Application Report

How to Merge DP83825 On Linux Sitara AM335

Rio Chan

ABSTRACT

This application note includes information on how to merge the Texas Instruments 10/100 Ethernet Phy DP83825 onto TI Sitara AM335 BeagleBoneBlack.

Table of Contents

List of Figures

Copyright © 2021 Texas Instruments Incorporated

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

DP83825 is a low-cost Ethernet Phy. This document validates that the DP83825 Phy is compatible with the Linux Platform and uses the smallest TI Linux Platform EVM (BeagleBoneBlack) for a base. Follow the steps in this document to use the DP83825 Phy as specified.

2 Device Overview

Figure 2-1, is a Sitara MDIO / MII arch example. In the Figure 2-1, the MDIO is a control interface to control the Phy (example: DP83825), and the MII is an interface for the Data Channel between CPU and Phy.

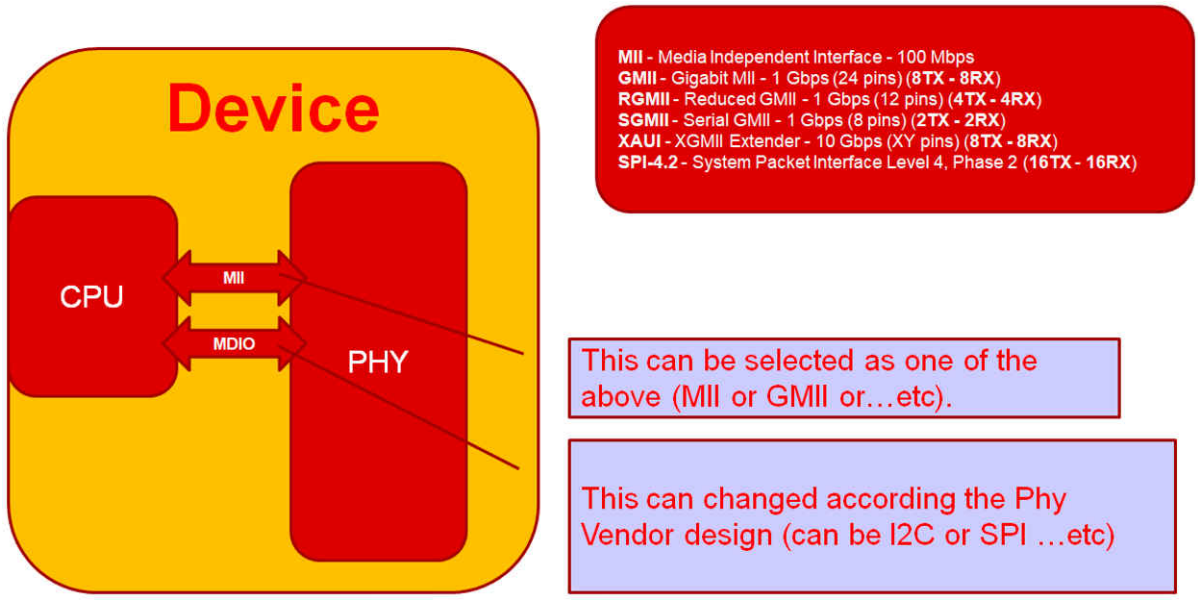

Figure 2-1. Sitara MDIO / MII Arch Example

Copyright © 2021 Texas Instruments Incorporated

Figure 2-2, show an example that the MII can be replaced by the other Ethernet Phy spec, for example (RMII, RGMII, SGMII and so on). MDIO control interface can be replaced by I^2C/SPI.

While using TI Sitara platform to adapt the TI Phy, we are continuing with MDIO. The EMAC in the Figure 2-2 is an Ethernet MAC layer.

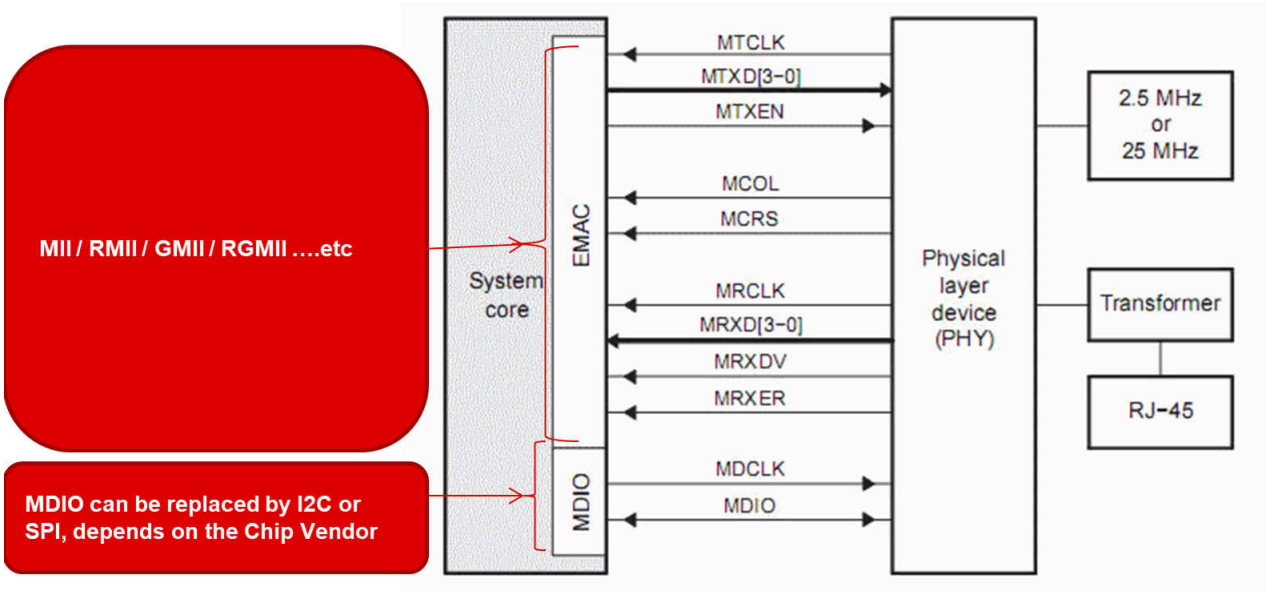

Figure 2-2. Sitara MDIO / MII Detail Arch

Copyright © 2021 Texas Instruments Incorporated

3 Merging DP83825 Onto Sitara Steps

For additional information about merging these two sections, Figure 3-1 shows the merged concept.

The BBB uses MII with Microchip Phy LAN8710A, we need to change the hardware and software setting to RMII for adapting to the TI Phy DP83825.

1. To find the difference between DP83825 vs, DP83822, DP83822 is RGMII + RMII + MII; DP83825 is only having RMII.
2. Check the Phy on AM335 BeagleBoneBlack (BBB).
3. De-populate the MicroChip LAN8710A from BBB.
4. Identify the BBB LAN8710A connection.
5. Identify the DP83825 connection.
6. Use the TI Pinmux tool.
7. Generate the DTSI file for DP83825 pin assignment.
8. Build the uboot / kernel for BBB.
9. Ping test with BBB+DP83825.

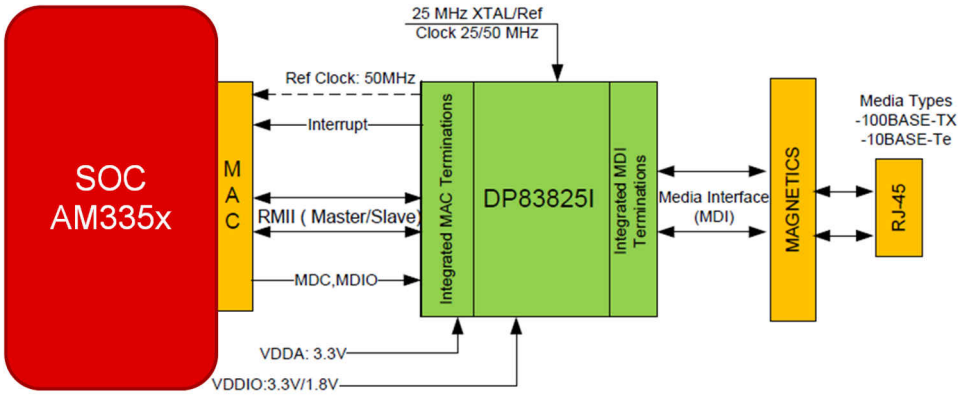

Figure 3-1. Merged Concept

Copyright © 2021 Texas Instruments Incorporated

3.1 Step 1 – DP83825 vs. DP83822

Use the DP83822 driver and modify it to support the DP83825. You need to verify the difference between these two parts.

Figure 3-2 shows the difference between the TI Ethernet Phy. DP83822 is MII/RMII/RGMII. DP83825 is RMII only.

The blue arrows on the Figure 3-2 indicate the Ethernet 10/100 interface has two options: MII / RMII.

We selected the RMII, because the DP83825 has only 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

Figure 3-2. TI Ethernet Phy Comparison

3.2 Step 2 – Check 8710A Pins with AM335

The Microchip LAN8710A is supporting MII/RMII interface. To connect to Sitara AM335, from the Figure 3-3, the LAN8710A is using MII interface to connect the AM335. To merge the TI DP83825, select the RMII PIN correctly, and change the software parts using RMII config.

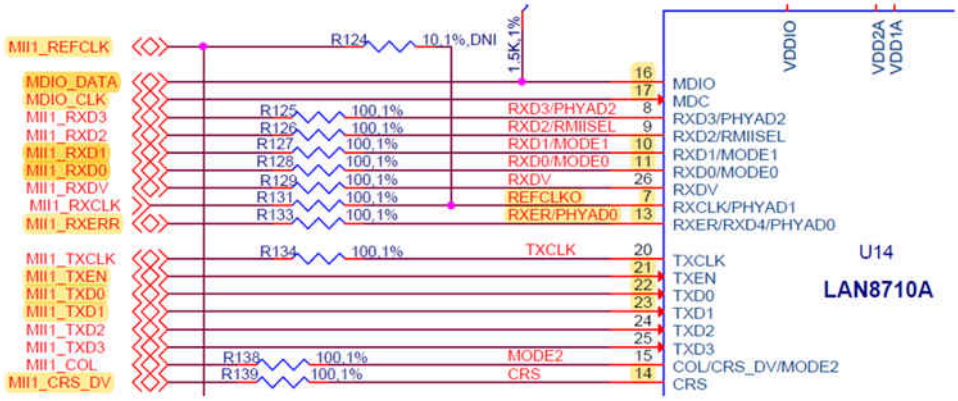

Figure 3-3. LAN8710A MII Pin with AM335

Copyright © 2021 Texas Instruments Incorporated

3.3 Step 3 – De-populate the LAN8710A from AM335 EVM - BBB

The orange circled area in Figure 3-4 is the LAN8710A. Please de-pop it.

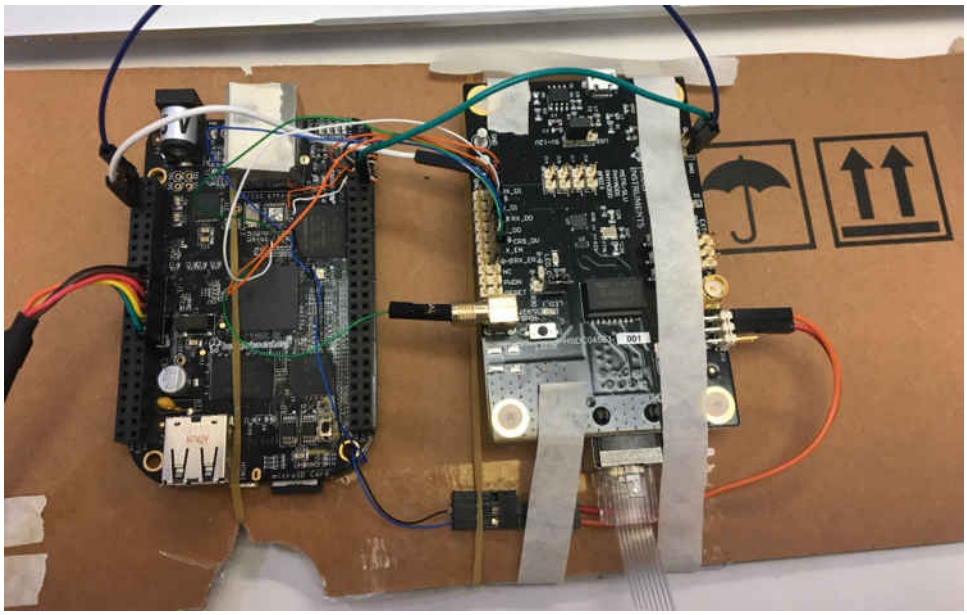

Figure 3-4. De-pop LAN8710A

3.4 Step 4 – Check the AM335 (BBB) RMII Connection Pins

Figure 3-5 yellow highlighted are the RMII required pins.

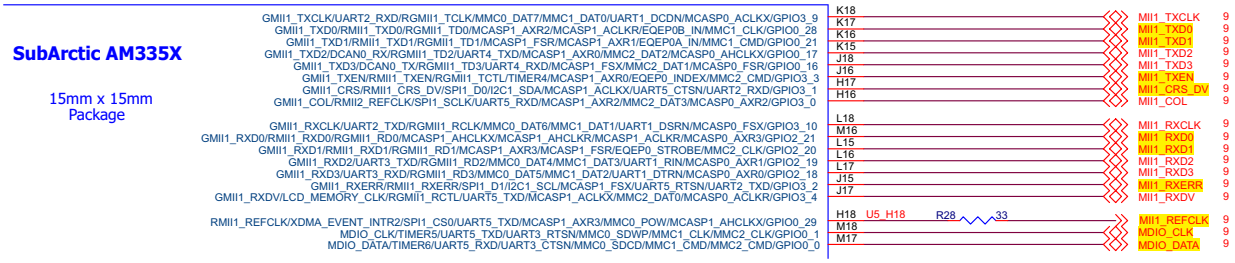

Figure 3-5. AM335 RMII Selection

Copyright © 2021 Texas Instruments Incorporated

3.5 Step 5 – Check the DP83825 Connection Pins

Figure 3-6 is the DP83825 required pin for using RMII interface. We highlighted the required pins in yellow. Next, connect those pins onto AM335 BBB EVM.

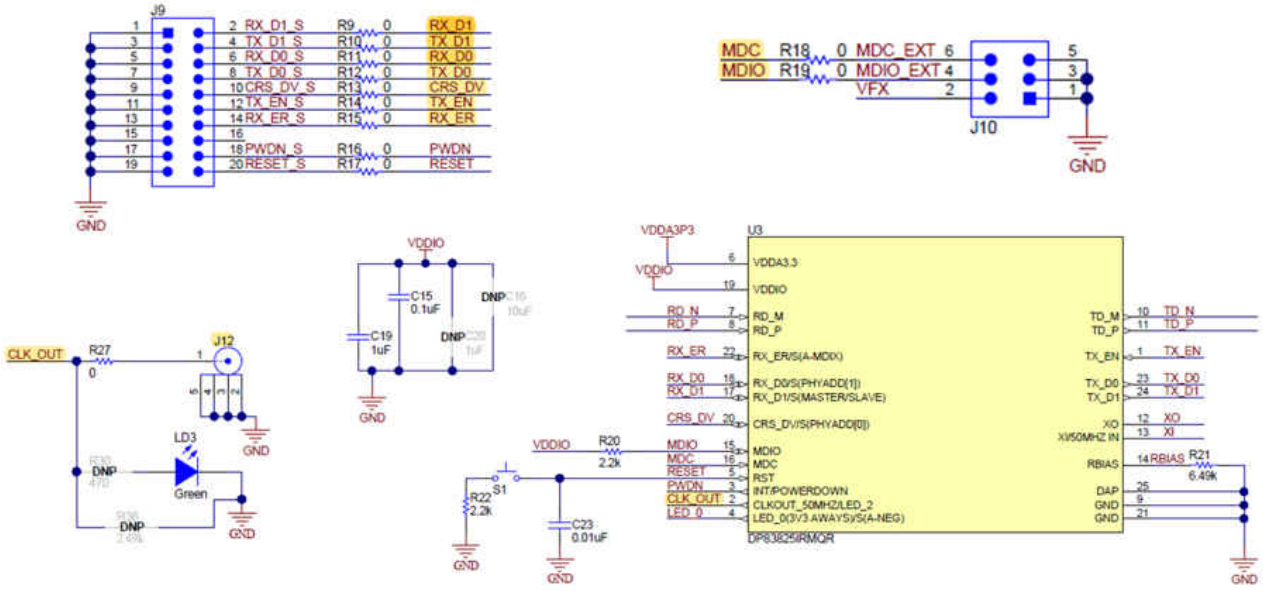

Figure 3-6. DP83835 Required Pins

Figure 3-7 is used for the connections between the AM335 BeagleBoard and DP83825 EVM, there are two key points to consider:

1. Sending the 5 V from the AM335 EVM to the DP83825 EVM.
2. Sending the 50Mhz from DP83825 EVM to the AM335 EVM.
3. The 50Mhz blue wired soldering would be better to use 1 wire and shielding it if possible.

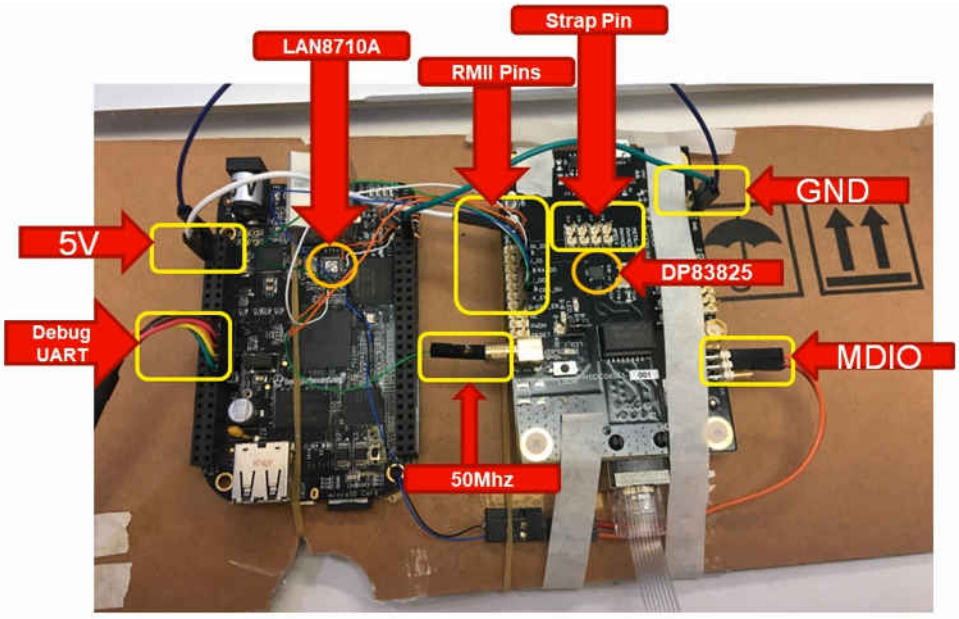

Figure 3-7. AM335 + DP83835 Connections

Copyright © 2021 Texas Instruments Incorporated

3.6 Step 6 – PinMux Tool To Generate DTS/DTSI Files.

This section includes information for the user to generate the AM335 RMII PINMUX codes shown in Figure 3-8.

Figure 3-8. TI PinMux Tool (Cloud Version)

1. Open the TI PinMux tool (Cloud Version).
2. Select the device as **AM335x**, Part is **Default**. Select **Start**.
3. Select **Start** directly as shown in Figure 3-8.
4. See Figure 3-9 for detail.

The Green Arrow is the RMII interface, click the **+** button. The window displays the RMII, you need to switch to the **RMII"1"**, the Csr_Drv is using H17 Pin as shown in Figure 3-9 (Red arrow).

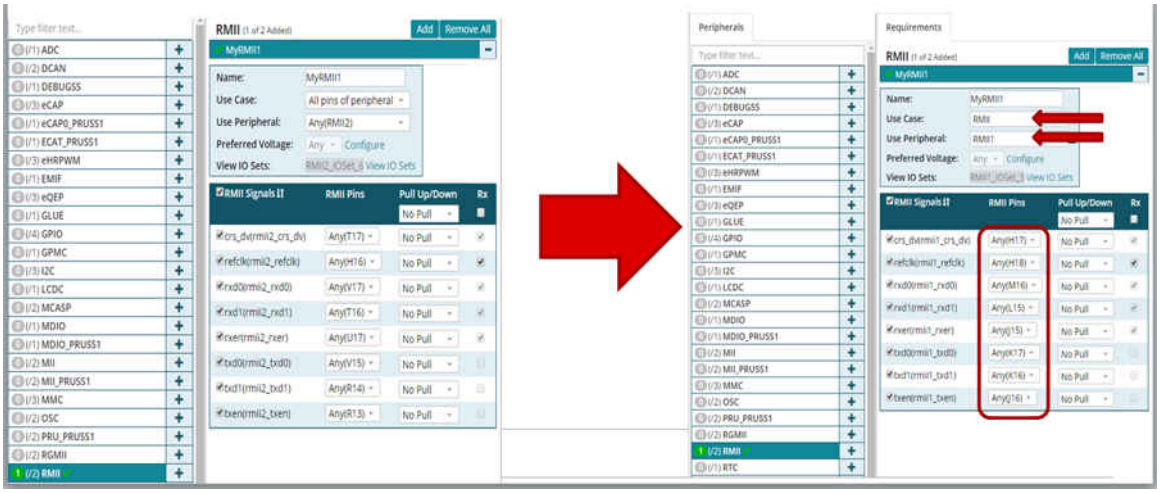

Figure 3-9. RMII *1* for DP83825 by PinMux Tool

Copyright © 2021 Texas Instruments Incorporated

3.7 Step 7 – DP83822 Code Base Review

We now create the SW Linux Phy driver for DP83825. The DP82825 Phy driver needs the DP83822 Phy driver as the base, and we will adapt the DP83825 patch on this base.

The Phy DP82822 initial driver submission is shown in Figure 3-10.

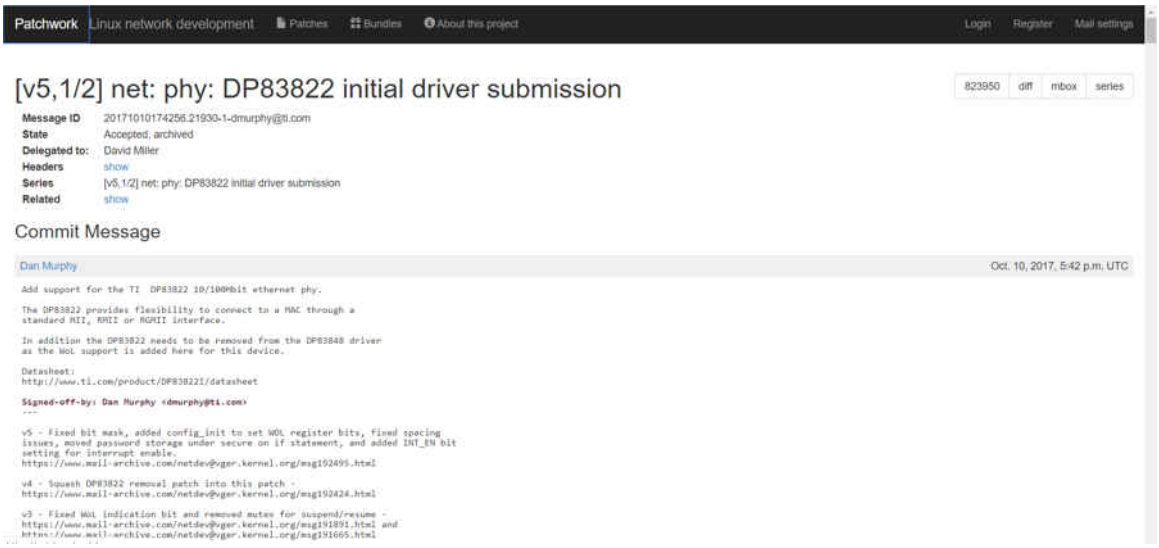

Figure 3-10. DP83822 Linux Phy Driver

3.8 Step 8 – DP83825 Code Base Review / Patch Adaption

This step is easy to add the DP82825 Phy Driver on the DP82822 base. You can download the DP83825 Patch from the LKML.org link. Then patch the following file: /drivers/net/phy/dp83822.c as shown in Figure 3-11

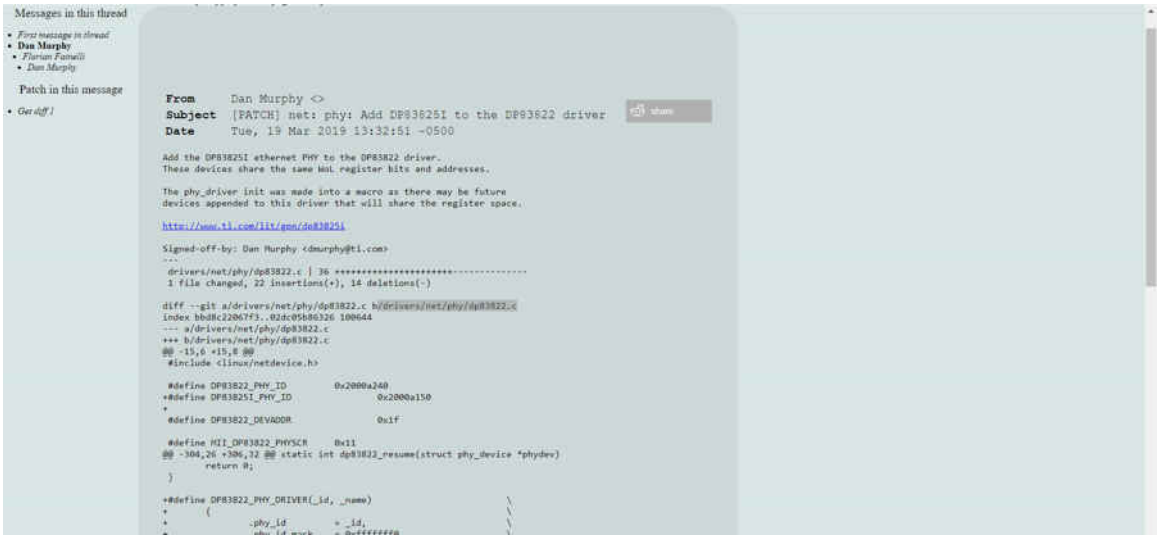

Figure 3-11. DP83825 Linux Phy Driver Patch

Copyright © 2021 Texas Instruments Incorporated

3.9 Step 9 – Patch the Linux uboot/kernel

Porting the PinMux tool Generated Device Tree to this path/this file:

- Uboot path:
  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
- Kernel path:
  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
  Patch the device tree according to the file that the pinmux tool generated.

3.10 Step 10 – Change the Menu Config

To create the menu config command: Go to this path:
/opt/ti-processor-sdk-linux-am335x-evm-05.02.00.10/board-support/linux-4.14.79+gitAUTOINC+bde58ab01e-gbde58ab01e#

Issue this cmd:
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- menuconfig

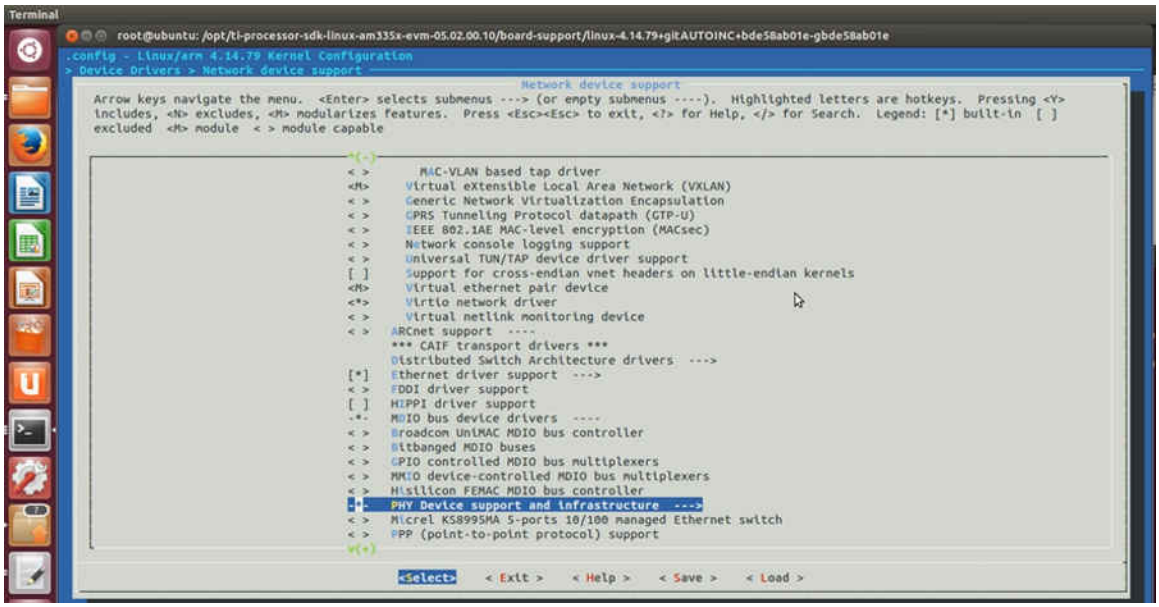

Figure 3-12. Menu Configuration Step A

Copyright © 2021 Texas Instruments Incorporated

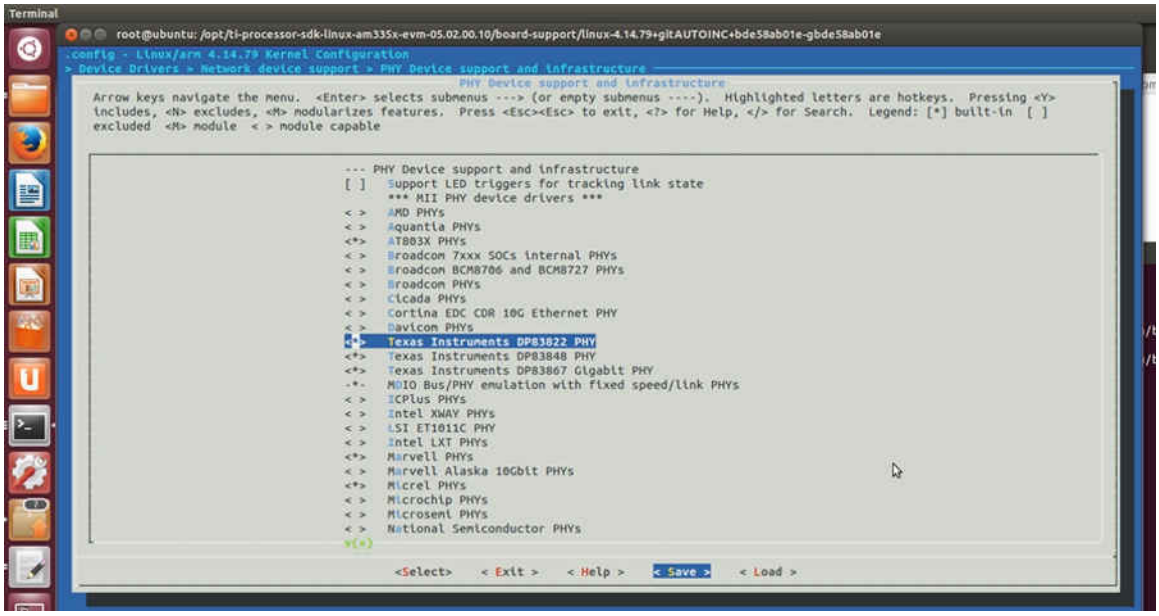

Figure 3-13. Menu Configuration Step B

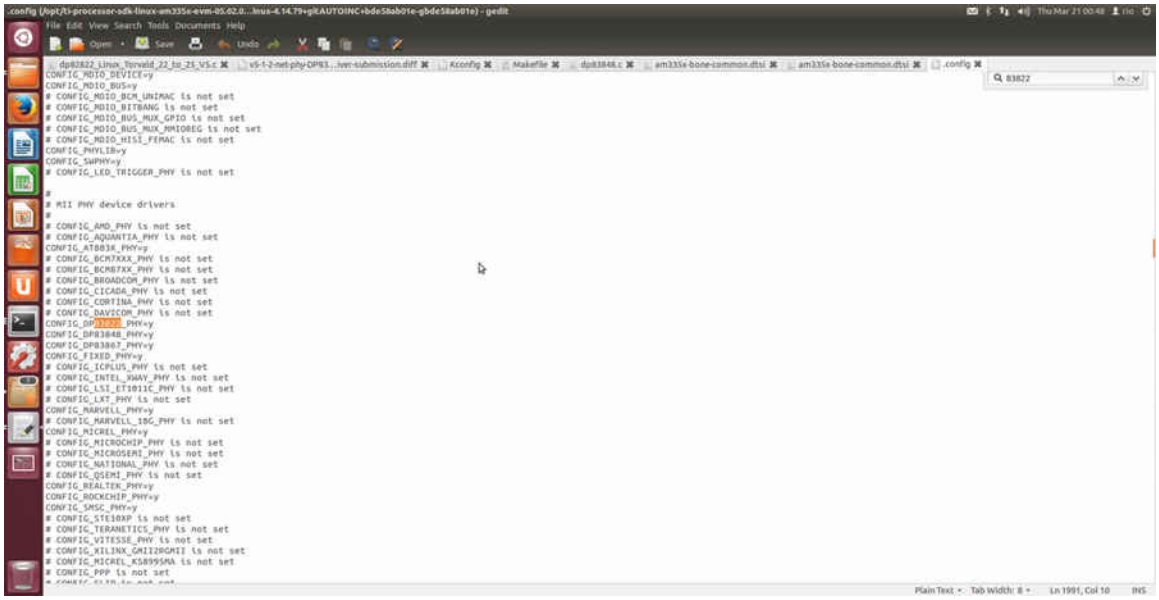

Figure 3-14. Menu Configuration Step C

1. Figure 3-12 is used to set the Phy Interface.
2. Figure 3-13 is used to set the TI Phy.
3. Figure 3-14 is used to set the DP83822 Phy.

We are using the DP83822 Phy driver as the base and we set DP83822 in the MenuConfig. We patched (modified) the DP83922 driver to allow it to act as DP83825.

Copyright © 2021 Texas Instruments Incorporated

3.11 Step 11 – Building the Components (menuconfig/dtb/zimage)

The menuconfig, dtb, and zimage commands are required to be built sequentially to make the DP83825 Phy work with Linux. See the red highlight to notice the goal. These commands are using the SDK 5.02.00.10 as the base.

- 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-gnueabihf- tisdk_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-gnueabihf- 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-gnueabihf- 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-gnueabihf- am335x-boneblack.dtb

3.12 Step 12 – (Optional) Important Fix for DTS Build

- If you have this DTS build warning: – 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-gnueabihf- 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'
- Please merge this patch: – https://patchwork.kernel.org/patch/10052007/

3.13 Step 13 – Copy the Built Files onto the SD Card

Please note, my SD card is on this path: – /media/rootfs/boot/

User might need to change the right path to copy.

- Use this one to copy the 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/
- Not recommend this one to copy the 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/
- Use this one to copy the 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

Copyright © 2021 Texas Instruments Incorporated

3.14 Step 14 – Register Checking for the DP83825.

```
•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 Step 15 – Linux Command To Assist The Debug

Ethernet tool command debug guide

Ethtool important command:

```
•ethtool -A eth0 autoneg on  rx on  tx on
Purpose: This command will turn on the RX/TX + auto-negotiation.
•ethtool -a eth0
Purpose: This command will dump the RX/TX/auto-nego Eth0 (for example) status.
•ethtool -S eth0
Purpose: This command will dump the detail Eth0 (for example) status.
•ethtool eth0
Purpose: This command will dump the Eth0 (for example) status roughly.
•ethtool -d eth0
Purpose: This command will dump the Eth0 Registers.
•ethtool -i eth0
Purpose: This command will dump the Eth0 Driver version.
```

Linux ifconfig command debug guide

```
•ifconfig
Purpose: This command will dump all the available network interface status.
The important is you can see the TX/RX packets counting.
•ifconfig eth0 up
Purpose: This command will turn on the Eth0 (for example) interface.
•ifconfig eth0 down
Purpose: This command will turn off the Eth0 (for example) interface.
```


Copyright © 2021 Texas Instruments Incorporated

3.16 Step 16 – Linux ethtool Command Dumps (example)

[Case1 : ethtool eth0] , success example

```
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□Rio: DP83825 will support 10/100.
        Supported pause frame use: Symmetric Receive-only
        Supports auto-negotiation: Yes□Rio: This is a key point to success.
        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□Rio: This is a key point to success.
        Advertised FEC modes: Not reported
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full □Rio: Remote side support.
        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□Rio: Current link speed.
        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 □Rio: Current link Status, this is a key point to success.
```

[Case2 : ethtool –i eth0] , success

```
root@am335x-evm:~# ethtool -i eth0
driver: cpsw□Rio: Prove that TI CPSW Ethernet driver is okay.
version: 1.0
firmware-version:
expansion-rom-version:
bus-info: 4a100000.ethernet
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

[Case3 : ethtool –S eth0] , success example

```
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□Rio: No Collision detect, good sign.
     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
```


Copyright © 2021 Texas Instruments Incorporated

```
     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
```

[Case4 : ethtool –a eth0] , success example

```
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”.
```

[Case5 : ethtool –A eth0 ….] , success example

```
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.
```

[Case6 : ethtool –s eth0 ….]

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


Copyright © 2021 Texas Instruments Incorporated

3.17 Step 17 – Linux dmesg to Check the Ethernet Driver Status.

The *dmesg* is to get all the kernel log when the system is booting into the kernel. Type those commands to extract the network status will help you to debug.

- dmesg | grep *mdio* Purpose: This command will let you know the mdio status.
- dmesg | grep *link* Purpose: This command will let you know the link status. When you plug/unplug the cable, you can use this command to know the detail.
- dmesg | grep *cpsw* Purpose: This command will let you know the cpsw status. For additional information about CPSW, please go to: Sitara Linux Dual Emac Mode
- dmesg | grep *net* Purpose: This command will let you know the network (IPV4/IPV6 status).

3.18 Step 18 – Testing Result with Detail Log Analysis (Success Case).

The success case dump of the kernel are here as below. We list here by sections. When the DP83825 driver is initialized, you will have this kind of dmesg (AKA: kernel log)

```
[    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
```


Copyright © 2021 Texas Instruments Incorporated

3.19 Step 19 – YouTube Demonstration Video

From the YouTube application, search for *TI Rio Ethernet*.

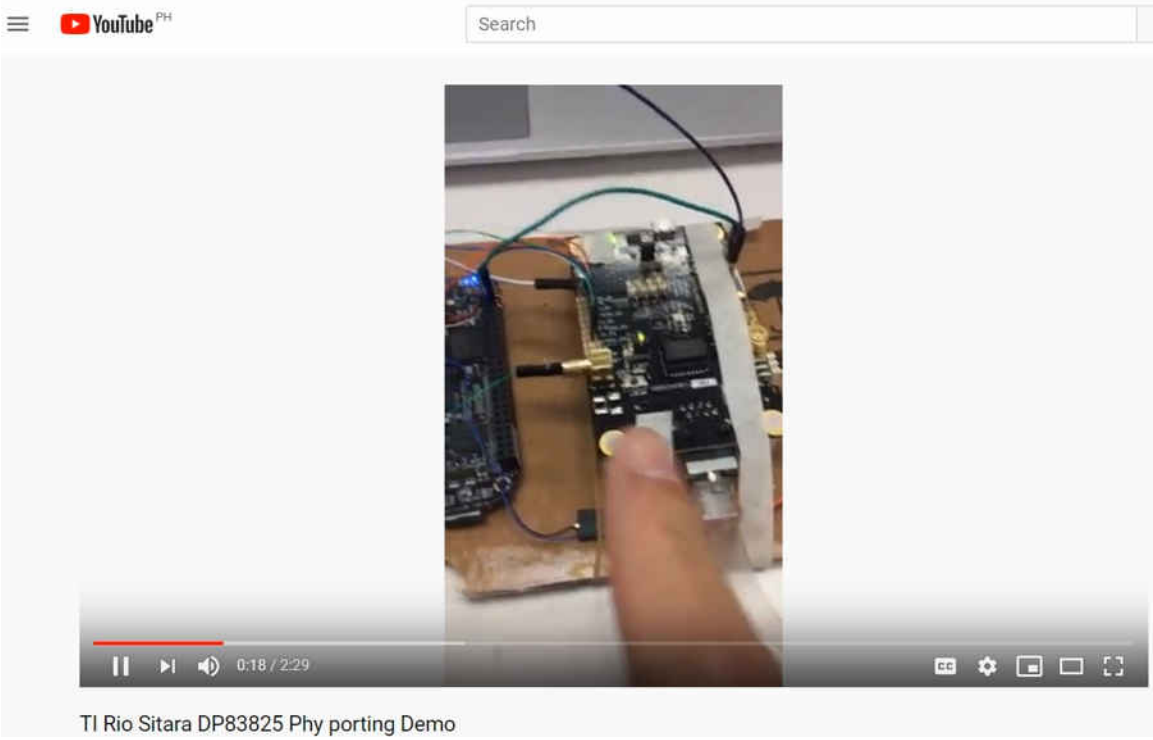

Figure 3-15. YouTube Demonstration Video

4 Required Hardware and Software

4.1 Hardware

- TI AM335 BeagleBoneBlask Version C
- TI DP83825 EVM

4.2 Software

- Pin mux tool
- TI AM335 Linux SDK Version 5.02
- DP83822 Patch
- DP83825 Patch

Copyright © 2021 Texas Instruments Incorporated

5 References

- Texas Instruments, DP83822 EVM User's Guide
- Texas Instruments, DP83822 Robust, Low Power 10/100 Mbps Ethernet Physical Layer Transceiver Data Sheet
- Texas Instruments, TI DP82825 EVM User's Guide
- DP83825 Low Power 10/100 Mbps Ethernet Physical Layer Transceiver Data Sheet
- TI AM335 BeagleBoard EVM schematic
- Texas Instruments, ICSS EMAC LLD Developers Guide
- Texas Instruments, Processor SDK Linux Software Developer's Guide
- Texas Instruments, PRU-ICSS Ethernet
- Texas Instruments, Sitara Linus Dual Emac Mode
- Texas Instruments, EtherNet/IP™ on TI's Sitara™ Processors White Paper
- Texas Instruments, Ethernet System Hardware on AM-Class Devices
- Texas Instruments, TMS320C6472/TMS320TCI6486 EMAC Implementation Guide Application Report (Page 8)
- Microchip, LAN8710A Data Sheet.

6 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision * (April 2020) to Revision A (July 2021)	Page
• Updated the numbering format for tables, figures and cross-references throughout the document	2

Copyright © 2021 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated