参考資料

User's Guide

MSPM33C セキュリティ ユーザー ガイド

TEXAS INSTRUMENTS

概要

このドキュメントは、アプリケーション開発時に MSPM33C デバイス ファミリに組み込まれているセキュリティ機能を利用するためのガイドとして活用できます。セキュリティ機能には TrustZone® と複数のセキュリティ アクセラレータが含まれます。セキュア ブートまたはセキュア デバッグの実装の詳細については、M33 TRM を参照してください。

目次

1 セキュリティ機能の概要 ... 3
1.1 用語 ... 3
1.2 プラットフォームのセキュリティ イネーブラ ... 4
2 デバイス セキュリティ モデル ... 6
2.1 デバイス ID ... 6
2.2 ブート時の初期条件 ... 6
2.3 ブート構成ルーチン (BCR) ... 7
2.4 ブートストラップ ローダ (BSL) ... 8
2.5 ライフサイクル管理 ... 8
2.5.1 ライフサイクルの遷移 ... 9
2.6 ブート サービス ... 10
2.7 ブート フロー ... 11
2.8 ユーザー指定のセキュリティ ポリシー ... 14
2.8.1 ブート構成ルーチン (BCR) ポリシー ... 14
2.8.2 ブートストラップ ローダ (BSL) のセキュリティ ポリシー ... 19
2.8.3 カスタマ セキュア コード (CSC) セキュリティ ポリシー ... 20
2.8.4 構成データのエラー耐性 ... 20
3 セキュア ブート ... 22
3.1 MCUBoot ... 22
3.2 ROM ベースのセキュア ブート ... 25
3.2.1 ROM ベースのセキュア ブート フロー ... 25
3.2.2 顧客キー ストレージ ... 27
3.2.3 アンチ ロールバック保護 ... 29
3.3 CSC ベースのセキュア ブート ... 29
3.3.1 CSC ベースのセキュア ブート フロー ... 30
3.3.2 FLASH メモリ マップ ... 31
3.3.3 CSC キー ストレージ ... 32
3.3.4 バンク スワップ ... 33
3.3.5 CSC 保護 ... 33
3.4 2 段階セキュア ブート ... 34
3.4.1 2 段階セキュア ブート フロー ... 34
4 セキュアな実行環境 ... 37
4.1 TrustZone® ... 37
4.1.1 実装定義属性ユニット ... 37
4.1.2 セキュリティ属性単位 ... 38
4.1.3 TrustZone® ソフトウェア開発 ... 38
4.2 メモリ保護ユニット ... 39
4.2.1 TrustZone® および MPU ... 39
4.3 グローバル セキュリティ コントローラ ... 39
4.3.1 GSC メモリ構成 ... 39

English Document: SLUUDG7
Copyright © 2026 Texas Instruments Incorporated

5 セキュア ストレージ 42
5.1 書き込み消去保護 42
5.1.1 WEP でのグローバル セキュリティ コントローラ構成 43
5.1.2 WEP での NONMAIN フラッシュ構成 43
5.2 フラッシュ非表示保護 44
5.2.1 HDP でのグローバル セキュリティ コントローラ構成 45
5.3 セキュアなキー ストレージ 45
5.4 データ整合性の検証 46
5.5 セキュアと特権保護 46
5.6 ハードウェア単調カウンタ 46
6 セキュリティ モジュール 48
6.1 AESADV 48
6.1.1 AES の概要 48
6.1.2 AES の使用法 49
6.2 キーストア 50
6.2.1 概要 50
6.2.2 キーストアの使用方法 50
6.3 SHA2 50
6.3.1 SHA の概要 51
6.3.2 SHA の性能 51
6.3.3 SHA の使用法 51
6.4 PKA 52
6.4.1 PKA の概要 52
6.4.2 PKA の使用法 52
6.5 PQC 53
6.5.1 ML-DSA 53
7 改訂履歴 54

商標

TrustZone®, Arm®, and Cortex® are registered trademarks of Arm Limited.
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 セキュリティ機能の概要

MSPM33C デバイス ファミリが搭載しているセキュリティ機能は、デバイスに展開したセキュア データの整合性と真正性を維持するためのフレームワークを構築します。

MSPM33C デバイス ファミリに搭載されている主なセキュリティ機能は次のとおりです。

- 変更不可能な信頼ルート (iROT)
- セキュア ブート
  - ROM ベースのセキュア ブート
  - CSC ベースのセキュア ブート
  - 2 段階セキュア ブート
- セキュアな実行環境
  - TrustZone®
    - 実装定義属性ユニット (IDAU)
    - セキュリティ属性ユニット (SAU)
  - メモリ保護ユニット (MPU)
    - TrustZone® および MPU
  - グローバル セキュリティ コントローラ (GSC)
    - SRAM 保護コントローラ (SPC)
    - フラッシュ保護コントローラ (FPC):
    - ペリフェラル保護コントローラ (PPC)
- セキュア ストレージ
  - 書き込み消去保護 (WEP)
  - フラッシュ非表示保護 (HDP)
  - セキュアなキー ストレージ
  - データ整合性の検証
  - ハードウェア単調カウンタ
- セキュリティ アクセラレータ
  - 高度暗号化標準 (AES)
  - キーストア
  - セキュア ハッシュ アルゴリズム (SHA)
  - 公開鍵アクセラレータ (PKA)
  - ポスト量子暗号 (PQC)

1.1 用語

GSC	グローバル セキュリティ コントローラ
NONMAIN	デバイスのブート関連パラメータを構成する専用フラッシュ メモリ領域。NONMAIN の操作ガイドについては、『MSPM33 NONMAIN フラッシュ操作ガイド』を参照してください。
SRAM	シンクロナス ランダム アクセス メモリ
MMR	メモリ マップト レジスタ
CSC	カスタマ セキュア コード
SHA	セキュア ハッシュ アルゴリズム
ECDSA	楕円曲線デジタル署名アルゴリズム
MLDSA	モジュール格子ベース デジタル署名アルゴリズム
SAU	セキュリティ属性単位
IDAU	実装定義属性ユニット
MPU	メモリ保護ユニット

Copyright © 2026 Texas Instruments Incorporated

HDP	フラッシュ非表示保護
WEP	書き込み消去保護
AES	高度暗号化規格
TRNG	真の乱数ジェネレータ
PKA	公開鍵アクセラレータ

1.2 プラットフォームのセキュリティ イネーブラ

表 1-1 に、MSPM33 デバイスに搭載されているセキュリティ イネーブラを示します。デバッグ セキュリティ機能およびメイン フラッシュ メモリの整合性検証機能については、デバイス シリーズのテクニカル リファレンス マニュアルの「NONMAIN レイアウト タイプ」および「NONMAIN レジスタ」セクションを参照してください。セキュア ブート、セキュア ストレージ、暗号化アクセラレータの各機能は、各デバイス固有のデータシートに記載されています。

表 1-1. MSPM33 マイコン プラットフォームのセキュリティ イネーブラ

セキュリティ イネーブラ	デバイスの機能	M33C321x
デバッグのセキュリティ	認証を使用したデバッグ アクセス [1]	パスワード / 証明書 [2]
	認証を使用したブートストラップ ローダ アクセス	パスワード / 証明書 [2]
	認証を使用したメイン / データ フラッシュ メモリの一括消去	パスワード / 証明書 [2]
	認証を使用した完全な工場出荷時リセット	パスワード / 証明書 [2]
	不良デバイス変換時の認証を使用したコマンド	パスワード / 証明書 [2]
	シリアル ワイヤ デバッグ (SWD) インターフェイスの完全なハードウェア ディセーブル	有
	デバイス構成データを永続的にロック可能	有
	エラー耐性のあるデバイス構成データ	有
変更不可能な信頼ルート	簡素化されたキー プロビジョニング フローのサポート	有
	セキュア キー プロビジョニング フローのサポート	
	OTP キー ストレージ領域	有
	ファームウェア セキュア インストール	有
	ライフサイクルの維持とサポート	有
	ROM ベースのセキュア ブート	有
	非対称ファームウェア イメージ認証ルーチン (ECDSA P-256、SHA2-256)	変更不可能
	ロールバック保護	有
CSC ベースのセキュア ブート	CSC が存在します	有
	メイン フラッシュ メモリを永続的にロック可能	有
	ブート時に MAIN フラッシュ アプリケーションへのエントリ ポイントを 1 つに制限	有
	非対称ファームウェア イメージ認証ルーチン(ECDSA P-256、SHA2-256、MLDSA-87、いずれもソフトウェア実装)	変更可能
	公開鍵の失効およびロールバック保護のためのロック可能なフラッシュ	有
	バンク スワップのサポート	有
	GSC を使用した書き込み / 消去 / 読み取り / 実行保護	有
セキュアな実行環境	TrustZone アーキテクチャ	有
	TrustZone での SAU および IDAU 構成	有
	TrustZone での GSC 構成	有
	メモリ保護ユニット	有
	MPU での GSC 構成	有

Copyright © 2026 Texas Instruments Incorporated

表 1-1. MSPM33 マイコン プラットフォームのセキュリティ イネーブラ (続き)

セキュリティ イネーブラ	デバイスの機能	M33C321x
セキュア ストレージ	メインおよびデータ フラッシュ領域の書き込み消去保護 (GSC を使用)	有
	メイン フラッシュ領域のフラッシュ非表示保護 (GSC を使用)	有
	CRC-32 検証を使用したメイン フラッシュ領域	有
	SHA2-256 検証を使用したメイン フラッシュ メモリ領域	有
	AES キーストア	有
	ハードウェア単調カウンタ	有
暗号化アクセラレーション	セルフテスト機能を備えた真性乱数生成器 (TRNG)	有
	高度な AES アクセラレータ (GCM/CMAC/GHASH をサポート)	有
	SHA (SHA-224、HMAC による SHA-256)	有
	公開鍵アクセラレータ (ECDSA、RSA)	有
デバイス ID	固有のデバイス識別子 (96 ビット)	有
認証	ARM PSA レベル	L2 計画済み
	EVITA 機能	EVITA-Light
	ISO 21434 プロセス準拠	予定
攻撃抵抗分析	3P ファームウェアの脆弱性分析	有
	ブート構成ルーチンによるフォルト注入攻撃への対策	有

1. trustzone をサポートするデバイスの場合、認証されたデバッグ アクセスは分割され、セキュア デバッグ アクセスと非セキュア アクセスが実現されます。詳細情報については、セクション 2.8.1.1 を参照してください。
2. HSFS デバイスでは、認証はパスワード ベースです。HSSE デバイスでは、認証は証明書ベースです。

Copyright © 2026 Texas Instruments Incorporated

2 デバイス セキュリティ モデル

MSPM33 ファミリには、次の 3 つの大規模なセキュリティ機能段階があります。

1. フラッシュ アプリケーションに入る前に、ユーザー定義の構成によって制御される、TI が作成した静的なブートコード
2. MAIN フラッシュに格納され、静的に書き込み保護された、ユーザー作成のカスタマ セキュア コード (CSC)。BCR の後に実行され、アプリケーションへジャンプする前に、追加のポリシーを適用し、またはアプリケーションの真正性および完全性を検証します
3. TrustZone アーキテクチャによって分割された信頼済み実行環境のアプリケーション。第一段階と第二段階のポリシー設定は、アプリケーションに関連する保護も適用されます。

ファイアウォールなどの追加のセキュリティ ポリシーを適用できるのは CSC だけですが、第二段階はすべての使用事例で必要とされるわけではありません。一部の MSPM33 デバイスは、アプリケーション認証用の第一段階で ROM ベースのセキュア ブートをサポートしています。アプリケーション認証については、「セキュア ブート」セクションで詳しく説明するため、本章では追加ポリシーの適用に絞って説明し、第二段階を CSC と呼ぶことにします。

すべての MSPM33 デバイスが TrustZone をサポートしているわけではありません。非 TrustZone デバイスでは、他のセキュリティ ポリシーを適用できますが、セキュア / 非セキュア パーティション化は使用できません。TrustZone の構成については、「信頼済み実行環境」セクションで詳しく説明します。

このセクションでは、デバイスのブート プロセスの概要と、幅広い使用事例を可能にするために設定できる両カテゴリのユーザー指定ポリシーについて説明します。

2.1 デバイス ID

すべての MSPM33 デバイスには、96 ビットのユニット固有のデバイス ID が含まれており、アプリケーション ソフトウェアで読み取ることができます。デバイス ID の詳細については、テクニカル リファレンス マニュアルとデバイスのデータシートを参照してください。

MSPM33 デバイスは 128 ビットの SOC ID も提供しています。この ID は、ウェハー番号、ロット番号、デバイスの X 座標と Y 座標からチップ上で計算され、デバイスごとに一意です。SOC ID にアクセスできるのは ROM コードのみです。ユーザーは SWD ポートを使用して、DSSM コマンド (セクション 2.6 の DSSM_COMMAND_GET_SOC_ID) で SOC ID を読み取ることができます。

デバイス ID と SOC ID はどちらも TI によって出荷される各ユニット固有の値となるように設計されており、そのため特定のユニットを他のユニットと識別または区別するために使用できます。ID は固有ですが、一部のビットは部品番号や製品リビジョンなどのデバイス特性に対応しているため、暗号論的にランダムではありません。

表 2-1. MSPM33 のデバイス ID

	デバイス ID	SOC ID
ID	32 ビットのトレース ID + 32 ビットのデバイス ID + 32 ビットのユーザー ID を含む 96 ビットの値	128 ビット値
アクセス権	ユーザー アプリケーションを読み取ることが可能	SOC ID にアクセスできるのは ROM のみです。ユーザーは DSSM コマンドで SOC ID を読み取ることができます。

2.2 ブート時の初期条件

コールド パワーアップ (POR) 中、デバイスはセキュア状態にリセットされます。デジタル IO ピンは高インピーダンスとなり、すべてのペリフェラル機能が切断され、NRST ピンは NRST モード、シリアル ワイヤ デバッグ (SWD) インターフェイス ピンは SWD モードになります。ブラウンアウト リセット (BOR) の解放後、シリアル ワイヤ デバッグ ポート (SW-DP) が最初に有効になり、デバッグ プローブからデバッグ サブシステムへの初期接続が確立されます。

ブート プロセスのこの時点で、デバッグ プローブからアクセス可能なデバッグ アクセス ポート (DAP) は、構成アクセス ポイント (CFG-AP) とセキュア アクセス ポイント (SEC-AP) のみです。CFG-AP は、接続されたデバッグ プローブで汎用デバイス情報 (デバイスの汎用部品番号など) を読み取るために使用できます。SEC-AP は、ブート構成ルーチンにコマンド メッセージを渡すために使用できます。デバイスへのアプリケーション デバッグ アクセス (AHB-AP、ET-AP、および PWR-AP DAP 経由) は、ハードウェア ファイアウォールによってブロックされたままです。そのため、デバイスの電源投入

Copyright © 2026 Texas Instruments Incorporated

時には、デバイス ハードウェアでプロセッサ、EnergyTrace 状態、または電源構成へのデバッグ アクセスは許可されません。

ブラウンアウト リセット (BOR) 後は、ブート リセット (BOOTRST) が生成され、ブート構成ルーチンの実行が開始されます。

2.3 ブート構成ルーチン (BCR)

MSPM33 デバイスには、読み取り専用メモリ (ROM) に変更不可能な信頼ルート ブート構成ルーチンが含まれています。ブート構成ルーチン (BCR) は、デバイスの BOOTRST に続いて Cortex-M33 プロセッサで実行される最初のコードです。BCR は、BSL エントリを認証するために必要なブートストラップ ローダ (BSL) のソフトウェア起動時にも実行されます。BCR の主な機能は次のとおりです。

1. 適切なデバイス動作に必要な TI の工場出荷時データを FACTORY フラッシュ メモリ領域からロジックにロードし、CRC-32 を使用して工場出荷時データ (デバイスのトリム データを含む) の整合性を検証
2. ユーザー指定のデバイス構成 (セキュリティ ポリシーを含む) を NONMAIN フラッシュ メモリ領域からロジックにロードし、CRC-32 を使用してユーザー構成データの整合性を検証。NONMAIN フラッシュの構成については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください
3. シリアル ワイヤ デバッグ (SWD) インターフェイス経由で送信されたブート コマンドを確認し、(該当する場合は) それらを認証して、(認証された場合は) 処理
4. ブートストラップ ローダ (BSL) がイネーブルになっている場合は BSL の呼び出し条件を確認し、有効な呼び出しが発生した場合に BSL を開始
5. セキュリティ保護されたデバイス タイプのライフサイクル検出、保守、および関連ポリシーのサポート。
   a. HS-FS (高度セキュリティ対応、フィールドでのセキュア化可能)
   b. HS-KP (高度セキュリティ対応、キー プロビジョニング)
   c. HS-SE (高度セキュリティ対応、コード プロビジョニング)
6. 顧客キーのインストールとライフサイクルの変換のためのセキュア キー プロビジョニングをサポート
7. (HSFS デバイス) ユーザー アプリケーションを起動する前に、CRC-32 または SHA-256 を使用して、ユーザー アプリケーション コードを含む MAIN フラッシュ メモリ領域のユーザー定義部分の整合性を確認
8. (HSSE デバイス) ECDSA_P256 を使用してユーザー アプリケーションを開始する前にアプリケーション コードを認証する ROM セキュア ブート。
9. BCR の終了時、CSC が INITDONE を発行した後 (詳細はセクション FIX 参照)、またはまったく行わない、のいずれかとして、AHP-AP、ET-AP、および PWR-AP の DAP を解放するかどうかを決定します。
10. (HSFS デバイス) 起動エラーを CFG-AP にログ記録
11. ブート時間を短縮できるように、PLL 構成をサポート
12. (HSFS デバイス) MAIN フラッシュ内の 0x0000.0000 からスタック ポインタを、0x0000.0004 からリセット ベクタをフェッチすることで、MAIN フラッシュ内の単一のエントリ ポイントから実行を開始するようにハードウェアをトリガ
13. (HSSE デバイス) ROM セキュア ブートに合格した後、認証されたアプリケーションのエントリ ポイントから実行を開始するためのハードウェアをトリガ

表 2-2. BCR のさまざまなライフサイクルにおける責任

BCR の責任	HSFS	HSKP	HSSE
TI の工場出荷時データを FACTORY からロード	✓	✓	✓
ユーザー指定の構成を NONMAIN からロード	✓	✓	✓
ブート コマンドをサポート	✓	✓	✓
ブートストラップ ローダ (BSL) 起動チェック	✓	✓	
ライフサイクルの検出と保守	✓	✓	✓
セキュア キー プロビジョニングのサポート	✓		
CRC32/SHA256 を使用したユーザー定義メモリの整合性チェック	✓		
ECDSA_P256 を使用した ROM セキュア ブート			✓
デバッグ機能での DAP の解放	✓	✓	✓
ブート エラーを CFG-AP に記録	✓		

Copyright © 2026 Texas Instruments Incorporated

表 2-2. BCR のさまざまなライフサイクルにおける責任 (続き)

BCR の責任	HSFS	HSKP	HSSE
ブート時間を短縮するための PLL 構成	✓	✓	✓
ハードウェアをトリガし、MAIN フラッシュ内の単一のエントリ ポイントから実行を開始	✓		
認証されたアプリケーションのエントリ ポイントから実行を開始するためのハードウェアをトリガ			✓

2.4 ブートストラップ ローダ (BSL)

MSPM33 デバイスには、読み取り専用メモリ (ROM) に変更不可能なブートストラップ ローダ (BSL) が含まれています。BSL では、シリアル ワイヤ デバッグ (SWD) インターフェイスではなく、標準のシリアル インターフェイス (UART、I2C、または CAN) を使用して、デバイス メモリの内容をプログラムおよび検証できます。

BSL は BCR でのみ起動できます。BCR は、有効な BSL 起動条件 (ソフトウェア起動、IO ピン起動、デバッグ メール ボックス コマンド起動) をチェックし、BSL が有効になっていることを確認してから BSL を開始します。BSL が終了すると、BCR が再実行され、現在のデバイス セキュリティ ポリシーがロードされて、ユーザー アプリケーションが起動します。

表 2-3 に、さまざまなライフサイクルの BSL 起動条件を示します。BSL は、HSSE デバイスではサポートされていません。

BSL は 256 ビットのユーザー指定パスワードで保護されており、BSL セッションを開始するときに、パスワードを UART、I2C、または CAN インターフェイス経由で BSL に渡す必要があります。BSL を使用しない場合は、無効にできます。

BSL の実装については、『MSPM33 ブートローダー ユーザー ガイド』も参照してください。

注

セクション 2.5 の HSSE は最高レベルのセキュリティを備えているため、HSSE デバイスは BSL をサポートしていません。

表 2-3. さまざまなライフサイクルの BSL 起動条件

BSL 起動条件	HSFS	HSKP	HSSE
ソフトウェア起動	✓		
IO ピンを起動	✓	✓	
メールボックスのデバッグ コマンド	✓		
プリブート アプリケーションの検証	✓		

2.5 ライフサイクル管理

MSPM33 C シリーズ デバイスは、TI の工場から出荷され、開発、量産、廃棄に至るまでのデバイス プロセスを管理するため、ライフサイクル管理をサポートしています。ライフサイクルは、デバイスの寿命のさまざまな段階で有効にする機能を制御します。

ライフサイクル管理により、ユーザーはすべてのオープンからすべてのクローズまでのセキュリティ レベルを管理された方法で向上させることができます。また、デバイスが故障したときや、故障解析中にデバイスを特定の状態に変換することで、情報の漏えいを防ぐことができます。

ブートコードは、デバイスの動作状態を示す 2 つの状態で SoC のライフサイクルに対応しています。

- HS (高セキュリティ) — 運用ユニット
  - HS ライフサイクル状態は、デバイスが動作する特定の構成に応じて複数の**デバイス サブタイプ**状態に分類できます。
- DCOM/BAD — 廃止または故障ユニット

HS ライフサイクル状態は、次の**デバイス サブタイプ**状態に分類されます。ブートコードは、対応するセキュリティ セクタを読み出して、高セキュリティのデバイス サブタイプを判定します。

Copyright © 2026 Texas Instruments Incorporated

- HS-FS (フィールドセキュア化可能) — TI から出荷されたデバイスはこの状態になっています。この状態では、デバイスはセキュリティ機能の有無にかかわらず使用できます。セキュリティ機能が不要な場合、デバイスは、ライフサイクルの残りの期間、この状態になります (デバイスが BAD 状態になるか、障害分析が実行される場合を除く)。セキュリティ機能が必要な場合、デバイスはこの状態でキー プロビジョニングを行う準備ができています。
- HS-KP (キー プロビジョニング) — キー (セキュア ブートに必要) がプロビジョニングされた後、デバイスはこの状態に設定されます。この状態では、デバイスは署名済みのカスタマ セキュア コード (CSC) または署名済みアプリケーションのいずれかを書き込む準備ができています。
- HS-SE (セキュリティ適用) — 署名済みの CSC またはアプリケーションがセキュア ブートによって書き込まれ、検証された後に遷移するデバイスの状態です。

上記の状態に加えて、デバイスで障害分析が有効になると、デバイスは HS-FA (障害分析) 状態に遷移します。

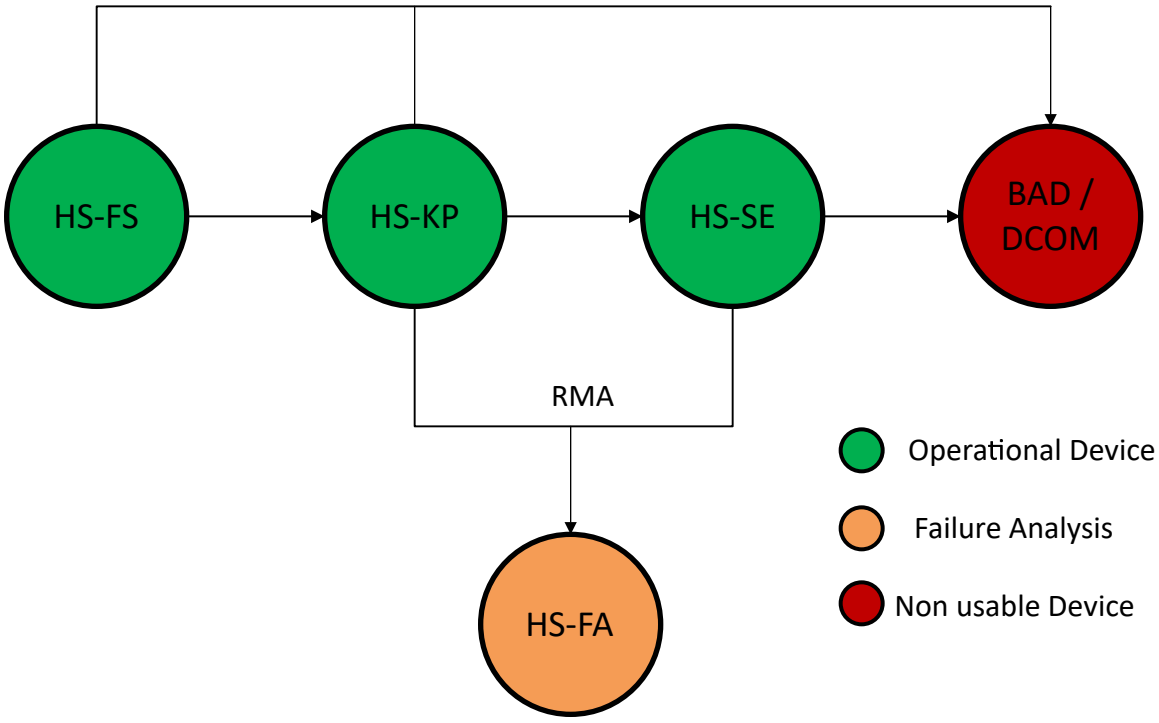

図 2-1. ライフサイクルの遷移

2.5.1 ライフサイクルの遷移

デバイスのライフサイクルでは、デバイスの現在の状態と適用されるセキュリティ設定に応じて状態が変化していきます。

以下に、HS-FS から HS-KP への遷移の一連のステップを示します。

- ユーザーがデバイスに対して DSSM コマンド (DSSM_COMMAND_SECURE_KEY_PROVISIONING) を実行します。
- 8 つの RoT 公開鍵 (セキュア ブートの場合は 6 つ、セキュア デバッグの場合は 2 つ) がデバイスに送信されます。
- ROM がこのデバイスを HS-KP としてマーキングします。

以下に、HS-KP から HS-SE への遷移ステップのシーケンスを示します。

- デバッグ ポートがロックされていない場合は、デバッグ ポートを使用してアプリケーション / CSC をプログラムできます。
- あるいは、BSL を呼び出してアプリケーションをプログラムします。
- BSL 経由で、署名済み CSC またはアプリケーションをプログラムします。
- ROM が CSC またはアプリケーションを検証すると、デバイスは HS-SE としてマークされます。

以下は、HS-KP/HS-SE から HS-FA への遷移ステップのシーケンスです。

- ユーザーがデバイスに対して DSSM コマンド (DSSM_COMMAND_RMA) を実行します。

Copyright © 2026 Texas Instruments Incorporated

- 結合された証明書が受信され、検証されると、ROM はデバイスを HS-FA としてマーキングします。
- デバイスは、HS-FA 認証フローに移行します

デバイスを DCOM/BAD に変換するには、DSSM コマンド (DSSM_COMMAND_CONVERT_BAD_DEVICE) を送信できます。

注

顧客返却フロー ユーザーは、故障解析のためにデバイスを TI に返却する前に、デバイスの状態を HS-KP/HS-SE から HS-FA に移行する必要があります。顧客は、さまざまな資産 (コード、鍵など) をプログラミング / 消去することで保護するオプションを選択できます。コンテンツを消去すると、一部の障害のデバッグが制限されることに注意してください (たとえば、フラッシュ データの保持に関する障害は、コンテンツが消去されるとデバッグできなくなります)。

2.6 ブート サービス

下表に、DSSM を使用したブート サービスを示します。BCR はこれらの DSSM コマンドを処理し、実行します。

DSSM 用の ROM では、さまざまなサービスがサポートされています。大まかに見ると、これらのサービスは次のように分類されます。

1. 認証なし — このシナリオでは、コマンドはオープンであり、認証は必要ありません。
2. パスワード ベースの認証 — 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 BCR/BSL 構成で設定できます。
3. セキュア サービス — HS-SE すべての DSSM サービスには証明書ベースの認証が必要です。

表 2-4. さまざまなライフサイクルに対応する DSSM サービス

コマンド	サービス	説明	HSFS	HSKP	HSSE	HSFA/BAD/DCOM
0x03	DSSM_WAIT_FOR_DEBUG		L0 — 認証なし L1、L2 — DEBUG_LOCK のパスワード (1)	無効	無効	無効
0x04	DSSM_INVOKE_BSL	BSL を起動します	認証なし	無効	無効	無効
0x05	DSSM_FACTORY_RESET	すべてのユーザー フラッシュ コンテンツを消去し、デバイスを工場出荷時の状態にリセットします	FACTORY_RESET に認証またはパスワードがありません (1)	無効	Mpuk6 および Mpuk7 のアクティブ DSSM キーによってセキュア サービスが検証されます (2)	無効
0x06	DSSM_MASS_ERASE	フラッシュ メモリ全体を消去します	MASS_ERASE に認証またはパスワードがありません (1)	無効	Mpuk6 および Mpuk7 のアクティブ DSSM キーによってセキュア サービスが検証されます (2)	無効
0x07	DSSM_COMMAND_DEBUG	証明書認証を使用して、デバイス上でセキュア デバッグ アクセスを開きます	L0 — 不要 L1、L2 — DEBUG_LOCK のパスワード (1)	無効	L0 — 不要 L1、L2 — Mpuk7 によってセキュア サービスが検証されます (2)(3)	無効
0x0A	DSSM_GET_LIFECYCLE	デバイスのライフサイクル状態を読み取ります	認証なし	認証なし	認証なし	認証なし

Copyright © 2026 Texas Instruments Incorporated

表 2-4. さまざまなライフサイクルに対応する DSSM サービス (続き)

コマンド	サービス	説明	HSFS	HSKP	HSSE	HSFA/BAD/DCOM
0x0B	DSSM_GET_DEVICE_ID	96 ビットのデバイス ID を読み取ります	L0 — 認証なし L1、L2 — DEBUG_LOCK のパスワード (1)	無効	L0 — 認証なし L1、L2 — Mpuk6 および Mpuk7 のアクティブ DSSM キーによってセキュア サービスが検証されます (2)	無効
0x0D	DSSM_COMMAND_RMA (CUSTOMER)	顧客 RMA 証明書認証要求です。	無効	無効	Mpuk6 および Mpuk7 のアクティブ DSSM キーによってセキュア サービスが検証されます (2)	無効
0x0E	DSSM_SECURE_KEY_PROVISIONING	キー プロビジョニング フローを開始します	認証なし	無効	無効	無効
0x0F	DSSM_SECURE_KEY_SERVICES	暗号鍵を失効させます	無効	無効	Mpuk6 および Mpuk7 のアクティブ DSSM キーによってセキュア サービスが検証されます (2)	無効
0x12	DSSM_COMMAND_DEBUG_NS	証明書認証を使用して、デバイス上で非セキュア デバッグ アクセスを開きます	L0、L1 — 不要 L2 — DEBUG_NS_LOCK のパスワード(1)	無効	L0、L1 — 不要 L2 — Mpuk6 によってセキュア サービスが検証されます(2) (3)	無効
0x15	DSSM_CONVERT_BAD_DEVICE	デバイスの状態を BAD/DECOM に変換します	BAD_CONV のパスワード(1)	無効	Mpuk6 および Mpuk7 のアクティブ DSSM キーによってセキュア サービスが検証されます (2)	無効
0x16	DSSM_COMMAND_GET_SOC_ID	128 ビットの SOC ID を読み取ります	認証なし	無効	認証なし	無効

(1) debug_lock、debug_ns_lock、factory_reset、mass_erase、bad_conv はすべて、パスワード設定用のメイン以外のフラッシュのレジスタです。
(2) mpuk6 および mpuk7 は、otp 領域でプロビジョニングされた顧客キーであり、dssm 証明書の検証に使用されます。
(3) mpuk6 が無効化されている場合、ns デバッグのロック解除は使用できません。ただし、s デバッグのロック解除は常に利用可能であり、mpuk7 を使用します。

2.7 ブート フロー

図 2-2 に、MSPM33 デバイスの大まかなブート フローを示します。

BOOTRST で、TI ブートコードの実行が開始されます。ライフサイクル ステータスに従い、デバイスは対応するプロセスをセキュア ブート、通常のブート、再テストフロー、またはその他のプロセスとして実行します。

Copyright © 2026 Texas Instruments Incorporated

ブートローダー (BSL) 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。ブートローダーにより、標準のシリアル インターフェースを使用してデバイスのフラッシュ メモリのプログラムと検証が可能になります。

ブートコードの実行中、デバッグ サブシステム メールボックス (DSSM) により、デバッグ プローブが SWD インターフェイス経由でターゲット デバイスにメッセージを渡し、ターゲット デバイスはデバッグ プローブにデータを返します。ブートコードでは、DSSM により、ライフサイクル状態変更要求や、パスワードで保護されたデバッグ、一括消去、工場出荷時のリセット動作を行うためのデバッグ プローブの認証などのコマンドをデバイスに送信することができます。

正常にブートが完了すると、ブートコードは BOOTDONE を発行します。この時点で、SYSCTL はデバイスに SYSRST を発行し、フラッシュ メモリからの実行をトリガします。ブート構成レコードに応じて、これにより、メインアプリケーションの起動 (この構成で CSC が存在しない場合) または CSC の起動 (CSC が設定されている場合) のいずれかに進みます。

CSC (カスタマ セキュア コード) はフラッシュ メモリから実行され、追加のセキュリティ要素を設定します。CSC は、ユーザーが所有するセキュアソフトウェアであることに注意してください。この 2 ステップのセキュア ブート コンセプトにより、アプリケーション固有のポリシーの実装と管理を CSC で行うことができます。一方、標準的なセキュリティ イネーブラはブートコードで実装されます。CSC の実行が完了すると、システムは 2 番目の SYSRST を実行し、その後メイン アプリケーションが開始されます。

図 2-2 に、BOOTRST からメイン アプリケーションまでの大まかなブート フローを示します。

Copyright © 2026 Texas Instruments Incorporated

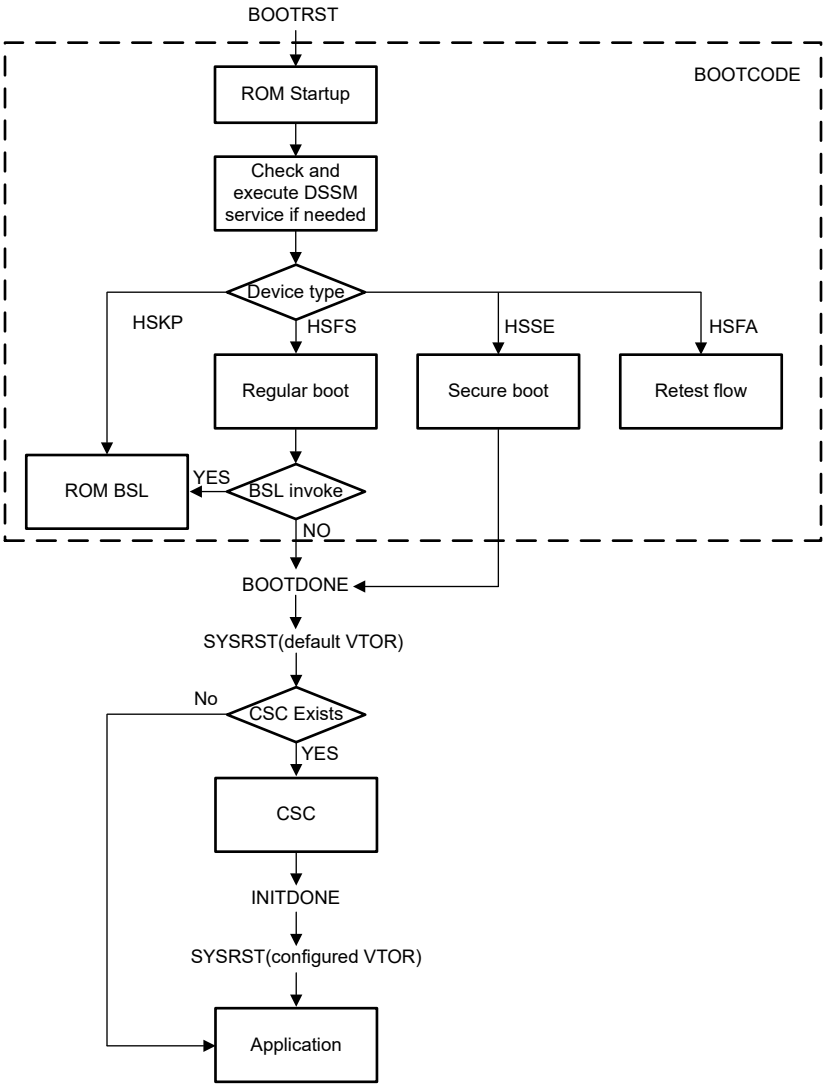

図 2-2. 大まかなブート フロー

注

セキュリティが有効化されていないデバイスとのブート フロー互換性を保つため、ブート構成のデフォルト設定は「CSC が存在しない」状態に設定されています。

セキュアな実行フローは、CSC_EXISTS = YES のパスです。この場合、BOOTRST の後、メイン アプリケーションが起動する前に 2 つの SYSRST が発行されることがあります。デフォルトの VTOR を使用した最初の SYSRST の後、顧客のスタートアップ コードが実行されます。セキュリティ ポリシーを構成し、VTOR をメイン アプリケーションに設定し、INITDONE を発行します。この時点で、セキュリティ設定はロックされ、強制されます。この時点で、VTOR が構成された 2 番目の SYSRST が発行され、メイン アプリケーションが起動します。

セクション 2.3 と セクション 2.4 のどちらにも、ロック可能な NONMAIN フラッシュ メモリ領域にユーザー指定の構成データ構造が格納されています。これらのデータ構造で指定されているセキュリティ ポリシーについては、セクション 2.8 を参照してください。

Copyright © 2026 Texas Instruments Incorporated

2.8 ユーザー指定のセキュリティ ポリシー

MSPM33 デバイスには、ユーザー指定のセキュリティおよびデバイス構成ポリシーを保存するための専用のフラッシュ メモリ領域があります。この領域を NONMAIN フラッシュ領域と呼びます。ブート構成ルーチン (BCR) およびブートストラップ ローダ (BSL) は、デバイスを動作用に構成するため、NONMAIN フラッシュ領域に保存されているユーザー指定のデータを参照します。

製造時に、デバイスの NONMAIN フラッシュ メモリ領域に必要なポリシーをプロビジョニングする必要があります。このセクションでは、NONMAIN 構成メモリを使用して構成可能なセキュリティ ポリシーについて説明します。

NONMAIN フラッシュ領域は、次の 2 つの異なるデータ構造に分割されています。

- BCR 構成 (セクション 2.8.1 を参照):ブート構成のセキュリティ ポリシーを設定
- BSL 構成 (セクション 2.8.2 を参照):ブートローダのセキュリティ ポリシーを設定

構成の詳細については、ユーザーは『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』も参照することができます。

どちらのデータ構造も、セクション 2.8.4 方式の一部として使用される独自の 32 ビット CRC ダイジェストによって支えられています。

注

BCR および BSL 構成構造には、このドキュメントに記載されているパラメータ以外のパラメータも含まれています。このドキュメントでは、セキュリティに関連するパラメータに焦点を当てています。NONMAIN フラッシュ メモリ領域における BCR および BSL 構成構造の詳細については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』または該当するテクニカル リファレンス マニュアルのアーキテクチャ章の「ブート構成」セクションを参照してください。

2.8.1 ブート構成ルーチン (BCR) ポリシー

以下のセクションでは、フラッシュの NONMAIN 領域に静的に定義され、ブート構成ルーチン (BCR) によって適用されるポリシーについて説明します。特に記述のない限り、すべての MSPM33 デバイスは、これらのポリシーを強制できます。

2.8.1.1 シリアル ワイヤ デバッグ関連のポリシー

シリアル ワイヤ デバッグ関連のポリシーは、デバイスの物理的デバッグ インターフェイス (SWD) 経由で利用できる機能を構成します。デフォルトでは、MSPM33 デバイスは テキサス・インスツルメンツから無制限の状態で出荷されます。この状態により、製造時のプログラミング、評価、開発が容易になります。ただし、この無制限の状態では、大きな攻撃対象領域が存在したままになるため、量産では推奨されません。構成プロセスをシンプルに保ちながら各種のニーズに対応するため、MSPM33 デバイスは、制限なし、カスタム制限、完全制限という 3 つの汎用セキュリティ レベルをサポートしています。表 2-5 に、最も制限の少ないものから最も制限の厳しいものの順に、3 つの汎用セキュリティ レベルを示します。

SWD インターフェイスには、保護することを考慮する必要がある 3 つの主な用途があります。

- アプリケーション デバッグ アクセス。以下のものが含まれます。
  - AHB-AP 経由でプロセッサ、メモリ マップ、ペリフェラルにフル アクセス
  - ET-AP 経由でデバイス EnergyTrace+ 状態情報にアクセス
  - PWR-AP 経由でデバッグを行うためのデバイス電源状態制御へのアクセス
- 以下を含む一括消去アクセス:
  - SWD 経由でコマンドを送信して MAIN および DATA メモリ領域を消去し、NONMAIN デバイス構成メモリはそのまま保持する機能
- 以下を含む工場出荷時リセットへのアクセス:
  - SWD 経由でコマンドを送信して MAIN および DATA メモリ領域を消去し、NONMAIN デバイス構成メモリを TI 工場出荷時デフォルト (制限なし) にリセットする機能

表 2-5. 汎用セキュリティ レベル

シナリオ	SW-DP アクセスポリシー	アプリケーション デバッグ ポリシー	一括消去ポリシー	工場出荷時のリセット ポリシー
制限なし	EN	L0 — フル アクセス	EN	EN

Copyright © 2026 Texas Instruments Incorporated

表 2-5. 汎用セキュリティ レベル (続き)

シナリオ	SW-DP アクセスポリシー	アプリケーション デバッグ ポリシー	一括消去ポリシー	工場出荷時のリセット ポリシー
カスタム制限	EN	L0 — フル アクセス L1 — PW による EN セキュア、EN 非セキュア L2 — PW による EN セキュア、PW による EN 非セキュア DIS — ディスエーブル	イネーブル、パスワードでイネーブル、ディセーブル	イネーブル、パスワードでイネーブル、ディセーブル
完全制限	DIS	関係なし(1) (SW-DP が無効な場合はアクセス不可)		

(1) SW-DP ポリシーが **SW-DP ディスエーブル**の場合、SWD インターフェイスの観点からは、一括消去と工場出荷時リセットのポリシーは関係ありません。ただし、ブートストラップ ローダ (BSL) がイネーブルになっている場合、一括消去および工場出荷時リセットのポリシーは、BSL を使用して利用可能な機能に影響を及ぼします。BSL のセキュリティ保護の詳細については、BSL セキュリティのセクションを参照してください。

2.8.1.1.1 制限なしのシナリオ

「制限なし」は、SWD セキュリティ状態の中で最も制限の少ない状態です。これは テキサス・インスツルメンツの新しいデバイスのデフォルト状態であり、工場出荷時リセットが成功した後のデバイスの状態でもあります。この状態で故障解析を行うため、アプリケーションのデバッグ アクセス、一括消去、工場出荷時リセットに制限はありません。

「制限なし」は、デバイス メモリのプログラミングや、プロセッサとペリフェラルのデバッグが可能なため、プロトタイプ作成や開発に最適です。

量産環境では、「制限なし」を使用しないでください。攻撃者が自由にデバイス メモリの内容を読み取り、デバイスの実行を操作し、(フラッシュ メモリの書き込み保護方式に応じて) フラッシュ メモリの内容を変更できてしまいます。

2.8.1.1.2 カスタム制限シナリオ

「カスタム制限」では、カスタマイズされたセキュリティ構成を行うことができます。物理的デバッグ ポート (SW-DP) は有効のままにし、各機能 (アプリケーション デバッグ、一括消去コマンド、工場出荷時リセット コマンド) は個別に、有効化、無効化、または (場合によって) パスワード認証を使用して有効化でき、特定の使用事例に合わせてデバイスの動作をカスタマイズすることが可能です。

TrustZone をサポートするデバイスでは、「カスタム制限」でセキュア デバッグと非セキュア デバッグで個別のパスワード認証を使用できます。一般的に、セキュア デバッグはデバイス上のすべてのリソースにアクセスでき、非セキュア デバッグはデバイスの非セキュア領域のみアクセスできます。

カスタム制限は、制限されたプロトタイプ作成 / 開発シナリオや、特定の SWD 機能 (工場出荷時リセット、TI 故障解析など) を保持しながら、アプリケーション デバッグなどの機能を無効化する量産シナリオに最適です。表 2-6 に、カスタマイズ構成の一般的な例を示します。

表 2-6.「カスタム制限」構成の例

カスタム制限シナリオ	構成		
	アプリケーションのデバッグ	一括消去	工場出荷時リセット
このシナリオでは、ユーザー指定のパスワードを使用してセキュア デバッグ アクセスを制限しますが、非セキュア デバッグ アクセスと工場出荷時リセットはそのまま残ります。この構成では、セキュア領域へのアクセスを遮断して保護しつつ、非セキュア デバッグが非セキュア領域にアクセスできるようになります。このシナリオでは、パスワードによる安全なデバッグが可能になり、また工場出荷時リセットによってデバイスをデフォルトの「制限なし」状態に戻すこともできます。	PW による EN セキュア、EN 非セキュア	DIS	EN
このシナリオでは、セキュア デバッグ アクセスと、別々のユーザー指定パスワードを使用した非セキュア デバッグ アクセスを制限しますが、工場出荷時リセットは維持されます。この構成では、セキュア デバッグ (パスワードを使用) または非セキュア デバッグ (パスワードを使用) が可能になり、さらに工場出荷時リセットによってデバイスをデフォルトの「制限なし」状態に戻すこともできます。	PW による EN セキュア、PW による EN 非セキュア	DIS	EN

Copyright © 2026 Texas Instruments Incorporated

表 2-6.「カスタム制限」構成の例 (続き)

カスタム制限シナリオ	構成		
	アプリケーションのデバッグ	一括消去	工場出荷時リセット
このシナリオではデバッグは不可。工場出荷時リセットは可能だが、ユーザー指定のパスワードが必要。これにより、MAIN メモリの内容をクリアし、パスワードがわかっている場合はデバイスを「制限なし」状態に戻すことにより、現場でデバイスを開けることが可能となります。工場出荷時リセット用のパスワードが漏洩した場合でも、攻撃者は MAIN フラッシュ メモリ内の独自の情報を読み取ることは不可。	DIS	DIS	パスワードでイネーブル

注

「制限なし」は、ほとんどの標準的な量産使用事例で推奨される構成です。セキュア ブートを必要としないアプリケーションの場合、TI は、工場出荷時リセットを有効にしたまま (パスワードを使用)「カスタム制限」を量産環境で使用することを推奨します。このような構成では、ユーザー (パスワードを使用) または テキサス・インスツルメンツ (障害解析の返却フロー) のいずれかによるプロビジョニング後に、デバイスをより制限の少ない状態に回復できます。最大限のセキュア ブート保護を必要とする使用事例では、量産環境でより制限の厳しい「カスタム制限」または「完全制限」を使用できます。ただし、デバイスをプロビジョニングした後は、より制限の低い状態に回復できない可能性があるというトレードオフがあります。

デバイスへの完全なアクセスが必要な場合は、プロトタイプ作成中に「カスタム制限」を使用せず、「制限なし」を使用してください。

また、最大限の制限が必要で、SWD 機能を有効化しない量産シナリオでも、「カスタム制限」を使用しないでください。このような場合は「完全制限」を使用して、SWD 物理インターフェイス全体を直接無効化し、誤設定の可能性を最小限に抑える必要があります。

注

アプリケーション デバッグと工場出荷時リセットをでディセーブルしてデバイスを構成した場合、ユーザーがデバイスへのデバッグ アクセスを回復する唯一の方法は、ユーザー アプリケーション コードで NONMAIN 構成をより制限の少ない状態に変更するメカニズムを提供することです。NONMAIN が静的な書き込み保護によってロックされている場合は、状態を変更できず、ユーザーがデバッグ アクセスを回復することはできません。

2.8.1.1.3 完全制限シナリオ

SWD の「完全制限」では、デバイスは最大制限状態に構成されます。物理デバッグ ポート (SW-DP) は完全に無効になり、SWD でアクセス可能なすべての機能 (アプリケーションのデバッグ、一括消去、工場出荷時リセット) には、個別の構成に関係なく、SWD からアクセスできません。

「完全制限」が選択されている場合 (SW-DP ディスエーブル)、アプリケーションのデバッグ構成は無視できるフィールドとなり、デバイス構成には影響しません。

BSL がディセーブルの場合、一括消去および工場出荷時リセットの構成フィールドも無関係なフィールドになります。BSL がイネーブルの場合は、BSL インターフェイスから送信される一括消去または出荷時リセットのコマンドを認証するため、BSL は引き続き一括消去および工場出荷時リセットの構成フィールドを使用します。

「完全制限」は、SWD 機能へのいかなる追加アクセスも不要で、デバイスを最大限にセキュアな状態にする量産環境でのみ使用します。

次の場合は、「完全制限」を使用しないでください。

- 将来的に、SWD を介したアプリケーションのデバッグまたは再プログラミングが必要です
- SWD を介して一括消去または工場出荷時リセット コマンドを送信し、フラッシュ メモリから機密情報を削除します

Copyright © 2026 Texas Instruments Incorporated

注

デバイスを「完全制限」(SW-DP ディスエーブル) で構成した後は、SWD 経由でデバイスにアクセスすることは**できなくなります**。デバイスを SWD アクセスが可能な「制限なし」または「カスタム制限」の状態に戻すことができるのは、BSL と工場出荷時リセットの両方が有効 (BSL で工場出荷時リセット コマンドを送信可能) になっている場合か、またはユーザー アプリケーション コードに含まれるメカニズムにより、NONMAIN 構成を制限の少ない状態に変更できる場合のみです。いずれの場合も、NONMAIN が静的な書き込み保護によってロックされている場合は、「完全制限」状態から変更できず、SWD アクセスを回復することはできません。

2.8.1.2 ブートストラップ ローダ (BSL) のイネーブル / ディセーブル ポリシー

ブートストラップ ローダ (BSL) では、シリアル ワイヤ デバッグ インターフェイスではなく、標準のシリアル インターフェイス (UART、I2C、または CAN) を使用して、デバイス メモリをプログラムおよび検証できます。BSL には独自の構成ポリシーがありますが、BSL の起動をイネーブルにするか、ディセーブル (起動不可) にするかは、BCR が決定します。

BSL には追加の攻撃対象領域が存在するため、アプリケーションで BSL を使用しない場合は、ユーザー指定のブート セキュリティ ポリシーでディセーブルにできます。BSL をアプリケーションで使用する場合、BSL セキュリティ設定 (BSL アクセス用のパスワードを含む) は セクション 2.8.2 で管理されます。

2.8.1.3 フラッシュ メモリの保護と整合性ポリシー

フラッシュ メモリの保護および整合性ポリシーは、フラッシュ メモリのどのセクタが変更からロックされるか、どのセクタがセキュリティ保護に設定されるか、どのセクタが特権保護に設定されるかを指定します。保護には、静的書き込み消去保護、セキュリティ保護、特権保護の 3 種類があります。また、これらのポリシーでは、ユーザー アプリケーションを開始する前にブート プロセス中に整合性をチェックするセクタも指定します。

2.8.1.3.1 アプリケーション (MAIN) フラッシュ メモリのロック

MSPM33 マイコンは、静的書き込み保護方式 (GSC を使用) を実装しており、実行時に MAIN フラッシュ領域のユーザー定義セクタをプログラムまたは消去操作からロックアウトします。目的の静的書き込み保護方式は、NONMAIN フラッシュ領域のブート セキュリティ ポリシーの一部として構成されます。ブート プロセス中、ROM ブートコードは、定義されたセクタに書き込み消去保護を適用するように GSC を設定します。

注

GSC は、ROM 内のブートコードまたは実行時のユーザー アプリケーションによって構成できます。ユーザーは、アプリケーション コード中の GSC の書き込み消去保護を構成することもできます。詳細については、セクション 5.1 を参照してください。

注

静的書き込み保護のための GSC レジスタのアクセス タイプは「R/W1S」です。「W1S」とは、レジスタ ビットを 0 ～ 1 の範囲でしか設定できず、レジスタは BOOTRST 以上のリセット タイプでのみリセット可能であることを意味します。つまり、適用された書き込み保護を直接解除することはできません。このため、ROM のブートコードで静的書き込み消去保護が設定されている場合、アプリケーションで削除することはできません。

静的書き込み保護により、次の特性を持つ固定のユーザー定義アプリケーションをフラッシュ メモリに配置できるようになります:

- 一度プログラムされてロックされると、そのアプリケーションはアプリケーション コードや ROM ブートローダーによって変更することはできません
- アプリケーションがフラッシュ メモリの先頭に配置されている場合、ROM ブート構成ルーチンが実行をユーザー アプリケーションに移した際に、このアプリケーションが最初に実行されるコードになります

MSPM33 静的書き込み保護は両方の特性をサポートしており、セキュア ブート イメージ マネージャを実装するにはこれらの特性を満たす必要があります。

NONMAIN のセクタが書き込みロックされている場合、ブート構成ルーチンがブートストラップ ローダまたは MAIN フラッシュ内のユーザー アプリケーション コードに実行を移行する場合、機能を変更することはできません。静的に保護された

Copyright © 2026 Texas Instruments Incorporated

セクタをアプリケーション コードまたはブートストラップ ローダでプログラムまたは消去しようとすると、ハードウェア フラッシュ動作エラーが発生し、セクタは変更されません。

静的書き込み保護により、アプリケーション コードやブートローダーからの変更は防止されますが、SWD インターフェイスを介して送信される一括消去や工場出荷時リセット コマンドは受け付けられます。この動作が望ましくない場合は、一括消去または工場出荷時リセットを行う SWD コマンドを固有のパスワードを使用して保護するか、無効化できます (セクション 2.8.1.1 を参照)。静的に書き込み保護された MAIN フラッシュ セクタを変更する手段を完全に削除するには、一括消去および工場出荷時リセット コマンド (または SW-DP) をディセーブルする必要があります。また、NONMAIN ブート構成メモリを静的に書き込み保護し、アプリケーション コードが NONMAIN 領域の内容を変更して下位の書き込み保護方式を変更するのを防止する必要があります。これについては、次のセクションで説明します。

2.8.1.3.2 構成 (NONMAIN) フラッシュ メモリのロック

MSPM33 マイコンには、静的書き込み保護機能が実装されており、NONMAIN フラッシュ領域を実行時のプログラム / 消去操作からロックアウトできます。書き込み保護機能は、NONMAIN フラッシュ領域のブート セキュリティ ポリシーの一部として構成されます。

デフォルトでは、NONMAIN 構成メモリ (ユーザー指定のブート セキュリティ ポリシーとブートストラップ ローダ ポリシーを含む) は書き込み保護の状態になっていません。これにより、プロビジョニング中にユーザーが NONMAIN を消去し、量産時に使用されるユーザー指定のポリシーで再プログラミングすることができます。

多くの場合、構成メモリはプロビジョニング後にロックするのが適切です。構成メモリをロックすると、セキュリティ ポリシー、ブートストラップ ローダ ポリシー、静的書き込み保護ポリシーが、ブートストラップ ローダまたはアプリケーション コードによって不正に変更されるのを防止できます。ほとんどのアプリケーションでは、量産デバイスの構成メモリの変更は、デバイスのファームウェアが更新された場合でも必要ありません。

保護するように構成した場合、NONMAIN 領域全体が書き込みロックされ、ブート構成ルーチンがブートストラップ ローダまたは MAIN フラッシュ内のユーザー アプリケーション コードに実行を移行したときに、機能を変更することはできません。NONMAIN 領域をアプリケーション コードまたはブートストラップ ローダでプログラムまたは消去しようとすると、ハードウェア フラッシュ動作エラーが発生し、セクタは変更されません。

静的書き込み保護は、アプリケーション コードまたはブートローダによる変更は防止しますが、SWD インターフェイス経由で送信される工場出荷時リセット コマンドは実行されます。この動作が望ましくない場合は、工場出荷時リセットを行う SWD コマンドを固有のパスワードを使用して保護するか、無効化できます (セクション 2.8.1.1 を参照)。NONMAIN 構成メモリを変更する手段を完全に削除するには、工場出荷時リセット コマンド (または SW-DP) を無効化する必要があります。

注

NONMAIN が静的に書き込み保護されており、工場出荷時リセット コマンド (または SW-DP) が無効の場合、NONMAIN は変更不可能な読み取り専用メモリと同等であり、いかなる方法でもデバイス構成を変更することはできません。さらに、MAIN メモリ領域セクタのいずれかが静的保護機能を使用して構成されている場合、これらのセクタはいかなる方法でも変更できず、変更不可能とみなすことができます。

2.8.1.3.3 アプリケーション (MAIN、DATA) フラッシュ メモリのセキュリティと特権保護

TrustZone と MPU をサポートする MSPM33 マイコンの場合、ユーザーは MAIN フラッシュまたは DATA フラッシュのセキュリティ属性と特権属性を NONMAIN フラッシュ領域に構成でき、ROM ブートコードは GSC を設定して、定義されたセクタにセキュリティと特権保護を適用します。セキュア領域のコードおよびデータは、非セキュア領域から保護されます。特権領域のコードおよびデータは、非特権領域から保護されます。

構成の詳細については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

注

MSPM33 上のセキュアおよび非セキュア パーティションの場合、final secure 属性は GSC 構成だけでなく SAU および IDAU の設定にも依存します。構成の詳細については、セクション 4 または『MSPM33 マイコンにおけるセキュア アプリケーション用 Trustzone のイネーブル』を参照してください。

Copyright © 2026 Texas Instruments Incorporated

注

GSC は、ROM 内のブートコードまたは実行時のユーザー アプリケーションによって構成できます。ユーザーは、いずれかの方法を選択して、GSC のセキュリティ / 特権属性を構成できます。

2.8.1.3.4 アプリケーション (MAIN) フラッシュ メモリの整合性の検証

BCR は、BCR (ROM 内) からユーザー アプリケーション (MAIN フラッシュ メモリ内) に実行を移行する前に、MAIN フラッシュ メモリ内のユーザー指定アドレス範囲のデータ整合性をチェックすることをサポートしています。

整合性チェックは、ブート ROM (通常はセキュア ブート イメージ マネージャ) の後に最初に実行されるコードに対して、予測される値と一致する CRC/SHA256 ダイジェストが含まれていることを検証するための追加ステップとして使用できます。この整合性チェックにより、フラッシュ メモリ内の重要なコード (残りのユーザー アプリケーション ソフトウェア イメージの認証を実行) が破損しているためにセキュリティの脆弱性が発生する可能性が低減されます。

開始アドレス、長さ、および ISO-3309 CRC-32 または SHA2-256 ダイジェストを、NONMAIN 構成メモリにプロビジョニングできます。ブート プロセス中に、BCR は MAIN フラッシュ メモリ内の指定された範囲の CRC-32 ダイジェストを計算し、計算されたダイジェストをプロビジョニングされた (予測される) ダイジェストと比較して検証します。値が一致している場合は、ユーザー アプリケーションが開始します。値が一致しない場合は、ユーザー アプリケーションは開始せず、重大なブート エラーが発生します。

構成の詳細については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

2.8.2 ブートストラップ ローダ (BSL) のセキュリティ ポリシー

BSL セキュリティ ポリシーは、ブートローダが呼び出されたときにブートローダーにより解釈され、次のパラメータが含まれます。

- BSL アクセス パスワード (セクション 2.8.2.1 を参照)。
- BSL 読み出しポリシー (セクション 2.8.2.2 を参照)。
- BSL セキュリティ アラート ポリシー (改ざん検出) (セクション 2.8.2.3 を参照)。

2.8.2.1 BSL アクセス パスワード

BSL へのアクセスは、ユーザーが指定した 256 ビットのパスワードで保護されます。パスワードをディセーブルするオプションはありません。ほとんどの BSL 機能にアクセスできるようにするには、BSL を呼び出した後にパスワードを入力する必要があります。パスワードを入力しない場合、使用できる BSL コマンドは Get Identity および Start Application のみです。

BSL に誤ったパスワードを供給すると、BSL は 2 秒間停止し、その後パスワードを入力し直すことができます。パスワード入力に 3 回失敗すると、セキュリティ アラート機能がアクティブになります (セクション 2.8.2.3 を参照)。

2.8.2.2 BSL 読み出しポリシー

BSL はオプションで、デバッグや診断の目的でデバイス メモリの読み出しをサポートできます (パスワードが一致して BSL にアクセスできるようになった後)。デフォルトでは、デバイスから機密コードやデータが抽出されるのを防止するため、この機能はセキュリティ上ディセーブルになっています。BSL 読み出しポリシーがディセーブルの場合、BSL インターフェイス経由でホストに送信できる情報は、最小セグメント長が 1KB であるメモリ セグメントの CRC32 ダイジェストのみです。デバイス メモリを直接読み出す必要がある場合は、BSL 構成でイネーブルにできます。

2.8.2.3 BSL セキュリティ アラート ポリシー

BSL には、改ざんの疑いがある場合に対処するためのアラート メカニズムがあります。具体的には、1 つの BSL セッション中に誤ったパスワードが 3 回 BSL に渡された場合、セキュリティ アラートがアクティブになり、BSL は指定されたセキュリティ アラート ポリシーに基づいて、次の 3 つの方法のいずれかで応答します。

1. 工場出荷時リセットの発行 (MAIN フラッシュを消去し、NONMAIN フラッシュ領域をリセット)
2. BSL をディセーブル (MAIN フラッシュはそのままにし、NONMAIN を再構成して BSL アクセスをブロック)
3. 無視 (構成は変更せず、パスワードの試行を継続的に許可)

Copyright © 2026 Texas Instruments Incorporated

注

オプション 1 および 2 を選択するには、NONMAIN フラッシュ領域が静的に書き込み保護されていることが必要です (セクション 2.8.1.3.2 を参照)。

オプション 1 を選択した場合、静的に書き込み保護されている MAIN メモリ領域 (セクション 2.8.1.3.1 を参照) は、工場出荷時リセット時に消去されません。

2.8.3 カスタマ セキュア コード (CSC) セキュリティ ポリシー

以下のセクションでは、対応デバイスにおいてカスタマ セキュア コード (CSC) によって強制されるポリシーについて説明します。すべてのポリシーは、BOOTRST から抜けた後に実行される CSC の処理中に設定され、デバイスで INITDONE 信号が発行される前にのみ変更できます。INITDONE が発行されて SYSRST が発生した後も、ポリシーは有効なままで、BOOTRST または POR まで変更できません。

2.8.3.1 CSC によるバンク スワップ強制

複数の MAIN フラッシュ バンクを備えるデバイスでは、システム内でアプリケーションを 2 つのバージョンでサポートでき、各バンクに 1 つずつ配置できます。この場合、CSC はイメージのバージョンおよび正当性に基づいて、いずれか一方を実行することを選択できます。決定プロセスについては、セクション 3.3 セクションで詳しく説明します。

CSC におけるバンク スワップの実装については、ユーザーは『MSPM33 マイコンにおけるカスタマ セキュア コードの評価』も参照できます。

2.8.3.2 KEYSTORE への CSC キー書き込み

AES キーは KEYSTORE に安全に格納できます。KEYSTORE は、INITDONE が発行される前の CSC 実行中にのみアクセス (読み出しおよび書き込み) できます。アプリケーション プログラムの実行中 (INITDONE 後) は、アプリケーションが AES エンジンにロードされる KEYSTORE のキーを制御できますが、ロード処理の全過程において、キーはアプリケーションからは可視ではありません。詳細については、セクション 5.3 を参照してください。

2.8.4 構成データのエラー耐性

MSPM33 デバイスには、NONMAIN 構成メモリにデータ エラーが発生してセキュリティが失われる可能性を削減するため、複数のメカニズムが備えられています。

2.8.4.1 CRC で保護された構成データ

NONMAIN メモリの BCR 構成データと BSL 構成データ構造には、それぞれの構造の CRC ダイジェストに対応する CRC 値が含まれています。デバイスのブート プロセス中に、BCR はデータ構造の CRC ダイジェストを計算し、格納されている CRC 値と比較して、構造体内に含まれるデータが信頼できるものかを確認します。

CRC の計算:

- 多項式は、特定の規格 (CRC32-ISO3309 または CRC16-CCITT) に従う必要があります。関連データシートを参照してください。
- 反映された入力
- 反映された出力
- 初期値は 0xFFFFFFFF である必要があります
- 最終的な XOR 値は 0x0 にする必要があります

2.8.4.1.1 BCR 構成の CRC エラーの処理

ブート中に BCR 構成データ (SWD ポリシー、BSL イネーブル / ディセーブル ポリシー、フラッシュ メモリ保護および整合性チェック ポリシーを含む) の CRC チェックがエラーとなった場合、致命的なブート エラーが発生し、以下の制限が課されます。

- エラーの原因をブート診断として CFG-AP に記録
- BSL はイネーブルに設定されていても起動しない
- ユーザー アプリケーションを開始しない
- アプリケーションのデバッグ アクセスはすべてディセーブル

Copyright © 2026 Texas Instruments Incorporated

- 保留中の SWD 工場出荷時リセット コマンドがイネーブルの場合、またはパスワードを使用してイネーブルになった場合は、それを実行
- 保留中の テキサス・インスツルメンツ故障解析フロー エントリがイネーブルの場合は適用
- ブート プロセスを最大 3 回再試行
  - 2 回目または 3 回目の試行で成功した場合、デバイスを通常どおり起動
  - 3 回目の試行に失敗した場合、次に BOR または POR が実行されるまで、それ以上のブート試行は許可しない

この CRC チェックの利点は、静的書き込み保護構成 (セキュア ブートの中核) などの構成データに反転ビットがある場合に、それをブート プロセス中に高い信頼性で検出できることです。エラー処理手順では、BSL およびユーザー アプリケーションの実行が明示的に防止され、サポートされているオプション (SWD 工場出荷時リセットおよび TI FA) のみが セクション 2.8.4.3 によって保護されます。

2.8.4.1.2 テキサス・インスツルメンツの工場出荷時トリム データの CRC エラー処理

ユーザーが指定した構成データに加えて、ブート中に テキサス・インスツルメンツの工場出荷時トリムの CRC チェックがエラーとなった場合も、致命的なブート エラーが発生し、以下の制限が発生します。

- エラーの原因をブート診断として CFG-AP に記録
- BSL はイネーブルに設定されていても起動しない
- ユーザー アプリケーションを開始しない
- アプリケーションのデバッグ アクセスはすべてディセーブル
- 保留中の テキサス・インスツルメンツ故障解析フロー エントリがイネーブルの場合は適用
- ブート プロセスを最大 3 回再試行
  - 2 回目または 3 回目の試行で成功した場合、デバイスを通常どおり起動
  - 3 回目の試行に失敗した場合、次に BOR または POR が実行されるまで、それ以上のブート試行は許可しない

2.8.4.2 BSL 構成の CRC エラーの処理

BSL 呼び出し中に BSL 構成データ (BSL パスワードおよび BSL ポリシーを含む) の CRC チェックがエラーとなった場合、致命的なブート エラーが発生し、以下の制限が課されます。

- このエラーの原因をブート診断として CFG-AP に記録
- BSL はイネーブルに設定されていても起動しない
- ユーザー アプリケーションを開始しない
- アプリケーションのデバッグ アクセスはすべてディセーブル
- ブート プロセスを最大 3 回再試行
  - 2 回目または 3 回目の試行で成功した場合、デバイスを通常どおり起動
  - 3 回目の試行に失敗した場合、次に BOR または POR が実行されるまで、それ以上のブート試行は許可しない

この CRC チェックの利点は、BSL 構成データ内に反転ビットがある場合に、それを起動プロセス中に高い信頼性で検出できることです。このエラー処理手順により、BSL が無効なデータから開始するのを防止できます。無効なデータから開始すると、セキュリティが失われる可能性があります。

2.8.4.3 クリティカル フィールドの 16 ビット パターン一致

SWD セキュリティ ポリシーなどの BCR 構成メモリ内の重要なポリシーは、NONMAIN メモリ内の 16 ビット パターン一致フィールドとして実装され、以下の特徴があります。

- より低いセキュリティ状態をイネーブルにするにはパターンが正確に一致することが必要
- 16 ビット フィールドのいずれかの値が定義されたパターンに正確に一致しない場合、対応するパラメータが最大のセキュア状態になる

これにより、シングル ビットの反転によって、デバイスが指定されていた状態より低いセキュリティ状態に移行してしまうことが防止されます。

Copyright © 2026 Texas Instruments Incorporated

3 セキュア ブート

MSPM33 デバイスは、ハードウェアとソフトウェアの機能を組み合わせることで、アプリケーション ソフトウェアの認証 (セキュア ブート) をサポートしており、マイコンは信頼できるメーカーによって明示的に検証され、改変されておらず、署名されたファームウェアのみを実行することを保証します。MSPM33 デバイスには、2 つのセキュア ブート方法があります。

1. ROM ベースのセキュア ブート
2. カスタマ セキュア コード (CSC) ベースのセキュア ブート

表 3-1 に、MSPM33 がサポートするセキュア ブートの機能を示します。ROM ベースのセキュア ブートは ROM に配置された変更不可能なセキュア ブートです。一方、CSC ベースのセキュア ブートは MAIN フラッシュに配置され、さまざまな保護機能を備えた変更可能なセキュア ブートです。

セクション 3.4 で説明されている 2 段階セキュア ブートは、信頼の連鎖を構築するための 2 つの方法を組み合わせることで、両方の方法の機能を実現します。

表 3-1. MSPM33 がサポートするセキュア ブート方式

	ROM ベースのセキュア ブート	CSC ベースのセキュア ブート
認証メカニズム	SHA-256 + キー ハッシュ + ECDSA_p256 (固定)	SHA-256 + キー ハッシュ + ECDSA_p256 キー ハッシュ + MLDSA 87 (ユーザーによって変更可能)
ライフサイクル状態	HSSE	HSFS/HSSE
セキュア ブートの場所	ROM	MAIN フラッシュの変更不可能領域
顧客キー ストレージ	NONMAIN フラッシュの OTP 領域	MAIN フラッシュの変更不可能領域
失敗時	ブート失敗	CSC 失敗
鍵失効	方法 1 — 鍵失効のための DSSM サービス 方法 2 — 現在のアクティブ +1 キーで署名された新しいイメージ	-
バンク スワップ	-	対応
イメージ形式	MCUBoot	MCUBoot

上記のセキュア ブート方法は、オープン ソースのセキュア ブートローダー方式である MCUBoot 形式に従います。TI はまた、MCUBoot 互換性のあるツールとソフトウェアの包括的なスイートを取り揃えており、開発プロセスをサポートしています。

3.1 MCUBoot

MCUBoot は、デバイスの起動プロセスを管理し、正規かつ認証済みで無傷のファームウェアのみが実行されることを検証するオープン ソースのセキュア ブートローダーです。図 3-1 に示されているように、どちらのセキュア ブート方式も MCUBoot イメージ形式に従います。

- **イメージ ヘッダー** — ヘッダー マジック (0x96F3B83D)、イメージ サイズ、イメージ バージョンなど、ファームウェア イメージに関する重要なメタデータを含む固定サイズ構造 (32 バイト)。イメージ ヘッダーは、アプリケーション割り込みベクタの前のアドレス 0x200 バイト (デフォルト) に配置されます。
- **イメージ ペイロード** — 元の符号なしアプリケーション コード
  - 割り込みベクタ
  - アプリケーション コード
- **イメージ TLV (Type-Length-Value)** — メタデータ、暗号ハッシュ、署名、およびその他の情報が含まれます。MCUBOOT は、イメージの末尾の後に配置され、TLV (Type-length-value) レコードを定義します。MSPM33 CSC で定義される TLV:TLV マジック (0x6907)、イメージ ハッシュ、ECDSA 公開鍵ハッシュ、ECDSA 署名、MLDSA 公開鍵ハッシュ、MLDSA 署名が含まれます。詳細については、mcuboot/docs/design.md at main · mcu-tools/mcuboot · GitHub を参照してください。
- **イメージ トレーラー** — 更新処理中、特にスワップ ベースの更新時に使用される情報が含まれています。

Copyright © 2026 Texas Instruments Incorporated

イメージ ヘッダー、イメージ TLV、イメージ トレーラー：MCUBOOT (<mspm33_sdk_path>\source\third_party\mcuboot\scripts にある Python スクリプトと共に配置) によって提供される署名ツール imgtool によって生成されます。これらのパーツは、CCS ビルドの後処理ステップで生成され、コンパイル済みのアプリケーション イメージにマージされます。MSPM33 SDK には、署名済みイメージが CCS 内でどのように構築されているかを示す customer_secure_sample_image の例が含まれています。

MCUBoot イメージ形式は、実際のアプリケーション コードを特定のメタデータでラップする構造化されたバイナリ形式です。このメタデータは重要です。これにより、アプリケーションの開始場所や検証方法を「推測」する必要なしに、セキュア ブートで更新プロセスを識別、検証、管理できるためです。

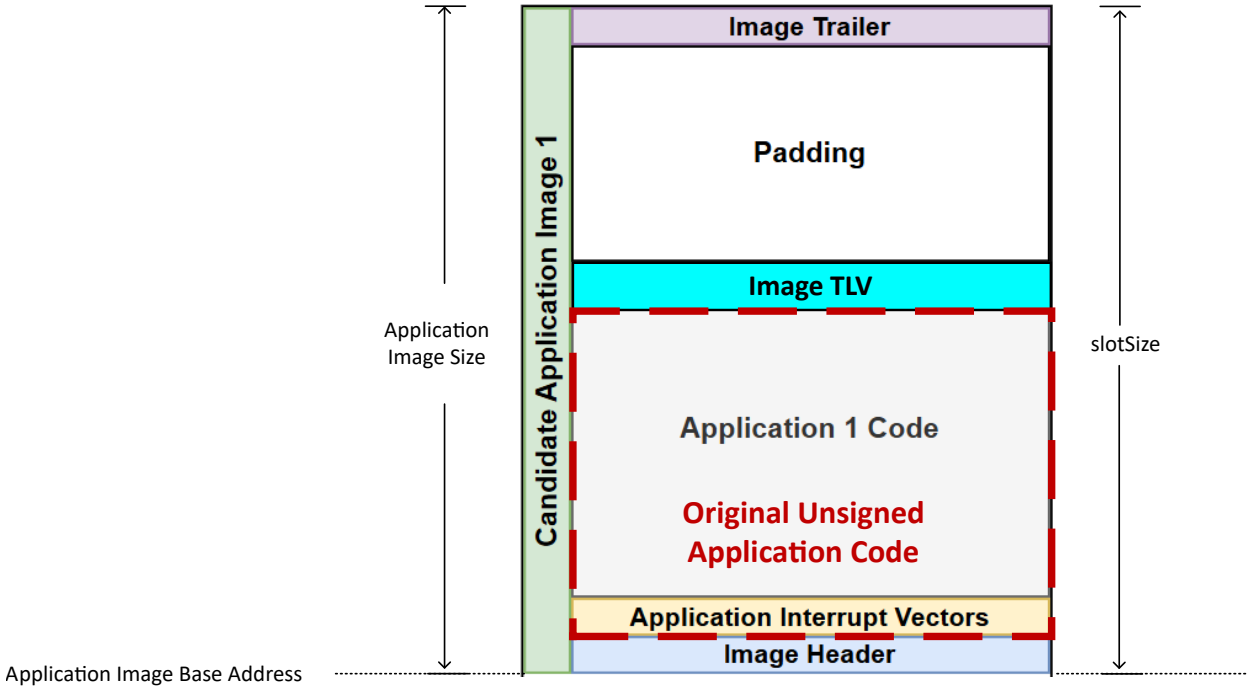

図 3-1. MCUBoot イメージ形式

図 3-2 に、TLV セクションでハッシュ、署名、およびその他のメタデータを生成する一般的なプロセスを示します。図 3-2 は、SDK に含まれている CSC ベースのセキュア ブートの例に関連しており、ECDSA_P256 と MLDSA-87 の両方をサポートしています。

イメージを MCUBoot 形式でビルドするには、MCUBoot が提供する imgtool を使用します。TI は SDK に imgtool を含めています。

Copyright © 2026 Texas Instruments Incorporated

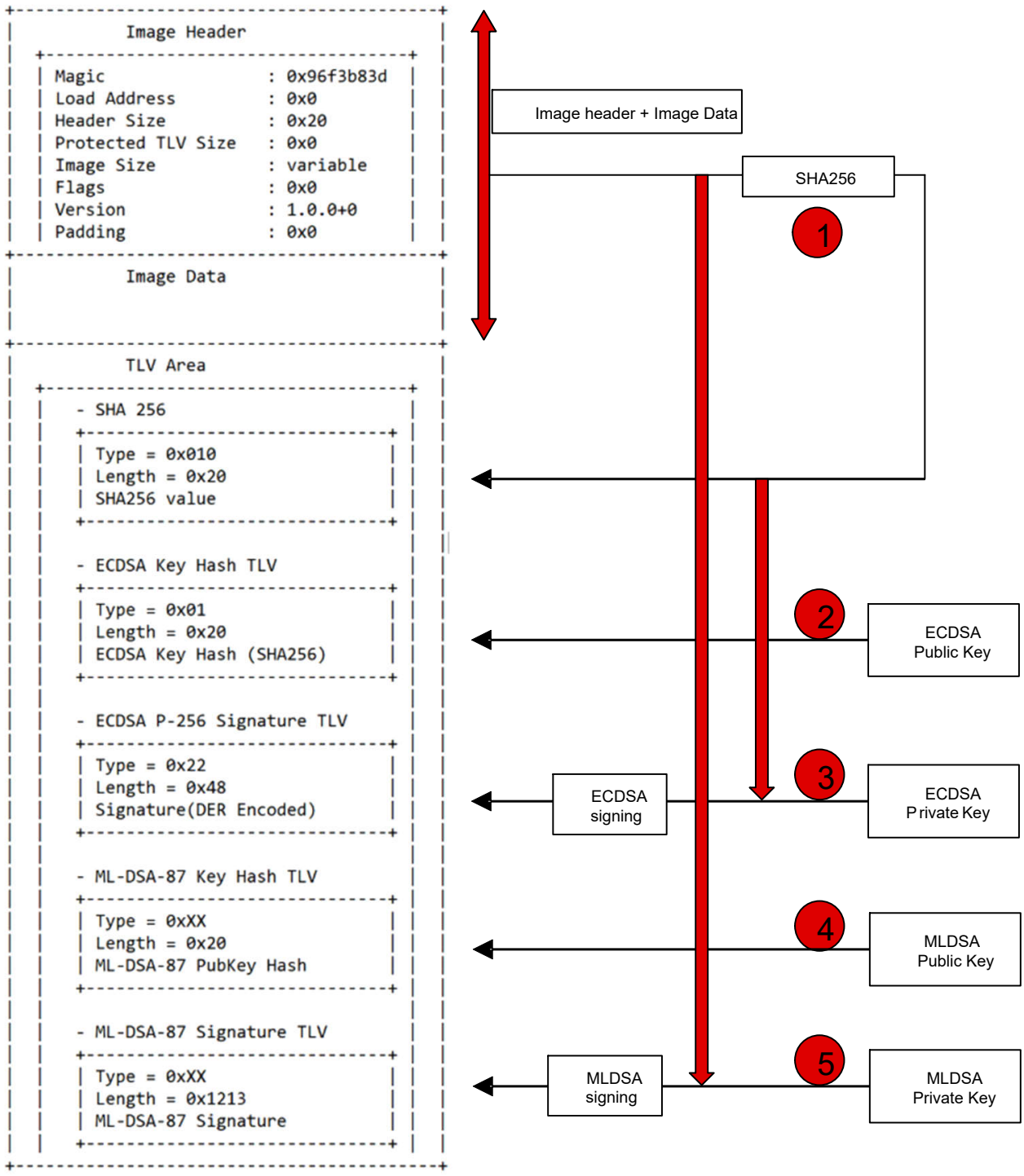

図 3-2. TLV セクションでメタデータを生成するプロセス

一般的なイメージ検証プロセスを次に示します。

1. イメージ マジックをチェックします
2. 有効な場合は、イメージのハッシュを計算します (イメージ ヘッダー + イメージ データ)
3. ハッシュ TLV を検索し、計算されたハッシュと比較します
4. 一致する場合は、キー TLV と署名 TLV を確認します

Copyright © 2026 Texas Instruments Incorporated

5. 見つかった場合は、公開鍵を使用して署名を検証します
6. セキュリティ カウンタが存在する場合は、現在の値よりも新しいことを確認します

3.2 ROM ベースのセキュア ブート

MSPM33 C シリーズ デバイスは、デバイスのライフサイクル管理を活用した ROM での変更不可能なセキュア ブートをサポートしています。変更不可のセキュア ブートは、プロビジョニング状態 (HS-FS) から永続的に保護された動作状態 (HS-SE) へ移行するデバイスのライフサイクル管理を利用して信頼の連鎖を確立し、デバイス上で認証および承認されたファームウェアを実行できるようにします。

セキュア ブートは、SHA-256 ハッシュを通してイメージの整合性を検証し、鍵ハッシュを使用して公開鍵の真正性を検証して、ECDSA 署名検証を実行するメカニズムを提供します。

ROM セキュア ブートは、以下の機能をサポートしています。

- ECDSA (楕円曲線デジタル署名アルゴリズム) P-256 曲線を使用したファームウェアの認証
- SHA-256 ハッシュ検証によるファームウェアの整合性
- セキュリティ上の脆弱性が存在する可能性のある古いファームウェアへのダウングレードを防ぐアンチロールバック
- 侵害されたキーを無効化するための鍵失効 — 最大 6 つの鍵をサポート
- セキュリティ メタデータ用の業界標準の MCUBoot トレーラー処理
- 起動シーケンスを最適化する高速セキュアブート モード

ファームウェア イメージ認証用の信頼ルートは、変更不可の OTP メモリに保存されている公開鍵に基づいています。セキュア ブートはこれらの鍵を使用して、対応する秘密鍵によって署名されたファームウェアの署名を検証します。MSPM33 C シリーズ デバイスは、最大 6 つの公開鍵をサポートしています。セキュア ブートは、検証のために最初の公開鍵を使用します。残りの公開鍵は、最初の鍵が失効した場合に使用されます。

セキュア ブートは、デバイスが HS-SE 状態のときにのみ実行されます。ブート プロセスについては、セクション 2.7 を参照してください。

3.2.1 ROM ベースのセキュア ブート フロー

MSPM33 C シリーズ デバイスの電源がオンになると、ROM コードが最初に実行されます。デバイスが HSSE のライフサイクル状態の場合、ROM ベースのセキュア ブートが実行され、MAIN フラッシュ内のイメージが認証されます。

MAIN フラッシュにスタンドアロン アプリケーションのみが存在する場合 (ユーザーが NONMAIN フラッシュの BOOTCFG2.FLASHBANKSWAPPOLICY フィールドを無効にする必要がある)、ROM ベースのセキュアブートは、図 3-3 に示されているように、物理アドレス 0x0 から始まる唯一のイメージをチェックして認証します。イメージでの認証が失敗するか、0x0 アドレスのデータがないと、ブートが失敗します。

MAIN フラッシュにデュアル アプリケーションが存在する場合 (ユーザーが NONMAIN フラッシュの BOOTCFG2.FLASHBANKSWAPPOLICY フィールドを有効にする必要がある)、図 3-4 に示されているように、ROM ベースのセキュア ブートはまずバンク 0 とバンク 1 の 2 つのイメージを確認し、まず上位バージョンでイメージを認証します。失敗した場合は、ROM ベースのセキュア ブートの後に別のイメージを検証します。両方のイメージで認証が失敗するか、バンク 0 およびバンク 1 の開始アドレスのデータがない場合、ブートが失敗します。

注

デュアル アプリが同じバージョンの場合、ROM ベースのセキュア ブートは、まずバンク 0 でアプリを検証します。

注

NONMAIN フラッシュの BOOTCFG2.FLASHBANKSWAPPOLICY フィールドはデフォルトで無効になっています。このフィールドを有効にすると、バンク スワップは CSC ベースのセキュア ブートに対してのみ有効になり、ROM ベースのセキュア ブートのバンク スワップはありません。ただし、このフィールドによっては、ROM ベースのセキュア ブートは、スタンドアロン アプリ (バンク スワップが無効の場合) またはデュアル アプリ (バンク スワップが有効の場合) をチェックおよび検証します。このフィールドを変更する方法については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

Copyright © 2026 Texas Instruments Incorporated

ROM ベースのセキュア ブートが成功した後、ブートコードは BOOTDONE を発行します。この時点で、デバイスに対して SYSRST が発行され、MAIN フラッシュ内の認証済みアプリケーションからの実行がトリガされます。

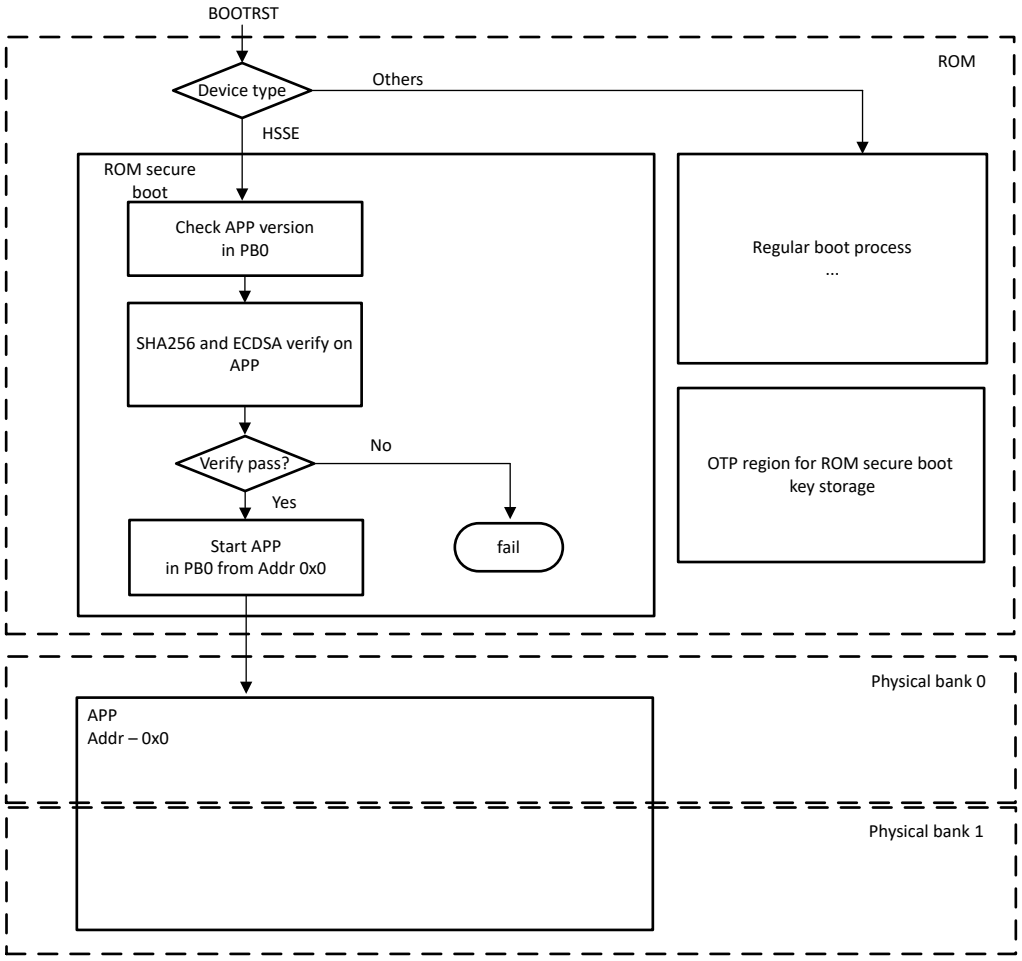

図 3-3. スタンドアロン アプリによる ROM ベースのセキュア ブート

Copyright © 2026 Texas Instruments Incorporated

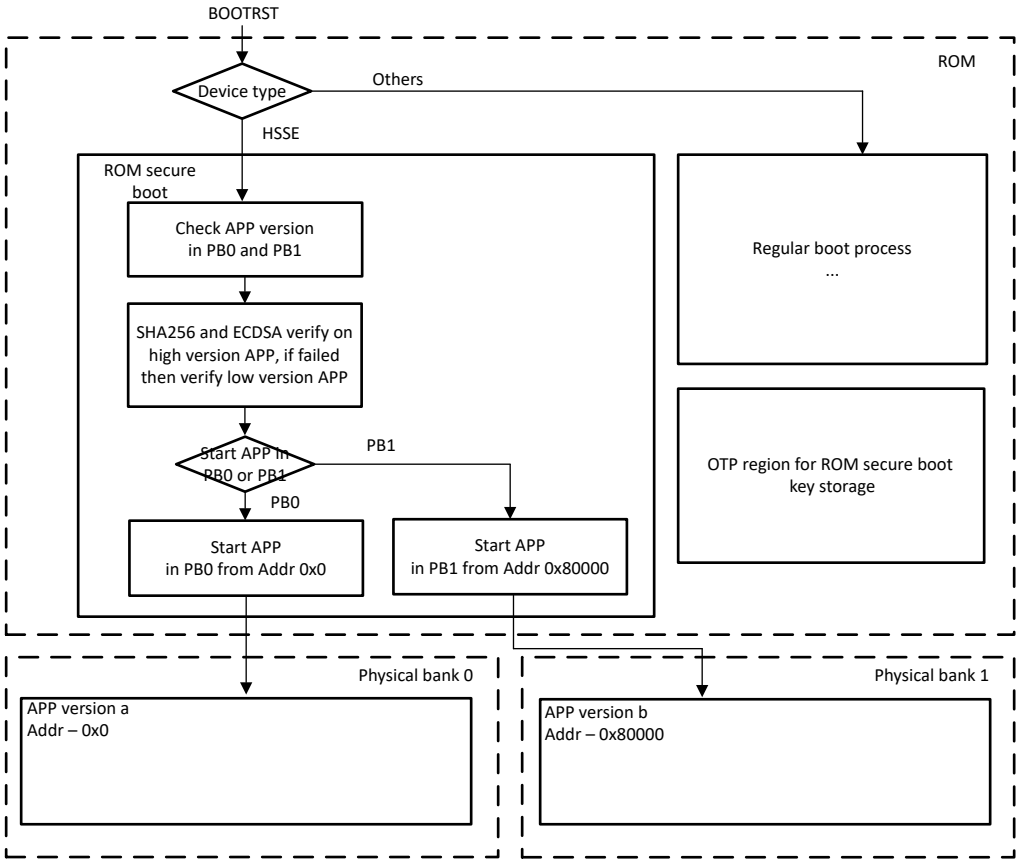

図 3-4. デュアル アプリによる ROM ベースのセキュア ブート

セクション 3.1 に示されているように、ROM ベースのセキュア ブートで検証されたイメージは MCUBoot 形式に従っている必要があります。

3.2.2 顧客キー ストレージ

ファームウェア イメージの認証のための ROM ベースのセキュア ブートは、OTP メモリに格納されている公開鍵に基づいています。

セキュア ブートはこれらの鍵を使用して、対応する秘密鍵によって署名されたファームウェアの署名を検証します。

MSPM33 C シリーズ デバイスは、OTP 領域で最大 8 つの公開鍵をサポートしています。図 3-5 に示されているように、セキュア ブートでは最大 6 つの公開鍵が使用されます。表 2-4 で説明されているように、残りの 2 つの鍵は DSSM コマンドまたは通信での認証に使用されます。

表 3-2. MSPM33 での顧客キーの使用方法

	ROM ベースのセキュア ブート	HSSE デバイスのデバッグ ロック解除	HSSE デバイスのその他の DSSM サービス
Mpuk0-Mpuk5	セキュア ブート認証用の現在のアクティブ キー	-	-
Mpuk6	-	非セキュア デバッグ ロック解除	DSSM サービスの現在のアクティブ キー
Mpuk7		セキュア デバッグ ロック解除	

顧客を OTP 領域にプロビジョニングするには、ユーザーは セクション 2.5.1 に示されている HS-FS から HS-KP への移行手順に従う必要があります。

Copyright © 2026 Texas Instruments Incorporated

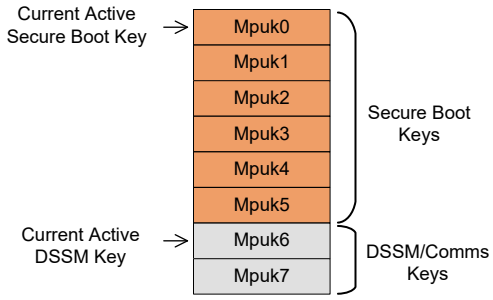

図 3-5. OTP 領域の顧客キー ストレージ

セキュア ブートは、検証のために最初の公開鍵 (Mpuk0) を使用します。残りの公開鍵は、最初の鍵が失効した場合に使用されます。鍵失効には 2 つの方法があります。

- 鍵失効の DSSM サービス
- 現在のアクティブ +1 キーで署名された新しいイメージ

3.2.2.1 DSSM を使用したキー失効

Mpuk0 (現在のアクティブ セキュア ブート キー) が漏洩したために、今後使用を無効化する必要があると仮定します。

デバイスは、現在のアクティブ DSSM キー (MPrK6) を使用して署名された MCUBOOT 証明書とともに送信します。デバイス ペイロードには、無効にする必要がある現在のキーで署名されたイメージが必要です。

最大で 6 つのセキュア ブート キーと 2 つの DSSM キーがあるため、両方のタイプの最後のキーで失効することが重要です。一般的に、ユーザーは最後のセキュア ブート キーまたは最後の DSSM キーを失効させてはなりません。

注

最後のアクティブ キーである MPuK5 を失効させる必要がある場合、アクティブ化するキーはデバイスに残っていません。

注

Mpuk6 と Mpuk7 は DSSM キーに使用されます。Mpuk6 が失効すると、非セキュア デバッグ ロック解除が使用できなくなり、Mpuk7 は現在のアクティブ DSSM キーになります。MPuK7 は失効させないでください。そうしないと、顧客がデバイスのライフサイクルを **BAD** に変更しようとした場合、デバイス内にアクティブ キーが残らなくなってしまいます。

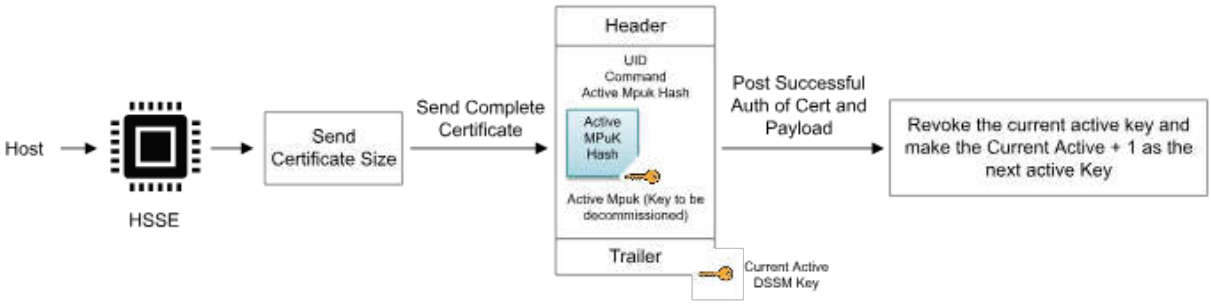

図 3-6. DSSM を使用したキー失効

Copyright © 2026 Texas Instruments Incorporated

3.2.2.2 ファームウェア アップデートを使用したキー失効

DSSM コマンドのほかに、ファームウェアの更新によってもキー失効を実現できます。次のキー (現在のキー + 1) で署名された新しいファームウェアがインストールされた場合、ROM セキュア ブートは最初に現在のアクティブ キーを使用してファームウェアを確認してから次のキーを確認します。次のキーによる認証が成功すると、現在のアクティブ キーが失効します。

注

この方法は、セキュア ブート キーに対してのみ機能します。

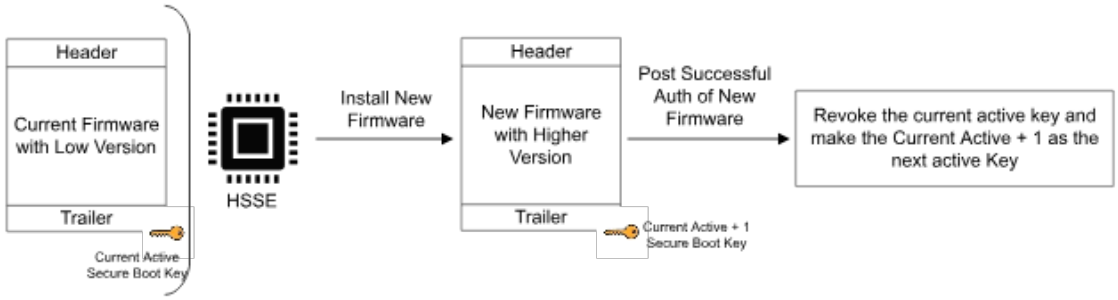

図 3-7. ファームウェア アップデートを使用したキー失効

3.2.3 アンチ ロールバック保護

アンチ ロールバックは、既知の脆弱性を含む古いファームウェアへのダウングレードを防止するセキュリティ メカニズムです。ROM ベースのセキュア ブートにより、MCUboot ヘッダーのバージョン情報について単調バージョン チェックが強制されます。

MCUboot ヘッダー内のイメージ バージョンの構造は次のとおりです。MSPM33C シリーズ ROM ベースのセキュア ブートがチェックおよび比較するのは iv_major のみです。同じメジャー バージョンまたはそれ以上のメジャー バージョンはバージョン チェックを渡すことができますが、下位メジャー バージョンは無視されます。

ROM ベースのセキュア ブートは、マイナー バージョンまたはリビジョンをチェックしないため、ユーザーはマイナー バージョン間でダウングレードできます。ユーザーは、ファームウェア更新のために、イメージ バージョンを適切に維持する必要があります。

```
struct image_version {
    uint8_t iv_major;
    uint8_t iv_minor;
    uint16_t iv_revision;
    uint32_t iv_build_num;
};
```

3.3 CSC ベースのセキュア ブート

MSPM33 C シリーズ デバイスの変更不可能なセキュア ブートに追加の機能を実装する場合は、デバイスのメイン フラッシュにカスタマ セキュア コード (CSC) (セカンダリ ブートローダーとも呼ばれます) を実装できます。CSC ベースのセキュア ブートは、以下の機能をサポートしています。

- ECDSA (楕円曲線デジタル署名アルゴリズム) P-256 と MLDSA (モジュール格子ベース デジタル署名アルゴリズム) –87 を使用したファームウェアの認証
- SHA-256 ハッシュ検証によるファームウェアの整合性
- セキュリティ上の脆弱性が存在する可能性のある古いファームウェアへのダウングレードを防ぐアンチロールバック
- ファームウェアのブート開始のバンク スワップ ポリシーを設定
- AES で使用するための KEYSTORE を設定し、ユーザーの意図に応じて鍵を動的に管理および失効
- セキュリティ メタデータ用の業界標準の MCUBoot トレーラー処理

Copyright © 2026 Texas Instruments Incorporated

注

MAIN フラッシュに配置されている CSC は、アルゴリズムと機能をユーザーが変更できることを意味します。変更不可能なセキュア ブートを実装するには、静的書き込み保護など、必要な保護を適用します。

セクション 2.8.3 に示されているように、CSC を有効にするには、NONMAIN フラッシュで CSC (および必要に応じてバンク スワップ) を有効にする必要があります。ROM は、CSC をサポートするための以下の機能を提供しています。

- バンク スワップの決定
- セキュア キーストアの初期化

MSPM33 SDK は、評価用の CSC の例を提供します。CSC 評価のステップ バイ ステップ ガイドについては、『MSPM33 マイコンにおけるカスタマ セキュア コードの評価』を参照してください。

3.3.1 CSC ベースのセキュア ブート フロー

図 3-8 に示されているように、ブートが成功すると、デバイスは、MAIN フラッシュ メモリからの実行をトリガするために SYSRST を発行し、CSC を実行して実行バンクを判定し、アプリケーションの整合性と信頼性の検証などを行います。

CSC は、ユーザーが変更可能な CSC を使用して、顧客定義のアルゴリズムによってアプリの整合性と認証を検証します。SDK の CSC 例では、SHA256+ECDSA_p256 と MLDSA87 の両方を実装しています。バージョンが異なる 2 つのアプリがある場合、CSC はまず上位バージョン アプリを検証します。失敗した場合、CSC は下位バージョンのアプリを検証します。両方のアプリで認証が失敗するか、バンク 0 とバンク 1 で定義されたアドレスにデータがないと、CSC が失敗します。CSC は変更可能であるため、ユーザーは必要に応じてフローまたは機能を変更できます。

CSC ベースのセキュア ブートに成功すると、GSC VTOR は信頼済みアプリケーションに構成されます。CSC は、キー値 0x9d とともにパス値 (0x1) を指定して SYSCTL.SECCFG.INITDONE レジスタに書き込むことで、CSC 実行の終了を示します。INITDONE レジスタに正常に書き込むと、二回目の SYSRST 動作が発生し、バンク スワップが有効になります。構成済みの VTOR では、信頼済みアプリケーションが起動します。セクション 2.7 を参照してください。

図 3-8 に、HSFS デバイスでの CSC フローを示します。通常のブート プロセスの後、物理アドレス 0x0 が唯一のエントリポイントになります。一般的に、ユーザーは書き込み消去保護 (またはその他の保護) を適用して、CSC を変更不可能にすることができます。

図 3-11 に、HSSE デバイスでの CSC フローを示します。ROM ベースのセキュア ブートと CSC の組み合わせが、信頼の連鎖を形成します。詳細については、セクション 3.4 を参照してください。

Copyright © 2026 Texas Instruments Incorporated

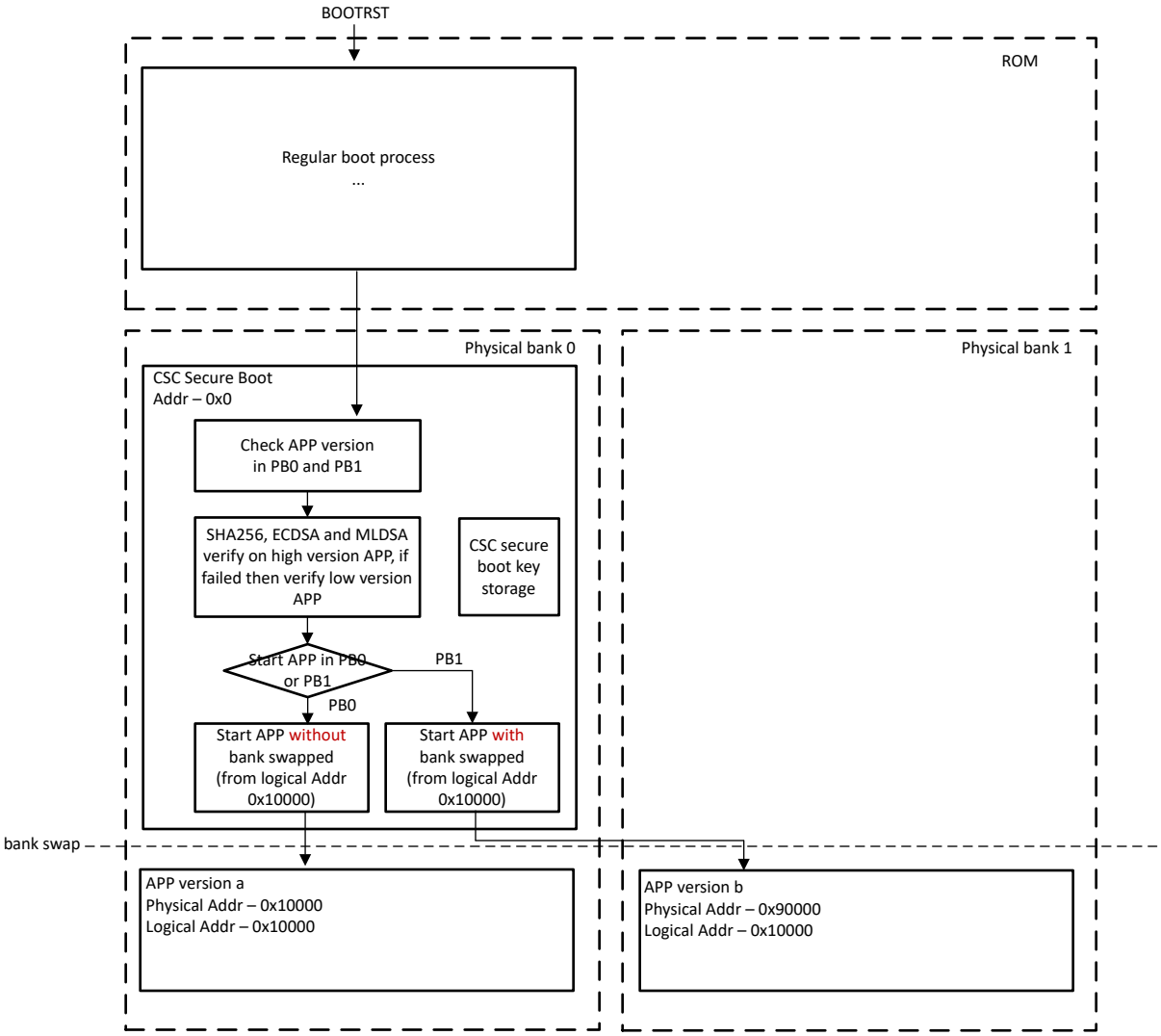

図 3-8. HSFS デバイスにおける CSC ベースのセキュア ブート

3.3.2 FLASH メモリ マップ

図 3-9 に、CSC セキュア ブートの詳細なフラッシュ メモリ マップを示します。CSC のセクションの説明を次に示します。

- SECRET：SECRET は CSC からは見えますが、読み取り保護ファイアウォールによって保護されているため、アプリケーションからは SECRET は見えません。SECRET 領域には、実行時に KEYSTORE にロードされる不揮発性キーを保存できます。したがって、アプリケーションは読み取りアクセス権なしでこれらのキーを使用できます。SECRET をカスタマイズして、タグやキーなどの追加情報を含めることができます。
- ロック可能なフラッシュ メモリ：ロック可能なフラッシュ メモリは、CSC によって書き込まれる必要があり、アプリケーションによって読み取られるものの変更はされない重要な情報に対して、動的な書き込み保護を提供します。セキュリティ カウンタ (ロールバック保護)、キーストア ハッシュ テーブル、およびイメージ ハッシュは通常、ロック可能なフラッシュに格納されます。ロック可能なコンテンツは、両方のバンク アプリケーション プログラムがこの領域にアクセスできるように、PB0 で CSC 領域にのみプログラムされていることに注意してください。
- CSC 割り込みベクタ：これらはカスタマ セキュア コード用の割り込みベクタです。この割り込みベクタ テーブルは、BOOTRST またはデフォルトの VTOR を指定した SYSRST が発生した場合に、フラッシュ メモリから最初に実行されるものです。CSC では、VTOR がアプリケーション割り込みベクタのアドレスで更新され、その後、構成された VROT を持つ SYSRST によってデバイスがアプリケーションから起動されます。

Copyright © 2026 Texas Instruments Incorporated

注

SYSRST は VTOR をリセットしません。つまり、CSC が実行された後、次の BOOTRST まで、デバイスは CSC に再度入ることはありません。

- CSC コード:メイン コードとセキュリティ プリミティブが、カスタマ セキュア コードの大部分を占めます。通常、CSC コードと割り込みベクタは物理バンク 0 にのみプログラムすれば十分です。これは、CSC はバンク スワップが有効になる前に実行されるためです。バンク スワップ中、INITDONE によってトリガされた SYSRST の後、プログラムは常に論理バンク 0 (論理アドレス) のアプリケーションから実行されます。FLASHCTL は、バンク スワップ ポリシー構成に従って、アドレスを PB0 アドレスまたは PB1 アドレスにマッピングします。

アプリケーション イメージのセクションは、セクション 3.1 に示されている MCUboot 形式に従います。

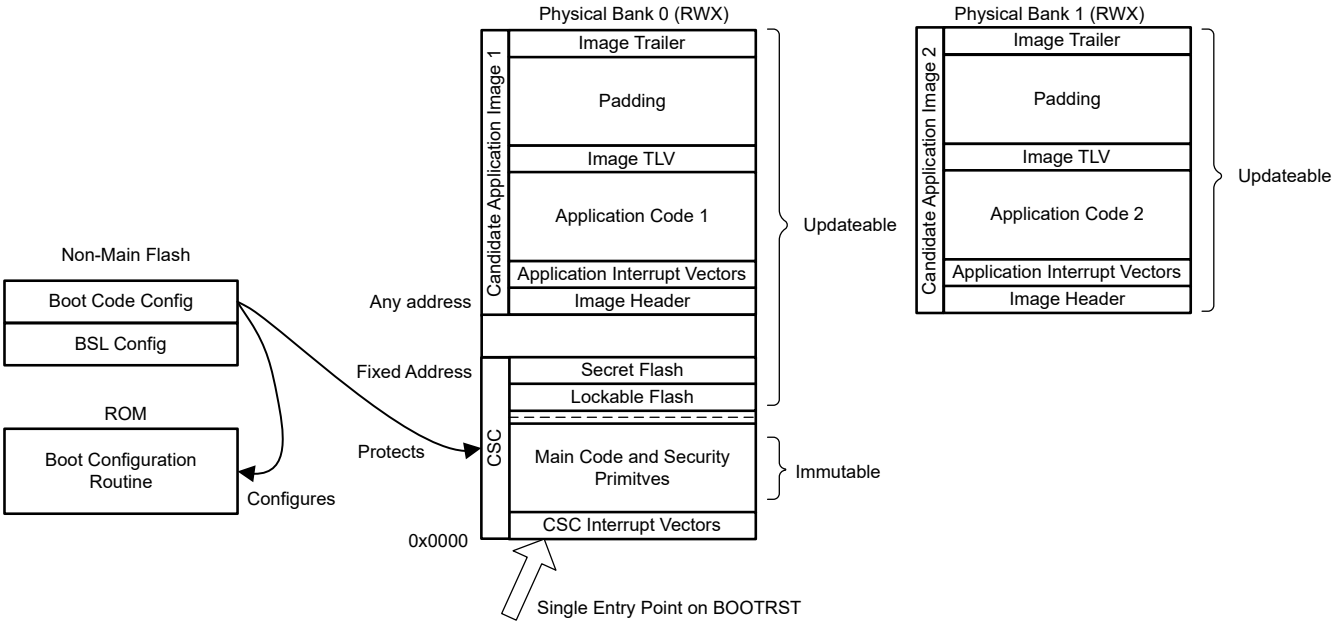

図 3-9. CSC フラッシュマップ

3.3.3 CSC キー ストレージ

3.3.3.1 CSC ベースのセキュア ブート用のキー ストレージ

ROM ベースのセキュア ブート以外に、CSC ベースのセキュア ブートで使用される公開鍵は CSC コードで格納されます。セキュア ブートはこれらの鍵を使用して、対応する秘密鍵によって署名されたファームウェアの署名を検証します。

MSPM33 CSC の例では、ECDSA_p256 公開鍵と MLDSA87 公開鍵は DER 形式で boot_keys.c に格納されています。

注

ユーザーは CSC コードを変更し、必要なアルゴリズムや他の機能を有効にすることができます。

```
const unsigned char ecdsa_pub_key[] = {
...
};
const unsigned int ecdsa_pub_key_len = 91;
const unsigned char mldsa_pub_key[] = {
...
};
const unsigned int mldsa_pub_key_len = 2592;
const struct bootutil_key bootutil_keys[] = {
    {
        .key = ecdsa_pub_key,
        .len = &ecdsa_pub_key_len,
    },
    {
        .key = mldsa_pub_key,
```


Copyright © 2026 Texas Instruments Incorporated

```
        .len = &mldsa_pub_key_len,
    },
};
const int bootutil_key_cnt = 2;
```

3.3.3.2 AES の KEYSTORE

デフォルトでは、CSC の例は AES のような対称アルゴリズムを実装していません。ただし、CSC は AES キーを「SECRET」セクションに格納し、KEYSTORE を構成するために使用されます。これにより、ユーザー アプリケーションは KEYSTORE を使用して AES アルゴリズムをアプリケーション内に安全に実装できます。詳細については、セクション 5.3 を参照してください。

3.3.4 バンク スワップ

ユーザーは、NONMAIN フラッシュの BOOTCFG2.FLASHBANKSWAPPOLICY フィールドを構成することで、CSC ベースのセキュア ブートでバンク スワップを有効化できます。バンク スワップの動作は CSC 段階で発生し、デバイスが INITDONE を発行した後の動作は有効になります。

注

NONMAIN フラッシュの BOOTCFG2.FLASHBANKSWAPPOLICY フィールドはデフォルトで無効になっています。このフィールドを有効にすると、バンク スワップは CSC ベースのセキュア ブートに対してのみ有効になり、ROM ベースのセキュア ブートのバンク スワップはありません。ただし、このフィールドによっては、ROM ベースのセキュア ブートは、スタンドアロン アプリ (バンク スワップが無効の場合) またはデュアル アプリ (バンク スワップが有効の場合) をチェックおよび検証します。このフィールドを変更する方法については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

バンク スワップ方式では、実際のファームウェアは物理的な場所を移動せず、代わりに 2 つのバンクのアドレスが交換されます。図 3-8 に示されているように、物理バンク 0 (PB0) と物理バンク 1 (PB1) があります。これらのバンクは、バンク スワップ構成に応じてアドレスを切り替えることができます。

アドレス スワッピングを示すために、物理バンク 0 または物理バンク 1 にマッピングされる 2 つの論理バンク (LB) を定義できます。INITDONE より前の論理バンク 0 は、常にアドレス PB0 に、論理バンク 1 は PB1 にマッピングされます。ただし、バンク スワップの実行によっては、INITDONE 後の LB0 を PB0 または PB1 にマッピングできます。

一般に、バンク スワップ機能が有効な場合、CSC は常に LB0 でアプリを起動します。そのため、実行中のアプリが PB0 にある場合、CSC は FLBANKSWP.USEUPPER を 0h (デフォルト) に設定してバンク スワップが発生しないように設定し、LB0 を PB0 にマッピングします。実行中のアプリが PB1 にある場合、CSC は FLBANKSWP.USEUPPER を 1h (デフォルト) に設定し、LB0 を PB1 にマッピングします。

注

コードは論理バンク 0 からのみ実行されるため、コードが論理バンク 0 から実行されるようにすべてのコードをコンパイルします。したがって、CSC が 0x0000 ～ 0x47FF にマッピングされ、1 つのアプリケーション イメージが 0x4800 以降にマッピングされている場合、アプリケーション イメージが実効物理アドレスで論理バンク 1 に書き込まれる場合でも、各アプリケーション イメージを 0x4900 (0x100 トレーラー、次に 0x4800 + 0x100 = 0x4900 に配置されたアプリケーション割り込みベクタ) を基準にコンパイルします。

3.3.5 CSC 保護

顧客が所有するセキュア コードは、GSC を活用して、以下のような追加のセキュリティ機能を実装できます。

- フラッシュ書き込み消去保護 (WEP) - セクション 5.1
- フラッシュ非表示保護 (HDP) - セクション 5.2
- セキュアまたは非セキュア、および特権または非特権属性 - セクション 4

HSFS デバイスの場合、ユーザーは CSC 領域に WEP を適用して CSC を変更不可能にすることができます。HSSE デバイスの場合、ROM ベースのセキュア ブートは CSC と連携して信頼チェーンとして動作することができ、CSC の更新が可能になります。

Copyright © 2026 Texas Instruments Incorporated

一般的に、アプリ / デバッグからの読み取り / アクセスを避けるために、SECRET 領域で HDP が必要となります。

また、NONMAIN でデバッグ ホールドを設定して、CSC 実行中に SWD ポートをロックすることもできます。

3.4 2 段階セキュア ブート

図 3-10 に示されているように、ROM ベースのセキュア ブートと CSC ベースのセキュア ブートをベースにすることで、ユーザーは変更不可能な信頼ルート (iRoT) から開始する信頼の連鎖を構築できます。第一段階のセキュア ブートは変更できませんが、第二段階のセキュア ブートはユーザーが変更できます。

- 第一段階 (BL1) — ROM ベースのセキュア ブート。デバイスで認証および承認された CSC を実行できます。
- 第一段階の (BL2) — CSC ベースのセキュア ブート。デバイスで認証および承認されたアプリを実行できます。

2 段階セキュア ブートは、これまでの 2 つのセキュア ブート方式を組み合わせたもので、デバイスの寿命期間中にユーザーが CSC を更新できるようにするものです。第一段階の特徴については、セクション 3.2 を参照してください。第二段階の特徴については、セクション 3.3 を参照してください。2 段階セキュア ブートを有効にするには、デバイスを HS-SE 状態に遷移し、CSC を有効にする必要があります。ブート プロセスについては、セクション 2.7 を参照してください。

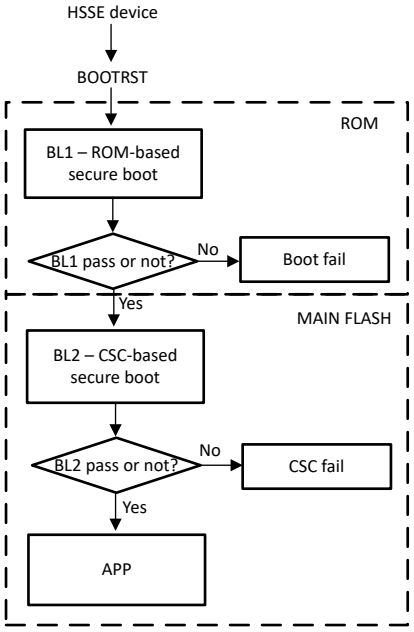

図 3-10. 信頼の連鎖

3.4.1 2 段階セキュア ブート フロー

図 3-11 に示されているように、MSPM33 C シリーズ デバイスの電源がオンになると、最初に ROM コードが実行され、BL1 (ROM ベースのセキュア ブート) が強制されて BL2 の認証が行われます。

BL1 は、SHA-256 経由で B2 の整合性と、ECDSA_p256 を使用してファームウェアの認証を検証します。バージョンが異なる 2 つの BL2 がある場合、BL1 は最初に上位バージョン BL2 を検証します。失敗した場合、BL1 は下位バージョン BL2 を検証します。BL2 の両方で認証が失敗するか、バンク 0 およびバンク 1 の開始アドレスのデータがない場合、ブートが失敗します。BL1 は変更不可能であるため、ユーザーは BL1 のフローまたは機能は変更できません。BL1 は、セクション 3.2 に示されているすべての機能をサポートしています。

注

デュアル BL2 に同じバージョンがある場合、ROM ベースのセキュア ブートは、バンク 0 で BL2 を最初に検証します。

Copyright © 2026 Texas Instruments Incorporated

注

セクション 3.1 に示されているように、BL2 は MCUboot 形式になっている必要があります。

BL1 が正常に完了すると、ブートコードは BOOTDONE を発行します。この時点で、デバイスに対して SYSRST が発行され、MAIN フラッシュ内の認証済み BL2 からの実行がトリガされます。図 3-11 に示されているように、BL2 にはアプリの認証が強制されます。

BL2 は、ユーザーが変更可能な BL2 を使用して、顧客定義のアルゴリズムによってアプリの整合性と認証を検証します。SDK の CSC 例では、SHA256+ECDSA_p256 と MLDSA87 の両方を実装しています。バージョンが異なる 2 つのアプリがある場合、BL2 はまず上位バージョン アプリを検証します。失敗した場合、BL2 は下位バージョンのアプリを検証します。両方のアプリで認証が失敗するか、バンク 0 とバンク 1 で定義されたアドレスにデータがないと、CSC が失敗します。BL2 は変更可能であるため、ユーザーは必要に応じてフローまたは機能を変更できます。BL2 は、セクション 3.3 に示されているすべての機能をサポートしています。

BL2 が正常に完了すると、GSC VTOR は信頼済みアプリに構成されます。また、BL2 は INITDONE を発行し、その結果、2 回目の SYSRST 動作が実行され、バンク スワップが有効になります。構成済みの VTOR では、信頼済みアプリが起動します。

注

バンク スワップは INITDONE によって発行された SYSTST の後に有効になるため、2 つのバンクのアプリは同じ論理アドレスを使用できますが、2 つのバンクの BL2 は異なる物理アドレスに基づきます。

Copyright © 2026 Texas Instruments Incorporated

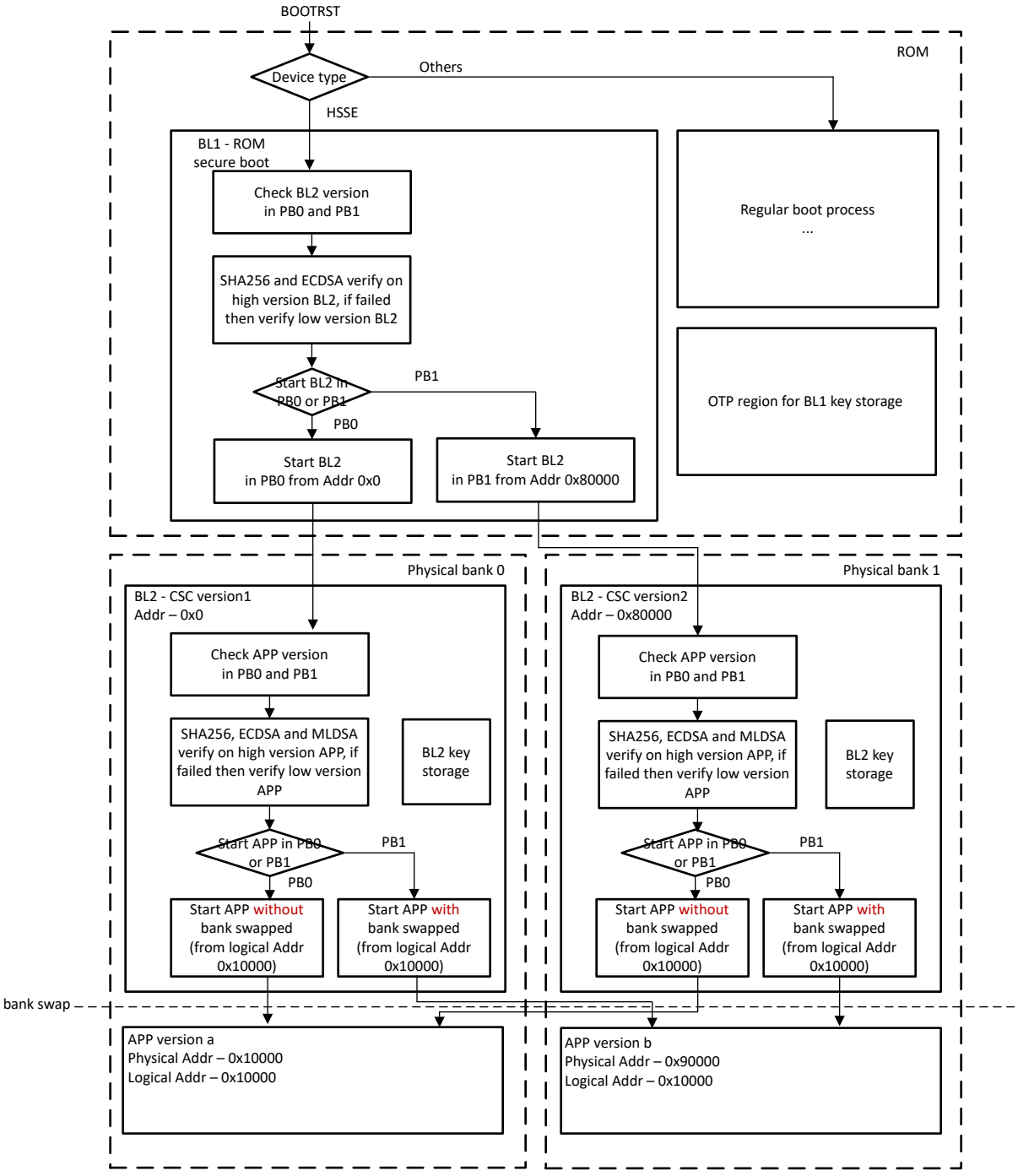

図 3-11. 2 段階セキュア ブート

Copyright © 2026 Texas Instruments Incorporated

4 セキュアな実行環境

TrustZone ®アーキテクチャは、Arm® Cortex®-M33 によって実現されます。TrustZone® を使用すると、プログラマはメモリのセクションをセキュアと非セキュアに分割でき、権限のないユーザーによるメモリのセクションへのアクセスを制限できます。さらに、MSPM33C デバイスにはグローバル セキュリティ コントローラ (GSC) が搭載されており、CPU サブシステム (CPUSS) の外部に追加のセキュリティを提供できます。これにより、DMA などの外部ペリフェラルが、非セキュア プロセッサ状態からメモリのセキュア領域にアクセスすることを防止できます。

4.1 TrustZone®

Cortex®-M33 TrustZone® セキュリティを使用すると、プログラマはメモリをセキュア、非セキュア、および非セキュア呼び出し可能の 3 種類の領域に分割できます。セキュア状態のプロセッサは、非セキュア リソースとセキュア リソースの両方にアクセスできますが、非セキュア状態のプロセッサは、非セキュア リソースにしかアクセスできません。非セキュア呼び出し可能なメモリ領域は、非セキュア領域とセキュア領域間のセキュア ゲートウェイ (橋渡し) として機能し、ユーザーが SG 命令を呼び出したときにプログラマがセキュア メモリ領域にジャンプできるようにします。詳細については、『MSPM33 マイコンにおけるセキュア アプリケーション用 Trustzone のイネーブル』または『Armv8-M アーキテクチャの TrustZone® テクノロジー』ドキュメントを参照してください。

セキュアおよび非セキュアなメモリ領域を定義するために、セキュリティ属性ユニット (SAU) および実装定義属性ユニット (IDAU) が使用されます。IDAU は RTL レベルに設定され、TI によって構成されています。IDAU の構成方法については、『MSPM33C3 シリーズ 160MHz マイコン テクニカル リファレンス マニュアル』の「CPU」セクションを参照してください。

SAU は、CPU サブシステム (CPUSS) によって使用され、メモリ領域をセキュアまたは非セキュアとして定義します。このペリフェラルにより、プログラマはコードをセキュアまたは非セキュアとして分割できます。SAU および IDAU がどのように連携して領域をセキュアまたは非セキュアとして定義するかについては、Armv8-M アーキテクチャの TrustZone® テクノロジーのドキュメントを参照してください。

『MSPM33 マイコンにおけるセキュア アプリケーション用 Trustzone のイネーブル』には、コード開発時に TrustZone® を構成する方法、および MSPM33 SDK の TrustZone® の例を評価する方法が記載されています。

4.1.1 実装定義属性ユニット

実装定義属性ユニット (IDAU) は TI によって定義されており、『MSPM33C3 シリーズ 160MHz マイコン テクニカル リファレンス マニュアル』の「CPU」セクションに説明があります。IDAU の構成は SAU と一緒に使用され、領域の最終的なセキュリティ レベルが構成されるため、IDAU の構成方法を知ることが重要です。たとえば、MSPM33C32 デバイスでは、アドレスの 8 番目のビットが 1 のとき、メモリ領域は非セキュア呼び出し可能として定義され、0 のときは非セキュアとしてのみ定義されるように IDAU が構成されています。たとえば、アドレス 0x1000.0000 は NSC として定義されます。

IDAU と SAU の両方を使用して、最終的なセキュリティ レベルが決定されます。この 2 つのうち、より高いセキュリティ レベルが、最終的なセキュリティ属性レベルとして設定されます。MSPM33C321A における例については、図 4-1 を参照してください。

Memory Location	IDAU	SAU	Final Security Level
0x0000.0000 to 0x00FF.FFFF	Non-Secure	Non-Secure	Non-Secure
0x1000.0000 to 0x10FF.FFFF	Non-Secure Callable	Non-Secure Callable	Non-Secure Callable
0x2000.0000 to 0x203F.FFFF	Non-Secure	Secure	Secure
0x3000.0000 to 0x303F.FFFF	Non-Secure Callable	Non-Secure	Non-Secure Callable
0x4000.0000 to 0x4FFF.FFFF	Non-Secure	Non-Secure callable	Non-Secure callable
0x5000.0000 to 0x5FFF.FFFF	Non-Secure Callable	Secure	Secure

図 4-1. MSPM33C321A の IDAU および SAU 構成例

Copyright © 2026 Texas Instruments Incorporated

4.1.2 セキュリティ属性単位

セキュリティ属性単位 (SAU) は、プログラマがメモリ領域のセキュリティを定義するために使用できます。最終的なセキュリティ レベルでは、SAU 構成と IDAU 構成の両方が使用されます。メモリのセクションのセキュリティ レベルを決定するには、SAU と IDAU のうち、より高いセキュリティ レベルを採用します。たとえば、SAU がある領域をセキュアとして定義し、IDAU がその領域を非セキュアとして定義している場合、その領域はセキュアとして定義されます。

MSPM33 デバイスでは、SAU は、8 つの領域をセキュア、非セキュア、または非セキュア呼び出し可能として定義するように構築されています。

4.1.2.1 領域をセキュアとして定義

SAU の主要な用途の 1 つは、領域をセキュアまたは非セキュアとして定義することです。領域を定義するには、次の手順で SAU のレジスタに書き込みます。

1. SAU->RNR レジスタを使用して、構成する領域を選択します。
2. SAU>RBAR レジスタを使用して、領域のベース アドレスを選択します。
3. SAU->RLAR を設定して、領域のサイズを構成します。
4. すべての領域について、手順 1 ～ 3 を繰り返します。
5. SAU->CTL レジスタを構成して、構成したすべての領域に伝播します。

4.1.3 TrustZone® ソフトウェア開発

プログラマのコード開発に SAU の使用を統合するには、CMSE ライブラリを使用することをお勧めします。このライブラリは Arm® によって設計されており、ユーザーは関数をセキュアまたは非セキュアとして簡単に定義できます。このガイドラインの詳細については、『MSPM33 マイコンにおけるセキュア アプリケーション用 Trustzone のイネーブル』または『ARMv8-M セキュア ソフトウェア ガイドライン』を参照してください。

ARMv8-M instruction は、BXNS (Branch and Exchange Non-Secure) および SG (Secure Gateway) を使用して、非セキュア コードとセキュア コード間の遷移を行います。BXNS はセキュア コードから非セキュア コードへの移行に使用され、SG は非セキュアからセキュアへの遷移に使用されます。SG コマンドは、NSC 領域を使用してこの遷移を行います。これらの命令をプログラミング環境に適用するために、CMSE ライブラリではこれらの機能を関数呼び出しに統合しています。

4.1.3.1 関数をセキュアおよび非セキュアとして定義

CMSE ライブラリの主な機能の 1 つは、プログラマが関数をセキュアまたは非セキュアとして定義できることです。これにより、CPU 状態が決定され、プログラマはセキュア領域と非セキュア領域の間を遷移できます。CMSE ライブラリをプログラムに追加するには、次のヘッダーを使用します。

```
#include <arm_cmse.h>
```

コードに SG コマンドを実装するには、関数に cmse_nonsecure_entry 属性を指定します。この関数定義は、コードの先頭でセキュア ゲートウェイ ベニヤを呼び出します。これは、非セキュア コードがアクセス可能な、セキュア メモリの情報を使用する関数を定義する際に便利です。

```
__attribute__((cmse_nonsecure_entry)) secure_fxn()
{

}
```

セキュア コードから非セキュア関数を呼び出すには、BXNS 関数を使用できます。CMSE ライブラリを使用してこれをプログラミング環境に簡単に統合するには、関数に cmse_nonsecure_call 属性を指定してください。例として以下のコード ブロックを参照してください。

```
__attribute__((cmse_nonsecure_call)) nonsecure_fxn()
{

}
```


Copyright © 2026 Texas Instruments Incorporated

4.2 メモリ保護ユニット

MSPM33C デバイスには、CPUSS にメモリ保護ユニット (MPU) が搭載されています。MPU では、プログラマはメモリ領域を特権付きまたは特権なしとして定義できます。これは、SAU が領域をセキュアおよび非セキュアとして定義する方法と同様に動作します。主な違いは、デバイスがスレッド モードかハンドラ モードかに基づいて特権モードが変わることです。

リセット状態から復帰すると、デバイスはスレッド モードになり、プロセッサまたはペリフェラルによって例外が発行されるとハンドラ モードに移行します。例外は常に特権モードで動作し、スレッド モードでは MPU を使用して特権レベルを決定できます。詳細については、『MSPM33C3 シリーズ 160MHz マイコン テクニカル リファレンス マニュアル』を参照してください。

4.2.1 TrustZone® および MPU

MSPM33C CPUSS では、MPU、SAU、IDAU を使用して、さまざまな権限とセキュリティ レベルを組み合わせた属性をメモリ領域に割り当てます。これにより、プログラマは特権とセキュリティ制限を使用して、アクセス権限を多層的に設定できます。図 4-2 に、プログラマが使用できる組み合わせの例を示します。デバイスがハンドラ モードの場合、プロセッサは常に特権状態になります。

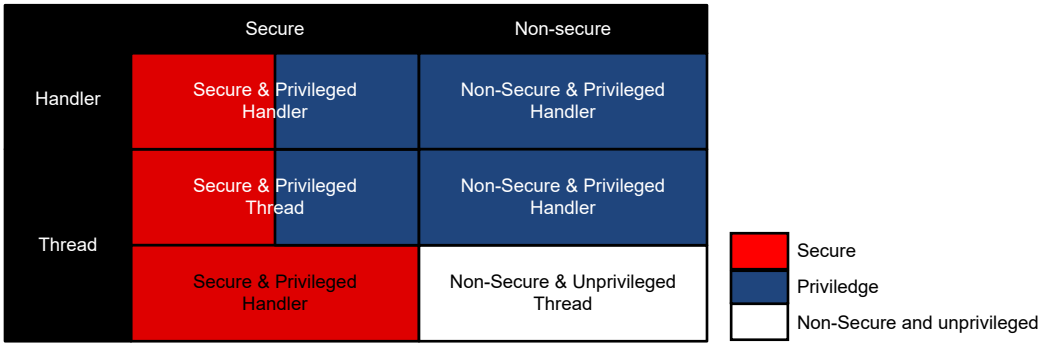

図 4-2. セキュリティと特権メモリ属性

4.3 グローバル セキュリティ コントローラ

グローバル セキュリティ コントローラ (GSC) は、MSP に対する独自のペリフェラルであり、TrustZone® アーキテクチャに、さらにセキュリティ層を追加します。GSC は、ペリフェラル、フラッシュ、および SRAM のセキュア属性と特権属性の両方を構成します。これにより、プログラマは 1 つの IP を使用してすべてのリソースを保護でき、セキュアでないペリフェラルがメモリのセキュア領域にアクセスするのを防ぎます。1 つの例として、非セキュアな DMA 読み取りセキュア メモリが挙げられます。GSC を使用することで、セキュアでない DMA がメモリのセキュア領域にアクセスしようとしたときにフォルトを発生させ、これらの攻撃からシステムを保護できます。GSC の詳細については、『MSPM33C3 シリーズ 160MHz マイコン テクニカル リファレンス マニュアル』の「グローバル セキュリティ コントローラ」の章を参照してください。

4.3.1 GSC メモリ構成

GSC は MPU と SAU の両方を利用し、セキュリティ属性および特権属性を用いてメモリを構成します。この構成を行うため、GSC はペリフェラル保護コントローラ (PPC)、SRAM 保護コントローラ (SPC)、フラッシュ保護コントローラ (FPC) の 3 つの個別のコントローラを備えています。レジスタの構成および使用方法の詳細については、『MSPM33C3 シリーズ 160MHz マイコン テクニカル リファレンス マニュアル』を参照してください

GSC は、SAU および MPU と同様の特性を持っています。これにより、メモリの保護にどのペリフェラルを使用すべきか混乱が生じる可能性があります。SAU および MPU は、構成可能な領域の数またはアドレスの粒度に制限されます。たとえば、M33 デバイスの SAU では、非セキュア属性とセキュア属性を 8 つのメモリ領域にのみ設定できます。未構成のメモリ領域は、セキュア属性のままとなります。ただし、GSC は 2kB サイズの領域に属性を割り当てます。1MB デバイス上で 255 の異なるセキュリティ領域を設定可能です。これらの 2kB 領域のそれぞれに、セキュア アクセスまたは特権アクセスのいずれかの属性が割り当てられます。

Copyright © 2026 Texas Instruments Incorporated

4.3.1.1 GSC および SAU によるセキュリティ例外

GSC と SAU の両方に、メモリ領域にセキュリティ属性を割り当てる機能があります。両者の主な違いの 1 つは、GSC がセキュア NMI を生成するのに対し、SAU はセキュア フォルトを生成することです。セキュア NMI はエラー集約モジュール (EAM) によって処理されるため、プログラマはエラーに関する特定の詳細を参照できます。エラーを引き起こしたペリフェラルや、エラーが発生したアドレスなどです。また、GSC は 8 つの領域に限定されません。フラッシュ、SRAM、およびペリフェラルは、セキュリティ属性に対して独自の粒度を持っています。詳細については、『MSPM33 マイコンにおけるセキュア アプリケーション用 Trustzone のイネーブル』または『MSPM33C3 シリーズ 160MHz マイコン』テクニカル リファレンス マニュアルを参照してください。

GSC および SAU を使用して、メモリ領域にセキュアまたは非セキュアの属性を割り当てるときに発生するエラーについては、表 4-1 を参照してください。また、図 4-3 に、これについてのイメージを示します。

表 4-1. GSC および SAU のセキュリティ属性割り当てに基づくセキュリティ割り込み

プロセッサの状態	SAU + IDAU セキュリティ属性割り当て	GSC セキュリティ属性割り当て	結果の割り込み	アクセス違反
セキュアな機能	セキュアな機能	セキュアな機能	なし	なし
非セキュア	セキュアな機能	セキュアな機能	セキュリティ故障	あり (CPU によってブロック)
セキュアな機能	セキュアな機能	非セキュア	なし	なし
非セキュア	セキュアな機能	非セキュア	セキュリティ故障	あり (CPU によってブロック)
セキュアな機能	非セキュア	セキュアな機能	セキュリティ NMI	あり (GSC によってブロック)
非セキュア	非セキュア	セキュアな機能	セキュリティ NMI	あり (GSC によってブロック)
セキュアな機能	非セキュア	非セキュア	なし	なし
非セキュア	非セキュア	非セキュア	なし	なし

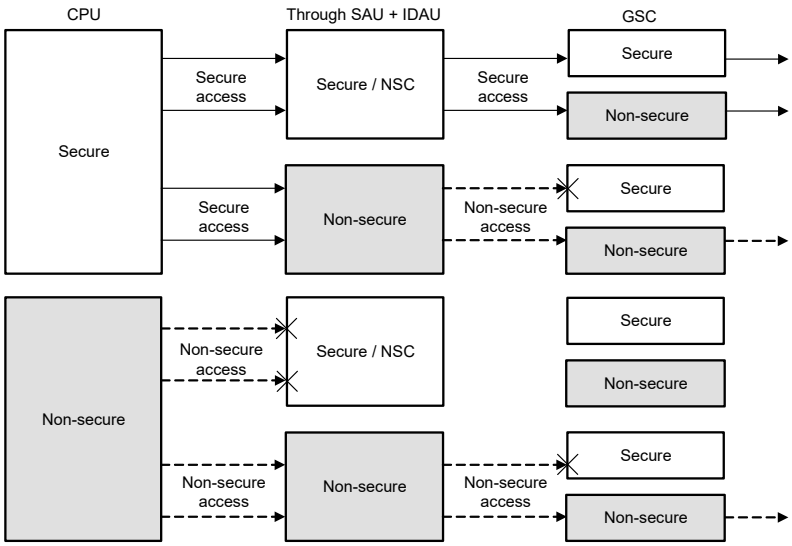

図 4-3. SAU+GSC および GSC によるトランザクション フィルタリング (デフォルトの ATTRIBVIOLS_CONFIG を使用)

4.3.1.2 GSC と MPU を使用した特権例外

GSC と MPU を組み合わせて使用すると、メモリ領域を特権または非特権として定義できます。これは、GSC および SAU がメモリ領域をセキュアまたは非セキュアとして定義する方法と同様です。これにより、プログラマは特権アクセスを使用してメモリ領域に別の保護層を追加することができます。特権状態は例外の発生によっても変化するため、割り込みが発生したときに期待される特権アクセスが正しく設定されていることを確認してください。

GSC および MPU を使用してメモリの領域に特権または非特権の属性を割り当てるときに、どちらのフォルトが優先的に発生するかを判断するには、表 4-2 を使用します。

Copyright © 2026 Texas Instruments Incorporated

表 4-2. GSC および MPU 特権属性割り当てに基づく特権例外

プロセッサの状態	MPU 特権属性割り当て	GSC 特権属性割り当て	結果の割り込み	アクセス違反
特権	特権	特権	なし	なし
非特権	特権	特権	MemMange フォルト	あり (CPU によってブロック)
特権	特権	非特権	なし	なし
非特権	特権	非特権	MemMange フォルト	あり (CPU によってブロック)
特権	非特権	特権	なし	なし
非特権	非特権	特権	特権 NMI	あり (GSC によってブロック)
特権	非特権	非特権	なし	なし
非特権	非特権	非特権	なし	なし

Copyright © 2026 Texas Instruments Incorporated

5 セキュア ストレージ

MSPM33 は、フラッシュ、AES 秘密鍵、SRAM にさまざまな種類のメモリ保護メカニズムを搭載しており、多様なセキュリティ要件に対応できます。MSPM33 デバイス シリーズのこれらの機能については、プラットフォームのセキュリティ イネーブラ を確認してください。

5.1 書き込み消去保護

MSPM33 マイコンには、静的書き込み保護 (WEP) 方式が実装されており、フラッシュ領域のユーザー定義セクタを実行時のプログラム / 消去操作からロックアウトできます。目的の静的書き込み保護方式は、グローバル セキュリティ コントローラ (GSC) で構成され、ユーザーは MAIN フラッシュ、DATA フラッシュ、または NONMAIN フラッシュに書き込み保護を適用できます。

静的書き込み保護により、プログラムおよびロック後にフラッシュ メモリに固定のユーザー定義領域を配置できるため、その領域はアプリケーション コードまたは ROM ブートローダーで変更できません。

- MAIN フラッシュ メモリの先頭に配置された場合、その領域内のアプリケーションは、ROM ブート構成ルーチンが実行をユーザー アプリケーションに転送する際に最初に実行されるコードとなり、これはセキュア ブート イメージ マネージャを実装するための要件の 1 つです。
- NONMAIN フラッシュに配置された場合、NONMAIN フラッシュ内の構成メモリはロックされ、ブートストラップ ローダまたはアプリケーション コードによるセキュリティ ポリシー、ブートストラップ ローダ ポリシー、および静的書き込み保護ポリシーの不正な変更を防止できます。

GSC で書き込みロックするように構成されている領域は、ブート構成ルーチンが MAIN フラッシュ内のブートストラップ ローダまたはユーザー アプリケーション コードに実行を移すと、機能を変更することはできません。静的に保護された領域をアプリケーション コードまたはブートストラップ ローダでプログラムまたは消去しようとすると、ハードウェア フラッシュ動作エラーが発生し、セクタは変更されません。

注

書き込み保護用の GSC レジスタのアクセス タイプは「R/W1S」です。「W1S」とは、レジスタ ビットを 0 ～ 1 の範囲でしか設定できず、レジスタは BOOTRST 以上のリセット タイプでのみリセット可能であることを意味します。つまり、適用された書き込み保護を直接解除することはできません。

MSPM33 は、NONMAIN フラッシュでの書き込み保護ポリシーの構成もサポートしています。ROM のブートアップ中、ブートによって NONMAIN フラッシュに従って GSC が構成されます。この構成は、アプリケーション コードを入力する前に有効になります。これは、NONMAIN フラッシュを再度変更しない限り、アプリケーション コードから書き込み保護を削除することはできないことを意味します。構成については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

静的書き込み保護により、アプリケーション コードやブートローダーからの変更は防止されますが、SWD インターフェイスを介して送信される一括消去や工場出荷時リセット コマンドは受け付けられます。この動作が望ましくない場合は、一括消去または工場出荷時リセットを行う SWD コマンドを固有のパスワードを使用して保護するか、無効化できます。静的書き込み保護された MAIN フラッシュ セクタを変更する手段を完全に削除するには、以下の手順に従ってください。

- NOMAIN の MAIN フラッシュへの書き込み保護を設定します。
- NONMAIN の一括消去および工場出荷時リセット コマンド (または SW-DP) を無効化します。
- アプリケーション コードが NONMAIN 領域の内容を変更することで、基盤となる書き込み保護方式を変更することを防ぐため、NONMAIN フラッシュ メモリに書き込み保護を設定します。

上記の手順については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

注

デフォルトでは、NONMAIN 構成メモリ (ユーザー指定のブート セキュリティ ポリシーとブートストラップ ローダ ポリシーを含む) は書き込み保護の状態になっていません。これにより、プロビジョニング中にユーザーが NONMAIN を消去し、量産時に使用されるユーザー指定のポリシーで再プログラミングすることができます。

Copyright © 2026 Texas Instruments Incorporated

5.1.1 WEP でのグローバル セキュリティ コントローラ構成

グローバル セキュリティ コントローラ (GSC) は、MSPM33 の書き込み消去保護を構成できるペリフェラルです。セキュリティ プロパティに違反すると、ノンマスカブル割り込み (NMI) が発生し、EAM モジュールにログ記録されます。GSC でサポートされているその他の保護機能については、『MSPM33 C3 シリーズ 160MHz マイコン』テクニカル リファレンス マニュアルの *GSC* の章を参照してください。

注

GSC は、BOOTRST またはより高レベルのリセットによってリセットできます。SYSRST または CPURST は GSC をリセットできません。

フラッシュ WEP 構成の場合、フラッシュはセクタ (0 ～ 64KB) またはブロック (64KB ～ 512KB) ごとに構成されます。以下のアドレスは、論理アドレスではなく物理アドレスであることに注意してください。フラッシュ バンクのスワップを行うときは、このアドレスに注意する必要があります。

表 5-1. GSC におけるセクタまたはブロックごとの WEP の構成

フラッシュ バンク	物理アドレス	レジスタ	きめ細かい制御
BANK0	0x00000000-0x0000FFFF (0-64KB)	FPC_WEPROTA0	セクタ (2KB)
	0x00010000-0x0007FFFF (64KB-512KB)	FPC_WEPROTB0	ブロック (8 セクタ、16KB)
BANK1	0x00080000-0x0008FFFF (0-64KB)	FPC_WEPROTA1	セクタ (2KB)
	0x00090000-0x000FFFFF (64KB-512KB)	FPC_WEPROTB1	ブロック (8 セクタ、16KB)
DATA フラッシュ	0x80000000-0x80007FFF (0-32KB)	FPC_WEPROTA_DFLASH	セクタ (2KB)

デバイスが 512KB のデュアル バンク デバイスの場合、すべてのバンクは 512KB ではなく 256KB になります。

注

セキュリティ権限属性、フラッシュ非表示保護、静的書き込み保護などのグローバル セキュリティ コントローラ (GSC) を構成する必要がある場合、GSC の関連レジスタは物理バンクに基づいています。バンク スワップ可能な構成では、物理アドレスに基づいて GSC を慎重に構成してください。

5.1.2 WEP での NONMAIN フラッシュ構成

ユーザーは、NONMAIN フラッシュの書き込み保護ポリシーを構成することもできます。NONMAIN フラッシュ構成を使用すると、アプリケーションの代わりに ROM ブート時に WEP を適用するように GSC を構成できます。MSPM33 に安定した書き込み消去保護を実装できます。NONMAIN フラッシュでの構成については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

注

WEP を永続的に有効にするには、NONMAIN 構成の変更を避けるために、NONMAIN 自体を保護する必要があります。

バンク 0 の静的書き込み保護の場合、BANK0_WRITE_ERASE_PROTECTION_ A フィールドと BANK0_WRITE_ERASE_PROTECTION_ B フィールドは、512KB メモリ全体を保護します。値 *0* は、対応するセクタに書き込み保護が適用されることを示します。BANK0_WRITE_ERASE_PROTECTION_ A の場合、1 ビットはフラッシュ メモリ (2KB) の 1 セクタを表します。BANK0_WRITE_ERASE_PROTECTION_ B の場合、1 ビットはフラッシュ メモリ (16KB) の 8 セクタを表します。

Copyright © 2026 Texas Instruments Incorporated

バンク 1 の静的書き込み保護の場合、BANK1_WRITE_ERASE_PROTECTION_ A フィールドと BANK1_WRITE_ERASE_PROTECTION_ B フィールドは、512KB メモリ全体を保護します。値 *0* は、対応するセクタに書き込み保護が適用されることを示します。BANK1_WRITE_ERASE_PROTECTION_ A の場合、1 ビットはフラッシュ メモリ (2KB) の 1 セクタを表します。BANK1_WRITE_ERASE_PROTECTION_ B の場合、1 ビットはフラッシュ メモリ (16KB) の 8 セクタを表します。

DATA フラッシュの静的書き込み保護の場合、DBANK_WRITE_ERASE_PROTECTION フィールドは、データ バンク メモリ全体を保護します。値 *0* は、対応するセクタに書き込み保護が適用されることを示します。DBANK_WRITE_ERASE_PROTECTION の場合、1 ビットはフラッシュ メモリ (2KB) の 1 セクタを表します。

NONMAIN を保護するように BANK0_NM_USER_CONFIG を構成すると、NONMAIN 領域全体が書き込みロックされます。これは、ブート構成ルーチンがブートストラップ ローダまたは MAIN フラッシュ内のユーザー アプリケーション コードのいずれかに実行を移行する場合、当該領域を機能的に変更不可能な状態にします。NONMAIN をアプリケーション コードまたはブートストラップ ローダでプログラムまたは消去しようとすると、ハードウェア フラッシュ動作エラーが発生し、セクタは変更されません。

表 5-2. NONMAIN におけるセクタまたはブロックごとの WEP の構成

フラッシュ バンク	物理アドレス	レジスタ	きめ細かい制御
BANK0	0x00000000-0x0000FFFF (0-64KB)	BANK0_WRITE_ERASE_PROTECTION_ A	セクタ (2KB)
	0x00010000-0x0007FFFF (64KB-512KB)	BANK0_WRITE_ERASE_PROTECTION_ B	ブロック (8 セクタ、16KB)
BANK1	0x00080000-0x0008FFFF (0-64KB)	BANK1_WRITE_ERASE_PROTECTION_ A	セクタ (2KB)
	0x00090000-0x000FFFFF (64KB-512KB)	BANK1_WRITE_ERASE_PROTECTION_ B	ブロック (8 セクタ、16KB)
DATA フラッシュ	0x80000000-0x80007FFF (0-32KB)	DBANK_WRITE_ERASE_PROTECTION	セクタ (2KB)
NONMAIN フラッシュ	0x80101800-0x80101FFF	BANK0_NM_USER_CONFIG	BCR & BSL 構成

デバイスが 512KB のデュアル バンク デバイスの場合、すべてのバンクは 512KB ではなく 256KB になります。

5.2 フラッシュ非表示保護

MSPM33 は非表示保護 (HDP) 機能をサポートしており、アプリケーションがコードの一部を一度だけ実行し、その後のコードの読み取りや実行アクセスを防止することができます。たとえば、これは第二段階の認証コードを実行する場合に特に有用な機能です。これにより、メイン アプリケーションを実行前に検証し、保存されている対称鍵または秘密鍵へのアクセスを防止できます。目的の HDP 方式は、グローバル セキュリティ コントローラ (GSC) で構成され、ユーザーは MAIN フラッシュ、DATA フラッシュ、または NONMAIN フラッシュに HDP を適用できます。『MSPM33 C3 シリーズ 160MHz マイコン』テクニカル リファレンス マニュアルの「*GSC*」の章を参照してください。

HDP が正しく構成され有効化された後、CPU が構成された HDP 領域から実行されると、HDP が有効化されます。設定済みの HDP 領域外への関数呼び出しまたはジャンプが行われると、この保護機構が自動的に有効になり、アプリケーション コードが HDP 領域へ再進入するのを防止します。アクセス、実行、読み取りは、マスク不可能割り込み (NMI) を生成します。

- メイン フラッシュバンクで HDP が有効な場合、読み取り、書き込み、実行、消去動作がブロックされます。
- メイン以外のフラッシュ バンクで HDP がイネーブルになっている場合、読み取りおよび実行動作はブロックされます。書き込みおよび消去保護を有効化する場合の正しい構成については、セクション 5.1 を参照してください。

Copyright © 2026 Texas Instruments Incorporated

注

HDP の GSC レジスタのアクセス タイプは「R/W1S」または「WOnce」です。「W1S」とは、レジスタ ビットを 0 ～ 1 の範囲でしか設定できないことを意味します。「WOnce」とは、レジスタ ビットをリセット前に 1 回しか設定できないことを意味します。これらのレジスタは BOOTRST またはそれ以上のリセット タイプでのみリセットできます。これは、印加されている HDP を直接除去できないことを意味します。

注

HDP 領域でコードを実行した後に SRAM に分岐すると、SRAM コードは HDP 領域にアクセスできます。SRAM コードの実行がすべて完了したら、HDP 機構を有効にするため、実行を分岐して HDP 領域から退出し、その後はフラッシュ上で実行を継続する必要があります。アプリケーションが SRAM から HDP 領域へのアクセスを許可しない場合、SRAM コードへ分岐する前に HDP を有効化するトランポリン関数をコードに実装する必要があります。

5.2.1 HDP でのグローバル セキュリティ コントローラ構成

グローバル セキュリティ コントローラ (GSC) は、MSPM33 でフラッシュ非表示保護を構成できるペリフェラルです。セキュリティ プロパティに違反すると、ノンマスカブル割り込み (NMI) が発生し、EAM モジュールにログ記録されます。GSC でサポートされているその他の保護機能については、『MSPM33 C3 シリーズ 160MHz マイコン』テクニカル リファレンス マニュアルの GSC の章を参照してください。

注

GSC は、BOOTRST またはより高レベルのリセットによってリセットできます。SYSRST または CPURST は GSC をリセットできません。

HDP 構成の場合、フラッシュはセクタによって構成されます。以下のアドレスは、論理アドレスではなく物理アドレスであることに注意してください。フラッシュ バンクのスワップを行うときは、このアドレスに注意する必要があります。

表 5-3. フラッシュのセクタごとに HDP を構成

フラッシュ バンク	物理アドレス	レジスタ	きめ細かい制御
BANK0	0x00000000-0x0007FFFF (0-512KB)	FPC_HDPPROT0_CONFIG	セクタ (2KB)
BANK1	0x00080000-0x000FFFFF (0-512KB)	FPC_HDPPROT1_CONFIG	セクタ (2KB)
DATA フラッシュ	0x80000000-0x80007FFF (0-32KB)	FPC_HDPPROT_DBANK_CONFIG	セクタ (2KB)

デバイスが 512KB のデュアル バンク デバイスの場合、すべてのバンクは 512KB ではなく 256KB になります。

注

セキュリティ権限属性、フラッシュ非表示保護、静的書き込み保護などのグローバル セキュリティ コントローラ (GSC) を構成する必要がある場合、GSC の関連レジスタは物理バンクに基づいています。バンク スワップ可能な構成では、物理アドレスに基づいて GSC を慎重に構成してください。

5.3 セキュアなキー ストレージ

KEYSTORE は AES キーを安全に格納できる保護された SRAM メモリであり、キーは INITDONE の前に CSC で設定されます。アプリケーションは INITDONE 後、KEYSTORE から AES エンジンへのキー転送をトリガできますが、これらのキーに直接アクセス (読み取りまたは書き込み) することはできません。キーストアの使用方法の詳細については、セクション 6.2 を参照してください。

INITDONE 以前に KEYSTORE に鍵を設定できるのは CSC のみです。その後、メイン アプリケーションは、保存されているキーのいずれかを使用するように暗号化エンジンを設定できますが、保存されているキーにはアクセス (読み取りまた

Copyright © 2026 Texas Instruments Incorporated

は書き込み) できません。KEYSTORE から暗号化エンジンへのキー転送は安全に実行され、アプリケーション コードからは見えません。

KEYSTORE の内容は BOOTRST ごとに消去され、ストアは書き込み可能な状態でアンロックされます。KEYSTORE の内容は SYSRST およびそれより下位のリセットの影響を受けません。

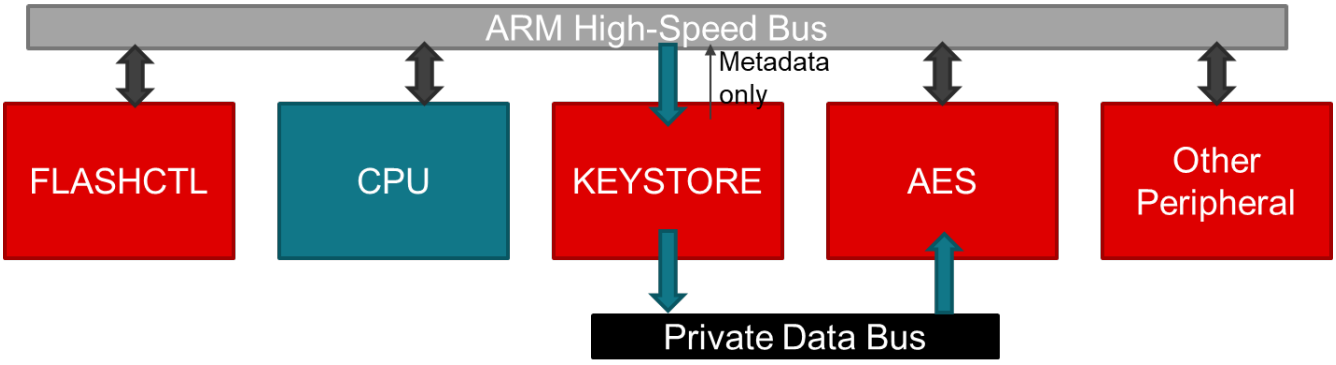

図 5-1. KEYSTORE の動作プロセス

5.4 データ整合性の検証

BCR は、BCR (ROM 内) からユーザー アプリケーション (MAIN フラッシュ メモリ内) に実行を移行する前に、MAIN フラッシュ メモリ内のユーザー指定アドレス範囲のデータ整合性をチェックすることをサポートしています。構成については、『MSPM33 NONMAIN フラッシュ メモリ構成ガイド』を参照してください。

整合性チェックは、ブート ROM (通常はセキュア ブート イメージ マネージャ) の後に最初に実行されるコードに対して、予測される値と一致する CRC/SHA256 ダイジェストが含まれていることを検証するための追加ステップとして使用できます。この整合性チェックにより、フラッシュ メモリ内の重要なコード (残りのユーザー アプリケーション ソフトウェア イメージの認証を実行) が破損しているためにセキュリティの脆弱性が発生する可能性が低減されます。

開始アドレス、長さ、および ISO-3309 CRC-32 または SHA2-256 ダイジェストを、NONMAIN 構成メモリにプロビジョニングできます。ブート プロセス中に、BCR は MAIN フラッシュ メモリ内の指定された範囲の CRC-32 ダイジェストを計算し、計算されたダイジェストをプロビジョニングされた (予測される) ダイジェストと比較して検証します。値が一致している場合は、ユーザー アプリケーションが開始します。値が一致しない場合は、ユーザー アプリケーションは開始せず、重大なブート エラーが発生します。

5.5 セキュアと特権保護

一部の MSPM33 は、TrustZone® アーキテクチャによるセキュアな保護と、PMU による特権保護をサポートしています。セクション 4 では、TrustZone® を使用してセキュアな実行環境を構築する方法について説明します。サポート ステータスを確認するには、データシートを参照してください。

ユーザーは、必要なデータやコードをセキュア領域に保存し、セキュア ストレージを実装できます。非セキュア領域または非特権領域からのアクセスは、セキュリティ / 特権違反につながります。MSPM33 のセキュアまたは特権保護の構成については、『MSPM33 マイコンにおけるセキュア アプリケーション用 Trustzone のイネーブル』を参照してください。

5.6 ハードウェア単調カウンタ

一部の MSPM33 デバイスは、NONMAIN メモリ内に OTP 領域 (2KB) をプロビジョニングします。この領域には、以下の制限があります。

- このセクタは決して消去できません (工場出荷時リセットや一括消去操作を含みます)
- HSFS、HSKP、HSSE、HSFA を含むすべてのライフサイクル ステータスで記述できます。

この OTP 領域には、ROM ブートで使用されるデバイス構成が含まれており、これは MSPM33 キー プロビジョニング、ライフサイクル維持、および HSSE デバイスのハードウェア保護されたセキュア / 特権属性構成に関連しています。

Copyright © 2026 Texas Instruments Incorporated

ユーザーは、未使用の OTP 領域にハードウェア ベースの単調カウンタ機能を実装することもできます。このセクタは追記のみでプログラムされるため、実質的には不揮発性のアップ カウンタ (ソフトウェアでの解釈によってはダウン カウンタ) として動作します。

デバイスが NONMAIN メモリ内の OTP 領域をサポートしているかどうかを確認するには、デバイス固有のデータシートを参照してください。

注

OTP 領域を消去して再プログラムすることはできません。OTP 領域 (TI 定義のレジスタ) へのプログラミング ミスは、顧客キー、ライフサイクル維持、フラッシュ上のセキュリティ属性または特権属性に対する復元不可能な誤った構成を引き起こし、マイコンのロックにつながる可能性があります。

Copyright © 2026 Texas Instruments Incorporated

6 セキュリティ モジュール

M33C デバイス ファミリに搭載されている複数のモジュールを利用することで、セキュアな実行環境を実装できます。セキュリティ モジュールには以下が含まれています。

- 高度暗号化標準 (AES)
- キーストア
- セキュア ハッシュ アルゴリズム (SHA)
- 公開鍵アクセラレータ (PKA)
- ポスト量子暗号 (PQC)
  - ML-DSA

6.1 AESADV

AESADV アクセラレータ モジュールは、FIPS PUB 197 の高度暗号化標準 (AES) に基づき、暗号化および復号処理をハードウェアで高速化します。

6.1.1 AES の概要

AES アクセラレータ モジュールは、ハードウェアで 128 ビットまたは 256 ビットのキーを使用して、128 ビットのデータ ブロックの暗号化と復号化を実行します AES は、FIPS PUB 197 で規定されている対称キー ブロック暗号アルゴリズムです。

AES アクセラレータには、次のような機能があります。

- AES 128 ビット ブロックの暗号化および復号化
- ハードウェア内でのキー スケジューリング
- ENC /復号化のみのモード：CBC、CFB-1、CFB-8、CFB-128、OFB-128、CTR/ICM
- 認証専用モード：CBC-MAC、CMAC
- AES-CCM
- AES-GCM
- AES-CCM および AES-GCM モードは、ペイロード データのホールド / レジュームによる継続をサポートしています
- 32 ビット ワードのアクセスにより、キー データ、入力データ、および出力データを供給
- AES 準備完了割り込み
- 入出力データの DMA トリガ
- RUN モードと SLEEP モードをサポート (デバイスのテクニカル リファレンス マニュアルの「動作モード」セクションを参照)

図 6-1 に、AES エンジンのブロック概略図を示します。AES エンジンは、暗号化 / 復号処理に加えてガロア体乗算も実行する処理コアで構成されています。このコアは、ソフトウェアがメモリ マップト レジスタを介して設定する構成入力およびデータ入力によって駆動されます。

Copyright © 2026 Texas Instruments Incorporated

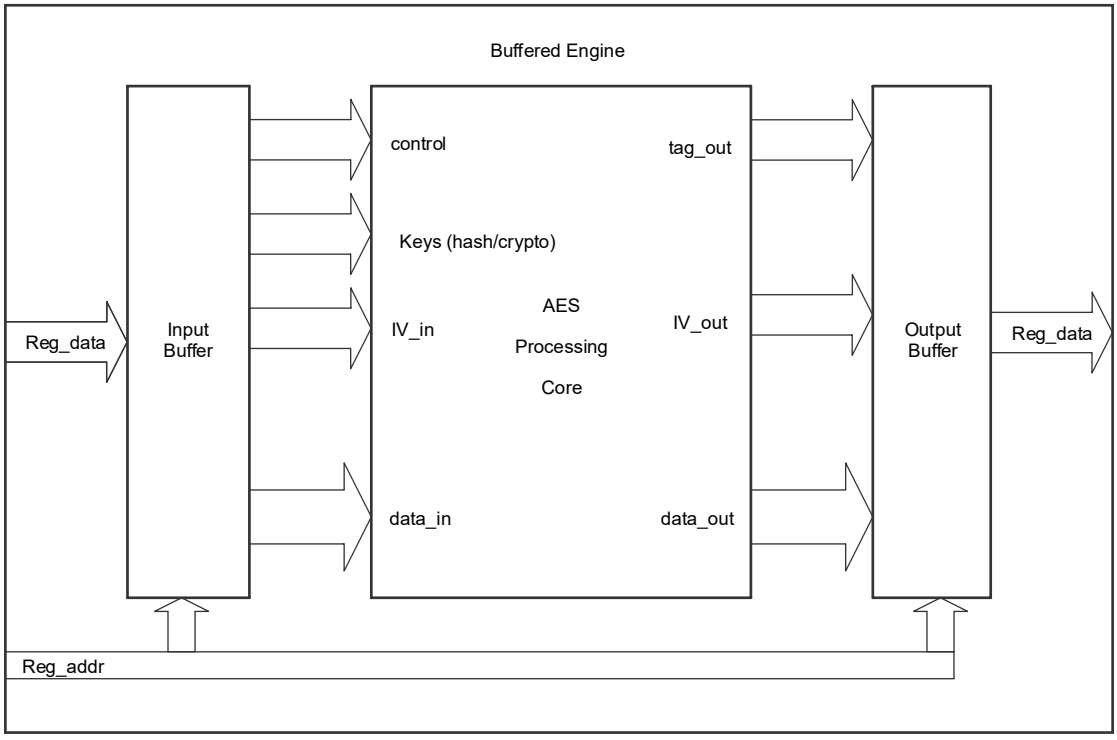

図 6-1. AES のブロック図

6.1.2 AES の使用法

AES モジュールの使用方法は、テキサス・インスツルメンツの MSPM33 ドライバ ライブラリ (DriverLib) にわかりやすく説明されています。AES DriverLib API を活用するサンプルは、MSPM33 SDK の最新リリース内の、評価基板固有の非 RTOS のサンプル サブフォルダにあります。

6.1.2.1 構成

AES モジュールの構成は、SysConfig を使用して実行できます。基本構成では、ユーザーは次のことができます。

- aesadvhp キー長の設定 (128/256 ビットのキー長)
- 操作タイプ (暗号化 / 復号化) の構成
- 暗号モードの構成

詳細については、選択したデバイスのデータシートを参照してください。

6.1.2.2 構成

AES モジュールの標準セットアップでは、次の 4 つの手順を実行します。

1. AES 状態変化監視のセットアップ
   a. CPU は、2 つの方法 (トリガされた割り込みとレジスタ ポーリング) を使用して AES 状態の変化について通知を受けることができます。AES 割り込みを使用して状態の変化を監視する場合、AES モジュールを初期化する前に、これらの割り込みを構成して有効化する必要があります。
2. AES キーのセットアップ
   a. モジュールを初期化する前に、AES キーを AES モジュールで設定する必要があります。
3. AES 暗号モードのセットアップ
   a. SysConfig 構成で選択された暗号モードは、手順 1 および 2 を実行した後に初期化できます。
4. AES モジュールの準備が完了するまで待ちます。

Copyright © 2026 Texas Instruments Incorporated

a. 手順 1 で選択した状態変更監視方法を使用して、AES モジュールの準備が完了したことが CPU に通知されます。

6.1.2.3 動作

構成およびセットアップが完了した後、データを AES モジュールにロードし、暗号化または復号化できます。データがモジュールにロードされたら、CPU は、モジュールがデータの処理を完了するまで待機してから、暗号化または復号化された出力データを読み取る必要があります。この待機は、以前に選択した状態監視方法で実行できます。いずれかの時点で動作タイプを変更した場合は、AES モジュールを再初期化する必要があります。

6.2 キーストア

6.2.1 概要

キーストア コントローラは、Advanced Encryption Engine (AES) キーの安全な管理を提供します。キーストア コントローラの使用モデルは、顧客のセキュア コードの実行中にキーを安全に格納し、その後 AES エンジンがオブザーバーにキーデータを漏らさずに安全な方法でそれらにアクセスすることです。128 ビットと 256 ビットのキーは、キーストアのキー スロットに格納できます。キーストアと AES エンジンとの相互作用は、部分的なキー変更攻撃を阻止するなど、安全な操作を可能にするように設計されています。

6.2.2 キーストアの使用方法

キーストア モジュールの使用方法は、テキサス・インスツルメンツの MSPM33 ドライバ ライブラリ (DriverLib) にわかりやすく説明されています。Keystore DriverLib API を使用したサンプルは、評価基板固有の非 RTOS サンプル サブフォルダにあります。

6.2.2.1 構成

キーストア モジュールの構成は、SysConfig を使用して行うことができます。基本構成では、ユーザーは次のことができます。

- キー ストレージ スロット (0 ～ 3) またはスロット (256 ビット キー用) の構成
- キー長の設定 (128/256 ビットのキー長)
- キー暗号転送先 (AES モジュール) の構成
- キーのロード場所の構成

詳細については、選択したデバイスのデータシートを参照してください。

6.2.2.2 構成

キーストア モジュールの標準セットアップでは、次の 2 つの手順を実行します。

1. 256 ビット キーの数のセットアッ
   a. 使用可能なキース ロットを調整できるように、キーストア モジュールに 256 ビット キーの数を通知する必要があります。すべての 256 ビット キーは、使用可能な最小のスロットに書き込まれます。
2. キー書き込みのセットアップ
   a. キーを転送する前に、キーストア モジュールのキー書き込み構成を実行する必要があります。

6.2.2.3 動作

構成とセットアップが完了後、キーストア モジュール ポインタとキー転送設定を指定することで、キー転送を開始できます。

6.3 SHA2

SHA2 (セキュア ハッシュ アルゴリズム–2) は、NIST 標準に準拠した一連の暗号化ハッシュ関数を提供します。この章では、SHA モジュールの動作について説明します。

Copyright © 2026 Texas Instruments Incorporated

6.3.1 SHA の概要

SHA 暗号化ペリフェラルは、メッセージ認証アプリケーションに使用できる FIPS 準拠のセキュア ハッシュ アルゴリズム (SHA-224、SHA-256) とハッシュベースのメッセージ認証 (HMAC) をサポートしています。

6.3.1.1 SHA の機能

- FIPS PUB 180-2 および FIPS PUB 180-3 の Secure Hash Standard に準拠した実装
- FIP PUB 198-1 に準拠するすべてのアルゴリズムをサポートする HMAC
- クロック サイクル当たり 1 回のハッシュ反復 (7.88 ビット) の性能を備え、高速なスループットを実現する高性能ハッシュ
- SHA-224 および SHA-256 の HMAC と基本ハッシュ演算をサポート
- ハッシュおよび HMAC コンテキスト スイッチング
- MAC キーの XOR およびメッセージ パディングをサポート
- アルゴリズム ブロック サイズよりも短い、等しい、または大きい MAC キーをサポートし
- メッセージ データの自動スケジューリングをサポートし
- 8 ビット単位で最大 2^{64} ビットのメッセージ サイズをサポート
- メッセージ長に基づいて DMA 要求を自動生成し、CPU のオーバーヘッドを低減
- 入力メッセージ解析用の定数アドレスと増分アドレスの両方をサポート

6.3.2 SHA の性能

表 6-1 は 32MHz で動作する MSPM33C321A を使用して、マルチブロック イメージの性能指標を提供します。

表 6-1. SHA ハードウェア アクセラレータの主な性能指標 (マルチブロック)

画像のサイズ	SHA 224 HASH	SHA 224 HMAC	SHA 256 HASH	SHA 256 HMAC
1kb	0.016ms	0.018ms	0.018ms	0.018ms
32kb	0.436ms	0.436ms	0.436ms	0.436ms
64kb	0.867ms	0.866ms	0.866ms	0.868ms
128kb	1.811ms	1.811ms	1.802ms	1.809ms

6.3.3 SHA の使用法

SHA モジュールの使用方法は、テキサス・インスツルメンツの MSPM33 ドライバ ライブラリ (DriverLib) にわかりやすく説明されています。SHA DriverLib API を活用するサンプルは、MSPM33 SDK の最新リリース内の、評価基板固有の非 RTOS のサンプル サブフォルダにあります。

6.3.3.1 構成

SHA モジュールの構成は、SysConfig を使用して実行できます。基本構成では、ユーザーは次のことができます。

- SHA モードの構成 (HASH/HMAC)
- SHA アルゴリズム (224/256) の構成

詳細については、選択したデバイスのデータシートを参照してください。

6.3.3.2 構成

SHA モジュールの標準セットアップでは、次の 4 つの手順を実行します。

1. SHA 状態監視のセットアップ
   a. CPU は、2 つの方法 (トリガされた割り込みとレジスタ ポーリング) を使用して SHA 状態の変化について通知を受けることができます。SHA 割り込みを使用して状態の変化を監視する場合、SHA モジュールを使用する前に、これらの割り込みを設定して有効化する必要があります。
2. SHA データ長のセットアップ
   a. SHA モジュールを使用する前に、SHA データの長さを設定する必要があります。
   b. データ長については、SHA モードに含まれるデータ長をバイト単位で指定する必要があります。
   c. その後、モードとデータ長が利用可能であることが SHA モジュールに通知されます。
3. MAC キーのセットアップ (HMAC モードのみ)

Copyright © 2026 Texas Instruments Incorporated

a. SHA HMAC モードを使用する場合、SHA モジュールを使用する前に、暗号化に使用する MAC キーをモジュールに渡す必要があります。
b. MAC キーは、合計キー サイズを含めて、SHA モジュールにワードで書き込む必要があります。
c. その後、キーが利用可能であることが SHA モジュールに通知される必要があります。
d. キーの処理を開始できることが SHA モジュールに通知される必要があります。
e. 最後に、CPU は SHA モジュールの処理が完了するまで待機する必要があります。

4. ダイレクト メモリ アクセス (DMA) のセットアップ
   a. SHA モジュールはデータ ブロックで動作するため、モジュールがダイジェストするにはデータを手動で供給する必要があります。これを実行するには、CPU 経由または DMA 転送によってデータを供給できます。DMA 転送を使用する場合は、SHA モジュールを使用する前に DMA チャネルを設定する必要があります。
   b. DMA ソース アドレスは、ハッシュするメッセージの先頭を指す必要があります。
   c. DMA 宛先アドレスは、SHAW_DATA_FIXED レジスタを指す必要があります。
   d. DMA メッセージの長さは、メッセージ長をワードで設定する必要があります。
   e. DMA のソースおよび宛先の幅は、ワードのサイズである必要があります。
   f. DMA ソースはシングル インクリメント モードに構成する必要があり、宛先は変更できません。
   g. DMA トリガは SHAW_TRIGGER に設定する必要があります。
   h. 最後に、DMA チャネルを有効化します。

6.3.3.3 動作

構成とセットアップが完了した後、DMA または CPU によりデータを SHA モジュールにロードできるようになります。データがモジュールにロードされたら、CPU は、モジュールがデータの処理を完了するまで待機してから、出力のハッシュされたデータを読み取る必要があります。この待機は、以前に選択した状態監視方法で実行できます。ハッシュされたデータが SHA モジュールから読み出された後、そのデータを解放して、次のハッシュ要求に使用する必要があります。

6.4 PKA

6.4.1 PKA の概要

PKA は、公開鍵アクセラレーションのための統合モジュールであり、多くのリソースを消費する公開暗号演算の計算負荷を軽減します。

6.4.1.1 PKA 機能

- PKA エンジンでは、次のような基本的な動作が行われます。
  - 大きいベクトルの加算、減算、および加算と減算の組み合わせ
  - 大きいベクトルの左または右へのシフト
  - 大きいベクトルの乗算と除算 (商の有無にかかわらず)
  - 大きいベクトルの比較とコピー
- PKA エンジンでは、次のような複雑な動作が行われます。
  - 大きいベクトルの符号なし値モジュラべき乗
  - 事前に計算された Q 逆ベクトルを用いた CRT 法による、大きいベクトルの符号なし値モジュラべき乗
  - モジュラ逆数
  - 2 種類の曲線上の ECC 演算：モンゴメリー曲線は Curve25519 や Curve448 のようなもので、$y^2 = x^3 + ax + b \pmod p$ という形の任意の曲線です。
  - 入力 / 出力としてアフィンまたは射影点を持つ楕円曲線上の ECC ポイント加算 / 2 倍算
  - 楕円曲線上の ECC 点の乗算
- 不正な状態およびタイミング攻撃の検出

6.4.2 PKA の使用法

PKA モジュールの使用方法は、テキサス・インスツルメンツの MSPM33 ドライバ ライブラリ (DriverLib) にわかりやすく説明されています。PKA DriverLib API を使用したサンプルは、評価基板固有の非 RTOS のサンプル サブフォルダにあります。

Copyright © 2026 Texas Instruments Incorporated

6.4.2.1 構成

PKA モジュールをプロジェクトに追加するだけで、SysConfig を使用して PKA モジュールの構成を行うことができます。

詳細については、選択したデバイスのデータシートを参照してください。

6.4.2.2 構成

初期化ステップが自動的に生成されるため、PKA モジュールのセットアップ要件は SysConfig によって処理されます。生成された *SYSCFG_DL_init()* 関数を呼び出すだけで十分です。

6.4.2.3 動作

構成およびセットアップが完了すると、PKA モジュールを使用して暗号操作を実行できるようになります。PKA 操作を開始してから、PKA ステータスをポーリングして操作が終了するのを待つことをお勧めします。PKA ステータスがクリアされると、PKA 操作の結果を取得できます。

6.5 PQC

ポスト量子暗号 (PQC) は、量子コンピュータによる脅威に対抗するために設計された暗号化標準のセットであり、米国国立標準技術研究所 (NIST) によって定義されています。これらの標準の 1 つに、Module-Lattice-Based Digital Signature Algorithm (ML-DSA) があります。

6.5.1 ML-DSA

6.5.1.1 ML-DSA の概要

ML-DSA は、量子コンピュータによる脅威から保護するために使用される、NIST が定義した PQC アルゴリズムの 1 つです。この標準は、顧客セキュア コード (CSC) と組み合わせて使用することで、組込みアプリケーションの真正性を確保できます。

6.5.1.2 ML-DSA の使用方法

ML-DSA 標準の使用方法は、テキサス・インスツルメンツの MSPM33 Crypto Library (暗号ライブラリ) にわかりやすく説明されています。ML-DSA 規格と MCUBoot を組み合わせて活用する例は、MSPM33 SDK の最新リリース内の、評価基板固有の非 RTOS サンプル サブフォルダにあります。

6.5.1.2.1 構成

MCUBoot で ML-DSA を使用するように構成するには、SysConfig を使用できます。基本構成では、次の操作が可能です。

- ML-DSA を使用して認証を有効にする
- ECDSA および ML-DSA を使用して認証を有効にする

詳細については、選択したデバイスのデータシートを参照してください。

6.5.1.2.2 構成

MCU ブートでの ML-DSA 使用のセットアップは、MLDSA キーとキー長を MCUBoot が使用するグローバル構造体 bootutil_keys に格納することで完了となりきます。以下にこの例を示します。

```
const struct bootutil_key bootutil_keys[] = {
    {
        .key = mldsa_pub_key,
        .len = &mldsa_pub_key_len,
    },
};
const int bootutil_key_cnt = 1;
```

6.5.1.2.3 動作

構成とセットアップが完了すると、MCUBoot は、boot_go が開始時にアプリケーションイ メージを検証するために、ML-DSA および事前に設定した公開 ML-DSA キーを使用します。

Copyright © 2026 Texas Instruments Incorporated

7 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。

Changes from December 16, 2025 to November 12, 2026 (from Revision * (December 2025) to Revision A (November 2026)) **Page**

- 「デバイス セキュリティ モデル」セクションとすべてのサブセクションを追加....6
- 「セキュア ブート」セクションとすべてのサブセクションを追加 22
- 「セキュア ストレージ」セクションとすべてのサブセクションを追加....42
- 保存済みコンテキスト準備完了割り込みの使用方法に関する注記を追加。レジスタを更新。.... 48
- DMA でのモジュールの使用法を明確化。....50

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