Linux Driver for USB WiFi Adapters that use the RTL8812BU and RTL8822BU Chipsets

github.com/morrownr/88x2bu

88x2bu (88x2bu.ko)

- v5.8.7.4 (Realtek) (2020-09-22)
- Plus updates from the Linux community

Features

- IEEE 802.11 b/g/n/ac WiFi compliant
- 802.1x, WEP, WPA TKIP and WPA2 AES/Mixed mode for PSK and TLS (Radius)
- WPA3-SAE (Personal)
- WPS - PIN and PBC Methods
- IEEE 802.11b/g/n/ac Client mode
    - Support wireless security for WEP, WPA TKIP and WPA2 AES PSK
    - Support site survey scan and manual connect
    - Support WPA/WPA2 TLS client
    - Support power saving mode
- Soft AP mode
- WiFi-Direct
- MU-MIMO
- Mesh
- Wake on WLAN
- Supported interface modes:
    - IBSS
    - Managed
    - AP (WiFi Hotspot) (Master mode)
    - Monitor
    - P2P-client
    - P2P-GO
- USB mode control
- Log level control
- LED control
- Power saving control
- VHT control (allows 80 MHz channel width in AP mode)

Compatible CPUs

- x86, amd64
- ARM, ARM64

Compatible Kernels

- Kernels: 2.6.24 - 5.8 (Realtek)
- Kernels: 5.9 - 5.11

Tested Linux Distributions

- Arch Linux (kernel 5.4)
- Arch Linux (kernel 5.9)
- Linux Mint 20.1 (Linux Mint based on Ubuntu) (kernel 5.4)
- Linux Mint 20 (Linux Mint based on Ubuntu) (kernel 5.4)
- Linux Mint 19.3 (Linux Mint based on Ubuntu) (kernel 5.4)
- LMDE 4 (Linux Mint based on Debian) (kernel 4.19)
- Manjaro 20.1 (kernel 5.9)
- Raspberry Pi OS (2021-01-11) (ARM 32 bit) (kernel 5.4)
- Ubuntu 20.10 (kernel 5.8)
- Ubuntu 20.04 (kernel 5.4)
- Ubuntu 18.04 (kernel 5.4)

Download Locations for Tested Linux Distributions

Tested Hardware

Cudy WU1400 AC 1300Mbps USB 3.0 WiFi Adapter

Note: Cudy does a good job of posting updated source code from Realtek. Support those who support us.

- EDUP EP-AC1605GS WiFi Adapter 1300Mbps USB 3.0 High Gain Wireless Adapter
- FIDECO 6B21-AC1200M WiFi Adapter - AC1200 Dual Band

Compatible Devices

Note: Some adapter makers change the chipsets in their products while keeping the same model number so please check to confirm that the product you plan to buy has the chipset you are expecting.

- ASUS AC1300 USB-AC55 B1
- ASUS U2
- Cudy WU1400
- Dlink - DWA-181
- Dlink - DWA-182
- Edimax EW-7822ULC
- Edimax EW-7822UTC
- EDUP EP-AC1605GS
- FIDECO 6B21-AC1200M
- NetGear A6150
- TP-Link Archer T3U
- TP-Link Archer T3U Plus
- TP-Link Archer T4U V3
- TRENDnet TEW-808UBM
- Numerous additional products that are based on the supported chipsets

Installation Information

The installation instructions are for the novice user. Experienced users are welcome to alter the installation to meet their needs.

Temporary internet access is required for installation. There are numerous ways to enable temporary internet access depending on your hardware and situation. One method is to use tethering from a phone.

Another method to enable temporary internet access is to keep an ultra cheap wifi adapter that uses an in-kernel driver in your toolkit.

You will need to use the terminal interface. The quick way to open a terminal: Ctrl+Alt+T (hold down on the Ctrl and Alt keys then press the T key)

DKMS is used for the installation. DKMS is a system utility which will automatically recompile and install this driver when a new kernel is installed. DKMS is provided by and maintained by Dell.

It is recommended that you do not delete the driver directory after installation as the directory contains information and scripts that you may need in the future.

Installation Steps

Step 1: Open a terminal (Ctrl+Alt+T)

Step 2: Update the system (select the option for the OS you are using)

```
    Option for Debian based distributions such as Ubuntu, Linux Mint and the Raspberry Pi OS

    $ sudo apt-get update

    Option for Arch based distributions such as Manjaro

    $ sudo pacman -Syu
```

Step 3: Install the required packages (select the option for the OS you are using)

```
    Option for Raspberry Pi OS

    $ sudo apt-get install -y raspberrypi-kernel-headers bc build-essential dkms git

    Option for LMDE (Debian based)

    $ sudo apt-get install -y linux-headers-$(uname -r) build-essential dkms git

    Option for Linux Mint or Ubuntu (all flavors)

    $ sudo apt-get install -y dkms git

    Option for Arch based distributions such as Manjaro

    $ sudo pacman -S --noconfirm linux-headers dkms git
```

Step 4: Create a directory to hold the downloaded driver

```
$ mkdir src
```

Step 5: Move to the newly created directory

```
$ cd ~/src
```

Step 6: Download the driver

```
$ git clone https://github.com/morrownr/88x2bu.git
```

Step 7: Move to the newly created driver directory

```
$ cd ~/src/88x2bu
```

Step 8: Run a preparation script if required (Raspberry Pi *hardware* requires a preparation script)

```
    Option for 32 bit operating systems to be installed to Raspberry Pi
hardware

    $ sudo ./raspi32.sh

    Option for 64 bit operating systems to be installed to Raspberry Pi
hardware

    $ sudo ./raspi64.sh
```

Step 9: Run the installation script

```
$ sudo ./install-driver.sh
```

Step 10: Reboot

```
$ sudo reboot
```

Driver Options

A file called `88x2bu.conf` will be installed in `/etc/modeprob.d` by default.

`/etc/modprobe.d/88x2bu.conf`

This file will be read and applied to the driver on each system boot.

To edit the driver options file, run the `edit-options.sh` script.

```
$ sudo ./edit-options.sh
```

The driver options are as follows

Log level options (rtw_drv_log_level)

```
 0 = NONE (default)
 1 = ALWAYS
 2 = ERROR
 3 = WARNING
 4 = INFO
 5 = DEBUG
 6 = MAX
```

Note: You can save a log of RTW log entries by running the following in a terminal:

$ sudo ./save-log.sh

LED control options (rtw_led_ctrl)

```
 0 = Always off
 1 = Normal blink (default)
 2 = Always on
```

VHT enable options (rtw_vht_enable)

```
  0 = Disable
  1 = Enable (default)
  2 = Force auto enable (use caution)
```

Notes:

- Unless you know what you are doing, don't change the default for rtw_vht_enable.
- A non-default setting can degrade performance greatly in some operational modes.
- For AP mode, such as when you are using Hostapd, setting this option to 2 will allow 80 MHz channel width.

Power saving options (rtw_power_mgnt)

```
 0 = Disable power saving
 1 = Power saving on, minPS (default)
 2 = Power saving on, maxPS
```

Note: 0 may be useful in unattended server setups or if dropouts are experienced.

USB mode options (rtw_switch_usb_mode)

```
 0 = No switch (default)
 1 = Switch from usb 2.0 to usb 3.0
 2 = Switch from usb 3.0 to usb 2.0
```

Note: When changing USB options, a cold boot is recommended.

Information about USB 3 support

USB 3 support is off by default as there can be problems with older USB 3 ports, however, almost all USB 3 ports on modern systems work well so turning USB 3 support on should work fine for almost everyone and the difference in performance can be large.

See what your USB mode is:

```
$ lsusb -t
```

```
USB 2 =  480 Mb/s
USB 3 = 5000 Mb/s
```

iperf3 test results with USB 3 mode on

```
Bitrate
-------------
566 Mbits/sec
545 Mbits/sec
556 Mbits/sec
577 Mbits/sec
566 Mbits/sec
556 Mbits/sec
556 Mbits/sec
556 Mbits/sec
565 Mbits/sec
```

Removal of the Driver

Step 1: Open a terminal (Ctrl+Alt+T)

Step 2: Move to the driver directory

```
$ cd ~/src/88x2bu
```

Step 3: Run the removal script

```
$ sudo ./remove-driver.sh
```

Step 4: Reboot

```
$ sudo reboot
```

Recommended Router Settings for WiFi

Note: These are general recommendations based on years of experience but may not apply to your situation so testing to see if any help fix your problem is recommended.

Security: Use WPA2-AES. Do not use WPA or WPA2 mixed mode or TKIP.

Channel Width for 2.4G: Use 20 MHz fixed width. Do not use 40 MHz or 20/40 automatic.

Channel width for 5G: Using a 40 MHz fixed width may help in some situations.

Channels for 2.4G: Use 1 or 6 or 11. Do not use automatic channel selection.

Mode for 2.4G: Use G/N or B/G/N. Do not use N only.

Network names: Do not set the 2.4G Network and the 5G Network to the same name. Note: Many routers come with both networks set to the same name.

Power Saving: Set to off. This can help in some situations. If you try turning it off and you see no improvement then set it back to on so as to save electricity.

After making these changes, reboot the router.

Set regulatory domain to correct setting in OS

Check the current setting

```
$ sudo iw reg get
```

If you get 00, that is the default and may not provide optimal performance.

Find the correct setting here: http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

Set it temporarily

```
$ sudo iw reg set US
```

Note: Substitute your country code if you are not in the United States.

Set it permanently

```
$ sudo nano /etc/default/crda
```

```
Change the last line to read:
```

```
REGDOMAIN=US
```

Recommendations regarding USB

- If connecting your USB WiFi adapter to a desktop computer, use the USB ports on the rear of the computer. Why? The ports on the rear are directly connected to the motherboard which will reduce problems with interference and disconnection that can happen with front ports that use cables.

- If your USB WiFi adapter is USB 3 capable then you need to plug it into a USB 3 port.

- If you use an extension cable and your adapter is USB 3 capable, the cable needs to be USB 3 capable.

- Some USB WiFi adapters require considerable electrical current and push the capabilities of the power available via USB port. One example is devices that use the Realtek 8814au chipset. Using a powered multiport USB extension can be a good idea in cases like this.

How to disable onboard WiFi on Raspberry Pi 3B, 3B+, 3A+, 4B and Zero W.

Add the following line to /boot/config.txt

```
dtoverlay=disable-wifi
```

How to forget a saved WiFi network on a Raspberry Pi

1. Edit wpa_supplicant.conf

```
$ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

2. Delete the relevant WiFi network block (including the 'network=' and opening/closing braces.

3. Press ctrl-x followed by 'y' and enter to save the file.

4. Reboot