

JETSON ORIN NANO

????

📁 Jetson📁 AI📁

- [📁 jtop](#)
- [📁 CUDA](#)
- [📁 opencv with cuda](#)
- [📁 pytorch📁 torchvision](#)
- [📁 ollama](#)
- [📁 Jetson-container](#)
- [📁](#)
- [4G📁](#)
- [📁](#)
- [📁 LOGO](#)

??jtop

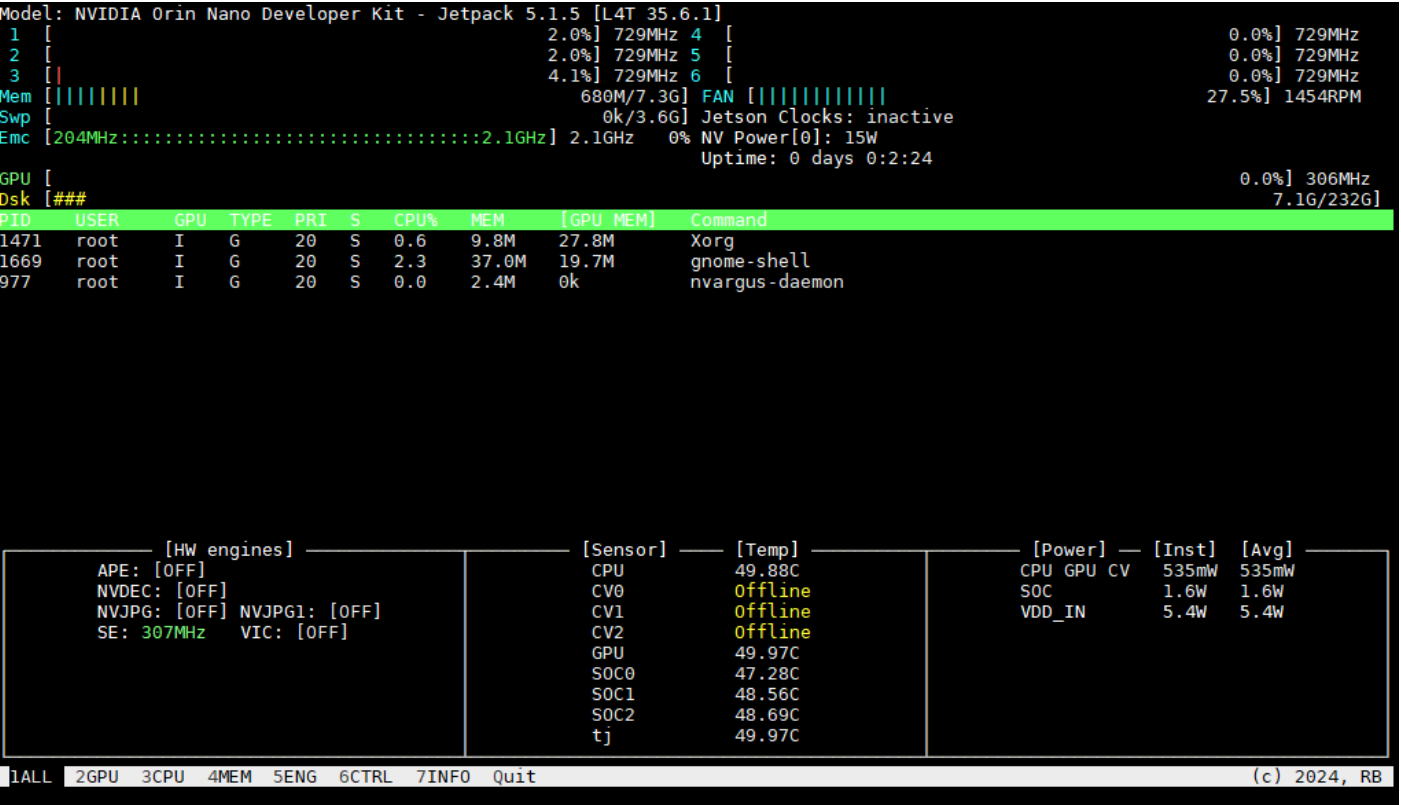
jtop Jetson

1 ??jtop

- jtop
- ```
sudo apt update
sudo apt install python3
sudo apt install python3-pip
```
- jtop
- ```
sudo pip3 install -U pip -i https://pypi.tuna.tsinghua.edu.cn/simple
sudo pip3 install jetson-stats -i https://pypi.tuna.tsinghua.edu.cn/simple
sudo systemctl restart jtop.service
```

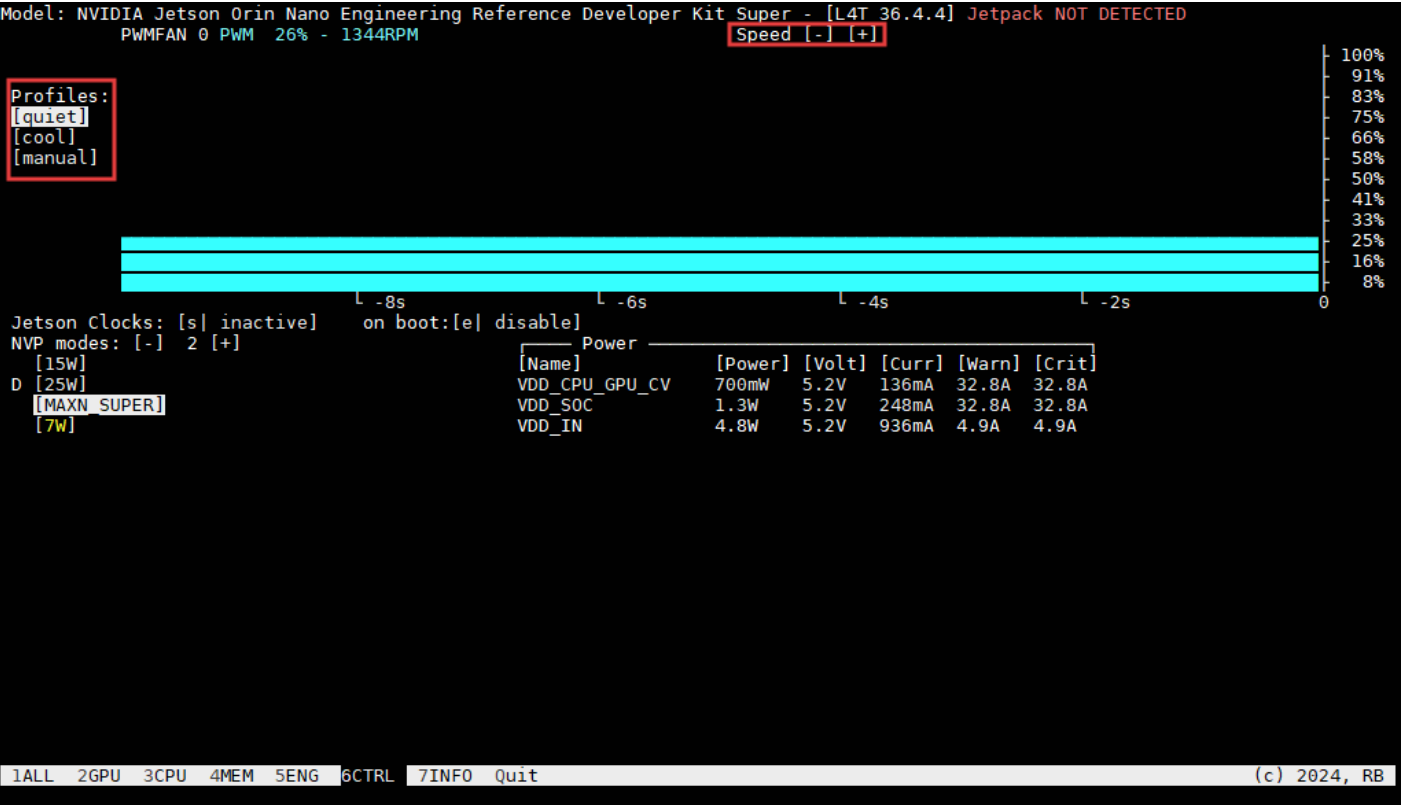
2 jtop???

- jtop

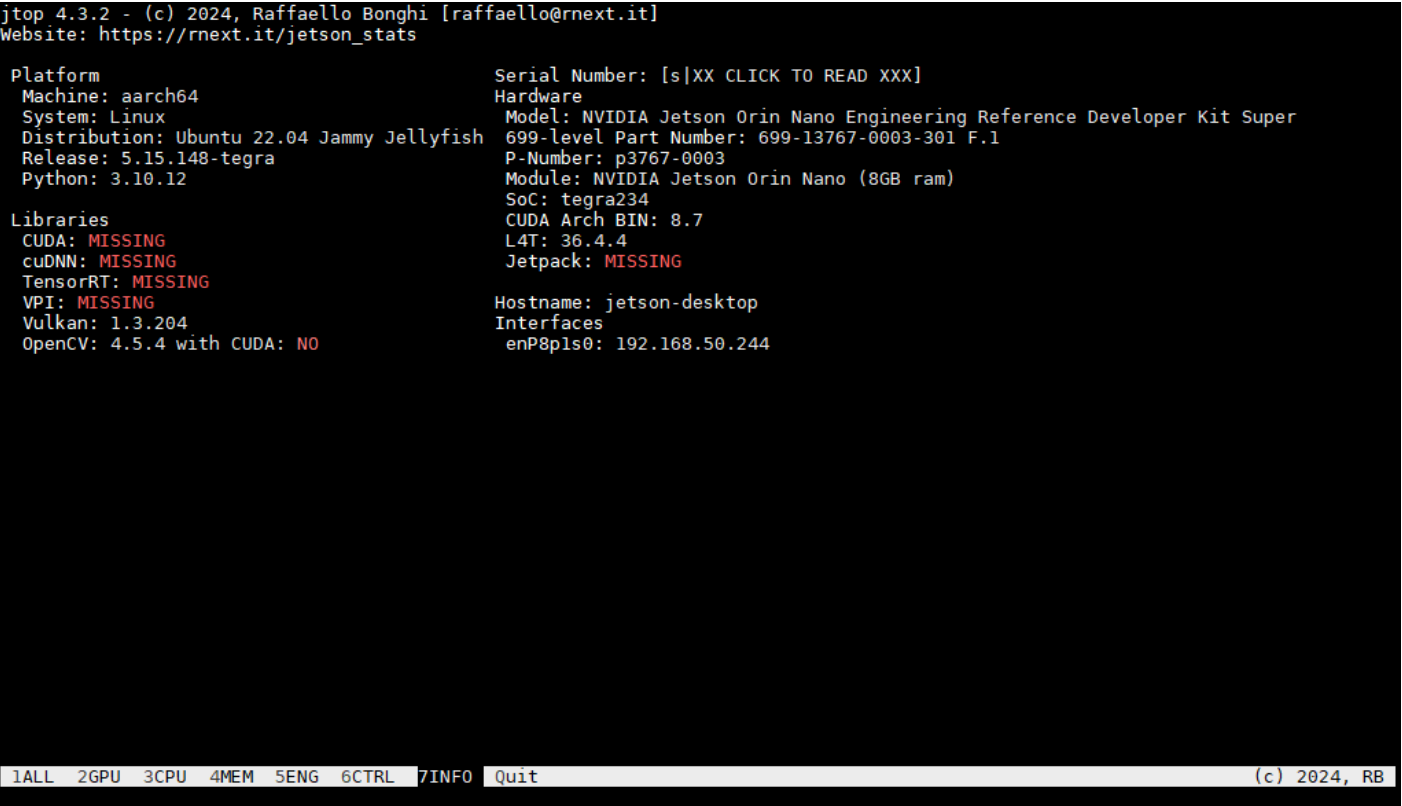


2.1

2.2



2.3



??CUDA

Jetson CUDA NVIDIA AI ARM Jetson GPU

??CUDA JETSON SDK

```
sudo apt update
sudo apt install nvidia-jetpack
```

jtop 4.3.2 - (c) 2024, Raffaello Bonghi [raffaello@rnext.it]
Website: https://rnext.it/jetson_stats

Platform

XXX]chine: aarch64

System: Linux

Distribution: Ubuntu 22.04 Jammy Jellyfish

Release: 5.15.148-tegra

Python: 3.10.12

Libraries

CUDA: 12.6.68

cuDNN: 9.3.0.75

TensorRT: 10.3.0.30

VPI: 3.2.4

Vulkan: 1.3.204

OpenCV: 4.8.0 with CUDA: NO

Serial Number: [s|XX CLICK TO READ Hardware

Model: NVIDIA Jetson Orin Nano Eng

699-level Part Number: 699-13767-0

P-Number: p3767-0003

Module: NVIDIA Jetson Orin Nano (8

SoC: tegra234

CUDA Arch BIN: 8.7

L4T: 36.4.4

Jetpack: MISSING

Hostname: jetson-desktop

Interfaces

enP8p1s0: 192.168.50.244

docker0: 172.17.0.1

1ALL 2GPU 3CPU 4MEM 5ENG 6CTRL 7INFO Quit

(c) 2024, RB

??????????

JetPack 6.1 (rev1) JetPack 5.1.5

```
sudo nvpmodel -m 2 #nano 2super
sudo jetson_clocks --fan
```

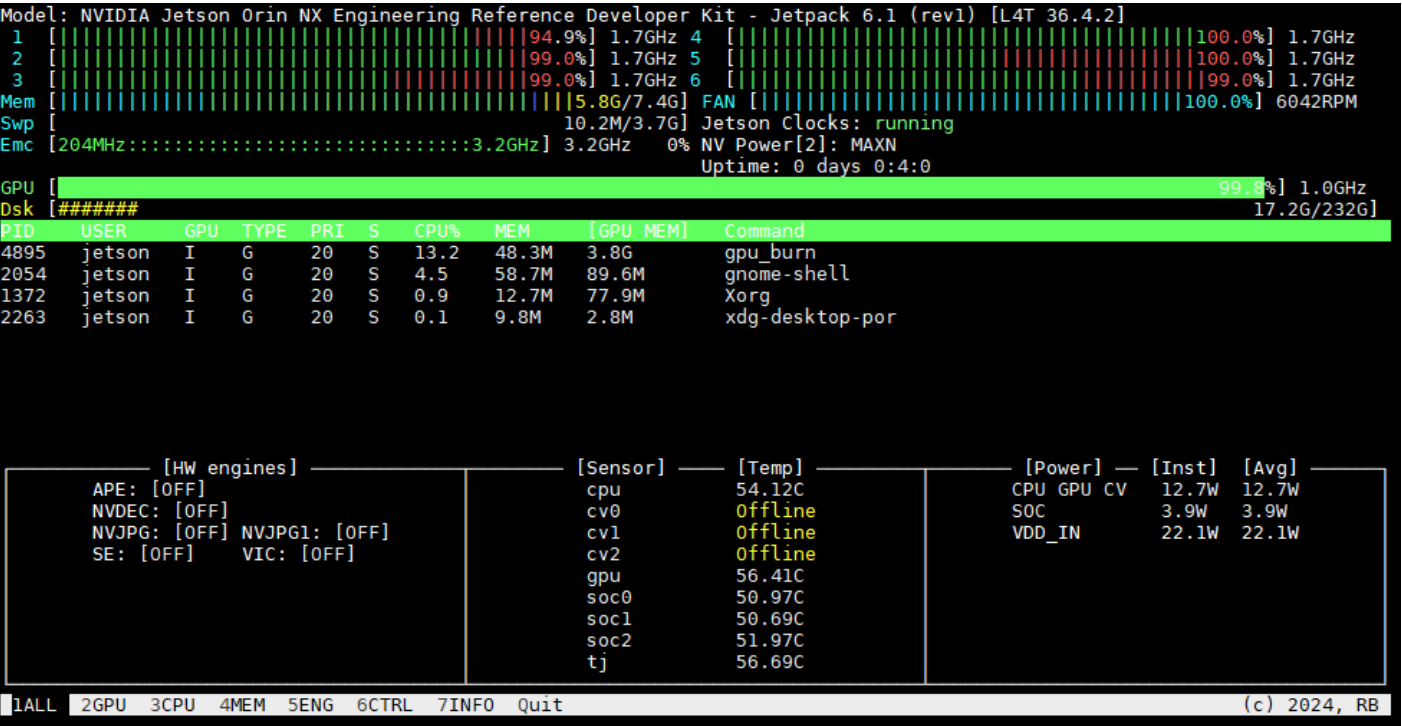
cpu

```
sudo apt install stress
stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --hdd 1 --hdd-bytes 1024M
```

gpu

```
git clone https://github.com/anseeto/jetson-gpu-burn/
cd jetson-gpu-burn
make
./gpu_burn 100000
```

```
sudo jtop
```



Orin Nano 30W.

Orin NX 40W.

?????opencv with cuda

JetPack███ opencv███ cuda██████████

```
jtop 4.3.2 - (c) 2024, Raffaello Bonghi [raffaello@rnext.it]
Website: https://rnext.it/jetson_stats

Platform
XXX]chine: aarch64
System: Linux
Distribution: Ubuntu 22.04 Jammy Jellyfish
Release: 5.15.148-tegra
Python: 3.10.12

Serial Number: [s|XX CLICK TO READ
Hardware
Model: NVIDIA Jetson Orin Nano Eng
699-level Part Number: 699-13767-0
P-Number: p3767-0003
Module: NVIDIA Jetson Orin Nano (8
SoC: tegra234
CUDA Arch BIN: 8.7
L4T: 36.4.4
Jetpack: MISSING

Libraries
CUDA: 12.6.68
cuDNN: 9.3.0.75
TensorRT: 10.3.0.30
VPI: 3.2.4
Vulkan: 1.3.204
OpenCV: 4.8.0 with CUDA: NO

Hostname: jetson-desktop
Interfaces
enP8p1s0: 192.168.50.244
docker0: 172.17.0.1

1ALL 2GPU 3CPU 4MEM 5ENG 6CTRL 7INFO Quit (c) 2024, RB
```

1. ???????

```
#!/bin/bash
# opencv_install.sh
# Modified from
https://github.com/AastaNV/JEP/blob/master/script/install_opencv4.10.0_Jetpack6.1.sh

version="4.10.0"
folder="workspace"
remove_old=""

set -e

# Parse command-line arguments
for arg in "$@"; do
```

```

case $arg in
    --version=*)
        version="${arg#*=}"
        ;;
    --folder=*)
        folder="${arg#*=}"
        ;;
    --remove-old=*)
        remove_old="${arg#*=}"
        ;;
    --help|-h)
        echo "Usage: $0 [--version=4.x.x] [--folder=dir] [--remove-old=yes/no]"
        exit 0
        ;;
    *)
        echo "Unknown parameter: $arg"
        echo "Usage: $0 [--version=4.x.x] [--folder=dir] [--remove-old=yes/no]"
        exit 1
        ;;
esac

done

# Create installation directory if it doesn't exist
if [ ! -d "$folder" ]; then
    echo "Creating directory: $folder"
    mkdir -p "$folder"
fi

cd "$folder" || exit

# Old OpenCV removal logic
if [ -z "$remove_old" ]; then
    read -rp "Do you want to remove system-installed OpenCV? (yes/no): " remove_old
fi

case "$remove_old" in
    [yY] | [yY][eE][sS])
        echo "** Removing system OpenCV packages"
        sudo apt -y purge *libopencv*
        sudo apt -y autoremove
        ;;

```

```

*)
    echo "*** Skipping system OpenCV removal"
    ;;
esac

echo "-----"
echo "*** Installing dependencies (1/4)"
echo "-----"
sudo apt-get update
sudo apt-get install -y build-essential cmake git libgtk2.0-dev pkg-config \
    libavcodec-dev libavformat-dev libswscale-dev libgstreamer1.0-dev \
    libgstreamer-plugins-base1.0-dev python3-dev python3-numpy libtbb2 \
    libtbb-dev libjpeg-dev libpng-dev libtiff-dev libv4l-dev v4l-utils qv4l2 curl

# Verify essential dependencies installed
for dep in g++ cmake git pkg-config; do
    if ! command -v "$dep" > /dev/null; then
        echo "Error: $dep installation failed"
        exit 1
    fi
done

echo "-----"
echo "*** Downloading OpenCV ${version} (2/4)"
echo "-----"

# Check if source files already exist
download_opencv=false
download_contrib=false

if [ ! -f "opencv-${version}.zip" ]; then
    echo "Downloading opencv-${version}.zip"
    wget -O opencv-${version}.zip https://github.com/opencv/opencv/archive/${version}.zip || {
        echo "Download failed! Check your internet connection or verify the version exists"
        exit 1
    }
    download_opencv=true
else
    echo "opencv-${version}.zip exists, skipping download"
fi

```



```

if [ ! -f "opencv_contrib-${version}.zip" ]; then
    echo "Downloading opencv_contrib-${version}.zip"
    wget -O opencv_contrib-${version}.zip
    https://github.com/opencv/opencv_contrib/archive/${version}.zip || {
        echo "Download failed! Check your internet connection or verify the version exists"
        exit 1
    }
    download_contrib=true
else
    echo "opencv_contrib-${version}.zip exists, skipping download"
fi

# Unpack source files
if [ ! -d "opencv-${version}" ] || $download_opencv; then
    if [ -d "opencv-${version}" ]; then
        echo "Removing existing opencv-${version} directory"
        rm -rf "opencv-${version}"
    fi
    echo "Unpacking opencv-${version}.zip"
    unzip -q opencv-${version}.zip || {
        echo "Extraction failed! File may be corrupt"
        exit 1
    }
fi

if [ ! -d "opencv_contrib-${version}" ] || $download_contrib; then
    if [ -d "opencv_contrib-${version}" ]; then
        echo "Removing existing opencv_contrib-${version} directory"
        rm -rf "opencv_contrib-${version}"
    fi
    echo "Unpacking opencv_contrib-${version}.zip"
    unzip -q opencv_contrib-${version}.zip || {
        echo "Extraction failed! File may be corrupt"
        exit 1
    }
fi

# Clean up zip files after successful extraction
if [ $? -eq 0 ]; then

```

```

rm -f opencv-${version}.zip opencv_contrib-${version}.zip
fi

cd opencv-${version} || exit

echo "-----"
echo "*** Building OpenCV ${version} (3/4)"
echo "-----"
mkdir -p release
cd release

# Auto-detect CUDA architecture
cuda_arch=""
if command -v nvidia-smi &> /dev/null; then
    gpu_name=$(nvidia-smi --query-gpu=name --format=csv,noheader | head -n1)
    if [[ $gpu_name == *"Orin"* ]] || [[ $gpu_name == *"Jetson"* ]]; then
        cuda_arch="8.7"
    elif [[ $gpu_name == *"A100"* ]]; then
        cuda_arch="8.0"
    fi
fi

cmake_cmd="cmake -D WITH_CUDA=ON -D WITH_CUDNN=ON -D OPENCV_GENERATE_PKGCONFIG=ON "
cmake_cmd+="-D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-${version}/modules "
cmake_cmd+="-D WITH_GSTREAMER=ON -D WITH_LIBV4L=ON -D BUILD_opencv_python3=ON "
cmake_cmd+="-D BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -D BUILD_EXAMPLES=OFF "
cmake_cmd+="-D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local "

# Add CUDA architecture if detected
if [ -n "$cuda_arch" ]; then
    echo "Detected NVIDIA GPU: ${gpu_name}, using CUDA_ARCH_BIN=${cuda_arch}"
    cmake_cmd+="-D CUDA_ARCH_BIN=${cuda_arch} -D CUDA_ARCH_PTX=\"\" "
else
    echo "No supported GPU detected, skipping CUDA architecture flags"
fi

# Execute CMake configuration
echo "CMake command: $cmake_cmd .."
$cmake_cmd .. || {
    echo "CMake configuration failed"
}

```

```

    exit 1
}

# Parallel build (leave one core for system stability)
cpu_cores=$((nproc - 1))
[ $cpu_cores -lt 1 ] && cpu_cores=1
echo "Building with ${cpu_cores} CPU cores"
make -j${cpu_cores} || {
    echo "Compilation failed"
    exit 1
}

echo "-----"
echo "*** Installing OpenCV ${version} (4/4)"
echo "-----"
sudo make install || {
    echo "Installation failed"
    exit 1
}

# Add environment variables to .bashrc (only if not already present)
bashrc=~/.bashrc
env_lines=(
    "export LD_LIBRARY_PATH=/usr/local/lib:\$LD_LIBRARY_PATH"
    "export PYTHONPATH=/usr/local/lib/python3.10/site-packages/:\$PYTHONPATH"
)

for line in "${env_lines[@]"; do
    if ! grep -Fxq "$line" "$bashrc"; then
        echo "Adding to .bashrc: $line"
        echo "$line" >> "$bashrc"
    else
        echo "Environment variable already exists: $line"
    fi
done

source ~/.bashrc

echo "*** OpenCV ${version} installation completed"
echo "Verification commands:"

```

```
echo " pkg-config --modversion opencv4"
echo " python3 -c 'import cv2; print(cv2.__version__)'"

echo "*** Installation successful!"
```

2. ????

2.1 删除 opencv

```
sudo apt-get purge libopencv*
sudo apt autoremove
sudo apt-get update
```

2.2 安装依赖

```
sudo apt-get update
sudo apt-get install -y build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev
libavformat-dev libswscale-dev
sudo apt-get install -y libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev python3.10-dev
python3-numpy
sudo apt-get install -y libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libv4l-dev v4l-
utils qv4l2
sudo apt-get install -y curl
```

2.3 下载 opencv 源码 (版本 4.10.0)

```
version=4.10.0
wget -O "opencv-${version}.zip" "https://github.com/opencv/opencv/archive/${version}.zip"
wget -O "opencv_contrib-${version}.zip"
"https://github.com/opencv/opencv_contrib/archive/${version}.zip"
unzip "opencv-${version}.zip"
unzip "opencv_contrib-${version}.zip"
rm "opencv-${version}.zip" "opencv_contrib-${version}.zip"
cd "opencv-${version}/"
```

2.4 编译

编译选项



```
mkdir build
cd build/
cmake -D WITH_CUDA=ON -D WITH_CUDNN=ON -D CUDA_ARCH_BIN="8.7" -D CUDA_ARCH_PTX="" -D
OPENCV_GENERATE_PKGCONFIG=ON -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-
${version}/modules -D WITH_GSTREAMER=ON -D WITH_LIBV4L=ON -D BUILD_opencv_python3=ON -D
BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -D BUILD_EXAMPLES=OFF -D CMAKE_BUILD_TYPE=RELEASE -D
CMAKE_INSTALL_PREFIX=/usr/local ..
make -j$(nproc)
```

2.5

```
sudo make install
echo 'export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH' >> ~/.bashrc
echo 'export PYTHONPATH=/usr/local/lib/python3.10/site-packages/:$PYTHONPATH' >> ~/.bashrc
source ~/.bashrc
```

??????

```
jtop 4.3.2 - (c) 2024, Raffaello Bonghi [raffaello@rnext.it]
Website: https://rnext.it/jetson_stats

Platform
Machine: aarch64
System: Linux
Distribution: Ubuntu 22.04 Jammy Jellyfish
Release: 5.15.148-tegra
Python: 3.10.12

Serial Number: [s]XX CLICK TO READ XXX]
Hardware
Model: NVIDIA Jetson Orin NX Engineering Reference Developer Kit Super
699-level Part Number: 699-13767-0001-301 F.1
P-Number: p3767-0001
Module: NVIDIA Jetson Orin NX (8GB ram)
SoC: tegra234
CUDA Arch BIN: 8.7
L4T: 36.4.4
Jetpack: MISSING

Libraries
CUDA: 12.6.85
cuDNN: 9.3.0.75
TensorRT: 10.7.0.23
VPI: 3.2.4
Vulkan: 1.3.204
OpenCV: 4.10.0 with CUDA: YES

Hostname: jetson-desktop
Interfaces
wlp1p1s0: 192.168.50.188
docker0: 172.17.0.1
```

```

#--test_cuda.cpp

#include <opencv2/opencv.hpp>
#include <opencv2/core/cuda.hpp>
#include <opencv2/cudaarithm.hpp>
#include <iostream>
#include <chrono>

// CPU 测试
void cpu_matrix_mult(cv::Mat& a, cv::Mat& b, cv::Mat& result) {
    for (int i = 0; i < 50; i++) {
        result = a * b;
    }
}

// GPU 测试
void gpu_matrix_mult(cv::cuda::GpuMat& d_a, cv::cuda::GpuMat& d_b, cv::cuda::GpuMat& d_result)
{
    cv::cuda::Stream stream;

    for (int i = 0; i < 50; i++) {
        cv::cuda::gemm(d_a, d_b, 1.0, cv::cuda::GpuMat(), 0, d_result, 0, stream);
        stream.waitForCompletion();
    }
}

int main() {
    try {
        std::cout << "--- OpenCV CUDA Matrix Multiplication Test ---\n";

        // 创建 1000x1000 矩阵
        cv::Mat mat_a(1000, 1000, CV_32FC1);
        cv::Mat mat_b(1000, 1000, CV_32FC1);
        cv::randu(mat_a, 0.0f, 1.0f);
        cv::randu(mat_b, 0.0f, 1.0f);

        cv::Mat cpu_result;

        // CPU 测试
        auto start_cpu = std::chrono::high_resolution_clock::now();

```

```

    cpu_matrix_mult(mat_a, mat_b, cpu_result);
    auto end_cpu = std::chrono::high_resolution_clock::now();
    double cpu_time = std::chrono::duration_cast<std::chrono::milliseconds>(end_cpu -
start_cpu).count();

    // GPU
    cv::cuda::GpuMat d_mat_a, d_mat_b, d_result;
    d_mat_a.upload(mat_a);
    d_mat_b.upload(mat_b);

    auto start_gpu = std::chrono::high_resolution_clock::now();
    gpu_matrix_mult(d_mat_a, d_mat_b, d_result);
    auto end_gpu = std::chrono::high_resolution_clock::now();
    double gpu_time = std::chrono::duration_cast<std::chrono::milliseconds>(end_gpu -
start_gpu).count();

    //
    cv::Mat gpu_result;
    d_result.download(gpu_result);

    // (L2)
    double diff = cv::norm(cpu_result, gpu_result, cv::NORM_L2);
    std::cout << "Result difference: " << diff << "\n";

    std::cout << "Performance Results:\n"
        << " - CPU time: " << cpu_time << " ms\n"
        << " - GPU time: " << gpu_time << " ms\n"
        << " - Speedup: " << cpu_time / gpu_time << "x\n";

    std::cout << "\n CUDA matrix multiplication test completed\n";
    return 0;

} catch (const cv::Exception& e) {
    std::cerr << "OpenCV Error (" << e.err << "): " << e.what() << "\n";
    return -1;
} catch (const std::exception& e) {
    std::cerr << "Standard Error: " << e.what() << "\n";
    return -2;
}
}

```



```
jetson@jetson-desktop:~/work$ g++ test_cuda.cpp -o test_cuda `pkg-config --cflags --libs opencv4`
```

```
jetson@jetson-desktop:~/work$ ./test_cuda
```

```
--- OpenCV CUDA Performance Test ---
```

```
Performance Results:
```

- CPU time: 2451 ms
- GPU time: 918 ms
- Speedup: 2.66993x

```
□ CUDA performance test completed
```


pytorch torchvision

PyTorch

Python

API

Python

GPU

NVIDIA

Jetson

PyTorch Version	NVIDIA Framework Container	NVIDIA Framework Wheel	JetPack Version
2.8.0a0+5228986c39	25.06	-	6.2
2.8.0a0+5228986c39	25.05	-	6.2
2.7.0a0+79aa17489c	25.04	-	6.2
2.7.0a0+7c8ec84dab	25.03	-	6.2
2.7.0a0+6c54963f75	25.02	-	6.2
2.6.0a0+ecf3bae40a	25.01	-	6.1
2.6.0a0+df5bbc09d1	24.12	-	6.1
2.6.0a0+df5bbc0	24.11	-	6.1
2.5.0a0+e000cf0ad9	24.10	-	6.1
2.5.0a0+b465a5843b	24.09	24.09	6.1
2.5.0a0+872d972e41	24.08	-	6.0
2.4.0a0+3bcc3cddb5	24.07	24.07	6.0
2.4.0a0+f70bd71a48	24.06	24.06	6.0
2.4.0a0+07cecf4168	24.05	24.05	6.0
2.3.0a0+6ddf5cf85e	24.04	24.04	6.0 Developer Preview
2.3.0a0+40ec155e58	24.03	24.03	
2.3.0a0+ebedce2	24.02	24.02	
2.2.0a0+81ea7a4	23.12, 24.01	23.12, 24.01	
2.2.0a0+6a974bec	23.11	23.11	

PyTorch Version	NVIDIA Framework Container	NVIDIA Framework Wheel	JetPack Version
2.1.0a		23.06	5.1.x
2.0.0		23.05	
2.0.0a0+fe05266f		23.04	
2.0.0a0+8aa34602		23.03	
1.14.0a0+44dac51c		23.02, 23.01	
1.13.0a0+936e930		22.11	5.0.2
1.13.0a0+d0d6b1f		22.09, 22.10	
1.13.0a0+08820cb	22.07	22.07	
1.13.0a0+340c412	22.06	22.06	5.0.1
1.12.0a0+8a1a93a9	22.05	22.05	5.0
1.12.0a0+bd13bc66		22.04	
1.12.0a0+2c916ef		22.03	
1.11.0a0+bfe5ad28		22.01	4.6.1

📁📁📁📁 **JetPack6.2.1 cuda12.6** 📁📁📁

1.??torch???

1.1?????torch , torchvision

```
wget https://pypi.jetson-ai-lab.io/jp6/cu126/+f/62a/1beee9f2f1470/torch-2.8.0-cp310-cp310-  
linux_aarch64.whl  
wget https://pypi.jetson-ai-lab.io/jp6/cu126/+f/907/c4c1933789645/torchvision-0.23.0-cp310-  
cp310-linux_aarch64.whl  
pip install torch-2.8.0-cp310-cp310-linux_aarch64.whl torchvision-0.23.0-cp310-cp310-  
linux_aarch64.whl -i https://pypi.tuna.tsinghua.edu.cn/simple
```

1.2 ?????????

📁 python📁📁📁📁📁📁

```
jetson@jetson-desktop:~$ python  
Python 3.10.16 (main, Dec 11 2024, 16:18:56) [GCC 11.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import torch
>>> print(torch.__version__)
2.8.0
>>> print(torch.cuda.is_available())
True
```

2. ??YOLO11

YOLO

2.1 ??miniconda

```
curl -L https://repo.anaconda.com/miniconda/Miniconda3-py310_25.3.1-Linux-aarch64.sh | bash
source ~/miniconda3/bin/activate
conda --version
```

2.2 conda??

```
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkgs/main/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkgs/free/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/cloud/conda-forge/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/cloud/msys2/
conda config --set show_channel_urls yes
```

2.3??conda??

```
conda create -n jetson-ai python=3.10
```

2.4 ??conda??

```
conda activate jetson-ai
```

2.5 ??torch?torchvision

```
wget https://pypi.jetson-ai-lab.io/jp6/cu126/+f/62a/1beee9f2f1470/torch-2.8.0-cp310-cp310-  
linux_aarch64.whl  
wget https://pypi.jetson-ai-lab.io/jp6/cu126/+f/907/c4c1933789645/torchvision-0.23.0-cp310-  
cp310-linux_aarch64.whl  
pip install torch-2.8.0-cp310-cp310-linux_aarch64.whl torchvision-0.23.0-cp310-cp310-  
linux_aarch64.whl -i https://pypi.tuna.tsinghua.edu.cn/simple
```

2.6 ??ultralytics

```
pip install ultralytics -i https://pypi.tuna.tsinghua.edu.cn/simple
```

2.7 ?????????



```
import cv2
import time
from ultralytics import YOLO
from ultralytics import YOLOWorld

# Load the YOLO model
model = YOLO("yolo11s.pt")

# Open the video file
video_path = 0
cap = cv2.VideoCapture(video_path)

# Loop through the video frames
while cap.isOpened():

    # Read a frame from the video
    success, frame = cap.read()
    start = time.time()

    if success:
        # Run YOLO inference on the frame
        results = model(frame)

        inf_time = time.time() - start

        # Visualize the results on the frame
        annotated_frame = results[0].plot()

        fps = 1.0 / inf_time if inf_time > 0 else 0

        # show FPS
        cv2.putText(annotated_frame, f"FPS: {fps:.2f}", (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0,255,0), 2)

        cv2.imshow("YOLO Inference", annotated_frame)

        # Break the loop if 'q' is pressed
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break
```

```
else:
```

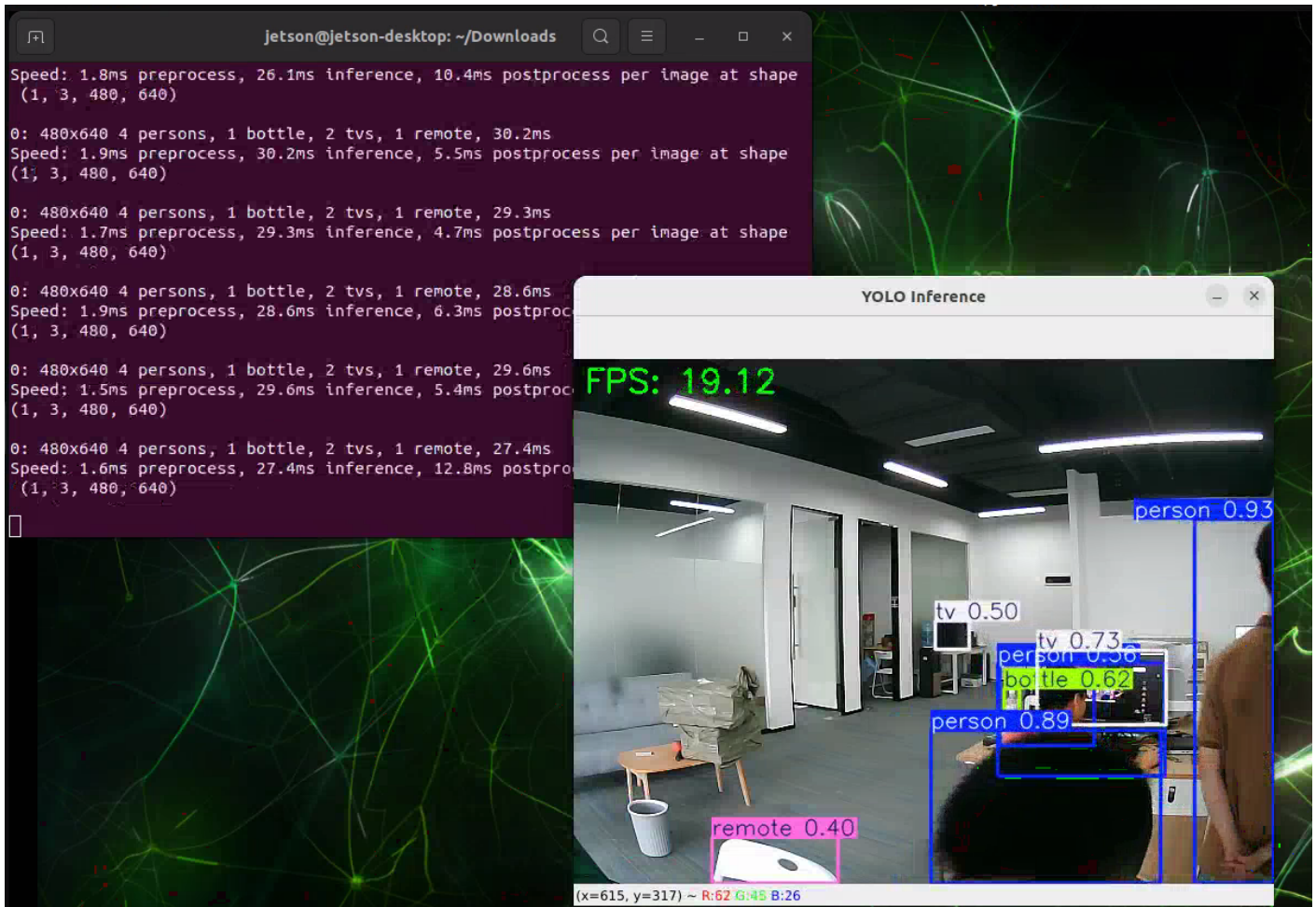
```
    # Break the loop if the end of the video is reached
```

```
    break
```

```
# Release the video capture object and close the display window
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```



[Ultralytics YOLO11 -Ultralytics YOLO](#)

????ollama

1. ????????ollama

```
curl -fsSL https://ollama.com/install.sh | sh
```

2. ?????????

```
jetson@jetson-desktop:~$ ollama -v  
ollama version is 0.9.6
```

3. ?????deepseek-r1 1.5b??

```
ollama run deepseek-r1:1.5b
```



```
jetson@jetson-desktop: ~  
jetson@jetson-desktop:~$ ollama run deepseek-r1:1.5b  
>>> 介绍一下英伟达  
Thinking...  
好，我现在要介绍一下英伟达。首先，我知道英伟达是计算机公司，主要生产GPU，这些设备用于加速机器学习和AI训练。不过我有点记不太清楚了，可能需要再确认一下。  
  
我记得英伟达最初是为游戏而设计的，后来发展成电脑图形处理器（GPU）。他们生产不同的架构，比如RTX、TGPU和HPU，这些都有不同的性能和用途。比如RTX可能用于普通工作负载，而HPU则适合高计算任务，比如AI训练。  
  
我应该先从基本信息开始介绍，比如成立时间、公司的背景、主要产品线等。然后详细介绍GPU技术，包括架构、核心数、功耗管理等等。还要提到英伟达在AI和游戏中的应用，以及他们目前的市场表现和未来计划。  
  
嗯，可能还需要提及他们与NVIDIA的关系，毕竟这两个公司在图形处理器领域有合作。还有可能提到他们的产品线发展如何影响整个计算机行业的发展。  
  
现在把这些点组织成一个连贯的介绍，确保信息准确，并且逻辑清晰。  
...done thinking.  
  
英伟达是一家全球知名的计算机公司，总部位于美国加利福尼亚州伯克利，专注于开发高性能计算设备和图形处理器（GPU）。它们在电子、汽车、医疗保健等多个领域
```

????Jetson-container

docker

Jetson-container NVIDIA Jetson
AI

Docker

CUDA □ cuDNN □ TensorRT

Jetson Orin NX 16GB JetPack6.2.1 Comfy-UI

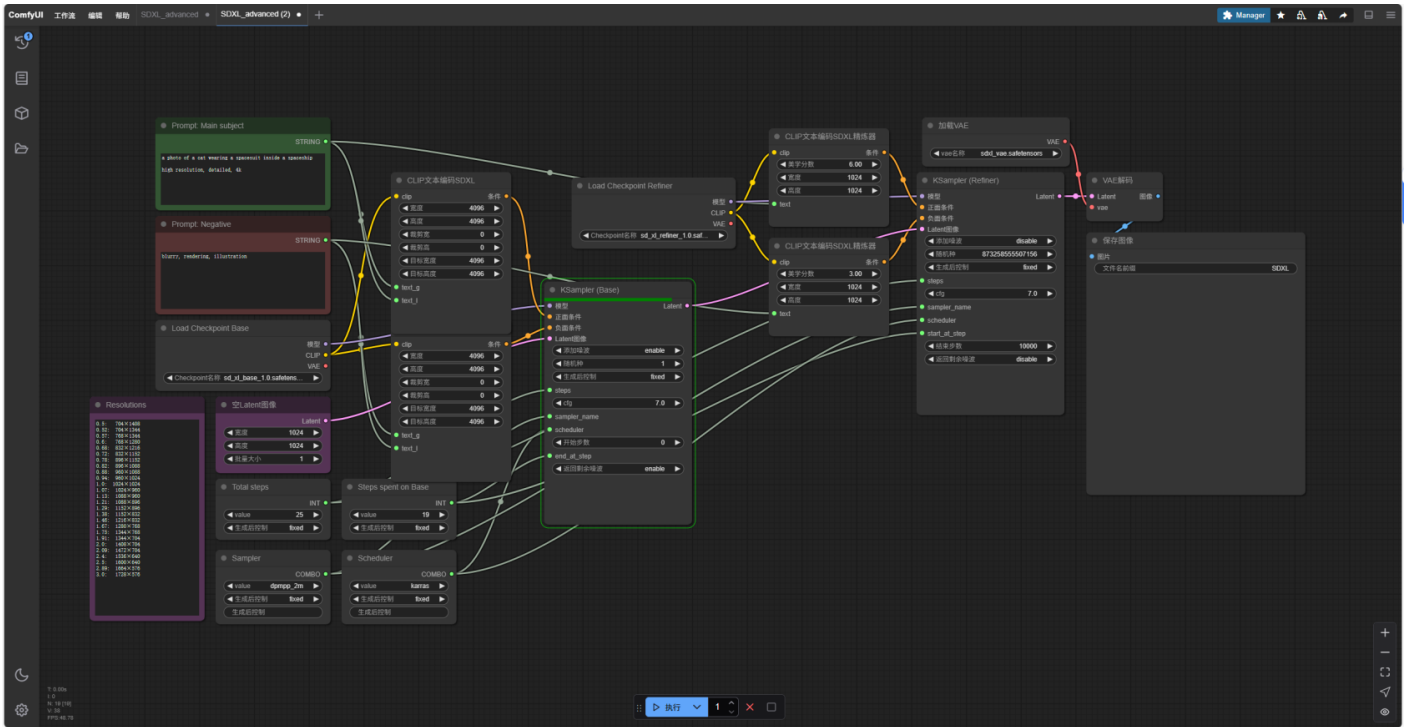
--	--	--	--

Comfy-UI

[illegible]

LoRA

ControlNet[]

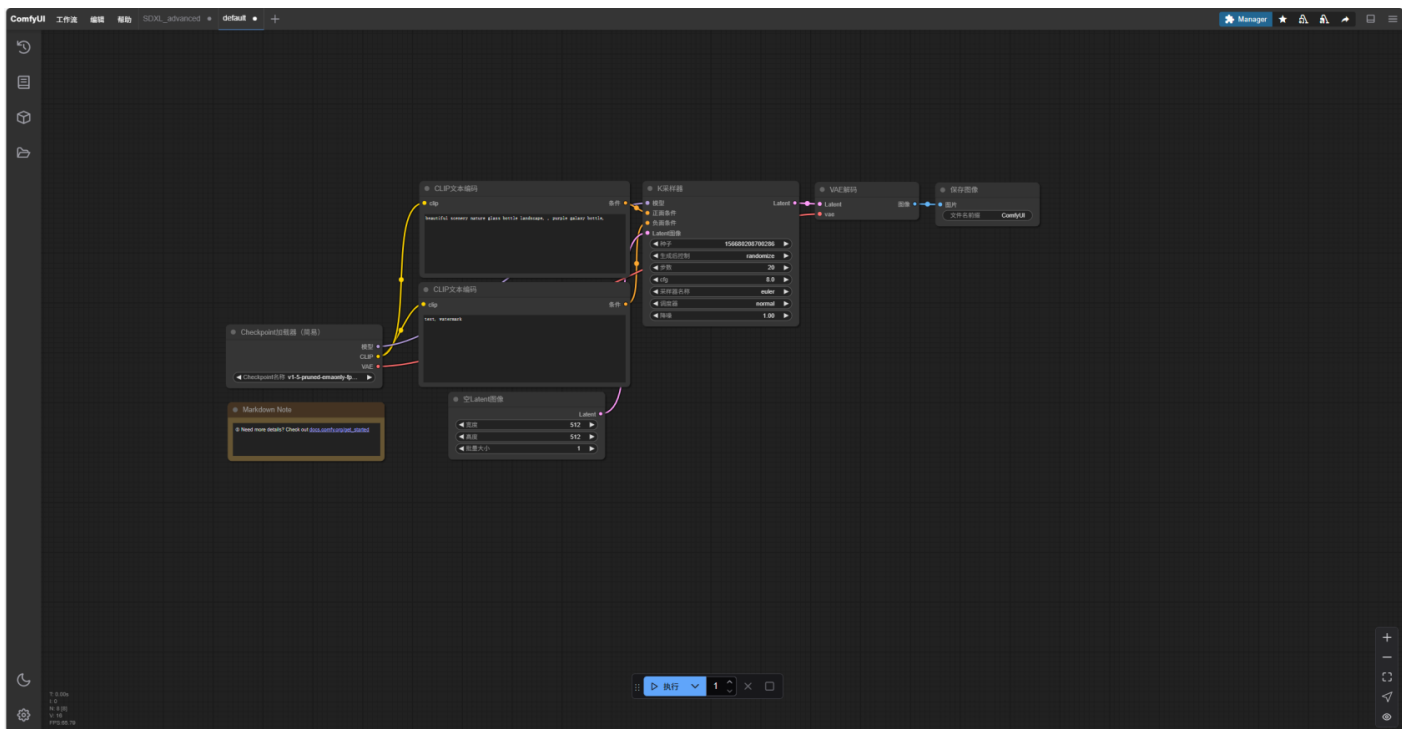
1. `jetson-container`

```
git clone https://github.com/dusty-nv/jetson-containers.git
cd jetson-containersbash
jetson-containers/install.sh
```

2. ☐ docker☐☐☐☐☐

```
jetson-containers run dustynv/comfyui:r36.4.3
```

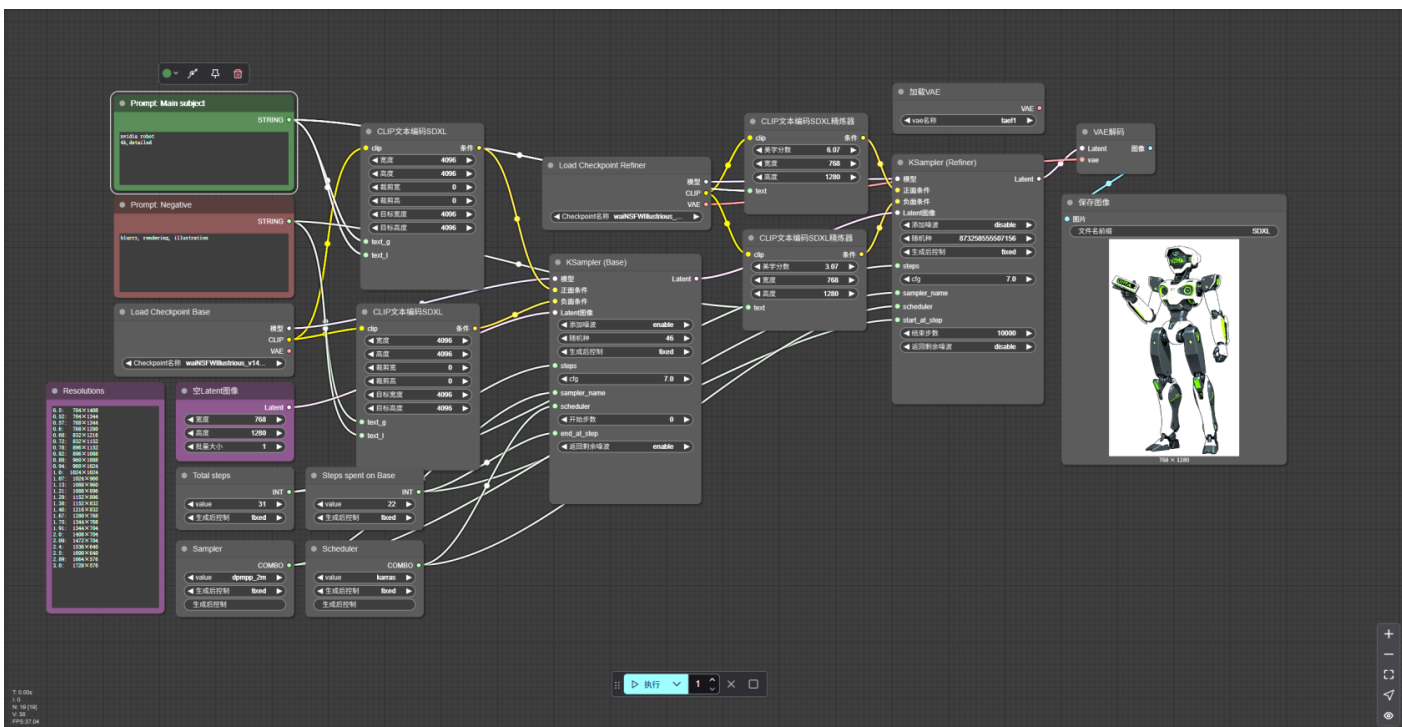
[illegible]



5.

```
jetson-containers run ~/models/~/opt/ComfyUI/models/checkpoints dustynv/comfyui:r36.4.3
```

6.



ComfyUI

ComfyUI | AI

????

JetPack6

Intel 8625NGW

Intel AX200

AX210

Intel

1.

```
sudo lshw -C network
```

product: Wi-Fi 6 AX200

UNCLAIMED

2. iwlmwifi

iwlmwifi Intel

Linux

Centrino Wi-Fi 6/6E/7

Linux

```
sudo apt update
```

```
sudo apt install backport-iwlmwifi-dkms
```

3.

```
sudo reboot
```

4.

```
jetson@jetson-desktop:~$ sudo lshw -C network
```

```
[sudo] password for jetson:
```

```
*-network
```

```
description: Wireless interface
```

```
product: Wi-Fi 6 AX200
```

```
vendor: Intel Corporation
```

```
physical id: 0
```

```
bus info: pci@0001:01:00.0
```

```
logical name: wlan0
```

```
version: 1a
```

```
serial: ac:12:03:a0:4c:db
```

```
width: 64 bits
```

```
clock: 33MHz
```

```
capabilities: pm msi pciexpress msix bus_master cap_list ethernet physical wireless
```

```
configuration: broadcast=yes driver=iwlwifi driverversion=5.10.216-tegra
firmware=59.601f3a66.0 cc-a0-59.ucode latency=0 link=no multicast=yes wireless=IEEE 802.11
resources: irq:55 memory:20a8000000-20a8003fff
*-network
```

description

Wireless interface / Ethernet interface

4G??????

Diagram illustrating the mapping of 5G NR frequency ranges to various 4G LTE and M2M standards:

- 3-5 MHz: LTE Standard EM05
- 10-15 MHz: IoT/M2M
- 15-20 MHz: LTE Cat 4
- 20-25 MHz: M.2
- 25-30 MHz: NGFF

EM05 [] [] [] [] [] [] [] [] 150 Mbps [] [] [] [] [] [] 50 Mbps, [] [] [] [] [] [] EM05-CN [] EM05-E []
EM05-G [] [] LTE-FDD [] LTE-TDD [] DC-HSDPA [] HSPA+ [] HSDPA [] HSUPA [] WCDMA [] CDMA
[] [] [] [] [] []

Jetson EM-05

????????

1.1 ????????

1.2 ?????????

1.3 ?????????

1.4 ?????????

??????????

2.1 ????????

-

```
sudo apt install build-essential bc git bison flex libssl-dev zip libncurses-dev make git
```

- | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|
| | | | | | | | | |
|--|--|--|--|--|--|--|--|--|

```
mkdir $HOME/l4t-gcc-toolchain
cd $HOME/l4t-gcc-toolchain
```

- [illegible]

```
wget https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v3.0/toolchain/aarch64--glibc--stable-2022.08-1.tar.bz2
tar xf aarch64--glibc--stable-2022.08-1.tar.bz2
```

-
- A horizontal number line starting with a solid black dot at 0. To the right of the dot are 11 empty rectangular boxes, each representing a unit interval for counting.

```
cd
$HOME/nvidia/nvidia_sdk/JetPack_6.2.1_Linux_JETSON_ORIN_NANO_TARGETS/Linux_for_Tegra/source #
■■■■■■■■■■
```

- 选择 release tag 查看 [Jetson Linux Archive | NVIDIA Developer](#)

```
./source_sync.sh -k -t <release-tag> #<release-tag>■■■■■■■■■■ jetson_36.4.3
```

2.2 ??????

■■■■■■■■ ■■ [Linux&Andriod](#)■■■■■■■■■■

Jetson Orin Nano■■■■■■■■ arch/arm64/configs/defconfig

- 选择 USB■■■■■■■■■■

```
CONFIG_USB_SERIAL=y
CONFIG_USB_SERIAL_WWAN=y
CONFIG_USB_SERIAL_OPTION=y
CONFIG_USB_NET_DRIVERS=y
CONFIG_USB_USBNET=y
CONFIG_USB_NET_QMI_WWAN=y
CONFIG_USB_WDM=y
```

- 选择 drivers/usb/serial/option.c 选择 USB■■■■■■■■ VID 和 PID

```
static const struct usb_device_id option_ids[] = {
#if 1 // 2025-04-24 Added by Quectel
    { USB_DEVICE(0x2C7C, 0x0125) },
#endif
    ... ..
}
```

- 选择 USBNet■■■■■■■■ drivers/usb/serial/option.c

```
static int option_probe(struct usb_serial *serial,
                        const struct usb_device_id *id)
{
    ... ..
}
```

```

#if 1 // 2025-04-24 Added by Quectel
if (serial->dev->descriptor.idVendor == cpu_to_le16(0x2C7C)) {
    __u16 idProduct = le16_to_cpu(serial->dev->descriptor.idProduct);
    struct usb_interface_descriptor *intf = &serial->interface->cur_altsetting->desc;

    if (intf->bInterfaceClass != 0xFF || intf->bInterfaceSubClass == 0x42) {
        //ECM, RNDIS, NCM, MBIM, ACM, UAC, ADB
        return -ENODEV;
    }

    if ((idProduct&0xF000) == 0x0000) {
        //MDM interface 4 is QMI
        if (intf->bInterfaceNumber == 4 && intf->bNumEndpoints == 3
            && intf->bInterfaceSubClass == 0xFF && intf->bInterfaceProtocol ==
0xFF)
            return -ENODEV;
    }
}
#endif

/* Store the device flags so we can use them during attach. */
usb_set_serial_data(serial, (void *)device_flags);

return 0;
}

```

-  drivers/usb/serial/usb_wwan.c

```

static struct urb *usb_wwan_setup_urb(struct usb_serial_port *port,
                                     int endpoint,
                                     int dir, void *ctx, char *buf, int len,
                                     void (*callback) (struct urb *))
{
    ... ..
    usb_fill_bulk_urb(urb, serial->dev,
                     usb_sndbulkpipe(serial->dev, endpoint) | dir,
                     buf, len, callback, ctx);

#if 1 //2025-04-24 Added by Quectel for zero packet
    if (dir == USB_DIR_OUT) {

```

```

        struct usb_device_descriptor *desc = &serial->dev->descriptor;

        if (desc->idVendor == cpu_to_le16(0x2C7C))
            urb->transfer_flags |= URB_ZERO_PACKET;
    }
#endif
    return urb;
}

```

- `Reset-resume` `drivers/usb/serial/option.c`

```

static struct usb_serial_driver option_lport_device = {
    ... ..
#ifdef CONFIG_PM
    .suspend          = usb_wwan_suspend,
    .resume           = usb_wwan_resume,
#if 1 //2025-04-24 Added by Quectel
    .reset_resume     = usb_wwan_resume,
#endif
#endif
};

```

- `QMI_WWAN` `drivers/net/usb/qmi_wwan.c` `drivers/net/usb/Makefile`

```

#Makefile
# must insert qmi_wwan_q.o before qmi_wwan.o
obj-$(CONFIG_USB_NET_QMI_WWAN) += qmi_wwan_q.o
obj-$(CONFIG_USB_NET_QMI_WWAN) += qmi_wwan.o

```

2.3 ????

`kernel_out`

- `kernel_out`

```

cd ../../
mkdir kernel_out

```

- 编译内核

```
export CROSS_COMPILE=$HOME/l4t-gcc-toolchain/aarch64--glibc--stable-2022.08-1/bin/aarch64-
buildroot-linux-gnu-          #交叉编译工具链
make -C kernel                  #编译 Jetson Linux 内核
sudo -E make install -C kernel  #安装内核
```

- 复制内核文件

```
cp kernel/kernel-jammy-src/arch/arm64/boot/Image ../Linux_for_Tegra/kernel/Image
```

- 编译 NVIDIA 设备树

```
export CROSS_COMPILE=$HOME/l4t-gcc-toolchain/aarch64--glibc--stable-2022.08-1/bin/aarch64-
buildroot-linux-gnu-
export KERNEL_HEADERS=$PWD/kernel/kernel-jammy-src
make modules
```

- 安装设备树

```
export
INSTALL_MOD_PATH=$HOME/nvidia/nvidia_sdk/JetPack_6.2.1_Linux_JETSON_ORIN_NANO_TARGETS/Linux_for_Tegra/rootfs/ #安装设备树
sudo -E make modules_install
```

- 编译 DTB

```
export CROSS_COMPILE=$HOME/l4t-gcc-toolchain/aarch64--glibc--stable-2022.08-1/bin/aarch64-
buildroot-linux-gnu-
export KERNEL_HEADERS=$PWD/kernel/kernel-jammy-src
make dtbs
```

- 复制 dtb 文件

```
cp kernel-devicetree/generic-dts/dtbs/* ../kernel/dtb/
```

2.4 烧录

- 烧录固件

```
cd
$HOME/nvidia/nvidia_sdk/JetPack_6.2.1_Linux_JETSON_ORIN_NANO_TARGETS/Linux_for_Tegra/source #
```


□□□□□□□□

□□□□□□

- super□□

```
sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 -c
tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c bootloader/generic/cfg/flash_t234_qspi.xml"
--showlogs --network usb0 jetson-orin-nano-devkit-super internal
```

- □□□□

```
sudo ./flash.sh jetson-orin-nano-devkit-nvme internal
```

2.5 ????

- □ Quectel_Qconnectmanager□□□□□
- □□□□

```
unzip Quectel_QConnectManager_Linux_V1.6.5.zip
cd Quectel_QConnectManager_Linux_V1.6.5
make
```

□□□□□

4G□□□□

```
jetson@jetson-desktop:~/Downloads/Quectel_QConnectManager_Linux_V1.6.5$ sudo ./quectel-CM
[08-08_11:39:04:415] QConnectManager_Linux_V1.6.5
[08-08_11:39:04:416] Find /sys/bus/usb/devices/1-2.2 idVendor=0x2c7c idProduct=0x125,
bus=0x001, dev=0x007
[08-08_11:39:04:416] Auto find qmichannel = /dev/cdc-wdm0
[08-08_11:39:04:417] Auto find usbnet_adapter = wwan0
[08-08_11:39:04:417] netcard driver = qmi_wwan_q, driver version = V1.2.6
[08-08_11:39:04:417] Modem works in QMI mode
[08-08_11:39:04:447] cdc_wdm_fd = 7
[08-08_11:39:04:523] Get clientWDS = 5
[08-08_11:39:04:557] Get clientDMS = 1
[08-08_11:39:04:589] Get clientNAS = 2
[08-08_11:39:04:620] Get clientUIM = 1
[08-08_11:39:04:653] Get clientWDA = 1
[08-08_11:39:04:684] requestBaseBandVersion EM05CNFDR08A03M1G_ND
[08-08_11:39:04:812] requestGetSIMStatus SIMStatus: SIM_READY
[08-08_11:39:04:876] requestGetProfile[pdp:1 index:1] ctnet///0/IPV4V6
```

```
[08-08_11:39:04:908] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
[08-08_11:39:04:940] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[08-08_11:39:04:941] ip addr flush dev wwan0
[08-08_11:39:04:947] ip link set dev wwan0 down
[08-08_11:39:05:003] requestSetupDataCall WdsConnectionIPv4Handle: 0x8723e530
[08-08_11:39:05:132] ip link set dev wwan0 up
[08-08_11:39:05:141] No default.script found, it should be in '/usr/share/udhcpc/' or
'/etc//udhcpc' depend on your udhcpc version!
[08-08_11:39:05:142] busybox udhcpc -f -n -q -t 5 -i wwan0
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 10.21.181.66
udhcpc: lease of 10.21.181.66 obtained, lease time 7200
[08-08_11:39:05:282] ip -4 address flush dev wwan0
[08-08_11:39:05:286] ip -4 address add 10.21.181.66/30 dev wwan0
[08-08_11:39:05:292] ip -4 route add default via 10.21.181.65 dev wwan0
```

□□□□□

```
jetson@jetson-desktop:~$ ifconfig wwan0
wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500
    inet 10.21.181.66 netmask 255.255.255.252
    inet6 fe80::5804:41ff:feda:ce83 prefixlen 64 scopeid 0x20<link>
    ether 5a:04:41:da:ce:83 txqueuelen 1000 (Ethernet)
    RX packets 9 bytes 2304 (2.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22 bytes 1854 (1.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

□□□□□

```
[08-08_13:35:08:924] Get clientNAS = 2
[08-08_13:35:08:956] Get clientUIM = 1
[08-08_13:35:08:988] Get clientWDA = 1
[08-08_13:35:09:020] requestBaseBandVersion EM05CNFDR08A03M1G_ND
[08-08_13:35:09:148] requestGetSIMStatus SIMStatus: SIM_READY
[08-08_13:35:09:212] requestGetProfile[pdp:1 index:1] ctnet///0/IPV4V6
[08-08_13:35:09:244] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached,
DataCap: LTE
[08-08_13:35:09:276] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[08-08_13:35:09:276] ip addr flush dev wwan0
[08-08_13:35:09:281] ip link set dev wwan0 down
[08-08_13:35:09:341] requestSetupDataCall WdsConnectionIPv4Handle: 0x86d75fc0
[08-08_13:35:09:468] ip link set dev wwan0 up
[08-08_13:35:09:474] No default.script found, it should be in '/usr/share/udhcpc
/' or '/etc//udhcpc' depend on your udhcpc version!
[08-08_13:35:09:474] busybox udhcpc -f -n -q -t 5 -i wwan0
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 10.21.181.66
udhcpc: lease of 10.21.181.66 obtained, lease time 7200
[08-08_13:35:09:645] ip -4 address flush dev wwan0
[08-08_13:35:09:649] ip -4 address add 10.21.181.66/30 dev wwan0
[08-08_13:35:09:657] ip -4 route add default via 10.21.181.65 dev wwan0
```

```
jetson@jetson-desktop: ~  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=9 ttl=115 time=95.7 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=10 ttl=115 time=96.9 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=11 ttl=115 time=105 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=12 ttl=115 time=105 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=13 ttl=115 time=96.4 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=14 ttl=115 time=105 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=15 ttl=115 time=106 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=16 ttl=115 time=103 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=17 ttl=115 time=96.0 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=18 ttl=115 time=104 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=19 ttl=115 time=103 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=20 ttl=115 time=104 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=21 ttl=115 time=101 ms  
64 bytes from 150.171.28.10 (150.171.28.10): icmp_seq=22 ttl=115 time=101 ms
```

??????

????

- | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|
| | | | | | | | | |
|--|--|--|--|--|--|--|--|--|

```
sudo apt install build-essential bc git bison flex libssl-dev zip libncurses-dev make git
```

--	--	--	--	--	--	--	--	--

```
mkdir $HOME/l4t-gcc-toolchain
cd $HOME/l4t-gcc-toolchain
```

-
- A horizontal number line with arrows at both ends. There are 11 vertical tick marks, creating 10 equal intervals. A solid black dot is placed on the first tick mark to the right of the origin. A horizontal bracket is drawn above the line, starting from the first tick mark and ending at the second tick mark. Below the bracket is the fraction $\frac{1}{10}$.

```
wget https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v3.0/toolchain/aarch64-  
glibc--stable-2022.08-1.tar.bz2  
tar xf aarch64--glibc--stable-2022.08-1.tar.bz2
```

-
- A horizontal number line is shown. On the left end, there is a solid black dot representing the number 0. To the right of this dot is a horizontal line with 10 vertical tick marks, creating 10 equal intervals. The first tick mark after 0 is labeled with the number 1. The remaining 9 intervals are empty.

```
cd $HOME/nvidia/nvidia_sdk/JetPack_6.2.1_Linux_JETSON_ORIN_NANO_TARGETS/Linux_for_Tegra/source
#
```

- 111

```
./source_sync.sh -k -t <release-tag> #<release-tag> jetson_36.4.3
```

??????????

Intel????

```
/usr/lib/modules/$(uname -r)/updates
```

--	--	--	--	--	--	--	--	--

Linux for Tegra/rootfs/lib/modules/5.15.148-tegra/updates/ 

realtek 8125??

[illegible]

ko□□□

```
Linux for Tegra/rootfs/lib/modules/5.15.148-tegra/updates/dkms
```

r8125.ko

```
sudo depmod -a
sudo modprobe r8125
```

4G??

4G [2.2](#)

??HDMI 4K

tegra234-dcb-p3737-0000.dtsi

nvidia,dcb-image HDMI-dcb public/overlay/tegra234-dcb-p3767-0000-hdmi.dts

source/hardware/nvidia/t23x/nv-

USB???

tegra234-p3768-0000.dtsi padctl@3520000

usb3-2

```
padctl@3520000 {
    status = "okay";

    pads {
        usb2 {
            lanes {
                usb2-0 {
                    nvidia,function = "xusb";
                    status = "okay";
                };

                usb2-1 {
                    nvidia,function = "xusb";
                    status = "okay";
                };

                usb2-2 {
                    nvidia,function = "xusb";
                    status = "okay";
                };
            };
        };
    };
};
```

```

usb3 {
    lanes {
        usb3-0 {
            nvidia,function = "xusb";
            status = "okay";
        };

        usb3-1 {
            nvidia,function = "xusb";
            status = "okay";
        };

        usb3-2 {
            nvidia,function = "xusb";
            status = "okay";
        };
    };
};

ports {
    /* recovery port */
    usb2-0 {
        mode = "otg";
        vbus-supply = <&vdd_5v0_sys>;
        status = "okay";
        usb-role-switch;
    };

    /* hub */
    usb2-1 {
        mode = "host";
        vbus-supply = <&vdd_1v1_hub>;
        status = "okay";
    };

    /* M.2 Key-E */
    usb2-2 {

```

```

mode = "host";
vbus-supply = <&vdd_5v0_sys>;
status = "okay";
};

/* hub */
usb3-0 {
    nvidia,usb2-companion = <1>;
    status = "okay";
};

/* J5 */
usb3-1 {
    nvidia,usb2-companion = <0>;
    status = "okay";
};
usb3-2 {
    nvidia,usb2-companion = <2>;
    status = "okay";
};
};
};

usb@3550000 {
    status = "okay";

    phys = <&{/bus@0/padctl@3520000/pads/usb2/lanes/usb2-0}>,
          <&{/bus@0/padctl@3520000/pads/usb3/lanes/usb3-1}>;
    phy-names = "usb2-0", "usb3-0";
};

usb@3610000 {
    status = "okay";

    phys = <&{/bus@0/padctl@3520000/pads/usb2/lanes/usb2-0}>,
          <&{/bus@0/padctl@3520000/pads/usb2/lanes/usb2-1}>,
          <&{/bus@0/padctl@3520000/pads/usb2/lanes/usb2-2}>,
          <&{/bus@0/padctl@3520000/pads/usb3/lanes/usb3-0}>,
          <&{/bus@0/padctl@3520000/pads/usb3/lanes/usb3-1}>,

```

```

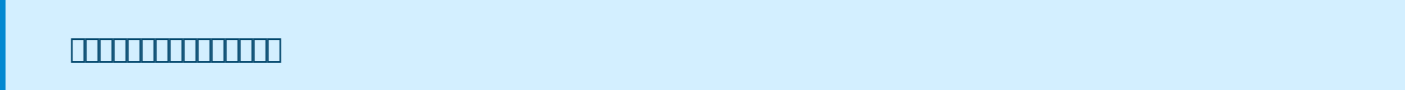
    <{/bus@0/padctl@3520000/pads/usb3/lanes/usb3-2}>;

    phy-names = "usb2-0", "usb2-1", "usb2-2", "usb3-0",
    "usb3-1", "usb3-2";
};

```

????????????

2.3 ????



- | | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | |
|--|--|--|--|--|--|--|--|

```
cd ../../  
mkdir kernel_out
```

- | | | | | | | |
|--|--|--|--|--|--|--|
| | | | | | | |
|--|--|--|--|--|--|--|

```
export CROSS_COMPILE=$HOME/l4t-gcc-toolchain/aarch64--glibc--stable-2022.08-1/bin/aarch64-
buildroot-linux-gnu- #编译
make -C kernel #编译 Jetson Linux 内核
sudo -E make install -C kernel #安装内核
```

- [illegible]

```
cp kernel/kernel-jammy-src/arch/arm64/boot/Image ../kernel/
```

-  NVIDIA

```
export CROSS_COMPILE=$HOME/l4t-gcc-toolchain/aarch64--glibc--stable-2022.08-1/bin/aarch64-  
buildroot-linux-gnu-  
export KERNEL_HEADERS=$PWD/kernel/kernel-jammy-src  
make modules
```

- □□□□□□□□

```
export
INSTALL_MOD_PATH=$HOME/nvidia/nvidia_sdk/JetPack_6.2.1_Linux_JETSON_ORIN_NANO_TARGETS/Linux_for_Tegra/rootfs/ #0000000000
sudo -E make modules_install
```


- `dtb`

```
export CROSS_COMPILE=$HOME/l4t-gcc-toolchain/aarch64--glibc--stable-2022.08-1/bin/aarch64-  
buildroot-linux-gnu-  
export KERNEL_HEADERS=$PWD/kernel/kernel-jammy-src  
make dtbs
```

- `dtb`

```
cp kernel-devicetree/generic-dts/dtbs/* ../kernel/dtb/
```

??????

- `flash`

```
sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 \  
-c tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c  
bootloader/generic/cfg/flash_t234_qspi.xml" \  
--showlogs --network usb0 jetson-orin-nano-devkit internal
```

- `super`

```
sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 \  
-c tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c  
bootloader/generic/cfg/flash_t234_qspi.xml" \  
--showlogs --network usb0 jetson-orin-nano-devkit-super internal
```

???QSPI???

- `flash`

```
sudo ./flash.sh -c bootloader/t186ref/cfg/flash_t234_qspi.xml --no-systemimg jetson-orin-nano-  
devkit nvme0n1p1
```

- `super`

```
sudo ./flash.sh -c bootloader/t186ref/cfg/flash_t234_qspi.xml --no-systemimg jetson-orin-nano-  
devkit-super nvme0n1p1
```


?????LOGO

UEFI

NVIDIA LOGO
UEFI

UEFI

docker

docker

docker

```
sudo apt install docker.io
```

docker

```
sudo usermod -a -G docker ${USER}
sudo reboot
```

```
export EDK2_DEV_IMAGE="ghcr.io/tianocore/containers/ubuntu-22-dev:latest"
export EDK2_USER_ARGS="-v \"${HOME}\" : \"${HOME}\" -e EDK2_DOCKER_USER_HOME=\"${HOME}\""
export EDK2_BUILD_ROOT="/build"
export EDK2_BUILDR00T_ARGS="-v \"${EDK2_BUILD_ROOT}\" : \"${EDK2_BUILD_ROOT}\""
alias edk2_docker="docker run -it --rm -w \"$(pwd)\" ${EDK2_BUILDR00T_ARGS} ${EDK2_USER_ARGS}
\"${EDK2_DEV_IMAGE}\""
```

```
edk2_docker echo hello
```

```
jetson@ubuntu: ~  
jetson@ubuntu:~$ export EDK2_BUILDROOT_ARGS="-v \"${EDK2_BUILD_ROOT}\"/:\"${EDK2_BUILD_ROOT}\"\"  
jetson@ubuntu:~$ alias edk2_docker="docker run -it --rm -w \"$(pwd)\" \"${EDK2_BUILDROOT_ARGS} ${EDK2_USER_ARGS} \"${EDK2_DEV_IMAGE}\"\"  
jetson@ubuntu:~$ edk2_docker echo hello  
Unable to find image 'ghcr.io/tianocore/containers/ubuntu-22-dev:latest' locally  
latest: Pulling from tianocore/containers/ubuntu-22-dev  
e6fdc8487bfe: Pull complete  
e5acbe445ea4: Pull complete  
f032eba38c52: Pull complete  
7d3217c69926: Pull complete  
792090d3d20d: Pull complete  
fa3210d58a0e: Pull complete  
31018c77db05: Pull complete  
67efaa724b88: Pull complete  
1a9ea5b856fa: Pull complete  
37f1b0ceb1ae: Pull complete  
832ec19c975b: Pull complete  
0f3cc2c8048e: Pull complete  
Digest: sha256:4fdb1fb41f33b28fb3ed438505663bbb38ffa494566b15538c4d710f2b61ba2e  
Status: Downloaded newer image for ghcr.io/tianocore/containers/ubuntu-22-dev:latest  
hello  
jetson@ubuntu:~$
```

|||||

||| edk2||

```
edk2_docker init_edkrepo_conf  
edk2_docker edkrepo manifest-repos add nvidia https://github.com/NVIDIA/edk2-edkrepo-  
manifest.git main nvidia
```

|| JetPack||||| uefi|| (|| **JetPack6.2.1/r36.4.4**||)

```
edk2_docker edkrepo clone nvidia-uefi-r36.4.4 NVIDIA-Platforms r36.4.4-updates
```

||||||| git|||||||

||||||| LOGO||

```
cd nvidia-uefi-r36.4.4/edk2-nvidia/Silicon/NVIDIA/Assets/
```

- 🕒 Recent
- ★ Starred
- 🏠 Home
- 🖨 Desktop
- 📄 Documents
- ⬇ Downloads
- 🎵 Music
- 🖼 Pictures
- 📺 Videos
- 🗑 Trash
- 📀 CDROM

▲
- 💿 Floppy Disk
- 📀 Ubuntu 20.0...

▲
- + Other Locations



LOGO

```
cd nvidia-uefi-r36.4.0/edk2-nvidia/Platform/NVIDIA/NVIDIA.fvmain.fdf.inc
```



```
jetson@ubuntu: ~/work/nvidia-uefi-r36.4.4
INFO - Copying DTB images/BootOrderHttp_Jetson_RELEASE.dtbo
INFO - Copying DTB images/BootOrderUsb_Jetson_RELEASE.dtbo
INFO - Copying DTB images/AcpiBoot_Jetson_RELEASE.dtbo
INFO - Copying DTB images/BootOrderUfs_Jetson_RELEASE.dtbo
INFO - Copying DTB images/BootOrderEmmc_Jetson_RELEASE.dtbo
INFO - Copying DTB images/DgpuDtEfifbSupport_Jetson_RELEASE.dtbo
INFO - Copying DTB images/BootOrderSD_Jetson_RELEASE.dtbo
INFO - Copying DTB images/BootOrderPxe_Jetson_RELEASE.dtbo
INFO - Copying DTB images/AndroidConfiguration_Jetson_RELEASE.dtbo
INFO - Copying DTB images/BootOrderSata_Jetson_RELEASE.dtbo
INFO - Copying DTB images/BootOrderNvme_Jetson_RELEASE.dtbo
INFO - Copying DTB images/L4TConfiguration_Jetson_RELEASE.dtbo
INFO - Copying DTB images/T234AcpiWinBoot_Jetson_RELEASE.dtbo
DEBUG - Plugin Success: Linux GCC5 Tool Chain Support
DEBUG - Plugin Success: Debug Macro Check Plugin
INFO - Writing BuildToolsReports to /home/jetson/work/nvidia-uefi-r36.4.4/Build/
Jetson/RELEASE_GCC5/BUILD_TOOLS_REPORT
DEBUG - Plugin Success: Build Tools Report Generator
PROGRESS - End time: 2025-10-14 09:16:05.452442 Total time Elapsed: 0:04:54
SECTION - Log file is located at: /home/jetson/work/nvidia-uefi-r36.4.4/Build/BU
ILDLOG_Jetson.txt
SECTION - Summary
PROGRESS - Success
jetson@ubuntu:~/work/nvidia-uefi-r36.4.4 [r36.4.4-updates]$
```

□□□□

□□□□