Modern Linux Wi-Fi Adapter Setup & Configuration Guide

In this How-To guide, we will show you how to configure Wi-Fi on the BeagleBone Green, or any other Linux device running a relatively recent (2018+) kernel using Connman, a lightweight connection manager designed for embedded and IoT devices. It supports Bluetooth, LTE/3G, NFC, and more.

Why did we write Yet Another Wi-Fi Guide™? The other student guides either use older Kernels, still use init.d instead of systemd, required manual editing of files that could break USB to Host ethernet, or just didn't work for us. Our project also required multiple adapters per device, so we walkthrough how we did this.

This document guides the user through:

1. Setting up the Wi-Fi adapter as a client (Connman)
2. Using multiple adapters
    a. Predictable Interface Names (udev)
    b. Connman interface blacklist
    c. systemd wait for interface

Required components:

- Beaglebone
- Wi-Fi Adapter(s)

Connman is installed by default on the BeagleBone Stretch image, but if it is not then you can install it:

```
sudo apt install connman
```

Contents

Hint: Newer versions of systemd (v197+) use "predictable stable interface names". You can tell if this is enabled if your wireless adapters have names like "wlp2s0" or ethernet names like "ens1". To follow this guide, you can either change references to "wlan0" to your interface, or mask udev's .link file to get the default policy by running as root: `ln -s /dev/null /etc/systemd/network/99-default.link`

Client Setup (Connman)

Note: Ensure that the interface has enumerated. You should see a wlan0 or a similar interface in ifconfig

```
debian@beagle:~$ ifconfig
wlan0: flags=-28669<UP,BROADCAST,MULTICAST,DYNAMIC>  mtu 1500
        ether 48:02:2a:53:bd:88  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Connman can be setup interactively as well as through the command line.
Situation: Connecting once to the Wi-Fi
    Recommendation: Interactive mode
Situation: Connecting to different Wi-Fi networks automatically
    Recommendation: Command line, using a script to automate the process

We will be using TELUS0123 as the Wi-Fi that we want to connect to.

Connman interactive mode:

Connman has an interactive mode that works like a command prompt or REPL.

```
debian@beagle:~$ connmanctl

connmanctl> enable Wi-Fi
Enabled Wi-Fi

connmanctl> agent on
Agent registered

connmanctl> scan Wi-Fi
Scan completed for Wi-Fi

connmanctl> services
    TELUS0123     wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk

connmanctl> connect wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk
Agent RequestInput wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory, Alternates=[ WPS ] ]
  WPS = [ Type=wpspin, Requirement=alternate ]
Passphrase? *******

Connected wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk

connmanctl> exit
```

Connman CLI:

Connman can also be configured using regular terminal commands.

```
debian@beagle:~$ connmanctl enable Wi-Fi
Enabled Wi-Fi

debian@beagle:~$ connmanctl scan Wi-Fi
Scan completed for Wi-Fi
```

```
debian@beagle:~$ connmanctl services
  TELUS0123         wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk
```

Write the config file into /var/lib/connman using the service name. Must do this as root

```
echo "[service_wifi_wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk]
  Type = wifi
  Name = TELUS0123
  Passphrase = *******
  Security = psk" |
sudo tee
/var/lib/connman/wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk.config
```

```
debian@beagle:~$ connmanctl services
*AO TELUS0123         wifi_48022a53bd88_4c696e6b73797a7a7a_managed_psk
```

The state *AO means that the Wi-Fi network is associated and connected.

Troubleshooting:

- Scanning and listing the services do not show anything
  - Ensure Wi-Fi is enabled
  - Restart the Wi-Fi adapter
    `sudo ifconfig wlan0 [down/up]`
  - Try restarting connman
    `sudo systemctl restart connman`
- The state is not *AO when running connmanctl services
  - Give it time to authenticate with the Wi-Fi network
  - Check that the correct Wi-Fi password is used
  - Restart the Wi-Fi adapters
  - Restart connman

Predictable Interface Names (udev)

In the case that multiple adapters or interfaces are being used, it is possible to create predictable interface names. This allows different interfaces to be designated for different tasks since Wi-Fi adapters can be used as a client, access point, and monitor (dependent on chipset).

Example: Wi-Fi repeater

Using udev rules, interfaces can be named to perform different tasks. Create a file in `/etc/udev/rules.d/` called `01-wifi.rules`

An example rule file may look like:

```
KERNEL=="wlan*", ATTRS{idProduct}=="5370", ATTRS{idVendor}=="148f",
NAME="BeagleWiFi%n"
KERNEL=="wlan*", ATTR{address}=="48:02:2a:53:bd:88", NAME="BeagleAP"
```

The first rule will rename all adapters with a product id of "5370" and vendor id of "148f" to become BeagleWiFi**X**, where X is the kernel bus number.

The second rule will rename the adapter with the mac address of "48:02:2a:53:bd:88" to BeagleAP.

These attributes can be found by running:

```
udevadm info -a /sys/class/net/[Interface]
```

More information can be found on https://wiki.debian.org/udev

Connman Interface Blacklist

If using udev rules, it is not a bad idea to combine it with connman blacklist rules. This prevents connman from attempting to use the interface before udev can rename the interface, as well as only allowing specific interfaces to be used for Wi-Fi.

Using the example from above, we may edit the `/etc/connman/main.conf` file to include:

```
NetworkInterfaceBlacklist=SoftAp0,usb0,usb1,BeagleAP,wlan
```

This prevents connman from setting up the BeagleAP and wlan interfaces as Wi-Fi clients.
Note: We include wlan interfaces as udev fails to rename interfaces if connman attempts to use the interface

Systemd interface dependencies

This allows services to be started after the interfaces have been enumerated and/or renamed. Wi-Fi adapters do not have a predictable enumeration time and we may only want to run things once these interfaces have come up.

These lines may be added to the [Unit] section of a service file:

```
BindsTo= sys-subsystem-net-devices-BeagleAP.device
After= sys-subsystem-net-devices-BeagleAP.device
```

To find the interface to depend on, run:

```
systemctl list-units --no-pager
```

Connecting to eduroam or SFUNET-SECURE

Run the first 2 commands in the Connman CLI section to make sure your Beagle can see the network.

To connect to a WPA2-Enterprise network we need to create a config file before connecting. Create a file called `/var/lib/connman/eduroam.config` with the following contents:

```
[service_eduroam]
Type=wifi
# Can also be SFUNET-SECURE
Name=eduroam
EAP=peap
# Path to the thawte Primary Root CA - G3
CACertFile=/etc/ssl/certs/certificate.cer
Phase2=MSCHAPV2
# This is the Server name, very important to prevent MiTM
SubjectMatch=secure.sfu.ca
AnonymousIdentity=anonymous@sfu.ca
Identity=bfraser@sfu.ca
Passphrase=daSecurePassw0rd
# You should chown root:root and chmod 600
# this file so only root can read your password
```

Next, acquire the SFU certificate. The easiest way is to go to http://xpressconnect.its.sfu.ca

Click “Other Operating Systems” then “None of These” (or BeagleBone if that’s an option) then eduroam. Download the Primary Root CA. Copy this to the same file as specified in CACertFile line in the config above.

Finally run the command `connmanctl connect service_eduroam` and you should be connected.

To use SFUNET-SECURE instead, just change the Name line, everything else is the same.