TechNexion
INNOVATORS OF TECHNOLOGY

# Android Pie User Manual

**VER. 2.10**
**September 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 |

## Table of Contents

# 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:

- UART serial test app
- Termux app
- libgpiod CLI(command-line) tool
- i2c-tools CLI tool
- Fastboot
- OTA update
- A/B system
- Bluetooth-LE
- Voice-HAT support
- Dual Camera preview demo
- Software web browser (IMX7 platforms only)
- A2DP audio

## 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 **(will be released soon)**
- IMX7 Series
  - PICO-IMX7
  - EDM-IMX7 **(will be released soon)**

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 |

**IMX6/IMX7 Series**

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

## 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)<br>boot_a (backup partition) | boot.img:<br>• kernel image<br>• recovery mode ramdisk |
| Partition 4 (FAT32)<br>boot_b | boot.img:<br>• Image: kernel image<br>• ramdisk.img: recovery mode ramdisk |
| Partition 5 (EXT4)<br>system_a (backup partition) | system.img |
| Partition 6 (EXT4)<br>system_b | system.img |
| Partition 7<br>misc | For recovery store bootloader message, reserve |
| Partition 8<br>metadata | For system slide show |
| Partition 9<br>presistdata | Option to operate unlock\unlock |
| Partition 10 (EXT4)<br>vendor_a (backup partition) | vendor.img |

| | |
|---|---|
| Partition 11 (EXT4)<br>vendor_b | vendor.img |
| Partition 12 (EXT4)<br>userdata | Application data storage for system application, and for internal media partition, in /mnt/sdcard/dir |
| Partition 13 (EXT4)<br>fbmisc | For storing the state of lock or unlock |
| Partition 12<br>vbmeta_a (backup partition) | For storing the verify boot's metadata |
| Partition 13<br>vbmeta_b | For storing the verify boot's metadata |

**IMX6/IMX7 Series**

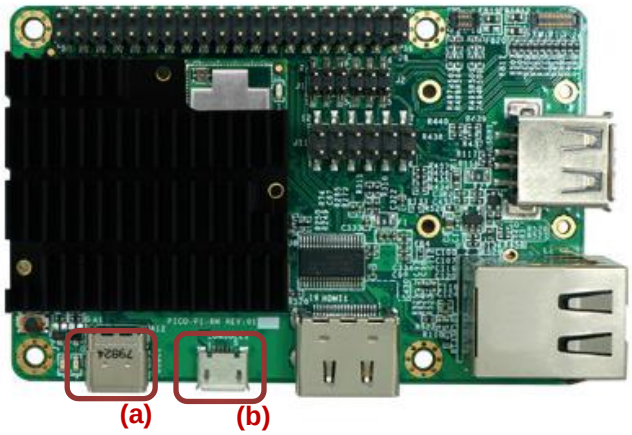
| **Section** | **Description** |
|---|---|
| GPT | Partition information |
| Bootloader | SPL: First stage u-boot image, u-boot.img: Second stage image |
| Partition 1 dtbo (backup partition) | dtbo.img |
| Partition 2 (FAT32)<br>boot | boot.img:<br>• kernel image<br>• ramdisk |
| Partition 3 (FAT32)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

## 3.3 Image Installation

Android 9 distribution provide an eMMC flasing tool as following this link 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.

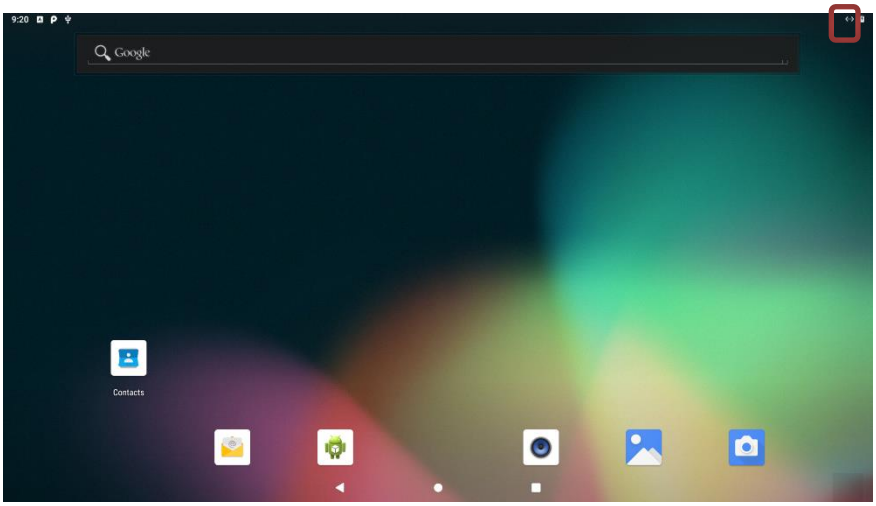



**PI baseboard**

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.

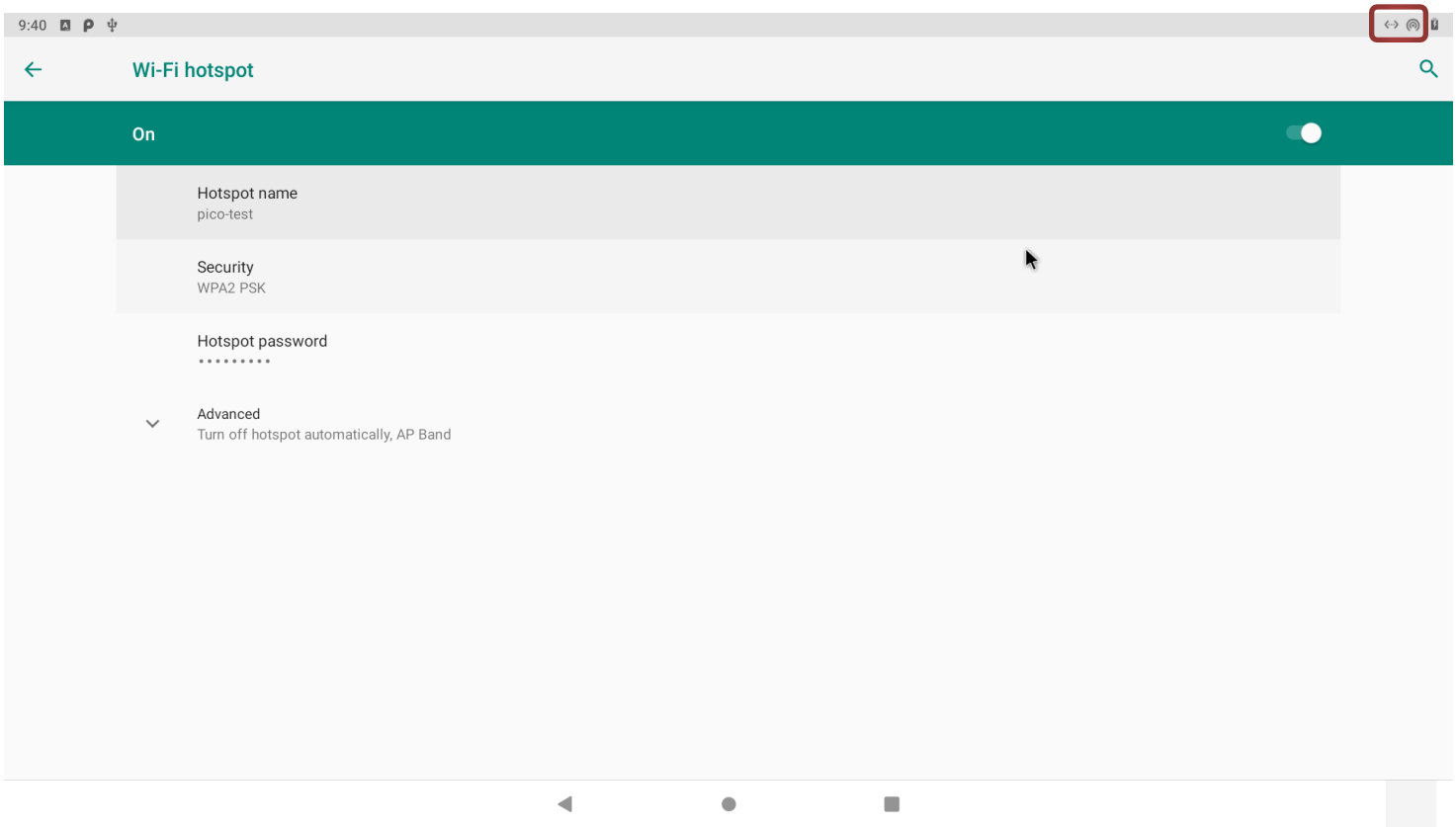



- WiFi
  - Station mode
    Both support 2.4GHz and 5GHz band
  - AP(Access Point) mode
    Support 2.4GHz only with WPA- PSK security password

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.

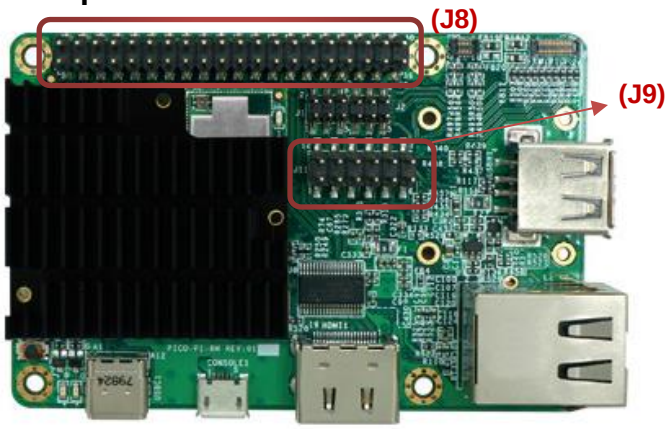
```
# ifconfig eth0 down
```

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.

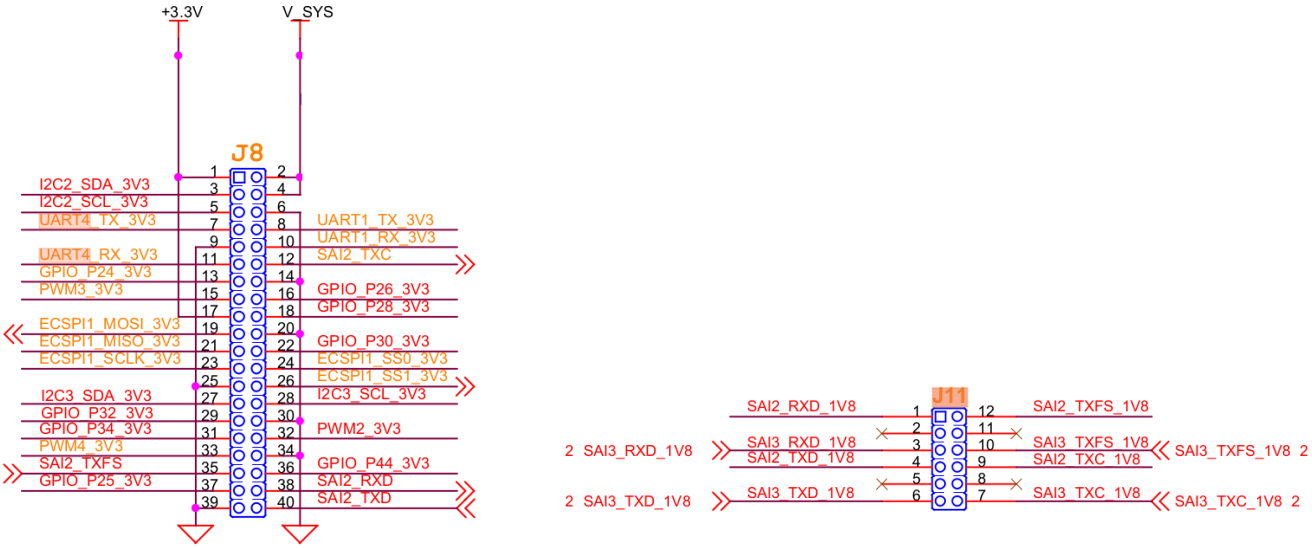



- Bluetooth
  - Bluetooth Classic mode
  - Bluetooth LE mode

## 3.6 Expansion Pin Header



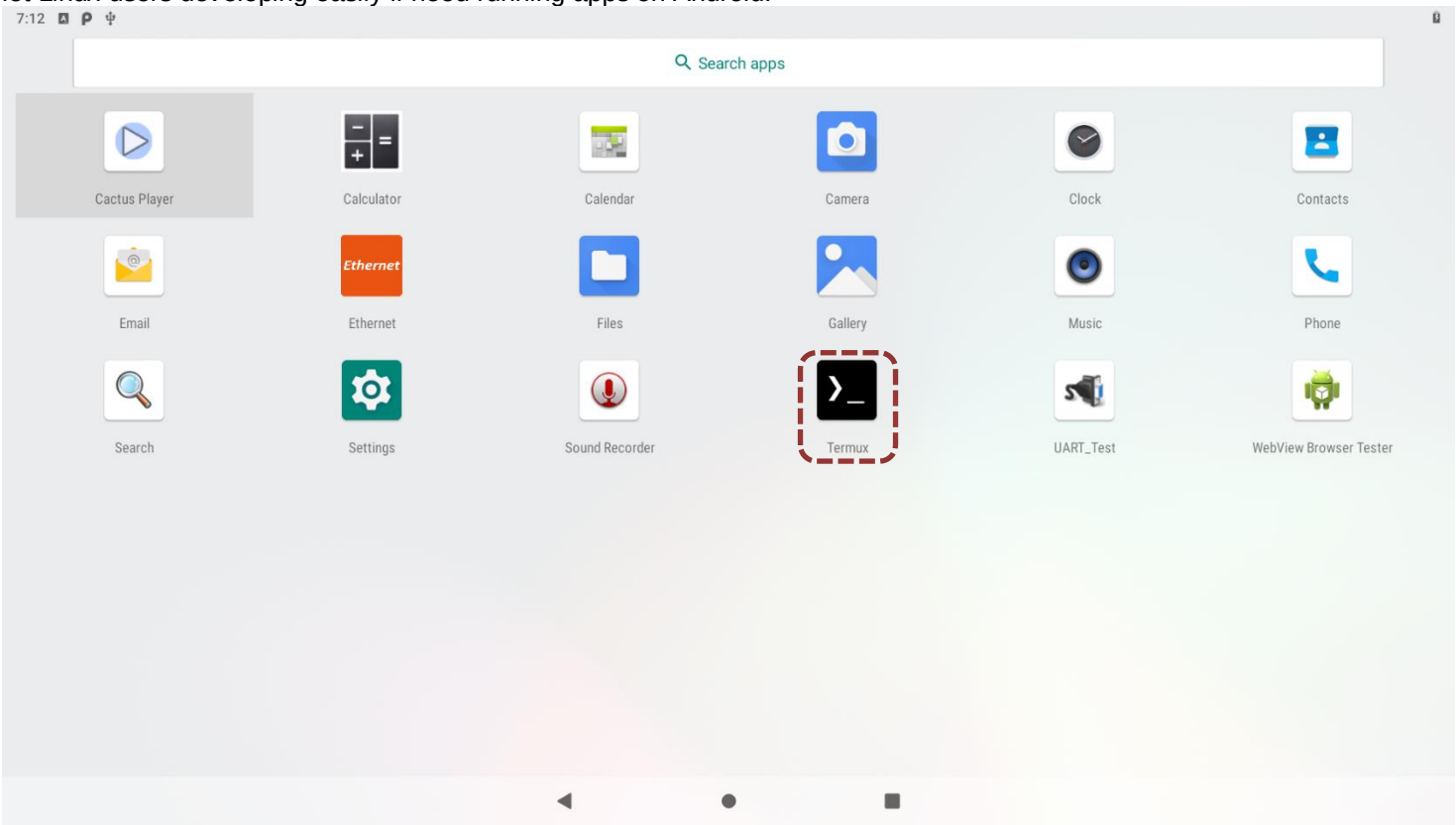

**PI baseboard**




PI baseboard provide an expansion pin header as above pin definition, the customers can easy to expand the customized I/O devices using GPIO, $I^2S$, $I^2C$, SPI and PWM low speed interfaces.

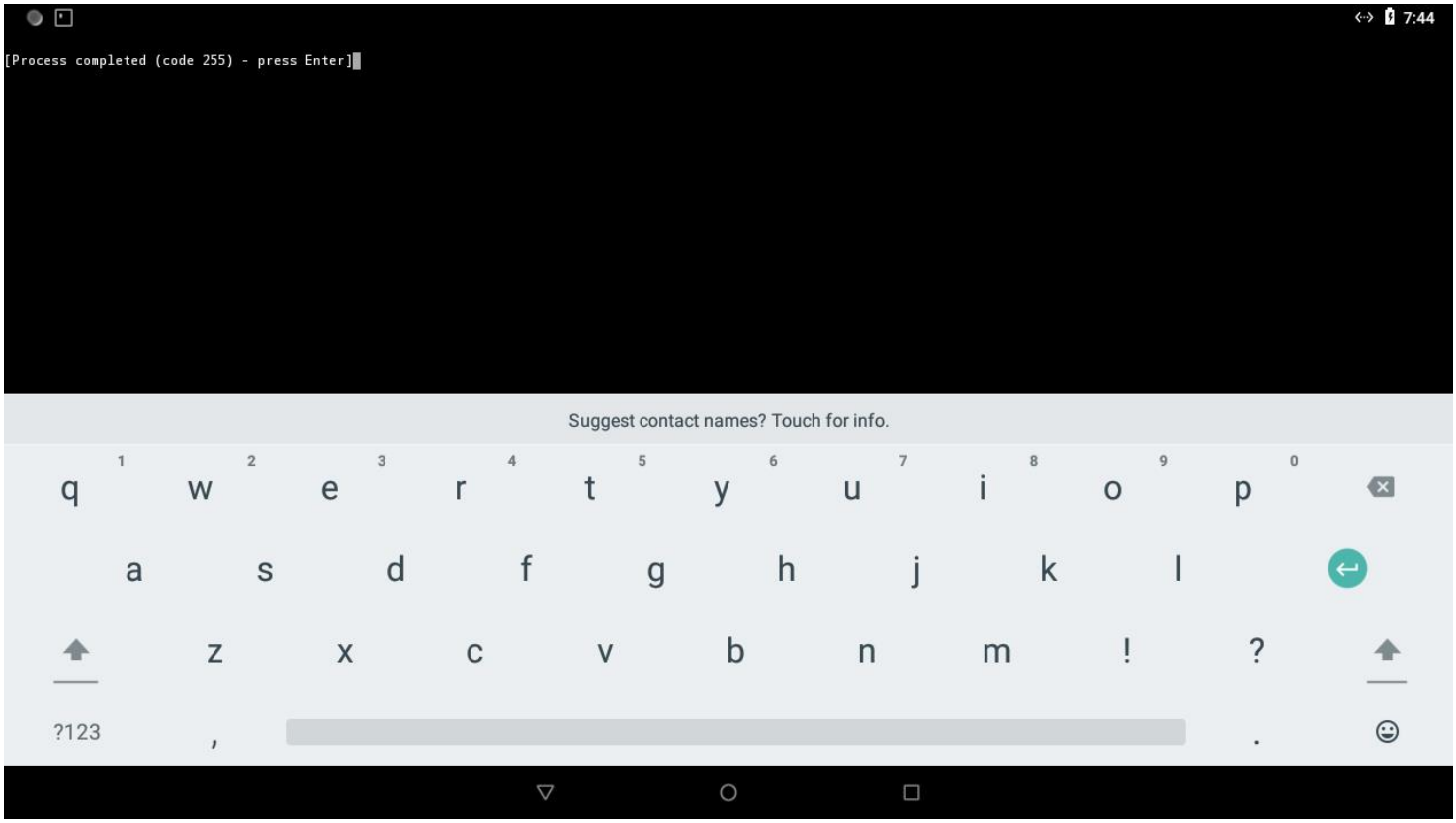
## 3.7 Termux app

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.

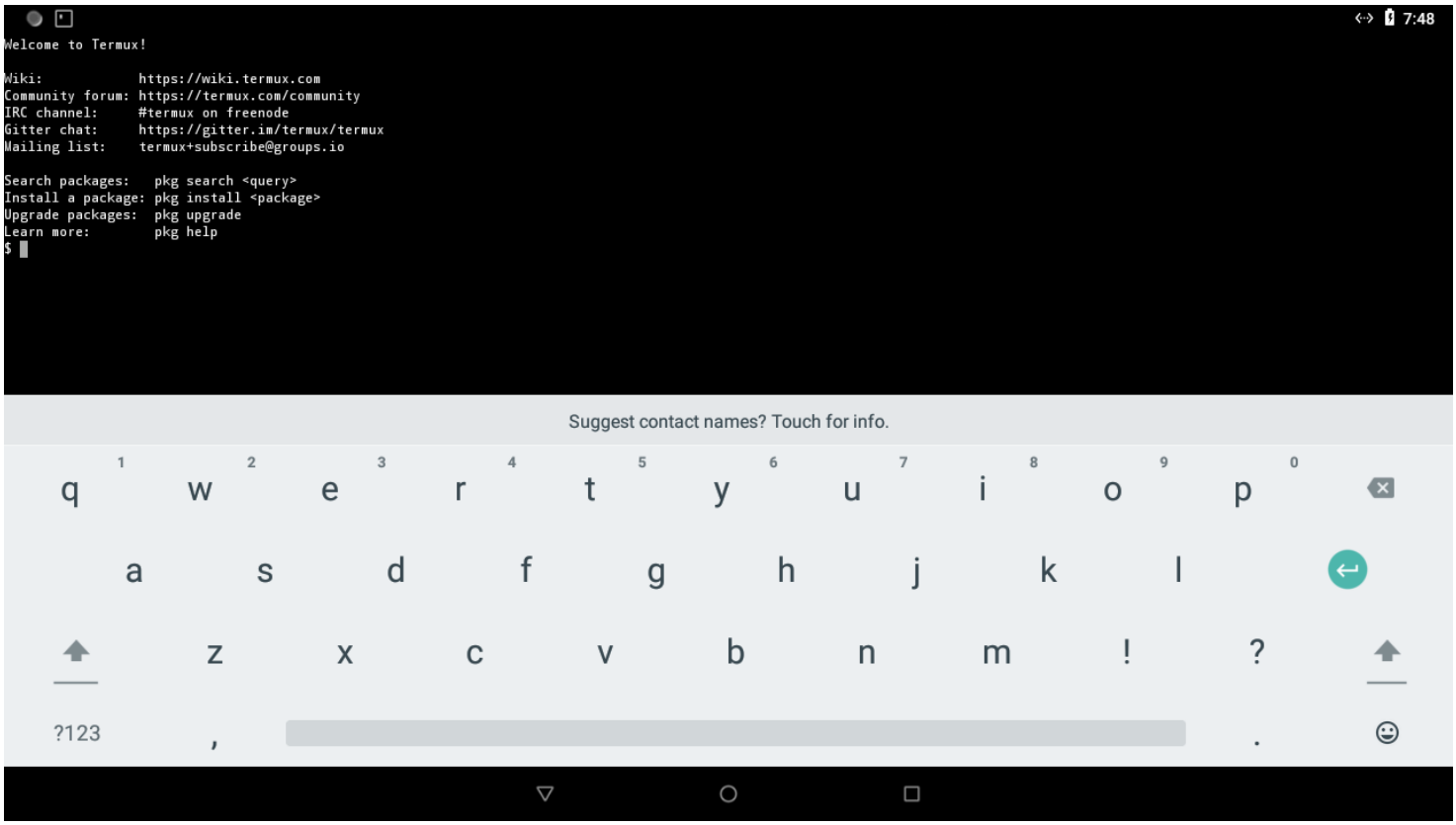



However, considering the security part (selinux on Android), so Termux cannot running on the normal user mode, users have to issue the commands to unlock the security limitation first, then starting enjoying the Linux apps development:

```
$ adb shell
$ su
# setenforce 0
```




Termux not working when Android selinux locked.




Termux works when Android selinux unlocked.

## 3.8 Audio HAT


**Audio-HAT**

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 also support Audio-HAT using sysfs way, future will support in android framework for next distribution.

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.

| Prev | Play | Next | Volume Up | Volume Down | Mute | Home | N/A |
|---|---|---|---|---|---|---|---|

- 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)
```

  2. 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)
```

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

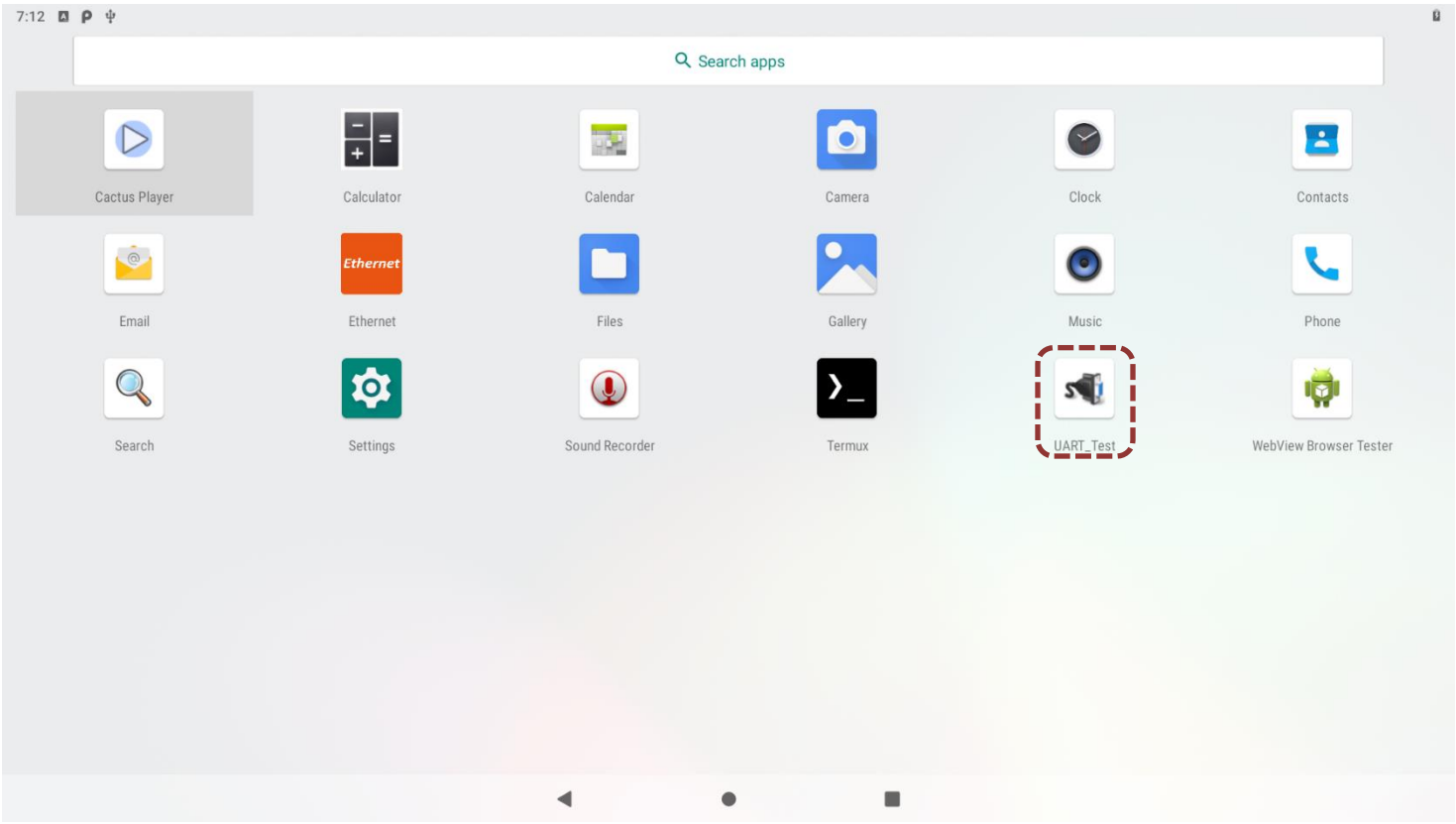
## 3.9 Serial Link

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

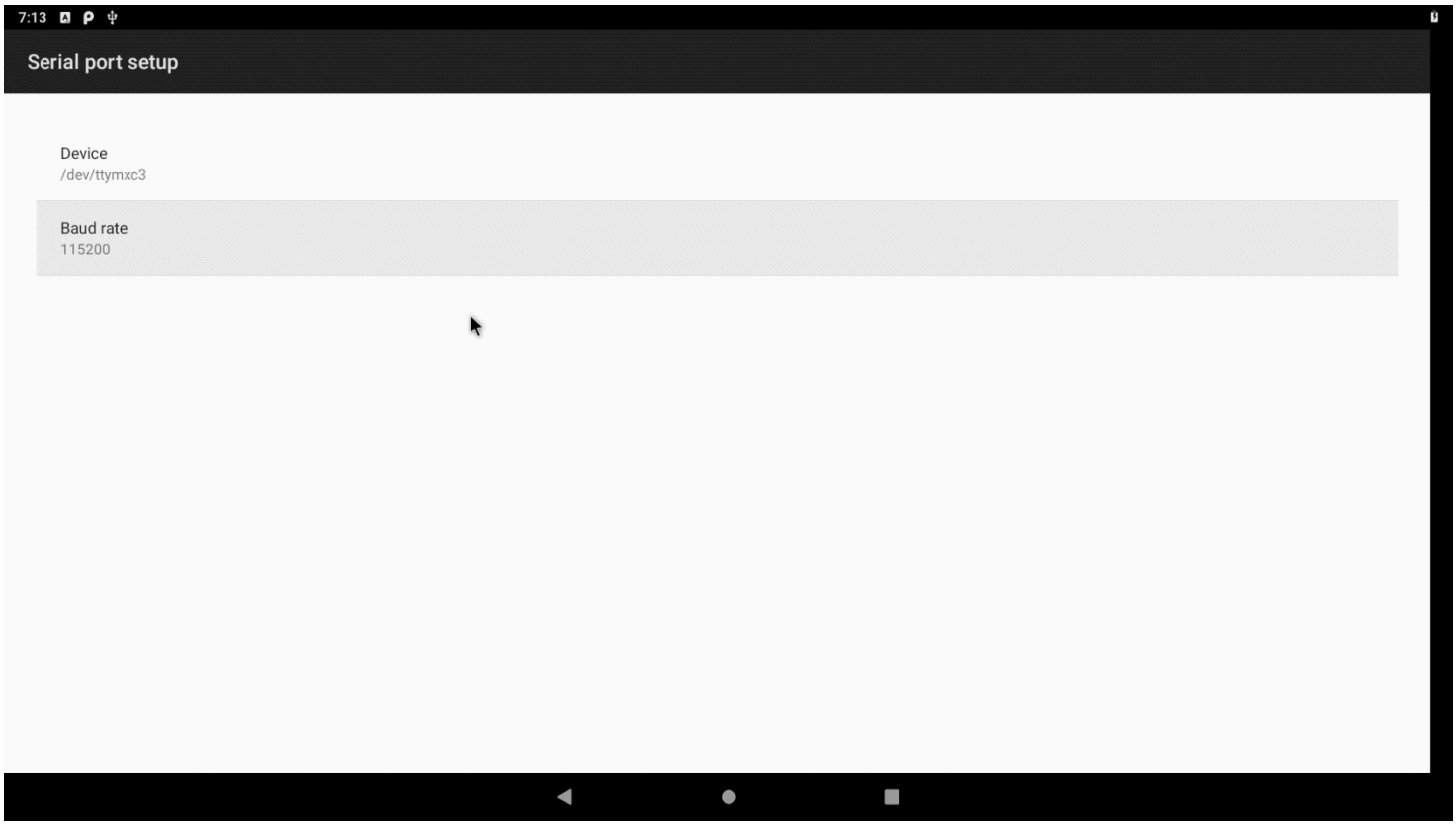
TX: Pin 7

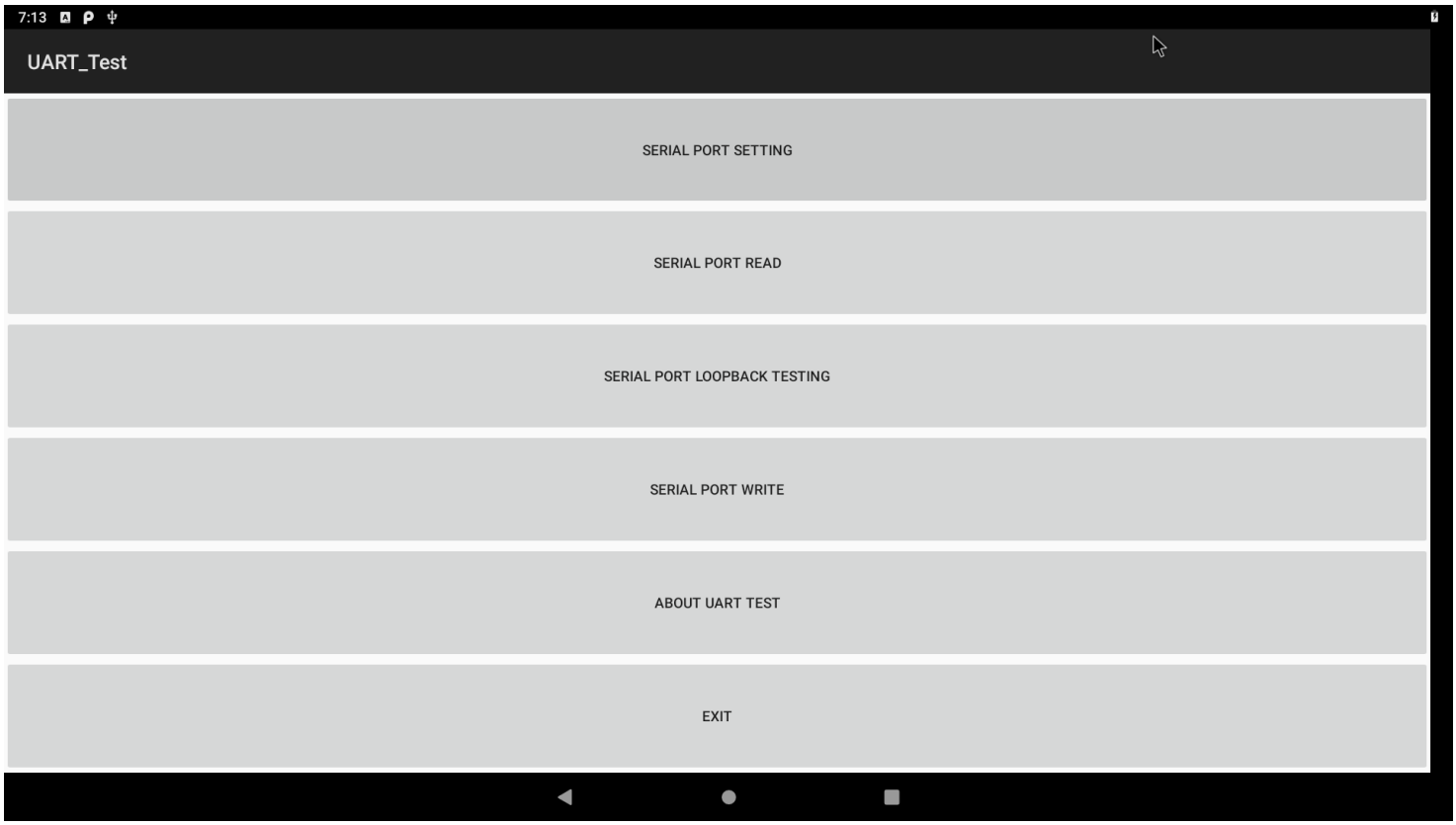
RX: Pin 11

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

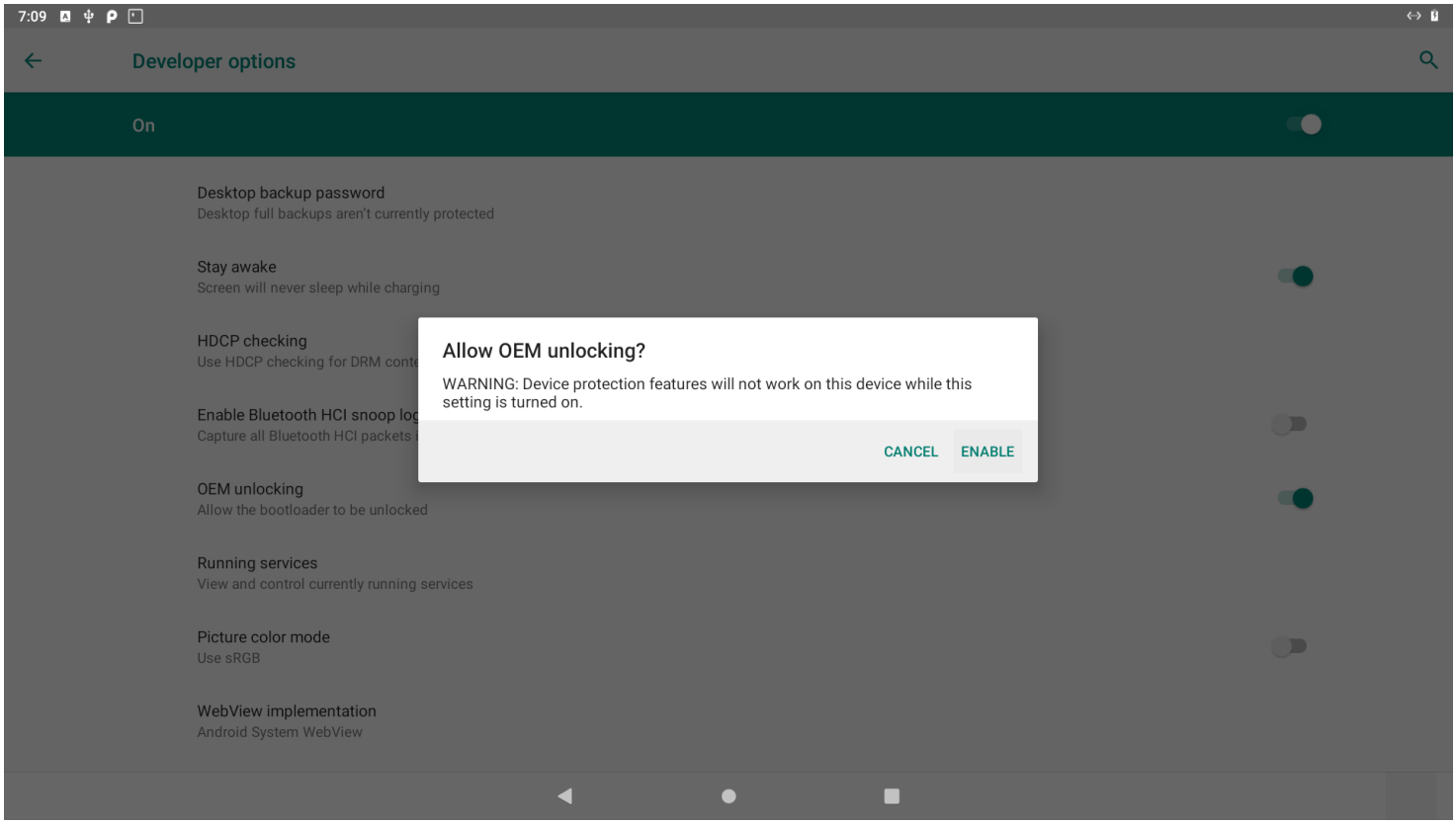
- ADB

  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 Framework

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.

For Example (pico-imx8m with pi baseboard):
<source>/vendor/nxp-opensource/kernel_imx/arch/arm64/boot/dts/freescale/imx8mq-pico-pi.dts

```
+&gpio3 {
+       gpio-line-names =
+               "", "", "", "", "TEST", "", "", "",
+               "", "", "", "", "", "", "", "",
+               "", "", "", "", "", "", "", "",
+               "", "", "", "", "", "", "", "";
+
+       pinctrl-0 = <&pinctrl_libgpiod_test>;
+};
+
+
 &iomuxc {
        imx8mq-pico {
                pinctrl_otg_vbus: otgvbusgrp {
@@ -161,6 +172,12 @@
                                MX8MQ_IOMUXC_SPDIF_EXT_CLK_GPIO5_IO5    0x19
                        >;
                };
+
+               pinctrl_libgpiod_test: libgpiodtestgrp {
+                       fsl,pins = <
+                               MX8MQ_IOMUXC_NAND_CE3_B_GPIO3_IO4               0x41
+                       >;
+               };
        };
 };
```

After compiled and flashed, gpio3-4 will be defined in gpio driver, issue the commands when boot Android up:

```
pico_imx8m:/ # gpiodetect
gpiochip0 [30200000.gpio] (32 lines)
gpiochip1 [30210000.gpio] (32 lines)
gpiochip2 [30220000.gpio] (32 lines)
gpiochip3 [30230000.gpio] (32 lines)
gpiochip4 [30240000.gpio] (32 lines)
```

```
Set gpio3-4 to lo:
pico_imx8m:/ # gpioset 30220000.gpio 4=0

Set gpio3-4 to hi:
pico_imx8m:/ # gpioset 30220000.gpio 4=1
```

Most relative commands supporting in Technexion libgpiod framework, enjoy.

# 4. Software Development and Upgrade

Online Github SDK including the source code and instruction.
https://github.com/technexion-android/cookers/

## 5. Q & A

1. 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.

2. 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
```

3. 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
```