

Ethernet Boot Guide

- [Introduction](#)
- [Supported Combinations](#)
- [DHCP Setup](#)
- [TFTP Setup](#)
- [Steps to run Ethernet boot](#)
 - [Create a network between EVM and host PC](#)
 - [For Windows System](#)
 - [For Linux System](#)
 - [Build the example](#)
 - [Steps to run the example](#)
- [Sample Output](#)

Introduction

This document describes how to boot AM64x-SK board through Ethernet.

The bootloader is transferred from a PC running a DHCP server to the board via TFTP transfer. This bootloader does SOC initializations and provides the option to receive an application image via UDP over Ethernet and flashing the received application image to 0x80000 location in the OSPI Flash after successful completion of the image transferred over Ethernet.

Supported Combinations

Parameter	Value
CPU + OS	r5fss0-0 nortos
Toolchain	ti-arm-clang
Board	am64x-sk
Example folder	examples/drivers/boot/sbl_ospi_enet

Please Note: The SBL has been developed for AM64x-SK EVM (with DP83867 ETHPHY, RGMII interface and OSPI as the flash media). This will have to be further modified as per the custom board.

DHCP Setup

- On your PC, install DHCP server

```
sudo apt install isc-dhcp-server
```

- Disable services before configuring

```
sudo systemctl disable --now isc-dhcp-server.service isc-dhcp-server6.service
```

- DHCP server setup

Run the *ip link* or *ifconfig* command to find the name of your network interface:

```
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether c0:18:03:bd:b1:a6 txqueuelen 1000 (Ethernet)
    RX packets 2733979 bytes 1904440459 (1.9 GB)
    RX errors 0 dropped 3850 overruns 0 frame 0
    TX packets 796807 bytes 84534764 (84.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xe2200000-e2220000

enxf8e43b8cffe8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether f8:e4:3b:8c:ff:e8 txqueuelen 1000 (Ethernet)
    RX packets 95 bytes 31160 (31.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 89 bytes 17445 (17.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 85238 bytes 7244462 (7.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 85238 bytes 7244462 (7.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

A USB to Ethernet adapter connected is to one end on the PC and other to board.

*enxf8e43b8cffe8 is the interface for usb-to-ethernet adapter.

- Do following changes in /etc/dhcp/dhcpd.conf

```
subnet 192.168.0.0 netmask 255.255.255.0
{
    range dynamic-bootp 192.168.0.137 192.168.0.142;
    if substring (option vendor-class-identifier, 0, 16) = "TI K3 Bootp Boot"
    {
        filename "sbl_qlspi_enet.Release.hs_fs.tiimage";
    }
}
```

```
default-lease-time 60000;
max-lease-time 720000;
next-server 192.168.0.136;
}
```

- Do following changes in /etc/default/isc-dhcp-server

```
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
INTERFACESv4="enxf8e43b8cffe8"
INTERFACESv6=""
```

- For your interface, change IP address, gateway and netmask to the following (the below values are hard-coded in the SBL)

```
IP address - 192.168.0.136
Netmask - 255.255.255.0
Gateway - 192.168.0.195
```

- Enable DHCP

```
sudo systemctl enable --now isc-dhcp-server
```

- To see if there is any configuration error or if DHCP is running, run the below command

```
sudo service isc-dhcp-server status
* If its shows error then something is wrong with configuration
```

TFTP Setup

- Follow the link below to setup the TFTP server

```
https://linuxhint.com/install\_tftp\_server\_ubuntu/
```

Steps to run Ethernet boot

Add the files from the zip folder to their respective locations.

Project name: mcu_plus_sdk/examples/drivers/boot/sbl_osp_i_enet

Create a network between EVM and host PC

The IP addresses used here are also used by the python script and the EVM application, so ensure to change the IP addresses in all the places if there's any conflict with existing devices.

For Windows System

- Connect an Ethernet cable between the PC and the EVM.
- Open Ethernet settings on your PC >> select corresponding Ethernet adapter.

If setting the static IP for the first time, the Ethernet adapter may appear as "Unidentified network", which is the network to be selected.

- Edit the IP settings with the IP address as 192.168.0.136, Subnet Prefix Length as 24 and Gateway as 192.168.0.195.
- Make sure to set the connection as private and metered connection is set to off.
- Due to the limitations presented by Enet LLD in the Uniflash application, we need to manually add an ARP entry for the EVM in order for the PC to not drop the packets (PC will drop packets from sources for which it doesn't have an ARP entry.)
- Creating a static ARP entry requires admin privileges. Run the following commands in PowerShell as admin.

```
New-NetNeighbor -InterfaceIndex <ifIndex> -IPAddress '192.168.0.195'  
-LinkLayerAddress '<EVM MAC Addr>' -State Permanent
```

Replace *<ifIndex>* with the interface index of the connection between PC and EVM.

To find out the interface index corresponding to the Ethernet interface between the PC and the EVM, use the following PowerShell command. This does not require admin privileges

```
Get-NetAdapter
```

```
PS C:\Users\<username> > get-netadapter
```

Name	InterfaceDescription	ifIndex	Status	MacAddress	LinkSpeed
Ethernet	Realtek USB GbE Family Controller	13	Disconnected	<redacted>	0 bps
Ethernet 2	TP-LINK Gigabit Ethernet USB Adapter	11	Up	28-87-BA-3E-41-77	0 bps

In the above figure, "Ethernet 2" is the required interface and its interface index is 11.

The default EVM MAC Address is 70:ff:76:1d:ec:f2 (as set from SysConfig (Enet(CPSW) >> System Integration Config >> MAC Address)).

Replace *<EVM MAC Addr>* with the MAC Address of the EVM, as a continuous string like **70ff761decf2**.

An example command to set the ARP looks as below based on the above shown images.

```
New-NetNeighbor -InterfaceIndex 11 -IPAddress '192.168.0.195' -LinkLayerAddress  
'70ff761decf2' -State Permanent
```

For Linux System

- Run the *ip link* or *ifconfig* command to find the name of your network interface.
- Add a static ARP entry with the below command.

```
sudo arp -i <interface-name> -s <IP-Address> <MAC-Address>
```

Replace <EVM MAC Addr> with the MAC Address of the EVM, as 70:FF:76:1D:EC:F2 and <IP-Address> with IP address of the EVM, as 192.168.0.195

An example command to set the ARP looks as below based on the above shown images.

```
sudo arp -i eno1 -s 192.168.0.195 70:FF:76:1D:EC:F2
```

Build the example

- Import the project into CCS.
- Make sure to set the value of the MACRO **ENET_HOST_PC_MAC_ADDRESS** in the file **sbl_enet.h** to the MAC address of the corresponding Ethernet adapter, in the above case this is Ethernet 2's MAC address.
- Build the example in Release mode.
- Suppose your tftp directory (from above) is /tftp. Copy the generated **sbl_qspi_enet.Release.hs_fs.tiimage** into the /tftp folder on your PC.

Steps to run the example

- Power off the board and switch it to Ethernet boot mode.
- Connect to the UART terminal on CCS to view the SBL logs.
- Power on the board.
- Run below python command on the Windows command prompt (cmd.exe) or Linux bash shell to flash the files.

```
cd ${SDK_INSTALL_PATH}/tools/boot  
python enet_uniflash.py --cfg=sbl_prebuilt/am64x-sk/default_sbl_enet_app.cfg
```

- When the python script starts, it will display the message "Starting Linkup ...".
- After flashing is successful, the flashed application code will be automatically run after the SBL completes



Please Note

If Ethernet boot is set as Backup boot, it may take a while to switch to Ethernet boot and start TFTP transfer. This can result in Timeout of the python script.
To prevent that, run the python script right after the SBL logs appear on the CCS terminal.

Sample Output

Output in case of successful flashing and booting of the board:

CCS terminal Logs

```
DMSC Firmware Version 10.0.8--v10.00.08 (Fiery Fox)
DMSC Firmware revision 0xa
DMSC ABI revision 4.0

[ ENETSBL ] Starting Ethernet Transfer ...
Enabling clocks!
EnetAppUtils_reduceCoreMacAllocation: Reduced Mac Address Allocation for CoreId:1
  From 4 To 2
Open MAC port 2
EnetPhy_bindDriver:1843
PHY 0 is alive
PHY 1 is alive
EnetMod_ioctl:1608
Cpsw_registerIoctlHandler:1844
EnetPer_ioctl:1394
Enet_ioctl:1057
Failed to set dscp Priority map for Port 1 - -1
[ ENETSBL ] initQs() txFreePktInfoQ initialized with 8 pkts
[ ENETSBL ] EVM MAC address: 70:ff:76:1d:ec:f2
[ ENETSBL ] PHY 0 is alive
[ ENETSBL ] PHY 1 is alive
[ ENETSBL ] Please wait for Linkup ...
Cpsw_handleLinkUp:1450
[ ENETSBL ] Linkup Done!
[ ENETSBL ] Receiving file, please wait ...
[ ENETSBL ] Status:0
[ ENETSBL SUCCESS ] Ethernet Transfer Done.
[ ENETSBL ] Packets Received      : 46
[ ENETSBL ] Total File Size       : 65315 Bytes
[ ENETSBL ] Flash Offset          : 0x80000

Cpsw_handleLinkDown:1476
Disabling clocks for ENET: 5, inst:0!
Skipped boot-up, flashing complete!
```

Python script logs

```
C:\mcu_plus_sdk\tools\boot> python enet_uniflash.py --cfg=default_sbl_enet_app.cfg
[LOG] Parsing config file ...
[LOG] Ensure that sbl_qspi_enet has already been sent over before running this
script.
[LOG] Found 1 command(s) !!!
[LOG] Creating socket
Starting Linkup ...
Starting Linkup ...
Received.
('192.168.0.195', 5001)
```

```
[LINKUP] (SUCCESS) EVM Linked up. Starting transfer ...
```

```
[STATUS] Sent file hello_world.release.appimage.hs_fs of size 64923 bytes in 0.03s.
```

```
[FLASH] (SUCCESS) Flashing successful !!!
```

```
[LOG] All commands from config file are executed !!!
```