

Things to do after flashing Jetson.

Connect to Jetson using HDMI or Headless.

HDMI

- you can connect to the Jetson using and HDMI cable.

Headless

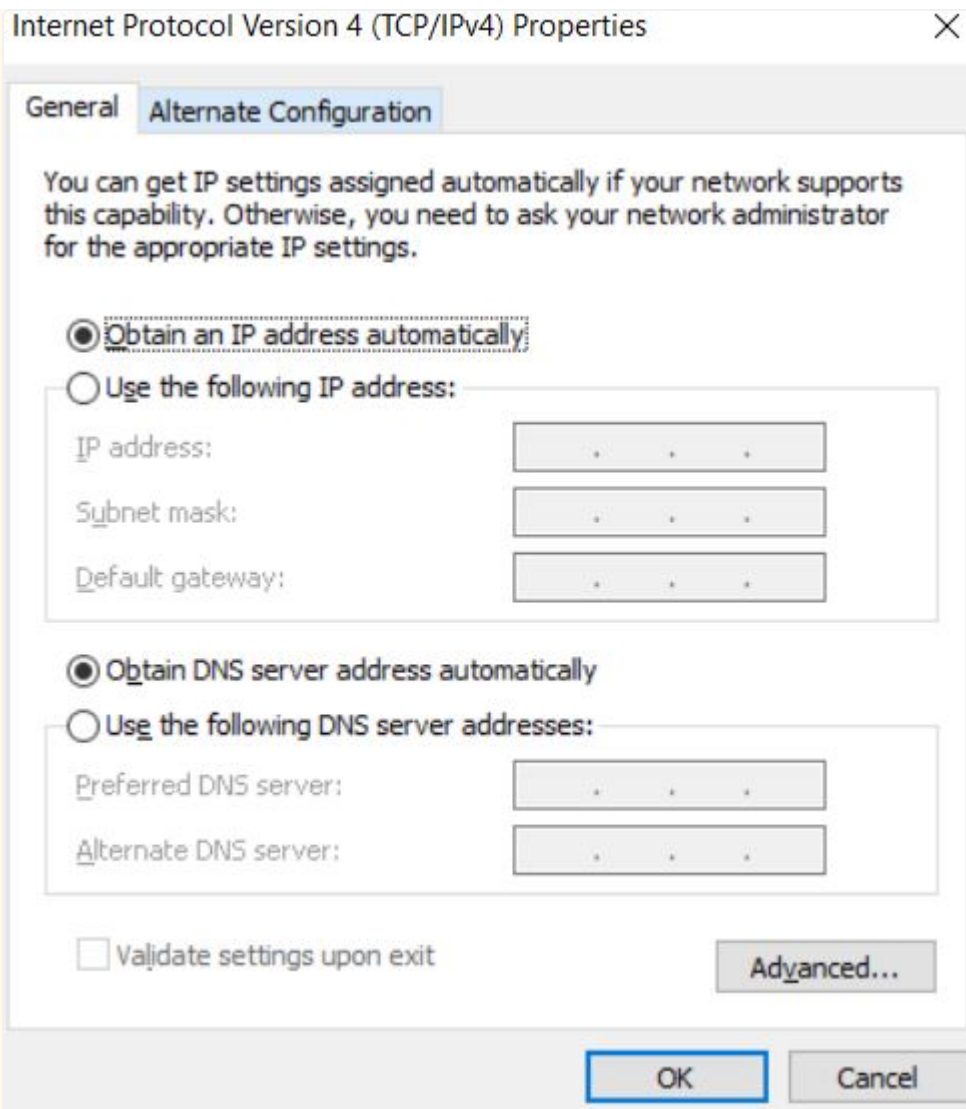
- you can connect to the Jetson with SSH by establishing a network with a micro/Type C cable (depends on your board production).
- the example below shows a micro cable connection to Jetson NX.



- after connecting the cable we should be able to ping the device using the following IP

192.168.55.1

Warning: if you can't ping the Jetson using 192.168.55.1 you may need to change the IPV4 protocol settings so that it obtains the IP addresses automatically.



- if you can ping the Jetson use the SSH protocol to access it

```
ssh <username>@192.168.55.1
```

Warning: if you get the following **REMOTE HOST IDENTIFICATION HAS CHANGED!**

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:nvrRt6cAzX3gmPNCfbrTS9HffuovUDCziRE8Yc1/4d0.
Please contact your system administrator.
```

just delete the SSH folder found in `C:\users/<your_user_name>` .

Connect to WIFI from the terminal using nmcli.

What Is nmcli?

NMCLI (NetworkManager Command-line) manages the network manager interface and identifies available internet connections. It can be used to activate, edit, and delete wireless network connections.

Connect to WIFI.

- use the following command to connect to your WIFI.

```
sudo nmcli dev wifi connect <network-ssid> password <network-password>
```

Disconnect from WIFI.

```
nmcli con down ssid/uuid
```

Connect to a saved WIFI.

```
nmcli con up ssid/uuid
```

Show all saved connections.

```
nmcli con show
```

Install XRDP.

```
sudo apt-get install xrdp
```

Warning: you may need to use `sudo apt-get update` before installing xrdp.

Install Jtop

```
sudo -H pip3 install -U jetson-stats
```

Note: we need to reboot the system before using jtop

Install VS-Code.

- go to [Download Visual Studio Code - Mac, Linux, Windows](#)
- download .deb `arm64` version.
- do the following commands.

```
sudo dpkg -i <path_to_downloaded_vs-code_.deb>
```

```
sudo apt install apt-transport-https
```

```
sudo apt update
```

```
sudo apt install
```

- we can start vs-code in our current directory using `code .`
-

Python Libraries Installation

Pip3

```
sudo apt install python3-pip
```

PyCUDA

```
pip3 install PyCUDA
```

Numpy

```
pip3 install cython  
pip3 install numpy
```

warning: on the jetson platform you may need to install cython before numpy.

Pillow and Matplotlib

```
pip3 install --upgrade pip  
pip3 install pillow  
pip3 install matplotlib
```

Jetson Stats

Installation

```
sudo -H pip3 install -U jetson-stats
```

Note: we need to reboot the system before using jtop

Usage

```
jtop
```

Documentation

- click [here](#) to go to documentation
-

VS-Code Installation

- go to [Download Visual Studio Code - Mac, Linux, Windows](#)
- download .deb `arm64` version.
- do the following commands.

```
sudo dpkg -i <path_to_downloaded_vs-code_.deb>
```

```
sudo apt install apt-transport-https
```

```
sudo apt update
```

```
sudo apt install code
```

- we can start vs-code in our current directory using `code .`

Build QGroundControl on Jetson

QT-5.15.2 Requirements.

Build essentials

```
sudo apt-get install build-essential perl python3 git
```

Libxcb

```
sudo apt-get install '^libxcb.*-dev' libx11-xcb-dev libglu1-mesa-dev libxrender-dev  
libxi-dev libxkbcommon-dev libxkbcommon-x11-dev
```

Accessibility

```
sudo apt-get install libatspi2.0-dev libdbus-1-dev
```

Qt WebKit

```
sudo apt-get install flex bison gperf libicu-dev libxslt-dev ruby
```

Qt WebEngine

```
sudo apt-get install libxcursor-dev libxcomposite-dev libxdamage-dev libxrandr-dev  
libxtst-dev libxss-dev libdbus-1-dev libevent-dev libfontconfig1-dev libcap-dev  
libpulse-dev libudev-dev libpci-dev libnss3-dev libasound2-dev libegl1-mesa-dev gperf  
bison nodejs
```

Qt Multimedia

```
sudo apt-get install libasound2-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev  
libgstreamer-plugins-good1.0-dev libgstreamer-plugins-bad1.0-dev
```

QDoc Documentation Generator Tool

```
sudo apt install clang libclang-dev
```

SSL Support

```
sudo apt install libssl-dev
```

Warning: SSL Support by default is optional in QT so you may be attempted to skip it; don't because we need it to build QGroundControl

QGroundControl Requirements.

Additional Packages

```
sudo apt-get install speech-dispatcher libudev-dev libsdl2-dev patchelf build-essential curl
```

Download and Build QT-5.15.2

Download QT-5.15.2

```
wget http://master.qt.io/archive/qt/5.15/5.15.2/single/qt-everywhere-src-5.15.2.tar.xz
tar -xpf qt-everywhere-src-5.15.2.tar.xz
cd qt-everywhere-src-5.15.2
```

Configure QT-5.15.2

```
sudo ./configure -opensource -confirm-license -dbus-linked -openssl-linked -skip
qtwebengine
```

Note: make sure that you are inside `qt-everywhere-src-5.15.2` folder.

you should see the following

```
Note: Also available for Linux: linux-clang linux-icc
Note: When linking against OpenSSL, you can override the default
library names through OPENSSL_LIBS.
For example:
    OPENSSL_LIBS='-L/opt/ssl/lib -lssl -lcrypto' ./configure -openssl-linked
Qt is now configured for building. Just run 'make'.
Once everything is built, you must run 'make install'.
Qt will be installed into '/usr/local/Qt-5.15.2'.
Prior to reconfiguration, make sure you remove any leftovers from
the previous build.
```

Build QT-5.15.2

```
sudo make
sudo make install
```

Note: `sudo make` step will take a couple of hours.

Copy/Create missing files and folder to `/usr/local/Qt-5.15.2`

```
sudo cp /usr/lib/aarch64-linux-gnu/libicu18n.so* /usr/local/Qt-5.15.2/lib/  
sudo cp /usr/lib/aarch64-linux-gnu/libicudata.so* /usr/local/Qt-5.15.2/lib/  
sudo cp /usr/lib/aarch64-linux-gnu/libicuuc.so* /usr/local/Qt-5.15.2/lib/  
sudo mkdir /usr/local/Qt-5.15.2/plugins/texttospeech
```

Warning: the missing files/folder shown above are needed to build QGroundControl.

[Download and Build QGroundControl.](#)

Clone QGroundControl Repository

```
git clone --recursive -j8 https://github.com/mavlink/qgroundcontrol.git
```

Note: make sure that you aren't inside `qt-everywhere-src-5.15.2` folder.

Configure QGroundControl

```
cd qgroundcontrol/  
mkdir build  
cd build  
sudo /usr/local/Qt-5.15.2/bin/qmake DEFINES=DISABLE_AIRMAP ../
```

Build QGroundControl

```
sudo make
```

Warning: you may want to use 1gb swap memory if you are building QGroundControl on Jetson Nano!.

Access Serial Devices

```
sudo usermod -a -G dialout $USER  
sudo apt-get remove modemmanager
```

Launch QGroundControl

```
./staging/QGroundControl
```