



LX -TI DM8148

Technical Documentation

Software Developer Guide

09/10/2012

Document version :	0.2
Date :	09/10/2012
Redactor :	Romain Duchêne
Approbation :	David Freidenberg Shai Felman

LX TI DM8148

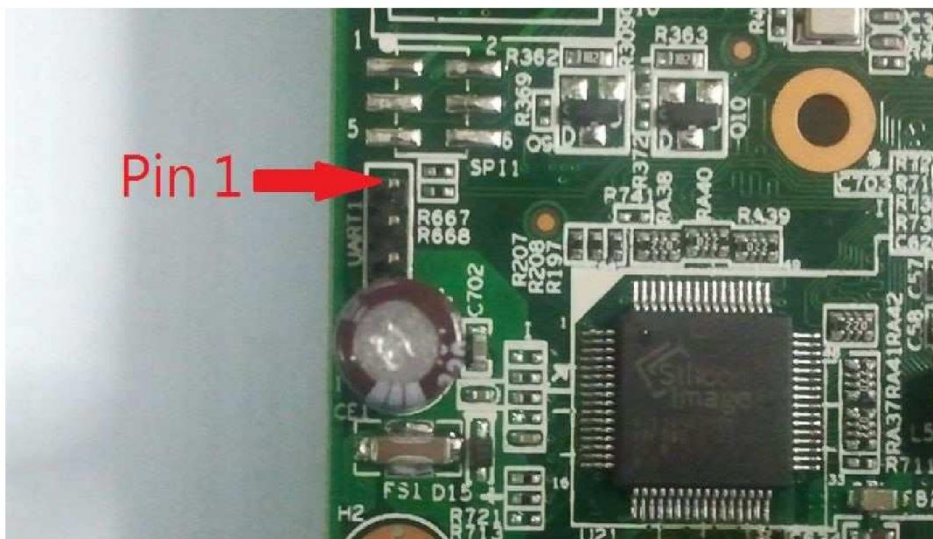
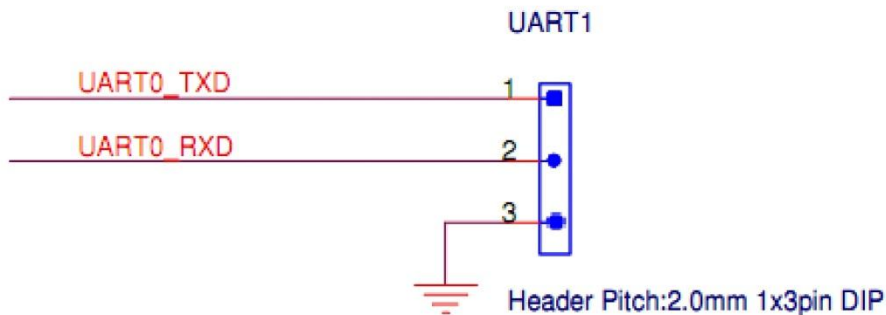
Software Developer Guide

CONTENTS

1. DEBUG PORT USAGE.....	3
2. SOFTWARE DEVELOPMENT.....	4
2.1 Environment	4
2.1.1 Install Toolchain.....	4
2.1.2 Install EZSDK.....	4
2.1.3 Install RDK	4
2.1.3.1 Apply EZSDK and LX TI DM8148 Patches.....	5
2.1.3.2 Install RemoteFX Component	5
2.2 How to Build.....	5
2.2.1 Build Everything	5
2.2.2 Build Specific Components	6
2.2.2.1 Set Installation Target Path	6
2.2.2.2 Compile x-loader and u-boot	7
2.2.2.3 Compile Kernel	7
2.2.2.4 Compile Extra Kernel Modules	8
2.3 Update Boot Files and Kernel Modules	8
2.4 Update Rootfs	10
2.5 Using External Rootfs.....	10
2.6 Fail Recovery - UART boot.....	12
3. REFERENCE LINKS	14

1. DEBUG PORT USAGE

There is a debug port noted as “UART1” on the board. To connect debug port with host, you can use *Chip PC USB-to-serial fixture* or make a similar cable by yourself. The pin definition of UART1 can be found as below:



If you use *Chip PC USB-to-serial fixture*, please download the VCP driver and install it on your host. The VCP driver can be found in the following link:
<http://www.ftdichip.com/Drivers/VCP.htm>

You can use terminal emulator software such as Tera Term (<http://logmett.com/index.php?/download/tera-term-474-freeware.html>) to control the console interface on host side. Please remember set UART baud rate to **115200**, otherwise the terminal emulator can't correctly show the console messages.

2. SOFTWARE DEVELOPMENT

2.1 Environment

Host: Ubuntu 11.04 32-bit

Toolchain: Sourcery G++ Lite 2009q1-203

EZSDK version: 5.03.01.15

RDK version: 0.6 (under TI NDA only)

Note! In the following sections, we use $\{\text{EZSDK}\}$ and $\{\text{RDK}\}$ to indicate EZSDK and RDK installation paths separately.

Note! In the following sections, we use some tags listed below to indicate which environment the commands apply in.

Host:\$	means PC host
TI-MIN#	means x-loader
TI8148_EVM#	means u-boot
LXTI:\$	means command line mode on LX TI DM8148

2.1.1 Install Toolchain

Please visit the link below. Find “Download the GNU/Linux Release” under the “ARM processors” block and fill the information to get direct download links.

<http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-edition/>

Then open the direct download links. Choose “Sourcery G++ Lite 2009q1-203” and “IA32 GNU/Linux Installer” to download toolchain bin file. Finally copy “arm-2009q1-203-arm-none-linux-gnueabi.bin” to host and start installation.

2.1.2 Install EZSDK

You can find DM814x EZSDK in the TI site below: http://software-dl.ti.com/dsps/dsps_public_sw/ezsdk/5_03_01_15/index_FDS.html

Then run the downloaded bin file in terminal and pass the option “--forcehost” to install EZSDK in Ubuntu 11.04.

2.1.3 Install RDK

Double click RDK bin file to install it. Then extract generated tarball file to the path you like as $\{\text{RDK}\}$.

2.1.3.1 Apply EZSDK and LX TI DM8148 Patches

Please follow the commands below to apply patches to U-boot and Kernel source code and other EZSDK components.

```
Host:$ cd ${EZSDK}
Host:$ patch -p1 < ${RDK}/ubuntu-script_0.6/ezsdk-patches/all-in-one.patch
Host:$ for f in ${RDK}/ezsdk-patches/clientron-D500/00*; \
do patch -p1 < $f; done
```

Note! If you have a v0.3 Chip PC LX TI DM8148 board, apply the patches in `${RDK}/ezsdk-patches/clientron-D500/v0.3` as well.

2.1.3.2 Install RemoteFX Component

If you need to test RemoteFX function, please follow the commands below to install extra RemoteFX component to EZSDK.

```
Host:$ cd ${RDK}/ubuntu-script_0.6/rfx
Host:$ ./install.sh ${EZSDK}
```

2.2 How to Build

In the RDK development environment (under TI NDA only), you can build u-boot, kernel and other firmware together or just compile those components you modified. Please refer the following subsections for detail steps.

2.2.1 Build Everything

Before you start building RDK, please open the file, $\$(RDK)/ubuntu-script_0.6/build-scripts/target/dm814x/binary/debs/install.sh$, and modify the “HTTP_PROXY” as your company proxy settings. Then you can use commands below to build every component LX TI DM8148 need and install them to rootfs contents.

```
Host:$ cd  $\$(RDK)/ubuntu-script_0.6/build-scripts/host/dm81xx$   
Host:$ ./build.sh --sdk  $\{EZSDK\}$  --platform d500-v0.3
```

The default action of “build.sh” is checking if there is a rootfs tarball in current path. If there is a matched rootfs tarball, it will be extracted and then the script will install the built firmware and other prebuilt packages to it. Or the script will generate a new ubuntu rootfs as the installation target. You can also force script to generate a new ubuntu rootfs by adding rootstock parameter as below.

```
Host:$ cd ${RDK}/ubuntu-script_0.6/build-scripts/host/dm81xx
Host:$ ./build.sh --sdk ${EZSDK} --platform d500-v0.3 --rootstock \
--wmgr lxde
```

2.2.2 Build Specific Components

For efficient development, you can only compile specific parts and skip the other untouched parts. Here we introduce the steps of compiling most frequently modified parts: boot loader and kernel.

2.2.2.1 Set Installation Target Path

The path which the rebuilt u-boot/kernel files will be installed to is set in the file `${EZSDK}/Rules.make`. You can find the string “EXEC_DIR=” at the end of this file. Then modify it as where you want to place the rebuilt files.

2.2.2.2 Compile x-loader and u-boot

x-loader and u-boot files are TI two-stage boot loaders and both necessary as LX TI DM8148 boot files. Use the commands below to build x-loader and u-boot. The x-loader file is named as “MLO” and u-boot file is named as “u-boot.bin” and both files are placed in `${EXEC_DIR}/boot/`.

```
Host:$ cd ${EZSDK}
Host:$ make u-boot_clean
Host:$ make u-boot
Host:$ make u-boot_install
```

2.2.2.3 Compile Kernel

Follow the commands below to build kernel and kernel modules. The kernel file is named as “uImage” and placed in `${EXEC_DIR}/boot /`. The built kernel modules are also placed in `${EXEC_DIR}`.

```
Host:$ cd ${EZSDK}
Host:$ make linux_clean
Host:$ make linux
Host:$ make linux_install
```

2.2.2.4 Compile Extra Kernel Modules

There are some modules needed on DM8148 platform and these source codes do not join to the kernel tree. Therefore follow the commands below to build these modules and install to `$_EXEC_DIR`.

```
Host:$ cd $_EZSDK}
Host:$ make cmem_clean
Host:$ make cmem
Host:$ make cmem_install
Host:$ make syslink_clean
Host:$ make syslink
Host:$ make syslink_install
Host:$ make rfx_module_clean
Host:$ make rfx_module
Host:$ make rfx_module_install
Host:$ sudo depmod -ae -F $_EXEC_DIR}/boot/System.map \ -b
    $_EXEC_DIR} 2.6.37
```

2.3 Update Boot Files and Kernel Modules

The basic boot files required by LX TI DM8148 are MLO (i.e. x-loader), u-boot.bin (i.e. u-boot), uImage (i.e. kernel) and boot.scr.

“boot.scr” includes the bootargs and indicates what u-boot does to load kernel and start booting procedure. The default boot.scr content is pasted below. If you want to create a new boot.scr, please refer to `$_EZSDK}/bin/README.boot.scr`.

```
setenv bootargs 'console=ttyO0,115200n8 serialtty=ttyO0 root=/dev/mmcbk0p2
rw mem=352M mem=320M@0x9FC00000 earlyprink
notifyk.vpssm3_sva=0xBF900000 vmalloc=500M vram=102M printk.time=y
noinitrd rootfstype=ext3 ip=off rootwait'
setenv bootcmd 'mmc init; if fatload mmc 1 0x80009000 uImage; then bootm
0x80009000; fi'
boot
```

DM8148 has some strict definition with FAT32 format. Therefore if you have some special application with the first FAT32 partition of eMMC, please take

`${EZSDK}/bin/mksdboot.sh` as reference.

Chip PC
Tel: +972 (4) 8501-121
info@chippc.com

Chip PC (UK) Ltd
Tel: +44 (0) 870 606
1007 info@chippc.co.uk

Chip PC GmbH
Tel: +49 (0) 201 4513 100
info@chippc.de

Chip PC France
Tel: +33 1 83 62 05 12
info.fr@chippc.com

Chip PC Inc US
Tel: +1 972-827-0379
info.us@chippc.com

For updating boot files, the best way is listed below:

Step 1. Backup the original boot files.

Step 2. Format partition 1 by the following commands.

```
LXTI:$ sudo umount /dev/mmcbk0p1
LXTI:$ sudo mkfs.vfat -F 32 -n boot /dev/mmcbk0p1
```

Step 3. Copy the new boot files to the clean FAT32 partition.

Because file modification on FAT32 partition may be failed easily (especially on small files), we suggest that use a big dummy file to make sure that all boot files are complete copied to FAT32 partition. Or LX TI DM8148 may fail on boot due to the incomplete boot files. Here we list a sample sequence of step 3.

(Assume the FAT32 partition mounting path is `${BOOT}` and the directory which contains new boot files is denoted as `${newbootfiles}`)

```
LXTI:$ cd ${newbootfiles}
LXTI:$ sudo mount /dev/mmcbk0p1 ${BOOT}
LXTI:$ sudo cp boot.scr ${BOOT}
LXTI:$ sudo cp MLO ${BOOT}
LXTI:$ sudo cp u-boot.bin ${BOOT}
LXTI:$ sudo cp uImage ${BOOT}
LXTI:$ sudo cp uImage ${BOOT}/dummy
LXTI:$ sync
```

The kernel modules in rootfs may also need to be updated to match the updated kernel. Please follow the commands below to backup old kernel modules and install the new ones.

```
Host:$ cd ${EXEC_DIR}/lib/modules Host:$ tar -jcvf kmodules.tbz 2.6.37
(Then copy kmodules.tbz to LX TI DM8148.)

LXTI:$ cd /lib/modules
LXTI:$ sudo mv 2.6.37 2.6.37.bak
LXTI:$ sudo tar -jxvf kmodules.tbz
LXTI:$ sync
```

2.4 Update Rootfs

The only way to update the onboard eMMC rootfs at power on status is using external rootfs. (Please refer section 2.5 for external rootfs mounting method.) When LX TI DM8148 uses external rootfs to boot up, the onboard eMMC becomes a simple storage device. Then the developer can easily format the old rootfs partition and put the new one.

Therefore the procedure of updating eMMC rootfs is:

Step 1. Use external rootfs to boot LX TI DM8148.

Step 2. Format eMMC partition 2 by the following commands.

```
LXTI:$ sudo umount /dev/mmcblk0p2  
LXTI:$ sudo mkfs.ext3 -j -L rootfs /dev/mmcblk0p2
```

Step 3. Extract new rootfs to eMMC as follows.

```
LXTI:$ sudo mount /dev/mmcblk0p2 /mnt  
LXTI:$ sudo tar --numeric-owner -zxvf <new_rootfs> -C /mnt  
LXTI:$ sync
```

*1. Please change <new_rootfs> as your new rootfs tarball name.

2.5 Using External Rootfs

For making development process more efficiency, the developer may prefer external rootfs than the rootfs on eMMC. The only work needed to do is set bootargs in u-boot, and then you can use the rootfs on **USB HDD** or **NFS**.

The steps of using external rootfs are listed below:

Step 1. Connect debug port with PC and power up LX TI DM8148.

Step 2. When the Centaurus logo (*2) shows on terminal emulator, press any key to stop at u-boot.

Step 3. Paste the different commands below to use rootfs from USB HDD or NFS.

2.6 Fail Recovery - UART boot

Due to the possible file/partition crash or abnormal actions in development process, sometimes developer will need to use “UART boot” method to boot up LX TI DM8148.

Note! Before start the UART boot procedure, you need to **add a jumper to JP1** port or short the two pins of JP1 for switching to UART boot mode. Besides, you also need to prepare an x-loader file for UART boot (i.e. u-boot.min.uart). About how to compile u-boot.min.uart, please refer to the link below:

http://processors.wiki.ti.com/index.php/DM814x_C6A814x_AM387x_PSP_U-Boot#U-Boot-MIN_for_UART_boot

Please following the steps listed below to use UART boot. Here we use Tera Term v4.71 as terminal emulator.

Step 1. Connect LX TI DM8148 debug port with PC and open Tera Term. Step 2. Load x-loader (u-boot.min.uart)

Step 2.1 Open “File” menu on Tera Term, and then choose Transfer→**XMODEM**→Send.

Step 2.2 Choose “1K” option and open u-boot.min.uart file.

Step 2.3 Plug in the adapter and then x-loader will be transferred to LX TI DM8148. (You will notice that LX TI DM8148 is booting up and shut down automatically. This action comes from LX TI DM8148 *AC LOST mechanism*.)

Step 3. Load x-loader **AGAIN**

Step 3.1 Repeat step 2.1~step 2.2 once.

Step 3.2 Press power button. LX TI DM8148 will load u-boot.min.uart from UART and then stop at x-loader.

Step 4. Load u-boot

Step 4.1 Type commands below to clear memory and load u-boot.

```
TI-MIN# mw.b 0x81000000 0xff 0x40000  
TI-MIN# loadb 0x81000000
```

Step 4.2 Open “File” menu on Tera Term, and then choose Transfer→**Kermit**→Send.

Step 4.3 Choose and open u-boot.bin file.

Step 4.4 After file transferring process finishes, type command below to start u-

boot. Remember to press any key to stop on u-boot when the Centaurus logo (*2) appears.

```
TI-MIN# go 0x81000000
```

Step 5. Load kernel: Here we introduce 3 ways for loading kernel. The 3 different methods are denoted as A, B, and C.

Step 5.A. Load kernel from **eMMC**

Step 5.A.1 Type the commands below to load kernel from eMMC.

```
TI8148_EVM# mmc init  
TI8148_EVM# fatload mmc 1 0x82000000 uImage
```

Step 5.B. Load kernel from **LAN**

Step 5.B.1 Type the commands below to load kernel from LAN. (Replace <TFTP Server IP Address> as your setting.)

```
TI8148_EVM# dhcp 0x82000000 <TFTP Server IP Address>:uImage
```

Step 5.C. Load kernel from **UART**

Step 5.C.1 Type the commands below to load kernel from UART.

```
TI8148_EVM# mw.b 0x82000000 0xff 0x400000  
TI8148_EVM# loadb 0x82000000
```

Step 5.C.2 Open “File” menu on Tera Term, and then choose Transfer→**Kermit**→Send.

Step 5.C.3 Choose and open uImage file to start loading. It will take about 10 minutes to complete.

Step 6. Use the commands below to start booting procedure.

```
TI8148_EVM# mmc init  
TI8148_EVM# setenv bootargs 'console=ttyO0,115200n8 serialtty=ttyO0  
root=/dev/mmcbk0p2 rw mem=352M mem=320M@0x9FC00000  
earlyprink notifyk.vpssm3_sva=0xBF900000 vmalloc=500M vram=102M  
printk.time=y noinitrd rootfstype=ext3 ip=off rootwait'  
TI8148_EVM# bootm 0x82000000
```

Note! In the future release of LX TI DM8148 sample, Chip PC will provide an Ubuntu rootfs

tarball file under the path “/usr/local/src/” on eM MC. The clients can backup it for the rootfs recovery need.

3. REFERENCE LINKS

1. TMS320DM8148
<http://www.ti.com/product/tms320dm8148>
2. EZSDK for DM814x/DM816x
<http://www.ti.com/tool/linuxezsdk-davinci>
3. EZSDK 5.03.01 Software Developers Guide
http://processors.wiki.ti.com/index.php/DM814x_EZ_5.03.01_Software_Developers_Guide
4. DM8148 e2e Forum
http://e2e.ti.com/support/dsp/davinci_digital_media_processors/f/716.aspx
5. TI Embedded Processors Wiki
http://processors.wiki.ti.com/index.php/Main_Page
6. DM814x Overview
http://processors.wiki.ti.com/index.php/DM814x_Overview
7. TI81xx PSP Porting Guide
http://processors.wiki.ti.com/index.php/TI81xx_PSP_Porting_Guide
8. DM814x AM387x PSP User Guide
http://processors.wiki.ti.com/index.php/DM814x_AM387x_PSP_User_Guide
9. DM814x AM387x PSP U-Boot
http://processors.wiki.ti.com/index.php/DM814x_C6A814x_AM387x_PSP_U-Boot
10. Understanding u-boot-min startup for DM814x
http://processors.wiki.ti.com/index.php/Understanding_u-boot-min_startup_for_DM814x
11. Modifying U-boot
http://processors.wiki.ti.com/index.php/Modifying_U-boot
12. DM814X AM387X VPSS Video Driver User Guide
http://processors.wiki.ti.com/index.php/DM814X_AM387X_VPSS_Video_Driver_User_Guide_PSP_04.01.00.06
13. SysLink UserGuide
http://processors.wiki.ti.com/index.php/SysLink_UserGuide