

Maior portal de Automação Comercial do Brasil! Encontre o que sua empresa precisa com preços especiais, atendimento especializado, entrega rápida e pagamento facilitado.





Adaptador Wireless TP-Link TL-WN823N

O Mini Adaptador USB Wireless N de 300Mbps TLWN823N da TP-LINK é projetado para conectar seu laptop ou computador a uma rede wireless e desfrutar de uma performance de alta velocidade wireless.





Installation Guide for Linux

Ubuntu 16.04 LTS						
1. Development Environment						
Compile the Driver						
2.1. Compilation tool and kernel sources						
2.2. Compile the Driver						
3. Load the Driver4						
Mint 18.034						
1. Development Environment4						
2. Compile the Driver4						
2.1. Compilation tool and kernel sources 4						
2.2. Compile the Driver						
3. Load the Driver5						
Raspberry Pi35						
1. Development Environment5						
2. Compile the Driver						
2.1. Compile Kernel source						
2.1.1. Download and Install Tools						
2.1.2. Get Kernel source						
2.1.3. Modify Kernel						
2.1.4. Complet the Kernel						
3 Load the Driver						
All 2010.1						
1. Development Environment						
2. Compile the Driver						
2.1. Install the Kernel Header File						
2.1.1. Opdate the Soltware Source						
2.1.3. Compile Driver Source						
3. Load the Driver						
se the Graphical Interface						
se the Command Line						
1. Identify the Device14						
2. Create the Interface						
Change the Interface Status to Up 14						
4. Start wpa_supplicant in the background						

Contents

4.1.	Scan Wireless Networks (SSID)	15
4.2.	Connect to the AP	15
4.3.	Enable DHCP client	17

Ubuntu 16.04 LTS

1. Development Environment

The development environment in Ubuntu is required as follows:

Development Environment					
OS	Ubuntu 16.04 LTS				
Kernel version	4.13.0-36-generic				
Gcc version	5.4.0				

2. Compile the Driver

2.1. Compilation tool and kernel sources

Before you compile the driver, please make sure you have the correct compile tool and kernel sources. In Ubuntu 16.04 LTS, we can install compile tool gcc by command "apt-get install gcc"

\$ apt-get install gcc

Note: We recommend you install the same version tool to compile the driver.

For example:

```
ubuntu@ubuntu:~$ cat /proc/version
Linux version 4.13.0-36-generic (buildd@lgw01-amd64-033) (gcc version 5.4.0 2016
0609 (Ubuntu 5.4.0-6ubuntu1~16.04.9)) #40~16.04.1-Ubuntu SMP Fri Feb 16 23:25:58
UTC 2018
ubuntu@ubuntu:~$
```

According to the command "*cat /proc/version*", you could see your Ubuntu 16.04 LTS system is compiled by gcc5.4.0. By default, gcc5.4.0 is already installed in Ubuntu 16.04 LTS, you could use gcc5.4.0 to compile the driver directly.

Generally, compatible kernel headers are already built in Ubuntu 16.04 LTS, so you don't need to separately download and compile the kernel sources. However, if no related kernel headers are integrated in your system, please install the kernel sources first.

2.2. Compile the Driver

Use Terminal to go to the driver directory and run the following commands to compile the driver.

```
$ make clean
$ make
```

After compiling, you can see a *name of the chip.ko* file is stored in the directory of the driver.

3. Load the Driver

Here we show the 88x2bu.ko wireless driver loading process as an example. Run the following command to load the driver.

\$ sudo cp 88x2bu.ko /lib/modules/[kernel version]/kernel/drivers/net/wireless/ #[kernel version] is the directory name of the system kernel version

\$ sudo depmod –a

\$ sudo modprobe 88x2bu

Or directly use insmod to load the driver.

\$ sudo insmod 88x2bu.ko

After loading the driver, run the following command to check if the driver is successfully loaded.

\$ Ismod

Mint 18.03

1. Development Environment

The development environment in Mint is required as follows:

Development Environment					
OS	Mint 18.03				
Kernel version	4.10.0-38.generic				
Gcc version	5.4.0				

2. Compile the Driver

2.1. Compilation tool and kernel sources

Before you compile the driver, please make sure you have the correct compile tool and kernel sources. In Mint, we can install compile tool gcc by command "apt-get install gcc"

\$ apt-get install gcc

Note: We recommend you install the same version tool to compile the driver. For example:

```
who@who-B85M-D3V-A ~ $ cat /proc/version
Linux version 4.10.0-38-generic (buildd@lgw01-amd64-059) (gcc version 5.4.6 2016
0609 (Ubuntu 5.4.0-6ubuntu1~16.04.4) ) #42~16.04.1-Ubuntu SMP Tue Oct 10 10:32:2
0 UTC 2017
who@who-B85M-D3V-A ~ $ \[
```

According to the command "*cat /proc/version*", you could see your Mint system is compiled by gcc5.4.0, so we should use gcc5.4.0 to compile the driver.

Generally, compatible kernel headers are already built in Mint, so you don't need to separately download and compile the kernel sources. However, if no related kernel headers are integrated in your system, please install the kernel sources first.

2.2. Compile the Driver

Use Terminal to go to the driver directory and run the following commands to compile the driver.

```
$ make clean
$ make
```

After compiling, you can see a *name of the chip.ko* file is stored in the directory of the driver.

3. Load the Driver

Here we show the 88x2bu.ko wireless driver loading process as an example. Run the following command to load the driver.

\$ sudo cp 88x2bu.ko /lib/modules/[kernel version]/kernel/drivers/net/wireless/

\$ sudo depmod -a

\$ sudo modprobe 88x2bu

Or directly use insmod to load the driver.

\$ sudo insmod 88x2bu.ko

After loading the driver, run the following command to check if the driver is successfully loaded.

\$ Ismod

Raspberry Pi3

1. Development Environment

The development environment in Raspberry Pi 3 is required as follows

Development Environment						
OS	6.3.0-18-rpi-deb9u1					
Kernel version	4.9.80-v7					
Hardware version of	Raspberry Pi 3 Model B					
development board						

2. Compile the Driver

Before you compile the driver, please make sure you have the correct compile tool and kernel sources.

2.1. Compile Kernel source

Here we illustrate the instructions for local building to compile the driver for Linux.

2.1.1. Download and Install Tools

Note: Before local building, make sure your raspberrypi system is connected to the internet.

Install Git, bc and other related tools.

\$ sudo apt-get install git bc

2.1.2. Get Kernel source

Click the following links to download raspberrypi kernel source and other related tools.

https://github.com/raspberrypi/linux

https://github.com/raspberrypi/tools

Before local building, make sure if you need to update the kernel. If your adapter supports the current kernel version, you don't need to update the kernel, and just download the kernel sources of this version. If you have to update the kernel, choose the kernel sources of the desired version. Here we download the version 4.9 kernel sources.

Create Linux-src directory in the local user's root directory to store kernel sources. If you have installed Git, you can use Git to obtain Linux kernel sources from Github; if you directly download the .zip file, use the following jar command to decompress this file.

\$ sudo jar –xf XXX.zip

Note: It is recommended not to use the unzip software to decompress the .zip file.

2.1.3. Modify Kernel

Run the following commands to modify Linux kernel. You can also modify the kernel according to your demands.

\$ cd linux	/* go the directory of kernel sources */
\$ KERNEL=kernel7	
\$ make bcm2709_defconfig	
<u>+ ·····</u> -· <u>=</u> ····· - 5	

Note: The instructions for Raspberry Pi3 and other versions of Raspberry are slightly different, for details of other versions, please refer to the instructions on Raspberry official website.

2.1.4. Compile the Kernel

Run the following commands to compile and install the kernel and related device tree. It

may take a few minutes.

\$ make –j4 zlmage modules dtbs

\$ sudo make modules_install

\$ sudo cp arch /arm/boot/dts/*.dtb /boot/

\$ sudo cp arch/arm/boot/dts/overlays/*.dtb* /boot/overlays/

\$ sudo cp arch/arm/boot/dts/overlays/README /boot/overlays/

\$ sudo cp arch/arm/boot/zImage /boot/\$KERNEL.img

Note: "-j4" refers to using *Raspberry Pi3 and 4* to compile to accelerate the compilation process.

Power off the development board of Raspberry Pi3 and then run the following command to confirm the kernel version.

\$ uname –a or \$ cat /proc/version

2.2. Compile the Driver Source

Go to the driver's directory, open the Makefile file to support Raspberry Pi3. By default, the CONFIG_PLATFORM_I386_PC Set macro is enabled. the value for the value CONFIG_PLATFORM_BCM2709 to and set for У, CONFIG_PLATFORM_I386_PC to n.

CONFIG_PLATFORM_BCM2709 = y CONFIG_PLATFORM_I386_PC = n CONFIG_PLATFORM_ANDROID_X86 = n

After setting the parameters, use Terminal to go to the directory in which the driver source file is stored. Run the following commands to compile the driver.

\$ make clean

\$ make

3. Load the Driver

Here we show the 8192eu.ko wireless driver loading process as an example. Run the following command to load the driver.

```
$ sudo cp 8192eu.ko /lib/modules/[kernel version]/kernel/drivers/net/wireless/
```

\$ sudo depmod -a

\$ sudo modprobe 8192eu

Or directly use insmod to load the driver.

```
$ sudo insmod 8192eu.ko
```

After loading the driver, run the following command to check if the driver is successfully loaded.

Kali 2018.1

1. Development Environment

The development environment in Kali 2018.1 is required as follows.

Development Environment						
OS	Kali 2018.1					
Kernel Source Version	4.14.0-kali3-amd64					

2. Compile the Driver

2.1. Install the Kernel Header File

Before compiling the driver in Kali 2018, make sure you have installed and compiled the right Linux header file. Follow the instructions to install and compile the Linux header file.

2.1.1. Update the Software Source

Run the following commands to update the software source and related tools.

\$ sudo apt-get clean\$ sudo apt-get update\$ sudo apt-get upgrade

2.1.2. Install the Kernel Header File

1) Method 1: Run the following command to install the kernel header file.

\$ sudo apt-get install linux-headers-\$(uname -r)

After running this command, the system will automatically find the matched kernel header file to download and install it. If the Kali server is updated, you may not find the specific file, in this case, you can manually download and install the header file.



2) Method 2: Manually Download and Compile to Install

Find the matched kernel header file in the download source of your Kali software.

Click the following link to go to the official website to download Linux header file and related tools.

http://http.kali.org/kali/pool/main/l/linux

Check the system version of Kali

\$ uname -r

The system version we used here is shown as below.

root@kali:/home/kali#Juname_lr_4.14 4.14.0-kali3-amd64_4_0_kali3-amd64

Download and compile linux-kbuild

In the Kali download links, find the linux-kbuild file of your system. Here we choose linux-kbuild-4.14_4.14.17.-1kali_amd64.deb as an example.

Iinux-kbuild-4.14-dbgsym_4.14.17-1kali1_amd64.deb	2018-02-16 12:48 609K
Iinux-kbuild-4.14-dbgsym_4.14.17-1kali1_arm64.deb	2018-02-16 12:16 627K
Iinux-kbuild-4.14-dbgsym_4.14.17-1kali1_armel.deb	2018-02-16 17:40 599K
Iinux-kbuild-4.14-dbgsym_4.14.17-1kali1_armhf.deb	2018-02-16 18:54 593K
👔 <u>linux-kbuild-4.14-dbgsym 4.14.17-1kali1 i386.deb</u>	2018-02-16 12:54 562K
Iinux-kbuild-4.14_4.14.17-1kali1_amd64.deb	2018-02-16 12:48 743K
🕈 <u>linux-kbuild-4.14_4.14.17-1kali1_arm64.deb</u>	2018-02-16 12:16 720K
Iinux-kbuild-4.14_4.14.17-1kali1_armel.deb	2018-02-16 17:40 722K
Iinux-kbuild-4.14_4.14.17-1kali1_armhf.deb	2018-02-16 18:54 724K

After downloading the file, use Terminal to go to the directory and run the following command to install the file.

|--|

Download and compile linux-header-common

In the Kali download links, find the linux-header-common file of your system. Here we choose linux-header-4.14.0-kali3-common_4.14.17-1kali_all.deb as an example.

linux-headers-4.14.0-kali3-common-rt_4.14.17-1kali1_all.deb	2018-02-16 12:47 5.7M
linux-headers-4.14.0-kali3-common_4.14.17-1kali1_all.deb	2018-02-16 12:47 7.5M
👔 linux-headers-4.14.0-kali3-marvell_4.14.17-1kali1_armel.deb	2018-02-16 17:40 345K
👔 linux-headers-4.14.0-kali3-rt-686-pae_4.14.17-1kali1_i386.deb	2018-02-16 12:53 450K
👔 linux-headers-4.14.0-kali3-rt-amd64_4.14.17-1kali1_amd64.deb	2018-02-16 12:47 453K
🕐 linux-headers-4.15.0-kali1-686-pae_4.15.4-1kali1_i386.deb	2018-02-23 10:22 450K
linux-headers-4.15.0-kali1-686_4.15.4-1kali1_i386.deb	2018-02-23 10:22 450K

After downloading the file, use Terminal to go to the directory and run the following command to install the file.

\$ sudo dpkg -i linux-header-4.14.0-kali3-amd64_4.14.17-1kali_amd64.deb

Run the following command to check if the kernel header file is successfully installed.

\$ dpkg-query -s linux-headers-\$(uname -r)

After the linux-header is successfully installed, run the following command to display the

detailed linux-header information.



Check the /lib/modules/<kernel-version>/ directory and you will see a build link file. li:/home/kali# ls_-l_/lib/modules/4.14.0-kali3-amd64

CATES CFT.				FNUGR' FXT'				
total 158	184						EUNG	
lrwxrwxrw	K 1	root	root	41	Feb	16	03:38	<pre>build -> /usr/src/linux-headers-4.14.0-kali3-amd64</pre>
drwxr-xr-:	ĸ 12	root	root	4096	Apr	10	23:22	kernel
d rwx rwx r -:	K 24	root	root	ewd i s k4096	Nov	12	13:46	linux-stable-4.14 rebuilts/acc/linux-x86/arm/arm-eabi-4
-rwxr-xr-:	κ_ 1	root	root	157647725	Apr	11	22:54	<pre>linux-stable-4.14.tar.gz/shieldtablet/obi/KERNEL/</pre>
-rw-rra	1	root	root	1056336	Apr	19	20:55	modules.alias
-rw-rr-	- 1	root	root	1006117	Apr	19	20:55	modules.alias.bin
-rw-rr-	- 1	root	root	4106	Jan	8	10:09	modules.builtin
-rw-rr-	DNF ¹	root	root	_{NV TK} 5626	Apr	19	20:55	modules.builtin.bin
-rw-rr-	1051	root	root	398797	Apr	19	20:55	modules.dep
	AGS^1	root	root	550106	Apr	19	20:55	modules.dep.bin
- rw- r r-	AGS^1	root	root	G TOCT 434	Apr	19	20:55	modules.devname EVENT
-rw-rr-	- 1	root	root	133414	Jan	8	10:09	modules.order
zewarsera	rm 1	root	root	772	Apr	19	20:55	modules.softdep
-rw-rr-	- 1	root	root	508979	Apr	19	20:55	modules.symbols
cRWsS-cSm	PIL ¹	root	root	628638	Apr	19	20:55	modules.symbols.bin
lrwxrwxrw	K 1	root	root	42	Feb	16	03:38	<pre>source -> /usr/src/linux-headers-4.14.0-kali3-common</pre>
root@kali	:/ho	me/ka	li#					

2.1.3. Compile Driver Source

Use Terminal to go to the driver directory. Run the following commands to compile the driver.

\$ make clean

\$ make

After compiling, you can see a name of the chip.ko file is stored in the directory of the driver.

3. Load the Driver

Here we show the 88x2bu.ko wireless driver loading process as an example. Run the following command to load the driver.

\$ sudo cp 88x2bu.ko /lib/modules/[kernel version]/kernel/drivers/net/wireless/

\$ sudo depmod -a

\$ sudo modprobe 88x2bu

Or directly use insmod to load the driver.

\$ sudo insmod 88x2bu.ko

After loading the driver, run the following command to check if the driver is successfully loaded.

\$ Ismod

Use the Graphical Interface

Ubuntu, Mint, Raspberry Pi and Kali all provide friendly graphical interface. After the adapter driver is successfully installed, you can use the graphical interface to manage your wireless settings. The interfaces for different system version are slightly different and here we use the interfaces for Kali 2018.1 as an example for illustration.

 After successfully loading the driver, you will see a network connection icon in the task bar. Choose Wi-Fi Not Connected > Wi-Fi Settings to display the available wireless networks.



Q Settings	Wi-Fi — — 🔿	
후 Wi-Fi	Visible Networks	
Bluetooth	TP-Link_D003_5G 🖀 📚	
Background	TP-LINK_1234_5G 🖀 📚	
Notifications	C9 5G 🖀 🕿	-
Q Search	yxf_5GHz 🗢 🗢	_
Region & Language	TP-Link_Extender_5GHz_F72566	
Our Universal Access	TP-Link_1F6D_5G_1	
∎ Online Accounts	TP-Link_Guest_1F6D_5G_1	
🗱 Privacy	TP-Link Extender 5GHz	-
< Sharing		
≠ Sound	C7 voi 5611111111111	
🔓 Power	izeb 5a	
딮 Network		
	IP-Link_Guest_2522_5G T ♥	~

2. Select TP-Link_D003_5G and enter its password to connect to this network.

P	Authentication required by wireless network Passwords or encryption keys are required to access the wireless network "TP-Link_D003_5G". Password:				
	Cancel	C	Connect		

3. After connecting to this network, you can check its detailed wireless settings.

٩	Settings	Wi-Fi Connected	
()•	Wi-Fi	Mattala Makanala	
*	Bluetooth	Cancel TP-Link_D003_5G Apply	
٢	Background	Details Identity IPv4 IPv6 Security Signal Strength Excellent	1 7 7 8
9	Notifications	Security WPA2	
۹	Search	IPv4 Address 192.168.1.102	1
0	Region & Language	IPv6 Address Hardware Address	\$
6	Universal Access	Default Route 192.168.1.1	î
-D.c	Online Assounts	DNS 192.168.1.1	Ŷ
-0~		✓ Connect automatically	
000	Privacy	✓ Make available to other users	ę
<	Sharing		<u>n</u> 2
#	Sound	Forget Connection	
Ge	Power		
Sp	Network	TP-Link_1F6D_5G_1	*
		TP-Link Guest 1E6D 5G 1	* *

Use the Command Line

You can use commands to manage your wireless setting in Linux. Here we use the interfaces for Kali 2018.1 as an example for illustration.

1. Identify the Device

Inert the USB wireless adapter, and run the following command to check if the adapter is identified.

\$ Isusb

2. Create the Interface

Run the following command to check if the wireless network interface is created.

\$ ifconfig

3. Change the Interface Status to Up

Check if the WLAN interface is *up*. If not, run the following command. Here we use *wlan1* as an example.

\$ ifconfig wlan1 up

If it failed to change to up, run the following command to set the state again.

\$ rfkill unblock wifi\$ ifconfig wlan1 up

4. Start wpa_supplicant in the background

Run the following command:

```
$ wpa_supplicant -Dnl80211 -iwlan1 -c ./ wpa_0_8.conf -B
```

Note: wpa_0_8.conf is a file in the current driver directory, go to the driver directory when running the command.

If the command above is not effective, run the following command to end the wpa_supplicant procedure and then run the above command again.

\$ killall wpa_supplicant

If your Linux kernel does not support 802.11, run the following command.

\$ wpa_supplicant -Dwext -iwlan0 -c ./wpa_0_8.conf -B

4.1. Scan Wireless Networks (SSID)

Run the following commands.

\$ wpa_cli -p /var/run/wpa_supplicant scan
\$ wpa_cli -p /var/run/wpa_supplicant scan_results

4.2. Connect to the AP

1) Open

Run the following commands

\$ wpa_cli -p /var/run/wpa_supplicant remove_network 0

\$ wpa_cli -p /var/run/wpa_supplicant ap_scan 1

\$ wpa_cli -p /var/run/wpa_supplicant add_network

\$ wpa_cli –p /var/run/wpa_supplicant set_network 0 ssid "tplink" //tplink is the SSID of the desired AP. The SSID is in double quotation marks and then as a whole enclosed by single quotation marks.

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 key_mgmt NONE

\$ wpa_cli -p /var/run/wpa_supplicant select_network 0

2) WEP40 with open system

\$ wpa_cli -p /var/run/wpa_supplicant remove_network 0
\$ wpa_cli -p /var/run/wpa_supplicant ap_scan 1
\$ wpa_cli -p /var/run/wpa_supplicant add_network
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 ssid ""tplink"'
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 key_mgmt NONE
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_key0 1234567890
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0
\$ wpa_cli -p /var/run/wpa_supplicant select_network 0

3) WEP40 with shared key

\$ wpa_cli -p /var/run/wpa_supplicant remove_network 0
\$ wpa_cli -p /var/run/wpa_supplicant ap_scan 1
\$ wpa_cli -p /var/run/wpa_supplicant add_network
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 ssid "tplink"
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 key_mgmt NONE
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_key0 1234567890
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 auth_alg SHARED \$ wpa_cli -p /var/run/wpa_supplicant select_network 0

4) WEP 104 with open system

\$ wpa_cli -p /var/run/wpa_supplicant remove_network 0

\$ wpa_cli -p /var/run/wpa_supplicant ap_scan 1

\$ wpa_cli -p /var/run/wpa_supplicant add_network

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 ssid ""tplink"

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 key_mgmt NONE

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_key0

12345678901234567890123456

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0

\$ wpa_cli -p /var/run/wpa_supplicant select_network 0

5) WEP 104 with open system

\$ wpa_cli -p /var/run/wpa_supplicant remove_network 0
\$ wpa_cli -p /var/run/wpa_supplicant ap_scan 1
\$ wpa_cli -p /var/run/wpa_supplicant add_network
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 ssid "tplink"
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 key_mgmt NONE
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_key0
12345678901234567890123456
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0
\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 auth_alg SHARED
\$ wpa_cli -p /var/run/wpa_supplicant select_network 0

Note:

If the WEP key is ASCII, run the following command:

#WEP40: wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0 "'12345"'
#WEP104: wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0
''1234567890123"'
If the index for WEP key is 0-3, run the following command
#wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_keyX
12345678901234567890123456
#wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx X

6) TIKP/AES

\$ wpa_cli -p /var/run/wpa_supplicant remove_network 0
\$ wpa_cli -p /var/run/wpa_supplicant ap_scan 1

\$ wpa_cli -p /var/run/wpa_supplicant add_network

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 ssid "tplink"

\$ wpa_cli -p /var/run/wpa_supplicant set_network 0 psk "12345678"

\$ wpa_cli -p /var/run/wpa_supplicant select_network 0

4.3. Enable DHCP client

Run the following command

\$ dhclient wlan1

After running the command, the adapter will get an IP assigned by the AP. Then you can run the ping command to check if the wireless connection is successful.

```
li:/home/kali/Documents/wpa_supplicant_hostapd# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
       ether 40:8d:5c:1b:34:28 txqueuelen 1000
                                               (Ethernet)
       RX packets 9950 bytes 5963340 (5.6 MiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 7420 bytes 676707 (660.8 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
       inet 127.0.0.1 netmask 255.0.0.0
       inet6 ::1 prefixlen 128 scopeid 0x10<host>
       loop txqueuelen 1000 (Local Loopback)
       RX packets 474 bytes 38286 (37.3 KiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 474 bytes 38286 (37.3 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
vlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.113 netmask 255.255.255.0 broadcast 192.168.1.255
       inet6 fe80::c0cc:8e6c:6977:cf24 prefixlen 64 scopeid 0x20<link>
       ether 50:3e:aa:44:65:51 txqueuelen 1000 (Ethernet)
       RX packets 118 bytes 14574 (14.2 KiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 103 bytes 11253 (10.9 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
     ali:/home/kali/Documents/wpa_supplicant_hostapd# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
54 bytes from 192.168.1.1: icmp seq=1 ttl=64 time=1.24 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.45 ms
64 bytes from 192.168.1.1: icmp seq=3 ttl=64 time=1.00 ms
-- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
tt min/avg/max/mdev = 1.001/2.530/7.867/2.672 ms
       i:/home/kali/Documents/wpa_supplicant_hostapd# route
Kernel IP routing table
              Gateway
Destination
                              Genmask
                                              Flags Metric Ref
                                                                 Use Iface
default
               Archer.lan
                              0.0.0.0
                                             UG
                                                   600
                                                          0
                                                                   0 wlan0
                              255.255.255.0
               0.0.0.0
                                                   600
                                                          0
192.168.1.0
                                             U
                                                                   0 wlan0
 oot@kali:/home/kali/Documents/wpa_supplicant_hostapd#
```

Note:

- 1. Run the commands under the root account.
- 2. If you use *ifconfig* command to confirm you have obtained the IP address and use *ping* command to confirm you wireless connection is successful, but the internet is still unavailable, you can run the following commands to change the default system

gateway to the router's LAN IP.

\$ route del default wlan0//Delete the default gateway of wlan0\$ route add default gw 192.168.1.1//Add the router's LAN IP as the default gateway.