



摘要

本应用报告提供了 OMAP-L132/L138 片上只读存储器 (ROM) 引导加载程序的使用指南，整合了软件资源，并解答了常见问题。本文档对 [使用 OMAP-L132/L138 引导加载程序](#) 中提供的信息进行了补充，并涵盖了其他主题，如引导实用程序、直接使用 ROM 引导加载程序引导器件的示例，以及调试引导。此外，还提供了启动时间估算值。建议开发人员在阅读本文档之前，先参阅 [使用 OMAP-L132/L138 引导加载程序](#)。

对于需要额外定制的应用以及 ROM 引导加载程序中不存在的功能，建议使用 [Processor SDK 软件开发人员指南](#) 的 [引导](#) 一节描述的两阶段引导过程。

可从以下 URL 下载本应用报告中讨论的示例和实用程序：<https://www.ti.com/lit/zip/spracm8>。

内容

1 OMAP-L138 启动过程	2
1.1 复位期间.....	2
1.2 复位后.....	2
2 启动实用程序	3
2.1 AISgen.....	3
2.2 串行引导和闪存加载实用程序.....	3
2.3 对定制板的修改.....	5
2.4 重建和定制启动实用程序.....	5
3 启动示例	8
3.1 启动二进制文件.....	8
3.2 在 AM1808/OMAPL138 上启动 DSP 二进制文件.....	9
4 调试引导加载程序	11
5 OMAP-L138 开机基准测试	12
5.1 主机启动性能.....	12
5.2 测试详细信息.....	12
6 OMAP-L138 引导加载程序常见问题解答	14
7 参考文献	15
A 使用 SYS/BIOS 设置 c_int00	16
修订历史记录.....	16

表格清单

表 5-1. 主机启动性能.....	12
表 5-2. 硬件表.....	13

商标

Code Composer Studio™ is a trademark of Texas Instruments.

Microsoft® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

所有商标均为其各自所有者的财产。

1 OMAP-L138 启动过程

本应用报告描述了 OMAP-L138 Arm® + DSP SoC 的启动过程。除非另有说明，本节内容也适用于 AM1808 和 C6748 器件。

1.1 复位期间

有两种类型的复位：

- 上电复位 (POR) - 当 RESET 和 TRST 均为低电平时，发生 POR。在此状态下清除内部内存。当 RESET 达到高电平时，将锁存启动引脚。
- 热复位 - 当 RESET 为低电平，TRST 为高电平时，发生热复位。在此状态下保留内部内存。当 RESET 达到高电平时，不会锁存启动引脚，而是保留与上次 POR 相同的值。

对于器件来说，在最初上电时要进行 POR 是很重要的。这将清除仿真和 PLL 逻辑，并正确锁存启动引脚。因此，请确保从外部将 TRST 下拉到电路板上。不这样做可能会导致许多与启动相关的问题。

当器件保持复位状态时，从内部下拉所有器件 I/O。

对于启动和配置引脚，如果它们均被引出并且是三态（未驱动），强烈建议采用外部上拉/下拉电阻器。尽管内部上拉/下拉电阻器存在于这些引脚上，并且电阻器可能匹配有所需的配置值，但提供外部连接有助于确保将有效逻辑电平锁存在这些器件启动和配置引脚上。此外，在启动和配置引脚上应用外部上拉/下拉电阻器既增加了用户调试的便利性，又提高了切换工作模式的灵活性。

有关如何选择与内部上拉/下拉电阻器相对的值的信息，请参阅 [OMAP-L138 C6000™ DSP+ ARM® 处理器数据手册](#) 的上拉和下拉电阻器一节。

1.2 复位后

在 RESETn 的上升沿，将发生以下步骤：

1. 启动引脚被锁定。

NOTE

这仅在退出 POR 时发生，因为启动引脚在热复位后不会被重新锁存。

2. 主内核启动¹开始执行 ROM 引导加载程序。
 - a. ROM 引导加载程序从启动介质中读取数据，并开始依次执行启动映像中的 AIS 命令²。
 - b. 最后一个 AIS 命令将跳转到可执行文件的入口点。

¹ 对于 OMAPL138 和 AM1808，ARM9 是启动主内核。对于 C6748，C674x 内核是启动主内核。

² 对于非 AIS 启动模式（如 NOR legacy 或 NOR direct），过程是不同的。有关这些启动模式的详细信息，请参阅 [使用 OMAP-L132/L138 引导加载程序](#)。

2 启动实用程序

2.1 AISgen

AISgen 是一款基于 Windows 的工具，用于生成 AIS 格式的启动映像。节 3 演示如何使用 AISgen 获取可执行文件并将其转换为可启动映像。

- [AISgen 下载链接](#)

建议将该工具安装在其默认位置。

2.2 串行引导和闪存加载实用程序

该软件包包含一组从 Windows 命令行运行的实用程序，用于通过串行端口刷新 OMAP-L138 EVM 的 NAND、NOR 和串行外设接口 (SPI) 闪存。

可以从 [下载最新的源代码和二进制文件](#)。

串行引导和闪存加载实用程序 (提供与以前称为 DVFlasher 的程序相同的功能) 可执行文件称为 sfh_OMAP-L138.exe。

这些程序各自封装了一个独特的二进制 UBL，该 UBL 通过通用异步接收器/发送器 (UART) 传输。这意味着芯片必须在 UART 启动模式下运行，从而显示 BOOTME 提示符。

NOTE

假设器件的 UART 在 115200, 8N1 上运行。如果与器件一起使用的振荡器与 EVM (在定制平台上) 上使用的振荡器不匹配，则波特率假设可能不正确。

2.2.1 编译

包含一个 makefile 来编译每个实用程序的主机部分和目标部分。首先构建目标部分，因为它们被嵌入到主机可执行文件中。这些目标部分通过 UART 引导模式被加载到 OMAP-L138 器件。

请注意，预构建的可执行文件已经过测试，可在 EVM 上运行。除非需要对定制板进行更改，否则无需重新构建这些文件。

2.2.1.1 在 Windows 下

对于特定平台，此软件包中包含了关于重建实用程序的详细说明，请参阅节 2.4.4。

2.2.1.2 在 Linux 下

Mono Framework 必须已安装并位于路径中。RPM 可从 [Mono 网站](#) 上找到。以下说明假设 GNU 交叉编译器工具 (arm-arago-linux-gnueabi-gcc 等) 位于当前路径中。

1. 构建某些组件还需要 C6x 编译器工具。这些都可以在 [TI 网站](#) 上免费获得。
2. 您需要将 bin 目录添加到路径中，并按照安装程序的指示设置环境变量。编辑 Common 目录下的 build.mak 文件，并添加到 TI Armand C6x 编译器的路径。
3. 然后转至软件包的 GNU 目录，并运行：

```
make
```

2.2.2 运行

2.2.2.1 在 Windows 下

在安装了 2.0 或更高版本 .Net Framework 的 Windows 下，可通过执行以下步骤从命令行运行该实用程序：

1. 将启动引脚设置为 UART2 启动模式。

对于 LogicPD OMAP-L138 EVM：

引脚编号	1	2	3	4	5	6	7	8
位置	关闭	关闭	关闭	关闭	关闭	关闭	打开	打开

对于 Spectrum Digital EVM：

引脚编号	1	2	3	4	5	6	7	8
位置	关闭	关闭	关闭	关闭	关闭	打开	关闭	打开

2. 将串行电缆从主机连接到 OMAP-L138 EVM。
3. 在主机上打开命令提示符，并将目录更改为 SFH 可执行文件所在的位置。
4. 使用以下命令运行刷新实用程序：
 - a. `..\OMAP-L138\gnu\sfh_OMAP-L138.exe [options]`
5. 导通电路板。

2.2.2.2 在 Linux 下

串行闪存器可以在安装了最新开源 MonoFramework 的 Linux 计算机上运行。除了命令外，步骤与 Windows 环境相同：

- `mono ./sfh_OMAP-L138.exe [options]`

2.2.3 串行闪存器选项

串行闪存器有三种使用模式：

- 擦除目标闪存类型 - 这将擦除闪存的全部内容。
 - `..\sfh_OMAP-L138.exe -erase`
- 使用单个应用程序映像刷新内存 - 这会将应用程序映像放置在闪存的地址 0x0 上。。这必须是一个 AIS 格式的二进制文件，可以使用与 [使用 OMAP-L132/L138 引导加载程序](#) 和相关文件绑定的 AISgen 实用程序创建。
 - `..\sfh_OMAP-L138.exe -flash_noubl <binary application file>`
- 自定义使用次级引导加载程序：工具中支持第三个选项，允许用户使用用户自定义的次级引导加载程序 (SBL)，该引导加载程序使用两步引导过程。在此设置中，次级引导加载程序在闪存的地址 0x0 处编程，应用程序在偏移 0x10000 处刷新。

用例：

- `..\sfh_OMAP-L138.exe -flash <UBL binary file> <binary application file>`

有关这些器件的次级引导方法的使用，请参阅 Processor SDK RTOS。

对于 C6748 器件，请使用以下选项：“-targetType C6748”。

对于 AM1808 器件，请使用以下选项：“-targetType AM1808”。

NOTE

- 只有在 OMAP-L138/GNU/UBL 目录中找到的 UBL 才能与串行闪存器一起使用。其他 UBL 与串行闪存器格式不兼容。
- “flash_noubl”和“flash”选项自动擦除必要的内存量以适合 UBL 或应用程序映像。

目前，支持的闪存类型只有 NAND、NOR 和 SPI。

其他选项如下所示：

```
-targetType      : Specifies exact target type within OMAP-L138 family (default OMAP-L138)
-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 target device
-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)
-APPFlashBlock   : Changes the block to flash the image into (only for no_ubl mode)
```

一旦运行任何命令，就会出现“Waiting for BOOTME...”提示。重启电路板，或按下复位按钮继续。

2.3 对定制板的修改

工具使用的默认设置只适用于 EVM。对于定制板，很可能需要进行更改，并且必须重建工具。应该对 OMAP-L138/Common/src 和 OMAP-L138/Common/include 目录中的文件进行定制更改。常见的更改包括：

- DDR 配置
 - OMAP-L138/Common/src/device.c：修改传递到函数 DEVICE_ExternalMemInit 的参数，以匹配定制板的 DDR 计时要求。
- UART 设置
 - OMAP-L138/Common/include/device_uart.h：修改 #define DEVICE_UART_PERIPHNUM，以设置将哪个 UART 连接到主机 PC。
 - 请注意，由于未使用流量控制，因此只需连接 RX 和 TX 线用于引导和闪存。
- SPI 设置
 - OMAP-L138/Common/include/device_spi.h：修改全局宏，以选择适当的外围设备和芯片选择编号。
 - OMAP-L138/Common/src/device_spi.c：修改 DEVICE_SPI_MEM_params 中的闪存组织。
 - 无需解锁 EVM 上的 SPI 闪存，即可擦除或写入。一些 SPI 闪存可能需要在写入之前设置 BL 位。要查看是否需要将此步骤添加到初始化中，请参阅器件专用数据表。
- NAND 设置
 - OMAP-L138/Common/include/device_async_mem.h：修改 #define DEVICE_ASYNC_MEM_NANDBOOT_BUSWIDTH，以匹配 NAND 的总线宽度。
- PLL 设置 (对于非 24MHz 输入时钟)
 - OMAP-L138/Common/src/device.c：修改传递到函数 DEVICE_PLL0Init 和 DEVICE_PLL1Init 的参数，以设置合适的 PLL 输出频率。
 - OMAP-L138/Common/src/device_uart.c：修改 DEVICE_UART_config 中的“分压器”字段，以实现 115200 波特率。

2.4 重建和定制启动实用程序

2.4.1 下载闪存实用程序和启动实用程序

从 下载。

NOTE

对于使用版本 2.36 的用户，强烈建议升级到版本 2.40 以获得更好的安装和构建体验。

2.4.2 安装和配置所需的软件

2.4.2.1 Cygwin

1. 可从 [Cygwin 网站](#) 下载 Cygwin。
2. 安装时，请添加以下默认未选中的包：
 - a. Devel-->make：“make”实用程序的 GNU 版本
 - b. Devel-->subversion：版本控制子系统
 - c. Editor-->vim (或类似)
3. 安装后，请验证变量 TMP 和 TEMP 是否都包含 /tmp。

```
echo $TMP echo $TEMP
```
4. 两个命令都应该返回“/tmp”。如果由于某种原因而未返回，您需要编辑 cygwin\home\[user]\.bashrc 文件来创建命令：
 - a. 导出 TMP=/tmp
 - b. 导出 TEMP=/tmp
5. 完成此步骤后，重新启动 Cygwin。

2.4.2.2 Microsoft® .NET Framework

- 下载最新版本的 [.NET Framework \(4.0 或更高版本 \)](#)。
- 将 C# 编译器 (csc.exe) 的位置添加到系统路径环境变量中。
 - 通常这是 C:\WINDOWS\Microsoft.NET\Framework\v4.0

2.4.3 编译器工具

请注意，构建串行闪存器的所有组件都需要使用 C6x 编译器工具，即使对于仅 ARM 器件也是如此。

2.4.3.1 编译器工具 (CODESOURCERY G++ LITE)

- 从 [Mentor Graphics 网站](#) 下载编译器工具。请注意，Code Sourcery 现在只对注册用户开放，不再免费开放。
 - 如果使用 v2.40 或更高版本：编辑 Common/build.mak 文件以获得正确的 ARM_TOOLS_PATH 和 ARM_TOOLS_PREFIX 变量。
 - 如果使用 v2.36 或更低版本：将交叉编译器工具的 bin 目录添加到系统路径环境变量中。

2.4.3.2 C6X 编译器工具

- 可从 [TI 网站](#) 下载 C6X 编译器工具。
 - 如果使用 v2.40 或更高版本：编辑 Common/build.mak 文件以获得正确的 DSP_TOOLS_PATH 变量
 - 如果使用 v2.36 或更低版本：将 TI C6000 代码生成工具的 bin 目录添加到系统路径环境变量中 (C:\Program Files\Texas Instruments\C6000 Code Generation Tools 6.1.11\bin)

由于 cl6x 程序 (C6000 编译器) 是本地 Windows 应用，构建环境期望用户设置 Windows 包含路径和库搜索路径，因此使用 Cygwin 路径设置 DSP_TOOLS_PATH，而使用 Windows 规范设置 DSP_LIB_PATH。Cygwin 路径可帮助您的 make 设置找到编译器，Windows 库搜索路径则可使编译器找到 RTS 库。

例如：

```
DSP_TOOLS_PATH?=/cygdrive/c/ccsvx/tools/compiler/c6000/  
DSP_LIB_PATH?="C:\ccsvx\tools\compiler\c6000\lib"
```

2.4.3.3 较新版本 CCS

可将 CCSv3 项目文件导入较新版本的 Code Composer Studio™。建议在执行此操作时使用默认导入选项。请勿将项目文件复制到工作区中。应将其留在原位。

2.4.4 为特定平台重建串行闪存和启动实用程序包

1. 对于特定平台，提取的包由一个“Common”目录和一个“<PlatformName>”目录组成。
2. 打开 Cygwin 提示符，类似于 Windows 下的 Unix/Linux 提示符。

- a. 如果使用 v2.36 或更早版本，请将必要的组件添加到路径中：

```
export PATH=<arm-compiler-root>/bin:<C6000-Compiler-Root>/bin:$PATH
```

- b. 如果使用 v2.40 或更高版本，请根据系统需要来编辑 Common/build.mak 中的 ARM_TOOLS_PATH、ARM_TOOLS_PREFIX 和 DSP_TOOLS_PATH 变量。

3. 进入 <PlatformName> 目录。

```
cd <PlatformName>
```

4. 编辑 device.mak，以便仅包含所需的器件型号和闪存类型，以加快构建过程。

```
vim device.mak
```

5. [可选] 要仅重建命令行工具（而不是 CCS 项目），请 cd 进入 GNU 目录。

```
cd GNU
```

6. [可选] 要仅重建 CCS 项目，请 cd 进入 CCS 目录

```
cd CCS
```

7. 如果要重建所有内容，请留在 <Platform> 目录中。

8. 运行“make clean”和“make”

```
make clean  
make
```

9. 如果您希望清理已构建的组件，请在您希望清理的路径上运行“make clean”。

2.4.5 为 OMAPL13x 重建 HexAIS 实用程序

HexAIS 实用程序是串行闪存和启动实用程序包中提供的启动映像生成命令行实用程序。可以在根目录的 Common 目录中找到该包的源文件。源文件分布在 AIS 和 UtilLib 目录下。主机实用程序是用 C# 编写的。要重建该实用程序，请按照节 2.4.2 中的说明设置环境变量。将目录更改为 OMAP-L13x/GNU/AISUtils/HexAIS，并运行“make clean”和“make”。

NOTE

在尝试构建 SFH 和 SFT 包之前，您需要构建该实用程序。

3 启动示例

3.1 启动二进制文件

OMAP-L138 包含一个 Arm9 内核和一个 C674x DSP 内核。复位后，DSP 保持在复位状态，并开始执行引导加载程序。以下说明演示了如何启动使 OMAP-L138 EVM 上的 LED 闪烁的代码。有一节是启动代码在 Arm 上运行，有一节是启动代码在 DSP 上运行。

3.1.1 说明

本节提供了有关获取可执行文件并使用 AISgen 工具将其转换为可启动映像的信息。

本节不适用于 C6748 器件。

使用一个简单的测试程序，在启动后使 OMAP-L138 EVM 上的 LED 反复闪烁。

SOC 设计要求 Arm 内核处于管理员模式，以配置引脚多路复用 (PINMUX) 寄存器。TI Arm 工具链在初始化期间以用户模式配置内核，因此 Arm 应用需要包含 boot.asm 源文件，以将 Arm 执行状态更改为管理员模式。该文件应包含在需要访问某些 SYSCFG 寄存器并需要管理员模式的所有项目中。

3.1.2 获取软件

需要以下软件：

- 可从 <https://www.ti.com/lit/zip/spracm8> 下载示例代码和可启动映像。
 - 内容包括：
 - 用于使 OMAP-L138 EVM 上的 LED 闪烁的预编译 AIS 引导映像 (SPI、NAND、NOR、UART)。
 - CCS 版本 3.3 项目，以及用于使 OMAP-L138 EVM 上的 LED 闪烁的源代码。
 - 用于生成 AIS 启动映像的 AISgen 工具的配置文件。
- AISgen 工具。有关更多信息，请参阅 [节 2.1](#)。
- 串行启动和闪存加载实用程序。有关更多信息，请参阅 [节 2.2](#)。

3.1.3 运行

1. 编译源代码 (可选)。

- a. 在 CCS 中打开项目文件并单击“Build”。预编译的 .out 文件位于软件包的 Debug 目录中。

Windows 7 用户

在管理权限下运行 AISGen 和 HexAISGen 实用程序。在 AISGen 工具的命令行实用程序上，右键单击并以管理员身份打开，然后键入以下内容：

```
C:> runas /user:Administrator HexAIS_OMAP-L138 <options>
```

2. 生成 AIS 映像 (可选)：

- a. 打开 AISgen 工具，并选择“File -> Load Configuration”。
- b. 选择所需启动介质对应的 .cfg 文件 (位于此软件包的“Debug”目录中)。
- c. 为您使用的芯片版本选择正确的器件类型。如果不确定，请使用 [节 4](#) 中提供的 GEL 文件来确定。
- d. 在“Application File”框中提供 OMAP-L138-ARM-LED.out 文件的路径。
- e. 提供将创建目标 AIS 文件的路径。
- f. 单击“Generate AIS”。

3. 闪存 AIS 映像：

- a. 对于 NAND、NOR 和 SPI 启动模式，使用“-flash_noubl”选项通过串行闪存器将映像闪存到所需的启动介质。有关更多信息，请参阅 [节 2.2.3](#)。
- b. 对于 UART 启动模式，请使用串行加载程序从 UART 主机引导，如下所示：

```
./slh_OMAP-L138.exe -waitForDevice OMAPL138-ARM-LED-uart.bin
```

3.2 在 AM1808/OMAPL138 上启动 DSP 二进制文件

3.2.1 说明

本节提供有关如何在 Arm 启动器件上启动 DSP 可执行文件的信息。用户引导加载程序 (UBL) 将首先运行，并用来唤醒 DSP，从而开始执行 DSP 应用。Arm 和 DSP 程序组合成一个可被直接刷新和启动的 AIS 映像。

NOTE

本节不适用于 AM1808 器件。

使用一个简单的测试程序，其中 DSP 在启动后使 OMAP-L138 EVM 上的 LED 反复闪烁。

这是一个非常简单的过程，需要记住一些关键事项：

- Arm 必须设置 HOST1CFG 寄存器以更改 DSP 复位向量。这应该指向 DSP 可执行文件的入口点，但它只能被设置为 1KB 边界（保留底部 10 位，读作 0）。因此，DSP 应用的链接器命令文件应专门迫使入口点也对齐到 1KB 边界上。在本示例中，地址 0x80010000 处创建了一个名为“entry_point”的内存区域，并将“.text:_c_int00”段分配给该内存区域。

NOTE

如果您使用 SYS/BIOS，请参阅 [Appendix A](#) 了解如何设置 c_int00。

- 和 DSP .map 文件不能重叠。如果两个程序使用相同的内存地址，它们将相互覆盖并很可能崩溃。应编写链接器命令文件以防止这种情况发生。
- 对于芯片版本 1.0 和 1.1，在写入 HOST1CFG 寄存器之前必须解锁 KICK 寄存器。此外，对于所有的修订版本，Arm 必须处于管理员模式。SOC 设计要求 Arm 内核处于管理员模式，以配置引脚多路复用 (PINMUX) 寄存器。TI Arm 工具链在初始化期间以用户模式配置内核，因此 Arm 应用需要包含 boot.asm 源文件，以将 Arm 执行状态更改为管理员模式。该文件应包含在需要访问某些 SYSCFG 寄存器并需要管理员模式的所有项目中。
- 如果 DSP 存储器映射的任何段在 DSP L2 RAM (0x11800000) 中，请注意两点：
 - 请确保 DSP 链接器命令文件中的所有 L2 RAM 地址都是在 0x118xxxxx 范围内而不是 0x008xxxxx 范围内引用的，因为这些地址无法写入 0x008xxxxx 地址范围
 - 使用 AISGen 的“配置 PSC”功能启用 DSP LPSC (PSC0,#15)。如果 DSP 巨型模块处于复位状态，则 L2 RAM 将无法访问，并且段加载将失败。该项目中包含的 AISGen CFG 文件默认启用 PSC0,#15。

3.2.2 获取软件

需要以下软件：

- 可从 <https://www.ti.com/lit/zip/spracm8> 下载示例代码和可启动映像。
 - 内容包括：
 - 用于使 OMAP-L138 和 C6748 EVM 上的 LED 闪烁的预编译 AIS 启动映像 (SPI、NAND、NOR、UART)。
 - CCS 版本 3.3 和 DSP 项目，以及通过 OMAP-L138 和 C6748 EVM 上的 DSP 使 LED 闪烁的源代码。
 - 用于生成 AIS 启动映像的 AISgen 工具的配置文件。
- AISgen 工具。有关更多信息，请参阅 [节 2.1](#)。
- 串行启动和闪存加载实用程序。有关更多信息，请参阅 [节 2.2](#)。

3.2.3 运行

3.2.3.1 OMAP-L138 EVM

1. 编译源代码：
 - a. 提供了两个项目：一个用于 UBL，一个用于 DSP LED 闪烁程序。在 CCS 中打开每个项目文件，然后在相应的处理器窗口中点击“Build”。预编译的 .out 文件位于软件包的 Debug 目录中。
2. 生成 AIS 映像。
3. 打开 AISgen 工具，并选择“File -> Load Configuration”。
4. 选择与所需启动介质对应的 .cfg 文件（位于 OMAPL138-DSP-LED 文件夹的基本目录中）。
5. 为正在使用的芯片版本选择正确的器件类型。如果不确定，请使用 [节 4](#) 中提供的 GEL 文件来确定。
6. 使用 UBL 和 DSP 程序填写“Application File”字段。当给定两个源应用程序文件时，AISgen 工具将按照它们列出的顺序将它们合并成一个映像。
 - a. 首先，单击“...”按钮并提供 OMAPL138-DSP-LED-ARM.out 文件的路径。
 - b. 然后，单击“+”按钮并选择 OMAPL138-DSP-LED-DSP.out 文件。
7. 通过给出目标 AIS 文件将创建的路径来填写“AIS Output File”字段。
8. 单击“Generate AIS”。

闪存 AIS 映像：

1. 对于 NAND、NOR 和 SPI 启动模式，使用“-flash_noubl”选项通过串行闪存器将映像闪存到所需的启动介质。有关更多信息，请参阅 [节 2.2.3](#)。
2. 对于 UART 启动模式，请使用串行加载程序从 UART 主机启动，如下所示：
 - a. `./slh_OMAP-L138.exe -waitForDevice OMAPL138-ARM-LED-uart.bin`

3.2.3.2 C6748 EVM

在这种情况下，只需要 DSP 程序。上述相同的说明适用于一些例外情况：

- 使用 C6748-LED-x.cfg 文件创建 C6748-LED-x.bin。请注意，只有 DSP 应用需要作为输入文件。
- 闪烁时，请使用 -targetType C6748 选项。

4 调试引导加载程序

OMAP-L1x Debug GEL 文件可用于调试引导问题、确定 ROM 版本以及显示 PLL 设置等其他信息。此 GEL 文件是为所有 OMAP-L13x、AM1x 和 C674x 器件而创建的。

Debug GEL 文件与 CCS 安装捆绑在一起，可在
CCS_INSTALL_PATH\ccs_base\emulation\boards\levmomapl138\gel 中找到。

CCS 说明：

1. 连接到处理器，可以是任何 OMAP-L1x、AM1x 或 TMS320C674x 器件的 Arm 或 DSP。
2. Tools -> Gel Files。
3. 右键单击窗口并选择“Load Gel”。
4. 转至“Scripts -> Diagnostics -> Run All”。

NOTE

确保在运行脚本时禁用 MMU，以便正确读取寄存器。可通过转至“Tools -> Advanced Features”，禁用 CCS 中的 MMU。

GEL 文件将打印出以下信息：

- ROM ID：启动 ROM 的修订号
- 芯片修订号
- 启动模式：当前启动模式，由复位时锁存的启动引脚选择
- ROM 状态代码：ROM 代码的当前状态
- 说明：有关 ROM 在启动过程中可能遇到的任何错误信息的说明
- 程序计数器：所连接器件（或 DSP）的当前程序计数器
- 器件信息：从 TI 获得支持时可能会有所帮助的通用器件信息
- 时钟信息：输出 PLLm_SYSClKn

NOTE

如果电路板使用 24MHz 以外的输入时钟，您需要相应地修改 gel 文件开头处定义。

- PSC 状态信息

5 OMAP-L138 开机基准测试

本节给出了 OMAP-L138/AM1808/C6748 和衍生产品所支持的各种启动模式的启动时间测量结果。尽管认为这些数字是典型值，但它们只适用于确切的基准测试环境。

5.1 主机启动性能

表 5-1. 主机启动性能

器件	启动模式	启动时间 (ms)	
		1KB	124KB
C6748	NAND8	60	80
	NAND16	36	52
	NOR	3	15
	SPI 闪存	4	96
	SD	34	195
	SDHC	42	180
	MMC	68	84
OMAP-L138 AM1808	NAND8	200	244
	NAND16	144	203
	NOR16	32	56
	SPI 闪存	36	312
	SD	320	592
	SDHC	336	544
	MMC	264	460

5.2 测试详细信息

5.2.1 方法

使用的测试用例会在主函数开始时立即切换 GPIO。从 RESETn 的上升沿到 GPIO 的下降沿测量引导时间。

5.2.2

电路板开始时处于通电状态，以便消除上电时间的不确定性。例如，一些 MMC/SD 卡在通电后需要等待 1 秒的时间才能访问。

使用了两种映像尺寸：

- 1KB 映像尺寸的启动时间基本上表示从器件读取任何数据的初始化时间。
- 124KB 的映像尺寸显示了启动一个更大映像所需的时间。

以下公式可用于估算其他映像尺寸 (size_x) 的启动时间 (time_x)：

$$\text{time}_x = \text{time}_{1\text{kb}} + \text{size}_x * (\text{time}_{124\text{kb}} - \text{time}_{1\text{kb}}) / 123$$

5.2.3 软件

测试用例完全加载到 L3 共享内存中，因此没有使用外部内存。引导加载程序启用 PLL，主机 CPU 以 456MHz 的频率运行。

5.2.4 硬件

对 OMAP-L138/AM1808/C6748 器件的修订版 2.1 进行了测试。对于某些启动模式，启动时间可能与以前的修订版大不相同。

表 5-2. 硬件表

硬件	器件型号	频率
电路板	LogicPD OMAP-L138 EVM	-
SoC	OMAP-L138/AM1808/C6748 修订版 2.1	456MHz
NAND8	Micron MT29F4G08AAC 8 位 NAND 闪存	-
NOR	Intel PC28F640P30T85 16 位 NOR 闪存	-
SPI 闪存	Winbond W25X64VSFIG 64M 串行闪存	45.6 MHz
SDHC	Transcend 4GB Micro SD HC	38MHz
SD	Patriot 2GB SD	38MHz
MMC	palmOne 256MB MultiMediaCard	38MHz

5.2.5 讨论

结果表明，OMAP-L138/AM1808 启动时间（启动器件）通常比 C6748 启动时间（DSP 启动器件）长得多。对此有几种解释：

- 默认情况下不启用缓存，而启用 DSP 缓存。这意味着代码（尤其是延时循环），将在 DSP 上执行得更快。
- OMAP-L138/AM1808 在引导加载程序开始执行之前需要额外的时间。
- DSP 编译器为延时循环生成的指令比编译器少，因此执行速度更快。

对于 OMAP-L138/AM1808 上较大的启动映像，最好通过 AIS 命令启用缓存。AISgen 的未来版本将提供该选项。

6 OMAP-L138 引导加载程序常见问题解答

问题: 如何启动可执行文件？

答案: 必须使用 AISgen 工具将 COFF/ELF 格式的可执行文件转换为 AIS 格式。有关此过程的更多详细信息和示例，请参阅 [节 3.1](#)。

问题: 如何启动 DSP 可执行文件？

答案: 由于 OMAP-L138 是 ARM 启动器件，因此必须首先引导可执行文件以唤醒 DSP。对于 C6748，可以直接启动 DSP AIS 二进制文件。有关此过程的更多详细信息和示例，请参阅 [节 3.2](#)。

问题: 是否有任何适用于 EVM 的示例 AISgen 配置文件？

答案: 以下文件包含将内核/mDDR 频率设置为 456/150MHz 和 300/132MHz 的 AISgen 配置文件。

下载示例 AISgen 配置文件。这些可从以下位置上的 AISgen 包获得：`：AISgen for D800K008\cfg_files`。

请注意，选择 OPP 时必须适当调整内核电压。

问题: 为什么我的程序可以在 CCS 中运行，但无法从闪存中启动？

答案: Gel 文件的依赖性

常见的问题是在代码需要的 GEL 文件中做了一些配置，如外部内存配置、pinmux 配置或 PLL 配置。大多数功能也可以由引导加载程序使用 AISgen 工具来执行。启动后比较 PLL、pinmux 和 DDR 寄存器，并验证它们是否与 GEL 文件设置的情况相匹配。

答案: 错误的外部内存配置

如果您的代码在外部内存中有一些段，则引导加载程序必须在复制该代码之前，先配置控制器设置。验证 AISgen 中使用的 DDR 配置设置是否与 GEL 文件中使用的设置相匹配。如果 DDR 配置正确，您应该能够在 CCS (0xC0000000) 中插入 DDR 内存区域。

答案: 管理员模式与用户模式

SOC 设计要求 Arm 内核处于管理员模式，以配置引脚多路复用 (PINMUX) 寄存器。TI Arm 工具链在初始化期间以用户模式配置内核，因此 Arm 应用需要包含 boot.asm 源文件，以将 Arm 执行状态更改为管理员模式。可在 [节 3.2.2](#) 的 zip 文件中找到该文件。在 CCS 中，应在项目链接器选项中勾选“Search libraries in priority order (--priority, -priority)”框以避免链接错误。

答案: KICK 寄存器解锁

在以前版本的器件 (版本 1.0 和 1.1) 上，在访问某些受保护的寄存器之前，必须先解锁 KICK 寄存器。虽然 CCS 中的 GEL 文件可以自动执行此操作，但应在代码开始执行以使文件可移植。有关该过程的更多信息，请参阅 [OMAP-L138 C6000 DSP+ARM 处理器技术参考手册](#)。

答案: 错误的启动模式

再次检查启动引脚在复位上升沿处的电压是否合适。使用 [节 4](#) 中的 Debug GEL 文件确定器件锁存的启动模式，并查看任何其他 ROM 错误消息。另外，请验证 TRST 是否从外部下拉。

答案: 使用引导加载程序共享内存

ROM 引导加载程序本身使用从 0x80000000 开始的 16KB 共享 RAM，用于多种用途。该内存不应该被用户应用程序的任何初始化段使用。

答案：RAM 与 ROM 自动初始化模型

通过 ROM 启动代码时，请确保在链接器选项中使用 ROM 自动初始化模型 (-c)。如果使用 RAM 自动初始化模型 (-cr)，即使某些内存位置在 CCS 下，也永远不会被加载。可以在此处找到构建选项：

- CCS3.3 : Build Options -> Linker -> Autoinit Model
- CCS4+ : Build Options -> Runtime Environment

问题：是否需要次级引导加载程序 (UBL)？

答案：在引导加载程序无法解析 AIS 文件的较旧器件上，需要次级引导加载程序，又名用户引导加载程序 (UBL)。通过将 AISgen 工具与 OMAP-L138 引导加载程序一起使用，以前由 UBL 执行的大部分功能都可以由引导加载程序完成。

对于典型的 Linux 应用程序，旧流程如下所示：

- UBL (设置 DDR、PSC 并将 U-Boot 复制到内存)
- U-Boot (加载 Linux 和文件系统)
- Linux

OMAP-L138 的流程如下所示：

- AIS 签名的 U-Boot (设置 DDR、PSC 并加载 Linux 和文件系统)
- Linux

所以一般来说，不需要单独的 UBL，因为 AIS 功能可以执行大部分相同的任务。

问题：如何为 NOR direct 或 NOR legacy 启动模式准备映像？

答案：有关创建启动映像的说明，请参阅 [处理器 SDK 软件开发人员指南](#) 的启动 一节。

问题：应如何格式化 SD/MMC 卡以便于启动？

答案：有关如何准备 SD/MMC 卡的详细信息，请参阅 [处理器 DK 软件开发人员指南](#) 的启动 一节。。

7 参考文献

德州仪器 (TI) : [OMAP-L138 C6000™ DSP+ Arm® 处理器数据手册](#)

德州仪器 (TI) : [OMAP-L138 C6000 DSP+Arm 处理器技术参考手册](#)

德州仪器 (TI) : [使用 OMAP-L132/L138 引导加载程序](#)

A 使用 SYS/BIOS 设置 c_int00

如果您正在使用 SYS/BIOS，则由 XDCtools 处理系统启动和初始化。对于某些目标，c_int00 代码是一个 C 函数。C 代码使用“-mo”标志编译，它将符号 _c_int00 放入 .text 的子段 (.text:_c_int00)。

要将 .text 符号放置在显式地址上，请使用类似于以下的指令将自定义链接器命令 (.cmd) 文件添加到您的项目中：

```
SECTIONS {
    .text:_c_int00 > 0xc3000000
}
```

在其他目标上，_c_int00 代码是用汇编语言编写的。从 XDC 3.24.02 开始，该汇编代码放置在 .text 而不是 .text:_c_int00 中。此问题将在 XDC 和 BIOS 的未来版本中修复。但是，在那之前，可以使用以下链接器解决方案来放置 _c_init00 函数。您必须相应地更新路径和库名称。请检查 .map 文件以查看引导库名称，检查生成的 linker.cmd 文件获得完整路径。

```
boot : > 0x3D8000 PAGE = 0
{-1"C:\ti\ccsv5_3_0_00042\xdctools_3_24_02_30\packages\ti\targets\rts2800\lib\boot.a28FP"
<boot_cg.o28FP> (.text)
}
```

修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (June 2019) to Revision A (January 2021)	Page
• 在摘要中进行了更新.....	1
• 更新了整个文档中的表格、图和交叉引用的编号格式。.....	2
• 在节 2.2 中进行了更新。.....	3
• 在节 3.1.2 中进行了更新。.....	8

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司