

EM3288 Android7.1 User Manual

V1.0



Boardcon Embedded Design

www.boardcon.com



1. Introduction

1.1. About this Manual

This manual is intended to provide the user with an overview of the board and benefits, complete features specifications, and set up procedures. It contains important safety information as well.

1.2. Feedback and Update to this Manual

To help our customers make the most of our products, we are continually making additional and updated resources available on the Boardcon website (www.boardcon.com , www.armdesigner.com). These include manuals, application notes, programming examples, and updated software and hardware. Check in periodically to see what's new!

When we are prioritizing work on these updated resources, feedback from customers is the number one influence, If you have questions, comments, or concerns about your product or project, please no hesitate to contact us at support@armdesigner.com.

1.3. Limited Warranty

Boardcon warrants this product to be free of defects in material and workmanship for a period of one year from date of buy. During this warranty period Boardcon will repair or replace the defective unit in accordance with the following process:

A copy of the original invoice must be included when returning the defective unit to Boardcon. This limited warranty does not cover damages resulting from lighting or other power surges, misuse, abuse, abnormal conditions of operation, or attempts to alter or modify the function of the product. This warranty is limited to the repair or replacement of the defective unit. In no event shall Boardcon be liable or responsible for any loss or damages, including but not limited to any lost profits, incidental or consequential damages, loss of business, or anticipatory profits arising from the use or inability to use this product.

Repairs make after the expiration of the warranty period are subject to a repair charge and the cost of return shipping. Please contact Boardcon to arrange for any repair service and to obtain repair charge information.



Revision History

Ver	Description	Author	Date
V1.0	Initial version android7.1.2	Zhao Linhai	2019-04-04

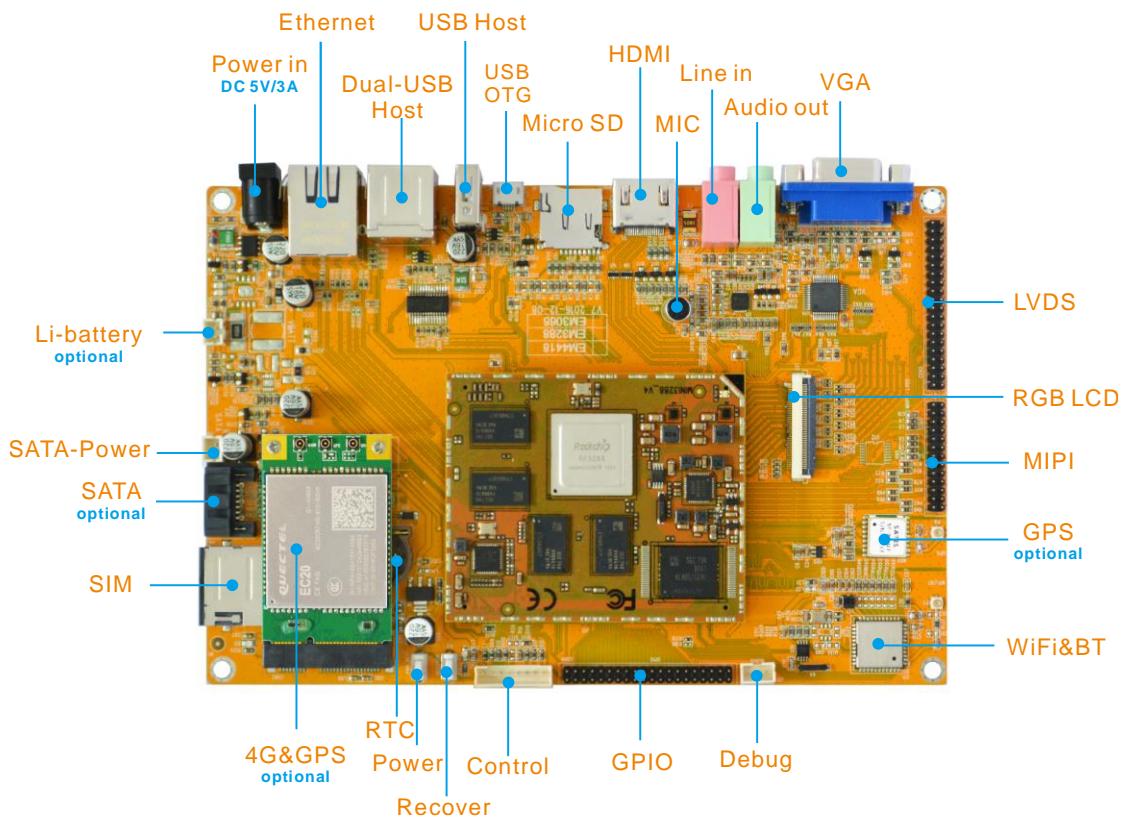


Content

1 EM3288 Introduction	4
2 Compiler Environment.....	6
2.1 Vmware10.0+ubuntu16.04	6
2.2 Install OpenJDK1.8	6
2.3 Install Tools.....	6
3 Compile Source.....	7
4 Images Operation.....	8
4.1 Pack Image	8
4.2 Unzip Firmware	9
5 Install Tools.....	10
5.1 Install CP2102 Driver	10
5.2 Install Rockchip Driver Assistant	10
5.3 Install Serial Terminal Tool.....	12
6 Burn Images	14
7 Android Application.....	17
7.1 HDMI Display	17
7.2 SD Card	17
7.3 USB Host	18
7.4 Video Player.....	18
7.5 Ethernet.....	19
7.6 Record.....	21
7.7 RTC	23
7.8 WiFi	23
7.9 Bluetooth	24
7.10 4G Network	25
7.11 GPS	27
7.12 Camera	28
7.13 Same/Different Display.....	29
7.14 Miracast.....	31



1 EM3288 Introduction



Feature	Specifications
CPU	<ul style="list-style-type: none">Rockchip RK3288, Quad Core Cortex-A17 @ 1.8GHz28nm HKMG process
GPU	<ul style="list-style-type: none">ARM Mali-T764 GPU, with TE, ASTC, AFBC technologySupport OpenGL ES1.1/2.0/3.0, OpenVG1.1, OpenCL, DirectX11
Memory	2GB DDR3
Flash	4G/8GB eMMC Flash
Power	5V/3A
USB	3x USB2.0 Host, 1x USB2.0 OTG
LCD	<ul style="list-style-type: none">1x 40-pin LVDS for 10.1-inch 1280 x 800 LCD with multi-dot capacitive touchscreen;1x 40-pin TTL LCD connector
VGA	1x VGA connector. Automatically adjust according to display size
Ethernet	100/1000M, RJ45 interface. RTL8211E-VB-CG controller
Serial port	1x 3pin connector, for debug
HDMI	HDMI V2.0, up to 4Kx2K@60fps. Audio sync-output
Audio	3.5mm jacks, MIC. ES8388 audio codec



SD card	1x Micro SD card slot
WiFi & Bluetooth	AP6236 module. WiFi - 2.4GHz, 802.11b/g/n. Bluetooth4.0.
4G	Quectel EC20, PCIe connector
GPS	SATES ST-91-U7
Camera	Supporting MIPI camera, and most of USB CMOS camera on the market,
RTC	Real Time Clock, powered by external lithium battery
Button	Power, Recover
GPIO	1x 8-pin Control, 1x 40-pin GPIO
Other interfaces	1x SATA, 1x SATA-Power, 1x SIM Card, 1x Lithium battery interface
Dimension	117.5 x 175.3mm



2 Compiler Environment

2.1 Vmware10.0+ubuntu16.04

Install Vmware10.0 in windows OS, and then install ubuntu16.04 in VMWare to compile. Please visit the official website <http://www.ubuntu.com/> to download and install ubuntu operating system.

Note: Android7.1 should be complied by ubuntu 64bit OS.

2.2 Install OpenJDK1.8

```
# sudo mkdir /usr/lib/java
# sudo tar zxvf java-8-openjdk-amd64.tar.gz -C /usr/lib/java/
```

Add the following information in the end of “/etc/profile”

```
export JAVA_HOME=/usr/lib/java/java-8-openjdk-amd64
export JRE_HOME=/usr/lib/java/java-8-openjdk-amd64/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/jre/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/jre/bin:$PATH
```

```
# source /etc/profile
```

Check if the jdk has been installed successfully and check the revised version:

```
# java -version
```

2.3 Install Tools

PC OS: ubuntu system

Network: online

Permission: root

```
# sudo apt-get install build-essential
# sudo apt-get install zlib1g-dev
# sudo apt-get install flex
# sudo apt-get install libx11-dev
# sudo apt-get install gperf
# sudo apt-get install libncurses5-dev
# sudo apt-get install bison
# sudo apt-get install lsb-core
# sudo apt-get install lib32z1-dev
# sudo apt-get install g++-multilib
# sudo apt-get install lib32ncurses5-dev
# sudo apt-get install uboot-mkimage
# sudo apt-get install g++-4.4-multilib
```



3 Compile Source

Step 1, unzip the source.

```
# tar zxvf em3288_mid_android-7.1.tar.gz
```

Step 2, compile uboot

```
# make rk3288_secure_defconfig  
# ./mkv7.sh
```

Step 3, compile the kernel

```
# cd em3288_mid_android-7.1/kernel  
# make ARCH=arm rockchip_defconfig  
# make ARCH=arm rk3288-evb-android-act8846-lvds.img
```

kernel.img and **resource.img** are generated in current directory.

Step 4, compile the android

```
# cd ..  
# source build/envsetup.sh  
# lunch  
Choose rk3288-userdebug  
# make -j8
```

Step 5, Generated image file

```
# ./mkimage.sh  
# cd rockdev/Image-rk3288  
# ls
```

Images are generated in current directory.



4 Images Operation

4.1 Pack Image

Step 1, copy all the files in Android directory **rockdev/Image** to the windows **AndroidTool_Release_v2.43/rockdev/Image**

Step 2, enter **AndroidTool_Release_v2.43/rockdev/**, double-click to run **mkupdate.bat**.

Step 3, the **update.img** will be generated in **rockdev** directory.

The screenshot displays two windows of the "Android Firmware Package Tool v1.5".

The top window shows the command being run:

```
F:\rk3288\android7.1\U2\rockdev>Afptool -pack ./Image\update.img
----- PACKAGE -----
ParsePackage:Can't get file size(path:.\\backupimage\\package-file)----- FAILED
-----
```

The bottom window shows the output of the command:

```
----- PACKAGE -----
Add file: .\package-file
Add file: .\Image\MiniLoaderAll.bin
Add file: .\Image\parameter.txt
Add file: .\Image\trust.img
Add file: .\Image\uboot.img
Add file: .\Image\misc.img
Add file: .\Image\resource.img
Add file: .\Image\kernel.img
Add file: .\Image\boot.img
Add file: .\Image\recovery.img
Add file: .\Image\system.img
Add file: .\Image\vendor0.img
Add file: .\Image\vendor1.img
Add file: .\update-script
Add file: .\recover-script
Add CRC...
----- OK -----
F:\rk3288\android7.1\U2\rockdev>RKImageMaker.exe -RK32 MiniLoaderAll.bin Image\update.img update.img -os_type:androidos
*****RKImageMaker ver 1.6*****
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!

F:\rk3288\android7.1\U2\rockdev>rem update.img is new format, Image\update.img is old format, so delete older format
F:\rk3288\android7.1\U2\rockdev>del Image\update.img
F:\rk3288\android7.1\U2\rockdev>pause
请按任意键继续. . .
```

A red box highlights the output of the RKImageMaker command, specifically the message "Generating new image, please wait..." and the success messages at the end.



4.2 Unzip Firmware

Step 1, copy **update.img** to the android source directory

RKTools/linux/Linux_Pack_Firmware/rockdev/

Step 2, execute the following command

```
# cd RKTools/linux/Linux_Pack_Firmware/rockdev/  
# chmod 777 unpack.sh  
# ./unpack.sh  
# ls output/  
# ls output/Image/
```

The terminal window shows the execution of the `unpack.sh` script. It starts by listing files in the current directory, then runs `chmod 777 unpack.sh` and executes `./unpack.sh`. The script outputs progress messages: "start to unpack update.img...", "RKImageMaker ver 1.63", "Unpacking image, please wait...", "Exporting boot.bin", "Exporting firmware.img", "Unpacking image success.", "Android Firmware Package Tool v1.62", "Check file... OK", "----- UNPACK -----", and a detailed list of file offsets and addresses. It concludes with "Unpack firmware OK!", "----- OK -----", "Unpacking update.img OK.", and a prompt to press any key to quit.

```
192.168.0.141 x serial-com4  
/Linux_Pack_Firmware/rockdev$ ls  
afptool mkupdate.sh readme.txt unpack.sh  
Image package-file rkImageMaker update.img  
zhaolinhai@boardcon:~/zh1/0_rk3288/6_rk3288/3288/em3288_mid_android-7.1/RKTools/linux  
/Linux_Pack_Firmware/rockdev$ chmod 777 unpack.sh  
zhaolinhai@boardcon:~/zh1/0_rk3288/6_rk3288/3288/em3288_mid_android-7.1/RKTools/linux  
/Linux_Pack_Firmware/rockdev$ ./unpack.sh  
start to unpack update.img...  
*****RKImageMaker ver 1.63*****  
Unpacking image, please wait...  
Exporting boot.bin  
Exporting firmware.img  
Unpacking image success.  
Android Firmware Package Tool v1.62  
Check file... OK  
----- UNPACK -----  
package-file 0x00000000000000800 0x0000000000000002A2  
Image/MiniLoaderAll.bin 0x000000000000001000 0x000000000000002494E  
Image/parameter.txt 0x00000000000026000 0x0000000000000000304  
Image/trust.img 0x00000000000026800 0x00000000000400000  
Image/uboot.img 0x00000000000426800 0x00000000000400000  
Image/misc.img 0x00000000000826800 0x0000000000000000C000  
Image/resource.img 0x00000000000832800 0x00000000000000003D800  
Image/kernel.img 0x00000000000870000 0x00000000000000007C3524  
Image/boot.img 0x00000000001033800 0x000000000000000016E63C  
Image/recovery.img 0x000000000011A2000 0x000000000000005F0ABC  
Image/system.img 0x00000000001793000 0x000000000023D33F10  
Image/vendor0.img 0x0000000000254C7000 0x00000000000023D3394  
Image/vendor1.img 0x00000000002789A800 0x00000000000023D3394  
Unpack firmware OK!  
-----  
----- OK -----  
Unpacking update.img OK.  
Press any key to quit:  
zhaolinhai@boardcon:~/zh1/0_rk3288/6_rk3288/3288/em3288_mid_android-7.1/RKTools/linux  
/Linux_Pack_Firmware/rockdev$
```

The unzip files will be generated in **output** directory.

The terminal window shows the contents of the `output` directory. It lists several files: `MiniLoaderAll.bin`, `parameter.txt`, `boot.img`, `misc.img`, `resource.img`, `trust.img`, `vendor0.img`, `kernel.img`, `recovery.img`, `system.img`, `uboot.img`, and `vendor1.img`.

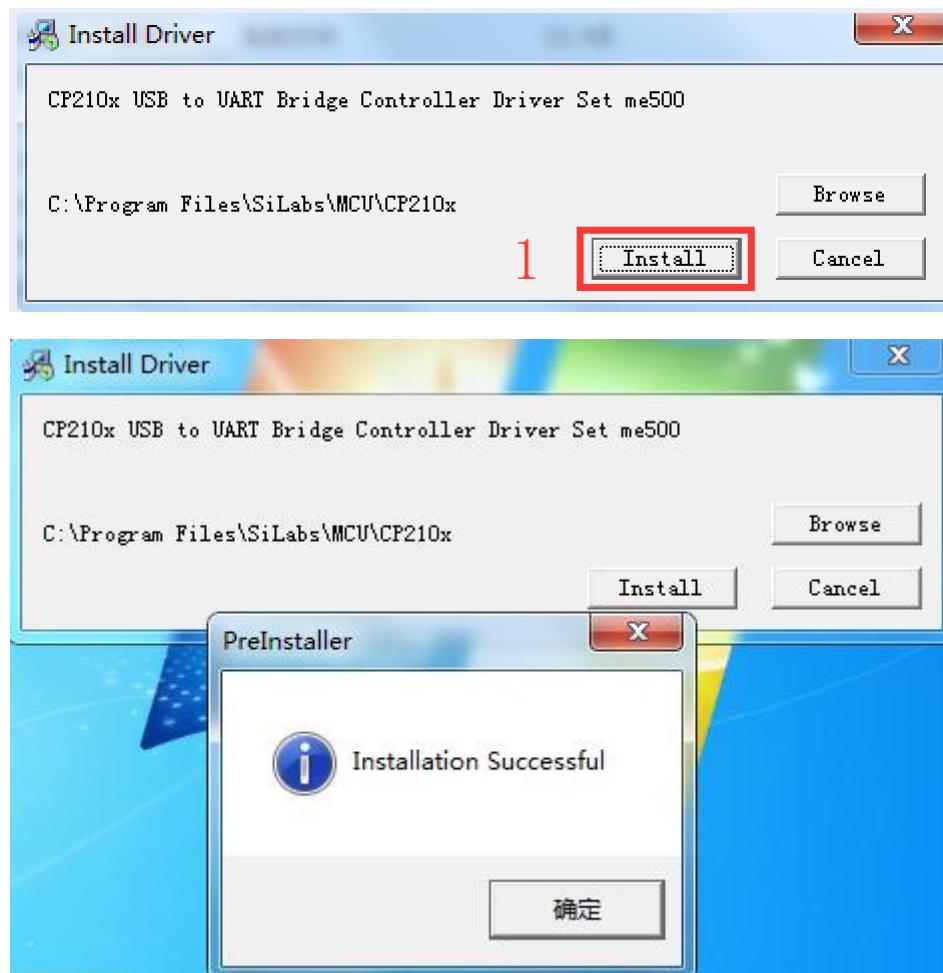
```
192.168.0.141 x serial-com4  
zhaolinhai@boardcon:~/zh1/0_rk3288/6_rk3288/3288/em3288_mid_android-7.1/RKTools/linux  
/Linux_Pack_Firmware/rockdev$ ls  
afptool mkupdate.sh package-file rkImageMaker update.img  
Image output readme.txt unpack.sh  
zhaolinhai@boardcon:~/zh1/0_rk3288/6_rk3288/3288/em3288_mid_android-7.1/RKTools/linux  
/Linux_Pack_Firmware/rockdev$ ls output/  
Image MiniLoaderAll.bin package-file parameter.txt  
zhaolinhai@boardcon:~/zh1/0_rk3288/6_rk3288/3288/em3288_mid_android-7.1/RKTools/linux  
/Linux_Pack_Firmware/rockdev$ ls output/Image/  
boot.img misc.img resource.img trust.img vendor0.img  
kernel.img recovery.img system.img uboot.img vendor1.img  
zhaolinhai@boardcon:~/zh1/0_rk3288/6_rk3288/3288/em3288_mid_android-7.1/RKTools/linux  
/Linux_Pack_Firmware/rockdev$
```



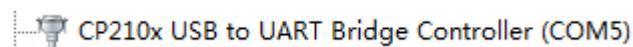
5 Install Tools

5.1 Install CP2102 Driver

Plug the **USB-to-UART cable CP2102** to the PC, unzip **CP2102WIN7.rar** on Windows, then click **preInstaller.exe** to install

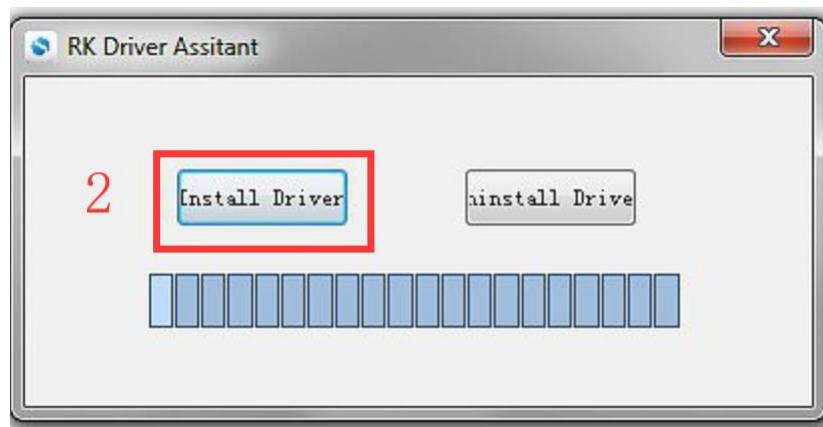
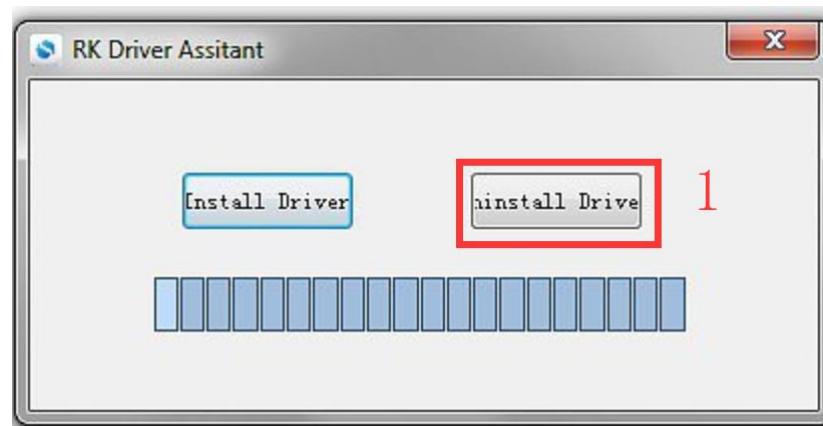


Now the device will be listed under **Device Manager -> PORTS** with unique serial port assigned

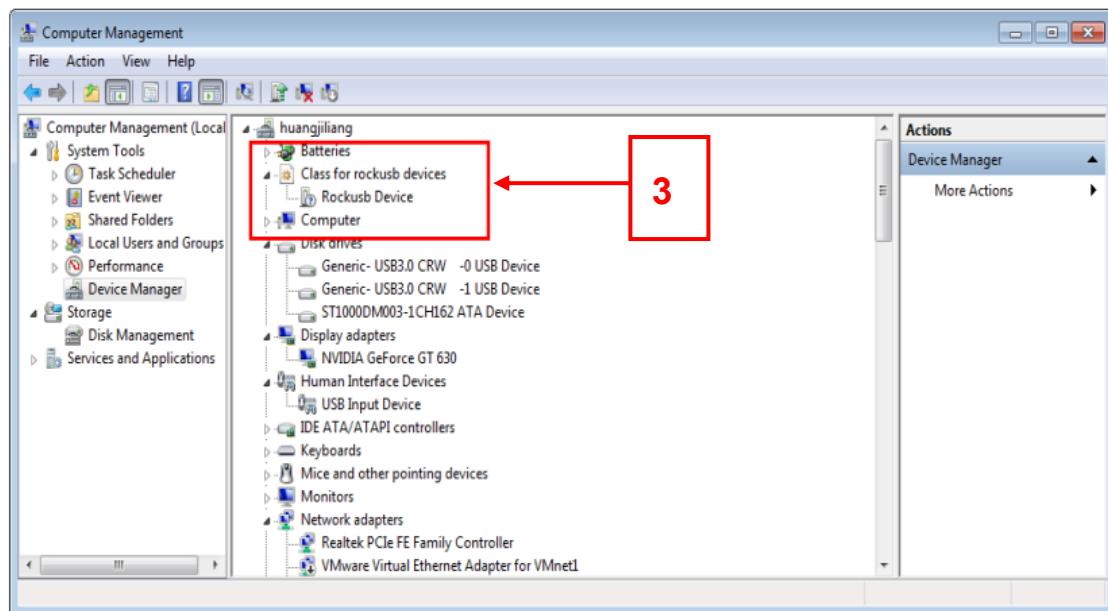


5.2 Install Rockchip Driver Assistant

Path: Release_DriverAssitant/DriverInstall.exe



After the installation is complete, connect the board and PC with Micro USB cable (USB powered), in [Computer Management](#) can see the following information:

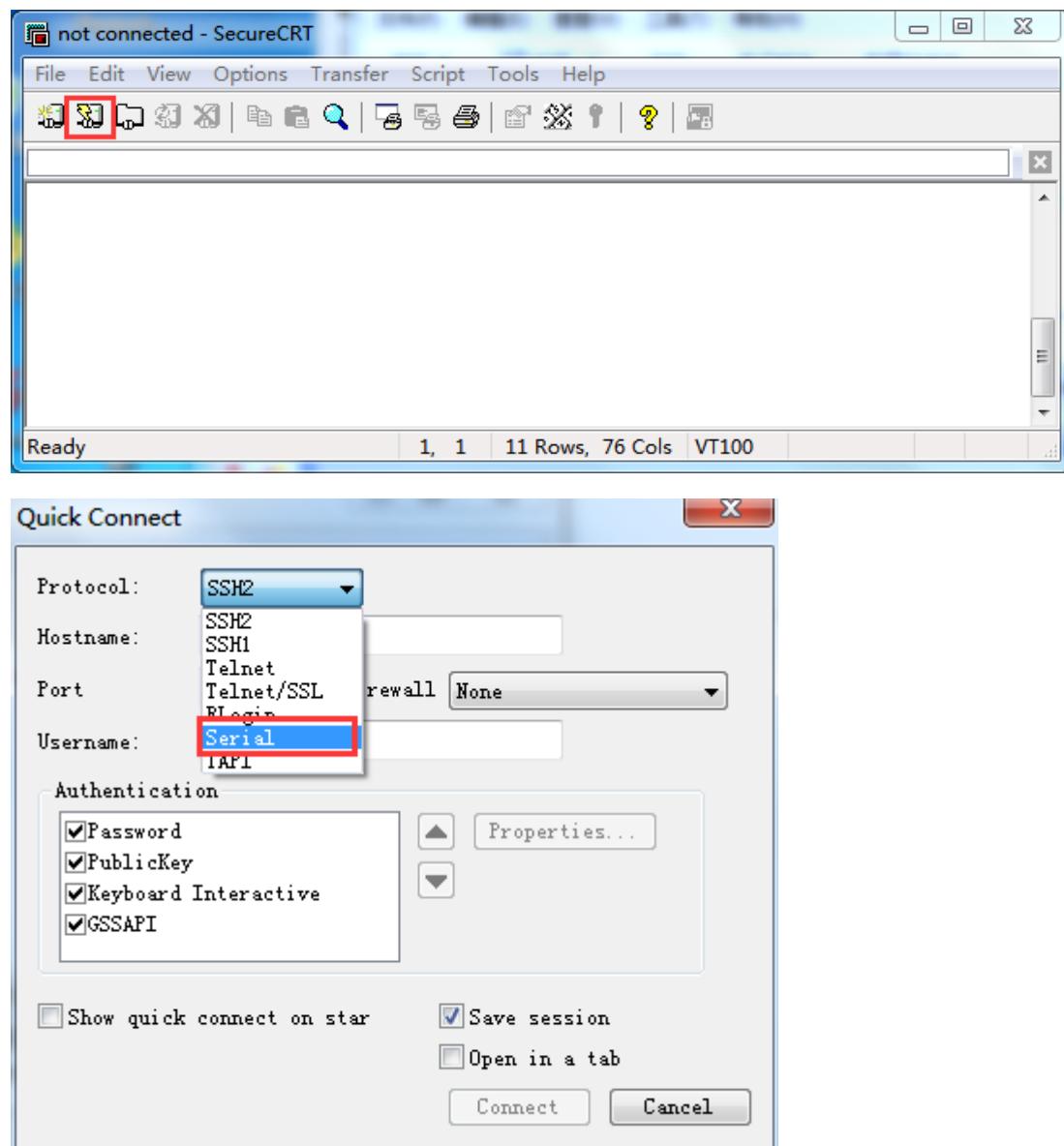




5.3 Install Serial Terminal Tool

The serial terminal SecureCRT is used for debugging. It can be used directly after decompression.

Open SecureCRT.exe after copy to PC (path: tools\windows\SecureCRT.exe), then click the icon **Quick Connect** to config.



Set the parameters as follow:

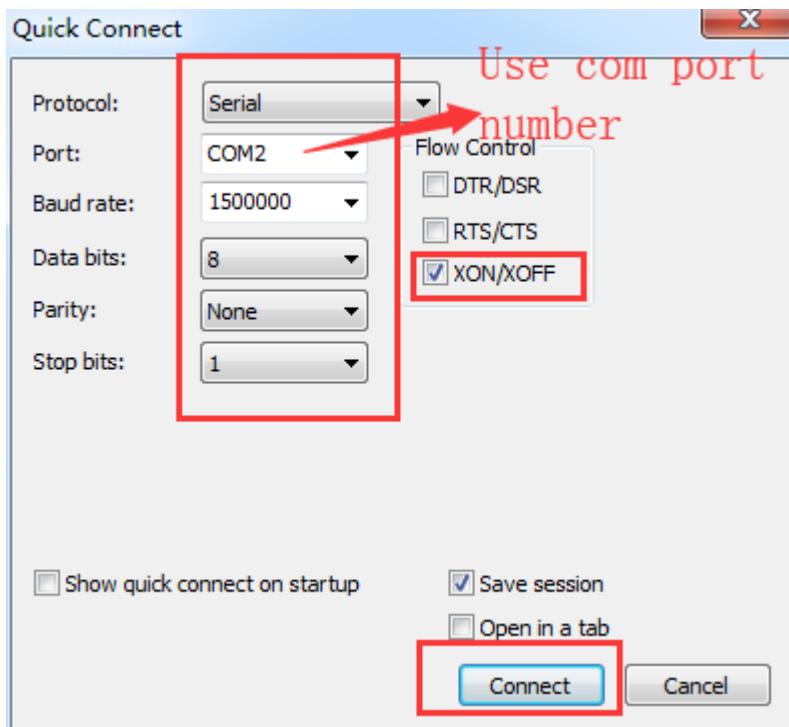
Protocol: Serial

Port: To be specified by user PC

Baud rate: 1500000

Please check XON/XOFF but not RTS/CTS

Check Save session



After all, click **connect**

Illusion: If open more than one serial terminal tools, and they use the same serial port, there will be reported **the port is busy**.

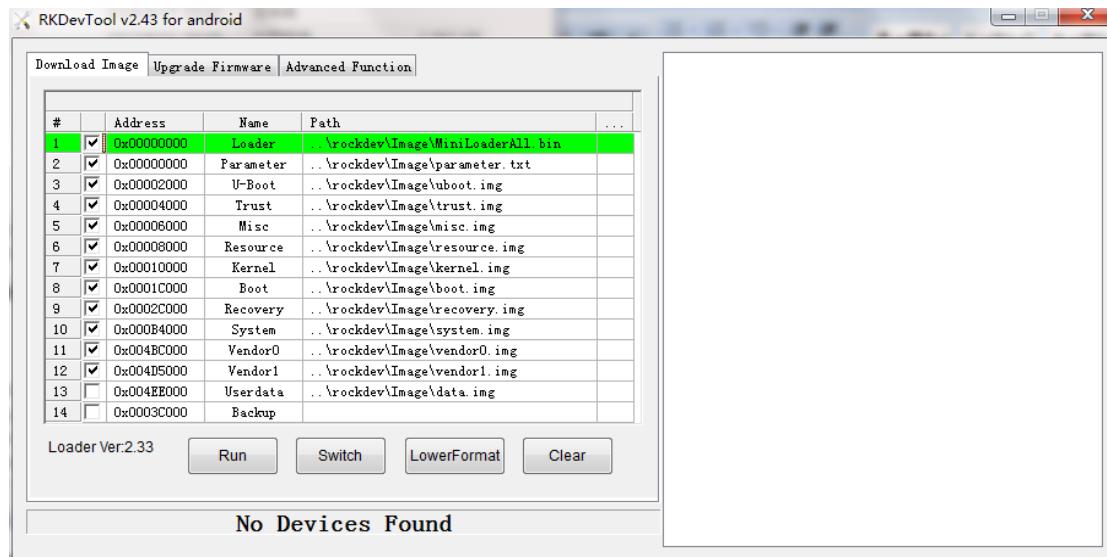
Solution: Turn off the serial tool that unnecessary.



6 Burn Images

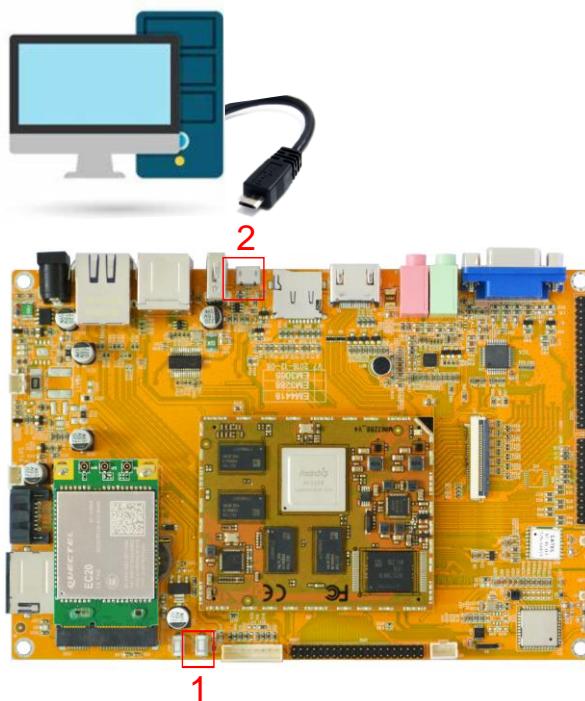
Step 1, unzip **AndroidTool_Release_v2.43** on Windows.

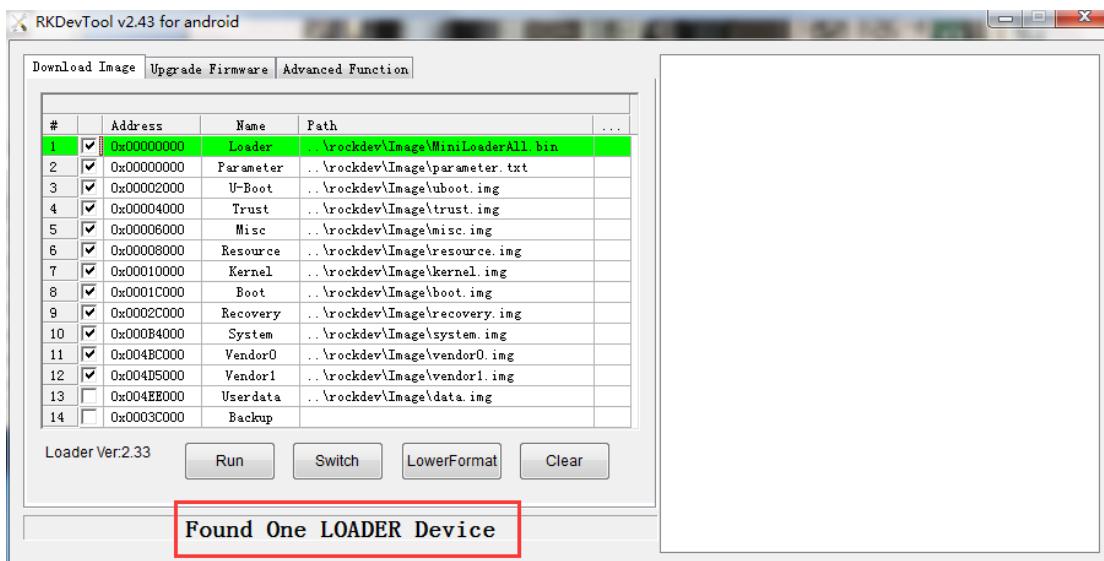
Step 2, open **AndroidTool.exe** (*Path: AndroidTool_Release_v2.43\AndroidTool.exe*)



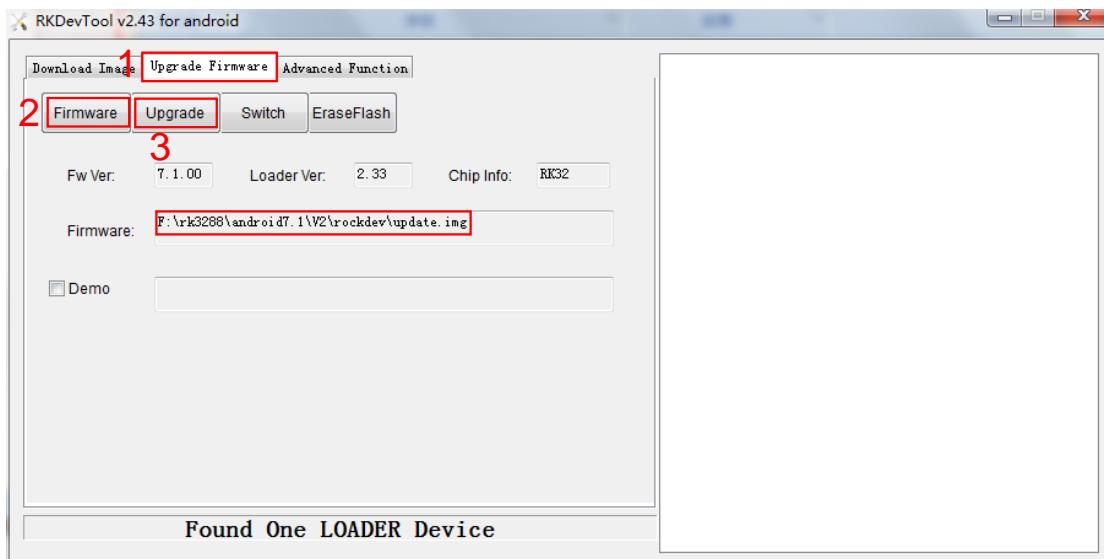
Step 3, keep pressing the **Recover Key**, then connect PC and development board with Micro USB cable until the windows PC shows **Found one LOADER Device**.

The USB power supply is only available for programming, and the current is not enough for the board to run.

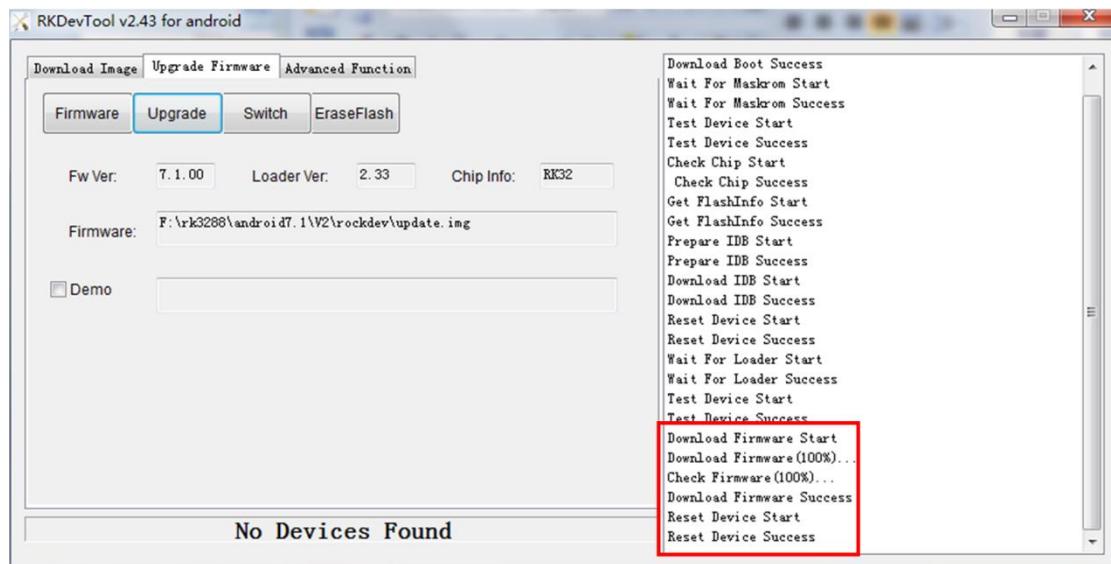




Step 4, click **Upgrade Firmware** -> **Firmware**, select [update.img](#). Click **Upgrade** to flash.



Download completed.

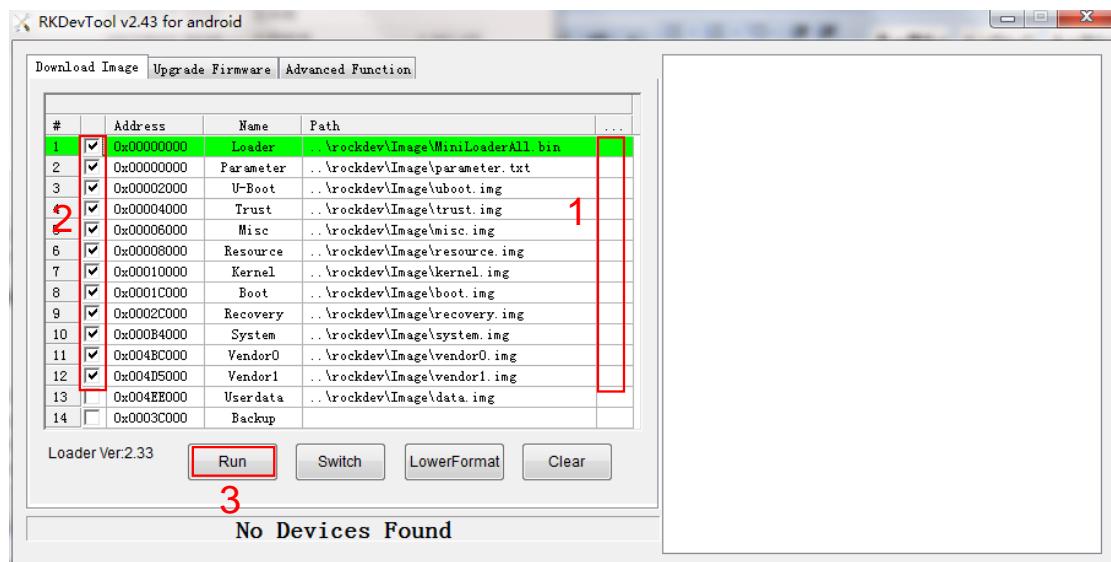


User can also update the firmware separately.

Step 1, Click the column on the right side for the path of the file want to flash.

Step 2, Select the checkbox on the left.

Step 3, Click “run” to flash the image.

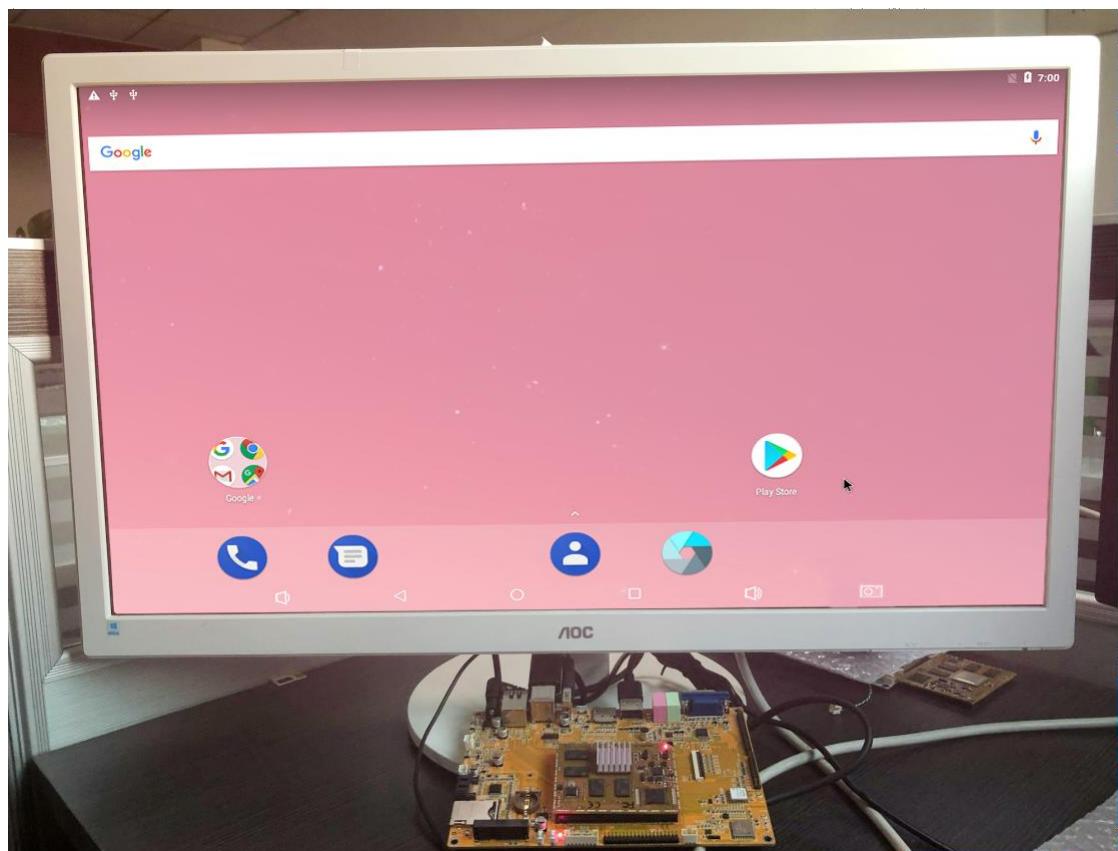




7 Android Application

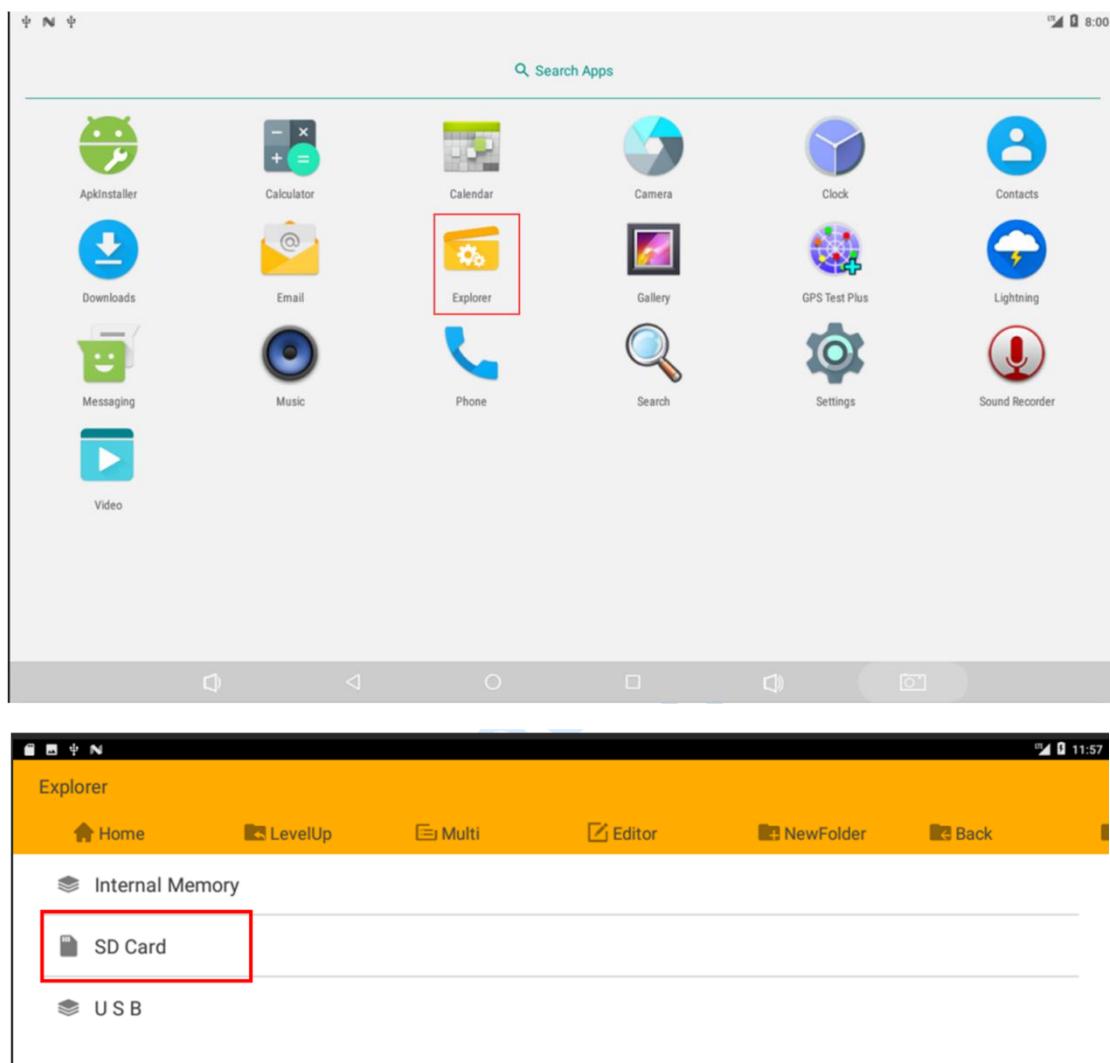
7.1 HDMI Display

Connect the board and monitor with a HDMI cable, then start up.



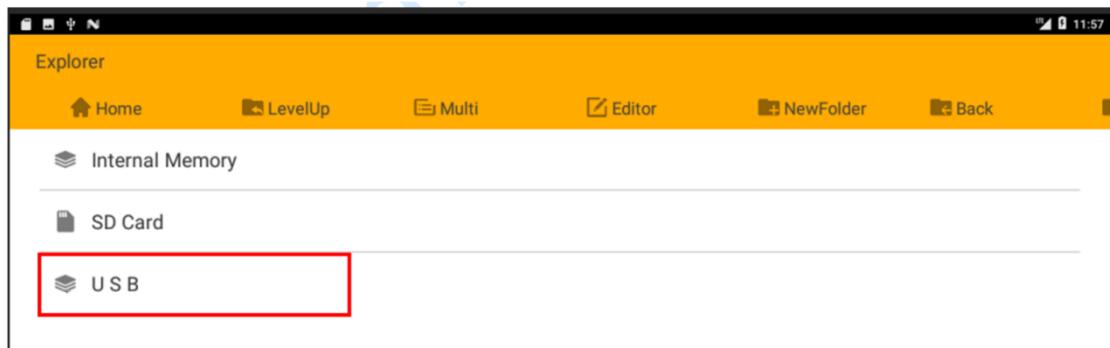
7.2 SD Card

EM3288 supports SD Hot-plug.



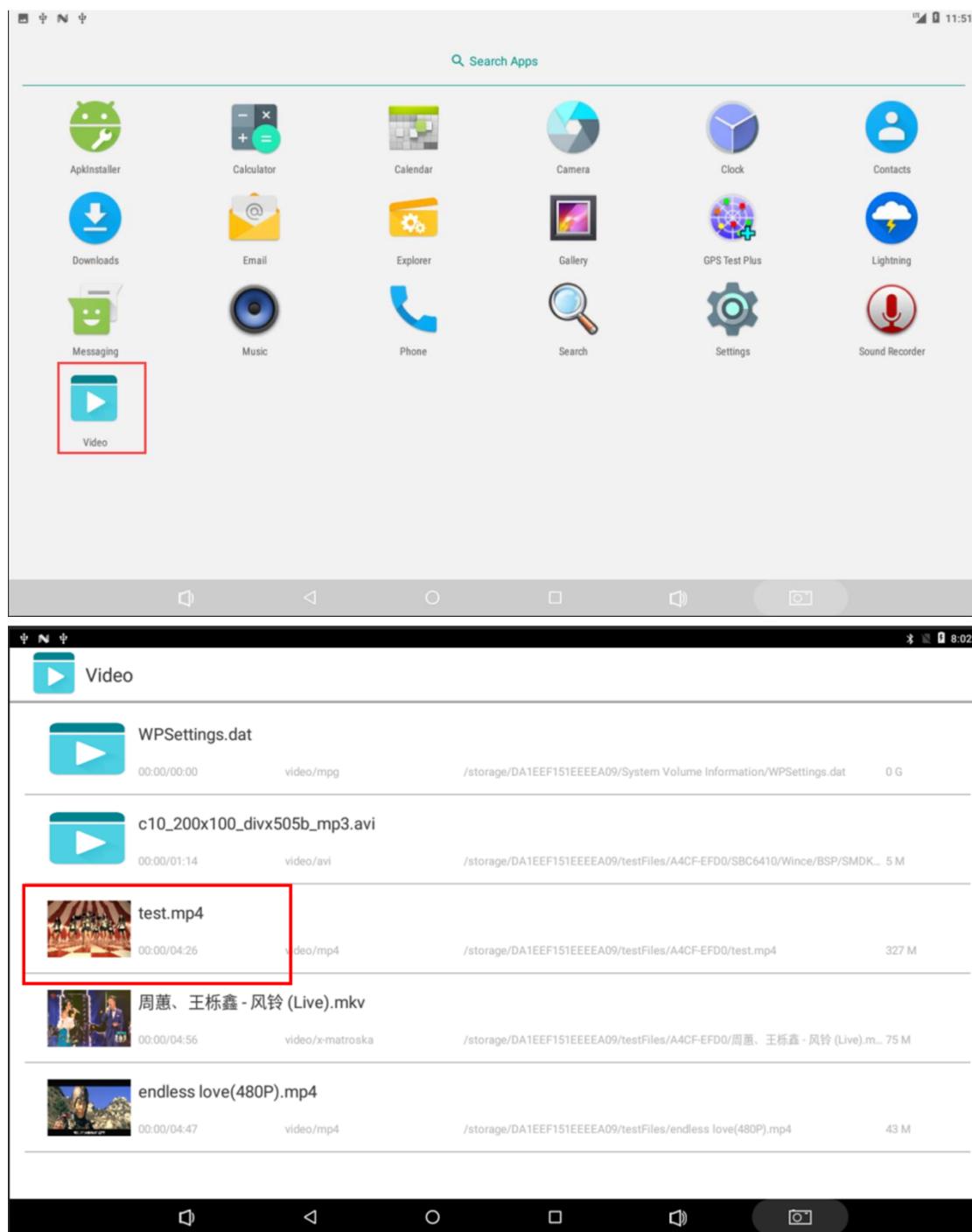
7.3 USB Host

The USB Host can be used to connect USB mouse, USB keyboard, U-Disk or other USB devices.



7.4 Video Player

Open **Video** and select file to play.



7.5 Ethernet

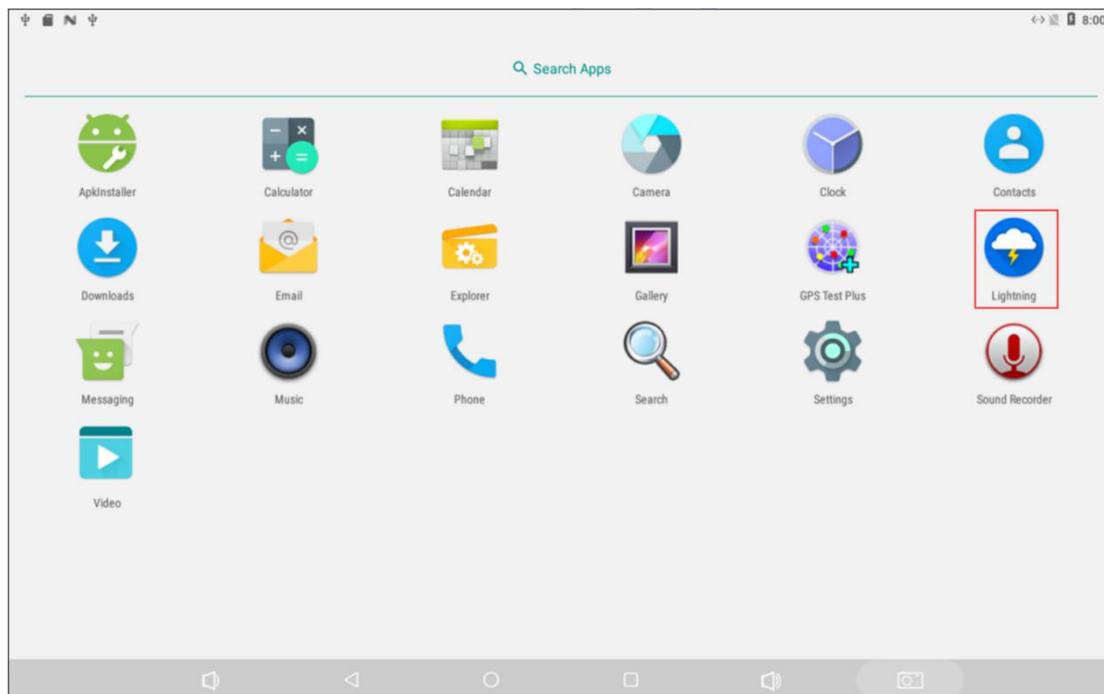
Connect the Board and router with an Ethernet cable (default DHCP=Yes). User can ping URL/IP at terminal, or open the browser to test Network.

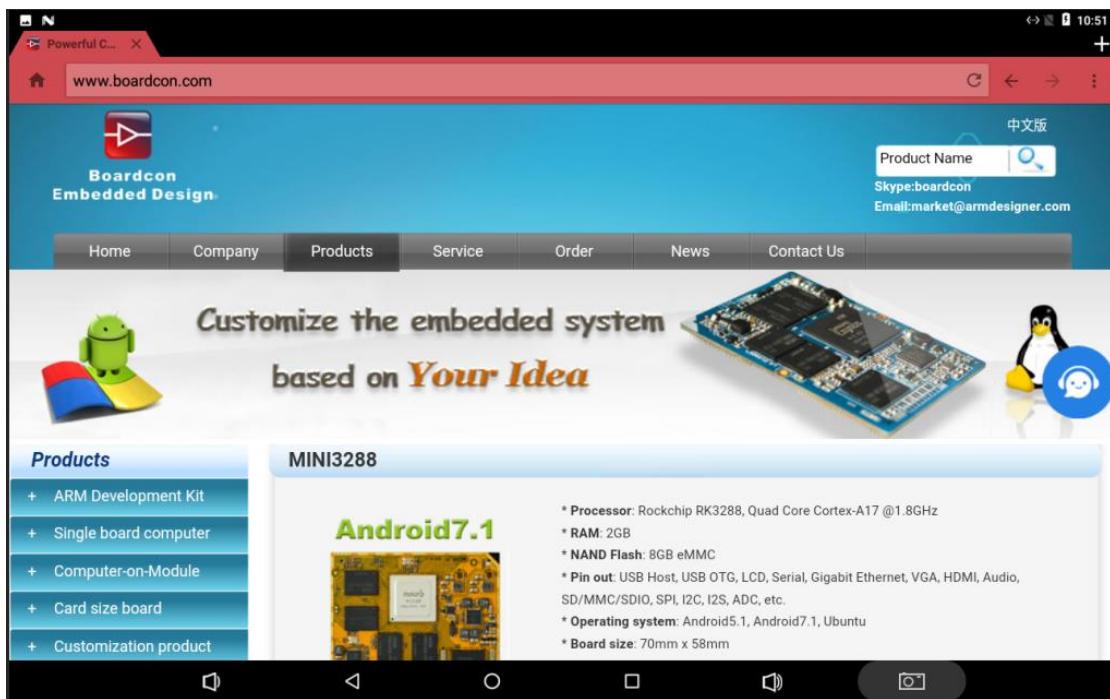
```
# su
# ping www.baidu.com
```



```
rk3288:/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr ce:82:be:ad:2c:71
          inet  addr:192.168.0.149  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::cc82:beff:fead:2c71/64 Scope: Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:149 errors:0 dropped:0 overruns:0 frame:0
             TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:15002 TX bytes:2160
             Interrupt:41

rk3288:/ # ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38: icmp_seq=1 ttl=56 time=7.61 ms
64 bytes from 14.215.177.38: icmp_seq=2 ttl=56 time=7.78 ms
64 bytes from 14.215.177.38: icmp_seq=3 ttl=56 time=7.68 ms
64 bytes from 14.215.177.38: icmp_seq=4 ttl=56 time=7.25 ms
64 bytes from 14.215.177.38: icmp_seq=5 ttl=56 time=7.65 ms
64 bytes from 14.215.177.38: icmp_seq=6 ttl=56 time=6.73 ms
^C
--- www.a.shifen.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 6.732/7.455/7.789/0.366 ms
rk3288:/ # █
```



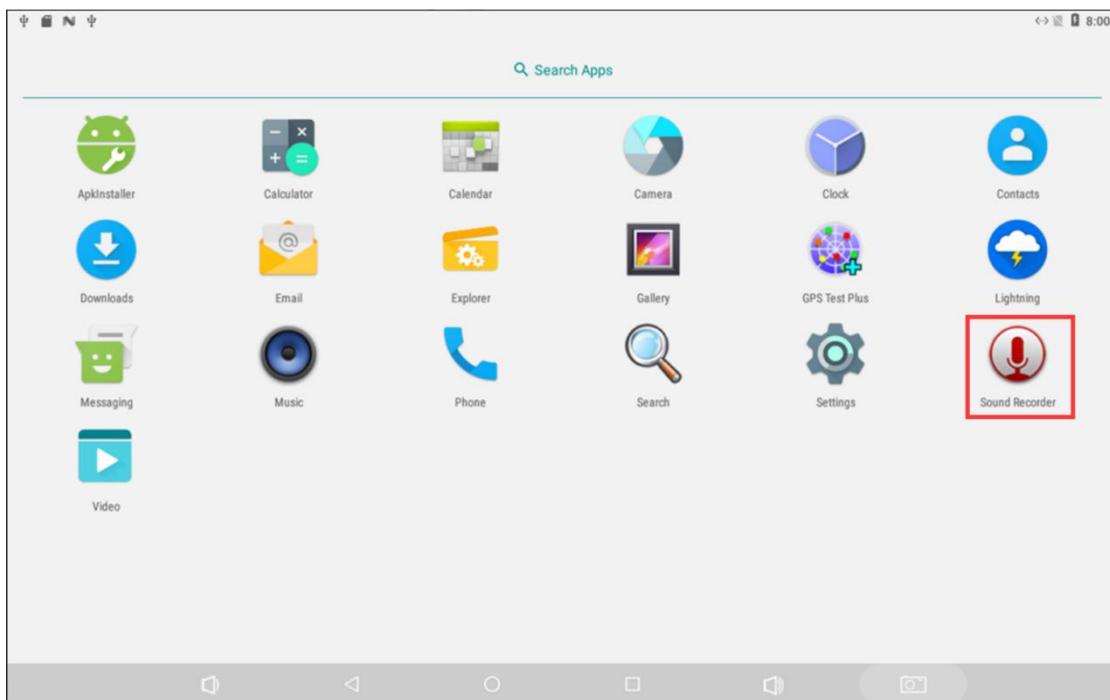


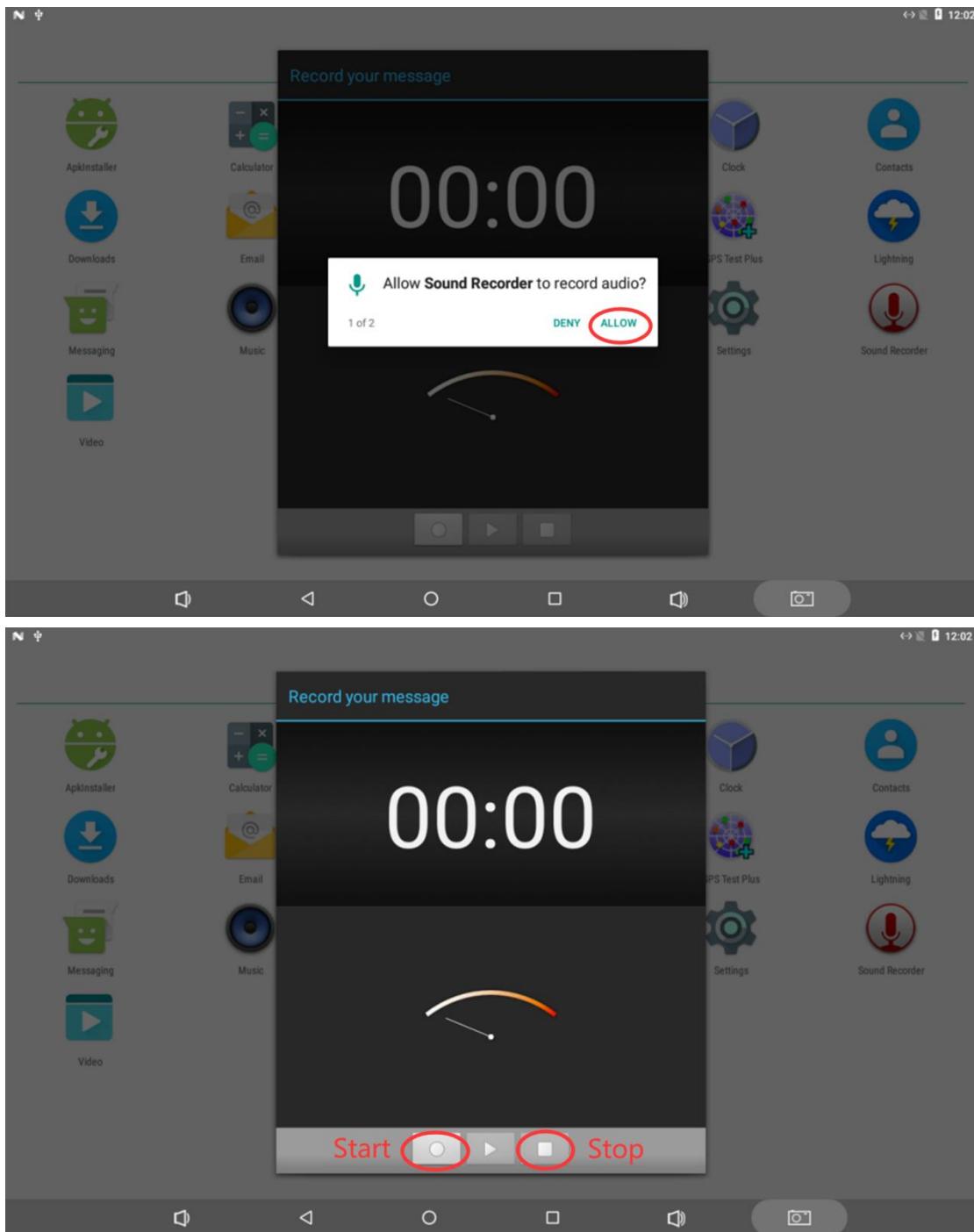
7.6 Record

Step 1, open the APP Recorder in Android.

Step 2, click on the APP to start recording, speech in front of the microphone then can record.

Note: Default microphone recording, if inserted the headset will switch to the headset recording automatically.





After finish recording, click **stop** menu and select **Done** to store file.

The default storage path is

Internal Memory/Android/data/com.android.soundrecorder/files/Download



7.7 RTC

Execute the command **busybox hwclock** at CRT terminal

```
# busybox hwclock
```

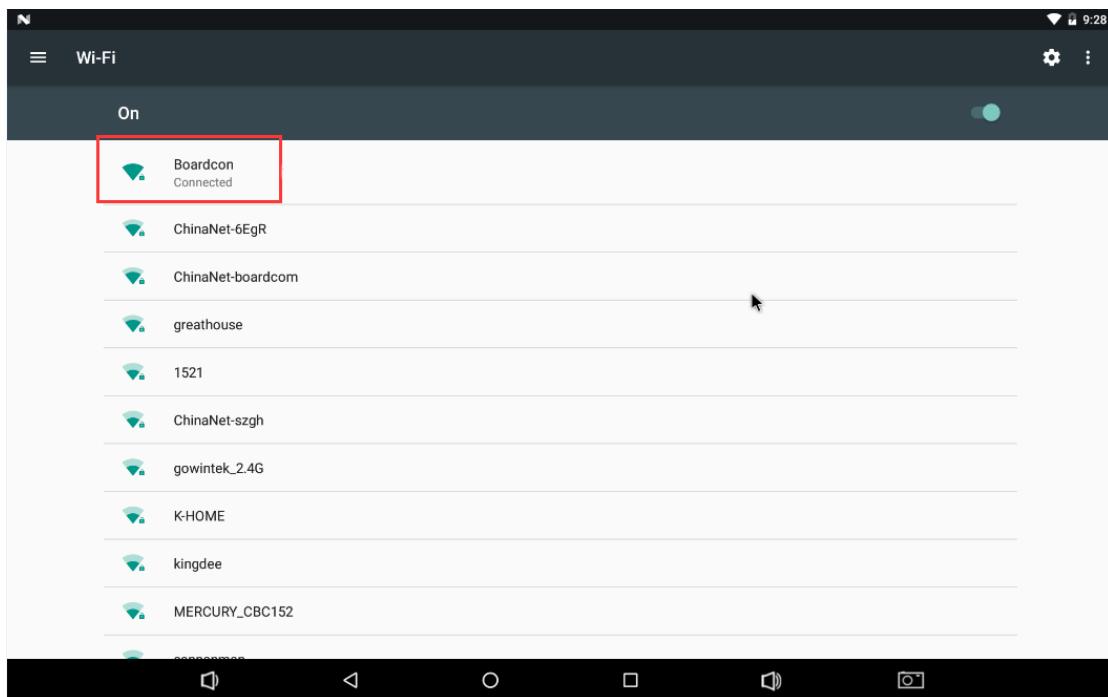
Wait a minute then run **busybox hwclock** again, it can be seen the time has changed.

```
rk3288:/ # busybox hwclock
Sun Jan  1 12:41:54 2017  0.000000 seconds
rk3288:/ # busybox hwclock
Sun Jan  1 12:42:03 2017  0.000000 seconds
rk3288:/ #
```

7.8 WiFi

Connect the WiFi antenna, then click **Settings -> WiFi -> turn on**, select the SSID from the list of available networks and enter the password.

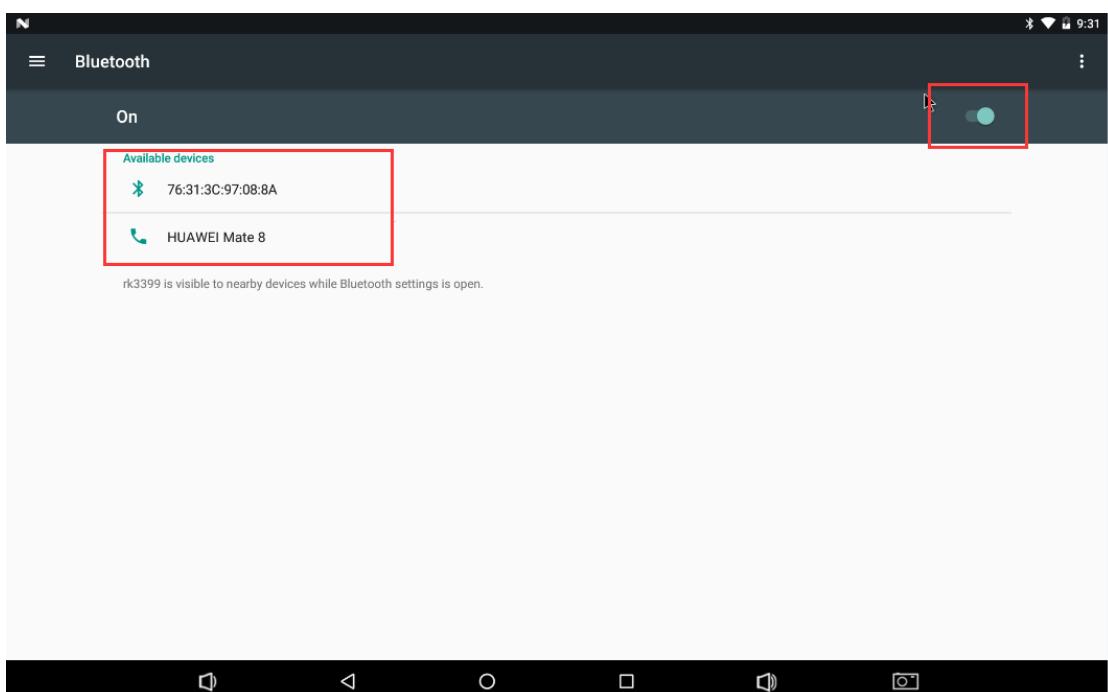
After connected, user can open the browser to browse the web.



7.9 Bluetooth

Click **Settings -> Bluetooth -> turn on**

Select the available device in the list to pair.



After pairing, devices can connect with each other automatically



7.10 4G Network

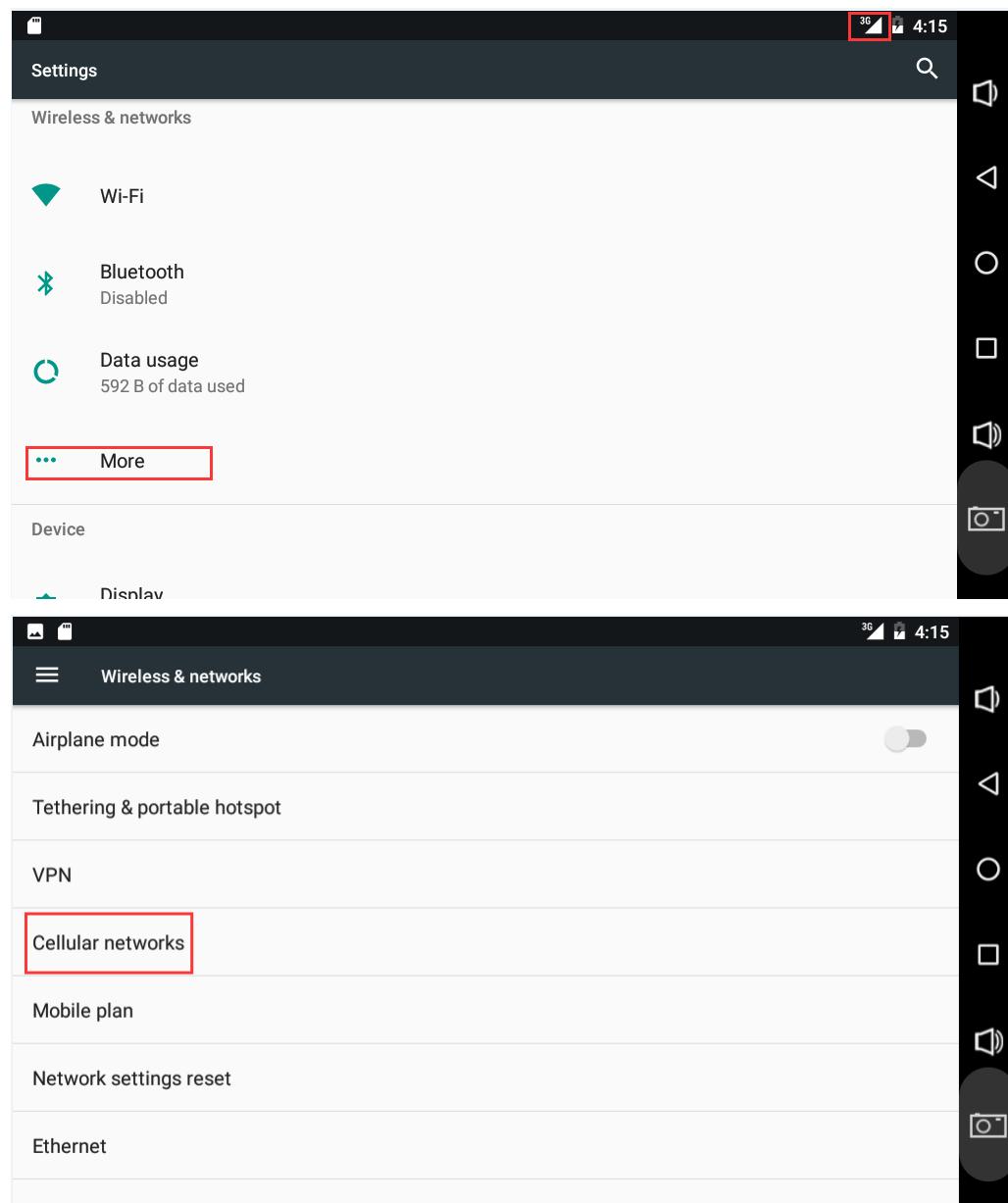
Step 1, Insert 4G module to PCI-E slot (4G model:EC20).

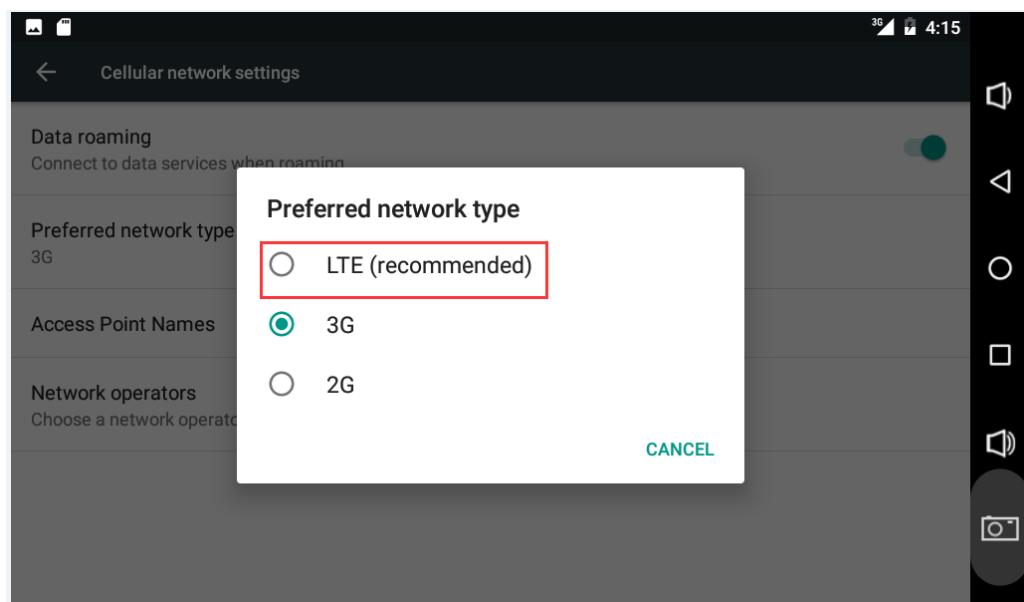
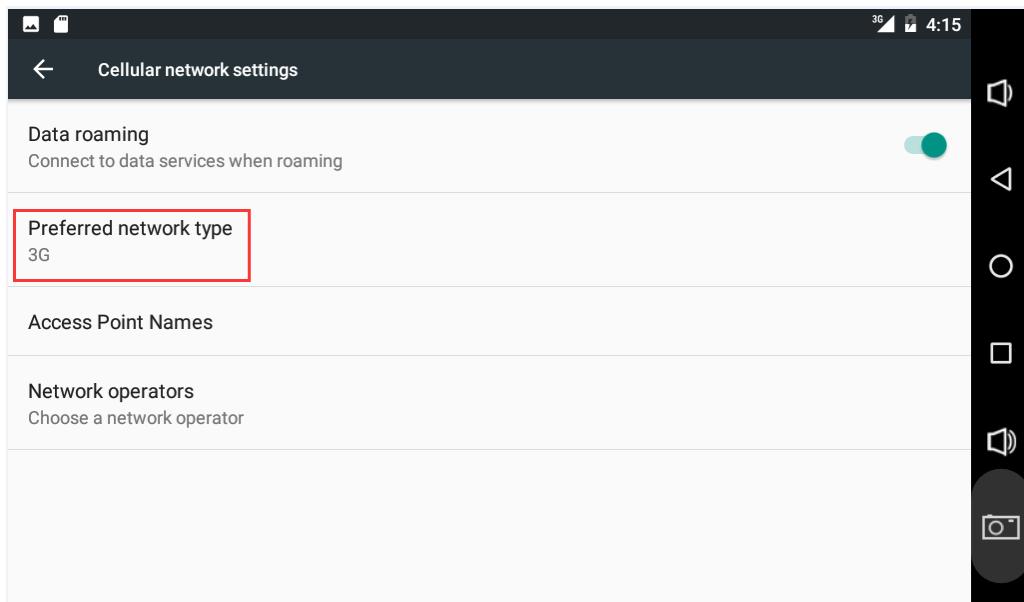
Step 2, Connect antenna and insert SIM card.

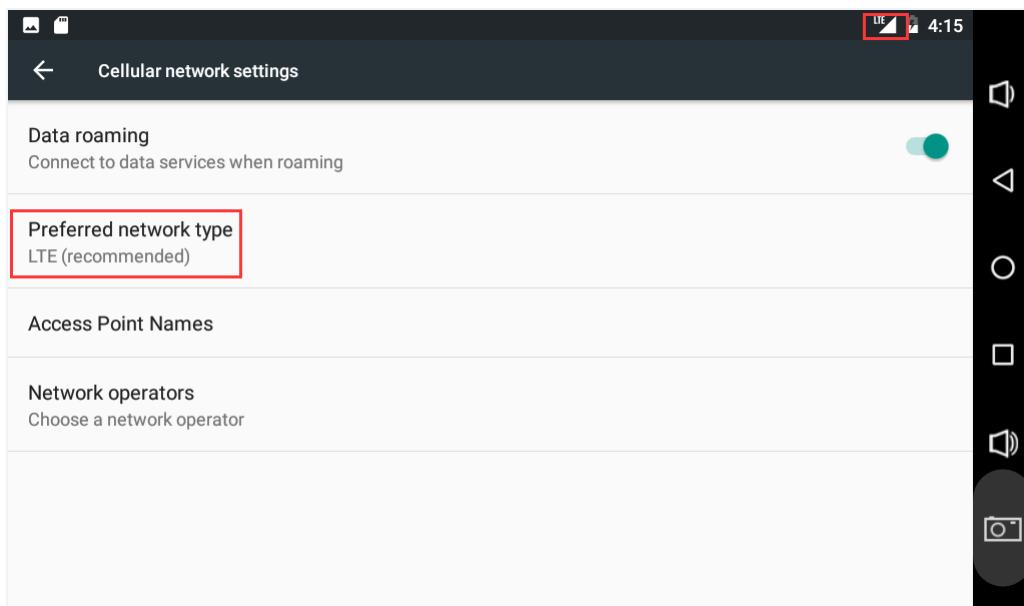
Step 3, The default connection is 3G network after power on.

4G network settings:

Settings -> **Wireless&networks** -> **More** -> **Cellular networks** -> **Preferred network type** -> **LTE**

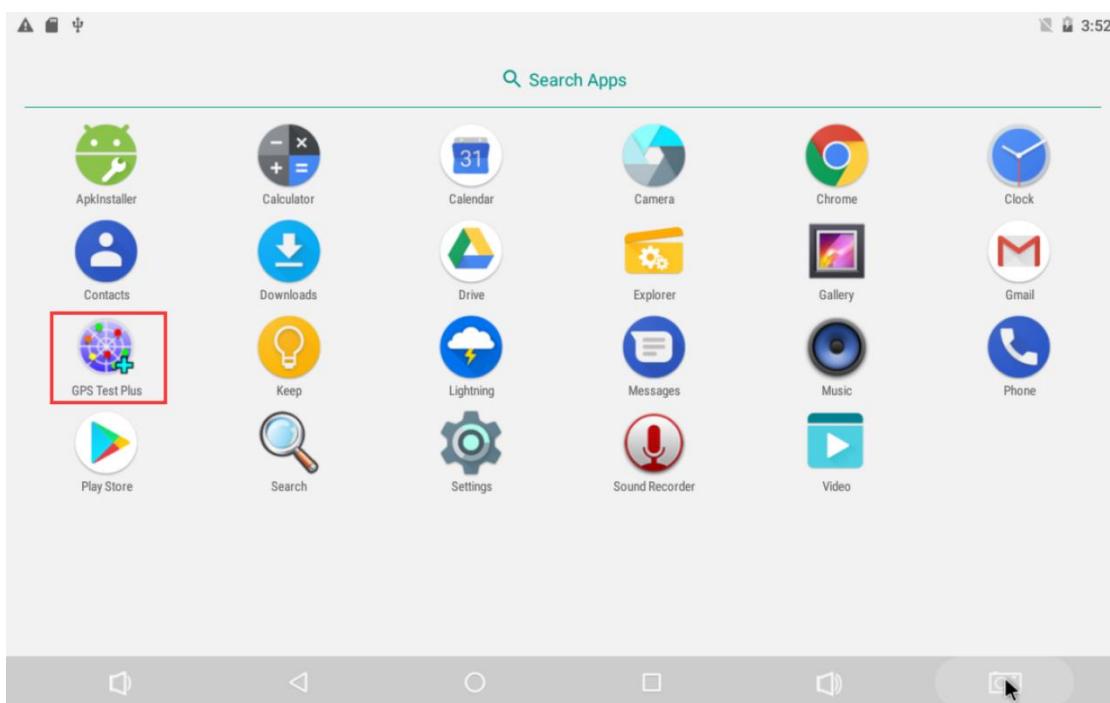






7.11 GPS

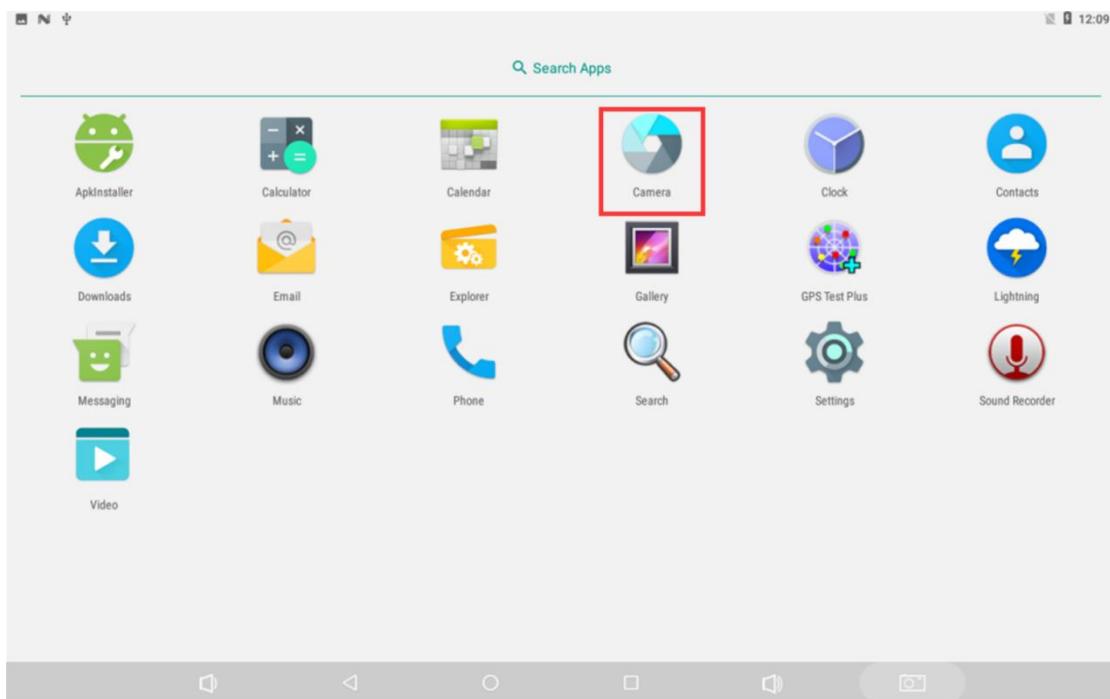
Connect GPS antenna, then power on and install the APP **GPS_test1.2.4.apk** (*path: CD/Tools/*)





7.12 Camera

Connect the camera module (OV13850) to the development board **before power on**, then click the camera app to test.



7.13 Same/Different Display

EM3288 supports **display different** contents or **same** content on **two different** monitors.

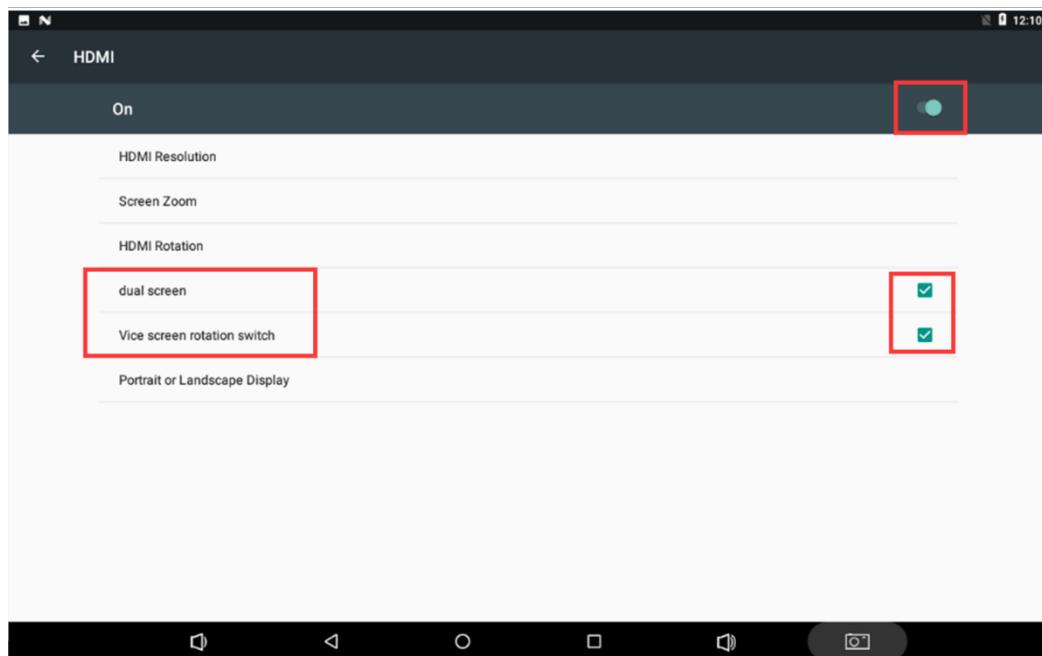
Connect the HDMI and LVDS, the board display **same** content by default.





Config as follow to **display different** content.

Settings -> Display -> HDMI -> Turn On -> Check **dual screen** and **Vice screen rotation**.



Open a video file, press and hold **volume +** and **volume -** simultaneously about two seconds,



Display as follow.





7.14 Miracast

Miracast is a wireless display standard designed for mirroring a smartphone, tablet, or PC's screen to a television without requiring any physical HDMI cables.

Miracast uses WiFi protocol, it must be connected to the same WiFi.

Setting as follows to enable EM3288 Miracast

Settings -> Display -> Cast -> Check Enable wireless display

