



Rio Chan

## 摘要

本应用手册介绍了如何将德州仪器 (TI) 的 10/100 以太网 Phy DP83825 合并到 TI Sitara AM335 BeagleBoneBlack 上。

## 内容

1 引言.....	2
2 器件概述.....	2
3 将 DP83825 合并到 Sitara 上的步骤.....	4
3.1 第 1 步 - DP83825 与 DP83822.....	5
3.2 第 2 步 - 检查 8710A 引脚与 AM335.....	5
3.3 第 3 步 - 从 AM335 EVM - BBB 上取出 LAN8710A.....	6
3.4 第 4 步 - 检查 AM335 (BBB) RMI 连接引脚.....	6
3.5 第 5 步 - 检查 DP83825 连接引脚.....	7
3.6 第 6 步 - 使用 PinMux 工具生成 DTS/DTSI 文件。.....	8
3.7 第 7 步 - DP83822 代码库审查.....	9
3.8 第 8 步 - DP83825 代码库审查/补丁修改.....	9
3.9 第 9 步 - 修补 Linux uboot/内核.....	10
3.10 第 10 步 - 更改菜单配置.....	10
3.11 第 11 步 - 构建元件 (menuconfig/dtb/zimage).....	12
3.12 第 12 步 - ( 可选 ) 用于 DTS 构建的重要修复.....	12
3.13 第 13 步 - 将构建文件复制到 SD 卡上.....	12
3.14 第 14 步 - 检查 DP83825 的寄存器。.....	13
3.15 第 15 步 - 协助进行调试的 Linux 命令.....	13
3.16 第 16 步 - Linux ethtool 命令转储 ( 示例 ) .....	14
3.17 第 17 步 - 用 Linux dmesg 检查以太网驱动程序状态。.....	16
3.18 第 18 步 - 包含详细日志分析的测试结果 ( 成功情况 ) 。.....	16
3.19 第 19 步 - YouTube 演示视频.....	17
4 需要的硬件和软件.....	17
4.1 硬件.....	17
4.2 软件.....	17
5 参考文献.....	18
6 修订历史记录.....	18

## 插图清单

图 2-1. Sitara MDIO/MII 架构示例.....	2
图 2-2. Sitara MDIO/MII 详细架构.....	3
图 3-1. 合并概念.....	4
图 3-2. TI 以太网 Phy 比较.....	5
图 3-3. LAN8710A MII 引脚与 AM335.....	5
图 3-4. 取出 LAN8710A.....	6
图 3-5. AM335 RMI 选择.....	6
图 3-6. DP83825 所需的引脚.....	7
图 3-7. AM335 + DP83825 连接.....	7
图 3-8. TI PinMux 工具 ( 云版本 ) .....	8
图 3-9. PinMux 工具中用于 DP83825 的 RMI 1 .....	8

图 3-10. DP83822 Linux Phy 驱动程序.....	9
图 3-11. DP83825 Linux Phy 驱动程序补丁.....	9
图 3-12. 菜单配置步骤 A.....	10
图 3-13. 菜单配置步骤 B.....	11
图 3-14. 菜单配置步骤 C.....	11
图 3-15. YouTube 演示视频.....	17

## 商标

所有商标均为其各自所有者的财产。

## 1 引言

DP83825 是一款低成本以太网 Phy。本文档确认 DP83825 Phy 与 Linux 平台兼容，并将超小型 TI Linux 平台 EVM (BeagleBoneBlack) 作为基础。请按照本文档中的步骤，按规定使用 DP83825 Phy。

## 2 器件概述

图 2-1 是一个 Sitara MDIO/MII 架构示例。在图 2-1 中，MDIO 控制接口用于控制 Phy (例如：DP83825)，而 MII 接口用作 CPU 与 Phy 之间的数据通道。

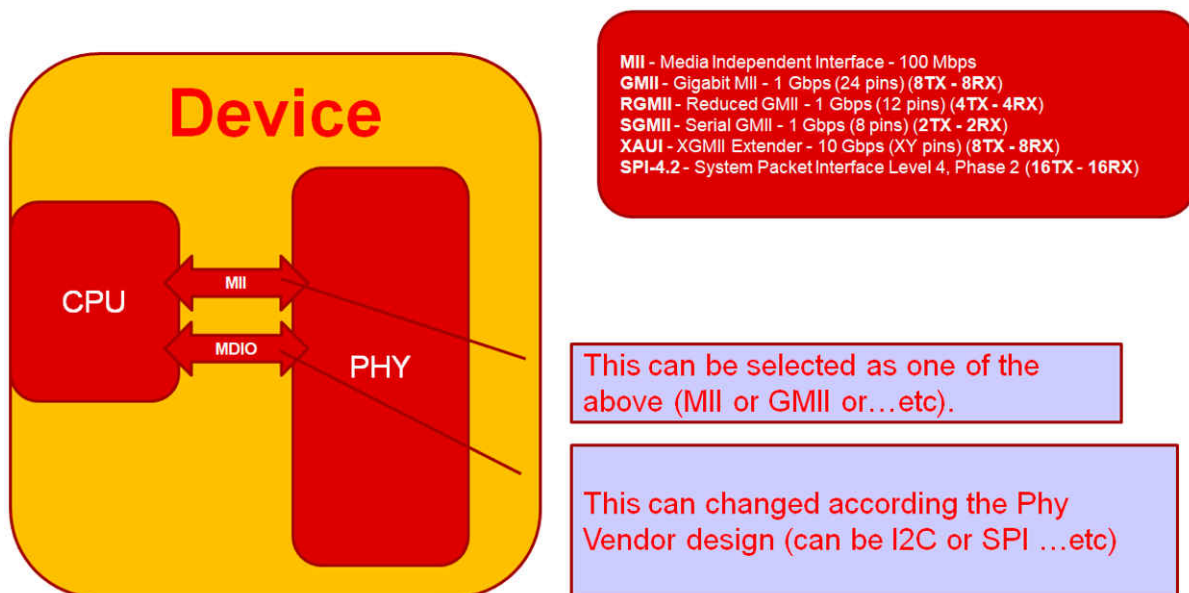


图 2-1. Sitara MDIO/MII 架构示例

图 2-2 举例说明了 MII 可由其他以太网 Phy 规格 ( 例如, RMII、RGMII、SGMII 等 ) 取代。MDIO 控制接口可由 I<sup>2</sup>C/SPI 取代。

在使用 TI Sitara 平台匹配 TI Phy 的同时, 我们将继续使用 MDIO。图 2-2 中的 EMAC 表示以太网 MAC 层。

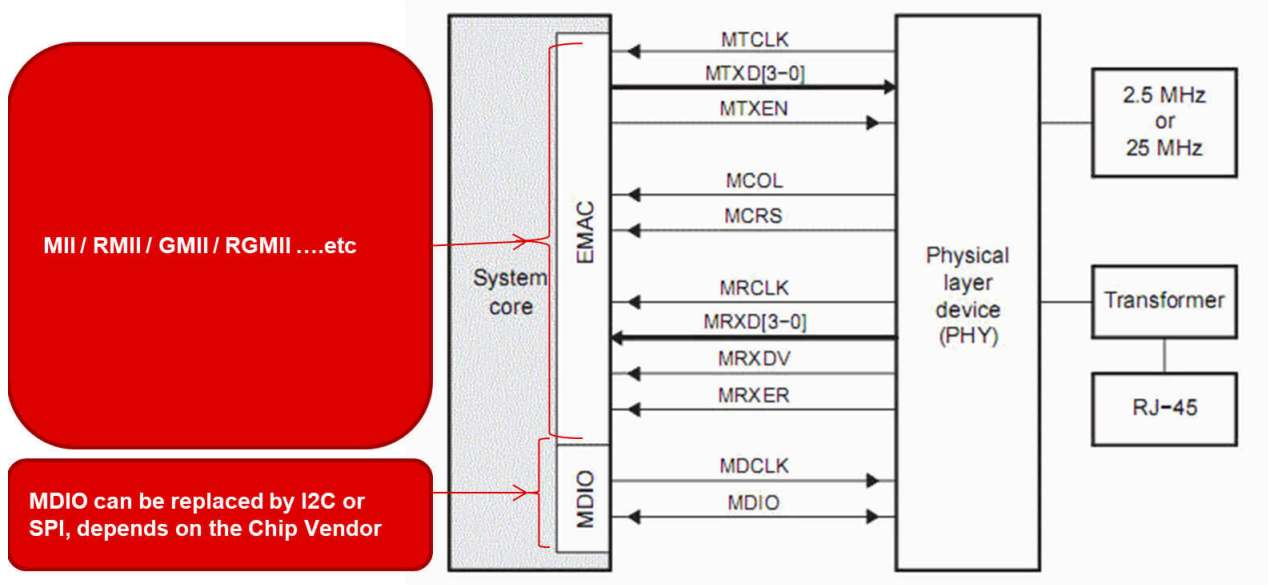


图 2-2. Sitara MDIO/II 详细架构

### 3 将 DP83825 合并到 Sitara 上的步骤

更多有关合并这两款器件的信息，请参阅图 3-1 中显示的合并概念。

BBB 使用 MII 和 Microchip Phy LAN8710A，我们需要将硬件和软件设置更改为 RMII 以匹配 TI Phy DP83825。

1. DP83825 与 DP83822 的区别在于，DP83822 是 RGMII + RMII + MII，而 DP83825 只有 RMII。
2. 检查 AM335 BeagleBoneBlack (BBB) 上的 Phy。
3. 从 BBB 上取出 MicroChip LAN8710A。
4. 确定 BBB LAN8710A 连接。
5. 确定 DP83825 连接。
6. 使用 TI Pinmux 工具。
7. 生成用于 DP83825 引脚分配的 DTSI 文件。
8. 为 BBB 构建 uboot/内核。
9. 使用 BBB+DP83825 进行 Ping 测试。

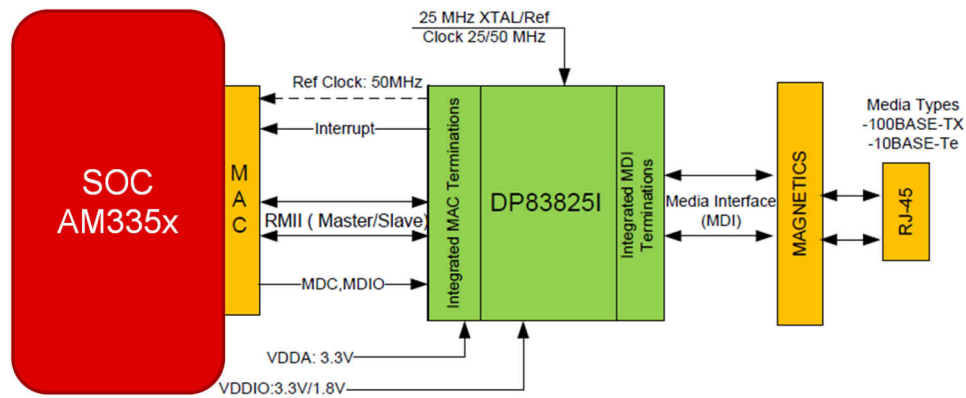


图 3-1. 合并概念

### 3.1 第 1 步 - DP83825 与 DP83822

使用 DP83822 驱动程序并对其进行修改以支持 DP83825，并需要确认这两个器件之间的差异。

图 3-2 所示为 TI 以太网 Phy 之间的差异。DP83822 是 MII/RMII/RGMII。DP83825 仅仅是 RMII。

图 3-2 上的蓝色箭头表示以太网 10/100 接口有两个选项：MII/RMII。

我们选择了 RMII，因为 DP83825 只有 RMII。

**TI 100M Ethernet Comparison**

Device	DP83825 (Target)	DP83822
MAC Interface	RMII Master and Slave	MII, RMII, RGMII
Package	3mmx3mm, 24 QFN	5mmx5mm, 32 QFN
Power Dissipation	130 mWatt	<220mW
Standby Power	< 1 mWatt	
Power Supply	3.3V Single Supply	3.3/1.8V
IO Supply	3.3/1.8	3.3/2.5/1.8
EEE	Yes	Yes
WoL	Yes	Yes
Fiber	No	Yes
Integrated Termination	Yes	Yes (MAC Side Only)
EMI/EMC Performance	Grade A	Grade A
ESD	5 KV HBM	16 kV HBM
Operating Temperature Range	-40 to 85C	-40 to 125C

图 3-2. TI 以太网 Phy 比较

### 3.2 第 2 步 - 检查 8710A 引脚与 AM335

Microchip LAN8710A 支持 MII/RMII 接口。为了连接到 Sitara AM335，如图 3-3 所示，LAN8710A 使用 MII 接口连接 AM335。若要合并 TI DP83825，请正确选择 RMII 引脚，并使用 RMII 配置来更改软件部分。

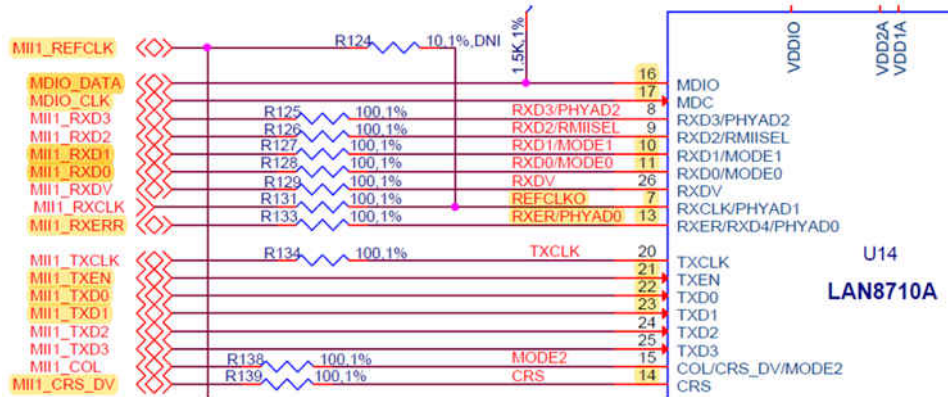


图 3-3. LAN8710A MII 引脚与 AM335

### 3.3 第 3 步 - 从 AM335 EVM - BBB 上取出 LAN8710A

图 3-4 中的橙色圆圈区域为 LAN8710A。请将它取出。

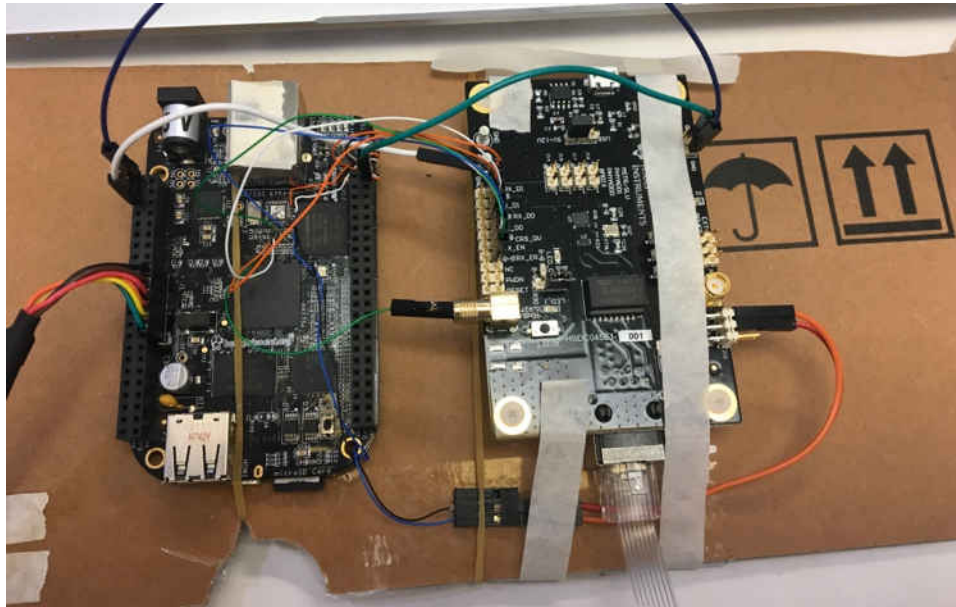


图 3-4. 取出 LAN8710A

### 3.4 第 4 步 - 检查 AM335 (BBB) RMII 连接引脚

图 3-5 中用黄色突出显示的内容是 RMII 所需的引脚。

SubArctic AM335X 15mm x 15mm Package	Pin	Signal	Pin	Signal	Pin	Signal
GMI11_TXCLK/UART2_RXD/RGMI11_TCLK/MMC0_DAT7/MMC1_DAT0/UART1_DCDN/MCASP0_ACLKX/GPIO3_9	K18	MI11_TXCLK	9			
GMI11_TXD0/RMII1_TXD0/RGMI11_TD0/MCASP1_AXR2/MCASP1_ACLKR/EQEP0B_IN/MMC1_CLK/GPIO0_28	K17	MI11_TXD0	9			
GMI11_TXD1/RMII1_TXD1/RGMI11_TD1/MCASP1_FSR/MCASP1_AXR1/EQEP0A_IN/MMC1_CMD/GPIO0_21	K16	MI11_TXD1	9			
GMI11_TXD2/DCA0_RX/RGMI11_TD2/UART4_TXD/MCASP1_AXR0/MMC2_DAT2/MCASP0_AHCLKX/GPIO0_17	K15	MI11_TXD2	9			
GMI11_TXD3/DCA0_TX/RGMI11_TD3/UART4_RXD/MCASP1_FSX/MMC2_DAT1/MCASP0_FSR/GPIO0_16	J18	MI11_TXD3	9			
GMI11_TXEN/RMII1_TXEN/RGMI11_TCTL/TIMER4/MCASP1_AXR0/EQEP0_INDEX/MMC2_CMD/GPIO3_3	J16	MI11_TXEN	9			
GMI11_CRS/RMII1_CRS_DV/SP1_D0/I2C1_SDA/MCASP1_ACLKX/UART5_CTSN/UART2_RXD/GPIO3_1	H17	MI11_CRS_DV	9			
GMI11_COL/RMII2_REFCLK/SP1_SCLK/UART5_RXD/MCASP1_AXR2/MMC2_DAT3/MCASP0_AXR2/GPIO3_0	H16	MI11_COL	9			
GMI11_RXCLK/UART2_TXD/RGMI11_RCLK/MMC0_DAT6/MMC1_DAT1/UART1_DSRN/MCASP0_FSX/GPIO3_10	L18	MI11_RXCLK	9			
GMI11_RXD0/RMII1_RXD0/RGMI11_RD0/MCASP1_AHCLKX/MCASP1_ACLKR/MCASP0_AXR3/GPIO2_21	M18	MI11_RXD0	9			
GMI11_RXD1/RMII1_RXD1/RGMI11_RD1/MCASP1_AXR3/MCASP1_FSR/EQEP0_STROBE/MMC2_CLK/GPIO2_20	L15	MI11_RXD1	9			
GMI11_RXD2/UART3_TXD/RGMI11_RD2/MMC0_DAT4/MMC1_DAT3/UART1_RIN/MCASP0_AXR1/GPIO2_19	L16	MI11_RXD2	9			
GMI11_RXD3/UART3_RXD/RGMI11_RD3/MMC0_DAT5/MMC1_DAT2/UART1_DTRN/MCASP0_AXR0/GPIO2_18	L17	MI11_RXD3	9			
GMI11_RXERR/RMII1_RXERR/SP1_D1/I2C1_SCL/MCASP1_FSX/UART5_RTSN/UART2_TXD/GPIO3_2	J15	MI11_RXERR	9			
GMI11_RXDV/LCD_MEMORY_CLK/RGMI11_RCTL/UART5_TXD/MCASP1_ACLKX/MMC2_DAT0/MCASP0_ACLKR/GPIO3_4	J17	MI11_RXDV	9			
RMII1_REFCLKXDMA_EVENT_INTR2/SP1_CS0/UART5_TXD/MCASP1_AXR3/MMC0_POW/MCASP1_AHCLKX/GPIO0_29	H18	MI11_REFCLK	9	U5 H18	R28	33
MDIO_CLK/TIMER5/UART5_TXD/UART3_RTSN/MMC0_SDWP/MMC1_CLK/MMC2_CLK/GPIO0_1	M18	MDIO_CLK	9			
MDIO_DATA/TIMER6/UART5_RXD/UART3_CTSN/MMC0_SDCD/MMC1_CMD/MMC2_CMD/GPIO0_0	M17	MDIO_DATA	9			

图 3-5. AM335 RMII 选择

### 3.5 第 5 步 - 检查 DP83825 连接引脚

图 3-6 所示为使用 RMII 接口时 DP83825 所需的引脚。所需的引脚以黄色突出显示。接下来，将这些引脚连接到 AM335 BBB EVM 上。

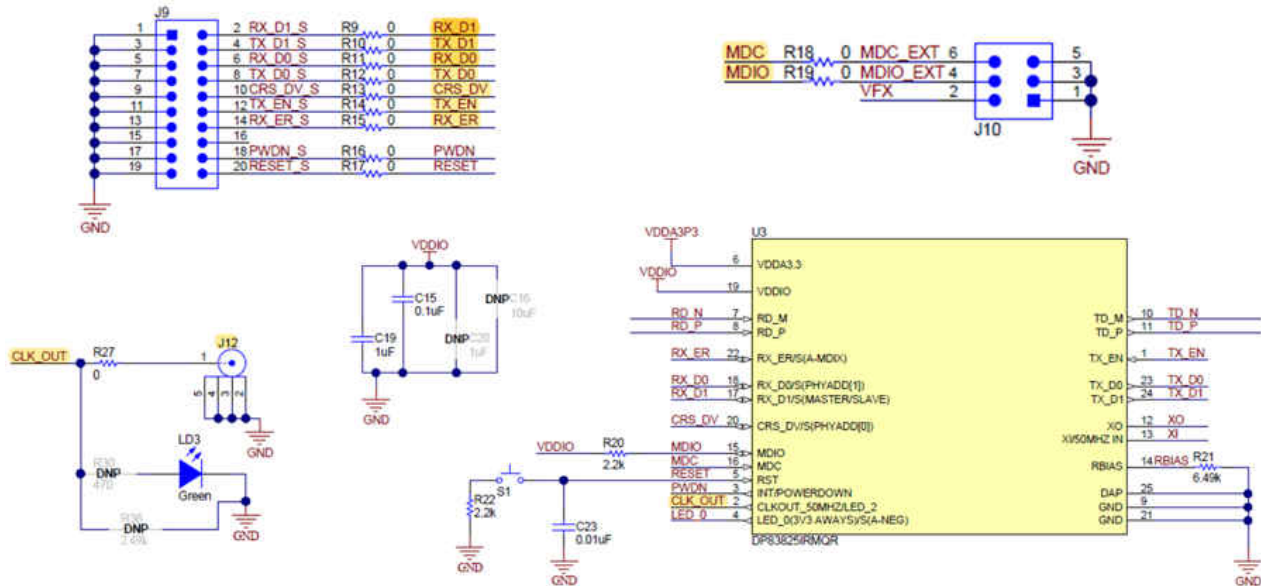


图 3-6. DP83825 所需的引脚

图 3-7 用于 AM335 BeagleBoard 和 DP83825 EVM 之间的连接，有两个关键点需要考虑：

1. 将 5V 从 AM335 EVM 发送到 DP83825 EVM。
2. 将 50Mhz 从 DP83825 EVM 发送到 AM335 EVM。
3. 50Mhz 蓝色有线焊件最好使用 1 根电线并尽可能将其屏蔽。

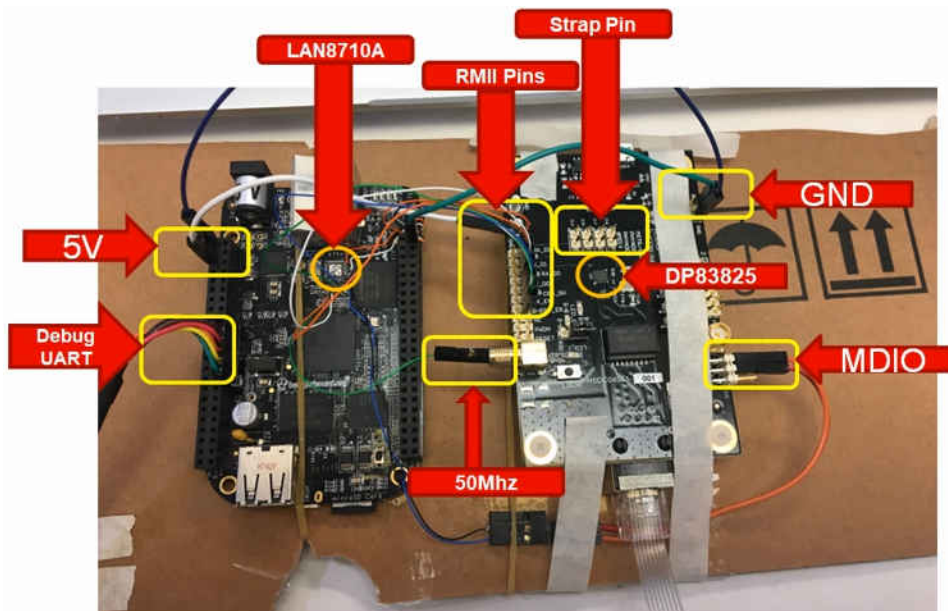


图 3-7. AM335 + DP83825 连接

### 3.6 第 6 步 - 使用 PinMux 工具生成 DTS/DTSI 文件。

本部分包含用户生成 AM335 RMII PINMUX 代码的信息，如图 3-8 所示。



图 3-8. TI PinMux 工具 (云版本)

1. 打开 TI PinMux 工具 (云版本)。
2. 在“Device” (器件) 中选择 AM335x，且“Part” (部件) 设置为 Default (默认)。选择 Start (开始)。
3. 直接选择 Start (开始)，如图 3-8 所示。
4. 请参阅图 3-9，了解详情。

绿色箭头是 RMII 接口，点击 + 按钮。窗口中显示的是 RMII，需要切换到 RMII “1”，Csr\_Drv 使用 H17 引脚，如图 3-9 (红色箭头) 所示。

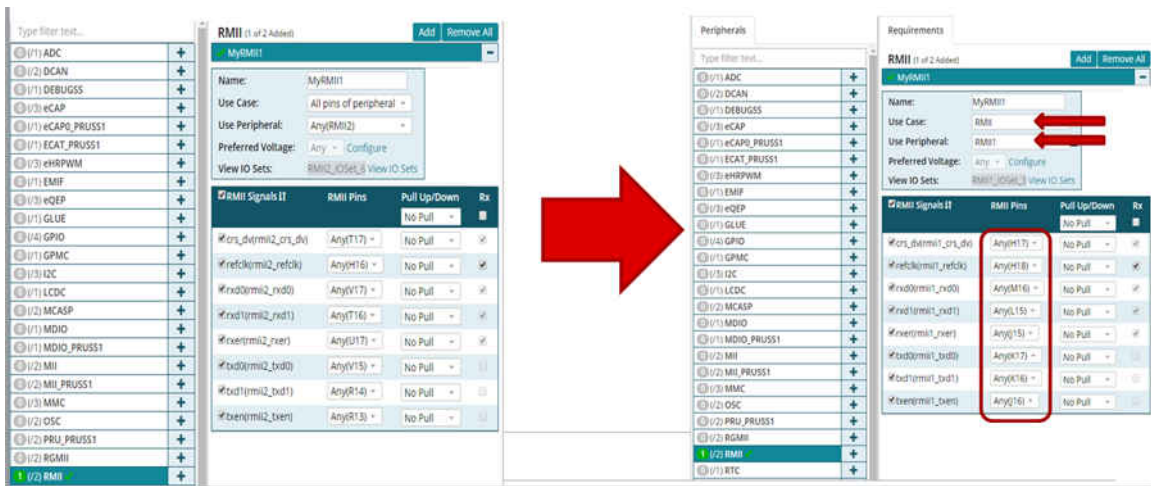


图 3-9. PinMux 工具中用于 DP83825 的 RMII 1



### 3.7 第 7 步 - DP83822 代码库审查

我们现在为 DP83825 创建 Linux Phy 软件驱动程序。DP83825 Phy 驱动程序需要以 DP83822 Phy 驱动程序作为基础，我们将在此基础上修改 DP83825 补丁。

Phy DP82822 初始驱动程序提交内容如图 3-10 所示。

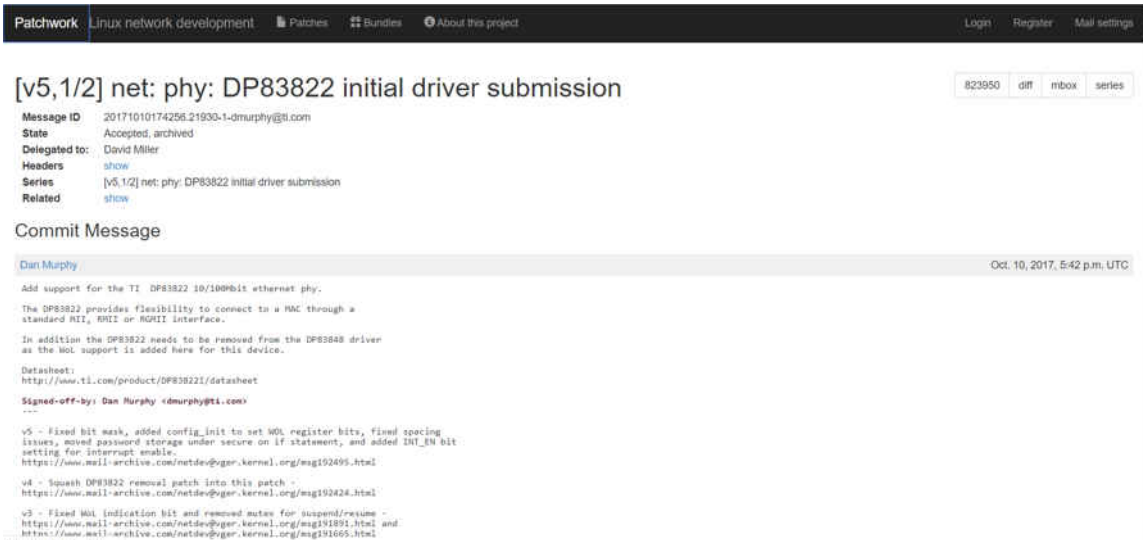


图 3-10. DP83822 Linux Phy 驱动程序

### 3.8 第 8 步 - DP83825 代码库审查/补丁修改

此步骤可在 DP83822 基础上轻松添加 DP83825 Phy 驱动程序。可通过 [LKML.org](http://LKML.org) 链接下载 DP83825 补丁。然后，给以下文件打补丁：/drivers/net/phy/dp83822.c，如图 3-11 所示

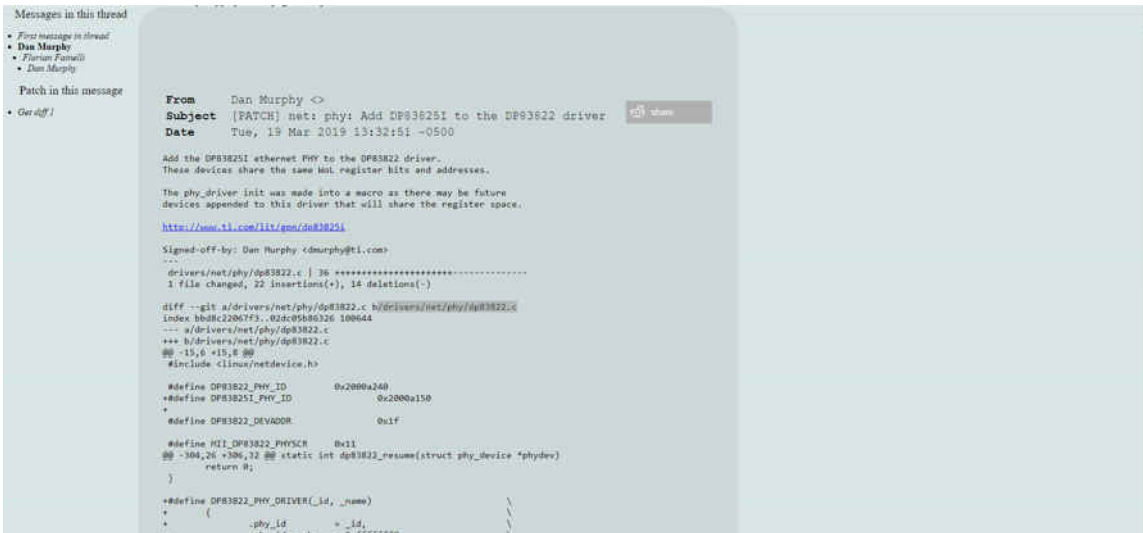


图 3-11. DP83825 Linux Phy 驱动程序补丁

### 3.9 第 9 步 - 修补 Linux uboot/内核

将 PinMux 工具生成的器件树移植到此路径/此文件：

- Uboot 路径：  
dtsi: /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/u-boot-2018.01+gitAUTOINC+313dcd69c2-g313dcd69c2/arch/arm/dts# gedit am335x-bone-common.dtsi
- 内核路径：  
dtsi: /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e/arch/arm/boot/dts# gedit am335x-bone-common.dtsi  
根据 Pinmux 工具生成的文件来修补器件树。

### 3.10 第 10 步 - 更改菜单配置

若要创建菜单配置命令，请执行以下操作：转到以下路径：

/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e#

发出以下命令：

make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- menuconfig

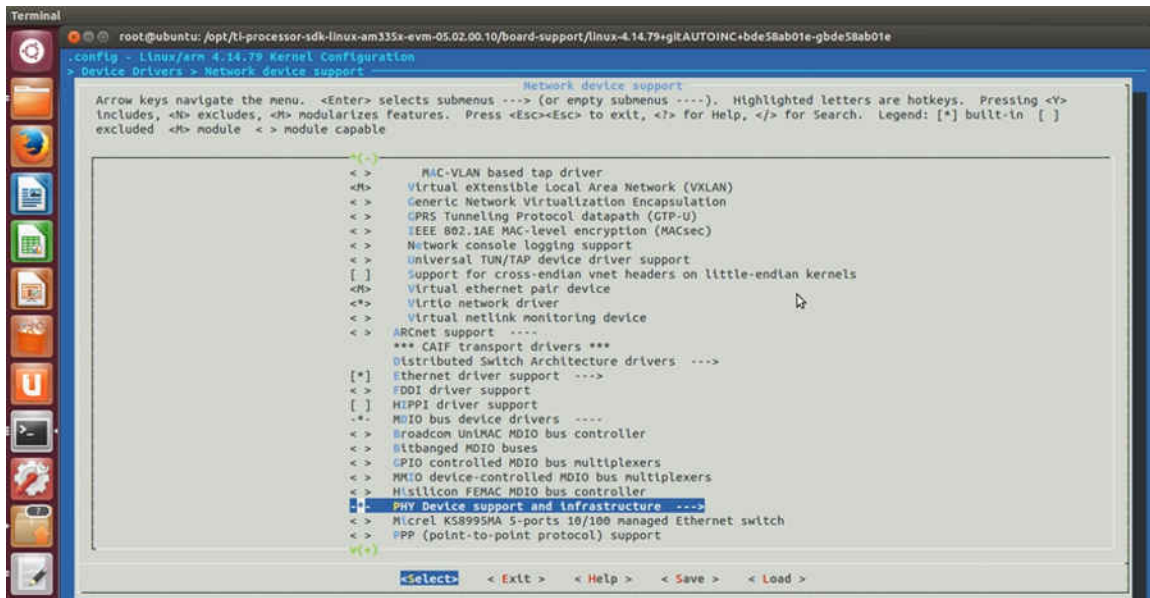


图 3-12. 菜单配置步骤 A

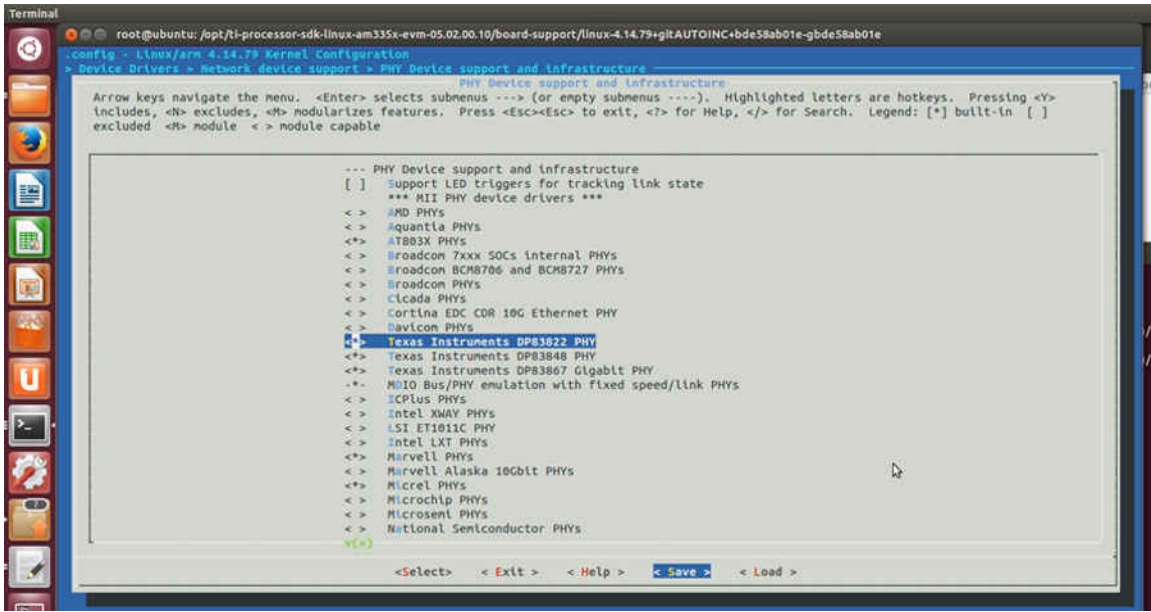


图 3-13. 菜单配置步骤 B

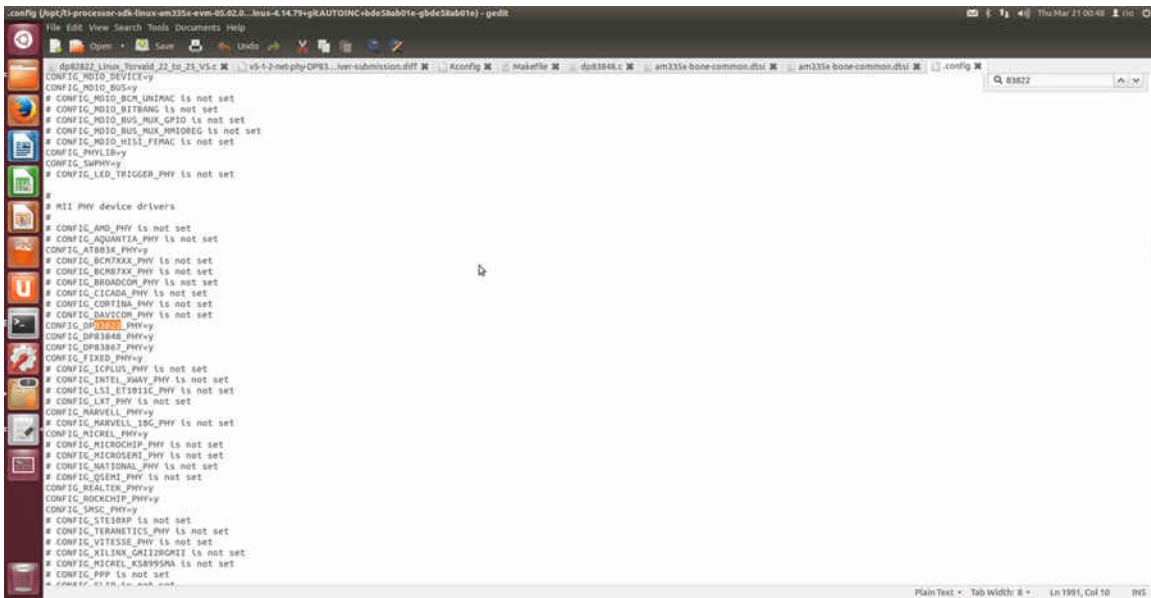


图 3-14. 菜单配置步骤 C

1. 图 3-12 用于设置 Phy 接口。
2. 图 3-13 用于设置 TI Phy。
3. 图 3-14 用于设置 DP83822 Phy。

我们以 DP83822 Phy 驱动程序作为基础，并在 MenuConfig 中设置 DP83822。我们修补 ( 修改 ) 了 DP83922 驱动程序，使其能够用作 DP83825。

### 3.11 第 11 步 - 构建元件 (menuconfig/dtb/zimage)

为了使 DP83825 Phy 适用于 Linux，需要依次构建 menuconfig、dtb 和 zimage 命令。请查看用红色突出显示的内容以注意目标。以下命令使用 SDK 5.02.00.10 作为基础。

- `Make -C /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e ARCH=arm CROSS_COMPILE=/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabi-hf-tisd_k_am335x-evm_defconfig`
  - `Make -C /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e ARCH=arm CROSS_COMPILE=/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabi-hf-menuconfig`
  - `Make -j 8 -C /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e ARCH=arm CROSS_COMPILE=/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabi-hf-zImage`
- `Make -j 8 -C /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e ARCH=arm CROSS_COMPILE=/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabi-hf-am335x-boneblack.dtb`

### 3.12 第 12 步 - ( 可选 ) 用于 DTS 构建的重要修复

- 如果看到以下 DTS 构建警告： - port/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e ARCH=arm CROSS\_COMPILE=/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/linux-devkit/sysroots/x86\_64-arago-linux/usr/bin/arm-linux-gnueabi-hf-am335x-boneblack.dtb make: Entering directory `/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e' CHK scripts/mod/devicetable-offsets.h DTC arch/arm/boot/dts/am335x-boneblack.dtb arch/arm/boot/dts/am335x-boneblack.dtb: Warning (phys\_property): Missing property '#phy-cells' in node /ocp/usb@47400000/usb-phy@47401300 or bad phandle (referred from /ocp/usb@47400000/usb@47401000:phys[0]) arch/arm/boot/dts/am335x-boneblack.dtb: Warning (phys\_property): Missing property '#phy-cells' in node /ocp/usb@47400000/usb-phy@47401b00 or bad phandle (referred from /ocp/usb@47400000/usb@47401800:phys[0]) make: Leaving directory `/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e'
- 请合并此补丁： - <https://patchwork.kernel.org/patch/10052007/>

### 3.13 第 13 步 - 将构建文件复制到 SD 卡上

请注意，我的 SD 卡位于此路径： - /media/rootfs/boot/

用户可能需要更改正确的复制路径。

- 用这个来复制 dtb：
  - `sudo cp /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e/arch/arm/boot/dts/am335x-boneblack.dtb /media/rootfs/boot/`
- 建议不要用这个来复制 dtb：
  - `sudo cp -r /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e/arch/arm/boot/dts/ /media/rootfs/boot/`
- 用这个来复制 zimage：
  - `sudo cp /opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e/arch/arm/boot/zImage /media/rootfs/boot`

### 3.14 第 14 步 - 检查 DP83825 的寄存器。

```

•root@am335x-evm:~# devmem2 0x44E1090c
•/dev/mem opened.
•Memory mapped at address 0xb6fbd000.
•Read at address 0x44E1090C (0xb6fbd90c): 0x00000029
•root@am335x-evm:~# devmem2 0x44E10910
•/dev/mem opened.
•Memory mapped at address 0xb6f39000.
•Read at address 0x44E10910 (0xb6f39910): 0x00000029
•root@am335x-evm:~# devmem2 0x44E10914
•/dev/mem opened.
•Memory mapped at address 0xb6fad000.
•Read at address 0x44E10914 (0xb6fad914): 0x00000009
•root@am335x-evm:~# devmem2 0x44E10928
•/dev/mem opened.
•Memory mapped at address 0xb6f84000.
•Read at address 0x44E10928 (0xb6f84928): 0x00000009
•root@am335x-evm:~# devmem2 0x44E10924
•/dev/mem opened.
•Memory mapped at address 0xb6f3c000.
•Read at address 0x44E10924 (0xb6f3c924): 0x00000009
•root@am335x-evm:~# devmem2 0x44E10940
•/dev/mem opened.
•Memory mapped at address 0xb6fe9000.
•Read at address 0x44E10940 (0xb6fe9940): 0x00000029
•root@am335x-evm:~# devmem2 0x44E1093c
•/dev/mem opened.
•Memory mapped at address 0xb6f76000.
•Read at address 0x44E1093C (0xb6f7693c): 0x00000029
•root@am335x-evm:~# devmem2 0x44E10944
•/dev/mem opened.
•Memory mapped at address 0xb6f81000.
•Read at address 0x44E10944 (0xb6f81944): 0x00000028
•root@am335x-evm:~#

```

### 3.15 第 15 步 - 协助进行调试的 Linux 命令

#### 以太网工具命令调试指南

Ethtool 重要命令：

```

•ethtool -A eth0 autoneg on rx on tx on
用途：此命令将开启 RX/TX + 自动协商。
•ethtool -a eth0
用途：此命令将转储 RX/TX/自动协商 Eth0 (示例) 状态。
•ethtool -S eth0
用途：此命令将转储 Eth0 (示例) 详细状态。
•ethtool eth0
用途：此命令将粗略转储 Eth0 (示例) 状态。
•ethtool -d eth0
用途：此命令将转储 Eth0 寄存器。
•ethtool -i eth0
用途：此命令将转储 Eth0 驱动程序版本。

```

#### Linux ifconfig 命令调试指南

```

•ifconfig
用途：此命令将转储所有可用的网络接口状态。
您可以查看 TX/RX 数据包计数。
•ifconfig eth0 up
用途：此命令将导通 Eth0 (示例) 接口。
•ifconfig eth0 down
用途：此命令将关闭 Eth0 (示例) 接口。

```

### 3.16 第 16 步 - Linux ethtool 命令转储 ( 示例 )

#### [案例 1 : ethtool eth0] , 成功示例

```

root@am335x-evm:~# ethtool eth0
Settings for eth0:
    Supported ports: [ TP AUI BNC MII FIBRE ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
    Speed: 100Mb/s
    Auto-negotiation: on
    Supports Wake-on: d
    Wake-on: d
    Current message level: 0x00000000 (0)
    Link detected: yes

    Link partner advertised link modes:   10baseT/Half 10baseT/Full
                                           100baseT/Half 100baseT/Full
    Link partner advertised pause frame use: Symmetric Receive-only
    Supports auto-negotiation: Yes
    Supported FEC modes: Not reported
    Advertised link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
    Advertised pause frame use: No
    Advertised auto-negotiation: Yes
    Advertised FEC modes: Not reported
    Link partner advertised link modes:   10baseT/Half 10baseT/Full
                                           100baseT/Half 100baseT/Full
    Link partner advertised pause frame use: Symmetric Receive-only
    Link partner advertised auto-negotiation: Yes
    Link partner advertised FEC modes: Not reported
    Speed: 100Mb/s
    Duplex: Full
    Port: MII
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: on
    Supports Wake-on: d
    Wake-on: d
    Current message level: 0x00000000 (0)
    Link detected: yes
  
```

#### [案例 2 : ethtool -i eth0] , 成功

```

root@am335x-evm:~# ethtool -i eth0
driver: cpsw
firmware-version:
expansion-rom-version:
bus-info: 4a100000.ethernet
supports-statistics: yes
supports-test: no
supports-eee: no
supports-eee-access: no
supports-register-dump: yes
supports-priv-flags: no
  
```

#### [案例 3 : ethtool -S eth0] , 成功示例

```

root@am335x-evm:~# ethtool -S eth0
NIC statistics:
    Good Rx Frames: 0
    Broadcast Rx Frames: 0
    Multicast Rx Frames: 0
    Pause Rx Frames: 0
    Rx CRC Errors: 0
    Rx Align/Code Errors: 0
    Oversize Rx Frames: 0
    Rx Jabbers: 0
    Undersize (Short) Rx Frames: 0
    Rx Fragments: 0
    Rx Octets: 0
    Good Tx Frames: 0
    Broadcast Tx Frames: 0
    Multicast Tx Frames: 0
    Pause Tx Frames: 0
    Deferred Tx Frames: 0
    Collisions: 0
    Single Collision Tx Frames: 0
    Multiple Collision Tx Frames: 0
    Excessive Collisions: 0
    Late Collisions: 0
    Tx Underrun: 0
    Carrier Sense Errors: 0
    Tx Octets: 0
  
```

```

Rx + Tx 64 Octet Frames: 0
Rx + Tx 65-127 Octet Frames: 0
Rx + Tx 128-255 Octet Frames: 0
Rx + Tx 256-511 Octet Frames: 0
Rx + Tx 512-1023 Octet Frames: 0
Rx + Tx 1024-Up Octet Frames: 0
Net Octets: 0
Rx Start of Frame Overruns: 0
Rx Middle of Frame Overruns: 0
Rx DMA Overruns: 0
Rx DMA chan 0: head_enqueue: 1
Rx DMA chan 0: tail_enqueue: 127  Rio: RX is working.
Rx DMA chan 0: pad_enqueue: 0
Rx DMA chan 0: misqueued: 0
Rx DMA chan 0: desc_alloc_fail: 0
Rx DMA chan 0: pad_alloc_fail: 0
Rx DMA chan 0: runt_receive_buf: 0
Rx DMA chan 0: runt_transmit_bu: 0
Rx DMA chan 0: empty_dequeue: 0
Rx DMA chan 0: busy_dequeue: 0
Rx DMA chan 0: good_dequeue: 0
Rx DMA chan 0: requeue: 0
Rx DMA chan 0: teardown_dequeue: 0
Tx DMA chan 0: head_enqueue: 33
Tx DMA chan 0: tail_enqueue: 27  Rio: TX is working.
Tx DMA chan 0: pad_enqueue: 0
Tx DMA chan 0: misqueued: 1
Tx DMA chan 0: desc_alloc_fail: 0
Tx DMA chan 0: pad_alloc_fail: 0
Tx DMA chan 0: runt_receive_buf: 0
Tx DMA chan 0: runt_transmit_bu: 5
Tx DMA chan 0: empty_dequeue: 32
Tx DMA chan 0: busy_dequeue: 0
Tx DMA chan 0: good_dequeue: 33
Tx DMA chan 0: requeue: 0
Tx DMA chan 0: teardown_dequeue: 0
    
```

#### [案例 4 : ethtool -a eth0], 成功示例

```

root@am335x-evm:~# ethtool -a eth0
Pause parameters for eth0:
Autonegotiate: off  Rio: we can ignore this.
RX:             on  Rio: If the link is okay, this RX places will be "On".
TX:             on  Rio: If the link is okay, this TX places will be "On".
    
```

#### [案例 5 : ethtool -A eth0 ...], 成功示例

```

root@am335x-evm:~# ethtool -A eth0 autoneg on rx on tx on
rx unmodified, ignoring  Rio: It said: RX is already on.
tx unmodified, ignoring  Rio: It said: TX is already on.
    
```

#### [案例 6 : ethtool -s eth0 ...]

```

ethtool -s eth0 speed 100 autoneg on  Rio: This command will force to use 100M speed.
    
```

### 3.17 第 17 步 - 用 Linux dmesg 检查以太网驱动程序状态。

`dmesg` 可在系统引导进入内核之时获取所有内核日志。输入以下命令来提取网络状态将有助于进行调试。

- `dmesg | grep mdio` 用途：此命令用于了解 `mdio` 状态。
- `dmesg | grep link` 用途：此命令用于了解 `link` 状态。在插入/拔出电缆时，可使用此命令来了解详细信息。
- `dmesg | grep cpsw` 用途：此命令将用于了解 `cpsw` 状态。更多有关 `CPSW` 的信息，请访问：[Sitara Linux 双 Emac 模式](#)
- `dmesg | grep net` 用途：此命令用于了解网络 (`IPV4/IPV6` 状态)。

### 3.18 第 18 步 - 包含详细日志分析的测试结果 (成功情况)。

内核的成功情况转储如下。此处按不同的部分列出。当 `DP83825` 驱动程序已完成初始化后，您将拥有这样的 `dmesg` (又名：内核日志)

```
[ 1.093845] davinci_mdio 4a101000.mdio: phy[0]: device 4a101000.mdio:00, driver TI DP83825
[ 19.692992] TI DP83825 4a101000.mdio:00: attached PHY driver [TI DP83825]
(mii_bus:phy_addr=4a101000.mdio:00, irq=POLL)
When issuing "dmesg | cpsw", log will like this:
root@am335x-evm:~# dmesg | grep "cpsw"
[ 1.310922] cpsw 4a100000.ethernet: Detected MACID = 50:8c:b1:0f:7e:c8 □Rio: it's MAC ID.
[ 1.317580] cpsw 4a100000.ethernet: initialized cpsw ale version 1.4
[ 1.324016] cpsw 4a100000.ethernet: ALE Table size 1024
[ 1.329300] cpsw 4a100000.ethernet: cpts: overflow check period 500 (jiffies)
[ 24.986407] net eth0: initializing cpsw version 1.12 (0)
[ 26.081848] cpsw 4a100000.ethernet eth0: Link is Up - 10/100Mbps/Full - flow control rx/tx □Rio:
Link is up.
root@am335x-evm:~# dmesg | grep "link"
[ 20.815722] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 21.850443] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready □Rio: This will lead success.
[ 27.484687] IPv6: ADDRCONF(NETDEV_UP): usb0: link is not ready
root@am335x-evm:~# dmesg | grep "cpsw"
[ 1.321037] cpsw 4a100000.ethernet: Detected MACID = 50:8c:b1:0f:7e:c8
[ 1.327701] cpsw 4a100000.ethernet: initialized cpsw ale version 1.4
[ 1.334114] cpsw 4a100000.ethernet: ALE Table size 1024
[ 1.339411] cpsw 4a100000.ethernet: cpts: overflow check period 500 (jiffies)
[ 19.231743] net eth0: initializing cpsw version 1.12 (0)
[ 21.842513] cpsw 4a100000.ethernet eth0: Link is Up - 10/100Mbps/Full - flow control off
[ 53.402750] cpsw 4a100000.ethernet eth0: Link is Up - 10/100Mbps/Full - flow control off
root@am335x-evm:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 50:8C:B1:0F:7E:C8
          inet addr:192.168.1.102  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::528c:b1ff:fe0f:7ec8%3068425624/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:238 (238.0 B)  TX bytes:10806 (10.5 KiB) □ Rio: When linked, it will have RX/TX
packets
          Interrupt:45
Ping result is okay.
(Because we are using the blue wire soldering, so, the signal quality is not good.
You can see the "sequence number" is not continued.
The base case is to have the real PCB that mounted with the DP83825, not blue-wired.)
root@am335x-evm:~# ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101): 56 data bytes
64 bytes from 192.168.1.101: seq=6 ttl=128 time=113.975 ms
64 bytes from 192.168.1.101: seq=10 ttl=128 time=2.090 ms
64 bytes from 192.168.1.101: seq=13 ttl=128 time=1.708 ms
64 bytes from 192.168.1.101: seq=20 ttl=128 time=2.120 ms
64 bytes from 192.168.1.101: seq=22 ttl=128 time=2.055 ms
64 bytes from 192.168.1.101: seq=24 ttl=128 time=1.752 ms
64 bytes from 192.168.1.101: seq=31 ttl=128 time=2.167 ms
64 bytes from 192.168.1.101: seq=32 ttl=128 time=2.106 ms
64 bytes from 192.168.1.101: seq=34 ttl=128 time=1.910 ms
64 bytes from 192.168.1.101: seq=35 ttl=128 time=2.101 ms
```



### 3.19 第 19 步 - YouTube 演示视频

在 YouTube 应用程序中，搜索 *TI Rio Ethernet*。

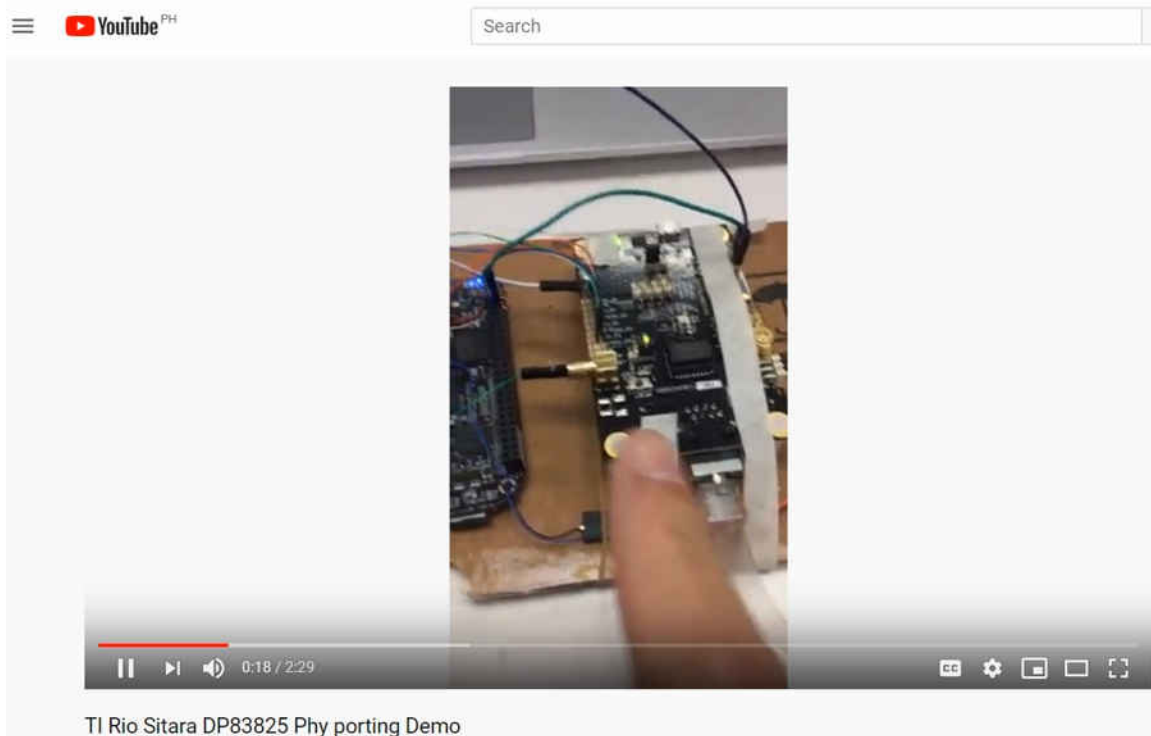


图 3-15. YouTube 演示视频

## 4 需要的硬件和软件

### 4.1 硬件

- [TI AM335 BeagleBoneBlask 版本 C](#)
- [TI DP83825 EVM](#)

### 4.2 软件

- [Pinmux 工具](#)
- [TI AM335 Linux SDK 版本 5.02](#)
- [DP83822 补丁](#)
- [DP83825 补丁](#)

## 5 参考文献

- 德州仪器 (TI), 《DP83822 EVM 用户指南》
- 德州仪器 (TI), 《DP83822 低功耗耐用型 10/100Mbps 以太网物理层收发器》数据表
- 德州仪器 (TI), 《TI DP82825 EVM 用户指南》
- 《DP83825 低功耗 10/100Mbps 以太网物理层收发器》数据表
- TI AM335 BeagleBoard EVM 原理图
- 德州仪器 (TI), 《ICSS EMAC LLD 开发者指南》
- 德州仪器 (TI), 《Processor SDK Linux 软件开发者指南》
- 德州仪器 (TI), PRU-ICSS 以太网
- 德州仪器 (TI), Sitara Linux 双 Emac 模式
- 德州仪器 (TI), 《TI Sitara™ 处理器上的 EtherNet/IP™》白皮书
- 德州仪器 (TI), AM-Class 器件上的以太网系统硬件
- 德州仪器 (TI), 《TMS320C6472/TMS320TCI6486 EMAC 实现指南》应用报告 (第 8 页)
- Microchip, LAN8710A 数据表

## 6 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

### Changes from Revision \* (April 2020) to Revision A (July 2021)

**Page**

- |                                 |   |
|---------------------------------|---|
| • 更新了整个文档中的表格、图和交叉参考的编号格式。..... | 2 |
|---------------------------------|---|

## 重要声明和免责声明

TI 提供技术和可靠性数据 (包括数据表)、设计资源 (包括参考设计)、应用或其他设计建议、网络工具、安全信息和其他资源, 不保证没有瑕疵且不做任何明示或暗示的担保, 包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任: (1) 针对您的应用选择合适的 TI 产品, (2) 设计、验证并测试您的应用, (3) 确保您的应用满足相应标准以及任何其他安全、安保或其他要求。这些资源如有变更, 恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务, TI 对此概不负责。

TI 提供的产品受 TI 的销售条款 (<https://www.ti.com/legal/termsofsale.html>) 或 [ti.com](https://www.ti.com) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

邮寄地址: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2021, 德州仪器 (TI) 公司

## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司