参考資料

User's Guide

MSPM33 ブートローダ

TEXAS INSTRUMENTS

概要

MSPM33 ブートローダー (BSL とも呼ばれます) は、UART や I2C などの複数のシリアル インターフェイスを介して、デバイスのメモリ (フラッシュ) をプログラムおよび検証をするために使用されます。

目次

1 概要 2
1.1 BSL 機能の概要 2
1.2 用語 2
1.3 その他の資料 2
2 BSL アーキテクチャ 3
2.1 設計 3
2.2 BSL の起動 4
2.3 メモリ 5
2.4 メイン以外の BSL 構成 6
2.5 BSL 構成の変更 12
3 ブートローダのプロトコル 15
3.1 パケット フォーマット 15
3.2 BSL プロトコル 15
3.3 ブートローダのコア コマンド 16
3.4 ブートローダーのコア応答 23
3.5 ブートローダのセキュリティ 26
4 ブートローダーによるプログラムのフローのサンプル 27
5 改訂履歴 29

商標

すべての商標は、それぞれの所有者に帰属します。

Copyright © 2026 Texas Instruments Incorporated

1 概要

1.1 BSL 機能の概要

ブートストラップ ローダ (BSL) は、標準の UART または I2C シリアル インターフェイスを使用してデバイスのメモリをプログラムまたは検証できます。

シリアル インターフェイスでアクセス可能な BSL の主な機能には、次のものがあります。

- フラッシュ メモリのプログラミングと消去
- プログラミングを検証するため、コードまたはデータ領域 (最小領域サイズ 1KB) の 32 ビット CRC を返す
- コードまたはデータの読み出しを有効化する (デフォルトでは無効)
- メイン フラッシュへのポインタを使用して、ファームウェアのバージョン番号を返す
- ハードウェアの GPIO を起動するように指定する
- アクセスを常に 256 ビットのパスワードで保護
- ブルートフォース攻撃に抵抗するため、セキュリティ アラート処理を構成可能
- UART の ROM ベースのインターフェイス プラグイン、自動検出機能を備えた I2C
- 複数のオプションを備えた UART のボーレート構成
- UART と I2C の構成可能なインターフェイス ピン

1.2 用語

ブートローダー (BSL) — データをデバイスのメモリにロードするために使用されるブート ルーチン

ブートコード (BCR) - BOOSTRST の後で実行されるスタートアップ ルーチンで、アプリケーションを実行するためにデバイスを構成します

BCR 構成 — ブート コードのユーザー構成可能なすべてのパラメータを含む構成構造 (これはメイン以外のフラッシュ メモリに格納されています)

BSL 構成 — ブートローダーのユーザー構成可能なすべてのパラメータを含む構成構造 (これはメイン以外のフラッシュ メモリに格納されています)

1.3 その他の資料

1. テクニカル リファレンス マニュアル
    a. MSPM33 C シリーズ マイコン

Copyright © 2026 Texas Instruments Incorporated

2 BSL アーキテクチャ

2.1 設計

ブートローダーの有効な起動条件が検出されると、ブートコードによってブートローダーが起動されます。起動するのは、BCR の構成の BSL モード フィールドでブートローダーが有効のときのみです。

ブートローダーが開始されると、最初に「初期」フェーズを実行します。ここで、BSL 構成の初期チェックが行われ、デバイスはブートローダー動作用に構成されます。

次に、ブートローダーは「インターフェイス自動検出」フェーズに入ります。このフェーズでは、BSL は使用可能なすべての BSL インターフェイスを構成します (登録されている場合)。その後、BSL はすべてのインターフェイスを 1 つずつ使用してデータをポーリングします。いずれかのインターフェイスで有効な接続パケットが受信されると、そのインターフェイスがアクティブなインターフェイスとして以後の通信に使用され、他のすべてのインターフェイスは無効化されます。インターフェイス検出は 4 秒間実行されます。インターフェイスが検出されない場合、デバイスは STANDBY モードになります。

次に、BSL は「コマンド受信」フェーズに入ります。このフェーズでは、BSL はホストからのコマンドを無限ループで待機します。有効なコマンドが受信されると、そのコマンドが処理され、BSL コアからの応答がホストに返送されます。それから、BSL はループに戻り、以後のコマンドを待ちます。「アプリケーションの開始」コマンドが受信されると、ブートローダーはシステム リセットをトリガします。その後でブートコードが実行され、アプリケーションが起動されます。このフェーズのタイムアウトも 4 秒です。有効なコマンドが受信されない場合、ブートローダーはロックされ、スリープ モードに移行します。

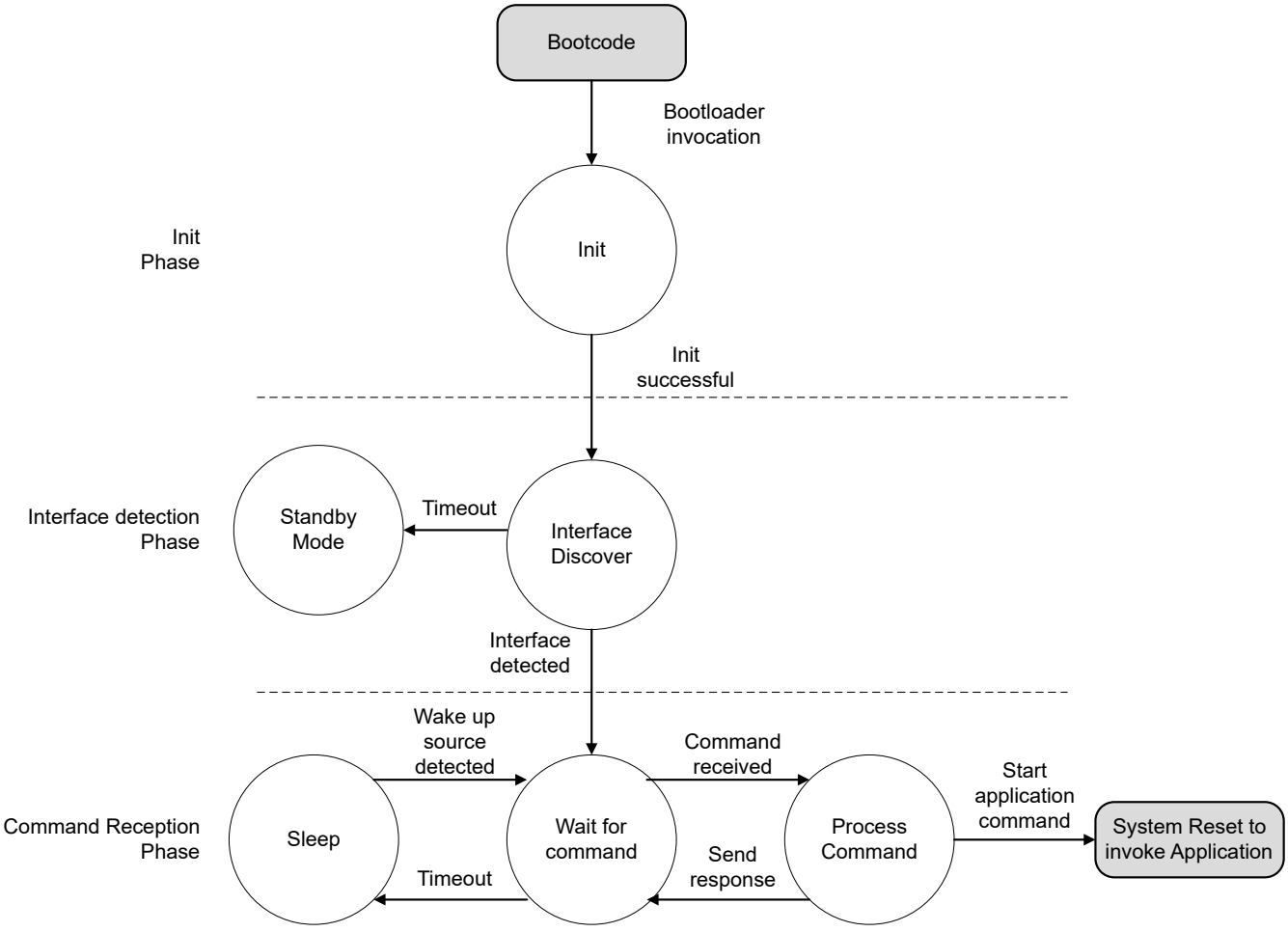


図 2-1. BSL アーキテクチャ

2.1.1 タイムアウト機能

アクティビティが検出されないとブートローダはタイムアウトし、低消費電力モードに移行して消費電力を低減します。

Copyright © 2026 Texas Instruments Incorporated

これは、次の 2 つのフェーズで実装されています。

1. インターフェイスの自動検出
2. コマンドの受信

2.1.1.1 インターフェイスの自動検出

インターフェイスの検出フェーズで、どのインターフェイスからも有効な接続コマンドが 4 秒間受信されない場合、ブートローダは STANDBY モードに移行します。

この状態を終了し、再度 BSL 起動条件を作り出してブートローダを使用するには、POR が必要です。

2.1.1.2 コマンドの受信

コマンド受信フェーズで、有効なコマンドが 4 秒間受信されない場合、ブートローダは SLEEP モードに移行します。SLEEP モードからデバイスをウェークアップするには、アクティブ インターフェイスでデータ トランザクションを実行する必要があります。

攻撃にさらされる場所を減らすため、ブートローダは SLEEP モードに移行する前にロックされます。このため、低消費電力モードからウェークアップした後で、256 ビットの BSL パスワードを送信してブートローダを再度ロック解除する必要があります (「ブートローダのロック解除コマンド」を参照)。

2.2 BSL の起動

BSL 起動の条件が満たされ、BCR 構成でブートローダーが有効の場合、ブートローダーはブート コードからのみ起動されます。

BCR の構成で高速ブート モードが有効のとき、ブートローダーはメールボックスのデバッグ コマンドとアプリケーション要求でのみブートローダーを起動できます。実行時間を短縮するため、他のチェックはスキップされます。

HS-FS (高度セキュリティ対応、フィールドでのセキュア化) 状態では、GPIO ベースの起動、アプリケーション要求、メールボックスのデバッグが有効になります。**HS-KP (高度セキュリティ対応、キー プロビジョニング)** 状態では、**GPIO ベースの起動**のみが有効化されます。キー プロビジョニング中にセキュリティを維持するため、アプリケーション要求とメールボックスのデバッグ起動は無効化されます。デバイスが **HS-SE (高度セキュリティ対応、セキュリティ強化)** に移行すると、ブートローダーは完全に無効化され、BSL 起動は可能ではないため、顧客のコードを最大限に保護できます。

2.2.1 アプリケーション要求

アプリケーションからブートローダーを起動するには、RESETLEVEL を BOOTLOADERENTRY に設定し、RESETCMD レジスタからリセットをトリガします。このシーケンスによってシステムがリセットされ、ブートコードが実行されてブートローダーが呼び出されます。

システム リセットが発行されるので、アプリケーションの終了時にすべてのペリフェラル構成がリセットされます。

2.2.2 GPIO ベースの起動

BSL の起動に使用される GPIO は、メイン以外のフラッシュの BSL 構成で設定できます。

新規デバイスは、TI によってデフォルト ピンが BSL の構成でプログラムされています。

GPIO ピン ベースの起動は、BCR の構成で無効化できます。デフォルトではイネーブルです。

POR の前に GPIO をアサートし、POR の後で少なくとも T_start ミリ秒の間、その状態を維持します。その後で、GPIO ピンの状態をアサート解除できます。

Copyright © 2026 Texas Instruments Incorporated

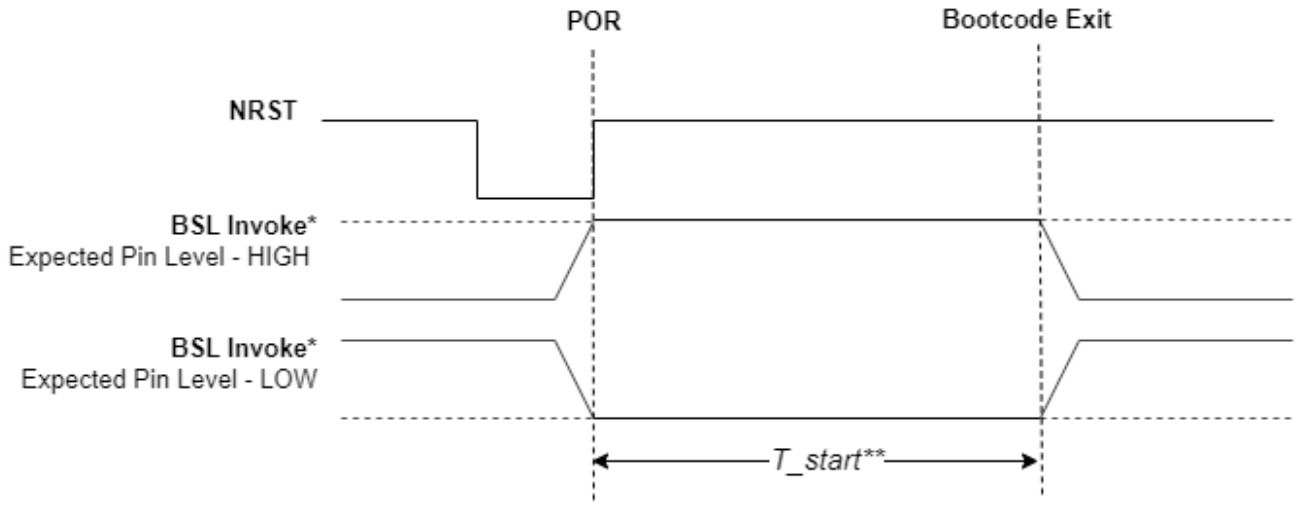


図 2-2. GPIO からの起動

** - 「BSL 起動」および「期待されるピンのレベル」として使用される GPIO ピンは、BSL の構成で設定できます*

** - T_start は、デバイス固有のデータシートに記載されているコールド ブートのスタートアップ時間を表します*

注

ピン ベースの BSL 起動がイネーブルのとき、構成された GPIO ピンは High または Low にします。フローティング状態のままにしておくと、予期しない BSL 実行が発生する可能性があります。

2.2.3 メールボックスのデバッグ コマンド

デバッグ インターフェイスが使用可能な場合、デバッグ サブシステム メールボックス (DSSM) 経由でブートローダー起動コマンドを送信できます。

DSSM コマンドの使用方法の詳細については、MSPM33C_Technical_Reference_Manual の「DSSM コマンド」セクションを参照してください。

2.2.4 その他の BSL 起動方式

2.2.4.1 プリブート アプリケーションの検証

BCR 構成において、アプリケーション CRC 検証が有効になっている場合、ブートコードは指定されたアプリケーション メモリ範囲の CRC を検証します。CRC が正しくない場合、ブートローダーが起動します。この場合、ブートローダーは有効のときのみ起動します。無効の場合、ブートコードは致命的なエラーをログに記録し、ブート障害を発生させます。

2.2.4.2 ブランク デバイス処理

従来型のデバイスとは異なり、ブランク デバイスが検出されたとき (空のフラッシュ メモリ)、ブートコードはブートローダーを自動的に起動しません。代わりに、while (1) ループを SRAM にコピーし、10 秒以内にデバッガが接続されるまで待機します。この時間フレーム内にデバッガ接続が確立されない場合、デバイスは低消費電力モードに移行します。このアプローチは、ブランク デバイスを使用する場合に、より柔軟な開発環境を実現します。

2.3 メモリ

2.3.1 SRAM メモリの使用法

MSPM33 BSL は、デバイスで利用可能な SRAM サイズに基づいて SRAM メモリを動的に割り当てます。SRAM メモリのレイアウトでは、ブートローダーの動作に使用されるメモリについて説明します。

SRAM メモリのレイアウトでは、ブートローダーの動作に使用されるメモリについて説明します。

Copyright © 2026 Texas Instruments Incorporated

- データとスタックのセクション — 自体の動作のために BSL によって使用されます。ブートローダーを終了すると、SRAM のこれらのセクションはクリアされます。
- 可変バッファ スペース - BSL の通信中に送受信されるデータ パケットを格納するために使用されるバッファ空間

SRAM メモリの割り当ては以下の構造に従います。

1. BSL バッファの開始アドレス：BootROM 動作に使用される SRAM データ セクションの終了アドレスとして計算され、次の 8 バイト境界にアラインされます。
2. BSL バッファの終了アドレス：RAM 終了アドレスからスタック サイズを引いた値として計算されます。
3. 使用可能なバッファ空間：BSL バッファの開始アドレスと BSL バッファの終了アドレスの間の空間は、次の 2 つの等しい部分に分割されます。
   a. RX バッファ：データ パケットの受信に使用され、BSL バッファの開始アドレスから開始されます
   b. TX バッファ：データ パケットの送信に使用され、BSL バッファの開始アドレス + バッファ サイズから開始されます

BSL プロトコルでは長さの定義に 2 バイトを使用するため、最大バッファ サイズは 32KB (0x7FFF バイト) に制限されます。MSPM33 には 256KB の SRAM が搭載されていますが、各バッファ (RX および TX) には 32KB のみが使用されます。

ホストによる読み取り/書き込みアクセスが可能な SRAM メモリは、BSL バッファの開始アドレスから [SRAM の終了アドレス — スタック サイズ] までです。ここで、SRAM の終了アドレスは、各デバイスで利用可能な SRAM メモリによって決定されます。可変バッファ領域と同じ SRAM 領域が共有されるため、SRAM の書き込み / 読み取り動作中に上書きされる可能性があります。

2.4 メイン以外の BSL 構成

メイン フラッシュ メモリ以外の BSL の構成では、BSL で使用される特定のパラメータをユーザーがカスタマイズできます。この構成は、オフセット 0x1C00 から始まるメイン以外のフラッシュ メモリに保存されます。

2.4.1 BSL の構成 ID

BSL_CONFIG_ID は、これをブートローダーの構成構造体として識別する、事前決定済みシグネチャ ID です。この 32 ビット フィールドは、BSL 構成の識別子として機能します。

使用方法：このフィールドでは、構造を BSL 構成構造として識別します。これは、ユーザーが変更してはならない固定値です。

デバイス情報の取得で返される値：BSL_CONFIG_ID は、デバイス情報の取得コマンド (CMD_GET_IDENTITY、0x19) への応答に含まれています。これにより、ホストは BSL 構成バージョンまたはタイプを識別できます。

2.4.2 BSL インターフェイス ピン (BLINTERFACE_PINS)

BLINTERFACE_PINS 構造は、BSL 通信インターフェイス (UART および I2C) の物理ピンとピン機能マルチプレクサの選択を設定します。この構成により、BSL はホストとの通信に特定のピンを使用できるようになります。

UART インターフェイス ピン (0x80101C04)

1. UART 受信ピン番号 (ビット 7...0)
   a. UART 経由でデータを受信するための物理ピン番号
2. UART 受信ピンの機能 (ビット 15...8)
   a. 受信ピンの機能マルチプレクサの選択
3. UART 送信ピン番号 (ビット 23...16)
   a. UART 経由でデータを送信するための物理ピン番号
4. UART 送信ピンの機能 (ビット 31...24)
   a. 送信ピンの機能マルチプレクサの選択

Copyright © 2026 Texas Instruments Incorporated

I2C インターフェイス ピン (0x80101C08)

1. I2C データ ピン番号 (ビット 7...0)
   a. I2C データ ライン (SDA) の物理ピン番号
2. I2C データ ピンの機能 (ビット 15...8)
   a. データ ピンの機能マルチプレクサの選択
3. I2C クロック ピン番号 (ビット 23...16)
   a. I2C クロック ライン (SCL) の物理ピン番号
4. I2C クロック ピンの機能 (ビット 31...24)
   a. クロック ピンの機能マルチプレクサの選択

予約済みピン情報 (0x80101C0C)

2.4.3 BSL 起動ピン構成 (BSLPIN_INVOKE)

BSLPIN_INVOKE 構造は、BSL 起動のトリガに使用する GPIO ピンを構成します。これにより、デバイスの起動時に特定のピンを特定の状態に設定することで、BSL を起動できます。

1. ピン構成およびトリガ レベル (ビット 7...0)
   a. BSL 起動のピン構成と想定されるトリガ レベルを定義します
   b. ビット フィールド
      i. PINCM インデックス (ビット 6...0):ピン構成モード インデックスを指定します
      ii. トリガ レベル (ビット 7):BSL 起動をトリガするピンの状態を指定します
         1. 0 = BSL は、ピンが LOW のときに起動されます
         2. 1 = BSL は、ピンが HIGH のときに起動されます
      iii. デフォルト値:0xA8
         1. PINCM インデックス = 0x28 (40)
         2. トリガ レベル = 1 (HIGH)

2. ピン選択 (ビット 15...8)
   a. BSL 起動に使用される物理ピンを定義します
   b. ビット フィールド
      i. ピン番号 (ビット 4...0):選択したポート内のピン番号を指定します
      ii. ポート インデックス (ビット 6...5):ピンが属する GPIO ポートを指定します
      iii. 予約済み (ビット 7):現在の実装では使用されません
   c. デフォルト値:0x12
      i. ピン番号 = 0x12 (18)
      ii. ポート インデックス = 0 (ポート A)

BSL 起動ピン構成により、デバイスの起動時に、特定の GPIO ピンが特定の状態に設定されたとき、デバイスは BSL モードに移行できます。デフォルトでは、デバイスの起動時にポート A のピン 18 が HIGH に設定されると、BSL が起動します。

構成済みピンを使用して BSL を起動するには、次の手順に従います。

1. 指定されたピン (デフォルト:ポート A のピン 18) を、指定された状態 (デフォルト:HIGH) に設定します
2. デバイスの電源をオンにするか、リセットを実行します
3. パワーオン リセット後、T_start ミリ秒以上ピンの状態を維持します
4. ピンの状態は、T_start ミリ秒後にデアサートできます

Copyright © 2026 Texas Instruments Incorporated

注

ピン ベースの BSL 起動が有効になっている場合、構成された GPIO ピンは、構成に応じて high または low にプルする必要があります。フローティング状態のままにしておくと、予期しない BSL 実行が発生する可能性があります。

2.4.4 メモリ読み出し構成

READOUT フィールドは、BSL がメモリ読み出し動作を許可するかどうかを制御します。有効の場合、メモリ読み戻しコマンドを使用して、フラッシュおよび SRAM メモリの内容を読み取ることができます。無効化すると、メモリ読み戻しコマンドはエラーを返します。

デフォルト値:BL_CFG_READBACK_EN (0xAABB)

有効な値:
1. BL_CFG_READBACK_EN (0xAABB):メモリの読み出しを有効化します
2. その他すべての値:メモリ読み出しが無効です

デフォルトでは、メモリ読み出しはセキュリティ上の理由から無効になっています。この機能は、開発中の環境の場合、またはメモリの読み出しが特に必要な場合にのみ有効化する必要があります。

メモリ読み出しを有効化しても、メモリ読み戻しコマンドを使用するには、正しいパスワードを使用して BSL のロックを解除する必要があります。これにより、メモリ内容への不正なアクセスを防止するためのセキュリティ層が強化されます。

2.4.5 BSL パスワード

メモリ アドレス:0x80101C14 - 0x80101C30

PASSWORD フィールドには、BSL 認証に使用されるパスワードの SHA-256 ハッシュ (32 バイト) が格納されます。ブートローダーのロックを解除し、保護された BSL コマンドにアクセスするには、このパスワードが必要です。

デフォルト値:すべての 0xFF バイトの文字列のハッシュ
1. ワード 0 (0x80101C14): 0x761396AF
2. ワード 1 (0x80101C18): 0x5F63720F
3. ワード 2 (0x80101C1C): 0x5A4AB4BD
4. ワード 3 (0x80101C20): 0x9FC3630A
5. ワード 4 (0x80101C24): 0xF930AF12
6. ワード 5 (0x80101C28): 0x5CEEA650
7. ワード 6 (0x80101C2C): 0x88E11B97
8. ワード 7 (0x80101C30): 0x51409CE8

使用方法

BSL パスワードは、デバイス メモリの変更や機密情報の読み取りを可能にする重要な BSL コマンドへのアクセスを保護するために使用されます。BSL が起動されると、ロックされた状態で起動します。BSL をロック解除し、保護されたコマンドにアクセスするには、ホストは正しいパスワードを使用してブートローダーのロック解除コマンド (CMD_UNLOCK_BSL、0x21) を送信する必要があります。

パスワード認証プロセスは次のように動作します。
1. ホストが、32 バイトのパスワードを使用してブートローダーのロック解除コマンドを送信します。
2. BSL が、受信したパスワードの SHA-256 ハッシュを計算します。
3. BSL が、計算されたハッシュを、BSL 構成に保存されているパスワード ハッシュと比較します。
4. ハッシュが一致した場合、BSL はロック解除され、保護されたコマンドが使用できるようになります。
5. ハッシュが一致しない場合、BSL はロックされたままとなり、BSL_PASSWORD_ERROR を返します。

BSL には、パスワード攻撃から保護するための以下のセキュリティ機能が含まれています。
1. **スリープ遅延**:誤ったパスワードが送信されると、デバイスは 2 秒間スリープ モードに移行し、この期間中はいかなるコマンドも受け付けません。これにより、ブルートフォース攻撃により時間がかかることになります。

Copyright © 2026 Texas Instruments Incorporated

2. **セキュリティ アラート**: 誤ったパスワードが 3 回送信されると、SECURITY_ALERT_LEVEL 構成に基づいてセキュリティ アラート アクションが実行されます。これには以下が含まれます。
   a. 工場出荷時リセット: すべてのフラッシュ メモリを消去します
   b. ブートローダーの無効化: ブートローダーを永久的に無効化します
   c. アクションなし: 何も実行されません
3. **パスワードのバックアップ**: 工場出荷時リセットを実行する前に、BSL はパスワードを SRAM にバックアップします。これにより、メイン以外のフラッシュのパスワード フィールドが消去されても、工場出荷時リセットの後に BSL のロックを解除できます。

BSL パスワードの変更

BSL のパスワードを変更する方法は複数あります

1. **BSL コマンドの使用**
   a. 現在のパスワードを使用して BSL のロックを解除します。
   b. 新しいパスワードの SHA-256 ハッシュを計算します。
   c. BSL 構成の PASSWORD フィールドを新しいハッシュで更新します。
   d. BSL 構成の新しい CRC を計算します。
   e. データのプログラム コマンドを使用して、更新された BSL 構成をメイン以外のフラッシュにプログラムします。
2. **SysConfig と Code Composer Studio (CCS) の使用** SysConfig ツールは、メイン以外の完全な構成を生成することができるメイン以外の構成ツールを提供します。メイン以外のフラッシュ領域が保護されていない場合は、Code Composer Studio のフラッシュ ローダを使用して BSL パスワードを直接変更できます。
   a. SysConfig ツールを CCS で、またはスタンドアロン アプリケーションとして開きます。
   b. メイン以外のコンフィギュレータ セクションに移動します。
   c. 新しいパスワードのハッシュを使用する PASSWORD フィールドを含め、すべての BSL 設定を構成します。
   d. メイン以外の構成バイナリを生成します。
   e. CCS フラッシュ ローダまたは BSL コマンドを使用して、生成されたバイナリを、メイン以外のフラッシュ領域にプログラムします。

注

BSL パスワードを忘れた場合は、DSSM の工場出荷時リセットを実行して、デフォルトのパスワードを復元する必要があります。ただし、これにより、アプリケーション コードや構成データを含め、すべてのフラッシュ メモリが消去されます。あるいは、メイン以外のフラッシュ領域が保護されていない場合は、CCS と フラッシュ ローダを使用して、新しいパスワードで BSL 構成を再プログラムすることができます。

2.4.6 アプリケーション リビジョン ポインタ

メモリ アドレス: 0x80101C34

APP_REV_POINTER フィールドには、メイン フラッシュ メモリに格納されているアプリケーション バージョン情報へのポインタが含まれます。これにより、BSL はホストから要求されたときに、アプリケーション バージョンを取得して報告できます。

デフォルト値: 0xFFFFFFFF (アプリケーション バージョン情報が利用できないことを示します)

使い方:

APP_REV_POINTER は、デバイス情報の取得コマンド (CMD_GET_IDENTITY、0x19) の実行時に、BSL によってアプリケーション バージョン情報の場所を見つけて取得するために使用されます。これにより、ホストは、アプリケーションをロードして実行することなく、デバイスに現在インストールされているアプリケーション バージョンを照会できます。

デバイス情報の取得コマンドを実行すると、BSL は次のチェックを実行します。

1. APP_REV_POINTER に有効なフラッシュ アドレスが含まれていることを確認します

Copyright © 2026 Texas Instruments Incorporated

2. アドレスが空白でないかどうかをチェックします (0xFFFFFFFF ではないこと)
3. アドレスが 64 ビット アラインされていることを確認します
4. すべてのチェックに合格すると、指定されたアドレスからアプリケーション バージョンとして 8 バイト (64 ビット) の値を読み出します

これらのチェックのいずれかが失敗した場合、BSL はアプリケーション バージョンとして 0 を返します。

実装要件:

APP_REV_POINTER フィールドは、顧客が管理するもので、次の実装手順が必要です。

1. **アプリケーション リンカ ファイルでセクションを割り当て:**顧客は、アプリケーション バージョン情報を保存するために、アプリケーション リンカ ファイル内の専用セクションを割り当てる必要があります。このセクションは 64 ビット アラインされており、メイン フラッシュ メモリ内に配置されている必要があります。
2. **アプリケーション バージョンを定義:**アプリケーション バージョンは、アプリケーション コードで 64 ビット (8 バイト) の値として定義する必要があります。これは、メジャー、マイナー、パッチのバージョン番号を含む構造化値、または 64 ビット内に収まるその他の形式です。
3. **割り当てられたセクションにバージョンを配置:**アプリケーションのビルド プロセス中に、割り当てられたセクションにアプリケーション バージョン値を配置する必要があります。
4. **APP_REV_POINTER を更新:**割り当てられたセクションのアドレスは、BSL 構成の APP_REV_POINTER フィールドに格納する必要があります。これは、「BSL 構成の変更」セクションに記載されている方法を使用して実行できます。

注

APP_REV_POINTER はオプション機能です。使用しない場合は、デフォルト値の 0xFFFFFFFF のままにします。これは、アプリケーション バージョン情報が利用できないことを示します。

2.4.7 セキュリティ アラート レベル

メモリ アドレス:0x80101C38

ビット:15...0

説明:SECURITY_ALERT_LEVEL フィールドは、複数回失敗したパスワードなど、セキュリティ違反が検出されたときに実行するアクションを定義します。

デフォルト値:0xAABB (工場出荷時リセット)

有効な値:

1. BL_CFG_SECURITY_FACTORY_RESET (0xAABB):セキュリティ違反が発生した場合に、工場出荷時リセットを実行します
2. BL_CFG_SECURITY_DISABLE_BSL (0xCCDD):セキュリティ違反が発生した場合に、ブートローダーを無効化します
3. その他のすべての値:セキュリティ違反が発生した場合に、何も行われません

注

SECURITY_ALERT_LEVEL は、誤ったパスワードを 3 回連続して試行した後にのみチェックされます。1 回目および 2 回目の誤った試行に対して、この設定に関係なく、デバイスは 2 秒間スリープ モードに移行します。

2.4.8 UART ボーレート

メモリ アドレス:0x80101C38

ビット:31..16

UART_BAUD_RATE フィールドは、BSL の UART 通信に使用されるデフォルトのボーレートを指定します。これにより、BSL 動作に UART インターフェイスを使用するときの通信速度が決まります。

Copyright © 2026 Texas Instruments Incorporated

有効な値:
1. BL_CFG_UART_BAUDRATE_4800 (0x1):4800 ビット / 秒
2. BL_CFG_UART_BAUDRATE_9600 (0x2):9600 ビット / 秒
3. BL_CFG_UART_BAUDRATE_19200 (0x3):19200 ビット / 秒
4. BL_CFG_UART_BAUDRATE_38400 (0x4):38400 ビット / 秒
5. BL_CFG_UART_BAUDRATE_57600 (0x5):57600 ビット / 秒
6. BL_CFG_UART_BAUDRATE_115200 (0x6):115200 ビット / 秒
7. BL_CFG_UART_BAUDRATE_1000000 (0x7):1000000 ビット / 秒
8. BL_CFG_UART_BAUDRATE_2000000 (0x8):2000000 ビット / 秒
9. BL_CFG_UART_BAUDRATE_3000000 (0x9):3000000 ビット / 秒

ボーレートの変更:

UART ボーレートを変更する方法は 2 つあります。
1. 永続的な変更:「BSL 構成の変更」セクションで説明されている方法を使用して、BSL 構成の UART_BAUD_RATE フィールドを変更します。これにより、BSL が起動するときに使用されるデフォルトのボーレートが変更されます。
2. 実行時の変更:BSL セッション中に、ボーレートの変更コマンド (CMD_CHANGE_BAUD_RATE) を使用して、ボーレートを一時的に変更します。この変更は、現在の BSL セッションに対してのみ有効で、デバイスがリセットされたときにデフォルト値に戻ります。

注

BSL 通信に UART インターフェイスを使用する場合、ホストとデバイスの両方が同じボーレートを使用するように構成する必要があります。ボーレートの変更コマンドを使用してボーレートを変更する場合、ホストも一致するようにボーレートを変更する必要があります。

2.4.9 I2C スレーブ アドレス

メモリ アドレス:0x80101C3C

I2C_SLAVE_ADDR フィールドは、I2C モードで動作しているときに BSL が使用するスレーブ アドレスを指定します。外部デバイスはこのアドレスを使用して、I2C インターフェイス経由で BSL と通信する必要があります。

ビット:15...0

デフォルト値:0x48

I2C_SLAVE_ADDR フィールドは、I2C インターフェイス経由の通信時に BSL が使用するスレーブ アドレスを設定します。BSL を起動して、I2C インターフェイスを選択すると、このスレーブ アドレスに応答するように I2C ペリフェラルが構成されます。

2.4.10 構成 CRC

メモリ アドレス:0x80101C4C

CRC フィールドには、BSL 構成構造全体 (CRC フィールド自体を除く) にわたって計算された 32 ビット巡回冗長検査値が含まれています。この値は、構成データの整合性を検証するために使用されます。

デフォルト値:0xB4808AA4 (デフォルト構成値に基づいて計算)

有効な値:BSL 構成の内容に基づいて計算された値

CRC フィールドには、いくつかの重要な目的があります。
1. データ整合性検証:BSL 構成データが破損していないこと、また誤って変更されていないことを確認します。
2. 構成の検証:BSL 構成構造が有効で完了していることを確認します。
3. エラーに対する保護:予期しない動作の原因となり得る誤った構成値を BSL が使用することを防止します。

BSL 構成のフィールドが変更されるたびに、CRC を再計算する必要があります。これには、以下の注意点があります。

Copyright © 2026 Texas Instruments Incorporated

1. 初期構成:BSL 構成が最初に作成またはプログラムされたとき。
2. 構成の更新:構成のフィールドが変更されたとき。
3. 工場出荷時リセット:工場出荷時リセットの後、デフォルト構成について新しい CRC が計算されます。

BSL 構成で CRC が正しくないと、致命的なエラーが発生して回復不可能なため、永続的な損失が発生します。

注

SysConfig や CCS などのツールを使用して BSL 構成を変更する場合、通常は CRC が自動的に計算されて更新されます。ただし、BSL コマンドまたは他の方法を使用して構成を手動で変更する場合は、デバイスの永続的な損傷を避けるため、CRC が正しく再計算されていることを確認する必要があります。

2.5 BSL 構成の変更

メイン以外のフラッシュ メモリに保存された BSL 構成は、いくつかの方法で変更できます。このセクションでは、BSL 構成パラメータの更新に使用可能なアプローチについて説明します。

2.5.1 リファレンス

前提条件:

BCR 構成で、工場出荷時リセットを有効にしておく必要があります (「イネーブル」または「イネーブルでパスワードが必要」)

手順:

1. サポートされている呼び出し方法のいずれか (GPIO、アプリケーション要求、メールボックスのデバッグ、または SFI) を使用してブートローダーを起動します
2. 接続コマンド (CMD_CONNECTION、0x12) を使用して接続を確立します
3. 現在のパスワードでブートローダーのロック解除コマンド (CMD_UNLOCK_BSL、0x21) を使用して、BSL のロックを解除します
4. 工場出荷時リセット コマンド (CMD_FACTORY_RESET、0x30) を実行して、メイン以外のフラッシュ構成を消去します
   a. これにより、メイン フラッシュ (アプリケーション メモリ) とメイン以外のフラッシュ (構成メモリ) の両方が消去されます
   b. BCR 構成で必要な場合は、リセット パスワードを工場出荷時に提供します (デフォルト:すべて 0xFF)
   c. 工場出荷時リセット後、BSL パスワードはデフォルト (すべての 0xFF のハッシュ) に戻ります
5. 新しい BSL 構成構造を準備し、BSL 構成用に CRC32 を計算します。
   a. 構成構造全体の CRC32-ISO3309 を計算します (CRC フィールド自体を除く)
   b. 初期シード 0xFFFFFFFF でビット反転の構成を使用します
   c. 構成構造のサイズ:76 バイト (0x80101C00 ～ 0x80101C4B)
   d. 計算された CRC32 の値を CRC フィールド (0x80101C4C) に格納します
6. データのプログラム コマンド (CMD_PROGRAM_DATA、0x20) を使用して、新しい BSL 構成をメイン以外のフラッシュにプログラムします。
   a. 開始アドレス:0x80101C00
   b. データ:完全な BSL 構成構造 (CRC を含む合計 80 バイト)
   c. データのプログラム コマンドで要求されている通り、データが 16 バイト境界にアラインされていることを確認します
7. デバイスをリセットして、新しい構成を適用します
8. 注:構成フィールドが変更されるたびに、CRC フィールドを再計算する必要があります。• CRC が間違っていると、致命的なエラーが発生し、デバイスが永久にロックアウトされます。• BCR 構成で工場出荷時リセットが有効になっていない場合、メイン フラッシュ以外のプログラミングは失敗します。• 個々のフィールドだけでなく、構成構造全体がプログラムされていることを確認します

Copyright © 2026 Texas Instruments Incorporated

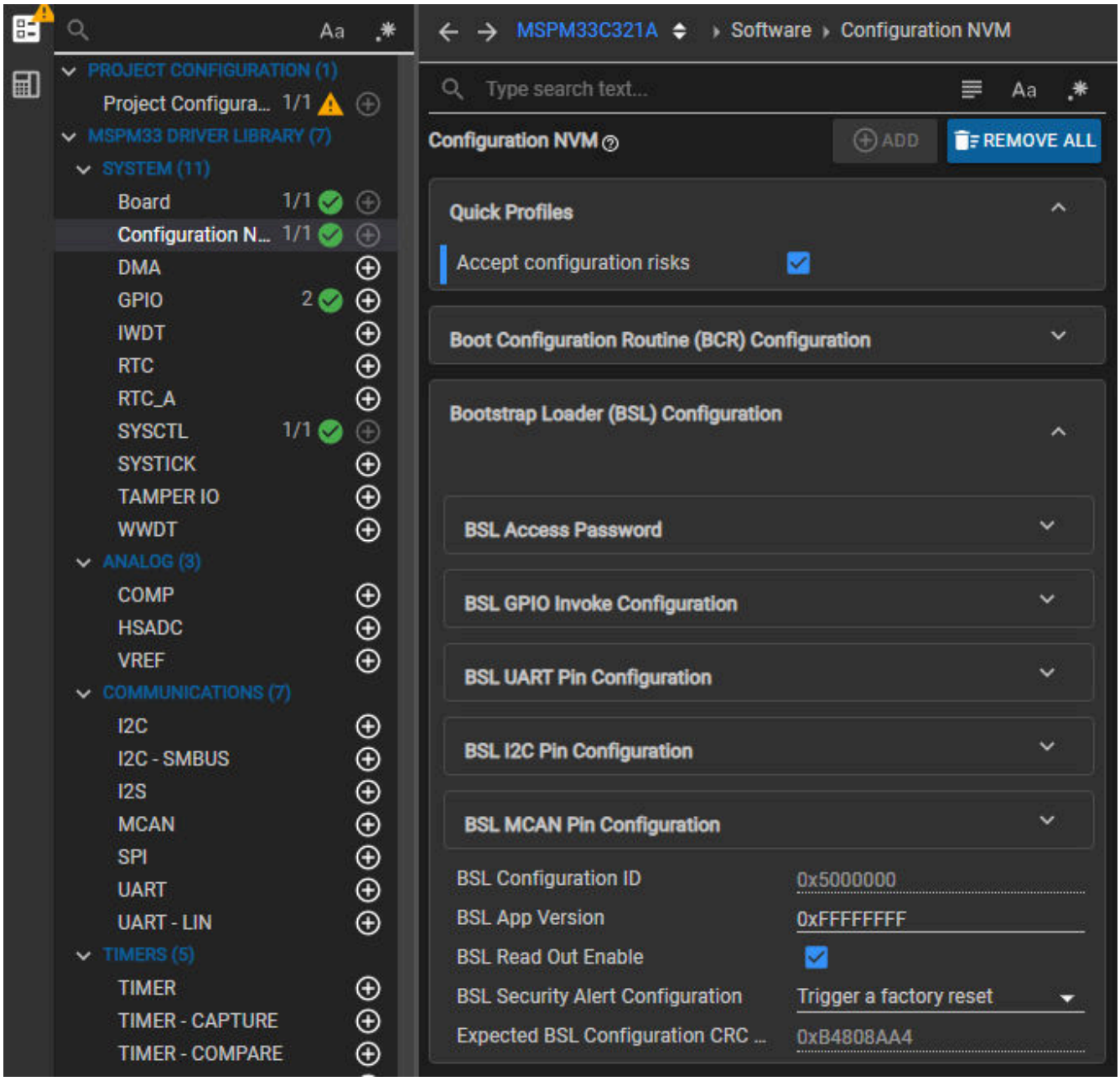
注

- 構成フィールドが変更されるたびに、CRC フィールドを再計算する必要があります
- CRC が正しくないと、致命的なエラーと永続的なデバイスのロックアウトが発生します
- BCR 構成で工場出荷時リセットが有効でない場合、メイン以外のフラッシュ プログラミングは失敗します
- 個々のフィールドだけでなく、構成構造全体がプログラムされていることを確認します

2.5.2 SysConfig と Code Composer Studio (CCS) の使用

SysConfig ツールは、BSL パラメータを構成するための以下のグラフィカル インターフェイスを提供します。

1. SysConfig を Code Composer Studio で、またはスタンドアロン アプリケーションとして開きます



2. メイン以外のコンフィギュレータ セクションに移動します
3. 目的の BSL パラメータを構成します。
   a. インターフェイス ピン (UART、I2C)

Copyright © 2026 Texas Instruments Incorporated

b. BSL 起動ピン構成
c. メモリ読み出しのイネーブル / ディスエーブル
d. BSL パスワード: 事前計算済みの 32 バイト SHA-256 ハッシュを入力します (平文パスワードではありません)
e. アプリケーション リビジョン ポインタ
f. UART ボーレート
g. I2C スレーブ アドレス

4. このツールは、すべての構成済みパラメータに基づいて、CRC フィールドを自動的に計算して更新します
5. メイン以外の構成バイナリ ファイルを生成します
6. 生成されたバイナリをメイン以外のフラッシュにプログラムするには、次のいずれかの方法を使用します。
   a. CCS フラッシュ ローダ (メイン以外が書き込み保護されておらず、デバッグ アクセスが利用可能な場合)
   b. BSL データのプログラム コマンド (BCR 構成で工場出荷時リセットが有効の場合)
   c. デバッグ インターフェイス プログラミング ツール

注

- SysConfig では、SHA-256 パスワード ハッシュが計算されません。事前に計算されているハッシュを指定する必要があります
- このツールは、構成構造の整合性を考慮して CRC32 のみを計算します
- 平文パスワードと SHA-256 ハッシュの両方を安全に記録します

2.5.3 デバッグ インターフェイスの使用

メイン以外のフラッシュ領域が書き込み保護されていない場合、デバッグ プローブを使用して BSL 構成を直接プログラムできます。

1. デバッグ プローブ (XDS110 など) をデバイスに接続します
2. Code Composer Studio フラッシュ ローダまたは類似のツールを使用します
3. BSL 構成バイナリ ファイルをロードします
4. アドレス 0x80101C00 から始まるメイン以外のフラッシュにプログラムします

Copyright © 2026 Texas Instruments Incorporated

3 ブートローダのプロトコル

3.1 パケット フォーマット

BSL のデータ パケットは階層構造です。BSL のコア コマンドには、BSL によって処理される実際のコマンド データが含まれています。標準の BSL コマンドに加えて、各コア コマンドの前後に、ペリフェラル インターフェイス コード (PI コード) と呼ばれるラッパー データを置くことができます。このラッパー データは、使用されているペリフェラルやプロトコルに固有の情報で、BSL コア コマンドを正しく送信するための情報が含まれています。ラッパーとコア コマンドを組み合わせると、BSL データ パケットが構成されます

PI コード	BSL コア データ	PI コード

3.2 BSL プロトコル

UART および I2C BSL プロトコルのデータ パケットの構造は次のとおりです。

- ヘッダーのバイトは、使用されているプロトコルとパケットのタイプ (コマンドまたは応答パケット) を示します。
- 長さフィールドには、BSL コア データのサイズがバイト単位で含まれます。
- BSL コア データには、コマンドに必要なコマンド / 応答 ID とアドレスおよびデータが含まれています
- CRC32 フィールドには、BSL コア データ内のデータについて計算された CRC が含まれています

PI コード		BSL コア データ	PI コード
ヘッダー (1 バイト)	長さ (2 バイト)	BSL コアのコマンド / 応答	CRC32 (4 バイト)

データ パケットは、コア データ フィールドによってコマンド パケットまたは応答パケットに分類されます。

コマンド パケットは、BSL に送信される最初のパケットです。2 番目のパケットは、BSL から受信される応答パケットです。応答パケットには、BSL アクノリッジと BSL コア応答の 2 つのコンポーネントが含まれています。このうち、アクノリッジは送信されるすべてのコマンド パケットについて、BSL から受信されます。これに対して、BSL コア応答はすべてのコマンドについて受信されるわけではありません。

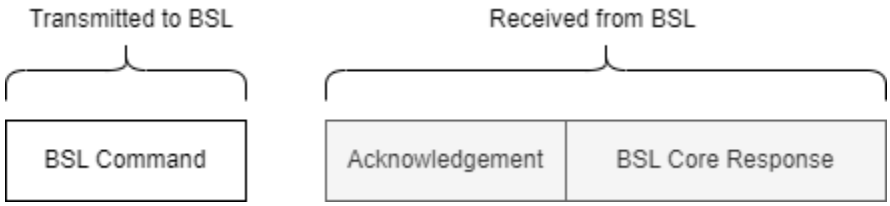


図 3-1. BSL プロトコル

3.2.1 BSL アクノリッジ

BSL ソフトウェアのペリフェラル インターフェイス セクションは、BSL データ パケットのラッパー セクションを解析します。データ送信でエラーが発生した場合、エラー メッセージが直ちに送信されます。すべてのデータが正常に受信された後に ACK が送信されます。これは、データ パケットが正しくフォーマットされ、BSL コア ソフトウェアに渡されて解釈が行われたという意味で、コマンドが正しく実行されたという意味ではなく、コマンドが有効であったことも保証されていません。

BSL プロトコルでは、送信されるすべての BSL データ パケットに対して、送信されるすべての BSL データ パケットに加えて、単一バイトのアクノリッジ応答が行われると規定されています。BSL からのアクノリッジ応答値を、表に示します。ACK 以外のアクノリッジ バイトが送信された場合、BSL は BSL データ パケットを送信しません。ホスト プログラマはアクノリッジ エラーをチェックし、送信を再試行する必要があります。

データ	意味
0x00	BSL_ACK (パケットを正常に受信した)
0x51	BSL_ERROR_HEADER_INCORRECT
0x52	BSL_ERROR_CHECKSUM_INCORRECT
0x53	BSL_ERROR_PACKET_SIZE_ZERO

Copyright © 2026 Texas Instruments Incorporated

データ	意味
0x54	BSL_ERROR_PACKET_SIZE_TOO_BIG
0x55	BSL_ERROR_UNKNOWN_ERROR
0x56	BSL_ERROR_UNKNOWN_BAUD_RATE

3.2.2 ペリフェラルの構成

3.2.2.1 UART

次の構成で UART がイネーブルです。

- UC12 が使用されます
- ボーレートはデフォルトで 9600bps です。これは、ボーレートの変更コマンドで更新できます
- データ幅:8 ビット
- ストップ ビット:1
- パリティなし
- RXD および TXD に使用されるピンは、BSL の構成から取得されます

3.2.2.2 I2C

BSL の I2C インターフェイスは、I2C ターゲットとして動作できます。ホストはコントローラとして動作し、通信を行います。

- UC15_0 が使用されます
- I2C ターゲット アドレスはデフォルトで 0x48 です。BSL の構成で設定できます
- SCL および SDA ラインには外部プルアップが必要です
- SDA および SCL に使用されるピンは、BSL の構成から取得されます

3.2.2.3 CRC

次のものについて、データの CRC を計算する必要があります。

- CRC32-ISO3309 多項式
- ビット反転の構成
- 初期シード - 0xFFFFFFFF

3.3 ブートローダのコア コマンド

3.3.1 接続

構造

ヘッダー	長さ		CMD	CRC32			
0x80	0x01	0x00	0x12	C1	C2	C3	C4

説明

接続コマンドは、特定のインターフェイス (UART または I2C) を経由してホストとターゲットとの間の接続を確立するために使用される、最初のコマンドです。

保護

無

コマンドの戻り値

BSL アクノリッジのみ

例

ホスト:80 01 00 12 3A 61 44 DE

BSL: 00

3.3.2 デバイス情報の取得

構造

Copyright © 2026 Texas Instruments Incorporated

ヘッダー	長さ		CMD	CRC32			
0x80	0x01	0x00	0x19	C1	C2	C3	C4

説明

このコマンドは、バージョン情報と、データ トランザクションで使用可能なバッファ サイズを取得するために使用されます。

保護

無

コマンドの戻り値

BSL アクノリッジと、デバイス情報を含む BSL コアの応答。詳細については、「デバイス情報」を参照してください。

例

ホスト:80 01 00 19 B2 B8 96 49

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

3.3.3 ブートローダーのロック解除

構造

ヘッダー	長さ		CMD	データ	CRC32			
0x80	0x21	0x00	0x21	D1...D32	C1	C2	C3	C4

説明

このコマンドは、ブートローダーのロック解除に使用されます。ブートローダーのロック解除後にのみ、セクション 3.3 に記載されている、保護されているすべてのコマンドが BSL によって処理されます。

ホストが誤ったパスワードを送信した場合、BSL はエラー応答を送信し、デバイスは 2 秒間スリープ モードに移行します。この 2 秒間のウィンドウでは、いかなるコマンドも受信または処理されません。2 秒後に、ホストは次のパスワードを送信できます。誤ったパスワードが 3 回送信された場合、セキュリティ アラートのアクションが実行されます。セキュリティ アラートのアクションは、BSL の構成で設定できます。詳細については、セクション 3.5.1.1 を参照してください。

保護

無

データ

32 バイトの BSL パスワードが、BSL の構成メモリに保存されます。新しいデバイスのデフォルト値はすべて 0xFF です。

コマンドの戻り値

BSL アクノリッジ、および動作ステータスについてのメッセージを含む BSL コア応答。詳細については、「BSL コア メッセージ」セクションを参照してください。

例

ホスト:80 21 00 21 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 02 AA F0 3D

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

3.3.4 データのプログラム

構造

ヘッダー	長さ		CMD	アドレス	データ	CRC32			
0x80	L1	L2	0x20	A1...A4	D1...Dn	C1	C2	C3	C4

Copyright © 2026 Texas Instruments Incorporated

説明

このコマンドは、データを特定のメモリ アドレスにプログラムするために使用されます。データは、メイン フラッシュ メモリ、情報フラッシュ メモリ (メイン以外)、またはデータ バンク フラッシュ メモリにプログラムできます。

重要な要件:

- 開始アドレスは、16 バイト アラインされている必要があります (アドレス % 16 == 0)
- データ長は、16 バイト アラインされている必要があります (長さ % 16 == 0)
- 情報フラッシュ メモリ (メイン以外) へのプログラミングを行うには、BCR 構成で工場出荷時リセットを有効化する必要があります

保護

有

アドレス

プログラムするメモリ領域の開始アドレス。A1...A4 で、A1 は 32 ビット アドレスの最下位バイトです。

データ

指定されたアドレスに書き込まれるデータ バイト。送信可能なデータの最大サイズは、デバイスのバッファ サイズによって制限されます。バッファ サイズは、デバイス情報の取得コマンドで確認できます。

コマンドの戻り値

BSL アクノリッジ、および動作ステータスについてのメッセージを含む BSL コアの応答。詳細については、「セクション 3.4.1」セクション を参照してください。

例

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

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

3.3.5 データの高速プログラム

構造

ヘッダー	長さ		CMD	アドレス	データ	CRC32			
0x80	L1	L2	0x24	A1...A4	D1...Dn	C1	C2	C3	C4

説明

データの高速プログラム コマンドは、データのプログラム コマンドと同じですが、このコマンドはプログラミング処理を高速にするため、ノンブロッキング書き込みを行います。BSL は、このコマンドに対して、プログラミングが成功したかどうかを示す BSL コア メッセージ応答を送信しません。

保護

有

アドレス

プログラムするメモリ領域の開始アドレス。A1...A4 で、A1 は 32 ビット アドレスの最下位バイトです。

データ

指定されたアドレスに書き込まれるデータ バイト。送信可能なデータの最大サイズは、デバイスのバッファ サイズによって制限されます。バッファ サイズは、デバイス情報の取得コマンドで確認できます。

コマンドの戻り値

BSL アクノリッジ。

例

Copyright © 2026 Texas Instruments Incorporated

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

BSL: 00

3.3.6 データの読み戻し

構造

ヘッダー	長さ		CMD	アドレス	データ	CRC32			
0x80	0x09	0x00	0x29	A1...A4	L1...L4	C1	C2	C3	C4

説明

このコマンドは、アドレス A1...A4 から始まるデータを読み出すために使用されます。

このコマンドを使用してデータを読み出すには、BSL の構成で読み出しをイネーブルにする必要があります。この機能は、BSL 構成のデフォルトで無効になっています。

データは、メイン フラッシュ (アプリケーション メモリ)、メイン以外のフラッシュ (構成メモリ)、SRAM メモリから読み出すことができます。

注

ホストから SRAM メモリへのアクセスには制限があります。詳細については、「SRAM メモリの使用」を参照してください。

保護

有

アドレス

読み戻すメモリ領域の開始アドレス。A1...A4 で、A1 は 32 ビット アドレスの最下位バイトです。

データ

読み取るデータのサイズをバイト単位で、L1...L4 により表します。ここで、L1 は最下位バイトです。読み取るデータの最大サイズは、デバイスのバッファ サイズによって制限されます。バッファ サイズは、デバイス情報の取得コマンドで確認できます。

コマンドの戻り値

BSL アクノリッジと、読み戻しコマンドが有効なら要求されたデータを含む BSL コア応答。詳細については、セクション 3.4.3 を参照してください。

読み戻しコマンドのアドレスまたは長さが無効、あるいは読み出しが無効であれば、BSL アクノリッジに続いて、メッセージ応答として対応するエラーが送信されます。

例

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

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

3.3.7 フラッシュの範囲消去

構造

ヘッダー	長さ		CMD	アドレス	データ	CRC32			
0x80	0x09	0x00	0x23	A1...A4 (開始アドレス)	A1...A4 (終了アドレス)	C1	C2	C3	C4

説明

Copyright © 2026 Texas Instruments Incorporated

フラッシュの範囲消去コマンドは、フラッシュ メモリの指定した領域を消去するために使用されます。フラッシュはセクタ (2KB) 単位で消去され、それより小さなサイズを消去することはできません。

開始アドレスと終了アドレスが異なるフラッシュ セクタに存在する場合、BSL は開始アドレスと終了アドレスの間にあるすべてのフラッシュ セクタを消去します。これは、これらのアドレスを含むセクタも対象になります。

このコマンドは、メイン フラッシュ メモリのみを消去できます。メイン以外は消去できません。

終了アドレスを開始アドレスよりも小さくしないでください。

保護

有

アドレス

消去するメモリ領域の開始アドレス。A1...A4 で、A1 は 32 ビット アドレスの最下位バイトです。

データ

消去するメモリ領域の終了アドレス。A1...A4 で、A1 は 32 ビット アドレスの最下位バイトです。

コマンドの戻り値

BSL アクノリッジ、および動作ステータスについてのメッセージを含む BSL コアの応答。詳細については、「セクション 3.4.1」セクション を参照してください。

例

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

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

3.3.8 一括消去

構造

ヘッダー	長さ		CMD	CRC32			
0x80	0x01	0x00	0x15	C1	C2	C3	C4

説明

一括消去コマンドは、デバイスで使用可能なメイン フラッシュ メモリ (アプリケーション メモリ) 全体を消去します。

BCR の構成メモリの一括消去構成は、この BSL コマンドに影響しません。

BCR の構成メモリでフラッシュ領域が静的書き込み保護されている場合、その領域は消去できません。

保護

有

コマンドの戻り値

BSL アクノリッジ、および動作ステータスについてのメッセージを含む BSL コアの応答。詳細については、セクション 3.4.1 を参照してください。

例

ホスト:80 01 00 15 99 F4 20 40

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

3.3.9 工場出荷時リセット

構造

Copyright © 2026 Texas Instruments Incorporated

ヘッダー	長さ		CMD	データ	CRC32			
0x80	L1	L2	0x30	D1...D16	C1	C2	C3	C4

説明

工場出荷時リセット コマンドは、メインのフラッシュ (アプリケーション) メモリとメイン以外のフラッシュ (構成) メモリをすべて消去します。

このコマンドの処理は、BCR の構成メモリの工場出荷時リセット構成によって変わります。

工場出荷時リセットの許可は、次のように決定されます。

- 「イネーブル」なら、パスワードなしで許可
- 「イネーブルでパスワードが必要」なら、パスワードを指定すれば許可
- 「ディスエーブル」なら不許可

BCR の構成メモリでフラッシュ領域が静的書き込み保護されている場合は、その領域を消去できません。

保護

有

データ

16 バイトの工場出荷時リセット パスワードが、BCR の構成メモリに保存されます。デフォルトのパスワードは、全て 0xFF です。パスワードは、BCR の構成で工場出荷時リセットが「イネーブルでパスワードが必要」になっている場合のみ必要です。

コマンドの戻り値

BSL アクノリッジ、および動作ステータスについてのメッセージを含む BSL コアの応答。詳細については、セクション 3.4.1 を参照してください。

> **注意**
>
> 工場出荷時リセットを行った後で、メイン以外の構成が復元されるまで、システムは潜在的なロックアウト状況に対して非常に脆弱です。これが発生すると、デバイスに再度アクセスできなくなります。

例

シナリオ 1:パスワードなしでの工場出荷時リセット

ホスト:80 01 00 30 DE 20 24 0B

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

シナリオ 2:パスワードを使用した工場出荷時リセット

ホスト:80 11 00 30 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 8A 28 EA DC

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

3.3.10 スタンドアロン検証

構造

ヘッダー	長さ		RSP	データ	CRC32			
0x08	0x05	0x00	0x32	D1...D4	C1	C2	C3	C4

説明

このコマンドは、指定されたメモリ範囲に格納されているデータの CRC を検証するために使用されます。これにより、プログラムされたデータをより迅速に検証できます。データ サイズは最小 1kB にする必要があります。

Copyright © 2026 Texas Instruments Incorporated

CRC 検証は、メイン フラッシュ (アプリケーション メモリ) とメイン以外のフラッシュ (構成メモリ) の両方で許可されています

保護

有

データ

検証されるデータのサイズをバイト単位で、D1...D4 により表します。ここで、D1 は最下位バイトです。1KB ≦ サイズ ≦ 64KB。

コマンドの戻り値

BSL アクノリッジと、要求されたメモリ領域について計算された CRC 値を含む BSL コア応答。詳細については、セクション 3.4.5 を参照してください。

検証コマンドのアドレスまたは長さが無効な場合、BSL アクノリッジに続いて、対応するエラーがメッセージ応答として送信されます。セクション 3.4.1 を参照してください。

例

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

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

3.3.11 アプリケーションの開始

構造

ヘッダー	長さ		CMD	CRC32			
0x80	0x01	0x00	0x40	C1	C2	C3	C4

説明

アプリケーションの開始コマンドは、システム リセットを発行します。これによりブートローダーは終了し、ブートコードが再実行されて、アプリケーションが開始されます。

保護

無

コマンドの戻り値

BSL アクノリッジ

例

ホスト:80 01 00 40 E2 51 21 5B

BSL:00

3.3.12 ボーレートの変更

構造

ヘッダー	長さ		CMD	データ	CRC32			
0x80	0x02	0x00	0x52	D1	C1	C2	C3	C4

説明

このコマンドを使用して、UART インターフェイスのボーレートを変更できます。新しいボーレートは、このパケットについて BSL アクノリッジを送信した後に有効になります。

BSL UART のデフォルトのボーレートは 9600bps です。

Copyright © 2026 Texas Instruments Incorporated

注

ボーレートを更新した後で、ブートローダーのロック解除コマンドで誤った BSL パスワードが送信されると、ボーレートの設定はデフォルト値に戻ります。それ以後の通信はデフォルトのボーレートで行われます。

保護

無

データ

表に記載されている D1 ボーレート。

ID	ボーレート (bps)
1	4800
2	9600
3	19200
4	38400
5	57600
6	115200
7	1000000
8	2000000
9	3000000

コマンドの戻り値

BSL アクノリッジ

例

ホスト:80 02 00 52 03 6C 83 A2 AF

BSL: 00

3.4 ブートローダーのコア応答

BSL の応答	RSP	データ
メモリの読み戻し	0x30	D1…Dn
デバイス情報の取得	0x31	D1…D24
スタンドアロン検証	0x32	D1…D4
メッセージ	0x3B	MSG
詳細なエラー	0x3A	D1..D3

略語:

MSG

BSL コアからの応答を含む 1 バイトで、要求されたアクションの結果を表しています。エラー コード、または動作が成功したことを示すアクノリッジです。BSL がデータ (メモリ、バージョン、CRC、バッファ サイズなど) を返す必要がある場合、動作が成功したという応答は返されず、BSL コアは直接データを送信します。

D1..Dn

データ バイト。「n」は BSL の最大バッファ サイズによって制限されます

3.4.1 BSL のコア メッセージ

構造

Copyright © 2026 Texas Instruments Incorporated

ヘッダー	長さ		RSP	データ	CRC32			
0x08	0x02	0x00	0x3B	MSG	C1	C2	C3	C4

説明

BSL は一部のコマンドについて、処理したコマンドのステータスを示すメッセージ応答をホストに送信します。BSL から返される可能性があるすべてのメッセージを、次の表に示します。

MSG	意味	考えられる原因(1)
0x00	動作の成功	
0x01	BSL ロック中のエラー	BSL は、ブートローダのロック解除パスワード コマンドでロック解除されていない。または、BSL のロック解除後に、コマンド受信フェーズでタイムアウトが発生した。
0x02	BSL のパスワード エラー	ブートローダのロックを解除するために送信されたパスワードが正しくない。
0x03	複数回の BSL パスワード エラー。セキュリティ アラートのアクションが実行される。	ブートローダーのロック解除で、正しくないパスワードが 3 回送信された。
0x04	未知のコマンド	BSL に与えられたコマンドが有効なコマンドとして認識されなかった
0x05	無効なメモリ範囲	指定されたメモリ範囲は無効。
0x06	無効なコマンド	BSL に与えられたコマンドは既知のコマンドであるが、その時点では無効なもので、処理できない。
0x07	工場出荷時リセットが無効	BCR の構成で、工場出荷時のリセットが無効になっている
0x08	工場出荷時リセットのパスワード エラー	BCR の構成で工場出荷時リセットが「有効でパスワードが必要」に設定されているときに、工場出荷時リセットのコマンドでパスワードが送信されていないか、正しくない
0x09	読み出しエラー	メモリの読み出しが、BCR の構成で無効になっている
0x0A	アドレスまたは長さのアライメントが無効	フラッシュへのプログラムで、開始アドレスまたはデータの長さが 8 バイト境界になっていない
0x0B	スタンドアロン検証の長さが無効	スタンドアロン検証用に送信されたデータのサイズが 1KB 未満

(1) この表には、ステータスやエラーが発生した原因として考えられるものが記載されていますが、これ以外の原因がないとは限りません。また、ここに記載されているのはエラーの原因のうち、ソフトウェアに関する、ホストによって訂正可能なものだけです。

3.4.2 詳細なエラー

構造

ヘッダー	長さ		RSP	データ	CRC32			
0x08	0x02	0x00	0x3A	D1..D3	C1	C2	C3	C4

説明

D1 - エラー タイプ

D3、D2 - エラーの詳細

可能な値

エラーのタイプ		エラーの詳細	
値	説明	値	説明
0xF0	フラッシュ エラー	0xXX	FLASHCTL.STATCMD レジスタの値が含まれています

3.4.3 メモリの読み戻し

構造

Copyright © 2026 Texas Instruments Incorporated

ヘッダー	長さ		RSP	データ	CRC32			
0x08	0x02	0x00	0x30	D1...Dn	C1	C2	C3	C4

説明

このコマンドは、読み戻しコマンドへの応答として、要求されたデータを返します

データ

データ D1..DN。「n」は BSL の最大バッファ サイズによって制限されます。

3.4.4 デバイス情報

構造

ヘッダー	長さ		RSP	データ	CRC32			
0x08	0x19	0x00	0x31	D1...D24	C1	C2	C3	C4

説明

このコマンドは、ID の取得コマンドへの応答として、バージョン情報と BSL バッファ サイズを返します

データ

ID バイト	データ バイト
コマンド インタープリタのバージョン	[D02-D01]
ビルド ID	[D04-D03]
アプリケーションのバージョン	[D08-D05]
アクティブ プラグイン インターフェイスのバージョン	[D10-D09]
BSL の最大バッファ サイズ	[D12-D11]
BSL バッファの開始アドレス	[D16-D13]
BCR の構成 ID	[D20-D17]
BSL の構成 ID	[D24-D21]

アプリケーションのバージョン:

32 ビットのアプリケーション バージョンは、BSL の構成で指定されているアドレスから取得されます

BSL バッファ サイズ:

送受信される BSL データ パケットを格納するために使用できる RAM データ バッファのサイズ。

例

ホスト: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

上の応答は次のように解釈されます。

コマンド インタープリタのバージョン - 0x0100

ビルド ID - 0x0100

アプリケーションのバージョン - 0x00000000

アクティブ プラグイン インターフェイスのバージョン - 0x0001

BSL の最大バッファ サイズ - 0x06C0

BSL バッファの開始アドレス - 0x20000160

BCR の構成 ID - 0x00000001

Copyright © 2026 Texas Instruments Incorporated

BSL の構成 ID - 0x00000001

3.4.5 スタンドアロン検証

構造

ヘッダー	長さ		RSP	データ	CRC32			
0x08	0x05	0x00	0x32	D1...D4	C1	C2	C3	C4

説明

このコマンドは、スタンドアロンの検証コマンドへの応答として CRC 値を返します

データ

要求されたメモリ領域について計算された 32 ビットの CRC 値。D1...D4 で、D1 は CRC32 の最下位バイトです。

3.5 ブートローダのセキュリティ

3.5.1 パスワードで保護されているコマンド

メモリのデータに直接または間接的にアクセスできるすべてのコマンドは、パスワードで保護されています。このパスワードは、メイン以外のメモリの BSL 構成で設定できます。

誤ったパスワードが送信されると、デバイスは次の 2 秒間スリープ状態になり、ブルートフォース攻撃を困難にするため、この期間中はいかなるコマンドも受け付けません。誤ったパスワードが 3 回送信されると、BSL によってセキュリティ アラート アクションが実行されます。

3.5.1.1 セキュリティ アラート

セキュリティ アラートは、BSL 構成で、次に示す 3 つのモードのいずれかに構成できます。

1. **工場出荷時リセット** — 工場出荷時リセットは、メインおよびメイン以外のフラッシュ メモリ全体を消去します。この消去は、BCR の構成の工場出荷時リセット モードに関係なく行われます。フラッシュの特定の部分が静的書き込み保護されている場合、BSL はフラッシュ メモリ全体を消去できません。
2. **ブートローダーの無効化** — BCR の構成でブートローダーを無効化し、BSL を終了します。BCR のメイン以外の構成を更新してブートローダーを有効にしない限り、それ以後は BSL を使用できません。メイン以外について静的書き込み保護が有効になっている場合、BSL は無効化されません。
3. **何もしない** - 何の動作も行いません。

注

工場出荷時リセットのセキュリティ アラート構成で、メイン以外が消去されると、ブートローダーのパスワードはデフォルトに戻ります。これを防止するために、メイン以外を書き込み保護できます。メイン以外が消去済み状態のままの場合、デバイスはロックされ、以後デバイスにアクセスできなくなります。

3.5.2 BSL エントリ

ブートローダへのエントリは、ブートコード経由でのみ許可されます。

ブートローダは、BCR の構成で無効にできます。

Copyright © 2026 Texas Instruments Incorporated

4 ブートローダーによるプログラムのフローのサンプル

このセクションでは、BSL ホストがブートローダーでイメージをロードするときの代表的なシーケンスについて説明します。このサンプル シーケンスは、フラッシュ メモリを消去し、新しいファームウェアをプログラムします。

- ブートローダーは、以下のいずれかの方法で起動する必要があります。
  - ピン ベースの起動 (GPIO)
  - アプリケーション要求
  - メールボックスのデバッグ コマンド
- 起動すると、目的のインターフェイスを経由して接続コマンドを送信し、BSL との接続を確立します。
- UART インターフェイスを使用する場合は、ボーレートをより高い値に変更して、通信をさらに高速化することができます。これはオプションです。
- フラッシュ メモリを完全に消去するには、一括消去コマンドを使用します。メイン以外のフラッシュを更新する必要がある場合のみ、工場出荷時リセット コマンドを使用します。これは、メイン以外のフラッシュが消去され、プログラムされていない状態のままだと、デバイスがロックされるためです。
- ファームウェア イメージをプログラムします
- プログラムされたメモリ領域の CRC 検証を行い、プログラムされたデータの正確性をチェックします。この手順はオプションです。
- アプリケーションは、「アプリケーションの開始」コマンドで開始できます。

Copyright © 2026 Texas Instruments Incorporated

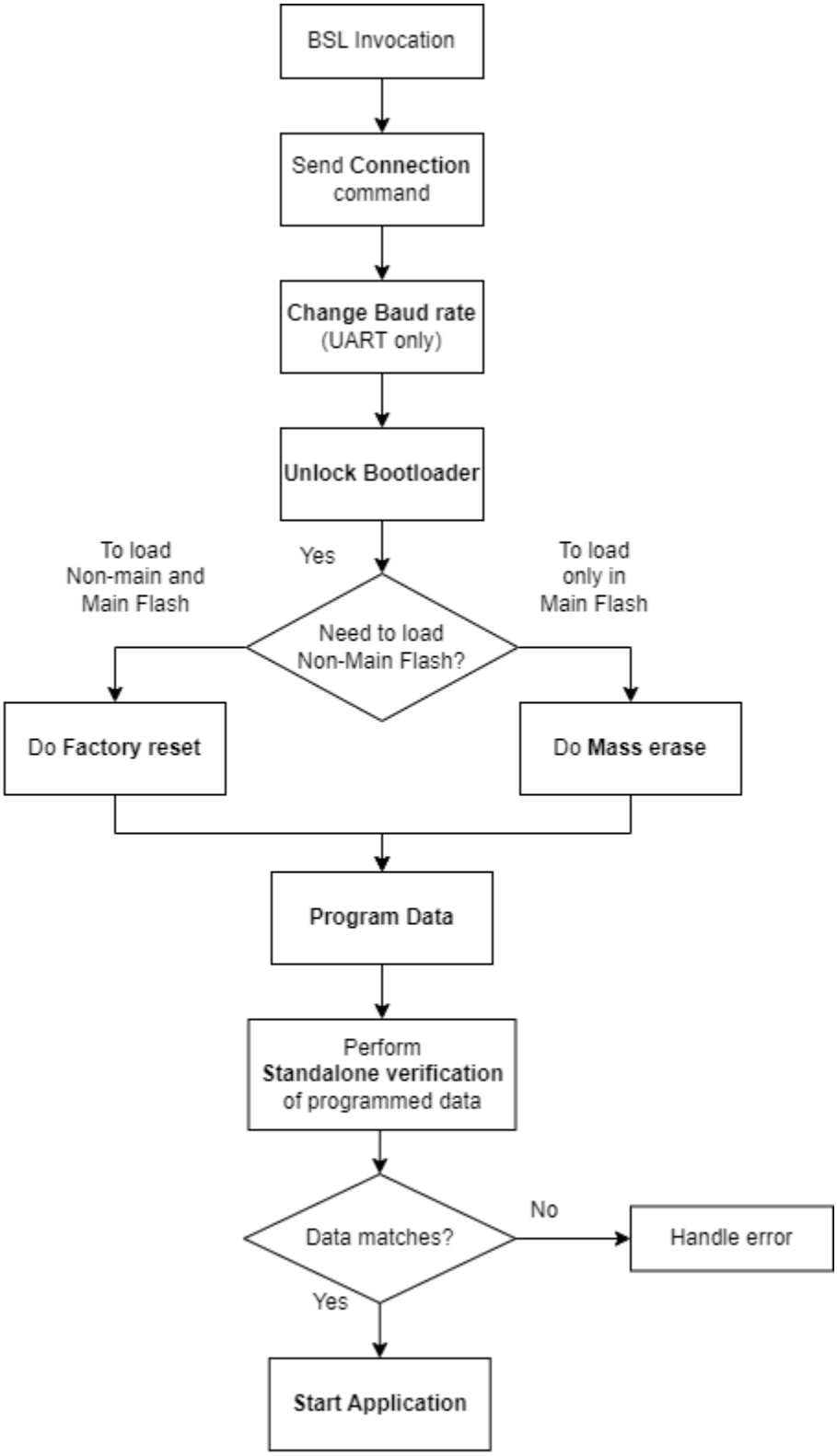


図 4-1. BSL ホストのシーケンス

Copyright © 2026 Texas Instruments Incorporated

5 改訂履歴

資料番号末尾の英字は改訂を表しています。その改訂履歴は英語版に準じています。

日付	改訂	注
2025 年 12 月	*	初版リリース

Copyright © 2026 Texas Instruments Incorporated

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、 TI の販売条件 、 TI の総合的な品質ガイドライン 、 ti.com または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。 TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月