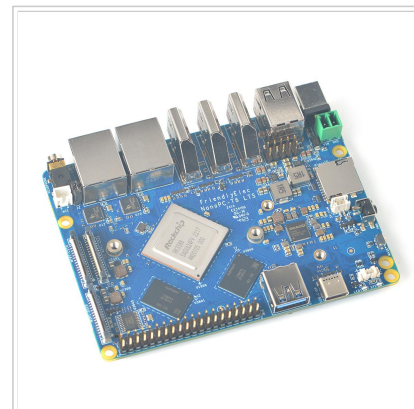


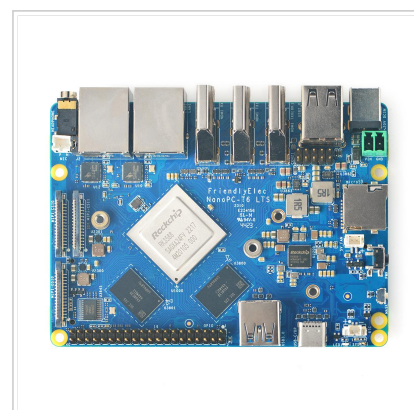
NanoPC-T6

Contents

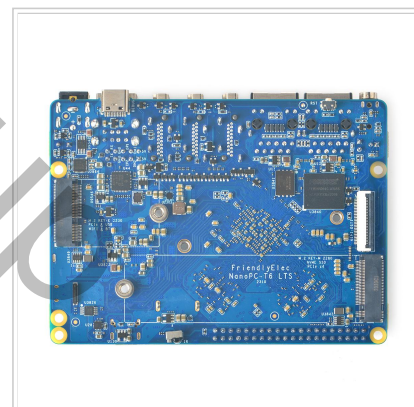
- 1 Introduction
- 2 Hardware Spec
- 3 Diagram, Layout and Dimension
 - 3.1 NanoPC-T6-LTS
 - 3.2 NanoPC-T6
 - 3.3 Layout
- 4 Get Started
 - 4.1 Essentials You Need
 - 4.2 TF Cards We Tested
 - 4.3 Configure parameters for serial port
 - 4.4 Install OS
 - 4.4.1 Downloads
 - 4.4.1.1 Official image
 - 4.4.1.2 Tools (optional)
 - 4.4.2 Flashing the OS to the microSD card
 - 4.4.3 Install OS to eMMC
 - 4.4.3.1 Option 1: Install OS via TF Card
 - 4.4.3.1.1 Flash Official OS to eMMC
 - 4.4.3.1.2 Flash third party OS (Image file) to eMMC
 - 4.4.3.2 Option 2: Install OS on Web Page
 - 4.4.3.3 Option 3: Install OS via USB
 - 4.4.3.3.1 Step 1: Install USB Driver and Tools/Utilities
 - 4.4.3.3.2 Step 2: Connect NanoPC-T6 to PC and Enter Installation Mode
 - 4.4.3.3.3 Step 3: Install image to eMMC
 - 4.4.4 Installing the System to M.2 or USB Drive
- 5 Work with FriendlyWrt
 - 5.1 Introduction to FriendlyWrt
 - 5.2 First boot
 - 5.3 Account & Password
 - 5.4 Login FriendlyWrt
 - 5.5 Recommended security settings
 - 5.6 Change LAN IP in LuCI
 - 5.7 Safe shutdown operation
 - 5.8 Soft Factory Reset
 - 5.9 Install Software Packages
 - 5.9.1 Set up openwrt official opkg source
 - 5.9.2 Update Package List
 - 5.9.3 List Available Packages
 - 5.9.4 List Installed Packages
 - 5.9.5 Install Packages
 - 5.9.6 Remove Packages
 - 5.10 Disable IPv6
 - 5.11 Configure the function of the user button
 - 5.12 Configuring Quectel EC20 (4G module) dial-up networking
 - 5.13 Some common issues of FriendlyWrt
 - 5.14 Use USB2LCD to view IP and temperature
 - 5.15 How to use M.2 WiFi
 - 5.15.1 AP Mode
 - 5.15.2 Wireless Repeater Mode
 - 5.16 How to use USB WiFi
 - 5.16.1 Check USB WiFi Device with Command Line Utility
 - 5.16.2 Configure a USB WiFi Device as AP
 - 5.16.3 Common USB WiFi issues
 - 5.16.4 Change the default WiFi hotspot configuration
 - 5.17 Work with Docker Applications
 - 5.17.1 Work with Docker: Install JellyFin
 - 5.17.2 Work with Docker: Install Personal Nextcloud
 - 5.17.3 Expand Docker Storage
 - 5.17.4 Docker FAQ and solutions
 - 5.17.4.1 Unable to access the network services provided by the Docker container
 - 5.18 Mount smbfs
 - 5.19 Use sdk to compile the package
 - 5.19.1 Install the compilation environment
 - 5.19.2 Download and decompress sdk from the network disk
 - 5.19.3 Compile the package
 - 5.19.4 Install the ipk to NanoPi
 - 5.20 Build FriendlyWrt using GitHub Actions
- 6 Work with Ubuntu 24.04 (Noble) Desktop
 - 6.1 Introduction to Ubuntu 24.04 Desktop
 - 6.2 Account & Password
 - 6.3 Install Ubuntu software center
 - 6.4 Install the kernel-header package
 - 6.5 Install the kernel-image package
 - 6.6 Change time zone



Overview



Front



Back



Case

- 6.6.1 Check the current time zone
 - 6.6.2 List all available time zones
 - 6.6.3 Set the time zone (e.g. Shanghai)
 - 6.7 Change startup LOGO and Wallpaper
 - 6.7.1 Change startup LOGO
 - 6.8 Soft Factory Reset
 - 6.9 Start the program automatically at startup(For example Firefox)
 - 6.10 Disable auto-mounting
 - 6.11 Setup Chinese language and Input method
 - 6.11.1 Setup Chinese language
 - 6.12 Video playback with hardware decoding
 - 6.12.1 GUI
 - 6.12.2 Command line
 - 6.13 Using USB Camera
 - 6.14 Install Docker Engine
 - 6.14.1 Install Docker Engine
 - 6.14.2 Run Docker as a non-root user
 - 6.14.3 Testing Docker: Installing Nextcloud with docker
 - 6.15 Disable Automatic Login
 - 6.16 WiFi Connection
 - 6.16.1 Gui
 - 6.16.2 Console
 - 6.17 Test OpenGL ES
 - 6.18 Chromium web browser
 - 6.18.1 GPU
 - 6.18.2 VPU
 - 6.18.3 Check Supported Hardware Decoding Formats
- 7 Work with Android
 - 7.1 WiFi models supported by Android
 - 7.1.1 M.2 WiFi Module
 - 7.1.1.1 Usb Dongle
 - 7.2 Bluetooth models supported by Android
 - 7.2.1 Bluetooth Adapters
 - 7.2.2 Bluetooth Remote
 - 7.3 How to use ADB
 - 7.3.1 USB connection
 - 7.3.2 For Android Tablet
 - 7.3.3 For Android TV
 - 7.3.4 Using ADB
 - 7.3.5 Using ADB over the network
 - 7.4 How to Change Default Launcher in Android TV
 - 7.5 Wired networks on Android
 - 7.6 EC20 4G LTE module on Android
- 8 Work with Debian11 Desktop
 - 8.1 Introduction to Debian11 Desktop
 - 8.2 Account & Password
 - 8.3 View IP address
 - 8.4 Connect to Debian via SSH
 - 8.5 Update Software Packages
 - 8.6 Install x11vnc Server on Debian for Remote Access
 - 8.6.1 Install x11vnc server
 - 8.6.2 Set your password
 - 8.6.3 Setup x11vnc server with systemd auto start up
 - 8.6.4 Testing remote access
 - 8.7 Install the kernel-header package
 - 8.8 Change time zone
 - 8.8.1 Check the current time zone
 - 8.8.2 List all available time zones
 - 8.8.3 Set the time zone (e.g. Shanghai)
 - 8.9 Change startup LOGO and Wallpaper
 - 8.9.1 Change startup LOGO
 - 8.9.2 Change Wallpaper
 - 8.10 Soft Factory Reset
 - 8.11 Start the program automatically at startup(For example Kodi)
 - 8.12 Disable auto-mounting
 - 8.13 Setup Chinese language and Input method
 - 8.13.1 Setup Chinese language
 - 8.13.2 Installing Chinese input method
 - 8.14 Installing Plex Multimedia Server
 - 8.15 Install Docker on Debian
 - 8.16 How to test NPU
 - 8.17 How to test VPU
 - 8.18 WiFi Connection
 - 8.18.1 Gui
 - 8.18.2 Console
 - 8.19 Cancel auto-login
 - 8.20 Test OpenGL ES
 - 8.21 HDMI/DP LCD Resolution
 - 8.22 HiDPI and display scaling
 - 8.23 Adjust HDMI overscan
 - 8.23.1 Query which resolutions the display supports
 - 8.23.2 Set resolution
 - 8.23.3 Adjust the HDMI overscan
 - 8.23.4 Automatic adjustment at boot
 - 8.24 Chromium web browser
 - 8.24.1 GPU
 - 8.24.2 VPU



Case



Case

- 8.25 Test hardware encoding
 - 8.25.1 Check Supported Hardware Decoding Formats
- 8.26 How to test HDMI-IN
 - 8.26.1 Using script
 - 8.26.2 Using V4L2
 - 8.26.3 Using GStreamer
- 9 Work with Debian10 Desktop
- 10 Work with Ubuntu 20.04 (Focal) Desktop
 - 10.1 Introduction to Ubuntu 20.04 Desktop
 - 10.2 Account & Password
 - 10.3 Turn off auto-sleep and screen saver
 - 10.4 Set up auto login
 - 10.5 Test Video Playback
 - 10.6 Audio Settings
 - 10.6.1 Switching Audio Output Channel
 - 10.6.2 Multi-Device Audio Output
- 11 Work with FriendlyCore
 - 11.1 FriendlyCore User Account
 - 11.2 Update Software Packages
 - 11.3 Setup Network Configurations
 - 11.3.1 Set static IP address
 - 11.3.2 Set a DNS
 - 11.3.3 Set up to use another network interface
 - 11.4 Setup Wi-Fi
 - 11.4.1 WiFi models supported
 - 11.4.1.1 M.2 WiFi Module
 - 11.4.1.2 Usb Dongle
 - 11.5 Install the kernel-header package
 - 11.6 Config status LEDs
 - 11.7 Delete Qt5 and related files
- 12 Installing Third-Party Armbian System
- 13 How to Compile
 - 13.1 Setup Development Environment
 - 13.1.1 Method 1: Using docker to cross-compile
 - 13.1.2 Method 2: Setup build environment on the host machine
 - 13.1.2.1 Install required packages
 - 13.1.2.2 Setting the compiler path
 - 13.2 Build Openwrt/Friendlywrt
 - 13.2.1 Download Code
 - 13.2.1.1 FriendlyWrt 24.10
 - 13.2.1.2 FriendlyWrt 23.05
 - 13.2.2 First compilation step
 - 13.2.3 Secondary compilation steps
 - 13.2.4 Build u-boot only
 - 13.2.5 Build kernel only
 - 13.2.6 Build friendlywrt only
 - 13.3 Build Buildroot
 - 13.4 Build Other Linux
 - 13.4.1 Kernel and u-boot versions
 - 13.4.2 Build kernel linux-v6.1.y
 - 13.4.3 Build u-boot v2017.09
 - 13.4.4 Running the build
 - 13.4.4.1 Install to target board
 - 13.4.4.2 Packaging and creating an SD image
 - 13.4.4.3 USB flashing
 - 13.4.4.3.1 Linux
 - 13.5 Build the code using scripts
 - 13.5.1 Download scripts and image files
 - 13.5.2 Compile the kernel
 - 13.5.3 Compile the kernel headers
 - 13.5.4 Compile the uboot
 - 13.5.5 Generate new image
 - 13.6 Building AOSP from source
 - 13.6.1 Hardware and Software Requirements
 - 13.6.2 Download source from the netdrive
 - 13.6.3 Tablet profile build (First Build)
 - 13.6.4 TV profile build (First Build)
 - 13.6.5 Second build
 - 13.6.6 Running your AOSP build
 - 13.6.6.1 USB Flashing
 - 13.6.6.2 SD-to-eMMC Flashing
 - 13.6.7 Pack the new SD Image
- 14 Using On-Board Hardware Resources
 - 14.1 Using VPU
 - 14.2 Using NPU
 - 14.3 DTS files
- 15 Backup rootfs and create custom SD image (to burn your application into other boards)
 - 15.1 Backup rootfs
 - 15.2 Making a bootable SD card from a root filesystem
- 16 Connect NVME SSD High Speed Hard Disk
 - 16.1 Detection of SSD
 - 16.2 Partition of SSD
 - 16.3 Format Section to EXT4
 - 16.4 Auto Mount SSD on System Startup
- 17 Common Linux-based operating system operations
 - 17.1 Using ADB on Linux Systems
 - 17.1.1 Enabling ADB in Buildroot System

- 17.1.2 Enabling ADB in Ubuntu and Debian Systems
 - 17.1.3 How to Connect
 - 17.2 Install Kernel Headers
 - 17.3 Update Kernel to Latest Version
 - 17.4 Setting Kernel Boot Parameters (eMMC Only)
- 18 Unbricking Method
 - 18.1 Windows Users
 - 18.1.1 Download Required Files
 - 18.1.2 Enter Maskrom Mode to Erase the Storage Device
 - 18.2 Linux Users
 - 18.2.1 Download the Required Files
 - 18.2.2 Installation for upgrade_tool
 - 18.2.3 Enter Maskrom Mode to Erase the Storage Device
 - 18.3 Mac Users
- 19 Link to Rockchip Resources
- 20 Schematic, PCB CAD File
- 21 Update Logs
 - 21.1 2025-04-18
 - 21.2 2025-01-23
 - 21.3 2025-01-17
 - 21.3.1 Linux Kernel
 - 21.3.2 Debian/Ubuntu/OpenMediaVault/ProxmoxVE
 - 21.4 2024-11-12
 - 21.5 2024-10-16
 - 21.5.1 New System
 - 21.5.2 Linux Kernel
 - 21.5.3 Linux System
 - 21.5.4 Others
 - 21.6 2024-06-13
 - 21.6.1 Android
 - 21.7 2024-05-22
 - 21.7.1 Linux Kernel
 - 21.8 2024-05-14
 - 21.8.1 Debian/Ubuntu Desktop
 - 21.9 2024-05-11
 - 21.9.1 Linux Kernel
 - 21.9.2 Debian11 Desktop
 - 21.9.3 Ubuntu Focal Desktop
 - 21.9.4 OpenMediaVault
 - 21.9.5 Other
 - 21.10 2024-04-21
 - 21.10.1 OpenMediaVault
 - 21.11 2024-03-15
 - 21.11.1 Ubuntu focal desktop
 - 21.11.2 Debian/Ubuntu/FriendlyCore
 - 21.12 2024-01-31
 - 21.12.1 Debian/Ubuntu/FriendlyCore/Buildroot
 - 21.12.2 Android 12 & Android TV
 - 21.12.3 FriendlyWrt
 - 21.13 2024-01-24
 - 21.13.1 Buildroot
 - 21.14 2024-01-17
 - 21.14.1 Linux
 - 21.14.2 Android 12
 - 21.15 2023-12-23
 - 21.15.1 Android 12 & Android TV
 - 21.15.2 FriendlyWrt
 - 21.16 2023-12-01
 - 21.16.1 FriendlyWrt
 - 21.17 2023-11-13
 - 21.17.1 Debian/Ubuntu/FriendlyCore
 - 21.17.2 Android 12
 - 21.18 2023-10-31
 - 21.18.1 Add a new system
 - 21.18.2 Debian/Ubuntu/FriendlyCore
 - 21.18.3 FriendlyWrt
 - 21.19 2023-09-09
 - 21.19.1 Android 12
 - 21.20 2023-08-15
 - 21.21 2023-07-19
 - 21.21.1 Ubuntu Focal Desktop
 - 21.21.2 Ubuntu Jammy Desktop
 - 21.21.3 Debian/Ubuntu/Android
 - 21.21.4 Ubuntu Jammy X11 Desktop
 - 21.22 2023-07-01
 - 21.22.1 Debian11
 - 21.23 2023-06-25
 - 21.23.1 Debian11
 - 21.23.2 Android TV & Android 12
 - 21.23.3 FriendlyCore
 - 21.24 2023-06-16
 - 21.24.1 Debian11
 - 21.25 2023-06-09
 - 21.25.1 FriendlyWrt
 - 21.25.2 Ubuntu22
 - 21.26 2023-06-06
 - 21.26.1 Android TV & Android 12

- 21.27 2023-06-01
 - 21.27.1 Android TV
- 21.28 2023-05-27
 - 21.28.1 Ubuntu22
- 21.29 2023-05-26
 - 21.29.1 Ubuntu22
 - 21.29.2 FriendlyWrt
- 21.30 2023-05-21
 - 21.30.1 Debian11
- 21.31 2023-05-15
 - 21.31.1 FriendlyCore Focal
- 21.32 2023-05-05
 - 21.32.1 Android
- 21.33 2023-04-26
 - 21.33.1 FriendlyWrt:
- 21.34 2023-02-10
 - 21.34.1 Android update
 - 21.34.2 Added Debian11
- 21.35 2023-01-09
 - 21.35.1 New OS:
 - 21.35.2 Android:
 - 21.35.3 Ubuntu jammy:
- 21.36 2022-12-13
 - 21.36.1 Ubuntu jammy:
- 21.37 2022-12-04
 - 21.37.1 FriendlyWrt:
- 21.38 2022-11-17
 - 21.38.1 FriendlyWrt:
 - 21.38.2 Ubuntu jammy desktop:
- 21.39 2022-10-25

1 Introduction

NanoPC-T6 LTS (as "T6") is a one-for-all high performance open source platform for edge computing, designed and developed by the FriendlyElec team.

It has two 2.5G Ethernet ports. It is based on Rockchip's RK3588, and has 4GB/8GB/16G LPDDR4x RAM and an optional 32GB/64GB/256/0GB eMMC flash. It work FriendlyWrt, Android, Android TV, Debian, Ubuntu etc. It supports GPU and VPU acceleration.

T6 is a compact board of 110 x 80 mm with rich hardware resources and an optional CNC metal case. It has 2 HDMI output ports and 1 HDMI IN port. T6 can play video streams including 8K60p H.265/VP9 , 8K30p H264 etc and record video streams including 4K60p H.265. T6 has one M.2 B-Key slot that supports an M.2 NVME SSD disk, one M.2 E-Key that supports an M.2 2230 WiFi module.

In addition, T6 has one USB 3.0 and two USB 2.0 ports, and one full-featured USB-C port that is powered by DC-12V.

In summary, T6 is well suited for enterprise customers to develop mini machine vision systems with multiple Ethernet ports and for embedded system hobbyists to expl

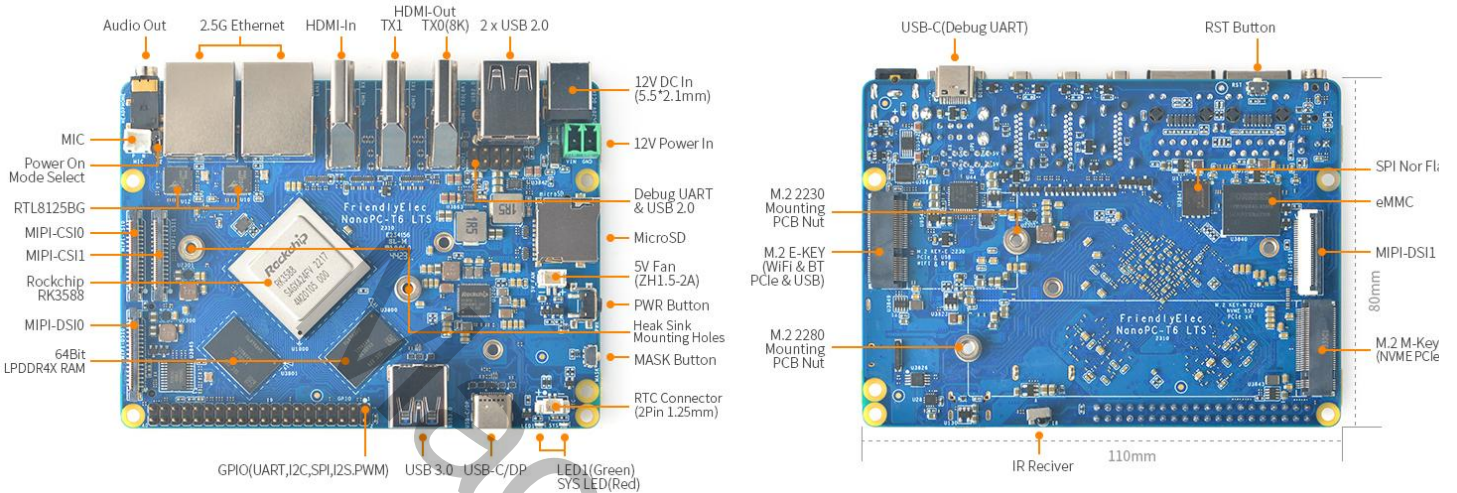
2 Hardware Spec

- SoC: Rockchip RK3588
 - 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 OpenGL ES 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/16GB LPDDR4X at 2133MHz
- eMMC Flash: 32GB/64GB/256GB eMMC, at HS400 mode
- Nor Flash: 32MB SPI Nor Flash
- microSD: support up to SDR104 mode
- Ethernet: 2 x PCIe 2.5G Ethernet
- 1 x USB 3.0 Type-A,
- 1 x Full function USB Type-C™ port, support DP display up to 4Kp60, USB 3.0
- 2 x USB 2.0 Type-A
- Video input:
 - 1 x Standard HDMI input port, up to 4Kp60
 - 2 x 4-lane MIPI-CSI, compatible with MIPI V1.2
- Video output:
 - 2 x Standard HDMI output ports
 - compatible with HDMI2.1, HDMI2.0, and HDMI1.4 operation
 - one support displays up to 7680x4320@60Hz, another one support up to 4Kp60
 - Support RGB/YUV(up to 10bit) format
 - 2 x 4-lane MIPI-DSI, compatible with MIPI DPHY 2.0 or CPHY 1.1
- Audio:
 - 1 x 3.5mm jack for stereo headphone output
 - 1 x 2.0mm PH-2A connector for analog microphone input
- GPIO:
 - 40-pin 2.54mm header connector
 - up to 2 x SPIs, 6 x UARTs, 1 x I2Cs, 8 x PWMs, 2 x I2Ss, 28 x GPIOs
- M.2 Connectors
 - one M.2 M-Key connector with PCIe 3.0 x4 for NVMe SSDs up to 2,500 MB/s
 - one M.2 E-key connector with PCIe 2.1 x1 and USB2.0 Host
- others:
 - 10 Pin 2.54mm header connector including Debug UART(3.3V TTL) and 2 x USB 2.0 Host
 - 2 Pin 1.27/1.25mm RTC battery input connector for low power RTC IC HYM8563TS

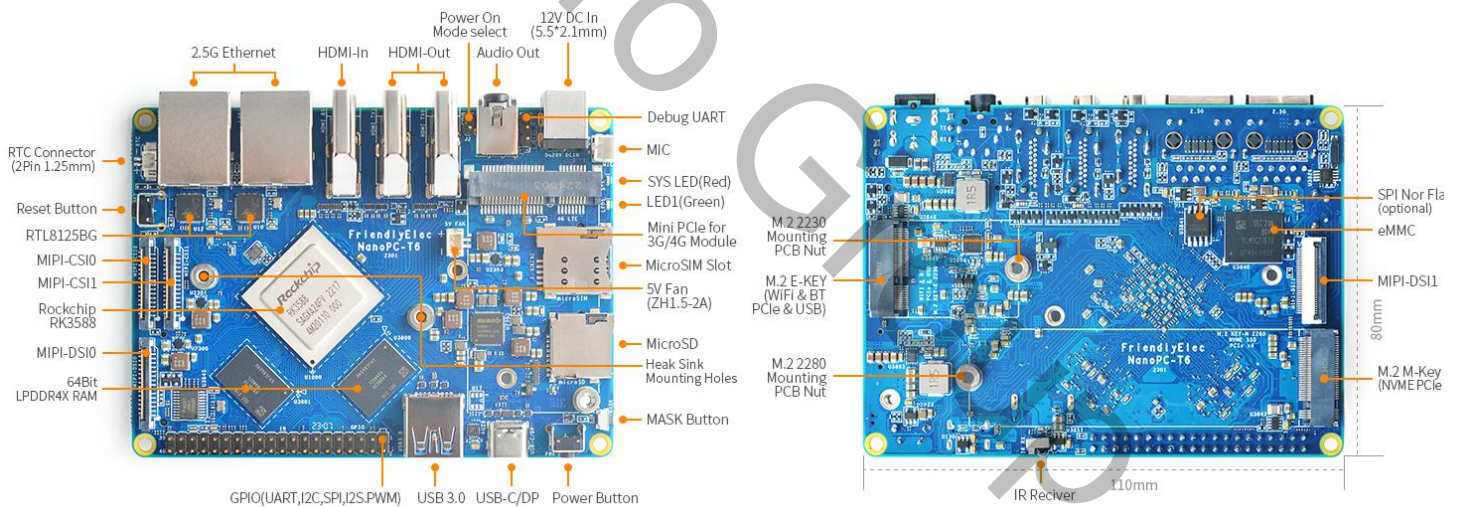
- one 38Khz IR receiver
- MASK button for eMMC update, reset button, and Power button
- one 5V Fan connector
- USB-C to Debug UART
- 2 x GPIO Controlled LED (SYS, LED1)
- Power supply: 5.5*2.1mm DC Jack & 2-Pin 3.5mm pitch connector, 12VDC input.
- PCB: 8 Layer, 110x80x1.6mm
- Ambient Operating Temperature: 0°C to 70°C

3 Diagram, Layout and Dimension

3.1 NanoPC-T6-LTS



3.2 NanoPC-T6



3.3 Layout

- 40-pin GPIO

Pin#	GPIO	SPI	UART	I2C	I2S	PWM	POWER	Descrip
1,17							VCC3V3_SYS_S3	3.3V Po Output, 500mA Max
2,4							VCC_5V0	5V Pow Output, 500mA Max
6,9,14,20,25,30,34,39							GND	Power a Signal Ground
3	GPIO1_D7			I2C8_SDA_M2				pulled u 3.3V wi 2.2K on
5	GPIO1_D6			I2C8_SCL_M2				pulled u 3.3V wi 2.2K on
7	GPIO3_B2				I2S2_SDI_M1			3.3V
8	GPIO0_C5		UART0_TX_M0			PWM4_M0		3.3V
10	GPIO0_C4		UART0_RX_M0					3.3V
11	GPIO3_C2					PWM14_M0		3.3V
12	GPIO3_B7							3.3V
13	GPIO3_C3					PWM15_IR_M0		3.3V
15	GPIO1_A7							3.3V
16	GPIO3_B3				I2S2_SDO_M1			3.3V
18	GPIO3_B4				I2S2_MCLK_M1			3.3V
19	GPIO1_B2	SPI0_MOSI_M2	UART4_RX_M2					3.3V
21	GPIO1_B1	SPI0_MISO_M2						3.3V
22	GPIO1_B5	SPI0_CS1_M0	UART7_TX_M2					3.3V
23	GPIO1_B3	SPI0_CLK_M2	UART4_TX_M2					3.3V
24	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2					3.3V
26	GPIO1_B0							3.3V
27	GPIO1_A0		UART6_RX_M1					3.3V
28	GPIO1_A1		UART6_TX_M1					3.3V
29	GPIO3_B5		UART3_TX_M1		I2S2_SCLK_M1	PWM12_M0		3.3V
31	GPIO3_B6		UART3_RX_M1		I2S2_LRCK_M1	PWM13_M0		3.3V
32	GPIO0_C6					PWM5_M1		3.3V
33	GPIO3_B0					PWM9_M0		3.3V
35	GPIO3_A0	SPI4_MISO_M1			I2S3_MCLK	PWM10_M0		3.3V
36	GPIO3_A3	SPI4_CS0_M1	UART8_RX_M1		I2S3_SDO			3.3V
37	GPIO3_A4	SPI4_CS1_M1			I2S3_SDI			3.3V
38	GPIO3_A1	SPI4_MOSI_M1			I2S3_SCLK			3.3V
40	GPIO3_A2	SPI4_CLK_M1	UART8_TX_M1		I2S3_LRCK			3.3V

■ MIPI-DSI

0.5mm FPC Connector

Pin#	MIPI-DSI0	MIPI-DSI1	Description
1,2,3	VCC_5V0	VCC_5V0	5V Power output
4,7,9,11,15,18,21,24,27,30	GND	GND	Power and Signal Ground
5	I2C5_SDA_M0	I2C4_SDA_M3	3.3V, I2C Data, pulled up to 3.3V with 2.2K on T6
6	I2C5_SCL_M0	I2C4_SCL_M3	3.3V, I2C Clock, pulled up to 3.3V with 2.2K on T6
8	GPIO3_C0	GPIO4_A0	3.3V, GPIO
10	GPIO3_B1/PWM2_M1	GPIO3_D5/PWM11_M3	3.3V, GPIO/PWM
12	GPIO3_A6	GPIO4_A3	3.3V, GPIO
13	/NC	/NC	No Connection
14	GPIO3_C1	GPIO4_A1	3.3V, GPIO
16	MIPI_DPHY0_TX_D3N	MIPI_DPHY1_TX_D3N	MIPI TX Lane3 output N
17	MIPI_DPHY0_TX_D3P	MIPI_DPHY1_TX_D3P	MIPI TX Lane3 output P
19	MIPI_DPHY0_TX_D2N	MIPI_DPHY1_TX_D2N	MIPI TX Lane2 output N
20	MIPI_DPHY0_TX_D2P	MIPI_DPHY1_TX_D2P	MIPI TX Lane2 output P
22	MIPI_DPHY0_TX_D1N	MIPI_DPHY1_TX_D1N	MIPI TX Lane1 output N
23	MIPI_DPHY0_TX_D1P	MIPI_DPHY1_TX_D1P	MIPI TX Lane1 output P
25	MIPI_DPHY0_TX_D0N	MIPI_DPHY1_TX_D0N	MIPI TX Lane0 output N
26	MIPI_DPHY0_TX_D0P	MIPI_DPHY1_TX_D0P	MIPI TX Lane0 output P
28	MIPI_DPHY0_TX_CLKN	MIPI_DPHY1_TX_CLKN	MIPI TX Clock output N
29	MIPI_DPHY0_TX_CLKP	MIPI_DPHY1_TX_CLKP	MIPI TX Clock output P

■ MIPI-CSI

0.5mm FPC Connector

Pin#	MIPI-CSI0	MIPI-CSI1	Description
1,2	VCC_5V0	VCC_5V0	5V Power output
3,13,15,18,21,24,27,30	GND	GND	Power and Signal Ground
4,5,7	/NC	/NC	No Connection
6	VCC_1V8_S3	VCC_1V8_S3	1.8V Power output, 100mA Max
8	VSYNC_MASTER	VSYNC_SLAVE	Have been Connected together on T6 for sensor synchronization
9	I2C3_SCL_M0	I2C7_SCL_M0	1.8V, I2C Clock, pulled up to 1.8V with 2.2K on T6
10	I2C3_SDA_M0	I2C7_SDA_M0	1.8V, I2C Data, pulled up to 1.8V with 2.2K on T6
11	GPIO4_C4	GPIO2_C1	1.8V, GPIO
12	GPIO4_C5	GPIO2_C2	1.8V, GPIO
14	MIPI_CAM1_CLKOUT	MIPI_CAM2_CLKOUT	1.8V, Clock output for Sensor
16	MIPI_CSI0_RX_D3P	MIPI_CSI1_RX_D3P	MIPI RX Lane3 input P
17	MIPI_CSI0_RX_D3N	MIPI_CSI1_RX_D3N	MIPI RX Lane3 input N
19	MIPI_CSI0_RX_D2P	MIPI_CSI1_RX_D2P	MIPI RX Lane2 input P
20	MIPI_CSI0_RX_D2N	MIPI_CSI1_RX_D2N	MIPI RX Lane2 input N
22	MIPI_CSI0_RX_D1P	MIPI_CSI1_RX_D1P	MIPI RX Lane1 input P
23	MIPI_CSI0_RX_D1N	MIPI_CSI1_RX_D1N	MIPI RX Lane1 input N
25	MIPI_CSI0_RX_CLK0P	MIPI_CSI1_RX_CLK0P	MIPI RX Clock input P
26	MIPI_CSI0_RX_CLK0N	MIPI_CSI1_RX_CLK0N	MIPI RX Clock input N
28	MIPI_CSI0_RX_D0P	MIPI_CSI1_RX_D0P	MIPI RX Lane0 input P
29	MIPI_CSI0_RX_D0N	MIPI_CSI1_RX_D0N	MIPI RX Lane0 input N

■ 10 Pin Debug UART & USB 2.0 Header

Debug UART is 3.3V level signals, 1500000bps

Pin#	Assignment	Description	Pin#	Assignment	Description
1	USB 5V	5V Power Output(1A Max)	2	USB 5V	5V Power Output(1A Max)
3	USB_HS2_DM	USB 2.0 Host DM	4	USB_HS3_DM	USB 2.0 Host DM
5	USB_HS2_DP	USB 2.0 Host DP	6	USB_HS3_DP	USB 2.0 Host DP
7	GND	0V	8	GND	0V
9	UART2_TX_M0_DEBUG	Debug UART TX	10	UART2_RX_M0_DEBUG	Debug UART RX

■ About Power

- Power supply via 5.5*2.1mm DC Jack. 5V~20V, 12V is recommended.
- Power Output Capacity

Port	Max Output	Port	Max Output
USB-A 3.0	5V/2A	USB-C/DP	5V/2A
M.2 M-Key	3.3V/3A	M.2 E-Key	3.3V/3A
MIPI-CSI0	5V/0.5A	MIPI-CSI1	5V/0.5A
MIPI-DSI0	5V/1A	MIPI-DSI1	5V/1A
40 Pin GPIO	5V/0.5A, 3.3V/0.5A	10 Pin USB2.0	5V/2A
USB-A 2.0	5V/2A	Total	35W

▪ Debug UART Pin Spec (On the old T6 2301)

3.3V level signals, 1500000bps

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

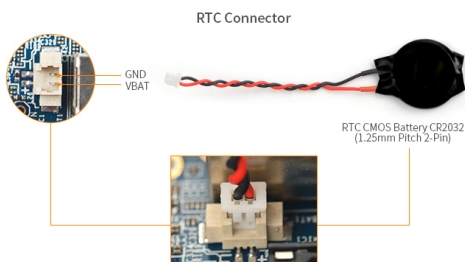
▪ About Power (For the old T6 2301)

- Power supply via 5.5*2.1mm DC Jack. 5V~20V, 12V is recommended.
- Power Output Capacity

Port	Max Output	Port	Max Output
USB-A 3.0	5V/2A	USB-C/DP	5V/2A
M.2 M-Key	3.3V/3A	M.2 E-Key	3.3V/3A
MIPI-CSI0	5V/0.5A	MIPI-CSI1	5V/0.5A
MIPI-DSI0	5V/1A	MIPI-DSI1	5V/1A
GPIO	5V/0.5A, 3.3V/0.5A	miniPCIe	3.3V/3A
Total	35W		

▪ RTC

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



▪ IR receiver

Connected to PWM3_IR_M0
38Khz carrier frequency
compatible with NEC protocol, User code is 3B4C
Support FriendlyELEC RC100 IR controller

4 Get Started

4.1 Essentials You Need

Before starting to use your NanoPC-T6 get the following items ready

- NanoPC-T6
- MicroSD Card/TF Card: Class 10 or Above, minimum 8GB SDHC
- A DC 12V/2A power
- 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 Configure parameters for serial port

Use the following serial parameters:

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











4.4 Install OS

4.4.1 Downloads

4.4.1.1 Official image

Visit download link (<http://download.friendlyelec.com/NanoPC-T6>) 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-bookworm-core-6.1-arm64-YYYYMMDD.img.gz	bookworm	Debian 12 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 browser, 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 browser, supports HW acceleration	6.1.y
	rk3588-XYZ-ubuntu-noble-desktop-6.1-arm64-YYYYMMDD.img.gz	noble	Ubuntu 24.04 with GNOME and Wayland with recommended software	6.1.y
	rk3588-XYZ-ubuntu-noble-minimal-6.1-arm64-YYYYMMDD.img.gz	noble	Lightweight Ubuntu 24.04 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-android14-YYYYMMDD.img.zip	14	Android 14 Tablet	6.1.y
	rk3588-XYZ-androidtv-YYYYMMDD.img.zip	12	Android 12 TV	5.10.y
	rk3588-XYZ-android12-YYYYMMDD.img.zip	12	Android 12 Tablet	5.10.y
	rk3588-XYZ-openmediavault-6.1-YYYYMMDD.img.gz	Shaitan	OpenMediaVault NAS system, base on Debian 12	6.1.y
	rk3588-XYZ-proxmox-6.1-YYYYMMDD.img.gz	8.2.7	Proxmox VE OS (Preview), base on Debian 12	6.1.y
	rk3588-XYZ-friendlywrt-24.10-YYYYMMDD.img.gz	24.10	FriendlyWrt, based on OpenWrt 24.10	6.1.y
	rk3588-XYZ-friendlywrt-24.10-docker-YYYYMMDD.img.gz	24.10	FriendlyWrt with Docker, based on OpenWrt 24.10	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				
	FriendlyWrt (Github Actions)	24.10,23.05	FriendlyWrt (https://github.com/friendlyarm/Actions-FriendlyWrt/releases)	6.1.y
	Alpine-Linux (Github Actions)	-	Alpine-Linux (https://github.com/friendlyarm/Actions-Alpine-Linux/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, this image disables automatic flashing at startup; you will need to manually select the OS to flash.	

4.4.1.2 Tools (optional)

Visit download link (<http://download.friendlyelec.com/NanoPC-T6/>) 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

4.4.2 Flashing the OS to the microSD card

Follow the steps below:

- Get an 8G microSD card;
- Visit download link (<http://download.friendlyelec.com/NanoPC-T6/>) 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 NanoPC-T6's microSD card slot;
- Power on NanoPC-T6 and it will be booted from your TF card, some models may require pressing the Power button to start;

4.4.3 Install OS to eMMC

4.4.3.1 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)
Power On	Solid On
System Boot	Slow Flashing
Installation in Progress	Fast Flashing
Installation Done	Slow Flashing

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.

4.4.3.1.1 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/NanoPC-T6/>) 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 NanoPC-T6's microSD card slot.
- Turn on NanoPC-T6, 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 NanoPC-T6, NanoPC-T6 will automatically reboot and boot from eMMC.

4.4.3.1.2 Flash third party OS (Image file) to eMMC

- Auto Install (Default Behavior)

- 1) Download an "eflasher" firmware from network drive (<http://download.friendlyelec.com/NanoPC-T6/>) (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 the image file ending with .raw or .gz into the directory (Note: if your file is in .img format, please rename it to .raw format).
- 3) Open the eflasher.conf file on the TF card, set "autoStart=" to the name of your image file, such as:

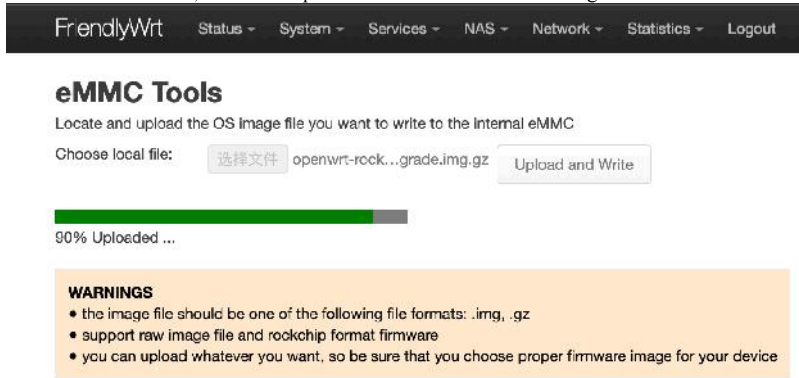
```
autoStart=openwrt-rockchip-armv8_nanopi-ext4-sysupgrade.img.gz
```

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-24.10-YYYYMMDD.img.gz

- 4) Eject the TF card, insert the TF card to NanoPC-T6, power it on it will automatically install your firmware. You can watch the installation progress by observing the LEDs' status.

4.4.3.2 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-24.10-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.

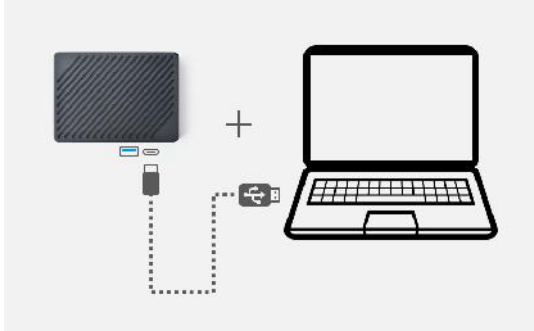
4.4.3.3 Option 3: Install OS via USB

4.4.3.3.1 Step 1: Install USB Driver and Tools/Utilities

Download a driver file DriverAssitant_v5.12.zip under the “tools” directory from network drive (<http://download.friendlyelec.com/NanoPC-T6>), extract and install it. Under the same directory, download a utility RKDevTool_Release_v2.84.zip and extract it.

4.4.3.3.2 Step 2: Connect NanoPC-T6 to PC and Enter Installation Mode

- 1) 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;
- 2) Use a USB C-to-A cable, connect NanoPC-T6 to a PC;



4.4.3.3.3 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

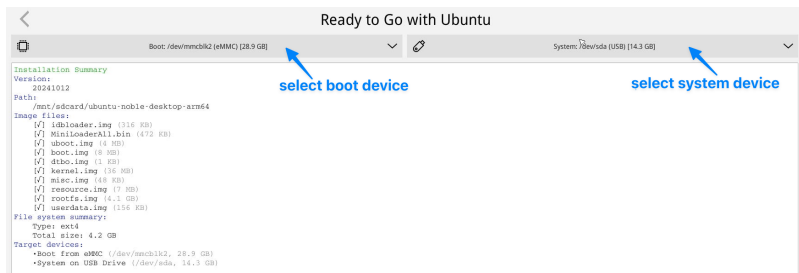
Go to network drive (<http://download.friendlyelec.com/NanoPC-T6>) to download your needed package and extract it (in the "01_Official images/03_USB upgrade images). After it is extracted, you will see some utilities and a configuration file under the directory. double click on RKDevTool.exe, you will 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.

4.4.4 Installing the System to M.2 or USB Drive

You can use a TF card to boot the eFlasher system, allowing the boot and system to be installed on different storage devices. However, since the CPU doesn't support booting directly from M.2 and USB devices, the system can be installed on M.2 and USB devices, but the boot must still be installed on eMMC or a TF card. Steps are as follows:

- Prepare a TF card with a capacity of 32GB or larger.
- Visit [the download link here](<http://download.friendlyelec.com/NanoPC-T6>) to download the firmware file named XXXX-eflasher-multiple-os-YYYYMMDD-30g.img.gz (located in the “01_Official images/02_SD-to-eMMC images” directory).
- Flash the firmware to the TF card, connect the storage device you intend to use on NanoPC-T6, insert the TF card and power on, we need to perform the operations in the eFlasher GUI. If your NanoPC-T6 does not have a display interface, you can use VNC; refer to Using VNC to Operate eFlasher (https://wiki.friendlyelec.com/wiki/index.php/EFlasher#Remote_Control_eFlasher_via_VNC).

- In the eFlasher GUI, select the OS to install, and in the OS settings interface, choose the destination for boot installation (typically eMMC), then choose the destination for system installation (options include eMMC, M.2 hard drive, USB storage, etc.), as shown below:



- If no eMMC is available, the TF card can serve as the boot by inserting another TF card into the USB port via a USB card reader and selecting it as the boot destination, enabling booting from the TF card with the system stored on the M.2 or USB drive.
- After flashing, eject the SD card from NanoPC-T6. If booting from eMMC, NanoPC-T6 will automatically restart into the newly flashed system. If boot installation is on a TF card, power off, insert the boot TF card, and power on again.
- For a more detailed installation guide, please refer to this link (https://wiki.friendlyelec.com/wiki/index.php/EFlasher#Installing_the_System_to_M.2_or_USB_Drive_.28Rockchip_Platform_Only.29).

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 NanoPC-T6 to the Internet.

5.4 Login FriendlyWrt

Connect the PC to the LAN port of NanoPC-T6. If your PC without a built-in ethernet port, connect the LAN port of the wireless AP to the LAN port of NanoPC-T6, 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://\[fd00:ab:cd::1\]](http://[fd00:ab:cd::1])

The above is the LAN port address of NanoPC-T6. 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 NanoPC-T6 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/triggershappy/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 NanoPC-T6 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 of NanoPC-T6 and power on, the IP address and CPU temperature will be displayed on the LCD:



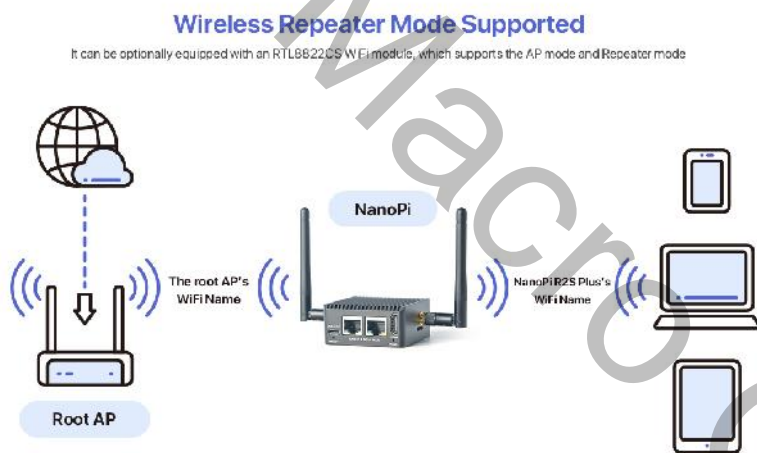
5.15 How to use M.2 WiFi

- M.2 WiFi models supported: rtl8822ce

5.15.1 AP Mode

- FriendlyWrt's wireless function is disabled by default, to enable WiFi, you can click "Network > Wireless", and then click the "Enable" button.
- Search for a WiFi hotspot with a name like FriendlyWrt-xx:yy:zz on your phone, enter the default password "password" to connect.

5.15.2 Wireless Repeater Mode



- Connect to the wireless router

Click "Network" -> "Wireless", Click the "Scan" button next to "Generic MAC80211 802.11ac/b/g/n" to scan for networks, select the router you want to connect and then click the "Join Network" button,
In the network joining configuration page, turn on "Replace wireless configuration", enter the key in the "WPA passphrase" field, keep other configurations as default, click "Submit",
Click "Advanced Settings", input "wlan0" in "Interface Name" field, and then click "Save",
Finally, click "Save and Apply", if normal, you can see the IP address on the FriendlyWrt homepage.

- Create wireless hotspot

Click "Network" -> "Wireless", Click the "Add" button next to "Generic MAC80211 802.11ac/b/g/n", the "Edit wireless network" page will pop up,
Set the wireless channel in the "Operating frequency" field (e.g., "40 (5200 MHz)"),
In the "ESSID" field, input the hotspot name,
In the "Network" field, select "lan",
Click on "Wireless Security", set the encryption type,
Click "Advanced Settings", input "wlan1" in "Interface Name" field, then click "Save" and "Save and Apply",
It requires a reboot to take effect, click on "System" menu, then select "Reboot".

5.16 How to use USB WiFi

5.16.1 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.

5.16.2 Configure a USB WiFi Device as AP

(1) Connect a USB WiFi device to the NanoPC-T6. 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 NanoPC-T6

(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

5.16.3 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 WiFi (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.

5.16.4 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:

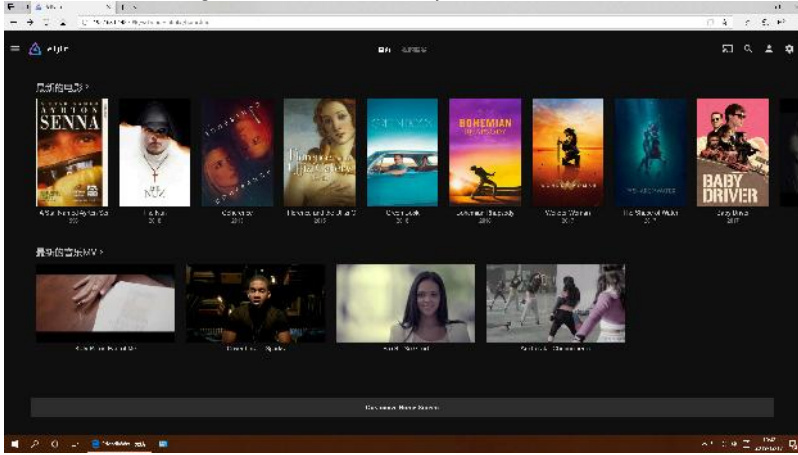
```
/lib/wifi/mac80211.sh
```

5.17 Work with Docker Applications

5.17.1 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:



5.17.2 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 installation, visit port 8888.

5.17.3 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:

Docker - Overview

An overview with the relevant data is displayed here with which the LuCI docker client is connected.

Info	
Docker Version	20.10.12
Api Version	1.41
CPUs	4
Total Memory	1.91 GB
Docker Root Dir	/opt/docker (220.71 GB Available)
Index Server Address	https://index.docker.io/v1/

5.17.4 Docker FAQ and solutions

5.17.4.1 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";

5.18 Mount smbfs

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

5.19 Use sdk to compile the package

5.19.1 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>)

5.19.2 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
```

5.19.3 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
```


5.19.4 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
```

5.20 Build FriendlyWrt using GitHub Actions

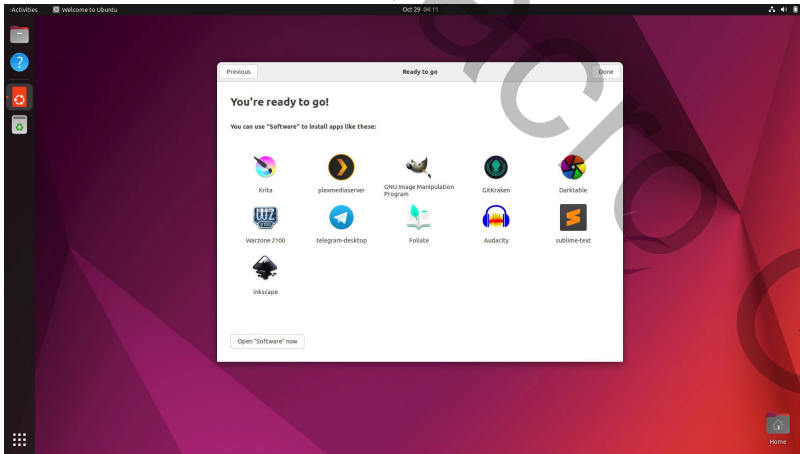
Please refre this link: <https://github.com/friendlyarm/Actions-FriendlyWrt>

6 Work with Ubuntu 24.04 (Noble) Desktop

6.1 Introduction to Ubuntu 24.04 Desktop

Ubuntu 24.04 Desktop has the following features:

- Uses GNOME 46 as default desktop;
- Uses Wayland as default display server;
- Mali GPU-based OpenGL ES support;
- Support Rockchip 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 Install Ubuntu software center

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

6.4 Install the kernel-header package

```
sudo dpkg -i /opt/archives/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
```

6.5 Install the kernel-image package

You can visit <http://112.124.9.243/archives/> to check if there is a new version of the kernel deb package (compare with the current version in the /opt/archives/ directory). If a new version is available, download it locally and install it using dpkg -i.

6.6 Change time zone

6.6.1 Check the current time zone

```
timedatectl
```

6.6.2 List all available time zones

```
timedatectl list-timezones
```

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

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

6.7 Change startup LOGO and Wallpaper

6.7.1 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-noble-desktop-arm64
./mk-emmc-image.sh ubuntu-noble-desktop-arm64
```

6.8 Soft Factory Reset

Execute the following command in a terminal:

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

6.9 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/
```

6.10 Disable auto-mounting

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

6.11 Setup Chinese language and Input method

6.11.1 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
```

6.12 Video playback with hardware decoding

6.12.1 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)

6.12.2 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=1K-nYDmC1Dk
```

6.13 Using USB Camera

(Note: The following test commands use the Logitech HD Pro Webcam C920.)

- List camera devices

```
$ v4l2-ctl --list-devices
```

- Display supported formats

```
$ v4l2-ctl -d /dev/video1 --list-formats-ext --all
```

- Preview

```
$ export XDG_RUNTIME_DIR=/run/user/0
$ gst-launch-1.0 v4l2src device=/dev/video1 ! image/jpeg,width=1280,height=720,framerate=30/1 ! jpegdec ! videoconvert ! glimagesink
```

- Record video

```
$ gst-launch-1.0 v4l2src device=/dev/video1 ! image/jpeg,width=1280,height=720,framerate=30/1 ! \
  jpegdec ! videoconvert ! queue ! mpegtsenc ! h265parse ! \
  queue ! mux. alsasrc device=hw:CARD=realtekrt5616co ! audio/x-raw,channels=2 \
  ! audioconvert ! voaacenc ! queue ! mux. matroskamux name=mux ! filesink location="demo.mkv"
```

6.14 Install Docker Engine

6.14.1 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 /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Let's verify:

```
sudo docker info
```

6.14.2 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
```

6.14.3 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.

6.15 Disable Automatic Login

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

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

6.16 WiFi Connection

6.16.1 Gui

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

6.16.2 Console

Please visit: Use NetworkManager to configure network settings

6.17 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'
```

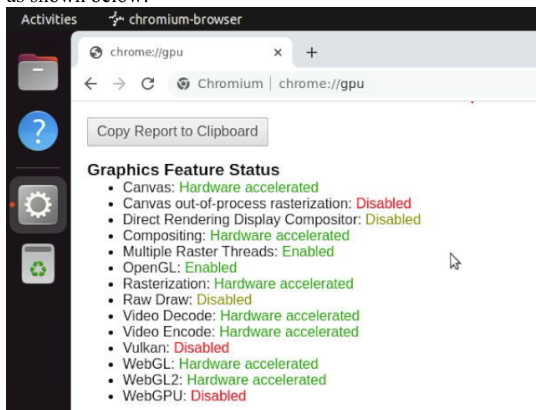
Start glmark2-es2-wayland:

```
glmark2-es2-wayland
```

6.18 Chromium web browser

6.18.1 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:



6.18.2 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.

6.18.3 Check Supported Hardware Decoding Formats

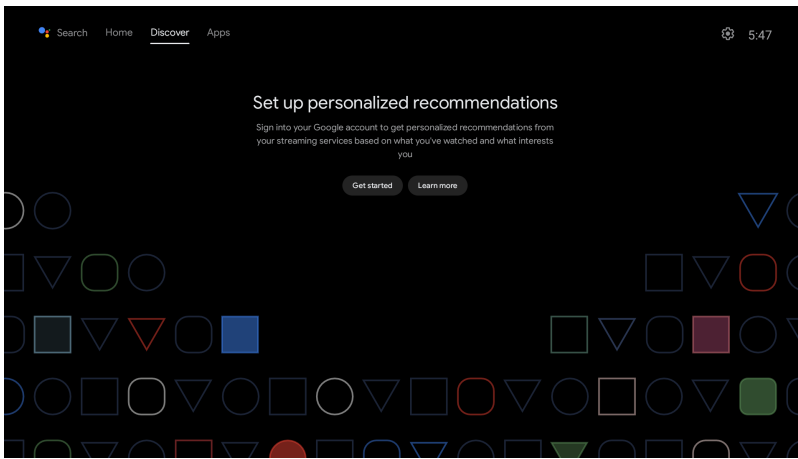
Enter about://gpu in your browser's address bar and scroll to the bottom of the page to view the "Video Acceleration Information" table.

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 Bluetooth models supported by Android

7.2.1 Bluetooth Adapters

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

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, and then reboot;
- To use ADB over the network, you need to connect to WiFi first, then enable Wireless debugging. In the prompt "Allow wireless debugging on the network", select "Always allow on this network", and then click "Allow".

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, and then reboot;
- To use ADB over the network, on Android TV, which supports both WiFi and wired networks, enable network ADB by checking "Internet ADB" on the "Developer options" UI.

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:/ $
```

7.3.5 Using ADB over the network

- On an Android Tablet, go to Settings -> System -> Advance -> Developer options, then click on Wireless debugging to view the IP address and port.
- The default network ADB port for Android TV is 5555.

Assuming the IP address and port displayed on the Wireless debugging interface are 192.168.1.167:45055, the ADB commands are as follows:

- To connect to the device:

```
$ adb connect 192.168.1.167:45055
```

connected to 192.168.1.167:45055

- To enter ADB shell:

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

If there are multiple devices, use the -s parameter to specify the device's IP and port, as shown below:

```
$ adb -s 192.168.1.167:45055 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
```

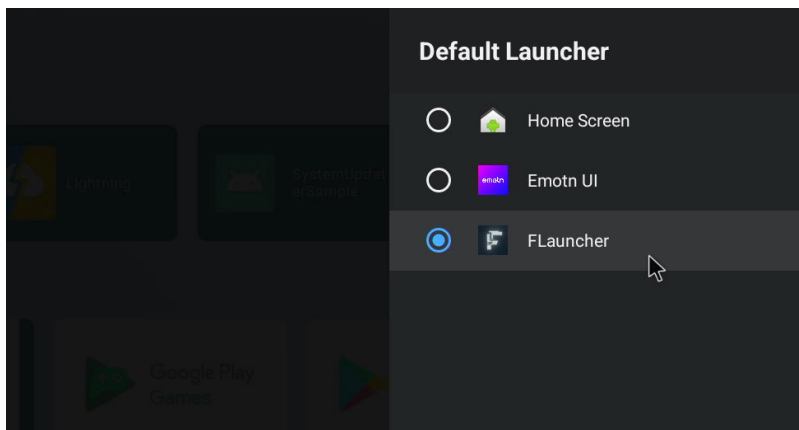
- Then comes the critical step, you need to disable the native launcher using the following command:

```
$ 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):

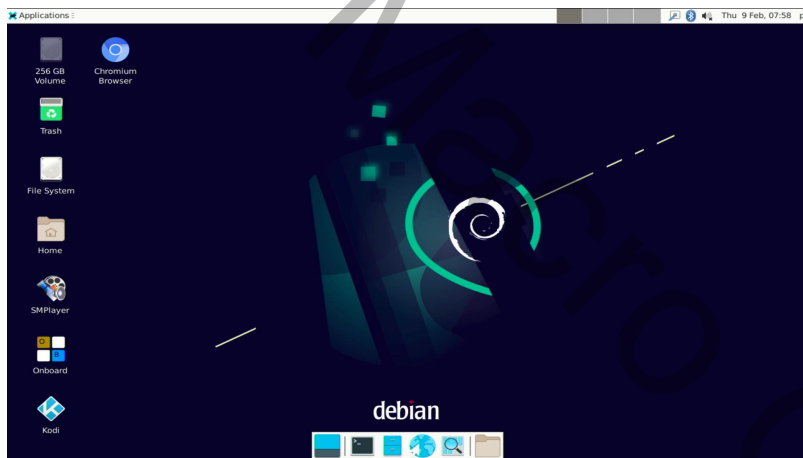
```
su
setprop persist.vendor.radio.no_modem_board 0
```

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 Rockchip 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 NanoPC-T6

8.4 Connect to Debian via SSH

Run the following command: ssh pi@NanoPC-T6

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 systemd 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/archives/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 repack 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:

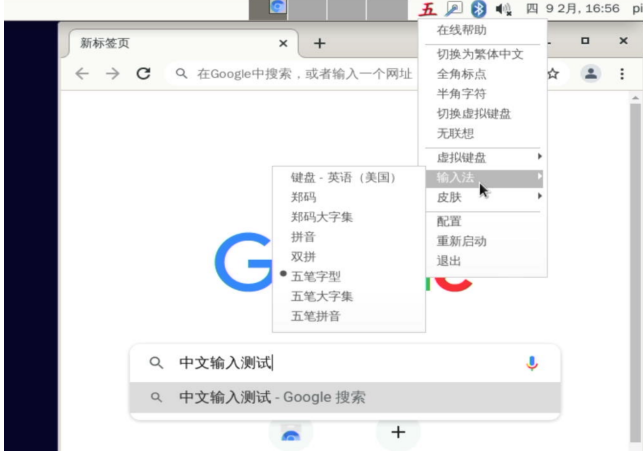
```
sudo reboot
```

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-frontends-gtk2 fcitx-frontends-gtk3 fcitx-frontends-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

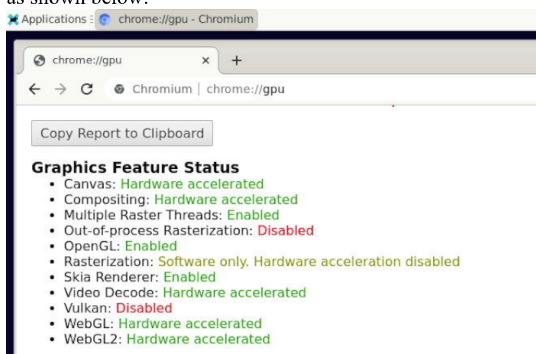
Edit `~/config/autostart/lxrandr-autostart.desktop`, Write the full xrandr command to the key at the beginning of "Exec=" as shown below :

```
[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.

8.26 How to test HDMI-IN

8.26.1 Using script

The debian integrates the `hdmirx_preview.sh` test script, just run the script directly:

```
pi@NanoPC-T6:~$ hdmirx_preview.sh
```

the script path: `/usr/local/bin/hdmirx_preview.sh`

Please note that only the NV12 format can achieve optimal performance.

8.26.2 Using V4L2

The node of HDMI-IN device is: `/dev/video0`, which can be operated by the `v4l2-ctl` command.

- Obtain device information

```
pi@NanoPC-T6:~$ v4l2-ctl -d /dev/video0 -V -D
Driver Info:
  Driver name      : rk_hdmirx
  Card type       : rk_hdmirx
  Bus info        : fdee0000.hdmirx-controller
  Driver version  : 5.10.110
  Capabilities    : 0x84201000
                   Video Capture Multiplanar
                   Streaming
                   Extended Pix Format
                   Device Capabilities
  Device Caps     : 0x04201000
                   Video Capture Multiplanar
                   Streaming
                   Extended Pix Format
Format Video Capture Multiplanar:
  Width/Height    : 3840/2160
  Pixel Format     : 'NV12' (Y/CbCr 4:2:0)
  Field           : None
  Number of planes : 1
  Flags           : premultiplied-alpha, 0x000000fe
  Colorspace      : Unknown (0x11507070)
  Transfer Function : Unknown (0x000000b8)
  YCbCr/HSV Encoding: Unknown (0x000000ff)
  Quantization    : Default
  Plane 0        :
  Bytes per Line  : 3840
  Size Image      : 12441600
```

- View the resolution and image format of the currently connected device

```
pi@NanoPC-T6:~$ v4l2-ctl -d /dev/video0 --get-fmt-video
Format Video Capture Multiplanar:
  Width/Height    : 3840/2160
  Pixel Format     : 'NV12' (Y/CbCr 4:2:0)
  Field           : None
  Number of planes : 1
  Flags           : premultiplied-alpha, 0x000000fe
  Colorspace      : Unknown (0x1193b008)
  Transfer Function : Unknown (0x000000b8)
  YCbCr/HSV Encoding: Unknown (0x000000ff)
  Quantization    : Default
  Plane 0        :
  Bytes per Line  : 3840
  Size Image      : 12441600
```

- Capture a frame

```
pi@NanoPC-T6:~$ v4l2-ctl -d /dev/video0 --set-fmt-video=width=3840,height=2160,pixelformat='NV12' \
--stream-mmap=4 --stream-skip=10 --stream-to=/home/pi/4k_nv12.yuv --stream-count=1 \
--stream-poll
```

- Preview

use `ffplay`:

```
export DISPLAY=:0.0
ffplay -f rawvideo -video_size 3840x2160 -pixel_format nv12 4k_nv12.yuv
```

use `mpv`:

```
export DISPLAY=:0.0
mpv 4k_nv12.yuv --demuxer=rawvideo --demuxer-rawvideo-w=3840 --demuxer-rawvideo-h=2160 \
--demuxer-rawvideo-mp-format=nv12 --demuxer-rawvideo-fps=60
```

- View HDMI-IN audio device

```
pi@NanoPC-T6:~$ cat /proc/asound/card*
0 [rockchipdp0 ]: rockchip_dp0 - rockchip,dp0
rockchip,dp0
1 [rockchiphdmi0]: rockchip_hdmi0 - rockchip,hdmio
rockchip,hdmio
2 [realtekrt5616co]: realtek_rt5616 - realtek,rt5616-codec
```

```

realtek,rt5616-codec
3 [rockchiphdmi1 ]: rockchip_hdmi1 - rockchip_hdmi1
rockchip_hdmi1
4 [rockchiphdmiin]: rockchip_hdmiin - rockchip_hdmiin
rockchip_hdmiin

```

As you can see, the sound card number of HDMI-IN is 4.

- Recording audio (recording 10 seconds)

```
pi@NanoPC-T6:~$ arecord -D hw:4,0 -f cd test.wav -d 10
```

- Playback recorded audio (output to HDMI0)

```
pi@NanoPC-T6:~$ aplay test.wav -D 'hw:rockchiphdmi0'
```

8.26.3 Using GStreamer

- Live Preview

```

pi@NanoPC-T6:~$ export DISPLAY=:0.0
pi@NanoPC-T6:~$ gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=NV12,width=3840,height=2160,framerate=60/1 \
! queue ! xvimagesink

```

- Audio and video recording

```

pi@NanoPC-T6:~$ gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=NV12,width=3840,height=2160,framerate=60/1 \
! queue ! mpeg4enc ! h265parse ! queue ! mux. alsasrc device=hw:CARD=rockchiphdmiin ! audio/x-raw,channels=2 \
! audioconvert ! voaacenc ! queue ! mux. matroskamux name=mux ! filesink location="4kp60.mkv"

```

- Live preview + audio and video recording

```

pi@NanoPC-T6:~$ export DISPLAY=:0.0
pi@NanoPC-T6:~$ gst-launch-1.0 -e v4l2src device=/dev/video0 ! 'video/x-raw,format=NV12,width=3840,height=2160' \
! tee name=t. ! mpeg4enc bps=20000000 bps-max=40000000 rc-mode=vbr ! h265parse ! mp4mux name=mux \
! filesink location=4k60.mp4 alsasrc device=hw:CARD=rockchiphdmiin ! opusenc ! mux. t. ! queue leaky=1 ! autovideosink sync=false

```

9 Work with Debian10 Desktop

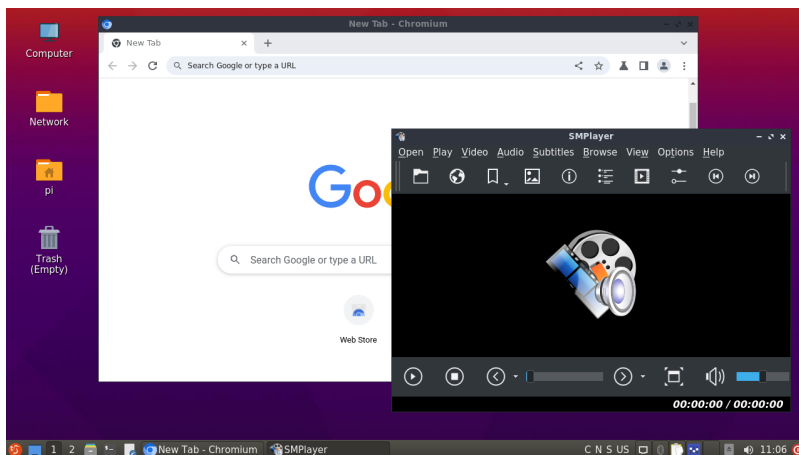
- Refer to:
 - Debian Buster

10 Work with Ubuntu 20.04 (Focal) Desktop

10.1 Introduction to Ubuntu 20.04 Desktop

Ubuntu 20.04 Desktop has the following features:

- Base on Xorg, using LXQT as the desktop environment;
- SDDM is the default display manager;
- Mali GPU-based OpenGL ES support;
- Support Rockchip MPP video hard coding and hard decoding;
- Pre-installed mpv, support video hardware decoding;



10.2 Account & Password

Regular Account:

User Name: pi
Password: pi

Root:

User Name: root
Password: fa

10.3 Turn off auto-sleep and screen saver

```
sed -i 's/enableIdlenessWatcher=true/enableIdlenessWatcher=false/g' .config/lxqt/lxqt-powermanagement.conf
echo "mode: off" > ~/.xscreensaver
sudo reboot
```

10.4 Set up auto login

Create dir and file:

```
sudo mkdir /etc/sddm.conf.d
sudo vim /etc/sddm.conf.d/autologin
```

The contents of the autologin file :

```
[Autologin]
User=pi
Session=lxqt.desktop
Relogin=false
```

Reboot.

10.5 Test Video Playback

- mpv

```
mpv ~/Videos/demo.mp4
```

- GStreamer Command Line

```
gst-play-1.0 ~/Videos/demo.mp4 --videosink=ximagesink
```

For more details, please search and refer to the official documentation <<rockchip_user_guide_linux_gstreamer_en.pdf>>.

10.6 Audio Settings

10.6.1 Switching Audio Output Channel

To switch the audio output channel, go to "System Tools" -> "Qterminal", enter the following command in the terminal to check available audio output channels:

```
export XDG_RUNTIME_DIR=/run/user/1000
pactl list sinks short
```

On the NanoPC-T6, the results appear as follows. The string beginning with alsa_output in the second column is the value we will use:

0	alsa_output.platform-dp0-sound.stereo-fallback	module-alsa-card.c	s16le	2ch	44100Hz	SUSPENDED
1	alsa_output.platform-hdmi0-sound.stereo-fallback	module-alsa-card.c	s16le	2ch	48000Hz	SUSPENDED
2	alsa_output.platform-hdmi1-sound.stereo-fallback	module-alsa-card.c	s16le	2ch	44100Hz	SUSPENDED
3	alsa_output.platform-rt5616-sound.multichannel-output	module-alsa-card.c	s16le	2ch	44100Hz	SUSPENDED

Use an editor to modify the PulseAudio configuration file:

```
sudo vim /etc/pulse/default.pa
```

If you need to set the audio output to HDMI1, replace the following line at the end of the file:

```
set-default-sink alsa_output.platform-hdmi0-sound.stereo-fallback
```

with:

```
set-default-sink alsa_output.platform-hdmi1-sound.stereo-fallback
```

Save the file and restart the system for the changes to take effect, or restart the PulseAudio service:

```
pulseaudio -k
pulseaudio --start
```

10.6.2 Multi-Device Audio Output

Edit the PulseAudio configuration file:

```
sudo vim /etc/pulse/default.pa
```

Scroll to the end of the file, delete any line that starts with set-default-sink, and add the following lines at the end:

```
load-module module-combine-sink sink_name=combined slaves=alsa_output.platform-hdmi0-sound.stereo-fallback,alsa_output.platform-hdmi1-sound.stereo-fallback
set-default-sink combined
```


Restart the system. With this configuration, audio will output simultaneously to both the HDMI0 and HDMI1 interfaces.

11 Work with FriendlyCore

11.1 FriendlyCore User Account

- Non-root User:

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

- Root:

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

11.2 Update Software Packages

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

11.3 Setup Network Configurations

11.3.1 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

11.3.2 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
```

11.3.3 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.

11.4 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

11.4.1 WiFi models supported

11.4.1.1 M.2 WiFi Module

- RTL8822CE

11.4.1.2 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)

11.5 Install the kernel-header package

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

11.6 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/usr_led/trigger
echo eth0 > /sys/class/leds/usr_led/device_name
echo 1 > /sys/class/leds/usr_led/link
```

11.7 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}
```

12 Installing Third-Party Armbian System

Please refer to: Armbian Installation Guide

13 How to Compile

13.1 Setup Development Environment

13.1.1 Method 1: Using docker to cross-compile

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

13.1.2 Method 2: Setup build environment on the host machine

13.1.2.1 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

13.1.2.2 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 ".":

```
source ~/.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/
Thread model: posix
Supported LTO compression algorithms: zlib
gcc version 11.3.0 (ctng-1.25.0-119g-FA)
```

13.2 Build Openwrt/Friendlywrt

13.2.1 Download Code

Two versions are available, please choose as required:

13.2.1.1 FriendlyWrt 24.10

```
mkdir friendlywrt24-rk3588
cd friendlywrt24-rk3588
git clone https://github.com/friendlyarm/repo --depth 1 tools
tools/repo init -u https://github.com/friendlyarm/friendlywrt_manifests -b master-v24.10 \
-m rk3588.xml --repo-url=https://github.com/friendlyarm/repo --no-clone-bundle
tools/repo sync -c --no-clone-bundle
```

13.2.1.2 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
```

13.2.2 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
```

13.2.3 Secondary compilation steps

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

13.2.4 Build u-boot only

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

13.2.5 Build kernel only

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

13.2.6 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
```

13.3 Build Buildroot

please refer to: Buildroot

13.4 Build Other Linux

13.4.1 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/friendlyrockchip/tree/nanopi5-v5.10.y_opt)
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/friendlyrockchip/tree/nanopi6-v6.1.y)
ubuntu-noble-desktop-arm64						
ubuntu-noble-minimal-arm64						
ubuntu-jammy-x11-desktop-arm64						
ubuntu-focal-desktop-arm64						
friendlycore-focal-arm64						
debian-bookworm-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.

13.4.2 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 refer to #Running the build

13.4.3 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/u-boot-rockchip --single-branch --depth 1 -b nanopi6-v2017.09
export PATH=/opt/FriendlyARM/toolchain/11.3-aarch64/bin/:$PATH
cd u-boot-rockchip/
./make.sh nanopi6
```

After the compilation, the following files will be generated:

u-boot.img	rk3588_spl_loader_xx.yy.zzz.bin (aka MiniLoaderAll.bin)
------------	---

Run your build:

Please refer to #Running the build

13.4.4 Running the build

13.4.4.1 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=u-boot.img of=/dev/mmcblk2p1 bs=1M
```

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

13.4.4.2 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)

13.4.4.3 USB flashing

13.4.4.3.1 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).

13.5 Build the code using scripts

13.5.1 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 xvfz /path/to/netdrive/03_Partition\ image\ files/friendlycore-focal-arm64-images.tgz
```

13.5.2 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
```

13.5.3 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
```

13.5.4 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
```

13.5.5 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
```

13.6 Building AOSP from source

13.6.1 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>).

13.6.2 Download source from the netdrive

Netdisk URL: Click here (<http://download.friendlyelec.com/NanoPC-T6>)

File location on netdisk: "07_Source codes/rk35xx-android12-xxxxxx-YYYYMMDD.tgz" (YYYYMMDD represents the date of the package, and xxxxxx represents the final commit-id)

Unzip and fetch updates:

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

13.6.3 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
```

13.6.4 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
```

13.6.5 Second build

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

13.6.6 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.

13.6.6.1 USB Flashing

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

13.6.6.2 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 NanoPC-T6 and re-flash;

13.6.7 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

14 Using On-Board Hardware Resources

14.1 Using VPU

Please refer to NPU

14.2 Using NPU

Please refer to NPU

14.3 DTS files

Please refer to DTS files

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

15.1 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.

15.2 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 xvfz /path/to/netdrive/03_Partition\ image\ files/debian-bullseye-desktop-arm64-images.tgz
tar xvfz /path/to/netdrive/03_Partition\ image\ files/emmc-eflasher-images.tgz
scp pi@BOARDIP:/rootfs.tar.gz /rootfs.tar.gz
mkdir rootfs
tar xvfz rootfs.tar.gz -C rootfs --numeric-owner --same-owner
./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
```

16 Connect NVME SSD High Speed Hard Disk

16.1 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.

16.2 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.

16.3 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
```

16.4 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-8f81-35a8bd46057" 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-8f81-35a8bd46057 /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
```

17 Common Linux-based operating system operations

17.1 Using ADB on Linux Systems

17.1.1 Enabling ADB in Buildroot System

Enable on Startup

```
mv /etc/init.d/K50usbdevice.sh /etc/init.d/S50usbdevice.sh
reboot
```

Enable Temporarily

```
usbdevice-wrapper start
```

17.1.2 Enabling ADB in Ubuntu and Debian Systems

Enable on Startup

```
sudo systemctl enable usbdevice
sudo reboot
```

Enable Temporarily

```
usbdevice-wrapper start
```

17.1.3 How to Connect

When using ADB, the port connected to the computer is the same as the USB flashing port.

17.2 Install Kernel Headers

To install the .deb file located in the /opt/archives directory:

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

To download and update the kernel header files online:

```
wget http://112.124.9.243/archives/rk3588/linux-headers-$(uname -r)-latest.deb
sudo dpkg -i ./linux-headers-latest.deb
```

You can visit <http://112.124.9.243/archives/rk3588> to see which kernel deb packages are available.

17.3 Update Kernel to Latest Version

This will update the kernel to the latest version and update the kernel and resource partitions, as well as the kernel modules:

```
wget http://112.124.9.243/archives/rk3588/linux-image-$(uname -r)-latest.deb
sudo dpkg -i ./linux-image-latest.deb
sudo reboot
```

17.4 Setting Kernel Boot Parameters (eMMC Only)

Flash the firmware file XXXX-eflasher-multiple-os-YYYYMMDD-30g.img.gz to a TF card, then insert the TF card into your computer. Windows will usually recognize the TF card partition automatically (formatted as exFAT). For Linux or Mac users, manually mount the first partition of the TF card. Assuming the TF card's device name is /dev/sdX, mount /dev/sdX1.

Edit the info.conf configuration file in the OS directory on the TF card, adding the bootargs-ext parameter. For example:

```
bootargs-ext=rockchipdrm.fb_max_sz=2048
```

To remove a specified parameter, set it to empty. For example, to remove the userdata parameter:

```
bootargs-ext=userdata=
```

After editing, use this TF card to flash the system to eMMC.

To set kernel boot parameters during the creation of a mass production card, refer to the following script (example for RK3588): https://github.com/friendlyarm/sd-fuse_rk3588/blob/kernel-6.1.y/test/test-custom-bootargs.sh

18 Unbricking Method

If the ROM is not installed correctly, causing the development board to become bricked, and you might not have the opportunity to reinstall the ROM via an SD card, you need to enter Maskrom mode to unbrick it by erasing the storage device.

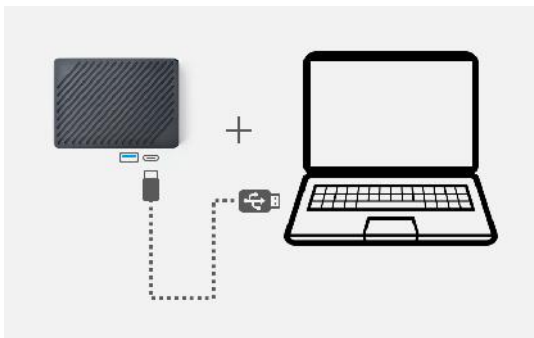
18.1 Windows Users

18.1.1 Download Required Files

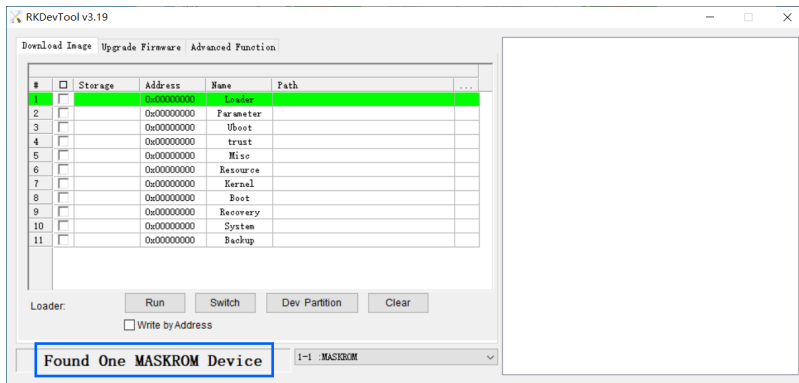
- **Get the necessary tools:** Visit here (<https://dl.friendlyelec.com/NanoPC-T6/>), find **RKDevTool_v3.19_for_window.zip** and **DriverAssitant_v5.12.zip** in the **05_Tools** directory, and download them to your local machine.
- **Install Rockchip USB driver and RKDevTool:** Extract **DriverAssitant_v5.12.zip** to install the Rockchip USB driver, and extract **RKDevTool_v3.19_for_window.zip** to obtain the Rockchip flashing tool **RKDevTool**.
- **Get the loader:** Visit here (<http://112.124.9.243/dvdfiles/>), enter the tools directory corresponding to your CPU model, and download **MiniLoaderAll.bin**.

18.1.2 Enter Maskrom Mode to Erase the Storage Device

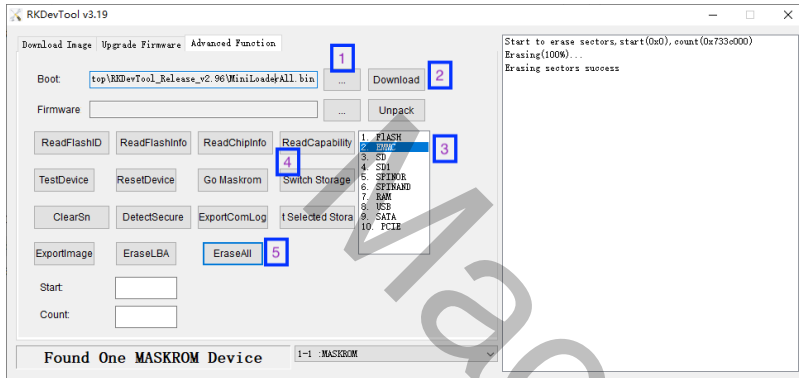
- Remove the SD card, USB device, and other peripherals from the development board
- Start **RKDevTool** on your computer.
- Press and hold the “Mask” key, power on the board. After the status LED has been on for at least 3 seconds, release the Mask key
- Use a USB C-to-A cable, connect NanoPC-T6 to a PC



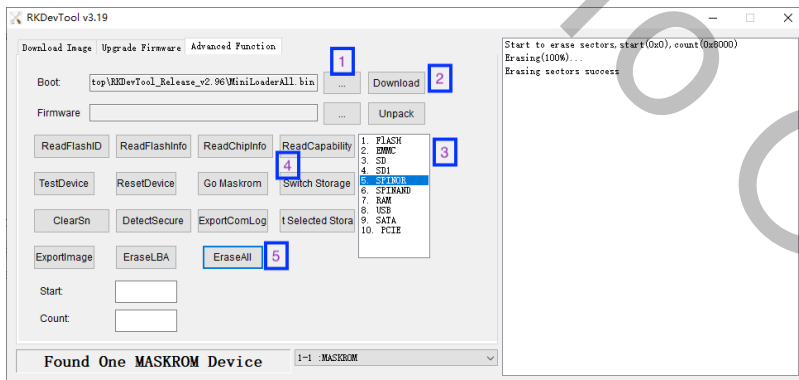
- You will see **Found One MASKROM Device** displayed at the bottom of the **RKDevTool** interface, as shown below:



- Click the **Advanced Function** tab in the **RKDevTool** interface.
- In the **Boot** text box, select **MiniLoaderAll.bin**, then click the **Download** button.
- Select **EMMC**, click **Switch Storage**, then click the **EraseAll** button to erase the eMMC.



- (Optional): If your NanoPC-T6 has SPI Nor Flash, select **SPINOR**, click **Switch Storage**, then click the **EraseAll** button to erase the SPI Nor Flash.



- At this point, NanoPC-T6 is restored to its initial state and can be normally booted via SD card or eMMC.

18.2 Linux Users

18.2.1 Download the Required Files

- Get the necessary tools:** Visit here (<https://dl.friendlyelec.com/NanoPC-T6>) and find **upgrade_tool_v2.30_for_linux.tgz** in the **05_Tools** directory and download it locally.
- Get the loader:** Visit here (<http://112.124.9.243/dvdfiles/>), enter the tools directory corresponding to your CPU model, and download **MiniLoaderAll.bin**.

18.2.2 Installation for upgrade_tool

Using the following commands:

```
tar xzf upgrade_tool_v2.30_for_linux.tgz
cd upgrade_tool_v2.30_for_linux
sudo cp upgrade_tool /usr/local/sbin/
sudo chmod 755 /usr/local/sbin/upgrade_tool
```

18.2.3 Enter Maskrom Mode to Erase the Storage Device

- Connect NanoPC-T6 to the computer using a USB data cable.
- Disconnect the power from NanoPC-T6, hold down the **MASK** button, connect the power, and release the button after 4 seconds.
- Check the connection with the following command:

```
upgrade_tool LD
```

A result similar to "DevNo=1 Vid=0x2207,Pid=0x350b,LocationID=13 Mode=Maskrom SerialNo=" indicates that the device has been detected.

- Erase the eMMC with the following command:

```
upgrade_tool EF MiniLoaderAll.bin
```

- **(Optional):** If your NanoPC-T6 has SPI Nor Flash, erase the SPI Nor Flash with the following commands:

```
upgrade_tool DB MiniLoaderAll.bin
upgrade_tool SSD # Select 5, SPINOR
dd if=/dev/zero of=zero.img bs=1M count=16 # For 16M NOR FLASH
upgrade_tool WL 0 zero.img
```

- At this point, NanoPC-T6 has been restored to its initial state and can boot the system normally via SD card or eMMC.

18.3 Mac Users

Our tests found that upgrade_tool_v2.25 does not work properly on macOS. Therefore, we recommend using Windows or Linux unless an updated version of upgrade_tool becomes available.

19 Link to Rockchip Resources

- Rockchip RK3588 Datasheet V1.7-20231117.pdf (https://wiki.friendlyelec.com/wiki/images/9/95/Rockchip_RK3588_Datasheet_V1.7-20231117.pdf)

20 Schematic, PCB CAD File

- NanoPC-T6 2301 Schematic: https://wiki.friendlyelec.com/wiki/images/9/97/NanoPC-T6_2301_SCH.PDF
- NanoPC-T6 LTS 2310 Schematic: https://wiki.friendlyelec.com/wiki/images/4/44/NanoPC-T6_LTS_2310_SCH.pdf
- NanoPC-T6 2301 PCB CAD File: https://wiki.friendlyelec.com/wiki/images/9/90/NanoPC-T6_2301_DXF.zip
- NanoPC-T6 LTS 2310 PCB CAD File: https://wiki.friendlyelec.com/wiki/images/0/08/NanoPC-T6_LTS_2310_DXF.zip

21 Update Logs

21.1 2025-04-18

- Added Android 14

21.2 2025-01-23

- Update Chromium to the new stable version (130) for Debian 11 and Ubuntu 20 systems
- Add the gstreamer1.0-rockchip package for video hardware encoding and decoding on Ubuntu Noble Desktop (Example: USB_Camera (https://wiki.friendlyelec.com/wiki/index.php/NanoPC-T6#Using_USB_Camera))
- Add udev rules on Ubuntu 24 Core and Debian 12 Core systems to fix permissions for VPU/GPU device nodes

21.3 2025-01-17

21.3.1 Linux Kernel

- Synchronized upstream kernel update to version 6.1.99
- Updated rknpu driver to version v0.9.8 (Document: NPU (<https://wiki.friendlyelec.com/wiki/index.php/NPU>))

21.3.2 Debian/Ubuntu/OpenMediaVault/ProxmoxVE

- Updated Debian Bullseye system to linux-5.10-gen-rkr9 (including updates mpp, xserver, rkaiq, gstreamer-rockchip, rga2, libv4l-rkmpp, libmali, and related packages)
- Updated Rockchip hardware-related packages for Ubuntu Focal Desktop system (same as Debian Bullseye)
- Integrated hardware acceleration packages such as mpp, gstreamer, rknn, and libmali into the Core system without a desktop environment

21.4 2024-11-12

- Improved ramdisk and eflasher for more flexibility and reliability when installing the system on external storage

21.5 2024-10-16

21.5.1 New System

- Added Ubuntu 24.04 Desktop system, using Wayland, supporting VPU/GPU acceleration, and integrated with the new version of Chromium.
- Added Proxmox VE system

21.5.2 Linux Kernel

- KVM is enabled by default in the kernel

21.5.3 Linux System

- Updated Buildroot system to linux-5.10-gen-rkr8
- Updated Debian Bullseye system to linux-5.10-gen-rkr8 (updated packages such as mpp/xserver/rkaiq/gstreamer-rockchip/rga2/libv4l-rkmpp, etc.)
- Updated Rockchip-related software packages in Ubuntu Focal Desktop system (same as Debian Bullseye)
- Upgraded Chromium to the new version (stable version 129), improving performance by 50% compared to the previous version, with support for GPU acceleration and hardware video decoding, as well as HiDPI resolution
- Updated FriendlyWrt to version openwrt-23.05.05

21.5.4 Others

- Eflasher firmware flashing now supports selecting the target storage device, allowing the Linux system root partition to be installed on M.2 SSD or USB drive (but still requires eMMC/TF card for boot)
- Eflasher can customize kernel boot parameters via configuration file
- Updated ramdisk to support btrfs filesystem (can use sd-fuse_rk3568 (https://github.com/friendlyarm/sd-fuse_rk3568) to package btrfs firmware)
- Updated ramdisk to improve compatibility for filesystem repair

21.6 2024-06-13

21.6.1 Android

- Added support for PWM fan
- Improved HDMI output compatibility
- Added support for Intel RealSense Depth Camera, enable it by setting the CONFIG_USB_VIDEO_CLASS_REALSENSE option in the kernel
- Updated support for Android video decoding

21.7 2024-05-22

21.7.1 Linux Kernel

- Fixed the issue where HDMI output at 4K 120p was displaying incorrectly.
- Fixed the issue where only one display might be working when using dual HDMI outputs.
- Fixed the RGA3 issue.
- Modified kernel LSM to add apparmor,selinux.

21.8 2024-05-14

21.8.1 Debian/Ubuntu Desktop

- Fix alsa issue

21.9 2024-05-11

21.9.1 Linux Kernel

- Synchronize upstream kernel updates to 6.1.57.
- Add support for NVMe hardware monitoring, temperature of NVMe devices can be viewed through the /sys/class/hwmon node.
- Introduce support for PWM fans, it can be disabled by unloading the pwm_fan module.
- Refactor rknpu (v0.9.3) as a module for easier updates or version changes.
- Enhance compatibility for HDMI output.
- Introduce a simple driver for querying input power voltage (node: /sys/class/power_supply/simple-vin/voltage_now).
- Add support for Intel RealSense Depth Camera, enable it by configuring the kernel option CONFIG_USB_VIDEO_CLASS_REALSENSE.

21.9.2 Debian11 Desktop

- Synchronize upstream Debian version to linux-5.10-gen-rkr7.1.
- Update core libraries such as libmali, mpp, rga2, npu, gstreamer, etc., to newer versions.
- Update web browser Chromium to a new version.

21.9.3 Ubuntu Focal Desktop

- Add support for multiple languages (including Chinese).
- Fix potential issue of no display when connecting a screen after boot.
- Update core libraries such as libmali, mpp, rga2, npu, gstreamer, etc., to newer versions.

21.9.4 OpenMediaVault

- Improve the compatibility of hard drive, allowing hard drives with the same ID

21.9.5 Other

- sd-fuse_rk3588 (https://github.com/friendlyarm/sd-fuse_rk3588) Support packaged image on arm64 platform

21.10 2024-04-21

21.10.1 OpenMediaVault

- Update to 7.0.5-1
- Update to Debian12

21.11 2024-03-15

21.11.1 Ubuntu focal desktop

- Fix the bluetooth issue

21.11.2 Debian/Ubuntu/FriendlyCore

- Updated uboot: Fixed the problem that DVI mode may not be displayed.
- Updated 6.1 kernel: fixed rknpu not being able to tune and specify different cores

21.12 2024-01-31**21.12.1 Debian/Ubuntu/FriendlyCore/Buildroot**

- Add adb support

21.12.2 Android 12 & Android TV

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

21.12.3 FriendlyWrt

- Add wireless repeater mode support for rtl8822ce

21.13 2024-01-24**21.13.1 Buildroot**

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

21.14 2024-01-17**21.14.1 Linux**

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

21.14.2 Android 12

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

21.15 2023-12-23**21.15.1 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

21.15.2 FriendlyWrt

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

21.16 2023-12-01**21.16.1 FriendlyWrt**

- Update to openwrt-23.05.2

21.17 2023-11-13**21.17.1 Debian/Ubuntu/FriendlyCore**

- Updated npu driver to 0.9.2, fixed known issues

21.17.2 Android 12

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

21.18 2023-10-31**21.18.1 Add a new system**

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

21.18.2 Debian/Ubuntu/FriendlyCore

- Update to kernel 6.1

21.18.3 FriendlyWrt

- Update to kernel 6.1
- Update to openwrt-23.05

21.19 2023-09-09**21.19.1 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)

21.20 2023-08-15

- Add support for Buildroot

21.21 2023-07-19**21.21.1 Ubuntu Focal Desktop**

- Add Ubuntu 20.04 desktop, with LXQT lightweight desktop

21.21.2 Ubuntu Jammy Desktop

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

21.21.3 Debian/Ubuntu/Android

- Fix RTL9210 enclosures slowdown issue

21.21.4 Ubuntu Jammy X11 Desktop

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

21.22 2023-07-01**21.22.1 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

21.23 2023-06-25**21.23.1 Debian11**

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

21.23.2 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)

21.23.3 FriendlyCore

- Fix headphone jack no sound issue

21.24 2023-06-16**21.24.1 Debian11**

- Merge changes from upstream 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

21.25 2023-06-09**21.25.1 FriendlyWrt**

- Add support for MediaTek MT7921 wireless card

21.25.2 Ubuntu22

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

21.26 2023-06-06**21.26.1 Android TV & Android 12**

- Auto expand storage space for SD boot

21.27 2023-06-01**21.27.1 Android TV**

- Fix home button issue

21.28 2023-05-27**21.28.1 Ubuntu22**

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

21.29 2023-05-26**21.29.1 Ubuntu22**

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

21.29.2 FriendlyWrt

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

21.30 2023-05-21**21.30.1 Debian11**

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

21.31 2023-05-15**21.31.1 FriendlyCore Focal**

- Add Qt 5.10 support

21.32 2023-05-05**21.32.1 Android**

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

21.33 2023-04-26**21.33.1 FriendlyWrt:**

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

21.34 2023-02-10**21.34.1 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

21.34.2 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

21.35 2023-01-09**21.35.1 New OS:**

- Android12: Android 12 Tablet

- FriendlyCore: Lightweight version of Linux, based on Ubuntu Core 20.04, no desktop system

21.35.2 Android:

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

21.35.3 Ubuntu jammy:

- optimized the systemd service

21.36 2022-12-13

21.36.1 Ubuntu jammy:

- Fix WiFi issue

21.37 2022-12-04

21.37.1 FriendlyWrt:

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

21.38 2022-11-17

21.38.1 FriendlyWrt:

- Fix 1G LAN port not working issue

21.38.2 Ubuntu jammy desktop:

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

21.39 2022-10-25

Initial Release

Retrieved from "<https://wiki.friendlyelec.com/wiki/index.php?title=NanoPC-T6&oldid=26684>"

-
- This page was last modified on 21 January 2025, at 10:10.