注意: この日本語版文書は参考資料としてご利用ください。
最新情報は必ずオリジナルの英語版をご参照願います。

ECC608-TMNGTLS CryptoAuthentication™ データシート

ECC608-TMNGTLS

はじめに

ECC608-TMNGTLSは、ATECC608のTrust Managerプロビジョニング済みバージョンです。ECC608-TMNGTLSデバイスはKudelski IoTのkeySTREAM™ SaaS(サービスとしてのソフトウェア)と組み合わせて動作します。このデバイスにはkeySTREAM SaaSに接続するための一連の暗号鍵が事前プロビジョニングされています。ECC608-TMNGTLS搭載IoTデバイスは市場に投入されると、キーストリームSaaSに接続します。このSaaSはカスタムPKI(公開鍵基盤)によってデバイスを「フィールドで」プロビジョニングし、IoTデバイスの所有権をしかるべき所有者に付与します。秘密情報の交換に物理的な介入は一切必要ないため、完全にゼロタッチのソリューションです。暗号鍵はセキュアな認証ICの物理的な境界内で保護されながら、遠隔的かつ動的に管理されるようになります。このソリューションを使う事で、IoTデバイス展開後の鍵管理に関連する製品ライフサイクル管理にセキュリティ対策を確実に導入できます。

本データシートにはECC608-TMNGTLSデバイスのスロットおよび鍵コンフィグレーション情報を記載しています。これらの情報は、各Dataゾーンスロットのアクセスポリシーを定義します。本書には一部のコマンドとI/O動作情報のみを記載しています。ユーザー アプリケーションの開発に役立つMicrochip社のハードウェアおよびソフトウェアツールを紹介する特定のセクションも含まれています。ガイド付きのユースケースをMicrochip社のTPDS (Trust Platform Design Suite)で活用できます。必要な秘密情報と秘密鍵はkeySTREAM SaaSを通じてフィールドでプロビジョニングされます。

特長

- 完全に設定済みのConfigurationゾーン
- I^2Cインターフェイス - I^2Cアドレスは一度だけ変更可能
- JIL評価「高」 - 「JIL Application of Attack Potential to Smartcards and Similar Devices, Version 3.1」に基づいて検証済み
- 高品質NIST SP 800-90A/B/C TRNG(真性乱数生成器)を内蔵、NIST CMVP ESV認証済み
- Kudelski keySTREAM SaaSと連携する定義済みスロット アクセスポリシー:
  - ECC P-256デバイスID鍵
  - ECC P-256アテステーション秘密鍵
  - お客様固有ID情報
- 以下のフィールドでのプロビジョニング用スロット:
  - デバイスおよび署名者圧縮証明書スロット
- I/Oレベル: 1.8～5.5 V
  電源電圧レベル: 2.0～5.5 V
- 標準産業用温度レンジ: -40～+85 °C
- 公称スリープ電流: 30 nA
- 8ピンUDFNおよび8ピンSOICパッケージで提供され、プロトタイプ用は10個固定のリールサイズ、量産用のMOQ(最小注文数量)は2kです。

ユースケース

- カスタムのルートCAのセットアップと関連PKIを含むセキュアなIoT TLS 1.2および1.3の接続
- ローテーション、失効、更新を含む動的な証明書管理
- セキュアブート
- エンドツーエンドのデータ保護
- セキュリティの敏捷性を高める秘密鍵ローテーション

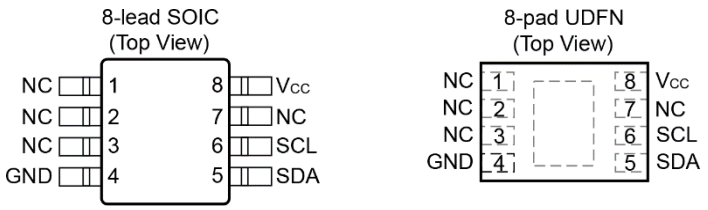
目次

1. ピンの構成と配置

表1-1. ピン配置

ピン名	機能
NC	未接続
GND	グランド
SDA	I^2Cシリアルデータ
SCL	I^2C Serial Clock Input
V_{CC}	電源

図1-1. UDFNおよびSOICパッケージのピン配置



Note: UDFNパッケージの底面パッドは、GNDに接続する事を推奨します。

関連リンク

10.2. 8ピンUDFN
10.3. 8ピンSOIC

2. Trust Manager体験

ECC608-TMNGTLSのセキュア認証ICは、プロジェクトのライフサイクル全体を通して高度なセキュリティを維持しながらIoT製品のハードウェア セキュリティをセルフサービスの体験にする事に重点を置いています。

keySTREAM SaaSが提供するもの

keySTREAM SaaSサービスはルートCAと関連PKIをセットアップし、それをECC608-TMNGTLSのセキュア境界内にTLS暗号鍵、データ、証明書と共にフィールドでプロビジョニングするのに必要なHSM(ハードウェアセキュリティ モジュール) インフラストラクチャを提供します。これにより、製品のライフサイクル全体を通してこれらの暗号認証情報がアクティブに管理される事になります。このソリューションにより以下の事が可能になります。

- 任意のクラウド プラットフォームへのセキュアな認証を確実なものにする証明書の管理。証明書の有効期限切れによってIoT製品が中断されないように証明書を更新またはローテーションできます。その結果、デバイスの管理コストと複雑性が軽減されます。
- このIaaS(インフラストラクチャに依存しないSaaS)はAWS®とMicrosoft Azure®でそのまま実行できます。
- 暗号鍵のkeySTREAM HSMへホスティングのコスト効率の良さと、関連する可用性と保守性

keySTREAM SaaS製品はメモリに制約のあるMCUからMPU製品に至るまで幅広く適合する省スペースな組み込みライブラリ(keySTREAM Trusted Agent KTA)を備えています。これにより、メモリに大きく負担をかける事なく、幅広い複雑さのデバイスにセキュリティを実装できます。これらのツールはIoT製品の幅広いニーズに合わせて拡張できるように設計された電気通信グレードのクラウド ホスティング プラットフォームに組み込まれています。

Trust Managerのフロー

ECC608-TMNGTLSは比較的簡単なプロセスで使い始められるよう設計されています。

1. Kudelski IoTアカウントに申し込み、keySTREAM Security Management Servicesへのアクセス権を取得します。お客様の製品が複製不可能な一意の識別情報を持つよう、お客様の会社およびプロジェクトに関する一意の情報を提供します。keySTREAM SaaSの自動請求プロセスに従い、必ずMicrochip社注文システムで使われた購入者のメールアドレスを登録してください。
2. ECC608-TMNGTLSデバイスを注文します。Microchip社はデバイス評価の初期費用を抑えるために10個入りのサンプルパックを用意しています。これらのデバイスは既製品として提供され、購入前に手動でプロビジョニングする必要はありません。keySTREAM SaaSは特定のデバイスがMicrochip社によるkeySTREAM SaaSリモート管理用にプロビジョニングされているかを検証できます。
3. keySTREAM SaaSを使ってカスタムのルートCAと関連するPKIを作成します。
4. keySTREAM SaaSサービスを使ってmicrochipDIRECTで購入したデバイスを請求します。その後、組み込みシステムがKudelski keySTREAM SaaSに接続されると、フィールドでのプロビジョニングが実行されます。keySTREAM SaaSサービスにより、ホスティングおよび管理されたHSMにおけるカスタムのPKIの作成、一意のお客様情報の入力、証明書の作成、セキュアな通信のためのI/O保護鍵の作成が可能になります。
5. 製品をフィールドに展開します。
6. フィールド管理機能を使って鍵のローテーションや、証明書の更新またはローテーションを実行します。

3. EEPROMメモリの構成とDataゾーン アクセスポリシー

EEPROMメモリは1,400バイトの総容量を持ち、以下のゾーンに分割されます。

表3-1. ECC608-TMNGTLSのEEPROMゾーン

ゾーン	概要	命名法
Configuration	以下を格納する128バイト(1,024ビット)のEEPROMゾーン • デバイス コンフィグレーション • スロット アクセスポリシー情報 • カウンタ値 • デバイスシリアル番号 • ロック情報 LockConfigバイトは設定済みです。Configurationゾーンに直接書き込む事は一切できません。Configurationゾーンは常に読み出し可能です。	Config[a:b] = Configurationゾーンの1フィールドに含まれるバイトのレンジ
Data	1,208バイト(9.7 Kビット)のゾーンが16個の読み出し専用または読み書き可能汎用メモリスロットに分割されます。Configurationゾーン内のバイトによって定義されるアクセスポリシー情報により、各スロットへのアクセス方法が決まります。ECC608-TMNGTLS内の各DATAゾーンスロットのアクセスポリシーは設定済みであり、Configurationゾーンによって定義されたスロット アクセスポリシーは完全な効力を有します。	Slot[YY] = DataゾーンのスロットYYに保存された内容
OTP (One Time Programmable)	64バイト(512ビット)のゾーンが2つの32バイトブロックに分割されます。ECC608-TMNGTLSでは、OTPゾーンには定義値が書き込み済みです。OTPゾーンは変更できませんが、いつでも読み出せます。詳細は3.3. ECC608-TMNGTLSのEEPROM OTP (One Time Programmable)ゾーンを参照してください。	OTP[bb] = OTPゾーン内の1バイト、OTP[aa:bb] = OTPゾーン内のバイトのレンジ

表3-2. 本書内の用語の意味

本書内で使う用語の意味を下表に示します。

用語	意味
ブロック	1ブロックは、特定メモリゾーン内の256ビット(32バイト)領域です。業界標準のSHA-256文書でもブロックという用語が使われますが、これはメッセージ入力の512ビットセクションを意味します。本書では、「ブロック」はハッシュ入力メッセージに言及する際にのみ使われます。
KeyID	KeyIDは、鍵値を保持するために割り当てられたスロットの番号です。例えば、Key 1 (Key[1]と表記する場合もあり)は、Slot[1]に保存されます。16個ある全てのスロットは鍵を保持できますが、平文読み出しを許可するよう設定されたスロットが暗号コマンドによって公開鍵または秘密鍵として使われる事は一般的にありません。
mode[b]	Modeパラメータのビットbを示します。
SRAM	入力および出力バッファと内部状態保存領域を格納します。ユーザーが直接このメモリにアクセスする事はできません。
ワード	1ワードは、ブロックに対して読み書きする4バイトのデータです。ワードはデータアクセスの最小単位です。
LSB/MSB	最下位バイト(LSB)/最上位バイト(MSB)です。
LSb/MSb	最下位ビット(LSb)/最上位ビット(MSb)です。

関連リンク

3.1. ECC608-TMNGTLSのConfigurationゾーン
3.2. Dataゾーンとアクセスポリシー
3.3. ECC608-TMNGTLSのEEPROM OTP (One Time Programmable)ゾーン

3.1. ECC608-TMNGTLSのConfigurationゾーン

ECC608-TMNGTLSのConfigurationゾーンは固定されており、ユーザーによる変更はできません。デバイスのコンフィグレーションに関連する情報を以下またはスロット情報に示します。鍵はkeySTREAM SaaSによってフィールドでプロビジョニングされます。

デバイス コンフィグレーション情報

- 各デバイスのシリアル番号は一意であり、バイト[0:3, 8:12]に保存されます。バイト[0:1]は0x01 0x23、バイト[8]は0x01です。その他の全てのバイトは一意です。
- 既定値の7ビットI²Cアドレスは0x38です。書き込みに使われるI²C 8ビット アドレスバイトは0x70で、読み出しに使われるバイトは0x71です。I²Cアドレスは、`UpdateExtra`コマンドを使って上書きできます。

> **Important:** ECC608-TMNGTLSの既定値のI²Cアドレスは標準のECC608デバイスのものとは異なります。

- セキュアブート公開鍵(スロット15)、セキュアブート ダイジェスト (スロット13)、I/O保護(スロット7)を除く全てのデータスロットはKudelskiによって管理されます。
- AES128動作は有効です。
- ChipModeの特長
  - クロック分周器は実行時間を最大化するために設定されています。
  - WDT(ウォッチドッグ タイマ)は最大タイムアウト(1.3 s)に設定されています。
  - I/Oレベルは固定の参照電圧レベルに設定されているため、ホストプロセッサはECC608-TMNGTLSの動作電圧よりも低電圧で駆動可能です。
  - 代替I²Cアドレスの設定が有効です。
- SecureBoot機能
  - FullStore Digestモードが有効です。
  - セキュアブートECC P-256公開鍵はスロット15に保存され、製造時に読み込まれます。
  - セキュアブート ダイジェストはスロット13に保存されます。
  - ランダムノンス (ワンタイム トークン)は必須ではありませんが、使用を推奨します。
  - セキュアブート持続性ラッチは無効です。
  - I/O保護鍵でコマンドの出力を暗号化
- チップオプション機能
  - I/O保護鍵の使用は、スロット7に保存された鍵により有効になります。
  - KDF機能の出力は平文ですが、`KDF`コマンドのモードに基づいて暗号化できます。
  - KDF AESモードは有効です。
  - ECDHマスタ秘密鍵の出力は平文ですが、`ECDH`コマンドのモードに応じて暗号化できます。
  - ヘルステストに合格しなかったためにコマンド実行が失敗した場合、常にHealth Test Failureビットがクリアされます。失敗の症状が過渡的であった場合、コマンドは再実行時にテストに合格します。
  - パワーオン セルフテストは無効です。必要な場合、セルフテストを明示的に実行する必要があります。
  - ヘルステストに合格しなかったためにコマンド実行が失敗した場合、常にHealth Test Failureビットがクリアされます。失敗の症状が過渡的であった場合、コマンドは再実行時にテストに合格します。

- モノトニック カウンタはどの鍵にも割り当てられておらず、システムによって任意に使えます。
- Configurationゾーンは恒久的にロックされており、コンフィグレーションの更新は禁止されています。

関連リンク

3.2.3. ECC608-TMNGTLSのスロット設定のまとめ

3.2. Dataゾーンとアクセスポリシー

以下では、各スロットに割り当てられているアクセスポリシー情報について説明します。実際のアクセスポリシー情報は、EEPROM Configurationゾーン内のSlotConfigおよびKeyConfigセクションに保存されます。各Dataゾーンスロットには、2つのSlot Configurationバイトと2つのKey Configurationバイトが割り当てられています。これらの4バイトにより、各スロットのアクセスポリシーが構成されます。スロットに保存されるデータのタイプは、そのスロットのアクセスポリシーによって決まります。

3.2.1. Dataゾーンのデータタイプ

以下では、ECC608-TMNGTLSのデータスロットに保存できる各種データタイプについて説明します。

3.2.1.1. 秘密鍵

ECC P-256秘密鍵は、ECC608- TMNGTLSのECCセキュリティの基本構成要素です。これらの鍵は非公開かつ各デバイスに対して一意であり、読み取りできません。通常、ECC秘密鍵はセキュア エレメントのTRNGによりランダムに生成され、ECC秘密鍵として設定されたスロットでセキュアに保持されます。

プライマリ秘密鍵

これはプライマリ認証鍵で、スロット0に保存されます。各デバイスは独自の一意秘密鍵を保有します。この鍵はスロット0がロックされていない限り変更できます。

この鍵は、以下の2つのプライマリ楕円曲線関数向けに使えます。

- 認証のためのECDSA署名
- 鍵合意のためのECDH。ECDH出力を暗号化する必要がある場合、I/O保護鍵を最初にセットアップする必要があります。セットアップの詳細は3.2.1.7. I/O保護鍵を参照してください。

このプライマリ秘密鍵は、対応する公開鍵とX.509証明書を生成するための基本の鍵です。

セカンダリ秘密鍵

スロット1とスロット3には、追加のサービスに利用可能な追加の秘密鍵が格納されています。これらの鍵はKudelskiツールで使うために予約されています。

これらの鍵は、以下のプライマリ楕円曲線関数向けに使えます。

- 認証のためのECDSA署名
- 鍵合意のためのECDH。ECDH出力を暗号化する必要がある場合、I/O保護鍵を最初にセットアップする必要があります。セットアップの詳細は3.2.1.7. I/O保護鍵を参照してください。
- GenKey
  内部で生成した新しいランダム秘密鍵をスロットに上書きするために使います。

これらのその他の鍵は内部生成した新しい鍵(`GenKey`コマンド、モード = 0x04を使用)で上書きする事により、鍵の削除およびローテーションとリモート プロビジョニングが可能です。これらの鍵はスロットのロックも可能です(KeyConfig.Lockableビット = 0に設定)。すなわち、SlotLockモードで`Lock`コマンドを使う事により、現在の鍵を恒久的に保存できます(`GenKey`コマンドによる変更から保護できます)。鍵を変更する際は、新しい鍵が本当に本デバイスから生成された物である事を他のシステムに対して証明するために、「鍵の証明(Key Attestation)」が必要です。

鍵の証明(Key Attestation)

スロット1内の秘密鍵は、内部署名専用の鍵として設定されます。すなわち、この鍵は、GenKeyまたはGenDigコマンドによって内部で生成されたメッセージに署名する事しかできません(この鍵を使って外部メッセージに署名する事はできません)。この機能は、本デバイスを使う側に、デバイスの署名者鍵(公開鍵)と鍵に関わるデバイスの設定情報が信頼できる事を証明します。

3.2.1.2. 公開鍵

公開鍵はECC秘密鍵と関連付けられます。各ECC秘密鍵は独自の一意公開鍵を持ちます。デバイスに保存された秘密鍵に関連する公開鍵はGenKeyコマンドを使って取得できます。チップ外に保存されたECC秘密鍵からの公開鍵は、デバイスの公開鍵用に設定されたスロットに保存できます。

ECC608-TMNGTLSでは、公開鍵はスロット11、14、15に保存されています。

3.2.1.3. 証明書の保存

ECC608-TMNGTLSのストレージは、鍵を安全に保存できるように構成されています。X.509証明書は、ECC608-TMNGTLSの1スロットのサイズより大きくなる傾向があるため、圧縮フォーマットを使います。この方法は、動的な証明書情報をデバイスに保存してある程度の制限を課するため、部分的証明と呼ばれる手法よりも優れます。動的な情報とは、デバイスごとに異なると期待できる証明書内容です(例: 公開鍵、有効期限等)。ファームウェアには、静的情報(全ての証明書で同じ内容のデータ)を格納した完全なX.509証明書のテンプレートによる証明書定義と、圧縮された証明書内の動的情報から完全な証明書を再構成するための機能を備えている事が求められます。

以下のアプリケーション ノートに、圧縮された証明書のフォーマットが記載されています。ATECC圧縮証明書の定義

CryptoAuthLibライブラリにも、圧縮された証明書向けのAPIが含まれています。

デバイス証明書

デバイス証明書は、実際のエンドユニットに関する情報で構成されます。ECC608- TMNGTLSの場合、圧縮デバイス証明書はスロット10に保存されます。

署名者証明書

署名者証明書は、デバイス証明書に署名するために使われた署名者認証局に関連する情報で構成されます。ECC608-TMNGTLSの場合、圧縮署名者証明書はスロット12に保存されます。完全な署名者証明書を再構成するには署名者公開鍵も必要です。

署名者公開鍵

署名者公開鍵は、署名者と圧縮された署名者証明書に関する情報を検証するために必要な公開鍵です。ECC608-TMNGTLSの場合、署名者公開鍵はスロット11に保存されます。

下表に、証明書に関係するECC608-TMNGTLS内の全てのスロットを示します。

スロット	概要
0	プライマリ秘密鍵: この公開鍵は、GenKeyコマンドをMode = 0x00で使う事により、いつでも生成できます。
10	デバイス証明書: この証明書は圧縮フォーマットでこのスロットに保存されます。
11	署名者公開鍵です。
12	署名者証明書: この証明書は圧縮フォーマットで保存されます。

ECC608-TMNGTLSでは、個々のスロットがロックされていない限りこれらのスロットは上書きできます。

関連リンク

3.1. ECC608-TMNGTLSのConfigurationゾーン
3.2.4. ECC608-TMNGTLSスロット アクセスポリシーの詳細
5.5.3. 証明書の保存

3.2.1.4. セキュアブート

ECC608-TMNGTLSでは、SecureBootコマンドは有効です。このため、システムは完全なブートを実行する前に、ブートローダを介してファームウェアを暗号論的に検証できます。この機能を使うと、新しいファームウェア イメージをロードする前に検証する事もできます。

セキュアブート機能を使う前に、ECC P-256ファームウェア署名鍵を確立する必要があります。秘密鍵は、ファームウェア イメージの署名用に、ファームウェア開発者が保有します。委託メーカーでの製造時に、公開鍵を「セキュアブート公開鍵」スロットに書き込み、スロットをロックする事で鍵を恒久的とする必要があります。

セキュアブートを実装するには、各種のデータスロットが必要です。

セキュアブート ダイジェスト

セキュアブート ダイジェストは、ファームウェア アプリケーション コードに対して計算される32バイトのSHA-256ダイジェストです。このダイジェストは、ファームウェアを更新するたびに更新する必要があります。ECC608-TMNGTLSの場合、ダイジェストはスロット13に保存されます。

セキュアブート公開鍵

セキュアブート公開鍵は、セキュアブート ダイジェストと署名を有効にするための検証機能を実行するために使われます。セキュアブート公開鍵はスロット15に保存されます。

I/O保護鍵

I/O保護鍵はスロット7に保存されており、SecureBootコマンドの出力を保護するために使う事ができます。

関連リンク

3.1. ECC608-TMNGTLSのConfigurationゾーン
3.2.4. ECC608-TMNGTLSスロット アクセスポリシーの詳細
6.2.3. SecureBootコマンド

3.2.1.5. 秘密鍵

スロット5は、32バイトの秘密鍵を保存するために使えます。この鍵は、ECC608-TMNGTLSの対称鍵コマンド(GenDig、MAC、CheckMac、KDF、SHA/HMAC、AES)で使えます。

この鍵はスロット6に保存された暗号鍵を使って暗号化書き込みによって書き込む必要があります。

3.2.1.6. AES鍵ストレージ

ECDHおよびKDF等のコマンドは対称鍵を出力します。オプションにより、これらの鍵はスロットに安全に保存してAES鍵として使えます。AES鍵ストレージ スロットは、これらの鍵の保存先として設定済みです。1～4個の16バイトAES128鍵をデータスロット9に保存できます。

3.2.1.7. I/O保護鍵

Verify、ECDH、SecureBoot、KDFコマンドでは、オプションによりI/O保護機能を使って一部のパラメータを暗号化し、一部のレスポンスを検証できます(MACコマンドによる)。これは物理的I^2Cバスに対する中間者攻撃から保護するために役立ちます。I/O保護鍵はセキュアブート中にリプレイ攻撃を防ぐためにも役立ちます。しかし、この機能を使う前に、MCUとECC608-TMNGTLSは一意のI/O保護鍵を生成

して保存する事で、原則的に互いにペアリングする必要があります。ペアリング処理はお客様の製造施設で実施する必要があります。

I/O保護鍵の生成:

1. MCUはrandomコマンドを使ってランダムな32バイトI/O保護鍵を生成します。
2. MCUはI/O保護鍵をMCU内部のフラッシュに保存します。
3. MCUはI/O保護鍵をI/O保護鍵スロットに書き込みます。
4. MCUはそのスロットをロックする事で、I/O保護鍵を恒久的とします。

ペアリングを確認するため、MCUはMACコマンドを使ってチャレンジをI/O保護鍵に対して発行し、フラッシュに保存されているI/O保護鍵とECC608-TMNGTLS内のI/O保護鍵が一致するか検証します。

3.2.1.8. 一般的データストレージ

スロット15は、セキュアブートが未実装でこのスロットにセキュアブート公開鍵が格納されていない場合、汎用データストレージとして使う事ができます。このスロットは、制約なしにアクセスできる任意のデータを保存するために使え、常に平文で読み出し/書き込みできます。

3.2.2. スロット設定の用語

以下に、設定オプションの説明に使う用語をアルファベット順に記載します。

用語	意味
AES Key:	AESコマンドの鍵ソースと使えるスロット。ECC608- TMNGTLSでは、AES鍵は128ビット幅を持つ。
Always Write:	Writeコマンドを使っていつでも平文で書き込めるスロット
Clear Read:	パブリックである(秘密ではない)とみなされ、その内容をReadコマンドを使って平文で読み出せるスロット
ECDH:	楕円曲線ディフィー ヘルマン(Elliptic Curve Diffie Hellman)。ECDHコマンドで使える秘密鍵
Ext Sign:	外部(任意の)メッセージを署名するために使える秘密鍵
Int Sign:	GenKeyまたはGenDigコマンドによって生成された内部メッセージに署名するために使える秘密鍵。デバイス内部の鍵と設定を証明するために使用
Lockable:	将来のある時点でロックできるスロット。一度ロックしたスロットの内容は変更できない(読み出し/使用のみ可能)。
No Read:	秘密であるとみなされ、Readコマンドを使って読み出す事ができないスロット。秘密鍵と対称秘密鍵は常に「No Read」として設定する必要がある。
No Write:	Writeコマンドを使って変更できないスロット
Permanent:	恒久的で変更不可能な秘密鍵。この秘密鍵は工場でのプロビジョニング中に内部生成される。
Updatable:	後でランダムに内部生成された秘密鍵により上書き可能な秘密鍵。初期値は工場でのプロビジョニング中に内部生成される。

3.2.3. ECC608-TMNGTLSのスロット設定のまとめ

ECC608-TMNGTLSは、Trust Managerソリューションの一部として特定の目的のために設定される16個のスロットを備えています。スロットは複数のタイプに分類されます。これを下表のスロットタイプ列に示します。以下のスロットタイプが定義されています。

- K = Kudelski専用のkeySTREAM SaaSスロット
- P = keySTREAM SaaSによってプロビジョニングされるアプリケーション スロット。この情報はお客様のアプリケーションで使われますが、keySTREAM SaaSの管理下にあります。
- C = お客様プロビジョニング用スロット。この情報を知っているのはお客様のみで、お客様の製造ラインで更新されます。
- A = アプリケーション特化プロビジョニング用スロット。このスロットはランタイムにプロビジョニングされます。
- R = 予約済み。このデータシート作成時点では未使用ですが、将来のアプリケーションで使われる可能性があります。

スロット	スロットタイプ	鍵名	概要
0	P	デバイスID鍵	スロット10に保存されているデバイス圧縮証明書に署名するために使われるプライマリ秘密鍵。
1	K	アテステーション鍵	デバイスの真正性を証明するためにKudelskiによって使われる秘密鍵。対応する公開鍵を知っているのはKudelskiのみです。
2	K	封印ID	このスロットに保存されている一意のプロファイルUID(ユーザーID)。
3	K	非対称鍵	Kudelskiによって管理、予約済み
4	K	対称鍵	Kudelskiによって管理、予約済み
5	R	フィールドでのプロビジョニングのための対称鍵	お客様の対称鍵を保持するスロット。このスロットはkeySTREAM™ SaaSのみによって更新できます。スロットは暗号化書き込みに設定されており、WriteKeyはスロット6にあります。
6	R	暗号化書き込み鍵	スロット5とスロット14のための暗号化WriteKeyで、Microchip社の施設でプロビジョニングされます。ペアレント鍵を知っているのはKudelskiのみです。保存された鍵はDiversifyされています。
7	C	I/O保護鍵	お客様のI/O保護鍵を保持するスロット。このスロットはお客様の製造ラインでお客様により更新されます。
8	K	keySTREAM SaaS オンボーディング データ	複数のkeySTREAM SaaS動作に使われるKudelski固有のデータ。
9	A	AES128鍵	更新可能なお客様のAES128鍵。
10	P	デバイス圧縮証明書	圧縮デバイス証明書。
11	P	署名者公開鍵	Kudelski署名者に関連する公開鍵。
12	P	署名者圧縮証明書	Kudelski圧縮署名者証明書。
13	P	セキュアブート ダイジェスト	セキュアブート ダイジェストを保持するスロット(Stored Digestモード)。セキュアブート コマンドを使って内部でのみ更新できます。
14	R	フィールドでのプロビジョニングのための公開鍵。	お客様固有の公開鍵を保持するスロット。このスロットはkeySTREAM SaaSのみによって更新できます。スロットは暗号化書き込みに設定されており、WriteKeyはスロット6にあります。
15	P	セキュアブート公開鍵またはCデータ	お客様のセキュアブート公開鍵を保持するためのスロット。このスロットはお客様の製造ラインでプロビジョニングされる事を前提としています。

関連リンク

3.2.4. ECC608-TMNGTLSスロット アクセスポリシーの詳細

3.2.4. ECC608-TMNGTLSスロット アクセスポリシーの詳細

各データスロット用のECC608-TMNGTLSスロット アクセスポリシーは完全に設定済みです。各スロットにはそれぞれ異なる目的があります。ほとんどの場合、特定のユースケースで適切な動作とセキュリティを実現するには全てのスロットが必要です。スロットはkeySTREAM SaaS動作環境の一部として主にKudelskiによって使われるKスロットに分類されます。Cスロットはお客様の特定の用途用です。

スロットロック オプション

スロットロック オプションは、各スロットで以下のどちらかに設定できます。

Lockable: このスロットロック オプションに設定されたスロットは、初期製造後の任意時点でエンドユーザーによるロックが可能です。このオプションにより、本デバイスがMicrochip社から出荷された後の製造工程中に鍵またはデータを設定できます。または、エンドユーザーによる鍵の設定が可能となります。このオプションが設定されたスロットは、`Lock`コマンドを使ってロックできます。スロットが一度ロックされると、その中のデータは二度と変更できなくなります。

Permanent Lock: デバイスがMicrochip社の工場から出荷された後は、恒久的にロックされたスロットを変更する事は決してできません。これらのデバイスのプロビジョニングに先立ち、正しいデータと鍵をMicrochip社に提供する必要があります。

スロット設定の詳細

以下の各表に、本デバイスの各スロットにおけるスロットと鍵の詳細な設定を示します。鍵とスロットの設定に関連する情報を「有効機能」の一部として示します。特定の鍵タイプは鍵名の後の丸括弧()内に示します。鍵タイプの説明は3.2.3.「ECC608-TMNGTLSのスロット設定のまとめ」に記載しています。

表3-3. スロット0: デバイスID鍵(P)

スロット	設定値	有効機能
0	鍵:	• P256 NIST ECC秘密鍵を格納します。 • 秘密鍵を生成できます。 • 公開鍵は秘密鍵から生成できます。 • コマンド実行前にランダムノンスが必要です。 • スロットは個別にロックできます。
	スロット:	• このスロットは秘密です。 • 外部メッセージの署名に使えます。 • ECDHコマンドでセッションキーを生成するのに使えます。

表3-4. スロット1: アテステーション鍵(K)

スロット	設定値	有効機能
1	鍵:	• P256 NIST ECC秘密鍵を格納します。 • 鍵はプロビジョニング時に書き込まれます。 • 対応する公開鍵を生成する事はできません。
	スロット:	• このスロットは秘密です。 • GenDigまたはGenKeyコマンドによって内部生成されたメッセージに署名できます。

表3-5. スロット2: 封印ID (K)

スロット	設定値	有効機能
2	鍵:	• ユーザーを識別するために使われるデータが格納されており、デバイスのプロファイルUIDが格納されます。 • スロットは個別にロックできます。
	スロット:	• 常に平文読み出しが可能です。 • 常に平文書き込みが可能です。

表3-6. スロット3: 非対称鍵(K)

Note: このスロットはKudelski IoT専用です。

スロット	設定値	有効機能
3	鍵:	• P256 NIST ECC秘密鍵を格納します。 • 対応する公開鍵はいつでも生成できます。 • ランダムノンスが必要です。 • スロットは個別にロックできます。
	スロット:	• このスロットでは、ロックする前にGenKeyを使って新しいECC秘密鍵を生成できます。 • このスロットは秘密です。 • 外部メッセージに署名できます。 • ECDHコマンドでセッションキーを生成するのに使えます。

表3-7. スロット4: 対称鍵(K)

Note: このスロットはKudelski IoT専用です。

スロット	設定値	有効機能
4	鍵:	• スロットにはAESセッションキーが含まれます。 • スロット3に格納されている鍵を使ってECDHコマンドの出力を格納します。 • スロットは個別にロックできません。
	スロット:	• スロットは秘密であり、読み出しはできません。 • スロットは直接書き込みできません。

表3-8. スロット5: フィールドでのプロビジョニング用対称鍵(R)

スロット	設定値	有効機能
5	鍵:	• AES鍵を格納しています。 • 常にランダムノンスが必要です。
	スロット:	• 内容は秘密であり、決して読み出しできません。 • 暗号化データは、スロット6内の鍵を使ってこのスロットに書き込む事ができます。 • 暗号化書き込みにはMACが必要です。

表3-9. スロット6: 暗号化書き込み鍵(R)

スロット	設定値	有効機能
6	鍵:	• データを暗号化する鍵が格納されています。 • この鍵を使うにはランダムノンスが必要です。 • このスロットは恒久的にロックされています。
	スロット:	• このスロットからの読み出しは禁止されています。 • このスロットへの書き込みは常に禁止されます。

表3-10. スロット7: I/O保護鍵(C)

スロット	設定値	有効機能
7	鍵:	• I/Oデータを暗号化する鍵が格納されています。 • この鍵を使う前にランダムノンスが必要です。 • スロットは個別にロックできます。
	スロット:	• このスロットは読み込みできません。 • このスロットに対して平文書き込みが可能です。

⚠ CAUTION 一般的に、スロット7に保存されたI/O保護鍵はSlot Lockableオプションのままにする必要があります。ほとんどの場合、I/O保護鍵は各デバイスで一意です。ユースケースによって、全てのデバイスで同じI/O保護鍵を使う場合、Permanent Lockオプションを選択できます。

表3-11. スロット8: keySTREAM™ SaaSオンボーディング データ(K)

スロット	設定値	有効機能
8	鍵:	• データスロットはKudelskiがお客様をオンボーディングする用途で予約済みです。 • このスロットはロック可能です。
	スロット:	• このスロットからは平文読み出しが可能です。 • ロックされていない場合、このスロットに対して平文書き込みが可能です。

表3-12. スロット9: AES鍵(A)

スロット	設定値	有効機能
9	鍵:	• このスロットは、最大で4個のAES 128ビット対称鍵を格納できます。
	スロット:	• スロットは秘密で、鍵は読み出しできません。 • このスロットに対して平文の書き込みが可能です。

表3-13. スロット10: デバイス圧縮証明書(P)

スロット	設定値	有効機能
10	鍵:	• このスロットは、その他のデータの保存用として定義されています • スロットは個別にロックできます。
	スロット:	• データは常に平文で読み出し可能です。 • スロットがロックされていない場合、データを平文で書き込みできます。

表3-14. スロット11: 署名者公開鍵(P)

スロット	設定値	有効機能
11	鍵:	• 署名者証明書に関連するP256 NIST ECC公開鍵がこのスロットに格納されています。 • スロットはロックできます。
	スロット:	• データは常に平文で読み出し可能です。 • スロットがロックされていない場合、データを平文で書き込みできます。

表3-15. スロット12: 署名者圧縮証明書(P)

スロット	設定値	有効機能
12	鍵:	• このスロットは、その他のデータの保存用として定義されています • スロットは個別にロックできます。
	スロット:	• データは常に平文で読み出し可能です。 • スロットがロックされていない場合、データを平文で書き込みできます。

表3-16. スロット13: セキュアブート ダイジェスト(P)

スロット	設定値	有効機能
13	鍵:	• スロットはセキュアブート ダイジェストを保持するよう定義されています。 • スロットは個別にロックできません。
	スロット:	• このスロットからデータは読み出せません。 • このスロットへはデータを直接書き込めません。Secure Bootコマンドを使ってダイジェストを格納できます。

表3-17. スロット14: フィールド プログラミング用の公開鍵 (R)

スロット	設定値	有効機能
14	鍵:	• フィールド プログラミングでの使用に関連するP256 NIST ECCお客様用公開鍵です。 • スロットはプロビジョニング時にロック解除されます。
	スロット:	• データは常に平文で読み出し可能です。 • スロット6内の鍵を使った暗号化書き込みによってこのスロットにデータを書き込みできます。 • 暗号化書き込みにはMACを含める必要があります。

表3-18. スロット15: セキュアブート公開鍵またはCデータ(P)

スロット	設定値	有効機能
15	鍵:	• `Verify`コマンドでセキュアブート動作を検証するためのP256 NIST ECCお客様用公開鍵。 • スロットは個別にロックできます。
	スロット:	• データは常に平文で読み出し可能です。 • スロットがロックされていない場合、データを平文で書き込みできます。

関連リンク

3.2.1. Dataゾーンのデータタイプ

3.3. ECC608-TMNGTLSのEEPROM OTP (One Time Programmable)ゾーン

OTPゾーンはEEPROMアレイ内の64バイト(512ビット)領域であり、読み出し専用ストレージとして使います。このゾーンは、2個のブロック(各32バイト)として構成されます。ECC608-TMNGTLSでは、OTPゾーンはロック済みで出荷され、以下のデータを格納します。

```
02 25 70 13 37 B9 CF F0 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

OTPゾーンに書き込まれたデータバイト値は、4バイトまたは32バイト読み出しを使っていつでも読み出せますが、変更はできません。

NOTICE OTPゾーン内のバイトは今後変更される可能性があります。これらの値を暗号計算に使わない事を推奨します。

4. スタティックRAM (SRAM)メモリ

本デバイスはSRAMアレイも内蔵し、そこに入力コマンドまたは出力結果、ノンス、中間計算値、使い捨て鍵、SHAコンテクスト等を保存できます。SRAMの内容はセキュア エレメントによって内部でのみ使われ、直接読み出す事は決してできません。このメモリの内容は、デバイスがスリープモードに移行するか電源が遮断されると無効になります。

SRAMアレイは以下のバッファにより構成されます。

TempKey

TempKeyはSRAMアレイ内のプライマリ ストレージ レジスタであり、各種コマンドによって生成される中間値を保存するために使えます。TempKeyは64バイトの長さを持ち、32バイトずつの上位部と下位部に分割されます。このレジスタの内容をデバイスから読み出す事はできません(デバイス自体は内部でこの内容を読み出して使えます)。

メッセージ ダイジェスト バッファ

メッセージ ダイジェスト バッファは64バイトのレジスタであり、TempKeyレジスタが異なる情報を保持する必要がある場合に入力メッセージ ダイジェストを`Verify`および`Sign`コマンドに渡すために使います。`SHA`コマンドを使うとダイジェストをこのレジスタに直接書き込む事ができるため、外部ホストのプログラミングを簡素化できます。

代替鍵バッファ

代替鍵バッファは32バイトのレジスタです。TempKeyレジスタで異なる情報を保持する必要がある場合、`KDF`コマンドを使ってこのレジスタに鍵を保存できます。`Nonce`コマンドを使うとこのレジスタに固定入力値を書き込む事ができ、`KDF`コマンドを使うとこのレジスタに秘密値を書き込む事ができます。

SHAコンテクスト バッファ

SHAコンテクスト バッファにより、ダイジェストの生成中に割り込んで他の機能を実行するか他のダイジェストを生成する事ができます。`SHA`コマンドは標準的な3フェイズのフローを使用します。初期化(Initialize)、更新(Update)、最終化(Finalize)の3つです。多くの場合、Updateフェイズは何度も実行されます。フェイズとフェイズの間の中間状態(SHAコンテクストとも呼ぶ)を保存するために内部SRAMメモリを使います。

5. 一般コマンド情報

ECC608-TMNGTLSにおけるI/Oトランザクション、コマンド構造、エラーコード、メモリアドレス指定、鍵と署名のフォーマットに関する一般情報を以下に記載します。

5.1. I/Oトランザクション

ECC608-TMNGTLSはI²Cプロトコルを使ってホスト マイクロコントローラと通信します。セキュリティ コマンドが本デバイスへ送信されると、そのトランザクション内で本デバイスから以下の構造を持つレスポンスが返されます。

表5-1. I/Oトランザクションのフォーマット

バイト	名称	意味
0	Count	そのグループ内でデバイスが送受信するバイトの数です(Countバイト、パケットバイト、チェックサム バイトを含む)。従って、Countバイトの値は常にN+1です(N = パケット内のバイトと2個のチェックサム バイトを含むバイト数)。1つのCountバイトを持つグループの場合、パケットバイトの数が50であれば、2個のチェックサム バイトを含めてN = 52であるため、Countバイトの値は53に設定されます。グループの最大サイズ(Countの値)は155バイト、最小サイズは4バイトです。このレンジから外れた場合、デバイスはI/Oエラーを返します。
1～(N-2)	Packet	コマンド、パラメータ、データ、レスポンスのいずれかです。一般的なコマンドパケット情報については5.2. コマンドパケットを参照してください。各コマンドに固有のパラメータについては6. デバイスコマンドを参照してください。
N-1、N	Checksum	カウントおよびパケットバイトのCRC-16値です(CRC多項式は0x8005)。CRC計算の開始前に、CRCレジスタは0に初期化されます。カウントおよびパケットバイトの最終ビットが送信された後に、内部CRCレジスタ内の値はそのブロック内のチェックサム バイトの値と一致する必要があります。最初に送信されるCRCバイト (グループ内のN-1番目のバイト)はCRC値のLSBであり、グループ内の最後のバイトはCRC値のMSBです。

ECC608-TMNGTLSは、入力グループ内のCount値がコマンド パラメータで指定されたサイズ要件を満たしている事を要求します。Count値がパケット内のコマンド オペコードまたはパラメータに対して矛盾している場合、ECC608-TMNGTLSはコマンドに応じて異なる方法で応答します。応答にエラー通知が含まれる場合もあれば、一部の入力バイトが黙って無視される場合もあります。

5.2. コマンドパケット

コマンドパケットは表5-2の通りに構成されます。

表5-2. コマンドパケット

バイト	名称	意味
0	Opcode	コマンドコードです(6. デバイスコマンドを参照)
1	Param1	第1パラメータ(必須)
2～3	Param2	第2パラメータ(必須)
0～155	データ	残りの入力データ(任意)

ECC608-TMNGTLSはグループ内の全てのバイトを受信した後にビジー状態に移行し、コマンドの実行を開始します。デバイスがビジー中の場合、ステータスも結果もデバイスから読み出す事はできません。ビジー中の場合、デバイスのI/OインターフェイスはI²C上のSDA入力信号の全ての遷移を無視します。

5.3. ステータス/エラーコード

本デバイスは専用のステータス レジスタを備えていません。ステータス、エラー、コマンド結果は全て出力FIFOに格納されます。デバイスからの全ての出力は、入力グループと同じフォーマット(以下を含む)を持つ完全なグループとしてシステムへ返されます。

- Count
- Packet
- 2バイトのCRC

デバイスが入力コマンドグループの最初のバイトを受信した後は、システムから残りの全てのバイトがデバイスへ送信されるまで、システムはデバイスから何も読み出せません。

デバイスが復帰してコマンドを実行した後は、デバイスの出力レジスタにエラー、ステータス、結果バイトのいずれかが格納され、システムからその内容を読み出す事ができます。グループの長さが4バイトである場合に返されるコードの詳細を表5-3に示します。一部のコマンドでは、実行に成功した場合に4バイトより長いコードが返されます。デバイスから返されるパケットの説明は、6.「デバイスコマンド」を参照してください。

CRCエラーは常に他のどのタイプのエラーよりも前に返されます。それらは、I/Oエラーが発生した事およびコマンドをデバイスへ再送信可能である事を示します。複数のエラーが発生した場合、CRCエラー以外のエラーの間に特別な優先順位はありません。

表5-3. 4バイトグループ内のステータス/エラーコード

状態	エラー/ステータス	概要
Successful Command Execution	0x00	コマンドは正常に実行された。
CheckMac or Verify Miscompare	0x01	CheckMacまたはVerifyコマンドは正しくデバイスへ送信されたが、入力レスポンスが期待した値に一致しなかった。
Parse Error	0x03	コマンドは正しく受信されたが、ECC608-TMNGTLSの状態(揮発性メモリおよび/またはEEPROM内の設定)に関係なく長さ、コマンド オペコード、パラメータのいずれかが無効であった。コマンドが再試行される前に、コマンドビットの値を変更する必要があります。
ECC Fault	0x05	ECC処理中に計算エラーが発生したため、結果は無効であった。コマンドを再実行すると成功する場合があります。
Self Test Error	0x07	セルフテスト エラーが発生したため、デバイスはフォルトモード中(フォルトがクリアされるまで待機中)。
Health Test Error	0x08	乱数生成器のヘルステスト エラーが発生したため、エラーがクリアされるまでデバイスは乱数を必要とする後続のコマンドを実行できない。
Execution Error	0x0F	コマンドは正しく受信されたが、デバイスの現在の状態では実行できなかった。コマンドが再試行される前に、デバイスの状態またはコマンドビットの値を変更する必要があります。
After Wake、Prior to First Command	0x11	ECC608-TMNGTLSは正しいWakeトークンを受信した。
Watchdog About to Expire	0xEE	WDTがタイムアウトする前に指定されたコマンドを完了できない。システムは、アイドルまたはスリープモードへの移行によってWDTをリセットする必要があります。
CRC or other Communications Error	0xFF	コマンドはECC608-TMNGTLSによって正しく受信されなかったため、システム内のI/Oドライバによってコマンドを再送信する必要がある。コマンドの構文解析または実行は試みられませんでした。

5.4. アドレスの指定

以下では、ECC608-TMNGTLSの各種メモリゾーンをアドレス指定するための詳細な方法を説明します。

5.4.1. Configurationゾーンのアドレス指定

Configurationゾーンには、4バイトまたは32バイト単位でアクセスできます。1バイト単位でのアクセスはできません。Configurationゾーンのアドレスは2バイト(16ビット)です。アドレスワードの最下位5ビットのみがConfigurationゾーンのアドレス指定用に使われます。ECC608-TMNGTLSの場合、これらのアドレスは`Read`コマンドでのみ使えます。

表5-4. アドレスのフォーマット

バイト1: Addr[15:8]	バイト0: Addr[7:0]		
未使用	未使用	ブロック	オフセット
Addr[15:8]	Addr[7:5]	Addr[4:3]	Addr[2:0]

表5-5. Configurationゾーンのアドレス

ブロックNo. (Addr[4:3])	オフセット値(Addr[2:0])							
	000	001	010	011	100	101	110	111
00	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
01	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]
10	[64:67]	[68:71]	[72:75]	[76:79]	[80:83]	[84:87]	[88:91]	[92:95]
11	[96:99]	[100:103]	[104:107]	[108:111]	[112:115]	[116:119]	[120:123]	[124:127]

5.4.2. OTPゾーンのアドレス指定

OTPゾーンには、4バイトまたは32バイト単位でアクセスできます。このゾーンは全部で64バイトです。1バイト単位でのアクセスはできません。OTPゾーンのアドレスは2バイト(16ビット値)です。最下位4ビットのみがアドレス指定用に使われます。

ECC608-TMNGTLSの場合、これらのアドレスは`Read`コマンドでのみ使えます。

表5-6. アドレスのフォーマット

バイト1: Addr[15:8]	バイト0: Addr[7:0]		
未使用	未使用	ブロック	オフセット
Addr[15:8]	Addr[7:4]	Addr[3]	Addr[2:0]

表5-7. OTPゾーンのバイトアドレス

ブロックNo. (Addr[3])	ブロック オフセット値(Addr[2:0])							
	000	001	010	011	100	101	110	111
0	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
1	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]

5.4.3. Dataゾーンのアドレス指定

Dataゾーンへの読み書きアクセスは、ConfigurationおよびOTPゾーンに比べると非常に複雑です。16個のスロットが存在し、サイズはスロットごとに異なります。各スロットのアクセスポリシー(そのスロットに対する読み出し/書き込みアクセスを許可するかどうか)は、スロットごとに制御します。

ECC608-TMNGTLSの場合:

- データスロット2、4、7～12、15は平文として書き込み可能に設定されます。
- データスロット5、14は暗号文で書き込み可能です。
- データスロット2、8、10～12、14～15は平文として読み出し可能です。
- 未指定のスロットは、読み出しも書き込みもできません。

表5-8. データスロットのアドレス フォーマット

Dataゾーン	バイト1: Addr[15:8]		バイト0: Addr[7:0]		
	未使用	ブロック	未使用	スロット	オフセット
データスロット[7:0]	Addr[15:9]	Addr[8]	Addr[7]	Addr[6:3]	Addr[2:0]
データスロット[8]	Addr[15:12]	Addr[11:8]	Addr[7]	Addr[6:3]	Addr[2:0]
データスロット[15:9]	Addr[15:10]	Addr[9:8]	Addr[7]	Addr[6:3]	Addr[2:0]

データスロット[7:0]

これらのスロットの1つに完全にアクセスするには、2回の32バイトアクセスまたは9回の4バイトアクセスが必要です。

表5-9. Dataゾーンアドレス - スロット0～7

スロットNo. (Addr[6:3])	ブロックNo. (Addr[8])	ブロック オフセット値(Addr[2:0])							
		000	001	010	011	100	101	110	111
0x0～0x7	00	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
	01	[32:35]	無効	無効	無効	無効	無効	無効	無効

データスロット[8]

このスロットに完全にアクセスするには、13回の32バイトアクセスまたは104回の4バイトアクセス(またはそれらのアクセスの組み合わせ)が必要です。

表5-10. Dataゾーンアドレス - スロット8

スロット No. (Addr[6:3])	ブロック No. (Addr[8])	ブロック オフセット値(Addr[2:0])							
		000	001	010	011	100	101	110	111
0x8	0x0	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
	0x1	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]
	...	...	...	...	...	...	...	...	...
	0xC	[384:387]	[388:391]	[392:395]	[396:399]	[400:403]	[404:407]	[408:411]	[412:415]

データスロット[15:9]

これらのスロットに完全にアクセスするには、3回の32バイトアクセスまたは18回の4バイトアクセス(またはそれらのアクセスの組み合わせ)が必要です。

表5-11. Dataゾーンアドレス - スロット9～15

スロットNo. (Addr[6:3])	ブロック No. (Addr[8])	ブロック オフセット値(Addr[2:0])							
		000	001	010	011	100	101	110	111
0x9～0xF	00	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
	01	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]
	10	[64:67]	[68:71]	無効	無効	無効	無効	無効	無効

5.5. 鍵、署名、証明書のフォーマット

以下では、ECC鍵、署名、圧縮した証明書のフォーマットについて詳細に説明します。

5.5.1. ECC鍵のフォーマット

公開鍵と秘密鍵のフォーマットは、コマンドと鍵の長さによって決まります。一般的に、最上位バイト(MSB)はバス上に最初に現れ、メモリ内の最低アドレスに格納されます。以下では、ページの左側に記載するバイトがMSBです。一貫性を保つため、Microchip社は全てのパディングバイトを0に設定する事を推奨します。

- ECC秘密鍵は、`PrivWrite`コマンドに対する入力パラメータとしてのみユーザーに提供されます。このパラメータの長さは常に36バイトであり、最初の4バイト(32ビット)は全てパディングビットです。ECC公開鍵は各種コマンドに対する入力または出力パラメータとして提供され、EEPROMに保存する事もできます。これらの鍵はX値とY値で構成され、X値が最初にバスまたはメモリ上に現れます。これらのフォーマットは状況に応じて異なります。
- **公開鍵が`GenKey`コマンドの出力または`Verify`コマンドへの入力である場合**: 32バイトのXの後に32バイトのY。(36バイト) パディングバイトなし
- **`Write`コマンド**:

  公開鍵は`Write`コマンドを使ってEEPROMに直接書き込む事ができ、常に72バイト長でフォーマットは以下の通りです。パディングバイト(4バイト)、X (32バイト)、パディングバイト(4バイト)、Y (32バイト)
- **`GenKey`コマンド**:

  SHAメッセージ: 公開鍵は、`GenKey`コマンドによりハッシュ化してTempKeyに格納できます。SHAメッセージは、鍵のサイズとは無関係な各種のバイトを格納します。これらのバイトの後にパディングバイト(25バイト)、X (32バイト)、Y (32バイト)が続きます。
- **`Verify`コマンド**:

  SHAメッセージ: 保存された公開鍵の検証に`Verify`コマンドを使う場合、メモリに保存されている鍵のSHA-256ダイジェストに対して生成された署名の入力が必要です。そのような内部のSHA計算は、常に72バイト フォーマットに対して実行されます(公開鍵はEEPROM内でパディングバイト(4バイト)、X (32バイト)、パディングバイト(4バイト)、Y (32バイト)として保存されるため)。

公開鍵が`Verify`コマンドによって検証されるよう設定されている場合、デバイスは検証ステートを保存するためにメモリ内の最初のバイトの最上位4ビットを内部で使います。検証ステートは、`Write`コマンドによって常に無効ステート(`0xA`)に設定され、その後に`Verify`コマンドによって有効ステート(`0x5`)に設定する事ができます。

I/Oプロトコルの最下層について以下で説明します。I/Oプロトコルより上層では、コマンドを実装するために、完全に同じバイトがデバイスに対して双方向に転送されます。以下では、エラーコードについて説明します。

5.5.1.1. 公開鍵のフォーマット

ECC608-TMNGTLSは、2通りのフォーマットでP-256楕円曲線公開鍵を使います。以下の例により、2つのフォーマットを詳細に示します。

これらの例では以下のサンプル公開鍵(XおよびYを固定幅のビッグ エンディアン符号なし整数として表現)を使います。

```
X: b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
Y: a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

コマンドに関わる公開鍵のフォーマット

公開鍵(GenKey)を返すコマンドまたは公開鍵をパラメータとして受け付けるコマンド(Verify、ECDH)は、XおよびYビッグ エンディアン符号なし整数を連結した64バイトとして公開鍵をフォーマットします。

例:

```
b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

保存された公開鍵のフォーマット

Verifyコマンドで使うために公開鍵をスロットに保存する場合、整数XおよびYをそれぞれ36バイトとなるようパディングしてからそれらを連結した72バイトとして公開鍵をフォーマットします。

例:

```
00000000b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
00000000a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

Note: スロット8～15のみが公開鍵を保存可能なサイズを有します。

保存された有効な公開鍵のフォーマット

有効または無効な公開鍵のフォーマットは、LSBの最上位4ビットを除けば、保存された公開鍵のフォーマットと同じです。鍵が有効である場合の最下位ニブルは0x5であり、無効である場合の最下位ニブルは0xAです。これらの値は、有効または無効モードでVerifyコマンドを実行する事によって変更できます。書き込み後の鍵の初期状態は無効です。

有効な公開鍵の例:

```
50000000b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
00000000a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

無効な公開鍵の例:

```
A0000000b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
00000000a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

Note: スロット8～15のみが公開鍵を保存可能なサイズを有します。

5.5.2. 署名のフォーマット

Signコマンドによって生成および出力されるECDSA署名およびVerifyコマンドへ入力されるECDSA署名は常に64バイトです。署名はR部分とS部分に分割されます。どちらも32バイトの長さを持ち、常にRがSの前にバス上に現れます。署名の各部分はMSB先頭でバス上に現れます(すなわち、署名のMSBは最低メモリ位置に格納されます)。

R/S署名の例

署名を返すコマンド(Sign)または署名をパラメータとして受け付けるコマンド(Verify)は、署名をRおよびSビッグ エンディアン符号なし整数を連結した64バイトとしてフォーマットします。

例:

```
R: 7337887F8C39DF79FD8BF88DDFBFB9DB15D7B1AD68196AE3FB0CE5BFA2842DF3
S: 72868A43A42831E950E1DA9F73B29F5C0ED8A96B2889E3CBBE8E61EA6C67F673
```

5.5.3. 証明書の保存

完全なX.509証明書をデバイス内に保存すると、複数のEEPROMメモリスロットを急速に使い果たしてしまう可能性があります。
証明書の保存用にこれらのスロットを使った方が良い場合もあれば、そうではない場合もあります。これらのメモリ制限のため、Microchip社は最小限の情報から完全なX.509証明書を再構成可能なエンコードを定義しています。

実際のX.509証明書の再構成はホストシステムによって行われますが、その方法はエンコード済み証明書内に保存されているデータによって決まります。システム内の全デバイスに共通するデータは、ホストシステム内に保存できます。その他のデータは、デバイス内に保存されたデータから計算または抽出できます。表5-12に、X.509証明書内に保存されるデータのタイプと、1つの72バイトスロットに証明書を収める事ができるエンコード方法を示します。

表5-12. 証明書の保存

X.509証明書		エンコード済み証明書		
X.509エレメント	サイズ(バイト)	エンコード済み証明書のエレメント	デバイス証明書(ビット)	署名者証明書(ビット)
シリアル番号	8-20	シリアル番号のソース	4	4
発行日付	13	圧縮フォーマット	19	19
有効期限	13	有効期限の年数	5	5
署名者ID(2)	4	証明書の署名に使う特定署名者のID(デバイス証明書の場合)、または署名者自身のID(署名者証明書の場合)	16	16
AuthorityKeyIdentifier	20	Authority公開鍵のSHA1 HASH	0	0
SubjectKeyIdentifier	20	Subject公開鍵のSHA1 HASH	0	0
署名のR	32	デバイス内に保存	256	256
署名のS	32	デバイス内に保存	256	256
公開鍵のX(1)	32	秘密鍵から計算、または、デバイス内に保存(1)	0	256
公開鍵のY(1)	32	秘密鍵から計算、または、デバイス内に保存(1)	0	256
n/a	0	証明書フォーマット	4	4
n/a	0	テンプレートID	4	4
n/a	0	チェーンID	4	4
n/a	0	予約済み/ユーザー定義	8	8
合計	(206～218バイト)	–	576ビット(72バイト)	1088ビット(136バイト)

Note:

1. デバイス証明書の場合、デバイス公開鍵は秘密鍵から再生できます。署名者証明書の場合、通常、公開鍵は別のスロットに保存されます。
2. デバイス証明書の場合、証明書の署名に使う署名者のIDが保存されます。署名者証明書の場合、署名者の実際のIDは、デバイスがそれを識別できるように保存されます。

スロット8は全部で416バイトを格納します。証明書に保存されているシリアル番号のサイズによっては、2つの完全な証明書を保存できない場合があります。多くの場合、信頼の輪(Chain of Trust)が構築済みのデバイスではデバイス証明書、署名者証明書、署名者公開鍵をデバイス内に保存する必要があります。

詳細は、アプリケーション ノート『Compressed Certificate Definition』を参照してください。

6. デバイスコマンド

以下では、ECC608-TMNGTLSで使用可能な全てのコマンドと、各コマンドのコマンドモードについて詳細に説明します。これらのコマンドは以下の3つのカテゴリに分類されます。

1. **一般デバイスコマンド**

   これらのコマンドは、さらに2つのカテゴリに分類されます。

   - 一般デバイス アクセスコマンド: デバイスとのデータ送受信用に使います。通常、これらのコマンドは暗号機能を実行しません。
   - 一般暗号コマンド: デバイスまたはシステムはこれらのコマンドを使えます。通常、これらのコマンドは特定データスロットを対象として実行されません。

2. **非対称暗号コマンド**

   これらのコマンドは、ECC公開鍵または秘密鍵を使う非対称暗号演算(鍵の生成、メッセージの署名、メッセージの検証)を実行します。これらのコマンドは、ECC Dataゾーンスロットに対してのみ使えます。

3. **対称暗号コマンド**

   これらのコマンドは対称暗号関数(ダイジェストまたはMACの生成、鍵導出、AES暗号化/復号等)を実行します。

全てのコマンドの入力パラメータ

特に明記しない場合、複数バイトから成る入力パラメータは、入力パラメータ表にビッグ エンディアン(MSB先頭)として記載されます。ECC608-TMNGTLSは、データがリトル エンディアン(LSB先頭)で送信される事を期待する事に注意が必要です。

表6-1. コマンド、概要、コマンドカテゴリ

コマンド	概要	コマンドカテゴリ
AES	AES-ECB暗号化/復号機能を実行します。ガロア体乗算(GFM)を計算します。	対称暗号コマンド
CheckMac	他のCryptoAuthentication™デバイスで計算されたMACを検証します。	対称暗号コマンド
Counter	モノトニック カウンタの1つを読み出すかインクリメントします。	一般デバイスコマンド
ECDH	保存されている秘密鍵と入力された公開鍵を使ってECDHプリマスタ秘密鍵を生成します。	非対称暗号コマンド
GenDig	乱数または入力シードと鍵からデータ ダイジェストを生成します。	対称暗号コマンド
GenKey	ECC公開鍵を生成します。オプションでECC秘密鍵を生成する事もできます。	非対称暗号コマンド
Info	デバイスのステート情報を返します。	一般デバイスコマンド
KDF	PRFまたはHKDF鍵導出関数を実装します。	対称暗号コマンド
Lock	デバイスのゾーンまたはスロットに対する後続の変更を禁止します。	一般デバイスコマンド
MAC	SHA-256を使って鍵とその他の内部データからダイジェスト (レスポンス)を計算します。	対称暗号コマンド
Nonce	32バイトの乱数と内部保存されるノンスを生成します。	一般デバイスコマンド
Random	乱数を生成します。	一般デバイスコマンド

...続き		
コマンド	**概要**	**コマンドカテゴリ**
Read	デバイスから4また32バイトを読み出します(平文または暗号文)。	一般デバイスコマンド
SecureBoot	電源投入時にコード署名またはコード ダイジェストを検証します。	非対称暗号コマンド
SelfTest	内部の各種暗号計算エレメントをテストするために使います。	一般デバイスコマンド
Sign	ECDSA署名計算	非対称暗号コマンド
SHA	システムによって汎用的に使われるSHA-256またはHMACダイジェストを計算します。	一般デバイスコマンド
UpdateExtra	Configurationゾーンがロックされた後に、Configurationゾーン内のバイト84または85を更新します。	一般デバイスコマンド
Verify	ECDSA検証計算	非対称暗号コマンド
Write	デバイスに4また32バイトを書き込みます(平文または暗号文)。	一般デバイスコマンド

関連リンク

6.1. 一般デバイスコマンド
6.2. 非対称暗号コマンド
6.3. 対称暗号コマンド

6.1. 一般デバイスコマンド

表5-1に、一般デバイスコマンドの一覧を示します。

表6-2. 一般デバイスコマンド

コマンド名	概要
Counter	モノトニック カウンタのインクリメントと読み出しを行います。
Info	デバイスからリビジョンおよびステータス情報を読み出すために使います。
Lock	デバイス内のロック可能スロットを個々にロックするために使います。
Nonce	ノンス (ワンタイムトークン)を生成するかデバイスに渡すために使います。
Random	システムによって使われる32バイト乱数を生成するために使います。
Read	デバイスの各種ゾーンを読み出すために使います。
SelfTest	内部の各種暗号計算エレメントをテストするために使います。
SHA	システムによって汎用的に使われるSHA-256またはHMACダイジェストを計算します。
UpdateExtra	Configurationゾーンがロックされた後に、Configurationゾーン内のバイト84または85を更新するために使います。
Write	デバイスに4また32バイトを書き込むために使います(平文または暗号文)。

6.1.1. Counterコマンド

Counterコマンドは、デバイスのConfigurationゾーン内に配置された2つのモノトニック カウンタの1つから2進数カウント値を読み出します。カウンタのカウント可能最大値は2,097,151です。この値を超えてカウントしようとすると、エラーコードが生成されます。これらのカウンタは、カウント動作中に給電が中断してもカウント値が失われないように設計されています。電源喪失条件によっては、カウント値が2つ以上インクリメントする場合があります。

ECC608-TMNGTLSの場合、カウンタはどの鍵にも割り当てられていませんが、システムによってカウンタを使う事ができます。各カウント値は既定値に設定され、最大値までカウント可能です。

6.1.2. Infoコマンド

Infoコマンドは、デバイスのステータスと状態を読み出すために使います。この情報は、エラーの特定と各種コマンドの実行に役立ちます。

6.1.3. Lockコマンド

ECC608-TMNGTLSの場合、Configurationゾーンはロック済みであり、Dataゾーンのアクセスポリシーは設定済みです。しかし、一部のスロットは他のコマンドを使って更新可能です。必要に応じ、その中の一部のスロットはLockコマンドのSlotLockモードを使って恒久的にロックできます。ロックすると、そのスロットは永久に変更できなくなります。

6.1.4. Nonceコマンド

Nonceコマンドは、乱数(内部または外部で生成可能)とシステムからの入力値を組み合わせる事により、後続のコマンド向けにノンス(Nonce: Number used Once)を生成します。生成されたノンスは、内部で以下の3つのバッファのいずれかに保存されます。TempKeyバッファ、メッセージ ダイジェスト バッファ、代替鍵バッファノンスを生成する代わりに固定値をデバイスに渡す事もできます。

6.1.5. Randomコマンド

Randomコマンドは、システムによって使われる乱数を生成します。乱数は内部のNIST 800-90 A/B/C乱数生成器により生成されます。このコマンドは、常にバスへ32バイト値を出力します。この値をデータスロットまたはSRAM内に保存する事はできません。

6.1.6. Readコマンド

Readコマンドを使うと、ECC608-TMNGTLSの全てのEEPROMゾーンにアクセスできます。Dataゾーンへのアクセスは、スロットごとに設定されたアクセスポリシーにより制限されます。暗号読み出しは、特定のアクセスポリシーが設定されたDataゾーンスロットに対してのみ可能です。

6.1.7. SelfTestコマンド

SelfTestコマンドは、ECC608-TMNGTLS内の1つまたは複数の暗号エンジンのテストを実行します。入力モードパラメータに応じて、全てまたは一部のアルゴリズムがテストされます。

ECC608-TMNGTLSの場合、電源投入または復帰イベント後の自動的なSelfTestコマンド実行は無効にされています。しかし、システムは必要に応じてこのコマンドを実行できます。このテストの実行に関する要件はありません。

電源投入または復帰時に自動的に実行された場合でも、このコマンドによって実行された場合でも、セルフテストに失敗するとデバイスはFailureステートに移行し、デバイスの動作は制限されます。保存されたFailureステートは、復帰時または電源再投入時にクリアされます。

6.1.8. SHAコマンド

SHAコマンドは、システムによって汎用的に使われるSHA-256またはHMAC/SHAダイジェストを計算します。SHA計算は、ECC608-TMNGTLSの内部メモリの特別なセクション (コンテクスト バッファ)内で実行されます。他のコマンドを使ってこのセクションを読み書きする事はできません。SHAコマンドの各種フェイズとフェイズの間に任意のコマンドを挿入できます。その際のSHAコンテクストは、電源投入および復帰時に無効になります。SHAコマンドの実行中にエラーが発生しても、多くの場合はコンテクストが変更される事なく保持されます。

6.1.9. UpdateExtraコマンド

UpdateExtraコマンドは、UpdateExtraバイトとUpdateExtraAddバイト(Configurationゾーン内のバイト84とバイト85)を更新するために使います。これらのバイトは、このコマンドによってのみ更新できます。これらのバイトは、現在の値が0x00である場合にのみ一度だけ更新が可能です。現在の値が0x00ではない場合、更新を試みるとエラーが発生します。

ECC608-TMNGTLSの場合、UpdateExtraAddバイト (バイト85)は代替I^2Cアドレスに設定済みです。

6.1.10. Writeコマンド

ECC608-TMNGTLSの場合、ConfigurationゾーンとOTPゾーンはロック済みであり、これらのゾーンの更新はできません。Dataゾーンに対する書き込みは、各スロットのアクセスポリシーに基づいて制限されます。

6.2. 非対称暗号コマンド

非対称暗号コマンドセットは、ECC鍵を生成または使用するための特別なコマンドで構成されます。鍵は通常Dataゾーンに保存されますが、一部のコマンドではSRAMアレイに保存されます。

表6-3. 非対称暗号コマンド

コマンド名	概要
ECDH	保存されている秘密鍵と入力された公開鍵を使ってECDHプリマスタ秘密鍵を生成します。
GenKey	保存されている秘密鍵からECC秘密鍵またはオプションによりECC公開鍵を生成します。
SecureBoot	電源投入時にコード署名またはコード ダイジェストを検証します。
Sign	ECC秘密鍵を使ってECDSA署名計算により内部または外部のメッセージ ダイジェストに署名します。
Verify	ECC秘密鍵を使ってECDSA検証計算により内部または外部のメッセージ ダイジェストを検証します。

6.2.1. ECDHコマンド

ECDHコマンドは、2つのデバイスの間で共有する秘密鍵を生成します。2つのデバイスは、それぞれ他方のデバイスからECC公開鍵を受け取り、スロットに保存されているECC秘密鍵またはTempKeyに保存されている使い捨て鍵と組み合わせます。これにより、両方のデバイスで同じ共有プリマスタ秘密鍵を生成します。さらに、この鍵を双方で共有する他のデータと組み合わせる事により、共有セッション鍵を生成する事ができます。共有秘密をさらにDiversifyするため、KDFコマンドがTLSセッションでしばしば使われます。

6.2.2. GenKeyコマンド

GenKeyコマンドにより、ECC秘密鍵の生成、秘密鍵からのECC公開鍵の生成、公開鍵ダイジェストの生成が可能です。このコマンドは、ECC秘密鍵または公開鍵向けに設定されたスロットに対してのみ使えます。非ECCスロットに対してこのコマンドを実行するとエラーが発生します。

6.2.3. SecureBootコマンド

SecureBootコマンドは、外部MCUまたはMPUのセキュアブート向けのサポートを提供します。一般的に、システム内のブートコードは、ブート後に実行されるアプリケーション コードの検証を支援するためにECC608-TMNGTLSを使います。ECC608-TMNGTLSは、Stored DigestモードのSecureBootコマンドを使って動作するよう設定済みです。ダイジェストはスロット13に保存され、SecureBootの検証に必要な公開鍵はスロット15に保存されます。

ホストとECC608-TMNGTLSの間の配線の改ざんを検知し保護するため、コマンドのモードに応じて、戻りコードの代わりにMACを各種データ(TempKeyに書き込まれたノンス、I/O保護秘密鍵等)から生成できます。

6.2.4. Signコマンド

Signコマンドは、ECDSAアルゴリズムを使って署名を生成します。これには、KeyIDによって指定されたスロット内のECC秘密鍵が使われます。何に署名するかに応じて異なるモードが利用できます。

6.2.5. Verifyコマンド

Verifyコマンドは、入力されたメッセージ ダイジェストと公開鍵に基づき、ECDSA [R,S]署名が正しく生成されたかどうかを検証します。いかなる場合も、署名がこのコマンドへの入力です。公開鍵はデバイスに保存する事も入力として提供する事もできます。

中間者攻撃を防ぐため、Verifyコマンドからオプションの MACを返す事ができます。署名が入力ダイジェストから正しく生成された事が検証計算により示された場合、TempKeyに保存されている入力ノンスと、ECC608-TMNGTLSとホストMCUの両方に保存されているI/O保護秘密鍵に基づいて、MACが計算されます。MAC出力は、ExternalおよびStoredモードでのみ生成可能です。MACを計算するには、I/O保護機能を有効にする必要があります。

スロット4および9に格納されたAES鍵は使用前に検証する必要があります。

6.3. 対称暗号コマンド

対称暗号コマンドセットは、対称鍵の生成または使用に関係するコマンドの集まりです。鍵は通常Dataゾーンに保存されますが、一部のコマンドではSRAMメモリアドレスに保存されます。

表6-4. 対称暗号コマンド

コマンド名	概要
AES	AES-ECB暗号化/復号機能を実行します。ガロア体乗算(GFM)を計算します。
CheckMac	他のCryptoAuthentication™デバイスで計算されたMACを検証します。
GenDig	乱数または入力シードと鍵からデータ ダイジェストを生成します。
KDF	PRFまたはHKDF鍵導出関数を実装します。
MAC	SHA-256を使って鍵とその他の内部データからダイジェスト (レスポンス)を計算します。

6.3.1. AESコマンド

AESコマンドは、AES鍵を使ってデータの16バイトブロックを暗号化または復号するために使えます。鍵は、指定されたスロット内の16バイト(128ビット)位置またはTempKeyの最初の16バイトに保存されます。指定されたスロットには複数の鍵を保存してアクセスできます。これらの鍵は、連続する16バイト境界(バイト0-15からスロットのサイズまで)に格納できますが、1つのスロットに5個以上の鍵を保存する事はできません。ECC608-TMNGTLSの場合、AES鍵はスロット4、5、または9に保存できます。スロット4および5は最大で2個のAES鍵を格納でき、スロット9は最大で4個のAES鍵を格納できます。

AES暗号化/復号の他に、AESコマンドを使ってGFM (Galois Field Multiply)を生成する事もできます。これは、ECC608-TMNGTLSが直接サポートしない暗号演算をサポートするために使います。

6.3.2. CheckMacコマンド

CheckMacコマンドは、本デバイスとは異なるCryptoAuthenticationデバイス[1]上で生成されたMACレスポンスを計算し、その結果を入力値と比較します。このコマンドは、比較の結果を示す真偽値を返します。

[1] 互換性のあるMACレスポンスを生成するデバイスには、ATECC608A/B/C、ATECC508A、ATSHA204A、SHA104があります。これには、該当する製品に関連するTrust製品のバリエーションが含まれます。

TempKey内の値をCheckMacへの入力として使う場合、CheckMacコマンドの前にNonceおよび/またはGenDigコマンドを実行する必要があります。

6.3.3. GenDigコマンド

GenDigコマンドは、SHA-256ハッシュを使って、保存されている値または入力値とTempKeyの内容を組み合わせます。このコマンドを実行する前に、TempKeyの内容を検証する必要があります。保存値はデータスロットの1つ、Configurationゾーン、いずれかのOTPページ、モノトニック カウンタから取り込めます。デバイスのモードに応じて、GenDig計算にどのデータを含めるかが決まります。

場合によっては、何らかのコマンドを実行する前にGenDigを実行する必要があります。与えられたコマンドを実行する前に、GenDigコマンドを複数回実行する事により、ダイジェストに追加のデータを含める事ができます。その結果得られたダイジェストはTempKeyで保持され、以下の4通りの方法で使えます。

1. MAC、Sign、CheckMacコマンドによって使われるメッセージの一部としてダイジェストを含める事ができます。MACレスポンス出力はGenDig計算で使われたデータとMACコマンドからの秘密鍵の両方を含むため、ダイジェストはDataおよび/またはOTPゾーンに保存されているデータの認証用に使えます。
2. 後続のReadまたはWriteコマンドは、ダイジェストを使ってデータに認証および/または機密性を提供できます。この場合、ダイジェストはデータ保護ダイジェストと呼びます。
3. このコマンドは、トランスポート鍵配列からの値を使う事により、セキュア パーソナライズ用に使えます。結果として得られたデータ保護ダイジェストは、Writeコマンドによって使われます。
4. 入力値(通常はリモートデバイスからのノンス)と現在のTempKey値が組み合わされて共有ノンスが生成され、その中で両方のデバイスはRNGが含まれている事を証明できます。

6.3.4. KDFコマンド

ECC608-TMNGTLSの場合、KDFコマンドは各種の鍵導出関数(TLS 1.2向けPRF、TLS 1.3向けHKDFおよびAESを含む)を実装します。この関数はCryptoAuthLib内で利用できます。
詳細は7.2.3.「CryptoAuthLib」を参照してください。

KDFコマンドの詳細については、Microchip社正規代理店までお問い合わせください。

6.3.5. MACコマンド

MAC (Message Authentication Code)コマンドは、メッセージのSHA-256ダイジェストを生成するために使われます。このダイジェストは、デバイス内に保存された鍵、チャレンジ、デバイスに関するその他の情報を含みます。このコマンドの出力は、このメッセージのダイジェストです。

このコマンドを使うための通常のフローは以下の通りです。

1. Nonceコマンドを実行して入力チャレンジをロードします。オプションにより、このチャレンジと生成された乱数を組み合わせる事ができます。この演算の結果は、ノンスとしてデバイス内部に保存されます。
2. 必要に応じ、GenDigコマンドを1回または複数回実行する事で、デバイス内のEEPROM位置に保存されている値をノンスと組み合わせる事ができます。その結果はデバイス内部に保存されます。この機能により、複数の鍵をレスポンス生成の一部として使う事ができます。
3. このMACコマンドを実行して上記ステップ1(および必要に応じてステップ2)の出力とEEPROM鍵を組み合わせる事で、出力レスポンス(すなわちダイジェスト)を生成します。

あるいは、同じGenDigメカニズムを通して、秘密である事が要求されない任意のスロット内のデータをレスポンスに蓄積する事ができます。これは、その位置に保存されている値を認証する効果を有します。

7. 応用のための情報

ECC608-TMNGTLSは、Microchip社のCryptoAuthentication™ Trust Platformファミリに属します。Trust Manager製品はセルフサービスの暗号鍵管理デバイスです。デバイスの実装は簡単で、Microchip社とKudelski IoTの技術とプロビジョニング インフラストラクチャを活用しながら、たとえ小量生産であってもエンドシステムにセキュアな認証機能を実装できます。

ECC608-TMNGTLSは、IoT対応製品にセキュリティ機能を追加する際の煩雑な手間を取り除く事を目的に開発されました。本製品はDTLS接続を通じて任意のクラウド インフラストラクチャに容易に接続できるように事前設定されています。

実際のセキュリティ デバイスをKudelski IoT keySTREAM SaaSサービスと組み合わせると、製品のライフサイクル全体にわたって完全な信頼性ソリューションが実現されます。このソリューションがサポートするユースケースは複数あります。

7.1. ユースケース

ECC608-TMNGTLSは、IoT市場向けに定義されています。Kudelski keySTREAM SaaSを使う事で、幅広いクラウド ネットワークと特定の認証サービスにアクセスできます。システムの構造上の柔軟性により、Microchip社の工場を介在させる事なく、カスタムのPKIをフィールドで確立できます。これにより、証明書の書き換えやローテーション、セキュリティ ポリシーの更新や維持が可能になり、これらのアプリケーションの多様な用途とユースケースに対応できます。本デバイスの主なユースケースについて以下で簡単に説明します。これらのユースケースは、単独で実装する事も、互いに組み合わせて実装する事もできます。これらのユースケースの試作および実装をサポートするため、Microchip社はハードウェアおよびソフトウェア ツールを提供しています。

セキュアTLS接続

ECC608-TMNGTLSでは、TLSプロトコルのセキュリティ認証を作成できます。デバイスにはMicrochip社の認証済みセキュア プロビジョニング施設内でMicrochip社によって生成された出生時秘密鍵が保存されます。この鍵はkeySTREAM SaaSに向けてECC-P256署名を発行し、keySTREAM SaaSはECC608 TrustManagerを認証します。SaaSはカスタムPKIを作成し、新しいカスタム証明書をデバイスにプロビジョニングします。これがカスタムPKIのフィールドでのプロビジョニングです。ルート証明書と中間証明書に関連付けられた秘密鍵は、お客様のアカウント セットアップ時に設定されるKudelski IoT HSM内で保護されます。これにより、幅広いクラウド事業者に接続できるようになります。KDF (Key Derivation Function)の各種モードにより、TLS1.2/TLS1.3セキュア接続インターネット プロトコル(およびそれ以前のバージョン)をサポートする適切な鍵を生成できます。

証明書管理

アカウント セットアップ時keySTREAM SaaSにセットアップされたカスタムPKIを通して、SaaSを使って証明書の有効期限、失効、更新を管理、監視できるようになります。

アカウントの初期セットアップ後、証明書の有効期限を設定してECC608-TMNGTLSに保存できます。証明書の有効期限切れによるサービスの停止を防ぐため、証明書の有効期限に達したらKudelski IoTポリシーで規定された通りに自動で更新されるように設定できます。

証明書の失効を必要とするセキュリティ侵害またはその他のインシデントが発生した場合、KeySTREAM SaaSを使う事で、ECC608-TMNGTLSデバイス証明書を失効させる事ができます。証明書を失効させると、デバイスはクラウドに接続できなくなります。必要な場合、デバイスのセキュア認証を復旧するため、デバイスに保存された現在の証明書を置き換える新しい証明書を発行して証明書を更新できます。失効と更新により、証明書のローテーションが実現できます。

秘密鍵ローテーション

keySTREAM SaaSはECC608-TMNGTLS内の秘密鍵のローテーションを制御できます。最終製品の秘密鍵が信頼できないと判断された場合、keySTREAM SaaSを使って、スロット0に保存された新しい秘密鍵を再生成できます。リスクは、スロット0においてkeySTREAM SaaSがその特定のデバイスについてSaaSへのアクセスを拒否するサービス拒否です。

セキュアブート

マイクロコントローラまたはマイクロプロセッサのファームウェア イメージの保護は、多くのベンダーにとっての懸念事項です。実行中のコードが信頼できる事(改ざんされていない事)を検証するための機能により、システム全体の保全性が維持されます。ECC608-TMNGTLSは、システムのコード ダイジェストをデバイスのデータスロットに保存する事によりセキュアブートを可能にするよう設定済みです。コードの初期実行時に、システムはシステム ファームウェアに対するダイジェストを再構成し、それをECC608-TMNGTLSに保存されているダイジェストと比較する事で、ファームウェアが改ざんされていない事を確認できます。

知的財産およびデータ保護

知的財産(IP)の保護は、OEMの競争力を維持するために不可欠です。IPを保護する事で、OEMが開発したファームウェアまたはハードウェアが複製される事を防ぎます。ファームウェアIP保護は、単なるソフトウェア ベースのアプローチで実現可能ですが、ファームウェア内部の鍵情報は攻撃に対してきわめて脆弱です。

製品を確実に正しいファームウェアで動作させるため、ECC608-TMNGTLSはハードウェア ベースのセキュアな鍵ストレージを提供します。本デバイスは対称認証と非対称認証の両方を実行できます。これらは、鍵をセキュア エレメント内で安全に保存する事により、ハッカーによって鍵が抽出および改変される可能性を低減します。ランタイム動作の任意の時点において、スロット9のAES対称鍵を使うとファームウェア イメージのハッシュを検証でき、スロット15の公開鍵を使うと署名付きイメージの署名を検証できます。

エンドツーエンドのデータ保護

対称鍵を使う事で、セキュアなデータ通信を実装できます。ECDH関数を使って対称鍵を生成して、デバイスとエンドポイント間の通信の暗号化と復号を実現できます。データはクラウドからデバイス、デバイスからクラウドまで完全に保護されます。

一般的データストレージ

データスロット8は複数のkeySTREAM SaaS動作に必要な特定のお客様/アプリケーションのオンボーディング情報を格納するために使われます。このスロットはお客様が一般用途で使う事はできません。システムに小量の追加情報を保存したい場合があります。データスロット15は追加のお客様固有またはアプリケーション固有の情報を格納するために使う事ができます。ECC608-TMNGTLSは、読み出し/書き込みアクセスが可能なデータスロットを活用する事でこの用途に使う事ができます。このため、データを保存するためだけにEEPROMメモリデバイスを追加する必要はありません。

制限事項: データスロット15を一般的なデータストレージに使う場合、セキュアブート機能は実装できません。

7.2. 開発ツール

ECC608-TMNGTLSは各種のハードウェアおよびソフトウェア ツールと、アプリケーション開発を迅速に進めるためのバックエンド サービスによりサポートされます。初期開発は、使いやすいTrust Platform Design Suiteツールファミリを使って始める事ができます。これらのツールは、ユースケースを実装するためのグラフィカルな手段を提供し、最終的にアプリケーションの実装に必要なCコードを生成します。

定義済みのTrust Platform Design Suiteツールがお客様のアプリケーションに対応していない場合、CryptoAuthLibまたはCryptoAuthLibのPython®バージョンとCryptoAuthToolを使ってアプリケーションを開発できます。CryptoAuthLibは、Trust Platform Design Suiteツールから出力されるコードのバックボーンでもあります。

ハードウェア ツールとECC608-TMNGTLSのサンプルデバイスにより、アプリケーションの完全な検証が可能です。本デバイスのアクセスポリシーは設定済みであるため、お客様はシステムレベルのコード開発に集中できます。

アプリケーションが完成したら、Microchip社からECC608-TMNGTLSデバイスを注文できます。

7.2.1. Trust Platform Design Suite

実装手順を簡素化するため、Microchip社はウェブベースのTrust Platform Design Suiteツールを開発しました。これらのツールは、コンセプト段階から量産段階までお客様の開発を支援します。これらのツールを使うと、ECC608-TMNGTLSのコンフィグレーションと定義済みアクセスポリシーによる制約内で、特定のアプリケーションを実装するために必要なトランザクション ダイアグラムとコードを開発できます。

Note: これらのツールの詳細はMicrochip社の「Trust Platform」情報ページを参照してください。

7.2.2. ハードウェア ツール

ECC608-TMNGTLSを使ったアプリケーションの開発には、各種のハードウェア ツールが役立ちます。本書に記載していないツールについては、Microchip社ウェブサイトをご覧ください。

EA06V72A/EV10E69A - CryptoAuthentication Trust Managerプラットフォーム

EA06V72A/EV10E69AはATSAMD21マイクロコントローラ、ECC608-TMNGTLS Trust Managerデバイス、USBハブ、mikroBUS™コネクタ、オンボード デバッガを備えたコンパクトな開発システムです。TPDS (Trust Platform Design Suite)ツールを使ってECC608-TMNGTLSデバイス向けの各種ユースケースを実装できます。これはKudelski keySTREAM SaaSツールとMicrochip社のTrust Platform Development Suiteツールと組み合わせて使う事が推奨されるキットです。

EA06V72AはECC608-TMNGTLSのアーリーアダプタ向けボードですが、長期的にはEV10E69A量産ボードに置き換わります。このキットは、Trust Manager開発フローとKudelski keySTREAM SaaSツールでTPDSを使う限りは全く同じように動作します。EV10E69AにはEA06V72Aには含まれていない追加のユーザーLEDとユーザースイッチが複数含まれています。

DT100104 ATECC608 Trustボード

DT100104 ATECC608 Trustボードのリビジョン4には1つのECC608-TMNGTLSデバイスの他にATECC608C Trustデバイスが含まれています。このTrust Managerデバイスは、microchipDIRECTで注文可能な、またはTrust Manager Platformボードに搭載されているECC608-TMNGTLSプロトタイプ デバイスと同様の構成です。このボードはmikroBUSフォーマットを備えており、CryptoAuth Trust Manager PlatformまたはCryptoAuth Trust Platform (DM320118)に容易に接続できます。CryptoAuth Trust PlatformとDT100104(Rev 4)ボードを組み合わせてDIPスイッチを適切に設定する事で、CryptoAuth Trust Manager Platformボードと全く同じように動作します。

Secure UDFN/SOIC Clickボード

ECC608-TMNGTLSのSOICまたはUDFNのどちらかのサンプルとTrust Manager Platform開発ボードを合わせて使う場合、Trust Manager Platformボードと連携する専用設計のmikroBUSソケットボードを入手する事を推奨します。これらのソケットボードはmikroElektronikaから提供されており、Trust Manager PlatformボードのmikroBUSヘッダに挿入するよう設計されています。これらのソケットボードを使うと、1つまたは複数のデバイスをプログラムして、それをプロトタイプ ボードまたは量産ボードで使う事が可能になります。ECC608-TMNGTLSのサンプルデバイスを別途入手する必要があります。

- Secure SOIC Clickボード
- Secure UDFN Clickボード

7.2.3. CryptoAuthLib

CryptoAuthLibは、Microchip社のCryptoAuthenticationデバイスファミリをサポートするソフトウェア ライブラリです。ECC608-TMNGTLSを使ったアプリケーションの開発には、このライブラリを使う事を推奨します。このライブラリは、本書に記載したコマンドを実行するために必要なAPI関数の呼び出しを実装します。

このライブラリは、多くのMicrochip社製マイクロコントローラで簡単に動作させる事ができますが、HAL (Hardware Abstraction Layer)を介して他のマイクロコントローラ(他社製品を含む)向けに容易に拡張できます。

これらのツールの詳細は、以下でご覧ください。

- CryptoAuthLib - ウェブリンク
- CryptoAuthLib - GitHub

API関数の呼び出し

本書に記載した各コマンドには、1つまたは複数のAPI呼び出しが割り当てられています。通常、全ての入力パラメータを指定可能なベースAPI呼び出しが存在します。コマンドおよびサブセクションに示されるパラメータは、このコマンドで使えます。各API呼び出しには複数のモードがあります。下表にコマンドとベースAPI呼び出しの例を示します。詳細なAPI情報は、GitHub情報を参照してください。

表7-1. CryptoAuthLib API関数の呼び出しに対するコマンド例

デバイスコマンド	API呼び出し	注釈
Info	atcab_info_base()	
Write	atcab_write()	
Read	atcab_read_zone()	
SHA	atcab_sha_base()	
Sign	atcab_sign_base()	
Random	atcab_random()	
Verify	atcab_verify()	

8. I²Cインターフェイス

I²Cインターフェイスは、SDAピンとSCLピンを使って、各種のI/O状態をECC608-TMNGTLSに対して示します。このインターフェイスは、他のMicrochip社CryptoAuthenticationおよびシリアルEEPROMデバイスとプロトコルレベルで互換となるよう設計されています。正確なコマンドとデバイスI²Cアドレスは異なる場合があります。

ECC608-TMNGTLSクライアントの出力ピンはオープンドレイン ドライバしか備えていないため、SDAピンは外付けプルアップ抵抗を使ってHighに駆動する必要があります。バスホストは、オープンドレインまたはトーテムポールが使えます。後者を使う場合、ECC608-TMNGTLSがバス上でSDAラインを駆動している時にバスホストはトライステートになる必要があります。SCLピンは入力であり、常に外部デバイスまたは外付け抵抗によってHighおよびLowに駆動される必要があります。

ECC608-TMNGTLSでは、既定値の7ビットI²CアドレスはHex 0x38に定義されています。この値の場合、I²Cアドレスの書き込みバイト値はHex 0x70、読み出しバイト値はHex 0x71になります。必要な場合、I²Cアドレス値は`UpdateExtra`コマンドを使って1回に限り上書きできます。

Remember: I²C規格では「マスタ」および「スレーブ」という用語を使いますが、本書では同義のMicrochip社用語として「ホスト」および「クライアント」を使っています。

関連リンク

9.3.1. ACパラメータ: I²Cインターフェイス

8.1. I/O条件

本デバイスは以下のI/O条件に応答します。

8.1.1. スリープ中のデバイス

スリープ中のデバイスはWake条件を除く全ての条件を無視します。

- Wake - SDAがt_{WLO}以上の間Lowを保持した後にSDAの立ち上がりエッジが発生するとデバイスは低消費電力モードを終了します。遅延時間t_{WHI}の後に、デバイスはI²Cコマンドを受信可能となります。
- アイドルまたはスリープ中のデバイスは、t_{WLO}が過ぎるまでSCLピン上の全てのレベルまたは状態遷移を無視します。t_{WHI}中のある時点でSCLピンが有効になり、デバイスは8.1.2.「アクティブ中のデバイス」に記載した条件に応答します。

Wake条件が成立するには、SDAピンがシステム プロセッサによってt_{WLO}の間Lowに駆動されるか、0x00のデータバイトが十分に遅い(すなわちSDAのLow期間がt_{WLO}より長い)クロックレートで転送される必要があります。デバイスが復帰した時点で、通常のプロセッサI²Cハードウェアおよび/またはソフトウェアはデバイス通信用に使用可能となります。デバイス通信には、デバイスを低消費電力(スリープ) モードへ戻すために必要なI/Oシーケンスも含まれます。

ヒント: Wakeパルスは、`0x00`のバイトを100 kHzで送信する事により簡単に生成できます。後続のコマンドは、これより高い周波数で実行できます。

I²Cモードでは、既に復帰済みのデバイスはWakeシーケンスを無視します。

バス上に複数デバイスが存在する場合

バス上に複数のデバイスが存在する場合、I²Cインターフェイスが約300 kHz[2]以下で動作すると、特定データパターンの送信によってバス上のECC608-TMNGTLSが復帰します。周波数が低いほど、デバイスはより確実に復帰します。バスで送信される後続のデバイスアドレスは宛先のデバイスとのみ一致するためECC608-TMNGTLSは応答しませんが、復帰はします。低周波数で他のデバイスと通信した後に、スリープまたはアイドル シーケンスを発行してECC608-TMNGTLSを既知のステートに戻す事を推奨します。

Important: t_{WLO}は、ECC608-TMNGTLSが全ての製造および環境条件で確実に復帰できるようにするためにシステムが提供する必要のある最小時間です。実際には、これよりも短いパルス幅でもデバイスは復帰します。

I²Cバス上に複数のデバイスが存在する場合、電源投入シーケンス後、かつバス上のデバイスと通信する前にECC608-TMNGTLSデバイスを復帰させる事を推奨します。そうする事で、ECC608-TMNGTLSを確実に適切に初期化できます。

関連リンク

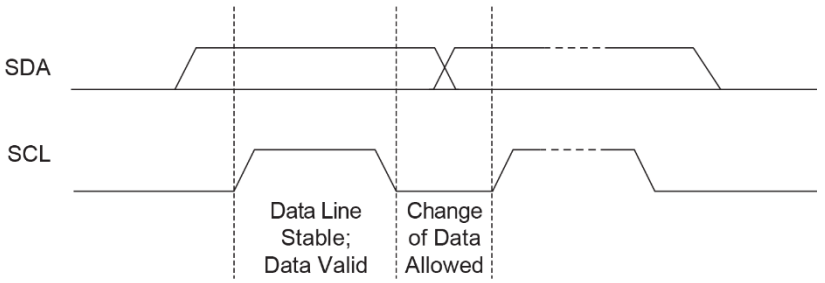
8.1.2. アクティブ中のデバイス
9.3. ACパラメータ: 全I/Oインターフェイス

8.1.2. アクティブ中のデバイス

アクティブ中のデバイスは以下の条件に応答します。

- **DATA = 0**: SCLがLow→High→Lowと遷移する間SDAがLowを保持した場合、「0」のビットがバス上で転送されます。SDAはSCLがLowの時に遷移できます。
- **DATA = 1**: SCLがLow→High→Lowと遷移する間SDAがHighを保持した場合、「1」のビットがバス上で転送されます。SDAはSCLがLowの時に遷移できます。

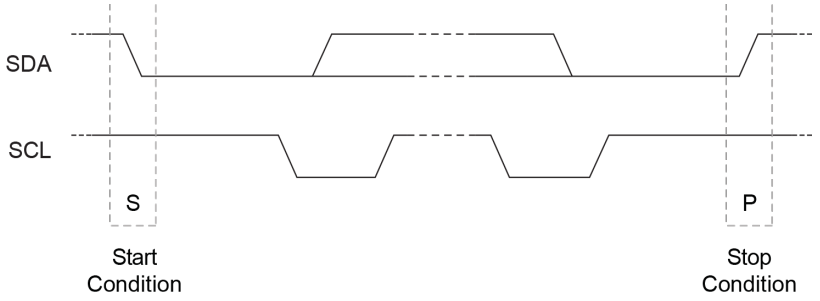
図8-1. I²Cインターフェイスにおけるデータビットの転送



- **スタート条件**: SCLがHighの時にSDAがHighからLowへ遷移するとスタート条件が発生します。全てのコマンドの前にスタート条件が必要です。
- **ストップ条件**: SCLがHighの時にSDAがLowからHighへ遷移するとストップ条件が発生します。デバイスは、ストップ条件を受信した後に現在のI/Oトランザクションを終了します。デバイスはコマンドの実行に必要な全てのバイトを入力で受信すると、ビジー状態に移行してコマンドの実行を開始します。ストップ条件は、デバイスへ送信される全てのパケットの最後で送信される必要があります。

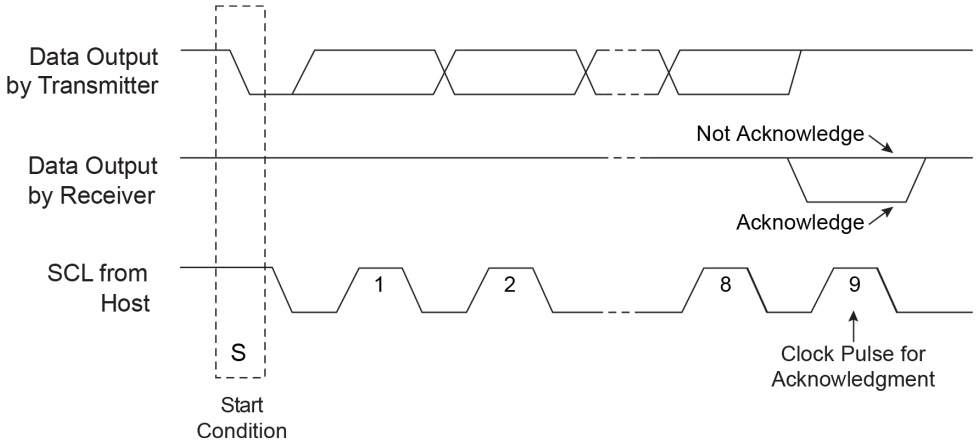
[2] 実際の周波数は、デバイスごとの製造ばらつきと環境要因によって変化します。この値は、全ての条件でデバイスが確実に復帰できるとみなせる周波数です。

図8-2. I²Cインターフェイスのスタート条件とストップ条件



- **肯定応答(ACK)**: 各アドレスバイトまたはデータバイトが転送された後の9番目のクロックサイクルで、レシーバはSDAピンをLowにする事によって、そのバイトを正常に受信した事を知らせます。
- **否定応答(NACK)**: 各アドレスバイトまたはデータバイトが転送された後の9番目のクロックサイクルで、レシーバはSDAピンをHighのままにする事によって、そのバイトの受信に問題があった事またはそのバイトでそのグループの転送が完了する事を知らせます。

図8-3. I²CインターフェイスのNACKおよびACK条件



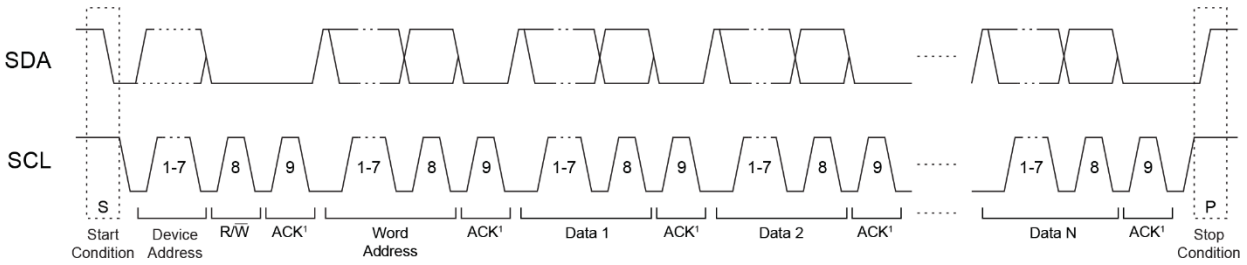
Configurationゾーン内で設定されているI2C_Addressが互いに異なる複数のECC608-TMNGTLSは、同じI²Cインターフェイス信号を容易に共有できます。デバイスアドレスの7ビットは全て設定可能であるため、ECC608-TMNGTLSはシリアルEEPROMを含む任意のI²Cデバイスとの間でI²Cインターフェイスを共有する事もできます。

8.2. ECC608-TMNGTLSへのI²C送信

システムからECC608-TMNGTLSへのデータ送信の概要を表7-4に示します。送信の順序は以下の通りです。

- スタート条件
- デバイスアドレス バイト
- ワードアドレス バイト
- データバイト(1～N) (必要に応じて)
- ストップ条件

図8-4. ECC608-TMNGTLSへの通常のI²C送信



ACK期間中にSDAはECC608-TMNGTLSによってLowに駆動されます。

表7-1に、I/Oトランザクションの各バイトを示します。「I²C名」列は、AT24C16のデータシートに記載されているバイト名です。

表8-1. ECC608-TMNGTLSへのI²C送信

名称	I²C名	概要
Device Address	Device Address	このバイトは、I²Cバス上の特定デバイスを選択します。このバイトのbit 1～7がConfiguration ゾーン内のI2C_Addressバイトのbit 1～7に一致すると、そのECC608-TMNGTLSが選択されます。このバイトのbit 0は標準I²CのR/Wビットであり、書き込み動作(デバイスアドレス バイトに続くバイトをホストからクライアントへ転送する)を示す「0」である事が必要です。
Word Address	Word Address	通常動作では、このバイトの値は0x03である事が必要です。
コマンド	Data1, N	カウント、コマンドパケット、2バイトCRCで構成されるコマンドグループです。CRCは、サイズおよびパケットバイトに対して計算されます。

本デバイスはコマンド入力バッファをFIFOとして扱うため、入力グループは1つまたは複数のI²Cコマンドグループに格納してデバイスへ送信できます。デバイスへ最初に送信されるバイトはカウント(この後にデバイスが受信するバイトの数)です。デバイスは、実行が終了するまでこの数を超える後続の受信バイトを無視します。

システムは、最後のコマンドバイトの後にストップ条件を送信する必要があります。これにより、ECC608- TMNGTLSはコマンドの処理を開始します。ストップ条件を送信しないと、最終的に同期が喪失する可能性があります(同期の回復手順は8.2.2. 「I²Cの同期」参照)。

8.2.1. ワードアドレス値

I²Cパケット書き込み中は、ECC608-TMNGTLSは2番目のバイトをワードアドレスとして解釈します。ワードアドレス値は、表7-2の通りにパケットの機能を示します。

表8-2. ワードアドレス値

名称	値	概要
Reset	0x00	アドレスカウンタをリセットします。次のI²C読み出しまたは書き込みトランザクションは、I/Oバッファの先頭位置で始まります。
Sleep (Low-power)	0x01	ECC608-TMNGTLSは低消費電力スリープモードに移行し、次のWakeフラグまで後続のI/Oトランザクションを全て無視します。デバイスの揮発性ステートは全てリセットされます。
Idle	0x02	ECC608-TMNGTLSはアイドルモードに移行し、次のWakeフラグまで後続のI/Oトランザクションを全て無視します。TempKey、メッセージ ダイジェスト バッファ、代替鍵レジスタの内容は保持されます。

...続き		
名称	**値**	**概要**
コマンド	0x03	後続のバイトを入力コマンドバッファ(前回の書き込み位置の次のアドレス)に書き込みます。これは標準の動作です。
Reserved	0x04～0xFF	これらアドレス値をデバイスへ送信してはいけません。

8.2.2. I²Cの同期

システムリセットやI/Oノイズ等によってシステムとECC608-TMNGTLS上のI/Oポートの間の同期が失われる可能性があります。このような場合、ECC608- TMNGTLSは期待通りに応答できなくなります(スリープ状態になるか、ホストがデータを送信するタイミングでクライアントがデータを送信してしまう等)。再同期するには、以下の手順が必要です。

1. I/Oチャンネルを確実にリセットするため、システムは以下の通りに標準I²Cソフトウェア リセットシーケンスを送信する必要があります。
   - スタートビット条件
   - システムのプルアップ抵抗によってSDAをHighに保持した状態で9サイクルのSCL
   - 次のスタートビット条件
   - ストップビット条件

   以上の手順によって同期が正しく確立されると、読み出しシーケンスの送信が可能になり、ECC608-TMNGTLSはデバイスアドレスに対してACKを返します。データ期間中に、デバイスはデータを返すかバスをフロート状態(システムによって値が0xFFのデータとして解釈される)にします。

   デバイスがデバイスアドレスに対してACKを返した場合、システムは内部アドレスカウンタをリセットする必要があります。これにより、ECC608-TMNGTLSはそれまでに送信された不完全な入力コマンドを無視します。アドレスカウンタは、ワードアドレス0x00(リセット)への書き込みシーケンスを送信した後にストップ条件を生成する事によりリセットできます。
2. デバイスがデバイスアドレスに対してACKで応答しない場合、デバイスはスリープ中である可能性があります。この場合、システムは完全なWakeトークンを送信し、立ち上がりエッジ後にt_{WHI}が過ぎるまで待機する必要があります。その後、システムは読み出しシーケンスを再度送信できます。同期が確立されていれば、デバイスはデバイスアドレスに対してACKを返します。
3. それでもデバイスがデバイスアドレスに対してACKで応答しない場合、デバイスはビジー状態(コマンドの実行中)である可能性があります。システムは最長のt_{EXEC} (max.)が過ぎるまで待機してから読み出しシーケンスを送信する事で、デバイスからACKが返されます。

関連リンク

8.1.1. スリープ中のデバイス
8.1.2. アクティブ中のデバイス

8.3. ECC608-TMNGTLSからのI²C送信

ECC608-TMNGTLSがアクティブであり、かつビジー状態ではない場合、ホストはI²C読み出しを使ってデバイスから現在の出力バッファの内容を読み出せます。有効なコマンド結果が得られた場合、返されるグループのサイズは、実行されたコマンドによって決まります。結果が無効である場合、グループ(および返される最初のバイト)のサイズは常に4(カウントバイト + ステータス/エラーバイト + 2バイトのCRC)です。

表8-3. ECC608-TMNGTLSからのI²C送信

名称	I²C名	方向	概要
Device Address	Device Address	To Client	このバイトはI²Cインターフェイス上の特定デバイスを選択します。このバイトのbit 1～7がConfigurationゾーン内のI2C_Addressバイトのbit 1～7に一致すると、そのECC608- TMNGTLSが選択されます。このバイトのビット0はI²C R/Wビットです。このビットは、読み出し動作(デバイスアドレスに続くバイトはクライアントからホストへ転送される)を指定するために「1」である事が必要です。
データ	Data1, N	To Host	出力グループはカウントバイト + ステータス/エラーバイトまたは出力パケット + 2バイトのCRCで構成されます。

ホストはステータス、エラー、コマンド出力を繰り返し読み出せます。I²Cインターフェイスを介してECC608-TMNGTLSへ`Read`コマンドが送信されるたびに、デバイスは出力バッファ内の次のバイトを送信します。本デバイスによるアドレスカウンタの扱い方は、この後で説明します。

ECC608-TMNGTLSがビジー、アイドル、スリープ状態のいずれかである場合、デバイスは読み出しシーケンス中のデバイスアドレスに対してNACKを返します。部分的なコマンドがデバイスへ送信された後に読み出しシーケンス`[Start + DeviceAddress(R/W == R)]`がデバイスへ送信された場合、ECC608-TMNGTLSはデバイスアドレスに対してNACKを返す事で、読み出し可能なデータが存在しない事を示します。

8.4. スリープ シーケンス

システムは、ECC608-TMNGTLSの使用を終了した時点でスリープ シーケンスを発行してデバイスを低消費電力モードに移行させる事が推奨されます。このシーケンスはデバイスアドレス、値0x01(ワードアドレス)、ストップ条件で構成されます。低消費電力状態に移行すると、デバイスの内部コマンドエンジンと入出力バッファは完全にリセットされます。このシーケンスは、デバイスがアクティブかつ非ビジーの時にいつでもデバイスへ送信できます。

関連リンク

8.1.1. スリープ中のデバイス
8.1.2. アクティブ中のデバイス

8.5. アイドル シーケンス

コマンドの総シーケンス時間が$t_{WATCHDOG}$を超えた場合、デバイスは自動的にスリープに移行し、揮発性レジスタ内の情報は全て失われます。これを防ぐには、ウォッチドッグ期間が終了する前にデバイスをアイドルモードに移行させる必要があります。デバイスは、Wakeトークンを受信した時にウォッチドック タイマを再始動し、実行を継続できます。

このアイドル シーケンスはデバイスアドレス、値0x02(ワードアドレス)、ストップ条件で構成されます。このシーケンスは、デバイスがアクティブかつ非ビジーの時にいつでもデバイスへ送信できます。

9. 電気的特性

9.1. 絶対最大定格

動作温度	-40～+85 °C
保管温度	-65～+150 °C
最大動作電圧	6.0 V
DC出力電流	5.0 mA
全ピンの電圧	-0.5 V～(V_{CC} + 0.5 V)
ESD耐圧:	
HBM(人体モデル)ESD	> 4 kV
CDM(デバイス帯電モデル)ESD	> 1 kV

Note: ここに記載した「絶対最大定格」を超える条件は、デバイスに恒久的な損傷を生じさせる可能性があります。これはストレス定格です。本書の動作表に示す条件外でのデバイスの運用は想定していません。絶対最大定格条件で長期間印加するとデバイスの信頼性に影響が及ぶ可能性があります。

9.2. 信頼性

ECC608-TMNGTLSはMicrochip社の高信頼性CMOS EEPROM製造技術を採用しています。

表9-1. EEPROMの信頼性

パラメータ	Min.	Typ.	Max.	単位
書き込み耐性 @+85 °C (各バイト)	400,000	–	–	書き込みサイクル
データ保持寿命 @+70 °C	15	–	–	年
データ保持寿命 @+55 °C	45	–	–	年
読み出し耐性	制限なし			読み出しサイクル

9.3. ACパラメータ: 全I/Oインターフェイス

図9-1. 復帰タイミング: 全インターフェイス

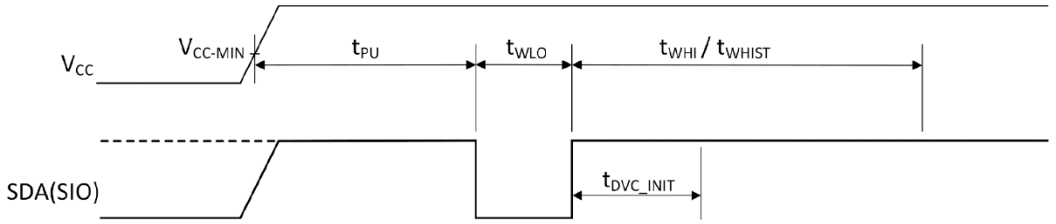


図9-2. 入力ノイズ抑制: 全インターフェイス

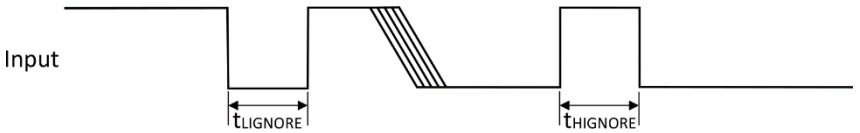


表9-2. ACパラメータ: 全I/Oインターフェイス

パラメータ	シンボル	方向	Min.	Typ.	Max.	単位	条件
電源投入遅延	t_{PU}	To Crypto Device	100	–	–	µs	V_{CC} > V_{CC} minからt_{WLO}の計測を開始するまでの最小時間
復帰Low期間	t_{WLO}	To Crypto Device	60	–	–	µs	–
初期化時間(1)	t_{DVC_INIT}	Internal	–	–	400	µs	復帰パルスの立ち上がりエッジからデバイスが初期化されるまでの最大時間(2)。
データ通信開始前の復帰HIGH遅延	t_{WHI}	To Crypto Device	1500	–	–	µs	ポーリング方式を採用していない場合、この期間中にSDAが安定してHighを保持する事が推奨されます。電源投入時にセルフテスト機能は無効です。
セルフテスト機能が有効な場合の復帰HIGH遅延	t_{WHIST}	To Crypto Device	20	–	–	ms	ポーリング方式を採用していない場合、この期間中にSDAが安定してHighを保持する事が推奨されます。
アクティブ時High レベル グリッチ フィルタ(1)	$t_{HIGNORE\_A}$	To Crypto Device	45	–	–	ns	アクティブ時に、デバイスはそのステートに関係なく、この時間より短いパルスを無視します。
アクティブ時Low レベル グリッチ フィルタ(1)	$t_{LIGNORE\_A}$	To Crypto Device	45	–	–	ns	アクティブ時に、デバイスはそのステートに関係なく、この時間より短いパルスを無視します。
スリープ時Low レベル グリッチ フィルタ(1)	$t_{LIGNORE\_S}$	To Crypto Device	15	–	–	µs	スリープモード時に、デバイスはこの時間より短いパルスを無視します。
ウォッチドッグ タイムアウト	$t_{WATCHDOG}$	To Crypto Device	0.7	1.3	1.7	s	Config.ChipMode[2] = 0の場合、復帰してからデバイスをスリープモードへ移行させるまでの時間です。

Note:

1. これらのパラメータは特性データであり、製造時の検査は実施していません。
2. t_{DVC_INIT}時間が経過するまでは、I²Cバス上で復帰パルス以外の通信を実行しない事を推奨します。

9.3.1. ACパラメータ: I²Cインターフェイス

図9-3. I²C同期データタイミング

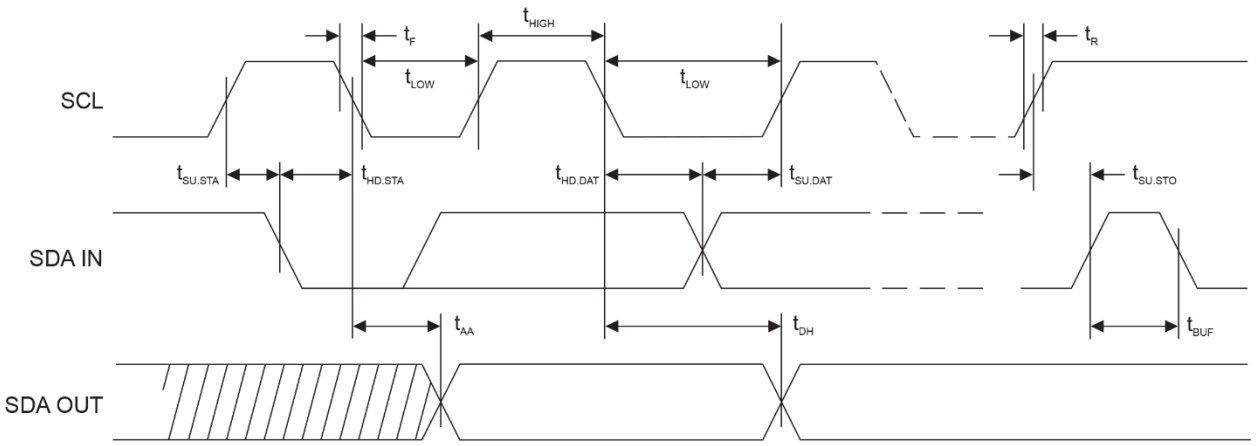


表9-3. I²Cインターフェイスの AC特性[2]
特に明記しない限り、T_A - -40～+85 °C、V_{CC} = +2.0～+5.5 V、C_L = 1 TTLゲート+100 pFの推奨動作レンジに適用

パラメータ	シンボル	Min.	Max.	単位
SCLクロック周波数	f_{SCL}	0	1	MHz
SCL High時間	t_{HIGH}	400	–	ns
SCL Low時間	t_{LOW}	400	–	ns
スタート条件セットアップ時間	$t_{SU.STA}$	250	–	ns
スタート条件ホールド時間	$t_{HD.STA}$	250	–	ns
ストップ条件セットアップ時間	$t_{SU.STO}$	250	–	ns
データ入力セットアップ時間	$t_{SU.DAT}$	100	–	ns
データ入力ホールド時間	$t_{HD.DAT}$	0	–	ns
入力立ち上がり時間[1]	t_R	–	300	ns
入力立ち下がり時間[1]	t_F	–	100	ns
クロックLowからデータ出力確定までの時間	t_{AA}	50	550	ns
データ出力ホールド時間	t_{DH}	50	–	ns
SMBusタイムアウト遅延	$t_{TIMEOUT}$	25	35	ms
次の伝送が開始可能になるまでに必要なバスフリー時間[1]	t_{BUF}	500	–	ns

Note:
1. これらのパラメータは特性データであり、製造時の検査は実施していません。
2. AC計測条件:
   – R_L (SDAとV_{CC}の間を接続): 1.2 kΩ (V_{CC} = +2.0～+5.0 V)
   – 入力パルス電圧: 0.3 V_{CC}～0.7 V_{CC}
   – 入力立ち上がり/立ち下がり時間: ≤ 50 ns
   – 入出力タイミング参照電圧: 0.5 V_{CC}

9.4. DCパラメータ: 全I/Oインターフェイス

表9-4. 全I/Oインターフェイスの DCパラメータ

パラメータ	シンボル	Min.	Typ.	Max.	単位	条件
動作時周囲温度	T_A	-40	–	+85	°C	標準産業用温度レンジ
電源電圧	V_{CC}	2.0	–	5.5	V	—
アクティブ時消費電流	I_{CC}	–	2	3	mA	I/O転送中のI/O待機時または非ECCコマンドの実行時(クロック分周値とは無関係)
		–	–	14	mA	ECCコマンドの実行時(クロック分周比 = 0x0)
アイドル時消費電流	I_{IDLE}	–	800	–	µA	デバイスがアイドルモード中の時、V_{SDA}およびV_{SCL} < 0.4 Vまたは> V_{CC} – 0.4
スリープ電流	I_{SLEEP}	–	30	150	nA	デバイスがスリープモード中の時、V_{CC} ≤ 3.6 V、V_{SDA}およびV_{SCL} < 0.4 Vまたは> V_{CC} – 0.4、T_A ≤ +55 °C
		–	–	2	µA	デバイスがスリープモード中の時 V_{CC}の全レンジおよび-40～85 °Cの温度レンジ

...続き						
パラメータ	シンボル	Min.	Typ.	Max.	単位	条件
出力Low電圧	V_{OL}	–	–	0.4	V	デバイスがアクティブモード中、 V_{CC} = 2.5～5.5 V
出力Low電流	I_{OL}	–	–	4	mA	デバイスがアクティブモード中、 V_{CC} = 2.5～5.5 V、V_{OL} = 0.4 V
ΘJA	Θ_{JA}	–	166	–	°C/W	SOIC (SS)
		–	173	–	°C/W	UDFN (MA)

9.4.1. V_{IH}とV_{IL}の仕様

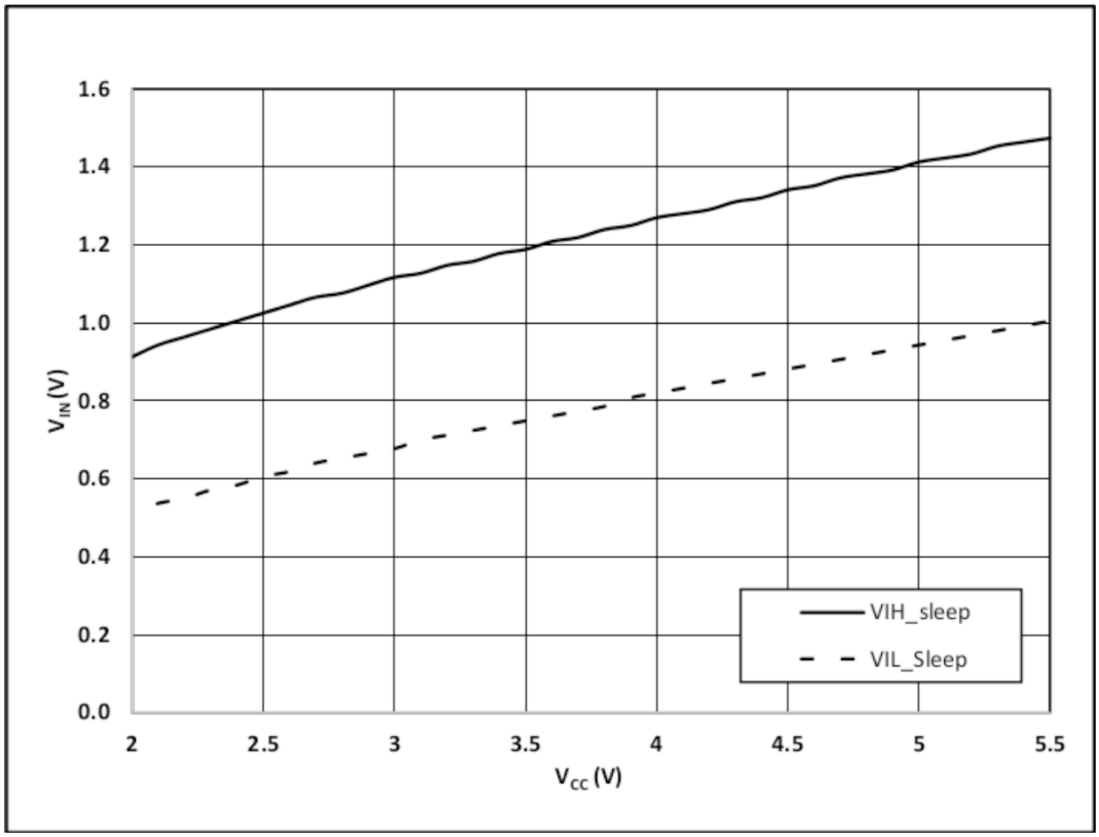
デバイスの入力レベルは、デバイスのモードと電圧に応じて変化します。スリープまたはアイドルモード中の入力電圧しきい値は、図9-4に示す通り、V_{CC} レベルに応じて変化します。スリープまたはアイドルモード中は、TTLenableビットは効力を有しません。

ECC608-TMNGTLSのアクティブ入力レベルは固定されており、V_{CC}レベルと一緒に変化しません。デバイスへ送信される入力レベルは、下表に従う必要があります。

表9-5. 全I/OインターフェイスでのV_{IL}とV_{IH} (TTLenable = 0)

パラメータ	シンボル	Min.	Typ.	Max.	単位	条件
入力Low電圧	V_{IL}	-0.5	–	0.5	V	デバイスがアクティブかつコンフィグレーションメモリ内のTTLenableビットが「0」の場合(これ以外の場合は上記参照)
入力High電圧	V_{IH}	1.5	–	V_{CC} + 0.5	V	デバイスがアクティブかつコンフィグレーションメモリ内のTTLenableビットが「0」の場合(これ以外の場合は上記参照)

図9-4. スリープおよびアイドルモード中のV_{IH}とV_{IL}



10. パッケージ図面

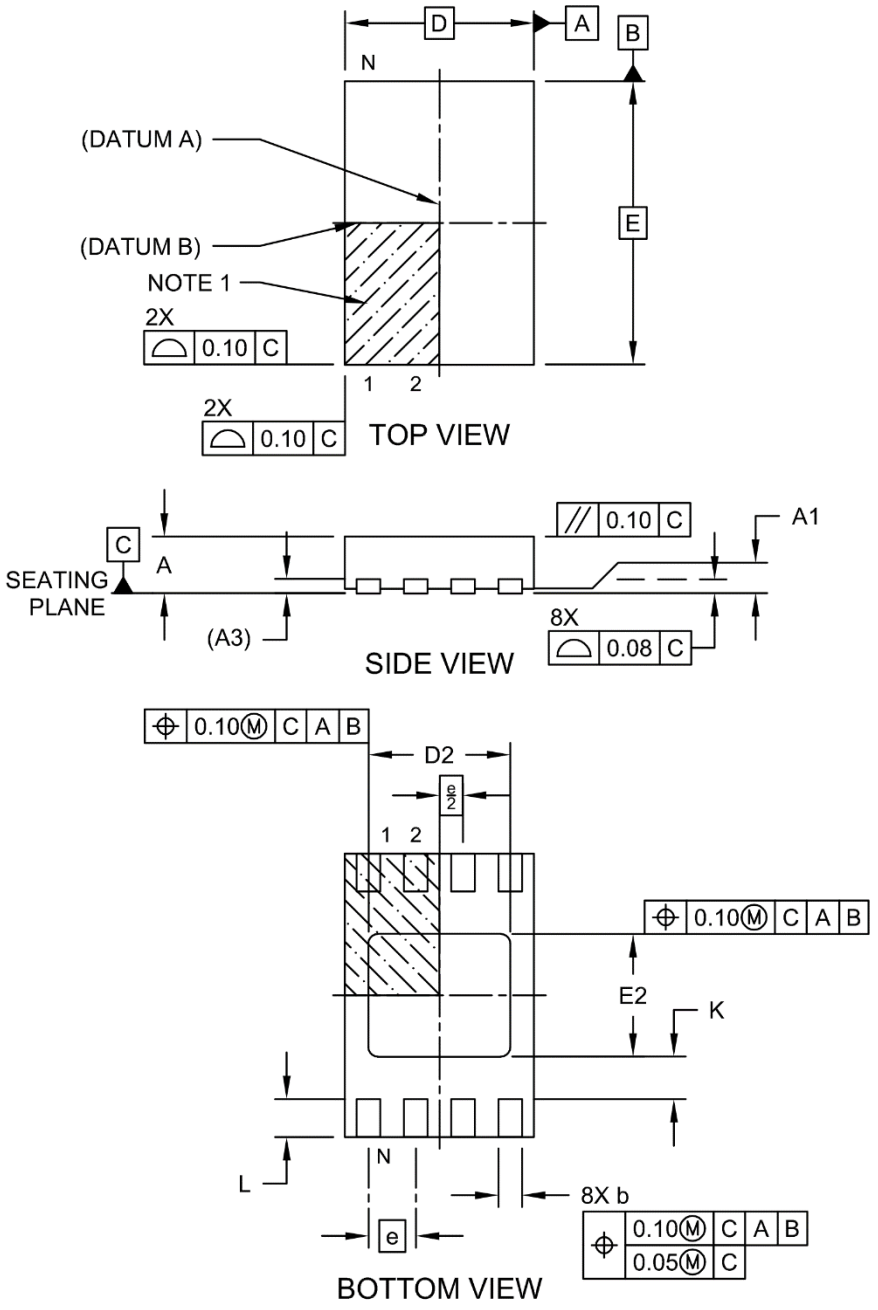
10.1. パッケージのマーキング情報

Microchip社の全体的なセキュリティ対応の一環として、全ての暗号デバイスの製品マーキングは意図的に曖昧にされています。パッケージ上面のマークは、デバイスのタイプやデバイスの製造者に関する情報を一切提供しません。パッケージ上の英数字コードは製造情報を提供し、アセンブリロットに応じて異なります。受領検査時は、パッケージのマーキングに頼らない事を推奨します。

10.2. 8ピンUDFN

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (Q4B) - 2x3 mm Body [UDFN]
Atmel Legacy Global Package Code YNZ

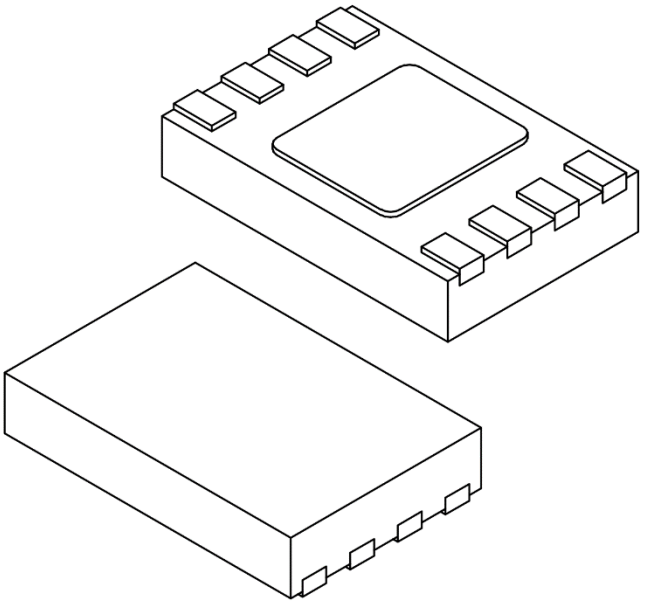
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-21355-Q4B Rev C Sheet 1 of 2

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (Q4B) - 2x3 mm Body [UDFN]
Atmel Legacy Global Package Code YNZ

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		8	
Pitch	e		0.50 BSC	
Overall Height	A	0.50	0.55	0.60
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.152 REF	
Overall Length	D		2.00 BSC	
Exposed Pad Length	D2	1.40	1.50	1.60
Overall Width	E		3.00 BSC	
Exposed Pad Width	E2	1.20	1.30	1.40
Terminal Width	b	0.18	0.25	0.30
Terminal Length	L	0.25	0.35	0.45
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

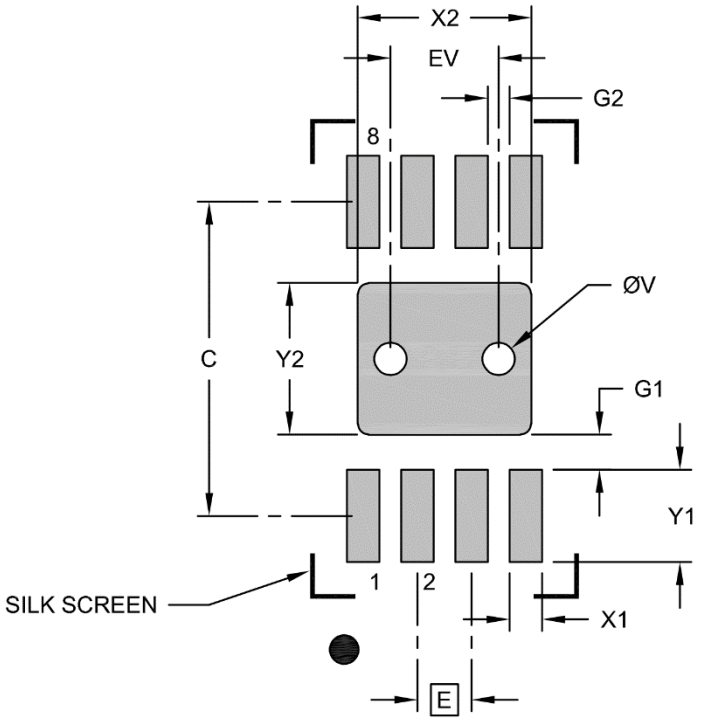
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21355-Q4B Rev C Sheet 2 of 2

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (Q4B) - 2x3 mm Body [UDFN]
Atmel Legacy Global Package Code YNZ

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	X2			1.60
Optional Center Pad Length	Y2			1.40
Contact Pad Spacing	C		2.90	
Contact Pad Width (X8)	X1			0.30
Contact Pad Length (X8)	Y1			0.85
Contact Pad to Center Pad (X8)	G1	0.33		
Contact Pad to Contact Pad (X6)	G2	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

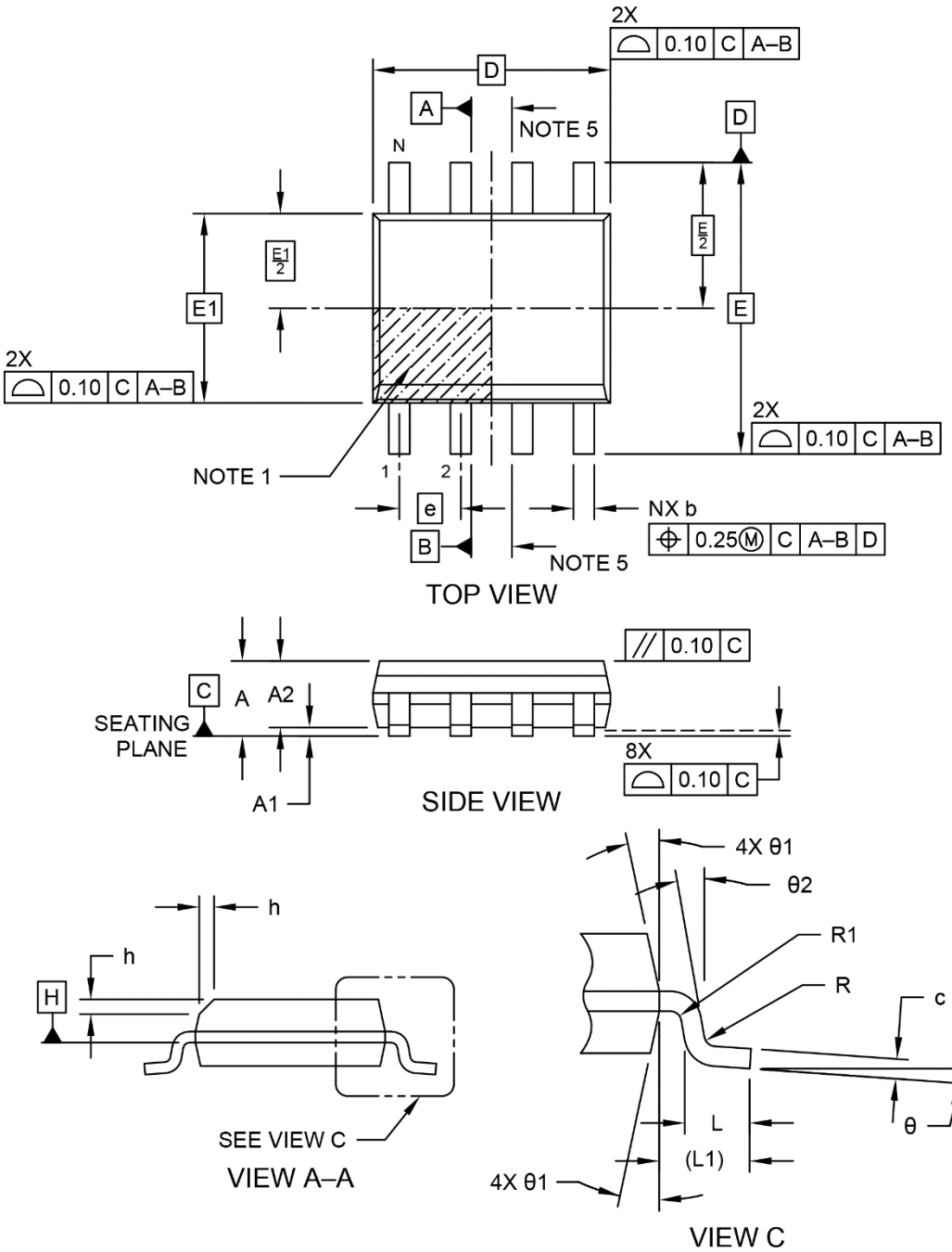
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23355-Q4B Rev C

10.3. 8ピンSOIC

8-Lead Plastic Small Outline (C2X) - Narrow, 3.90 mm (.150 In.) Body [SOIC]
Atmel Legacy Global Package Code SWB

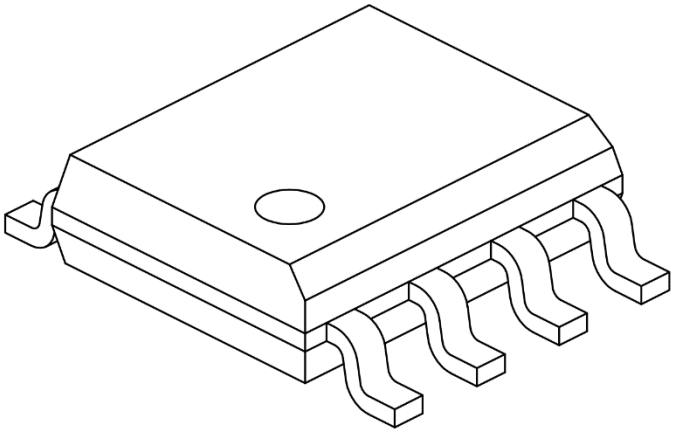
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing No. C04-057-C2X Rev K Sheet 1 of 2

8-Lead Plastic Small Outline (C2X) - Narrow, 3.90 mm (.150 In.) Body [SOIC]
Atmel Legacy Global Package Code SWB

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		8	
Pitch	e		1.27 BSC	
Overall Height	A	–	–	1.75
Molded Package Thickness	A2	1.25	–	-
Standoff §	A1	0.10	–	0.25
Overall Width	E		6.00 BSC	
Molded Package Width	E1		3.90 BSC	
Overall Length	D		4.90 BSC	
Chamfer (Optional)	h	0.25	–	0.50
Foot Length	L	0.40	–	1.27
Footprint	L1		1.04 REF	
Lead Thickness	c	0.17	–	0.25
Lead Width	b	0.31	–	0.51
Lead Bend Radius	R	0.07	–	–
Lead Bend Radius	R1	0.07	–	–
Foot Angle	θ	0°	–	8°
Mold Draft Angle	θ1	5°	–	15°
Lead Angle	θ2	0°	–	–

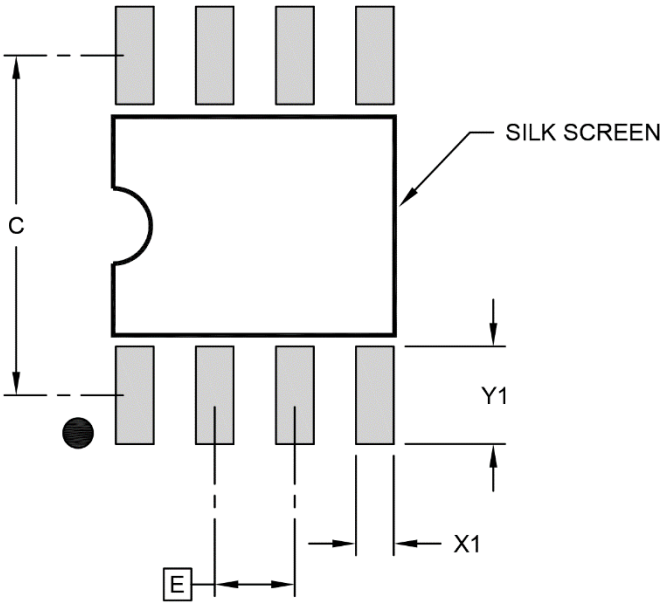
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-057-C2X Rev K Sheet 2 of 2

8-Lead Plastic Small Outline (C2X) - Narrow, 3.90 mm (.150 In.) Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		5.40	
Contact Pad Width (X8)	X1			0.60
Contact Pad Length (X8)	Y1			1.55

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2057-C2X Rev K

11. 改訂履歴

リビジョンA (2024年3月)

- 本書の初版です。

Microchip社の情報

Microchip社ウェブサイト

Microchip社はウェブサイト(www.microchip.com)を通してオンライン サポートを提供しています。当ウェブサイトでは、お客様に役立つ情報やファイルを提供しています。以下を含む各種の情報をご覧になれます。

- **製品サポート** - データシートとエラッタ、アプリケーション ノートとサンプル プログラム、設計リソース、ユーザーガイドとハードウェア サポート文書、最新のソフトウェアと過去のソフトウェア
- **技術サポート** - FAQ(よく寄せられる質問)、技術サポートのご依頼、オンライン ディスカッション グループ、Microchip社のデザイン パートナー プログラムおよびメンバーリスト
- **ご注文とお問い合わせ** - 製品セレクタと注文ガイド、最新プレスリリース、セミナー/イベントの一覧、お問い合わせ先(営業所/正規代理店)の一覧

お客様への通知サービス

Microchip社のお客様への通知サービスは、お客様にMicrochip社製品の最新情報をお届けする配信サービスです。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、リビジョン、エラッタ情報をいち早くメールにてお知らせします。

http://www.microchip.com/pcnにアクセスし、登録手続きをしてください。

お客様サポート

Microchip社製品をお使いのお客様は、以下のチャンネルからサポートをご利用頂けます。

- 正規代理店
- 技術サポート

サポートは正規代理店にお問い合わせください。本書の最後のページに各国の営業所の一覧を記載しています。

技術サポートは以下のウェブページからもご利用頂けます。www.microchip.com/support

製品識別システム

ご注文や製品の価格、納期につきましては正規代理店にお問い合わせください。

PART NO	-	Trust Type	Trust Option	Package Type	-	Shipping Option
xxxxx	-	ttt	vvv	p	-	x

デバイス:	ECC608: セキュアなハードウェア ベース鍵ストレージを備えた設定済み暗号コプロセッサ	
Trustのタイプ	TMNG	Microchip社Trust製品のタイプ
Trustオプション	TLS	Trust製品に関連するコンフィグレーション。
パッケージ オプション	U	8ピン 2 x 3 x 0.6 mmボディ、熱的に強化されたPlastic Ultra Thin Dual Flat (UDFN)鉛フリー パッケージ
	S	8ピン(0.150"幅ボディ)、Plastic Gull Wing Small Outline (JEDEC SOIC)
梱包形態(1,2)		必要最小注文数量は2k個です。
	B	10ユニットバルク - プロトタイプ ユニット

例:

- ECC608-TMNGTLSU: Trust Manager TLS、プロビジョニング済み、8-UDFN、2k個MOQ、I²Cインターフェイス
- ECC608-TMNGTLSU-B: Trust Manager TLS、プロビジョニング済みプロトタイプ、8-UDFN、10個バルク、I²Cインターフェイス
- ECC608-TMNGTLSS: Trust Manager TLS、プロビジョニング済み、8-SOIC、2k個MOQ、I²Cインターフェイス
- ECC608-TMNGTLSS-B: Trust Manager TLS、プロビジョニング済みプロトタイプ、8-SOIC、10個バルク、I²Cインターフェイス

Note:

1. 量産注文はテープ&リールとして納品されます。リールの実際のサイズはお客様のご注文により異なります。許容MOQ(最小注文数量)は2k個です。
2. プロトタイプ ユニットはI²Cインターフェイスの10個のサンプル数量のみで、SOICまたはUDFNパッケージ オプションで提供できます。

Microchip社のデバイスコード保護機能

Microchip 社製品のコード保護機能について以下の点にご注意ください。

- Microchip社製品は、該当するMicrochip 社データシートに記載の仕様を満たしています。
- Microchip社では、通常の条件ならびに動作仕様書の仕様に従って使った場合、Microchip 社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- Microchip社はその知的財産権を重視し、積極的に保護しています。Microchip 社製品のコード保護機能の侵害は固く禁じられており、デジタル ミレニアム著作権法に違反します。
- Microchip社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、Microchip 社が製品を「解読不能」として保証するものではありません。コード保護機能は常に進化しています。Microchip 社では、常に製品のコード保護機能の改善に取り組んでいます。

法律上の注意点

本書および本書に記載されている情報は、Microchip 社製品を設計、テスト、お客様のアプリケーションと統合する目的を含め、Microchip 社製品に対してのみ使う事ができます。それ以外の方法でこの情報を使う事はこれらの条項に違反します。デバイス アプリケーションの情報は、ユーザーの便宜のためにのみ提供されるものであり、更新によって変更となる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。その他のサポートはMicrochip 社正規代理店にお問い合わせ頂くか、https://www.microchip.com/en-us/support/design-help/client-support-servicesをご覧ください。

Microchip 社は本書の情報を「現状のまま」で提供しています。Microchip 社は明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、非侵害性、商品性、特定目的への適合性の暗黙的保証、または状態、品質、性能に関する保証をはじめとするいかなる類の表明も保証も行いません。

いかなる場合もMicrochip 社は、本情報またはその使用に関連する間接的、特殊的、懲罰的、偶発的または必然的損失、損害、費用、経費のいかんにかかわらず、またMicrochip 社がそのような損害が生じる可能性について報告を受けていた場合あるいは損害が予測可能であった場合でも、一切の責任を負いません。法律で認められる最大限の範囲を適用しようとも、本情報またはその使用に関連する一切の申し立てに対するMicrochip 社の責任限度額は、使用者が当該情報に関連してMicrochip 社に直接支払った額を超えません。

Microchip 社の明示的な書面による承認なしに、生命維持装置あるいは生命安全用途にMicrochip社の製品を使う事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、Microchip 社は擁護され、免責され、損害をうけない事に同意するものとします。特に明記しない場合、暗黙的あるいは明示的を問わず、Microchip社が知的財産権を保有しているライセンスは一切譲渡されません。

商標

Microchip 社の名称とロゴ、Microchip ロゴ、Adaptec、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi ロゴ、MOST、MOST ロゴ、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST ロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron、XMEGA は米国とその他の国におけるMicrochip Technology Incorporated の登録商標です。

ISBN: 978-1-6683-4860-4

Quality Management System品質管理システム

Microchip社の品質管理システムについてはwww.microchip.com/qualityをご覧ください。

各国の営業所とサービス

南北アメリカ	アジア/太平洋	アジア/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術サポート: http://www.microchip.com/support URL: www.microchip.com	**オーストラリア - シドニー** Tel: 61-2-9868-6733	**インド - バンガロール** Tel: 91-80-3090-4444	**オーストリア - ヴェルス** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393
アトランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455	**中国 - 北京** Tel: 86-10 -8569-7000	**インド - ニューデリー** Tel: 91-11-4160-8631	**デンマーク - コペンハーゲン** Tel: 45-4485-5910 Fax: 45-4485-2829
オースティン、TX Tel: 512-257-3370	**中国 - 成都** Tel: 86-28-8665-5511	**インド - プネ** Tel: 91-20-4121-0141	**フィンランド - エスポー** Tel: 358-9-4520-820
ボストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088	**中国 - 重慶** Tel: 86-23-8980-9588	**日本 - 大阪** Tel: 81-6-6152-7160	**フランス - パリ** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79
シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075	**中国 - 東莞** Tel: 86-769-8702-9880	**日本 - 東京** Tel: 81-3-6880-3770	**ドイツ - ガーヒンク** Tel: 49-8931-9700
ダラス Addison, TX Tel: 972-818-7423 Fax: 972-818-2924	**中国 - 広州** Tel: 86-20-8755-8029	**韓国 - 大邱** Tel: 82-53-744-4301	**ドイツ - ハーン** Tel: 49-2129-3766400
デトロイト Novi, MI Tel: 248-848-4000	**中国 - 杭州** Tel: 86-571-8792-8115	**韓国 - ソウル** Tel: 82-2-554-7200	**ドイツ - ハイルブロン** Tel: 49-7131-72400
ヒューストン、TX Tel: 281-894-5983	**中国 - 香港SAR** Tel: 852-2943-5100	**マレーシア - クアラルンプール** Tel: 60-3-7651-7906	**ドイツ - カールスルーエ** Tel: 49-721-625370
インディアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380	**中国 - 南京** Tel: 86-25-8473-2460	**マレーシア - ペナン** Tel: 60-4-227-8870	**ドイツ - ミュンヘン** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44
ロサンゼルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800	**中国 - 青島** Tel: 86-532-8502-7355	**フィリピン - マニラ** Tel: 63-2-634-9065	**ドイツ - ローゼンハイム** Tel: 49-8031-354-560
ローリー、NC Tel: 919-844-7510	**中国 - 上海** Tel: 86-21-3326-8000	**シンガポール** Tel: 65-6334-8870	**イスラエル - ラーナナ** Tel: 972-9-744-7705
ニューヨーク、NY Tel: 631-435-6000	**中国 - 瀋陽** Tel: 86-24-2334-2829	**台湾 - 新竹** Tel: 886-3-577-8366	**イタリア - ミラノ** Tel: 39-0331-742611 Fax: 39-0331-466781
サンノゼ、CA Tel: 408-735-9110 Tel: 408-436-4270	**中国 - 深圳** Tel: 86-755-8864-2200	**台湾 - 高雄** Tel: 886-7-213-7830	**イタリア - パドヴァ** Tel: 39-049-7625286
カナダ - トロント Tel: 905-695-1980 Fax: 905-695-2078	**中国 - 蘇州** Tel: 86-186-6233-1526	**台湾 - 台北** Tel: 886-2-2508-8600	**オランダ - ドリューネン** Tel: 31-416-690399 Fax: 31-416-690340
	中国 - 武漢 Tel: 86-27-5980-5300	**タイ - バンコク** Tel: 66-2-694-1351	**ノルウェー - トロンハイム** Tel: 47-7288-4388
	中国 - 西安 Tel: 86-29-8833-7252	**ベトナム - ホーチミン** Tel: 84-28-5448-2100	**ポーランド - ワルシャワ** Tel: 48-22-3325737
	中国 - 厦門 Tel: 86-592-2388138		**ルーマニア - ブカレスト** Tel: 40-21-407-87-50
	中国 - 珠海 Tel: 86-756-3210040		**スペイン - マドリッド** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91
			スウェーデン - ヨーテボリ Tel: 46-31-704-60-40
			スウェーデン - ストックホルム Tel: 46-8-5090-4654
			イギリス - ウォーキンガム Tel: 44-118-921-5800 Fax: 44-118-921-5820