

# Nvidia Jetson AGX Orin + CYW54591-PCIE + FMAC Linux Driver (Fafnir ES100.1)

---

1	AGX Orin Preparation .....	3
2	Ubuntu PC Preparation.....	4
3	Download Jetson Linux Packages on Ubuntu PC.....	5
4	Install Toolchain.....	6
5	Install Binary and Source of BSP, and Sample Root File System .....	7
6	Build and Update Kernel.....	8
7	Deploy image on the board .....	9
8	Download the latest FMAC release & Build FMAC driver .....	10
9	Build wpa_supplicant/hostapd.....	11
10	Connect WLAN module to AGX Orin .....	12
11	Load Driver .....	13

Please refer the following NVIDIA web sites as well.

Jetson AGX Orin Technical Brief

<https://resources.nvidia.com/en-us-jetson-agx-orin-pathfactory-content>

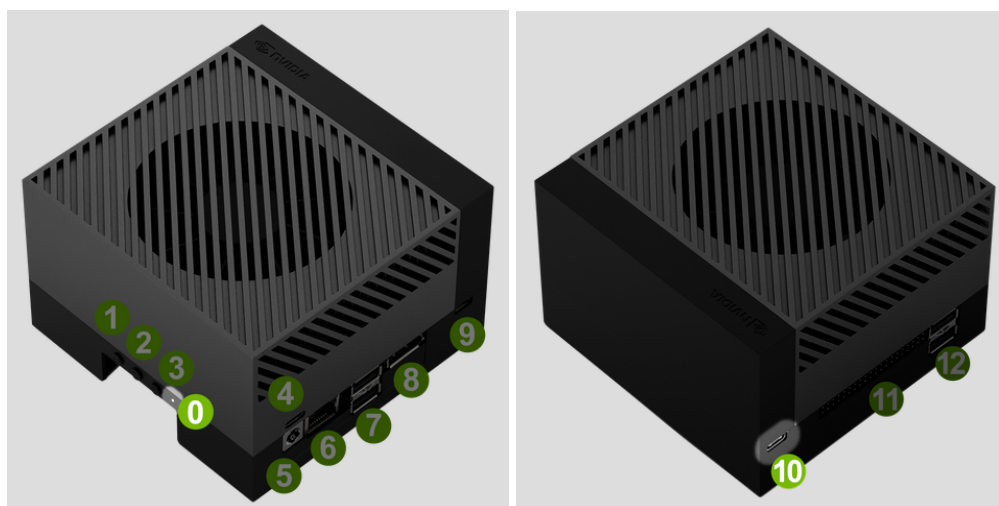
NVIDIA Jetson AGX Orin Developer Kit User Guide

<https://developer.nvidia.com/embedded/learn/jetson-agx-orin-devkit-user-guide/index.html>

## 1 AGX Orin Preparation

You need to do the initialization of Jetson AGX Orin at first. Recovery mode does not work in default. After initialization is done, you can enable recovery mode.

- Connect USB Type-C port next to the 40-pin connector to PC's USB Type-A connector. This is for serial terminal.
- Connect power supply to USB Type-C port next to the DC plug. Jetson AGX Orin will boot automatically in around 20 sec.
- Open serial terminal (e.g. minicom, putty) on /dev/ttyACM0.  
PC \$> sudo minicom -D /dev/ttyACM0
- Follow the instruction and finish initialization. Then power off. At the next boot, you can enable Recovery Mode.



0: LED, 1: Power Button, 2: Recovery Mode Button, 3: Reset Button, 4: USB Type-C for Power Supply, 10: USB Type-C for UART/Virtual Ethernet(Normal Mode) and USB-OTG(Recovery Mode)

## 2 Ubuntu PC Preparation

Ubuntu 20.04 or 18.04.

Install the following packages.

```
PC $> sudo apt install build-essential bc git qemu-user-static bison flex
```

### 3 Download Jetson Linux Packages on Ubuntu PC

Go to Jetson Linux site.

<https://developer.nvidia.com/embedded/jetson-linux>

Then, download the following 4 package. Copy them under <working directory> on Ubuntu PC.

- Driver Package (BSP): Jetson\_Linux\_R35.3.1\_aarch64.tbz2  
[https://developer.nvidia.com/downloads/embedded/l4t/r35\\_release\\_v3.1/release/jetson\\_linux\\_r35.3.1\\_aarch64.tbz2/](https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v3.1/release/jetson_linux_r35.3.1_aarch64.tbz2/)
- Sample Root Filesystem: Tegra\_Linux\_Sample-Root-Filesystem\_R35.3.1\_aarch64.tbz2  
[https://developer.nvidia.com/downloads/embedded/l4t/r35\\_release\\_v3.1/release/tegra\\_linux\\_sample-root-filesystem\\_r35.3.1\\_aarch64.tbz2/](https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v3.1/release/tegra_linux_sample-root-filesystem_r35.3.1_aarch64.tbz2/)
- Driver Package (BSP) Sources: public\_sources.tbz2  
[https://developer.nvidia.com/downloads/embedded/l4t/r35\\_release\\_v3.1/sources/public\\_sources.tbz2/](https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v3.1/sources/public_sources.tbz2/)
- Bootlin Toolchain gcc 9.3: aarch64--glibc--stable-final.tar.gz  
<https://developer.nvidia.com/embedded/jetson-linux/bootlin-toolchain-gcc-93>

## 4 Install Toolchain

```
PC $> sudo mkdir -p /opt/l4t_toolchain
PC $> sudo tar zxvf aarch64--glibc--stable-final.tar.gz -C /opt/l4t_toolchain

PC $> export CROSS_COMPILE_AARCH64_PATH=/opt/l4t_toolchain
PC $> export CROSS_COMPILE_AARCH64=${CROSS_COMPILE_AARCH64_PATH}/bin/aarch64-buildroot-
linux-gnu-
```

## 5 Install Binary and Source of BSP, and Sample Root File System

Define Directories.

```
PC $> L4T_BASE_DIR=$PWD/Linux_for_Tegra
PC $> L4T_ROOTFS_DIR=$L4T_BASE_DIR/rootfs
PC $> L4T_KERNEL_SRC=$L4T_BASE_DIR/source/public/kernel/kernel-5.10
PC $> L4T_KERNEL_OUT=$L4T_BASE_DIR/my_bins
```

Unpack packages under appropriate directories.

```
PC $> tar xf Jetson_Linux_R35.3.1_aarch64.tbz2
PC $> tar xf public_sources.tbz2
PC $> cd ${L4T_ROOTFS_DIR}
PC $> sudo tar xpf ../../Tegra_Linux_Sample-Root-Filesystem_R35.3.1_aarch64.tbz2
PC $> cd ${L4T_BASE_DIR}
PC $> sudo ./apply_binaries.sh
```

## 6 Build and Update Kernel

Build kernel, dtb, and kernel modules.

```
PC $> cd source/public/
PC $> tar xjf kernel_src.tbz2
PC $> mkdir -p ${L4T_KERNEL_OUT}
PC $> ./nvbuild.sh -o ${L4T_KERNEL_OUT}
```

Update kernel, dtb, and kernel modules.

```
PC $> sudo cp ${L4T_KERNEL_OUT}/arch/arm64/boot/Image ${L4T_BASE_DIR}/kernel
PC $> sudo cp ${L4T_KERNEL_OUT}/arch/arm64/boot/dts/* ${L4T_BASE_DIR}/kernel/dtb/
PC $> sudo cp ${L4T_KERNEL_OUT}/arch/arm64/boot/dts/* ${L4T_ROOTFS_DIR}/boot/
PC $> sudo make -C ${L4T_KERNEL_SRC} O=${L4T_KERNEL_OUT} INSTALL_MOD_PATH=${L4T_ROOTFS_DIR}
modules_install
```



## 7 Deploy image on the board

Set AGX Orin to Recovery Mode

- Connect USB Type-C port next to 40-pin to Ubuntu PC.
- Connect Power Supply to USB Type-C port next to DC plug.
- Push recovery mode button (Second button on left side from LED) and hold it.
- Push power button (Third button on left side from LED) and release it.
- Release recovery button.

Check if Recovery mode device exists among USB device list.

```
PC $> lsusb
```

The following device should exist. ID=7023 (not 7020)

```
Bus 001 Device 032: ID 0955:7023 NVidia Corp.
```

Flash the image via USB.

```
PC $> sudo ./flash.sh jetson-agx-orin-devkit mmcblk0p1
```

After flashing is done, AGX Orin reboot automatically.

Open serial terminal.

```
PC $> sudo minicom -D /dev/ttyACM0
```

You need to go through initialization procedure again.

Delete or rename fmac driver, cypress firmware, and wpa\_supplicant which are installed in default.

```
Board $> sudo rm /lib/modules/5.10.104-tegra/kernel/net/wireless/cfg80211.ko
Board $> sudo rm /lib/modules/5.10.104-tegra/kernel/drivers/net/wireless/broadcom/
brcm80211/brcmutil/brcmutil.ko
Board $> sudo rm /lib/modules/5.10.104-tegra/kernel/drivers/net/wireless/broadcom/
brcm80211/brcmfmac/brcmfmac.ko
Board $> sudo rm /lib/firmware/cypress/*
Board $> sudo rm /usr/sbin/hostapd
Board $> sudo rm /usr/sbin/hostapd_cli
```

## 8 Download the latest FMAC release & Build FMAC driver

Go to visit Infineon Community - <https://community.infineon.com/t5/Wi-Fi-Bluetooth-for-Linux/Cypress-Linux-WiFi-Driver-Release-FMAC-2023-08-01/td-p/459849>

Open another terminal.

```
PC $> tar xvf cypress-backports-v5.15.58-2023_0222-module-src.tar.gz
PC $> cd v5.15.58-backports
PC $> vi net/wireless/Kconfig
(change "default" of CFG80211_REQUIRE_SIGNED_REGDB and CFG80211_USE_KERNEL_REGDB_KEYS from
"y" to "n".

PC $> export MY_KERNEL=<working directory>/Linux_for_Tegra/my_bins # the same with
$L4T_KERNEL_OUT
PC $> make KLIB=$MY_KERNEL KLIB_BUILD=$MY_KERNEL defconfig-brcmfmac

PC $> export ARCH=arm64
PC $> export CROSS_COMPILE=/opt/l4t_toolchain/bin/aarch64-buildroot-linux-gnu-
PC $> make KLIB=$MY_KERNEL KLIB_BUILD=$MY_KERNEL modules -j`nproc`
```

- Copy \*.ko which are built above to the board.
  - ./compat/compat.ko
  - ./net/wireless/cfg80211.ko
  - ./drivers/net/wireless/broadcom/brcm80211/brcmutil/brcmutil.ko
  - ./drivers/net/wireless/broadcom/brcm80211/brcmfmac/brcmfmac.ko
- Unpack cypress-firmware-v5.15.58-2023\_0222.tar.gz and copy firmware/\* to /lib/firmware/cypress.
- Copy appropriate nvram files to /lib/firmware/cypress.

## 9 Build wpa\_supplicant/hostapd

You need to build it on AGX Orin since libraries are not available in cross-compile environment.

Log in the board. Connect it to the internet via Ethernet. Then, Install libraries for wpa\_supplicant/hostapd as follows.

```
Board $> sudo apt install libnl-3-dev libnl-genl-3-dev libnl-route-3-dev libnl-nf-3-dev  
libssl-dev libdbus-1-dev
```

Copy cypress-hostap\_2\_10-2023\_0222.tar.gz to the board.

Build wpa\_supplicant/hostapd as follows.

```
Board $> git clone git://w1.fi/hostap.git  
Board $> cd hostap/  
Board $> git checkout cff80b4f7d3c  
  
Board $> tar xvf cypress-hostap_2_10-2023_0222.tar.gz  
Board $> for i in cypress-hostap_2_10/*.patch; do patch -p1 < $i; done  
  
Board $> cd hostapd/  
Board $> cp defconfig_base .config  
Board $> make clean  
Board $> make -j`nproc`  
  
Board $> cd ../wpa_supplicant/  
Board $> cp defconfig_base .config  
Board $> make clean  
Board $> make -j`nproc`
```

Copy binaries to /usr/sbin.

```
Board $> sudo cp hostapd/hostapd /usr/sbin  
Board $> sudo cp hostapd/hostapd_cli /usr/sbin  
Board $> sudo cp hostapd/wpa_supplicant /usr/sbin  
Board $> sudo cp hostapd/wpa_cli /usr/sbin
```

## 10 Connect WLAN module to AGX Orin

Connect CYW54591-PCIE module on M.2 slot as below picture.



## 11 Load Driver

Check if device exist.

```
Board $> lspci
(...)
0001:01:00.0 Network controller: Broadcom Inc. and subsidiaries Device 4417 (rev 0d)
```

Load fmac driver.

```
Board $> sudo insmod ./compat.ko
Board $> sudo insmod ./cfg80211.ko
Board $> sudo insmod ./brcmutil.ko
Board $> sudo insmod ./brcmfmac.ko
```

Log is as follows.

```
[ 552.801671] compat: loading out-of-tree module taints kernel.
[ 552.807754] compat: module verification failed: signature and/or required key missing -
tainting kernel
[ 552.818857] Loading modules backported from Linux version v5.15.58-2023_0222-0-
gd4d1a61163b5
[ 552.818860] Backport generated by backports.git v5.15.58-1-0-g42a95ce7
[ 552.990016] usbcore: registered new interface driver brcmfmac
[ 552.990813] brcmfmac 0001:01:00.0: Adding to iommu group 9
[ 552.991490] brcmfmac 0001:01:00.0: enabling device (0000 -> 0002)
[ 553.102994] brcmfmac: brcmf_fw_alloc_request: using cypress/cyfm54591-pcie for chip
BCM4355/13
[ 553.125690] brcmfmac 0001:01:00.0: Direct firmware load for cypress/cyfm54591-
pcie.nvidia,p3737-0000+p3701-0000.bin failed 2
[ 553.138518] brcmfmac 0001:01:00.0: Falling back to sysfs fallback for: cypress/
cyfm54591-pcie.nvidia,p3737-0000+p3701-0000.n
[ 553.162006] brcmfmac: brcmf_fw_request_firmware: no board-specific nvram available
(ret=-2), device will use cypress/cyfm54t
[ 553.759557] brcmfmac: brcmf_fw_alloc_request: using cypress/cyfm54591-pcie for chip
BCM4355/13
[ 553.789000] brcmfmac: brcmf_c_preinit_dcnds: Firmware: BCM4355/13 wl0: Jan 3 2023 19:52:2
6 version 13.35.294 (38f3803 CY) FW4
```

#### Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

**Edition yyyy-mm-dd**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2023 Infineon Technologies AG.**

**All Rights Reserved.**

**Do you have a question about this document?**

**Email:** [erratum@infineon.com](mailto:erratum@infineon.com)

**Document reference**

**AppNote Number**

#### IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office ([www.infineon.com](http://www.infineon.com)).

#### WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.