

# Hexgon llama.cpp on Qualcomm Linux

---

using qnn docker follower [Qairt compile host install](#) , you can pass the qairt python env install

---

## Download HEXAGON\_SDK\_6.6

using qpm-cli or go to the [Qualcomm Package Manager 3](#) to download

---

## Compiler LlamaCpp

init env, if you don't have add this into the bashrc, plz run this script.

```
source /usr/local/oe-core-x86_64/environment-setup-armv8-2a-qcom-linux
export QNN_SDK_ROOT=~/.qairt/2.46/
export LD_LIBRARY_PATH=$QNN_SDK_ROOT/lib/x86_64-linux-clang
export ADSP_LIBRARY_PATH=$QNN_SDK_ROOT/lib/hexagon-v73/unsigned
source ${QNN_SDK_ROOT}/qnn_env/bin/activate
source ${QNN_SDK_ROOT}/bin/envsetup.sh

export PATH=${QNN_SDK_ROOT}/bin/x86_64-linux-clang:$PATH
```

After init the env

```
git clone https://github.com/ggml-org/llama.cpp.git
cd llama.cpp
git checkout 3278e921b1e66f0b21f83cc993a1c66019f38b68

export HEXAGON_TOOLS_ROOT=/home/user/Hexagon_SDK_6.6.0/tools/HEXAGON_Tools/19.0.07/
export HEXAGON_SDK_ROOT=/home/user/Hexagon_SDK_6.6.0

cp docs/backend/snapdragon/CMakeUserPresets.json .
cmake --preset arm64-linux-snapdragon-release -B build-snapdragon

mkdir -p /home/user/Hexagon_SDK_6.6.0/ipc/fastrpc/qaic/bin && \
ln -sf /home/user/Hexagon_SDK_6.6.0/ipc/fastrpc/qaic/Ubuntu/qaic \
/home/user/Hexagon_SDK_6.6.0/ipc/fastrpc/qaic/bin/qaic && \
cd /home/user/llama.cpp && \
cmake --build build-snapdragon -j$(nproc)

cmake --install build-snapdragon --prefix pkg-snapdragon
zip -r pkg-snapdragon.zip pkg-snapdragon
```

---

## On target device (IQ9)

copy the pkg-snapdragon into the target device and unzip the file

```
#check the qcom-fastrpc1 has instal by apt
#if not found, plz apt install qcom-fastrpc1
```

```
unzip pkg-snapdragon
cd pkg-snapdragon
```

```
wget https://huggingface.co/Qwen/Qwen3.5-2B/resolve/main/chat_template.jinja
mv chat_template.jinja qwen3.5.jinja
wget https://huggingface.co/unsloth/Qwen3.5-2B-GGUF/resolve/main/Qwen3.5-2B-Q4_0.gguf?download=true
```

```
mv Qwen3.5-2B-Q4_0.gguf\?download=true Qwen3.5-2B-Q4_0.gguf
```

```
export LD_LIBRARY_PATH=./lib
export ADSP_LIBRARY_PATH=./lib
```

```
./bin/llama-server -m Qwen3.5-2B-Q4_0.gguf --device HTP0 --reasoning auto -c 64000 --host 10.91.215.9 --port 8080 --jinja --chat-template-file qwen3.5.jinja --fit off --temp 0 --top-k 0 -tb 8 -t 6 -np 1 --kv-unified
```

## Qairt compile host install

---

# docker image install

```
#FROM docker-registry.qualcomm.com/library/ubuntu:22.04
FROM ubuntu:22.04

ENV DEBIAN_FRONTEND=noninteractive \
    PIP_NO_CACHE_DIR=1 \
    PIP_DISABLE_PIP_VERSION_CHECK=1 \
    LC_ALL=C.UTF-8 \
    LANG=C.UTF-8

RUN apt-get update \
    && apt-get install -y --no-install-recommends \
        # --- base utilities ---
        ca-certificates \
        curl \
        sudo \
        bash-completion \
        vim \
        xdg-utils \
        tmux \
        wget \
        unzip \
        tar \
        git \
        file \
        clang \
        # --- python ---
        python3.10 \
        python3.10-venv \
        python3-pip \
        python3-distutils \
        # --- build tools ---
        cmake \
        diffstat \
        gawk \
        cpio \
        pkg-config \
        build-essential \
        gcc \
        g++ \
        openssh-client \
        # --- aarch64 cross-compiler ---
        gcc-aarch64-linux-gnu \
        g++-aarch64-linux-gnu \
        # --- ffmpeg / media libs ---
        libavformat-dev \
        libavcodec-dev \
        libavdevice-dev \
        libavutil-dev \
        libavfilter-dev \
        libswscale-dev \
        libswresample-dev \
    && rm -rf /var/lib/apt/lists/*

RUN update-alternatives --install /usr/bin/python python /usr/bin/python3.10 10 && \
    update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.10 10 && \
    update-alternatives --set python /usr/bin/python3.10 && \
    update-alternatives --set python3 /usr/bin/python3.10 && \
    python --version && python3 --version

# UID=1000, GID=1000
RUN groupadd -g 1000 user \
    && useradd -m -u 1000 -g 1000 -s /bin/bash user \
    && usermod -aG sudo user \
    && echo 'user:123' | chpasswd \
    && sed -i 's@^%sudo.*@%sudo ALL=(ALL) NOPASSWD: ALL@' /etc/sudoers
```

```
RUN mkdir /home/user/qairt
```

```
WORKDIR /home/user
```

```
USER user
```

```
docker build -t your_image_name
```

---

## Go into the Docker container

### toolchain install

for qnn-model-lib-generator you need to using the -t xxx, plz follow this

<https://artifacts.codelinaro.org/ui/native/qli-ci/flashable-binaries/qimpsdk/qcs9075-ig-9075-evk/>

JFrog [x86-qcom-6.6.90-QLI.1.5-Ver.1.1\\_qim-product-sdk-2.0.1.zip](#)

Verify : QLI1.5

after compress,

```
sudo bash target\qcs9075-ig-9075-evk\sdk\qcom-wayland-x86_64-qcom-multimedia-image-armv8-2a-qcs9075-ig-9075-evk-toolchain-1.5-ver.1.1.sh
```

setting the path to

```
/home/<user_name>/qcom-wayland_sdk/environment-setup-armv8-2a-qcom-linux
```

---

## Download qairt sdk

go to qpm to download and put into the /home/user/qairt , rename to x.yy  
qpm-cli --license-activate Qualcomm\_AI\_Runtime\_SDK

## qairt-python env

```
cd /home/user/qairt
python3.10 -m venv qairt_env
source /home/user/qairt/qairt_env/bin/activate
pip install --upgrade pip
```

need to extral download and put the piqaro-1.0.3-cp310-cp310-linux\_x86\_64.whl in /home/user/qairt

```
piqaro @ file:///home/user/qairt/piqaro-1.0.3-cp310-cp310-
linux_x86_64.whl#sha256=a33fecb6c0e6e70f1a74b4d34f5fc1253e8dabced5c4b622bebc7cd5c52be15f
```

copy the requestment.txt and using the `pip install -r requestment.txt`

absl-py==2.1.0  
aenum==3.1.15  
aiohappyeyeballs==2.6.1  
aiohttp==3.13.3  
aiosignal==1.4.0  
alembic==1.18.4  
annotated-doc==0.0.4  
annotated-types==0.7.0  
ansi2html==1.9.2  
anyio==4.12.1  
argon2-cffi==25.1.0  
argon2-cffi-bindings==25.1.0  
arrow==1.4.0  
asttokens==3.0.1  
astunparse==1.6.3  
async-lru==2.2.0  
async-timeout==5.0.1  
attrs==23.2.0  
babel==2.18.0  
backoff==2.2.1  
bcrypt==5.0.0  
beautifulsoup4==4.14.3  
bidict==0.23.1  
bleach==6.3.0  
botocore==1.42.46  
certifi==2026.1.4  
cffi==2.0.0  
chardet==5.2.0  
charset-normalizer==3.4.4  
click==8.3.1  
cloudpickle==3.1.2  
cmaes==0.12.0  
colorama==0.4.6  
coloredlogs==15.0.1  
colorlog==6.10.1  
comm==0.2.3  
concurrent-log-handler==0.9.28  
contourpy==1.3.2  
cryptography==43.0.3  
cyclr==0.12.1  
dash==2.12.1  
dash-core-components==2.0.0  
dash-html-components==2.0.0  
dash-table==5.0.0  
debugpy==1.8.20  
decorator==4.4.2  
defusedxml==0.7.1  
deprecation==2.1.0  
dm-tree==0.1.9  
exceptiongroup==1.3.1  
executing==2.2.1  
fastapi==0.128.7  
fastjsonschema==2.21.2  
filelock==3.20.3  
Flask==2.2.5  
flatbuffers==25.12.19  
fonttools==4.61.1  
fqdn==1.5.1  
frozenlist==1.8.0  
fsspec==2026.2.0  
gast==0.7.0  
google-api-core==2.30.3  
google-api-python-client==2.196.0  
google-auth==2.53.0  
google-auth-httpplib2==0.4.0  
google-pasta==0.2.0  
googleapis-common-protos==1.75.0  
greenlet==3.3.1  
grpcio==1.78.0

h11==0.16.0  
h5py==3.15.1  
httpcore==1.0.9  
httplib2==0.31.2  
httpx==0.28.1  
humanfriendly==10.0  
idna==3.11  
iniconfig==2.3.0  
invoke==1.7.3  
ipykernel==7.2.0  
ipython==8.38.0  
isoduration==20.11.0  
itsdangerous==2.2.0  
jedi==0.19.2  
Jinja2==3.1.6  
jmespath==1.1.0  
joblib==1.4.0  
json5==0.13.0  
jsonpointer==3.0.0  
jsonschema==4.19.0  
jsonschema-specifications==2025.9.1  
jupyter-events==0.12.0  
jupyter-lsp==2.3.0  
jupyter\_client==8.8.0  
jupyter\_core==5.9.1  
jupyter\_server==2.17.0  
jupyter\_server\_terminals==0.5.4  
jupyterlab==4.5.6  
jupyterlab\_pygments==0.3.0  
jupyterlab\_server==2.28.0  
keras==3.12.1  
kiwisolver==1.4.9  
libclang==18.1.1  
lightning-utilities==0.15.3  
lxml==5.2.1  
Mako==1.1.0  
Markdown==3.10.2  
markdown-it-py==4.0.0  
MarkupSafe==3.0.3  
matplotlib==3.10.8  
matplotlib-inline==0.2.1  
mdurl==0.1.2  
mistune==3.2.0  
ml\_dtypes==0.5.4  
mock==3.0.5  
mpmath==1.3.0  
multidict==6.7.1  
namex==0.1.0  
narwhals==2.16.0  
nbclient==0.10.4  
nbconvert==7.17.0  
nbformat==5.10.4  
nest-asyncio==1.6.0  
networkx==3.4.2  
notebook==7.5.5  
notebook\_shim==0.2.4  
numpy==1.26.4  
nvidia-cublas-cu12==12.4.5.8  
nvidia-cuda-cupti-cu12==12.4.127  
nvidia-cuda-nvrtc-cu12==12.4.127  
nvidia-cuda-runtime-cu12==12.4.127  
nvidia-cudnn-cu12==9.1.0.70  
nvidia-cufft-cu12==11.2.1.3  
nvidia-cufile-cu12==1.13.1.3  
nvidia-curand-cu12==10.3.5.147  
nvidia-cusolver-cu12==11.6.1.9  
nvidia-cuspars-cu12==12.3.1.170  
nvidia-cusparselt-cu12==0.7.1  
nvidia-nccl-cu12==2.21.5

nvidia-nvjitlink-cu12==12.4.127  
nvidia-nvshmem-cu12==3.3.20  
nvidia-nvtx-cu12==12.4.127  
oauth2client==4.1.3  
onnx==1.16.1  
onnx-tf==1.10.0  
onnx2torch==1.5.15  
onnxoptimizer==0.4.2  
onnxruntime==1.20.0  
onnxsim==0.4.36  
opencv-python==4.6.0.66  
opt\_einsum==3.4.0  
optree==0.18.0  
optuna==3.3.0  
overrides==7.7.0  
packaging==24.0  
pandas==2.3.3  
pandocfilters==1.5.1  
paramiko==3.5.1  
parso==0.8.6  
pathlib2==2.3.6  
pexpect==4.9.0  
pillow==10.2.0  
piqaro @ file:///home/user/qairt/piqaro-1.0.3-cp310-cp310-  
linux\_x86\_64.whl#sha256=a33fecb6c0e6e70f1a74b4d34f5fc1253e8dabced5c4b622bebc7cd5c52be15f  
platformdirs==4.9.4  
plotly==5.20.0  
pluggy==1.6.0  
polars==1.38.1  
polars-runtime-32==1.38.1  
portalocker==3.2.0  
prettytable==3.17.0  
prometheus\_client==0.24.1  
prompt\_toolkit==3.0.52  
propcache==0.4.1  
proto-plus==1.28.0  
protobuf==6.31.1  
psutil==6.1.1  
ptyprocess==0.7.0  
pure\_eval==0.2.3  
pyaml==26.2.1  
pyasn1==0.6.3  
pyasn1\_modules==0.4.2  
pycparser==3.0  
pydantic==2.8.2  
pydantic\_core==2.20.1  
PyDrive2==1.21.3  
Pygments==2.19.2  
PyNaCl==1.6.2  
pyOpenSSL==24.2.1  
pyparsing==3.3.2  
pytest==8.1.1  
python-dateutil==2.9.0.post0  
python-engineio==4.13.1  
python-json-logger==4.0.0  
python-socketio==5.16.1  
pytz==2025.2  
PyYAML==6.0.3  
pyzmq==26.4.0  
qai-hub==0.44.0  
qairt-visualizer==0.8.0  
referencing==0.37.0  
requests==2.32.5  
requests-toolbelt==1.0.0  
retrying==1.4.2  
rfc3339-validator==0.1.4  
rfc3986-validator==0.1.1  
rich==13.9.4  
rpds-py==0.30.0

```
rsa==4.9.1
s3transfer==0.13.1
safetensors==0.4.3
scikit-learn==1.7.2
scikit-optimize==0.9.0
scipy==1.15.3
semver==3.0.4
Send2Trash==2.1.0
sentencepiece==0.2.1
simple-websocket==1.1.0
six==1.16.0
soupsieve==2.8.3
SQLAlchemy==2.0.46
stack-data==0.6.3
starlette==0.52.1
sympy==1.13.1
tabulate==0.9.0
tenacity==9.1.4
tensorboard==2.20.0
tensorboard-data-server==0.7.2
tensorflow==2.20.0
tensorflow-addons==0.23.0
tensorflow-probability==0.25.0
termcolor==3.3.0
terminado==0.18.1
tf_keras==2.20.1
tflite==2.18.0
threadpoolctl==3.6.0
tinycss2==1.4.0
tomli==2.4.0
torch==2.5.1
torchmetrics==1.9.0
torchvision==0.20.1
tornado==6.5.5
tqdm==4.67.3
traitlets==5.14.3
triton==3.1.0
typeguard==2.13.3
typing-inspection==0.4.2
typing_extensions==4.14.0
tzdata==2025.3
ultralalytics==8.4.14
ultralalytics-thop==2.0.18
uri-template==1.3.0
uritemplate==4.2.0
urllib3==2.6.3
uvicorn==0.40.0
wcwidth==0.6.0
webcolors==25.10.0
webencodings==0.5.1
websocket-client==1.9.0
Werkzeug==2.2.3
wrapt==2.1.1
wsproto==1.3.2
XlsxWriter==1.2.2
yarl==1.22.0
```

## install qairt linux package

```
sudo bash ~/qairt/2.46/bin/check-linux-dependency.sh
```

---



# Init the env

the qairt version you can change : QNN\_SDK\_ROOT=~/.qairt/2.46/

the hexagon version you can change : \$QNN\_SDK\_ROOT/lib/hexagon-v73/unsigned

```
source /usr/local/oe-core-x86_64/environment-setup-armv8-2a-qcom-linux
export QNN_SDK_ROOT=~/.qairt/2.46/
export LD_LIBRARY_PATH=$QNN_SDK_ROOT/lib/x86_64-linux-clang
export ADSP_LIBRARY_PATH=$QNN_SDK_ROOT/lib/hexagon-v73/unsigned
source ${QNN_SDK_ROOT}/qnn_env/bin/activate
source ${QNN_SDK_ROOT}/bin/envsetup.sh

export PATH=${QNN_SDK_ROOT}/bin/x86_64-linux-clang:$PATH
```

---

[← 返回 : Hexgon llama.cpp on Qualcomm Linux](#)