



## 摘要

MSP430™ 微控制器 (MCU) 上的引导加载程序 (BSL) 允许用户在原型设计、最终生产和使用期间，与 MSP430 MCU 中的嵌入式存储器进行通信。可编程存储器 (FRAM 存储器) 和数据存储器 (RAM) 均可按要求修改。

不要将引导加载程序与某些数字信号处理器 (DSP) 中能够从外部存储器自动加载程序代码 (和数据) 到 DSP 内部存储器的程序混为一谈。这些程序也被称为引导加载程序。

## 内容

<b>1 引言</b> .....	<b>2</b>
1.1 BSL 限制.....	2
1.2 其他实用文档.....	2
<b>2 BSL 特性概览</b> .....	<b>3</b>
<b>3 BSL 架构</b> .....	<b>4</b>
3.1 通信接口.....	4
3.2 BSL 存储器.....	4
3.3 调用 BSL.....	4
3.4 BSL 超时特性.....	7
3.5 BSL 版本号.....	7
3.6 BSL (用户) 配置.....	8
<b>4 BSL 协议</b> .....	<b>11</b>
4.1 BSL 数据包.....	11
4.2 BSL 安全.....	27
<b>5 常见的 BSL 用例</b> .....	<b>28</b>
5.1 概述和流程图.....	28
5.2 建立连接.....	29
5.3 擦除器件.....	29
5.4 下载应用程序.....	29
5.5 验证应用程序.....	29
5.6 运行应用程序.....	30
<b>6 定制 BSL</b> .....	<b>30</b>
<b>7 引导加载程序版本</b> .....	<b>31</b>
7.1 FR2xx BSL 版本.....	31
7.2 FR4xx BSL 版本.....	32
7.3 FR57xx BSL 版本.....	32
7.4 FR58xx 和 FR59xx BSL 版本.....	33
7.5 FR6xx BSL 版本.....	34
<b>8 修订历史记录</b> .....	<b>35</b>

## 商标

MSP430™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

## 1 引言

MSP430 BSL 允许用户在原型设计、最终生产和使用过程中，与 MSP430 微控制器 (MCU) 中的嵌入式存储器进行通信。可编程存储器 (FRAM 存储器) 和数据存储器 (RAM) 均可按要求予以修改。不要将引导加载程序与某些数字信号处理器 (DSP) 中能够从外部存储器自动加载程序代码 (和数据) 到 DSP 内部存储器的程序混为一谈。这些程序也被称为引导加载程序。

若要启动引导加载程序，必须将特定的 BSL 进入序列应用于专用引脚。如果应用程序代码将 PC 指针设置为 Z 区域中的 BSL 起始地址，也可调用 BSL。在 FR26xx、FR24xx 和 FR23xx MCU 上，空的复位矢量 (例如，在未编程的器件上) 也会调用 BSL。在 BSL 启动后，可向 BSL 发送一系列命令以执行所需功能 (例如，解锁器件、对存储器进行编程或重新编程，或者验证写入的数据)。可通过在定义的用户程序地址继续运行或按照复位条件来退出引导加载程序会话。

即使通过禁用 JTAG 来保护器件，仍然可以使用 BSL。为避免意外覆盖 BSL 代码，代码存储在安全的 ROM 存储器位置。为了避免不必要的存储器读出，任何可实现数据直接或者间接读取或写入的 BSL 命令都受到密码保护。因此，可使用 BSL 密码防止未经授权通过 BSL 访问器件存储器。BSL 密码就是器件上的中断矢量表中的内容。有关受密码保护的命令的更多信息，请参阅节 4.2。

### 1.1 BSL 限制

与基于 MSP 闪存的器件上的 BSL 相比，基于 MSP FRAM 的器件上的引导加载程序 (BSL) 会编程到只读存储器 (ROM) 中。请参阅《MSP430FRBoot - 适用于 MSP430 FRAM 大型存储器型号器件的主存储器引导加载程序和无线更新》，了解如何定制 FR5xx 和 FR6xx 器件中的引导加载程序。可使用 UART 和 I<sup>2</sup>C 通信协议。对于 FR4xx、FR21xx 和 FR20xx 器件而言，仅可使用 9600 波特率的 UART。

### 1.2 其他实用文档

#### 数据表

特定器件的数据表会介绍用于 BSL 通信的引脚和支持的协议。若要查找每个器件的数据表，请访问 [MSP430 超低功耗 MCU 产品页面](#)。

#### 系列用户指南

《MSP430FR58xx、MSP430FR59xx 和 MSP430FR6xx 系列用户指南》

《MSP430FR57xx 系列用户指南》

《MSP430FR4xx 和 MSP430FR2xx 系列用户指南》

#### 工具

[MSP BSL 工具文件夹](#)

#### 培训视频

[MSP BSL 概述](#)

[MSP BSL 选项](#)

[MSP 加密引导加载程序培训系列](#)

## 2 BSL 特性概览

表 2-1 汇总了按器件系列整理的 MSP430 MCU 的 BSL 特性。

表 2-1. BSL 概览

		MSP430								
		F20xx、G2xx0、G2xx1、G2xx2、I20xx	F1xx、F2xx、F4xx、G2xx3	F5xx、F6xx <sup>(1)</sup>		FR5xx、FR6xx		FR231x、FR242x、FR243x、FR25xx、FR263x	FR215x、FR235x、FR247x、FR267x	FR20xx、FR21xx、FR41xx
				非 USB	USB	工厂	加密引导加载程序 <sup>(4)</sup>			
常规	BSL 存储器类型	无 BSL	ROM	闪存 <sup>(2)</sup>	闪存 <sup>(2)</sup>	ROM	FRAM	ROM	ROM	ROM
	BSL 存储器大小	不适用	1KB	2KB	2KB	2KB	4KB	3KB	3KB	1KB
	由 TLV 配置的外设					✓	✓		✓	
	用户配置								✓	
	UART		✓	✓		✓	✓	✓	✓	✓
	I <sup>2</sup> C			✓		✓	✓	✓	✓	
	SPI									
协议	USB				✓					
	“1xx、2xx、4xx” 协议		✓							
	“5xx、6xx” 协议			✓	✓	✓	✓	✓	✓	✓
调用机制	I/O 上的进入序列	TEST/RST 上的序列	✓	✓		✓	✓	✓	✓	✓
		绑定到 V <sub>USB</sub> 的 PUR 引脚				✓				
		已定义 I/O 上的序列					✓			
	空的重置矢量调用 BSL				✓		✓	✓	✓	
	从软件应用程序调用 BSL		✓	✓	✓	✓	✓	✓	✓	✓
	无效或不完整的应用程序						✓			
工具支持	硬件	MSP-BSL “Rocket”	✓	✓		✓	✓	✓	✓	✓
		MSP-FET		✓		✓	✓	✓	✓	✓
		USB 电缆				✓				
		USB 转串行转换器 <sup>(3)</sup>	✓							
	软件 <sup>(4)</sup>	BSL 脚本编辑器			✓	✓	✓	✓	✓	✓
		BSLDEMO	✓							
	MSPBSL 库	✓	仅 UART	✓	仅 UART				✓	
安全性	密码保护		32 字节	32 字节 <sup>(5)</sup>	32 字节	32 字节		32 字节	32 字节	32 字节
	批量擦除错误密码 <sup>(6)</sup>	✓	✓	✓	✓	✓		✓	✓	✓
	使用签名或擦除 BSL 来完全禁用 BSL			✓	✓	✓	✓	✓	✓	✓
	BSL 有效载荷加密						✓			
	通过引导代码更新 IP 受保护的区域									
	认证加密						✓			
	附加安全性						✓ <sup>(7)</sup>			

(1) 有关这些器件上可用的 TI BSL 协议，请参阅专用器件数据表。TI 为每个闪存器件提供了专用 BSL 协议。

(2) 闪存中的 BSL 允许使用自定义版本替换 BSL。

(3) USB 转串行转换器与 BSLDEMO 兼容。在 DTR 引脚上为 RESET 引脚生成调用信号，并在 RTS 引脚上为 TEST 引脚生成调用信号。

(4) BSL 工具文件夹中提供了所有 BSL 软件附属资料（应用程序、示例、源代码和固件映像）。

[MSP430 USB 开发人员包](#)包括额外的 USB BSL 示例应用程序。

(5) F543x（非 A）有一个 16 字节的密码。

(6) 某些器件可以在密码错误时禁用批量擦除功能。请参阅器件系列用户指南。

(7) 通过 CRC 验证固件。

## 3 BSL 架构

### 3.1 通信接口

如节 2 所述，可用于 FRAM 器件 BSL 的通信协议是 UART 和 I<sup>2</sup>C。每个器件特定数据表中的 *引导加载程序 (BSL)* 部分提供了有关可用于 BSL 的通信协议和硬件引脚的信息。

#### 3.1.1 UART BSL

BSL 使用的 UART 协议定义为：

- 波特率配置为从 9600 波特开始。FR4xx、FR21xx 和 FR20xx 器件仅在 9600 波特下运行，而其他器件可在初始化后将波特率更改为更高的速度。
- UART 数据配置：起始位、8 个数据位 (LSB 在前)、一个偶校验位和 1 个停止位
- 在半双工模式下运行 (一次一个发送方)
- 由一个确认字符执行握手
- 从 MSP430 BSL 接收到字符之后到再发送新字符之前的最小延时时间为 1.2ms。

---

#### 备注

在初始化时应用 9600 波特以外的波特率会导致通信问题。

---

#### 3.1.2 I<sup>2</sup>C BSL

BSL 使用的 I2C 协议定义为：

- MSP430 BSL 是从站，主站必须向 BSL 从站请求数据。
- 采用 7 位寻址方式，从站地址为 0x48。
- 除了硬件 ACK 之外，还由一个确认字符执行握手。
- 从 MSP430 BSL 接收到字符之后到再发送新字符之前的最小延时时间为 1.2ms。
- BSL 不需要重复启动，但可以使用。

## 3.2 BSL 存储器

### 3.2.1 BSL 存储器布局

BSL 应用程序已在出厂时编程到 MSP430 器件的只读存储器区域中，并且无法进行定制。BSL 应用程序的大小因器件系列而异：

- FR4xx、FR21xx 和 FR20xx 的 BSL 大小为 1KB，处于地址 0x1000 到 0x13FF 之间。
- FR26xx、FR25xx、FR24xx 和 FR23xx 的 BSL 大小总共为 3KB，前 2KB 处于地址 0x1000 到 0x17FF 之间，剩余的 1KB 处于 0xFFC00 到 0xFFFFF 之间。
- FR5xx 和 FR6xx 的 BSL 大小为 2KB，处于地址 0x1000 到 0x17FF 之间。

### 3.2.2 BSL Z 区

在受保护的情况下，无法从 BSL 存储器外部的的位置读取或跳转到 BSL 存储器。这使得 BSL 更加安全并能防止 BSL 执行错误。但是，如果以这种方式保护整个 BSL 存储器空间，也意味着用户应用程序代码无法以任何方式调用 BSL (例如特意启动 BSL 或使用某些公共 BSL 函数)。

Z 区是存储器中一个特殊部分，旨在以受控方式公开访问受保护的 BSL。Z 区是位于地址 0x1000 和 0x100F 之间的可让外部应用程序代码跳转到并读取的 BSL 存储器部分。该区域用作网关，可从它跳转到 BSL 存储器中的任何位置。默认 TI BSL 使用此区域跳转到 BSL 的开头以及跳转到 BSL 公共函数中。

### 3.2.3 BSL 存储器注意事项

BSL 在 BSL 初始化开始时清除 RAM 部分内容，以防止读取任何可能位于其中的应用程序数据或代码。初始化完成后，BSL 会将 RAM 用于数据缓冲区和局部变量。从主应用程序调用 BSL 时，任何 RAM 内容都可能丢失。节 7 中的表格列出了被清除的 RAM 范围。

## 3.3 调用 BSL

可使用以下方法调用 MSP430 FRAM 器件上的 BSL 应用程序：

- BSL 可被应用软件调用 ( 请参阅节 3.3.1 )
- 在器件的 TEST 和  $\overline{\text{RST}}$  引脚上应用一个硬件进入序列, 器件启动代码就可调用 BSL ( 请参阅节 3.3.2 )
- 如果器件为空白, 则可在启动时调用 BSL ( 请参阅节 3.3.3 )。仅适用于 FR26xx、FR25xx、FR24xx 和 FR23xx 器件

### 3.3.1 软件 BSL 调用

BSL Z 区是存储器中的一小部分, 可从应用程序代码读取并调用此区域。它位于存储器地址 0x1000 到 0x100F 之间。

#### 3.3.1.1 从外部软件应用程序启动 BSL

存储器位置 0x1000 包含一条指向 BSL 开头的跳转指令, 可用于通过采用将程序计数器设置为 0x1000 的方式从正在运行的应用程序调用 BSL。堆栈始终被复位, 而 RAM 则被清空。BSL 不会禁用中断功能, 应在调用 BSL 之前由应用程序禁用中断功能。

TI 建议清除 BSL 应用程序中使用的任何模块寄存器的配置, 因为外部应用程序的配置会使 BSL 应用程序中断并导致意外行为。例如, 在 FR23xx 和 FR26xx MCU 中, Timer\_B 模块执行 BSL 的超时计算。如果 Timer\_B 也在外部应用程序中被使用并且在跳转到 BSL 应用程序之前未被清除, 则会导致意外行为。

位置 0x1000 可作为 C 函数被调用, 如以下示例代码所示:

```
_disable_interrupt(); // 禁用中断功能
((void (*)( ))0x1000)(); // 跳转到 BSL
```

#### 备注

FR5xx 和 FR6xx 器件上的 BSL 必须以 8MHz 的最大频率运行。如果器件在高于 8MHz 的频率下运行, 则在调用 BSL 之前必须将 MCLK 频率设置为 8MHz 或更低。

#### 3.3.1.2 BSL 操作

存储器位置 0x1002 包含一个前往“BSL 操作”函数的跳转。需要三个参数来调用此操作函数。第一个参数是一个描述函数的编号, 另外两个参数是已知值, 表明该函数被特意调用。

R12 : 函数编号

R13 : 0xDEAD

R14 : 0xBEEF

无论函数编号参数是什么, FR5xx 和 FR6xx 器件上的 BSL 始终执行 BSL 操作函数 2。

### 3.3.1.2.1 BSL 操作函数 2

函数编号： 2  
 函数名称： 返回到 BSL  
 描述： 任何提供的函数编号都会调用“返回到 BSL”函数。如果 BSL 已将一个程序写入 FRAM 或 RAM，通过“设置 PC”启动了该程序，然后该程序需要返回到 BSL，则可以使用此函数。此函数执行以下代码：

```

RETURN_TO_BSL
POP.W RET_low ; 从返回地址删除第一个字
POP.W RET_high ; 从返回地址删除第二个字
RETA ; 现在应该返回到 BSL 位置
    
```

### 3.3.2 硬件 BSL 调用

在  $\overline{\text{RST/NMI}}$  和 TEST 引脚上应用一个适当的进入序列，将强制 MSP430 从 BSL RESET 矢量（而不是从地址 FFFFh 的 RESET 矢量）开始执行程序。

如果应用程序与计算机 UART 连接，在通过电平转换器之后，这两个引脚可由串行通信接口 (RS232) 的 DTR 和 RTS 信号驱动。请参阅《MSP430™ 闪存器件引导加载程序 (BSL) 用户指南》中的“硬件描述”部分，了解硬件原理图。如果 TEST 保持在低电平，同时  $\overline{\text{RST/NMI}}$  从低电平上升为高电平，则会使用 0xFFFFE 地址的标准用户复位矢量（标准方法，请参阅图 3-1）。

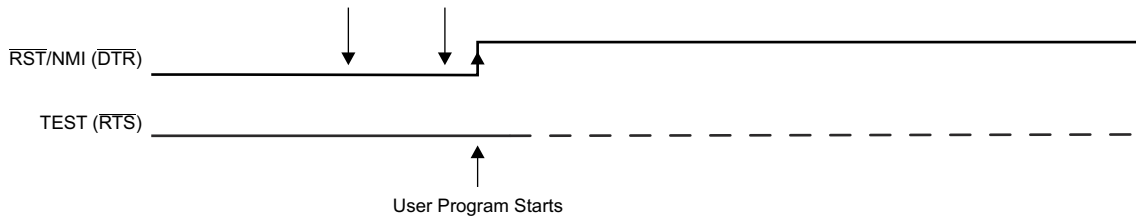
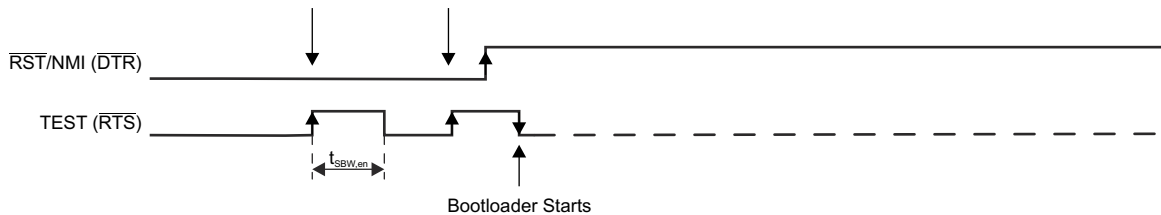


图 3-1. 标准 RESET 序列

当 TEST 引脚接收到至少两个上升沿（低电平到高电平的转换），并且 TEST 为高电平，同时  $\overline{\text{RST/NMI}}$  从低电平上升为高电平时，BSL 程序将开始执行（BSL 进入方法，请参阅图 3-2）。此电平转换触发机制提升了 BSL 启动的稳定性。TEST 引脚的第一个高电平必须至少为  $t_{\text{SBW,En}}$ （请参阅特定器件数据表，了解  $t_{\text{SBW,En}}$  参数）。



A. 引脚状态建议的最小时间为 250ns。

图 3-2. 共享 JTAG 引脚上的 BSL 进入序列

The TEST signal is typically used to switch the port pins between their application function and the JTAG function.在具有 BSL 功能的器件中，TEST 和  $\overline{\text{RST/NMI}}$  引脚也用于调用 BSL。若要调用 BSL， $\overline{\text{RST/NMI}}$  引脚必须配置为  $\overline{\text{RST}}$ ，并且必须在将 TEST 引脚拉高以及在 TEST 引脚上应用后续两个边沿（下降沿和上升沿）时保持低电平。在  $\overline{\text{RST/NMI}}$  引脚被释放且 TEST 引脚保持低电平的状态下，BSL 会启动（请参阅图 3-2）。

### 3.3.2.1 阻止硬件调用 BSL 的因素

在下列情况下，BSL 不由 BSL RESET 矢量启动：

- 当  $\overline{\text{RST}}/\text{NMI}$  为低电平时，TEST 引脚上的上升沿少于两个。
- 当  $\overline{\text{RST}}/\text{NMI}$  从低电平上升到高电平时，TEST 引脚在 TEST 引脚的第二个上升沿后不保持高电平。
- JTAG 可控制 MSP430 MCU 资源。
- 电源电压 ( $V_{CC}$ ) 下降至低于其阈值，并执行上电复位 (POR)。
- 针对 NMI 功能配置  $\overline{\text{RST}}/\text{NMI}$  引脚 (设置 NMI 位)。
- TCK 和 TMS 引脚上的悬空状态会提高器件进入 JTAG 模式的可能性。TI 建议对引脚进行外部端接以避免出现此问题。在 TCK 和 TMS 上添加一个 47k $\Omega$  上拉电阻器和一个 1nF 下拉电容器。根据系统中的噪声，可能需要更强的终端。

### 3.3.3 空白器件检测

FR26xx、FR25xx、FR24xx 和 FR23xx MCU 上的启动代码支持空白器件检测，从而避免 BSL 进入序列。这既节省了时间，也避免了为 BSL 调用序列设置两条额外的线路 (TEST、RST)。启用空白检测后，可绕过 BSL 进入序列。仅当复位矢量的值为 0xFFFF 时，器件才会直接跳转到 BSL 并绕过进入序列。

## 3.4 BSL 超时特性

FR26xx、FR25xx、FR24xx 和 FR23xx MCU 上的 BSL 实现了用于自动检测 BSL 接口的低功耗超时特性。如果没有在十秒内建立通信，器件将进入 LPM4 模式。若要再次调用 BSL，必须对器件进行下电上电操作，或者必须接收一次复位或 NMI。

## 3.5 BSL 版本号

主机编程器可通过使用“TX BSL 版本”命令来请求 BSL 版本号 (请参阅节 4.1.5.7，了解更多信息)。

### 字节 1：BSL 供应商信息

TI BSL 始终为 0x00。非 TI BSL 能够以另一种方式使用此区域。

### 字节 2：命令解释器版本

对 BSL 内核命令进行解释的那部分代码的版本号。

### 字节 3：API 版本

读取和写入 MSP430 内存的那部分代码的版本号。

保留位：

0x00 到 0x0F：表示此 BSL API 与闪存进行交互。

0x30 到 0x3F：表示此 BSL API 与 FRAM 进行交互。

0x80 到 0x8F：表示此 BSL 只能执行以下命令：

RX 数据块快速模式 (并且只能写入 RAM)、RX 命令、设置 PC

### 字节 4：外设接口版本

管理通信的那部分代码的版本号。

保留数：

0x00 到 0x2F：表示基于 Timer\_A 的 UART

0x30 到 0x4F：表示 USB

0x50 到 0x6F：表示基于 USCI 的 UART

0x70 到 0x8F：表示基于 eUSCI 的 UART

0x90 到 0x9F：表示基于 USCI 的 I<sup>2</sup>C

0xA0 到 0xAF：表示基于 eUSCI 的 I<sup>2</sup>C

0xB0 到 0xBF：表示混合 eUSCI I<sup>2</sup>C 和 UART

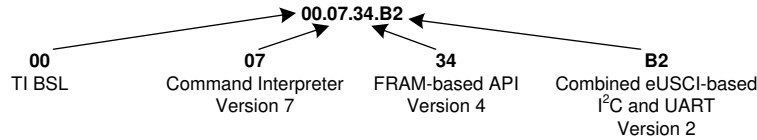


图 3-3. BSL 版本示例

### 3.6 BSL (用户) 配置

FR235x 和 FR215x BSL 从器件描述符 (TLV) 中读取其配置，并从用户配置的区域中读取其他设置信息。TLV 中的配置数据是经工厂编程的，无法更改。表 3-1 说明了 BSL 使用的外设模块和其他数据。节 3.6.1 介绍了用户可以修改的设置，包括接口选择和 RAM 清除。与其他 BSL 版本相比，基于 TLV 数据的 BSL 的初始化需要更多时间来完成。基于 TLV 的 BSL 的初始化从 BSL 进入调用序列开始，直到 BSL 准备好接收第一条命令，一共需要大约 300ms。

表 3-1. BSL 器件描述符

描述	地址偏移量	值
BSL 标签字段	00h	B5h
BSL 长度字段	01h	1Ch
计时器基地址 (用于超时的计时器模块)	02h、03h	标么值
eUSCI_A 基地址 (适用于 UART)	04h、05h	标么值
eUSCI_B 基地址 (适用于 I <sup>2</sup> C)	06h、07h	标么值
端口基地址 (适用于 UART 引脚)	08h、09h	标么值
端口 SELx 和引脚掩码 (适用于 UART) (PxSEL0、PxSEL1、RX 引脚掩码、TX 引脚掩码)	0Ah、0Bh、0Ch、0Dh	标么值
端口基地址 (适用于 I <sup>2</sup> C 引脚)	0Eh、0Fh	标么值
端口 SELx 和引脚掩码 (适用于 I <sup>2</sup> C) (PxSEL0、PxSEL1、SDA 引脚掩码、SCL 引脚掩码)	10h、11h、12h、13h	标么值
保留	14h、15h	-
保留	16h、17h	-
I <sup>2</sup> C 从站地址	18h	标么值
Tiny RAM 长度	19h	标么值
Tiny RAM 起始地址	1Ah、1Bh	标么值
BSL 用户配置数据的地址	1Ch、1Dh	标么值

#### 3.6.1 配置 BSL

BSL 用户配置是位于特定 FRAM 位置的一种数据结构，由 BSL 解析并用于配置 BSL 的特性。

用户配置是可选的。如果在存储器位置找不到用户配置，则 BSL 会使用默认值。

若要启用用户配置，它必须：

- 放置在正确的位置 (请参阅节 7.1，了解特定器件的地址。)
- 以正确的签名开头 (请参阅表 3-2)

表 3-2 列出了用户配置数据的结构。地址偏移量与 BSL 用户配置位置有关。

表 3-2. BSL 用户配置结构

描述	地址偏移量	值	
BSL 配置签名	0h	位 15-0	必须写成 695Ah，否则 BSL 用户配置将被忽略。



表 3-2. BSL 用户配置结构 (continued)

描述	地址偏移量	值	
BSL 配置	2h	位 15-8	必须写成 5Ah, 否则 BSL 用户配置将被忽略。
		位 7-4	保留
		位 3	在调用 BSL 时不要清除 Tiny RAM。如 BSL 密码错误, 则存储器总是会被清除。(并非每个器件都有 Tiny RAM。) 0b = 清除 Tiny RAM 1b = 不清除 Tiny RAM
		位 2	在调用 BSL 时不要清除 RAM。如 BSL 密码错误, 则存储器总是会被清除。请参阅节 7, 了解清除的 RAM 区域。 0b = 清除 RAM 1b = 不清除 RAM
		位 1-0	BSL 接口选择: 00b = 自动检测 UART 或 I2C BSL 通信 01b = 仅 UART 接口 10b = 仅 I2C 接口 11b = 保留
保留	4h-17h	位 15-0	保留
I <sup>2</sup> C 从站地址	18h	位 15-7	保留
		位 6-0	BSL I <sup>2</sup> C 接口的 7 位地址

### 3.6.1.1 BSL 用户配置示例

表 3-3 是 MSP430FR2355 BSL 的一个用户配置示例。在此示例中, 已禁用 RAM 清除, 且 I<sup>2</sup>C 从站地址设置为 0x60。用户配置存储器位于地址 FF88h 处。

表 3-3. BSL 用户配置示例

地址 (偏移量)	值	描述
0xFF88 (0h)	0x695A	BSL 配置签名
0xFF8A (2h)	0x5A04	BSL 配置: 禁用 RAM 清除
0xFF8C 到 0xFF9E	0xFFFF	保留
0xFFA0	0xFF60	BSL I <sup>2</sup> C 接口的 7 位地址

表 3-3 中的示例表示为 TI-TXT:

```
@FF88
5A 69 04 5A FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF 60 FF
q
```

### 3.6.1.2 在应用程序代码中实现 BSL 配置

FR235x 和 FR215x MCU 的链接器命令文件包含以下部分：

- .bslconfigsignature
- .bslconfig
- .bsli2caddress

在应用程序代码中，定义以下配置值：

```
#define BSL_CONFIG_SIGNATURE                (0x695A) // BSL Configuration Signature
#define BSL_CONFIG_PASSWORD                (0x5A00) // BSL User's Configuration Signature
#define BSL_CONFIG_TINY_RAM_ERASED_BY_INIT (0x0008) // Tiny RAM is erased during BSL invocation
#define BSL_CONFIG_TINY_RAM_NOT_ERASED_BY_INIT (0x0000) // (Default) Tiny RAM is not erased during BSL invocation
#define BSL_CONFIG_RAM_ERASED_BY_INIT     (0x0000) // RAM is erased during BSL invocation
#define BSL_CONFIG_RAM_NOT_ERASED_BY_INIT (0x0004) // (Default) RAM is not erased during BSL invocation
#define BSL_CONFIG_INTERFACE_UART_I2C    (0x0000) // Enable UART and I2C
#define BSL_CONFIG_INTERFACE_UART_ONLY   (0x0001) // Enable UART
#define BSL_CONFIG_INTERFACE_I2C_ONLY    (0x0002) // Enable I2C
#define BSL_CONFIG_INTERFACE_DEFAULT     (0x0003) // (Default / Reserved) Enable UART and I2C
#define BSL_I2C_ADDRESS_7BIT             (0x60) // Customized I2C Address
```

将预期的配置值应用到 BSL 配置部分：

```
#pragma RETAIN(bslConfigurationSignature)
#pragma DATA_SECTION(bslConfigurationSignature, ".bslconfigsignature")
const uint16_t bslConfigurationSignature = BSL_CONFIG_SIGNATURE;
#pragma RETAIN(bslConfig)
#pragma DATA_SECTION(bslConfig, ".bslconfig")
const uint16_t bslConfig = (BSL_CONFIG_PASSWORD | BSL_CONFIG_INTERFACE_UART_ONLY) ;
#pragma RETAIN(bslI2CAddr)
#pragma DATA_SECTION(bslI2CAddr, ".bsli2caddress")
const uint16_t bslI2CAddr = (0xFF00 | BSL_I2C_ADDRESS_7BIT) ;
```

## 4 BSL 协议

### 4.1 BSL 数据包

BSL 数据包有一个分层结构。BSL 内核命令包含由 BSL 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存在被称为外设接口代码 (PI 代码) 的包装器数据。这个包装器数据是所使用的外设和协议的特定信息，并且它包含可使 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 (请参阅表 4-1)。请参阅节 4.1.5，了解 BSL 内核命令的详细信息。

表 4-1. BSL 数据包

PI 代码	BSL 内核命令	PI 代码
-------	----------	-------

PI 代码取决于使用的协议是 UART 还是 I<sup>2</sup>C。以下几节将介绍 UART 和 I<sup>2</sup>C 外设接口包装器。

#### 4.1.1 UART 外设接口包装器

UART BSL 协议接口在多个数据包中采用。第一个数据包发送到 BSL 器件，并包含 BSL 内核命令及其包装器。表 4-2 汇总了数据包格式。

表 4-2. UART BSL 命令

标头	长度	长度	BSL 内核命令	CKL	CKH
0x80	NL	NH	请参阅节 4.1.5	CKL	CKH

第二个数据包接收自 BSL 器件，并包含 BSL 确认和 BSL 内核响应。表 4-3 列出了 BSL 响应数据包结构。

表 4-3. UART BSL 响应

确认	标头	长度	长度	BSL 内核响应 <sup>(1)</sup>	CKL	CKH
来自 BSL 的 ACK	0x80	NL	NH	请参阅节 4.1.4	CKL	CKH

(1) 并非总是包含 BSL 内核响应。

### 4.1.2 I<sup>2</sup>C 外设接口包装器

I<sup>2</sup>C BSL 协议接口在多个数据包中采用。第一个数据包作为写入请求发送到 BSL 从站地址，并包含 BSL 内核命令及其包装器。表 4-4 所示为此格式。

表 4-4. I<sup>2</sup>C BSL 命令

I2C	标头	长度	长度	BSL 内核命令	CKL	CKH
S/A/W	0x80	NL	NH	请参阅 节 4.1.5	CKL	CKH

第二个数据包作为读取请求发送到 BSL 从站地址，并包含 BSL 确认和 BSL 内核响应。表 4-5 所示为此 BSL 响应的格式。

表 4-5. I<sup>2</sup>C BSL 响应

I2C	ACK	标头	长度	长度	BSL 内核响应 <sup>(1)</sup>	CKL	CKH	I2C
S/A/R	来自 BSL 的 ACK	0x80	NL	NH	请参阅 节 4.1.4	CKL	CKH	停止

(1) 并非总是包含 BSL 内核响应。

### 4.1.3 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。如果数据传输中存在错误，则会立即发出一个错误消息。成功接收到所有数据后会发出一个 ACK，但这并不意味着命令已经成功执行（或者甚至于命令是有效的）而是意味着已将数据包正确格式化并已将其传递到 BSL 内核软件进行分析。

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 4-6 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。

表 4-6. UART 错误消息

数据	含义
0x00	ACK
0x51	帧头不正确。数据包没有以所需的 0x80 值开头。
0x52	校验位不正确。数据包没有正确的校验位值。
0x53 <sup>(1)</sup>	数据包大小为零。BSL 内核命令的大小被指定为 0。
0x54 <sup>(1)</sup>	数据包大小超出了缓冲器的范围。指定的数据包大小对于 RX 缓冲器来说太大。
0x55	未知错误
0x56 <sup>(1)</sup>	未知波特率。所提供的用于改变波特率的数据不是一个已知的值。
0x57	数据包大小错误。

(1) 不支持 FR4xx、FR21xx 和 FR20xx。

#### 4.1.4 BSL 内核响应和 BSL 内核消息

对于某些命令，BSL 以包含单字节消息的特定 BSL 内核响应进行回复。这个 1 字节消息被定义为 BSL 内核消息。有关哪个命令在其响应中具有 BSL 内核消息的详细信息，请参阅节 4.1.5 中的示例。

**表 4-7. BSL 内核响应结构 ( 具有 BSL 内核消息字节 )**

CMD 字节	BSL 内核消息
0x3B	消息 ( 请参阅表 4-8 )

**表 4-8. BSL 内核消息**

消息	含义
0x00	操作成功
0x01	存储器 ( 例如，闪存或 FRAM ) 写入校验失败。编程后，在编程数据上运行一个 CRC。如果 CRC 与预计的结果不匹配，则返回一个错误。
0x04	BSL 被锁定。解锁 BSL 所需的正确密码尚未提供。
0x05	BSL 密码错误。当尝试解锁时，向 BSL 提供了错误的密码。
0x07	未知指令。指定给 BSL 的命令未被识别。

### 4.1.5 BSL 内核命令

表 4-9 汇总了 BSL 内核命令。

表 4-9. BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应	部分
RX 数据块	是	0x10	(AL)	(AM)	(AH)	D1...Dn	是	节 4.1.5.1
RX 密码	否	0x11	-	-	-	D1...D32	视器件而定	节 4.1.5.2
批量擦除 <sup>(1)</sup>	否	0x15	-	-	-	-	视器件而定	节 4.1.5.3
CRC 校验 <sup>(1)</sup>	是	0x16	(AL)	(AM)	(AH)	长度 (低字节), 长度 (高字节)	是	节 4.1.5.4
加载 PC	是	0x17	(AL)	(AM)	(AH)	-	否	节 4.1.5.5
TX 数据块	是	0x18	(AL)	(AM)	(AH)	长度 (低字节), 长度 (高字节)	是	节 4.1.5.6
TX BSL 版本 <sup>(1)</sup>	是	0x19	-	-	-	-	是	节 4.1.5.7
RX 数据块快速模式 <sup>(1)</sup>	是	0x1B	(AL)	(AM)	(AH)	D1...Dn	否	节 4.1.5.8
更改波特率 <sup>(1) (2)</sup>	否	0x52	-	-	-	D1	否	节 4.1.5.9

(1) 不支持 FR4xx、FR21xx 和 FR20xx

(2) 仅适用于 UART 外设接口

#### AL , AM , AH

地址字节。低、中、上字节分别代表一个地址。

#### D1...Dn

数据字节 1 到 n (注意 : n 必须比 BSL 缓冲器的大小少 4 个字节)

#### 长度

包含 1 到 255 之间的值的一个字节描述了被发送或者在一个 CRC 中使用的字节数量。在多长度字节情况下, 它们如描述的那样被组合在一起以形成一个更大的值来描述所需的字节数量。

#### 4.1.5.1 RX 数据块

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
RX 数据块	是	0x10	(AL)	(AM)	(AH)	D1...Dn	是

##### 描述

BSL 内核从地址字段中指定的位置开始写入字节 D1 到字节 Dn。该命令与“RX 数据块快速模式”的不同之处在于它返回写入操作的状态。

##### 备注

当一个块的一部分被写入器件的存储器之外时（例如，开始写入 FRAM 但超出了存储器的末端），则不会写入整个数据块。

##### 保护

此命令受密码保护，如果未发送密码，则会失败。

##### 命令

0x10

##### 命令地址

应写入所接收数据的目标地址。

##### 命令数据

命令包含要写入的数据 D1 到数据 Dn。命令包含 n 个字节，其中 n 最大为 256。

##### 命令返回

BSL 确认和带有操作状态的 BSL 内核响应。请参阅节 4.1.4，了解更多有关 BSL 内核响应的信息。

##### UART PI 示例

将数据 0x76543210 写入地址 0x010000：

标头	长度	长度	CMD	AL	AM	AH	D1	D2	D3	D4	CKL	CKH
0x80	0x08	0x00	0x10	0x00	0x00	0x01	0x10	0x32	0x54	0x76	0x93	0xCA

成功写入数据的 BSL 响应：

ACK	标头	长度	长度	CMD	MSG	CKL	CKH
0x00	0x80	0x02	0x00	0x3B	0x00	0x60	0xC4

##### I<sup>2</sup>C PI 示例

将数据 0x76543210 写入地址 0x010000：

I2C	标头	长度	长度	CMD	AL	AM	AH	D1	D2	D3	D4	CKL	CKH
S/A/W	0x80	0x08	0x00	0x10	0x00	0x00	0x01	0x10	0x32	0x54	0x76	0x93	0xCA

成功写入数据的 BSL 响应：

I2C	ACK	标头	长度	长度	CMD	MSG	CKL	CKH	I2C
S/A/R	0x00	0x80	0x02	0x00	0x3B	0x00	0x60	0xC4	停止

#### 4.1.5.2 RX 密码

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
RX 密码	否	0x11	-	-	-	D1...D32	视器件而定

##### 描述

BSL 内核接收包含在数据包内的密码，如果密码与 BSL 中断矢量表（位于地址 0xFFE0 和 0xFFFF 之间）中的头 16 个字匹配的话，解锁 BSL 受保护的命令。

当提供的密码不正确时，将启动批量擦除。对于 MSP430FR5xx 和 MSP430FR6xx 器件而言，这意味着所有代码 FRAM 都会被擦除，但不会擦除信息存储器。对于 MSP430FR2xx 和 MSP430FR4xx 器件而言，这意味着包括信息存储器在内的所有代码 FRAM 都将被擦除。

执行批量擦除后，密码的所有字节始终为 0xFF。这通常用于访问空器件或在没有密码的情况下会将新应用程序加载到已锁定的器件。可以按照相应系列用户指南中所述的设置 BSL 签名来禁用批量擦除安全功能（请参阅节 1.2）。

##### 保护

此命令不受密码保护。

##### 命令

0x11

##### 命令地址

不适用

##### 命令数据

命令数据的长度为 32 字节，并包含器件密码。

##### 命令返回

MSP430FR5xx 和 MSP430FR6xx 引导加载程序不发送有关密码错误的 BSL 内核响应。应忽略来自器件的任何确认并再次初始化与 BSL 的通信。

MSP430FR2xx 和 MSP430FR4xx 引导加载程序会发送 BSL 确认和带有操作状态的 BSL 内核响应。请参阅节 4.1.4，了解更多有关 BSL 内核响应的信息。

##### UART PI 示例

解锁一个空白器件：

标头	长度	长度	CMD	D1	D2	D3	D4	D5	D6
0x80	0x21	0x00	0x11	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
D17	D18	D19	D20	D21	D22	D23	D24	D25	D26
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
D27	D28	D29	D30	D31	D32	CKL	CKH		
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x9E	0xE6		



密码成功的 BSL 响应：

ACK	标头	长度	长度	CMD	MSG	CKL	CKH
0x00	0x80	0x02	0x00	0x3B	0x00	0x60	0xC4

## I<sup>2</sup>C PI 示例

解锁一个空白器件：

I2C	标头	长度	长度	CMD	D1	D2	D3	D4	D5
S/A/W	0x80	0x21	0x00	0x11	0xFF	0xFF	0xFF	0xFF	0xFF
D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
D16	D17	D18	D19	D20	D21	D22	D23	D24	D25
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
D26	D27	D28	D29	D30	D31	D32	CKL	CKH	
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x9E	0xE6	

密码成功的 BSL 响应：

I2C	ACK	标头	长度	长度	CMD	MSG	CKL	CKH	I2C
S/A/R	0x00	0x80	0x02	0x00	0x3B	0x00	0x60	0xC4	停止

### 4.1.5.3 批量擦除

#### 结构 BSL 内核命令

对于 FR23xx、FR25xx 和 FR26xx：

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
批量擦除	否	0x15	-	-	-	-	是

对于 FR5xx 和 FR6xx：

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
批量擦除	否	0x15	-	-	-	-	否

#### 描述

器件中所有的代码 FRAM 都会被擦除。对于 MSP430FR5xx 和 MSP430FR6xx 器件而言，此功能不会擦除信息存储器。对于 MSP430FR23xx、MSP430FR25xx、MSP430FR24xx 和 MSP430FR26xx 器件而言，此功能会擦除信息存储器。

FR4xx、FR21xx 和 FR20xx MCU 上的 BSL 不支持“批量擦除”命令。触发批量擦除的另一种替代方法是发送包含错误密码的“RX 密码”命令。

#### 保护

此命令不受密码保护。

#### 命令

0x15

#### 命令地址

不适用

#### 命令数据

不适用

#### 命令返回

MSP430FR5xx 和 MSP430FR6xx 引导加载程序不发送有关批量擦除执行的 BSL 内核响应。BSL 确认是 0x00 或 0xFF。应忽略该确认并再次初始化与 BSL 的通信。

MSP430FR2xx 和 MSP430FR4xx 引导加载程序会发送 BSL 确认和带有操作状态的 BSL 内核响应。请参阅节 4.1.4，了解更多有关 BSL 内核响应的信息。

#### UART PI 示例

启动批量擦除：

标头	长度	长度	CMD	CKL	CKH
0x80	0x01	0x00	0x15	0x64	0xA3

BSL 响应 (操作成功)：

ACK	标头	长度	长度	CMD	MSG	CKL	CKH
0x00	0x80	0x02	0x00	0x3B	0x00	0x60	0xC4

## I<sup>2</sup>C PI 示例

启动批量擦除：

I2C	标头	长度	长度	CMD	CKL	CKH
S/A/W	0x80	0x01	0x00	0x15	0x64	0xA3

BSL 响应 ( 操作成功 )：

I2C	ACK	标头	长度	长度	CMD	MSG	CKL	CKH	I2C
S/A/R	0x00	0x80	0x02	0x00	0x3B	0x00	0x60	0xC4	停止

#### 4.1.5.4 CRC 校验

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
CRC 校验	是	0x16	(AL)	(AM)	(AH)	长度 (低字节), 长度 (高字节)	是

##### 描述

MSP430 根据 CCITT 标准执行一次 16 位 CRC 校验。指定的地址为 CRC 校验的第一个字节。两个字节用于表示长度。请参阅每个系列用户指南 (请参阅节 1.2) 中的 CRC 一章, 了解关于所采用的 CRC 硬件计算的更多详细信息。

##### 保护

此命令受密码保护, 如果未发送密码, 则会失败。

##### 命令

0x16

##### 命令地址

要开始 CRC 校验的地址。

##### 命令数据

CRC 校验的 16 位长度。D1 是长度的低字节, 而 D2 是长度的高字节。

##### 命令返回

BSL 确认和带有 CRC 值的 BSL 内核响应。请参阅节 4.1.4, 了解更多有关 BSL 内核响应的信息。

##### UART PI 示例

从地址 0x4400 到 0x4800 (大小为 1024) 执行 CRC 校验:

标头	长度	长度	CMD	AL	AM	AH	D1	D2	CKL	CKH
0x80	0x06	0x00	0x16	0x00	0x44	0x00	0x00	0x04	0x9C	0x7D

BSL 响应, 其中 0x55 是所计算校验和的低字节, 而 0xAA 是所计算校验和的高字节:

ACK	标头	长度	长度	CMD	D1	D2	CKL	CKH
0x00	0x80	0x03	0x00	0x3A	0x55	0xAA	0x12	0x2B

##### I<sup>2</sup>C PI 示例

从地址 0x4400 到 0x4800 (大小为 1024) 执行 CRC 校验:

I2C	标头	长度	长度	CMD	AL	AM	AH	D1	D2	CKL	CKH
S/A/W	0x80	0x06	0x00	0x16	0x00	0x44	0x00	0x00	0x04	0x9C	0x7D

BSL 响应, 其中 0x55 是所计算校验和的低字节, 而 0xAA 是所计算校验和的高字节:

I2C	ACK	标头	长度	长度	CMD	D1	D2	CKL	CKH	I2C
S/A/R	0x00	0x80	0x03	0x00	0x3A	0x55	0xAA	0x12	0x2B	停止

#### 4.1.5.5 加载 PC

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
加载 PC	是	0x17	(AL)	(AM)	(AH)	-	否

##### 描述

使 BSL 使用一个 CALLA 指令在指定地址开始执行。BSL 代码会立即与这个指令一同退出，所以无法预计内核响应。主应用程序可通过使用 BSL 操作函数 2 “返回到 BSL” 返回 BSL。请参阅节 3.3.1.2，了解更多信息。

请注意，密码保护此时并未激活。跳转到用户应用程序不会使器件重置，因此，来自 BSL 应用程序的寄存器配置会被保留。这可能会导致用户应用程序出现意外行为。一个示例是闪烁 LED 应用程序，它没有任何时钟模块配置（因此它使用默认的 1MHz 时钟）并且会以更快的速度闪烁，因为时钟模块由 BSL 应用程序设置为以 8MHz 运行。

##### 保护

此命令受密码保护，如果未发送密码，则会失败。

##### 命令

0x17

##### 命令地址

要设置程序计数器的地址

##### 命令数据

不适用

##### 命令返回

BSL 不返回确认。

##### UART PI 示例

将程序计数器设置为 0x4400：

标头	长度	长度	CMD	AL	AM	AH	CKL	CKH
0x80	0x04	0x00	0x17	0x00	0x44	0x00	0x42	0x0F

应用程序获得控制权后，BSL 不会响应。

##### I<sup>2</sup>C PI 示例

将程序计数器设置为 0x4400：

I2C	标头	长度	长度	CMD	AL	AM	AH	CKL	CKH	I2C
S/A/R	0x80	0x04	0x00	0x17	0x00	0x44	0x00	0x42	0x0F	停止

应用程序获得控制权后，BSL 不会响应。

#### 4.1.5.6 TX 数据块

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
TX 数据块	是	0x18	(AL)	(AM)	(AH)	长度 (低字节), 长度 (高字节)	是

##### 描述

BSL 从命令地址开始发送数据并包括大小命令数据。如果数据大小大于或等于缓冲区大小，此命令将启动多个数据包。

##### 保护

此命令受密码保护，如果未发送密码，则会失败。

##### 命令

0x18

##### 命令地址

开始发送数据的起始地址。

##### 命令数据

要发送的数据的 16 位长度。D1 是长度的低字节，而 D2 是长度的高字节。

##### 命令返回

BSL 确认和带有  $n$  个数据包的 BSL 内核响应，其中  $n$  为：

$$n = \text{ceiling}\left(\frac{\text{length}}{\text{buffer size} - 1}\right)$$

例如，如果请求 512 字节，且缓冲区大小为 260，则 BSL 会发送两个数据包，第一个数据包的长度为 259，第二个数据包的长度为 253。

请参阅节 4.1.4，了解更多有关 BSL 内核响应的信息。

##### UART PI 示例

从 RAM 地址 0x1C00 发送 4 个字节的数据：

标头	长度	长度	CMD	AL	AM	AH	D1	D2	CKL	CKH
0x80	0x06	0x00	0x18	0x00	0x1C	0x00	0x04	0x00	0x87	0x81

BSL 响应，其中 D1..D4 是请求的数据字节：

ACK	标头	长度	长度	CMD	D1	D2	D3	D4	CKL	CKH
0x00	0x80	0x05	0x00	0x3A	0x11	0x33	0x55	0x77	0x90	0x55

##### I<sup>2</sup>C PI 示例

从 RAM 地址 0x1C00 发送 4 个字节的数据：

I2C	标头	长度	长度	CMD	AL	AM	AH	D1	D2	CKL	CKH
S/A/W	0x80	0x06	0x00	0x18	0x00	0x1C	0x00	0x04	0x00	0x87	0x81

BSL 响应，其中 D1..D4 是请求的数据字节：

I2C	ACK	标头	长度	长度	CMD	D1	D2	D3	D4	CKL	CKH	I2C
S/A/R	0x00	0x80	0x05	0x00	0x3A	0x11	0x33	0x55	0x77	0x90	0x55	停止

#### 4.1.5.7 TX BSL 版本

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
TX BSL 版本	是	0x19	-	-	-	-	是

##### 描述

BSL 发送其版本信息 ( 请参阅节 3.5 , 了解更多详细信息 ) 。

##### 保护

此命令受密码保护 , 如果未发送密码 , 则会失败。

##### 命令

0x19

##### 命令地址

不适用

##### 命令数据

不适用

##### 命令返回

BSL 确认和带有版本号的 BSL 内核响应。数据按其在存储器中出现的顺序发送 , 包括以下数据字节 :

版本字节	数据字节
BSL 供应商	D1
命令解释器	D2
API	D3
外设接口	D4

请参阅节 4.1.4 , 了解更多有关 BSL 内核响应的信息。

##### UART PI 示例

请求 BSL 版本 :

标头	长度	长度	CMD	CKL	CKH
0x80	0x01	0x00	0x19	0xE8	0x62

BSL 响应 ( BSL 版本 00.07.34.B2 ) :

ACK	标头	长度	长度	CMD	D1	D2	D3	D4	CKL	CKH
0x00	0x80	0x05	0x00	0x3A	0x00	0x07	0x34	0xB2	0x14	0x90

**I<sup>2</sup>C PI 示例**

请求 BSL 版本 :

I2C	标头	长度	长度	CMD	CKL	CKH
S/A/W	0x80	0x01	0x00	0x19	0xE8	0x62

BSL 响应 ( BSL 版本 00.07.34.B2 ) :

I2C	ACK	标头	长度	长度	CMD	D1	D2	D3	D4	CKL	CKH	I2C
S/A/R	0x00	0x80	0x05	0x00	0x3A	0x00	0x07	0x34	0xB2	0x14	0x90	停止



#### 4.1.5.8 RX 数据块快速模式

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
RX 数据块快速模式	是	0x1B	(AL)	(AM)	(AH)	D1...Dn	否

##### 描述

此命令与节 4.1.5.1 相同，唯一的不同之处在于没有回复指示数据已正确编程。“RX 数据块快速模式”主要用于在 MSP430F5xx 和 MSP430F6xx 系列器件上进行的 USB 编程加速。

##### 备注

当一个块的一部分被写入器件的存储器之外时（例如，开始写入 FRAM 但超出了存储器的末尾），则不会写入整个数据块。

##### 保护

此命令受密码保护，如果未发送密码，则会失败。

##### 命令

0x1B

##### 命令地址

要写入所接收数据的目标地址。

##### 命令数据

命令包含要写入的数据 D1 到数据 Dn。命令包含 n 个字节，其中 n 最大为 256。

##### 命令返回

BSL 确认

##### UART PI 示例

将数据 0x76543210 写入地址 0x010000：

标头	长度	长度	CMD	AL	AM	AH	D1	D2	D3	D4	CKL	CKH
0x80	0x08	0x00	0x1B	0x00	0x00	0x01	0x10	0x32	0x54	0x76	0x3C	0x1C

BSL 响应：

ACK
0x00

##### I<sup>2</sup>C PI 示例

将数据 0x76543210 写入地址 0x010000：

I2C	标头	长度	长度	CMD	AL	AM	AH	D1	D2	D3	D4	CKL	CKH
S/A/W	0x80	0x08	0x00	0x10	0x00	0x00	0x01	0x10	0x32	0x54	0x76	0x93	0xCA

BSL 响应：

I2C	ACK	I2C
S/A/R	0x00	停止

#### 4.1.5.9 更改波特率

##### 结构 BSL 内核命令

BSL 命令	是否受保护	CMD	AL	AM	AH	数据	BSL 内核响应
更改波特率	否	0x52	-	-	-	D1	否

##### 描述

此命令会更改所有后续接收到的数据包波特率。在更改为新的波特率之前，通过发送单个 ACK 或错误字节以及旧的波特率来对该命令进行确认。随后的消息包无法预计。

##### 保护

此命令不受密码保护。

##### 命令

0x52

##### 命令地址

不适用

##### 命令数据

单字节 D1，指定要使用的新波特率。

D1	波特率
0x02	9600
0x03	19200
0x04	38400
0x05	57600
0x06	115200

##### 命令返回

BSL 确认

##### UART PI 示例

将波特率更改为 115200：

标头	长度	长度	CMD	D1	CKL	CKH
0x80	0x02	0x00	0x52	0x06	0x14	0x15

BSL 响应：

ACK
0x00

## 4.2 BSL 安全

### 4.2.1 受保护命令

为了保护器件内的数据，大多数内核命令受到保护。发送带有正确密码的“RX 密码”命令可解锁器件，只有顺利执行这一步之后，受保护的命令才能成功完成。特定于外设接口的命令不受保护。

### 4.2.2 RAM 擦除

在启动时，BSL 会擦除器件中的某些 RAM 位置并向其中写入常量字。通常这些位置是一个系列中最小的共享 RAM 地址。请参阅节 7，了解擦除的 RAM 地址。

### 4.2.3 进入 BSL

只能从 BSL 开头的 Z 区访问 BSL。这会将 BSL 操作限制在几个受控的入口点。无法对其余 BSL 代码进行“读取”和“执行”访问。

请参阅节 3.2.2，了解有关 Z 区的更多信息。

## 5 常见的 BSL 用例

本部分介绍了主机编程器与 MSP430 BSL 建立连接的流程。

### 5.1 概述和流程图

一个常见的 BSL 用例是擦除某个器件并在其中加载新的固件。通过使用错误密码发送一条“RX 密码”命令或通过“批量擦除”命令完成擦除。

将应用程序擦除后，密码的所有字节始终为 0xFF。使用正确的密码发送第二条“RX 密码”命令则可获得器件访问权。解锁 BSL 后，可通过使用“TX 数据块”命令来加载应用程序。如果需要额外的验证，主机编程器可请求应用程序的 CRC 或读取编程的应用程序代码。在验证应用程序后，可将器件复位，或者主机编程器可为 BSL 提供一个地址来设置程序计数器并开始执行。

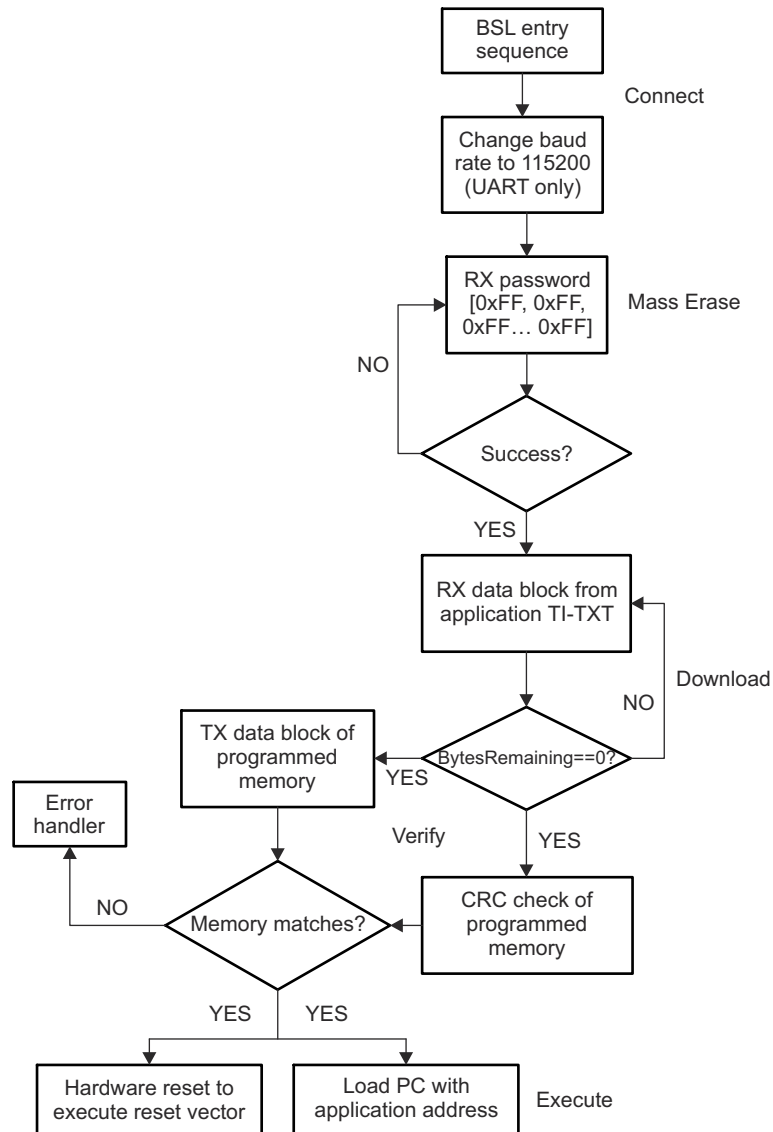


图 5-1. 流程图

## 5.2 建立连接

建立与器件 BSL 的连接是通过 BSL 加载应用程序的第一步。发起连接的方法是将节 3.3.2 中描述的 BSL 进入序列提供给 RST 和 TEST 引脚。

对于 UART 通信，可通过提高波特率来加快通信和下载速度。BSL 以 9600 波特开始运行，但允许主机请求更改为高达 115200 波特的各种预配置速率。对于大型应用程序和生产编程，TI 建议提高波特率。

有关“更改波特率”命令的更多信息，请参阅节 4.1.5.9。

## 5.3 擦除器件

应用程序编程过程的下一步是擦除器件。使用“批量擦除”命令，或使用“RX 密码”命令发送错误的密码，即可擦除器件。发送错误密码而不是使用批量擦除功能的优点在于，可将相同的编程流程用于空器件或已编程器件。主机发送一个空器件的密码（全为 0xFF）并检查结果。对于空器件，该密码将成功应用。对于已编程器件，该密码将失效，器件将自行完成批量擦除。主机可以重复此过程并发送相同的密码，现在对于已擦除器件，该密码将成功应用。这种做法对于从主机处理器进行生产编程特别有用，在这种情况下，工厂和现场的处理流程是相同的。

有关“RX 密码”命令的更多信息，请参阅节 4.1.5.2。有关“批量擦除”命令的更多信息，请参阅节 4.1.5.3。

## 5.4 下载应用程序

此步骤使用“RX 数据块”命令将应用程序代码下载到器件上。可由 MSP430 IDE 轻松生成的常见文件格式是 TI-TXT。下载内容分成较小的数据包，并使用重复的“RX 数据块”命令进行发送。发送完所有字节并且没有剩余数据后，主机编程器可继续进行下一步操作。

有关“RX 数据块”命令的更多信息，请参阅节 4.1.5.1。

## 5.5 验证应用程序

将应用程序下载到目标位置后，下一步是验证所下载的代码。对于此步骤，可以使用两种方法：请求 CRC 校验或读取应用程序代码。

第一种验证下载的方法是使用“CRC 校验”命令来请求器件对器件存储器的连续部分执行 CRC 校验。计算出的 CRC 值可与 IDE 或主机编程器生成的应用程序 CRC 作比较。

第二种验证下载的方法是使用“TX 数据块”命令来请求器件发送应用程序代码。数据从器件发送到主机，主机可以将其原始数据与从器件下载的应用程序数据作比较。

使用“CRC 校验”命令而不是“TX 数据块”命令进行应用程序验证的优势在于，器件对存储器执行 CRC 校验并随后发送两字节的 CRC 显著快于将应用程序中的每个字节发送回主机（相当于器件编程时间加倍）。

有关“CRC 校验”命令的更多信息，请参阅节 4.1.5.4。有关“TX 数据块”命令的更多信息，请参阅节 4.1.5.6。

## 5.6 运行应用程序

最后一步是运行应用程序。可以使用两种方法：将器件程序计数器设置为开始执行应用程序的地址，或执行一次器件复位。对包括复位矢量在内的整个应用程序进行编程时，通常最简单的方法是执行复位，以便启动代码调用由复位矢量指向的应用程序。这样做的优点是不必将显式地址读取或编码到主机编程器中，并且只要下载的任何应用程序包含复位矢量，就可以对其使用相同的复位过程。或者，BSL 可使用“TX 数据块”命令来读取复位矢量，然后通过“加载 PC”命令来调用读取的地址。

有关“加载 PC”命令的更多信息，请参阅节 4.1.5.5。有关“TX 数据块”命令的更多信息，请参阅节 4.1.5.6。

## 6 定制 BSL

MSP430 FRAM 器件中的 BSL 位于 ROM 中，不允许将定制 BSL 加载到标准 BSL 空间中。可绕过标准 BSL，并可使用作用与 BSL 类似的定制启动序列。这种类型的启动序列位于主存储器中，为开发人员提供了充分的灵活性和控制权。MSPBoot 是此类引导加载程序的一个示例。《[MSP430FRBoot - 适用于 MSP430™ FRAM 大型存储器型号器件的主存储器引导加载程序和无线更新](#)》介绍了位于主存储器中的 MSP430 FRAM 微控制器引导加载程序的实现方式，其中使用通用异步接收器/发送器 (UART) 通信或串行外设接口 (SPI) 总线以及 CC110x 射频收发器来完成无线下载 (OAD)。此引导加载程序不仅实现了高度灵活性和模块化，占用空间仍然很小，因此是一种非常具有成本效益的解决方案，并支持大型存储器型号（存储器空间大于 16KB 的器件）。

## 7 引导加载程序版本

### 7.1 FR2xx BSL 版本

**表 7-1.**

BSL 版本	00.08.35.B3
器件	MSP430FR2311、MSP430FR2310、MSP430FR2433、MSP430FR2633、MSP430FR2533、MSP430FR2632、MSP430FR2532、MSP430FR2522、MSP430FR2422
擦除 RAM	0x2000 到 0x23FF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	eUSCI_A
外设 I <sup>2</sup> C	eUSCI_B
用于超时的计时器模块	FR23xx : TIMER_B , FR26xx : TIMER_A
已知错误	无

**表 7-2.**

BSL 版本	00.87.45.74
器件	MSP430FR2111、MSP430FR2110、MSP430FR2100、MSP430FR2000、MSP430FR2033、MSP430FR2032
擦除 RAM	0x2000 到 0x21FF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	eUSCI_A
外设 I <sup>2</sup> C	不适用
用于超时的计时器模块	不适用
已知错误	无

**表 7-3.**

BSL 版本	00.09.36.B4
器件	MSP430FR2355、MSP430FR2353、MSP430FR2155、MSP430FR2153
擦除 RAM	0x2000 到 0x21FF
用于内核的缓冲器大小	260
值得注意的信息	<ol style="list-style-type: none"> <li>1. 用户覆盖配置可以置于 0xFF88 下。器件没有 Tiny RAM，清除功能无效。</li> <li>2. 与其他 BSL 版本相比，基于 TLV 数据的 BSL 的初始化需要更多时间来完成。基于 TLV 的 BSL 的初始化从 BSL 进入调用序列开始，直到 BSL 准备好接收第一条命令，一共需要大约 300ms。</li> </ol>
时钟配置	以 8MHz 运行
外设 UART	eUSCI_A
外设 I <sup>2</sup> C	eUSCI_B
用于超时的计时器模块	TIMER_B
已知错误	无

表 7-4.

BSL 版本	00.09.36.B5
器件	MSP430FR2676、MSP430FR2476、MSP430FR2673、MSP430FR2672、MSP430FR2675、MSP430FR2475
擦除 RAM	0x2000 到 0x21FF
用于内核的缓冲器大小	260
值得注意的信息	1. 与其他 BSL 版本相比，基于 TLV 数据的 BSL 的初始化需要更多时间来完成。基于 TLV 的 BSL 的初始化从 BSL 进入调用序列开始，直到 BSL 准备好接收第一条命令，一共需要大约 300ms。
时钟配置	以 8MHz 运行
外设 UART	eUSCI_A
外设 I2C	eUSCI_B
用于超时的计时器模块	TIMER_B
已知错误	无

## 7.2 FR4xx BSL 版本

表 7-5.

BSL 版本	00.87.45.74
器件	MSP430FR4133、MSP430FR4132、MSP430FR4131
擦除 RAM	0x2000 到 0x21FF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	eUSCI_A
外设 I <sup>2</sup> C	不适用
用于超时的计时器模块	不适用
已知错误	无

## 7.3 FR57xx BSL 版本

表 7-6.

BSL 版本	00.04.31.71
器件	MSP430FR5739、MSP430FR5738、MSP430FR5737、MSP430FR5736、MSP430FR5735、MSP430FR5734、MSP430FR5733、MSP430FR5732、MSP430FR5731、MSP430FR5730、MSP430FR5729、MSP430FR5728、MSP430FR5727、MSP430FR5726、MSP430FR5725、MSP430FR5724、MSP430FR5723、MSP430FR5722、MSP430FR5721、MSP430FR5720
擦除 RAM	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260
值得注意的信息	1.在器件数据表中注明了 TX 和 RX 引脚 2.批量擦除命令或错误的密码会触发 BSL 重置。这会将 BSL 状态重置为默认设置 ( 9600 波特，密码已锁定 )
时钟配置	以 8MHz 运行
外设 UART	eUSCI_A
外设 I <sup>2</sup> C	不适用
用于超时的计时器模块	不适用
已知错误	1.不能保证波特率在所有时钟、电压和温度变化范围内均为 115k。



## 7.4 FR58xx 和 FR59xx BSL 版本

**表 7-7.**

BSL 版本	00.07.34.B2
器件	MSP430FR5969、MSP430FR59691、MSP430FR5968、MSP430FR5967、MSP430FR5949、MSP430FR5948、MSP430FR5947、MSP430FR59471、MSP430FR5959、MSP430FR5958、MSP430FR5957、MSP430FR5869、MSP430FR5868、MSP430FR5867、MSP430FR58671、MSP430FR5849、MSP430FR5848、MSP430FR5847、MSP430FR58471、MSP430FR5859、MSP430FR5858、MSP430FR5857、MSP430FR5972、MSP430FR59721、MSP430FR5872、MSP430FR58721、MSP430FR5989、MSP430FR59891、MSP430FR5988、MSP430FR59881、MSP430FR5987、MSP430FR59871、MSP430FR5986、MSP430FR5889、MSP430FR58891、MSP430FR5888、MSP430FR58881、MSP430FR5887、MSP430FR58871、MSP430FR5970、MSP430FR5870、MSP430FR5922、MSP430FR59221
擦除 RAM	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	( FR5xxx 或 FR6xxx ) eUSCI_A0
外设 I <sup>2</sup> C	( FR5xxx1 或 FR6xxx1 ) eUSCI_B0
用于超时的计时器模块	不适用
已知错误	无

**表 7-8.**

BSL 版本	00.08.35.B3
器件	MSP430FR5994、MSP430FR59941、MSP430FR5992、MSP430FR5964、MSP430FR59641、MSP430FR5962
擦除 RAM	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	( FR5xxx 或 FR6xxx ) eUSCI_A0
外设 I <sup>2</sup> C	( FR5xxx1 或 FR6xxx1 ) eUSCI_B0
用于超时的计时器模块	不适用
已知错误	无

## 7.5 FR6xx BSL 版本

表 7-9.

BSL 版本	00.07.34.B2
器件	MSP430FR6972、MSP430FR69721、MSP430FR6872、MSP430FR68721、MSP430FR6922、MSP430FR69221、MSP430FR6822、MSP430FR68221、MSP430FR6970、MSP430FR6870、MSP430FR6920、MSP430FR6820、MSP430FR6989、MSP430FR69891、MSP430FR6988、MSP430FR69881、MSP430FR6987、MSP430FR69871、MSP430FR6889、MSP430FR68891、MSP430FR6888、MSP430FR68881、MSP430FR6887、MSP430FR68871、MSP430FR6979、MSP430FR6879、MSP430FR6977、MSP430FR6877、MSP430FR6928、MSP430FR6927
擦除 RAM	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	( FR5xxx 或 FR6xxx ) eUSCI_A0
外设 I <sup>2</sup> C	( FR5xxx1 或 FR6xxx1 ) eUSCI_B0
用于超时的计时器模块	不适用
已知错误	无

表 7-10.

BSL 版本	00.08.35.B3
器件	MSP430FR6047、MSP430FR60471、MSP430FR6045、MSP430FR6037、MSP430FR60371、MSP430FR6035、MSP430FR6007、MSP430FR6005
擦除 RAM	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	( FR5xxx 或 FR6xxx ) eUSCI_A0
外设 I <sup>2</sup> C	( FR5xxx1 或 FR6xxx1 ) eUSCI_B0
用于超时的计时器模块	不适用
已知错误	无

表 7-11.

BSL 版本	00.08.35.B3
器件	MSP430FR6043、MSP430FR60431、MSP430FR6041、MSP430FR5043、MSP430FR50431、MSP430FR5041
擦除 RAM	0x3000 到 0x5FFF
用于内核命令的缓冲器大小	260
值得注意的信息	无
时钟配置	以 8MHz 运行
外设 UART	( FR5xxx 或 FR6xxx ) eUSCI_A3
外设 I <sup>2</sup> C	( FR5xxx1 或 FR6xxx1 ) eUSCI_B0
用于超时的计时器模块	不适用
已知错误	无

## 8 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

<b>Changes from FEBRUARY 20, 2021 to SEPTEMBER 22, 2022</b>	<b>Page</b>
• 从表 2-1 BSL 概览中删除了 MSP432P4xx 列 ( 这些器件已停产 ) .....	3
• 删除了表 7-4 中有关 Tiny RAM 的错误注释.....	31

## 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