



Setting up WiFi with BeagleBone Black

Created by Tony DiCola



<https://learn.adafruit.com/setting-up-wifi-with-beaglebone-black>

Last updated on 2023-08-29 02:36:55 PM EDT

Table of Contents

Overview	3
Hardware	4
• HDMI Port Interference	
• Power	
Configuration	7
• Kernel Upgrade	
• WiFi Reset Service	
• WiFi Configuration	

Overview



Have you been pulling your hair out trying to get WiFi working consistently on a BeagleBone Black? Don't worry there are some easily solved problems that could be causing your WiFi grief! For example did you know the HDMI port can interfere with small WiFi adapter antennas? Or that you can upgrade the Linux kernel to improve WiFi driver stability? Follow this guide to learn how to avoid common pitfalls and configure your BeagleBone Black to work with a USB WiFi adapter!

Before you get started make sure you're running the [latest official Debian image \(\)](#) on your BeagleBone Black. It's very important to use the Debian operating system on your BeagleBone Black to follow this guide! If you aren't familiar with how to install an operating system on the device, check out this handy [BeagleBone Black operating system install guide \(\)](#).

Also if you are new to the BeagleBone Black you will want to familiarize yourself with how to connect to it and interact with the command shell using SSH. [This guide will show you the basics of accessing the BeagleBone Black with SSH \(\)](#).

Hardware



To setup WiFi with your BeagleBone Black you'll need a USB WiFi adapter that is compatible with the BeagleBone Black's Debian operating system. Be sure to read the [BeagleBone Black wiki's list of known working WiFi adapters \(\)](#) and consider using a device from that list.

Unfortunately the choice of WiFi adapter for the BeagleBone Black is a little more complicated compared to other Linux boards like the Raspberry Pi. The BeagleBone Black's official Debian image uses an old version 3.8 of the Linux kernel compared to more recent versions like 3.15+. Since WiFi drivers are typically included in the Linux kernel source, running an old kernel means running old and potentially buggy WiFi adapter drivers.

Out of the box the 3.8 kernel seems to have problems with popular WiFi adapters, like the Realtek-based adapters in the Adafruit store. In particular you might experience problems like the WiFi adapter randomly not connecting to a wireless network on boot. I'll show later in the guide that a simple script to reset the wireless adapter on boot greatly improves reliability of Realtek and other adapters on the BeagleBone Black.

If you're a more experienced Linux user you might be wondering if you can upgrade the Linux kernel to a later version like 3.14 or 3.15 to get more reliable WiFi drivers.

You can, but you need to be careful because later kernels right now don't support the same functionality, like [device tree overlays \(\)](#), as the 3.8 kernel. Kernel support beyond 3.8 is improving every day so you should search and consult the [BeagleBone](#)

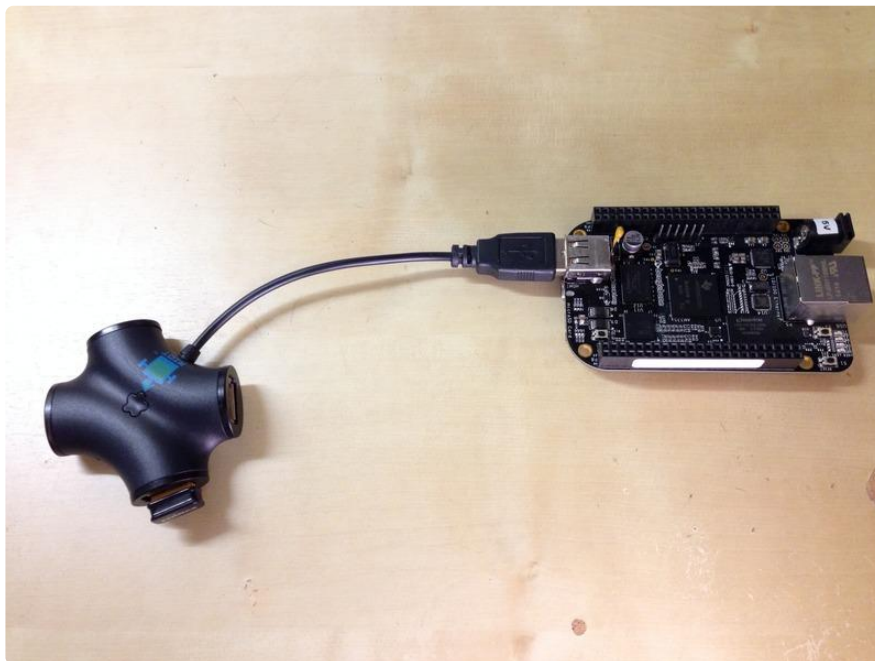
[Black support group \(\)](#) for the latest information on kernel stability and feature support if you want to use a newer kernel. I'll show later in the guide how to make sure you're using the latest 3.8 version kernel so you're getting the most up to date drivers.

HDMI Port Interference

One common issue with WiFi on the BeagleBone Black is poor performance from WiFi adapters with small internal antennas. The problem is that [the ground and power planes of the HDMI port are right below the USB port and dampen the WiFi signal \(\)](#).

This will manifest as poor signal quality and reliability with small USB WiFi adapters that are very close to the USB port.

There are two ways to deal with this interference issue, one is to use a small [USB extension cable / adapter \(http://adafru.it/974\)](#) or [small USB hub \(http://adafru.it/961\)](#) (even an unpowered hub would work fine) to move the WiFi adapter farther away from the BeagleBone Black. For example the photo below shows a small hub that moves a tiny WiFi adapter away from the BeagleBone Black:



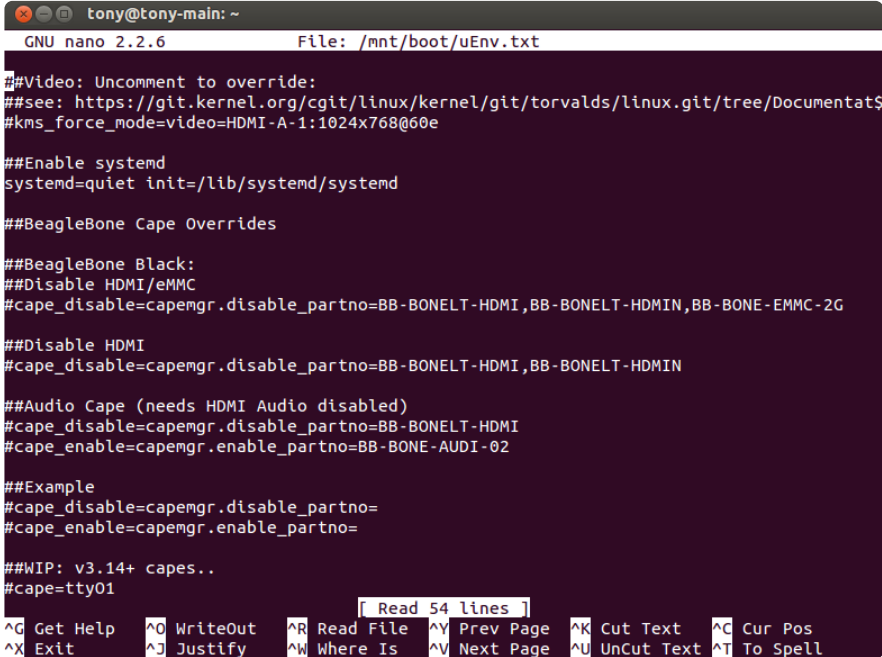
A second option is to completely disable the HDMI port by [manipulating the device tree \(\)](#). This option is only possible if you do not need to output any video or audio signal over the HDMI port!

Of the two options I would stick with the first, moving the adapter away from the board, since it's simpler and will still let you use the HDMI port. However if you can't move the WiFi adapter away you can follow the steps below to disable HDMI.

First connect your BeagleBone Black's USB port to your computer and [connect to it with SSH \(\)](#). Then mount and edit the uEnv.txt file that controls boot configuration by executing:

```
mkdir /mnt/boot
mount /dev/mmcblk0p1 /mnt/boot
nano /mnt/boot/uEnv.txt
```

You should see the nano text editor appear with the contents of the uEnv.txt file like this:



```
tony@tony-main: ~
GNU nano 2.2.6 File: /mnt/boot/uEnv.txt
##Video: Uncomment to override:
##see: https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/DocumentatS
#kms_force_mode=video=HDMI-A-1:1024x768@60e

##Enable systemd
systemd=quiet init=/lib/systemd/systemd

##BeagleBone Cape Overrides

##BeagleBone Black:
##Disable HDMI/eMMC
#cape_disable=capemgr.disable_partno=BB-BONELT-HDMI, BB-BONELT-HDMIN, BB-BONE-EMMC-2G

##Disable HDMI
#cape_disable=capemgr.disable_partno=BB-BONELT-HDMI, BB-BONELT-HDMIN

##Audio Cape (needs HDMI Audio disabled)
#cape_disable=capemgr.disable_partno=BB-BONELT-HDMI
#cape_enable=capemgr.enable_partno=BB-BONE-AUDI-02

##Example
#cape_disable=capemgr.disable_partno=
#cape_enable=capemgr.enable_partno=

##WIP: v3.14+ capes..
#cape=tt01

Read 54 lines
^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^N Next Page ^U UnCut Text ^T To Spell
```

Notice the lines in the middle that mention disabling HDMI:

```
##Disable HDMI
#cape_disable=capemgr.disable_partno=BB-BONELT-HDMI, BB-BONELT-HDMIN
```

Remove the # in front of the cape_disable command so it looks like:

```
##Disable HDMI
cape_disable=capemgr.disable_partno=BB-BONELT-HDMI, BB-BONELT-HDMIN
```

Be careful to edit the correct line! If you disable the HDMI/eMMC line you might disable booting from the internal flash memory of the BeagleBone Black.

Save the file by pressing Ctrl-O and then Enter. Finally quit nano by pressing Ctrl-X. Reboot your BeagleBone Black and it should have the HDMI port disabled.

If you need to enable the HDMI port again just edit the uEnv.txt file with the same procedure above and add the # comment back in front of the disable HDMI line.

Power

Finally make sure you are powering your BeagleBone Black with a sufficient power supply to support your WiFi adapter. Most computer or laptop USB ports do not supply enough power to support a BeagleBone Black and WiFi adapter!

Make sure you're using a 5 volt power supply with at least 1 amp or more of current. You can plug a simple [external 5V power supply \(http://adafru.it/276\)](http://adafru.it/276) right into the BeagleBone Black's barrel jack.

Don't skimp on powering your device properly or else you will have instability and poor WiFi performance!

Configuration

Once you have the hardware setup you're ready to configure the BeagleBone Black to use your WiFi adapter.

Kernel Upgrade

First you will want to make sure you're running the very latest version of the 3.8 series kernel. Like mentioned in the previous page, you want to use the most recent Linux kernel to get the most recent and stable WiFi drivers. Luckily you don't need to be a Linux expert to upgrade the kernel because a script is included with your BeagleBone Black's Debian installation to automatically upgrade to the latest stable 3.8 version kernel.

In my testing I found upgrading the BeagleBone Black's kernel was mandatory to get reliable WiFi performance. Don't skip this step, be sure to upgrade the kernel!

Make sure your device is connected to the internet (through the ethernet port for example), power it on, and connect to the terminal with SSH. Run the following commands to start the kernel upgrade:

```
cd /opt/scripts/tools/  
./update_kernel.sh
```

You should see the script download a kernel image and perform the installation automatically. After a few minutes the upgrade will complete and tell you to reboot your device. Follow the instructions and reboot by executing the command:

```
reboot
```

After the device reboots the kernel upgrade is complete!

WiFi Reset Service

The next step is to install a small script that will reset the WiFi interface by bringing it down and back up again automatically on boot. With the latest 3.8 kernel I found this reset script was necessary to get reliable performance with Realtek and some Atheros WiFi adapters, so don't skip installing it! Later kernels or more stable adapters might not require the reset service, but it can't hurt to install it for them too.

To install the service connect again to the device in a terminal with SSH and execute these commands:

```
cd ~
ntpddate -b -s -u pool.ntp.org
apt-get update &&& apt-get install git
git clone https://github.com/adafruit/wifi-reset.git
cd wifi-reset
chmod +x install.sh
./install.sh
```

After executing the install.sh script you should see a response like the following:

```
Installing wifi reset service to /opt/wifi-reset.
Installing systemd service to run at boot.
Enabling systemd service.
```

If you see an error message, go back and carefully check the commands above were executed.

That's all you need to do to install the wifi reset service!

Note that if you'd ever like to disable the WiFi reset service you can execute this command to do so:

```
systemctl disable wifi-reset.service
```


WiFi Configuration

Finally it's time to configure the wireless connection for your BeagleBone Black. To start, power down the device completely and insert the WiFi adapter into the USB port. Remember the HDMI port can cause interference so you might need to use a small extension or hub to move your adapter away from the board!

Power up your BeagleBone Black and connect to it in a terminal with SSH, then run the following command to list any wireless interfaces available:

```
iwconfig
```

You should see a response with a wlan0 name and some details about the wireless adapter's capabilities. For example I see the following response for my adapter:

```
wlan0 IEEE 802.11bgn ESSID:off/any
      Mode:Managed Access Point: Not-Associated Tx-Power=0 dBm
      Retry long limit:7 RTS thr=2347 B Fragment thr:off
      Encryption key:off
      Power Management:on

lo no wireless extensions.

eth0 no wireless extensions.

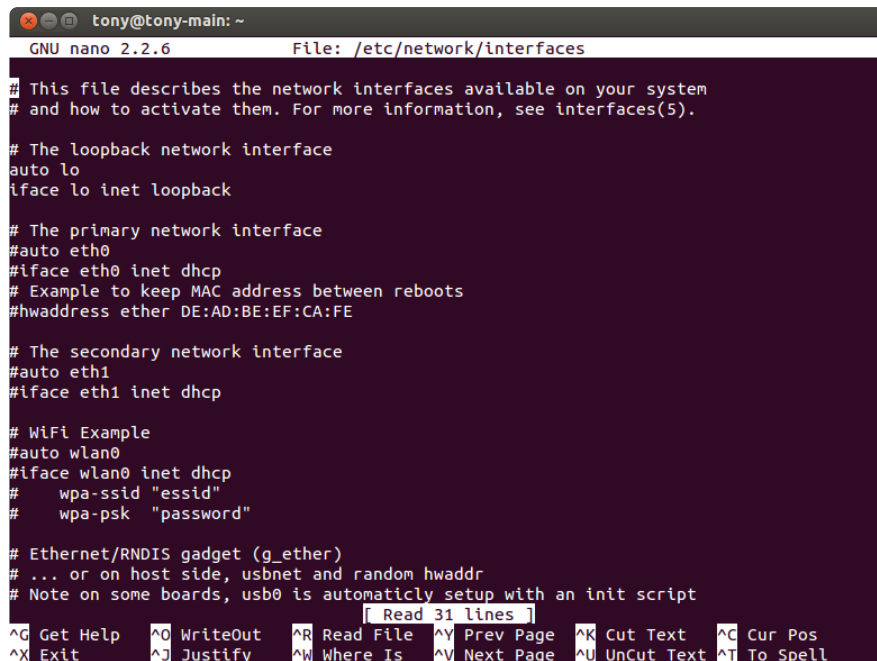
usb0 no wireless extensions.
```

Note that if you see a different name than wlan0, like ra0, take note of that name as it will be used in the configuration later.

Now you'll edit the `/etc/network/interfaces` file to specify WiFi network connection details. Run the following command to open the nano text editor:

```
nano /etc/network/interfaces
```

You should see a file like the following appear:



```
tony@tony-main: ~
GNU nano 2.2.6 File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#auto eth0
#iface eth0 inet dhcp
# Example to keep MAC address between reboots
#hwaddress ether DE:AD:BE:EF:CA:FE

# The secondary network interface
#auto eth1
#iface eth1 inet dhcp

# WiFi Example
#auto wlan0
#iface wlan0 inet dhcp
#   wpa-ssid "ssid"
#   wpa-psk "password"

# Ethernet/RNDIS gadget (g_ether)
# ... or on host side, usbnet and random hwaddr
# Note on some boards, usb0 is automatically setup with an init script

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
```

Notice there's a commented block showing a WiFi configuration example:

```
# WiFi Example
#auto wlan0
#iface wlan0 inet dhcp
#   wpa-ssid "ssid"
#   wpa-psk "password"
```

Configuring most WiFi networks is as easy as uncommenting the lines and filling in the details for your network. For example if I were connecting to a WiFi network with name "adafruit" and password "mypassword" I would edit the configuration to look like:

```
# WiFi Example
auto wlan0
iface wlan0 inet dhcp
    wpa-ssid "adafruit"
    wpa-psk "mypassword"
```

If you're connecting to a more complex WiFi network, such as one with a 'hidden' SSID or that needs a static IP address, you might need to add more to the configuration. Check out the guide on [setting up WiFi with a Raspberry Pi \(\)](#) as the configuration in /etc/network/interfaces is the same on the BeagleBone Black as the Raspberry Pi.

Also be aware that if you found the iwconfig command returned a different interface name, like ra0, you should change all references from wlan0 to the appropriate interface name.

Once you've modified the file, save it by pressing Ctrl-O and Enter, then press Ctrl-X to quit nano.

Now test the connection by running the following to bring up the WiFi connection manually:

```
ifup wlan0
```

Again change wlan0 to the name of your WiFi interface if you found it differed with the iwconfig command.

If the WiFi connection is made with your network you should see an IP address acquired with DHCP:

```
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan0/e8:de:27:11:62:7c
Sending on   LPF/wlan0/e8:de:27:11:62:7c
Sending on   Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 6
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 9
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPOFFER from 192.168.1.1
DHCPACK from 192.168.1.1
bound to 192.168.1.127 -- renewal in 36491 seconds.
```

The exact response might differ, but the important thing is that you receive a DHCPOFFER response and are assigned an IP address.

You can also confirm the BeagleBone Black is connected to the network by running the following command to list network details:

```
ifconfig wlan0
```

If your device is connected to the network it should show connection details with an IP address (192.168.1.127 in my case) like:

```
wlan0      Link encap:Ethernet  HWaddr e8:de:27:11:62:7c
           inet addr:192.168.1.127  Bcast:192.168.1.255  Mask:255.255.255.0
           inet6 addr: fe80::eade:27ff:fe11:627c/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:15413 errors:0 dropped:0 overruns:0 frame:0
           TX packets:3301 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:21575587 (20.5 MiB)  TX bytes:323282 (315.7 KiB)
```

Finally you can test internet access by trying to ping a website such as adafruit.com by executing:

```
ping adafruit.com
```

You should see successful ping responses such as:

```
PING adafruit.com (207.58.139.247) 56(84) bytes of data.  
64 bytes from vps3.ladyada.net (207.58.139.247): icmp_req=1 ttl=52 time=86.1 ms  
64 bytes from vps3.ladyada.net (207.58.139.247): icmp_req=2 ttl=52 time=85.8 ms  
64 bytes from vps3.ladyada.net (207.58.139.247): icmp_req=3 ttl=52 time=87.0 ms  
64 bytes from vps3.ladyada.net (207.58.139.247): icmp_req=4 ttl=52 time=87.8 ms  
64 bytes from vps3.ladyada.net (207.58.139.247): icmp_req=5 ttl=52 time=86.1 ms  
64 bytes from vps3.ladyada.net (207.58.139.247): icmp_req=6 ttl=52 time=86.7 ms  
64 bytes from vps3.ladyada.net (207.58.139.247): icmp_req=7 ttl=52 time=86.5 ms
```

Press Ctrl-C to quit the ping application.

If any of the checks above fail or indicate no network access, carefully check the details in the `/etc/network/interfaces` file are correct for your wireless network and try again.

Now the final test is to reboot your BeagleBone Black to verify the wireless connection is made automatically. Execute the following to reboot the device:

```
reboot
```

Once the device has rebooted, connect to it in an SSH session and run the network connection tests above again. If they show you're connected to the network, congratulations you've successfully configured your BeagleBone Black to work with WiFi!

If the device fails to connect on boot, try running the following to manually reset the WiFi connection:

```
ifdown wlan0  
ifup wlan0
```

Again changing wlan0 to match the name of the interface you found earlier.

If the `ifup` command fails to get an IP address, go back and check the WiFi network details in `/etc/network/interfaces` are correct for your network and try again. If you still can't get a connection to work, try posting details about the WiFi adapter you're using and the error or issue you're hitting on the [BeagleBone Black support group \(\)](#) or the [Adafruit forums \(\)](#).