

Jetson Nano Developer Kit

使用指南



版本说明¹

Version	Date	Authors	Description of Change
1.0	March 18, 2019	plawrence	初版

=

说明

欢迎使用 NVIDIA Jetson 平台。使用本平台前，您需要先做两件事：

1. 注册 [NVIDIA Developer Program](#) 账号-注册之后，您就可以在 [NVIDIA Jetson Forums](#)（论坛）上提问以及回答，并且可以在 [Jetson Download Center](#)（资料下载中心）上访问所有的资料，获取更多信息和资源
2. 仔细阅读本手册。这里还有一些重要的链接提供参考：
 - [Jetson FAQ](#) – 请仔细阅读这里的常见问题
 - [Support Resource](#) – 这个网页提供了各个重要资源的链接，包括 Jetson 论坛和 Jetson 生态系统网页链接
 - [L4T Release Note](#) – L4T 是 Jetson 平台的关键部分，为你的开发套件提供了示例文件系统。具体可以查阅最新发布的说明

The NVIDIA Jetson Team

¹ 【译者注】此手册由深圳市微雪电子基于 NVIDIA 官方提供的 Jetson Nano Developer Kit user Guide V1.0 版本整理的中文翻译版，仅做学习交流用途，不对文档的完整性，准确性和内容保证或者承担任何责任。更多资料，请关注深圳市微雪电子有限公司官方网站以及微雪百科

目录

开发套件设置以及硬件	4
准备.....	4
开发套件原装箱中包含	4
接口说明.....	5
接口信息.....	5
模块.....	5
底板.....	5
供电说明.....	7
Micro-USB 供电.....	7
其他供电.....	7
功耗预算.....	7
JETPACK.....	8
JetPack 组件.....	8
系统镜像.....	8
函数库和 APIs.....	8
示例应用程序.....	8
开发工具.....	9
文档.....	9
JetPack 安装步骤.....	10
使用 L4T.....	11
认证信息（略去）	12

开发套件设置以及硬件

英伟达 (NVIDIA®) Jetson Nano™ Developer Kit 是一个小型的人工智能计算机，是一个适用于创客，学习者以及开发人员的低功耗，易上手的 AI 硬件平台。为使用者提供了一些入门级的通用外设，外围模块以及示例工程。

Jetson Nano 适用 NVIDIA 的 JetPack™ SDK，具有运行现代 AI 工程所需要的性能和功能。JetPack 所需的器件：

- 装有 NVIDIA 驱动的桌面版 Linux 系统²
- AI 和计算机视觉库以及 APIs
- 开发工具（比如 Jetson Nano 和相关外围模块）
- 文档和例程

准备

在使用开发套件之前，你还需要准备一张 SD 卡，SD 卡要烧写好操作系统，以及 JetPack 工程所需的其他组件。一个最简单的方法就是直接下载官方提供的镜像文件，并且按照 [Getting Started with Jetson Nano Developer Kit](#) 教程操作。

大概步骤：

- 你需要准备一张 16G（至少）的 SD 卡，一个 HDMI（或者 DP）屏幕³，一个 USB 键盘和鼠标。一个 5V/2A 的 micro USB 接口的电源适配器
- 下载镜像并将镜像烧写到 SD 卡上
- 将 SD 卡插入到 Jetson Nano 模块背面的 SD 卡卡槽，然后将屏幕，键盘鼠标，网线（或者无线网卡）连接好

关于具体方法，可以查看 [JetPack 安装步骤](#) 部分

开发套件原装盒中包含

- Jetson Nano 模块⁴
- 开发套件底板
- 一张小的说明卡（快速手册）
- 一个纸质的支架

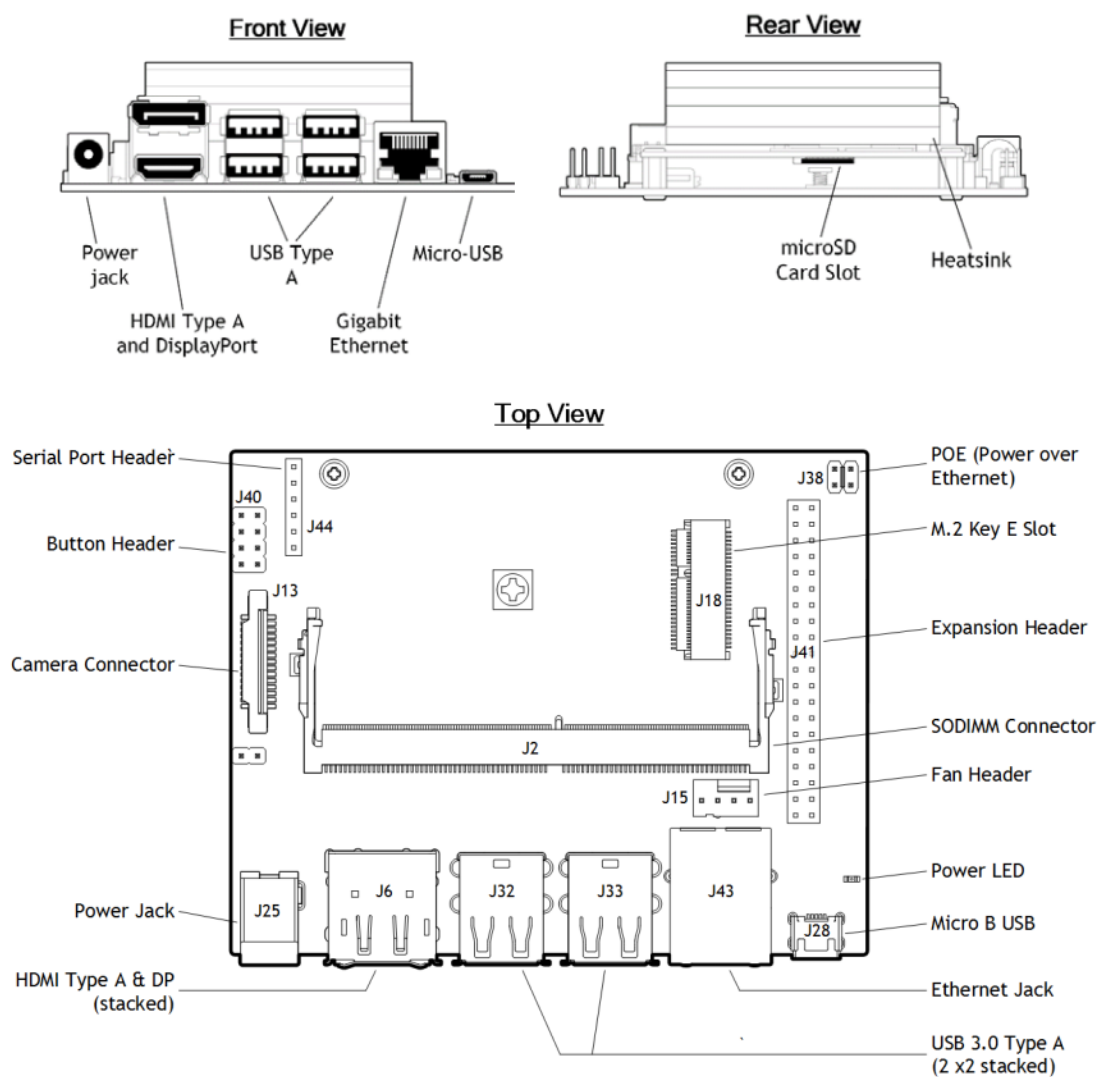
² 【译者注】官方提供有一个 Ubuntu 系统，可以直接使用

³ 【译者注】微雪电子有几款 HDMI 屏幕可以支持 Jetson Nano,有需要可以咨询微雪电子销售人员

⁴ 【译者注】实际拿到手的时候，Jetson Nano 模块跟底板以及散热板是组装好的，所以你看到的是完整的开发套件。

接口说明

开发套件模块以及底板：



接口信息

这部分主要列举开发套件上的一些接口，如果想要查看更加完整的硬件详细信息，可以查看 Jetson Nano Developer Kit Carrier Board Specification 部分

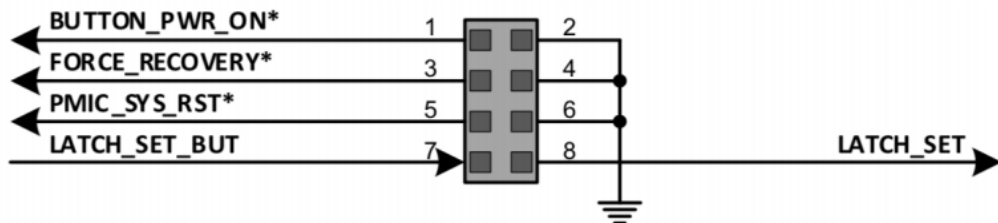
模块

- [J501] SD 卡卡槽
- 无源散热板，可以支持 10W 的模块在 25°C 左右的环境温度下使用。如果你有额外的散热需求，你可以另外连接控制系统风扇。查看 Jetson Nano Supported Component List 了解官方验证的可以支持的风扇型号

底板

- [DS3] 电源指示灯。当开发套件正常供电时会亮起

- [J2] SO-DIMM 连接器，用来连接 Jetson Nano 模块
- [J6] HDMI 和 DP 接口
- [J13] 摄像头接口，可以连接使用 CSI 摄像头。Jetson Nano 开发条件支持 IMX219 摄像头模块，包括 Leopard Imaging 的 LI_IMX219-MIPI-FF-NANO 摄像头模块以及树莓派的 Camera Module V2 模块
- [J18] M.2 Key E 连接器，可以用来连接无线网卡，包括有 PCIe(x1)接口， USB2.0， UART， I2S 和 I2C 接口
- [J25] DC 电源接口，支持 5V/4A 的 DC 电源。接口规格是 2.1x5.5x9.5mm
- [J28] Micro USB 2.0 接口，有两种使用方式：
 - 如果 J48⁵ 引脚没有连接的话，你可以把这个接口当做一个供电接口使用，可以接入 5V/2A 的 USB 电源供电
 - 如果说 J48 引脚被连接上的话，这个接口可以作为设备模式使用
- [J32 和 J33]这里是 4 个 USB3.0 Type A 接口。每组 USB 接口（每组两个）最高可以输出 1A 的电流。所有 4 个 USB 接口通过底板上集成的一个 USB3.0 的 HUB 连接到 Jetson Nano 模块上
- [J38] PoE 接口，能够支持 J43 的以太网口（IEEE 802.3af 协议）对套件进行供电
- [J40] 8PIN 的按键排针。提供一些比如说系统电源，复位，强制重启的信号（可以查看的图像）



- 引脚 7 和 8 禁用自动上电功能
 - 引脚 1 和 2 启动上电（在自动上电功能被禁用的时候有效）
 - 引脚 5 和 6 可以复位系统
 - 引脚 3 和 4 设置开发套件进入 Force Recovery 模式，如果你在上电的情况下连接模块的话
- [J41] 40PIN 外扩引脚

⁵ 【译者注】上面的接口图没有标出 J48，这个是 CSI 摄像头接口下方的 2PIN 的那个排针

- 电源引脚

两个 3.3V 引脚，两个 5V 引脚。只要开发套件上电，这些引脚就会有电压输出，不可以被单独关闭

两个 5V 引脚每个最高可以输出 3A 的电流

- 接口信号脚

所有的引脚的逻辑电平都是 3.3V

默认情况下，所有的接口信号脚都被设置为 GPIOs，除了 PIN3 和 PIN5，PIN27 和 PIN28（两个 I2C 接口），PIN8 和 PIN10（UART 接口）。L4T 提供了一个 Python 库，Jetson.GPIO，可以用来控制 GPIOs，这个库的 API 跟 RPi.GPIO⁶是一样的，具体你可以查看 Jetson 系统里面的/opt/nvidia/jetson-gpio/doc/README.txt 文件

- [J43] 千兆以太网口

- [J44] 3.3V 串口接口，可以做串口调试终端⁷

- [J48]这个接口可以使能两个供电接口，在没有连接的情况下，开发套件使用 micro USB 接口作为供电接口，在连接（使用跳线帽或者排线）连接的情况下，可以使用 DC 电源接口进行供电

供电说明

Jetson Nano 开发套件需求 5V/2A（至少）的一个供电环境

Micro-USB 供电

拆箱后，开发套件默认是在 micro-USB 供电模式下。有些适配器为了避免受到线材损耗的影响，实际输出的电压是高于 5V 的，比如说 Adafruit 的 GEO151UB-6025 电源，就是实际输出 5.25V 的。需要注意 Jetson Nano 正常工作需要至少 4.75V 电压供电，所以如果你有线材损耗顾虑的话，建议购买高质量的电源或者电压稍高一点的电源

其他供电

如果说，你连接的外设模块比较多，导致总电流需求大于 2A 的话，你可以选择连接 J48 引脚，切换成 DC 电源供电，然后选用 5V 4A 的 DC 电源供电。另一个选择就是通过 J41 上的 5V 供电脚输出 5V、6A 的电源供电（每个引脚 3A）

DC 电源接口的深度为 9.5mm，可以使用外径 5.5，内径 2.1 的 DC 电源

功耗预算

开发套件的总功耗是根据由 Jetson Nano 模块，底板以及你连接的所有外设所决定的

在没有连接其他外设的情况下，底板的功耗在 0.5W（2A）到 1.25W（4A）之间

Jetson Nano 模块采用电源效率优化设计，支持两种功耗模式（软件可调），默认模式下，是大约 10W 的一个功耗预算方案，另一个是 5W 的功耗预算方案。这两个模式，通过将 GPU，GPU

⁶ 【译者注】这个是树莓派的 GPIO 库

⁷ 【译者注】这个接口的波特率是 115200

频率和 GPU 使用数量限制在一个预测水平内，来将功耗保持在 5W 和 10W 的范围内。关于功耗模式的具体信息，可以查看 [L4T Development Guide](#)。

注意，功耗预算是包括 Jetson 模块的两个主要功耗领域

Nano 模块：GPU (GPU_VDD) 和 CPU (CPU_VDD)。CORE 的各个功耗区域 (VDD_SOC)，比如说视频封装，视频解码是不包括在功耗预算中的。

预算方案能够将功耗限制在一定范围内，但是这并非精准的总功耗数值，实际功耗是跟你接入的外设和实际使用情况决定的。关于功耗的一个详细信息，可以查阅 [Jetson Nano module Data Sheet](#)。

实际功耗根据你接入的外设以及使用情况而定，所以在挑选电源的时候需要考虑一下你的实际使用情况。

JETPACK

NVIDIA JetPack SDK 是为构建 AI 应用程序提供最全面的解决方案，它包含了 Jetson 产品所需的最新系统镜像，各种函数库以及 APIs，示例程序，开发工具以及文档。

JetPack 组件

这一部分简单介绍 JetPack 中的每个组成部分。你可以参阅 JetPack 的在线文档，了解更多详情：<https://docs.nvidia.com/jetson/jetpack/index.html>

系统镜像

JetPack 包括了一个由 Ubuntu 系统派生的操作系统和参考文件系统

函数库和 APIs

JetPack 函数库和 APIs 包括：

- TensorRT 和 cuDNN：用于高性能深度学习项目
- CUDA：GPU 加速，这个是多种领域都涉及到的
- 多媒体 API 包：摄像头应用和传感器驱动开发
- VisionWorks 和 OpenCV：虚拟计算应用

示例应用程序

JetPack 包含了多个示例程序，用来演示 JetPack 组件的使用。这些示例程序保存在参考文件系统中，可以在开发套件上编译使用

JetPack 组件	例程在文件系统中的路径
TensorRT	/usr/src/tensorrt/samples/
cuDNN	/usr/src/cudnn_samples_<version>/
CUDA	/usr/src/cuda-<version>/samples
Multimedia API	/usr/src/tegra_multimedia_api/
VisionWorks	/usr/share/visionworks/sources/samples /usr/share/visionworks-tracking/sources/samples

	/usr/share/visionworks-sfm/sources/samples/
OpenCV	/usr/share/OpenCV/samples

开发工具

下面列举了 JetPack 包含的开发工具，这些工具有些是直接使用在 Jetson 系统上的，有一些是使用在跟 Jetson 系统连接的 Linux 主机上的

- 应用开发和调试工具：
 - [Nsight Eclipse Edition](#)：这个工具用来开发 GPU 加速应用，这个是运行在 Linux 主机上的，可支持所有的 Jetson 产品
 - [CUDA-GDB](#)：这个工具是用于应用调试的，运行在 Jetson 系统或者 Linux 主机，支持所有的 Jetson 产品
 - [CUDA-MEMCHECK](#)：用来调试应用内存错误的，运行在 Jetson 系统。支持所有的 Jetson 产品
- 应用分析和优化工具：
 - [Nsight System](#)：用于 GPU 和 CPU 之间的应用分析，运行在 Linux 主机，支持所有 Jetson 产品
 - [nvprof](#)：用于 GPU 和 CPU 之间的应用分析，运行在 Jetson 系统，支持所有 Jetson 产品
 - [Visual Profiler](#)：用于 GPU 和 CPU 之间的应用分析，运行在 Linux 主机，支持所有 Jetson 产品

注意：相对于 [Visual Profiler](#)，我们推荐开发者使用 [Nsight Systems](#) 或者 [Nsight Compute](#)

- [Nsight Graphics](#)：用于图形应用的分析和调试，运行在 Linux 主机，支持所有 Jetson 产品
- [Nsight Compute](#)：用于交互式 CUDA 内核分析，运行在 Linux 主机，支持所有 Jetson 产品
- [Nsight Compute CLI](#)：CUDA 内核分析，运行在 Linux 主机，支持 Jetson AGX Xavier

文档

JetPack 相关的文档包括：

- [JetPack Documentation](#)
- [NVIDIA Linux Driver Package Development Guide](#)
- [Tegra L4T Release Notes](#)
- [TensorRT Documentation](#)
- [cuDNN Documentation](#)
- [CUDA ToolKit](#)
- [Multimedia API Reference](#)
- [VisionWorks Documentation](#)
- [OpenCV Documentation](#)
- [Nsight Eclipse Edition Documentation](#)
- [CUDA-GDB Documentation](#)

- [CUDA-MEMCHECK Documentation](#)
- [Nsight Systems](#)
- [nvprof](#)
- [Visual Profiler](#)
- [Nsight Graphics](#)
- [Nsight Compute CLI](#)
- [Nsight Compute](#)

JetPack 安装步骤

这里有两种方式将 JetPack 安装到你的开发套件上

- 使用 SD 卡镜像（Jetson Nano 直接使用这个方法）
按照 [Getting Started with Jetson Nano Developer Kit](#) 下载系统镜像，然后烧写到 SD 卡上。将 SD 卡插入到开发套件就可以直接启动了
- 使用 NCIDIA SDK Manager
下载 SDK Manager 到 Linux 主机，然后使用这个工具将系统烧写到你的开发套件上，或者安装其他的 JetPack 组件。SDK Manager 同时还可以设置搭建好 Linux 主机的开发环境。

注意	<p>使用 SDK Manager 来安装 JetPack 必须满足以下条件</p> <ul style="list-style-type: none">● 开发套件处于 Force Recovery 模式● 开发套件不能用 micro USB 接口来供电，因为 micro USB 接口需要用来烧写和更新开发套件
-----------	--

在使用 SDK Manager 之前，要先上电开发套件，并且将套件设置成 Force Recovery 模式：

1. 将 J40 按键引脚上的 Pin3 和 Pin4 用跳线帽或者杜邦线连接起来
2. 将 J48 供电选择引脚用跳线帽或者杜邦线连接起来，将供电接口切换到 J25 DC 接口供电。
上电后，模块就会自动进入 Force Recovery 模式
3. 等到开发套件正常运行之后，去掉 Force Recovery 引脚上的跳线帽或者杜邦线
4. 参阅 [SDK Manager documentation](#) 进行下一步

使用 L4T

NVIDIA L4T (JetPack 中的操作系统组件) 提供了 Linux 内核, Bootloader, 硬件支持包 (BSP) 以及 Jetson 开发套件的例程文件系统。这些都可以在 Jetson Developer Site(Jetson 开发者网站)的 [L4T 主页](#)下载。

使用前请先阅读 [L4T Development Guide](#) -这个是非常重要的资源

认证信息 (略去)