MaaXBoard

(AES-MC-SBC-IMX8M-G)

Android User Manual

V1.0

Copyright Statement:

Disclaimer:

- Avnet Manufacturing Services does not take warranty of any kind, either expressed or implied, as to the program source code, software and documents provided along with the products, and including, but not limited to, warranties of fitness for a particular purpose; The entire risk as to the quality or performance of the program is with the user of products.

Regulatory Compliance:

- MaaXBoard single board computer has passed the CE & FCC certification.

Revision History

Rev.	Description	Author	Date
V1.0	Initial version	Sandy	20200301

Catalog

Chapter 1 Introduction

1.1 Package Content

The content of software release package is subject to the actual release sources. For the file structure and instructions, refer to the following table:

Release Folder tree

01Doc	Description
MaaXBoard-Android-ReleaseNote-Vxx.pdf	Release Note
MaaXBoard-Android-UserManual-Vxx.pdf	User Manual
MaaXBoard-Android-DevelopmentGuide-Vxx.pdf	Development Guide
01AndroidSourceCode	**Description**
Readme.txt/xx.tar.gz	Git Library path or Source Code package
02AndroidShipmentImage	**Description**
MaaXBoard-AndroidShipmentImage-Vxxx.img	Android system image file
03AndroidTools	**Description**
xxx.sh	Shell script to generate Android Image, etc.

1.2 Feature List

- U-Boot version: 2018.03
- Kernel version: 4.14.78
- Evaluation image Android P(9)
- Development based on NXP i.MX 8M
- Micro SD boot
- 1 Gigabit Ethernet (RJ45)
- 2 USB 3.0 can work in Host & Device mode
- 2 UART (TTL) include debug port
- External interfaces (I2C, UART, SPI, SAI and GPIO)
- WIFI & BLE 4.2
- MIPI-DSI Display
- HDMI Display and Audio
- MIPI camera and USB Camera
- Bluetooth audio
- USB audio

Chapter 2 **Quick Start**

The default version of MaaXBoard supports boot up from SD Card only. To burn the OOB image to SD Card, refer to Chapter 4. For the hardware connection and accessories details, please check the QSG.

2.1 Select Image

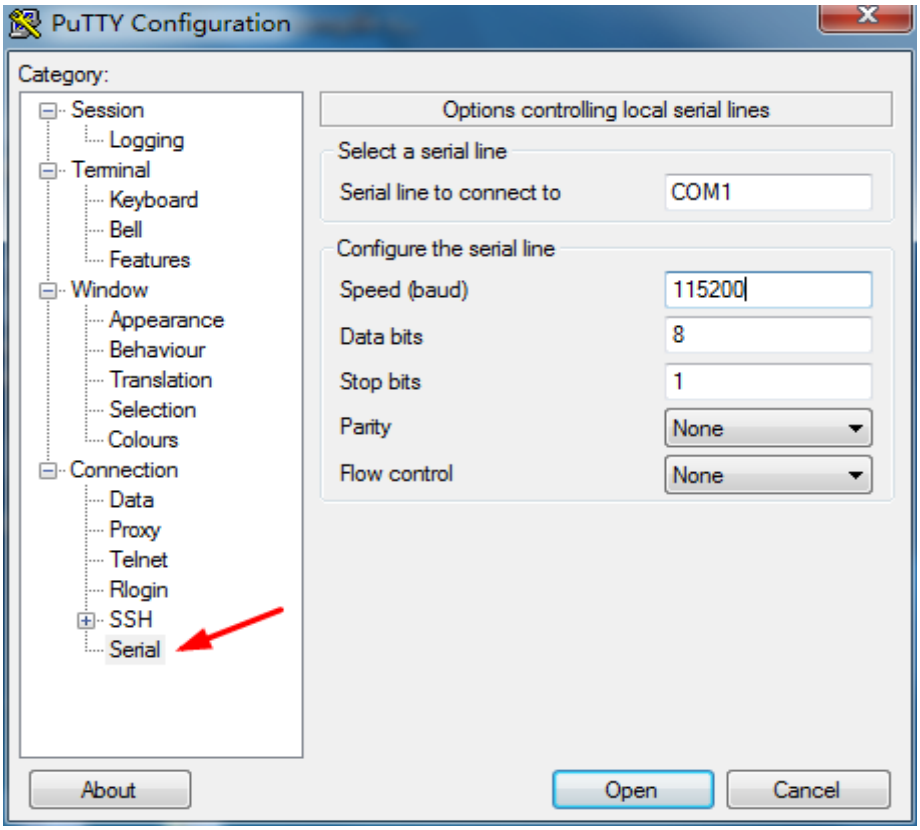
The Android system image support HDMI and MIPI screen, before booting up the board, select different image file according to the screen used. For example:

MIPI-DSI screen: MaaXBoard-AndroidShipmentImage-SDcard-MIPI-V1.0.2r03.img

HDMI screen: MaaXBoard-AndroidShipmentImage-SDcard-HDMI-V1.0.2r03.img

2.2 Boot from SDCard

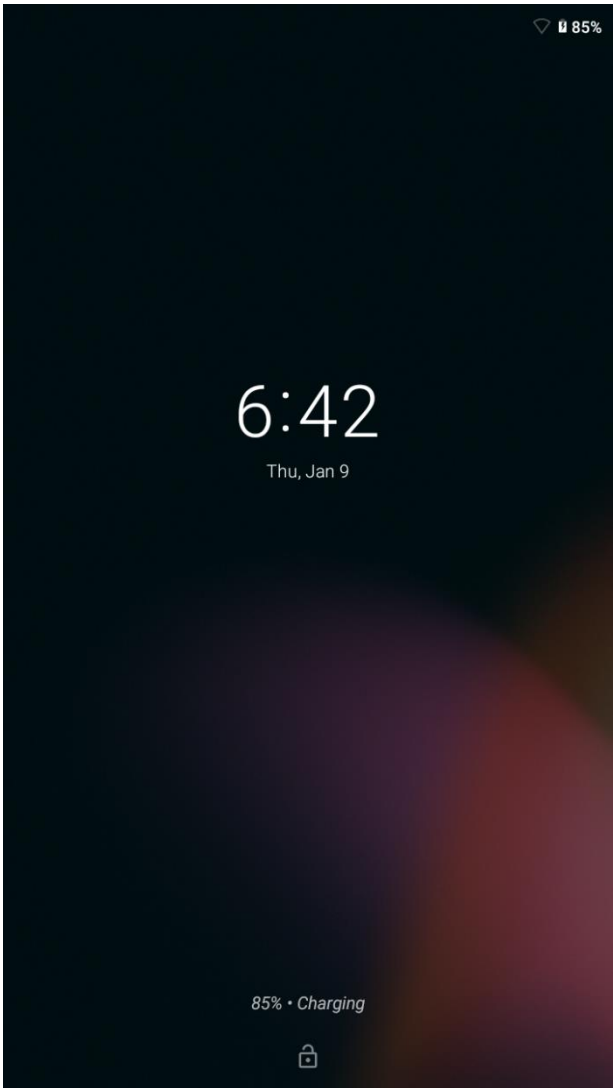
- Install the Serial Communication software (e.g. PUTTY), select the corresponding port number, baudrate as 115200, data bits as 8, stop bits as 1, parity as none.

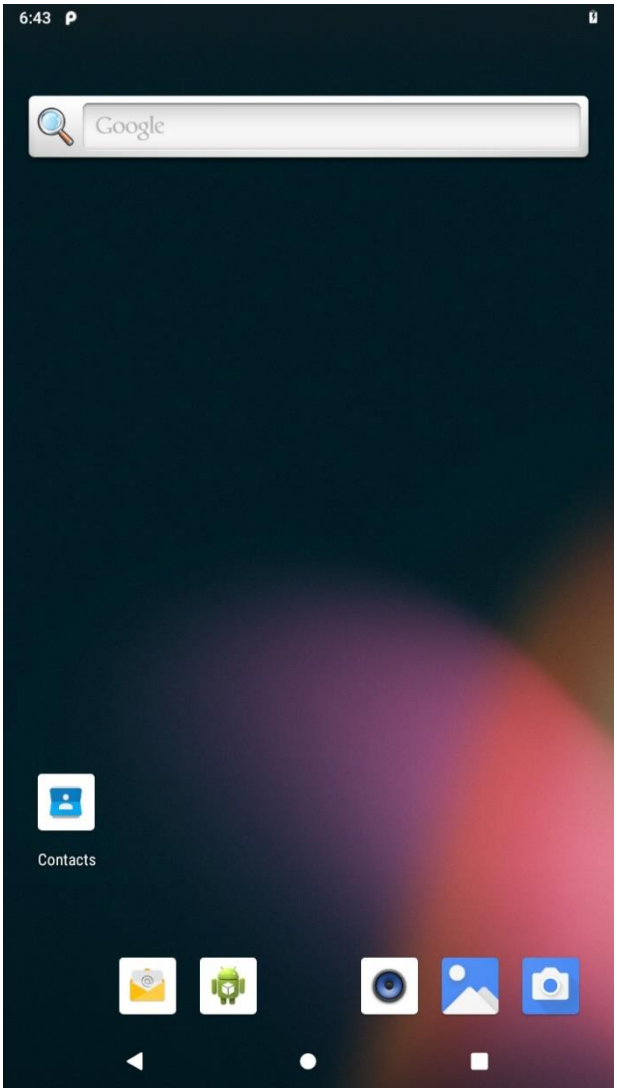


- Connect the debug interface to PC with USB to TTL converter. Pin 6, 8 and 10 of J10 to the GND, RXD and TXD pin of the USB to TTL converter. (optional)
- Connect displayer: (Choose one from following)

- Connect HDMI displayer to J9, users may need to connect USB mouse and keyboard as well.
- Connect MIPI-DSI screen to J16

- Insert the SD card (with pre-burned image) into the card slot J2.
- Powered the board with a 5V, 2A, Type-C interface power (to J4).
- When the system start up, the screen will show the Android boot up GUI.
- After the boot, the screen will display the Android lock screen wallpaper.
- Slide up to unlock the screen and enter HOME page.



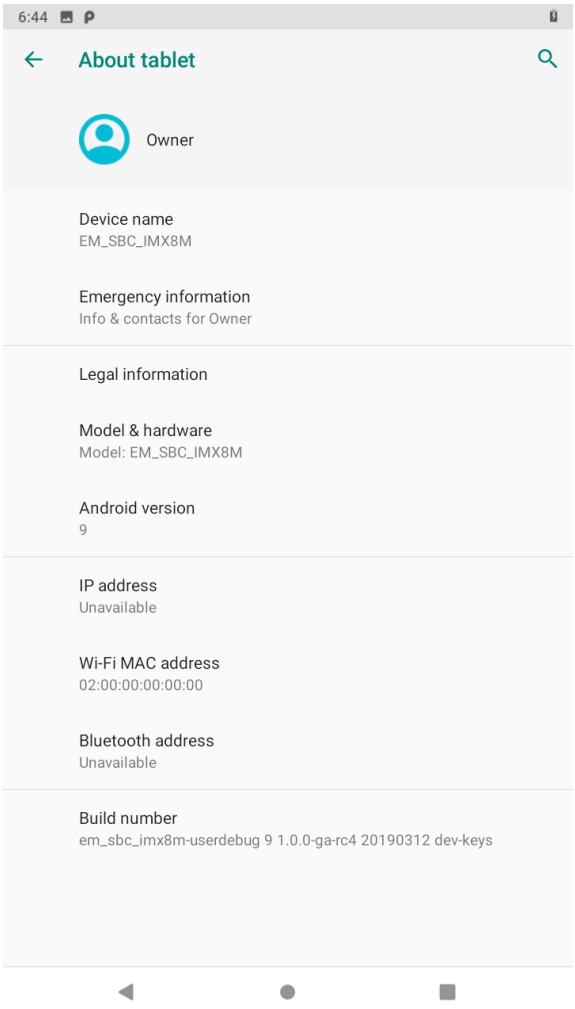


Chapter 3 Feature Configuration & Introduction

This section will use MIPI-DSI screen version as example, it introduces how to configure or use each function of MaaXBoard in Android OS. If you are using an HDMI displayer, the basic operation is consistent, while the screen direction might be different. Most HDMI display screens don't have the touch screen function; we can connect the USB mouse and keyboard to instead of using touch screen operation. First of all, please refer to the previous chapter and boot up the system. Then configure or use the functions according to the following guidance.

3.1 Software version

Go to HOME page, slide up to show the APP list. Go to “Setting” -> “System” -> “About tablet”. Then you will find software version as follows:

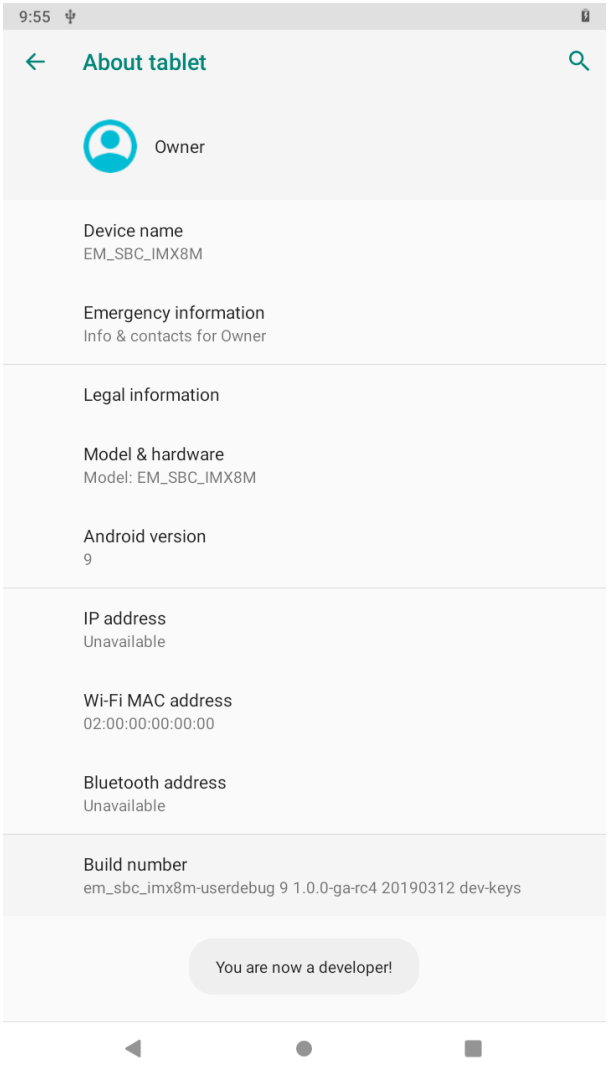


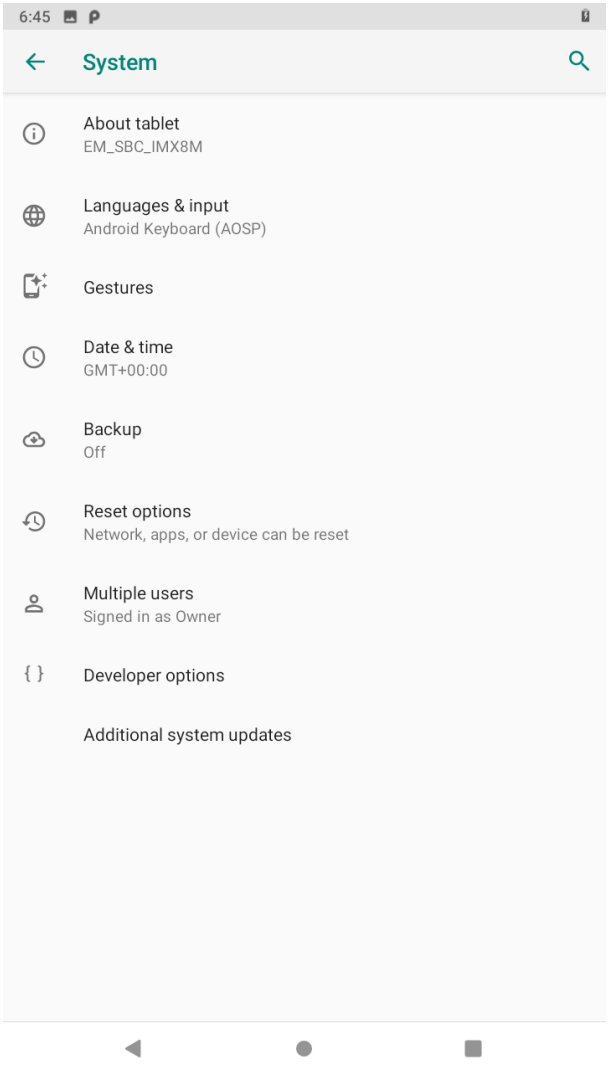
3.2 Developer options

3.2.1 Enable Developer options

The develop options screen is hidden by default, to make it visible, go to “Setting” -> “System” -> “About tablet” and tap “Build number” for 7 times. When you tap the option, it will prompt: You are now x steps away from being a developer. Continue tap, until the system prompt: You are now a developer! Then go back to the “System” page, you will find “Developer options”. Tap on this option, then you can enable or disable the developer mode.

Note: After disabling the Developer options, system settings screen will hide it again.

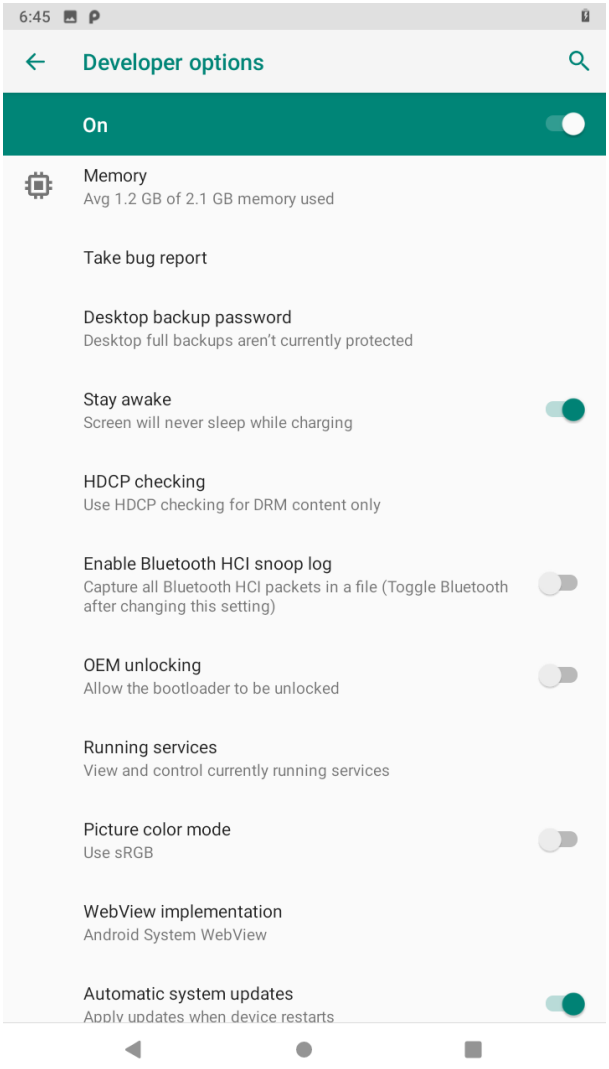


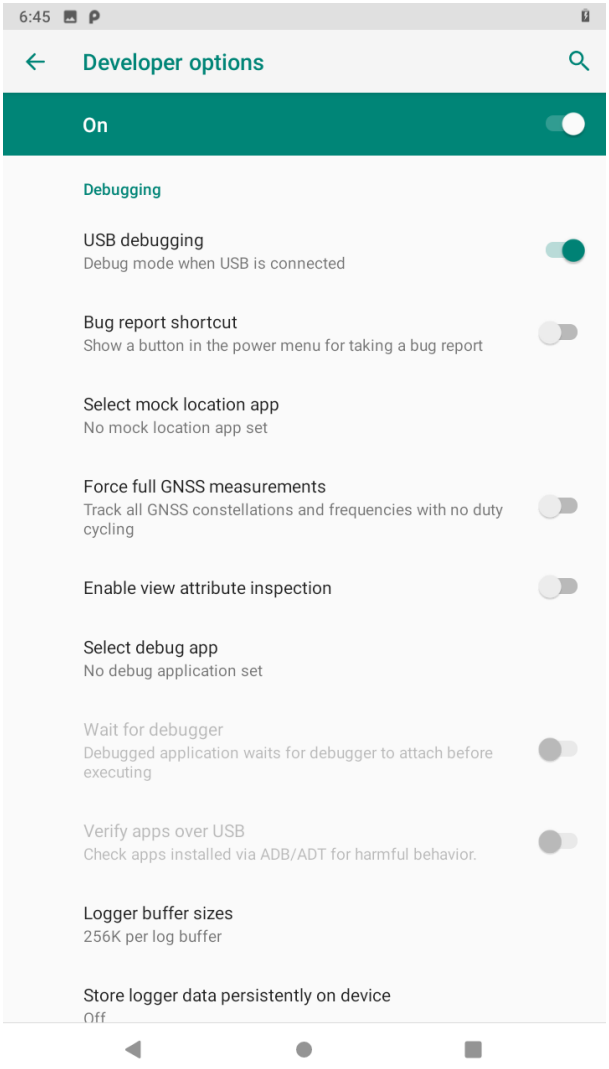


3.2.2 Common Important Options

Developers options include some important settings, while ordinary users may not need this, it's very important for developers, such as:

- Stay awake
- USB debugging
- Default USB configuration
- Input
- Drawing
- Apps





3.3 USB Debug

In Android development, we often use the USB debug mode to connect development board. There are many kinds of USB Debug applications on PC. Here we will with Android ADB and WIN10 as an example, to introduce how to use the USB debugging capabilities.

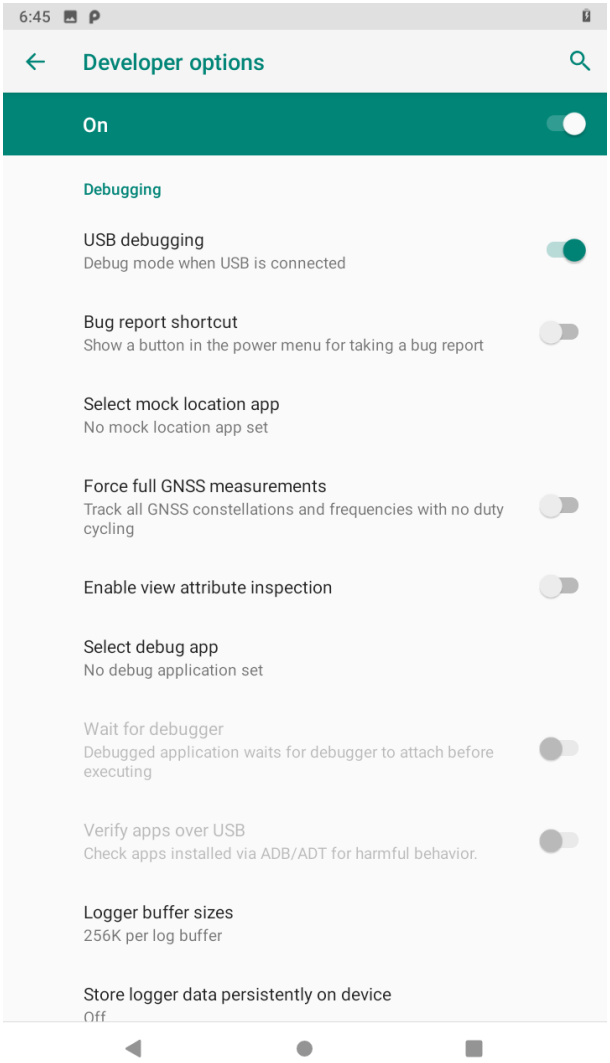
Download address: https://developer.android.com/studio/releases/platform-tools

Or download directly from: https://dl.google.com/android/repository/platform-tools-latest-windows.zip

Reference Usage: https://developer.android.com/studio/command-line/adb

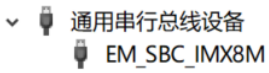
3.3.1 Enable USB Debugging

Enable USB Debugging in Developer options screen:



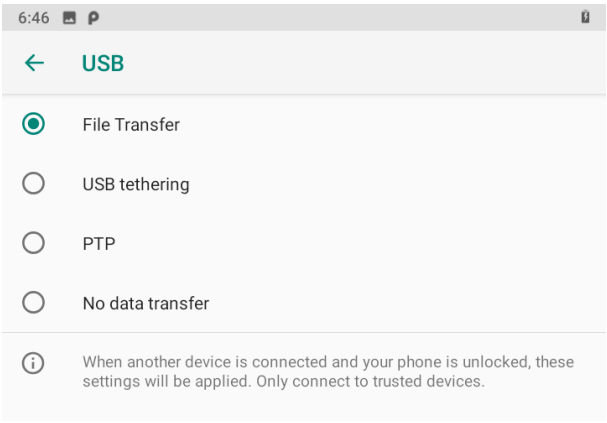
3.3.2 Connect to device

1. After system set up, connect USB0 (the lower one in USB interface J5) to PC, Windows will detect Universal serial bus device: EM_SBC_IMX8M.

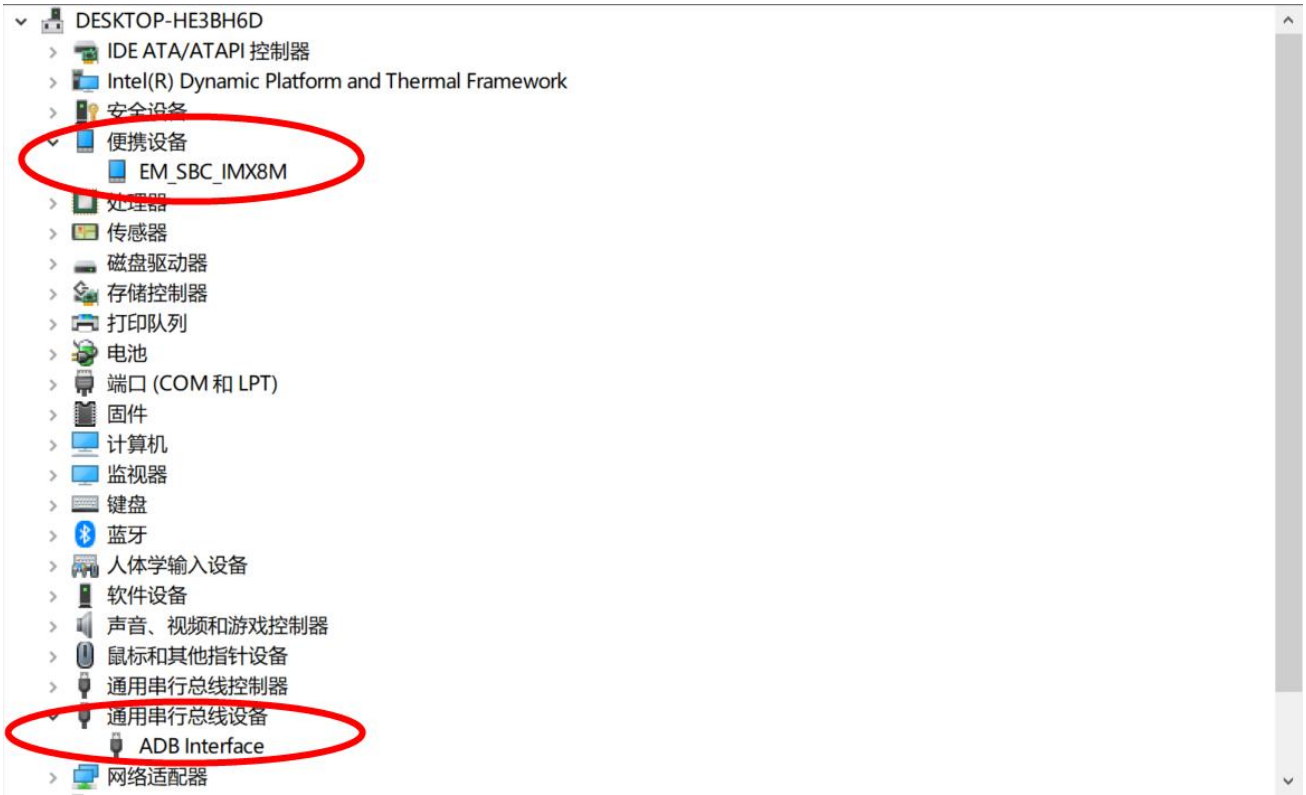


2. Then set the default USB connection to File Transfer in Developer options:

Note: You may need to execute this step each time connect the device.



3. Windows will install the USB driver automatically. After that, you will find the following 2 devices in device manager.



3.3.3 Start and Connect ADB

Run command line tool in Windows, go to the path of adb tool, execute adb devices to know what device instances are connected to the adb server.

```
D:\work\platform-tools>adb devices
* daemon not running; starting now at tcp:5037
* daemon started successfully
List of devices attached
161211d6f060954c            device
```

3.3.4 Issuing ADB Commands

You can issue adb commands to the board. The usage is:

```
adb [-d | -e | -s serial_number] command
```

If there's only one device connected, the adb command is sent to that device by default. Otherwise, you need to use the -d, -e, or -s option to specify the target device to which the command should be directed. For example:

```
D:\work\platform-tools>adb -s 161211d6f060954c pull sdcard ./sdcard2
sdcard/: 5 files pulled, 0 skipped. 0.8 MB/s (464703 bytes in 0.565s)
```

This operation copies the sdcard folder from the device (serial number: 161211d6f060954c) to the developers machine, path of adb tool, sdcard2 folder.

3.3.5 Issue shell commands

You can use the shell command to issue device commands through adb, or to start an interactive shell. To issue a single command use the shell command like this:

```
adb shell shell_command
```

for example:

```
D:\work\platform-tools>adb shell df -h
Filesystem                Size    Used Avail Use% Mounted on
/dev/root                 1.6G    1.0G    612M    65% /
tmpfs                     995M    448K    994M     1% /dev
tmpfs                     995M       0    995M     0% /mnt
/dev/block/dm-1           244M     49M    189M    21% /vendor
/dev/block/mmcblk0p12 8.6G     91M    8.4G     2% /data
/data/media               8.6G     91M    8.4G     2% /storage/emulated
```

To start an interactive shell on a device:

```
adb shell
```

For example:

```
D:\work\platform-tools>adb shell
em_sbc_imx8m:/ $ su
em_sbc_imx8m:/ # ls
acct          d                init.environ.rc                    init.usb.rc             oem
sdcard
bin           data             init.rc                            init.zygote32.rc        postinstall storage
bugreports default.prop init.recovery.freescale.emmc.rc init.zygote64_32.rc proc            sys
cache         dev              init.recovery.freescale.rc         lost+found              product
system
charger       etc              init.recovery.freescale.sd.rc      mnt                     res
ueventd.rc
config        init             init.usb.configfs.rc               odm                     sbin
vendor
em_sbc_imx8m:/ # exit
em_sbc_imx8m:/ $
```

To exit an interactive shell, press Control + D or type exit.

```
D:\work\platform-tools>adb shell
em_sbc_imx8m:/ $ ^D

D:\work\platform-tools>adb shell
em_sbc_imx8m:/ $ exit
```

3.3.6 Install an app

Execute commands:

```
adb install path_to_apk
```

For example:

```
D:\work\platform-tools>adb install D:\work\apk\com.shenyaocn.android.usbcamera.apk
Performing Streamed Install
Success
```

3.3.7 Uninstall an app

1. Check the installed apps

```
adb shell pm list packages
```

For example:

```
D:\work\platform-tools>adb shell pm list packages
package:com.android.cts.priv.ctsshim
package:com.android.internal.display.cutout.emulation.corner
```

```
package:com.example.android.livecubes
package:com.android.internal.display.cutout.emulation.double
package:com.android.providers.telephony
package:com.android.providers.calendar
......
package:com.shenyaocn.android.usbcamera
```

2. Uninstall the app

Execute commands:

```
adb uninstall package name
```

For example:

Search the package name for the app in above result, such as com.shenyaocn.android.usbcamera:

```
D:\work\platform-tools>adb uninstall com.shenyaocn.android.usbcamera
Success
```

3.3.8 Copy files to/from a device

Use the pull and push commands to copy files to and from a device.

To copy a file or directory and its sub-directories from the device, execute the command: adb pull remote local, for example:

```
D:\work\platform-tools>adb -s pull sdcard ./sdcard2
sdcard/: 5 files pulled, 0 skipped. 0.8 MB/s (464703 bytes in 0.565s)
```

To copy a file or directory and its sub-directories to the device, execute the command: adb push local remote, for example:

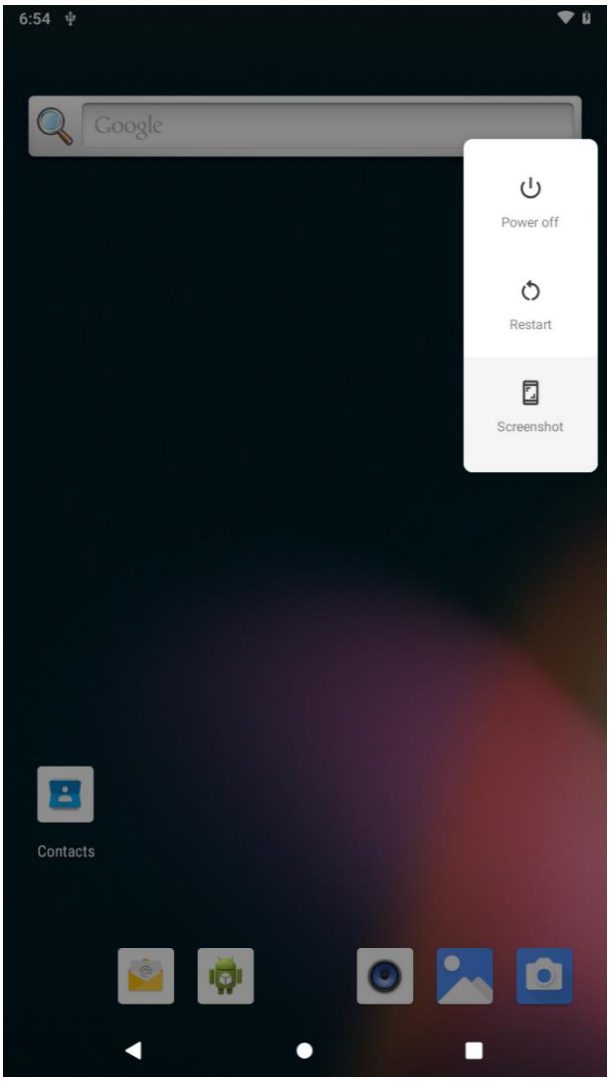
```
D:\work\platform-tools>adb push D:\work\apk\com.shenyaocn.android.usbcamera.apk /sdcard
D:\work\apk\com.shenyaocn.android.usbcamera.apk: 1 file pushed, 0 skipped. 0.3 MB/s (21895207 bytes in 61.445s)
```

3.4 Button

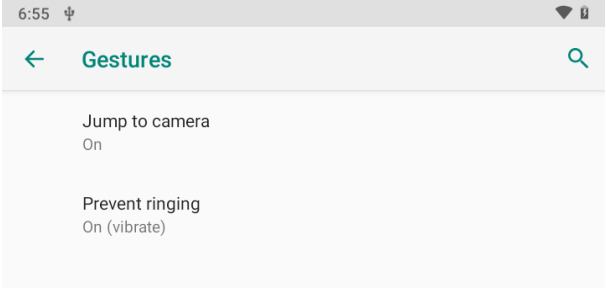
MaaXBoard support 3 physical buttons: BACK, HOME and PWR.

3.4.1 PWR

1. Short press PWR button, system will enter sleep mode, press PWR again, the system will resume from sleep mode.
2. Long press PWR, it will prompt 3 option on the right side of screen, “Power off”, “Restart”, and “Screenshot”.



3. In screen “Settings” -> “System” -> “Gestures”, enable option “Jump to camera”, then you can open the system application: “Camera” quickly by press PWR twice.



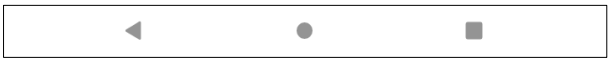
3.4.2 BACK & HOME

HOME: Back to HOME page

BACK: Back to last screen

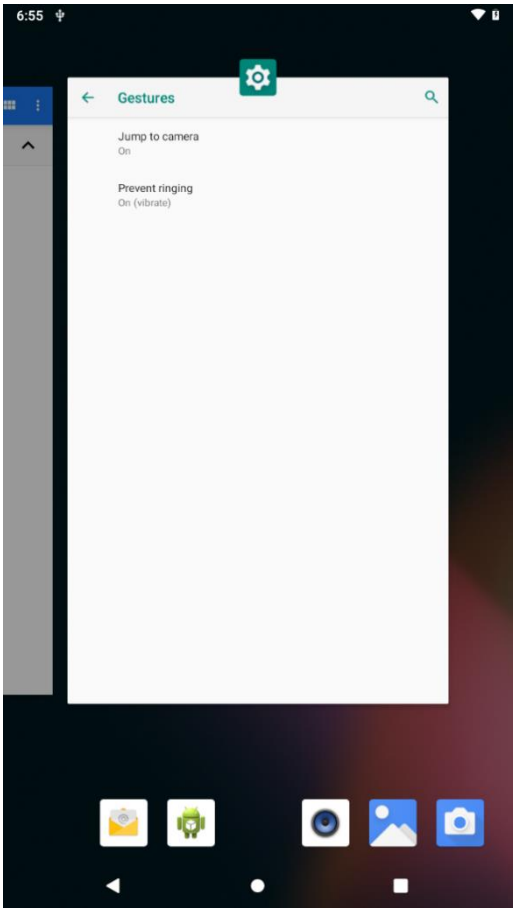
3.4.3 Virtual Button

In the bottom of the screen, there are three virtual buttons, in turn, “BACK” “HOME” “TASK”.



The function of BACK and HOME is the same with psychical button.

Tap TASK button to open recent task list, the you can switch to or clear all running applications.



3.5 Display

3.5.1 Displayer

MaaXBoard Android system supports 2 kinds of displayer: HDMI and MIPI-DSI screen. Users can connect the screen to MaaXBoard, select corresponding system image before boot up the system according to the following table.

Screen Type	Resolution	Interface
HDMI (Default screen)	720p,1080p,2k,4k	J9 (Standard HDMI Interface)
MIPI-DSI	720*1280	J16

3.5.1.1 HDMI

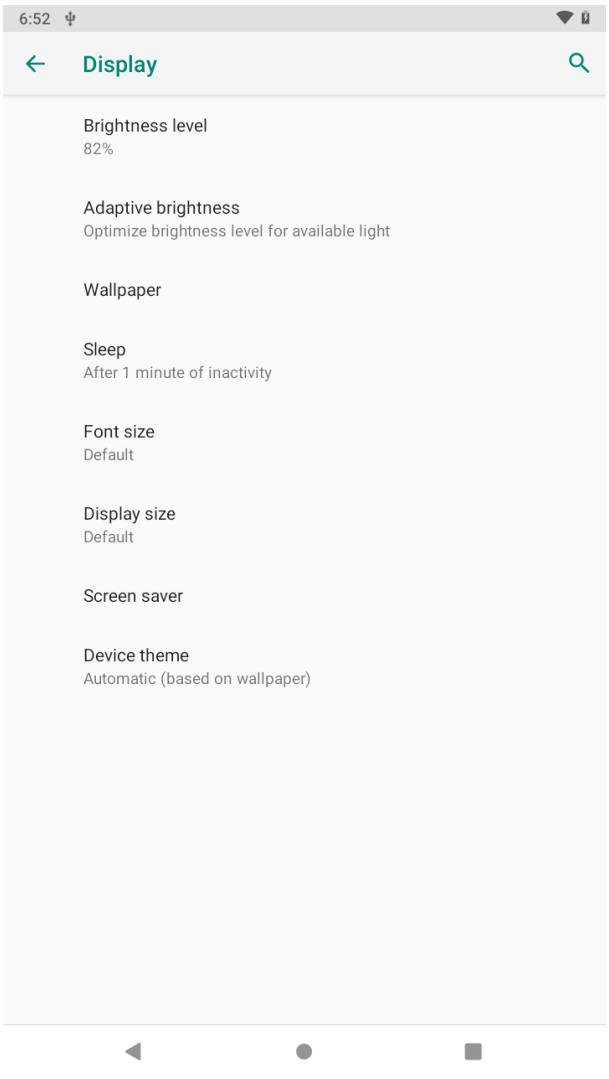
Most models of HDMI displayer can be automatic recognized and matching, if not, please contact us.
In common situation, HDMI displayer is landscape, touch screen is not support, so connect a USB mouse is necessary. HDMI displayer do not support backlight brightness adjustment.

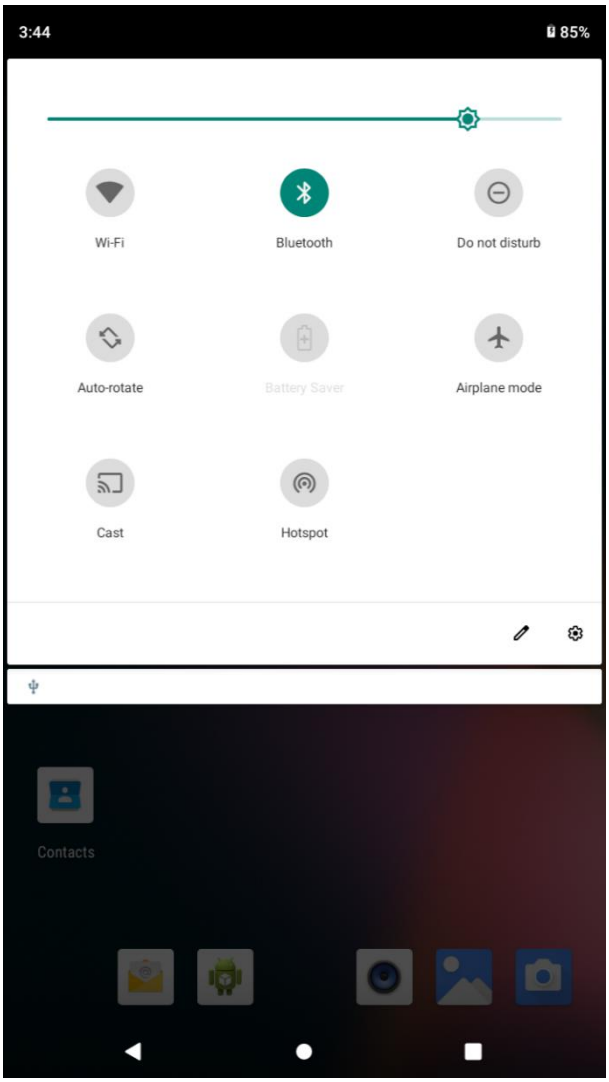
3.5.1.2 MIPI-DSI SCREEN

MIPI-DSI screen is portrait screen, the resolution is 720P. It supports touch screen and backlight brightness adjustment.

3.5.2 Display Settings

Go to “Settings” -> “Display” to configure displayer. In this screen, user could adjust Brightness level, Sleep, Font size, Display size, Screen saver, etc. You can also adjust brightness level by slide down from the top of screen.

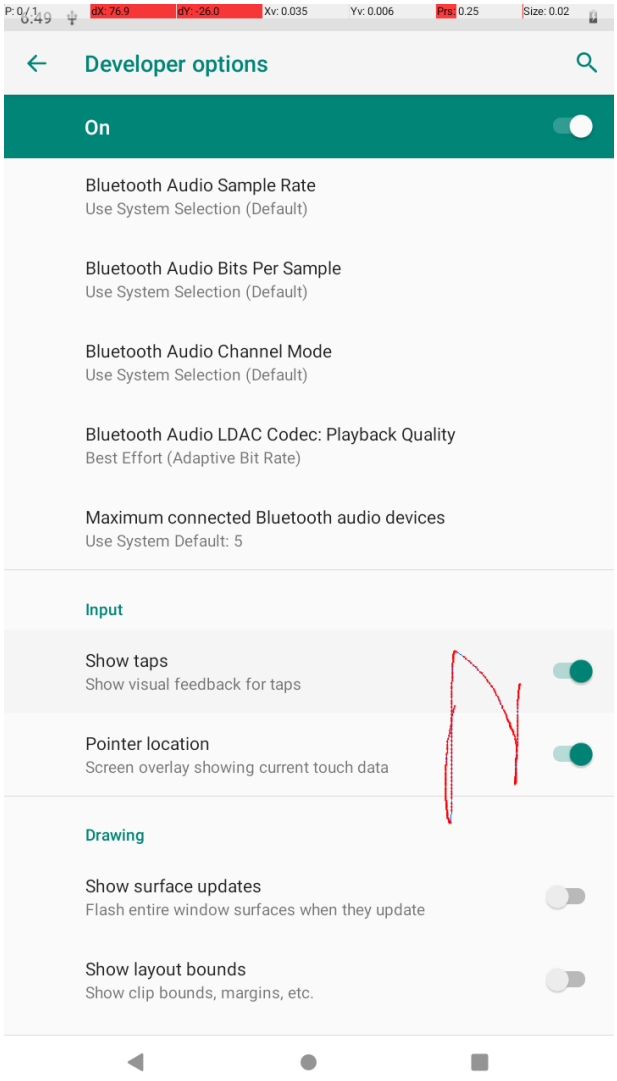




3.6 Touch Screen

MIPI-DSI screen support multi-touch function, tap the screen to operate the system.

- In Developer options screen, enable “Show taps” to display taps when you touch the screen. A circle appears under your finger or stylus and follows you as you move around the screen.
- Enable “Pointer Location” to show the pointer (tap) location on the device with cross-hairs. A bar appears across the top of the screen to track the cross-hair coordinates. As you move the pointer, the coordinates in the bar track the cross-hair location and the pointer path draws on the screen.



3.7 Audio

3.7.1 Audio Devices

MaaXBoard support HDMI Audio, USB Audio and Bluetooth audio.

3.7.1.1 HDMI AUDIO

Choose HDMI screen as displayer, connect HDMI displayer and the audio devices, then the system media audio output from HDMI audio. Connect the audio output device, you can play audio from HDMI device.

3.7.1.2 USB AUDIO DEVICE

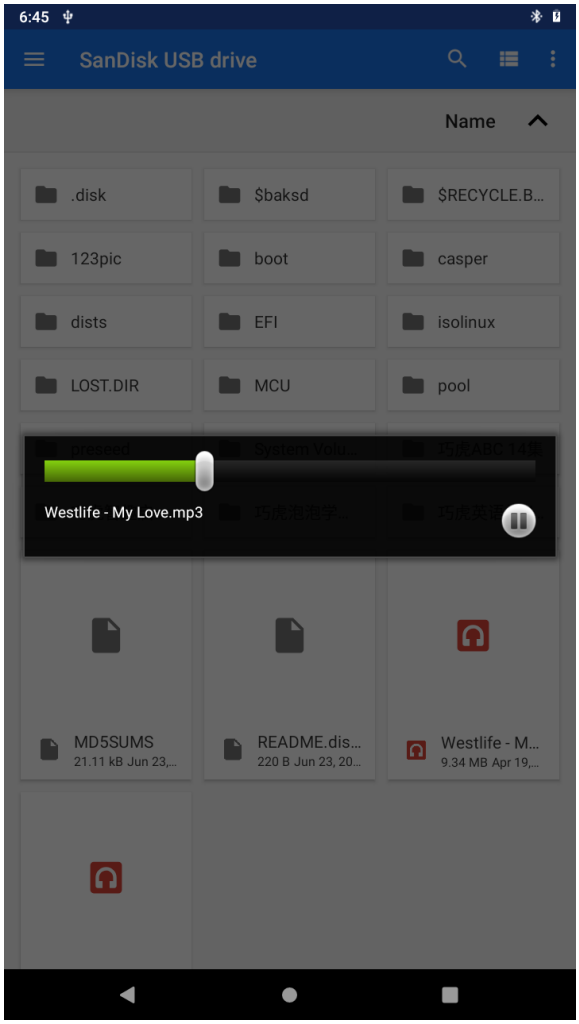
MaaXBoard could support USB audio device (which do not need specified driver) to play audio. Connect USB audio device to USB 1, you can play audio from USB audio device.

3.7.1.3 BLUETOOTH AUDIO

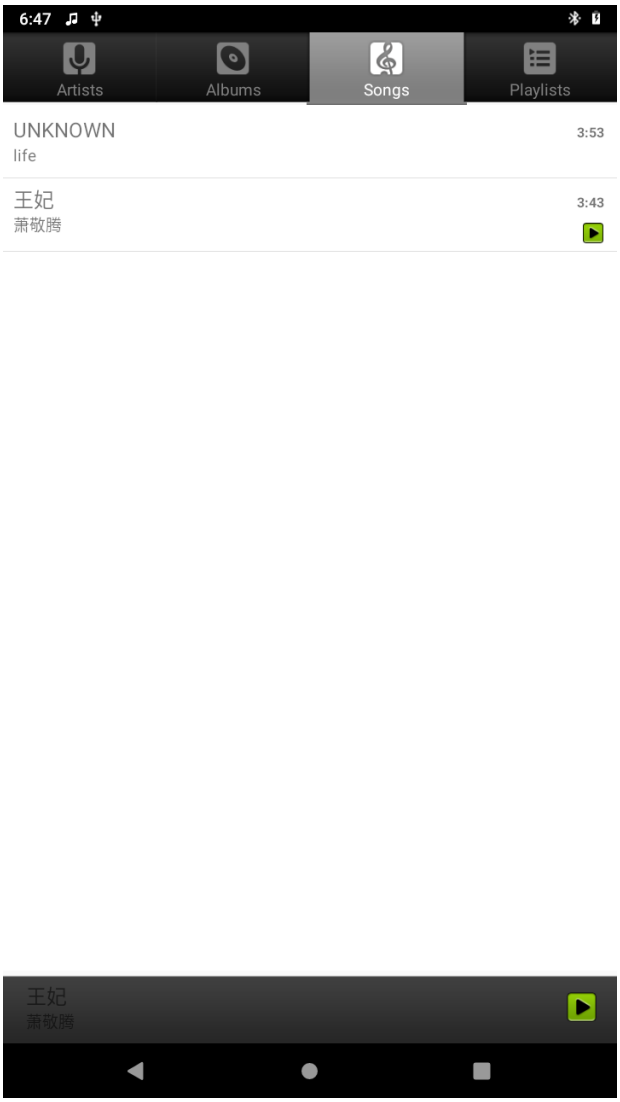
MaaXBoard also support play audio files via the Bluetooth audio device such as Bluetooth headset and speaker. To connect the Bluetooth device, refer to Bluetooth 5.0.

3.7.2 Play Audio

Copy wav or mp3 files to a U-disk, connect it to USB1, then open “Settings” -> “Storage” -> “SanDisk USB drive”, double click the audio file to play it.

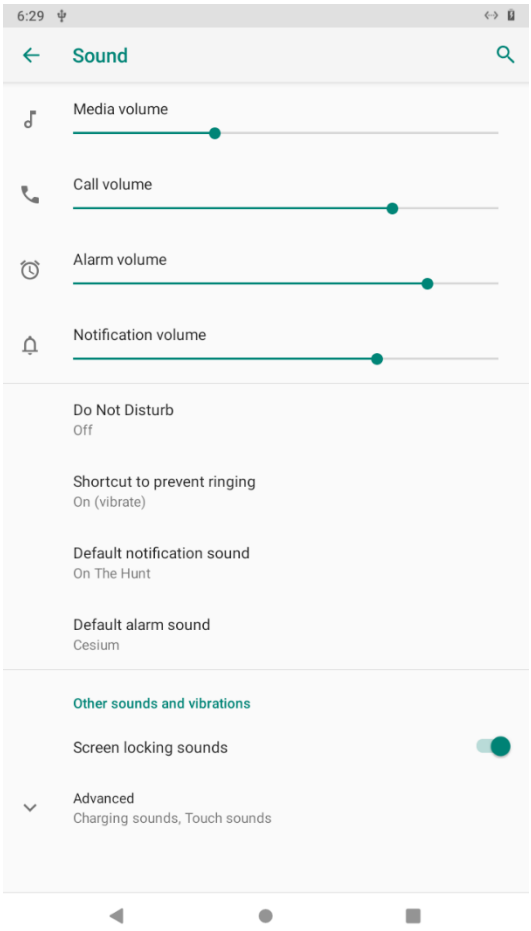


You can also copy the audio file to the Music folder of Android internal storage, open “Music” application to play the music.



3.7.3 Volume Adjustment

Go to “Settings” -> “Sound”, drag the volume bar to adjust it.



3.8 Camera

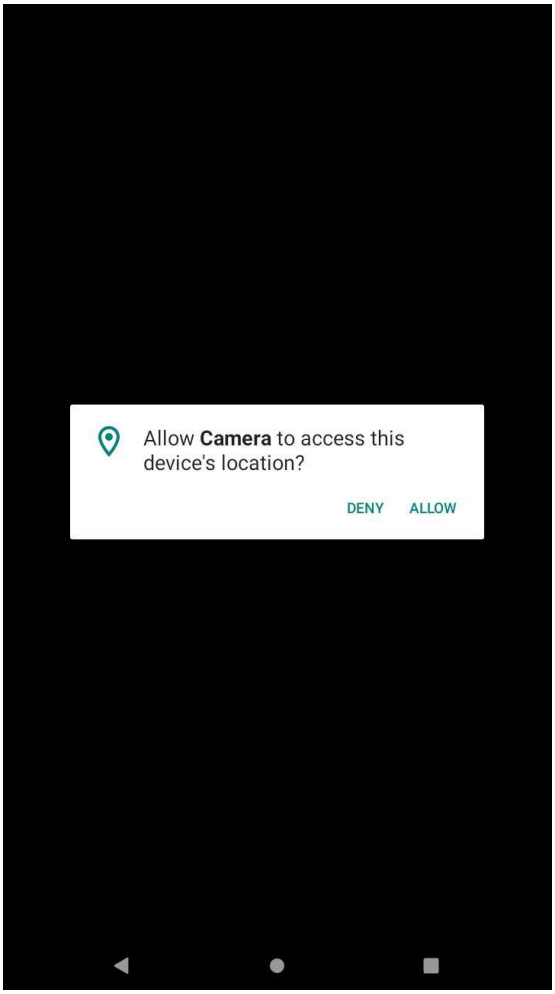
MaaXBoard support USB Camera and MIPI-CSI Camera.

3.8.1 MIPI-Camera

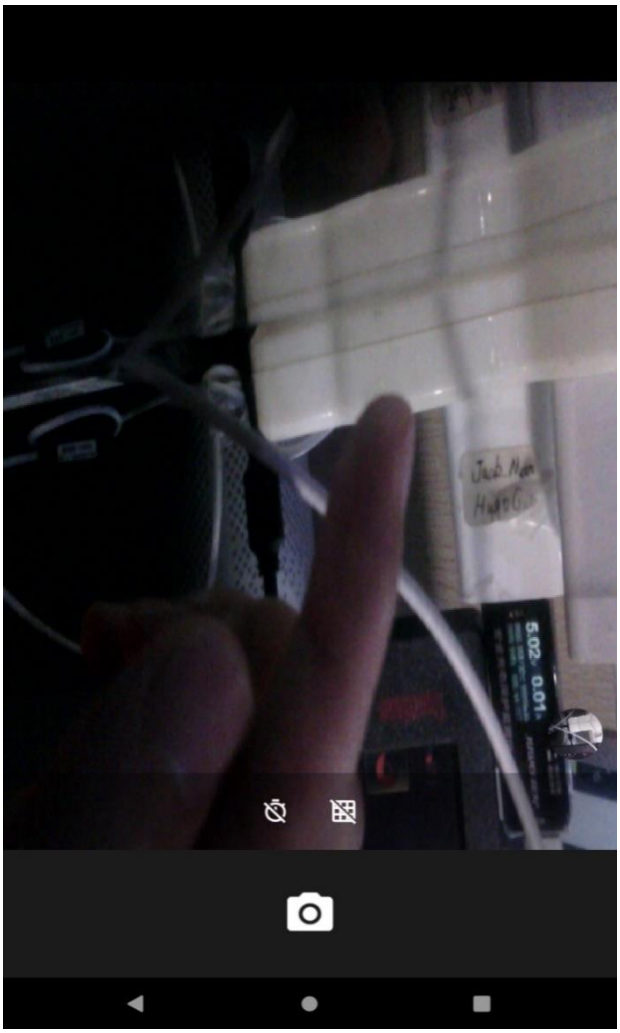
Connect the Camera module to J16 before system boot up. The Android system provide a Camera application, which could be used with MIPI Camera to catch photos.

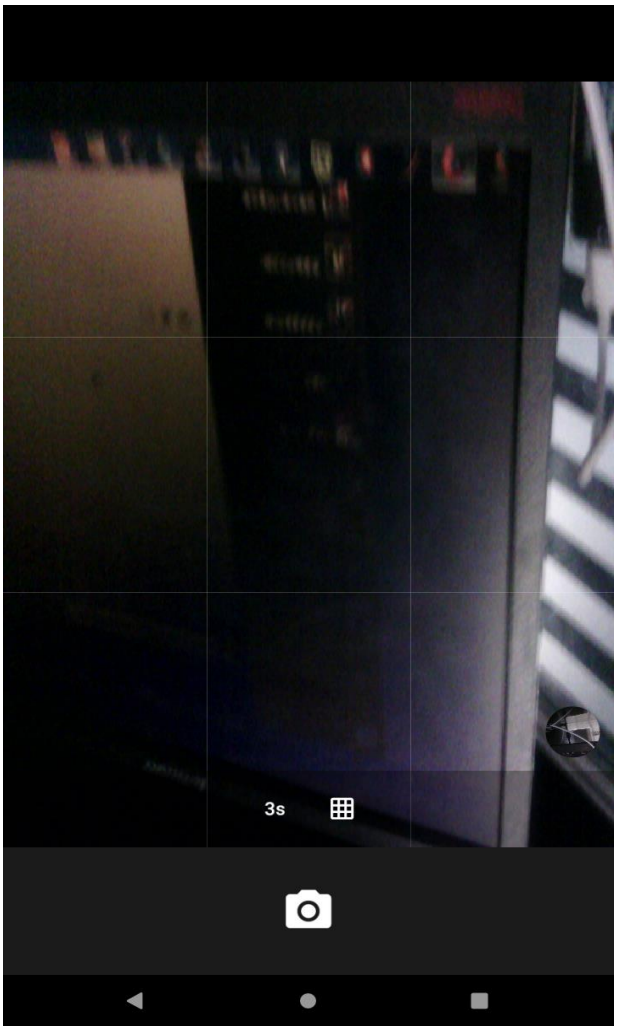
3.8.1.1 PREVIEW AND CATCH PHOTOS

1. Open the Camera APP, choose “Allow” in the Pop-up windows, then come into the UI of preview.
2. Click catch key to catch a photo.



3. Click the option button in the lower right corner to use the delay-catch mode and show gridlines in preview pictures.





4. The path of the photos: /sdcard/DCIM/Camera

3.8.1.2 RECORD VIDEO（TBD）

In current version, the Audio input device is not supported yet, so MIPI camera could not record video.

3.8.2 USB Camera

Connect USB Camera to USB1 after system start up.

If you connect the MIPI Camera and USB Camera to one board, open the default Camera APP, the USB Camera is used as Secondary Camera. You could switch to the USB Camera in the “Camera” APP.

As the default Camera APP of Android system could not support USB camera independently, we suggest to use another USB Camera APP, you should install it at the first. The reference link as below:

Refer link: http://app.mi.com/details?id=com.shenyaocn.android.usbcamera

Open that APP, it could preview, catch a photo and record video.

3.9 ETH

Connect the network cable to J13, after connecting the network cable, MaaXBoard will automatically obtain the IP by default. You can use the **ifconfig** command to view the IP information and use the following command to perform the network test:

```
em_sbc_imx8m:/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 36:88:ab:d8:f7:8d  Driver fec
          inet addr:192.168.2.99  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::4443:d4ea:45c9:22c6/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:221 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23360 TX bytes:4458
em_sbc_imx8m:/ # ping www.bing.com
PING cn-0001.cn-msedge.net (202.89.233.100) 56(84) bytes of data.
64 bytes from 202.89.233.100: icmp_seq=1 ttl=118 time=22.0 ms
64 bytes from 202.89.233.100: icmp_seq=2 ttl=118 time=23.7 ms
^C
--- cn-0001.cn-msedge.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 22.075/22.922/23.769/0.847 ms
```

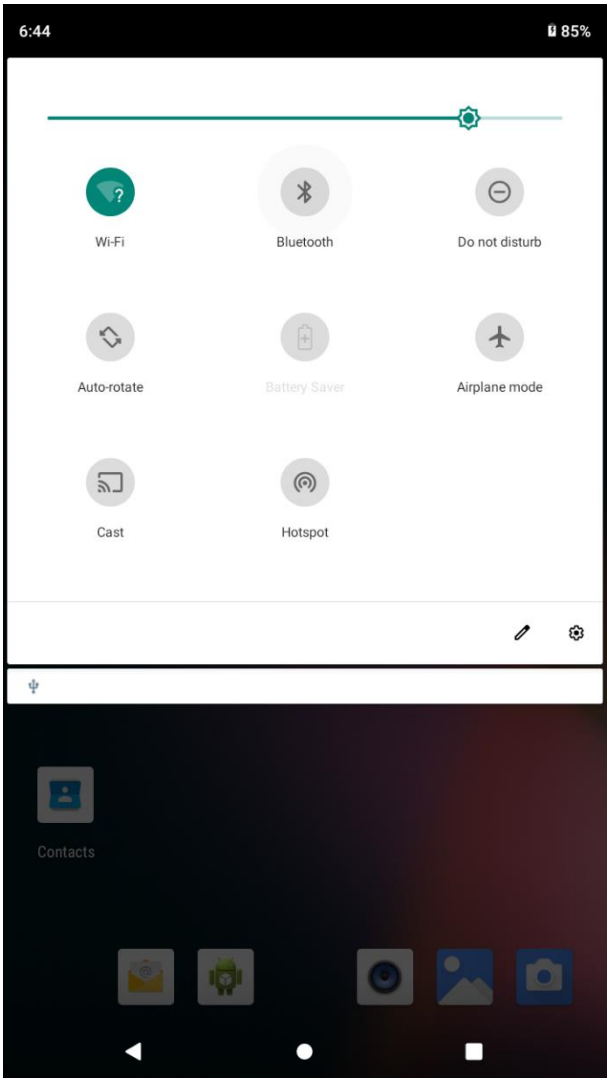
You could also use the “WebView Browser Tester” APP to browse website. Input the address in the address bar.

3.10 Wi-Fi

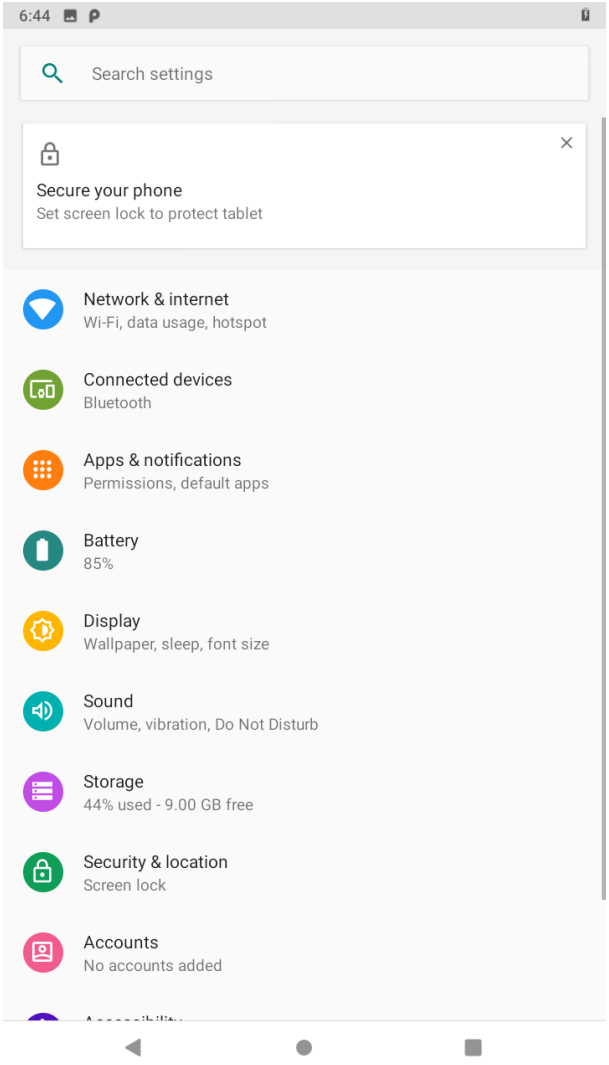
The on-board Wi-Fi module support 2.4G/5G network and hotspot.

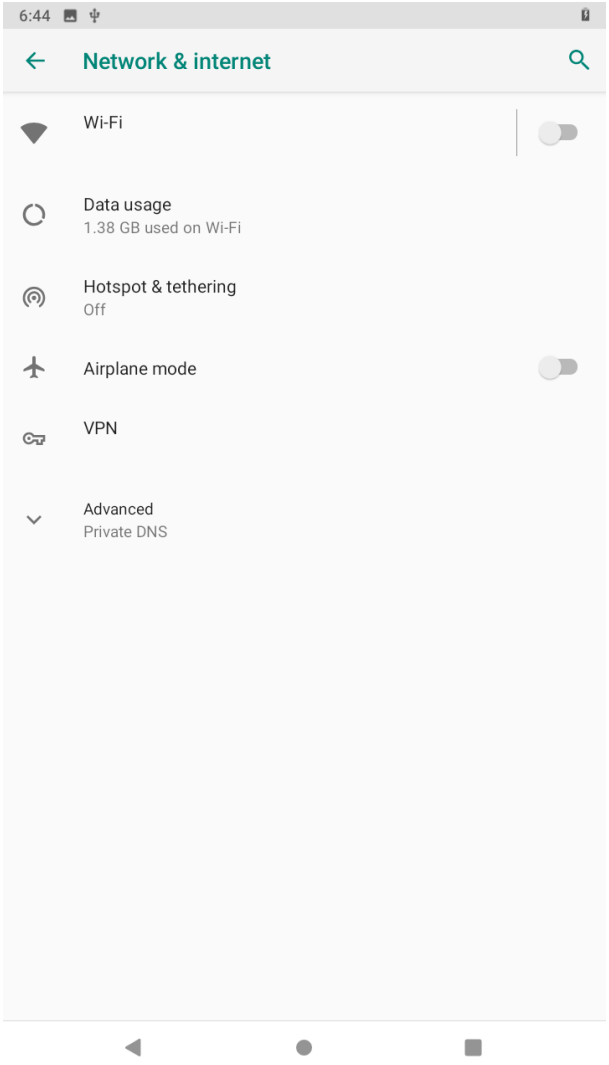
3.10.1 Enable and Disable Wi-Fi

There are 2 kinds of way to enable and disable the Wi-Fi, the first one is to click the Wi-Fi icon in the system drop-down box



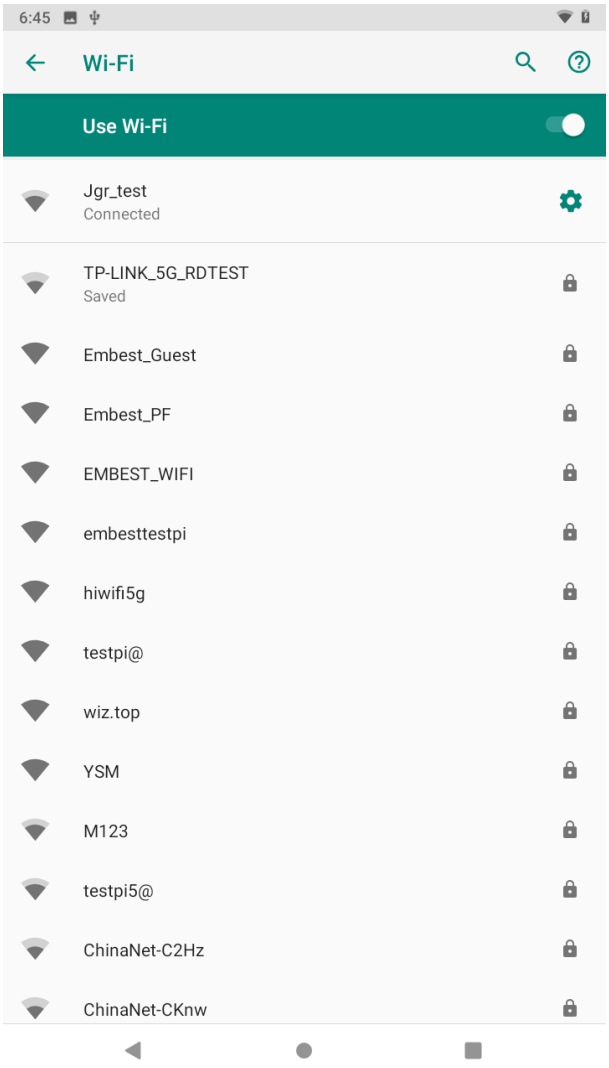
You also could open the Wi-Fi at Settings APP, steps as below:

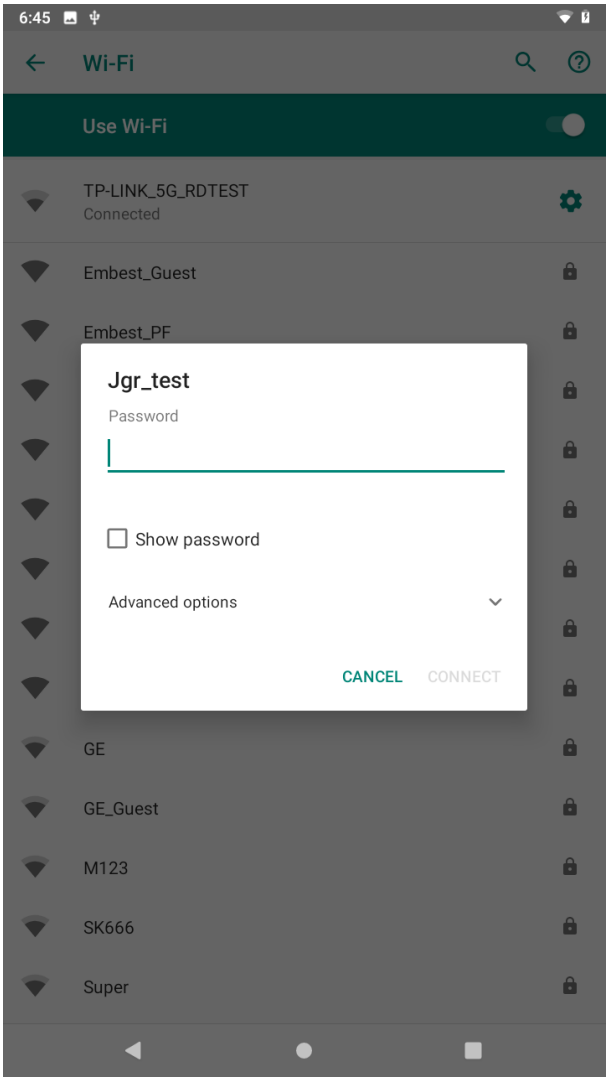




3.10.2 Connect the Wi-Fi Network

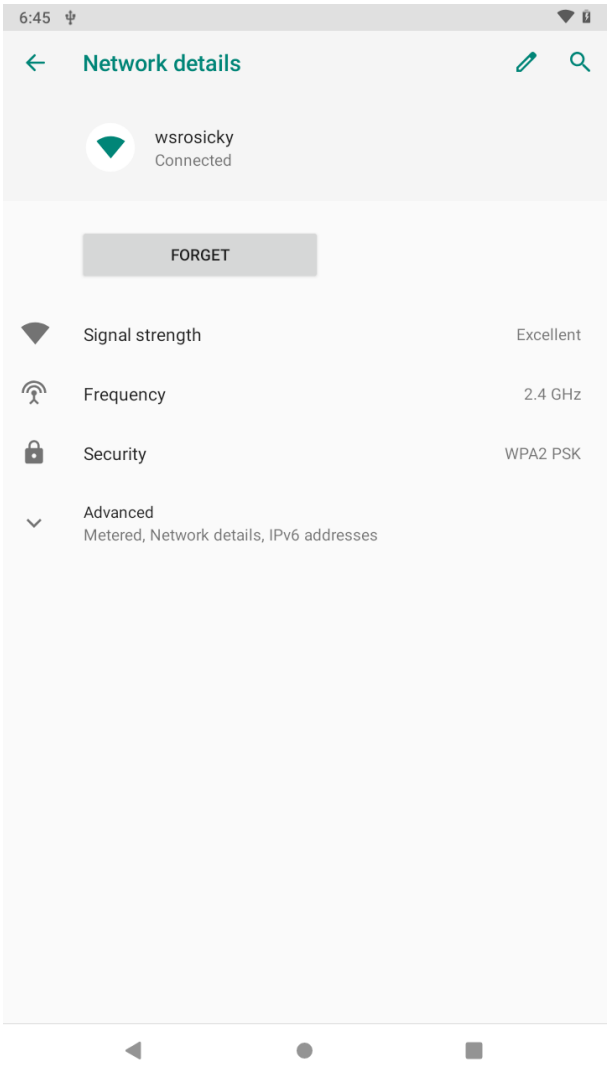
Click the Wi-Fi, enable “Use Wi-Fi”, it could auto scan the available Wi-Fi networks, click the network you want to connect and input the password, then you could using the Wi-Fi network.





3.10.3 Delete the network

Click the network details, and click the FORGET key.

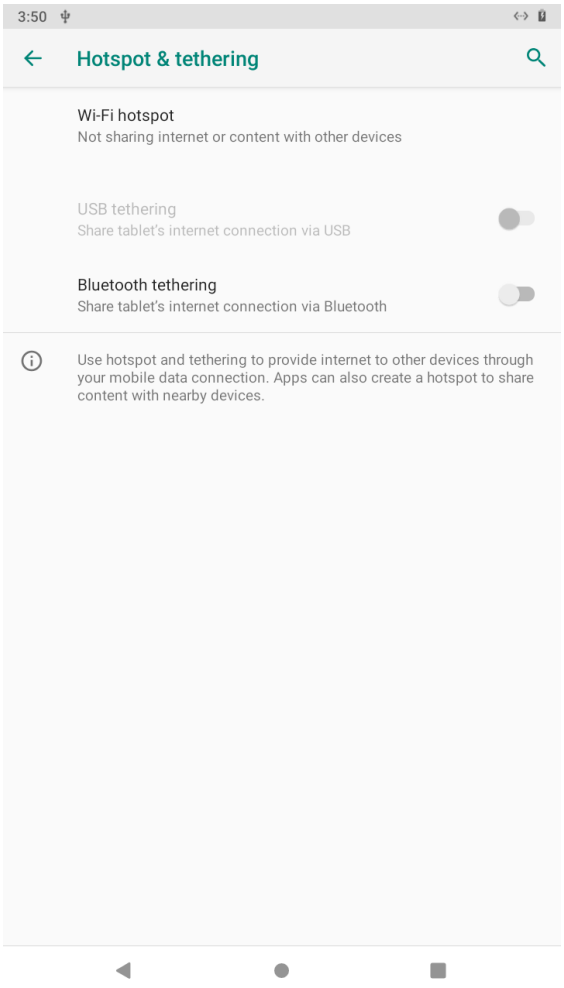


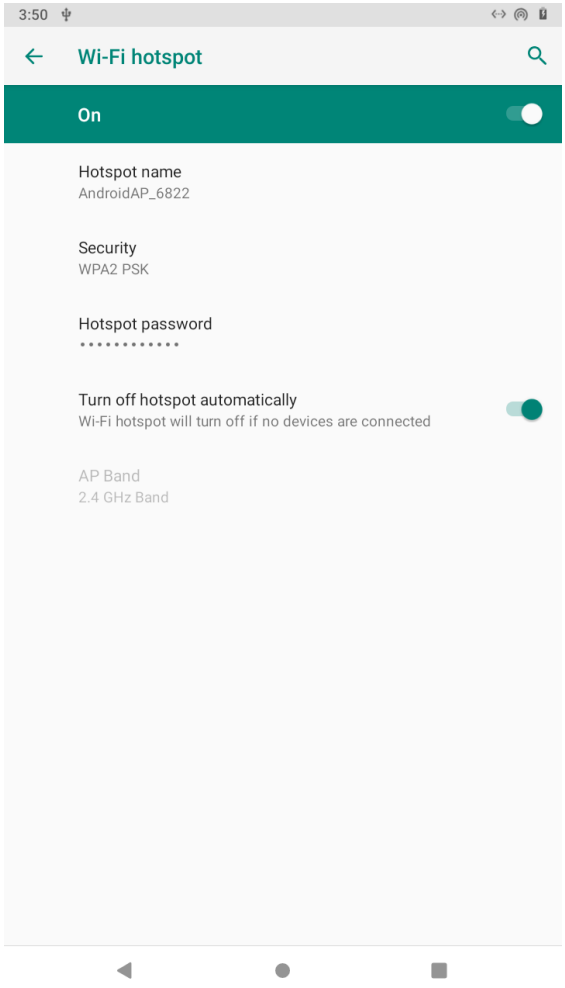
3.10.4 Wi-Fi Hotspot

Before using the Hotspot, you must disconnect the Wi-Fi, and connect the cable to the Ethernet interface. Then start the Settings APP and click into the “Network & internet” to start the configuration of Wi-Fi Hotspot

1. Enable the Hotspot

Click the Hotspot & tethering into the Hotspot configuration. Then click into the Wi-Fi hotspot and slide the enable key to “On”, the Wi-Fi hotspot is enabled.





2. Disable the Hotspot

Slide the enable key to “Off”, and the Wi-Fi Hotspot would be disable。

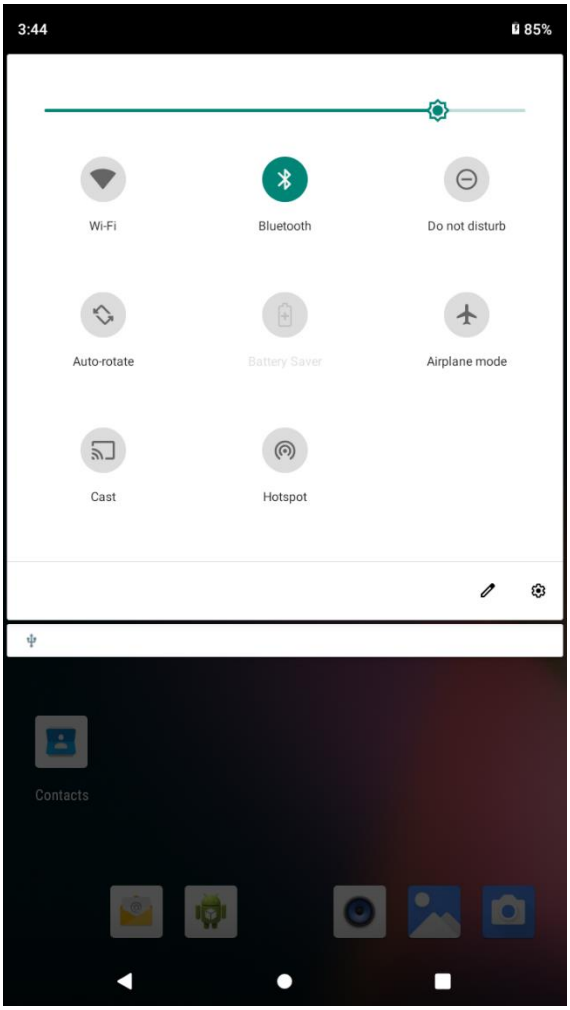
3. The configuration parameter

Setup the Hostname，Security and Hotspot Password as SSID，Security Mode and Password in proper sequence。

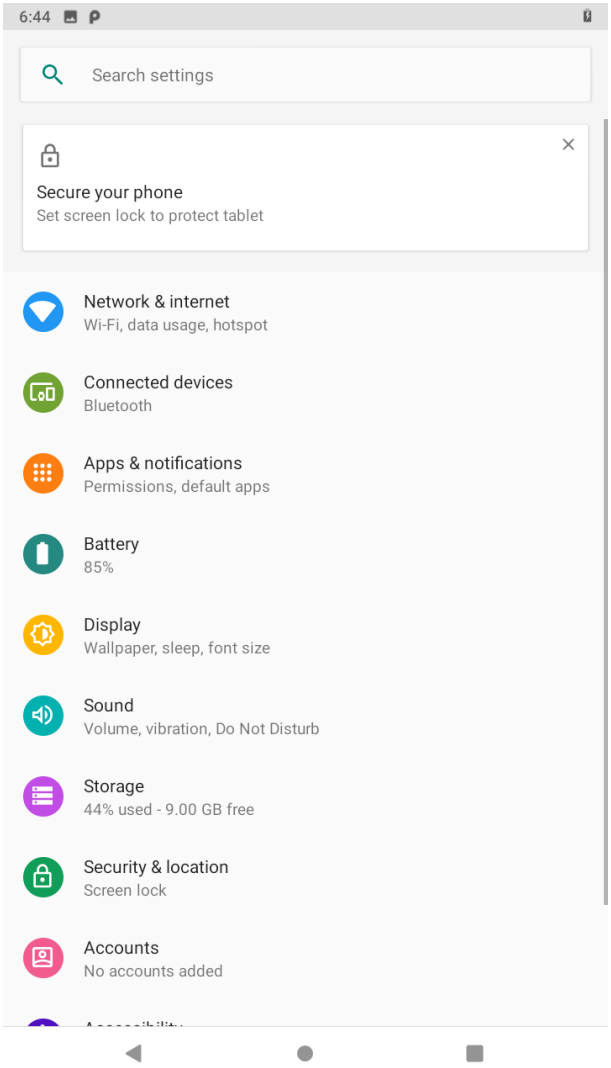
3.11 Bluetooth 4.2

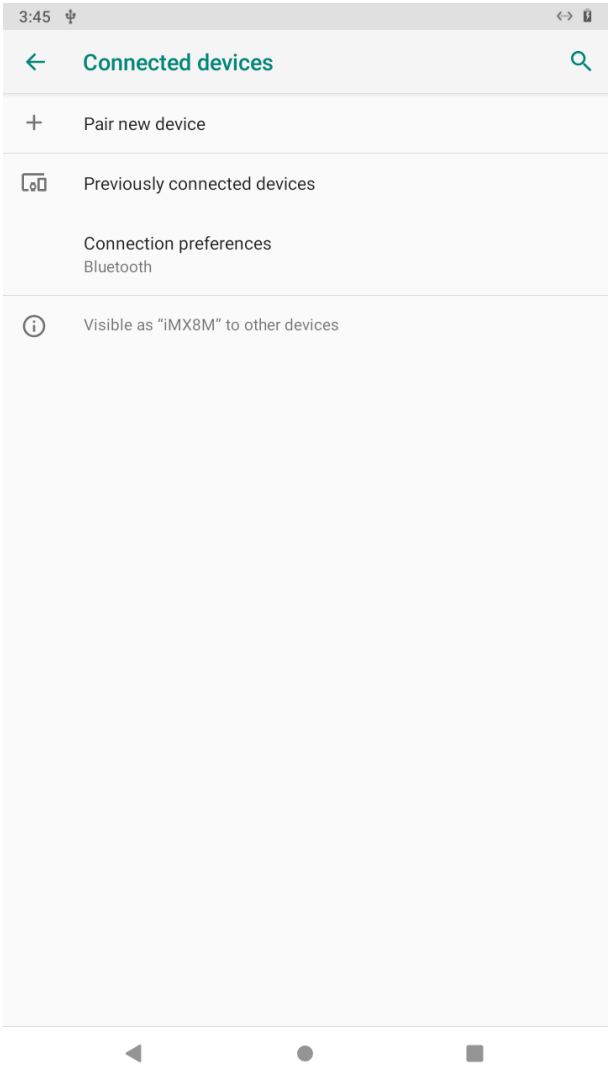
3.11.1 Start Bluetooth

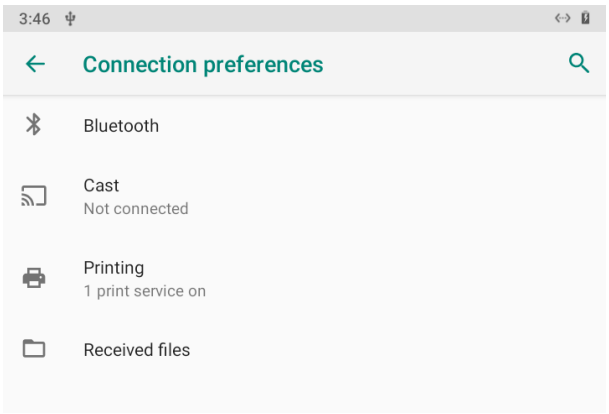
There is 2 ways to open the Bluetooth, first way is the Bluetooth icon of the system drop-down box

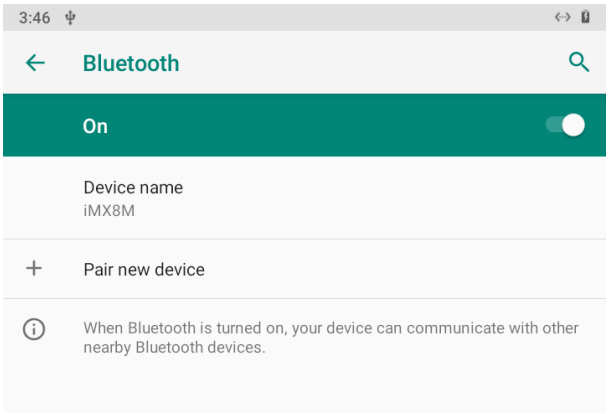


1. Another way is using Settings APP, start the APP , go to the Connected Device , Connection preferences in proper sequence and slide the key to “ON”, then the Bluetooth is enabled.



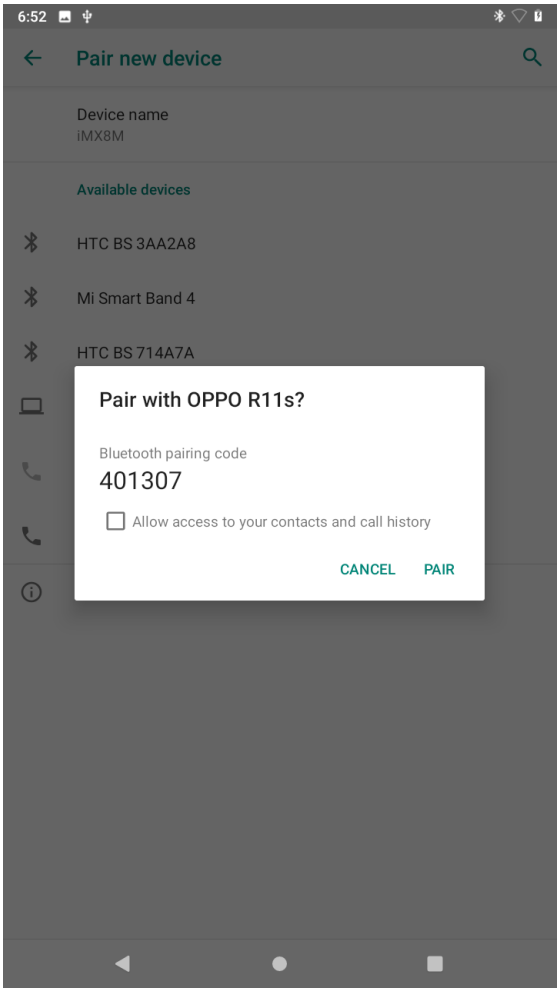






3.11.2 Scan and Connect the Bluetooth Devices

Click the Pair new device, system could scan the Bluetooth device, click the device you want connect in the available devices list. The board support the Bluetooth earphone, sound box and mobile phone



3.12 LED

User can control the 2 single color LED indicators, LED0 and LED1 (corresponding to usr_led and sys_led) on MaaXBoard Board. Execute the following instructions in terminal to control them. Users could also write their own applications to control the LED.

Light out LED:

```
em_sbc_imx8m:/ # echo 0 | tee /sys/class/leds/usr_led/brightness
em_sbc_imx8m:/ # echo 0 | tee /sys/class/leds/sys_led/brightness
```

Light up LED:

```
em_sbc_imx8m:/ # echo 1 | tee /sys/class/leds/usr_led/brightness
em_sbc_imx8m:/ # echo 1 | tee /sys/class/leds/sys_led/brightness
```

3.13 UART

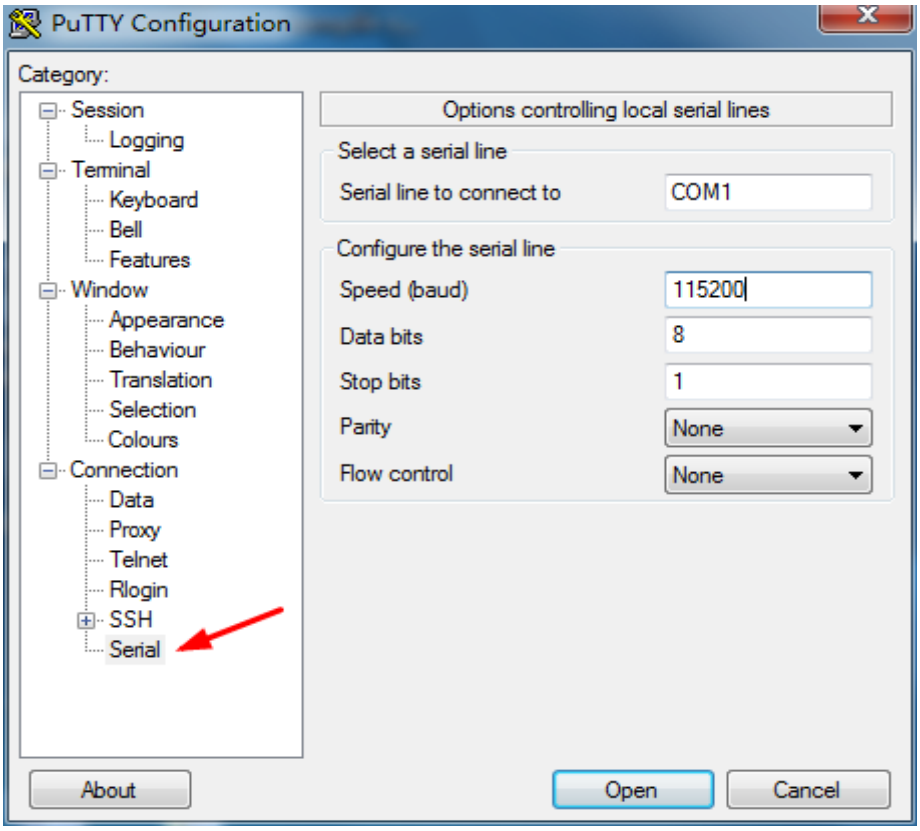
MaaXBoard supports 2 UART interface.

MaaXBoard (CPU)	Interface Type
UART1	UART TTL (Debug Interface)
UART2	UART TTL

3.13.1 UART1

UART1 is the default debug interface for Android OS, which can be used to login the shell environment of MaaXBoard. Connection methods:

1. Connect the debug interface to PC with USB to TTL converter. Pin 6, 8 and 10 of J10 to the GND, RXD and TXD pin of the USB to TTL converter.
2. Install the Serial Communication software (e.g. PUTTY), select the corresponding port number, baudrate as 115200, data bits as 8, stop bits as 1, parity as none.



3. In default, you are login with ordinary account permission, if you need permission for root, type su command:

```
em_sbc_imx8m:/ $ ls sdcard
ls: sdcard: Permission denied
1|em_sbc_imx8m:/ $ su
em_sbc_imx8m:/ # ls sdcard
Alarms    DCIM        Movies Notifications Podcasts
Android Download Music    Pictures         Ringtones
```

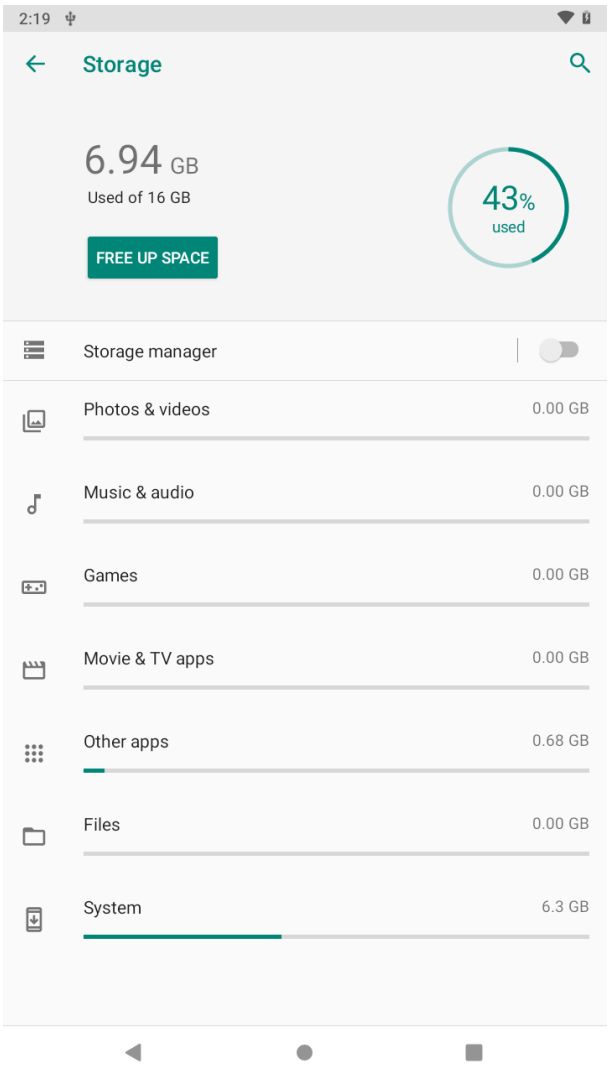
3.13.2 UART2

In the Android system, the node for UART2 is /dev/ttymxc1. Users could also write their own applications to control the LED.

3.14 Storage

The system image size is 16GB, to query storage amount, go to “Settings” -> “Storage”.

In this screen, you can also “FREE UP SPACE” or use “Storage manger” to manage storage.



If you need to write or read files from the storage, use Files app.

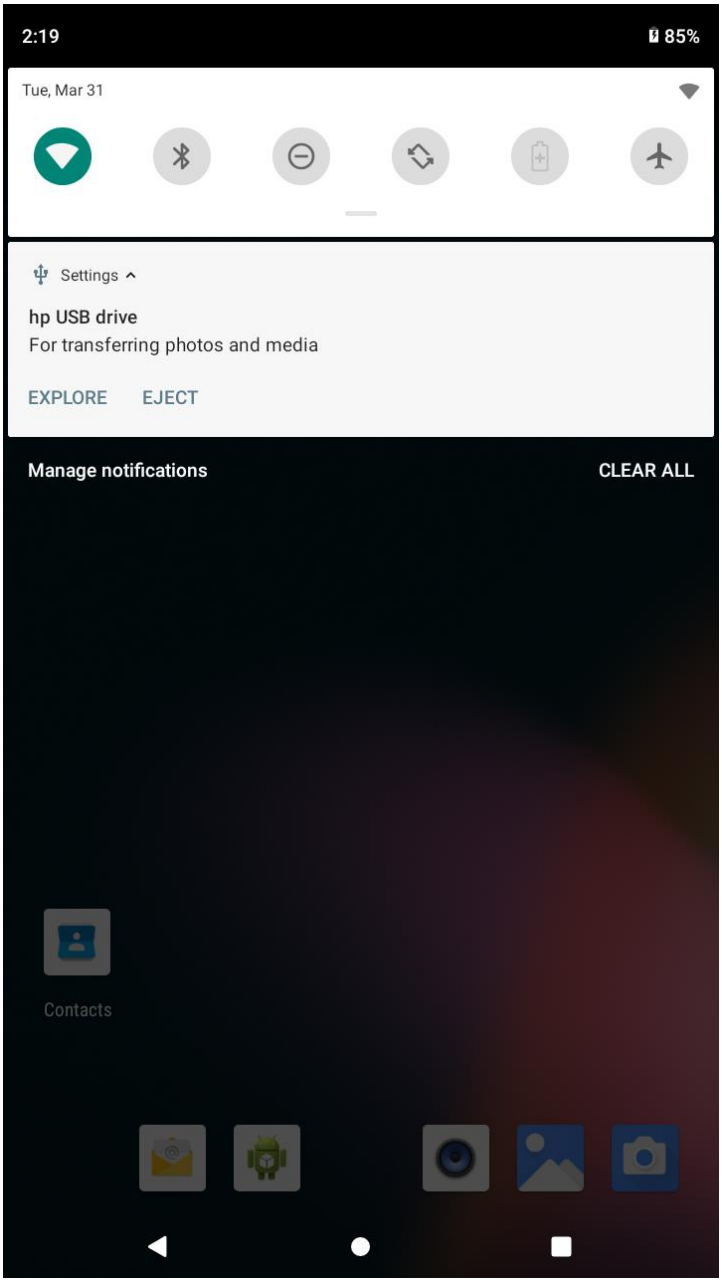
3.15 USB 3.0 Interface

The USB 3.0 interface J5 on have 2 USB Host Interface, the upper one is USB1, the lower one is USB0. USB1 support USB HOST function, USB0 support USB Device function.

3.15.1 USB Host

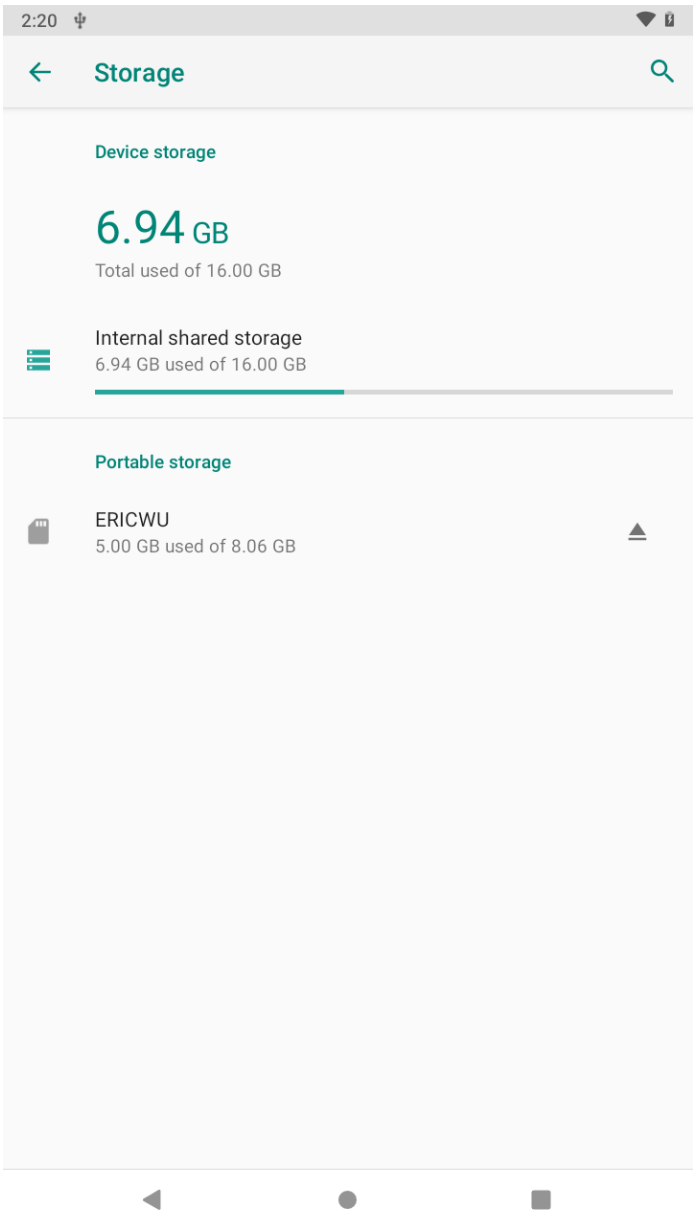
USB1 supports USB device such as USB HUB, U disk, USB Camera, key board, mouse, etc.

Insert a U-disk to USB1, system will mount the storage device automatically. In drop-down list, it will display as follows:



Users could open U disk to explore the files or eject the U-disk.

In “Settings” -> “Storage” screen, you will find portable storage



In this screen, click device name to open the U disk, eject or mount the device.

3.15.2 USB Device

USB0 support USB Device function could be used to burn the system image, USB debug or transfer files. For USB debug guide, refer to USB Debug.

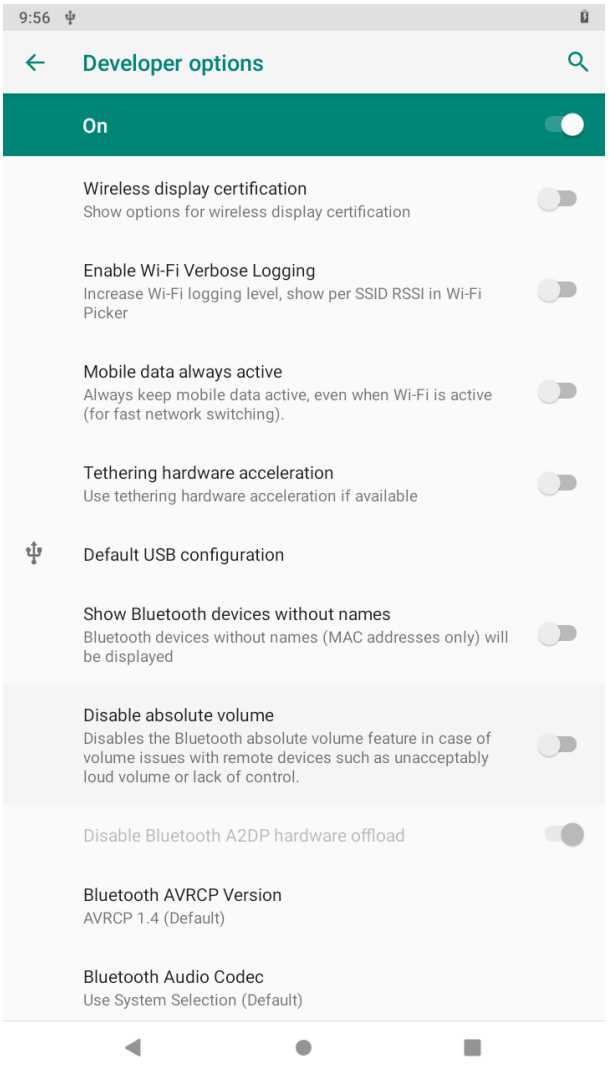
Note: Connect USB0 and PC after the system start up.

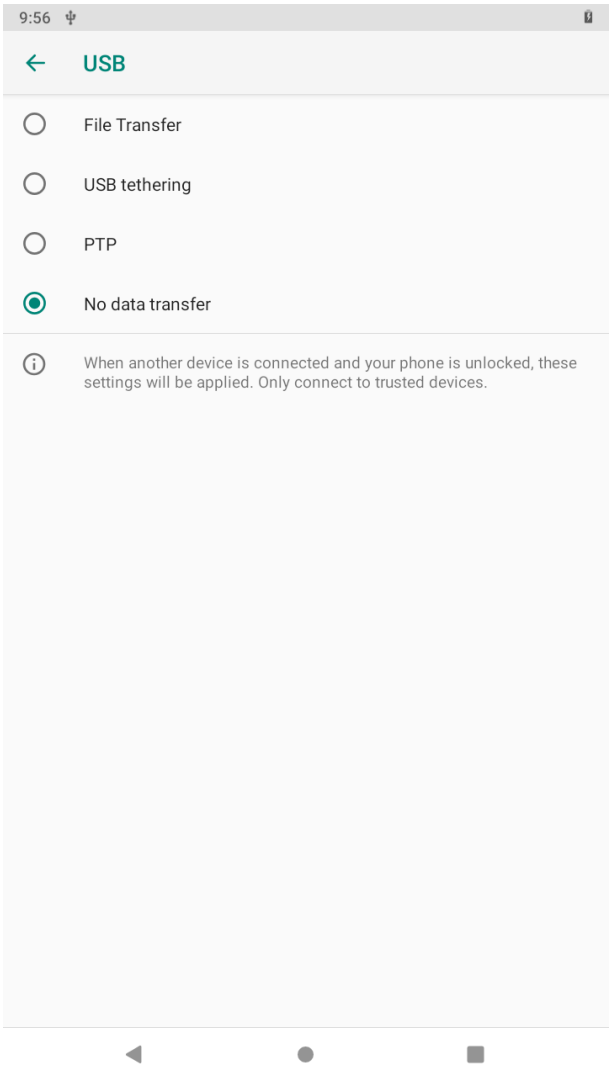
3.15.2.1 BURNING MODE

Connect USB0 and PC before power on the board. The system will not boot normally, it will enter burning mode. Then users could burn the system image to the development board using uuu tools. For the detail information, refer to MaaXBoard EMMC burning Guide.

3.15.2.2 TRANSFER FILES

Start up the system, then connect USB0 and PC, open default USB configuration option in Developer options.

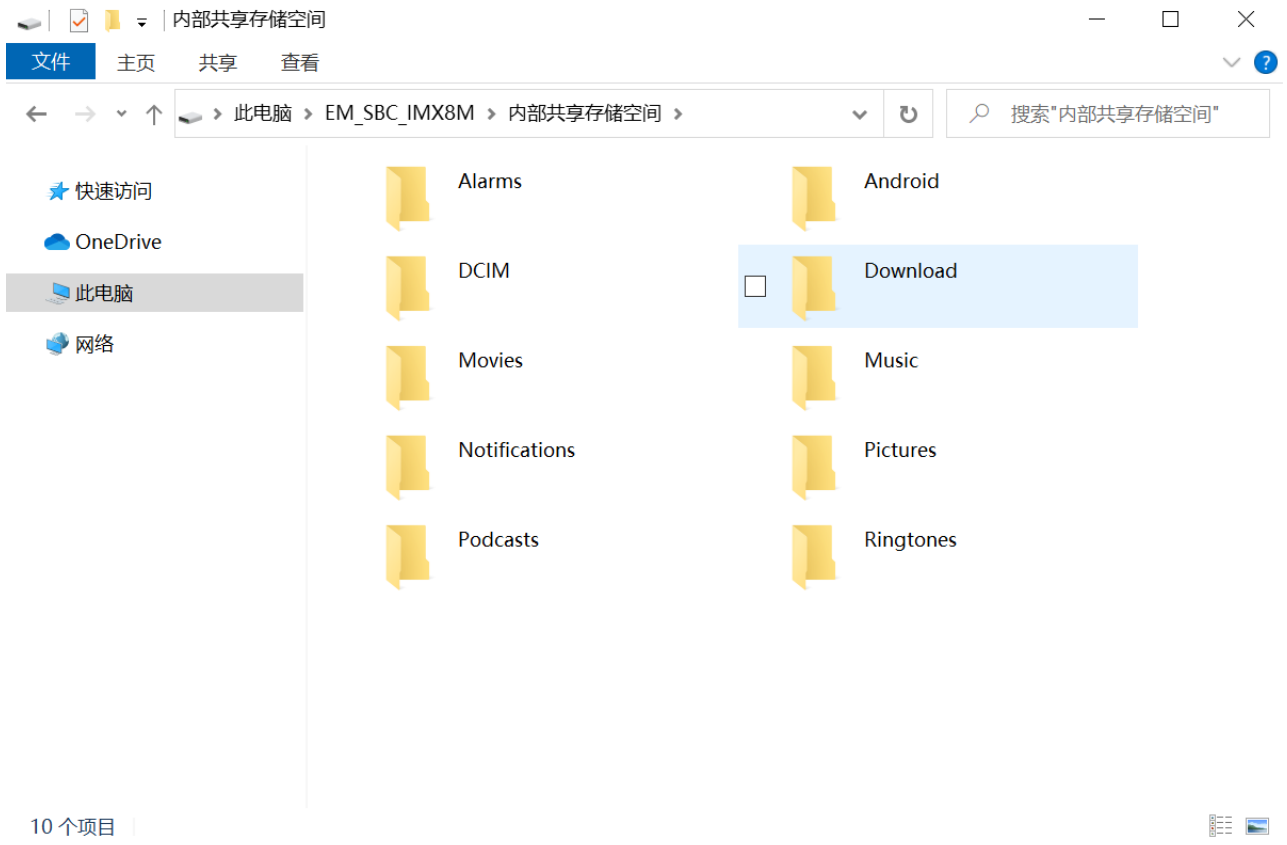




1. Choose “File Transfer”

The device manager on the computer will show portable device: EM_SBC_IMX8M, open computer, you will find EM_SBC_IMX8M.

Open EM_SBC_IMX8M, users could edit all the files / folders under /sdcard of the board.



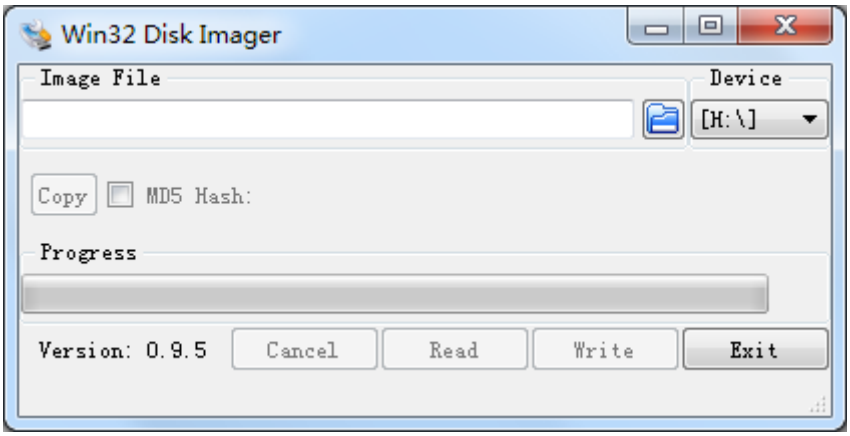
2. Choose “PTP”

EM_SBC_IMX8M will be listed in device manager and computer. Users could visit /sdcard/Pictures and /sdcard/DCIM on the board, edit the files / folders.

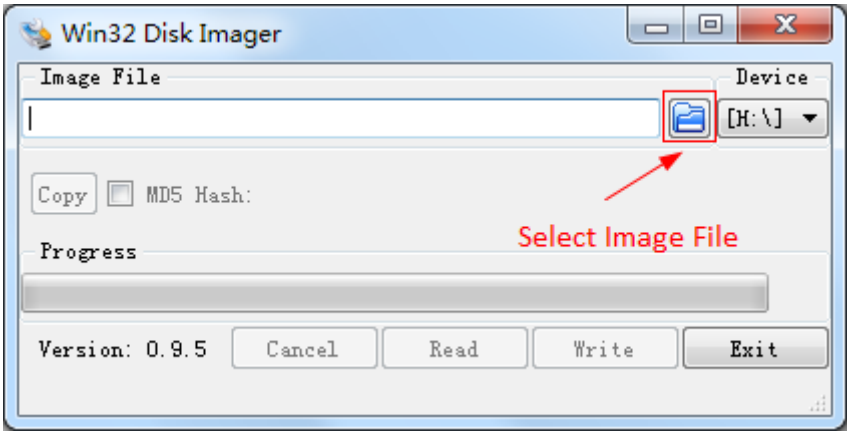
Chapter 4 Burn or update the system Image

4.1 Burn the System Image to SD Card under Windows OS

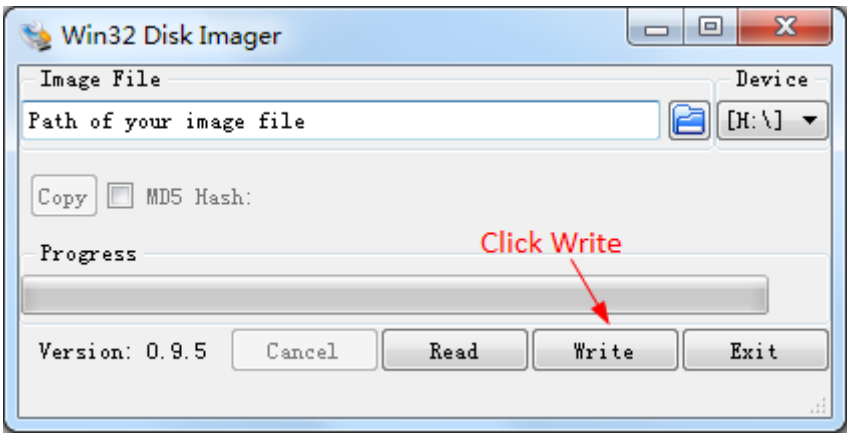
1. Firstly, you should prepare a SD card, which is no less than 16GB.
2. Then, download and install “Win32 Disk Imager” from: https://sourceforge.net/projects/win32diskimager/.



3. Select the system images file: eg: MaaXBoard-AndroidShipmentImage-SDcard-MIPI-V1.0.2r03.img



4. Click “Write” button to burn the images:



4.2 Burn the System Image to SD Card under Android OS

In Ubuntu or Debian OS, you can use bmap-tool to burn the image to SD Card. Here we use MaaXBoard-AndroidShipmentImage-SDcard-MIPI-V1.0.2r03.img as an example:

1. Install bmap-tools

```
$ sudo apt install bmap-tools
```

2. Enter the following instructions in command line to check the SD Card ID, in this example is: sdc

```
$ ls /dev/sd*
/dev/sda    /dev/sda2   /dev/sdb    /dev/sdb2   /dev/sdc    /dev/sdc2
/dev/sda1   /dev/sda5   /dev/sdb1   /dev/sdb5   /dev/sdc1
```

3. If SD Card is mounted, umount it.

```
$ sudo umount /dev/sdc1
$ sudo umount /dev/sdc2
```

4. Burn the SD card with following instructions:

```
$ bmaptool create -o burn.map MaaXBoard-AndroidShipmentImage-SDcard-MIPI-V1.0.2r03.img
$ sudo bmaptool copy --bmap burn.map
MaaXBoard-AndroidShipmentImage-SDcard-MIPI-V1.0.2r03.img /dev/sdc
```

4.3 Update System Image in eMMC

USB0 (The lower one in USB interface HUB1) support burning mode. Connect USB0 and PC before power on the board. The system will enter burning mode. Then users could burn the system image to the development board using uuu tools. For the detail information, refer to MaaXBoard EMMC burning Guide.

Chapter 5 Appendix

5.1 Hardware

For the detail hardware introduction, please refer to MaaXBoard Hardware User Manual.

5.2 Software

MaaXBoard support Linux Debian system and Android system, for the detail software introduction, please refer to related user manual.

- **Linux**
  - MaaXBoard Linux Software Release Note
  - MaaXBoard Linux Software User Manual
  - MaaXBoard Linux Software Development Guide
- **Android**
  - MaaXBoard Android Software Release Note
  - MaaXBoard Android Software User Manual
  - MaaXBoard Android Software Development Guide

5.3 Android Develop

- https://android.googlesource.com/
- https://developer.android.com

Chapter 6 Technical Support

6.1 Technical Support

- To receive technical support, please post any questions you may have to the forum on https://www.avnet.me/maaxboard or contact your local Avnet FAE.