Serial Boot and Flash Loading Utility for OMAP-L137

Contents

TI Flash and Boot Utilities

The following software supports the OMAP-L137, C6747, and AM1707 families of devices.

This package contains a set of utilities running from the command-line on Windows for flashing the various flash types on the the OMAP-L137 EVM via the serial port.

The Serial Flashing Host Utility (which offers the same functionality as a previous program called DVFlasher) executable is called sfh_OMAP-L137.exe.

These programs each encapsulate a distinct binary UBL which is transferred via the UART. This implies that the chip must be operating in the UART boot mode, showing the BOOTME prompt.

NOTE: The assumption is made that the UART of the device operates at 115200, 8N1. If the oscillator used with the device does not match the one used on the EVM (e.g. on a custom platform, or OMAPL137_v1), the baud rate assumption may be incorrect.

Obtaining the software

The latest software can be download from here (http://sourceforge.net/projects/dvflashutils/files/OMAP-L137/).

Compiling

A makefile is included for compiling the host and target parts of each utility. The target portions are built first since they are embedded into the the host executables. These target portions are loaded to the OMAP-L137 device via the UART boot mode.

Note that the pre-built executables have been tested to work on the EVMs. It is not necessary to rebuild them unless changes need to be made for a custom board.

Under Windows

Please see the following page for detailed instructions for rebuilding the utilities in included in this package:

http://processors.wiki.ti.com/index.php/Rebuilding_the_Flash_and_Boot_Utils_Package (http://processors.wiki.ti.com/index.php/Rebuilding_the_Flash_and_Boot_Utils_Package)

Under Linux

The Mono Framework must be installed and in the path. RPMs are available at 'http://www.mono-project.com/Downloads'. Then go to the GNU directory of the package and run:

```
make
```

The above instructions assume the that GNU ARM cross-compiler tools (arm-none-linux-gnueabi-gcc, etc.) are in the current PATH.

The C6x Compiler Tools are also required to build some components. These are available free of charge at https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/download.htm. You will need to add the bin directory to your PATH and to set the environment variables as directed by the installer.

Running

Under Windows This utility can be run from the command line under Windows with the .Net Framework 2.0 or later installed through the following steps:

1. Set the boot pins of the EVM to UART2 boot mode. This is done by setting boot switches on the OMAP-L137 EVM according to the following table:
   For all revisions after B:

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

For revisions A and B:

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

2. Connect a serial cable from the host computer to the OMAP-L137 EVM. Note that for OMAPL137_v1 the host computer must be able to run at a UART baud rate of 230400.
3. Test your UART connection by opening HyperTerminal and connecting to the correct COM port at 115200 (or 230400). When the reset button is pressed on the EVM, the words "BOOTME" should appear on the terminal. If not, recheck the cable connection, baud settings, and boot pins of the EVM. Please note that many PC serial ports do not support a baud rate of 230400. In these cases, a USB->Serial adapter should be used.
4. Disconnect from HyperTerminal.
5. Open a command prompt on the host computer and change directory to where the flashing utility was saved (ex. c:\flasher).
6. Run the flashing utility with this command:
   - **C:\flasher>sfh_OMAP-L137.exe [options]**
7. Reset the board

Under Linux

The serial flasher can be run on a Linux machine with the latest open-source Mono Framework installed. The steps are identical to the Windows environment except the command:

- **mono ./sfh_OMAP-L137.exe [options]**

Serial Flasher Options

There are four modes for using the serial flasher:

1. Erase the target flash type - This will erase the entire contents of the flash.
   - **C:\flasher>sfh_OMAP-L137.exe -erase**
2. Flash the memory with a single application image - This will place an application image at address 0x0 of the flash.
   - **C:\flasher>sfh_OMAP-L137.exe -flash_noubl <binary application file>**
3. Flash the memory with a UBL and application image - This will place the UBL at address 0x0 and an application image, such as u-boot, at address 0x10000. This is used for the AM1707 device.
   - **C:\flasher>sfh_OMAP-L137.exe -flash <UBL binary file> <binary application file>**
4. Flash the memory with a DSP UBL, ARM UBL, and application image - This will place a DSP AIS file at address 0x0 of the flash, an ARM UBL at address 0x2000, and an application image, such as u-boot, at address 0x8000. This is used for the OMAPL137_v1 and OMAPL137_v2 devices.
   - **C:\flasher>sfh_OMAP-L137.exe -flash_dsp <DSP UBL AIS file> <ARM UBL binary file> <binary application file>**

Additional options are shown below:

```
-targetType         : Specifies exact target type within OMAP-L137 family (OMAPL137_v2 [default], OMAPL137_v1, C6747, or AM1707)
-flashType          : Specifies exact flash type (default SPI_MEM)
-p <COM PORT NAME>  : Allows specifying com port other than default 'COM1' or '/dev/ttyS0'.
-h                  : Show help text.
-v                  : See verbose output from OMAP-L137.
-baud <BAUD RATE>   : Allows specifying baud rate other than default (115200)
-APPStartAddr       : Changes entry point of application (default 0xC1080000)
-APPLoadAddr        : Changes load address of application (default 0xC1080000)
```

Once any command is run, the "Waiting for BOOTME..." prompt shows. Power cycle the board or press the reset button to continue.

Please note, for the AM1707, the ARM UBL must come from the /gnu/ubl/ directory of this release (i.e. ubl_AM1707_SPI_MEM.bin). The serial flasher is not compatible with the UBLs found in the PSP release

For the OMAPL137_v1 and OMAPL137_v2, the DSP and ARM UBLs should come from the PSP release directory.

Restoring the OMAP-L137 EVM SPI Flash

If the contents of the OMAP-L137 SPI flash become corrupted, they can be restored through the following steps:

1. Set the boot pins to UART2 boot mode. This is done by setting the boot switches on the OMAP-L137 EVM according to the following table:

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

2. Obtain the DSP UBL AIS file (dsp-spi-ais.bin), the ARM UBL binary file (ubl-spi.bin), and the u-boot binary file (u-boot.bin).
   - **Pre-built images are located in the "images" directory of the PSP installation**
3. Run the following command to flash the UBL and u-boot binary files:
   - **C:\flasher>sfh_OMAP-L137.exe -spiflash_dsp <DSP UBL AIS file> <ARM UBL binary file> <u-boot binary file>**
4. Reset the board
5. After the UBL and u-boot files have been flashed, set the boot pins to SPI0 boot mode and reboot the board. This is done by setting the boot switches on the OMAP-L137 EVM according to the following table:

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

6. One u-boot has loaded, follow the instructions in the Getting Started Guide to copy the Linux kernel and filesystem to the SPI flash.
   - **Updating the Kernel Image**

Modifications for Custom Boards

The default settings used by the tools apply only to the EVMs. For custom boards, changes will most likely be required, and the tools must be rebuilt. The custom changes should be made to the files in the OMAP-L137/Common/src and OMAP-L137/Common/include directories. Common changes include:

- SDRAM Configuration
  - device.c: Modify the parameters passed to the function DEVICE_ExternalMemInit to match the SDRAM timing requirements for the custom board
- UART Settings
  - device_uart.h: Modify the #define DEVICE_UART_PERIPHNUM to set which UART is connected to the host PC.
  - Note that flow control is not used, so only the RX and TX lines need to be connected for boot and flashing purposes.
- SPI Settings
  - device_spi.h: Modify the global macros to select the appropriate peripheral and chip select numbers.
- PLL Settings (for non 24 MHz input clocks)
  - device.c: Modify the parameters passed into the function DEVICE_PLL0Init and DEVICE_PLL1Init to set the proper PLL output frequencies
  - device_uart.c: Modify the 'divider' field in DEVICE_UART_config to achieve 115200 baud rate

FAQ

Is there a size limit to the image I can flash?

No, as long as it can fit into external SDRAM there is no limit. Keep in mind that flashing large images at 115200 baud will take a very long time

Can the tool flash other boot media besides those listed above?

By default, no. The code would have to be modified and rebuilt to support other boot media.

Can I use this tool if I'm using something other than a 24 MHz input clock?

Unfortunately no. The ROM bootloader assume a 24 MHz input clock and sets up the internal dividers accordingly.

License

These tools are provided as freeware, with no guarantee.

{{ 1. switchcategory:MultiCore= - For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum - For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum Please post only comments related to the article **Serial Boot and Flash Loading Utility for OMAP-L137** here.	Keystone= - For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum - For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum Please post only comments related to the article **Serial Boot and Flash Loading Utility for OMAP-L137** here.	C2000=*For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article* ***Serial Boot and Flash Loading Utility for OMAP-L137*** *here.*	DaVinci=*For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article* ***Serial Boot and Flash Loading Utility for OMAP-L137*** *here.*	MSP430=*For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article* ***Serial Boot and Flash Loading Utility for OMAP-L137*** *here.*	OMAP35x=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article* ***Serial Boot and Flash Loading Utility for OMAP-L137*** *here.*	OMAPL1=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article* ***Serial Boot and Flash Loading Utility for OMAP-L137*** *here.*	MAVRK=*For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article* ***Serial Boot and Flash Loading Utility for OMAP-L137*** *here.*	*For technical su please post you questions at http://e2e.ti.cor Please post on comments abo article* ***Serial B and Flash Loa Utility for OMA L137*** *here.* }}

Links

Amplifiers & Linear
Audio
Broadband RF/IF & Digital Radio
Clocks & Timers
Data Converters

DLP & MEMS
High-Reliability
Interface
Logic
Power Management

Processors

- ARM Processors
- Digital Signal Processors (DSP)
- Microcontrollers (MCU)
- OMAP Applications Processors

Switches & Multiplexers
Temperature Sensors & Control ICs
Wireless Connectivity

Retrieved from "https://processors.wiki.ti.com/index.php?title=Serial_Boot_and_Flash_Loading_Utility_for_OMAP-L137&oldid=95154"

This page was last edited on 3 February 2012, at 13:31.