# MaaXBoard Osm93 Yocto User Manual

## REV. LF6.6.3-1.0.0

## Copyright Statement:

- The MaaXBoard Osm93 single board computer and its related intellectual property are owned by Avnet.
- Avnet has the copyright of this document and reserves all rights. Any part of the document should not be modified, distributed or duplicated in any approach and form without the written permission issued by Avnet.

## Disclaimer:

- Avnet 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 Osm93 Osm93 single board computer has passed the CE, FCC & SRRC certification.

# Catalog

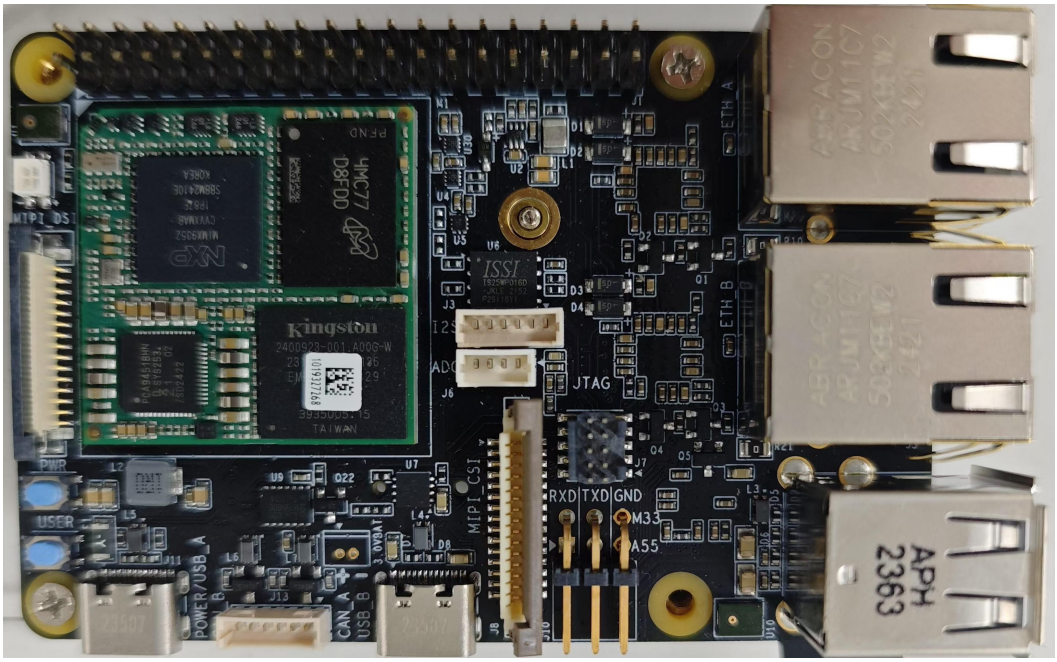
# Chapter 1 Introduction

## 1.1 Target Board: MaaXBoard Osm93

MaaXBoard OSM93 features an NXP i.MX 93 System on Chip compute module, with integrated AI/ML NPU accelerator, EdgeLock security enclave and Energy Flex architecture that supports separated processing domains, such as the Application domain with two Arm® Cortex®-A55 (1.7 GHz) cores, the real time domain with Arm® Cortex®-M33 (250 MHz) core and Flex domain with Arm® Ethos-U65 NPU (1 GHz). Other resources on the fitted MSC OSM-SF-IMX93 solder-down module include eMMC (16GB) memory, LPDDR4 (2GB, 3.7 GT/s) with inline ECC support, RTC clock and NXP PCA9451 PMIC.

The Raspberry Pi form-factor carrier SBC carrier board adds QSPI flash memory (16Mbit) plus connectivity and UI interfaces. High speed interfaces include four USB 2.0 interfaces (2x host type A, 1x host type-C, 1x device type-C), MIPI DSI display and MIPI CSI camera interfaces, two 1 Gbps Ethernet ports and two high-speed CAN interfaces. Expansion interfaces include a Pi-Hat 40pin-header, 6-pin ADC header and 6-pin SAI digital audio header (supplemented by two onboard PDM microphones). Level-shifted debug UARTs are pinned-out for the application and RT cores.

An M.2 key-E connector on back of the board facilitates easy integration of optional NXP based tri-radio M.2 module solutions, for concurrent Wi-Fi 6, Bluetooth (5.3) and 802.15.4 wireless operation.



## 1.2 Introduction

This document provides a guide to prepare MaaXBoard Osm93 to boot up with the Verified Linux Package and introduces how to use the functions of MaaXBoard Osm93.

## 1.3 Feature List

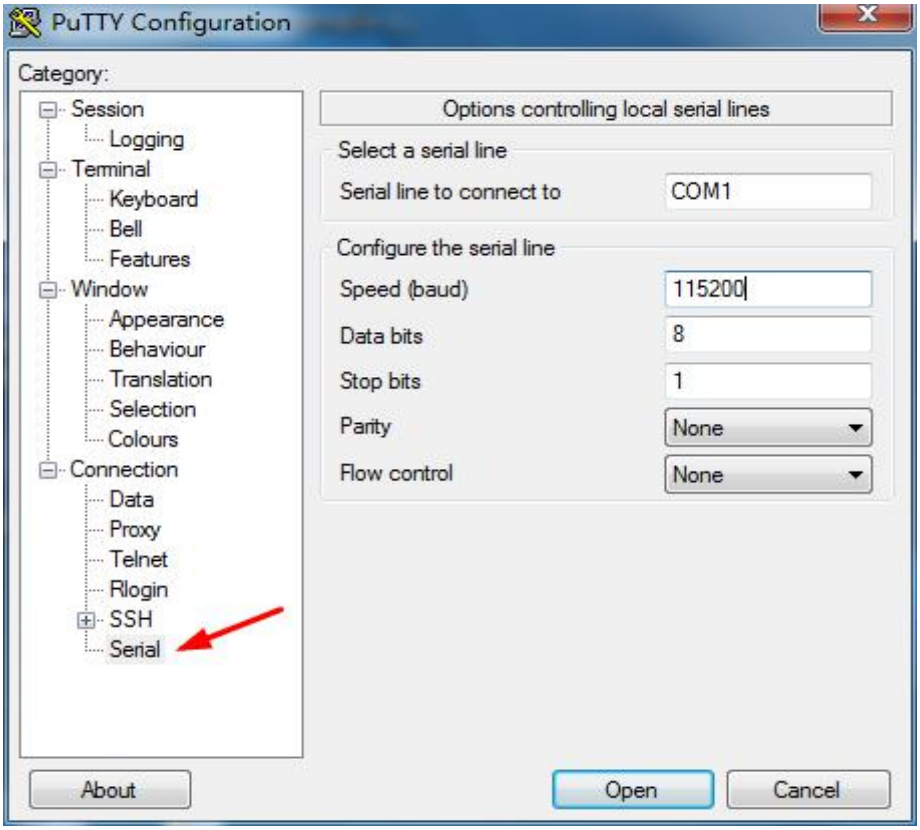
- Yocto version: nanbield, based on NXP SDK version: imx-6.6.3-1.0.0
- U-Boot version: 2023.04
- Kernel version: 6.6.3
- Evaluation image: Yocto nanbield
- Development based on NXP i.MX 93
- eMMC boot
- Device-tree Overlay support
- Desktop (Weston 11.0)
- 2x 1 Gbps Ethernet ports (RJ45)
- 4x USB 2.0 interfaces (2x host type A, 1x host type-C, 1x device type-C)
- 2x UART debug ports
- 2x high-speed CAN interfaces
- Pi-Hat 40pin-header (I2C,UART,SPI and GPIO)
- 6-pin ADC header
- 6-pin SAI digital audio header (supplemented by two onboard PDM microphones)
- WIFI & BLE 5.3
- MIPI-DSI display
- MIPI-CSI Camera/USB Camera

# Chapter 2 Quick Start

The default version of MaaXBoard Osm93 supports eMMC. To program the image into eMMC, refer to Chapter 4 Program or update the system Images. For the hardware connection and accessories details, please check the QSG.

## 2.1 Boot from eMMC

- 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 A55 debug interface to PC with USB to TTL converter. Pin 1, 3 and 5 of J10 to the TXD, RXD and GND pin of the USB to TTL converter.
- Powered the board with a 5V, 2A, Type-C interface power (to J11).
- When the system boot up, the serial terminal will print the following information:

```
NXP i.MX Release Distro 6.6-nanbield maaxboardosm93 ttyLP0
maaxboardosm93 login:
```

- Enter username as “root” to login.

```
maaxboardosm93 login: root
root@maaxboardosm93:~#
```

- Users could also connect keyboard and mouse to MaaXBoard Osm93 to login Yocto system.

## 2.2 Login system

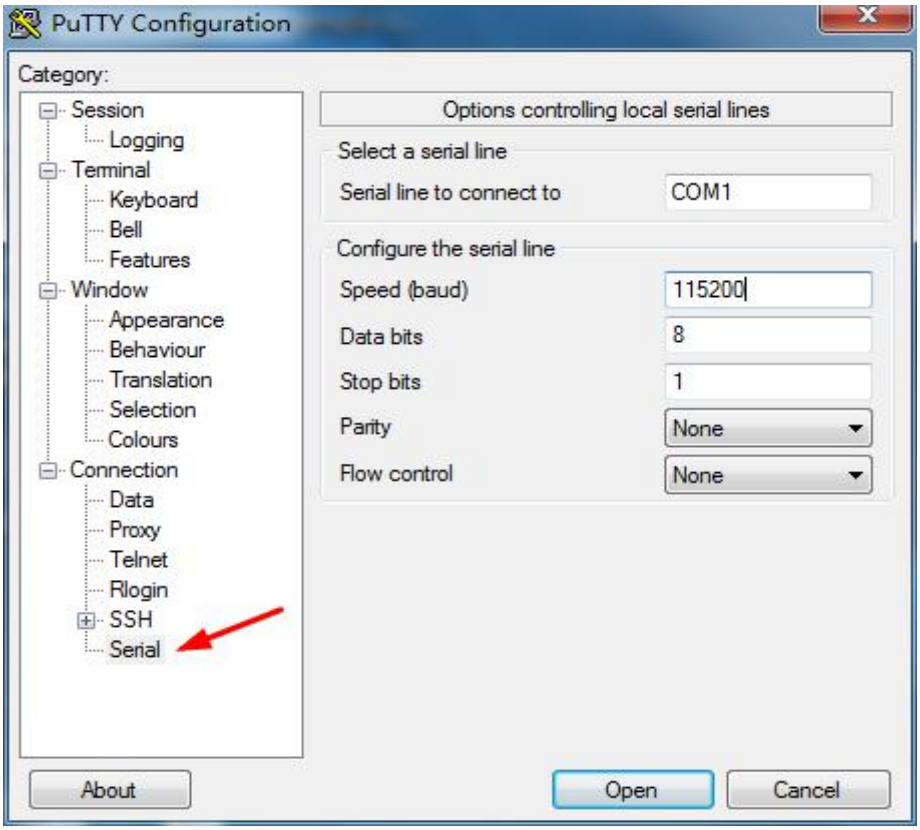
Yocto system support the following login methods: login directly, login from debug serial, login from SSH.

### 2.2.1 Login Directly

Connect screen and keyboard to MaaXBoard Osm93, username as “root”, to login Yocto system.

### 2.2.2 Login from Debug Serial

- 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 A55 debug interface to PC with USB to TTL converter. Pin 1, 3 and 5 of J10 to the TXD, RXD and GND pin of the USB to TTL converter.

```
NXP i.MX Release Distro 6.6-nanbield maaxboardosm93 ttyLP0
maaxboardosm93 login:
```

- Enter username as “root” to login.

### 2.2.3 Login from SSH

MaaXBoard Osm93 Yocto OS install and startup SSH service automatically by default. Connect to internet, then login the system using SSH.

Linux system support ssh in default, in windows OS, you can install ssh by yourself, or use other software which support ssh, such as PuTTY, WinSCP, etc.

#### 2.2.3.1 Preparation

Check the IP of MaaXBoard Osm93: The IP will be used in ssh login.

```
root@maaxboardosm93:~# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.98  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::230:d6ff:fe3b:b7eb  prefixlen 64  scopeid 0x20<link>
        ether 00:30:d6:3b:b7:eb  txqueuelen 1000  (Ethernet)
        RX packets 848  bytes 74009 (72.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 245  bytes 39424 (38.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

#### 2.2.3.2 Login Command line

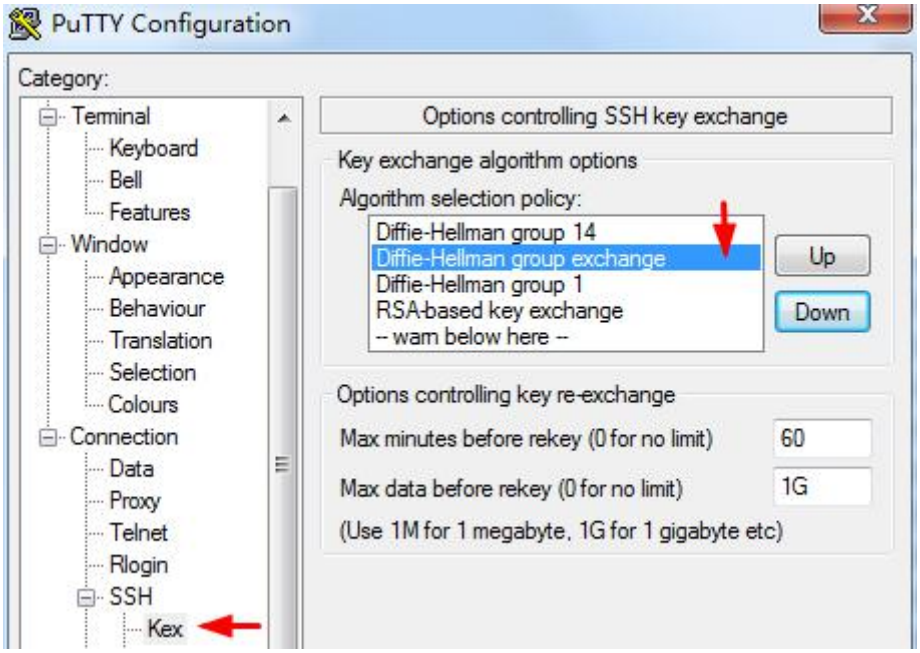
In this example, the IP of MaaXBoard Osm93 is ***192.168.2.98***, enter following command in command line window to connect: ssh root@192.168.2.98. Enter ***yes*** in the first connection, then will login to MaaXBoard Osm93, enter ***exit*** to logout.

```
$ ssh root@192.168.2.98
The authenticity of host '192.168.2.98 (192.168.2.98)' can't be established.
ED25519 key fingerprint is SHA256:6JrjXLt1U5GOzkPT0h+UgkFykxOpjO219TbDNd35f1A.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.98' (ED25519) to the list of known hosts.
Last login: Thu Jul  4 06:38:28 2024 from 192.168.2.203
root@maaxboardosm93:~#
```

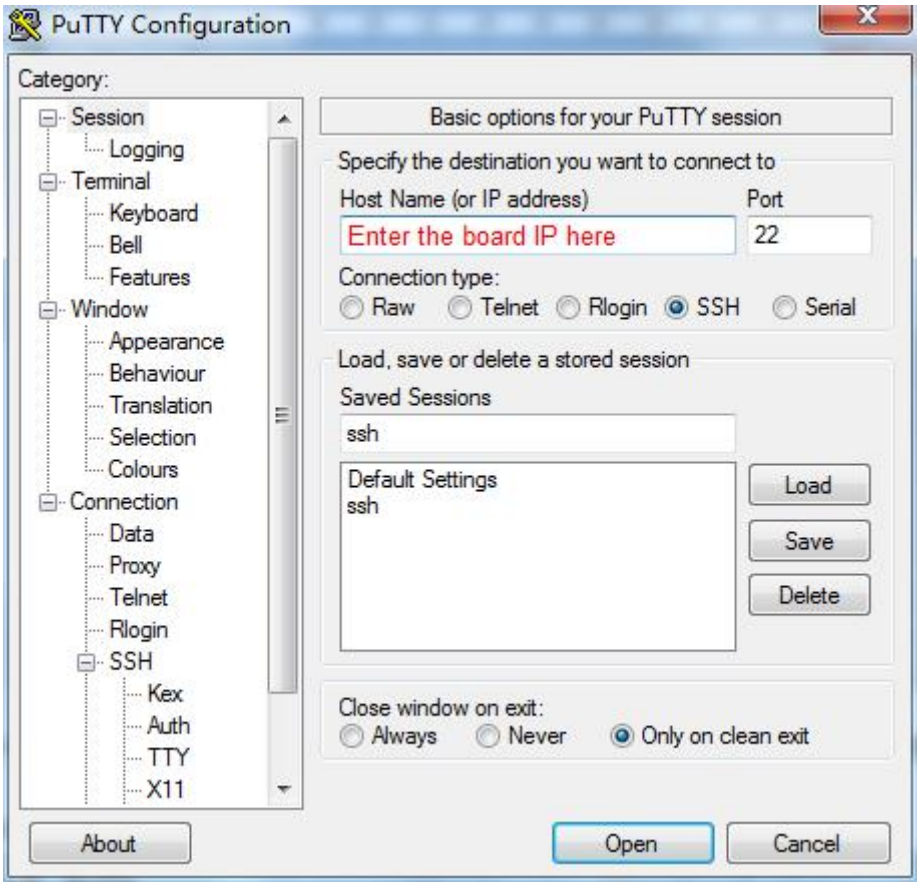
### 2.2.3.3 PuTTY

PuTTY support SSH, setting method as follows:

1. Run PUTTY, in Connection->SSH->Kex, change the **sequence** of algorithm.

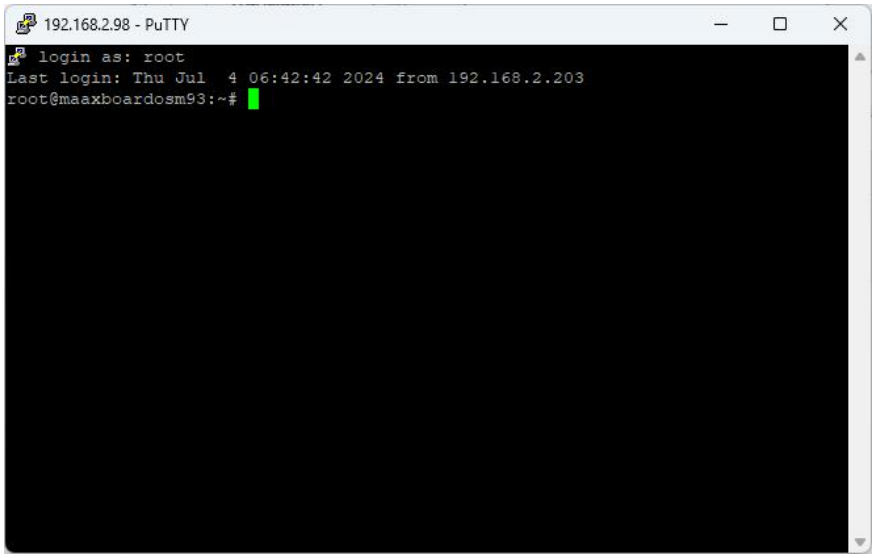


2. In Session, enter IP address, e.g. 192.168.2.98, port 22, and Connection type SSH, then click Open.




3. In the first connection, click Y in the popout window.

4. Enter username as “root” to login, enter **exit** to logout.



# Chapter 3 Feature Configuration & Introduction

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 Settings in uEnv.txt

User could configure some environment variables in uEnv.txt, which can be loaded in the U-boot stage. The uEnv.txt file has a very simple file format. The format is a single *property=value* statement on each line, where value is either an integer or a string. Comments may be added, or existing configuration values may be commented out and disabled, by starting a line with the **#** character.

The device-tree overlay function is supported from this version and the device-tree overlay file (*.dtbo) is placed in the overlay/ directory in the FAT partition of the eMMC. To load the device-tree overlay file (*.dtbo), you need to set ***fdt_file*** and ***dtoverlay_*** prefix variable in uEnv.txt. Also you could add other configurations defined in U-boot to the uEnv.txt file.

The specific description is as follows:

| Environment Variable | Value if Set (other invalid) | To be Loading in U-boot |
|---|---|---|
| dtoverlay_camera | ov5640 | camera-ov5640.dtbo |
| dtoverlay_display | mipi, mipi-ph720128t003 | display-{mipi,mipi-ph720128t003}.dtbo |
| dtoverlay_gpio | '1' or 'yes' | ext-gpio.dtbo |
| dtoverlay_i2c | '2" | ext-i2c2.dtbo |
| dtoverlay_spi | '1' | ext-spi1.dtbo |
| dtoverlay_wm8960 | '1' or 'yes' | ext-wm8960.dtbo |
| dtoverlay_extra | Other dtbo files to be loading, such as xxx.dtbo | |
| fdt_file | Board base dtb file, should be maaxboard-osm93.dtb | |
| console | Some u-boot environment variables | |

Note: ***fdt_file*** must be set to a device tree binary blob, which is the basis for applying dtbo file. ***fdt_file*** should be set, other configurations are optional.

Here is the default setting in uEnv.txt:

```
# Refer to readme.txt for more information on setting up U-Boot Env

# Device tree base file
fdt_file=maaxboard-osm93.dtb

# Camera can support ov5640
#dtoverlay_camera=ov5640
```

```
# Display can support mipi
#dtoverlay_display=mipi

# Set 40-pin extended GPIO pin default work as GPIO
dtoverlay_gpio=yes

# Enable lpi2c2 on 40-pin extended GPIO pin
#dtoverlay_i2c=2

# Enable lpspi1 on 40-pin extended GPIO pin
#dtoverlay_spi=1

# Enable wm8960 on 6-pin(J3) extended I2S
#dtoverlay_wm8960=yes

# Extra other device tree overlay
#dtoverlay_extra=1.dtbo 2.dtbo 3.dtbo

# U-boot bootargs for console
console=ttyLP0,115200 console=tty1
```

Modify ***uEnv.txt*** methods:

Mount the first partition of eMMC to the system, Then use **nano** or **vi** command to modify the uEnv.txt. After the modification, execute **sync** and **reboot** command to make it effect.

```
root@maaxboardosm93:~# mkdir mount
root@maaxboardosm93:~# mount /dev/mmcblk0p1 mount/
root@maaxboardosm93:~# vi mount/uEnv.txt
```

We can edit the uEnv.txt as needed and save it.

After the modification, execute **sync** and **reboot** commands to make it effect.

```
root@maaxboardosm93:~# sync
root@maaxboardosm93:~# reboot
```

## 3.2 USER LED

User can control a tricolour LED indicators, LED1 on MaaXBoard Osm93. Execute the following

instructions in serial terminal to control it.

Turn off the LED:

```
root@maaxboardosm93:~# echo 0 > /sys/class/leds/led_red/brightness
root@maaxboardosm93:~# echo 0 > /sys/class/leds/led_blue/brightness
root@maaxboardosm93:~# echo 0 > /sys/class/leds/led_green/brightness
```

Turn on a certain color of the tricolor LED:

```
root@maaxboardosm93:~# echo 1 > /sys/class/leds/led_red/brightness
root@maaxboardosm93:~# echo 1 > /sys/class/leds/led_blue/brightness
root@maaxboardosm93:~# echo 1 > /sys/class/leds/led_green/brightness
```

## 3.3 Button Switches

There are two push-button switches on MaaXBoard Osm93: USER and PWR.

1. Test USER button with following instructions:

Enter ***evtest*** command, then choose the event id for **gpio_keys**

```
root@maaxboardosm93:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      44440000.bbnsm:pwrkey
/dev/input/event1:      gpio_keys
/dev/input/event2:      fts_ts
Select the device event number [0-2]: 1
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio_keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 2 (KEY_1)
Properties:
Testing ... (interrupt to exit)
Event: time 1720076023.511763, type 1 (EV_KEY), code 2 (KEY_1), value 1
Event: time 1720076023.511763, ---------- SYN_REPORT ------------
Event: time 1720076023.799788, type 1 (EV_KEY), code 2 (KEY_1), value 0
Event: time 1720076023.799788, ---------- SYN_REPORT ------------
```

Use "*Ctrl+C*" to exit this test.

2. Press PWR button for 8s, system will enter suspend mode, press PWR again for 1s, the system will reboot.

Users could also test short press PWR button using **evtest** command:

```
root@maaxboardosm93:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      44440000.bbnsm:pwrkey
/dev/input/event1:      gpio_keys
/dev/input/event2:      fts_ts
Select the device event number [0-2]: 0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x0 product 0x0 version 0x0
Input device name: "44440000.bbnsm:pwrkey"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1720076206.279689, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1720076206.279689, ---------- SYN_REPORT ------------
Event: time 1720076206.535759, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1720076206.535759, ---------- SYN_REPORT ------------
```

Use "*Ctrl+C*" to exit this test.

## 3.4 Display Output

MaaXBoard Osm93 supports MIPI-DSI screen.

Users can connect the screen to MaaXBoard Osm93 before boot up the system according to the following table. When the system boot up, the screen will print the related startup message and login UI. Users can connect keyboard to login the MaaXBoard Osm93 file system.

| Screen Type | Screen Resolution | Interface |
|---|---|---|
| MIPI-DSI | 1280*720 | J4 |

Display device could be chosen by modify the fdt_file value in uEnv.txt.

**Modify methods:**

Mount the first partition of SD card or eMMC to the system, then use **nano** or **vi** command to modify the uEnv.txt. After the modification, execute **sync** and **reboot** command to make it effect.

```
root@maaxboardosm93:~# mkdir mount
root@maaxboardosm93:~# mount /dev/mmcblk0p1 mount/
root@maaxboardosm93:~# vi mount/uEnv.txt
```

### 3.4.1 MIPI-DSI Screen

MaaXBoard Osm93 supports two display models: *PH720128T003 & PH720128T005*

If you choose MIPI-DSI display and it's model# is *PH720128T005*, you should edit **uEnv.txt** as follows:

```
dtoverlay_display=mipi
```

If you choose MIPI-DSI display and it's model# is *PH720128T003*, you should edit uEnv.txt as follows:

```
dtoverlay_display=mipi-ph720128t003
```

MIPI-DSI supports backlight brightness adjustment. The backlight brightness has a range from 0 to 9, in which 9 means highest brightness, 0 means lowest.

Execute the following instructions on the serial terminal to implement the backlight test:

```
root@maaxboardosm93:~# echo 5 > /sys/class/backlight/pwm-backlight/brightness
root@maaxboardosm93:~# echo 9 > /sys/class/backlight/pwm-backlight/brightness
root@maaxboardosm93:~# echo 0 > /sys/class/backlight/pwm-backlight/brightness
```

## 3.5 Touchscreen

The MIPI-DSI screen support touch screen. Use evtest command to test it.

```
root@maaxboardosm93:~# evtest /dev/input/touchscreen0
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0
Input device name: "fts_ts"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 102 (KEY_HOME)
    Event code 139 (KEY_MENU)
    Event code 158 (KEY_BACK)
    Event code 330 (BTN_TOUCH)
```

```
  Event type 3 (EV_ABS)
    Event code 47 (ABS_MT_SLOT)
      Value      0
      Min        0
      Max        9
    Event code 48 (ABS_MT_TOUCH_MAJOR)
      Value      0
      Min        0
      Max      255
    Event code 53 (ABS_MT_POSITION_X)
      Value      0
      Min        0
      Max      720
    Event code 54 (ABS_MT_POSITION_Y)
      Value      0
      Min        0
      Max     1280
    Event code 57 (ABS_MT_TRACKING_ID)
      Value      0
      Min        0
      Max    65535
    Event code 58 (ABS_MT_PRESSURE)
      Value      0
      Min        0
      Max      255
Properties:
  Property type 1 (INPUT_PROP_DIRECT)
Testing ... (interrupt to exit)
Event: time 1720078287.684988, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 1
Event: time 1720078287.684988, type 3 (EV_ABS), code 58 (ABS_MT_PRESSURE), value 63
Event: time 1720078287.684988, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 7
Event: time 1720078287.684988, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 442
Event: time 1720078287.684988, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 531
Event: time 1720078287.684988, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
Event: time 1720078287.684988, ---------- SYN_REPORT ------------
Event: time 1720078287.695191, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 8
Event: time 1720078287.695191, ---------- SYN_REPORT ------------
```

Use "***Ctrl+C***" to exit this test.

## 3.6 Audio

MaaXBoard Osm93 supports USB audio devices, Bluetooth audio devices and provides one channel I2S Audio. If multiple devices are connected simultaneously, the priority is as follow:

I2S audio device < USB audio device < Bluetooth audio device

### 3.6.1 Check Audio Device IDs

Before playing or recording an audio interface, you should check the device ID.

Use the aplay -l and arecord -l commands to list the audio playback- and record- device IDs.

#### 3.6.1.1 I2S Audio Device

Refer to 3.14.4 to connect and enable the I2S audio player, it will play the sound from I2S audio device by default.

```
root@maaxboardosm93:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: wm8960soundaudi [wm8960-sound-audio], device 0: 443b0000.sai-wm8960-hifi
wm8960-hifi-0 [443b0000.sai-wm8960-hifi wm8960-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

#### 3.6.1.2 USB Audio Device

MaaXBoard Osm93 supports USB audio device (which do not need specified driver) to play audio. When using MIPI-DSI screens, you can play audio from USB audio device.

```
root@maaxboardosm93:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 1: Seri [Plantronics Blackwire 3225 Seri], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

#### 3.6.1.3 Bluetooth Audio Device

Yocto system also supports playing audio files via the Bluetooth audio device such as Bluetooth headset. For detail, refer to Bluetooth part.

### 3.6.2 Record Audio

There is a digital microphone with two channels on MaaXBoard Osm93 board, Use command ***arecord -l*** to check the device ID.

```
root@maaxboardosm93:~# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: micfilaudio [micfil-audio], device 0: micfil hifi snd-soc-dummy-dai-0 [micfil hifi
snd-soc-dummy-dai-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Use the following command to record audio to file audio_sample.wav:

```
root@maaxboardosm93:~# arecord -c 2 -f S32_LE -r 48000 audio_sample.wav -D hw:0,0
```

Note: press **Ctrl+C** ro exit record.

In the above command:

**S32_LE** = audio format

**-r 48000** = sample rate of the audio file (48KHz),

**-c 2** = 2 channel audio recording,

**-Dhw:0** = use audio card 0 to record,

Note: Change those parameters according to your device.

### 3.6.3 Play Audio file

After a playback device is successfully connected, run the following command to start playing the audio.

```
root@maaxboardosm93:~# pulseaudio -D -v
root@maaxboardosm93:~# aplay audio_sample.wav
root@maaxboardosm93:~# gst-play-1.0 audio_sample.wav
root@maaxboardosm93:~# mpg123 audio_sample1.mp3
```

The aplay command supports audio file in wav format, gst-play-1.0 command supports wav, mp3 and aac format, while the mpg123 command supports mp3 format.

When using above command. Audio will play from the default device.

## 3.7 Video

Yocto system support play video file in mp4 format, the largest support resolution is 4K, Use **gplay-1.0** or **gst-launch-1.0** command to play video files.

Connect MaaXBoard Osm93 to MIPI-DSI screen display, and edit the ***uEnv.txt*** with the corresponding value to dtoverlay_display.

Take an example to play the video.mp4 file, select one of the following four commands and enter it in the serial terminal.

```
root@maaxboardosm93:~# gplay-1.0 video.mp4
root@maaxboardosm93:~# gplay-1.0 --video-sink=waylandsink video.mp4
root@maaxboardosm93:~# gst-launch-1.0 playbin uri=file:///home/root/video.mp4
root@maaxboardosm93:~# gst-launch-1.0 filesrc location=video.mp4 typefind=true !
video/quicktime ! aiurdemux ! queue max-size-time=0 ! vpudec ! autovideosink
```

## 3.8 Camera

MaaXBoard Osm93 supports USB Camera and MIPI-CSI Camera. This part will introduce how to preview, photograph and record video under Command line.

To use the MIPI-CSI camera, the ***dtoverlay_camera*** value should be set:

```
dtoverlay_camera=ov5640
```

Note: To show the camera preview on the desktop, it is recommended that the dtoverlay_display option should be set in ***uEnv.txt***.

### 3.8.1 Check Device ID

```
root@maaxboardosm93:~# ls /dev/video*
/dev/video0   /dev/video1
```

In default, MIPI-CSI camera is /dev/video0, USB Camera is /dev/video1. The device ID will be used in following command.

### 3.8.2 Preview

Use the following instruction to open Camera and preview the video on the screen.

```
root@maaxboardosm93:~# gst-launch-1.0 v4l2src device=/dev/video0 ! autovideosink
```

Note: Press **Ctrl+C** to exit, change /dev/video0 to your device ID.

### 3.8.3 Take Photo

Use the following instruction to take a photo and saved to specific location.

```
gst-launch-1.0 v4l2src device=[video] num-buffers=1 ! jpegenc ! filesink location=[filename]
```

In above command, replace [video] to the camera device ID, filename] to the path and name of saved file. For example:

```
root@maaxboardosm93:~# gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=1 ! jpegenc ! filesink location=sample.jpg
```

Copy the photo to other device, such as computer to display it.

### 3.8.4 Record Video

Use the following instruction to record a video in mp4 format and saved to specific location.

```
root@maaxboardosm93:~# gst-launch-1.0 -e v4l2src device=/dev/video0 num-buffers=100 ! video/x-raw,format=YUY2,framerate=30/1, width=640, height=480 ! videoconvert ! x264enc ! video/x-h264, profile=baseline ! mp4mux ! filesink location=output.mp4
```

In above command, modify the camera device ID, the width and height of the video, the path and name of saved file, etc. The video file can be copy to other device, such as computer to display, or use gst-play-1.0 to display it on the screen directly.

```
root@maaxboardosm93:~# gst-play-1.0 output.mp4
```

## 3.9 Gigabit Ethernet Interface

Connect the network cable to J5(eth0) or J2(eth1), enter the following instructions to set the IP address:

The below IP address are example, replace it with your real network environment

### 3.9.1 Network Test

After connecting the network cable, it 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(Take eth0 as an example):

```
root@maaxboardosm93:~# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.223  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::230:d6ff:fe3b:b7eb  prefixlen 64  scopeid 0x20<link>
        ether 00:30:d6:3b:b7:eb  txqueuelen 1000  (Ethernet)
        RX packets 205044  bytes 17230879 (16.4 MiB)
```

```
        RX errors 0   dropped 0   overruns 0   frame 0
        TX packets 7844   bytes 360602 (352.1 KiB)
        TX errors 0   dropped 0 overruns 0   carrier 0   collisions 0
root@maaxboardosm93:~# ping www.avnet.com
PING www.avnet.com (184.50.90.49) 56(84) bytes of data.
64 bytes from a184-50-90-49.deploy.static.akamaitechnologies.com (184.50.90.49): icmp_seq=1
ttl=55 time=25.7 ms
64 bytes from a184-50-90-49.deploy.static.akamaitechnologies.com (184.50.90.49): icmp_seq=2
ttl=55 time=25.2 ms
```

Note: Press **Ctrl+C** to exit

### 3.9.2 Set Static IP

If you need to set a static IP, execute the following commands(take eth0 as an example):

```
root@maaxboardosm93:~# vi /etc/systemd/network/01-eth0.network
[Match]
Name=eth0

[Network]
Address=192.168.2.77/24
Gateway=192.168.2.1
DNS=8.8.8.8
DNS=114.114.114.114

root@maaxboardosm93:~# systemctl restart systemd-networkd
```

In above command, replace the IP address, router, DNS with your real network environment.

### 3.9.3 Set Dynamic IP

Execute the following commands to set dynamic IP(take eth0 as an example):

```
root@maaxboard:~# vi /etc/systemd/network/01-eth0.network
[Match]
Name=eth0

[Network]
DHCP=yes
root@maaxboard:~# systemctl restart systemd-networkd
```

## 3.10 USB 2.0 Interface

MaaXBoard Osm93 supports four USB 2.0 interfaces, 2x host type A, 1x host type-C, 1x device type-C.

### 3.10.1 USB Host Type A

The USB hub J9 has 2 USB Host type A Interfaces, the upper one is USB1, the lower one is USB0.

Insert a U-disk to USB0 or USB1, serial terminal will display the disk information:

```
[59907.423739] usb 4-1: new SuperSpeed USB device number 2 using xhci-hcd
[59907.458805] usb-storage 4-1:1.0: USB Mass Storage device detected
[59907.466534] scsi host0: usb-storage 4-1:1.0
[59908.493894] scsi 0:0:0:0: Direct-Access     Generic  MassStorageClass 1536 PQ: 0 ANSI: 6
[59908.839824] sd 0:0:0:0: [sda] 30449664 512-byte logical blocks: (15.6 GB/14.5 GiB)
[59908.850009] sd 0:0:0:0: [sda] Write Protect is off
[59908.857068] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[59908.870413]  sda:
[59908.874223] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

Execute the following instructions on the serial terminal:

```
root@maaxboard:~# ls /dev/sd*
/dev/sda  /dev/sdb  /dev/sdb1  /dev/sdb2
```

The storage node for U disk is /dev/sda, users could mount the storage device to the file system to read and write data.

### 3.10.2 USB Host Type C

MaaXBoard Osm93 has one USB Host type A Interface on board(J12), when connect a USB to it, serial terminal will display the disk information:

```
root@maaxboardosm93:~# [ 3549.552609] usb 1-1.1: new high-speed USB device number 5 using ci_hdrc
[ 3549.667377] usb-storage 1-1.1:1.0: USB Mass Storage device detected
[ 3549.674316] scsi host0: usb-storage 1-1.1:1.0
[ 3551.117356] scsi 0:0:0:0: Direct-Access                SD Card Reader   1.00 PQ: 0 ANSI: 6
[ 3551.127257] sd 0:0:0:0: [sda] 122142720 512-byte logical blocks: (62.5 GB/58.2 GiB)
[ 3551.139148] sd 0:0:0:0: [sda] Write Protect is off
[ 3551.145118] sd 0:0:0:0: [sda] No Caching mode page found
[ 3551.150509] sd 0:0:0:0: [sda] Assuming drive cache: write through
```

```
[ 3551.162111]  sda: sda1
[ 3551.165178] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

Execute the following instructions on the serial terminal:

```
root@maaxboardosm93:~# ls /dev/sd*
/dev/sda  /dev/sda1
```

MaaXBoard Osm93 also supports other USB device such as key board, mouse, Camera, etc.

## 3.11 Wi-Fi

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

### 3.11.1 Connect Wi-Fi Manually

Execute the following instructions on the serial terminal to search Wi-Fi network, It will print the information for all available network:

```
root@maaxboardosm93:~# ifconfig mlan0 up
root@maaxboardosm93:~# iwlist mlan0 scan | grep ESSID
                    ESSID:"TEST23"
                    ESSID:"AAAA_2.4G" [2]
                    ESSID:"Development " [3]
                    ESSID:"XXXX-5G" [4]
```

Configure SSID and SSID_PASSWD with the following command: (take "TEST23" as an example)

```
root@maaxboardosm93:~# wpa_passphrase "TEST23" "12345678" >> /etc/wpa_supplicant.conf
```

Or edit /etc/wpa_supplicant.conf directly and append the following parameters:

```
root@maaxboardosm93:~# vi /etc/wpa_supplicant.conf
```

Add following info into this file:

```
network={
    ssid="TEST23"
    psk="12345678"
}
```

Then execute the following command:

```
root@maaxboardosm93:~# wpa_supplicant -B -i mlan0 -c /etc/wpa_supplicant.conf
Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
```

```
rfkill: Cannot get wiphy information
```

Run the following command to check wireless connection, it will print the following info once connected:

```
root@maaxboardosm93:~# iwconfig mlan0 | grep ESSID
wlan0     IEEE 802.11 ESSID:"TEST23"
root@maaxboardosm93:~# ifconfig mlan0
```

Test Wi-Fi network with ping command:

```
root@maaxboardosm93:~# ping www.avnet.com -I mlan0
PING www.avnet.com (184.50.90.49) from 192.168.2.181 mlan0: 56(84) bytes of data.
64 bytes from a184-50-90-49.deploy.static.akamaitechnologies.com (184.50.90.49): icmp_seq=1
ttl=55 time=38.0 ms
64 bytes from a184-50-90-49.deploy.static.akamaitechnologies.com (184.50.90.49): icmp_seq=2
ttl=55 time=50.0 ms
```

Use **Ctrl+C** to exit this test.

### 3.11.2 Wi-Fi Hotspot

Use the following steps to configure and start the 2.4 GHz/5 GHz Access Point from the wireless module.

Make sure the Wi-Fi is disconnected:

```
root@maaxboardosm93:~# killall wpa_supplicant
root@maaxboardosm93:~# killall hostapd
```

Then use the following steps to set up Wi-Fi hotspot.

Edit the configuration file for hostapd:

```
root@maaxboardosm93:~# vi /etc/hostapd-5g.conf
```

Parameter values in the configuration file:

```
interface=uap0
# specify the band: hw_mode=g (2.4 GHz) and hw_mode=a (5 GHz)
hw_mode=a
channel=0
country_code=US
ssid=MY_HOSTAP
ieee80211n=1
```

Note: If you want to configure the 5 GHz Access Point, change hw_mode=a and default is 2.4 GHz AP.

If you want to configure WPA2 for the AP using open source supplicant, need to add the following additional lines:

```
wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
wpa_passphrase=123456789
```

Note: You can modify your ssid and wpa_passphrase in hostapd.conf file.

Create the configuration file for udhcp server:

```
root@maaxboardosm93:~# vi /etc/udhcpd.conf
```

Add the following content to udhcpd.conf file:

```
interface uap0
start 192.168.6.10
end 192.168.6.100
opt router 192.168.6.1
opt dns 114.114.114.114 8.8.8.8
```

Note: The IP address 192.168.6.x can be modified at will but it must be consistent with its related IP.

Command to start the 5 GHz Access Point and start udhcp server to assign the IP address:

```
root@maaxboardosm93:~# ifconfig uap0 192.168.6.1 netmask 255.255.255.0 up
root@maaxboardosm93:~# udhcpd /etc/udhcpd.conf
root@maaxboardosm93:~# hostapd -B /etc/hostapd-5g.conf
```

At this time, you can use other devices to scan the access point MY_HOSTAP, and enter the password 123456789 to connect. After obtaining the IP address, the device will display a status of Connected, no Internet.

If the Ethernet interface is connected to the Internet, you can use the following commands to add packet forwarding rules so that the devices connected to the hotspot can access the Internet.

```
root@maaxboardosm93:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@maaxboardosm93:~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
root@maaxboardosm93:~# iptables -A FORWARD -i eth0 -o uap0 -m state --state RELATED,ESTABLISHED -j ACCEPT
root@maaxboardosm93:~# iptables -A FORWARD -i uap0 -o eth0 -j ACCEPT
```

# 3.12 Bluetooth 5.3

## 3.12.1 Initialize the Bluetooth Module

Execute the following instructions on the serial terminal:

```
root@maaxboardosm93:~# hciattach /dev/ttyLP2 any 115200
Device setup completee
root@maaxboardosm93:~# hciconfig hci0 up
```

## 3.12.2 Connect Bluetooth Device

Use bluetoothctl to connect Bluetooth Device:

```
root@maaxboardosm93:~# bluetoothctl
[bluetooth]# power on
[bluetooth]# pairable on
[bluetooth]# agent on
[bluetooth]# default-agent
```

Make the MaaXBoard Osm93 discoverable by other Bluetooth device:

```
[bluetooth]# discoverable on
```

Enable and Disable Scan:

```
[bluetooth]# scan on
[bluetooth]# scan off
```

Pair and connect the device:

```
[bluetooth]# pair E8:EC:A3:21:57:6C
[bluetooth]# trust E8:EC:A3:21:57:6C
[bluetooth]# connect E8:EC:A3:21:57:6C
[device name]# disconnect E8:EC:A3:21:57:6C
```

Exit bluetoothctl.

```
[bluetooth]# exit
```

In above instructions, E8:EC:A3:21:57:6C is the address of the Bluetooth device, change it according to your device.

## 3.12.3 Configure A2DP sink or source

When connected to a remote Bluetooth device that supports the A2DP sink feature, MaaXBoard Osm93

can be configured as an A2DP Source. This type of bluetooth devices are usually bluetooth headsets, bluetooth speakers, etc.

Set the baud rate to 3M bps,

```
root@maaxboardosm93:~# hcitool -i hci0 cmd 0x3f 0x0009 0xc0 0xc6 0x2d 0x00
< HCI Command: ogf 0x3f, ocf 0x0009, plen 4
  C0 C6 2D 00
> HCI Event: 0x0e plen 4
  01 09 FC 00

root@maaxboardosm93:~# killall hciattach
[   132.691727] Bluetooth: hci0: sending frame failed (-49)

root@maaxboardosm93:~# hciattach /dev/ttyLP2 any -s 3000000 3000000 flow
Setting TTY to N_HCI line discipline
Device setup complete

root@maaxboardosm93:~# hciconfig hci0 up
root@maaxboardosm93:~# hciconfig
hci0:   Type: Primary   Bus: UART
        BD Address: 70:66:55:8A:AE:B2   ACL MTU: 1016:5   SCO MTU: 60:12
        UP RUNNING PSCAN
        RX bytes:798 acl:0 sco:0 events:53 errors:0
        TX bytes:2028 acl:0 sco:0 commands:53 errors:0
```

Run the following command to verify the Audio Sink Profile capability of the connected Bluetooth device:

```
root@maaxboardosm93:~# pulseaudio -D -v
root@maaxboardosm93:~# bluetoothctl info
```

Command output example showing the feature:

```
UUID: Audio Sink                (0000110b-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control        (0000110e-0000-1000-8000-00805f9b34fb)
        UUID: PnP Information           (00001200-0000-1000-8000-00805f9b34fb)
        UUID: Generic Access Profile    (00001800-0000-1000-8000-00805f9b34fb)
        UUID: Generic Attribute Profile (00001801-0000-1000-8000-00805f9b34fb)
        Modalias: usb:v1D6Bp0246d0525
```

Connect Bluetooth device:

```
root@maaxboardosm93:~# bluetoothctl
[bluetooth]# power on
[bluetooth]# pairable on
[bluetooth]# default-agent
[bluetooth]# scan on
[NEW] Device 8C:53:C3:21:8A:EE device-name
[bluetooth]# pair 8C:53:C3:21:8A:EE
Attempting to pair with 8C:53:C3:21:8A:EE
[CHG] Device 8C:53:C3:21:8A:EE Connected: yes
[CHG] Device 8C:53:C3:21:8A:EE UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device 8C:53:C3:21:8A:EE UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 8C:53:C3:21:8A:EE UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 8C:53:C3:21:8A:EE UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 8C:53:C3:21:8A:EE UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 8C:53:C3:21:8A:EE UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 8C:53:C3:21:8A:EE ServicesResolved: yes
[CHG] Device 8C:53:C3:21:8A:EE Paired: yes
Pairing successful
[bluetooth]# scan off
[bluetooth]# connect 8C:53:C3:21:8A:EE
Attempting to connect to 8C:53:C3:21:8A:EE
[CHG] Device 8C:53:C3:21:8A:EE Connected: yes
Connection successful
[device-name]# exit
root@maaxboardosm93:~#
```

Play the audio file using pulseaudio play utility, the music will be played from the connected Bluetooth device:

```
root@maaxboardosm93:~# gst-play-1.0 audio_sample.wav
```

### 3.12.4 Send Files

Run the OBEXD daemon and connect to the target Bluetooth device

```
root@maaxboardosm93:~# echo hello > 1.txt
root@maaxboardosm93:~# export $(dbus-launch)
root@maaxboardosm93:~# /usr/libexec/bluetooth/obexd -r /home/root -a -d & obexctl
[obex]# connect 94:87:E0:DF:90:2D
[94:87:E0:DF:90:2D]# send /home/root/1.txt
Attempting to send /home/root/1.txt to /org/bluez/obex/client/session2
```

```
[NEW] Transfer /org/bluez/obex/client/session2/transfer1
Transfer /org/bluez/obex/client/session2/transfer1
        Status: queued
        Name: 1.txt
        Size: 6
        Filename: /home/root/1.txt
        Session: /org/bluez/obex/client/session2
[CHG] Transfer /org/bluez/obex/client/session2/transfer1 Status: complete
[94:87:E0:DF:90:2D]# exit
```

In above instructions, 94:87:E0:DF:90:2D is the address of target device, change it according to your device.

## 3.13 UART

MaaXBoard Osm93 supports 2 UART interface.

| MaaXBoard Osm93 (CPU) | Interface Type |
|---|---|
| UART_M33 | UART TTL (M33 Debug Interface) |
| UART_A55 | UART TTL (A55 Debug Interface) |

## 3.14 CAN

MaaXBoard Osm93 provides 2 channels CAN on the board, execute the following commands to enable and set the baud rate for CAN:

```
root@maaxboardosm93:~# ip link set can0 down
root@maaxboardosm93:~# ip link set can1 down
root@maaxboardosm93:~# ip link set can0 type can bitrate 500000
root@maaxboardosm93:~# ip link set can1 type can bitrate 500000
root@maaxboardosm93:~# ip link set can1 up
root@maaxboardosm93:~# ip link set can0 up
```

After the CANs are enabled, short connect the pins on J13: CAN_A_H with CAN_B_H, CAN_A_L with CAN_B_L;

Open a serial terminal, execute the following commands to receive data:

```
root@maaxboardosm93:~# candump can1
```

Open another serial terminal, execute the following commands to receive data:

```
root@maaxboardosm93:~# cansend can0 123#0102030405060708
```

The receiver can correctly receive the data sent by the sender.

## 3.15 ADC

MaaXBoard Osm93 provides 2 channels ADC on the board. The following uses ADC0 as an example to describe how to test and view the results of ADC：

1. Connect ADC_0(J6.3), VDD1V8(J6.1), GND(J6.2) to a sliding rheostatic test device;
2. Turn the knob of the sliding rheostatic test device;
3. Execute the following command to read the value of ADC0:

```
root@maaxboardosm93:# cd /sys/bus/iio/devices/iio:device0
root@maaxboardosm93:/sys/bus/iio/devices/iio:device0# cat in_voltage0_raw
```

## 3.16 Control 40 Pin Interface

This chapter will provide the Control methods of 40 Pin interface, include GPIO, I2C and SPI.

### 3.16.1 GPIO

System use /sys/class/gpio to control the GPIO pin, refer to the following table:

Table: GPIO corresponding relation table

| GPIO number | PINMUX | Function | PIN | PIN | Function | PINMUX | GPIO number |
|---|---|---|---|---|---|---|---|
| | | 3.3V | 1 | 2 | 5V | | |
| | I2C_B_SDA | SDA2 | 3 | 4 | 5V | | |
| | I2C_B_SCL | SCL2 | 5 | 6 | GND | | |
| 648 | GPIO_B_0 | GPIO | 7 | 8 | UART_TX | UART_M33_TXD | |
| | | GND | 9 | 10 | UART_RX | UART_M33_RXD | |
| | NC | NC | 11 | 12 | GPIO | GPIO_B_2 | 650 |
| 544 | GPIO3_IO0 | GPIO | 13 | 14 | GND | | |
| 551 | GPIO3_IO7 | GPIO | 15 | 16 | GPIO | GPIO_B_3 | 651 |
| | | 3.3V | 17 | 18 | GPIO | GPIO_B_4 | 652 |
| 522 | GPIO2_IO10 | MOSI | 19 | 20 | GND | | |
| 521 | GPIO2_IO9 | MISO | 21 | 22 | GPIO | GPIO_B_5 | 653 |
| 523 | GPIO2_IO11 | SCLK | 23 | 24 | SPI3_CS0 | GPIO2_IO8 | 520 |
| | | GND | 25 | 26 | GPIO | GPIO_B_6 | 654 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 512 | GPIO2_IO0 | SDA_LCD | 27 | 28 | SCL_LCD | GPIO2_IO1 | 513 |
| 547 | GPIO3_IO3 | GPIO | 29 | 30 | GND | | |
| 548 | GPIO3_IO4 | GPIO | 31 | 32 | GPIO | GPIO3_IO27 | 571 |
| 549 | GPIO3_IO5 | GPIO | 33 | 34 | GND | | |
| 550 | GPIO3_IO6 | GPIO | 35 | 36 | GPIO | NC | |
| 545 | GPIO3_IO1 | GPIO | 37 | 38 | GPIO/PCM_DIN | GPIO3_IO2 | 546 |
| | | GND | 39 | 40 | GPIO | GPIO_B_7 | 655 |

Here we take PIN35(GPIO4_IO21) as an example:

1. In above table, the GPIO Number of connector PIN35 is 550.
2. Set the function of Pin35 to be GPIO output.

```
root@maaxboardosm93:~# echo 550 >/sys/class/gpio/export
root@maaxboardosm93:~# echo out >/sys/class/gpio/gpio550/direction
```

3. Set the level of Pin35, 0 means low, 1 means high.

```
root@maaxboardosm93:~# echo 1 >/sys/class/gpio/gpio550/value
```

### 3.16.2 SPI

To use the SPI of the 40-pins interface, enable the following options in uEnv.txt :

Add dtoverlay_spi=1 to uEnv.txt, then execute sync and reboot command to make it effect.

Short connect SPI_MOSI(#19) and SPI_MISO(#21), then execute spidev_test, the result:

```
root@maaxboardosm93:~# spidev_test -D /dev/spidev0.0 -v
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 kHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
F0 0D   |......@.....................|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
F0 0D   |......@.....................|
```

Disconnect SPI_MOSI(#19) and SPI_MISO(#21), then execute spidev_test, the result:

```
root@maaxboardosm93:~# spidev_test -D /dev/spidev0.0 -v
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 kHz)
```

```
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
F0 0D    |......@.....................|
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF    |.............................|
```

### 3.16.3 IIC

To use the IIC of the 40-pins interface, enable the following options in uEnv.txt :

Add ***dtoverlay_i2c=2*** to uEnv.txt, then execute ***sync*** and ***reboot*** command to make it effect.

Connect SDA2 (#3), SCL2 (#5), VCC and GND to a IIC device, then execute i2cdetect, MaaXBoard Osm93 can detect the IIC device:

```
root@maaxboardosm93:~# i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- UU -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

### 3.16.4 Extend SAI WM8960

1. Connect the test WM8960 Audio HAT to MaaXBoard Osm93 board as described in the following table:

| Pins of WM8960 Audio HAT | | | Pins of MaaXBoard Osm93 | |
|---|---|---|---|---|
| Pin Label | Pin Name | | Pin Label | Pin Name |
| P1.1 | 3.3V | ⟷ | J1.1 | 3.3V |
| P1.2 | 5V | ⟷ | J1.2 | 5V |
| P1.3 | I2C SDA | ⟷ | J1.3 | I2C_HAT_SDA |
| P1.5 | I2C SCL | ⟷ | J1.5 | I2C_HAT_SCL |
| P1.12 | I2S_CLK | ⟷ | J3.4 | I2S_BCLK_3V3 |
| P1.35 | I2S_LRCLK | ⟷ | J3.3 | I2S_LRCLK_3V3 |
| P1.38 | I2S_ADC | ⟷ | J3.6 | I2S_A_DATA_DIN_3V3 |
| P1.39 | GND | ⟷ | J3.2 | GND |
| P1.40 | I2S_DAC | ⟷ | J3.5 | I2S_A_DOUT_3V3 |

2. Add ***dtoverlay_wm8960=yes*** to uEnv.txt, then execute ***sync*** and ***reboot*** command to make it effect.

After WM8960 is enabled, You can view it by using command ***aplay -l*** and use it to play audio.

## 3.17 Gopoint

GoPoint for i.MX Applications Processors is a user-friendly application that allows the user to launch preselected demonstrations included in the NXP provided Linux Board Support Package (BSP).

MaaXBoard OSM93 board supports the GoPoint for i.MX Applications Processors.

This section describes the demo launcher and the demos included in the MaaXBoard OSM93 board simply and generally, For details, please refer to the documentation *GoPoint for i.MX Applications Processors User Guide*(document***GPNTUG***).

### 3.17.1 Preparation

#### 3.17.1.1 Change the host name

After maaxboard-osm93 board is powered on, run the following command to change the host name of the board to: ***imx93-11x11-lpddr4x-evk***

```
sudo hostname imx93-11x11-lpddr4x-evk
```

#### 3.17.1.2 Download the required model files

Download the required model files of gopoint from github of NXP, and transmit them to /home/root/.cache/gopoint/

```
git clone https://github.com/nxp-imx-support/nxp-demo-experience-assets.git
cp nxp-demo-experience-assets/models/* /home/root/.cache/gopoint/
```

### 3.17.2 Demo launcher

#### 3.17.2.1 Graphical user interface

On boards where the GoPoint for i.MX Applications Processors is available, an NXP logo is displayed on the top left-hand corner of the screen. Users can start the demo launcher by clicking this logo.

After opening the program, users can launch demos using the following options shown in the following figure:

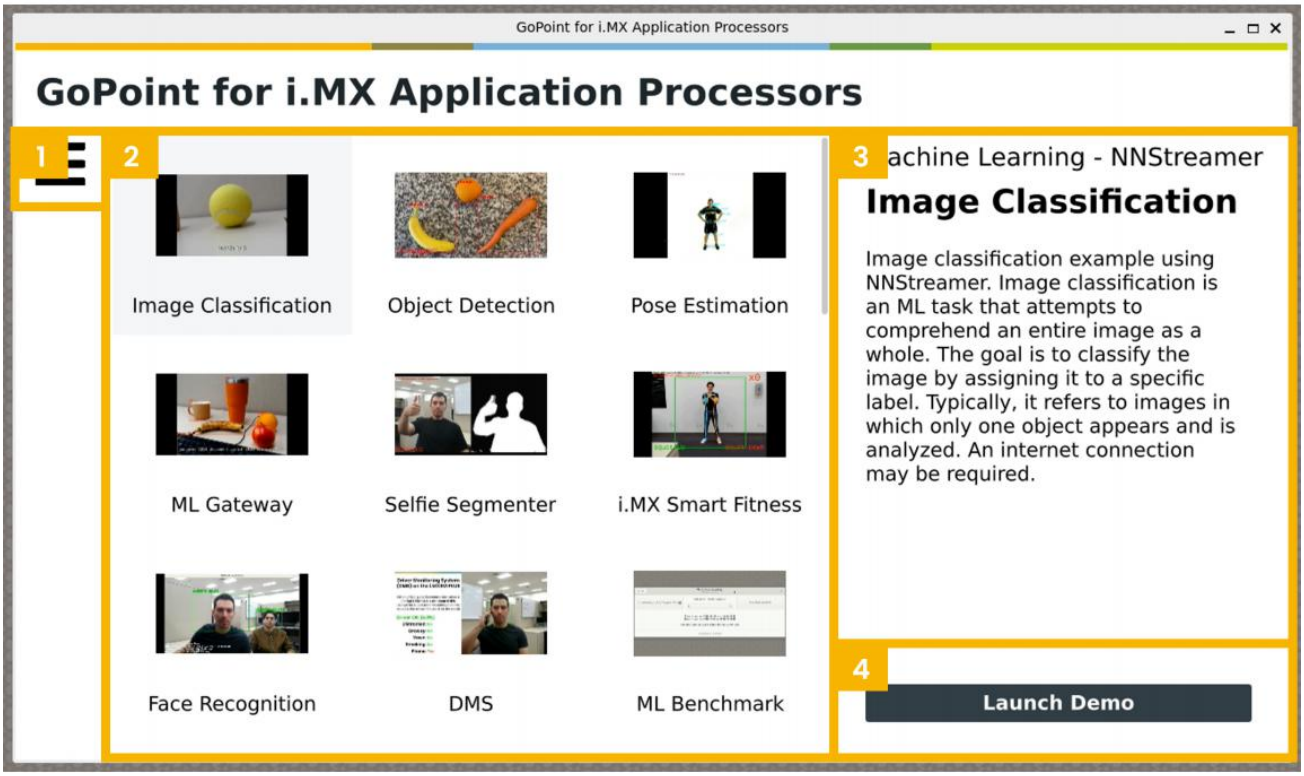
1. To filter the list, select the icon on the left to expand the filter menu. From this menu, users can select a category or subcategory that filters the demos displayed in the launcher.

2. A scrollable list of all the demos supported on that EVK appears in this area with any filters applied. Clicking a demo in the launcher brings up information about the demo.

3. This area displays the names, categories, and description of the demos.

4. Clicking Launch Demo launches the currently selected demo. A demo can then be force-quit by clicking the Stop current demo button in the launcher (appears once a demo is started).

*Note:* Only one demo can be launched at a time.




### 3.17.2.2 Text user interface

Demos can also be launched from the command line through log-in into the board remotely or using the onboard serial debug console. Keep in mind that most demos still require a display to run successfully.

Note: If prompted for a login, the default user name is "root" and no password is required.

To start the text user interface (TUI), type the following command into the command line:

```
# gopoint tui
```


The interface can be navigated using the following keyboard inputs:

• **Up and down arrow keys:** Select a demo from the list on the left

• **Enter key:** Runs the selected demo

• **Q key or Ctrl+C keys:** Quit the interface

• **H key:** Opens the help menu

Demos can be closed by closing the demo onscreen or pressing the "Ctrl" and "C" keys at the same time.

## 3.17.3 Included demos

### 3.17.3.1 LP Baby Cry Detection

If you want to test this application, you need to change the environment variables in uboot as following:

```
Normal Boot
Hit any key to stop autoboot:  0
u-boot=> setenv mmcargs setenv bootargs console=ttyLP0,115200 console=tty1
root=/dev/mmcblk0p2 rootwait rw clk_ignore_unused
u-boot=> boot
```

After starting the demo, the A-core goes into sleep mode, the terminal prints the following log:

The printing of the A-core serial terminal:

```
Starting LP Baby Cry Detection!
Quit the demo (Ctrl-C) to get back to demo select.

Start demo...
[   29.163462] remoteproc remoteproc0: powering up imx-rproc
[   29.169073] remoteproc remoteproc0: Direct firmware load for /home/root/.cache/gopoint/lp_baby_detection.elf failed with error -2
[   29.182045] remoteproc remoteproc0: Falling back to sysfs fallback for: /home/root/.cache/gopoint/lp_baby_detection.elf
[   29.194927] remoteproc remoteproc0: Booting fw image /home/root/.cache/gopoint/lp_baby_detection.elf, size 286016
[   29.726529] rproc-virtio rproc-virtio.1.auto: assigned reserved memory node vdevbuffer@a4020000
[   29.838407] imx-rproc remoteproc-cm33: imx_rproc_kick: failed (0, err:-62)
[   29.845310] virtio_rpmsg_bus virtio0: rpmsg host is online
[   29.850924] rproc-virtio rproc-virtio.1.auto: registered virtio0 (type 7)
[   29.857784] rproc-virtio rproc-virtio.2.auto: assigned reserved memory node vdevbuffer@a4020000
[   29.970412] imx-rproc remoteproc-cm33: imx_rproc_kick: failed (2, err:-62)
[   29.977312] virtio_rpmsg_bus virtio1: rpmsg host is online
[   29.982909] rproc-virtio rproc-virtio.2.auto: registered virtio1 (type 7)
[   29.989734] remoteproc remoteproc0: remote processor imx-rproc is now up
Suspend Linux...
[   31.028164] PM: suspend entry (deep)
[   31.037105] Filesystems sync: 0.005 seconds
[   31.041924] (NULL device *): Direct firmware load for /home/root/.cache/gopoint/lp_baby_detection.elf failed with error -2
[   31.053026] (NULL device *): Falling back to sysfs fallback for: /home/root/.cache/gopoint/lp_baby_detection.elf
[   31.065598] Freezing user space processes
[   31.071261] Freezing user space processes completed (elapsed 0.001 seconds)
[   31.078250] OOM killer disabled.
[   31.081516] Freezing remaining freezable tasks
[   31.087190] Freezing remaining freezable tasks completed (elapsed 0.001 seconds)
[   31.094594] printk: Suspending console(s) (use no_console_suspend to debug)
```

The printing of the M-core serial terminal:

```
####################  LP Baby Detection Task ####################

    Build Time: Jul 22 2024--15:14:35
    Core Clock: 200000000Hz
    PDM Clock: 196608000Hz
    Loop 11
Recording for 1 second...
Recording finished!
Start inferencing...!
invoke_status is 0
yes: 0, no: 255
WorkingTask 1: Transfer from RUN to SUSPEND
Will wakeup in 2 seconds.
Target powermode get in SUSPEND
== Power switch OK ==

Next loop

####################  LP Baby Detection Task ####################

    Build Time: Jul 22 2024--15:14:35
    Core Clock: 200000000Hz
    PDM Clock: 196608000Hz
    Loop 12
Recording for 1 second...
Recording finished!
Start inferencing...!
invoke_status is 0
yes: 0, no: 255
WorkingTask 1: Transfer from RUN to SUSPEND
Will wakeup in 2 seconds.
Target powermode get in SUSPEND
```

When playing audio with a baby crying, the M-core successfully recognizes and wakes up the A-core.

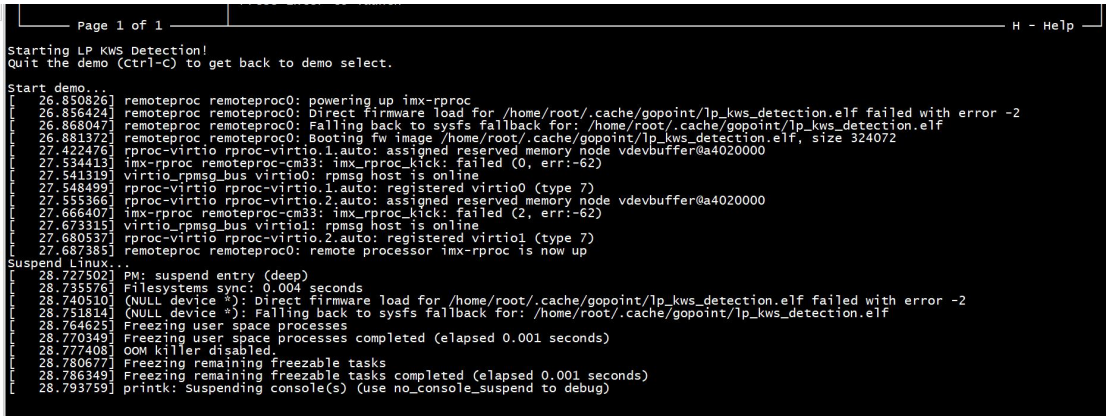
### 3.17.3.2 LP KWS Detection

This demo is similar to the "LP Baby Cry Detection" demo, you also need to change the environment variables in uboot as following:

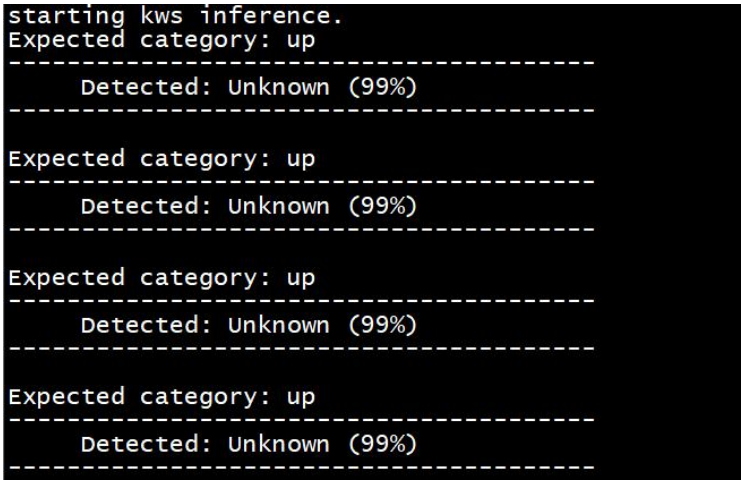
```
Normal Boot
Hit any key to stop autoboot:  0
u-boot=> setenv mmcargs setenv bootargs console=ttyLP0,115200 console=tty1
root=/dev/mmcblk0p2 rootwait rw clk_ignore_unused
u-boot=> boot
```

After starting the demo, the A-core goes into sleep mode, the terminal prints the following log:

The printing of the A-core serial terminal:




The printing of the M-core serial terminal:

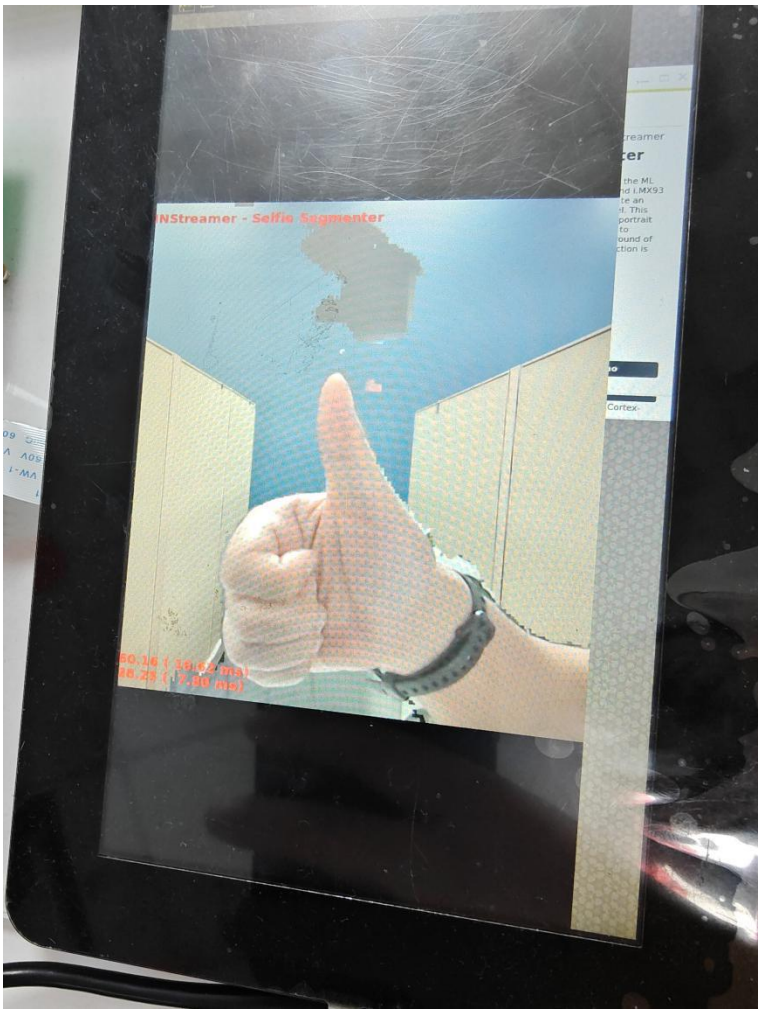



When speaking the keyword "*UP*" into the microphone towards the board, the M-core successfully recognizes and wakes up the A-core.

### 3.17.3.3 Selfie Segmenter

This demo runs successfully on MaaXBoard OSM93 board, as shown in the diagram.

Note: This demo requires to use USB camera.




### 3.17.3.4 i.MX Smart Fitness

This demo runs successfully on MaaXBoard OSM93 board, as shown in the diagram.


### 3.17.3.5 DMS

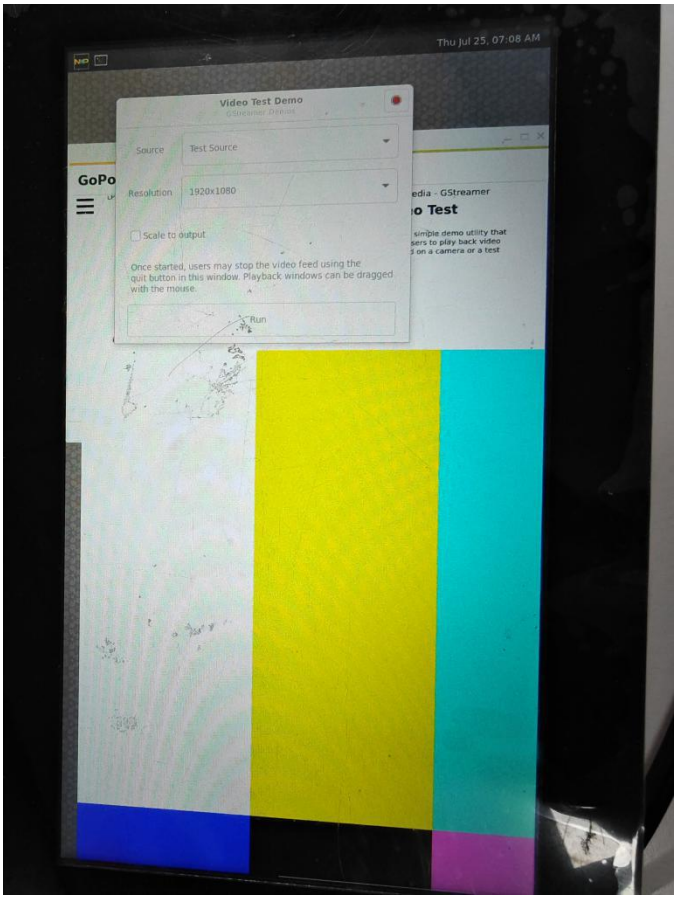
This demo runs successfully on MaaXBoard OSM93 board, as depicted in the diagram.


### 3.17.3.6 ML Benchmark

This demo runs successfully on MaaXBoard OSM93 board, as depicted in the diagram.


### 3.17.3.7 Video Test

This demo runs successfully on MaaXBoard OSM93 board, as depicted in the diagram.




## 3.18 Procedure to Increase eMMC Partition Size

Since the image is smaller than the storage device, it is not properly expanded when first flashed.

To avoid running out of eMMC or SD card space when installing the software package, perform the following steps to expand the rootfs partition.

- View the Partition Size using following command:

```
root@maaxboardosm93:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       4.9G  2.9G  1.8G  63% /
devtmpfs        196M  4.0K  196M   1% /dev
tmpfs           454M     0  454M   0% /dev/shm
tmpfs           182M  9.3M  173M   6% /run
tmpfs           4.0M     0  4.0M   0% /sys/fs/cgroup
tmpfs           454M  8.0K  454M   1% /tmp
tmpfs           454M  196K  454M   1% /var/volatile
```

```
/dev/mmcblk0p1     84M     34M     50M   41% /run/media/boot-mmcblk0p1
tmpfs              91M    4.0K     91M    1% /run/user/0
```

- Expand the partition using following command:

```
root@maaxboardosm93:~# expand_rootfs
[ 8554.839080] EXT4-fs (mmcblk0p2): resizing filesystem from 1340871 to 3803136 blocks
[ 8554.914794] EXT4-fs (mmcblk0p2): resized filesystem to 3803136
Expand rootfs size successfully, it will be enlarged upon the next reboot.
```

- View again, the partition size has been expanded:

```
root@maaxboardosm93:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        14G  2.9G   11G  22% /
devtmpfs        196M  4.0K  196M   1% /dev
tmpfs           454M     0  454M   0% /dev/shm
tmpfs           182M  9.3M  173M   6% /run
tmpfs           4.0M     0  4.0M   0% /sys/fs/cgroup
tmpfs           454M  8.0K  454M   1% /tmp
tmpfs           454M  196K  454M   1% /var/volatile
/dev/mmcblk0p1   84M   34M   50M  41% /run/media/boot-mmcblk0p1
tmpfs            91M  4.0K   91M   1% /run/user/0
```

# Chapter 4 Program or update the system Images

## 4.1 Program Images Using uuu

MaaXBoard Osm93 supports programming the system image to the development board using uuu tool.

### 4.1.1 Preparation

#### 4.1.1.1 Running Environment

- Programming Tool: Universal Update Utility (Short as UUU), Download Link: https://github.com/NXPmicro/mfgtools/releases(Version 1.5.179 or later)
- System Environment: Win10 64 bit OS or Ubuntu 64 bit OS, 16.14 LTS or later LTS version

#### 4.1.1.2 Images

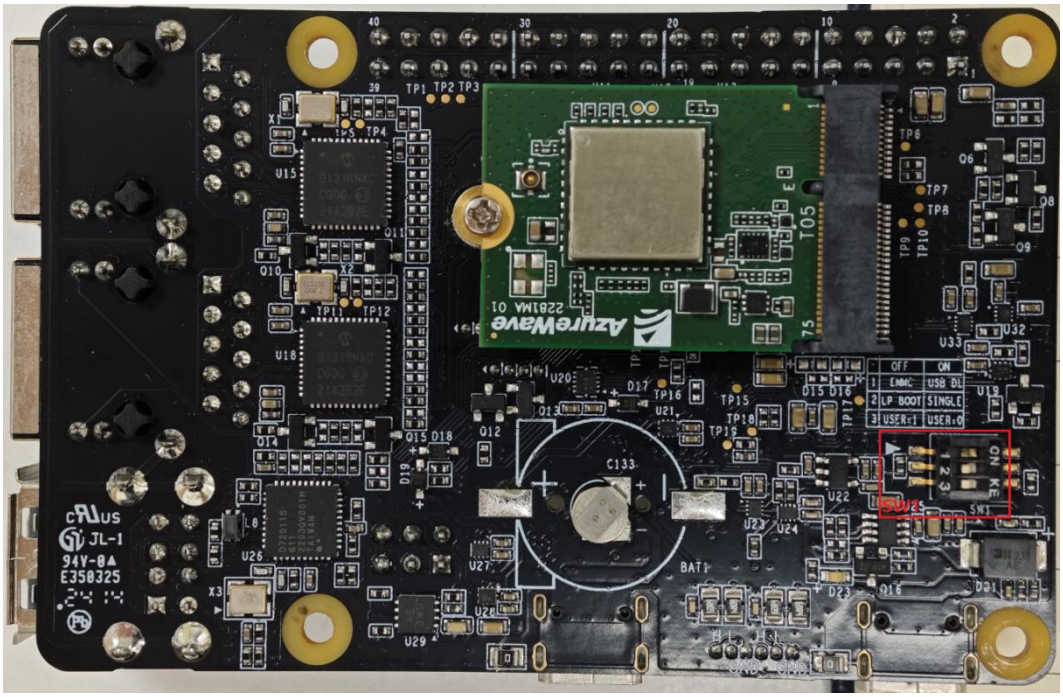
Put the following image files into the download directory

- MaaXBoard Osm93 Linux system image file, e.g: avnet-image-full-maaxboard-osm93.rootfs-20241122034816.wic
- U-boot image file: u-boot-maaxboard-osm93.imx, usually the same version as the one contained in the system Image

### 4.1.2 Program the Image under Windows OS

1. Set SW1 Boot Switch on MaaXBoard Osm93 to Serial Download Mode:

| | 1 | 2 | 3 |
|---|---|---|---|
| Serial Download Mode： | ON | 0 | 0 |
| Boot Mode： | 0 | 0 | 0 |



2. Connect J2/J5 to a network cable.

3. Put the built files: u-boot image, entire system image, and uuu.exe into a same folder.

After completing the preparations, take the following images as an example:

u-boot image: u-boot-maaxboard-osm93.imx

entire system image: avnet-image-full-maaxboard-osm93.wic

4. Powered the board with a 5V, 2A, Type-C interface power (to J11).

5. Execute the following commands to program the uboot image or the system image into the eMMC.

```
./uuu -b emmc u-boot-maaxboard-osm93.imx
./uuu -b emmc_all u-boot-maaxboard-osm93.imx
avnet-image-full-maaxboard-osm93.rootfs-20241122034816.wic
```


6. When programming finished, power down MaaXBoard Osm93, set SW1 to ***Boot Mode***, power on the

board again, the board will boot with the new system image.

### 4.1.3 Program the Image under Ubuntu OS

#### 4.1.3.1 Program the Image into eMMC under Ubuntu OS

1. Powered the board with a 5V, 2A, Type-C interface power (to J11).
2. Execute the following command to program the uboot image or the system image into the eMMC.

```
./uuu -b emmc u-boot-maaxboard-osm93.imx
./uuu -b emmc_all u-boot-maaxboard-osm93.imx
avnet-image-full-maaxboard-osm93.rootfs-20241122034816.wic
```

3. When programming finished, power down MaaXBoard Osm93, set SW1 to ***Boot Mode***, power on the board again, then the board will boot with the new system image.

# Chapter 5 Appendix

## 5.1 Hardware Documents

For the detail hardware introduction, please refer to ***MaaXBoard Osm93 Hardware user manual***.

## 5.2 Software Documents

MaaXBoard Osm93 supports Yocto Linux system, for additional information, please refer to the following documents:

- MaaXBoard Osm93 Linux Yocto UserManual

Describes how to boot MaaXBoard Osm93 and aspects of the BSP functionality (This document)

- MaaXBoard Osm93 Linux Yocto Development Guide

Detailed guidance on how to rebuild the Linux system image

## 5.3 Contact Information

- Website:
  https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/maaxboard/maaxboard-osm93/maaxboard-osm93-board-family