Jetson-AGX Orin Setup – (Yocto OTA)

Used Version

JetPack version: 5.1.3
Host Ubuntu version: 22.04
L4T (Linux for Tegra) Branch: Kirkstone

https://github.com/OE4T/meta-tegra/wiki/Which-branch-should-I-use%3F

1. Install JetPack in Orin

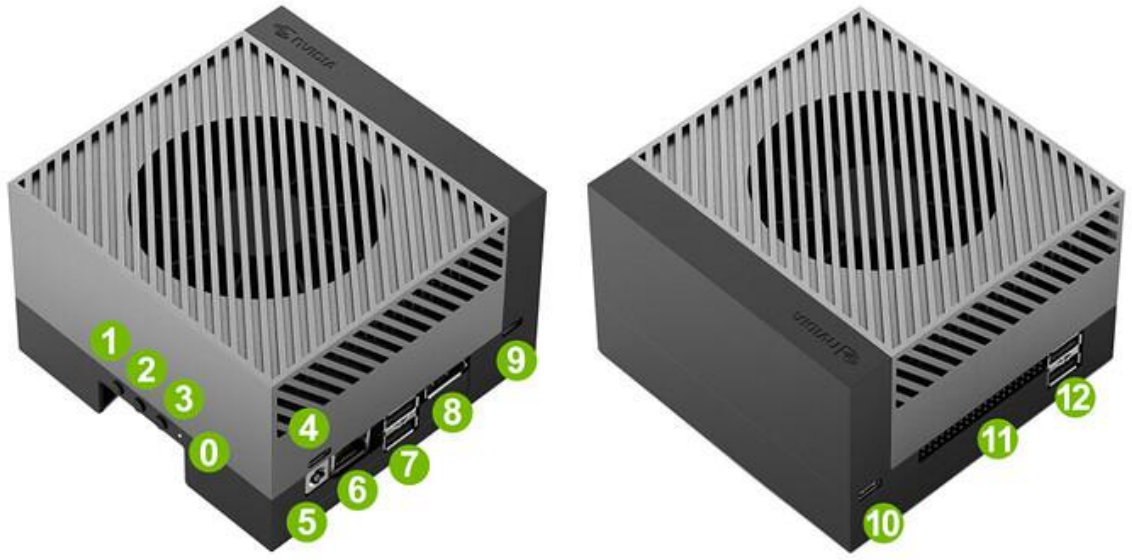
First step is install JetPack in Orin.

1. JetPack is ‘NVIDIA JetPack SDK’, that provides everything needed to flash, boot, and develop on Jetson devices.

2. L4T (Linux for Tegra) is Linux-based operating system that runs on the Jetson hardware. JetPack installs L4T onto your Jetson board. L4T is Actual Jetpack’s OS, so we have to match the L4T version – Yocto branch version.

1. Install JetPack in Orin

Install 'Nvidia sdk manager' in Host Machine. (maybe laptop)
https://docs.nvidia.com/sdk-manager/download-run-sdkm/index.html
(You have to make the nvidia ID)



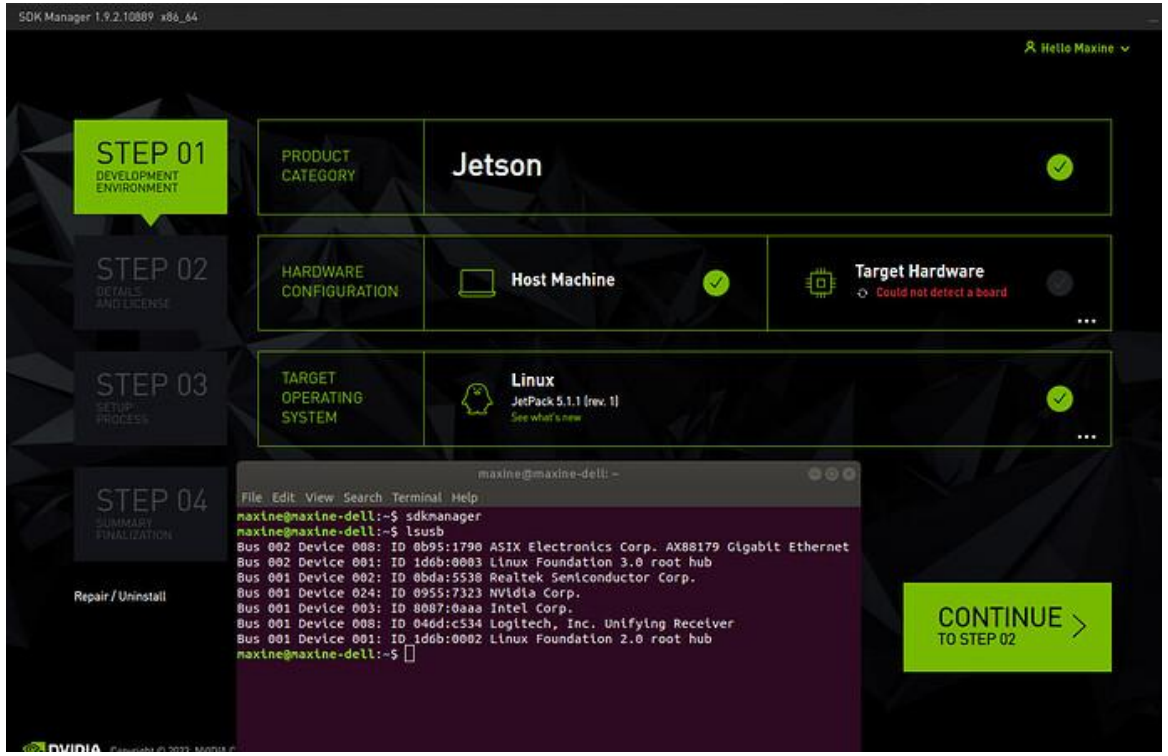
If you installed sdkmanager in Host,

1. Power off the Orin.
2. Connect USB-C in Port number 10, to Host
3. Push the Button 2, and maintain until step 6.
4. Connect Power of Orin, in Port number 4.
5. Then, LED will turn on.
6. After 2~3 sec, take off the button 2.
7. In Host, Use the Terminal command

Info) if you want to test whether Connecting with host is completely done or not, use the command $lsusb. Refer the next page.

1. Install JetPack in Orin

If you complete install, Use the command $sdkmanager in Host Ubuntu terminal.
Then the SDK kit will appear.



1. Select the Target Hardware (AGX Orin devkit 32GB)

2. Select JetPack 5.1.3, in NVMe partition.

3. Install and Flashing.

   in Step03, You can add or substract the options
   But, in this project, almost options are needed.

4. If all process done, disconnect powef of Orin, USB. and Boot Orin, connect HDMI to mornitor and keyboard. You can see the JetPack booting.

2. Setting Up Yocto for NVIDIA Jetson Platform

Second is Setting Up the Yocto Build System in Host.

1. Follow command in Your Host.
https://developer.ridgerun.com/wiki/index.php/Yocto_Support_for_NVIDIA_Jetson_Platforms_-_Setting_up_Yocto
From Install Yocto Dependencies, to Compile Yocto Image Step 2.

2. Build image you want to use, using this command

Bitbake core-image-x11
Bitbake core-image-base
Bitbake core-image-minimal .....

If you start Build Image along this,
maybe the image will build, but that image will not have any package.

[info] branch list is updating continue and continue. So, if the branch changed, you can find information in this. **https://github.com/OE4T/meta-tegra/wiki/Which-branch-should-I-use%3F**

2. Setting Up Yocto for NVIDIA Jetson Platform

If Building image has completed, the results may be in
$YOCTO_DIR/build/tmp/deploy/images/${machine}

Follow these commands to flash.

1. **Cd $YOCTO_DIR**
2. **Sudo touch deploy.sh and copy that content. (or, copy file to $YOCTO_DIR),**

```
#!/bin/bash

image=$1
machine=$2

scriptdir="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null && pwd )"
deployfile=${image}-${machine}.tegraflash.tar.gz
tmpdir=`mktemp`

rm -rf $tmpdir
mkdir -p $tmpdir
echo "Using temp directory $tmpdir"
pushd $tmpdir
cp $scriptdir/build/tmp/deploy/images/${machine}/$deployfile .
tar -xvf $deployfile
set -e
sudo ./doflash.sh
popd
echo "Removing temp directory $tmpdir"
rm -rf $tmpdir
```

2. Setting Up Yocto for NVIDIA Jetson Platform

3. Connect [Host laptop – Orin machine] by using USB-C.
4. Turn on Orin for Recovery Mode.
5. Use command $ bash deploy.sh [image-name] [machine-name]
   for example) bash deploy.sh core-image-base jetson-agx-orin-devkit [spend about 10 min.]
   [info] machine-name must be same with name in the [local.conf] file

6. Disconnect the Power and USB.

7. Open the result folder ($YOCTO_DIR/build/tmp/deploy/images/${machine}) [in host]

8. Copy 'Image' file (file name is 'Image'.) to USB or your E-mail… we have to copy this file to Orin.

9. Turn Orin. (not recovery mode. Just connect power cable)

10. Use command sudo cp Image /boot/Image_Yocto [in Orin]
11. Use command sudo mv /boot/Image /boot/Image_JetPack [in Orin]

2. Setting Up Yocto for NVIDIA Jetson Platform

12. We have to edit /boot/extlinux/extlinux.conf

```
TIMEOUT
DEFAULT jetpack

MENU TITLE L4T boot options

LABEL jetpack
    MENU LABEL JetPack NVMe
    LINUX /boot/Image_JetPack
    FDT /boot/dtb/kernel_tegra234-p3701-0005-p3737-0000.dtb
    INITRD /boot/initrd
    APPEND ${cbootargs} root=/dev/nvme0n1p1 rw rootwait rootfstype=ext4 mminit_loglevel=4 console=ttyTCU0,115
```

This is almost default [extlinux.conf] (not edited)

12-1. Copy content from LABEL jetpack.. To end of file.
12-2. Paste under the jetpack LABEL.

2. Setting Up Yocto for NVIDIA Jetson Platform

```
TIMEOUT
DEFAULT jetpack

MENU TITLE L4T boot options

LABEL jetpack
    MENU LABEL JetPack NVMe
    LINUX /boot/Image_JetPack
    FDT /boot/dtb/kernel_tegra234-p3701-0005-p3737-0000.dtb
    INITRD /boot/initrd
    APPEND ${cbootargs} root=/dev/nvme0n1p1 rw rootwait rootfstype=ext4 mminit_loglevel=4 console=ttyTCU0,115



LABEL yocto
    MENU LABEL Yocto rootfs
    LINUX /boot/Image_Yocto
    FDT /boot/dtb/kernel_tegra234-p3701-0005-p3737-0000.dtb
    INITRD /boot/initrd
    APPEND ${cbootargs} root=/dev/mmcblk0p1 rw rootwait rootfstype=ext4 mminit_loglevel=4 console=ttyTCU0,115
```

12-3. Edit same with this.

!!! But, [FDT, INITRD, APPEND] can be different compare that image.
just LABEL and Image are important.

!!! LABEL jetpack must be locate above the yocto.

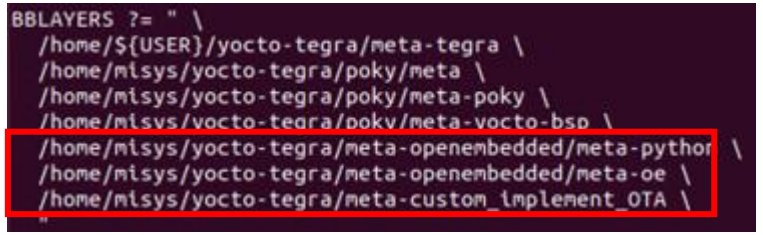
3. Add Custom layer in result of Step 2.

Third is Adding Custom Layer. This Process have to be executed in Host.

1. Copy Meta-custom_implement_OTA folder to $YOCTO_DIR .

2. Open *$YOCTO_DIR//conf/bblayer.conf*
build
3. Add
meta-openembedded/meta-python
meta-openembedded/meta-oe
meta-custom_implement_OTA

in ***$YOCTO_DIR/build/conf/bblayers.conf***



$YOCTO_DIR/build/conf/bblayers.conf

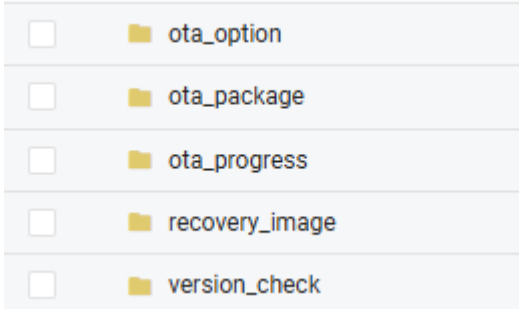
4. Copy local.conf to $YOCTO_DIR/build/conf/local.conf
[info] You can add extra package for your custom image.

5. Follow this command
5-1. cd $YOCTO_DIR
5-2. source /poky/oe-init-build-env build

4. Cloud ID Init

In Project, Used Cloud is Naver Cloud – Object Storage.

1. **Make ID for Ncloud object Storage.**
2. **To Open Console, and enter your storage. Make same directory with image. (if do not, you have to change directory name in OTA layer.)**



3. **You have to change two things in**
   $YOCTO_DIR/meta-custom_implement_OTA/recipes-core/boot-setup/files/s3cfg

   3-1. Check Ncloud – access key and secret key
   3-2. change access_key to yours.
   3-3. change secret_key to yours.

5. Edit OTA layer setting

You have to change something about setting.

In *meta-custom-implement_OTA/recipes-core/files/mjm_ota_start.sh*
(refer file's comment.)

1. **Wifi setting**
2. **SERVER_URL**
3. **Some harded coding URL (now something is fixed with 'mjmota' name.**

6. Setting for JetPack

You have to make some start setting for Nvme(JetPack).

1. **Boot Orin for General mode (not recovery. Just connect power cable)**
   - **1-1. $sudo vi** /etc/gdm3/custom.conf
   - **1-2. find** #AutomaticLoginEnable=false, change to true
   - **1-3. find** #AutomaticLogin=user1 , change to username (recommend misys)

2. **Executing script in booting**
   - **2-1. $sudo touch ota.desktop**
   - **2-2. copt this to ota.desktop**

```
[Desktop Entry]
Type=Application
Exec=gnome-terminal -- bash -c "/home/misys/start_ota.sh; exec bash"
Hidden=false
NoDisplay=false
X-GNOME-Autostart-enabled=true
Name=OTA Script
```

6. Setting for JetPack

2-2 . $sudo mkdir -p ~/.config/autostart
2-3 . $sudo cp ota.desktop ~/.config/autostart/

3. Edit Script (in JetPack)
refer file name ‘start_ota.sh’

Info) In Orin, Wifi have to be needed for checking cloud’s option…etc

7. Build Image and Flashing

Now, Build Yocto Image in Host again.

$ cd $YOCTO_DIR
$ source /poky/oe-init-build-env build
$ bitbake core-image-xxx (your select)

And use the deploy.sh with same way- Step2.

And, if the flashing is done, Boot Orin.
Orin will be booted with NVMe (JetPack Linux).

Use command in Orin.
$ sudo vi ./boot/extlinux/extlinux.conf

Change [DEFAULT jetpack] -> [DEFAULT yocto].
Finally, Use command $ sudo reboot
Yocto will be booted. All of Setting is done.