Steps to flash 22.08 Jetson CUDA-X AI Developer Preview on Jetson AGX Orin Developer Kit

1. **Download Jetson Orin tarball which includes BSP, Rootfs, CUDA, cuDNN, TensorRT**
   A. **Download zip on your ubuntu host system:**
      22.08_jetson_cuda-x_ai_dp.zip
   B. **Unzip the file**
      $ unzip 22.08_jetson_cuda-x_ai_dp.zip

2. **Prepare the BSP build on host .**

```
$  cd BSP_35.1.1
$  tar xpf Jetson_Linux_R35.1.1_aarch64.tbz2
$  cd Linux_for_Tegra/rootfs/
$  sudo tar xpf ../../Tegra_Linux_Sample-Root-Filesystem_R35.1.1_aarch64.tbz2
$  cd ..
$  sudo ./apply_binaries.sh
```

3. **Put the device in recovery mode to flash the device**

   1. Make sure the device is connected to the power adaptor , but powered off
   2. Connect the host computer to the front USB type C connector on the device
   3. Press and hold the middle (Force Recovery) button.
   4. Press and hold the left (Power) button.
   5. Release both buttons.

4. **Verify if the device is in recovery mode (must show in output :** 0955:7223**)**

```
$ lsusb | grep NVidia
  Bus 001 Device 036: ID 0955:7223 NVidia Corp.
```

5. **Now flash the device**

```
# Flash command to flash AGX orin with max N
$ sudo ./flash.sh jetson-agx-orin-devkit-maxn mmcblk0p1
```

6. Now perform the OEM config at first boot

If the HDMI is connected to the device ,

- Follow the instructions on Display and complete the system configuration via GUI.

OR

If the HDMI is not connected to the device ,

- Open /dev/ttyACM4 UART serial console (using minicom) to complete the system configuration via serial.

7. Copy CUDA/CUDNN/TensorRT from 22.08_jetson_cuda-x_ai_dp.zip to Jetson agx orin device .

8. Install CUDA on device

```
$ cd CUDA_11.4
$ sudo dpkg -i cuda-repo-l4t-11-4-local_11.4.14-1_arm64.deb
$ sudo cp /var/cuda-repo-l4t-11-4-local/cuda-82DB0B48-keyring.gpg /usr/share/keyrings/
$ sudo apt-get -y update
$ sudo apt-get -y install cuda-toolkit-11-4
```

9. Install CUDNN on device

```
$ cd cuDNN_8.5
$ sudo dpkg -i cudnn-local-repo-ubuntu2004-8.5.0.87_1.0-1_arm64.deb
$ sudo cp /var/cudnn-local-repo-ubuntu2004-8.5.0.87/cudnn-local-AAE02A33-keyring.gpg /usr/share/keyrings/
$ sudo apt-get update
$ sudo apt-get install -y libcudnn8 libcudnn8-dev libcudnn8-samples
```

10. Install TensorRT on device

```
$ cd TRT_8.5
$ sudo dpkg -i libnvinfer8_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvinfer-dev_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvinfer-plugin8_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvinfer-plugin-dev_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvonnxparsers8_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvonnxparsers-dev_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvparsers8_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvparsers-dev_8.5.0-1+cuda11.4_arm64.deb
```

```
$ sudo dpkg -i libnvinfer-bin_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i libnvinfer-samples_8.5.0-1+cuda11.4_all.deb
$ sudo dpkg -i tensorrt_8.5.0.4-1+cuda11.4_arm64.deb
$ sudo dpkg -i python3-libnvinfer_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i python3-libnvinfer-dev_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i graphsurgeon-tf_8.5.0-1+cuda11.4_arm64.deb
$ sudo dpkg -i uff-converter-tf_8.5.0-1+cuda11.4_arm64.deb
```