1. 下载官方镜像：<https://developer.nvidia.com/embedded/downloads>
2. 格式化SD卡
3. 下载烧录工具： <https://www.balena.io/etcher/>
4. 烧录镜像：傻瓜操作，很easy
5. 将SD卡插入nano，连接显示器，开机，初始简单配置--🡪进入桌面
6. 供电方式选择：首选USB供电,考虑到功率问题，采用5V/4A的高品质线（手机维修店都有的）

供电方式选择：DC电源供电，5V/4A 5V/6A均可，自制DC电源供电接口与电源模块

1. 更换APT源：

sudo gedit /etc/apt/sources.list 打开这个文件

用下面的内容代替刚才打开的文件里面的内容，保存，关闭文件

deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic main multiverse restricted universe

deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-security main multiverse restricted universe

deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-updates main multiverse restricted universe

deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-backports main multiverse restricted universe

deb-src http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic main multiverse restricted universe

deb-src http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-security main multiverse restricted universe

deb-src http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-updates main multiverse restricted universe

deb-src http://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-backports main multiverse restricted universe

1. 更新：sudo apt-get update  
    sudo apt-get upgrade
2. 安装ROS：<http://wiki.ros.org/ROS/Installation>

选择自己的ubuntun版本进行ROS安装，一步一步来

1. 安装rplidar驱动：

mkdir –p ~/ws\_rplidar/src

cd ws\_rplidar/src

catkin\_init\_workspace

git clone <https://github.com/robopeak/rplidar_ros>

cd ..

catkin\_make

sudo gedit .bashrc

source /home/nvidia/ws\_rplidar/devel/setup.bash (根据安装目录)

roslaunch rplidar\_ros rplidar.launch

(有可能会出现error: ……dev/ttyUSB0 permission denied……..)

解决办法：sudo usermod –aG dialout username (username位ubuntu用户名)

1. optitrack 及 vicon ros 通讯包安装

sudo apt-get install ros-melodic-vrpn-client-ros

mkdir vrpn\_ws

cd vrpn\_ws

mkdir src

cd src

git clone https://github.com/ros-drivers/vrpn\_client\_ros

cd ..

catkin\_make

运行：

roslaunch vrpn\_client\_ros sample.launch

1. nomachine 安装：

下载ARM64nomachine: <https://www.nomachine.com/download/linux&id=30&s=ARM>

将安装包放到home文件夹下：

sudo dpkg -i nomachine\_6.2.4\_1\_arm64.deb

1. 安装QTcreator

由于官网没有ARM版本，使用国内镜像安装

Sudo apt-get install qt5-default qtcreator –y

Reboot（重启）

1. 摄像头配置及测试：

在ros工作空间新建USB\_cam目录，在其下面新建SRC目录

cd ~/catkin\_ws/(假设工作空间是catkin\_ws)

mkdir -p usb\_cam/src

cd usb\_cam/src

catkin\_init\_workspace

git clone <https://github.com/bosch-ros-pkg/usb_cam.git>

cd ..

catkin\_make

sudo gedit .bashrc

添加：source /home/nvidia/ws\_rplidar/devel/setup.bash (根据安装目录)

source ~/catkin\_ws/devel/setup.bash

roscore

新建term

rosrun usb\_cam usb\_cam\_node

新建term

rosrun image\_view image\_view image:=/usb\_cam/image\_raw

1. Cuda版本查看问题：

nvcc –V

报错！！

添加环境变量

1. 如果确认cuda已经安装的话修改配置文件

sudo gedit ~/.bashrc

在文末添加

export PATH=/usr/local/cuda/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/local/cuda/lib64:$LD\_LIBRARY\_PATH

保存之后source一下：

source ~/.bashrc

然后再nvcc --version就可以查看到cuda的版本

1. 安装到吐的tensorflow的安装
2. 因为ROS自带的python版本为python2.7，想要是的ROS和tensorflow同时基于python3，折腾都是没用的，解决办法就是利用conda新建环境，与ros隔离开，但是还有不少的问题，首先anaconda官方并不支持jetson系列的aarch64架构，跳了一天的坑终于找到了这么个玩意儿—Archiconda,简单来说就aarch64架构版本的anaconda,官网下载地址：<https://github.com/Archiconda/build-tools/releases> 下载好之后安装就OK了
3. conda的问题解决之后可以安装jupyter，操作比较简单

sudo apt-get install python3-pip python3-dev #安装pip3

python3 -m pip install --upgrade pip

pip3 install --upgrade –user

sudo vim /usr/bin/pip3

改正文件里面的内容：

将原来的

from pip import main

if \_\_name\_\_ == '\_\_main\_\_':

sys.exit(main())

改成

from pip import \_\_main\_\_

if \_\_name\_\_ == '\_\_main\_\_':

sys.exit(\_\_main\_\_.\_main())

pip3 install jupyter notebook –user

1. 安装tensonflow,同样比较严重的问题，tensorflow只支持64位，那么aarch64架构的应该怎么安装呢，找到NVIDIA官网提供的aarch64架构的tensoeflow包

<https://developer.download.nvidia.cn/compute/redist/jp/v42/tensorflow-gpu/>

sudo apt install python3-numpy python3-scipy python3-pandas python3-matplotlib python3-sklearn libhdf5-serial-dev hdf5-tools

#安装依赖包

pip3 install --extra-index-url https://developer.download.nvidia.com/compute/redist/jp/v42 tensorflow-gpu==1.13.1+nv19.3 --user

如果出现版本不对应的问题的解决办法：

进入python后输入：import wheel.pep425tags

print(wheel.pep425tags.get\_supported())

根据版本提示信息选择自己的tensorflow版本，接下来继续安装就可以了。

安装完毕之后进行测试：

进入python

import tensorflow as tf

tf.\_\_version\_\_

能够查看tensorflow的版本即安装OK

1. Putty远程控制jetson nano

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> putppy现在地址

得到ip: ifconfig

打开putty,输入IP和端口，之后输入user\_name以及密码就OK

1. Opencv3.4.0的编译安装

下载两个压缩包：

<https://github.com/opencv/opencv/releases>  
<https://github.com/opencv/opencv_contrib/releases>

安装一些依赖，（根据自己的需要）

卸载原来的版本：

sudo apt-get purge libopencv\*

sudo apt-get update

在home目录下解压两个文件夹：

cd opencv-3.4.0

mkdir build

cd build

sudo apt install cmake

创建my\_cmake.sh

sudo gedit my\_cmake.sh

将下面的内容粘贴到打开的文件：

#!/bin/bash

cmake \

-D CMAKE\_BUILD\_TYPE=Release \

-D CMAKE\_INSTALL\_PREFIX=/usr \

-D BUILD\_PNG=OFF \

-D BUILD\_TIFF=OFF \

-D BUILD\_TBB=OFF \

-D BUILD\_JPEG=OFF \

-D BUILD\_JASPER=OFF \

-D BUILD\_ZLIB=OFF \

-D BUILD\_EXAMPLES=ON \

-D BUILD\_opencv\_java=OFF \

-D BUILD\_opencv\_python2=ON \

-D BUILD\_opencv\_python3=OFF \

-D ENABLE\_PRECOMPILED\_HEADERS=OFF \

-D WITH\_OPENCL=OFF \

-D WITH\_OPENMP=OFF \

-D WITH\_FFMPEG=ON \

-D WITH\_GSTREAMER=ON \

-D WITH\_GSTREAMER\_0\_10=OFF \

-D WITH\_CUDA=ON \

-D WITH\_GTK=ON \

-D WITH\_VTK=OFF \

-D WITH\_TBB=ON \

-D WITH\_1394=OFF \

-D WITH\_OPENEXR=OFF \

-D CUDA\_TOOLKIT\_ROOT\_DIR=/usr/local/cuda-9.0 \

-D CUDA\_ARCH\_BIN=6.2 \

-D CUDA\_ARCH\_PTX="" \

-D OPENCV\_EXTRA\_MODULES\_PATH=../opencv\_contrib-3.4.0/modules/ \

../

chmod u+x my\_cmake.sh

cd build

./my\_cmake.sh

Sudo make

Sudo make install

也就三个小时左右吧

之后就可以进入python环境import了