Qualcomm RB5 Robotics Development Platform

Introduction

SOM Overview

Carrier Boards
Qualcomm RB5 Development Kit

Firmware Development
Downloading SDK, Installing SDK, Generating Image and Flashing

Modes in RB5
EDL mode
FastBoot mode

Accessing the Board
- Using ADB
- Using Serial Console
- Using SSH

Device Setup
- WiFi Setup
- HDMI Setup

Neural Processing SDK

YOLOV5 Conversion to DLC

Introduction

The Qualcomm® Robotics RB5 development kit is a platform bringing together Qualcomm Technologies broad expertise in 5G and AI to empower developers and manufacturers to create the next generation of high-compute, low-power robots and drones for the consumer, enterprise, industrial and professional service sectors – and the comprehensive Qualcomm Robotics RB5 Development Kit helps ensure that developers have the customization and flexibility they need to make their visions a commercial reality. With 4G and 5G connectivity speeds via a companion module, the Qualcomm Robotics RB5 platform helps pave the way for the proliferation of 5G in robotics and intelligent systems.

The Qualcomm Robotics RB5 Platform supports the development of smart, power-efficient and cost-effective robots by combining high-performance heterogeneous compute, Qualcomm® Artificial Intelligence (AI) Engine for on-device machine learning, computer vision, vault-like security, multimedia, Wi-Fi and cellular connectivity solutions to help solve common robotics challenges.

- Contains advanced robotics platform Qualcomm® QRB5165 processor.
- Supports widely used Linux based distributions for robotics applications.
- Supports multiple SDKs and tools, including Qualcomm® Neural Processing SDK for AI, Qualcomm® Robotics Vision SDK, Qualcomm® Computer Vision SDK, Qualcomm® Hexagon™ DSP SDK, Robotics Operating System (ROS) 2, and multiple Linux distributions.
- Comprehensive set of demo applications and tutorials to accelerate development of robotics applications.
- Compliance with the 96Board, support for sensors such as multiple cameras, depth sensing solution, GMSL sensor, Ultrasonic Time-of-Flight Sensor with Extended Range, multi-mic and additional sensors like IMU, pressure sensor, magnetometer etc.
- Multiple interfaces and I/Os which can connect multiple sensors.

System on Module (SoM):

The SoM serves as the core computing hardware, encompassing CPU cores, GPU, hardware accelerators, memory, and interfaces for seamless communication—all integrated onto a single printed circuit board (PCB).

Carrier Board:

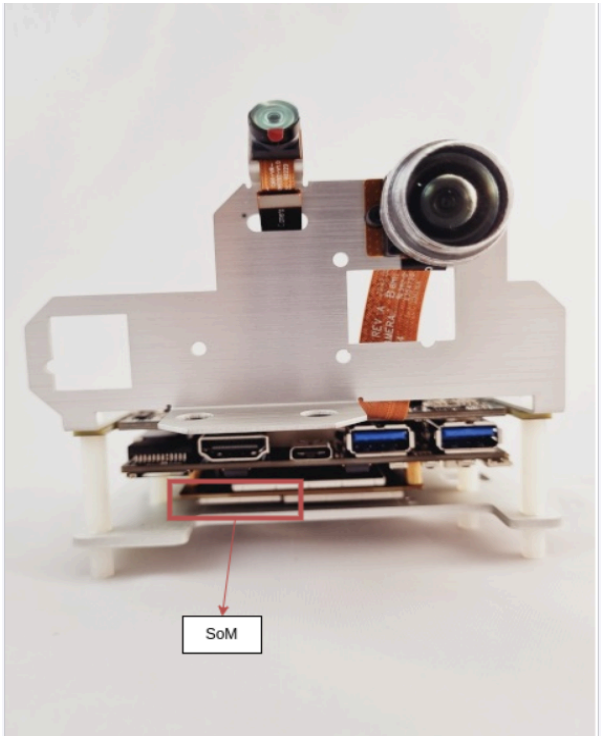
Acting as the physical interface for the SoM, the carrier board can either be a developer kit or a custom-designed board. During the prototyping phase, a developer kit proves invaluable for evaluating the SoM's capabilities. Subsequently, for specific use cases, transitioning to a custom carrier board tailored to unique requirements is common practice. The SoM is housed on the carrier board, and the official carrier board for RB5, known as the mainboard, supports the attachment of various mezzanines.

Reference Links: Thundercomm , Qualcomm

SOM Overview

aThe System-on-Module (SoM) for Qualcomm Robotics RB5 is a printed circuit board (PCB) housing critical components such as memory, power management IC, communication interfaces, and the robust heterogeneous computing architecture, featuring the QRB5165 SoC. This SoC integrates the CPU, GPU, DSP AI engine, ISP, Neural Processing Unit (NPU), and a Secure Processing Unit for enhanced security.

Figure 1. shows a picture of the Robotics RB5 SoM.



Graphics Processing Unit (GPU)

The Qualcomm® Adreno™ 650 GPU has the following characteristics:

- **Supports the following APIs:**
  - **For graphics: OpenGL ES 3.2, Vulkan and VulkanCompute, DX12.x**
  - **For computing: OpenCL 2.0 full profile, DirectCompute**

Central Processing Unit (CPU)

The Qualcomm® Kryo™ 585 CPU's cores have the following characteristics:

- **4 high-performance Kryo Gold cores**
- **4 low-power Kryo Silver cores**

Digital Signal Processor (DSP)

The Hexagon DSP is more efficient in power consumption than the CPU. It is conformed by several DSPs such as the compute DSP (cDSP) which is preferable for compute-intensive tasks such as, video enhancements, virtual reality applications and camera streaming; it also contains an audio DSP (aDSP) and sensor DSP. The Hexagon DSP comes with an Hexagon Tensor Accelerator (15 TOPS) for fixed-point deep convolutional neural networks and

Hexagon Scalar Accelerator to support AI processing and computer vision (CV).

Neural Processing Unit (NPU)

The NPU 230 for neural networks processing in high-performance machine learning. Used along with the Hexagon DSP.

Image Signal Processor (ISP)

The Qualcomm® Spectra 480 ISP can process 2 Gigapixels per second with the quality of 200 megapixel photos, record video at 8K and 4K HDR video capture. Has the following characteristics:

- **Real-time sensor input resolution: Up to two 25MP and five 2MP cameras**
- **64 MP 30 fps ZSL with a dual ISP**

Secure processing Unit (SPU)

The SPU 240 comes with several features such as the following:

- **Camera Security**
- **Crypto Engine**
- **Cryptographic Accelerator**
- **Qualcomm Trusted Execution Environment**
- **Secure Boot & Qualcomm® Crypto Engine Core is FIPS 140-2 certified**

Feature	Qualcomm Robotics RB5/RB6
GPU	Adreno GPU 650
CPU	8-core Qualcomm® Kryo™ 585 built on Arm® Cortex® technology, 64-bit CPU
CPU Max Freq	2.842 GHz
Memory	Four-channel PoP high-speed memory – 8 GB LPDDR5 or LPDDR4X SDRAM (4 × 16-bit) designed for a 2750 MHz (LPDDR5) or 2133 MHz (LPDDR4X) clock and system cache. Memory density up to 16GB
Storage	UFS 3.1 gear 4 2 lanes, 128GB
Wireless Connectivity	2x WLAN 2 802.11ax with DBS and Bluetooth® 5.1
Digital Signal Processing	Compute Hexagon DSP v66 at 1.5 GHz with 4x Hexagon Vector eXtensions with an Hexagon Tensor Accelerator delivering 15 TOPS of AI performance (70-100 TOPS in RB6).
Neural Processing Unit	NPU230 for high-performance machine learning
Image Signal Processing	Qualcomm Spectra 480 ISP to support up to 12 cameras by D-PHY and 18 cameras by C-PHY (7 concurrent). It offers six CSI 4-lane interfaces with either D-PHY or C-PHY:
Secure Processing Unit	SPU 240

Carrier Boards

The Qualcomm Robotics RB5 Developer Kit is compliant with the 96Boards open hardware specification to support mezzanine-board expansions, and it supports Linux, Ubuntu, ROS (Robot Operating System) 2 and AWS RoboMaker. The kit comes in two versions, the Core kit and the Vision Kit.

Qualcomm Robotics RB5 Vision Kit includes:

- The mainboard with the RB5 SoM
- Wall power adapter.
- USB Type-C cable
- Vision mezzanine
- Inbuilt Tracking and main camera

The Vision mezzanine board includes the following components:

Camera

1x Main camera sensor IMX577

Interface: MIPI

Resolution/fps: 4056x3040/60fps

1x Tracking camera sensor OV9282

Interface: MIPI

Resolution/fps: 1280x800/120fps or 640x480/180fps

2x GMSL2 camera inputs

Datasheet Link : Link

Firmware Development

Register into Thundercomm at Thundercomm's official page. This step is necessary to use the SDK Manager. Please check figure 1 and 2 for a guide on where to register.

We are using

OS: Ubuntu 20.04.6 LTS

Release: 20.04

Codename: focal

HENCE FOLLOWING STEPS IN MANUAL README_Linux.txt

STEP 1: For Docker users on any Ubuntu Desktop (18.04, 16.04, etc.), please follow the guide in "On Other Ubuntu Host" for all sections.

STEP 2: Enter a work directory with read-write permissions for sdkmanager. For Docker image users, create the work directory under /home/hostPC/.

STEP 3: If you are using a Linux desktop to flash the device, please run the following command before connecting the device to the desktop - $ sudo systemctl stop ModemManager

STEP 4: Installing packages :

For Robotics RB5 LU2.0 and RB5N LU2.0 Platform (required OS: Ubuntu 20.04)#

Required packages (minimum version): coreutils 8.30, fakechroot 2.19, fakeroot 1.24, kmod 27-1ubuntu2.1, libc6-arm64-cross 2.31, python 2.7.18, qemu-user-static 1:7.2+dfsg-5ubuntu1, udev 245.4-4ubuntu3.20, unzip 6.0, wget 1.20.3.

Run these commands to create soft links:

$ sudo rm -rf /lib/ld-linux-aarch64.so.1

$ sudo ln -sf /usr/aarch64-linux-gnu/lib/ld-2.31.so /lib/ld-linux-aarch64.so.1

$ sudo ln -sf /bin/bash /bin/sh

$ sudo dpkg -P qemu-user-static

$ wget http://archive.ubuntu.com/ubuntu/pool/universe/q/qemu/qemu-userstatic_6.2+dfsg-2ubuntu6_amd64.deb

$ sudo dpkg -i qemu-user-static_6.2+dfsg-2ubuntu6_amd64.deb

STEP 5: Install the dependency libraries to the host computer

$ sudo apt-get install coreutils fakechroot fakeroot kmod libc6-arm64-cross python2.7 qemu-user-static wget udev openssh-server -y

Generate Ubuntu Docker Image

NOTE: Different OS versions require different Docker images.

#Robotics RB5 LU2.0 and RB5N LU2.0 Platform#

For an Ubuntu 16.04 or 18.04 host, an Ubuntu 20.04 Docker image is required.

#RB5LU1.0, RB6, RB1, and RB2 Platforms#

For an Ubuntu 16.04 or 20.04 host, an Ubuntu 18.04 Docker image is required.

STEP 6: Install qemu-user-static, openssh-server and udev to the host PC.

$sudo apt-get install qemu-user-static openssh-server udev -y

STEP 7: # Generate Ubuntu 18.04 docker image {since we are using 20.04 ubuntu)

$ ln -sf Dockerfile_18.04 Dockerfile

$ sudo docker build -t ubuntu:18.04-sdkmanager .

STEP 8: On Ubuntu Docker

1. Create Docker container

Ubuntu 18.04 docker image

$ sudo docker run -v /home/${USER}:/home/hostPC/ --privileged -v /dev/:/dev -v /run/udev:/run/udev -d --name sdkmanager_container -p 36000:22 ubuntu:18.04-sdkmanager

STEP 9: 2. Launch sdk manager in Docker container

$ sudo docker exec -it sdkmanager_container sdkmanager

STEP 10: General

1. Provide Thudercomm login credentials

Credential Checking ...

Enter your Thundercomm username:

Enter your Thundercomm password:

Provide a working directory when asked for target directory (for example "/home/user")

Note: For Docker users, provide the work directory as /home/hostPC/[working directory]

STEP 11: Select a desired product

Select your product:

1: RB1

2: RB2

3: RB5

4: RB5N(Non-Pop)

5: RB6

Select one number of product(1 | 2 | 3 ...) to continue with:

STEP 12: Select a desired platform for Robotics RBx device

Choose a platform for Robotics RBx device

Enter 1 to use LU platform, 2 to use LE platform:

STEP 13: LU PLATFORM ##

1. Type the number of available version for image repack, for example, "1"

You chose LU platform

Enter 1 to use LU1.0 platform, 2 to use LU2.0 platform:

2. Type the number of available version for image repack, for example, "1"

You chose LU1.0 platform

Checking the current version of release ...

Available versions:

1: QRB5165.UBUN.x.x-xxxxxx (latest version)

Recommended version: QRB5165.UBUN.x.x-xxxxxx

Select one number of available version (1) to continue with:

STEP 14: Type "1" when below message was shown to start downloading LU resources and image repacking, or type "help" for more information

---

SDK has been successfully set up and is ready to be used

Type 'help' for commands

> help

commands:

help = Show usage help for LU platform

1 = Download LU resources and generate system.img with current release

2 = Flash full build (require system.img generation first)

q = exit sdk manager

You will see the following message once the images are built successfully

---

Move sparse images to full_build ...done

You may proceed to flash full_build to your device

NOTE: The repack process might take up to 40 minutes.

########## FLASH RBx DEVICE ##########

On Ubuntu Host and On Ubuntu Docker

1. Disconnect the device from PC before flashing full build, and follow the operation steps below:

1) Power off the device (unplug power cable and USB cable)

2) Plug in the power on device (12V)

3) Press and hold F_DL key, and connect board to PC via a type-C USB (It will switch the device to EDL mode)

4) Release F_DL key after the board is connected to PC

5) Start flashing process from the SDK manager (command 2: "Flash full build")

6) SDK manager will detect the device and start flash process automatically

7) Wait for the flashing process to be finished, the board will reboot automatically.

2. Enter the command below in a new terminal window on Host PC after the device successfully boots up:

$ adb wait-for-device shell

Modes in RB5

Connecting the board in USB flashing mode (aka EDL mode)

Steps to set up the board in the EDL Mode. EDL Mode is used for recovering/flashing the board over USB.

1. Power off the board and unplug the USB type-C cable.

2. Set the switch DIP_SW_0 to “0 1 1 0 0 0”, please check figure 1 and 2. This configuration means that the board debug UART is enabled and that the board will power on automatically when main power is asserted.

3. Press button 18, it is marked with "F_DL". This button is located on the back of the board. Please check figure 3 and 4.

4. While pressing the F_DL button, power on the device with the proper power supply. Do NOT stop pressing the button.

5. While pressing the F_DL button, plug in the USB type-C cable between the board and the host computer. Stop pressing the F_DL button.

6. To check if everything went well, you can check the USB connections with dmesg.

dmesg | grep usb

It will show something like:

[281089.835630] usb 2-1: Qualcomm USB modem converter now attached to ttyUSB0

Fastboot mode

Set up a board for it to be ready to be flashed. For this, we will show the steps to boot your RB5/RB6 into fastboot mode.

1. With the board powered off (unplugged from power), set the BOOT_CONFIG DIP Switch in the board to '0-0-0-0'; in other words, every switch should be in "OFF" position. Please check Figure 1 for reference on how to configure the DIP SWITCH.

- Disconnect the power cable from the board and check no USB cable is connected either.
- Hold down the "VOL-" button while connecting the power supply.
- Tap the "ON/OFF" button while continuing to hold the "VOL-" button, at least 5 seconds after plugging the power cable.
- Release "VOL-" button.
- Connect the USB3 Type C cable between the host PC and the RB5/RB6 board.

2. Check if the board is connected and in fastboot mode by opening a terminal in the host computer and running the following command:

user@desktop:~/work/QRB5_Exploration/Linaro$ sudo fastboot devices

3b73cc0 fastboot

You can also check the USB connections with dmesg, using the following command:

user@desktop:~/work/QRB5_Exploration/Linaro$ dmesg | grep -i usb

[281714.226579] usb 2-1: Product: Android

[281714.226582] usb 2-1: Manufacturer: Google

[281714.226585] usb 2-1: SerialNumber: 3b73cc0

Accessing the Board

Through ADB

ADB (Android Debug Bridge) is a command-line tool for communicating to a device with ADB enabled. The tool works as client/server communication, where the client can send commands to a device, the server is in charge of connecting to the device and a Daemon that actually executes the commands sent by the client. ADB allows actions such as file sharing between computer and device and running shell commands on the device.

1.Install adb tools

sudo apt install android-tools-fastboot -y

2. Connect the USB type-C cable to the board, and the other end to your host PC.Connect the power cable to the board.

3. In your host PC type the following command to check if the device is registered:

adb devices

It should look something like the following when the device is registered:

$ adb devices

List of devices attached

3b73cc0 device

adb root

4. Once the device is registered type the following command to enter the board:

adb shell

Using Serial Console

UART (Universal Asynchronous Receiver/Transmitter) is a circuit in charge of transmitting and receiving information in serial form. The communication is done exclusively between two devices which typically handle data in parallel form, so the UART does the conversion from one to the other.

1. Make sure the board is powered off and disconnected. Turn on the SW2 on the DIP Switch 0 to enable the USB debug and turn on SW3 to enable auto power up,

the rest of the switches can be in any position you prefer. See Figure 1 to locate the switches.

2. Connect the microUSB cable to the USB debug port, see Figure 2 for its location on the board, and the other end to a USB port on your host PC.
   Connect the power supply to the power connector on the board and the other end to the power outlet. The board will boot up in around 5 seconds. Once it finishes the boot process, a device file with the name ttyUSB0 in the /dev directory.
   Open a serial console tool in your host PC. In this case, Minicom will be used, the parameters for the serial communication to access the board are the following:
   a. Baud rate: 115200
   b. Data bits: 8
   c. Stop bits: 1
   d. Parity: None
   e. Flow control: None
3. Open a terminal in your host PC and type:
4. minicom -D /dev/ttyUSB0 -8 -b 11520
5. You will be prompted to insert the credentials to access the board, see Figure 3. The default credentials for Thundercomm official images are:

   User: root

   Password: oelinux123

Using SSH

SSH (Secure Socket Shell) is a network protocol that provides secure access, by authentication and data encryption during transmission, between two computers in an open network The protocol allows navigation on a remote computer's filesystem, access to resources and file transferring.

1.To access the board with ADB with your host PC and already have an internet connection in the board, to install ADB follow the step in Wifi Setup or Ethernet Connection utility must be installed

2. In your host PC, type the following command to access the board:

```
adb shell
# apt-get install openssh-server
```

3. Type the following commands to enable SSH access to the board:

```
# echo  "PermitRootLogin yes"  >> /etc/ssh/sshd_config
# service ssh restart
```

Accessing the board with SSH

1. First, make sure both the board and the computer are connected to the same local network.

2. Connect over Serial console or adb

3. Once connected to the board, write the following command to check the IP address of the board:

ifconfig eth0

4. SSH into the board with the following command, where <ipaddress>

ssh root@<ipaddress>

Device Setup

Wi-Fi Setup

Access the board with ADB

1. In your host PC, create a file named `wpa_supplicant.conf` with the following content, and replace the text inside the "<>" symbols with your information:

```
network={
  ssid="<your Wi-Fi network>"
  proto=WPA2
  key_mgmt=WPA-PSK
  pairwise=TKIP CCMP
  group=TKIP CCMP
  psk="<your Wi-Fi password>"
}
```

2. Using ADB, transfer the file into the board :

```
adb shell mount -o remount,rw /
adb push wpa_supplicant.conf /data/misc/wifi/
adb shell sync
```

3. Reboot the board with the following command so that the WiFi connection gets initialized:

```
adb reboot
```

4. Once the device is up again, after a few seconds, verify that the WiFi is up in the wlan0 interface:

```
adb shell
# ifconfig wlan0
```

If there is no internet connection and the ip address is not assigned, use the following command:

```
# wpa_supplicant -Dnl80211 -iwlan0 -c /data/misc/wifi/wpa_supplicant.conf -dddd &
```

HDMI Setup

Connect an HDMI cable to the board's HDMI port and the other end to a HDMI monitor.
2. In your host PC, type the following command to access the board:

```
adb shell
```

3.Type the following commands to enable HDMI display:

```
# mkdir -p /usr/bin/weston_socket
# export XDG_RUNTIME_DIR=/usr/bin/weston_socket
# export LD_LIBRARY_PATH=/usr/lib:/usr/lib/aarch64-linux-gnu/
# weston --tty=1 --connector=29 --idle-time=0
```

You should now see the weston display on your monitor.

Neural Processing SDK

The Neural Processing SDK for Artificial Intelligence provides a set of tools to work on Machine Learning/AI Applications on the RB5.

PREREQUISITES

The Neural Processing SDK runs on the host computer, not in the Robotics RB5 platform. The host computer must have:

- Required Operating System (OS): Ubuntu Linux 18.04.
- Python: Version 3.6
- One of the following frameworks: Caffe and Caffe2, TensorFlow, ONNX, PyTorch, or TensorFlow Lite.

First, install and check the requirements needed to work with the Neural Processing SDK.Install Python 3.6 Later, Install frameworks we are going to see how to install each of the supported frameworks to develop ML models. Finally, Get the Neural Processing SDK & download the Neural Processing SDK.

Install Python3.6 (DEFAULT)

1. Run the following command to check the version of Python 3:

```
$ python3 --version
Python 3.6.9
```

*If that is not your case, please continue with step 2.Do **NOT** delete any of the versions of Python already installed in your Ubuntu host computer for any reason. This could break your Ubuntu install.*

2. Run the next commands in your host computer to download and install Python 3.6

```
sudo apt-get update
sudo apt-get install python3.6
```

You can check if the installation was successful with the commands from step 1.

Make Python3.6 default version

Python version 3.6 should be the one pointed when you run commands like `python` *or check* `/usr/bin/python`*. To achieve this, you can use the following steps:*

1*. Create a list of alternatives for Python with the following commands:*

```
sudo update-alternatives --install /usr/bin/python python
/usr/bin/python2.7 1
sudo update-alternatives --install /usr/bin/python python
/usr/bin/python3.6 2
```

2*. Check the list you just created with the next command:*

```
update-alternatives --list python
```

3*. Select the Python version 3.6:*

```
sudo update-alternatives --config python
```

Install packages

We need some Ubuntu and Python packages to work with the Neural Processing SDK, everyone of them is listed in the command below.

1. Run the following command to install the SDK's dependencies.

```
sudo apt-get install python3-dev python3-matplotlib python3-numpy
python3-protobuf python3-scipy python3-skimage python3-sphinx wget zip
```

Install frameworks

As mentioned, you need one of the following frameworks to work with the Neural Processing SDK: Caffe and Caffe2, TensorFlow, ONNX, PyTorch, or TensorFlow Lite. In this section, we are showing you how to install any of them.

Caffe and Caffe2

For Caffe: `sudo apt install caffe-cpu`

```
caffe --version
```

TensorFlow

To install TensorFlow: *$* `python3.6 -m pip install tensorflow==2.3`

$ python3.6 -m pip show tensorflow

ONNX

To install ONNX: $ python3.6 -m pip install onnx

$ python3.6 -m pip show onnx

PyTorch

To install PyTorch: *$ python3.6 -m pip install torch torchvision torchaudio --extra-index-url https://download.pytorch.org/whl/cpu*

$ python3.6 -m pip show torch

TensorFlow Lite

To install TensorFlow Lite: *$ python3.6 -m pip install tflite==2.3.0*

$ python3.6 -m pip show tflite

Get the Neural Processing SDK

1. Download the SDK's Linux version on your host computer.

The Xs are the version number of the SDK we downloaded.

2. After the download is complete, a zip file with a name similar to `snpe-<version>.zip is visible` . To unzip it, run the following command:

`unzip -X snpe-<version>.zip`

Eg: If you downloaded version 2.05.0: `unzip -X snpe-2.5.0.4052.zip`

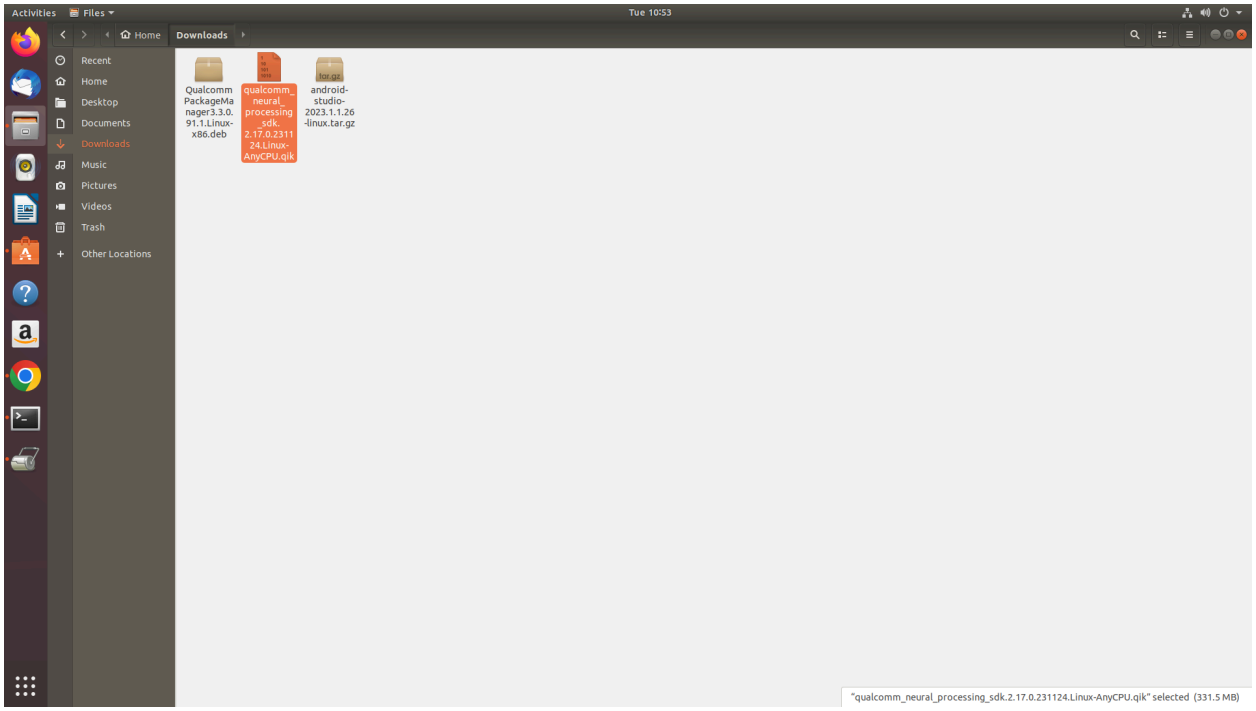
3. Verify that the downloading was successful by checking the files in the directory. For this, you can run the following commands:

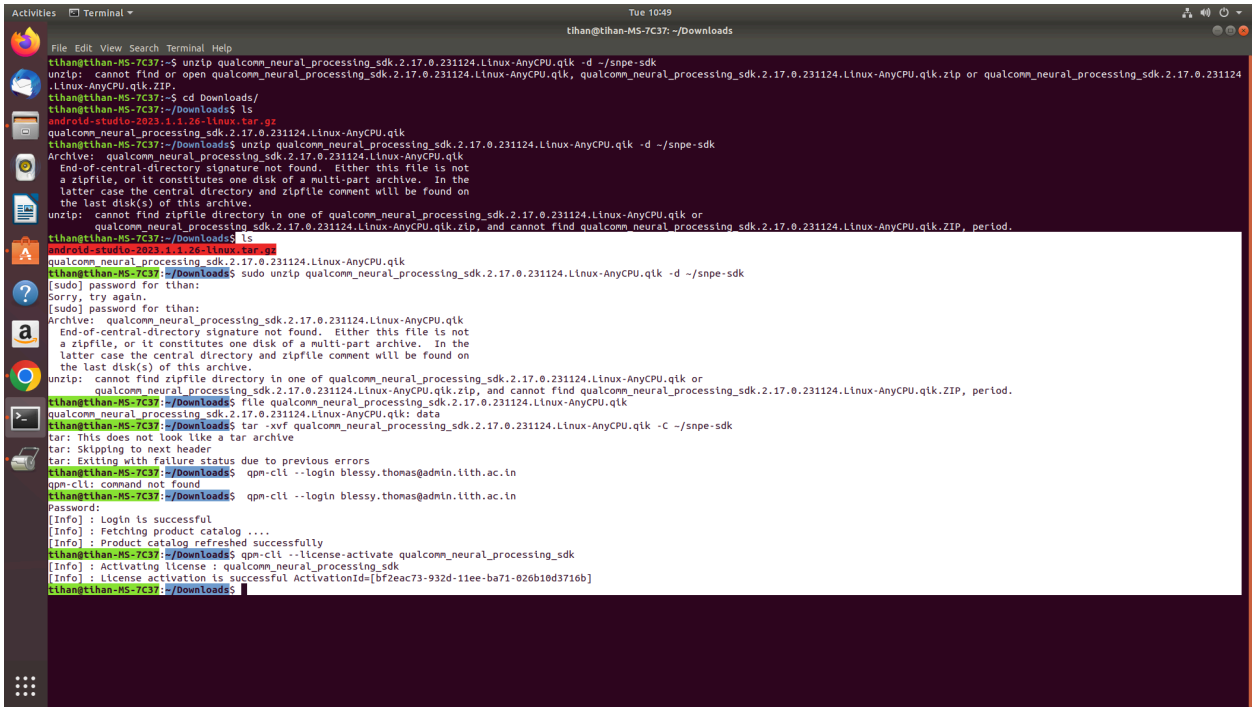
```
$ cd snpe-2.5.0.4052/
$ ls
```

Download and install Android Studio. Android Studio provides a set of tools like a SDK and NDK to work with the Neural Processing SDK.

On Linux 18.04
1) QP3 tools installation (so that we can use qpi commands) - Qualcomm Manager

2) Download data file for extraction





Neural processing SDK

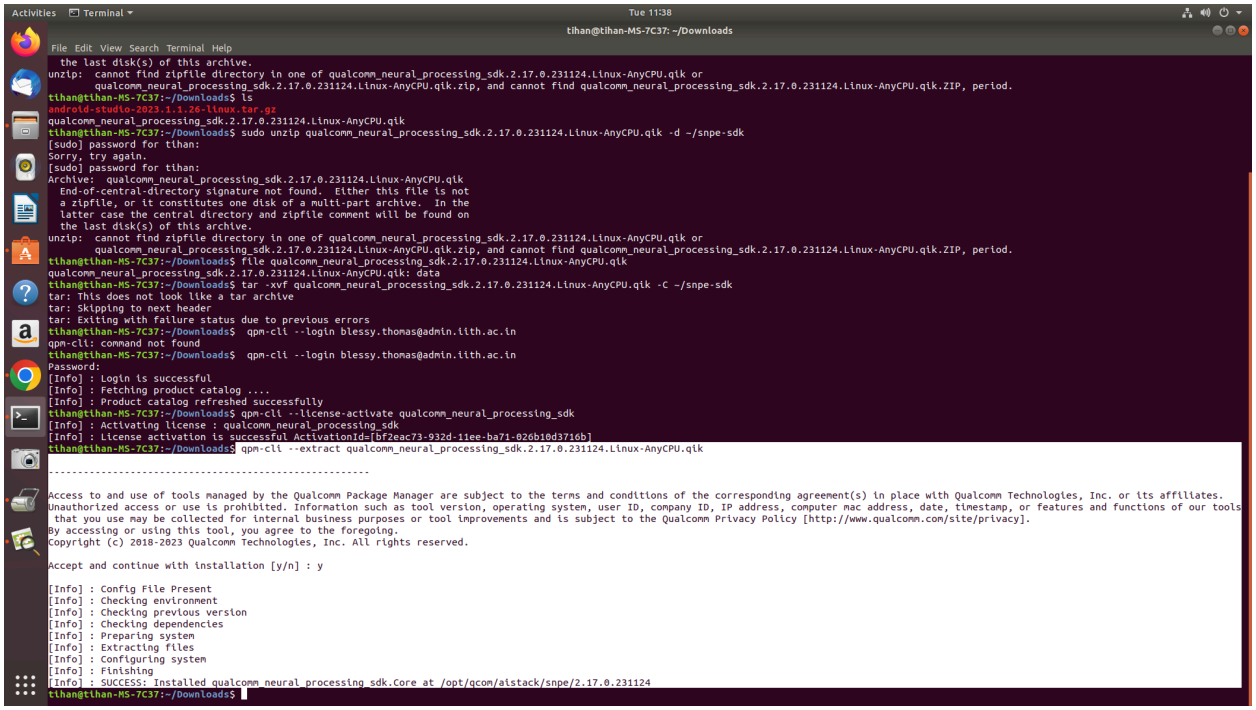


Fig3: Installed SDK.

Brief 8k record

Gigapixel speed ISP powered by top of the line Qualcomm® Spectra™ **480 ISP** with ability to process **2 Gigapixels** per second. This Gigapixel speed delivers new camera features including 8K video recording, 7 camera concurrency, capture 200-megapixel photos, and simultaneously capture 4K HDR video and 64 MP (with zero shutter lag) photos.

Up to 200 MP Photo Capture:

- Your camera can take really huge photos, up to 200 megapixels. More megapixels mean more detail in your pictures.

Up to 25 MP Dual Camera @ 30 FPS with Zero Shutter Lag:

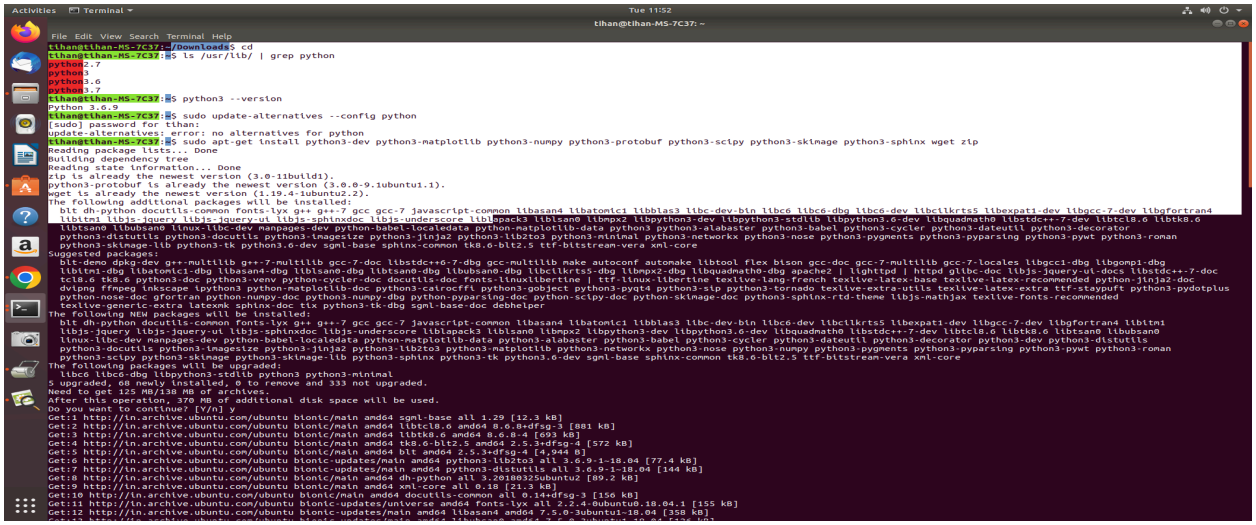
- It can handle two cameras working together, and each camera can capture photos up to 25 megapixels. They can do this at a rate of 30 frames per second, and there's no delay when you press the button to take a photo (Zero Shutter Lag).

Up to 64 MP Single Camera @ 30 FPS with Zero Shutter Lag:

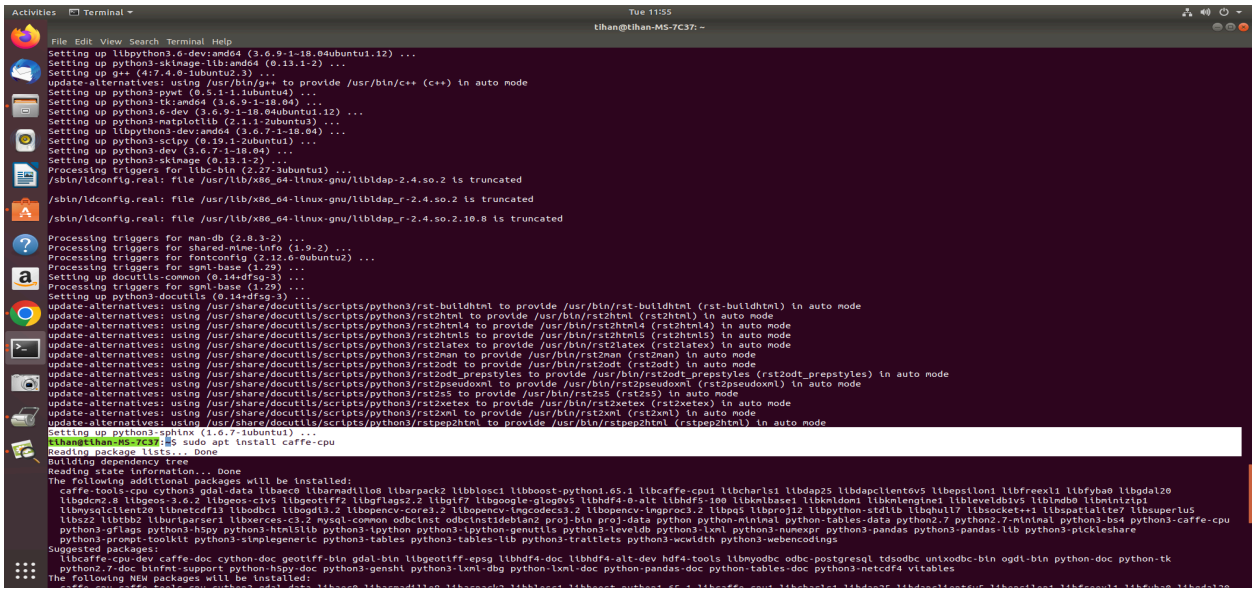
- If you're using just one camera, it can capture really high-quality photos, up to 64 megapixels. Again, it can do this at 30 frames per second, and there's no delay when taking photos.
- Support for 12 Cameras by D-PHY and 18 Cameras by C-PHY (7 Concurrent):
- Your device can connect to different cameras. Using something called D-PHY, it can connect to up to 12 cameras, and using C-PHY, it can connect to up to 18 cameras. And the cool thing is, it can work with 7 cameras at the same time (concurrent),meaning it can use data from all these cameras simultaneously.

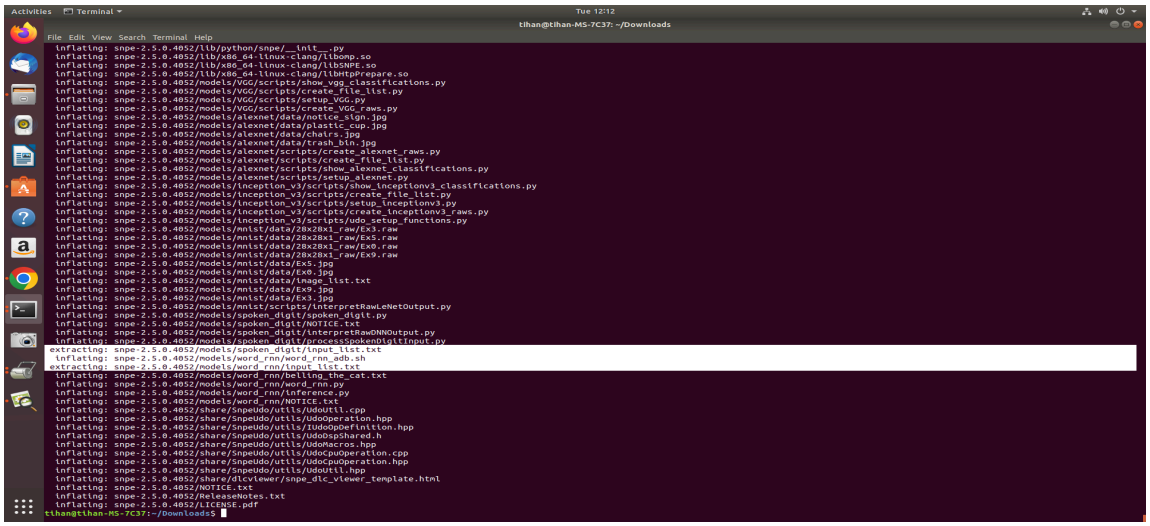
REFERENCE DOCUMENTATION :
https://developer.ridgerun.com/wiki/index.php/Qualcomm_Robotics_RB5/AI_hardware_acceleration/Neural_Processing_SDK/

1)Making Python3- 3.6.9 the default version. Installing necessary dependencies

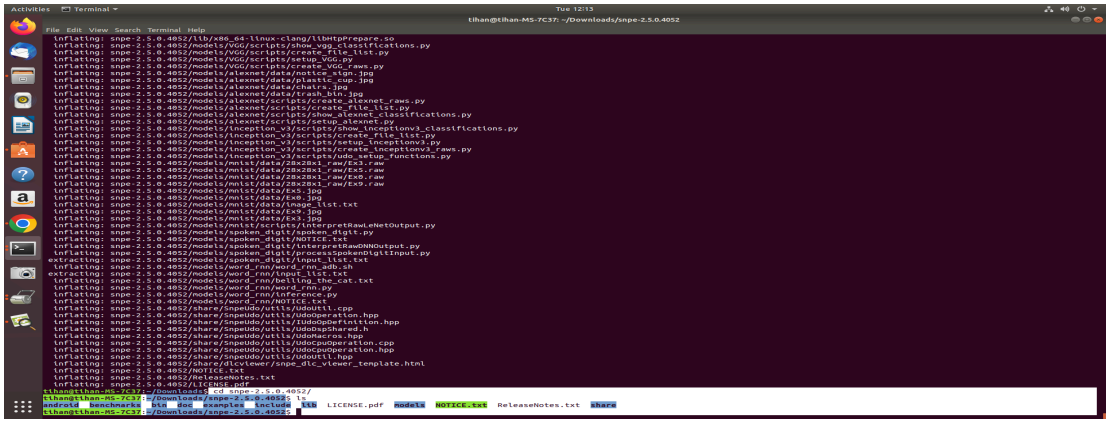


2) Installing frameworks : Caffe and Caffe2, TensorFlow, ONNX, PyTorch, or TensorFlow Lite.

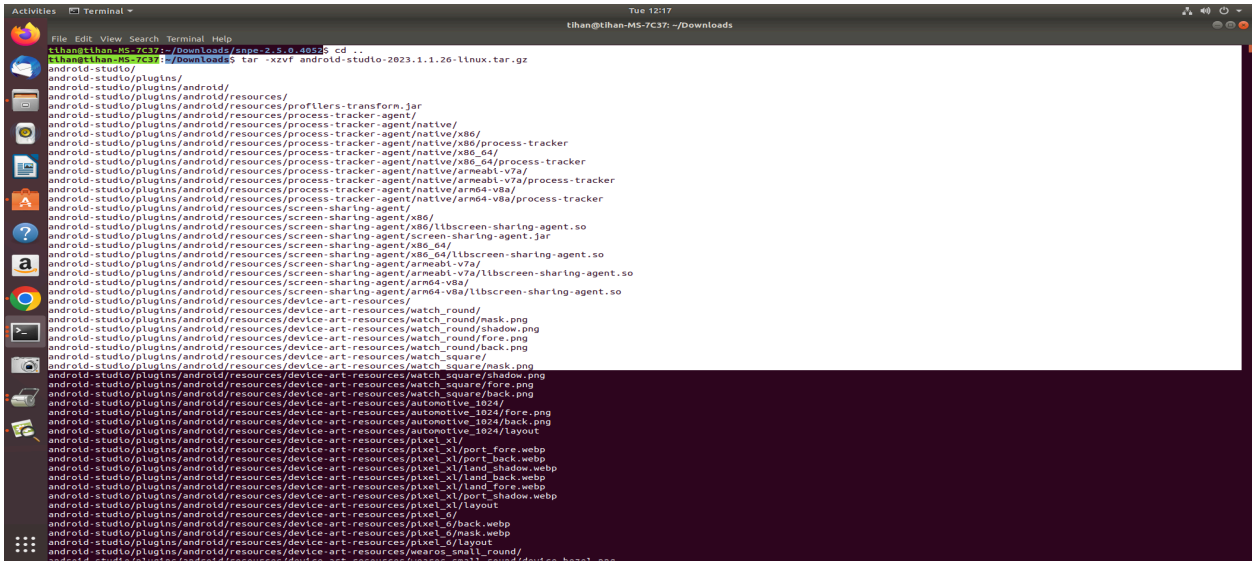




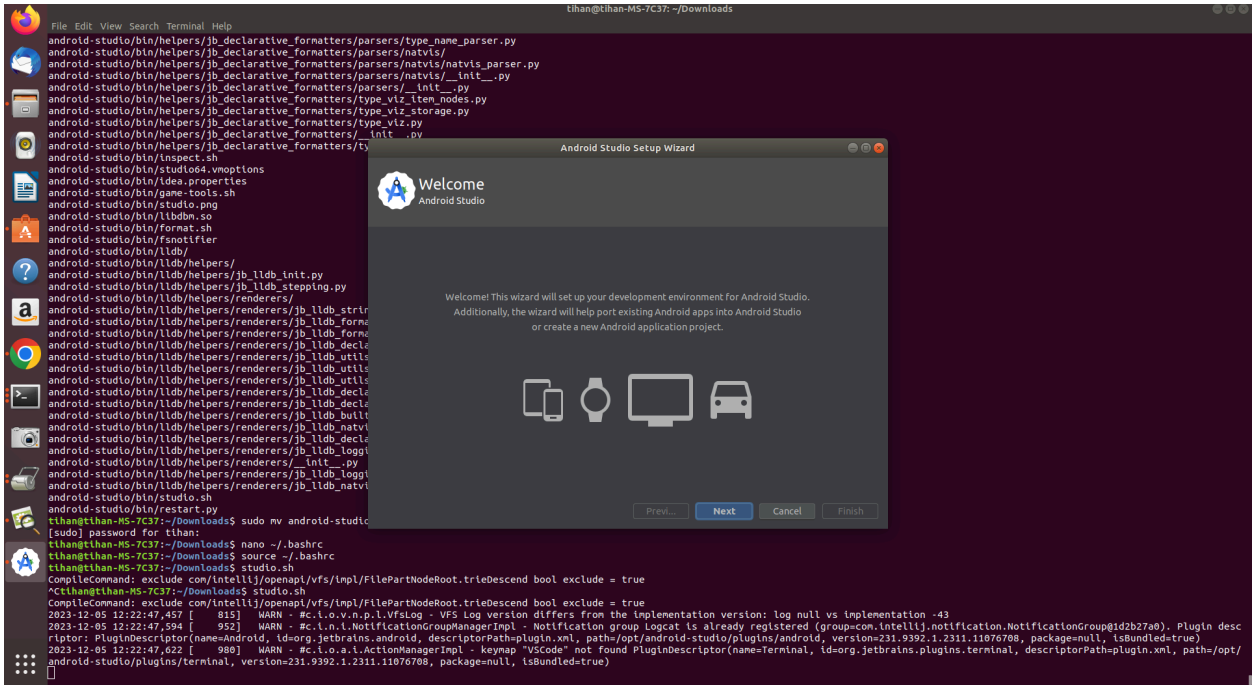

```
cd snpe-2.5.0.4052/
```



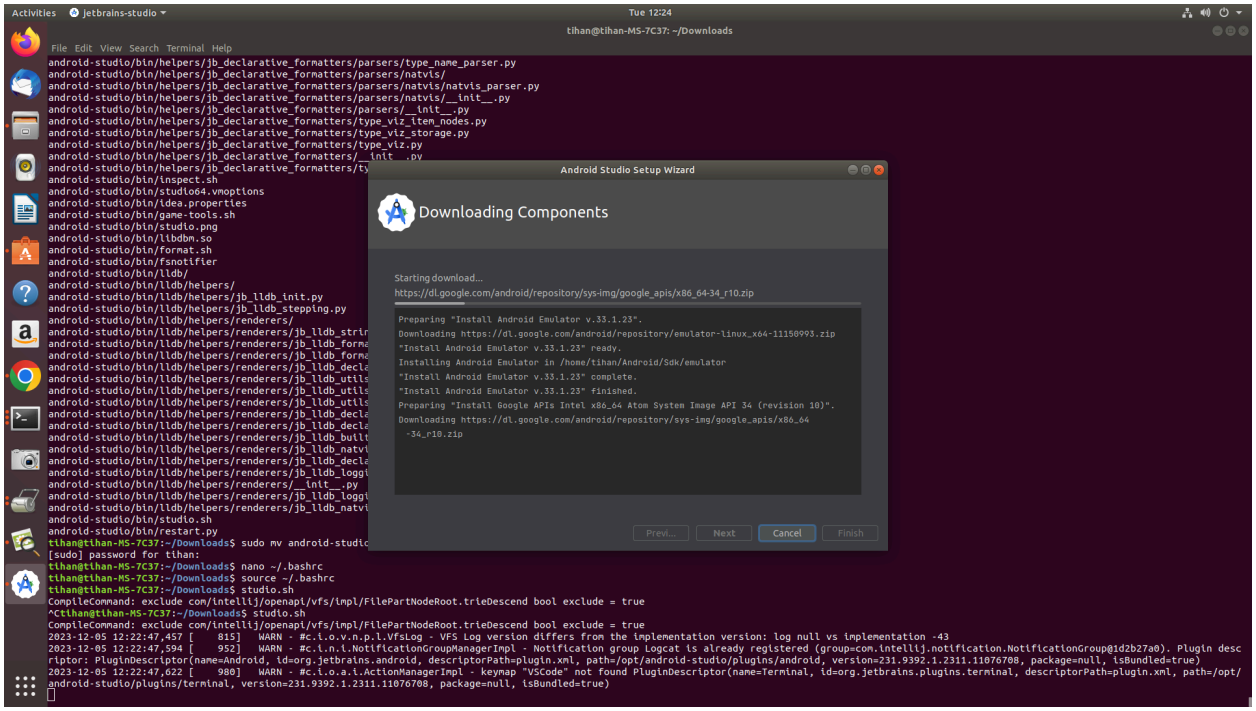
Install android studio



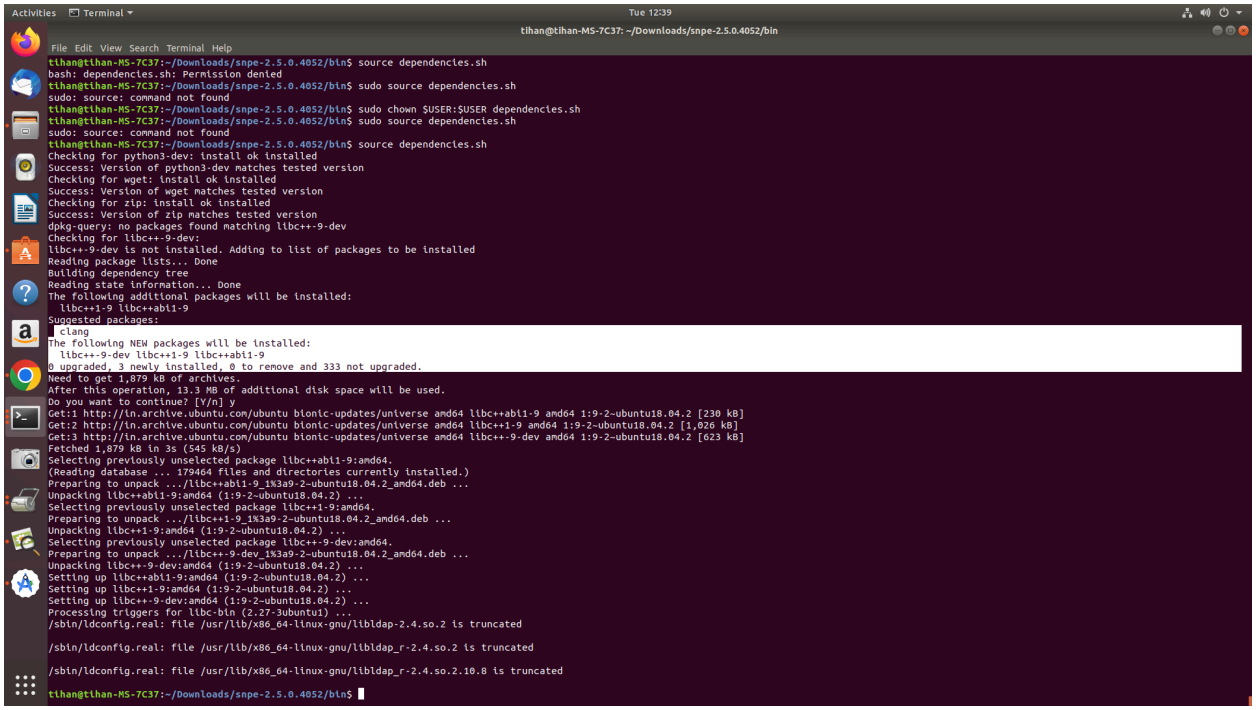
Installing Android studio

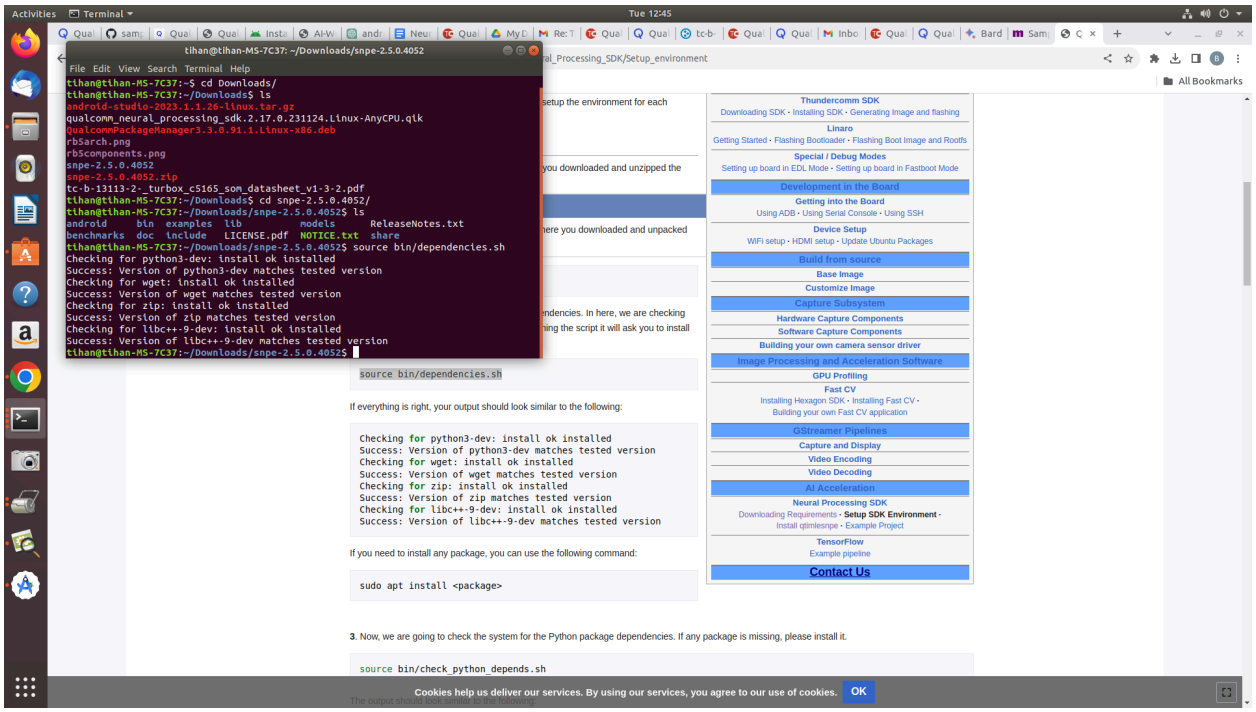


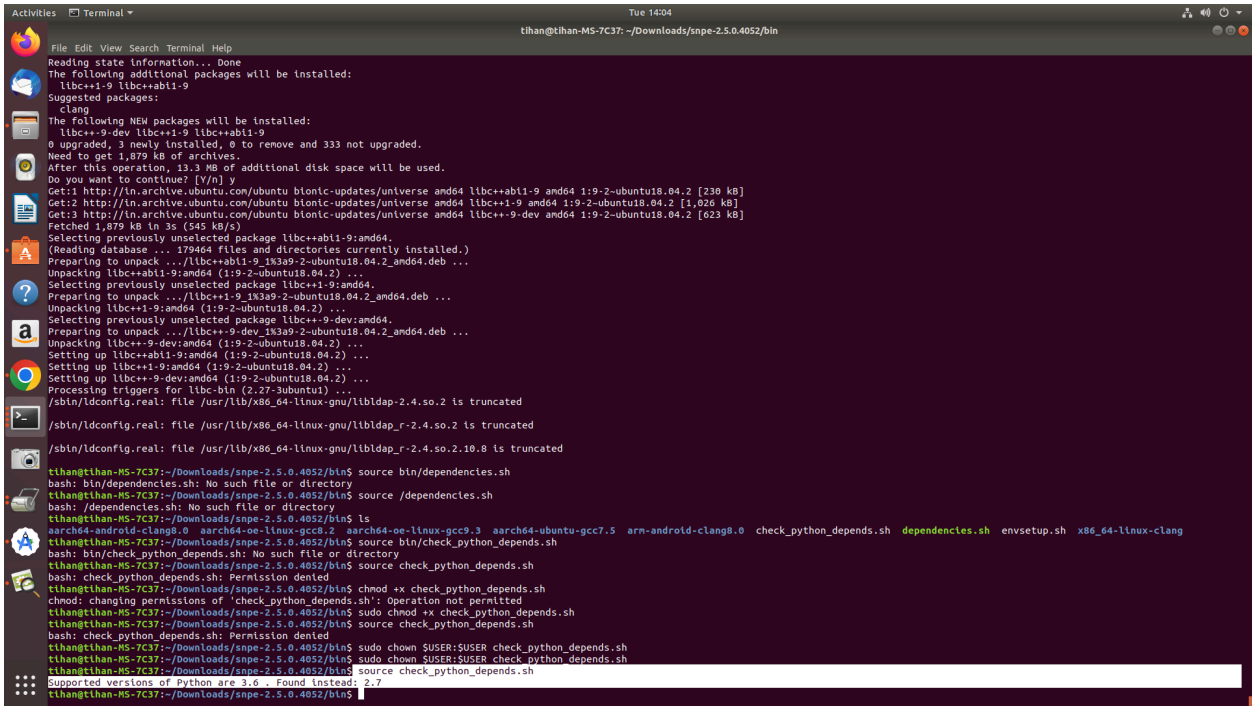
Downloading android studio components

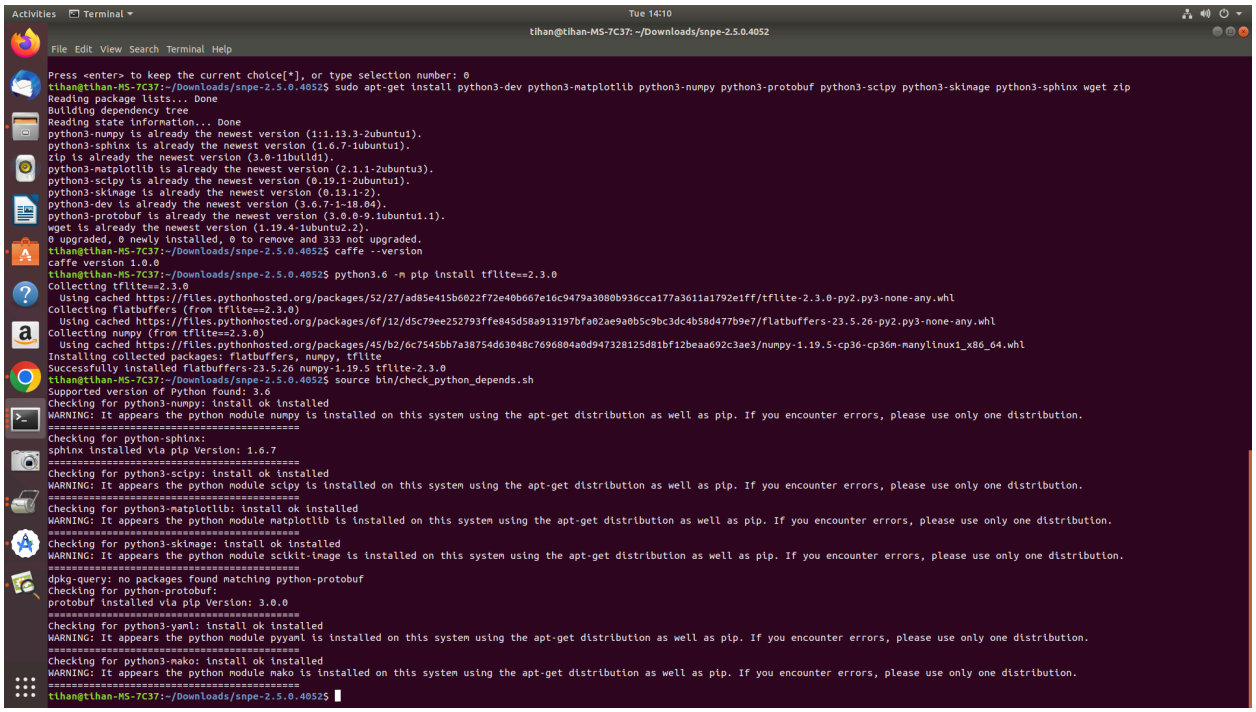


~/Downloads/snpe-2.5.0.4052

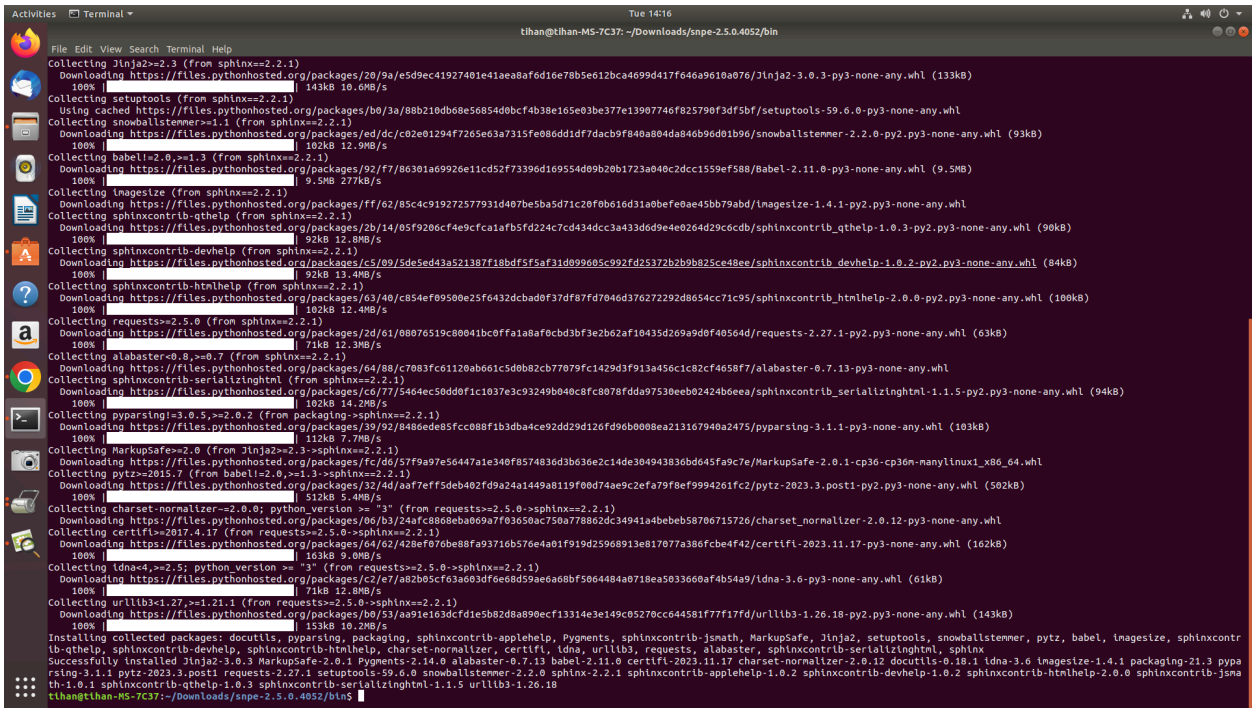

Activities
Terminal
Tue 12:39
tihan@tihan-MS-7C37: ~/Downloads/snpe-2.5.0.4052/bin
File Edit View Search Terminal Help
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source dependencies.sh
bash: dependencies.sh: Permission denied
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ sudo source dependencies.sh
sudo: source: command not found
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ sudo chown $USER:$USER dependencies.sh
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ sudo source dependencies.sh
sudo: source: command not found
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source dependencies.sh
Checking for python3-dev: install ok installed
Success: Version of python3-dev matches tested version
Checking for wget: install ok installed
Success: Version of wget matches tested version
Checking for zip: install ok installed
Success: Version of zip matches tested version
dpkg-query: no packages found matching libc++-9-dev
Checking for libc++-9-dev:
libc++-9-dev is not installed. Adding to list of packages to be installed
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
libc++1-9 libc++abi1-9
Suggested packages:
clang
The following NEW packages will be installed:
libc++-9-dev libc++1-9 libc++abi1-9
0 upgraded, 3 newly installed, 0 to remove and 333 not upgraded.
Need to get 1,879 kB of archives.
After this operation, 13.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libc++abi1-9 amd64 1:9-2~ubuntu18.04.2 [230 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libc++1-9 amd64 1:9-2~ubuntu18.04.2 [1,026 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libc++-9-dev amd64 1:9-2~ubuntu18.04.2 [623 kB]
Fetched 1,879 kB in 3s (545 kB/s)
Selecting previously unselected package libc++abi1-9:amd64.
(Reading database ... 179464 files and directories currently installed.)
Preparing to unpack .../libc++abi1-9_1%3a9-2~ubuntu18.04.2_amd64.deb ...
Unpacking libc++abi1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Selecting previously unselected package libc++1-9:amd64.
Preparing to unpack .../libc++1-9_1%3a9-2~ubuntu18.04.2_amd64.deb ...
Unpacking libc++1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Selecting previously unselected package libc++-9-dev:amd64.
Preparing to unpack .../libc++-9-dev_1%3a9-2~ubuntu18.04.2_amd64.deb ...
Unpacking libc++-9-dev:amd64 (1:9-2~ubuntu18.04.2) ...
Setting up libc++abi1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Setting up libc++1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Setting up libc++-9-dev:amd64 (1:9-2~ubuntu18.04.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
/sbin/ldconfig.real: file /usr/lib/x86_64-linux-gnu/libldap-2.4.so.2 is truncated
/sbin/ldconfig.real: file /usr/lib/x86_64-linux-gnu/libldap_r-2.4.so.2 is truncated
/sbin/ldconfig.real: file /usr/lib/x86_64-linux-gnu/libldap_r-2.4.so.2.10.8 is truncated
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$


Activities
Terminal
Tue 12:45
tihan@tihan-MS-7C37: ~/Downloads/snpe-2.5.0.4052
File Edit View Search Terminal Help
tihan@tihan-MS-7C37:~$ cd Downloads/
tihan@tihan-MS-7C37:~/Downloads$ ls
android-studio-2023.1.1.26-linux.tar.gz
qualcomm_neural_processing_sdk.2.17.0.231124.Linux-AnyCPU.qik
QualcommPackageManager3.3.0.91.1.Linux-x86.deb
rb5arch.png
rb5components.png
snpe-2.5.0.4052
snpe-2.5.0.4052.zip
tc-b-13113-2-_turbox_c5165_som_datasheet_v1-3-2.pdf
tihan@tihan-MS-7C37:~/Downloads$ cd snpe-2.5.0.4052/
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$ ls
android bin examples lib models ReleaseNotes.txt
benchmarks doc include LICENSE.pdf NOTICE.txt share
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$ source bin/dependencies.sh
Checking for python3-dev: install ok installed
Success: Version of python3-dev matches tested version
Checking for wget: install ok installed
Success: Version of wget matches tested version
Checking for zip: install ok installed
Success: Version of zip matches tested version
Checking for libc++-9-dev: install ok installed
Success: Version of libc++-9-dev matches tested version
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$
All Bookmarks
source bin/dependencies.sh
If everything is right, your output should look similar to the following:
Checking for python3-dev: install ok installed
Success: Version of python3-dev matches tested version
Checking for wget: install ok installed
Success: Version of wget matches tested version
Checking for zip: install ok installed
Success: Version of zip matches tested version
Checking for libc++-9-dev: install ok installed
Success: Version of libc++-9-dev matches tested version
If you need to install any package, you can use the following command:
sudo apt install <package>
3. Now, we are going to check the system for the Python package dependencies. If any package is missing, please install it.
source bin/check_python_depends.sh
Thundercomm SDK
Downloading SDK · Installing SDK · Generating Image and flashing
Linaro
Getting Started · Flashing Bootloader · Flashing Boot Image and Rootfs
Special / Debug Modes
Setting up board in EDL Mode · Setting up board in Fastboot Mode
Development in the Board
Getting into the Board
Using ADB · Using Serial Console · Using SSH
Device Setup
WiFi setup · HDMI setup · Update Ubuntu Packages
Build from source
Base Image
Customize Image
Capture Subsystem
Hardware Capture Components
Software Capture Components
Building your own camera sensor driver
Image Processing and Acceleration Software
GPU Profiling
Fast CV
Installing Hexagon SDK · Installing Fast CV ·
Building your own Fast CV application
GStreamer Pipelines
Capture and Display
Video Encoding
Video Decoding
AI Acceleration
Neural Processing SDK
Downloading Requirements · Setup SDK Environment ·
Install qtimlesnpe · Example Project
TensorFlow
Example pipeline
Contact Us
Cookies help us deliver our services. By using our services, you agree to our use of cookies.
OK

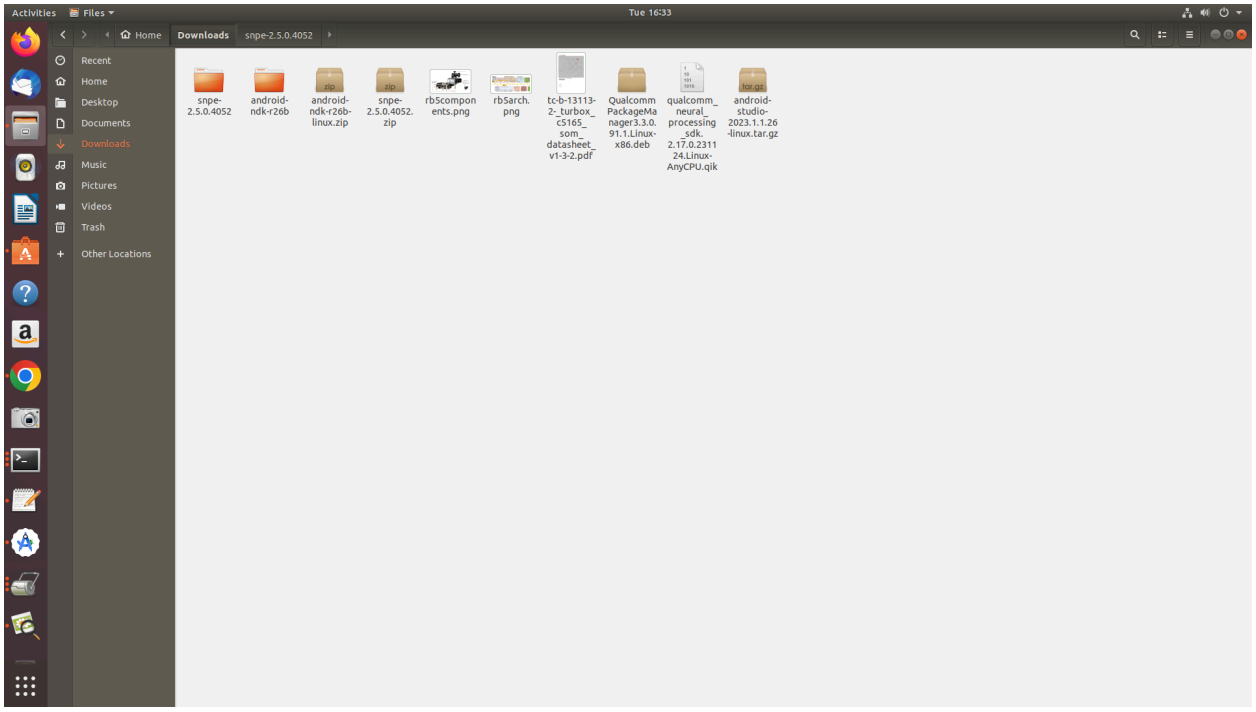

Activities
Terminal
Tue 14:04
tihan@tihan-MS-7C37: ~/Downloads/snpe-2.5.0.4052/bin
File Edit View Search Terminal Help
Reading state information... Done
The following additional packages will be installed:
libc++1-9 libc++abi1-9
Suggested packages:
clang
The following NEW packages will be installed:
libc++-9-dev libc++1-9 libc++abi1-9
0 upgraded, 3 newly installed, 0 to remove and 333 not upgraded.
Need to get 1,879 kB of archives.
After this operation, 13.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libc++abi1-9 amd64 1:9-2~ubuntu18.04.2 [230 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libc++1-9 amd64 1:9-2~ubuntu18.04.2 [1,026 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libc++-9-dev amd64 1:9-2~ubuntu18.04.2 [623 kB]
Fetched 1,879 kB in 3s (545 kB/s)
Selecting previously unselected package libc++abi1-9:amd64.
(Reading database ... 179464 files and directories currently installed.)
Preparing to unpack .../libc++abi1-9_1%3a9-2~ubuntu18.04.2_amd64.deb ...
Unpacking libc++abi1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Selecting previously unselected package libc++1-9:amd64.
Preparing to unpack .../libc++1-9_1%3a9-2~ubuntu18.04.2_amd64.deb ...
Unpacking libc++1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Selecting previously unselected package libc++-9-dev:amd64.
Preparing to unpack .../libc++-9-dev_1%3a9-2~ubuntu18.04.2_amd64.deb ...
Unpacking libc++-9-dev:amd64 (1:9-2~ubuntu18.04.2) ...
Setting up libc++abi1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Setting up libc++1-9:amd64 (1:9-2~ubuntu18.04.2) ...
Setting up libc++-9-dev:amd64 (1:9-2~ubuntu18.04.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
/sbin/ldconfig.real: file /usr/lib/x86_64-linux-gnu/libldap-2.4.so.2 is truncated
/sbin/ldconfig.real: file /usr/lib/x86_64-linux-gnu/libldap_r-2.4.so.2 is truncated
/sbin/ldconfig.real: file /usr/lib/x86_64-linux-gnu/libldap_r-2.4.so.2.10.8 is truncated
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source bin/dependencies.sh
bash: bin/dependencies.sh: No such file or directory
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source /dependencies.sh
bash: /dependencies.sh: No such file or directory
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ ls
aarch64-android-clang8.0 aarch64-oe-linux-gcc8.2 aarch64-oe-linux-gcc9.3 aarch64-ubuntu-gcc7.5 arm-android-clang8.0 check_python_depends.sh dependencies.sh envsetup.sh x86_64-linux-clang
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source bin/check_python_depends.sh
bash: bin/check_python_depends.sh: No such file or directory
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source check_python_depends.sh
bash: check_python_depends.sh: Permission denied
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ chmod +x check_python_depends.sh
chmod: changing permissions of 'check_python_depends.sh': Operation not permitted
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ sudo chmod +x check_python_depends.sh
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source check_python_depends.sh
bash: check_python_depends.sh: Permission denied
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ sudo chown $USER:$USER check_python_depends.sh
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ sudo chown $USER:$USER check_python_depends.sh
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$ source check_python_depends.sh
Supported versions of Python are 3.6 . Found instead: 2.7
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052/bin$


Activities
Terminal
Tue 14:10
tihan@tihan-MS-7C37: ~/Downloads/snpe-2.5.0.4052
File Edit View Search Terminal Help
Press <enter> to keep the current choice[*], or type selection number: 0
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$ sudo apt-get install python3-dev python3-matplotlib python3-numpy python3-protobuf python3-scipy python3-skimage python3-sphinx wget zip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-numpy is already the newest version (1:1.13.3-2ubuntu1).
python3-sphinx is already the newest version (1.6.7-1ubuntu1).
zip is already the newest version (3.0-11build1).
python3-matplotlib is already the newest version (2.1.1-2ubuntu3).
python3-scipy is already the newest version (0.19.1-2ubuntu1).
python3-skimage is already the newest version (0.13.1-2).
python3-dev is already the newest version (3.6.7-1~18.04).
python3-protobuf is already the newest version (3.0.0-9.1ubuntu1.1).
wget is already the newest version (1.19.4-1ubuntu2.2).
0 upgraded, 0 newly installed, 0 to remove and 333 not upgraded.
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$ caffe --version
caffe version 1.0.0
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$ python3.6 -m pip install tflite==2.3.0
Collecting tflite==2.3.0
Using cached https://files.pythonhosted.org/packages/52/27/ad85e415b6022f72e40b667e16c9479a3080b936cca177a3611a1792e1ff/tflite-2.3.0-py2.py3-none-any.whl
Collecting flatbuffers (from tflite==2.3.0)
Using cached https://files.pythonhosted.org/packages/6f/12/d5c79ee252793ffe845d58a913197bfa02ae9a0b5c9bc3dc4b58d477b9e7/flatbuffers-23.5.26-py2.py3-none-any.whl
Collecting numpy (from tflite==2.3.0)
Using cached https://files.pythonhosted.org/packages/45/b2/6c7545bb7a38754d63048c7696804a0d947328125d81bf12beaa692c3ae3/numpy-1.19.5-cp36-cp36m-manylinux1_x86_64.whl
Installing collected packages: flatbuffers, numpy, tflite
Successfully installed flatbuffers-23.5.26 numpy-1.19.5 tflite-2.3.0
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$ source bin/check_python_depends.sh
Supported version of Python found: 3.6
Checking for python3-numpy: install ok installed
WARNING: It appears the python module numpy is installed on this system using the apt-get distribution as well as pip. If you encounter errors, please use only one distribution.
Checking for python-sphinx:
sphinx installed via pip Version: 1.6.7
Checking for python3-scipy: install ok installed
WARNING: It appears the python module scipy is installed on this system using the apt-get distribution as well as pip. If you encounter errors, please use only one distribution.
Checking for python3-matplotlib: install ok installed
WARNING: It appears the python module matplotlib is installed on this system using the apt-get distribution as well as pip. If you encounter errors, please use only one distribution.
Checking for python3-skimage: install ok installed
WARNING: It appears the python module scikit-image is installed on this system using the apt-get distribution as well as pip. If you encounter errors, please use only one distribution.
dpkg-query: no packages found matching python-protobuf
Checking for python-protobuf:
protobuf installed via pip Version: 3.0.0
Checking for python3-yaml: install ok installed
WARNING: It appears the python module pyyaml is installed on this system using the apt-get distribution as well as pip. If you encounter errors, please use only one distribution.
Checking for python3-mako: install ok installed
WARNING: It appears the python module mako is installed on this system using the apt-get distribution as well as pip. If you encounter errors, please use only one distribution.
tihan@tihan-MS-7C37:~/Downloads/snpe-2.5.0.4052$

Download dependencies



cd /opt/android-studio/bin



1. JDK
2. Eclipse IDE
3. Android SDK Downloader

4. Android ADT
5. Android SDK platform support
6. Cygwin Environment
7. Android NDK
8.

Point a) ~/Downloads/snpe-2.5.0.4052$
To install : https://developer.qualcomm.com/software/qualcomm-computer-vision-sdk/setting-up

Very imp `sudo vi /etc/apt/apt.conf.d/10sandbox`

In the file, add the following line:

```
APT::Sandbox::User "root";
```

```
cp -r /yolov5/* /new1/
```