```
Banana Pi M2M is another ARM SoC powered development board that offers great computing performance in an ultra portable form fa
```

- Read more about : Banana Pi BPI-M2M

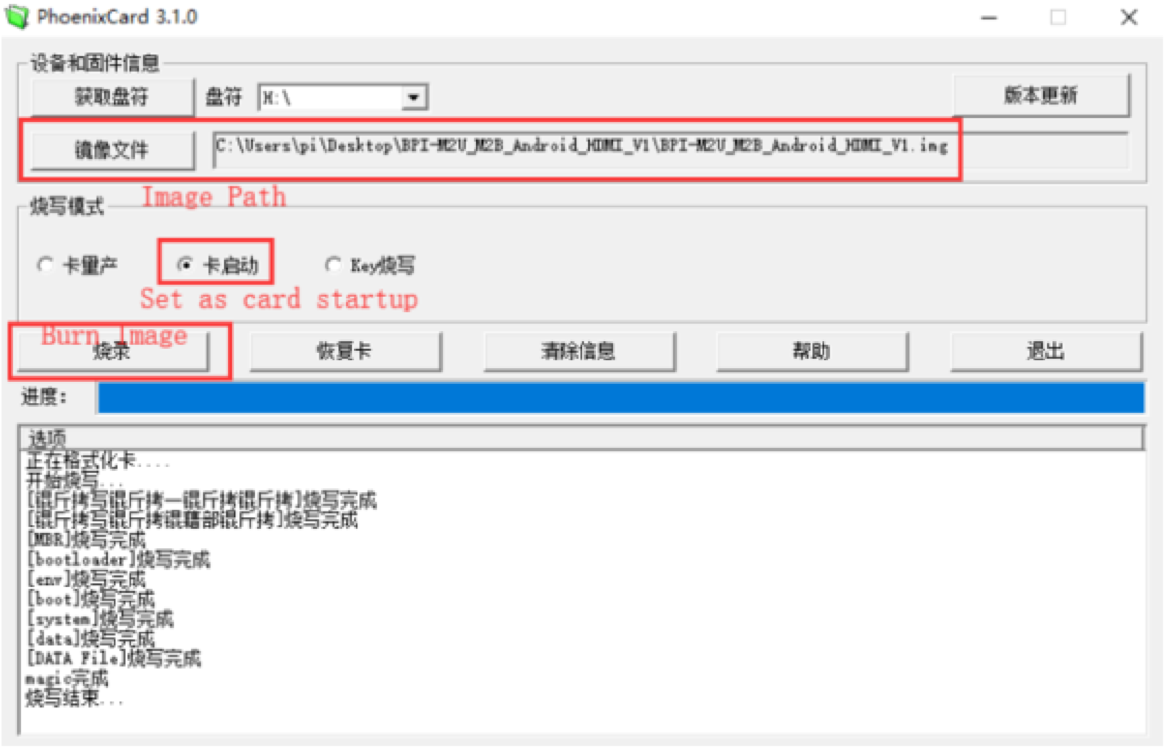
Key Features

- Quad Core ARM Cortex A7 CPU. R16
- 512MB DDR3 SDRAM
- WiFi (AP6212) & Bluetooth onboard

Development

Android

```
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.
```



盘符 M:\

C:\Users\pi\Desktop\BPI-M2U_M2B_Android_HDMI_V1\BPI-M2U_M2B_Android_HDMI_V1.img

Image Path

○ 卡量产 ● 卡启动 ○ Key烧写

Set as card startup

Burn Image

```
正在格式化卡....
开始烧写...
[锟斤拷写锟斤拷一锟斤拷锟斤拷]烧写完成
[锟斤拷写锟斤拷锟睫部锟斤拷]烧写完成
[MBR]烧写完成
[bootloader]烧写完成
[env]烧写完成
[boot]烧写完成
[system]烧写完成
[data]烧写完成
[DATA File]烧写完成
magic完成
烧写结束...
```

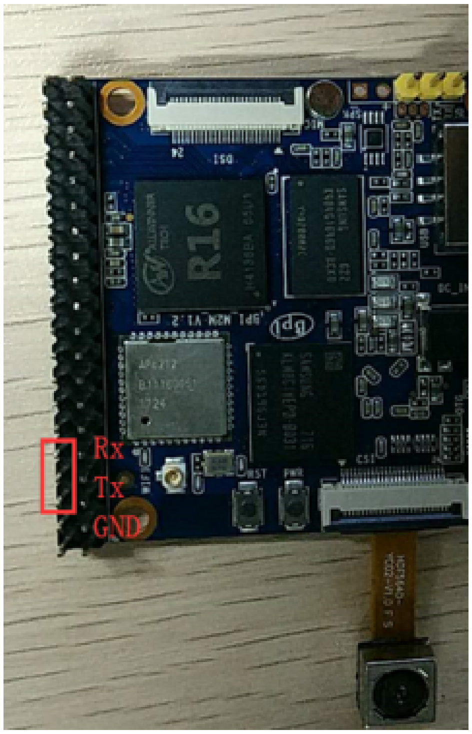
```
* Download PhoenixCard: https://pan.baidu.com/s/1-fjvPqtG_zewVzqnXf1AHw?pwd=eid9
```

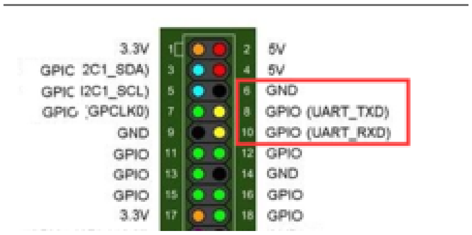
Linux

Prepare

1. Prepare 8G/above TF card, USB-Serial cable.
2. Connect USB-Serial to 40 pin header uart2 for kernel console debug.

Note: three pin header uart0 beside usb port is not debug uart for bpi release images because SDcard and uart0 are multiplex pin and default bpi images support both SD and eMMC. If you want bootup with emmc only and enable uart0 as debug port without SDcard support, please refer to enable uart0 debug for emmc only image section.





3. Install bpi-tools on your Linux PC, If you can't access this URL or any other problems, please go to bpi-tools repo and download this tools manually.

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

4. Download latest linux image, and confirm that the md5 checksum is correct.
5. Defautl login: pi/bananapi or root/bananapi.

Install Image to SDcard

1. Install image with bpi-tools on Linux, plug your sd card to your Linux PC

```
 $ sudo bpi-copy  xxx-bpi-m2m-xxx.img.zip /dev/sdX
```

2. Install image with dd command on Linux, umount SDcard device /dev/sdX partitions if mounted automatically, Actually bpi-copy is the same as this dd command.

```
  $ sudo apt-get install pv
  $ sudo unzip -p xxx-bpi-m2m-xxx.img.zip | pv | dd of=/dev/sdX bs=10M status=noxfer
```

3. Install bpi image with Etcher 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

1. Prepare a SDcard with Linux image flashed and bootup the board with this SDcard.
2. Copy emmc image to udisk, plugin the udisk to board and mount it.
3. Install with bpi-tools command

```
  $ sudo bpi-copy xxx-bpi-m2m-xxx.img.zip /dev/mmcblk1
```

4. Or Install with dd command, umount mmcblk1p1 and mmcblk1p2 partition if mounted automatically, Actually bpi-copy is the same as this dd command

```
  $ sudo apt-get install pv
  $ sudo unzip -p xxxb-bpi-m2m-xxx.img.zip | pv | dd of=/dev/mmcblk1 bs=10M status=noxfer
```

5. After flash complete, power off safely and eject the sdcard.

Note: If the emmc was flashed android image before, you must erase the boot partition of android before step 3 or step 4.

```
  root@bpi-iot-ros-ai:/# echo 0 > /sys/block/mmcblk1/mmcblk1boot0/force_ro
  root@bpi-iot-ros-ai:/# echo 0 > /sys/block/mmcblk1/mmcblk1boot1/force_ro
  root@bpi-iot-ros-ai:/#
  root@bpi-iot-ros-ai:/# dd if=/dev/zero of=/dev/mmcblk1boot0 bs=4096 count=1024
  1024+0 records in
  1024+0 records out
  4194304 bytes (4.2 MB, 4.0 MiB) copied, 0.421164 s, 10.0 MB/s
  root@bpi-iot-ros-ai:/# dd if=/dev/zero of=/dev/mmcblk1boot1 bs=4096 count=1024
  1024+0 records in
  1024+0 records out
  4194304 bytes (4.2 MB, 4.0 MiB) copied, 0.428427 s, 9.8 MB/s
```

Build Source Code

1. Get the bsp source code

```
$ git clone https://github.com/BPI-SINOVOIP/BPI-M2M-bsp
```

2. Build the bsp source code

Please read the source code README.md

Advance Development

Enable uart0 for emmc only image debug

1. Enable uart0 node and disable sdc0 node in sunxi-pack/sun8iw5p1/configs/BPI-M2M-LCD7/sys_config.fex.

```
 --- a/sunxi-pack/sun8iw5p1/configs/BPI-M2M-LCD7/sys_config.fex
 +++ b/sunxi-pack/sun8iw5p1/configs/BPI-M2M-LCD7/sys_config.fex
@@ -262,7 +262,7 @@ twi_sda         = port:PE13<3><default><default><default>
 ;uart_type      = 2:2 wire,4:4 wire,8:8 wire, full function
 ;----------------------------------------------------------------------------------
 [uart0]
-uart_used       = 0
+uart_used       = 1
 uart_port       = 0
 uart_type       = 2
 uart_tx           = port:PF02<3><1><default><default>
@@ -803,7 +803,7 @@ led3_active_low             = 1
 ;     that supports SD3.0 cards and eMMC4.4+ flashes
 ;-------------------------------------------------------------------------------
 [mmc0_para]
-sdc_used          = 1
+sdc_used          = 0
 sdc_detmode       = 3
 sdc_buswidth      = 4
 sdc_d1            = port:PF00<2><1><2><default>
```

2. Build the bsp source code.
3. Copy the bootloader file SD/bpi-m2m/100MB/BPI-M2M-LCD7-8k.img.gz to udisk. bpi also provide a prebuild bootloader for this requirement.
4. Bootup the m2m board from emmc, plugin the udisk and mount it.
5. Flash the bootloader to emmc

```
$ sudo bpi-bootsel BPI-M2M-LCD7-8k.img.gz /dev/mmcblk0
```

or

```
$ sudo gunzip -c BPI-M2M-LCD7-8k.img.gz | dd of=/dev/mmcblk0 bs=1024 seek=8
$ sync
$ sudo umount /dev/sda1
```

6. Mout boot partition and set kernel debug console in uEnv.txt

```
$ sudo mount -t vfat /dev/mmcblk0p1 /mnt
```

Change "console=ttyS2,115200" to "console=ttyS0,115200" in /mnt/bananapi/bpi-m2m/linux/lcd7/uEnv.txt

```
$ sudo umount /dev/mmcblk0p1
```

7. Safely poweroff the board and connect debug uart to uart0 three pin header.

Audio Path

1. 2-pin HP header output audio path

2-pin HP header output audio path		
Number id	Ctl_name	value
1	headphone volume	0-63
104	AIF1IN0L Mux	AIF1_DA0L
103	AIF1IN0R Mux	AIF1_DA0R
100	DACL Mixer AIF1DA0L Switch	1
96	DACR Mixer AIF1DA0R Switch	1
78	HP_R Mux	DACR HPR Switch
77	HP_L Mux	DACL HPL Switch
105	Headphone Switch	1

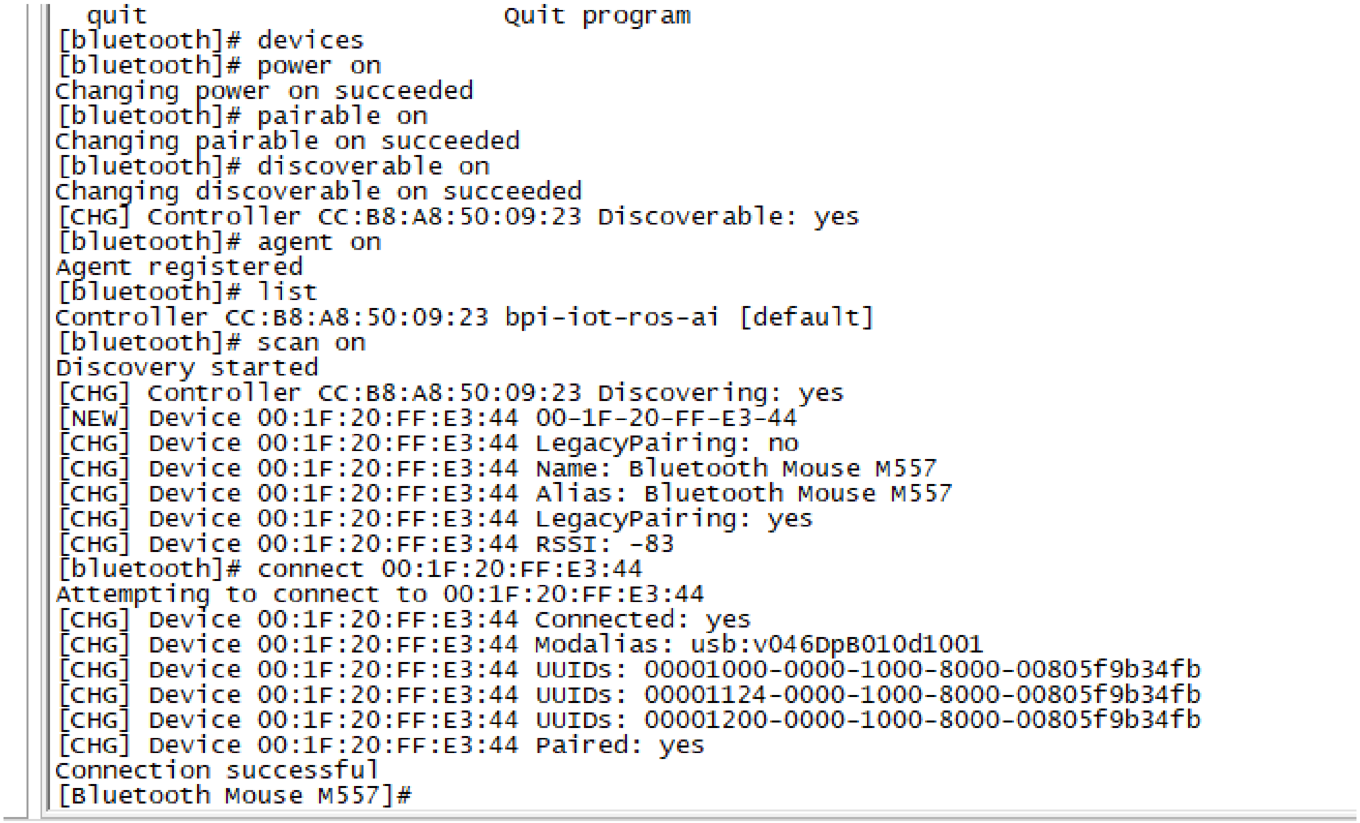
2. Mic 1 input audio path

Mic 1 input audio path		
Number id	Ctl_name	value
56	AIF1OUT0L Mux	AIF1_AD0L
55	AIF1OUT0R Mux	AIF1_AD0R
51	AIF1 AD0L Mixer ADCL Switch	1
47	AIF1 AD0R Mixer ADCR Switch	1
24	ADCR Mux	ADC
25	ADCL Mux	ADC
34	LEFT ADC input Mixer MIC1 boost Switch	1
27	RIGHT ADC input Mixer MIC1 boost Switch	1
4	MIC1 boost amplifier gain	0-7

3.Alsa configuration file /var/lib/alsa/alsa.state.bpi-m2m already enable 2-pin HP output and Mic 1 input audio path, and restore by /var/lib/bananapi/bpi-autorun.d/S10-audio-bpi-m2m.sh after system bootup.

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:



```
  quit                        Quit program
[bluetooth]# devices
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# pairable on
Changing pairable on succeeded
[bluetooth]# discoverable on
Changing discoverable on succeeded
[CHG] Controller CC:B8:A8:50:09:23 Discoverable: yes
[bluetooth]# agent on
Agent registered
[bluetooth]# list
Controller CC:B8:A8:50:09:23 bpi-iot-ros-ai [default]
[bluetooth]# scan on
Discovery started
[CHG] Controller CC:B8:A8:50:09:23 Discovering: yes
[NEW] Device 00:1F:20:FF:E3:44 00-1F-20-FF-E3-44
[CHG] Device 00:1F:20:FF:E3:44 LegacyPairing: no
[CHG] Device 00:1F:20:FF:E3:44 Name: Bluetooth Mouse M557
[CHG] Device 00:1F:20:FF:E3:44 Alias: Bluetooth Mouse M557
[CHG] Device 00:1F:20:FF:E3:44 LegacyPairing: yes
[CHG] Device 00:1F:20:FF:E3:44 RSSI: -83
[bluetooth]# connect 00:1F:20:FF:E3:44
Attempting to connect to 00:1F:20:FF:E3:44
[CHG] Device 00:1F:20:FF:E3:44 Connected: yes
[CHG] Device 00:1F:20:FF:E3:44 Modalias: usb:v046DpB010d1001
[CHG] Device 00:1F:20:FF:E3:44 UUIDs: 00001000-0000-1000-8000-00805f9b34fb
[CHG] Device 00:1F:20:FF:E3:44 UUIDs: 00001124-0000-1000-8000-00805f9b34fb
[CHG] Device 00:1F:20:FF:E3:44 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 00:1F:20:FF:E3:44 Paired: yes
Connection successful
[Bluetooth Mouse M557]#
```

WiFi Client

You have two ways to setup WiFi Client

1. Use commands to setup WiFi client

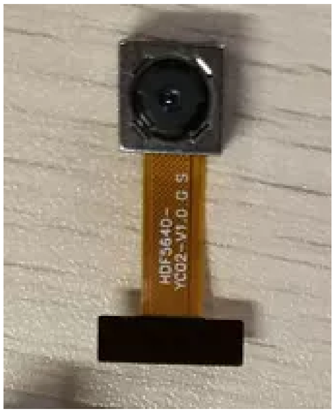
```
$ sudo su
# killall wpa_supplicant
# wpa_passphrase <ssid> <passphrase> > /etc/wpa_supplicant/wpa_supplicant.conf
# ifconfig wlan0 up
# iwlist wlan0 scan
# wpa_supplicant -B -iwlan0 -c/etc/wpa_supplicant/wpa_supplicant.conf
# dhclient wlan0
```

There are some other command line ways, please google for them.

2. Use UI interface to setup WiFi Client

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

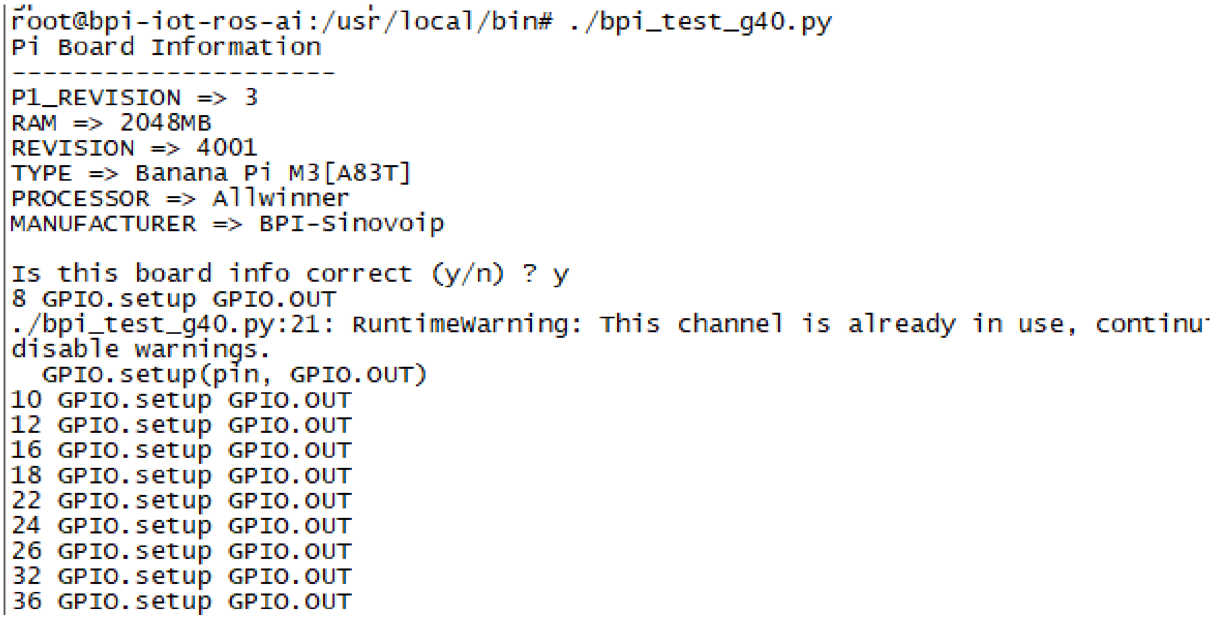
RPi.GPIO

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



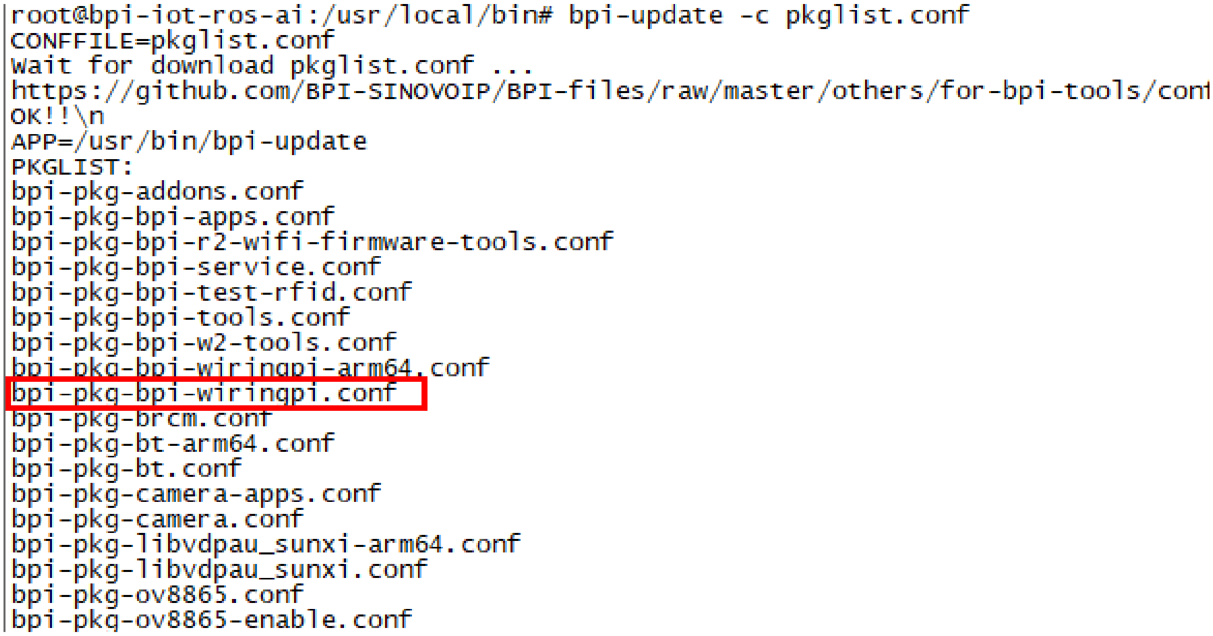
```
root@bpi-iot-ros-ai:/usr/local/bin# ./bpi_test_g40.py
Pi Board Information
---------------------
P1_REVISION => 3
RAM => 2048MB
REVISION => 4001
TYPE => Banana Pi M3[A83T]
PROCESSOR => Allwinner
MANUFACTURER => BPI-Sinovoip

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"

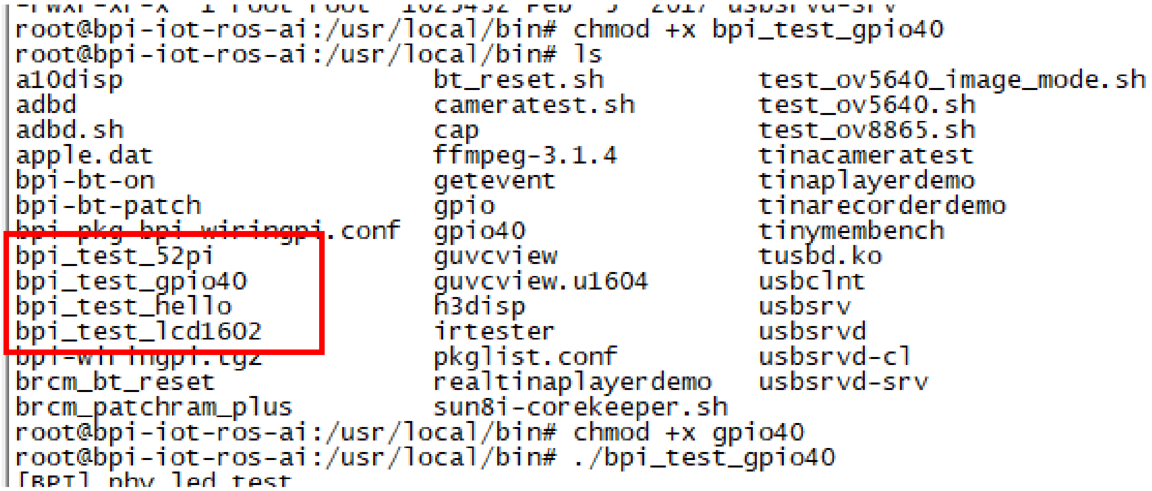
How to Update WiringPi

- 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
```

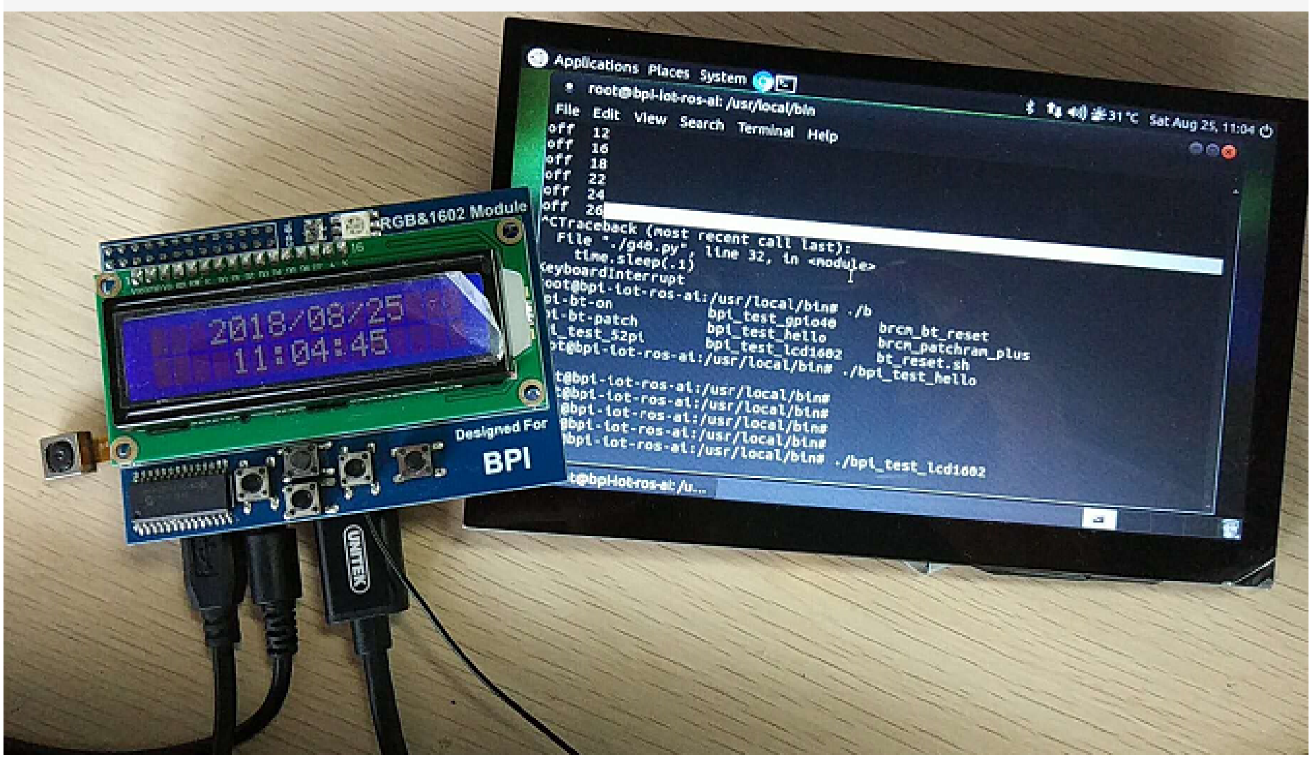
- Execute "bpi-update -c bpi-pkg-bpi-wiringpi.conf"



```
-rwxr-xr-x  1 root root  1025432 Feb  5  2017 usbsrvd-srv
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"

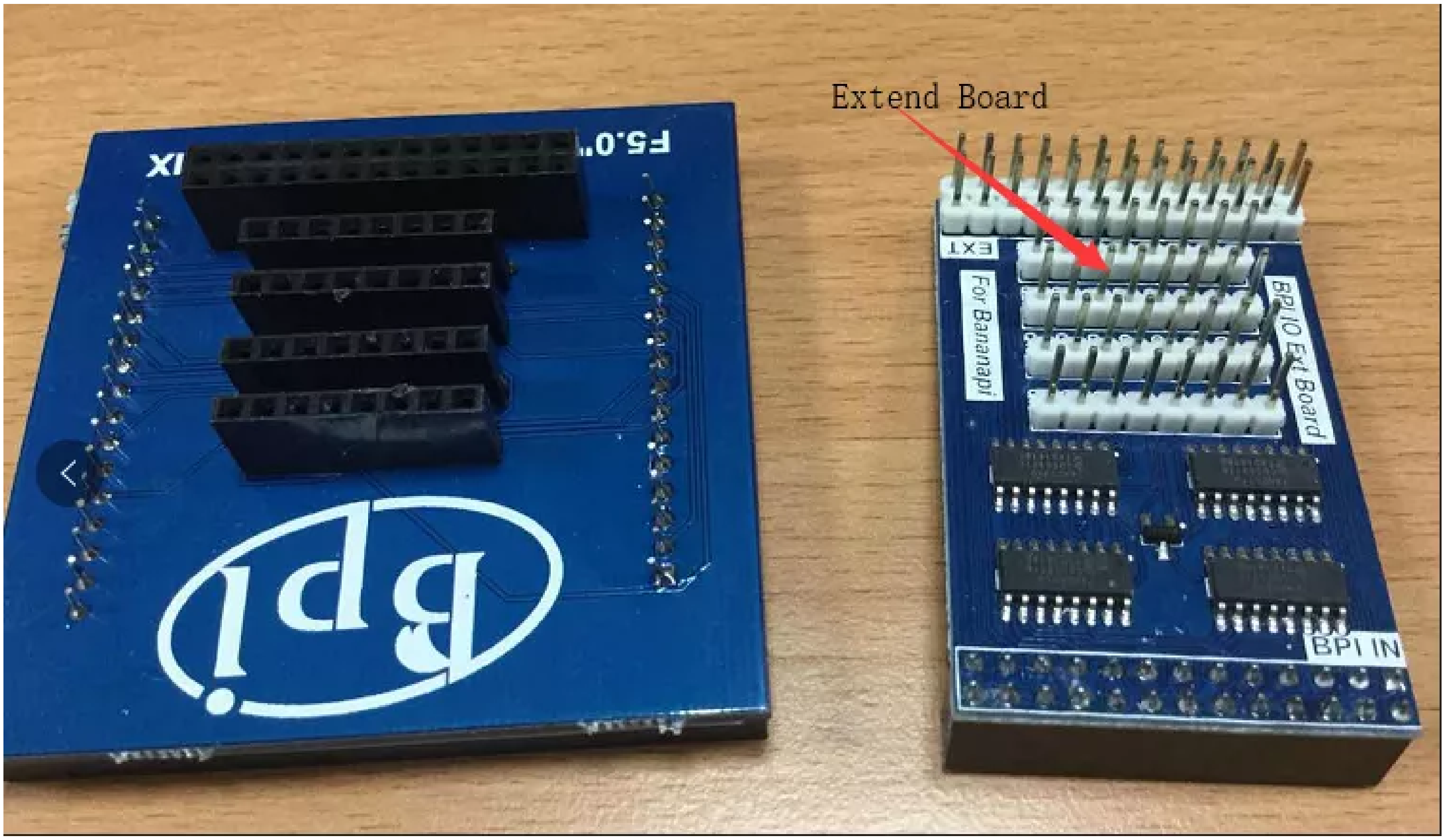

RGB&1602 Module
2018/08/25
11:04:45
Designed For
BPI
Applications Places System
root@bpi-iot-ros-ai: /usr/local/bin
File Edit View Search Terminal Help
UNITEK

0.96 Inch OLED Display

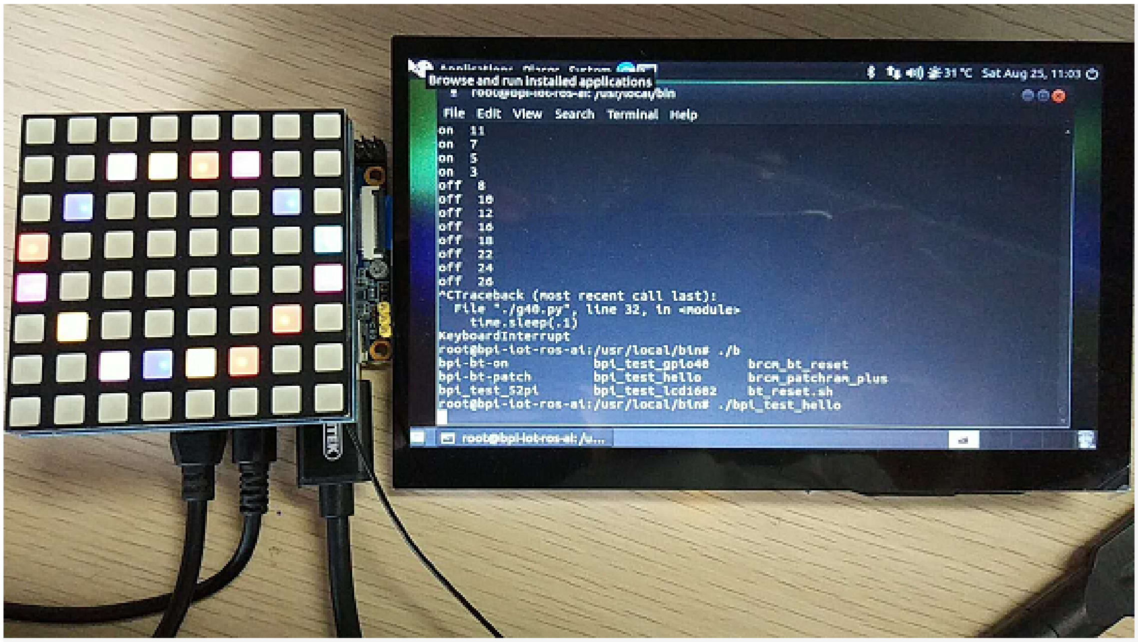
- Execute "/usr/local/bin/bpi_test_52pi.sh"

8x8 RGB LED Martix

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



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


Browse and run installed applications
root@bpi-iot-ros-ai: /usr/local/bin
File Edit View Search Terminal Help
31 °C Sat Aug 25, 11:03
root@bpi-iot-ros-ai: /u...

```
on  11
on  7
on  5
on  3
off  8
off  10
off  12
off  16
off  18
off  22
off  24
off  26
^CTraceback (most recent call last):
  File "./g40.py", line 32, in <module>
    time.sleep(.1)
KeyboardInterrupt
root@bpi-iot-ros-ai:/usr/local/bin# ./b
bpi-bt-on            bpi_test_gpio40     brcm_bt_reset
bpi-bt-patch         bpi_test_hello      brcm_patchram_plus
bpi_test_52pi        bpi_test_lcd1602    bt_reset.sh
root@bpi-iot-ros-ai:/usr/local/bin# ./bpi_test_hello
```