

NanoPi M1/zh

From FriendlyELEC WiKi

English

Contents

- 1 介绍
- 2 资源特性
- 3 接口布局和尺寸
 - 3.1 接口布局
 - 3.2 机械尺寸
- 4 快速入门
 - 4.1 准备工作
 - 4.2 经测试可选用的TF卡
 - 4.2.1 下载固件
 - 4.2.1.1 官方固件
 - 4.2.1.2 工具软件(可选)
 - 4.2.2 Linux-3.4和Linux-4.14系统固件差异
 - 4.2.3 烧写Linux系统
 - 4.2.3.1 烧写到TF卡
 - 4.2.4 烧写Android系统
 - 4.2.4.1 烧写到TF卡
- 5 Debian系统的使用
 - 5.1 连接有线网络
 - 5.2 连接无线网络
 - 5.3 安装Debian软件包
 - 5.4 选择系统默认音频设备
 - 5.5 通过VNC和SSH登录系统
 - 5.6 连接USB摄像头模块(FA-CAM202)
 - 5.7 连接摄像头测试OpenCV
 - 5.8 连接DVP摄像头模块(CAM500B)
 - 5.9 命令行查看CPU工作温度
 - 5.10 测试GPU
 - 5.11 测试VPU
- 6 FriendlyCore的使用
 - 6.1 介绍
 - 6.2 运行FriendlyCore
 - 6.3 开发Qt应用
 - 6.4 开机自动运行Qt示例程序
 - 6.5 扩展TF卡文件系统
 - 6.6 连接WiFi
 - 6.7 连接以太网
 - 6.8 访问GPIO/I2C/串口等硬件资源
 - 6.9 定制命令行的欢迎信息 (文字LOGO)
 - 6.10 修改时区
 - 6.11 选择系统默认音频设备
 - 6.12 连接DVP摄像头模块(CAM500B)
 - 6.13 连接USB摄像头模块(FA-CAM202)
 - 6.14 查看CPU温度和频率
 - 6.15 运行Qt示例程序
 - 6.16 Docker在armhf系统下的安装与使用
 - 6.16.1 安装 Docker
 - 6.16.2 测试 Docker
- 7 FriendlyWrt的使用
 - 7.1 介绍
 - 7.2 登录系统
 - 7.3 软件包管理
 - 7.4 查看系统状态
 - 7.5 查看Network->Interfaces的配置

- 7.6 使用USB WiFi
 - 7.7 使用华为随行WiFi 2 mini(E8372H-155)
- 8 更多OS
 - 8.1 DietPi
- 9 如何编译Linux 系统
 - 9.1 使用Linux-4.14 BSP
 - 9.2 使用Linux-3.4 BSP
 - 9.2.1 准备工作
 - 9.2.2 安装交叉编译器
 - 9.2.3 编译lichee源码
 - 9.2.4 编译U-boot
 - 9.2.5 编译Linux内核
 - 9.2.6 清理lichee源码
- 10 Android系统的使用
 - 10.1 连接USB WiFi
 - 10.2 使用红外遥控器(RC-100)
 - 10.3 播放4K视频
- 11 如何编译Android 系统
 - 11.1 准备工作
 - 11.2 编译Android
 - 11.3 清理lichee源码
- 12 开发者指南
 - 12.1 CVBS
- 13 编译内核头文件安装包
 - 13.1 本文适用于如下版本的固件
 - 13.2 安装所需软件包
 - 13.3 制作内核头文件安装包
 - 13.4 安装
 - 13.5 测试
- 14 更多OS
 - 14.1 DietPi
 - 14.2 Ubuntu-MATE
 - 14.3 Debian8 (Jacer)
 - 14.4 Android (Jacer)
 - 14.5 Armbian
- 15 固件下载
- 16 3D打印文件下载
- 17 迷你扩展板M1初学者套件
- 18 NanoPi M1初学者入门开发教程
- 19 开发者指南
- 20 资源链接
 - 20.1 手册原理图及开发资料
 - 20.2 开发教程及文档
- 21 硬件更新
- 22 更新日志

1 介绍

- NanoPi M1（以下简称M1）是友善之臂团队面向创客、嵌入式爱好者，电子艺术家、发烧友等群体推出的又一款完全开源的掌上创客神器，它的大小只有树莓派的大约2/3，可运行Debian、Ubuntu-MATE、Ubuntu-Core、Android等操作系统。
- NanoPi M1采用了全志高性能处理器Allwinner H3，集成以太网、红外接收、视频/音频输出等接口，支持HDMI、AVOUT视频输出等功能。
- 尽管体积很小，设计却紧凑美观。NanoPi M1引出了相当丰富的接口，包括HDMI、以太网、USB-Host、USB-OTG、DVP camera和AVOUT（音频+视频）等。而且集成了板载麦克风，红外接收器，并且兼容树莓派GPIO口，并且拥有独立的调试串口等。

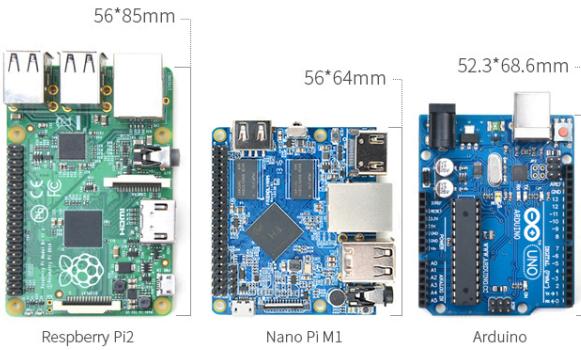
2 资源特性

- CPU: Allwinner H3, Quad-core Cortex-A7@1.2GHz
- GPU: Mali400MP2@600MHz, Supports OpenGL ES2.0
- DDR3 RAM: 512MB/1GB
- 网络: 10/100M以太网



概览

- 音频：3.5mm耳机座/Via HDMI
- 麦克风：板载麦克风
- 红外：板载红外接收模块
- USB Host：Type A型号，USB 2.0 x 3
- MicroSD Slot：x1
- MicroUSB：支持供电和数据传输，有OTG功能
- 视频输出：HDMI 1.4 1080P高清显示, CVBS
- DVP Camera接口：24pin, 0.5mm间距竖直贴片FPC座
- 调试串口：4Pin, 2.54mm排针
- GPIO：40pin, 2.54mm间距，兼容RaspberryPi2的扩展GPIO，含UART, SPI, I2C, PWM, IO等管脚资源
- 按键：GPIO按键x1，复位按键x1
- PC Size: 64 x 56mm
- Power Supply: DC 5V/2A
- 温度工作范围：零下30摄氏度到70摄氏度
- OS/Software: U-boot, Debian, Ubuntu-MATE, Ubuntu-Core

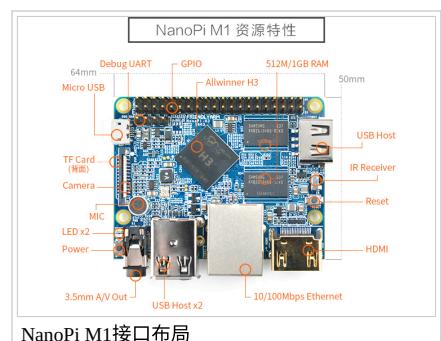


3 接口布局和尺寸

3.1 接口布局

■ GPIO管脚定义

Pin#	Name	Linux gpio	Pin#	Name	Linux gpio
1	SYS_3.3V		2	VDD_5V	
3	I2C0_SDA / GPIOA12		4	VDD_5V	
5	I2C0_SCL / GPIOA11		6	GND	
7	GPIOG11	203	8	UART1_TX / GPIOG6	198
9	GND		10	UART1_RX / GPIOG7	199
11	UART2_TX / GPIOA0	0	12	GPIOA6	6
13	UART2_RTS / GPIOA2	2	14	GND	
15	UART2_CTS / GPIOA3	3	16	UART1_RTS / GPIOG8	200
17	SYS_3.3V		18	UART1_CTS / GPIOG9	201
19	SPI0_MOSI / GPIOC0	64	20	GND	
21	SPI0_MISO / GPIOC1	65	22	UART2_RX / GPIOA1	1
23	SPI0_CLK / GPIOC2	66	24	SPI0_CS / GPIOC3	67
25	GND		26	SPDIF-OUT / GPIOA17	17
27	I2C1_SDA / GPIOA19 / PCM0_CLK / I2S0_BCK	19	28	I2C1_SCL / GPIOA18 / PCM0_SYNC / I2S0_LRCK	18
29	GPIOA20 / PCM0_DOUT / I2S0_SDO	20	30	GND	
31	GPIOA21 / PCM0_DIN / I2S0_SDIN	21	32	GPIOA7	7
33	GPIOA8	8	34	GND	
35	UART3_CTS / SPI1_MISO / GPIOA16	16	36	UART3_TX / SPI1_CS / GPIOA13	13
37	GPIOA9	9	38	UART3_RTS / SPI1_MOSI / GPIOA15	15
39	GND		40	UART3_RX / SPI1_CLK / GPIOA14	14



- Debug Port (UART0)

Pin#	Name
1	GND
2	VDD_5V
3	UART_RXD0 / GPIOA4
4	UART_RXD0 / GPIOA5 / PWM0

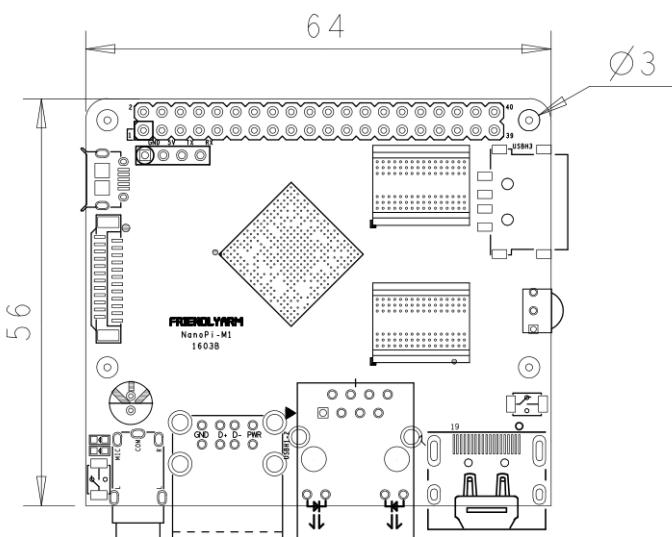
- DVP Camera IF 管脚定义

Pin#	Name	Description
1, 2	SYS_3.3V	3.3V电源输出给外部摄像头模块
7,9,13,15,24	GND	参考地, 0V
3	I2C2_SCL	I2C时钟信号
4	I2C2_SDA	I2C数据信号
5	GPIOE15	普通GPIO, 施加给外部摄像头模块的控制信号
6	GPIOE14	普通GPIO, 施加给外部摄像头模块的控制信号
8	MCLK	提供给外部摄像头模块的时钟信号
10	NC	没有连接
11	VSYNC	外部摄像头模块输出给CPU的行信号
12	HREF/HSYNC	外部摄像头模块输出给CPU的场信号
14	PCLK	外部摄像头模块输出给CPU的像数点信号
16-23	Data bit7-0	数据信号

说明

1. SYS_3.3V: 3.3V电源输出
2. VDD_5V: 5V电源输入/输出。当电压大于MicroUSB时，向板子供电，否则板子从MicroUSB取电。输入范围：4.7~5.5V
3. 全部信号引脚均为3.3V电平，输出电流为5mA,可以带动小负载模块，io都不能带负载
4. 更详细的信息请查看原理图：NanoPi-M1-V1.1_1804-Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/1/1e/Schematic_NanoPi-M1-V1.1_1804.pdf)

3.2 机械尺寸



详细尺寸：NanoPi_M1_v1.1_1804_PCB的dx文件 (http://wiki.friendlyelec.com/wiki/index.php/File:NanoPi_M1_v1.1_1804_PCB.rar)

4 快速入门

4.1 准备工作

要开启你的NanoPi M1新玩具，请先准备好以下硬件

- NanoPi M1主板

- microSD卡/TF卡: Class10或以上的 8GB SDHC卡
- 一个microUSB接口的外接电源，要求输出为5V/2A（可使用同规格的手机充电器）
- 一台支持HDMI输入的显示器或者电视
- 一套USB键盘鼠标，同时连接还需要USB HUB（或选购串口转接板，要PC上进行操作）
- 一台电脑，需要联网，建议使用Ubuntu 18.04 64位系统

4.2 经测试可选用的TF卡

制作启动TF卡时，建议Class10或以上的 8GB SDHC卡。以下是经友善电子测试验证过的高速TF卡：

- SanDisk闪迪 32GB TF (MicroSD) 存储卡 U3 C10 A1 V30 4K 至尊超极速移动版内存卡 (**开发者推荐**)



- SanDisk闪迪32GB TF (MicroSD) 存储卡 行车记录仪&安防监控专用内存卡 (**长时间运行推荐**)



- SanDisk闪迪 TF 8G Class10 microSD 高速 TF卡：

SanDisk 闪迪



- SanDisk闪迪 TF 128G 至尊高速 Class10 microSDXC TF 128G 48MB/S:



- 川宇 8G手机内存卡 TF 8G 卡存储卡 C10 高速 Class10 microSD卡：



4.2.1 下载固件

4.2.1.1 官方固件

访问此处的下载地址 (<http://download.friendlyelec.com/NanoPiM1>) 下载固件文件 (位于网盘的"01_系统固件"目录)：
下表列出了所有官方固件，文件名中的XYZ代表文件的不同用途，其含义如下：

- **sd**: 安装系统到TF卡时使用
- **eflasher**: 需要通过TF卡烧写系统到eMMC时使用

图标	文件名	版本	描述	内核版本
	h3-XYZ-debian-bookworm-core-4.14-armhf-YYYYMMDD.img.gz	bookworm	Debian12 精简版固件，没有桌面, 仅命令行	4.14.y
	h3-XYZ-debian-jessie-3.4-armhf-YYYYMMDD.img.gz	jessie	Debian8 固件, 有图形界面	3.4.y
	h3-XYZ-debian-jessie-4.14-armhf-YYYYMMDD.img.gz	jessie	Debian8 固件, 有图形界面	4.14.y
	h3-XYZ-friendlycore-focal-4.14-armhf-YYYYMMDD.img.gz	focal	FriendlyCore系统固件, 基于Ubuntu core 20.04构建, 内置Qt4.8	4.14.y
	h3-XYZ-friendlycore-jammy-4.14-armhf-YYYYMMDD.img.gz	jammy	FriendlyCore系统固件, 基于Ubuntu core 22.04构建, 内置Qt4.8	4.14.y
	h3-XYZ-friendlycore-xenial-4.14-armhf-YYYYMMDD.img.gz	xenial	FriendlyCore系统固件, 基于Ubuntu core 16.04构建, 内置Qt4.8	4.14.y
	h3-XYZ-friendlycore-xenial-3.4-armhf-YYYYMMDD.img.gz	xenial	FriendlyCore系统固件, 基于Ubuntu core 16.04构建, 内置Qt4.8	3.4.y
	h3-XYZ-friendlywrt-4.14-armhf-YYYYMMDD.img.gz	19.07.1	基于OpenWrt构建	4.14.y
Other Image				
	sun8iw7p1_android_h3_uart0.img.zip	Android4.4.2	Android系统, 仅支持SD卡启动	3.4.y
	h3-XYZ-multiple-os-YYYYMMDD-25g.img.gz	-	内含了多个操作系统的eMMC烧写文件, 方便测试各个OS	

4.2.1.2 工具软件(可选)

访问此处的下载链接 (<http://download.friendlyelec.com/NanoPiM1>) 下载所需要的工具软件 (位于网盘的"05_工具软件"目录).

文件名	描述
win32diskimager.rar	用于将映象文件写入SD卡
SD Card Formatter	用于清空SD卡中的引导数据

4.2.2 Linux-3.4和Linux-4.14系统固件差异

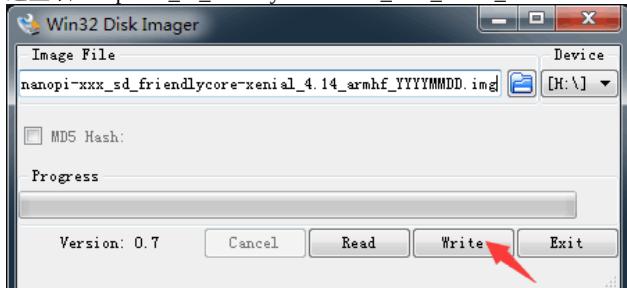
- Linux-3.4 为CPU芯片厂商全志科技官方提供的内核，全志为该内核做了很多的定制开发，所以该内核完善度高但是不够纯净，对应的系统固件发热量相对而言较大。如果您的产品必须使用VPU或者GPU功能的话，目前唯一的选择就是采用Linux-3.4内核的ROM，并且建议选购散热片。
- Linux-4.14 基于Linus Torvalds主线内核改造，并且尽可能地保持和主线内核一致，拥有和主线内核一致的特性，是一个非常纯净的内核，对应的系统固件发热量较小，稳定性更高。如果您的产品不需要使用VPU和GPU功能，强烈推荐使用该内核。
- 关于Linux-4.14内核，更详细的信息可以参考: Building U-boot and Linux for H5/H3/H2+/zh

4.2.3 烧写Linux系统

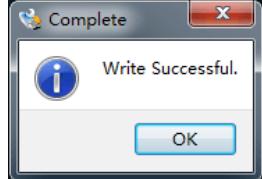
4.2.3.1 烧写到TF卡

- FriendlyCore / FriendlyWrt 等系统都属于 Linux 系统，所以它们的烧写方法是一样的。
- 将 Linux 系统固件和烧写工具 win32diskimager.rar 分别解压，在 Windows 下插入TF卡（限4G及以上的卡），以管理员身份运行烧写工具 win32diskimager，在烧写工具 win32diskimager 的界面上，选择你的TF卡盘符，选择Linux系统固件，点击 Write 按钮烧写。

这里以 nanopi-m1_sd_friendlycore-xenial_4.14_armhf_YYYYMMDD.img 为例，其他系统的烧写操作是类似的，烧写时的界面如下：



成功烧写后，会看到如下界面：

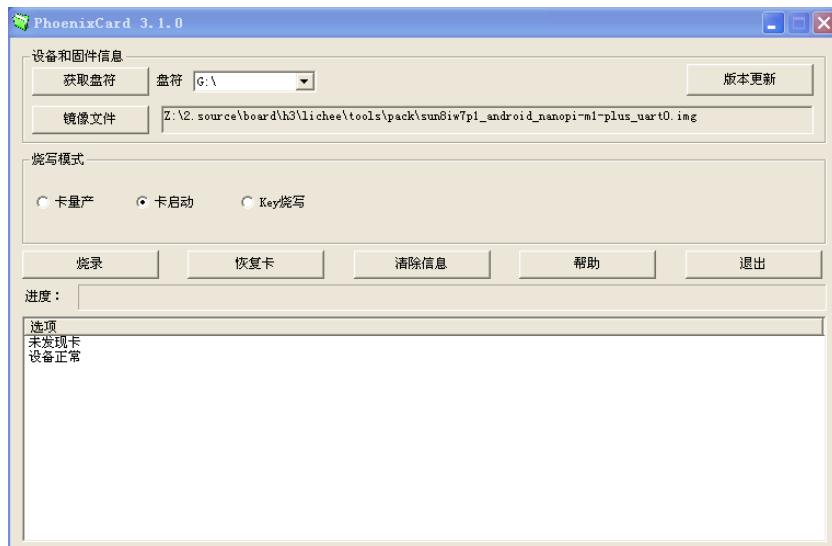


- 当制作完成TF卡后，拔出TF卡插入 BOOT 卡槽，上电启动（注意，这里需要5V/2A的供电），你可以看到STAT灯闪烁，这时你已经成功启动系统。

4.2.4 烧写Android系统

4.2.4.1 烧写到TF卡

- 烧写全志芯片的Android系统固件前，必须先格式化TF卡。在Windows系统下以管理员权限运行HDDLLF.4.40软件，并且格式化SD卡，格式化后把卡从电脑拔出来，再把卡插入电脑，使用Windows自带的格式化程序把TF卡格式化成FAT32格式，格式化后把卡拔出来；
- 将Android系统固件和烧写工具PhoenixCard_V310.rar分别解压，在Windows系统下插入TF卡（限4G及以上的卡）。以管理员身份运行PhoenixCard，在PhoenixCard的界面上，选择你的TF卡盘符，镜像文件选择为Android系统固件，烧写模式选择卡启动，点击 烧录 按钮烧写即可。



(图片里以NanoPi M1 Plus的Android固件为例，请注意使用正确的系统固件。)

- 当制作完成TF卡后，拔出TF卡插入BOOT卡槽，上电启动（注意，这里需要5V/2A的供电），你可以看到PWR灯常亮以及STAT灯闪烁，这时你已经成功启动Android系统。

5 Debian系统的使用

5.1 连接有线网络

- Debian系统在启动前，只要接上网线，系统启动后则会自动分配IP地址，不需要额外去配置。

5.2 连接无线网络

Debian系统使用NetworkManager来管理网络。

在Debian的桌面环境下，点击桌面任务栏右下角的网络图标，会弹出 NetworkManger 的菜单，列出当前的网络连接状态，如果有WiFi网络，会列出周边的无线热点，如下图所示：



你可以点击菜单上的无线热点，即可连接到该热点，如果热点是加密的，会弹出密码输入框提示你输入密码。
想进一步了解网络连接相关的内容，可参考这个页面：NetworkManager。

无论是SD WiFi还是USB WiFi, 它们的连接方式都是一样的。正基科技的APXX系列芯片属于SD WiFi, 另外系统默认也已经支持市面上众多常见的USB WiFi, 已测试过的USB WiFi型号如下:

序号	型号
1	RTL8188CUS/8188EU 802.11n WLAN Adapter
2	RT2070 Wireless Adapter
3	RT2870/RT3070 Wireless Adapter
4	RTL8192CU Wireless Adapter
5	小米WiFi mt7601
6	5G USB WiFi RTL8821CU
7	5G USB WiFi RTL8812AU

目前使用 NetworkManager 工具来管理网络, 其在命令行下对应的命令是 nmcli, 要连接WiFi, 相关的命令如下:

- 切换到root账户

```
$ su root
```

- 查看网络设备列表

```
$ nmcli dev
```

注意, 如果列出的设备状态是 unmanaged 的, 说明网络设备不受NetworkManager管理, 你需要清空 /etc/network/interfaces 下的网络设置, 然后重启.

- 开启WiFi

```
$ nmcli r wifi on
```

- 扫描附近的 WiFi 热点

```
$ nmcli dev wifi
```

- 连接到指定的 WiFi 热点

```
$ nmcli dev wifi connect "SSID" password "PASSWORD" ifname wlan0
```

请将 SSID 和 PASSWORD 替换成实际的 WiFi 名称和密码。

连接成功后, 下次开机, WiFi 也会自动连接。

更详细的NetworkManager使用指南可参考这篇文章: Use NetworkManager to configure network settings

如果你的USB WiFi无法正常工作, 大概率是因为文件系统里缺少了对应的USB WiFi固件。对于Debian系统, 可以在Debian-WiFi (<https://wiki.debian.org/WiFi>)里找到并安装USB WiFi芯片的固件。而对于Ubuntu系统, 则可以通过下列命令安装所有的USB WiFi固件:

```
$ apt-get install linux-firmware
```

一般情况下, 各种WiFi芯片的固件都存放在/lib/firmware 目录下。

5.3 安装Debian软件包

我们提供的是标准的Debian jessie系统, 你可以使用apt-get等命令来安装软件包, 如果板子是首次运行, 需要先用以下命令更新软件包列表:

```
apt-get update
```

然后就可以安装软件包了, 例如要安装ftp服务器, 使用以下命令:

```
apt-get install vsftpd
```

如果软件包下载速度不理想, 你可以编辑 /etc/apt/sources.list 更换一个更快的源服务器, 这个网址[1] (<http://www.debian.org/mirror/list>)有一份完整的源镜像服务器列表, 注意要选用一个带armhf架构的。

5.4 选择系统默认音频设备

如果当前系统存在多个音频设备, 例如HDMI-Audio、3.5mm耳机座、I2S-Codec时, 可以通过下列操作设置系统默认使用的音频设备。

- 启动板子后, 执行以下步骤安装alsa包:

```
$ apt-get update
$ apt-get install libasound2
$ apt-get install alsa-base
$ apt-get install alsa-utils
```

- 安装好需要的库后，查看系统当前所有的声卡设备的序列号。这里假设aplay的输出如下，并不是真实情况，请根据实际情况进行相对应的修改：

```
$ aplay -l
card 0: HDMI
card 1: 3.5mm codec
card 2: I2S codec
```

上面的信息表示card 0代表HDMI-Audio，card 1代表3.5mm耳机座, card 2代表I2S-Codec，修改配置文件/etc/asound.conf如下表示选择HDMI-Audio:

```
pcm.!default {
    type hw
    card 0
    device 0
}

ctl.!default {
    type hw
    card 0
}
```

如果将card 0修改为card 1，则表示选择3.5mm耳机座，以此类推。

拷贝一首.wav 格式的音乐到开发板上，播放音乐：

```
$ aplay /root/Music/test.wav
```

可以听见从系统默认的音频设备里输出音频。

如果您使用的开发板是H3/H5/H2+系列并且使用的是主线内核，那么更简便的方法是使用npi-config。

5.5 通过VNC和SSH登录系统

如果你不想连接HDMI，可以使用手机或电脑到这里 (<http://www.realvnc.com/download/>) 下载并安装一个名为VNC Viewer的软件，用VNC连接到开发板，默认的端口号为1，密码为：fa123456。

以下是在iPhone上用VNC登录的画面：



如果你不想连接HDMI和串口模块，可以通过SSH协议登录系统。假设通过路由器查看到开发板的IP地址为192.168.1.230，你可以在PC机上执行如下命令登录系统：

```
$ ssh root@192.168.1.230
```

密码为fa。

5.6 连接USB摄像头模块(FA-CAM202)

FA-CAM202是一款200万像素的USB摄像头模块。

连接摄像头后，在Debian系统点击左下角的菜单键“Other”-->xawtv，打开USB Camera软件。进入“welcome to xawtv！”，选择OK即可进行拍照。

5.7 连接摄像头测试OpenCV

OpenCV的全称是Open Source Computer Vision Library，是一个跨平台的计算机视觉库。

执行以下步骤测试OpenCV：

- 连接网线，然后启动系统，在HDMI界面下进行登录操作。
- 安装opencv库，执行命令：

```
$ apt-get update
$ apt-get install libcv-dev libopencv-dev
```

- 参考前面章节，确保摄像头工作正常：
- 运行OpenCV官方C++示例代码，执行下列命令编译运行：

```
$ cd /home/fa/Documents/opencv-demo
$ make
$ ./demo
```

5.8 连接DVP摄像头模块(CAM500B)

对于NanoPi-M1，CAM500B可配合Linux-3.4内核和Linux-4.14内核使用。
CAM500B是一款500万像素摄像头模块，以DVP并行信号输出，详细信息请参考Matirx-CAM500B的介绍章节。

连接开发板和摄像头，然后上电启动系统，连接网络，以root用户登录终端并编译运行mjpg-streamer:

```
$ cd /root/C/mjpg-streamer
$ make
$ ./start.sh
```

请自行修改start.sh, 确保使用正确的/dev/videoX节点, 下列命令可以用来确定摄像头的video节点:

```
$ apt-get install v4l-utils
$ v4l2-ctl -d /dev/video0 -D
Driver Info (not using libv4l2):
    Driver name   : sun6i-video
    Card type     : sun6i-csi
    Bus info      : platform:camera
    Driver version: 4.14.0
    ...

```

上述信息表示/dev/video0是摄像头的设备节点。mjpg-streamer是一个开源的网络视频流服务器，在板子上成功运行mjpg-streamer后会打印下列信息:

```
$ ./start.sh
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 1280 x 720
i: Frames Per Second.: 30
i: Format.....: YUV
i: JPEG Quality.....: 90
o: www-folder-path...: ./www/
o: HTTP TCP port.....: 8080
o: username:password.: disabled
o: commands.....: enabled
```

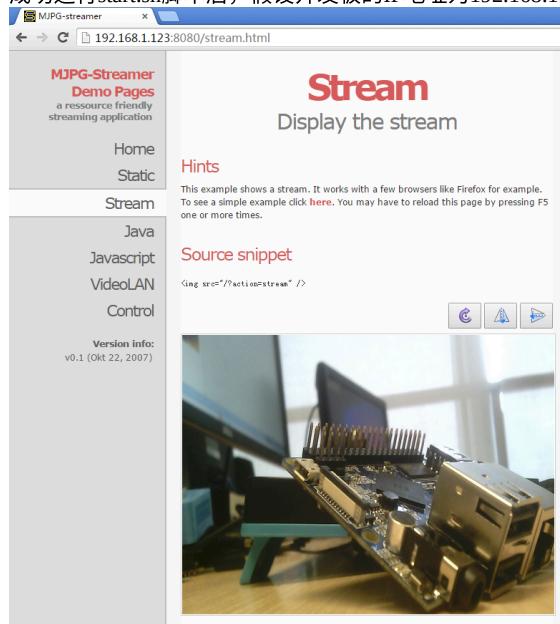
start.sh脚本里执行了下列2个命令:

```
export LD_LIBRARY_PATH=$(pwd)
./mjpg_streamer -i "./input_uvc.so -d /dev/video0 -y 1 -r 1280x720 -f 30 -q 90 -n -fb 0" -o "./output_http.so -w ./www"
```

mjpg_streamer相关参数的含义如下:

- i: 选择输入插件, input_uvc.so表示从摄像头采集数据;
- o: 选择输出插件, output_http.so表示使用http协议传输数据;
- d: 输入插件的子参数, 指定摄像头设备节点;
- y: 输入插件的子参数, 指定摄像头采集数据的格式, 1:yuyv, 2:yyvu, 3:uyvy 4:vyuy, 如果不使用-y参数, 则表示采集MJPEG格式;
- r: 输入插件的子参数, 指定摄像头采集分辨率;
- f: 输入插件的子参数, 指定使用的摄像头采集fps, 具体是否支持依赖于驱动;
- q: 输入插件的子参数, 指定libjpeg软编码的图像质量;
- n: 输入插件的子参数, 禁止dynctrls功能;
- fb: 输入插件的子参数, 指定是否在/dev/fbX上显示采集的图像;
- w: 输出插件的子参数, 指定包含网页的目录;

成功运行start.sh脚本后, 假设开发板的IP地址为192.168.1.230, 在PC的浏览器中输入 192.168.1.230:8080 就能浏览摄像头采集的画面了, 效果如下:



mjpg-streamer是用libjpeg对摄像头数据进行软编码, Linux-4.14内核的ROM目前并不支持视频硬编码, 但是如果使用H3板子 + Linux-3.4内核的ROM的话, 可以使用ffmpeg对摄像头数据进行硬编码, 这样能大大降低CPU的占用率并提高编码速度:

```
$ ffmpeg -t 30 -f v4l2 -channel 0 -video_size 1280x720 -i /dev/video0 -pix_fmt nv12 -r 30 -b:v 64k -c:v cedrus264 test.mp4
```

默认会录制30秒的视频，输入q能终止录制。录制完成后会在当前目录生成一个名为test.mp4的视频文件，可将其拷贝到PC上进行播放验证。

5.9 命令行查看CPU工作温度

在串口终端执行如下命令，可以快速地获取CPU的当前温度和运行频率等信息：

```
cpu_freq
```

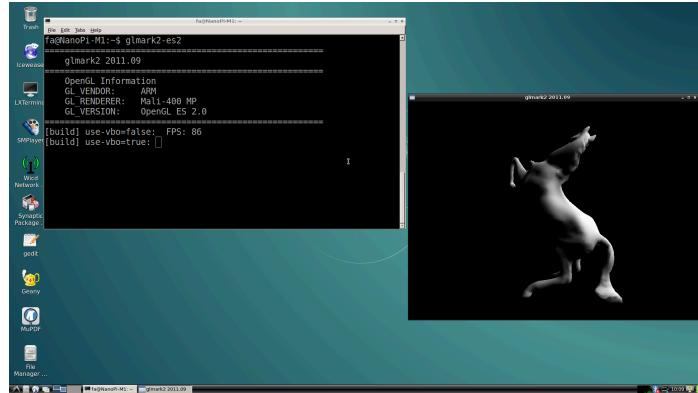
5.10 测试GPU

注意: 该功能仅支持使用Linux-3.4.y的系统固件。

启动系统，在HDMI界面下进行登录操作，打开终端并运行命令：

```
$ glmark2-es2
```

测试效果如下：



5.11 测试VPU

注意: 该功能仅支持使用Linux-3.4.y的系统固件。

访问此处下载地址 (<https://pan.baidu.com/s/1dF7HL0P>)的test-video目录下载视频文件，启动系统，在HDMI界面下登录系统，打开终端并运行命令：

```
$ sudo apt-get install mpv
$ video_play mpv ./big_buck_bunny_1080p_H264_AAC_25fps_7200K.MP4
```

经测试，可流畅硬解播放1080p视频。

6 FriendlyCore的使用

6.1 介绍

FriendlyCore，是一个没有X-windows环境，基于Ubuntu core构建的系统，使用Qt-Embedded作为图形界面的轻量级系统，兼容Ubuntu系统软件源，非常适合于企业用户用作产品的基础OS。

本系统除了保留Ubuntu Core的特性以外，还包括以下特性：

- 集成Qt4.8；
- 集成NetworkManager网络管理器；
- 集成bluez等蓝牙相关软件包；
- 集成alsa相关软件包；
- 集成命令行系统配置工具npi-config；
- 集成Python GPIO模块RPiGPIO；
- 集成Python/C语言编写的demo程序，位于/root目录；
- 使能512M的swap分区；

6.2 运行FriendlyCore

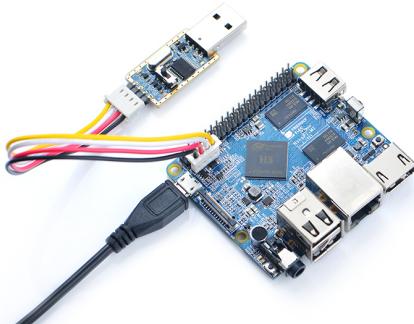
- 对于有HDMI接口的板子，如果要在电视上进行操作，您需要连接USB鼠标和键盘。
- 如果您需要进行内核开发，最好选购一个串口配件，连接了串口，则可以通过串口终端对开发板进行操作。

使用串口模块能有效地提升开发效率，以下是串口模块的连接方法：

接上串口后，您可以选择从串口模块的DC口或者从MicroUSB口（如果有）进行供电：



也可以使用USB转串口模块调试，请注意需要使用5V/2A电源给开发板MicroUSB供电：



- FriendlyCore默认帐户：

普通用户：

```
用户名: pi
密码: pi
```

Root用户：

```
用户名: root
密码: fa
```

默认会以 pi 用户自动登录，你可以使用 sudo npi-config 命令取消自动登录。

- 更新软件包：

```
$ sudo apt-get update
```

6.3 开发Qt应用

请参考 How to Build and Install Qt Application for FriendlyELEC Boards/zh

6.4 开机自动运行Qt示例程序

使用npi-config工具进行开启：

```
sudo npi-config
```

进入Boot Options -> Autologin -> Qt/Embedded，选择Enable然后重启即可。

6.5 扩展TF卡文件系统

第一次启动FriendlyCore系统时，系统会自动扩展文件系统分区，请耐心等待，TF卡/eMMC的容量越大，需要等待的时间越长，进入系统后执行下列命令查看文件系统分区大小：

```
df -h
```

6.6 连接WiFi

无论是SD WiFi还是USB WiFi, 它们的连接方式都是一样的。正基科技的APXX系列芯片属于SD WiFi, 另外系统默认也已经支持市面上众多常见的USB WiFi, 已测试过的USB WiFi型号如下:

序号	型号
1	RTL8188CUS/8188EU 802.11n WLAN Adapter
2	RT2070 Wireless Adapter
3	RT2870/RT3070 Wireless Adapter
4	RTL8192CU Wireless Adapter
5	小米WiFi mt7601
6	5G USB WiFi RTL8821CU
7	5G USB WiFi RTL8812AU

目前使用 NetworkManager 工具来管理网络, 其在命令行下对应的命令是 nmcli, 要连接WiFi, 相关的命令如下:

- 切换到root账户

```
$ su root
```

- 查看网络设备列表

```
$ nmcli dev
```

注意, 如果列出的设备状态是 unmanaged 的, 说明网络设备不受NetworkManager管理, 你需要清空 /etc/network/interfaces下的网络设置,然后重启.

- 开启WiFi

```
$ nmcli r wifi on
```

- 扫描附近的 WiFi 热点

```
$ nmcli dev wifi
```

- 连接到指定的 WiFi 热点

```
$ nmcli dev wifi connect "SSID" password "PASSWORD" ifname wlan0
```

请将 SSID和 PASSWORD 替换成实际的 WiFi名称和密码。

连接成功后, 下次开机, WiFi 也会自动连接。

更详细的NetworkManager使用指南可参考这篇文章: Use NetworkManager to configure network settings

如果你的USB WiFi无法正常工作, 大概率是因为文件系统里缺少了对应的USB WiFi固件。对于Debian系统, 可以在Debian-WiFi (<https://wiki.debian.org/WiFi>)里找到并安装USB WiFi芯片的固件。而对于Ubuntu系统, 则可以通过下列命令安装所有的USB WiFi固件:

```
$ apt-get install linux-firmware
```

一般情况下, 各种WiFi芯片的固件都存放在/lib/firmware目录下。

6.7 连接以太网

默认插上网线开机, 会自动连接并通过DHCP获取IP地址, 如需要配置静态IP地址, 请参考 NetworkManager 的相关文档: Use NetworkManager to configure network settings。

6.8 访问GPIO/I2C/串口等硬件资源

请参考下面的文档:

- WiringPi: NanoPi NEO/NEO2/Air GPIO Programming with C/zh
- RPi.GPIO : NanoPi NEO/NEO2/Air GPIO Programming with Python/zh

6.9 定制命令行的欢迎信息 (文字LOGO)

欢迎信息主要是这个目录下的脚本来打印的:

```
/etc/update-motd.d/
```

比如要修改 FriendlyELEC 的大字LOGO，可以修改/etc/update-motd.d/10-header 这个文件，比如要将LOGO改为HELLO，可将以下行：

```
TERM=linux toilet -f standard -F metal $BOARD_VENDOR
```

改为：

```
TERM=linux toilet -f standard -F metal HELLO
```

6.10 修改时区

例如更改为Shanghai时区：

```
sudo rm /etc/localtime
sudo ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

6.11 选择系统默认音频设备

如果当前系统存在多个音频设备, 例如HDMI-Audio、3.5mm耳机座、I2S-Codec时, 可以通过下列操作设置系统默认使用的音频设备。

- 启动板子后, 执行以下步骤安装alsa包:

```
$ apt-get update
$ apt-get install libasound2
$ apt-get install alsa-base
$ apt-get install alsa-utils
```

- 安装好需要的库后, 查看系统当前所有的声卡设备的序列号。这里假设aplay的输出如下, 并不是真实情况, 请根据实际情况进行相对应的修改:

```
$ aplay -l
card 0: HDMI
card 1: 3.5mm codec
card 2: I2S codec
```

上面的信息表示card 0代表HDMI-Audio, card 1代表3.5mm耳机座, card 2代表I2S-Codec, 修改配置文件/etc/asound.conf如下表示选择HDMI-Audio:

```
pcm.!default {
    type hw
    card 0
    device 0
}

ctl.!default {
    type hw
    card 0
}
```

如果将card 0修改为card 1, 则表示选择3.5mm耳机座, 以此类推。

拷贝一首.wav 格式的音乐到开发板上, 播放音乐:

```
$ aplay /root/Music/test.wav
```

可以听见从系统默认的音频设备里输出音频。

如果您使用的开发板是H3/H5/H2+系列并且使用的是主线内核, 那么更简便的方法是使用npi-config。

6.12 连接DVP摄像头模块(CAM500B)

对于NanoPi-M1, CAM500B可配合Linux-3.4内核和Linux-4.14内核使用。

CAM500B是一款500万像素摄像头模块, 以DVP并行信号输出, 详细信息请参考Matirx-CAM500B的介绍章节。

连接开发板和摄像头, 然后上电启动系统, 连接网络, 以root用户登录终端并编译运行mjpg-streamer:

```
$ cd /root/C/mjpg-streamer
$ make
$ ./start.sh
```

请自行修改start.sh, 确保使用正确的/dev/videoX节点, 下列命令可以用来确定摄像头的video节点:

```
$ apt-get install v4l-utils
$ v4l2-ctl -d /dev/video0 -D
Driver Info (not using libv4l2):
  Driver name   : sun6i-video
  Card type     : sun6i-csi
  Bus info      : platform:camera
  Driver version: 4.14.0
  ...
```

上述信息表示/dev/video0是摄像头的设备节点。mjpg-streamer是一个开源的网络视频流服务器, 在板子上成功运行mjpg-streamer后会打印下列信息:

```
$ ./start.sh
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 1280 x 720
i: Frames Per Second.: 30
i: Format.....: YUV
i: JPEG Quality....: 90
o: www-folder-path...: ./www/
o: HTTP TCP port....: 8080
o: username:password.: disabled
o: commands.....: enabled
```

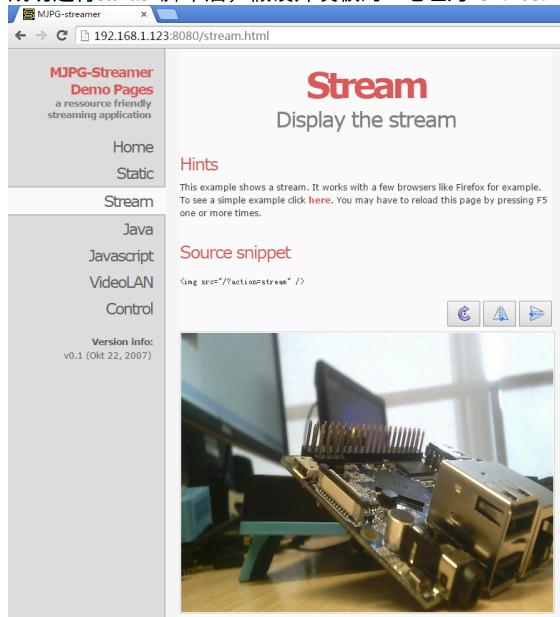
start.sh脚本里执行了下列2个命令:

```
export LD_LIBRARY_PATH=$(pwd)
./mjpg_streamer -i "./input_uvc.so -d /dev/video0 -y 1 -r 1280x720 -f 30 -q 90 -n -fb 0" -o "./output_http.so -w ./www"
```

mjpg_streamer相关参数的含义如下:

- i: 选择输入插件, input_uvc.so表示从摄像头采集数据;
- o: 选择输出插件, output_http.so表示使用http协议传输数据;
- d: 输入插件的子参数, 指定摄像头设备节点;
- y: 输入插件的子参数, 指定摄像头采集数据的格式, 1:yuv, 2:yyuv, 3:uyvy 4:vyuy, 如果不使用-y参数, 则表示采集MJPEG格式;
- r: 输入插件的子参数, 指定摄像头采集分辨率;
- f: 输入插件的子参数, 指定想使用的摄像头采集fps, 具体是否支持依赖于驱动;
- q: 输入插件的子参数, 指定libjpeg软编码的图像质量;
- n: 输入插件的子参数, 禁止dynctrls功能;
- fb: 输入插件的子参数, 指定是否在/dev/fbX上显示采集的图像;
- w: 输出插件的子参数, 指定包含网页的目录;

成功运行start.sh脚本后, 假设开发板的IP地址为192.168.1.230, 在PC的浏览器中输入 192.168.1.230:8080 就能浏览摄像头采集的画面了, 效果如下:



mjpg-streamer是用libjpeg对摄像头数据进行软编码, Linux-4.14内核的ROM目前并不支持视频硬编码, 但是如果使用H3板子 + Linux-3.4内核的ROM的话, 可以使用ffmpeg对摄像头数据进行硬编码, 这样能大大降低CPU的占用率并提高编码速度:

```
$ ffmpeg -t 30 -f v4l2 -channel 0 -video_size 1280x720 -i /dev/video0 -pix_fmt nv12 -r 30 -b:v 64k -c:v cedrus264 test.mp4
```

默认会录制30秒的视频, 输入q能终止录制。录制完成后会在当前目录生成一个名为test.mp4的视频文件, 可将其拷贝到PC上进行播放验证。

6.13 连接USB摄像头模块(FA-CAM202)

FA-CAM202是一款200万像素的USB摄像头模块, 连接开发板和摄像头, 然后上电启动系统, 连接网络, 以root用户登录终端并编译运行mjpg-streamer:

```
$ cd /root/C/mjpg-streamer
$ make
$ ./start.sh
```

请自行修改start.sh, 确保使用正确的/dev/videoX节点, 下列命令可以用来确定摄像头的video节点:

```
$ apt-get install v4l-utils
$ v4l2-ctl -d /dev/video0 -D
# fa-cam202有2个型号
Driver Info (not using libv4l2):
  Driver name   : uvcvideo
  Card type     : HC 3358+2100: HC 3358+2100 / USB 2.0 Camera: USB 2.0 Camera
  Bus info      : usb-1c1b000.usb-1
...
```

上述信息表示/dev/video0是摄像头的设备节点。mjpg-streamer是一个开源的网络视频流服务器, 在板子上成功运行mjpg-streamer后会打印下列信息:

```
$ ./start.sh
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 1280 x 720
i: Frames Per Second.: 30
i: Format.....: YUV
i: JPEG Quality....: 90
o: www-folder-path...: ./www/
o: HTTP TCP port....: 8080
o: username:password.: disabled
o: commands.....: enabled
```

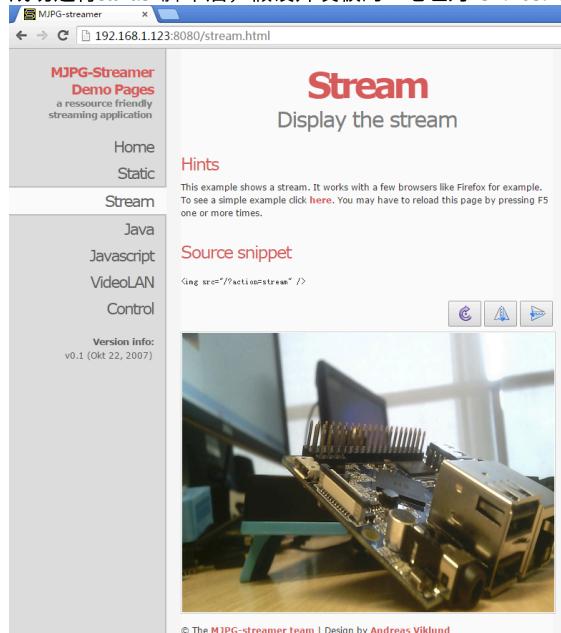
start.sh脚本里执行了下列2个命令:

```
export LD_LIBRARY_PATH=$(pwd)
./mjpg_streamer -i "./input_uvc.so -d /dev/video0 -y 1 -r 1280x720 -f 30 -q 90 -n -fb 0" -o "./output_http.so -w ./www"
```

mjpg_streamer相关参数的含义如下:

- i: 选择输入插件, input_uvc.so表示从摄像头采集数据;
- o: 选择输出插件, output_http.so表示使用http协议传输数据;
- d: 输入插件的子参数, 指定摄像头设备节点;
- y: 输入插件的子参数, 指定摄像头采集数据的格式, 1:yuv, 2:yyuv, 3:uyvy 4:vyuy, 如果不使用-y参数, 则表示采集MJPEG格式;
- r: 输入插件的子参数, 指定摄像头采集分辨率;
- f: 输入插件的子参数, 指定想使用的摄像头采集fps, 具体是否支持依赖于驱动;
- q: 输入插件的子参数, 指定libjpeg软编码的图像质量;
- n: 输入插件的子参数, 禁止dynctrls功能;
- fb: 输入插件的子参数, 指定是否在/dev/fbX上显示采集的图像;
- w: 输出插件的子参数, 指定包含网页的目录;

成功运行start.sh脚本后, 假设开发板的IP地址为192.168.1.230, 在PC的浏览器中输入 192.168.1.230:8080 就能浏览摄像头采集的画面了, 效果如下:



6.14 查看CPU温度和频率

命令行查看:

```
$ cat /proc/cpuinfo | grep freq
Available frequency(KHz):
    480000 624000 816000 1008000
Current frequency(KHz):
    CPU0 online=1 temp=26548C governor=ondemand freq=624000KHz
    CPU1 online=1 temp=26548C governor=ondemand freq=624000KHz
    CPU2 online=1 temp=26548C governor=ondemand freq=624000KHz
    CPU3 online=1 temp=26548C governor=ondemand freq=624000KHz
```

上述信息表示当前有4个CPU核在线, 温度均约为26.5摄氏度, 运行的策略均为根据需求来决定运行频率, 当前的运行频率均为624MHz, 设置频率的命令如下:

```
$ cpufreq-set -s 1008000
Available frequency(KHz):
    480000 624000 816000 1008000
Current frequency(KHz):
    CPU0 online=1 temp=36702C governor=userspace freq=1008000KHz
    CPU1 online=1 temp=36702C governor=userspace freq=1008000KHz
    CPU2 online=1 temp=36702C governor=userspace freq=1008000KHz
    CPU3 online=1 temp=36702C governor=userspace freq=1008000KHz
```

上述命令将4个CPU核的频率设置为1008MHz。

6.15 运行Qt示例程序

执行以下命令：

```
$ sudo /opt/QtE-Demo/run.sh
```

运行结果如下，这是一个开源的QtDemo (<https://github.com/friendlyarm/QtE-Demo>):



6.16 Docker在armhf系统下的安装与使用

6.16.1 安装 Docker

执行下列命令：

```
sudo apt-get update
sudo apt-get install docker.io
```

6.16.2 测试 Docker

执行下列命令运行一个简单的docker image:

```
git clone https://github.com/friendlyarm/debian-jessie-arm-docker
cd debian-jessie-arm-docker
./rebuild-image.sh
./run.sh
```

7 FriendlyWrt的使用

7.1 介绍

FriendlyWrt 基于 OpenWrt 改造而来，它是适合于嵌入式设备的一个 Linux 发行版，它不是一个单一、静态的固件，而是提供了一个可添加软件包的可写的文件系统。这使用户可以自由的选择应用程序和配置，而不必受设备提供商的限制，并且可以使用一些适合某方面应用的软件包来定制你的设备。对于开发者来说，OpenWrt 是一个框架，开发者不必麻烦地构建整个固件就能得到想要的应用程序；对于用户来说，这意味着完全定制的能力，与以往不同的方式使用设备，OPKG 包含超过3500个软件。更详细的介绍请参考OpenWrt官网 (<https://openwrt.org/>)。

7.2 登录系统

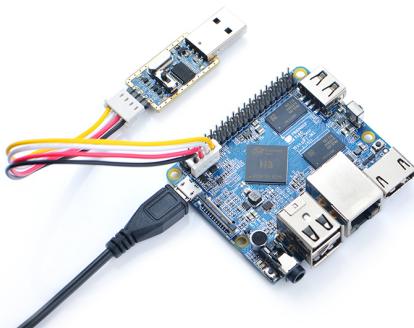
- 串口登录

如果您需要进行内核开发，最好选购一个串口配件，连接了串口，则可以通过串口终端对开发板进行操作。使用串口模块能有效地提升开发效率，以下是串口模块的连接方法：

接上串口后，您可以选择从串口模块的DC口或者从MicroUSB口（如果有）进行供电：



也可以使用USB转串口模块调试，请注意需要使用5V/2A电源给开发板MicroUSB供电：



默认会以 root 用户自动登录，并且没有设置root用户的密码，你可以使用 passwd 命令来设置 root 用户的密码。

```
BusyBox v1.28.3 () built-in shell (ash)

[   ] [   ] [   ] [   ] [   ] [   ]
| W I R E L E S S F R E E D O M |
[   ] [   ] [   ] [   ] [   ] [   ]

OpenWrt 18.06.1, r7258-5eb055306f

==== WARNING! =====
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.

root@OpenWrt:/#
```

第一次运行系统时，系统会自动拓展TF卡上文件系统分区到最大可用空间：

```
Begin: Resizing ext4 file system on /dev/mmcblk0p3 ... Model: SD SR64G (sd/mmc)
Disk /dev/mmcblk0: 100%
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
1 0.04% 0.11% 0.07% primary fat16
2 0.11% 0.53% 0.42% primary ext4
3 0.53% 100% 99.5% primary ext4

resize2fs 1.44.1 (24-Mar-2018)
[ 29.750417] random: crng init done
Resizing the filesystem on /dev/mmcblk0p3 to 62040064 (1k) blocks.
The filesystem on /dev/mmcblk0p3 is now 62040064 (1k) blocks long.
```

请耐心等待文件系统扩展完成。

■ SSH登录

在本开发板的FriendlyWrt系统里，有线网络(eth0)被配置为WAN功能。

启动系统前，请先用网线连接板子的有线以太网口到一级路由器的LAN口，以便该以太网(eth0)能通过DHCP获取到IP地址。

这里假设已经通过调试串口或者一级路由器确定板子的以太网(eth0)的IP地址为192.168.1.163，执行下列命令通过SSH登录系统：

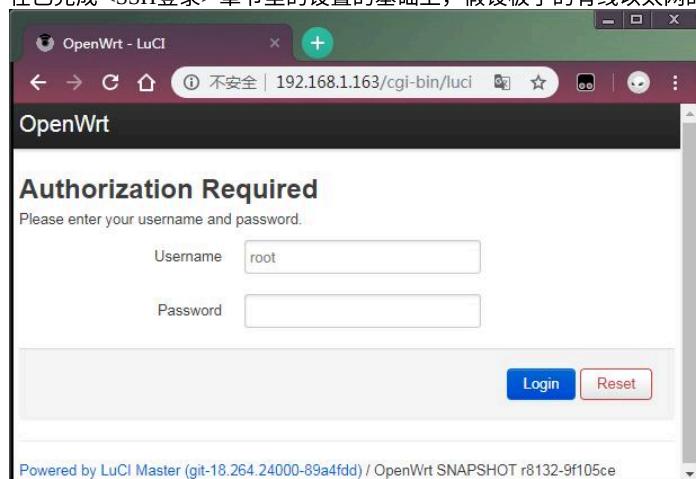
```
$ ssh root@192.168.1.163
```

无需密码，直接就可以登录。

■ Web登录

FriendlyWrt系统支持通过LuCI Web界面进行访问和配置。

在已完成 <SSH登录> 章节里的设置的基础上，假设板子的有线以太网的IP地址为192.168.1.163，在浏览器中输入该IP地址就可以登录LuCI界面了：



默认用户名为root，无需密码，直接点击"Login"按键即可登录。

7.3 软件包管理

FriendlyWrt使用opkg工具来管理软件包，执行如下命令可以获取opkg的帮助信息：

```
$ opkg
Package Manipulation:
  update           Update list of available packages
  upgrade <pkgs>    Upgrade packages
  install <pkgs>     Install package(s)
  configure <pkgs>   Configure unpacked package(s)
  remove <pkgs|regexp> Remove package(s)
  flag <flag> <pkgs>  Flag package(s)
  <flag>=hold|noprune|user|ok|installed|unpacked (one per invocation)

Informational Commands:
  list             List available packages
  list-installed   List installed packages
  list-upgradable  List installed and upgradable packages
  list-changed-conffiles List user modified configuration files
  files <pkg>       List files belonging to <pkg>
  search <file|regexp> List package providing <file>
  find <regexp>      List packages whose name or description matches <regexp>
  info [pkg|regexp]  Display all info for <pkg>
  status [pkg|regexp] Display all status for <pkg>
  download <pkg>     Download <pkg> to current directory
...
```

上面只截取了部分帮助信息，请自行查阅完整的帮助信息，下面会描述几个常用的opkg命令。

- 更新可用软件包列表

第一次安装软件前，建议先更新可用软件包列表：

```
$ opkg update
```

- 查看可安装的软件包：

```
$ opkg list
```

本WiKi编写时，可安装的软件包共有3241个。

- 查看已安装的软件：

```
$ opkg list-installed
```

本WiKi编写时，已安装的软件包共有124个。

- 安装/删除软件：

```
$ opkg install <pkgs>
$ opkg remove <pkgs>
```

- 查看已安装的软件包含什么文件：

```
$ opkg files <pkg>
```

- 安装LuCI中文语言包：

```
$ opkg install luci-i18n-base-zh-cn
```

- 查看当前系统中哪些配置文件被修改过：

```
$ opkg list-changed-conffiles
```

- 相关参考：

- openwrt opkg (<https://openwrt.org/docs/guide-user/additional-software/opkg>)

7.4 查看系统状态

- 命令行查看CPU温度和频率

```
$ cpu_freq
Available frequency(KHz):
  480000 624000 816000 1008000
Current frequency(KHz):
  CPU0 online=1 temp=26548C governor=ondemand freq=624000KHz
  CPU1 online=1 temp=26548C governor=ondemand freq=624000KHz
  CPU2 online=1 temp=26548C governor=ondemand freq=624000KHz
  CPU3 online=1 temp=26548C governor=ondemand freq=624000KHz
```

上述信息表示当前有4个CPU核在线, 温度均约为26.5摄氏度, 运行的策略均为根据需求来决定运行频率, 当前的运行频率均为624MHz, 设置频率的命令如下:

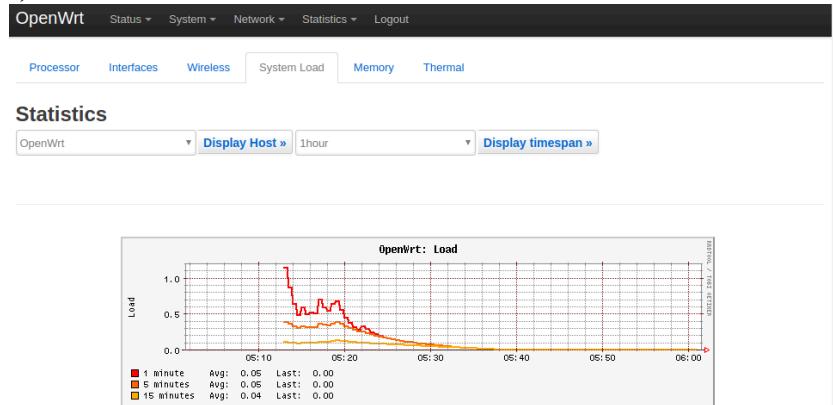
```
$ cpu_freq -s 1008000
Available frequency(KHz):
    480000 624000 816000 1008000
Current frequency(KHz):
    CPU0 online=1 temp=36702C governor=userspace freq=1008000KHz
    CPU1 online=1 temp=36702C governor=userspace freq=1008000KHz
    CPU2 online=1 temp=36702C governor=userspace freq=1008000KHz
    CPU3 online=1 temp=36702C governor=userspace freq=1008000KHz
```

上述命令将 4 个CPU核的频率设置为1008MHz。

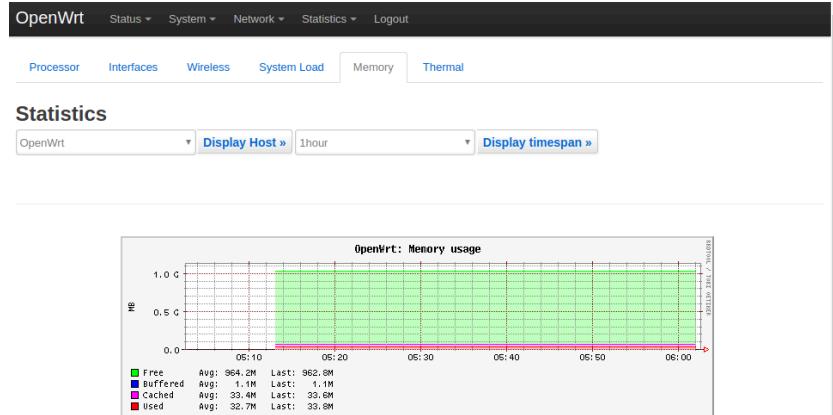
■ LuCI Web界面查看系统状态

登录LuCI界面后, 点击顶部的 Statistics ---> Graphs, 可以查看系统的各种软硬件状态信息(即statistics), 例如:

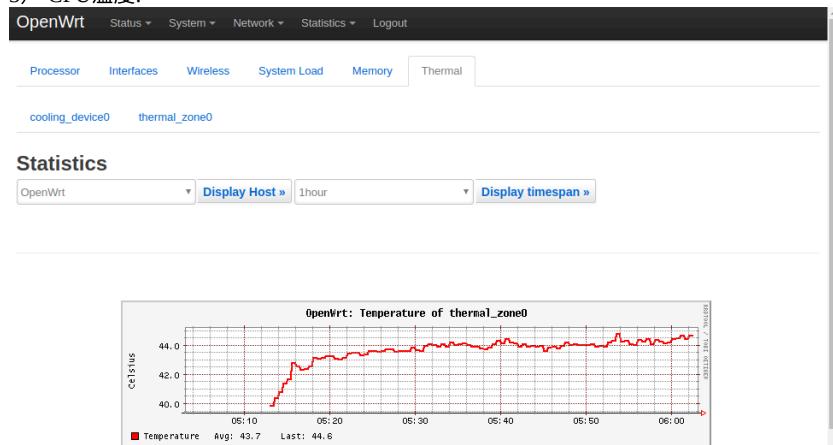
1) 系统负载:



2) 内存:



3) CPU温度:



Statistics界面对应软件包luci-app-statistics, luci-app-statistics软件包用Collectd工具收集状态数据并且用RRDtool工具将数据渲染为图表。你可以通过安装额外的collectd-mod-*软件包去使能更多的statistics. 所有的collectd-mod-*软件包对应同一个配置文件: /etc/config/luci_statistics。

■ 相关参考:

- openwrt luci_app_statistics (https://openwrt.org/docs/guide-user/luci/luci_app_statistics)
- openwrt statistics.chart.public (<https://openwrt.org/docs/guide-user/luci/statistics.chart.public>)
- openwrt statistic.custom (https://openwrt.org/docs/guide-user/perf_and_log/statistic.custom)

7.5 查看Network->Interfaces的配置

- 登录LuCI界面后, 点击顶部的 Network ---> Interfaces , 可以查看当前的网络设置:

No password set!
There is no password set on this router. Please configure a root password to protect the web interface and enable SSH.

[Go to password configuration...](#)

WAN

Interfaces

WAN	Protocol: DHCP client Uptime: 0h 2m 7s MAC: 02:81:77:9E:70:64 RX: 73.44 KB (617 Pkts.) TX: 182.75 KB (492 Pkts.) IPv4: 192.168.1.144/24	Restart Stop Edit Delete
-----	--	--

[Add new interface...](#)

Global network options

IPv6 ULA-Prefix: fdff:45a0:1964::/48

[Save & Apply](#) [Save](#) [Reset](#)

- Network -> Interfaces 界面的配置保存在/etc/config/network中。

7.6 使用USB WiFi

目前仅支持 RTL8821CU，即插即用，插入后默认设置为AP模式，热点名称为 "rtl8821cu-mac地址"，密码为"password";

7.7 使用华为随行WiFi 2 mini(E8372H-155)

即插即用，插入后，热点名称为 "HUAWEI-8DA5"，其他设备连接上该热点后就可以使用4G的流量上网了。

8 更多OS

8.1 DietPi



DietPi
#Lightweight.Justice for your SBC

DietPi is a highly optimised & minimal Debian-based Linux distribution. DietPi is extremely lightweight at its core, and also extremely easy to install and use. Setting up a single board computer (SBC) or even a computer, for both regular or server use, takes time and skill. DietPi provides an easy way to install and run favourite software you choose.

For more information, please visit this link <https://dietpi.com/docs/>.

DietPi supports many of the NanoPi board series, you may download the image file from here:

- <https://dietpi.com/docs/hardware/#nanopi-series-friendlyarm>

9 如何编译Linux 系统

9.1 使用Linux-4.14 BSP

M1支持使用Linux-4.14内核，Linux-4.14内核主要由开源社区完善，在此基础上友善官方进行自家硬件的适配。
关于H3芯片系列开发板使用主线U-boot和Linux-4.14的方法，请参考维基：[Building U-boot and Linux for H5/H3/H2+](#)

9.2 使用Linux-3.4 BSP

Linux3.4 BSP是由H3芯片厂商全志科技提供，在此基础上友善官方进行了自家硬件的适配。

9.2.1 准备工作

从github上克隆lichee源码，：

```
$ git clone https://github.com/friendlyarm/h3_lichee.git lichee --depth 1
```

注：lichee是全志为其CPU的板级支持包所起的项目名称，里面包含了U-boot，Linux等源码和众多的编译脚本。因为原厂的编译打包脚本依赖lichee这个路径，目录名称必须保持为lichee。

9.2.2 安装交叉编译器

访问此处下载地址 (<http://download.friendlyelec.com/nanopim1>)的toolchain目录，下载交叉编译器gcc-linaro-arm.tar.xz，将该压缩包放置在lichee/brandy/toolchain/目录下即可，无需解压。

9.2.3 编译lichee源码

编译全志 H3 的BSP源码包必须使用64bit的Linux PC系统，并安装下列软件包，下列操作均基于Ubuntu-14.04 LTS-64bit：

```
$ sudo apt-get install gawk git gnupg flex bison gperf build-essential \
zip curl libcurl4-openssl-dev:i386 x11proto-core-dev \
libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
libgl1-mesa-dev g++-multilib mingw32 tofrodos \
python-markdown libxml2-utils xscreensaver zlib1g-dev:i386
```

编译lichee源码包，执行命令：

```
$ cd lichee/fa_tools
$ ./build.sh -b nanopi-m1 -p linux -t all
```

该命令会一次性编译好U-boot、Linux内核和模块。

lichee目录里内置了交叉编译器，当进行源码编译时，会自动使用该内置的编译器，所以无需手动安装编译器。

下列命令可以更新TF卡上的U-boot：

```
$ cd lichee/fa_tools/
$ ./fuse.sh -d /dev/sdX -p linux -t u-boot
```

/dev/sdX请替换为实际的TF卡设备文件名。

内核boot.img和驱动模块均位于linux-3.4/output目录下，将boot.img拷贝到TF卡的boot分区的根目录即可更新内核。

9.2.4 编译U-boot

注意：必须先完整地编译整个lichee目录后，才能进行单独编译U-boot的操作。

如果你想单独编译U-boot，可以执行命令：

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-m1 -p linux -t u-boot
```

下列命令可以更新TF卡上的U-boot：

```
$ cd lichee/fa_tools/
$ ./fuse.sh -d /dev/sdX -p linux -t u-boot
```

/dev/sdX请替换为实际的TF卡设备文件名。

9.2.5 编译Linux内核

注意：必须先完整地编译整个lichee目录后，才能进行单独编译Linux内核的操作。

如果你想单独编译Linux内核，可以执行命令：

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-m1 -p linux -t kernel
```

编译完成后内核boot.img和驱动模块均位于linux-3.4/output目录下，将boot.img拷贝到TF卡的boot分区的根目录即可。

9.2.6 清理lichee源码

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-m1 -p linux -t clean
```

10 Android系统的使用

10.1 连接USB WiFi

Android系统目前仅支持型号为rtl8188etv/rtl8188eu的USB WiFi，即插即用。

10.2 使用红外遥控器(RC-100)

启动Android系统后，可用红外遥控器(型号为RC-100)进行远程操控。

RC-100上的按键功能如下：

按键名称	按键功能
POWER	开机/关机
F1	搜索
F2	打开浏览器
F3	进入/退出鼠标模式
UP	向上移动
DOWN	向下移动
LEFT	向左移动
RIGHT	向右移动
OK	确认
音量-	减小音量
音量静音	静音
音量+	增大音量
SETTING	打开设置
HOME	回到主界面
BACK	返回上一个界面

Android系统第一次启动时，需要点击屏幕上的按钮完成教学示范，用户可以按下 F3 进入鼠标模式，然后配合上下左右和OK按键完成教学操作。

10.3 播放4K视频

访问此处下载地址 (<http://download.friendlyelec.com/nanopim1>)的test-video目录，下载4K视频文件4K-Chimei-inn-60mbps.mp4，将其拷贝到SD卡或者U盘上。

在M1(512M RAM)上启动并运行Android系统，将带有视频文件的SD卡或者U盘接到M1上，通过文件浏览器ESFileExplorer找到视频文件，点击视频文件并选择使用系统自带应用Gallery播放视频，即可观看影片。

经测试，将视频文件拷贝到U盘播放效果会更佳。

11 如何编译Android 系统

11.1 准备工作

- 编译全志 H3 的BSP源码包必须使用 64bit 的Linux系统，并安装下列软件包，下列操作均基于Ubuntu-14.04 LTS-64bit：

```
$ sudo apt-get install gawk git gnupg flex bison gperf build-essential \
zip curl libcurl4-openssl-dev libncurses5-dev:i386 x11proto-core-dev \
libxi-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
libgl1-mesa-dev g++-multilib mingw32 tofrodos \
python-markdown libxml2-utils xsltproc zlib1g-dev:i386
```

- 由于Android映像的打包过程依赖lichee源码里的脚本工具，需要先克隆lichee源码：

```
$ git clone https://github.com/friendlyarm/h3_lichee.git lichee
```

注：lichee是全志为其CPU的板级支持包所起的项目名称，里面包含了U-boot，Linux等源码和众多的编译脚本。因为原厂的编译打包脚本依赖lichee这个路径，目录名称必须保持为“lichee”。

- 克隆Android源码：

```
$ git clone https://gitlab.com/friendlyelec/h3_android-4.4 android
```

由于Andoird映像的打包过程依赖lichee源码里的脚本工具，请务必要在lichee目录的同级目录下克隆Android源码，并且目录名称保持为“android”，效果如下：

```
$ ls ./
```

- 下载交叉编译器：

为了编译lichee源码，还需要访问此处下载地址 (<http://download.friendlyelec.com/nanopim1>)的toolchain目录，下载交叉编译器压缩包gcc-linaro-arm.tar.xz，然后将该压缩包放置在lichee/brandy/toolchain/目录下即可，无需解压。

11.2 编译Android

- 搭建编译环境

搭建编译Android的环境建议使用64位的Ubuntu-14.04 LTS-64bit，安装需要的包即可。

```
$ sudo apt-get install bison g++-multilib git gperf libxml2-utils make python-networkx zip flex libncurses5-dev zlib1g-dev gawk minicom
```

更多说明可查看：android_initializing (<https://source.android.com/source/initializing.html>)。

- 安装JDK

使用JDK1.6.0_45版本，下载和安装说明请查看Oracle官方网址：[Oracle JDK](http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html) (<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html>)，这里假设JDK已经成功安装到路径/usr/lib/jvm/下。

- 编译系统

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-m1 -p android -t all      # 编译lichee源码，为Android系统提供内核和驱动模块。
$ cd ../../android
$ export PATH=/usr/lib/jvm/jdk1.6.0_45/bin:$PATH
$ ./build.sh -b nanopi-m1                         # 编译android源码，并且打包生成Android映像文件。
```

编译完成后会在lichee/tools/pack/目录下生成Android系统固件sun8iw7p1_android_nanopi-m1_uart0.img。

11.3 清理lichee源码

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-m1 -p android -t clean
```

12 开发者指南

12.1 CVBS

Linux-3.4 - 配置 CVBS

13 编译内核头文件安装包

以下操作在在开发板上进行：

13.1 本文适用于如下版本的固件

固件文件名: nanopi-XXX_sd_friendlycore-focal_4.14_armhf_YYYYMMDD.img 具体信息:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04 LTS
Release:        20.04
Codename:       focal

$ cat /proc/version
Linux version 4.14.111 (root@ubuntu) (gcc version 4.9.3 (ctng-1.21.0-229g-FA)) #193 SMP Thu Jun 10 18:20:47 CST 2021
```

13.2 安装所需软件包

```
sudo apt-get update
sudo apt-get install dpkg-dev libarchive-tools
```

13.3 制作内核头文件安装包

```
git clone https://github.com/friendlyarm/linux -b sunxi-4.14.y --depth 1 kernel-h3
cd kernel-h3
rm -rf .git
make distclean
touch .scmversion
make CROSS_COMPILE= ARCH=arm sunxi_defconfig
alias tar=bsdtar
make CROSS_COMPILE= ARCH=arm bindeb-pkg -j4
```

显示如下信息表示成功：

```
dpkg-deb: building package 'linux-headers-4.14.111' in '../linux-headers-4.14.111_4.14.111-1_armhf.deb'.
dpkg-deb: building package 'linux-libc-dev' in '../linux-libc-dev_4.14.111-1_armhf.deb'.
dpkg-deb: building package 'linux-image-4.14.111' in '../linux-image-4.14.111_4.14.111-1_armhf.deb'.
dpkg-genchanges: warning: substitution variable ${kernel:debach} used, but is not defined
dpkg-genchanges: info: binary-only upload (no source code included)
```

13.4 安装

```
sudo dpkg -i ../linux-headers-4.14.111_4.14.111-1_armhf.deb
```

13.5 测试

以编译pf_ring模块为例, 参考文档 https://www.ntop.org/guides/pf_ring/get_started/git_installation.html.

```
git clone https://github.com/ntop/PF_RING.git
cd PF_RING/kernel/
make
```

编译完成后, 使用insmod尝试加载模块:

```
sudo insmod ./pf_ring.ko
```

14 更多OS

14.1 DietPi



DietPi

#Lightweight Justice for your SBC

DietPi is a highly optimised & minimal Debian-based Linux distribution. DietPi is extremely lightweight at its core, and also extremely easy to install and use. Setting up a single board computer (SBC) or even a computer, for both regular or server use, takes time and skill. DietPi provides an easy way to install and run favourite software you choose.

For more information, please visit this link <https://dietpi.com/docs/>.

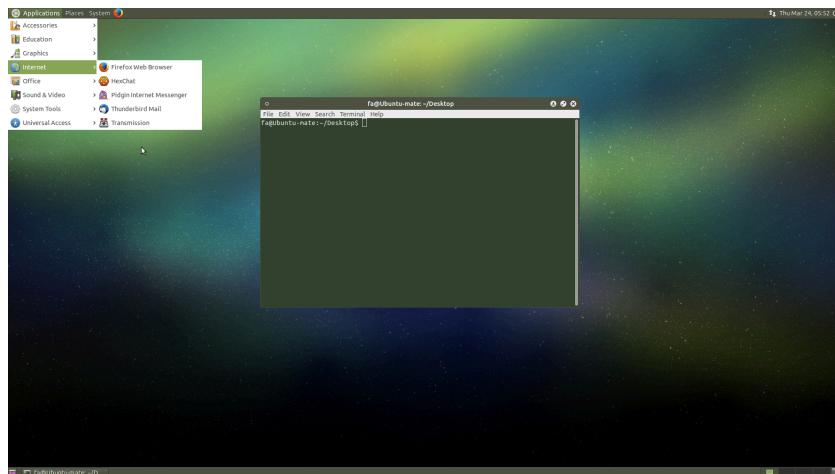
DietPi supports many of the NanoPi board series, you may download the image file from here:

- <https://dietpi.com/docs/hardware/#nanopi-series-friendlyarm>

14.2 Ubuntu-MATE

Ubuntu-MATE基于Ubuntu系统, 使用的桌面环境是MATE-desktop, 界面简洁易用, 需配合HDMI使用, 可通过ssh登录。仅提供给进阶爱好者交流使用, 不对该系统提供专业技术支持。

- 下载系统固件nanopi-m1-ubuntu-mate-sd4g.img.zip(offical-ROMs目录): 点击下载 (<http://wiki.friendlyelec.com/nanopim1/download/>)
- 将文件解压后得到系统固件, 在Windows下使用友善官方提供 win32diskimager 工具烧写Ubuntu-MATE即可。
- 烧写完成后, 将TF卡插入NanoPi M1, 上电即可体验Ubuntu-MATE。
- 登录账号: root或fa ; 登录密码: fa



14.3 Debian8 (Jacer)

Debian8 (Jacer) 系统是网友爱好者“Jacer”基于Debian8系统移植并支持, 使用的桌面环境是Debian8, 此系统对中文支持较好, 界面简洁易用, 需配合HDMI使用, 可通过ssh登录。

由于该系统由第三方爱好者移植提供, 友善官方仅提供给进阶爱好者交流使用, 不对该系统提供专业技术支持。

- 访问下载地址 (<http://download.friendlyelec.com/nanopim1>)下载固件Debian8(unofficial-Jacer).rar(unoffical-ROMs目录)。

- 将文件解压后得到系统固件，在Windows下使用友善官方提供win32diskimager工具烧写即可。
- 烧写完成后，将TF卡插入NanoPi M1，上电即可体验。
- 登录账号：fa；登录密码：fa

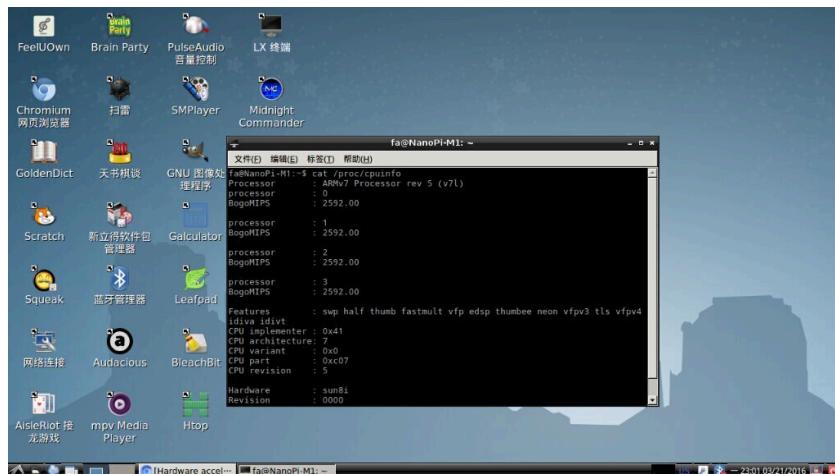
注：Debian8（Jacer）系统接HDMI桌面环境登录时，不建议使用root权限登录，否则界面为黑色，无法显示完整操作界面；

Debian8（Jacer）系统集成了GPU驱动和H264 H265硬解，分辨率默认使用的是720p；如果需要用1080P的分辨率显示，则需要将/boot分区的script.fex里面的HDMI MODE=后面改成相应的数字然后转换为script.bin，修改方法可以参考里面的文件h3disp.sh；

Debian8（Jacer）系统支持的无线网卡型号为：8192cu、8188cus、8188eu、rt3070。

Debian8（Jacer）支持：

- 1.Mali400 GPU驱动
- 2.mpv硬解H264 H265
- 3.最新Chromium浏览器支持flash视频
- 4.支持网易云音乐feuluown
- 5.纸牌 扫雷 象棋游戏
- 6.retroarch游戏模拟器
- 7.为512MB内存增加128mb交换分区虚拟内存
- 8.四核动态调整频率
- 9.aria2下载器
- 10.samba服务
- 11.8192cu/8188cus/8188eu/rt3070/rt2800/rt5370无线网卡支持
- 12.GIMP图形编辑软件
- 13.SSH连接
- 14.xrdp和vnc远程桌面服务
- 15.HTML5多媒体播放
- 16.goldendict词典
- 17.audacious音乐播放器
- 18.pulseaudio音量调整
- 19.USB蓝牙支持



14.4 Android（Jacer）

Android（Jacer）系统是网友爱好者“Jacer”基于Android4.4.2系统移植并支持，使用的桌面环境是Android，需配合HDMI使用，可通过ssh登录。由于该系统由第三方爱好者移植提供，友善官方仅提供给进阶爱好者交流使用，不对该系统提供专业技术支持。

- 下载相关软件及固件

访问下载地址 (<http://download.friendlyelec.com/nanopim1>) 下载固件Beelink_X2_v205k4_for_NanoPiM1(unofficial-ROMs目录)、SD卡格式化工具HDDLLF.4.40和烧写工具HDDLLF(tools目录)。

- 制作启动Android的SD卡

(1) 以管理员权限运行HDDLLF.4.40软件，并且格式化SD卡，格式化后把卡从电脑拔出来；
(2) 再把卡插入电脑，使用Windows自带的格式化程序把SD卡格式化成FAT32格式，格式化后把卡拔出来；
(3) 最后把卡插入电脑，使用全志的烧录软件(PhoenixCard)烧录Android 固件。

- 烧写完成后，将TF卡插入NanoPi M1，上电即可体验。
- Android（Jacer）支持：

(1) 按钮栏可隐藏 加入软关机按钮 频率动态调整；
(2) 含GAAPS；
(3) 支持rtl8188etv/eus 8189无线网卡和CSR蓝牙；

(4) 降低了电压 降低了运行温度;
更多功能, 请自行烧写系统体验。



14.5 Armbian

下载链接和烧写步骤请查看Armbian官方网站M1页面: armbian-m1 (<http://www.armbian.com/nanopi-m1/>)
Armbian官方提供了server和desktop两个版本, desktop运行界面如下:



15 固件下载

- 烧写系统固件: [2] (<http://download.friendlyelec.com/nanopim1>)

16 3D打印文件下载

- NanoPi M1 3D打印外壳: [3] (<http://www.thingiverse.com/thing:1592092>)



17 迷你扩展板M1初学者套件

- Matrix - Compact Kit B:/点击查看 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Compact_Kit_B)

18 NanoPi M1初学者入门开发教程

- 《硬件编程开发教程》点击下载
(http://wiki.friendlyelec.com/wiki/index.php/File:NanoPi_M1%E7%A1%AC%E4%BB%B6%E5%BC%80%E5%8F%91%E6%95%99%E7%A8%8B.pdf)

19 开发者指南

- 定制开发相关
 - Building U-boot and Linux for H5/H3/H2+
 - How to Build FriendlyWrt
 - Qt dev: How to Build, Install and Setting Qt Application
- 系统映像相关
 - How to make your own SD-bootable ROM
 - How to use overlayfs on Linux
 - EFlasher
- 系统配置相关
 - npi-config
 - Use NetworkManager to configure network settings
- 硬件访问相关
 - WiringNP: NanoPi NEO/NEO2/Air GPIO Programming with C
 - RPi.GPIO : NanoPi NEO/NEO2/Air GPIO Programming with Python
 - Hardware Misc (http://wiki.friendlyelec.com/wiki/index.php/Main_Page#Hardware_Misc_.28TBD.29)
 - Matrix (http://wiki.friendlyelec.com/wiki/index.php/Main_Page#Matrix)
 - BakeBit (http://wiki.friendlyelec.com/wiki/index.php/Main_Page#BakeBit)
 - HATs&Docks (http://wiki.friendlyelec.com/wiki/index.php/Main_Page#HATs.26Docks)

20 资源链接

20.1 手册原理图及开发资料

- 原理图
 - NanoPi-M1-V1.0-1804-Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/1/1e/Schematic_NanoPi-M1-V1.1_1804.pdf)
 - NanoPi-M1-1603-Schematic.pdf (<http://wiki.friendlyelec.com/wiki/images/d/d8/NanoPi-M1-1603-Schematic.pdf>)
 - NanoPi-M1-1603B-Schematic.pdf (<http://wiki.friendlyelec.com/wiki/images/6/68/NanoPi-M1-1603B-Schematic.pdf>)
- 尺寸图
 - NanoPi-M1-V1.0-1804-dimensions(dxf).rar (http://wiki.friendlyelec.com/wiki/index.php/File:NanoPi_M1_v1.1_1804_PCB.rar)
 - NanoPi-M1-1603-dimensions(dxf).zip (<http://wiki.friendlyelec.com/wiki/images/a/ad/NanoPi-M1-1603-dimensions%28dxf%29.zip>)
 - NanoPi-M1-1603B-dimensions(dxf).zip (<http://wiki.friendlyelec.com/wiki/images/0/01/NanoPi-M1-1603B-dimensions%28dxf%29.zip>)
- H3芯片手册 Allwinner_H3_Datasheet_V1.2.pdf (http://wiki.friendlyelec.com/wiki/images/4/4b/Allwinner_H3_Datasheet_V1.2.pdf)

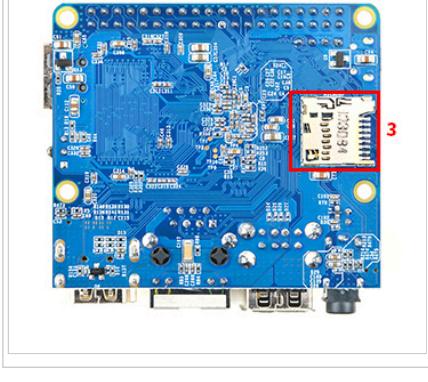
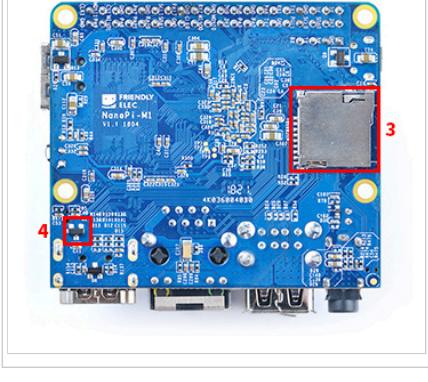
20.2 开发教程及文档

- 模块介绍以及开发文档：
 - 按键模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Button/zh)
 - LED模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_LED/zh)
 - 模数转换 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Analog_to_Digital_Converter/zh)
 - 继电器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Relay/zh)
 - 三轴重力加速度模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_3-Axis_Digital_Accelerometer/zh)
 - 三轴数字指南针模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_3-Axis_Digital_Compass/zh)
 - 温度传感器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Temperature_Sensor/zh)
 - 温湿度传感器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Temperature_and_Humidity_Sensor/zh)
 - 蜂鸣器 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Buzzer/zh)
 - 摆杆模块(Joystick) (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Joystick/zh)
 - I2C(PCF8574)+LCD1602 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_I2C_LCD1602_Keypad/zh)
 - 声音传感器 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Sound_Sensor/zh)
 - 超声波模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Ultrasonic_Ranger/zh)
 - GPS模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_GPS/zh)
 - 迷你扩展板Matrix - Compact Kit (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Compact_Kit/zh)
 - 火焰传感器 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Fire_Sensor)
 - CAM500 500万像素摄像头 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_CAM500A/zh)
 - 滚珠开关模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Ball_Rolling_Switch/zh)
 - 2'8 SPI Key TFT 2.8寸spi液晶屏 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_2%278_SPI_Key_TFT/zh)
 - 红外计数模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_IR_Counter/zh)
 - 红外接收模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_IR_Receiver/zh)
 - 电机驱动器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_L298N_Motor_Driver/zh)

- MQ-2 烟雾传感器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_MQ-2_Gas_Sensor/zh)
- MQ-3 气体传感器 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_MQ-3_Gas_Sensor/zh)
- 单点电容式数字触摸传感器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_One_Touch_Sensor/zh)
- 光敏电阻模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Photoresistor/zh)
- 电位器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Potentiometer/zh)
- 压力传感器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Pressure_and_Temperature_Sensor)
- RGB LED (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_RGB_LED/zh)
- RTC模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_RTC/zh)
- Rotary Encoder (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Rotary_Encoder/zh)
- 土壤湿度检测传感器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Soil_Moisture_Sensor/zh)
- 热敏电阻模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Thermistor/zh)
- USB WiFi (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_USB_WiFi/zh)
- 水位/水滴识别检测传感器模块 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Water_Sensor)

21 硬件更新

▪ NanoPi M1 Version Compare & List (Hardware)

version	NanoPi M1 1603B	NanoPi M1 V1.1 1804
Photo	 	 
电源部分	① VDD-CPUX使用SY8113B电源芯片	① VDD-CPUX使更换为MP2143DJ
Audio接口		② NanoPi M1 V1.1 1804版相对1603B版增加了C14电容，改善了录音质量
TF卡座		③ NanoPi M1 V1.1 1804版相对1603B版更换了TF卡座
复位电路		④ NanoPi M1 V1.1 1804版相对1603B版增加了复位IC

22 更新日志

注意: 本章节的描述针对所有的H3/H2+板子, 部分硬件相关的功能描述仅支持特定的板子, 请优先阅读开发板对应的维基以确定是否有相关的硬件功能。

2023-11-07

h3 FriendlyCore:

- 升级到 Ubuntu Core 22.04;

h3 Debian Core:

- 增加 Debian bookworm core;

2021-06-25

h3 FriendlyCore:

- 升级到 Ubuntu Core 20.04;

2021-04-25

h3 FriendlyCore：

- 修复 Linux 里 spi 驱动相关的 bug;
- 修复 WiringNP 无法操作 GPIOA0 的 bug;

2021-02-24

h3 FriendlyCore：

- 修复 i2s 左右声道异常交换的 bug;

2019-12-19

- 修复dma驱动里的bug；

- 修复播放音频前后有爆破声的问题；

- uboot添加dtb overlay的功能；

2019-11-19

- 修复 H3 OLED-ROM 无显示的问题；

2019-08-26

h3 OpenWrt：

- 将OpenWrt改名为FriendlyWrt；
- 修改WiFi热点名称为FriendlyWrt，并设置连接密码为password；
- 支持5g usb wifi RTL8821CU；
- 支持华为随行WiFi 2 mini(E8372H-155)；

2019-08-23

h3 FriendlyCore：

- 支持5g usb wifi RTL8821CU/RTL8812AU；

- 支持docker；

- 支持4G 模块ec20；

h3 eflasher：

- 精简rootfs，提升启动速度；

- 支持LED显示烧写状态：快闪表示正在烧写中，慢闪表示没有在烧写；

- windows系统下可以查看FriendlyARM分区，便于拷贝烧写image-for-eflasher文件；

2019-05-22

OpenWrt系统支持lcd2usb模块，开机可自动显示ip地址；

调整CPU DVFS，提升Linux-4.14 系统稳定性；

2019-04-25

内核从4.14.52升级到4.14.111；

Linux-4.14 的ROM启用overlayfs，降低异常关机rootfs损坏的概率；

所有H3/H2+的板子都添加OpenWrt的ROM；

使能Linux-4.14 R8188EU、R8712U、SCSI相关配置项；

提升Linux-4.14 WiFi AP模式的稳定性；

提升Linux-4.14 系统稳定性；

2019-03-19

NanoPi M1-Plus/NEO-Air支持OpenWrt；

2019-03-05

NanoPi M1-Plus/M1/NEO/NEO-Air/NEO-Core的Linux-4.14 FriendlyCore系统支持USB WiFi rtl8188eu，首次上传H3 FriendlyCore rootfs压缩包；

2019-01-21

NanoPi Duo2的Linux-4.14 FriendlyCore系统添加测试SIM800C模块的Python demo程序；

NanoPi Duo2发布Linux-4.14 FriendlyCore系统的rootfs压缩包；

2019-01-15

修复NanoPi Duo2 FriendlyCore系统第一次启动时蓝牙无法使用的bug；

NanoPi Duo2的Linux-4.14 FriendlyCore系统支持USB WiFi rtl8188eu；

2018-12-03

首次为NanoPi Duo2适配Linux-3.4内核；

2018-11-29

修复M1 Plus Android源码编译出来的Android镜像无法打开WiFi的Bug；

2018-11-20

提升系统的稳定性；

2018-10-24

H3开发板的Linux-3.4内核支持更多型号的eMMC；

2018-10-11

Linux-4.14 ov5640摄像头支持自动对焦；

规范images-for-eflasher的命名；

2018-09-19

规范系统固件的名称；
eflasher支持通过命令行方式备份和还原eMMC；

2018-07-30

NanoPi M1 / NanoPi M1 Plus的Android系统支持调整HDMI分辨率和HDMI画面大小；
添加TF卡脱机烧写Android系统到eMMC的说明章节；

2018-07-11

光盘里添加非官方开发和维护的ROM: DietPi；

2018-06-28

mjpg-streamer支持在指定的framebuffer上显示摄像头数据；
升级Linux4.14.0到Linux4.14.52；
Linux4.14.52内核支持I2S0 + PCM5102A；

2018-06-25

U-boot会打印启动设备的名称，例如: Boot device: emmc；
Linux-4.14支持摄像头Cam500B；
提高eflasher系统的稳定性，并且支持备份和还原emmc；

2018-05-11

Linux内核回退到LTS版本: 4.14；
4.14内核使能看门狗；
4.14内核使能红外接收；
4.14内核使能MicroUSB的OTG功能；
U-boot不再打印CPU ID, 4.14内核提供读CPU ID的接口(/sys/bus/nvmmem/devices/sunxi-sid0/nvmmem)；

2018-04-10

■ H3/H2+系列开发板

升级Linux到4.16.0版本Mainline U-boot and Linux；
支持DVP接口的摄像头Matrix - CAM500B；
支持I2S+NanoHat PCM5102A；
支持HDMI音频输出；
使能看门狗；
使能红外接收；
使能MicroUSB的OTG功能；
发布默认使能Matrix - 2'8 SPI Key TFT的ROM；
完善Npi-config，支持选择显示/声卡设备；
维基添加IR/watchdog说明章节；

2018-01-24

■ ROM(Linux-4.x内核):

WiFi-ap6212支持切换为ap模式；

2017-12-20

■ Android系统(Linux-3.x内核):

支持WiFi softap 模式；

2017-12-19

■ OLED-ROM修复sources.list异常的问题；

2017-12-13

■ Debian和FriendlyCore系统(Linux-4.x内核):

设置Micro USB的功能为Host,OTG驱动有bug；

2017-12-08

■ Debian和FriendlyCore系统(Linux-4.x内核):

升级Uboot到2017.11版本；
升级Linux内核到4.14版本；
Linux内核使能PPP相关的配置项；
支持HDMI音频输出；
修复以太网LED显示异常的问题；

2017-07-05

■ Debian和FriendlyCore系统:

使用NetworkManager作为网络管理工具；
优化内存使用策略，提升系统稳定性；

- Linux-4.11:

支持I2S0和NanoHat PCM5102A；
支持Matrix-2'8_SPI_Key_TFT；

2017-06-08

- 添加Linux-4.x和Linux-3.x系统固件差异的说明
- 添加FriendlyCore系统的使用说明
- 简化Linux-3.x的BSP的编译操作；

2017-05-31

- Debian和FriendlyCore系统(Linux-3.x内核):

修复DVP摄像头cam500B无法使用ffmpeg录制视频的问题；

2017-05-25

- 发布NAS-ROM；

2017-05-23

- Android系统(Linux-3.x内核):

修复千兆以太网/WiFi/BT无法使用的问题；

2017-05-19

- Debian和FriendlyCore系统(Linux-4.x内核):

支持通过npi-config使能/禁止i2c/spi/serial/pwm;
支持通过WiringPi来控制gpio引脚；
支持通过扩展3.5mm耳机孔录制和播放音频；
支持市面上大多数USB以太网卡/USB WiFi网卡；
支持串口打印内核启动信息；
支持软件生成唯一MAC地址功能；
支持使用Bakebit套件；
修复系统启动时欢迎界面温度显示异常的问题；

- 发布oled-ROM；

2017-04-18

- FriendlyCore系统:

修改了登录欢迎界面，登录时会打印系统的基本状态信息；
增加 npi-config 工具；
采用NetworkManager作为网络管理工具；
新增pi用户，配置为自动登录；

2017-03-01

- 发布基于U-boot-2017.x和Linux-4.x.y源码的ROM，和基于Linux-3.x的ROM共用Debianx和FriendlyCore文件系统；

2017-02-28

- Debian和FriendlyCore系统(Linux-3.x内核):

支持以太网使用唯一MAC地址的功能；

2017-02-27

- Debian和FriendlyCore系统(Linux-3.x内核):

ROM支持WiFi芯片AP6212A；

2017-02-20

- FriendlyCore系统(Linux-3.x内核):

添加nano编辑器；
解决“unable to resolve host FriendlyARM”的问题；
将fa用户添加到sudoers中；

2017-01-22

- FriendlyCore系统(Linux-3.x内核):

版本号从15.10升级到16.04；

- eflasher系统

支持命令行烧写系统到eMMC；

2017-01-20

- Linux-3.x-BSP源码:

将H3 BSP代码分为lichee和android两部分，并精简lichee目录；

更新H3 BSP里的交叉编译器，解决该编译器无法编译应用程序的问题；

完善OV5640驱动和视频采集程序Mjpg-streamer，支持更多的视频采集格式；

支持fastboot更新U-boot；

- Debian系统:

支持通过rpi-monitor检测系统状态；

- Debian和Ubuntu-core系统

支持第一次开机自动扩展文件系统，并且修复文件系统；

2016-12-13

- FriendlyCore系统(Linux-3.x内核):

支持通过rpi-monitor检测系统状态；

支持声卡配件NanoHat-PCM5102A；

2016-12-08

- Linux-3.x-BSP源码:

修复Android系统源码编译失败的问题；

提供一个快速编译Android系统的脚本；

- Android系统(Linux-3.x内核):

添加了系统应用Gallery，可用于播放视频(含4K)和浏览图片；

添加了应用ESFileExplorer，可用于浏览文件；

2016-09-07

- Debian和FriendlyCore系统(Linux-3.x内核):

支持摄像头模块CAM500A，集成mjpg-streamer和ffmpeg以便于测试；

- Debian和FriendlyCore系统(Linux-3.x内核):

支持显示bootlogo；

2016-08-04

- FriendlyCore系统(Linux-3.x内核):

支持市面上常见的USB WiFi模块；

2016-07-28

- Android系统(Linux-3.x内核):

增加以太网的设置界面，可配置使用StaticIP或DHCP方式连接有线网络；

2016-07-04

- FriendlyCore系统(Linux-3.x内核):

降低发热量；

- Debian系统(Linux-3.x内核):

增加不输出hdmi的script.bin，降低发热量，支持麦穗配件；

2016-06-28

- 发布FriendlyCore系统，该系统是基于UbuntuCore定制而来的；

2016-05-05

- 支持Armbian；

- 支持OpenWRT；

- 修复Linux-3.x内核容易被root的问题；

2016-04-25

- 更新资源特性, "DDR3 RAM: 512MB" 改为 "DDR3 RAM: 512MB/1GB"；
- 更新机械尺寸为最新版本(1603B)；
- 添加1603B的dxf文件和原理图到资源链接；

2016-04-20

- Android系统(Linux-3.x内核):

支持红外遥控器RC-100；
支持USB WiFi，型号包括rtl8188etv/rtl8188eu；
修复第2个USB HOST口无法使用的问题；

- Debian和Android系统(Linux-3.x内核):

支持UART1；

2016-04-13

- Debian系统(Linux-3.x内核):

默认分辨率修改为720P-60Hz；
添加512MB swap虚拟内存；
支持sys子系统操作GPIO；
支持市面上常见的USB WiFi模块；
支持市面上常见的USB转串口模块；
增加实用小工具fs_resize\video-play\cpu-freq；

- Android系统(Linux-3.x内核):

增加开机蓝色LED闪烁功能；

- Debian和Android系统(Linux-3.x内核):

支持2级动态电压调节，优化高负载时的功耗；

- 发布Android源代码和更新lichee源码；

2016-04-07

- 首次发布H3板子的维基；

Retrieved from "https://wiki.friendlyelec.com/wiki/index.php?title=NanoPi_M1/zh&oldid=25326"

-
- This page was last modified on 14 November 2023, at 08:38.