

Wi-Fi USB Adapter guide

4th December 2017

Wireless functionality is a great feature to include in many projects, e.g. home automation. However, connecting to a wireless network via the command-line on the BeagleBone can be a non-trivial task, when compared to the process on a graphical user interface.

PARTS USED

1 x Edimax Wi-Fi Nano USB Adapter (EW-7811Un)

INTERFACING WITH ADAPTER

1. Insert the adapter into the USB-A port on the BeagleBone.
 - a. Note: if your project requires the use of multiple high-current USB devices to be connected to the BeagleBone, such as an external keyboard and mouse, the safest option to ensure enough power is available to each device is to use a powered USB hub.
2. Check that the Wi-Fi USB device is recognized by the system:

```
# lsusb
Bus 001 Device 002: ID 7392:7811 Edimax Technology Co., Ltd EW-7811Un 802.11n Wireless Adapter [Realtek RTL8188CUS]
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

- a. If the adapter is not displayed in the output, ensure you have the latest firmware:

```
# apt-get install firmware-realtek
```

 - i. If apt-get fails, check that you have internet access on your target [1].
3. Verify the correct drivers are loaded. The highlighted lines indicate the modules loaded on our system after the adapter was plugged in.

```
# lsmod
Module                Size  Used by
ccm                    6197  2
arc4                   1775  2
rtl8192cu              53490  0
rtl_usb                8487  1 rtl8192cu
rtl8192c_common        37662  1 rtl8192cu
rtlwifi                53165  3 rtl_usb,rtl8192c_common,rtl8192cu
mac80211               481075  3 rtl_usb,rtlwifi,rtl8192cu
cfg80211               399384  2 mac80211,rtlwifi
rfkill                 17761  2 cfg80211
usb_f_acm              7164  1
u_serial               11086  3 usb_f_acm
```

```

usb_f_rndis          22680  1
g_multi             5324  0
usb_f_mass_storage  42852  2 g_multi
u_ether             11806  2 usb_f_rndis,g_multi
libcomposite        43426  4
usb_f_acm,usb_f_rndis,g_multi,usb_f_mass_storage
omap_rng            4366  0
rng_core            7469  1 omap_rng
spi_omap2_mcspi     10689  0
uio_pdrv_genirq     3419  0
uio                  8463  1 uio_pdrv_genirq

```

- a. If some or all of the drivers are missing after plugging the adapter in:
 - i. Ensure that you have the realtek firmware installed (see troubleshooting for step 2)
 - ii. For custom kernels, ensure that the build config flags for RTL8192 drivers are set to 'y' and the drivers are copied over to the target in addition to the kernel image [2].
 - iii. As a last resort, drivers can be compiled directly for the target [3].
4. Run 'iwconfig' to show information on the wireless interface, wlan0:

```

# iwconfig
wlan0      IEEE 802.11bgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20
dBm
          Retry short limit:7   RTS thr=2347 B   Fragment thr:off
          Encryption key:off
          Power Management:off

lo         no wireless extensions.

eth0      no wireless extensions.

usb0      no wireless extensions.

```

- a. If iwconfig only contains entries stating 'no wireless extensions', check for CRDA (Central Regulatory Domain Agent) errors using dmesg. For errors stating "... *cfg80211: Exceeded CRDA call max attempts. Not calling CRDA*", install the latest crda, wireless-regdb and iw packages:

```
# apt-get install crda wireless-regdb iw
```
- b. Note: if iwconfig contains an interface called "tether", the Wi-Fi adapter is likely set to broadcast an ad-hoc network. Refer to the troubleshooting steps for connmanctl to fix this.

CONNECTING TO A NETWORK WITH CONNMANCTL

At this point, we are ready to configure the adapter to connect to a wireless access point. For this process, we will be using `connmanctl`, which is a terminal-based network manager [4] [although the reference guide for ConnMan is very useful, it doesn't describe any of the roadblocks that we encountered]. `Connmanctl` can be interacted with using either an interpreter-style command prompt or by directly passing the commands as arguments. We will be using the first method in this guide.

1. Enter the interactive shell by running `'connmanctl'`
2. Run `'technologies'` to check that the Wi-Fi adapter is powered and not set to tethering mode:

```
# connmanctl
connmanctl> technologies
/net/connman/technology/p2p
  Name = P2P
  Type = p2p
  Powered = False
  Connected = False
  Tethering = False
/net/connman/technology/wifi
  Name = WiFi
  Type = wifi
  Powered = True
  Connected = False
  Tethering = False
  TetheringIdentifier = BeagleBone-9400
  TetheringPassphrase = BeagleBone
/net/connman/technology/ethernet
  Name = Wired
  Type = ethernet
  Powered = True
  Connected = False
  Tethering = False
connmanctl>
```

- a. If the Wi-Fi adapter is not powered, run `'enable wifi'`.
 - b. If tethering is enabled, run `'tether wifi off'` as we cannot connect to a wireless network otherwise.
3. Scan for wireless networks by running `'scan wifi'`. As interactive commands in `connmanctl` are run asynchronously, the output of time-consuming commands will be inserted before the next prompt after a delay. Within a few seconds, "Scan completed for wifi" should be printed to the screen.
 - a. If no output is printed within 30 seconds or an error occurs, verify that tethering is indeed disabled.

- To view the wireless SSIDs that were discovered, run 'services'. The result should be similar to the following mock output:

```
connmanctl> services
*A0 myWifi1234          wifi_74da388f63ab_managed_psk
    myWifi4567          wifi_54454c555333_managed_psk
connmanctl>
```

- Run 'agent on' to prepare the wireless agent for entering passphrases.
- Connect to a wireless network by running 'connect <service-name>' and then entering the passphrase, if required. The easiest way is to copy-paste the service name into our command. In our example, we want to connect to myWifi1234:

```
connmanctl> connect wifi_74da388f63ab_managed_psk
Agent RequestInput wifi_74da388f63ab_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory, Alternates=[ WPS ]
]
  WPS = [ Type=wpspin, Requirement=alternate ]
Passphrase? password123
Connected wifi_74da388f63ab_managed_psk
connmanctl>
```

- If you see "Error /net/connman/service/<service-name>: Not registered" when you try connecting, the wireless agent may not be enabled. Run 'agent on' and then run the connect command again.
- Exit out of connmanctl by pressing Ctrl+d or typing 'quit'.
 - Ensure that you have been leased an IP address from the wireless access point:

```
# ifconfig
...

wlan0      Link encap:Ethernet  HWaddr 74:da:38:8f:63:ab
           inet addr:192.168.1.80  Bcast:192.168.1.255
Mask:255.255.255.0
           inet6 addr: fe80::76da:38ff:fe8f:63ab/64 Scope:Link
           inet6 addr: 2001:569:773a:f100:76da:38ff:fe8f:63ab/64
Scope:Global
           UP BROADCAST RUNNING MULTICAST DYNAMIC MTU:1500 Metric:1
           RX packets:7650 errors:0 dropped:0 overruns:0 frame:0
           TX packets:882 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1433405 (1.3 MiB)  TX bytes:129570 (126.5 KiB)
```

- Test DNS resolution and network connectivity by pinging google.ca. Note that once a connection has been stored, your BeagleBone will automatically connect to registered WiFi networks on booting into the Debian OS, provided the adapter is plugged in.

-
- a. A common issue we noticed is that pinging can fail initially after the connection is added. Waiting for around 30-60 seconds seems to be enough to let the network configuration changes to propagate through.
 - b. Note: in order to delete stored connections, the easiest method we found is the following:

```
# systemctl stop connman
# cd /var/lib/connman
# rm -rf wifi_74da388f63ab_managed_psk
# systemctl start connman
# systemctl status connman
● connman.service - Connection service
   Loaded: loaded (/lib/systemd/system/connman.service; enabled)
   Active: active (running) since Mon 2017-12-04 22:56:19 UTC;
1s ago
   Main PID: 22265 (connmand)
   CGroup: /system.slice/connman.service
           └─22265 /usr/sbin/connmand -n
...

```

ADDITIONAL RESOURCES

[1] “Networking Guide” by Brian Fraser (since internet connectivity is required to run apt-get)

<http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/Networking.pdf>

[2] “Driver Creation Guide” by Brian Fraser

<http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/DriverCreationGuide.pdf>

[3] Compiling RTL8192 drivers. “Beaglebone: Adding USB Wi-Fi & Building a Linux Kernel” by Derek Molloy <https://youtu.be/HJ9nUqYMjqs?t=41m50s>

[4] Complete ConnMan guide <https://wiki.archlinux.org/index.php/ConnMan>