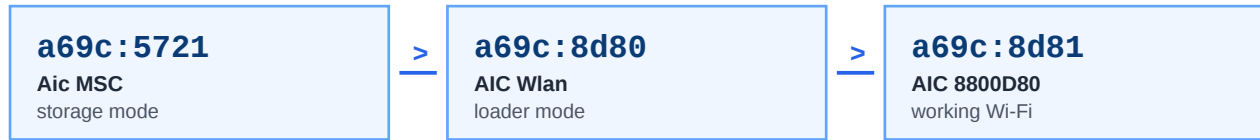


How to get the AICSemI a69c:8d81 USB Wi-Fi module working under Olimage

Expected USB state transition



What the screenshots in this PDF mean

Dark terminal panels show commands to type and examples of output to expect. The exact bus/device numbers and Wi-Fi interface name can differ between boards.

This guide is for the AICSemI AIC8800D80-based USB Wi-Fi module that appears as 'a69c:8d81' after firmware loads.

OLIMEX A20 release images:

<https://images.olimex.com/release/a20/>

OLIMEX product page:

<https://www.olimex.com/Products/USB-Modules/WiFi/USB-WIFI6-5G/>

Driver source used:

<https://github.com/shenmintao/aic8800d80/tree/legacy-mcu1>

OLIMAGE source:

<https://github.com/OLIMEX/olimage>

Short summary

In simple terms, the work is:

1. Write the latest OLIMAGE Bookworm base image to a microSD card.
2. Boot the A20-OLinuXino-LIME2 board with serial console, Ethernet, and 5 V DC power connected.
3. Run 'apt update', then 'apt dist-upgrade', then reboot.
4. Install the matching OLIMEX kernel headers separately.
5. Install build tools, clone the AICSemI 'legacy-mcu1' driver, and build it with DKMS.
6. Reboot or replug the adapter, confirm the USB state changes to 'a69c:8d81', then connect with 'nmtui'.

The full OLIMEX kernel source workaround is no longer needed with the tested release packages.

What You Need

Hardware:

1. A20-OLinuXino-LIME2 or similar OLIMAGE-supported board.
2. AICSemi AIC8800D80-based USB Wi-Fi adapter.
3. microSD card with the OLIMAGE Bookworm base image.
4. USB serial console cable.
5. Ethernet cable for Internet access during setup.
6. Stable 5 V DC power supply.

Software/network:

1. BalenaEtcher or another image writer.
2. Root shell on the board.
3. Internet access from the board.
4. Public OLIMEX package repository enabled.

Check free disk space after boot:

TERM Terminal command

```
df -h / /usr/src /tmp
```

The driver source is small, but package upgrades and DKMS builds need working free space. A few GB free is comfortable.

Tested Setup

These instructions were tested on a clean card prepared from:

TERM Expected terminal output

```
A20-OLinuXino-bookworm-base-20260323-113959.img.7z
```

from:

TERM Expected terminal output

```
https://images.olimex.com/release/a20/
```

After public-release upgrade, the tested OLIMEX packages were:

TERM Expected terminal output

```
linux-image-5.10.180-olimex 5.10.180-20260827-071822
linux-image-olimex          5.10.180-20260827-071822
linux-headers-5.10.180-olimex 5.10.180-20260827-071822
linux-headers-olimex        5.10.180-20260827-071822
```

The running kernel was:

TERM Expected terminal output

```
5.10.180-olinux
```

The product names tested were Olimex 'USB-WIFI6-5G' and 'USB-WIFI6-5G-ANT'. Use the USB ID/state transition to choose this guide, even if the product label or product-family name is similar to another module:

TERM Expected terminal output

```
Initial storage mode:      a69c:5721  aicsemi Aic MSC
Post-eject loader mode:   a69c:8d80  aicsemi AIC Wlan
Operational Linux mode:   a69c:8d81  AICSemi AIC 8800D80
Operational Windows mode: a69c:8d83  AIC8800D80 USB WiFi
Linux driver used:        aic_load_fw + aic8800_fdrv
```

The tested units reported:

TERM Expected terminal output

```
chip_id=7, chip_mcu_id = 1
```

That MCU revision needs the 'legacy-mcu1' driver/firmware branch. The normal/main firmware branch can fail with:

TERM Expected terminal output

```
cmd timed-out
cmd queue crashed
bin upload fail: 170400
```

This is not the Realtek module. The Realtek version uses '0bda:c811' and needs a different guide.

1. Prepare and Boot the Board

Download the latest A20 OLIMAGE Bookworm base image from:

TERM Expected terminal output

```
https://images.olinux.com/release/a20/
```

Write the image to a microSD card with BalenaEtcher.

Insert the card into the A20-OLinuXino-LIME2 board.

Connect:

1. USB serial console cable.
2. Ethernet cable.
3. 5 V DC power supply.

Log in over the serial console and use a root shell.

2. Update OLIMAGE from the Public Repository

Do not add the staging repository for normal customer instructions. The fixed packages were pushed to the public release repository.

Run:

TERM Terminal command

```
apt update
apt dist-upgrade
```

If asked to continue, answer:

TERM Expected terminal output

```
Y
```

This can take a long time on A20/LIME2-class boards. The upgrade may update many Debian packages as well as the OLIMEX kernel image. Do not interrupt 'apt' or 'dpkg' unless the board is clearly no longer responding.

Reboot after the upgrade:

TERM Terminal command

```
reboot
```

After reboot, confirm the kernel and OLIMEX kernel image package:

TERM Terminal command

```
uname -r
dpkg -l 'linux-*olimax*' | grep '^ii'
```

Expected kernel:

TERM Expected terminal output

```
5.10.180-olimax
```

3. Install the Matching Kernel Headers

Important: 'apt dist-upgrade' updates the kernel image, but the kernel headers may not be installed automatically.

Install the matching headers separately:

TERM Terminal command

```
apt install -y linux-headers-olimax linux-headers-$(uname -r)
```

Confirm the kernel image and headers are installed:

TERM Terminal command

```
uname -r && dpkg -l 'linux-*olimax*' | grep '^ii'
```

Expected package family:

TERM Expected terminal output

```
linux-headers-5.10.180-olimax
linux-headers-olimax
linux-image-5.10.180-olimax
linux-image-olimax
```

The tested fixed release package version was:

TERM Expected terminal output

```
5.10.180-20260827-071822
```

4. Verify the DKMS Build Tree

Check the files that are needed by external module builds:

TERM Terminal command

```
for f in scripts/module.lds include/generated/uapi/linux/version.h scripts/mod/modpost include/generated/utsrelease.h include/generated/timeconst.h include/generated/autoconf.h include/config/kernel.release; do test -e /lib/modules/$(uname -r)/build/$f && echo "OK $f" || echo "MISSING $f"; done
```

Expected:

TERM Expected terminal output

```
OK scripts/module.lds
OK include/generated/uapi/linux/version.h
OK scripts/mod/modpost
OK include/generated/utsrelease.h
OK include/generated/timeconst.h
OK include/generated/autoconf.h
OK include/config/kernel.release
```

Note: 'include/linux/version.h' is not expected in this OLIMEX header package. The generated header used by Kbuild is:

TERM Expected terminal output

```
include/generated/uapi/linux/version.h
```

5. Identify the AICSemI USB Device

Plug the AICSemI adapter and check:

TERM Terminal command

```
lsusb
lsblk -o NAME,MODEL,SIZE,RO,TYPE
```

Before the driver/modeswitch rule runs, the adapter may appear as a tiny read-only mass-storage device:

TERM Expected terminal output

```
Bus 001 Device 002: ID a69c:5721 aicsemi Aic MSC
sda flash 3.6M 1 disk
```

If it is still in storage mode and you need to switch it manually, eject the storage device shown by 'lsblk':

TERM Terminal command

```
eject /dev/sda
```

Use the correct '/dev/sdX' from your board. After eject, it may appear as:

TERM Expected terminal output

```
a69c:8d80 aicsemi AIC wlan
```

After the driver and firmware load successfully, it should appear as:

TERM Expected terminal output

```
a69c:8d81 AICSem i AIC 8800D80
```

6. Install Build Tools

Install the build dependencies and USB switching tools:

TERM Terminal command

```
apt install -y git dkms build-essential bc bison flex libssl-dev libelf-dev eject usb-modesw  
itch iw network-manager
```

If ‘apt’ says a lock is held by unattended upgrades, wait. To check package activity:

TERM Terminal command

```
ps -ef | grep -E 'apt|dpkg|unattended|mandb' | grep -v grep
```

This helper is normal and does not mean apt is busy:

TERM Expected terminal output

```
/usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signa  
l
```

If the board lost power or hung during package installation, check package state:

TERM Terminal command

```
dpkg --audit
```

If it reports half-configured packages, repair them with:

TERM Terminal command

```
dpkg --configure -a
```

7. Download and Build the AICSem Driver

Clone the working driver branch:

TERM Terminal command

```
cd /root
git clone --branch legacy-mcu1 https://github.com/shenmintao/aic8800d80.git aic8800d80-legacy-mcu1
cd /root/aic8800d80-legacy-mcu1
```

Run the installer:

TERM Terminal command

```
bash ./install.sh
```

On A20/LIME2-class boards, the DKMS compile step can take several minutes. Do not interrupt it unless it clearly failed or the board is no longer responding.

Successful install lines look like:

TERM Expected terminal output

```
Driver installed via DKMS
DKMS: aic8800/1.0.0, 5.10.180-olimax, armv7l: installed
```

The installer also installs firmware and an auto-eject udev rule for AICSem storage-mode IDs.

Confirm DKMS status:

TERM Terminal command

```
dkms status | grep -i aic
```

Expected:

TERM Expected terminal output

```
aic8800/1.0.0, 5.10.180-olimax, armv7l: installed
```

8. Load and Verify the Driver

Load the firmware/driver module:

TERM Terminal command

```
modprobe aic_load_fw
```

If the adapter was already plugged in, unplug and replug it, or reboot.

Check USB state:

TERM Terminal command

```
lsusb
```

Expected working state:

TERM Expected terminal output

```
a69c:8d81 AICSemI AIC 8800D80
```

Check interfaces:

TERM Terminal command

```
ip link
```

Expected: a new Wi-Fi interface named like:

TERM Expected terminal output

```
wlx001d43100a9d
```

The exact interface name depends on the adapter MAC address.

Scan for nearby Wi-Fi networks:

TERM Terminal command

```
nmcli device wifi list
```

Expected: nearby 2.4 GHz and 5 GHz SSIDs are listed.

9. Connect with nmtui

```

TERM  Generated nmtui-style screen

+-- NetworkManager TUI -----+
| Activate a connection          |
|                                |
|   Wired                        |
|   Wi-Fi                       |
|   > OlimexTest5G               |   Signal: ****
|   OlimexAP                     |   Signal: ***
|   OlimexGuest-5GHz             |   Signal: **
|                                |
|                                |
|                                |   <Activate>   <Back>
+-----+

```

The easiest interactive method is:

```

TERM  Terminal command

nmtui

```

In the text UI:

1. Select 'Activate a connection'.
2. Select the Wi-Fi network SSID.
3. Enter the Wi-Fi password.
4. Select 'OK' or 'Activate'.
5. Exit 'nmtui'.

Verify Internet access:

```

TERM  Terminal command

ping -c 4 8.8.8.8
ping -c 4 www.google.bg

```

Successful test result:

```

TERM  Expected terminal output

packets transmitted, packets received, 0% packet loss

```

Troubleshooting

If `/lib/modules/$(uname -r)/build` is missing or the DKMS build fails, first confirm the headers were installed:

TERM Terminal command

```
apt install -y linux-headers-olimax linux-headers-$(uname -r)
```

Then re-run the DKMS build-tree check from step 4.

If the build fails with:

TERM Expected terminal output

```
No rule to make target 'scripts/module.lds'
```

then the installed OLIMEX header package is too old. Run `'apt update'`, `'apt dist-upgrade'`, reboot, then install the headers again.

If logs show:

TERM Expected terminal output

```
chip_id=7, chip_mcu_id = 1
cmd timed-out
cmd queue crashed
bin upload fail: 170400
```

then the wrong AIC firmware/driver branch is probably being used. Use the `'legacy-mcu1'` branch.

If the adapter stays as storage mode:

TERM Expected terminal output

```
a69c:5721 aicsemi Aic MSC
```

eject it manually:

TERM Terminal command

```
lsblk -o NAME,MODEL,SIZE,RO,TYPE
eject /dev/sda
```

Use the correct `'/dev/sdX'` from `'lsblk'`.

If this is a retry after a failed AICSemi install, clean the DKMS state:

TERM Terminal command

```
dkms remove -m aic8800 -v 1.0.0 --all
```

If a future OLIMAGE kernel update changes 'uname -r', reboot into the new kernel, install matching headers with 'apt install -y linux-headers-olinux linux-headers-\$(uname -r)', then rebuild the DKMS driver.