

NanoPi R1

Contents

- 1 Introduction
- 2 Hardware Spec
- 3 Diagram, Layout and Dimension
 - 3.1 Layout
 - 3.2 Board Dimension
- 4 Get Started
 - 4.1 Essentials You Need
 - 4.2 TF Cards We Tested
 - 4.3 Install OS
 - 4.3.1 Download Image Files
 - 4.3.1.1 Flash to eMMC
 - 4.3.1.1.1 Flash OS with eflasher Utility
- 5 Work with OpenWrt
 - 5.1 Introduction
 - 5.2 System Login
 - 5.3 Manage Software Packages
 - 5.4 Check System Status
 - 5.5 Check Network->Interfaces Configurations
 - 5.6 Check Network->Wireless Configurations
 - 5.7 Check LED Configurations
 - 5.8 Check BUTTON Configurations
 - 5.9 USB WiFi
 - 5.10 Huawei's WiFi 2 mini(E8372H-155) Module
- 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 Transfer files using Bluetooth
 - 6.8 WiFi
 - 6.9 Ethernet Connection
 - 6.10 Custom welcome message
 - 6.11 Modify timezone
 - 6.12 Connect to USB Camera(FA-CAM202)
 - 6.13 Check CPU's Working Temperature
 - 6.14 How to install and use docker (for armhf system)
 - 6.14.1 How to Install Docker
 - 6.14.2 Test Docker installation
 - 6.15 Using RTC
- 7 Build Kernel Headers Package
 - 7.1 Software Version
 - 7.2 Install the required packages
 - 7.3 Build Kernel Headers Package



- 8 Installation=
 - 8.1 Testing
- 9 Developer's Guide
- 10 More OS Support
 - 10.1 DietPi
- 11 Resources
 - 11.1 Schematics and Datasheets
- 12 Update Log
 - 12.1 Jan-02-2019
 - 12.2 Jan-17-2019
 - 12.3 April-24-2019

1 Introduction

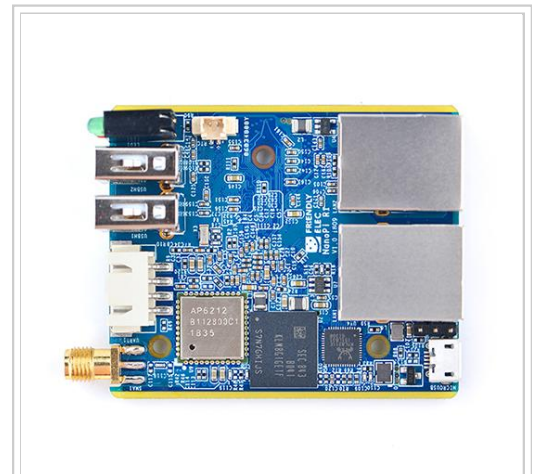
- The NanoPi R1("R1") is a complete open source board developed by FriendlyElec for makers, hobbyists, fans and etc.
- The NanoPi R1 has one Gbps Ethernet port and one Fast Ethernet port. It has an onboard 2.4G Wi-Fi module. FriendlyElec ported OpenWRT to the R1. With some additional settings it will work like a router. Its good networking performance and features make it a good platform for various network applications.

2 Hardware Spec

- CPU: Allwinner H3, Quad-core Cortex-A7 Up to 1.2GHz
- DDR3 RAM: 512MB/1GB
- Storage: NC/8GB eMMC
- Network:
 - 10/100/1000M Ethernet x 1 ,
 - 10/100M Ethernet x 1
- WiFi: 802.11b/g/n, with SMA antenna interface
- Bluetooth: 4.0 dual mode
- USB Host: Type-A x2
- MicroSD Slot x 1
- MicroUSB: for OTG and power input
- Debug Serial Port: 3Pin 2.54mm pitch pin-header
- UART: 4Pin 2.54mm pitch pin-header
- LED: LED x 3
- KEY: KEY x 1
- PC Size: 50.5 x 60mm
- Power Supply: DC 5V/2A
- Temperature measuring range: -20°C to 70°C
- OS/Software: U-boot, Ubuntu-Core, OpenWRT
- Weight: xxg



Overview



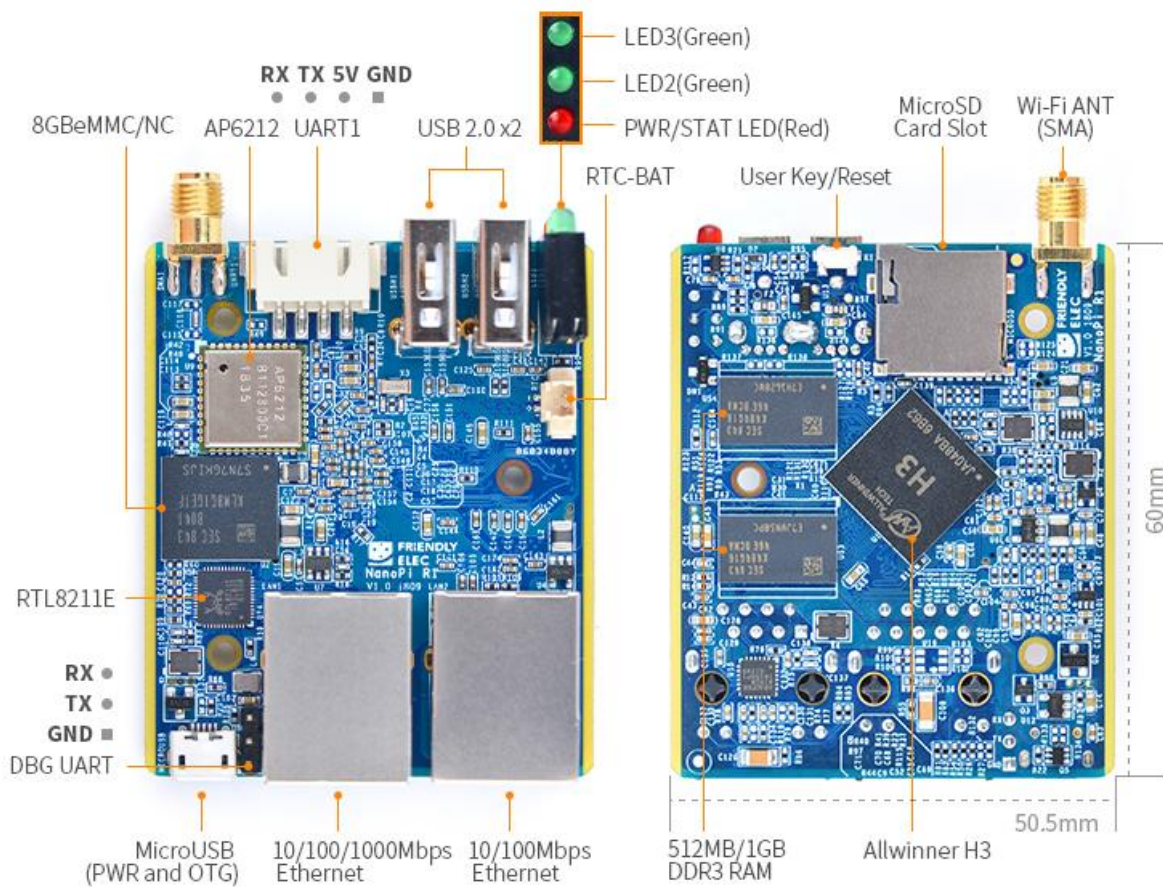
Front

3 Diagram, Layout and Dimension

3.1 Layout

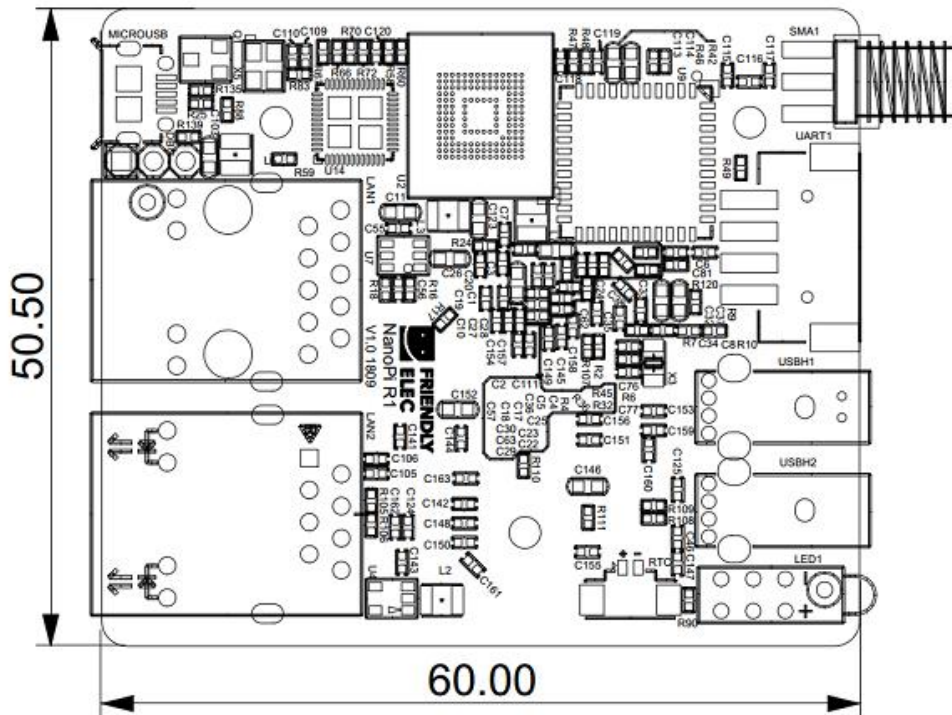


Back



For more details refer to the document: NanoPi R1 V1.0 1809 Schematic.pdf
(http://wiki.friendlyelec.com/wiki/images/a/ab/NanoPi_R1_V1.0_1809-Schematic.pdf)

3.2 Board Dimension



For more details refer to: NanoPi_R1 pcb file in dxf format (http://wib_DXF.rar)

4 Get Started

4.1 Essentials You Need

Before starting to use your NanoPi R1 get the following items ready

- NanoPi R1
- MicroSD Card/TF Card: Class 10 or Above, minimum 8GB SDHC
- MicroUSB 5V/2A power adapter
- 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 网站



- 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 Download Image Files

Go to download link (<http://download.friendlyelec.com/nanopir1>) to download the image files under the officail-ROMs directory and the flashing utility under the tools directory:

Image Files		
nanopi-r1_sd_friendlycore-xenial_4.14_armhf_YYYYMMDD.img.zip		Based on UbuntuCore and Linux-4.14 kernel
nanopi-r1_sd_friendlywrt_4.14_armhf_YYYYMMDD.img.zip		Base on OpenWrt and Linux-4.14 kernel
nanopi-r1_eflasher_friendlycore-xenial_4.14_armhf_YYYYMMDD.img.zip		eflasher image which is used to install FriendlyCore(Linux-4.14) to eMMC
nanopi-r1_eflasher_friendlywrt_4.14_armhf_YYYYMMDD.img.zip		eflasher image which is used to install OpenWrt(Linux-4.14) to eMMC
Flashing Utility		
win32diskimager.rar		Windows utility. 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-rl_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

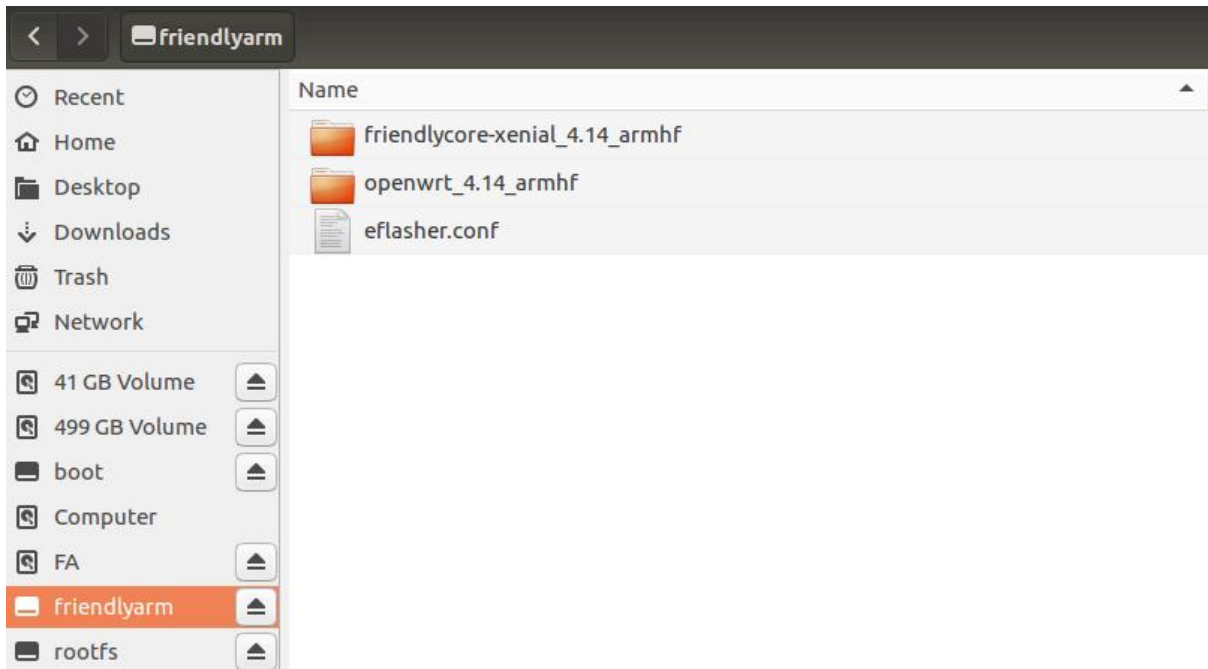
Total size:
    2.48 GB
Kernel parameter:
    Default
-----
>>> Do you wish to continue? (yes/no) :
```

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 Work with OpenWrt

5.1 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/>).

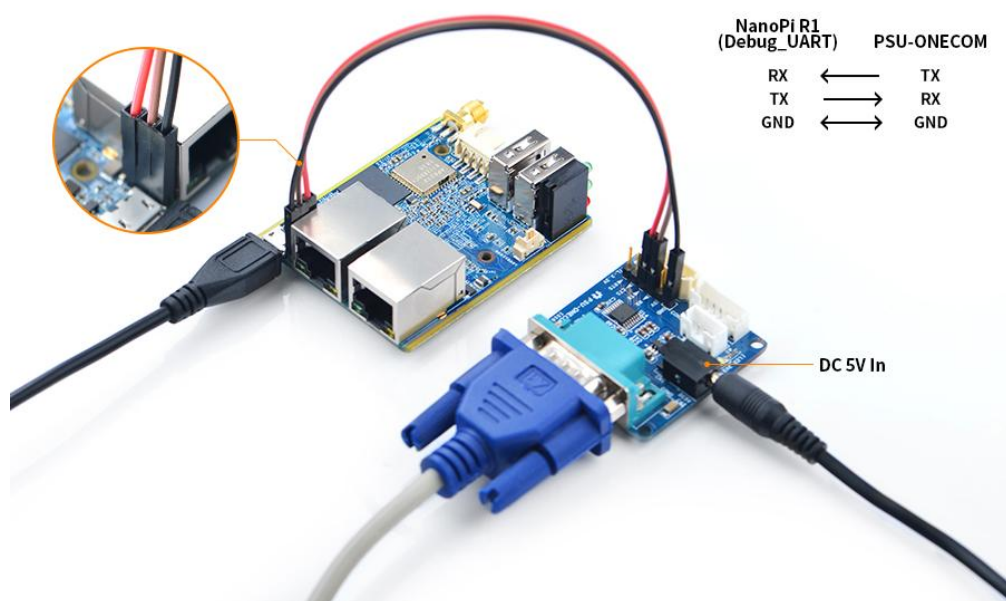
5.2 System Login

■ Login via Serial Port

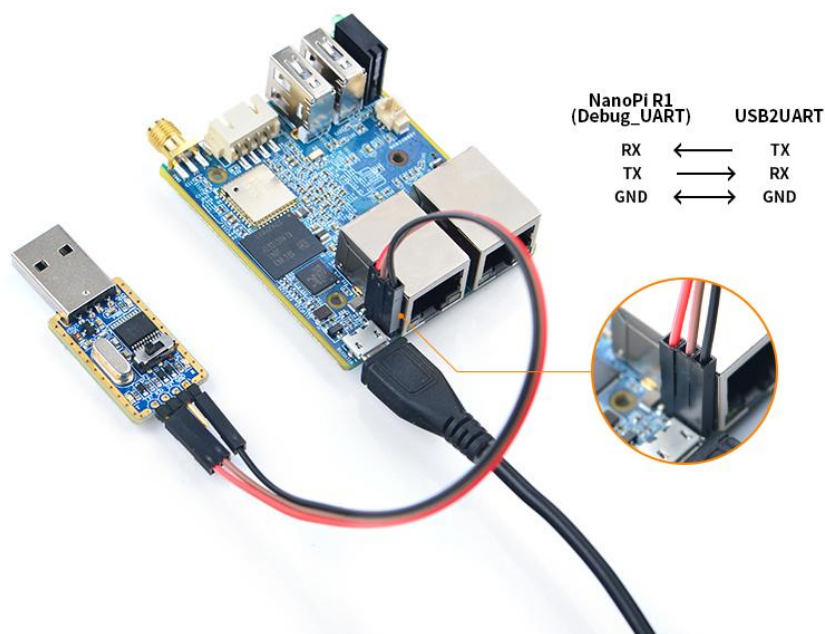
When you do kernel development you'd better get a serial communication board. After you connect your board to a serial communication board you will be able to do development work from a commandline utility.

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:

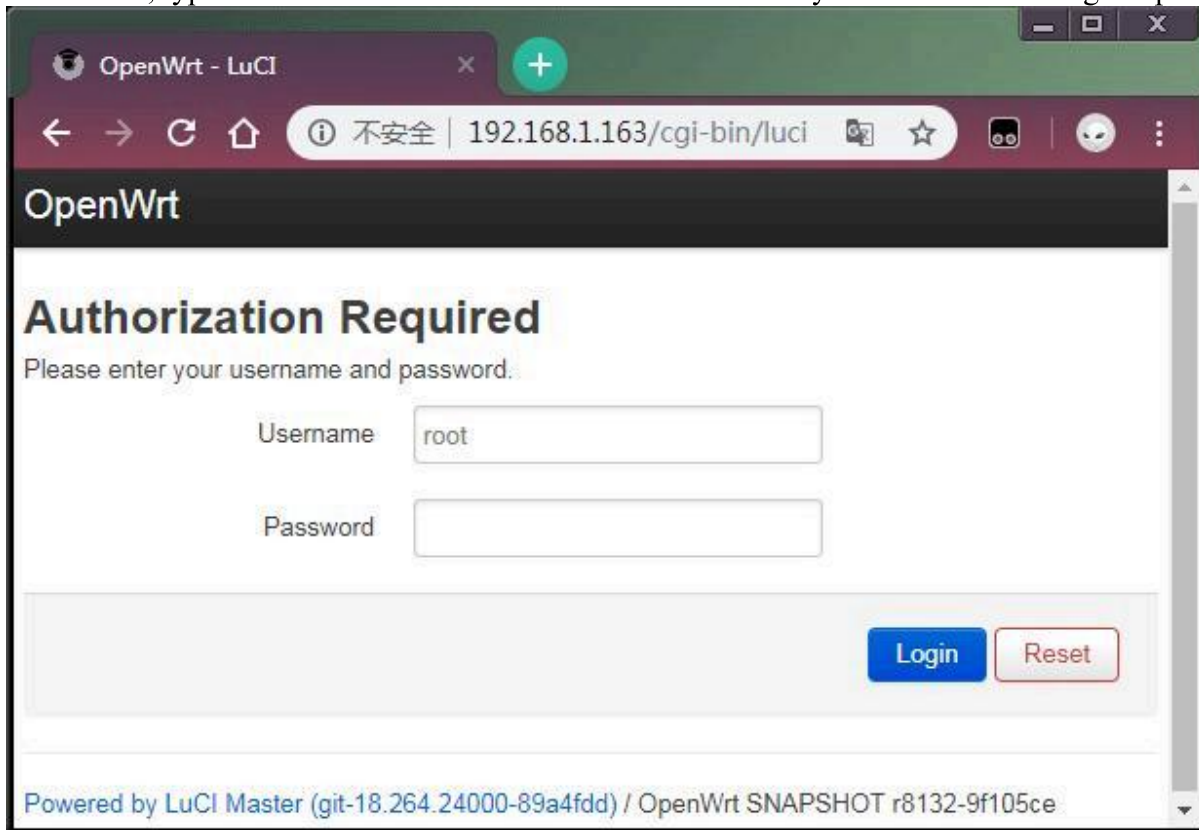
```
$ ssh root@192.168.1.163
```

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.

5.3 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|regex>    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|regex>    List package providing <file>
  find <regex>           List packages whose name or description matches <regex>
  info [pkg|regex]       Display all info for <pkg>
  status [pkg|regex]     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 \(https://openwrt.org/docs/guide-user/additional-software/opkg\)](https://openwrt.org/docs/guide-user/additional-software/opkg)

5.4 Check System Status

- Check CPU Temperature & Frequency via Commandline

```
$ 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
```

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:

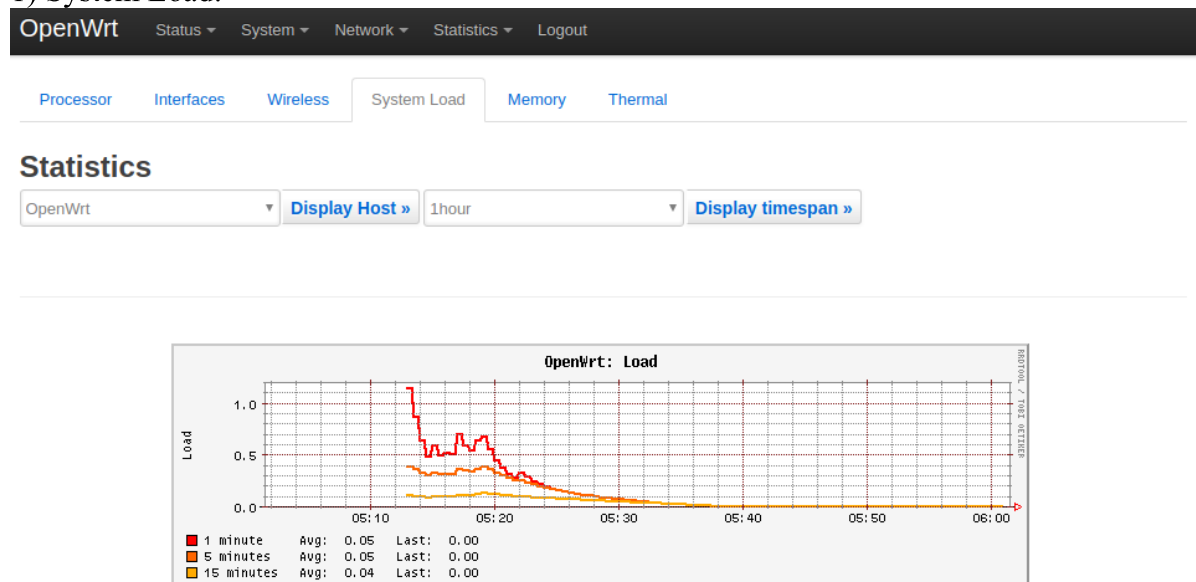
```
$ cpu_freq -s 1008000
Available frequency(KHz):
 480000 624000 816000 1008000
Current frequency(KHz):
CPU0 online=1 temp=36702C governor=userspace freq=1008000KHz
CPU1 online=1 temp=36702C governor=userspace freq=1008000KHz
CPU2 online=1 temp=36702C governor=userspace freq=1008000KHz
CPU3 online=1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

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.

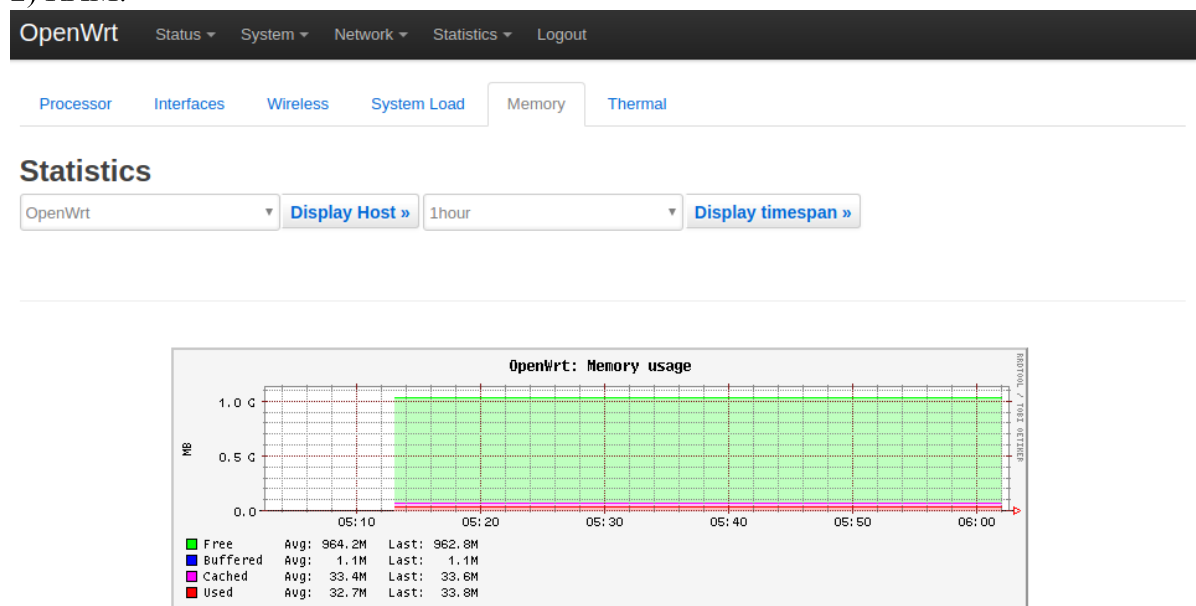
■ Check System Status on OpenWrt-LuCI Web Page

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

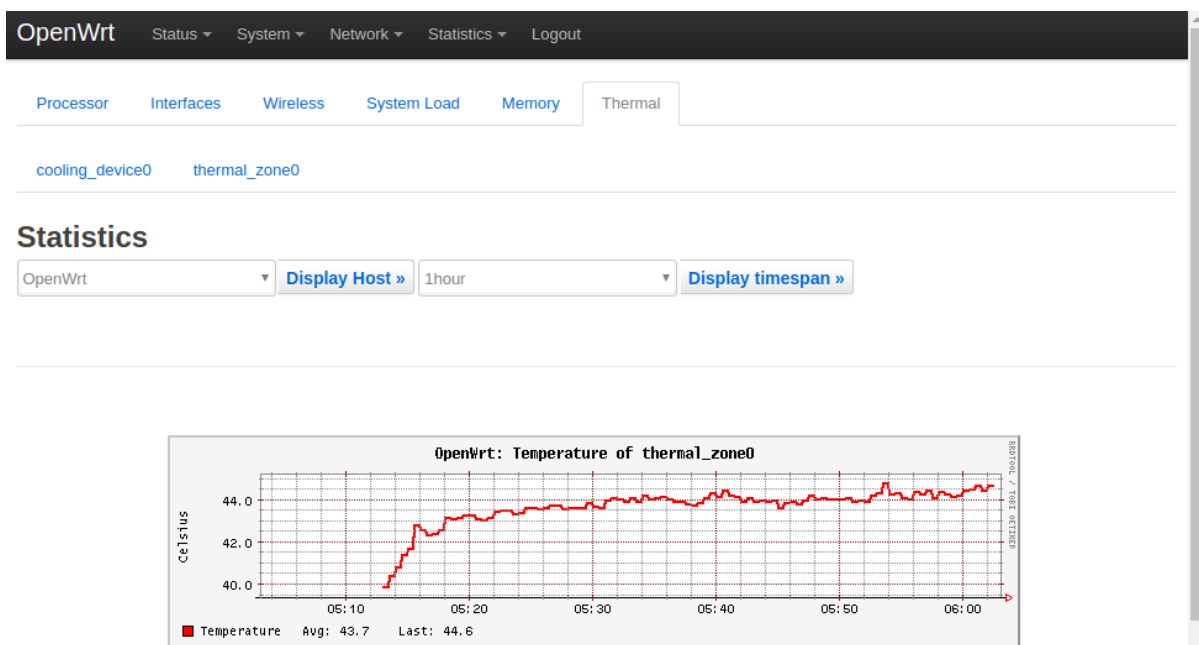
1) System Load:



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)

5.5 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.

5.6 Check Network->Wireless Configurations

- After open the OpenWrt-LuCI page, go to Network ---> Wireless and you will see the WiFi hotspot's configurations:

OpenWrt Status ▾ System ▾ Network ▾ Statistics ▾ Logout AUTO REFRESH ON

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...](#)

radio0: Master "OpenWrt-cc:b8:a8:b6:59:7c"

Wireless Overview

radio0	Generic MAC80211 802.11bgn Channel: 1 (2.412 GHz) Bitrate: 54 Mbit/s	Restart	Scan	Add
100%	SSID: OpenWrt-cc:b8:a8:b6:59:7c Mode: Master BSSID: CC:B8:A8:B6:59:7C Encryption: None	Disable	Edit	Remove

Associated Stations

Network	MAC-Address	Host	Signal / Noise	RX Rate / TX Rate
Master "OpenWrt-cc:b8:a8:b6:59:7c" (wlan0)	00:26:F7:B1:57:46	wvd lan (192.168.2.224)	-35 / 0 dBm	54.0 Mbit/s, 20MHz 54.0 Mbit/s, 20MHz

A default WiFi AP's hotspot name looks like "OpenWrt-10:d0:7a:de:3d:92". It doesn't have a password. You can connect your smart phone to it and browse the internet.

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

5.7 Check LED Configurations

- After open the OpenWrt-LuCI page, go to System ---> LED Configuration and you will see the LED's configurations:

OpenWrt
Status
System
Network
Logout

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...

LED Configuration

Customizes the behaviour of the device LEDs if possible.

Name	led1-sys	Delete
LED Name	LED1	
Default state	<input type="checkbox"/>	
Trigger	heartbeat	
Name	led2-wan	Delete
LED Name	LED2	
Default state	<input type="checkbox"/>	
Trigger	netdev	
Device	eth0	
Trigger Mode	<input checked="" type="checkbox"/> Link On <input type="checkbox"/> Transmit <input type="checkbox"/> Receive	
Name	led3-lan	Delete
LED Name	LED3	
Default state	<input type="checkbox"/>	

- By default the LED is configured as follows:

LED1: heart-beat LED. If this LED doesn't blink it means the system is dead and it needs to be restarted.
LED2: status LED for Gbps Ethernet eth0 WAN. If WAN works this LED will be solid on otherwise it will be off. You can check the "Transmit" / "Receive" box to set the LED to blink when WAN transmits / receives data.
LED3: status LED for Fast Ethernet eth1 LAN. If LAN works this LED will be solid on otherwise it will be off. You can check the "Transmit" / "Receive" box to set the LED to blink when LAN transmits / receives data.

- Reference Links:
 - openwrt led_configuration (https://openwrt.org/docs/guide-user/base-system/led_configuration)

5.8 Check BUTTON Configurations

FriendlyElec's OpenWrt system uses the triggerhappy utility to configure BUTTON's functions. By default when BUTTON is pressed a reboot will be triggered. If the system needs to be rebooted we suggest you use BUTTON to trigger a reboot. This prevents the file system from being damaged by accidental system shutdown.

The configurations for the triggerhappy utility are stored in the "/etc/triggerhappy/triggers.d/example.conf" file.

- Reference Links:
 - openwrt hardware.button (<https://openwrt.org/docs/guide-user/hardware/hardware.button>)

5.9 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";

5.10 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.

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.

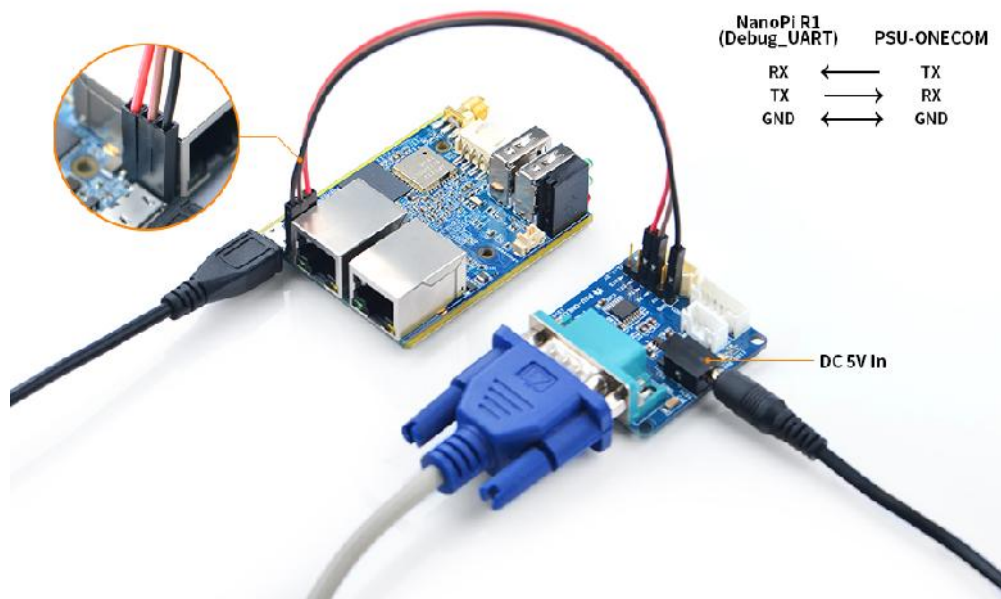
Besides the regular Ubuntu Core's features FriendlyCore has the following additional features:

- it integrates Qt4.8;
- it integrates NetworkManager;
- it has bluez and Bluetooth related packages;
- it has alsa packages;
- it has npci-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

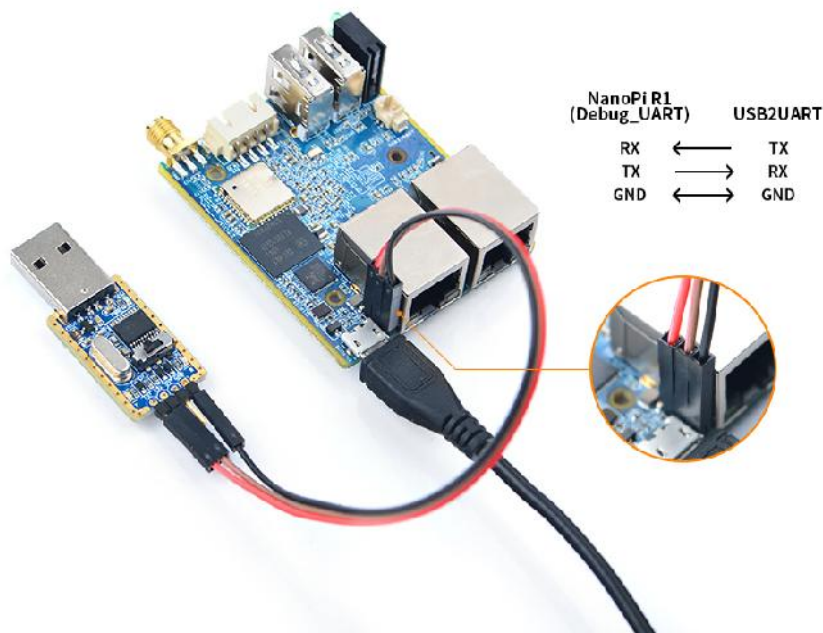
- If your board is connected to an HDMI monitor you need to use a USB mouse and keyboard.
- If you want to do kernel development you need to use a serial communication board, ie a PSU-ONECOM board, which will

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 npci-config" to disable auto login.

- Update packages

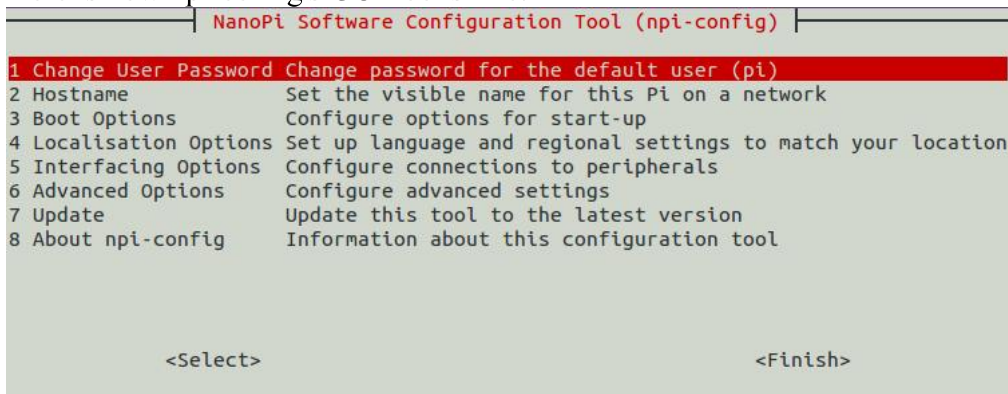
```
$ sudo apt-get update
```

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 Transfer files using Bluetooth

Take the example of transferring files to the mobile phone. First, set your mobile phone Bluetooth to detectable status, then execute the following command to start Bluetooth search. :

```
hcitool scan
```

Search results look like :

```
Scanning ...  
2C:8A:72:1D:46:02   HTC6525LVW
```

This means that a mobile phone named HTC6525LVW is searched. We write down the MAC address in front of the phone name, and then use the sdptool command to view the Bluetooth service supported by the phone :

```
sdptool browser 2C:8A:72:1D:46:02
```

Note: Please replace the MAC address in the above command with the actual Bluetooth MAC address of the mobile phone.

This command will detail the protocols supported by Bluetooth for mobile phones. What we need to care about is a file transfer service called OBEX Object Push. Take the HTC6525LVW mobile phone as an example. The results are as follows :

```
Service Name: OBEX Object Push  
Service RecHandle: 0x1000b  
Service Class ID List:  
  "OBEX Object Push" (0x1105)  
Protocol Descriptor List:  
  "L2CAP" (0x0100)  
  "RFCOMM" (0x0003)  
    Channel: 12  
  "OBEX" (0x0008)  
Profile Descriptor List:  
  "OBEX Object Push" (0x1105)  
    Version: 0x0100
```

As can be seen from the above information, the channel used by the OBEX Object Push service of this mobile phone is 12, we need to pass it to the obexftp command, and finally the command to initiate the file transfer request is as follows :

```
obexftp --nopath --noconn --uuid none --bluetooth -b 2C:8A:72:1D:46:02 -B 12 -put example.jpg
```

Note: Please replace the MAC address, channel and file name in the above command with the actual one.

After executing the above commands, please pay attention to the screen of the mobile phone. The mobile phone will pop up a prompt for pairing and receiving files. After confirming, the file transfer will start.

Bluetooth FAQ:

1) Bluetooth device not found on the development board, try to open Bluetooth with the following command :

```
rfkill unblock 0
```

2) Prompt can not find the relevant command, you can try to install related software with the following command:

```
apt-get install bluetooth bluez obexftp openobex-apps python-gobject ussp-push
```

6.8 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.

6.9 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.

6.10 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 example, change to Shanghai timezone:

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

6.12 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-lc1b000.usb-1
  ...
```

The above messages indicate that "/dev/video0" is camera's device node. The mjpg-streamer application is an open source video stream 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
```

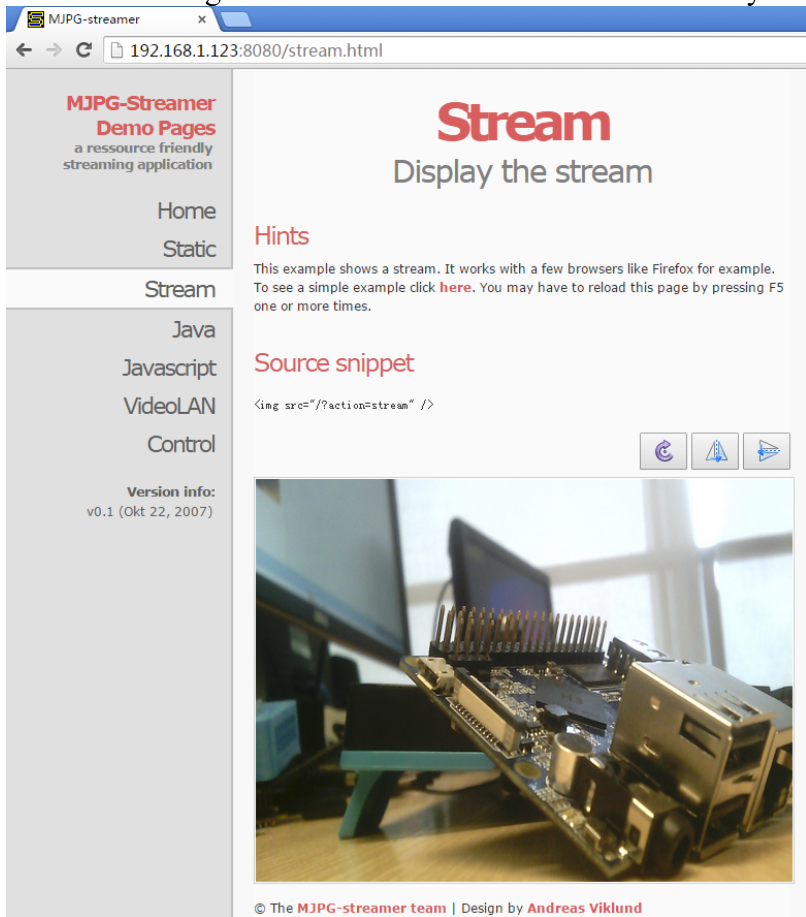
start.sh runs the following two commands:

```
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 dyncntrl 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:



6.13 Check CPU's Working Temperature

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

```
$ 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
```

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
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
```

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

6.14.1 How to Install Docker

Run the following commands:

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

6.14.2 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
```

6.15 Using RTC

NanoPi-R1提供了RTC电池接口，使用系统的RTC功能需要连接CR2032带线RTC纽扣电池给板子的RTC电路供电，电池连接如下图所示：

File:NanoPi R1-RTC-BAT-en.png

Dimensional diagram of on board RTC battery connector

(<http://wiki.friendlyelec.com/wiki/images/c/c5/1.25mm-2A.pdf>)

- ■ ■ 待完善****

7 Build Kernel Headers Package

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

7.1 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
```

7.2 Install the required packages

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

7.3 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)
```

8 Installation=

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

8.1 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
```

9 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 (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)

10 More OS Support

10.1 DietPi



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>

11 Resources

11.1 Schematics and Datasheets

- Schematics
 - NanoPi_R1_V1.0_1809-Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/a/ab/NanoPi_R1_V1.0_1809-Schematic.pdf)
- Dimensional Diagram
 - NanoPi_R1_V1.0_1809 pcb的dxf文件 (http://wiki.friendlyelec.com/wiki/index.php/File:Dimension_NanoPi_R1_V1.0_1809_PCB.rar)
- H3 datasheet Allwinner_H3_Datasheet_V1.2.pdf (http://wiki.friendlyelec.com/wiki/images/4/4b/Allwinner_H3_Datasheet_V1.2.pdf)

12 Update Log

12.1 Jan-02-2019

- Released English Version

12.2 Jan-17-2019

- Updated section 5

12.3 April-24-2019

- Updated section 5.6

Retrieved from "https://wiki.friendlyelec.com/wiki/index.php?title=NanoPi_R1&oldid=22854"

Category: Pages with broken file links

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