How to compile IW416 linux driver for x86 & ubuntu system

1. Preparation

① Downloading linux kernel source code

Here is 5.x: https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/

--linux-5.5.2.tar.xz

--linux-5.4.70.tar.xz

-- linux-5.2.2.tar.xz

Here is 4.X : https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/

--linux-4.19.35.tar.xz (in the document, we will use this one to test)

[Note]

Most drivers code supports Linux kernel from 2.6.32 to 5.5.2, so we will test above 4 versions of linux kernel.

② Installing ubuntu 16.04 to vmware player 16

If you are in China, you can change source to be that of Tsinghua University, which will make speed improved. Add the following lines to /etc/source.list

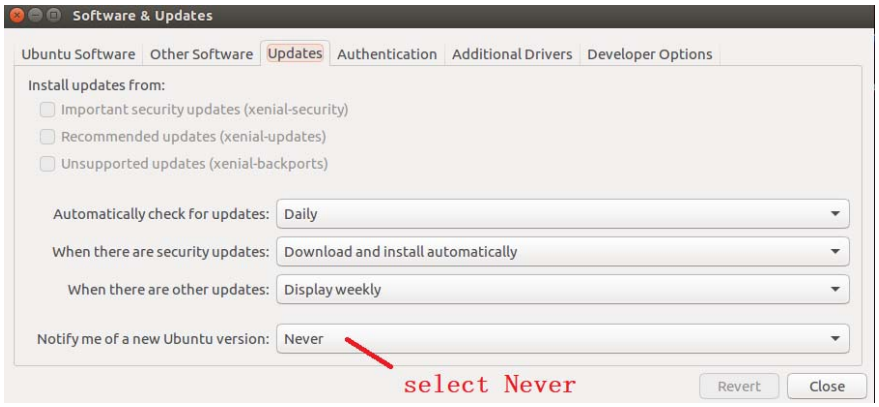
sudo cp /etc/source.list /etc/apt/source.list.bak

sudo gedit /etc/apt/source.list

```
# deb cdrom:[Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1)]/ xenial main restricted
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial main restricted
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-updates main restricted
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial universe
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-updates universe
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial multiverse
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-updates multiverse
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-backports main restricted universe multiverse
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-security main restricted
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-security universe
deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-security multiverse
```

sudo apt-get update

Then system settings--- >software & Updates -- >Updates



sudo apt-get upgrade

③ Installing dependent packages

sudo apt-get install libncurses-dev bison flex openssh-server openssh-clien libelf-dev libssl-dev

2. Compiling Linux kernel and IW416 driver On ubuntu 16.04 platform

➢ Compiling linux-4.19.35 source code

Create a new directory at ~/, for example, ~/linux-kernel, copy 4 kernel source code we downloaded before to the directory.

Then decompress them.

xz -d linux-4.19.35.tar.xz

tar -xvf linux-4.19.35.tar

cd linux-4.19.35

make menuconfig

If you don't want to change selection, save it and exit.

make -j4

If no errors occur, begin to compile IW416 Linux driver.

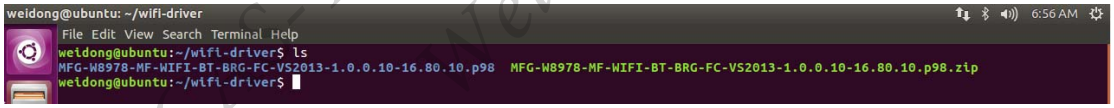
[Note] For other 3 versions of kernel source code, users can try.

➢ Compiling IW416 Linux Driver For 4.19.35 kernel

cd ~/

mkdir wifi-driver

Copy MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98.zip to the directory, then decompress it here.



cd MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98

cd Drivers

tar -zxvf SD-UAPSTA-8978-U16-MMC-W16.68.10.p101-C4X16679_V0-app-src.tgz

tar -zxvf SD-UAPSTA-8978-U16-MMC-W16.68.10.p101-C4X16679_V0-MGPL-src.tgz

tar -zxvf SD-UAPSTA-8978-U16-MMC-W16.68.10.p101-C4X16679_V0-mlan-src.tgz

cd SD-UAPSTA-8978-U16-MMC-W16.68.10.p101-C4X16679_V0-mlan-src

cd wlan_src

make KERNELDIR=/home/weidong/linux-kernel/linux-4.19.35 build

mlan.ko and sd8xxx.ko are driver for IW416. We can load drivers now.

mkdir /lib/firmware/nxp

Then copy firmware files to the directory

sudo cp ~/wifi-driver/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/bin/FwImage/*.bin /lib/firmware/nxp

insomd mlan.ko

insmod sd8xxx.ko 'cfg80211_wext=0xf wfd_name=p2p cal_data_cfg=none drv_mode=7 fw_name=nxp/w8978_SDIO_UART_UART.bin'

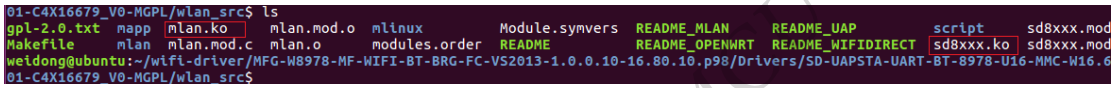
- **Compiling IW416 Linux Driver (SDIO) For the current kernel of ubuntu 16.04**

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

cd ~/wifi-driver/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/Drivers/SD-UAPSTA-UART-BT-8978-U16-MMC-W16.68.10.p101-16.26.10.p101-C4X16679_V0-MGPL/wlan_src

make clean

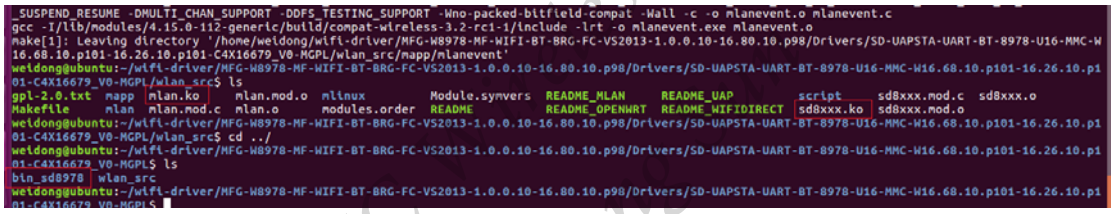
make



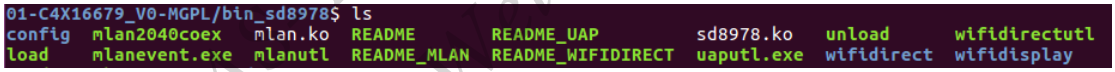
Use **make build** to compile it again.

make clean

make build



In the upper directory , bin_sd8978 subdirectory will be generated. drivers and tools are installed to the directory.



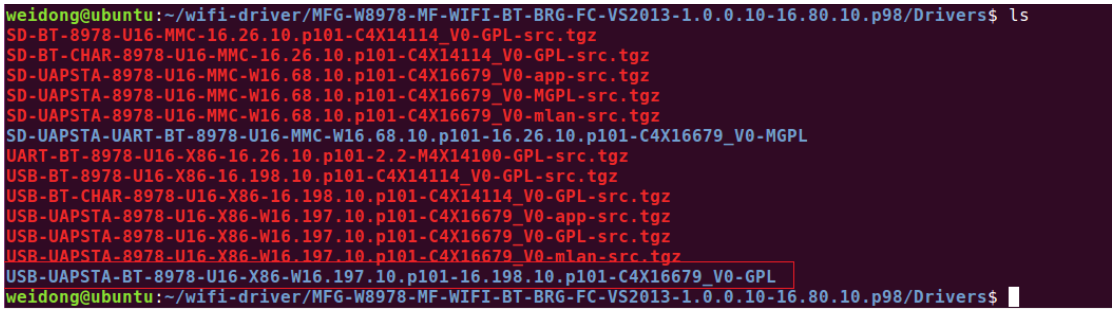
- **Compiling IW416 Linux Driver (USB) For the current kernel of ubuntu 16.04**

cd ~/wifi-driver/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/Drivers

tar -zxvf USB-UAPSTA-8978-U16-X86-W16.197.10.p101-C4X16679_V0-app-src.tgz

tar -zxvf USB-UAPSTA-8978-U16-X86-W16.197.10.p101-C4X16679_V0-GPL-src.tgz

tar -zxvf USB-UAPSTA-8978-U16-X86-W16.197.10.p101-C4X16679_V0-mlan-src.tgz

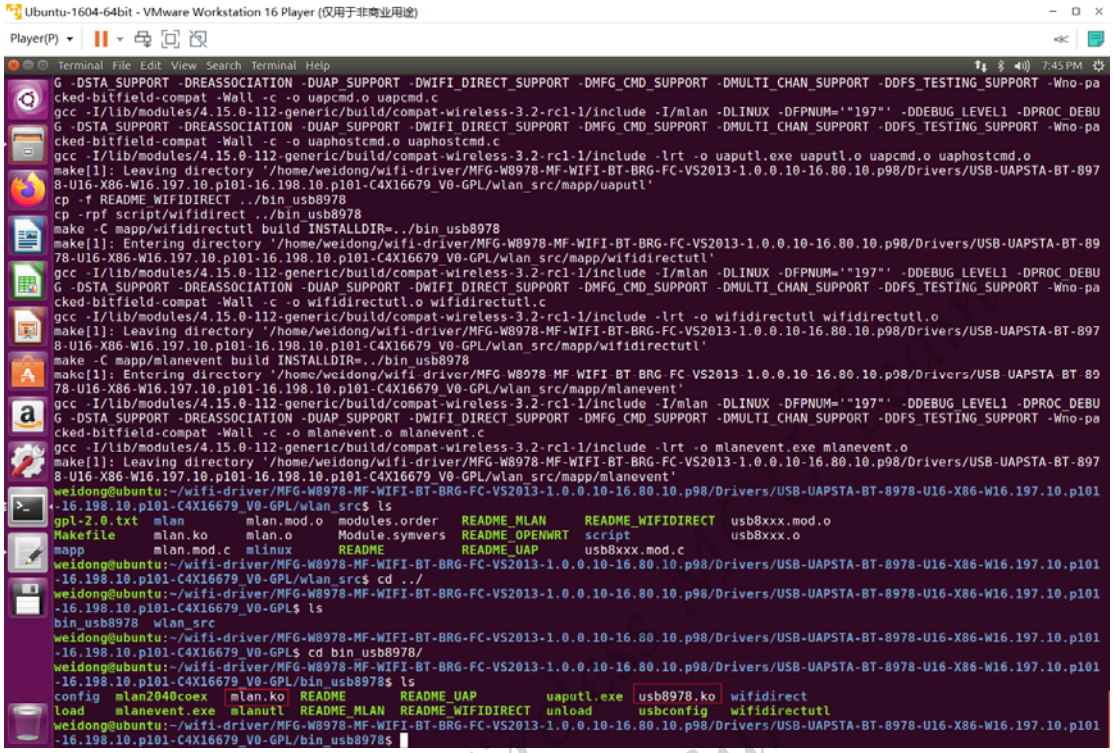


cd USB-UAPSTA-BT-8978-U16-X86-W16.197.10.p101-16.198.10.p101-C4X16679_V0-GPL

cd wlan_src

- **Compiling IW416 USB driver for 4.19.35 kernel**

make KERNELDIR=/home/weidong/linux-kernel/linux-4.19.35 build



So we can find upper directory bin_usb8978 has been created and drivers , tools all have installed here.

- **Compiling IW416 USB Driver For the current kernel of ubuntu 16.04**

rm -rf rm -rf bin_usb8978/

cd cd wlan_src/

make clean

make

Errors occurred:

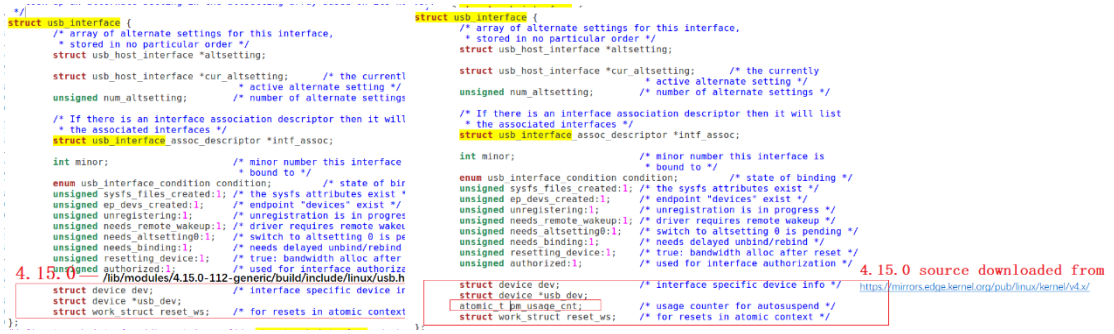
……

error: 'struct usb_interface' has no member named 'pm_usage_cnt'

atomic_t pm_usage_cnt; /* usage counter for autosuspend */

……

Download 4.15.0 kernel source from kernel source website. And compare usb.h with /lib/modules/4.15.0-112-generic/build/include/linux/usb.h :



It means the struct usb_interface in usb.h of ubuntu kernel 4.15.0-112 is not updated. So we will have to update the current kernel to 4.15.0 from kernel source website.

Download it from from kernel source website: linux-4.15.tar.xz, and decompress it, then begin to compile it.

make mrproper

make clean

make menuconfig

Don't need to change any choice, save and exit.

make -j4

After compilation is done, we begin to install modules.

sudo make modules_install

Begin to install kernel image.

sudo make install

```
weidong@ubuntu:~/linux-kernel/linux-4.15$ sudo make install
[sudo] password for weidong:
sh ./arch/x86/boot/install.sh 4.15.0 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.15.0 /boot/vmlinuz-4.15.0
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.15.0 /boot/vmlinuz-4.15.0
update-initramfs: Generating /boot/initrd.img-4.15.0
run-parts: executing /etc/kernel/postinst.d/pm-utils 4.15.0 /boot/vmlinuz-4.15.0
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.15.0 /boot/vmlinuz-4.15.0
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.15.0 /boot/vmlinuz-4.15.0
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.15.0 /boot/vmlinuz-4.15.0
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is set is no longer supported.
Found linux image: /boot/vmlinuz-4.15.0-112-generic
Found initrd image: /boot/initrd.img-4.15.0-112-generic
Found linux image: /boot/vmlinuz-4.15.0
Found initrd image: /boot/initrd.img-4.15.0
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
weidong@ubuntu:~/linux-kernel/linux-4.15$
```

Then change to 4.1.5.0 kernel version:

sudo gedit /etc/default/grub

Comment line "GRUB_HIDDEN_TIMEOUT=0" , like below;

```
 1 # If you change this file, run 'update-grub' afterwards to update
 2 # /boot/grub/grub.cfg.
 3 # For full documentation of the options in this file, see:
 4 #   info -f grub -n 'Simple configuration'
 5
 6 GRUB_DEFAULT=0
 7 #GRUB_HIDDEN_TIMEOUT=0
 8 GRUB_HIDDEN_TIMEOUT_QUIET=true
 9 GRUB_TIMEOUT=20
10 GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
11 GRUB_CMDLINE_LINUX_DEFAULT="quiet"
12 GRUB_CMDLINE_LINUX="find preseed=/preseed.cfg auto noprompt priority=critical locale=en US"
```

sudo update-grub

reboot

Then grub menu will showed, select 4.15.0 kernel item to boot ubuntu.

uname -r

```
weidong@ubuntu:~$ uname -r
4.15.0
weidong@ubuntu:~$
```

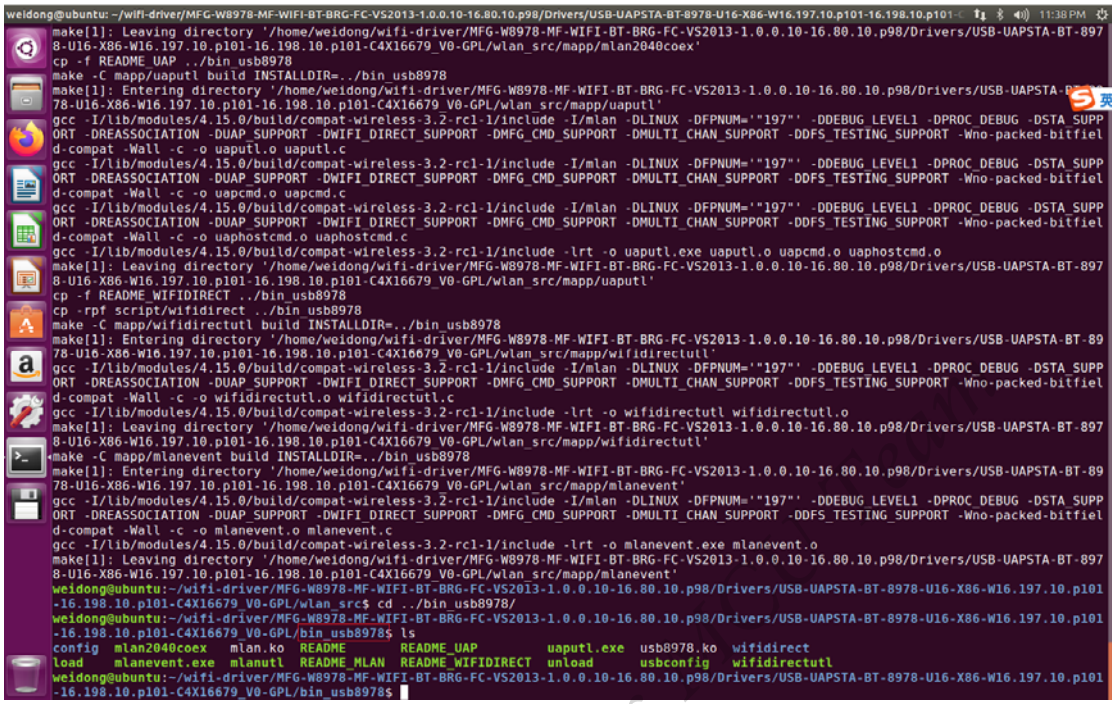
Go back to the directory of IW416 driver source code, then compile it again:

```
weidong@ubuntu:~/wifi-driver/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/Drivers/USB-UAPSTA-BT-8978-U16-X86-W16.197.10.p101
-16.198.10.p101-C4X16679_V0-GPL$ cd wlan_src/
weidong@ubuntu:~/wifi-driver/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/Drivers/USB-UAPSTA-BT-8978-U16-X86-W16.197.10.p101
-16.198.10.p101-C4X16679_V0-GPL/wlan_src$ ls
gpl-2.0.txt  Makefile  mapp  mlan  mlan.o  mlinux  README  README_MLAN  README_OPENWRT  README_UAP  README_WIFIDIRECT  script
weidong@ubuntu:~/wifi-driver/MFG-W8978-MF-WIFI-BT-BRG-FC-VS2013-1.0.0.10-16.80.10.p98/Drivers/USB-UAPSTA-BT-8978-U16-X86-W16.197.10.p101
-16.198.10.p101-C4X16679_V0-GPL/wlan_src$
```

make clean

make build

Done. Check the result of compilation:



[Summary]

This article demonstrates two ways to compile the driver for x86 linux kernel:

1. Compile the driver for 4.19.35 kernel.
2. Compile the current kernel of the driver for ubuntu 16.04.

If setting CROSS_COMPILE and ARCH, the driver can be generated with make command, but the bin_sd8978 subdirectory will not be generated. The utility needs to be compiled separately. Only for x86 arch, when compiling the driver using **make build**, the bin_sd8978 directory will be generated. Users need to pay attention to this.

3. For USB drivers of IW416, The usb.h in the original ubuntu 16.04 kernel has no update, **struct usb_interface** structure, lack of pm_usage_cnt members, so we must update the ubuntu kernel.

NXP CAS-TIC Wireless MCU Team
Weidong Sun
05/26/2021