

AIW FAQ

Porting AIW-173 WiFi driver for RK3588 Debian Linux OS			
Category	AIW	Date	2025/12/2
Keyword	AIW-173HQ-GI1, WiFi, Debian, Rockchip RK3588, ASR-A501		
HW model	ASR-A501 (Rockchip RK3588)		
OS version	Debian 12 (Linux kernel version: 6.1.75-rt16) (Image version: DIV110057) BSP version: RK3588_RISC_DIV110057		

Category

1.	Build Environment on Host.....	2
2.	Download the driver and patches and Setting Environment	2
3.	Build Kernel Image.....	2
4.	Build WLAN Driver	3
5.	Flash Image	3
6.	Build wpa_supplicant and hostapd [On target device]	3
7.	Copy FW files [On target device]	4
8.	Install the driver and bring up the interface [On target device]	4
9.	Use wpa_supplicant command to connect a WiFi 6G AP [On target device]	5

This document guides the porting of the AIW-173HQ-GI1 WiFi driver to the Advantech ASR-A501 (RK3588/Debian 12). It covers the complete workflow: setting up the build environment, patching and rebuilding the kernel, compiling userspace tools (wpa_supplicant and hostapd) on the target, and verifying the WiFi 6E connection."

1. Build Environment on Host

You can have more detail about the compiling environment setup from the [MEMO RK3588 Debian Build Process manual](#). Before proceed the following steps, please make sure you start and enter to the docker environment

2. Download the driver and patches and Setting Environment

The driver and kernel need to be applied with the patch for driver porting. Please download the following 2 patches and follow the steps to apply. If possible, you can apply the patch manually to avoid unexpected conflicts.

Please download the "[kernel.patch](#)" and "[driver.patch](#)".

Command and Response	Description
\$ cd ~/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08	Download the AIW-173 WiFi driver
\$ sudo git clone ssh://adv_ae@172.22.24.155:/volume1/git-server/AIW_173_driver	
\$ cd AIW_173_driver/AIO/drivers/core_tech_modules	Note: password=123456
\$ cp include/linux/mhi.h <kernel_path>/include/linux/	Copy header files to the kernel
\$ cp include/trace/events/qrtr.h <kernel_path>/include/trace/events/	
\$ cp inc/cnss2.h <kernel_path>/include/net/	
\$ cp inc/cnss_nl.h <kernel_path>/include/net/	
\$ cp inc/cnss_utils.h <kernel_path>/include/net/	
\$ cp inc/cnss_prealloc.h <kernel_path>/include/net/	
kernel path=~/BSP/ ASRA501A2EG90AIM38DBV110057_2025-04-08/kernel	
\$ cd ~/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08	Copy the kernel patch file above to the BSP
\$ cp kernel.patch ./	
\$ git apply kernel.patch	Apply the patch (Fix conflicts manually if happened)
\$ cp driver.patch ./	Copy the driver patch file above to the BSP
\$ git apply driver.patch	Apply the patch (Fix conflicts manually if happened)

3. Build Kernel Image

After apply the patch for kernel, it needs to rebuild the kernel to make it effect.

Command and Response	Description
\$ cd ~/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08	Clean the kernel and rebuild it again.
\$./build.sh clean:kernel	
\$./build.sh kernel	

4. Build WLAN Driver

Please make sure the patch is applied before you build the WiFi driver. You can have more detail about how to set up cross compiling environment from the [MEMO RK3588 Debian Build Process manual](#).

Command and Response	Description
<pre>\$ cd ~/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08/AIW_173_driver/AIO/build \$ make KERNELARCH=arm64 ARCH=arm64 KERNELPATH=/home/adv/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08/kernel/ KERNEL_SRC=/home/adv/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08/kernel/ CROSS_COMPILE=/home/adv/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08/prebuilts/gcc/linux-x86/aarch64/gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu- CONFIG_FULL_CORE_BUILD=y CONFIG_PERF_BUILD=y drivers</pre>	Go to the driver source code and compile it
<pre>\$ ls ~/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08/AIW_173_driver/AIO/rootfs-tr-f10.build/lib/modules/ wlan_cnss_core_pcie.ko wlan.ko</pre>	After the compiling, you can see wlan.ko and wlan_cnss_core_pcie.ko drivers.

5. Flash Image

For image building and flashing , you can refer “4.2 Image building” and “5.3 Flash image” in the [MEMO RK3588 Debian Build Process manual](#).

6. Build wpa_supplicant and hostapd [On target device]

Because the ASR-A501 rootfs source code is confidential to share, it needs to build wpa_supplicant and hostapd commands on the ASR-A501 device directly. So, the following steps from this chapter are on the ASR-A501 device.

Command and Response	Description
<pre>\$ sudo apt-get install libssl3 libssl-dev libnl-3-dev libnl-genl-3-dev libnl-route-3-dev \$ sudo git clone ssh://adv_ae@172.22.24.155:/volume1/git-server/AIW_173_driver/etc</pre>	Install extra packages for compiling. Download the driver include the hostapd and wpa_supplicant source code. Note: Connect an ethernet cable to LAN port.

For the “wpa_supplicant”, please download “[wpa_supplicant_config.patch](#)” and “[wpa_supplicant_makefile.patch](#)”.

Command and Response	Description
<pre>\$ cd /etc/AIW_173_driver/AIO/apps/hostap/wpa_supplicant \$ cp wpa_supplicant_config.patch ./ \$ git apply wpa_supplicant_config.patch \$ cp wpa_supplicant_makefile.patch ./ \$ git apply wpa_supplicant_makefile.patch \$ make</pre>	Copy the patch files above to the /etc. Apply the patch to .config and Makefile for compiling wpa_supplicant. Compile the wpa_supplicant command

For the “hostapd”, please download “[hostapd_config.patch](#)” and “[hostapd_makefile.patch](#)”.

Command and Response	Description
<pre>\$ cd /etc/AIW_173_driver/AIO/apps/hostap/hostapd \$ cp hostapd_config.patch ./ \$ git apply hostapd_config.patch \$ cp hostapd_makefile.patch ./ \$ git apply hostapd_makefile.patch \$ make</pre>	Copy the patch files above to the /etc. Apply the patch to .config and Makefile for compiling hostapd. Compile the hostapd command

After the compiling, copy the “wpa_supplicant” and “hostapd” commands to the “/sbin” to replace the original commands.

Command and Response	Description
<pre>\$ wpa_supplicant -v wpa_supplicant v2.11-devel Copyright © 2003-2022, Jouni Malinen j@w1.fi and contributors</pre>	Confirm the wpa_supplicant version is “2.11”
<pre>\$ hostapd -v hostapd v2.11-devel User space daemon for IEEE 802.11 AP management, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator Copyright © 2003-2022, Jouni Malinen j@w1.fi and contributors</pre>	Confirm the hostapd version is “2.11”

7. Copy FW files [On target device]

Before install the driver, it needs to copy the FW files to the correspond path for driver loading. Otherwise, it will kernel panic.

Command and Response	Description
<pre>\$ cp -fr /etc/AIW_173_driver/AIO/drivers/firmware/WLAN-firmware/* /lib/firmware</pre>	Copy the FW files from driver source code to /lib/firmware

8. Install the driver and bring up the interface [On target device]

To use the wpa_supplicant or hostapd command manually, it needs to disable the NetworkManager service. Here is an example to show you how to connect a WiFi 6G AP.

Command and Response	Description
<pre>\$ systemctl disable NetworkManager \$ reboot</pre>	Disable the NetworkManager service to

<pre>\$ modprobe cfg80211 (optional) \$ insmod wlan_cnss_core_pcie.ko \$ insmod wlan.ko country_code=US \$ iw reg set US</pre>	avoid conflicts. A reboot is required to make it effect. Insert 2 .ko files to initial the driver. The value country_code should be the same as “iw reg set” and it’s different depend on the connected WiFi AP or region.
<pre>\$ ifconfig -a</pre> <pre>wlan0 Link encap: Ethernet HWaddr 00:0E:8E:C5:41:19 UP BROADCAST 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)</pre>	After loading the driver, use “ifconfig” to check wlan0 interface.

9. Use wpa_supplicant command to connect a WiFi 6G AP [On target device]

Command and Response	Description
<pre>\$ vi /etc/wpa_supplicant.conf</pre> <pre>ctrl_interface=/var/run/wpa_supplicant update_config=1 ap_scan=1 fast_reauth=1</pre> <pre>pmf=1 sae_pwe=1</pre> <pre>network={ ssid="XXXX" psk="XXXXX" key_mgmt=SAE ieee80211w=1 proto=RSN group=CCMP TKIP pairwise=CCMP TKIP scan_ssid=1 }</pre>	Create and edit the wpa_supplicant.conf file to configure WiFi connection parameters such like WiFi AP SSID and password.
<pre>\$ wpa_supplicant -D nl80211 -i wlan0 -c /etc/wpa_supplicant.conf</pre> <pre>\$ dhclient wlan0 (optional)</pre> <pre>\$ ifconfig -a</pre> <pre>wlan0 Link encap: Ethernet HWaddr 00:0E:8E:C5:41:19 inet addr:192.168.1.170 Bcast:192.168.1.255 Mask:255.255.255.0 inet6 addr: fe80:af88:76b5:ca1b:148c/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:52 errors:0 dropped:0 overruns:0 frame:0 TX packets:37 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:15176 (14.8 KiB) TX bytes:4264 (4.1 KiB)</pre>	Establish a WiFi connection If the interface is not assigned an IP, you can use “dhclient” command to request DHCP assignment. Then you should be able to see that “wlan0” interface has been assigned an IP.