

WeAct Studio

NANO&XAVIER

TX2 NX 底板

使用教程

目录

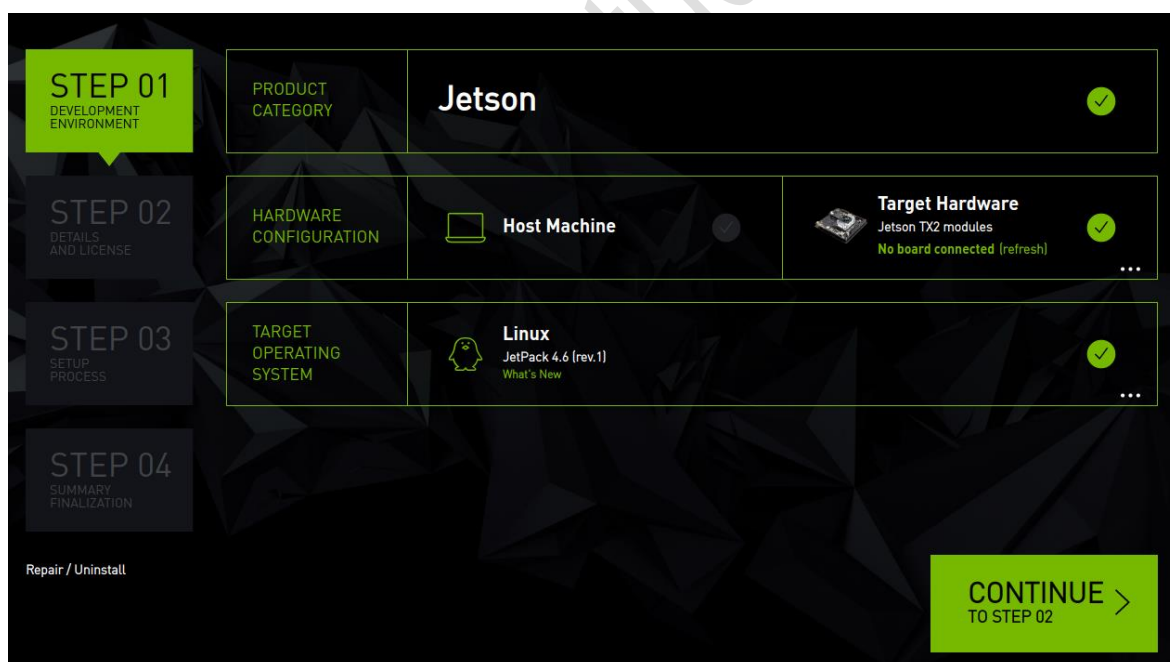
Revision History	3
1. 搭建烧写环境	4
2. 为 Nano/NX 更新设备树	7
3. 使用 CAN 进行通信	11
4. GPIO 在 shell 中使用	13
联系我们	14

REVISION HISTORY

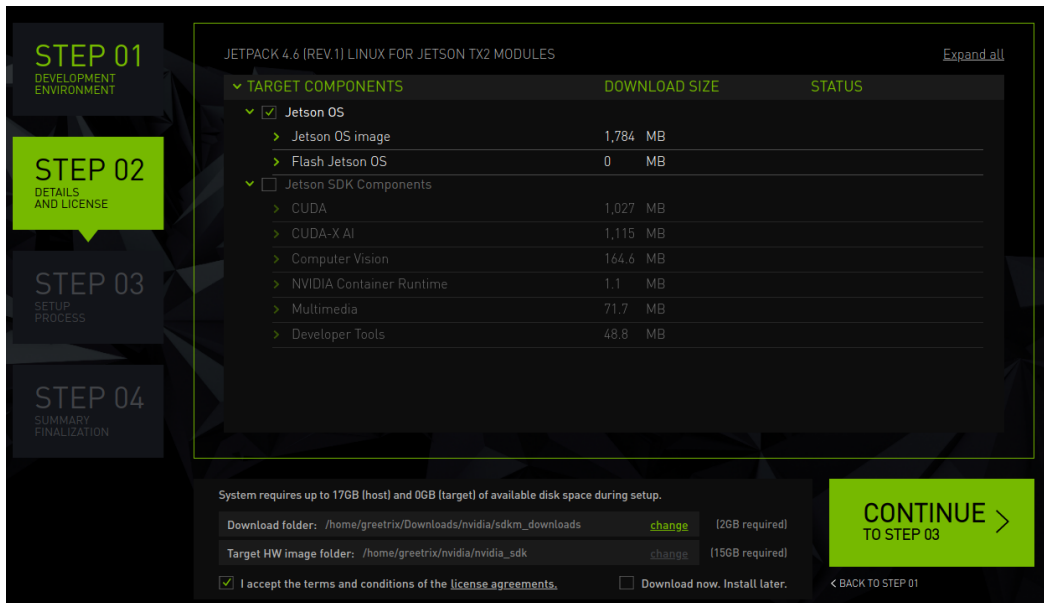
Draft Date	Revision	Description
2021.9.20	V1.0	1. 初始版本

1. 搭建烧写环境

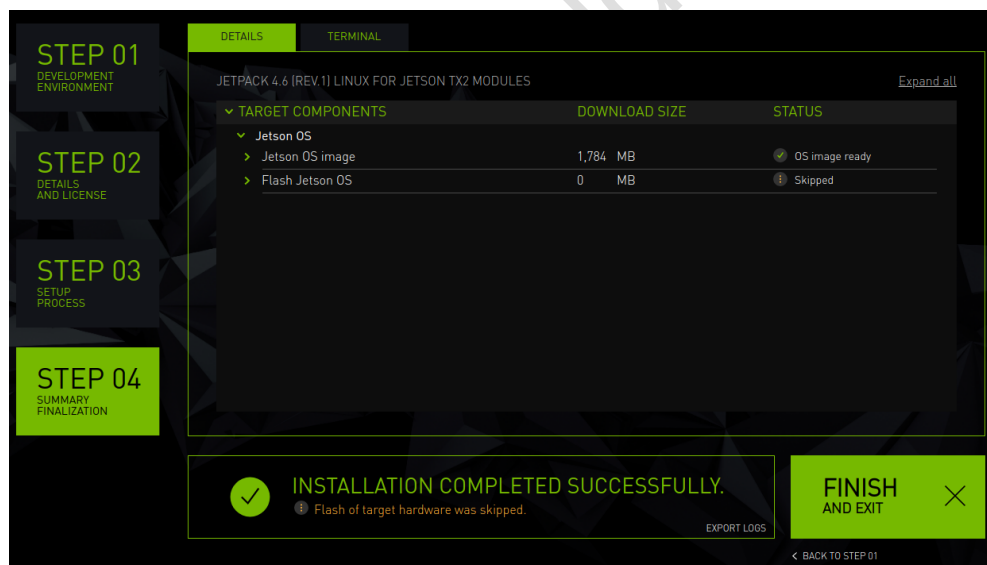
- a) 首先，需要一台装有 **Ubuntu16.04** 以上的电脑作为 HOST 端给 Nano/NX 烧写，或者可以在 Windows 上安装 VMware 来实现。
- VMware 上如何安装 Ubuntu18.04:
<https://blog.csdn.net/u012556114/article/details/82751089>
- b) 在 NVIDIA 下载最新的 **SDK-Manager** 并在 ubuntu18.04 中安装（需要注册一个 NVIDIA 账号，后面也需要用到）
- SDK-Manager 下载地址: <https://developer.nvidia.com/nvidia-sdk-manager>
- c) 选择需要 **Target Hardware** 以及 **JetPack** 版本，**不勾选 HostMachine**，这里以 **TX2NX** 为例选择，点击 **Continue**



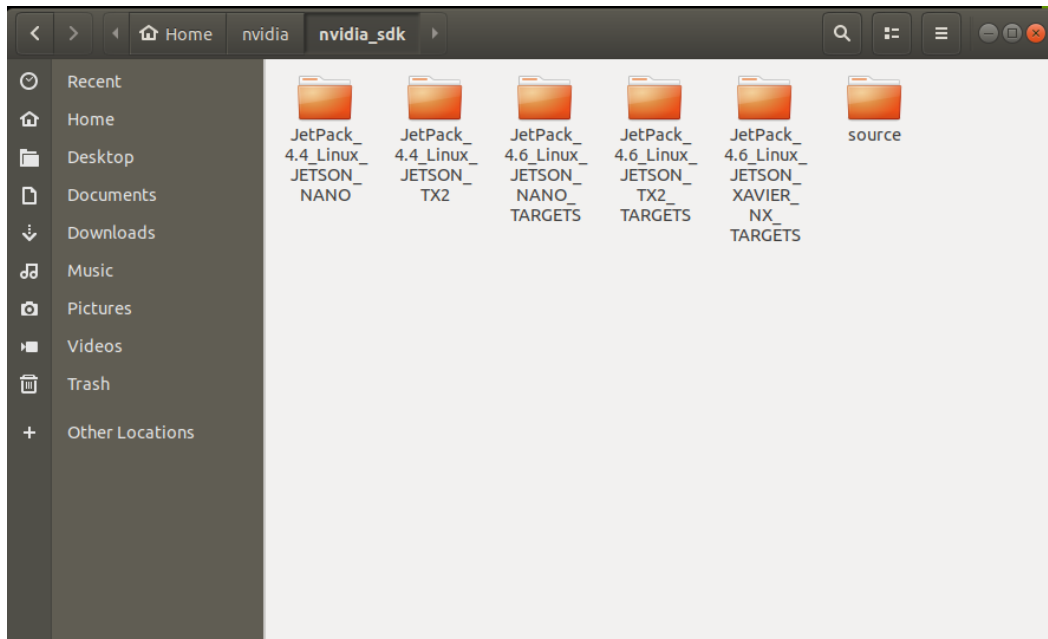
- d) 这里需要勾选 **I accept the terms and conditions of the license agreements**，取消勾选 **Jetson SDK Components**，点击 **CONTINUE** 进行下一步。



- e) 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，安装过程中会弹出联网烧写的信息，选择 **Skip**。



- 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 NANO/NX 更新设备树

P.S: WeAct 设备树与官方设备树区别（其他功能相同），如果无需求，可以不更新设备树。

!!! 注意，更新设备树不影响系统任何文件，请放心更新

NVIDIA 与 WeAct 设备树差异

	NVIDIA	WeAct Studio
Nano-SD	相同	相同
Nano-EMMC	无法使用 SD 卡	可以使用 SD 卡
TX2NX	无法使用 SD 卡&UART1	可以使用 SD 卡&UART1
XavierNX	无法使用 SD 卡	可以使用 SD 卡

a) 这里以 **TX2NX** 为例，在 WeAct Studio 的 **github** 或者**码云**上下载相应的设备树文件。

➤ Github: https://github.com/WeActTC/Nano_TX2-Xavier_NX-CB

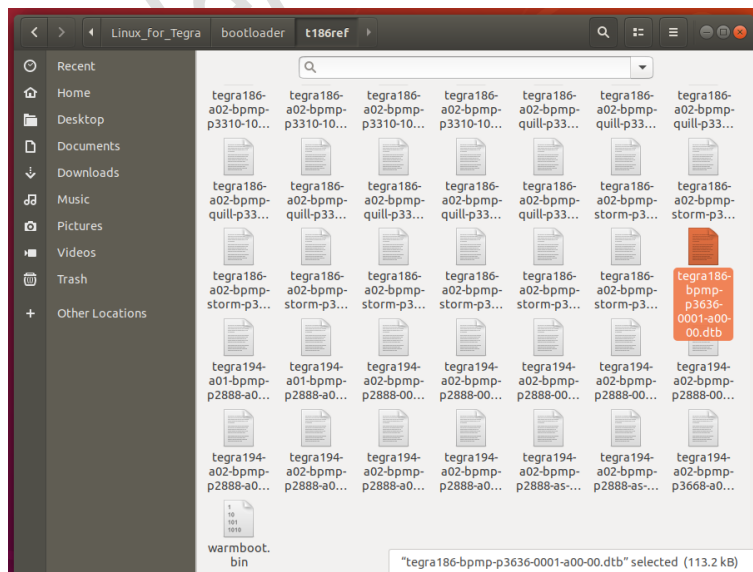
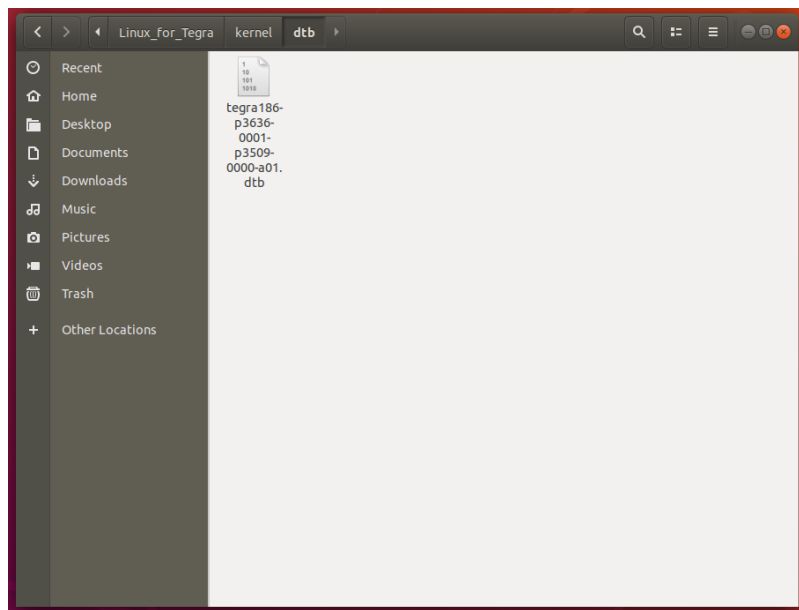
➤ 码云: https://gitee.com/WeAct-TC/Nano_TX2-Xavier_NX-CB

各设备设备树更新路径及设备树名称

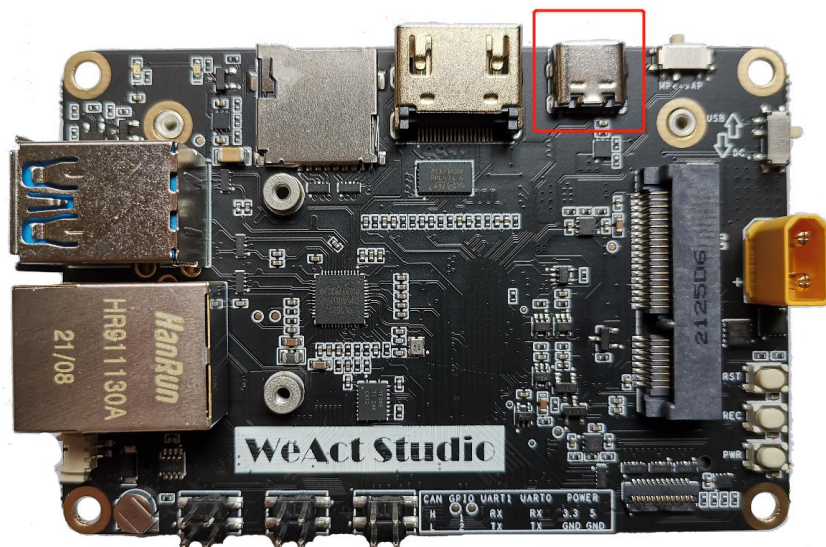
	Linux_for_Tegra/kernel/dtb	Linux_for_Tegra/bootloader/t186ref(t210f)
Nano-EMMC	tegra210-p3448-0002-p3449-0000-b00	无
TX2NX	tegra186-p3636-0001-p3509-0000-a01	tegra186-bpmp-p3636-0001-a00-00
XavierNX	tegra194-p3668-all-p3509-0000	无

b) 找到相应版本的设备树

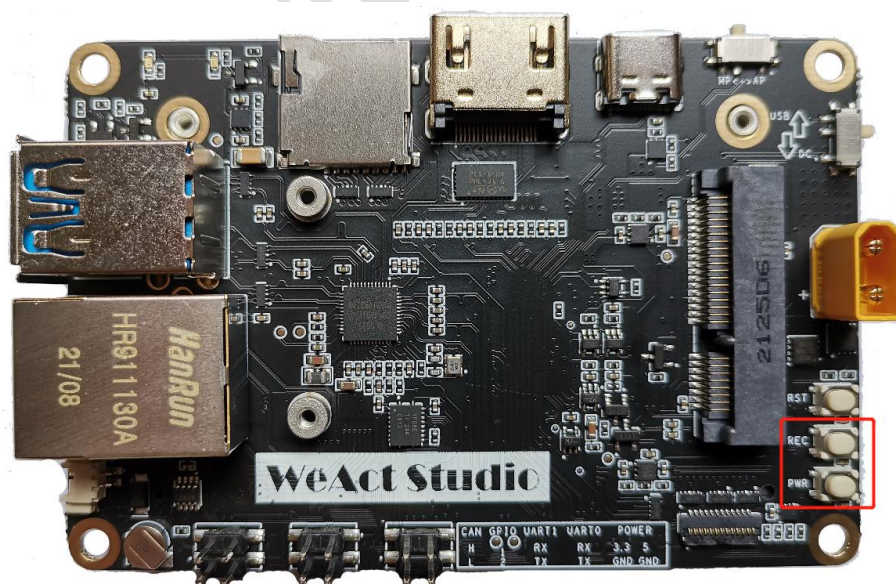
1. 进入 `~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra/kernel/dtb`，复制提供的设备树 `tegra186-p3636-0001-p3509-0000-a01.dtb` 至该目录
2. 进入 `~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra/bootloader/t186ref`，复制提供的设备树 `tegra186-bpmp-p3636-0001-a00-00.dtb` 至该目录【**仅 TX2NX 需要更新**】



3. 使用 **USB Type-C** 线连接载板上的 **USB OTG** 接口。



4. 将开机键拨至 **MP** (手动开机) , 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.**



5. 进入 `~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra`, 打开终端, 运行命令 `sudo ./flash.sh -r -k kernel-dtb jetson-xavier -nx-devkit-tx2-nx mmcblk0p1`, 等更新成功就可以使用了。

各设备更新设备树命令

设备	设备树更新命令
Nano-SD	<code>sudo ./flash.sh -r -k DTB jetson-nano-qspi-sd mmcblk0p1</code>
Nano-EMMC	<code>sudo ./flash.sh -r -k DTB jetson-nano-emmc mmcblk0p1</code>
TX2-NX	<code>sudo ./flash.sh -r -k kernel-dtb jetson-xavier-nx-devkit-tx2-nx mmcblk0p1</code>
Xavier-SD	<code>sudo ./flash.sh -r -k kernel-dtb jetson-xavier-nx-devkit-qspi mmcblk0p1</code>
Xavier-EMMC	<code>sudo ./flash.sh -r -k kernel-dtb jetson-xavier-nx-devkit-emmc mmcblk0p1</code>

更新设备树后，会有 Successfully!显示，如下图所示。

```

File Edit View Search Terminal Help
[ 11.1401 ] tegradevflash_v2 --iscpubl
[ 11.1423 ] Cannot Open USB
[ 11.9533 ]
[ 12.9584 ] tegrarcm_v2 --isapplet
[ 13.2306 ]
[ 13.2341 ] tegradevflash_v2 --iscpubl
[ 13.2354 ] Bootloader version 01.00.0000
[ 13.3996 ] Bootloader version 01.00.0000
[ 13.4611 ]
[ 13.4611 ] Writing partition
[ 13.4647 ] tegradevflash_v2 --write kernel-dtb 1_kernel_tegra186-p3636-0001-p3
509-0000-a01_sigheader.dtb.encrypt
[ 13.4676 ] Bootloader version 01.00.0000
[ 13.6334 ] Writing partition kernel-dtb with 1_kernel_tegra186-p3636-0001-p350
9-0000-a01_sigheader.dtb.encrypt
[ 13.6352 ] [.....] 100%
[ 13.7256 ]
[ 13.7259 ] Coldbooting the device
[ 13.7283 ] tegradevflash_v2 --reboot coldboot
[ 13.7306 ] Bootloader version 01.00.0000
[ 13.9214 ]
*** The [kernel-dtb] has been updated successfully. ***

```

3. 使用 CAN 进行通信

- a) Tx2-NX/XavierNX 上集成了 2 个 CAN 控制器 (CAN0/CAN1)，另外 WeAct Studio 的载板上设计了 1 个 CAN 收发器 (CAN0)，可直接挂载 CAN 物理总线使用。
- b) Tx2-NX/XavierNX 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。

- e) 通过 ifconfig 查看是否配置成功。

- f) 在一个终端通过 `cansend can0(can1) xxx` 命令来发送数据, 另一个终端通过 `candump can1(can0)` 完成实际信号收发测试

4. GPIO 在 SHELL 中使用

a) Nano/TX2-NX/Xavier-NX 可直接通过 shell 命令控制 GPIO 输入输出

	GPIO1	GPIO2
Nano	194	38
TX2-NX	338	269
Xavier-NX	196	105

b) 以 TX2-NX GPIO1 为例

- 先激活 IO: `sudo echo 338 > /sys/class/gpio/export`
- 设置 IO 方向: `echo out > /sys/class/gpio/gpio338/direction`
- 设置输出: `echo 1 > /sys/class/gpio/gpio338/value`

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