USER MANUAL

VAB-800

Linux BSP 1.4

1.2-09192014-144600

Copyright

Trademarks

All brands, product names, company names, trademarks and service marks are the property of their respective holders.

Disclaimer

VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right the make changes to the products described in this manual at any time without prior notice.

Revision History

Version	Date	Remarks
1.0	4/29/2013	Initial external release
1.1	8/14/2014	Modified vab-800_rootfs_patch
1.2	9/17/2014	Added XrandR dual display setting

Table of Contents

Lists of Figures

Lists of Tables

1. Introduction

The purpose of this document is to provide a practical introduction on evaluating Freescale's Ubuntu demo image for the VAB-800/AMOS-800/ARTiGO A800. It also guides the users to make an Ubuntu demo image, and then copy it into Micro SD storage card or eMMC.

The AMOS-800/ARTiGO A800 is Fanless ARM-based embedded systems with VIA VAB-800 Pico-ITX Board. This document is using VAB-800 as an example instead of AMOS-800/ARTiGO A800, since the VAB-800 demo image can be used in AMOS-800/ARTiGO A800 system.

There is Canonical trademark policy when using Ubuntu in commercial usage or redistribution. The VAB-800 evaluation package does not provide Ubuntu DEMO image for evaluation actively.

User should follow Freescale's policy and get the demo image from Freescale official web site when user would like to evaluate Ubuntu on VAB-800.

1.1. Overview

There are three major boot components for Linux, the **"u-boot.bin", "uImage"** and **"Root File System"**. The **"u-boot.bin"** is for initial peripheral hardware parameter. The **"uImage"** is the Linux kernel image, and the **"Root File System"** is for Linux O.S.. The system will not boot successfully into a Linux environment if one of these files does not exist on the boot media (SD storage card or onboard eMMC).

1.2. Package Content

There are two folders in VAB-800 evaluation package.

Figure 1. VAB-800 evaluation package content

1.3. EVK Folder Contents

- **u-boot.bin:** U-boot image.
- **uImage:** Linux kernel image.
- **vab-800_rootfs_patch.tar.bz2:** Including driver modules, configuration files and FlexCAN utility. User can evaluate VAB-800 with Freescale Ubuntu demo root file system.

Note:

If user needs the supporting files for all software mentioned in VAB-800 Linux BSP document, please contact our regional sales representative for assistance.

2. Making Ubuntu Demo Image

2.1. Getting Ubuntu demo image

Step 1

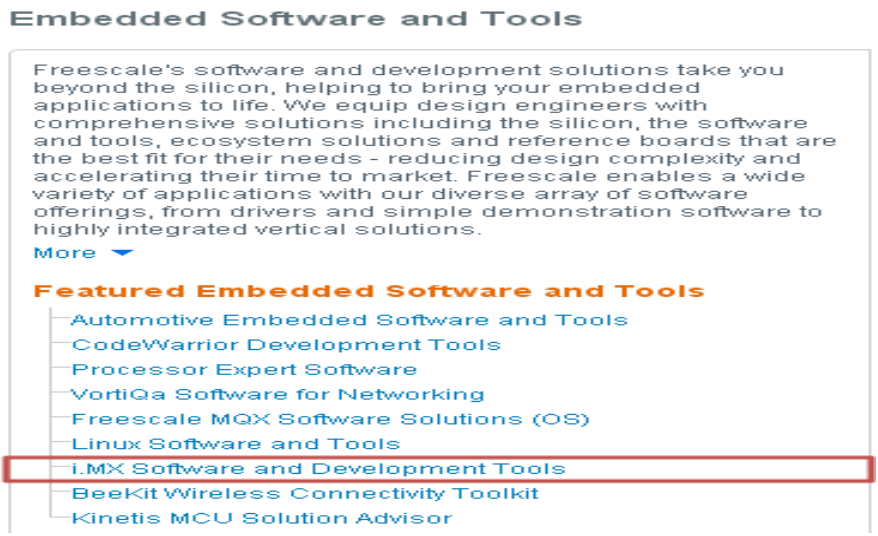
Register your account at **http://www.freescale.com/** to be able to download the files.

Step 2

Go to **"Design Resources"** item --> **Software and Tools**--> then click **"All Software and Tools"**

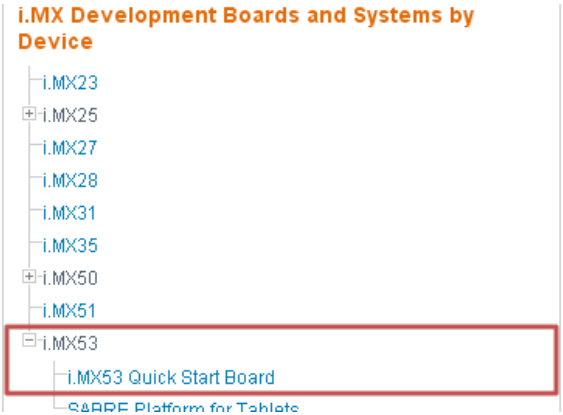
Step 3

Go to **"Featured Embedded Software and Tools"**. Click the **"i.MX Software and Development Tools"**.



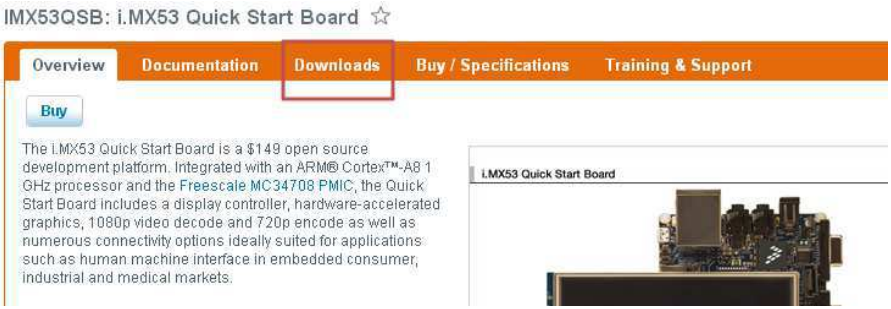
Step 4

Go to **"i.MX Development Boards and Systems by Device"**. Click the **"i.MX53"** plus sign to expand. Then click **"i.MX53 Quick Start Board"**.



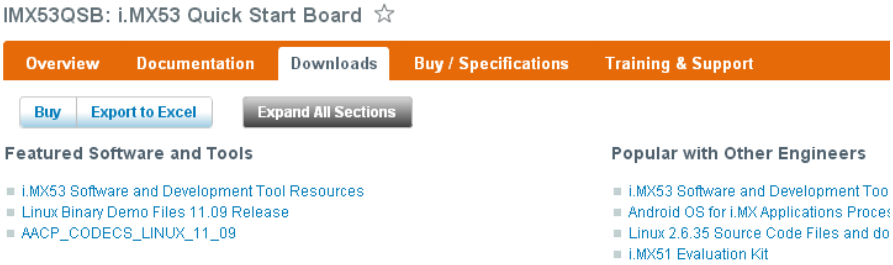
Step 5

Click **"Downloads"**.



There are development tools and pre-built images shown here.

You can select Run-time Software to expand all the items.



Step 6

Download MX53_QSB_UBUNTU_SD_DEMO_IMAGE

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX53QSB&fpsp=1&tab=Design_Tools_Tab#

User can get a zip file **"MCIMX53-START-UBUNTU-11-09-Demo-Image.zip"**.

2.2. Making demo image into Micro SD

Step 1

Prepare a Micro SD storage card (at least 4GB size and Class 4).

Step 2

Refer to the document **"MCIMX53-START-R Flashing Doc.pdf"** after unzipped. Follow the steps to make bootable Micro SD storage card.

2.3. Replace U-boot/Kernel/Modules of VAB-800

User has to replace u-boot, kernel and modules binary files built from VAB-800 BSP on the Micro SD storage card. These files can be found from EVK folder.

Step 1

Getting u-boot, kernel and modules:

User can get u-boot.bin, uImage and vab-800_rootfs_patch.tar.bz2 from EVK folder.

Binary	Description
u-boot.bin	U-Boot bootloader
uImage	Kernel
vab-800_rootfs_patch.tar.bz2	Modify configuration files if user would like to evaluate VAB-800 with Ubuntu (contains **modules.tar.bz2 and CAN_utils.tar.bz2**)

Table 1. u-boot/kernel/modules

Step 2

Copy the u-boot/uImage/driver modules:

2.1. Insert the Micro SD storage card which the OS has made by **"MCIMX53-START-R-BSP-11-09_C.exe"** into your Linux developing PC.

2.2. Umount the Micro SD storage card if Linux OS auto mount your Micro SD card.

The Micro SD card code name for this example is **sdb1**.

```
user@user:~ $ sudo umount /dev/sdb1
```

The path of EVK for this example is /home/user/EVK.

2.3. Copy u-boot **"u-boot.bin"** to Micro SD storage card.

```
user@user:~/EVK$ sudo dd if=./u-boot.bin of=/dev/sdb bs=512 seek=2
skip=2
```

2.4. Copy kernel **"uImage"** to Micro SD storage card.

```
user@user:~/EVK$ sudo dd if=uImage of=/dev/sdb bs=512 seek=2048
```

2.5. Put modification files in root file system.

The configuration files should be modified for VAB-800. User can use files **"vab-800_rootfs_patch.tar.bz2"** in EVK folder.

- Unzip vab-800_rootfs_patch.tar.bz2

```
user@user:~/EVK$ sudo tar jxvf vab-800_rootfs_patch.tar.bz2
```

User will get the followings:

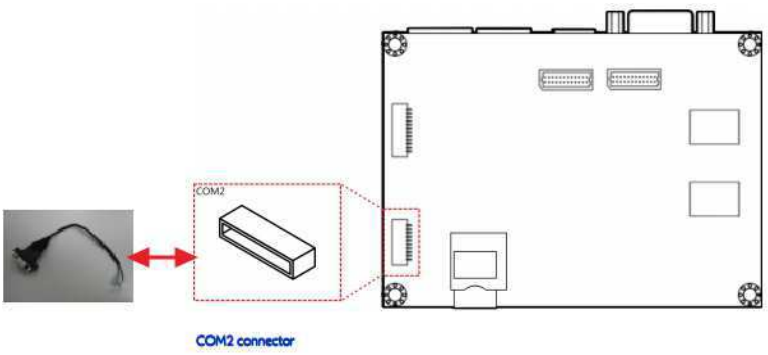
- User runs the script **"vab-800.sh"** to copy modification files automatically.

```
user@user:~/EVK$ sudo ./vab-800.sh /mnt/mountpoint
user@user:~/EVK$ sudo sync && sync
user@user:~/EVK$ sudo umount /mnt/mountpoint
```

Now, unplug Micro SD storage card from your computer.

2.6. Boot from Micro SD storage card (on VAB-800).

- Attach UART cable to VAB-800 COM2, and connect to your computer.



- To modify parameters in u-boot

To setup or modify u-boot parameters in u-boot:

[Display parts]

```
VAB-800 U-Boot > setenv vga 'setenv bootargs console=ttymxc1,115200
video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga'
VAB-800 U-Boot > setenv hdmi 'setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1024x768M@60 hdmi di0_primary'
VAB-800 U-Boot > setenv lvds1 'video=mxcdi0fb:RGB24,480C60 di0_primary
ldb=di0 vga=off'
VAB-800 U-Boot > setenv lvds2 'video=mxcdi1fb:RGB24,480C60 di1_primary
ldb=di1 vga=off'
```

[Storage device parts]

```
VAB-800 U-Boot > setenv bootargs_mmc 'set bootargs ${bootargs}
root=/dev/mmcblk1p1 rw rootwait'
VAB-800 U-Boot > setenv bootargs_emmc 'setenv bootargs ${bootargs}
root=/dev/mmcblk0p2 rootwait rw'
VAB-800 U-Boot > setenv bootcmd_emmc 'run bootargs_base
bootargs_emmc;fatload mmc 1 0x70800000 uImage;bootm'
```

[Boot storage device parts]

Here, user can set the boot device as either Micro SD or eMMC, and should set the below parameter according to the boot selection jumper (J4).

Set boot device as Micro SD:

```
VAB-800 U-Boot > setenv bootcmd 'run bootcmd_mmc; bootm ${loadaddr}'
```

Set boot device as eMMC (optional):

```
VAB-800 U-Boot > setenv bootcmd 'run bootcmd_emmc; bootm ${loadaddr}'
```

[Others]

```
VAB-800 U-Boot > setenv bootargs 'console=ttymxc1,115200 gpu_nommu
setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1920x1080M@60 hdmi di0_primary root=/dev/nfs
ip=dhcp nfsroot=10.29.240.144:/tftpboot/rootfs,v3,tcp'
```

User has to save the parameters when completing the settings.

```
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > boot
```

Finally, user should get the parameter in u-boot after modifying, type command **pri** in u-boot. In this example: the display is VGA and boot from Micro SD card.

Default user account and password are:

Account: lucid

Password: lucid

```
baudrate=115200
loadaddr=0x70800000
netdev=eth0
ethprime=FEC0
uboot=u-boot.bin
kernel=uImage
ethact=FEC0
bootcd=run bootcmd_mmc
bootargs_nfs=setenv bootargs ${bootargs} root=/dev/nfs ip=dhcp
nfsroot=${serverip}:${nfsroot},v3,tcp
bootcmd_net=run bootargs_base bootargs_nfs; tftpboot ${loadaddr}
${kernel}; bootm
bootcmd_mmc=run bootargs_base bootargs_mmc; mmc read ${loadaddr} 0x800
0x1800; bootm
bootargs_mmc=set bootargs ${bootargs} root=/dev/mmcblk1p1 rw rootwait
vga=setenv bootargs console=ttymxc1,115200 video=mxcdi1fb:GBR24,VGA-XGA
di1_primary vga
bootcmd_obds=ext2load mmc 0:1 0x70800000 /unit_tests/obds.bin; go
70800000
nfsroot=/tftpboot/rootfs
lcd=video=mxcdi0fb:RGB24,SEIKO-WVGA di0_primary
lvds=video=mxcdi0fb:RGB666,XGA di0_primary ldb=di0
lvds1=video=mxcdi0fb:RGB24,480C60 di0_primary ldb=di0 vga=off
lvds2=video=mxcdi1fb:RGB24,480C60 di1_primary ldb=di1 vga=off
bootcmd=run bootcmd_mmc; bootm ${loadaddr}
bootfile=uImage
bootargs=console=ttymxc1,115200 gpu_nommu setenv bootargs
console=ttymxc1,115200 video=mxcdi0fb:RGB24,1920x1080M@60 hdmi
di0_primary root=/dev/nfs ip=dhcp
nfsroot=10.29.240.144:/tftpboot/rootfs,v3,tcp
fileaddr=70800000
netmask=255.255.254.0
hdmi=setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1024x768M@60 hdmi di0_primary
serverip=192.168.0.1
ipaddr=192.168.0.2
gatewayip=192.168.0.1
bootargs_base=setenv bootargs console=ttymxc1,115200 gpu_nommu ${vga}
dnsip=192.168.0.1
bootdelay=3
stdin=serial
stdout=serial
stderr=serial
```

2.4. Making demo image to eMMC (optional)

This section will guide user to copy images to eMMC for evaluation. However, it is a must to finish first the section 2.2~2.3 before applying this section.

To make a demo image compression file, follow the steps below:

Step 1

Getting u-boot, and kernel image:

User can directly get the u-boot.bin, and uImage from EVK folder.

Binary	Description
u-boot.bin	U-Boot bootloader Image
uImage	Linux Kernel Image

Table 2. u-boot bootloader and Linux Kernel uImage

Step 2

Making Ubuntu root file system:

2.1. Insert the Micro SD storage card which the OS has made in section A.3 to your Linux developing PC. (Ubuntu 10.04.x x86 at least)

2.2. Compress Micro SD storage card as a compression file.

The Micro SD card code name for this example is **sdb1**, and EVK folder path for this example is /home/user/EVK in your development PC.

```
user@user:~ $ sudo mount /dev/sdb1 /mnt/mountpoint
user@user:~ $ cd /mnt/mountpoint
user@user: /mnt/mountpoint$ sudo tar jcvf ~/EVK/rootfs.tar.bz2 ./*
...
user@user: /mnt/mountpoint$ sudo sync && sync
user@user: /mnt/mountpoint$ cd ~
user@user:~ $ sudo umount /mnt/mountpoint
```

It would take 30~40 minutes to generate a **"rootfs.tar.bz2"**. Here, user could see the u-boot, uImage and rootfs in the EVK folder.

Binary	Description
u-boot.bin	U-Boot bootloader
uImage	Kernel
rootfs.tar.bz2	The Ubuntu demo rootfs made from Micro SD card

Table 3. u-boot/kernel/rootfs

User can put EVK folder to USB pen or Micro SD card. The storage for this example is an USB pen, and assumes the mount point is /media/usbpen.

Step 3

To copy images into eMMC, you must boot first from Micro SD card on VAB-800.

Step 4

To partition eMMC.

The eMMC can be identified when boot into VAB-800 from Micro SD card.

```
lucid@lucid:~/$ ls -l | grep -i mmcblk
...
```

Sometimes the eMMC will auto mount if it is not the first time of using it. Umount the eMMC before partition it. The eMMC code name for this example is identified as **mmcblk0**.

```
lucid@lucid:~/$ sudo umount /dev/mmcblk0*
```

The following steps show how to partition the eMMC.

```
lucid@lucid:~/$ sudo fdisk /dev/mmcblk0
Type the following parameters (each followed by <ENTER>):
u          [switch the unit to sectors instead of cylinders]
d          [repeat this until no partition is reported by the 'p'
command]
n          [create a new partition]
p          [create a primary partition]
1          [the first partition]
4032       [the starting at the offset sector for this example is #4032,
the size is 2MB, which leaves enough space for the boot loader and its
configuration data]
14335      [ending at the offset sectors. The offset sector for this
example is #14335. The size is 5MB, which leaves enough space for
kernel]
n          [create a new partition]
p          [create a primary partition]
2          [the first partition]
14336      [Starting at the offset sectors]
<enter>    [using the default value will create a partition that spans
to the last sector of the medium]
w         [write the partition table]
```

Note:
Users have to create the partitions which leave enough space for the kernel, the boot loader and its configuration data made by users themselves.

Here is an example; a new partition has been created on eMMC. The two partitions of eMMC are for storing the kernel image, and for putting the root file system. You have to apply the new partition table immediately, in order to format it.

```
lucid@lucid:~/$ sudo partprobe
```

Make sure to umount the eMMC before doing the steps below. You can type the command to format the partitions:

```
lucid@lucid:~/$ sudo mkfs.msdos /dev/mmcblk0p1
lucid@lucid:~/$ sudo mkfs.ext3 /dev/mmcblk0p2
```

Step 5

Copy u-boot **"u-boot.bin"** to eMMC.

The image path for this example is /media/usbpen/EVK.

```
lucid@lucid:~/$ sudo mount /dev/sda1 /media/usbpen
lucid@lucid:~/$ cd /media/usbpen/EVK
lucid@lucid:/media/usbpen/EVK$ sudo dd if=./u-boot.bin of=/dev/mmcblk0
bs=512 seek=2 skip=2
lucid@lucid:/media/usbpen/EVK $ sudo sync && sync
```

Step 6

Copy kernel **"uImage"** to eMMC.

The image path for this example is /media/usbpen/EVK.

```
lucid@lucid:~/$ sudo mkdir /mnt/mountpoint
lucid@lucid:~/$ sudo mount /dev/mmcblk0p1 /mnt/mountpoint
lucid@lucid:~/$ cd /media/usbpen/EVK
lucid@lucid:/media/usbpen/EVK$ sudo cp ./uImage /mnt/mountpoint
lucid@lucid:/media/usbpen/EVK$ sudo sync && sync
lucid@lucid:/media/usbpen/EVK$ sudo umount /mnt/mountpoint
```

Step 7

Copy root file system to eMMC.

The compression file path for this example is /media/usbpen/EVK.

```
lucid@lucid:~/$ sudo mount /dev/mmcblk0p2 /mnt/mountpoint
lucid@lucid:~/$ cd /mnt/mountpoint
lucid@lucid:/mnt/mountpoint$ sudo tar jxvf
/media/usbpen/EVK/rootfs.tar.bz2 ./
...
lucid@lucid:/mnt/mountpoint$ sudo sync && sync
lucid@lucid:/mnt/mountpoint$ cd ~
lucid@lucid:~/$ sudo umount /mnt/mountpoint
```

Notes:

1. VAB-800 Linux BSP won't provide Ubuntu root file system for evaluation actively. Users can get an Ubuntu demo image from Freescale official web site and follow Freescale's policy to evaluate. For more details, refer to Appendix A.
2. For details on how to make a compression root file system for evaluation, refer to Appendix A.
3. Via debug port in VAB-800 is COM2. It has to modify the setting from tty1 to ttymxc1 in /etc/inittab if user uses root file system which generated in LTIB.

Step 8

Setup the u-boot parameter at the first time we boot from eMMC. Set the J4 jumper setting (refer to Table 4) to make it boot from eMMC.

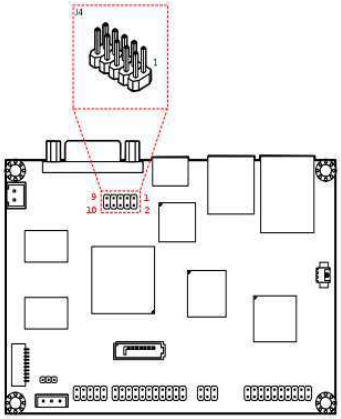


Figure 2. Select eMMC boot at J4

	J4 : Boot Sect				
	1-2	3-4	5-6	7-8	9-10
eMMC	Short	Short	Short	Open	Open
Micro-SD	Open	Short	Open	Open	Open
USB-OTG	Open	Open	Open	Open	Short

Table 4. J4 jumper settings

Connect the VAB-800 and host PC through J14 (COM2) of VAB-800. Run "putty" on host pc to receive the booting message. Power on the VAB-800 and press any key to stop the booting process as shown in Figure 3.

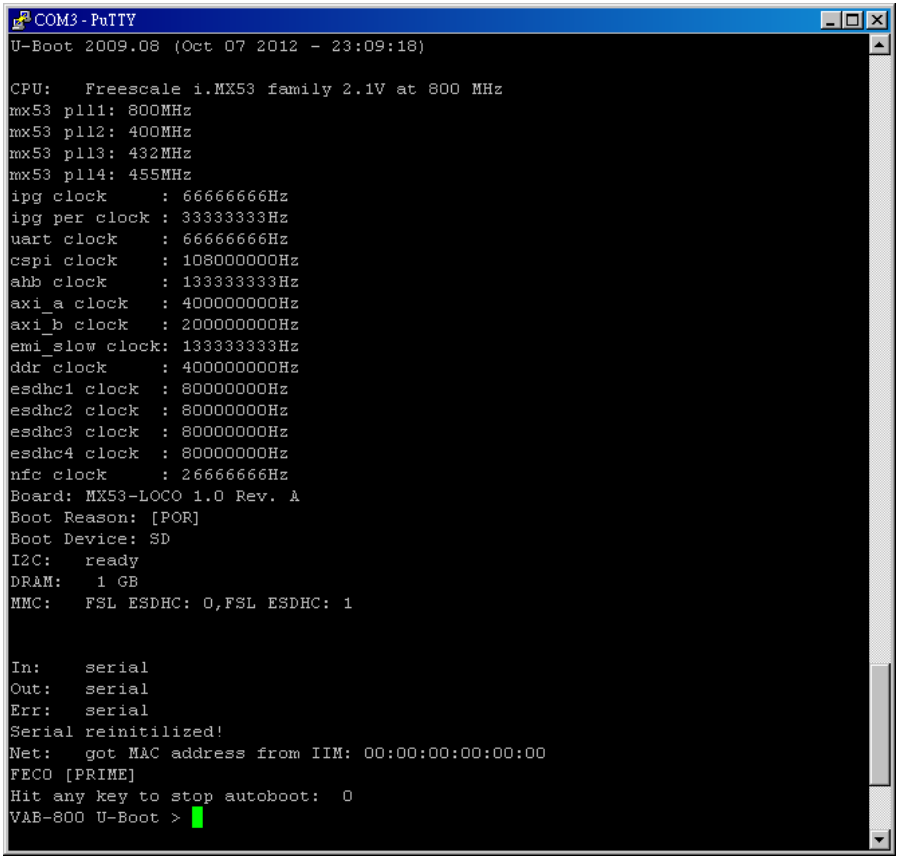


Figure 3. u-boot parameter

To check the parameter in u-boot:

```
VAB-800 U-Boot > pri
bootargs_emmc=setenv bootargs ${bootargs} root=/dev/mmcblk0p2 rootwait
rw
bootcmd_emmc=run bootargs_base bootargs_emmc;fatload mmc 1 0x70800000
uImage;bootm
bootcmd=run bootcmd_emmc; bootm ${loadaddr}
```

To boot from eMMC, you have to set the parameters like the example below. Then the VAB will boot from eMMC.

```
VAB-800 U-Boot > setenv bootcmd 'run bootcmd_emmc; bootm ${loadaddr}'
VAB-800 U-Boot > setenv bootargs_emmc 'setenv bootargs ${bootargs}
root=/dev/mmcblk0p2 rootwait rw'
VAB-800 U-Boot > setenv bootcmd_emmc 'run bootargs_base
bootargs_emmc;fatload mmc 1 0x70800000 uImage;bootm'
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > boot
```

Appendix A. Touch Panel Calibration

Since there are several touch screen drivers in the Freescale Ubuntu DEMO image, the touch panel calibration function might conflict with drivers. Therefore, user should remove unnecessary drivers and install the correct one.

This section describes how to calibrate Touch Panel with VAB-800.

Step 1

Remove the **"evtouch"** driver, which might cause the conflict with xinput or something else. To do this, use the command in the terminal:

```
lucid@lucid:~/$ sudo apt-get remove xserver-xorg-input-evtouch
```

Note:
After removing the evtouch driver, it may look more normal in the xinput device list. To check the touch screen device in xinput list, execute the command "xinput list". Then you will see one touch screen device, mxc_ts, in the list instead of three.

Step 2

Install the **"tslib"** driver.

Check the installation of **"tslib"** driver. The **"tslib"** driver in the current image might show as installed in the apt list, but actually it is not. To ensure the installation, remove the driver and re-install it again. (internet connection is required)

```
lucid@lucid:~/$ sudo apt-get remove xserver-xorg-input-tslib
lucid@lucid:~/$ sudo apt-get install xserver-xorg-input-tslib
```

Step 3

Set up the environment variables.

User may either do this during the calibration or write in the /etc/profile to make it permanent (re-login is required).

```
lucid@lucid:~/$ sudo vi /etc/profile
```

Add parameters into /etc/profile

```
export TSLIB_TSDEVICE=/dev/input/ts0
export TSLIB_CALIBFILE=/etc/pointercal
export TSLIB_CONFFILE=/etc/ts.conf
export TSLIB_CONSOLEDEVICE=/dev/tty
export TSLIB_FBDEVICE=/dev/fb0
```

Step 4

Create **"ts.conf"** file in /etc and write the below content in it.

```
lucid@lucid:~/$ sudo vi /etc/ts.conf
```

Add parameters into /etc/ts.conf

```
module_raw input
module variance delta=30
module dejitter delta=100
module linear
```

Step 5

Modify /usr/lib/X11/xorg.conf.d/05-evdev.conf

Find out the Section below, and replace the driver **"evdev"** by **"tslib"**. (Relogin is required).

```
lucid@lucid:~/$ sudo vi /usr/lib/X11/xorg.conf.d/05-evdev.conf
```

Find below parameter:

```
Section "InputClass"
Identifier "evdev touchscreen catchall"
MatchIsTouchscreen "on"
MatchDevicePath "/dev/input/event*"
Driver "evdev"
EndSection
```

Modify as below:

```
Section "InputClass"
Identifier "evdev touchscreen catchall"
MatchIsTouchscreen "on"
MatchDevicePath "/dev/input/event*"
Driver "tslib"
EndSection
```

Step 6

To calibrate Touch Panel.

Run **"ts_calibrate"** to calibrate the touch screen. (Root required)

```
lucid@lucid:~/$ sudo -i
root@lucid:~/# ts_calibrate
```

Note:

If you write the environment variables in /etc/profile in step 3 and do the calibration by ``sudo", it may show some error message because the environment variables is not being read by root user. Under this circumstances you should switch to root user by using the command ``sudo su -", then make the calibration.

For unknown reason, screen may act abnormally after the calibration. Do reboot, re-login or restart manually and it should work.

Appendix B. Setting U-boot

Step 1

Setting the display devices.

[VGA]

To set VGA as display output.

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${vga}'
```

To set VGA resolution.

```
VAB-800 U-Boot > setenv vga 'setenv bootargs console=ttymxc1,115200
video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga'
```

To replace the resolution, change the red color part with the desired resolution mode.

For example: To set 1680 x 1050 resolution, change VGA-XGA to VGA-WSXGA+.

```
VAB-800 U-Boot > setenv vga 'setenv bootargs console=ttymxc1,115200
video=mxcdi1fb:GBR24,VGA-WSXGA+ di1_primary vga'
```

Note:
There are four modes to be set in VAB-800:
VGA-WSXGA+ : 1680x1050p-60
VGA-SXGA : 1280x1024p-60
VGA-XGA : 1024x768p-60
VGA-SVGA : 800x600p-60

[HDMI]

To set HDMI as display output.

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${hdmi}'
```

To set HDMI resolution.

```
VAB-800 U-Boot > setenv hdmi 'setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1024x768M@60 hdmi di0_primary'
```

To replace the resolution, change the red color part with the desired resolution mode.

For example: To set 1920x1080, change 1024x768 to 1920x1080.

```
VAB-800 U-Boot > setenv hdmi 'setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1920x1080M@60 hdmi di0_primary'
```

[LVDS]

To set LVDS as display output

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${lvds1}'
```

or

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${lvds2}'
```

To set LVDS port

```
VAB-800 U-Boot > setenv lvds1 'video=mxcdi0fb:RGB24,480C60 di0_primary
ldb=di0 vga=off'
VAB-800 U-Boot > setenv lvds2 'video=mxcdi1fb:RGB24,480C60 di1_primary
ldb=di1 vga=off'
```

Note:

VAB-800 supports AUO 7" 800x480 LVDS panel by default.

Step 2

Setting storage devices.

[Micro SD storage card]

```
VAB-800 U-Boot > setenv bootargs_mmc 'set bootargs ${bootargs}
root=/dev/mmcblk1p1 rw rootwait'
VAB-800 U-Boot > setenv bootcmd_mmc 'run bootargs_base bootargs_mmc; mmc
read ${loadaddr} 0x800 0x1800; bootm'
```

[eMMC]

```
VAB-800 U-Boot > setenv bootargs_emmc 'setenv bootargs ${bootargs}
root=/dev/mmcblk0p2 rootwait rw'
VAB-800 U-Boot > setenv bootcmd_emmc 'run bootargs_base
bootargs_emmc;fatload mmc 1 0x70800000 uImage;bootm'
```

Step 3

Set booting storage devices.

Here, user can set the boot device as either Micro SD or eMMC.

Set boot device as Micro SD:

```
VAB-800 U-Boot > setenv bootcmd 'run bootcmd_mmc; bootm ${loadaddr}'
```

Set boot device as eMMC:

```
VAB-800 U-Boot > setenv bootcmd 'run bootcmd_emmc; bootm ${loadaddr}'
```

Step 4

Setting MAC address.

Here, user should set the MAC address in either u-boot or eFuse*

Write into u-boot:

```
VAB-800 U-Boot > setenv ethaddr xx:xx:xx:xx:xx:xx
```

or

Write into eFuse*:.

```
VAB-800 U-Boot > iim blow fecmac xx:xx:xx:xx:xx:xx
```

Note:
1) Generally, the MAC Address will be written into SoC eFuse. User can skip this setting if the MAC address exists.
2) Be careful to write MAC address into eFuse. The reason is that the eFuse can only be written once.

It is a must to reset after you set MAC address and save already:

```
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > reset
```

Appendix C. XRandR Dual Display Setting

Display combination:

- HDMI(up to 1280x720)+VGA
- LVDS1+VGA
- HDMI(up to 1280x720) +LVDS2
- LVDS1+LVDS2

RandR function supports:

- Switch mode on the fly
- Switch single/clone/extend mode on fly
- Rotation

Note:
1) Since the H/W EDID of VGA and HDMI is co-lay on the same I2C bus on VAB800 Rev.A1. HDMI may not be lighted if both HDMI and VGA are connected.
2) Clone mode: the two monitors should have the same resolution. Otherwise, the one with smaller resolution only shows partial content.
3) Extend mode: The desktop content shown on two monitors may partially overlap or missing after mode switch.
4) The total height/width should be less than 2048 when EXA is enabled. It's because max buffer of Z160 is 2048x2048.

Step 1

Connect the two display devices to the target board and boot up.

Step 2

Setup u-boot.

Please set the following parameters in Uboot first according to the display combination.

- HDMI + VGA

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${test}'
VAB-800 U-Boot > setenv test 'console=tty1,38400
video=mxcdi0fb:RGB24,1280x720M@60 di0_primary hdmi
video=mxcdi1fb:GBR24,VGA-XGA vga'
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > reset
```

- LVDS1 + VGA

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${test}'
VAB-800 U-Boot > setenv test 'console=tty1,38400
video=mxcdi0fb:RGB24,480C60 di0_primary ldb=di0
video=mxcdi1fb:GBR24,VGA-XGA vga'
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > reset
```

- HDMI + LVDS2

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${test}'
VAB-800 U-Boot > setenv test 'console=tty1,38400
video=mxcdi0fb:RGB24,1280x720M@60 di0_primary hdmi
video=mxcdi1fb:RGB24,480C60 ldb=di1 vga=off'
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > reset
```

- LVDS1 + LVDS2

```
VAB-800 U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc1,115200 gpu_nommu ${t1} ${t2}'
VAB-800 U-Boot > setenv t1 'console=tty1,38400
video=mxcdi0fb:RGB24,480C60 di0_primary
ldb=separate,di=0,di=1,ch0_map=SPWG,ch1_map=SPWG'
VAB-800 U-Boot > setenv t2 'video=mxcdi1fb:RGB24,480C60 vga=off'
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > reset
```

Step 3

[Change mode]

Open a terminal, and input the command:

```
lucid@lucid:~/$ xrandr –q
```

The combination for this example is HDMI(1280x720) and VGA(1024x768), the mode setup in u-boot.

```
lucid@lucid:~/$ xrandr –q
Screen 0: minimum 640 x 400, current 1280 x 768, maximum 1920 x 1280
DISP3 BG - DI1 connected 1024x768+0+0 (normal left inverted right x axis
y axis) 0mm x 0mm
   builtin        60.0*+
   D:1680x1050p-60   60.0
   D:1280x1024p-60   60.0
   D:1024x768p-60   60.0
   D:1024x600p-60   60.0
   D:800x600p-60   60.3
   D:800x480p-60   67.5
   D:640x480p-60   59.9
DISP3 BG connected 1280x720+0+0 (normal left inverted right x axis y
axis) 641mm x 401mm
   builtin        60.0*+
   1920x1200      60.0 +
   1920x1080P     60.0 +   50.0     24.0
   1600x1200      60.0
   1280x1024      75.0     60.0
   1152x864       75.0
   1280x720P      60.0     50.0
   1024x768       75.0     60.0
   800x600        75.0     60.3
   720x576P_4_3   50.0
   720x576P_16_9   50.0
   720x480P_16_9   59.9
   720x480P_4_3   59.9
   640x480        75.0     60.0     59.9
```

```
lucid@lucid:~/$ xrandr --output [screen0] --mode [select one mode listed
above of screen0]
```

Example (HDMI):

```
lucid@lucid:~/$ xrandr --output "DISP3 BG" --mode 1280x720P
```

Note:

[screen0]: "DISP3 BG"
[screen1]: "DISP3 BG - DI1"

[Switch Single/Clone/Extend mode]

<Clone mode>

```
lucid@lucid:~/$ xrandr –q
Screen 0: minimum 640 x 400, current 1280 x 768, maximum 1920 x 1280
DISP3 BG - DI1 connected 1024x768+0+0 (normal left inverted right x axis
y axis) 0mm x 0mm
   builtin        60.0*+
   D:1680x1050p-60   60.0
   D:1280x1024p-60   60.0
   D:1024x768p-60   60.0
   D:1024x600p-60   60.0
   D:800x600p-60   60.3
   D:800x480p-60   67.5
   D:640x480p-60   59.9
DISP3 BG connected 1280x720+0+0 (normal left inverted right x axis y
axis) 641mm x 401mm
   builtin        60.0*+
   1920x1200      60.0 +
   1920x1080P     60.0 +   50.0     24.0
   1600x1200      60.0
   1280x1024      75.0     60.0
   1152x864       75.0
   1280x720P      60.0     50.0
   1024x768       75.0     60.0
   800x600        75.0     60.3
   720x576P_4_3   50.0
   720x576P_16_9   50.0
   720x480P_16_9   59.9
   720x480P_4_3   59.9
   640x480        75.0     60.0     59.9
```

```
lucid@lucid:~/$ xrandr --output [screen0] --mode [select one mode listed
of screen0] --output[screen1] --mode [select one mode listed of screen1]
```

Example:

```
lucid@lucid:~/$ xrandr --output "DISP3 BG" --mode 1280x720P --output
"DISP3 BG - DI1" --mode D:640x480p-60
```

<Extend mode>

```
lucid@lucid:~/$ xrandr --output [screen0] --[left-of, right-of, above,
below] [screen1]
```

Example:

```
lucid@lucid:~/$ xrandr --output "DISP3 BG" –left-of "DISP3 BG - DI1"
```

<Single mode>

To disable one of the dual display.

```
lucid@lucid:~/$ xrandr --output [screen0/screen1] --off
```

Example:

```
lucid@lucid:~/$ xrandr --output "DISP3 BG" --off
```

Note:

1) Please use command if user would like to set clone mode from extend mode.
xrandr --output [screen0] --same-as [screen1]
2) Please use command under debug port (COM2) if user disables display under single mode.
$export DISPLAY=:0.0
$xrandr --output [screen0] --auto
Or
$xrandr --output [screen0] --mode [mode list]

[Rotation]

```
lucid@lucid:~/$ xrandr --output [screen0] --rotate [left, right,
inverted, normal]
```

Taiwan Headquarters
1F, 531 Zhong-Zheng Road
Xindian, Taipei, 23148
Taiwan

TEL: 886.2.2218.5452
FAX: 886.2.2218.5453
Email: embedded@via.com.tw

USA
940 Mission Court
Fremont, CA 94539
USA

TEL: 1.510.683.3300
FAX: 1.510.687.4654
Email: embedded@viatech.com

Europe
In den Dauen 6
53117 Bonn
Germany

TEL: 49.228.688565.0
FAX: 49.228.688565.19
Email: embedded@via-tech.eu

China
Tsinghua Science Park Bldg. 7
No. 1 Zongguancun East Road
Haiden District, Beijing, 100084
China

TEL: 86.10.59852288
FAX: 86.10.59852299
Email: embedded@viatech.com.cn

Japan
3-15-7 Ebisu MT Bldg. 6F
Higashi, Shibuya-ku
Tokyo 150-0011
Japan

TEL: 81.3.5466.1637
FAX: 81.3.5466.1638
Email: embedded@viatech.co.jp

Korea
2F, Sangjin Bldg., 417
Dogok Dong, Gangnam-Gu
Seoul 135-854
South Korea

TEL: 82.2.571.2986
FAX: 82.2.571.2987
Email: embedded@via-korea.com