DIMM-MX6, EMCON-MX6 Developer Kit for Android 4.4.3

User Manual

Rev3 / 19.06.2015

emtrion GmbH

Revision: **3 / 19.06.2015**

Rev	Date/Signature	Changes
1	18.11.2013/Ha	--
2	23.04.2015/Rr	Changes for Kitkat 4.4.3
3	19.06.2015/Rr	Support for emcon_mx6 added

This document is published by:

emtrion GmbH
Alter Schlachthof 45
D-76131 Karlsruhe
Germany

Tel.: +49 (0) 721 / 62725 - 0
Fax.: +49 (0) 721 / 62725 - 19
E-mail: mail@emtrion.de

emtrion
embedded systems

Internet: www.emtrion.de

1 Contents

1 Contents ... 3
2 Definitions ... 4
3 Introduction ... 5
4 The Bootloader ... 6
4.1 Communication settings ... 6
4.2 Dip switch setup for DIMM-MX6 ... 6
4.3 Bootloader prompt ... 7
4.3.1 Print/Change environment variables ... 7
4.3.2 Network setup ... 8
4.4 Updating Android Images ... 8
5 Android Application quick start Guide ... 14
5.1 Preparation ... 14
5.2 My Application: an Hello World example ... 15
6 Running Android via NFS ... 21
6.1 Setting up the bootloader: ... 21
6.2 Troubleshouting ... 23
7 Android for advanced user ... 24
7.1 Modifying the Android File System for DIMM-MX6 ... 24
7.2 Modifying the Android File System for EMCON-MX6 ... 24
7.3 Rebuilding the package for DIMM_MX6 ... 25
7.4 Rebuilding the package for EMCON_MX6 ... 26
7.5 Backup ... 26

2 Definitions

The table below lists some definitions of terms in this manual.

DIMM-MX6	The target platform
DIMM-MX6-PCIe	Target platform with PCIe support (no HDMI)
EMCON-MX6	Target platform with emcon interface
AOSP	Android Open Source Project
VM	Virtual Machine
IDE	integrated development environment
NFS	Network File System
OS	Operating System

3 Introduction

Emtrion has designed this Android starter-kit to help you design your Android application quickly and efficiently, and evaluate the Hardware.

By simply running the virtual machine provided, you can start developing your application using Android Studio. The SDK is already installed. No download or installation is required to start.

The version of Android OS running on the starter-kit is a custom Android version made by Emtrion. It is compliant with the vanilla Android 4.4.3 named "Kitkat" but with some restrictions on specific points. You can find those points below on the section "Available devices/interfaces".

The process of making or rebuilding the Android OS is substantial and requires a lot of knowledge regarding Linux and Android. If you have any requests or particular needs, Emtrion can build a custom Android OS for you. Please contact sales@emtrion.de for any questions regarding this point.

4 The Bootloader

This section gives a brief description of the bootloader used in this Developer Kit. When you are more interested in the function scope of the bootloader, please refer to the detailed description of the bootloader on the homepage of the U-Boot project: http://www.denx.de/wiki/U-Boot/

4.1 Communication settings

The bootloader's communication settings are:

Baudrate	115200 bps
Data bits	8
Stop bits	1
Parity	none
Handshake	none

4.2 Dip switch setup for DIMM-MX6

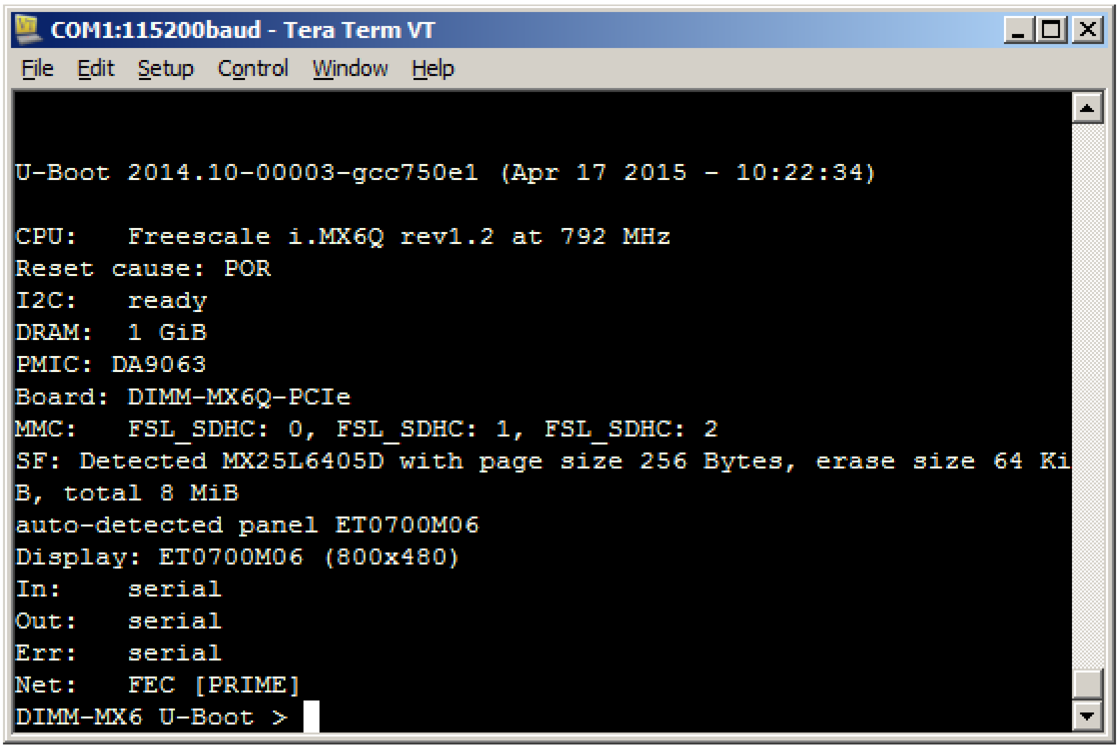
The DIMM-MX6 module carries two dip switches, which have to be setup as follows for a successful start up of the bootloader.

DIP Switch setting for successful start up:

	2	1
off		X
on	X	

4.3 Bootloader prompt

The bootloader prompt is reached if you press a key in the console window when the boot delay is counted down. The bootloader prompt allows you to change settings of the bootloader, to update the Android image in flash or boot via NFS.



We are using mainline U-Boot. A detailed description of the bootloader can be found on the homepage of the U-Boot project: http://www.denx.de/wiki/U-Boot/.

4.3.1 Print/Change environment variables

The environment variables are handled by using 3 commands: *printenv, setenv, saveenv. Printenv* shows you the current setting of all environment variables. *Setenv <variable> <value>* changes the value of an environment variable. This change is only in RAM and will be lost after reset. The changes can be made permanent by using *saveenv.* The following example shows how the boot command is set up.

```
DIMM-MX6 U-Boot > setenv bootcmd 'run flash_boot'
DIMM-MX6 U-Boot > saveenv
```

For the emcon-mx6 module it's similar.

```
EMCON-MX6 U-Boot > setenv bootcmd 'run flash_boot'
EMCON-MX6 U-Boot > saveenv
```

4.3.2 Network setup

The network setup of the bootloader is also handled by environment variables:

autoload	Set this to "no". This prevents that the use of the "dhcp" command automatically starts a tftp download
ipaddr	IP address of the device. Only effective if dhcp is deactivated
serverip	IP address of the host PC which acts as TFTP server
netmask	Subnet mask of the device
ip-method	Set this to "static" or "dhcp" according to your setup. This is used by the update_uboot script

If you have a DHCP server in your network and want to configure the device via dhcp simple use the command *"dhcp":*

```
DIMM-MX6 U-Boot > dhcp
BOOTP broadcast 1
DHCP client bound to address 172.26.1.10 (6 ms)
```

If there is no DHCP server you have to set the variables *"ipaddr"* and *"netmask"* by hand.

To test your network setting you can ping the host PC from the device running the bootloader. To do so use the command *ping <ip address>*. **Please note, that the device running the bootloader cannot be pinged**.

For emcon-mx6 it's the same procedure.

4.4 Updating Android Images

On the Linux System provided as a Virtual Machine, there is an exported NFS share. For example, for the DIMM-MX6 system:

```
/home/hico/share/dimm-mx6q/
```

With the sub-directory:

```
/home/hico/share/dimm-mx6q/images
```

and

```
/home/hico/share/dimm-mx6q/root/rootfs/boot
```

In the sub-directory „images“, the following files must appear:

```
zImage
```

```
android-datafs-dimm_mx6.tar.bz2
android-rootfs-dimm_mx6.tar.bz2
```

In the sub-directory „root/rootfs/boot“, the following files must appear too:

```
emPURS_plat
imx6q-dimm_mx6.dtb
initramfs-dimm-mx6.igz
uboot_script
zImage
```

For emcon-mx6 it's similar the string dimm has to be replaced by the string emcon

```
/home/hico/share/emcon-mx6q/
```

With the sub-directory:

```
/home/hico/share/emcon-mx6q/images
```

and

```
/home/hico/share/emcon-mx6q/root/rootfs/boot
```

In the sub-directory „images“, the following files must appear:

```
zImage
android-datafs-emcon_mx6.tar.bz2
android-rootfs-emcon_mx6.tar.bz2
```

In the sub-directory „root/rootfs/boot“, the following files must appear too:

```
emPURS_plat
imx6q-emcon_mx6.dtb
initramfs-emcon-mx6.igz
uboot_script
zImage
```

Be sure the path of those directories is exported by verifying the file /etc/exports. It must be like this:

```
/home/hico/share
0.0.0.0/0.0.0.0(rw,all_squash,anonuid=1000,anongid=1000,no_subtree_ch
eck,sync)
```

You can restart the NFS share by typing on a terminal:

```
~$sudo service nfs-kernel-server restart
```

This is mandatory when you make a change in the file /etc/exports.

First of all, your target must have the correct settings, i.e the IP address of the Linux System in the Virtual Machine and the path of the NFS share. See the example in the log below in blue or check the section 6.1.

Then you will be able to update the Android system by only typing:

```
run restore_sys
```

This will flash the Android image automatically and independently. This process takes few minutes to complete and the board must not be powered off.

You can find below, an example of what the board output on the serial during this process.

Note: the highlighted commands in green must be entered to change the settings. With "saveenv", the entries are stored permanently in the boot loader. The procedure is shown for dim-mx6 but it is the same for emcon-mx6. It might differ a bit from the actual version.

```
U-Boot 2014.10-00003-gcc750e1 (Apr 17 2015 - 10:22:34)

CPU:   Freescale i.MX6Q rev1.2 at 792 MHz
Reset cause: POR
I2C:   ready
DRAM:  1 GiB
PMIC: DA9063
Board: DIMM-MX6Q-PCIe
MMC:   FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
SF: Detected MX25L6405D with page size 256 Bytes, erase size 64 KiB, total 8 MiB
auto-detected panel ET0700M06
Display: ET0700M06 (800x480)
In:    serial
Out:   serial
Err:   serial
Net:   FEC [PRIME]
Hit any key to stop autoboot:  0
DIMM-MX6 U-Boot >
DIMM-MX6 U-Boot > setenv nfsroot /home/hico/share/dimm-mx6q/root/rootfs
DIMM-MX6 U-Boot > setenv serverip 172.26.1.3
DIMM-MX6 U-Boot > saveenv
Saving Environment to SPI Flash...
SF: Detected MX25L6405D with page size 256 Bytes, erase size 64 KiB, total 8 MiB
Erasing SPI flash...Writing to SPI flash...done
```

```
DIMM-MX6 U-Boot > run restore_sys
BOOTP broadcast 1
DHCP client bound to address 172.26.1.10 (5 ms)
Using FEC device
File transfer via NFS from server 172.26.1.21; our IP address is 172.26.1.10
Filename '/home/hico/share/dimm-mx6q/root/rootfs/boot/uboot_script'.
Load address: 0x10800000
Loading: #
done
Bytes transferred = 2810 (afa hex)
Booting remote rescue ...
Using FEC device
File transfer via NFS from server 172.26.1.21; our IP address is 172.26.1.10
Filename '/home/hico/share/dimm-mx6q/root/rootfs/boot/imx6q-dimm_mx6.dtb'.
Load address: 0x18000000
Loading: ##########
done
Bytes transferred = 49133 (bfed hex)
Using FEC device
File transfer via NFS from server 172.26.1.21; our IP address is 172.26.1.10
Filename '/home/hico/share/dimm-mx6q/root/rootfs/boot/initramfs-dimm_mx6.igz'.
Load address: 0x14000000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #######
done
Bytes transferred = 3028070 (2e3466 hex)
Using FEC device
File transfer via NFS from server 172.26.1.21; our IP address is 172.26.1.10
Filename '/home/hico/share/dimm-mx6q/root/rootfs/boot/zImage'.
Load address: 0x10800000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         ####################################
done
Bytes transferred = 5508408 (540d38 hex)
BOOTP broadcast 1
DHCP client bound to address 172.26.1.10 (5 ms)
Kernel image @ 0x10800000 [ 0x000000 - 0x540d38 ]
## Loading init Ramdisk from Legacy Image at 14000000 ...
   Image Name:   core-image-purs-dimm-mx6-2012110
   Image Type:   ARM Linux RAMDisk Image (gzip compressed)
   Data Size:    3028006 Bytes = 2.9 MiB
   Load Address: 11000000
   Entry Point:  11000000
```

```
   Verifying Checksum ... OK
## Flattened Device Tree blob at 18000000
   Booting using the fdt blob at 0x18000000
   Using Device Tree in place at 18000000, end 1800efec

Starting kernel ...

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+
+ Log from Linux Kernel Starting…
+
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Executing init script S60 for passing kernel command line
cmdline= console=ttymxc1,115200n8 serverip=172.26.1.21 empurs_cmd=production
boot_mode=nfs boot_dir=/home/hico/share/dimm-mx6q/root/rootfs ip=dhcp
Parsing result SERVERIP=172.26.1.21  EMPURS_CMD=production  BOOT_MODE=nfs
BOOT_DIR=/home/hico/share/dimm-mx6q/root/rootfs
########################################################################################
###############
##########################         script /usr/sbin/emPURS is executed
#########################################
##########################
#########################################
########   USAGE:    The script requires four parameters.
########
########
########
########   1st parameter:     command(etc. production, update_kernel, update_rfs)
########
########   2nd parameter:     ipaddress of the server
########
########   3rd parameter:     boot mode nfs or tftp
########
########   4th parameter:     boot directory:
########
########                      tftp: path of the tftp subdirectory with the closing slash
########
########                            for example part1/part2/part3/
########
########                      nfs: path of the shared directory
########
########
########
########################################################################################
###############
****  SERVERIP   ok
**** SERVERIP: 172.26.1.21
######################################
######   BOOT_DIR= /home/hico/share/dimm-mx6q/root/rootfs    ######
######################################
######################################
######   NFS_SHARE= /home/hico/share/dimm-mx6q  ######
######################################
#### Rescue CMD: production
#### Server IP:  172.26.1.21
#### Boot Mode:  nfs
#### Boot Dir:   /home/hico/share/dimm-mx6q/root/rootfs
####   BOOT_DIR= /home/hico/share/dimm-mx6q/root/rootfs
####  NFS_SHARE= /home/hico/share/dimm-mx6q
####################################################################
########  Do NOT power off or reset while producing board   ########
########           producing without rescue system          ########
####################################################################
####################################################################
#############      creating partitions on mmc0      ################
```

```
######################################################################
10+0 records in
10+0 records out
****************************************************************************
**********                                                        **********
**********   mmcblk0:  INFO= 2095MB  SIZE= 2095  UNIT= MB   **********
**********                                                        **********
****************************************************************************
#############     detected unit MB     ################
###########      SIZE of linux partition: 2095     ################
#############      creating partitions on mmc0      ################
10+0 records in
10+0 records out
****  CREATING_PARTITIONS   ok
Model: SD 2GB (sd/mmc)
Disk /dev/mmcblk0: 2095MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name   Flags
 1      1049kB  512MB   511MB   ext4         linux
 2      512MB   2095MB  1583MB  ext4         data

######################################################################
#############    formating root partition           ################
######################################################################
sh: -q: unknown operand
sh: -q: unknown operand
mke2fs 1.42.1 (17-Feb-2012)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
124928 inodes, 498688 blocks
24934 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67633152
61 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

mke2fs 1.42.1 (17-Feb-2012)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
96768 inodes, 386555 blocks
19327 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=398458880
12 block groups
32768 blocks per group, 32768 fragments per group
8064 inodes per group
Superblock backups stored on blocks:
```

```
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

****  BOOT_PARTITION_FORMATED   ok
EXT4-fs (mmcblk0p1): mounted filesystem with ordered data mode. Opts: (null)
EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
****  MOUNTING_FS   ok
####################################################################
#############  preparing and filling rescue system  ################
####################################################################
####################################################################
########                                                   #########
########  Do NOT power off or reset while getting the rfs  #########
########                                                   #########
########  Please wait while updating the root file system  #########
########         This can take a few minutes               #########
########                                                   #########
####################################################################
****  REQUESTING_RFS_STARTED   ok
****  INSTALLING_RFS_STARTED   ok
nf_conntrack: automatic helper assignment is deprecated and it will be removed soon. Use
the iptables CT target to attach helpers instead.
****  RFS_INSTALLED   ok
****  UNMOUNTING_FS   ok
####################################################################
#############  setting realtime clock               ################
####################################################################
New date from ntp server received, updating RTC now
New time and date now:
Sat Nov 10 13:22:04 UTC 2012
****  REALTIME_CLOCK_SET   ok
****  PRODUCTION   ok
INIT: Switching to runlevel: 6
INIT: Sending processes the TERudhcpc (v1.19.4) started
Sending discover...
Sending select for 172.26.1.10...
Lease of 172.26.1.10 obtained, lease time 43200
Stopping syslogd/klogd: no syslogd found; none killed
Deconfiguring network interfaces... done.
Sending all processes the TERM signal...
Sending all processes the KILL signal...
Unmounting remote filesystems...
Deactivating swap...
Unmounting local filesystems...
Rebooting... Restarting system.
```

The system image and data partition are re-programmed and the system can be restarted.

5 Android Application quick start Guide

5.1 Preparation

First of all, you need the VM to run on your computer. You can use the provided VMware player and the emDroid VM included in the DVD with the starter kit.

The emDroid VM is an Ubuntu 12.04 LTS 32bits Linux machine with everything installed for you to start developing an Android Application. Android Studio is preinstalled in the version 1.0.1.

The login and the password are the same: **hico**

Once you are up and running, you can power up your dim-mx6 development board and connect the USB Device (connector J23 the Cadun baseboard) to your computer.

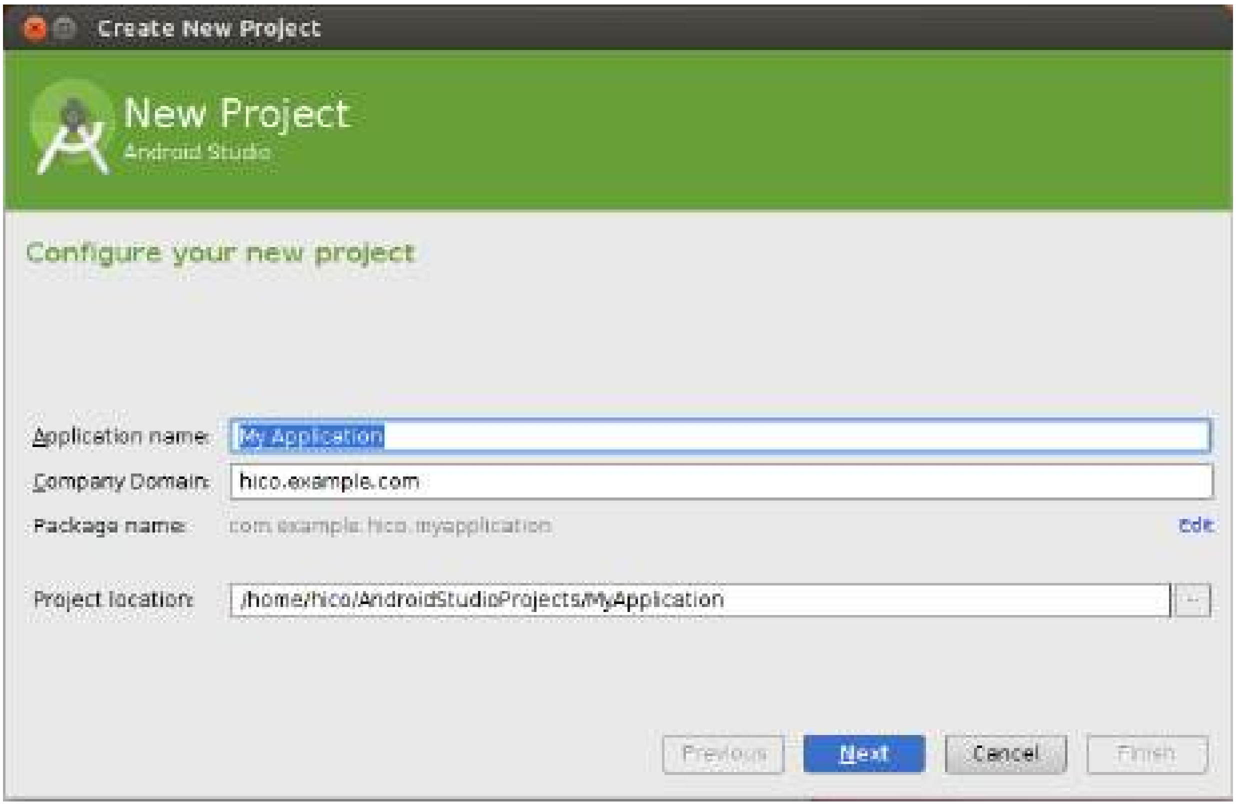
When the Android operating system is functional on the development board, you can click on the android-studio application icon available on the emDroid VM:

android-studio

The Android studio application is now starting you are ready to start.

5.2 My Application: an Hello World example

You can start by clicking on a new project or use the recent project if available.

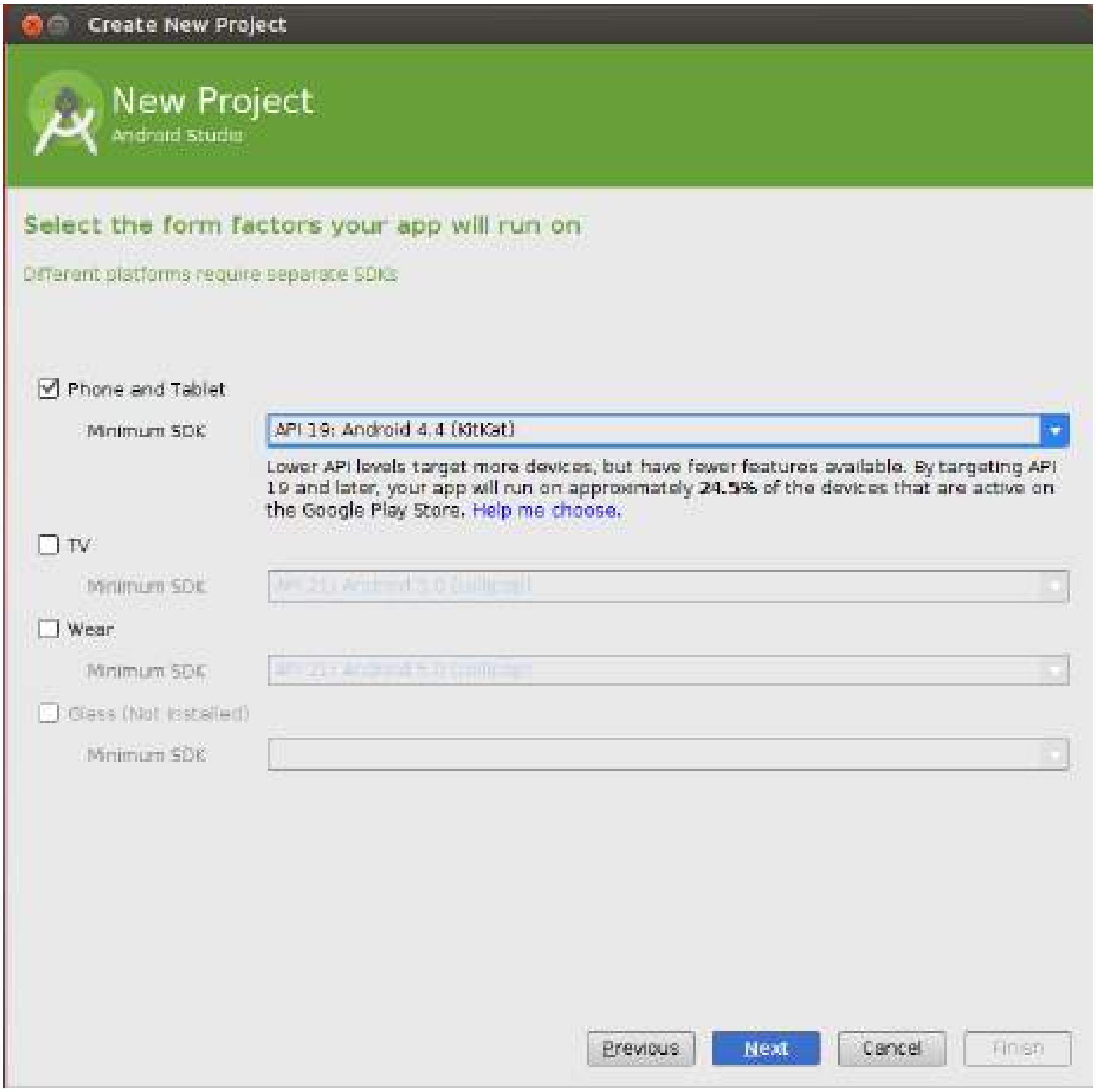


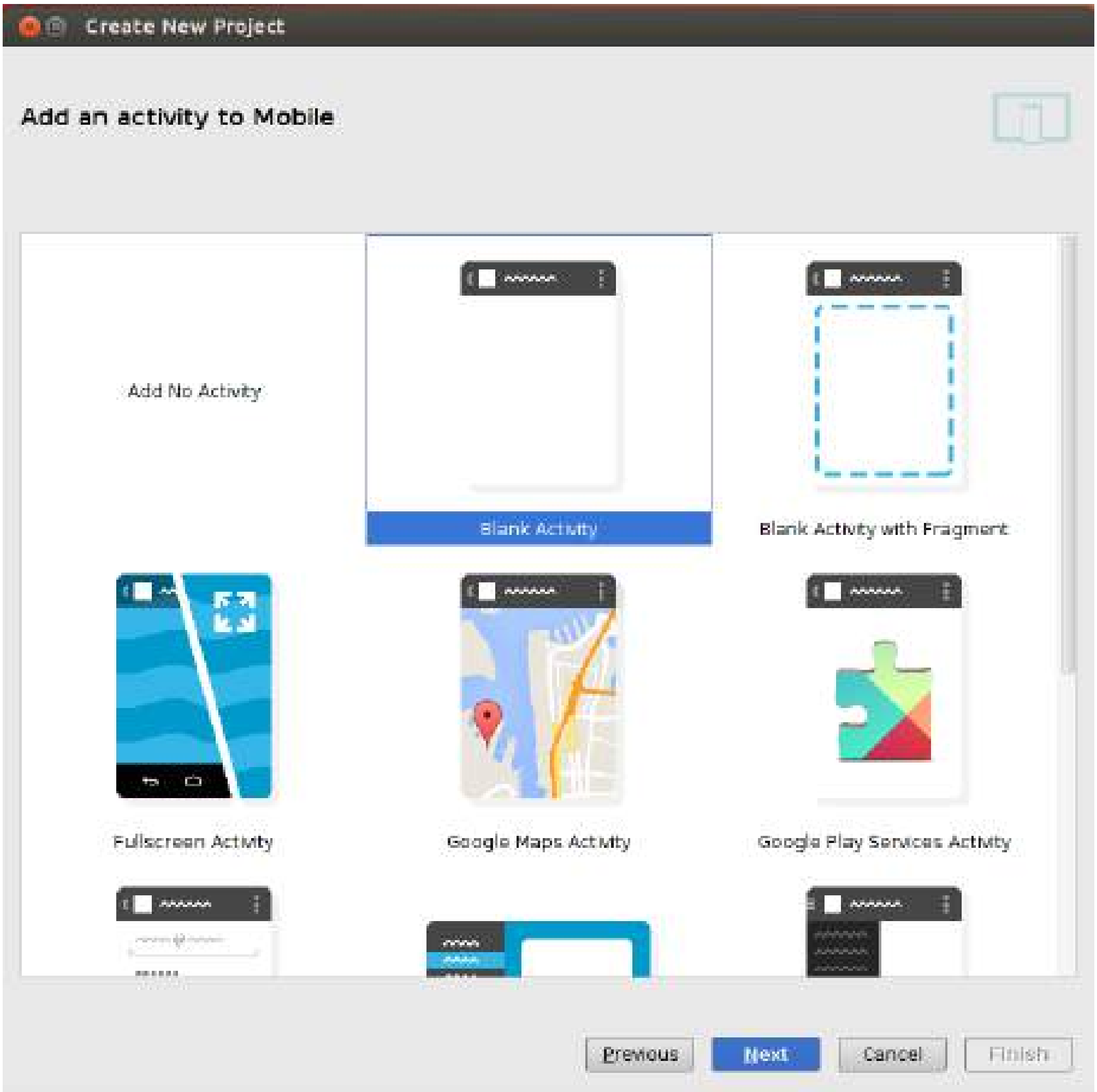
Follow the next instructions:

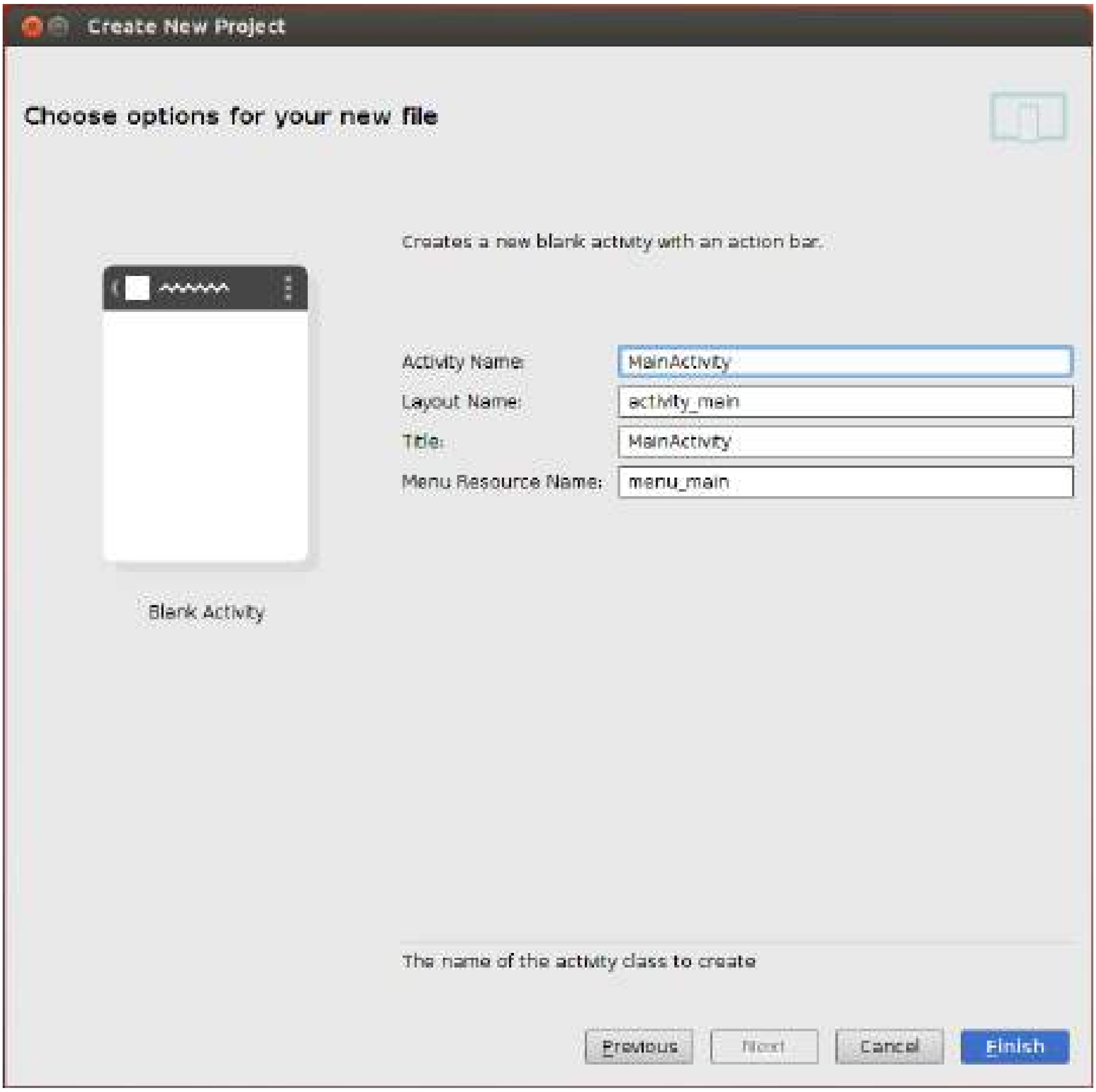
1) Name your project (ex: My Application)
2) In Package name: change com.example.myapplication (ex: com.emtrion.application)
3) Minimum required SDK: API 19
4) Target SDK: API 19

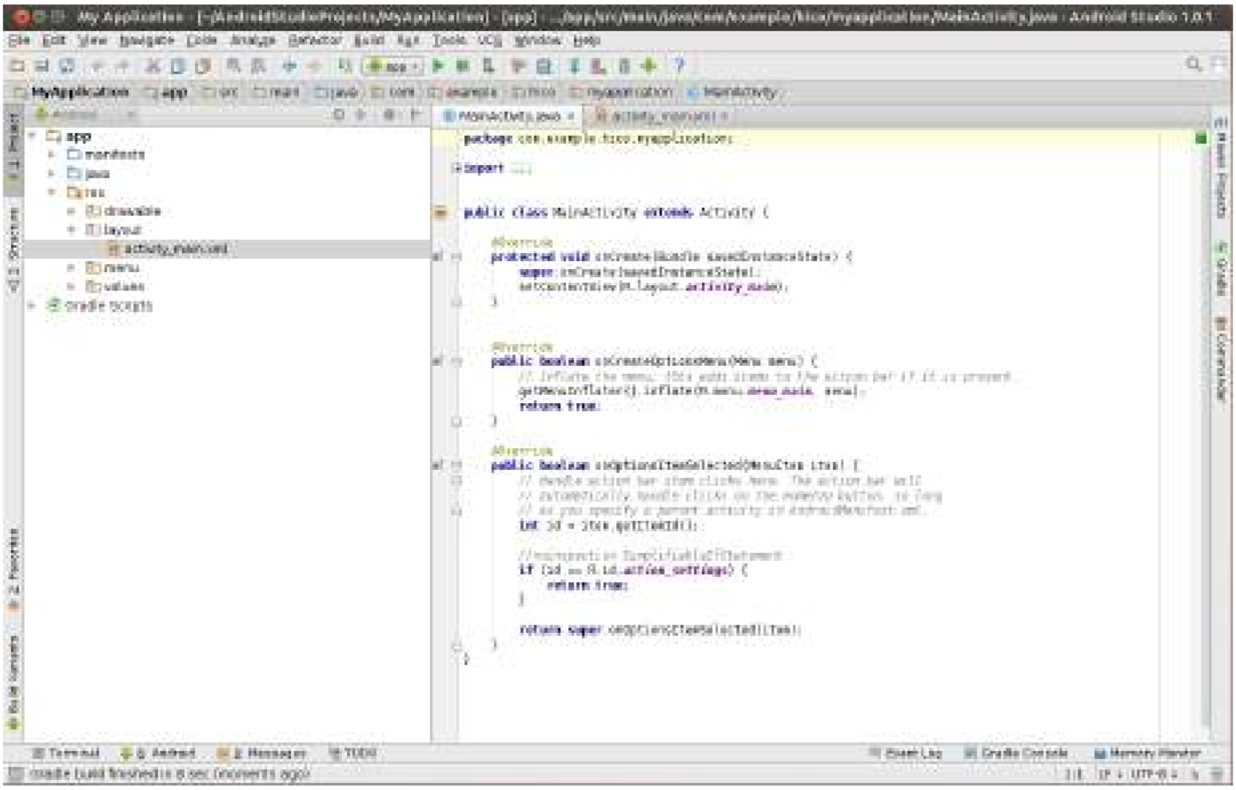
Now, you click on "Next" until "Finish".

Now the IDE is launching. You are reading to start coding.

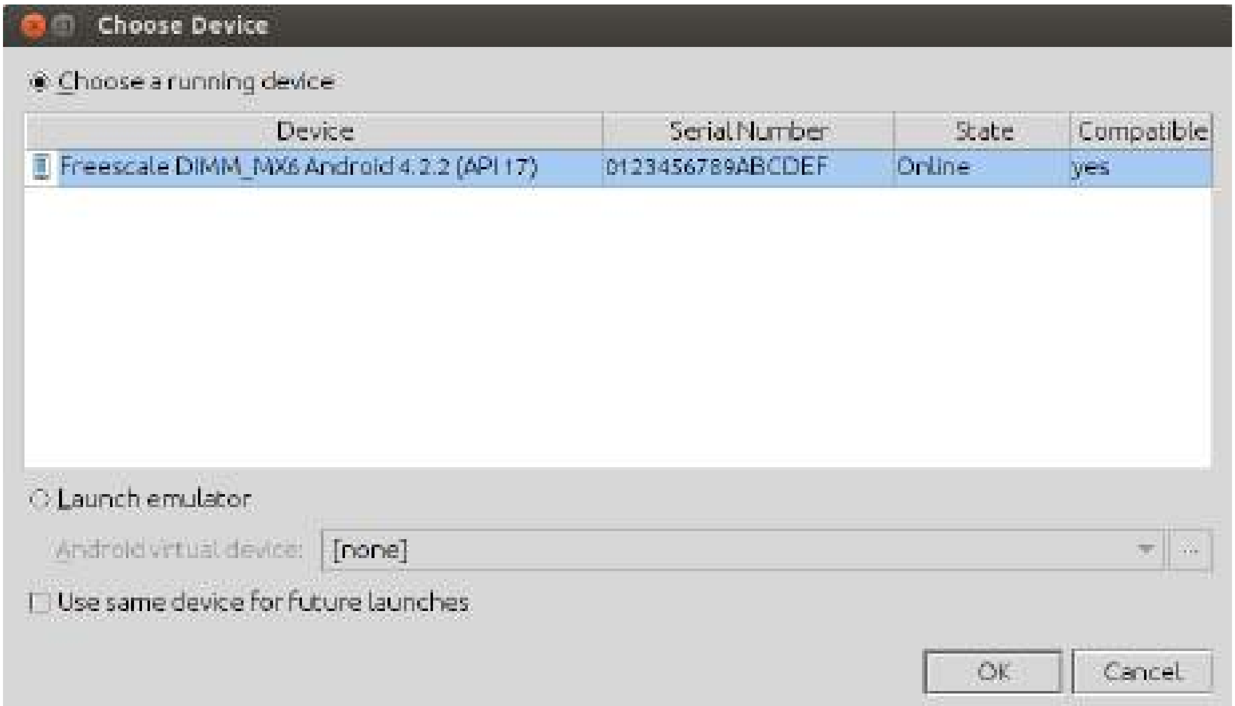

Create New Project
New Project
Android Studio
Select the form factors your app will run on
Different platforms require separate SDKs
Phone and Tablet
Minimum SDK
API 19: Android 4.4 (KitKat)
Lower API levels target more devices, but have fewer features available. By targeting API 19 and later, your app will run on approximately 24.5% of the devices that are active on the Google Play Store. Help me choose.
TV
Minimum SDK
Wear
Minimum SDK
Glass (Not Installed)
Minimum SDK
Previous
Next
Cancel
Finish


Create New Project
Add an activity to Mobile
Add No Activity
Blank Activity
Blank Activity with Fragment
Fullscreen Activity
Google Maps Activity
Google Play Services Activity
Previous
Next
Cancel
Finish


Create New Project
Choose options for your new file
Creates a new blank activity with an action bar.
Activity Name:
MainActivity
Layout Name:
activity_main
Title:
MainActivity
Menu Resource Name:
menu_main
Blank Activity
The name of the activity class to create
Previous
Next
Cancel
Finish



Once you are ready, you can either click on the green triangle run “My Application” to run the application in “release” mode or you can lunch the Debugger (the tiny green bug on the right).



The IDE ask you to choose the device. You normally should see the DIMM_MX6 Android device or the EMCON_MX6 Android device.

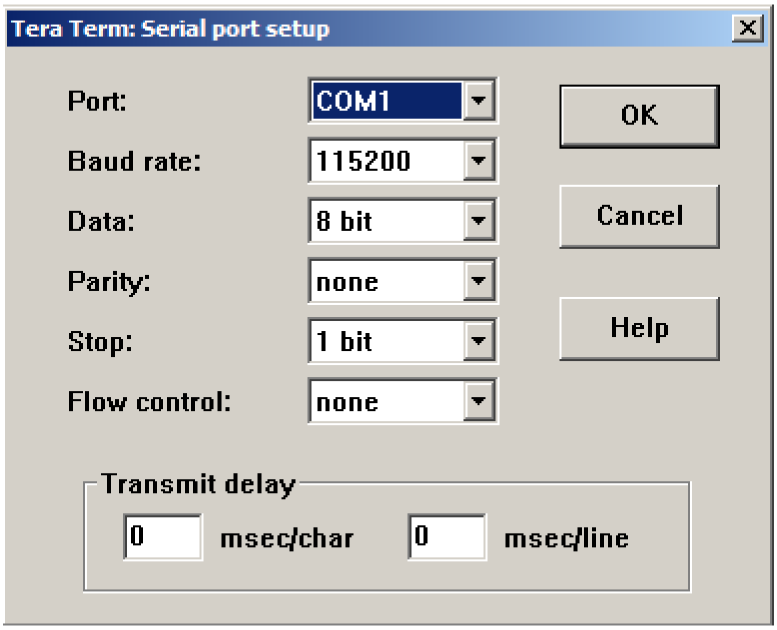
Once running, you can see your Application on the Development board:

Congratulation you are running your first Android application on the DIMM-MX6!

6 Running Android via NFS

6.1 Setting up the bootloader:

First of all, connect your board to a serial line with the following parameter:



Then power up your board and hit quickly any keyboard key to stop the auto-boot like this:

```
U-Boot 2013.04-00005-g83a2fe5 (Oct 30 2013 - 14:25:50)

CPU:   Freescale i.MX6Q rev1.2 at 792 MHz
Reset cause: WDOG
Board: DIMM-MX6
DRAM:  1 GiB
PMIC: PFUZE100 device id=10
MMC:   FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
SF: Detected MX25L6405D with page size 64 KiB, total 8 MiB
No panel detected: default to UMSH
In:    serial
Out:   serial
Err:   serial
Net:   FEC [PRIME]
Hit any key to stop autoboot:  0
DIMM-MX6 U-Boot > 
```

Then type the following line and hit enter:

```
set serverip xx.xx.xx.xx
```

with “xx.xx.xx.xx” being the ip address of your VM.

Tips : You can find your ip address by just typing in a terminal “ifconfig” and you’ll find it on this line: “inet addr: 192.168.1.2”.

Then type the following line and hit enter:

```
set nfsroot '/home/hico/share/dim-mx6q/root/rootfs'
```

and to finish, type the following line:

```
saveenv
```

Then you can type this line to run the NFS on the target board:

```
run net_boot
```

Normally, the target board will run Android after ~30 seconds.

If you want to run the Android on NFS at startup, type this command line:

```
set bootcmd 'run net_boot'
```

Then

```
saveenv
```

Any time, you can get back to flash booting by typing:

```
setenv bootcmd 'run flash_boot'
```

6.2 Troubleshouting

In case of NFS mounting error, like the following:

```
DIMM-MX6 U-Boot > run net_boot
BOOTP broadcast 1
DHCP client bound to address 172.26.1.19
Using FEC device
File transfer via NFS from server 172.26.1.13; our IP address is 172.26.1.19
Filename '/home/hico/android_nfs/dimm_mx6_android/root/rootfs/boot/uboot_script'.
Load address: 0x10800000
Loading: *** ERROR: Cannot mount
T T T T T T T T T T T T T T T T T T T T T T T T T T T T T T
```

On your Host computer, inside the VM, open a terminal and type:

```
sudo service avahi-daemon stop
```

And

```
sudo service nfs-kernel-server restart
```

7 Android for advanced user

7.1 Modifying the Android File System for DIMM-MX6

Within the VM, you can modify files of the Android File System to modify the behavior. This is only recommended for advanced user as you can break your NFS.

For example, you can modify the *.RC files and change the log level of the OS.

Instead of modifying directly the File system here:

```
/home/hico/share/dimm-mx6q/root/rootfs/
```

You must do your change in:

```
/home/hico/dimm_mx6-fs
```

Once ready to test, you can start this script in the folder ~/script :

```
./dim_mx6_prep_nfs.sh
```

The script copies all files from dimm_mx6-fs folder to the NFS path with the proper files rights.

Now you can test your modification by testing via NFS (See section 6).

7.2 Modifying the Android File System for EMCON-MX6

Within the VM, you can modify files of the Android File System to modify the behavior. This is only recommended for advanced user as you can break your NFS.

For example, you can modify the *.RC files and change the log level of the OS.

Instead of modifying directly the File system here:

```
/home/hico/share/emcon-mx6q/root/rootfs/
```

You must do your change in:

```
/home/hico/emcon_mx6-fs
```

Once ready to test, you can start this script in the folder ~/script :

```
./emcon_mx6_prep_nfs.sh
```

The script copies all files from dimm_mx6-fs folder to the NFS path with the proper files rights.

Now you can test your modification by testing via NFS (See section 6).

7.3 Rebuilding the package for DIMM_MX6

When you are satisfied with your modified file system, you can build packages and flash it into your target.

To do this, simply run the following script in the folder ~/script:

```
./dimm_mx6_userland_package.sh
```

It creates the following packages in your Data partition (see section 7.3):

```
android-datafs-dimm_mx6.tar.bz2
android-rootfs-dimm_mx6.tar.bz2
```

Then you can replace the packages files by the one in:

```
/home/hico/share/dimm-mx6q/images
```

And launch an update of the Android System (see section 4.4)

emtrion
embedded systems

7.4 Rebuilding the package for EMCON_MX6

When you are satisfied with your modified file system, you can build packages and flash it into your target.

To do this, simply run the following script in the folder ~/script:

```
./emcon_mx6_userland_package.sh
```

It creates the following packages in your Data partition (see section 7.3):

```
android-datafs-emcon_mx6.tar.bz2
android-rootfs-emcon_mx6.tar.bz2
```

Then you can replace the packages files by the one in:

```
/home/hico/share/emcon-mx6q/images
```

And launch an update of the Android System (see section 4.4)

7.5 Backup

The VM include an extra partition called “Data”.

You can find it here:

```
/media/Data
```

Within this extra partition you find backups of your original Android file system and your original Android image. This is also the location of your custom packages build.

Present Archives in sub directory dimm_mx6:

- Images.tar : the content of /home/hico/share/dimm-mx6q/images (i.e original packages system)
- dimm_mx6-fs.tar : the content of the Android NFS root file system.

Present Archives in sub directory emcon_mx6:

- Images.tar : the content of /home/hico/share/emcon-mx6q/images (i.e original packages system)
  emcon_mx6-fs.tar : the content of the Android NFS root file system.