



ASR553X 系列

USB 模块 Ubuntu 使用说明

文档版本 1.0.0

发布日期 2022-05-25

版权所有 © 2022 翱捷科技

关于本文档

本文旨在提供 ASR553X 系列芯片（2.4G 2T2R Wi-Fi 4+BLE 5.1 Combo SoC）的开发板在使用 Ubuntu 系统的台式机/笔记本上的移植和使用说明。

读者对象

本文档主要适用于以下工程师：

- 单板硬件开发工程师
- 软件工程师
- 技术支持工程师

产品型号


本文档适用于 ASR553X 系列 2.4G 2T2R Wi-Fi 4+BLE 5.1 Combo SoC 芯片。

Model	Protocol	Core	Memory	Key Functions
ASR553X	Wi-Fi 4 BLE 5.1 BLE SIG MESH	ARM STAR, 120 MHz	352 KB RAM 176 KB Boot ROM 1K-bit OTP	2x2 MIMO, SDIO 2.0/3.0, USB 2.0

版权公告

版权归 © 2022 翱捷科技股份有限公司所有。保留一切权利。未经翱捷科技股份有限公司的书面许可，不得以任何形式或手段复制、传播、转录、存储或翻译本文档的部分或所有内容。

商标声明

 ASR、翱捷和其他翱捷商标均为翱捷科技股份有限公司的商标。

本文档提及的其他所有商标名称、商标和注册商标均属其各自所有人的财产，特此声明。

防静电警告

静电放电（ESD）可能会损坏本产品。使用本产品进行操作时，须小心进行静电防护，避免静电损坏产品。

免责声明

翱捷科技股份有限公司对本文档内容不做任何形式的保证，并会对本文档内容或本文中介绍的产品进行不定期更新。

本文档仅作为使用指导，本文的所有内容不构成任何形式的担保。本文档中的信息如有变更，恕不另行通知。

本文档不负任何责任，包括使用本文档中的信息所产生的侵犯任何专有权行为的责任。

翱捷科技股份有限公司

地址：上海市浦东新区科苑路399号张江创新园10号楼9楼 邮编：201203

官网：<http://www.asrmicro.com/>

文档修订历史

日期	版本号	发布说明
2022.05	V1.0.0	首次发布。

目录

1. 概述	1
2. 环境准备	2
2.1 硬件环境准备	2
2.2 软件环境准备	3
2.3 ASR SDK 目录说明.....	4
2.4 软件版本信息	5
3. 驱动编译和安装	6
3.1 软件安装	6
3.2 驱动编译	6
3.3 驱动装载	7
4. 模式配置	9
4.1 Station 模式配置使用	9
4.2 AP 模式配置使用	10
5. Iperf 测试说明	11

ASR Confidential

表格

表 2-1 软件清单	5
表 5-1 ASR553X 吞吐参考表.....	11

ASR Confidential

图 2-1 ASR553X 开发板硬件连接示意图 2

ASR Confidential

1.

概述

本文旨在提供 ASR553X 系列芯片（2.4G 2T2R Wi-Fi 4+BLE 5.1 Combo SoC）的开发板在使用 Ubuntu 系统的台式机/笔记本上的移植和使用说明，用户可以据此准备硬件和软件环境，完成驱动的编译和安装，借助 wpa_supplicant/hostapd/iperf 等第三方工具进行 Station 和 AP 等模式的配置和使用，并完成 iperf 数据流的 demo。

ASR Confidential

2.

环境准备

2.1 硬件环境准备

- 一台千兆 Lan 口的路由器：支持 2T2R（双天线即可），使用时配置 20M 带宽。
- 一台使用 Ubuntu 系统的 PC：安装有 lperf V2.0，带 USB 2.0 接口，用于连接 ASR553X。
- 一台使用任意操作系统的普通 PC：带 RJ45 千兆网口并连接路由器，无线网卡支持 2T2R 并连接 ASR553X AP 热点，安装有 lperf V2.0。
- 一块带 USB 接口的 ASR553X 开发板：接有双天线和两根 USB 线，其中一根 USB 线接 Ubuntu PC 做数传，另一根接普通 PC，用于供电和串口 log 打印。

需注意 ASR553X 的供电问题，如果 ASR553X 串口 log 中频繁出现“1F2E3xx”，说明电源不稳且造成了 ASR553X 重启现象，建议更换 USB 接口或者 PC。

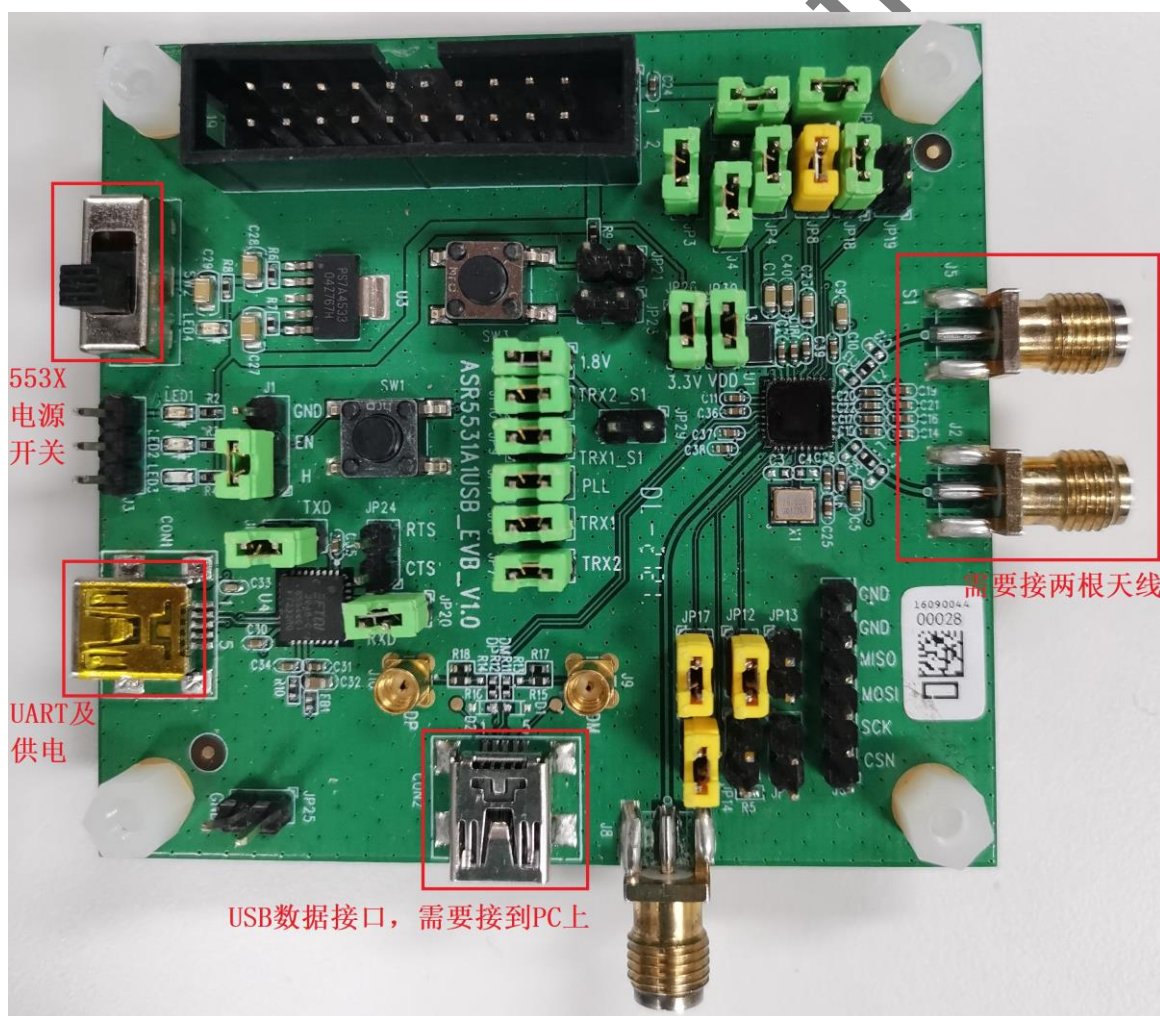


图 2-1 ASR553X 开发板硬件连接示意图

2.2 软件环境准备

- ASR553X fmacfw_asr5531_usb.bin, 固件版本号为 v3.0.1 及以上。
- ASR553X 驱动代码, 版本号为 v5.4.4.2 及以上。
- Ubuntu OS, 建议版本为 16.04。
- STA 模式下, 路由器需配置为: ssid 为 AP1、加密方式 open、channel 7、20M 带宽 (非 20/40M)、bgn mixed。
- AP 模式下, 默认热点是 “ASR_2G4”, 无密码, IP 是 192.168.0.1, 地址段是 192.168.0.2 ~192.168.0.250。

ASR Confidential

2.3 ASR SDK 目录说明

ASR553X SDK 完整目录及说明如下：

asrwifidrv



2.4 软件版本信息

移植 Ubuntu 系统所需的 GCC、Make、Toolchain、Supplicant、Hostapd 和 Iperf 等软件均通过运行 `setup_env.sh` 这个脚本来安装。

所需软件的详细清单和截图如下。

表 2-1 软件清单

软件	版本
Supplicant	2.4
Toolchain GCC	GCC 5.4
Hostapd	2.4
Iperf	2.0x
Ubuntu OS	Ubuntu 16.04 64 位
Make	4.1
DHCP	4.3.3

```
john@E470:~/asr_work/test_1826_drv/legasdioldrv$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/5/lto-wrapper
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 5.4.0-6ubuntu1~16.04.12' --with-bugurl=file:///usr/share/doc/gcc-5/README.Bugs --
ava,go,d,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-5 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-incl
osix --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale-gnu --enable-libstdcxx-debug --enable-libstdcxx-tune=yes --with-default-l
unique-object --disable-vtable-verify --enable-lto --enable-plugin --with-system-zlib --disable-browser-plugin --enable-java-awt-gtk --enab
=/usr/lib/jvm/java-1.5.0-gcj-5-amd64/jre --enable-java-home --with-jvm-root-dir=/usr/lib/jvm/java-1.5.0-gcj-5-amd64 --with-jvm-jar-dir=/usr/lib
-amd64 --with-arch-directory=amd64 --with-ecj-jar=/usr/share/java/ecj.jar --enable-objc-gc --enable-multiarch --disable-werror --with-a
with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gn
Thread model: posix
gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.12)
```

```
john@E470:~/asr_work/test_1826_drv/legasdioldrv$ iperf -v
iperf version 2.0.5 (08 Jul 2010) pthreads
john@E470:~/asr_work/test_1826_drv/legasdioldrv$
```

```
john@E470:~/asr_work/test_1826_drv/legasdioldrv$ hostapd -v
hostapd v2.4
User space daemon for IEEE 802.11 AP management,
IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator
Copyright (c) 2002-2015, Jouni Malinen <j@w1.fi> and contributors
john@E470:~/asr_work/test_1826_drv/legasdioldrv$ wpa_supplicant -v
wpa_supplicant v2.4
Copyright (c) 2003-2015, Jouni Malinen <j@w1.fi> and contributors
john@E470:~/asr_work/test_1826_drv/legasdioldrv$
```

```
john@E470:~/asr_work/test_1826_drv/legasdioldrv$ uname -a
Linux E470 4.15.0-142-generic #146~16.04.1-Ubuntu SMP Tue Apr 13 09:27:15 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
john@E470:~/asr_work/test_1826_drv/legasdioldrv$
```

```
john@E470:~/asr_work/test_1826_drv/asrwifidrv$ make -version
GNU Make 4.1
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
john@E470:~/asr_work/test_1826_drv/asrwifidrv$
```

```
john@E470:~/asr_work/test_1826_drv/legasdioldrv$ dhcpcd --version
isc-dhcpd-4.3.3
john@E470:~/asr_work/test_1826_drv/legasdioldrv$
```

3.

驱动编译和安装

3.1 软件安装

依次执行下列指令来安装相关软件的编译环境，同时 `fmacfw_asr5531_usb.bin` 会被复制到 `/lib/firmware` 目录下。

```
cd asrwifdrv/script
sudo ./setup_env.sh
```

3.2 驱动编译

在 Ubuntu 系统上编译驱动（下文用 `drv` 指代），首先需确保这 4 个宏在驱动 `Makefile` 里的配置如下，其它宏请保持默认配置。

`CONFIG_NOT_USED_DTS = y` 表示不使用 DTS
`CONFIG_ASR_USB = y` 表示使用 USB 模式
`CONFIG_ASR_SDIO = n` 表示不使用 SDIO 模式
`CONFIG_PLATFORM_I386_PC = y` 表示在 Ubuntu 平台上跑 `drv`

```
1  # without bootroom debug, not download firmware
2  CONFIG_ASR_NO_BOOTROOM ?= n
3
4  ##### use dts or not #####
5  #use CONFIG_NOT_USED_DTS to test whether need dts to compile
6
7  CONFIG_NOT_USED_DTS = y
8  ifeq ($(CONFIG_NOT_USED_DTS), y)
9  CONFIG_ASR_USB := y
10 CONFIG_ASR_SDIO := n
11 CONFIG_ASR5531 := m
12 CONFIG_MACH_PXA1826_DT := n
13 CONFIG_DEBUG_FS := y
14 CONFIG_ASR_DBG := y
15 CONFIG_DYNAMIC_DEBUG=y
16 CONFIG_OF=n
17 CONFIG_DTC=n
18 #CONFIG_WIRELESS_EXT=y
19 #CONFIG_WEXT_PRIV=y
20 endif
21
22 ##### used for debug #####
23 #CONFIG_OLD_USB := y
24 #CONFIG_FW_HAVE_NOT_MM_INFO_MSG := y
25 ASR_DRV_DEBUG_TIMER := n
26
27 ##### Platform Related #####
28 CONFIG_PLATFORM_ARM_RPI = n
29 CONFIG_PLATFORM_ARM_AARCH64 = n
30 CONFIG_PLATFORM_I386_PC = y
31 CONFIG_PLATFORM_LEWEI_ARM = n
32
```

依次执行下列指令：

```
cd asrwifdrv/
make clean
make
```

```
john@E470:~/asr_work/test_1826_drv/asrwifdrv$ make clean;make
cd drv_code && make clean
make[1]: Entering directory '/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code'
make V=s -j32 -C /lib/modules/4.15.0-142-generic/build ARCH=x86_64 CROSS_COMPILE= M=/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp src=/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp Makefile
make[2]: Entering directory '/usr/src/linux-headers-4.15.0-142-generic'
make[2]: Leaving directory '/usr/src/linux-headers-4.15.0-142-generic'
make[1]: Leaving directory '/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code'
cd drv_code && make
make[2]: Entering directory '/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code'
touch /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/Makefile
make V=s -j32 -C /lib/modules/4.15.0-142-generic/build STAGING_DIR=/lib/modules/4.15.0-142-generic/kernel/drivers/staging \
ARCH=x86_64 \
CROSS_COMPILE= \
M=/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp src=/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code \
modules
make[2]: Entering directory '/usr/src/linux-headers-4.15.0-142-generic'
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_msg_tx.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_msg_rx.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_ioctl.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_cmds.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_irqs.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_ioctl.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_txq.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_cfgfile.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_stfs.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_rx.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_tx.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_txq.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_mbn.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_mod_params.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_platform.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_target_platform.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_ate.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_host.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_usb.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_debugfs.o
CC [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr_idle_mode.o
LD [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr5531.o
Building modules stage 2.
MODPOST 1 modules
CC /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr5531.mod.o
LD [M] /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/asr5531.ko
make[2]: Leaving directory '/usr/src/linux-headers-4.15.0-142-generic'
cp /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/tmp/*.ko /home/john/asr_work/test_1826_drv/asrwifdrv/drv_code/../../out/
make[1]: Leaving directory '/home/john/asr_work/test_1826_drv/asrwifdrv/drv_code'
john@E470:~/asr_work/test_1826_drv/asrwifdrv$
```

这时系统会生成 **asr5531.ko** 文件，并自动保存在 **out** 目录下。

3.3 驱动装载

驱动安装前，请确认 ASR553X 和 Ubuntu PC 已通过一根 USB 线连接上，且在 Ubuntu PC 上，可以通过 **lsusb** 指令看到 ID 为 2ecc:5000 的 ASR553X USB 设备。

```
john@E470:~/asr_work/test_1826_drv/legasdiordrv$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID 04f2:b5c0 Chicony Electronics Co., Ltd
Bus 001 Device 010: ID 2ecc:5000
Bus 001 Device 002: ID 0bda:818b Realtek Semiconductor Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
john@E470:~/asr_work/test_1826_drv/legasdiordrv$
```

从脚本 **setup_env.sh** 来看，**asrwifdrv** 文件夹里面的 **fmacfw_asr5531_usb.bin** 已经复制到了 **/lib/firmware** 目录下，用户可再自行确认。当用户更新 drv 代码后，如果 FW 也更新了，此时需要用户手动将 **fmacfw_asr5531_usb.bin** 复制到 **/lib/firmware** 目录下。

```
john@E470:~/asr_work/test_1826_drv/legasdiordrv$ cd /lib/firmware
john@E470:/lib/firmware$ ls fmawfw_asr5531_usb.bin
fmacfw_asr5531_usb.bin
john@E470:/lib/firmware$
```

使用命令 `sudo rmmod asr5531` 卸载之前加载过的驱动。

如果 Ubuntu PC 系统没有自动加载 `cfg80211.ko` 文件，则需通过指令 `sudo modprobe cfg80211` 先加载它，然后再通过指令 `sudo insmod ./asr5531.ko nss=2` 加载 `asr5531.ko` 文件，其中 `nss=2` 表示使用 2T2R，默认是 1T1R。

加载完驱动后，执行 `dmesg` 指令可以看到 `drv` 的版本号信息等。

```
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$ sudo insmod ../out/asr5531.ko nss=2
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$ dmesg
[99264.973480] ASR wireless: Driver version:v5.4.4.2
[99264.973575] asr-platform asr-platform.0: asr_platform_probe: start
[99264.973607] asr_usb_probe: check device : bNumConfigurations 1, bDeviceClass 255, bNumInterfaces 1
[99264.973615] asr_usb_probe: check control interface: class 255, subclass 2, proto 255
[99264.973617] asr_usb_probe: check control endpoint 3
[99264.973619] asr_usb_probe: [1 of 2]check data endpoint 2
[99264.973621] asr_usb_probe: [2 of 2]check data endpoint 2
[99266.767610] usb 1-3: USB disconnect, device number 119
[99266.767627] asr_usb_resetcfg: Cannot talk to Dongle. Firmware is not UP, 100 ms ret:1
[99266.767636] asr_usb_attach: Dongle is not running now!
[99266.767640] asr_usb_attach: failed!
[99266.767679] asr_usb_probe: failed with errno -19
[99266.768070] usbcore: registered new interface driver asr5531
[99267.074055] usb 1-3: new high-speed USB device number 120 using xhci_hcd
[99267.222850] usb 1-3: New USB device found, idVendor=2ecc, idProduct=5000
[99267.222859] usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[99267.222865] usb 1-3: Product: ASR553x
[99267.222869] usb 1-3: Manufacturer: ASR
[99267.223779] asr_usb_probe: check device : bNumConfigurations 1, bDeviceClass 255, bNumInterfaces 1
[99267.223796] asr_usb_probe: check control interface: class 255, subclass 2, proto 255
[99267.223803] asr_usb_probe: check control endpoint 3
[99267.223811] asr_usb_probe: [1 of 2]check data endpoint 2
[99267.223817] asr_usb_probe: [2 of 2]check data endpoint 2
[99267.224243] usb 1-3: mac_param=NULL
[99267.224251] usb 1-3: failed read mac address, will use random address
[99267.224281] usb 1-3: mac address is AA:00:CD:EF:4E:7A
[99267.224922] usb 1-3: asr_usb_send_cmd: tx msg 0,0
[99267.248301] usb 1-3: asr_rx_handle_msg MM_RESET_CFM T:0 - I:1 0->11 l:0
[99267.248328] usb 1-3: asr_usb_send_cmd: tx msg 0,4
[99267.248811] usb 1-3: asr_rx_handle_msg MM_VERSION_CFM T:0 - I:5 0->11 l:24
[99267.248833] usb 1-3: asr_usb_send_cmd: tx msg 0,94
[99267.249713] usb 1-3: asr_rx_handle_msg MM_FW_SOFTVERSION_CFM T:0 - I:95 0->11 l:32
[99267.249742] usb 1-3: ASR: fw version(v3.0.1)
[99267.249752] usb 1-3: asr_usb_send_cmd: tx msg 0,104
[99267.250801] usb 1-3: asr_rx_handle_msg MM_GET_INFO_CFM T:0 - I:105 0->11 l:12
[99267.250823] usb 1-3: asr_read_mm_info:vif_num=1,sta_num=2
[99267.250830] ieee80211 phy59: PHY features: [NSS=2][CHBW=40]
[99267.250837] ieee80211 phy59: FW features: [BCN][AUTOBCN][HWSCAN][CMON][MROLE][AMPDU][CHNL_CTXT][REORD][UMAC]
[99267.250843] usb 1-3: HT supp 1
[99267.250849] usb 1-3: asr_usb_send_cmd: tx msg 5,0
[99267.251918] usb 1-3: asr_rx_handle_msg ME_CONFIG_CFM T:5 - I:1 5->11 l:0
[99267.252082] usb 1-3: asr_usb_send_cmd: tx msg 5,2
[99267.252880] usb 1-3: asr_rx_handle_msg ME_CHAN_CONFIG_CFM T:5 - I:3 5->11 l:0
```

4.

模式配置

4.1 Station 模式配置使用

参考下列指令，使 ASR553X 连接 AP，并获取 IP 地址。有关详情，请参见 `usb_sta.sh` 和 `asr_wpa_supplicant.conf` 文件。

```
ifconfig wlan0 up
```

```
wpa_supplicant -Dnl80211 -iwlan0 -c asr_wpa_supplicant.conf -B -d -t -f /tmp/wpa.txt
```

```
dhclient -i wlan0
```

```
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$ sudo ./usb_sta.sh
wlan0    Link encap:Ethernet  HWaddr aa:00:cd:ef:d5:5f
         inet addr:192.168.6.105  Bcast:192.168.6.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:6  errors:0  dropped:0  overruns:0  frame:0
         TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:1264 (1.2 KB)  TX bytes:0 (0.0 B)

john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$
```

```
[99081.244227] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
[99083.848851] usb 1-3: DHCP:0,2,2,67,68,192:168:6:105,AA:00:CD:EF:D5:5F
[99083.910995] usb 1-3: DHCP:0,2,5,67,68,192:168:6:105,AA:00:CD:EF:D5:5F
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$ dmesg |grep wlan0 |grep ready
[99075.751635] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[99075.756932] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[99081.244227] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$
```


4.2 AP 模式配置使用

参考下列指令进行 AP 模式的配置、启动以及使其它设备连接该 AP。有关详情，请参见 `usb_ap.sh`，`asr_hostapd.conf` 和 `asr_dhcpd.conf` 文件。

```
sudo ifconfig wlan0 up
sudo ifconfig wlan0 192.168.0.1/24
sudo hostapd -B /etc/hostapd/hostapd.conf
sudo dhcpd wlan0 &
```

注意，AP 模式下，由于 FW 限制，目前最多只能连接两个 client 设备。

ASR553X 做 AP 时，默认热点名称为 “ASR_2G4”，无密码，信道 6。

```
john@E470:~/asr_work/test_1826_drv/asrwifidrv$ cd script/
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$ sudo ./usb_ap.sh
wlan0    Link encap:Ethernet  HWaddr aa:00:cd:ef:37:cc
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$
```

```
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$ dmesg |grep wlan0|grep ready
[98915.087587] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[98915.092890] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[98919.529486] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[98919.568000] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
john@E470:~/asr_work/test_1826_drv/asrwifidrv/script$
```


5.

Iperf 测试说明

说明：Iperf 测试是在非屏蔽室环境下进行的，测得的速率仅供参考。

表 5-1 ASR553X 吞吐参考表

模式	Iperf	指标 (Mbps)	测试步骤说明（粗体表示在 ASR553X 上运行。STA 模式下，ASR553X 的 IP 为 192.168.43.10，做 AP 时，ASR553X 的 IP 为 192.168.0.1）
Station 模式	tcp tx	60	server: iperf -i1 -s -p4439 client: iperf -i1 -c192.168.43.1 -w2M -p4439 -t60
	tcp rx	66.6	server: iperf -i1 -s -p4439 client: iperf -i1 -c192.168.43.10 -w2M -p4439 -t60
	udp tx	85	server: iperf -i1 -s -p4439 -u client: iperf -i1 -c192.168.43.1 -w2M -p4439 -t60 -u -b300M
	udp rx	94.2	server: iperf -i1 -s -p4439 -u client: iperf -i1 -c192.168.43.10 -w2M -p4439 -t60 -u -b300M
AP 模式	tcp tx	70.6	server: iperf -i1 -s -p4439 client: iperf -i1 -c192.168.0.10 -w2M -p4439 -t60
	tcp rx	70.3	server: iperf -i1 -s -p4439 client: iperf -i1 -c192.168.0.1 -w2M -p4439 -t60
	udp tx	98	server: iperf -i1 -s -p4439 -u client: iperf -i1 -c192.168.0.10 -w2M -p4439 -t60 -u -b300M
	udp rx	97.9	server: iperf -i1 -s -p4439 -u client: iperf -i1 -c192.168.0.1 -w2M -p4439 -t60 -u -b300M

如需查看 Iperf 测试 log，请参考 `doc/iperf_usb/` 目录下的 log 文件。