

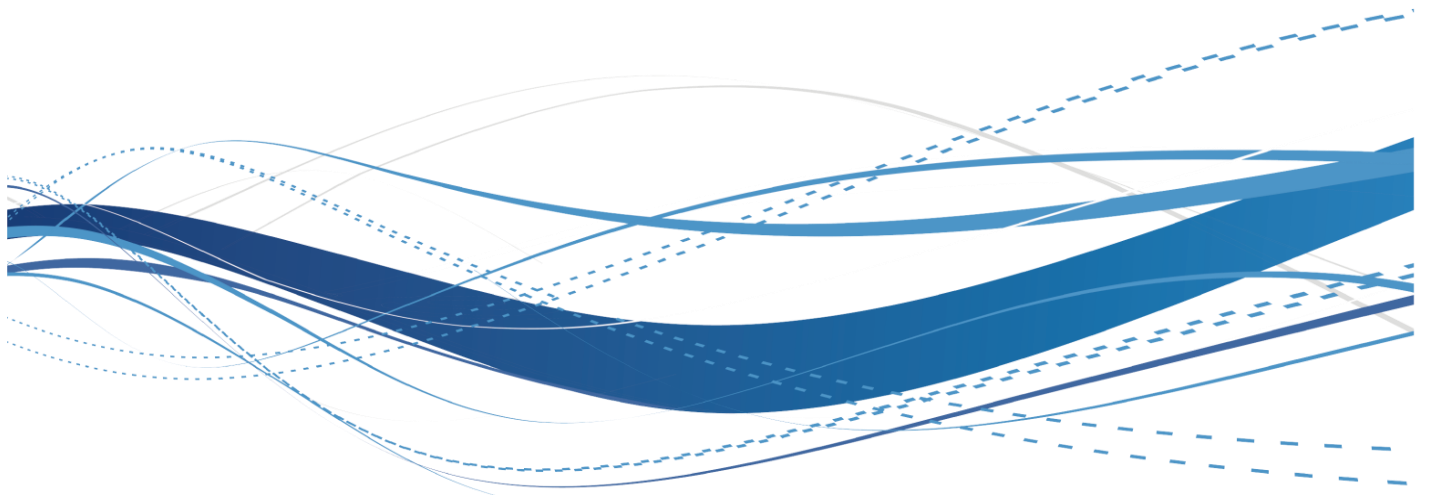


深圳市英蓓特科技公司

# EM-TF-EVK-AM5728

## Linux 软件用户手册

V1.0



## 版权声明

- ◆ EM-TF-EVK-AM5728 评估板及其相关知识产权由深圳市英蓓特科技有限公司所有。
- ◆ 本文档由深圳市英蓓特科技有限公司版权所有，并保留一切权利，在未经英蓓特公司书面许可的情况下，不得以任何形式来修改、分发或复制本文档的任何部分。

## 免责声明

- ◆ 产品附带提供的程序源代码、软件、资料文档等，深圳市英蓓特有限公司不提供任何类型的担保；不论是明确的，还是隐含的，包括但不限于合适特定用途的保证，全部的风险，由使用者来承担。

## 版本记录

版本	描述	作者	日期
V0.1	草稿	Jerry/David	20180806
V1.0	正式发布	Sandy	20180827

# 目录

版本记录 .....	3
目录 .....	4
第 1 章 介绍 .....	7
1.1 软件包介绍 .....	7
1.2 功能列表 .....	8
第 2 章 快速启动 .....	9
2.1 从 eMMC 启动系统 .....	9
2.2 从 SD 卡启动系统 .....	11
第 3 章 功能的配置与说明 .....	12
3.1 USER LED .....	12
3.2 蜂鸣器 .....	12
3.3 按键 .....	12
3.4 继电器 .....	13
3.4.1 LS1 .....	13
3.4.2 LS2 .....	13
3.5 光耦 .....	13
3.5.1 OC1 .....	13
3.5.2 OC2 .....	14
3.6 RTC .....	14
3.7 eMMC .....	14
3.8 QSPI_Flash .....	15
3.9 显示 .....	16
3.9.1 选择显示设备 .....	16
3.9.2 LCD 屏幕亮度调节 .....	16
3.10 触摸屏 .....	16
3.11 音频 .....	17
3.11.1 板载音频接口 .....	17
3.11.2 HDMI 音频播放 .....	18
3.12 UART .....	18
3.12.1 UART1 .....	18
3.12.2 UART10 .....	18

3.12.3	RS485 .....	19
3.13	板载千兆网口 .....	19
3.13.1	配置静态 IP .....	19
3.13.2	配置自动获取 IP .....	19
3.14	PCIE 网卡 .....	19
3.14.1	PCIE 网卡的使用 .....	错误! 未定义书签。
3.15	4G .....	20
3.15.1	4G 网络 .....	20
3.15.2	4G 电话 .....	21
3.15.3	4G 短信 .....	21
3.16	CAN .....	21
3.17	USB 2.0/3.0 接口 .....	21
3.17.1	Host 接口 .....	21
3.18	Camera .....	22
3.19	SATA 接口 .....	23
3.20	Wi-Fi .....	23
3.20.1	连接 Wi-Fi 网络 .....	23
3.21	Bluetooth .....	23
3.21.1	初始化蓝牙模块 .....	23
3.21.2	扫描蓝牙设备 .....	24
第 4 章	Matrix 界面体验 .....	25
4.1	开机界面 .....	25
4.2	ARM 功能演示 .....	26
4.3	3D 功能演示 .....	27
4.4	Setting 功能演示 .....	28
4.4.1	Terminal .....	28
4.4.2	Memory .....	29
4.4.3	System Shutdown .....	29
4.4.4	Matrix Shutdown .....	30
4.4.5	Network Settings .....	31
4.4.6	System Info .....	31
4.4.7	Task Info .....	32
4.5	Qt5 功能演示 .....	33
4.5.1	Deform .....	34
4.5.2	Animated Tiles .....	34
4.5.3	Calculator .....	35
4.5.4	Browser .....	35

4.6	Video Analytics Demo 功能演示 .....	36
4.7	Multimedia 功能演示.....	38
4.8	OpenCL 功能演示.....	39
第 5 章	烧写和更新系统镜像 .....	40
5.1	在 Windows 环境下烧写镜像到 SD 卡 .....	40
5.2	在 Linux 环境下烧写镜像到 SD 卡.....	41
5.3	更新 eMMC 中的镜像 .....	41
第 6 章	附录 .....	42
6.1	硬件 .....	42
第 7 章	技术支持和保修服务 .....	43
7.1	技术支持 .....	43
7.2	保修服务 .....	43
第 8 章	联系方式 .....	44

# 第1章 介绍

## 1.1 软件包介绍

软件发布包的内容以实际发布为准，发布文件结构及说明请参考下表：

Release Folder tree

```

├── 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 功能列表

- ◆ 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

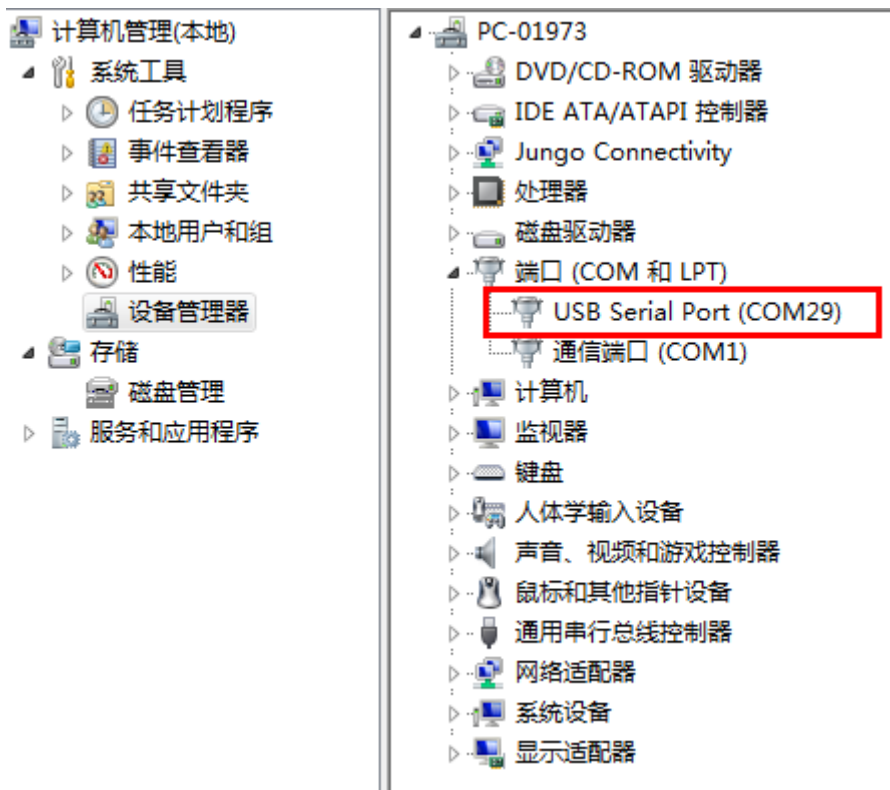


## 第2章 快速启动

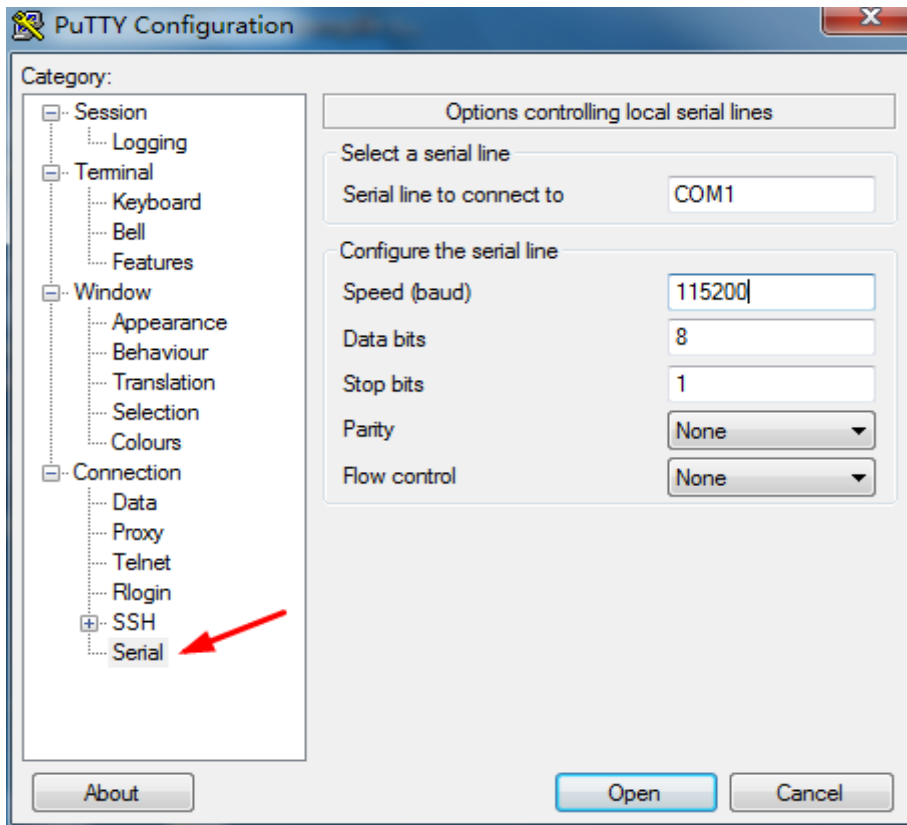
EM-TF-EVK-AM5728 默认已烧写镜像到 EMMC，可直接从 EMMC 启动，如需使用最新的系统镜像，请参考[第5章烧写和更新系统镜像](#)。关于电路连接和使用的配件等详细信息，请参考 QSG 文档

### 2.1 从 eMMC 启动系统

- ◆ 用 MiniUSB 线把板子上的 DEBUG 接口(CON1)和 PC 的 USB host 相连。
- ◆ 给板子上电，并安装 FTDI 的 USB 转 TTL 驱动(Tools 目录或访问 <http://www.ftdichip.com/Drivers/VCP.htm> 下载)，安装成功，设备管理器将显示 USB 串口的端口



- ◆ 在 PC 上安装串口软件（例如 PUTTY），选择正确的端口号，波特率 115200，8 位数据位，1 位停止位，无奇偶校验



- ◆ 参照下表设置拨码开关 S6 和 S7:

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

- ◆ 用 12V,2A 的电源，给板子供电(J6)，上电，系统将自动启动
- ◆ 如果板子上接了触摸屏，系统第一次启动的时候触摸屏会提示做校准，此时只需要根据屏幕提示点击相应的点。
- ◆ 系统启动完毕之后，串口终端输入用户名 `root` 登录。

```

Arago Project
Arago Project http://arago-project.org embest-am57xx tty02
Arago 2017.05 embest-am57xx tty02
embest-am57xx login: root
    
```

## 2.2 从 SD 卡启动系统

从 SD 卡启动系统的步骤和从 eMMC 启动的流程基本相同，在给板子上电启动前还需要执行下面 2 个操作：

- ◆ 把准备好的 SD 卡插入板上的插槽(J5)
- ◆ 参照下表设置拨码开关 S6 和 S7:

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

## 第3章 功能的配置与说明

首先, 请参考[第二章](#), 把系统启动起来, 然后跟随下面的指引使用 AM5728 的各项功能。

### 3.1 USER LED

用户能够控制 EM-TF-BB-AM57x 上的 D55,D56 两个单色 LED 指示灯 (分别对应 User LED1 和 User LED2)。在串口终端中执行以下命令来进行控制:

熄灭 LED:

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

点亮 LED:

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

### 3.2 蜂鸣器

EM-TF-BB-AM57X 底板上有一个蜂鸣器 PZ1

打开蜂鸣器:

```
root@embest-am57xx:~# echo 1 > /sys/class/misc/buzzer_ctl/state
```

关闭蜂鸣器:

```
root@embest-am57xx:~# echo 0 > /sys/class/misc/buzzer_ctl/state
```

### 3.3 按键

EM-TF-BB-AM57X 底板有三个按键 S3、S4、S5。其中 S3 是复位按键, S4 和 S5 是两个用户按键:

通过下面的命令, 按下 S4 或者 S5 后会有一串数字打印出来则为正常:

```
root@embest-am57xx:~# 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 继电器

EM-TF-BB-AM57X 底板上有两个继电器 LS1 和 LS2，这两个继电器分别由 GPIO5\_4 和 GPIO7\_30 控制。

### 3.4.1 LS1

配置控制引脚 GPIO5\_4:

```
root@embest-am57xx:~# echo 132 > /sys/class/gpio/export
root@embest-am57xx:~# echo out > /sys/class/gpio/gpio132/direction
```

打开继电器 LS1，J19 DO2+脚和 DO2-脚导通:

```
root@embest-am57xx:~# echo 1 > /sys/class/gpio/gpio132/value
```

关闭继电器 LS1，J19 DO2+脚和 DO2-脚断开:

```
root@embest-am57xx:~# echo 0 > /sys/class/gpio/gpio132/value
```

### 3.4.2 LS2

配置控制引脚 GPIO7\_30:

```
root@embest-am57xx:~# echo 222 > /sys/class/gpio/export
root@embest-am57xx:~# echo out > /sys/class/gpio/gpio222/direction
```

打开继电器 LS2，J19 DO1+脚和 DO1-脚导通:

```
root@embest-am57xx:~# echo 1 > /sys/class/gpio/gpio222/value
```

关闭继电器 LS2，J19 DO1+脚和 DO1-脚断开:

```
root@embest-am57xx:~# echo 0 > /sys/class/gpio/gpio222/value
```

## 3.5 光耦

EM-TF-BB-AM57X 底板上有两个光耦 OC1 和 OC2，这两个光耦分别由 GPIO4\_11 和 GPIO4\_10 控制。

### 3.5.1 OC1

配置控制引脚 GPIO4\_11:

```
root@embest-am57xx:~# echo 107 > /sys/class/gpio/export
root@embest-am57xx:~# echo in > /sys/class/gpio/gpio107/direction
```

读取当前状态，1 表示光耦未导通

```
root@embest-am57xx:~# cat /sys/class/gpio/gpio107/value
```

```
1
```

J19 DI2+ DI2-接入分别接入 5V 电源和 GND，注意 DI2+接正极，DI2-接负极，读取 GPIO 状态

```
root@embest-am57xx:~# cat /sys/class/gpio/gpio107/value
```

```
0
```

读到 0 表示光耦导通

### 3.5.2 OC2

配置控制引脚 GPIO4\_10:

```
root@embest-am57xx:~# echo 106> /sys/class/gpio/export
root@embest-am57xx:~# echo out > /sys/class/gpio/gpio106/direction
```

读取当前状态, 1 表示光耦未导通

```
root@embest-am57xx:~# cat /sys/class/gpio/gpio106/value
1
```

J19 DI1+ DI1-接入分别接入 5V 电源和 GND, 注意 DI1+接正极, DI1-接负极, 读取 GPIO 状态

```
root@embest-am57xx:~# cat /sys/class/gpio/gpio106/value
0
```

读到 0 表示光耦导通

## 3.6 RTC

RTC 时间由电容保持, 在串口终端输入:

查看当前时间:

```
root@embest-am57xx:~# date
Tue Jul 3 00:09:37 UTC 2018
```

设置当前时间, 如: 2018 年 6 月 12 日 10 时 15 分:

```
root@embest-am57xx:~# date 061210152018
Tue Jun 12 10:15:00 UTC 2018
```

把系统时钟写入 RTC:

```
root@embest-am57xx:~# hwclock -w
```

读取 RTC:

```
root@embest-am57xx:~# hwclock
Tue Jun 12 10:16:03 2018 0.000000 seconds
```

可以看到, 硬件时钟 RTC 被设置成 2018 年 6 月 12 日, 系统时钟被保存到硬件时钟里。

重启系统并查看时间:

```
root@embest-am57xx:~# date
Tue Jun 12 10:17:19 UTC 2018
```

## 3.7 eMMC

EM-TF-BB-AM57XX 的板载 eMMC 存储器在文件系统中的节点为/dev/mmcblk1。

系统默认启动方式是从 eMMC 启动, 如何更新 eMMC 中的镜像, 请参考开发手册。

下面的示例介绍了, 从 SD 卡启动时, 对 eMMC 进行简单的读写操作。

在串口终端执行:

```
root@embest-am57xx:~# touch emmc_read emmc_write
```

用 vi 命令编辑 emmc\_write:

```
root@embest-am57xx:~# vi emmc_write
```

例如写入 “emmc write test”

写 eMMC 命令:

```
root@embest-am57xx:~# dd if=emmc_write of=/dev/mmcblk1
```

```
0+1 records in
```

```
0+1 records out
```

读 eMMC 命令:

```
root@embest-am57xx:~# dd if=/dev/mmcblk1 of=emmc_read bs=1K count=10
```

```
10+0 records in
```

```
10+0 records out
```

查看 emmc\_read:

```
root@embest-am57xx:~# cat emmc_read
```

```
emmc write test
```

## 3.8 QSPI\_Flash

可以通过以下的命令查看 QSPI\_Flash 分区信息:

```
root@embest-am57xx:~# 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"
```

下面介绍向 QSPI\_Flash 进行读写测试:

使用下面的命令将 MLO 写入 QSPI\_Flash 的第一分区:

```
root@embest-am57xx:~# dd if=/run/media/mmcblk0p1/MLO of=/dev/mtdblock0
```

```
231+1 records in
```

```
231+1 records out
```

然后从第一分区读出 MLO 大小的字节到文件 MLO.test, MLO 文件的大小可以通过下面的命令获取:

```
root@embest-am57xx:~# ls /run/media/mmcblk0p1/MLO -l
```

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

读出第一分区内容到 MLO.test

```
root@embest-am57xx:~# dd if=/dev/mtdblock0 of=MLO.test bs=1 count=118578
118578+0 records in
118578+0 records out
```

比较两个文件的差别:

```
root@embest-am57xx:~# diff MLO.test /run/media/mmcbk0p1/MLO
root@embest-am57xx:~#
```

没有返回信息, 表示两个文件相同, 即读写成功。

## 3.9 显示

当前的系统支持 HDMI 和 LVDS 屏这 2 种显示设备, 可以参照下表连接屏幕到 AM5728, 然后启动系统。默认的显示设备是 HDMI 屏幕。

屏幕类型	接口编号
HDMI 屏 (默认显示屏)	J13 (标准 HDMI 接口)
LVDS 屏	J10

### 3.9.1 选择显示设备

显示设备可以由通过修改 uEnv.txt 文件中 fdt\_file 参数进行切换

修改方法:

启动后, 用 vi 命令修改 uEnv.txt 文件, 执行 sync 同步, 然后重启系统使修改生效。

使用 HDMI 屏显示:

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

使用 LVDS 显示:

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

### 3.9.2 LCD 屏幕亮度调节

LVDS 屏幕支持屏幕背光亮度调节,

LVDS 屏的亮度设置范围为 0-8, 0 表示亮度最低, 8 表示亮度最高, 在串口终端下输入如下命令调节亮度:

最暗:

```
root@embest-am57xx:~# echo 0> /sys/class/backlight/backlight/brightness
```

最亮:

```
root@embest-am57xx:~# echo 8> /sys/class/backlight/backlight/brightness
```

## 3.10 触摸屏

连接触摸屏模块到 J10, 屏的触摸接口有两个, 分别是 I2C 和 USB。I2C 触摸接底板的 J11, USB 触摸接底板的 J12。



### 3.10.1 I2C 触摸屏

接入触摸屏后，在串口终端输入以下命令：

```
root@embest:~# evtest /dev/input/event0
```

点击屏幕，串口终端中出现打印信息：

```
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
```

注意：按 **Ctrl+C** 退出程序

### 3.10.2 USB 触摸屏

接入触摸屏后，在串口终端输入以下命令：

```
root@embest:~# evtest /dev/input/event1
```

点击屏幕，串口终端中出现打印信息：

```
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
```

注意：按 **Ctrl+C** 退出程序

## 3.11 音频

### 3.11.1 板载音频接口

板上带音频输入接口 J18、输出接口 J17，支持录放音，连接录放音设备（如 3.5mm 的双插头耳机）后，可使用如下命令录制和播放音频：

录制音频之前之前需要进行配置：

```
root@embest-am57xx:~# amixer -c AM5728EmbestTLV sset 'Left PGA Mixer Mic2L' on
root@embest-am57xx:~# amixer -c AM5728EmbestTLV sset 'Right PGA Mixer Mic2R' on
root@embest-am57xx:~# amixer -c AM5728EmbestTLV sset 'PGA' 40
root@embest-am57xx:~# amixer -c AM5728EmbestTLV sset 'Output Driver Power-On time' 200ms
```

录制音频文件 K：

```
root@embest-am57xx:~# arecord -t wav -c 1 -r 44100 -f S16_LE -v k
```

播放音频文件 K:

```
root@embest-am57xx:~# aplay -t wav -c 2 -r 44100 -f S16_LE -v k
```

本次发布提供了 22\_3, 32\_3, 44\_3 和 48\_3 共 4 个音频文件，分别对应 22K, 32K, 44.1K 和 48K 几种采样频率，可用下列命令播放：

```
root@embest-am57xx:~# aplay test_audio/22_3
```

### 3.11.2 HDMI 音频播放

设置显示设备为 HDMI 屏并启动系统，连接 HDMI 显示器及配套放音设备，播放音频文件

```
root@embest-am57xx:~# aplay -D plughw:1,0 test_audio/48_3
```

## 3.12 UART

EM-TF-EVK-AM5728 底板共引出 6 路串口，各功能如下。

UART3 为 debug 串口。

UART7 & UART9 连接了底板的 RS485 收发芯片。

UART1 & UART10 引出到底板 J31。

### 3.12.1 UART1

UART1 在系统中的节点为 /dev/ttyO0

用户可使用系统自带的 uart\_test 测试程序做自发自收测试。

短接底板 J31 第 28, 30 号 pin，在串口终端输入：

```
root@embest-am57xx:~# ./uart_test -d /dev/ttyO0 -b 115200
/dev/ttyO0 SEND: 1234567890
/dev/ttyO0 RECV 10 total
/dev/ttyO0 RECV: 1234567890
```

SEND 和 RECV 的结果相同，表示测试成功

**注意：**按 **Ctrl+C** 退出程序

### 3.12.2 UART10

UART10 在系统中的节点为 /dev/ttyO9

用户可使用系统自带的 uart\_test 测试程序做自发自收测试。

短接底板 J31 第 24, 26 号 pin，在串口终端输入：

```
root@embest-am57xx:~# ./uart_test -d /dev/ttyO9 -b 115200
/dev/ttyO9 SEND: 1234567890
/dev/ttyO9 RECV 10 total
/dev/ttyO9 RECV: 1234567890
```

SEND 和 RECV 的结果相同，表示测试成功

**注意：**按 **Ctrl+C** 退出程序

### 3.12.3 RS485

EM-TF-BB-AM57X 底板上有两路 RS485。可以利用我们提供的测试程序进行对接测试。

J22 插座 1、2、3 脚分别接 J23 插座的 1、2、3 脚。

UART9 后台接收：

```
root@embest-am57xx:~# ./uart_receive -d /dev/ttyO8 -b 115200 &
```

UART2 发送：

```
root@embest-am57xx:~# ./uart_send -d /dev/ttyO1 -b 115200
```

如果测试成功，UART9 收到字符“f”

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

```
f
```

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

## 3.13 板载千兆网口

连接网线到 J27 (eth0) 或者 J28 (eth1)。假设网线接到 J27，在串口终端中输入以下命令来设置 IP 地址：

(以下 IP 地址仅为范例)

```
root@embest-am57xx:~# ifconfig eth0 192.168.52.64
```

网络测试：

```
root@embest-am57xx:~# ping 192.168.52.1
```

### 3.13.1 配置静态 IP

用 vi 修改/etc/network/interfaces，在 The primary network interface 段增加或修改参数如下：

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

修改后执行 sync，并重启系统使设置生效

### 3.13.2 配置自动获取 IP

同上，修改/etc/network/interfaces，删除 eth0 的 address，gateway，netmask 参数，只保留

```
auto eth0
iface eth0 inet manual
```

修改后执行 sync，并重启系统使设置生效

## 3.14 PCIE 网卡

EM-TF-BB-AM57X 底板上有一个 PCIE 接口 J14。

### 3.14.1 PCIE 网卡的使用

将 PCIE 网卡插入卡槽 J14，将网线接到 PCIE 网卡，然后启动系统，下面以 Realtek 的 PCIE 网卡为例，在串口终端中执行下列命令查看网卡信息。

```
root@embest-am57xx:~# lspci
00:00.0 PCI bridge: Texas Instruments Multicore DSP+ARM KeyStone II SOC (rev 01)
01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 09)
```

用下列命令查看 PCIE 网卡是否加载成功：

```
root@embest-am57xx:~# ifconfig -a

enp1s0  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:19 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)
```

上图中的 enp1s0 就是我们的 PCIE 网卡。

注:如果 J27、J28 连接网线，将其断开

为了让 PCIE 网卡能够自动获取 IP，需要修改下面的配置文件。在 eth\*后追加 enp\*

```
root@embest-am57xx:~# cat /etc/systemd/network/10-eth.network
[Match]
Name=eth* enp*
KernelCommandLine=!root=/dev/nfs

[Network]
DHCP=yes
```

注：目前仅支持 Realtek (RTL8111/8168/8144) 和 Intel (82574L) 的 PCIE 网卡。

## 3.15 4G

EM-TF-BB-AM57X 底板上的 CN2 接口可以安装 4G 模块 EC20（需外购）。SIM 卡安装到 J4 卡槽内。

### 3.15.1 4G 网络

安装好 4G 模块和 SIM 卡，启动系统以后，执行下面的命令连接 4G 网络：

```
root@embest-am57xx:~# pppd call quectel-ppp&
```

连接以后可以 ping 百度网址进行测试：

```
root@embest-am57xx:~# ping www.baidu.com -I ppp0
```

### 3.15.2 4G 电话

我们提供了一个 4G 测试的应用 4G\_test。执行下面的命令打电话：

```
root@embest-am57xx:~# ./4G_test -d -D /dev/ttyUSB2 -n 13100690411
```

打电话给号码 13100690411，延时一段时间后自动挂断。

注意：目前尚不支持通话功能。

### 3.15.3 4G 短信

执行下面的命令发一条短信到知道的电话号码：

```
root@embest-am57xx:~# ./4G_test -m -D /dev/ttyUSB2 -n 13100690411 -s "hello,world!"
```

发送一条短信，内容为“hello,world!”到号码为 13100690411 的手机。

## 3.16 CAN

下面将以 CAN0 和 CAN1 自发自收为例，介绍 CAN 的使用方法。

将 CAN0 和 CAN1 连接，即将 J2 的 27pin 与 28pin 对接，29pin 与 30pin 对接。

使用方法：

#### 1. 配置并打开 CAN0 CAN1

设置 CAN0 和 CAN1 的波特率为 50000

```
root@embest-am57xx:~# canconfig can0 bitrate 50000
```

```
root@embest-am57xx:~# canconfig can1 bitrate 50000
```

打开 CAN0 和 CAN1

```
root@embest-am57xx:~# canconfig can0 start
```

```
root@embest-am57xx:~# canconfig can1 start
```

#### 2. 发送与接收：

设置 CAN0 为后台接收。

```
root@embest-am57xx:~# candump can1&
```

CAN1 发送，CAN0 收到数据，表示通信正常。

```
root@embest-am57xx:~# cansend can1 -i 0x10 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
```

#### 3. 关闭 can0 can1

```
root@embest-am57xx:~# ip link set can0 down
```

```
root@embest-am57xx:~# ip link set can1 down
```

## 3.17 USB 2.0/3.0 接口

### 3.17.1 Host 接口

底板的 HUB1 有 3 个 USB Host 接口，两个 USB2.0 接口(J15)，一个 USB3.0(J16)，将 U 盘插入任意一个 USBHost 口，串口终端中将显示磁盘信息：

```

root@embest-am57xx:~# 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
    
```

在串口终端输入如下命令：

```

root@embest:~# ls /dev/sdb*
/dev/sdb  /dev/sdb1
    
```

/dev 下的设备节点 sdb1 即 U 盘，可以用 mount 命令挂载设备，进行后续操作。  
还可以连接其他 USB 设备，如键盘，鼠标等。

### 3.18 Camera

EM-TF-BB-AM57X 底板有两个并口摄像头接口 J7 和 J29，连接摄像头模块到 J7（目前只支持 OV5640），将 HDMI 屏幕接到底板的 HDMI 接口（J17），可以用下列两种方法进行预览测试：

#### 1. 在 Matrix GUI 上预览

在 Matrix GUI 上点击 Camera → Dual Camera Demo → RUN 图标

如果只接一个摄像头，在主屏幕上就会显示这个摄像头的画面，如果接两个摄像头，主屏幕会以画中画的形式显示两个摄像头的画面。

#### 2. 通过 gst-launch-1.0 进行预览

预览前需要关闭 weston

```

root@embest-am57xx:~# /etc/init.d/matrix-gui-2.0 stop
root@embest-am57xx:~# /etc/init.d/weston stop
    
```

然后输入以下命令进行预览：

```

root@embest-am57xx:~#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
    
```

上面的命令中，device 就是选择摄像头设备，width 和 height 设置分辨率，framerate 设置帧率，connector 设置显示设备。显示设备的 connector ID 可以通过下面的命令获取：

```

root@embest-am57xx:~# modetest -c
    
```

## 3.19 SATA 接口

EM-TF-BB-AM57X 底板支持挂载 mSATA 3.3V 供电的硬盘设备。将 mSATA 接到 CN1，再启动系统。  
在串口终端中执行以下命令查看硬盘的设备节点和分区信息：

```
root@embest-am57xx:~# ls /dev/sda*
/dev/sda
root@embest-am57xx:~# 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
```

支持对 SATA 进行分区，挂载，读写等操作。

## 3.20 Wi-Fi

板载 Wi-Fi 模块支持 2.4G/5G 联网功能，目前仅支持 STA 模式。

### 3.20.1 连接 Wi-Fi 网络

设置好无线路由器的 SSID 和 Password，然后再板子上创建配置文件：

```
root@embest-am57xx:~# wpa_passphrase Embest 12345678 > /etc/wpa_supplicant.conf
```

其中 Embest 是 SSID，12345678 是 Wi-Fi 密码。

然后连接 Wi-Fi：

```
root@embest-am57xx:~# wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant.conf -D nl80211
```

获取 IP 地址：

```
root@embest-am57xx:~# udhcpc -i wlan0 &
```

ping 测试

```
root@embest-am57xx:~# ping 192.168.1.1 -I wlan0
```

```
root@embest-am57xx:~# ping www.baidu.com -I wlan0
```

## 3.21 Bluetooth

### 3.21.1 初始化蓝牙模块

在串口终端中输入下列命令初始化蓝牙：

```
root@embest-am57xx:~# hciattach /dev/ttyO6 bcm43xx 115200 nflow
```

```
root@embest-am57xx:~# hciconfig hci0 up
```

### 3.21.2 扫描蓝牙设备

在串口终端中输入下列命令：

```
root@embest-am57xx:~# hcitool scan
Scanning ...
    94:87:E0:DF:90:2D 小米手机
```



## 第4章 Matrix 界面体验

### 4.1 开机界面

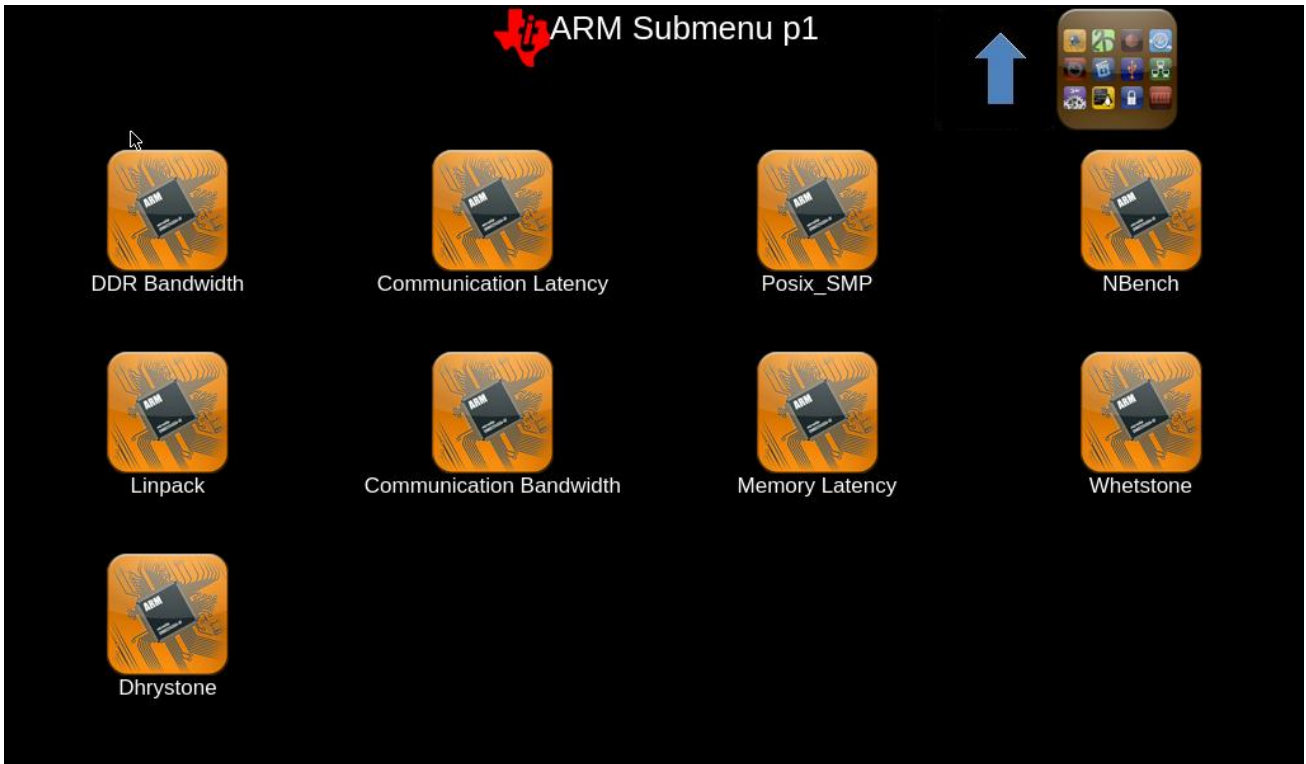
当系统完全起来以后，Matrix GUI 会自动启动，Matrix GUI 支持鼠标和触屏操作。该演示主要针对 Ti 官方提供的 Matrix 桌面(TI Matrix User Guide Wiki: [TI Matrix User Guide](#))。目前支持 ARM 性能测试、3D 效果、Settings 信息查看、Qt5 简单例程演示、OpenCL 计算能力、H265 解码演示。开发板开机之后即可看到 HDMI 上显示 Matrix 的主界面，也可以按照 [3.9.1](#) 介绍的切换到 LCD 输出显示：



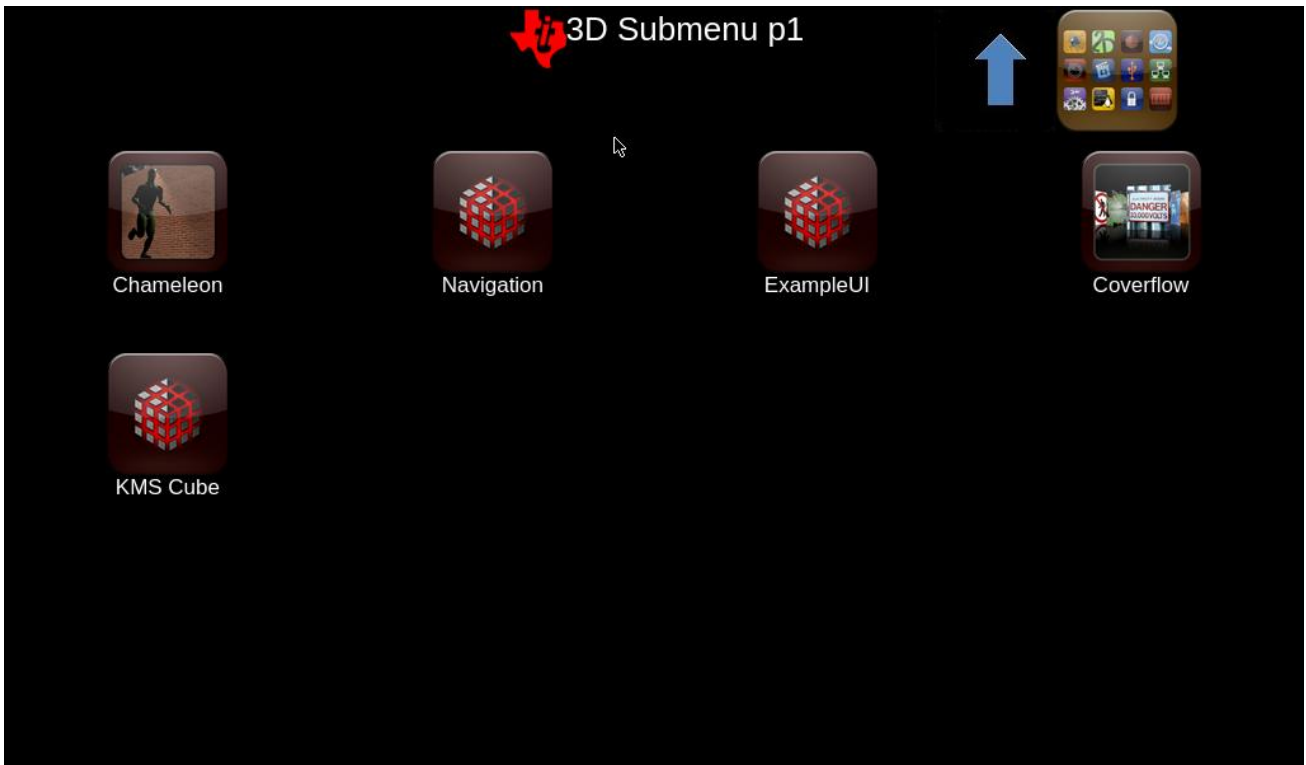
注意：Ethernet 功能，暂时不支持。

## 4.2 ARM 功能演示

点击 ARM 图标，进入 ARM 文件目录下，界面如下：



## 4.3 3D 功能演示



这个目录下有 5 个 3D 演示的例子。

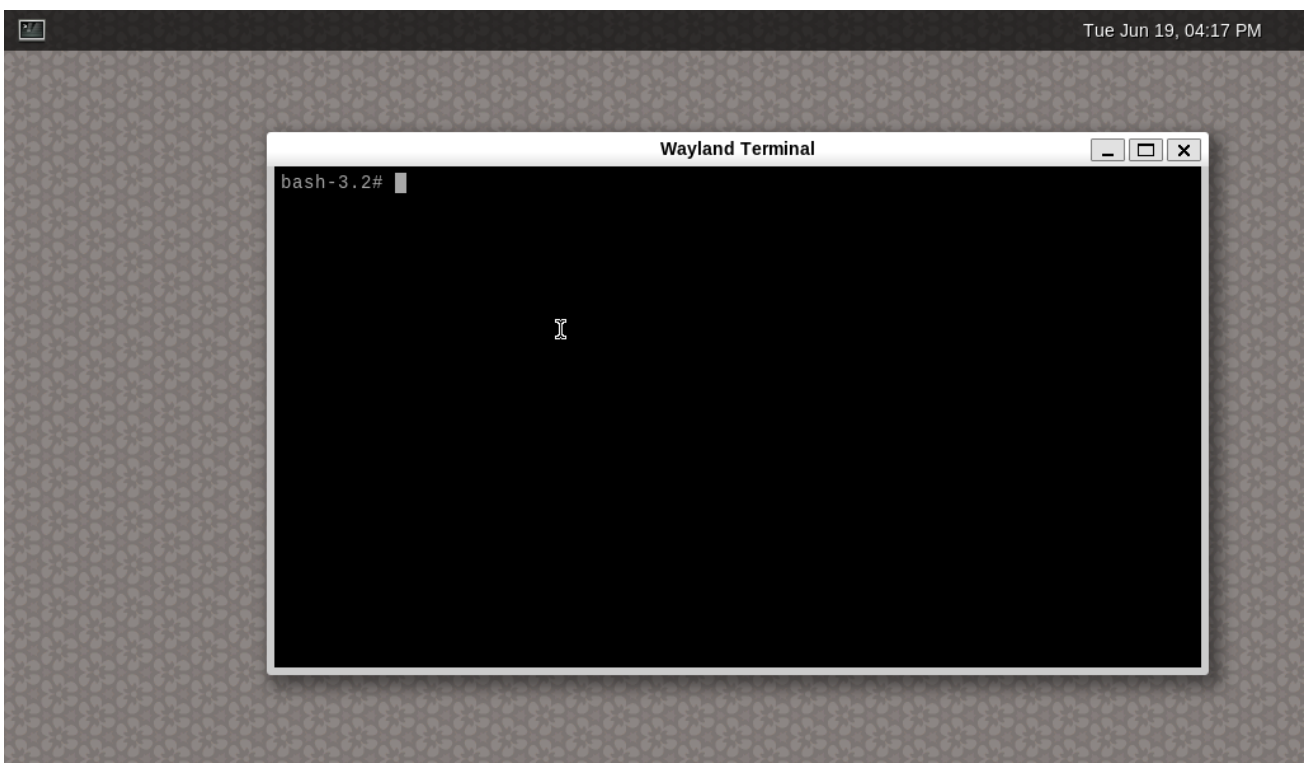
Demo Name	说明
ChameleonMan	演示了一个矩阵蒙皮字符结合凹凸贴图
CoverFlow	这是一个 Coverflow 样式效果的演示
ExampleUI	该演示展示了如何有效地渲染 sprites 和界面元素。
Navigation	演示如何为导航软件实现渲染算法的演示。
Kmscube	该演示展示了如何渲染和显示多色旋转立方体

## 4.4 Setting 功能演示



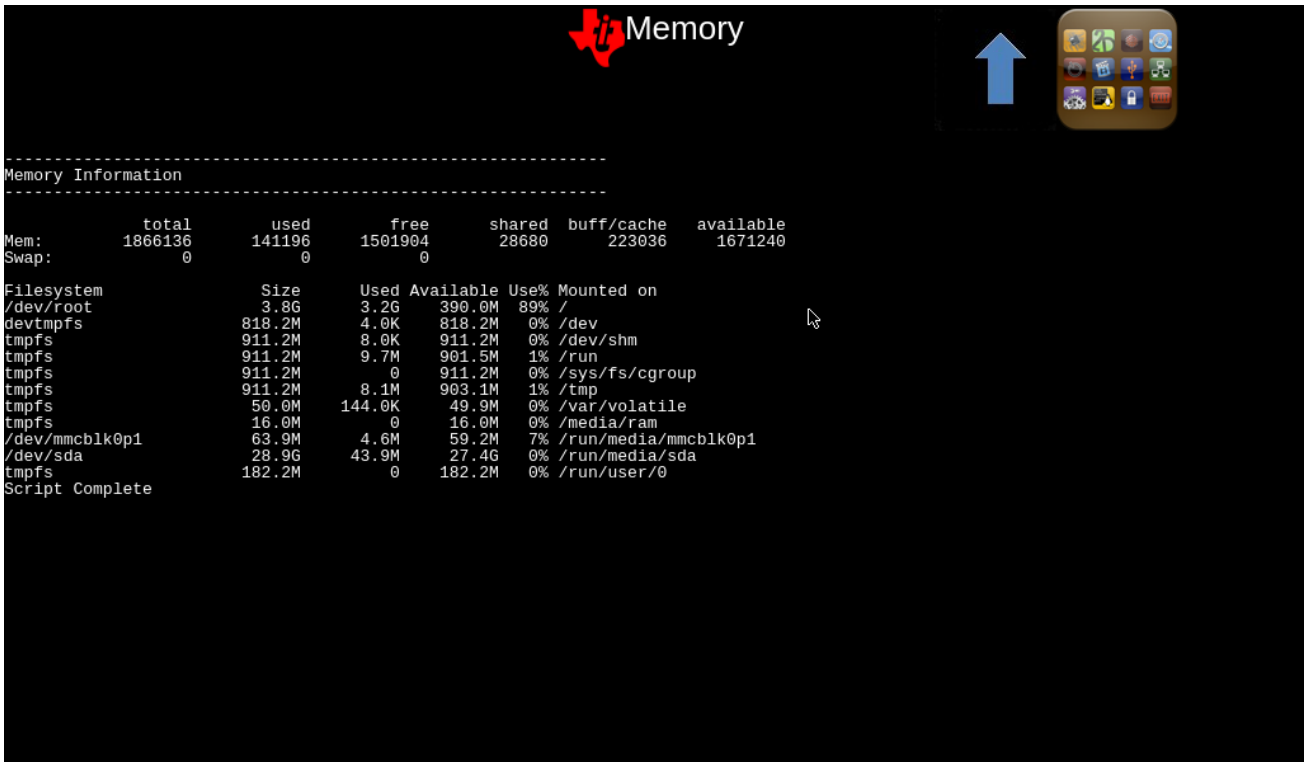
### 4.4.1 Terminal

点击 Terminal 图标会打开一个 Weston Terminal



## 4.4.2 Memory

点击 Memory 查看磁盘信息，相当于 Linux 命令 df:



```

-----
Memory Information
-----
Mem:          total      used      free      shared  buff/cache  available
Swap:         0          0         0          28680    223036     1671240

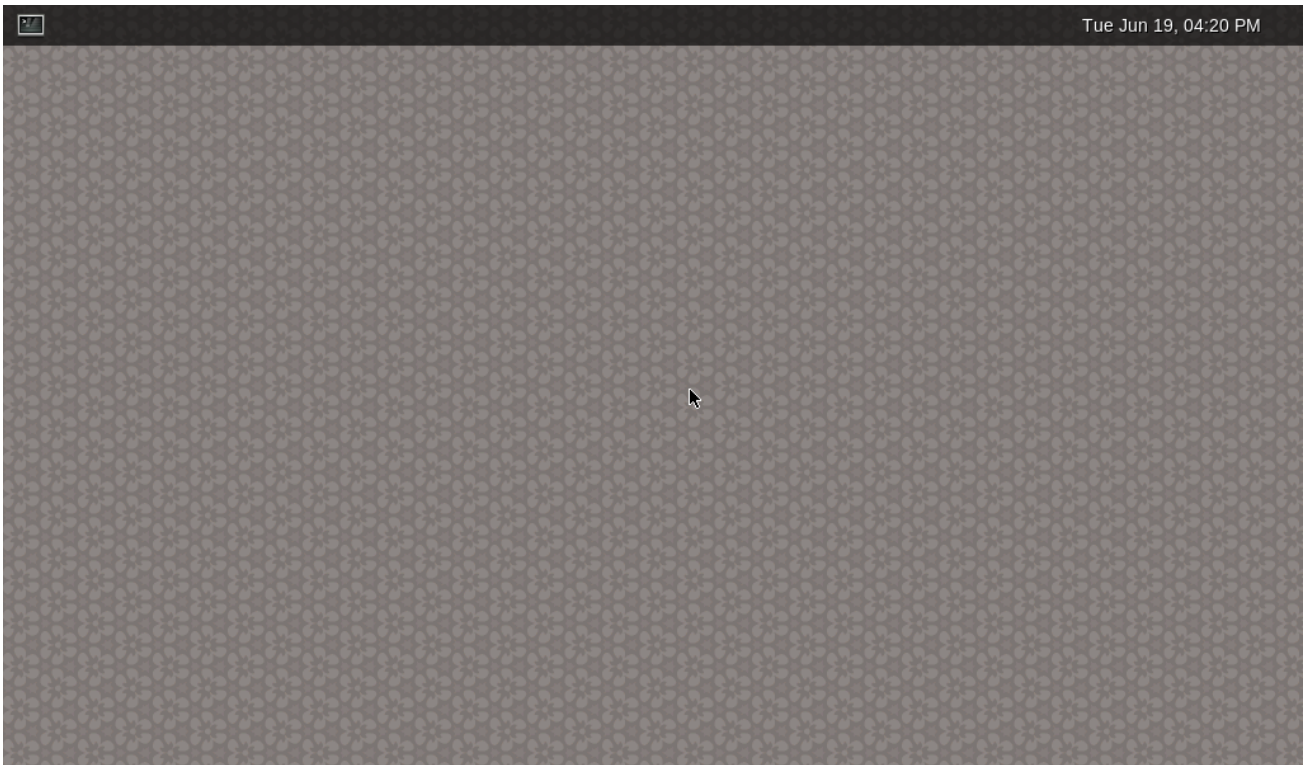
Filesystem      Size      Used Available Use% Mounted on
/dev/root       3.8G     3.2G   390.0M   89% /
devtmpfs       818.2M     4.0K   818.2M    0% /dev
tmpfs          911.2M     8.0K   911.2M    0% /dev/shm
tmpfs          911.2M     9.7M   901.5M    1% /run
tmpfs          911.2M     0       911.2M    0% /sys/fs/cgroup
tmpfs          911.2M     8.1M   903.1M    1% /tmp
tmpfs          50.0M    144.0K   49.9M    0% /var/volatile
tmpfs          16.0M     0       16.0M    0% /media/ram
/dev/mmcblk0p1 63.9M     4.6M   59.2M    7% /run/media/mmcblk0p1
/dev/sda       28.9G    43.9M   27.4G    0% /run/media/sda
tmpfs         182.2M     0     182.2M    0% /run/user/0
Script Complete
    
```

## 4.4.3 System Shutdown

点击 System Shutdown 系统会关机

#### 4.4.4 Matrix Shutdown

点击 Matrix Shutdown 将关闭图标矩阵，

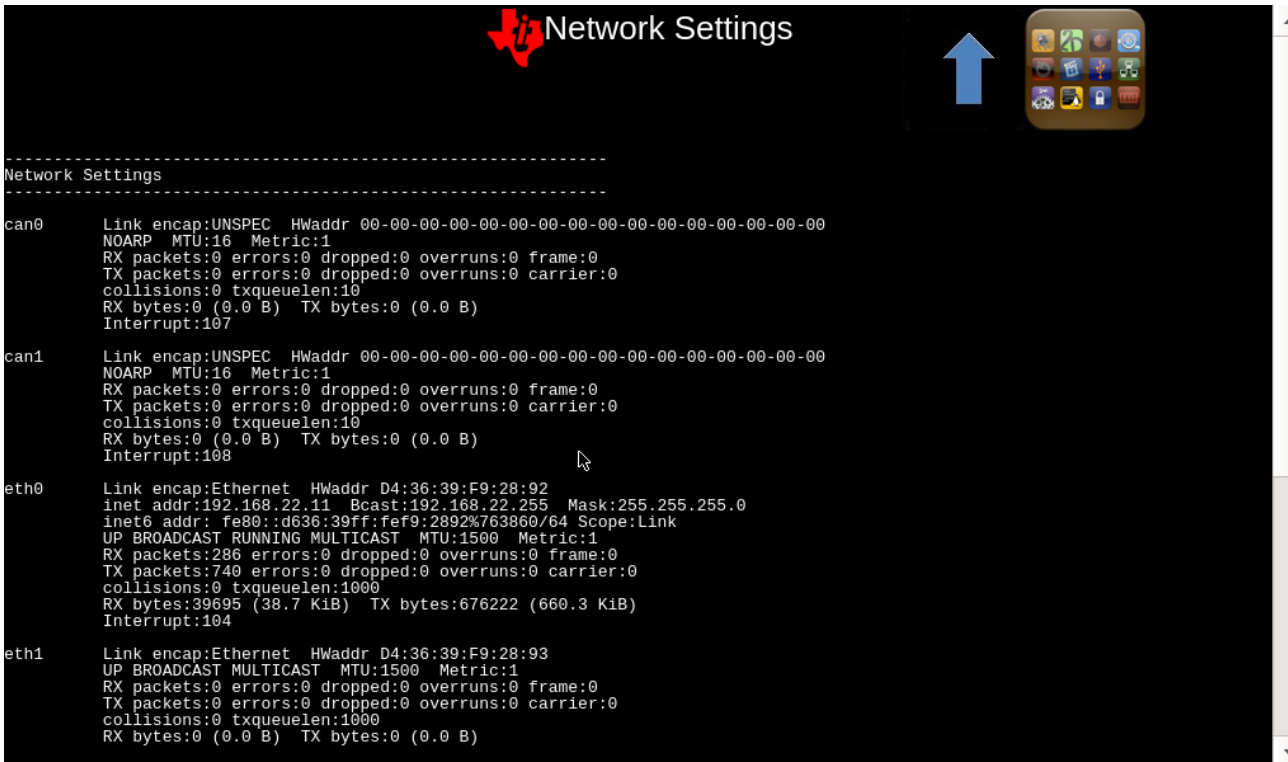


如需恢复 Matrix 界面需要运行下面的命令：

```
root@embest-am57xx:/# /etc/init.d/matrix-gui-2.0 start
```

#### 4.4.5 Network Settings

点击 Network Settings 查看网络信息，相当于 Linux 命令 ifconfig:



```

Network Settings
-----
can0    Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        NOARP MTU:16 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:10
        RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
        Interrupt:107

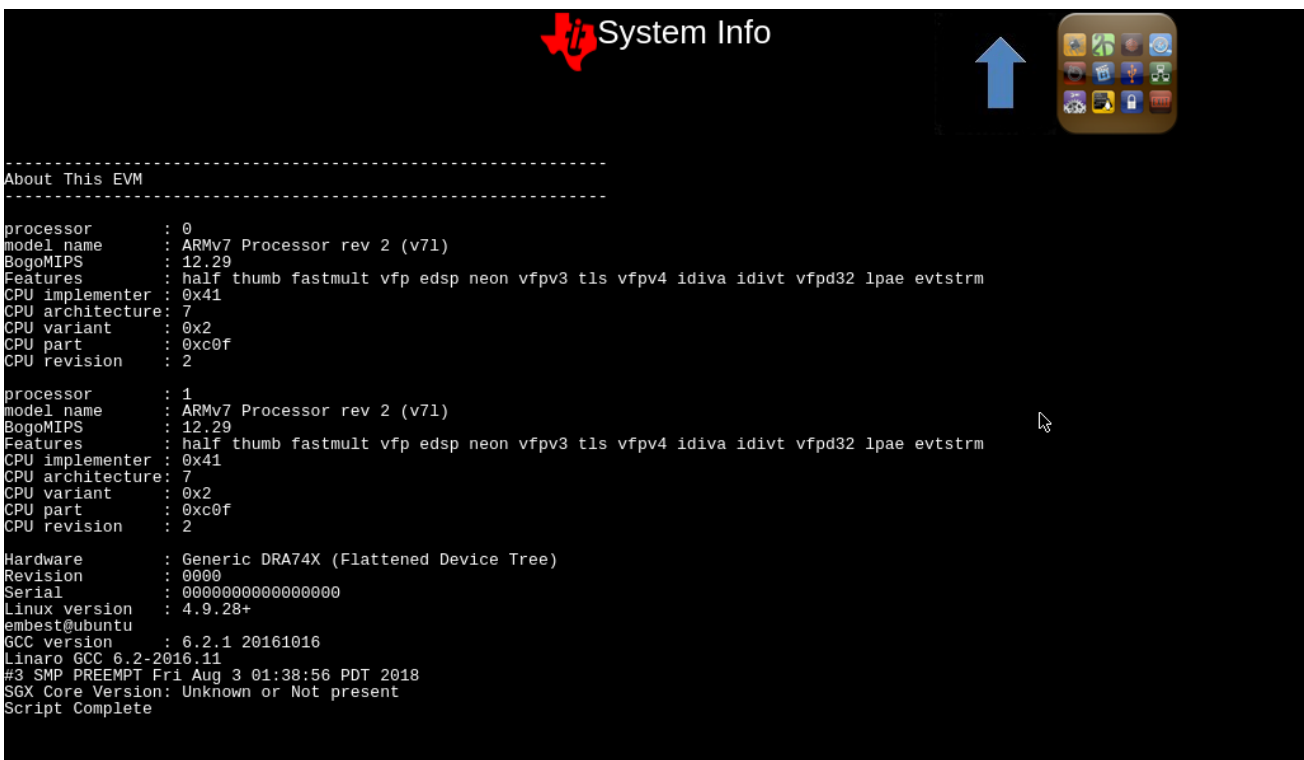
can1    Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        NOARP MTU:16 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:10
        RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
        Interrupt:108

eth0    Link encap:Ethernet HWaddr D4:36:39:F9:28:92
        inet addr:192.168.22.11 Bcast:192.168.22.255 Mask:255.255.255.0
        inet6 addr: fe80::d636:39ff:fe9:2892%763800/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:286 errors:0 dropped:0 overruns:0 frame:0
        TX packets:740 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:39695 (38.7 KiB) TX bytes:676222 (660.3 KiB)
        Interrupt:104

eth1    Link encap:Ethernet HWaddr D4:36:39:F9:28:93
        UP BROADCAST MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
    
```

#### 4.4.6 System Info

点击 System Info 查看系统信息:



```

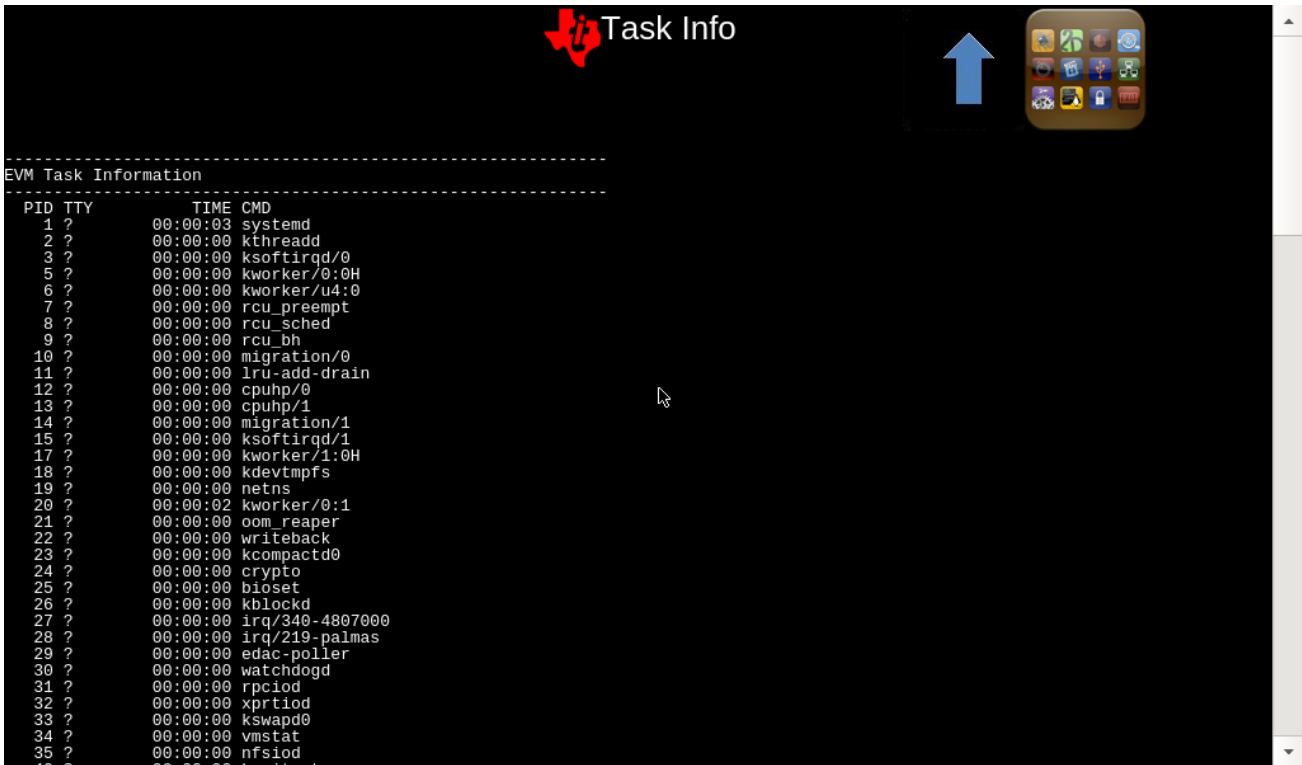
-----
About This EVM
-----
processor      : 0
model name    : ARMv7 Processor rev 2 (v7l)
BogoMIPS     : 12.29
Features     : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm
CPU implementer : 0x41
CPU architecture: 7
CPU variant   : 0x2
CPU part      : 0xc0f
CPU revision  : 2

processor      : 1
model name    : ARMv7 Processor rev 2 (v7l)
BogoMIPS     : 12.29
Features     : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm
CPU implementer : 0x41
CPU architecture: 7
CPU variant   : 0x2
CPU part      : 0xc0f
CPU revision  : 2

Hardware      : Generic DRA74X (Flattened Device Tree)
Revision     : 0000
Serial       : 0000000000000000
Linux version : 4.9.28+
embest@ubuntu
GCC version   : 6.2.1 20161016
Linaro GCC 6.2-2016.11
#3 SMP PREEMPT Fri Aug 3 01:38:56 PDT 2018
SGX Core Version: Unknown or Not present
Script Complete
    
```

#### 4.4.7 Task Info

点击 Task Info 查看进程信息，相当于 Linux 命令 ps:



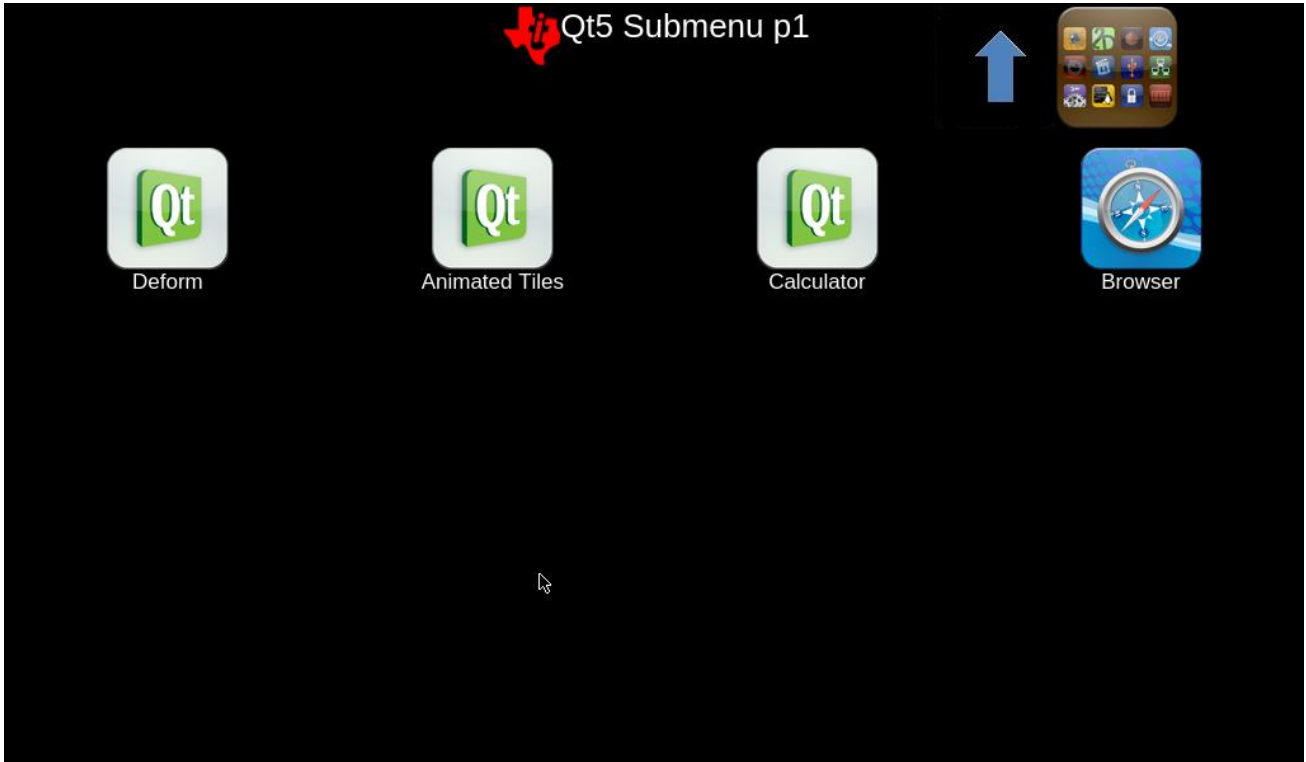
```

-----
EVM Task Information
-----
PID TTY          TIME CMD
  1 ?             00:00:03 systemd
  2 ?             00:00:00 kthreadd
  3 ?             00:00:00 ksoftirqd/0
  5 ?             00:00:00 kworker/0:0H
  6 ?             00:00:00 kworker/u4:0
  7 ?             00:00:00 rcu_preempt
  8 ?             00:00:00 rcu_sched
  9 ?             00:00:00 rcu_bh
 10 ?            00:00:00 migration/0
 11 ?            00:00:00 lru-add_drain
 12 ?            00:00:00 cpuhp/0
 13 ?            00:00:00 cpuhp/1
 14 ?            00:00:00 migration/1
 15 ?            00:00:00 ksoftirqd/1
 17 ?            00:00:00 kworker/1:0H
 18 ?            00:00:00 kdevtmpfs
 19 ?            00:00:00 netns
 20 ?            00:00:02 kworker/0:1
 21 ?            00:00:00 oom_reaper
 22 ?            00:00:00 writeback
 23 ?            00:00:00 kcompactd0
 24 ?            00:00:00 crypto
 25 ?            00:00:00 bioset
 26 ?            00:00:00 kblockd
 27 ?            00:00:00 irq/340-4807000
 28 ?            00:00:00 irq/219-palmas
 29 ?            00:00:00 edac-poller
 30 ?            00:00:00 watchdogd
 31 ?            00:00:00 rpciod
 32 ?            00:00:00 xpriod
 33 ?            00:00:00 kswapd0
 34 ?            00:00:00 vmstat
 35 ?            00:00:00 nfsiod
  
```



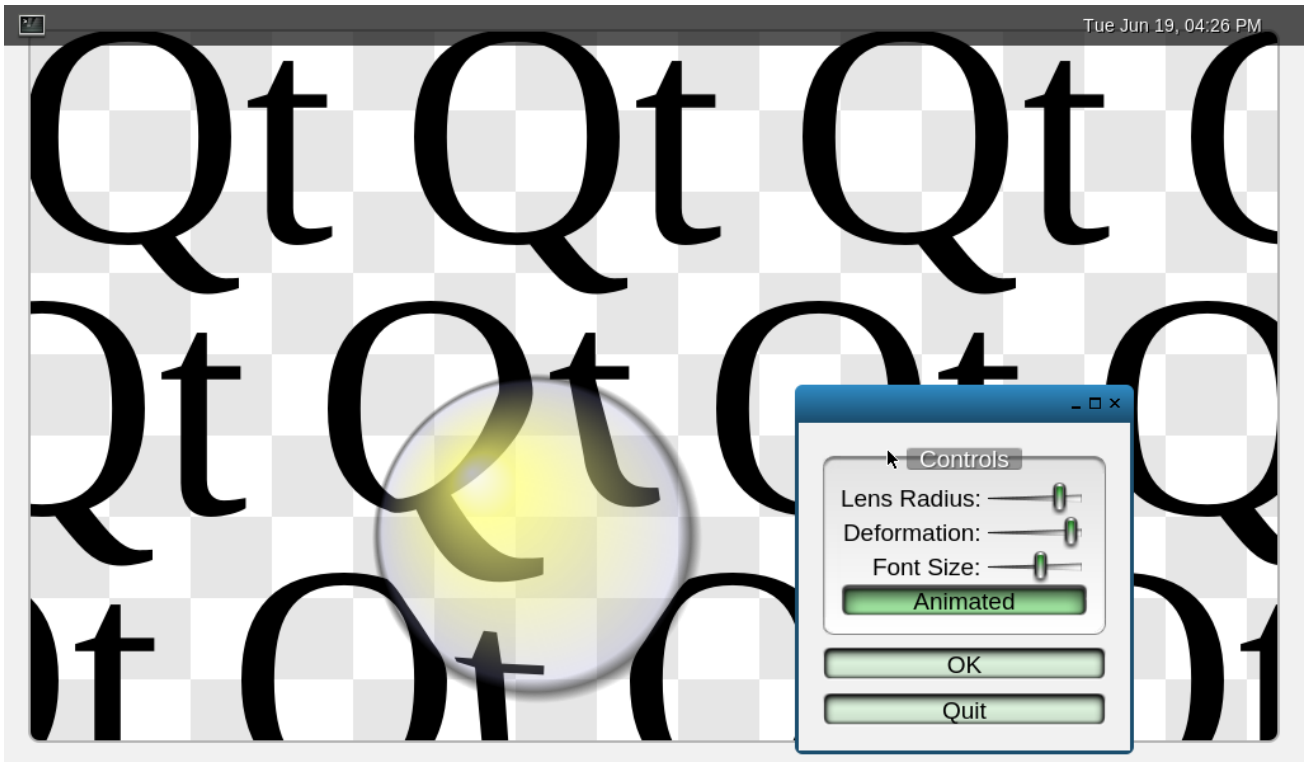
## 4.5 Qt5 功能演示

点击主界面的 QT5 按钮进入以下界面：



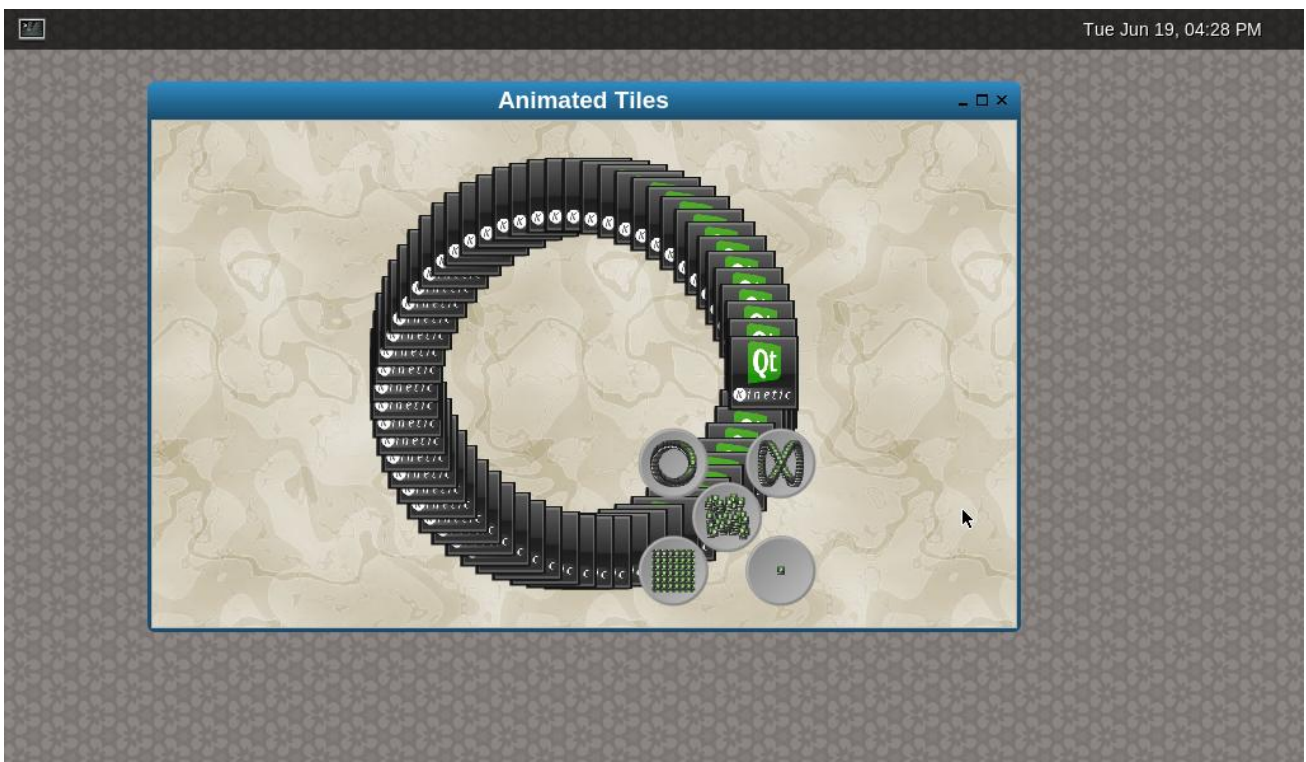
### 4.5.1 Deform

右键点击 Deform 界面会出现一个小窗口的设置界面：



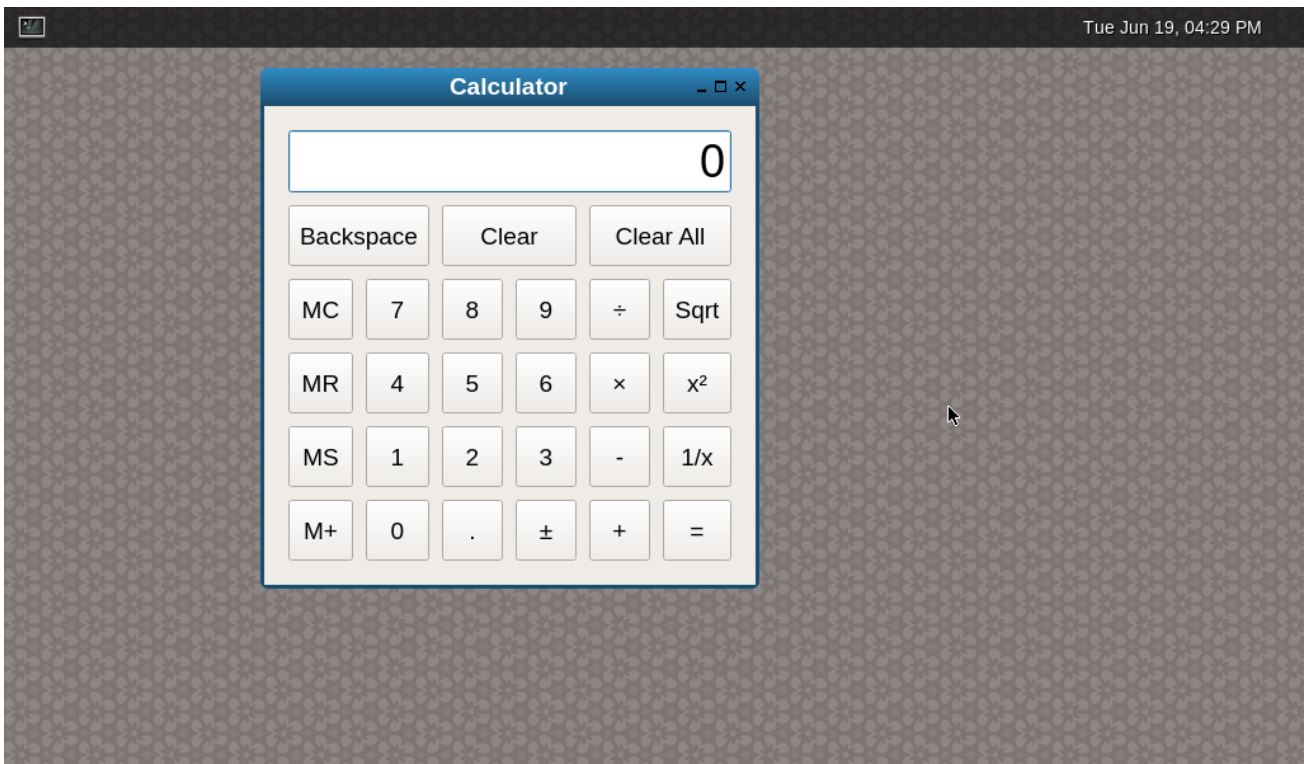
### 4.5.2 Animated Tiles

点击 Animated Tiles，点击右下角的图标选择不同的图标变化。



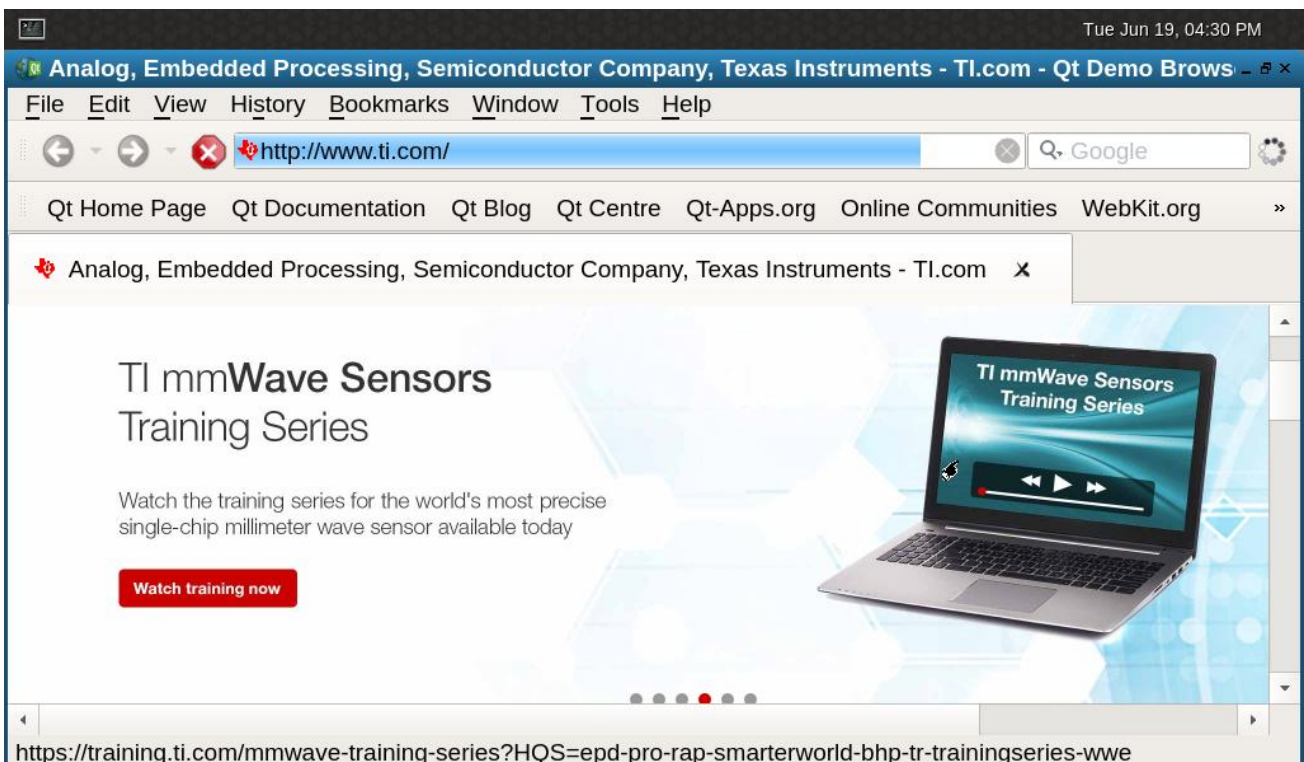
### 4.5.3 Calculator

点击 Calculator，会弹出简易计算器程序



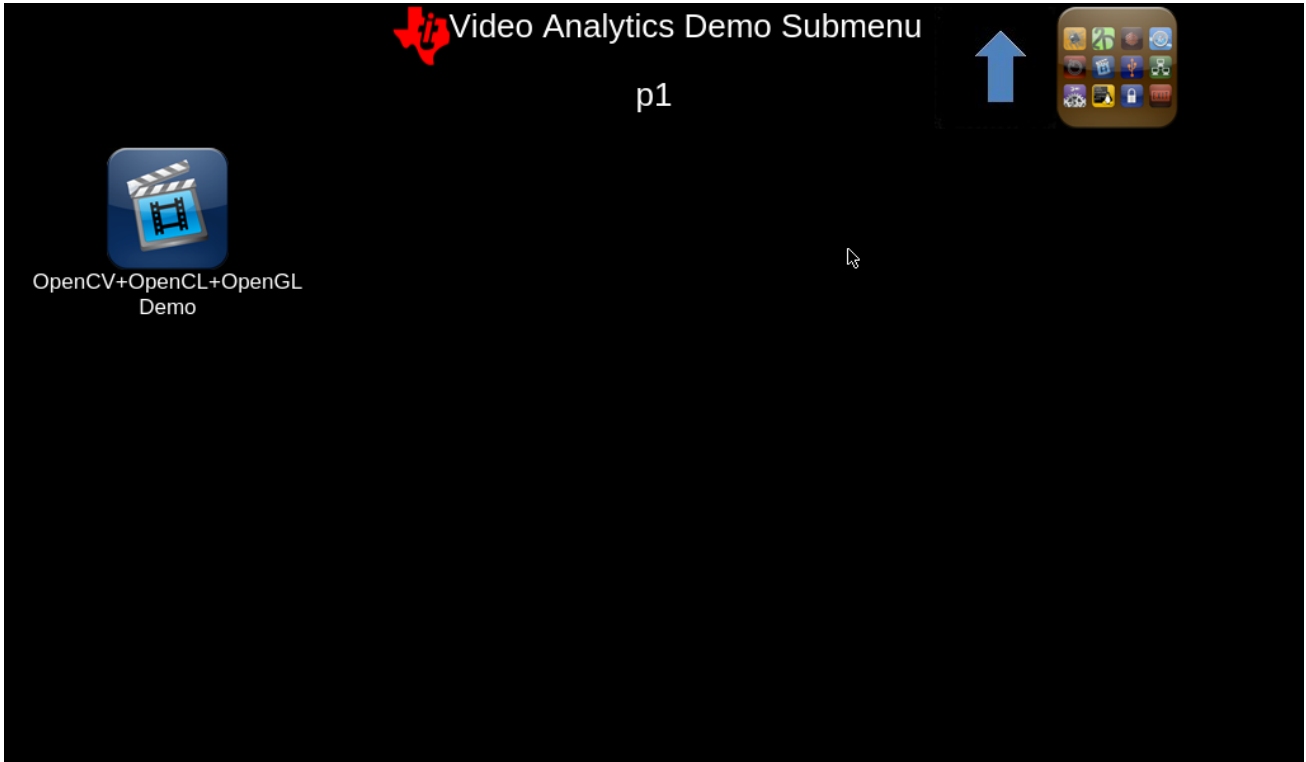
### 4.5.4 Browser

点击 Browser，会打开一个浏览器，默认访问 TI 官网，注意需要插上网线：

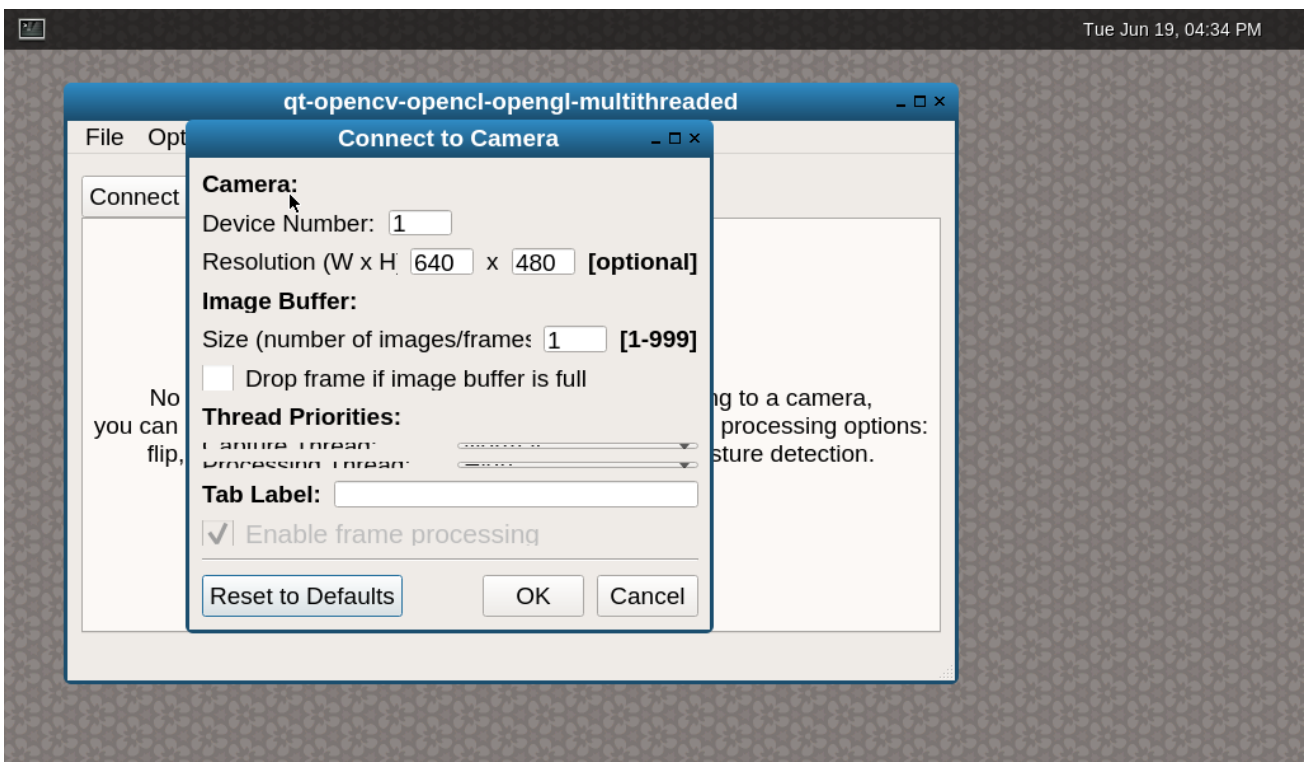


## 4.6 Video Analytics Demo 功能演示

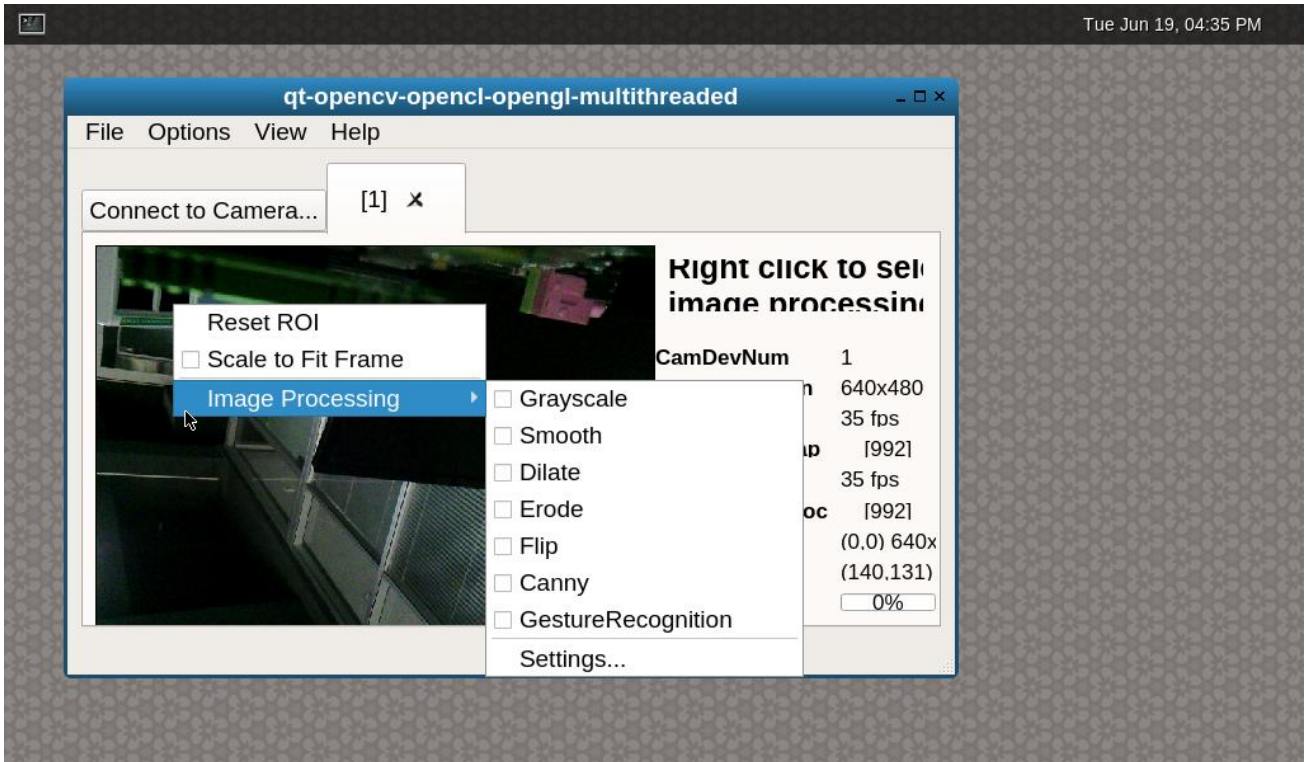
点击 Video Analytics Demo 图标，



点击 OpenCV+OpenCL+OpenGL Demo ，这是一个结合了 Camera、OpenCV、OpenCL 和 OpenGL 的 Demo。点击以后需要先连接到 Camera:

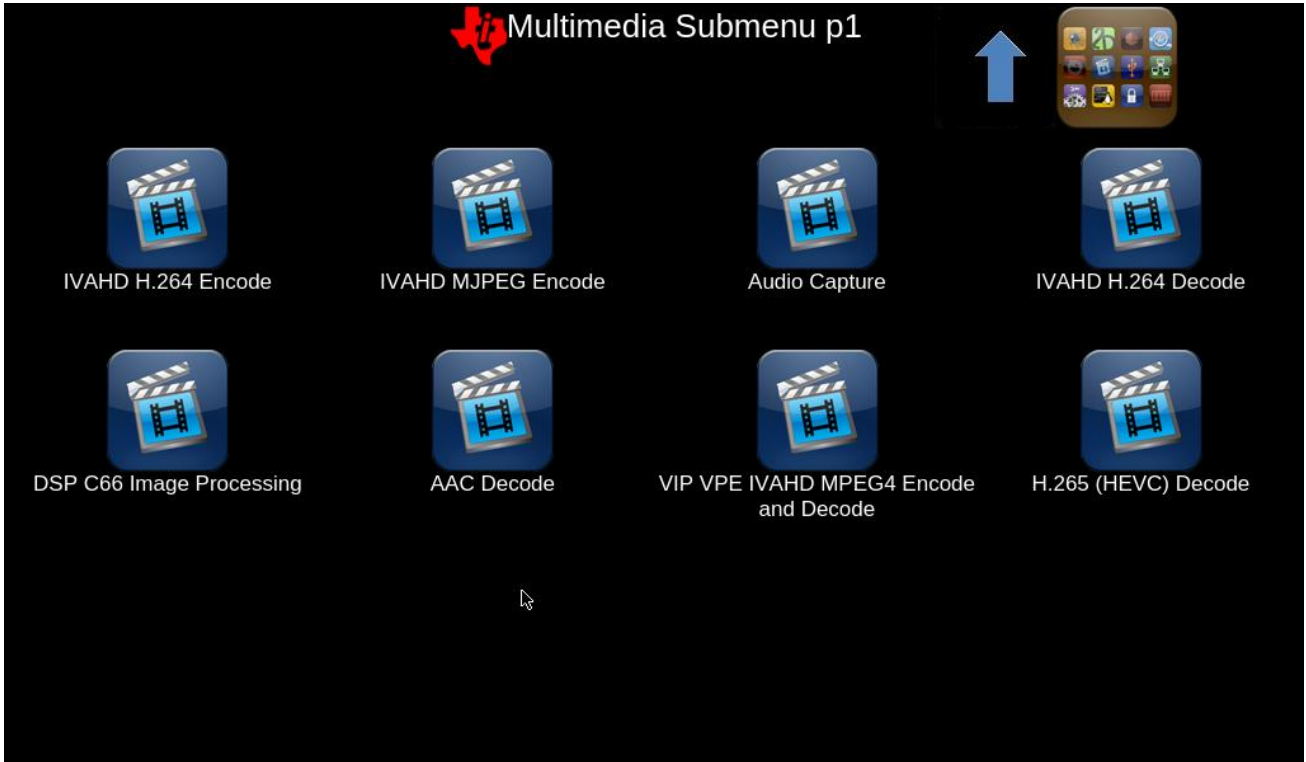


设置好摄像头参数以后连接到摄像头，鼠标在图像上右键点击可以对摄像头的图像进行处理，包括灰度图，图像翻转和手势识别等：



## 4.7 Multimedia 功能演示

点击 Matrix 主界面的 Multimedia 进入以下目录：



Demo Name	说明
IVAHD H264 Decode	这个演示运行一个 gstreamer playbin 管道，使用 IVAHD 解码 H264。演示也播放音频，可以接耳机或扬声器听。
IVAHD H264 Encode	这个演示程序运行一个 gstreamer 管道，在 IVAHD 上执行 H264 编码。输入剪辑为 NV12 格式。输出被保存到/home/root 目录
IVAHD MJPEG Encode	这个演示程序运行一个 gstreamer 管道，在 IVAHD 上执行 MJPEG 编码。输入剪辑为 NV12 格式。输出被保存到/home/root 目录
AAC Decode	演示运行用于 ARM 音频解码和播出的 gstreamer playbin 管道。
H.265 (HEVC) Decode	演示了 ARM 上的 HEVC 解码。gstreamer 管道解码并显示 H265 流。
VIP VPE IVAHD MPEG4 Encode and Decode	演示了视频输入端口（VIP）的视频捕捉，色彩空间转换和视频处理引擎（VPE），IVAHD MPEG4 编码，IVAHD MPEG4 解码和显示的缩放
DSP C66 Image Processing	演示了使用 DSP C66x 插件（dsp66videokernel）将图像处理任务卸载到 DSP。

点击 H.265 (HEVC) Decode, 会播放 H.265 视频流



## 4.8 OpenCL 功能演示

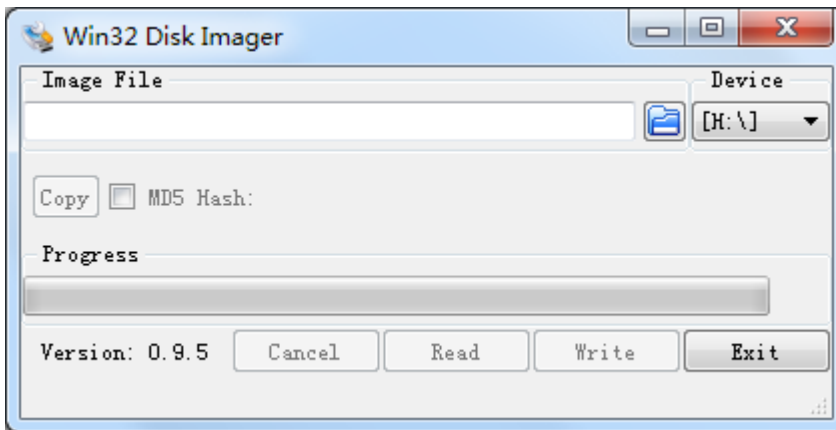
点击 OpenCL 图标, 这个目录下有两个通用计算, 分别为向量和运算、浮点运算。



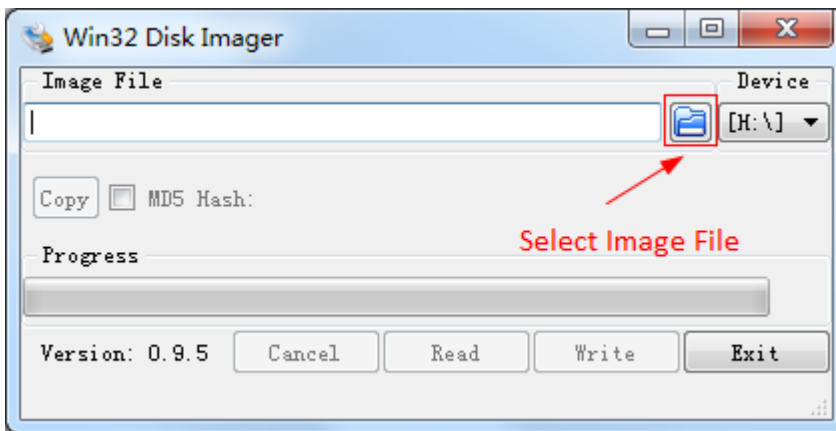
## 第5章 烧写和更新系统镜像

### 5.1 在 Windows 环境下烧写镜像到 SD 卡

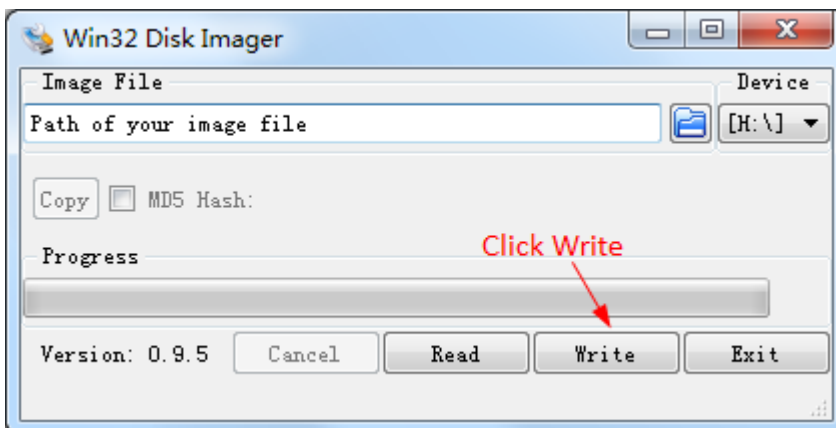
1. 首先，准备一张不小于 4G 的 SD 卡
2. 然后，从 <https://sourceforge.net/projects/win32diskimager/> 下载并安装 Win32 Disk Imager 到 PC



3. 选择需要烧写的镜像，如：EM-TF-EVK-AM5728-TI-ShipmentImage-SDcard-V1.0.4r05.img



4. 点击 Write 烧写镜像：





## 5.2 在 Linux 环境下烧写镜像到 SD 卡

在 Ubuntu 或 Debian 系统中，可以用 bmap-tool 工具烧写镜像到 SD 卡，以镜像 EM-TF-EVK-AM5728-TI-ShipmentImage-SDcard-V1.0.4r05.img 为例：

1. 安装 bmap-tools 工具

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

2. 在命令行中输入下列命令查看 SD 卡编号，本例中为 sdc

```
$ ls /dev/sdc*
```

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

3. 如果 SD 卡被自动挂载，还需要输入下列命令取消挂载

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

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

4. 用下列命令进行烧写

```
$ 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-ShipmentImage-SDcard-V1.0.4r05.img /dev/sdc
```

## 5.3 更新 eMMC 中的镜像

以 eMMC 镜像 EM-TF-EVK-AM5728-TI-ShipmentImage-EMMC-V1.0.4r05.img 为例：将镜像文件拷贝到 U 盘。参考 2.2 节，先从 SD 卡启动系统，再将 U 盘插入 EM-TF-BB-AM57X 的 USB 接口（J15/J16）：

在串口终端输入：

```
root@embest-am57xx:~# ls /dev/sd*
```

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

```
root@embest:~# mount /dev/sdb1 /mnt/
```

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

注意：烧写时间需要 10~25 分钟，请耐心等待...

烧写结束后掉电，参照下表设置拨码开关 S6 和 S7，拔掉 SD 卡，上电启动。

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

## 第6章 附录

### 6.1 硬件

详细硬件介绍请参考：

核心板：《EM-TF-SOM-AM5728 硬件用户手册》

底板：《EM-TF-BB-AM5728 硬件用户手册》

## 第7章 技术支持和保修服务

### 7.1 技术支持

英蓓特科技对所销售的产品提供一年的免费技术支持服务，技术支持服务范围：

- ◆ 提供英蓓特科技嵌入式平台产品的软硬件资源；
- ◆ 帮助用户正确地编译和运行我们提供的源代码；
- ◆ 用户在按照本公司提供的产品文档操作的情况下，如本公司的嵌入式软硬件产品出现异常问题，我们将提供技术支持；
- ◆ 帮助用户判定是否存在产品故障。
- ◆ 以下情况不在我们的免费技术支持服务范围内，但我们将根据情况酌情处理：
  - ◆ 用户自行开发中遇到的软硬件问题；
  - ◆ 用户自行修改嵌入式操作系统遇到的问题；
  - ◆ 用户自己的应用程序遇到的问题；
  - ◆ 用户自行修改本公司提供的软件代码遇到的问题。

### 7.2 保修服务

- ◆ 产品自出售之日起，在正常使用状况下为印刷电路板提供 12 个月的免费保修服务；
  - ◆ 以下情况不属于免费服务范围，英蓓特科技将酌情收取服务费用：
    - ◆ 无法提供产品有效购买凭证、产品识别标签撕毁或无法辨认，涂改标签或标签与实际产品不符；
    - ◆ 未按用户手册操作导致产品损坏的；
    - ◆ 因天灾 (水灾、火灾、地震、雷击、台风等) 或零件之自然耗损或遇不可抗力导致的产品外观及功能损坏；
    - ◆ 因供电、磕碰、房屋漏水、动物、潮湿、杂 / 异物进入板内等原因导致的产品外观及功能损坏；
    - ◆ 用户擅自拆焊零件或修改而导致不良或授权非英蓓特科技认可的人员及机构进行产品的拆装、维修，变更产品出厂规格及配置或扩充非英蓓特科技公司销售或认可的配件及由此引致的产品外观及功能损坏；
    - ◆ 用户自行安装软件、系统或软件设定不当或由电脑病毒等造成的故障；
    - ◆ 非经授权渠道购得此产品者。
    - ◆ 非英蓓特科技对用户做出的超出保修服务范围的承诺（包括口头及书面等）由承诺方负责兑现，英蓓特科技恕不承担任何责任；
  - ◆ 保修期内由用户发到我们公司的运费由用户承担，由我们公司发给用户的运费由我们承担；保修期外的全部运输费用由用户承担。
  - ◆ 若板卡需要维修，请联系技术支持服务部。
- ⚠ 英蓓特科技公司对于未经本公司许可私自寄回的产品不承担任何责任。

## 第8章 联系方式

- ◆ 电话: +86-755-33190846/33190847/33190848
- ◆ 邮箱:
  - ◆ 技术支持: support@embest-tech.com
  - ◆ 销售: chinasales@embest-tech.com
- ◆ 传真: +86-755-25616057
- ◆ 网站: <http://www.embest-tech.cn>
- ◆ 地址: 深圳市南山区留仙大道 4093 号南山云谷创新产业园山水楼 4 楼 B

单击下面可查看定价，库存，交付和生命周期等信息

[>>Avnet manufacturing service\(英蓓特\)](#)