



Application Development Kit for Android™
Installation Guide

©2012 Digi International Inc.
All rights reserved.

Digi, Digi International, the Digi logo, the Digi website, a Digi International Company, Digi ESP, and ConnectCore Wi-MX51 are trademarks or registered trademarks of Digi International, Inc. in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners. Specifically, Android is a trademark of Google Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Installation Instructions

This document will guide you through the process of downloading and installing the Digi® Application Development Kit for Android™, and preparing your hardware to boot prebuilt Android™ firmware images. The instructions within this document are valid for ConnectCore for i.MX51 and ConnectCore for i.MX53 targets.

1. Download software

Based on the product you are using, download the Digi Application Development Kit for Android DVD ISO image from one of the following locations:

ConnectCore for i.MX51 users:

http://ftp1.digi.com/support/patches/i.MX51/CC_i.MX51_Updates.htm

ConnectCore for i.MX53 users:

http://ftp1.digi.com/support/patches/i.MX51/CC_i.MX53_Updates.htm

2. Install software from the DVD

To install the software from the DVD perform the following steps:

1. Burn the downloaded ISO image onto a DVD, or mount it in a virtual unit.
2. Run the appropriate installer for your operating system within the “installers” folder on the DVD.
 - For Windows: <DVD>:\installers\windows\setup.exe
 - For Linux: <DVD>/installers/linux/aadk_installer.run
3. Follow the instructions on the installer to complete the installation.

Notes for Linux users:

- The DVD might be mounted without execution permissions which will prevent you from running the installer. Make sure you remount the DVD with execution permissions, or copy the installer to a local folder and give it execution permissions.
- It is recommended to install the software as a standard user, not as root.
- By default, the software will be installed at **/opt/** folder. You must create the install folder with the appropriate permissions for the user, prior to the installation:



```
# install -m 0755 <your_user> -g <your_group> -d /opt/Digi/Android/aadk-X.Y
```

where:

- `<your_user>` is your user name
- `<your_group>` is your user's group name
- `X.Y` is the version of the Digi Application Development Kit for Android.

3. Prepare hardware with Android firmware

This section will guide you through the process of writing Android firmware images to the flash of your hardware platform.

NOTE: In order to update the firmware, it is required that a TFTP server is running on your host computer.

For Windows users, a TFTP server can be downloaded from the same URL you selected in Section 1 - Download software on the previous page.

For Linux users, use your package administrator tool to install any available TFTP server, like `tftpd-hpa`.

3.1 Locate firmware images for your hardware

Software is installed to the following paths by default:

- For Windows: `C:\Program Files\Digi\Android\aadk-X.Y` (where `X.Y` is the software version number)
- For Linux: `/opt/Digi/Android/aadk-X.Y` (where `X.Y` is the software version number)

Navigate to the appropriate default folder for your operating system (or any alternative folder you may have specified) and locate the subfolder called "images". Within this folder, locate the folder for your hardware platform (depending on the platform an additional board version folder may exist).

Copy the following image files to your TFTP server exposed folder (typically called `tftpboot`):

Firmware Image	Filename Pattern	Example Name
U-Boot boot loader	<code>u-boot-<i>platform</i>.bin</code>	<code>u-boot-ccwmx51js.bin</code>
Android Kernel	<code>uImage-android-<i>platform</i></code>	<code>uImage-android-ccwmx51js</code>
Android Root File System	<code>android-<i>platform-eraseblock</i>.ubifs</code> ¹	<code>android-ccwmx51js-128.ubifs</code>

¹ If you don't know the erase block size of the Flash memory of your module, run the command "nand info" in U-Boot to find it.

3.2 Configure the network settings of your platform

Connect a serial cable between the hardware platform and your PC and open a serial terminal. Power up the target and press any key to stop the auto-boot process and break into U-Boot monitor.

Configure the appropriate network settings using U-Boot commands:



```
# setenv ipaddr AAA.AAA.AAA.AAA
# setenv netmask BBB.BBB.BBB.BBB
# setenv serverip EEE.EEE.EEE.EEE
# saveenv
```

where:

- **AAA.AAA.AAA.AAA** is the IP address of the target's wired Ethernet (it must be in the same subnet as the host computer's IP).
- **BBB.BBB.BBB.BBB** is the target's subnet mask.
- **EEE.EEE.EEE.EEE** is the IP address of the host computer.
- The last command, `saveenv`, saves the target's network settings in NVRAM.

Connect your hardware platform to the network using an Ethernet cable.

3.3 Update U-Boot boot loader



Do not power off the module during this process or your device may become un-bootable.

Run the following command to update the U-Boot boot loader (when asked for confirmation press y for yes):



```
# update uboot tftp
Partition marked read-only/fixed. Do you want to continue? (y/n) y
TFTP from server 192.168.103.2; our IP address is 192.168.103.51
Filename 'u-boot-ccwm51js.bin'.
Load address: 0x94000000
Loading: #####
done
Bytes transferred = 402672 (624f0 hex)
Calculated checksum = 0xecdd466c
Do you really want to overwrite U-Boot flash partition? (y/n) y
Updating partition 'U-Boot'
Erasing: complete
Writing: complete
Verifying: complete
Root system to update takes effect
Update successful
```

After the update completes successfully, reset the module to boot with the new boot loader.

3.4 Setup the partition table for Android firmware

Break again into U-Boot monitor and run the “flpart” command to modify the Flash partition table. At the prompt, type the following keys, in order:

- ‘r’ for Reset partition table
- ‘a’ for Android default partition table
- ‘q’ for Quit
- ‘y’ for saving changes



```

CCWMX51 # flpart
Nr | Name | Start | Size | Type | FS | Flags
-----|-----|-----|-----|-----|-----|-----
0 | U-Boot | 0 | 768 KiB | U-Boot | | fixed
1 | NVRAM | 768 KiB | 512 KiB | NVRAM | | fixed
2 | Splash | 1280 KiB | 1 MiB | Splash-Screen | | 
3 | Kernel | 2304 KiB | 3 MiB | Linux/Android-Kernel | | 
4 | RootFS | 5376 KiB | 120 MiB | Filesystem | JFFS2 | rootfs
5 | UserFS | 128256 KiB | 396032 KiB | Filesystem | JFFS2 | 
Commands:
a) Append partition
d) Delete partition
m) Modify partition
p) Print partition table
r) Reset partition table
q) Quit
Cmd (? for help) > r
OS Types:
n) None
l) Linux
a) Android
w) WinCE
o) NET+OS
OS (? for help)> a
Partition table reset
Nr| Name | Start | Size | Type | FS | Flags
-----|-----|-----|-----|-----|-----|-----
0 | U-Boot | 0 | 768 KiB | U-Boot | | fixed
1 | NVRAM | 768 KiB | 512 KiB | NVRAM | | fixed
2 | Splash | 1280 KiB | 1 MiB | Splash-Screen | | 
3 | Kernel | 2304 KiB | 3 MiB | Linux/Android-Kernel | | 
4 | RootFS | 5376 KiB | 120 MiB | Filesystem | UBIFS | rootfs
5 | UserFS | 128256 KiB | 396032 KiB | Filesystem | UBIFS | rootfs
Cmd (? for help)> q
Partition table has been modified. Save? (y/n): y
Writing Parameters to NVRAM
Warning: The following UBI partitions were modified:
- RootFS
- UserFS
A firmware update of the listed partitions is needed to recreate their UBI
volumes.
CCWMX51 #

```

3.5 Temporarily Change the loadaddr Variable

Due to Android's rootfs image size, we need to temporarily change the loadaddr variable in U-Boot. This is the address where U-Boot's TFTP command puts downloaded images.

Make sure you do not execute 'saveenv' to avoid changing the saved 'loadaddr'.



```
CCWMX51 # set loadaddr a0000000
```

NOTE: Set this 'loadaddr' variable for the MX51 module only.

3.6 Write the Android Kernel

Run the following command to update the Android kernel partition from U-Boot:



```
# update android tftp
TFTP from server 192.168.103.2; our IP address is 192.168.103.51
Filename 'uImage-android-ccwmx51js'.
Load address: 0x94000000
Loading: #####
          #####
          #####
done
Bytes transfered = 2689420 (29098c hex)
Calculated checksum = 0x91b899f0
Erasing: complete
Writing: complete
Verifying: complete
Update successful
```

3.7 Write the Android Root File System

Run the following command to update the Android root file system partition from U-Boot (remember to pass the filename with the appropriate platform and erase block size):



```
# update androidfs tftp
Creating 1 MTD partitions on "nand0":
0x0000000540000-0x0000008540000 : "mtd=4"

Removing all user volume
All volumes removed
Creating dynamic volume RootFS of size 130314240
Using FEC0 device
TFTP from server 192.168.103.2; our IP address is 192.168.103.51
Filename 'android-ccwmx51js-128.ubifs'.
Load address: 0x91000000
Loading: #####
#####
...
#####
#####
#####
done
Bytes transfered = 79994880 (4c4a000 hex)
Calculated checksum = 0x2da7a85a

Writing UBI image
Volume "RootFS" found at volume id 0
Volume RootFS found at volume id 0
read 79994880 bytes from volume 0 to 91000000 (buf address)
Update successful
```


4. Setup the video

Run the U-Boot command “video” to setup the video interface that's connected to your hardware. If you have an LCD, the device can be controlled using the touch screen. If you have a VGA or HDMI monitor, you will need a USB mouse and keyboard to interface with it.

The first time it boots, Android firmware launches a touch screen calibration utility that cannot be operated except with a touch screen interface. Therefore, you must (at least for the first time) setup the video for the LCD and connect the LCD to your hardware board:



```
# video
Video interface
  1) Video 1
  2) Cancel
Select interface: 1

Displays/video mode
  1) VGA
  2) HDMI
  3) LCD display
  4) Cancel
Select displays/video mode: 3

LCD display
  1) LQ070Y3DG3B (800x480 JumpStart Kit display)
  2) LQ064V3DG01 (640X480)
  3) LQ121K1LG11 and LQ121K1LG52(1280X800)
  4) LQ121K1LG52 (1280X800)
  5) LQ106K1LA05 (1280X768)
  6) LQ104V1DG62 (640X480)
  7) custom1 configuration
  8) custom2 configuration
  9) cancel
Select LCD display: 1

Setting variable:
  video=displayfb:LCD@LQ070Y3DG3B
Remember to save the configuration with 'saveenv'

# saveenv
```

Execute the command “saveenv” to save the video configuration.



The maximum supported resolution for HDMI for the ConnectCore for i.MX51 is 1280x720.

The maximum supported resolution for HDMI for the ConnectCore for i.MX53 is 1920x1080 (Full HD).

5. Setup default boot command

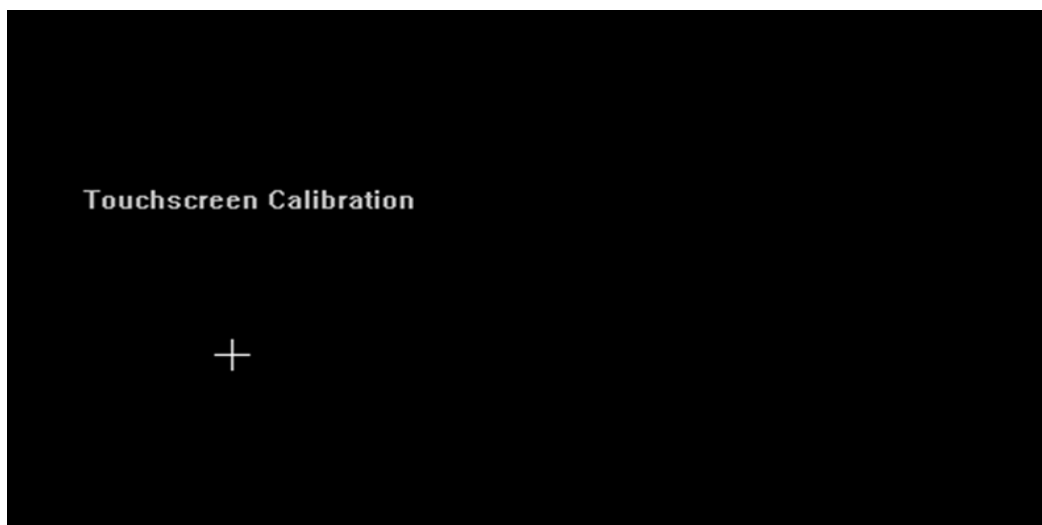
Run the following commands from U-Boot to make the Android firmware auto-boot from the Flash:



```
# setenv bootcmd dboot android flash
# saveenv
```

6. Start Android firmware and calibrate the touch screen

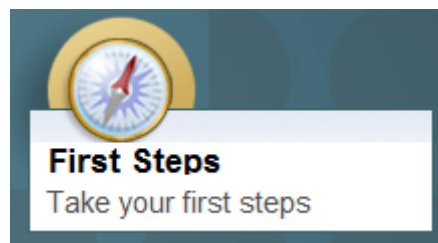
Reset your target and let it auto-boot the Android firmware. The first time it boots, the Touchscreen Calibration utility will run. Touch at the cross (shown below) to calibrate the touch screen.



After the touch screen is calibrated wait a few some seconds for the Android firmware to boot completely.

7. What's next?

On your host computer, run Digi ESP™ for Android. Within Digi ESP navigate to the **Help > Welcome** menu. From the Welcome page, click on the **First Steps** icon (as shown).



Within the First Steps page click on the **Getting Started with the Android Application Development Kit** link. The instructions displayed will guide you through the process of creating, downloading and running your first Android application.