NanoPC-T3 Plus

Contents

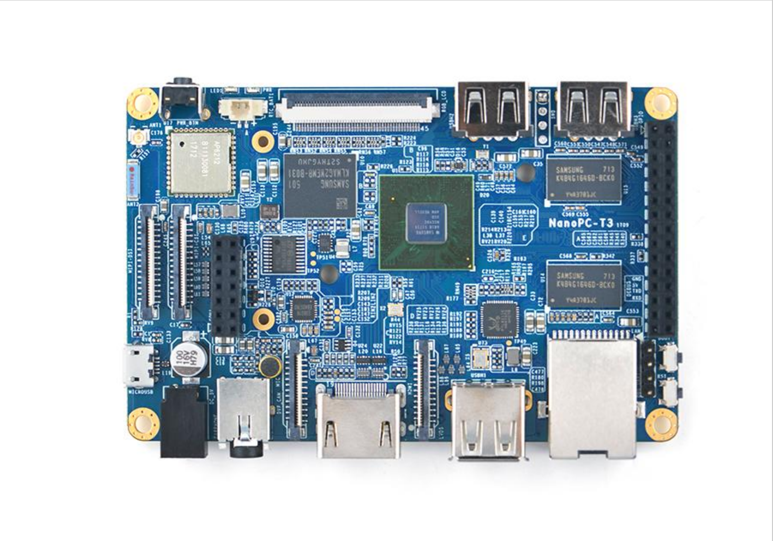
1 Introduction

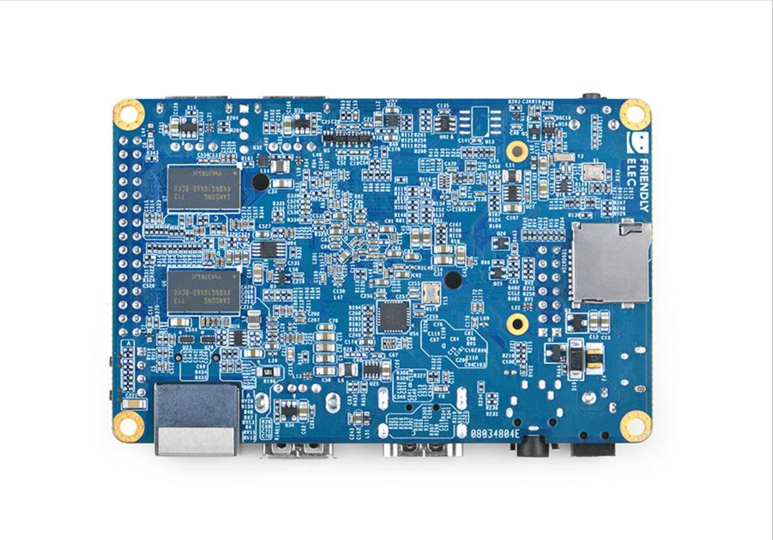
- The NanoPC-T3 Plus octa-core single board computer is designed and developed by FriendlyELEC for professional and enterprise users. It uses the Samsung Octa-Core Cortex-A53 S5P6818 SoC. Compared to the

FriendlyELEC NanoPC-T2 the NanoPC-T3 Plus not only has all the T2's interfaces and ports but also has a more powerful SoC. Its dynamic frequency scales from 400M up to 1.4GHz. The NanoPC-T3 Plus has 16G eMMC onboard, audio jack, video input/output interfaces, built-in WiFi, Bluetooth and Gbps Ethernet port. In addition the NanoPC-T3 Plus) has power management, on board porcelain antenna and serial debug port. To avoid overheat issues the NanoPC-T3 Plus has a heat sink with mounting holes.

- The NanoPC-T3 Plus has two camera interfaces: a DVP camera interface and a MIPI-CSI interface, and four video interfaces: HDMI 1.4A, LVDS, parallel RGB-LCD interface and MIPI-DSI interface. It supports RTC and has RTC interface pins. It has four USB ports with three being type A ports and one being 2.54mm pitch pin-headers.
- The NanoPC-T3 Plus supports muitple OS systems e.g. Android5.1, Debian and UbuntoCore+Qt. It is an open source project with rich interfaces and ports. It is born a choice for professional and enterprise users.

Overview


Front


Back

2 Hardware Spec

- SoC: Samsung S5P6818 Octa-Core Cortex-A53, 400M Hz - 1.4G Hz
- PMU Power Management: Implemented by an MCU, support software power-off, and RTC alarm power-on functions
- System Memory: 2GB 32bit DDR3 RAM
- SD Storage: 1 x microSD Card Socket
- Ethernet: Gbit Ethernet(RTL8211E)
- WiFi: 802.11b/g/n
- Bluetooth: 4.0 dual mode
- Antenna: Porcelain Antenna IPX Interface
- eMMC: 16GB
- Video Input: DVP Camera/MIPI-CSI (two camera interfaces)
- Video Output: HDMI Type-A / LVDS / Parallel RGB-LCD / MIPI-DSI (four video output interfaces)
- Audio: 3.5 mm audio jack / via HDMI
- Microphone: onboard Microphone
- USB Host: 4 x USB 2.0 Host, three type A ports and one 2.54 mm pitch pin-headers
- MicroUSB: 1 x MicroUSB 2.0 Client, Type A
- LCD Interface: 0.5mm pitch 45 pin FPC seat, full color RGB 8-8-8
- HDMI: 1.4A Type A, 1080P
- DVP Camera: 0.5mm pitch 24 pin FPC seat
- GPIO: 2.54 mm pitch 30 pin-header

- I2S/USB: 2.54 mm pitch 14 pin-header
- Serial Debug Port: 2.54mm pitch 4-pin-header
- User Key: power, Reset, boot selection
- LED: 1 x power LED and 1 x system LED
- Other Resources: CPU's internal TMU
- RTC Battery: RTC Battery Seat
- Heat Sink: 1 x Heat Sink with mounting holes
- Power: DC 5V/3A
- PCB: Six Layer
- Dimension: 100 mm x 64 mm
- Working Temperature: -40℃ to 80℃
- OS/Software: uboot, Android and Debian

3 Diagram, Layout and Dimension

3.1 Layout

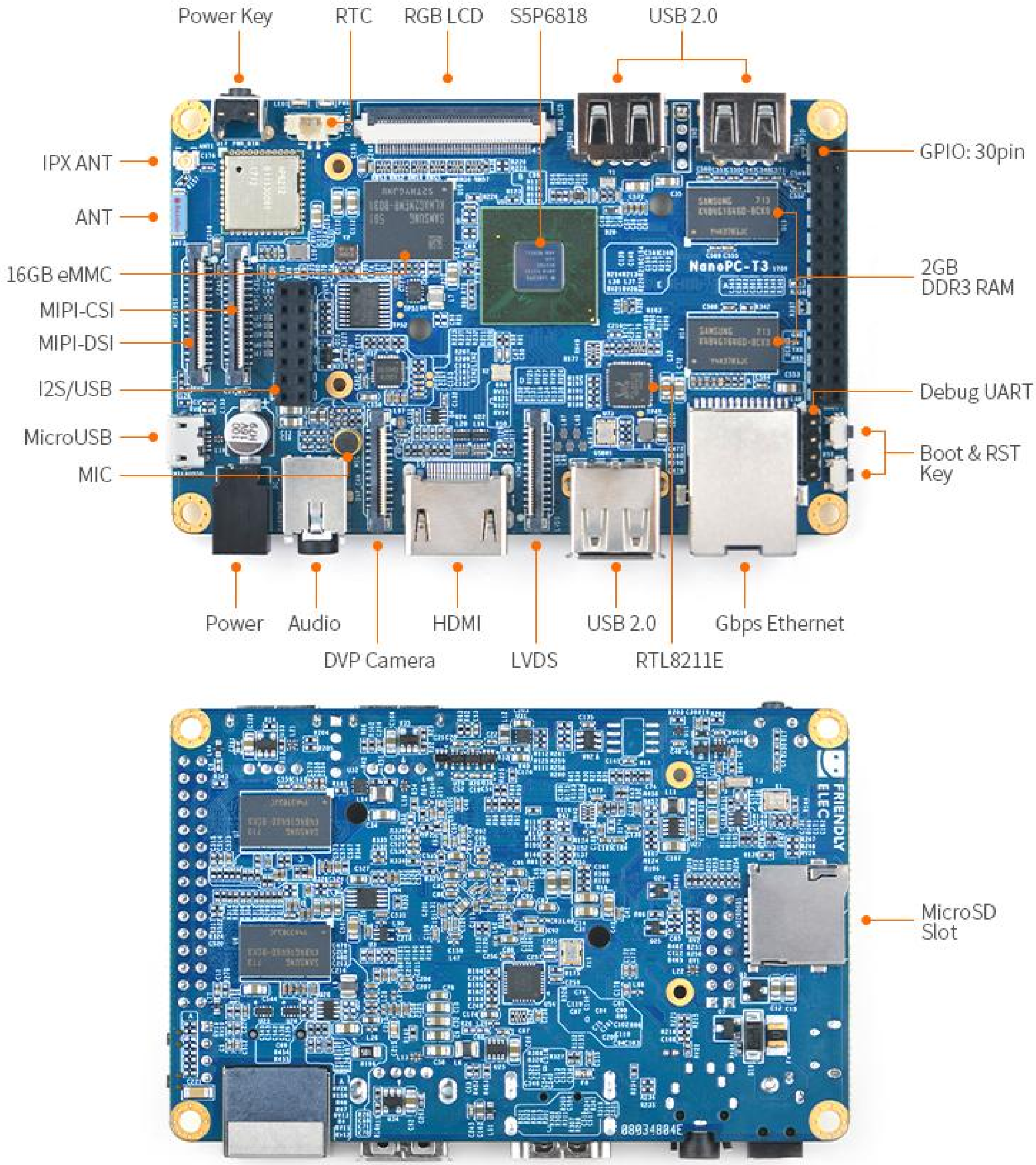
- **30Pin GPIO Pin Spec**

Pin#	Name	Pin#	Name
1	SYS_3.3V	2	DGND
3	UART2_TX/GPIOD20	4	UART2_RX/GPIOD16
5	I2C0_SCL	6	I2C0_SDA
7	SPI0_MOSI/GPIOC31	8	SPI0_MISO/GPIOD0
9	SPI0_CLK/GPIOC29	10	SPI0_CS/GPIOC30
11	UART3_TX/GPIOD21	12	UART3_RX/GPIOD17
13	UART4_TX/GPIOB29	14	UART4_RX/GPIOB28
15	UART5_TX/GPIOB31	16	UART5_RX/GPIOB30
17	GPIOC4	18	GPIOC7
19	GPIOC8	20	GPIOC24
21	GPIOC28	22	GPIOB26
23	GPIOD1/PWM0	24	GPIOD8/PPM
25	GPIOC13/PWM1	26	AliveGPIO3
27	GPIOC14/PWM2	28	AliveGPIO5
29	VDD_5V_OUT	30	DGND

- **14Pin I2S/USB Pin Spec**

Pin#	Name	Pin#	Name
1	VDD_5V	2	VDD_5V
3	USB_DM2	4	LED1
5	USB_DP2	6	I2S_SDIN1
7	DGND	8	I2S_SDOUT1
9	PWRKEY	10	I2S_MCLK1
11	NRESETIN	12	I2S_BCLK1
13	DGND	14	I2S_LRCK1

- **DVP Camera Interface Pin Spec**



NanoPC-T3 Plus Layout

Pin#	Name
1, 2	SYS_3.3V
7,9,13,15,24	DGND
3	I2C0_SCL
4	I2C0_SDA
5	GPIOB14
6	GPIOB16
8	GPIOC13/PWM1
10	NC
11	VSYNC
12	HREF
14	PCLK
16-23	Data bit7-0

- **LVDS**

Pin#	Name
1	VDD_5V_OUT
2	VDD_5V_OUT
3	VDD_5V_OUT
4	LVDS_Y0M
5	LVDS_Y0P
6	DGND
7	LVDS_Y1M
8	LVDS_Y1P
9	DGND
10	LVDS_Y2M
11	LVDS_Y2P
12	DGND
13	LVDS_CLKM
14	LVDS_CLKP
15	DGND
16	LVDS_Y3M
17	LVDS_Y3P
18	DGND
19	GPIOC15
20	DGND
21	I2C2_SCL
22	I2C2_SDA
23	GPIOC16
24	DGND

- **Debug Port（UART0）**

Pin#	Name
1	DGND
2	VDD_5V
3	UART_TXD0
4	UART_RXD0

- **RGB LCD Interface Pin Spec**

Pin#	Name	Description
1, 2	VDD_5V_OUT	5V Output, it can be used to power LCD modules
11，20，29，37，38，39，40, 45	DGND	Ground
3-10	Blue LSB to MSB	RGB blue
12-19	Green LSB to MSB	RGB green
21-28	Red LSB to MSB	RGB red
30	GPIOB25	available for users
31	GPIOC15	occupied by FriendlyARM one wire technology to recognize LCD models and control backlight and implement resistive touch, not applicable for users
32	XnRSTOUT Form CPU	low when system is reset
33	VDEN	signal the external LCD that data is valid on the data bus
34	VSYNC	vertical synchronization
35	HSYNC	horizontal synchronization
36	LCDCLK	LCD clock, Pixel frequency
41	I2C2_SCL	I2C2 clock signal, for capacitive touch data transmission
42	I2C2_SDA	I2C2 data signal, for capacitive touch data transmission
43	GPIOC16	interrupt pin for capacitive touch, used with I2C2
44	NC	Not connected

- **MIPI-DSI Interface Pin Spec**

Pin#	Name
1, 2, 3	VDD_5V_OUT
4	DGND
5	I2C2_SDA
6	I2C2_SCL
7	DGND
8	GPIOC16
9	DGND
10	GPIOC1
11	DGND
12	NC
13	nRESETOUT
14, 15	DGND
16	MIPIDSI_DN3
17	MIPIDSI_DP3
18	DGND
19	MIPIDSI_DN2
20	MIPIDSI_DP2
21	DGND
22	MIPIDSI_DN1
23	MIPIDSI_DP1
24	DGND
25	MIPIDSI_DN0
26	MIPIDSI_DP0
27	DGND
28	MIPIDSI_DNCLK
29	MIPIDSI_DPCLK
30	DGND

- **MIPI-CSI Interface Pin Spec**

Pin#	Name
1, 2	SYS_3.3V
3	DGND
4	I2C0_SDA
5	I2C0_SCL
6	DGND
7	SPI2_MOSI/GPIOC12
8	SPI2_MISO/GPIOC11
9	SPI2_CS/GPIOC10
10	SPI2_CLK/GPIOC9
11	DGND
12	GPIOB23
13	GPIOC2
14, 15	DGND
16	MIPICSI_DN3
17	MIPICSI_DP3
18	DGND
19	MIPICSI_DN2
20	MIPICSI_DP2
21	DGND
22	MIPICSI_DN1
23	MIPICSI_DP1
24	DGND
25	MIPICSI_DN0
26	MIPICSI_DP0
27	DGND
28	MIPICSI_DNCLK
29	MIPICSI_DPCLK
30	DGND

- **RTC**

  3.35uA@3V

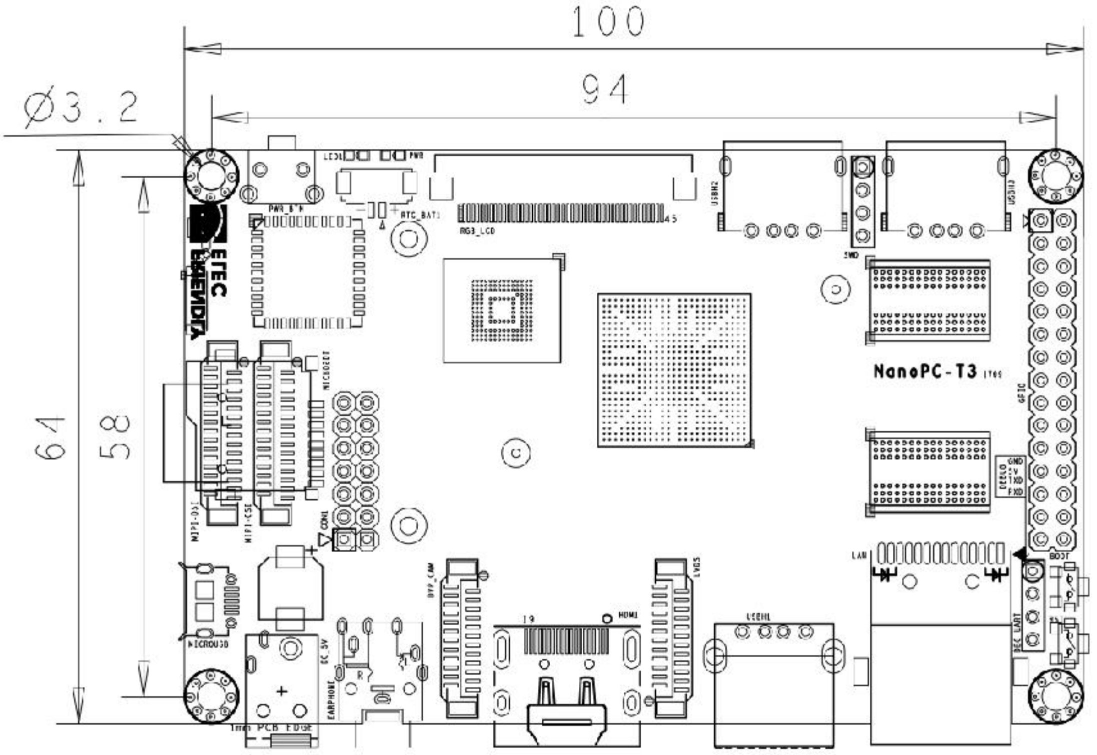
- **USB 2.0 Host**

  with 1A over current protection

  Notes

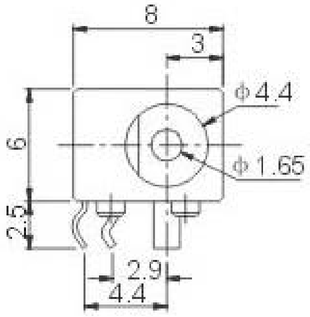
  1. SYS_3.3V: 3.3V power output
  2. VDD_5V/VDD_5V_OUT: 5V power output
  3. For more details refer to the document: NanoPC-T3 Plus Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/d/d0/NanoPC-T3_Plus_1709-Schematic.pdf)

3.2 Board Dimension



For more details refer to the document: NanoPC-T3 Plus Drawing(dxf).zip (http://wiki.friendlyelec.com/wiki/images/9/9e/NanoPC-T3_Plus_1709_Drawing%28dxf%29.zip)

- **Power Jack**
  - DC 4.7~5.6V IN, 4.0*1.7mm Power Jack



4 Get Started

4.1 Essentials You Need

Before starting to use your NanoPC-T3-Plus get the following items ready

- NanoPC-T3-Plus
- SD Card: Class 10 or Above, minimum 8GB SDHC
- A DC 5V/2A power is a must
- HDMI monitor or LCD
- USB keyboard, mouse and possible a USB hub(or a TTL to serial board)
- A host computer running Ubuntu 18.04 64 bit system

4.2 Boot from SD Card

Get the following files from here download link (http://download.friendlyelec.com/NanoPC-T3Plus):

- Get a 8G SDHC card and backup its data if necessary.

Image Files	
s5p6818-sd-friendlycore-xenial-4.4-armhf-YYYYMMDD.img.zip	FriendlyCore(32bit) with Qt 5.10.0 (base on Ubuntu core) image file
s5p6818-sd-friendlycore-xenial-4.4-arm64-YYYYMMDD.img.zip	FriendlyCore(64bit) with Qt 5.10.0 (base on Ubuntu core) image file
s5p6818-sd-lubuntu-desktop-xenial-4.4-armhf-YYYYMMDD.img.zip	LUbuntu Desktop image file with X Window
s5p6818-sd-friendlywrt-4.4-YYYYMMDD.img.zip	FriendlyWrt (http://wiki.friendlyelec.com/wiki/index.php/How_to_Build_FriendlyWrt) image file (base on OpenWrt)
s5p6818-sd-android7-YYYYMMDD.img.zip	Android7 image file
s5p6818-sd-android-lollipop-YYYYMMDD.img.zip	Android5.1 image file
s5p6818-eflasher-lubuntu-desktop-xenial-4.4-armhf-YYYYMMDD.img.zip	SD card image, which is used to install a lubuntu desktop to eMMC
s5p6818-eflasher-friendlywrt-4.4-YYYYMMDD.img.zip	SD card image, which is used to install a FriendlyWrt (http://wiki.friendlyelec.com/wiki/index.php/How_to_Build_FriendlyWrt) to eMMC
s5p6818-eflasher-android7-YYYYMMDD.img.zip	SD card image, which is used to install a android7 to eMMC
s5p6818-eflasher-android-lollipop-YYYYMMDD.img.zip	SD card image, which is used to install an Android to eMMC
s5p6818-eflasher-friendlycore-xenial-4.4-arm64-YYYYMMDD.img.zip	SD card image, which is used to install a FriendlyCore-arm64 to eMMC
s5p6818-eflasher-friendlycore-xenial-4.4-armhf-YYYYMMDD.img.zip	SD card image, which is used to install a FriendlyCore-armhf to eMMC
Flash Utility:	
win32diskimager.rar	Windows utility. Under Linux users can use "dd"

- Uncompress these files. Insert an SD card(at least 4G) into a Windows PC and run the win32diskimager utility 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.
- Insert this card into your board's boot slot, press and hold the boot key (only applies to a board with onboard eMMC) and power on (with a 5V/2A power source). If the PWR LED is on and LED1 is blinking this indicates your board has successfully booted.

4.3 Flash image to eMMC with eflasher

- Download eflasher image file

An image file's name is as : s5p6818-eflasher-OSNAME-YYYYMMDD.img.zip
The "OSNAME" is the name of an OS e.g. android, friendlycore and etc;
This image file is used for making an installation SD card and it contains a Ubuntu core system and a utility

EFlasher;
Download s5p6818-eflasher-OSNAME-YYYYMMDD.img.zip to a host PC and get a windows utility win32diskimager.rar as well;

- Make Installation SD Card with eflasher

Extract the package with a 7z utility and you will get a file with an extension ".img". Insert an SDHC card(minimum 8G or above) to a PC running Windows, run the Win32DiskImager utility as administrator, click on "Image File" to select your wanted file, select your SD card and click on "Write" to start flashing the Image to your SD card;
If your PC runs Linux you can command "dd" to extract the package and get an ".img" file and write it to your SD card;

- Operate in GUI Window: Flash OS to eMMC

Insert your SD card to NanoPC-T3-Plus, connect an HDMI monitor or LCD to your board, press and hold the "boot" key beside the Ethernet port, power on the board you will see a pop-up window asking you to select an OS for installation. Select your wanted OS and start installation.

- Operate in Commandline Utility: Flash OS to eMMC

Insert an installation SD card to NanoPC-T3-Plus, log into or SSH to your board and run the following command to start EFlasher:

```
sudo eflasher
```

4.3.1 Make Installation Card under Linux Desktop

- 1) Insert your SD card into a host computer running Ubuntu and check your SD card's device name

```
dmesg | tail
```

Search the messages output by "dmesg" for similar words like "sdc: sdc1 sdc2". If you can find them it means your SD card has been recognized as "/dev/sdc". Or you can check that by commanding "cat /proc/partitions"

- 2) Downlaod Linux script

```
git clone https://github.com/friendlyarm/sd-fuse_s5p6818.git
cd sd-fuse_s5p6818
```

- 3) Here is how to make a Lubuntu desktop SD card

```
sudo ./fusing.sh /dev/sdx lubuntu
```

(Note: you need to replace "/dev/sdx" with the device name in your system)
When you run the script for the first time it will prompt you to download an image you have to hit “Y” within 10 seconds otherwise you will miss the download

- 4) Run this command to make a complete image file:

```
sudo ./mkimage.sh lubuntu
```

More content please refre: Assembling the SD card image yourself

4.4 Extend SD Card Section

- When Debian/Ubuntu is loaded the SD card's section will be automatically extended.
- When Android is loaded you need to run the following commands on your host PC to extend your SD card's section:

```
sudo umount /dev/sdx?
sudo parted /dev/sdx unit % resizepart 4 100 resizepart 7 100 unit MB print
sudo resize2fs -f /dev/sdx7
```

(Note: you need to replace "/dev/sdx" with the device name in your system)

4.5 LCD/HDMI Resolution

When the system boots our uboot will check whether it is connected to an LCD or to an HDMI monitor. If it recognizes an LCD it will configure its resolution. Our uboot defaults to the HDMI 720P configuration.
If you want to modify the LCD resolution you can modify file "arch/arm/plat-s5p6818/nanopi3/lcds.c" in the kernel and recompile it.
If your NanoPC-T3-Plus is connected to an HDMI monitor and it runs Android it will automatically set the resolution to an appropriate HDMI mode by checking the "EDID". If your NanoPC-T3-Plus is connected to an HDMI monitor and it runs Debian by default it will set the resolution to the HDMI 720P configuration. If you want to modify the HDMI resolution to 1080P modify your kernel's configuration as explained above.

4.6 Update SD Card's boot parameters From PC Host

Insert your SD card into a host PC running Linux, if you want to change your kernel command line parameters you can do it via the fw_setevn utility.
Check the current Command Line:

```
git clone https://github.com/friendlyarm/sd-fuse_s5p6818.git
cd sd-fuse_s5p6818/tools
./fw_printenv /dev/sdx | grep bootargs
```

For example, to disable android SELinux, You can change it this way:

```
./fw_setenv /dev/sdc bootargs XXX androidboot.selinux=permissive
```

The "XXX" stands for the original bootargs' value.

5 Work with FriendlyCore

5.1 Introduction

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

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

- it supports our LCDs with both capacitive touch and resistive touch(S700, X710, HD702, S430, HD101 and S70)
- it supports WiFi
- it supports Ethernet
- it supports Bluetooth and has been installed with bluez utilities
- it supports audio playing

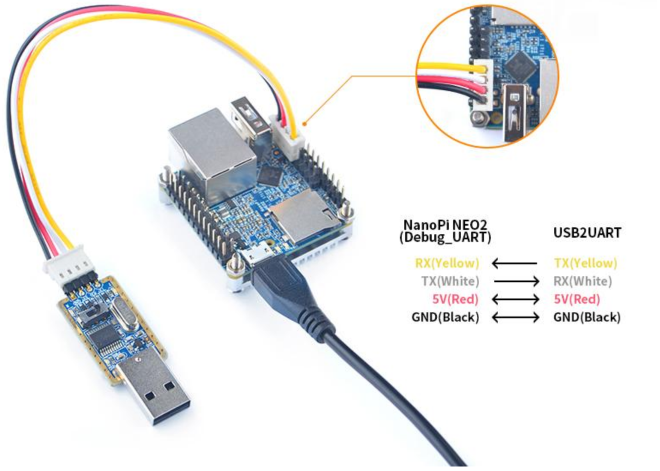
- it supports Qt 5.10.0 EGLES and OpenGL ES1.1/2.0 (Only for S5P4418/S5P6818)

5.2 System Login

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

For example, NanoPi-M1：

You can use a USB to Serial conversion board too.
Make sure you use a 5V/2A power to power your board from its MicroUSB port:
For example, NanoPi-NEO2：



- FriendlyCore User Accounts:

Non-root User:

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

Root:

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

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

- Update packages

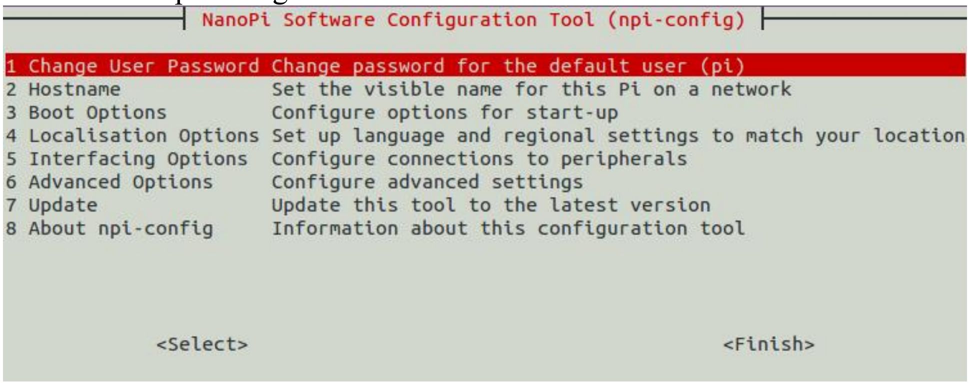
```
$ sudo apt-get update
```

5.3 Configure System with npi-config

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

```
$ sudo npi-config
```

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



```
NanoPi Software Configuration Tool (npi-config)

1 Change User Password Change password for the default user (pi)
2 Hostname             Set the visible name for this Pi on a network
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Advanced Options     Configure advanced settings
7 Update               Update this tool to the latest version
8 About npi-config     Information about this configuration tool

             <Select>                                       <Finish>
```

5.4 Develop Qt Application

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

5.5 Setup Program to AutoRun

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

```
sudo npi-config
```

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

5.6 Extend TF Card's Section

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

```
$ df -h
```

5.7 Transfer files using Bluetooth

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

```
hcitool scan
```

Search results look like：

```
Scanning ...
```

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

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

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

Note: Please replace the MAC address in the above command with the actual Bluetooth MAC address of the mobile phone.
This command will detail the protocols supported by Bluetooth for mobile phones. What we need to care about is a file transfer service called OBEX Object Push. Take the HTC6525LVW mobile phone as an example. The results are as follows：

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

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

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

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

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

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

```
rfkill unblock 0
```

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

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

5.8 WiFi

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

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

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

- Change to root

```
$ su root
```

- Check device list

```
$ nmcli dev
```

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

- Start WiFi

```
$ nmcli r wifi on
```

- Scan Surrounding WiFi Sources

```
$ nmcli dev wifi
```

- Connect to a WiFi Source

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

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

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

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

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

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

5.9 Setup Wi-Fi AP

Follow the steps below. Since our OS image by default already has the NetworkManager utility you will be prompted to uninstall it first:

```
sudo turn-wifi-into-apmode yes
```

After you uninstall the NetworkManager reboot your board.
After your board is rebooted run the above commands again and you will be prompted to type in a WIFI's name and password. Type in your wanted name and password

If this is successful you will be able to find and connect your board to a WIFI. Login to your board at 192.168.8.1:

```
ssh root@192.168.8.1
```

Type in a password. In our system the password is "fa".

To login smoothly via SSH we recommend you turning off WIFI's power save mode by running the following commands:

```
sudo iwconfig wlan0 power off
```

You can check your WiFi's mode by running the following command:

```
sudo cat /sys/module/bcmdhd/parameters/op_mode
```

Number 2 means your WiFi is in AP mode. You can switch to the Station mode by running the following command:

```
sudo turn-wifi-into-apmode no
```

5.10 Bluetooth

Search for surrounding bluetooth devices by running the following command:

```
$ su root
$ hciconfig hci0 up
$ hcitool scan
```

You can run "hciconfig" to check bluetooth's status.

5.11 Ethernet Connection

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

5.12 Custom welcome message

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

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

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

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

To:

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

5.13 Modify timezone

For exampe, change to Shanghai timezone:

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

5.14 Select the system default audio device

You can set the system default audio device by following the steps below.
Use the following command to view all the sound card devices in the system (Note: different development boards will have different results):

```
pi@NanoPi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: nanopi2audio [nanopi2-audio], device 0: c0055000.i2s-ES8316 HiFi ES8316 HiFi-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: nanopi2audio [nanopi2-audio], device 1: c0059000.spdiftx-dit-hifi dit-hifi-1 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

As you can see, the following sound card devices are available on the hardware:

Sound card device	Sound card number	Description
nanopi2audio	device 0	3.5mm jack interface
nanopi2audio	device 1	HDMI

To configure the audio output to the 3.5mm jack, create or modify the configuration file /etc/asound.conf and modify it to the following:

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

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

To configure to output audio to HDMI, change the device 0 above to device 1.

5.15 Run the X11 application

FriendlyCore system built-in lightweight Xorg， although there is no window manager, you can still run a single X-Windows application， For example, the program to run is ~/YourX11App， use the following command：

```
. /usr/bin/setqt5env-xcb
startx ~/YourX11App -geometry 1280x800
```

Note that there is a space between "." and /usr/bin/setqt5env-xcb. In addition, the resolution after -geometry should be changed to the actual resolution of your screen.

5.16 Run Qt 5.10.0 Demo with GPU acceleration

Run the following command

```
$ sudo qt5demo
```

Cinematic Experience

Welcome to 'Cinematic Experience' demo. This application demonstrates the power of Qt5 and few of the new additions available in QtQuick 2.0. Below is a short summary of those new features which have been used in this demo application.

Rendering

Qt5 has brand new rendering backend 'QML SceneGraph' which is optimized for hardware accelerated rendering. This allows to take fu gains out of OpenGL powered GPUs on desktop and embedded devi Not just performance, new Qt5 rendering backend also allows featu which have not been possible earlier.

Particles

Qt5 comes with a fresh particles plugin 'QtQuick.Particles 2.0' which is superior compared to Qt4 particles. In this demo application, twinkling stars, shooting star and fog/smoke have been implemented using this

5.17 Run Qt 5.10.0 Demo with OpenGL

Run the following command

```
. setqt5env
cd $QTDIR
cd /examples/opengl/qopenglwidget
./qopenglwidget
```

For more Qt 5.10.0 examples, please go to:
cd $QTDIR/examples/

5.18 Play HD Video with Hardware-decoding

gst-player is console player, it base on GStreamer, support VPU with Hardware-decoding:

```
sudo gst-player /home/pi/demo.mp4
```

The equivalent gsteamer command is as follows：

```
sudo gst-launch-1.0 filesrc location=/home/pi/demo.mp4 ! qtdemux name=demux demux. ! queue ! faad ! audioconvert ! audioresampl
```

5.19 Connect to DVP Camera CAM500B

The CAM500B camera module is a 5M-pixel camera with DVP interface. For more tech details about it you can refer to Matrix - CAM500B.
Enter the following command to preview the video：

```
gst-launch-1.0 -e v4l2src device=/dev/video6 ! video/x-raw,format=I420,framerate=30/1,width=1280,height=720 ! nxvideosink
```

Enter the following command to start recording (VPU hardware encoding)：

```
gst-launch-1.0 -e v4l2src device=/dev/video6 ! video/x-raw,format=I420,framerate=30/1,width=1280,height=720 ! tee name=t t. \
 ! queue ! nxvideosink t. ! queue ! nxvideoenc bitrate=12000000 ! mp4mux ! \
 filesink location=result_720.mp4
```

5.20 Power Off and Schedule Power On

“PMU Power Management” feature helps us to auto power on the board at a specific time, it is implemented by an MCU, support software power-off, and RTC alarm power-up functions.

Here’s a simple guide:
Turn on automatically after 100 seconds. (Time must be greater than 60 seconds.):

```
$ sudo echo 100 > /sys/class/i2c-dev/i2c-3/device/3-002d/wakealarm
```

After setting up the automatic boot, turn off board with the 'poweroff’ command:

```
$ sudo poweroff
```

Cancel automatic boot:

```
$ sudo echo 0 > /sys/class/i2c-dev/i2c-3/device/3-002d/wakealarm
```

Query the current settings, in the front is current time, followed by the time of automatic booting: If no automatic boot is set, it will display "disabled”.

```
$ sudo cat /sys/class/i2c-dev/i2c-3/device/3-002d/wakealarm
```

Note that some older versions of hardware may not support this feature, if you don't see this file node in your system:
/sys/class/i2c-dev/i2c-3/device/3-002d/wakealarm
your board may be it does not support this feature.

5.21 Installing and Using OpenCV 4.1.2

OpenCV has been pre-installed in FriendlyCore (Version after 20191126) and does not require manual installation.
Please refre this link: https://github.com/friendlyarm/install-opencv-on-friendlycore/blob/s5pxx18/README.md
Quick test：

```
. /usr/bin/cv-env.sh
. /usr/bin/setqt5env-eglfs
cd /usr/local/share/opencv4/samples/python
python3 turing.py
```

5.22 Installing and Using Caffe

```
git clone https://github.com/friendlyarm/install-caffe-on-friendlycore
cd install-caffe-on-friendlycore
sudo ./install-caffe.sh
```

5.23 How to install and use docker (for aarch64 system)

5.23.1 How to Install Docker

Run the following commands：

```
wget https://download.docker.com/linux/ubuntu/dists/xenial/pool/stable/arm64/containerd.io_1.2.6-3_arm64.deb
wget https://download.docker.com/linux/ubuntu/dists/xenial/pool/stable/arm64/docker-ce-cli_19.03.2~3-0~ubuntu-xenial_arm64.deb
wget https://download.docker.com/linux/ubuntu/dists/xenial/pool/stable/arm64/docker-ce_19.03.2~3-0~ubuntu-xenial_arm64.deb
sudo dpkg -i containerd.io_1.2.6-3_arm64.deb
sudo dpkg -i docker-ce-cli_19.03.2~3-0~ubuntu-xenial_arm64.deb
sudo dpkg -i docker-ce_19.03.2~3-0~ubuntu-xenial_arm64.deb
```

5.23.2 Test Docker installation

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

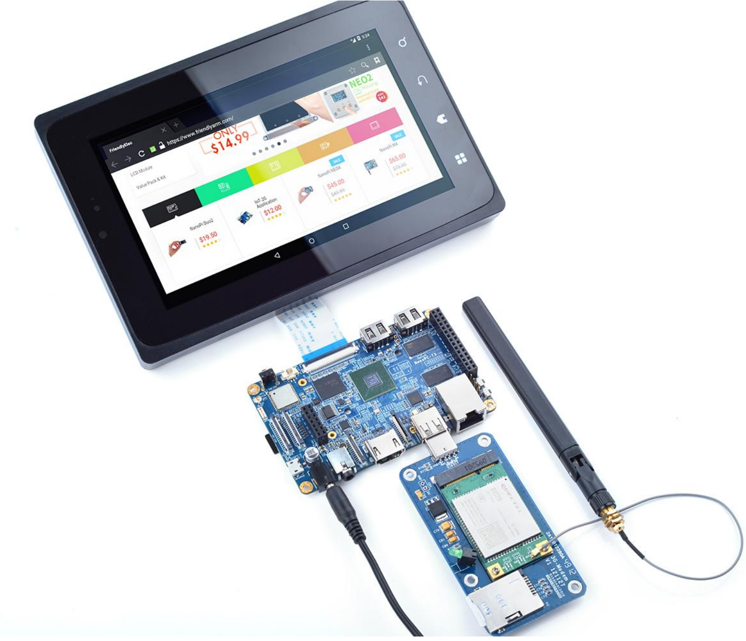
```
git clone https://github.com/friendlyarm/debian-jessie-arm-docker
cd debian-jessie-arm-docker
./rebuild-image.sh
./run.sh
```

6 Work with Android

6.1 Work with 4G Module EC20 under Android5

6.1.1 Hardware Setup

Connect an EC20 module to a USB to miniPCIe board and connect the board to an ARM board's USB Host. Here is a hardware setup:



Power on the board and you will be able to surf the internet with the 4G module like using an Android phone.

6.2 Modify the Android boot Logo

Replace the logo.bmp:

```
/opt/FriendlyARM/smart4418/android/device/friendly-arm/nanopi3/boot/logo.bmp
/opt/FriendlyARM/smart4418/android/device/friendly-arm/nanopi2/boot/logo.bmp
```

Replace the bootanimation.zip:

```
/opt/FriendlyARM/smart4418/android/device/friendly-arm/nanopi3/bootanimation.zip
/opt/FriendlyARM/smart4418/android/device/friendly-arm/nanopi2/bootanimation.zip
```

Re-compile android.

6.3 Use fastboot command to flash android firmware

Enter the uboot command line mode on the serial terminal when powering on, and then enter the following command:

```
fastboot 0
```

For S5P4418:

```
fastboot flash partmap partmap.txt
fastboot flash 2ndboot bl1-mmcboot.bin
fastboot flash fip-loader loader-mmc.img
fastboot flash fip-secure bl_mon.img
fastboot flash fip-nonsecure bootloader.img
fastboot flash boot boot.img
fastboot flash system system.img
fastboot flash cache cache.img
fastboot flash userdata userdata.img
```

For S5P6818:

```
fastboot flash partmap partmap.txt
fastboot flash 2ndboot bl1-mmcboot.bin
fastboot flash fip-loader fip-loader.img
fastboot flash fip-secure fip-secure.img
fastboot flash fip-nonsecure fip-nonsecure.img
fastboot flash boot boot.img
fastboot flash system system.img
fastboot flash cache cache.img
fastboot flash userdata userdata.img
```

6.4 Android Keys

Android 5:

```
  vendor/friendly-arm/nanopi3/security/
```

Android 7:

```
  build/target/product/security/
```

6.5 Optimizing HDMI Performance on Android 7

6.5.1 Note

By default, the driver initializes two framebuffers, one for the primary LCD display and the other for HDMI. If your project specifically requires the use of HDMI and not the LCD, you can follow the steps outlined in this chapter to make modifications. After making these changes, HDMI will be configured as the primary display, resulting in the initialization of only one framebuffer. This optimization conserves resources and leads to corresponding improvements in UI performance and boot speed.
The content of this chapter is applicable exclusively to S5P6818 running Android 7. For S5P4418, the modification process is similar, with adjustments needed in the corresponding files.

6.5.2 Modify the kernel

You need to modify the kernel Device Tree Source (DTS) to disable the dp_drm_lvds node, as shown below:

```
--- a/arch/arm64/boot/dts/nexell/s5p6818-nanopi3-common.dtsi
+++ b/arch/arm64/boot/dts/nexell/s5p6818-nanopi3-common.dtsi
@@ -810,6 +810,7 @@
                        plane-names = "video", "rgb", "primary";
                };
                port@1 {
+                       status = "disabled";
                        reg = <1>;
                        back_color = < 0x0 >;
                        color_key = < 0x0 >;
@@ -820,7 +821,7 @@

 &dp_drm_lvds {
        remote-endpoint = <&lcd_panel>;
-       status = "ok";
+       status = "disabled";

        display-timing {
                clock-frequency = <50000000>;
```

After compilation, you will obtain a new arch/arm64/boot/dts/nexell/s5p6818-nanopi3-rev*.dtb file.
During the testing phase, you can directly update it to the board using adb with the following command:

```
adb root; adb wait-for-device; adb shell mkdir /storage/sdcard1/; adb
shell mount -t ext4 /dev/block/mmcblk0p1 /storage/sdcard1/
adb push arch/arm64/boot/dts/nexell/s5p6818-nanopi3-rev*.dtb /storage/sdcard1/
```

For a complete firmware update, you will need to replace the files in the device/friendlyelec/nanopi3/boot directory of the Android 7 source code.

6.5.3 Modify env.conf

To modify the device/friendlyelec/nanopi3/boot/env.conf file and add a line

```
lcdtype     HDMI1080P60
```

This mode needs to match the mode detected by Android 7 after startup; otherwise, it may result in a prolonged black screen state or even no display output. In such cases, you may need to manually set it in the U-Boot command-line environment:

```
setenv lcdtype HDMI1080P60; saveenv; reset
```

6.5.4 Modify system.prop

To modify the device/friendlyelec/nanopi3/system.prop file in Android 7

```
ro.sf.lcd_density=240
```

Alternatively, you can adjust the system property or experiment with different values that you deem more appropriate. You can also use the following command to change the display density under the serial or adb environment and observe if the effect is suitable:

```
adb shell wm density 240
```

6.5.5 Compiling Android

Follow the instructions in the wiki to compile Android 7 and conduct testing. If you encounter any exceptions, please carefully review the preceding steps.

7 More OS Support

7.1 DietPi

DietPi is a highly optimised & minimal Debian-based Linux distribution. DietPi is extremely lightweight at its core, and also extremely easy to install and use.
Setting up a single board computer (SBC) or even a computer, for both regular or server use, takes time and skill. DietPi provides an easy way to install and run favourite software you choose.
For more information, please visit this link https://dietpi.com/docs/.

DietPi supports many of the NanoPi board series, you may download the image file from here:

- https://dietpi.com/docs/hardware/#nanopi-series-friendlyarm

8 Make Your Own OS Image

8.1 Install Cross Compiler

8.1.1 Install aarch64-linux-gcc 6.4

Download the compiler package:

```
git clone https://github.com/friendlyarm/prebuilts.git -b master --depth 1
cd prebuilts/gcc-x64
cat toolchain-4.9.3-armhf.tar.gz* | sudo tar xz -C /
```

Then add the compiler's directory to "PATH" by appending the following lines in "~/.bashrc":

```
export PATH=/opt/FriendlyARM/toolchain/4.9.3/bin:$PATH
export GCC_COLORS=auto
```

Execute "~/.bashrc" to make the changes take effect. Note that there is a space after the first ".":

```
. ~/.bashrc
```

This compiler is a 64-bit one therefore it cannot be run on a 32-bit Linux machine. After the compiler is installed you can verify it by running the following commands:

```
arm-linux-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gcc
COLLECT_LTO_WRAPPER=/opt/FriendlyARM/toolchain/4.9.3/libexec/gcc/arm-cortexa9-linux-gnueabihf/4.9.3/lto-wrapper
Target: arm-cortexa9-linux-gnueabihf
Configured with: /work/toolchain/build/src/gcc-4.9.3/configure --build=x86_64-build_pc-linux-gnu
--host=x86_64-build_pc-linux-gnu --target=arm-cortexa9-linux-gnueabihf --prefix=/opt/FriendlyARM/toolchain/4.9.3
--with-sysroot=/opt/FriendlyARM/toolchain/4.9.3/arm-cortexa9-linux-gnueabihf/sys-root --enable-languages=c,c++
--with-arch=armv7-a --with-tune=cortex-a9 --with-fpu=vfpv3 --with-float=hard
...
Thread model: posix
gcc version 4.9.3 (ctng-1.21.0-229g-FA)
```

8.2 Compile Linux kernel for FriendlyCore/Lubuntu/EFlasher

8.2.1 Compile Kernel

- Download Kernel Source Code

```
git clone https://github.com/friendlyarm/linux.git -b nanopi2-v4.4.y --depth 1
cd linux
```

The kernel source for S5P6818 is in the "nanopi2-v4.4.y" branch. Before you start compiling it you need to switch to this branch.

- Compile Ubuntu Kernel

```
touch .scmversion
make ARCH=arm64 nanopi3_linux_defconfig
make ARCH=arm64
```

After your compilation succeeds an "arch/arm/boot/Image" will be generated and a DTB file(s5p6818-nanopi3-rev*.dtb) will be generated in the "arch/arm/boot/dts/nexell" directory. You can use them to replace the existing Image and DTB files in the boot partition of your bootable SD card.

8.2.2 Use Your Generated Kernel

- Update kernel in SD card

If you use an SD card to boot Ubuntu you can copy your generated Image and DTB files to your SD card's boot partition(e.g. partition 1 /dev/sdX1).

- Update kernel in eMMC

If you boot your board from eMMC you can update your kernel file by following the steps below:
1) Usually after OS is loaded eMMC's boot partition (in our example eMMC's device name was /dev/mmcblk0p1) will be automatically mounted and you can verify that by running "mount"
2) Connect your board to a host PC running Ubuntu and copy the Image and DTB files to eMMC's boot partition
3) Or you can copy your generated kernel file to an external storage card(e.g. an SD card or a USB drive), connect

the storage card to your board the move the file from the card to eMMC's boot partition
4) After update is done type "reboot" to reboot your board. Note: don't just directly disconnect your board from its power source or press the reset button to reboot the board. These actions will damage your kernel file

- Generate Your boot.img

Refer to this repo: https://github.com/friendlyarm/sd-fuse_s5p6818

8.3 Compile Linux kernel for Android7

The Android 7.1.2 source code already contains the pre-compiled kernel. If you need to customize it, you can compile the kernel according to the following guide.

```
git clone https://github.com/friendlyarm/linux.git -b nanopi2-v4.4.y --depth 1
cd linux
touch .scmversion
make ARCH=arm64 nanopi3_nougat_defconfig
make ARCH=arm64
```

The newly generated kernel is arch/arm64/boot/Image， The new DTB file is also included under the directory arch/arm64/boot/dts/nexell/.(s5p6818-nanopi3-rev*.dtb).
If you only want to debug the kernel, you can quickly update it with adb:

```
adb root; adb shell mkdir /storage/sdcard1/; adb shell mount -t ext4 /dev/block/mmcblk0p1 /storage/sdcard1/;
adb push arch/arm64/boot/Image arch/arm64/boot/dts/nexell/s5p6818-nanopi3-rev*.dtb /storage/sdcard1/
```

If you want to generate boot.img for burning, you can copy the kernel Image and DTB files to the Android7 source code directory: device/friendlyelec/nanopi3/boot, then recompile Android7.

8.4 Compile U-Boot for Android7/FriendlyCore/Lubuntu/EFlasher

Download the U-Boot v2016.01 source code and compile it. Note that the github's branch is nanopi2-v2016.01:

```
git clone https://github.com/friendlyarm/u-boot.git
cd u-boot
git checkout nanopi2-v2016.01
make s5p6818_nanopi3_config
make CROSS_COMPILE=aarch64-linux-
```

After your compilation succeeds a fip-nonsecure.img will be generated. If you want to test it flash it to your installation SD card to replace an existing U-Boot v2016.01 file via fastboot, sd-fuse_s5p6818 or eflasher ROM.
For Android7: You can copy fip-nonsecure.img to the Android7 source directory device/friendlyelec/nanopi3/boot and recompile Android7.
Note: you cannot use mixed U-Boot files. For example you cannot use fastboot to update an existing U-Boot V2014.07 and you cannot use bootloader.img to replace an existing u-boot.bin.

8.5 Compile Android 7.1.2

8.5.1 Install Cross Compiler

Install 64 bit Ubuntu 16.04 on your host PC.

```
sudo apt-get install bison g++-multilib git gperf libxml2-utils make python-networkx zip
sudo apt-get install flex curl libncurses5-dev libssl-dev zlib1g-dev gawk minicom
sudo apt-get install openjdk-8-jdk
sudo apt-get install exfat-fuse exfat-utils device-tree-compiler liblz4-tool
```

For more details refer to https://source.android.com/source/initializing.html 。

8.5.2 Download Android7 Source Code

There are two ways to download the source code:

- **repo archive file on netdisk**

Netdisk URL: Click here (http://download.friendlyelec.com/NanoPC-T3Plus)
File location on netdisk： sources/s5pxx18-android-7.git-YYYYMMDD.tgz (YYYYMMDD means the date of packaging)
After extracting the repo package from the network disk, you need to execute the sync.sh script, which will pull the latest code from gitlab:

```
tar xvzf /path/to/netdisk/sources/s5pxx18-android-7.git-YYYYMMDD.tgz
cd s5pxx18-android-7
./sync.sh
```

- **git clone from gitlab**

NanoPC-T3-Plus source code is maintained in gitlab, You can download it by running the following command:

```
git clone https://gitlab.com/friendlyelec/s5pxx18-android-7.git -b master
```

8.5.3 Compile Android7

```
cd s5pxx18-android-7
source build/envsetup.sh
lunch aosp_nanopi3-userdebug
make -j8
```

After your compilation succeeds the following files will be generated in the "out/target/product/nanopi3/" directory.

filename	partition	Description
bl1-mmcboot.bin	raw	boot firmware
fip-loader.img	raw	boot firmware
fip-secure.img	raw	boot firmware
fip-nonsecure.img	raw	uboot-v2016.01
env.conf	-	Uboot environment variable containing Android kernel command line parameters
boot.img	boot	kernel Image, DTBs; logo; Android ramdisk
cache.img	cache	-
userdata.img	userdata	-
system.img	system	-
partmap.txt	-	Partition description file

9 Build Kernel Headers Package

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

9.1 Software Version

The OS image file name: s5p6818-sd-friendlycore-xenial-4.4-arm64-YYYYMMDD.img, s5p6818-eflasher-friendlycore-xenial-4.4-arm64-YYYYMMDD.img

```
pi@NanoPC-T3:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.6 LTS
Release:        16.04
Codename:       xenial
pi@NanoPC-T3:~$ cat /proc/version
Linux version 4.4.172-s5p6818 (root@jensen) (gcc version 6.4.0 (ctng-1.23.0-150g-FA) ) #1 SMP PREEMPT Thu Jan 21 18:51:13 CST 2
```

9.2 Install the required packages

```
sudo apt-get update
sudo apt-get install -y dpkg-dev bsdtar
```

9.3 Build Kernel Headers Package

```
git clone https://github.com/friendlyarm/linux -b nanopi2-v4.4.y --depth 1 kernel-s5pxx18-arm64
cd kernel-s5pxx18-arm64
rm -rf .git
make distclean
touch .scmversion
sed -i '/^CONFIG_CROSS_COMPILE/d' ./arch/arm64/configs/nanopi3_linux_defconfig
make CROSS_COMPILE= ARCH=arm64 nanopi3_linux_defconfig
alias tar=bsdtar
make CROSS_COMPILE= ARCH=arm64 bindeb-pkg -j4
```

The following message is displayed to indicate completion:

```
dpkg-deb: building package 'linux-firmware-image-4.4.172-s5p6818' in '../linux-firmware-image-4.4.172-s5p6818_4.4.172-s5p6818-1
dpkg-deb: building package 'linux-headers-4.4.172-s5p6818' in '../linux-headers-4.4.172-s5p6818_4.4.172-s5p6818-1_arm64.deb'.
dpkg-deb: building package 'linux-libc-dev' in '../linux-libc-dev_4.4.172-s5p6818-1_arm64.deb'.
dpkg-deb: building package 'linux-image-4.4.172-s5p6818' in '../linux-image-4.4.172-s5p6818_4.4.172-s5p6818-1_arm64.deb'.
dpkg-genchanges: binary-only upload (no source code included)
```

9.4 Installation

```
sudo rm -f /lib/modules/4.4.172-s5p6818/build
sudo rm -f /lib/modules/4.4.172-s5p6818/source
sudo dpkg -i ../linux-headers-4.4.172-s5p6818_4.4.172-s5p6818-1_arm64.deb
```

9.5 Testing

To compile the pf_ring module as an example, refer to the documentation: https://www.ntop.org/guides/pf_ring/get_started/git_installation.html.

```
git clone https://github.com/ntop/PF_RING.git
cd PF_RING/kernel/
make CROSS_COMPILE=
```

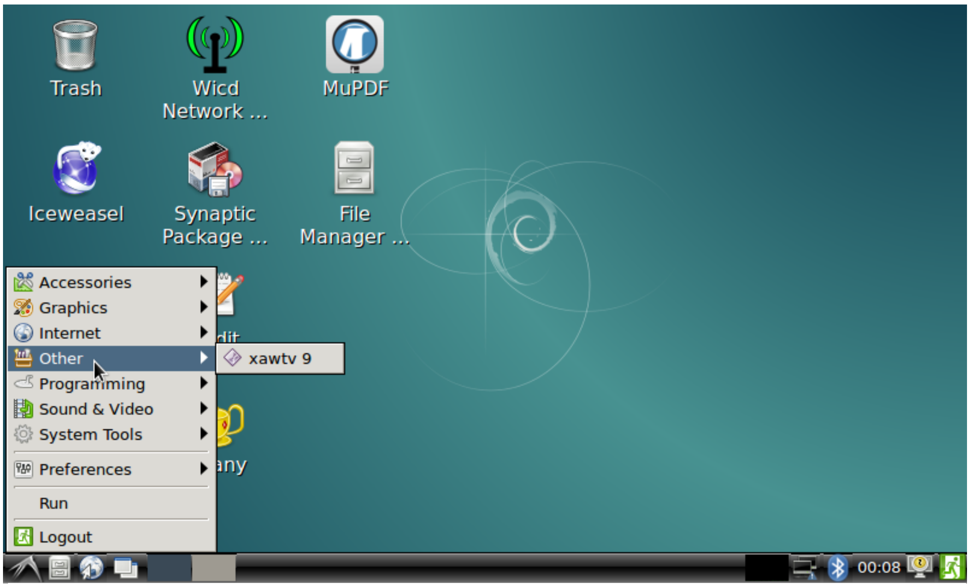
After compiling, use insmod to try to load the module:

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

10 Connect NanoPC-T3-Plus to External Modules

10.1 Connect NanoPC-T3-Plus to USB Camera(FA-CAM202)

- In this use case the NanoPC-T3-Plus runs Debian. If you connect your NanoPC-T3-Plus to our LCD or an HDMI monitor after Debain is fully loaded click on "other"-->"xawtv" on the left bottom of the GUI and the USB Camera application will be started. After enter "welcome to xawtv！ " click on "OK" to start exploring.



10.2 Connect NanoPC-T3-Plus to CMOS 5M-Pixel Camera

For more details about the CAM500A camera refer to [1] (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_CAM500A)

- If your NanoPC-T3-Plus runs Android5.1 and it is connected to our LCD or an HDMI monitor after Android is fully loaded click on the "Camera" icon and the application will be started. You can take pictures or record videos

- Under Debian a camera utility "nanocams" is available for previewing 40 frames and picture taking. You can try it by following the commands below

```
sudo nanocams -p 1 -n 40 -c 4 -o IMG001.jpg
```

For more details about the usage of the nanocams run "nanocams -h". You can get its source code from our git hub:

```
git clone https://github.com/friendlyarm/nexell_linux_platform.git
```

- Under FriendlyCore (kernel 4.4), You can try it by following the commands below:

Enter the following command to preview the video：

```
gst-launch-1.0 -e v4l2src device=/dev/video6 ! video/x-raw,format=I420,framerate=30/1,width=1280,height=720 ! nxvideosink
```

Enter the following command to start recording (VPU hardware encoding)：

```
gst-launch-1.0 -e v4l2src device=/dev/video6 ! video/x-raw,format=I420,framerate=30/1,width=1280,height=720 ! tee name=t t. \
 ! queue ! nxvideosink t. ! queue ! nxvideoenc bitrate=12000000 ! mp4mux ! \
 filesink location=result_720.mp4
```

10.3 Use OpenCV to Access USB Camera

- The full name of "OpenCV" is Open Source Computer Vision Library and it is a cross platform vision library.
- When the NanoPC-T3-Plus runs Debian users can use OpenCV APIs to access a USB Camera device.

1. Here is a guideline on how to use OpenCV with C++ on the NanoPC-T3-Plus:

- Firstly you need to make sure your NanoPC-T3-Plus is connected to the internet.Login to your NanoPC-T3-Plus via a serial terminal or SSH. After login type in your username(root) and password(fa):
- Run the following commands:

```
apt-get update
apt-get install libcv-dev libopencv-dev
```

2. Make sure your USB camera works with the NanoPC-T3-Plus. You can test your camera with NanoPC-T3-Plus's camera utility.

3. Check your camera device:

```
ls /dev/video*
```

- Note:in our test case video0 was the device name.

4. OpenCV's code sample(official code in C++) is under /home/fa/Documents/opencv-demo. Compile the code sample with the following commands:

```
cd /home/fa/Documents/opencv-demo
make
```

After it is compiled successfully a "demo" executable will be generated

5. Connect NanoPC-T3-Plus to USB Keyboard & Run the Following Command:

```
./demo
```

opencv is successfully started

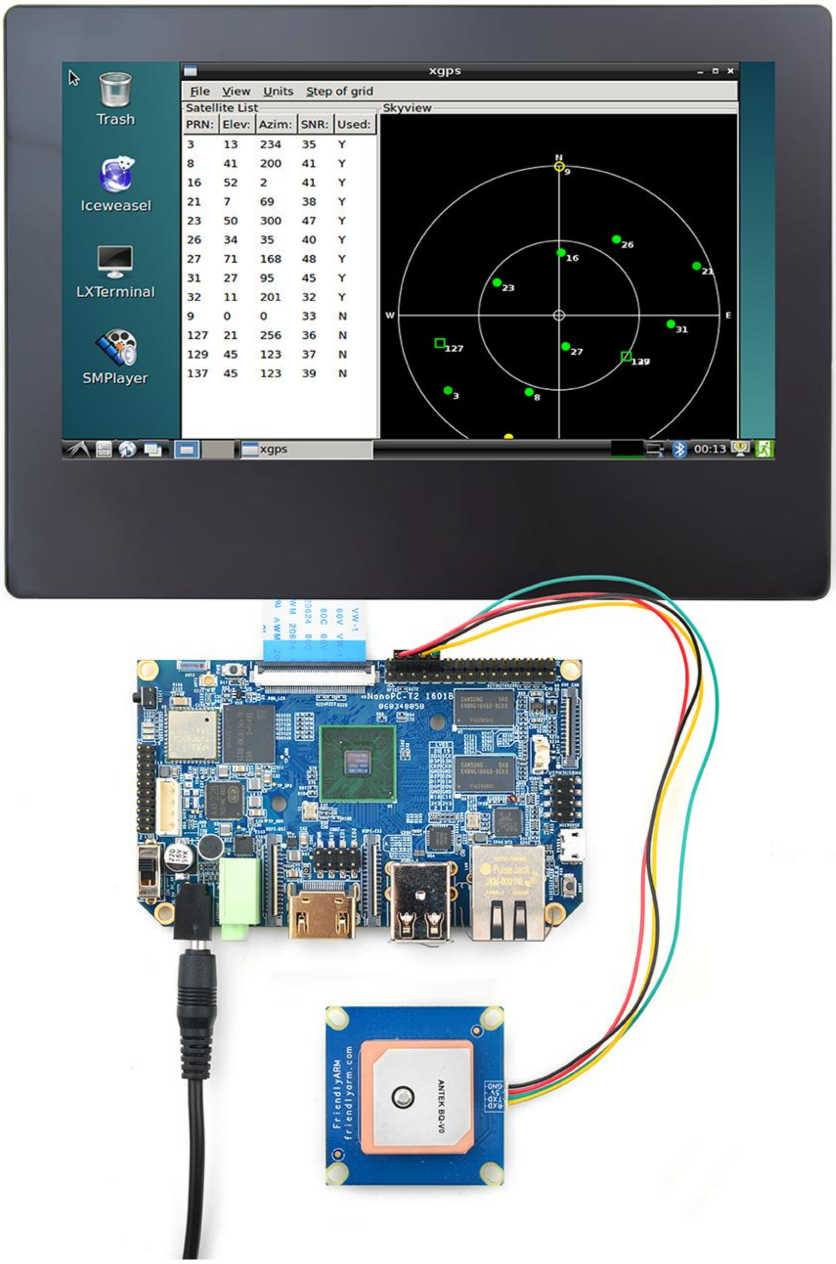
10.4 Connect NanoPC-T3-Plus to Matrix GPS Module

- The Matrix-GPS module is a small GPS module with high performance. It can be used in navigation devices, four-axle drones and etc.
- The Matrix-GPS module uses serial communication. When the NanoPC-T3-Plus is connected to the Matrix GPS module, after the NanoPC-T3-Plus is powered up type in the following command in a terminal or click on the xgps icon it will be started.

```
$su - fa -c "DISPLAY=:0 xgps 127.0.0.1:9999"
```

- Or on the Debian GUI start the LXTerminal, type in "xgps" and enter it will be started too.

For more details about this GPS module refer to Click to check (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_GPS)
Refer to the following diagram to connect the NanoPC-T3-Plus to the Matrix-GPS:



Satellite List

PRN:	Elev:	Azim:	SNR:	Used:
3	13	234	35	Y
8	41	200	41	Y
16	52	2	41	Y
21	7	69	38	Y
23	50	300	47	Y
26	34	35	40	Y
27	71	168	48	Y
31	27	95	45	Y
32	11	201	32	Y
9	0	0	33	N
127	21	256	36	N
129	45	123	37	N
137	45	123	39	N

Connection Details:

Matrix-GPS	NanoPC-T3-Plus
RXD	Pin11
TXD	Pin12
5V	Pin29
GND	Pin30

11 Access Hardware under Android

FriendlyElec developed a library called “libfriendlyarm-things.so”, for android developer to access the hardware resources on the development board in their android apps, the library is based on Android NDK.
Accessible Modules:

- Serial Port
- PWM

- EEPROM
- ADC
- LED
- LCD 1602 (I2C)
- OLED (SPI)

Interfaces & Ports:

- GPIO
- Serial Port
- I2C
- SPI

Refer to the following url for details：

- Homepage: http://wiki.friendlyelec.com/wiki/index.php/FriendlyThings
- Examples: https://github.com/friendlyarm/friendlythings-examples
- Guide to API: http://wiki.friendlyelec.com/wiki/index.php/FriendlyThings_APIs

12 Connect NanoPC-T3-Plus to FriendlyARM LCD Modules

- Android

Here are the LCDs that are supported under Android:S430, S700/S701, S702, HD700, HD702, HD101 and X710 all of which are LCDs with capacitive touch.

- FriendlyCore & Lubuntu Desktop

Here are the LCDs that are supported under FriendlyCore and Lubuntu Desktop:S430, S700/S701, S702, HD700, HD702, HD101 and X710 all of which are LCDs with capacitive touch;
W35B, H43, P43, S70D and Matrix 2.8" SPI Key TFT LCD all of which are LCDs with resistive touch
All these LCD's tech details can be obtained on our wiki site:LCDModules (http://wiki.friendlyelec.com/wiki/index.php/Main_Page#LCDModules)

13 Schematics & Mechanical drawing

- Schematic(NanoPC-T3 Plus 1709 Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/d/d0/NanoPC-T3_Plus_1709-Schematic.pdf))
- PCB Dimension(NanoPC-T3_Plus_1709_Drawing(dxf).zip (http://wiki.friendlyelec.com/wiki/images/9/9e/NanoPC-T3_Plus_1709_Drawing%28dxf%29.zip))
- Datasheet （SEC_S5P6818X_Users_Manual_preliminary_Ver_0.00.pdf (http://wiki.friendlyelec.com/wiki/images/8/8b/SEC_S5P6818X_Users_Manual_preliminary_Ver_0.00.pdf)）

14 Source Code and Image Files Download Links

- Image File: [2] (http://download.friendlyelec.com//NanoPC-T3-Plus)
- Source Code: [3] (https://github.com/friendlyarm)

15 Tech Support

If you have any further questions please visit our forum http://www.friendlyarm.com/Forum/ and post a message or email us at techsupport@friendlyarm.com. We will endeavor to get back to you as soon as possible.

16 Update Log

16.1 2023-01-09

16.1.1 FriendlyCore：

- optimized the systemd service

16.2 2020-10-26

- **FriendlyCore, Lubuntu：**

Fix Bluetooth stability issue

16.3 2019-12-27

- **FriendlyWrt：**

Upgrade to OpenWrt r19-snapshot 64bit, support Docker CE

- **eflasher：**

1) Supports flashing only some files, such as updating only the kernel and uboot in emmc
2) Added gui option to disable overlay filesystem
3) Add command line parameters to achieve one-click installation without interaction
4) Fix the issue that the same mac address will appear on different devices after backup and restore image
5) UI interface can now be configured with title, hide interface menus and buttons

16.4 2019-11-26

- **FriendlyCore:**

Pre-installed OpenCV 4.1.2

16.5 2019-11-14

- **Introducing a new system FriendlyWrt**:

FriendlyWrt is a customized OpenWrt system developed by FriendlyElec. It is open source and suitable for applications in IoT, NAS etc.
Please refre: http://wiki.friendlyelec.com/wiki/index.php/How_to_Build_FriendlyWrt

- **FriendlyCore, Lubuntu updated as follows:**

1) Added support for new 4.3-inch screen YZ43
2) Compile bcmdhd as a module.

- **Android7 update is as follows:**

1) Added support for new 4.3-inch screen YZ43
2) Optimize the touch experience when using HD900 screen under Android 7 system

16.6 2019-10-18

- **Android7, FriendlyCore, Lubuntu：**

Fixed audio playback issue.

16.7 2019-09-30

- **Android7 updated as follows:**

1）Added support for Android hardware access library (named FriendlyThing), support access to hardware resources such as GPIO, PWM, RTC, serial port and watchdog, providing open source demo
2) Added support for camera CAM500B (OV5640)
3) Added support for LCD W500 (800x480)
4) Fixed LCD-S430 compatibility issues

- **FriendlyCore, FriendlyDesktop updated as follows:**

1) Kernel version updated to v4.4.172, same as Android 7
2) Added Docker support, support 32bit and 64bit file systems
3) Kernel configuration items are optimized to enable more features and device drivers

16.8 2019-07-18

- **Introducing a new system Android 7.1.2**

1) Features similar to the old version of Android 5, support 4G, WiFi, Ethernet, Bluetooth, etc.
2) Kernel version: 4.4.172
3) Known issue: The camera is not working yet

- **Android/FriendlyCore/Lubuntu updated as follows：**

1) Fix an issue where HD101B can't be touched in some cases
2) Fix GPIO configuration of Power key
3) Solve the problem of too small volume: the volume of the DAC is changed from -20dB to -6dB during playback.
4) Add more models of USB Wi-Fi support, built-in driver rtl8821CU.ko, rtl88XXau.ko

- **Updates for Lubuntu only：**

1) Modify Lubuntu's Power key behavior to (without pop-ups) shut down directly
2) Add script xrotate.sh to simplify screen rotation settings (Note: screen rotation will lose performance)

- **The following updates are only available for NanoPC T3/T3+, Smart6818:**

Support for reading Ethernet Mac addresses from the onboard EEPROM, only supports the following systems: FriendlyCore, Lubuntu, Android7

16.9 2019-06-25

Linux(Ubuntu 16.04/18.04) uses OverlayFS to enhance filesystem stability.

16.10 2019-06-03

1) Configure LED1 to be in heartbeat mode
2) Fix HDMI 1080P may have no display problem in some cases
3) Fix the issue that mysql cannot be installed under Linux
4) Fix the issue that the 1-wire touch resistance screen cannot be used under lubuntu

16.11 2019-01-24

1) Update uboot-v2014.07, uboot-v2016.01 for HD702V LCD
2) Adjust Qt5 font path

16.12 2018-12-17

- **Android5 updated as follows:**

1) Add support for 4G network, support module: Quectel EC20
2) Add audio setting UI, you can set the default output to headphones or HDMI
3) Synchronously turn off the backlight of the one-line touch screen when the system Shutdown

- **FriendlyCore updated as follows:**

1) Add OV5640 camera support
2) Update BL1 to improve system startup stability

- **Lubuntu updated as follows:**

1) Add Chrome-browser browser, support web page 1080P hardware decoding, support WebGL
2) Set the audio output channel to HDMI by default (can be changed via /etc/asound.conf)
3) Update BL1 to improve system startup stability
4) Fixed some issues regarding the package error in the previous version
5) Adjust DPMS settings, turn off automatic sleep by default

16.13 April-28-2016

- Released English version

16.14 June-30-2016

- Added sections 5.2.4 and 8

16.15 Sep-27-2016

- Added section 9
- Updated sections 5.2.2 and 8.2

16.16 Nov-2-2016

- Updated sections 6.4 and 11

16.17 June-20-2017

- Updated sections 6.2 and 6.3: wireless connection and setting up WIFI AP
- Updated section 8.4.1: added compiling kernel for UbuntuCore
- Added section 3: software features
- Added section 7: UbuntuCore
- Added section 9.5: LCD support

16.18 March-28-2018

- Updated sections 6.10

Retrieved from "https://wiki.friendlyelec.com/wiki/index.php?title=NanoPC-T3_Plus&oldid=26146"

- This page was last modified on 3 July 2024, at 08:14.