BPI-M2 Zero

Banana Pi M2 Zero is an ultra compact single board computer measures only 60mm*30mm. It uses quad-core Cortex A7 allwinner H2+ processor, with 512MB RAM memory. It's ideal for light-weight systems with some space-limited applications. Like other members of Banana Pi, it supports both linux and android operating system.

Key Features

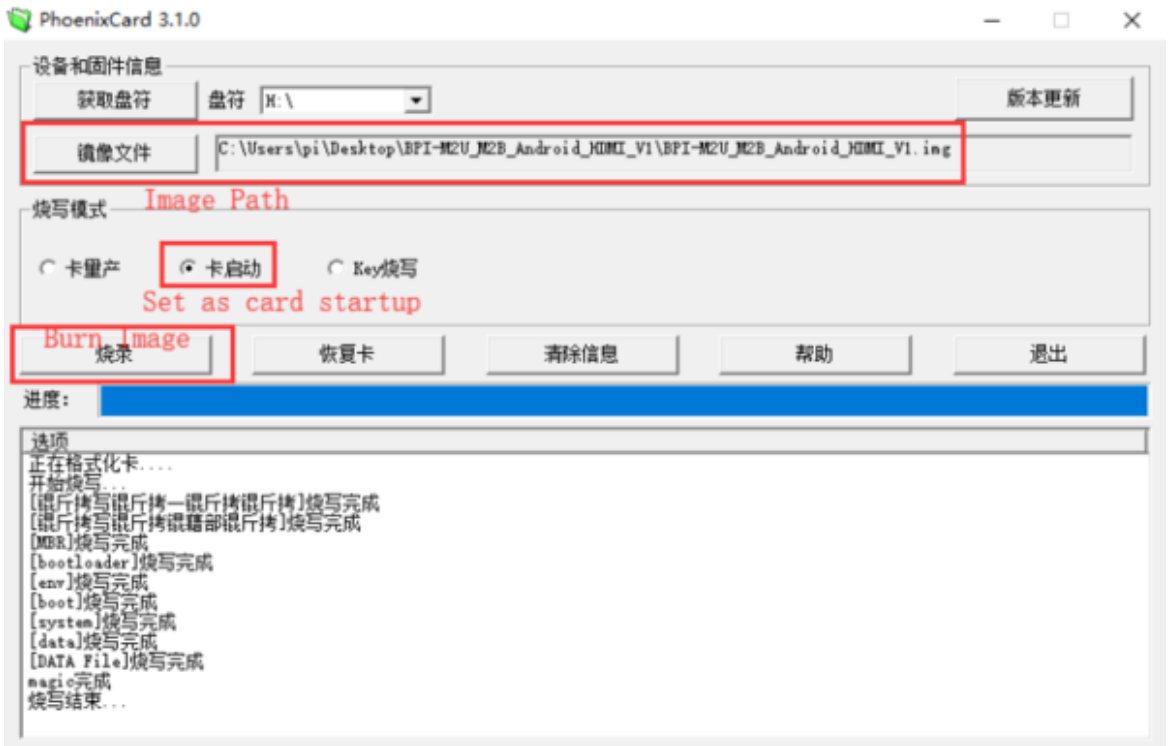
- Quad Core ARM Cortex A7 CPU H2+
- 512MB SDRAM.
- WiFi (AP6212) & Bluetooth onboard.
- Mini HDMI.

Development

Basic Development

Load your first Android image on M2 Zero

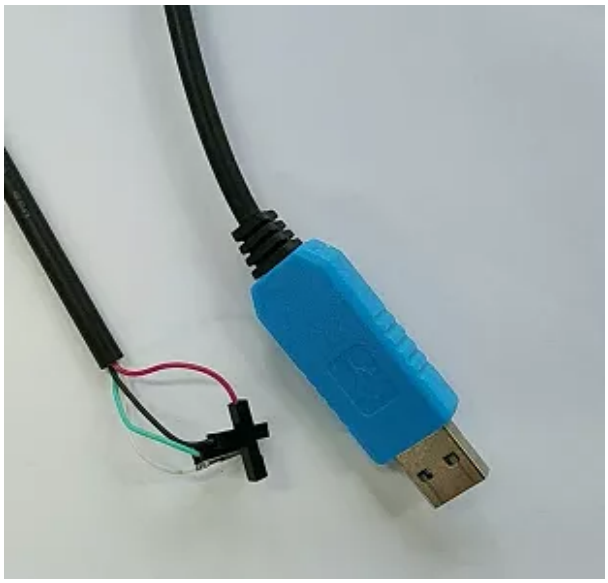
```
1.You could download latest image from our forum.

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.
```

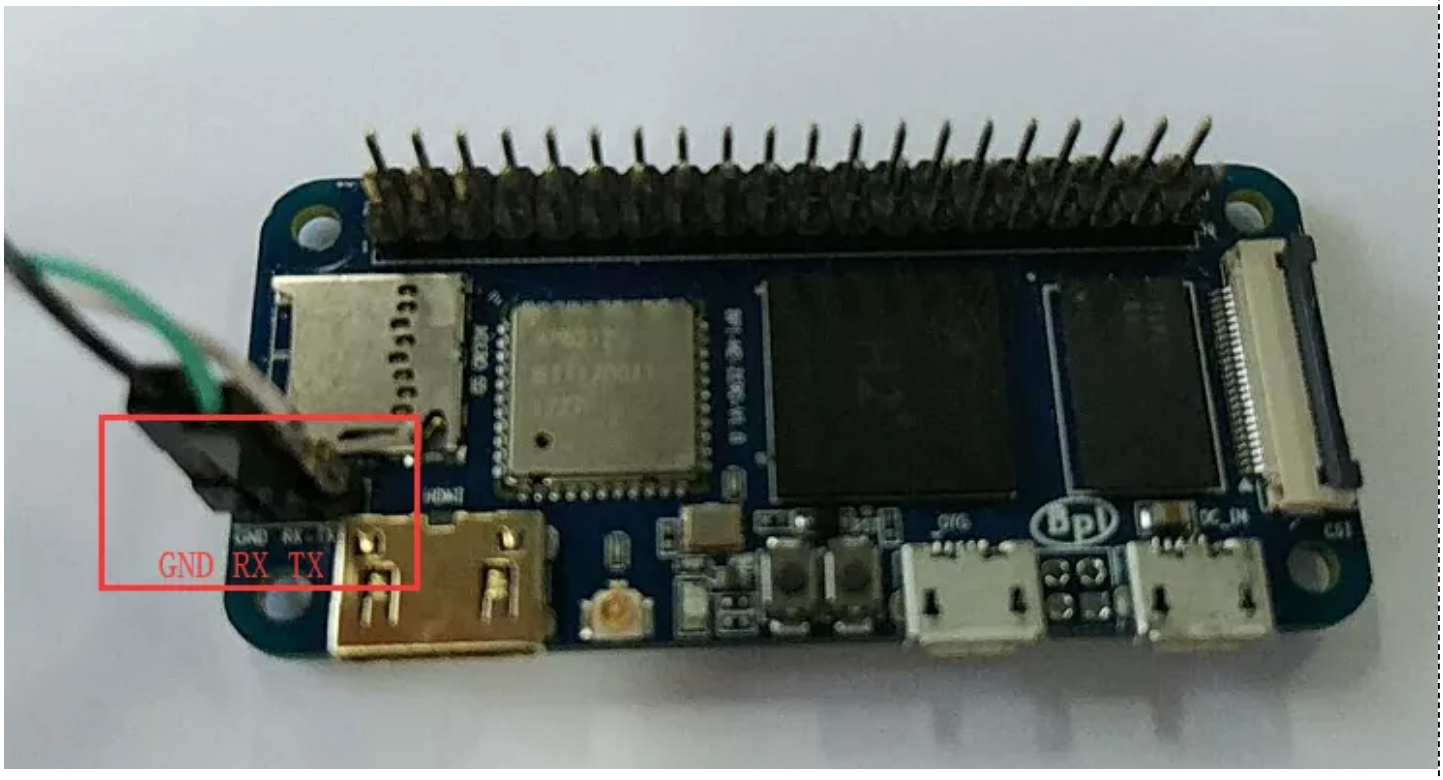


```
* Download PhoenixCard: https://pan.baidu.com/s/1-fjvPqtG_zewVzqnXf1AHw?pwd=eid9
```

Prepare to develop

```
* Prepare 8G/above TF card, USB-Serial interface, PC with Ubuntu System
* Using your USB-Serial Connect debug console on M2 Zero
```





Install Linux Image on M2 Zero

Download latest Linux image, default login user/password is pi/bananapi or root/bananapi.

Install Image to SDcard on Linux PC with bpi-tools

1. Install bpi-tools on your system

```
$ apt-get install pv
$ curl -sL https://github.com/BPI-SINOVOIP/bpi-tools/raw/master/bpi-tools | sudo -E bash
```

If you can't access this URL or any other problems, please go to bpi-tools repo and install this tools manually.

2. Insert your SDcard into your PC

```
$ bpi-copy xxx.img /dev/sdx
```

Install Image to SDcard with Ether on Windows, Linux and MacOS

Balena Etcher is an open source project by Balena, Flash OS images to SD cards & USB drives

Install Image to EMMC with SD Ubuntu

1.Prepare a sd which is installed ubuntu image and bootup with sdcard
2.Copy emmc image to udisk, plug in board, then mount udisk.
3.After mount udisk, use command "bpi-copy xxx-emmc-xxx.img" to install image on Emmc.
4.After success install, power off the board, eject the sdcard and poweron with emmc boot.

Update your image

1. Get the m2 zero bsp source code

```
$ git clone https://github.com/BPI-SINOVOIP/BPI-M2P-bsp-4.4
```

2. Build the source code according to the README.md, and update the packages to the sdcard with bpi image flashed.

Advanced Development

How to create an image

- Prepare a SD card which have installed system(Ubuntu/Raspbian/..)
- Boot your SD card with M2 Zero, after M2 Zero finish starting, copy your files and config your system, then poweroff M2 Zero. [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-m2z.conf -c ubuntu-mate-from-sd.conf -d /dev/sdx"
- Then you could get your own image now

OTG

1. On M2 Zero console:

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

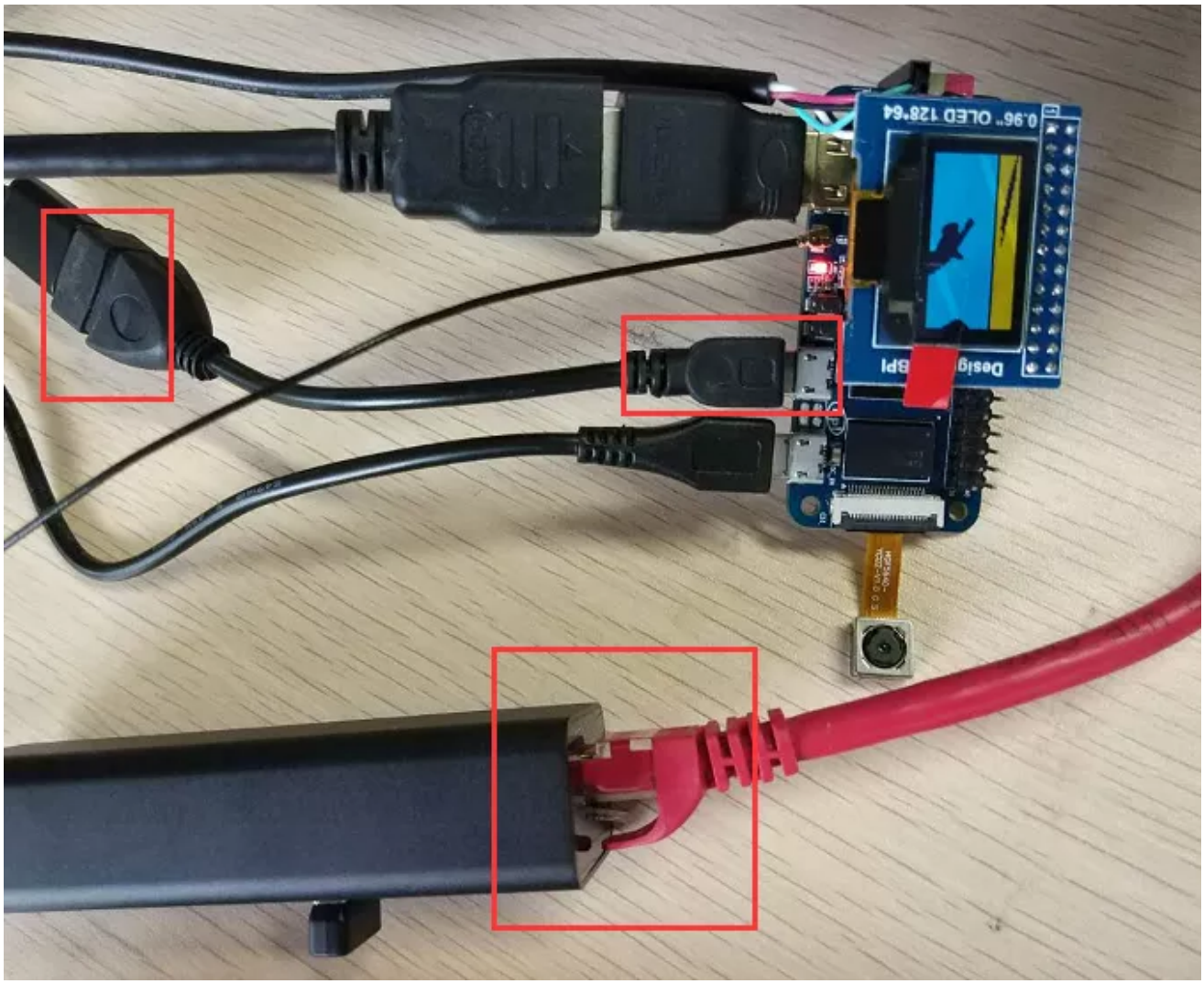
2. On PC terminal:

- If adbd was succeed to set up, insert OTG-USB interface to M2 Zero and PC(with Ubuntu system)
- Execute "adb devices" to see if PC has recognised M2 ZeroP OTG

- If yes, we could execute "adb shell" to connect M2 Zero by adb now

USB Ethernet

- Prepare a USB to OTG wire, usb ethernet adapter



- Use iperf3 to test network

```
root@bpi-iot-ros-ai:~# iperf3 -c 192.168.30.199
Connecting to host 192.168.30.199, port 5201
[  4] local 192.168.30.111 port 52792 connected to 192.168.30.199 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  30.8 MBytes   258 Mbits/sec    0   1.38 MBytes
[  4]   1.00-2.01   sec  29.3 MBytes   244 Mbits/sec    0   1.13 MBytes
[  4]   2.01-3.00   sec  28.2 MBytes   238 Mbits/sec    0   1.24 MBytes
[  4]   3.00-4.01   sec  29.4 MBytes   244 Mbits/sec    0   1.32 MBytes
[  4]   4.01-5.00   sec  28.1 MBytes   238 Mbits/sec    0   1.38 MBytes
[  4]   5.00-6.00   sec  28.8 MBytes   241 Mbits/sec    0   1.42 MBytes
[  4]   6.00-7.02   sec  29.7 MBytes   245 Mbits/sec    0   1.45 MBytes
[  4]   7.02-8.03   sec  29.1 MBytes   242 Mbits/sec    0   1.46 MBytes
[  4]   8.03-9.02   sec  28.6 MBytes   242 Mbits/sec    0   1.08 MBytes
[  4]   9.02-10.01  sec  28.2 MBytes   239 Mbits/sec    0   1.14 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.01  sec   290 MBytes   243 Mbits/sec    0             sender
[  4]   0.00-10.01  sec   289 MBytes   242 Mbits/sec                  receiver

iperf Done.
root@bpi-iot-ros-ai:~# iperf3 -u -c 192.168.30.199
Connecting to host 192.168.30.199, port 5201
[  4] local 192.168.30.111 port 42593 connected to 192.168.30.199 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec   120 KBytes   983 Kbits/sec  15
[  4]   1.00-2.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   2.00-3.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   3.00-4.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   4.00-5.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   5.00-6.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   6.00-7.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   7.00-8.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   8.00-9.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   9.00-10.00  sec   128 KBytes  1.05 Mbits/sec  16
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  4]   0.00-10.00  sec  1.24 MBytes  1.04 Mbits/sec  0.547 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:

```
root@bpi-iot-ros-ai:~# bluetoothctl
[NEW] Controller AA:AA:AA:AA:AA:AA bpi-iot-ros-ai [default]
[NEW] Device 00:1F:20:FF:E3:44 Bluetooth Mouse M557
[NEW] Device 34:88:5D:43:0C:0E Keyboard K380
[NEW] Device 34:88:5D:29:41:92 Bluetooth Mouse M557
[DEL] Device 34:88:5D:29:41:92 Bluetooth Mouse M557
[DEL] Device 34:88:5D:43:0C:0E Keyboard K380
[DEL] Device 00:1F:20:FF:E3:44 Bluetooth Mouse M557
[CHG] Controller AA:AA:AA:AA:AA:AA Powered: no
[CHG] Controller AA:AA:AA:AA:AA:AA Discovering: no
[DEL] Controller AA:AA:AA:AA:AA:AA bpi-iot-ros-ai [default]
[NEW] Controller AA:AA:AA:AA:AA:AA bpi-iot-ros-ai [default]
[NEW] Device 34:88:5D:29:41:92 Bluetooth Mouse M557
[NEW] Device 34:88:5D:43:0C:0E Keyboard K380
[NEW] Device 00:1F:20:FF:E3:44 Bluetooth Mouse M557
[CHG] Controller AA:AA:AA:AA:AA:AA UUIDs:
        00001200-0000-1000-8000-00805f9b34fb
        00001800-0000-1000-8000-00805f9b34fb
        00001801-0000-1000-8000-00805f9b34fb
        0000110e-0000-1000-8000-00805f9b34fb
        0000110c-0000-1000-8000-00805f9b34fb
[CHG] Controller AA:AA:AA:AA:AA:AA Pairable: yes
[CHG] Device 00:1F:20:FF:E3:44 Class: 0x000580
[CHG] Device 00:1F:20:FF:E3:44 Icon: input-mouse
[CHG] Device 00:1F:20:FF:E3:44 Connected: yes
```

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/wpasupplicant/wpa_supplicant.conf

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

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

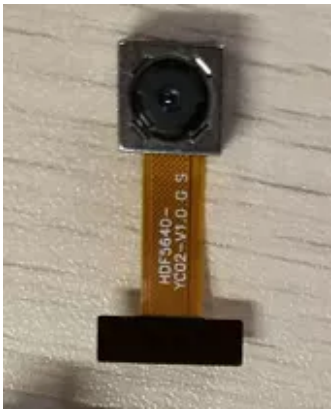
2. Use UI interface to setup WiFi Client

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.



Guvcview

- Use your UI interface to operate camara
- Applications -> Sound & Video -> guvcview

Shell

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

Display

How to change display resolution

For Example： we change M2Z HDMI display 1080P.

1. First, mount /dev/mmcblk0p1 /mnt, then enter to /mnt/bananapi/bpi-m2z/linux, find "sys_config.fex";

2. "vim sys_config.fex", change "screen0_output_mode = 5" to "screen0_output_mode = 10"

```
hdmi_mode_check = 1

[disp_init]
disp_init_enable = 1
disp_mode = 0
screen0_output_type = 3
screen0_output_mode = 10
screen1_output_type = 3
screen1_output_mode = 10
fb0_format = 0
fb0_width = 0
fb0_height = 0
fb1_format = 0
```

3. After save changed, use "fex2bin" command to transfer sys_config.fex to bin file, "fex2bin sys_config.fex script.bin ", reboot.

parameters meaning：

```
#;output_type  (0:none; 1:lcd; 2:tv; 3:hdmi; 4:vga)
#;output_mode  (used for tv/hdmi output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5::
720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50 10:1080p60 11:pal 14:ntsc)
#
# output HDMI 480P (type:3 mode:2)
# output HDMI 720P (type:3 mode:5)
# output HDMI 1080P (type:3 mode:10)
```

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 python-dev python3-dev"
- Execute "sudo python 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
--------------------
P1_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, continu
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
CONFFILE=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                 tinarecorderdemo
bpi-pkg-bpi-wiringpi.conf   gpio40               tinymembench
bpi_test_52pi               guvcview             tusbd.ko
bpi_test_gpio40             guvcview.u1604       usbclnt
bpi_test_hello              h3disp               usbsrv
bpi_test_lcd1602            irtester             usbsrvd
bpi-wiringpi.tgz            pkglist.conf         usbsrvd-cl
brcm_bt_reset               realtinaplayerdemo   usbsrvd-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
[BPI] phy led test
```

RGB 1602 LCD

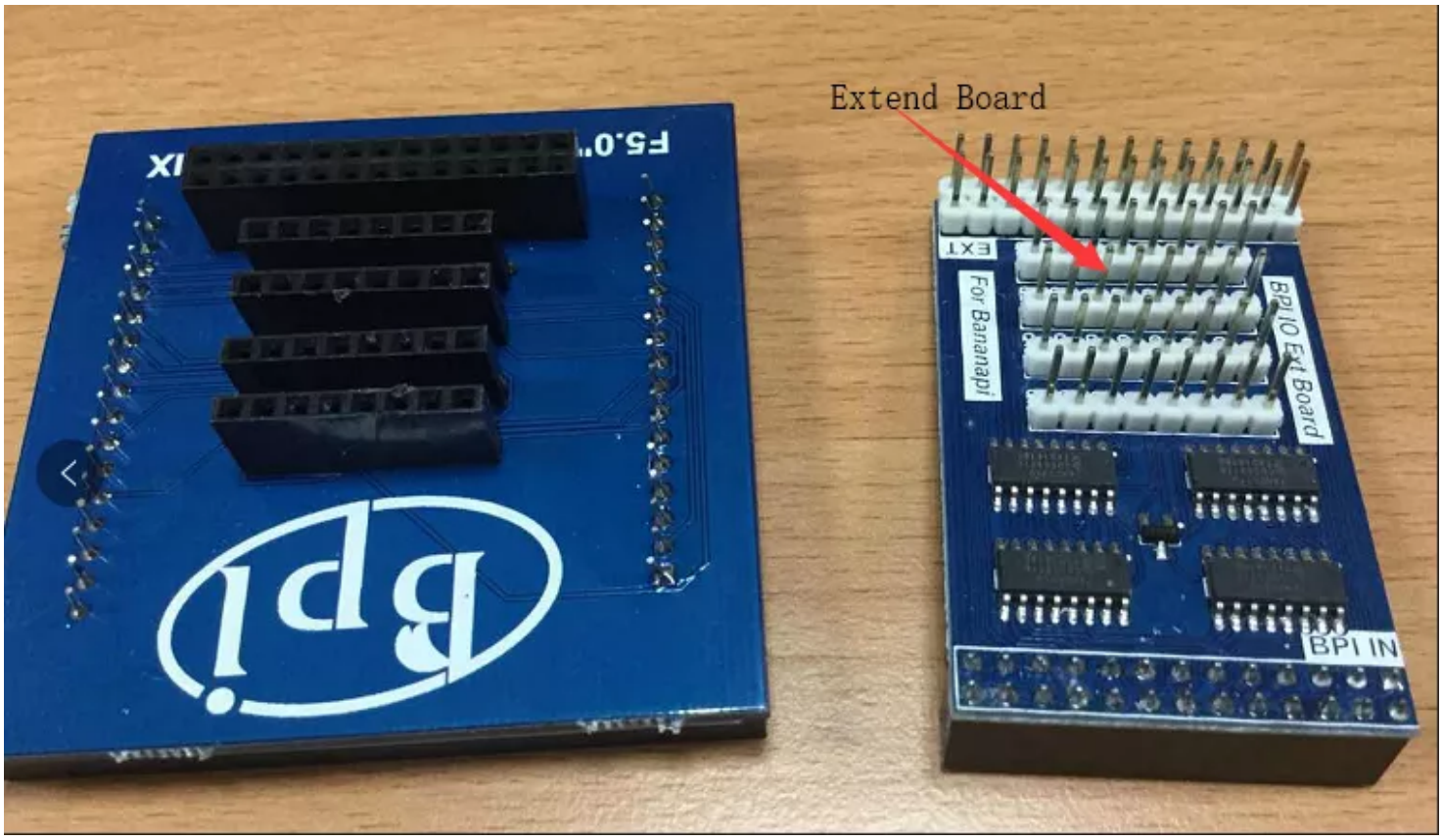
- Execute "/usr/local/bin/bpi_test_lcd1602.sh"

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"