

## ***FAQ - Enabling 8-Port Ethernet with Linux in J721E EVM***

### **1 Overview**

This document shows the steps required to enable all 8 Ethernet ports of CPSW\_9G in J721E with Linux. The instructions are applicable for SDK 7.1.

By default, Ethernet Firmware (EthFw) in SDK 7.1 enables only the four RGMII ports in GESI expansion board, but it doesn't enable the QSGMII ports in the QpENet (QSGMII) daughter board.

<b>Software Requirement</b>	<b>Hardware Requirements</b>
Processor SDK 7.1 Linux ( <a href="#">link</a> )	J721E EVM
Processor SDK 7.1 RTOS ( <a href="#">link</a> )	GESI expansion board
	QP Ethernet (QPENet) expansion board

The steps required to enable the 4 QSGMII ports can be divided into:

- Prepare an SD card with SDK 7.1 and make sure that Linux boots fine and default EthFw is loaded correctly.
- Disable PCIe SERDES0 configuration from Linux – this prevents conflicts between Linux and EthFw with SERDES usage, so EthFw can use it for QSGMII functionality.
- Enable QSGMII on EthFw.
- Take QSGMII PHY out of reset – this is done via u-boot commands.

### **2 Disable PCIe SERDES0**

If building kernel from source files:

- Apply the Linux kernel patch which is provided separately. This patch disables PCIe SERDES0 in J721E related device-tree files.
- Recompile Linux kernel dtbs.
- Copy the generated dtb files to <rootfs>/boot partition of the SD card.

Alternatively, one could just copy the precompiled dtb files that can be provided separately.

Boot Linux with the new dtb files and make sure that PCIe0 is disabled by running “lspci” on Linux terminal.

```
root@j7-evm:~# lspci
0000:00:00.0 Non-VGA unclassified device: Texas Instruments Device b00d
0001:00:00.0 Non-VGA unclassified device: Texas Instruments Device b00d
```

Also make sure that EthFw is still being loaded and running properly. Check EthFw logs in Main UART2 terminal.

```

=====
                        CPSW Ethernet Firmware
=====
CPSW_9G Test on MAIN NAVSS
EnetPhy_bindDriver: PHY 12: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK
EnetPhy_bindDriver: PHY 0: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK
EnetPhy_bindDriver: PHY 3: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK
EnetPhy_bindDriver: PHY 15: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK
PHY 0 is alive
PHY 3 is alive
PHY 12 is alive
PHY 15 is alive
PHY 23 is alive

ETHFW Version      : 0.01.01
ETHFW Build Date:  Nov 12, 2020
ETHFW Build Time:  01:42:42
ETHFW Commit SHA:  b4628e2d

Host MAC address:  70:ff:76:1d:92:c2
IPC_echo_test (core : mcu2_0) .....
[NIMU_NDK] ENET has been started successfully
Remote demo device (core : mcu2_0) .....
EthApp_initIpcTask: Ipc_lateVirtioCreate failed: -1
Function:CpswProxyServer_attachExtHandlerCb,HostId:0,CpswType:6
Cpsw_handleLinkUp: Port 1: Link up: 1-Gbps Full-Duplex
Function:CpswProxyServer_registerMacHandlerCb,HostId:0,Handle:a2bee00c,CoreKey
:38acb7e6, MacAddress:70:ff:76:1d:92:c1, FlowIdx:172, FlowIdxOffset:0
Cpsw_ioctlInternal: CPSW: Registered MAC address.ALE entry:11, Policer
Entry:0Function:CpswProxyServer_registerIpv4MacHandlerCb,HostId:0,Handle:a2bee
00c,CoreKey:38acb7e6, MacAddress:70:ff:76:1d:92:c1 IPv9

CPSW NIMU application, IP address I/F 1: 192.168.1.136

=====LLI Table entries=====
EthFw: TimeSync PTP enabled

Number of Static ARP Entries: 1
Rx Flow for Software Inter-VLAN Routing is up

SNo.          IP Address          MAC Address
-----
1             192.168.1.139        70:FF:76:1D:92:C1

```

### 3 Enabling QSGMII in EthFw

Enabling the 4 QSGMII ports requires changes in the Enet LLD as well as the EthFw. The changes are:

```

project ethfw/
diff --git a/apps/app_remoteswitchcfg_server/mcu_2_0/main_tirtos.c
b/apps/app_remoteswitchcfg_server/mcu_2_0/main_tirtos.c
--- a/apps/app_remoteswitchcfg_server/mcu_2_0/main_tirtos.c
+++ b/apps/app_remoteswitchcfg_server/mcu_2_0/main_tirtos.c
@@ -122,6 +122,8 @@
 /* HTTP webpage server header files */
 #include "webdata/webpage.h"

+#define ENABLE_QSGMII_PORTS
+
+/* =====
+*/
+/*
+/*                               Macros & Typedefs
+/*
+/* =====
+*/

project pdk/packages/ti/drv/enet/
diff --git a/examples/utils/makefile b/examples/utils/makefile
--- a/examples/utils/makefile
+++ b/examples/utils/makefile
@@ -18,7 +18,7 @@ include
$(PDK_INSTALL_PATH)/ti/drv/enet/examples/build/config.mk
#
#           | Bypass | Bypass |   UART |
#       App  |   I2C  | QSGMII | allowed | Library
# -----+-----+-----+-----+-----
-#   EthFw J721E |   Yes |   Yes |   Yes | enet_example_utils
+#   EthFw J721E |   Yes |   No  |   Yes | enet_example_utils
#   EthFw J7200 |   Yes |   No  |   No  | enet_example_utils
# CPSW examples |   No  |   No  |   Yes | enet_example_utils_full

@@ -34,9 +34,6 @@ else
    ENET_CFLAGS += -DSDK_6_2_CORE_SDK_IMAGE
    ifeq ($(CORE),$(filter $(CORE), mcu2_0))
        ENET_CFLAGS += -DENETAPPUTILS_BYPASS_I2C
-        ifeq ($(SOC),$(filter $(SOC), j721e))
-            ENET_CFLAGS += -DENETAPPUTILS_BYPASS_QSGMII
-        endif
        ifneq ($(SOC),$(filter $(SOC), j7200))
            ENET_CFLAGS += -DENETAPPUTILS_UART_ALLOWED
        endif

```

Since the Enet LLD change is in a makefile, a differential compilation will not recompile the relevant source files. It's recommended to make a clean PDK build.

The EthFw change is in a source file, so one can just recompile via `make -s -j ethfw_all SOC_LIST=J721E`.

Alternatively, one can just replace the EthFw image with the precompiled binary provided separately.

The EthFw binary must copied to  
<rootfs>/lib/firmware/ethfw/app\_remoteswitchcfg\_server\_strip.xer5f.

## 4 QSGMII PHY out of reset via U-boot commands

There are two GPIOs that need to be set for the QSGMII PHY to be functional. This can be done via u-boot commands.

```
# setenv init_main_cpsw0_qsgmii_phy "gpio set gpio@22_17; gpio clear  
gpio@22_16"  
# setenv bootcmd "run findfdt; run envboot; run init_${boot}; run  
init_main_cpsw0_qsgmii_phy; run boot_rprocs; run get_kern_${boot}; run  
get_fdt_${boot}; run get_overlay_${boot}; run run_kern"  
# saveenv
```

Above commands will set the relevant GPIOs automatically when u-boot is booting from SD card.

## 5 Testing new EthFw image

Finally, reboot the device and make sure that all 8 PHYs are being detected and connecting an Ethernet cable in any of the QSGMII ports results in EthFw reporting link-up event.

```
=====  
CPSW Ethernet Firmware  
=====  
CPSW_9G Test on MAIN NAVSS  
EnetPhy_bindDriver: PHY 12: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK  
EnetPhy_bindDriver: PHY 0: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK  
EnetPhy_bindDriver: PHY 3: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK  
EnetPhy_bindDriver: PHY 15: OUI:080028 Model:23 Ver:01 <-> 'dp83867' : OK  
EnetPhy_bindDriver: PHY 16: OUI:0001c1 Model:27 Ver:00 <-> 'vsc8514' : OK  
EnetPhy_bindDriver: PHY 17: OUI:0001c1 Model:27 Ver:00 <-> 'vsc8514' : OK  
EnetPhy_bindDriver: PHY 18: OUI:0001c1 Model:27 Ver:00 <-> 'vsc8514' : OK  
EnetPhy_bindDriver: PHY 19: OUI:0001c1 Model:27 Ver:00 <-> 'vsc8514' : OK  
PHY 0 is alive  
PHY 3 is alive  
PHY 12 is alive  
PHY 15 is alive  
PHY 16 is alive  
PHY 17 is alive  
PHY 18 is alive  
PHY 19 is alive  
PHY 23 is alive
```

```

ETHFW Version      : 0.01.01
ETHFW Build Date:  Jan 21, 2021
ETHFW Build Time:  18:48:44
ETHFW Commit SHA:  d2d55eab

Host MAC address:  70:ff:76:1d:92:c2
IPC_echo_test (core : mcu2_0) .....
[NIMU_NDK] ENET has been started successfully
Remote demo device (core : mcu2_0) .....
CpswMacPort_checkSgmiiStatus: MAC 4: SGMII link parter config port: link up:
1-Gbps Full-Duplex
Cpsw_handleLinkUp: Port 5: Link up: 1-Gbps Full-Duplex

CPSW NIMU application, IP address I/F 1: 192.168.1.136

EthFw: TimeSync PTP enabled
Rx Flow for Software Inter-VLAN Routing is up
REMOTE_SERVICE: Init ... !!!
REMOTE_SERVICE: Init ... Done !!!
Function:CpswProxyServer_attachExtHandlerCb,HostId:0,CpswType:6
Function:CpswProxyServer_registerMacHandlerCb,HostId:0,Handle:a2cd8da0,CoreKey
:38acb7e6, MacAddress:70:ff:76:1d:92:c1, FlowIdx:172, FlowIdxOffset:0
Cpsw_ioctlInternal: CPSW: Registered MAC address.ALE entry:13, Policer
Entry:2Function:CpswProxyServer_registerIpv4MacHandlerCb,HostId:0,Handle:a2cd8
da0,CoreKey:38acb7e6, MacAddress:70:ff:76:1d:92:c1 IPv8

=====LLI Table entries=====

Number of Static ARP Entries: 1

SNo.          IP Address          MAC Address
-----          -
1             192.168.1.138       70:FF:76:1D:92:C1

```

Port 5 in the logs above corresponds to the first QSGMII port. Similar messages will be seen when other ports are connected, either in GESI or QSGMII expansion boards.

Ports 1-4 are in GESI expansion board. Ports 5-8 are in QSGMII expansion board.

In the Linux terminal, check that the virtual Ethernet interface (eth1) is up by running the following command:

```
root@j7-evm:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 50:51:A9:FC:65:7E
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 70:FF:76:1D:92:C1
          inet addr:192.168.1.139  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::72ff:76ff:feld:92c1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 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:2034 (1.9 KiB)  TX bytes:6712 (6.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:82 errors:0 dropped:0 overruns:0 frame:0
          TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6220 (6.0 KiB)  TX bytes:6220 (6.0 KiB)
```

eth1 should show up in the interface list, and it will have an IP assigned if any of the switch ports is connected to a network running DHCP.