

Application Note

在 TI TDA4x 上使用

单个 eMMC 以统一引导模式进行冗余引导



Johnny Kim

Jacinto 7

摘要

在汽车高级驾驶辅助系统 (ADAS) 中，系统可靠性至关重要，尤其在引导过程中。本应用手册介绍了一种方法，用于在 TI TDA4x Jacinto 器件上使用单个 eMMC 器件实现冗余引导功能。系统使用嵌入式多媒体卡 (eMMC) 引导模式作为主引导模式，并使用多媒体卡/安全磁盘 (MMC/SD) 引导模式作为备用模式。两条路径共用同一个 eMMC 硬件，而无需更改物理引导模式或使用额外的存储器件。

建议的设计可确保即使主引导映像损坏（例如由于更新期间意外断电），系统也可以成功恢复和引导。要做到这一点，需要精心组织引导加载程序映像，利用不同的引导模式，并在同一器件内准备不同的引导路径。验证方法是故意损坏主映像，并观察自动回退到备用引导路径的情况。此方法为 ADAS 应用提供了可靠、具有成本效益且方便维护的冗余引导设计，从而在现实的汽车环境中提供了系统稳健性和可用性。

内容

1 简介.....2

2 物理引导模式 DIP 开关配置.....3

 2.1 主引导模式的 eMMC 引导模式开关.....4

 2.2 用于备用引导模式的 MMC/SD 引导模式开关.....5

3 实验.....6

 3.1 为主引导模式准备 eMMC 引导映像.....6

 3.2 为备用引导模式准备 MMC/SD 引导映像.....6

 3.3 eMMC 存储器布局.....7

 3.4 验证冗余引导.....8

4 总结.....9

5 参考资料.....9

商标

所有商标均为其各自所有者的财产。

1 简介

TI 的 TDA4x Jacinto 器件支持从八线串行外设接口 (OSPI) 闪存、eMMC、通用异步接收器/发送器 (UART) 等各种介质进行引导。上电后，微控制器单元 (MCU) ROM 代码通过读取物理引导模式 (通常通过物理配置) 来启动引导序列。根据此配置，ROM 代码确定要访问哪个引导介质以找到所需的引导映像。

默认情况下，ROM 代码首先尝试从主引导介质加载映像。如果主引导映像有效并通过完整性和身份验证检查，则系统将会使用这些映像成功引导。但是，如果主引导映像丢失或损坏，则 ROM 代码会自动继续尝试备用引导介质。备用介质的引导过程遵循与主介质相同的顺序和验证步骤。这种双路径引导机制为系统提供了两次引导机会，大幅大大提高了整体可靠性和容错能力，这对于 ADAS 应用尤为重要。高级引导序列遵循的过程如下：

1. 上电，启动 MCU ROM 的引导过程。
2. MCU ROM 代码读取引导模式引脚，以识别主引导源。
3. MCU ROM 代码请求器件设备管理和安全控制器 (DMSC) 以验证引导映像的完整性。
4. 如果完整性检查通过，则系统将使用主映像进行引导。
5. 如果完整性检查失败，则 MCU ROM 代码会再次读取引导模式引脚，以识别备用引导源。
6. DMSC 对备用映像执行完整性检查。
7. 如果成功，则系统将从备用介质引导。

此过程提供了冗余引导功能，可在映像损坏或更新失败时实现稳健的恢复。

本应用手册的以下各节介绍了如何使用单个 eMMC 器件配置这种冗余引导机制，从而无需切换物理引导模式或使用多个存储器件。

2 物理引导模式 DIP 开关配置

ROM 代码会检查两组引脚，以确定引导程序过程：BOOTMODE 引脚和 MCU_BOOTMODE 引脚。这些引脚用于配置各种设置，例如 POST 配置、PLL 配置等。

但是，本应用手册特别关注以下与引导相关的配置，如相应表格表中所述：

- 主引导模式 A
- 主引导模式 B
- 主引导模式配置
- 备用引导模式
- 备用引导模式配置

表 2-1. MCU_BOOTMODE 引脚映射

9	8	7	6	5	4	3	2	1	0
POST Config		保留	仅 MCU	主引导模式 A			PLL 配置		

表 2-2. 引导模式引脚映射

7	6	5	4	3	2	1	0
备份引导模式配置	主引导模式配置			备用引导模式			主引导模式 B

在 TDA4VM EVM 上，这些引脚会物理映射到 DIP 开关，如下所示：

- BOOTMODE[0:7] → BOOTMODE SW8[1:8]
- MCU_BOOTMODE[2:9] → BOOTMODE SW9[1:8]

备注

映射配置直接发生在 TDA4 器件的通用处理器板上，可能会根据定制板设计的具体细节进行更改。

eMMC 存储器可用于 eMMC 引导模式或 MMC/SD 卡引导模式。本应用手册建议使用 eMMC 引导模式作为主引导模式，使用 MMC/SD 卡引导模式作为备用引导模式，两者均使用单个 eMMC 器件。

2.1 主引导模式的 eMMC 引导模式开关

表 2-3、表 2-4 和 表 2-6 提供了将主引导模式配置为复位后尝试的第一个引导模式的详细信息。

要将 eMMC 引导配置为主引导模式，请使用以下引脚设置：

1. 选择 eMMC 引导模式

- 主引导模式 B 引脚：BOOTMODE[0] = 0b1
- 主引导模式 A 引脚：MCU_BOOTMODE[3:5] = 0b100

表 2-3. 当仅 MCU = 0 时的主引导模式选择

主引导模式 B 引脚	主引导模式 A 引脚			已选引导模式
0	MCU 5	MCU 4	MCU 3	
1	0	0	1	eMMC

2. 配置 eMMC 引导设置，另请参阅 表 2-4 和 表 2-5

- BOOTMODE[4:6] = 0b000 → 1.8V I/O，最大总线宽度（端口 0 上为 8 位），使用端口 0

3. 仅 MCU 配置

- MCU_BOOTMODE[6] = 0b0

表 2-4. 主引导模式配置

主引导模式配置引脚			主引导模式 B 引脚	主引导模式 A 引脚			主引导模式
6	5	4	0	MCU 5	MCU 4	MCU 3	
端口	总线宽度	电压	1	0	0	1	eMMC

表 2-5. eMMC 引导配置字段

BOOTMODE 引脚	字段	值	说明
6	端口	0	端口 0
		1	端口 1
5	总线宽度	0	端口最大宽度（端口 0 上为 8
		1	位） 仅 1 位
4	电压	0	1.8V
		1	3.3V

因此，DIP 开关设置的配置方式如下：

- BOOTMODE SW8[1:8]: 1XXX_000X
- MCU_BOOTMODE SW9[1:8]: X100_0XXX

备注

“X”表示“无关项”，在此场景下不会影响配置。

2.2 用于备用引导模式的 MMC/SD 引导模式开关

表 2-6 和 表 2-7 详细介绍了当主引导模式失败或返回时，如何配置备用引导模式。

在使用备用引导模式的情况下，配置 MMC/SD 卡引导模式，而不是 eMMC 引导模式。为此，需要进行以下引脚设置：

- 使用 表 2-6 中定义的备用引导模式引脚，选择 MMC/SD 卡引导模式
 - BOOTMODE[1:3] = 0b101 → MMC/SD 卡引导模式

表 2-6. 当仅 MCU = 0 时的备用模式选择

备用引导模式引脚			所选的备用引导模式
3	2	1	
1	0	1	MMC/SD 卡

- 配置 MMC/SD 引导模式设置
 - BOOTMODE[4:5] = 0b00 → 4 位总线宽度，文件系统 (FS) 模式
 - BOOTMODE[7] = 0b0 → 选择端口 0

表 2-7. MMC/SD 引导配置字段

BOOTMODE 引脚	字段	值	说明
7	端口	0	端口 0
		1	端口 1
5	总线宽度	0	4 位
		1	1 位
4	FS/Raw	0	FS 模式
		1	原始模式

因此，MMC/SD 卡备用引导模式的 DIP 开关设置如下：

- BOOTMODE SW8[1:8]: X101_00X0
- MCU_BOOTMODE SW9[1:8]: XXXX_XXXX (与备用引导配置无关)

要使用连接到 MMC 控制器端口 0 的单个 eMMC 器件来支持主引导模式和备用引导模式，请使用以下统一开关配置：

- BOOTMODE SW8[1:8]: 1101_0000
- MCU_BOOTMODE SW9[1:8]: 0100_0000

3 实验

要验证建议的冗余引导方法，需要两组引导映像，一组用于主引导，另一组用于备用引导。用于验证的主要组件包括以下二进制文件：

- 次级引导加载程序 (SBL)
- 应用程序 (bootapp)
- TI 基础安全 (TIFS)

为简单起见，选择 QNX OS 作为高级操作系统 (OS)。要引导至 QNX 控制台，需要以下二进制文件：

- lateapp1、lateapp2
- atf_optee.appimage
- ifs_qnx.appimage

3.1 为主引导模式准备 eMMC 引导映像

以下构建命令可用于为 eMMC 引导 (作为主引导模式) 生成引导映像：

```
cd ${PDK_PATH}/packages/ti/build
make -sj6 sbl_emmc_boot0_img BOARD=j721e_evm CORE=mcu1_0
make -sj6 boot_app_mmcscd_qnx HLOSBOOT=qnx BOARD=j721e_evm CORE=mcu1_0
```

构建过程完成后，就会生成以下二进制文件：

- tiboot3.bin (SBL)
- 应用程序 (bootapp)

构建二进制文件后，按预定义的偏移量将其刷写到 eMMC boot1 分区。

```
mmc dev 0 1
mmc partconf 0 1 1 1
mmc bootbus 0 2 0 0
fatload mmc 1 ${loadaddr} tiboot3.bin
mmc write ${loadaddr} 0x0 0x400
fatload mmc 1 ${loadaddr} tifs.bin
mmc write ${loadaddr} 0x400 0x1000
fatload mmc 1 ${loadaddr} app
mmc write ${loadaddr} 0x1400 0x2000
```

tiboot3.bin 在 0x0 处刷写，tifs.bin 在 0x400 处刷写，APP 在 0x1400 处刷写。

3.2 为备用引导模式准备 MMC/SD 引导映像

要支持 MMC/SD 引导作为备用引导模式，必须准备一组单独的引导映像。使用相应的构建命令来生成所需的二进制文件。构建完成后将生成以下文件：

- tiboot3.bin (SBL)
- 应用程序 (bootapp)

```
cd ${PDK_PATH}/packages/ti/build
make -sj6 sbl_emmc_uda_img BOARD=j721e_evm CORE=mcu1_0
make -sj6 boot_app_mmcscd_qnx HLOSBOOT=qnx BOARD=j721e_evm CORE=mcu1_0
```

将这些二进制文件复制到 eMMC 用户数据区域 (UDA) 分区，因为备用引导过程会从该区域加载这些二进制文件。

```
mount /dev/mmcblk0p1 ./emmc_uda_partition
export DST_DIR=/home/root/emmc_uda_partition

cp tiboot3.bin ${DST_DIR}/
cp app ${DST_DIR}/
cp tifs.bin ${DST_DIR}/
sync
```

备注

修改用于备用引导的 tiboot3.bin，以便在 eMMC UDA 分区（而不是从 eMMC 引导分区）中正确定位 tifs.bin 和应用程序。

3.3 eMMC 存储器布局

表 3-1. 主引导模式和备用引导模式的 eMMC 布局

eMMC 布局	二进制	注释
主引导的引导区域分区 (RAW 模式)	tiboot3.bin	失调电压：0x0
	tifs.bin	失调电压：0x400
	应用	失调电压：0x1400
用于备用引导的用户数据区域 (文件系统)	tiboot3.bin	
	tifs.bin	
	应用	
用户数据区域 (文件系统)	lateapp1	主要和备用引导模式通用
	lateapp2	
	atf_optee.appimage	
	ifs_qnx.appimage	

主要和备用应用程序 (bootapps) 必须从 eMMC 用户数据区域 (UDA) 分区加载指定的二进制文件。

- lateapp1
- lateapp2
- atf_optee.appimage
- ifs_qnx.appimage

无论采用哪条引导路径，这种安排都能提供一致的运行时环境。

3.4 验证冗余引导

tiboot3.bin 的主引导版本和备用引导版本在 *bootapp* 二进制文件的位置上略有不同。为了验证冗余引导机制，在 *tiboot3.bin* 的每个版本中都插入了唯一的日志消息，以区分主引导模式和备用引导模式。

如这些示例日志输出中所示，系统从 eMMC 引导分区上的主引导模式引导。

```
SBL Revision: 01.00.10.01 (Jun 9 2024 - 13:16:51) from eMMC boot partition #1
TIFS ver: 9.2.4--v09.02.04 (Kool Koala)Starting Sciserver..... PASSED
BOOT_APP (Jun 9 2024 - 13:16:54) from eMMC boot partition #1 in boot_app_main.c
MCU R5F App started at 7539 usecs
Loading BootImage
:
```

要模拟故障并触发冗余引导，可使用以下命令来损坏 eMMC 引导分区中的 *tiboot3.bin*：

```
mmc dev 0 1
mmc partconf 0 1 1 1
mmc bootbus 0 2 0 0
mw ${loadaddr} 0x00 0x1000
mmc write ${loadaddr} 0x0 0x400
```

重新引导系统后，ROM 代码会检测损坏的主映像，并自动切换到备用引导模式。此行为可通过与备用 *tiboot3.bin* 关联的唯一日志输出进行确认。

```
SBL Revision: 01.00.10.01 (Jun 9 2024 - 13:46:16) from eMMC UDA partition
TIFS ver: 9.2.4--v09.02.04 (Kool Koala)Starting Sciserver..... PASSED
BOOT_APP (Jun 9 2024 - 13:46:18) from eMMC UDA partition in boot_app_main.c
MCU R5F App started at 7193 usecs
Loading BootImage
:
```

4 总结

本应用手册演示了一种在 TI TDA4x Jacinto 器件上使用单个 eMMC 器件实现冗余引导的强大而实用的方法。系统配置为 eMMC 引导作为主模式，MMC/SD 引导作为备用模式，从而提供引导时的可靠性，无需更改硬件或使用多个存储器器件。

此方法的主要优势包括：

- 单个物理引导模式配置，以支持主引导路径和备用引导路径
- 用于主引导和备用引导的共享 eMMC 硬件接口
- 在损坏或更新失败时无缝回退到备用映像

通过映像准备、正确的引导加载程序配置以及通过受控映像损坏进行验证，这种方法经证明是一种可靠的设计，适用于对弹性引导行为要求极高的汽车系统。这种设计适用于生产环境，可提高系统可用性，减少现场维护工作量，并增强 ADAS 应用的整体安全性和稳健性。

5 参考资料

1. 德州仪器 (TI)，[TDA4VM 产品页面](#)
2. 德州仪器 (TI)，[《J721E DRA829/TDA4VM 处理器硅片版本 2.0，1.1 技术参考手册》](#)
3. 德州仪器 (TI)，[J721EXCPXEVm：用于 Jacinto™ 7 处理器的通用处理器板](#)
4. 德州仪器 (TI)，[PROCESSOR-SDK-RTOS-J721E](#) 和 [PROCESSOR-SDK-QNX-J721E](#)。

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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
版权所有 © 2025，德州仪器 (TI) 公司