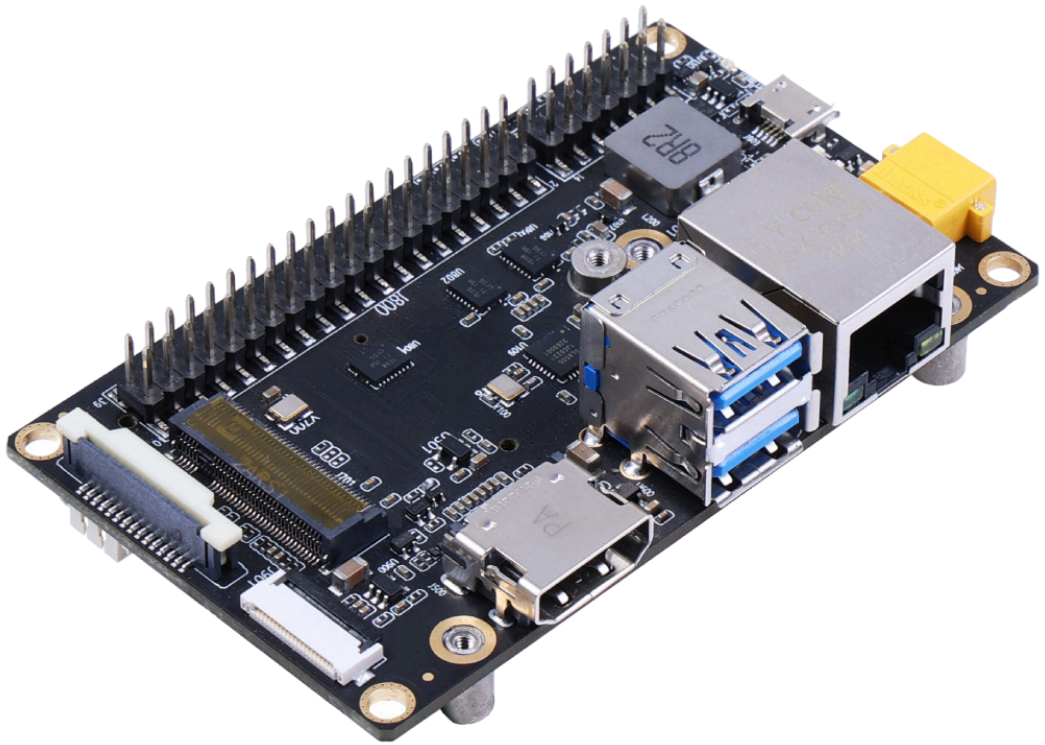
Flash JetPack OS to A603 Carrier Board



Get One Now

A603 Carrier Board is a powerful extension board that supports Jetson Orin™ NX/Nano modules. It features 1 GbE port, M.2 Key M for SSD, M.2 Key E for WiFi/BlueTooth, CSI, and HDMI for high-quality video capture and display. It also contains 4x USB ports, fan, RTC, flexible 9-20V power supply. By the compact design, it can be flexible and easy to integrate into a variety of edge computing applications. In this wiki, we will show you how to flash Jetpack to an NVMe SSD and a USB Flash drive connected to the A603 Carrier Board.

Supported Module

- NVIDIA® Jetson Orin™ Nano Module 4GB
- NVIDIA® Jetson Orin™ Nano Module 8GB
- NVIDIA® Jetson Orin™ NX Module 8GB
- NVIDIA® Jetson Orin™ NX Module 16GB

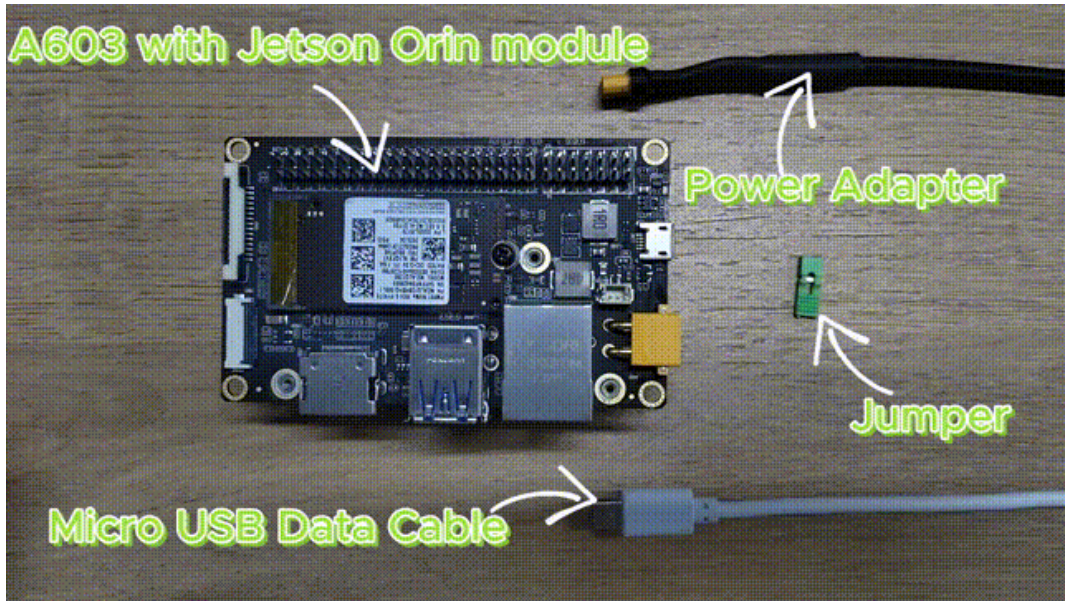
Prerequisites

- Ubuntu Host PC
- A603 Carrier Board with Jetson Orin module
- Micro-USB data transmission cable

Enter Force Recovery Mode

NOTE

Before we can move on to the installation steps, we need to make sure that the board is in force recovery mode.



step-by-step detailed tutorial

Download the peripheral drivers

First of all, you need to install the peripheral drivers for this board. These are needed for some hardware peripherals to function on the board. Click the below links to download the drivers according to the Jetson module

Jetson Module	JetPack Version	L4T Version	Download Link
Jetson Orin NX 8GB/ 16GB	5.1	35.2.1	Download
	5.1.1	35.3.1	Download
Jetson Orin Nano 4GB/ 8GB	5.1.1	35.3.1	Download
Jetson Orin NX 8GB/ 16GB, Jetson Orin Nano 4GB/ 8GB	5.1.2	35.4.1	Download
Jetson Orin NX 8GB/ 16GB, Jetson Orin Nano 4GB/ 8GB	5.1.4	35.6.0	Download
Jetson Orin NX 8GB/ 16GB, Jetson Orin Nano 4GB/ 8GB	6.0	36.3	Download
Jetson Orin NX 8GB/ 16GB,	6.1	36.4	Download

Jetson Module	JetPack Version	L4T Version	Download Link
Jetson Orin Nano 4GB/ 8GB			
Jetson Orin NX 8GB/ 16GB, Jetson Orin Nano 4GB/ 8GB	6.2	36.4.3	Download

> **INFO**
>
> To verify the integrity of the downloaded firmware, you can compare the SHA256 hash value.
>
> On an Ubuntu host machine, open the terminal and run the command `sha256sum <File>` to obtain the SHA256 hash value of the downloaded file. If the resulting hash matches the SHA256 hash provided here, it confirms that the firmware you downloaded is complete and intact.

Note: Currently we provide the above drivers. We will keep updating the drivers in the future with the release of new JetPack versions.

Flash to Jetson

Here is a video for flashing JetPack 6.1 onto the A603 carrier board + Orin Nx 16GB module. You can refer to the video and the detailed steps below to flash your device.

NOTE

Before moving onto flashing, it should be noted that Jetson Orin NX module only supports JetPack 5.1 and above, while Jetson Orin Nano module only supports JetPack 5.1.1 and above.

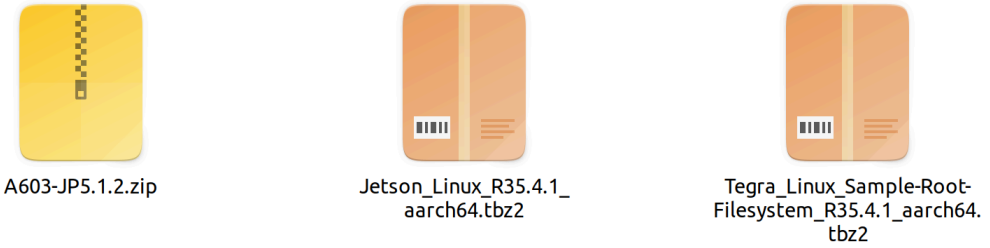
JP5.1.1 for Jetson Orin NX | JP5.1.1 for Jetson Orin Nano | **JP5.1.2** | JP5.1.4 | JP6.0 | JP6.1 | JP6

Here we will install **Jetpack 5.1.2** on the A603 Carrier Board with Jetson Orin module.

Step 1: Download the NVIDIA drivers on the host PC. The required drivers are shown below:

	Jetson Orin modules and developer kit	Jetson Xavier modules and developer kits
DRIVERS	Driver Package (BSP)	
	Sample Root Filesystem	
SOURCES	Driver Package (BSP) Sources	
	Sample Root Filesystem Sources	
	Sensor Processing Engine Sources	

Step 2: Move the downloaded peripheral drivers from before into the same folder with NVIDIA drivers. Now you will see three compressed files in the same folder.



Step 3: Extract **Jetson_Linux_R35.4.1_aarch64.tbz2** and **Tegra_Linux_Sample-Root-Filesystem_R35.4.1_aarch64.tbz2** by navigating to the folder containing these files, apply the changes and install the necessary prerequisites.

```
tar xf Jetson_Linux_R35.4.1_aarch64.tbz2
sudo tar xpf Tegra_Linux_Sample-Root-Filesystem_R35.4.1_aarch64.tbz2 -C Linux_for_Tegra/rootfs/
cd Linux_for_Tegra/
sudo ./apply_binaries.sh
sudo ./tools/l4t_flash_prerequisites.sh
```

Step 4: Extract **A603-JP5.1.2.zip**. Here we additionally install the **unzip** package which is needed to decompress the .zip file.

```
cd ..
sudo apt install unzip
unzip A603-JP5.1.2.zip
```

Step 5: Configure your username, password & hostname so that you do not need to enter the Ubuntu installation wizard after the device finishes booting.

```
sudo tools/l4t_create_default_user.sh -u {USERNAME} -p
{PASSWORD} -a -n {HOSTNAME} --accept-license
```

For example (username:"nvidia", password:"nvidia", device-name:"nvidia-desktop"):

```
sudo tools/l4t_create_default_user.sh -u nvidia -p
nvidia -a -n nvidia-desktop --accept-license
```

Step 6: Flash the system to NVMe SSD.

```
cd Linux_for_Tegra
sudo ./tools/kernel_flash/l4t_initrd_flash.sh --
external-device nvme0n1p1 -c
tools/kernel_flash/flash_l4t_external.xml -p "-c
bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs
--network usb0 jetson-orin-nano-devkit internal
```

You will see the following output if the flashing process is successful.

```
Writing /mnt/internal/qspi_bootblob_ver.txt (109 bytes) into  /dev/mtd0:66977792
Copied 109 bytes from /mnt/internal/qspi_bootblob_ver.txt to address 0x03fe0000 in flash
Writing gpt_secondary_3_0.bin (parittion: secondary_gpt) into /dev/mtd0
Sha1 checksum matched for /mnt/internal/gpt_secondary_3_0.bin
Writing /mnt/internal/gpt_secondary_3_0.bin (16896 bytes) into  /dev/mtd0:67091968
Copied 16896 bytes from /mnt/internal/gpt_secondary_3_0.bin to address 0x03ffbe00 in flash
[ 221]: l4t_flash_from_kernel: Successfully flash the qspi
[ 221]: l4t_flash_from_kernel: Flashing success
[ 221]: l4t_flash_from_kernel: The device size indicated in the partition layout xml is smaller than the actual size. This ut
ility will try to fix the GPT.
Flash is successful
Reboot device
Cleaning up...
Log is saved to Linux_for_Tegra/initrdlog/flash_1-3_0_20240522-135433.log
```

CAN Interfaces

Since there is a CAN transceiver on A603 carrier board, you don't extra transceiver like dev kit.

Step1. Install `devmem2` to write values to registers:

```
sudo apt-get install devmem2
```

Step2. Write values according to here.

```
sudo devmem2 0x0c303010 w 0xc400
sudo devmem2 0x0c303018 w 0xc458
```

```
seeed@seeed-desktop:~$ sudo devmem2 0x0c303010 w 0xc400
/dev/mem opened.
Memory mapped at address 0xffff992f4000.
Value at address 0xC303010 (0xffff992f4010): 0xC400
Written 0xC400; readback 0xC400
seeed@seeed-desktop:~$
```

```
seeed@seeed-desktop:~$ sudo devmem2 0x0c303018 w 0xc458
/dev/mem opened.
Memory mapped at address 0xffff896c6000.
Value at address 0xC303018 (0xffff896c6018): 0xC458
Written 0xC458; readback 0xC458
seeed@seeed-desktop:~$
```

Step3. Load Kernel modules:

```
sudo modprobe can
sudo modprobe can_raw
sudo modprobe mttcan
```

After loading these modules, you should be able to see these logs in `sudo dmesg`:

```
[  103.979487] can: controller area network core
[  103.979627] NET: Registered PF_CAN protocol family
[  106.097784] can: raw protocol
```

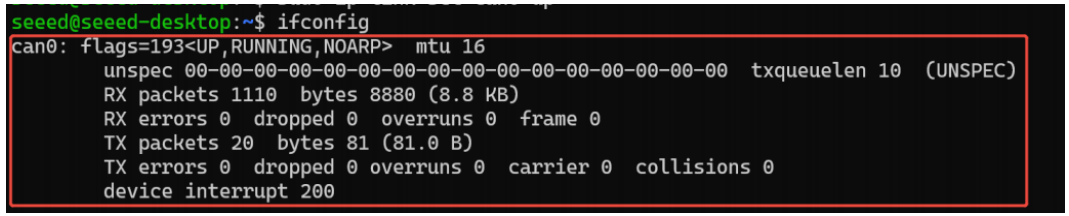
Step4. Bring up can0 interface:

```
sudo ip link set can0 type can bitrate 500000
```

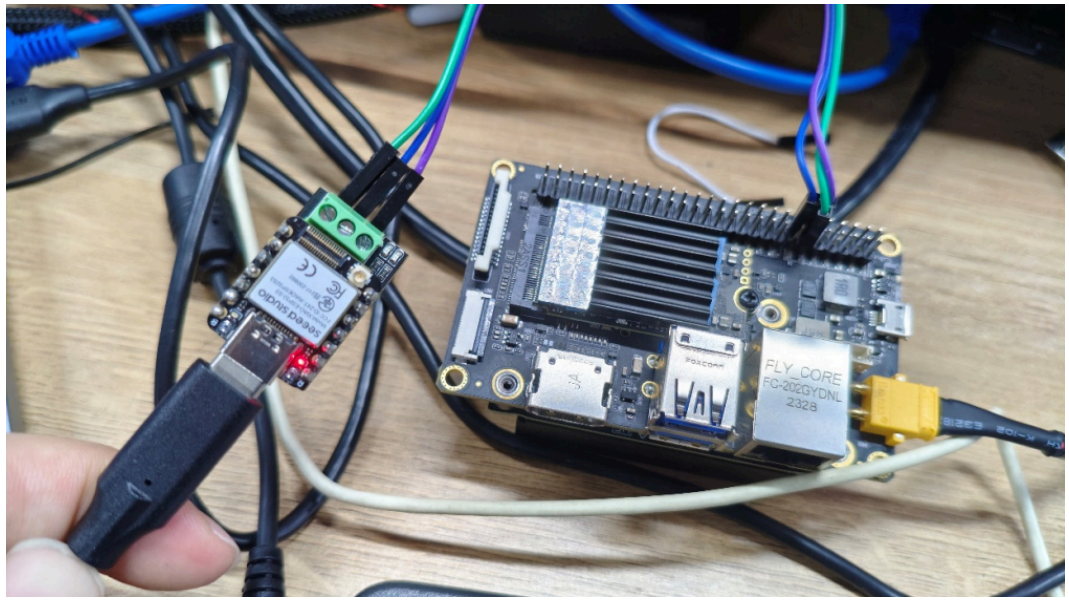
Optionally, you can change the bitrate to 1000000. Then, bring up can0:

```
sudo ip link set can0 up
```

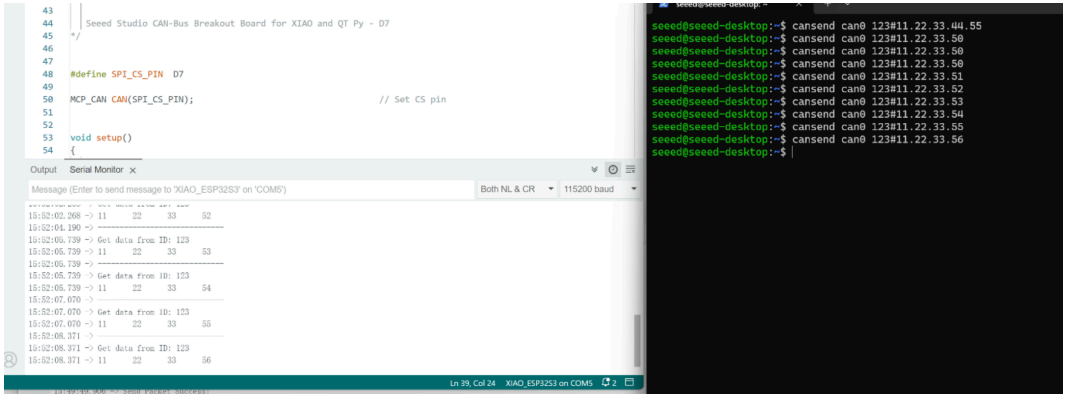
Check the interface with `ifconfig`:



Step5. Sending data (require can-utils installed). On the other side, we used a MCU with CAN Expansion board to receive data.

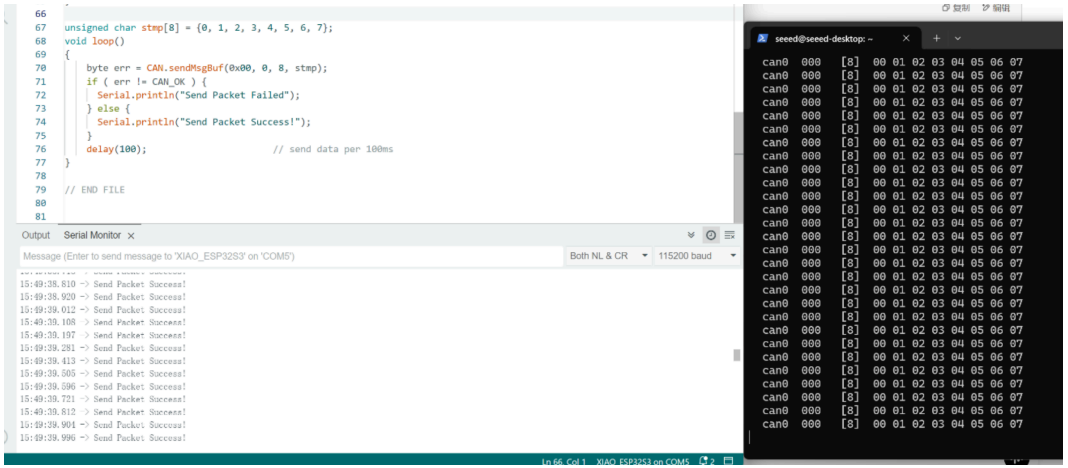


Run `cansend can0 123#11.22.33.50` on jetson terminal:



Step6. Receiving data. On the other side, we used a MCU with CAN Expansion board to send data.

Run `candump can0` on jetson terminal:



Tech Support & Product Discussion

Thank you for choosing our products! We are here to provide you with different support to ensure that your experience with our products is as smooth as possible. We offer several communication channels to cater to different preferences and needs.

Edit this page

*Last updated on **Apr 19, 2023** by **Lakshantha***

0 reactions

16 comments · 28 replies *– powered by giscus*

Oldest Newest