



Elin Wollert

摘要

本应用报告简要概述了驻留在 CC2538、CC13x0/CC26x0 和 CC13x2/CC26x2 器件 ROM 中的串行引导加载程序。本文档介绍了如何使用引导加载程序协议来执行基本操作，例如擦除器件的闪存和对其编程。设备引导加载程序支持通用异步接收器/发送器 (UART) 和串行外设接口 (SPI) 作为协议传输层。

本应用报告涵盖了 UART，旨在与相关示例文件一同使用，相关示例文件可从以下 URL 下载：<http://www.ti.com/cn/lit/zip/swra466>。该示例是在 Visual Studio® Professional 2015 中创建的，并利用 Serial Bootloader Library 库来演示如何在 Windows® 上实现串行引导加载程序协议。

内容

1 引言.....	3
2 ROM 引导加载程序.....	3
2.1 配置引导加载程序.....	4
2.2 CC26x2.....	5
2.3 通信协议.....	5
2.4 接口配置.....	6
3 Serial Bootloader Library (SBL).....	9
3.1 SBL 返回值.....	9
3.2 SBL API.....	10
4 示例项目.....	12
4.1 硬件设置.....	13
4.2 软件设置.....	14
4.3 程序流程.....	14
5 参考文献.....	19
6 修订历史记录.....	19

插图清单

图 2-1. 进入引导加载程序的简化流程图 (CC26x0 和 CC26x2)	3
图 2-2. 发送和接收协议的序列图表.....	6
图 2-3. 连接初始化的序列图表.....	8
图 3-1. Ping 函数调用的序列图表.....	9
图 4-1. CC2538 示例应用的成功执行.....	12
图 4-2. PC 到 UART 连接.....	13
图 4-3. XDS100v3 仿真器旁路接头上的 EM TX 和 RX 引脚.....	13
图 4-4. 带有未初始化引导加载程序的 initCommunication 函数的序列图表.....	15
图 4-5. 闪存扇区擦除的序列图表.....	16
图 4-6. 闪存写入的序列图表.....	17
图 4-7. CRC32 命令的序列图表.....	18
图 4-8. SBL 函数复位的序列图表.....	18

表格清单

表 2-1. 8 位引导加载程序配置字段的地址 (CC2538 变体)	4
表 2-2. CC2538 引导加载程序后门编码.....	4
表 2-3. CC26x0 CCFG:BL_CONFIG 编码.....	5
表 2-4. CC26x2 CCFG:BL_CONFIG 编码.....	5

表 2-5. ROM 引脚加载程序数据包格式.....	6
表 2-6. 数据包格式字段说明.....	6
表 2-7. 肯定确认/否定确认响应.....	6
表 2-8. 串行接口配置：评估模块套件.....	6
表 2-9. 串行接口配置：LaunchPad.....	7
表 2-10. 从引导加载程序返回的可能状态值.....	8
表 3-1. SBL 函数返回值.....	9
表 3-2. SBL 函数.....	10
表 3-3. 器件特定 SBL 函数.....	11
表 4-1. 应用示例 IO 配置：评估模块套件.....	14
表 4-2. 应用示例 IO 配置：LaunchPad.....	14
表 4-3. 配置：deviceType.....	14

商标

Visual Studio® and Windows® are registered trademarks of Microsoft Corporation in the United States and/or other countries, or both.

所有商标均为其各自所有者的财产。

1 引言

CC2538、CC13x0/CC26x0 和 CC13x2/CC26x2 ROM 引导加载程序的主要用途是支持通过 SPI 或 UART 将闪存映像编程到器件闪存中的功能。为了简短起见，本文档在下文中将 CC13x0/CC26x2 器件系列 (包括 CC2640R2) 称为 CC26x0，将 CC13x2/CC26x2 器件系列称为 CC26x2。使用时，CC26xx 指的是 CC26x0 和 CC26x2 器件系列。

本文档介绍如何使用引导加载程序执行基本操作，如擦除和编程闪存。本文档使用 UART 作为引导加载程序传输层。

2 ROM 引导加载程序

如果闪存中没有有效的应用程序映像 (由客户配置区域 (CCA/CCFG) 中的 “image valid” 字段确定)，CC2538、CC26x0 和 CC26x2 器件上的内置引导加载程序将在上电复位后开始运行。更多有关 CCA/CCFG 中 “image valid” 字段的信息，请参阅《CC2538 ROM 用户指南》[1]、《CC13x0、CC26x0 SimpleLink™ 无线 MCU 技术参考指南》[2] 和《CC13x2、CC26x2 SimpleLink™ 无线 MCU 技术参考手册》[3]。

或者，如果启用了所谓的引导加载程序后门，并且将打开后门的相关引脚设置为正确的逻辑电平，则引导加载程序会启动。激活引导加载程序后，会在上电复位 10ms 后与外部主机通信。

CC2538、CC26x0 和 CC26x2 ROM 引导加载程序支持可读取闪存的命令，因此出于安全原因，也可以完全禁用引导加载程序。CCA/CCFG 中配置了引导加载程序和后门功能。

图 2-1 所示为 CC26x0 和 CC26x2 启动代码的简化流程图。该流程类似于 CC2538 器件。

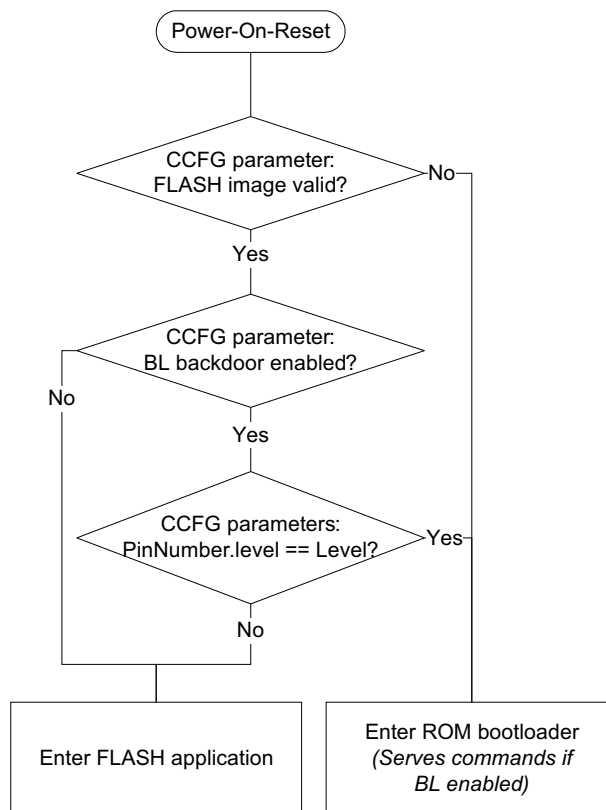


图 2-1. 进入引导加载程序的简化流程图 (CC26x0 和 CC26x2)

2.1 配置引导加载程序

2.1.1 CC2538

CC2538 的客户配置区被称为 CCA，位于最上面的闪存扇区，因此 CCA 的绝对地址取决于器件闪存大小。CCA 中的 8 位字段用于配置引导加载程序和后门功能（字节偏移 0x7D7）。表 2-1 为不同 CC2538 变体列出了此字节的绝对地址。

表 2-1. 8 位引导加载程序配置字段的地址 (CC2538 变体)

CC2538 变体	引导加载程序配置地址
Cx2538xF53 (512KB 闪存)	0x0027.FFD7
Cx2538xF23 (256KB 闪存)	0x0023.FFD7
Cx2538xF11 (128KB 闪存)	0x0021.FFD7

引导加载程序配置字节的结构如表 2-2 所示。可以打开引导加载程序后门的引脚是 PA0 - PA7。通过在后门配置字节的三个最低有效位中写入一个从 0 到 7 的值来选择要使用的引脚。

表 2-2. CC2538 引导加载程序后门编码

位	字段	值	说明	默认值
7-5	Reserved	0	保留。全都应为一。	111b
4	Enabled	0	启用和禁用后门功能	1
		0	后门和引导加载程序禁用	
		1	后门和引导加载程序启用	
3	Level	0	在焊盘 A 上设置所选引脚的运行电平	1
		0	低电平有效	
		1	高电平有效	
2-0	Pin number		后门启用时，焊盘 A 上使用的引脚编号 (0 - 7)。	111b (7)

2.1.2 CC26x0

CC26x0 的客户配置区被称为 CCFG，并位于最上面的闪存扇区，因此 CCFG 的绝对地址取决于器件闪存大小。128KB 闪存变体的引导加载程序配置绝对地址为 0x0001.FFD8，64KB 闪存变体为 0x0000.FFD8，32KB 闪存变体为 0x0000.7FD8。CC26x0 CCFG 也经过存储器映射，具有读取所有闪存变体的地址 0x5000.3000 的权限。CCFG 中的 32 位字段用于配置引导加载程序和后门功能（字节偏移 0xFD8）。

引导加载程序配置字段的结构如表 2-3 所示。配置结构是小端字节序，这意味着最低有效字节位于最低地址。通过将 DIO 编号写入配置结构的第二个字节来选择要使用的引脚。

表 2-3. CC26x0 CCFG:BL_CONFIG 编码

位	字段	值	说明	字节偏移	默认值
31:24	BOOTLOADER_ENABLE	0xC5 任何其他值	启用和禁用引导加载程序 引导加载程序已启用 引导加载程序已禁用	0xFDB	0xC5
23:17	RESERVED	0		0xFDA	0b111 1111
16	BL_LEVEL	0 1	设置所选引脚的运行电平。 低电平有效 高电平有效	0xFDA	1
15:8	BL_PIN_NUMBER		如果引导加载程序后门启用，则对 I/O 引脚编号进行电平检查。	0xFD9	0xFF
7:0	BL_ENABLE	0xC5 任何其他值	启用和禁用引导加载程序后门。 引导加载程序已启用 引导加载程序已禁用	0xFD8	0xFF

2.2 CC26x2

CC26x2 的客户配置区被称为 CCFG，位于最上面的闪存扇区。CC26x2 器件具有 1 个闪存大小，352KB，引导加载程序配置绝对地址为 0x0005.7FD8。CC26x2 CCFG 也经过存储器映射，具有读取地址 0x5000.3000 的权限。CCFG 中的 32 位字段用于配置引导加载程序和后门功能（字节偏移 0x1FD8）。

引导加载程序配置字段的结构如表 2-4 所示。配置结构是小端字节序，这意味着最低有效字节位于最低地址。通过将 DIO 编号写入配置结构的第二个字节来选择要使用的引脚。

表 2-4. CC26x2 CCFG:BL_CONFIG 编码

位	字段	值	说明	字节偏移	默认值
31:24	BOOTLOADER_ENABLE	0xC5 任何其他值	启用和禁用引导加载程序 引导加载程序已启用 引导加载程序已禁用	0x1FDB	0xC5
23:17	RESERVED	0		0x1FDA	0b111 1111
16	BL_LEVEL	0 1	设置所选引脚的运行电平。 低电平有效 高电平有效	0x1FDA	1
15:8	BL_PIN_NUMBER		如果引导加载程序后门已启用，则对 I/O 引脚编号进行电平检查。	0x1FD9	0xFF
7:0	BL_ENABLE	0xC5 任何其他值	启用和禁用引导加载程序后门。 引导加载程序已启用 引导加载程序已禁用	0x1FD8	0xFF

2.3 通信协议

CC2538、CC26x0 和 CC26x2 引导加载程序使用相同的格式来接收和发送数据包。SPI 和 UART 传输层上的实际信令不同，但数据包格式保持不变。表 2-5 中显示了数据包格式，表 2-6 中对各字段进行了描述。

表 2-5. ROM 引脚加载程序数据包格式

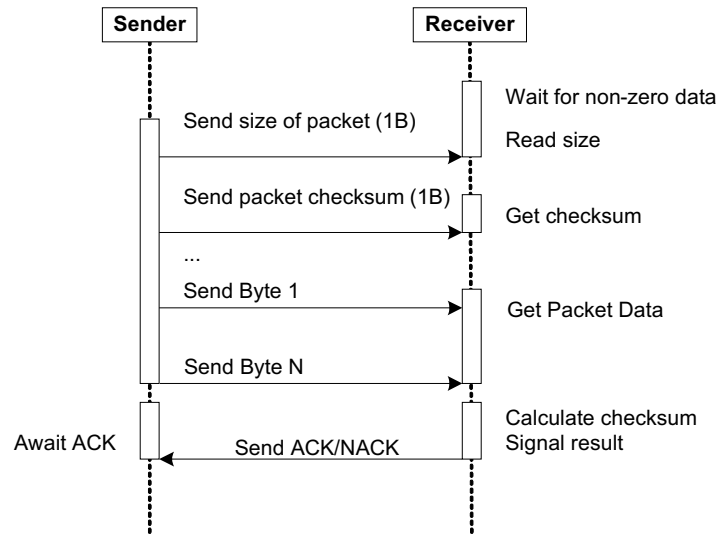
大小 (1 字节)	校验和 (1 字节)	数据字节 1	...	数据字节 N
-------------	--------------	--------	-----	--------

表 2-6. 数据包格式字段说明

数据包字段	大小 (字节)	说明
Size (大小)	1	数据包中的字节数, 包括大小字节。
Checksum (校验和)	1	数据的校验和。校验和算法是被截断为 8 位的数据字节的总和。 $校验和 = (\sum data) \bmod 256$
Data (数据)	0-253	实际数据字节。第一个数据字节通常是引导加载程序的命令字节。

数据包发送和数据包接收必须遵守如图 2-2 所示的简单协议。主机器件和 CC2538/CC26x0/CC26x2 引导加载程序都可以充当发送器和接收器。主机器件在等待来自引导加载程序的数据响应时成为接收器。

有关通信协议的更多详细信息, 请参阅 [1]、[2] 和 [3]。


图 2-2. 发送和接收协议的序列图表

2.3.1 ACK/NACK

接收器应以肯定确认 (ACK) 或否定确认 (NACK) 进行响应, 用于指示是否正确接收到命令。表 2-7 中显示了 ACK 和 NACK 签名。

表 2-7. 肯定确认/否定确认响应

协议字节	值
ACK	0xCC
NACK	0x33

2.4 接口配置

2.4.1 硬件引脚

表 2-8 和表 2-9 中显示了 ROM 引导加载程序用以通过 UART 和 SPI 进行通信的硬件引脚。

表 2-8. 串行接口配置: 评估模块套件

信号	CC2538	CC26x0			EM 引脚
		QFN48/7x7	QFN32/5x5	QFN32/4x4	
UART_RX	PA0	DIO2	DIO1	DIO1	1.07
UART_TX	PA1	DIO3	DIO0	DIO2	1.09
SPI_CLK	PA2	DIO10	DIO10	DIO8	1.16

表 2-8. 串行接口配置：评估模块套件 (continued)

信号	CC2538	CC26x0			EM 引脚
		QFN48/7x7	QFN32/5x5	QFN32/4x4	
SPI CSn	PA3	DIO11	DIO9	DIO7	1.14
SPI MOSI	PA4	DIO9	DIO11	DIO9	1.18
SPI MISO	PA5	DIO8	DIO12	DIO0	1.20

表 2-9. 串行接口配置：LaunchPad

信号	CC2640R2 (1)	CC26x2R	CC1312R	CC1352x	LaunchPad 引脚
UART_RX	DIO2	DIO2	DIO2	DIO12	3(2)
UART_TX	DIO3	DIO3	DIO3	DIO13	4(2)
SPI CLK	DIO10	DIO10	DIO10	DIO10	7
SPI CSn	DIO11	DIO11	DIO11	DIO11	18
SPI MOSI	DIO9	DIO9	DIO9	DIO9	15
SPI MISO	DIO8	DIO8	DIO8	DIO8	14

(1) 该引脚排列仅对 QFN48/7x7 封装有效。

(2) CC2640R2 LaunchPad 的相反顺序。

引导加载程序会选择外部器件访问的第一个接口。将禁用非运行接口 (UART 或 SPI)。若要切换到另一个接口，必须使用诸如复位引脚等来复位器件。

2.4.2 UART 配置

UART 数据格式固定为 8 个数据位，无奇偶校验，有 1 个停止位。UART 引导加载程序利用波特率的自动检测功能 (参见节 2.4.3)；因此，可使用任何低于最大值的波特率。

- CC2538 的最大 UART 波特率：460800 波特¹
- CC26x0 和 CC26x2 的最大 UART 波特率：1.5M 波特

¹ 如果外部 32MHz 晶体振荡器在使用中并使用 COMMAND_SET_XOSC 引导加载程序命令将其选中，则该数据速率数字可加倍。调用此命令后，必须重建 UART 通信 (参见节 2.4.3)。

2.4.3 建立通信

图 2-3 中显示了通过 UART 与引导加载程序建立通信所需的最低条件，其中包括发送值为 0x55 的两个字节以让器件检测波特率，然后读取器件响应，如果自动波特率例程成功，则预期会收到 ACK。如果器件没有响应自动波特率字节，它可能未处于引导加载程序模式，或者不支持波特率。

建立连接后，可以向引导加载程序发送任何命令。对于 CC2538、CC26x0 和 CC26x2，引导加载程序命令的完整列表可分别在 [1]、[2] 和 [3] 中找到。

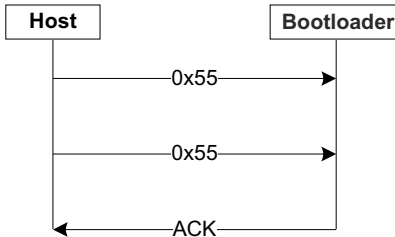


图 2-3. 连接初始化的序列图表

2.4.4 状态命令

若要检查引导加载程序的状态，可以使用 CMD_GET_STATUS 命令；此命令应在擦除或写入闪存存储器后使用，以确保在下一步之前成功完成擦除和写入。表 2-10 中显示了 CMD_GET_STATUS 命令的可能状态代码。

表 2-10. 从引导加载程序返回的可能状态值

状态定义	值	说明
COMMAND_RET_SUCCESS	0x40	成功执行的命令的状态
COMMAND_RET_UNKNOWN_CMD	0x41	未知命令的状态
COMMAND_RET_INVALID_CMD	0x42	无效命令的状态 (数据包大小不正确)
COMMAND_RET_INVALID_ADR	0x43	无效输入地址的状态
COMMAND_RET_FLASH_FAIL	0x44	编程或擦除闪存尝试失败的状态

3 Serial Bootloader Library (SBL)

SBL 是适用于 Microsoft Windows 的 PC 库，使用与 CC2538、CC26x0 和 CC26x2 串行引导加载程序进行通信的主机 API。SBL 库工程是在 Visual Studio C++ Professional 2015 中创建的。Serial Bootloader Library 使用 Windows API 与串行 COM 端口进行通信，因此不具有跨平台兼容性。

SBL 中的所有函数都是同步的；这意味着在收到 ACK 或 NACK 或发生错误之前，该函数不会返回。图 3-1 演示了 SBL ping() 函数的序列图表。

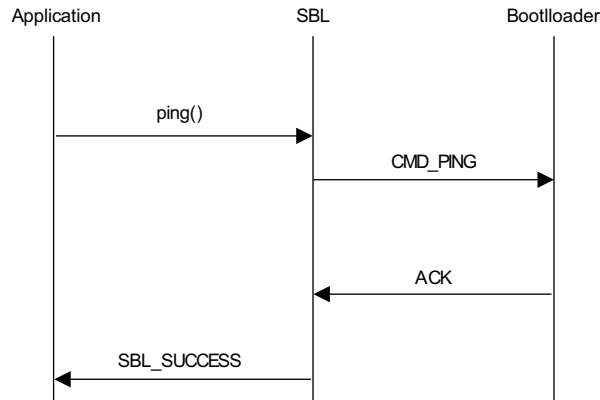


图 3-1. Ping 函数调用的序列图表

所有引导加载程序命令都可通过 SBL 中的函数访问，从而简化了直接通过 SBL 执行闪存存储器的擦除和写入等操作。

有关 ROM 引导加载程序以及如何使用所有串行命令的更详细说明，请参阅器件特定 ROM 用户指南 [1]、[2] 和 [3]。

3.1 SBL 返回值

每个 SBL 函数将通过解释引导加载程序响应来返回所需操作是否成功。表 3-1 中显示了 SBL 函数的可能返回值及其可能原因的列表。

表 3-1. SBL 函数返回值

常数名称	值	原因
SBL_SUCCESS	0	引导加载程序成功执行了命令
SBL_ERROR	1	执行命令期间出错
SBL_ARGUMENT_ERROR	2	SBL 函数参数无效
SBL_TIMEOUT_ERROR	3	在给定的尝试次数内未收到引导加载程序响应。
SBL_PORT_ERROR	4	无法向引导加载程序发送数据或从引导加载程序接收数据
SBL_ENUM_ERROR	5	未能枚举 COM 器件
SBL_UNSUPPORTED_FUNCTION	6	所选硬件不支持函数

3.2 SBL API

SBL API 概览如表 3-2 所示。直接映射到引导加载程序命令的 API 函数用 X 进行标记。

表 3-2. SBL 函数

SBL 函数名称	引导加载程序 CMD			说明
	CC2538	CC26x0	CC26x2	
创建	不适用	不适用	不适用	用于创建 SBL 器件对象的静态函数。
calculateCrc32	X	X	X	计算额定条件范围下的 CRC32。
连接				使用 ROM 引导加载程序初始化连接
枚举	不适用	不适用	不适用	用于在 PC 上枚举 COM 端口的静态函数。
eraseFlashBank		X	X	擦除整个闪存。CC2538 不支持。
eraseFlashRange	X	X	X	擦除指定范围内的扇区。使用 CMD_SECTOR_ERASE。
ping	X	X	X	发送 ping 命令。
readDeviceld				使用 CMD_MEMORY_READ 读取器件 ID。
readFlashSize				使用 CMD_MEMORY_READ 读取闪存大小。
readMemory32	X	X	X	从器件存储器中读取 32 位字。
readMemory8				使用 CMD_MEMORY_READ 从器件存储器中读取 8 位字。
readRamSize				使用 CMD_MEMORY_READ 读取 RAM 大小。
readStatus	X	X	X	读取引导加载程序状态。
复位	X	X	X	使用 CMD_RESET 复位器件。
运行	X			从指定地址运行器件 CPU。CC26xx 不支持。
setCCFG		X	X	设置 CC26xx CCFG。CC2538 不支持。
setXosc	X			切换到外部振荡器。CC26xx 不支持。
writeFlashRange	X	X	X	使用 CMD_DOWNLOAD 和 CMD_DATA_SEND 写入闪存。
writeMemory32	X	X	X	使用 CMD_MEMORY_WRITE 向器件存储器写入 32 位字。
writeMemory8				使用 CMD_MEMORY_READ 和 CMD_MEMORY_WRITE 向器件存储器执行 8 位写入操作。

3.2.1 器件特定函数

ROM 引导加载程序中有一些命令在 CC2538、CC26x0 和 CC26x2 之间不同，这意味着这些器件的 SBL 函数也有一些不同；表 3-3 中显示了这些差异。

所选硬件不支持的 SBL 函数会返回常量 SBL_UNSUPPORTED_FUNCTION，而不执行任何操作。

表 3-3. 器件特定 SBL 函数

SBL 函数名称	CC2538	CC26x0	CC26x2	说明
cmdDownloadCrc	不支持	不支持	支持 (CMD_DOWNLOAD_CRC)	CC2538 和 CC26x0 引导加载程序中未实现。
eraseFlashBank()	不支持	支持 (CMD_BANK_ERASE)	支持 (CMD_BANK_ERASE)	这将擦除 CC26x0 和 CC26x2 所有未受保护的闪存扇区；对于 CC2538，可通过对整个闪存存储器使用 eraseFlashRange 来实现。
setCCFG	不支持	支持 (CMD_SET_CCFG)	支持 (CMD_SET_CCFG)	CC2538 引导加载程序中未实现。
setXosc	支持 (CMD_SET_XOSC)	不支持	不支持	CC26x0 和 CC26x2 引导加载程序中未实现。
运行	支持 (CMD_RUN)	不支持	不支持	CC26x0 和 CC26x2 引导加载程序中未实现。

4 示例项目

SBL 的示例应用是为 Visual Studio C++ Professional 2015 创建的，并使用 CC2538、CC2650 和 CC2652R 开发套件中包含的硬件进行了测试。可通过以下网址下载示例应用：<http://www.ti.com/cn/lit/zip/swra466>

SblAppEx 是一个测试应用，它使用 CC2538、CC26x0 或 CC26x2 ROM 引导加载程序执行以下操作：

- 擦除闪存
- 为闪存编程
- 验证闪存内容
- 复位器件

测试应用的成功执行应该类似于图 4-1。

```
+-----+
| Serial Bootloader Library Example Application
+-----+

+-----+
| COM ports:
+-----+
|Idx  | Description
| 0    | XDS100v3 Class USB Serial Port (COM4)
+-----+

Select COM port index: 0
+-----+
| Supported devices:
+-----+
|Idx  | Device
| 0    | CC2538
| 1    | CC13x0/CC26x0
| 2    | CC2640R2
| 3    | CC13x2/CC26x2
+-----+

Select target device: 0
Enable device CC2538 XOSC? (Y/N): y

Connecting (COM4 @ 230400 baud) ...
Trying to set device XOSC.
Device XOSC activated.
100% (902.00ms)
Erasing flash ...
100% (5528.00ms)
Writing flash ...
100% (66620.00ms)
Calculating CRC on device ...
100% (301.00ms)
Comparing CRC ...
OK
Resetting device ...
OK
```

图 4-1. CC2538 示例应用的成功执行

4.1 硬件设置

SBL 通过 PC 上的串行 COM 端口与 ROM 引导加载程序进行通信。如果没有内置 COM 端口，USB 转串口可充当虚拟 COM 端口。

图 4-2 演示了将 PC 连接到器件的两种不同方式：一种是通过使用电平转换器将 UART 信号从 RS232 转换为 TTL 信号，另一种是使用类似于 SmartRF06EB 上所用的 USB 转 UART 桥 [4]。

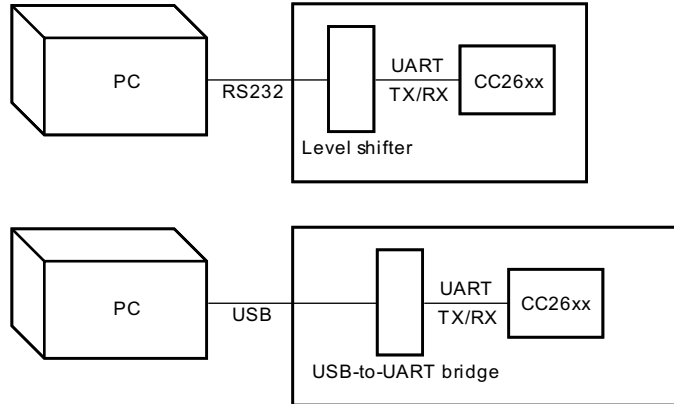


图 4-2. PC 到 UART 连接

4.1.1 SmartRF06EB 虚拟 COM 端口

SmartRF06EB [4] 提供对虚拟 COM 端口的内置支持，可与 CC2538EM [5] 或 CC2650EM 一同使用 [6]。

若要在 SmartRF06EB 上启用虚拟 COM 端口，必须在“Enable UART over XDS100v3”接头上安装跳线，并且应安装“XDS100V3 BYPASS”接头上的所有跳线。

4.1.1.1 外部串行接口

如果使用 SmartRF06EB 绕过 XDS100v3 仿真器以使用外部串行接口，请将外部串行接口连接到“XDS100v3 BYPASS”接头上的 EM RX 和 EM TX 引脚，如图 4-3 所示。

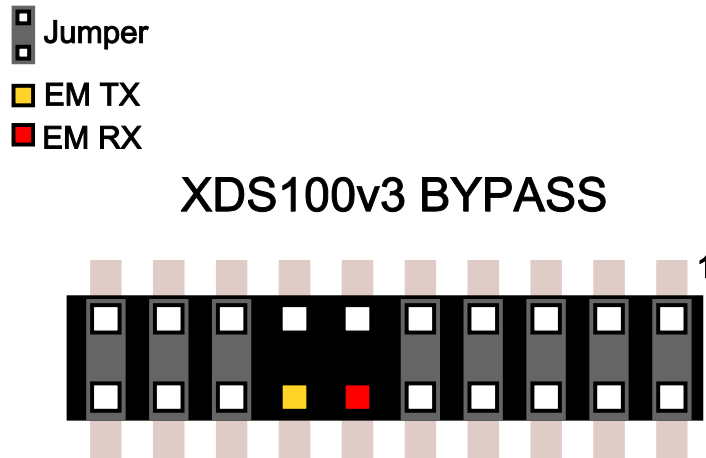


图 4-3. XDS100v3 仿真器旁路接头上的 EM TX 和 RX 引脚

4.1.2 LaunchPad 虚拟 COM 端口

适用于 CC2640R2 [7]、CC1312R [8]、CC1352R/P [9]/[10]、CC2642R [11] 和 CC2652R [11] 的 LaunchPad 提供对虚拟 COM 端口的内置支持。LaunchPad 配备了 XDS110 类辅助数据端口和 XDS110 类应用/用户 UART。在此应用中，应使用 XDS110 类应用/用户 UART。

4.1.3 引导加载程序后门

sblAppEx 示例是为 CC2538 和 CC2650 (7x7) 评估模块 (EM) 以及 CC26x2 和 CC2640R2 LaunchPad 编写的。编程到器件上的应用程序映像会触发 SmartRF06EB 或 LaunchPad 使 LED 闪烁。固件映像会启用引导加载程序后门，以便可以使用 IO 引脚触发引导加载程序。

表 4-1 和表 4-2 显示了应用程序映像打开引导加载程序后门所使用的 I/O 引脚。该 I/O 引脚连接到“SmartRF06EB SELECT”按钮。要进入引导加载程序后门，请在按下 SmartRF06EB 上“EM reset”按钮的同时按住“SELECT”按钮（对应于逻辑“0”）。对于 LaunchPad，当按下“LaunchPad reset”按钮进入引导加载程序后门时，引导加载程序后门使能引脚必须接地。

表 4-1. 应用示例 IO 配置：评估模块套件

信号	CC2538	CC26x0			EM 引脚
		QFN48/7x7	QFN32/5x5	QFN32/4x4	
UART_RX	PA0	DIO2	DIO1	DIO1	1.07
UART_TX	PA1	DIO3	DIO0	DIO2	1.09
引导加载程序后门使能	PA3	DIO11	DIO9	DIO7	1.14

表 4-2. 应用示例 IO 配置：LaunchPad

信号	CC2640R2 ⁽¹⁾	CC26x2R	CC1312R	CC1352x	LaunchPad 引脚
UART_RX	DIO2	DIO2	DIO2	DIO12	3 ⁽²⁾
UART_TX	DIO3	DIO3	DIO3	DIO13	4 ⁽²⁾
引导加载程序后门使能	DIO11	DIO11	DIO11	DIO11	18

(1) 该引脚排列仅对 QFN48/7x7 封装有效。

(2) CC2640R2 LaunchPad 的相反顺序。

4.2 软件设置

sblAppEx 示例有两个配置选项：器件类型和波特率。

4.2.1 器件类型

使用 sblAppEx.cpp 中的 deviceType 变量配置器件类型。它控制着 SBL 能够使用哪些引导加载程序命令，以及 SblAppEx 将哪个固件映像编程到器件上。deviceType 变量是器件名称的二进制编码小数 (BCD)。表 4-3 列出了支持的器件类型和对应的 deviceType 值。

表 4-3. 配置：deviceType

器件	器件类型值
CC2538	0x2538
CC13x0/CC26x0	0x2650
CC2640R2	0x2640
CC13x2/CC26x2	0x2652

4.2.2 波特率

使用 sblAppEx.cpp 中的“baudRate”变量来配置波特率。节 2.4.2 介绍了 CC2538、CC26x0 和 CC26x2 支持的 UART 波特率。所有器件都支持默认波特率。

4.3 程序流程

本节介绍了 sblAppEx 示例工程（可从以下网址下载：<http://www.ti.com/cn/lit/zip/swra466>）中讨论的 SBL 函数调用，以及使用的底层引导加载程序命令。

4.3.1 枚举 COM 端口

SBL 中的枚举函数使用 Windows API 来列出可用的 COM 端口。第一个参数是指向 ComPortElement 数组的指针。第二个参数用于指定要枚举的 COM 端口数上限。如果要使用的 COM 端口已知，则可以跳过此函数调用。

4.3.2 创建器件

必须告知 SBL 它正在使用哪个器件。Create 函数支持单个参数，表 4-3 中提供了支持的输入值。Create 函数会返回支持指定硬件的 SblDevice 类的实例。

4.3.3 连接

正确的函数采用以下两个参数：COM 端口号（参见节 4.3.1）和波特率（参见节 2.4.2）。

CC2538 ROM 引导加载程序支持从器件的内部振荡器切换到外部振荡器（如果可用）。切换到外部振荡器会增加 CC2538 ROM 引导加载程序支持的最大波特率。如果要使用外部振荡器，则可将第三个参数（布尔值 TRUE）传递给连接函数，第三个参数是可选的，默认情况下为 FALSE。

为了检查连接是否已经初始化，SBL 的 initCommunication 函数会发送一个虚拟命令并等待引导加载程序以 ACK 进行响应。如果不存在连接，initCommunication 函数会发送自动波特率例程（如节 2.4.2 所述），预期会收到 ROM 引导加载程序发出的 ACK。该序列的示例如图 4-4 所示。

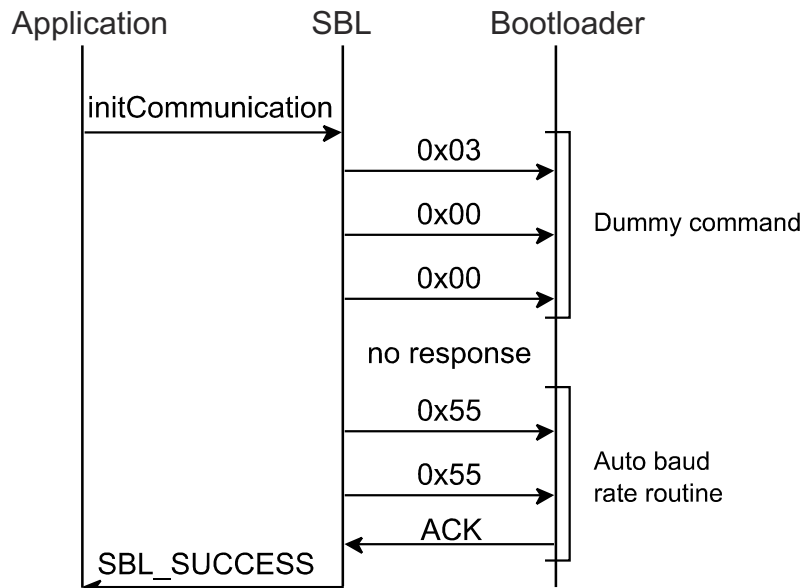


图 4-4. 带有未初始化引导加载程序的 initCommunication 函数的序列图表

建立连接后，连接函数使用串行引导加载程序命令 CMD_GET_CHIP_ID 检索器件 ID，使用命令 CMD_MEMORY_READ 从存储闪存大小和 RAM 大小的位置读取这些值。

4.3.4 擦除闪存范围

eraseFlashRange 函数对 CC2538 使用引导加载程序命令 CMD_ERASE，对 CC26x0 和 CC26x2 使用 CMD_SECTOR_ERASE。

CC26x0 和 CC26x2 CMD_SECTOR_ERASE 采用地址参数并擦除该地址所在的闪存扇区（CC26x0 器件为 4KB，CC26x2 器件为 8KB）。

CC2538 CMD_ERASE 命令需要另一个用于指定擦除大小的参数。CC2538 引导加载程序会擦除 [地址, 地址 + 大小] 范围覆盖的闪存扇区 (2KB)。

在每个引导加载程序擦除命令之后，eraseFlashRange 使用 CMD_GET_STATUS 命令来检查引导加载程序状态。

图 4-5 所示为使用串行引导加载程序协议进行闪存擦除的序列图表。命令中的最后四个字节 (datasize) 特定于 CC2538。对于 CC26x0 和 CC26x2，必须为要擦除的每个闪存扇区重复使用 CMD_SECTOR_ERASE 命令（以及随后的 CMD_GET_STATUS）。

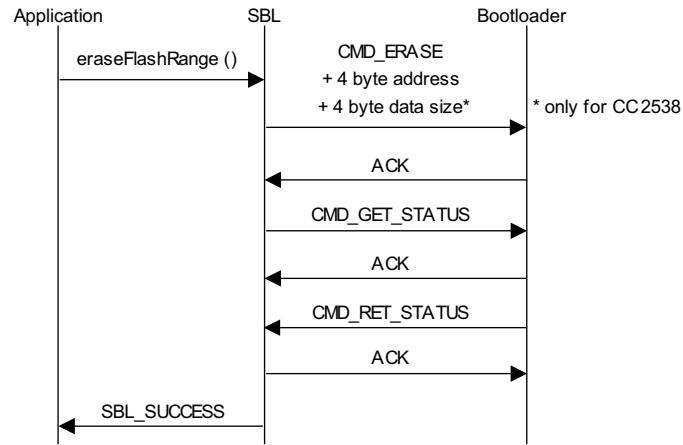


图 4-5. 闪存扇区擦除的序列图表

如果要擦除 CC26x0 和 CC26x2 上的整个闪存存储器，则应使用 `CMD_BANK_ERASE` 命令。这将在一次操作中擦除整个闪存存储器，比单独删除各个扇区要快。

4.3.5 写入闪存范围

若要将数据写入闪存存储器，可使用 SBL 函数 `writeFlashRange`；此函数会将 `CMD_DOWNLOAD` 命令连同起始地址和下载大小（以字节为单位）发送到引导加载程序。之后，引导加载程序便可接收指定数量的数据并将其从指定地址开始写入闪存。

`CMD_SEND_DATA` 命令可用于传输数据。每个 `CMD_SEND_DATA` 命令最多可传输 252 字节的数据。如果要下载的数据大于 252 字节，则必须重复使用 `CMD_SEND_DATA` 命令。SBL `writeFlashRange` 函数负责将数据传输拆分为多个 `CMD_SEND_DATA` 命令。

在 `CMD_DOWNLOAD` 命令和每个 `CMD_SEND_DATA` 命令之后，使用 `CMD_GET_STATUS` 命令来读取引导加载程序的状态。这是为了确保起始地址和固件大小有效，并且数据已成功编程到闪存中。如果状态指示错误，引导加载程序的内部地址指针不会递增，从而允许重新传输数据。

图 4-6 演示了使用 SBL 函数 writeFlashRange 的闪存写入序列。

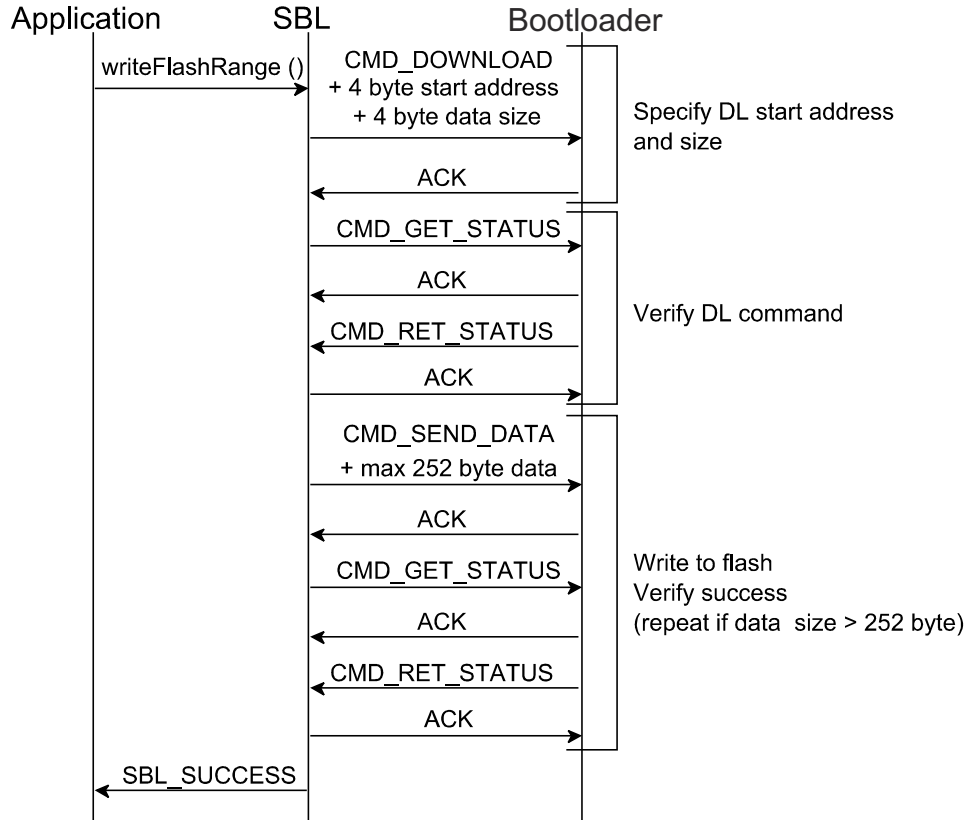


图 4-6. 闪存写入的序列图表

4.3.6 计算 CRC32

为了验证固件是否已成功编程到闪存存储器中，可以使用 SBL 函数 calculateCrc32 从引导加载程序中获取闪存存储器指定部分的 CRC32 校验和。calculateCrc32 函数使用命令 CMD_CRC32 以及会包含在 CRC32 校验和中的起始地址和字节数。

对于 CC26x0 和 CC26x2，引导加载程序还需要读取重复计数。将此设置为 0x00000000 可确保数据位置仅被读取一次。

CC2538、CC26x0 和 CC26x2 引导加载程序使用 CRC-32-IEEE 802.3 通过以下多项式来计算 CRC 校验和。

$$\text{CRC32}_{\text{poly}} = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

在 SBL 示例工程中包含了有关如何使用 `CRC32poly` 来计算校验和的示例。`calculateCrc32` 函数的序列图表如图 4-7 所示。

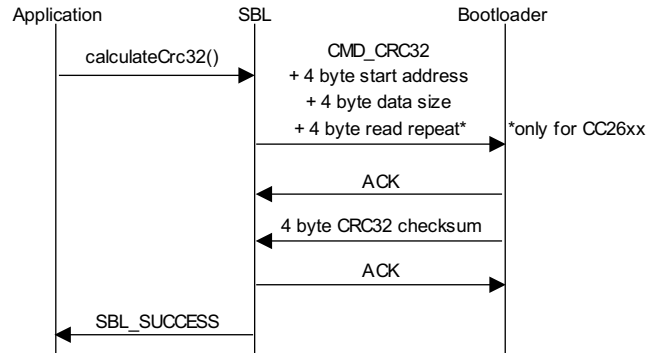


图 4-7. CRC32 命令的序列图表

4.3.7 复位

若要在写入和验证固件后将其运行，必须使用 SBL 函数复位。复位函数向引导加载程序发送 `CMD_RESET` 命令以调用复位。发送 `CMD_RESET` 命令并从引导加载程序接收到 `ACK` 后，主机和器件之间的连接将中断。可以在图 4-8 中观察到复位函数的序列图表。

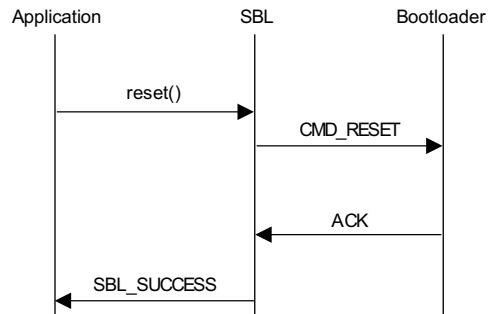


图 4-8. SBL 函数复位的序列图表

5 参考文献

1. 德州仪器 (TI) : 《CC2538 ROM 用户指南》
2. 德州仪器 (TI) : 《CC13x0、CC26x0 SimpleLink™ 无线 MCU 技术参考指南》
3. 德州仪器 (TI) : 《CC13x2、CC26x2 SimpleLink™ 无线 MCU 技术参考指南》
4. 德州仪器 (TI) : 《SmartRF06 评估板 (EVM) 用户指南》
5. CC2538DK : <https://www.ti.com.cn/tool/cn/CC2538DK>
6. CC2650DK : <https://www.ti.com.cn/tool/cn/CC2650DK>
7. CC2640R2 LaunchPad : <https://www.ti.com.cn/tool/cn/LAUNCHXL-CC2640R2>
8. CC1312R LaunchPad : <https://www.ti.com.cn/tool/cn/LAUNCHXL-CC1312R1>
9. CC1352R LaunchPad : <https://www.ti.com.cn/tool/cn/LAUNCHXL-CC1352R1>
10. CC1352P LaunchPad : <https://www.ti.com.cn/tool/cn/LAUNCHXL-CC1352P>
11. CC26x2R LaunchPad : <https://www.ti.com.cn/tool/cn/LAUNCHXL-CC26X2R1>

6 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

Changes from Revision C (March 2020) to Revision D (August 2021)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式。.....	3

Changes from Revision B (June 2019) to Revision C (March 2020)	Page
• 对节 4 进行了更新。.....	12
• 为示例应用添加了指向下载页面的网址。.....	12

重要声明和免责声明

TI 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安全、安保或其他要求。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 (<https://www.ti.com/legal/termsofsale.html>) 或 [ti.com](https://www.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。

邮寄地址: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2021, 德州仪器 (TI) 公司

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