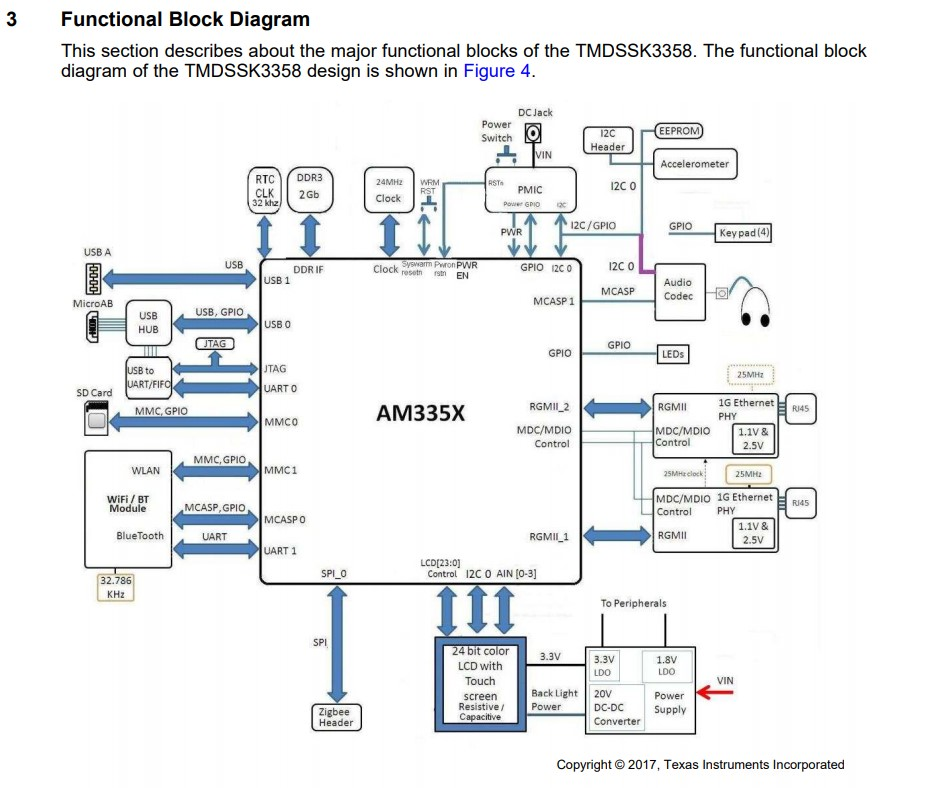
When posting to the TI e2e forums here are some recommendations on steps to try to isolate possible issues involving CPSW Ethernet on AM3x/4x/5x devices.

* Requirement
  + Any questions posted concerning the Linux CPSW Ethernet driver must be from using a TI Processor Linux SDK U-Boot/Linux Kernel and filesystem on either a TI EVM or a custom board. Texas Instruments only supports Linux Kernels and U-Boot images that are based on the TI Processors Linux SDK.

Yes.

* During the boot process check to make sure the Ethernet PHYs being are reported correctly in the console log. This means that the PHY being used should be reported and the driver being used is indicated. The CPSW should indicate it has initialized and finally it should report the interface is up. This is example with eth0 connected to a link partner. It should look something like this:

Yes.



[ 1.366149] davinci\_mdio davinci\_mdio.0: davinci mdio revision 1.6

[ 1.372619] davinci\_mdio davinci\_mdio.0: detected phy mask fffffffc

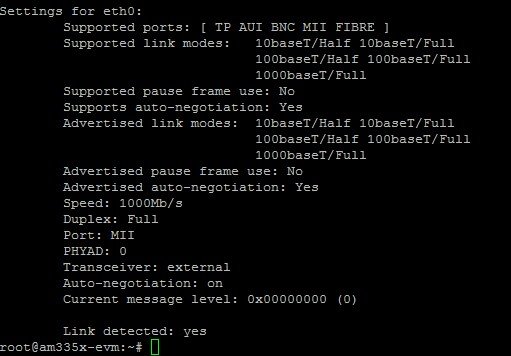
[ 1.380645] davinci\_mdio.0: probed

[ 1.384185] davinci\_mdio davinci\_mdio.0: phy[0]: device 0:00, driver unknown

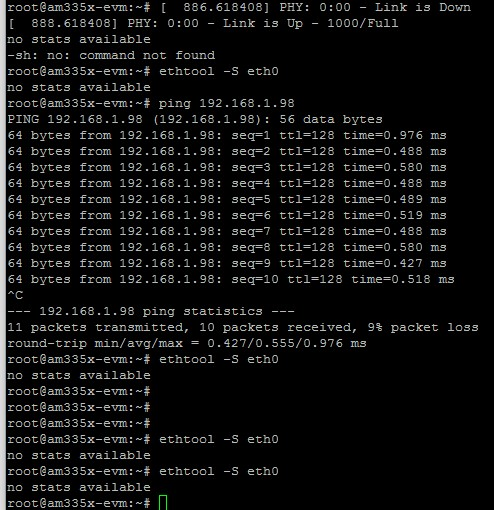
[ 1.391601] davinci\_mdio davinci\_mdio.0: phy[1]: device 0:01, driver unknown

[ 1.399230] usbcore: registered new interface driver zd1201

* Look at basic link elements after the system is booted to make sure that there is a link partner detected. No Ethernet traffic will be possible until the PHY is reporting a link detection. In fact, if no link is detected by the PHY that issue needs to be debugged first. **Do not proceed debugging until the PHY is detecting a link.** This means working with the PHY manufacturer to resolve the link detection issue. The commands are in bold.
  + **ethtool eth0** : This will show what the link status is. Here are some questions that might help with link status trouble-shooting.
    - Is auto negotiation enabled on both ends of the link?
    - Do the link speeds match between the TI device and the link partner?
    - Is the Link duplex mode correct?



* + **ethtool -S eth0** : This will show the hardware statistics. Here errors are reported for CRC errors, overruns etc. Errors reported by the hardware typically indicate the packet was dropped. Here are a couple of key errors to review:
    - RX Checksum errors – Can indicate bad cables, layout issues.
    - RX Overrun errors – This indicates packet drops due to the RX DMA descriptors being momentarily exhausted.
    - RX Start of Frame errors – This indicates that packets that have been dropped due the internal FIFOs of the switch being temporarily full.

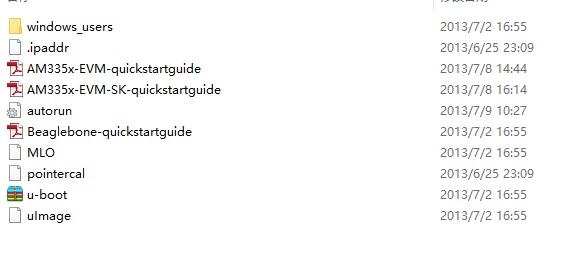


* What does the interface say:
  + ifconfig
    - Is the link up? Yes.
    - Is there an IP address? Yes.
    - Is the interface reporting data being received or transmitted? Yes.
* Network view
  + Use Wireshark from the view of a link partner, typically the Linux PC is used here to capture packets.
  + Start a ping transaction from the device which will start an ARP process. Do any ARP packets show up in Wireshark? Yes.
* Performance
  + When testing the interface with iperf a very common question is about not being able to match the numbers reported as part of the performance datasheet release with the TI SDK. The most common reason for this is because the developer started with a mainline kernel and not a TI SDK kernel. Also, when pulling a mainline kernel the defconfig supplied with mainline is not the same as the kernel defconfig supplied and tested with the TI SDK. Using a mainline kernel will have a significant performance drop.
* When posting to the forums please capture the information requested below. Please send these as attachments as some of the logs will be lengthy.
  + Kernel version and source, also include the results of this command: uname -a



* + File system, TI SDK or Arago/Yocto based filesystem

**TMDSSK3358**



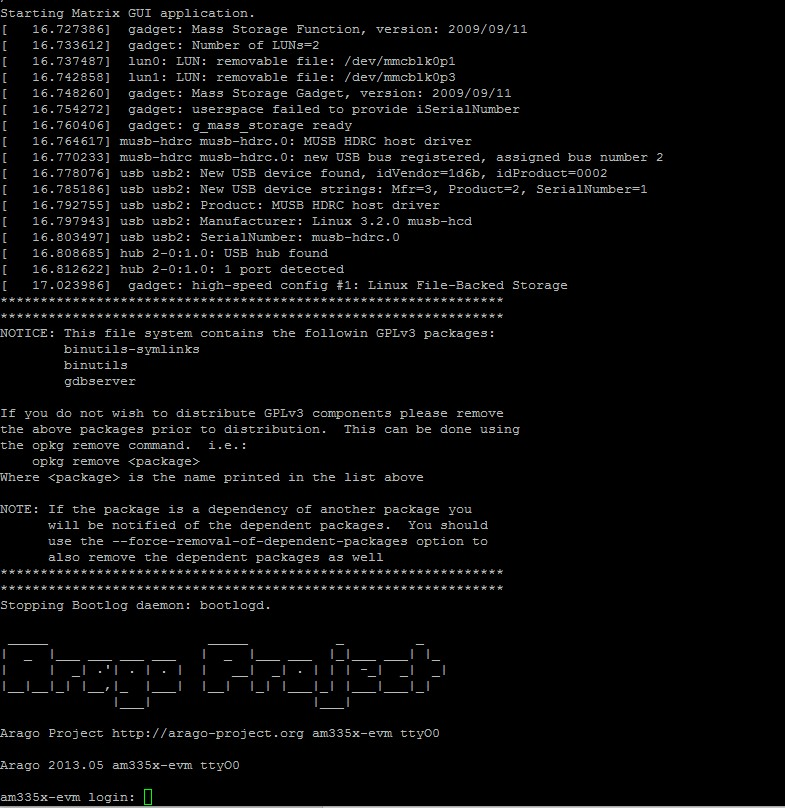
* + Custom board or TI board? Please include device tree source file.

TI board, **TMDSSK3358**

* + Console log of the boot process that includes U-Boot and the Kernel.



…



* + ethtool <interface such as eth0 or eth1>
  + ethtool -S <interface such as eth0 or eth1>
  + ifconfig <interface such as eth0 or eth1>ethtool

