: master offset         49 s0 freq   -9515 path delay      9364
ptp4l[1879668.549]: master offset        128 s2 freq   -9436 path delay      9338
ptp4l[1879668.549]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[1879669.449]: master offset        256 s2 freq   -9180 path delay      9338
ptp4l[1879670.349]: master offset       -230 s2 freq   -9589 path delay      9338
ptp4l[1879671.249]: master offset       -399 s2 freq   -9827 path delay      9360
ptp4l[1879672.149]: master offset        142 s2 freq   -9406 path delay      9360
ptp4l[1879673.049]: master offset        232 s2 freq   -9273 path delay      9347
ptp4l[1879673.949]: master offset       -303 s2 freq   -9739 path delay      9347
ptp4l[1879674.849]: master offset       -267 s2 freq   -9794 path delay      9338
ptp4l[1879675.749]: master offset        327 s2 freq   -9280 path delay      9335
ptp4l[1879676.649]: master offset        405 s2 freq   -9104 path delay      9335
ptp4l[1879677.549]: master offset       -156 s2 freq   -9543 path delay      9335
ptp4l[1879678.449]: master offset       -178 s2 freq   -9612 path delay      9335
ptp4l[1879679.349]: master offset       -100 s2 freq   -9587 path delay      9335
ptp4l[1879680.249]: master offset        -73 s2 freq   -9590 path delay      9335
ptp4l[1879681.149]: master offset        -79 s2 freq   -9618 path delay      9344
ptp4l[1879682.049]: master offset        -76 s2 freq   -9639 path delay      9344
ptp4l[1879682.949]: master offset        -59 s2 freq   -9645 path delay      9329
ptp4l[1879683.849]: master offset        -31 s2 freq   -9634 path delay      9329
ptp4l[1879684.750]: master offset         22 s2 freq   -9591 path delay      9329
ptp4l[1879685.650]: master offset         -9 s2 freq   -9615 path delay      9337
ptp4l[1879686.550]: master offset        -31 s2 freq   -9640 path delay      9337
ptp4l[1879687.450]: master offset         -3 s2 freq   -9621 path delay      9337
ptp4l[1879688.350]: master offset        -15 s2 freq   -9634 path delay      9351
```

12. 硬件信号测试

参考 Rockchip 硬件发布的信号测试文档，包括 RMII 或 RGMII， PHY 眼图测试。
《瑞芯硬件部100base-t测试指南-V1.1.doc》，《瑞芯硬件部1000base-t测试指南_V1.0.doc》。

13. 问题分析

13.1 DMA Initialization Failed

如果 GMAC 的驱动开机 log 上出现打印：`DMA engine initialization failed`，可以认为是 GMAC 的工作时钟出问题了。先测量时钟引脚是否有时钟，时钟频率以及幅度等指标是否正常，主要确认以下几个方面：

- IOMUX 出错，检查时钟脚寄存器值是否正确
- 时钟方向以及配置与硬件不匹配，参考本文第四章节的时钟设置
- 检查 clock tree 和 CRU 寄存器，确认时钟频率大小和时钟是否有使能

13.2 PHY 初始化失败

如果 GMAC 的驱动开机 log 上出现打印：No PHY found 或者 Cannot attach to PHY，表示找不到 PHY。驱动会通过 MDIO 先读取 PHY 的 ID，可以测量 MDC 和 MDIO 波形，波形是否正常，该总线类似于 I2C，MDC 频率要求是小于 2.5M。一般来说，找不到 PHY 有以下几个原因：

- 检查 MDC/MDIO IOMUX 寄存器值是否正确
- PHY 供电是否正常
- Reset IO 配置不正确
- Reset IO 时序不满足 PHY datasheet 要求，不同 PHY 的时序要求不一致，具体配置参考本文 DTS 章节
- 测试 MDIO/MDC 波形是否异常，其中 MDC 时钟频率要求小于 2.5M

13.3 Link 问题

如果出现了 Link 问题，有个排除法，即将 MDC/MDIO 与主控断开，与电脑直连，查看电脑端是否有同样的问题，以此排除软件上的干扰，那么需要重点排查下硬件上的影响，先测试 TXN/P 以及 RXN/P 是否有 Link 波形。

如果不断出现 Link up/Link down，可能原因 PHY 收到了错误的数据，

- EEE 模式下，发送的波形在 delayline 配置错误的情况下可能会导致不断 link up/down
- 供给 PHY 的时钟不对也会导致该问题

13.4 数据不通

首先排查一下是否是 TX 问题，或者 RX，还是二者都有问题。

13.4.1 TX

通过 ifconfig -a 查看 eth0 节点的 TX packets 是否在不断增加，如果为0，则有可能网线没有 link 上。通过查看节点可以看到网线是否是连接上的，carrier 为1表示是 link up，反之 0 为 link down。例如 RK3328：

```
console:/ # cat /sys/devices/platform/ff550000.ethernet/net/eth0/carrier
1
```

```
eth0      Link encap:Ethernet  HWaddr 16:21:8d:d9:67:0b  Driver rk_gmac-dwmac
          inet6 addr: fe80::c43d:3e5d:533:b7ea/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:2848
          Interrupt:45
```

假设 TX packets 是在不断增加，表示 TX 数据在 GMAC 有发出数据。

将板卡与 PC 连在同一个局域网内，在板卡上 ping PC，同时在 PC 端通过抓包工具（比如 Wireshark）抓包查看，如果有抓到板卡发过来的数据，表示 TX 数据是通的。如果没有抓到，需要确认 TX 在哪个链路位置上出现了异常，可以测试 GMAC 的 TX Clock 与 TX Data 的波形，来排除是 MAC 还是 PHY 出现了问题。MAC 可以检查以下几个方面：

- 检查 TX Clock/TX Data 的 iomux
- TXC 时钟是否正确
- RGMII 模式时，Tx Delayline 配置是否正确

PHY 端也可以测试 PHY 的 TXN/P 信号确认 PHY 是否有数据发出；对于不同的 PHY 来说，其配置可能是不一样，具体需要查看其 Datasheet。

13.4.2 RX

通过以上排查确定不是 TX 问题，重点排查 RX；连接上网线后通过 ifconfig -a 查看 eth0 节点的 RX packets 是否在不断增加，如果为0，表示 GMAC RX 没有收到数据

```
eth0      Link encap:Ethernet  HWaddr 16:21:8d:d9:67:0b  Driver rk_gmac-dwmac
          inet6 addr: fe80::c43d:3e5d:533:b7ea/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:341 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:48928 TX bytes:3741
          Interrupt:355
```

同样可以测试 PHY 的 RXN/P，以及 GMAC 的 RX Clock/RX Data，来排除是 MAC 还是 PHY 出现了问题。MAC 可以检查以下几个方面：

- 检查 RX Clock/RX Data 的 iomux

- RXC 时钟是否正确
- RGMII Tx Delayline 配置是否正确
- RGMII 模式时，Rx Delayline 配置是否正确

假设 TX packets 是在不断增加，但以太网还是不正常通讯，则有可能是以下原因：

- RMII 模式下 MAC 和 PHY 的参考时钟不是同一个
- PHY 模式配置不对，例如硬件上配置成了 MII 模式

13.5 TX queue0 timeout

认为是 TX 无法发出，一般是控制器异常了，可能是以下几个原因引起的控制器异常：

- 时钟问题，检查时钟配置是否正确，参考本文第三章节
- PHY 时序问题，PHY 的复位时序不对导致 PHY 给的时钟不对
- PHY 硬件问题，导致出现了冲突检测，无法发送数据
- 逻辑电压太低