User's Guide

MSPM33 引导加载程序

TEXAS INSTRUMENTS

摘要

MSPM33 引导加载程序（也称为 BSL）提供了一种通过多个串行接口（包括 UART 和 I2C）对器件存储器（闪存）进行编程和验证的方法。

内容

商标

所有商标均为其各自所有者的财产。

1 简介

1.1 BSL 特性概览

引导加载程序 (BSL) 提供了一种通过标准 UART 或 I2C 串行接口对器件存储器进行编程或验证的方法。

可通过串行接口访问的 BSL 主要特性包括：

- 闪存的编程和擦除
- 可以返回代码或数据区域的 32 位 CRC（最小区域大小为 1KB）以验证编程
- 可以启用代码或数据读出（默认禁用）
- 可以通过指向主闪存的指针返回固件版本号
- 可以指定硬件调用 GPIO
- 访问始终受到 256 位密码的保护
- 可配置的安全警报处理，用于抵抗蛮力攻击
- 基于 ROM 的接口插件，支持 UART、I2C 并具备自动检测功能
- UART 波特率配置，具有多个选项
- UART 和 I2C 的可配置接口引脚

1.2 术语

引导加载程序 (BSL) - 用于将数据加载到器件存储器的引导例程

引导代码 (BCR) - BOOTRST 之后运行的启动例程，用于配置器件以执行应用程序

BCR 配置 - 包含用于引导代码的所有用户可配置参数的配置结构，位于非主闪存中

BSL 配置 - 包含用于引导加载程序的所有用户可配置参数的配置结构，位于非主闪存中

1.3 其他资源

1. 技术参考手册
   a. MSPM33 C 系列微控制器

2 BSL 架构

2.1 设计

当检测到有效的引导加载程序调用条件时，引导代码将调用引导加载程序。仅当在 BCR 配置的 BSL 模式字段中启用了引导加载程序时，才会调用它。

引导加载程序启动后，它首先执行“Init”阶段，在该阶段完成 BSL 配置的初始检查，并将器件配置为运行引导加载程序。

接下来，引导加载程序进入“接口自动检测”阶段。在此阶段，BSL 会配置所有可用的 BSL 接口（如果已注册）。然后，BSL 逐个轮询所有接口的数据。当在其中一个接口中接收到有效的连接数据包时，该接口将被视为用于进一步通信的有源接口，所有其他接口都将被禁用。接口搜索将持续 4 秒，如果未检测到接口，则器件将进入待机模式。

接下来，BSL 进入“命令接收”阶段。在此阶段，BSL 将无限循环等待来自主机的命令。接收到有效命令后，将处理该命令，并将来自 BSL 内核的响应发送回主机。然后，它返回到循环并等待下一条命令，依此类推。如果接收到“Start Application”命令，引导加载程序将触发系统复位，然后执行引导代码并调用应用程序。该阶段的超时也为 4 秒。如果未接收到有效命令，引导加载程序将被锁定，并进入睡眠模式。

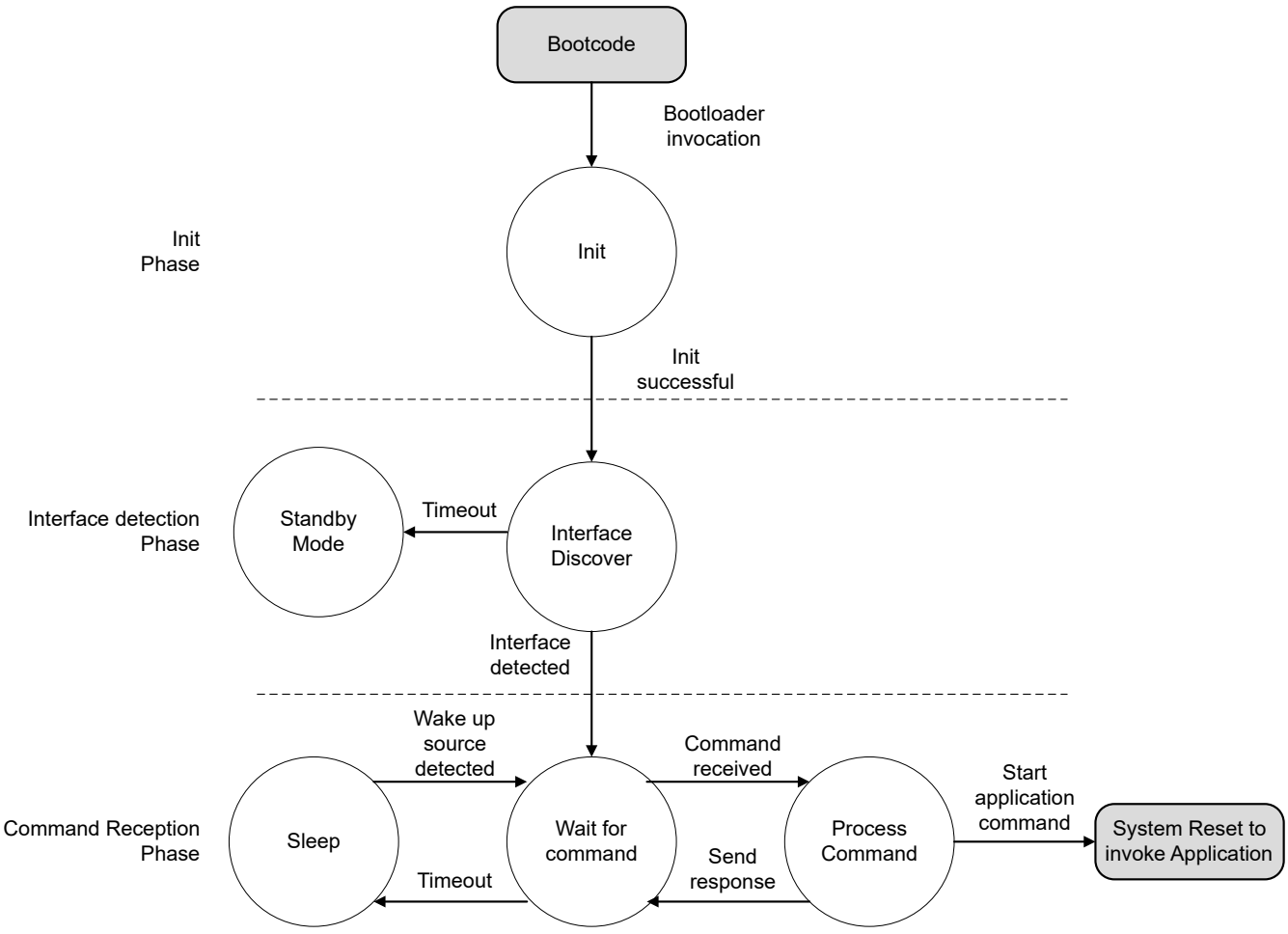


图 2-1. BSL 架构

2.1.1 超时特性

引导加载程序将在未检测到任何活动时超时，并进入低功耗模式以节省功耗。

这已在以下两个阶段实施。

1. 接口自动检测

2. 命令接收

2.1.1.1 接口自动检测

在接口检测阶段，如果在任何接口上持续 4 秒未收到有效的连接命令，引导加载程序将进入待机模式。

需要 POR 才能退出此状态，并通过再次创建 BSL 调用条件来使用引导加载程序。

2.1.1.2 命令接收

在命令接收阶段，如果 4 秒内未收到有效命令，引导加载程序将进入睡眠模式。要将器件从睡眠模式唤醒，应在有源接口上进行数据传输。

引导加载程序将在进入睡眠模式之前锁定，以减少攻击面。因此，从低功耗模式唤醒后，需要通过发送 256 位 BSL 密码来再次解锁引导加载程序（请参阅解锁引导加载程序命令）。

2.2 调用 BSL

只有在满足任何 BSL 调用条件并且在 BCR 配置中启用了引导加载程序时，引导加载程序才应由引导代码调用。

在 BCR 配置中启用快速引导模式后，引导加载程序只能由调试邮箱命令和应用程序请求调用。跳过其他检查以节省执行时间。

在 **HS-FS（高安全性 – 现场安全）**状态下，启用基于 GPIO 的调用、应用请求和调试邮箱。在 **HS-KP（高安全性 – 密钥配置）**状态下，仅启用**基于 GPIO 的调用**；禁用应用请求和调试邮箱调用，以在密钥配置期间保持安全性。一旦器件转换到 **HS-SE（高安全性 – 强制安全）**，引导加载程序就会完全禁用，并且无法调用 BSL，从而确保尽可能地保护客户代码。

2.2.1 应用程序请求

要从应用程序调用引导加载程序，请将 RESETLEVEL 设置为 BOOTLOADERENTRY 并通过 RESETCMD 寄存器触发复位。该序列会导致系统复位，并执行引导代码和调用引导加载程序。

由于系统复位已发出，因此在退出应用程序时所有外设配置均会复位。

2.2.2 基于 GPIO 的调用

用于 BSL 调用的 GPIO 可在非主闪存的 BSL 配置中进行配置。

全新器件将在 BSL 配置中具有 TI 编程的默认引脚详细信息。

可以在 BCR 配置中禁用基于 GPIO 引脚的调用。默认处于启用状态。

GPIO 应在 POR 前置为有效，并且此状态在 POR 后应至少保持 T_start ms。然后，可以取消置位 GPIO 引脚状态。

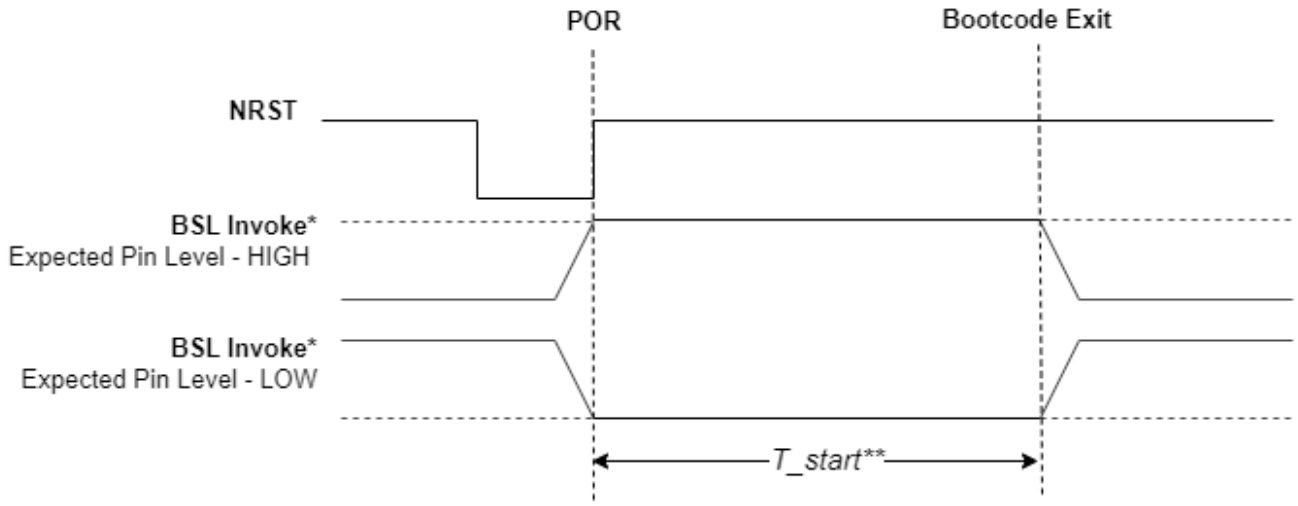


图 2-2. 从 GPIO 调用

** - 可在 BSL 配置中配置用作“BSL Invoke”和“Expected Pin Level”的 GPIO 引脚*

*** - T_start 是指器件特定数据表中指定的冷启动时间*

备注

如果启用了基于引脚的 BSL 调用，则应将所配置的 GPIO 引脚拉高或拉低。它不应保持悬空，否则可能会导致意外的 BSL 执行。

2.2.3 调试邮箱命令

当调试接口可用时，引导加载程序调用命令可通过调试子系统邮箱 (DSSM) 发送。

有关 DSSM 命令用法的更多详细信息，请参阅 MSPM33C_Technical_Reference_Manual DSSM 命令部分。

2.2.4 其他 BSL 调用方法

2.2.4.1 启动前应用程序验证

在 BCR 配置中，如果启用了应用程序 CRC 验证，则引导代码将验证给定应用程序存储器范围的 CRC。如果 CRC 不正确，则调用引导加载程序。在这种情况下，仅当启用引导加载程序时才会调用它。否则，引导代码会记录灾难性错误并导致引导失败。

2.2.4.2 空白器件处理

与以前的器件不同，当检测到空白器件（空闪存）时，引导代码不会自动调用引导加载程序。相反，它会将 while (1) 循环复制到 SRAM，并等待调试器在 10 秒内连接。如果在此时间范围内未建立调试器连接，则器件进入低功耗模式。在使用空白器件时，此方法可带来更灵活的开发体验。

2.3 存储器

2.3.1 SRAM 存储器使用情况

MSPM33 BSL 根据器件中的可用 SRAM 大小动态分配 SRAM 存储器。SRAM 存储器布局说明了用于引导加载程序运行的存储器。

SRAM 存储器布局说明了用于引导加载程序运行的存储器。

- 数据段和栈段 - 供 BSL 运行使用。在退出引导加载程序时，SRAM 的这些段会被清除。
- 可变缓冲区空间 - 用于存储 BSL 通信期间所接收/发送数据包的缓冲区空间

SRAM 存储器分配遵循以下结构：

1. BSL 缓冲区起始地址：计算为用于 BootROM 操作的 SRAM 数据段的结束地址，与下一个 8 字节边界对齐。
2. BSL 缓冲区结束地址：计算值为 RAM 结束地址减去栈大小。
3. 可用缓冲区空间：BSL 缓冲区起始地址和 BSL 缓冲区结束地址之间的空间分为两个相等的部分：
   a. RX 缓冲区：用于接收数据包，从 BSL 缓冲区起始地址开始
   b. TX 缓冲区：用于传输数据包，从 BSL 缓冲区起始地址 + 缓冲区大小开始

由于 BSL 协议有 2 个字节用于定义长度，因此最大缓冲区大小限制为 32KB（0x7FFF 字节）。尽管 MSPM33 拥有 256KB 的 SRAM，但每个缓冲区（RX 和 TX）仅使用 32KB。

主机允许进行读写访问的 SRAM 存储器是 BSL 缓冲区起始地址到 [SRAM 结束地址 – 栈大小]，其中 SRAM 结束地址由每个器件中可用的 SRAM 存储器决定。由于与可变缓冲空间共享同一 SRAM 空间，因此其内容在 SRAM 写入/读取操作期间有可能被覆盖。

2.4 BSL 非主配置

非主闪存存储器中的 BSL 配置允许用户自定义 BSL 使用的某些参数。该配置存储在从偏移 0x1C00 开始的非主闪存中。

2.4.1 BSL 配置 ID

BSL_CONFIG_ID 是预先确定的签名 ID，用于将其标识为引导加载程序配置结构。这个 32 位字段用作 BSL 配置的标识符。

Usage: 此字段将该结构标识为 BSL 配置结构。它是一个固定值，用户不应修改。

在获取器件信息中返回：BSL_CONFIG_ID 包含在对获取器件信息命令 (CMD_GET_IDENTITY, 0x19) 的响应中。这让主机能够识别 BSL 配置的版本或类型。

2.4.2 BSL 接口引脚 (BLINTERFACE_PINS)

BLINTERFACE_PINS 结构为 BSL 通信接口（UART 和 I2C ）配置物理引脚和引脚功能多路复用器选择。此配置允许 BSL 使用特定引脚与主机通信。

UART 接口引脚 (0x80101C04)

1. UART 接收引脚编号（位 7...0）
   a. 通过 UART 接收数据的物理引脚编号
2. UART 接收引脚功能（位 15...8）
   a. 接收引脚的功能多路复用器选择
3. UART 发送引脚编号（位 23...16）
   a. 通过 UART 发送数据的物理引脚编号
4. UART 发送引脚功能（位 31...24）
   a. 发送引脚的功能多路复用器选择

I2C 接口引脚 (0x80101C08)

1. I2C 数据引脚编号（位 7...0）
   a. I2C 数据线 (SDA) 的物理引脚编号
2. I2C 数据引脚功能（位 15...8）
   a. 数据引脚的功能多路复用器选择
3. I2C 时钟引脚编号（位 23...16）

a. I2C 时钟线 (SCL) 的物理引脚编号

4. I2C 时钟引脚功能 (位 31...24)
   a. 时钟引脚的功能多路复用器选择

保留引脚信息 (0x80101C0C)

2.4.3 BSL 调用引脚配置 (BSLPIN_INVOKE)

BSLPIN_INVOKE 结构会配置用于触发 BSL 调用的 GPIO 引脚。这允许通过在器件启动期间将特定引脚设置为特定状态来调用 BSL。

1. 引脚配置和触发电平 (位 7...0)
   a. 定义了 BSL 调用的引脚配置和预期触发电平
   b. 位字段
      i. PINCM 索引 (位 6...0) ：指定引脚配置模式索引
      ii. 触发电平 (位 7) ：指定触发 BSL 调用的引脚状态
         1. 0 = 当引脚为低电平时调用 BSL
         2. 1 = 当引脚为高电平时调用 BSL
      iii. 默认值：0xA8
         1. PINCM 索引 = 0x28 (40)
         2. 触发电平 = 1 (高电平)

2. 引脚选择 (位 15...8)
   a. 定义用于 BSL 调用的物理引脚
   b. 位字段
      i. 引脚编号 (位 4...0) ：指定所选端口内的哪个引脚编号
      ii. 端口索引 (位 6...5) ：指定引脚属于哪个 GPIO 端口
      iii. 保留 (位 7) ：在当前实现中未使用
   c. 默认值：0x12
      i. 引脚编号 = 0x12 (18)
      ii. 端口索引 = 0 (端口 A)

BSL 调用引脚配置可使器件在器件启动期间将特定 GPIO 引脚设置为特定状态时进入 BSL 模式。默认情况下，当端口 A 的引脚 18 在器件启动期间设置为高电平时，调用 BSL。

使用配置的引脚调用 BSL：

1. 将指定引脚 (默认值：端口 A 的引脚 18) 设置为指定状态 (默认值：高电平)
2. 打开器件电源或执行复位
3. 在上电复位后，将引脚状态保持至少 T_start 毫秒
4. 在 T_start 毫秒之后可以将引脚状态置为无效

备注

如果启用了基于引脚的 BSL 调用，则应根据配置将所配置的 GPIO 引脚拉高或拉低。它不应保持悬空，否则可能会导致意外的 BSL 执行。

2.4.4 存储器读取配置

READOUT 字段控制 BSL 是否允许存储器读取操作。启用后，存储器读回命令可用于读取闪存和 SRAM 存储器的内容。禁用后，存储器读回命令将返回错误。

默认值： BL_CFG_READBACK_EN (0xAABB)

有效值：

1. BL_CFG_READBACK_EN (0xAABB)：启用存储器读取
2. 所有其他值：禁用存储器读取

出于安全原因，默认情况下禁用存储器读取。只有在开发环境中或特别需要读取存储器时，才应启用该功能。

即使启用了存储器读取，仍然必须先使用正确的密码解锁 BSL，然后才能使用存储器读回命令。这提供了额外的安全层，可防止未经授权访问存储器内容。

2.4.5 BSL 密码

存储器地址：0x80101C14 - 0x80101C30

PASSWORD 字段存储用于 BSL 身份验证的密码的 SHA-256 哈希值（32 字节）。解锁引导加载程序和访问受保护的 BSL 命令需要此密码。

默认值：所有 0xFF 字节的字符串哈希值

1. 字 0 (0x80101C14)：0x761396AF
2. 字 1 (0x80101C18)：0x5F63720F
3. 字 2 (0x80101C1C)：0x5A4AB4BD
4. 字 3 (0x80101C20)：0x9FC3630A
5. 字 4 (0x80101C24)：0xF930AF12
6. 字 5 (0x80101C28)：0x5CEEA650
7. 字 6 (0x80101C2C)：0x88E11B97
8. 字 7 (0x80101C30)：0x51409CE8

用途

BSL 密码用于保护对可修改器件存储器或读取敏感信息的关键 BSL 命令的访问。调用 BSL 时，它会在锁定状态下启动。要解锁 BSL 并访问受保护的命令，主机必须使用正确的密码发送解锁引导加载程序命令 (CMD_UNLOCK_BSL, 0x21)。

密码身份验证过程的工作方式如下：

1. 主机发送使用了 32 字节密码的解锁引导加载程序命令。
2. BSL 计算接收到的密码的 SHA-256 哈希值。
3. BSL 将计算出的哈希值与 BSL 配置中存储的密码哈希值进行比较。
4. 如果哈希值匹配，BSL 将被解锁，受保护的命令变为可用。
5. 如果哈希值不匹配，BSL 保持锁定状态并返回 BSL_PASSWORD_ERROR。

BSL 包含几项安全功能，用于防范密码攻击：

1. **睡眠延迟**：如果发送了错误的密码，器件将进入睡眠模式 2 秒，并且在此期间不接受任何命令。这会使暴力攻击更加耗时。
2. **安全警报**：如果发送了 3 次错误的密码，则会根据 SECURITY_ALERT_LEVEL 配置采取安全警报措施。这可能包括：
   a. Factory Reset：擦除所有闪存
   b. 禁用引导加载程序：永久禁用引导加载程序
   c. 不执行任何操作：不采取任何操作
3. **密码备份**：在执行恢复出厂设置之前，BSL 会将密码备份到 SRAM。这样，即使非主闪存中的密码字段被擦除，也可以在恢复出厂设置后解锁 BSL。

更改 BSL 密码

可以通过多种方法更改 BSL 密码

1. **使用 BSL 命令**
   a. 使用当前密码解锁 BSL。
   b. 计算新密码的 SHA-256 哈希值。
   c. 使用新哈希值更新 BSL 配置中的 PASSWORD 字段。
   d. 计算 BSL 配置的新 CRC。
   e. 使用编程数据命令将更新后的 BSL 配置编程到非主闪存。
2. **使用 SysConfig 和 Code Composer Studio (CCS)**SysConfig 工具提供了非主配置器，可以生成完整的非主配置。如果非主闪存区域不受保护，您可以使用 Code Composer Studio 中的闪存加载程序直接修改 BSL 密码。
   a. 在 CCS 中或作为独立应用程序打开 SysConfig 工具。
   b. 导航至 Non-Main Configurator（非主配置器）部分。
   c. 配置所有 BSL 设置，其中 PASSWORD 字段需填入新密码的哈希值。
   d. 生成非主配置二进制文件。
   e. 使用 CCS 闪存加载程序或 BSL 命令将生成的二进制文件编程到非主闪存区域。

备注

如果忘记 BSL 密码，可能需要执行 DSSM 恢复出厂设置操作来恢复默认密码。但是，这将会擦除所有闪存，包括应用程序代码和配置数据。或者，如果非主闪存区域不受保护，您可以使用带有闪存加载程序的 CCS，通过新密码对 BSL 配置进行重新编程。

2.4.6 应用程序修订指针

存储器地址：0x80101C34

APP_REV_POINTER 字段包含指向存储在主闪存中的应用程序版本信息的指针。这样 BSL 就可以在主机发出请求时检索和报告应用程序版本。

默认值：0xFFFFFFFF（表示没有可用的应用程序版本信息）

Usage:

当执行获取器件信息命令 (CMD_GET_IDENTITY, 0x19) 时，BSL 使用 APP_REV_POINTER 定位和检索应用程序版本信息。这使得主机可以查询器件上当前安装的应用程序的版本，而无需加载和执行应用程序。

执行获取器件信息命令时，BSL 会执行以下检查：
1. 验证 APP_REV_POINTER 包含有效的闪存地址
2. 检查该地址是否为空（不是 0xFFFFFFFF）
3. 确保该地址为 64 位对齐
4. 如果所有检查都通过，则从指定地址读取一个 8 字节（64 位）值作为应用程序版本

如果这些检查中的任何一项未通过，BSL 将返回 0 作为应用程序版本。

实现要求：

APP_REV_POINTER 字段由客户维护，需要执行以下实现步骤：
1. **在应用程序链接器文件中分配一个段：**客户必须在应用程序链接器文件中分配专用的段来存储应用程序版本信息。该段应为 64 位对齐，并位于主闪存中。
2. **定义应用程序版本：**应用程序版本应定义为应用程序代码中的 64 位（8 字节）值。这可以是一个结构化值，其中包含主要版本、次要版本和补丁版本号，或者适合 64 位的任何其他格式。
3. **将版本放在分配的段中：**应用程序版本值应在应用程序构建过程中放在分配的段中。
4. **更新 APP_REV_POINTER：**所分配段的地址必须存储在 BSL 配置的 APP_REV_POINTER 字段中。这可以使用“更改 BSL 配置”部分中所述的方法完成。

备注

APP_REV_POINTER 是可选特性。如果未使用，则应将其保留为默认值 0xFFFFFFFF，表示没有可用的应用程序版本信息。

2.4.7 安全警报级别

存储器地址：0x80101C38

位：15...0

说明：SECURITY_ALERT_LEVEL 字段定义检测到安全违规时要执行的操作，例如检测到多次密码尝试失败。

默认值：0xAABB (恢复出厂设置)

有效值：

1. BL_CFG_SECURITY_FACTORY_RESET (0xAABB)：发生安全违规事件时执行恢复出厂设置
2. BL_CFG_SECURITY_DISABLE_BSL (0xCCDD)：发生安全违规事件时禁用引导加载程序
3. 所有其他值：发生安全违规事件时不采取任何措施

备注

只有在连续 3 次尝试错误密码后才会检查 SECURITY_ALERT_LEVEL。无论此配置如何，对于第一次和第二次错误尝试，器件都会进入睡眠模式 2 秒。

2.4.8 UART 波特率

存储器地址：0x80101C38

位：31..16

UART_BAUD_RATE 字段指定 BSL 中用于 UART 通信的默认波特率。这决定了使用 UART 接口进行 BSL 操作时的通信速度。

有效值：

1. BL_CFG_UART_BAUDRATE_4800 (0x1)：4800 位/秒
2. BL_CFG_UART_BAUDRATE_9600 (0x2)：9600 位/秒
3. BL_CFG_UART_BAUDRATE_19200 (0x3)：19200 位/秒
4. BL_CFG_UART_BAUDRATE_38400 (0x4)：38400 位/秒
5. BL_CFG_UART_BAUDRATE_57600 (0x5)：57600 位/秒
6. BL_CFG_UART_BAUDRATE_115200 (0x6)：115200 位/秒
7. BL_CFG_UART_BAUDRATE_1000000 (0x7)：1,000,000 位/秒
8. BL_CFG_UART_BAUDRATE_2000000 (0x8)：2,000,000 位/秒
9. BL_CFG_UART_BAUDRATE_3000000 (0x9)：3,000,000 位/秒

更改波特率：

有两种方法可以更改 UART 波特率：

1. 永久更改：使用“更改 BSL 配置”部分中所述的方法修改 BSL 配置中的 UART_BAUD_RATE 字段。这会更改调用 BSL 时使用的默认波特率。
2. 运行时更改：在 BSL 会话期间使用更改波特率命令 (CMD_CHANGE_BAUD_RATE) 临时更改波特率。此更改仅对当前 BSL 会话有效，在器件复位时恢复为默认值。

备注

使用 UART 接口进行 BSL 通信时，主机和器件必须配置为使用相同的波特率。如果使用更改波特率命令更改波特率，则主机还必须更改其波特率以进行匹配。

2.4.9 I2C 从器件地址

存储器地址：0x80101C3C

I2C_SLAVE_ADDR 字段指定在 I2C 模式下运行时 BSL 使用的从器件地址。外部器件必须使用该地址通过 I2C 接口与 BSL 通信。

位：15...0

默认值：0x48

I2C_SLAVE_ADDR 字段设置 BSL 在通过 I2C 接口进行通信时使用的从器件地址。当调用 BSL 并选择 I2C 接口时，它会配置 I2C 外设以响应该从器件地址。

2.4.10 配置 CRC

存储器地址：0x80101C4C

CRC 字段包含一个 32 位循环冗余校验值，该值基于完整的 BSL 配置结构计算得出（CRC 字段自身除外）。该值用于验证配置数据的完整性。

默认值：0xB4808AA4（根据默认配置值计算得出）

有效值：基于 BSL 配置的内容计算得出的值

CRC 字段有几个重要用途：

1. 数据完整性验证：确保 BSL 配置数据没有损坏或被意外修改。
2. 配置验证：确认 BSL 配置结构有效且完整。
3. 防错机制：有助于防止 BSL 使用可能导致意外行为的错误配置值。

每当修改 BSL 配置中的任何字段时，都必须重新计算 CRC。这包括：

1. 初始配置：首次创建或编程 BSL 配置时。
2. 配置更新：更改配置中的任何字段时。
3. Factory Reset：恢复出厂设置后，会针对默认配置计算新的 CRC。

BSL 配置中不正确的 CRC 将会造成永久性损失，因为它会导致不可恢复的灾难性错误。

备注

使用 SysConfig 或 CCS 等工具修改 BSL 配置时，通常会自动计算和更新 CRC。但是，当使用 BSL 命令或其他方法手动修改配置时，必须确保正确地重新计算 CRC，以避免永久损坏器件。

2.5 更改 BSL 配置

存储在非主闪存中的 BSL 配置可使用多种方法进行修改。本节介绍了更新 BSL 配置参数的可用方法。

2.5.1 参考

必要条件：

必须在 BCR 配置中启用恢复出厂设置（“Enabled”或“Enabled with Password”）

步骤：

1. 使用支持的调用方法之一（GPIO、应用程序请求、调试邮箱或 SFI）调用引导加载程序
2. 使用连接命令 (CMD_CONNECTION, 0x12) 建立连接
3. 使用解锁引导加载程序命令 (CMD_UNLOCK_BSL, 0x21) 通过当前密码解锁 BSL
4. 执行恢复出厂设置命令 (CMD_FACTORY_RESET, 0x30) 以擦除非主闪存配置
   a. 这将擦除主闪存（应用程序存储器）和非主闪存（配置存储器）
   b. 如果 BCR 配置有要求，则向生产工厂提供重置密码（默认值：全 0xFF）
   c. 恢复出厂设置后，BSL 密码恢复为默认值（全 0xFF 的哈希值）

5. 为 BSL 配置准备新的 BSL 配置结构并计算 CRC32：
   a. 针对完整配置结构计算 CRC32-ISO3309（CRC 字段自身除外）
   b. 使用位反转配置，初始种子值为 0xFFFFFFFF
   c. 配置结构大小：76 个字节（从 0x80101C00 到 0x80101C4B）
   d. 将计算出的 CRC32 值存储在 CRC 字段中 (0x80101C4C)
6. 使用编程数据命令 (CMD_PROGRAM_DATA, 0x20) 将新 BSL 配置编程到非主闪存：
   a. Start Address：0x80101C00
   b. Data: 完整 BSL 配置结构（总共 80 字节，包括 CRC）
   c. 确保数据按照编程数据命令的要求进行 16 字节对齐
7. 复位器件以应用新配置
8. 注意：每当修改任何配置字段时，都必须重新计算 CRC 字段 • 不正确的 CRC 将会导致灾难性错误和器件永久锁定 • 如果在 BCR 配置中未启用恢复出厂设置功能，非主闪存编程将会失败 • 确保对完整配置结构进行编程，而非仅对单个字段进行编程

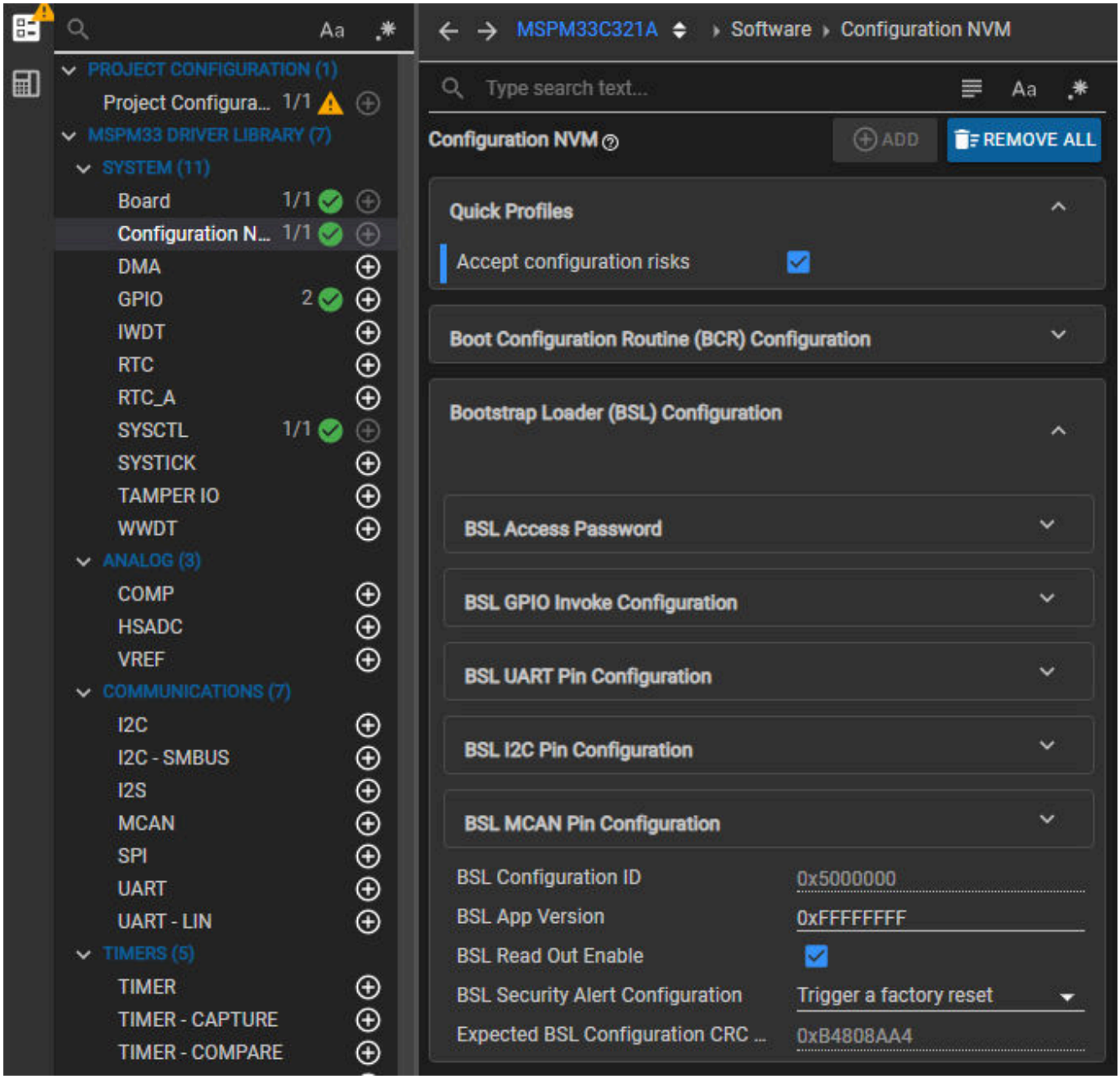
备注

- 每当修改任何配置字段时，都必须重新计算 CRC 字段
- 不正确的 CRC 将会导致灾难性错误和器件永久锁定
- 如果在 BCR 配置中未启用恢复出厂设置功能，非主闪存编程将会失败
- 确保对完整配置结构进行编程，而非仅对单个字段进行编程

2.5.2 使用 SysConfig 和 Code Composer Studio (CCS)

SysConfig 工具提供了用于配置 BSL 参数的图形界面：

1. 在 Code Composer Studio 中或作为独立应用程序打开 SysConfig



2. 导航至 Non-Main Configurator（非主配置器）部分
3. 配置所需的 BSL 参数：
   a. 接口引脚（UART、I2C）
   b. BSL 调用引脚配置
   c. 存储器读取启用/禁用
   d. BSL 密码：输入预先计算的 32 字节 SHA-256 哈希值（不是明文密码）
   e. 应用程序修订指针
   f. UART 波特率
   g. I2C 从器件地址
4. 该工具会根据所有配置的参数自动计算和更新 CRC 字段
5. 生成非主配置二进制文件
6. 使用下列方法之一将生成的二进制文件编程到非主闪存：

a. CCS 闪存加载程序（如果非主闪存未启用写保护且可进行调试访问）
b. BSL 编程数据命令（如果在 BCR 配置中启用了恢复出厂设置）
c. 调试接口编程工具

备注

- SysConfig 不会计算 SHA-256 密码哈希值 - 您必须提供预先计算的哈希值
- 该工具仅计算配置结构完整性的 CRC32
- 保存明文密码及其 SHA-256 哈希值的安全记录

2.5.3 使用调试接口

如果非主闪存区域未启用写保护，可以使用调试探针直接对 BSL 配置进行编程：

1. 将调试探针（XDS110 等）连接到器件
2. 使用 Code Composer Studio 闪存加载程序或类似工具
3. 加载 BSL 配置二进制文件
4. 从地址 0x80101C00 开始编程到非主闪存

3 引导加载程序协议

3.1 数据包格式

BSL 数据包具有分层结构。BSL 内核命令包含由 BSL 待处理的实际命令数据。除了标准 BSL 命令，每个内核命令的前后有被称为外设接口代码（PI 代码）的包装器数据。该包装数据是所使用的外设和协议的特定信息，并且它包含有允许 BSL 内核命令正确传输的信息。包装器和内核命令将构成一个 BSL 数据包

PI 代码	BSL 内核数据	PI 代码

3.2 BSL 协议

UART 和 I2C BSL 协议的数据包具有以下结构。

- 标头字节表示使用的协议和数据包类型（命令或响应数据包）。
- 长度字段包含以字节为单位的 BSL 内核数据的大小。
- BSL 内核数据，包含命令/响应 ID 和地址，即命令所需的数据
- CRC32 域包含针对 BSL 内核数据中的数据计算的 CRC

PI 代码		BSL 内核数据	PI 代码
标头（1 字节）	长度（2 字节）	BSL 内核命令/响应	CRC32（4 字节）

根据内核数据字段，数据包被归类为命令数据包或响应数据包。

命令数据包是发送到 BSL 的第一个数据包。第二个数据包是从 BSL 接收的响应数据包。响应数据包包含两个组件 BSL 确认和 BSL 内核响应。在这两个命令中，每个发送的命令数据包都会从 BSL 接收确认。但并非每个命令都会收到 BSL 内核响应。

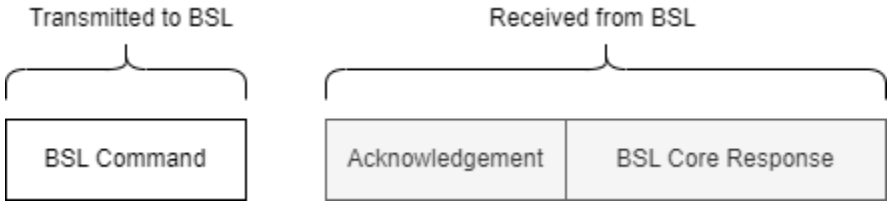


图 3-1. BSL 协议

3.2.1 BSL 确认

BSL 软件的外设接口部分对 BSL 数据包的包装器部分进行解析。如果数据传输出现错误，则会立即发送错误消息。在成功接收到所有数据后发送 ACK，这并不意味着命令已经正确执行（或者甚至说命令是有效的），而是意味着数据包被正确格式化并传递给 BSL 内核软件用于解释。

BSL 协议规定，每个发送的 BSL 数据包除了已发送的 BSL 数据包外，以单字节确认进行响应。表列出了来自 BSL 的确认响应。如果发送了除 ACK 以外的确认字节，则 BSL 不发送任何 BSL 数据包。主机编程器必须检查确认错误并重新尝试发送。

数据	含义
0x00	BSL_ACK（成功接收到数据包）
0x51	BSL_ERROR_HEADER_INCORRECT
0x52	BSL_ERROR_CHECKSUM_INCORRECT
0x53	BSL_ERROR_PACKET_SIZE_ZERO
0x54	BSL_ERROR_PACKET_SIZE_TOO_BIG
0x55	BSL_ERROR_UNKNOWN_ERROR
0x56	BSL_ERROR_UNKNOWN_BAUD_RATE

3.2.2 外设配置

3.2.2.1 UART

通过以下配置启用 UART：

- 使用 UC12
- 波特率默认为 9600bps。它可以通过 Change Baud Rate 命令进行更新
- 数据宽度：8 位
- 停止位数：1
- 无奇偶校验位
- 用于 RXD 和 TXD 的引脚取自 BSL 配置

3.2.2.2 I2C

BSL 中的 I2C 接口可用作 I2C 目标。主机充当控制器并驱动通信。

- 使用 UC15_0
- 默认情况下，I2C 目标地址为 0x48。它可以在 BSL 配置中进行配置
- SCL 和 SDA 线路需要外部上拉
- 用于 SDA 和 SCL 的引脚取自 BSL 配置

3.2.2.3 CRC

数据的 CRC 必须根据以下各项计算：

- CRC32-ISO3309 多项式
- 位反转配置
- 初始种子 - 0xFFFFFFFF

3.3 引导加载程序内核命令

3.3.1 连接

结构

接头	长度		CMD	CRC32			
0x80	0x01	0x00	0x12	C1	C2	C3	C4

说明

连接命令是第一个用于通过特定接口（UART 或 I2C）在主机和目标之间建立连接的命令。

受保护

否

命令返回

仅 BSL 确认

示例

主机：80 01 00 12 3A 61 44 DE

BSL：00

3.3.2 Get Device Info

结构

接头	长度		CMD	CRC32			
0x80	0x01	0x00	0x19	C1	C2	C3	C4

说明

该命令用于获取可用于数据事务的版本信息和缓冲区大小。

受保护

否

命令返回

BSL 确认和带有器件信息的 BSL 内核响应。请参阅器件信息了解更多详细信息

示例

主机：80 01 00 19 B2 B8 96 49

BSL:00 08 19 00 31 00 01 00 01 00 00 00 00 01 00 C0 06 60 01 00 20 01 00 00 00 01 00 00 00 49 61 57 8C

3.3.3 解锁引导加载程序

结构

接头	长度		CMD	数据	CRC32			
0x80	0x21	0x00	0x21	D1...D32	C1	C2	C3	C4

说明

该命令用于解锁引导加载程序。只有在引导加载程序解锁后，BSL 才会处理节 3.3 中列出的所有受保护命令。

如果主机发送了不正确的密码，BSL 会发送错误响应，然后器件进入睡眠模式 2 秒。在此 2 秒窗口期间，不会接收或处理任何命令。2 秒后，主机可以发送下一个密码。如果发送了 3 次错误的密码，则会采取安全警报措施。安全警报措施可在 BSL 配置中配置。有关更多详细信息，请参阅节 3.5.1.1。

受保护

否

数据

存储在 BSL 配置存储器中的 32 字节 BSL 密码。新器件的默认值全部为 0xFF。

命令返回

BSL 确认和带有有关操作状态的消息的 BSL 内核响应。有关详细信息，请参阅 BSL 内核消息部分。

示例

主机：80 21 00 21 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 02 AA F0 3D

BSL：00 08 02 00 3B 00 38 02 94 82

3.3.4 Program Data

结构

接头	长度		CMD	地址	数据	CRC32			
0x80	L1	L2	0x20	A1...A4	D1...Dn	C1	C2	C3	C4

说明

此命令用于将数据编程到特定存储器地址。数据可编程到主闪存、信息闪存（非主闪存）或数据存储体闪存。

重要要求：

- 起始地址必须是 16 字节对齐（地址 % 16 == 0）
- 数据长度必须是 16 字节对齐（长度 % 16 == 0）
- 为了对信息闪存（非主闪存）进行编程，必须在 BCR 配置中启用恢复出厂设置

受保护

是

地址

要编程的存储器区域的起始地址。A1...A4，其中 A1 是 32 位地址的最低有效字节。

数据

要写入指定地址的数据字节。可发送的数据的最大大小受器件缓冲区大小的限制。通过 Get Device Info 命令可以知道缓冲区大小。

命令返回

BSL 确认和带有有关操作状态的消息的 BSL 内核响应。有关更多详细信息，请参阅节 3.4.1 部分。

示例

主机：80 0D 00 20 00 00 00 00 00 00 00 04 00 00 00 08 7A DC AE B8

BSL：00 08 02 00 3B 00 38 02 94 82

3.3.5 快速编程数据

结构

接头	长度		CMD	地址	数据	CRC32			
0x80	L1	L2	0x24	A1...A4	D1...Dn	C1	C2	C3	C4

说明

Program Data Fast 命令与 Program Data 命令相同，只是该命令执行非阻塞写入，以加快编程过程。对于此命令，BSL 不会发送 BSL 内核消息响应来指示编程是否成功。

受保护

是

地址

要编程的存储器区域的起始地址。A1...A4，其中 A1 是 32 位地址的最低有效字节。

数据

要写入指定地址的数据字节。可发送的数据的最大大小受器件缓冲区大小的限制。通过 Get Device Info 命令可以知道缓冲区大小。

命令返回

BSL 确认。

示例

主机：80 0D 00 24 00 01 00 00 01 02 03 04 05 06 07 08 72 10 2A 18

BSL：00

3.3.6 回读数据

结构

接头	长度		CMD	地址	数据	CRC32			
0x80	0x09	0x00	0x29	A1...A4	L1...L4	C1	C2	C3	C4

说明

此命令用于从地址 A1...A4 开始读取数据。

应在 BSL 配置中启用读取，从而使用此命令读取数据。在 BSL 配置中，默认情况下会禁用它。

允许主闪存（应用程序存储器）、非主闪存（配置存储器）和 SRAM 存储器读取数据。

备注

主机无法完全访问 SRAM 存储器。有关更多详细信息，请参阅 SRAM 存储器使用情况。

受保护

是

地址

要回读的存储器区域的起始地址。A1...A4，其中 A1 是 32 位地址的最低有效字节。

数据

要读取的数据大小（以字节为单位），L1...L4，其中 L1 是最低有效字节。可读取的数据的最大大小受器件缓冲区大小的限制。通过 Get Device Info 命令可以知道缓冲区大小。

命令返回

如果回读命令有效，则使用 BSL 确认和带有所请求数据的 BSL 内核响应。有关更多详细信息，请参阅节 3.4.3。

如果回读命令具有无效的地址/长度，或者如果读数被禁用，相应的错误将在 BSL 确认后作为消息响应发送。

示例

主机：80 09 00 29 00 0C 00 00 08 00 00 00 32 9D B0 35

BSL：00 08 09 00 30 FF FF FF FF FF FF FF FF F6 2B A1 73

3.3.7 闪存范围擦除

结构

接头	长度		CMD	地址	数据	CRC32			
0x80	0x09	0x00	0x23	A1...A4（起始地址）	A1...A4（结束地址）	C1	C2	C3	C4

说明

Flash range erase 命令用于擦除指定的闪存存储器区域。闪存按扇区擦除 (2KB)，不能进行更小范围的擦除。

当起始地址和结束地址驻留在不同的闪存扇区中时，BSL 会擦除起始地址和结束地址之间的所有闪存扇区，包括包含这些地址的扇区。

此命令只能用于擦除主闪存存储器。不能擦除非主闪存存储器。

结束地址不应小于起始地址。

受保护

是

地址

要擦除的存储器区域的起始地址。A1...A4，其中 A1 是 32 位地址的最低有效字节。

数据

要擦除的存储器区域的结束地址。A1...A4，其中 A1 是 32 位地址的最低有效字节。

命令返回

BSL 确认和带有有关操作状态的消息的 BSL 内核响应。有关更多详细信息，请参阅节 3.4.1 部分。

示例

主机：80 09 00 23 00 01 00 00 FF 03 00 00 2B E6 BE D8

BSL：00 08 02 00 3B 00 38 02 94 82

3.3.8 批量擦除

结构

接头	长度		CMD	CRC32			
0x80	0x01	0x00	0x15	C1	C2	C3	C4

说明

批量擦除命令会擦除器件中可用的完整主闪存存储器（应用程序存储器）。

BCR 配置存储器中的批量擦除配置不会影响此 BSL 命令。

当一个闪存区域在 BCR 配置存储器中受到静态写保护时，此区域不能被擦除。

受保护

是

命令返回

BSL 确认和带有有关操作状态的消息的 BSL 内核响应。更多详细信息，请参阅节 3.4.1。

示例

主机： 80 01 00 15 99 F4 20 40

BSL：00 08 02 00 3B 00 38 02 94 82

3.3.9 恢复出厂设置

结构

接头	长度		CMD	数据	CRC32			
0x80	L1	L2	0x30	D1...D16	C1	C2	C3	C4

说明

factory reset 命令会擦除完整的主闪存（应用程序）存储器和非主闪存（配置）存储器。

处理此命令会受到 BCR 配置存储器中的出厂复位配置的影响。

恢复出厂设置

- 在无密码时允许（如果“Enabled”）
- 在有密码时允许（如果“Enabled with Password”）
- 不允许（如果“Disabled”）

当一个闪存区域在 BCR 配置存储器中受到静态写保护时，此区域不能被擦除。

受保护

是

数据

存储在 BCR 配置存储器中的 16 字节出厂复位密码。默认密码全为 0xFF。仅当 BCR 配置中的恢复出厂设置为“Enabled with Password”时，才需要密码。

命令返回

BSL 确认和带有有关操作状态的消息的 BSL 内核响应。更多详细信息，请参阅节 3.4.1。

小心

恢复出厂设置后，除非已恢复非主配置，否则系统极易受到潜在锁定情况的影响，在这种情况下，无法再次访问器件。

示例

场景 1：无密码即可恢复出厂设置

主机：80 01 00 30 DE 20 24 0B

BSL：00 08 02 00 3B 00 38 02 94 82

场景 2：通过密码恢复出厂设置

主机： 80 11 00 30 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 8A 28 EA DC

BSL： 00 08 02 00 3b 00 38 02 94 82

3.3.10 独立验证

结构

接头	长度		RSP	数据	CRC32			
0x08	0x05	0x00	0x32	D1...D4	C1	C2	C3	C4

说明

此命令用于验证存储在给定存储器范围内的数据的 CRC。这样可以更快地验证编程的数据。它要求数据大小至少为 1KB。

允许主闪存（应用程序存储器）和非主闪存（配置存储器）进行 CRC 验证

受保护

是

数据

要验证的数据大小（以字节为单位），D1...D4，其中 D1 是最低有效字节。1kB <= 大小 <= 64KB。

命令返回

BSL 确认和带有为所请求的存储器区域计算出的 CRC 值的 BSL 内核响应。有关更多详细信息，请参阅节 3.4.5。

如果验证命令的地址/长度无效，则相应的错误将作为 BSL 确认后的消息响应发送。请参阅节 3.4.1。

示例

主机：80 09 00 26 00 00 00 20 00 04 00 00 A0 97 D5 2E

BSL：00 08 02 00 3B 05 B7 F6 FE F2

3.3.11 启动应用程序

结构

接头	长度		CMD	CRC32			
0x80	0x01	0x00	0x40	C1	C2	C3	C4

说明

Start application 命令会发出系统复位，这会导致引导加载程序退出、重新运行引导代码，进而启动应用程序。

受保护

否

命令返回

BSL 确认

示例

Host:80 01 00 40 E2 51 21 5B

BSL:00

3.3.12 更改波特率

结构

接头	长度		CMD	数据	CRC32			
0x80	0x02	0x00	0x52	D1	C1	C2	C3	C4

说明

该命令可用于更改 UART 接口的波特率。新波特率将在为此数据包发送 BSL 确认后生效。

BSL UART 的默认波特率为 9600bps。

备注

更新波特率后，如果使用解锁引导加载程序命令发送了错误的 BSL 密码，则波特率设置将恢复为默认值。应以默认波特率进行进一步通信。

受保护

否

数据

表中指定的 D1 波特率。

ID	波特率 (bps)
1	4800
2	9600
3	19200
4	38400
5	57600
6	115200
7	1000000
8	2000000
9	3000000

命令返回

BSL 确认

示例

主机： 80 02 00 52 03 6C 83 A2 AF

BSL：00

3.4 引导加载程序内核响应

BSL 响应	RSP	数据
Memory read back	0x30	D1⋯Dn
Get Device Info	0x31	D1⋯D24
Standalone verification	0x32	D1⋯D4
消息	0x3B	MSG
详细错误	0x3A	D1..D3

缩写:

MSG

字节，包含 BSL 内核响应，并描述请求操作结果。其可以是一个错误代码，也可以是对成功操作的确认。当要求 BSL 响应数据时（例如，存储器、版本、CRC 或缓冲器大小），操作应答未成功发生， BSL 内核立即发送数据。

D1..Dn

数据字节，其中“n”受 BSL 最大缓冲区大小限制

3.4.1 BSL 内核消息

结构

接头	长度		RSP	数据	CRC32			
0x08	0x02	0x00	0x3B	MSG	C1	C2	C3	C4

说明

对于某些命令，BSL 会将消息响应发送到主机，以指示已处理命令的状态。该表列出了来自 BSL 的所有可能消息。

MSG	含义	可能原因(1)
0x00	操作成功	
0x01	BSL 锁定错误	尚未使用引导加载程序解锁密码命令解锁 BSL，或者在 BSL 解锁之后，命令接收阶段会发生超时
0x02	BSL 密码错误	发送了错误的密码来解锁引导加载程序。
0x03	多个 BSL 密码错误。已采取安全警报措施。	为解锁引导加载程序已发送错误的密码 3 次。
0x04	未知命令	提供给 BSL 的命令未被识别为有效命令
0x05	存储器范围无效	给定的存储器范围无效。
0x06	命令无效	指定给 BSL 的命令是已知命令，但在该时刻无效，无法处理。
0x07	已禁用恢复出厂设置	BCR 配置中禁用了恢复出厂设置
0x08	恢复出厂设置密码错误	当 BCR 配置的 factory reset 为“Enabled with password”时，使用 factory reset 命令发送的密码不正确或没有密码
0x09	读出错误	在 BCR 配置中禁用存储器读出
0x0A	地址或长度对齐无效	闪存编程的起始地址或数据长度不是 8 字节对齐的
0x0B	独立验证的长度无效	发送用于独立验证的数据大小小于 1KB

(1) 此处列出的可能原因不是出现状态或错误的唯一原因。它仅列出了导致出现错误（可由主机纠正）的可能软件原因。

3.4.2 详细错误

结构

接头	长度		RSP	数据	CRC32			
0x08	0x02	0x00	0x3A	D1..D3	C1	C2	C3	C4

说明

D1 - 错误类型

D3、D2 - 错误详细信息

可能的值

错误类型		错误详细信息	
值	说明	值	说明
0xF0	闪存错误	0xXX	包含 FLASHCTL.STATCMD 寄存器值

3.4.3 存储器回读

结构

接头	长度		RSP	数据	CRC32			
0x08	0x02	0x00	0x30	D1...Dn	C1	C2	C3	C4

说明

该命令会返回所请求的数据以响应回读命令

数据

数据 D1..Dn，其中“n”受 BSL 最大缓冲区大小限制。

3.4.4 器件信息

结构

接头	长度		RSP	数据	CRC32			
0x08	0x19	0x00	0x31	D1...D24	C1	C2	C3	C4

说明

该命令会返回版本信息和 BSL 缓冲区大小，以响应 Get Identity 命令

数据

识别字节	数据字节
命令解释器版本	[D02-D01]
构建 ID	[D04- D03]
应用程序版本	[D08-D05]
有源插件接口版本	[D10-D09]
BSL 最大缓冲区大小	[D12-D11]
BSL 缓冲区起始地址	[D16-D13]
BCR 配置 ID	[D20-D17]
BSL 配置 ID	[D24- D21]

应用程序版本：

32 位应用程序版本取自 BSL 配置中指定的地址

BSL 存储器大小：

可用于存储发送/接收到的 BSL 数据包的 RAM 数据缓冲区大小。

示例

主机：80 01 00 19 B2 B8 96 49

BSL：00 08 19 00 31 00 01 00 01 00 00 00 00 01 00 C0 06 60 01 00 20 01 00 00 00 01 00 00 00 49 61 57 8C

在上述给定响应中，

命令解释器版本 - 0x0100

构建 ID - 0x0100

应用程序版本 - 0x00000000

有源插件接口版本 - 0x0001

BSL 最大缓冲区大小 - 0x06C0

BSL 缓冲区起始地址 - 0x20000160

BCR 配置 ID - 0x00000001

BSL 配置 ID - 0x00000001

3.4.5 独立验证

结构

接头	长度		RSP	数据	CRC32			
0x08	0x05	0x00	0x32	D1...D4	C1	C2	C3	C4

说明

该命令会返回 CRC 值以响应 standalone verification 命令

数据

为所请求的存储器区域计算的 32 位 CRC 值。D1...D4，其中 D1 是 CRC32 的最低有效字节。

3.5 引导加载程序安全性

3.5.1 受密码保护的命令

可以直接或间接访问存储器中数据的所有命令都受密码保护。密码可在非主存储器的 BSL 配置中配置。

发送错误的密码后，器件将在接下来的 2 秒内睡眠，并且在此期间不接受任何命令，以使蛮力破解攻击更加困难。当错误密码发送 3 次时，BSL 会执行安全警报操作。

3.5.1.1 安全警报

在 BSL 配置中，安全警报可配置为以下三种模式中的任何一种。

1. **恢复出厂设置** - 恢复出厂设置将擦除整个主闪存和非主闪存存储器。无论 BCR 配置中的出厂复位模式如何，都将执行擦除。如果闪存的某些部分受到静态写保护，BSL 不能擦除整个闪存存储器。
2. **禁用引导加载程序** - 在 BCR 配置中禁用引导加载程序并退出 BSL。除非更新了 BCR 非主配置以启用引导加载程序，否则不能再进入 BSL。如果为非主器件启用了静态写保护，则不会禁用 BSL。
3. **不执行任何操作** - 不采取任何行动。

备注

对于恢复出厂设置安全警报配置，如果擦除了非主器件，引导加载程序密码将返回默认值。为了防止这种情况发生，可以对非主器件进行写保护。如果非主器件保持被擦除状态，器件会被锁定，无法再次访问器件。

3.5.2 BSL 条目

只允许通过引导代码进入引导加载程序。

可在 BCR 配置中禁用引导加载程序。

4 使用引导加载程序的示例程序流程

本节介绍了 BSL 主机通过引导加载程序加载映像的典型序列。此采样序列会擦除闪存存储器并对其中的新固件进行编程。

- 引导加载程序应通过以下方法之一启动：
  - 基于引脚的调用 (GPIO)
  - 应用程序请求
  - 调试邮箱命令
- 调用后，发送连接命令以通过所需的接口与 BSL 建立连接。
- 如果使用 UART 接口，则可以将波特率更改为更高的值，以加快进一步的通信，并且波特率是可选的。
- 要完全擦除闪存存储器，请使用 Mass erase 命令。仅当需要更新非主闪存时，才使用出厂复位命令。这是因为，如果非主闪存被擦除并保持未编程状态，器件就会被锁定。
- 对固件映像进行编程
- 对已编程的存储器区域进行 CRC 验证，以检查已编程数据的正确性。这是一个可选步骤。
- 应用程序可通过“Start application”命令启动。

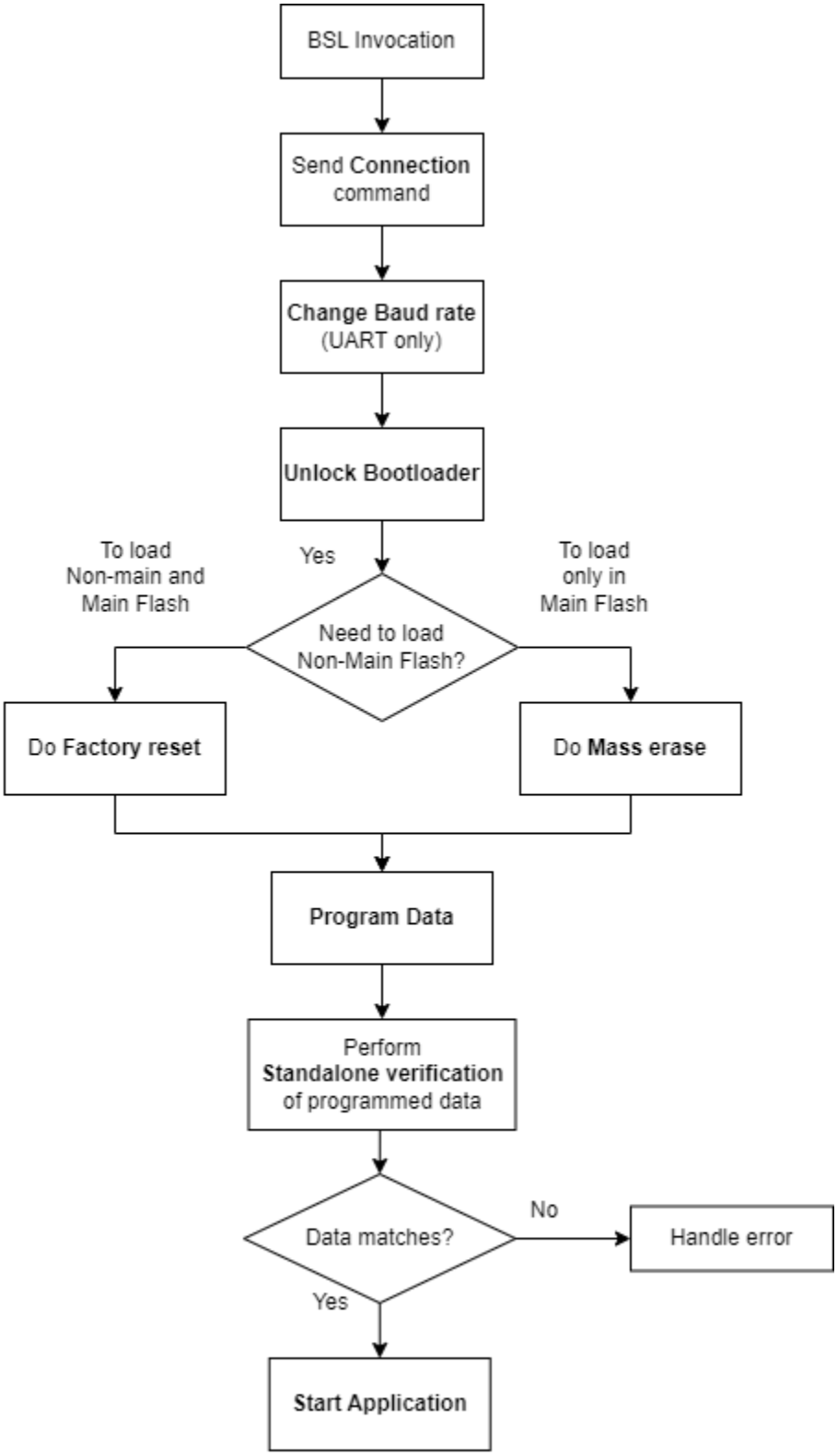


图 4-1. BSL 主机序列

5 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

日期	修订版本	注释
2025 年 12 月	*	初始发行版

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出任何明示或暗示的担保，包括但不限于对适销性、与某特定用途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安全、安保法规或其他要求。

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。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 TI 销售条款)、TI 通用质量指南 或 ti.com 上其他适用条款或 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。 除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

最后更新日期：2025 年 10 月