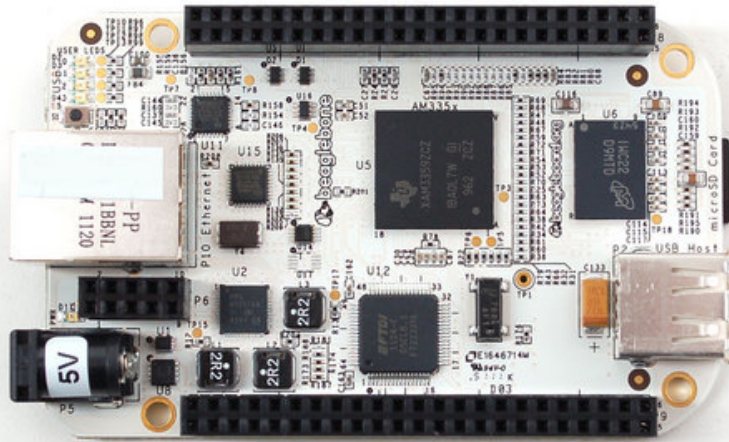




BeagleBone

Created by Ladyada



Last updated on 2013-07-25 06:30:32 PM EDT

Guide Contents

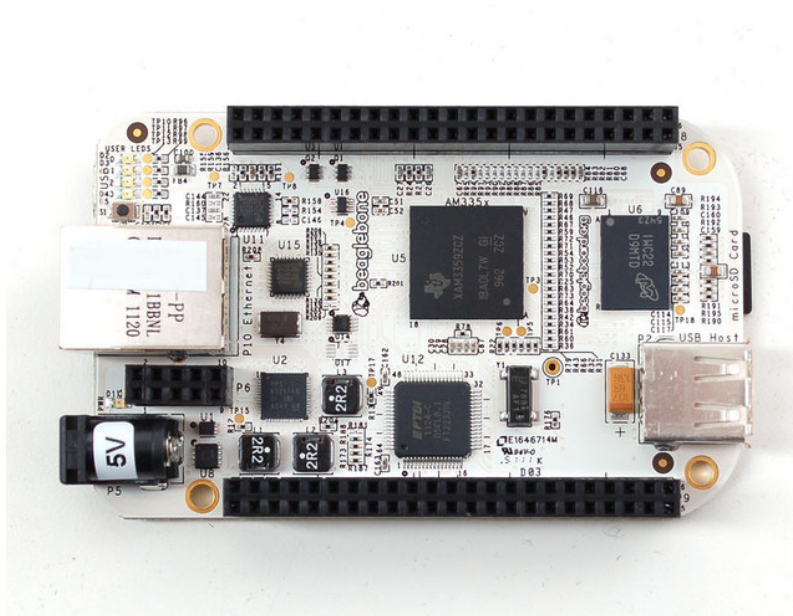
Guide Contents	2
Overview	3
Installing Drivers	4
Download & Install	4
Connect!	5
Ethernet	9
Terminal Software	9
dmesg	11
Ethernet Test	12
WiFi	15
Power and WiFi	15
Driver Install	16
Troubleshooting	19
Buy a BeagleBone	20
Adafruit Forums	21



1-800-972-2225 | www.elexp.com | electron@elexp.com

Overview

New from the fine people who have brought us the Beagle Board, we now have a smaller, lighter, but powerful single board linux computer, Beagle Bone! We like this move to a more compact and integrated SBC. For example, there is onboard Ethernet and USB host, as well as a USB client interface (a FTDI chip for shell access). It even comes preloaded with Angstrom Linux on the 4 GB microSD card!



The Beagle Bone is a great step up from **microcontrollers** (such as AVR, PIC, ARM Cortex M3, 8051, Propeller, etc) to **microcomputers**. Unlike a microcontroller, where the FLASH, EEPROM, RAM, etc is all in one chip, a microcomputer has them separated out, like a classic computer such as a desktop or laptop machine. The Beagle Bone has a main processor core running at 700MHz, a chunk of 256M DDR RAM, and permanent storage onto a microSD card. This makes for a powerful machine, that has no problems running Linux, a webserver, Python, FTP clients, SSH, etc.

The Bone also has great accessories built in, such as onboard Ethernet with 10/100M connectivity, mini USB port with TTL serial converter, JTAG debugger for advanced hacking, USB A host port for connecting a hub/WiFi/etc, power management IC that keeps the board safe from a misplugged adapter, and tons of 0.1" spaced breakouts

One of the powerful abilities of the Bone is that it has I2C, SPI, and GPIO at a hobbyist-friendly 3.3V level (instead of the more difficult to interface 1.8V) while also running complex applications such as a webserver. This allows for more complex projects that would tax an Arduino.

Installing Drivers

This section will detail how to install drivers for the USB/Serial connection (and the other USB devices) from the Bone onto your Windows computer. We'll try to have more documentation on using the Bone with a Mac & Linux at some point but since so many people use Windows and its tougher to install the drives on Win than other OS's we'll start here!

For this tutorial you will need:

- **Beagle Bone** (<http://adafru.it/513>)

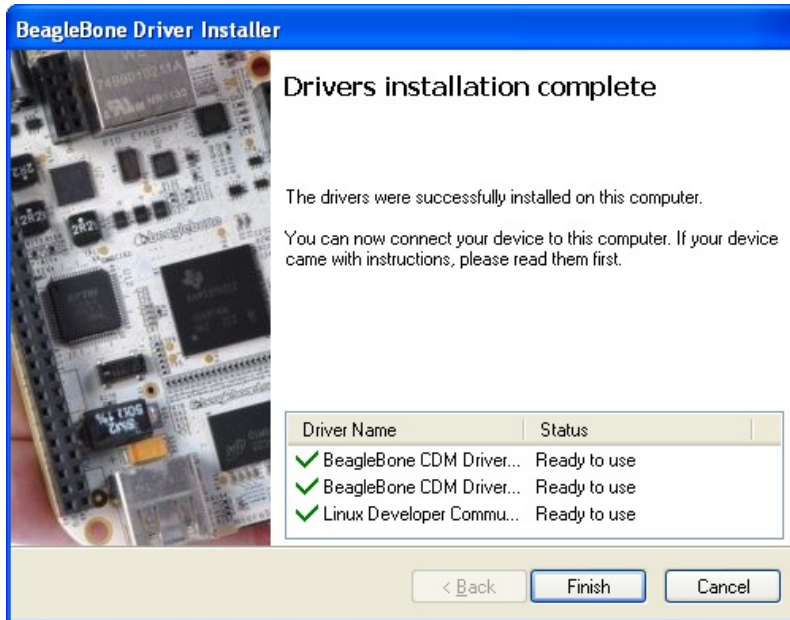
Pick these parts up at the Adafruit shop!

Download & Install

First, we'll install the Windows driver package. [Download this link to BONE_DRV.exe](#) (<http://adafru.it/aLL>) and double click it.

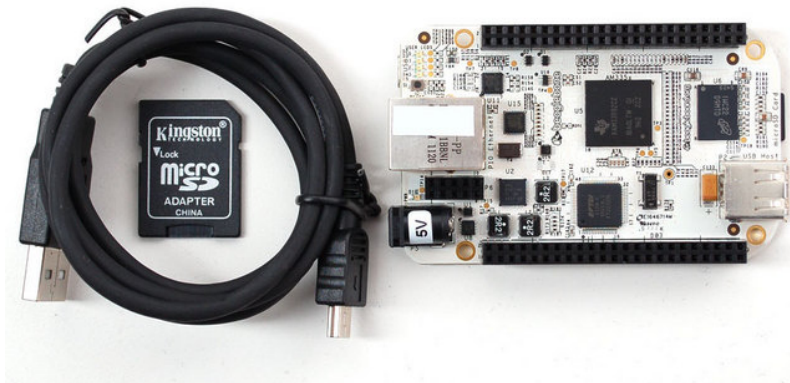
When prompted/warned about the software, click **Continue Anyways** - you'll need to do it 3 times - once for each driver.



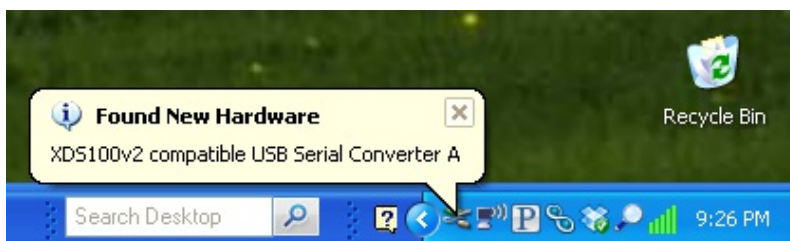


Connect!

Start by opening up your Bone packaging, and finding the MiniB USB cable



Plug the miniB side into the Bone, and the A side into your Windows computer. You'll see a popup saying the computer found a USB serial converter.



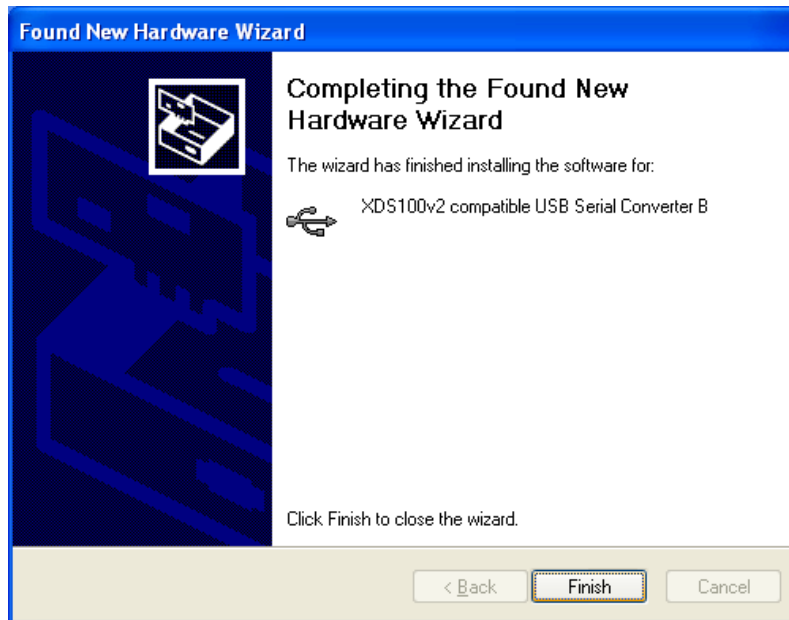
And then an install popup. Click **Install the software automatically** and **Next**.



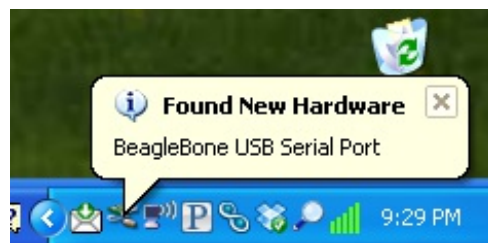
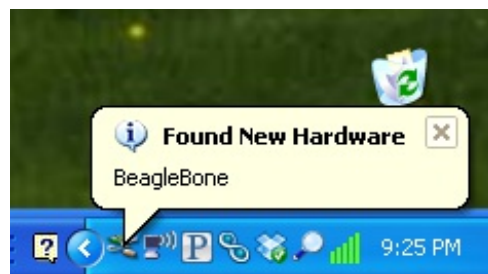
Click **Continue Anyway** when it warns you.



You should finish successfully.

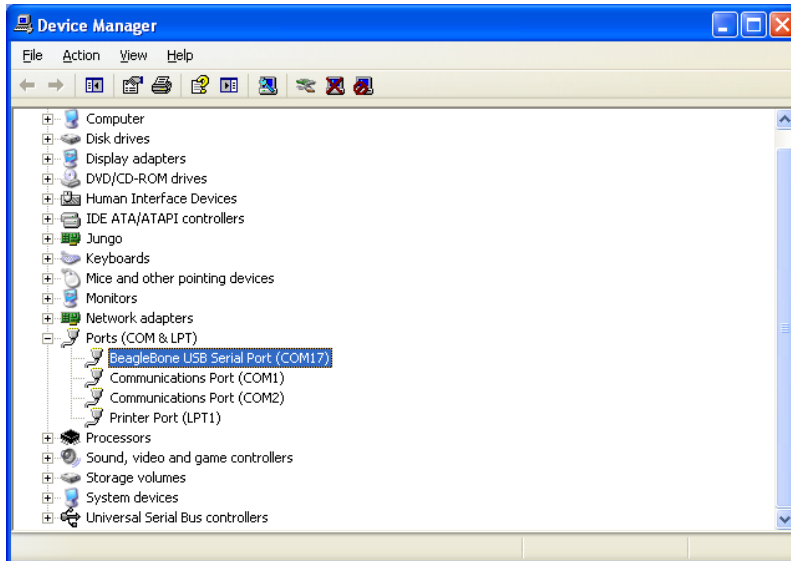


Next you'll go through the same process for the Disk Drive and Beaglebone devices.



Follow the same instructions, installing **Automatically** and clicking **Continue Anyways**.

Finally, you will have the new USB serial port. Go to the Device Manager on your computer to find the name of the COM port. In my case its **COM17**.



That's it, you've installed the drivers! Next up we'll connect via serial and log in.

Ethernet

This mini tutorial will show you how to connect to the Bone via the serial connection to determine the IP address, test the network connection and DNS. You'll need to know the COM serial port address, see the [Drivers](http://adafru.it/aLM) (<http://adafru.it/aLM>) tutorial on how to determine the COM and install drivers.

For this tutorial you will need:

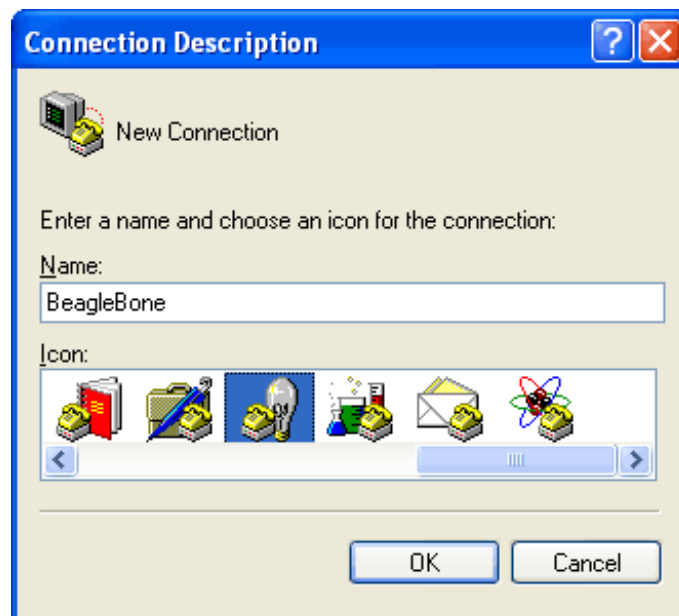
- **Beagle Bone** (<http://adafru.it/513>)
- **Ethernet Cable** (<http://adafru.it/730>)

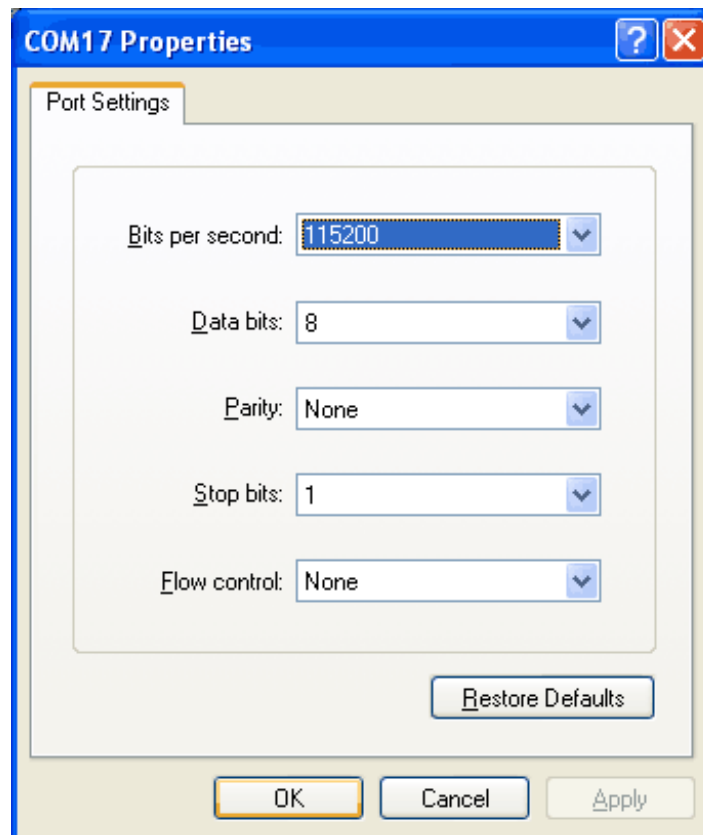
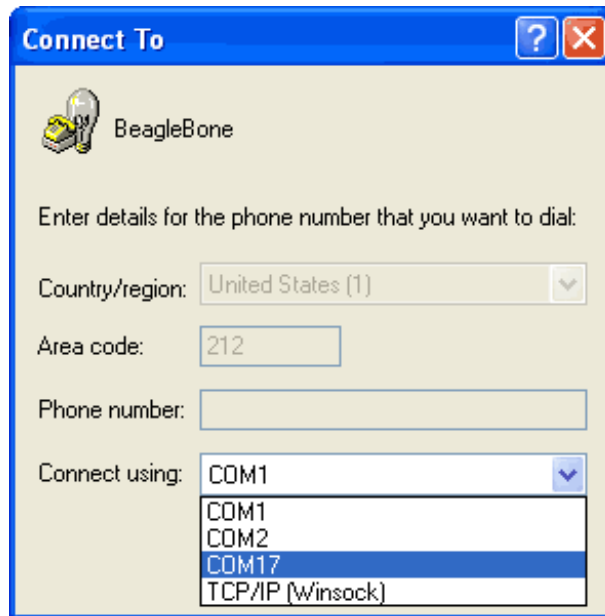
Pick these parts up at the Adafruit shop!

Terminal Software

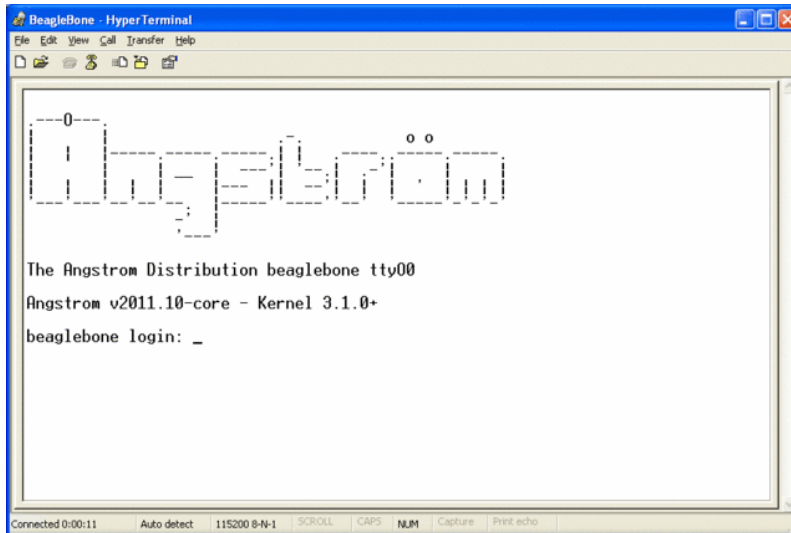
To connect via the USB cable, you'll need a terminal program. Built into Windows is Hyperterm. You can google around to find another good terminal program.

Connect to the Bone's COM port at 115200 baud, 8 bit, No parity, 1 stop bit, no flow control.

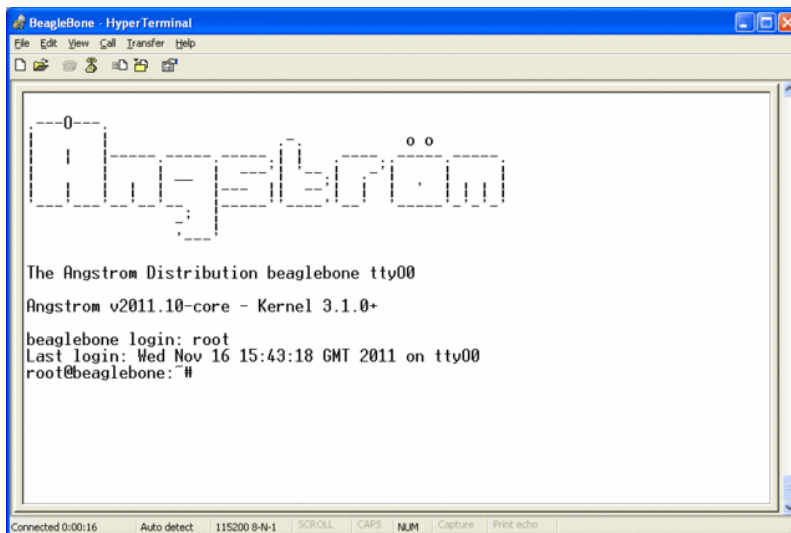




Hit return a few times, to show the login screen.



Log in with the user name **root** and no password.



That's it you're logged in!

dmesg

Now we can try out the Ethernet connection. Plug a standard straight-through cable from the Bone to your Ethernet router.

Our favorite tool is **dmesg** - this will tell you all the system messages, such as what hardware was found. Type **dmesg** and hit return at the **root@beaglebone: ~#** prompt.

```
BeagleBone HyperTerminal
File Edit View Call Transfer Help
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 3.1.0+ (koen@dominion) (gcc version 4.5.4 20110917
(prerelease) (GCC) ) #1 Tue Nov 15 15:51:15 CET 2011
[ 0.000000] CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=50c53c7f
[ 0.000000] CPU: VIPT nonaliasing data cache, VIPT aliasing instruction cache

[ 0.000000] Machine: am335xevm
[ 0.000000] Memory policy: ECC disabled, Data cache writeback
[ 0.000000] On node 0 totalpages: 65536
[ 0.000000] free_area_init_node: node 0, pgdat c047b7d0, node_mem_map c04c100
0
[ 0.000000] Normal zone: 512 pages used for memmap
[ 0.000000] Normal zone: 0 pages reserved
[ 0.000000] Normal zone: 65024 pages, LIFO batch:15
[ 0.000000] AM335X ES1.0 (neon)
[ 0.000000] pcpu-alloc: s0 r0 d32768 u32768 alloc=1*32768
[ 0.000000] pcpu-alloc: [0] 0
[ 0.000000] Built 1 zonelists in Zone order, mobility grouping on. Total pag
es: 65024
[ 0.000000] Kernel command line: console=tty00,115200n8 run_hardware_tests qu
iet root=/dev/mmcblk0p2 ro rootfstype=ext4 rootwait ip=none
[ 0.000000] PID hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.000000] Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
[ 0.000000] Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)

Connected 0:01:03 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo
```

```
BeagleBone HyperTerminal
File Edit View Call Transfer Help
[ 13.647629] CPSW phy found : id is : 0x7c0f1
[ 13.654577] PHY 0:01 not found
[ 13.680633] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 16.050720] gadget: Mass Storage Function, version: 2009/09/11
[ 16.050750] gadget: Number of LUNs=1
[ 16.050782] lun0: LUN: removable file: /dev/mmcblk0p1
[ 16.050837] gadget: Mass Storage Gadget, version: 2009/09/11
[ 16.050861] gadget: userspace failed to provide iSerialNumber
[ 16.050895] gadget: g_mass_storage ready
[ 16.050934] musb-hdrc musb-hdrc.0: MUSB HDRC host driver
[ 16.051028] musb-hdrc musb-hdrc.0: new USB bus registered, assigned bus numbe
r 2
[ 16.051169] usb usb2: New USB device found, idVendor=1d6b, idProduct=0002
[ 16.051191] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=
1
[ 16.051210] usb usb2: Product: MUSB HDRC host driver
[ 16.051225] usb usb2: Manufacturer: Linux 3.1.0+ musb-hcd
[ 16.051239] usb usb2: SerialNumber: musb-hdrc.0
[ 16.052133] hub 2-0:1.0: USB hub found
[ 16.052167] hub 2-0:1.0: 1 port detected
[ 16.391375] gadget: high speed config #1: Linux File-Backed Storage
[ 16.641109] PHY: 0:00 - Link is Up - 100/Full
[ 16.641336] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
root@beaglebone:~#

Connected 0:00:26 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo
```

As you can see the last part of boot up is to bring the ethernet connection **eth0** up.

Ethernet Test

You can verify the ethernet connection by typing in **ifconfig -a**

```
BeagleBone - HyperTerminal
File Edit View Call Transfer Help
[ 16.641109] PHY: 0:00 - Link is Up - 100/Full
[ 16.641336] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
root@beaglebone:~#
root@beaglebone:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr D4:94:A1:52:24:98
          inet addr:10.0.1.24  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::d694:a1ff:fe52:2498/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:41 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18504 (18.0 KiB)  TX bytes:7866 (7.6 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:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:280 (280.0 B)  TX bytes:280 (280.0 B)

root@beaglebone:~#
Connected 0:01:22  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

You can see under **inet addr:** the internet address of the Bone - it uses DHCP to automatically get an IP address and this is what the router gave us back. If you don't see anything, try rebooting the system by typing in **reboot** and hitting return. Make sure your Ethernet cable is well connected to both the Bone and the router.

Now you can test the outgoing connection. Type in **ping 18.70.0.160** and hit return.

```
BeagleBone - HyperTerminal
File Edit View Call Transfer Help
root@beaglebone:~# ping 18.70.0.160
PING 18.70.0.160 (18.70.0.160) 56(84) bytes of data:
64 bytes from 18.70.0.160: icmp_req=1 ttl=50 time=37.0 ms
64 bytes from 18.70.0.160: icmp_req=2 ttl=50 time=37.6 ms
64 bytes from 18.70.0.160: icmp_req=3 ttl=50 time=38.3 ms
^C
--- 18.70.0.160 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 37.005/37.657/38.356/0.574 ms
root@beaglebone:~#
Connected 0:01:46  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

If it works, you'll see the above. You can type Control-C to cancel.

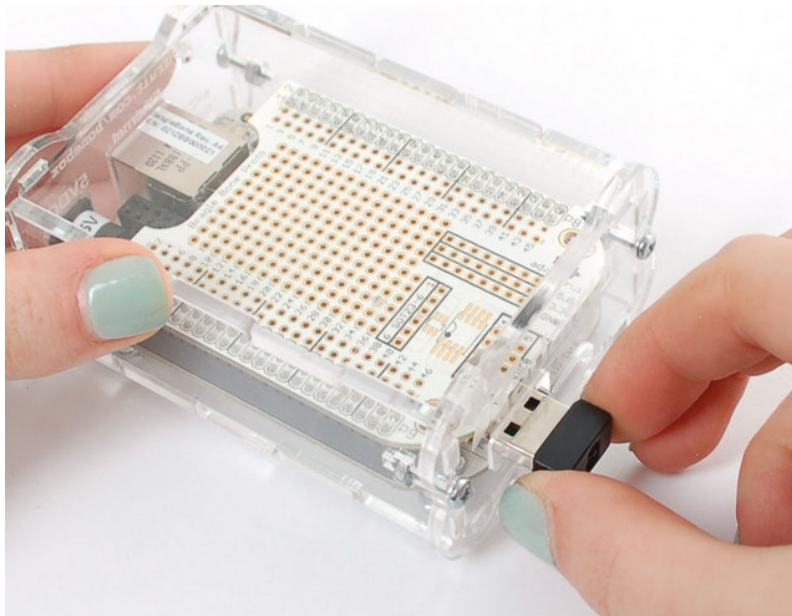
Next you can test the DNS system, by pinging www.google.com (<http://adafruit.it/aLN>), which should also succeed.

```
BeagleBone - HyperTerminal
File Edit View Call Transfer Help
root@beaglebone:~# ping www.google.com
PING www.l.google.com (173.194.73.106) 56(84) bytes of data:
64 bytes from vb-in-f106.1e100.net (173.194.73.106): icmp_req=1 ttl=45 time=33.4 ms
64 bytes from vb-in-f106.1e100.net (173.194.73.106): icmp_req=2 ttl=45 time=33.8 ms
64 bytes from vb-in-f106.1e100.net (173.194.73.106): icmp_req=3 ttl=45 time=35.3 ms
64 bytes from vb-in-f106.1e100.net (173.194.73.106): icmp_req=4 ttl=44 time=40.1 ms
64 bytes from vb-in-f106.1e100.net (173.194.73.106): icmp_req=5 ttl=45 time=36.6 ms
^C
--- www.l.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 33.449/35.896/40.167/2.420 ms
root@beaglebone:~# _
Connected 0:02:32  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```


WiFi

This tutorial may not work on beaglebone black or the latest versions of Angstrom (kernel 3.8+) due to changes in the OS. We're working on a new/updated tutorial, but we don't have an ETA - we'll post it as soon as we can! Thank you for your patience.

Now that you have your Bone up and running, and Ethernet works, wouldn't it be nice to get rid of that Ethernet cable? Yeah, let's go WiFi! This tutorial is specifically for the verified **WiFi adapter for Beagle Bone** (<http://adafru.it/814>) adapter in the Adafruit shop. It will not work with other WiFi adapters, as they all have different chipsets!



For this tutorial you will need:

- **Beagle Bone** (<http://adafru.it/513>)
- **WiFi adapter** (<http://adafru.it/814>)
- **5V 2000mA Power Adapter** (<http://adafru.it/276>)

Pick these parts up at the Adafruit shop!

Power and WiFi

The BeagleBone has the neat ability to power itself just through the mini USB port. However, this can cause some problems because the USB port cannot supply enough power for BOTH the Bone and a WiFi adapter.

An external power supply is required to use WiFi, due to the power requirements. Flaky behavior, crashes, etc will result if you do not plug in a 5V 2000mA adapter! If you're still having problems, try an external powered USB hub!

Driver Install

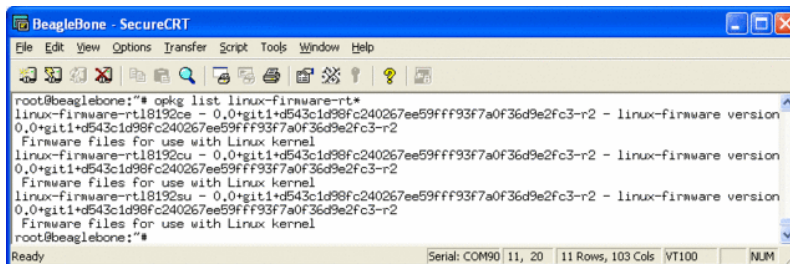
You'll need to have [Internet connectivity using Ethernet \(http://adafru.it/ckU\)](http://adafru.it/ckU) , and also be logged into the terminal to install the WiFi adapter's driver, so make sure to complete those tutorials first!

While logged in with Internet working, run **opkg update**

Then run **mkdir /home/root/tmp** to make a new temp directory then run **opkg -t /home/root/tmp upgrade**

```
root@beaglebone:~# opkg update
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/base/Packages.gz.
Inflating http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/base/Packages.gz.
Updated list of available packages in /var/lib/opkg/lists/base.
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/machine/beaglebone/Packages.gz.
Inflating http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/machine/beaglebone/Packages.gz.
Updated list of available packages in /var/lib/opkg/lists/beaglebone.
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/debug/Packages.gz.
Inflating http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/debug/Packages.gz.
Updated list of available packages in /var/lib/opkg/lists/debug.
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/gstreamer/Packages.gz.
Inflating http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/gstreamer/Packages.gz.
Updated list of available packages in /var/lib/opkg/lists/gstreamer.
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/all/Packages.gz.
Inflating http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/all/Packages.gz.
Updated list of available packages in /var/lib/opkg/lists/no-arch.
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/perl/Packages.gz.
Inflating http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/perl/Packages.gz.
Updated list of available packages in /var/lib/opkg/lists/perl.
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/python/Packages.gz.
Inflating http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/python/Packages.gz.
Updated list of available packages in /var/lib/opkg/lists/python.
root@beaglebone:~# _
```

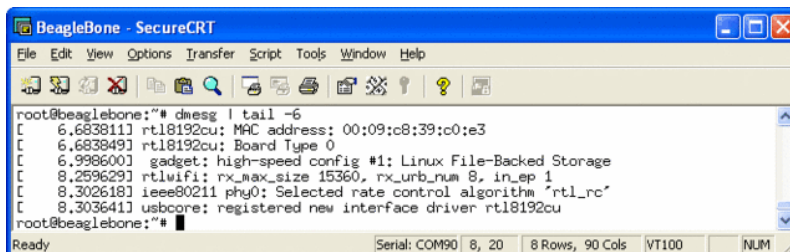
then type in **opkg list 'linux-firmware-rt*'** and hit return.



```
BeagleBone - SecureCRT
File Edit View Options Transfer Script Tools Window Help
root@beaglebone:~# opkg list linux-firmware-rt*
linux-firmware-rt18192ce - 0.0+git1+d543c1d98fc240267ee59fff93f7a0f36d9e2fc3-r2 - linux-firmware version
0.0+git1+d543c1d98fc240267ee59fff93f7a0f36d9e2fc3-r2
Firmware files for use with Linux kernel
linux-firmware-rt18192cu - 0.0+git1+d543c1d98fc240267ee59fff93f7a0f36d9e2fc3-r2 - linux-firmware version
0.0+git1+d543c1d98fc240267ee59fff93f7a0f36d9e2fc3-r2
Firmware files for use with Linux kernel
linux-firmware-rt18192su - 0.0+git1+d543c1d98fc240267ee59fff93f7a0f36d9e2fc3-r2 - linux-firmware version
0.0+git1+d543c1d98fc240267ee59fff93f7a0f36d9e2fc3-r2
Firmware files for use with Linux kernel
root@beaglebone:~#
```

Finally type in **opkg install linux-firmware-rt18192cu** and press return. Plug in the WiFi dongle, then type in **reboot** and return to reboot the machine.

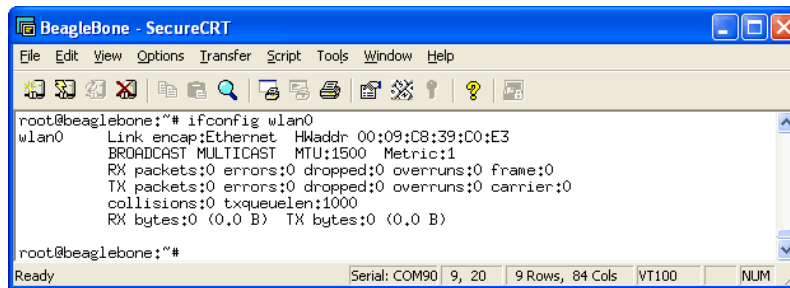
Now that its rebooted, check **dmesg** - you should see the following



```
BeagleBone - SecureCRT
File Edit View Options Transfer Script Tools Window Help
root@beaglebone:~# dmesg | tail -6
[ 6.683811] rt18192cu: MAC address: 00:09:c8:39:c0:e3
[ 6.683849] rt18192cu: Board Type 0
[ 6.998600] gadget: high-speed config #1: Linux File-Backed Storage
[ 8.259629] rtlwifi: rx_max_size 15360, rx_urb_num 8, in_ep 1
[ 8.302618] ieee80211 phy0: Selected rate control algorithm 'rtl_rc'
[ 8.303641] usbcore: registered new interface driver rt18192cu
root@beaglebone:~#
```

And if you type in **ifconfig wlan0** there should be a link, it wont be connected yet so there's a

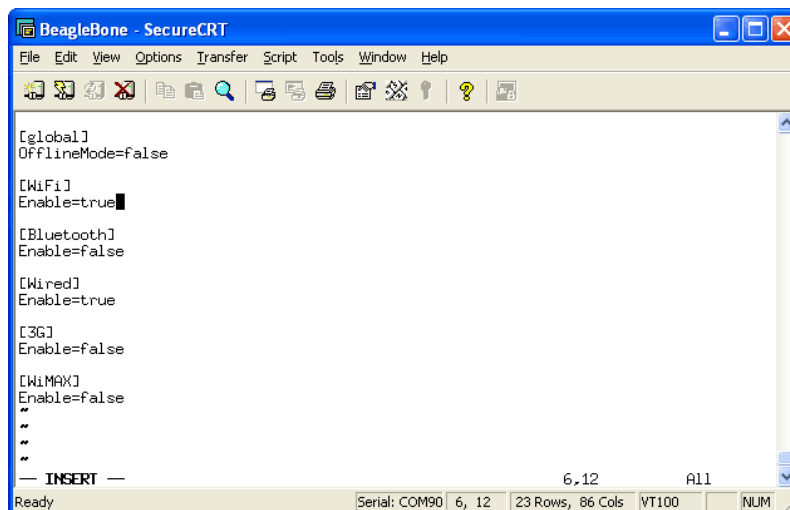
lot of 0's and no **inet addr**



```
root@beaglebone:~# ifconfig wlan0
wlan0    Link encap:Ethernet  HWaddr 00:09:08:39:C0:E3
          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)

root@beaglebone:~#
```

Now we can set up the connection manager to automatically manage the wifi. Edit **/var/lib/connman/settings** (I use vi but nano is also installed) and change WiFi from false to true, save it.



```
[global]
OfflineMode=false

[WiFi]
Enable=true

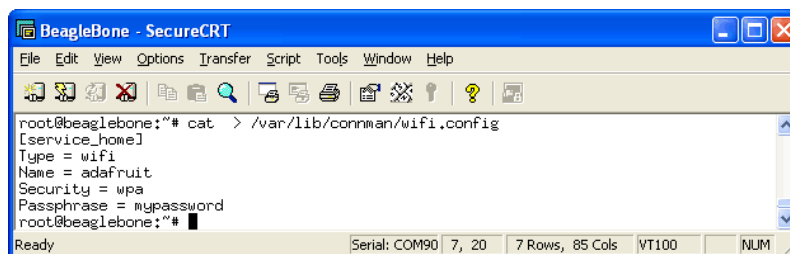
[Bluetooth]
Enable=false

[Wired]
Enable=true

[3G]
Enable=false

[WiMAX]
Enable=false
~
~
~
-- INSERT --
```

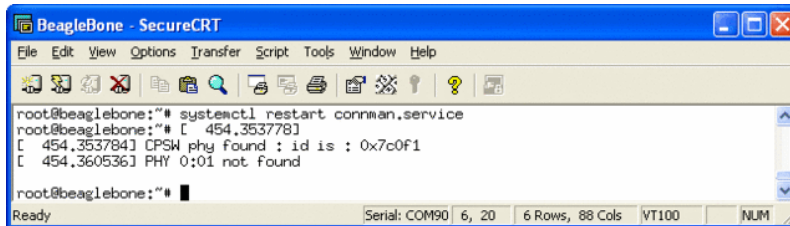
Create a file **/var/lib/connman/wifi.config** with your settings as shown below, starting with the **[service_home]** line and with a return after the **Passphrase** line, of course this should match your home network, not the adafruit one!



```
root@beaglebone:~# cat > /var/lib/connman/wifi.config
[service_home]
Type = wifi
Name = adafruit
Security = wpa
Passphrase = mypassword
root@beaglebone:~#
```

Restart connman to get it to accept the new settings:

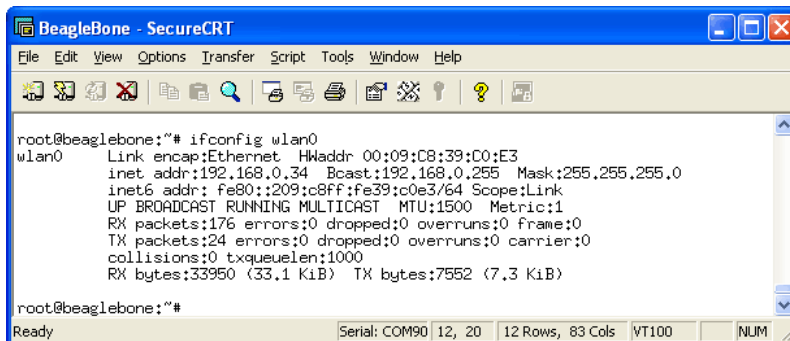
root@beaglebone:~# systemctl restart connman.service



```
BeagleBone - SecureCRT
File Edit View Options Transfer Script Tools Window Help
root@beaglebone:~# systemctl restart connman.service
root@beaglebone:~# [ 454.353784]
[ 454.353784] CPSW phy found : id is : 0x7c0f1
[ 454.360536] PHY 0:01 not found
root@beaglebone:~#
```

After less than 30 seconds or so, you should be connected:

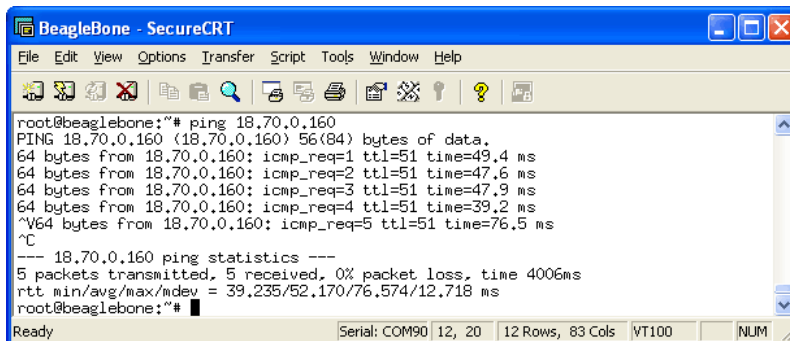
root@beaglebone: ~# ifconfig wlan0



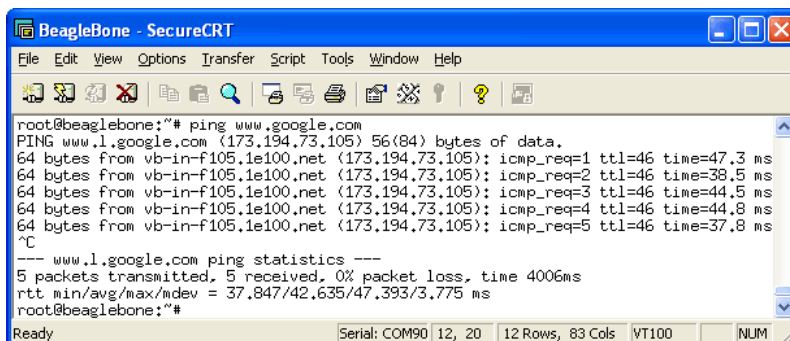
```
BeagleBone - SecureCRT
File Edit View Options Transfer Script Tools Window Help
root@beaglebone:~# ifconfig wlan0
wlan0  Link encap:Ethernet HWaddr 00:09:C8:39:C0:E3
        inet addr:192.168.0.34 Bcast:192.168.0.255 Mask:255.255.255.0
        inet6 addr: fe80::209:c8ff:fe39:c0e3/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:176 errors:0 dropped:0 overruns:0 frame:0
        TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:33950 (33.1 KiB)  TX bytes:7552 (7.3 KiB)

root@beaglebone:~#
```

There should now be an **inet addr**. You can then test pinging an IP address and a domain name.

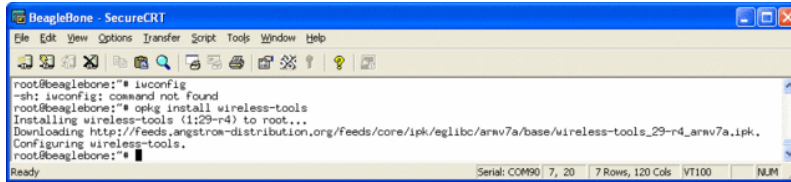


```
BeagleBone - SecureCRT
File Edit View Options Transfer Script Tools Window Help
root@beaglebone:~# ping 18.70.0.160
PING 18.70.0.160 (18.70.0.160) 56(84) bytes of data:
64 bytes from 18.70.0.160: icmp_req=1 ttl=51 time=49.4 ms
64 bytes from 18.70.0.160: icmp_req=2 ttl=51 time=47.6 ms
64 bytes from 18.70.0.160: icmp_req=3 ttl=51 time=47.9 ms
64 bytes from 18.70.0.160: icmp_req=4 ttl=51 time=39.2 ms
^V64 bytes from 18.70.0.160: icmp_req=5 ttl=51 time=76.5 ms
^C
--- 18.70.0.160 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 39.235/52.170/76.574/12.718 ms
root@beaglebone:~#
```

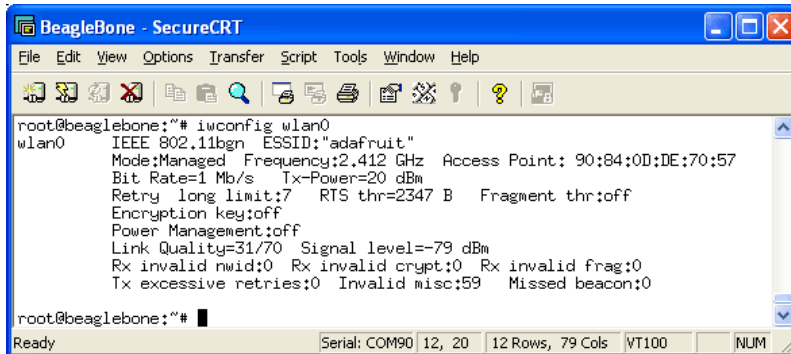


```
BeagleBone - SecureCRT
File Edit View Options Transfer Script Tools Window Help
root@beaglebone:~# ping www.google.com
PING www.l.google.com (173.194.73.105) 56(84) bytes of data:
64 bytes from vb-in-f105.1e100.net (173.194.73.105): icmp_req=1 ttl=46 time=47.3 ms
64 bytes from vb-in-f105.1e100.net (173.194.73.105): icmp_req=2 ttl=46 time=38.5 ms
64 bytes from vb-in-f105.1e100.net (173.194.73.105): icmp_req=3 ttl=46 time=44.5 ms
64 bytes from vb-in-f105.1e100.net (173.194.73.105): icmp_req=4 ttl=46 time=44.8 ms
64 bytes from vb-in-f105.1e100.net (173.194.73.105): icmp_req=5 ttl=46 time=37.8 ms
^C
--- www.l.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 37.847/42.635/47.393/3.775 ms
root@beaglebone:~#
```

Finally, if you want more detailed information about your link you can **opkg install wireless-tools** to get the **iwconfig** command, which will give you tons of details.



```
root@beaglebone:~# iwconfig
=sh: iwconfig: command not found
root@beaglebone:~# opkg install wireless-tools
Installing wireless-tools (1:29-r4) to root...
Downloading http://feeds.angstrom-distribution.org/feeds/core/ipk/eglbc/armv7a/base/wireless-tools_29-r4_armv7a.ipk.
Configuring wireless-tools.
root@beaglebone:~#
```



```
root@beaglebone:~# iwconfig wlan0
wlan0 IEEE 802.11bgn ESSID:"adafruit"
Mode:Managed Frequency:2.412 GHz Access Point: 90:84:0D:DE:70:57
Bit Rate=1 Mb/s Tx-Power=20 dBm
Retry long limit:7 RTS thr=2347 B Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=31/70 Signal level=-79 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:59 Missed beacon:0

root@beaglebone:~#
```

Troubleshooting

If you get an error device descriptor read/64, error -71, reboot and stop the boot process with the space bar. Then add the following boot option with the follow at the U-Boot prompt

```
setenv bootargs irqpoll RETURN
```

```
boot RETURN
```



1-800-972-2225 | www.elexp.com | electron@elexp.com