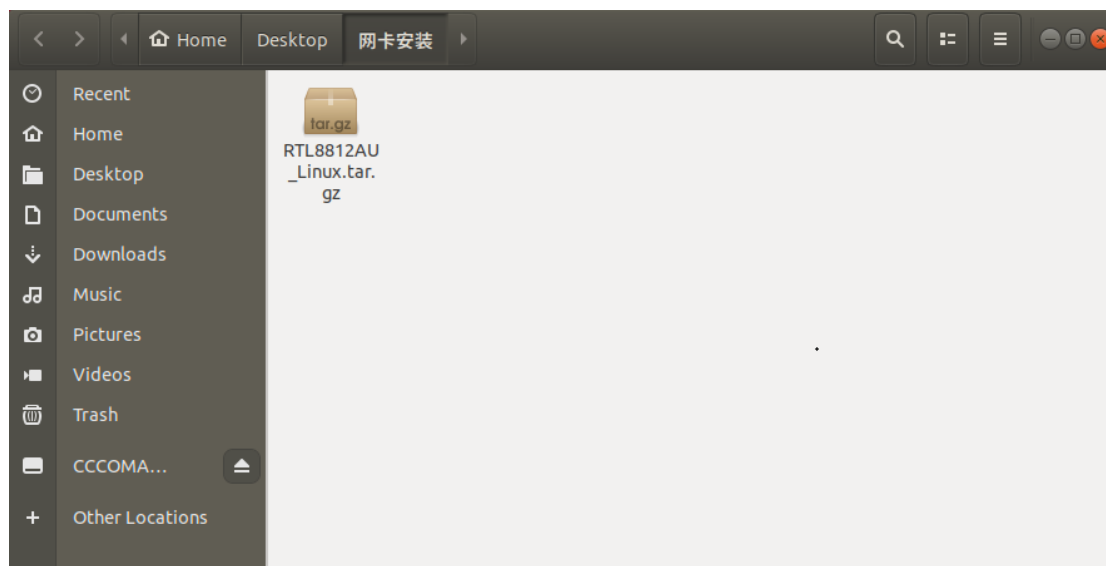


## Ubuntu 18.04（驱动最高支持 Linux 内核版本 5.8）

1) 将驱动拷贝到Linux系统桌面目录下。



2) 鼠标右键点击解压到当前文件夹，得到解压后的文件；或者在终端工具中使用命令解压：打开终端工具，界面键入：“tar xzf RTL8812AU\_Linux.tar.gz”也可以将文件解压在当前目录。

```
~:/Desktop/网卡安装$ tar xzf RTL8812AU_Linux.tar.gz
~:/Desktop/网卡安装$ ls
RTL8812AU_Linux  RTL8812AU_Linux.tar.gz
```

3) 右键点击解压后的文件，然后点击在终端中打开，进入终端管理界面，通过cd命令cd到解压后的文件目录下（也可以直接在终端管理界面通过cd命令进入到对应的目录）：

```
~:/Desktop/网卡安装$ cd RTL8812AU_Linux/
~:/Desktop/网卡安装/RTL8812AU_Linux$ ls
driver  install.sh  readme.txt  wpa_0_8.conf
~:/Desktop/网卡安装/RTL8812AU_Linux$
```

4) 进入该目录下后，执行命令“sudo bash install.sh”，执行install.sh文件。

```
~:/Desktop/网卡安装/RTL8812AU_Linux$ sudo bash install.sh
[sudo] password for root: 
```

5) 执行文件之前需要获取root权限，这里输入对应用户的密码获取root权限。

```
root@uos: ~/Desktop/网卡安装/RTL8812AU_Linux$ sudo bash install.sh
[sudo] password for root: 
```

6) 获取root权限后，即可进行驱动程序的安装过程，等待安装完成。

```
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-20-generic'
#####
Compile make driver ok!!
#####
Authentication requested [root] for install driver:
install -p -m 644 8812au.ko /lib/modules/4.15.0-20-generic/kernel/drivers/net/wireless/
/sbin/depmod -a 4.15.0-20-generic
Authentication requested [root] for remove driver:
Authentication requested [root] for insert driver:
#####
The Setup Script is completed !
#####
```

7) 安装完成之后，通过“ifconfig”命令查看接口是否存在，存在则可以正常连接无线使用网卡。

```
root@uos: ~/Desktop/网卡安装/RTL8812AU_Linux$ ifconfig
enp3s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 70:4d:7b:8b:45:c5 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

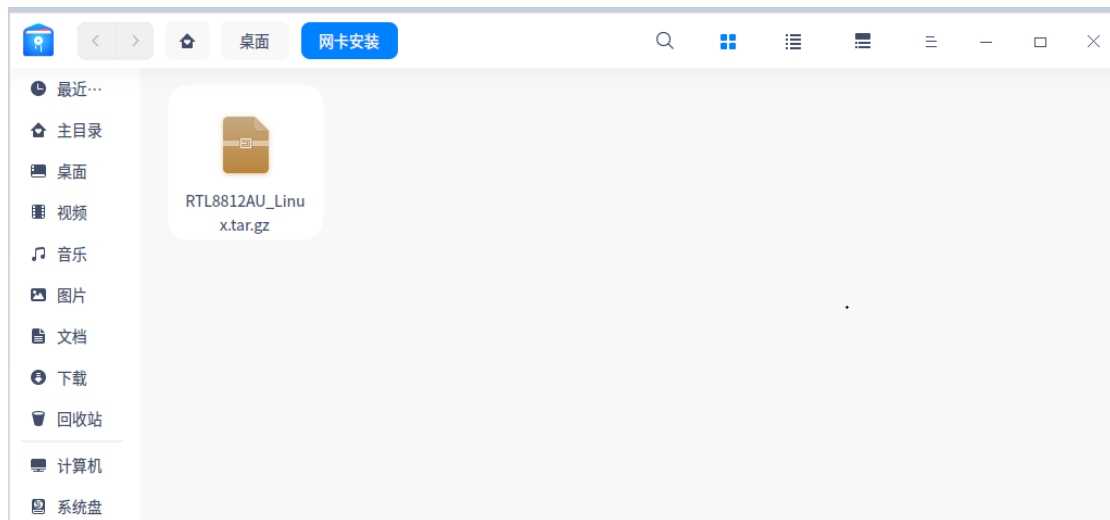
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 17814 bytes 1184373 (1.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17814 bytes 1184373 (1.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlx000c4312346f: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:0c:43:12:34:6f 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
```

注意：尽量不要在SSID或密码中使用单引号等特殊字符，否则通过图形操作界面连接无线时，可能会出现无法扫描到或者关联不上无线信号的情况；如果一定要使用这类字符，建议通过CLI命令行的操作方式关联无线信号，具体操作过程见后文的无线关联说明。

## 统信 UOS（驱动最高支持 Linux 内核版本 5.8）

1) 将驱动拷贝到Linux系统桌面目录下。



2) 鼠标右键点击解压到当前文件夹，得到解压后的文件；或者在终端工具中使用命令解压：打开终端工具，界面键入：“tar zxf RTL8812AU\_Linux.tar.gz” 也可以将文件解压在当前目录。

```
root@kali:~/Desktop/网卡安装$ ls
RTL8812AU_Linux.tar.gz
root@kali:~/Desktop/网卡安装$ tar zxf RTL8812AU_Linux.tar.gz
root@kali:~/Desktop/网卡安装$ ls
RTL8812AU_Linux  RTL8812AU_Linux.tar.gz
```

3) 右键点击解压后的文件，然后点击在终端中打开，进入终端管理界面，通过cd命令cd到解压后的文件目录下（也可以直接在终端管理界面通过cd命令进入到对应的目录）：

```
root@kali:~/Desktop/网卡安装$ cd RTL8812AU_Linux/
root@kali:~/Desktop/网卡安装/RTL8812AU_Linux$ ls
driver  install.sh  readme.txt  wpa_0_8.conf
```

4) 进入该目录下后，执行命令“sudo bash install.sh”，执行install.sh文件。

```
root@kali:~/Desktop/网卡安装/RTL8812AU_Linux$ ls
driver  install.sh  readme.txt  wpa_0_8.conf
root@kali:~/Desktop/网卡安装/RTL8812AU_Linux$ sudo bash install.sh
请输入密码
[sudo] root 的密码: 
```

5) 执行文件之前需要获取root权限，这里输入对应用户的密码获取root权限。

```
root@kali:~/Desktop/网卡安装/RTL8812AU_Linux$ ls
driver  install.sh  readme.txt  wpa_0_8.conf
root@kali:~/Desktop/网卡安装/RTL8812AU_Linux$ sudo bash install.sh
请输入密码
[sudo] root 的密码: 
```

6) 获取root权限后，即可进行驱动程序的安装过程，等待安装完成。

```
make[1]: 离开目录 "/usr/src/linux-headers-5.4.50-amd64-desktop"
#####
Compile make driver ok!!
#####
Authentication requested [root] for install driver:
install -p -m 644 8812au.ko /lib/modules/5.4.50-amd64-desktop/kernel/drivers/net/wireless/
/sbin/depmod -a 5.4.50-amd64-desktop
Authentication requested [root] for remove driver:
Authentication requested [root] for insert driver:
#####
The Setup Script is completed !
#####
```

7) 安装完成之后，通过“ifconfig”命令查看接口是否存在，存在则可以正常连接无线使用网卡。

```
root@kylin: ~/Desktop/网卡安装/RTL8812AU_Linux$ ifconfig
enp3s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 70:4d:7b:8b:45:c5 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

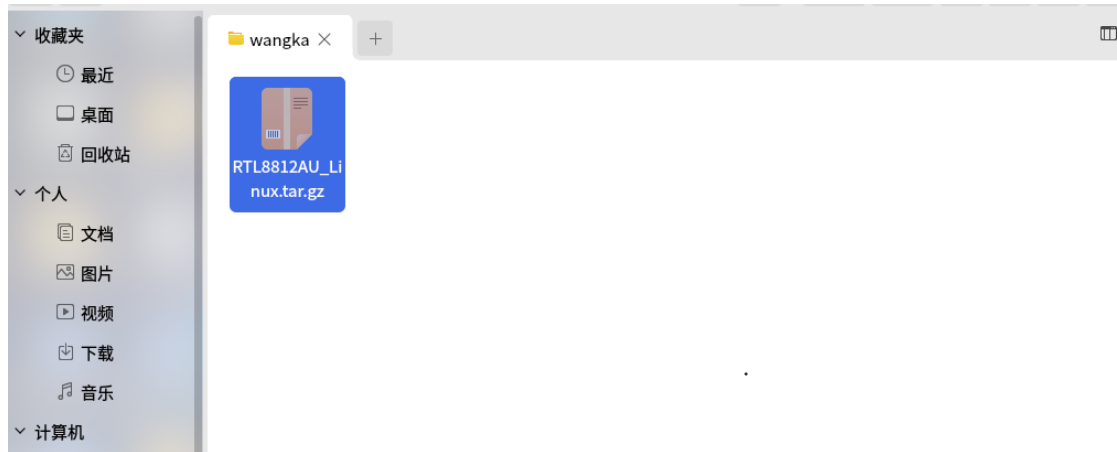
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 17814 bytes 1184373 (1.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17814 bytes 1184373 (1.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlx000c4312346f: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:0c:43:12:34:6f 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
```

注意：尽量不要在SSID或密码中使用单引号等特殊字符，否则通过图形操作界面连接无线时，可能会出现无法扫描到或者关联不上无线信号的情况；如果一定要使用这类字符，建议通过CLI命令行的操作方式关联无线信号，具体操作过程见后文的无线关联说明。

## 银河麒麟（Kylin）（驱动最高支持 Linux 内核版本 5.8）

1) 将驱动拷贝到Linux系统桌面目录下。



2) 鼠标右键点击解压到当前文件夹，得到解压后的文件；或者在终端工具中使用命令解压：打开终端工具，界面键入：“tar zxf RTL8812AU\_Linux.tar.gz” 也可以将文件解压在当前目录。

```
~:~/桌面/wangka$ ls
~:~/桌面/wangka$ tar zxf RTL8812AU_Linux.tar.gz
```

3) 右键点击解压后的文件，然后点击在终端中打开，进入终端管理界面，通过cd命令cd到解压后的文件目录下（也可以直接在终端管理界面通过cd命令进入到对应的目录）：

```
~:~/桌面/wangka$ ls
RTL8812AU_Linux  RTL8812AU_Linux.tar.gz
~:~/桌面/wangka$ cd RTL8812AU_Linux/
~:~/桌面/wangka/RTL8812AU_Linux$ ls
driver  install.sh  readme.txt  wpa_0_8.conf
```

4) 进入该目录下后，执行命令“sudo bash install.sh”，执行install.sh文件。

```
~:~/桌面/wangka/RTL8812AU_Linux$ ls
driver  install.sh  readme.txt  wpa_0_8.conf
~:~/桌面/wangka/RTL8812AU_Linux$ sudo bash install.sh
[sudo] 的密码：
```

5) 执行文件之前需要获取root权限，这里输入对应用户的密码获取root权限。

```
~:~/桌面/wangka/RTL8812AU_Linux$ ls
driver  install.sh  readme.txt  wpa_0_8.conf
~:~/桌面/wangka/RTL8812AU_Linux$ sudo bash install.sh
[sudo] 的密码：
```

6) 获取root权限后，即可进行驱动程序的安装过程，等待安装完成。

```

make[1]: 离开目录“/usr/src/linux-headers-5.4.18-15-generic”
#####
Compile make driver ok!!
#####
Authentication requested [root] for install driver:
install -p -m 644 8812au.ko /lib/modules/5.4.18-15-generic/kernel/drivers/net/wireless/
/sbin/depmod -a 5.4.18-15-generic
Authentication requested [root] for remove driver:
Authentication requested [root] for insert driver:
#####
The Setup Script is completed !
#####

```

7) 安装完成之后，通过“ifconfig”命令查看接口是否存在，存在则可以正常连接无线使用网卡。。

```

~/桌面/wangka/RTL8812AU_Linux$ ifconfig
enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 90:2b:34:65:5b:2d txqueuelen 1000 (以太网)
    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

enp4s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e0:05:c5:f1:9b:9a txqueuelen 1000 (以太网)
    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

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 (本地环回)
    RX packets 2418 bytes 180817 (180.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2418 bytes 180817 (180.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlx000c4312346f: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:0c:43:12:34:6f txqueuelen 1000 (以太网)
    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

```

注意：尽量不要在SSID或密码中使用单引号等特殊字符，否则通过图形操作界面连接无线时，可能会出现无法扫描到或者关联不上无线信号的情况；如果一定要使用这类字符，建议通过CLI命令行的操作方式关联无线信号，具体操作过程见后文的无线关联说明。

## 如何关联无线 SSID（图形操作界面/CLI 命令行）

### 通过图形操作界面连接无线：

Ubuntu，统信 UOS，麒麟系统都提供了友好的图形操作界面，当驱动安装成功之后，我们可以在图形操作界面直接连接 WIFI，这里就不详细介绍。

## 通过终端命令行操作连接无线：

### 1. 识别设备

接入 USB 无线网卡，并运行以下命令查看网卡是否被识别。

```
$ lsusb
```

### 2. 创建接口

运行以下命令检查是否创建了无线网络接口。

```
$ ifconfig
```

### 3. 将接口状态更改为 Up

检查 WLAN 接口是否正常。如果不是，请运行以下命令。这里我们以 wlan1 为例。

```
$ ifconfig wlan1 up
```

如果无法更改为 UP，请运行以下命令重新设置状态。

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

### 4. 在后台启动 wpa\_supplicant。

使用如下的命令：

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

注意：wpa\_0\_8.conf 是当前驱动程序目录中的一个文件，运行命令时转到驱动程序目录。

如果上述命令无效，请运行以下命令以结束 wpa\_supplicant 程序，然后再次运行上述命令。

```
$ killall wpa_supplicant
```

如果您的 Linux 内核不支持 802.11，请运行以下命令。

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

#### 4.1 扫描无线网络 (SSID)

运行如下命令：

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

#### 4.2 连接到接入点 (AP)

关联不同认证方式下的 SSID 所需的指令配置如下：

##### 1.开放系统认证

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.开放系统认证搭配 WEP40 加密

```
$ 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.共享密钥认证搭配 WEP40 加密



```
$ 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.开放系统搭配 WEP 104 加密

```
$ 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.预共享密钥认证搭配 WEP 104 加密

```
$ 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
```

注意：

如果 WEP 密钥是 ASCII，请运行以下命令：

```
#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""
```

如果 WEP 密钥的索引是 0-3，则运行以下命令：

```
#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 保存配置

运行以下命令保存当前的连接配置。

```
# wpa_cli -p /var/run/wpa_supplicant save_config
```

### 4.4 启用 DHCP 客户端

运行如下命令：

```
$ dhclient wlan1
```

运行命令后，适配器将获得 AP 分配的 IP。然后可以运行 ping 命令检查无线连接是否成功。

```

root@kali:/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

wlan0: 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

root@kali:/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.
64 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
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.08 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=7.86 ms
^C
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.001/2.530/7.867/2.672 ms
root@kali:/home/kali/Documents/wpa_supplicant_hostapd# route
Kernel IP routing table

```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	Archer.lan	0.0.0.0	UG	600	0	0	wlan0
192.168.1.0	0.0.0.0	255.255.255.0	U	600	0	0	wlan0

```

root@kali:/home/kali/Documents/wpa_supplicant_hostapd#

```

注意：

如果使用 ifconfig 命令确认已获得 IP 地址，并使用 ping 命令确认无线连接成功，但网络仍然不可用，则可以运行以下命令更改默认系统网关为路由器 LAN IP 地址。

\$ 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.

## 如何将网卡工作在 USB 3.0 状态

当使用网卡连接5G信号时，可以让设备工作在USB 3.0模式下，以便拥有更高的带宽使用速率，如果需要同同时使用2.4G和5G网络则不能将网卡工作在USB 3.0模式，系统默认将网卡工

作在USB 2.0的模式下。具体切换到USB 3.0的步骤如下：

1) 键入命令“lsusb -t” 回车，查看网卡接口对应安装的驱动信息，系统默认会让网卡工作在USB2.0模式下（最高速率480Mbps），因此需要手动将网卡工作在USB 3.0的模式下（最高速率5Gbps）。

```
~/Desktop/网卡安装/RTL8812AU_Linux$ lsusb -t
/: Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 5000M
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/10p, 480M
|__ Port 1: Dev 10, If 0, Class=Vendor Specific Class, Driver=rtl8812au, 480M
|__ Port 2: Dev 7, If 0, Class=Mass Storage, Driver=usb-storage, 480M
|__ Port 4: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
|__ Port 3: Dev 6, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M
|__ Port 3: Dev 6, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
|__ Port 4: Dev 4, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
|__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/6p, 480M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
|__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
```

2) 通过命令“modinfo 8812au | grep 'file'”找到刚安装的驱动的信息，找到对应的目录路径。

```
~/Desktop/网卡安装/RTL8812AU_Linux$ modinfo 8812au | grep 'file'
filename:       /lib/modules/4.15.0-20-generic/kernel/drivers/net/wireless/8812au.ko
parm:          rtw_phy_file_path:The path of phy parameter (charp)
parm:          rtw_load_phy_file:PHY File Bit Map (int)
parm:          rtw_decrypt_phy_file:Enable Decrypt PHY File (int)
```

3) 通过命令“sudo insmod （上面的驱动路径） rtw\_switch\_usb\_mode=1”将网卡工作在USB 3.0上。

```
~/Desktop/网卡安装/RTL8812AU_Linux$ sudo insmod /lib/modules/4.15.0-20-generic/kernel/drivers/net/wireless/8812au.ko rtw_switch_usb_mode=1
```

4) 完成以上步骤后，然后再通过命令“lsusb -t” 查看驱动是否工作在USB 3.0的模式，如果发现有5000Mbps就证明是工作在USB3.0的模式下，此时连接无线即可使用。

```
~/Desktop/网卡安装/RTL8812AU_Linux$ lsusb -t
/: Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 5000M
|__ Port 1: Dev 3, If 0, Class=Vendor Specific Class, Driver=rtl8812au, 5000M
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/10p, 480M
|__ Port 2: Dev 7, If 0, Class=Mass Storage, Driver=usb-storage, 480M
|__ Port 4: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
|__ Port 3: Dev 6, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M
|__ Port 3: Dev 6, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
|__ Port 4: Dev 4, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
|__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/6p, 480M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
|__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
```