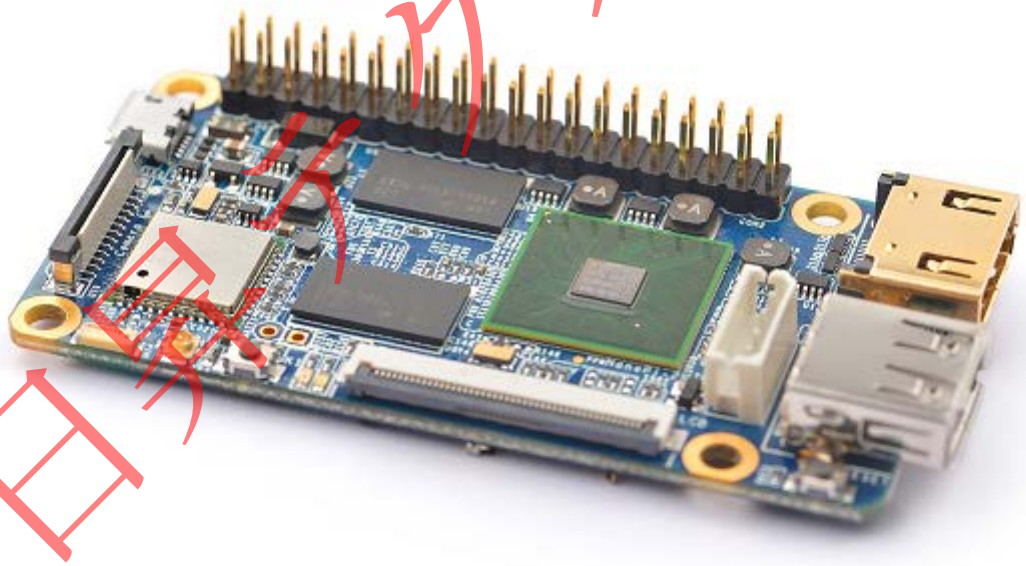
Cortex-A9 4コア S5P4418 ボード NanoPi2 簡易マニュアル

株式会社日昇テクノロジー
http://www.csun.co.jp
info@csun.co.jp

作成日 2016/1/27



• 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2015/11/20
2	Ver1.1	・Android 5.1.1 にバージョンアップ ・新しいイメージファイルでは LCD 検索をサポートしますので、元の LCD と HDMI 二つ分けているのを一つのイメージファイルになる ・Android ソースコードのダウンロードルートを nanopi2-lollipop-mr1 に変更 ・LCD 表示をサポートする Debian の uImage のコンパイル方法を追加	2015/12/14
3	Ver1.2	カメラモジュール、TF カードに関する説明分の追加	2016/1/27

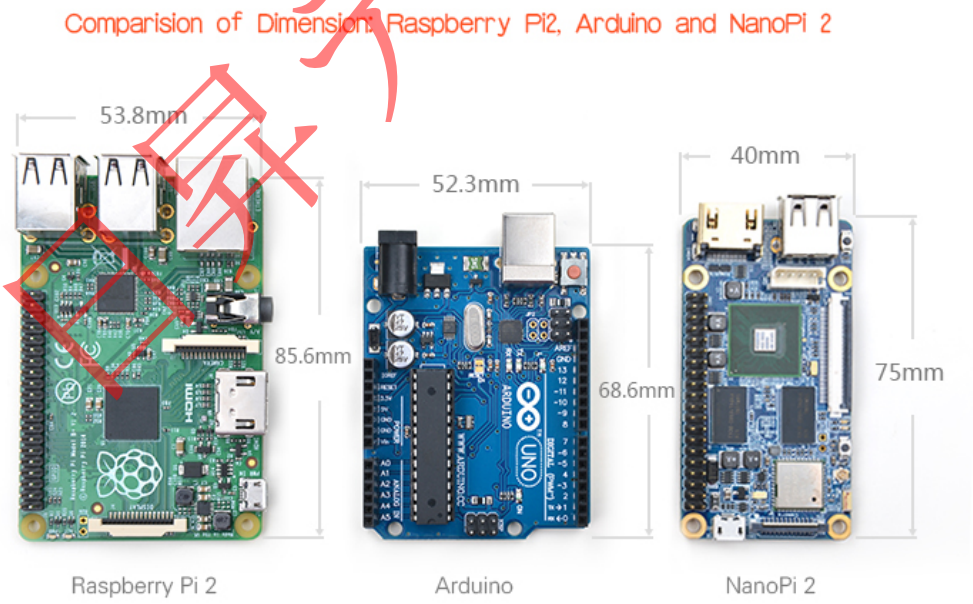
目次

1 紹介

NanoPi2はIoT設計のために開発された高性能のARMマスタコントロールボードである。 Samsungの Cortex-A9クアッド４コア S5P4418、1.4GHz、SoC 1G 32ビット DDR3を備えている。NanoPi2は802.11 b/g/nとBluetooth4.0をサポートするWiFiとBluetoothを内蔵している。TFカードからandroidと Debianシステムを実行することができ、HMDIとLCD I/Fを搭載している。Raspberry PiのGIPOと互換性がある。 PCBの寸法は75×40mmである。

2 主な仕様

CPU：Samsung S5P4418、動作周波数1.4GHz
RAM：1GB DDR3
統合 SDIO WiFi/Bluetoothモジュール
USB2.0 タイプA x1
デバッグ用シリアルポートx1
microSD Slot x2
microUSB x1：給電とデータの伝送をサポート
LCD I/F：0．5 mmピッチSMT FPCコネクタ、フルカラーLCDをサポート（RGB：8-8-8）
DVPカメラ I/F： 0.5mmピッチ省スペースタイプFPCソケット、ITU-R BT 601/6568ビット、I2CおよびIOを含む。
GPIO：2.54mmピッチ、40ピン、RasberryPiのGPIOと互換性がある 。UART、SPI、I2C、IOなどを含む。
ボタン：ユーザボタンx１、リセットボタンx１
LED: 電源LEDx1、ユーザLEDx1
PCBサイズ：75 x 40 mm
電源：DC 5V/2A
OS: Android、 Debian



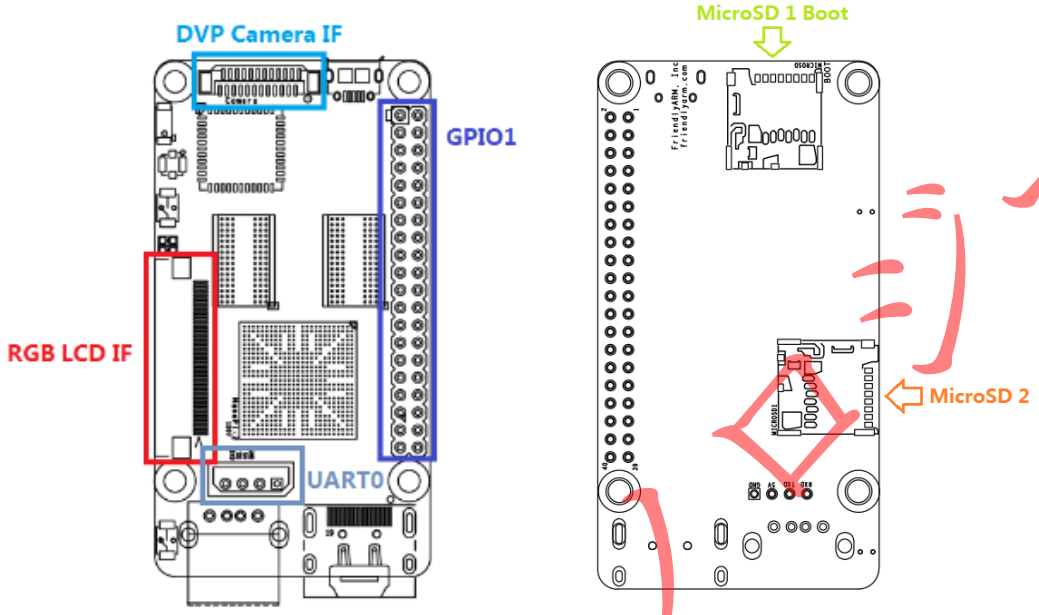
Comparision of Interfaces/Ports : Raspberry Pi 2,BeagleBone Black,NanoPi 2

Model	Raspberry Pi 2	BeagleBone Black	NanoPi 2
CPU Vendor	Broadcom	TI	Samsung
CPU Model	BCM2836	Sitara AM335x	S5P4418
CPU Core	Quad Core Cortex-A7	Single Core Cortex-A8	Quad Core Cortex-A9
CPU Clock	900MHz	1GHz	1.4GHz
GPU	VideoCore IV	PowerVR SGX530	Mali-400
Video Decoder	VideoCore IV 1080p video decoding for H.264, MPEG2* and VC1*1080p video encoding (H.264)* Extra license required	N/A	Max 1080P video decoding for H.264, VC-1, MPEG-1/2, MPEG-4, H.263P3, VP8, Theora, AVS, RV-8/9/10 and JPEG(8192 x 8192) Max 1080P video encoding for H.264, MPEG-4, H.263 and JPEG(8192 x 8192)
RAM	1GB DDR2	512M DDR3	1GB DDR3
eMMC	N/A	2/4GB	N/A
Ethernet	10/100M (USB to Ethernet chipset)	10/100M (Supported by SoC)	N/A
WiFi	N/A	N/A	Built-in 802.11 b/g/n
Bluetooth	N/A	N/A	Built-in Bluetooth 4.0
Antenna	N/A	N/A	Porcelain with IPEX socket
LCD Interface	Stand-alone MIPI Interface	RGB Interface (When connected to an HDMI monitor, it cannot connect to an LCD)	Stand-alone RGB Interface
Camera	CSI Interface SMD-1.0mm-15pin	N/A	DVP Interface SMD-0.5mm-24pin
HDMI	HDMI – A	HDMI – A	HDMI – A
Audio	Via HDMI Audio Jack	Via HDMI	Via HDMI
USB Host	4 x USB 2.0 Host (USB-A)	1 x USB 2.0 Host (USB-A)	1 x USB 2.0 Host (USB-A)
USB Client	N/A	1 x USB 2.0 Client (mini USB)	1 x USB 2.0 Client (micro USB)
Serial Debug	N/A	6 Pin single row pin header(6 x 2.54mm)	4 Pin single row pin header(4 x 2.54mm)
Micro SD	1 x Slot	1 x Slot	2 x Slot
Power Interface	Micro Usb	DC power jack	Micro USB for both power supply and data transmission
uboot	N/A	Open Source	Open Source
Android	N/A	Yes	Speedy
Debian/Linux	Yes	Yes	Yes
Dimension	Credit card sized 85.60 × 53.98 mm	Credit card sized 86.36 x 54.61mm	2/3 of credit card sized 75 X 40mm
Weight	45g	39.68g	22g
Language support			

3 インターフェースの配置及びサイズ

3.1 インターフェースの配置

3.1.1 GPIO1 ピン定義



Pin#	Name	Pin#	Name
1	VDD_SYS_3.3V	2	VDD_5V
3	I2C0_SDA	4	VDD_5V
5	I2C0_SCL	6	DGND
7	GPIOB28	8	UART3_TXD
9	DGND	10	UART3_RXD
11	GPIOB29	12	GPIOB26

13	GPIOB30	14	DGND
15	GPIOB31	16	PWM2
17	VDD_SYS_3.3V	18	GPIOB27
19	SPI0_MOSI	20	DGND
21	SPI0_MISO	22	PWM0
23	SPI0_CLK	24	SPI0_CS
25	DGND	26	PWM1
27	I2C1_SDA	28	I2C1_SCL
29	GPIOC8	30	DGND
31	SPI2_CLK	32	GPIOC28
33	SPI2_CS	34	DGND
35	SPI2_MOSI	36	GPIOC7
37	SPI2_MISO	38	ALIVEGPIO2
39	DGND	40	ALIVEGPIO3

3.1.2 Debug Port CON1 (UART0)

Pin#	Name
1	DGND
2	VDD_5V
3	TXD0
4	RXD0

3.1.3 DVP Camera IF ピン定義

Pin#	Name
1, 2	VDD_SYS_3.3V
7,9,13,15,24	DGND
3	SCL0
4	SDA0
5	GPIOB14
6	GPIOB16
8,10	NC

11	VSYNC
12	HREF
14	PCLK
16-23	Data bit7-0

3.1.4 RGB LCD IF ピン定義

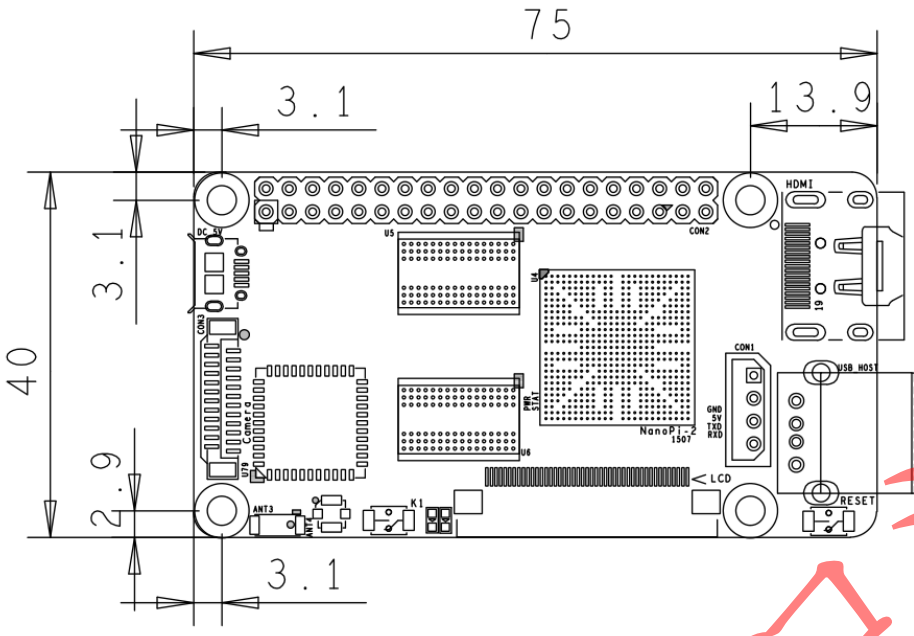
Pin#	Name
1, 2	VDD_5V
11, 20, 29,37,38,39,40,45	DGND
3-10	Blue LSB to MSB
12-19	Green LSB to MSB
21-28	Red LSB to MSB
30	GPIOB25
31	GPIOC15
32	XnRSTOUT Form CPU
33	VDEN

34	VSYNC
35	HSYNC
36	LCDCLK
41	SCL2
42	SDA2
43	GPIOC16
44	NC

説明

1. VDD_SYS_3.3V：3.3V電源の出力
2. VDD_5V：5 V電源入力/出力。電圧がMicroUSBより高い場合、ボードに給電、そうでない場合、ボードはMicroUSBから電源を取る。入力範囲：4.7～5.6V。
3. 更に詳しい情報については回路図：NanoPi-2-1507-Schematic.pdf をチェックしてください。

3.2 PCB サイズ



更に詳しい寸法については　NanoPi-2-1507-Dimesions(dxf).zip　をダウンロードしてください。

4 クイックスタート

4.1 ハードウェアの準備

・NanoPi2ボード
・microSDカード/ TFカード：Class10以上の8GB SDHCカードが必要
・microUSBインタフェースの外部電源、5V/2A
・HDMI入力サポートするディスプレイ或いはTV、或いはオプションのLCD液晶
・USBキーボード、USBマウス、同時に使う場合はUSB HUBも必要
・Linuxを実行するコンピュータ１台、オンラインネットワーキング、Ubuntu 14.04 64ビットシステムの使用を推奨

4.2 TFカードでテストする

NanoPiを起動させたTFカードを作る時、クラス10かそれ以上の8GB SDHDカードを推奨する。以下は試験実績のある高速TFカード。

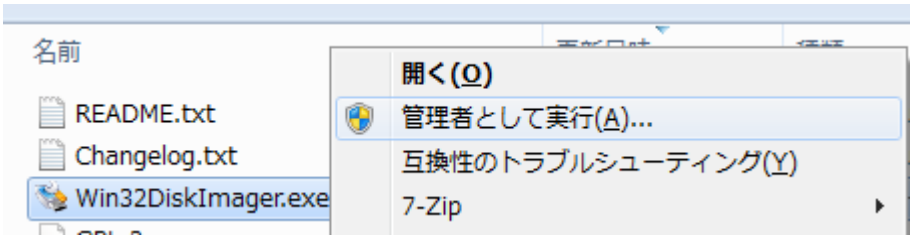
・Sandisk TF 8G クラス10 Micro/Sd高速TFカード

Sandisk TF128G MicroSDXC TF 128G クラス10 48MB/S

4.3 実行システムを持つ microSD カードを作成する

4.3.1 Windows 環境での作成

弊社HPからイメージファイル及びツールをダウンロードする。
Win32DiskImager.exeを右クリックして[管理者として実行（A）…]をクリックする。



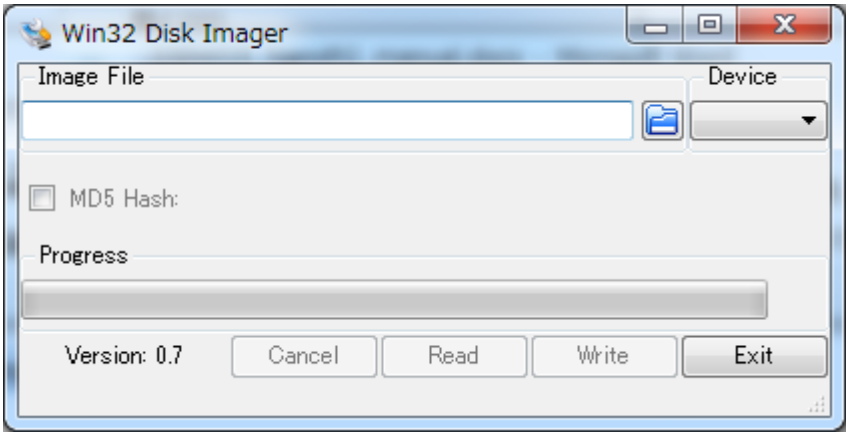


Image FileとDeviceのドライブを選択して、[Write]をクリックする。

下記URLからイメージファイル及びソースコードをダウンロードする。
https://drive.google.com/folderview?id=0BwAxPf5UWtWBNzNLRnNyRVN2V3c&usp=sharing

LCD 或いは HDMI の出力は下記のイメージファイルを使用	
nanopi2-debian-sd4g.img.zip	Debian イメージファイル
nanopi2-android-sd4g.img.zip	Android のイメージファイル
ボードのみで使用するユーザーは下記のファイルを使用	
nanopi2-debian-sd4g-wifiap.img.zip	Debian イメージファイル（デフォルトで、WiFi ホットスポットサービスが起動、ユーザーは VNC または SSH 経由でログインできる。）

4.3.2 Linux Desktop 環境での作成

1)microSDをUbuntuのパソコンに挿入　以下のコマンドでSDカードのデバイス名をチェックする

```
dmesg | tail
```

dmesgが「sdc：sdc1 sdc2」と類似した情報を出力する時、SDカード対応デバイス名は/dev/sdcになる。コマンドcat /proc/partitionsでも確認できる。

2)ファームウェアをダウンロードする

```
git clone https://github.com/friendlyarm/sd-fuse_nanopi2.git
czd sd-fuse_nanopi2
```

3)Androidの実行カードを作成する

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

（注：/dev/sdxを実際のSDカードのデバイスファイル名に変えてください）

初めて使う際、ダウンロードするか確認が必要。Yを押してダウンロードし、N或いは10秒間入力無い場合は取り消しする。

4)Debianの実行カードを作成する

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

4.4 パソコンでSDカード上のシステムの更新

システムを実行する前に、少し変更したい場合は、本節の内容をご参照ください。

作成したmicroSDカードをLinuxのパソコンに挿入して、SDカードのboot、rootfsをマウントして内容を変更できる。下記の場合変更が必要：

1） カーネルのコマンドラインパラメータを更新したい場合は、「sd-fuse_nanopi2/tools」の下にある、「fw_setenv」ツールを使用することができる。例えば、LCDがHD700であれば下記の方法で変更することができる。

現在のコマンドラインを確認する。

```
cd sd-fuse_nanopi2/tools
./fw_printenv /dev/sdc | grep bootargs
```

現在のAndroid 5.1.1_r6により SELinuxが有効になる。デフォルトモードは enforcing となり、Command Lineを通して変更することが可能。

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

直ぐに、permissive モードに変更でき、[XXX]は元の bootargs に置き換える必要がある。

2)カーネルの更新

新バージョンのUbootが起動時にLCDを認識した場合、SDカードのブートパーティションのuImage.hdmiを読み取る。

Androidにおいては、同じファイルであるため、直接新しいコンパイラのuImageで、SDカードのブートパーティションのファイルに交換する。

Debianにおいては、２つのファイルが異なるため、新しいコンパイラをサポートするLCD uImageで、直接SDカードのブートパーティションのファイルに交換する。HDMIのカーネルをサポートする場合は、uImage.hdmiに交換する。

4.5 Androidまたは Debian を実行する

microSDカードをNanoPi2に挿入し、HDMIモニターと接続して、電源（5V/2A）に接続すると、NanoPi2は自動的に起動する。青色LEDライトの点灯でシステムが起動していることが確認でき、またHDMIモニターには起動画面が表示される。

HDMIモニターで操作したい場合、USBマウス、キーボードが必要である。もしLCDと接続していれば、タッチパネルで操作可能。
カーネルを開発する場合、シリアルデバッグポート接続すれば、端末からNanoPi2を操作できる。
パスワード入力の提示がある場合、Debianのrootユーザーのデフォルトのパスワードは[fa] である。

4.6 VNC と SSH 経由で Debian にログイン

NanoPi2 を LCD・HDMI に接続せずに、 [-wifiap.img]のイメージファイルを実行した場合、WIFI 経由で携帯等の他のデバイスから [nanopi2-wifiap]の NanoPi2 にログインできる。ホットスポット wifiap のデフォルトパスワードは[123456789]。正常に NanoPi2 に接続した後、以下の URL から[VNCViewer]をダウンロード&インストールできる。VNC 経由で NanoPi2 にログインするには、IP アドレスとポートを 192.168.8.1:5901 に設定する必要がある。ディフォルトのパスワードは[fa123456]。

ユーザーログイン後のスクリーンショット

[SSH -l root 192.168.8.1] 経由でもログイン可能。[root]のデフォルトパスワードは[fa]である。SSH をスムーズにするには、WIFIの省電力モードをオフする。

```
iwconfig wlan0 power off
```

5 Debain システム

5.1 無線ネットワークに接続する

Debianがロードされた後、GUIの右上にあるネットワークアイコンをクリックすると、自動的に近くのWiFiホットスポットが検索される。リストからスポットを選択し、[Properties]をクリックする。
パスワードを入力、保存し、Connectをクリックする。

次の内容は[-wifiap.img]ファイルで実行されるNanoPi2のみに適用される。 デフォルトではWiFiのAP（アクセスポイント）モードはオンになっているため、無線ルーターに接続できない。以下の手順でWiFiのAPモードをオフにする。

接続する対象となる WiFi ルーターを設定する（SSH 経由で NanoPi にログイン）。次のコマンドを実行し、WiFi デバイスを確認する。[wlan] で始まるものが WiFi デバイスである。

```
ifconfig -a
```

デフォルトで[wlan0]は、WiFi デバイスである。[/etc/network/interfaces.d/]内に同じ名前のコンフィギュレーションファイル（例：[wlan0」ファイル等]を作成する必要があります。

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

wlan0 の内容は次のようになる。

```
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
```

上記の中で、[YourWiFiESSID]と[YourWiFiPassword]を実際のESSIDとパスワードに置き換える必要がある。最後に、下記コマンドでホットスポットモードをオフにする。rootユーザーとして実行する必要。コマンド

実行後ボードを再起動する。再起動したら、上記設定の通り自動的にWiFiに接続する。

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

5.2 Wi-Fi 無線ホットスポットの配置

WiFiホットスポットの配置を以下の手順で行う。

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

システムを再起動する。デフォルトのホットスポット名は[nanopi2-wifiap]で、パスワードは123456789。PCホストから[nanopi2-wifiap]に接続可能になる。接続が成功すれば、SSHをを介して192.168.8.1でNanoPi2に登録できる。

```
ssh root@192.168.8.1
```

パスワードは[fa]である。次のコマンドで無線LANモードを確認できる。

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

出力する数字が2であれば、現在無線ホットスポットモードとして機能していることを示す。
ステーションモードに切り替えたい場合、以下のコマンドを入力する。：

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

5.3 Bluetooth を使ってファイルを転送する

GUIの右上にあるBuluetoothのアイコンをクリックすると、メニューが表示される。[Make Discoverable]によってNanoPi2 が他のBuluetoothデバイスから検出可能になる。Devices...は検索画面を開き、近くのBluetoothデバイスを検索する（[Make Discoverable]は先に有効にする必要がある）。[Send Files to Divices]でNanoPi2が別のBuluetoothデバイス（NanoPI2とペア）にファイルを送ることができる。

5.4 Debian のパッケージソフトをインストールする

提供しているのは標準的なDebian jessieシステムである。apt-getなどのコマンドでパッケージソフトをインストールすることができる。初めてインストールする場合、まず以下のコマンドでパッケージソフトリストを更新する必要がある。

```
apt-get update
```

その後、パッケージソフトをインストールすることができる。例えばFTPサーバーをインストールするには以下のコマンドを使用する。

```
apt-get install vsftpd
```

/etc/apt/sources.listを編集することで、ダウンロードサーバーを変更することができる。
http://www.debian.org/mirror/listから全てのサーバーリストが取得可能。［armhf］が付くリストを選択することが必要。

6 システムのコンパイル方法

6.1 クロスコンパイラをインストールする

先ず、コンパイラをダウンロードして解凍する。

```
git clone https://github.com/friendlyarm/prebuilts.git
sudo mkdir -p /opt/FriendlyARM/toolchain
sudo tar xf prebuilts/gcc-x64/arm-cortexa9-linux-gnueabihf-4.9.3.tar.xz -C /opt/FriendlyARM/toolchain
```

コンパイラのパスをPATHに追加する。viでvi ~/.bashrcを実行して、末尾に以下の内容を追加する。

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

~/.bashrcスクリプトを実行してカレントshellで有効にする。"."の後ろにスペースがある。

```
. ~/.bashrc
```

コンパイラは64ビットのため、32ビットのLinuxでは実行できない。
インストールの完了後、インストールが成功したかを確認できる。

```
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 U-Boot のコンパイル

U-Bootソースコードをダウンロードし、コンパイルする。ブランチは[nanopi2-lollipop-mr1]であることに注意する。

```
git clone https://github.com/friendlyarm/uboot_nanopi2.git
cd uboot_nanopi2
git checkout nanopi2-lollipop-mr1
make s5p4418_nanopi2_config
make CROSS_COMPILE=arm-linux-
```

コンパイルに成功した後、u-boot.binを取得する。Fastbootで、NanoPi2のSDカードのUbootを更新する。

手順は下記の通り：

1）PCでコマンド [sudo apt-get install android-tools-fastboot]でfastbootツールをインストールする。

2）シリアルデバッグセットでNanoPi2とPCを接続する。起動後２秒以内、シリアル端末でEnterを押して、u-bootのコマンドラインモードに入る。

3）u-bootのコマンドラインモードでfastbootコマンドを入力し、Enterを押してfastbootモードに入る。

4）microUSBケーブルでNanoPi2とPCを接続する。PC側で下記コマンドを入力してu-boot.binを書き込む。

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

注意点：直接ddコマンドでSDカードを更新してはいけない。正常に起動できなくなる可能性がある。

6.3 mkimage を用意する

カーネルをコンパイルするには u-boot の mkimage ツールが必要。因って、カーネル uImage をコンパイルする前に、PC 側で実行できることの確認が必要。

直接 sudo apt-get install u-boot-tools コマンドでインストールできる。或いは自分でコンパイルしてインストールする。

```
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 Linux kernel のコンパイル

6.4.1 カーネルのコンパイル

1）カーネルのソースコードをダウンロードする。

NanoPi2のカーネルのソースコードは[nanopi2-lollipop-mr1]ブランチにある。

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

2）Androidカーネルをコンパイルする。

```
make nanopi2_android_defconfig
touch .scmversion
make uImage
```

3）Debianカーネルをコンパイルする。

```
make nanopi2_linux_defconfig
touch .scmversion
make uImage
```

コンパイル成功後、新しく生成したファイルはarch/arm/boot/uImage、HDMI出力をサポートする。SDカードのbootセクションにある同じファイル名のファイルと置き換えれば良い。

LCD表示をサポートするイメージファイルを作成するには設定を変更する必要。

```
touch .scmversion
make nanopi2_linux_defconfig
make menuconfig
  Device Drivers -->
    Graphics support -->
      Nexell Graphics -->
        [*] LCD
        [ ] HDMI
make uImage
```

6.4.2 カーネルモジュールのコンパイル

Androidはカーネルモジュールを含んでいる。場所はsystemセクションの/lib/modules/である。新しいカーネルモジュール或いはカーネルモジュールの設定が変更した場合、再コンパイルが必要である。

先ず、カーネルソースのモジュールをコンパイルする。

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

またAndroidのソースに２つのカーネルモジュールのソースがある。下記コマンドでコンパイルする：

```
cd /opt/FriendlyARM/s5p4418/android
./vendor/friendly-arm/build/common/build-modules.sh
```

“/opt/FriendlyARM/s5p4418/android”はAndroidのソースのTOPフォルダである、[-h]パラメータでヘルプ内容を確認できる。

コンパイル成功した後、生成したカーネルモジュールが表示される。

6.5 Andriod システムのコンパイル

6.5.1 コンパイル環境の構築

64ビットのUbuntu 14.04を推奨する。必要なパッケージをインストールすれば良い。

```
sudo apt-get install zlib1g-dev:i386
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
```

詳細内容は下記URLをご参照ください。

https://source.android.com/source/initializing.html

6.5.2 ソースコードをダウンロードする

Androidのソースコードをダウンロードするにはrepoが必要、インストール方法及び使用方法は下記URLをご参照ください。https://source.android.com/source/downloading.html

```
mkdir android && cd android
```

```
repo init -u https://github.com/friendlyarm/android_manifest.git -b nanopi2-lollipop-mr1
repo sync
```

上記の“android”はワークフォルダーの事である。

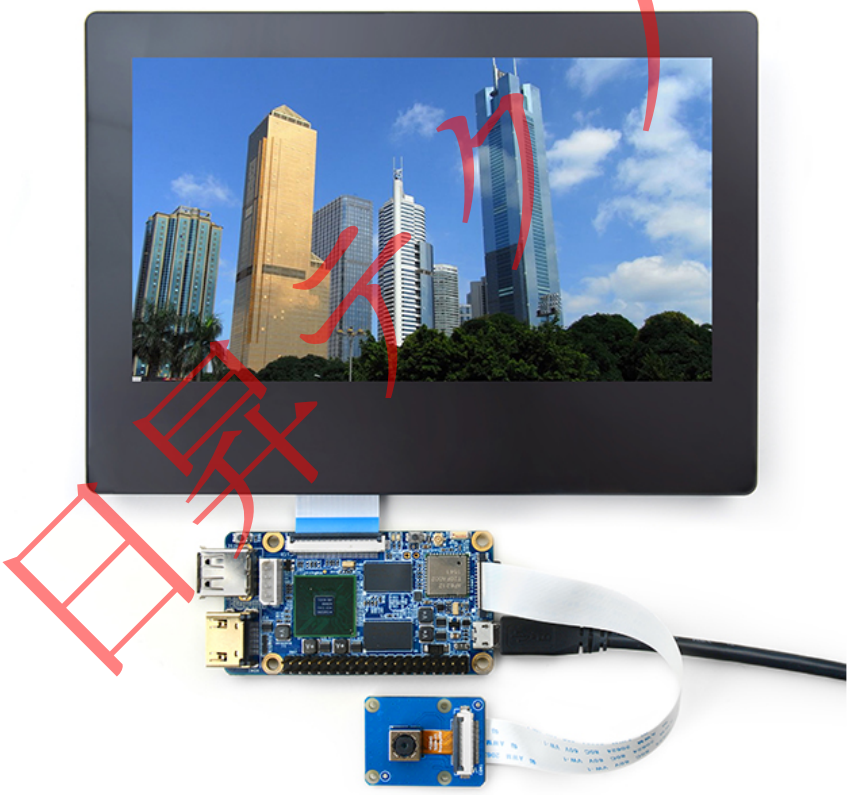
6.5.3 システムをコンパイルする

```
source build/envsetup.sh
lunch aosp_nanopi2-userdebug
make -j8
```

コンパイル終了後、out/target/product/nanopi2/のフォルダにイメージファイルが生成される。

7 カメラモジュールを接続する

500 万画素 ov5640 カメラモジュールを繋ぐ。
Android5.1 システムを実行して “camera”アイコンをクリックする。



8 NanoPi2 の TF カードのセクションを拡張

8.1 Debian 用

PC ホストの端末に以下のコマンドを実行する。

```
sudo umount /dev/sdx
sudo parted /dev/sdx unit % resizepart 2 100 unit MB print
sudo resize2fs -f /dev/sdx2
```

8.2 Android 用

PC ホストの端末に以下のコマンドを実行する。

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