

WeAct Studio

TX1/TX2 底板

使用教程

WEAct Studio

目录

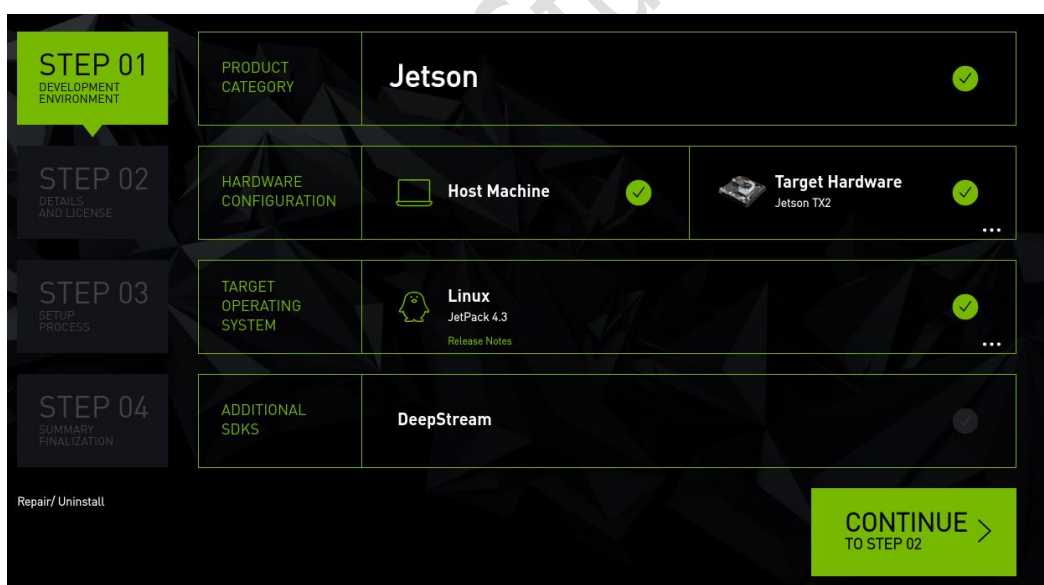
Revision History	3
1. 搭建烧写环境.....	4
2. 为 TX1/TX2 烧写系统	7
3. 安装系统组件(SDK)	10
4. 镜像备份与恢复.....	13
5. 使用 CAN 进行通信.....	15
联系我们.....	18

REVISION HISTORY

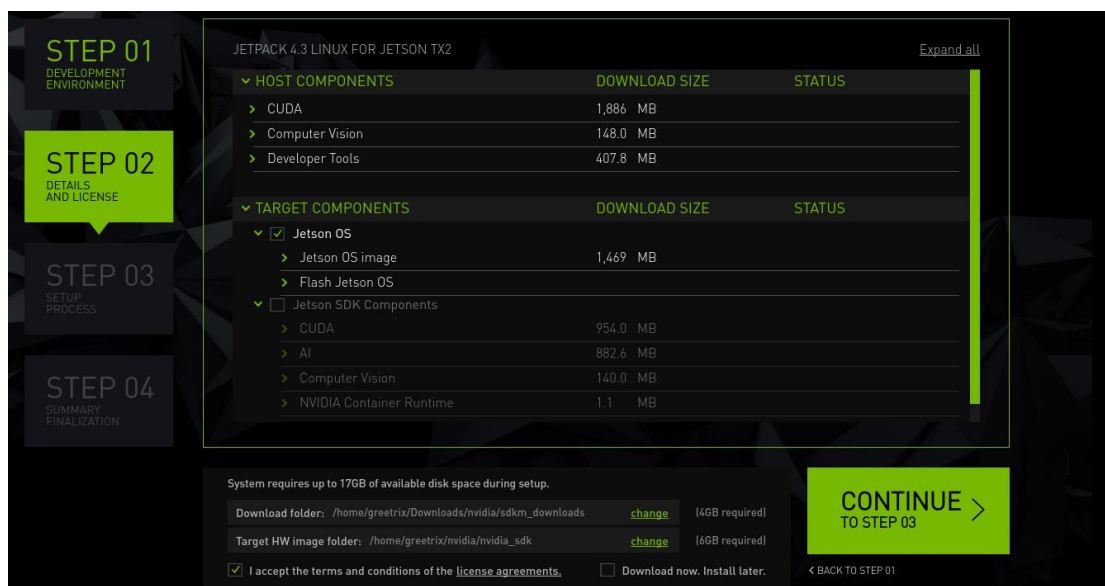
Draft Date	Revision	Description	Author
2020.1.19	V1.0	1. 初始版本	Greetrix
2020.1.25	V1.0.1	1. 增加 SDK 安装教程 2. 增加镜像备份教程	Greetrix
2020.1.26	V1.0.2	1. 增加 CAN 通信教程	Greetrix

1. 搭建烧写环境

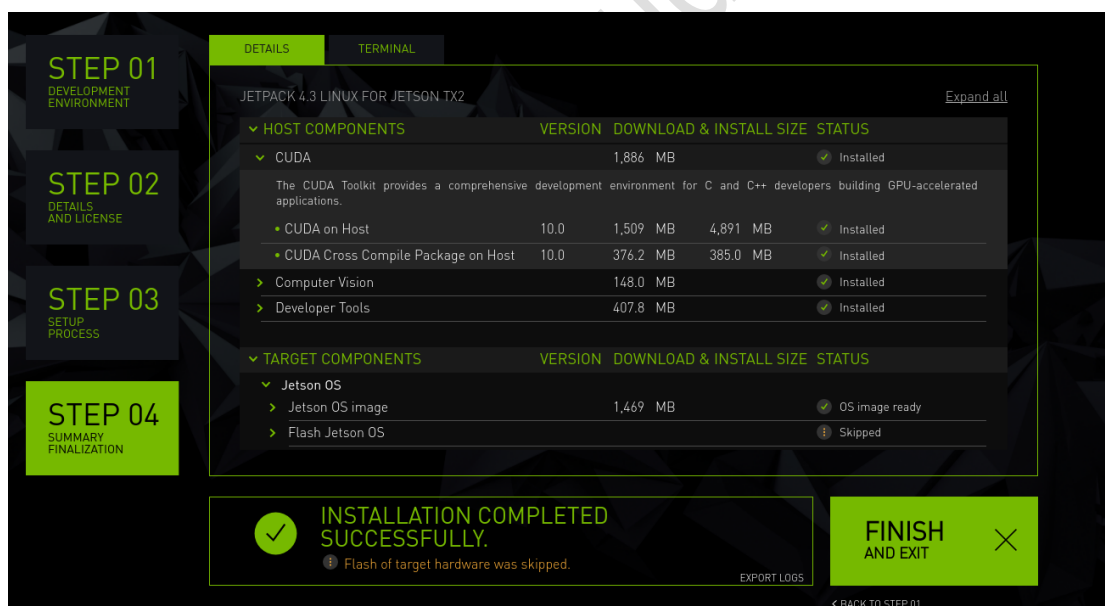
- a) 首先，需要一台装有 **Ubuntu16.04** 以上的电脑作为 HOST 端给 JetsonTX2 烧写，或者可以在 Windows 上安装 VMware 来实现。
- VMware 上如何安装 Ubuntu18.04:
<https://blog.csdn.net/u012556114/article/details/82751089>
- b) 在 NVIDIA 下载最新的 **SDK-Manager** 并在 ubuntu18.04 中安装（需要注册一个 NVIDIA 账号，后面也需要用到）
- SDK-Manager 下载地址: <https://developer.nvidia.com/nvidia-sdk-manager>
- c) 选择需要的 **Target Hardware** 以及 **JetPack** 版本（JetPack3.x 为 Ubuntu16.04, JetPack4.x 为 Ubuntu18.04），这里选择 **JetsonTX2** 的 JetPack4.3 版本（Ubuntu18.04），点击 CONTINUE 进行下一步。



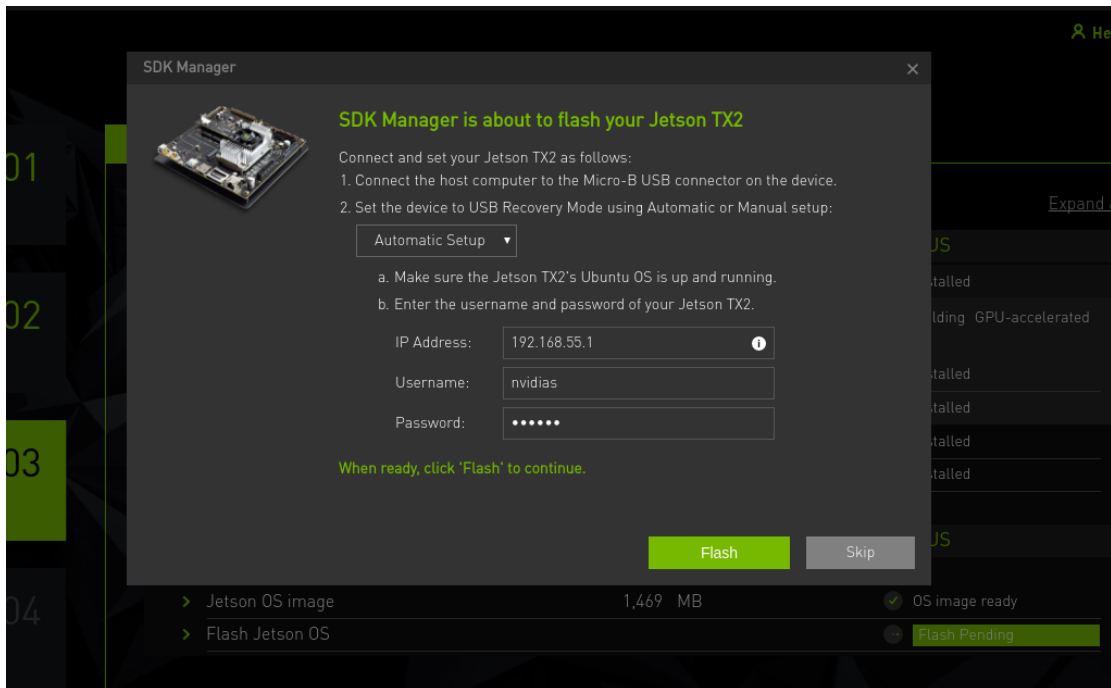
- d) 这里需要勾选 **I accept the terms and conditions of the license agreements**，取消勾选 **Jetson SDK Components**（后面会有专门教程安装 Jetson SDK），点击 CONTINUE 进行下一步。



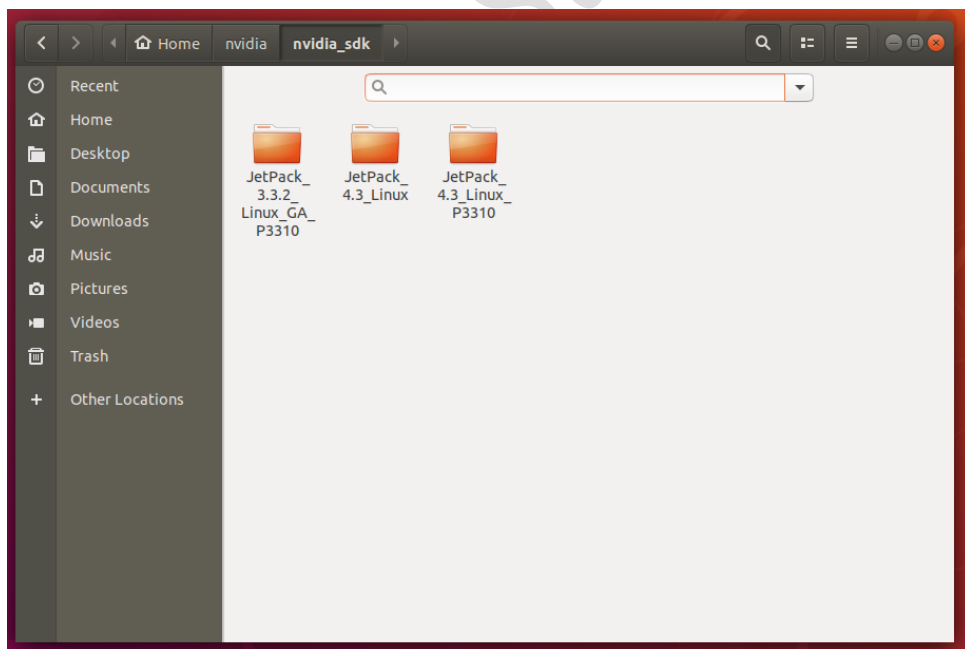
P.S: 请在畅通的网络环境下进行下载以及安装，下载或安装失败时，可点击 Retry 继续，直至全部状态为 Installed 并且显示绿色。



e) 安装过程中会弹出联网烧写的信息，选择 **Skip**（需要搭配我们提供的设备树烧写才能正常使用载板，后面教程将会单独介绍烧录）



f) 安装成功后，会在`~/nvidia/nvidia_sdk/`下有相应版本烧写所需的文件



g) 在终端通过 `sudo apt-get install python` 安装 python 支持以便后续烧写环境。

2. 为 TX1/TX2 烧写系统

a) 这里以 TX2 为例，在 WeAct Studio 的 github 或者码云上下载相应的设备树文件，对于 Jetpack3.x 版本使用 L4TR28，而 JetPack4.x 使用 L4TR32。

➤ Github: https://github.com/WeActTC/WeAct-TX1_2-CB

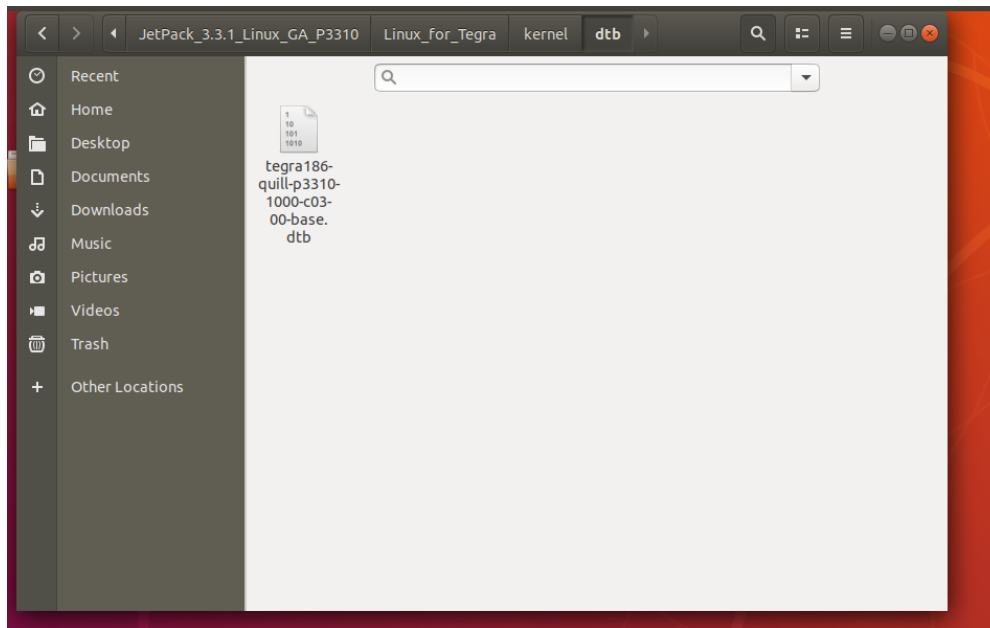
➤ 码云: https://gitee.com/WeAct-TC/WeAct-TX1_2-CB

b) 以 JetPack3.1 为例，首先需要进入 `~/nvidia/nvidia_sdk/JetPack_3.3.1_Linux_GA_P3310` 文件夹，打开这个文件夹下的 `p2771-0000.conf.common` 文件，将这个文件中的 ODMDATA 的值修改为 `0x7090000`，如下图所示，这个设置将 USB 配置为配置 4。

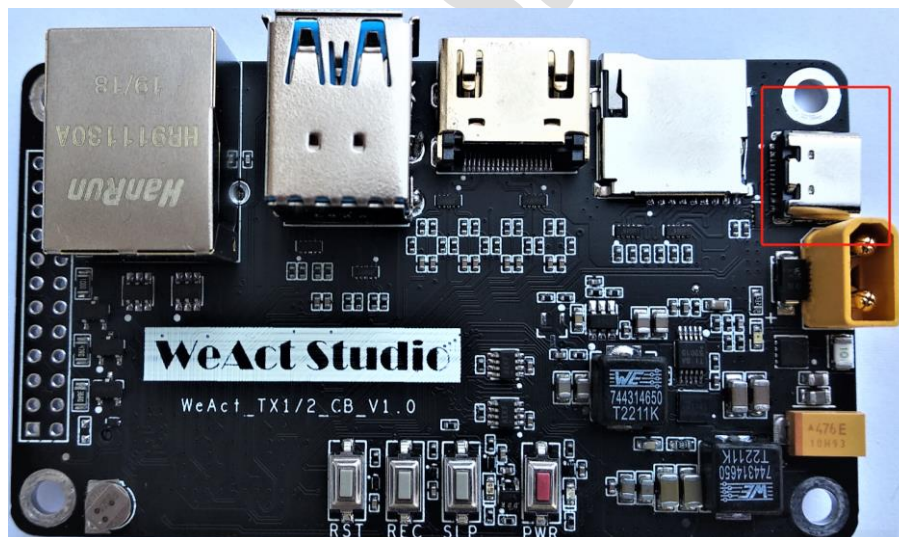
```
local bdv=${board_version^^};
local bid=${board_id^^};
local uboot_build=500;
local fromfab="-a00";
local tofab="-c03";
local pmicfab="-c00";
local bpfdtbfab="-c00";
local tbcdtbfab="-c03";
local kerndtbfab="-c03";
ODMDATA=0x7090000;
```

default = C03
default = C00
default = C00
default = C03
default = C03
default = C0X

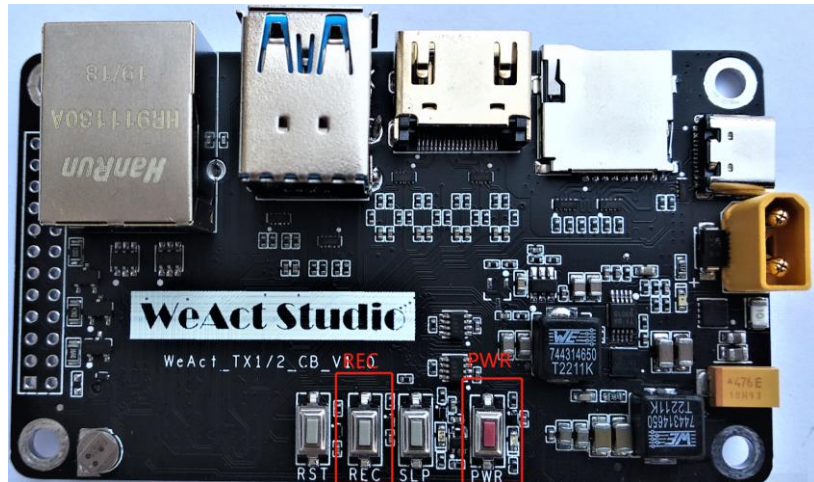
c) 找到相应版本的设备树 (`JetsonTX2/L4TR28/tegra186-quill-p3310-1000-c03-00-base.dtb`)，进入 `~/nvidia/nvidia_sdk/JetPack_3.3.1_Linux_GA_P3310/Linux_for_Tegra/kernel` 并把所有.dtb 文件删除，复制提供的设备树 `tegra186-quill-p3310-1000-c03-00-base.dtb` 至该目录。



d) 使用 **USB Type-C** 线连接载板上的 **USB OTG** 接口。



e) 摁住 **REC** 键，再摁 **PWR** 键开机，松开 **REC** 键进入 Recovery 模式，此时 VMWare 右下角会出现 **NVIDIA** 的 **USB 驱动标志**，或者打开终端，输入 **lsusb** 命令，会发现 **Nvidia Corp.**



- f) 进入 `~/nvidia/nvidia_sdk/JetPack_3.3.1_Linux_GA_P3310/Linux_for_Tegra`，打开终端，运行命令 `sudo ./flash.sh jetson-tx2 mmcblk0p1`，等烧录成功就可以使用啦。

```
[ 11.4190 ] Writing partition dram-ecc-fw with dram-ecc.bin
[ 11.4634 ] [.....] 100%
[ 11.6013 ] Writing partition spe-fw with spe_sigheader.bin.encrypt
[ 11.6578 ] [.....] 100%
[ 11.7094 ] Writing partition spe-fw_b with spe_sigheader.bin.encrypt
[ 11.7912 ] [.....] 100%
[ 11.8424 ] Writing partition mb2 with nvtboot_sigheader.bin.encrypt
[ 11.8987 ] [.....] 100%
[ 11.9497 ] Writing partition mb2_b with nvtboot_sigheader.bin.encrypt
[ 12.0183 ] [.....] 100%
[ 12.0678 ] Writing partition mts-preboot with preboot_d15_prod_cr_sigheader.bi
n.encrypt
[ 12.1376 ] [.....] 100%
[ 12.1874 ] Writing partition mts-preboot_b with preboot_d15_prod_cr_sigheader.
bin.encrypt
[ 12.2671 ] [.....] 100%
[ 12.3118 ] Writing partition SMD with slot_metadata.bin
[ 12.3921 ] [.....] 100%
[ 12.5518 ] Writing partition SMD_b with slot_metadata.bin
[ 12.5880 ] [.....] 100%
[ 12.6293 ] Writing partition master_boot_record with mbr_1.3.bin
[ 12.7024 ] [.....] 100%
[ 12.7468 ] Writing partition APP with system.img
[ 12.7741 ] [..] 005%
```

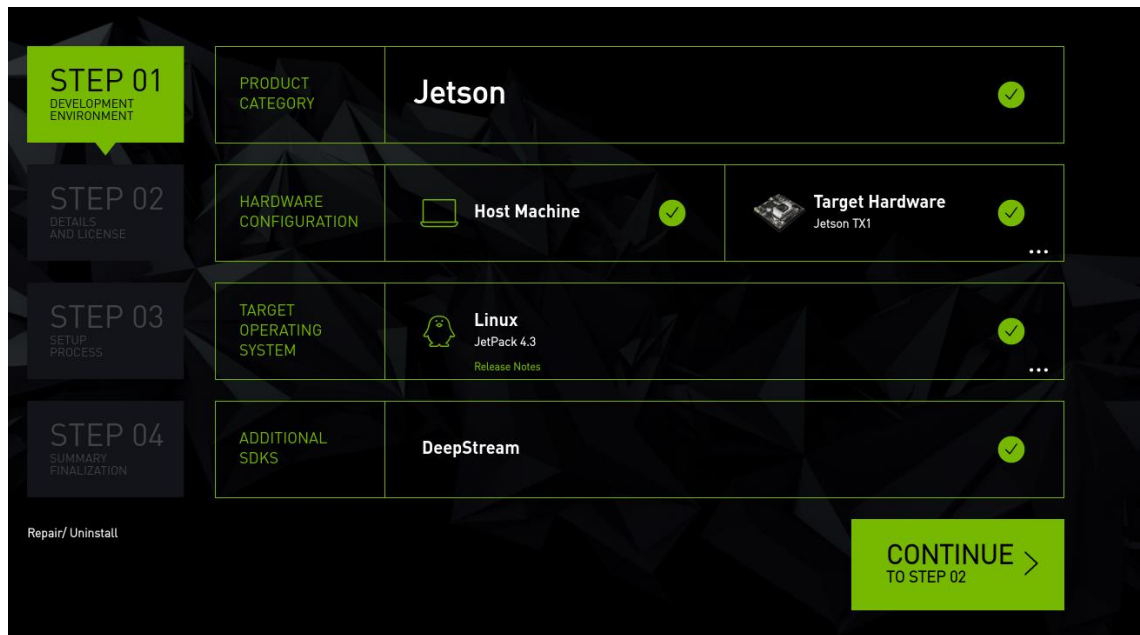
烧录成功后，会有 Successful!显示，如下图所示。

```
t.encrypt
[ 518.3405 ] Bootloader version 01.00.0000
[ 518.4021 ] Writing partition MB1_BCT with mb1_cold_boot_bct_MB1_sigheader.bct
encrypt
[ 518.4033 ] [.....] 100%
[ 518.5669 ]
[ 518.5699 ] tegradeflash_v2 --write MB1_BCT_b mb1_cold_boot_bct_MB1_sigheader
bct.encrypt
[ 518.5720 ] Bootloader version 01.00.0000
[ 518.6347 ] Writing partition MB1_BCT_b with mb1_cold_boot_bct_MB1_sigheader.b
t.encrypt
[ 518.6353 ] [.....] 100%
[ 518.7580 ]
[ 518.7581 ] Flashing completed

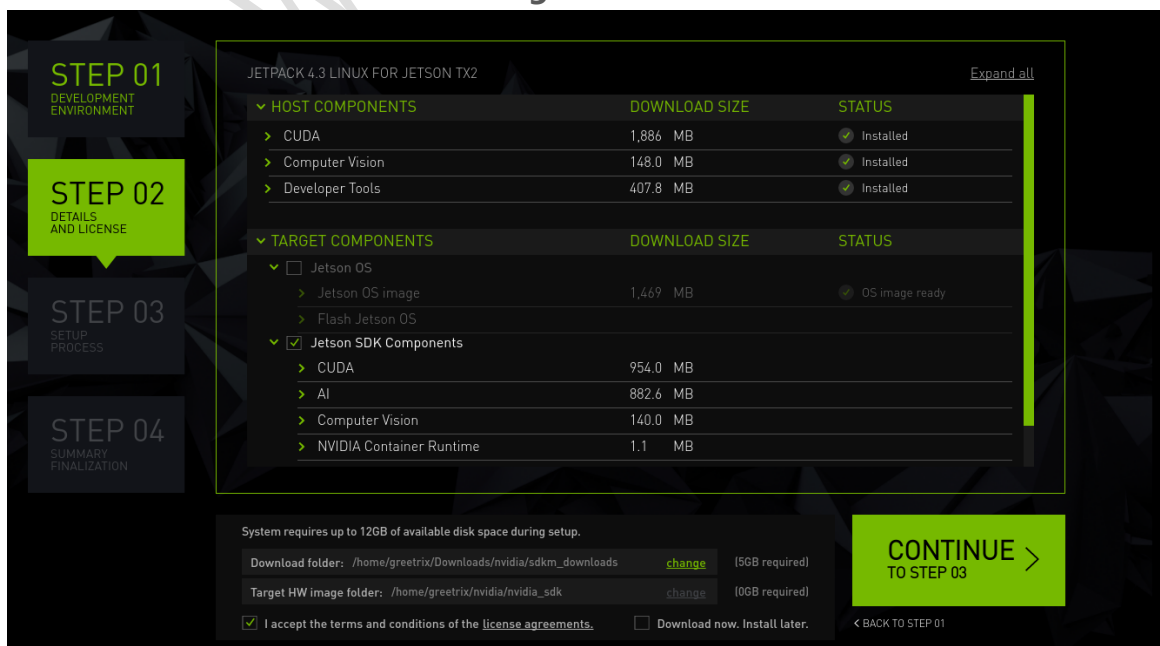
[ 518.7582 ] Coldbooting the device
[ 518.7598 ] tegradeflash_v2 --reboot coldboot
[ 518.7613 ] Bootloader version 01.00.0000
[ 518.8545 ]
*** The target t186ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
```

3. 安装系统组件(SDK)

a) 这里已 **JetPack4.3** 为例（支持最新的 DeepStream 组件），这里我们勾选 **DeepStream**。



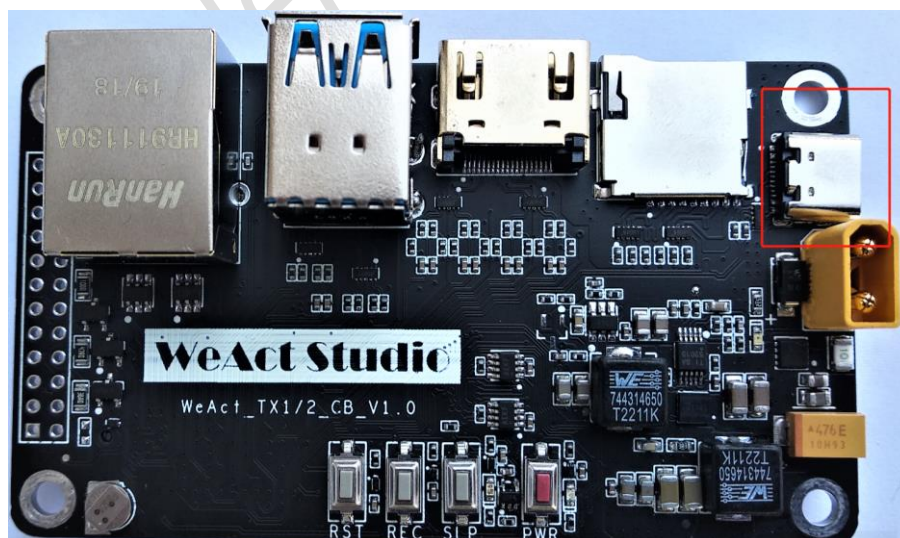
b) 这里需要勾选 **Jetson SDK Components**，并取消勾选 **Jetson OS**，勾选 **I accept the terms and conditions of the license agreements**。



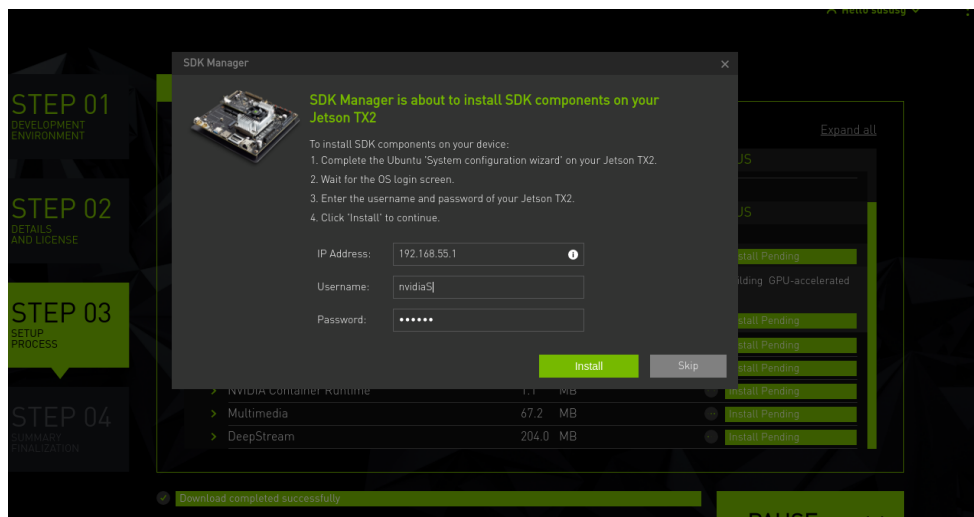
c) 等待下载完成。



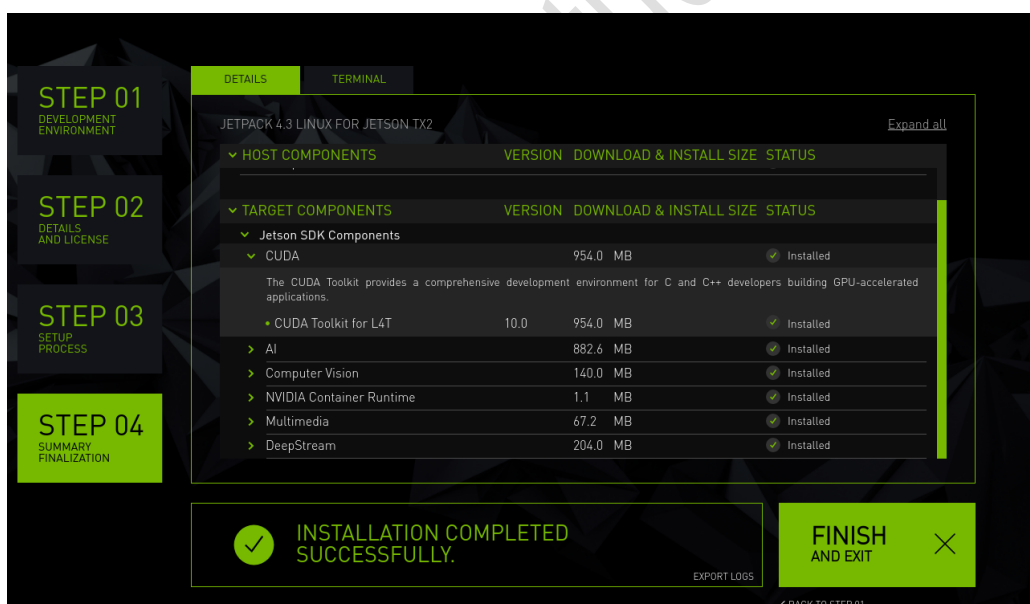
g) 使用 **USB Type-C** 线连接载板上的 **USB OTG** 接口，摁 **PWR** 正常开机，设置账号和密码，登入系统。这里使用 TX2 自带的 USB 网卡进行烧录。



d) 填入 TX2 的 Username 和 Password。点击 Install 进行 SDK 的安装。



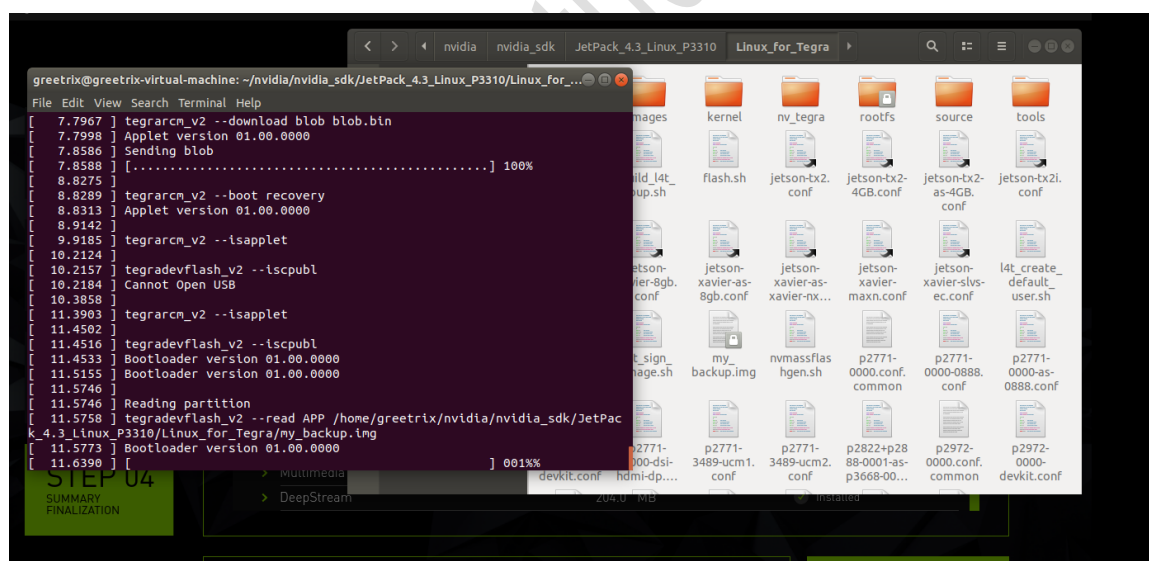
e) 等待安装完成，此时所有勾选的 SDK 已经安装在你的设备上了。



4. 镜像备份与恢复

➤ 镜像备份

- 在每次刷机前最好先备份镜像，防止刷机时不小心覆盖了镜像。首先，需要搭建烧写环境，参考教程 1。
- 参考教程 2，使 Jetson TX2 进入 Recovery 模式。
- 进入 `~/nvidia/nvidia_sdk/JetPack_3.3.1_Linux_GA_P3310/Linux_for_Tegra`，打开终端，运行命令 `sudo ./flash.sh -r -k APP -G my_backup.img jetson-tx2 mmcblk0p1` 后进行备份。备份完成后会在当前目录下生成的备份文件 `my_backup.img`。



➤ 镜像恢复

- a) 进入 `~/nvidia/nvidia_sdk/JetPack_3.3.1_Linux_GA_P3310/Linux_for_Tegra`，将 `my_backup.img` 复制一份，并且重命名为 `system.img`。
- b) 进入 `~/nvidia/nvidia_sdk/JetPack_3.3.1_Linux_GA_P3310/Linux_for_Tegra/bootloader`，打开终端【`mv system.img system_bak.img.bak`】，先把原有的系统镜像备份。
- c) 终端【`mv ../system.img system.img`】，复制生成的新镜像文件到 `bootloader` 目录。
- d) 烧录步骤参考教程 2，【重要】烧录命令：`sudo ./flash.sh -r jetson-tx2 mmcblk0p1` 多了 `-r` 参数。

5. 使用 CAN 进行通信

- a) JetsonTX2 上集成了 2 个 CAN 控制器 (CAN0/CAN1)，另外 WeAct Studio 的载板上设计了 2 个 CAN 收发器，可直接挂载 CAN 物理总线使用。
- b) TX2 自带 canbus 的驱动并集成到了镜像中，已经支持 canbus 无需多做处理。我们需要安装 canbus 模块。（在终端输入下面命令或者放入 rc.local 里面开启自启）

```
modprobe can      // 插入 can 总线子系统
modprobe can-raw  //插入 can 协议模块
modprobe can-bcm
modprobe can-gw
modprobe can_dev
modprobe mttcan   //真正的 can 口支持
```

- c) 通过 **lsmod** 检查是否安装成功。

```
nvidia@localhost:~$ lsmod
Module              Size  Used by
fuse                103841  2
mttcan              66251  0
can_dev             13306  1 mttcan
can_gw              10919  0
can_bcm             16471  0
can_raw             10388  0
can                 46600  3 can_raw,can_bcm,can_gw
zram                26166  6
overlay             48691  0
bcmdhd              934274  0
cfg80211            589351  1 bcmdhd
spidev              13282  0
nvgpu               1575721  20
bluedroid_pm        13912  0
ip_tables           19441  0
x_tables            28951  1 ip_tables
```


d) 配置 canbus 属性，和串口的波特率设置类似。

```
sudo ip link set can0 type can bitrate 500000
sudo ip link set up can0
sudo ip link set can1 type can bitrate 500000
sudo ip link set up can1
```

e) 通过 ifconfig 查看是否配置成功。

```
nvidia@localhost:~$ ifconfig
can0: flags=193<UP,RUNNING,NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 131

can1: flags=193<UP,RUNNING,NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 132
```

f) 由于载板上搭载了 CAN0/CAN1 的收发器，可以直接通过 CAN0(1)-H/L 互接进行通信 (CAN0_H 接 CAN1_H, CAN1_L 接 CAN1_L) 。

联系我们

- Github: <https://github.com/WeActTC>
- 码云: <https://gitee.com/WeAct-TC>
- 网站: <https://www.weact-tc.cn/>
- 淘宝: <https://shop118454188.taobao.com>



WeAct Studio
官方淘宝店