

## 摘要

MSP430™ 引导加载程序 (BSL, 之前称为引导装载程序) 方便用户在原型设计阶段、最终生产阶段和服务期间与 MSP430 微控制器 (MCU) 中的嵌入式存储器进行通信。可编程存储器 (闪存) 和数据存储器 (RAM) 都可以根据需要进行修改。务将引导加载程序与某些数字信号处理器 (DSP) 中将外部存储器中的程序代码 (和数据) 自动加载到 DSP 内部存储器的引导装载程序混为一谈。

要使用引导加载程序, 必须应用特定的 BSL 进入序列。添加的命令序列启动所需的函数。通过在定义的用户程序地址继续运行或通过重置条件可以退出引导加载会话。

如果通过禁用 JTAG 来保护器件, 则仍然可以使用 BSL。通过 BSL 对 MSP430 MCU 的访问受到 BSL 密码的保护, 以避免误用。BSL 密码等于器件上中断矢量表的内容。

## 内容

<b>1 引言</b> .....	<b>3</b>
1.1 补充在线信息.....	3
1.2 BSL 特性概览.....	4
1.3 调用 BSL.....	5
1.4 UART 协议.....	7
1.5 USB 协议.....	7
<b>2 引导加载程序协议 - 1xx、2xx 和 4xx 系列</b> .....	<b>8</b>
2.1 同步序列.....	8
2.2 命令.....	8
2.3 编程流程.....	8
2.4 数据帧.....	9
2.5 可史可法 BSL.....	14
2.6 退出 BSL.....	14
2.7 密码保护.....	14
2.8 代码保护保险丝.....	15
2.9 BSL 内部设置和资源.....	15
<b>3 引导加载程序协议 - F5xx 和 F6xx 系列</b> .....	<b>18</b>
3.1 BSL 数据包.....	18
3.2 UART 外设接口 (PI).....	18
3.3 I <sup>2</sup> C 外设接口.....	19
3.4 USB 外设接口.....	21
3.5 BSL 内核命令结构.....	21
3.6 BSL 安全.....	23
3.7 BSL 内核响应.....	23
3.8 BSL 公共函数和 Z 区域.....	26
<b>4 引导加载程序硬件</b> .....	<b>28</b>
4.1 硬件描述.....	28
<b>5 器件和引导加载程序版本间的差异</b> .....	<b>32</b>
5.1 1xx、2xx 和 4xx BSL 版本.....	32
5.2 ROM BSL 版本 1.10 的特殊考虑.....	39
5.3 1xx、2xx 和 4xx BSL 已知问题.....	40
5.4 有关 MSP430F14x 器件系列 BSL 的特殊说明.....	40
5.5 基于 F5xx 和 F6xx 闪存的 BSL 版本.....	41

6 引导加载程序 PCB 布局建议.....	48
7 修订历史记录.....	51

## 插图清单

图 1-1. 标准 RESET 序列.....	5
图 1-2. 共享 JTAG 引脚上的 BSL 进入序列.....	5
图 1-3. 专用 JTAG 引脚上的 BSL 进入序列.....	6
图 3-1. 基本协议 - 字节级确认.....	19
图 3-2. 字节级确认.....	20
图 4-1. 引导加载程序接口原理图.....	28
图 6-1. 通用 BSL 接口 PCB 布局, 顶视图.....	48
图 6-2. 通用 BSL 接口 PCB 布局, 底视图.....	48
图 6-3. 通用 BSL 接口组件位置.....	49
图 6-4. 通用 BSL 接口组件位置.....	50

## 商标

MSP430™ and TI E2E™ are trademarks of Texas Instruments.

所有商标均为其各自所有者的财产。

## 1 引言

引导加载程序提供了一种在 MSP430 项目开发和更新阶段对闪存进行编程的方法。这个方法可以通过使用 UART 协议发送命令的工具来激活。BSL 使用户能够控制 MSP430 MCU 的活动，并使用个人计算机或其他设备交换数据。

为避免意外覆盖 BSL 代码，此代码存储在安全的内存位置：要么在 ROM 中，要么在特别保护的闪存中。为了避免读取不必要的源代码，任何直接或者间接允许数据读取的 BSL 命令都受到密码保护。

要调用引导加载程序，必须将 BSL 条目序列应用于专用引脚。之后是同步字符，后跟专用命令的数据帧，以启动所需功能。

### 1.1 补充在线信息

作为本文档的补充，请访问[用于 MSP 低功耗微控制器的引导加载程序 \(BSL\)](#)。此页面包含附加的 BSL 用户指南、源代码、固件映像以及带有文档和代码示例的 BSL 脚本程序的链接。

[TI E2E™ 支持论坛](#)提供了额外的支持。

## 1.2 BSL 特性概览

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

表 1-1. 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									
	USB				✓					
协议	“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。

## 1.3 调用 BSL

### 1.3.1 硬件 BSL 调用

#### 1.3.1.1 具有共享 JTAG 引脚的 MSP430 器件

在  $\overline{\text{RST}}/\text{NMI}$  和 TEST 引脚上应用适当的进入序列，迫使 MSP430 MCU 在 BSL RESET 矢量而不是在地址 FFFEH 的 RESET 矢量上开始执行程序。

如果应用程序与计算机 UART 接口，这两个引脚可以在通过电平转换器后由串行通信接口 (RS232) 的  $\overline{\text{DTR}}$  和  $\overline{\text{RTS}}$  信号驱动。可在节 4 中找到硬件及相关注意事项的详细说明。如果 TEST 保持在低电平，同时  $\overline{\text{RST}}/\text{NMI}$  从低电平上升为高电平，则使用 FFFEH 地址处的普通用户重置矢量 (标准方法，请参阅图 1-1)。

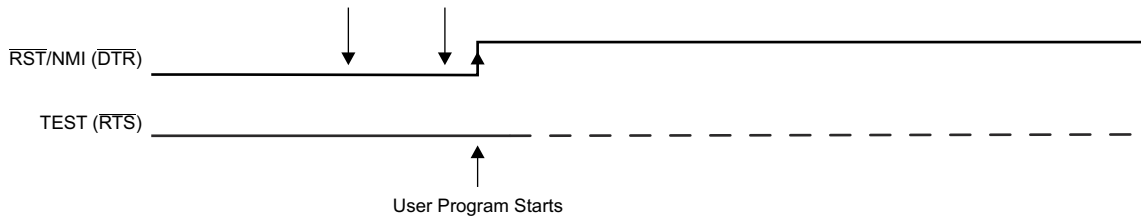


图 1-1. 标准 RESET 序列

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  $\overline{\text{RST}}/\text{NMI}$  从低电平上升为高电平时，BSL 程序开始执行 (BSL 进入方法，请参阅图 1-2)。这种电平转换触发机制提高了 BSL 启动的可靠性。TEST 引脚的第一个高电平必须至少为  $t_{\text{SBW,En}}$  (有关  $t_{\text{SBW,En}}$  参数，请参阅专用器件数据表)。

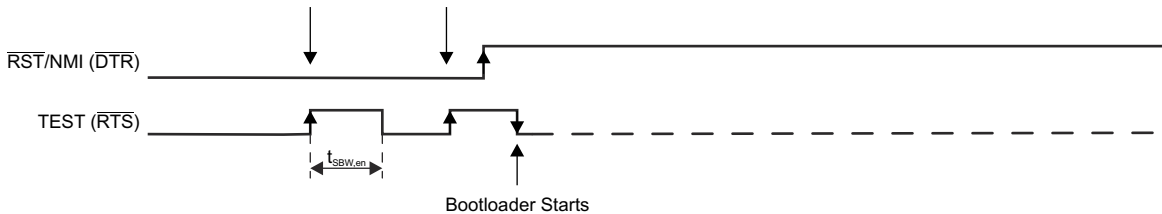


图 1-2. 共享 JTAG 引脚上的 BSL 进入序列

#### 备注

对于具有 DVIO 电源的 MSP430F522x 和 MSP430F521x 分离轨器件，进入序列应用于  $\overline{\text{RST}}/\text{NMI}$  和 BSLEN 引脚。有关引脚信息，请参阅专用器件数据表。有关更多信息，请参阅[使用 MSP430F522x 和 MSP430F521x 器件进行设计](#)中的引导加载程序部分。

#### 备注

引脚状态建议的最小时间为 250ns。有关任何差异，请参阅专用器件勘误表，因为某些 5xx 和 6xx 器件修订版需要专用进入序列。

TEST 信号通常用于切换应用函数和 JTAG 函数之间的端口引脚。在具有 BSL 功能的器件中，TEST 和  $\overline{\text{RST}}/\text{NMI}$  引脚也用于调用 BSL。若要调用 BSL， $\overline{\text{RST}}/\text{NMI}$  引脚必须配置为  $\overline{\text{RST}}$ ，并且必须保持为低电平，同时拉高 TEST 引脚，并在 TEST 引脚上应用后续两个边沿 (下降沿和上升沿)。RST/NMI 引脚被释放后，TEST 引脚保持低电，接着 BSL 会启动 (请参阅图 1-2)。

### 1.3.1.1.1 阻止使用共享 JTAG 引脚调用 BSL 的因素

在下列情况下，BSL 不由 BSL RESET 矢量启动：

- 当  $\overline{\text{RST/NMI}}$  为低电平时，TEST 引脚上的上升沿少于两个。
- 当  $\overline{\text{RST/NMI}}$  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再保持高电平。
- JTAG 可控制 MSP430 MCU 资源。
- 电源电压  $V_{CC}$  降至其阈值以下，并执行上电复位 (POR)。
- $\overline{\text{RST/NMI}}$  引脚配置为 NMI 功能 (设置 NMI 位)。
- 如果 TCK 和 TMS 引脚悬空，器件可能会无意中进入 JTAG 模式。为避免此问题，请应用推荐的外部端接。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。

### 1.3.1.2 具有专用 JTAG 引脚的 MSP430 闪存器件

具有专用 JTAG 引脚的器件使用 TCK 引脚而非 TEST 引脚。

每当 CK 引脚接收到至少两个下降沿，并且 TCK 为低电平，同时  $\overline{\text{RST/NMI}}$  从低电平上升为高电平，BSL 程序就会开始执行 (BSL 进入方法，请参阅图 1-3)。这种电平转换触发机制提高了 BSL 启动的可靠性。

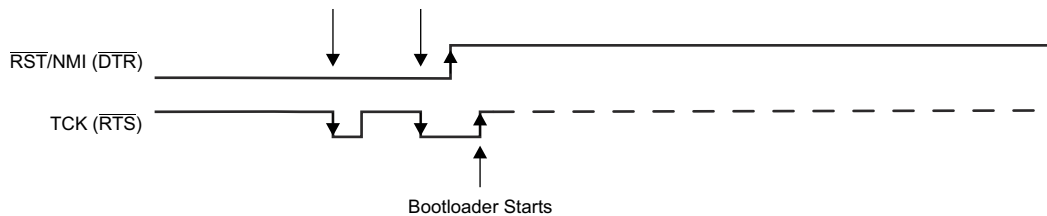


图 1-3. 专用 JTAG 引脚上的 BSL 进入序列

#### 备注

引脚状态建议的最小时间为 250ns。有关任何差异，请参阅专用器件勘误表，因为某些 5xx 和 6xx 器件修订版要求专用进入序列。

### 1.3.1.2.1 阻止使用专用 JTAG 引脚调用 BSL 的因素

在下列情况下，BSL 不由 BSL RESET 矢量启动：

- 当  $\overline{\text{RST/NMI}}$  为低电平，TCK 引脚上的下降沿少于两个。
- 如果  $\overline{\text{RST/NMI}}$  从低电平上升为高电平，TCK 为高电平。
- JTAG 可控制 MSP430 MCU 资源。
- 电源电压  $V_{CC}$  降至其阈值以下，并执行上电复位 (POR)。
- $\overline{\text{RST/NMI}}$  引脚配置为 NMI 功能 (设置 NMI 位)。

### 1.3.1.3 带有 USB 的器件

当器件由 VBUS 供电时，满足以下两个条件之一时，调用带有 USB 的器件：

- 器件由 USB 供电且复位矢量为空。
- 器件由连接到  $V_{BUS}$  的 PUR 引脚供电。

### 1.3.2 软件 BSL 调用

要从正在运行的应用中调用 BSL，请将程序计数器设置为 BSL 所在的地址。对于 MSP430F5xx 和 MSP430F6xx 器件，BSL 位于内存位置 0x1000。有关 MSP430x1xx、MSP430x2xx 和 MSP430x4xx 器件，请参阅节 5。

BSL 运行时，堆栈总是被复位，且 RAM 被清除。BSL 没有禁用中断，因此应用在调用 BSL 之前必须禁用中断。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可能会中断 BSL 应用程序并导致意外行为。一个例子是在具有 USB BSL 且支持 USB 的 MCU 中，Timer\_B 模块用于确定高频晶体的频率。如果在外部应用中也使用了 Time\_B，并且在跳转到 BSL 应用之前没有清除 Time\_B，则可能发生意外行为。

位置 0x1000 可作为 C 函数被调用，如以下示例代码所示：

```
__disable_interrupt(); // disable interrupts  
(void (*)(void))0x1000(); // jump to BSL
```

### 1.4 UART 协议

此处应用的 UART 协议的定义为：

- 在半双工模式下（一次一个发送器），波特率固定为 9600。
- 起始位，8 个数据位（LSB 第一），偶校验位，1 个停止位。
- 握手由确认字符执行。
- 从 MSP430 BSL 接收到字符之后，发送新字符前的最小延时：1.2ms

---

#### 备注

初始化时采用 9600 之外的波特率会导致通信问题或者违反闪存写入时序规范。闪存可能会受到很大的压力，也可能对不可靠的程序或擦除操作做出反应。

---

### 1.5 USB 协议

此处应用的 USB 协议定义为：

- 具有输入端点和输出端点的 HID 协议：每个端点的长度为 64 字节。
- VID：0x2047
- PID：0x0200

## 2 引导加载程序协议 - 1xx、2xx 和 4xx 系列

### 2.1 同步序列

在向 BSL 发送任何命令之前，必须先向 BSL 发送值为 80h 的同步字符 (SYNC)。该字符是计算所有基本的内部参数所必需的，这些参数用于维持 UART 和闪存程序和擦除时序。它提供了 BSL 系统时间基准。当接收到这个字符之后，BSL 会发回确认 DATA\_ACK=90h 以确认接收成功。

此序列必须在每个命令被发送到 BSL 之前完成。

---

#### 备注

该同步字符不是节 2.4 中描述的数据帧的一部分。

---

### 2.2 命令

两类命令可用：需要密码的命令和无需密码的命令。密码保护对任一有可能允许直接或者间接数据访问的命令进行保护。

#### 2.2.1 不受保护的命令

- 接收密码
- 批量擦除 (擦除整个闪存，主存储器以及信息存储器)
- 传输 BSL 版本 (V1.50 或者更高版本或者在可加载 BL\_150S\_14x.txt，而不是 V2.x BSL 中)
- 更改波特率 (V1.60 或 V1.61 或 V2.00，或者在可加载 BL\_150S\_14x.txt 中)

#### 2.2.2 受密码保护的命令

- 接收数据块以对闪存、RAM 或者外设进行编程
- 传输数据块
- 擦除段
- 擦除检查 (存在于 V1.60 或者更高版本或者在可加载 BL\_150S\_14x.txt 中)
- 设定内存偏移 (存在于 V2.12 或者更高版本中)
- 加载程序计数器并启动用户程序
- 更改波特率 (低于 V1.60 但高于 V2.00 的 BSL 版本)

### 2.3 编程流程

在线执行对闪存、RAM、或外设模块区域的写入访问 (RX 数据块命令)。这意味着一旦接收数据字节或字，则立即对其进行处理，并且在下一字节和字完全到达之前完成写入周期。因此，整个写入时间由波特率决定，且无需缓冲机制。

假定位于闪存区域地址下方的数据部分已加载到 RAM 或者外设模块区域中，因此，没有专用闪存控制位受到影响。

---

#### 备注

如果由于线路故障或违反数据帧的规范而失去了对 UART 协议的控制，唯一的恢复方法就是重新运行 BSL 进入队列来启动另外一个 BSL 程序。

---



## 2.4 数据帧

使用一种被称为串行标准协议 (SSP) 的数据帧协议来确保数据传输期间的高数据安全性。在表 2-1 中，BSL 被看成为接收器。

### 2.4.1 数据流结构

- 前八个字节 (从 HDR 到 LH) 是必须的 (xx 代表虚拟数据)。
- 数据字节 D1 至 Dn 是可选的。
- 校验和的两个字节 (CLK 和 CKH) 是必须的。
- 除使用 TX 数据块和 TX BSL 版本命令之外，由 BSL 确认是必需的。

表 2-1. BSL 命令的数据帧<sup>(1)(3)(6)</sup>

接收到的 BSL 命令	HDR	CMD	L1	L2	AL	AH	LL	LH	D1	D2...Dn	CKL	CKH	ACK <sup>(2)</sup>
RX 数据块	80	12	n	n	AL	AH	n - 4	0	D1	D2 ... Dn - 4	CKL	CKH	ACK
RX 密码	80	10	24	24	xx	xx	xx	xx	D1	D2 ... D20	CKL	CKH	ACK
擦除段	80	16	04	04	AL	AH	02	A5	-	- - -	CKL	CKH	ACK
擦除主存储器或者信息存储器	80	16	04	04	AL	AH	04	A5	-	- - -	CKL	CKH	ACK
批量擦除	80	18	04	04	xx	xx	06	A5	-	- - -	CKL	CKH	ACK
擦除检查 <sup>(4)</sup>	80	1C	04	04	AL	AH	LL	LH	-	- - -	CKL	CKH	ACK
更改波特率 <sup>(5)</sup>	80	20	04	04	D1	D2	D3	xx	-	- - -	CKL	CKH	ACK
设定存储器偏移	80	21	04	04	xx	xx	AL	AH	-	- - -	CKL	CKH	ACK
加载 PC	80	1A	04	04	AL	AH	xx	xx	-	- - -	CKL	CKH	ACK
TX 数据块	80	14	04	04	AL	AH	n	0	-	- - -	CKL	CKH	-
BSL 响应	80	xx	n	n	D1	D2 ...	...	...	...	... Dn	CKL	CKH	-
TX BSL 版本 <sup>(4)</sup>	80	1E	04	04	xx	xx	xx	xx	-	- - -	CKL	CKH	-
BSL 响应	80	xx	10	10	D1	D2 ...	...	...	...	... D10	CKL	CKH	-

- (1) 所有数字为采用十六进制符号的字节。
- (2) ACK 由 BSL 发回。
- (3) 同步序列不是数据帧的一部分。
- (4) 擦除检查和 TX BSL 版本命令是 BSL V1.50 或者更高版本 (不包括 2.x BSL) 中的标准命令集的命令。
- (5) 更改波特率命令不是标准命令集的命令 (它在 V1.60 或更高版本或在可载入的 BL\_150S\_14x.txt 中可用)。
- (6) 缩写:

**HDR**: 标头。80h 和 8Fh 之间的任一值 (通常为 80h)。

**CMD**: 命令标识

**L1、L2**: 由 AL 到 Dn 组成的字节的数量。限制: L1 = L2, L1 < 255, L1 偶数

**AL、AH**: 块起始地址或擦除 (校验) 地址或跳转地址 LO 或 HI 字节

**LL、LH**: 纯数据字节数 (最多 250 个) 或擦除信息 LO 或 HI 字节或擦除检查的块长度 (FFFFh 最大值)

**D1 ... Dn**: 数据字节

**CKL、CKH**: 16 位校验和 LO 或 HI 字节

**xx**: 可以是任何数据

- : 无字符 (数据字节) 接收或传输

**ACK**: BSL 返回的确认字符。可以是 DATA\_ACK = 90h : 帧已正确接收, 命令已成功执行, 或 DATA\_NAK = A0h : 帧无效 (例如, 错误的校验和, L1 ≠ L2)、命令未定义、不被允许或执行失败。

**n**: 由 AL 到 Dn 组成的字节的数量

### 2.4.2 校验和

通过 XORing 字 (两个连续字节) 以及反相结果, 在数据帧内所有接收或者发送的字节 B1 至 Bn 上计算出 16-位 (2 字节) 校验和 (除了校验和字节本身)

这意味着 B1 始终为 HDR 字节, 而 Bn 为 CKL 字节之前的最后一个数据字节。

#### 公式

$$\text{CHECKSUM} = \text{INV} [ (B1 + 256 \times B2) \text{ XOR } (B3 + 256 \times B4) \text{ XOR } \dots \text{ XOR } (Bn - 1 + 256 \times Bn) ]$$

或

$$CKL = INV [ B1 XOR B3 XOR \dots XOR Bn - 1 ]$$

$$CKH = INV [ B2 XOR B4 XOR \dots XOR Bn ]$$

### 2.4.3 示例序列

以下示例展示了从地址 0x0F00 读取 MSP430 MCU 存储器的请求。下面显示的所有值均为十六进制。

```
到 BSL :      80
               ( 发送到 BSL 的同步字符 )

来自 BSL :    90
               ( 来自 BSL 的确认 )

到 BSL :      80 14 04 04 00 0F 0E 00 75 E0
               ( 发送命令从 0x0F00 读取存储器，长度为 0x000E )

来自 BSL :    80 00 0E 0E F2 13 40 40 00 00 00 00 02 01 01 01 C0 A2
               ( BSL 返回的值 )
```

### 2.4.4 命令-详细说明

请参阅 [表 2-1](#)。

#### 2.4.4.1 概述

在帧头字节 HDR (80h) 和命令识别 CMD 之后，帧长度字节 L1 和 L2 (这两个数必须相等) 保存 L2 之后的字节数，不包括校验和字节 CLK 和 CKH。

字节 AL, AH, LL, LH, D1...Dn 为专用命令。然而，校验和字节 CLK (低字节) 和 CKH (高字节) 是必须的。

如果数据帧已正确接收且命令执行成功，BSL 会发回确认字符 DATA\_ACK=90h。错误接收的数据帧、失败的操作、以及被锁定或者未定义的命令均由 DATA\_NAK=A0h 确认。

---

#### 备注

低于 V1.30 的 BSL 版本仅支持字节访问操作。因此，无法正确访问 0100h 至 01FFh 的外设模块地址，这是因为它们是以字为单位进行访问的。在版本 V1.30 和更高版本中，地址 0000h 至 00FFh 以字节模式访问；所有其它地址以字模式访问。

---

#### 2.4.4.2 RX 数据块

接收数据块命令用于在 0000h 到 01FFh 对闪存、RAM 或外设模块控制寄存器的任何写入访问。此命令受密码保护。

在 AL (低字节) 和 AH (高字节) 中定义了 16 位偶数块启动地址。在 LL (低字节) 和 LH (高字节) 中定义了 16 位偶数块长度。由于纯数据字节被限制为最多 250，因此 LH 始终为 0。

后面的数据字节是校验和字节 CKL (低字节) 和 CKH (高字节)。如果合适的数据块接收和编程成功，BSL 将发回确认字符 DATA\_ACK。否则，BSL 使用 DATA\_NAK 进行确认。

---

#### 备注

BSL 版本 V1.40 和更高版本支持 MSP430 内部对地址 0200h 至 FFFFh 的在线验证，这减少了 50% 的编程和确认时间。在线验证是指数据立即与写入到闪存的数据进行验证，而无需再次传输。如果出现错误，可载入的引导加载程序 BL\_150S\_14x.txt 还会将第一个错误写入的地址 +3 存储到 RAM 中的错误地址缓冲器中，地址为 0200h (对于 F14x 器件，地址为 021Eh)。

---

#### 2.4.4.3 RX 密码

接收密码命令用于解锁受密码保护的命令，受密码保护的命令执行读取、写入或者段擦除存储器访问。此命令不受密码保护。

由于 32 字节的密码总是位于地址 FFFE0h 至 FFFFh 之间，所以既不需要起始地址，也无需块长度信息。数据字节 D1 至 D20h 包含有从地址为 FFE0h 的 D1 开始的密码信息。

当正确接收到来自主机的包并且包具有有效内容时，BSL 以 DATA\_ACK 进行响应，如表 2-1 所示。DATA\_ACK 不能反映密码是否正确（即，它匹配 FFE0h 到 FFFFh 的内容）或不正确。如果发送了错误的密码，其他命令将以 DATA\_NAK 进行响应，因为 BSL 仍处于锁定状态。

受保护的命令被解锁后，会保持解锁状态，直到启动进入另一个 BSL。

#### 2.4.4.4 批量擦除

批量擦除命令擦除整个闪存区域（主存储器加上信息存储器，请参阅相应的数据表）。此命令不受密码保护。

表 2-1 中显示的所有参数都是必需的。擦除后，BSL 发回确认字符 DATA\_ACK。

批量擦除将密码区域初始化为 0FFh 的 32 倍。

#### 备注

BSL 版本 2.01 和更高版本支持自动清除 LOCKA 位，以保护信息存储器。

当通过冷启动进入 BSL（即通过在 RST 和 TST 引脚上应用 BSL 硬件条目序列）时，LOCKA 位会自动解锁。在 BSL 通信期间执行的批量擦除会擦除信息存储器和主存储器的所有部分。

当通过热启动进入 BSL（即通过从软件函数跳转到 BSL 应用）时，LOCKA 位不会自动解锁。此状态下执行的批量擦除不会擦除信息存储器。因此，当软件调用 BSL 时，用户应用程序必须确保在 BSL 初始化之前清除 LOCKA，以便批量擦除命令可以擦除信息存储器。

#### 2.4.4.5 擦除段

擦除段命令擦除特定的闪存段。此命令收密码保护。

地址字节 AL（低字节）和 AH（高字节）选择合适的段。段内待擦除的任一偶数地址都是有效的。段擦除后，BSL（V1.40 或者更低版本）发回确认字符 DATA\_ACK。

如果擦除不成功，BSL 版本 V1.60 或者更高版本执行相应段的后续擦除检查，并用 DATA\_NAK 进行响应。在这种情况下，第一个未被擦除的地址 +1 存储在 RAM 中的错误地址缓冲器内，地址为 0200h（对于 F14x 器件，地址为 021Eh）。在这个版本中，如果只有一个信息内存段被擦除，则会产生一个问题。在这种情况下，由于对整个信息内存执行了自动擦除检查，所以会报告错误。作为应对，即使擦除报告了错误，要么擦除整个信息存储器，要么在擦除之后进行单独的擦除检查。

擦除段 0 清空密码区域，因此，剩余密码为 0FFh 的 32 倍。

当 LL=0x04 且 LH=0xA5 时，只对擦除主存储器进行批量擦除。实际上，必需执行该命令最少 12 次，才能使得总擦除时间 >200ms。没有对整个主存储器进行后续擦除检查。此外，使用擦除检查命令。有关必需满足的累积（批量）擦除时间和所需的擦除周期的更多信息，请查看器件数据表。

#### 2.4.4.6 擦除主存储器或者信息存储器

擦除主存储器或者信息存储器命令擦除特定的闪存部分。此命令受到密码保护。

地址字节 AL（低字节）和 AH（高字节）选择闪存（主存储器或者信息存储器）的适当部分。此部分内任一待擦除的偶数地址都是有效的。

#### 2.4.4.7 擦除检查

擦除检查命令验证特定地址范围内的闪存存储器是否擦除。此命令受密码保护。

AL（低字节）和 AH（高字节）中定义了 16 位块启动地址。LL（低字节）和 LH（高字节）中定义了 16 位块长度。两者都可以是奇数或者偶数以允许奇数边界检查。

如果对于适当的数据块的擦除检查是成功的（所有字节包含 0FFh），BSL 返回确认字符 DATA\_ACK。否则，BSL 使用 DATA\_NAK 进行确认，并且第一个未被擦除的位置地址 +1 存储在错误地址缓冲器中，地址为 0200h（对于 F14x 器件，地址为 021Eh）。

### 备注

这个命令不是标准命令集中的命令。它在 BSL 版本 V1.60 和更高版本中或在可载入引导加载程序 BL\_150S\_14x.txt 中实现。

#### 2.4.4.8 更改波特率

更改波特率命令提供了在更高波特率（高于默认的 9600 波特率）上进行数据传输的能力。通过更快的数据转换，可实现更短的编程周期，这对于大闪存设备特别重要。此命令不受密码保护。

三个控制字节（D1 至 D3）确定所选择的波特率。D1 和 D2 设定处理器频率（ $f \geq f_{\min}$ ），D3 间接地设定闪存时间发生器频率（ $f_{\text{FTGmin}} \leq f_{\text{FTG}} \leq f_{\text{FTGmax}}$ ）。详细说明：

- D1: F1xx：基本时钟模块控制寄存器 DCOCTL (DCO.2 至 DCO.0)  
 F2xx：基本时钟模块控制寄存器 DCOCTL (DCO.2 至 DCO.0)  
 F4xx：FLL+ 系统时钟控制寄存器 SCFI0 (D, FN\_8 至 FN\_2)
- D2: F1xx：基本时钟模块控制寄存器 BCSCCTL1 (XT2Off, Rsel.2 至 Rsel.0)  
 F2xx：基本时钟模块控制寄存器 BCSCCTL1 (XT2Off, Rsel.2 至 Rsel.0)  
 F4xx：FLL+ 系统时钟控制寄存器 SCFI1 (N<sub>DCO</sub>)
- D3 0：9600 波特  
 1：19200 波特  
 2：38400 波特

接收到数据帧之后，发回一个确认字符 DATA\_ACK，并且 BSL 为所选择的波特率做好准备。TI 建议 BSL 通信程序在波特率更改和下一次数据传输之间等待大约 10ms，以使 BSL 时钟系统有时间稳定下来。

### 备注

可以实现的最高波特率取决于各种系统和环境参数，例如电源电压、温度范围和最小和最大处理器频率。参阅特定器件的数据表。

### 备注

该命令在 BSL 版本 V1.60 或更高版本上实现，或在可加载的引导加载程序 BL\_150S\_14x.txt 中可用。

表 2-2. 针对 MSP430F149 [MSP430F449] 的建议<sup>(1)</sup>

$T_A = 25^\circ\text{C}$ ,  $V_{CC} = 3.0\text{V}$ ,  $f_{\max} = 6.7\text{MHz}$

波特率 (波特)	处理器频率, $f_{\min}$ (MHz) <sup>(4)</sup>	D1 DCOCTL [SCFI0] <sup>(2)</sup>	D2 BCSCCTL1 [SCFI1] <sup>(2)</sup>	D3 <sup>(2)</sup>	编程和验证 60KB (秒) <sup>(3)</sup>
9600 (初始波特率)	1.05	0x80 [00]	0x85 [98]	00 [00]	78+3.7 [0.0]
19200	2.1	0xE0 [00]	0x86 [B0]	01 [01]	39+3.7 [2.4]
38400	4.2	0xE0 [00]	0x87 [C8]	02 [02]	20+3.7 [2.4]

- (1) 括号[]中的值适用于 MSP430F449。  
 (2) D1 至 D3 为采用十六进制表示法的字节。  
 (3) 另外加载、验证和启动可加载的 BSL 需要 3.7 【2.4】秒。  
 (4) 最小处理器频率低于标准 ROM BSL 中的频率（请参阅初始化状态）。

表 2-3. 针对 MSP430F2131 的建议

$T_A = 25^\circ\text{C}$ ,  $V_{CC} = 3.0\text{V}$ ,  $f_{\max} = 6.7\text{MHz}$

波特率 (波特)	处理器频率, $f_{\min}$ (MHz) <sup>(2)</sup>	D1 DCOCTL [SCFI0] <sup>(1)</sup>	D2 BCSCCTL1 [SCFI1] <sup>(1)</sup>	D3 <sup>(1)</sup>	编程和验证 60KB (秒)
9600 (初始波特率)	1.05	0x80	0x85	00	78
19200	2.1	0x00	8x0B	01	39

**表 2-3. 针对 MSP430F2131 的建议 (continued)**

$T_A = 25^\circ\text{C}$ ,  $V_{CC} = 3.0\text{V}$ ,  $f_{\max} = 6.7\text{MHz}$

波特率 (波特)	处理器频率, $f_{\min}$ (MHz) <sup>(2)</sup>	D1 DCOCTL [SCFI0] <sup>(1)</sup>	D2 BCSC11 [SCFI1] <sup>(1)</sup>	D3 <sup>(1)</sup>	编程和验证 60KB (秒)
38400	4.2	0x80	0x8C	02	20

(1) D1 至 D3 为采用十六进制表示法的字节。

(2) 最小处理器频率低于标准 ROM BSL 中的频率 (请参阅[初始化状态](#))。

#### 2.4.4.9 设置内存偏移

对于内存超过 64KB 的器件,特别是 MSP430x 架构器件,可以设置内存指针的偏移量。内存偏移的值用作内存指针的上位字。

内存地址 = 偏移值 << 16 + 实际地址

#### 备注

该命令在 BSL 版本 V2.12 和更高版本上实现。

#### 2.4.4.10 加载 PC

加载程序计数器命令将程序计数器 (寄存器 R0) 定向到整个地址范围内的任何位置。此命令受密码保护。

收到数据帧后,BSL 发回确认字符 (DATA\_ACK)。然后将所选地址移入程序计数器。程序流继续运行,而 BSL 会话终止。

请注意,此时密码保护未激活。跳转到用户应用程序不会重置器件,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),因为时钟模块由 BSL 应用程序设置为以 8MHz 运行,故其闪烁地更快。

#### 2.4.4.11 TX 数据块

传输数据块命令用于在 0000h 至 01FFh 对闪存、RAM、或外设模块控制寄存器的任何读取访问。此命令受密码保护。

在 AL (低字节) 和 AH (高字节) 中定义了 16 位块起始地址。在 LL (低地址) 和 LH (高地址) 中定义了 16 位块长度。由于纯数据字节被限制为最多 250,所以 LH 总是为 0。校验和字节 CKL (低字节) 和 CKH (高字节) 紧随该信息。

现在 BSL 使用被请求的数据块进行响应。传输完 HRD,伪 CMD, L1 和 L2 之后,BSL 发送数据字节 D1 至 Dn,之后发送校验和字节 CKL (低字节) 和 CKH (高字节)。无需确认字符。

#### 2.4.4.12 TX BSL 版本

发送 BSL 版本命令为用户提供了有关芯片识别和引导加载程序软件版本的信息。此命令不受密码保护。

AL、AH、LL、LH 的值可为任意数据,但必须传输这些值,以满足协议要求。校验和字节 CKL (低字节) 和 CKH (高字节) 跟随该信息。

在这之后,BSL 使用 16 字节数据块进行响应。传输完 HRD、伪 CMD、L2 和 L1 之后,BSL 发送数据字节 D1 至 D16 (十进制),之后发送校验和字节 CKL (低字节) 和 CKH (高字节)。无需确认字符。

D1、D2 和 D11、D12 (十进制) 包含了专用信息。

- D1: 器件系列类型 (高字节)
- D2: 器件系列类型 (低字节)
- D11: BSL 版本 (高字节)
- D12: BSL 版本 (低字节)

其余 12 个字节仅供内部使用。

## 2.5 可史可法 BSL

为了升级 BSL 功能，有时适合将更高版本的 BSL 加载到器件的 RAM 中，并应用最近的更新。为此，使用下列 BSL 命令：

- RX 密码 ( 解锁下列命令的密码保护 )
- RX 数据块 ( 可加载 BSL 的代码，代码段地址  $\geq$  220h )
- TX 数据块 ( 用于验证 )
- RX 数据块 ( 从第一个代码段地址获得起始地址 )
- 加载程序计数器 PC ( 使用可已加载的 BSL 的起始地址 )
- 至少等待 5ms，直到新加载的 BSL 执行初始化例程
- RX 密码 ( 解锁加载 BSL 的密码保护 )
- 执行任一命令 ( 使用已加载的 BSL )

以下可加载的 BSL 可用：

- **BL\_150S\_14x.txt** 是具有 BSL 版本 1.10 的 F14x 和 F13x 系列器件的全套 BSL。支持 BSL 版本 V1.60 的所有特征。由于其代码大小大于 1KB，因此只能在 F1x8 和 F1x9 器件使用。用于 RX 块、擦除段和擦除检查命令的错误地址缓冲器地址为 021Eh。BL\_150S\_14x.txt 也可用来替换 PATCH.txt。
- **BS\_150S\_14x.txt** 是具有 BSL 版本 1.10 的 F14x 和 F13x 系列器件的小型 BSL，该 BSL 中，命令集已减少。。由于其代码长度小于 512B，因此可在 F14x 和 F19x 器件中使用。支持 BSL 版本 V1.60 的以下命令：更改波特率、RX 块 ( 带在线验证 )、擦除校验和加载 PC。如果需要 TX 块命令 ( 重定向至 ROM BSL ) ( 例如，用于传输错误地址或独立验证 )，RAM BSL 必须由加载 PC 命令再次调用。用于 RX 块和擦除检查命令的错误地址缓冲器地址为 021Eh。BS\_150S\_14x.txt 也可用于部分替换 PATCH.txt。无需密码，因为 RX 密码命令已删除。

有关下载不同引导加载程序的更多信息，请参阅 [MSP430 中采用闪存硬件和软件提议进行引导加载程序的应用](#)。

第三方软件通过使用可加载的 BSL 来执行大多数功能，例如在线验证、并提高合适的器件的速度。

## 2.6 退出 BSL

要退出 BSL 模式，有两种可能：

- 微控制器在由加载程序计数器命令调用的定义程序地址上继续运行。请注意，此时密码保护未激活。在这种情况下，用户应用程序应该确保闪存被锁定，因为这一操作不是由 BSL 完成。不锁定 BSL 会增加由系统或者软件错误而导致的错误修改闪存的风险。在 2xx 器件上，还应检查 LOCKA 位的正确设置。
- 应用标准的 RESET 序列 ( 请参阅 [图 1-1](#) ) 迫使微控制器从 0FFFEh 地址上的用户复位矢量开始。

## 2.7 密码保护

密码保护禁止任一有可能允许直接或者间接数据访问的命令。进入 BSL 之后，只有诸如批量擦除和 RX 密码 ( 可选，TX BSL 版本和更改波特率 ) 等不受保护的命令可在没有收到正确密码的情况下执行。

采用用于接收正确密码的 RX 密码命令解锁其余的命令。

一旦解锁，它将保持解锁状态直到启动进入另一个 BSL。

密码本身由 16 个位于 FFFE0h 至 FFFFh 地址 ( 256 位 ) 上的中断矢量组成，第一个字节从地址 FFE0h 开始。批量擦除和采用未编程器件之后，所有密码位都为逻辑高电平 (1)。

BSL 版本 2.00 和更好版本具有增强的安全特征。这些特性有位于中断矢量表地址 ( 例如，对于 MSP430F2131，地址为 0xFFDE ) 下方的闪存数据字控制。如果这个字包含：

- 0x0000：如果目标已接收到错误的 BSL 密码，则闪存不会被擦除。
- 0xAA55：BSL 被禁用。这意味着 BSL 不是从 [节 1.3.1](#) 中显示的默认初始化序列开始的。
- 所有其他值：如果传输了错误的密码，则自动擦除整个闪存地址空格。

---

### 备注

在修改中断矢量且启动另一个 BSL 会话之后，用户必须注意更新密码。TI 强烈建议初始化未使用的中断矢量以提高数据安全性。

---

## 2.8 代码保护保险丝

一旦 JTAG 保险丝 ( 代码保护保险丝 ) 熔断，则无法进一步访问 JTAG 测试功能。获得任何内存读取或写入访问权限的唯一方法是通过应用正确的密码来引导加载程序。

然而，BSL 不可能熔断 JTAG 保险丝。如果需要熔断保险丝，可使用 JTAG 编程技术。

## 2.9 BSL 内部设置和资源

下述段落描述了 BSL 内部设置和资源。由于同一器件可以实现不同的 BSL 版本，因此 BSL 通信程序了解相关设置和资源是非常重要的。资源可以依赖于器件 ( 例如，RX 或者 TX 引脚 ) 或者依赖于 BSL 版本 ( 例如，字节或者字访问 )。下述章节描述了可能的变体。

### 2.9.1 芯片识别和 BSL 版本

引导 ROM ( 0FF0h 至 0FFFh ) 的上 16 字节保存有与器件和以 BCD 表示的 BSL 版本号相关的信息。这对于所有器件和 BSL 版本通用：

- 0FF0h 至 0FF1h：芯片识别 ( 例如，F41x 器件的 F413h )。
- 0FFAh 至 0FFBh：BSL 版本号 ( 例如，BSL 版本 V1.30 的版本号为 0130h )。

参阅节 5 中 MSP430 器件到 BSL 版本的分配。

### 2.9.2 外部调用 BSL 的矢量

引导 ROM 的入口部分包含了程序访问 BSL 的调用矢量：

- 0C00h：冷启动矢量 ( 助记符：BR &0C00h ) ( 推荐 )
- 0C02h：热启动矢量 ( 助记符：BR &0C02h )。V1.30 或更高版本。
- 0C04h：供后续使用的矢量。该表是可扩展的。

---

### 备注

热启动不会更改堆栈指针。此外，BSL 状态寄存器未被清除，这可能会导致热启动的 BSL 处于解锁状态。热启动可能只针对高度专业化的实例，在这种情况下，强烈要求 BSL 传话之后返回正在运行的应用程序，而无需重置器件。在几乎所有情况下，最好通过调用冷启动矢量来从用户代码启动 BSL。

---

### 2.9.3 初始化状态

当激活 BSL 时，下列设置生效：

- 停止监视计时器
- 禁用所有中断 (GIE=0)
- **V1.10**  
堆栈指针不会被修改，除非它指向已排除的内存区域。如果是，则初始化为 021Ah。  
**V1.30 或更高版本**  
如果 BSL 由程序通过热启动矢量调用，则堆栈指针不会被修改。如果 BSL 由 BSL RESET 序列启动或者由程序通过冷启动矢量调用，则初始化为 0220h。
- **F1xx**  
确定基本时钟模块，使最低频率为 1.5MHz：  
只在冷启动时  
SR, BCSCCTL1 = 85h ( RSEL = 5, XT2Off = 1 )  
DCOCTL = 80h ( DCO = 4, MOD = 0 )，BCSCCTL2 = 00h：只在冷启动时，SCG1 = 00h ( SMCLK 开启 )
- **F2xx**  
确定基本时钟模块，使最低频率为 1.5MHz：  
只在冷启动时  
SR, BCSCCTL1 = 88h ( RSEL = 8, XT2Off = 1 )  
DCOCTL = 80h ( DCO = 4, MOD = 0 )  
, BCSCCTL2 = 00h：只在冷启动时，SCG1 = 00h ( SMCLK 开启 )
- **F4xx**  
确定 FLL 振荡器和系统时钟，使最低频率为 1.5MHz：  
SCFI0 = 00h ( D = 0, FN\_x = 0 )  
SCFI1 = 98h (N\_DCO)  
SCFQCTL : (M = 0)  
SR : SCG0 = 1 ( FLL 环路控制关闭 )  
只在冷启动时，FLL\_CTL1 = 00h
- SW-UART : Timer\_A 在带有 MCLK 源的连续模式 (Div=1) 下运行  
CCR0 用于比较  
CCTL0 用于 CCIFG0 的轮询
- TX 引脚设置为 RS232 闲置状态输出高电平
- RX 引脚设置为输入
- 受密码保护的命令被锁定 ( 只在冷启动时 )

系统初始化之后，BSL 准备运行，并等待第一个同步序列 (SS)，接着是包含第一个 BSL 命令的数据帧。



## 2.9.4 内存分配和资源

- BSL 程序代码位于引导 ROM 区域 0C00h 至 0FEFh 内。
- 地址 0FF0h 至 0FFFh 保存了器件识别信息。
- BSL 变量占用 RAM 区域
  - 0200h 至 0213h (V1.10)
  - 0200h 至 0219h ( V1.30 或更高版本 )
- BSL 堆栈占用 RAM 区域
  - 0214h 至 0219h (V1.10)
  - 021Ah 至 021Fh ( V1.30 或更高版本 , 仅在冷启动时 )
- 使用的工作寄存器是 :
  - R5 至 R9 ( V1.30 或更低版本 ) 或者
  - R5 至 R10 (V1.40) 或者
  - R5 至 R11 (V1.60) 或者
  - R5 至 R14 ( V2.00 或者更高版本 )

其内容未被缓冲。

- F1xx 和 F2xx :
  - 使用的基本时钟模块寄存器是 :
    - 地址 056h 上的 DCOCTL
    - 地址 057h 上的 BCSCTL1
- F4xx :
  - 使用的 FLL 振荡器和系统时钟寄存器是 :
    - 地址 050h 上的 SCFI0
    - 地址 051h 上的 SCFI1
    - 地址 052h 上的 SCFQCTL
  - 使用的 Timer\_A 控制寄存器是 :
    - 地址 0160h 上的 TACTL
    - 地址 0162h 上的 CCTL0
    - 地址 0170h 上的 TAR
    - 地址 0172h 上的 CCR0
  - 使用的闪存控制寄存器是 :
    - 地址 0128h 上的 FCTL1
    - 地址 012Ah 上的 FCTL2
    - 地址 012Ch 上的 FCTL3
- 对中断服务无影响。

## 3 引导加载程序协议 - F5xx 和 F6xx 系列

### 3.1 BSL 数据包

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都有包装数据，称为外设接口代码 (PI 代码)。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。包装数据和内核命令共同构成 BSL 数据包。

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

### 3.2 UART 外设接口 (PI)

#### 3.2.1 包装器

在每个非 USB MSP430F5xx 器件中编程的默认 BSL430 使用 UART 外设接口 (PI) 进行通信。UART 协议接口具有表 3-1 中显示的格式。所有数字均采用十六进制格式。

表 3-1. UART 协议接口

标头	长度	长度	BSL 内核命令	CKL	CKH	ACK
0x80	NL	NH	请参阅 表 3-5	CKL	CKH	(ACK) 来自 BSL

#### 3.2.2 缩写

CKL、CKH

CRC 校验和中的高字节和低字节。校验和只根据 BSL 内核命令部分中的字节计算。CRC 是使用 MSP430F5xx CRC 模块规格计算的 (有关详细信息, 请参阅 [MSP430F5xx](#) 和 [MSP430F6xx](#) 系列用户指南的 CRC 章节)。

NL、NH

BSL 内核数据包中的字节数, 分成高字节和低字节。

ACK

在收到数据包后由 BSL 发送以确认正确接收数据。这并不意味着 BSL 内核数据是正确的命令或者被正确执行。ACK 仅表示数据包按照预期的格式被格式化, 并且具有正确的校验和。

#### 备注

如果 PI 在接收数据包的任一阶段遇到错误, 它全立即以适当的错误消息进行响应。

#### 3.2.3 消息

BSL430 软件的外设接口部分解析 BSL 数据包的包装部分。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。

表 3-2. UART 错误消息

数据	含义
0x00	ACK
0x51	帧头错误。数据包没有以所需的 0x80 值作为开始。
0x52	校验和错误。数据包没有正确的校验和值。
0x53	数据包大小为零。BSL 核心命令的大小为 0。
0x54	数据包大小超过缓冲器。给定的数据包大小对于 RX 缓冲器来说太大。
0x55	未知错误
0x56	未知波特率。为波特率变化提供的数据不是已知值。

### 3.2.4 接口专用命令

Timer UART 协议当作为 BSL 内核数据传输时也接受下列命令。

BSL 命令	CMD	AL	AM	AH	数据
更改波特率	0x52	-	-	-	D1

#### 3.2.4.1 更改波特率

此命令会更改所有后续接收到的数据包的波特率。命令通过单个 ACK ( 以当前波特率而非新选择的波特率发送 ) 或错误字节进行确认。预计无后续消息包。

D1 : 有效值

- 0x02 : 9600
- 0x03 : 19200
- 0x04 : 38400
- 0x05 : 57600
- 0x06 : 115200

## 3.3 I<sup>2</sup>C 外设接口

### 3.3.1 I<sup>2</sup>C 协议定义

BSL 使用的 I<sup>2</sup>C 协议定义是 :

- 主服务器必须向 BSL 从服务器请求数据。
- 采用 7 位寻址方式, 从服务器侦听地址 0x48。
- 除了硬件 ACK 之外, 握手由确认字符执行。
- 从 MSP430 BSL 接收到字符之后, 发送新字符的最小延时时间为 1.2ms。
- BSL 不需要重复启动, 但可以使用。

### 3.3.2 具有字节级确认的基本协议

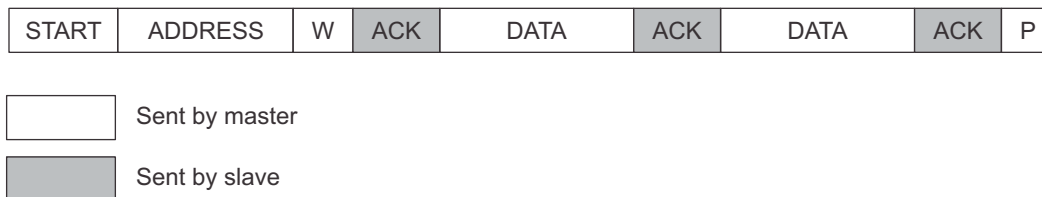


图 3-1. 基本协议 - 字节级确认

1. 发送开始位
2. 发送从机地址
3. 发送读 (R)-1 或写 (W)-0 位。
4. 等待或发送确认位
5. 发送或接收数据字节 ( 8 位 )

6. 期待或发送确认位
7. 发送停止位

### 3.3.3 BSL 的 I<sup>2</sup>C 协议 - 从从服务器读取

1. 发送启动序列 (S)
2. 发送从服务器的 I<sup>2</sup>C 地址，该地址具有低读/写位 (偶数地址) (ADDR) + W。
3. 发送从服务器寄存器内部地址的数据或地址 (NDATA)
4. 重新发送启动序列 (重复启动) (分辨率)
5. 发送从服务器的 I<sup>2</sup>C 地址，该地址具有高读/写位 (奇数地址) (ADDR)
6. 从从服务器读取数据字节 (RDATA)
7. 发送停止序列 (P)

来自主机 (主器件) 的所有 BSL 包装器命令都被视为数据 (表示为 NDATA，它们可以由 n 个字节组成)，从从服务器读取的所有数据也被视为数据并表示为 RDATA。

从主服务器到从服务器的所有通信的协议是：

从主服务器 → S + ADDR + W + NDATA + ReS + ADDR + R

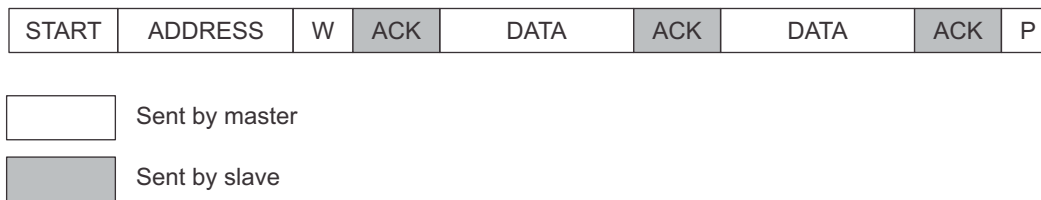
从从服务器 → RDATA

从主服务器 → P

### 3.3.4 确认 (ACK)

有两种确认级别。

- 低级确认表示接收每个作为 I<sup>2</sup>C 协议一部分的字节。如果从寄存器上设置了正确的 I<sup>2</sup>C，则由硬件管理。



**图 3-2. 字节级确认**

- 高级确认表示获得的 BSL 内核命令校验和正确且符合预期。在某些情况下，此确认可以表示命令已正确执行。这是 RDATA 的第一个字节。如果这是 NAK (0x00 除外)，则表示未收到正确的命令，主机应将该命令传输视为失败。如果这是 ACK (0x00)，则表示命令的发送或接收正确且校验和正确，并且紧随其后的数据是来自从机的响应 (如果有)。如果从机在响应命令之前需要时间进行处理，它可以将 CLK 线路保持为低电平。

### 3.3.5 包装器

BSL 数据包的包装器集成了常用的 UART BSL 核心命令包，但添加了长度、校验和以及确认，以在 I<sup>2</sup>C 通信中使用（请参阅表 3-3）。

表 3-3. 用于 I<sup>2</sup>C 的 BSL 内核命令包装器

标头	长度	长度	BSL 内核命令	CKL	CKH	ACK
0x80	NL	NH	请参阅 节 3.5	CKL	CKH	(ACK) 来自 BSL

#### CKL、CKH

CRC 校验和高字节和低字节。校验和仅根据 BSL 内核命令部分中的字节计算。

#### NL、NH

BSL 内核数据包中的字节数，分为高字节和低字节。

#### ACK

由 BSL 在接收到数据包之后发出以确认正确接收数据。这并不意味着 BSL 内核数据是正确的命令或已被正确执行。ACK 只是表示数据包按预期格式化且具有正确的校验和。

## 3.4 USB 外设接口

### 3.4.1 包装器

USB 引导加载程序的外设接口具有表 3-4 中所示的包装器格式。USB BSL 没有专用接口命令或回复。唯一的变量字节 NL 应描述包含在 BSL 内核命令数据包内的字节数。

表 3-4. USB 外设接口

标头	长度	BSL 内核命令
0x3F	NL	请参阅 节 3.5

### 3.4.2 硬件要求

USB 外设接口要求在 XT2 上使用高频振荡器。为了使 BSL 正常工作，振荡器可以是 24MHz，12MHz，8MHz，或者 4MHz。

## 3.5 BSL 内核命令结构

BSL 内核命令以表 3-5 中显示的格式传输。所有数字均采用十六进制格式。

#### 备注

有关命令与 MSP430F5438 (非 A 版本) 中的 BSL 如何一起使用的方法，请参阅节 5.5。

表 3-5. BSL 内核命令

BSL 命令	CMD	AL	AM	AH	数据
RX 数据块	0x10	(AL)	(AM)	(AH)	D1 ...Dn
RX 数据块快速	0x1B	(AL)	(AM)	(AH)	D1 ...Dn
RX 密码	0x11	-	-	-	D1 ...D33
擦除段	0x12	(AL)	(AM)	(AH)	-
解锁和锁定信息	0x13	-	-	-	-
保留	0x14	-	-	-	-
批量擦除	0x15	-	-	-	-
CRC 检查	0x16	(AL)	(AM)	(AH)	长度 (低字节)，长度 (高字节)
加载 PC	0x17	(AL)	(AM)	(AH)	-
TX 数据块	0x18	(AL)	(AM)	(AH)	长度 (低字节)，长度 (高字节)

表 3-5. BSL 内核命令 (continued)

BSL 命令	CMD	AL	AM	AH	数据
TX BSL 版本	0x19	-	-	-	-
TX 缓冲器大小 <sup>(1)</sup>	0x1A	-	-	-	-

(1) 当前未在 F5xx 和 F6xx MCU 上的 BSL 中执行 TX 缓冲区大小命令。

### 备注

在闪存中编程的且通过 USB 通信的 BSL 仅包含表 3-5 中所示的命令子集。这些命令可用于将完整的 BSL 载入到 RAM 以进行闪存编程。这个子集内的命令为 RX DATA BLOCK FAST、RX PASSWORD 和 LOAD PC。

所支持的特性也可以由 BSL 版本号来确定，如节 3.7.3 中所示。与本文档相关联的 zip 文件中给出了如何将功能完善的 BSL 载入 RAM 中的示例（请参阅节 1.1）。

### 3.5.1 缩写

-

不需要数据。不应给出延迟，并且任一随后需要的数据都应作为直接下一个字节发送。

AL, AM, AH

地址字节。分别为地址的低字节、中字节和高字节。

D1 ...Dn

数据字节 1 到 n (注意：n 必须比 BSL 缓冲器大小小 4)

长度

字节，包含 1 到 255 的值，并描述要在 CRC 传送或使用的字节数。在多长度字节情况下，它们如描述的那样组合在一起以形成一个更大的值来描述所需字节数。

### 3.5.2 命令说明

RX 数据块

BSL 内核从地址字段中指定的位置开始写入字节 D1 到 Dn。

RX 数据块快速

除了没有表明数据被正确编程的应答之外，该命令与 RX 数据块命令一样。它主要被用于加速 USB 编程。

RX 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 个字向匹配，BSL 内核则接收包含在数据包内的密码，并解锁 BSL 保护命令。当输入错误密码时，会启动批量擦除。这意味着所有的代码闪存被擦除，但不会擦除信息内存。

擦除段

包含给定地址的闪存段被擦除。

## 解锁和锁定信息

该命令使得 INFO\_A 锁切换以保护或者锁定 INFO\_A 段。有关此锁定的更多细节，请参阅 [MSP430F5xx](#) 和 [MSP430F6xx 系列用户指南](#)。该命令必须在 INFO\_A 的擦除段命令之前发送，但在批量擦除之前则不需要。

## 擦除块

包含给定地址的闪存块被擦除。

## 批量擦除

MSP430 MCU 中的所有代码闪存均被擦除。该功能**不会**擦除信息内存。

## CRC 检查

该 MCU 使用 CCITT 标准执行 16 位 CRC 检查。给定的地址为 CRC 检查的第一个字节。两个字节用于长度。

## 加载 PC

使用 CALLA 指令使 BSL 在指定地址开始执行。由于 BSL 代码随这个指令立即退出，所以无法预计内核响应。

## TX BSL 版本

BSL 传输其版本信息 ( 更多细节请参阅 [节 3.7.3](#) ) 。

## TX 缓冲器大小

BSL 传输一个值，该值表示其数据缓冲器中用于发送或接收 BSL 核心数据包的可用字节数。

## 3.6 BSL 安全

### 3.6.1 受保护的命令

为了保护器件内的数据，大多数内核命令都受到保护。只有在通过发送带有正确密码的 RX 密码命令来解锁器件之后，受保护的命令才能成功完成。此外，专用于外设接口的命令不受保护。

### 不受保护的命令

#### RX 密码

#### 批量擦除

#### 受密码保护的命令

#### RX 数据块到地址 ( 闪存或者 RAM )

#### TX BSL 版本

#### TX 数据块

#### 擦除段

#### 擦除库

#### 设定程序计数器

#### 切换 INFO\_A 锁

#### 擦除主存储器

#### CRC 校验

### 3.6.2 RAM 擦除

启动时，BSL 执行 RAM 擦除，将常量字写入器件中的某些 RAM 位置。通常这些是系列产品中最小的共享 RAM 地址。有关专用器件信息，请参阅 [节 5](#)。

## 3.7 BSL 内核响应

BSL 内核响应始终包装在外设接口包装器中，其格式与接收到的命令的格式相同。BSL 内核可以以 [表 3-6](#) 中显示的格式进行响应。所有数字采用十六进制格式。

表 3-6. BSL 内核响应。

BSL 响应	CMD	数据、
数据块	0x3A	D1 ...Dn
BSL 版本	0x3A	D1 ...D4
CRC 值	0x3A	DL , DH
缓冲器大小	0x3A	NL , NH
消息	3x0B	MSG

### 3.7.1 缩写

#### CMD

必填字段，用于区分 BSL 消息和 BSL 数据传输。

#### 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、Dx

数据字节，包含所请求数据。

#### DL、DH

分别为请求的 16 位 CRC 值的数据低字节和高字节。

#### NL、NH

数据字节，描述缓冲器尺寸的长度。为了管理超过 255 的尺寸，尺寸划分为低字节和高字节。



### 3.7.2 BSL 内核消息

表 3-7 描述了 BSL 内核消息。

**表 3-7. BSL 内核消息**

MSG	含义
0x00	成功运行
0x01	闪存写入校验失败。编程后，对编程数据运行 CRC。如果 CRC 与预期结果不匹配，则返回该错误。
0x02	闪存失败位设定。设置闪存控制器中 FAIL 位的操作（有关闪存失败位的更多详情，请参阅 <a href="#">MSP430F5xx</a> 和 <a href="#">MSP430F6xx</a> 系列用户指南）。
0x03	编程期间的电压变化。在请求的写入操作期间设置 VPE（有关 VPE 位的更多详情，请参阅 <a href="#">MSP430F5xx</a> 和 <a href="#">MSP430F6xx</a> 系列用户指南）。
0x04	BSL 被锁定。尚未提供正确的密码来解锁 BSL。
0x05	BSL 密码错误。尝试解锁时向 BSL 提供了错误的密码。
0x06	禁止字节写入。当在闪存区域中尝试写入字节时，返回该错误。
0x07	未知命令未识别出发给 BSL 的命令。
0x08	数据包长度超过缓冲器大小。提供的数据包长度值太大以至于无法在 BSL 接收缓冲器中保存。

### 3.7.3 BSL 版本号

BSL 版本号是一个 4 字节数组。

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

解释 BSL 内核命令的代码部分的版本号。

字节 3：API 版本

读取和写入 MSP430 MCU 存储器的代码部分的版本号。

保留位：

0x00 表示该 BSL API 与闪存接口。

0x30 表示该 BSL API 与 FRAM 接口：

0x80 表示该 BSL 只能执行下列命令：

RX 数据块快速（并且只能写入 RAM）

RX 密码

设定 PC

字节 4：外设接口版本

管理通信的代码部分的版本号。

保留编号：

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\_B 的 I<sup>2</sup>C

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

0xB0 到 0xCF：表示基于 eUSCI 的 I<sup>2</sup>C 和 UART

### 3.7.4 UART BSL 的示例序列

#### 备注

示例序列中的所有值均为十六进制。

将波特率改为 9600

主机：80 02 00 52 02 90 55

BSL：00

获得缓冲器大小

主机：80 01 00 1A 8B 52

BSL：00 80 03 00 3A 04 01 1D 12

获得 BSL 版本

主机：80 01 00 19 E8 62

BSL：00 80 05 00 3A 00 01 01 01 6C 4F

解锁 BSL 的 RX 密码

主机：80 11 00 11 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 5C 38 4F

BSL：00 80 02 00 3B 00 60 C4

### 3.8 BSL 公共函数和 Z 区域

BSL Z 区是可从应用程序代码读取并调用的一小部分内存。它位于内存地址 0x1000 到 0x100F 之间。

内存位置 0x1000 包含指向 BSL 开始的跳转指令，它可以用于从正在运行的应用中调用 BSL。

存储器位置 0x1002 包含前往“BSL 操作”函数的跳转。需要三个参数来调用此操作函数。第一个参数是描述哪个函数的编号，后两个参数只是已知值，用以表明该函数被有意调用。

R12：函数编号

R13：0xDEAD

R14：0xBEEF

#### 3.8.1 从外部应用中启动 BSL

将程序计数器设定到内存位置 0x1000 来启动 BSL。堆栈总是被复位，且 RAM 被清除。应该注意的是，GIE 位没有被禁用，所以如果不需要中断，应该通过调用应用来完成该操作，如果使用中断，则应该从“返回到 BSL”适当地返回。

由于堆栈被复位，位置 0x1000 也可以作为 C 函数被调用，如下述示例代码所示：

```

_disable_interrupt();
((void (*)(void))0x1000)();
    
```

如果在调用 USB BSL 之前 USB 堆栈正在运行，则必须先断开此 USB 堆栈。下述示例显示了 C 中建议的序列：

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可能会中断 BSL 应用程序并导致意外行为。一个示例是在 USB BSL 中，Timer\_B 模块用于时钟初始化。如果外部应用程序也使用了 Timer\_B，则可能会导致 BSL 初始化失败。

```

_disable_interrupt();
USBKEYPID = 0x9628; // Unlock USB configuration registers
USBCNF &= ~PUR_EN; // Set PUR pin to hi-Z, logically disconnect from host
USBPWRCTL &= ~VBOFFIE; // Disable VUSBoff interrupt
    
```

```
USEKEYPID = 0x9600;      // Lock USB configuration register
__delay_cycles(500000);
((void (*)(void*))0x1000)(); // Call BSL
```

TI 建议在各种主机上测试此序列。

### 3.8.2 返回到 BSL 函数描述

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 已将程序写入闪存或 RAM 中，并通过“设置 PC”启动该程序，且该程序需要返回到 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 ; remove first word from return addr POP.W RET_high
                          ; remove second word from return addr RETA
                          ; should now return to the BSL location
```

## 4 引导加载程序硬件

本章介绍了简单且价廉的硬件和软件解决方案，其通过 PC 串行端口 (RS-232) 来访问 MSP430 闪存器件的引导加载程序函数。

### 4.1 硬件描述

本文档中介绍的低成本硬件 (请参阅图 4-1) 主要由一个低压降电压稳压器、一些逆变器 and 运算放大器组成。还有一些电阻器、电容器和二极管。表 4-4 列出了所需的部件。

下面各节将详细地描述这些功能块。

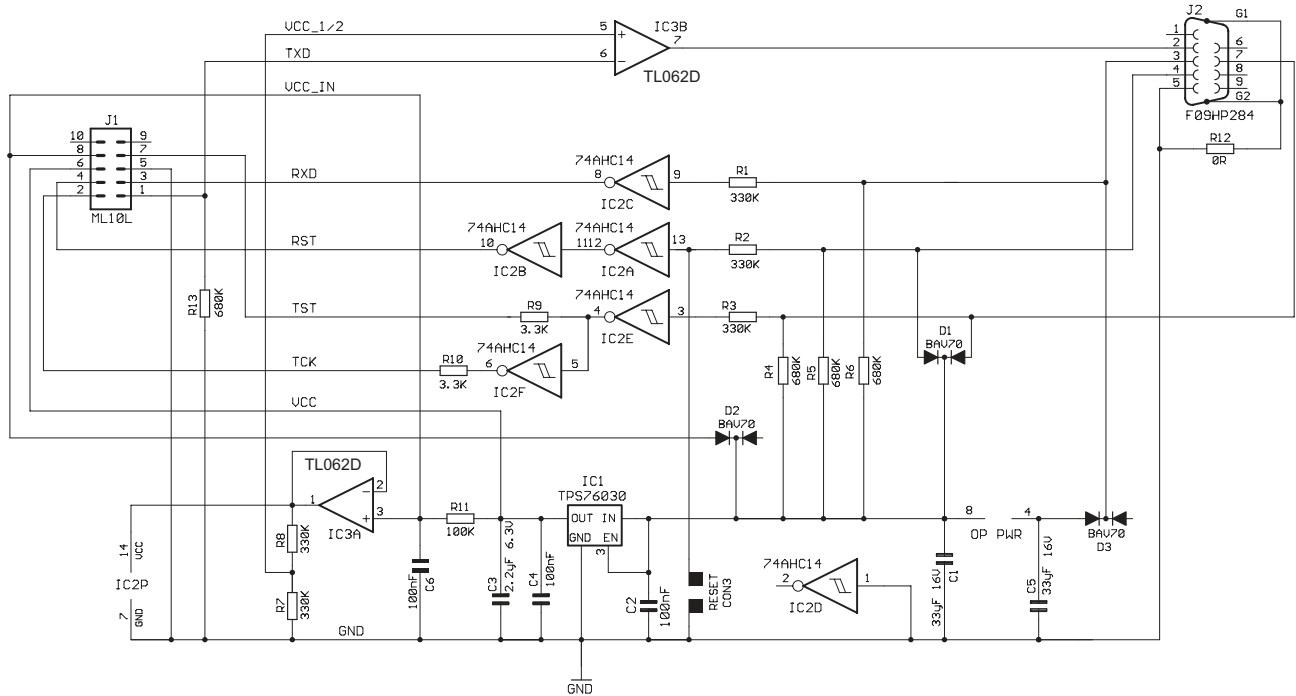


图 4-1. 引导加载程序接口原理图

#### 4.1.1 电源

可从 RS-232 接口为引导加载程序硬件供电。RS-232 信号 DTR (串行连接器的引脚 7) 和 RTS (串行连接器的引脚 4) 通常向负载电容器 C1 提供正电压，并为低压降稳压器 IC1 供电 (TI TPS76030 或者 LP2980-3.0, 或者等效的 3V 低压降稳压器)。

使用相当大的电容器，可以吸收高于驱动串行端口所能提供的短时电流。例如，需要该特征对闪存进行编程。

还可以将外部电源电压连接到 BSL 目标连接器 (J1) 引脚 8 的硬件上。二极管用于防止极性反流。

## 4.1.2 串行接口

表 4-1 显示了用于与引导加载程序通信的信号 ( 通过连接器 J2 ) 。这些名称反映了可从 PC 上看见的引脚功能。例如, PC 通过 RxD 引脚接收数据, 而引导加载程序需要驱动该信号。

表 4-1. 串行端口信号和引脚分配

引脚名称	全称 (PC)	9 引脚 SUB-D	BSL 接口上的功能
RxD	接收数据	2	向 PC 传输数据
TxD	传输数据	3	从 PC ( 和负电源 ) 接收数据
DTR	数据终端就绪	4	复位控制 ( 和正电源 )
RTS	请求发送	7	TEST 或者 TCK 控制 ( 和正电源 )
GND	接地	5	接地

### 4.1.2.1 电平移动

带有施密特触发器特性的简单 COMS 逆变器 (IC2) 用于将 RS-232 电平 ( 请参阅表 4-2 ) 转变为 CMOS 电平。

表 4-2. RS-232 电平

逻辑电平	RS-232 电平	RS-232 电压电平
1	标记	-3V 至 -15V
0	空格	3V 至 15V

逆变器由运算放大器 IC3A 供电。该放大器允许将提供的逻辑电平调整为连接的目标应用程序所需的电平。施加到 BSL 目标连接器 (VCC\_IN) 引脚 8 上的电压覆盖 IC1 和 100kΩ 串联电阻器 R11 提供的默认 3V 电平。因此, 运算放大器的输出电压被拉至施加的电压 VCC\_IN。

根据所选器件系列的过压保护, 过多的 ( 如 TI 74AHC14 中 ) 。如果保护二极管传导至 V<sub>CC</sub>, 运算放大器 IC3A 需要对过压进行补偿。因此, TI 推荐 74AHC14 器件, 该器件会传导接地 (GND)。

为了避免过多的功耗以及保护二极管的损坏, 使用串联二极管 ( R1, R2 和 R3 ) 来限制输入电流。

采用运算放大器 (IC3B) 从 CMOS 电平中生成 RS-232 电平。正输入上的电平设置为 V<sub>CC</sub>/2 ( 标称电压 1.5V ) 。如果负输入上的电平上升至高于这个电平, 输出被拉至运算放大器的负电源 ( 标记 ) 。如果电平下降到低于 V<sub>CC</sub>/2, 输出被拉至正轨 ( 空白 ) 。

运算放大器的正电源与稳压器的输入一样。独立电容器 (C5) 用于生成负电源电压。该电容器由引导加载程序硬件 ( SUB-D 连接器 J2 上的引脚 3 ) 的接收信号充电。

在异步串行通信中, 停止位和开始位的组合用于同步发送器和接收器。在数据字节输送后, 停止位强制传输线路进入定义的状态, 通常是逻辑 1, 或 RS-232 术语中的标记。这意味着当没有传输并且电容器可以充电时, 传输线路电压为负。二极管被用于防止电容器传输期间放电。

在极少数情况下, 发送到引导加载程序接口的数据可能包含过多的零, 从而使负电源所需的电容器 C5 放电, 引起接口故障。( 一个可能的应变方法是以较小的数据块发送数据。 ) 然而, 在正常运行条件下, 即使是包含所有零的数据也不会造成问题。

### 4.1.2.2 对 $\overline{RST}/NMI$ 和 TEST 或者 TCK 引脚的控制

用于调用 MSP430 MCU 的引导加载程序软件的两个引脚 —  $\overline{RST}/NMI$  和 TEST 或 TCK ( 用于没有专用 TEST 引脚的器件 ) 分别由 DTR 和 RTS 信号控制。这些信号还提供正电压为引导加载程序硬件供电。

对于带有专用 TEST 引脚的器件, 正常运行期间  $\overline{RST}/NMI$  和 TEST 上的电平分别为逻辑 1 和逻辑 0。为了实现这些电平并使用相应的 RS-232 信号作为电源线路, 有必要为  $\overline{RST}/NMI$  引脚使用两个逆变器 ( IC1A、IC2B ) 并为 TEST 引脚使用一个逆变器 (IC2E)。

没有 TEST 引脚的器件需要采用其 TCK 引脚上的倒置 TEST 引脚序列来调用引导加载程序。因此, 相应信号被逆反 ( 逆变器 IC2F ) 。

二极管防止电容器 C1 放电以实现 RS-232 线路 ( RTS 和 DTR ) 的控制。

### 4.1.3 目标连接器

表 4-3. 目标连接器的引脚分配<sup>(1)</sup>

引脚	信号名称	带有测试引脚的器件	MSP430F13x 或 MSP430F14x 上的引脚	MSP430F4xx 上的引脚
1	TXD	P1.1	P1.1	P1.0
2	TCK	不要连接 <sup>(2)</sup>	TCK	TCK
3	RXD	P2.2	P2.2	P1.1
4	RST	$\overline{\text{RST/NMI}}$	$\overline{\text{RST/NMI}}$	$\overline{\text{RST/NMI}}$
5	GND	GND	GND	GND
6	V <sub>CC</sub> (3.0V)	V <sub>CC</sub> <sup>(3)</sup>	V <sub>CC</sub> <sup>(3)</sup>	V <sub>CC</sub> <sup>(3)</sup>
7	TST	TEST	不要连接	不要连接
8	VCC_IN	V <sub>CC</sub> <sup>(3)</sup>	V <sub>CC</sub> <sup>(3)</sup>	V <sub>CC</sub> <sup>(3)</sup>
9	未连接	—	—	—
10	未连接	—	—	—

- (1) 有关专用器件的 BSL 引脚信息，请参阅可用器件数据表。
- (2) 严禁将信号 TCK 连接到具有 TEST 引脚的器件上。
- (3) 引脚 V<sub>CC</sub>(3.0V) 是能够提供有限电流的电压源，具体取决于串行端口驱动能力。如果使用外部电源，严禁将 V<sub>CC</sub>(3.0V) 连接到目标上。在这种情况下，外部电源必须连接到引脚 VCC\_IN。否则，必须断开 VCC\_IN。

#### 4.1.4 部件列表

表 4-4 列出了 BSL 接口部件。

表 4-4. 通用 BSL 接口部件列表

器件	值或部件型号	封装	说明
C1	33 $\mu$ F, 16V	SMD7243	
C2	100nF	SMD0805	
C3	2.2 $\mu$ F, 6.3V	SMD1206	
C4	100nF	SMD0805	
C5	33 $\mu$ F, 16V	SMD7243	
C6	100nF	SMD0805	
D1	BAV70	SOT23	高速双二极管
D2	BAV70	SOT23	高速双二极管
D3	BAV70	SOT 23	
IC1	TPS76030	SOT 23-5	TI
IC2	74AHC14	SO14	TI
IC3	TL062D	SO8	TI
R1	330k $\Omega$	SMD0805	
R2	330k $\Omega$	SMD0805	
R3	330k $\Omega$	SMD0805	
R4	680k $\Omega$	SMD0805	
R5	680k $\Omega$	SMD0805	
R6	680k $\Omega$	SMD0805	
R7	330k $\Omega$	SMD0805	
R8	330k $\Omega$	SMD0805	
R9	3.3k $\Omega$	SMD0805	
R10	3.3k $\Omega$	SMD0805	
R11	100k $\Omega$	SMD0805	
R12	0k $\Omega$	SMD0805	
R13	680k $\Omega$	SMD0805	
J1	标头 2x5	2X05	目标连接器 ( 请参阅表 4-3 )
J2	F09HP284	9-SUB-D 母头	RS-232 连接器
CON3	重置	SMD0805	用于连接可选重置按钮的焊盘

## 5 器件和引导加载程序版本间的差异

### 5.1 1xx、2xx 和 4xx BSL 版本

本节中的表格显示了 MSP430 器件到与其硬件和软件资源相关的 BSL 版本分配的关键信息。

**表 5-1. F13x、F14x(1) (不包括修订版本 AA)、F11x 和 F11x1 上的 BSL 版本 1.10**

器件		F13x F14x(1) 最高至修订版 N	F11x (不再使用) F11x1 (不再使用)
BSL 版本		1.10	
BSL 矢量地址	冷启动	0C00h	
	暖启动	—	
芯片 ID 地址		0FF0h	
芯片 ID 数据		F149h	F112h
BSL 版本地址		0FFAh	
BSL 版本数据		0110h	
批量擦除时间, 额定值 (ms)		17.2 <sup>(1)</sup>	
以 0000h 对 FFFFh 进行的读写访问		字节	
写入时验证 (在线)		否	
堆栈指针初始化	关键 SP	021Ah	
	非关键 SP	未更改	
<b>BSL 所使用的资源</b>			
发送引脚 (TX), 接收引脚 (RX)		P 1.1, P2.2	
所使用的 RAM 堆栈		0200h 至 0219h	
工作寄存器		R5 至 R9	
系统时钟, 受影响的控件		BCSCTL1, DCOCTL	
Timer_A, 受影响的控件		TACTL, TAR, CCTL0, CCR0	
为软件调用所做的准备		<pre> mov    #00h, &amp;CCTL0 bic.b  #02h, &amp;P1SEL bic.b  #04h, &amp;P2SEL bic.b  #32h, &amp;IE1 mov.b  #00h, &amp;BCSCTL2 mov    #00h, SR br     &amp;0C00h           </pre>	
注释 1 应变方法命令		载入 PATCH.TXT 来消除 ROM 错误 (请参阅节 5.2 和节 2.5)	
注释 2 仅可选 F148 和 F149: 使用可加载 BSL (要求大于 1KB 的 RAM)		载入 BL_150S_14x.txt 以获得 V1.60 的所有特性和有效擦除段命令 (请参阅节 2.5)。	
注释 3 可选 F1x4 至 F1x9: 使用小的可加载 BSL (要求小于 512B 的 RAM)		载入 BS_150S.14x.txt 来获得 V1.60 的一些特性 (请参阅节 2.5)。	

(1) 为了达到数据表中规定的所需批量擦除时间, 必须多次执行批量擦除命令。



表 5-2. F41x、F11x 和 F11x1 上的 BSL 版本 1.30

器件		F41x	F11x (不再使用) F11x1A
BSL 版本		1.30	
BSL 矢量地址	冷启动	0C00h	
	暖启动	0C02h	
芯片 ID 地址		0FF0h	
芯片 ID 数据		F143h	F112h
BSL 版本地址		0FFAh	
BSL 版本数据		0130h	
批量擦除时间, 额定值 (ms)		206.4	
读取和写入访问地址	0000h 至 00FFh	字节	
	0100h 至 FFEh	字	
写入时验证 (在线)		否	
堆栈指针初始化	冷启动	0220h	
	暖启动	未更改	
<b>BSL 所使用的资源</b>			
发送引脚 (TX)		P1.0	P1.1
接收引脚 (RX)		P2.1	P2.2
所使用的 RAM 堆栈		0200h 至 021Fh	
工作寄存器		R5 至 R9	
系统时钟, 受影响的控件		SCF10, SCF11, SCFQCTL	BCSCTL1, DCOCTL
Timer_A, 受影响的控件		TACTL, TAR, CCTL0, CCR0	
为软件调用所做的准备		<pre> mov    #00h, &amp;CCTL0 mov.b  #00h, &amp;FLLCTL1 br     &amp;0C00h </pre>	<pre> mov    #00h, &amp;CCTL0 mov.b  #00h, &amp;BCSCTL2 mov    #00h, SR br     &amp;0C00h </pre>

**表 5-3. F12x 上的 BSL 版本 1.40**

器件		F122 , F123x
BSL 版本		1.40
BSL 矢量地址	冷启动	0C00h
	暖启动	0C02h
芯片 ID 地址		0FF0h
芯片 ID 数据		F123h
BSL 版本地址		0FFAh
BSL 版本数据		0140h
批量擦除时间, 额定值 (ms)		206.4
读取和写入访问地址	0000h 至 00FFh	字节
	0100h 至 FFFEh	字
写入时验证 (在线)		地址 0200h 至 FFFEh
堆栈指针初始化	冷启动	0220h
	暖启动	未更改
<b>BSL 所使用的资源</b>		
发送引脚 (TX)		P1.1
接收引脚 (RX)		P2.2
所使用的 RAM 堆栈		0200h 至 021Fh
工作寄存器		R5 至 R10
系统时钟, 受影响的控件		BCSCTL1, DCOCTL
Timer_A, 受影响的控件		TACTL, TAR, CCTLO, CCR0
为软件调用所做的准备		<pre> mov.b #00h, &amp;BCSCTL2 mov    #00h, SR br     &amp;0C00h                     </pre>

**表 5-4. F11x2、F12x2、F43x、F44x、FE42x、FW42x、F43x、FG43x、F415、F417 上的 BSL 版本 1.60**

器件		F1122、 F1132	F1222、 F1232	F43x、 F44x	FE42x、 FW42x、 F415、 F417	F43x、 FG43x
BSL 版本		1.60				
BSL 矢量地址	冷启动	0C00h				
	暖启动	0C02h				
芯片 ID 地址		0FF0h				
芯片 ID 数据		1132h	1232h	F449h	F427h	F439h
BSL 版本地址		0FFAh				
BSL 版本数据		0160h				
批量擦除时间, 额定值 (ms)		206.4				
读取和写入访问地址	0000h 至 00FFh	字节				
	0100h 至 FFFEh	字				
写入时的验证 (在线)		地址 0200h 至 FFFEh				
擦除校验命令		是 (错误地址 0200h)				
擦除段命令		带有擦除验证 (错误地址 0200h)				
TX 识别命令		支持				
更改波特率命令		支持				
堆栈指针初始化	冷启动	0220h				
	暖启动	未更改				
<b>BSL 所使用的资源</b>						
发送引脚 (TX)		P1.1		P1.0		
接收引脚 (RX)		P2.2		P1.1		
所使用的 RAM 堆栈		0200h 至 021Fh				
工作寄存器		R5 至 R12				
系统时钟, 受影响的控件		BCSCTL1, DCOCTL		SCFI0, SCFI1, SCFQCTL		
Timer_A, 受影响的控件		TACTL, TAR, CCTLO, CCR0				
为软件调用所做的准备		mov.b #00h, &BCSCTL2 mov #00h, SR br &0C00h		mov.b #00h, &FLLCTL1 br &0C00h		
注释	擦除段命令	连续验证地址 1000h 至 11FFh (三个段)。也使用擦除校验命令。				

**表 5-5. F16x、F161x、F42x0、F13x 修订版 AA、F14x(1) 修订版 AA、F47x、FG47x 上的 BSL 版本 1.61**

器件		F16x	F161x	F149 修订 版本 AA	F42x0	F41x2	F47197	FG47x
BSL 版本		1.61						
BSL 矢量地址	冷启动	0C00h						
	暖启动	0C02h						
芯片 ID 地址		0FF0h						
芯片 ID 数据		0F169h	0F16Ch	F149h	F427h	4152h	F47Fh	0F479h
BSL 版本地址		0FFAh						
BSL 版本数据		0161h						
批量擦除时间, 额定值 (ms)		206.4						
读取和写入访问地址	0000h 至 00FFh	字节						
	0100h 至 FFFEh	字						
写入时验证 (在线)		地址 0200h 至 FFFEh						
擦除校验命令		是 (错误地址 0200h)						
擦除段命令		带有擦除验证 (错误地址 0200h)						
TX 识别命令		支持						
更改波特率命令		支持						
堆栈指针初始化	冷启动	0220h						
	暖启动	未更改						
<b>BSL 所使用的资源</b>								
发送引脚 (TX)		P1.1			P1.0			
接收引脚 (RX)		P2.2			P1.1			
所使用的 RAM 堆栈		0200h 至 021Fh						
工作寄存器		R5 至 R14						
系统时钟, 受影响的控件		BCSCTL1, DCOCTL			SCFI0, SCFI1, SCFQCTL			
Timer_A, 受影响的控件		TACTL, TAR, CCTL0, CCR0						
为软件调用所做的准备		mov.b #00h, &BCSCTL2 mov #00h, SR br &0C00h			mov.b #00h, &FLLCTL1 br &0C00h			
注释	擦除段命令	连续验证地址 1000h 至 11FFh (三个段)。也使用擦除校验命令。						

**表 5-6. F21xx、F22xx、F23xx、F24xx、F261x 上的 BSL 版本 2.02 和 2.13**

器件		F21xx	F22xx	F23xx	F24x	F261x
BSL 版本		2.02				2.13
BSL 矢量地址	冷启动	0C00h				
	暖启动	0C02h <sup>(1)</sup>				
芯片 ID 地址		0FF0h				
芯片 ID 数据		F213h	F227h	F237h	F249h	F26Fh
BSL 版本地址		0FFAh				
BSL 版本数据		0202h				0213h
读取和写入访问地址	0000h 至 00FFh	字节				
	0100h 至 FFFEh	字				
写入时验证 ( 在线 )		地址 0200h 至 FFFEh				
擦除校验命令		是 ( 错误地址 0200h )				
擦除段命令		带有擦除验证 ( 错误地址 0200h )				
TX 识别命令		支持				
更改波特率命令		支持				
堆栈指针初始化	冷启动	0220h				0224h
	暖启动	未更改				
<b>BSL 所使用的资源</b>						
发送引脚 (TX)		P1.1				
接收引脚 (RX)		P2.2				
所使用的 RAM 堆栈		0200h 至 021Fh				0200h 至 0223h
工作寄存器		R5 至 R14				R4 至 R15
系统时钟, 受影响的控件		BCSCTL1, DCOCTL				SCF10, SCF11, SCFQCTL
Timer_A, 受影响的控件		TACTL, TAR, CCTL0, CCR0				
为软件调用所做的准备		mov.b #00h, &BCSCTL2 mov #00h, SR br &0C00h				
注释	擦除段命令	连续验证地址 1000h 至 11FFh ( 五个段 )。也使用擦除校验命令。				

(1) 在进入 BSL 之前, 用户应用必须清除 LOCK 和 LOCKA 位:  
mov.w #FWKEY+LOCKA, &FCTL3

**表 5-7. G2xx3、G2xx4、G2xx5、TCH5E 上的 BSL 版本 2.02 和 2.03<sup>(1)</sup>**

器件		G2xx4	G2xx5	G2xx3	TCH5E
BSL 版本		2.02		2.03	
BSL 矢量地址	冷启动	0C00h			
	暖启动	0C02h <sup>(2)</sup>			
芯片 ID 地址		0FF0h			
芯片 ID 数据		F227h	2955h	2553h	255Ch
BSL 版本地址		0FFAh			
BSL 版本数据		0202h		0203h	
读取和写入访问地址	0000h 至 00FFh	字节			
	0100h 至 FFEh	字			
写入时验证 ( 在线 )		地址 0200h 至 FFEh			
擦除校验命令		是 ( 错误地址 0200h )			
擦除段命令		带有擦除验证 ( 错误地址 0200h )			
TX 识别命令		支持			
更改波特率命令		支持			
堆栈指针初始化	冷启动	0220h			
	暖启动	未更改			
<b>BSL 所使用的资源</b>					
发送引脚 (TX)		P1.1		P1.1	
接收引脚 (RX)		P2.2		P1.5	
所使用的 RAM 堆栈		0200h 至 021Fh			
工作寄存器		R5 至 R14			
系统时钟, 受影响的控件		BCSCTL1, DCOCTL			
Timer_A, 受影响的控件		TACTL, TAR, CCTLO, CCR0			
为软件调用所做的准备		<pre>mov.b #00h, &amp;BCSCTL2 mov #00h, SR br &amp;0C00h</pre>			
注释	擦除段命令	连续验证地址 1000h 至 11FFh ( 五个段 )。也使用擦除校验命令。			

(1) 并非所有 Value Line 器件都包含 BSL；请参阅器件特定数据表。

(2) 在进入 BSL 之前，用户应用必须清除 LOCK 和 LOCKA 位：

```
mov.w #FWKEY+LOCKA, &FCTL3
```

**表 5-8. FG46xx、F471xx 上的 BSL 版本 2.12 和 2.13**

器件		FG46xx	F471xx
BSL 版本		2.12	2.13
BSL 矢量地址	冷启动	0C00h	
	暖启动	0C02h <sup>(1)</sup>	
芯片 ID 地址		0FF0h	
芯片 ID 数据		F46Fh	
BSL 版本地址		0FFAh	
BSL 版本数据		0212h	0213h
批量擦除时间, 额定值 (ms)		206.4	
读取和写入访问地址	0000h 至 00FFh	字节	
	0100h 至 FFFEh	字	
写入时验证 (在线)		地址 0200h 至 FFFEh	
擦除校验命令		是 (错误地址 0200h)	
擦除段命令		带有擦除验证的擦除段命令 (错误地址 0200h)	
TX 识别命令		支持	
更改波特率命令		支持	
堆栈指针初始化	冷启动	0224h	
	暖启动	未更改	
<b>BSL 所使用的资源</b>			
发送引脚 (TX)		P1.0	
接收引脚 (RX)		P1.1	
所使用的 RAM 堆栈		0200h 至 0223h	
工作寄存器		R4 至 R15	
系统时钟, 受影响的控件		SCF10, SCF11, SCFQCTL	
Timer_A, 受影响的控件		TACTL, TAR, CCTLO, CCR0	
为软件调用所做的准备		<pre>mov.b #00h, &amp;FLLCTL1 br    &amp;0C00h</pre>	
注释	擦除段命令	连续验证地址 1000h 至 11FFh (五个段)。也使用擦除校验命令。	

(1) 在进入 BSL 之前, 用户应用必须清除 LOCK 和 LOCKA 位:  
`mov.w #FWKEY+LOCKA, &FCTL3.`

## 5.2 ROM BSL 版本 1.10 的特殊考虑

ROM BSL 的第一个官方版本 V1.10 需要小的可加载补丁序列 PATCH.TXT 来可靠地执行 RX 块命令。如果将可加载 BSL 下载到此类器件中, 则必须执行相同的程序。BSL 启动后, 按以下方式执行:

1. RX 密码 (解锁下列命令的密码保护)
2. 用 0C22h 加载程序计数器 (PC) (将堆栈指针初始化至安全地址)
3. 再次 RX 密码 (解锁后续命令的密码保护)
4. RX 数据块 (可加载补丁代码, 代码段地址为 0220h)
5. TX 数据块 (用于验证的可加载补丁代码)

从现在开始, RX 块和 TX 块命令的使用有一个限制条件: 在调用它们之前, 程序计数器必须设置为补丁的起始地址。

1. 使用可加载补丁的起始地址 0220h 加载程序计数器 (PC)
2. RX 数据块 (在任一位置进行编程的代码), 或者
3. TX 数据块 (来自任一位置)

### 5.3 1xx、2xx 和 4xx BSL 已知问题

BSL 命令	擦除主存储器或者信息存储器
受影响版本	1.x
说明	当提供信息存储器中的地址时，请务擦除信息存储器
应变方法	使用“擦除段”命令

BSL 命令	擦除主存储器或者信息存储器
受影响版本	1.x
说明	当代码存储器的第一段作为地址提供时，报告失败。然而，擦除被正确执行。
应变方法	使用任一其他主存储器地址

BSL 命令	擦除段
受影响版本	1.x
说明	当在信息存储器中使用时，报告失败。然而，擦除被正确执行。
应变方法	无

### 5.4 有关 MSP430F14x 器件系列 BSL 的特殊说明

MSP430F14x 器件的修订版本 AA 具有更新至版本 1.61 的 BSL。这个变化的主要原因是为了提高 IP 安全性，确保正确的闪存编程和批量擦除，以及在编程过程中无需使用任何补丁。为了确保转换顺利，应对编程环境进行更新以顾及下述变化：

- 批量擦除命令只需发出一次，执行该命令的时间较长。
- 只允许对闪存进行字写入。
- 无需补丁或者 RAM 可加载 BSL。
- BSL 具有“在线验证”以加快编程。
- 内存分配已更改（请见节 5.3 中 BSL 版本图表）。
- 如果段擦除失败，BSL 可以返回 NAK。
- BSL 版本的传输和波特率的改变现在不受保护。



## 5.5 基于 F5xx 和 F6xx 闪存的 BSL 版本

**表 5-9.**

器件	MSP430F5438、MSP430F5437、MSP430F5435、MSP430F5436、MSP430F5435、MSP430F5419、MSP430F5418
BSL 版本	00.01.01.01
RAM 已擦除	无
内核命令的缓冲器大小	260 字节
显著信息	<ol style="list-style-type: none"> <li>在器件数据表中注明了 UART TX 和 RX BSL 引脚</li> <li>由于已知的勘误表 SYS4 和 FLASH32，BSL 不可重复编程。</li> <li>然而，JTAGLOCK KEY (位于 BSL 区域) 仍然可以通过 BSL 本身进行编程。请注意，这需要能够一次锁定和解锁 JTAG。锁定 JTAG 是通过向 JTAGLOCK KEY 写入 0x5555 来执行的，而解锁 JTAG 是通过写入 0x0000 来执行的。一旦完成后，就没有机会背锁 JTAG，因为它需要整个 BSL 编程。</li> <li>唯一支持的波特率为 9600 和 57600。</li> <li>BSL 在 TA0.0 上传输，在 TA0.1 上接收。</li> <li>BSL 不期望校验位。</li> </ol>
已知漏洞	<ol style="list-style-type: none"> <li>BSL 密码是地址 0xFFFF0 和 0xFFFF 之间的字节。这意味着该 BSL 版本只期待 RX 密码命令中一个密码的 16 个字节。发送 32 个字节会返回错误。</li> <li>如果地址 0x20396 或者 0x20397 包含在 CRC 命令的地址范围内，则返回的数据错误。</li> <li>批量擦除命令也会擦除 Info_A。</li> <li>如果密码错误，器件将擦除所有 RAM，包括其堆栈。因此，不能保证正确返回错误代码。</li> <li>CRC 函数的字节总数被 0x7FFF 屏蔽，因此限制为 32767。</li> </ol>

**表 5-10.**

器件	MSP430F5438A、MSP430F5437A、MSP430F5435A、MSP430F5436A、MSP430F5435A、MSP430F5419A、MSP430F5418A
BSL 版本	00.05.04.03 (修订版 A 至修订版 E) 00.07.05.04 (修订版 F 和后续版本)
RAM 已擦除	0x1C00 至 0x5BFF
用于内核命令的缓冲器大小	260 字节
显著信息	<ol style="list-style-type: none"> <li>在器件数据表中注明了 UART TX 和 RX BSL 引脚</li> </ol>
已知漏洞	<ol style="list-style-type: none"> <li>无法保所有时钟、电压和温度变化都达到 115k 的波特率。</li> </ol>

**表 5-11.**

器件	CC430F6147、CC430F6145、CC430F6143、CC430F6137、CC430F6135、CC430F6127、CC430F6126、CC430F6125、CC430F5147、CC430F5145、CC430F5143、CC430F5137、CC430F5135、CC430F5133、CC430F5125、CC430F5123
BSL 版本	00.05.04.52 (修订版 A 至修订版 C) 00.07.05.53 (修订版 D 和后续版本)
RAM 已擦除	0x1C00 至 0x23FF
用于内核命令的缓冲器大小	260 字节
显著信息	<ol style="list-style-type: none"> <li>UART TX 和 RX BSL 引脚在引脚 P1.6 (TXD) 和 P1.5 (RXD) 上实现</li> </ol>
已知漏洞	<ol style="list-style-type: none"> <li>无法保所有时钟、电压和温度变化都达到 115k 的波特率。</li> </ol>

**表 5-12.**

器件	MSP430F5510、MSP430F5500、MSP430F5501、MSP430F5502、MSP430F5503、MSP430F5504、MSP430F5505、MSP430F5506、MSP430F5507、MSP430F5508、MSP430F5509
BSL 版本	00.03.83.33 ( 修订版 A 至修订版 E ) 00.07.88.38 ( 修订版 F 直到 2015 年 5 月 ) 00.08.88.39 ( 修订版 F 及后续版本 )
RAM 已擦除	0x2400 至 0x33FF
用于内核命令的缓冲器大小	62 字节
显著信息	<ol style="list-style-type: none"> <li>1. 器件使用工厂 USB BSL 进行编程。</li> <li>2. 工厂 USB BSL 仅支持 RAM 写入。完整的 BSL 必须首先加载到器件 RAM 中并开始执行闪存写入。仅支持 RX PASSWORD、RX DATA BLOCK FAST 和 SET PC 命令。</li> <li>3. 当从应用程序启动此 BSL 时，应用程序应首先取消枚举自身，然后在启动 BSL 之前延迟 ( 大约 500ms )。这允许对主机进行正确的重新枚举。</li> <li>4. XT2 需要外部晶振以确保 USB 运行。</li> </ol>
已知漏洞	USB 模块未被 BSL 正确锁定。由于遗留原因，保留此行为。

**表 5-13.**

器件	MSP430F5529、MSP430F5513、MSP430F5514、MSP430F5515、MSP430F5517、MSP430F5519、MSP430F5521、MSP430F5522、MSP430F5524、MSP430F5525、MSP430F5526、MSP430F5527、MSP430F5528
BSL 版本	00.03.83.33 ( 修订版 A 至修订版 H ) 00.07.85.36 ( 修订版 I ) 00.07.87.37 ( 修订版 J ) 00.07.88.38 ( 修订版 K 直到 2015 年 5 月 ) 00.08.88.39 ( 修订版 K 及后续版本 )
RAM 已擦除	0x2400 至 0x33FF
用于内核命令的缓冲器大小	62 字节
显著信息	<ol style="list-style-type: none"> <li>1. 器件使用工厂 USB BSL 进行编程。</li> <li>2. 工厂 USB BSL 仅支持 RAM 写入。完整的 BSL 必须首先加载到器件 RAM 中并开始执行闪存写入。仅支持 RX PASSWORD、RX DATA BLOCK FAST 和 SET PC 命令。</li> <li>3. 当从应用程序启动此 BSL 时，应用程序应首先取消枚举自身，然后在启动 BSL 之前延迟 ( 大约 500ms )。这允许对主机进行正确的重新枚举。</li> <li>4. XT2 需要外部晶振以确保 USB 运行。</li> </ol>
已知漏洞	USB 模块未被 BSL 正确锁定。由于遗留原因，保留此行为。

**表 5-14.**

器件	MSP430F5172、MSP430F5152、MSP430F5132、MSP430F5171、MSP430F5151、MSP430F5131
BSL 版本	00.07.05.04
RAM 已擦除	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260 字节
显著信息	<ol style="list-style-type: none"> <li>1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚</li> </ol>
已知漏洞	<ol style="list-style-type: none"> <li>1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。</li> </ol>

**表 5-15.**

器件	MSP430F5229、MSP430F5227、MSP430F5219、MSP430F5217、MSP430F5224、MSP430F5222、MSP430F5213、MSP430F5212
BSL 版本	00.07.05.04
RAM 已擦除	0x2400 至 0x43FF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-16.**

器件	MSP430F5249、MSP430F5247、MSP430F5244、MSP430F5242、MSP430F5239、MSP430F5237、MSP430F5234、MSP430F5232
BSL 版本	00.08.08.04
RAM 已擦除	0x2400 至 0x43FF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-17.**

器件	MSP430F5255、MSP430F5254、MSP430F5253、MSP430F5252
BSL 版本	00.08.08.04
RAM 已擦除	0x2400 至 0x43FF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚 2. 在自定义 BSL 包中可以下载使用 DVIO 电源域中的引脚的 BSL 固件映像。
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-18.**

器件	MSP430F5259、MSP430F5258、MSP430F5257、MSP430F5256
BSL 版本	00.07.06.94
RAM 已擦除	0x1C00 至 0x23FF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 I <sup>2</sup> C 引脚
已知漏洞	1. 长度大于 260 的 I <sup>2</sup> C 读取命令不会返回正确的数据。

**表 5-19.**

器件	MSP430F5310、MSP430F5309、MSP430F5308、MSP430F5304、MSP430F5340、MSP430F5341、MSP430F5342、MSP430F5329、MSP430F5324、MSP430F5325、MSP430F5326、MSP430F5327、MSP430F5328
BSL 版本	00.06.04.04
RAM 已擦除	0x1C00 至 0x33FF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-20.**

器件	MSP430F6638、MSP430F6637、MSP430F6636、MSP430F6635、MSP430F6634、MSP430F6633、MSP430F6632、MSP430F6631、MSP430F6630、MSP430F5638、MSP430F5637、MSP430F5636、MSP430F5635、MSP430F5634、MSP430F5633、MSP430F5632、MSP430F5631、MSP430F5630
BSL 版本	00.04.84.34 (修订版 A 至修订版 D) 00.08.88.38 (修订版 E 直到 2015 年 5 月) 00.08.88.39 (修订版 E 及后续版本)
RAM 已擦除	0x2400 至 0x33FF
用于内核命令的缓冲器大小	62 字节
显著信息	1. 器件使用工厂 USB BSL 进行编程。 2. 工厂 USB BSL 仅支持 RAM 写入。完整的 BSL 必须首先加载到器件 RAM 中并开始执行闪存写入。仅支持 RX PASSWORD、RX DATA BLOCK FAST 和 SET PC 命令 3. 当从应用程序启动此 BSL 时，应用程序应首先取消枚举自身，然后在启动 BSL 之前延迟 (大约 500ms)。这允许对主机进行正确的重新枚举。
已知漏洞	USB 模块未被 BSL 正确锁定。由于遗留原因，保留此行为。

**表 5-21.**

器件	MSP430F6659、MSP430F6658、MSP430F5659、MSP430F5658
BSL 版本	00.07.86.36 (修订版 A) 00.08.88.38 (修订版 B 直到 2015 年 5 月) 00.08.88.39 (修订版 B 及后续版本)
RAM 已擦除	0x2400 至 0x33FF
用于内核命令的缓冲器大小	62 字节
显著信息	1. 器件使用工厂 USB BSL 进行编程。 2. 工厂 USB BSL 仅支持 RAM 写入。完整的 BSL 必须首先加载到器件 RAM 中并开始执行闪存写入。仅支持 RX PASSWORD、RX DATA BLOCK FAST 和 SET PC 命令 3. 当从应用程序启动此 BSL 时，应用程序应首先取消枚举自身，然后在启动 BSL 之前延迟 (大约 500ms)。这允许对主机进行正确的重新枚举。
已知漏洞	

**表 5-22.**

器件	MSP430F6438、MSP430F6436、MSP430F6435、MSP430F6433、MSP430F5338、MSP430F5336、MSP430F5335、MSP430F5333、MSP430F6459、MSP430F6458、MSP430F5359、MSP430F5358
BSL 版本	00.07.05.04
RAM 已擦除	0x1C00 至 0x43FF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚

**表 5-22. (continued)**

已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。
------	----------------------------------

**表 5-23.**

器件	MSP430F6736、MSP430F6720、MSP430F6721、MSP430F6723、MSP430F6724、MSP430F6725、MSP430F6726、MSP430F6730、MSP430F6731、MSP430F6733、MSP430F6734、MSP430F6735、MSP430F6736A、MSP430F6735A、MSP430F6734A、MSP430F6733A、MSP430F6731A、MSP430F6730A、MSP430F6726A、MSP430F6725A、MSP430F6724A、MSP430F6723A、MSP430F6721A、MSP430F6720A
BSL 版本	00.07.05.04
RAM 已擦除	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-24.**

器件	MSP430F6779、MSP430F6745、MSP430F6746、MSP430F6747、MSP430F6748、MSP430F6749、MSP430F6765、MSP430F6776、MSP430F6767、MSP430F6768、MSP430F6769、MSP430F6775、MSP430F6776、MSP430F6777、MSP430F6778、MSP430F67791、MSP430F67451、MSP430F67461、MSP430F67471、MSP430F67481、MSP430F67491、MSP430F67651、MSP430F67761、MSP430F67671、MSP430F67681、MSP430F67691、MSP430F67751、MSP430F67761、MSP430F67771、MSP430F67781
BSL 版本	00.07.05.04
RAM 已擦除	0x1C00 至 0x5BFF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-25.**

器件	MSP430F6779A、MSP430F6778A、MSP430F6777A、MSP430F6776A、MSP430F6775A、MSP430F6769A、MSP430F6768A、MSP430F6767A、MSP430F6766A、MSP430F6765A、MSP430F6749A、MSP430F6748A、MSP430F6747A、MSP430F6746A、MSP430F6745A、MSP430F67791A、MSP430F67781A、MSP430F67771A、MSP430F67761A、MSP430F67751A、MSP430F67691A、MSP430F67681A、MSP430F67671A、MSP430F67661A、MSP430F67651A、MSP430F67491A、MSP430F67481A、MSP430F67471A、MSP430F67461A、MSP430F67451A
BSL 版本	00.07.05.04
RAM 已擦除	0x1C00 至 0x5BFF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-26.**

器件	MSP430F67641、MSP430F67621
BSL 版本	00.07.05.04
RAM 已擦除	0x1C00 到 0x1FFF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-27.**

器件	MSP430FG6426、MSP430FG6425
BSL 版本	00.08.08.04
RAM 已擦除	0x1C00 至 0x43FF
用于内核命令的缓冲器大小	260 字节
显著信息	1. 在器件数据表中注明了 UART TX 和 RX BSL 引脚
已知漏洞	1. 无法保所有时钟、电压和温度变化都达到 115k 的波特率。

**表 5-28.**

器件	MSP430FG6626、MSP430FG6625
BSL 版本	00.08.88.38
RAM 已擦除	0x1C00 至 0x43FF
用于内核命令的缓冲器大小	260 字节
显著信息	<ol style="list-style-type: none"> <li>1. 器件使用工厂 USB BSL 进行编程。</li> <li>2. 工厂 USB BSL 仅支持 RAM 写入。完整的 BSL 必须首先加载到器件 RAM 中并开始执行闪存写入。仅支持 RX PASSWORD、RX DATA BLOCK FAST 和 SET PC 命令</li> <li>3. 当从应用程序启动此 BSL 时，应用程序应首先取消枚举自身，然后在启动 BSL 之前延迟 ( 大约 500ms ) 。这允许对主机进行正确的重新枚举。</li> <li>4. XT2 需要外部晶振以确保 USB 运行。</li> </ol>
已知漏洞	

## 6 引导加载程序 PCB 布局建议

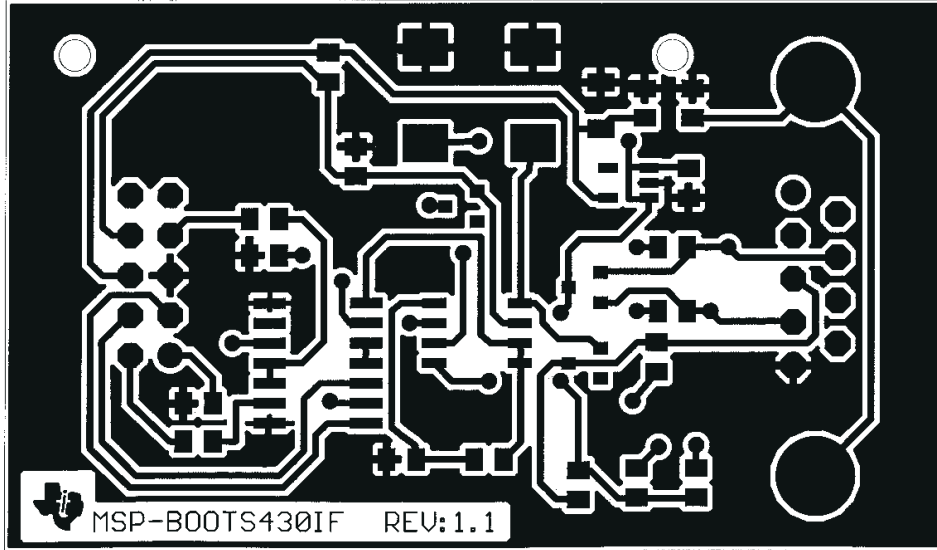


图 6-1. 通用 BSL 接口 PCB 布局，顶视图

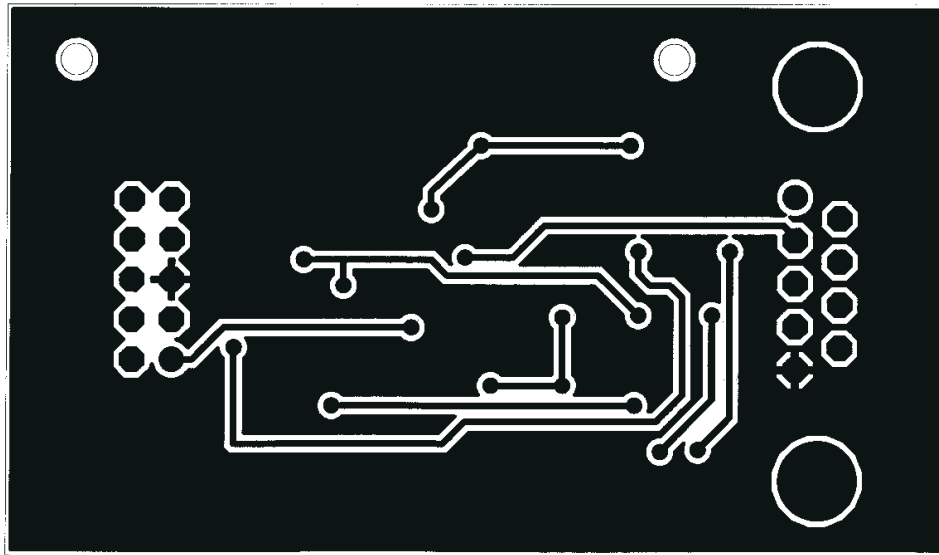


图 6-2. 通用 BSL 接口 PCB 布局，底视图



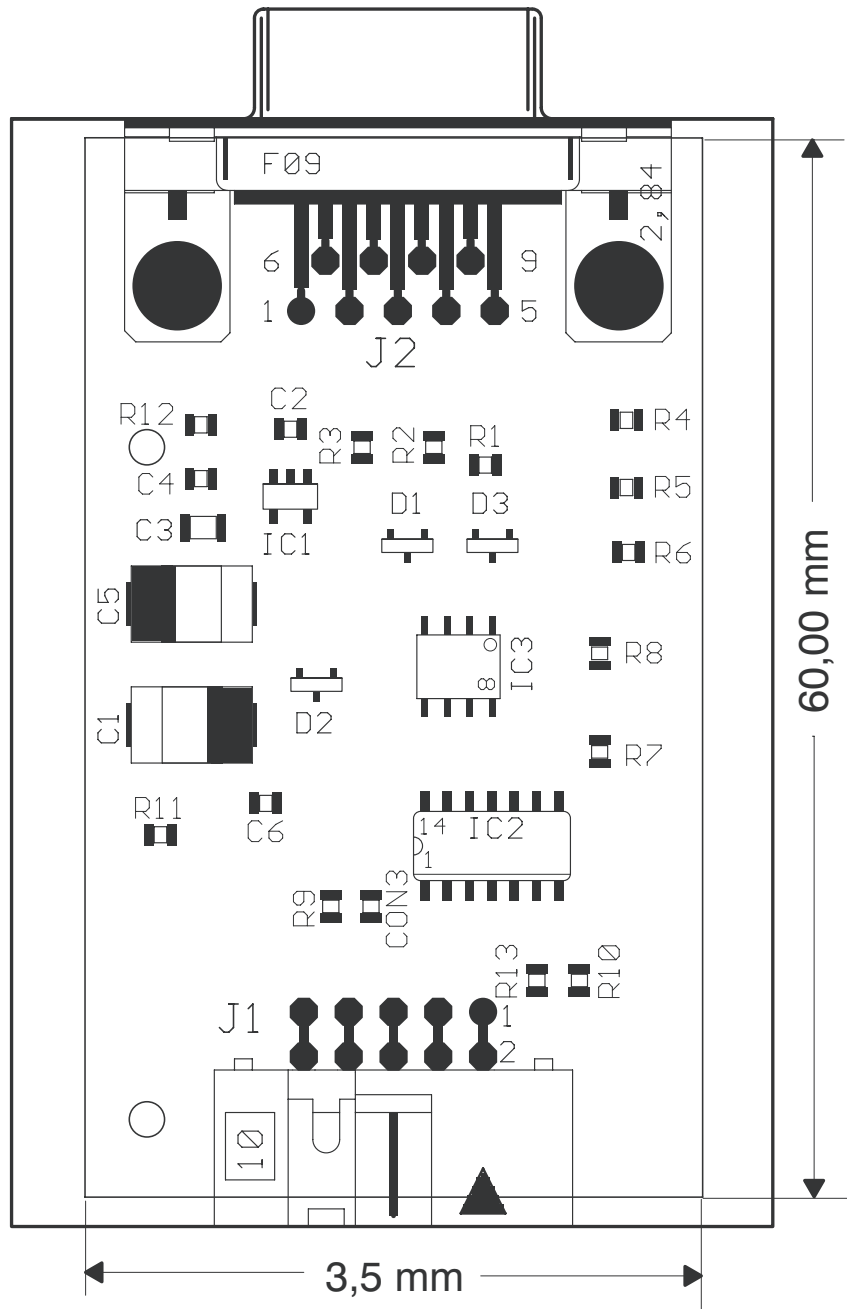


图 6-3. 通用 BSL 接口组件位置

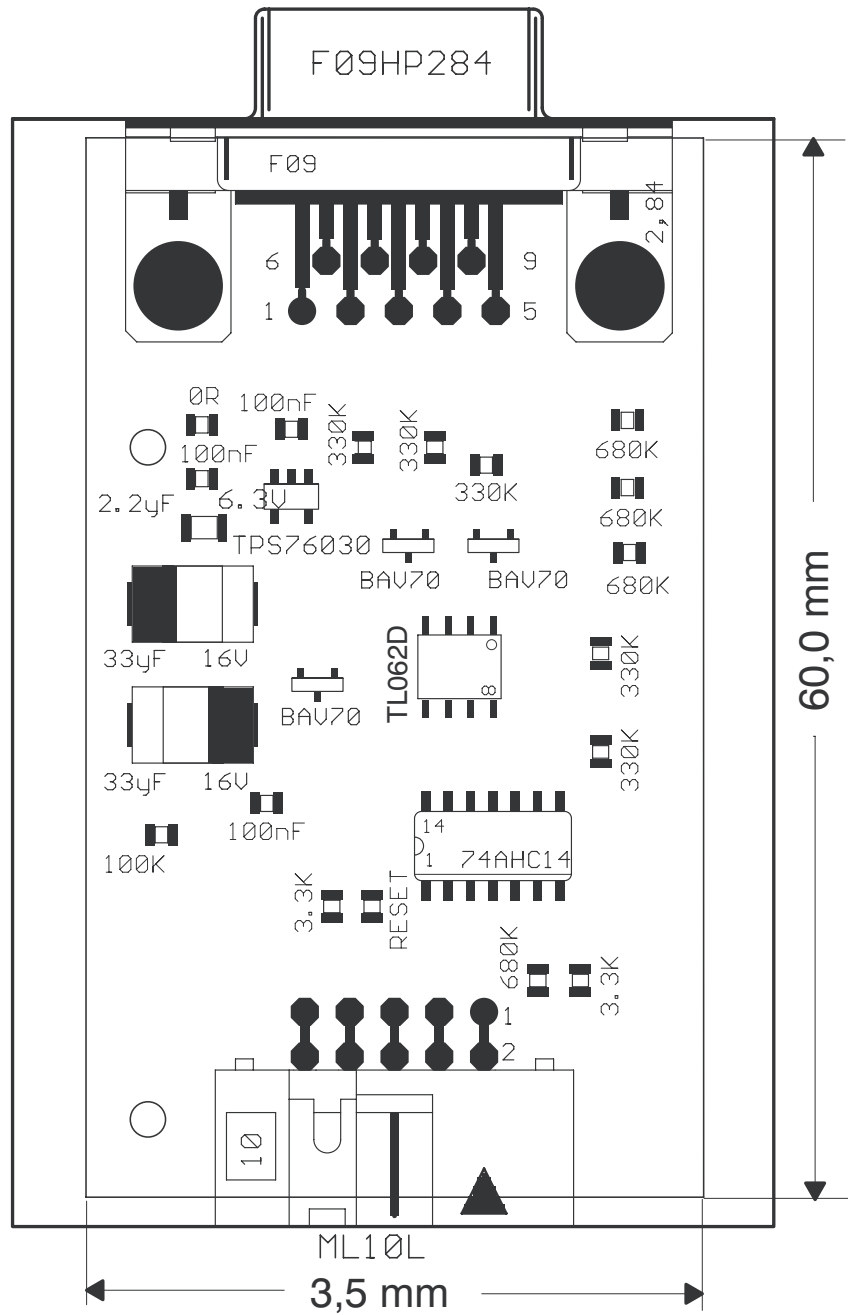


图 6-4. 通用 BSL 接口组件位置

## 7 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 APRIL 7, 2021 to SEPTEMBER 22, 2022</b>	<b>Page</b>
• 从表 1-1 BSL 概览中删除了 MSP432P4xx 列 ( 这些器件已停产 ) .....	4

## 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