

**Android Pie User Manual**

**VER. 2.20**

**December 20, 2019**

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Originator** | **Notes** |
| 1.00 | March 9, 2019 | TechNexion | First public release |
| 1.10 | March 30, 2019 | TechNexion | Add pico-imx8mm |
| 2.00 | July 30, 2019 | TechNexion | Improve system |
| 2.01 | August 7, 2019 | TechNexion | Add uuu install |
| 2.10 | September 20, 2019 | TechNexion | Add 32-bit platforms |
| 2.20 | December 20, 2019 | TechNexion | Add new devices and new features |

Table of Contents

[1. Introduction 4](#_Toc26447368)

[1.1 About Android Pie 4](#_Toc26447369)

[1.2 How to read this document 4](#_Toc26447370)

[2. Supported Hardware 5](#_Toc26447371)

[3. Software Configuration 6](#_Toc26447372)

[3-1 Software Revision 6](#_Toc26447373)

[3-2 Memory Layout of the Android 9 Image 6](#_Toc26447374)

[3.3 Image Installation 8](#_Toc26447375)

[3.4 Serial Debug 8](#_Toc26447376)

[3.5 Network 8](#_Toc26447377)

[3.6 Expansion Pin Header 10](#_Toc26447378)

[3.7 Termux app 11](#_Toc26447379)

[3.8 Audio HAT 13](#_Toc26447380)

[3.9 Serial Link 14](#_Toc26447381)

[3.10 Android Treble 16](#_Toc26447382)

[3.11 Libgpiod JNI Test APP 18](#_Toc26447383)

[4. Software Development and Upgrade 19](#_Toc26447384)

[5. Q & A 20](#_Toc26447385)

# 1. Introduction

## 1.1 About Android Pie

Android Pie (9.0) is the latest Android distribution from 2018 ends, keep improve user friendly and performance.

Customized features:

|  |  |  |  |
| --- | --- | --- | --- |
| Industry | Networking | OS Features | Multimedia |
| UART Serial Test APP | ETHTOOL runtime libraries | OTA upgrade | Dual Camera |
| LIBGPIOD Test APP | SSH runtime libraries | A/B system | Dual Display |
| CANBUS Test APP | Bluetooth-LE | Fastboot | Voice-HAT |
| I2C runtime libraries | BT A2DP audio source | Termux app | Software Web browser (IMX7) |

## 1.2 How to read this document

Chapter 2 is about this distribution.

Chapter 3 is about the software functions and features.

Chapter 4 is about how to develop your own android 9 on Technexion hardware boards.

Chapter 5 we will list common questions.

# 2. Supported Hardware

Following Android Pie distribution is provided for Technexion platforms:

* IMX8 Series
  + PICO-IMX8M
  + PICO-IMX8MM
  + FLEX-IMX8MM
  + EDM-IMX8M
* IMX6 Series
  + PICO-IMX6
  + EDM-IMX6
  + TC0700
  + TC1000
  + TEP-1010
  + TEP-1560
* IMX7 Series
  + PICO-IMX7
  + TEP1-IMX7

Each hardware has its own different functions and features, please refer the specifications first from Technexion official website if possible.

# 3. Software Configuration

## 3-1 Software Revision

**IMX8 Series**

|  |  |
| --- | --- |
| **Name** | **Revision** |
| u-boot | 2018.03-g84c3460 |
| linux kernel | 4.14.98-gfcddbab |
| Android | 9.0.0-2.0.1\_8m-ga-tn2.0 |

**IMX6/IMX7 Series**

|  |  |
| --- | --- |
| **Name** | **Revision** |
| u-boot | 2018.03-g84c3460 |
| linux kernel | 4.14.98-gfcddbab |
| Android | 9.0.0-2.2.0-ga-tn2.0 |

## 3-2 Memory Layout of the Android 9 Image

For the boards use eMMC/SD as boot storage:

**IMX8 Series**

|  |  |
| --- | --- |
| **Section** | **Description** |
| GPT | Partition information |
| Bootloader | u-boot.imx: First stage u-boot image |
| Partition 1 dtbo\_a (backup partition) | dtbo.img |
| Partition 2 dtbo\_b | dtbo.img |
| Partition 3 (FAT32) boot\_a (backup partition) | boot.img:   * kernel image * recovery mode ramdisk |
| Partition 4 (FAT32) boot\_b | boot.img:   * Image: kernel image * ramdisk.img: recovery mode ramdisk |
| Partition 5 (EXT4) system\_a (backup partition) | system.img |
| Partition 6 (EXT4) system\_b | system.img |
| Partition 7 misc | For recovery store bootloader message, reserve |
| Partition 8 metadata | For system slide show |
| Partition 9 presistdata | Option to operate unlock\unlock |
| Partition 10 (EXT4) vendor\_a (backup partition) | vendor.img |
| Partition 11 (EXT4) vendor\_b | vendor.img |
| Partition 12 (EXT4) userdata | Application data storage for system application, and for internal media partition, in /mnt/sdcard/dir |
| Partition 13 (EXT4) fbmisc | For storing the state of lock or unlock |
| Partition 12 vbmeta\_a (backup partition) | For storing the verify boot's metadata |
| Partition 13 vbmeta\_b | For storing the verify boot's metadata |

**IMX6/IMX7 Series**

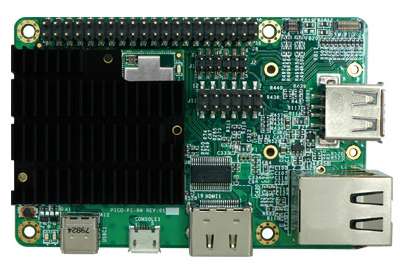
|  |  |
| --- | --- |
| **Section** | **Description** |
| GPT | Partition information |
| Bootloader | SPL: First stage u-boot image, u-boot.img: Second stage image |
| Partition 1 dtbo (backup partition) | dtbo.img |
| Partition 2 (FAT32)  boot | boot.img:   * kernel image * ramdisk |
| Partition 3 (FAT32) recovery | boot.img:   * kernel image * ramdisk of recovery mode |
| Partition 4 (EXT4) system | system.img |
| Partition 5 cache | Android cache for image store  of OTA |
| Partition 6 misc | For recovery storage  bootloader message, reserve |
| Partition 7 datafooter | For crypto footer of DATA  partition encryption |
| Partition 8 metadata | For system slide show |

## 3.3 Image Installation

Android 9 distribution provide an eMMC flasing tool as following this [link](https://github.com/TechNexion/u-boot-edm/wiki/Use-mfgtool-to-flash-eMMC) to flash the image which you download.

## 3.4 Serial Debug

Please plug-in a micro USB cable to the connector(b) on the PI baseboard, and open any terminal communicate application that you common use such as minicom, Putty, setting 115200 bps as default speed.

**PI baseboard**

**(a)**

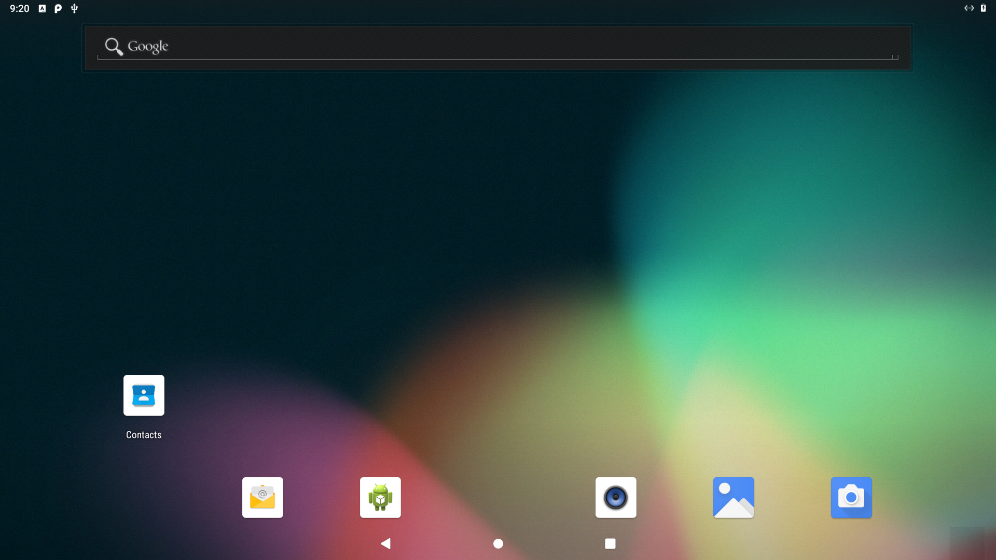
**(b)**

ADB way: plug-in a type-C USB cable to the connector(a) on the PI baseboard, and issue the command ‘adb shell’ such as android base mobile phone, then you can start debugging.

## 3.5 Network

* Ethernet

Android is auto running a DHCP daemon when boot, so we recommend it’s better for plug-in the cable before boot.



* WiFi
  + Station mode

Both support 2.4GHz and 5GHz band

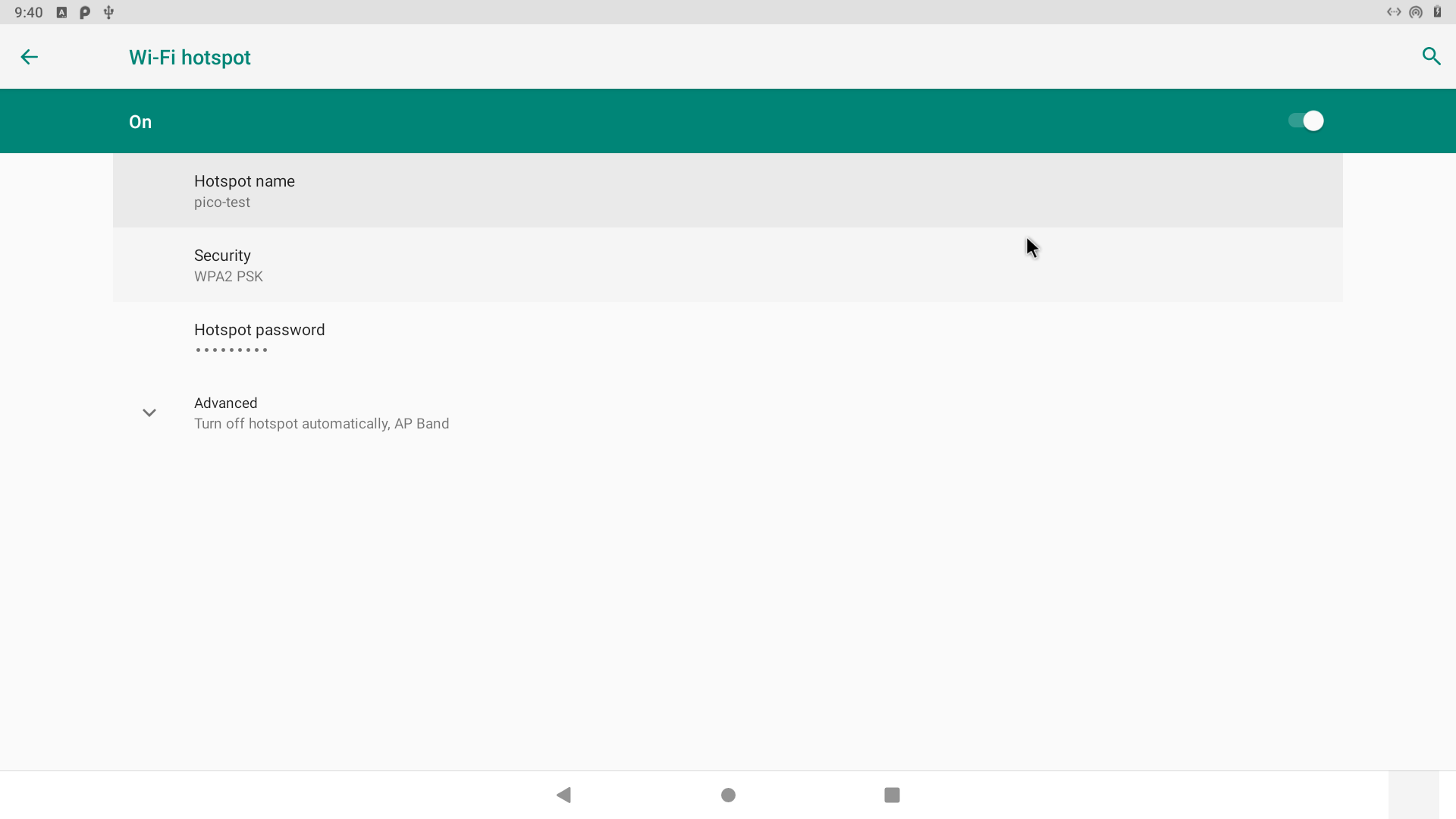
* + AP(Access Point) mode

Support 2.4GHz only with WPA- PSK security password

NOTE 1: It’s alternative between ethernet and WiFi station mode, it will show a failed WiFi connection when ethernet is working, please remove the ethernet cable first if you want to focus on WiFi station mode, or issue the command to disable the ethernet using chapter 3.2 way.

# ifconfig eth0 down

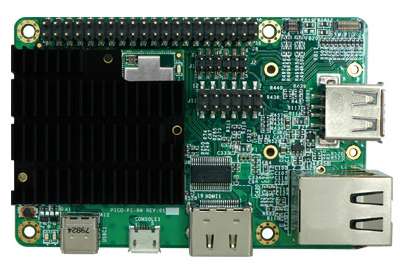
NOTE 2: Ethernet is necessary if you want to enable AP mode, ethernet will be a network node that can connect to outside on a Wide Area Network, so please plug-in the ethernet cable before boot up, or it will be a Local Area Network only.



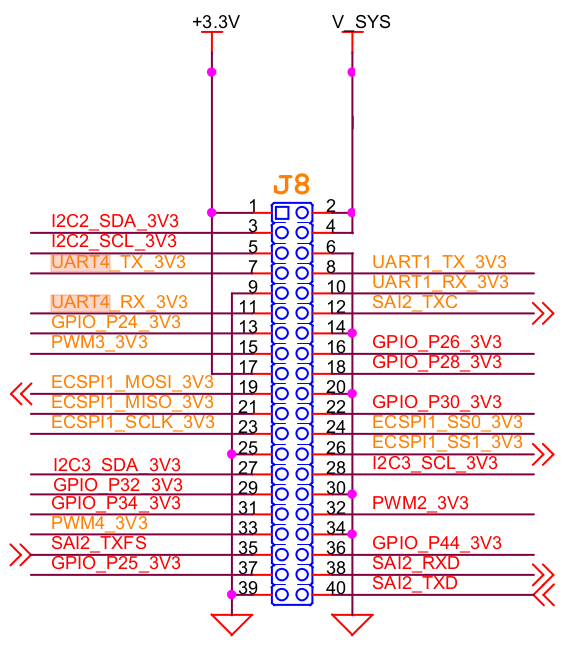
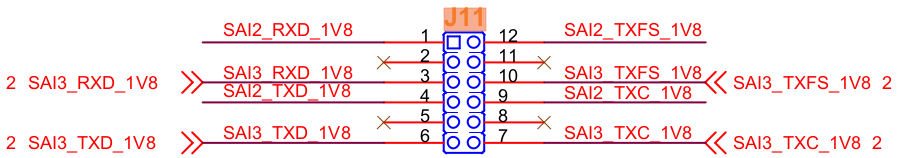
* Bluetooth
  + Support Bluetooth Classic mode
  + Support Bluetooth LE mode
  + Support A2DP source mode

## 3.6 Expansion Pin Header

**(J8)**

 **PI baseboard**

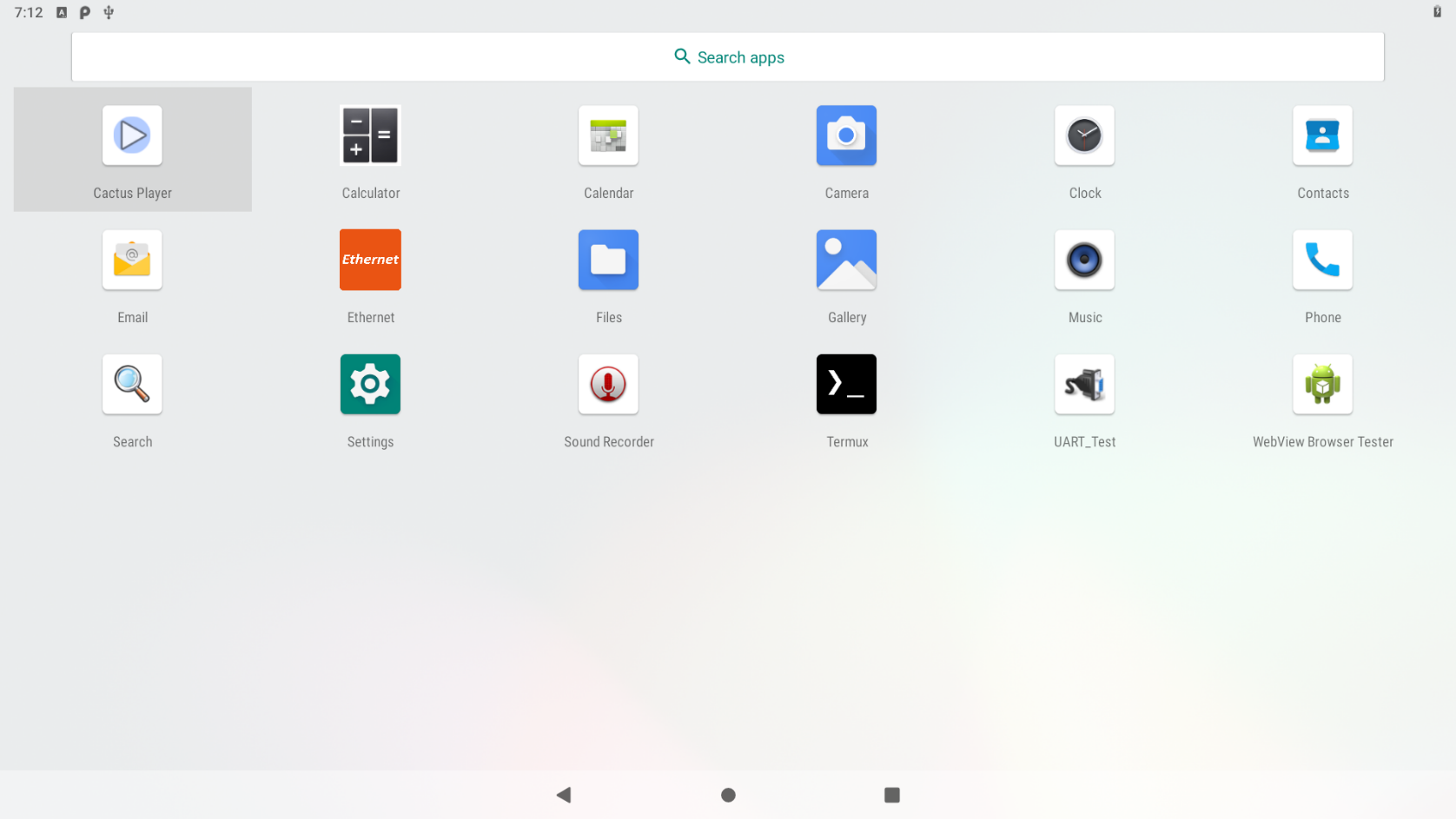
**(J9)**



PI baseboard provide an expansion pin header as above pin definition, the customers can easy to expand the customized I/O devices using GPIO, I2S, I2C, SPI and PWM low speed interfaces.

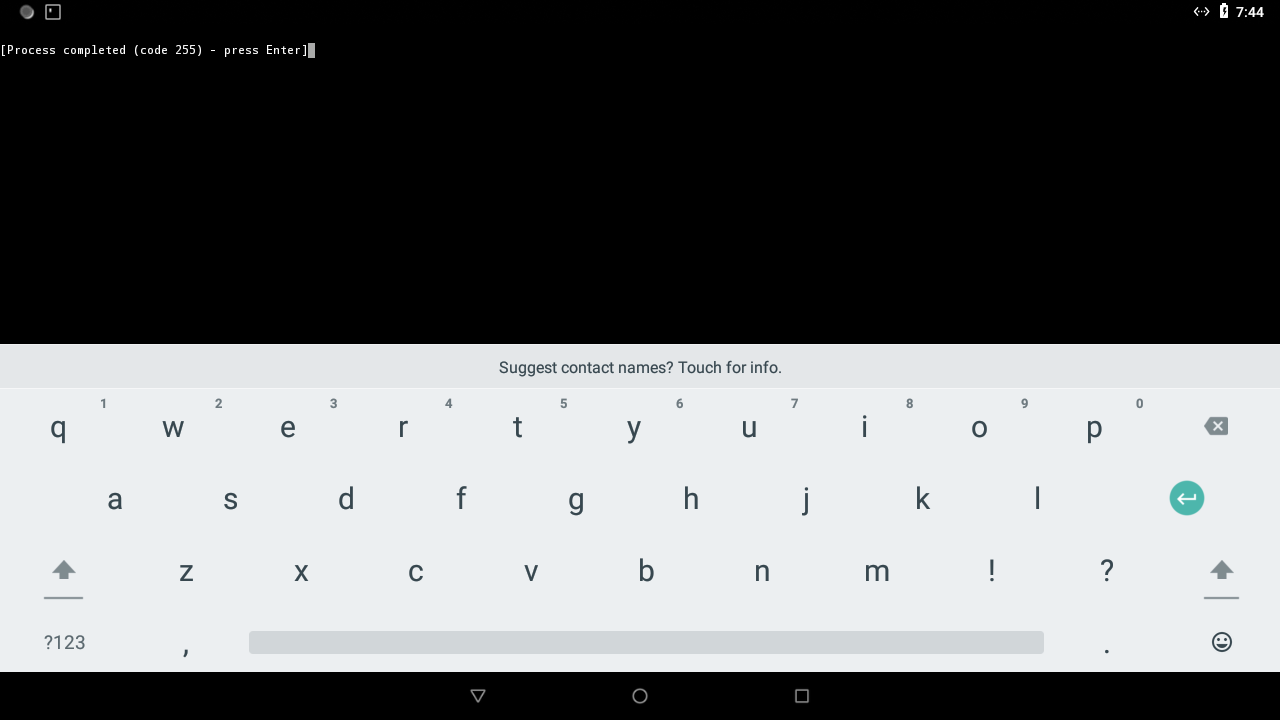
## 3.7 Termux app

Termux is a terminal emulator with Linux environment for Android, most of all, it support apt and opkg package managers, let Linux users developing easily if need running apps on Android.

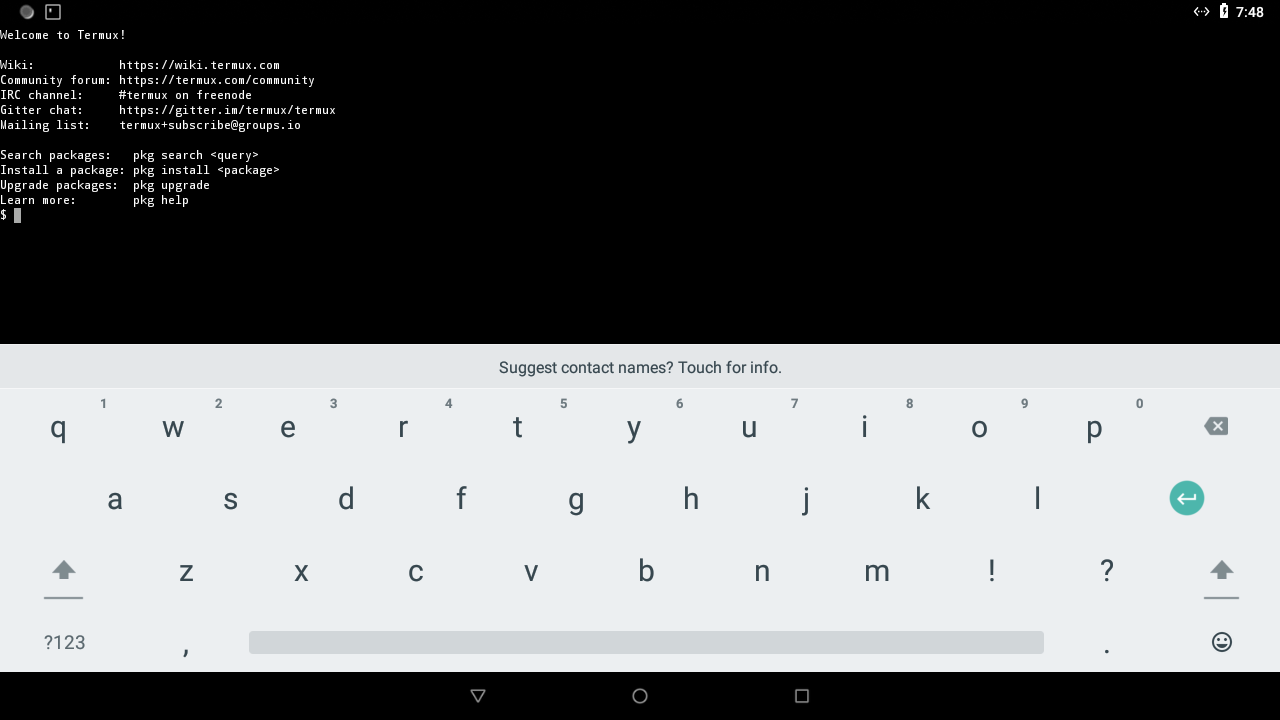


However, considering the security part (selinux on Android), so Termux cannot running on the normal user mode, users have to issue the commands to unlock the security limitation first, then starting enjoying the Linux apps development:

$ adb shell  
$ su  
# setenforce 0

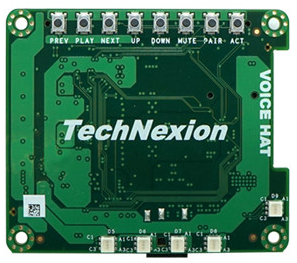


Termux not working when Android selinux locked.



Termux works when Android selinux unlocked.

## 3.8 Audio HAT

**Audio-HAT**

**(d)**

**(c)**

This Audio-HAT is design for expansion pin header as chapter 3.5, it can connect onto the expansion pin header directly, easy to use, Android 9 does support IMX7 and IMX8 platforms.

However, this distribution provide Termux app, it can easy to control the functions of Audio-HAT as following:

* + 16 channel LED array, **(d)** as above picture

# ls /sys/class/leds/

gpio-led pca995x:blue0 pca995x:blue4 pca995x:green2 pca995x:red1

mmc0:: pca995x:blue1 pca995x:blue5 pca995x:green3 pca995x:red2

mmc1:: pca995x:blue2 pca995x:green0 pca995x:green4 pca995x:red3

mmc2:: pca995x:blue3 pca995x:green1 pca995x:red0 pca995x:red4

echo 0 > /sys/class/leds/pca995x:blue0/brightness # (0% brightness)

echo 125 > /sys/class/leds/pca995x:blue0/brightness # (50% brightness)

echo 255 > /sys/class/leds/pca995x:blue0/brightness # (100% btightness)

* + Button array, **(c)** as above picture

Default button configuration as following:

These buttons can be modified the trigger event on kernel device tree of source code.

**Power**

**Prev**

**Play**

**Next**

**Mute**

**Volume Down**

**Volume Up**

**Home**

* + Speakers

1. Commands Testing

In Android, tinyalsa relative tool is easy to play the wave file, note that this speaker is support <= 16 bits.

# tinyplay test.wav -D 2 -c 2

Playing sample: 2 ch, 48000 hz, 16 bit (you can choose mono mode or stereo mode using -c parameter)

1. APP Developing

tn-p9.0.0\_2.0.0\_8m-ga is already support this speaker in system, the customers can use it directly.

* + MEMS microphone

1. Commands Testing

Note that this microphone is support 32 bits only.

# tinycap test.wav -D 0 -c 2 -r 48000 -b (you can choose mono mode or stereo mode using -c parameter)

1. APP Testing

Adapt the SoundRecorder app on desktop to test this, note that the sample rate is 48kHz

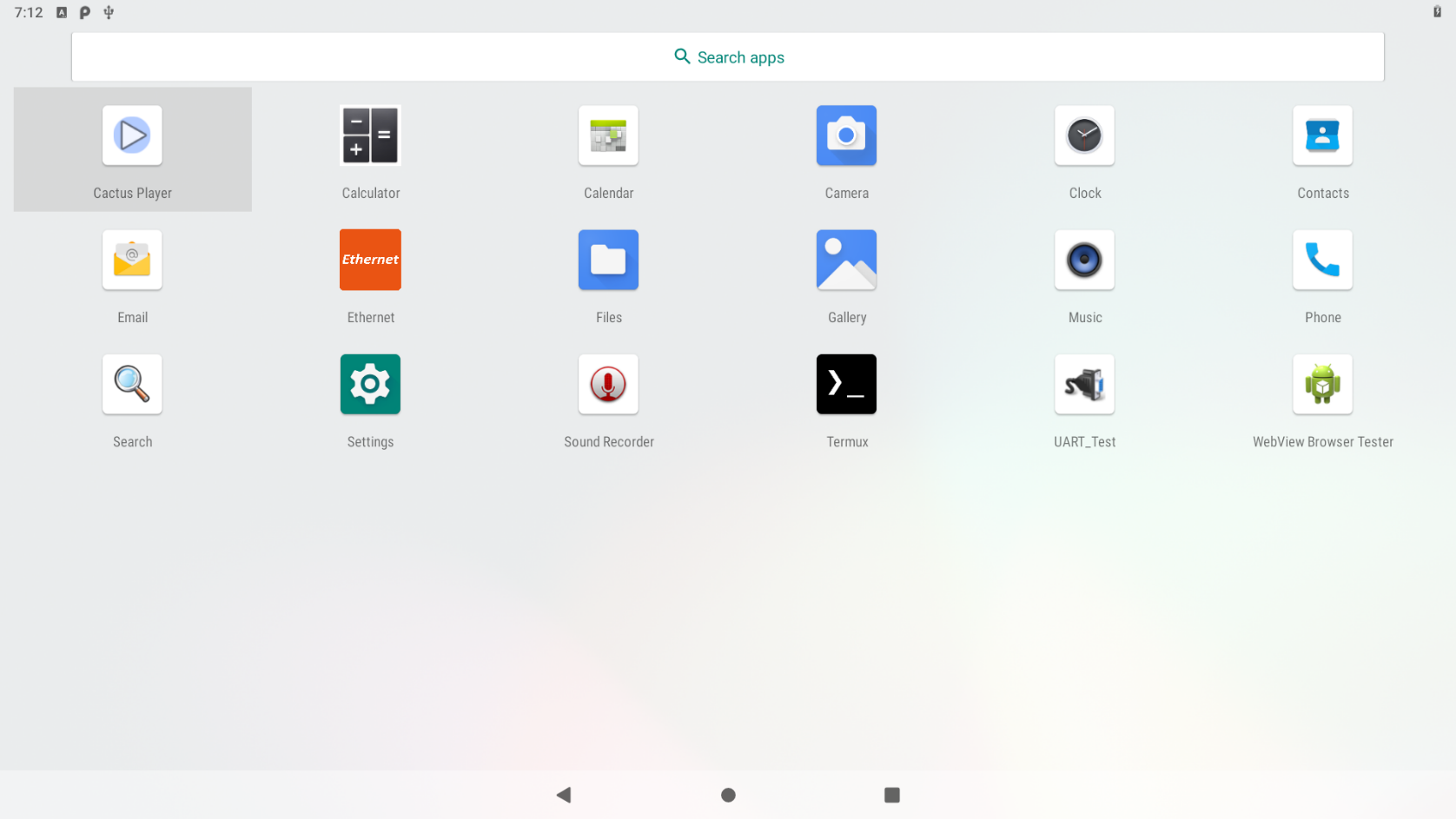
## 3.9 Serial Link

Supporting a 2-wire RS232 port on 40-pin expansion pin header as (J8) of chapter 3.5.

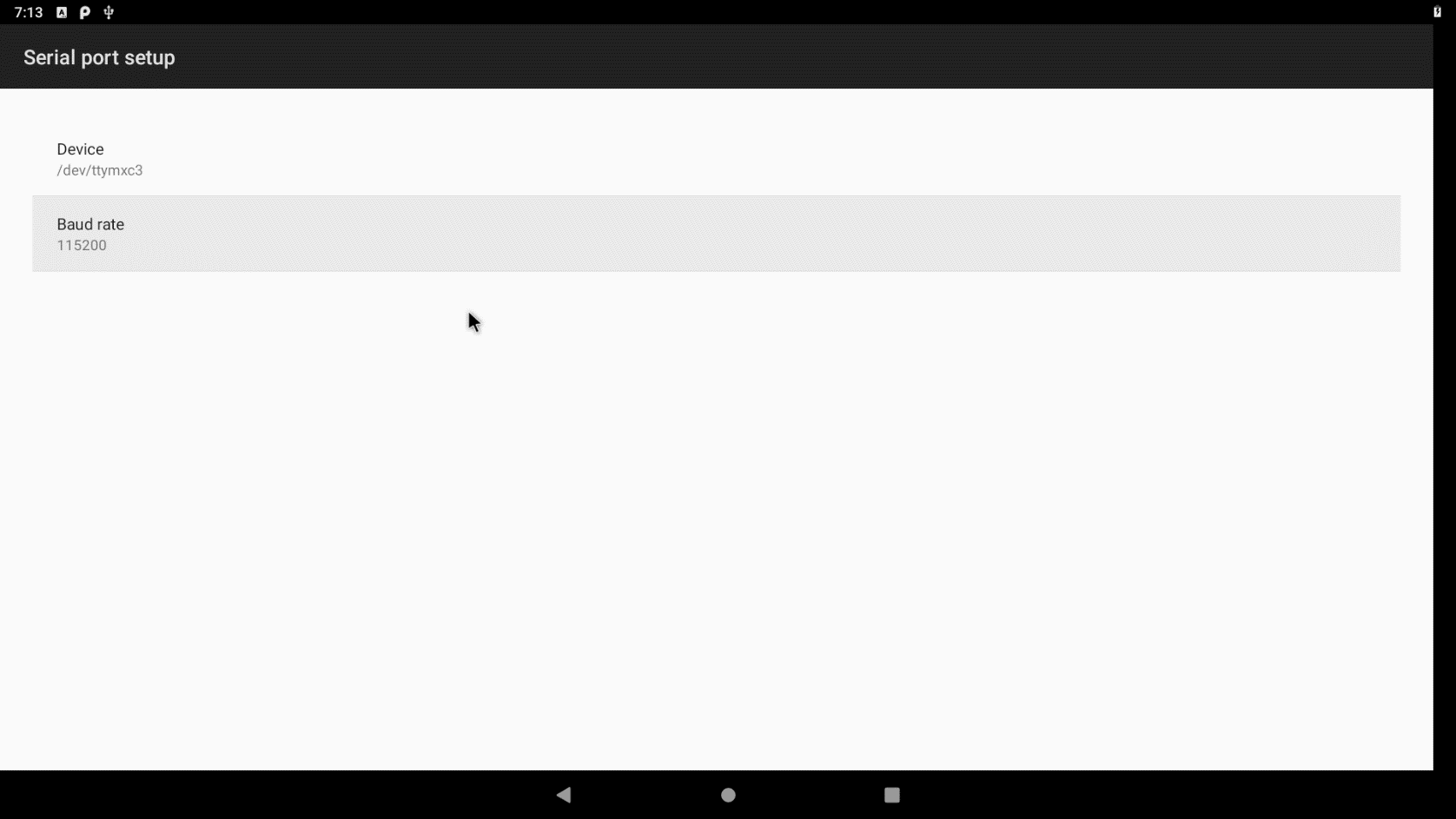
TX: Pin 7

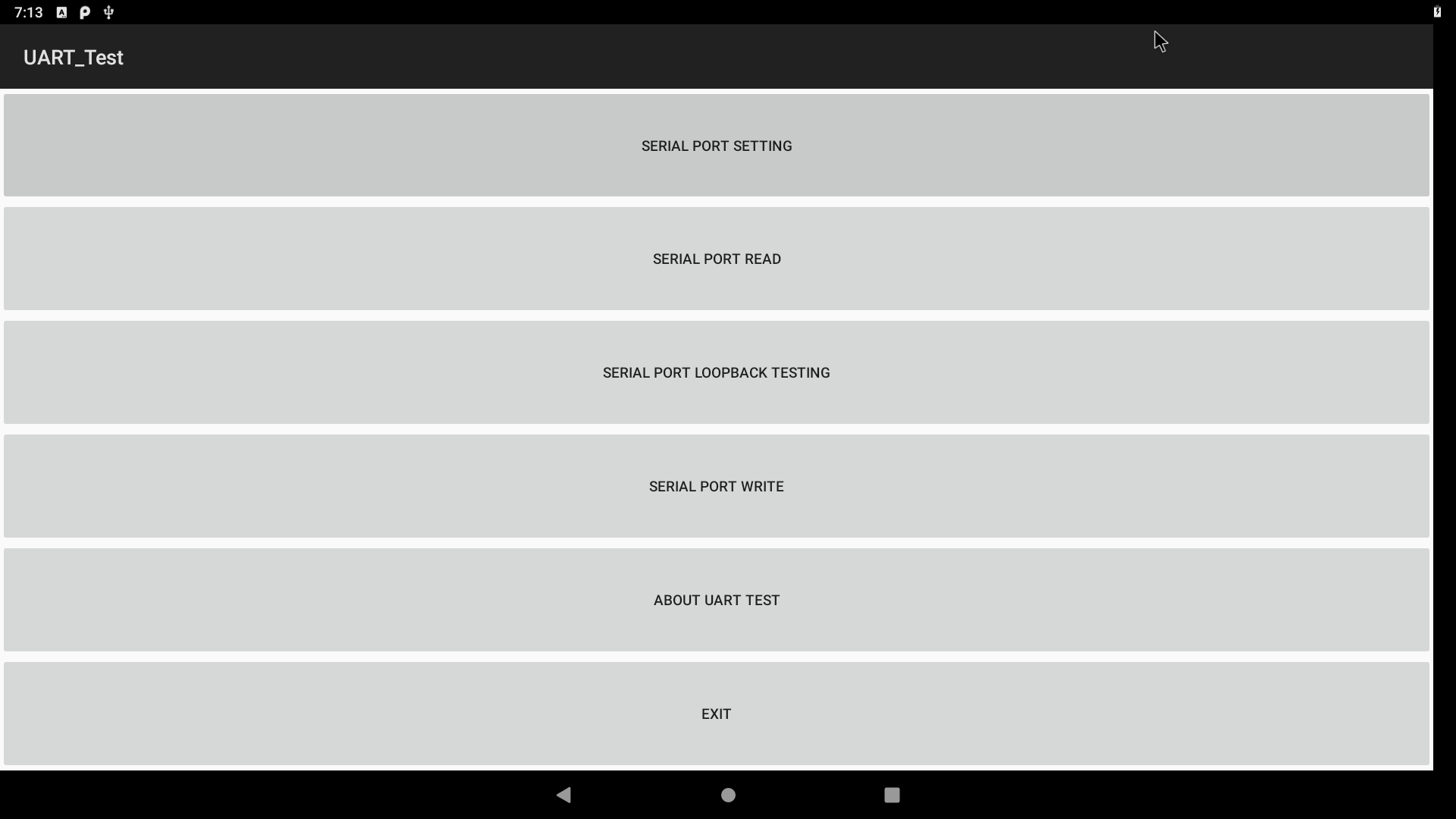
RX: Pin 11

Then open Technexion customized UART test app to do simple tests:



Clicking the “SERIAL PORT SETTING” to config the UART node and baud rate, then you can easy to test read, write and loopback function.





## 3.10 Android Treble

* + ADB

Connecting a USB type C cable between Host PC and hardware board, issue the commands to debug:

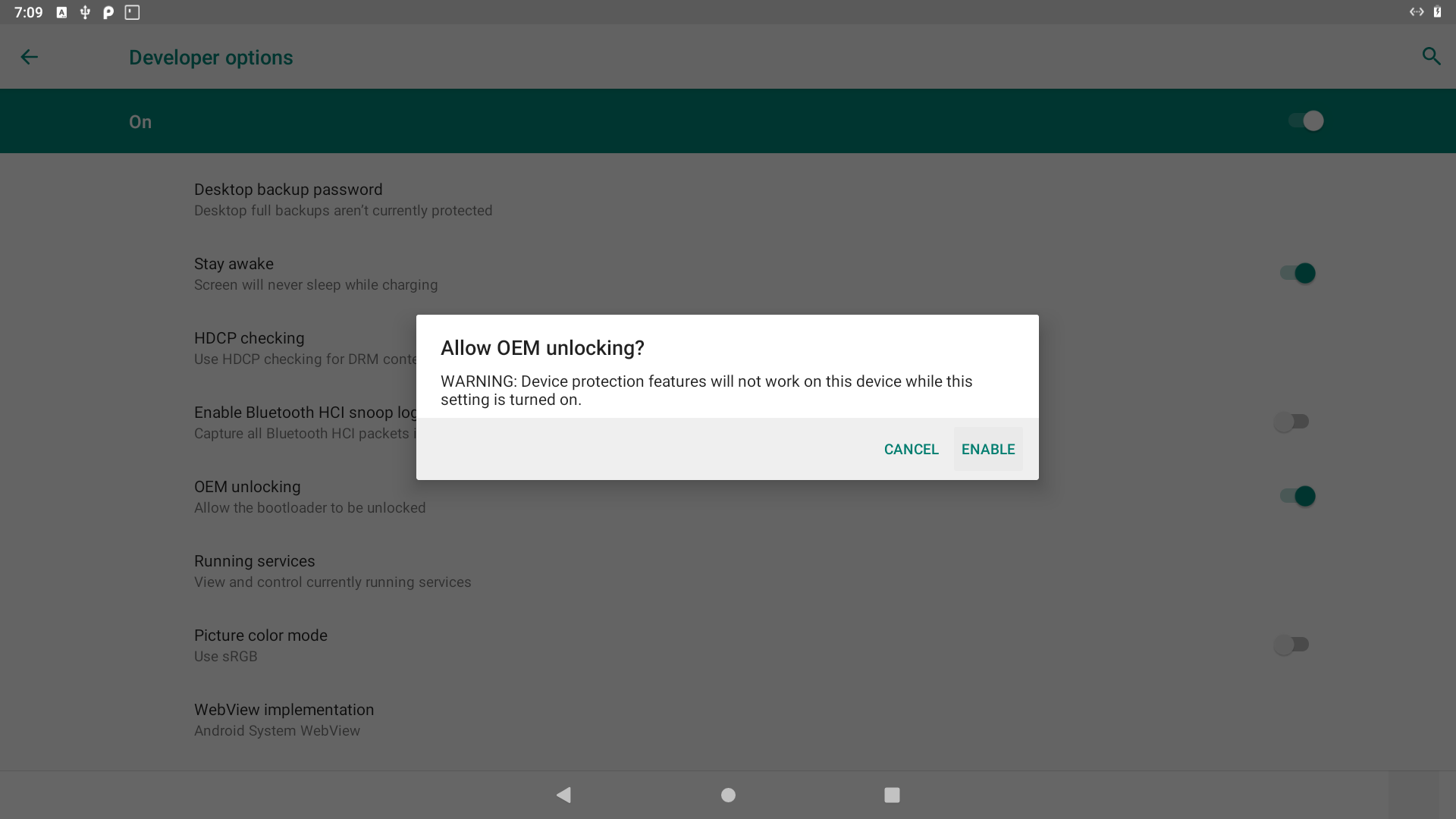
Root permission

$ adb root

Normal permission  
$ adb shell

* + Fastboot

In Android 9, users must be unlocking the permission of partition flashing using fastboot command, or you cannot do any upgrade behavior.



Unlock the flashing permission on developer options

After OEM unlocking, please reboot in fastboot mode (or issue fastboot -q on u-boot prompt) , issue the commands to unlock first and start flashing new images on your host PC:

$ sudo fastboot flashing unlock

# boot part upgrade

$ sudo fastboot flash dtbo\_a dtbo-imx8mq.img

$ sudo fastboot flash dtbo\_b dtbo-imx8mq.img

$ sudo fastboot flash boot\_a boot.img

$ sudo fastboot flash boot\_b boot.img

# system part upgrade

$ sudo fastboot flash vbmeta\_a vbmeta-imxpico\_8m.img  
$ sudo fastboot flash vbmeta\_b vbmeta-imxpico\_8m.img  
$ sudo fastboot flash system\_a system.img  
$ sudo fastboot flash system\_b system.img  
$ sudo fastboot flash vendor\_a vendor.img  
$ sudo fastboot flash vendor\_b vendor.img  
$ sudo fastboot reboot

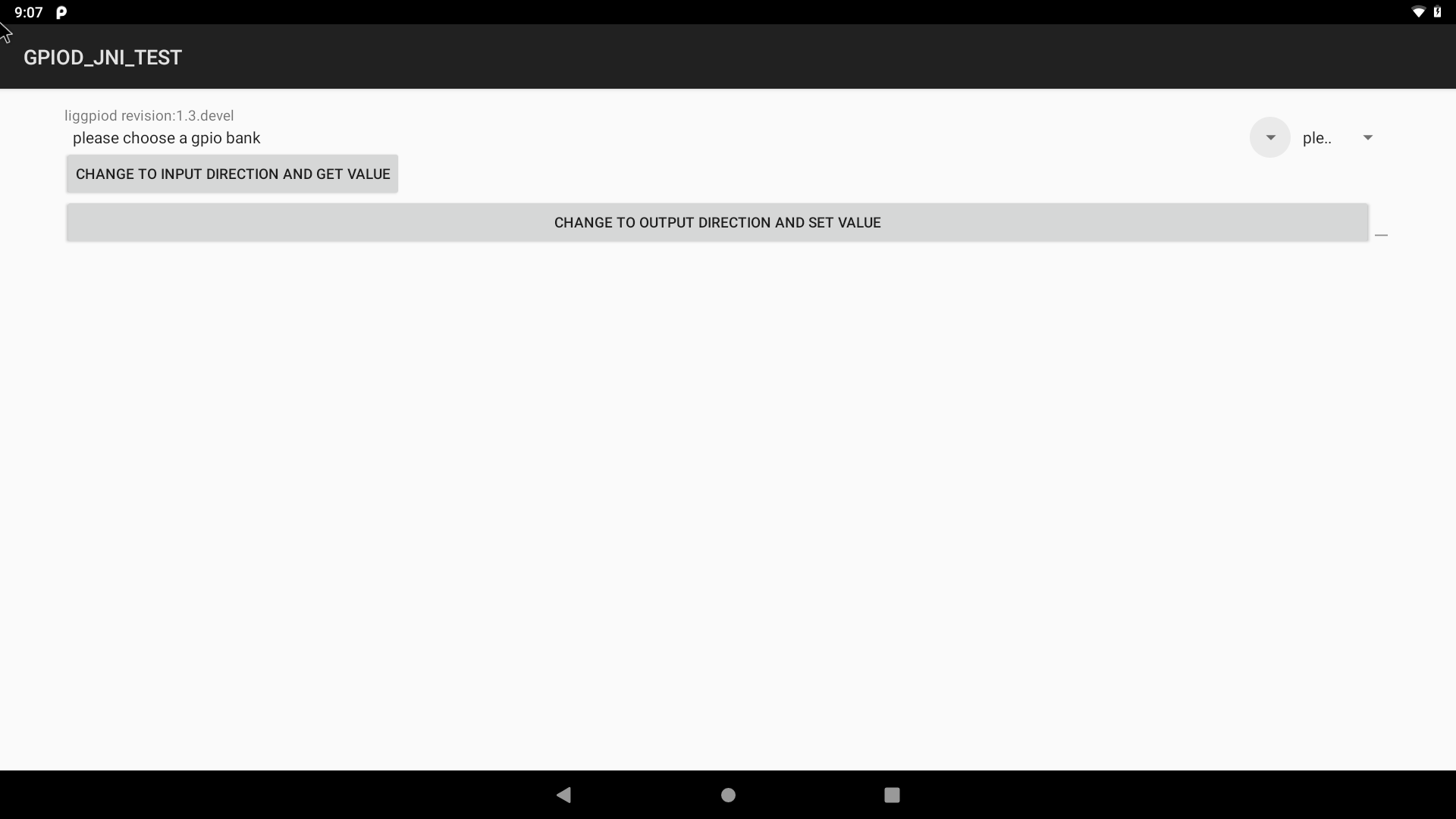
* + A/B system
  + OTA upgrade
  + Treble structure of source code

It’s fully supports on above three features, users can develop on our github SDK.

## 3.11 Libgpiod JNI Test APP

Libgpiod is a modular library on Linux which can easy to control the gpio pin instead of classic sysfs way, Technexion is porting to Android already, the customers can easy use libgpiod to developing HAL API, simple execute program or utility command-line tools, but the customers have to define the gpios on linux kernel device tree before use this library.

Latest TN2.0 release provide a libgpiod JNI test app, the customers can verify the JNI api easily.



**(4)**

**(5)**

**(3)**

**(2)**

**(1)**

Step 1. Choose a gpiobank

Step 2. Choose a gpioline

For example, gpio3\_7: gpiobank=3, gpioline=7

Step 3. The users can set this gpio as a GPIO input with a value getting, the hardware behavior will be triggered directly.

Or skip this step and go to next step

Step 4. The users can set this gpio as a GPIO output with a value, please input a value first and go to Step 5.

Step 5. Setting GPIO as a GPIO output, the hardware behavior will be triggered directly

# 4. Software Development and Upgrade

Online Github SDK including the source code and instruction.

<https://github.com/technexion-android/cookers/>

# 5. Q & A

1. IMX8: Video player only support portrait mode using MIPI-DSI LCD panel on landscape mode

**Ans:** It’s due to the OMX VPU library of chip vendor has some bugs, but we have a workaround way to change the parameters on your video using ffmpeg tool:

Ubuntu host example

$ ffmpeg -i original\_test.mov -vf "transpose=1" mipi\_lcd\_test.mov

$ ffmpeg -i mipi\_lcd\_test.mov -c copy -metadata:s:v:0 rotate=90 mipi\_lcd\_test\_out.mov

Then it will be working on landscape mode using video player if you adapt MIPI-DSI LCD panel.

1. IMX8: How to remount the system/vendor partition as a writeable partition?

**Ans:** Step 1. Unlock the flashing permission such as chapter 3.10 and reboot in fastboot mode.

# reboot bootloader

Step 2. Unlock the device again in fastboot mode on host side.

$ sudo fastboot oem unlock

$ sudo fastboot reboot

Step 3. Disabling the secure function using adb command on host side, then reboot again.

$ adb root

$ adb disable-verity

$ adb reboot

Step 4. Remount the all partitions as writeable partitions, for now, users can create a file to test.

## Host side

$ adb root

$ adb remount

$ adb shell

## Jump to Android side

$ su

# touch /system/test

# touch /vendor/test

1. IMX8: How to install uuu in different environment?

**Ans:** Windows

c:\uuu\_image\_folder>mkdir c:\utility

c:\uuu\_image\_folder>cpoy uuu.exe c:\utility

c:\uuu\_image\_folder>copy libusb-1.0.dll c:\utility

c:\uuu\_image\_folder>set PATH=%PATH%;c:\utility

**Ans:** Ubuntu

$ cd uuu\_image\_folder

$ sudo cp uuu /bin

1. IMX8: How to improve the graphic performance on demo image?

**Ans:** In demo image, the CMA configuration is for 1GB memory Sku., it’s in order to one image for all Sku., but it cause the performance is no good for 2GB, 4GB memory revision, please re-setting the CMA size again if you have requirement.

Re-config the boot argument on u-boot prompt (It means you need connect a debug console):

IMX8MQ:

# setenv bootargs console=ttymxc0,115200 earlycon=imxuart,0x30860000,115200

init=/init androidboot.hwrotation=0 androidboot.gui\_resolution=1080p androidboot.console=ttymxc0 consoleblank=0

androidboot.hardware=freescale **cma=1280M** androidboot.primary\_display=imx-drm

firmware\_class.path=/vendor/firmware transparent\_hugepage=never androidboot.wifivendor=qca androidboot.wificountrycode=TW quiet

* Note that your panel configuration
* Note that the good configuration are: CMA=800M for 2GB memory, CMA=1280M for 4GB memory

IMX8MM:

# setenv bootargs console=ttymxc1,115200 earlycon=ec\_imx6q,0x30890000,115200 init=/init androidboot.hwrotation=90 androidboot.console=ttymxc1 androidboot.hardware=freescale **cma=736M** androidboot.primary\_display=imx-drm firmware\_class.path=/vendor/firmware transparent\_hugepage=never quiet androidboot.wifivendor=qca androidboot.wificountrycode=TW