GSG: DA8x/OMAP-L1/AM1x DVEVM Additional Procedures

^ Up to main **Getting Started Guide for OMAP-L1** Table of Contents

Flashing images

DA850/OMAP-L138/AM18xx

On the OMAP-L138 (or AM18xx) SoC, the ARM boots first. On boot-up, the ARM runs the ARM UBL in AIS file format. The purpose of the ARM UBL is to initialize the PLLs, mDDR, and other hardware. Once done, it copies the U-Boot into mDDR and starts it.

U-Boot is an open source boot loader and is responsible for booting the Linux kernel.

Flashing images to SPI Flash

TIP

If U-Boot does not start upon powering on in the standard SPI boot mode (see Booting_the_OMAP-L138/AM18x_Out_of_the_Box), you must flash the SPI memory and restore the UBL and U-boot software. The simplest way accomplish this is to use the Serial flasher utility outlined in the first option below. A pre-compiled U-boot image is included in the SDK installation under the `/psp/prebuilt-images/` folder. The Serial flash utility only supports UBL images included with the utility or the PSP installation. If a self-compiled or custom UBL is needed, you must flash the SPI memory through CCS, outlined in the second step below.

There are two ways to flash images to SPI Flash:

- Serial flasher

1. See this page for instructions on using the command line serial flashing utility, which requires only a UART cable: Serial Boot and Flash Loading Utility for OMAP-L138

- CCS

1. The embedded emulation that comes with the EVM is XDS100 version 1, that does not support the ARM. So to connect to the ARM, an external emulator is necessary. Also, to use an external emulator, you need the full version of CCS, not the one that comes with the evm. If you do not have an external emulator, please use the Serial Flasher above.
2. Set the boot pins to emulation boot mode. This is done by setting switch S7 on the EVM according to the following table:
   For LogicPD EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON

For Spectrum Digital EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	ON	ON	ON

- For **03.20.xx.xx** Releases:

1. Start CCStudio and connect to the ARM. See details at: How to connect to the OMAP-L138/C6748 EVM board using CCS?. Ensure that you have the latest ARM GEL file from [[1]] for LogicPD EVM and [[2]]for Spectrum Digital EVM correctly specified.
2. Execute the GEL function "Full EVM-->SPI1_PINMUX" incase of LogicPD and "Test_setup-->SPI_setup" on Spectrum Digital. In case of CCSv3, this appears under the `GEL` menu. In case of CCSv4, this appears under `Scripts` menu.
3. Load the SPI flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/utils/omapl1x8/` directory of the PSP installation (`spiflash-writer.out`), or you can build your own by following the steps in the section on Rebuilding the SPI Flash writer
4. Run the SPI flasher program. You will be prompted for the input file type and the file path. For booting from SPI, an ARM AIS image and U-Boot are required.
    - To burn the ARM AIS image, for OMAP-L138/AM18xx, type `armais` as the image type. When prompted for a file name, provide the path to `arm-spi-ais.bin` file. A pre-built image is located in the `images/boot-strap/omapl1x8/` directory of the PSP installation.
    - To burn U-Boot, run the SPI flasher program again, and type `uboot` as the image type. When prompted for a file name, provide the path to the `u-boot.bin` file. A pre-built image is located in the `images/u-boot/omapl1x8/` directory of the PSP installation.

- For **03.21.xx.xx** Releases:

1. Start CCStudio v5 and connect to the ARM. Ensure that you have the latest ARM GEL file from [[1]] for LogicPD EVM and [[2]]for Spectrum Digital EVM correctly specified.
2. Execute the GEL function "Full EVM-->SPI1_PINMUX" incase of LogicPD and "Test_setup-->SPI_setup" on Spectrum Digital.This appears under the Scripts menu.
3. Load the SPI flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/boot-strap/ccs/omapl1x8/` directory of the PSP installation (`SPIWriter_OMAP-L138.out`), or you can build your own by following the steps in the section on Rebuilding the SPI Flash writer
4. Run the SPI flasher program. You will be prompted for the file path of ARM AIS image and U-Boot. For booting from SPI, an ARM AIS image and U-Boot are required.
    - First it prompts this message "Will you be writing UBL image ? (y or Y) ", Entering 'N' will skip the flashing of UBL. Y will allow to flash UBL.
    - To burn the ARM AIS image, for OMAP-L138/AM18xx . When prompted for a file name, provide the path to ubl_OMAPL138_SPI_MEM.bin file. A pre-built image is located in the images /boot-strap/serial_flash/omapl1x8/ directory of the PSP installation. Entering 'none' will skip this step.
    - To burn U-Boot. When prompted for a file name, provide the path to the u-boot.bin file. A pre-built image is located in the images/u-boot/omapl1x8/ directory of the PSP installation. Entering 'none' will skip this step.

7. Once the SPI flash has been written with the all the required files, disconnect CCStudio and power off the EVM. Proceed to boot from SPI flash

Flashing images to NAND Flash

Follow these steps to flash images to NAND Flash:

1. The embedded emulation that comes with the EVM is XDS100 version 1, that does not support the ARM. So to connect to the ARM, an external emulator is necessary. Also, to use an external emulator, you need the full version of CCS, not the one that comes with the evm. If you do not have an external emulator, you cannot load the NAND Flash using these instructions.

2. Set the boot pins to emulation boot mode. This is done by setting switch S7 on the EVM according to the following table:
   For LogicPD EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON

For Spectrum Digital EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	ON	ON	ON

- For **03.20.xx.xx** Releases:

1. Start CCStudio and connect to the ARM. See details at: How to connect to the OMAP-L138/C6748 EVM board using CCS?. Ensure that you have the latest ARM GEL file from LogicPD [3]/ Spectrum Digital [2] correctly specified.
2. Execute the GEL function "Full EVM-->EMIFA_NAND_PINMUX" for LogicPD and "Test_setup-->NAND_setup" for Spectrum Digital. In case of CCSv3, this appears under the `GEL` menu. In case of CCSv4, this appears under `Scripts` menu.
3. Load the NAND flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/utils/omapl1x8` directory of the PSP installation (`nand-writer.out`), or you can build your own by following the steps in the section on Rebuilding the NAND Flash writer
4. Run the NAND flasher program. You will be prompted for the input file type and the file path. For booting from NAND, an ARM AIS image and U-Boot are required.
   - To burn the ARM AIS image, for OMAP-L138/AM18xx, type `armais` as the image type. When prompted for a file name, provide the path to `arm-nand-ais.bin` file. A pre-built image is located in the `images/boot-strap/omapl1x8/` directory of the PSP installation.
   - To burn U-Boot, type `uboot` as the image type. When prompted for a file name, provide the path to the `u-boot.bin` file.

- For **03.21.xx.xx** Releases:

1. Start CCStudio v5 and connect to the ARM. Ensure that you have the latest ARM GEL file from LogicPD [3]/ Spectrum Digital [2] correctly specified.
2. Execute the GEL function "Full EVM-->EMIFA_NAND_PINMUX" for LogicPD and "Test_setup-->NAND_setup" for Spectrum Digital. This appears under `Scripts` menu.
3. Load the NAND flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/boot-strap/ccs/omapl1x8` directory of the PSP installation (`NANDWriter_OMAP-L138.out`), or you can build your own by following the steps in the section on Rebuilding the NAND Flash writer
4. Run the NAND flasher program. You will be prompted for the file path of ARM AIS and U-BOOT. For booting from NAND, an ARM AIS image and U-Boot are required.
   - CCS will prompt this message "Do you want to global erase NAND flash", Entering 'n' or 'N' will not erase the NAND flash.
   - To burn the ARM AIS image,for OMAP-L138/AM18xx. When prompted for a file name, provide the path to `ubl_OMAP-L138_NAND.bin` file. A pre-built image is located in the images/boot-strap/serial_flash/omapl1x8/ directory of the PSP installation. Entering 'none' will skip this step.

- To burn U-Boot, When prompted for a file name, provide the path to the u-boot.bin file. Entering 'none' will skip this step.

7. Once the NAND flash has been written with the all the required files, disconnect CCStudio and power off the EVM. Proceed to boot from NAND flash.

Flashing images to NOR Flash

Follow these steps to boot from NOR Flash:

1. The embedded emulation that comes with the EVM is XDS100 version 1, that does not support the ARM. So to connect to the ARM, an external emulator is necessary. Also, to use an external emulator, you need the full version of CCS, not the one that comes with the evm. If you do not have an external emulator, you cannot load the NOR Flash using these instructions.
2. Obtain the latest ARM GEL file from LogicPD [3]/ Spectrum Digital [2] . Run the CCStudio Setup tool and ensure that the ARM GEL file is correctly specified.
3. Set the boot pins to emulation boot mode. This is done by setting switch S7 on the EVM according to the following table:
   For Logic PD

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON

For Spectrum Digital

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	ON	ON	ON

- For **03.20.xx.xx** Releases:

1. Start CCStudio and connect to the ARM. See details at: How to connect to the OMAP-L138/C6748 EVM board using CCS?.
2. Execute the GEL function "Full EVM-->EMIFA_NOR_PINMUX" for Logic PD and "Test_setup-->NOR_setup" for Spectrum Digital. In case of CCSv3, this appears under the `GEL` menu. In case of CCSv4, this appears under `Scripts` menu.
3. Load the NOR flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/utils/omapl1x8` directory of the PSP installation (`norflash-writer.out`), or you can build your own by following the steps in the section on Rebuilding the NOR Flash writer.
4. Run the NOR flasher program. You will be prompted for the input file type and the file path. For booting from NOR, an ARM AIS image and U-Boot are required.
   - To burn the ARM AIS image, for OMAP-L138/AM18xx, type `armais` as the image type. When prompted for a file name, provide the path to `arm-nor-ais.bin` file. A pre-built image is located in the `images/boot-strap/omapl1x8/` directory of the PSP installation.
   - To burn U-Boot, type `uboot` as the image type. When prompted for a file name, provide the path to the `u-boot.bin` file built for NOR.

- For **03.21.xx.xx** Releases:

1. Start CCStudio v5 and connect to the ARM.
2. Execute the GEL function "Full EVM-->EMIFA_NOR_PINMUX" for Logic PD and "Test_setup-->NOR_setup" for Spectrum Digital. This appears under `Scripts` menu.
3. Load the NOR flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/boot-strap/ccs/omapl1x8` directory of the PSP installation

(`NORWriter_OMAP-L138.out`), or you can build your own by following the steps in the section on Rebuilding the NOR Flash writer.

4. Run the NOR flasher program. You will be prompted for the input file paths. For booting from NOR, an ARM AIS image and U-Boot are required.
   - To burn the ARM AIS image, for OMAP-L138/AM18xx. When prompted for a file name, provide the path to ubl_OMAP-L138_NOR.bin file. A pre-built image is located in the `images/boot-strap/serial_flash/omapl1x8/` directory of the PSP installation. Entering 'none' will skip this step.
   - To burn U-Boot, when prompted for a file name, provide the path to the `u-boot.bin` file built for NOR. Entering 'none' will skip this step.

8. Once the NOR flash has been written with the all the required files, disconnect CCStudio and power off the EVM. Proceed to boot from NOR flash

DA830/OMAP-L137

On the OMAP-L137 and DA830 SoCs, the DSP boots first, on boot-up, the DSP runs the DSP UBL in AIS file format. The purpose of the DSP UBL is to wake up the ARM and start the ARM UBL which is present in raw binary form. The purpose of the ARM UBL is to initialize the PLLs, SDRAM, and other hardware. Once done, it copies the U-Boot into SDRAM and starts it. U-Boot is an open source boot loader and is responsible for booting the Linux kernel.

Flashing images to SPI Flash

There are two ways to flash images to SPI Flash:

- Serial flasher

1. See this page for instructions on using the command line serial flashing utility, which requires only a UART cable: Serial Boot and Flash Loading Utility for OMAP-L137

- CCS

1. Setup the EVM in "emulation debug" mode by setting SW2 switch as follows:

For EVM revisions A and B:

Pin #	7	2	1	0	3
Position	1	1	1	1	1

For EVM revisions after B:

Pin #	7	2	1	0	3
Position	1	1	1	1	0

2. Start CCStudio and connect to the DSP. Once done, connect to the ARM. Ensure that you have the latest DSP GEL file from Spectrum Digital Support site [4] correctly specified.
3. Load the SPI flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/utils/omapl1x7/` directory of the PSP installation (`spiflash_writer.out`), or you can build your own by following the steps in the section on Rebuilding the SPI Flash writer
4. Run the SPI flasher program. You will be prompted for the input file type and the file path. Three files are required: DSP AIS, ARM UBL and U-Boot.

- To burn the DSP AIS image, type `dspais` as the image type. When prompted for a file name, provide the path to `dsp-spi-ais.bin` file. A pre-built image is located in the `images/boot-strap/omapl1x7/` directory of the PSP installation.
- To burn the ARM UBL image, type `armubl` as the image type. When prompted for a file name, provide the path to `ubl-spi.bin` file. A pre-built image is located in the `images/boot-strap/omapl1x7/` directory of the PSP installation.
- To burn U-Boot, run the SPI flasher program again, and type `uboot` as the image type. When prompted for a file name, provide the path to the `u-boot.bin` file. A pre-built image is located in the `images/u-boot/omapl1x7/` directory of the PSP installation.

5. Once the SPI flash has been written with the all the required files, disconnect CCStudio and power off the EVM. Proceed to boot from SPI flash.

Flashing images to NAND Flash

1. Setup the EVM in "emulation debug" mode by setting SW2 switch on base board as follows:

For EVM revisions A and B:

Pin #	7	2	1	0	3
Position	1	1	1	1	1

For EVM revisions after B:

Pin #	7	2	1	0	3
Position	1	1	1	1	0

2. On the User Interface card, set the SW1 switch as follows:

Pin #	1	2	3	4
Position	1	0	1	1

3. Start CCStudio and connect to the DSP. Once done, connect to the ARM. Ensure that you have the latest DSP GEL file from Spectrum Digital Support site [4] correctly specified.
4. Execute the GEL function `Setup_EMIFA_PinMux()`. In case of CCSv3, this appears under the `GEL` menu. In case of CCSv4, this appears under `Scripts` menu.
5. Load the NAND flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/utils/omapl1x7/` directory of the PSP installation (`nand_writer.out`), or you can build your own by following the steps in the section on Rebuilding the SPI Flash writer
6. Run the NAND flasher program. You will be prompted for the input file type and the file path. Three files are required: DSP AIS, ARM UBL and U-Boot.
   - To burn the DSP AIS image, type `dspais` as the image type. When prompted for a file name, provide the path to `dsp-nand-ais.bin` file. A pre-built image is located in the `images/boot-strap/omapl1x7/` directory of the PSP installation.
   - To burn the ARM UBL image, type `armubl` as the image type. When prompted for a file name, provide the path to `ubl-nand.bin` file. A pre-built image is located in the `images/boot-strap/omapl1x7/` directory of the PSP installation.
   - To burn U-Boot, run the SPI flasher program again, and type `uboot` as the image type. When prompted for a file name, provide the path to the `u-boot.bin` file. A pre-built image is located in the `images/u-boot/omapl1x7/` directory of the PSP installation.

7. Once the NAND flash has been written with the all the required files, disconnect CCStudio and power off the EVM. Proceed to boot from NAND flash

AM17xx

On the AM17xx SoC, the ARM boots first, on boot-up, the ARM runs the ARM UBL in AIS file format. The purpose of the ARM UBL is to initialize the PLLs, SDRAM, and other hardware. Once done, it copies the U-Boot into SDRAM and starts it. U-Boot is an open source boot loader and is responsible for booting the Linux kernel.

Flashing images to SPI Flash

1. Setup the EVM in "emulation debug" mode by setting SW2 switch as follows:

Pin #	**7**	**2**	**1**	**0**	**3**
Position	1	1	1	1	0

2. Start CCStudio and connect to the ARM. Ensure that you have the latest ARM GEL file from Spectrum Digital Support site [5] correctly specified.
3. Load the SPI flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/utils/omapl1x7/` directory of the PSP installation (`spiflash_writer.out`), or you can build your own by following the steps in the section on Rebuilding the SPI Flash writer
4. Run the SPI flasher program. You will be prompted for the input file type and the file path. Two files are required: ARM AIS and U-Boot.
   - To burn the ARM AIS image, type `armais` as the image type. When prompted for a file name, provide the path to `arm-spi-ais.bin` file. A pre-built image is located in the `images/boot-strap/am17xx/` directory of the PSP installation.
   - To burn U-Boot, run the SPI flasher program again, and type `uboot` as the image type. When prompted for a file name, provide the path to the `u-boot.bin` file. A pre-built image is located in the `images/u-boot/omapl1x7/` directory of the PSP installation.
5. Once the SPI flash has been written with the all the required files, disconnect CCStudio and power off the EVM. Proceed to boot from SPI flash

Flashing images to NAND Flash

1. Setup the EVM in "emulation debug" mode by setting SW2 switch on the base board as follows:

Pin #	**7**	**2**	**1**	**0**	**3**
Position	1	1	1	1	0

2. On the User Interface card, set the SW1 switch as follows:

Pin #	**1**	**2**	**3**	**4**
Position	1	0	1	1

3. Start CCStudio and connect to the ARM. Ensure that you have the latest ARM GEL file from Spectrum Digital Support site [5] correctly specified.
4. Execute the GEL function `Setup_EMIFA_PinMux()`. In case of CCSv3, this appears under the `GEL` menu. In case of CCSv4, this appears under `Scripts` menu.
5. Load the NAND flasher tool on to the ARM. You can either use the pre-built binary shipped in the `images/utils/omapl1x7/` directory of the PSP installation (`nand_writer.out`), or you can build your own by following the steps in the section on Rebuilding the SPI Flash writer

6. Run the NAND flasher program. You will be prompted for the input file type and the file path. Two files are required: ARM AIS and U-Boot.
    - To burn the ARM AIS image, type `armais` as the image type. When prompted for a file name, provide the path to `arm-nand-ais.bin` file. A pre-built image is located in the `images/boot-strap/am17xx/` directory of the PSP installation.
    - To burn U-Boot, run the NAND flasher program again, and type `uboot` as the image type. When prompted for a file name, provide the path to the `u-boot.bin` file.
7. Once the NAND flash has been written with the all the required files, disconnect CCStudio and power off the EVM. Proceed to boot from NAND flash

Flashing Boot Images on Linux Without CCS

This procedure mirrors that found for the Hawkboard [6]. It requires the Serial boot and Flash Package available here [7] and a compiled u-boot binary. The AIS boot image that is generated makes use of built-in ROM functions and does not use a UBL for booting.

Note: The following tools require the Mono Framework to run on a Linux system. Refer to your Linux distribution's package management instructions for information on how to check if Mono is installed or to install/update it.

OMAP-L138

1. Prepare a UART AIS boot image using the command-line HexAIS_OMAP-L138.exe and the attached INI file: . Note that there are different versions of the INI files for the D800K002 ROM, as that ROM revision (which was part of the first silicon revision) does not correctly handle mDDR initialization. Also, each file has mDDR configuration setup for the either 132MHz or 150MHz operation. Comment out the one you do not want to use and un-comment the one you do intend to use. Alternatively, you can use the AISGen GUI tool to perform this step.

```
host$ mono ./HexAIS_OMAP-L138.exe -ini OMAP-L138_EVM_uart.ini -o u-boot_uart.ais
```

3. Prepare a SPI AIS boot image using the command-line HexAIS_OMAP-L138.exe and the attached INI file. Alternatively, you can use the AISGen GUI tool to perform this step. This file should be placed in the /tftproot directory of your TFTP server so that it can be transferred to the EVM under u-boot using the Ethernet port.

```
host$ mono ./HexAIS_OMAP-L138.exe -ini OMAP-L138_EVM_spi.ini -o u-boot_spi.ais
```

5. Set the EVM to boot in UART2 boot mode. This is done by setting switch S7 on the EVM according to the following table:

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON

6. Start the command-line slh_OMAP-L138.exe tool to boot the board from your host PC.

```
host$ mono ../slh_OMAP-L138.exe -waitForDevice -v AISUtils/u-boot_uart_L138.ais
```

8. Power up the EVM. The UART boot process will start. It may take as long as 15 seconds to complete.

```
-----------------------------------------------------
   TI Serial Loader Host Program for OMAP-L138
   (C) 2010, Texas Instruments, Inc.
   Ver. 1.65
-----------------------------------------------------
```

```
Platform is Windows.
Attempting to connect to device COM1...
Press any key to end this program at any time.

Entering AIS Parser

Waiting for the OMAP-L138...
(AIS Parse): Read magic word 0x41504954.
(AIS Parse): Waiting for BOOTME... (power on or reset target now)
(AIS Parse): BOOTME received!
(AIS Parse): Performing Start-Word Sync...
(AIS Parse): Performing Ping Opcode Sync...
(AIS Parse): Processing command 0: 0x5853590D.
(AIS Parse): Performing Opcode Sync...
(AIS Parse): Executing function...
(AIS Parse): Processing command 1: 0x5853590D.
(AIS Parse): Performing Opcode Sync...
(AIS Parse): Executing function...
(AIS Parse): Processing command 2: 0x58535901.
(AIS Parse): Performing Opcode Sync...
(AIS Parse): Loading section...
(AIS Parse): Loaded 156384-Byte section to address 0xC1080000.
(AIS Parse): Processing command 3: 0x58535906.
(AIS Parse): Performing Opcode Sync...
(AIS Parse): Performing jump and close...
(AIS Parse): AIS complete. Jump to address 0xC1080000.
(AIS Parse): Waiting for DONE...
(AIS Parse): Boot completed successfully.

Operation completed successfully.
```

10. When boot is completed, start a serial terminal program (Hyperterminal, minicom) to connect to the EVM. Press "enter" to see the u-boot prompt.
11. Erase first 256 KB of the serial flash

```
U-Boot > sf probe 0
U-Boot > sf erase 0 40000
```

13. Setup the u-boot enviroment for TFTP download.

```
U-Boot > setenv serverip 172.24.156.199
U-Boot > dhcp
```

15. Use TFTP to transfer the SPI u-boot image generated earlier.

```
U-Boot > tftpboot 0xc0700000 u-boot_spi.ais
```

17. Write the SPI AIS u-boot image to the start of the SPI flash.

```
U-Boot > sf write c0700000 0 40000
```

19. Power off the EVM, and change the boot mode switches to SPI0 master boot. This is done by setting switch S7 on the EVM according to the following table:

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

20. Power on the EVM and the u-boot should start.

Booting U-Boot

U-Boot is an open source boot loader and is responsible for booting the Linux kernel.

Connect a serial cable from the serial port on the EVM to the COM port on the host machine. Set up the serial terminal software as described in Booting the EVM out of the box.

Note: Boot images may not have been pre-flashed on the EVM for all boot modes. In this case, follow the procedures in Flashing images to flash the required boot images.

DA850/OMAP-L138/AM18xx

Booting from SPI Flash

In order to boot from SPI flash, which has been written with the boot images, set the SW7 switch on the base board as follows:

For LogicPD EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

For Spectrum Digital EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF

Booting from NAND Flash

In order to boot from NAND flash, which has been written with the boot images, set the SW7 switch on the base board as follows:

For LogicPD EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF

For Spectrum Digital EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	ON	ON	OFF

Booting from NOR Flash

In order to boot from NOR flash, which has been written with the boot images, set the SW7 switch on the base board as follows:
For LogicPD EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	ON	ON	OFF

For Spectrum Digital EVM

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF

DA830/OMAP-L137/AM17xx

Booting from SPI Flash

- Set the SW2 switch on the DSK board as follows. (X indicates the setting is 'don't care')

Pin#	**7**	**2**	**1**	**0**	**3**
Position	0	1	0	1	X

Booting from NAND Flash

- Set the SW2 switch on the DSK board as follows. (X indicates the setting is 'don't care')

Pin#	**7**	**2**	**1**	**0**	**3**
Position	0	1	1	1	X

On the User Interface card, set the SW1 switch as follows:

Pin #	**1**	**2**	**3**	**4**
Position	1	0	1	1

Booting the Linux kernel using U-Boot

Booting the kernel requires a valid kernel image (`uImage`) and a target filesystem. A pre-built kernel image is included in the `images/kernel` directory of the PSP installation.

Booting Linux kernel using U-Boot describes various methods to boot Linux kernel.

Using extended memory available on OMAP-L138 EVM board

The OMAP-L138 eXperimenter board has 64MB of memory and the OMAP-L138 EVM has 128 MB available. Out of this 128 MB, some amount is used for Linux and the rest can be used for DSP or as DSP/ARM shared memory. Sometimes it is convenient to allow Linux to use discontinuous blocks of memory. Kernel parameter `mem=` can be used for this purpose. One example is shown here:

```
U-Boot> setenv bootargs console=ttyS2,115200n8 root=/dev/ram0 rw initrd=0xc1180000,4M ip=dhcp mem=32M@0xc0000000 mem=64M@0xc4000000
```

If you plan on using Codec Engine (or DSPLink and CMEM alone), you need to make sure that the insmod command for the cmemk.ko kernel module uses the "allowOverlap=1" option (see the CMEM documentation available in the LinuxUtils package). Failure to use this option will result in an error message, since the kernel still reports its memory usage to CMEM as a contiguous 96M range.

Creating bootable SD card for OMAP-L138 EVM board

The UBL on OMAP-L138 expects the u-boot descriptor to be present in sectors 1 to 25 of the SD card. Sector 0 is used for storing DOS partition information. Hence the u-boot descriptor is stored from sectors 1 till 24. The descriptor is 512 bytes in size and is replicated in each of these sectors. U-Boot binary starts at Sector 117.

The SD card shall be re-partitioned and formated to create some room for storing u-boot. Use fdisk utility (as super user) to delete the existing partition and create a new one.

```
host$ fdisk /dev/mmcblk0 (device name might change if USB card reader is used)
```

- Delete the existing partitions with 'd' command.
- Create a new partition with 'n' command, followed by 'p' command
- Mark the first cylinder as 20. Typical cylinder size is 32KBytes. So starting the first cylinder at 20 provides us about 600Kbytes for storing UBL and u-boot. If the fdisk utility displays a different cylinder size, make sure that you are leaving atleast 500K space before the first cylinder.
- Leave the last cylinder to default value (or) any other value depending on the partition size requirements.
- Save and exit with 'w' command

Using the **uflash** utility, place the u-boot binary on the SD card. Copy the u-boot.bin to uflash directory,

```
host$ ./uflash -d /dev/mmcblk0 -b u-boot.bin -p OMAPL138 -vv
        u-boot Size 252087
        U-Boot Magic Number     : a1aced66
        U-Boot Entry Point      : c1080000
        U-Boot Number of Blocks : 000001ec
        U-Boot Starting Block   : 00000075
        Load U-Boot Address     : c1080000
        Writing U-Boot Signature
        Writing U-Boot
        Done...
```

SD Card Writer Utility (uflash)

This is a Linux command line tool specific to TI's Davinci platforms, for flashing UBL/U-Boot to SD card.

Building uflash

1. Use your Linux host to extract the uflash source code from the `src/utils/mmcsd-writer-#.#.#.#.tar.gz` tarball from the OMAP-L138 Linux PSP package, which is located in the DaVinci-PSP-SDK-#.#.#.# directory under the main SDK installation directory. Use the `tar` command to extract the sources.
2. Use the native Linux host gcc compiler to build uflash

```
host$ gcc uflash.c -o uflash
```

Loading Linux Kernel Modules

Many of the kernel features can be built as run-time loadable modules so that they are not part of the kernel image, but can be inserted into the kernel at run-time to increase the running kernel's functionality.

To understand how to configure some features as modules and how to build them, refer to Rebuilding the Linux Kernel and Driver configuration in the Linux kernel.

Pre-built binaries for features configured by default as kernel modules are included in the PSP package in the `images/kernel/omapl1x7/modules/lib` directory for OMAP-L137 and in the `images/kernel/omapl1x8/modules/lib` directory for OMAP-L138. To use these modules, copy the contents of this directory to the /lib directory of your root file system.

On the target Linux command prompt use command `modprobe` command to load the module. `rmmod` command can be used to remove the module. Below is an example of loading the USB file storage gadget module. Similar steps can be followed for any driver module.

Loading USB 2.0

To load the USB file backed storage gadget:

1. Insert the g_file_storage.ko module with the following command where `/dev/blockdevX` is the storage device acting as the actual storage space. Replace it with the path to the actual block device name acting as physical storage, such as a MMC/SD card.

```
target$ modprobe g_file_storage file=/dev/blockdevX
```

3. Remove the module by using

```
target$ rmmod g_file_storage
```

DSP wakeup in U-Boot

On the OMAP-L138 SoC, the U-Boot wakes up the DSP by default. The DSP reset vector is set to `0x80000000`, and after wakeup it executes the `idle` instruction. To prevent the DSP from being woken up by U-Boot, you can do the following:

1. Set the `dspwake` environment variable to "no".
2. Save the environment space.
3. Reset the board.

This feature has been added to help DSP users wake up the DSP quickly, since on the OMAP-L138 EVM only the on-board emulation can access the DSP.

Modifying SPI Frequency in U-Boot

On DA850/OMAP-L138/AM18xx EVMs, in U-Boot, SPI flash has been configured to work at 30MHz. But in some situations one may have to modify the SPI frequency. For example:

- When operating at different voltages.
- When customer has a different SPI flash which operates at different frequency.

In such cases, SPI frequency can be changed in U-Boot by modifying the `CONFIG_SF_DEFAULT_SPEED` variable in `u-boot/include/configs/da850evm.h` file. After this modification, **rebuild u-boot** and **flash the new u-boot** binary file to SPI flash.

Restoring factory default U-Boot environment variables

To restore factory default U-Boot environment variables, the existing environment variables need to be erased from flash. You will need to know the offset in flash were environment variables are stored and the size of flash dedicated to storing the environment variables.

These values are represented by `CONFIG_ENV_OFFSET` and `CONFIG_ENV_SIZE` respectively in U-Boot EVM configuration file.

For OMAP-L138 (or DA850, AM18xx) SoCs, this file is `include/configs/da850_evm.h` inside the U-Boot source tree.
For OMAP-L137 (or DA830, AM17xx) SoCs, this file is `include/configs/da830_evm.h` inside the U-Boot source tree.

Follow this procedure based on the flash U-Boot stores its environment variables on. The size and offset values given below are to considered representative only.

SPI flash

On SPI flash, environment variables are stored at offset 256 KByes into the flash and environment variables sector size is 64 KBytes.

1. On the U-Boot command prompt:

```
U-Boot> sf probe 0
```

3. Erase environment sector length bytes (64 KBytes) from environment sector offset (256 KBytes) as follows:

```
U-Boot> sf erase 40000 10000
```

5. Reset the board and it will come up with default environment variables.
6. Save the new set of environment variables.

```
U-Boot> saveenv
```

NAND Flash

On the NAND flash, env variables are stored at offset 0 (zero) and environment variables sector size is 128 KBytes.

1. Erase environment sector length bytes (128 KBytes) from environment sector offset 0 (zero) as follows:

```
U-Boot> nand erase 0 20000
```

3. Reset the board and it will come up with default environment variables.
4. Save the new set of environment variables.

```
U-Boot> saveenv
```

NOR Flash

On the NOR flash, env variables are stored at the 4th sector of NOR flash and environment variables sector size is 128 KBytes.

1. Unprotect the 4th sector using:

```
U-Boot> protect off 60060000 +20000
```

3. Erase environment sector length bytes (128 KBytes) from environment sector offset 60060000 as follows:

```
U-Boot> erase 60060000 +20000
```

5. Reset the board and it will come up with default environment variables.
6. Save the new set of environment variables.

```
U-Boot> saveenv
```

Restoring MAC address on SPI Flash

On DA850/OMAP-L138/AM18x EVMs, Logic-PD would be pre-flashing the MAC address to the SPI flash. If SPI flash is completely erased then even the MAC address gets erased. MAC address can be written to SPI flash from U-Boot. The following commands need to be entered from U-Boot prompt.

1. Modify the DDR content at address c0000000. MAC address is printed on a sticker pasted on SoC module of EVM.

```
U-Boot> mm.b c0000000
c0000000: fb ? 00
c0000001: ef ? 08
c0000002: fd ? ee
c0000003: df ? 03
c0000004: ff ? 6a
c0000005: ff ? c4
c0000006: bf ? q
```

3. Make sure that modification is correct:

```
U-Boot> md.b c0000000
c0000000: 00 08 ee 03 6a c4 bf ff f3 fd fb ff fe fd bf ef    ....j...........
c0000010: ff ff fb fd 3f fd ff ff bb df 7f fd f7 ff 77 ff    ....?.........w.
c0000020: 7f ee f7 fd 3f fd fb fd ed ef bf ce 77 bf fb ff    ....?.......w...
c0000030: 6f bc b7 fe ff de fd ff 7b fd ff ff b7 fd ff ff    o.......{.......
```

5. Initialize SPI flash from u-boot.

```
U-Boot> sf probe 0
```

7. Write the MAC address from DDR to SPI flash:

```
U-Boot> sf write c0000000 7f0000 6
```

9. Reset the EVM. U-Boot/Kernel should read the MAC address which was written to the flash.

Enabling Write-Back Cache on DA830/OMAP-L137

On the DA830 and OMAP-L137 SoCs silicon revisions 1.1 and earlier, the ARM cache in write-back mode is not functional. This has been resolved starting silicon revision 2.0. To maintain backward compatibility with affected silicon revisions, the Linux kernel keeps write-back mode disabled in the default configuration. Users of silicon revisions 2.0 and higher can enable write-back cache from kernel configuration.

```
System Type  --->
    [ ] Force write through D-cache
```

Message logging on UART in ARM UBL

One may wish to enable messaging logging on the UART console. This feature can be enabled by supplying DEVICE_UART<n>_FOR_DEBUG, where <n> is the UART number. Presently UART0, UART1 and UART2 can be enabled for logging. However, UART<n> may be available for logging depending on the PINMUXing with core peripheral like NAND/SPI/NOR etc and the BOOT mode one wishes to have. Please check this and the settings in device.c/UARTInit()

For example if one wishes to send the console debug messages over UART2 the compiler flag needs to supplied as below (shown for NAND boot mode):

```
Options=-g -o3 -fr"$(Proj_dir)\..\nand" -fs"$(Proj_dir)\..\nand"
-i"$(Proj_dir)\..\include" -d"UBL_NAND" -d"USE_IN_ROM"
-d"DEVICE_UART2_FOR_DEBUG" -me -mv5e --abi=ti_arm9_abi
```

Linux Functional Test Bench

The Linux Functional Test Bench (LFTB) is the set of tools used to verify the various driver features. The test bench is included in the `test-suite` directory of the PSP installation. Use the `tar` command to extract the LFTB package.

Information on how to use LFTB and its various features is included in the LFTB package itself.

What's next?

Please continue on to the **Additional information** section of the OMAP-L1 Getting Started Guide.

References

[1] http://www.logicpd.com/products/development-kits/zoom-omap-l138-experimenter-kit|LogicPD

[2] http://support.spectrumdigital.com/boards/evmam1808/revb/

[3] http://www.logicpd.com/products/development-kits/zoom-omap-l138-experimenter-kit

[4] http://support.spectrumdigital.com/boards/evmomapl137/revd/

[5] http://support.spectrumdigital.com/boards/evmam1707

[6] http://elinux.org/Hawkboard#Booting

[7] http://sourceforge.net/projects/dvflashutils/files/OMAP-L138/v2.11/OMAP-L138_FlashAndBootUtils_2_11.tar.gz/download

Article Sources and Contributors

GSG: DA8x/OMAP-L1/AM1x DVEVM Additional Procedures *Source*: http://processors.wiki.ti.com/index.php?oldid=63440 *Contributors*: Akshay.s, Binary0101, D-allred, Dswag89, Jeff Cobb, Mariana, SekharNori, Sudhakar.raj, Sugumar, Technocratic, Xyvonned

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.
BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

License

1. Definitions

a. "**Adaptation**" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
b. "**Collection**" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.
c. "**Creative Commons Compatible License**" means a license that is listed at http:/ / creativecommons. org/ compatiblelicenses that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
d. "**Distribute**" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
e. "**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
f. "**Licensor**" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
g. "**Original Author**" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
h. "**Work**" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
i. "**You**" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
j. "**Publicly Perform**" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
k. "**Reproduce**" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

   i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
   ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
   iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.
b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.
c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Ssection 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.