

摘要

MSPM0 開機載入程式 (也稱為 BSL) 提供一種透過標準序列介面 (UART 或 I2C) 對裝置記憶體 (快閃記憶體和 RAM) 進行編程和驗證的方法。

目錄

1 BSL 功能概覽.....	2
2 詞彙.....	3
3 BSL 架構.....	4
3.1 設計.....	4
3.2 BSL 叫用.....	5
3.3 記憶體.....	6
3.4 BSL 配置.....	7
3.5 BSL 狀態.....	7
4 開機載入程式協定.....	9
4.1 封包格式.....	9
4.2 UART 和 I2C BSL 協定.....	9
4.3 開機載入程式核心命令.....	10
4.4 BSL 核心回應.....	17
4.5 開機載入程式安全性.....	19
5 使用開機載入程式的程式流範例.....	21
6 二次開機載入程式.....	23
6.1 二次開機載入程式範例.....	23
7 介面外掛程式.....	25
7.1 實作.....	25
7.2 Flash 外掛程式類型.....	27
7.3 覆寫現有介面.....	28
8 參考.....	29
9 修訂記錄.....	29

註冊商標

所有商標均為其各自所有者的財產。

1 BSL 功能概覽

Bootstrap 載入程式 (BSL) 提供了透過標準 UART 或 I2C 序列介面編程或驗證裝置記憶體的方法。

可透過序列介面存取的 BSL 主要功能包括：

- 對快閃記憶體進行編程和清除
- 可傳回程式碼或資料區域的 32 位元 CRC (1KB 最小區域) 以驗證程式設計
- 可啓用程式碼或資料讀取 (預設為停用)
- 可透過指向主快閃記憶體的指標傳回韌體版本號碼
- 可以指定硬體叫用 GPIO
- 存取權始終受 256 位元密碼保護
- 可配置用以抵禦暴力攻擊的安全警示處理
- 可將新介面新增為 Flash 外掛程式
- 可使用自訂開機載入程式

2 詞彙

開機載入程式 (BSL)：用於將資料載入到裝置記憶體の開機常式

啟動碼 (BCR)：在 BOOTRST 之後執行的啓動常式，配置裝置以執行應用程式

BCR 配置：配置結構包含啟動碼 (駐留在非主快閃記憶體中) 的所有使用者可配置參數

BSL 配置：配置結構包含開機載入程式 (駐留在非主快閃記憶體中) 的所有使用者可配置參數

3 BSL 架構

3.1 設計

當偵測到有效的開機載入程式叫用條件時，啟動碼將會叫用開機載入程式。只有在 BCR 配置的 BSL 模式欄位中啟用開機載入程式時，才會叫用該程式。

開機載入程式啟動後，會先執行「初始化」階段，在此階段完成 BSL 配置的初始檢查，並針對開機載入程式作業配置裝置。

接著，開機載入程式會進入「介面自動偵測」階段。在此階段中，BSL 配置所有可用的 BSL ROM 介面和快閃外掛程式介面 (如果已註冊)。然後 BSL 會透過所有介面逐一輪詢資料。在其中一個介面收到有效的連線封包時，該介面將被視為用於進一步通訊的作用介面，而所有其它介面都會停用。介面探索 10 秒完成，如果沒有偵測到介面，裝置會進入 STANDBY 待機模式。

接著，BSL 進入「命令接收」階段。在此階段中，BSL 將在無限迴圈中等待主機的命令。收到有效命令後，會處理該命令，將來自 BSL 核心的回應傳送回主機。接著，回到迴路並等待下一個命令等等。如果收到「啟動應用程式」命令，開機載入程式會觸發系統重設，之後再執行啟動碼，並叫用應用程式。此階段也有 10 秒的逾時時間。如果沒有收到有效的命令，則會鎖定開機載入程式並進入睡眠模式。

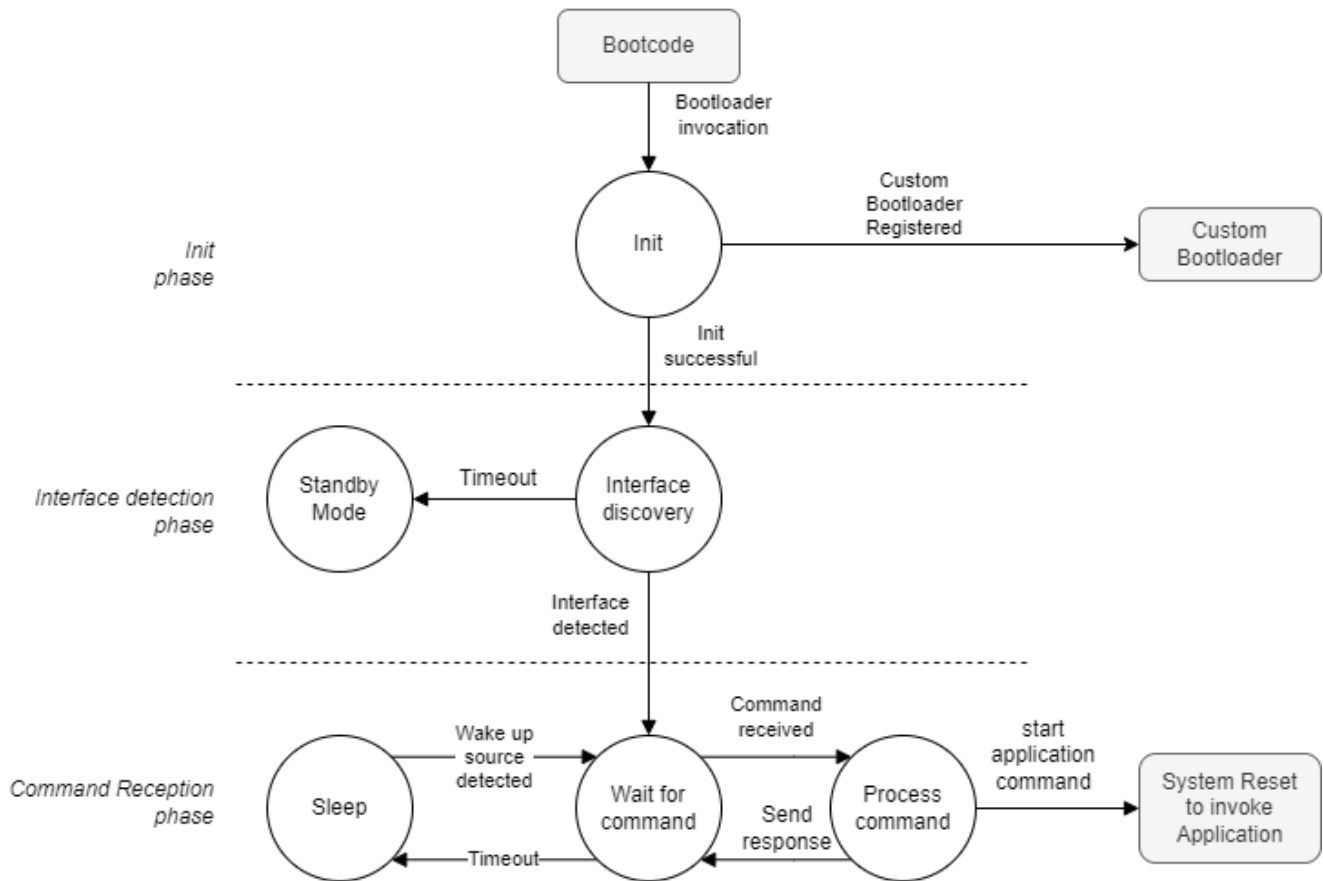


图 3-1. BSL 架構

3.1.1 逾時特性

當未偵測到任何活動時，開機載入程式將逾時，並進入低功耗模式以節省電力。

這已在以下兩個階段實作。

1. 介面自動偵測
2. 命令接收

3.1.1.1 介面自動偵測

在介面偵測階段，如果在任何介面上都沒有收到有效的連線命令達 10 秒之久，則開機載入程式會進入 STANDBY 待機模式。

需要使用 POR 才能退出此狀態，並透過再次建立 BSL 叫用條件來使用開機載入程式。

3.1.1.2 命令接收

在命令接收階段，如果 10 秒內未收到有效命令，則開機載入程式將進入 SLEEP 睡眠模式。若要從 SLEEP 睡眠模式喚醒裝置，應在作用介面上進行資料交易。

進入睡眠模式前，開機載入程式將會鎖定，以縮小攻擊面。因此，從低功耗模式喚醒後，需要透過傳送 256 位元 BSL 密碼 (請參見[解鎖開機載入程式命令](#)) 重新解鎖開機載入程式。

备注

開機載入程式使用者 LFCLK 用於逾時檢查。如果應用程式將外部時鐘配置為 LFCLK 的來源，則 BSL 在透過應用程式 BSL 請求叫用時會使用相同的時鐘

3.2 BSL 叫用

只有在符合 BSL 叫用條件，且如果在 BCR 配置 [TRM](#) 中啟用開機載入程式時，啟動碼才會叫用開機載入程式。

在 BCR 配置中啟用快速開機模式時，只有偵錯信箱命令和應用程式請求能叫用開機載入程式。略過其它檢查以節省執行時間。

3.2.1 空裝置

啟動碼透過檢查堆疊指標 (0x00000000) 的清除狀態和重設向量 (0x00000004) 位址偵測空白裝置。當兩個快閃記憶體位址都被發現為空白時，將會叫用開機載入程式。

3.2.2 應用程式請求

若要從應用程式叫用開機載入程式，請將 RESETLEVEL 設定為 BOOTLOADERENTRY，然後透過 RESETCMD 暫存器觸發重設。此順序會導致系統重設，並且執行啟動碼及叫用開機載入程式。

因為系統重設已發出，所以在退出應用程式時會重設所有周邊設備配置。

3.2.3 基於 GPIO 的叫用

用於 BSL 叫用的 GPIO 可以在非主記憶體的中 BSL 配置 [TRM](#) 中進行配置。

新裝置在 BSL 配置中具有 TI 編程的預設針腳詳細資訊。

可以在 BCR 配置中停用基於 GPIO 針腳的叫用。預設為啟用。

應在 POR 之前聲明 GPIO，而且該狀態在 POR 之後應至少維持 T_start 毫秒。然後才可以解除 GPIO 針腳狀態。

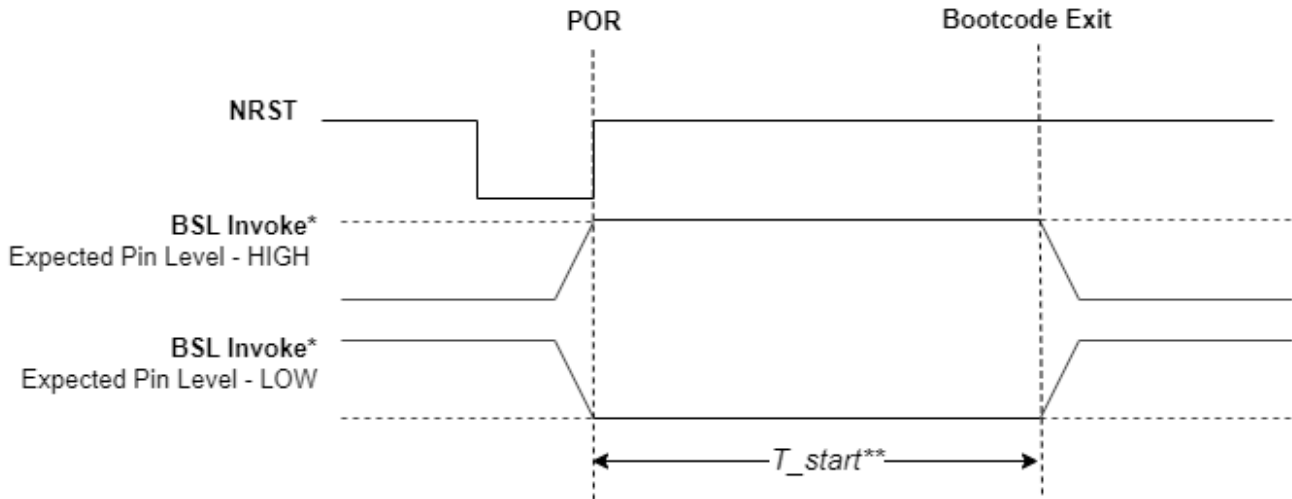


图 3-2. 從 GPIO 叫用

* - 可使用作為「BSL 叫用」的 GPIO 針腳，並且可在 BSL 配置中配置「預期針腳位準」

** - T_{start} 是指在裝置專屬產品規格表中指定的冷開機啟動時間

备注

如果啟用以針腳為基礎的 BSL 呼叫，則配置的 GPIO 針腳應拉高或降低。它不應該浮動，其可能導致意外的 BSL 執行。

3.2.4 偵錯信箱命令

當除錯介面可用時，可以透過偵錯系統信箱 (DSSM) 傳送開機載入程式叫用命令。

如需 DSSM 命令用法的詳細資訊，請參閱 [MSPM0x_Technical_Reference_Manual](#) DSSM 命令一節。

3.2.5 其他

3.2.5.1 預開機應用程式驗證

在 BCR 配置中，如果啟用了應用程式 CRC 驗證 [TRM](#)，啟動碼會驗證指定應用程式記憶體範圍的 CRC。如果 CRC 不正確，則會叫用開機載入程式。在這情況下，只有在啟用時才會叫用開機載入程式。否則，啟動碼會記錄災難性錯誤，並導致開機失敗。

3.3 記憶體

3.3.1 SRAM 記憶體使用量

SRAM 記憶體配置說明開機載入程式運作所使用的記憶體。

- 資料和堆疊部分：由 BSL 用於運作。結束開機載入程式時，會清除 SRAM 的這些部分。
- 可變緩衝區空間：用於儲存 BSL 通訊期間接收/傳送的資料封包的緩衝區空間

主機允許讀取和寫入存取的 SRAM 記憶體為 BSL 緩衝區起始位址到 [SRAM 結束位址 - 0x120]，其中 SRAM 結束位址由每個裝置中可用的 SRAM 記憶體決定。由於相同的 SRAM 空間與可變緩衝區空間共用，因此在 SRAM 寫入/讀取運作期間可能會遭到覆寫。

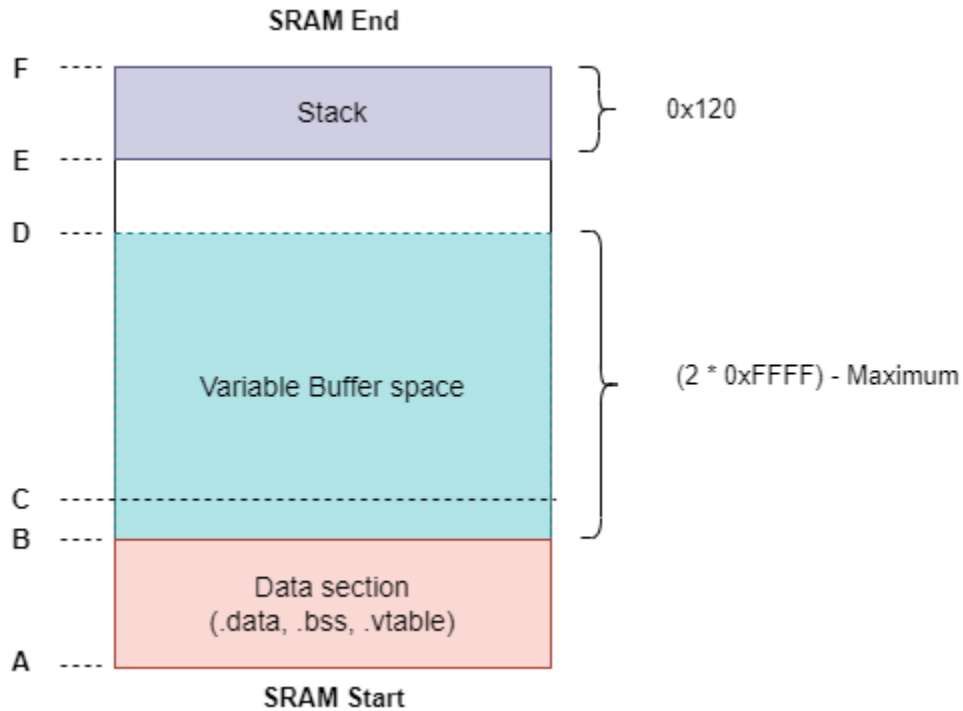


图 3-3. SRAM 使用量

A - SRAM 起始位址 (0x20000000)

B- 當沒有登錄 Flash 外掛程式介面時，從「取得裝置資訊」命令回應中得知「BSL 緩衝器開始位址」

C- 從「取得裝置資訊」命令回應中得知「BSL 緩衝器開始位址」。當沒有登錄 Flash 外掛程式介面時，此介面將與「B」相同

D- BSL 緩衝區結束位址 = 「BSL 緩衝區起始位址」 + (2 * 「BSL 最大緩衝區大小」)，其中 BSL 緩衝區起始位址和 BSL 最大緩衝區大小可從「取得裝置資訊」命令回應得知

E- 堆疊分配的起始位址 (E - 0x120)。當「BSL 最大緩衝區大小」小於 0xFFFF 時，此值將與「D」相同

F- 裝置中可用 SRAM 記憶體的結束位址。請參閱裝置專屬產品規格表以了解詳情。

B-C 部分：

- 在 BSL 配置中登錄時，將分配給 Flash 外掛程式運作的資料部分

C-D 部分：

- 用於儲存資料封包的緩衝區空間
- 最大尺寸為 (2 * 0xFFFF)

C-E 部分：

- 透過 BSL 命令進行 SRAM 讀取和寫入運作的可用記憶體

3.4 BSL 配置

非主快閃記憶體中的 BSL 配置允許 BSL 使用的某些參數由使用者自訂。

如需了解有關可用配置的詳細資訊，請參閱 MSPM0 技術參考手冊 - 配置記憶體章節。

3.5 BSL 狀態

BSL 狀態提供執行期間發生並導致 BSL 無回應的任何錯誤詳細資訊。此 32 位元狀態資訊儲存在特定 SRAM 位址 (0x200000C0) 中，如果在 BCR 配置中啟用偵錯存取，則可以透過偵錯器讀取。

表 3-1. BSL 狀態解譯

位元組 4	位元組 3	位元組 2	位元組 1
已保留	HW 錯誤	HW 錯誤詳情	SW 錯誤

SW 錯誤

錯誤	說明
0x01	BSL 配置 CRC 錯誤

HW 錯誤

錯誤	說明	HW 錯誤詳情
0x07	NMI 例外	NMIIDX - NMI 中斷索引暫存器資料

4 開機載入程式協定

4.1 封包格式

BSL 資料包有一個分層結構。BSL 內核命令包含由 BSL 處理的實際命令資料。除了標準 BSL 命令，每個內核命令的前後有被稱為外設介面程式碼 (PI 程式碼) 的包裝器資料。這個包裝器資料是所使用的外設和協議的特定資訊，並且它包含有允許 BSL 內核命令正確傳輸的資訊。和在一起，包裝器和內核命令組成了一個 BSL 資料包。

PI 程式碼	BSL 核心資料	PI 程式碼
--------	----------	--------

4.2 UART 和 I2C BSL 協定

UART 和 I2C BSL 協定的資料封包具有下列結構。

- 標頭位元組指出使用的協定和封包類型 (命令或回應封包)。
- 長度欄位包含 BSL 核心資料的大小 (以位元組為單位)。
- BSL 核心資料，包含命令/回應 ID 和位址，以及命令所需的資料
- CRC32 欄位包含為 BSL 核心資料中的資料計算的 CRC

PI 程式碼		BSL 核心資料	PI 程式碼
標頭 (1 位元組)	長度 (2 位元組)	BSL 核心命令/回應	CRC32 (4 位元組)

根據核心資料欄位，資料封包分類為命令封包或回應封包。

命令封包是第一個傳輸到 BSL 的封包。第二個封包是從 BSL 接收的回應封包。回應封包涵蓋兩個元件 BSL 確認和 BSL 核心回應。在這兩種情況下，系統會從 BSL 接收每個傳送的命令封包確認。但並未針對每個命令接收 BSL 核心回應。



图 4-1. BSL 協定

4.2.1 BSL 確認

BSL 軟體的周邊設備介面部分會對 BSL 資料封包的包裝函式部分進行分析。如果資料傳輸中有錯誤，會立即傳送錯誤訊息。在成功接收到所有資料後傳送一個 ACK，這不表示已經正確地執行命令 (或甚至命令為有效的)，而是意味著資料封包已正確地格式化，並傳遞到 BSL 核心軟體進行解譯。

BSL 通訊協定規定，除了傳送的 BSL 資料封包外，每個傳送的 BSL 資料封包都會以單位元組確認回應。表格列出 BSL 的確認回應。如果傳送了 ACK 以外的確認位元組，BSL 不會傳送任何 BSL 資料封包。主機程式設計工具必須檢查確認錯誤並重新嘗試傳輸。

資料	意義
0x00	BSL_ACK (成功接收封包)
0x51	BSL_ERROR_HEADER_INCORRECT
0x52	BSL_ERROR_CHECKSUM_INCORRECT
0x53	BSL_ERROR_PACKET_SIZE_ZERO
0x54	BSL_ERROR_PACKET_SIZE_TOO_BIG
0x55	BSL_ERROR_UNKNOWN_ERROR
0x56	BSL_ERROR_UNKNOWN_BAUD_RATE

4.2.2 周邊設備配置

4.2.2.1 UART

使用以下配置啟用 UART：

- 使用 UART0
- 預設情況下，傳輸速率為 9600 bps。它可以透過變更傳輸速率命令進行更新
- 資料寬度：8 位元
- 停止位元：1
- 無同位檢查
- 用於 RXD 和 TXD 的針腳取自 BSL 配置

4.2.2.2 I2C

BSL 中的 I2C 介面可做為 I2C 目標。主機充當控制器並促使通訊。

- 使用 I2C0
- 依預設，I2C 目標位址為 0x48。可以在 BSL 配置中設定位址
- SCL 和 SDA 線路需要外部上拉
- 用於 SDA 和 SCL 的針腳取自 BSL 配置

备注

- BSL 不會驗證非主要 UART 和 I2C 介面中配置的針腳細節。BSL 預期針腳配置是正確的。
- 請勿將相同的針腳用於 UART 和 I2C。

4.2.2.3 CRC

資料的 CRC 必須使用以下公式計算：

- CRC32-ISO3309 多項式
- 位元反向配置
- 初始種子 - 0xFFFFFFFF

4.3 開機載入程式核心命令

BSL 命令	受保護	CMD	起始位址	資料 (位元組)	BSL 核心回應
CMD 連線	否	0x12	-	-	否
CMD 解鎖開機載入程式	否	0x21	-	D1...D32 (密碼)	是
CMD 快閃記憶體範圍清除	是	0x23	A1...A4	A1...A4 (結束位址)	是
CMD 大量清除	是	0x15	-	-	有
CMD 程式資料	是	0x20	A1...A4	D1...Dn ,	是
CMD 程式資料快速	是	0x24	A1...A4	D1...Dn ,	否
CMD 記憶體讀回	是	0x29	A1...A4	L1...L4	是
CMD 原廠重設	是	0x30	-	D1...D16 (密碼)	是
CMD 取得裝置資訊	否	0x19	-	-	有
CMD 獨立驗證	是	0x26	A1...A4	L1...L4	是
CMD 啟動應用程式	否	0x40	-	-	否
CMD 變更傳輸速率	否	0x52	-	D1 (傳輸速率 ID)	否

縮寫：

A1...A4

位址位元組，其中 A1 是最小權重位元組

D1...Dn

資料位元組，其中 'n' 受 BSL 最大緩衝區大小限制

L1...L4

資料長度位元組，其中 C1 是最小權重位元組

4.3.1 通訊

架構

Header	長度		CMD	CRC32			
0x80	0x01	0x00	0x12	C1	C2	C3	C4

說明

連線命令是第一個透過特定介面 (UART 或 I2C) 建立在主機與目標之間的連線命令。

受保護

否

命令傳回

僅 [BSL 確認](#)

例

主機 : 80 01 00 12 3A 61 44 DE

BSL: 00

4.3.2 取得裝置資訊

架構

Header	長度		CMD	CRC32			
0x80	0x01	0x00	0x19	C1	C2	C3	C4

說明

該命令用於取得適用於資料交易的版本資訊和緩衝區大小。

受保護

否

命令傳回

BSL 確認和 BSL 核心回應了裝置資訊。如需詳細資訊，請參閱[裝置資訊](#)。

例

主機 : 80 01 00 19 B2 B8 96 49

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

4.3.3 解鎖開機載入程式

架構

Header	長度		CMD	資料	CRC32			
0x80	0x21	0x00	0x21	D1...D32	C1	C2	C3	C4

說明

該命令用於解鎖開機載入程式。只有在開機載入程式解鎖之後，BSL 才會處理 [節 4.3](#) 中列出的所有受保護命令。

如果主機傳送的密碼不正確，BSL 會傳送錯誤回應，接著裝置會進入睡眠模式 2 秒。在此 2 秒時段中，未接收或處理任何命令。2 秒後，主機可以傳送下一個密碼。如果傳送錯誤密碼 3 次，即會採取安全警示動作。在 BSL 配置中可以配置安全警示動作。如需詳細資訊，請參閱[節 4.5.1.1](#)。

受保護

否

資料

儲存在 BSL 配置記憶體中的 32 位元組 BSL 密碼。新裝置的預設值為所有 0xFF。

命令傳回

BSL 確認和 BSL 核心回應了有關運作狀態的訊息。如需詳細資訊，請參閱 [BSL 核心訊息](#) 一節。

例

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

BSL: 00 08 02 00 3B 00 38 02 94 82

4.3.4 程式資料
架構

Header	長度		CMD	地址	資料	CRC32			
0x80	L1	L2	0x20	A1...A4	D1...Dn	C1	C2	C3	C4

說明

程式命令用於在從 A1...A4 開始的記憶體位址中透過 Dn 寫入資料 D1。此命令執行封鎖寫入。完成編程後，向主機傳送訊息回應。

主快閃記憶體 (應用程式記憶體)、非主快閃記憶體 (配置記憶體) 和 SRAM 記憶體允許編程。如需絕對位址範圍的詳細資訊，請參閱裝置專屬產品規格表。

在進行程式設定之前，主機應清除快閃記憶體。有關清除主快閃記憶體區域的詳細資訊，請參閱快閃記憶體範圍清除、大量清除。非主快閃記憶體只能透過原廠重設命令清除。

由於快閃記憶體控制器的特性，資料的起始位址和長度應與快閃記憶體編程的 8 位元組對齊。

备注

主機無法完全存取 SRAM 記憶體。如需詳細資訊，請參閱 [節 3.3.1](#)。

受保護

是

地址

要編程的記憶體區域的起始位址。A1...A4，其中 A1 是 32 位元位址中最小權重位元組。

資料

要寫入指定位址的資料位元組。可傳送的最大資料受裝置的緩衝區大小限制。從「取得裝置資訊」[命令](#)得知緩衝區大小。

命令傳回

BSL 確認和 BSL 核心回應了有關運作狀態的訊息。如需詳細資訊，請參閱 [節 4.4.1](#) 一節。

例

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

BSL: 00 08 02 00 3B 00 38 02 94 82

4.3.5 程式資料快速
架構

Header	長度		CMD	地址	資料	CRC32			
0x80	L1	L2	0x24	A1...A4	D1...Dn	C1	C2	C3	C4

說明

程式資料快速命令與程式資料命令相同，但此命令執行非封鎖寫入，以加速編程流程。對於此命令，BSL 不會傳送 BSL 核心訊息回應來指出編程是否成功。

受保護

是

地址

要編程的記憶體區域的起始位址。A1...A4，其中 A1 是 32 位元位址中最小權重位元組。

資料

要寫入指定位址的資料位元組。可傳送的最大資料受裝置的緩衝區大小限制。從「取得裝置資訊」命令得知緩衝區大小。

命令傳回

BSL 確認。

例

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

BSL: 00

4.3.6 讀回資料

架構

Header	長度		CMD	地址	資料	CRC32			
0x80	0x09	0x00	0x29	A1...A4	L1...L4	C1	C2	C3	C4

說明

此命令用於讀出位址 A1...A4 起始的資料。

應在 BSL 配置中啟用讀出，以使用此命令讀出資料。在 BSL 配置中預設停用該項。

主快閃記憶體 (應用程式記憶體)、非主快閃記憶體 (配置記憶體) 和 SRAM 記憶體允許讀出資料。

备注

主機無法完全存取 SRAM 記憶體。如需詳細資訊，請參閱 [SRAM 記憶體使用量](#)。

受保護

是

地址

要讀回的記憶體區域的起始位址。A1...A4，其中 A1 是 32 位元位址中最小權重位元組。

資料

要讀取的資料大小 (以位元組為單位)，L1...L4 中 L1 是最小權重位元組。可讀取的最大資料受裝置的緩衝區大小限制。從「取得裝置資訊」命令得知緩衝區大小。

命令傳回

如果讀回命令有效，BSL 確認和 BSL 核心會回應請求的資料。如需詳細資訊，請參閱 [節 4.4.3](#)。

如果讀回命令的位址/長度無效，或停用讀出，對應的錯誤將當作 BSL 確認後的訊息回應來傳送。

例

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

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

4.3.7 快閃記憶體範圍清除

架構

Header	長度		CMD	地址	資料	CRC32			
0x80	0x09	0x00	0x23	A1...A4 (起始位址)	A1...A4 (結束位址)	C1	C2	C3	C4

說明

快閃記憶體範圍清除命令用於清除指定的快閃記憶體區域。快閃記憶體按分區 (1 KB) 清除，小於該大小的便無法清除。

當起始與結束位址位於不同的快閃記憶體區域時，BSL 會清除起始與結束位址之間的所有快閃記憶體區域，包括涵蓋這些位址的區域。

此命令僅可用於清除主快閃記憶體。無法進行非主要清除。

結束位址不得小於起始位址。

受保護

是

地址

要清除的記憶體區域的起始位址。A1...A4，其中 A1 是 32 位元位址中最小權重位元組。

資料

要清除的記憶體區域的結束位址。A1...A4，其中 A1 是 32 位元位址中最小權重位元組。

命令傳回

BSL 確認和 BSL 核心回應了有關運作狀態的訊息。如需詳細資訊，請參閱 [節 4.4.1](#) 一節。

例

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

BSL: 00 08 02 00 3B 00 38 02 94 82

4.3.8 大量清除

架構

Header	長度		CMD	CRC32			
0x80	0x01	0x00	0x15	C1	C2	C3	C4

說明

大量清除命令會清除裝置中可用的完整主快閃記憶體 (應用程式記憶體)。

BCR 配置記憶體中的大量清除配置不影響此 BSL 命令。

當快閃記憶體區域在 BCR 配置中受到靜態寫入保護，該區域無法清除。

受保護

是

命令傳回

BSL 確認和 BSL 核心回應了有關運作狀態的訊息。如需詳細資訊，請參閱 [節 4.4.1](#)。

例

主機：80 01 00 15 99 F4 20 40\

BSL: 00 08 02 00 3B 00 38 02 94 82

4.3.9 原廠重設

架構

Header	長度		CMD	資料	CRC32			
0x80	L1	L2	0x30	D1...D16	C1	C2	C3	C4

說明

原廠重設命令會將完整的主快閃 (應用程式) 記憶體和非主快閃 (配置) 記憶體清除。

處理此命令會受到 BCR 配置記憶體中的原廠重設配置影響。

原廠重設為

- 如果「已啟用」則允許不使用密碼
- 如果「以密碼啟用」則允許使用密碼
- 如果「已停用」則不允許

當快閃記憶體區域在 BCR 配置中受到靜態寫入保護，該區域無法清除。

受保護

是

資料

儲存在 BCR 配置記憶體中的 16 位元組原廠重設密碼。預設密碼為所有 0xFF。只有在 BCR 配置中的原廠重設為「以密碼啟用」時，才需要密碼。

命令傳回

BSL 確認和 BSL 核心回應了有關運作狀態的訊息。如需詳細資訊，請參閱節 4.4.1。

CAUTION

在執行原廠重設後，直到非主配置恢復為止，這當中系統很容易受到潛在的鎖定情況的影響，在這種情況下不可能再次存取裝置。

例

主機：80 01 00 30 DE 20 24 0B

BSL: 00 08 02 00 3B 00 38 02 94 82

4.3.10 獨立驗證

架構

Header	長度		RSP	資料	CRC32			
0x08	0x05	0x00	0x32	D1...D4	C1	C2	C3	C4

說明

此命令用於驗證儲存在指定記憶體範圍內的資料的 CRC。這可以更快地驗證已編程的資料。它要求資料大小至少為 1kB。

主快閃記憶體 (應用程式記憶體)、非主快閃記憶體 (配置記憶體) 和 SRAM 記憶體允許 CRC 驗證。

备注

主機無法完全存取 SRAM 記憶體。如需詳細資訊，請參閱節 3.3.1。

受保護

是

地址

要驗證的記憶體區域的起始位址。A1...A4，其中 A1 是 32 位元位址中最小權重位元組。

資料

要驗證的資料大小 (以位元組為單位)，L1...L4 中 L1 是最小權重位元組。1kB <=大小 <= 64 KB。

命令傳回

BSL 確認和 BSL 核心回應了針對請求記憶體區域計算的 CRC 值。如需詳細資訊，請參閱節 4.4.5。

如果驗證命令的位址/長度無效，對應的錯誤將當作 BSL 確認後的訊息回應來傳送。請參閱節 4.4.1。

例

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

BSL: 00 08 02 00 3B 05 B7 F6 FE F2

4.3.11 啟動應用程式

架構

Header	長度		CMD	CRC32			
0x80	0x01	0x00	0x40	C1	C2	C3	C4

說明

啟動應用程式命令，發出系統重設，致使開機載入程式退出，重新執行啟動碼，進而啟動應用程式。

受保護

否

命令傳回

BSL 確認

例

主機：80 01 00 40 E2 51 21 5B

BSL:00

4.3.12 變更傳輸速率

架構

Header	長度		CMD	資料	CRC32			
0x80	0x02	0x00	0x52	D1	C1	C2	C3	C4

說明

此命令可用於變更 UART 介面的傳輸速率。新的傳輸速率將在傳送此封包的 BSL 確認後生效。

BSL UART 的預設傳輸速率為 9600 bps。

备注

更新傳輸速率後，如果使用解鎖開機載入程式命令傳送錯誤的 BSL 密碼，傳輸速率設定將恢復為預設值。在預設傳輸速率下應進行進一步的通訊。

受保護

否

資料

D1 傳輸速率如表中所指定。

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

命令傳回

BSL 確認

4.4 BSL 核心回應

BSL 回應	RSP	資料
記憶體讀回	0x30	D1...Dn
取得裝置資訊	0x31	D1...D24
獨立驗證	0x32	D1...D4
訊息	0x3B	MSG
詳細錯誤	0x3A	D1..D3

縮寫：

MSG

包含來自 BSL 核心回應的位元組對請求動作的結果進行說明。這可以是錯誤程式碼或成功運作確認。請注意，在要求 BSL 以資料回應的情況下 (例如，記憶體、版本、CRC 或緩衝區大小)，沒有回覆成功運作，BSL 核心立即傳送資料。

D1..Dn

資料位元組，其中 'n' 受 BSL 最大緩衝區大小限制

4.4.1 BSL 核心訊息

架構

Header	長度		RSP	資料	CRC32			
0x08	0x02	0x00	0x3B	MSG	C1	C2	C3	C4

說明

對於某些命令，BSL 將訊息回應傳送到主機，該主機指出已處理命令的狀態。該表列出了 BSL 中的所有可能訊息。

MSG	意義	可能原因 ⁽¹⁾
0x00	成功執行	
0x01	BSL 鎖定錯誤	BSL 尚未使用開機載入程式解鎖密碼命令解鎖，或者在 BSL 解鎖後，命令接收階段會發生逾時
0x02	BSL 密碼錯誤	已傳送不正確的密碼解鎖開機載入程式。
0x03	多個 BSL 密碼錯誤。已採取安全警報措施。	傳送錯誤密碼以解鎖開機載入程式 3 次。
0x04	未知命令	指定給 BSL 的命令未被識別為有效命令

MSG	意義	可能原因 ⁽¹⁾
0x05	無效的記憶體範圍	指定的記憶體範圍無效。
0x06	指令無效	指定給 BSL 的命令是已知的命令，但該命令在當時無效，無法處理。
0x07	原廠重設已停用	在 BCR 配置中停用原廠重設
0x08	原廠重設密碼錯誤	當 BCR 配置將原廠重設為「以密碼啟用」時，使用原廠重設命令傳送不正確密碼或沒有密碼
0x09	讀出錯誤	在 BCR 配置中停用記憶體讀出
0x0A	無效的位址或長度對齊	快閃記憶體編程的起始位址或資料長度未對齊 8 位元組
0x0B	用於獨立驗證的長度無效	針對獨立驗證傳送的資料大小不足 1KB

(1) 此處列出的可能原因不是狀態或錯誤的唯一原因。它只列出導致錯誤的可能軟體原因，主機可以修正這些錯誤。

4.4.2 詳細錯誤

架構

Header	長度		RSP	資料	CRC32			
0x08	0x02	0x00	0x3A	D1..D3	C1	C2	C3	C4

說明

D1 - 錯誤類型

D3,D2 - 錯誤詳情

可能的值

錯誤類型		錯誤詳情	
價值觀	說明	價值觀	說明
0xF0	快閃記憶體錯誤	0xXX	包含 FLASHCTL.STATCMD 暫存器值

4.4.3 記憶體讀回

架構

Header	長度		RSP	資料	CRC32			
0x08	L1	L2	0x30	D1...Dn	C1	C2	C3	C4

說明

命令傳回請求的資料以回應讀回命令

資料

資料 D1..Dn，其中 'n' 受 BSL 最大緩衝區大小的限制。

4.4.4 裝置資訊

架構

Header	長度		RSP	資料	CRC32			
0x08	0x19	0x00	0x31	D1...D24	C1	C2	C3	C4

說明

該命令會傳回版本資訊和 BSL 緩衝區大小，以回應 Get Identity 命令

資料

身分識別位元組	資料位元組
命令直譯器版本	[D02-D01]
組建 ID	[D04- D03]
應用程式版本	[D08-D05]
有效外掛程式介面版本	[D10-D09]
BSL 最大緩衝區大小	[D12-D11]
BSL 緩衝區起始位址	[D16-D13]
BCR 配置 ID	[D20-D17]
BSL 配置 ID	[D24- D21]

應用程式版本：

32 位元應用程式版本取自 BSL 配置中指定的位址

BSL 緩衝區大小：

RAM 資料緩衝區大小可用於儲存傳送/接收的 BSL 資料封包。

例

主機：80 01 00 19 B2 B8 96 49

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

在上述指定回應中，

命令直譯器版本 - 0x0100

組建 ID - 0x0100

應用程式版本 - 0x00000000

有效外掛程式介面版本 - 0x0001

BSL 最大緩衝區大小 - 0x06C0

BSL 緩衝區起始位址 - 0x20000160

BCR Config ID - 0x00000001

BSL Config ID - 0x00000001

4.4.5 獨立驗證

架構

Header	長度		RSP	資料	CRC32			
0x08	0x05	0x00	0x32	D1...D4	C1	C2	C3	C4

說明

該命令傳回 CRC 值以回應獨立驗證命令

資料

針對請求記憶體區域計算的 32 位元 CRC 值。D1...D4 中 D1 是 CRC32 中最小權重位元組。

4.5 開機載入程式安全性

4.5.1 受密碼保護的命令

可以直接或間接存取記憶體中資料的所有命令，都受密碼保護。密碼可在非主記憶體的 BSL 配置中配置。

傳送不正確的密碼時，裝置會在接下來的 2 秒內休眠，並且在此期間不接受任何命令，致使暴力式攻擊更用力。如果傳送錯誤密碼 3 次，BSL 即會採取安全警示動作。

4.5.1.1 安全警告

在 BSL 配置中，可以將安全警告配置為以下三種模式之一。

1. 原廠重設：原廠重設會清除完整的主要和非主要快閃記憶體。無論 BCR 配置中的原廠重設模式為何，都會完成清除。如果快閃記憶體的某些部分受到靜態寫入保護，BSL 將無法清除完整的快閃記憶體。
2. 停用開機載入程式：在 BCR 配置中停用開機載入程式並退出 BSL。除非更新 BCR 非主配置以啓用開機載入程式，否則無法進一步進入 BSL。如果為非主配置啓用靜態寫入保護，則 BSL 不會停用。
3. 什麼事都不做：不執行任何動作。

备注

對於原廠重設安全警告配置，如果清除了非主要，則開機載入程式將返回預設值。為防止出現這種情況，可對非主要進行寫入保護。如果非主要處於清除狀態，則裝置會被鎖定，無法再次存取裝置。

4.5.2 BSL 輸入項

只能透過啟動碼進入開機載入程式。

可在 BCR 配置中停用開機載入程式。

5 使用開機載入程式的程式流範例

本節說明 BSL 主機透過開機載入程式載入映像的典型序列。此範例序列會清除快閃記憶體，並在其中為新韌體編程。

- 開機載入程式應透過空白裝置偵測、Pin 式呼叫或應用程式請求來啟動。
- 叫用後，傳送連線命令以透過所需介面與 BSL 建立連線。
- 若使用 UART 介面，則可將傳輸速率變更為較高的值，以加速進一步通訊，且為選擇性。
- 若要完全清除快閃記憶體，請使用大量清除命令。只有在需要更新非主要快閃記憶體時，才可使用原廠重設命令。因為若清除非主要快閃記憶體，並鎖定未編程的裝置。
- 為韌體映像編程
- 對已編程的記憶體區域進行 CRC 驗證，檢查編程資料的正確性。這是選用步驟。
- 可以使用「啟動應用程式」命令啟動應用程式。

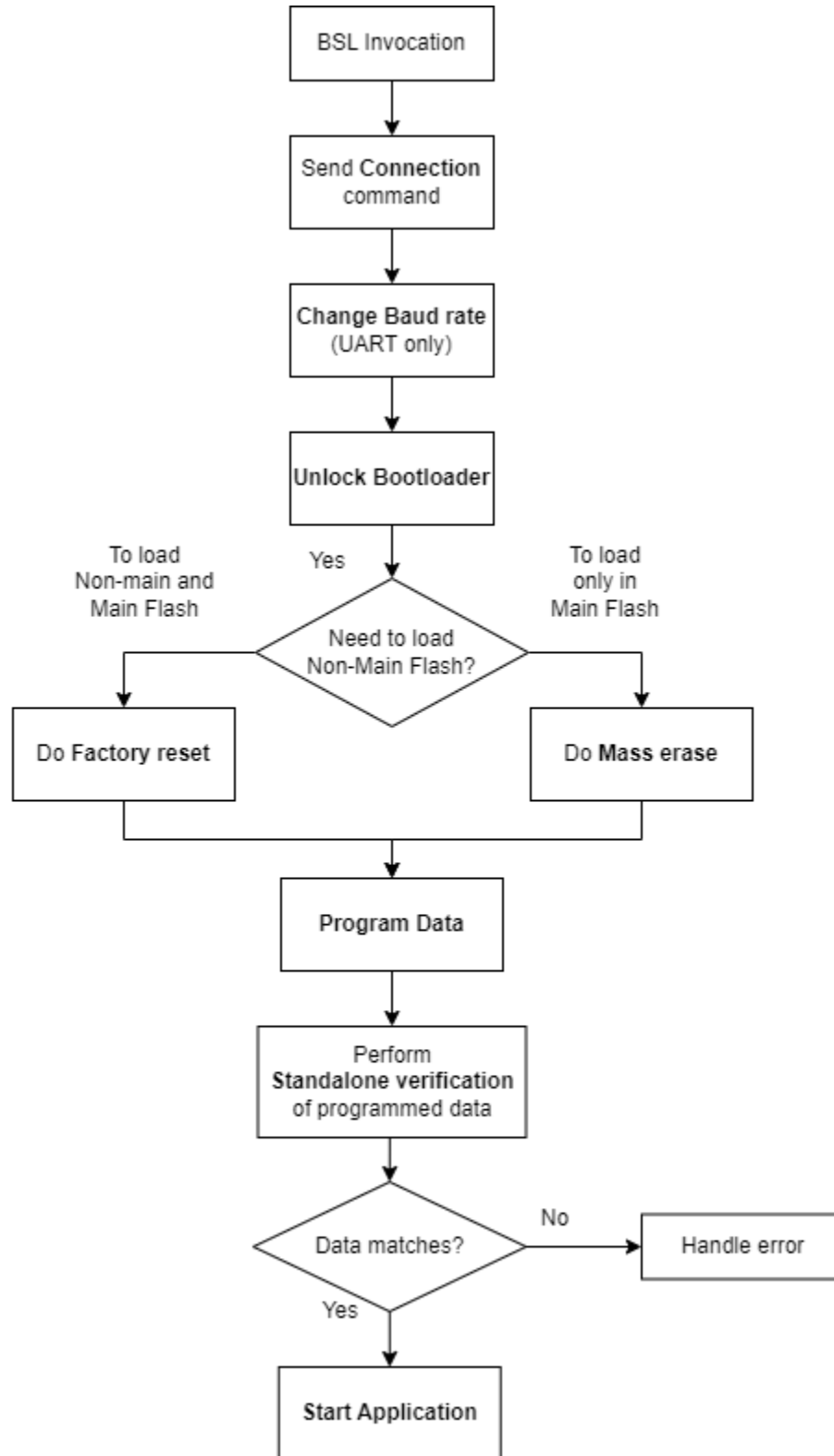


图 5-1. BSL 主機序列

6 二次開機載入程式

ROM 開機載入程式提供使用自訂開機載入程式的選項。為此，請在主快閃記憶體中載入自訂開機載入程式，並在非主快閃記憶體中的 BSL 配置中登錄該程式。下次叫用 ROM BSL 時，它將檢查二次開機載入程式配置欄位，並分流到該欄位，而且控制權不會返回 ROM BSL。

BCR 配置中的 BSL 模式配置也適用於二次 BSL。當此設定停用後，將不叫用二次 BSL。

載入自訂 BSL 的快閃記憶體區域應在 BCR 配置中受到寫入保護、以避免在開機載入期間意外清除。非主要寫入保護在二次開機載入程式應用實例中選用的。但在清除後，二次開機載入程式指標應適時還原。

备注

清除二次開機載入程式可能導致裝置鎖定。因此，必須寫入保護二次開機載入程式記憶體區域。

6.1 二次開機載入程式範例

提供二次開機載入程式的範例作為 SDK 範例的一部分，以供參考。本節提供更多相關詳細資訊。

說明

此範例二次開機載入程式支援對記憶體中的資料進行編程/驗證，其 BSL 協定格式與裝置中的主要 BSL (ROM BSL) 相同，可以用與 ROM BSL 相同的方式叫用。

其支援以下主要功能

- 程式資料
- 快閃記憶體清除
- 讀回資料
- CRC 驗證
- 啟動應用程式

它使用 UART 介面與主機通訊。

此範例負責二次開機載入程式的實作及登錄。因此，一旦將此映像載入到裝置中，就無法使用裝置中的主開機載入程式。只會啟用二次開機載入程式。若要將裝置恢復為使用主開機載入程式，則必須使用 SWD_Factory_Reset 命令 (透過偵錯子系統信箱進行原廠重設)。

範例使用

- 將 UART_RX 和 UART_TX 與 BSL 主機 (任何具有 UART 的微控制器) 連接。
- 編譯並載入範例。
- 使用 BSL 叫用針腳或其他任何叫用方法建立 BSL 叫用條件。
- 從主機傳送 **GetDeviceInfo** 命令。
- 裝置應以可用的版本資訊和 SRAM 緩衝區空間回應。
- 同樣的，傳送清除、編程和驗證命令以編程記憶體中的資料。

軟體檔案詳細資料

檔案名稱	詳細資訊
secondary_bsl.c	初始化 BSL 運作所需的周邊設備。從通訊介面接收命令封包，並將其傳遞到命令處理層。另外還負責在 BSL 配置記憶體中登錄二次開機載入程式。
bsl_ci.c	解譯命令封包、處理命令，並將回應傳回主機
bsl_ci.h	包含 BSL 命令和回應的定義。以及 bsl_ci.c 的函式宣告
bsl_uart.c	處理主機與 BSL 核心之間的通訊
bsl_uart.h	包含 BSL 確認和 bsl_uart.c 函式宣告的定義
ti_msp_dl_config.h	包含裝置專屬配置，例如 UART 針腳、使用周邊設備的基本位址等
boot_config.h	包含 BCR 和 BSL 配置結構
factory_config.c	執行功能以擷取原廠配置的裝置專屬資料，例如 SRAM 記憶體大小。
factory_config.h	包含 factory_config.c 的原廠配置結構和功能宣告

檔案名稱	詳細資訊
startup_mspm0x_ticlang	包含向量表、重設處理常式和其他處理常式的啟動檔
mspm0x.cmd	連結器命令檔，其指定二次開機載入程式映像應駐留在記憶體和 SRAM 區域中的記憶體區域。

自訂

此範例提供二次開機載入程式的參考實作。其可視需要自訂。BSL 核心層 (secondary_bsl.c, bsl_ci.c) 或介面層 (bsl_uart.c) 是完成自訂的主要位置。

遵循的步驟

- 根據需要修改程式碼
- 完成變更後，編譯程式碼
- 視情況修改 BCR 配置中的快閃記憶體寫入保護設定
- 計算 BCR 配置適用的 CRC 並儲存新的 CRC 值
- 再次編譯程式碼
- 載入自訂的 BSL 映像

7 介面外掛程式

ROM 開機載入程式提供一個選項，可以將自訂介面實作加入 ROM BSL 核心成為 Flash 外掛程式。其優點是提供自訂介面，而無需重新實作完整的 BSL 核心。

若使用此功能，可以：

- 將 ROM BSL 中不可用的新介面加入到介面清單中，以用於進行自動偵測。例如：SPI、CAN 等 (或)
- ROM 介面實作 (UART/I2C) 可被覆寫

若要使用此選項，應將 Flash 外掛程式映像載入到主快閃記憶體中，並在非主快閃記憶體的 BSL 配置中登錄。請參閱 MSPM0xx 技術參考手冊 - 配置記憶體 (非主要) 部分。

另外，載入外掛程式的快閃記憶體區域和非主配置記憶體應在 BCR 配置中獲得寫入保護，以防止 Flash 外掛程式和非主配置記憶體在開機載入過程中被清除。

备注

清除主快閃記憶體中的 Flash 外掛程式區域後，可能導致裝置鎖定。因此，需要寫入保護。

7.1 實作

外掛程式必須負責資料處理，以及和 BSL 主機通訊。此介面 Flash 外掛程式會透過以下 4 個 API 與 ROM BSL 核心耦合。

- 初始化
- 接收
- 傳輸
- 反初始化

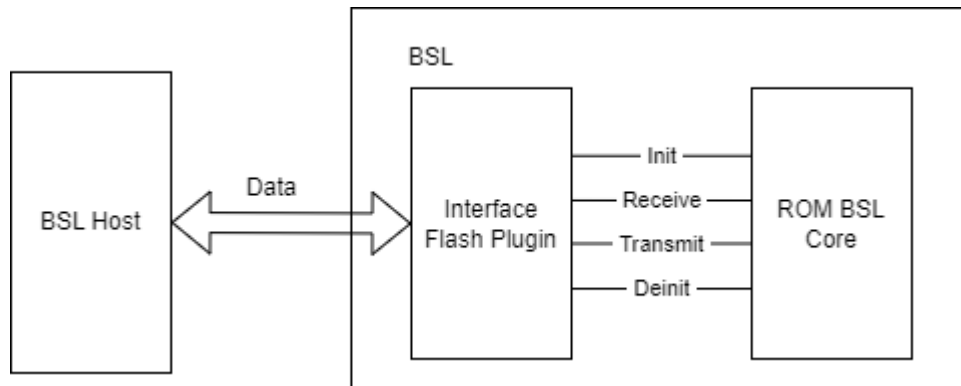


图 7-1. 外掛程式執行

Flash 外掛程式映像的構建方式應與其它任何應用程式相同，且將載入到主快閃記憶體中。但是，與應用程式不同的是，不會執行啟動程式碼或主函數。ROM BSL 只會透過在非主要記憶體 BSL 配置中登錄的勾點來呼叫上述 4 個 API。

7.1.1 初始化

原型

```
uint16_t init(uint8_t* buffer, uint16_t bufferSize);
```

buffer - 指向從 BSL 核心傳送的 SRAM 資料緩衝區的指標。

bufferSize - 可用作資料緩衝區的 SRAM 記憶體大小的 1/2。如果使用兩個緩衝區，1 個進行傳輸，1 個進行接收，一個緩衝區的位址將為「**buffer**」，另一個緩衝區的位址將為「**buffer + bufferSize**」

return - 傳回 16 位元外掛程式版本資訊

說明

Init 函數應負責配置所需介面，並初始化資料處理的參數。該函數還需負責初始化使用的其它任何全域變量，因為不會執行啟動程式碼將變量初始化。

7.1.2 接收

原型

uint32_t receive(void);

return - 傳回接收資料封包的 32 位元起始位址。資料封包的格式應與 ROM BSL 協定中所述的格式 [節 4](#) 相同。

說明

接收函數應負責從 BSL 主機讀取資料封包。只有在接收到完整的封包並檢查資料的正確性 (資料的 CRC 驗證) 時，才應與 BSL 核心共用封包。此外，每個資料封包只能共用該位址一次。如果 BSL 核心呼叫此函數，當沒有接收到資料封包，或正在接收資料封包時，應傳回「0」。

如果成功接收資料封包，沒有發生任何問題，則應如同 ROM BSL 外掛程式的作法向主機確認資料封包 (請參照 ROM BSL [確認](#))。如有任何問題，應透過 NACK 報告給主機，不應和 ROM BSL 核心共用封包。

7.1.3 傳輸

原型

uint8_t send(uint8_t* data, uint16_t length);

data - 指向要傳送到主機的 BSL 核心回應封包的指標。它的格式與 [節 4](#) 一節所述格式相同

length - BSL 核心回應封包的長度不包含 CRC 4 位元組的長度。

return - 如果傳輸成功，則傳回「1」。如果傳輸失敗，則傳回「0」。

說明

傳輸功能必須將 BSL 核心回應封包傳送至主機。此外也負責為封包計算並新增 CRC。傳輸 API 只能在完全傳輸資料後返回。

7.1.4 反初始化

原型

bool deinit(void);

Return

如果完成反初始化，則傳回 'True'。

說明

如果已註冊，此函數會重設介面配置，並取消註冊中斷處理常式。

7.1.5 重要注意事項

開發 Flash 外掛程式時、務必謹記重點。

1. 在開發 Flash 外掛程式時、必須注意以下幾件事項。
2. 載入 Flash 外掛程式的主要快閃記憶體區域應受靜態寫入保護
3. 所有全域變數都應透過 'Init' 函數進行初始化
4. 4 個外掛程式 API 的函數原型應按照 BSL 使用指南中的指定進行。
5. SRAM 記憶體使用量
 - a. VTOR - SRAM 的起始 (0x20000000)。如果使用中斷，則應將 VTOR 置於 SRAM 的起始位置，因為 ROM BSL 使用該位址空間
 - b. 堆疊起始位址 - 裝置中可用的 SRAM 記憶體結尾
 - c. 堆疊大小 - 不應超出 ROM BSL 堆疊大小
 - d. 資料部分 (.data, .bss)：當裝置中沒有登錄 Flash 外掛程式時，「取得裝置資訊」命令傳回的「BSL 緩衝區起始位址」應為資料部分的起始位址。
 - e. 資料段大小：應在 BSL 非主配置記憶體中配置資料部分 (.data, .bss) 消耗的大小。

7.2 Flash 外掛程式類型

Flash 外掛程式中使用的通訊介面類型，是在 BSL 配置記憶體中的 BSLPLUGINCFG.PLUGINTYPE 欄位中配置 (請參閱 節 8 中的技術參考手冊)。

如果外掛程式類型具有任何 ROM 介面類型 (UART 或 I2C)，Flash 外掛程式會使 ROM 實作無效。如果該類型的新類型在 ROM 中無法使用 (例如 SPI)，則該外掛程式將以新介面形式加入。

表 7-1. 外掛程式類型值

編號	介面	外掛程式類型
1	UART	0x1000
2	I2C	0x2000
3	其它任何介面	0xFFFF

如 图 7-2 中所示，檢查 Flash 外掛程式類型後，更新的介面清單會用於自動偵測。即使在 Flash 外掛程式註冊後，如果這些介面未被覆寫，即會使用 ROM 介面。

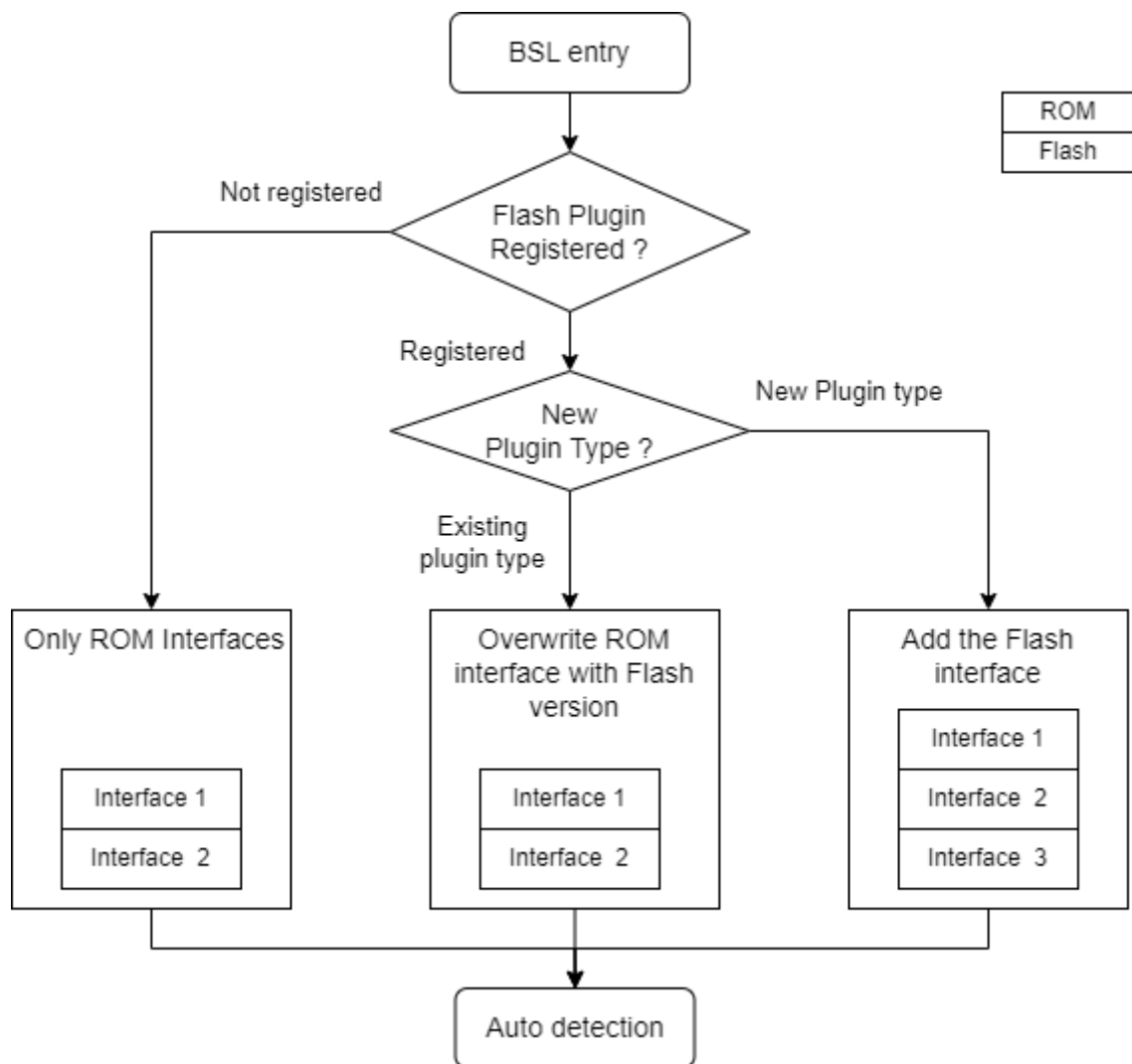


图 7-2. Flash 外掛程式

7.3 覆寫現有介面

此功能可讓使用者以 Flash 版本覆寫現有 ROM BSL 介面、UART 和 I2C。在這種情況下，只要載入 Flash 外掛程式，ROM 版本就會變成非作用中，而且永遠無法使用。若要將裝置恢復為使用 ROM 介面，請使用 SWD_Factory_Reset (透過偵錯子系統信箱進行原廠重設)。

7.3.1 UART 介面 Flash 外掛程式範例

使用 UART 通訊的範例 Flash 外掛程式，做為 SDK 範例 [SDK](#) 的一部分供參考。本節提供更多相關詳細資訊。

說明

UART 介面 Flash 外掛程式會透過下列 4 個 API 勾點處理 BSL 主機和 ROM BSL 之間的資料交易。

- BSL_PI_UART_init
- BSL_PI_UART_receive
- BSL_PI_UART_send
- BSL_PI_UART_deinit

UART Flash 外掛程式主要用於在需要時以自訂實作覆寫 ROM BSL UART 介面。

範例使用

- 將 UART_RX 和 UART_TX 與 BSL 主機 (任何具有 UART 介面的微控制器) 連線。
- 編譯並載入範例。
- 使用 BSL 叫用針腳或其他任何叫用方法建立 BSL 叫用條件。
- 從主機傳送 **Connection** 命令。成功時會收到 BSL 確認。
- 從主機傳送 **GetDeviceInfo** 命令。
- BSL 會回應 UART 介面 Flash 外掛程式版本資訊。
- 同樣的，傳送清除、編程和驗證命令以編程記憶體中的資料。

軟體檔案詳細資料

檔案名稱	詳細資訊
bsl_uart.c	處理主機與 BSL 核心之間的通訊。定義四個介面 API 初始化、接收、傳送和反初始化。
bsl_uart.h	包含 BSL 確認和 bsl_uart.c 函式宣告的定義
ti_msp_dl_config.h	包含裝置專屬配置，例如 UART 針腳、時鐘配置及其它。
boot_config.h	包含 BCR 和 BSL 配置結構
startup_mspm0x_ticlang	僅包含預設處理常式函式定義的啟動檔。不像一般啟動檔沒有中斷向量表或重設處理常式，因為這些特性從未在 Flash 外掛程式中使用，並且被移除，以減少記憶體消耗。
mspm0x.cmd	連結器命令檔案，其指定 Flash 外掛程式映像駐留在記憶體和 SRAM 區域中以進行運作的記憶體區域。

自訂

此範例提供 Flash 外掛程式適用的參考實作。可視需要自訂。介面 Flash 外掛程式 API 是主要的變更位置。

遵循的步驟：

- 視需要修改 Flash 外掛程式 API
- 完成變更後，編譯程式碼
- 如果 API 指標已更新，則計算 BSL 配置的 CRC
- 視情況修改 BCR 配置中的快閃記憶體寫入保護設定
- 計算 BCR 配置的 CRC
- 在 BCR 和 BSL 配置中儲存新的 CRC。
- 再次編譯程式碼。
- 載入 Flash 外掛程式映像

8 參考

1. [MSPM0 G 系列 80-MHz 微控制器技術參考手冊](#)
2. [MSPM0 L 系列 32-MHz 微控制器技術參考手冊](#)
3. MSPM0 SDK

9 修訂記錄

注：以前版本的页码可能与当前版本的页码不同

日期	修訂	備註
2023 年 2 月	*	初始版本

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2023, Texas Instruments Incorporated