目錄

引言 1.1
一. 基本介绍 1.2

二. 使用指南

1. 刷基础固件和android系统 2.1
2. 刷入Debian和安装ROS 2.2
3. usb转串口驱动安装 2.3
4. wifi 驱动修复 2.4
5. debian镜像(有opencl和tensorflow支持) 2.5

- 引言

引言

本文档总结了hikey 970的一些相关信息和使用说明，方便开发者进行相关的开发。

在线阅读

- 海思 hikey970 开发板简介
  - 基本信息
  - 硬件配置简介
  - 96boards 简介
  - 文档信息

海思 hikey970 开发板简介

基本信息

2018年3月19日的Linaro Connect大会上华为正式发布了HiHope Hikey 970开源硬件。HiHope Hikey 970基于华为海思麒麟970，采用了台积电10nm工艺，拥有4核Cortex-A73@2.36GHz和4核Cortex-A53@1.8GHz，12核的Mali-G72。麒麟970是全球首款内置独立NPU（神经网络单元）的移动AI计算平台。HiHope Hikey 970拥有6GB的LPDDR4 和 64GB UFS。外围部分拥有支持4K高清的HDMI 2.0a接口，2个USB3.0接口，2个USB Type-C接口，1个千兆网口，支持CAN V2.0B协议，可以运行AOSP、Ubuntu、Debian等系统。

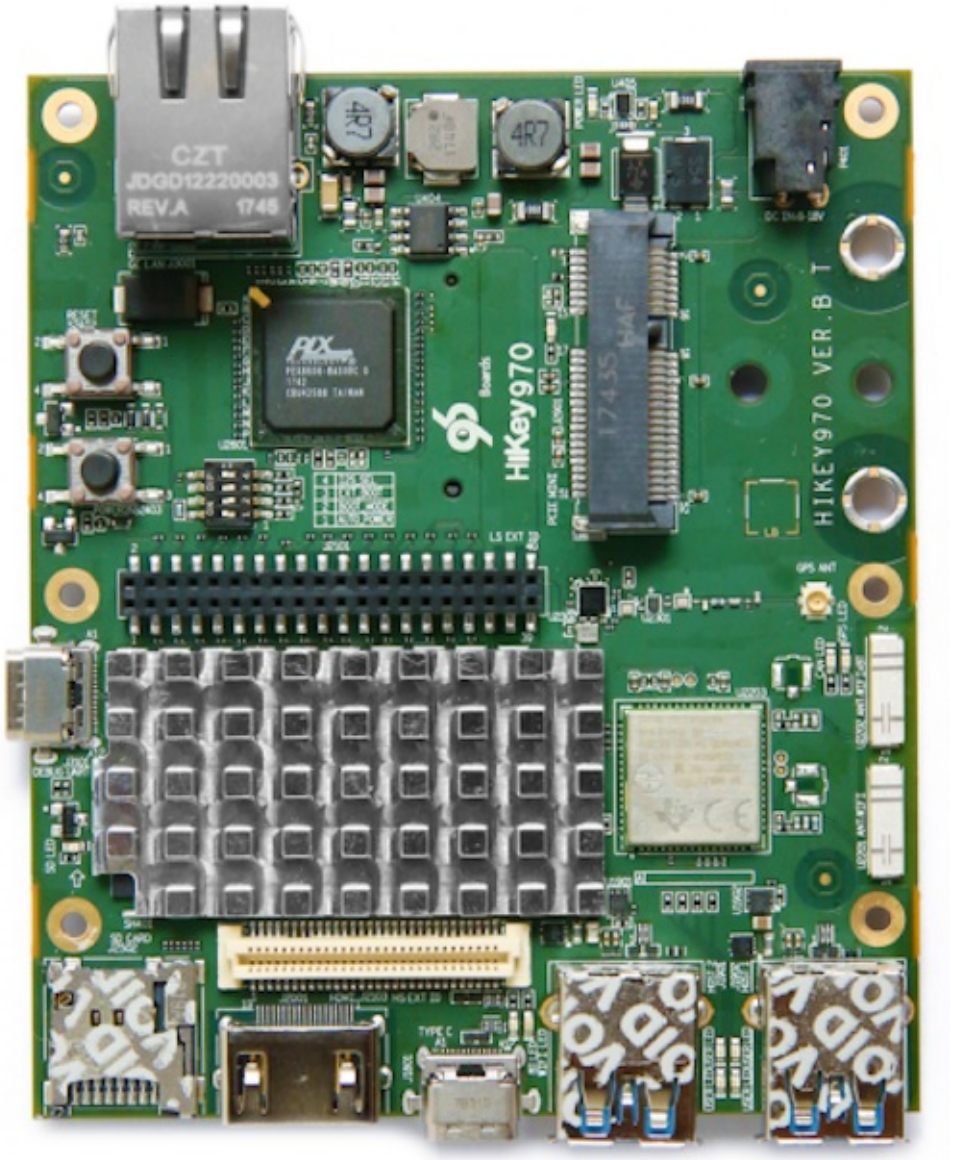
目前淘宝已经可以买到。

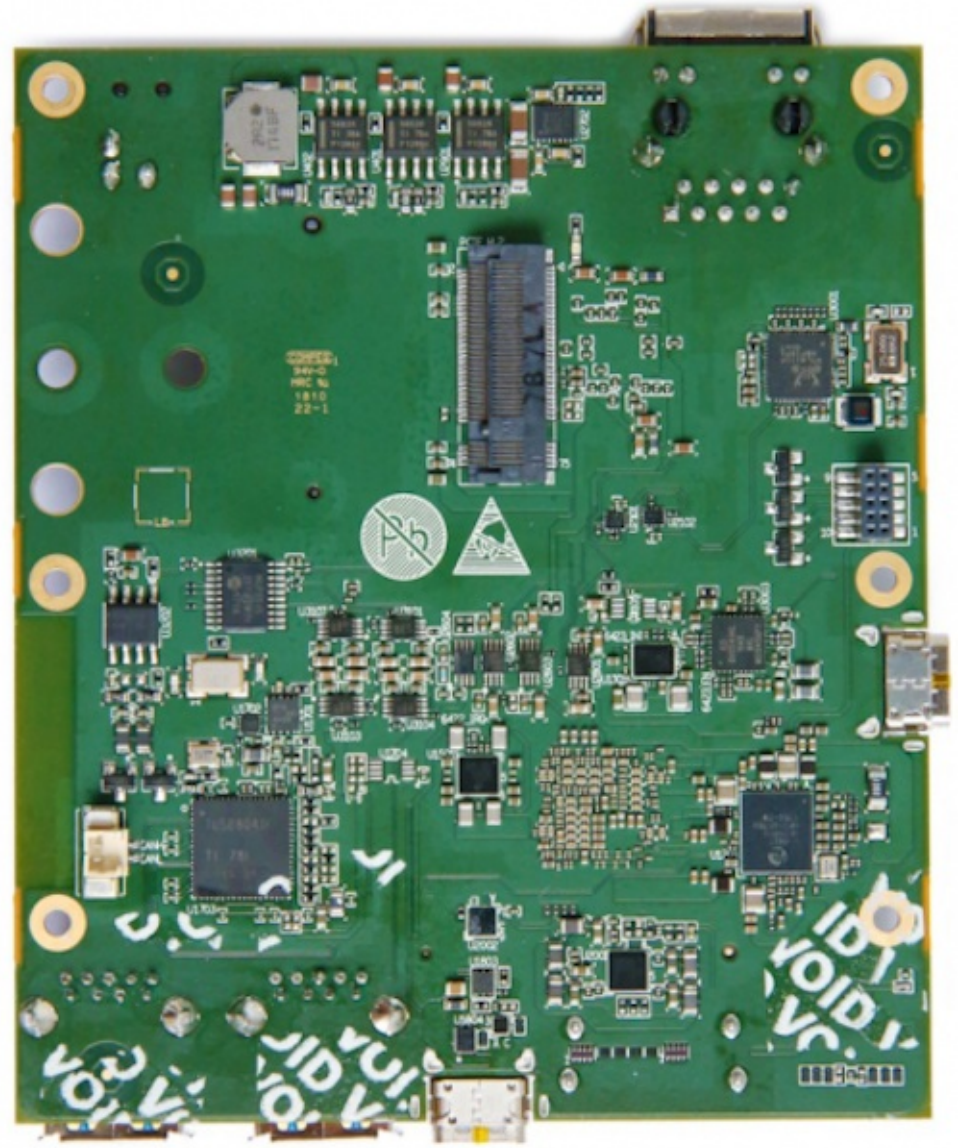
HiHope Hikey 970以强悍的性能和兼容于96Boards CE的标准，在计算机视觉、高性能计算开发、Android标准化程序开发测试等领域有广泛的应用，HiHope Hikey 970不只是开源硬件中的性能怪兽，因为集成了NPU（NPU相比与CPU在能效上有50倍以上的提升，性能上有25倍以上的提升），并且得到了华为HiAI生态的全力支持，在AI等人工智能领域开发中拥有巨大的发展前景，HiHope Hikey 970可运用于深度学习算法、智能机器人、无人驾驶和智慧城市等多个领域。

硬件配置简介

项目	参数
主板:	海思 麒麟 970 - HiAI 架构, NPU
CPU:	ARM Cortex-A73 MPCore4 @up to2.36GHz, ARM Cortex-A53 MPCore4 @up to1.8GHz
GPU:	ARM Mali-G72 MP12 GPU
内存:	6GB LPDDR4X 1866MHz
存储:	64GB UFS 2.1, Micro SD
网络:	Bluetooth/WIFI/GPS
视频:	1080p@60Hz HDMI, 4 line MIPI/LCD port
摄像头:	4 line MIPI port, 2 line MIPI port
扩展接口:	一个40pin的低速扩展口 UART, SPI, I2S, I2C x2, GPIO x12, DC power 一个60pin 的高速扩展口(HS) 4L-MIPI DSI, USB, I2C x2, 2L+4L-MIPI CSI

用户接口:	电源、复位按钮 8个LED指示灯 4 个用户可以自己控制 3 用来显示连接状态 (蓝牙 wifi和 GPS) 1个用来显示can总线
电源:	直流: +8V to +18V
系统支持:	Android Linux
尺寸:	105.26mm X 100mm 满足 96Boards 客户版本标准尺寸要求.
环境:	运行温度: 0°C to +70°C





96boards 简介

HiKey系列开发板是开源组织Linaro 96Boards的核心成员。Linaro是一个非盈利的开源代码组织，由ARM、飞思卡尔、IBM、Samsung、ST-Ericsson 及德州仪器(TI)等半导体厂商联合，在2010年3月成立。

“市场上这么多公司其实都需要开源软件作为接下来后续推广的平台，客户通过Linaro这个互相协作的平台可能只需要花一成的力气或投入就能实现过去十成投入的效果。”ARM合作伙伴支持部门总经理Monica Biddulph表示，这正是Linaro平台的意义所在。

Linaro成立后，华为、阿里巴巴、中兴通讯、LeMaker、AMD、AppliedMicro、Calxeda、Canonical、Cavium、Facebook、HP、Marvell、红帽等行业内重要公司纷纷加入，可见这个组织在行业内备受认可。而华为海思是Linaro的核心成员，首批“96 boards”开发板便由华为海思提供，这意味着海思在Linaro的开源软件开发方面拥有更大的话语权，可以在芯片开发的过程中及早取得主流开源软件的支持。

而96Boards是ARM开放平台规范，是第一个定义Cortex-A开发板的开放规范，该规范由Linaro社区委员会组织维护，取名96Boards意在说明囊括了“32位+64位”的板卡。基于96Boards规范，可以通过使用标准化平台，缩短开发周期，降低开发成本，加快产品的创新和量产速度。使用96Boards的用户众多，包括各种SoC设计、做外围芯片开发、软件应用设计等方面的客户。

96Boards目前制定了消费版、企业版和物联网版本，目前市面上消费版更为常见。华为此前就推出了支持该规范、基于麒麟620处理器的开发板“HiKey 620”，值得一提的是，HiKey 620还是全球首款符合Linaro 96Boards CE标准的开源硬件产品。

目前(2018.5.19)这块板应该是96boards上的机皇。

文档信息

这款虽然是华为的产品但是主要的文档都在96boards的官网上。

文档主页

Github 文档主页

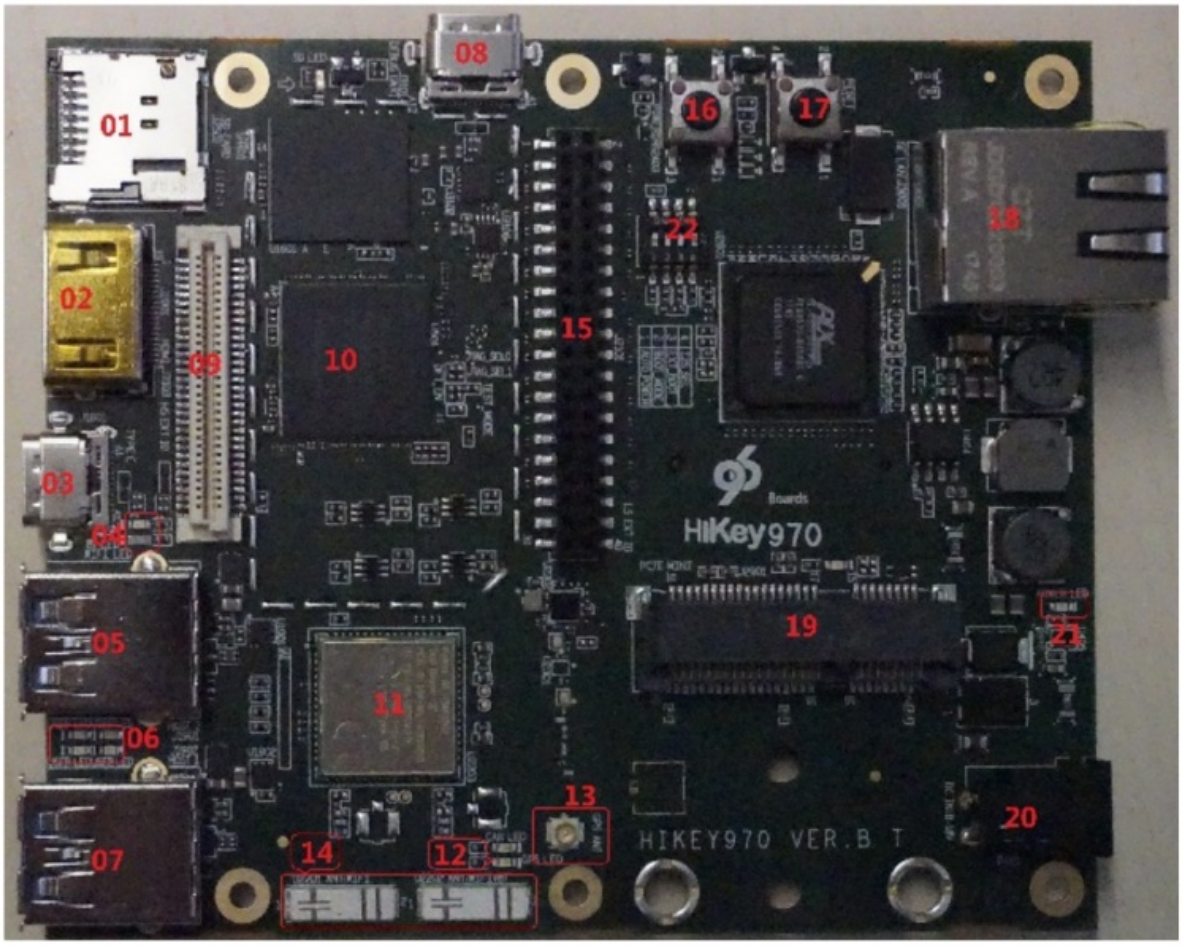
hikey970 相关软件 这里放的是hikey系列板子的相关软件。有些是某些板子独有的，从名称上能看出来比如 tools-images-hikey970。还有一些是通用的，但是针对不同的板子有不同的分支。比如 l-loader。

注意目前的文档很不完善，里面有些东西是有问题的。有些软件也不能完全按照文档说明使用。虽然官方说明支持AOSP,Ubuntu,Debian。但是目前(2018.5.19)还是只支持AOSP。

- hikey 970 开发板刷基础固件和Android系统
  - 安装基础程序
  - 基础固件
    - 下载镜像
    - 下载镜像工具
    - 设置开发板启动方式
    - 刷入固件
  - 系统镜像

hikey 970 开发板刷基础固件和Android系统

刷板子的系统主要包括两个方面，一个是基础固件,一个就是系统文件。根据板子上面四个开关状态的不同板子会处在不同的状态。



开关位于图中的22的位置 这四个开关的功能分别是

1. 自动开机 这个开关如果是On,当板子通电的时候就会自动启动系统。反之则上电之后要按电源键启动。
2. 启动模式 开关是On的时候上电后启动至刷基础固件模式，这时候可以通过程序给板子刷基础固件。当是off的时候系统会从已经安装的bootloader启动。

3. 系统启动模式 当是On的时候系统会以UEFI模式启动，反之则会直接启动系统
4. 扩展功能选择 当是On的时候会使用低速扩展口功能，当是Off的时候会使用蓝牙功能。

基础固件包含了板子的bootloader等等。如果基础固件刷错了系统就不能再启动了。甚至连fastboot也不能使用，也无法再刷其他系统。正常情况下只是刷系统是不需要刷基础固件的。

安装基础程序

```
sudo apt-get install android-tools-adb
sudo apt-get install android-tools-fastboot
```

基础固件

以下的操作都是在Linux系统环境下完成的

下载镜像

从96boards的970下载主页下载最新的系统镜像。目前(2018.5.19)只有Android的。这个文件中包含了基础固件和系统镜像。

下载完成后你需要解压这个文件。在image文件夹中包含了需要的文件。

下载镜像工具

```
git clone https://github.com/96boards-hikey/tools-images-hikey970
```

官方的工具有错误，我们需要进行修改后才能用 在 recovery-flash.sh 的第一行

```
#/bin/bash
```

改为

```
#!/bin/bash
```

将第九行

```
UEFI_BUILD_PATH=/home/qubo/tools-images-hikey970
```

改为

```
UEFI_BUILD_PATH=${PWD}
```

然后把刚才下载的文件中的 `image` 里的所有文件复制到 `tools-images-hikey970` 文件夹

设置开发板启动方式

把板子的开关设置成 On On On Off,然后上电。按照之前的说明，这时候板子处在刷基础固件的状态。用数据线通过USB把板子和电脑连接起来。

正常情况下你会在/dev/ 文件夹下面发现一个ttyUSBX的设备，X是一个数字可能是0,1之类的。确认自己的板子是对应哪个设备。

刷入固件

运行

```
sudo ./recovery-flash.sh ttyUSBX
# 如果你的设备是ttyUSB0那么你可以省略第二个参数
```

程序卡在了 `wait for devices` 可能的原因是

1. 你安装过modemmanager，这个程序会给我们的板子发数据导致程序写入失败。解决方法很简单，卸载这个软件

```
sudo apt-get purge modemmanager
```

1. 你刷入了错误的固件。可能你的固件是从别的地方下载的，也可能是自己编译的。如果是这样那么很有可能是固件自身出了问题。

系统镜像

实际上在刷入上个基础固件的时候会自动的把系统镜像也刷入进去。这时候把板子上的开关设置为On Off On Off就可以启动系统了。第一次启动的时候要多等一会，因为Android要先初始化一下。

如果只是要刷系统固件可以只刷入系统固件中的下面几个文件。执行

```
sudo fastboot flash boot boot.img
sudo fastboot flash cache cache.img
sudo fastboot flash system system.img
sudo fastboot flash userdata  userdata.img
```

- 在hikey 970上安装debian并运行ROS
    - 设置机器开关
    - 刷入Debian 系统
    - 启动系统
    - 安装常用工具
    - 安装ROS

在hikey 970上安装debian并运行ROS

经过长时间的等待hikey 970的debian系统终于发布了。

系统下载地址

首先保证机器能够正常运行Android,基础固件没有问题。下载完成后解压文件

设置机器开关

刷系统前需要让板子开机后进入fastboot模式，这个可以通过设置板子上的开关完成。把开关状态设置成On Off On Off。然后上电启动。

刷入Debian 系统

在Linux下刷入Debian系统

如果你是Linux系统则按照此处的指令刷入Debian系统

```
#进入解压后的文件夹
fastboot flash xloader sec_xloader.img
fastboot flash ptable prm_ptable.img
fastboot flash fastboot l-loader.bin
fastboot flash fip fip.bin
fastboot flash boot boot2grub.uefi.img
fastboot flash system rootfs.sparse.img
# 注意刷入系统文件的过程需要很多时间，需要耐心等待
# 刷入分区表
wget http://www.bwbot.org/s/uhzKfx -O '64gtoendprm_ptable.img'
fastboot flash ptable 64gtoendprm_ptable.img
```

在Windows 下刷人Debian系统

如果你是在Windows环境下刷入Debian系统则执行下面的指令

```
#进入解压后的文件夹
.\update_Hikey970.bat
```

整个刷入时间要很久，需要耐心等待 刷入完成后再次刷入分区表 分区表下载地址 下载完成后放入当前文件夹执行下面的指令

```
fastboot.exe flash ptable 64gtoendprm_ptable.img
```

启动系统

拔掉板子的电源，将开关拨至 On Off Off Off 状态，然后上电。等待板子启动完成。给板子插上网线，这时候可以看到网口的灯在闪烁。说明板子已经正常启动了。

在路由器上查找板子的ip

正常情况下应该能够看到如图所示的设备。然后通过ssh 连接就可以了。用户名和密码都是hoperun

```
ssh hoperun@xxx.xxx.xxx.xxx
```

```
hoperun@192.168.0.120's password:
Linux hoperun-developer 4.9.78-147538-g244928755bbe #1 SMP PREEMPT Thu May 31 10:58:02 CST 2018 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jun  8 07:01:20 2018 from 192.168.0.8
hoperun@hoperun-developer:~$
```

安装常用工具

```
sudo apt-get install bash-completion #增加自动补全功能
sudo apt-get install htop #查看系统资源使用情况工具
/bin/bash -c "$(curl -sL https://git.io/vokNn)" #安装apt-fast，安装软件更快
```

安装ROS

ROS的安装过程和一般的ROS版本安装是一样的，但是奇怪的是我的源里面没有找到kinetic版本。下面安装的是melodic版本。可能是这个版本的Debian只支持melodic。

添加ROS软件源

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

添加key

```
sudo apt-fast update
sudo apt-fast install dirmngr
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F
717815A3895523BAEEB01FA116
```

开始安装

```
sudo apt-fast update
sudo apt-fast install ros-melodic-desktop-full -y
```

配置环境

```
sudo rosdep init
rosdep update
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

安装编译软件包依赖

```
sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-ess
ential
```

测试一下

```
hoperun@hoperun-developer:~$ roscore
... logging to /home/hoperun/.ros/log/608389b6-6af0-11e8-ba7e-501d93042bab/roslaunch-hoperun-developer-15715.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://hoperun-developer:35713/
ros_comm version 1.14.1


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.1

NODES

auto-starting new master
process[master]: started with pid [15725]
ROS_MASTER_URI=http://hoperun-developer:11311/

setting /run_id to 608389b6-6af0-11e8-ba7e-501d93042bab
process[rosout-1]: started with pid [15736]
started core service [/rosout]
```

成功了！

- Hikey 970 USB转串口驱动安装
    - 下载Linux内核源代码
    - 配置内核源代码
    - 编译驱动module
    - 安装驱动module
    - 测试驱动
    - 自动加载驱动

Hikey 970 USB转串口驱动安装

在已经发布的hikey 970 Debian系统中是没有包含U转串驱动的。在没有安装驱动的情况下插上U转串设备时在/dev下面是没有ttyUSB设备的。

安装驱动需要自己编译对应的驱动程序。安装方法如下。

下载Linux内核源代码

执行下面的指令下载hikey linux内核源代码

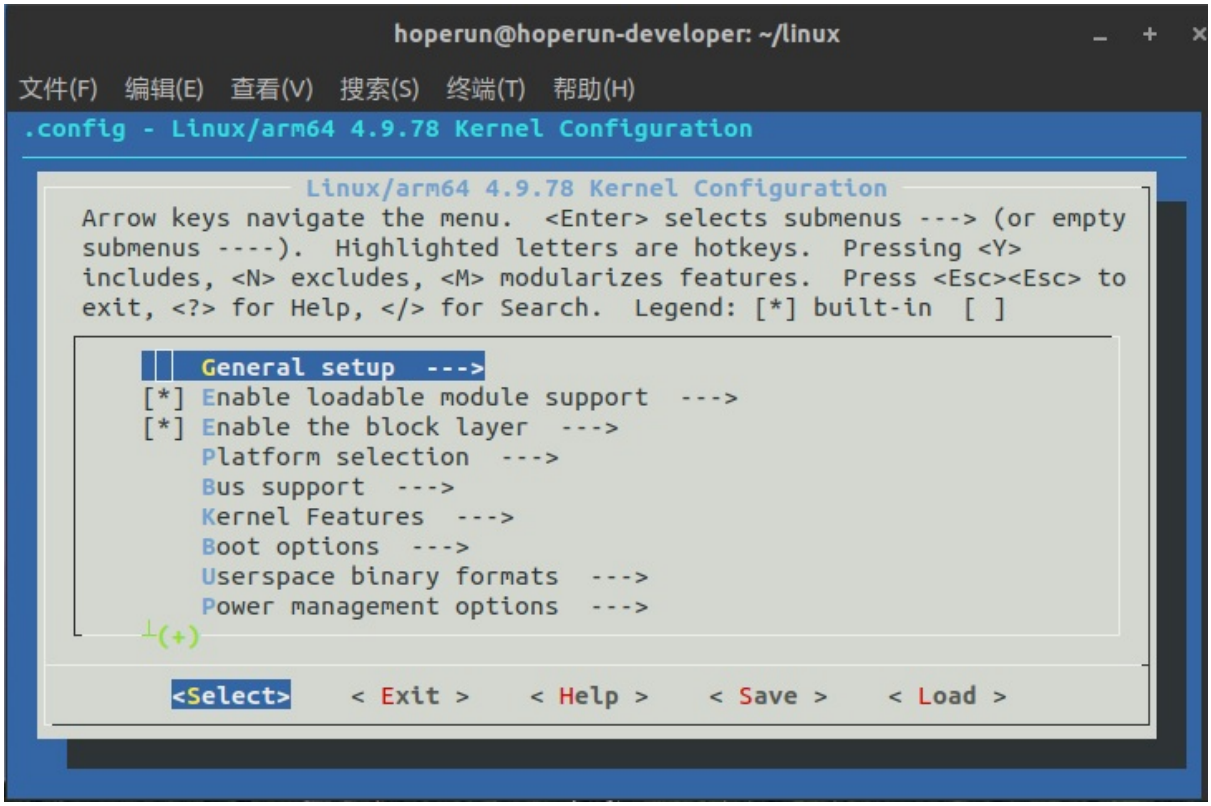
```
git clone --single-branch -b hikey970-v4.9 --depth=1 https://github.com/96boards-hikey/lin
ux
# 切换到hikey 970分支
cd linux
git checkout hikey970-v4.9
```

配置内核源代码

获取内核配置文件

```
cp /proc/config.gz ~/
gzip -d ~/config.gz
# 进入内核源代码文件夹
cd ~/linux
# 将内核配置文件复制到此处
make mrproper
cp ~/config .config
sudo chmod 666 .config
# 配置内核文件
sudo apt-get install libncurses5-dev
sudo apt-get install bc
make menuconfig
```

正常情况下会显示如下的内核配置界面



找到 Device Drivers-->USB Support --> USB Serial Converter Support 将其设置成M。编译内核module有两种模式，一种是直接编译到内核里面，另一种是编译成独立的.ko文件module。我们采用的是.ko的模式。这样不用重新编译内核更加方便。 继续进入此选项将想要编译的驱动设置成M，如果不清楚自己的型号可以全部设置成M 设置完成后选择保存，之后再退出此界面

编译驱动module

执行下面的语句开始编译内核

```
make modules_prepare
sudo make -j4 modules # 需要执行这个才会生成modules.order,modules.builtin
make M=drivers/usb/serial
```

正常情况下应该能够看到终端输出如下

```
  LD      drivers/usb/serial/built-in.o
  Building modules, stage 2.
  MODPOST 50 modules
  CC      drivers/usb/serial/aircable.mod.o
  LD [M]  drivers/usb/serial/aircable.ko
  CC      drivers/usb/serial/ark3116.mod.o
  LD [M]  drivers/usb/serial/ark3116.ko
  CC      drivers/usb/serial/belkin_sa.mod.o
  LD [M]  drivers/usb/serial/belkin_sa.ko
  CC      drivers/usb/serial/ch341.mod.o
  LD [M]  drivers/usb/serial/ch341.ko
  CC      drivers/usb/serial/cp210x.mod.o
  LD [M]  drivers/usb/serial/cp210x.ko
```

```
  CC      drivers/usb/serial/cyberjack.mod.o
  LD [M]  drivers/usb/serial/cyberjack.ko
  CC      drivers/usb/serial/cypress_m8.mod.o
  LD [M]  drivers/usb/serial/cypress_m8.ko
  CC      drivers/usb/serial/digi_acceleport.mod.o
  LD [M]  drivers/usb/serial/digi_acceleport.ko
  CC      drivers/usb/serial/empeg.mod.o
  LD [M]  drivers/usb/serial/empeg.ko
  CC      drivers/usb/serial/f81232.mod.o
  LD [M]  drivers/usb/serial/f81232.ko
  CC      drivers/usb/serial/ftdi_sio.mod.o
  LD [M]  drivers/usb/serial/ftdi_sio.ko
  CC      drivers/usb/serial/garmin_gps.mod.o
  LD [M]  drivers/usb/serial/garmin_gps.ko
  CC      drivers/usb/serial/io_edgeport.mod.o
  LD [M]  drivers/usb/serial/io_edgeport.ko
  CC      drivers/usb/serial/io_ti.mod.o
  LD [M]  drivers/usb/serial/io_ti.ko
  CC      drivers/usb/serial/ipaq.mod.o
  LD [M]  drivers/usb/serial/ipaq.ko
  CC      drivers/usb/serial/ipw.mod.o
  LD [M]  drivers/usb/serial/ipw.ko
  CC      drivers/usb/serial/ir-usb.mod.o
  LD [M]  drivers/usb/serial/ir-usb.ko
  CC      drivers/usb/serial/iuu_phoenix.mod.o
  LD [M]  drivers/usb/serial/iuu_phoenix.ko
  CC      drivers/usb/serial/keyspan.mod.o
  LD [M]  drivers/usb/serial/keyspan.ko
  CC      drivers/usb/serial/keyspan_pda.mod.o
  LD [M]  drivers/usb/serial/keyspan_pda.ko
  CC      drivers/usb/serial/kl5kusb105.mod.o
  LD [M]  drivers/usb/serial/kl5kusb105.ko
  CC      drivers/usb/serial/kobil_sct.mod.o
  LD [M]  drivers/usb/serial/kobil_sct.ko
  CC      drivers/usb/serial/mct_u232.mod.o
  LD [M]  drivers/usb/serial/mct_u232.ko
```

编译完成之后可以在drivers/usb/serial中看到生成了许多.ko文件。这些就是我们需要的驱动文件。

安装驱动module

```
# 创建module文件目录
sudo mkdir -p /lib/modules/$(uname -r)/kernel/drivers/usb/serial/
sudo cp drivers/usb/serial/*.ko /lib/modules/$(uname -r)/kernel/drivers/usb/serial/
# 复制depmod依赖文件
sudo cp ~/linux/modules.order /lib/modules/$(uname -r)/
sudo cp ~/linux/modules.builtin /lib/modules/$(uname -r)/
# 生成对应文件
cd /lib/modules/$(uname -r)
sudo depmod -a
# 加载驱动
sudo modprobe pl2303
```

测试驱动

查看驱动是否正常加载

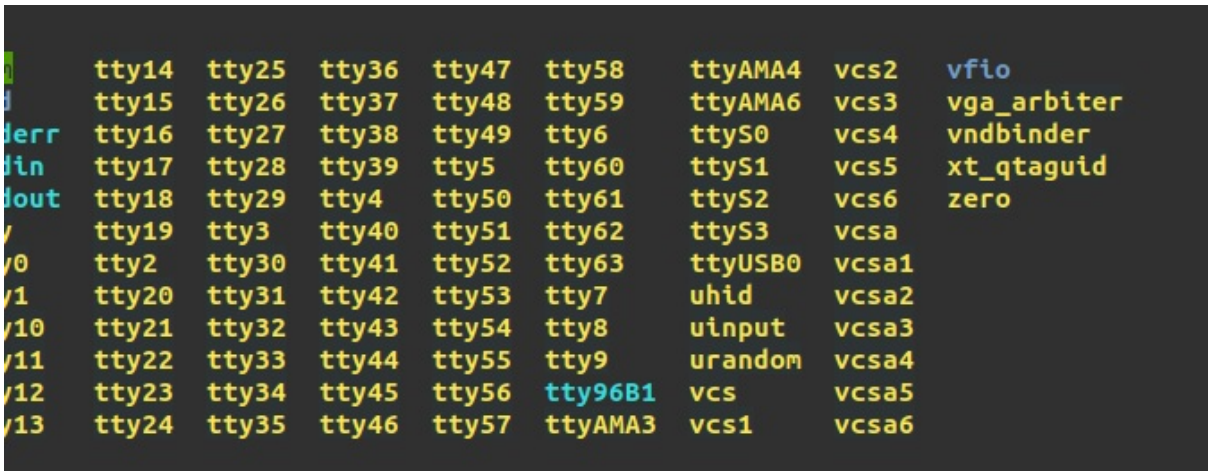
```
lsmod
```

正常输出如下

```
Module                  Size  Used by
ftdi_sio               49152  0
pl2303                 20480  0
usbserial              40960  2 ftdi_sio,pl2303
```

可以看到pl2303驱动已经成功加载。

这时再插上U转串试一下



可以看到已经有ttyUSB0了。至此串口已可以正常使用了。

自动加载驱动

修改 /etc/modules文件 在其中加入想要加载的内核模块的名称，比如对于我的设备就是pl2303。文件内容如下

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
pl2303
```

保存退出，下次在系统启动时就会自动加载这个驱动了。

- hikey 970 debian wifi 驱动修复

hikey 970 debian wifi 驱动修复

执行下面指令安装无线网卡驱动

```
sudo apt-get install firmware-ti-connectivity
```

连接wifi

```
sudo nmcli device wifi connect TP-LINK_5G_A134 password blueWhale
#其中TP-LINK_5G_A134为wifi的ssid，blueWhale为wifi的密码，请根据自己的情况设置。
```

- hikey 970 debian 镜像（有opencl 和 tensorflow）
  - 下载镜像
  - 刷入镜像
  - 启动系统
  - 调整分区
  - 测试tensorflow

hikey 970 debian 镜像（有opencl 和 tensorflow）

最近(2018.8.30) LeMaker发布了一个新的 Hikey 970 debian镜像。此镜像增加了HDMI支持，稳定性也提高了。同时增加了OpenCL和Tensorflow（只有python3.5）支持。此 Tensorflow 应该是进行了优化的，具体的是不是用到了NPU还不确定。经过我自己的测试此 Tensorflow 的速度比我自己安装的速度要快一倍。但是此镜像仍然不支持自带的蓝牙。

下面是安装和使用方法

下载镜像

镜像下载链接

从上面链接下载镜像并解压

```
tar -xzvf hikey970-lebian-9.tar.gz
```

刷入镜像

把hikey 970板子开关拨至 on off on off。链接USB线，并给板子上电。然后自己的电脑上执行下面的指令开始刷机

```
./flash-all-binaries.sh
```

等待刷机完成

刷入分区补丁

默认的系统分区很小，所以需要打一个补丁，之后可以调整系统分区到60G。

分区补丁下载

从上面的的下载地址下载补丁，之后解压。 把解压后的内容复制到刚才的 hikey970-lebian-9 文件夹内。

然后执行

```
sudo fastboot flash boot boot-hikey970.uefi.2.img
```

```
sudo fastboot flash userdata hikey970-lebian9-tf.img
```

等待程序执行完成

启动系统

断开板子的电源，然后把开关拨至on off off off。给板子连接上鼠标键盘网线和显示器。然后给板子上电。 等待系统启动完成。正常情况下应该可以看到登陆界面。用户名和密码都是 `shunya` 。

调整分区

运行下面的指令调整分区

```
sudo resize2fs /dev/sdd15
```

这样能够把系统分区扩展到20G，剩下的需要用gparted扩展

```
sudo apt-get install gparted
sudo gparted-pkexec # 注意此指令只能再外接显示器的情况下才能运行
```

选择60G的硬盘，然后把未分配的空间全部扩展到最后一个分区。

```
shunya@hikey970:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        53G  7.5G   44G  15% /
devtmpfs        2.8G     0  2.8G   0% /dev
tmpfs           2.9G     0  2.9G   0% /dev/shm
tmpfs           2.9G   25M  2.8G   1% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           2.9G     0  2.9G   0% /sys/fs/cgroup
tmpfs           577M     0  577M   0% /run/user/107
tmpfs           577M     0  577M   0% /run/user/1000
```

可以看到系统空间已经增大到53G了。

测试tensorflow

系统默认安装了Python 3.5的 tensorflow。下面用 pose-tensorflow 来测试一下。这个软件是一个利用 tensorflow 识别人体关键位置的程序。

下载程序

```
git clone https://github.com/eldar/pose-tensorflow
```

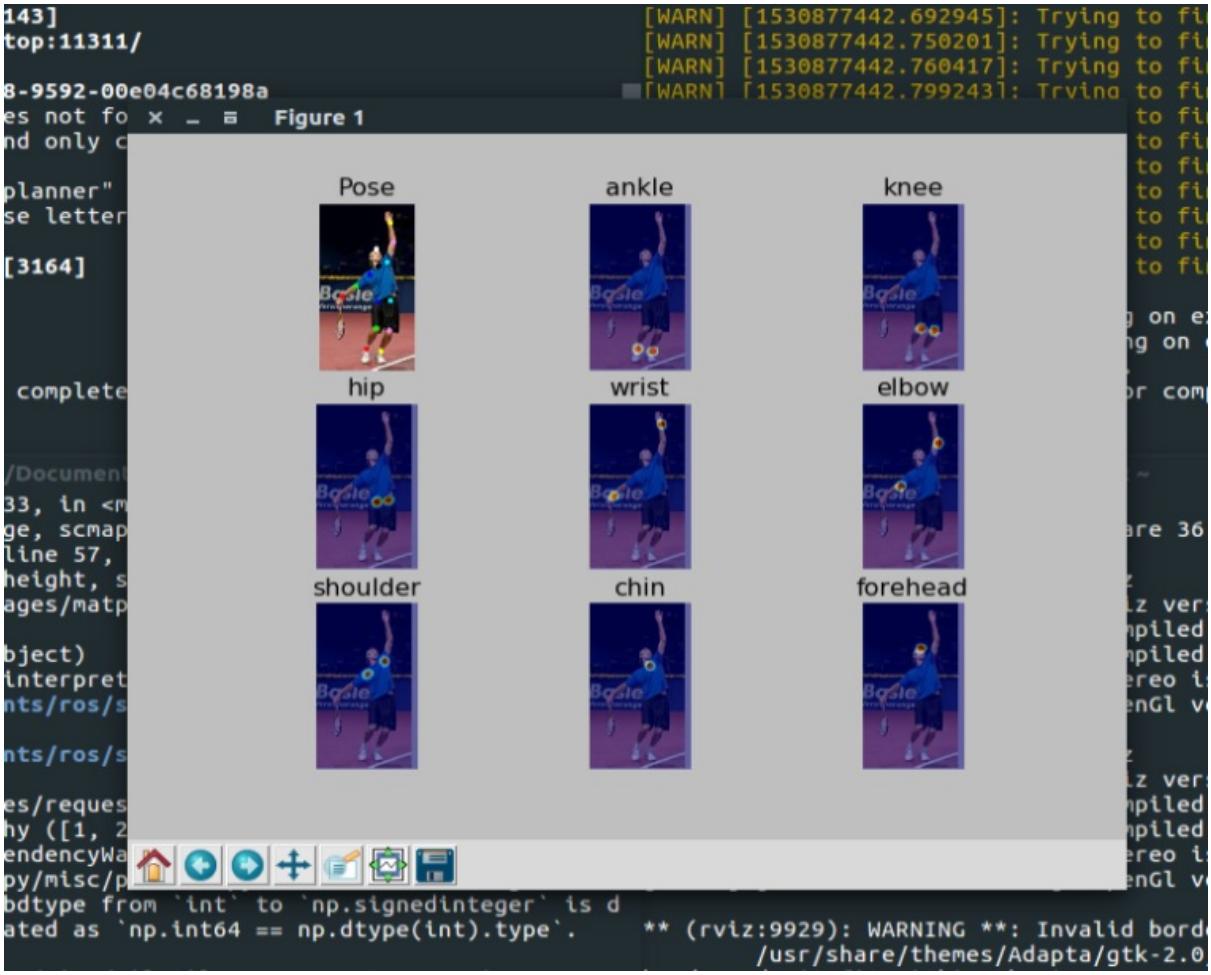
安装依赖

```
pip3 install scipy scikit-image matplotlib pyyaml easydict cython munkres
```

下载模型

```
# Download pre-trained model files
$ cd models/mpii
$ ./download_models.sh
$ cd -
```

运行模型

```
# Run demo of single person pose estimation
$ TF_CUDNN_USE_AUTOTUNE=0 python3 demo/singleperson.py
```

正常应该会显示下图