目录

准备工作 2
配件需求 2
烧写镜像 2
硬件连接 2
软件配置 3
简单入门 3
Jetson.GPIO 3
环境配置 4
例程测试 5
摄像头 7
Jetson-Inference 8

准备工作

配件需求

1. Jetson Nano Developer Kit
2. 16G 以上 SD 卡
3. HDMI (或者 DP)显示器，这里我们使用 7inch HDMI LCD (H) (带外壳) 为例
4. 5V/2A 电源
5. 键盘和鼠标
6. MIPI-CSI 摄像头，这里使用的是 IMX219-77 Camera
1. 无线网卡

烧写镜像

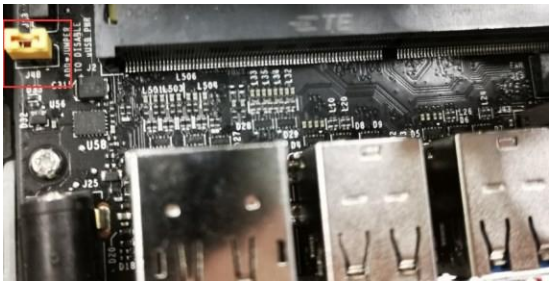
1. 从 NVIDIA 官方网站下载 Jetson Nano 镜像
2. 使用 Win32 Disk Imager 或者 Ether 软件将镜像烧写到 SD 卡中

硬件连接

1. 连接无线网卡到 Jetson Nano
   -如果使用 M.2 KEY E 接口的无线网卡，需要先查下 Jetson Nano 核心板，然后接入无线网卡，再重新组装好核心板

   -如果使用 USB 无线网卡，直接将无线网卡插入到 Jetson Nano 的 USB 接口即可
2. 连接 7inch HDMI LCD (H) 的 Touch 接口到 Jetson Nano 的 USB 接口
3. 连接 7inch HDMI LCD (H) 的 HDMI 接口到 Jetson Nano 的 HDMI 接口
4. 连接键盘和鼠标到 Jetson Nano 的 USB 接口
5. 连接 IMX219-77 Camera 到 Jetson Nano 的 CSI 摄像头接口。注意金属面朝向散热板一侧
6. 将烧写好镜像的 SD 卡插入到 Jetson Nano 的 SD 卡卡槽
   -卡槽在 Jetson Nano 核心板的背面，主要这是个按压式的卡槽，插入时检查一下 SD 卡是否已经插紧
7. 插入电源上电启动 Jetson Nano
   - 如果使用的是 USB 电源，直接接入到开发套件的 micro USB 接口即可
   - 如果使用的是 DC 电源，使用一个跳线帽，短接 J48 接口，然后将电源插入到开发套件的

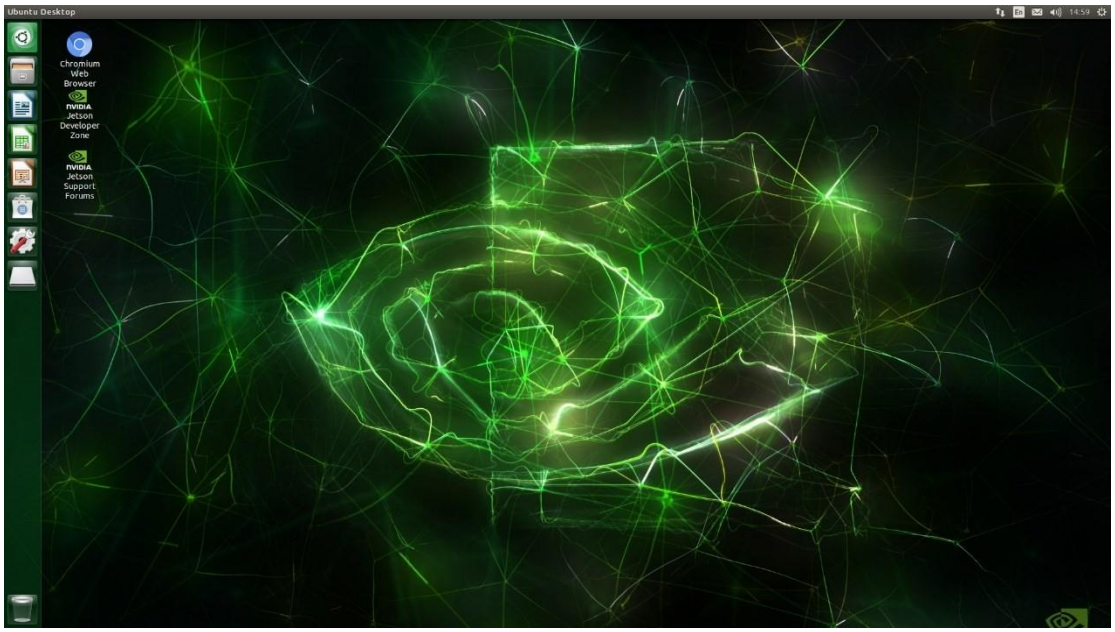
DC 电源接口



软件配置

第一次启动 Jetson nano, 需要先设置并安装 Jetpack 插件，请按照提示完成配置。

安装完成后就可以看到 NVIDA logo 桌面



简单入门

本章节简单讲一下如何使用 Jetson nano 开发套件。

系统中没有 nano 编辑器，如果不习惯 vi 的用户可以自己手动装一下

nano sudo apt-get install nano

JETSON.GPIO

Jetson Nano 开发套件引出了跟树莓派类似的 40PIN 排针，官方提供了一个 jetson.GPIO 库来调用这些引脚。这里我们简单介绍一下如何使用这个库

关于 Jetson.GPIO，这里有详细说明：

https://pypi.org/project/Jetson.GPIO/

或者

https://github.com/NVIDIA/jetson-gpio

环境配置

1. 下载 jetson-gpio：

   git clone https://github.com/NVIDIA/jetson-gpio

```
waveshare@waveshare-desktop:~$ git clone https://github.com/NVIDIA/jetson-gpio
Cloning into 'jetson-gpio'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 47 (delta 20), reused 40 (delta 13), pack-reused 0
Unpacking objects: 100% (47/47), done.
waveshare@waveshare-desktop:~$
```

2. 将下载的文件移动到目录：/opt/nvidia 中

   sudo mv jetson-gpio /opt/nvidia/

```
waveshare@waveshare-desktop:~$ ls
Desktop  Documents  Downloads  examples.desktop  jetson-gpio  Music  Pictures  Public  Templates  Videos
waveshare@waveshare-desktop:~$ cd jetson-gpio/
waveshare@waveshare-desktop:~/jetson-gpio$ ls
etc  lib  LICENSE.txt  MANIFEST.in  README.md  samples  setup.py
waveshare@waveshare-desktop:~/jetson-gpio$ cd
waveshare@waveshare-desktop:~$ sudo mv jetson-gpio/ /opt/nvidia/
mv: cannot move 'jetson-gpio/' to '/opt/nvidia/jetson-gpio': Directory not empty
waveshare@waveshare-desktop:~$ sudo mv /opt/nvidia/jetson-gpio/ /opt/nvidia/jetson-gpio_bak/
waveshare@waveshare-desktop:~$ sudo mv jetson-gpio/ /opt/nvidia/
waveshare@waveshare-desktop:~$
```

   注意：这里提示说 jetson-gpio 文件夹已存在，所以我们先把里面的 jetson-gpio 文件夹重新命名成 jetson-gpio_bak 然后再把库复制进去

3. 安装 pip3 工具：

   sudo apt-get install python3-pip

4. 进到 jetson-gpio 文件夹，并安装库：

   cd /opt/nvidia/jetson-gpio

   sudo python3 setup.py

   install

5. 使用前，还需要创建一个 gpio 组，把你的当前的账号加到这个组，并赋予使用权限

   sudo groupadd -f -r gpio

   sudo usermod -a -G gpio your_user_name

   sudo cp /opt/nvidia/jetson-gpio/etc/99-gpio.rules

   /etc/udev/rules.d/ sudo udevadm control –reload-rules && sudo

   udevadm trigger

```
waveshare@waveshare-desktop: /opt/nvidia/jetson-gpio
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio$ sudo groupadd -f -r gpio
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio$ sudo usermod -a -G gpio waveshare
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio$
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio$ sudo cp /opt/nvidia/jetson-gpio/etc/99-gpio.rules /etc/udev/rules.d/
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio$ sudo udevadm control --reload-rules && sudo udevadm trigger
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio$
```

   注意：your_user_name 是你使用的用户名，比如说 waveshare

例程测试

环境配置好了之后就可以测试一下例程了。在 jetson-gpio 上提供了几个简单的例程我们可以简单测试一下，先进入示例程序目录

cd ~/opt/nvidia/jetson-gpio/samples/

SIMPLE_INPUT.PY

这个是一个简单的输入程序，使用的是 BCM 的引脚编码模式，可以读取 PIN12 的值并打印到屏幕。

运行程序：

sudo python3 simple_input.py

预期效果：

运行程序后，可以看到终端打印信息，默认情况下 Pin18 的值是低电平，找一个杜邦线将第 12 号引脚连到 3.3V,可以看到读取的值变成了 HIGH，如果连到 GND，会显示 LOW

```
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio/samples$ sudo python3 simple_input.py
Starting demo now! Press CTRL+C to exit
Value read from pin 18 : LOW
Value read from pin 18 : HIGH
```

【注意】

- 这里的 18 是指的 BCM 编码，上面的 PIN12 是指物理编码，也就是板子上印的丝印的编码。
- Jetson nano 的引脚的工作电平是 3.3V，所以使用的时候尽量不要接 5V 电平

SIMPLE_OUT.PY

程序会输出高电平和低电平（每 2 秒交替更新）到物理引脚 PIN12

运行程序

sudo python3 simple_out.py

预期效果

连接一个 LED 到 12 号引脚，运行程序后可以看到 LED 灯闪烁。

```
waveshare@waveshare-desktop: /opt/nvidia/jetson-gpio/samples
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio/samples$ sudo python3 simpl
Starting demo now! Press CTRL+C to exit
Outputting 1 to pin 18
Outputting 0 to pin 18
Outputting 1 to pin 18
Outputting 0 to pin 18
Outputting 1 to pin 18
```

BUTTON_LED.PY

程序使用轮询方式通过按键来控制 LED 灯。

硬件连接

需要将一个按键连接到 PIN18 和GND，同时使用一个上拉电阻连接 PIN18 到 3.3V。连接一个LED 灯到 PIN12

运行程序

sudo python3 button_led.py

预期效果

连好硬件后，运行程序，由于上拉电阻的原因，PIN18 默认为高电平，LED 熄灭，当按下按键时，PIN18 转为低电平，判断按键按下，LED 点亮。

```
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio/samples$ sudo python3 button_led.py
Starting demo now! Press CTRL+C to exit
Outputting 1 to Pin 12
Outputting 0 to Pin 12
Outputting 1 to Pin 12
Outputting 0 to Pin 12
Outputting 1 to Pin 12
Outputting 0 to Pin 12
Outputting 1 to Pin 12
Outputting 0 to Pin 12
Outputting 1 to Pin 12
Outputting 0 to Pin 12
Outputting 1 to Pin 12
Outputting 0 to Pin 12
Outputting 1 to Pin 12
```

BUTTON_EVENT.PY

程序使用事件等待方式通过按键状态来控制 LED

灯硬件连接

需要将一个按键连接到 PIN18 和GND，同时使用一个上拉电阻连接 PIN18 到 3.3V。连接一个LED 灯到 PIN12

运行程序

sudo python3 button_event.py

预期效果

连好硬件后，运行程序，由于上拉电阻的原因，PIN18 默认为高电平，LED 熄灭，当按下按键时，PIN18 转为低电平，判断按键按下，点亮 LED 大约 1s,然后熄灭。

本程序的实验效果跟 button_led.py 类似，都是通过按键点亮 LED 灯，不同的是 button_led.py 采用轮询的方式去监听按键状态，本程序采用等待按键事件的方式，可以节省 CPU。

```
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio/samples$ sudo python3 button_event.py
Starting demo now! Press CTRL+C to exit
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
Button Pressed!
Waiting for button event
```

BUTTON_INTERRUPT.PY

本程序采用中断的方式，通过按键来控制 LED

灯硬件连接

将一个按键连接到 PIN18 和GND，同时连接上拉电阻（3.3V）到 PIN18

将一个 LED 灯（接入限流电阻）到 PIN12 （LED

1） 将一个 LED 灯（接入限流电阻）到 PIN13

（LED 2） 运行程序

sudo python3 button_interrupt.py

预期效果

连好硬件后，运行程序，两个 LED 先熄灭，然后 LED 1 亮起，LED2 熄灭，按下按键之后，LED

2

会快速闪烁 5 次然后熄灭。

```
waveshare@waveshare-desktop:/opt/nvidia/jetson-gpio/samples$ sudo python3 button_interrupt.py
Starting demo now! Press CTRL+C to exit
Blink LED 2
Blink LED 2
Blink LED 2
Blink LED 2
Blink LED 2
```

摄像头

这里我们介绍一下在 Jetson Nano 上如果使用指令快速测试摄像头是否正常

★ 测试 CSI 摄像头：
DISPLAY=:0.0 gst-launch-1.0 nvarguscamerasrc ! ‘video/x-raw(memory:NVMM), width=1920, height=1080, format=(string)NV12, framerate=(fraction)30/1’ ! nvoverlaysink -e

★ USB UVC 摄像头
DISPLAY=:0.0 gst-launch-1.0 v4l2src device=/dev/video0 ! 'video/x-raw, format=YUY2, width=640, height=480, framerate=30/1' ! xvimagesink -ev

【注意】NV12 是数字 12 而不是字母 l

JETSON-INFERENCE

HELLO-AI-World 是 NVIDIA 提供的供 Jetson 使用者入门使用 Jetson 开发套件的项目，这里我们来

试着运行一下试试

1. 下载 camke 和 git 工具
   sudo apt-get update
   sudo apt-get install git cmake
2. 下载 Jetson-Inference
   git clone https://github.com/dusty-nv/jetson-
   inference cd jetson-inference
   git submodule update –init
3. 配置 cmake（时间有点长，可以休息一下，稍后继续）
   sudo mkdir build
   cd build
   cmake ..
   /
4. 编译项目
   cd jetson-inference/build
   make
   sudo make install

编译完成之后，就可以试着使用一下示例程序了。示例程序的可执行文件统一放置在 jetson- inference/build/aarch64/bin 目录下。

基本所有的例程中都调用了 imageNet（图像识别）以及 detectNet (物体跟踪)。这两个库都是继承通用的 TensorRT。有兴趣的可以单独查资料了解。