EM-TF-EVK-AM5728

Linux User Manual

Version: 1.0

2024-05-21

Disclaimer

- Shenzhen Emtop Co., Ltd. does not provide any type of guarantee for the program source code, software, documents, etc. provided with the product; Regardless of whether it is described or implied, including but not limited to the warranty of fitness for a particular purpose, the entire risk is born by the user.

Revision History：

Version	Date	Description
0.1	2024-05-21	First Release

Table of Contents

1. Product Overview

1.1 Introduction

The content of the software release package shall be subject to the actual release. Please refer to the following table for the release file structure and description.

```
├──01Doc
|   ├──RN
|   |   EM-TF-EVK-AM5728-ReleaseNote-Vxx.pdf
|   └──UM
|     EM-TF-EVK-AM5728-UserManual-Vxx.pdf
|     EM-TF-EVK-AM5728-DevelopmentGuide-Vxx.pdf
└──02Linux
├──01LinuxSourceCode
|   linux_4.9.28_ xxx.tar.gz
|   u-boot_2017.01._xxx.tar.gz
├──02LinuxShipmentImage
|   EM-TF-EVK-AM5728-TI-ShipmentImage-EMMC-Vxxx.rar
|   EM-TF-EVK-AM5728-TI-ShipmentImage-SDcard-Vxxx.rar
└──03LinuxTools
    gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi.tar.xz
```

01Doc	Description
EM-TF-EVK-AM5728-ReleaseNote-Vxx.pdf	Release Note
EM-TF-EVK-AM5728-UserManual-Vxx.pdf	User Manual
EM-TF-EVK-AM5728-DevelopmentGuide-Vxx.pdf	Development Guide
01LinuxSourceCode	**Description**
linux_4.9.28_xxx.gz	Linux kernel source code: 4.9.28 version
u-boot_2017.01._xxx.tar.gz	u-boot source code: 2015.04
02LinuxShipmentImage	**Description**
EM-TF-EVK-AM5728-TI-ShipmentImage-SDCard-Vxxx.rar	Arago image with firmwares, SD Card Image
EM-TF-EVK-AM5728-TI-ShipmentImage-EMMC-Vxxx.rar	Arago image with firmwares, eMMC Image
03LinuxTools	**Description**
gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi.tar.xz	Gcc compiler for u-boot, kernel and applications

xxx	Other tools

1.2 Function List

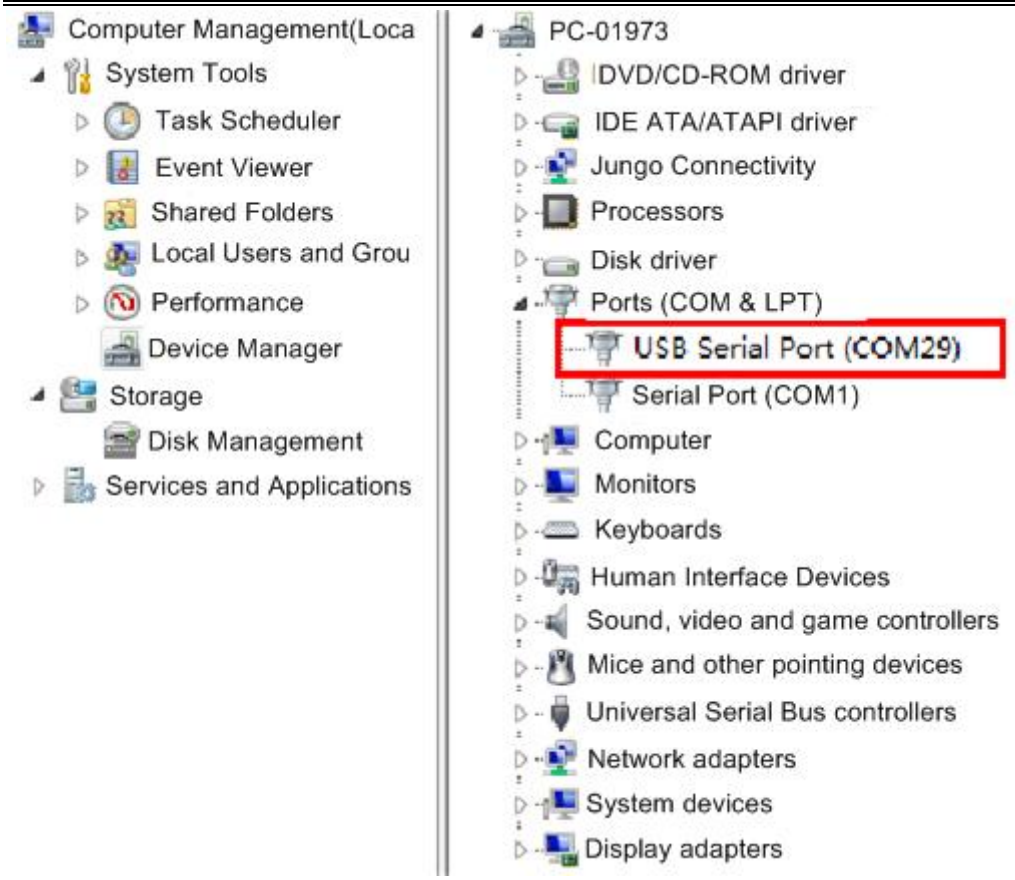
- U-Boot version: 2017.01
- Kernel version: 4.9.28
- Evaluaton image Arago
- Qt 5.5.1 Library
- MatrixBox desktop demo
- eMMC(default)or Micro SD boot
- LCD /LVDS and HDMI display
- Audio output & Micphone
- HDMI audio output
- 2 Gigabit Ethernet (RJ45)
- 2 CAN port
- 2 high-speed USB2.0 Host and 1 usb3.0 Host
- 2RS485 port and 1 USB-UART debug port
- External interfaces (I2CX. UART,SPI ,SATA,PCIE and GPIO)
- RTC clock
- WIFI(2.4G/5G)& Bluetooth BR/EDR/BLE
- 12 bit digital camera
- 4G

2. Quick Start

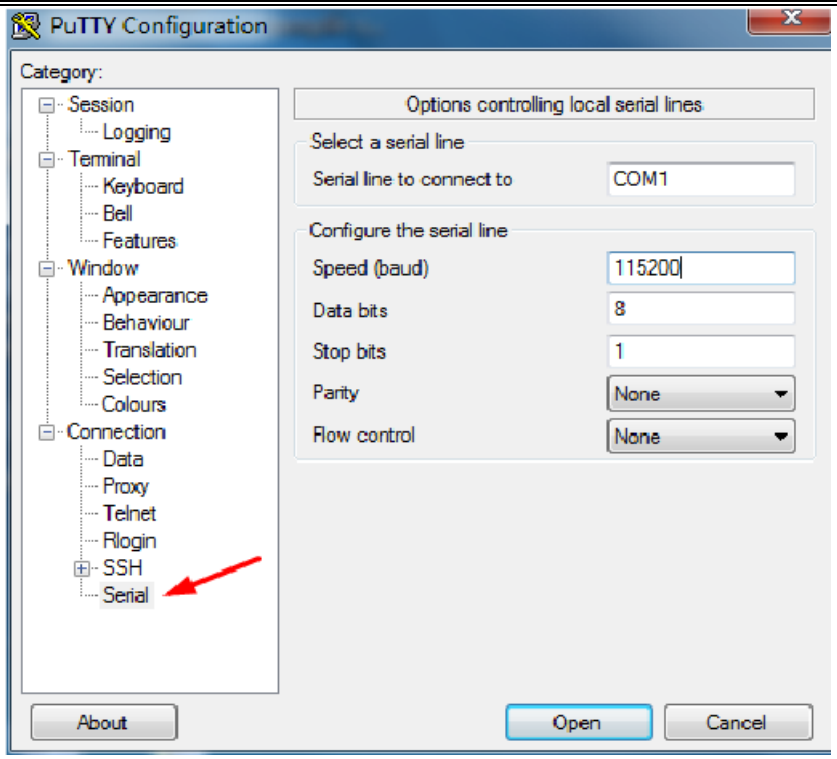
EM-TF-EVK-AM5728 has the image burned into EMMC by default and it can boot directly from EMMC. If you need to use the latest system image, please refer to Chapter ***Chapter 5: Programming and updating system image***. For detailed information on circuit connections and accessories used, please refer to the QSG documentation

2.1 BOOT FROM EMMC

- Use a MiniUSB cable to connect the DEBUG interface (CON1) on the board to the USB host of the PC.
- Power on the board and install FTDI's USB to TTL driver (Tools directory or visit http://www.ftdichip.com/Drivers/VCP.htm), after installation completes, the device manager will display the port of the USB serial port.



- Install the serial port software (such as PUTTY) on the PC, select the correct port number, baud rate 115200, 8 data bits, 1 stop bit, No parity.



- Set DIP switches S6 and S7 refer to the table below.

BOOT CONFIG	
BOOT0	L
BOOT1	L
BOOT2	L
BOOT3	L
BOOT4	L
BOOT5	H

- Use a 12V, 2A power supply to power the board (J6). When powered on, the system will start automatically.
- If a touch screen is connected to the board, the system will hint for calibration when the system is started for the first time. At this time, you only need to click the corresponding point display on screen.
- After the system starts, enter the user name **root** on the serial terminal to log in.

2.2 BOOT FROM SD CARD

The steps to boot the system from the SD card are basically the same as the process of booting from the eMMC. The following 2 operations need to be performed before powering on the board:

- Insert the prepared SD card into the slot on the board (J5)
- Set DIP switches S6 and S7 refer to the table below.

BOOT CONFIG	
BOOT0	L
BOOT1	H
BOOT2	L
BOOT3	L
BOOT4	L
BOOT5	H

3. Function Configuration and Description

Please refer to Chapter 2 to start the system, and then follow the instructions below to use the functions of AM5728.

3.1 USER LED

User can control the two single-color LED indicators D55 and D56 on EM-TF-BB-AM57x (corresponding to User LED1 and User LED2 respectively). Execute the following commands in the serial terminal to control them:

Turn off LEDs:

- root@arm:~# **echo 0 > /sys/class/leds/user-led1/brightness**
- root@arm:~# **echo 0 > /sys/class/leds/user-led2/brightness**

Turn on LEDs:

- root@arm:~# **echo 1 > /sys/class/leds/user-led1/brightness**
- root@arm:~# **echo 1 > /sys/class/leds/user-led2/brightness**

3.2 BUZZER

There is a buzzer PZ1 on base board.

Turn on the buzzer:

- root@arm:~# **echo 1 > /sys/class/misc/buzzer_ctl/state**

Turn off the buzzer:

- root@arm:~# **echo 0 > /sys/class/misc/buzzer_ctl/state**

3.3 BUTTON

The EM-TF-BB-AM57X base board has 3 buttons S3, S4, and S5. S3 is the reset button, and S4 and S5 are two user buttons.

Run the following command, after pressing S4 or S5, a string of numbers will be printed out, that means it's normal.

- root@arm:~# **od -x /dev/input/event0**

```
0000000 51eb 5b21 b98f 000c 0001 0100 0001 0000
0000020 51eb 5b21 b98f 000c 0000 0000 0000 0000
0000040 51ec 5b21 4685 0000 0001 0100 0000 0000
0000060 51ec 5b21 4685 0000 0000 0000 0000 0000
0000100 51ed 5b21 be73 0009 0001 0101 0001 0000
0000120 51ed 5b21 be73 0009 0000 0000 0000 0000
0000140 51ed 5b21 5e12 000d 0001 0101 0000 0000
0000160 51ed 5b21 5e12 000d 0000 0000 0000 0000
```

3.4 RELAY

There are two relays LS1 and LS2 on the EM-TF-BB-AM57X base board, which are controlled by GPIO5_4 and GPIO7_30 respectively.

3.4.1 LS1

Configure GPIO5_4:

- root@arm:~# **echo 132 > /sys/class/gpio/export**
- root@arm:~# **echo out > /sys/class/gpio/gpio132/direction**

Turn on relay LS1, J19 DO2+ pin and DO2- pin are connected

- root@arm:~# **echo 1 > /sys/class/gpio/gpio132/value**

Turn off relay LS1, J19 DO2+ pin and DO2- pin are disconnected

- root@arm:~# **echo 0 > /sys/class/gpio/gpio132/value**

3.4.2 LS2

Configure GPIO7_30:

- root@arm:~# **echo 222 > /sys/class/gpio/export**
- root@arm:~# **echo out > /sys/class/gpio/gpio222/direction**

Turn on relay LS2, J19 DO1+ pin and DO1- pin are connected

- root@arm:~# **echo 1 > /sys/class/gpio/gpio222/value**

Turn off relay LS2, J19 DO1+ pin and DO1- pin are disconnected

- root@arm:~# **echo 0 > /sys/class/gpio/gpio222/value**

3.5 OPTOCOUPLER

There are two optocouplers OC1 and OC2 on the EM-TF-BB-AM57X base board, which are controlled by GPIO4_11 and GPIO4_10 respectively.

3.5.1 OC1

Configure GPIO4_11:

- root@arm:~# **echo 107 > /sys/class/gpio/export**
- root@arm:~# **echo in > /sys/class/gpio/gpio107/direction**

Read the current status, 1 means the optocoupler is not conducting.

- root@arm:~# **cat /sys/class/gpio/gpio107/value**

```
1
```

J19 DI2+ DI2- is connected to the 5V power supply and GND respectively. Note that DI2+ is connected to the positive pole and DI2- is connected to the negative pole, and then read the GPIO status.

- root@arm:~# **cat /sys/class/gpio/gpio107/value**

```
0
```

The read value 0 means the optocoupler is conducting.

3.5.2 OC2

Configure GPIO4_10:

- root@arm:~# **echo 106 > /sys/class/gpio/export**
- root@arm:~# **echo in > /sys/class/gpio/gpio106/direction**

Read the current status, 1 means the optocoupler is not conducting.

- root@arm:~# **cat /sys/class/gpio/gpio106/value**

```
1
```

J19 DI1+ DI1- is connected to the 5V power supply and GND respectively. Note that DI1+ is connected to the positive pole and DI1- is connected to the negative pole, and then read the GPIO status.

- root@arm:~# **cat /sys/class/gpio/gpio106/value**

```
0
```

The read value 0 means the optocoupler is conducting.

3.6 RTC

The RTC module is powered by the capacitor. Run below command to test it.

Read the system time:

- root@arm:~# **date**

```
Tue Jul 3 00:09:37 UTC 2018
```

Set the current time, such as: June 12, 2018 10:15

- root@arm:~# **date 061210152018**

```
Tue Jun 12 10:15:00 UTC 2018
```

Write the system time into hardware RTC:

- root@arm:~# **hwclock -w**

Read time from hardware RTC:

- root@arm:~# **hwclock**

```
Tue Jun 12 10:16:03 2018 0.000000 seconds
```

As you can see, the hardware clock RTC is set to June 12, 2018, and the system clock is saved to the hardware RTC.

Reboot and check the time:

- root@arm:~# **date**

```
Tue Jun 12 10:17:19 UTC 2018
```

3.7 eMMC

The node of the onboard eMMC memory of EM-TF-BB-AM57XX in the file system is /dev/mmcblk1. The default system boot media is eMMC. Please refer to the development manual about how to update the image in eMMC. The following steps show simple read and write operations to eMMC when booting from the SD card.

- root@arm:~# **touch emmc_read emmc_write**
- root@arm:~# **echo "emmc write test" > emmc_write**

Write the file into eMMC:

- root@arm:~# **dd if=emmc_write of=/dev/mmcblk1**

```
0+1 records in
0+1 records out
```

Read content from eMMC, and keep into file:

- root@arm:~# **dd if=/dev/mmcblk1 of=emmc_read bs=1K count=10**

```
10+0 records in
10+0 records out
```

Check the file emmc_read:

- root@arm:~# **cat emmc_read**

```
emmc write test
```

3.8 QSPI_Flash

You can view QSPI_Flash partition information with the following command:

- root@arm:~# **cat /proc/mtd**

```
dev: size erasesize name
mtd0: 00030000 00010000 "QSPI.SPL"
mtd1: 00010000 00010000 "QSPI.SPL.backup1"
mtd2: 00100000 00010000 "QSPI.u-boot"
```

```
mtd3: 00080000 00010000 "QSPI.u-boot-spl-os"
mtd4: 00010000 00010000 "QSPI.u-boot-env"
mtd5: 00010000 00010000 "QSPI.u-boot-env.backup1"
mtd6: 00800000 00010000 "QSPI.kernel"
mtd7: 01620000 00010000 "QSPI.user-data"
```

The following introduces the read and write test of QSPI_Flash:

Use the following command to write MLO to the first partition of QSPI_Flash:

- root@arm:~# **dd if=/run/media/mmcblk0p1/MLO of=/dev/mtdblock0**

```
231+1 records in
231+1 records out
```

Then read the MLO size bytes from the first partition to the file MLO.test, the size of the MLO can be obtained with the following command:

- root@arm:~# **ls /run/media/mmcblk0p1/MLO -l**

```
-rwxrwx--- 1 root disk 118578 Aug 3 2018 /run/media/mmcblk0p1/MLO
```

Read the contents of the first partition to MLO.test.

- root@arm:~# **dd if=/dev/mtdblock0 of=MLO.test bs=1 count=118578**

```
118578+0 records in
118578+0 records out
```

Compare the two files:

- root@arm:~# **diff MLO.test /run/media/mmcblk0p1/MLO**

```
/* Nothing reports, it means the two files are same */
```

3.9 DISPLAY

The current system supports two types of display device: HDMI and LVDS. You can refer to the table below to connect the screen to the AM5728, and then start the system. The default display device is HDMI.

DISPLAY TYPE	INTERFACE
HDMI [DEFAULT]	J13 (Standard HDMI)

LVDS	J10

3.9.1 CHOOSE DISPLAY DEVICE

The display device can be switched by modifying the **fdt_file** parameter in the **uEnv.txt** file.

- root@arm:~# **vi uEnv.txt**

HDMI:

```
fdt_file=embest-SOM_AM572x_TM-mode0.dtb
```

LVDS:

```
fdt_file=embest-SOM_AM572x_TM-mode0-LCD.dtb
```

- root@arm:~# **sync; reboot**

3.9.2 ADJUST LVDS BRIGHTNESS

LVDS interface supports brightness adjustment. The valid brightness level is: 0 ~ 8, from dim to bright.

Set minimum brightness:

- root@arm:~# **echo 0 > /sys/class/backlight/backlight/brightness**

Set maximum brightness:

- root@arm:~# **echo 8 > /sys/class/backlight/backlight/brightness**

3.10 TOUCH PANEL

Connect the touch panel to J10. There are two touch interfaces on the base board: I2C and USB. The header J11 is for I2C touch and J12 for USB touch.

3.10.1 I2C TOUCH

After installing the touch screen, enter the following command:

- root@arm:~# **evtest /dev/input/event0**

Touch the panel, and the system will report event message:

```
Testing ... (interrupt to exit)
Event: time 1498681510.561501, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 0
Event: time 1498681510.561501, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 345
Event: time 1498681510.561501, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 412
Event: time 1498681510.561501, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
Event: time 1498681510.561501, type 3 (EV_ABS), code 0 (ABS_X), value 345
```

Note：

🕮 Press CTRL+C to terminate **evtest** program.

3.10.2 USB TOUCH

After installing the touch screen, enter the following command:

- root@arm:~# **evtest /dev/input/event1**

Touch the panel, and the system will report event message:

```
Testing ... (interrupt to exit)
Event: time 1498681543.597190, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 0
Event: time 1498681543.597190, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 2639
Event: time 1498681543.597190, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 2408
Event: time 1498681543.597190, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
```

Note：

🕮 Press CTRL+C to terminate **evtest** program.

3.11 AUDIO

3.11.1 AUDIO INTERFACE ON BOARD

The board is equipped with an audio input interface J18 and an output interface J17, which support recording and playback. After connecting the recording and playback equipment (such as 3.5mm headphone and microphone), you can use the following commands to record and play audio.

Configure before record:

- root@arm:~# **amixer -c AM5728EmbestTLV sset 'Left PGA Mixer Mic2L' on**
- root@arm:~# **amixer -c AM5728EmbestTLV sset 'Right PGA Mixer Mic2R' on**
- root@arm:~# **amixer -c AM5728EmbestTLV sset 'PGA' 40**
- root@arm:~# **amixer -c AM5728EmbestTLV sset 'Output Driver Power-On time' 200ms**

Start recording:

- root@arm:~# **arecord -t wav -c 1 -r 44100 -f S16_LE -v k**

Playback:

- root@arm:~# **aplay -t wav -c 2 -r 44100 -f S16_LE -v k**

There are several wav files with different sample rate in the shipment image: 22_3 (22KHz), 32_3 (32KHz), 44_3 (44KHz) and 48_3 (48KHz).

Play them with below command:

- root@arm:~# **aplay test_audio/22_3**

3.11.2 HDMI AUDIO

Choose display device to HDMI and bring up the system, connect the HDMI display and HDMI audio equipment, play audio files:

- root@arm:~# **aplay -D plughw:1,0 test_audio/48_3**

3.12 UART

The EM-TF-EVK-AM5728 base board has 6 serial ports:

UART PORT	USAGE
UART1	J31
UART3	DEBUG PORT
UART7	RS485
UART9	RS485
UART10	J31

3.12.1 UART1

The node of UART1 under the Linux system is /dev/ttyO0. User can use the **uart_test** program integrated in the shipment image to do send-and-receive test. Connect the 28th and 30th pins of J31 on the base board, and input command:

- root@arm:~# **./uart_test -d /dev/ttyO0 -b 115200**

```
/dev/ttyO0 SEND: 1234567890
/dev/ttyO0 RECV 10 total
/dev/ttyO0 RECV: 1234567890
```

The contents sent are the same with received, it means the test passes.

Note:

Press CTRL+C to terminate **uart_test** program.

3.12.2 UART10

The node of UART10 under the Linux system is /dev/ttyO9. User can use the **uart_test** program integrated in the shipment image to do send-and-receive test. Connect the 24th and 26th pins of J31 on the base board, and input command:

- root@arm:~# **./uart_test -d /dev/ttyO9 -b 115200**

```
/dev/ttyO9 SEND: 1234567890
/dev/ttyO9 RECV 10 total
/dev/ttyO9 RECV: 1234567890
```

The contents sent are the same with received, it means the test passes.

Note:

Press CTRL+C to terminate **uart_test** program.

3.12.3 RS485

There are two RS485 interfaces on the base board. User can use the test program for testing. Connect pins 1, 2, and 3 of the J22 socket to pins 1, 2, and 3 of the J23 socket respectively.

Start background process receiving from UART9:

- root@arm:~# **./uart_receive -d /dev/ttyO8 -b 115200 &**

Send data from UART2:

- root@arm:~# **./uart_send -d /dev/ttyO1 -b 115200**

If the test is successful, UART9 receives the character "f"

```
/dev/ttyO1 SEND: f
```

```
f
[1]+ Done ./uart_receive -d /dev/ttyO8 -b 115200
```

3.13 GIGABIT ETHERNET ONBOARD

Connect the network cable to J27 (eth0) or J28 (eth1). Assuming the network cable is connected to J27. Enter the following command in the serial terminal to set the IP address:

- root@arm:~# **ifconfig eth0 192.168.52.64**

Network connection test:

- root@arm:~# **ping 192.168.52.1**

3.13.1 CONFIGURE STATIC IP

Modify **/etc/network/interfaces** and add or modify parameters in "The primary network interface" section as below:

```
auto eth0
iface eth0 inet static
address 192.168.1.139
gateway 192.168.1.1
netmask 255.255.255.0
```

- root@arm:~# **sync; reboot**

3.13.2 CONFIGURE DHCP

Modify **/etc/network/interfaces**, delete address, gateway and netmask parameters under eth0, and only keep:

```
auto eth0
```

```
iface eth0 inet manual
```

- root@arm:~# **sync; reboot**

3.14 PCIE NETWORK

Insert the PCIE network card into the card slot J14, connect the network cable to the PCIE network card, and then boot up the board. Let's takes Realtek's PCIE network card as an example. Execute the following command in the terminal to view the network card information:

- root@arm:~# **lspci**

```
00:00.0 PCI bridge: Texas Instruments Multicore DSP+ARM KeyStone II SOC (rev 0
1)
01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI
Express Gigabit Ethernet
Controller (rev 09)
```

Check whether the PCIE network card is loaded successfully with the following command:

- root@arm:~# **ifconfig -a**

```
eth0      Link encap:Ethernet  HWaddr 00:E0:4C:18:05:25
          inet addr:192.168.22.19  Bcast:192.168.22.255  Mask:255.255.255.0
          inet6 addr: fe80::2e0:4cff:fe18:525%763860/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6529 (6.3 KiB)  TX bytes:9614 (9.3 KiB)
```

Our PCIE network card information is shown as above.

Note: If J27 and J28 are connected to network cables, disconnect them.

In order for the PCIE network card to automatically obtain IP, the following configuration file needs to be modified. Append enp* after eth*

- root@arm:~# **cat /etc/systemd/network/10-eth.network**

```
[Match]
Name=eth* enp*
KernelCommandLine=!root=/dev/nfs

[Network]
DHCP=yes
```

Note: Currently only PCIE network cards from Realtek (RTL8111/8168/8144) and Intel (82574L) are supported.

3.15 4G

The CN2 interface on the EM-TF-BB-AM57X base board can be installed with the 4G module EC20 (needs to be purchased). Install the SIM card into the J4 card slot.

- root@arm:~# **pppd call quectel-ppp &**

After connecting completes, you can ping the Baidu URL for testing.

- root@arm:~# **ping www.baidu.com -I ppp0**

3.15.1 4G CALL

We provide a 4G test application 4G_test. Execute the following command to make a call:

- root@arm:~# **./4G_test -d -D /dev/ttyUSB2 -n 13100690411**

Call the number 13100690411, it will automatically hang up after a few seconds.

Note:

Voice communication via 4G module is not supported yet.

3.15.2 4G MESSAGE

Execute the following command to send a text message to the target phone number:

- root@arm:~# **./4G_test -m -D /dev/ttyUSB2 -n 13100690411 -s "hello,world!"**

It will send a text message with the content "hello, world!" to the mobile phone number 13100690411.

3.16 CAN

Let's take CAN0 and CAN1 self-sending-and-receiving as an example to introduce the steps of using CAN bus.

Connect CAN0 and CAN1, that is, connect the 27pin of J2 to the 28pin, and the 29pin to the 30pin.

- Configure CAN bus:

Set baud rate to 50000 bps:

- root@arm:~# **canconfig can0 bitrate 50000**
- root@arm:~# **canconfig can1 bitrate 50000**

- root@arm:~# **canconfig can0 start**
- root@arm:~# **canconfig can1 start**

- Send and receive:

Let CAN0 keep receiving background:

- root@arm:~# **candump can0 &**

Let CAN1 send data. If CAN0 can receive the data, the CAN bus communication i s OK.

- root@arm:~# **cansend can1 -i 0x10 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88**

- Terminate the candump background process and close CAN bus:

- root@arm:~# **kill `pidof candump`**
- root@arm:~# **ip link set can0 down**

- root@arm:~# **ip link set can1 down**

3.17 USB 2.0/3.0

3.17.1 USB HOST

HUB1 on the base board has three USB Host interfaces, two USB2.0 interfaces (J15) and one USB3.0 (J16). Insert the U disk into any USB Host port, the disk information will display in the terminal:

```
usb 3-1.2: new high-speed USB device number 6 using xhci-hcd
usb-storage 3-1.2:1.0: USB Mass Storage device detected
scsi host1: usb-storage 3-1.2:1.0
scsi 1:0:0:0: Direct-Access Kingston DataTraveler 2.0 1.00 PQ: 0 ANSI: 4
sd 1:0:0:0: [sdb] 15131636 512-byte logical blocks: (7.75 GB/7.21 GiB)
sd 1:0:0:0: [sdb] Write Protect is off
sd 1:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or
 FUA
sdb: sdb1
sd 1:0:0:0: [sdb] Attached SCSI removable disk
```

Enter the following command:

- root@arm:~# **ls /dev/sdb***

```
/dev/sdb  /dev/sdb1
```

The device node /dev/sdb1 is the USB flash drive. You can use the mount command to mount the device for file reading and writing operations.

You can also connect other USB devices such as keyboard, mouse, etc.

3.18 CAMERA

The EM-TF-BB-AM57X base board has two parallel camera interfaces J7 and J29.

Connect the camera module to J7 (currently only supports OV5640).

The HDMI screen is connected to the HDMI interface (J17) on the baseboard. You can use the following two methods for camera preview testing.

- Preview on Matrix GUI

Click **Camera** -> **Dual Camera Demo** -> **RUN** icon

If only one camera is connected, the image of this camera will be displayed on the main screen. If two cameras are connected, the main screen will display the images from two cameras in the form of picture-in-picture.

- Preview with gstreamer

It's recommended to close weston desktop before gstreamer running:

- root@arm:~# **/etc/init.d/matrix-gui-2.0 stop**
- root@arm:~# **/etc/init.d/weston stop**

Run below command to preview:

- root@arm:~# **gst-launch-1.0 v4l2src device=/dev/video1 num-buffers=1000 io-mode=4 ! 'video/x-raw, format=(string)YUY2, width=(int)1920, height=(int)1080, framerate=60/1' ! vpe ! kmssink scale=false connector=32 -v**

In the above command, **device** is to select the camera device, **width** and **height** set the resolution, **framerate** sets the frame rate, and **connector** sets the display device. The connector ID of the display device can be obtained through the following command:

- root@arm:~# **modetest -c**

3.19 SATA

The EM-TF-BB-AM57X base board supports mounting mSATA 3.3V powered hard disk devices. Connect mSATA to CN1 and then start the system. Execute the following command in the serial terminal to view the device node and partition information of the hard disk:

- root@arm:~# **ls /dev/sda***

```
/dev/sda
```

- root@arm:~# **fdisk /dev/sda -l**

```
Disk /dev/sda: 29.5 GiB, 31675383808 bytes, 61865984 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

The system supports partitioning, mounting, reading and writing operations on SATA.

3.20 Wi-Fi

The onboard Wi-Fi module supports 2.4G/5G networking function, currently only supports STA mode.

3.20.1 CONNECT TO Wi-Fi ROUTER

Set the SSID and Password of the wireless router, and then create a configuration file on board:

- root@arm:~# **wpa_passphrase Embest 12345678 > /etc/wpa_supplicant.conf**

SSID: Embest. Password: 12345678

Start to connect

- root@arm:~# **wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant.conf -D nl80211**

Request IP:

- root@arm:~# **udhcpc -i wlan0 &**

ping test:

- root@arm:~# **ping 192.168.1.1 -I wlan0**
- root@arm:~# **ping www.baidu.com -I wlan0**

3.21 BLUETOOTH

3.21.1 INITIALIZE BLUETOOTH

Initialize Bluetooth module with below command:

- root@arm:~# **hciattach /dev/ttyO6 bcm43xx 115200 noflow**
- root@arm:~# **hciconfig hci0 up**

3.21.2 SCAN BLUETOOTH DEVICE

- root@arm:~# **hcitool scan**

```
Scanning ...
94:87:E0:DF:90:2D MyPhone
```

4. MATRIX GUI

4.1 MAIN INTERFACE

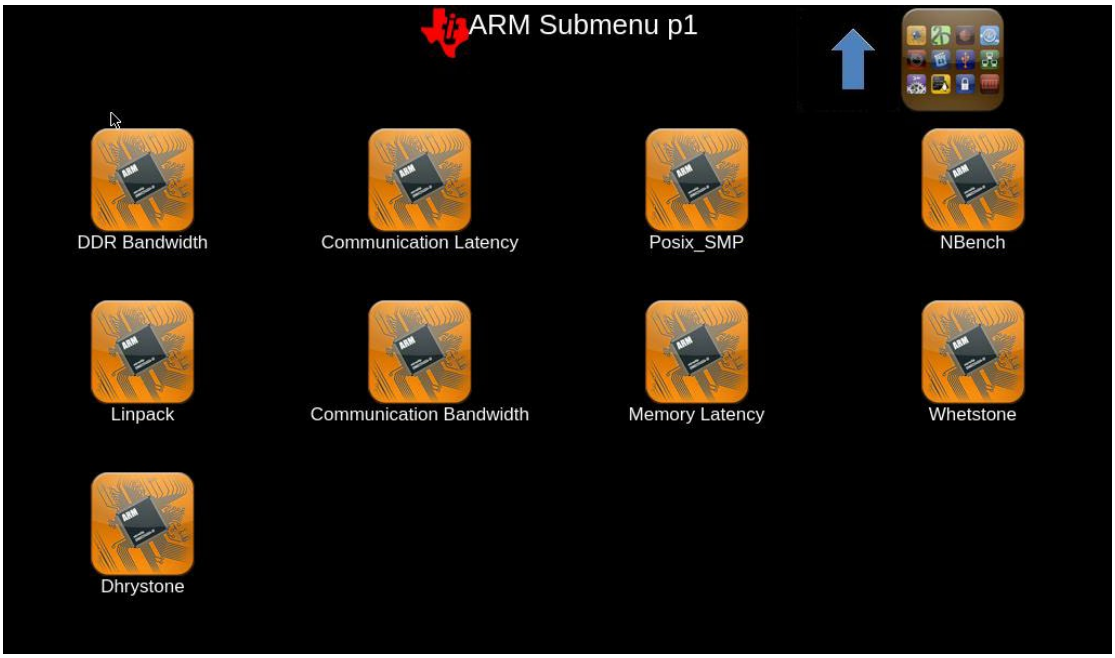
After the system boots up completely, Matrix GUI will automatically start. Matrix GUI supports mouse and touch screen operations. This demonstration is mainly for the Matrix desktop officially provided by Ti (TI Matrix User Guide WiKi: TI Matrix User Guide). It currently supports ARM performance testing, 3D effects, Settings, Qt5 simple routine demonstration, OpenCL computing, H265 decoding demonstration. After the ARM board is turned on, you can see the main interface of Matrix displayed on HDMI, or you can switch to LVDS output display as described in 3.9.1.

Note:

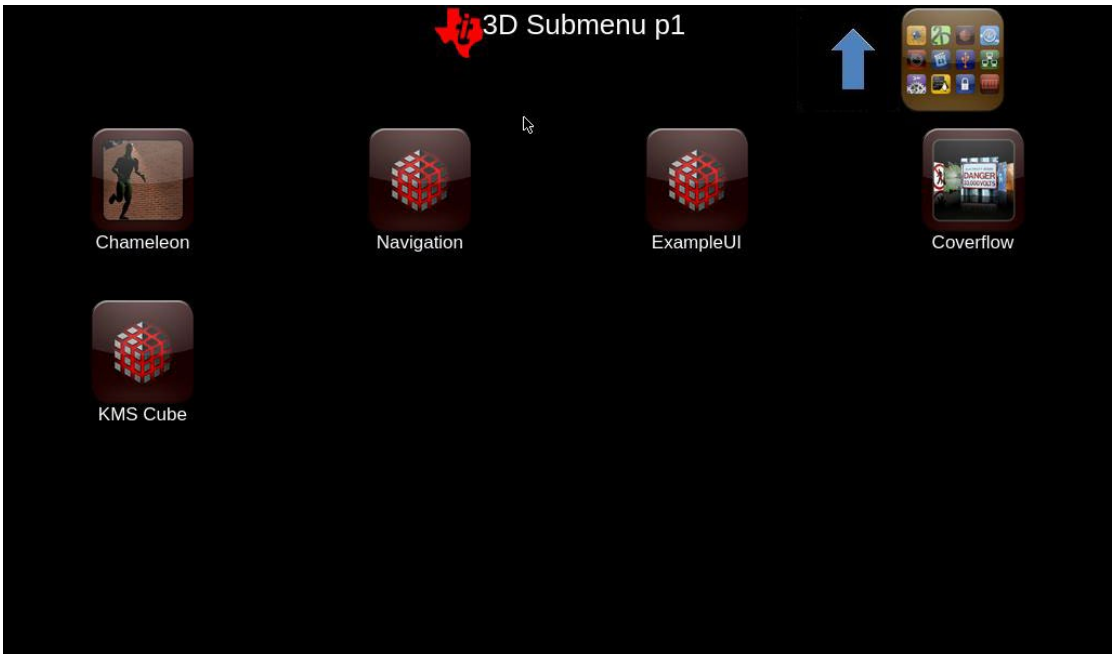
Ethernet function is not supported yet.

4.2 ARM FUNCTION DEMONSTRATION

Click the ARM icon and enter the ARM file directory. The interface is shown as below:



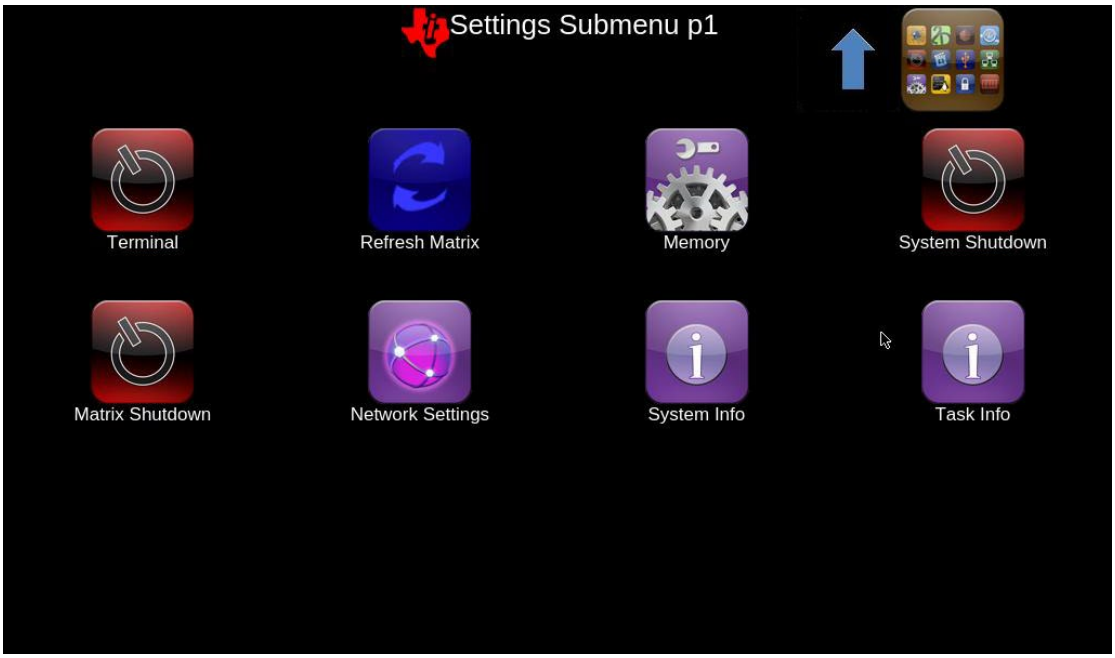
4.3 3D FUNCTION DEMONSTRATION



There are 5 3D demo examples in this directory.

Demo Name	Description
ChameleonMan	Demonstrates a matrix skinned character combined with bump mapping
CoverFlow	This is a demonstration of Coverflow style effect
ExampleUI	This demo shows how to efficiently render sprites and interface elements
Navigation	Demonstration showing how to implement rendering algorithms for navigation application
Kmscube	This demo shows how to render and display a multi-color rotating cube

4.4 SYSTEM SETTINGS

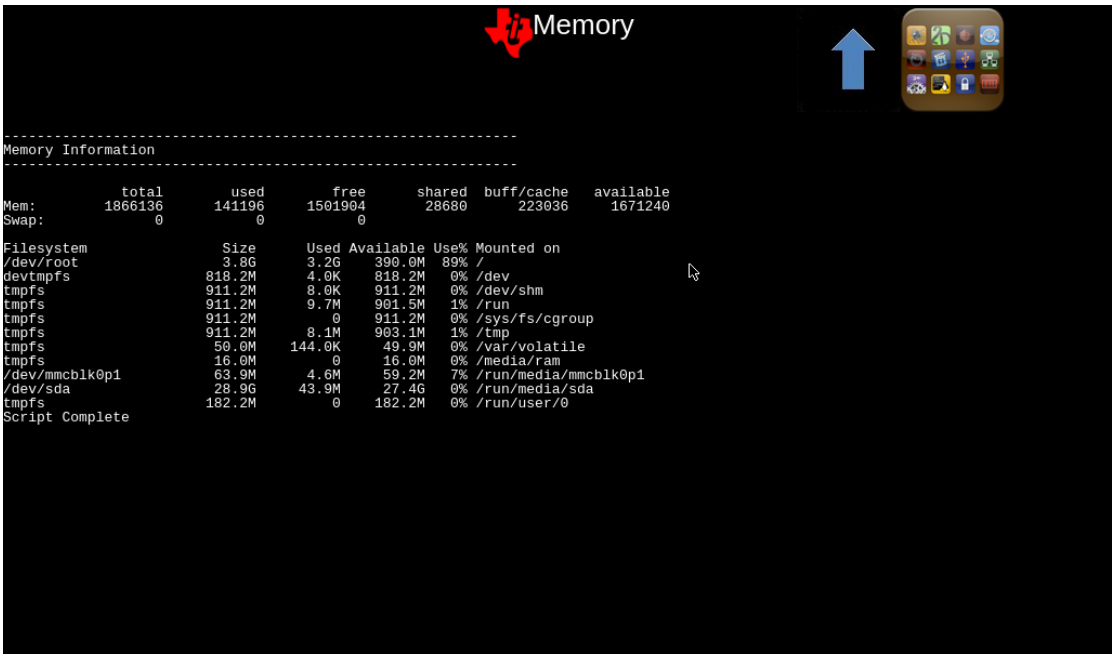


4.4.1 TERMINAL

Clicking the Terminal icon will open a Weston Terminal window.

4.4.2 MEMORY

Click Memory to view memory and storage information, which is equivalent to the Linux command **free** and **df**:

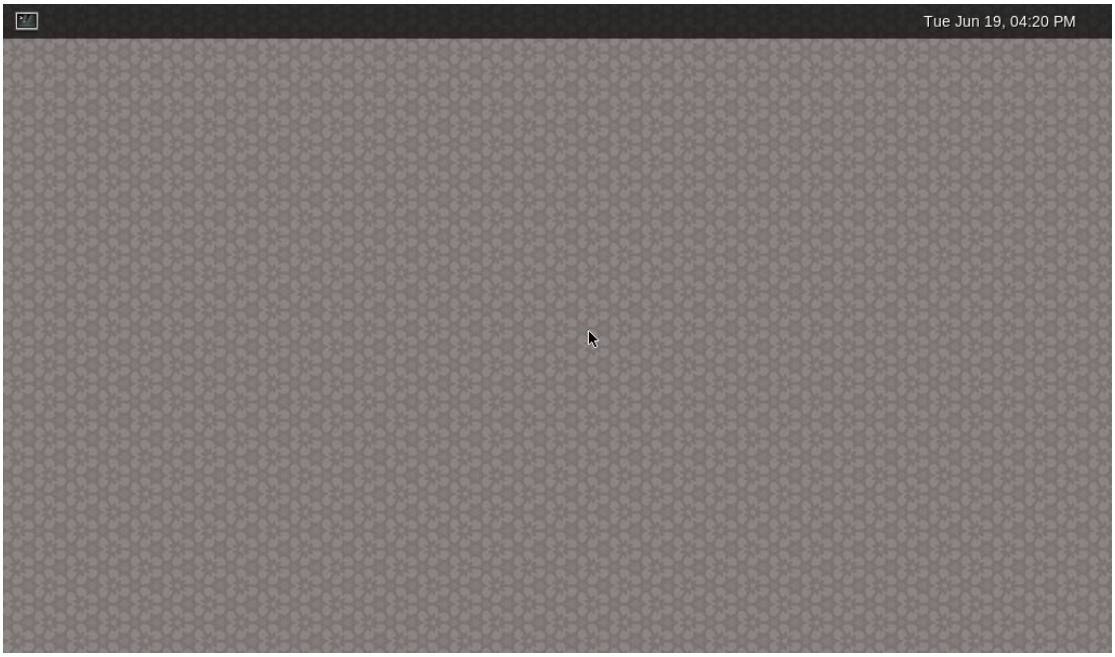


4.4.3 SYSTEM SHUTDOWN

Click System Shutdown and the system will shut down.

4.4.4 MATRIX SHUTDOWN

Clicking Matrix Shutdown will close the Matrix GUI, only display Weston desktop.

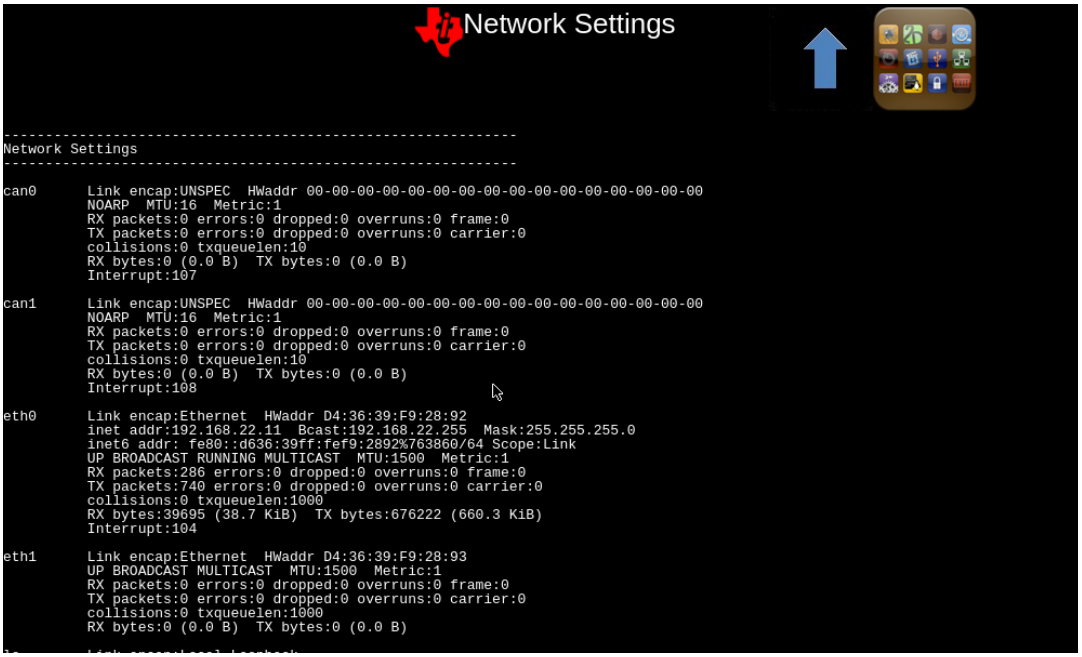


To restart the Matrix GUI, you need to run the following command:

- root@arm:~# **/etc/init.d/matrix-gui-2.0 start**

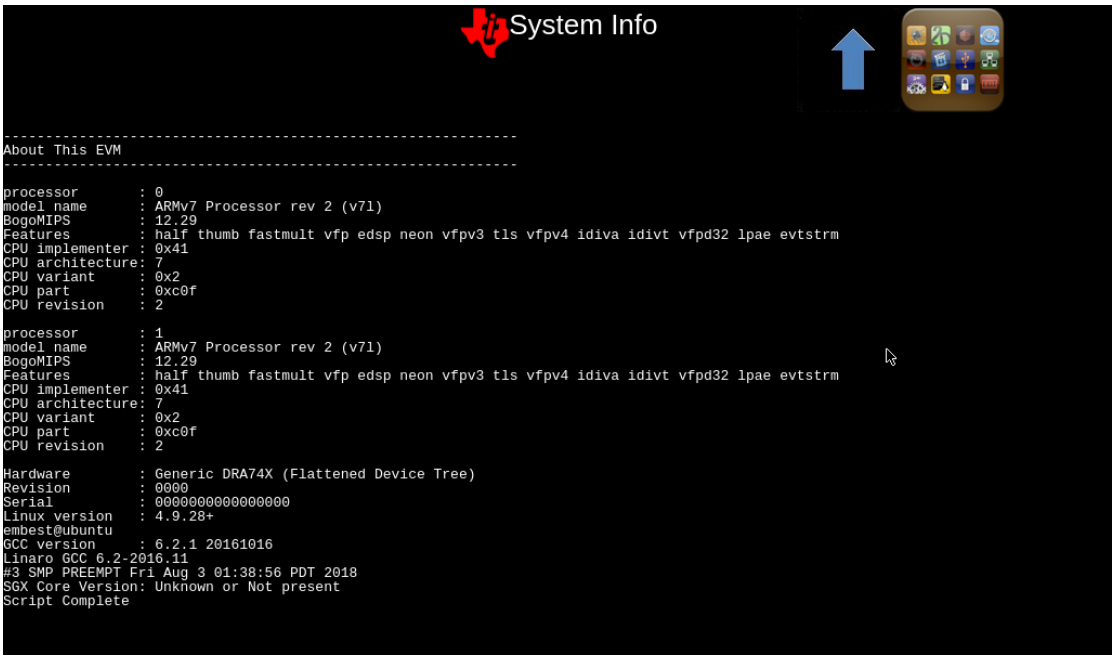
4.4.5 NETWORK SETTINGS

Click Network Settings to view network information, which is equivalent to the Linux command **ifconfig**:



4.4.6 SYSTEM INFO

Click System Info icon to view system information:

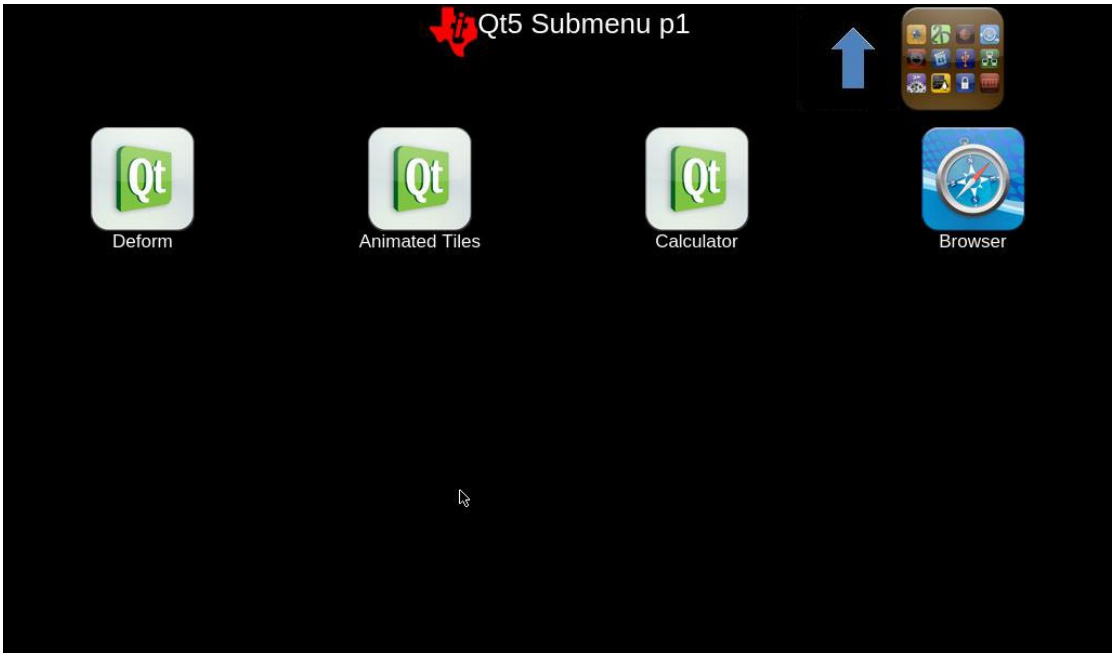


4.4.7 TASK INFO

Click Task Info icon to view process information, which is equivalent to the Linux command **ps**:

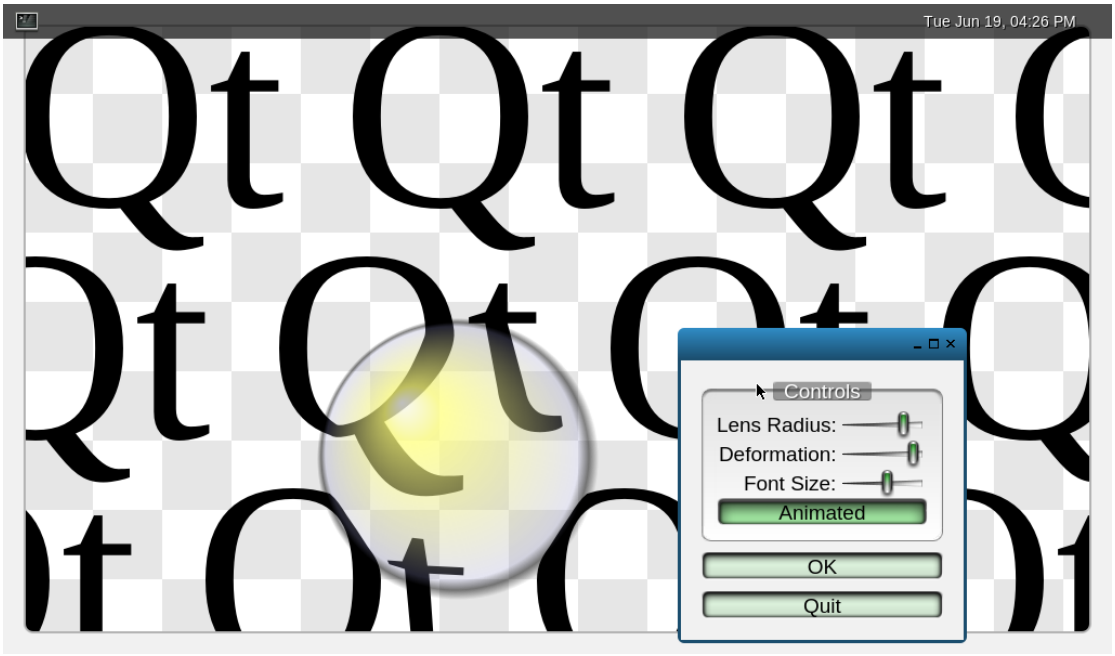
4.5 QT5 FUNCTION DEMONSTRATION

Click the QT5 button on the main interface to enter the following interface:



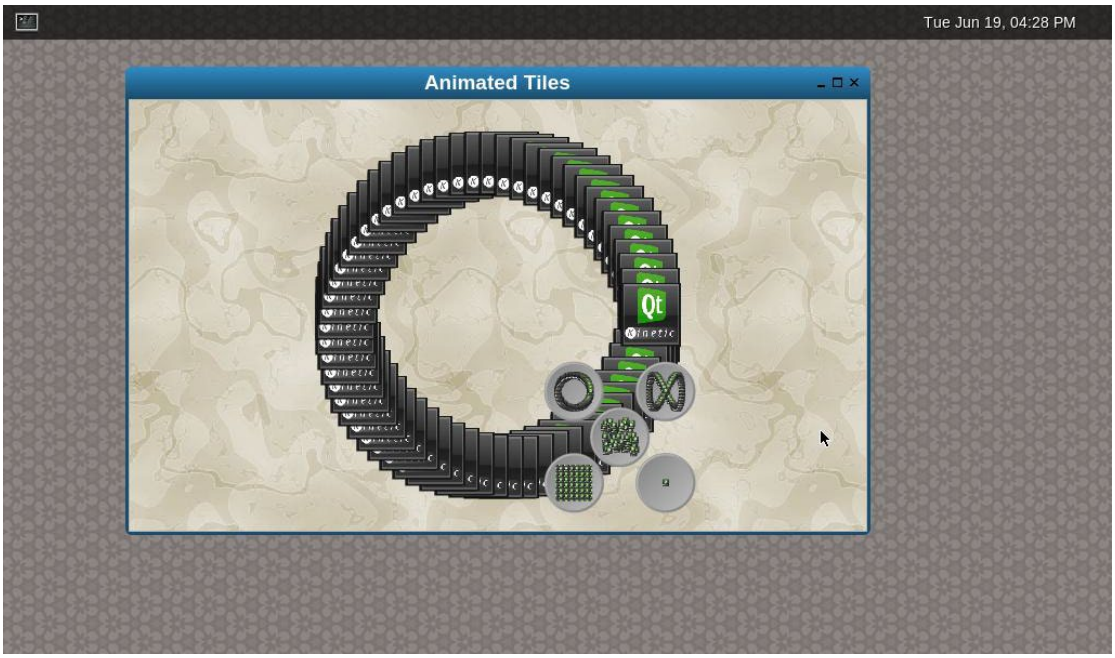
4.5.1 DEFORM

Right-click on the Deform interface and a small window of settings dialog will pop up:



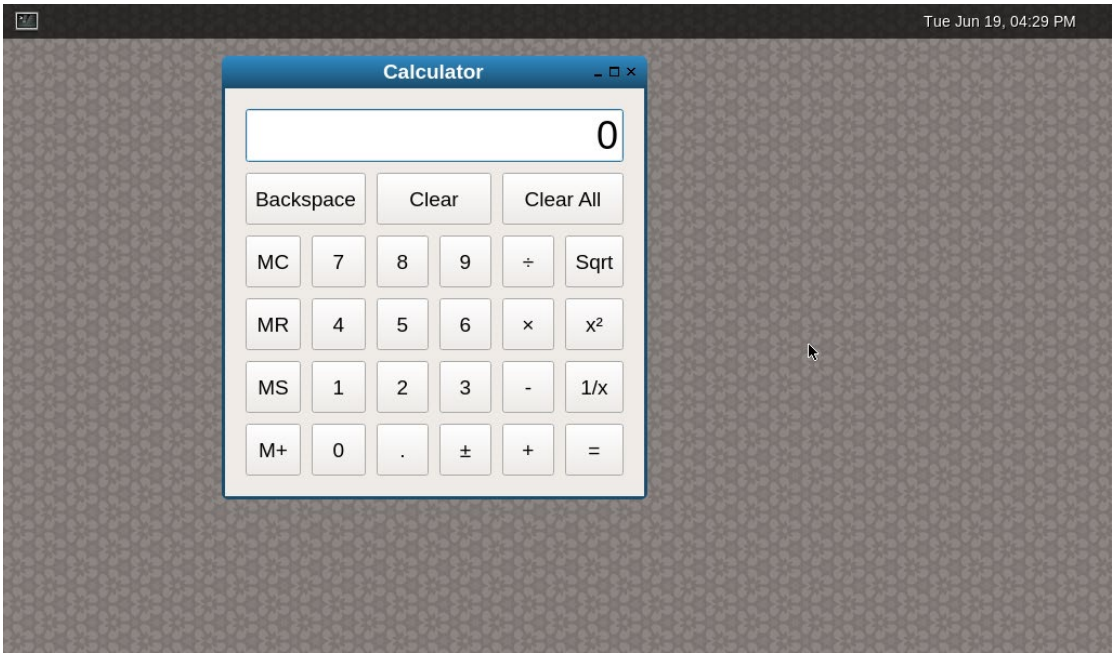
4.5.2 ANIMATED TILES

Click Animated Tiles and click the icon in the bottom right corner to choose different icon alignment shape.



4.5.3 CALCULATOR

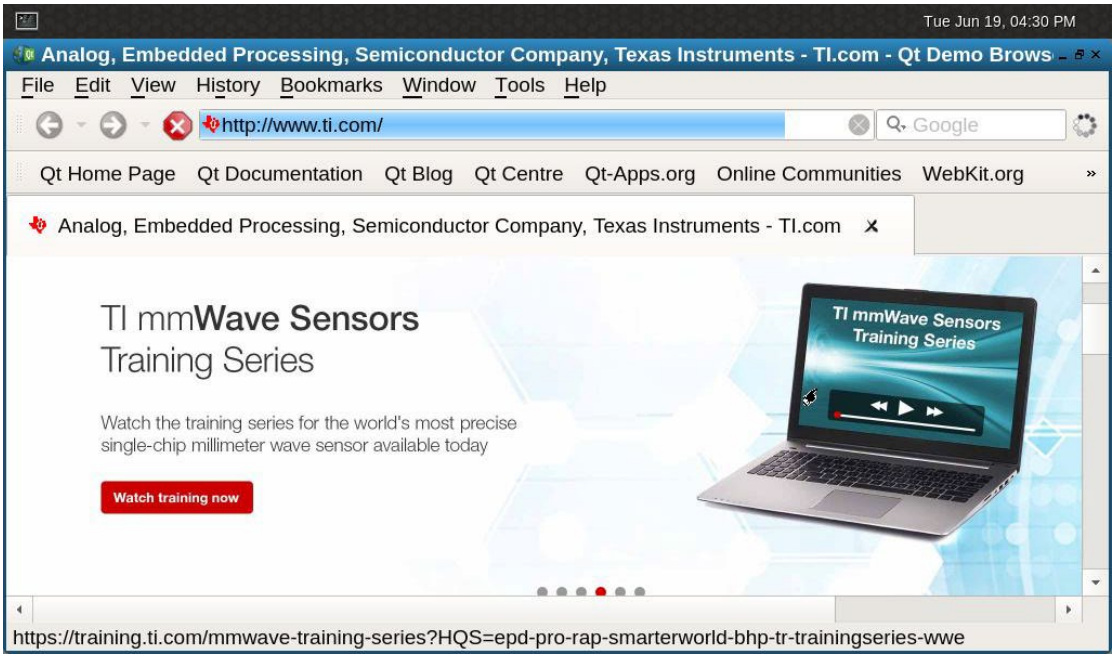
Click Calculator icon and a simple calculator program will pop up.



4.5.4 BROWSER

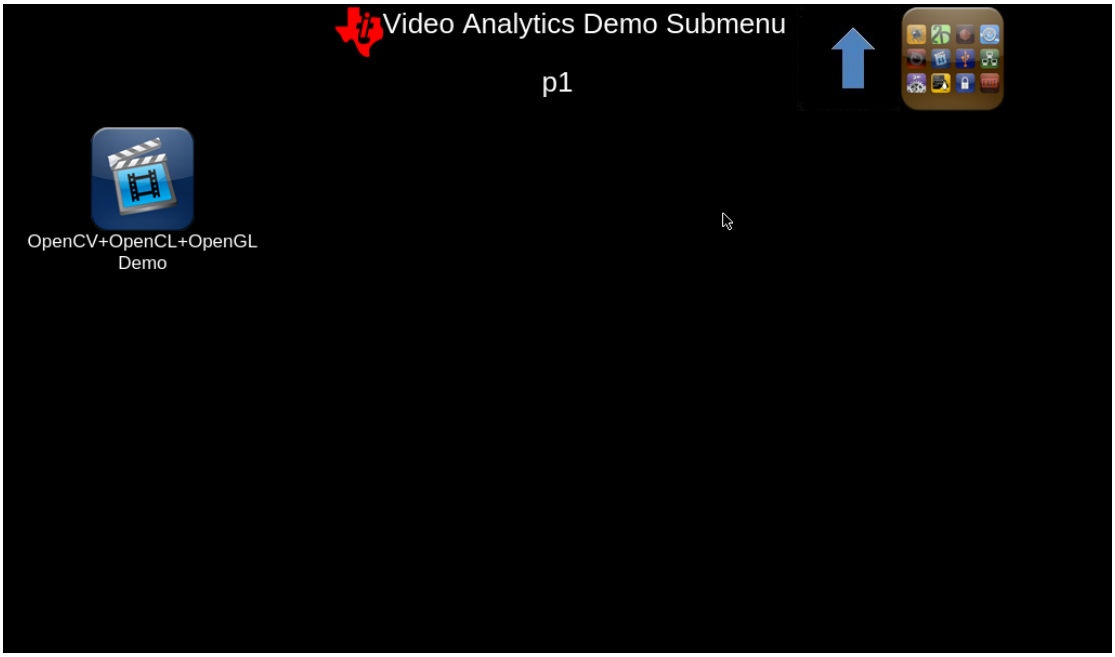
Click Browser, a browser will open, which will access the TI official website by default.

Note: you need to connect to Internet through wire or wireless network interfaces:

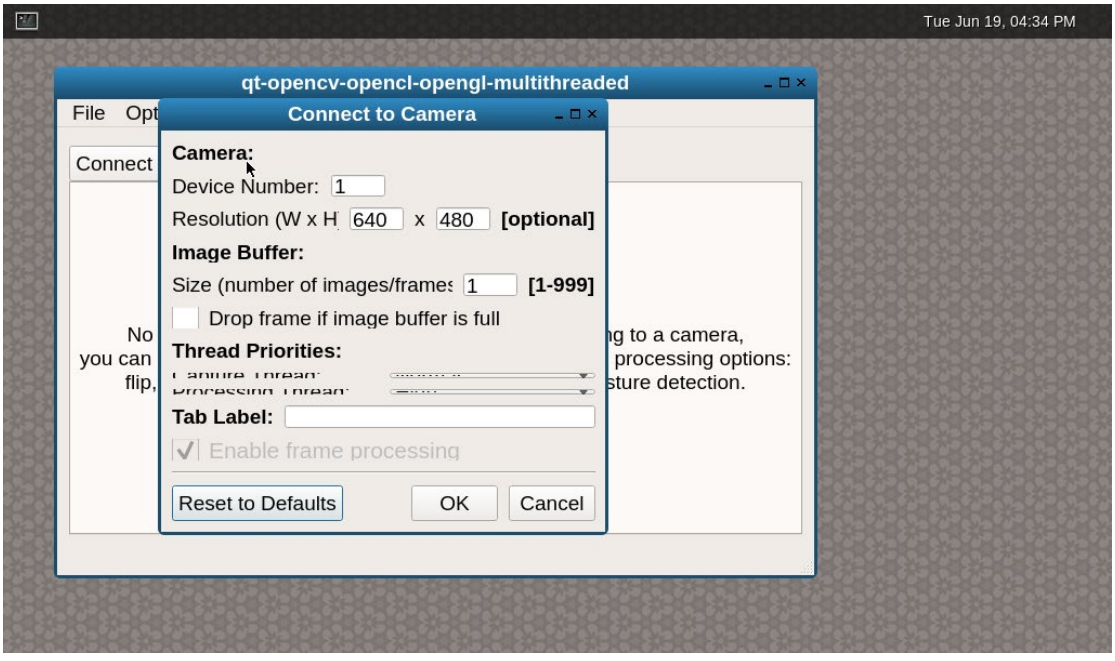

Tue Jun 19, 04:30 PM
Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com - Qt Demo Brows
File Edit View History Bookmarks Window Tools Help
http://www.ti.com/
Google
Qt Home Page Qt Documentation Qt Blog Qt Centre Qt-Apps.org Online Communities WebKit.org
Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com
TI mmWave Sensors
Training Series
Watch the training series for the world's most precise single-chip millimeter wave sensor available today
Watch training now
TI mmWave Sensors
Training Series
https://training.ti.com/mmwave-training-series?HQS=epd-pro-rap-smarterworld-bhp-tr-trainingseries-wwe

4.6 VIDEO ANALYTICS FUNCTION DEMONSTRATION

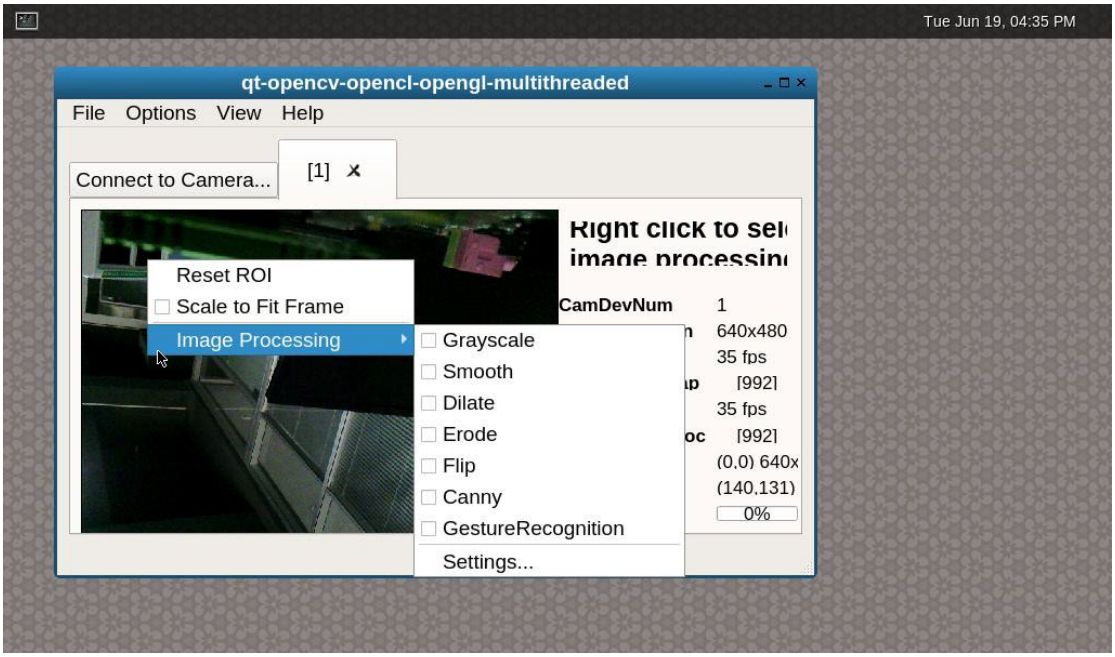
Click on the Video Analytics Demo icon.



Click OpenCV+OpenCL+OpenGL Demo icon, which is a Demo that integrates Camera, OpenCV, OpenCL and OpenGL. After clicking, you need to connect to the Camera first:

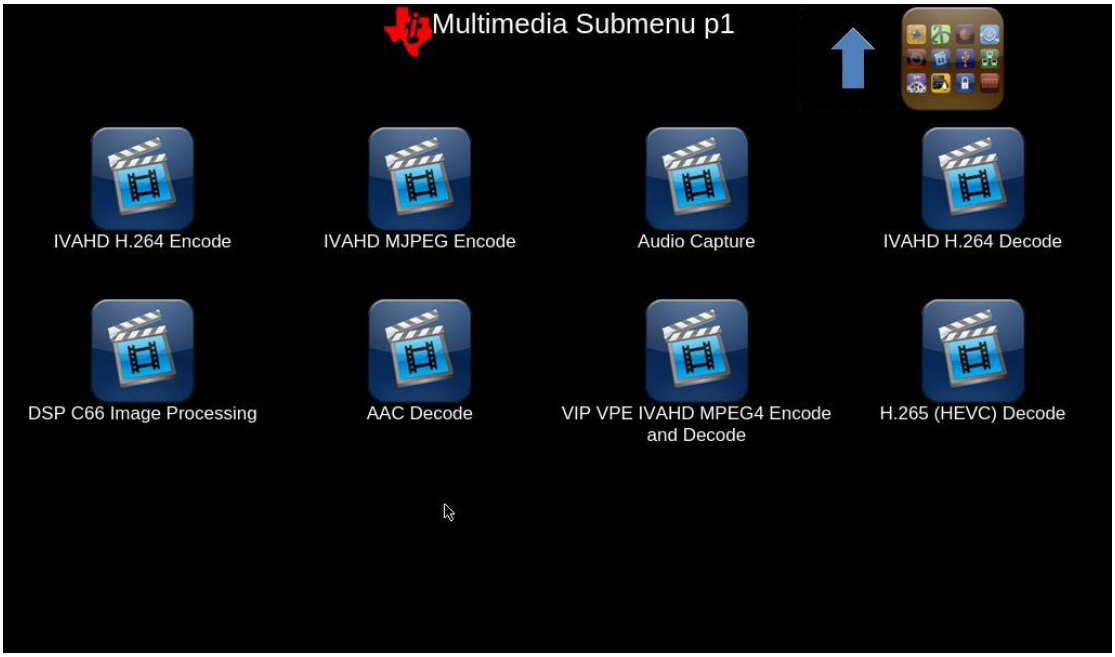


After setting the camera parameters and connecting to the camera, you can right-click on the image to process the camera image, including grayscale images, image flipping and gesture recognition, etc.:



4.7 MULTIMEDIA FUNCTION DEMONSTRATION

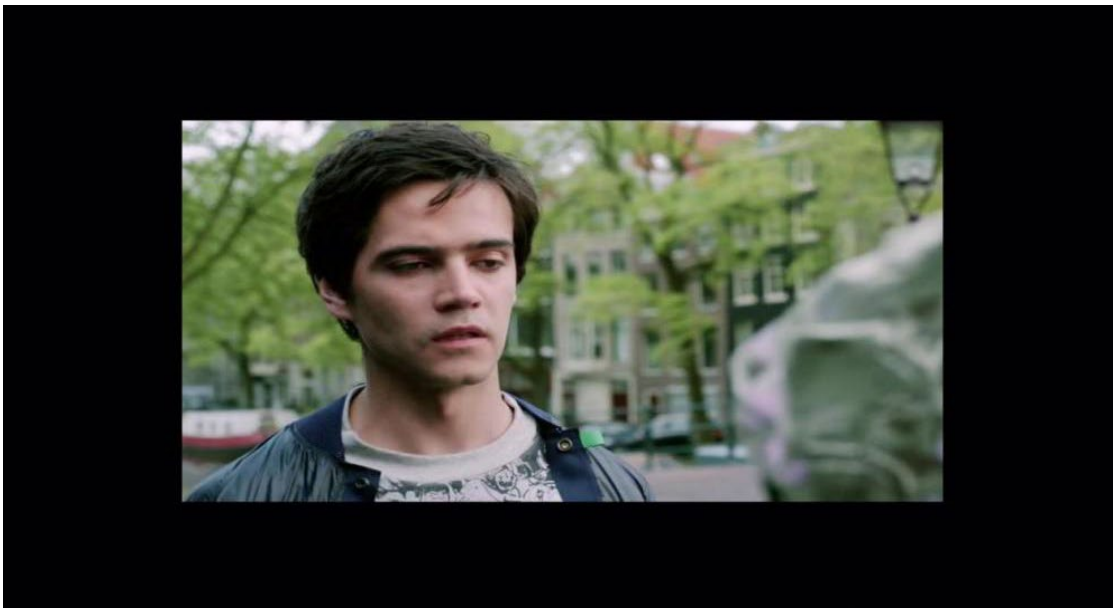
Click Multimedia on the main interface of Matrix GUI to enter the following directory:



Demo Name	Description
IVAHD H264 Decode	This demo runs a gstreamer playbin pipeline to decode H264 using IVAHD. The demo also plays audio, which can be listened to through headphones or speakers.
IVAHD H264 Encode	This demo program runs a gstreamer pipeline to perform H264 encoding on IVAHD. Input clips are in NV12 format. The output is saved to the /home/root directory
IVAHD MJPEG Encode	This demo program runs a gstreamer pipeline to perform MJPEG encoding on IVAHD. Input clips are in NV12 format. The output is saved to the /home/root directory
AAC Decode	Demo runs the gstreamer playbin pipeline for ARM audio decoding and playout.
H.265 (HEVC) Decode	Demonstrates HEVC decoding on ARM. The gstreamer pipeline decodes and displays the H265 stream.
VIP VPE IVAHD MPEG4 Encode and Decode	Demonstrates video capture of the Video Input Port (VIP), color space conversion and video processing engine (VPE), IVAHD MPEG4 encoding, IVAHD MPEG4 decoding and display scaling

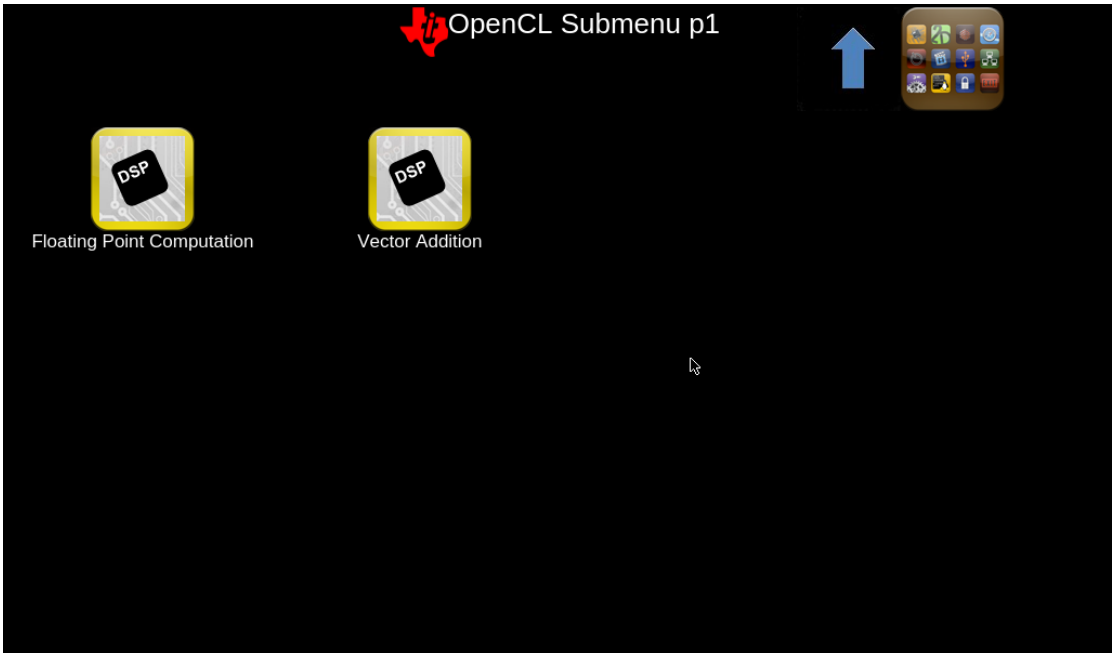
DSP C66 Image Processing	Demonstrates the use of the DSP C66x plug-in (dsp66videokernel) to offload image processing tasks to the DSP.

Click H.265 (HEVC) Decode and the H.265 video stream will be played.



4.8 OPENCL FUNCTION DEMONSTRATION

Click the OpenCL icon. There are two general calculations in this directory, named vector sum operations and floating point operations.

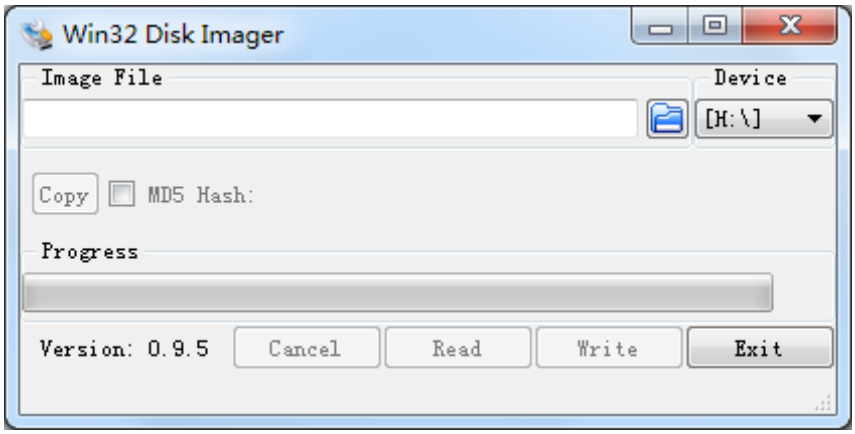


5. BURN-IN AND UPDATE SYSTEM IMAGE

5.1 BURN IMAGE TO SD CARD UNDER WINDOWS OS

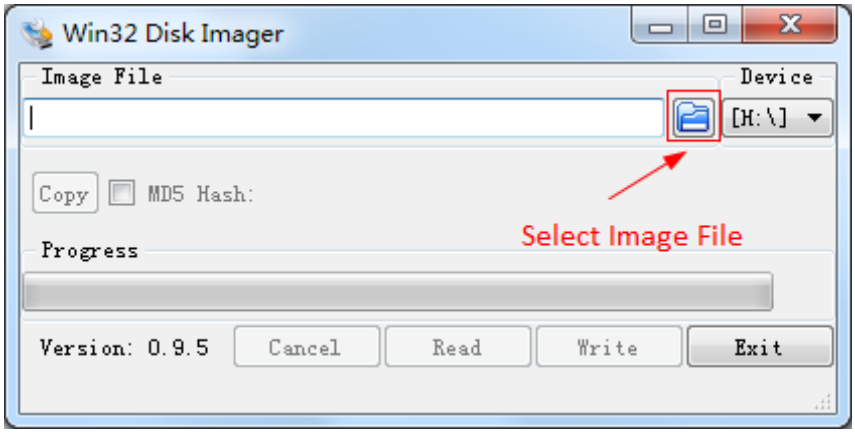
- First, prepare an SD card, its capacity shouldn't be smaller than 4GB.
- Then, download and install Win32 Disk Imager from

  https://sourceforge.net/projects/win32diskimager to PC.

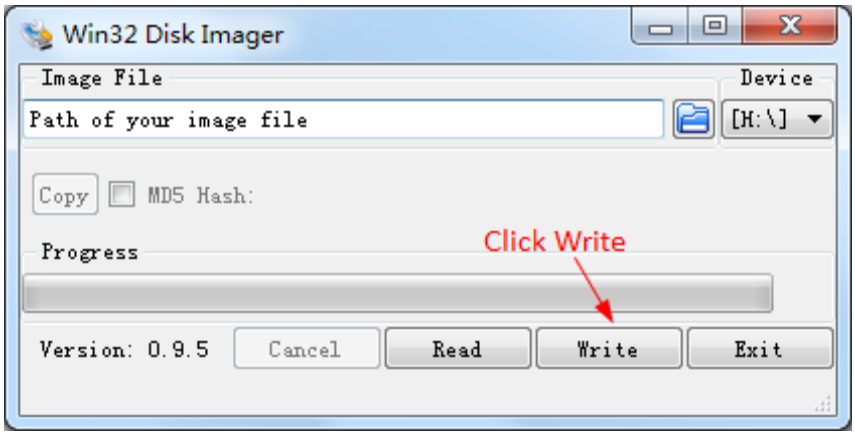


- Select the image to be burned, such as:

  EM-TF-EVK-AM5728-TI-ShipmentImage-SDcard-V1.0.4r05.img.



- Click Write to burn the image.



5.2 BURN IMAGE TO SD CARD UNDER LINUX OS

Under Ubuntu or Debian systems, you can use the **bmap-tool** tool to burn the image to the SD card. Take the image **EM-TF-EVK-AM5728-TI-ShipmentImage-SDcard-V1.0.4r05.img** as an example.

- Install bmap-tools.
  - **$ sudo apt install bmap-tools**
- Enter the following command to view the SD card node, in this case it's sdc.
  - **$ ls /dev/sdc***

```
/dev/sdc  /dev/sdc2  /dev/sdc1
```

- If the SD card is automatically mounted, you need to enter the following command to umount it.
  - **$ sudo umount /dev/sdc1**
  - **$ sudo umount /dev/sdc2**
- Start to burn-in.
  - **$ bmaptool create -o burn.map EM-TF-EVK-AM5728-TI-ShipmentImage-SDcard-V1.0.4r05.img**
  - **$ sudo bmaptool copy --bmap burn.map EM-TF-EVK-AM5728-TI-ShipmentImag**

e-SDcard-V1.0.4r05.img /dev/sdc

5.3 UPDATE EMMC

Take the eMMC image **EM-TF-EVK-AM5728-TI-ShipmentImage-EMMC-V1.0.4r05.img** as an example: copy the image file to a USB flash drive. Refer to Section 2.2, first boot the system from the SD card, and then install the USB disk into the USB interface (J15/J16):

- root@arm:~# **ls /dev/sd***

```
/dev/sda  /dev/sdb  /dev/sdb1
```

- root@arm:~# **mount /dev/sdb1 /mnt/**
- root@arm:~# **dd if=/mnt/EM-TF-EVK-AM5728-TI-ShipmentImage-EMMC-V1.0.4r05.img of=/dev/mmcblk1**

Note: The programming time takes 10~25 minutes, please wait patiently...

After programming completes, power off, set the DIP switches S6 and S7 according to the table below, remove the SD card, and power on.

BOOT CONFIG	
BOOT0	L
BOOT1	L
BOOT2	L
BOOT3	L
BOOT4	L
BOOT5	H