

Rockchip MAC TO MAC Linux Guide

文件标识: RK-KF-YF-128

发布版本: V1.0.0

日期: 2020-09-21

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

版权所有 © 2020 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

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

本文提供一个 MAC 连接 MAC，没有 PHY 的方案，适用于两个 AP 通过 MAC 相连，或者 AP 的 MAC 和 SWITCH 的 MAC 相连。两个 AP 经 MAC 相连的方案，通过该方式，可以节省两个 PHY 的成本。分为 RMII 和 RGMII 两种连接方式。

产品版本

芯片名称	内核版本
所有芯片	所有版本

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	2020-09-21	初始版本

目录

Rockchip MAC TO MAC Linux Guide

1. RMII
 - 1.1 硬件连接
 - 1.2 软件配置
 - 1.3 测试结果
 - 1.3.1 TCP 测试
 - 1.3.2 UDP 测试
 - 1.3.3 PING 测试
2. RGMII
 - 2.1 硬件连接
 - 2.2 软件配置
 - 2.3 Delayline 配置

1. RMII

1.1 硬件连接

RMII 直连如下所示，其中 RX_ERR 需要接地。

MAC0	--RMII--	MAC1
TXD[1:0]	-----	RXD[1:0]
TX_EN	-----	RX_DV
REF_CLK	-----	REF_CLK
RXD[1:0]	-----	TXD[1:0]
RX_DV	-----	TX_EN
RX_ERR	-----	GND
GND	-----	RX_ERR

1.2 软件配置

以 PX30 和 RV1126 为例，RV1126 输出 50M 参考时钟，PX30 配置为时钟输入模式。

- rv1126 clock output:

该补丁为 Linux4.19 内核下的。

```
diff --git a/arch/arm/boot/dts/rv1126-evb-v10.dtsi b/arch/arm/boot/dts/rv1126-evb-v10.dtsi
index 396ef1516054..a384e657ebac 100644
--- a/arch/arm/boot/dts/rv1126-evb-v10.dtsi
+++ b/arch/arm/boot/dts/rv1126-evb-v10.dtsi
@@ -568,26 +568,21 @@
    };

    &gmac {
-       phy-mode = "rgmii";
-       clock_in_out = "input";
+       phy-mode = "rmii";
+       clock_in_out = "output";

-       snps,reset-gpio = <&gpio3 RK_PA0 GPIO_ACTIVE_LOW>;
-       snps,reset-active-low;
-       /* Reset time is 20ms, 100ms for rtl8211f */
-       snps,reset-delays-us = <0 20000 100000>;
-
-       assigned-clocks = <&cru CLK_GMAC_SRC>, <&cru CLK_GMAC_TX_RX>, <&cru CLK_GMAC_ETHERNET_OUT>;
-       assigned-clock-parents = <&cru CLK_GMAC_SRC_M1>, <&cru RGMII_MODE_CLK>;
-       assigned-clock-rates = <125000000>, <0>, <25000000>;
+       assigned-clocks = <&cru CLK_GMAC_SRC_M1>, <&cru CLK_GMAC_SRC>, <&cru CLK_GMAC_TX_RX>;
+       assigned-clock-rates = <0>, <50000000>;
```

```

+     assigned-clock-parents = <&cru CLK_GMAC_RGMII_M1>, <&cru
CLK_GMAC_SRC_M1>, <&cru RMII_MODE_CLK>;

+     pinctrl-names = "default";
-     pinctrl-0 = <&rgmiim1_pins &clk_out_ethernetm1_pins>;
-
-     tx_delay = <0x2a>;
-     rx_delay = <0x1a>;
+     pinctrl-0 = <&rmii_m1_pins &gmac_clk_m1_drv_level0_pins>;

-     phy-handle = <&phy>;
+     status = "okay";
+     fixed-link {
+         speed = <100>;
+         full-duplex;
+     };
+ };

&i2c0 {

```

- px30 clock input:

该改动为 Linux4.4 内核下的补丁。

```

diff --git a/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10-linux.dts
b/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10-linux.dts
index 7693764a0dbe..6f548808e3ec 100644
--- a/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10-linux.dts
+++ b/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10-linux.dts
@@ -326,11 +326,17 @@

&gmac {
+     phy-supply = <&vcc_phy>;
-     clock_in_out = "output";
-     snps,reset-gpio = <&gpio2 13 GPIO_ACTIVE_LOW>;
-     snps,reset-active-low;
-     snps,reset-delays-us = <0 50000 50000>;
+     clock_in_out = "input";
+     assigned-clocks = <&cru SCLK_GMAC>;
+     assigned-clock-parents = <&gmac_clk_in>;
+     pinctrl-names = "default";
+     pinctrl-0 = <&rmii_pins &mac_refclk>;
+     status = "okay";
+
+     fixed-link {
+         speed = <100>;
+         full-duplex;
+     };
+ };

&gpu {

```

```
diff --git a/arch/arm64/configs/px30_linux_defconfig
b/arch/arm64/configs/px30_linux_defconfig
index b73d05c8ad26..486e971c2d90 100644
--- a/arch/arm64/configs/px30_linux_defconfig
+++ b/arch/arm64/configs/px30_linux_defconfig
@@ -136,6 +136,7 @@ CONFIG_STMMAC_ETH=y
 # CONFIG_NET_VENDOR_VIA is not set
 # CONFIG_NET_VENDOR_WIZNET is not set
 CONFIG_ROCKCHIP_PHY=y
+CONFIG_FIXED_PHY=y
 CONFIG_USB_RTL8150=y
 CONFIG_USB_RTL8152=y
 CONFIG_USB_NET_CDC_MBIM=y
```

1.3 测试结果

以 PX30 和 RV1126 为例的测试结果。

1.3.1 TCP 测试

- RV1126 -> PX30

```
[root@RV1126_RV1109:/]# iperf -c 192.168.1.101 -i 1 -t 10
-----
Client connecting to 192.168.1.101, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[ 3] local 192.168.1.100 port 48618 connected with 192.168.1.101 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec  11.6 MBytes  97.5 Mbits/sec
[ 3] 1.0- 2.0 sec  11.0 MBytes  94.3 Mbits/sec
[ 3] 2.0- 3.0 sec  11.1 MBytes  93.3 Mbits/sec
[ 3] 3.0- 4.0 sec  11.3 MBytes  93.3 Mbits/sec
[ 3] 4.0- 5.0 sec  11.2 MBytes  94.4 Mbits/sec
[ 3] 5.0- 6.0 sec  11.3 MBytes  94.3 Mbits/sec
[ 3] 6.0- 7.0 sec  11.2 MBytes  94.3 Mbits/sec
[ 3] 7.0- 8.0 sec  11.3 MBytes  93.3 Mbits/sec
[ 3] 8.0- 9.0 sec  11.1 MBytes  94.3 Mbits/sec
[ 3] 9.0-10.0 sec  11.2 MBytes  93.3 Mbits/sec
[ 3] 0.0-10.0 sec  112 MBytes  94.0 Mbits/sec
```

- PX30 -> RV1126

```
[root@px30_64:/]# iperf -c 192.168.1.100 -i 1 -t 10
-----
Client connecting to 192.168.1.100, TCP port 5001
TCP window size: 45.0 KByte (default)
-----
[ 3] local 192.168.1.101 port 52690 connected with 192.168.1.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec  11.5 MBytes  96.5 Mbits/sec
[ 3] 1.0- 2.0 sec  11.2 MBytes  94.4 Mbits/sec
```

```
[ 3] 2.0- 3.0 sec 11.4 MBytes 95.4 Mbits/sec
[ 3] 3.0- 4.0 sec 11.1 MBytes 93.3 Mbits/sec
[ 3] 4.0- 5.0 sec 11.2 MBytes 94.4 Mbits/sec
[ 3] 5.0- 6.0 sec 11.1 MBytes 93.3 Mbits/sec
[ 3] 6.0- 7.0 sec 11.4 MBytes 95.4 Mbits/sec
[ 3] 7.0- 8.0 sec 11.2 MBytes 94.4 Mbits/sec
[ 3] 8.0- 9.0 sec 11.1 MBytes 93.3 Mbits/sec
[ 3] 9.0-10.0 sec 11.2 MBytes 94.4 Mbits/sec
[ 3] 0.0-10.0 sec 113 MBytes 94.4 Mbits/sec
```

1.3.2 UDP 测试

- RV1126 -> PX30

```
[root@RV1126_RV1109:/]# iperf -c 192.168.1.101 -i 1 -t 10 -u -b 100M
-----
Client connecting to 192.168.1.101, UDP port 5001
Sending 1470 byte datagrams, IPG target: 112.15 us (kalman adjust)
UDP buffer size: 160 KByte (default)
-----
[ 3] local 192.168.1.100 port 48888 connected with 192.168.1.101 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec 11.5 MBytes 96.3 Mbits/sec
[ 3] 1.0- 2.0 sec 11.4 MBytes 95.7 Mbits/sec
[ 3] 2.0- 3.0 sec 11.4 MBytes 95.9 Mbits/sec
[ 3] 3.0- 4.0 sec 11.4 MBytes 95.5 Mbits/sec
[ 3] 4.0- 5.0 sec 11.4 MBytes 95.6 Mbits/sec
[ 3] 5.0- 6.0 sec 11.4 MBytes 95.6 Mbits/sec
[ 3] 6.0- 7.0 sec 11.4 MBytes 95.6 Mbits/sec
[ 3] 7.0- 8.0 sec 11.4 MBytes 96.0 Mbits/sec
[ 3] 8.0- 9.0 sec 11.4 MBytes 95.7 Mbits/sec
[ 3] 9.0-10.0 sec 11.4 MBytes 95.6 Mbits/sec
[ 3] 0.0-10.0 sec 114 MBytes 95.7 Mbits/sec
[ 3] Sent 81437 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec 114 MBytes 95.7 Mbits/sec 0.000 ms 0/81437 (0%)
```

- PX30 -> RV1126

```
[root@px30_64:/]# iperf -c 192.168.1.100 -i 1 -t 10 -u -b 100M
-----
Client connecting to 192.168.1.100, UDP port 5001
Sending 1470 byte datagrams, IPG target: 112.15 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 3] local 192.168.1.101 port 41144 connected with 192.168.1.100 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec 11.3 MBytes 95.0 Mbits/sec
[ 3] 1.0- 2.0 sec 11.4 MBytes 95.6 Mbits/sec
[ 3] 2.0- 3.0 sec 11.4 MBytes 95.6 Mbits/sec
[ 3] 3.0- 4.0 sec 11.3 MBytes 95.0 Mbits/sec
[ 3] 4.0- 5.0 sec 11.4 MBytes 96.0 Mbits/sec
[ 3] 5.0- 6.0 sec 11.2 MBytes 94.3 Mbits/sec
[ 3] 6.0- 7.0 sec 11.4 MBytes 95.6 Mbits/sec
[ 3] 7.0- 8.0 sec 11.4 MBytes 95.6 Mbits/sec
```

```
[ 3] 8.0- 9.0 sec 11.4 MBytes 95.7 Mbits/sec
[ 3] 0.0-10.0 sec 114 MBytes 95.4 Mbits/sec
[ 3] Sent 81133 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec 114 MBytes 95.4 Mbits/sec 0.000 ms 0/81133 (0%)
```

1.3.3 PING 测试

- RV1126 -> PX30

```
[root@RV1126_RV1109:/]# ping -s 65500 192.168.1.101 -c 100
PING 192.168.1.101 (192.168.1.101) 65500(65528) bytes of data.
65508 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=12.5 ms
65508 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=13.1 ms
65508 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=50.8 ms
65508 bytes from 192.168.1.101: icmp_seq=4 ttl=64 time=12.5 ms
65508 bytes from 192.168.1.101: icmp_seq=5 ttl=64 time=12.6 ms
65508 bytes from 192.168.1.101: icmp_seq=6 ttl=64 time=12.5 ms
.....
65508 bytes from 192.168.1.101: icmp_seq=95 ttl=64 time=12.7 ms
65508 bytes from 192.168.1.101: icmp_seq=96 ttl=64 time=12.5 ms
65508 bytes from 192.168.1.101: icmp_seq=97 ttl=64 time=12.6 ms
65508 bytes from 192.168.1.101: icmp_seq=98 ttl=64 time=14.5 ms
65508 bytes from 192.168.1.101: icmp_seq=99 ttl=64 time=46.6 ms
65508 bytes from 192.168.1.101: icmp_seq=100 ttl=64 time=12.9 ms

--- 192.168.1.101 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99155ms
rtt min/avg/max/mdev = 12.369/15.634/15.890/0.572 ms
```

- PX30 -> RV1126

```
[root@px30_64:/]# ping -s 65500 192.168.1.100 -c 100
PING 192.168.1.100 (192.168.1.100) 65500(65528) bytes of data.
65508 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=12.8 ms
65508 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=12.9 ms
65508 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=12.5 ms
65508 bytes from 192.168.1.100: icmp_seq=4 ttl=64 time=12.8 ms
65508 bytes from 192.168.1.100: icmp_seq=5 ttl=64 time=12.4 ms
65508 bytes from 192.168.1.100: icmp_seq=6 ttl=64 time=13.1 ms
65508 bytes from 192.168.1.100: icmp_seq=7 ttl=64 time=12.3 ms
65508 bytes from 192.168.1.100: icmp_seq=8 ttl=64 time=12.6 ms
.....
65508 bytes from 192.168.1.100: icmp_seq=95 ttl=64 time=12.3 ms
65508 bytes from 192.168.1.100: icmp_seq=96 ttl=64 time=13.0 ms
65508 bytes from 192.168.1.100: icmp_seq=97 ttl=64 time=12.7 ms
65508 bytes from 192.168.1.100: icmp_seq=98 ttl=64 time=12.6 ms
65508 bytes from 192.168.1.100: icmp_seq=99 ttl=64 time=12.8 ms
65508 bytes from 192.168.1.100: icmp_seq=100 ttl=64 time=12.6 ms

--- 192.168.1.100 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99184ms
rtt min/avg/max/mdev = 12.177/12.748/14.039/0.384 ms
```


2. RGMII

2.1 硬件连接

RGMII 直连如下所示。

```
MAC0      --RGMII--  MAC1

TXD[3:0]  -----  RXD[3:0]
TX_EN     -----  RX_DV
TX_CLK    -----  RX_CLK
RXD[3:0]  -----  TXD[3:0]
RX_DV     -----  TX_EN
RX_CLK    -----  TX_CLK
```

2.2 软件配置

以两个 RK3399 直连为例，需要能输出 125M TXC 时钟，配置为时钟输出模式。
该补丁为 Linux4.4 内核下的。

```
diff --git a/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
b/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
index a4076b888f7d..27a853b48c8a 100644
--- a/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
@@ -216,17 +216,23 @@
    &gmac {
        phy-supply = <vcc_phy>;
        phy-mode = "rgmii";
-       clock_in_out = "input";
+       clock_in_out = "output";
        snps,reset-gpio = <gpio3 15 GPIO_ACTIVE_LOW>;
        snps,reset-active-low;
        snps,reset-delays-us = <0 10000 50000>;
        assigned-clocks = <&cru SCLK_RMII_SRC>;
-       assigned-clock-parents = <&clkin_gmac>;
+       assigned-clock-parents = <&cru SCLK_MAC>;
+       assigned-clock-rates = <125000000>;
        pinctrl-names = "default";
        pinctrl-0 = <&rgmii_pins>;
        tx_delay = <0x28>;
        rx_delay = <0x11>;
        status = "okay";

+       fixed-link {
+           speed = <1000>;
+           full-duplex;
+       }
    };
```

2.3 Delayline 配置

RGMII 接口需要配置 Delayline，一般的做法是通过 PHY 来扫这个窗口，但 MAC To MAC 方式没有 PHY，所以现在通过示波器来测量 TX 的 Delayline。关闭两个 MAC 的 RX Delayline，调节 TX 的 Delayline，使得 Delay 在 1.5-2ns 之间。

