

## 在 STM32F7x0 超值系列和 STM32H750 超值系列 MCU 上执行外部存储器代码

### 引言

因为对能够支持新功能和复杂功能的应用的需求不断上升，对配备更大闪存区的设备的需求也在增加。

使用外部闪存可以提供具有近似性能水平的更高存储能力，同时为增加闪存区的需求提供一种经济划算的解决方案。

这样使得 STM32F7x0 超值系列和 STM32H750 超值系列设备可以更小的内部闪存区来应对市场需求。

本应用笔记描述了在这些超值系列设备上从外部存储器执行代码构建应用程序所需的步骤。

它详细讲解了如何从内部闪存启动，然后跳转到片外存储器的用户程序去执行。

### 相关文档

意法半导体网站 [www.st.com](http://www.st.com) 上提供以下文档：

- *STM32Cube MCU 软件包用于配备 HAL、底层驱动和专用中间件的 STM32F7 系列简明数据手册 (DB2601)*
- *STM32Cube MCU 软件包用于配备 HAL 和专用中间件的 STM32H7 系列简明数据手册 (DB3259)*
- *STM32F75xxx 和 STM32F74xxx 基于 32 位 MCU 的高级 Arm® 参考手册 (RM0385)*
- *STM32H743/753 基于 ARM® 的高级 32 位 MCU (RM0433) 参考手册*
- *STM32F7 系列系统架构和性能应用笔记 (AN4667)*
- *STM32 微控制器上的 Quad-SPI (QSPI) 接口应用笔记 (AN4760)*
- *STM32H7x3 硬件开发入门应用笔记 (AN4938)*
- *STM32F7 系列 MCU 硬件开发入门应用笔记 (AN4661)*
- *STM32CubeProgrammer 软件描述用户手册 (UM2337)*

## 1 概述

---

本文档适用于基于 Arm® 的器件。  
提示 *Arm* 是 *Arm Limited* (或其子公司) 在美国和/或其他地区的注册商标。



## 2 外部存储器代码执行概述

### 2.1 外部存储器代码执行原则

STM32CubeF7 v1.12.0 和 STM32CubeH7 v1.3.0 固件包提供多个应用程序，用于演示如何从内部闪存启动以及如何配置外部存储器并跳转到用户应用程序（位于外部存储器上）。有两个可能的用例可用：XiP 和 BootROM。

- **XiP** 用例旨在从外部闪存（QSPI 或 FMC-NOR 闪存）“芯片内执行”。用户应用程序代码应链接到目标执行存储器地址（外部 QSPI 或 FMC-NOR 闪存）。
- **BootROM** 用例旨在演示如何从内部闪存启动，配置外部 RAM 存储器（SDRAM 或 SRAM），将用户应用程序二进制文件从代码存储区域（SDCARD 或 SPI-Flash 存储器）复制到外部 SDRAM 或外部 SRAM，然后跳转到用户应用程序。用户应用程序代码应链接到目标执行存储器地址（外部 SDRAM 或 SRAM）。

下表中所述的应用程序可在固件包中的\Applications\ExtMem\_CodeExecution 路径下获得，供下列板使用：

- STM32F723E-Discovery 板针对 STM32F730 器件
- STM32F756G\_EVAL 板针对 STM32F750 器件
- STM32H743I\_EVAL 板针对 STM32H750 器件。

**表 1. 应用详情**

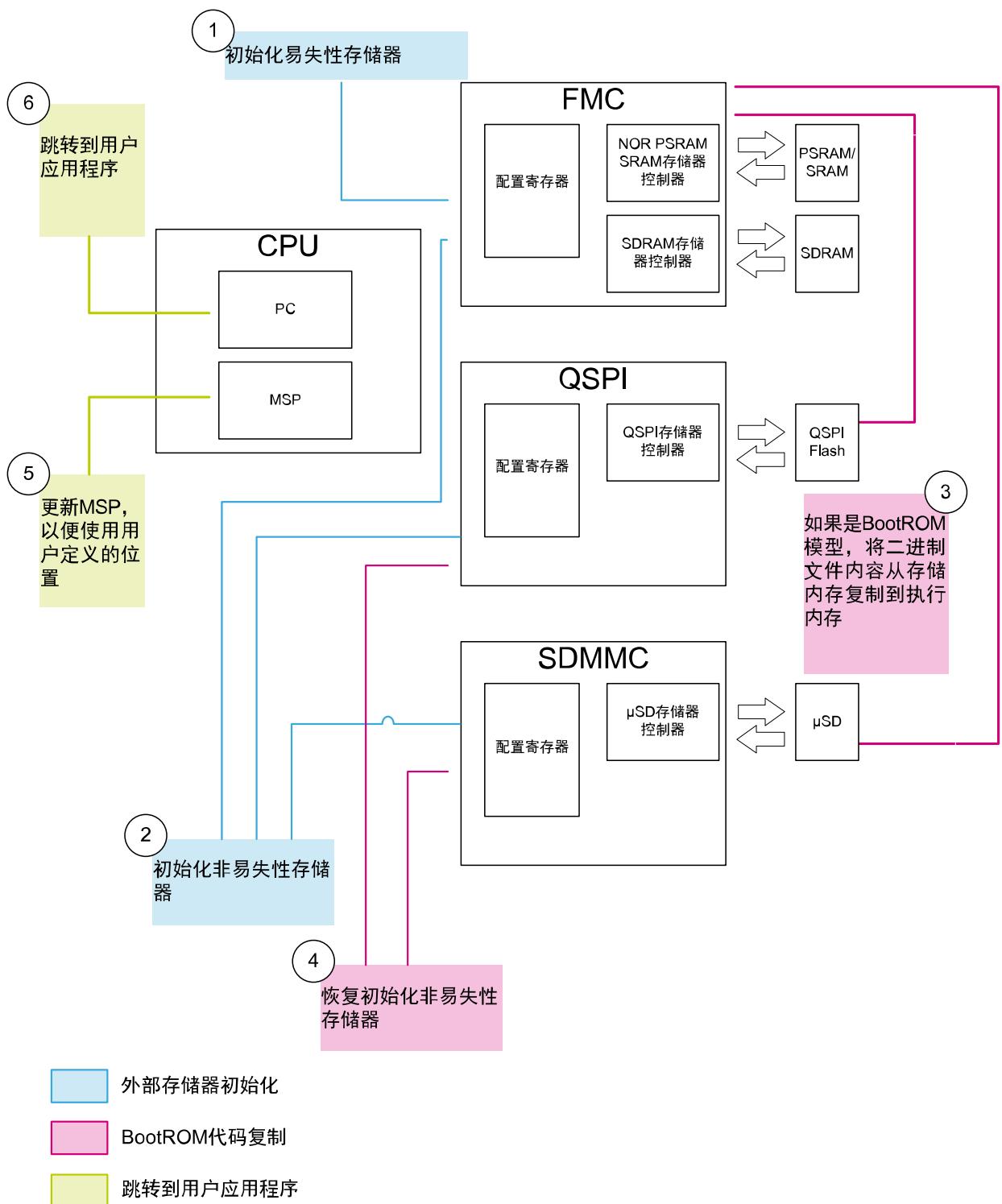
应用	说明
ExtMem_Boot	演示如何从内部闪存启动，配置外部存储器，然后跳转到位于外部存储器上的用户应用程序。 用户可以选择 QSPI 闪存、FMC-NOR 闪存、外部 SDRAM 或外部 SRAM 执行代码。
ExtMem_Application\LedToggling	从外部闪存（QSPI 闪存或 FMC-NOR 闪存）、外部 SRAM 或外部 SDRAM 运行的示例应用程序
ExtMem_Application\FreeRTOS	从外部闪存（QSPI 闪存或 FMC-NOR 闪存）、外部 SRAM 或外部 SDRAM 执行的示例 FreeRTOS 应用程序

**外部存储器启动**应用程序负责初始化所需资源，以使外部存储器随时可用。该应用程序根据用户配置初始化所需资源（参见 [第 3.3 节 配置](#)）。

**外部存储器启动**应用程序必须设置主堆栈指针，并将应用程序配置为在外部存储器上执行。该类型启动方案支持大小可调的用户应用程序。

外部存储器启动应用程序确保在跳到用户应用程序之前重置或释放安装阶段之后不再需要的任何资源。下图展示了该启动方案

图 1. 外部存储器代码启动方案

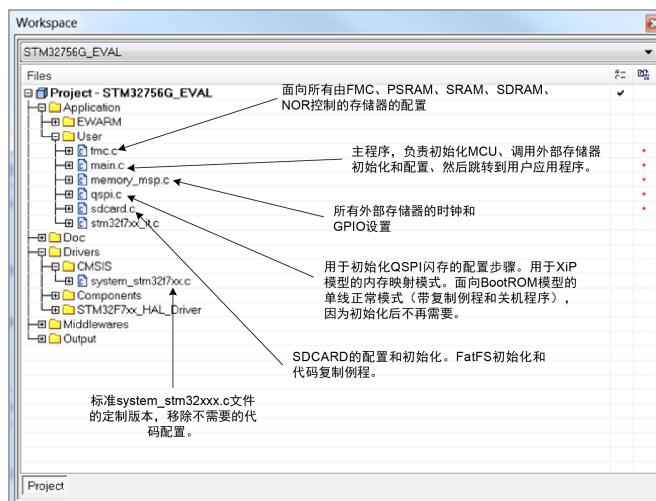


## 2.2 外部存储器启动应用程序描述

外部存储器启动应用程序包含 STM32CubeF7/H7 包上的一组源文件，这些定制的源文件可以匹配每个硬件平台支持的配置。

下图显示了所有受支持配置的所有文件超集示例。

图 2. 外部存储器启动应用程序源文件超集



## 3 支持的启动模型

该应用程序支持两种执行方式：

- 支持芯片内执行（支持 XIP）
- 支持 BootROM

用户必须通过修改 *memory.h* 头文件来选择匹配需求的配置。

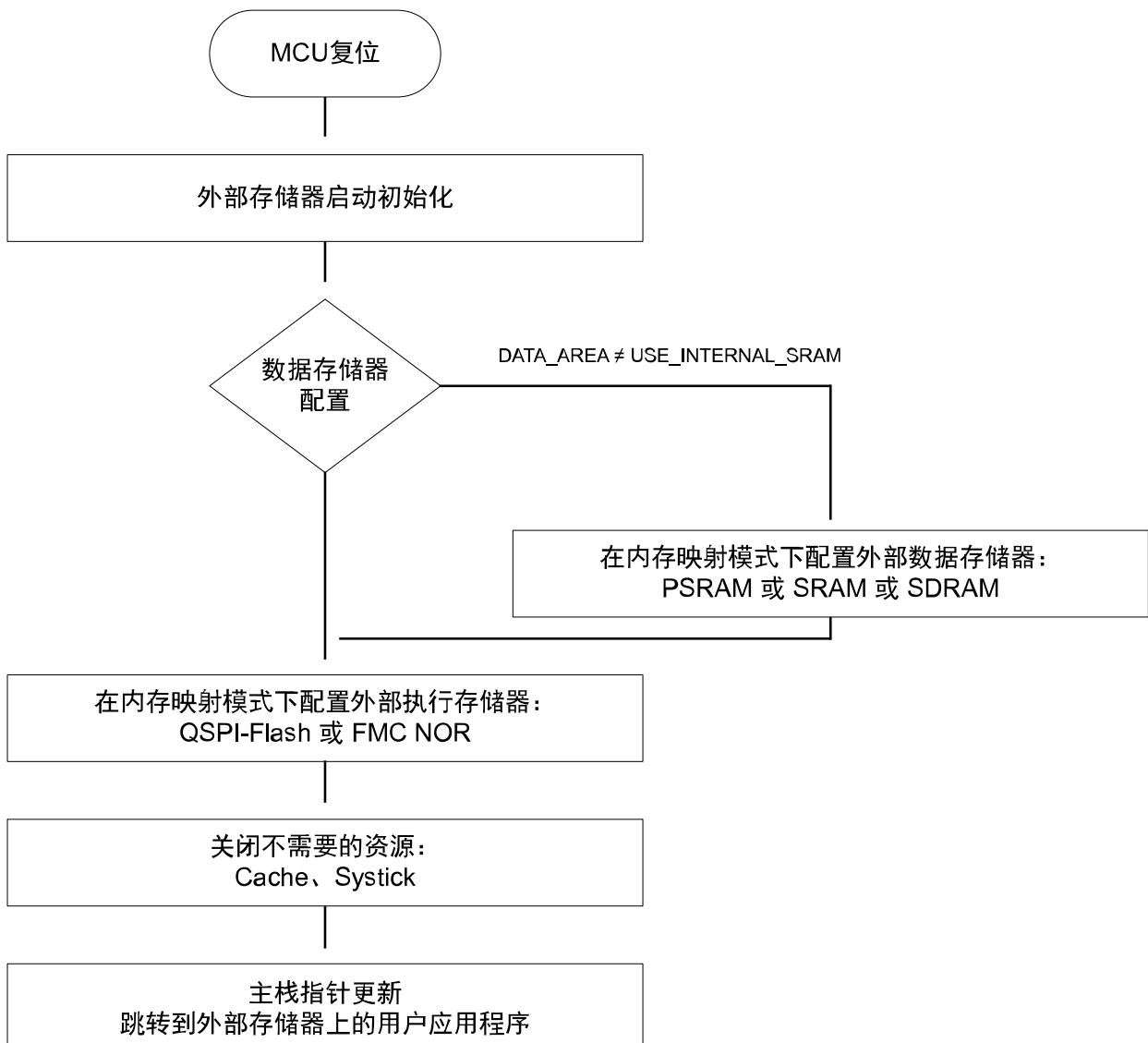
### 3.1 支持芯片内执行（XIP）

XIP 模型基于直接从用于代码存储的外部非易失性存储器执行代码。该执行模型需要内存映射支持，以允许 CPU 直接访问代码以执行用户应用程序。XIP 模型可通过 FMC/QSPI 接口在外部 NOR/QSPI 闪存上使用。

外部存储器启动应用程序基于 *memory.h* 文件中的用户配置对下列易失性存储器之一进行配置：SDRAM、SRAM、PSRAM 或内部 SRAM。在该模型中，易失性存储器仅用于数据访问。

下面的流程图说明了 XiP 模型的操作流程。

图 3. XiP 模型操作流程



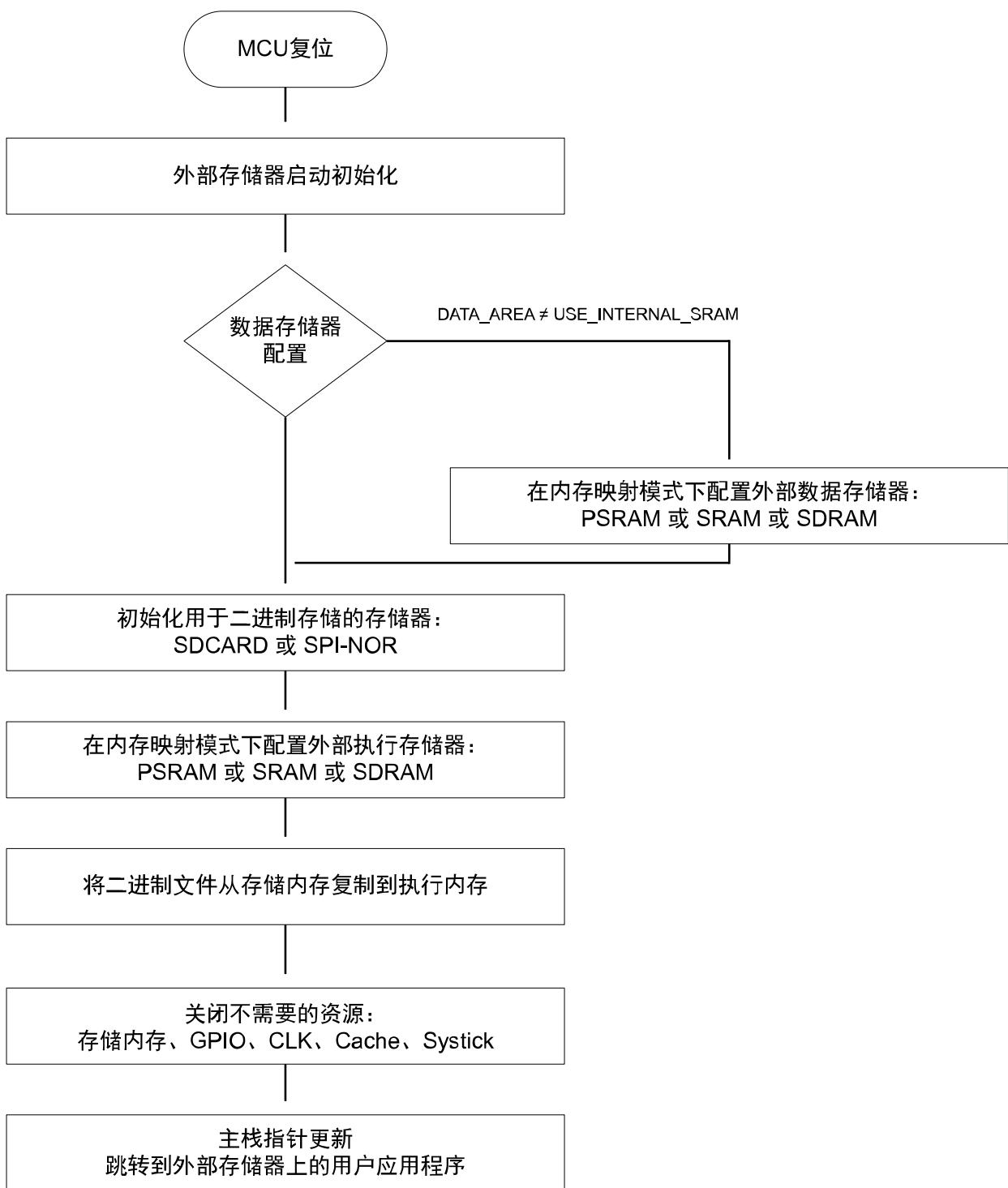
### 3.2 支持 BootROM

BootROM 模型基于从选定的易失性存储器中执行代码。当二进制数据存储在非内存映射模式的存储器（比如 SDCARD）中时，该执行模型非常适合。当二进制数据存储在低吞吐量的存储器（比如 SPI-NOR（使用单线 QSPI 进行仿真））中时，此模型也适用。

外部存储器启动应用程序基于 *memory.h* 文件中的用户配置对下列其中两个易失性存储器进行配置：SDRAM、SRAM、PSRAM 或内部 SRAM。在该模型中，二进制数据从一个非易失性存储器复制到一个易失性存储器，然后由外部存储器启动应用程序执行。第二个易失性存储器用于数据。

下面的流程图说明了 BootROM 模型的操作流程。

图 4. BootROM 模型操作流程



### 3.3 配置

用户配置由以下定义确定：

- **DATA\_AREA:** 用于指定用于数据存储的易失性存储器。支持的存储器（取决于所用的板）有：
  - USE\_EXTERNAL\_SDRAM: 外部 SDRAM 用于数据存储
  - USE\_EXTERNAL\_SRAM: 外部 SRAM 用于数据存储
  - USE\_EXTERNAL\_PSRAM: 外部 PSRAM 用于数据存储
  - USE\_INTERNAL\_SRAM: 内部 SRAM 用于数据存储
- **CODE\_AREA:** 用于指定用户应用程序的执行位置。该区域可以是用于 BootROM 方案的易失性存储器，也可以是用于 XiP 方案的非易失性存储器。支持的存储器（取决于所用的硬件）有：
  - XiP 模型: BINARY\_AREA 必须是未定义的:
    - USE\_QSPI: QSPI Flash 用于代码执行
    - USE\_NOR: FMC-NOR Flash 用于代码执行
  - BootROM 模型: BINARY\_AREA 必须已定义
    - USE\_EXTERNAL\_SDRAM: 外部 SDRAM 用于代码执行
    - USE\_EXTERNAL\_SRAM: 外部 SRAM 用于代码执行
    - USE\_EXTERNAL\_PSRAM: 外部 PSRAM 用于代码执行
    - USE\_INTERNAL\_SRAM: 内部 SRAM 用于代码执行
- **BINARY\_AREA:** 仅在 BootROM 模型中定义。它用于指定包含用户应用程序的二进制文件位置。根据所选配置，需要附加定义。支持的存储器（取决于所用的硬件）有：
  - USE\_SPI\_NOR: SPI NOR Flash 用于二进制存储
    - BINARY\_BASE\_OFFSET: SPI NOR Flash 中的二进制偏移
    - BINARY\_SIZE: 二进制图像大小
  - USE\_SDCARD: SD Card 用于二进制存储
    - BINARY\_FILENAME: 要执行的二进制文件名称

用户应确保所选存储器包含代码和数据，以至少覆盖一个适当的用户应用程序启动。然后，用户应用程序可以初始化所需的任何其他存储器。

### 3.4 外部存储器部件总结

下表总结了与所用板和启动模型对应的外部存储器部件编号。由于 STM32F7x0 超值系列和 STM32H750 超值系列器件没有专用板，使用的板（与兼容器件）为：

- STM32F723E-Discovery 用于模拟 STM32F730 器件。
- STM32F756G\_EVAL 用于模拟 STM32F750 器件。
- STM32H743I\_EVAL 用于模拟 STM32H750 器件。

**表 2. 启动模型在每个板上使用的外部存储器**

启动模型	存储器	STM32F723E-Discovery, 配置为 STM32F730	STM32756G_EVAL, 配 置为 STM32F750	STM32H743I_EVAL, 配置 为 STM32H750
XiP	QSPI 闪存	MX25L51245GZ2I-08G (总线宽度: 4 线)	N25Q512A13GSF40E (总线宽度: 4 线)	两个 Quad-SPI Flash MT25QL512ABB8ESF-0SIT 或 一个双 Quad-SPI Flash MT25TL01GHBB8ESF-0SIT (总线宽度: 8 线)
	NOR 型闪存 (在 FMC 上)	-	PC28F128M29EWL A (总线宽度: 16 位)	MT28EW128ABA1L PC-0SI T
BootROM	SPI-NOR (用 QSPI 1 线进行模 拟)	-	N25Q512A13GSF40E (总线宽度: 2 线)	两个 Quad-SPI Flash MT25QL512ABB8ESF-0SIT 或

启动模型	存储器	STM32F723E-Discovery, 配置为 STM32F730	STM32756G_EVAL, 配 置为 STM32F750	STM32H743I_EVAL, 配置为 STM32H750
BootROM				一个双 Quad-SPI Flash MT25TL01GHBB8ESF-0SIT (总线宽度: 2 线)
	SDCARD	-	原生支持	收发器 IP4856CX25/ C_Module_REV
易失性存储器	内部 SRAM	原生支持	原生支持	原生支持
	外部 SRAM	-	IS61WV102416BL L -10ML I (总线宽度: 16 位)	IS61WV102416BL L -10ML I (总线宽度: 16 位)
	外部 SDRAM	-	IS42S32800G-6BL I (总线宽度: 32 位)	IS42S32800G-6BL I (总线宽度: 32 位)
	外部 PSRAM	IS66WV51216EBLL-55BLI (总线宽度: 16 位)	-	-

## 4

## 需要考虑的资源约束

初始化后，不再需要的任何资源（中断、正在进行的传输、未使用的引脚）都应在跳到用户应用程序之前释放。必须这样做以避免额外的功耗，并限制对用户应用程序的任何干扰。特别是对于 BootROM 模型，由于不再需要用于二进制存储的外设，因此应将其重置。

用户应考虑外部存储器启动应用程序使用的资源数量，以确保外部存储器接口保持正常运行。资源约束与以下因素有关：

- 引脚分配和配置
- 接口配置（不应修改 QSPI IP 寄存器，FMC IP 寄存器即使是部分更新）
- 配置 RCC，以避免 IP 重置时钟，禁用和以有害的方式更新时钟频率/源。

下面的引脚分配表作为参考，实际中根据使用的板进行引脚选择是必须的。可根据可用的替代功能选择使用其他引脚。

表 3. 板为每个存储器分配的引脚

存储器/板	STM32F723E-Discovery, 配置为 STM32F730	STM32756G_EVAL, 配置为 STM32F750	STM32H743I_EVAL, 配置为 STM32H750
QSPI 闪存	PB (6、2)、PC (9、10)、PE2、PD13	PB (6、2)、PF (8、9、8、6)	PB2、PG (6、9、14)、PF (6、7、8、9)、PH (2、3)、PC11
NOR 型闪存 (在 FMC 上)	-	PD (0、1、4、5、6、7、8、9、10、11、12、13、14、15) PE (2、3、4、5、6、7、8、9、10、11、12、13、14、15) PF (0、1、2、3、4、5、12、13、14、15) PG (0、1、2、3、4、5)	PD (0、1、4、5、6、7、8、9、10、11、12、13、14、15) PE (2、3、4、5、6、7、8、9、10、11、12、13、14、15) PF (0、1、2、3、4、5、12、13、14、15) PG (0、1、2、3、4、5)
外部 SRAM	-	PD (0、1、3、4、5、8、9、10、11、12、13、14、15) PE (0、1、3、4、7、8、9、10、11、12、13、14、15) PF (0、1、2、3、4、5、12、13、14、15) PG (0、1、2、3、4、5、10)	PD (0、1、3、4、5、8、9、10、11、12、13、14、15) PE (0、1、3、4、7、8、9、10、11、12、13、14、15) PF (0、1、2、3、4、5、12、13、14、15) PG (0、1、2、3、4、5、6、9、10、12、13、14)
外部 SDRAM	-	PD (0、1、8、9、10、14、15) PE (0、1、7、8、9、10、11、12、13、14、15) PF (0、1、2、3、4、5、11、12、13、14、15) PG (0、1、4、5、8、15) PH (2、3、5、8、9、10、11、12、13、14、15) PI (0、1、2、3、4、5、6、7、9、10)	PD (0、1、8、9、10、14、15) PE (0、1、7、8、9、10、11、12、13、14、15) PF (0、1、2、3、4、5、11、12、13、14、15) PG (0、1、2、3、4、5、8,15) PH (5、6、7、8、9、10、11、12、13、14、15) PI (0、1、2、3、4、5、6、7、9、10)
外部 PSRAM	PD (0、1、4、5、7、8、9、10、11、12、14、15) PE (0、1、7、8、9、10、11、12、13、14、15) PF (0、1、2、3、4、5、12、13、14、15) PG (0、1、2、3、4、5)	-	-

下表总结了应保持不变的资源。它列出了不应修改的外设（或外设的一部分），以避免外部存储不可用。不应重置上述外设或禁用时钟，也不应以可能改变其行为的方式进行重新配置。

**提示** 部分要素可能会根据为所选板选择的外部存储器启动应用程序配置和平台硬件而更改。

表 4. 内存类型所需的外设

	STM32F723E-Discovery, 配置为 STM32F730	STM32756G_EVAL, 配置为 STM32F750	STM32H743I_EVAL, 配置为 STM32H750
QSPI 闪存	QSPI1 (0x9000000)	QSPI1 (0x9000000)	双线 QSPI 模式 QSPI 不再可用
NOR Flash (在 FMC 上)	-	FMC-NOR (无 FMC-PSRAM / FMC-SRAM)	FMC-NOR (无 FMC-PSRAM / FMC-SRAM)
外部 SRAM	-	FMC-SRAM (无 FMC-PSRAM / FMC-NOR)	FMC-SRAM (无 FMC-PSRAM / FMC-NOR)
外部 SDRAM	-	FMC-SDRAM (200 MHz 最大系统频率)	FMC-SDRAM
外部 PSRAM	FMC-SRAM (无 FMC-SRAM / FMC-NOR)	-	-

## 5 外部存储器用户应用程序描述

### 5.1 需要的更新

外部存储器应用程序基于特定的启动方案，该方案与标准的启动方案不同，支持从片上应用到片外应用的平稳过渡。

当应用程序的位置发生变化时，用户必须进行两项更新：

- 确保使用所需的链接器文件，并提供与所选启动选项相对应的内存映射。
- 更新 VTOR 设置，以使用正确的地址。

### 5.2 加载和调试

STM32F723E-Discovery、STM32756G\_EVAL 和 STM32H743I\_EVAL 这三种板都有面向外部非易失性内存的加载器。这些加载器在 STM32CubeF7/H7 中以如下形式提供：

- EWARM IDE 的补丁
- MDK-ARM IDE 的专用包

XiP 模型提供了类似于内部 Flash 调试的无缝加载和调试体验。对于 SW4STM32 IDE，应使用 STM32CubeProgrammer 在外部闪存上加载应用程序。

在 BootROM 模型中，应用程序被编译和链接，以便从外部易失性内存执行：

- External SDRAM：链接器地址 0xD0000000 用于 STM32H750 超值系列，而 0xC0000000 用于 STM32F7x0 超值系列
- External SRAM：链接器地址 0x68000000 用于 STM32H750 超值系列和 STM32F7x0 超值系列

然后将应用程序二进制文件存储到 SPI\_NOR 闪存或 SDCARD 中。由启动应用程序将用户应用程序从存储区域复制到执行 RAM 区域。

因此，应用程序的加载模式不能由 IDE（MDK-ARM 或 EWARM）外部存储 Flash 加载器处理（因为应用程序的链接地址和存储地址不同）。

根据 BINARY\_AREA 定义（在启动应用程序的“memory.h”文件中指定），该模型需要使用以下两种不同的加载模式：

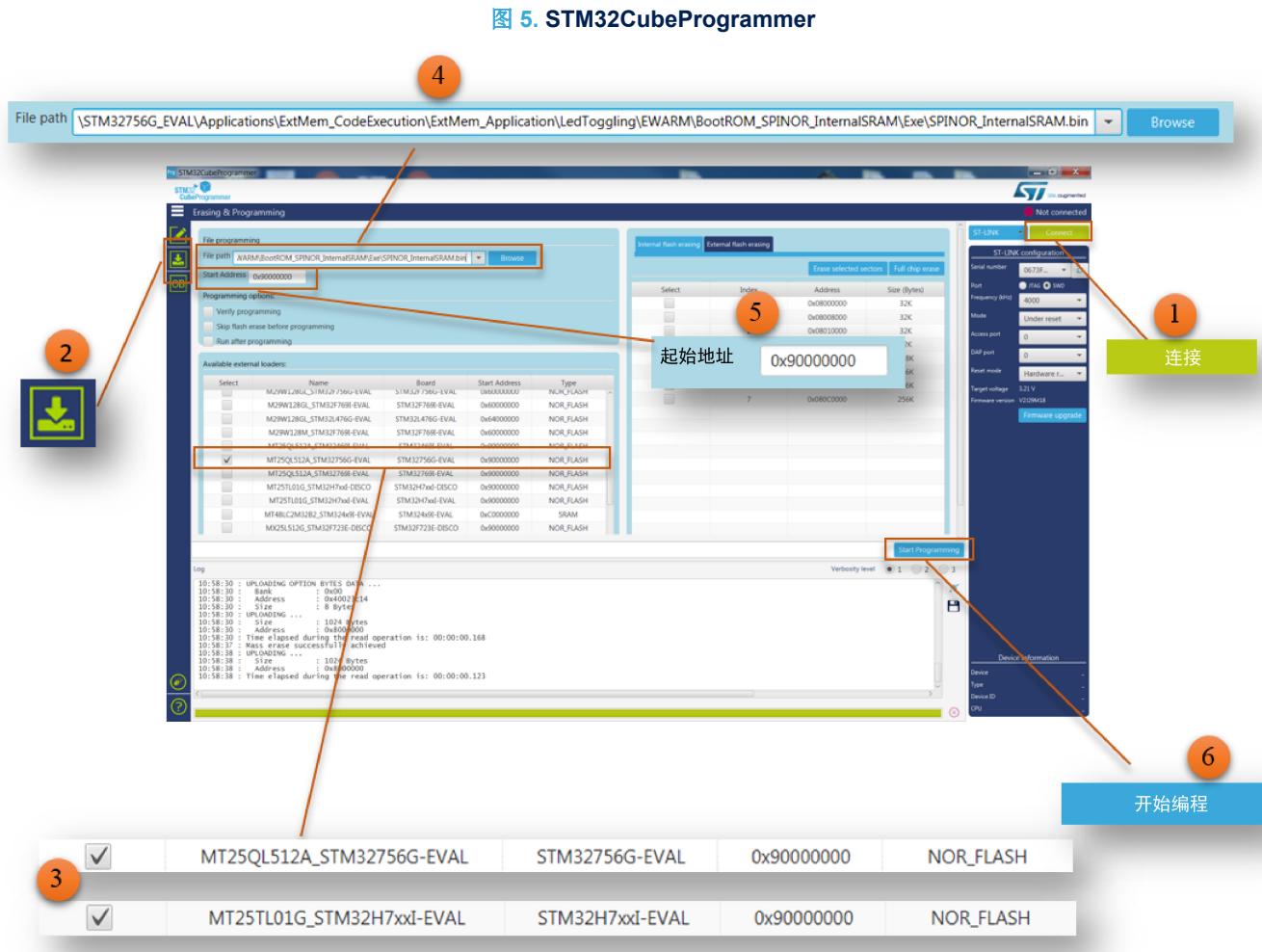
- SPI\_NOR

用户应用程序应存储在地址为 0x90000000 的 SPI-NOR 闪存中。必须使用 STM32CubeProgrammer 来完成。应用程序的输出应为二进制格式，以便能够指定一个不同的加载地址，即 SPI-Flash 地址。详细信息请参见下图。

- SDCARD

用户应手动将二进制文件（构建的输出）复制到用于存储用户应用程序的 SDCARD 中，然后将 SDCARD 插入评估板。

下图显示了加载和调试的步骤：

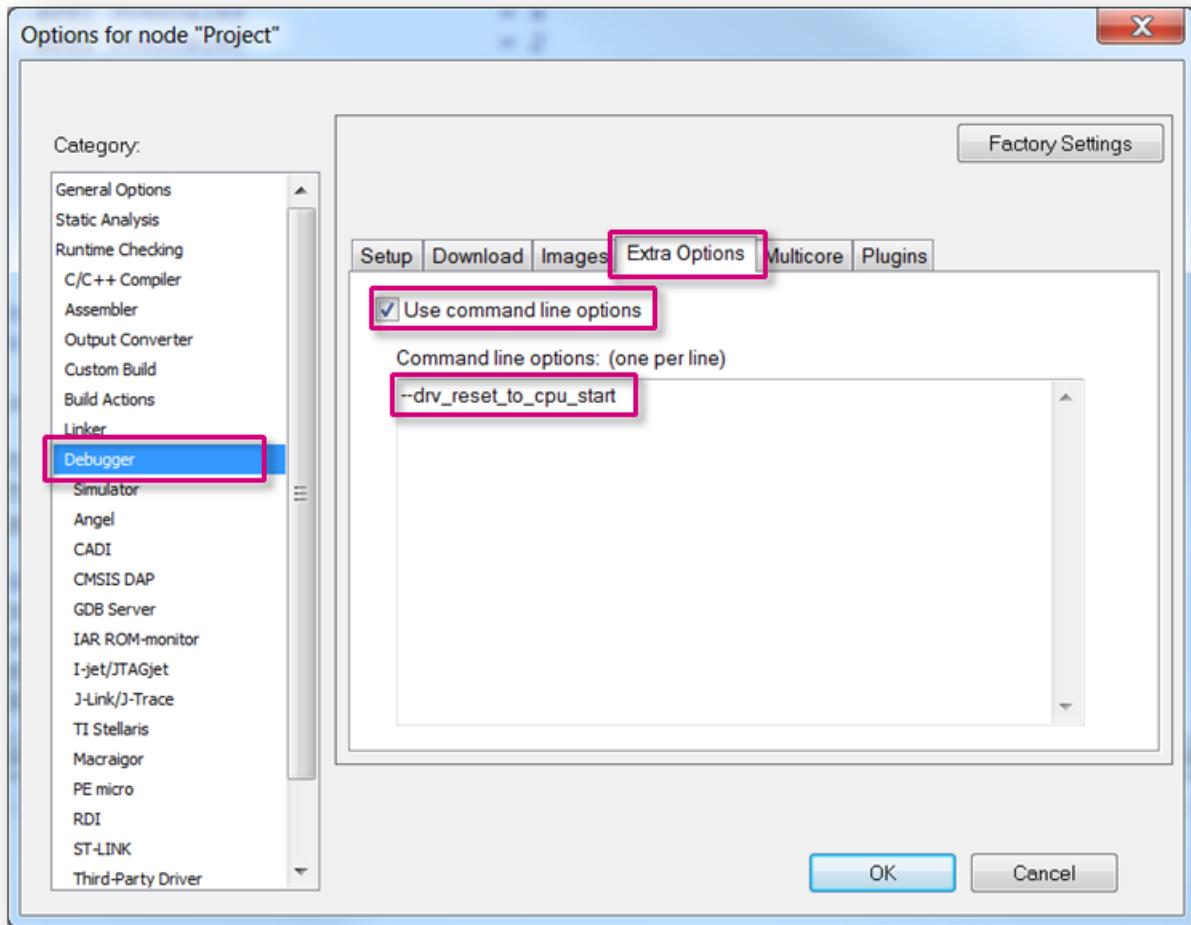


### 5.3 使用 EWARM IDE 进行调试

在调试从外部存储器运行的用户应用程序时，需要特别注意 EWARM IDE。EWARM 将 PC（程序计数器）的默认 CPU 重置值覆盖为用户应用程序中给定的值（外部执行存储器中的地址值）。

在该启动方案中，在执行外部存储器启动应用程序之前，用户应用程序 PC 地址保持不可访问（因此外部存储器已经准备好，并通过 FMC 或 QSPI 映射内存）。如果 EWARM 直接跳到用户应用程序的起始点，就会生成 hardfault。为了避免 hardfault，用户应在调试器选项中添加“`--drv_reset_to_cpu_start`”命令行，如下图所示。此设置防止 EWARM 强制 PC，并让外部存储器启动应用程序在跳到用户应用程序之前配置外部存储器。

图 6. 调试器命令行选项



## 6 性能特性

当从外部存储器执行时，由于外部闪存延迟和较长的指令/数据路径，性能会受到影响。如果使用 STM32F7x0 超值系列和 STM32H750 超值系列设备，Cortex-M7 L1-cache 可以减少这种影响。

下表总结了每个 ROM / RAM 组合取得的 EEMBC® CoreMark® 分数。当从内部闪存执行时，可以获得最佳性能。然而，当从外部存储器执行时，损失会显著减少。

这些数字说明了从外部存储器进行操作时对 CPU 性能的影响。提供内部 Flash 配置分数作为参考。

**表 5. 每种配置的 EEMBC® CoreMark® 分数**

ROM 存储器	RAM 存储器	STM32F723E-Discovery, 配置为 STM32F730 (I/D Cache 8K/8K)	STM32756G_EVAL, 配置为 STM32F750 (I/D Cache 4K/4K)	STM32H743I_EVAL, 配置为 STM32H750 (I/D Cache 16K/16K)
QSPI 闪存	内部 SRAM	1089	948	2020
	外部 SRAM	-	940	1972
	外部 SDRAM	-	871	1972
	外部 PSRAM	1079	-	-
NOR Flash	内部 SRAM	-	906	2020
	外部 SRAM	-	899	1972
	外部 SDRAM	-	833	1972
外部 SRAM	内部 SRAM	-	1016	2020
外部 SDRAM	内部 SRAM	-	989	2020
内部 Flash	内部 SRAM	1092	1082	2020

## 版本历史

表 6. 文档版本历史

日期	版本	变更
2018 年 7 月 11 日	1	初始版本。

## 目录

<b>1</b>	<b>概述</b>	<b>2</b>
<b>2</b>	<b>外部存储器代码执行概述</b>	<b>3</b>
<b>2.1</b>	<b>外部存储器代码执行原则</b>	<b>3</b>
<b>2.2</b>	<b>外部存储器启动应用程序描述</b>	<b>6</b>
<b>3</b>	<b>支持的启动模型</b>	<b>9</b>
<b>3.1</b>	<b>支持芯片内执行（XiP）</b>	<b>9</b>
<b>3.2</b>	<b>支持 BootROM</b>	<b>12</b>
<b>3.3</b>	<b>配置</b>	<b>14</b>
<b>3.4</b>	<b>外部存储器部件总结</b>	<b>14</b>
<b>4</b>	<b>需要考虑的资源约束</b>	<b>16</b>
<b>5</b>	<b>外部存储器用户应用程序描述</b>	<b>19</b>
<b>5.1</b>	<b>需要的更新</b>	<b>19</b>
<b>5.2</b>	<b>加载和调试</b>	<b>19</b>
<b>5.3</b>	<b>使用 EWARM IDE 进行调试</b>	<b>20</b>
<b>6</b>	<b>性能特性</b>	<b>22</b>
	<b>Revision history</b>	<b>23</b>

## 表一览

表 1.	应用详情 . . . . .	3
表 2.	启动模型在每个板上使用的外部存储器 . . . . .	14
表 3.	板为每个存储器分配的引脚 . . . . .	17
表 4.	内存类型所需的外设 . . . . .	18
表 5.	每种配置的 EEMBC® CoreMark® 分数 . . . . .	22
表 6.	文档版本历史 . . . . .	23



## 图一览

图 1.	外部存储器代码启动方案 . . . . .	5
图 2.	外部存储器启动应用程序源文件超集 . . . . .	8
图 3.	XIP 模型操作流程 . . . . .	11
图 4.	BootROM 模型操作流程 . . . . .	13
图 5.	STM32CubeProgrammer . . . . .	20
图 6.	调试器命令行选项 . . . . .	21

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 标志是 ST 的商标。关于 ST 商标的其他信息，请访问 [www.st.com/trademarks](http://www.st.com/trademarks)。其他所有产品或服务名称是其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2019 STMicroelectronics - 保留所有权利