

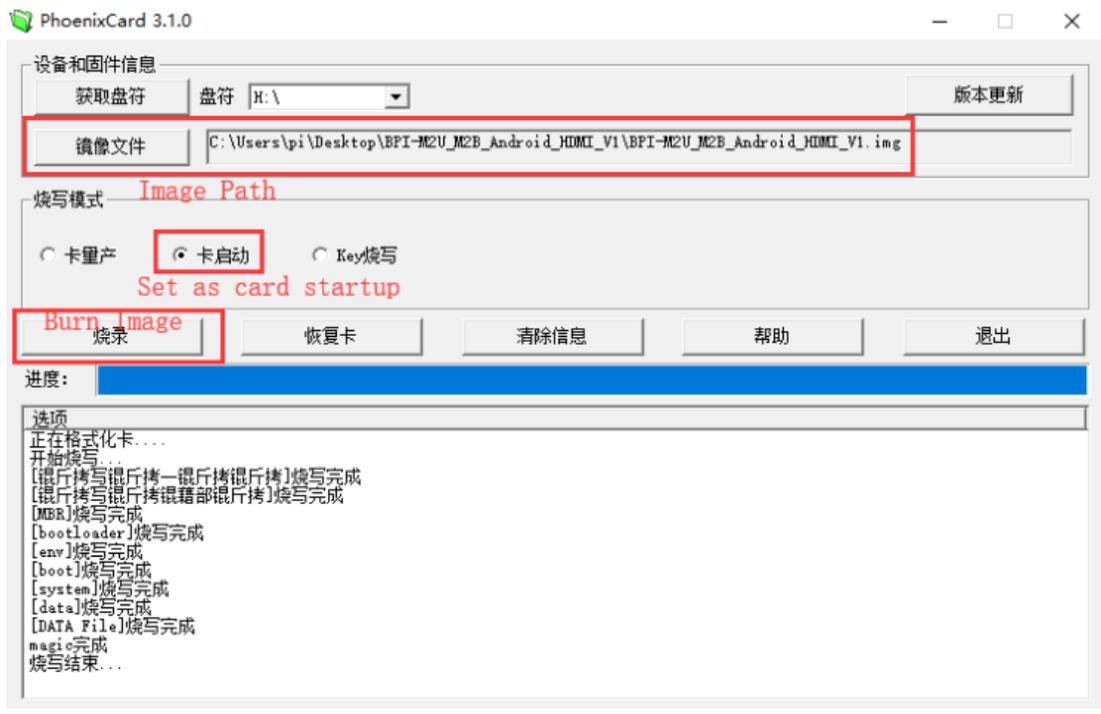
Development For Android

Load your first image on M2P

* Download PhoenixCard: https://pan.baidu.com/s/1-fjvPqtG_zewVzqnXf1AHw?pwd=eid9

- 1.You could download latest image from our forum.
Ex: <http://forum.banana-pi.org/t/bananapi-bpi-m2p-h3-new-image-android7-0-release-2018-6-30/6147>
- 2.Put your TF card into a TF-USB adapter, and then plug adapter in your Windows PC usb interface.
- 3.Prepare your image, and download image burning tools PhoenixCard.exe.
- 4.Use "PhoenixCard.exe" to burn android image to TF card.

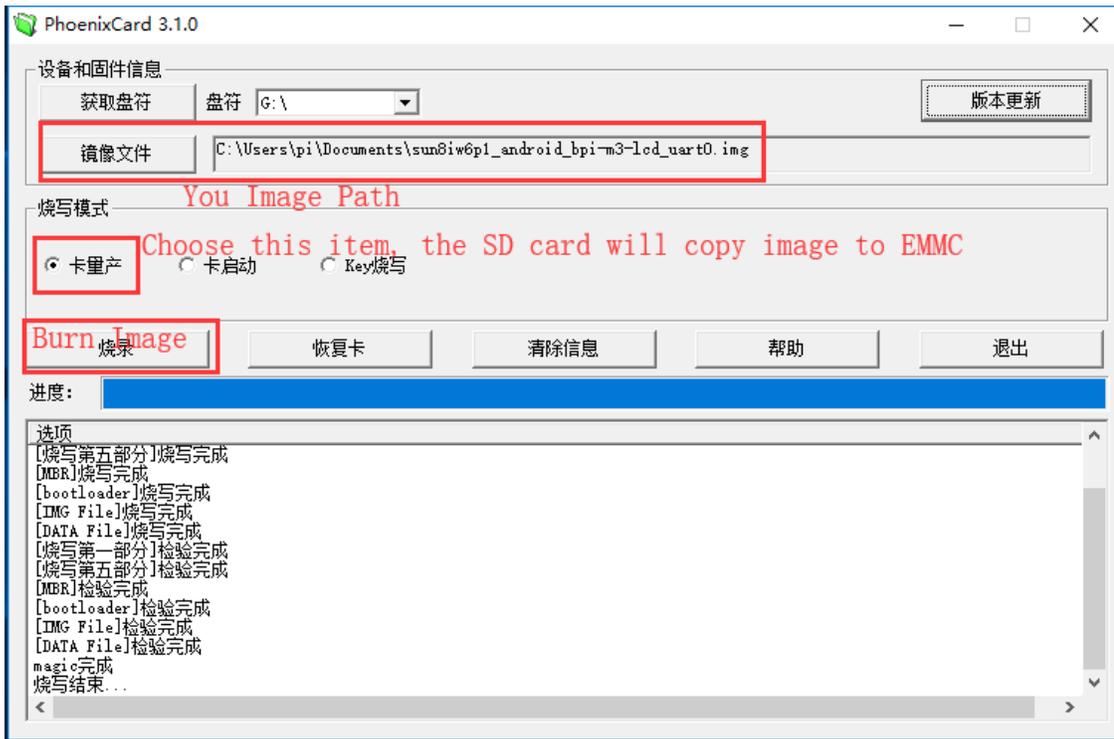
* Here is the example of M3



Load your image on M2P EMMC

- 1.The only different with sd card burning is the image burning mode item choice

* Here is the example of M3



2. After succeed to burn image to SD, then plug SD card in your M2P

3. Press power button, device will copy image to EMMC automatically

* Download PhoenixCard: https://pan.baidu.com/s/1-fjvPqtG_zewVzqnXf1AHw?pwd=eid9

Development For Linux

Basic Development

Prepare to develop

- * Prepare 8G/above TF card, USB-Serial interface, PC with Ubuntu System
- * Using your USB-Serial Connect debug console on M2P

Load your first image on M2P

1. You could download latest image from our forum
* Here is the example forum thread link: <http://forum.banana-pi.org/t/banana-pi-bpi-m2p-new-image-release-ubuntu-16-04-v1-1/5719>
2. Install bpi-tools on your Ubuntu. If you can't access this URL or any other problems, please go to bpi-tools repo and install this tools manually.
* apt-get install pv
* curl -sL https://github.com/BPI-SINOVOIP/bpi-tools/raw/master/bpi-tools | sudo -E bash
3. After you download the image, insert your TF card into your Ubuntu
* Execute "bpi-copy xxx.img /dev/sdx" to install image on your TF card.
4. After step 3, then you can insert your TF card into M2P, and press power button setup M2P.
5. Login user/password: pi/bananapi or root/bananapi.

Load your first image on M2P EMMC

- * Run your M2P with TF card
- * Copy "xxx-sd-emmc-xxx.img.zip / xxx-sd-emmc-xxx.img" to your USB disk
- * Plug your USB disk in M2P
- * After M2P recognise USB disk, execute "bpi-copy xxx-sd-emmc-xxx.img.zip / xxx-sd-emmc-xxx.img" to install image on EMMC
- * Then power off M2P, take TF card out, power on M2P with EMMC

Update your image

For example, update your image to support new emmc5.1

https://wiki.banana-pi.org/Getting_Started_with_M2P

```
* execute "bpi-tools", to update your bpi tools;
* execute "bpi-update -c bpi-m2p.conf", to download new driver to update your image
* execute "file *.tgz", to check download files' type is compressed data
* execute "bpi-bootsetl", you will see the bootloader path, "/usr/lib/u-boot/bananapi/bpi-m2p/BPI_M2P_720P.img.gz"
* execute "bpi-bootsetl /usr/lib/u-boot/bananapi/bpi-m2p/BPI_M2P_720P.img.gz", to update your bootloader
* reboot
```

Advanced Development

How to build uboot & kernel

Install tools

- apt-get update
- apt-get install gcc-arm-linux-gnueabi u-boot-tools
- apt-get install pv
- curl -sL <https://github.com/BPI-SINOVOIP/bpi-tools/raw/master/bpi-tools> | sudo -E bash

Clone code

- git clone: <https://github.com/BPI-SINOVOIP/BPI-M2P-bsp.git>
- ./build.sh

How to create an image

- Prepare a SD card which have installed system(Ubuntu/Raspbian/..)
- Boot your SD card with M2P, after M2P finish starting, copy your files and config your system, then poweroff M2P. [If you don't want to config your system, you can skip this step]
- Plug your SD card in PC(which is running Linux), "cd /media", then "ln -s <your account> pi"
- Execute "bpi-migrate -c bpi-m2p.conf -c ubuntu-mate-from-sd.conf -d /dev/sdx"
- Then you could get your own image now

GPIO Control

- To access a GPIO pin you first need to export it with

```
echo XX > /sys/class/gpio/export
```

- with XX being the number of the desired pin. To obtain the correct number you have to calculate it from the pin name (like PH18)

```
(position of letter in alphabet - 1) * 32 + pin number
for PH18 this would be ( 8 - 1) * 32 + 18 = 224 + 18 = 242 (since 'h' is the 8th letter)
```

- echo "out/in" > /sys/class/gpio/gpio*NUMBER*/direction
- echo "0/1" > /sys/class/gpio/gpio*NUMBER*/value

OTG

1. On M2P console:

- Execute "./adbd.sh", then execute "ps -ax | grep adbd" to see if adbd is set up

2. On PC terminal:

- If adbd was succeed to set up, insert OTG-USB interface to M2P and PC(with Ubuntu system)
- Execute "adb devices" to see if PC has recognised M2P OTG
- If yes, we could execute "adb shell" to connect M2P by adb now

GMAC

- Use iperf3 to test gmac

```

root@bpi-iot-ros-ai:~#
root@bpi-iot-ros-ai:~# iperf3 -c 192.168.30.124
Connecting to host 192.168.30.124, port 5201
[ 4] local 192.168.30.181 port 46018 connected to 192.168.30.124 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr   Cwnd
[ 4] 0.00-1.01      sec    104 MBytes    863 Mb/s         0     208
[ 4] 1.01-2.01      sec   100 MBytes    838 Mb/s         0     208
[ 4] 2.01-3.01      sec   104 MBytes    868 Mb/s         0     208
[ 4] 3.01-4.01      sec   105 MBytes    881 Mb/s         0     208
[ 4] 4.01-5.01      sec   105 MBytes    880 Mb/s         0     208
[ 4] 5.01-6.01      sec   105 MBytes    883 Mb/s         0     208
[ 4] 6.01-7.00      sec   100 MBytes    846 Mb/s         0     208
[ 4] 7.00-8.01      sec   106 MBytes    882 Mb/s         0     208
[ 4] 8.01-9.01      sec   105 MBytes    882 Mb/s         0     329
[ 4] 9.01-10.01     sec   105 MBytes    882 Mb/s         0     329

[ ID] Interval           Transfer     Bandwidth       Retr
[ 4] 0.00-10.01      sec   1.01 GBytes    871 Mb/s         0
[ 4] 0.00-10.01      sec   1.01 GBytes    871 Mb/s         0
                               sender
                               receiver

iperf Done.
root@bpi-iot-ros-ai:~#
root@bpi-iot-ros-ai:~# iperf3 -u -c 192.168.30.124
Connecting to host 192.168.30.124, port 5201
[ 4] local 192.168.30.181 port 44173 connected to 192.168.30.124 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[ 4] 0.00-1.00      sec    120 KBytes    983 kb/s         15
[ 4] 1.00-2.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 2.00-3.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 3.00-4.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 4.00-5.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 5.00-6.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 6.00-7.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 7.00-8.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 8.00-9.00      sec    128 KBytes    1.05 Mb/s        16
[ 4] 9.00-10.00     sec    128 KBytes    1.05 Mb/s        16

[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[ 4] 0.00-10.00      sec    1.24 MBytes    1.04 Mb/s       0.498 ms  0/159 (0%)
[ 4] sent 159 datagrams

iperf Done.

```

Bluetooth

- Use bluetoothctl tool to operate BT
- Execute "bluetoothctl"
- If you don't know how to use bluetoothctl, type "help", you will see more commands
- Execute these commands:

```

[bluetooth]# power on
Changing power on succeeded
[bluetooth]# pairable on
Changing pairable on succeeded
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller B0:F1:EC:FD:4A:7A Discovering: yes
[NEW] Device 68:DF:DD:5D:22:5C 小米手機
[NEW] Device 88:B1:11:75:F3:27 ZY
[NEW] Device 38:59:F9:58:6A:CC ezgo-0
[NEW] Device 9C:B7:0D:02:81:2D DESKTOP-BE1KH00
[NEW] Device 2C:57:31:B6:26:37 Hulkwang
[NEW] Device 40:70:4A:48:6F:43 RG100
[CHG] Device 38:59:F9:58:6A:CC RSSI: -63
[CHG] Device 9C:B7:0D:02:81:2D RSSI: -82
[CHG] Device 88:B1:11:75:F3:27 RSSI: -76
[CHG] Device 9C:B7:0D:02:81:2D RSSI: -65
[CHG] Device 88:B1:11:75:F3:27 RSSI: -94
[CHG] Device 9C:B7:0D:02:81:2D Class: 0x2a010c
[CHG] Device 9C:B7:0D:02:81:2D Icon: computer
[CHG] Device 9C:B7:0D:02:81:2D Connected: yes
[CHG] Device 9C:B7:0D:02:81:2D Connected: no
[DEL] Device 9C:B7:0D:02:81:2D DESKTOP-BE1KH00
[NEW] Device 9C:B7:0D:02:81:2D DESKTOP-BE1KH00
[bluetooth]# pair 9C:B7:0D:02:81:2D
Attempting to pair with 9C:B7:0D:02:81:2D
[CHG] Device 9C:B7:0D:02:81:2D Class: 0x2a010c
[CHG] Device 9C:B7:0D:02:81:2D Icon: computer
[CHG] Device 9C:B7:0D:02:81:2D Connected: yes
Request confirmation
[agent] Confirm passkey 472236 (yes/no): yes

```

添加设备

你的设备已准备就绪!

bananapim2plus
已连接

```

serial-com4 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com4 x
trust <dev>          Trust device
untrust <dev>       Untrust device
block <dev>         Block device
unblock <dev>      Unblock device
remove <dev>        Remove device
connect <dev>       Connect device
disconnect <dev>    Disconnect device
version              Display version
quit                Quit program
[bluetooth]# list
Controller 43:29:B1:55:01:01 bananapim2plus [default]
[CHG] Device 9C:B7:0D:02:81:2D RSSI: -84
[CHG] Device 88:B1:11:75:F3:27 RSSI: -86
[CHG] Device 9C:B7:0D:02:81:2D RSSI: -73
[CHG] Device 9C:B7:0D:02:81:2D RSSI: -65
[CHG] Device 38:59:F9:58:6A:CC RSSI: -90
[CHG] Device 9C:B7:0D:02:81:2D RSSI: -75
[bluetooth]# info 9C:B7:0D:02:81:2D
Device 9C:B7:0D:02:81:2D
  Name: DESKTOP-BE1KH00
  Alias: DESKTOP-BE1KH00
  Class: 0x02010c
  Icon: computer
  Paired: yes
  Trusted: no
  Blocked: no
  Connected: yes
  LegacyPairing: no
  UUID: Service Discovery Serve.. (00001000-0000-1000-8000-00805f9b34fb)
  UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00805f9b34fb)
  UUID: A/V Remote Control      (0000110e-0000-1000-8000-00805f9b34fb)
  UUID: PANU                     (00001115-0000-1000-8000-00805f9b34fb)
  UUID: PnP Information          (00001200-0000-1000-8000-00805f9b34fb)
  UUID: Vendor specific          (c7f94713-891e-496a-a0e7-983a0946126e)
  Modalias: bluetooth:v0006p0001d0A00
[CHG] Controller 43:29:B1:55:01:01 Discoverable: no
[CHG] Device 38:59:F9:58:6A:CC RSSI: -66
[CHG] Device 9C:B7:0D:02:81:2D RSSI: -66
[bluetooth]#

```

WiFi on M2P

Driver code:

- https://github.com/BPI-SINOVOIP/BPI_WiFi_Firmware

WiFi Client

You have two ways to setup WiFi Client

1. Use commands to setup WiFi client

- `ip link set wlan0 up`
- `iw dev wlan0 scan | grep SSID`
- `vim /etc/wpa_supplicant/wpa_supplicant.conf`

```

network={
ssid="ssid"
psk="password"
priority=1
}

```

- `wpa_supplicant -iwlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf`
- `dhclient wlan0`

2. Use UI interface to setup WiFi Client

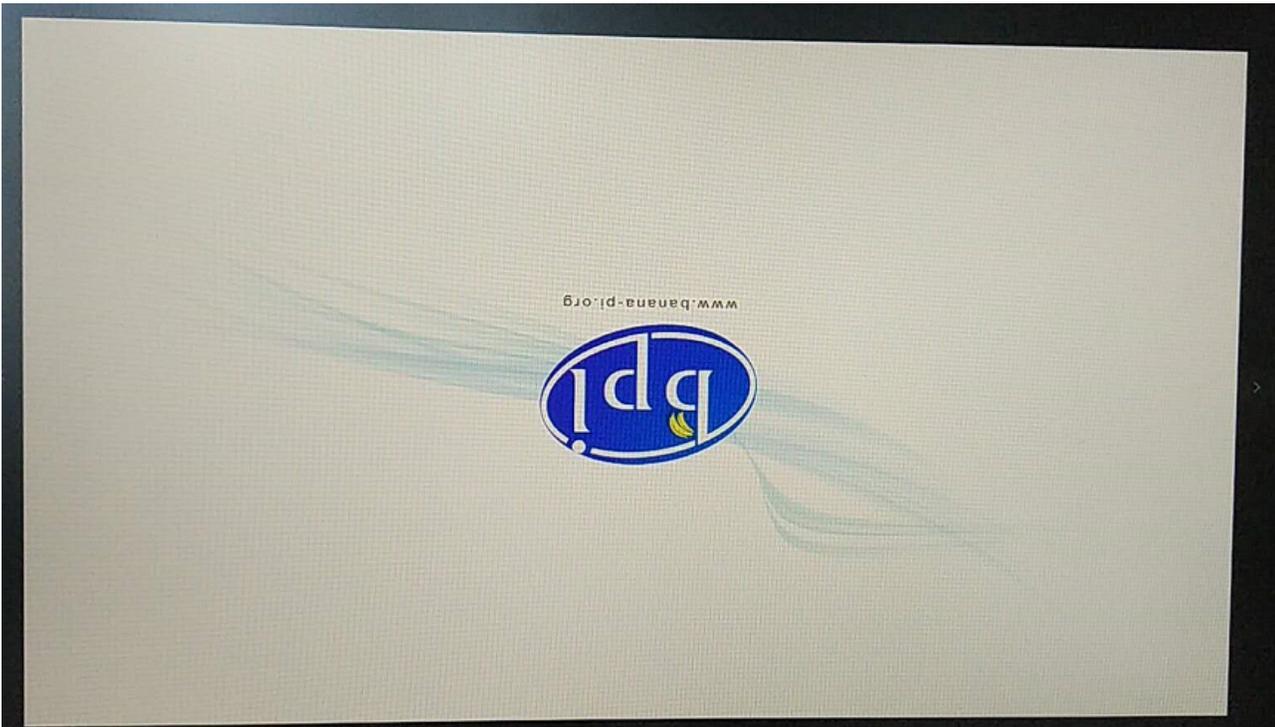
Change Logo

1. Download M2P bsp code

- Execute command `git clone https://github.com/BPI-SINOVOIP/BPI-M2P-bsp`
- After you cloned project, execute command `cd BPI-M2P-bsp`

2. Change to your boot logo

- Prepare a ".bmp" picture, here I rotate 180°, as follows :



- Change your picture name as "bootlogo.bmp"
- put your picture to "sunxi-pack/chips/sun8iw7p1/configs/BPI-M2P-xxxP/"

Here I replaced "bootlogo.bmp" which is under "sunxi-pack/chips/sun8iw7p1/configs/BPI-M2P-720P/" as an example :

```
BPI-M2P-1080P/ BPI-M2P-480P/ BPI-M2P-720P/ default/
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp# ls sunxi-pack/chips/sun8iw7p1/configs/BPI-M2P-720P/
bootlogo.bmp env.cfg script.bin sys_config.fex sys_partition.fex uEnv.txt
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp#
```

3. Build your code

- "./build.sh BPI-M2P-720P"
- choose 1

```
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp#
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp#
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp# ./build.sh BPI-M2P-720P
top dir
NOTICE:
new build.sh default select BPI-M2P-720P and pack all boards
supported boards:
BPI-M2P-1080P
BPI-M2P-480P
BPI-M2P-720P

BPI-M2P-720P configured. Now run `make`
This tool support following building mode(s):
-----
1. Build all, uboot and kernel and pack to download images.
2. Build uboot only.
3. Build kernel only.
4. kernel configure.
5. Pack the builds to target download image, this step must execute after u-boot,
   kernel and rootfs build out
6. update files for SD
7. Clean all build.
-----
Please choose a mode(1-7):
```

- After you built the project, you will see "SD" directory

```

lib/modules/3.4.39-BPI-M2P-kernel/kernel/net/rxrpc/af-rxrpc.ko
lib/modules/3.4.39-BPI-M2P-kernel/kernel/net/rxrpc/rxkad.ko
lib/modules/3.4.39-BPI-M2P-kernel/source
lib/modules/3.4.39-BPI-M2P-kernel/modules.alias.bin
lib/modules/3.4.39-BPI-M2P-kernel/modules.symbols
lib/modules/3.4.39-BPI-M2P-kernel/modules.devname
lib/modules/3.4.39-BPI-M2P-kernel/modules.dep.bin
lib/modules/3.4.39-BPI-M2P-kernel/modules.order
lib/modules/3.4.39-BPI-M2P-kernel/modules.dep
usr/lib/u-boot/bananapi/
usr/lib/u-boot/bananapi/bpi-m2p/
usr/lib/u-boot/bananapi/bpi-m2p/BPI-M2P-1080P.img.gz
usr/lib/u-boot/bananapi/bpi-m2p/BPI-M2P-720P.img.gz
usr/lib/u-boot/bananapi/bpi-m2p/BPI-M2P-480P.img.gz

Build success!

root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp#
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp# ls
add_hash  allwinner-tools  chosen_board.mk  env.sh  Makefile  README.md  scripts  sunxi-pack
add_hash.sh  build.sh  configure  linux-sunxi  output  rootfs  SD  u-boot-sunxi
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp#

```

4. Install a raspbian image on your SD card

5. Plug your SD card into your Ubuntu PC

(1) check your SD card was recognised as /dev/sdxx, as you can see, mine sd card was recognised as /dev/sde

```

Disk /dev/sde 29.7 GiB, 31914983424 bytes, 62333952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xb2dbb41b

Device Boot Start End Sectors Size Id Type
/dev/sde1 204800 729087 524288 256M c W95 FAT32 (LBA)
/dev/sde2 729088 14940159 14211072 6.8G 83 Linux

```

6. Then “cd SD/bpi-m2p/100MB”

7. Execute command “bpi-bootsel BPI-M2P-720P.img.gz /dev/sde”

```

root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp/SD/bpi-m2p/100MB# bpi-bootsel BPI-M2P-720P.img.gz /dev/sde
Warning: Try to write to BOOTDISK /dev/sde
umount device: /dev/sde
umount /dev/sde1
umount /dev/sde2

#####
# Burning the bootloader #
#####

sudo gunzip -c BPI-M2P-720P.img.gz | dd of=/dev/sde bs=1024 seek=8
102392+0 records in
102392+0 records out
104849408 bytes (105 MB, 100 MiB) copied, 10.1666 s, 10.3 MB/s
OK!! You can remove the BOOTDISK /dev/sde now!!
root@JackBpi:~/Code/projects/BPI-H3/BPI-M2P-bsp/SD/bpi-m2p/100MB#

```

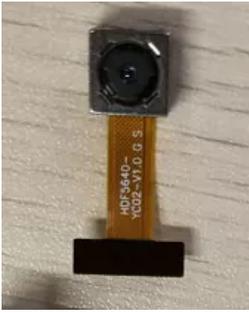
8. Insert your updated SD card to board, and power on, you will see:

Clear boot

- git clone <https://github.com/BPI-SINOVOIP/BPI-files/tree/master/SD/100MB>
- bpi-bootsel BPI-cleanboot-8k.img.gz /dev/sdX

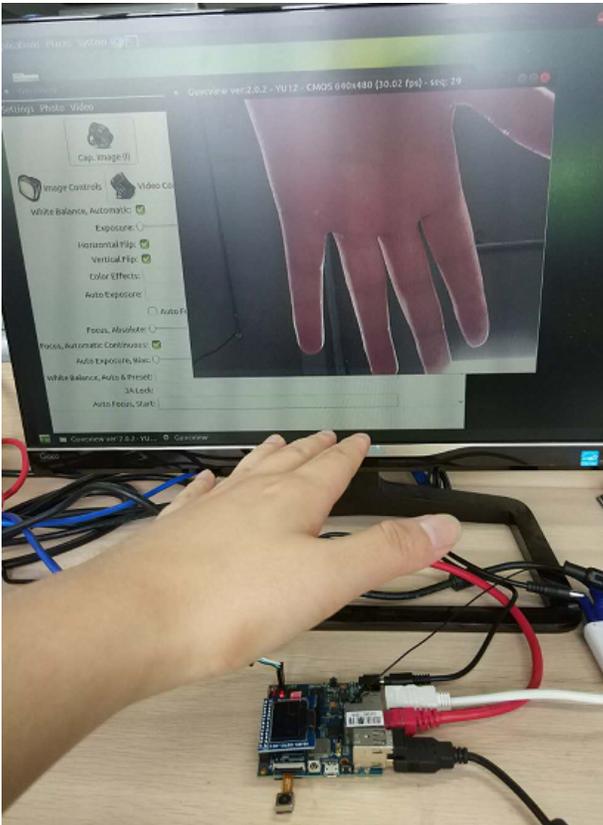
Camara function

We use HDF5640 camara.



Gview

- Use your UI interface to operate camera
- Applications -> Sound & Video -> gview



Shell

- We also have built-in command in /usr/local/bin to test camera
- `./test_ov5640_image_mode.sh` to test picture taking function
- `./cameratest.sh` to test video recording function

IR function

- Execute "getevent"
- Use your IR device to send information to M2P

```

root@bpi-iot-ros-ai:/usr/local/bin# getevent
add device 1: /dev/input/event5
  name: "sunxi-ir"
could not open /dev/input/by-path, Is a directory
could not open /dev/input/by-id, Is a directory
add device 2: /dev/input/event4
  name: "sunxi-ths"
add device 3: /dev/input/event3
  name: "HID 04b4:6005"
could not get driver version for /dev/input/mouse1, Not a tty
add device 4: /dev/input/event2
  name: "HID 04b4:6005"
add device 5: /dev/input/event1
  name: "sunxi-gpiokey"
add device 6: /dev/input/event0
  name: "vmouse"
could not get driver version for /dev/input/mouse0, Not a tty
could not get driver version for /dev/input/mice, Not a tty
poll 7, returned 1
/dev/input/event5: 0001 000a 00000001
poll 7, returned 1
/dev/input/event5: 0000 0000 00000000
poll 7, returned 1
/dev/input/event5: 0001 000a 00000000
poll 7, returned 1
/dev/input/event5: 0000 0000 00000000
poll 7, returned 1
/dev/input/event5: 0001 0047 00000001
poll 7, returned 1
/dev/input/event5: 0000 0000 00000000
poll 7, returned 1
/dev/input/event5: 0001 0047 00000000
poll 7, returned 1
/dev/input/event5: 0000 0000 00000000

```

BPI-Tools

Install Bpi-tools

- Execute "curl -sL https://github.com/BPI-SINOVOIP/bpi-tools/raw/master/bpi-tools | sudo -E bash -"

Update Bpi-tools

- Execute "bpi-tools"

```

root@bpi-iot-ros-ai:~#
root@bpi-iot-ros-ai:~# bpi-tools
bpi-tools(v1.2.1(github)), bananapi system tools.

Usage: bpi-tools [OPTIONS]...
       bpi-tools [ --help | -v | --version ]
       bpi-tools

Options:
  -A, --all           all for tools
  -u, --update        update index files
  -U, --upgrade       donwload & upgrade files
  -G, --download      donwload files
  -h, --help          Print usage
  -v, --version       Print version information and quit

Info:
  default without options will turn on -A for auto install

How to insatll from github:
curl -sL https://github.com/BPI-SINOVOIP/bpi-tools/raw/master/bpi-tools | su

BPIFILE=/root/.bpi-tools.lst
wait for download index file ...

```

RPi.GPIO

Install RPi.GPIO

- Execute "git clone https://github.com/BPI-SINOVOIP/RPi.GPIO"
- after clone the repo, cd RPi.GPIO
- Execute "sudo apt-get update"
- Execute "sudo apt-get install python3-dev python3-dev"
- Execute "sudo python3 setup.py install" or "sudo python3 setup.py install" to install the module

Using RPi.GPIO

- cd /usr/local/bin
- Execute "./bpi_test_g40.py" to test RPi.GPIO

```

root@bpi-iot-ros-ai:/usr/local/bin# ./bpi_test_g40.py
Pi Board Information
-----
PI_REVISION => 3
RAM => 2048MB
REVISION => 4001
TYPE => Banana Pi M3[A83T]
PROCESSOR => Allwinner
MANUFACTURER => BPI-Sinovoip

Is this board info correct (y/n) ? y
8 GPIO.setup GPIO.OUT
./bpi_test_g40.py:21: RuntimeWarning: This channel is already in use, continuing
disable warnings.
  GPIO.setup(pin, GPIO.OUT)
10 GPIO.setup GPIO.OUT
12 GPIO.setup GPIO.OUT
16 GPIO.setup GPIO.OUT
18 GPIO.setup GPIO.OUT
22 GPIO.setup GPIO.OUT
24 GPIO.setup GPIO.OUT
26 GPIO.setup GPIO.OUT
32 GPIO.setup GPIO.OUT
36 GPIO.setup GPIO.OUT

```

WiringPi

- GitHub: <https://github.com/BPI-SINOVOIP/BPI-WiringPi2.git>
- We also have built-in test command in "/usr/local/bin"

How to Update WiringPi

- Execute "bpi-update -c pkglist.conf"

```

root@bpi-iot-ros-ai:/usr/local/bin# bpi-update -c pkglist.conf
CONFIGFILE=pkglist.conf
wait for download pkglist.conf ...
https://github.com/BPI-SINOVOIP/BPI-files/raw/master/others/for-bpi-tools/conf
OK!!\n
APP=/usr/bin/bpi-update
PKGLIST:
bpi-pkg-addons.conf
bpi-pkg-bpi-apps.conf
bpi-pkg-bpi-r2-wifi-firmware-tools.conf
bpi-pkg-bpi-service.conf
bpi-pkg-bpi-test-rfid.conf
bpi-pkg-bpi-tools.conf
bpi-pkg-bpi-w2-tools.conf
bpi-pkg-bpi-wiringpi-arm64.conf
bpi-pkg-bpi-wiringpi.conf
bpi-pkg-brcm.conf
bpi-pkg-bt-arm64.conf
bpi-pkg-bt.conf
bpi-pkg-camera-apps.conf
bpi-pkg-camera.conf
bpi-pkg-libvdpau_sunxi-arm64.conf
bpi-pkg-libvdpau_sunxi.conf
bpi-pkg-ov8865.conf
bpi-pkg-ov8865-enable.conf

```

- Execute "bpi-update -c bpi-pkg-bpi-wiringpi.conf"

```

root@bpi-iot-ros-ai:/usr/local/bin# chmod +x bpi_test_gpio40
root@bpi-iot-ros-ai:/usr/local/bin# ls
a10disp          bt_reset.sh          test_ov5640_image_mode.sh
adbd             cameratest.sh        test_ov5640.sh
adbd.sh          cap                   test_ov8865.sh
apple.dat        ffmpeg-3.1.4         tinacameratest
bpi-bt-on        getevent             tinaplayerdemo
bpi-bt-patch     gpio                 tinarrecorderdemo
bpi-pkg-bpi-wiringpi.conf  gpio40              tinymembench
bpi_test_52pi    guvcview             tusbd.ko
bpi_test_gpio40  guvcview.u1604       usbc1nt
bpi_test_hello   h3disp              usbsrv
bpi_test_lcd1602  irtester            usbsrzd
bpi-wiringpi.tgz  pkglist.conf         usbsrzd-cl
brcm_bt_reset    realtinaplayerdemo  usbsrzd-srv
brcm_patchram_plus  sun8i-corekeeper.sh
root@bpi-iot-ros-ai:/usr/local/bin# chmod +x gpio40
root@bpi-iot-ros-ai:/usr/local/bin# ./bpi_test_gpio40
[RPPI] bhv led test

```

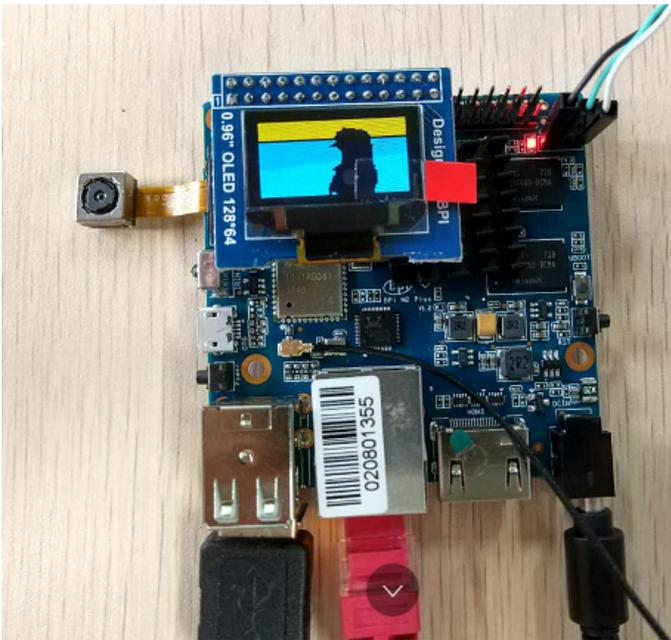
RGB 1602 LCD

- Execute "/usr/local/bin/bpi_test_lcd1602.sh"



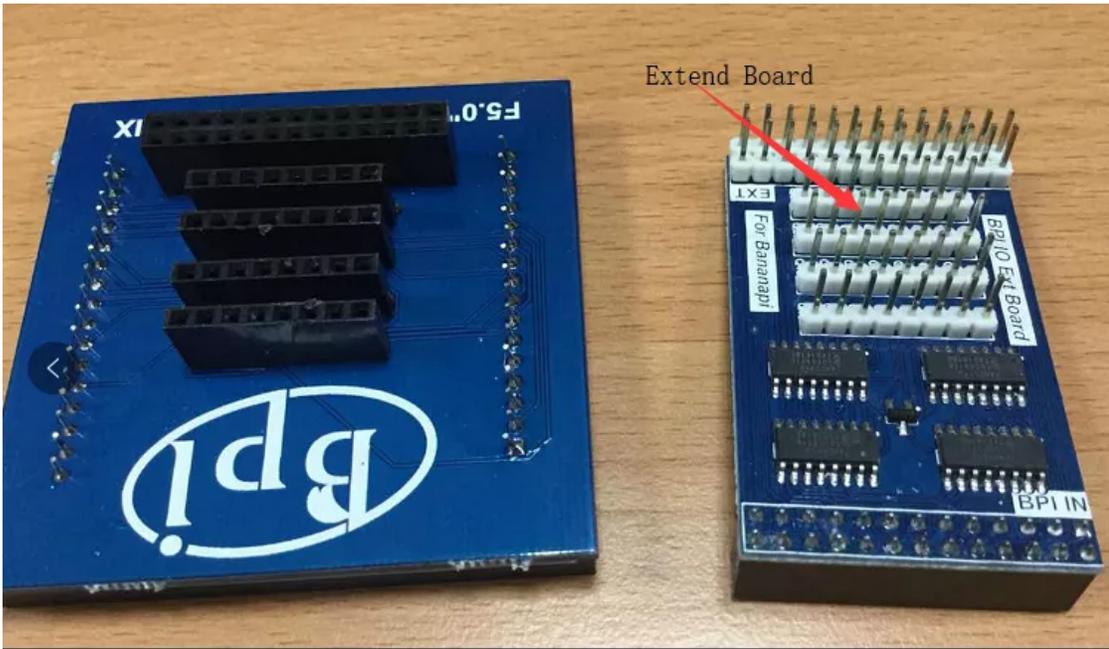
0.96 Inch OLED Display

- Execute `"/usr/local/bin/bpi_test_52pi.sh"`



8x8 RGB LED Martix

- Firstly you need a GPIO Extend Board for 8x8 LED Martix



- Execute `"/usr/local/bin/bpi_test_gpio40.sh"`

