




# Installation Guide For Linux Driver



# TP-LINK Statement for Linux driver

TP-LINK has released Linux driver for some TP-LINK USB wireless adapters to support Linux system. The driver file has included an installation guide about how to install and use the adapter on Linux OS.

The driver is recommended to be installed on Linux OS which applies the kernel version that we have listed on TP-LINK official website.

<a href="#">Archer_T2U_V1_150901</a> 	<b>Published Date</b>	01/09/15	<b>Language</b>	English	<b>File Size</b>	6.29 MB
	<b>Operating System</b>	Linux (Kernel version 2.6~3.16)				
	<b>Notes</b>	For Archer T2U V1				

Since Linux is developed at an open system with various branches, we cannot guarantee that our driver could work on your Linux system.

Given the specificity of the Linux system, we are very sorry that we cannot provide more guidance on the installation, except the existing one, so we sincerely recommend you to seek instruction on the related forums.

We have updated some FAQs about installing driver, please click this link:

<http://www.tp-link.com/en/faq-1076.html>

# Contents

1. Development Environment .....	1
2. Compile the Driver .....	1
2.1. Compilation tool and kernel sources .....	1
2.2. Compile and install the Driver.....	1
3. Join the Wireless Network .....	2
3.1. Use the Graphical Interface.....	2
3.2. Use the Command.....	3
3.2.1. Identify the device .....	3
3.2.2. Create the interface .....	4
3.2.3. Change the interface status to <i>up</i> .....	4
3.2.4. Start wpa_supplicant in the background.....	4
3.2.5. Scan AP and See Results.....	4
3.2.6. Connect to the AP .....	4
3.2.7. Enable DHCP client .....	6

# 1. Development Environment

System version: Ubuntu 14.04.1

Kernel version: 3.16.0-30-generic

Gcc version: 4.8.2

## 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.

We can install compile tool gcc by command `“apt-get install gcc”`

**Note :** We recommend you use a suitable compile tool to compile our driver.

For example:

```
tplinku@tplinku-Vostro-3900:~$ cat /proc/version
Linux version 3.13.0-35-generic (buildd@roseapple) (gcc version 4.8.2 (Ubuntu 4.8.2-19ubuntu1) ) #62-Ubuntu SMP Fri Aug 15 01:58:01 UTC 2014
tplinku@tplinku-Vostro-3900:~$
```

According to the command `“cat /proc/version”`, we could see your linux system is compiled by gcc4.8.2. So we recommend you use gcc4.8.2 to compile our driver if possible.

### 2.2. Compile and install the Driver

1. Access the directory of driver.
2. Before compile, make sure the parameters in `“makefile.c”` is suitable for your compile environment of your Linux system.

```
ifeq ($(CONFIG_PLATFORM_I386_PC), y)
EXTRA_CFLAGS += -DCONFIG_LITTLE_ENDIAN
SUBARCH := $(shell uname -m | sed -e s/i.86/i386/)
ARCH ?= $(SUBARCH)
CROSS_COMPILE ?=
KVER := $(shell uname -r)
KSRC := /lib/modules/$(KVER)/build
MODDESTDIR := /lib/modules/$(KVER)/kernel/drivers/net/wireless/
INSTALL_PREFIX :=
endif
```

Explanation:

- KSRC is used to specify the kernel source path for driver compilation.
- CROSS\_COMPILE is used to specify the toolchain.
- ARCH is used to specify the target platform's CPU architectures such as arm, mips, i386 and so on.

If your Linux kernel does not support 802.11, please annotate macro "CONFIG\_IOCTL\_CFG80211" in "makefile.c".

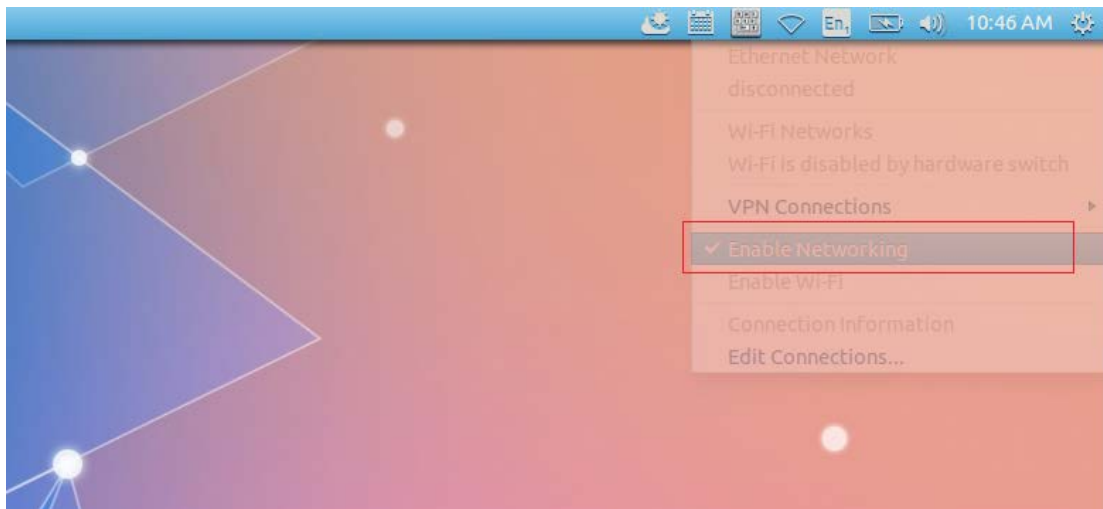
```
CONFIG_IOCTL_CFG80211=n
ifeq ($(strip &(CONFIG_IOCTL_CFG80211)),y)
EXTRA_CFLAGS += -DCONFIG_IOCTL_CFG80211 = 1
EXTRA_CFLAGS += -DRTW_USE_CFG80211_STA_EVENT = 1
endif
```

3. Type "sudo make" to compile the driver file.
4. Type "sudo make install" to install the driver file.

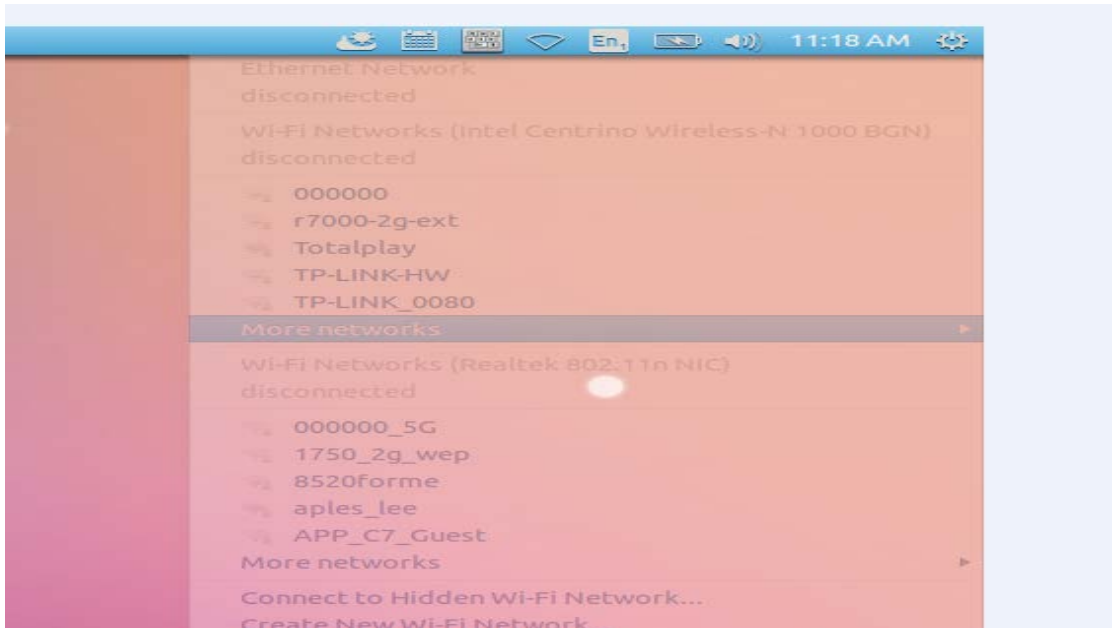
### 3. Join the Wireless Network

#### 3.1. Use the Graphical Interface

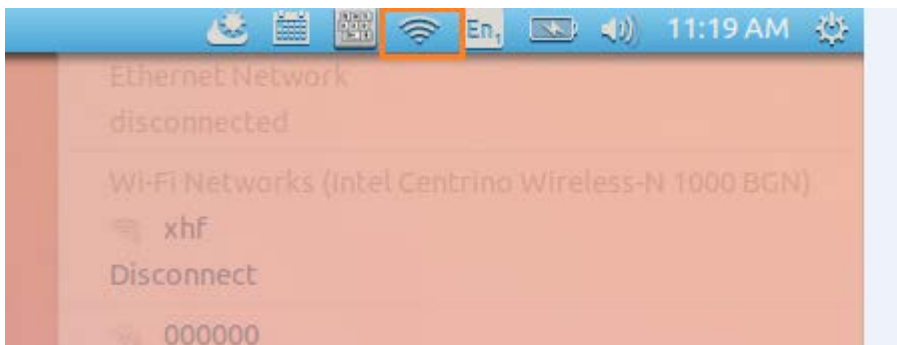
1. Enable your network shown as below.



2. Select the desired wireless network and specify the corresponding adapter interface for connection.



3. The wireless icon indicates a successful connection.



### 3.2. Use the Command

#### 3.2.1. Identify the device

Insert the USB adapter and run the following command to check if the adapter is

identified.

```
lsusb
```

### 3.2.2. Create the interface

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

```
ifconfig
```

### 3.2.3. 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.

```
rfkill unblock wifi  
ifconfig wlan1 up
```

### 3.2.4. 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
```

### 3.2.5. Scan AP and See Results

Run the following command:

```
wpa_cli -p /var/run/wpa_supplicant scan  
wpa_cli -p /var/run/wpa_supplicant scan_results
```

### 3.2.6. Connect to the AP

1) Open

Run the following command:

```
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 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
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
```

### 3.2.7. 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.

```
tplink@tplink-Inspiron-N4010:~/driver$ ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:295 errors:0 dropped:0 overruns:0 frame:0
          TX packets:295 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22543 (22.5 KB)  TX bytes:22543 (22.5 KB)

wlan1     Link encap:Ethernet  HWaddr c4:e9:84:1f:df:3c
          inet addr:192.168.1.102  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::c6e9:84ff:fe1f:df3c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:699 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2068 (2.0 KB)  TX bytes:11368 (11.3 KB)

tplink@tplink-Inspiron-N4010:~/driver$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data:
64 bytes from 192.168.1.1: icmp_seq=1 ttl=254 time=11.8 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=254 time=7.05 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=254 time=1.97 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.971/6.968/11.882/4.046 ms
tplink@tplink-Inspiron-N4010:~/driver$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.1.1    0.0.0.0         UG    0     0      0 wlan1
192.168.1.0     *              255.255.255.0  U     0     0      0 wlan1
```

*Note: Run the commands under the root account.*