

妙算环境配置

1. 刷机

妙算自带的环境为jetpack3.3, ubuntu16.04, 不符合我们的需求, 所以需要把妙算的系统刷为jetpack4.4, ubuntu18.04的版本

刷机需要的环境有:DJI提供的内测版镜像, 一台ubuntu16.04的host主机, 妙算-2G及其配件

DJI镜像

没有镜像的可以向售后发邮件索要, 在这里我们提供百度云连接, 请自行下载

链接: https://pan.baidu.com/s/1BkU2a0fb5PXXe67F8Ae_fw
提取码: odcn

这里我们下载下来后可以看到有6G左右

 manifold2G_image_V0.4.4.0.tar.gz	2021/8/9 16:31	GZ 压缩文件	6,328,456...
--	----------------	---------	--------------

Host主机

我们还需要准备一台ubuntu16.04的host主机, 用来给妙算烧写系统

ubuntu16.04的镜像文件在此链接中下载: [ubuntu16.04](#)

接下来我们会在VMWare中安装虚拟机(镜像制作过程占用空间较大,请至少分配100G的硬盘空间), 该步骤请自行完成

妙算进入恢复模式



将该配件连接到妙算的扩展IO上

然后将你的host主机与妙算的USB3.0-micro-B接口相连

在接上妙算的电源时, 请先按住扩展上的RCV按钮, 然后接上电源, 按住RST键, 2秒后同时松开RST与RCV

这时候你在你的host主机上输入:

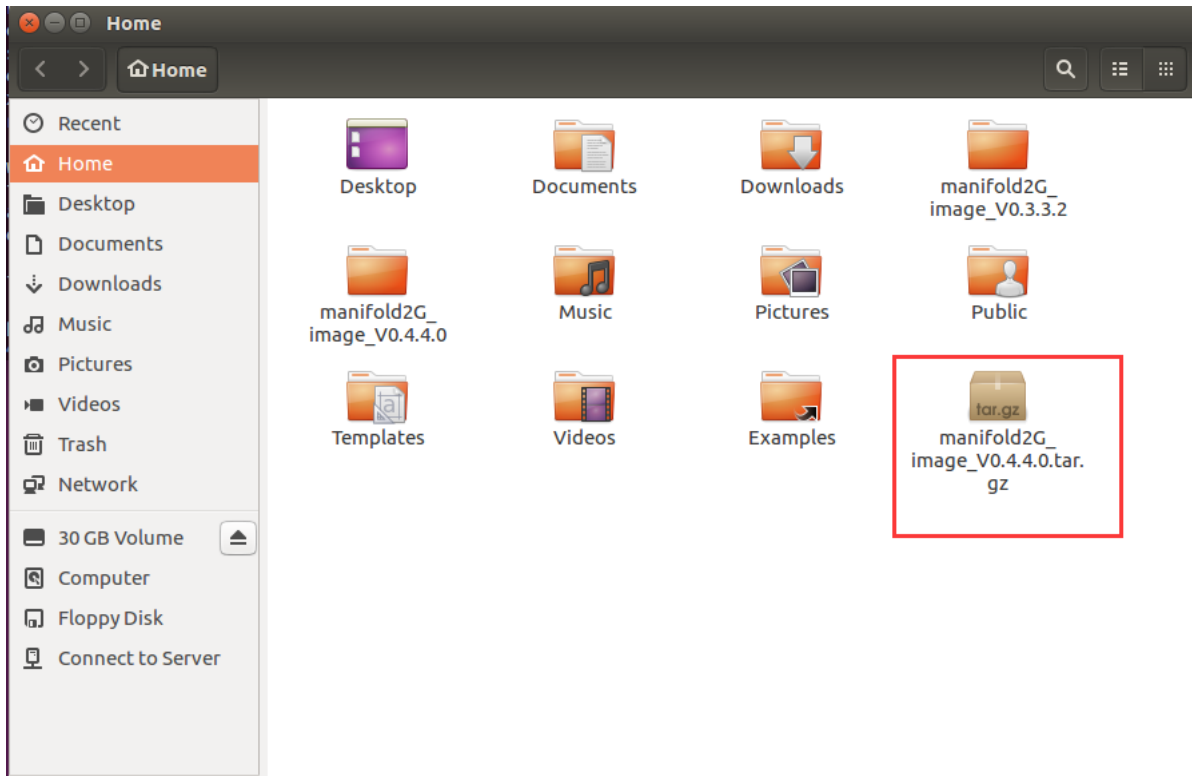
```
lsusb
```

假如出现以下输出, 就代表妙算进入了恢复模式:

```
Bus 001 Device 009: ID 0955:7c18 NVidia Corp.
```

上面三步完成之后, 就可以开始正式刷机了

首先, 将我们的DJI镜像文件上传到Host主机中



然后我们在该文件夹下打开终端, 依次输入以下命令:

```
sudo tar -zxvf manifold2G_image_V0.4.4.0.tar.gz #解压该压缩包
cd manifold2G_image_V0.4.4.0/Linux_for_Tegra/ #进入文件夹
sudo ./flash.sh jetson-tx2 mmcblk0p1 #开始烧写系统
```

此时就开始烧写系统了, 等待一阵后, 如果我们看到以下内容, 就说明烧写成功了

```
*** The target t186ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
```

重启妙算:



至此, 妙算系统烧写完成

2. 基本配置

ssh配置

妙算具有wifi功能, 我们在没有屏幕的时候可以通过ssh登录妙算

```
sudo apt update
sudo apt install openssh-*
sudo service ssh restart
```

开启高性能模式

```
sudo nvpmodel -m 0
```

3. 配置Pytorch

在刷好的妙算上面是没有python3的, 只有默认的python2, 所以我们需要安装先python3和pip3

```
# 更新软件源
sudo apt update
# 安装python3.6
sudo apt install python3.6
# 安装pip3
sudo apt install python3-pip
# 升级pip
sudo pip3 install --upgrade pip
```

有时候在跑模型时会出现一些模块没有的错误, 例如:

```
"No module named torchvision"
```

可以直接pip安装:

```
pip3 install torchvision
```

但是有的模块需要一些依赖, 我们需要先安装依赖, 比如常用的pillow库

```
# 安装必要依赖
sudo apt-get install libtiff5-dev libjpeg8-dev zlib1g-dev libfreetype6-dev
liblcms2-dev libwebp-dev libharfbuzz-dev libfribidi-dev tcl8.6-dev tk8.6-dev
python3-tk

# 安装pillow
pip3 install pillow
```

上述步骤弄完后, 便可以开始安装pytorch

我们使用Nvidia官方提供的torch包, 方便安装, [torch安装包链接](#)

PyTorch for Jetson - version 1.9.0 now available

Home > Autonomous Machines > Jetson & Embedded Systems > Jetson Nano



dusty_nv Moderator

43 Mar 27 '19

Below are pre-built PyTorch pip wheel installers for Python on Jetson Nano, Jetson TX1/TX2, and Jetson Xavier NX/AGX with JetPack 4.2 and newer.

You can now download the 14t-pytorch and 14t-m1 containers from NGC for JetPack 4.4 or newer

Download one of the PyTorch binaries from below for your version of JetPack, and see the installation instructions to run on your Jetson. These pip wheels are built for ARM aarch64 architecture, so run these commands on your Jetson (not on a host PC).

PyTorch pip wheels

- ▶ PyTorch v1.9.0
- ▶ PyTorch v1.8.0
- ▶ PyTorch v1.7.0
- ▶ PyTorch v1.6.0
- ▶ PyTorch v1.5.0
- ▶ PyTorch v1.4.0
- ▶ PyTorch v1.3.0
- ▶ PyTorch v1.2.0
- ▶ PyTorch v1.1.0
- ▶ PyTorch v1.0.0

Instructions

Mar 2019

1 / 876
Mar 2019

Aug 3



在这里我们按需选择合适版本的torch, 我选择用最新的版本1.9.0

将安装包上传到妙算上, 在安装包文件夹下打开终端, 输入以下命令:

```
# 安装torch
pip3 install torch-1.9.0-cp36-cp36m-linux_aarch64.whl
# 安装torchvision
pip3 install torchvision
```

安装完成后我们在终端输入:

```
python3
```

```
import torch
print(torch.cuda.is_available())
```

若无报错且输出为True那么代表pytorch配置完成

我在进行import torch的时候遇到了如下的错误:

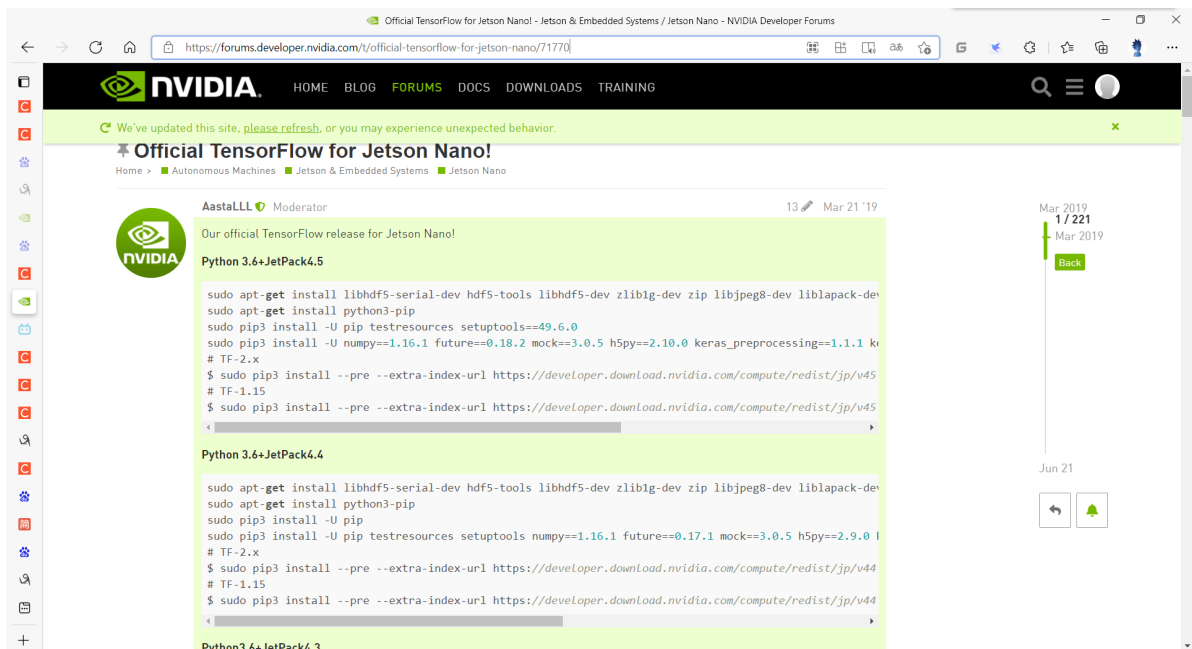
```
>>> import torch
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/dji/.local/lib/python3.6/site-packages/torch/__init__.py", line 197, in <module>
    from torch._C import * # noqa: F403
ImportError: libopenblas.so.0: cannot open shared object file: No such file or directory
```

我们只需apt安装相应的依赖就行, 依赖名格式一般为libxxx-dev

```
sudo apt install libopenblas-dev
```

4.配置Tensorflow

英伟达官网我们可以找到tensorflow的安装方式: [tensorflow安装](#)



我们的环境为JetPack4.4, Python3.6, 所以我们选择以下的安装方式:

```
sudo apt-get install libhdf5-serial-dev hdf5-tools libhdf5-dev zlib1g-dev zip libjpeg8-dev liblapack-dev libblas-dev gfortran
sudo apt-get install python3-pip
sudo pip3 install -U pip
sudo pip3 install -U pip testresources setuptools numpy==1.16.1 future==0.17.1 mock==3.0.5 h5py==2.9.0 keras_preprocessing==1.0.5 keras_applications==1.0.8 gast==0.2.2 futures protobuf pybind11
# tensorflow版本视需求安装, 我选择的是2.3版本
# TF-2.x
sudo pip3 install --pre --extra-index-url
https://developer.download.nvidia.com/compute/redis/jp/v44
tensorflow==2.3.1+nv20.12
# TF-1.15
sudo pip3 install --pre --extra-index-url
https://developer.download.nvidia.com/compute/redis/jp/v44 'tensorflow<2'
```

上述步骤完成后我们在python命令行中输入以下代码:

```
import tensorflow as tf
print(tf.test.is_built_with_cuda())
```

若输出True代表tensorflow安装完成且cuda可用

By:力创RM-张辉阳