NanoPi NEO Core

From FriendlyELEC WiKi

查看中文

Contents

- 1 Introduction
- 2 Hardware Spec
- 3 Diagram, Layout and Dimension
  - 3.1 Layout
  - 3.2 Dimensional Diagram
- 4 Get Started
  - 4.1 Essentials You Need
  - 4.2 TF Cards We Tested
  - 4.3 Install OS
    - 4.3.1 Get Image Files
      - 4.3.1.1 Flash to eMMC
        - 4.3.1.1.1 Flash OS with eflasher Utility
- 5 Mini Shield for NanoPi NEO Core/Core2
- 6 Work with FriendlyCore
  - 6.1 Introduction
  - 6.2 System Login
  - 6.3 Configure System with npi-config
  - 6.4 Develop Qt Application
  - 6.5 Setup Program to AutoRun
  - 6.6 Extend TF Card's Section
  - 6.7 WiFi
  - 6.8 Ethernet Connection
  - 6.9 WiringPi and Python Wrapper
  - 6.10 Custom welcome message
  - 6.11 Modify timezone
  - 6.12 Set Audio Device
  - 6.13 Connect to USB Camera(FA-CAM202)
  - 6.14 Check CPU's Working Temperature
  - 6.15 Test Infrared Receiver
  - 6.16 Run Qt Demo
  - 6.17 How to install and use docker (for armhf system)
    - 6.17.1 How to Install Docker
    - 6.17.2 Test Docker installation
- 7 Work with OpenWrt
  - 7.1 Introduction
  - 7.2 System Login
  - 7.3 Manage Software Packages
  - 7.4 Check System Status
  - 7.5 Check Network->Interfaces Configurations
  - 7.6 USB WiFi
  - 7.7 Huawei's WiFi 2 mini(E8372H-155) Module
- 8 Make Your Own FriendlyCore
  - 8.1 Use Linux-4.14 BSP

- 9 Build Kernel Headers Package
  - 9.1 Software Version
  - 9.2 Install the required packages
  - 9.3 Build Kernel Headers Package
- 10 Installation=
  - 10.1 Testing
- 11 Connect External Modules to NEO Core
  - 11.1 Connect Mini Shield for NanoPi NEO Core/Core2 to NEO Core
  - 11.2 Connect Python Programmable NanoHat OLED to NEO Core
  - 11.3 Connect Python Programmable NanoHat Motor to NEO Core
  - 11.4 Connect NanoHat PCM5102A to NEO Core
  - 11.5 Connect Arduino Compatible UNO Dock to NEO Core
  - 11.6 Connect Power Dock to NEO Core
  - 11.7 Connect NanoHat Proto to NEO Core
  - 11.8 Connect Matrix - 2'8 SPI Key TFT to NanoPi NEO Core
- 12 3D Printing Files
- 13 Developer's Guide
- 14 Resources
  - 14.1 Datasheet & Schematics
- 15 Update Log
  - 15.1 Dec-1-2017

1 Introduction

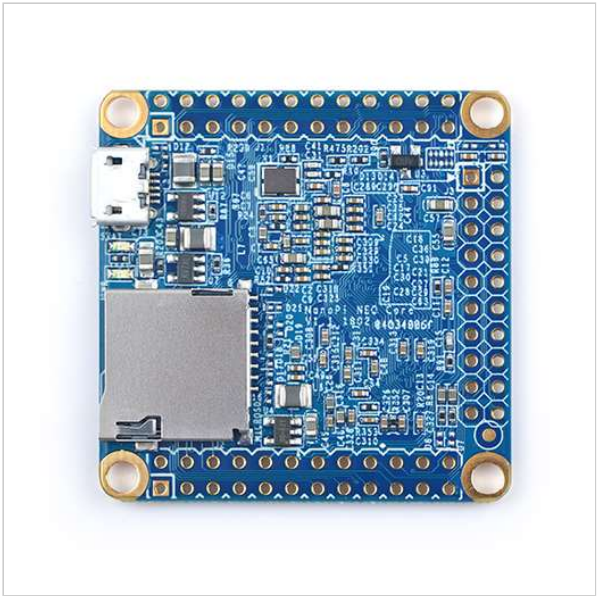
- The NanoPi NEO Core(abbreviated as "NEO Core") is an alternative NanoPi NEO that works like a CPU board with male pin-headers. It has the same form factor as the NanoPi NEO and same pin descriptions. The connectors and ports are populated to pin-headers on the NEO Core. The NanoPi NEO Core has ESD protection for its MicroUSB port and TF card slot. In addition the NEO Core can have an optional onboard eMMC flash which is preferred by industrial customers.
- The NEO Core uses a popular Allwinner H3 SoC and has onboard 256M/512M DDR3 RAM. FriendlyElec offers models with three eMMC options: 8GB/16GB/32GB and one that doesn't have eMMC at all.
- FriendlyElec migrated UbuntuCore with mainline kernel 4.14 for it.
- FriendlyElec develops a Mini Shield for NanoPi NEO Core/Core2 which has the same form factor as the RPi 3. When a NanoPi NEO Core is connected to this Mini Shield the whole assembled module can be well fit into a common RPi 3's case.

Overview

2 Hardware Spec

- CPU: Allwinner H3, Quad-core Cortex-A7 Up to 1.2GHz
- DDR3 RAM: 256MB/512MB DDR3 RAM
- Storage: NC/8GB/16GB/32GB eMMC
- MicroSD Slot x 1
- MicroUSB: OTG and power input

- GPIO: two 2.54mm spacing 12x2pin header,one 2.54mm spacing 10x2pin header
- Connectivity: 10/100M Ethernet(6Pin, included in 2.54mm pitch pin header)
- USB Host x3(included in 2.54mm pitch pin header)
- Debug Serial Port(4Pin, included in 2.54mm pitch pin header)
- Audio input/output Port(4Pin, included in 2.54mm pitch pin header)
- GPIO:It includes UART, SPI, I2C, IO etc
- PC Size: 40 x 40mm
- Power Supply: DC 5V/2A
- Temperature measuring range: -20℃ to 70℃
- OS/Software: U-boot，Ubuntu-Core
- Weight: xxg(WITHOUT Pin-headers)



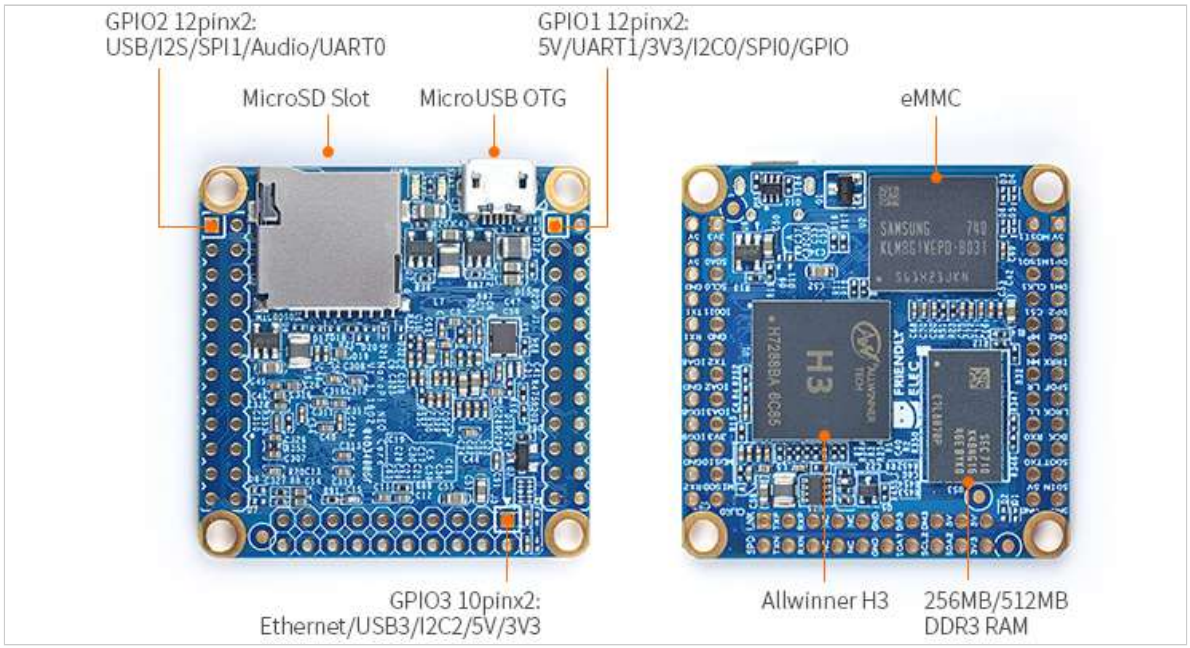
Front

3 Diagram, Layout and Dimension

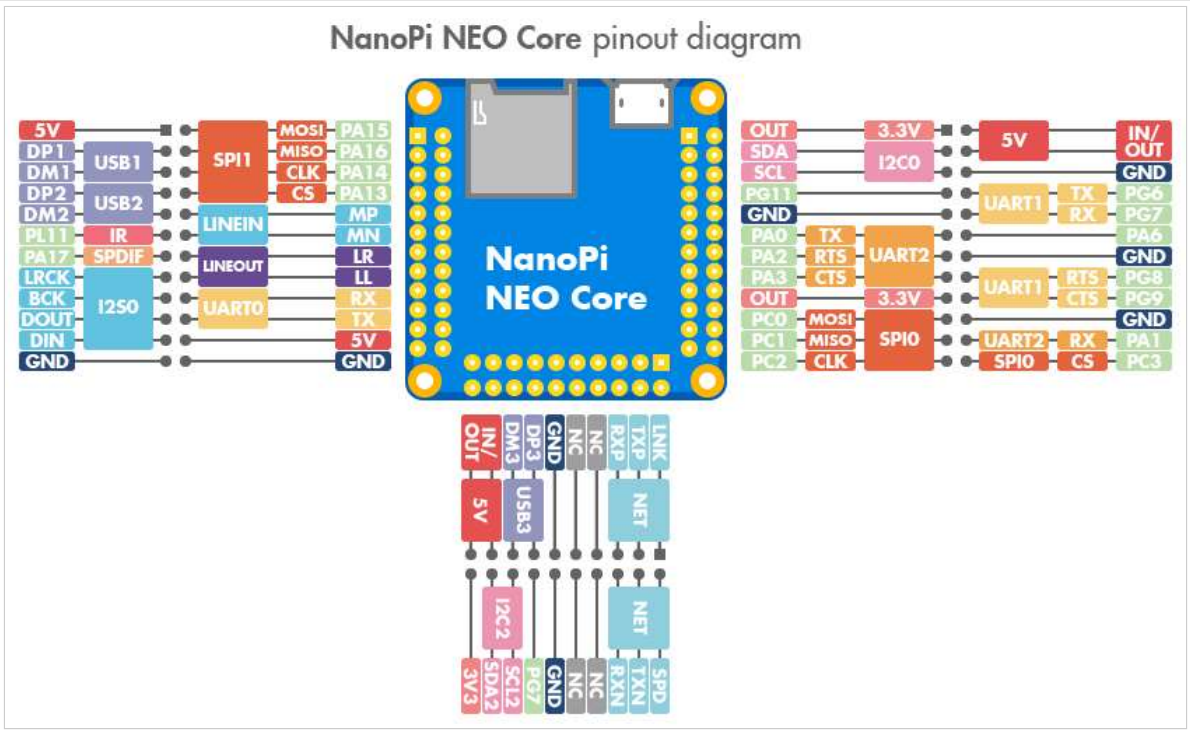
3.1 Layout

- **GPIO1 Pin Description**

Back



NanoPi NEO Core Layout



pinout

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

- **GPIO2 Pin Description**

Pin#	Name	Linux gpio	Pin#	Name	Linux gpio
1	VDD_5V		2	SPI1_MOSI / GPIOA15	15
3	USB-DP1		4	SPI1_MISO / GPIOA16	16
5	USB-DM1		6	SPI1_CLK / GPIOA14	14
7	USB-DP2		8	SPI1_CS / GPIOA13	13
9	USB-DM2		10	MICIN1P	
11	GPIOL11/IR-RX	363	12	MICIN1N	
13	SPDIF-OUT/GPIOA17	17	14	LINEOUTR	
15	PCM0_SYNC/I2S0_LRCK/I2C1_SCL		16	LINEOUTL	
17	PCM0_CLK/I2S0_BCK/I2C1_SDA		18	UART_RXD0 / GPIOA5 / PWM0	5
19	PCM0_DOUT/I2S0_SDOUT		20	UART_TXD0 / GPIOA4	4
21	PCM0_DIN/I2S0_SDIN		22	VDD_5V	
23	GND		24	GND	

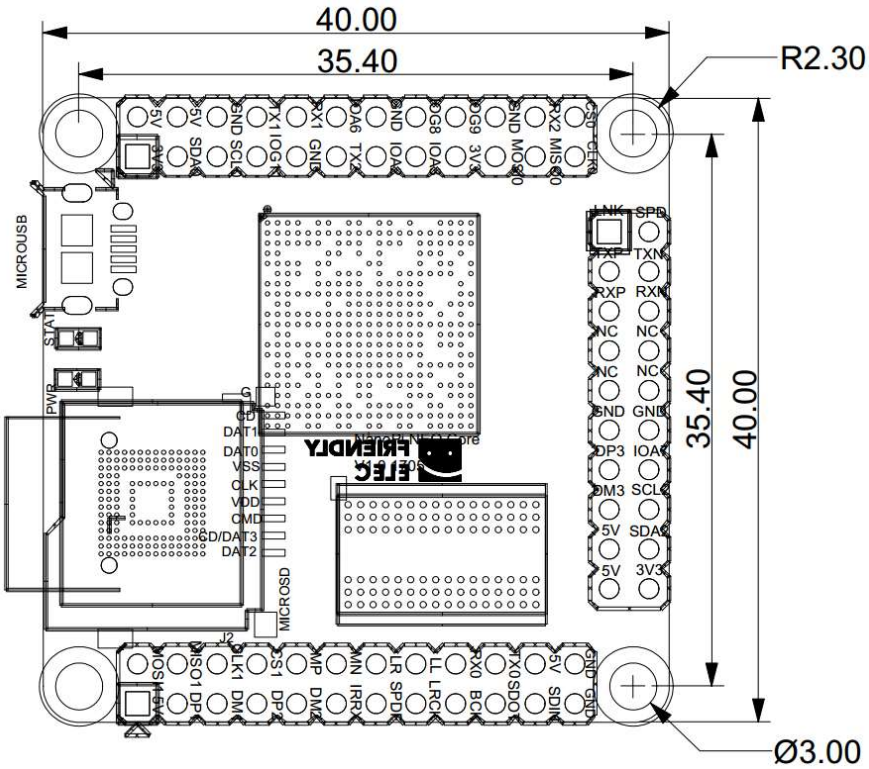
- **GPIO3 Pin Description**

Pin#	Name	Linux gpio	Pin#	Name	Linux gpio
1	EPHY-LINK-LED		2	EPHY-SPD-LED	
3	EPHY-TXP		4	EPHY-TXN	
5	EPHY-RXP		6	EPHY-RXN	
7	NC		8	NC	
9	NC		10	NC	
11	GND		12	GND	
13	USB-DP3		14	GPIOA7	7
15	USB-DM3		16	I2C2_SCL / GPIOE12	
17	5V		18	I2C2_SDA / GPIOE13	
19	5V		20	SYS_3.3V	

Note:

1. SYS_3.3V: 3.3V power output
2. VVDD_5V: 5V power input/output. When the external device's power is greater than the MicroUSB's the external device is charging the board otherwise the board powers the external device. The input range is 4.7V ~ 5.6V
3. All pins are 3.3V, output current is 5mA
4. For more details refer to its schematic NanoPi-NEO-Core-V1.1-1802-Schematic.pdf (https://wiki.friendlyelec.com/wiki/images/a/a4/Nanopi_neo_core-v1.1-1802.pdf)

3.2 Dimensional Diagram



For more details refer to the document: NanoPi NEO Core V1.1 1802 pcb in dxf format (https://wiki.friendlyelec.com/wiki/index.php/File:NanoPi_NEO_Core-V1.1_pcb-Dim.rar)

4 Get Started

4.1 Essentials You Need

Before starting to use your NanoPi NEO Core get the following items ready

- NanoPi NEO Core
- microSD Card/TF Card: Class 10 or Above, minimum 8GB SDHC
- microUSB power. A 5V/2A power is a must
- A host computer running Ubuntu 18.04 64 bit system

4.2 TF Cards We Tested

To make your device 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 MicroSDHC V30 32GB Extreme Pro **(Developer choice)**

- SanDisk 32GB High Endurance Video MicroSDHC Card with Adapter for Dash Cam and Home Monitoring Systems **(High reliability)**

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

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

- 川宇 8G C10 High Speed class10 micro SD card:

4.3 Install OS

4.3.1 Get Image Files

Visit this link download link (http://download.friendlyelec.com/nanopineocore) to download image files and the flashing utility:

Image Files:	
nanopi-neo-core_sd_friendlycore-xenial_4.14_armhf_YYYYMMDD.img.zip	Base on UbuntuCore, kernel: Linux-4.14
nanopi-neo-core_sd_friendlywrt_4.14_armhf_YYYYMMDD.img.zip	Base on OpenWrt, kernel:Linux-4.14
nanopi-neo-core_eflasher_friendlycore-xenial_4.14_armhf_YYYYMMDD.img.zip	eflasher image, for flashing FriendlyCore(Linux-4.14) to eMMC
nanopi-neo-core_eflasher_openwrt_4.14_armhf_YYYYMMDD.img.zip	eflasher image, for flashing OpenWrt(Linux-4.14) to eMMC
Flash Utility:	
win32diskimager.rar	Windows utility for flashing Debian image. Under Linux users can use "dd"

4.3.1.1 Flash to eMMC

4.3.1.1.1 Flash OS with eflasher Utility

- For more details about eflasher refer to the wiki link: EFlasher。
- Extract the eflasher Image and win32diskimager.rar files. Insert a TF card(at least 4G) 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.
- Insert this card into your board's BOOT slot and power on (with a 5V/2A power source). If the green LED is on and the blue LED is blinking this indicates your board has successfully booted.
- If your board doesn't support HDMI or no monitor is connected you can select an OS by running the following command:

```
$ su root
$ eflasher
```

The password for "root" is "fa".

We take "nanopi-neo-core_eflasher_friendlycore-xenial_4.14_armhf_YYYYMMDD.img" as an example. After you run the "eflasher" command you will see the following messages:

```
-----------------------------------------------------------------------
  EFlasher v1.2 b190412 running on NanoPi
    Doc: http://wiki.friendlyarm.com/wiki/index.php/EFlasher
    eMMC: 14.56 GB
-----------------------------------------------------------------------
  # Select an OS to install:
    1) friendlycore-xenial_4.14

  # Select your backup target device:
     tf) [*] TF card  (/dev/mmcblk0p3 - 790.69 MB free - 3.44 GB total )
    usb) [ ] USB disk  (<none>)

  # Backup eMMC flash to TF card:
    Not enough free disk space on your TF card

  # Restore eMMC flash from backup file:
    No backup files found

  # Configure automatic job:
    aui) Automatic installing (Curr:Off)
    aur) Automatic restoring (Curr:Off)

  # Format drive
    ftf) Format TF card back to original size
-----------------------------------------------------------------------
>>> Enter an option (1/tf/usb/aui/aur/ftf) :
```

Type "1", select writing friendlycore system to eMMC you will see the following messages:

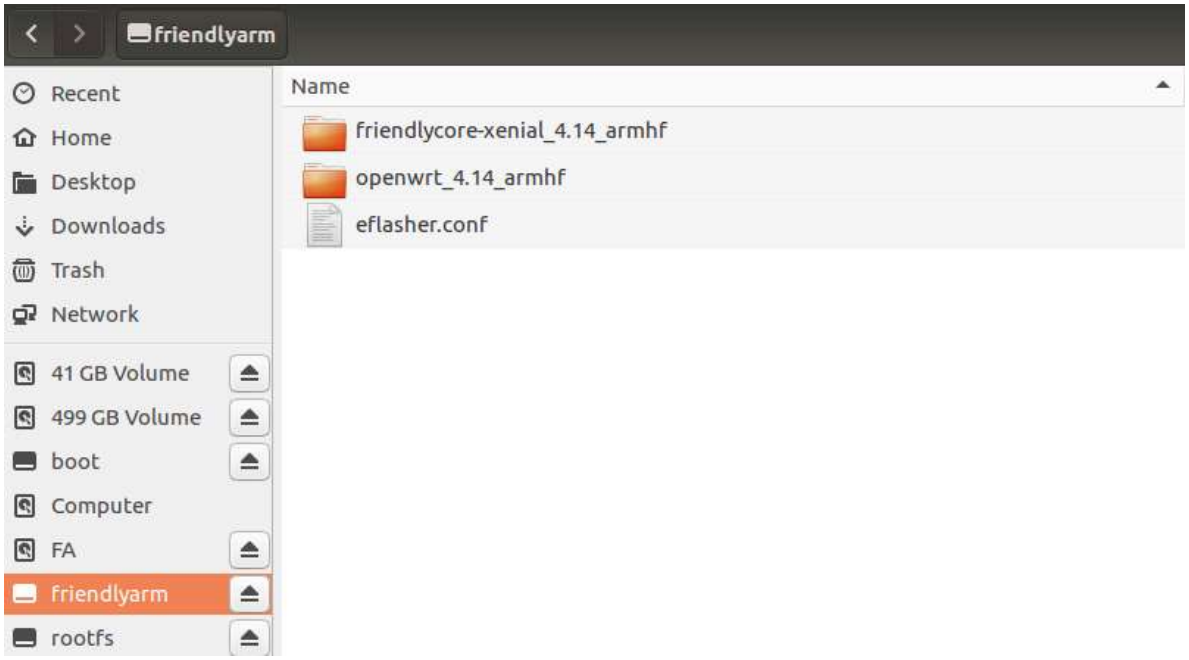
```
-----------------------------------------------------------------------
  Ready to Go with FriendlyCore
-----------------------------------------------------------------------
  Ready to install
  Version:
                2019-04-25
  Path:
                /mnt/sdcard/friendlycore-xenial_4.14_armhf
  Image files:
                u-boot-sunxi-with-spl.bin 1.99 MB
                boot.img 40.00 MB
                rootfs.img 2.44 GB

  Total size:
                2.48 GB
  Kernel parameter:
                Default
-----------------------------------------------------------------------
>>> Do you wish to continue? (yes/no) :
```

Type "yes" to start installation:

```
-----------------------------------------------------------------------
  Installing FriendlyCore
-----------------------------------------------------------------------
  Speed: 17.65 MB/s
  Remaining Time: 00:02:10
  [=====_____________________________________________]  11%
-----------------------------------------------------------------------
  >>>Enter "c" to cancel.
```

After it is done power off the system, take off the TF card, power on again your system will be booted from eMMC.

- If you want to flash other system to eMMC you can download the whole images-for-eflasher directory and extract the package under that directory to the FRIENDLYARM partition of an installation SD card.



5 Mini Shield for NanoPi NEO Core/Core2

Here is a setup where we connect a NanoPi NEO Core to a Mini Shield for NanoPi NEO Core/Core2. Here is an introduction to Mini Shield for NanoPi NEO Core/Core2 Mini Shield (https://wiki.friendlyelec.com/wiki/index.php/Mini_Shield_for_NanoPi_NEO_Core/Core2/zh#.E4.BB.8B.E7.BB.8D)

:

6 Work with FriendlyCore

6.1 Introduction

FriendlyCore is a light Linux system without X-windows, based on ubuntu core, It uses the Qt-Embedded's GUI and is popular in industrial and enterprise applications.

Besides the regular Ubuntu Core's features FriendlyCore has the following additional features:

- it integrates Qt4.8;
- it integrates NetworkManager;
- it has bluez and Bluetooth related packages;
- it has alsa packages;
- it has npi-config;
- it has RPiGPIO, a Python GPIO module;
- it has some Python/C demo in /root/ directory;
- it enables 512M-swap partition;

6.2 System Login

- If your board is connected to an HDMI monitor you need to use a USB mouse and keyboard.
- 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 board to a PC via the PSU-ONECOM and you can power on your board from either the PSU-ONECOM or its MicroUSB:

You can use a USB to Serial conversion board too.
Make sure you use a 5V/2A power to power your board from its MicroUSB port:

- FriendlyCore 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

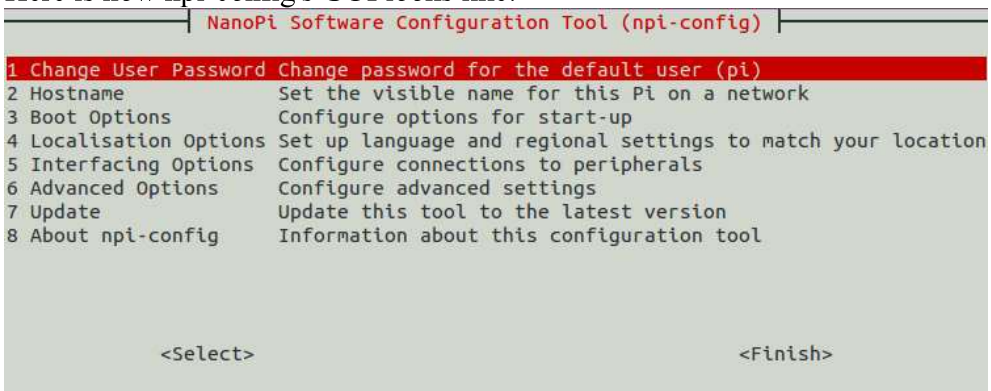
```
$ sudo apt-get update
```

6.3 Configure System with npi-config

The npi-config is a commandline utility which can be used to initialize system configurations such as user password, system language, time zone, Hostname, SSH switch , Auto login and etc. Type the following command to run this utility.

```
$ sudo npi-config
```

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



6.4 Develop Qt Application

Please refer to: How to Build and Install Qt Application for FriendlyELEC Boards

6.5 Setup Program to AutoRun

You can setup a program to autorun on system boot with npi-config:

```
sudo npi-config
```

Go to Boot Options -> Autologin -> Qt/Embedded, select Enable and reboot.

6.6 Extend TF Card's Section

When FriendlyCore is loaded the TF card's section will be automatically extended.You can check the section's size by running the following command:

```
$ df -h
```

6.7 WiFi

For either an SD WiFi or a USB WiFi you can connect it to your board in the same way. The APXX series WiFi chips are SD WiFi chips. By default FriendlyElec's system supports most popular USB WiFi modules. Here is a list of the USB WiFi modules we tested:

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

You can use the NetworkManager utility to manage network. You can run "nmcli" in the commandline utility to start it. Here are the commands to start a WiFi connection:

- Change to root

```
$ su root
```

- Check device list

```
$ nmcli dev
```

Note: if the status of a device is "unmanaged" it means that device cannot be accessed by NetworkManager. To make it accessed you need to clear the settings under "/etc/network/interfaces" and reboot your system.

- Start WiFi

```
$ nmcli r wifi on
```

- Scan Surrounding WiFi Sources

```
$ nmcli dev wifi
```

- Connect to a WiFi Source

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

The "SSID" and "PASSWORD" need to be replaced with your actual SSID and password.If you have multiple WiFi devices you need to specify the one you want to connect to a WiFi source with iface
If a connection succeeds it will be automatically setup on next system reboot.

For more details about NetworkManager refer to this link: Use NetworkManager to configure network settings

If your USB WiFi module doesn't work most likely your system doesn't have its driver. For a Debian system you can get a driver from Debian-WiFi (https://wiki.debian.org/WiFi) and install it on your system. For a Ubuntu system you can install a driver by running the following commands:

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

In general all WiFi drivers are located at the "/lib/firmware" directory.

6.8 Ethernet Connection

If a board is connected to a network via Ethernet before it is powered on it will automatically obtain an IP with DHCP activated after it is powered up. If you want to set up a static IP refer to: Use NetworkManager to configure network settings。

6.9 WiringPi and Python Wrapper

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

6.10 Custom welcome message

The welcome message is printed from the script in this directory：

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

For example, to change the FriendlyELEC LOGO, you can change the file /etc/update-motd.d/10-header. For example, to change the LOGO to HELLO, you can change the following line：

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

To:

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

6.11 Modify timezone

For exampe, change to Shanghai timezone:

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

6.12 Set Audio Device

If your system has multiple audio devices such as HDMI-Audio, 3.5mm audio jack and I2S-Codec you can set system's default audio device by running the following commands.

- After your board is booted run the following commands to install alsa packages:

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

- After installation is done you can list all the audio devices by running the following command. Here is a similar list you may see after you run the command:

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

"card 0" is HDMI-Audio, "card 1" is 3.5mm audio jack and "card 2" is I2S-Codec. You can set default audio device to HDMI-Audio by changing the "/etc/asound.conf" file as follows:

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

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

If you change "card 0" to "card 1" the 3.5mm audio jack will be set to the default device.
Copy a .wav file to your board and test it by running the following command:

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

You will hear sounds from system's default audio device.
If you are using H3/H5/H2+ series board with mainline kernel, the easier way is using npi-config。

6.13 Connect to USB Camera(FA-CAM202)

The FA-CAM202 is a 200M USB camera. Connect your board to camera module. Then boot OS, connect your board to a network, log into the board as root and run "mjpg-streamer":

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

You need to change the start.sh script and make sure it uses a correct /dev/videoX node. You can check your camera's node by running the following commands:

```
$ apt-get install v4l-utils
$ v4l2-ctl -d /dev/video0 -D
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
        ...
```

The above messages indicate that "/dev/video0" is camera's device node.The mjpg-streamer application is an open source video steam server. After it is successfully started the following messages will be popped up:

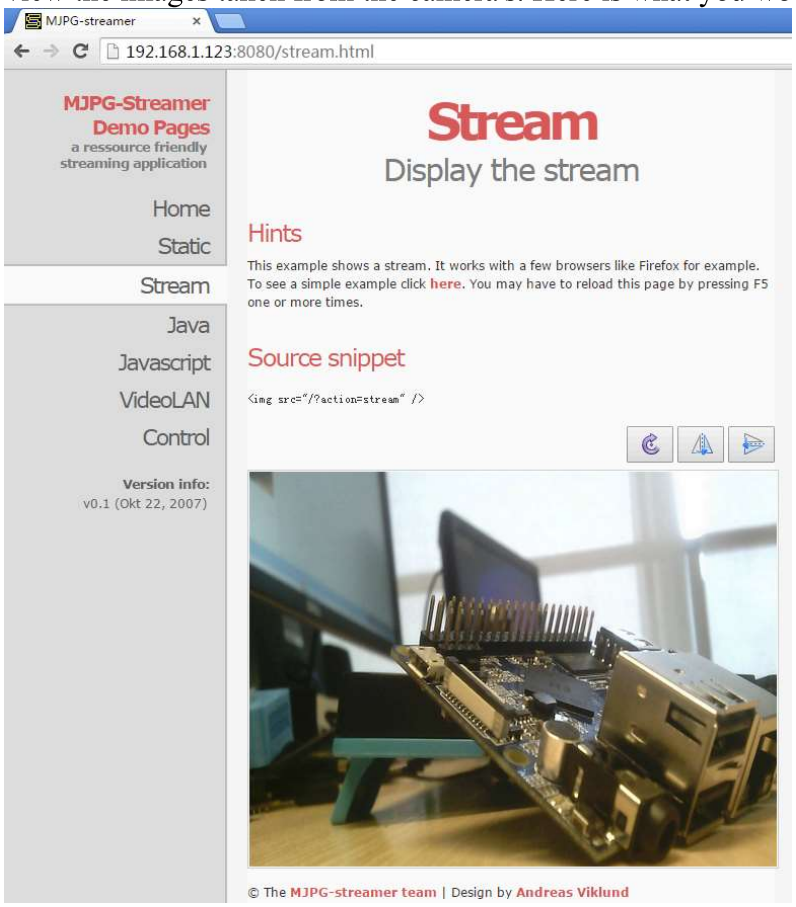
```
$ ./start.sh
 i: Using V4L2 device.: /dev/video0
 i: Desired Resolution: 1280 x 720
 i: Frames Per Second.: 30
 i: Format............: YUV
 i: JPEG Quality......: 90
 o: www-folder-path...: ./www/
 o: HTTP TCP port.....: 8080
 o: username:password.: disabled
 o: commands..........: enabled
```

start.sh runs the following two commands:

```
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"
```

Here are some details for mjpg_streamer's major options:
-i: input device. For example "input_uvc.so" means it takes input from a camera;
-o: output device. For example "output_http.so" means the it transmits data via http;
-d: input device's subparameter. It defines a camera's device node;
-y: input device's subparameter. It defines a camera's data format: 1:yuyv, 2:yvyu, 3:uyvy 4:vyuy. If this option isn't defined MJPEG will be set as the data format;
-r: input device's subparameter. It defines a camera's resolution;
-f: input device's subparameter. It defines a camera's fps. But whether this fps is supported depends on its driver;
-q: input device's subparameter. It defines the quality of an image generated by libjpeg soft-encoding;
-n: input device's subparameter. It disables the dynctrls function;
-fb: input device's subparameter. It specifies whether an input image is displayed at "/dev/fbX";
-w: output device's subparameter. It defines a directory to hold web pages;

In our case the board's IP address was 192.168.1.230. We typed 192.168.1.230:8080 in a browser and were able to view the images taken from the camera's. Here is what you would expect to observe:



6.14 Check CPU's Working Temperature

You can get CPU's working temperature by running the following command:

```
$ cpu_freq
Aavailable 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
```

This message means there are currently four CPUs working. All of their working temperature is 26.5 degree in Celsius and each one's clock is 624MHz.
Set CPU frequency:

```
$ cpu_freq -s 1008000
Aavailable 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
```

6.15 Test Infrared Receiver

Note: Please Check your board if IR receiver exist.
By default the infrared function is disabled you can enable it by using the npi-config utility:

```
$ npi-config
    6 Advanced Options     Configure advanced settings
        A8 IR              Enable/Disable IR
            ir Enable/Disable ir[enabled]
```

Reboot your system and test its infrared function by running the following commands:

```
$ apt-get install ir-keytable
$ echo "+rc-5 +nec +rc-6 +jvc +sony +rc-5-sz +sanyo +sharp +mce_kbd +xmp" > /sys/class/rc/rc0/protocols   # Enable infrared
$ ir-keytable -t
Testing events. Please, press CTRL-C to abort.
```

"ir-keytable -t" is used to check whether the receiver receives infrared signals. You can use a remote control to send infrared signals to the receiver. If it works you will see similar messages as follows:

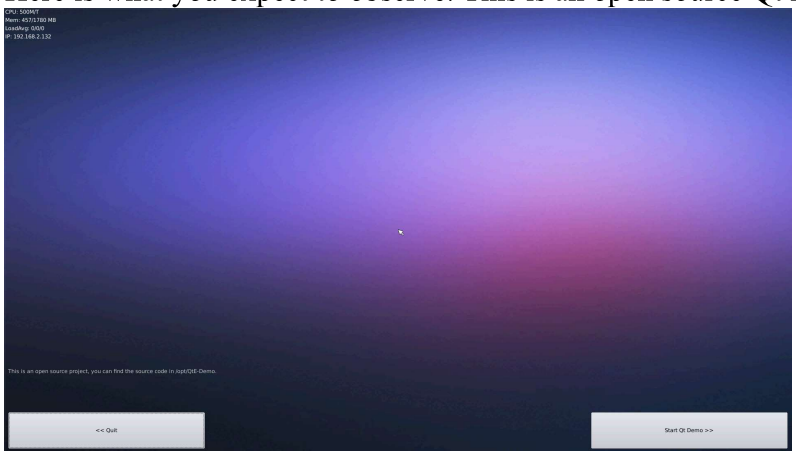
```
1522404275.767215: event type EV_MSC(0x04): scancode = 0xe0e43
1522404275.767215: event type EV_SYN(0x00).
1522404278.911267: event type EV_MSC(0x04): scancode = 0xe0e42
1522404278.911267: event type EV_SYN(0x00).
```

6.16 Run Qt Demo

Run the following command

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

Here is what you expect to observe. This is an open source Qt Demo (https://github.com/friendlyarm/QtE-Demo):



6.17 How to install and use docker (for armhf system)

6.17.1 How to Install Docker

Run the following commands：

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

6.17.2 Test Docker installation

Test that your installation works by running the simple docker image:

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

7 Work with OpenWrt

7.1 Introduction

OpenWrt is a highly extensible GNU/Linux distribution for embedded devices.Unlike many other distributions for routers, OpenWrt is built from the ground up to be a full-featured, easily modifiable operating system for embedded devices. In practice, this means that you can have all the features you need with none of the bloat, powered by a modern Linux kernel. For more details you can refer to:OpenWrt Website (https://openwrt.org/).

7.2 System Login

- **Login via Serial Port**

When you do kernel development you'd better get a serial communication board. After you connect your board to a serial communication board you will be able to do development work from a commandline utility.
Here is a hardware setup:
After you connect your board to a serial communication board (e.g. FriendlyElec's serial communication board) you

can power the whole system from either the DC port on the serial communication board or the MicroUSB port(if there is one) on your board:

or you can use a USB to serial board and power on the whole system at the MicroUSB port with a 5V/2A power:

By default you will login as root without a password. You can use "passwd" to set a password for root.

```
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:/#
```

On first boot the system will automatically extend the file system on the TF card to the max capacity:

```
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.
```

Please wait for this to be done.

- **Login via SSH**

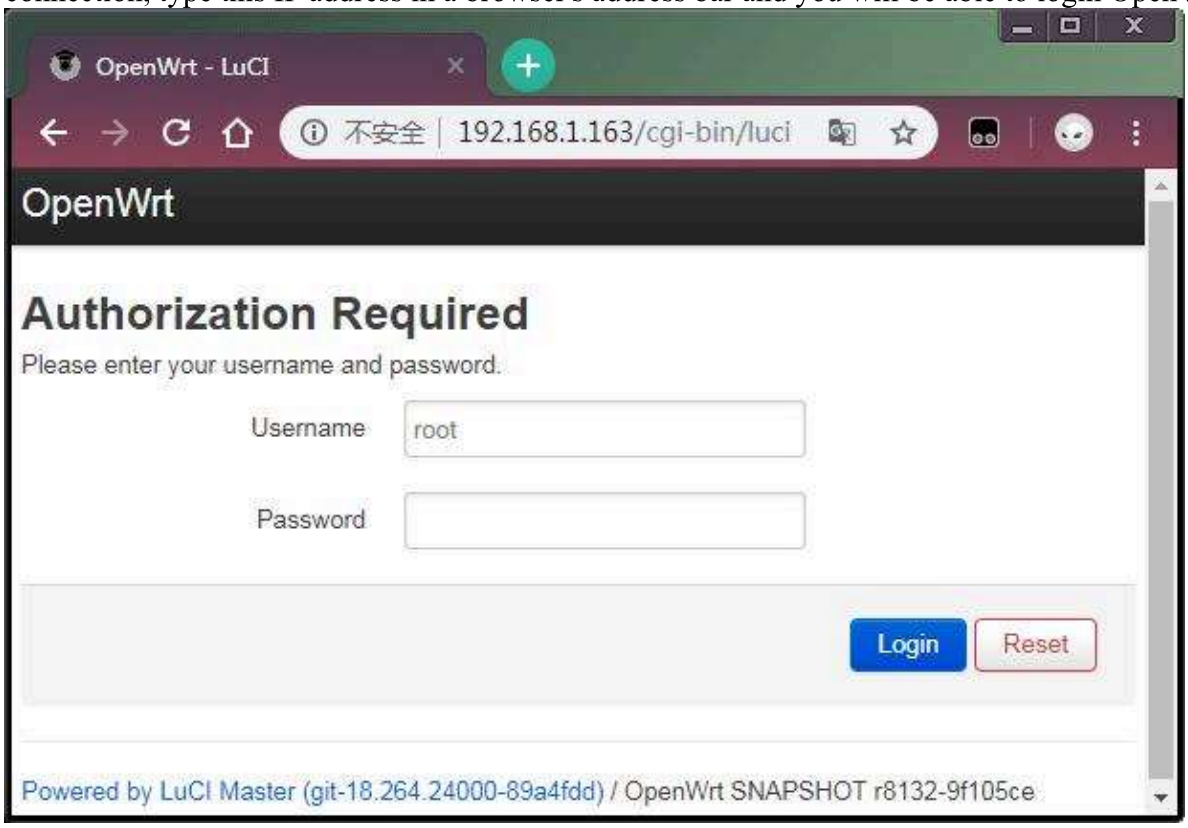
In FriendlyElec's OpenWrt system the Ethernet(eth0) is configured as WAN.
Before power on your board make sure your board is connected to a master router's LAN with an Ethernet cable and the eth0 will be assigned an IP address by DHCP.
For example, if your eth0 is assigned an IP address 192.168.1.163 you can login with SSH by running the following command:

```
$ ssh root@192.168.1.163
```

You can login without a password.

- **Login via Web**

You can login OpenWrt via a LuCI Web page.
After you go through all the steps in <Login via SSH> and get an IP address e.g. 192.168.1.163 for the Ethernet connection, type this IP address in a browser's address bar and you will be able to login OpenWrt-LuCI:



By default you will login as root without a password, just click on "Login" to login.

7.3 Manage Software Packages

OpenWrt has a package management utility: opkg. You can get its details by running the following command:

```
$ 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
...
```

These are just part of the manual. Here are some popular opkg commands.

- Update Package List

Before you install a package you'd better update the package list:

```
$ opkg update
```

- Check Available Packages

```
$ opkg list
```

At the time of writing there are 3241 packages available.

- Check Installed Packages:

```
$ opkg list-installed
```

At the time of writing 124 packages have been installed.

- Install/Delete Packages:

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

- Check Files Contained in Installed Packages:

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

- Install Chinese Language Package for LuCI

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

- Check Changed Files:

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

- Reference Links:
  - openwrt opkg (https://openwrt.org/docs/guide-user/additional-software/opkg)

7.4 Check System Status

- **Check CPU Temperature & Frequency via Commandline**

```
$ cpu_freq
Aavailable 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
```

These messages mean that there are four CPU cores working online simultaneously. Each core's temperature is 26.5 degrees in Celsius, the scheduling policy is on-demand and the working frequency is 624MHz. You can set the frequency by running the following command:

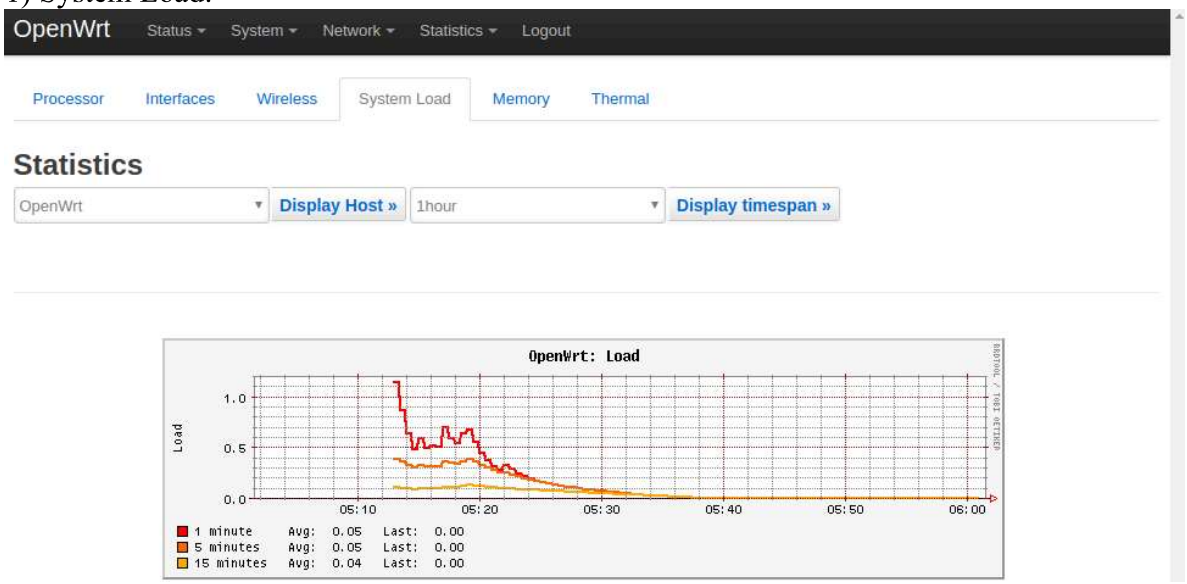
```
$ cpu_freq -s 1008000
Aavailable frequency(KHz):
        480000 624000 816000 1008000
Current frequency(KHz):
        CPU0 online=1 temp=36702C governor=userspace freq=1008000KHz
        CPU1 online=1 temp=36702C governor=userspace freq=1008000KHz
        CPU2 online=1 temp=36702C governor=userspace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

These messages mean four CPU cores are working online. Each core's temperature is 26.5 degrees. Each core's governor is on demand and the frequency is 480 MHz.

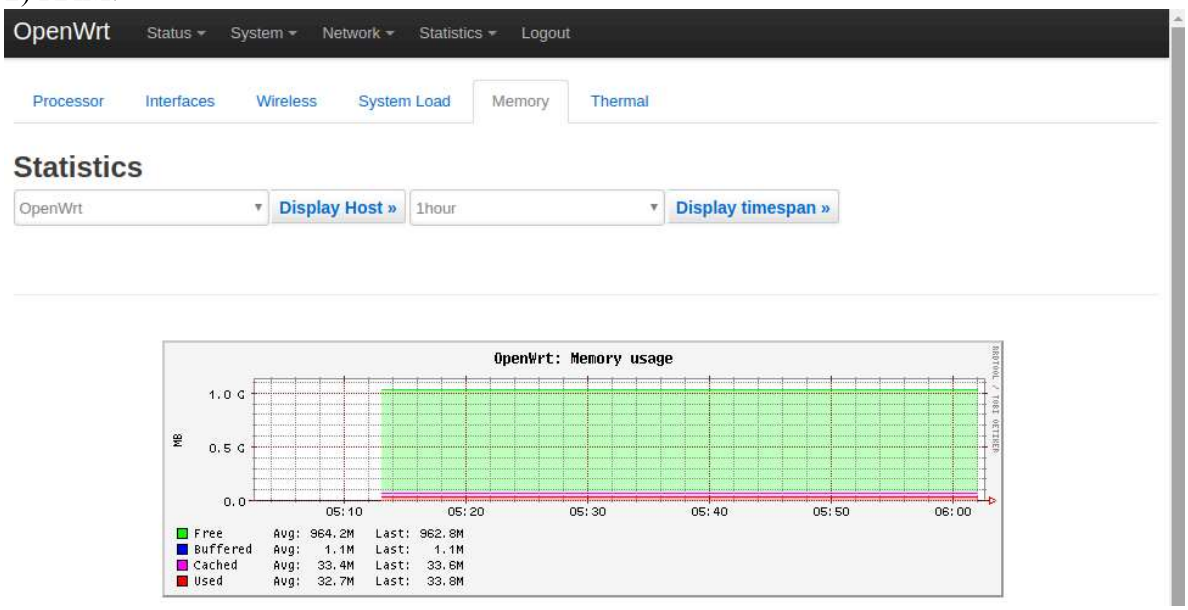
- **Check System Status on OpenWrt-LuCI Web Page**

After open the OpenWrt-LuCI page, go to "Statistics ---> Graphs" and you will see various system statistics e.g.:

1) System Load:



2) RAM:



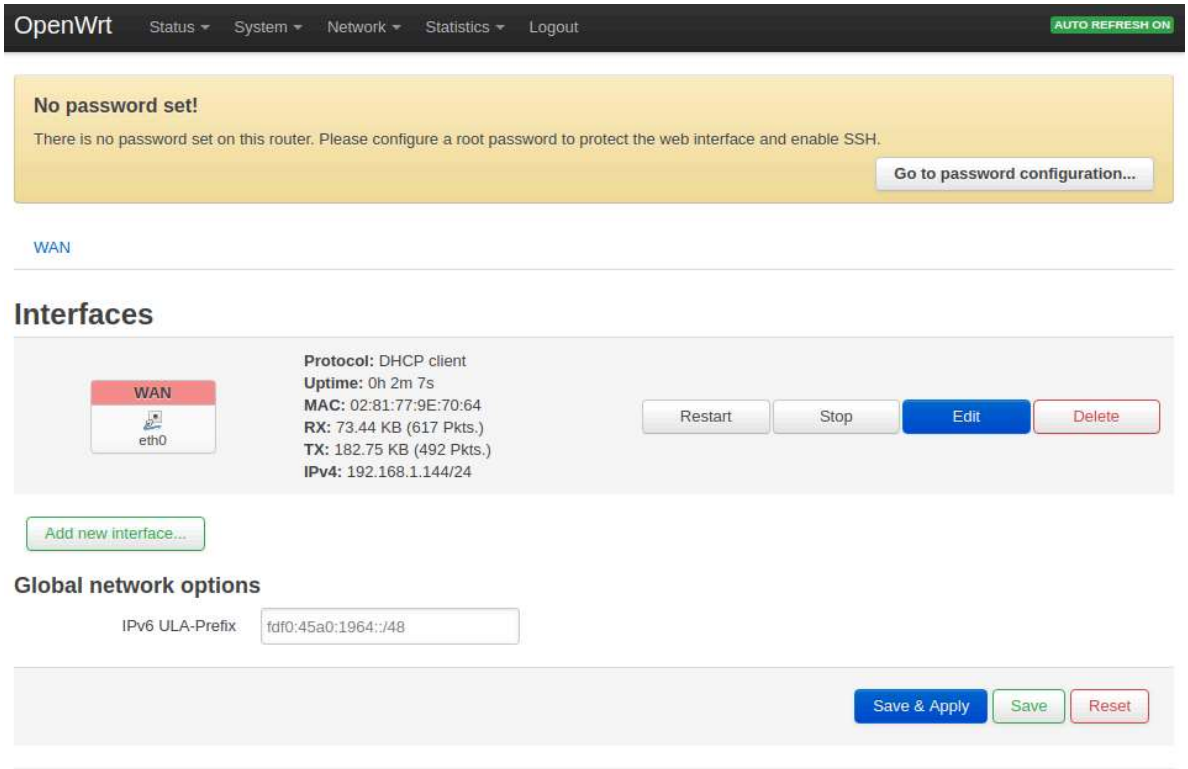
3) CPU Temperature:

All the statistics listed on the Statistics page are presented by the luci-app-statistics package which uses the Collectd utility to collect data and presents them with the RRDtool utility.
If you want to get more statistics you can install other collectd-mod-* packages. All collectd-mod-* packages use the same configuration file: /etc/config/luci_statistics.

- Reference Links:
  - 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 Check Network->Interfaces Configurations

- After open the OpenWrt-LuCI page, go to "Network" ---> "Interfaces" and you will see the current network's configurations:



- All the configurations listed on the Network->Interfaces page are stored in the "/etc/config/network" file.

7.6 USB WiFi

Currently the NanoPi NEO2 Black only works with a RTL8821CU USB WiFi dongle, plug and play. After this module is connected to the board it will by default work under AP mode and the hotspot's name is "rtl8821cu-mac address" and the password is "password";

7.7 Huawei's WiFi 2 mini(E8372H-155) Module

After this module is connected to the board it will be plug and play. The hotspot's name is "HUAWEI-8DA5". You can connect a device to the internet by connecting to this hotspot.

8 Make Your Own FriendlyCore

8.1 Use Linux-4.14 BSP

The NanoPi NEO Core has gotten support for kernel Linux-4.14. For more details about how to use mainline u-boot and Linux-4.14 refer to :Building U-boot and Linux for H5/H3/H2+

9 Build Kernel Headers Package

The following commands need to be executed on the development board:

9.1 Software Version

The OS image file name: nanopi-XXX_sd_friendlycore-focal_4.14_armhf_YYYYMMDD.img

```
$ 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
```

9.2 Install the required packages

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

9.3 Build Kernel Headers Package

```
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
```

The following message is displayed to indicate completion:

```
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:debarch} used, but is not defined
dpkg-genchanges: info: binary-only upload (no source code included)
```

10 Installation=

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

10.1 Testing

To compile the pf_ring module as an example, refer to the documentation: 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
```

After compiling, use insmod to try to load the module:

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

11 Connect External Modules to NEO Core

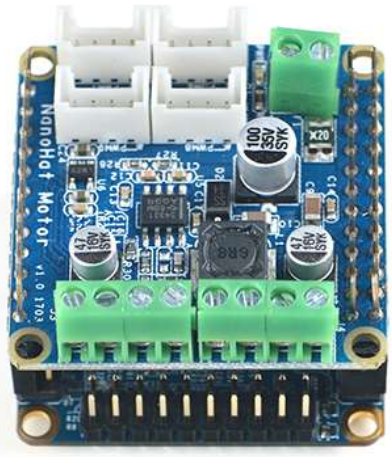
11.1 Connect Mini Shield for NanoPi NEO Core/Core2 to NEO Core

11.2 Connect Python Programmable NanoHat OLED to NEO Core

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

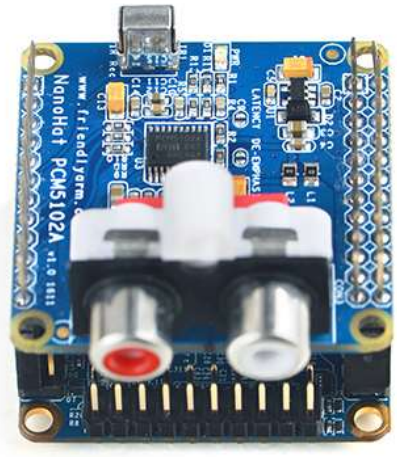
11.3 Connect Python Programmable NanoHat Motor to NEO Core

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



11.4 Connect NanoHat PCM5102A to NEO Core

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



11.5 Connect Arduino Compatible UNO Dock to NEO Core

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 NEO Core'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

11.6 Connect Power Dock to NEO Core

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

11.7 Connect NanoHat Proto to NEO Core

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

11.8 Connect Matrix - 2'8 SPI Key TFT to NanoPi NEO Core

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

12 3D Printing Files

13 Developer's Guide

- System Development
  - Building U-boot and Linux for H5/H3/H2+
  - How to Build FriendlyWrt
  - Qt dev: How to Build, Install and Setting Qt Application

- Image Utilities
  - How to make your own SD-bootable ROM
  - How to use overlayfs on Linux
  - EFlasher

- System Configurations
  - npi-config
  - Use NetworkManager to configure network settings

- Hardware Access
  - WiringNP: NanoPi NEO/NEO2/Air GPIO Programming with C
  - RPi.GPIO : NanoPi NEO/NEO2/Air GPIO Programming with Python
  - Hardware Misc (https://wiki.friendlyelec.com/wiki/index.php/Main_Page#Hardware_Misc_.28TBD.29)
  - Matrix (https://wiki.friendlyelec.com/wiki/index.php/Main_Page#Matrix)
  - BakeBit (https://wiki.friendlyelec.com/wiki/index.php/Main_Page#BakeBit)
  - HATs&Docks (https://wiki.friendlyelec.com/wiki/index.php/Main_Page#HATs.26Docks)

14 Resources

14.1 Datasheet & Schematics

- Schematic
  - NanoPi-NEO-Core-V1.0-1705-Schematic.pdf (https://wiki.friendlyelec.com/wiki/images/7/7a/NANOPI_NEO_CORE-V1.0.pdf)
  - NanoPi-NEO-Core-V1.1-1802-Schematic.pdf (https://wiki.friendlyelec.com/wiki/images/a/a4/Nanopi_neo_core-v1.1-1802.pdf)
- Dimensional Diagram
  - NanoPi-NEO-Core-1705 pcb的dxf文件 (https://wiki.friendlyelec.com/wiki/index.php/File:NanoPi_NEO_Core-1_0_1705_Dimensions.rar)
  - NanoPi-NEO-Core-1802 pcb的dxf文件 (https://wiki.friendlyelec.com/wiki/index.php/File:NanoPi_NEO_Core-V1.1_pcb-Dim.rar)
- H3's datasheet
  - Allwinner_H3_Datasheet_V1.2.pdf (https://wiki.friendlyelec.com/wiki/images/4/4b/Allwinner_H3_Datasheet_V1.2.pdf)

15 Update Log

15.1 Dec-1-2017

- Released English version

Retrieved from "https://wiki.friendlyelec.com/wiki/index.php?title=NanoPi_NEO_Core&oldid=22850"

Category: Pages with broken file links

- This page was last modified on 21 March 2022, at 09:39.