NanoPi M3

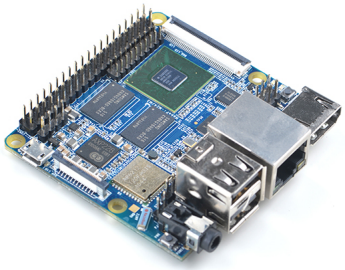
From FriendlyELEC WiKi

查看中文

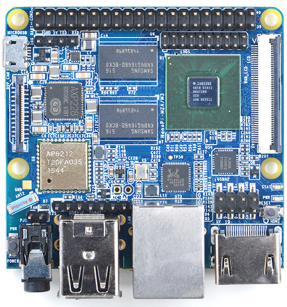
Contents

1 Introduction

- The NanoPi M3 is another ARM board developed by FriendlyARM for makers, hobbyists and hackers.
- The NanoPi M3 uses the Samsung Octa-Core Cortex-A53 S5P6818 SoC. Its dynamic frequency scales from 400M up to 1.4GHz. It has Gbps Ethernet port, built-in WiFi and Bluetooth. Its AXP228 PMU supports software power-off, sleep and wakeup functions. In addition the NanoPi M3 has MicroUSB port for power supply and on-board porcelain antenna.
- On this tiny board various ports and interfaces are integrated. It has DVP Camera/LVDS/HDMI/LCD interfaces, Gbps Ethernet, I2S, 3.5mm audio jack, four USB ports and a serial debug port.



Overview

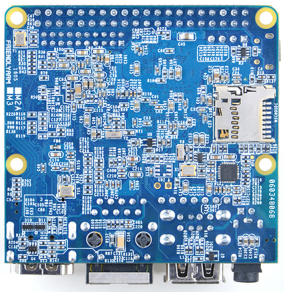


Front

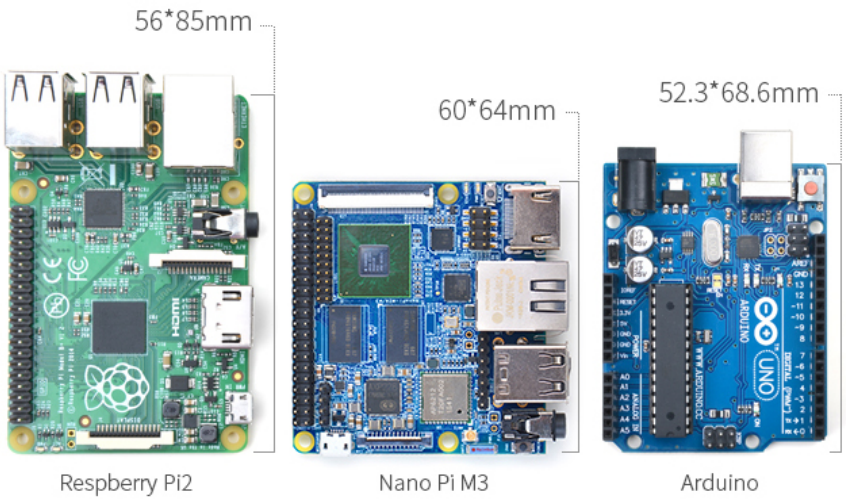
2 Hardware Spec

- SoC： Samsung S5P6818 Octa-Core Cortex-A53, 400M Hz - 1.4G Hz
- PMU Power Management: Implemented by a Cortex-M0 MCU, support software power-off, and RTC alarm power-on functions
- System Memory： 1GB 32bit DDR3 RAM
- Storage： 1 x MicroSD Card Socket

- Ethernet： Gbit Ethernet(RTL8211E)
- WiFi： 802.11b/g/n
- Bluetooth： 4.0 dual mode
- Antenna： onboard porcelain antenna for WiFi and Bluetooth
- Audio： 3.5 mm audio jack / via HDMI
- I2S: 7pin, 2.54mm pitch pin-header
- USB Host： 4 x USB 2.0 Host, two type A ports and two 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
- LVDS: 20pin 2.00mm pitch pin-header
- DVP Camera： 0.5mm pitch 24 pin FPC seat
- GPIO： 2.54 mm pitch 40 pin-header
- Serial Debug Port： 2.54mm pitch 4pin-header
- User Key: K1 (power), Reset
- LED： 1 x power LED and 1 x status LED
- RTC Battery： RTC Battery Header
- Power： DC 5V/2A
- PCB： 8 Layer, ENIG
- Dimension： 64 mm x 60 mm
- Temperature measuring range: -30℃ to 80℃
- OS/Software: uboot, Android and Debian



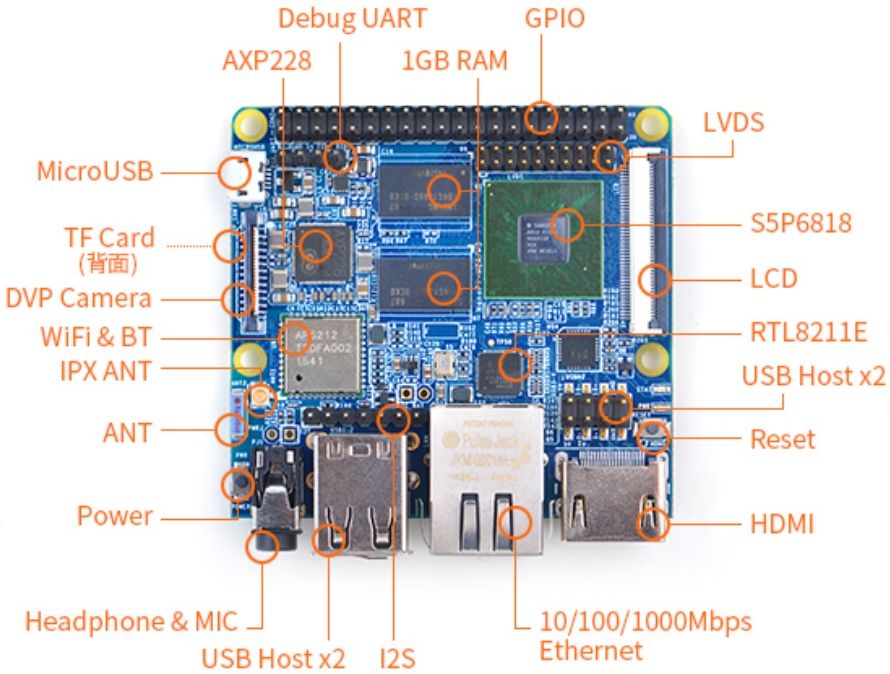
Back



3 Diagram, Layout and Dimension

3.1 Layout

- **GPIO Pin Spec**



NanoPi-M3 Layout

Pin#	Name	Pin#	Name
1	SYS_3.3V	2	VDD_5V
3	I2C0_SDA	4	VDD_5V
5	I2C0_SCL	6	DGND
7	GPIOD8/PPM	8	UART3_TXD/GPIOD21
9	DGND	10	UART3_RXD/GPIOD17
11	UART4_TX/GPIOB29	12	GPIOD1/PWM0
13	GPIOB30	14	DGND
15	GPIOB31	16	GPIOC14/PWM2
17	SYS_3.3V	18	GPIOB27
19	SPI0_MOSI/GPIOC31	20	DGND
21	SPI0_MISO/GPIOD0	22	UART4_RX/GPIOB28
23	SPI0_CLK/GPIOC29	24	SPI0_CS/GPIOC30
25	DGND	26	GPIOB26
27	I2C1_SDA	28	I2C1_SCL
29	GPIOC8	30	DGND
31	GPIOC7	32	GPIOC28
33	GPIOC13/PWM1	34	DGND
35	SPI2_MISO/GPIOC11	36	SPI2_CS/GPIOC10
37	AliveGPIO3	38	SPI2_MOSI/GPIOC12
39	DGND	40	SPI2_CLK/GPIOC9

The M3's GPIO pin spec is slightly different from the NanoPi 2's. Here is a comparison table:40 pins GPIO comparison table

- **Debug Port（UART0）**

Pin#	Name
1	DGND
2	VDD_5V
3	UART_TXD0
4	UART_RXD0

- **DVP Camera Interface Pin Spec**

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

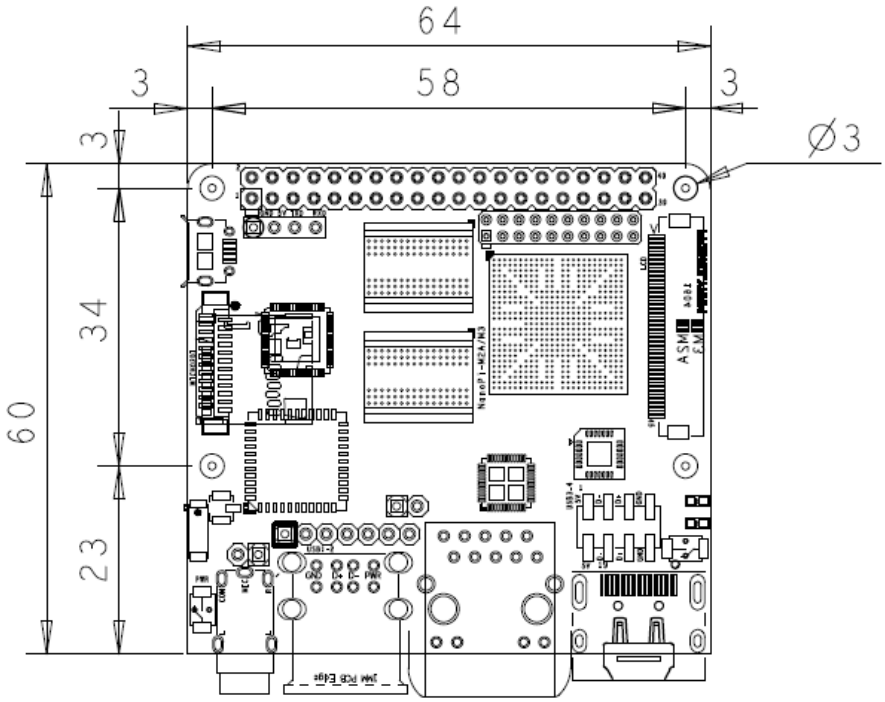
- **RGB LCD Interface Pin Spec**

Pin#	Name	Description
1, 2	VDD_5V	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

Notes

1. SYS_3.3V: 3.3V power output
2. VDD_5V: 5V power input/output. The input range is 4.7V ~ 5.6V
3. All pins are 3.3V, output current is 5mA
4. For more details refer to the document: NanoPi-M2A-M3-1712-Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/a/ac/NanoPi-M2A-M3-1712-Schematic.pdf)

3.2 Board Dimension



For more details refer to the document:PCB dxf file (http://wiki.friendlyelec.com/wiki/images/9/9c/NanoPi-M2A-M3-1604-pcb-dxf.zip)

4 Get Started

4.1 Essentials You Need

Before starting to use your NanoPi M3 get the following items ready

- NanoPi M3
- microSD Card/TFCard: Class 10 or Above, minimum 8GB SDHC
- microUSB power. A 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 TF Cards We Tested

To make your device boot and run fast we highly recommend you use a Class10 8GB SDHC TF card or a better one. The following cards are what we used in all our test cases presented here:

- Sandisk MicroSDHC V30 32GB Extreme Pro **(Developer choice)**

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

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

SanDisk 闪迪

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

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

4.3 Make an Installation SD Card

4.3.1 Boot NanoPi M3 from SD Card

Get the following files from the download link (http://download.friendlyelec.com/nanopim3) to download necessary files:

- Get a 4G SDHC card and backup its data if necessary.

Image Files:	
s5p6818-debian-sd4g-20160426.img	Debian image files
s5p6818-android-sd4g-20160426.img	Android image files
s5p6818-ubuntu-core-qte-sd4g-20160426.img	Ubuntu Core + QT image files
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 NanoPi M3's boot slot, press and hold the boot key and power on (with a 5V/3A power source). If the green LED is on and the blue LED is blinking this indicates your NanoPi M3 has successfully booted.

4.3.2 Under Linux Desktop

- 1) Insert your microSD card into your host 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 is recognized as "/dev/sdc". Or you can check that by commanding "cat /proc/partitions"

- 2) Download Flashing Script

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

- 3) Flash Android Firmware to MicroSD Card

```
su
./fusing.sh /dev/sdx
```

(Note: you need to replace "/dev/sdx" with the device name in your system) When you do “git clone” you have to hit “Y” within 10 seconds after it prompts you to download image files otherwise you will miss the download.

- 4) Flash Debian Firmware to MicroSD Card

```
./fusing.sh /dev/sdx debian
```

4.3.3 Extend NanoPi M3's TF Card Section

- When Debian/Ubuntu is loaded the TF 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 TF 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)

4.3.4 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 NanoPi M3 is connected an HDMI monitor and it runs Android it will automatically set the resolution to an appropriate HDMI mode by checking the "EDID". If your NanoPi M3 is connected an HDMI monitor and it runs Debian by default it will set the resolution to HDMI 720P. If you want to modify the resolution to 1080P modify your kernel's configuration as explained above.

4.4 Update Image Files in MicroSD Card From PC Host

If you want to make some changes to the image files in your MicroSD card follow steps below otherwise you can skip this section.
Insert your MicroSD card into a PC host running Linux, mount the boot and rootfs sections of the SD card and follow the steps below:
1) If you want to change your kernel command line parameters you can do it via the fw_setevn utility under "sd-fuse_s5p6818/tools".
Check the current Command Line

```
cd sd-fuse_s5p6818/tools
./fw_printenv /dev/sdc | grep bootargs
```

Android 5.1.1_r6 starts SELinux. By default it is enforcing. You can change it this way:

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

This sets it to "permissive". The "XXX" stands for the original bootargs' value.

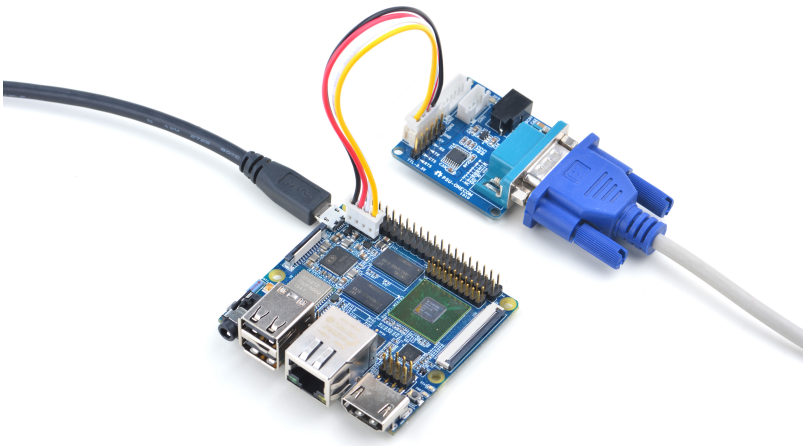
2) Update Kernel
Our customized uboot will check the LCD type when it boots.
For Android it doesn't make any difference which display device is detected. You can use your generated uImage to replace the existing one under "boot".
For Debian if your generated kernel is for an LCD you need to replace the existing uImage or if your kernel is for HDMI you need to replace the existing uImage.hdmi.

4.5 Run Android or Debian

- Insert a MicroSD card with Android/Debian image files into your NanoPi M3, connect the NanoPi M3 to an HDMI monitor and a 5V/2A power source the NanoPi M3 will be automatically powered up. If you can see the PWR LED is on and LED1 is flashing it means your board is working and you will see Android/Debain loading on the HDMI monitor.

1) If you connect the NanoPi M3 to an HDMI monitor you need to use a USB mouse and a USB keyboard to operate. If you connect it to an LCD with capacitive touch you can operate directly on the LCD.
2) If you want to do kernel development you need to use a serial communication board, ie a PSU-ONECOM board, which will allow you to operate the board via a serial terminal.

- Here is a setup where we connect a NanoPi M3 to a PC running Ubuntu and Minicom via a serial cable you will see system messages output to the PC’s minicom terminal:



- Under Debian the password for "root" is "fa".

5 Working with Debian

5.1 Ethernet Connection

- If the NanoPi M3 is connected to a network via Ethernet before it is powered on, it will automatically obtain an IP after it is powered up.

5.2 Wireless Connection

Open the file "/etc/wpa_supplicant/wpa_supplicant.conf" with vi or gedit and append the following lines:

```
network={
        ssid="YourWiFiESSID"
        psk="YourWiFiPassword"
}
```

The "YourWiFiESSID" and "YourWiFiPassword" need to be replaced with your actual ESSID and password.
Save, exit and run the following commands your board will be connected to your specified WiFi.

```
ifdown wlan0
ifup wlan0
```

If your WiFi password has special characters or you don't want your password saved as plain text you can use "wpa_passphrase" to generate a psk for your WiFi password. Here is how you can do it:

```
wpa_passphrase YourWiFiESSID
```

Following the prompt type in your password. If you open the file "/etc/wpa_supplicant/wpa_supplicant.conf" you will find that your password has been updated and you can delete your clear-text password.

If the system's WiFi AP mode is on it cannot search and connect to a wireless router. You need to turn off the WiFi AP mode by following the instructions below:

```
su
turn-wifi-into-apmode no
```

5.3 Setup Wi-Fi AP

You can follow the steps below to setup WiFi AP:

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

Reboot the system as prompted. By default the AP's name is "nanopi2-wifiap"

and the password is 123456789.

Now you are able to find the "nanopi2-wifiap" from a host PC and connect to it. If a connection is successful you will be able to SSH to this NanoPi M3 at "192.168.8.1":

```
ssh root@192.168.8.1
```

The password for it is "fa".

To make SSH session run faster turn off the WiFi's power saving mode by using the following command:

```
iwconfig wlan0 power off
```

You can check the WiFi mode via the following command:

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

If the result is "2" it means it is currently working as a WiFi AP.If you want to switch back to the Station mode you can do it this way:

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

5.4 Bluetooth

Here are the steps to transfer a file from T2 to a mobile phone. Run the following command to search a surrounding Bluetooth device:

```
hcitool scan
```

In our example a mobile phone was detected and the following messages were listed:
Scanning ...
38:BC:1A:B1:7E:DD MEIZU MX4

These messages indicated that a MEIZU MX4 mobile phone was detected. We then checked the Bluetooth services this phone supported with its MAC address presented in front of its device name

```
sdptool browse 38:BC:1A:B1:7E:DD
```

Note: you need to use your device's name and its MAC address when you run these commands.

The command listed all the services the phone supported. We needed the "OBEX Object Push" service which is for file transfers.

```
Service Name: OBEX Object Push
Service RecHandle: 0x1000b
Service Class ID List:
"OBEX Object Push" (0x1105)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 25
"OBEX" (0x0008)
Profile Descriptor List:
"OBEX Object Push" (0x1105)
Version: 0x0100
```

From the above messages we could get the channel number 25 for the "OBEX Object Push" service. We input this number to the "ussp-push" by running the following command:

```
ussp-push 38:BC:1A:B1:7E:DD@25 example.jpg example.jpg
```

Note: you need to use your device's name, its MAC address and channel number when you run these commands.

Usually after the above commands are run a popup window will show on the phone that communicates with T2 and you can start file transfers.

Common Issues:
1) If T2 cannot find a Bluetooth device you can try this command to restart its Bluetooth:

```
rfkill unblock 0
```

2) If any of these commands is not installed you can try this command to install it:

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

5.5 Install Debian Packages

We provide a Debian Jessie image. You can install Jessie's packages by commanding "apt-get". If this is your first installation you need to update the package list by running the following command:

```
apt-get update
```

You can install your preferred packages. For example if you want to install

an FTP server you can do this:

```
apt-get install vsftpd
```

Note: you can change your download server by editting "/etc/apt/sources.list". You can get a complete server list from [1] (http://www.debian.org/mirror/list). You need to select the one with "armhf".

5.6 Audio Output from HDMI or 3.5mm Jack under Debian

Our default Debian image for the NanoPi M3 doesn't support audio output. If you want to enable this function you need to install the alsa package.

- Make sure your Debian OS is our latest version and your board has access to the internet;
- Power up your board and run the following commands on your board's commandline utility to install the alsa package:

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

- After the installation is done copy a ".wav" audio file to your NanoPi M3, connect your M3 to a earphone or speaker and try playing this audio file(By default Debian's audio output is from the 3.5mm audio jack):

```
aplay music.wav
```

- By default Debian's audio output is from the 3.5mm audio jack. If you want audio output from the HDMI you need to change the setting by editing the "/etc/asound.conf" file:

```
pcm.!default {
    type hw
    card 1
    device 0}

ctl.!default {
    type hw
    card 1}
```

card 0 stands for the 3.5mm audio jack and card 1 stands for the HDMI audio. After you make your change reboot your board to make it effective.

6 Make Your Own OS Image

6.1 Install Cross Compiler

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)
```

6.2 Compile U-Boot

Download the U-Boot source code and compile it. Note that the github's branch is nanopi2-lollipop-mr1:

```
git clone https://github.com/friendlyarm/uboot_nanopi2.git
cd uboot_nanopi2
git checkout nanopi2-lollipop-mr1
make s5p6818_nanopi3_config
make CROSS_COMPILE=arm-linux-
```

After your compilation succeeds a u-boot.bin will be generated. If you want to test it flash it to your installation SD card via fastboot. Here is how you can do it:
1) On your host PC run "sudo apt-get install android-tools-fastboot" to install the fastboot utility;

2) Connect your NanoPi M3 to your host PC via a serial cable (e.g. PSU-ONECOME). Press the enter key within two seconds right after you power on your NanoPi M3 and you will enter uboot's command line mode;
3) After type in "fastboot" and press "enter" you will enter the fastboot mode;
4) Connect your NanoPi M3 to this host PC via a microUSB cable and type in the following command to flash u-boot.bin;

```
fastboot flash bootloader u-boot.bin
```

Warning: you cannot update this SD card by commanding "dd". This command will cause trouble when booting the NanoPi M3.

6.3 Prepare mkimage

You need the mkimage utility to compile a U-Boot source code package. Make sure this utility works well on your host before you start compiling a uImage.
You can install this utility by either commanding "sudo apt-get install u-boot-tools" or following the commands below:

```
cd uboot_nanopi2
make CROSS_COMPILE=arm-linux- tools
sudo mkdir -p /usr/local/sbin && sudo cp -v tools/mkimage /usr/local/sbin
```

6.4 Compile Linux Kernel

6.4.1 Compile Kernel

- Download Kernel Source Code

```
git clone https://github.com/friendlyarm/linux-3.4.y.git
cd linux-3.4.y
git checkout nanopi2-lollipop-mr1
```

The NanoPi M3's kernel source code lies in the "nanopi2-lollipop-mr1" branch

- Compile Android Kernel

```
make nanopi3_android_defconfig
touch .scmversion
make uImage
```

- Compile Debian Kernel

```
make nanopi3_linux_defconfig
```

```
touch .scmversion
make uImage
```

After your compilation succeeds a uImage will be generated in the "arch/arm/boot/uImage" directory. This kernel is for HDMI 720P. You can use it to replace the existing uImage.hdmi.
If you want to generate a kernel for HDMI 1080P you can do it this way:

```
touch .scmversion
make nanopi3_linux_defconfig
make menuconfig
  Device Drivers -->
    Graphics support -->
      Nexell Graphics -->
        [ ] LCD
        [*] HDMI
        (0)   Display In  [0=Display 0, 1=Display 1]
              Resolution (1920 * 1080p)  --->
make uImage
```

After your compilation succeeds a uImage will be generated for HDMI 1080P. You can use it to replace the existing uImage.

6.4.2 User Your Generated Kernel

- Update the kernel file in SD card

If you use an SD card to boot Android you can copy your generated uImage file to your SD card's boot partition(e.g. section 1 /dev/sdX1).
If you use an SD card to Debian and you generated a uImage for an HDMI monitor you can use that uImage to replace the uImage.hdmi file in the SD card's boot partition. If you use an SD card to Debian and you generated a uImage for an LCD you can use that uImage to replace the uImage file in the SD card's boot partition.

6.4.3 Compile Kernel Modules

Android contains kernel modules which are in the "/lib/modules" directory in the system section. If you want to add your own modules to the kernel or you changed your kernel configurations you need to recompile these new modules.
Compile Original Kernel Modules:

```
cd linux-3.4.y
make CROSS_COMPILE=arm-linux- modules
```

Here we have two new modules and we can compile them by following the commands below:

```
cd /opt/FriendlyARM/s5p6818/android
./vendor/friendly-arm/build/common/build-modules.sh
```

The "/opt/FriendlyARM/s5p6818/android" directory points to the top directory of Android source code. You can get more details by specifying option "-h".
After your compilation succeeds new modules will be generated.

6.5 Compile Android

- 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 libncurses5-dev zlib1g-dev gawk minicom
```

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

- Download Source Code

You need to use repo to get the Android source code. Refer to https://source.android.com/source/downloading.html 。

```
mkdir android && cd android
repo init -u https://github.com/friendlyarm/android_manifest.git -b nanopi3-lollipop-mr1
repo sync
```

The "android" directory is the working directory.

- Compile System Package

```
source build/envsetup.sh
lunch aosp_nanopi3-userdebug
make -j8
```

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

filename	partition	Description
boot.img	boot	-
cache.img	cache	-
userdata.img	userdata	-
system.img	system	-
partmap.txt	-	partition file

- Flash Image to SD Card

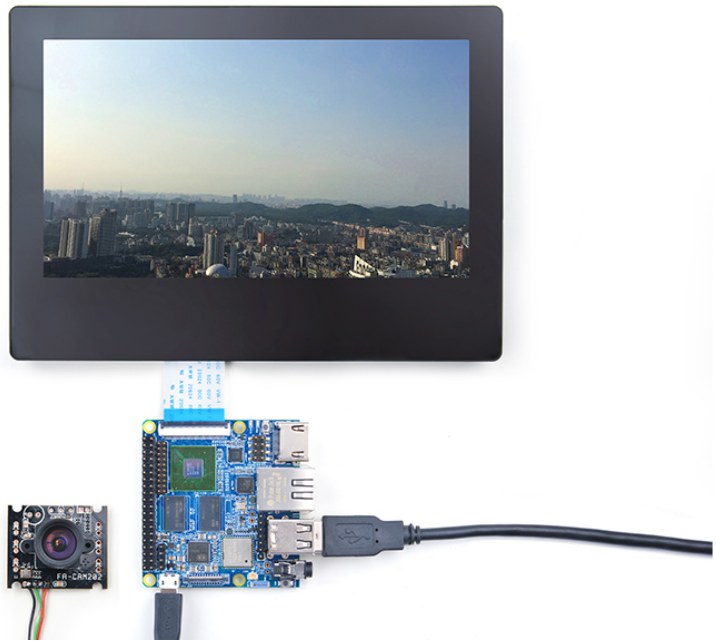
If you want to boot your board from an SD card you need to copy your

generated image file to the "sd-fuse_s5p6818/android/" directory and flash it to your SD card with our script. For more details refer to #Under Linux Desktop.

7 Connect NanoPi M3 to External Modules

7.1 Connect NanoPi M3 to USB Camera(FA-CAM202)

- In this use case the NanoPi M3 runs Debian. If you connect your NanoPi M3 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.



7.2 Connect NanoPi M3 to CMOS 5M-Pixel Camera

Check this link [2] (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_CAM500B) for details about the camera CAM500B we tested in this case

- In this use case the NanoPi M3 runs Android5.1. If you connect your NanoPi M3 to our LCD or an HDMI monitor after Android is fully loaded click on the "Camera" icon and the application will be started.

- There is a camera utility "nanocams" in Debian/Ubuntu. You can preview and save 40 frames to an image file by running the following command

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

For more details about the usage of the utility you can command "nanocams -h". If you want to download the source code you can get it from our github:

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

7.3 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 NanoPi M3 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 NanoPi M3:

- Firstly you need to make sure your NanoPi M3 is connected to the internet.Login to your NanoPi M3 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 NanoPi M3. You can test your camera with NanoPi M3'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 NanoPi M3 to USB Keyboard & Run the Following Command:

```
./demo
```

opencv is successfully started
OpenCV-M3

8 Android Hardware Access

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)

Accessible Ports：

- GPIO
- Serial Port
- I2C
- SPI

Please 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

9 Source Code and Image Files Download Links

- Image File: [3] (http://download.friendlyelec.com/nanopim3)
- Source Code: [4] (https://github.com/friendlyarm)

10 Schematics & Mechanical drawing

- Schematic(NanoPi-M2A-M3-1604-Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/4/4c/NanoPi-M2A-M3-1604-Schematic.pdf))
- Schematic(NanoPi-M2A-M3-1712-Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/a/ac/NanoPi-M2A-M3-1712-Schematic.pdf))
- PCB Dimension(NanoPi-M2A-M3-1604-pcb-dxf.zip (http://wiki.friendlyelec.com/wiki/images/9/9c/NanoPi-M2A-M3-1604-pcb-dxf.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)）

11 More OS Support

11.1 DietPi

DietPi
#LightweightJustice for your SBC

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

12 Resources

- SOC Datasheet: S5P6818 Datasheet (http://wiki.friendlyelec.com/wiki/images/8/8b/SEC_S5P6818X_Users_Manual_preliminary_Ver_0.00.pdf)
- Schematic: NanoPi-M2A-M3-1604-Schematic.pdf (http://wiki.friendlyelec.com/wiki/images/4/4c/NanoPi-M2A-M3-1604-Schematic.pdf)
- PCB Dimension: PCB DXF file (http://wiki.friendlyelec.com/wiki/images/9/9c/NanoPi-M2A-M3-1604-pcb-dxf.zip)
- Matrix Modules & Wiki Sites:
  - Button (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Button)
  - LED (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_LED)
  - A/D Converter (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Analog_to_Digital_Converter)
  - Relay (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Relay)
  - 3-Axis Digital Accelerometer (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_3-Axis_Digital_Accelerometer)
  - 3-Axis Digital Compass (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_3-Axis_Digital_Compass)
  - Temperature Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Temperature_Sensor)
  - Temperature & Humidity Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Temperature_and_Humidity_Sensor)
  - Buzzer (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Buzzer)
  - Joystick (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Joystick)
  - I2C(PCF8574)+LCD1602 (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_I2C_LCD1602_Keypad)
  - Sound Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Sound_Sensor)
  - Ultrasonic Ranger (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Ultrasonic_Ranger)
  - GPS (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_GPS)
  - Matrix - Compact Kit (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Compact_Kit)
  - Fire Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Fire_Sensor)
  - CAM500A Camera (http://wiki.friendlyelec.com/wiki/index.php/

Matrix_-_CAM500A)
- BAll Rolling Switch (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_BAll_Rolling_Switch)
- 2'8 SPI Key TFT 2.8" SPI LCD (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_2%278_SPI_Key_TFT)
- IR Counter (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_IR_Counter)
- IR Receiver (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_IR_Receiver)
- L298N Motor Driver (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_L298N_Motor_Driver)
- MQ-2 Gas Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_MQ-2_Gas_Sensor)
- MQ-3 Gas Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_MQ-3_Gas_Sensor)
- One_Touch_Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_One_Touch_Sensor)
- _Photoresistor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Photoresistor)
- _Potentiometer (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Potentiometer)
- Pressure & Temperature Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Pressure_and_Temperature_Sensor)
- RGB LED (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_RGB_LED)
- RTC (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_RTC)
- Rotary Encoder (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Rotary_Encoder)
- Soil Moisture Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Soil_Moisture_Sensor)
- Thermistor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Thermistor)
- USB WiFi (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_USB_WiFi)
- Water Sensor (http://wiki.friendlyelec.com/wiki/index.php/Matrix_-_Water_Sensor)

13 3D Printing File

- NanoPi M3 3D Printing Files:[5] (http://wiki.friendlyelec.com/wiki/index.php/File:M3%E5%AE%A2%E6%88%B7%E6%89%93%E5%8D%B0%E6%96%87%E4%BB%B6.7z)

14 Update Log

14.1 May-20-2016

- Released English version

14.2 July-05-2016

- Added sections 4.3.3, 7 and 10

14.3 Nov-02-2016

- Updated Sections 2, 4.3.1, 5.4 and 7.2
- Added Section 8

14.4 Nov-05-2016

- Updated Section 5.4

Retrieved from "https://wiki.friendlyelec.com/wiki/index.php?title=NanoPi_M3&oldid=25034"

Category: Pages with broken file links

- This page was last modified on 4 September 2023, at 11:05.