NanoPi R6C

From FriendlyELEC WiKi

查看中文

Contents

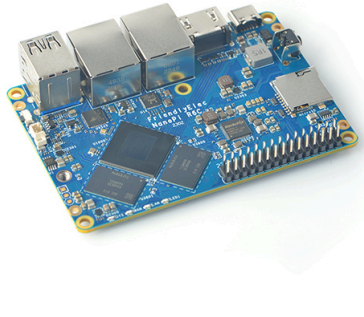
1 Introduction

The NanoPi R6C (as “R6C”) is an open-sourced mini IoT gateway device with one 2.5G and one Gbps Ethernet ports, designed and developed by FriendlyElec.It is integrated with a Rockchip RK3588S CPU, 4GB/8GB LPDDR4x RAM, and 32GB eMMC flash or none. It supports booting with TF cards and works with operating systems such as FriendlyWrt, Android, Debian, Ubuntu etc and works with headless systems as well.

The NanoPi R6C has rich hardware resources with a compact size of 90 x 62 mm, a 30-pin header is provided to expose hardware resources for secondary development, including interfaces such as GPIO, I2C, I2S, PWM and SPI, It alsa has one HDMI port. It supports decoding 8K@60fps H.265/VP9 and 8K@30fps H.264 formatted videos.

The NanoPi R6C has one M.2 NVME port and two USB ports, and supports USB type-C power delivery. It is an ideal portable drive for saving images and videos. FriendlyElec has released a carefully-designed custom CNC housing for it.

All in all, the NanoPi R6C is a board featured with multiple Ethernet ports, light NAS and video playing. It is a cannot-miss platform with infinite possibilities for geeks, fans and developers.

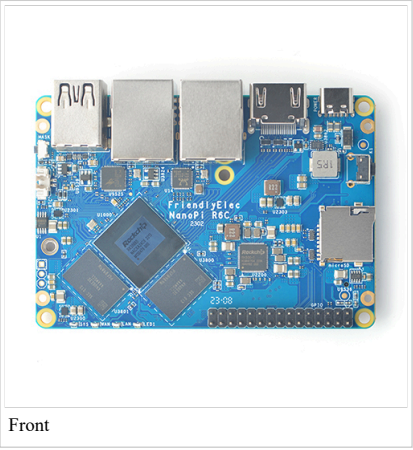


Overview

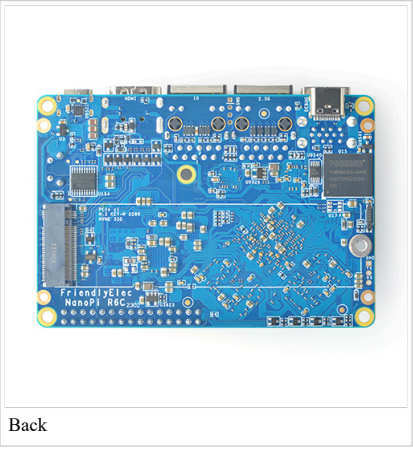
2 Hardware Spec

- SoC: Rockchip RK3588S
    - CPU: Quad-core ARM Cortex-A76(up to 2.4GHz) and quad-core Cortex-A55 CPU (up to 1.8GHz)
    - GPU: Mali-G610 MP4, compatible with OpenGLES 1.1, 2.0, and 3.2, OpenCL up to 2.2 and Vulkan1.2
    - VPU: 8K@60fps H.265 and VP9 decoder, 8K@30fps H.264 decoder, 4K@60fps AV1 decoder, 8K@30fps H.264 and H.265 encoder
    - NPU: 6TOPs, supports INT4/INT8/INT16/FP16
- RAM: 64-bit 4GB/8GB LPDDR4X at 2133MHz
- Flash: 32GB/None eMMC, at HS400 mode

- Ethernet: one Native Gigabit Ethernet, and one PCIe 2.5G Ethernet
- USB: one USB 3.0 Type-A and one USB 2.0 Type-A
- PCIe: one M.2 Key M connector with PCIe 2.1 x1
- HDMI:
  - compatible with HDMI2.1, HDMI2.0, and HDMI1.4 operation
  - support up to 7680x4320@60Hz
  - Support RGB/YUV(up to 10bit) format
- microSD: support up to SDR104 mode
- GPIO:
  - 30-pin 2.54mm header connector
  - up to 1x SPI, 3x UARTs, 3x I2Cs, 2x SPDIFs, 1x I2Ss, 3x PWMs, 20x GPIOs
- Debug: UART via 3-Pin 2.54mm header, or on-board USB-C to UART
- LEDs: 4 x GPIO Controlled LED (SYS, WAN, LAN, LED1)
- others:
  - 2 Pin 1.27/1.25mm RTC battery input connector for low power RTC IC HYM8563TS
  - MASK button for eMMC update
  - one user button
- Power supply: USB-C, support PD, 5V/9V/12V/20V input
- PCB: 8 Layer, 62x90x1.6mm
- Ambient Operating Temperature: 0°C to 70°C



Front



Back

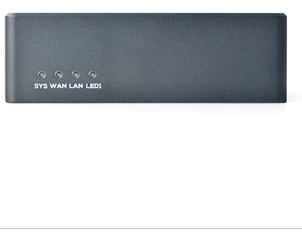
3 Diagram, Layout and Dimension

3.1 Layout

- **30-pin GPIO**

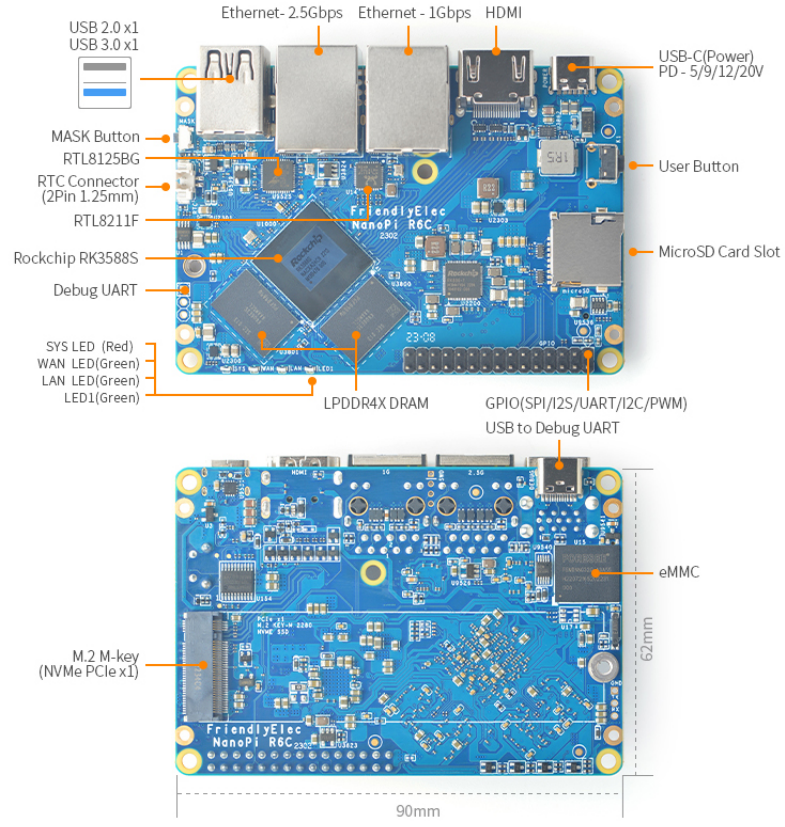
Case

Case



Case

Case



NanoPi R6C Layout

Pin#	Assignment	Pin#	Assignment
1	VCC_3V3_S3*	2	VCC_5V0*
3	I2C8_SDA_M2	4	VCC_5V0*
5	I2C8_SCL_M2	6	GND
7	GPIO4_A0/I2S1_MCLK_M0	8	GPIO3_C4/UART5_TX_M1
9	GND	10	GPIO3_C5/UART5_RX_M1
11	GPIO4_A1/I2S1_SCLK_M0	12	GPIO0_D4/PWM3_IR_M0
13	GPIO4_A2/I2S1_LRCK_M0	14	GND
15	GPIO4_A5/I2S1_SDI0_M0	16	GPIO4_A6/I2S1_SDI1_M0
17	VCC_3V3_S3*	18	GPIO4_B1/I2S1_SDO0_M0
19	GPIO1_B2/SPI0_MOSI_M2/UART4_RX_M2	20	GND
21	GPIO1_B1/SPI0_MISO_M2	22	GPIO4_B2/I2S1_SDO1_M0
23	GPIO1_B3/SPI0_CLK_M2/UART4_TX_M2	24	GPIO1_B4/SPI0_CS0_M2
25	GND	26	GPIO1_A2/PWM0_M2/I2C4_SDA_M3
27	GPIO1_B6/UART1_TX_M1/I2C5_SCL_M3/SPDIF0_TX_M0	28	GPIO1_B7/UART1_RX_M1/I2C5_SDA_M3/SPDIF1_TX_M0
29	GPIO1_A3/PWM1_M2/I2C4_SCL_M3	30	GND

The logic level of all signals is 3.3V.
VCC_3V3_S3 is 3.3V output with a maximum output of 500mA.
VCC_5V0 is 5V output with a maximum output of 1A.

- **Debug UART Pin Spec**

  3.3V level signals, 1500000bps

Pin#	Assignment	Description
1	GND	0V
2	UART2_TX_M0_DEBUG	output
3	UART2_RX_M0_DEBUG	intput

- **USB-C Debug Port**

  This is a usb to serial port, connected to the Debug UART. Just plug and use.
  Windows driver: http://www.wch-ic.com/downloads/CH341SER_ZIP.html

- **USB-A Port**

  Each USB Type-A port has 1.4A overcurrent protection.

- **RTC**

  RTC backup current is 0.25μA TYP (VDD =3.0V, TA =25℃).
  Connector P/N: Molex 53398-0271

- **User Button**

  User Button is connected to GPIO1_C0, and is active-low.

4 Get Started

4.1 Essentials You Need

Before starting to use your NanoPi-R6C get the following items ready

- NanoPi-R6C
- MicroSD Card/TF Card: Class 10 or Above, minimum 8GB SDHC
- USB C PD Charger (10W & above)
- If you need to develop and compile,you need a computer that can connect to the Internet. It is recommended to install Ubuntu 20.04 64-bit system and use the following script to initialize the development environment, or use docker container:
  - How to setup the Compiling Environment on Ubuntu bionic (https://github.com/friendlyarm/build-env-on-ubuntu-bionic)
  - docker-cross-compiler-novnc (https://github.com/friendlyarm/docker-cross-compiler-novnc)

4.2 TF Cards We Tested

Refer to： TFCardsWeTested

4.3 PD Power Adapters We Tested

Refer to： PD Power Adapters We Tested

4.4 Configure parameters for serial port

The {{{1}}} has a built-in USB to TTL chip, you may use a Type-C cable to connect to the USB port of your PC. It works plug-and-play on Linux, but requires driver installation on Windows and Mac systems. The download links for drivers are as follows:
Windows driver download link: Matrix - CH341SER.ZIP (https://wiki.friendlyelec.com/wiki/index.php/File:CH341SER.ZIP)

Mac driver download link: - CH341SER_MAC_ZIP (http://www.wch-ic.com/downloads/CH341SER_MAC_ZIP.html)

Serial port parameter:

Baud rate	1500000
Data bit	8
Parity check	None
Stop bit	1
Flow control	None

4.5 Install OS

4.5.1 Downloads

4.5.1.1 Official image

Visit download link (http://download.friendlyelec.com/NanoPiR6C) to download official image files (in the "01_Official images" directory).
The table below lists all official images, the word 'XYZ' in image filename meaning:

- **sd**: Use it when you need to boot the entire OS from the SD card
- **eflasher**: Use it when you need to flash the OS to eMMC via TF card
- **usb**: Use it when you need to flash the OS to eMMC via USB

Icon	Image Filename	Version	Description	Kernel Version
	rk3588-XYZ-debian-bullseye-core-6.1-arm64-YYYYMMDD.img.gz	bullseye	Debian 11 Core, command line only	6.1.y
	rk3588-XYZ-debian-bullseye-minimal-6.1-arm64-YYYYMMDD.img.gz	bullseye	Debian 11 Desktop, Xfce desktop, no pre-installed recommended software, supports HW acceleration	6.1.y
	rk3588-XYZ-debian-bullseye-desktop-6.1-arm64-YYYYMMDD.img.gz	bullseye	Debian 11 Desktop, Xfce desktop, pre-installed mpv, smplayer and chromium brower, supports HW acceleration	6.1.y
	rk3588-XYZ-ubuntu-focal-desktop-6.1-arm64-YYYYMMDD.img.gz	focal	Ubuntu 20.04 Desktop, LXQT desktop, pre-installed mpv, smplayer and chromium brower, supports HW acceleration	6.1.y
	rk3588-XYZ-ubuntu-jammy-desktop-6.1-arm64-YYYYMMDD.img.gz	jammy	Ubuntu 22.04 with GNOME and Wayland with recommended software	6.1.y
	rk3588-XYZ-ubuntu-jammy-minimal-6.1-arm64-YYYYMMDD.img.gz	jammy	Lightweight Ubuntu 22.04 with GNOME and Wayland	6.1.y
	rk3588-XYZ-ubuntu-jammy-x11-desktop-arm64-YYYYMMDD.img.gz	jammy	Ubuntu 22.04 with Xubuntu and X11, use Panfrost GPU driver	6.1.y
	rk3588-XYZ-friendlycore-focal-6.1-arm64-YYYYMMDD.img.gz	focal	FriendlyCore， command line only, pre-installed Qt5， based on Ubuntu core 20.04	6.1.y
	rk3588-XYZ-androidtv-YYYYMMDD.img.zip	12	Android 12 TV	6.1.y
	rk3588-XYZ-android12-YYYYMMDD.img.zip	12	Android 12 Tablet	6.1.y
	rk3588-XYZ-openmediavault-6.1-YYYYMMDD.img.gz	Shaitan	OpenMediaVault NAS system, base on Debian 11, already built-in Docker.	6.1.y
	rk3588-XYZ-friendlywrt-21.02-YYYYMMDD.img.gz	21.02	FriendlyWrt, based on OpenWrt 21.02	6.1.y
	rk3588-XYZ-friendlywrt-21.02-docker-YYYYMMDD.img.gz	21.02	FriendlyWrt with Docker, based on OpenWrt 21.02	6.1.y
	rk3588-XYZ-friendlywrt-23.05-YYYYMMDD.img.gz	23.05	FriendlyWrt, based on OpenWrt 23.05	6.1.y
	rk3588-XYZ-friendlywrt-23.05-docker-YYYYMMDD.img.gz	23.05	FriendlyWrt with Docker, based on OpenWrt 23.05	6.1.y
Other Image				
	Github Actions - FriendlyWrt	21.02,23.05	FriendlyWrt (https://github.com/friendlyarm/Actions-FriendlyWrt/releases)	6.1.y
	rk3588-eflasher-multiple-os-YYYYMMDD-25g.img.gz	-	It contains multiple OS image files, making it convenient for testing different operating systems	

4.5.1.2 Tools (optional)

Visit download link (http://download.friendlyelec.com/NanoPiR6C) to download tools (in the "05_Tools" directory).

Filename	Description
win32diskimager.rar	This program is designed to write a raw disk image to a removable device or backup a removable device to a raw image file
SD Card Formatter	A program (application) that allows easy and quick clear the SD card
RKDevTool_Release_v2.84.zip	Rockchip flashing tool, for USB upgrade

4.5.2 Flashing the OS to the microSD card

Follow the steps below:

- Get an 8G microSD card;
- Visit download link (http://download.friendlyelec.com/NanoPiR6C)to download image files (in the "01_Official images/01_SD card images" directory);
- Download the win32diskimager tool (in the "05_Tools" directory), or use your preferred tool;
- Extract the .gz format compressed file to get the .img format image file;
- Run the win32diskimager utility under Windows as administrator. On the utility's main window select your SD card's drive, the wanted image file and click on "write" to start flashing the SD card.
- Take out the SD and insert it to NanoPi-R6C's microSD card slot;
- Power on NanoPi-R6C and it will be booted from your TF card, some models may require pressing the Power button to start;

4.5.3 Install OS to eMMC

4.5.3.1 Option 1: Install OS via TF Card

This method firstly boots a mini Linux from a TF card and then automatically runs an EFlasher utility to install the OS to eMMC. You can connect your system to an HDMI monitor and watch its progress.
This is optional. You can watch its progress by observing its LEDs as well:

Progress	SYS LED(Red)	LAN LED(Green)	WAN LED(Green)
Power On	Solid On	Off	Off
System Boot	Slow Flashing	Off	Off
Installation in Progress	Fast Flashing	Off	Off
Installation Done	Slow Flashing	Solid On	Solid On

By default, flashing starts automatically upon power-up, so be sure to back up the data in eMMC. If you don't want it to start automatically, you can use image file with a filename containing the words 'multiple-os' and manually select the OS you want to flash on the interface.

4.5.3.1.1 Flash Official OS to eMMC

Follow the steps below:

- Get an SDHC card with a minimum capacity of 8G
- Visit download link (http://download.friendlyelec.com/NanoPiR6C)to download image files (in the "01_Official images/02_SD-to-eMMC images" directory) and win32diskimager tool (in the "05_Tools" directory);
- Extract the .gz format compressed file to get the .img format image file;
- Run the win32diskimager utility under Windows as administrator. On the utility's main window select your SD card's drive, the wanted image file and click on "write" to start flashing the SD card.
- Eject your SD card and insert it to NanoPi-R6C's microSD card slot.
- Turn on NanoPi-R6C, it will boot from the SD card and automatically run EFlasher to install the OS to the board's eMMC.
- After flashing is complete, eject the SD card from NanoPi-R6C, NanoPi-R6C will automatically reboot and boot from eMMC.

4.5.3.1.2 Flash third party OS (Image file) to eMMC

- Auto Install (Default Behavior)

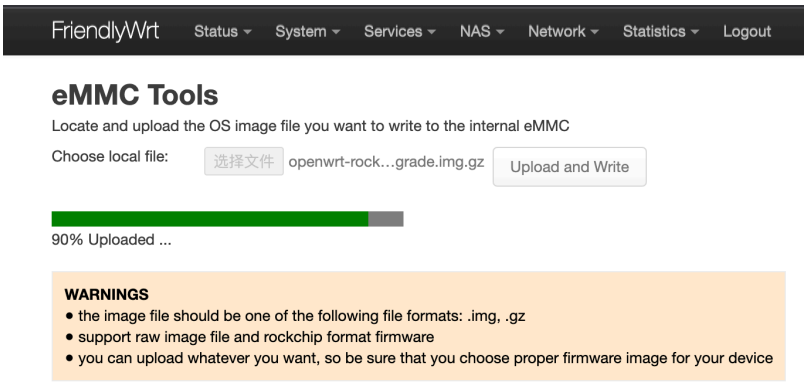
1) Download an "eflasher" firmware from network drive (http://download.friendlyelec.com/NanoPiR6C)(in the "01_Official images/02_SD-to-eMMC images" directory), extract it and install it to a TF card ;
2) Eject and insert the TF card to your PC, after a "FriendlyARM" device shows up(Under Linux, it is a "FriendlyARM" directory), copy an .img or .gz file to the TF card.
3) Open the eflasher.conf file on the TF card, set "autoStart=" to the name of your image file, such as:

```
autoStart=openwrt-rockchip-armv8_nanopi-ext4-sysupgrade.img.gz
```

In addition to third-party image, official image files which with the '-sd-' word in the filename are also supported, for example: rk3NNN-sd-friendlywrt-21.02-YYYYMMDD.img.gz
4) Eject the TF card, insert the TF card to NanoPi-R6C, power it on it will automatically install your firmware. You can watch the installation progress by observing the LEDs' status.

4.5.3.2 Option 2: Install OS on Web Page

Get a TF card which has been installed with FriendlyWrt, log in FriendlyWrt on the web page, click on "System" ->"eMMC Tools". Click on "Select file" to select your wanted image file, either an official image (filename containing '-sd-') or a third party image. The file should be a ".gz" or ".img" file.
After a file is selected, click on "Upload and Write" to start installing an OS.



After installation is done, eject the SD card, the system will automatically reboot and load the OS from eMMC. After the OS begins to load, if the system LED is flashing and the network LED is on, it means the the OS has loaded successfully. If the OS is FriendlyWrt, you can click on “Go to Homepage” to enter the homepage. For official OS, you need select the file with the filename containing '-sd-', for example: rk3NNN-sd-friendlywrt-21.02-YYYYMMDD.img.gz, the compression file only supports the .gz format. If the file is too large, you can compress it into .gz format before uploading.

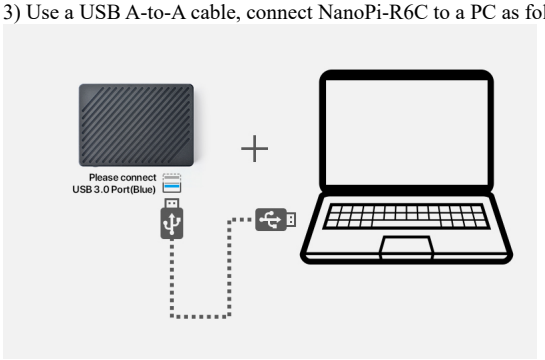
4.5.3.3 Option 3: Install OS via USB

4.5.3.3.1 Step 1: Install USB Driver and Tools/Utilities

Download a driver file DriverAssitant_v5.1.1.zip under the “tools” directory from network drive (http://download.friendlyelec.com/NanoPiR6C), extract and install it. Under the same directory, download a utility RKDevTool_Release_v2.84.zip and extract it.

4.5.3.3.2 Step 2: Connect NanoPi-R6C to PC and Enter Installation Mode

1) Disconnect the connected USB cable and the power cord from NanoPi-R6C, eject the TF card;
2) Press and hold the “Mask” key, power on the board. After the status LED has been on for at least 3 seconds, release the Mask key.
3) Use a USB A-to-A cable, connect NanoPi-R6C to a PC as follows:



4.5.3.3.3 Step 3: Install image to eMMC

A firmware in general is packaged in either of the two options: the first is an whole image (ie, update.img) which is often offered by third party developers, the second is that an image is packaged and placed in multiple partition images. FriendlyElec offers an image in the latter option.

- Option 1: Install whole image (ie, update.img)

On a PC which has the extracted RKDevTool_Release_v2.84 utility, go to the RKDevTool_Release_v2.84 directory, run the RKDevTool.exe file. If everything works, you will see a “Found a new Maskrom device” message on the utility;
Go to “Upgrade Firmware(升级固件)”, click on “Firmware(固件)”, select your wanted image file, and click on “Upgrade(升级)” to install. After installation is done, your board will reboot automatically and load the system from eMMC;

- Option 2: Install OS that is packaged & placed in multiple partition images

Go to network drive (http://download.friendlyelec.com/NanoPiR6C) to download your needed package and extract it (in the "01_Official images/03_USB upgrade images). After it is extracted, you will see some utilities and a configuration file under the directory. double click on RKDevTool.exe, you will see a “Found a new Maskrom device” message on the utility. Click on the “Execute”, wait a moment and it will be installed. After installation is done your system will automatically reboot and load the system from eMMC.

4.6 The Boot order between eMMC and SD card

By default, the system will be booted from the TF card first, but this is not the case under all conditions. This section will explain all situations in detail;

Refer to rockchip official document [1] (http://opensource.rock-chips.com/wiki_Boot_option), there are two types of loader program:
1) U-Boot TPL/SPL (i.e. upsream U-Boot, also called mainline U-Boot)
2) Rockchip MiniLoader

Things to note:
1) FriendlyELEC's image uses Rockchip MiniLoader
2) The third-party image usually uses U-Boot TPL/SPL

The following situations will always start from eMMC:
If the system in the eMMC, or the system in the TF card uses the first Loader type U-Boot TPL/SPL, it will always boot from the eMMC;

If you want to boot from the TF card, there are the following methods:
Method 1: Clear the Loader on the eMMC, the clearing method is as follows, after starting from the eMMC, enter the following command on the command line to clear the Loader on the eMMC:

```
dd if=/dev/zero of=/dev/mmcblk2 bs=8M count=1
```

Method 2: Insert the TF card, Press Maskrom Key (or short-circuit the Maskrom contacts) and then power on (need to keep the short-circuit for about 3 seconds), it will start from the TF card

The summary is as follows:

eMMC current system	TF card current system	Boot priority
No system	Any image	TF card
FriendlyELEC's image	FriendlyELEC's image	TF card
FriendlyELEC's image	Image with Mainline U-boot	eMMC
Image with Mainline U-boot	FriendlyELEC's image	eMMC
Image with Mainline U-boot	Image with Mainline U-boot	eMMC

5 Work with FriendlyWrt

5.1 Introduction to FriendlyWrt

FriendlyWrt is a customized system made by FriendlyElec based on an OpenWrt distribution. It is open source and well suitable for developing IoT applications, NAS applications etc.

5.2 First boot

For the first boot, the system needs to do the following initialization work：
1）Extended root file system
2）Initial setup（will execute /root/setup.sh)
So you need to wait for a while (about 2~3 minutes) to boot up for the first time, and then set FriendlyWrt, you can enter the ttyd terminal on the openwrt webpage, when the prompt is displayed as root@FriendlyWrt, it means the system has been initialized.

```
root@FriendlyWrt
```

5.3 Account & Password

The default password is password (empty password in some versions). Please set or change a safer password for web login and ssh login. It is recommended to complete this setting before connecting NanoPi-R6C to the Internet.

5.4 Login FriendlyWrt

Connect the PC to the LAN port of NanoPi-R6C. If your PC without a built-in ethernet port, connect the LAN port of the wireless AP to the LAN port of NanoPi-R6C, and then connect your PC to the wireless AP via WiFi , Enter the following URL on your PC's browser to access the admin page:

- http://friendlywrt/
- http://192.168.2.1/
- http://[fd00:ab:cd::1]

The above is the LAN port address of NanoPi-R6C. The IP address of the WAN port will be dynamically obtained from your main router through DHCP.

5.5 Recommended security settings

The following settings are highly recommended to complete before connecting NanoPi-R6C to the Internet。

- Set a secure password
- Only allow access to ssh from lan, change the port
- Check the firewall settings

Set up as you wish.

5.6 Change LAN IP in LuCI

1) Click on Network → Interfaces, then click on the Edit button of the LAN Network;
2) In General Setup tab, input new IP address (for example: 192.168.11.1), click "Save" and then click "Save & Apply";
3) On the pop-up window with the title “Connectivity change“, click "Apply and revert on connectivity loss";
4) Wait a moment, enter the new address in your computer's browser and login to FriendlyWrt;

5.7 Safe shutdown operation

Enter the "Services" -> "Terminal", enter the "poweroff" command and hit enter, wait until the led light is off, and then unplug the power supply.

5.8 Soft Factory Reset

Enter "System"->"Backup/Flash firmware"， Click “Perform reset“ Button, Your device's settings will be reset to defaults like when FriendlyWrt was first installed.
You can also do this in the terminal:

```
firstboot && reboot
```

5.9 Install Software Packages

5.9.1 Set up openwrt official opkg source

```
sed -i -e 's/mirrors.cloud.tencent.com/downloads.openwrt.org/g' /etc/opkg/distfeeds.conf
opkg update
```

5.9.2 Update Package List

Before install software packages update the package list:

```
$ opkg update
```

5.9.3 List Available Packages

```
$ opkg list
```

5.9.4 List Installed Packages

```
$ opkg list-installed
```

5.9.5 Install Packages

```
$ opkg install <package names>
```

5.9.6 Remove Packages

```
$ opkg remove <package names>
```

5.10 Disable IPv6

```
. /root/setup.sh
disable_ipv6
reboot
```

5.11 Configure the function of the user button

By default, the user button is configured to reboot the device, as shown below:

```
echo 'BTN_1 1 /sbin/reboot' >> /etc/triggerhappy/triggers.d/example.conf
```

You can change its behavior by changing the configuration file above.

5.12 Configuring Quectel EC20 (4G module) dial-up networking

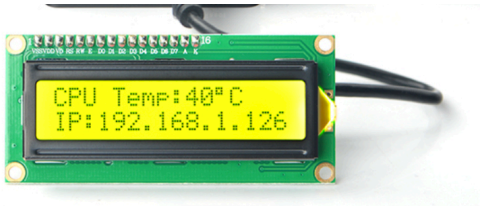
- Go to "Network" -> "Interfaces"
- Click "Delete" next to "WAN6", then click "Save & Apply"
- Click "Edit" next to "WAN", in the "Device" drop-down menu, select "Ethernet Adapter: wwan0", in the "Protocol" drop-down menu, select "QMI Cellular" and click "Switch Protocol"
- Click the "Modem Device" drop-down menu, select "/dev/cdc-wdm0", fill in the APN information (e.g. for China Mobile, enter "cmnet")
- Click "Save" to close the dialog, Finally, click "Save & Apply" at the bottom of the page to initiate the dial-up process
- Devices connected to LAN will have access to the Internet, If your device has a WiFi module, you can enable wireless AP functionality on the "Wireless" page and connect to the Internet via devices connected wirelessly

5.13 Some common issues of FriendlyWrt

- Unable to dial up
  - Go to "Network" -> "Firewall" and set "Inbound Data", "Outbound Data" and "Forwarding" in "WAN Zone" to "Accept";
  - If you still cannot access the Internet, you can try to turn off IPV6;
- Dial-up successful, but no outgoing traffic
  - Go to "Services" -> "Terminal" and type "fw4 reload" to try to reload the firewall settings again;
- Unable to power on
  - Try to replace the power adapter and cable. It is recommended to use a power supply with specifications above 5V/2A;
  - Note that some fast chargers with Type-C interface will have a delay, it may take a few seconds to start providing power;
- When doing secondary routing, the computer cannot connect to the Internet
  - If your main network is IPv4, and NanoPi-R6C works in IPv6, the computer may not be able to connect to the Internet. It is recommended to turn off IPv6 (the method is described later in this WiKi), or switch the main route to IPv6;
- If you have questions or have better suggestions, please send an email to techsupport@friendlyarm.com;

5.14 Use USB2LCD to view IP and temperature

Plug the USB2LCD module to the USB interface ofNanoPi-R6C and power on, the IP address and CPU temperature will be displayed on the LCD:



5.15 How to use USB WiFi

5.15.1 Check USB WiFi Device with Command Line Utility

(1) Click on "services>ttyd" to start the command line utility

(2) Make sure no USB devices are connected to your board and run the following command to check if any USB devices are connected or not

```
lsusb
```

(3) Connect a USB WiFi device to the board and run the command again

```
lsusb
```

You will see a new device is detected. In our test the device's ID was 0BDA:C811

(4) Type your device's ID (in our case it was "0BDA:C811" or "VID_0BDA&PID_C811") in a search engine and you may find a device that matches the ID. In our case the device we got was Realtek 8811CU.

5.15.2 Configure a USB WiFi Device as AP

(1) Connect a USB WiFi device to the NanoPi-R6C. We recommend you to use the following devices:

WiFi Chipset \ OS	Distro Support		AP Mode
	FriendlyWrt OpenWrt 19.07.5	Ubuntu Core Ubuntu 20.04 64-bit	
RTL8188CUS/8188EU 802.11n WLAN Adapter	Preinstalled driver	Yes	×
RT2070 Wireless Adapter	Preinstalled driver	Yes	×
RT2870/RT3070 Wireless Adapter	Preinstalled driver	Yes	×
RTL8192CU Wireless Adapter	Preinstalled driver	Yes	×
Ralink MT7601/MT7601U	Preinstalled driver	Yes	×
5G USB WIFI RTL8821CU/RTL8811CU (VID_0BDA & PID_C811)	Plug and play, Access Point mode by default	Yes	✔
5G USB WIFI RTL8812BU (VID_0BDA & PID_B812)	Plug and play, Access Point mode by default	Yes	✔
5G USB WiFi RTL8812AU (VID_0BDA & PID_8812)	Plug and play, Access Point mode by default	Yes	✔
5G USB WIFI MediaTek MT7662 (VID_0E8D & PID_7612)	Plug and play, Access Point mode by default	No	✔

Note: devices that match these VID&PIDs would most likely work.

(2) Click on "System>Reboot" and reboot your NanoPi-R6C

(3) Click on "Network>Wireless" to enter the WiFi configuration page

(4) Click on "Edit" to edit the configuration

(5) On the "Interface Configuration" page you can set the WiFi mode and SSID, and then go to "Wireless Security" to change the password. By default the password is "password". After you make your changes click on "Save" to save

(6) After you change the settings you can use a smartphone or PC to search for WiFi

5.15.3 Common USB WiFi issues

1) It is recommended to plug in the usb wifi in the off state, then power it on, FriendlyWrt will automatically generate the configuration file /etc/config/wireless, if not, see if there is wlan0 by ifconfig -a, if there is no wlan0, usually there is no driver.
2) If ifconfig -a sees wlan0, but the hotspot is not working properly, try changing the channel and country code, an inappropriate country code can also cause the WiFi to not work.
3) Some USB WiFis (e.g. MTK MT7662) work in CD-ROM mode by default and need to be switched by usb_modeswitch, you can try to add usb_modeswitch configuration to the following directory: /etc/usb_modeswitch.d.

5.15.4 Change the default WiFi hotspot configuration

FriendlyWrt sets the country, hotspot name and other parameters for USB WiFi by default, with the aim of being as plug-and-play as possible, but this does not guarantee that all modules will be compatible with this setting, you can change these behaviors by modifying the following file：

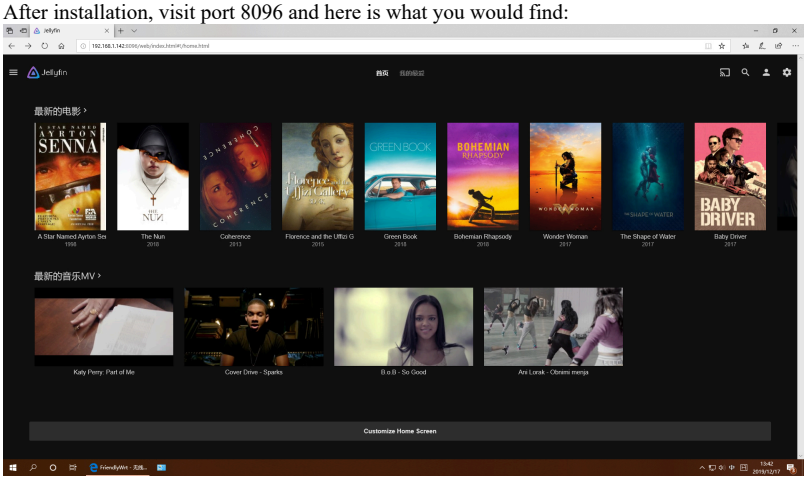
```
/lib/wifi/mac80211.sh
```

5.16 Work with Docker Applications

5.16.1 Work with Docker: Install JellyFin

```
mkdir -p /jellyfin/config
mkdir -p /jellyfin/videos
docker run --restart=always -d -p 8096:8096 -v /jellyfin/config:/config -v /jellyfin/videos:/videos jellyfin/jellyfin:10.1.0-arm64 -name myjellyfin
```

After installation, visit port 8096 and here is what you would find:



5.16.2 Work with Docker: Install Personal Nextcloud

```
mkdir /nextcloud -p
docker run -d -p 8888:80  --name nextcloud  -v /nextcloud/:/var/www/html/ --restart=always --privileged=true  arm64v8/nextcloud
```

After installtion, visit port 8888.

5.16.3 Expand Docker Storage

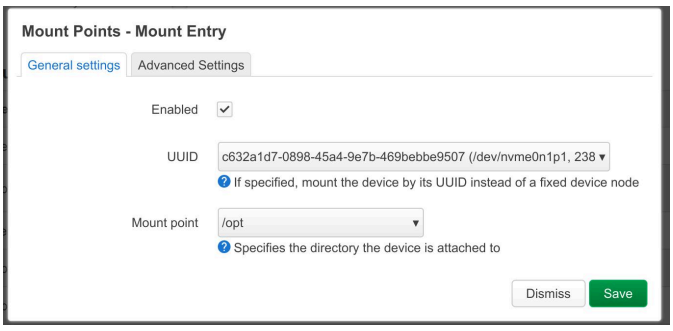
- Stop docker service first:

```
/etc/init.d/dockerd stop
```

- Rename the original /opt directory, create an empty /opt directory:

```
mv /opt /opt-old && mkdir /opt
```

- Format your drive as ext4, and mount it to the /opt directory:



- Enter the command "mount | grep /opt" to check the mount status:

```
root@FriendlyWrt:~# mount | grep /opt
/dev/nvme0n1p1 on /opt type ext4 (rw,relatime)
root@FriendlyWrt:~#
```

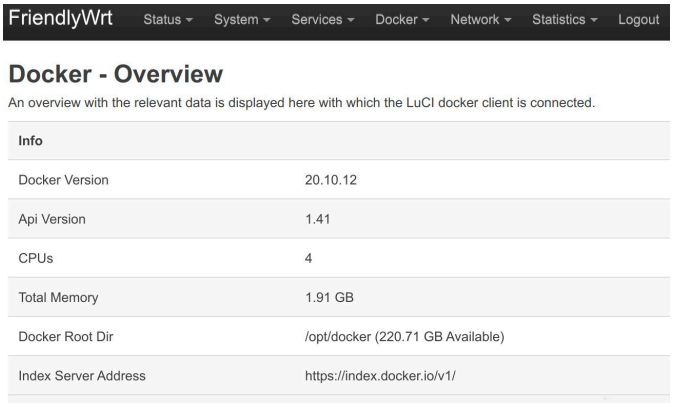
- Copy the files from the original /opt directory to the new /opt directory:

```
cp -af /opt-old/* /opt/ && rm -rf /opt-old
```

- Reboot the device

```
reboot
```

- After reboot, go to the "Docker" -> "Overview" page, check the information in the "Docker Root Dir" line, you can see that the Docker space has been expanded:



5.16.4 Docker FAQ and solutions

5.16.4.1 Unable to access the network services provided by the Docker container

Solution:

- Go to the "Firewall" settings and set "Forwarding" to "Accept";
- Turn off "Software Offload";

5.17 Mount smbfs

```
mount -t cifs //192.168.1.10/shared /movie -o username=xxx,password=yyy,file_mode=0644
```

5.18 Use sdk to compile the package

5.18.1 Install the compilation environment

Download and run the following script on 64-bit Ubuntu (version 18.04+): How to setup the Compiling Environment on Ubuntu bionic (https://github.com/friendlyarm/build-env-on-ubuntu-bionic)

5.18.2 Download and decompress sdk from the network disk

The sdk is located in the toolchain directory of the network disk:

```
tar xvf openwrt-sdk-*-rockchip-armv8_gcc-11.2.0_musl.Linux-x86_64.tar.xz
# If the path is too long, it will cause some package compilation errors, so change the directory name here
mv openwrt-sdk-*-rockchip-armv8_gcc-11.2.0_musl.Linux-x86_64 sdk
cd sdk
./scripts/feeds update -a
./scripts/feeds install -a
```

5.18.3 Compile the package

download the source code of the example (a total of 3 examples are example1, example2, example3), and copy to the package directory:

```
git clone https://github.com/mwarning/openwrt-examples.git
cp -rf openwrt-examples/example* package/
rm -rf openwrt-examples/
```

Then enter the configuration menu through the following command:

```
make menuconfig
```

In the menu, select the following packages we want to compile (actually selected by default):

```
"Utilities" => "example1"
"Utilities" => "example3"
"Network" => "VPN" => "example2"
```

execute the following commands to compile the three software packages:

```
make package/example1/compile V=99
make package/example2/compile V=99
make package/example3/compile V=99
```

After the compilation is successful, you can find the ipk file in the bin directory, as shown below:

```
$ find ./bin -name example*.ipk
./bin/packages/aarch64_generic/base/example3_1.0.0-220420.38257_aarch64_generic.ipk
```

```
./bin/packages/aarch64_generic/base/example1_1.0.0-220420.38257_aarch64_generic.ipk
./bin/packages/aarch64_generic/base/example2_1.0.0-220420.38257_aarch64_generic.ipk
```

5.18.4 Install the ipk to NanoPi

You can use the scp command to upload the ipk file to NanoPi:

```
cd ./bin/packages/aarch64_generic/base/
scp example*.ipk root@192.168.2.1:/root/
```

Then use the opkg command to install them:

```
cd /root/
opkg install example3_1.0.0-220420.38257_aarch64_generic.ipk
opkg install example1_1.0.0-220420.38257_aarch64_generic.ipk
opkg install example2_1.0.0-220420.38257_aarch64_generic.ipk
```

5.19 Build FriendlyWrt using GitHub Actions

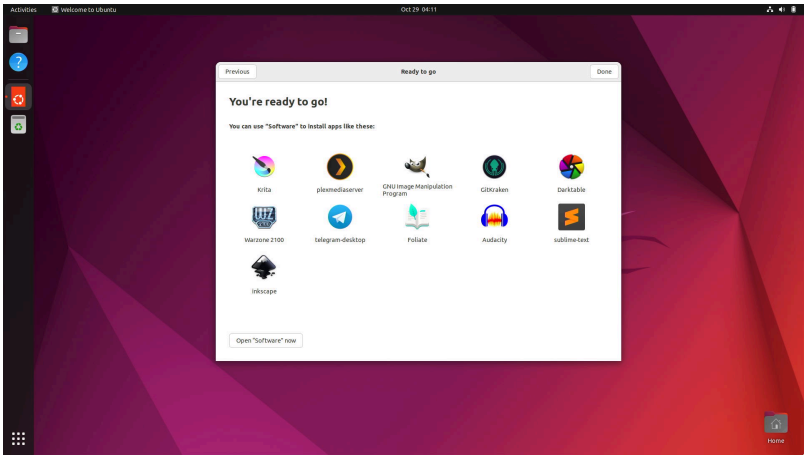
Please refre this link: https://github.com/friendlyarm/Actions-FriendlyWrt

6 Work with Ubuntu 22.04 Desktop

6.1 Introduction to Ubuntu 22.04 Desktop

Ubuntu 22.04 Desktop has the following features：

- Uses GNOME 42 as default desktop;
- Uses Wayland as default display server;
- Mali GPU-based OpenGL ES support;
- Support Rockhip MPP video hard coding and hard decoding;
- Pre-installed mpv and kodi, support video hardware decoding;
- Compatible with Docker and Plex server;



6.2 Account & Password

Regular Account:
User Name: pi
Password: pi

Root:
User Name: root
Password: fa

6.3 View IP address

Since the hostname is FriendlyElec.lan by default, you can use the ping command to get the IP address:

```
ping FriendlyElec.lan
```

6.4 Connect to Ubuntu via SSH

```
ssh pi@FriendlyElec.lan
```

The default password is: pi

6.5 Update Software Packages

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

6.6 Install Ubuntu software center

```
$ sudo apt-get install snapd
$ sudo snap install snap-store
```

6.7 Install the kernel-header package

```
sudo dpkg -i /opt/linux-headers-*.deb
```

try to compile a kernel module:

```
sudo apt update
sudo apt install git gcc make bc
git clone https://github.com/RinCat/RTL88x2BU-Linux-Driver.git
cd RTL88x2BU-Linux-Driver
make -j$(nproc)
sudo make install
sudo modprobe 88x2bu
```

6.8 Change time zone

6.8.1 Check the current time zone

```
timedatectl
```

6.8.2 List all available time zones

```
timedatectl list-timezones
```

6.8.3 Set the time zone (e.g. Shanghai)

```
sudo timedatectl set-timezone Asia/Shanghai
```

6.9 Change startup LOGO and Wallpaper

6.9.1 Change startup LOGO

Replace the following two files in the kernel source code directory and recompile the kernel：
kernel/logo.bmp
kernel/logo_kernel.bmp
Or use the script to operate, as shown below：

- Download scripts:

```
git clone https://github.com/friendlyarm/sd-fuse_rk3588.git -b kernel-6.1.y
cd sd-fuse_rk3588
```

- Compile kernel and repackage firmware

```
convert files/logo.jpg -type truecolor /tmp/logo.bmp
convert files/logo.jpg -type truecolor /tmp/logo_kernel.bmp
LOGO=/tmp/logo.bmp KERNEL_LOGO=/tmp/logo_kernel.bmp ./build-kernel.sh ubuntu-jammy-desktop-arm64
./mk-emmc-image.sh ubuntu-jammy-desktop-arm64
```

6.10 Soft Factory Reset

Execute the following command in a terminal:

```
sudo firstboot && sudo reboot
```

6.11 Start the program automatically at startup(For example Firefox)

Put the desktop file in the ~/.config/autostart/ directory, for example：

```
mkdir ~/.config/autostart/
cp /usr/share/applications/firefox.desktop ~/.config/autostart/
```

6.12 Disable auto-mounting

```
sudo systemctl mask udisks2
sudo reboot
```

6.13 Setup Chinese language and Input method

6.13.1 Setup Chinese language

Enter the following command and select 'zh_CN.UTF-8':

```
sudo dpkg-reconfigure locales
```

Add environment variables to .bashrc:

```
echo "export LC_ALL=zh_CN.UTF-8" >> ~/.bashrc
echo "export LANG=zh_CN.UTF-8" >> ~/.bashrc
echo "export LANGUAGE=zh_CN.UTF-8" >> ~/.bashrc
```

Reboot device:

```
sudo reboot
```

6.14 Video playback with hardware decoding

6.14.1 GUI

- Locate the video file in the file browser, right click and select "Play with mpv media player"
- Using Kodi player
- Using Chromium web browser, you can play videos on the web with hardware decoding (limited to the video formats supported by the CPU)

6.14.2 Command line

- Play local video file

```
export DISPLAY=:0.0
mpv --fs /home/pi/Videos/demo.mp4
```

- Play web-video

```
export DISPLAY=:0.0
mpv --fs https://www.youtube.com/watch?v=lK-nYDmC1Dk
```

6.15 How to Install Plex Media Server

Run the following command to install:

```
echo deb https://downloads.plex.tv/repo/deb public main | sudo tee /etc/apt/sources.list.d/plexmediaserver.list
curl https://downloads.plex.tv/plex-keys/PlexSign.key | sudo apt-key add -
sudo apt update
sudo apt install plexmediaserver
```

After successful installation, enable Plex (starts automatically at system startup):

```
sudo systemctl enable plexmediaserver
sudo systemctl start plexmediaserver
sudo systemctl status plexmediaserver
```

After installation, login to Plex server by entering the following address in your computer browser: http://IPAddress:32400/web/

6.16 Install Docker Engine

6.16.1 Install Docker Engine

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /e
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Let’s verify:

```
sudo docker info
```

6.16.2 Run Docker as a non-root user

```
sudo groupadd docker
sudo gpasswd -a ${USER} docker
sudo systemctl restart docker
sudo chmod a+rw /var/run/docker.sock
```

Let’s verify:

```
docker images
```

6.16.3 Testing Docker: Installing Nextcloud with docker

```
mkdir ~/nextcloud -p
docker run -d -p 8888:80  --name nextcloud  -v ~/nextcloud/:/var/www/html/ --restart=always --privileged=true  arm64v8/nextcloud
```

After installation, visit: http://Device-IP-Address:8888 on your computer browser to view the nextcloud web page.

6.17 Disable Automatic Login

Edit the /etc/gdm3/custom.conf file, set AutomaticLoginEnable to false:

```
[daemon]
AutomaticLoginEnable = false
```

6.18 WiFi Connection

6.18.1 Gui

Click on the icon on the top right in the FriendlyDesktop's main window, select your wanted WiFi hotspot and proceed with prompts

6.18.2 Console

Please visit: Use NetworkManager to configure network settings

6.19 Test OpenGL ES

First, change the CPU governor to performance:

```
sudo sh -c 'echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor'
sudo sh -c 'echo performance > /sys/devices/system/cpu/cpufreq/policy4/scaling_governor'
sudo sh -c 'echo performance > /sys/devices/system/cpu/cpufreq/policy6/scaling_governor'
```

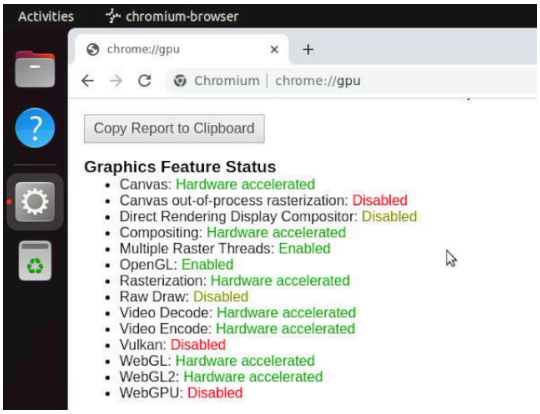
Start glmark2-es2-wayland:

```
glmark2-es2-wayland
```

6.20 Chromium web browser

6.20.1 GPU

Chromium web browser has enabled hardware acceleration by default, supports WebGL, and can view hardware acceleration details by entering the URL chrome://gpu, as shown below:



6.20.2 VPU

Play a video in the browser, then use fuser on the command line to view the mpp device node to confirm that the vpu interface is being called:

```
pi@FriendlyElec:~$ fuser /dev/mpp_service
/dev/mpp_service:     3258
```

If there is no content output from the fuser command, it means software decoding.

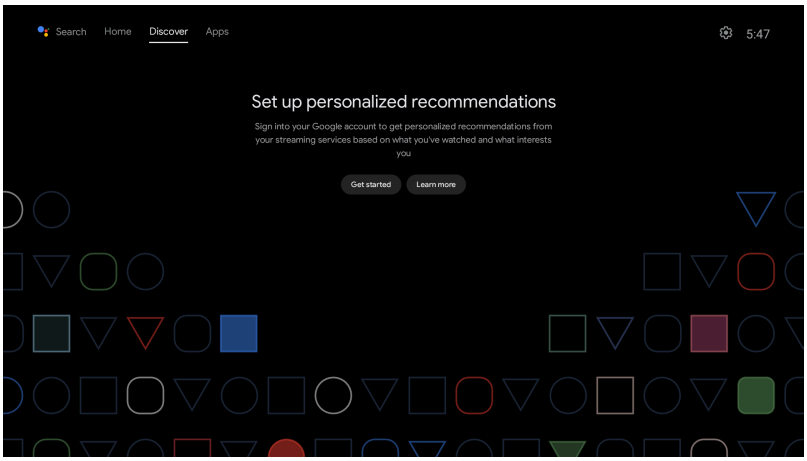
6.20.3 Check Supported Hardware Decoding Formats

Enter about://gpu in your browser's address bar and scroll to the bottom of the page to view the "Video Acceleration Information" table.
After playing a video, enter about://media-internals in your browser's address bar to check if hardware decoding was enabled for the most recent playback.

7 Work with Android

Android include the following features:

- There are two versions, TV and Tablet;
- Support infrared remote control (only for models with Ir Receiver);
- Support Bluetooth remote control (requires USB or M.2 Bluetooth module);
- Support wired network;
- Support WiFi (requires external USB or M.2 WIFI module);
- Support video hard decoding;



7.1 WiFi models supported by Android

7.1.1 M.2 WiFi Module

- RTL8822CE

7.1.1.1 Usb Dongle

- RTL8821CU (Vid: 0BDA, Pid: C811) (Test sample:TP-Link TL-WDN5200H)
- RTL8812AU (Vid: 0BDA, Pid: 8812)
- MediaTek MT7662 (Vid: 0E8D, Pid: 7612) (Test sample:COMFAST CF-WU782AC V2)

7.2 Bluetooch models supported by Android

7.2.1 Bluetooth Adapters

- RTL8822CE
- RTL8761B
- CSR8510 A10 Bluetooth Dongle 0a12:0001

(Note: unsupported device ID 0x2B89:0x8761)

7.2.2 Bluetooth Remote

- Amazon Fire TV Remote

7.3 How to use ADB

7.3.1 USB connection

Please note: After turning on the ADB, the USB3 port will work in Device mode, if you need to connect a device such as a USB stick, you need to turn off the ADB and restart the board
In general, ADB is disabled by default, please follow the steps below to enable it:

- Connect your development board to your computer using a USB A-to-A data cable, referring to the figure below, be sure to connect it to the USB port closest to the edge:

7.3.2 For Android Tablet

- Go to Settings -> About tablet -> tap the "Build number" at the bottom of the screen 7 times in a row;
- Go to Settings -> System -> Advance -> Developer options, check USB-Debugging;
- Reboot

7.3.3 For Android TV

- Click the Settings icon -> Device Preferences -> About -> tap the "Android TV OS build" at the bottom of the screen 7 times in a row;
- Click the Settings icon -> Device Preferences -> Developer options, check USB-Debugging;
- Reboot

7.3.4 Using ADB

- Install ADB drivers and commands based on your operating system;

- Normally, the Android status bar will prompt "USB debugging connected", indicating that ADB has been enabled. Enter the following command on your computer to check the connection：

```
$ adb devices
List of devices attached
27f7a63caa3faf16        device
```

- Enter adb shell:

```
$ adb shell
nanopi3:/ $
```

7.4 How to Change Default Launcher in Android TV

- Refer to the previous section to enable adb
- For example, install the third-party launcher Emotn UI via APK, visit the website https://app.emotn.com/ui/ to download the APK, and then install it using ADB:

```
$ adb install com.oversea.aslauncher_1.0.9.0_5094.apk
Performing Streamed Install
Success
```

- After the installation is complete, launch it, and then enter the following ADB command to obtain its package name:

```
$ adb shell dumpsys window | grep mCurrentFocus
mCurrentFocus=Window{7a950fb u0 com.oversea.aslauncher/com.oversea.aslauncher.ui.main.MainActivity}
```

- As you can see, the package name of Emotn UI is com.oversea.aslauncher, set it as the default launcher:

```
$ adb shell pm set-home-activity com.oversea.aslauncher
Success
```

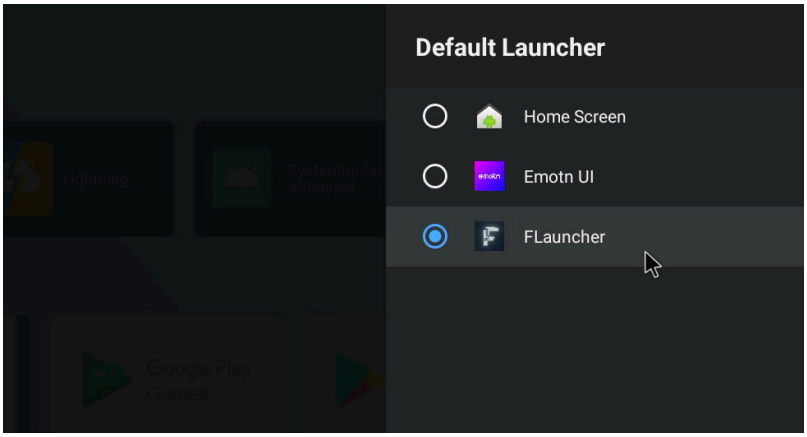
- Then comes the critical step, you need to disable the native launcher using the following command:

```
$ adb shell pm disable-user --user 0 com.google.android.tvlauncher
Package com.google.android.tvlauncher new state: disabled-user
```

- Finally, restart the device to see the effect, the device should boot directly into Emotn UI:

```
$ adb shell reboot
```

- From now on, if you want to install another launcher, you can switch between them through the GUI. for example, after installing FLauncher, you can enter the following settings interface to set FLauncher as the default launcher: Settings -> Device Preferences -> Advanced setting -> Default Launcher:



7.5 Wired networks on Android

- Any network port can connect to the network via DHCP
- If you want to configure a static IP, only eth0 interface is supported
- Some applications may have compatibility issues and report no network connection error, but the network is actually connected

7.6 EC20 4G LTE module on Android

EC20 support is disabled by default, you can check the status with the following command, if the EC20 is disabled, the number 1 will be displayed:

```
su
getprop persist.vendor.radio.no_modem_board
```

To enable EC20, use the following command (takes effect after a reboot):

```
su
setprop persist.vendor.radio.no_modem_board 0
```

8 Work with Debian11 Desktop

8.1 Introduction to Debian11 Desktop

Debian11 Desktop is a light-weighted debian desktop system，it has the following features：

- Uses Xfce as default desktop;
- Mali GPU-based OpenGL support;
- Support Rockhip MPP video hard coding and hard decoding;
- Pre-installed mpv and smplayer, both support 4K video hardware decoding;
- Pre-installed Chromium browser, support vpu/gpu hardware acceleration (video hard decoding limited to h264/mp4 format);
- Compatible with Plex Server and Docker;

8.2 Account & Password

Regular Account:
User Name: pi
Password: pi

Root:
the root user account is disabled by default, you may configure the root password through the 'sudo passwd root' command.

8.3 View IP address

Since the Debian Bullseye hostname is the hardware model by default, you can use the ping command to get the IP address:`ping NanoPi-R6C`

8.4 Connect to Debian via SSH

Run the following command`ssh pi@NanoPi-R6C`
The default password is: pi

8.5 Update Software Packages

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

8.6 Install x11vnc Server on Debian for Remote Access

8.6.1 Install x11vnc server

The following command to install x11vnc server:

```
sudo apt-get install x11vnc
```

8.6.2 Set your password

```
sudo x11vnc -storepasswd /etc/x11vnc.pwd
```

8.6.3 Setup x11vnc server with systemd auto start up

Create service configuration file:

```
sudo vi /lib/systemd/system/x11vnc.service
```

Let’s copy and paste the following configuration into our newly create service file:

```
[Unit]
Description=Start x11vnc at startup.
Requires=display-manager.service
After=syslog.target network-online.target
Wants=syslog.target network-online.target
```

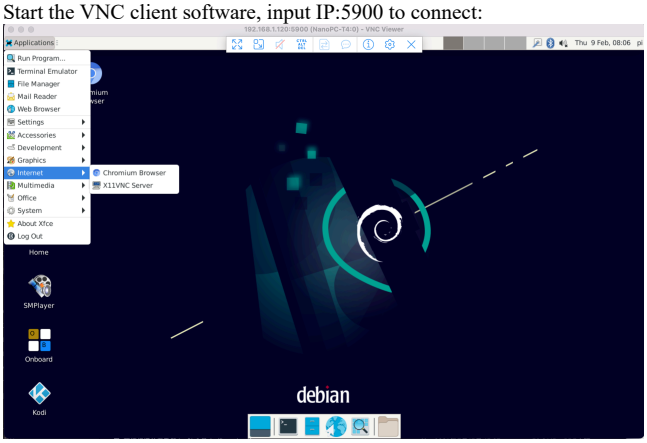
```
[Service]
Type=simple
ExecStart=/usr/bin/x11vnc -display :0 -forever -loop -noxdamage -repeat -rfbauth /etc/x11vnc.pwd -rfbport 5900 -shared -capslock -nomodtweak
ExecStop=/usr/bin/x11vnc -R stop
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

The following commands to reload the systmd system and to enable and start the x11vnc service:

```
sudo systemctl daemon-reload
sudo systemctl enable x11vnc.service
sudo systemctl start x11vnc
```

8.6.4 Testing remote access

Start the VNC client software, input IP:5900 to connect:



8.7 Install the kernel-header package

```
sudo dpkg -i /opt/linux-headers-*.deb
```

try to compile a kernel module:

```
sudo apt update
sudo apt install git gcc make bc
git clone https://github.com/RinCat/RTL88x2BU-Linux-Driver.git
cd RTL88x2BU-Linux-Driver
make -j$(nproc)
sudo make install
sudo modprobe 88x2bu
```

8.8 Change time zone

8.8.1 Check the current time zone

```
timedatectl
```

8.8.2 List all available time zones

```
timedatectl list-timezones
```

8.8.3 Set the time zone (e.g. Shanghai)

```
sudo timedatectl set-timezone Asia/Shanghai
```

8.9 Change startup LOGO and Wallpaper

8.9.1 Change startup LOGO

Replace the following two files in the kernel source code directory and recompile the kernel：
kernel/logo.bmp
kernel/logo_kernel.bmp
Or use the script to operate, as shown below：

- Download scripts:

```
git clone https://github.com/friendlyarm/sd-fuse_rk3588.git -b kernel-6.1.y --single-branch
cd sd-fuse_rk3588
```

- Compile kernel and repackage firmware

```
convert files/logo.jpg -type truecolor /tmp/logo.bmp
convert files/logo.jpg -type truecolor /tmp/logo_kernel.bmp
sudo LOGO=/tmp/logo.bmp KERNEL_LOGO=/tmp/logo_kernel.bmp ./build-kernel.sh debian-bullseye-desktop-arm64
sudo ./mk-sd-image.sh debian-bullseye-desktop-arm64
sudo ./mk-emmc-image.sh debian-bullseye-desktop-arm64
```

Note: If your system is not debian-bullseye-desktop-arm64, please specify according to the actual situation

8.9.2 Change Wallpaper

Modify the following configuration file:

```
/home/pi/.config/xfce4/xfconf/xfce-perchannel-xml/xfce4-desktop.xml
```

8.10 Soft Factory Reset

Execute the following command in a terminal:

```
sudo firstboot && sudo reboot
```

8.11 Start the program automatically at startup(For example Kodi)

Put the desktop file in the ~/.config/autostart/ directory, for example：

```
mkdir ~/.config/autostart/
cp /usr/share/applications/kodi.desktop ~/.config/autostart/
```

8.12 Disable auto-mounting

```
sudo systemctl mask udisks2
sudo reboot
```

8.13 Setup Chinese language and Input method

8.13.1 Setup Chinese language

Enter the following command and select 'zh_CN.UTF-8':

```
sudo dpkg-reconfigure locales
```

Add environment variables to .bashrc:

```
echo "export LC_ALL=zh_CN.UTF-8" >> ~/.bashrc
echo "export LANG=zh_CN.UTF-8" >> ~/.bashrc
echo "export LANGUAGE=zh_CN.UTF-8" >> ~/.bashrc
```

Reboot device:

```
sudo reboot
```

8.13.2 Installing Chinese input method

Enter the following command to install fcitx and Pinyin input method:

```
sudo apt update
sudo apt-get install fcitx fcitx-pinyin
sudo apt-get install im-config
sudo apt-get install fcitx-table*
sudo apt-get install fcitx-ui-classic fcitx-ui-light
sudo apt-get install fcitx-frontend-gtk2 fcitx-frontend-gtk3 fcitx-frontend-qt4
sudo apt-get remove --purge scim* ibus*
sudo reboot
```

After reboot, press Ctrl+Space to switch between Chinese and English input methods, and the input method icon will appear in the upper right corner, right-click the input method icon in the upper right corner to switch input methods in the pop-up menu, as shown below:

8.14 Installing Plex Multimedia Server

Visit the Plex website: https://www.plex.tv/media-server-downloads/
On the download page, select the category "Plex Media Server", choose "Linux" for the platform and "Ubuntu(16.04+)/Debian(8+) - ARMv8" for the version,
After downloading the deb package, use the dpkg command to install the package:

```
sudo dpkg -i plexmediaserver_1.31.0.6654-02189b09f_arm64.deb
```

After installation, login to the Plex server by typing the following URL into your computer browser: http://IP地址:32400/web/

8.15 Install Docker on Debian

Please refer to: How to Install Docker on Debian

8.16 How to test NPU

Please refer to: NPU

8.17 How to test VPU

Please refer to: VPU

8.18 WiFi Connection

8.18.1 Gui

Click on the icon on the top right in the Debian's main window, select your wanted WiFi hotspot and proceed with prompts

8.18.2 Console

Please visit: Use NetworkManager to configure network settings

8.19 Cancel auto-login

Edit file:

```
sudo vim /etc/lightdm/lightdm.conf
```

Comment out the following two lines (insert # in front of them):

```
autologin-user=pi
autologin-user-timeout=0
```

8.20 Test OpenGL ES

You can test it by clicking on the Terminator icon to start a commandline utility in the System Tools and run the following commands:

```
glmark2-es2
```

8.21 HDMI/DP LCD Resolution

Open the system's menu and go to Settings -> Display to customize your settings.

8.22 HiDPI and display scaling

Xfce supports HiDPI scaling which can be enabled using the settings manager: Go to Settings Manager > Appearance > Settings > Window Scaling and select 2 as the scaling factor.
Or Edit this file: ~/.config/xfce4/xfconf/xfce-perchannel-xml/xsettings.xml

8.23 Adjust HDMI overscan

Open the command line terminal and enter the command to operate, Note:
1) You need to login to the desktop;
2) If you are using ssh terminal, please use the same username as the desktop login. The default is pi. You cannot use the root user. you also need to assign the DISPLAY variable:

```
export DISPLAY=:0.0
```

8.23.1 Query which resolutions the display supports

```
xrandr -q
```

8.23.2 Set resolution

For example set to 1920X1080@60Hz:

```
xrandr --output HDMI-1 --mode 1920x1080 --refresh 60
```

8.23.3 Adjust the HDMI overscan

For example, the transformation scaling horizontal coordinates by 0.8, vertical coordinates by 1.04 and moving the screen by 35 pixels right and 19 pixels down:

```
xrandr --output HDMI-1 --transform 0.80,0,-35,0,1.04,-19,0,0,1
```

8.23.4 Automatic adjustment at boot

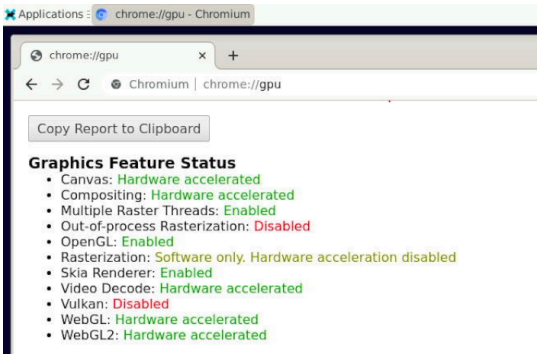
Edit ~/.config/autostart/lxrandr-autostart.desktop， Write the full xrandr command to the key at the beginning of "Exec= as shown below：

```
[Desktop Entry]
Type=Application
Name=LXRandR autostart
Comment=Start xrandr with settings done in LXRandR
Exec=sh -c 'xrandr --output HDMI-1 --mode 1920x1080 --refresh 50 --transform 1.04,0,-35,0,1.05,-30,0,0,1'
```

8.24 Chromium web browser

8.24.1 GPU

Chromium web browser has enabled hardware acceleration by default, supports WebGL, and can view hardware acceleration details by entering the URL chrome://gpu, as shown below:



8.24.2 VPU

Play a video in the browser, then use fuser on the command line to view the mpp device node to confirm that the vpu interface is being called:

```
pi@FriendlyElec:~$ fuser /dev/mpp_service
/dev/mpp_service:     3258
```

If there is no content output from the fuser command, it means software decoding.

8.25 Test hardware encoding

```
mpi_enc_test -w 1920 -h 1080 -t 7 -f 0 -o test.h264 -n 300
export XDG_RUNTIME_DIR=/run/user/0
ffplay test.h264
```

8.25.1 Check Supported Hardware Decoding Formats

Enter about://gpu in your browser's address bar and scroll to the bottom of the page to view the "Video Acceleration Information" table.
After playing a video, enter about://media-internals in your browser's address bar to check if hardware decoding was enabled for the most recent playback.

9 Work with Debian10 Desktop

- Refer to：
  - Debian Buster

10 Work with FriendlyCore

10.1 FriendlyCore User Account

- Non-root User:

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

- Root:

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

10.2 Update Software Packages

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

10.3 Setup Network Configurations

10.3.1 Set static IP address

By default "eth0" is assigned an IP address obtained via dhcp. If you want to change the setting you need to change the following file:

```
vi /etc/network/interfaces.d/eth0
```

For example if you want to assign a static IP to it you can run the following commands:

```
auto eth0
iface eth0 inet static
    address 192.168.1.231
    netmask 255.255.255.0
    gateway 192.168.1.1
```

The other ethernet port are set up with static IP addresses, as follows:
eth1: 192.168.2.1

10.3.2 Set a DNS

You also need to modify the following file to add the DNS configuration:

```
vi /etc/systemd/resolved.conf
```

For example, set to 192.168.1.1:

```
[Resolve]
DNS=192.168.1.1
```

Restart the systemd-resolved service with the following command:

```
sudo systemctl restart systemd-resolved.service
sudo systemctl enable systemd-resolved.service
```

10.3.3 Set up to use another network interface

To change the setting of "eth1" you can add a new file similar to eth0's configuration file under the /etc/network/interfaces.d/ directory.

10.4 Setup Wi-Fi

First, use the following command to check if Network-Manager is installed on your system:

```
which nmcli
```

If you have installed it, refer to this link to connect to WiFi: Use NetworkManager to configure network settings, If you do not have Network-Manager installed on your system, please refer to the following method to configure WiFi,
By default the WiFi device is "wlan0". You need to create a configuration file under "/etc/network/interfaces.d/" for WiFi:

```
vi /etc/network/interfaces.d/wlan0
```

Here is a sample wlan0 file:

```
auto lo
iface lo inet loopback
auto wlan0
iface wlan0 inet dhcp
wpa-driver wext
wpa-ssid YourWiFiESSID
wpa-ap-scan 1
wpa-proto RSN
wpa-pairwise CCMP
wpa-group CCMP
wpa-key-mgmt WPA-PSK
wpa-psk YourWiFiPassword
```

Please replace "YourWiFiESSID" and "YourWiFiPassword" with your WiFiESSID and password. After save and close the file you can connect to your WiFi source by running the following command:

```
sudo systemctl daemon-reload
sudo systemctl restart networking
```

After you power on your board it will automatically connect to your WiFi source.
Please note that if you use one TF card to boot multiple boards the WiFi device name will likely be named to "wlan1", "wlan2" and etc. You can reset it to "wlan0" by deleting the contents of the following file and reboot your board: /etc/udev/rules.d/70-persistent-net.rules

10.4.1 WiFi models supported

10.4.1.1 M.2 WiFi Module

- RTL8822CE

10.4.1.2 Usb Dongle

- RTL8821CU (Vid: 0BDA, Pid: C811) (Test sample:TP-Link TL-WDN5200H)
- RTL8812AU (Vid: 0BDA, Pid: 8812)
- MediaTek MT7662 (Vid: 0E8D, Pid: 7612) (Test sample:COMFAST CF-WU782AC V2)

10.5 Install the kernel-header package

```
sudo dpkg -i /opt/linux-headers-*.deb
```

10.6 Config status LEDs

First determine whether the system already exists the leds initialization service:

```
sudo systemctl status leds
```

If the leds service already exists, change the default behavior of the LEDs by editing the following file:

```
/etc/init.d/leds.sh
```

Since there is no leds service in the early firmware, you need to refer to the following guide to manually configure the LEDs. First, set the following kernel modules to be automatically loaded at boot:

```
modprobe ledtrig-netdev
echo ledtrig-netdev > /etc/modules-load.d/ledtrig-netdev.conf
```

Put the following into the autorun script to associate the status leds with the ethernet interface, and you can configure it to behave in other ways by referring to these content:

```
echo netdev > /sys/class/leds/wan_led/trigger
echo eth0 > /sys/class/leds/wan_led/device_name
echo 1 > /sys/class/leds/wan_led/link

echo netdev > /sys/class/leds/lan1_led/trigger
echo eth1 > /sys/class/leds/lan1_led/device_name
echo 1 > /sys/class/leds/lan1_led/link
```

The onboard LED1 can be configured as needed, the node is /sys/class/leds/usr_led.

10.7 Delete Qt5 and related files

Execute the following commands：

```
su root
cd /
rm -rf usr/local/Trolltech/Qt-5.10.0-rk64one usr/local/Trolltech/Qt-5.10.0-rk64one-sdk usr/bin/setqt5env* usr/bin/qt5demo etc/qt5
rm -rf opt/{qt5-browser,Qt5_CinematicExperience,qt5-multi-screen-demo,qt5-nmapper,qt5-player,qt5-smarthome,QtE-Demo,qt5-qml-image-viewer,dual-camera}
```

11 How to Compile

11.1 Setup Development Environment

11.1.1 Method 1: Using docker to cross-compile

Please refre to docker-cross-compiler-novnc (https://github.com/friendlyarm/docker-cross-compiler-novnc)

11.1.2 Method 2: Setup build environment on the host machine

11.1.2.1 Install required packages

Install and run requirements ubuntu 20.04, install required packages using the following commands:

```
sudo apt-get -y update
sudo apt-get install -y sudo curl
sudo bash -c \
  "$(curl -fsSL https://raw.githubusercontent.com/friendlyarm/build-env-on-ubuntu-bionic/master/install.sh)"
```

The following cross-compilers will be installed:

Version	Architecture	Compiler path	Purpose
4.9.3	armhf	/opt/FriendlyARM/toolchain/4.9.3	Can be used to build 32-bit ARM applications
6.4	aarch64	/opt/FriendlyARM/toolchain/6.4-aarch64	Can be used to build kernel 4.4
11.3	aarch64	/opt/FriendlyARM/toolchain/11.3-aarch64	Can be used to build kernel 4.19 or higher and U-Boot

11.1.2.2 Setting the compiler path

Based on the table in the previous section, select the appropriate version of the compiler and add the compiler's path to PATH. For example, if you want to use the 11.3 cross-compiler, edit ~/.bashrc using vi and add the following content to the end:

```
export PATH=/opt/FriendlyARM/toolchain/11.3-aarch64/bin:$PATH
export GCC_COLORS=auto
```

Run the ~/.bashrc script to make it effective in the current commandline. Note: there is a space after ".":

```
. ~/.bashrc
```

To verify if the installation was successful：

```
$ aarch64-linux-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-linux-gcc
COLLECT_LTO_WRAPPER=/opt/FriendlyARM/toolchain/11.3-aarch64/libexec/gcc/aarch64-cortexa53-linux-gnu/11.3.0/lto-wrapper
Target: aarch64-cortexa53-linux-gnu
Configured with: /home/cross/arm64/src/gcc/configure --build=x86_64-build_pc-linux-gnu --host=x86_64-build_pc-linux-gnu --target=aarch64-cortexa53-linux-gnu --prefix=/opt/FriendlyARM/t
Thread model: posix
Supported LTO compression algorithms: zlib
gcc version 11.3.0 (ctng-1.25.0-119g-FA)
```

11.2 Build Openwrt/Friendlywrt

11.2.1 Download Code

Two versions are available, please choose as required:

11.2.1.1 FriendlyWrt 21.02

```
mkdir friendlywrt21-rk3588
cd friendlywrt21-rk3588
git clone https://github.com/friendlyarm/repo --depth 1 tools
tools/repo init -u https://github.com/friendlyarm/friendlywrt_manifests -b master-v21.02 \
        -m rk3588.xml --repo-url=https://github.com/friendlyarm/repo  --no-clone-bundle
tools/repo sync -c  --no-clone-bundle
```

11.2.1.2 FriendlyWrt 23.05

```
mkdir friendlywrt23-rk3588
cd friendlywrt23-rk3588
git clone https://github.com/friendlyarm/repo --depth 1 tools
tools/repo init -u https://github.com/friendlyarm/friendlywrt_manifests -b master-v23.05 \
        -m rk3588.xml --repo-url=https://github.com/friendlyarm/repo  --no-clone-bundle
tools/repo sync -c  --no-clone-bundle
```

11.2.2 First compilation step

```
./build.sh rk3588.mk  # or rk3588-docker.mk
```

All the components (including u-boot, kernel, and friendlywrt) are compiled and the sd card image will be generated, then execute the following command to generate the image file for installing the system into the emmc:

```
./build.sh emmc-img
```

After making changes to the project, the sd card image needs to be repackaged by running the following command:

```
./build.sh sd-img
```

11.2.3 Secondary compilation steps

```
cd friendlywrt
make menuconfig
rm -rf ./tmp
make -j${nproc}
cd ../
./build.sh sd-img
./build.sh emmc-img
```

11.2.4 Build u-boot only

```
./build.sh uboot
```

11.2.5 Build kernel only

```
./build.sh kernel
```

11.2.6 Build friendlywrt only

```
./build.sh friendlywrt
```

Or go to the friendlywrt directory and follow the standard openwrt commands. If you get an error with the above command, try using the following command to compile in a single thread:

```
cd friendlywrt
make -j1 V=s
```

11.3 Build Buildroot

please refer to: Buildroot

11.4 Build Other Linux

11.4.1 Kernel and u-boot versions

Operating System	Kernel Version	U-boot version	Cross-compiler	Partition type	Packaging Tool	Kernel branch
buildroot	linux v5.10.y	u-boot v2017.09	11.3-aarch64	GPT (https://github.com/friendlyarm/sd-fuse_rk3588/blob/master/prebuilt/parameter.template)	sd-fuse (https://github.com/friendlyarm/sd-fuse_rk3588/tree/master)	nanopi5-v5.10.y_opt (https://github.com/friendly rockchip/tree/nanopi5-v5.1(
openmediavault-arm64	linux v6.1.y			GPT (https://github.com/friendlyarm/sd-fuse_rk3588/blob/kernel-6.1.y/prebuilt/parameter-ext4.txt)	sd-fuse (https://github.com/friendlyarm/sd-fuse_rk3588/tree/kernel-6.1.y)	nanopi6-v6.1.y (https://github.com/friendly rockchip/tree/nanopi6-v6.1.
ubuntu-jammy-desktop-arm64				GPT (https://github.com/friendlyarm/sd-fuse_rk3588/blob/kernel-6.1.y/prebuilt/parameter.template)		
ubuntu-jammy-minimal-arm64						
ubuntu-jammy-x11-desktop-arm64						
ubuntu-focal-desktop-arm64						
friendlycore-focal-arm64						
debian-bullseye-core-arm64						
debian-bullseye-desktop-arm64						
debian-bullseye-minimal-arm64						
friendlywrt21						
friendlywrt21-docker						
friendlywrt23						
friendlywrt23-docker						

- Kernel git repo：https://github.com/friendlyarm/kernel-rockchip
- U-boot git repo：https://github.com/friendlyarm/uboot-rockchip
- The cross-compile toolchain is located in the path: /opt/FriendlyARM/toolchain/
- The sd-fuse (https://github.com/friendlyarm/sd-fuse_rk3588/tree/kernel-6.1.y) is a helper script to make bootable SD card image.

11.4.2 Build kernel linux-v6.1.y

Clone the repository to your local drive then build:

```
git clone https://github.com/friendlyarm/kernel-rockchip --single-branch --depth 1 -b nanopi6-v6.1.y kernel-rockchip
cd kernel-rockchip
export PATH=/opt/FriendlyARM/toolchain/11.3-aarch64/bin/:$PATH
touch .scmversion
# Configuring the Kernel
# Load default configuration
make CROSS_COMPILE=aarch64-linux-gnu- ARCH=arm64 nanopi6_linux_defconfig
# Optionally, load configuration for FriendlyWrt
# make CROSS_COMPILE=aarch64-linux-gnu- ARCH=arm64 nanopi6_linux_defconfig friendlywrt.config
# Optionally, if you want to change the default kernel config
# make CROSS_COMPILE=aarch64-linux-gnu- ARCH=arm64 menuconfig
# Start building kernel
make CROSS_COMPILE=aarch64-linux-gnu- ARCH=arm64 nanopi6-images -j$(nproc)
# Start building kernel modules
mkdir -p out-modules && rm -rf out-modules/*
make CROSS_COMPILE=aarch64-linux-gnu- ARCH=arm64 INSTALL_MOD_PATH="$PWD/out-modules" modules -j$(nproc)
make CROSS_COMPILE=aarch64-linux-gnu- ARCH=arm64 INSTALL_MOD_PATH="$PWD/out-modules" modules_install
KERNEL_VER=$(make CROSS_COMPILE=aarch64-linux-gnu- ARCH=arm64 kernelrelease)
[ ! -f "$PWD/out-modules/lib/modules/${KERNEL_VER}/modules.dep" ] && depmod -b $PWD/out-modules -E Module.symvers -F System.map -w ${KERNEL_VER}
(cd $PWD/out-modules && find . -name \*.ko | xargs aarch64-linux-strip --strip-unneeded)
```

The generated files:

kernel.img	resource.img	~~boot.img This img is deprecated~~	The kernel modules are located in the out-modules directory

Run your build:
Please refre to #Running the build

11.4.3 Build u-boot v2017.09

Clone the repository to your local drive then build:

```
git clone https://github.com/friendlyarm/rkbin --single-branch --depth 1 -b nanopi6
git clone https://github.com/friendlyarm/uboot-rockchip --single-branch --depth 1 -b nanopi6-v2017.09
export PATH=/opt/FriendlyARM/toolchain/11.3-aarch64/bin/:$PATH
cd uboot-rockchip/
./make.sh nanopi6
```

After the compilation, the following files will be generated:

uboot.img	rk3588_spl_loader_v1.08.111.bin (aka MiniLoaderAll.bin)

Run your build:
Please refre to #Running the build

11.4.4 Running the build

11.4.4.1 Install to target board

RK3588 uses GPT partitions by default, you can use the dd command, but be careful to choose the right output device:

- The SD/TF Card device node: /dev/mmcblk0
- The eMMC device node: /dev/mmcblk2

Use the 'parted' command to view the partition layout:

```
parted /dev/mmcblk2 print
```

Sample outputs:

```
Model: MMC A3A551 (sd/mmc)
Disk /dev/mmcblk2: 31.0GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name      Flags
 1      8389kB  12.6MB  4194kB               uboot
 2      12.6MB  16.8MB  4194kB               misc
 3      16.8MB  21.0MB  4194kB               dtbo
 4      21.0MB  37.7MB  16.8MB               resource
 5      37.7MB  79.7MB  41.9MB               kernel
 6      79.7MB  113MB   33.6MB               boot
 7      113MB   147MB   33.6MB               recovery
 8      147MB   31.0GB  30.9GB  ext4         rootfs
```

as shown above, the resource partition is located at 4 and the kernel partition is located at 5. Use the dd command to write the resource.img and kernel.img files to these partitions, the commands are as follows:

```
dd if=resource.img of=/dev/mmcblk2p4 bs=1M
dd if=kernel.img of=/dev/mmcblk2p5 bs=1M
```

If you want to update u-boot:

```
dd if=uboot.img of=/dev/mmcblk2p1 bs=1M
```

To update new driver modules, copy the newly compiled driver modules to the appropriate directory under /lib/modules.

11.4.4.2 Packaging and creating an SD image

To create a new OS image file, you need to use the "sd-fuse" packaging tool.

"sd-fuse" is a collection of scripts that can be used to create bootable SD card images for FriendlyElec boards. Its main features include:

- Creation of root filesystem images from a directory
- Building of bootable SD card images
- Simple compilation of kernel, U-Boot, and third-party drivers

Please click on the following link to find out more:

Kernel version	Packaging Tool
linux v6.1.y	sd-fuse_rk3588 (https://github.com/friendlyarm/sd-fuse_rk3588/tree/kernel-6.1.y)

11.4.4.3 USB flashing

11.4.4.3.1 Linux

Reboot the board and enter loader mode with the following command:

```
sudo reboot loader
```

To flash U-Boot and kernel using the "upgrade_tool_v2.17_for_linux" tool, please use the following command:

```
sudo upgrade_tool di -k kernel.img
sudo upgrade_tool di -re resource.img
sudo upgrade_tool di -u uboot.img
sudo upgrade_tool RD
```

Note: "upgrade_tool" is a command-line tool provided by Rockchip for Linux operating systems (Linux_Upgrade_Tool).

11.5 Build the code using scripts

11.5.1 Download scripts and image files

```
git clone https://github.com/friendlyarm/sd-fuse_rk3588.git --single-branch -b kernel-6.1.y
cd sd-fuse_rk3588
tar xvzf /path/to/netdrive/03_Partition\ image\ files/friendlycore-focal-arm64-images.tgz
```

11.5.2 Compile the kernel

Download the kernel source code and compile it. the relevant image files in the friendlycore-focal-arm64 directory will be automatically updated, including the kernel modules in the file system:

```
git clone https://github.com/friendlyarm/kernel-rockchip --depth 1 -b nanopi6-v6.1.y kernel-rk3588
KERNEL_SRC=$PWD/kernel-rk3588 ./build-kernel.sh friendlycore-focal-arm64
```

11.5.3 Compile the kernel headers

```
git clone https://github.com/friendlyarm/kernel-rockchip --depth 1 -b nanopi6-v6.1.y kernel-rk3588
MK_HEADERS_DEB=1 BUILD_THIRD_PARTY_DRIVER=0 KERNEL_SRC=$PWD/kernel-rk3588 ./build-kernel.sh friendlycore-focal-arm64
```

11.5.4 Compile the uboot

Download the uboot source code and compile it. the relevant image files in the friendlycore-focal-arm64 directory will be automatically updated:

```
git clone https://github.com/friendlyarm/uboot-rockchip --depth 1 -b nanopi6-v2017.09
UBOOT_SRC=$PWD/uboot-rockchip ./build-uboot.sh friendlycore-focal-arm64
```

11.5.5 Generate new image

Repackage the image file in the friendlycore-focal-arm64 directory into sd card image:

```
./mk-sd-image.sh friendlycore-focal-arm64
```

After the command is completed, the image is in the out directory, you can use the dd command to make the SD boot card, for example:

```
dd if=out/rk3588-sd-friendlycore-focal-5.10-arm64-YYYYMMDD.img of=/dev/sdX bs=1M
```

11.6 Building AOSP from source

11.6.1 Hardware and Software Requirements

- Your computer should have at least 16GB of RAM and 300GB of disk space. We recommend using a machine with 32GB of RAM and a large-capacity, high-speed SSD, and we do not recommend using virtual machines.
- If you encounter compilation errors, they may be caused by problems with the compilation environment. We recommend using the following Docker container for compilation: docker-cross-compiler-novnc (https://github.com/friendlyarm/docker-cross-compiler-novnc).

11.6.2 Download source from the netdrive

Netdisk URL: Click here (http://download.friendlyelec.com/NanoPiR6C)
File location on netdisk："07_Source codes/rk35xx-android12-xxxxxxx-YYYYMMDD.tgz" (YYYYMMDD represents the date of the package, and xxxxxxx represents the final commit-id)
Unzip and fetch updates:

```
tar xzf '/path/to/netdisk/07_Source codes/rk35xx-android12-xxxxxxx-YYYYMMDD.tgz'
cd rk35xx-android12
git pull
```

11.6.3 Tablet profile build (First Build)

```
echo "ROCKCHIP_DEVICE_DIR := device/rockchip/rk3588/nanopi6" > .rockchip_device.mk
# export INSTALL_GAPPS_FOR_TESTING=yes  # include google apps
. setenv.sh
./build.sh -FMu
```

11.6.4 TV profile build (First Build)

```
echo "ROCKCHIP_DEVICE_DIR := device/rockchip/rk3588/nanopi6_box" > .rockchip_device.mk
# export INSTALL_GAPPS_FOR_TESTING=yes  # include google apps
. setenv.sh
./build.sh -FMu
```

11.6.5 Second build

```
# export INSTALL_GAPPS_FOR_TESTING=yes  # include google apps
. setenv.sh
make
./build.sh -Mu
```

11.6.6 Running your AOSP build

After the Android compilation is completed, the image file will be stored in the rockdev/Image-aosp_nanopi3 subdirectory of the Android source code directory.

11.6.6.1 USB Flashing

Use the rockchip tool to flash the following file: rockdev/Image-aosp_nanopi3/update.img

11.6.6.2 SD-to-eMMC Flashing

Refer to the following steps：
1) Insert the SD card of the eflasher system into the host;
2) Copy the files in the rockdev/Image-aosp_nanopi3 directory to the android12 or androidtv directory in the FRIENDLYARM partition of the SD card:

```
sudo cp -af parameter.txt config.cfg MiniLoaderAll.bin uboot.img \
    dtbo.img vbmeta.img boot.img recovery.img \
    misc.img pcba_small_misc.img pcba_whole_misc.img \
    baseparameter.img super.img /media/$USER/FriendlyARM/android12
```

3) Insert the SD card into NanoPi-R6C and re-flash;

11.6.7 Pack the new SD Image

```
git clone https://github.com/friendlyarm/sd-fuse_rk3588.git
SDFUSE=$PWD/sd-fuse_rk3588
mkdir $SDFUSE/android12

cd /path/to/rk35xx-android12/rockdev/Image-aosp_nanopi3
cp -af parameter.txt config.cfg MiniLoaderAll.bin uboot.img \
    dtbo.img vbmeta.img boot.img recovery.img \
    misc.img pcba_small_misc.img pcba_whole_misc.img \
    baseparameter.img super.img $SDFUSE/android12

cd $SDFUSE/
./mk-sd-image.sh android12

tar xvzf /path/to/netdrive/03_Partition\ image\ files/emmc-flasher-images.tgz
./mk-emmc-image.sh android12
```

For more information, please refer to #Packaging and creating an SD image

12 Backup rootfs and create custom SD image (to burn your application into other boards)

12.1 Backup rootfs

Run the following commands on your target board. These commands will back up the entire root partition:

```
sudo passwd root
su root
cd /
tar --warning=no-file-changed -cvpzf /rootfs.tar.gz \
    --exclude=/rootfs.tar.gz --exclude=/var/lib/docker/runtimes \
    --exclude=/etc/firstuser --exclude=/etc/friendlyelec-release \
    --exclude=/usr/local/first_boot_flag --one-file-system /
```

Note: if there is a mounted directory on the system, an error message will appear at the end, which can be ignored.

12.2 Making a bootable SD card from a root filesystem

Run the following script on your Linux PC host, we'll only mention "debian-bullseye-desktop-arm64 os" for brevity, but you can apply the same process for every linux OS.

```
su root
git clone https://github.com/friendlyarm/sd-fuse_rk3588 --single-branch -b kernel-6.1.y
cd sd-fuse_rk3588
tar xvzf /path/to/netdrive/03_Partition\ image\ files/debian-bullseye-desktop-arm64-images.tgz
tar xvzf /path/to/netdrive/03_Partition\ image\ files/emmc-eflasher-images.tgz
scp pi@BOARDIP:/rootfs.tar.gz /rootfs.tar.gz
mkdir rootfs
tar xzf rootfs.tar.gz -C rootfs
./build-rootfs-img.sh rootfs debian-bullseye-desktop-arm64
./mk-sd-image.sh debian-bullseye-desktop-arm64
./mk-emmc-image.sh debian-bullseye-desktop-arm64 autostart=yes
```

13 Configuring kernel command line parameters

13.1 eMMC Boot

Here are the steps:
Make an eflahser bootable SD card (use the firmware file starting with rk3xxxx-eflasher-),
Insert the SD card into your computer, go to the SD card's OS-related directory, and edit the file parameter.txt, which is a text file containing command-line parameters,
Then boot from the SD card and burn the system to the eMMC.

13.2 SD Boot

To modify the command line parameters of the SD card, you need to repackage the SD card image file,
you can use the sd-fuse script we provide to assist packaging:

```
git clone https://github.com/friendlyarm/sd-fuse_rk3588.git -b kernel-6.1.y --single-branch
cd sd-fuse_rk3588
tar xvzf /path/to/netdrive/03_Partition\ image\ files/friendlywrt23-images.tgz
tar xvzf /path/to/netdrive/03_Partition\ image\ files/emmc-flasher-images.tgz
vim friendlywrt23/parameter.txt   # Edit command-line parameters
./mk-sd-image.sh friendlywrt23    # Repackage sd image file
./mk-emmc-image.sh friendlywrt23  # Repackage sd-to-emmc image file
```

14 Connect NVME SSD High Speed Hard Disk

14.1 Detection of SSD

```
root@FriendlyELEC:~# cat /proc/partitions
major minor  #blocks  name
   1        0       4096 ram0
 259        0  125034840 nvme0n1
```

If there is a nvme0n1 device node it means an SSD is recognized.

14.2 Partition of SSD

To mount an SSD under Linux we re-partition it as one section by running the following command:

```
(echo g; echo n; echo p; echo 1; echo ""; echo ""; echo w; echo q) | fdisk /dev/nvme0n1
```

If you want to re-partition it to multiple sections you can run "fdisk /dev/nvme0n1". For more detail about this command refer to the fdisk's manual.

14.3 Format Section to EXT4

After an SSD is successfully partitioned you can check its sections by running "cat /proc/partitions". The /dev/nvme0n1p1 section is used to store data:

```
root@FriendlyELEC:~# cat /proc/partitions
major minor  #blocks  name

   1        0       4096 ram0
```

```
259        0  125034840 nvme0n1
259        2  125033816 nvme0n1p1
```

The following command formats a section to ext4:

```
mkfs.ext4 /dev/nvme0n1p1
```

14.4 Auto Mount SSD on System Startup

Before we mount an SSD's section you need to know its Block ID. You can check it by running "blkid":

```
blkid /dev/nvme0n1p1
/dev/nvme0n1p1: UUID="d15c4bbf-a6c3-486f-8f81-35a8dbd46057" TYPE="ext4" PARTUUID="887628f0-01"
```

Add a "Block ID" to "/etc/fstab" and here is what it looks like

```
UUID=<Block ID> /media/nvme ext4 defaults 0 0
```

You need to replace <Block ID> with the UUID obtained by running "blkid". To mount the SSD in our example we made the "/etc/fstab" file as follows:

```
UUID=d15c4bbf-a6c3-486f-8f81-35a8dbd46057 /media/nvme ext4 defaults 0 0
```

We want to mount an SSD to "/media/nvme" but this directory doesn't exist. Therefore we create it and change its access right by running the following commands:

```
mkdir -p /media/nvme
chmod 777 /media/nvme
```

Run "mount" to check if the SSD is mounted successfully:

```
mount /media/nvme
```

You can reboot your board to check if your SSD will be automatically mounted:

```
reboot
```

15 Link to Rockchip Resources

- Rockchip_RK3588S_Datasheet_V1.0-20211221.pdf (https://wiki.friendlyelec.com/wiki/images/1/17/Rockchip_RK3588S_Datasheet_V1.0-20211221.pdf)

16 Schematic, PCB CAD File

- Schematic: NanoPi_R6C_2302_SCH.PDF (https://wiki.friendlyelec.com/wiki/images/f/f7/NanoPi_R6C_2302_SCH.PDF)
- PCB CAD File：NanoPi_R6C_2302_2d_dxf.zip (https://wiki.friendlyelec.com/wiki/images/9/91/NanoPi_R6C_2302_2d_dxf.zip)

17 Update Logs

17.1 2024-01-31

17.1.1 Debian/Ubuntu/FriendlyCore/Buildroot

- Add adb support

17.1.2 Android 12 & Android TV

- Add wifidisplay (no hdcp) support
- Add Wi-Fi direct support for rtl8822ce and rtl8812au

17.1.3 FriendlyWrt

- Add wireless repeater mode support for rtl8822ce

17.2 2024-01-24

17.2.1 Buildroot

- Fix "no sound from headphone jack" issue on nanopc-t6

17.3 2024-01-17

17.3.1 Linux

- Fix NPU interrupt exception
- Fix Buildroot default audio output setting
- Fix mpp decoding issue on Buildroot

17.3.2 Android 12

- Fix the issue that headphone audio not working if hdmi-in connected

17.4 2023-12-23

17.4.1 Android 12 & Android TV

- Fix the UsbCamera preview abnormal issue
- Fix connection for ps5/dualshock controller
- Improve support for non-16:9 resolutions in HDMI output
- Update SDK to Rockchip Android 12.1 rkr14.2

17.4.2 FriendlyWrt

- Fix an issue that unable to use eMMC tools for large-capacity eMMC

17.5 2023-12-01

17.5.1 FriendlyWrt

- Update to openwrt-23.05.2

17.6 2023-11-13

17.6.1 Debian/Ubuntu/FriendlyCore

- Updated npu driver to 0.9.2, fixed known issues

17.6.2 Android 12

- Built-in EC20 support, no system updates required (#EC20_4G_LTE_module_on_Android)

17.7 2023-10-31

17.7.1 Add a new system

- Add NAS system OpenMediaVault, base on Debian 11 with kernel 6.1

17.7.2 Debian/Ubuntu/FriendlyCore

- Update to kernel 6.1

17.7.3 FriendlyWrt

- Update to kernel 6.1
- Update to openwrt-23.05

17.8 2023-09-09

17.8.1 Android 12

- Add hardware library support (Click to view details)
- Add support for OTA and virtual A/B partitions
- Add EC20 support for NanoPC-T6 (#EC20_4G_LTE_module_on_Android)

17.9 2023-08-15

- Add support for Buildroot

17.10 2023-07-19

17.10.1 Ubuntu Focal Desktop

- Add Ubuntu 20.04 desktop, with LXQT lightweight desktop

17.10.2 Ubuntu Jammy Desktop

- Update chromium to a freshly compiled version 114.0.5735.198
- Update the libmali driver to the new version g13p0-6

17.10.3 Debian/Ubuntu/Android

- Fix RTL9210 enclosures slowdown issue

17.10.4 Ubuntu Jammy X11 Desktop

- Add Ubuntu X11 desktop, based on Xubuntu, using panfrost (https://gitlab.com/panfork/mesa) driver for UI acceleration, rkmpp/mpv etc for video acceleration

17.11 2023-07-01

17.11.1 Debian11

- Update to rockchip bsp linux-5.10-gen-rkr5, improve xserver and video playback performance
- Update kernel to 5.10.160
- Fix some known issues

17.12 2023-06-25

17.12.1 Debian11

- Update mpp/ffmpeg/gsteamer-rockchip/libv4l-rkmpp/libdrm-cursor packages
- Fix mouse cursor disappearing issue
- Fix headphone jack no sound issue

17.12.2 Android TV & Android 12

- Update to rockchip android-12.1-mid-rkr14
- Upgrade kernel to 5.10.160
- Add support for MediaTek MT7921 wireless card (WiFi only)

17.12.3 FriendlyCore

- Fix headphone jack no sound issue

17.13 2023-06-16

17.13.1 Debian11

- Merge changes from upsteam version rockchip linux-5.10-gen-rkr4.1
- Improved UI performance and video playback performance
- Switch desktop to XFCE
- Add support for MediaTek MT7921 wireless card
- Fix some known issues
- WiKi add HDMI-IN video capture instructions

17.14 2023-06-09

17.14.1 FriendlyWrt

- Add support for MediaTek MT7921 wireless card

17.14.2 Ubuntu22

- Add support for MediaTek MT7921 wireless card
- Add panfrost GPU driver (located in /opt/panfrost)

17.15 2023-06-06

17.15.1 Android TV & Android 12

- Auto expand storage space for SD boot

17.16 2023-06-01

17.16.1 Android TV

- Fix home button issue

17.17 2023-05-27

17.17.1 Ubuntu22

- Fix multi-screen display issue (4K+4K+4K or 8K+4K)

17.18 2023-05-26

17.18.1 Ubuntu22

- Fixed audio issues with DP
- Fixed 8K display (HDMI TX0 port)
- Optimized r8125 performance and stability

17.18.2 FriendlyWrt

- Updated v22.03 to openwrt-22.03.5
- Updated v21.02 to openwrt-21.02.7
- Optimized r8125 performance and stability

17.19 2023-05-21

17.19.1 Debian11

- Update to Rockchip sdk version linux-5.10-gen-rkr4
- Switch desktop to LXDE

17.20 2023-05-15

17.20.1 FriendlyCore Focal

- Add Qt 5.10 support

17.21 2023-05-05

17.21.1 Android

- Add USB bluetooth dongle support (rtl8761bu)
- Add M.2 RTL8822CE bluetooth support

17.22 2023-04-26

17.22.1 FriendlyWrt：

- Upgrade v22.03 to openwrt-22.03.4
- Upgrade v21.02 to openwrt-21.02.6

17.23 2023-02-10

17.23.1 Android update

- android support direct boot from sd card
- Fix the problem that in some cases hdmi has no signal output and needs to be plugged in once to recover
- Support enable adb function through android interface

17.23.2 Added Debian11

There are three versions:

- Debian11 Core: Command-line only
- Debian11 Minimal: With Xfce desktop, lite version
- Debian11 Desktop: With Xfce desktop, full version

17.24 2023-01-09

17.24.1 New OS：

- Android12: Android 12 Tablet
- FrienclyCore: Lightweight version of Linux, based on Ubuntu Core 20.04, no desktop system

17.24.2 Android：

- Add support for USB WiFi module (Model: RTL8822CE/RTL8812AU/MediaTek MT7662)
- Update SDK to Rockchip android-12.1-mid-rkr12

17.24.3 Ubuntu jammy：

- optimized the systemd service

17.25 2022-12-13

17.25.1 Ubuntu jammy：

- Fix WiFi issue

17.26 2022-12-04

17.26.1 FriendlyWrt：

- Fix the issue that the storage space cannot be expanded
- Improve stability of the eMMC Tools

17.27 2022-11-17

17.27.1 FriendlyWrt：

- Fix 1G LAN port not working issue

17.27.2 Ubuntu jammy desktop：

- Add chromium video hardware decoding support
- Add hardware cursor support

17.28 2022-10-25

Initial Release

Retrieved from "https://wiki.friendlyelec.com/wiki/index.php?title=NanoPi_R6C&oldid=24940"

- This page was last modified on 3 August 2023, at 07:16.