# NanoPi NEO2

看中文

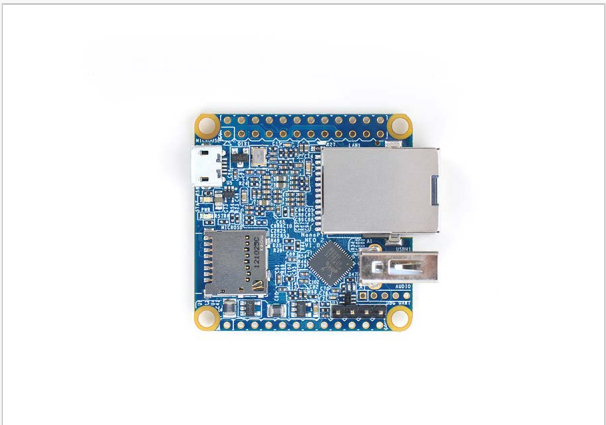
**Contents** [hide]

# Introduction

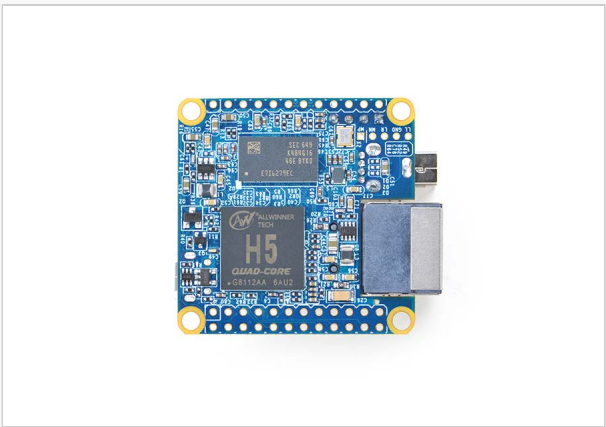
- The NanoPI NEO2 is a newly released super tiny ARM board by FriendlyElec. It uses Allwinner's 64-bit H5 quad-core SoC (ARM Cortex-A53). It has internal hexa-core Mail450 GPU, 512M DDR3 RAM. A UbuntuCore and Armbian image files are ready for it.
- The NanoPi NEO2 inherits NEO's form factor and has compatible interfaces and ports with NEO. In addition in such a small dimension it has Gbps Ethernet and one USB host port. These features make it especially suitable for applications that require high data throughput , speedy data transmission and high performance. Hobbyists and makers will just love it.

Overview

# Hardware Spec



Front



Back

- CPU: Allwinner H5, Quad-core 64-bit high-performance Cortex A53
- DDR3 RAM: 512MB
- Connectivity: 10/100/1000M Ethernet, RTL8211E-VB-CG chip
- USB Host: USB Type A x 1 and USB pin header x 2
- MicroSD Slot: MicroSD x 1 for system boot and storage
- LED: Power LED x 1, System LED(Blue) x 1
- GPIO1: 2.54mm pitch 24 pin-header, compatible with Raspberry Pi's GPIO pin1 - pin 24. It includes UART, SPI, I2C, IO etc
- GPIO2: 2.54mm pitch 12 pin-header. It includes USB, IR receiver, I2S, IO etc
- Serial Debug Port: 2.54mm pitch 4pin-header
- Audio In/Out: 2.0mm pitch 5 pin-header
- MicroUSB: Power input(5V/2A) and OTG
- PCB Dimension: 40 x 40mm
- Working Temperature: -30℃ to 70℃
- Weight: 13g(WITHOUT Pin-headers)
- OS/Software: u-boot,Ubuntu Core

## Software Features

### uboot

- mainline uboot released on May 2017

### UbuntuCore 16.04

- 64-bit system
- mainline kernel: Linux-4.11.2
- rpi-monitor: check system status and information
- npi-config: system configuration utility for setting passwords, language, timezone, hostname, SSH and auto-login,and enabling/disabling i2c, spi, serial and PWM. When enabling PWM it will prompt

that Serial debug port will be disabled.

- software utility: wiringNP to access GPIO pins
- software utility: RPi.GPIO_NP to access GPIO pins
- networkmanager: manage network
- system log output from serial port
- supports USB WiFi module: refer to #Connect USB WiFi to NEO
- supports audio recording and playing with 3.5mm audio jack
- supports I2C 0/1
- fixed MAC address

### Ubuntu OLED

- mainline kernel: Linux-4.11.2
- supports FriendlyElec's OLED module

### Debian

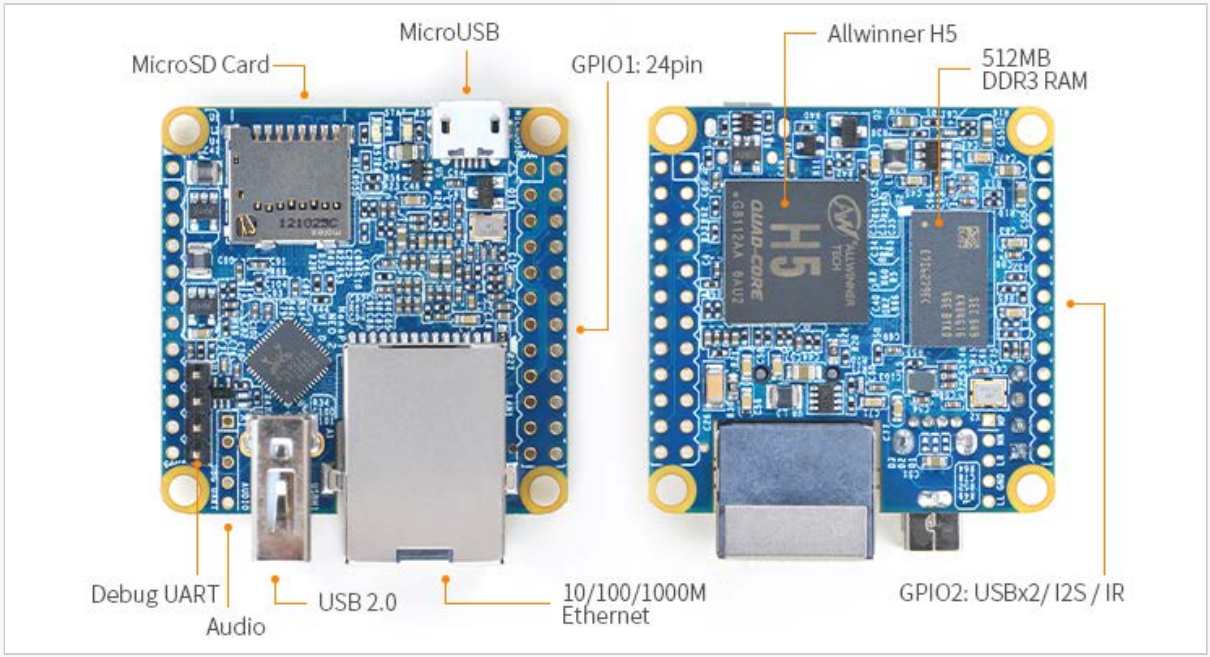
- welcome window with basic system information and status

### Debian for NAS Dock

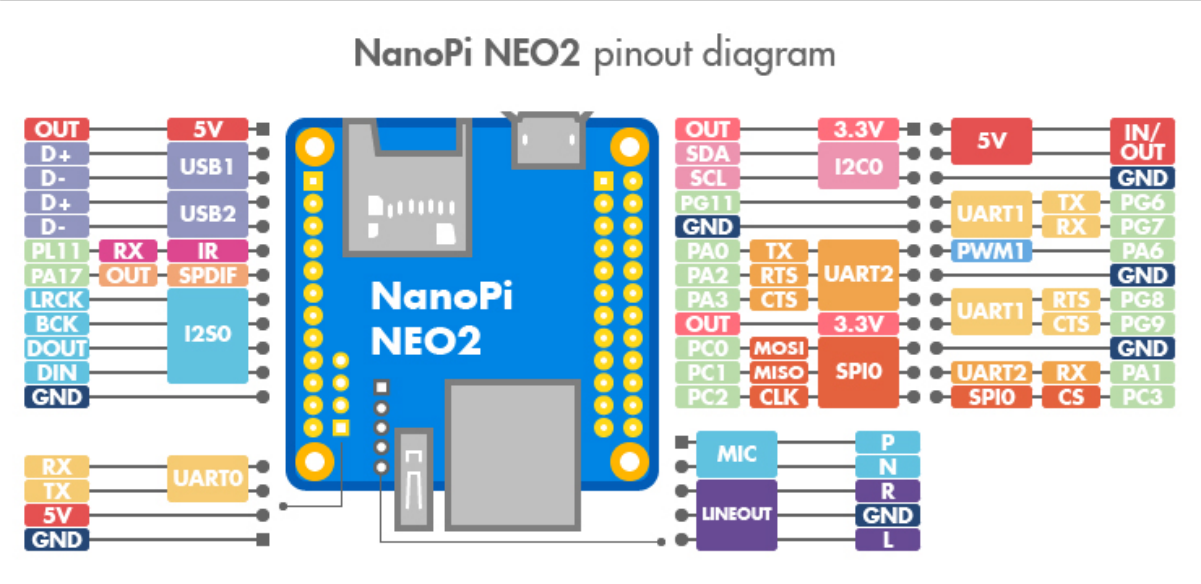
- mainline kernel: Linux-4.11.2
- supports FriendlyElec's NAS Dock
- optimized OpenMediaVault configuration options
- allocated swap section

## Diagram, Layout and Dimension

### Layout




NanoPi NEO2 Layout




pinout

- **GPIO Pin Description**

| Pin# | Name | Linux gpio | Pin# | Name | Linux gpio |
|---|---|---|---|---|---|
| 1 | SYS_3.3V | | 2 | VDD_5V | |
| 3 | I2C0_SDA/GPIOA12 | 12 | 4 | VDD_5V | |
| 5 | I2C0_SCL/GPIOA11 | 11 | 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 |

- **USB/Audio/IR Pin Descripton**

| **NanoPi-NEO2** | | |
|---|---|---|
| Pin# | Name | Description |
| 1 | VDD_5V | 5V Power Out |
| 2 | USB-DP1 | USB1 DP Signal |
| 3 | USB-DM1 | USB1 DM Signal |
| 4 | USB-DP2 | USB2 DP Signal |
| 5 | USB-DM2 | USB2 DM Signal |

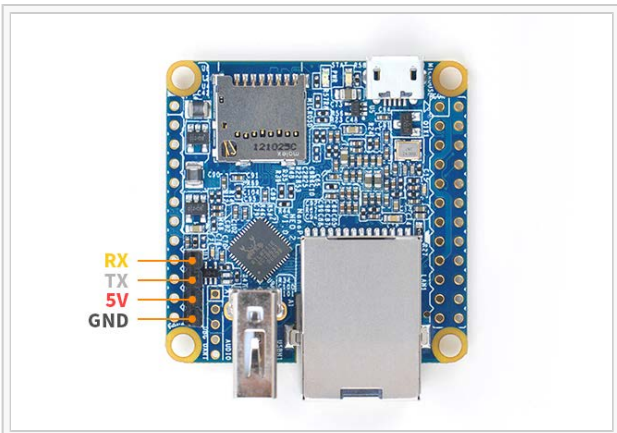
| | | |
|---|---|---|
| 6 | GPIOL11/IR-RX | GPIOL11 or IR Receive |
| 7 | SPDIF-OUT/GPIOA17 | GPIOA17 or SPDIF-OUT |
| 8 | PCM0_SYNC/I2S0_LRC | I2S/PCM Sample Rate Clock/Sync |
| 9 | PCM0_CLK/I2S0_BCK | I2S/PCM Sample Rate Clock |
| 10 | PCM0_DOUT/I2S0_SDOUT | I2S/PCM Serial Data Output |
| 11 | PCM0_DIN/I2S0_SDIN | I2S/PCM Serial Data Input |
| 12 | GND | 0V |

- **Audio**

| Pin# | Name | Description |
|---|---|---|
| 1 | MICIN1P | Microphone Positive Input |
| 2 | MICIN1N | Microphone Negative Input |
| 3 | LINEOUTR | LINE-OUT Right Channel Output |
| 4 | GND | 0V |
| 5 | LINEOUTL | LINE-OUT Left Channel Output |

- **Debug Port（UART0）**

| Pin# | Name |
|---|---|
| 1 | GND |
| 2 | VDD_5V |
| 3 | UART_TXD0 |
| 4 | UART_RXD0 |

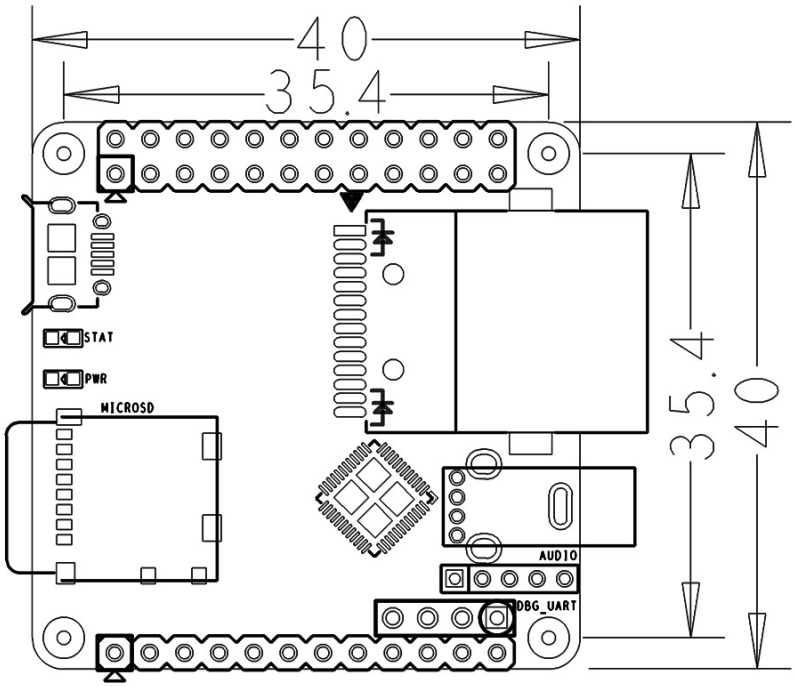


DBG_UART

**Note**

1. SYS_3.3V: 3.3V power output
2. VDD_5V: 5V power input/output. The input range is 4.7V ~ 5.6V. It can take power input from the MicroUSB.
3. All pins are 3.3V and output current is 5mA
4. For more details refer to the document: NanoPi_NEO2-1701-Schematic.pdf

## Dimensional Diagram




For more details refer to pcb file in dxf format

# Get Started

## Essentials You Need

Before starting to use your NanoPi NEO2 get the following items ready

- NanoPi NEO2
- microSD Card/TFCard: Class 10 or Above, minimum 8GB SDHC
- microUSB power. A 5V/2A power is a must
- A Host computer running Ubuntu 14.04 64 bit system

## TF Cards We Tested

To make your NanoPi NEO2 boot and run fast we highly recommend you use a Class10 8GB SDHC TF card or a better one. The following cards are what we used in all our test cases presented here:

- SanDisk TF 8G Class10 Micro/SD TF card:


- SanDisk TF128G MicroSDXC TF 128G Class10 48MB/S:

川宇 8G C10 High Speed class10 micro SD card:


## Make an Installation TF Card

### Get Image File

Get the following files from download link to download image files (under the officail-ROMs directory) and the flashing utility(under the tools directory):

| Image Files: | |
|---|---|
| nanopi-neo2_ubuntu-core-xenial_3.10.y_YYYYMMDD.img.zip | Ubuntu-Core with Qt-Embedded Image File, kernel:Linux-3.10 |
| nanopi-neo2_ubuntu-core-xenial_4.x.y_YYYYMMDD.img.zip | Ubuntu-Core with Qt-Embedded Image File, kernel:Linux-4.x |
| nanopi-neo2_debian-nas-jessie_4.x.y_YYYYMMDD.img.zip | Image File with Support for NAS Dock, Kernel: Linux-4.x, applicable to 1-bay NAS Dock |
| nanopi-neo2_ubuntu-oled_4.x.y_YYYYMMDD.img.zip | Image File with Support for OLED Module, Kernel: Linux-4.x, applicable to NanoHat OLED |
| Flash Utility: | |
| win32diskimager.rar | Windows utility. Under Linux users can use "dd" |

### Make Ubuntu-Core with Qt-Embedded Image Card

- Extract the UbuntuCore file. Insert a TF card(at least 8G) into a Windows PC and run the win32diskimager utility as administrator. On the utility's main window select your TF card's drive, the wanted image file and click on "write" to start flashing the TF card.
- After this writing process is done insert this card into your NanoPi NEO2's TF card slot and power on (with a 5V/2A power source). If the blue LED is blinking this indicates your NanoPi NEO2 has successfully booted.

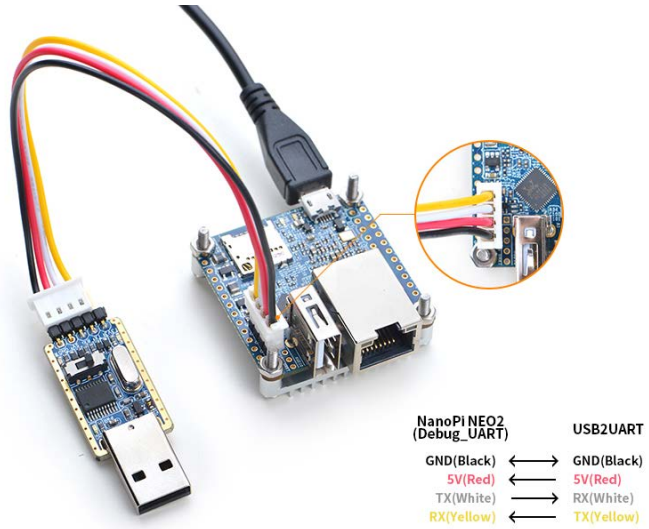
Note:you can make both a Debian and a Ubuntu image card in this way.

# Work with Ubuntu-Core with Qt-Embedded

## Run Ubuntu-Core with Qt-Embedded

- If you want to do kernel development you need to use a serial communication board, ie a PSU-ONECOM board, which will allow you to operate the board via a serial terminal. Here is a setup where we connect a NanoPi NEO2 to a PC via the PSU-ONECOM and you can power on your NEO2 from either the PSU-ONECOM or the board's MicroUSB:

Or you can use a USB to Serial module to do it. Here is a hardware setup:




- Ubuntu-Core's User Accounts:

Non-root User:

```
User Name: pi
Password: pi
```

Root:

```
User Name: root
Password: fa
```

The system is automatically logged in as "pi". You can do "sudo npi-config" to disable auto login.

- Update packages:

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

### Extend TF Card's rootfs Section

When you boot Debian/UbuntuCore for the first time with your image card your OS will automatically resize the file system and this process takes a relatively long time.After your OS is fully loaded you can check the file system's size by using the following command:

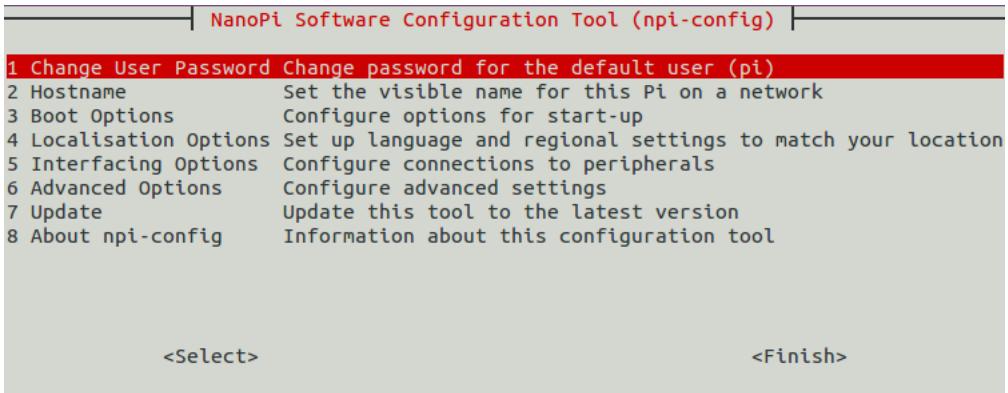
```
$ df -h
```

### Configure System with npi-config

The npi-config is a system configuration utility for setting passwords, language, timezone, hostname, SSH and auto-login,and enabling/disabling i2c, spi, serial and PWM. Type the following command to run this utility.

```
$ sudo npi-config
```

Here is how npi-config's GUI looks like:




### Ethernet Connection

If a NanoPi NEO2 is connected to a network via Ethernet before it is powered on it will automatically obtain an IP after it is powered up. If it is not connected via Ethernet or its DHCP is not activated obtaining an IP will fail and system will hang on for about 15 to 60 seconds.In this case you can try obtaining an IP by using the following command:

```
$ dhclient eth0
```

### Connect USB WiFi to NEO2

Our system has support for popular USB WiFi drivers. Many USB WiFi modules are plug and play with our system. Here is a list of models we tested;

| Number | Model |
|---|---|
| 1 | RTL8188CUS 802.11n WLAN Adapter |
| 2 | RT2070 Wireless Adapter |
| 3 | RT2870/RT3070 Wireless Adapter |
| 4 | RTL8192CU Wireless Adapter |

| | |
|---|---|
| 5 | NetGear, Inc. WG111v3 54 Mbps Wireless [realtek RTL8187B] |

- List network devices

```
$ sudo nmcli dev
```

Note: if a network device's status is "unmanaged" it means that device is not accessed by the NetworkManager and you need to clear the settings in " /etc/network/interfaces" and reboot your board.

- Start WiFi

```
$ sudo nmcli r wifi on
```

- Scan Nearby WiFi Hotspots

```
$ sudo nmcli dev wifi
```

- Connect to a WiFi Hotspot

```
$ sudo nmcli dev wifi connect "SSID" password "PASSWORD"
```

The "SSID" and "Password" need to be replaced with your actual SSID and password.
If a connection is successfully established your board will be automatically connected to your specified WiFi on system reboot.

For more details about the NetworkManager refer to this article:NetworkManager

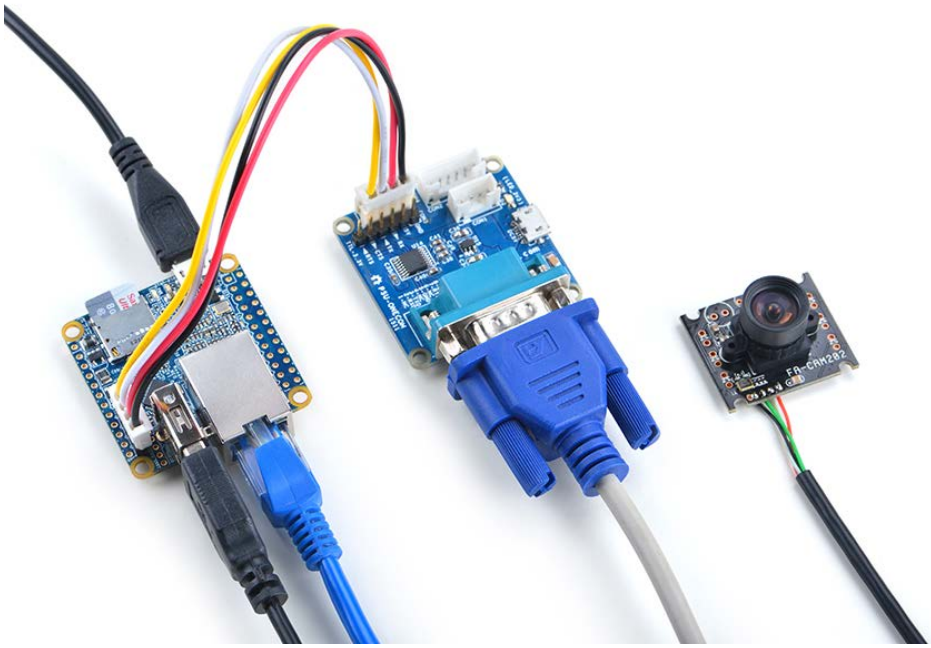
## Login via SSH

The NanoPi NEO2 doesn't have a video output interface. You can log into the board via SSH. In our test the IP address detected by our router was 192.168.1.230 and we ran the following command to log into the NanoPi NEO2:

```
$ ssh root@192.168.1.230
```

The password is fa

## Connect NanoPi NEO2 to USB Camera(FA-CAM202)



The FA-CAM202 is a 2M-pixel USB camera module.Here is its wiki page Matrix - USB_Camera(FA-CAM202)

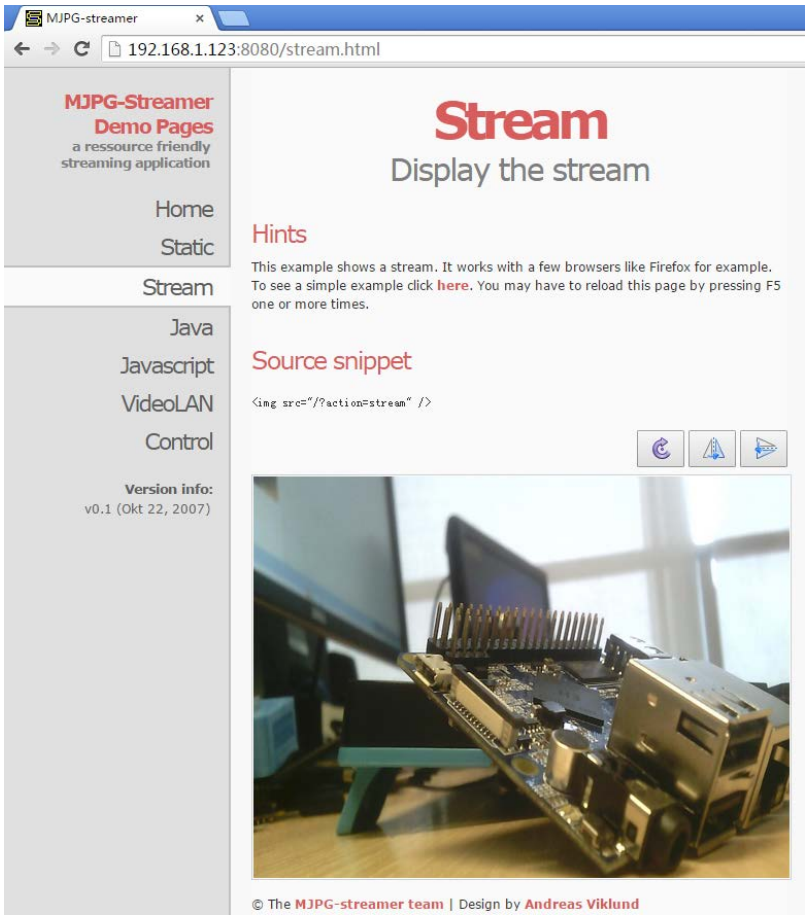
Boot your NEO2, connect NEO2 to the internet, log in the system as root, compile and run the mjpg-streamer utility:

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

The mjpg-streamer is an open source media server. After it is started successfully you will see the following messages:

```
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
```

In our case our NEO2's IP address was 192.168.1.123. We typed "192.168.1.123:8080" on a browser, entered and we got the following screenshot:

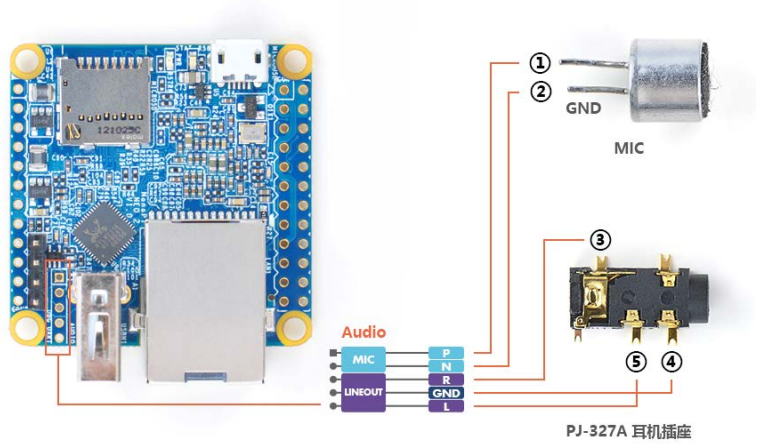



## Play & Record Audio

The NanoPi NEO2 has an audio interface (2.0mm pitch 5-pin header) whose pin description is as follows:

| Pin# | Name | Description |
|---|---|---|
| 1 | MICIN1P | Microphone Positive Input |
| 2 | MICIN1N | Microphone Negative Input |
| 3 | LINEOUTR | LINE-OUT Right Channel Output |
| 4 | GND | Ground |
| 5 | LINEOUTL | LINE-OUT Left Channel Output |

Here is a hardware setup on how to connect an audio device to a NEO2:




Before begin to play or record a audio make sure your NEO2 is connected to an audio device.
Check a recognized audio device:

```
$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: Codec [H3 Audio Codec], device 0: CDC PCM Codec-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Both Allwinner's H5 and H3 have an internal codec which is named as [H3 Audio Codec] in mainline kernels.

Play an audio file:

```
$ aplay /root/Music/test.wav -D plughw:0
```

Record an audio file:

```
$ arecord -f cd -d 5 test.wav
```

### Access GPIO Pins/Wirings with WiringNP

The wiringPi library was initially developed by Gordon Henderson in C. It contains libraries to access GPIO, I2C, SPI, UART, PWM and etc. The wiringPi library contains various libraries, header files and a commandline utility:gpio. The gpio utility can be used to read and write GPIO pins.
FriendlyElec integrated this utility in its UbuntuCore system allowing users to easily access GPIO pins.
For more details refer to [1]

### Access GPIO Pins/Wirings with RPi.GPIO_NP

RPi.GPIO is a famous library in python for Raspberry Pi. FriendlyElec ported it to the NanoPi NEO/NEO2's UbuntuCore images and renamed it as RPi.GPIO_NP.
For more details refer to [2]

## Make Your Own Ubuntu-Core with Qt-Embedded

### Mainline U-boot & Linux(64 bit)

Now the NanoPi NEO2 can run a 64-bit Linux kernel with 64-bit Ubuntu Core 16.04. Here is a detailed reference on how to run mainline U-boot and Linux on H5: Mainline U-boot & Linux

## Use Allwinner's BSP

### Preparations

Visit this link download link and enter the "sources/nanopi-H5-bsp" directory and download all the source code.Use the 7-zip utility to extract it and a lichee directory and an Android directory will be generated.You can check that by running the following command:

```
$ ls ./
$ lichee
```

Or you can get it from our github:

```
$ git clone https://github.com/friendlyarm/h5_lichee.git lichee
```

Note: "lichee" is the project name named by Allwinner for its CPU's source code which contains the source code of U-boot, Linux kernel and various scripts.

### Install Cross Compiler

Visit this site download link, enter the "toolchain" directory, download the cross compiler "gcc-linaro-arm-4.6.3.tar.xz" and "gcc-linaro-aarch64.tar.xz" and copy them to the "lichee/brandy/toochain/" directory.
"gcc-linaro-arm-4.6.3.tar.xz"is for compiling u-boot and "gcc-linaro-aarch64.tar.xz" is for compiling Linux kernel.

### Compile lichee Source Code

Compilation of the H5's BSP source code must be done under a PC running a 64-bit Linux.The following cases were tested on Ubuntu-14.04 LTS-64bit:

```
$ sudo apt-get install gawk git gnupg flex bison gperf build-
essential \
zip curl libc6-dev libncurses5-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 xsltproc zlib1g-dev:i386
```

Enter the lichee directory and run the following command to compile the whole package:

```
$ cd lichee/fa_tools
$ ./build.sh -b nanopi-neo2 -p linux -t all
```

After this compilation succeeds a u-boot, Linux kernel and kernel modules will be generated.
Note: the lichee directory contains cross-compilers we have setup. When the build.sh script runs it will automatically call these cross-compilers.

The following commands can be used to update the u-boot on an installation TF card:

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

Note: you need to replace "/dev/sdx" with the device name in your system.
The boot.img and kernel modules are under the "linux-3.10/output" directory. You can copy the new boot.img file to your TF card's boot section.

#### Compile U-boot

Note:you need to compile the whole lichee directory before you can compile U-boot individually.
You can run the following commands to compile u-boot individually:

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-neo2 -p linux -t u-boot
```

#### Compile Linux Kernel

Note:you need to compile the whole lichee directory before you can compile Linux kernel individually.
You can run the following commands to compile Linux kernel individually:

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-neo2 -p linux -t kernel
```

The boot.img and kernel modules are under the "linux-3.10/output" directory. You can copy the new boot.img file to your TF card's boot section.

#### Clean Source Code

```
$ cd lichee/fa_tools/
$ ./build.sh -b nanopi-neo2 -p linux -t clean
```

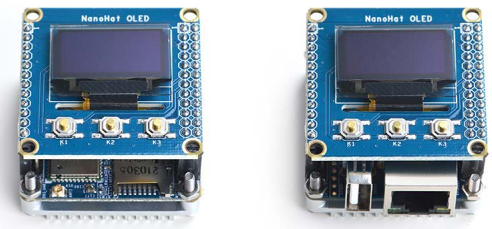
## Connect External Modules to NEO2

### DIY NAS Server with 1-bay NAS Dock & NEO2

The 1-bay NAS Dock is an expansion board which can be used to connect an external hard disk to a NanoPi NEO2.It uses JSM568 USB3.0 to SATA IC and communicates with a NanoPi NEO2 via USB interface. It works with a 2.5" SATA hard disk.It uses TI's DC-DC chipset to convert a 12V input to 5V. It has a power switch for users to turn on/off the device.It supports an onboard RTC battery. FriendlyElec migrated mainline Linux-4.11 kernel and Debian-Jessie with OpenMediaVault. Together with FriendlyElec's customized aluminum case you can quickly assemble a storage server. Here is a hardware setup :1-bay NAS Dock v1.2 for NanoPi NEO/NEO2

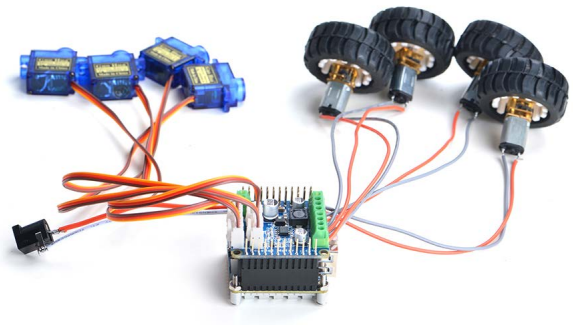
## Connect Python Programmable NanoHat OLED to NEO2

The NanoHat OLED module is a small and cute monochrome OLED module with low power consumption. It has three user buttons. We provide its driver's source code and a user friendly shell interface on which you can check system information and status.A customized aluminum case is made for it. You cannot miss this lovely utility! Here is a hardware setup:NanoHat OLED



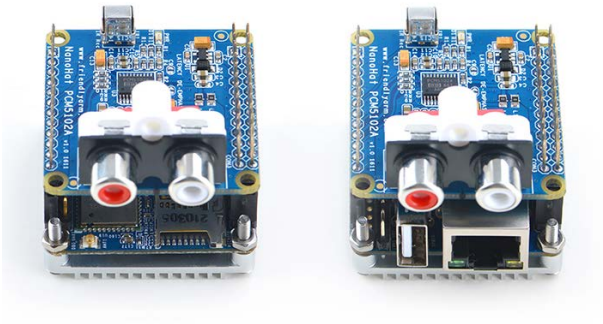

## Connect Python Programmable NanoHat Motor to NEO2

The NanoHat Motor module can drive four 5V PWM steering motors and four 12V DC motors or four 5V PWM steering motors and two 12V four-wire step motors.Here is a hardware setup: NanoHat Motor



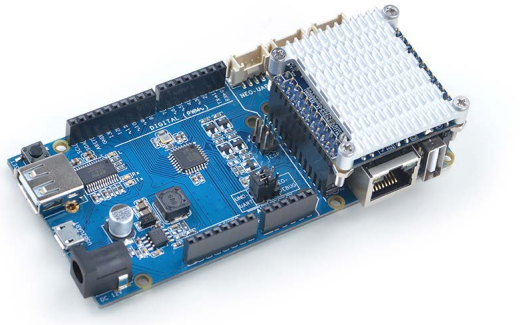
## Connect NanoHat PCM5102A to NEO2

The NanoHat PCM5102A module uses TI's DAC audio chip PCM5102A, a convenient and easy-to-use audio module for hobbyists. Here is a hardware setup:NanoHat PCM5102A



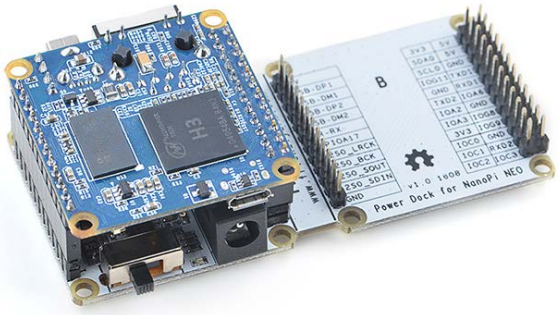
## Connect Arduino Compatible UNO Dock to NEO2

The UNO Dock module is an Arduino board compatible with Arduino UNO and works with Arduino programs.You can use Arduino IDE to run all Arduino programs on the Dock.It also exposes the NanoPi NEO2's pins.It converts 12V power input to 5V/2A output.You can search for various code samples from Ubuntu's ecosystem and run on the Dock. These features make it a powerful platform for IOT projects and cloud related applications. Here is a hardware setup:UNO Dock for NanoPi NEO v1.0



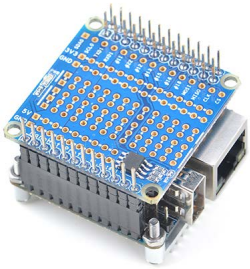
## Connect Power Dock to NEO2

The Power Dock for NanoPi NEO2 is a high efficiency power conversion module. It provides stable and reliable power source. Here is a hardware setup:Power Dock for NanoPi NEO



## Connect NanoHat Proto to NEO2

The NanoHat Proto is an expansion board which exposes NEO2's various pins.It has an onboard EEPROM for data storage.Here is a hardware setup:NanoHat Proto



### Connect Matrix - 2'8 SPI Key TFT to NanoPi NEO2

The Matrix-2'8_SPI_Key_TFT module is a 2.8" TFT LCD with resistive touch. It uses the ST7789S IC and XPT2046 resistive touch IC. It has SPI interface and three configurable user keys.Here is its wiki page Matrix - 2'8 SPI Key TFT


## 3D Printing Files

downloadfile

frameless

## Resources

### Datasheet & Schematics

- Schematics
  - NanoPi-NEO2-1701-Schematic.pdf
- Dimensional Diagram
  - NanoPi-NEO2-1701 pcb in dxf format
- H5 Datesheet
  Allwinner_H5_Datasheet_V1.0.pdf

## Update Log

### March-14-2017

- Released English Version

### April-5-2017

- Added sections 5.2 and 5.8

### May-7-2017

- Added sections 7: mainline support for H5
- Added sections 8: support for external modules

### May-17-2017

- Added sections 5.9: WiringNP support for H5

### May-24-2017

- Added section 3: Software Features

### June-4-2017

- Updated section 5.3.1
- Updated section 3: added more OS features

### June-8-2017

- Updated section 3.2: added support for RPi.GPIO_NP
- Added section 6.10: added support for RPi.GPIO_NP

### July-5-2017

- Updated sections 5.3.2, 6.5 and 6.8
- Added section 8.8: connect 2.8"TFT to NEO2

### July-9-2017

- Updated section 7.2