

Networking Guide for BeagleBone (Green or Black)

by Brian Fraser

Last update: Jan 11, 2024

This document guides the user through...

1. Setting up and configuring networking for the BeagleBone black or green with the host.

Guide has been tested on

BeagleBone (Target): **Debian 11.8**
PC OS (host): **Debian 11.8** (or higher)

Table of Contents

1. Network Setup.....	2
2. Ethernet over USB (Recommended).....	3
2.1 Initial Connection between Host and Target.....	3
2.2 Connecting Target to Internet.....	6
3. Wired Ethernet Network.....	9
4. Direct Ethernet Connection (Not Recommended).....	14
5. Linux Network Commands.....	17
5.1 General Commands.....	17
5.2 Set a static IP Address (optional).....	17
5.2.1 Temporarily Set IP Address.....	17
5.2.2 Permanently Set IP Address Host's GUI Tools.....	17
5.2.3 Permanently Set IP Address via Configuration File.....	18
5.3 Configure Network Adapter in Linux.....	18
5.3.1 Useful Networking Commands.....	20

Note: Guide not yet tested in the SFU Surrey Linux Lab. Some changes may be needed.

Formatting

1. Commands for the host Linux's console are show as:
`(host)$ echo "Hello PC world!"`
2. Commands for the target (BeagleBone) Linux's console are shown as:
`(bbg)$ echo "Hello embedded world!"`
3. Almost all commands are case sensitive.

Revision History

- Jan 11, 2024: Updated for Debian 11.8 (Bullseye)

1. Network Setup

You need to have an internet connection to both your Linux development OS (host PC) and the BeagleBone. Plus, you need to have your development OS be able to connect to your BeagleBone (we'll use SSH), and your BeagleBone be able to connect to your development OS (we'll use NFS). There are a couple ways you can connect your devices.

1. **Ethernet over USB** – Recommended

Have your BeagleBone connect to your development computer using Ethernet over USB, and your development computer connect to the internet via WiFi or wired Ethernet (see section 2).

This needs one USB-A to micro-USB cable (included in kit) to make the connection. No Ethernet cable is required.

2. **Physical Ethernet to Wired Network**

Have your development computer and your BeagleBone plugged into the same wired network, such as a home Ethernet network (see section 3).

This needs one Ethernet cable connected to the BeagleBone. The other end is plugged into your router.

3. **Ethernet via direct connection** – Not Recommended

Have your BeagleBone directly connected to your development computer via a single Ethernet cable (see section 4).

This needs one Ethernet cable: one end is connected to the BeagleBone; the other end is plugged into your computer (either via an internal Ethernet port, or an external Ethernet adapter, such as a USB Ethernet adapter).

See the necessary section below for the option you are using; you may skip the sections which do not relate to your setup. Section 5 covers some useful Linux networking commands.

2. Ethernet over USB (Recommended)

Expected setup is:

- host PC running Linux (native or in a VM) is connected to the Internet via WiFi or Ethernet;
- host PC connected to BeagleBone (target) via micro-USB cable for “Ethernet over USB”; no physical Ethernet cable is required.
- (optional) host PC connected to BeagleBone (target) via a serial cable (such as the micro-USB connection to the Zen cape, which can be used via `screen`) – for debugging only

2.1 Initial Connection between Host and Target

1. If using a VM:
 - Configure your VM's network connection to use NAT (Network Address Translation). This was likely the default when you installed your VM.
 - Launch the VM.
 - Ensure that inside the VM you are able to access the Internet by pinging Google (for example).

```
(host) $ ping google.ca
```
2. Connect the micro-USB cable from the host PC to the target (connector is on the BeagleBone, to the left of the Ethernet port).
 - Note that micro-USB port will power the target so to physically cycle the power you may need to disconnect the USB.
 - The USB cable required is a USB-A (host side) to micro-USB (target side).
3. If using a VM, map the USB connection to the VM.
 - In VMware, use the “Player” button in the top left, go to Removable Devices > Linux Foundation BeagleBoneGreen and select to Connect.
 - In Virtual Box, via VM's menu at the very top of the window select Devices → USB Devices → Seeed BeagleBoneGreen
4. On the host, check the network connection

```
$(host) ip addr
```

 - You may need to install the tools on the host via:

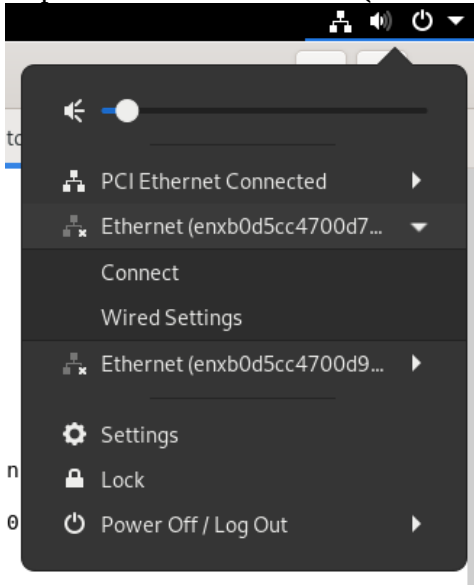
```
(host) $ sudo apt install net-tools
```
 - If it shows four network connections, and one has the IP address

```
inet 192.168.6.1/24 ...
```

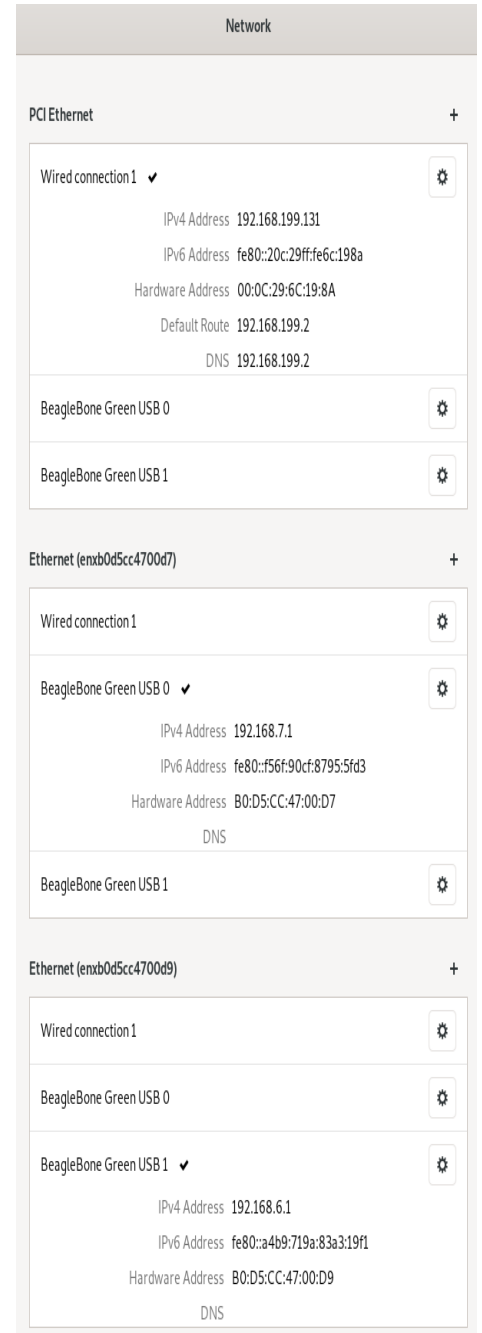
Then skip the Configure Network Profiles step
 - If it shows less than four network connections, then you may not have your BBG correctly connected to your VM. Repeat the step to map the BBG to the VM.

5. Configure Network Profiles

- Under host OS Debian, you will likely need to tell it to enable **both**:
 - a) your host OS's connection to the internet, and
 - b) your host OS's connection to the target (via Ethernet over USB).
- Click the Network Manager's tray UI icon in the top right hand corner. Expand the first **"Ethernet (enxb.....)"** entry and select **Wired Settings**



- Click the + icon to the right of the first "Ethernet (enx...)" entry:
 - Name it "BeagleBone Green USB 0"
 - Under IPv4, leave it on Automatic (DHCP).
 - Click "Add" button.
 - You should now see that profile with an IPv4 address of 192.168.7.1
- Click the + icon to the right of the other "Ethernet (enx...)" entry. Note that they may have re-ordered; you are now selecting the entry which has no IPv4 address.
 - Name it "BeagleBone Green USB 1"
 - Under IPv4, leave it on Automatic (DHCP).
 - Click "Add" button
 - You should now see a profile with an IPv4 address of 192.168.6.1
- Back on the Network Settings page, as shown on the right, each of the networking interfaces should be enabled with a different profile checked:
 - PCI Ethernet → Wired Connection 1



(VM's connection to internet)

- Ethernet (enx...) → BeagleBone Green USB 0, with IP address 192.168.7.1
(first BBG connection)
 - Ethernet (enx...) → BeagleBone Green USB 1, with IP address 192.168.6.1
(second BBG connection)
- Repeat the above `ip addr` command and ensure you now see one device with IP address 192.168.7.1 (among other entries).
6. Your Ethernet over USB should now be configured!
 - **The target will default to have a fixed IP address of 192.168.7.2** via the emulated USB network connection.
 7. Ping the target from the host; it should reply in time < 3ms.
(host) \$ `ping 192.168.7.2`
Press Ctrl-C to cancel the ping.
 8. Ping the host from the target (via SSH with default password `temppwd`, as shown here, or use a serial connection via the Zen cape)
(host) \$ `ssh debian@192.168.7.2`
(bbg) \$ `ping 192.168.7.1`
 - When you SSH into the target, it may ask you if you want to store the “ECDSA key fingerprint”; answer yes.
 9. Troubleshooting
 - Ensure you have connected the BeagleBone’s micro-USB connector to the computer (instead of the Zen cape’s; it’s OK to have them both connected to the host). Ensure the BBG’s lights are on and flashing showing that it has activity.
 - Ensure you give it enough time to finish booting. Connect to BBG using `screen` to see when it shows a log-in prompt on the serial port. At this time the BBG should connect to the host via Ethernet-over-USB.
 - If using a virtual machine, ensure the BeagleBone has been mapped to the Linux VM. Run `dmesg` in the VM before and after the BBG is connected. You should see some indication that it has begin interfacing with the BBG.
 - Ensure that the host is detecting the BeagleBone and creating an Ethernet-over-USB connection for it. To do so, use `ip addr` as follows:
 - Disconnect the BeagleBone from the host and run
(host) \$ `ip addr`
 - Reconnect the BeagleBone to the host, wait until boot-up has finished, and rerun
(host) \$ `ip addr`
 - Compare the two to ensure that a new networking adapter appears. Also, note name of the adapter (such as `eth7`, `usb0`, or something more cryptic like `enx8030dc9359be`).
 - If the target reboots, the host will lose the Ethernet-over-USB connection. The connection should come back when the board finishes rebooting.
 - If running VirtualBox, ensure you have installed the Virtual Box Extension Pack.

- Try rebooting the BBG, VM (if any), and your computer. Sometimes there can be USB issues.
- Plug the BBG into a different USB port on the host.
- If the host does not get the 192.168.7.2 IP address when the BeagleBone is connected, try assigning a static IP to the Ethernet-over-USB on the host. See Section 5.2 for directions.
- If software on the target stops working, the Ethernet connection on the host may disappear.
- If unable to ping between the host and target, run `ip addr` on each to ensure correct IP addresses.
 - Target's `usb0` connection should be IP address 192.168.7.2.
 - Host's `eth1/enxb...` connection should be IP address 192.168.7.1.
 - You will likely also see a connection with 192.168.6.1 and 192.168.6.2; it can be ignored. *The 192.168.7.1 uses a driver that Windows has built-in (RNDIS); 192.186.6.1 uses a driver that MacOS has built-in (CDC ECM). Under Linux (like us!) we have both connections show up. Either should work; we'll always use 192.168.7.1.*

2.2 Connecting Target to Internet¹

By following the above steps, you'll be able to ping between the host and the target. However, the target will not yet be able to access the internet. This section configures the host to allow the target to access the internet through its network connection.

1. Ensure your host PC (the Linux VM, likely) has access to the internet:

```
(host)$ ping google.ca
```

2. On the host PC, you should have two network adapters, likely named `eth0` (actual network connection) and `eth1` (virtual connection to target). Find them with:

```
(host)$ ip -4 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ..state UP group default qlen 1000
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 83993sec preferred_lft 83993sec
3: enxb0d5cc4700d8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ... state UNKNOWN ... qlen 1000
   inet 192.168.7.1/24 brd 192.168.7.255 scope global noprefixroute enxb0d5cc4700d8
       valid_lft forever preferred_lft forever
4: enxb0d5cc4700da: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ... state UP ... qlen 1000
   inet 192.168.6.1/24 brd 192.168.6.255 scope global noprefixroute enxb0d5cc4700da
       valid_lft forever preferred_lft forever
```

- Note: Connection to internet may come up as something like `ens33`, and the target's Ethernet over USB connection may come up as something like `enxb0d5cc4700d7`.

Order may be reversed of the 192.18.7.1 and 192.168.6.1 ports.

3. On the host, use the `iptables` program to configure the Linux kernel firewall rules:

```
(host)$ sudo iptables --table nat --append POSTROUTING --out-interface enp0s3 -j MASQUERADE
(host)$ sudo iptables --append FORWARD --in-interface enxb0d5cc4700d8 -j ACCEPT
```

4. Turn on IP forwarding:

¹ Directions from Molloy (2014) Exploring BeagleBone, Ch2.

```
(host)$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

5. On BBG, configure the target to use the host (via Ethernet over USB) as the default gateway:

```
(bbg)$ sudo route add default gw 192.168.7.1
```

- You can check this succeeded by viewing the routing tables on the target using `route`:

```
(bbg)$ route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          brian-VirtualBo 0.0.0.0          UG    0      0      0 usb0
192.168.7.0     *                255.255.255.252 U      0      0      0 usb0
```

6. Check the target has an internet connection:

```
(bbg)$ ping 8.8.8.8
```

- This should successfully ping the server. Note, you won't yet be able to ping `google.ca`!

7. Configure a DNS:

- Append a new domain nameserver (DNS) to the end of `etc/resolv.conf`:

```
(bbg)$ echo nameserver 8.8.8.8 | sudo tee -a /etc/resolv.conf
```

- Where `8.8.8.8` is a free public DNS owned by Google.
- The file `/etc/resolv.conf` may not exist on your board, and may get erased each boot. So you may need to do this step each time the board reboots (make a script, below).
- View the `/etc/resolv.conf` file to ensure it looks reasonable:

```
(bbg)$ cat /etc/resolv.conf
...
# No DNS servers known.
search .
nameserver 8.8.8.8
```

Note: Your `resolv.conf` file may only have the last line; this seems OK.

- Test with:

```
(bbg)$ ping google.ca
```

8. These settings will *not* persist between reboots, so you must re-do all of these steps each time you reboot in order to re-enable the settings.

- On the host, create an `internetToTarget.sh` file with contents similar to:

```
#!/bin/sh
sudo iptables --table nat --append POSTROUTING --out-interface ens33 -j MASQUERADE
sudo iptables --append FORWARD --in-interface enxb0d5cc4700d7 -j ACCEPT
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

- On the target, create an `internetToTarget.sh` script with contents similar to:

```
#!/bin/sh
sudo route add default gw 192.168.7.1
echo nameserver 8.8.8.8 | sudo tee -a /etc/resolv.conf
```

- Change each of these to be executable:

```
$ sudo chmod +x internetToTarget.sh
```

9. Trouble shooting:

- If you rebooted either your target or host, you'll need to redo the settings on that device as the settings are not persistent.

- If you are not able to ping 8.8.8.8 from the target, first try pinging the host:

```
(bbg)$ ping 192.168.7.1
```

 If this fails, it likely means that your BBG is connected to your main OS on the computer, not to the VM. Connect the BBG to the VM and retry the ping.
- If you are not able to ping 8.8.8.8 from the target, but have done all the above steps, then you may need to try the following on your host (after other iptables commands and before turning on IP forwarding)²:

```
(host)$ sudo iptables --append FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```
- If unable to ping 8.8.8.8 from the target, then verify that you have selected the correct network adapter (as listed by ip addr) on the host.
 - Install traceroute:

```
(host)$ sudo apt install traceroute
```
 - Run traceroute:

```
(host)$ traceroute 8.8.8.8
```

```
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  2.150 ms  2.074 ms  1.922 ms
 2  ...
```
 - Ensure that the first IP address listed in hop 1 of output matches the IP address listed for the Ethernet adapter you are using.
- If you get the error message: “ping: unknown host google.ca” when trying to ping google.ca, then you don't have a DNS server setup correctly. Double check you have completed the DNS steps above.
- On the target, when trying to ping, if you get:

```
connect: Network is unreachable
```

 double check that you have successfully executed the command:

```
(bbg)$ sudo route add default gw 192.168.7.1
```

 When you run route, it should show that there is a default row in the routing table.
- On the target, when trying to ping google.ca, if you get:

```
“ping: google.ca: Temporary failure in name resolution”
```

 Ensure you nameserver 8.8.8.8 command above.

2 Thank you to a student for this; for reference, see: <https://serverfault.com/questions/436678/ping-reply-not-getting-to-lan-machines-but-getting-in-linux-router-gateway/436683#436683>

3. Wired Ethernet Network

This section assumes you are connecting your BeagleBone to the same wired network that your development PC is connected to. If this is not your setup, you may skip this section.

Note: This section has not been verified to work with a Debian 11 (Bullseye) host; some changes may be necessary. Consult the [Debian NetworkManager page](#) for more help.

1. If using a virtual machine on the development computer, completely shutdown (power down, not just close/suspend) your virtual machine and change the network adapter type to “Bridged” instead of “Network Address Translation (NAT)”.
 - Under VMWare Player, select the VM and then “Edit Virtual Machine Settings” → Network Adapter → Bridged. OK to “Replicate physical network connection state”. Restart the VM.
2. Connect an Ethernet cable to the target board and have it get an IP address from your network's DHCP server (likely your home router).
 - The board should automatically acquire a IP address in a few seconds (~30); if not, you may need to re-run, on the BeagleBone, either `dhclient` or `udhcpd` (depending on which is installed) to force it.
(bbg) \$ **sudo dhclient**
 - You may need to reboot the target to have it acquire an IP address at startup:
(bbg) \$ **reboot**

3. Check the IP address of the target board using the serial connection:

```
(bbg) $ ifconfig
eth0      Link encap:Ethernet  HWaddr c8:a0:30:aa:dd:a0
          inet addr:192.168.0.145  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::caa0:30ff:feaa:dda0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0
          TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4176 (4.0 KiB)  TX bytes:9945 (9.7 KiB)
          Interrupt:40
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:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb0     Link encap:Ethernet  HWaddr 6a:41:80:a7:bf:db
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
          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)
```

- Look for `eth0`'s “`inet addr:`”, such as:

```
inet addr:192.168.0.145  
(Highlighted in above example)
```

- If you do not see `eth0`, it means the board has likely not yet found an IP address. Wait up to a minute longer for it to get an IP address. If it does not, double check all Ethernet cable connections and reboot the target.
- You may see (and ignore) some messages on the target such as:
[24.172268] net eth0: phy 4a101000.mdio:01 not found on slave 1
[29.810445] libphy: PHY 4a101000.mdio:01 not found

4. Check the IP address of the Host PC:

```
(host)$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:0c:29:7f:47:bc  
          inet addr:192.168.0.114  Bcast:192.168.0.255  Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:fe7f:47bc/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:196832 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:5079 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:41734380 (41.7 MB)  TX bytes:1109534 (1.1 MB)  
  
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:747 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:747 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:277191 (277.1 KB)  TX bytes:277191 (277.1 KB)
```

- Find `eth0`'s IP address, such as:
`inet addr:192.168.0.114`
- If running in a virtual machine, check the host OS's IP address as well: under Windows run `ipconfig` from the command line.

5. The addresses of the host and target (and host OS if running a VM) should be on the same subnet (have the first couple octets the same). If they are not the same then check the troubleshooting section below.

- Note that if your target device is plugged into a different network than your host PC they will be on different subnets. This may be the case in the SFU labs.

6. From the **host PC**, **ping the Target**. Command and output shown below. Press Ctrl-C to stop the pinging as needed. Replace the IP address with whatever address your target has.

```
(host)$ ping 192.168.0.145  
PING 192.168.0.145 (192.168.0.145) 56(84) bytes of data.  
64 bytes from 192.168.0.145: icmp_seq=1 ttl=64 time=1.10 ms  
64 bytes from 192.168.0.145: icmp_seq=2 ttl=64 time=0.585 ms  
64 bytes from 192.168.0.145: icmp_seq=3 ttl=64 time=0.492 ms  
^C  
--- 192.168.0.145 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2001ms  
rtt min/avg/max/mdev = 0.492/0.727/1.105/0.270 ms
```

7. From the **target**, **ping the host PC** using your serial terminal. Command and output shown

below. Press Ctrl-C to stop the pinging as needed. Replace the IP address with whatever address your host has.

```
(bbg) $ ping 192.168.0.114
PING 192.168.0.114 (192.168.0.114) 56(84) bytes of data.
64 bytes from 192.168.0.114: icmp_req=1 ttl=64 time=1.49 ms
64 bytes from 192.168.0.114: icmp_req=2 ttl=64 time=0.619 ms
64 bytes from 192.168.0.114: icmp_req=3 ttl=64 time=0.620 ms
64 bytes from 192.168.0.114: icmp_req=4 ttl=64 time=0.707 ms
^C
--- 192.168.0.114 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.619/0.861/1.499/0.370 ms
```

8. Ensure your target and host can both access the internet:

```
(host) $ ping google.ca
(bbg) $ ping google.ca
```

- If the ping fails, double check your network setup.

9. Troubleshooting:

- If the VM is not on same subnet as host OS, then make sure the VM's network adapter is configured for Bridged mode, instead of NAT. NAT uses the host OS's IP address for access to the network which will work fine for outgoing connections (such as browsing to Google), but not work for incoming connections (such as running an NFS server). You may need to restart the VM if you make any changes.
- If your target board (BeagleBone) is on a different subnet than your host PC, double check that your host PC and target board are plugged into the same network. For example, you could have a problem if the host PC is connecting to a wireless network but the target device is connecting to a completely separate wired network. Reboot the target device to have it acquire a new IP address.

- If `ifconfig` on the target does not show `eth0` you may need to manually bring it up and possibly assign it an IP address:

```
(bbg) $ ifconfig eth0 up
(bbg) $ ifconfig eth0 192.168.2.2
```

- `dmesg` may show output similar to:

```
[...] net eth0: initializing cpsw version 1.12 (0)
[...] net eth0: phy found : id is : 0x7c0f1
[...] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[...] cpsw 4a100000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
[...] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```
- Note that the Ethernet LEDs seem to turn on at the same time the final two lines of this output are displayed.
- Check the BeagleBone Ethernet LEDs:
 - On the BeagleBone's *Ethernet connector* there are two LEDs:
 - Yellow LED indicates a connection. When lit it indicates that the Ethernet port is powered on and detecting that an Ethernet cable is plugged in and connecting it to another device (likely a router or a direct connection to computer).

- Green LED indicates network activity by flashing. Usually lit, flickers off when transmitting/receiving data.
- On the target, if both LEDs are turned off, you may need to manually bring the Ethernet adapter down and up, and possibly assign it an IP address:


```
(bbg) $ ifconfig eth0 down
(bbg) $ ifconfig eth0 up
(bbg) $ ifconfig eth0 192.168.2.2
```

 - I have had it where this sequence of commands failed to immediately bring the Ethernet up, but the Ethernet LEDs did turn on about a minute later.
 - There have been issues with BeagleBone hardware (resolved now?) where the Ethernet physical-layer chip (the phy) could incorrectly initialize at startup due to a hardware race condition. A physical reset (pull power and reboot) could sometimes fix it (software reset insufficient).
- Test if there are issues with the target's Linux install with respect to networking by using UBoot to ping the host:
 - Using the serial port via the screen program, connect to the board and enter UBoot on boot by pressing any key when prompted.
 - In Uboot, set the target to a static IP and ping the server (assuming server is on 192.168.2.1):


```
=> setenv ipaddr 192.168.2.2
=> ping 192.168.2.1
```

 - Note that Uboot can sometimes behave funny with the backspace character if you make an error typing the command. Therefore you may want to enter the commands in a text editor on the host and copy-and-paste them into screen.
 - Works: Expected working output:


```
=> ping 192.168.2.1
link up on port 0, speed 100, full duplex
Using cpsw device
host 192.168.2.1 is alive
```
 - Fails: Pinging an address which no computer is at:


```
=> setenv ipaddr 192.168.2.2
=> ping 192.168.2.123
link up on port 0, speed 100, full duplex
Using cpsw device

ARP Retry count exceeded; starting again
ping failed; host 192.168.2.123 is not alive
=>
```
 - Fails: Pinging without Ethernet connection:


```
=> ping 192.168.2.1
cpsw Waiting for PHY auto negotiation to complete..... TIMEOUT !
Using cpsw device

ARP Retry count exceeded; starting again
ping failed; host 192.168.2.1 is not alive
```
- Test with another BeagleBone in the identical setup as your board (same cable, same

connections, same settings) and see if it works. This can help track down issues with router behaviour, cables, etc. Don't underestimate how commonly cables fail!

4. Direct Ethernet Connection (Not Recommended)

You can directly connect the target device to a host PC using an Ethernet cable (or crossover). You may skip this section if you are connecting to the BeagleBone via either Ethernet over USB or through a wired network (router/switch).

Note: This section has not been verified to work with a Debian 11 (Bullseye) host; some changes may be necessary. Consult the [Debian NetworkManager page](#) for more help.

Expected connection is your PC (or virtual machine) is connected to the web via either WiFi or Ethernet, and you are going to connect an Ethernet cable directly from your PC to the target. (i.e., the target is connected to the host only, not to any network such as a router).

1. On the target, via a serial connection, set it to a static IP address:

```
(bbg) $ ifconfig eth0 192.168.2.2
```

- You will need root access to run this command. You may need to run the “su” command first (password is “root”) to get a root command prompt.
- Note that this will revert back to DHCP when the target is rebooted. You can make the change permanent by editing `/etc/network/interfaces` and adding:

```
auto eth0
iface eth0 inet static
    address 192.168.2.2
    netmask 255.255.255.0
    network 192.168.2.0
    broadcast 192.168.2.255
```

- After changing the file, restart networking on the board (or reboot)

```
(bbg) $ /etc/init.d/networking restart
```

2. Edit the settings for your virtual machine.

Virtual Box:

- You must have your VM powered down, not just suspended.
- Right click the VM in your Oracle VM VirtualBox Manager and select Settings.
- Under Network, enable Adapter 1 and set to NAT.
This adapter will give your VM access to the internet through your host OS.
- Under Networking, enable Adapter 2 and set to Bridged.
 - In the “Name” drop-down select your Ethernet adapter.
 - This adapter will be the connection to the target.
- Launch your virtual machine.

VM Ware

- Click the “Player” button in the top left, and select Manage, then Virtual Machine Settings.
- On the Hardware tab, select “Add...”, and add a new “Network Adapter”
- Change new network adapter to:

- Network Connection: “Bridged”
 - Check “Replicate physical network connection state”
 - Under “Configure Adapters”, **only check the wired Ethernet adapter on your computer** (ensure it's not a wireless one, and not a BlueTooth one).
 - Press OK.
3. On the host (inside your VM), edit the `/etc/network/interfaces` file and add:


```
auto eth1
iface eth1 inet static
    address 192.168.2.1
    netmask 255.255.255.0
    network 192.168.2.0
    broadcast 192.168.2.255
```

 - Note that `eth1` may need to be changed to another number if you are running Linux natively (likely `eth0`), or have a different number of network adapters connected to your VM. Use `ifconfig` to list Ethernet adapters and pick the correct number.
 4. Restart the networking:


```
(host)$ sudo /etc/init.d/networking restart
```
 5. Test your host's internet settings on the host:
 - Run `ifconfig`. `eth0` will likely have an IP address in the range of 10.0.2.15 and `eth1` should have an IP address of 192.168.2.1.
 - Ping Google to prove you have a good internet connection.


```
(host)$ ping google.ca
```
 - See troubleshooting below if there are any problems.
 6. Ping the target from the host, and the host from the target.


```
(bbg)$ ping 192.168.2.1
(host)$ ping 192.168.2.2
```
 7. Troubleshooting
 - Run `ifconfig` on the target and host to ensure they have the correct addresses.
 - If your VM cannot access the Internet (say pinging Google or installing via `apt`), then ensure that your host OS (Windows, for example) has a good connection to the Internet.
 - Ensure your VM has the correct network adapters installed. In VirtualBox, check that it has enabled the network connection:
 - In the VM's menu, select Devices → Network
 - Ensure that two network connections are listed and that both are enabled (filled background around the icon).
 - Check that the network cable connected correctly.
 - Check that the connection is enabled in Linux. If `ifconfig` does not show `eth0` (on the target) or `eth1` (on the host), then enable the adapter using a command such as:


```
(host)$ sudo ifconfig eth0 up
```
 - Double check the VM's networking settings. Ensure you select NAT and Bridged correctly.

- If using VMWare, ensure that your VM's bridged network adapter's "configured adapter" is set to only connect to your Ethernet (and not bluetooth, WiFi, or any other systems).
- For more information on configuring the network settings, see: <https://help.ubuntu.com/community/NetworkConfigurationCommandLine/Automatic>
- If the networking does not work, ensure you are setting up your bridged Ethernet adapter correctly. Ensure that you are selecting the correct one. A physical network adapter may say "PCI" or "Realtec" (and "Ethernet" may not actually be the one you need). Try changing the adapter you are bridging to be different ones in the list.
- In VMWare Player, if there are no network adapters shown in the drop-down list when you try to configure your bridged network, you may want to install the latest version of VMWare Pro ([free through SFU CMPT](#)), or [try something like vmnetcfg](#) and reset networking settings.
- Read section 3. of this guide and its troubleshooting steps for more ideas on what to try.

5. Linux Network Commands

5.1 General Commands

Here are some useful Linux commands for working with Ethernet (may need root access). These can be used on either the host or the target.

- Display Ethernet configuration information:
`$ ip addr`
or
`$ ifconfig`
- Turn off, and turn on Ethernet connection (to cause it to redetect it):
`$ sudo ifconfig eth0 down`
`$ sudo ifconfig eth0 up`
Or:
`$ sudo ifdown eth0`
`$ sudo ifup eth0`
- Run DHCP to acquire an IP address (only on some systems):
`$ dhclient`

5.2 Set a static IP Address (optional)

Usually, IP addresses are assigned automatically via DHCP (for normal Ethernet), or by the configuration of the Ethernet-over-USB system. However, when these are not working it is possible to force the connection to have an IP address manually using static IP address assignment.

5.2.1 Temporarily Set IP Address

- Force a new (static) IP address:
`$ ifconfig eth0 192.168.0.212`
- Note that when the system reboots, these changes will be lost. Plus, some systems seem to automatically override this change every now and then.

5.2.2 Permanently Set IP Address Host's GUI Tools

- In the host PC, in the top-right corner, click on the networking button to launch the Network Manager tray UI (or search Network in the settings).
- To know which connection to edit, click on each of network entries in the list to see their Hardware Address. Match this to the `HWaddr` shown by `ip addr` or `ifconfig`.
 - If you are unsure which connection in `ifconfig` is for the connection of interest, try disconnecting the BeagleBone's Ethernet-over-USB and see what connection disappears. This may help you identify them.
- If turning on a profile for one network connection disables the profile for another connection, then create a new network profile for each network connection.
- Select the network listed for the BeagleBone, and click Options (or the config icon).
- Go to the IPv4 Settings tab.
- Change the Method to Manual

- Click Add to add an IP Address and set desired address.
 - For the Ethernet-over-USB, use IP 192.168.7.1, netmask 255.255.255.0, gateway 0.0.0.0
- Click Save.

5.2.3 Permanently Set IP Address via Configuration File

- Make a backup of the internet configuration file:


```
$ sudo cp /etc/network/interfaces /etc/network/interfaces.bak
```
- Edit configuration file to set a static IP at boot for Debian based (such as Ubuntu) systems:


```
$ sudo gedit /etc/network/interfaces
```
- For the Ethernet connection in question (assumed to be `eth5` for this example), add the following to the end of the file (change connection name and IP address as needed):


```
# some comment to describe what you are doing!
auto eth5
iface eth5 inet static
    address 192.168.7.1
    netmask 255.255.255.0
    network 192.168.7.0
    broadcast 192.168.7.255
```

 - Note: Debian 6.0 (Squeeze) and higher will not use the Network Manager for any network interface which is listed in the `/etc/network/interfaces` config file.
- Restart networking (try these until one works!):


```
$ sudo service NetworkManager restart
$ sudo service network-manager restart
$ sudo service networking restart
```

 - If this does not work, try (replace `eth5` with your connection name):


```
$ sudo ifdown eth5
$ sudo ifup eth5
```
 - Or try:


```
$ sudo /etc/init.d/networking restart
```
 - Or, if that does not work, reboot.
- Now try pinging between host and target.
- For more information, see the [Debian Networking Wiki](#).

5.3 Configure Network Adapter in Linux

Usually Linux will detect a networking adapter automatically. If you have problems, try:

- Check if you can access the internet:


```
$ ping 8.8.8.8
```

 - If running as a VM, check your host OS can access the internet too!
- Check if Linux is detecting any networking adapters:


```
(host)$ dmesg | grep -i eth
[...] e1000 0000:02:01.0 eth0: (PCI:66MHz:32-bit) 00:0c:29:8c:bd:f1
[...] e1000 0000:02:01.0 eth0: Intel(R) PRO/1000 Network Connection
[...] e1000 0000:02:01.0 ens33: renamed from eth0
```

 - If it finds something, it means the kernel is detecting your hardware and you ‘just’ need to

configure it in user-land. Look for a line “XYZ: renamed from ethXXX”: it tells you what Linux is calling your network adapter. Mine is `ens33` in this example.

- If none are found, then check your hardware, or your VM settings to map a network adapter to the VM.
 - Ensure that your VM software has configured an Ethernet adapter (bridged or NAT) to the VM.
 - Ensure that the network adapter is enabled via the VM software.

- See if `ifconfig` detects it:

```
$ ifconfig
```

- If your interface is not listed, then the interface is not yet online and working. So, check if `ifconfig` even knows it is there at all:

```
$ ifconfig -a
```

If this lists your adapter but has no IP address (`inet`), then you may need to force Linux’s user-land programs to configure the network by editing the `interfaces` file (below).

- Try bringing the interface down/up manually:

```
$ sudo ifdown ens33
```

```
$ sudo ifup ens33
```

- If present, you can check your interfaces file: `/etc/network/interfaces`. The BeagleBone likely has this file; however, Ubuntu does not.

```
$ cat /etc/network/interfaces
```

```
# --- DEFAULT CONTENTS OF /etc/network/interfaces FILE:
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
```

```
auto lo
```

```
iface lo inet loopback
```

- You may need to force Linux user-land software to configure the network adapter. For reference, I have had to change the file after the VM has a problem:

```
$ sudo nano /etc/network/interfaces
```

```
# --- MODIFIED CONTENTS OF /etc/network/interfaces FILE:
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
```

```
auto lo
```

```
iface lo inet loopback
```

```
iface ens33 inet dhcp
```

```
auto ens33
```

- Force networking to restart:

```
$ /etc/init.d/networking restart
```

```
[ ok ] Restarting networking (via systemctl): networking.service.
```

- If the restart fails, try the following two commands to give you some visibility into what is going wrong:

```
$ systemctl status networking.service
```

```
$ journalctl -xe
```

- List the network interfaces that Linux recognizes:

```
$ ip link show
```

- UDev manages what happens when Linux detects the BeagleBone connected to the host.

- You can see `udev` messages with:

```
$ udevadm monitor
```

- When detected, networking adapters are listed in:

```
$ ls /sys/class/net/
```

5.3.1 Useful Networking Commands

- `ip -4 addr`
 - Good for listing what IP addresses you currently have
 - Note that “addr” is short for “address”, and could even be shortened to “a”
- `ip -4 -o addr`
 - Shows IPv4, one line per.
- `ip link`
 - Shows link level details (but not the IP address)
- `ip route`
 - Shows routing and IP interface info
 - Shows “kernel” if it’s a kernel widget, such as for Ethernet over USB.
- Colourize output of `ip` (add `-c` option):
 - `ip -c -4 addr`
 - Default to colour
(host)\$ nano ~/.bashrc
(bbg)\$ nano ~/.profile

 - add (for colour)
alias ip="ip -c"
- Set static IP under Ubuntu:
 - `sudo nano /etc/netplan/01-network-manager-all.yaml`
 - Apply changes:
`sudo netplan apply`
 - Troubleshooting:
 - No network adapters show up in Settings > Network (just shows VPN), then Linux may have problems reading the netplan config file. Review the file and look for errors. Try renaming the config file:
(host)\$ sudo mv /etc/netplan/01-network-manager-all.yaml \
/etc/netplan/01-network-manager-all.yaml.BAK
(host)\$ sudo netplan apply
 - Ensure you have applied any pending changes in your netplan file:
(host)\$ sudo netplan apply
 - Unable to ping target from host:
(host)\$ ping 192.168.7.2
PING 192.168.7.2 (192.168.7.2) 56(84) bytes of data.
From 192.168.7.1 icmp_seq=1 Destination Host Unreachable
 - Double check that you have the correct IP address assigned to the correct network connection:
(host)\$ ip addr

Ensure that the smaller device number (such as enxb0d5cc4700d8) is assigned 192.168.7.1. Note addresses are in hex, and the **larger** interface number is listed first in “ip addr”

- See network devices hardware:

```
$ sudo apt install lshw  
$ sudo lshw -class network
```
- If network is “DOWN” under Ubuntu, then you likely have a problem with your netplan config file. Review it for issues. You could try the following to force the network adapter to turn on (however, this often has limited success):

```
$ sudo ip link set dev [INTERFACE] up
```
- View what Linux names the interfaces in the UI (host only):

```
(host)$ nmcli dev status
```
- Temporarily set an IP address:

```
$ ip addr add 192.168.7.1/24 dev enxb0d5cc4700d7
```

 - If you forget the /24 then it will default to the wrong netmask and not think it’s on a network with any other devices. Hence, it will show you have an IP address, but not be willing to use it to communicate with other devices.