Eder F. Torres Lopez

Eder.fernando.torres3@gmail.com

Abstract

For embedded systems projects based on single board computers (SBC), it is often necessary or desirable to have continues connection to the internet. The original BeagleBone Black (BBB) only contains an ethernet port and not native way to communicate wirelessly. This document aims to explain step by step how to compile drivers for a reliable USB Wi-Fi card to enable the BBB for wireless internet communication.

Compiling 8821au wifi card driver for linux

With BeagleBone Black

Contents

[Keywords and Glossary 2](#_Toc120015975)

[Introduction and Prerequisites 3](#_Toc120015976)

[1. Update Kernel 4](#_Toc120015977)

[2. Install Linux-Headers and other apps 4](#_Toc120015978)

[3. Downloading the Wi-Fi driver source files, compile and install the driver 5](#_Toc120015979)

[4. Configure the Wi-Fi Connection 6](#_Toc120015980)

[Conclusion 8](#_Toc120015981)

[References and additional information 8](#_Toc120015982)

# Keywords and Glossary

**Build**: in C, the build process is the complete process of converting the high-level source code representation of your embedded software into an executable binary image

**Compile**: is one of the steps of the building process, it converts high level language file into an object file, but it still needs to be linked or “glued” to libraries that the C code might have used. Since in this tutorial we don’t deal with object files, if the words “compile the code” are used, it’ means, “build the code”

**Ethernet**: is a type of communication protocol that connects computers within what’s called a “local area network (LAN)” or “wire area network (WAN)”. The connector used for this type of network is an RJ45 connector. The cables used for this network is called an “ethernet cable”

**GitHub**: is a distributed version-control platform where users can collaborate on or adopt open-source code projects, fork code, share ideas and more

**IP Address**: is a unique string of characters (set of numbers) that identifies each computer using the Internet Protocol to communicate over a network. Example of IP Address: 192.168.1.23

**Kernel**: is a computer program at the core of a computer’s operating system and generally has complete control over everything in the system, it facilitates interactions between hardware and software components.

**Linux-Headers**: is a package that provides the Linux kernel headers which are installed separately from Linux. They define function declarations contained in the kernel to be used on userspace programs. Programs that require kernel or hardware functionality need to be compile with these Linux-Headers

**Make**: is a Linux command used to build and maintain groups of programs and files from the source code

**Makefile**: it’s a special file that contains the shell commands needed to maintain the project. It defines targets and contains flgas that will determine the specific configuration of the building process

**Single Board Computer**: is a complete computer built on a single circuit board, with microprocessor, memory, input/output and other features required of a functional computer.

# Introduction and Prerequisites

The original BeagleBone Black doesn’t have integrated Wi-Fi capability included. There are a couple of options to provide the board with wireless internet capability. One is to connect it to a bridge repeater (which is a device with the same construct as a wireless network router), as show in Figure 1, that is battery powered to let the BBB be mobile for projects were mobility is required. The downside of this solution is that that it is very bulky, expensive and consumes significant amounts of power. Another method is to purchase a USB Wi-Fi Card that works out of the box with Debian in the BBB. This option is excellent when internet speed and bandwidth is not a priority, it also a low cost. The downside with this solution is that the signal reception and transmission is very weak and, in my experience, not very reliable, it also takes up a USB port, which can be solved by adding an externally powered USB Hub. The last option is using a USB Wi-Fi adapter meant to have good transmission and reception power (with a high-gain antenna) as shown in Figure 2, the advantage on these devices that they get a stable connection, they are relatively low priced. One of the major disadvantages is that there are no drivers on the BeagleBone’s Debian version right out of the box (and many other Linux distributions), therefore, they need to be downloaded and compiled (if you can find open-sourced drivers).

A picture containing indoor, military vehicle, transport

Description automatically generated

Figure 1: Mobile Robot that uses a repeater bridge (black object on top of robot) to give internet connection to a BBB

A picture containing wall, indoor, microscope, projector

Description automatically generated

Figure 2: Mobile robot that uses a USB-Wi-Fi Adapter with a high-gain antenna (black device on the back of robot that looks like a stick) to give internet connection to a BBB

To follow this guide successfully it’s assumed that the following list have been gathered/followed prior to starting.

* BeagleBone Black board (original with no Wi-Fi) powered through the barrel connector, 5V and at least 1 amp
* tp-link – Archer T2U Plus – AC600 High Gain Wireless Dual Band USB Adapter
* Follow the instructions in the following link that I have posted in hackster.io: <https://www.hackster.io/ederfernandotorres3/setup-beaglebone-black-with-device-tree-overlays-9e0ded>

All the commands that will be run on the BBB terminal will be enclosed in double quotations and will be typed in Courier New font. A quick explanation of what the command does will be given before or after the command. For example: “sudo apt update” <- fetches the updated catalog for current installed programs.

Don’t expect to be commanded to “type” or “enter” before the command.

# Update Kernel

The following listed commands are needed to update the kernel in the BeagleBone

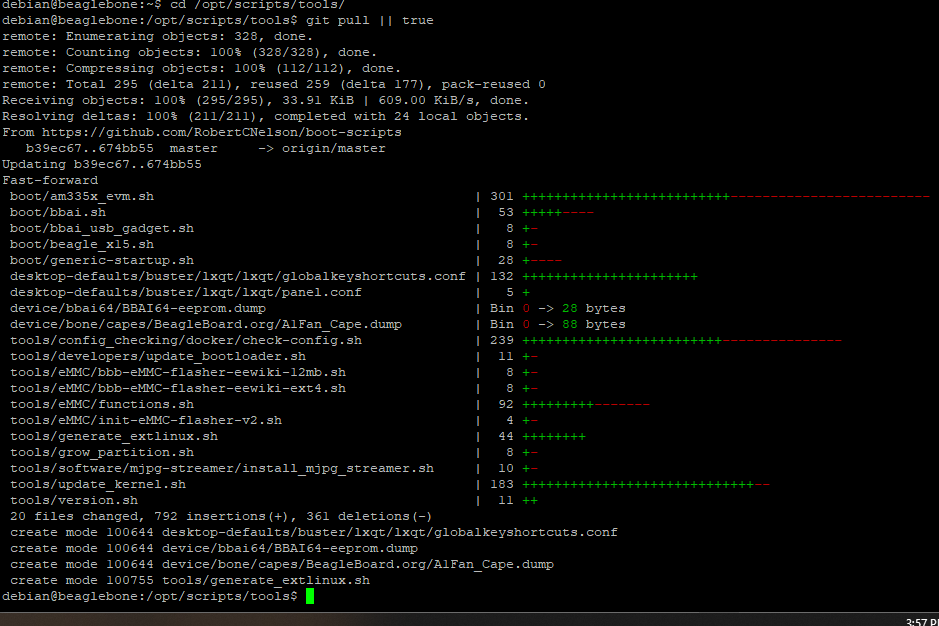
1. “sudo apt update” <- to update the app catalog
2. “sudo apt upgrade” <-to install the updated apps
3. “cd /opt/scripts/tools/” <- to go to a folder with scripts
4. “git pull || true” <- it will use git to make sure that this folder is up to date

Figure 3: This shows the terminal’s appearance after updating the “/tools” directory

1. “sudo ./update\_kernel.sh” <- this will run the script to update the kernel. Enter the password if you are asked to
2. “sudo reboot” <- This will reboot the system

# Install Linux-Headers and other apps

These are the steps needed to be done before compiling and installing the adapter’s drivers in the BBB. In this section we make sure that the BBB has all the required software installed such as the kernel’s header files, compiler, framework and a version control system need to download the source code from its repository.

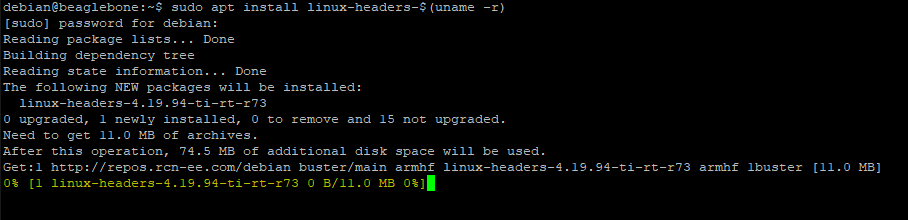
1. “sudo apt install linux-headers-$(uname -r) <- $(uname -r)” <- This command will be install the current Linux-headers. These headers will be needed if you are compiling software that use functions that the Linux kernel provides.

Figure 4: Installing the Linux-Headers

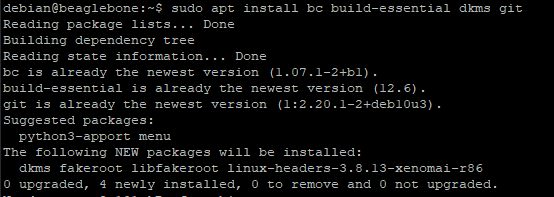
1. “sudo apt install bc” <- installs a basic calculator needed by the source code
2. “sudo apt install build-essential” <- is a “meta-package” that will install programs needed to compile C/C++ programs such as GNU C compiler, etc
3. “sudo apt install dkms” <- is a framework that enables generating Linux kernel modules
4. “sudo apt install git” <- is a version control system, it lets you download files from GitHub in this document (and other functions that we won’t be using in this guide)
   1. Optionally you can type: “sudo apt install bc build-essential dkms git” to install all the above apps (excluding the Linux-Headers) in a single command.

Figure 5: Installing more than one app in a single command line

# Downloading the Wi-Fi driver source files, compile and install the driver

In this step we will be downloading the source files of the Wi-Fi adapter driver. Instructions to compile the drivers for different systems are included in a file named README, this guide is based on that file. There is a file named “Makefile” that contains the instructions for the compiler on how to build the code. The maintainers of this source code in GitHub have included flags in the Makefile to choose the target machine that it’s being built for, for example, you can change the flag of “CONFIG\_PLATFORM\_I386\_PC” from “n” to “y” to tell the compiler to build executable code for an Intel x86 compatible processor. The BBB is not in the list of platforms, but there is one for the 32-bit Raspberry Pi, which runs in an ARM processor just like the BBB, and the endianness and other settings are the same, so we can select the “CONFIG\_PLATFORM\_ARM\_RPI” as the flag for the BBB and it will work.

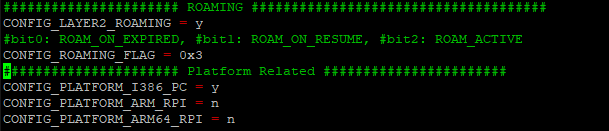
1. “mkdir wifi\_driver” <- This creates a directory to hold the downloaded driver, I will call mine “wifi\_driver”
2. “cd wifi\_driver” <- It enters into the directory you just created
3. “git clone <https://github.com/morrownr/8821au-20210708.git>” <- This will copy the source code found in GitHub int the directory that you just created
4. “cd 8821au-20210708” <- it will change the current directory to the one that was just created in the previous step
5. Even though this is not the Raspberry Pi, we will use the same settings as if it was:
   1. “sudo nano Makefile” and look for the area labeled “Platform Related”

Figure 6: This is the area of the Makefile that we are looking for, it's labeled: "Platform Related"

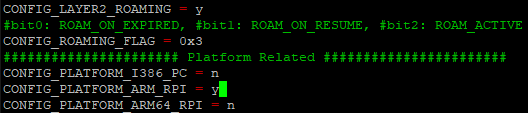
* 1. Change: “CONFIG\_PLATFORM\_I386\_PC = y” to “CONFIG\_PLATFORM\_I386\_PC = n”
  2. Change: “CONFIG\_PLATFORM\_ARM\_RPI = n” to “CONFIG\_PLATFORM\_ARM\_RPI = y”

Figure 7: Platform flags with the correct value to build for the BBB

* 1. Press on your keyboard: “ctrl” + “x” then “y” and then “enter” to exit and save changes

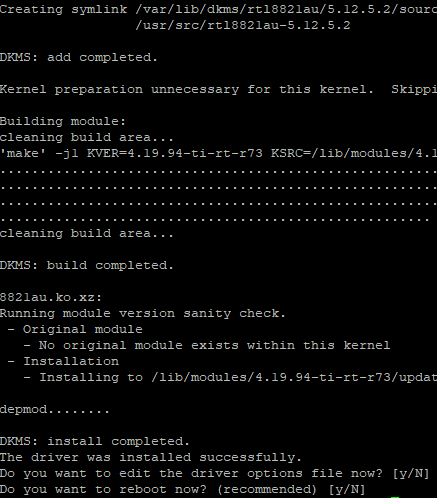
1. Now we’ll run the install script by typing the following commands:
   1. “sudo ./install-driver.sh” <- it will take a few minutes (around 25). This file automatically runs the “make” command and prints messages to the console to make the installation more “human interactive”
   2. After it finishes compiling and installing, it will ask if you want to edit the driver options, you may click “n” for no.

Figure 8: Messages sent through installation and questions asked after the installation is done.

* 1. You will be asked if you want to reboot, type “y” and then “enter” if you are not prompted. Type: “sudo reboot”

1. Now you may connect the Wi-Fi card to the USB port, make sure your BBB is connected to the 5V barrel connector with at least 1 Amp.

# Configure the Wi-Fi Connection

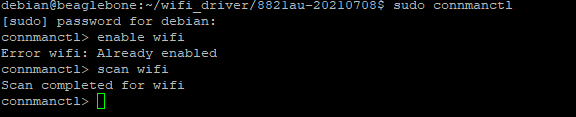
1. Run the command “sudo connmanctl” <- Connmanctl is a utility that facilitates the connection to a wireless network in an easy manner.
2. Type: “enable Wi-Fi” <- you might get an error that the Wi-Fi is already enabled, this is fine
3. Type: “scan Wi-Fi” and wait until Connmanctl it says that it has completed

Figure 9: Scanning for Wi-Fi connections using Connmanctl

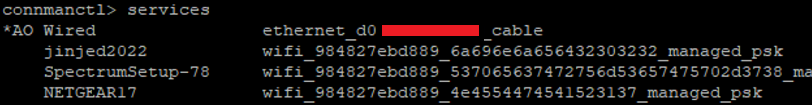
1. Type: “services” to show the list of the scanned networks

Figure 10: “services” command on Connmanctl lists the available internet connections

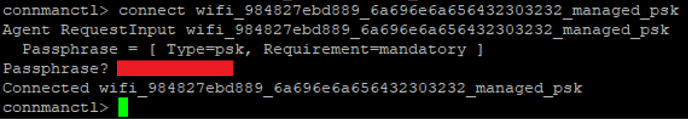
1. Type: “agent on” to turn on the agent and allow to enter the passphrase (password) of the connection
2. Type: “connect <id of the network>” <- the “id of the network” is the string of letters after the SSID (name of the Wi-Fi connection), it starts with “wifi\_”. For example, to connect to jinjed2022, I would have to type: “connect wifi\_984827ebd889\_6a696e6a656432303232\_managed\_psk”
3. You will be asked for a passphrase: type it.

Figure 11: This is where you type the password of your Wi-Fi connection

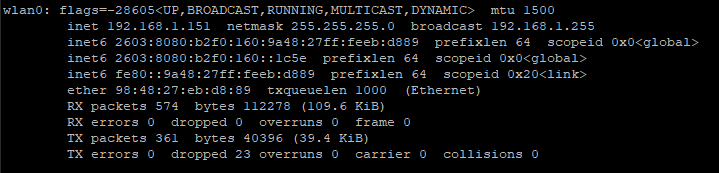
1. To verify that the previous steps worked, type: “services” you should see “\*A0” or “\*AR” next to the desired Wi-Fi connection
2. Type: “quit” to exit connmanctl
3. Type: “ifconfig” <- this command is used to view network connections on Linux (and can be used to change the configuration of the network interfaces)
4.  Find the wlan0 connection. Write down the IP address (inet), in my case its private IP address is 192.168.1.151

Figure 12: wlan0 information after entering the ifconfig command

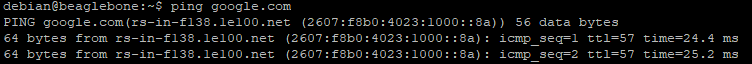
1. Type “sudo shutdown now” to turn off the BBB. And disconnect the ethernet cable if that’s how you are remoting into the board
2. Turn on the BBB with the USB Wi-Fi card connected. It might take longer than before to boot up. Remote in with the IP address gathered in step 11. If it doesn’t connect, go to this guide: <https://www.hackster.io/ederfernandotorres3/setup-beaglebone-black-with-device-tree-overlays-9e0ded>. And go to section where it explains how to use IP Scanner to find the new IP address of the BBB
3. Type: “ping google.com” to test if you the BBB can communicate to the internet
4. Press “CTRL” and “c” on your keyboard to stop pinging google.com

Figure 13: Successful connection to google.com shows how many packages were received and how long it took

# Conclusion

This guide can be used to give the BBB Wi-Fi capabilities. This can facilitate in projects where a wired connection does not work, and the bulky solution of a bridge repeater is not an option, or it might be too expensive. This solution can be used on mobile robots or sensor nodes that are apart from the router. The bandwidth of this solution is good to send video feed in HD (or smaller resolution for a more resilient connection).

# References and Additional Information

<https://manpages.debian.org/testing/connman/connmanctl.1.en.html>

<https://www.fis.gatech.edu/how-to-configure-bbw-wifi/>

<https://github.com/morrownr/8821au-20210708>