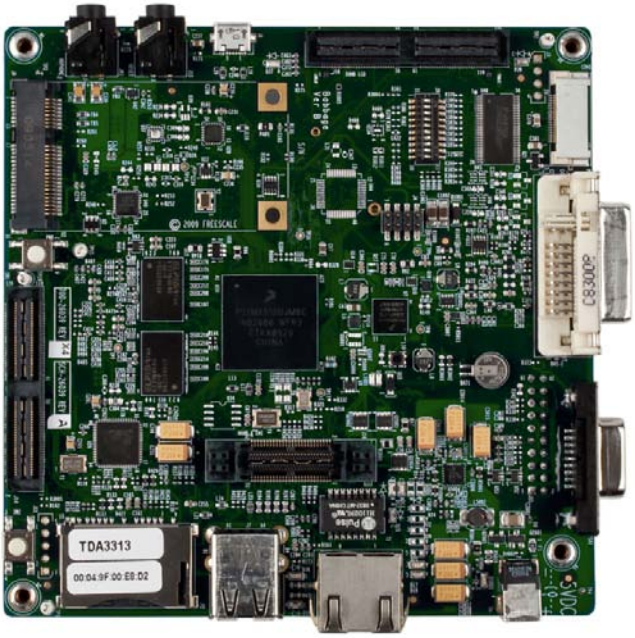
i.MX51 EVK Android

Quick Start Guide



Doc Number: 926-78444
Rev. R7
02/2010

Contents

Chapter 1
About the Board

This chapter provides detailed information about the EVK board and identifies the locations of the main connectors and switches.

1.1 About the EVK Platform System

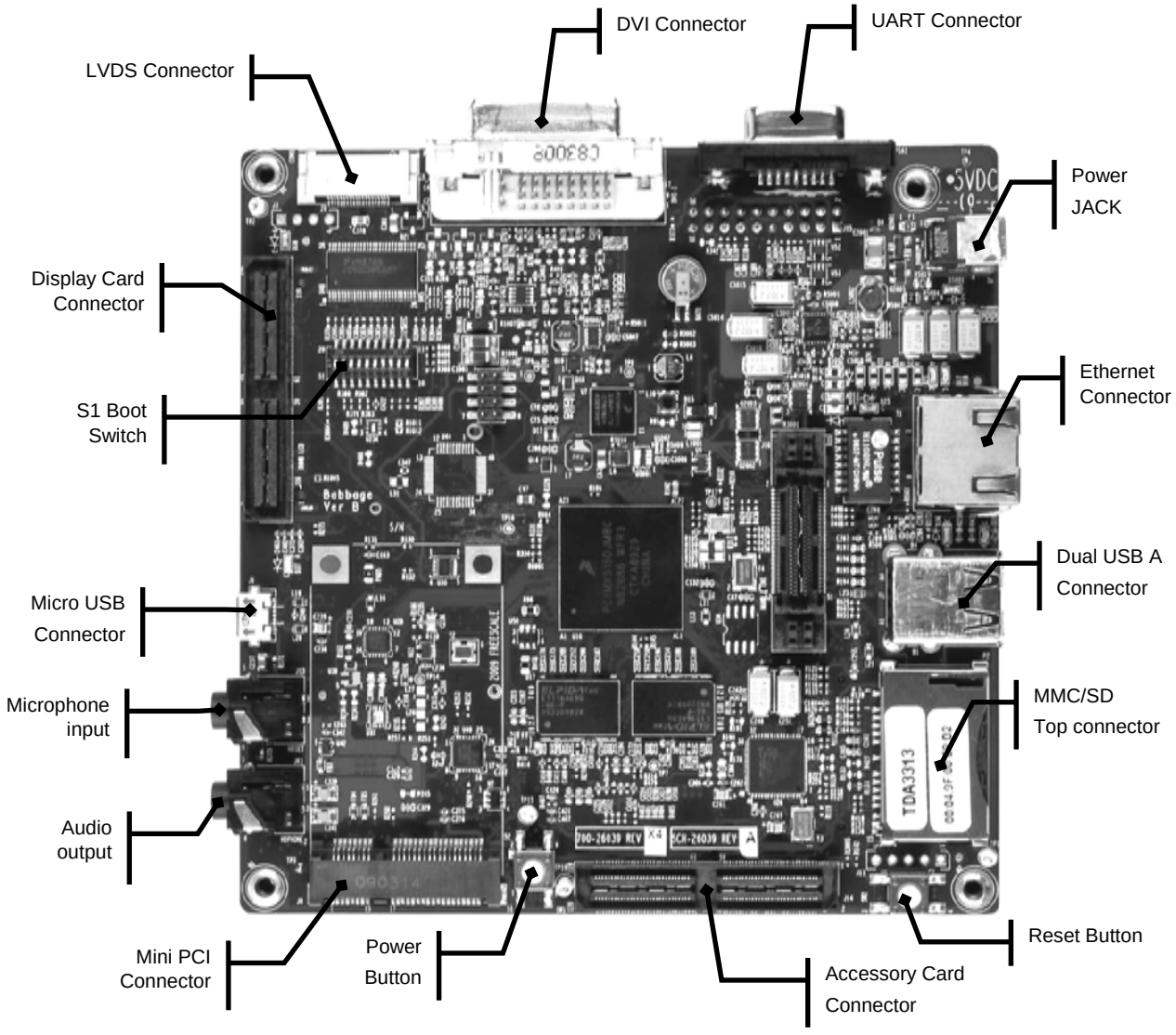
Freescale introduces the i.MX51 EVK, designed for developing multimedia and connectivity applications using the i.MX51 Cortex A8® Applications Processor.

The EVK System decreases the time to market of products providing a near-to-final product design, which can be used as a hardware and software development platform.

Freescale's EVK platform hardware is a single board including CPU, connectors and peripherals; the Board Support Packages (BSP) for the EVK, contain drivers optimized for multimedia operations using the i.MX51.

1.2 EVK Board

Figure 1-1 illustrates the top of the i.MX51 EVK board and figure 1-2 illustrates the bottom side of the EVK.



.

Figure 1-1 Top view of i.MX51 EVK board

The S1 switch, located in the top side of the board, allows the selection of different boot modes. Table 1-1 contains the S1 settings for the different boot options.

Table 1-1 Boot Mode Setting (S1)

Boot Mode Device	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	SW9	SW0
SPI FLASH	Off	Off	On	On	On	Off	On	On	Off	Off
SD/MMC	Off	Off	Off	Off	Off	Off	On	On	Off	Off
BootStrap	On	On	Off	Off	Off	Off	On	On	Off	On

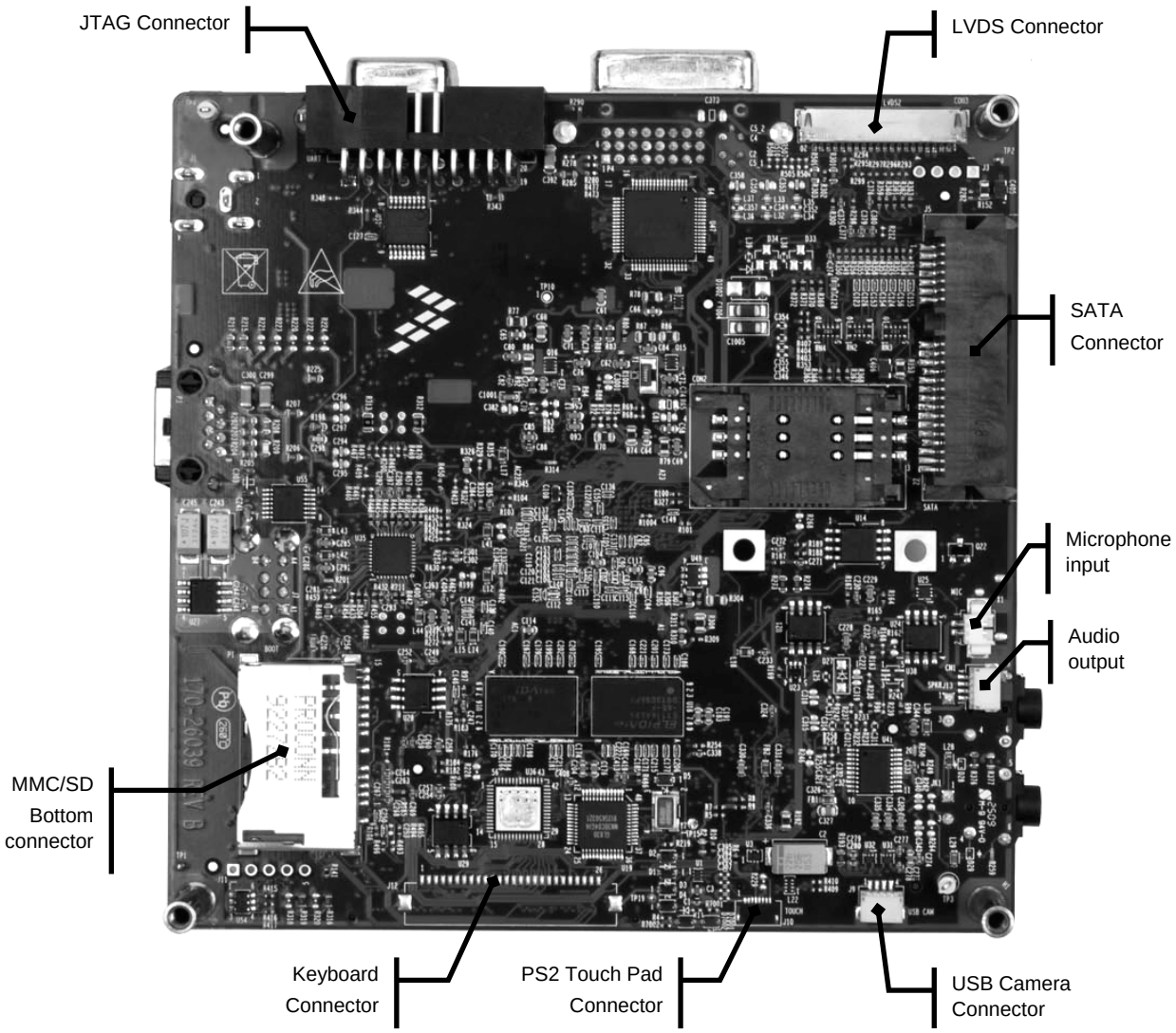


Figure 1-2 Bottom view of i.MX51 EVK board

For details about the hardware implementation, see the i.MX51 EVK Hardware Users Guide, which is included in the documentation package of this release.

Chapter 2
Getting Started

2.1 Unpack the Kit

The EVK contains the items listed in Table 2-1.

Table 2-1 EVK Development Kit Contents

Item	Description
Boards	• i.MX51 EVK Main Board
Cables	• DB9 M/F RS-232 serial cable • USB type A/M to MicroUSB type B/M, shielded cable • Ethernet straight cable
Data storage	• Two 4GB SD cards
Power Supply	• 100/240 VAC – 5 VDC, 3.8A, with AC adaptors
Documentation	• Quick Start Guide (this document) • CD containing links to the i.MX51 EVK Product Summary webpage containing the current software and documentation. • Warranty card • Freescale Support card • End-User License Agreement • Packing list

Verify that all the accessories are contained in the package. Remove the board from the anti-static bags and inspect it for any visible damage.

For additional information and updated content, refer to the Freescale website.

2.2 Development PC Requirements

To develop applications using the i.MX51 EVK development kit, it is needed a PC with the requirements shown in Table 2-3.

Table 2-3 Development PC Requirements

Type	Requirement
Operating System	Linux OS. LTIB was tested in the following are platforms: - Ubuntu 9.04
Network	Internet access
Software Tools	Serial terminal software like Hyperterminal (in Windows) or Minicom (in Linux)
PC Hardware	• 933 MHz Pentium II or later processor; • 2 GHz processor recommended • 512 MByte of RAM; 1 GByte recommended • 1 GByte of available space required on system drive • 10 GByte of available hard-disk space • CD/ DVD ROM drive • 1024x768 or higher resolution display with 256 colors

NOTE

You can use other Linux distributions, however, default configuration (tools, packages, modules, shells, etc.) may be different and it may be need to set the host up for the i.MX51 EVK Linux BSP to work properly.

Ubuntu 9.04 or higher is recommended for Android builds

Linux OS can be installed on a virtual machine so both Windows and Linux can be used on the same host if needed (recommended for testing only).

Chapter 3
Using the Platform

This chapter describes how to assemble a development platform connecting the i.MX51 EVK to the peripherals and host PC. Figure 3-1 shows the minimum connections needed.

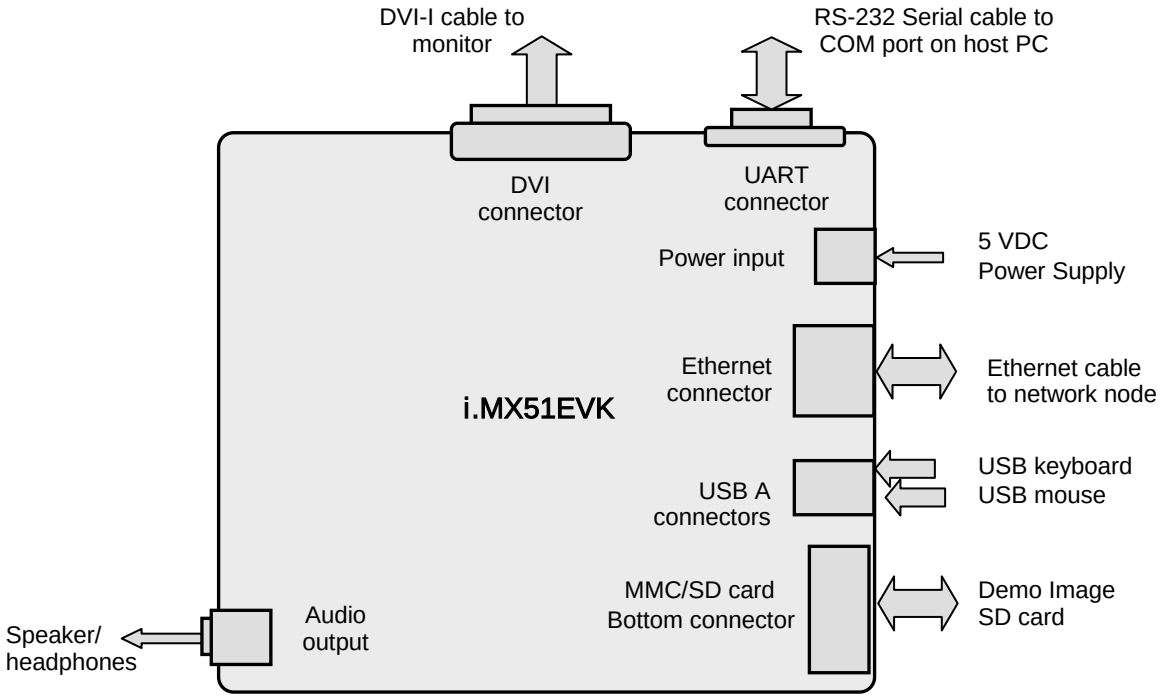


Figure 3-1. i.MX51 EVK connections to peripherals and host PC

3.1 Building a Development Platform

This section explains how to connect the input and output devices to the i.MX51 EVK. Refer to Figures 1-1 and 1-2 to locate the referenced connectors.

1. Plug the image SD card included in the EVK package into the SD card bottom connector
2. Use USB keyboard and mouse as input devices to the EVK, connect those to any USB-A connectors in the board
3. Connect a monitor equipped with a Digital Video Interface input to the EVK board using a DVI-I cable

4. Connect the EVK to the local network using an Ethernet cable

NOTE

Ethernet connectivity is required to perform the default boot process; however, if a network connection is not available you can disable this feature modifying the bootloader configuration as described in the i.MX51 EVK Linux User's Guide.

3.2 Connect Development Platform to PC; Run Preloaded Image

To connect the EVK platform to your host PC, follow the steps below:

1. Plug the power adapter into an electrical outlet and connect the regulated 5V power supply to the power input on the EVK board
2. Configure the boot mode switch for SD card boot as indicated in Table 1-1
3. Connect one end of the RS-232 serial cable included in the kit to the serial port connector on the EVK board; connect the other cable end to a COM port on the host PC
4. Start a serial console application on your host PC. Use the configuration settings shown in Table 3-1

Table 3-1 Serial Console Configuration

Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

5. Keep pressed the power switch (see Figure 1-1) until the EVK board powers-up and then release the switch

Image pre-loaded in the SD card starts execution and status messages appear on the serial console on the host PC as shown in figure 3-2.

To interrupt the boot process press **CTRL-C** at the host PC terminal before the RedBoot prompt is displayed.

Linux operating system logo is shown at the DVI monitor and status messages related to the kernel loading process appear in both the host PC console and the monitor.

If using the WVGA screen the first time that the system boots a calibration application will appear. After that the User Interface from Android appears. From this point the applications can be accessed by using the touch pad or the expansion board (if connected)

If using DVI, the User Interface appears after the booting process finishes. In order to start using the applications is recommended to connect a USB keyboard and an USB mouse. Press F1 key to unlock the system and start the usage of the applications.

It is also possible to navigate inside the Linux root file system and use common Linux commands for interaction with the EVK board from the host PC, however with a vey limited set of commands.

NOTE

The Software release on the SD card is a demo release of Android. For additional details, refer to the Android documentation included in the release.

3.2.1 Flashing the Android Binary to a SD card

In order to flash the SD card with the Android Demo Binary available in the web page use the steps below. You will need the android binary file ANDROIDMX51_R7.bin.gz

In a Linux host

1. Log into Linux and open terminal
2. Insert a 4G SD card (you will need a SD card reader through USB interface that is able to read an write SDHC devices)
3. Check the dev node assigned to the SD card (use the cat /proc/partitions command and look for a device of 4G). In this example it will be used /dev/sdb
4. Run command:
   a. `sudo chmod a+rwx /dev/sdg`
   b. `sudo gunzip -c ANDROIDMX51_R7.bin.gz  > /dev/sdg`
   c. `sudo sync`
5. Insert the SD card into your EVKJ board, make sure SW1 is set for SD card boot.
6. Power on board, you can see the Android loading (as default this demo images requires a DVI connection and the usage of a USB keyboard and a USB mouse)

Chapter 4
Using the Demo Image

4.1 Look and Feel

By default i.MX Android delivers many User Interface applications. Therefore, it is easy to try out the look and feel of Android and investigate interesting features.

4.1.1 Key Map

The key mapping for a Dell USB keyboard we use on EVK/BBG3 board (defined in /system/usr/keylayout/Dell_Dell_USB_Keyboard.kl):

Key	KeyCode	ActAs
ESC	1	BACK
F1	59	MENU
F2	60	SOFT_RIGHT
F3	61	CALL
F4	62	ENDCALL
F6	64	ENDCALL
F8	66	HOME
F9	67	DPAD_CENTER
UP	103	DPAD_UP
DOWN	108	DPAD_DOWN
BACK	14	DEL
ENTER	28	ENTER

Thekey map of a keypad on the EVK (BBG3) add-on peripheral board (defined in /system/usr/keylayout/mxckpd.kl)

POWER	STAR	CALL		DPAD_LEFT		MENU
			DPAD_DOWN	DPAD_CENTER	DPAD_UP	HOME
CAMERA	POUND	ENDCALL		DPAD_RIGTH		BACK

4.1.2 Music Play

Use the following steps to play music

- Insert a SD card with music files (mp3) on FAT partition
- Connect headset/headphone to your board

- Launch the main menu
    - On the EVK board (BBG3), use the UP/DOWN/LEFT/RIGHT keys to highlight the MENU icon at the right side of home screen, press ENTER (or F9)
- Launch the **Music** application by selecting **Music** icon

4.1.3 Video Play

Use the following steps to play a video:

- Insert a SD card with video stream files (.3gp, .mp4) on FAT partition
- Launch main menu
- Launch the **Gallery** application by selecting **Gallery** icon
- During play, touching the screen displays a semi-transient pop-up menu You can select backward/forward/pause/play.

4.1.4 Camera (Still Image Capture)

Use the following steps to capture a still image:

- Insert a formatted SD card into the board
- Connect a headset to the board (this is a workaround for the PIN max issue. See the Known Issues in the release note documentation)
- Launch the main menu
- Launch the **Camera** application by selecting **Camera** icon
- A preview screen appears (as a viewfinder) (takes about 3~4 seconds to see the preview)
- Click the **Shot** icon on the preview screen
- A still image is captured and saved as a JPEG file on the SD card
- Go to Gallery application to see the captured picture.

4.1.5 TVout (Video Only)

- On the EVK Board, the WVGA and TVOUT cannot both be enabled at the same time, and the output resolution of DVI should be limited to 800x600 for 720p TVOut setting. The **hdtv and dmfc=1** boot parameter should be set to enable 720p setting for TVOUT. The valid setting in Uboot is as follows:
    - setenv bootargs_android 'setenv bootargs ${bootargs} init=/init androidboot.console=ttymxc0 video=mxcfb:800x600M-16@60 hdtv dmfc=1'
- Connect the board (expansion board on EVK) to TV using a composite video cable
- In serial console
    - echo U:720x480i-60 > /sys/class/graphics/fb1/mode
      you can change “U:720x480i-60” with below value:
      U:720x576i-50
      U:720x480i-60
      U:240x320p-60
    - setprop rw.TVOUT_VIDEO_DISPLAY 1

- Choose Main Menu -> Gallery, to start to play the video. Now video should be displayed on the TV, but the UI is kept on the LCD

4.1.6 TVout (Video and UI)

Use the following steps for video and UI TVOut:

- On the EVK Board, the WVGA and TVOUT cannot both be enabled at the same time, and the output resolution of DVI should be limited to 800x600 for 720p TVOut setting. The **hdtv and dmfc=3** boot parameter should be set to enable 720p setting for TVOUT. The valid setting in Uboot is as follows:
  - setenv bootargs_android 'setenv bootargs ${bootargs} init=/init androidboot.console=ttymxc0 video=mxcfb:1024x768M-16@60 hdtv dmfc=3'
- Connect the board (expansion board on EVK) to TV using a composite video cable.
- Change the below text entry in the rootfs/init.rc:

```
# Define UI to TVOUT,change this setting to 1 and uncomment below write commands
setprop ro.UI_TVOUT_DISPLAY 0
#write /sys/class/graphics/fb0/blank 1
#write /sys/class/graphics/fb1/blank 1
#write /sys/class/graphics/fb2/blank 1
#write /sys/class/graphics/fb0/fsl_disp_property 1-layer-fb
#write /sys/class/graphics/fb1/mode U:720x480i-60
#write /sys/class/graphics/fb1/blank 0
```

to the text entry below:

```
# Define UI to TVOUT,change this setting to 1 and uncomment below write commands
setprop ro.UI_TVOUT_DISPLAY 1
write /sys/class/graphics/fb0/blank 1
write /sys/class/graphics/fb1/blank 1
write /sys/class/graphics/fb2/blank 1
write /sys/class/graphics/fb0/fsl_disp_property 1-layer-fb
write /sys/class/graphics/fb1/mode U:1280x720p-60
write /sys/class/graphics/fb1/blank 0
```

U:1280x720p-60 is for setting TV in 720P mode. The valid settings for TV mode are as follows:
U:1280x720p-60
U:720x576i-50
U:720x480i-60
U:240x320p-60

4.1.7 Sound Recording

Use the following steps sound recording:

- Insert a formated SD card into the board
- Connect microphone to the board
- Choose Main Menu -> Sound Recorder

- Press the record button to start recording
- Speak into the microphone
- Press the stop button to pause the recording
- Press play to listen what has been recorded
- Select Use this recording. The application exits and a new file (.3gp) is saved into the SD card
- Choose Main Menu > Music. If you can't see the new .3gp file, try below:
    - Choose Main Menu > SD card & phone storage -> Umount SD card
    - Remove the SD card and insert the SD card again
    - Choose Main Menu > Music. You should be able to see the recorded sound file

4.1.8 Camcorder

Use the following steps to record video:

- Insert a formatted SD card into the board
- Connect a microphone to the board
- Choose Main Menu -> Camcorder, to see the preview screen
- Click the record icon, which changes to a stop icon.
- Show something before camera sensor and say something to the microphone
- Click the stop icon.
- Quit the Camcorder application
- Choose Main Menu > Gallery to play the video file that has been recorded

4.1.9 Adjust Media Volume for Music and Video

Use the following steps to adjust the volume:

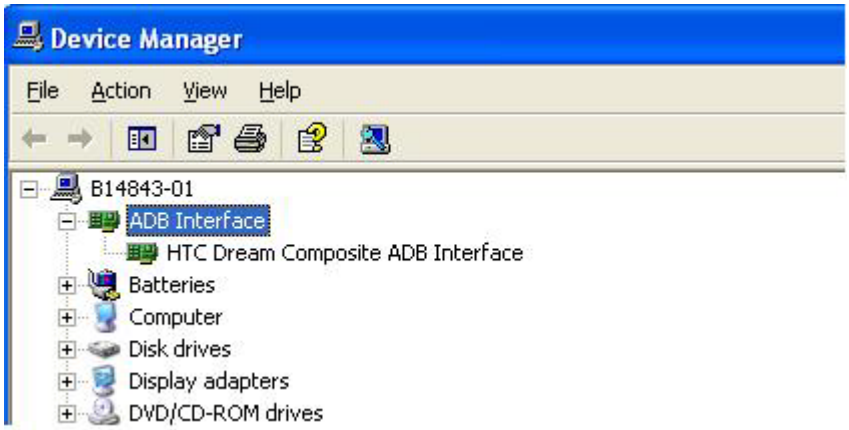
Choose Main Menu -> Setting -> Sound & display -> Media volume

4.1.10 ADB over USB

A PC with ADB installed is required: a windows PC installed with Android SDK package (adb is default under YOUR_SDK_ROOT_DIR/tools/) or a Linux PC on which you had build Android from source (adb is generated at YOUR_ANDROID_REPO_DIR/out/host/linux-x86/bin).

Connect the board to the PC using USB. Connect the USB cable to the OTG port instead of Host port. On the PC, run adb devices to see a list of the attached device (the board). If yes, you can now use any "adb" command.

If using a windows PC, you can see a ADB Interface in Hardware list.



4.1.11 USB Mass Storage

Here the method to enable UMS:

1. Manual enable UMS.
   After connect USB cable to the board, enable UMS by:
   $ sdutil ums enable
   This causes vold to umount local storage and then export it to the PC

Now should be able to access on-board storage (e.g. SD card) from your PC.

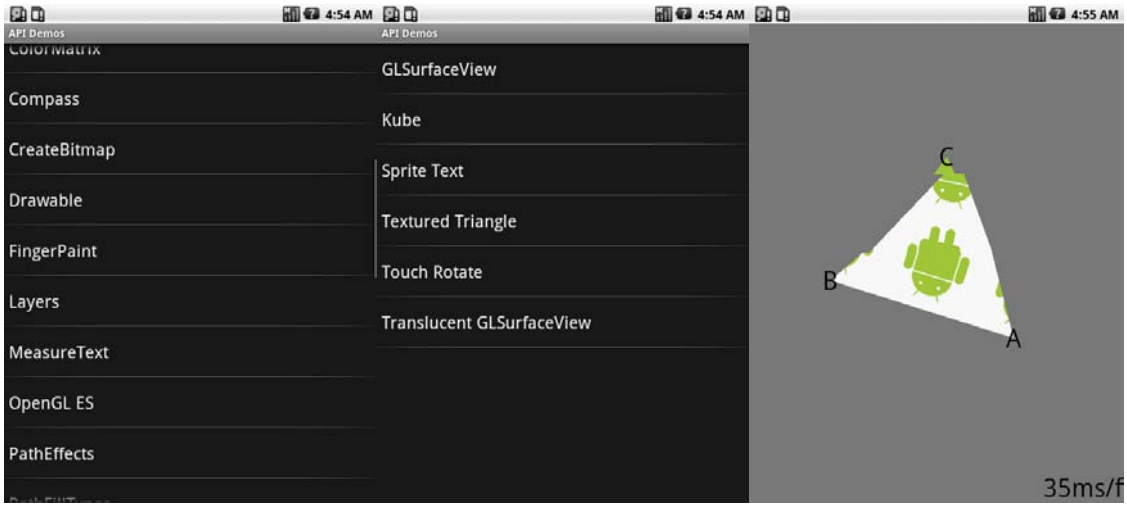
4.1.12 Run 3D application

Android has an API demo application that includes a test function for OpenGL/ES. However by default it is not built. You can build it:

$ cd myandroid
$ make ApiDemos

The generated API demo application is:
out/target/product/generic/data/app/ApiDemos.apk. Copy it into your Android file system (/system/app/) and then reboot the board.

Main menu > ApiDemo > Graphic > OpenGLES > ...



Note: The image distortion in the above 3D screen is caused by screen capture.

To disable hardware 3D (use Android software 3D implementation), use this command:
$ setprop debug.sf.enable_hgl 0
To show FPS(frame per second) in LOGCAT, use this command:
$ setprop debug.sf.showfps 1

4.1.13 Muti-Storage access

There are total three storages supported in i.MX51 BBG.

i.MX51 BBG supports the following two storage types:

- SD card in MMC/SD connector P1
- SD card in MMC/SD connector P2
- UDISK connected to the board through USB port

They are mounted to /sdcard, /extsd and /udisk in Android rootfs specifically.

Since Android SDK only supports the export /sdcard as external storage, the Android Application can only access one external storage. Freescale extends the Android SDK to multi-storage access from Android Application. This extension can be summarized into two categories:

- The SDK API Environment.getExternalStorageDirectory()/getExternalStorageState() returns the external storage directory based on the Storages present on the Board. If there is a SD card in the MMC/SD connector P1, it returns the directory of /sdcard. If there is a UDISK connected to the board through USB port, and no SD card in MMC/SD connector P1, it returns the directory of /udisk. If there is a SD card in the MMC/SD connector P2, and no SD card in MMC/SD connector P1, no UDISK connected with the board through USB port, it returns the directory of /extsd. It dynamically determines which directory to return based on whether the valid device connected to the board based on the sequence /sdcard, /udisk and /extsd. Without modification, the Android Application can access one

external storage at a time. And the media file linked with the URL content://media/external also follow the logical.

- The APIs are added into the Android SDK API to support access the external storages as request:
  Environment.getExternalStorageDirectorySD**()**
  Environment.getExternalStorageDirectoryUDISK**()**
  Environment.getExternalStorageDirectoryEXTSD**()**
  Environment.getExternalStorageStateSD**()**
  Environment.getExternalStorageStateUDISK**()**
  Environment.getExternalStorageStateEXTSD**()**
  And to access media through MediaProvider, Android Application should use the URL content://media/external_sd**,** content://media/external_udisk**, and** content://media/external_extsd for the specific media files in /sdcard, /udisk, and /extsd. Refer to the code change in the Camera Application for the use of the API extension.

  Mountservice broadcast a intent for each external storages state change. This intent had the URL file://sdcard, file://udisk, or file://extsd in extra data of this intent, to indicate which storage changed. Upon receiving those intent, Android Application should check the extra data of this intent, or count on the SDK API Environment.getExternalStorageDirectory()/getExternalStorageState**()** or FSL extended storage API to fetch its interested external storage state and directory.

Chapter 5
Ready to Begin Your Development?

If you are ready to develop new applications using the i.MX51 EVK and Android, use the following documents to locate the information required for your development:

- i_MX_Android_R7_User_Guide.html: Provides the information to build, flash and deploy the android binaries to the EVK board.
- i_MX_Android_R7_Release_Note.html: lists the general information about the android release like the contents, the know issues and supported features.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 010 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

PN 926-78444