Renesas Synergy™ プラットフォーム

Synergy デバイス識別アプリケーション

R11AN0348JU0100
Rev.1.00
2019.4.18

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

要旨（Introduction）

このアプリケーションノートは IoT セキュリティ全般について説明し、さまざまな鍵（key）生成オプションを含め、Synergy MCU が提供しているセキュリティ機能を簡単に紹介します。このパッケージに付属のサンプルアプリケーションは、セキュア暗号化エンジン（Secure Crypto Engine: SCE）モジュールを使用して、ハードウェアベースの一意のデバイス ID（identity）を生成します。セキュリティ MPU（Security MPU）とフラッシュアクセスウィンドウ（Flash Access Window: FAW）を使用して、この ID を内部フラッシュ（internal flash）にセキュアに保存することができます。

このアプリケーションノートは、Synergy SCE モジュールを利用した製品設計をガイドします。このガイドに従って開発中の製品設計に SCE Synergy ソフトウェアパッケージ (SSP) モジュールを追加し、対象アプリケーションに合わせて正しく設定した後、付属しているサンプルアプリケーションコードを参照してコードを作成できます。API に関するより詳細な説明と、このモジュールのより高度な使用方法を示す他のアプリケーションプロジェクトを掲載した参考資料は、『Synergy ソフトウェアパッケージ (SSP) ユーザーズマニュアル』に掲載されており、より複雑な設計を実施する場合にもご活用いただけます。

現在、Synergy デバイス ID アプリケーション (デバイス識別アプリケーション) は、PK-S5D9/AE-CLOUD2 キットを使用して実装およびテストされています。他の Synergy キットや MCU への対応は、今後のリリースで提供する予定です。

必須リソース（Required Resources）

Synergy デバイス ID サンプルアプリケーションをビルドして実行するには、以下のリソースが必要です。

開発ツールとソフトウェア

- e² studio ISDE v6.2.1 またはそれ以降 (renesassynergy.com/devtools/e2studio)、あるいは IAR Embedded Workbench® for Renesas Synergy™ v8.23.1 またはそれ以降 (renesassynergy.com/devtools/ewsyn)
- Synergy ソフトウェアパッケージ (SSP) v1.5.0 またはそれ以降 (renesassynergy.com/software/ssp)、あるいは Synergy Standalone Configurator (SSC) 6_2_1_R20180629 またはそれ以降 (renesassynergy.com/devtools/ssc)
- SEGGER J-link® USB ドライバ (renesassynergy.com/devtools/jlink)

ハードウェア

- Renesas Synergy PK-S5D9 キット (renesassynergy.com/kits/pk-s5d9) または AE-Cloud2 キット
- Windows® 7/10 OS が動作しているテスト 用 Windows PC
- 2 本の Micro USB ケーブル

前提条件と対象ユーザ（Prerequisites and Intended Audience）

このアプリケーションノートは、Renesas e² studio ISDE と Synergy ソフトウェアパッケージ (SSP) の使用経験があることを前提としています。このアプリケーションノートの手順を実行する前に、『SSP ユーザーズマニュアル』の手順に従い「Blinky」プロジェクトをビルドして実行する必要があります。それにより、e² studio と SSP の使用に慣れ、ボードへのデバッグ接続が適切に機能していることを確認できます。加えて、このアプリケーションノートは、暗号化と、Synergy 暗号化エンジンに関する知識があることも前提としています。

対象読者は、Renesas Synergy™ S5 または S7 MCU シリーズと SCE モジュールを使用したアプリケーション開発に従事するユーザです。

目次

1. IoT セキュリティの要旨（Introduction to IoT Security）

この章は IoT セキュリティの概要 (全般) について説明し、Synergy MCU が提供しているセキュリティ機能のさまざまな側面を紹介します。

1.1 概要（Overview）

一般的な IoT (Internet of Things：モノのインターネット) 環境は、以下の要素で形成されています。

- IoT デバイス（IoT Devices）
- クラウドサーバ（Cloud Server）
- デバイス管理サービス（Device Management services）
- 認証局 (Certificate Authority：CA)

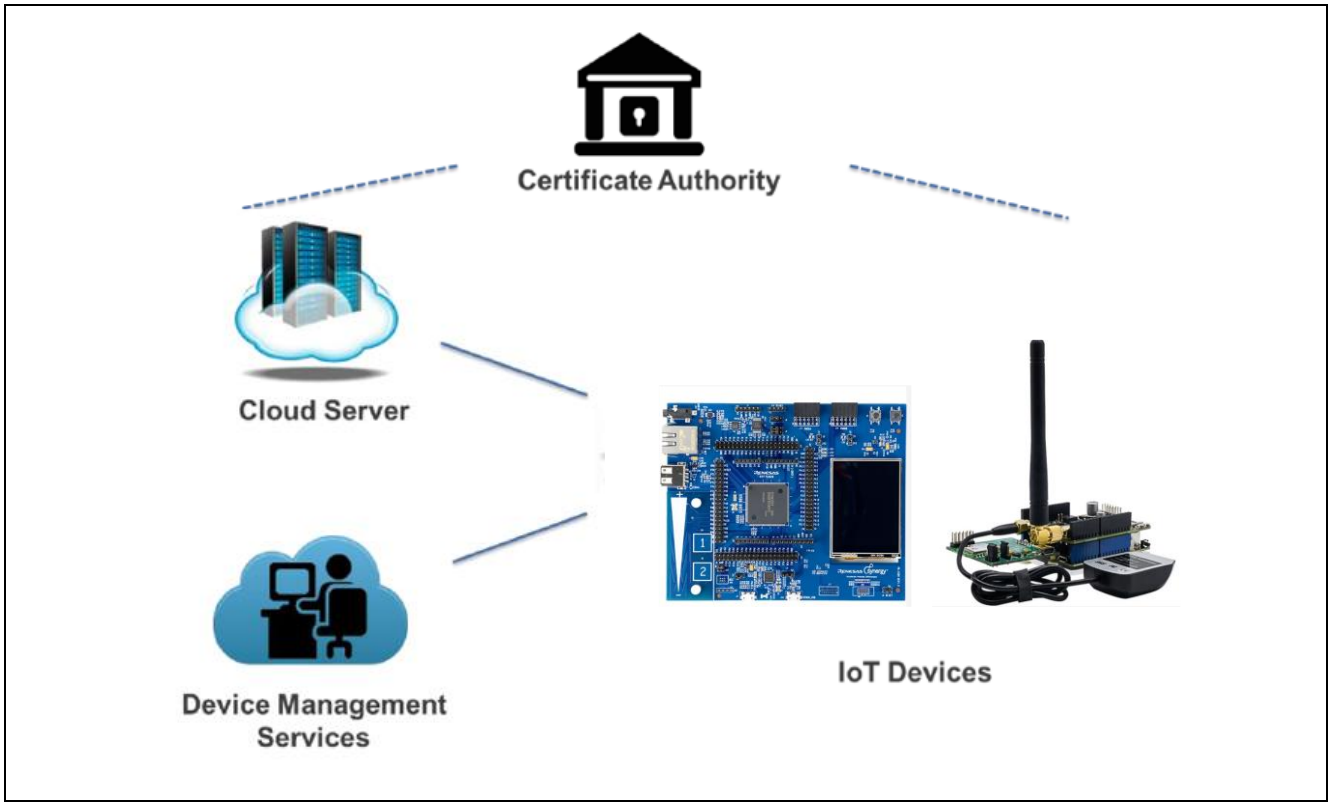


図 1　IoT 環境の概要

IoT デバイス

IoT デバイスは、ソフトウェア、センサ、接続機能を組み込んだ物理デバイスのネットワークを構築しています。これらのデバイスは、これらの機能を使用して、接続や、データの収集と交換を行うことができます。デバイスは、セキュアな場所とセキュアでない場所のどちらにも配置できますが、全体として攻撃（attack）に対して脆弱になってきています。

クラウドサーバ

サービスと、これらのデバイスへのアクセスを提供するポータル（portal）です。通常、非常にセキュアで管理されたデータセンターに配置されています。

デバイス管理サービス

デバイス管理サービスは、IoT デバイス管理機能の包括的なサービスを提供し、さまざまな規模の IoT 顧客が自社のデバイスとデータを完全に管理できるようにしています。以下のものが該当しますが、これらに限定されません。

- アプリケーションセキュリティ
  クラウドサーバを識別する鍵/証明書（certificate）
- デバイス管理セキュリティ

  — 一意の各 IoT デバイスを識別する鍵/証明書

— デバイス管理サービスサーバを識別する鍵/証明書
— 初期のファームウェア展開（deployment）とそれ以降のファームウェア更新。ファームウェアは、自らの正当性（authenticity）を検証するためのシグネチャ（signature）を保持しており、暗号化することも可能です。

認証局（Certificate Authority）(CA)
証明書発行サービスを実施する、認証されかつ信頼された存在（entity）を一般的に CA と呼びます。証明書を使用して公開鍵（public key）に ID を割り当てることができるので、このような鍵を持つデバイスも ID を取得できます。鍵に対応する証明書を生成するプロセスは、セキュリティスキーム（security scheme）の一部として、適切に定義されたプロセスです。認証局は、パブリック（public、公的）とプライベート（private、自営）のどちらかの方式で使用できます。開発デバイスを厳格なエコシステム内で管理する場合 (例えば、産業用環境向けのデバイス)、CA をプライベートにする可能性が高いと考えられます。開発デバイスをコンシューマチャネル (消費者市場) 全般に配布し、さまざまなベンダがサービスとハードウェアを提供する可能性が高い場合 (例えば、監視カメラ、サーモスタット、ホームセキュリティシステムなど)、CA をパブリック CA にする可能性が高いと考えられます。

1.2 IoT エコシステム内でのデバイス ID の重要性（Importance of Device Identity in an IoT Ecosystem）

強力なデバイス ID の確立によって、複数の IoT デバイスを接続し、他のデバイス、サービス、およびユーザとの間のセキュアかつ暗号化された通信を保証しようとするときに、各 IoT デバイスを一意に識別かつ認証することができます。

強力なデバイス ID は、IoT セキュリティの重要な要件（core IoT security requirements）を満たすことができます。

- 信頼（Trust）
  デバイスをネットワークに接続するとき、そのデバイスを認証し、他のデバイス、サービス、およびユーザとの間で信頼を確立する必要が生じます。信頼を確立した後、デバイス、ユーザ、およびサービスはセキュアな方法で通信を実施し、暗号化されたデータと情報を交換することができます。
- プライバシー（Privacy）
  ネットワーク接続型の IoT デバイスが多くなるほど、より多くのデータの生成、収集、共有が行われます。このようなデータに該当するのは、個人、機密、財務などの情報であり、プライベートかつセキュアな状態に維持することが必須です。多くの場合、このようなデータは法令順守の対象になっています。IoT デバイスを互いに接続するとき、デバイス ID は認証と識別を確実に実施するための手段になります。
- 整合性（Integrity）
  デバイス整合性（device integrity）は、デバイスと、IoT エコシステム内で送信されているデータの両方に適用されます。デバイス整合性の基礎は、デバイスが自らの情報を他者に通知するときに、その内容を証明できるようにすることです。強力な一意のデバイス ID が存在する場合、そのデバイスが正当な存在であることを証明できます。この結果、偽造製品を減らし、会社のブランドを保護することができます。データ整合性（data integrity）はしばしば見過ごされやすい要件ですが、ネットワーク接続型のデバイスとシステムは、送信される情報の正当性と信頼性に依存しています。

1.3 Synergy ハードウェアのセキュリティ機能（Synergy Hardware Security Features）

Synergy MCU は、セキュアなメモリ領域に配置された、ファームウェアモジュールのみがアクセスできる保護されたメモリ (読み書き可能) を通じて、ハードウェアによる信頼の基点（root-of-trust）を実現しています。Synergy デバイスが実現する保護されたメモリ機能は、セキュアブートコード（secure boot code）と、デバイスの証明書/鍵の格納に使用できます。デバイス ID アプリケーションにとって不可欠な機密データは複数存在しますが、証明書/鍵はその中でも特に重要です。

1.3.1 ARM MPU

ARM MPU はメモリ保護（Memory Protection）機能であり、CPU はこの機能を使用して、さまざまなメモリ領域（memory region）に対して、「特権アクセスのみ、またはフルアクセス」といったメモリアクセス権限（memory access permission）と、「バッファリング可能、またはキャッシング可能」といったメモリ属性（memory attribute）を定義できます。

- ARM MPU は最大 8 個のプログラマブルなメモリ領域をサポートでき、各領域には独自のプログラマブルな開始アドレス、サイズ、設定を割り当てることができます。
- メモリ保護ユニット（Memory Protection Unit、MPU）を使用すると、プログラミングのエラー (未発見のバグ) や、システムまたはハードウェアの不良に起因するエラーまで、多くの潜在的なエラーからアプリケーションを保護できます。

ARM MPU 構成の詳細は、このアプリケーションプロジェクトの範囲外であり、ここでは説明しません。ARM MPU の動作と機能の詳細は、『Synergy MCU ハードウェアユーザーズマニュアル』を参照してください。

1.3.2 バスマスタ MPU（Bus Master MPU）

バスマスタ MPU は、MCU のアドレス空間全体 (0000 0000h ～ FFFF FFFFh) で、バスマスタがアクセスするアドレスを監視します。読み取りと書き込みの権限で形成されたアクセス制御情報（access control information）は、最大 32 の領域に対して互いに独立した形で設定できます。バスマスタ MPU は、これらの設定に基づいて各領域へのアクセスを監視します。

- 保護された領域へのアクセスを検出した場合、バスマスタ MPU はリセット（reset）またはノンマスカブル割り込み（non-maskable interrupt）を生成します。
- 以下のような 3 つのバスマスタ MPU グループがあります。
  — DMA バス (グループ A)
  — ETHER (イーサ) バス (グループ B)
  — GPX バス (グループ C)
- セキュア MPU 領域は、バスマスタから保護されます。

バスマスタ MPU 構成の詳細は、このアプリケーションプロジェクトの範囲外であり、ここでは説明しません。バスマスタ MPU の動作と機能の詳細は、『Synergy MCU ハードウェアユーザーズマニュアル』を参照してください。

1.3.3 バススレーブ MPU（Bus Slave MPU）

バススレーブ MPU は、フラッシュや SRAM など、バススレーブ機能へのアクセスを監視します。この機能にアクセスできるのは、4 個のバスマスタ、すなわち CPU と、バスマスタ MPU のグループ A、B、C であり、以下の方法で利用できます。

- バススレーブ MPU は、4 個のバスマスタのそれぞれに対応する個別の保護レジスタを搭載しており、読み取り権限と書き込み権限で形成された、互いに独立したアクセス保護制御機能を実現します。
- 保護された領域へのアクセスを検出した場合、バススレーブ MPU はリセットまたはノンマスカブル割り込みを生成するほか、バスエラーアドレス（bus error address）、バスエラーステータス（bus error status）、エラーアクセスステータス（error access status）を格納することができます。

バススレーブ MPU 構成の詳細は、このアプリケーションプロジェクトの範囲外であり、ここでは説明しません。バススレーブ MPU の動作と機能の詳細は、『Synergy MCU ハードウェアユーザーズマニュアル』を参照してください。

1.3.4 セキュリティ MPU（Security MPU）

Synergy MCU は、4 個のセキュア領域に対応する 1 個のセキュリティ MPU を搭載しています。4 個のセキュア領域に該当するのは、コードフラッシュ内の個別領域、SRAM 内の個別領域、および 2 個のセキュリティ機能領域です。これらの領域にアクセスできるのは、「セキュアコード」（secure code）のみです。セキュア領域は、以下のものによる未承認アクセス（unauthorized access）から保護されます。

- セキュアではないプログラム
- DMA や DTC のような他のバスマスタ
- デバッガインタフェース

このメカニズムにより、信頼されたコードとともに、信頼されていないコードが存在して動作することを許可できます。セキュリティ MPU の設定はフラッシュ内に保存され、リセットベクタ（reset vector）をフェッチする前に有効にされます。

セキュリティ MPU の詳細は、『Synergy MCU ハードウェアユーザーズマニュアル』を参照してください。

1.3.5 FAW (フラッシュアクセスウィンドウ)（FAW (Flash Access Window)）

フラッシュアクセスウィンドウ (FAW) レジスタは、コードフラッシュ（code flash）のアドレス範囲を設定する目的で使用します。コードフラッシュ領域は消去とプログラム (書き込み) が可能です。この範囲の外部にあるアドレスを、フラッシュアクセスウィンドウの外部領域（outside the Flash Access Window）と呼び、一度プログラムした（書き込んだ）後は変更できません。この機能を使用して、デバイス ID (鍵/証明書) が消去または再プログラムされることを防止します。

このパッケージに付属しているサンプルアプリケーションプロジェクトは、デバイス ID のユースケース向けに、SSP が提供する API を使用してフラッシュアクセスウィンドウを構成するための参照コード（プログラム）を提供しています。ユーザはこのアプリケーションプロジェクトの残りの部分で、FAW 構成に関連する他のユースケース向けにセキュアデータを参照することもできます。

FAW の詳細は、「参考資料」の章に掲載の SSP ユーザーズマニュアルのリンクを参照してください。

注記: FAW は書き込み可能なフラッシュ領域に対して設定するものです。そのため、ロックされている (書き込み禁止) メモリ領域は、FAW アドレス範囲の外側を指します。

FSPR (ワンタイムプログラマブル設定)

FSPR ビットを使用して、FAW レジスタと MPU レジスタを永続的に設定することができます。このビットはワンタイムプログラマブル (1 回限り書き込み可能) です。このビットをセットする（1 に設定する）場合は、

すべての設定を確認し、デバイスを量産工程から送り出す準備ができたときのみにしてください。

Renesas Secure Boot Manager (セキュアブートマネージャ) を採用するユースケースでは、ブートローダ（boot loader）のコード内で FSPR のセットを実施します。この場合、デバイスの証明書/鍵をセキュアに格納する目的でメモリ領域を割り当てる必要のあるアプリケーションは、セキュアブートローダ設計の一環として考慮する必要があります。

FAW レジスタと MPU レジスタがフィールド (市場) で変更（改ざん）される事態を防止するために、このビットをセットすることは重要です。例えば、セキュリティ MPU を使用する場合、FAW の設定と FSPR ビットのセットを行い、MPU の設定をロックする必要があります。

1.3.6　セキュア暗号化エンジンモジュール（Secure Crypto Engine Module）

セキュア暗号化エンジン (SCE) は、Synergy ハードウェアの周辺機能であり、いくつかのセキュリティ機能や、NIST 認定済みアルゴリズム（NIST certified algorithm）、複数の基本的な暗号化アルゴリズムのサポートを搭載しています。

Synergy は 3 つのバリエーションのセキュリティハードウェアがあります。

以下の図のように MCU のシリーズに依存します。

- SCE 7
- SCE 5
- セキュリティと暗号化（Security and Encryption）

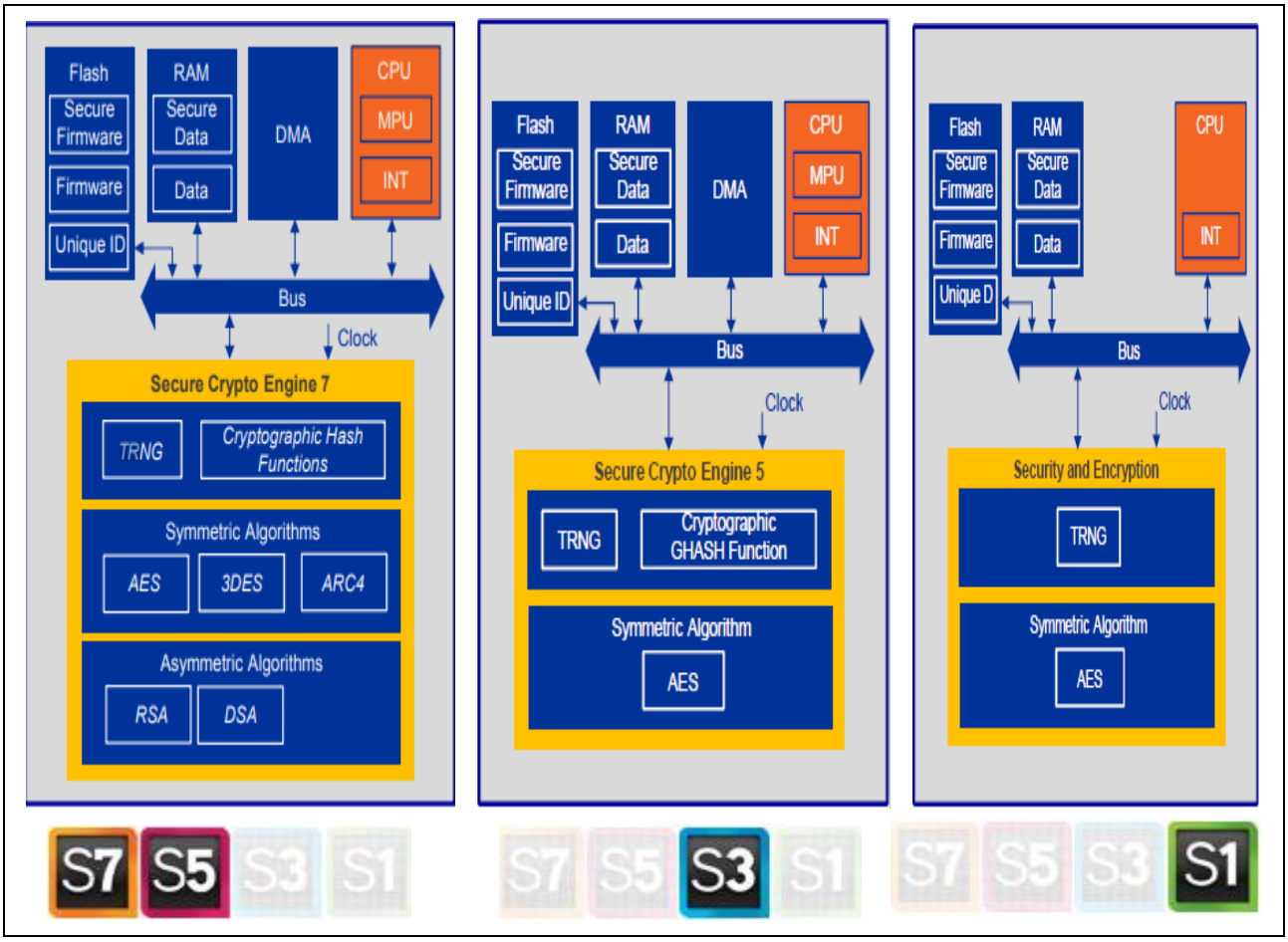


図 2　Synergy MCU で使用可能なセキュリティハードウェア周辺機能

Synergy デバイスが提供する SCE エンジンは、このサンプルアプリケーションプロジェクトの以下の分野で使用されます。

- ECC 鍵ペア（key pair）の生成 (公開鍵とラッピングされた秘密鍵：public key and wrapped private key)
- ECC 秘密鍵を使用した、チャレンジ（challenge）文字列への署名

2.　Synergy MCU での鍵生成の概要（Overview of Key Generation in Synergy MCU）s

2.1　鍵のラッピング（Key Wrapping）

SCE ハードウェアモジュールを使用して Synergy MCU 内部で生成したデバイス鍵（device key）は、平文（plain text、プレーンテキスト）またはラッピング形式（wrapped）のどちらかにすることができます。

平文鍵

平文とは、暗号化されていない、言い換えると保護されていない形式の情報またはデータを意味します。平文は、特別な処理を必要とせず、人間または機械が読み取ることができます。平文は他のシステムでも使用できますが、秘密鍵を平文形式で作成した場合、保護する必要があります。

ラッピング鍵

ラッピング鍵（wrapped key）とは、MCU のユニーク IDを関与させる方法を使用して、SCE が既に暗号化した鍵です。この鍵のラッピング解除 (暗号化解除) を実行するには MCU のユニークID が必要であり、この鍵をラッピング解除できるのは、この鍵をラッピングしたのと同じ MCU のみです。ラッピング鍵を使用できるのはその鍵を生成した Synergy MCU のみであり、その MCU の外部では使用できないため、Synergy MCU でラッピングした鍵はセキュアであると考えられます。。

ラッピング鍵には、以下の利点があります。

- ラッピング鍵を使用できるのは、その鍵を生成した Synergy MCU のみです。
- その鍵を他の Synergy MCU に移動することはできません。他の Synergy デバイスに移動した場合、ラッピング鍵から元の鍵を復元することはできません。

2.2 デバイス内での鍵生成（Key Generation in the Device）

SCE モジュールを使用して、Synergy MCU 内部でデバイス特有の鍵を生成するのは、一般的な使用方法です。Synergy ソフトウェアパッケージ (SSP) を使用してデバイス鍵を生成するには、SF Crypto Key (暗号化鍵) フレームワークモジュールを使用します。

SF_CRYPTO_KEY フレームワークは高レベル API を提供し、SF_CRYPTO_KEY SSP モジュールという形で実装しています。このSF_CRYPTO_KEY フレームワークは、セキュア暗号化エンジン (SCE) HAL モジュールによって暗号化鍵生成サービスを提供します。続いて、このモジュールはデバイス上の SCE IP を実行します。

SF CRYPTO KEY フレームワークモジュールの機能

SCE7 ハードウェアを活用する SF_CRYPTO_KEY モジュールを使用して、以下のタイプの鍵を生成できます。

- RSA 2048 ビット、1024 ビット平文/1024 ビット生鍵（raw key）の標準形式および中国剰余定理（Chinese Remainder Theorem：CRT）形式
- RSA 2048 ビット、1024 ビット標準形式でラッピングした秘密鍵 (公開鍵は平文)
- ECB、CBC、CTR、GCM の各チェーンモデル向けに AES 128 ビット、192 ビット、256 ビットでラッピングした鍵
- XTS チェーンモデル向けに AES 128 ビットおよび 256 ビットでラッピングした鍵
- ECC 192 ビット、256 ビットの平文/生の公開鍵およびラッピングした秘密鍵

このアプリケーションでは、暗号化 HAL モジュールを使用してラッピング ECC 鍵を生成しています。SF_CRYPTO_KEY フレームワークの代わりに暗号化 HAL モジュールを使用する理由は、Renesas が提供する Secure Boot Manager (セキュアブートマネージャ) など、RTOS を使用しない環境でデバイス ID アプリケーションを使用できるようにするためです。

2.3 セキュアインフラストラクチャからの鍵インジェクション（Key Injection from Secure Infrastructure）

鍵インジェクション（Key Injection）とは、セキュアな環境で MCU の外部で鍵を生成し、次にその鍵を MCU に導入する（inject）使用状況を意図したセキュリティ機能です。一般的にRSA 鍵を生成する場合、PC ツールを使用してセキュアな環境で MCU デバイスの外部で生成する状況に比べて、MCU 内部で生成する方が多くの時間を要します。

アプリケーションにハードウェアベースのユニーク ID が必須でない場合や、顧客特有の鍵をセキュアに導入することが必要な場合、鍵インジェクションを利用できます。このプロセスは、個別のアプリケーションプロジェクトで対応する予定です。

3. デバイス ID 設計の概要（Device Identity Design Overview）

この章では、Synergy のハードウェア機能とソフトウェア機能を組み合わせて、デバイスごとに一意のデバイス ID を作成するための方法を説明します。

鍵の生成

デバイス ID 作成の最初のステップは、鍵を生成することです。鍵は Synergy MCU 内部で生成すること、または安全な環境で MCU 外部で生成した後 Synergy デバイスに導入（injected）することができます。どちらの方法を採用する場合でも、アプローチごとに利点と欠点があります。ユーザの使用事例に基づいて、決定を下す必要があります。

認証局 (CA)

デバイス鍵を生成または導入した後、デジタル証明書を発行する存在（entity）が必要になります。CA として使用できるのは、クラウド内に位置するパブリック CA またはプライベート CA、あるいは社内（on-premises）CA (ローカル CA) です。社内 CA は通常、1 台のセキュアサーバ上でホストされています。

デバイス ID のセキュア確保

デバイス ID を作成し、Synergy デバイスにプログラムした (書き込んだ) 後、ID の盗難または破損を防止するために、ID をセキュアに格納する必要があります。この作業は、Synergy MCU が搭載しているセキュリティ MPU と FAW 機能を使用して実行できます。これらの機能は、内部コードフラッシュの一部を、セキュアなコード/データ領域として構成します。セキュアコード領域は、セキュアコード領域での動作のみを承認された API 関数を格納します。セキュアデータ領域は、デバイス証明書など鍵の情報を保持します。Synergy MCU 上で動作している、セキュアでないコードはいずれも、このセクション (領域) のアクセスや変更を実施できません。

これらの領域が変更される事態を防止するために、セキュアな環境 (プログラミングセンター) から送り出す前に、FAW 機能を活用しワンタイムプログラマブルな FPSR ビットを使用することにより、セキュリティ MPU の設定をロックします。

4. デバイス ID アプリケーションサンプル（Device Identity Application Example）

4.1 概要（Overview）

このサンプルアプリケーションプロジェクトは、MCU内蔵の Synergy SCE モジュールを使用して、Renesas Synergy™ デバイス ID アプリケーションを示します。デモの目的で、このアプリケーションは Windows PC 上で動作しているローカル認証局 (CA) を使用し、署名用の鍵（signing key）と、デバイス証明書に署名する目的で使用するルート CA 証明書（root CA certificate）を生成します。このプロジェクトは後ほど、サードパーティ CA を使用してデバイス証明書を生成する仕様にアップグレードする予定です。PK-S5D9/AE-Cloud2 キットと、Windows PC 上で動作するホストコンソールアプリケーションの間の主要な通信インタフェースとして、USB-CDC を使用します。

4.2 ソフトウェアアーキテクチャの概要（Software Architecture Overview）

以下の図に、Synergy デバイス ID アプリケーションプロジェクトのソフトウェアアーキテクチャ全体を示します。

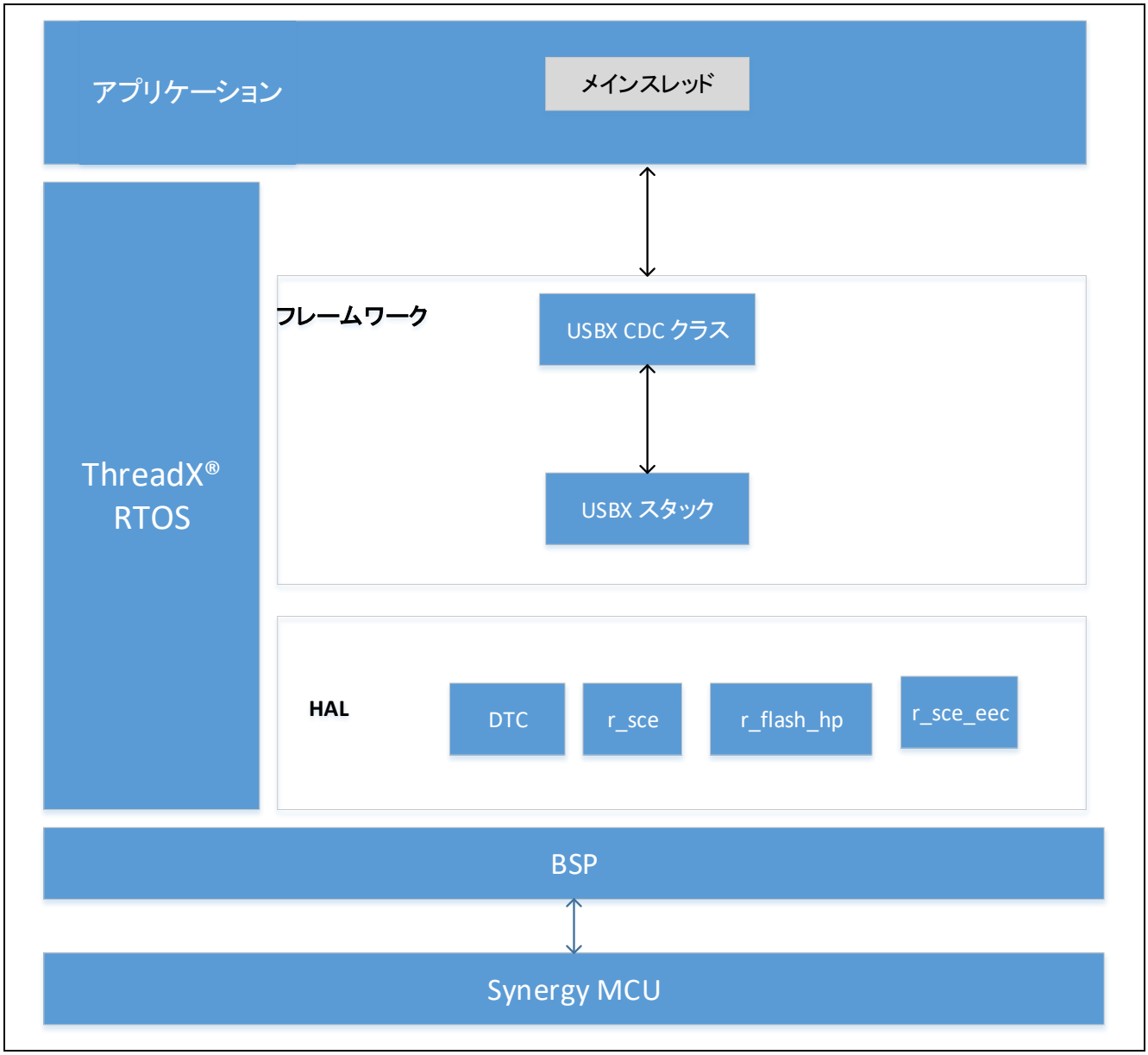


図 3　Synergy デバイス ID アプリケーションのソフトウェアアーキテクチャ

このアプリケーションの主なソフトウェアコンポーネントは以下のとおりです。

- USBX CDC デバイスフレームワーク（USBX CDC Device framework）
- r_sce 暗号化ドライバ（r_sce crypto driver）
- r_flash_hp フラッシュドライバ（r_flash_hp flash driver）

このアプリケーションは、以下のスレッドで形成されています。

- メインスレッド（Main Thread）

メインスレッド（Main Thread）

これはメイン制御スレッドであり、以下の機能を処理します。
1. USBX CDC デバイスフレームワークを使用した、USB データの受信/送信。
2. コマンドをデコードし、適切なコマンドハンドラ関数を呼び出します。続いて、その関数はそれに対応するコマンド機能を処理します。

メインスレッドは、以下のコマンドを処理します。
- WRAPPER_KEY_REQUEST
- WRAPPED_KEY_CERT_PROGRAM
- WRAPPED_KEY_CHALLENGE_RESP

WRAPPER_KEY_REQUEST

このコマンドを処理するのは、以下の API 関数です。`handleHrkKeyCreation ()`

この関数は、SSP 暗号化モジュールを使用して鍵生成を処理します。このアプリケーションは、ECC 鍵ペアの生成をサポートしています。このアプリケーションが生成する公開鍵と秘密鍵のペアはラッピングされます。鍵ペアを生成した時点で、デバイス証明書の用途で使用する目的で、公開鍵をホストアプリケーションに送信します。

ラッピング秘密鍵は内部でデータフラッシュに保存され、後でチャレンジ応答（challenge response）に署名する際にこの秘密鍵を使用します。

WRAPPED_KEY_CERT_PROGRAM

このコマンドを処理するのは、以下の API 関数です。`handleHrkCertProgram ()`

この関数は、ホストアプリケーションから受け取ったデバイス証明書（dvice certificate）を内部コードフラッシュのセキュア領域にプログラムする (書き込む) 処理を実施します。

WRAPPED_KEY_CHALLENGE_RESP

このコマンドを処理するのは、以下の API 関数です。`handleHrkCertChallengeResp ()`

このチャレンジ要求（challenge request）の意図は、認証しようとする公開鍵に対応するデバイス秘密鍵に関して、ターゲットがその秘密鍵の所有者（ownership）であることを証明できるようにすることです。

この関数は、ホストアプリケーションから送信されたチャレンジ応答要求（challenge response request）を処理します。要求を受け取った時点で、この関数は WRAPPER_KEY_REQUEST コマンドの一環として生成された秘密鍵を使用し、送信された要求の一部である文字列に署名します。検証の目的で、署名済みの文字列をホストアプリケーションに送り返します。ホストアプリケーションは署名済みの文字列を受け取った時点で、デバイス証明書から抽出したデバイス公開鍵を使用してその署名を検証します。署名の検証に成功した場合、ホストアプリケーションはデバイス証明書をデバイスに送信します。その証明書は、セキュリティ MPU と FAW を使用してセキュアに保存されます。

4.3　動作フローの全体像（Operational Overflow）

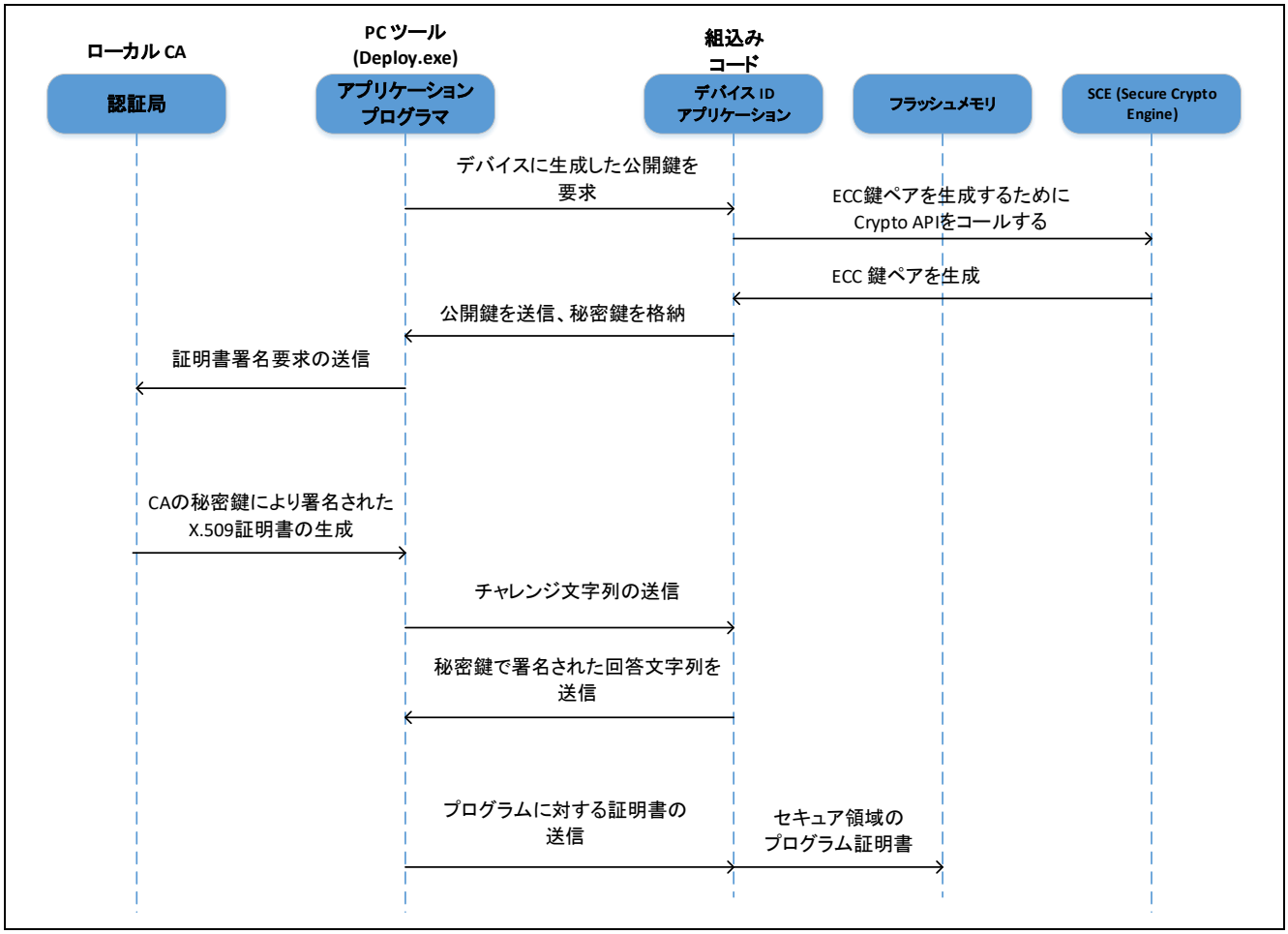


図 4　動作フローの全体像

このアプリケーションプロジェクトは、2 個のソフトウェアプロジェクトで形成されています。

- PK-S5D9/AE-Cloud2 キット上で動作する組み込みプロジェクト
- Windows (7/10) 上で動作するホストアプリケーション

PK-S5D9/AE-Cloud2 キットの電源をオンにした時点で、組み込みソフトウェアはプラットフォームおよび基礎となる USB CDC スタックを初期化します。このスタックは、ホストアプリケーションとの通信を行う目的で Windows PC で動作しています。初期化が終了した時点で、組み込みソフトウェアは USB デバイス接続イベントを待ちます。ユーザが USB ケーブルを使用してキットを Windows PC に接続した時点で、USB デバイス検出イベントが処理され、USB CDC インスタンスが作成されます。この段階で、組み込みソフトウェアはホストアプリケーションからのコマンドを待ちます。

ユーザが Windows PC でホストユーティリティを実行すると、そのユーティリティは使用可能な COM ポートをスキャンし、PK-S5D9/AE-Cloud2 キットの接続先となるポートを開きます。COM ポートを開くことに成功した場合、ユーティリティは署名用の鍵とルート CA 証明書を生成します。これらは後でデバイス証明書に署名する際に使用します。この時点で、ホストアプリケーションは **WRAPPER_KEY_REQUEST** コマンドを生成し、そのコマンドをキットに送信します。この要求を受信した時点で、ターゲットキット上で動作している組み込みコードはデバイス鍵のペアを生成し、公開鍵をホストアプリケーションに送信します。ホストアプリケーション側で、デバイスから公開鍵を受け取り、(CA の署名用鍵を使用して署名された) デバイス証明書を生成します。

デバイス証明書を発行する前に、デバイスが適切な秘密鍵（private key）を所有していることを証明できるように、ホストアプリケーションはデバイスに対してチャレンジ文字列を発行します。組み込みソフトウェアはチャレンジ文字列を受け取った時点で、自らの秘密鍵を使用してその文字列に署名し、署名済みの文字列をホストアプリケーションに送り返します。ホストアプリケーションはデバイス公開鍵を使用して署名を検証します。検証に成功した場合、デバイス証明書を PK-S5D9/AE-Cloud2 キットに送信し、キットは Synergy MCU のセキュリティ MPU と FAW を使用してその証明書をセキュアに保存します。

4.4　デバイス ID のセキュアな保存（Securely Storing Device Identity）

このアプリケーションの一環として、以下の 2 個のユニークなデバイス ID が作成されます。

- ラッピング ECC 秘密鍵（Wrapped ECC private key）
- デバイス証明書（Device certificate）

これら 2 個のデバイス ID は、アクセスまたは変更されることを防止するために、セキュリティ MPU と FAW を使用して、Synergy MCU 内部にセキュアに保存する必要があります。このアプリケーションの一環として生成された秘密鍵は既にラッピングされているため、デバイス鍵をセキュアに保存するステップを省略できます。ただし、特定の状況では、デバイス内で鍵が誤用される事態を防止するために、ユーザはラッピング鍵をセキュアな場所に保存する方針を選択します。デバイス証明書を保存する場合も、同じステップを使用できます。

以下の図に、現行のデバイス ID アプリケーションプロジェクトのメモリマップ（memory map）を示します。

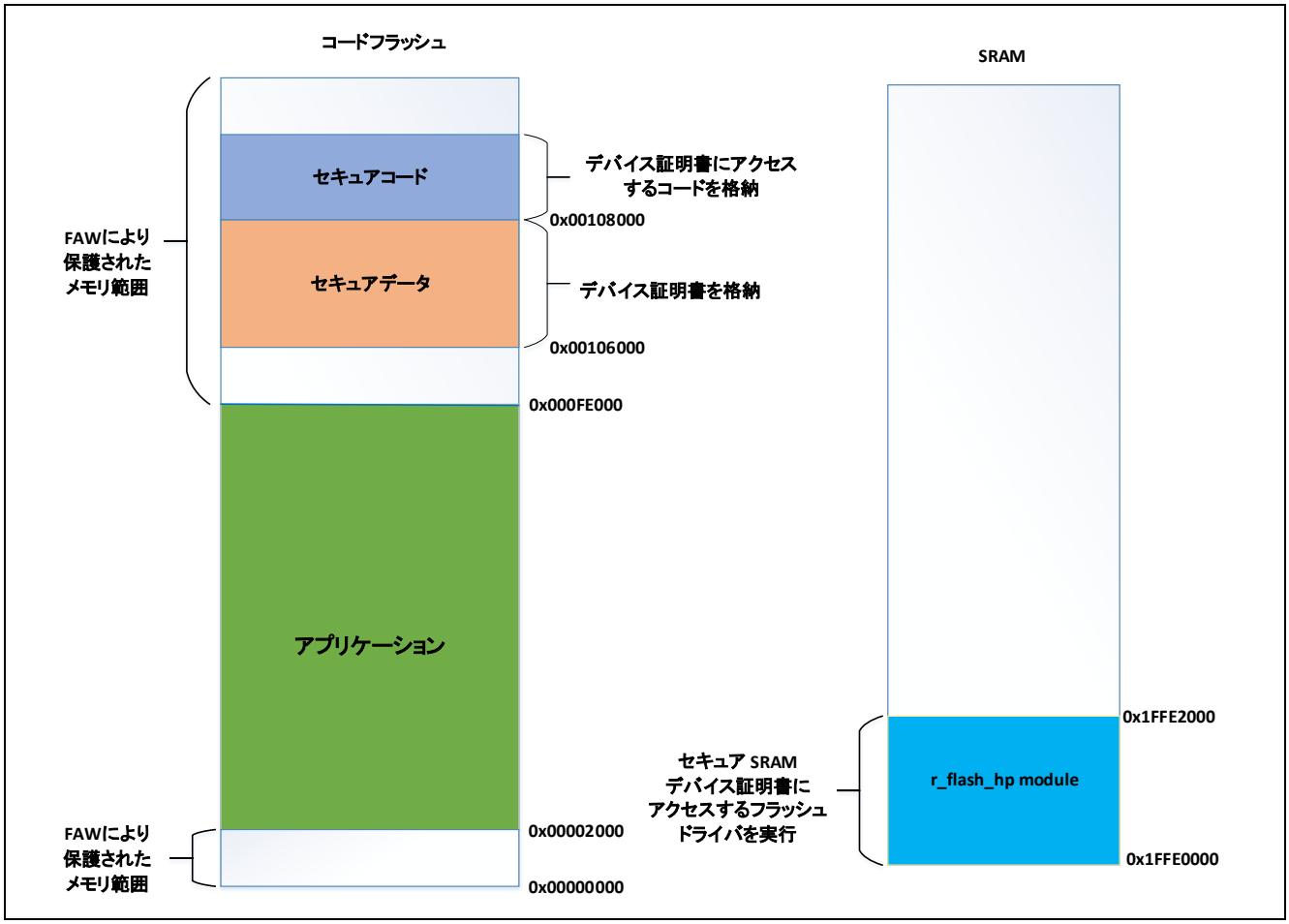


図 5　アプリケーションプロジェクトが使用しているメモリマップ

図 5 の左側に、このアプリケーションプロジェクトが使用しているコードフラシュのメモリレイアウトを示します。緑色はアプリケーションの配置先領域、オレンジ色はデバイス証明書保存用の予約領域、青色はセキュアデータ領域にアクセスするセキュアコード用の予約領域を表します。

上記のメモリマップに示すように、0x0 ～ 0x2000 のアドレス範囲はセキュリティ MPU の設定を保持しています。また、0xFE000 以降 (アドレス境界で整列されているブロック) はセキュアデータとセキュアコードを保持しており、ユーザアプリケーションによる消去から保護する必要があります。この目的で、0x2000 ～ 0xFE000 のメモリ範囲をアクセスウィンドウとするように FAW を構成してあります。FAW の開始アドレス（start address）を 0x2000 に、また終了アドレス（end address）を 0xFE000 に設定する方法で、これらのアドレス範囲の外側にあるセクションは変更 (改ざん) から保護されます。Synergy_Device_Identity_Solution\embedded\common\src\framedProtocolTarget.c ファイル内で、この FAW 設定を実装している flash_FAW_Set() API を参照してください。

図 5 の右側に、このアプリケーションプロジェクト内の SRAM 領域メモリマップを示します。SSP フラッシュドライバモジュール (r_flash_hp) は、既にセキュア SRAM 領域として構成してある、SRAM の下端（bottom portion）にマップされています。これは、デバイス証明書をロードしながら、フラッシュドライバがセキュアデータ領域を消去およびプログラムする (書き込む) するために必要です。

これらのセキュアなデータ/コード SRAM 領域の割り当てとマッピングを行う方法は、サンプルアプリケーションプロジェクトに関連するリンカスクリプトを参照してください。また、図 6 に示す Synergy コンフィギュレータの設定も参照してください。内部コードフラッシュ内にある特定の領域をセキュア領域として構成する方法の詳細に関して、アプリケーションプロジェクトはこの設定を使用しています。コードフラッシュ/SRAM 内にある特定の領域をセキュア領域として割り当てる場合、ユーザは設計時にこのサンプルソフトを参考にできます。そのセキュア領域に、デバイス ID のほか、保護された IP またはサードパーティからライセンスを付与された FW (ファームウェア) を保存できます。

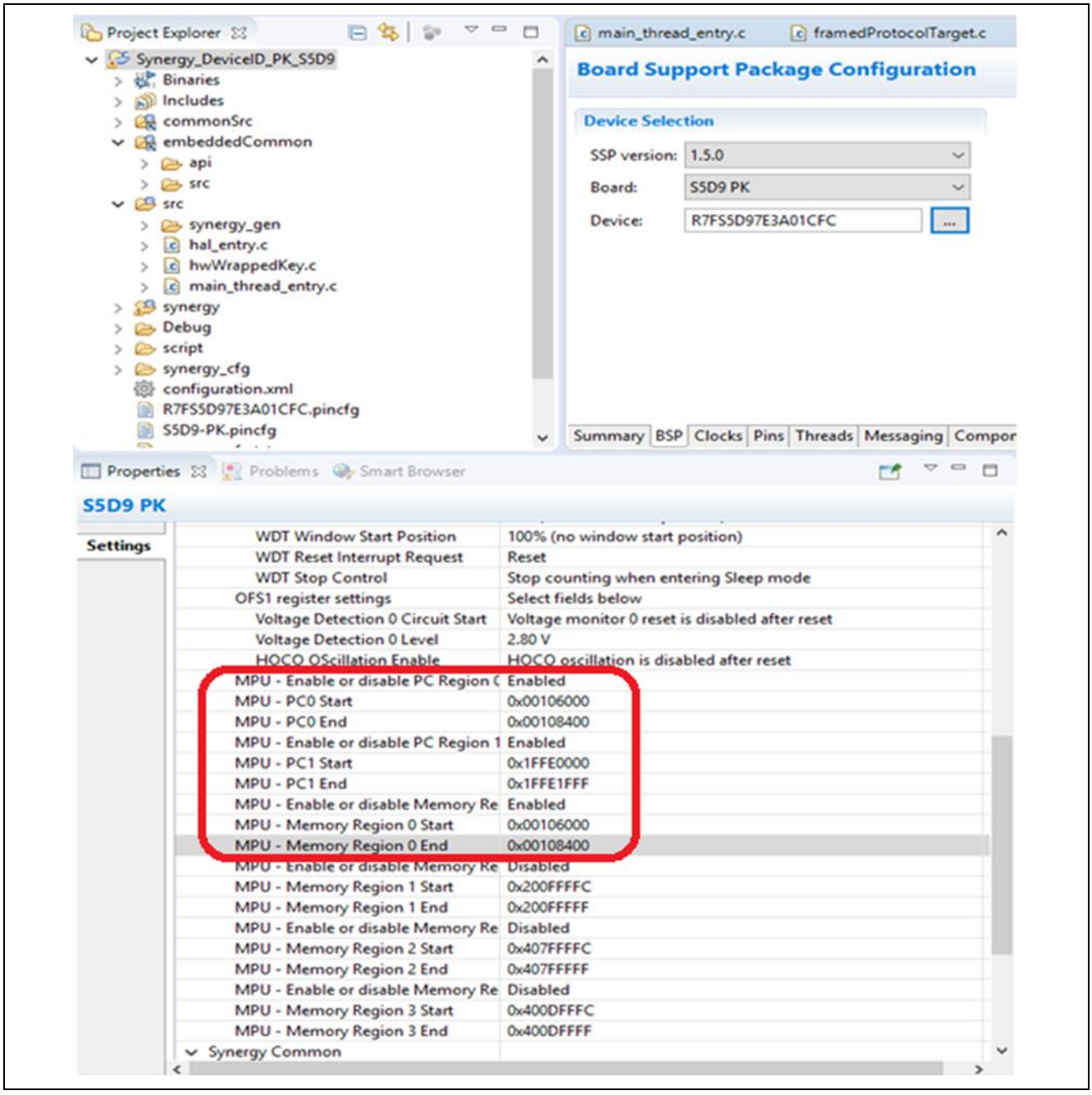


図 6　e² studio ISDE コンフィギュレータを使用したセキュリティ MPU の設定

5. デバイス ID アプリケーションサンプルの実行（Running the Device Identity Application Example）

5.1 プロジェクトのインポート、ビルド、および実行（Importing, Building, and Running the Project）

組み込みプロジェクトは、**Synergy_Device_Identity_Solution\embedded** フォルダに配置されています。以下の説明は、e² studio のワークスペースにこれらのプロジェクトをインポートする方法を示します。

e² studio ISDE で、**[File]** (ファイル) -> **[Import…]** (インポート...) -> **[Existing Projects into Workspace]** (既存のプロジェクトをワークスペースに) を選択し、**[Select Root Directory]** (ルートディレクトリの選択) セクションで上記のフォルダを参照します。

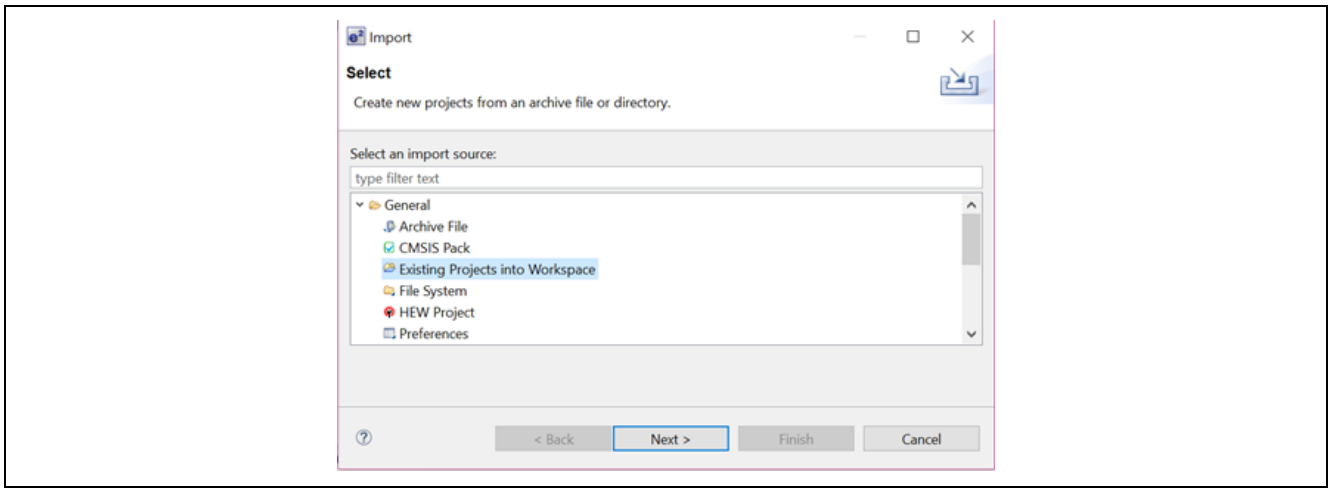


図 7　プロジェクトのインポート

以下の図のように、ネストされたすべてのプロジェクトをインポートするように（Search for nested projects のチェックボックスをオンにする）選択を行います。**[Copy projects into workspace] (ワークスペースにプロジェクトをコピーする) のチェックボックスはオンにしないでください。**

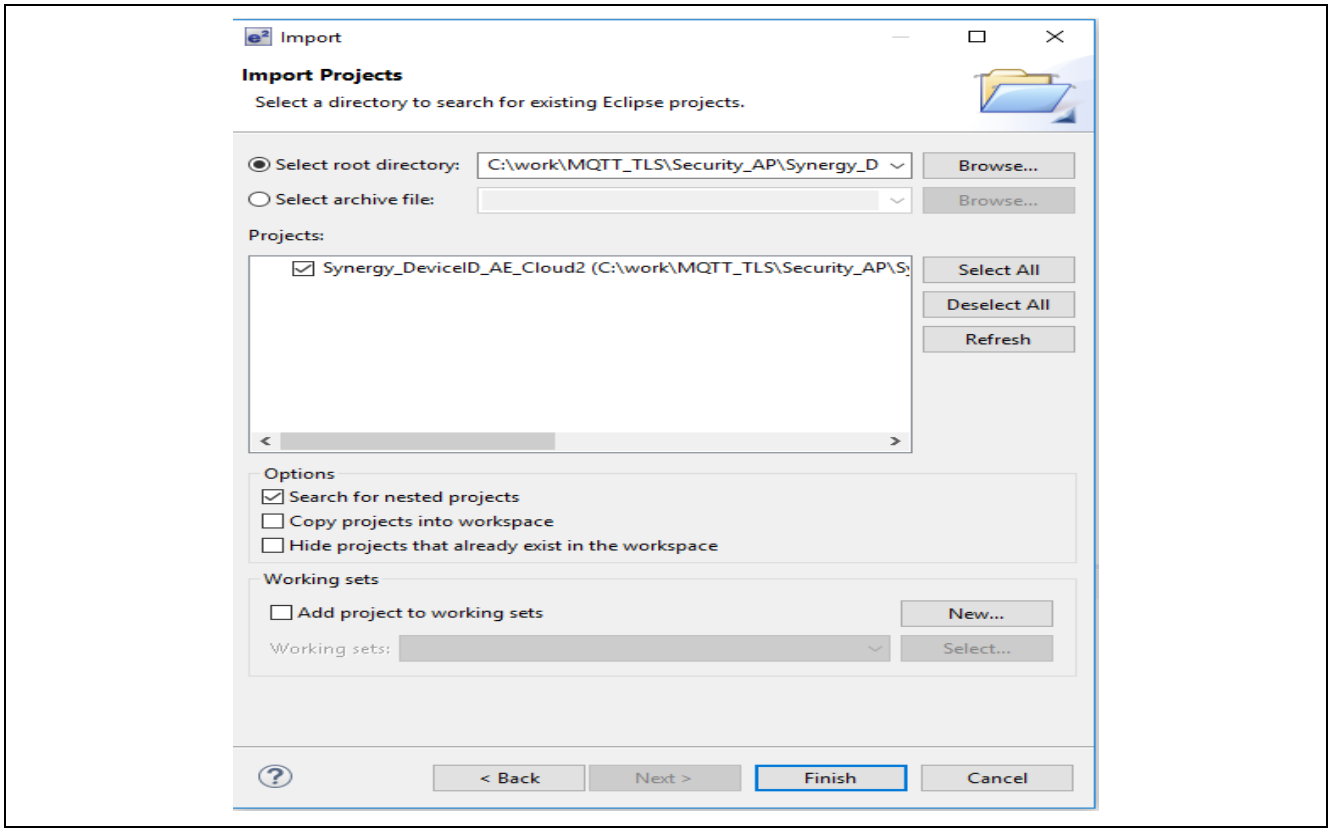


図 8　ネストされたすべてのプロジェクトをインポートするための選択

5.2　ボードの電源投入（Powering up the Board）

電源をボードに接続するには、以下の説明に従ってください。

1. 付属している USB ケーブルの Micro USB コネクタを、PK-S5D9 ボードの J19 コネクタ (DEBUG_USB) または AE-Cloud2 ボードの J6 コネクタ (DEBUG_USB) に接続します。

注記：　このキットは、SEGGER J-Link® On-board (OB) を搭載しています。J-Link は、PK-S5D9/AE-Cloud2 ボードのフルデバッグ機能とプログラミング機能を実現します。

2. USB ケーブルのもう一方のコネクタを、開発用の Windows PCの USB ポートに接続します。

5.3　ITM の printf を使用したデバッグ（Debugging using ITM printf）

このパッケージに付属の組み込みアプリケーションプロジェクトは、多数の printf() 文を使用してシステムに関する情報を出力します。e² studio 内の Renesas Debug Console (Renesas デバッグコンソール) は、セミホストされた printf を使用しますが、この関数は Synergy プロセッサのリアルタイム実行を阻害します。それに対し、このプロジェクトは SWO 経由で printf を使用するようにセットアップしています。SWO の printf による出力は、[Live Trace Console] (ライブトレースコンソール) ウィンドウを使用して、e² studio 内で直接キャプチャおよび表示することができます。ただし、printf の出力をキャプチャする場合、[Live Trace Console] (ライブトレースコンソール) の性能は不十分です。SWO からキャプチャしたすべてのパケット (すなわち、すべての文字) に対してタイムスタンプを割り当て、プロジェクト内のトレースファイルに保存しているからです。代わりに、JLink ソフトウェアツールに付属の SWO Viewer (SWO ビューア) を使用すれば、大幅に高い性能で SWO 出力を表示することができます。

5.4　デモの確認（Verifying the Demonstration）

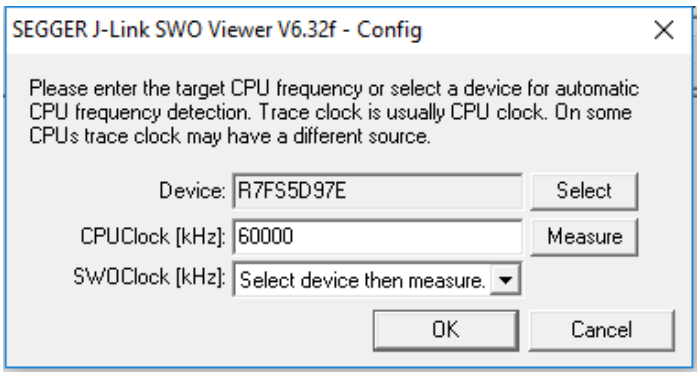
この段階では、5.1 章で説明した手順を既に実行し、アプリケーションプロジェクトのインポート、ビルド、ターゲットキットへのロードを実施済であることを想定しています。これらの手順を実施していない場合、5.1 章に戻り、記載されているステップを実施した後、この章のこれ以降の操作を続けてください。

ここで、e² studio プロジェクトのデバッグセッションを開始し、main() の直前まで実行します。

```
126
127     int main(void)
128     {
129         __disable_irq ();
130         tx_kernel_enter ();
131
132         return 0;
133     }
134
```

Config (構成) ダイアログボックスで、デバイスとして **[R7FS5D97E]** を選択し、**[Measure]** (測定) ボタンをクリックした後、CPU クロック (kHz) を「60000」に変更します

注記：　このアプリケーションで使用している JLINK ツールはv6.32f です。v6.32f またはそれ以降の JLINK ツールを使用することを推奨します。



[OK] をクリックします。データソースとしてポート 0 が選択されていることを確認します。

プロジェクトの実行を再開します。SWO の printf からの出力は、SWO ビューアで表示できます。

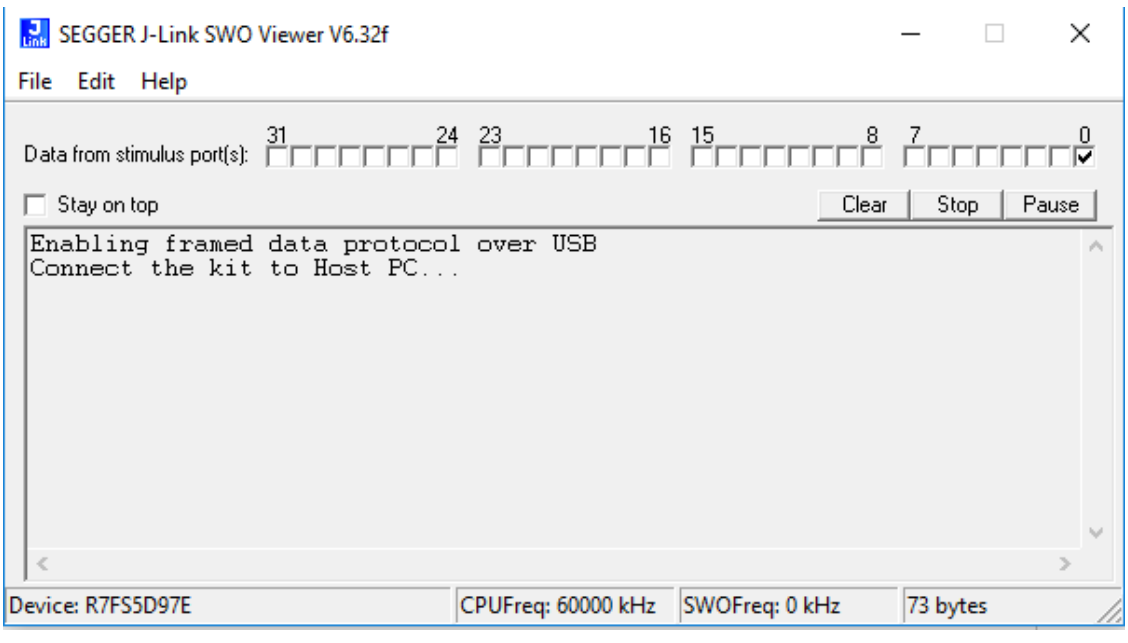


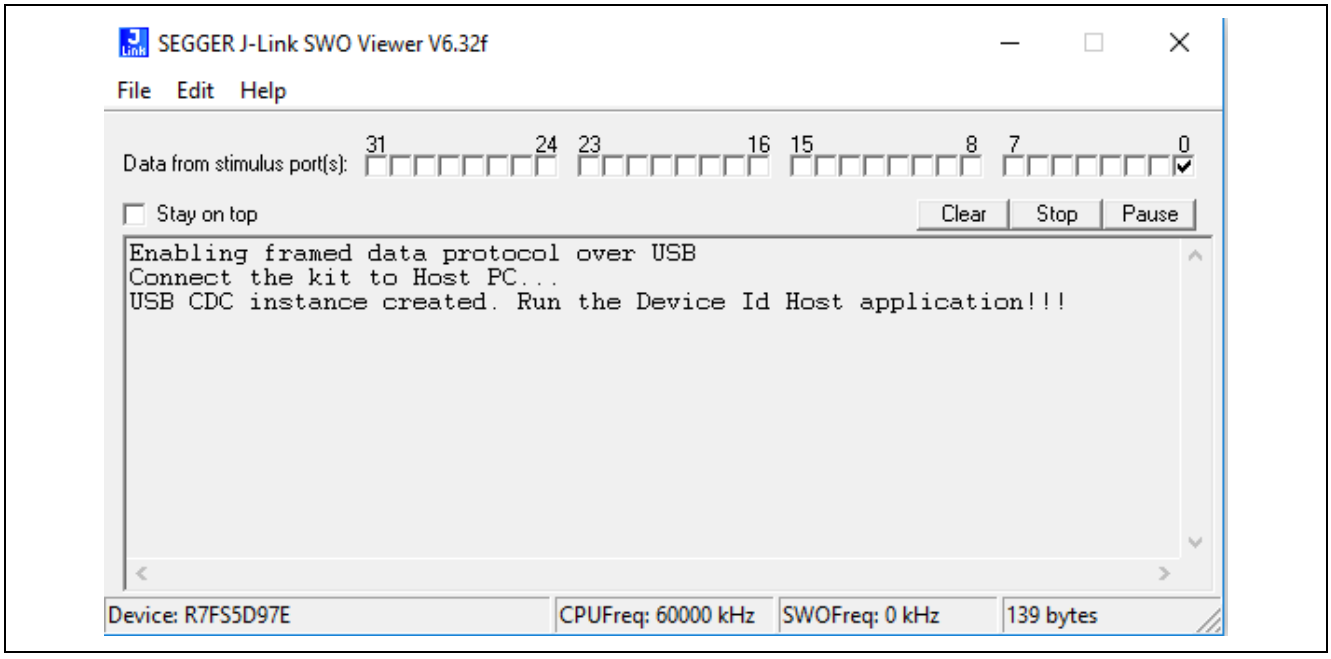
図 9　SWO Viewer (SWO ビューア) 内の開始メッセージ

ここで、Micro USB ケーブルの一方のコネクタをターゲットキットの USB デバイスポートに接続し、もう一方のコネクタを開発用の Windows PC に接続します。以下のリンクに掲載されている説明に従い、Windows (7/10) PC 用の Synergy USB CDC 通信ドライバをロードします。

https://www.renesas.com/en-us/products/synergy/software/add-ons/usb-cdc-drivers.html

デバイスマネージャ内で、ターゲットキットは **[Synergy USB Communication Port]** (Synergy USB 通信ポート) という名称で表示されています。デバイスマネージャで表示されている、ターゲットキットに対応する COM ポートの番号を書きとめます。

SWO ビューアで、以下の図のようなメッセージがコンソールに表示されます。USB CDC インスタンスが作成されたこと、またホストアプリケーションと通信する準備ができていることを示しています。



ここで、Windows PC でホストアプリケーションを実行します。アプリケーションを実行するために、Windows PC でコマンドウィンドウを開き、このアプリケーションプロジェクトの配置先フォルダに移動します。deploy.exe ファイルは、**Synergy_Device_Identity_Solution\pc\apps\deploy\Release** ディレクトリの下に配置されています。

このホストアプリケーションを実行するために、以下の図に示すように、コマンドウィンドウで以下のコマンドを入力します。

deploy.exe connect <**COM port number**>

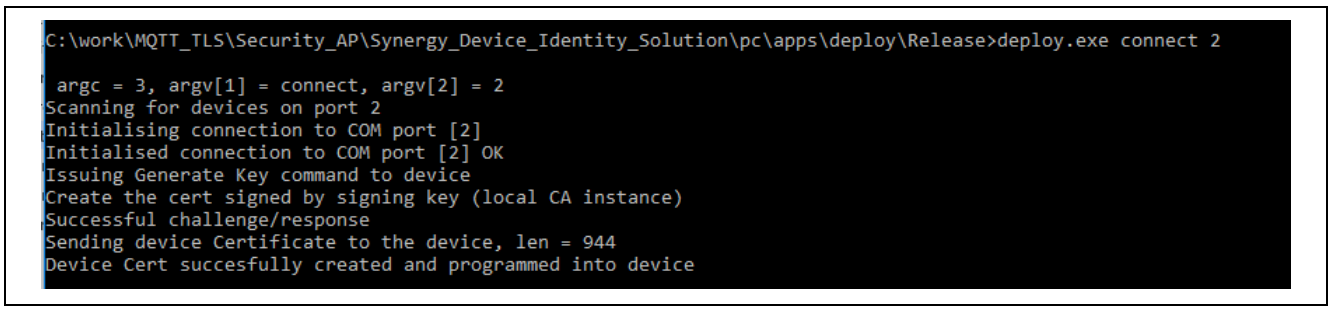


図 10　ホストアプリケーションのコンソールメッセージ

上図のようにホストアプリケーションから出力されたコンソールメッセージが、また以下の図のように SWO ビューア内のコンソールメッセージで状況が通知されます。

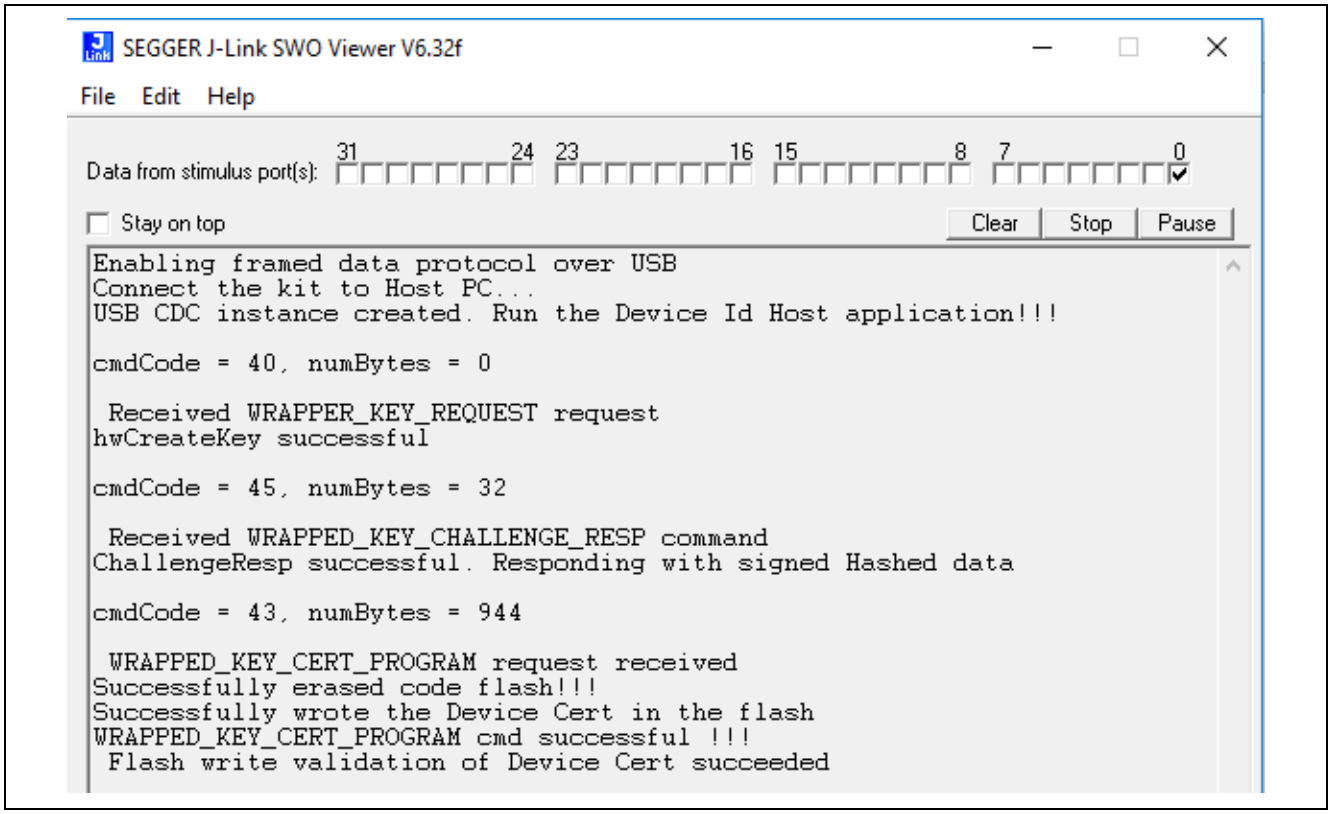


図 11　ターゲットキットに関するコンソールメッセージ

この段階で、USB-CDC 通信インタフェースを経由して、ホストアプリケーションはターゲットとの通信を開始します。4.3 章で示したように、ユーザアプリケーションは以下のタスクを実行します。

1. ユーザが指定した USB COM ポートをスキャンします。見つかった場合、シリアル接続を開始します。
2. シリアル接続に成功すると、ターゲットキットに対して Generate Key Command (鍵の生成コマンド) を発行します。
3. デバイス側で、SCE 暗号化モジュールを使用して ECC 鍵ペアを生成します。公開鍵をホストアプリケーションに送り返します。
4. ホストアプリケーションは、ルート CA 証明書を作成するほか、後の段階でデバイス証明書への署名に使用する署名用の鍵も作成します。
5. チャレンジ/応答（response）の文字列を生成し、ターゲットキットに送信します。
6. チャレンジ/応答に成功した場合、ホストアプリケーションはデバイス証明書に署名し、デバイスに送信します。
7. デバイス証明書は内部コードフラッシュ内にセキュアに保存され、セキュリティ MPU と FAW によって保護されます。

6. デバイス ID 識別アプリケーションの次のステップ（Device Identity Application Next Steps）

この章では、デバイス ID アプリケーションの第 2 段階で計画している機能の高レベルの概要を紹介します。

サードパーティ CA のサポート

フェーズ 1 で、このアプリケーションプロジェクトは Windows PC 上で動作しているローカル CA を使用して、自己署名のルート証明書を作成し、ローカル CA の署名用の鍵を使用して署名したデバイス証明書を生成します。フェーズ 2 では、サードパーティ CA を使用して、デバイス証明書が生成できるように、このアプリケーションプロジェクトのアップデートを計画しています。

この機能の一部として、別の CA にシームレスにプラグインすることができる汎用インタフェースを作成することを計画しています。

クラウドプロバイダ (AWS/Google/Azure) 用インタフェースのサポート
AWS/Google Cloud/Azure 用の Synergy Cloud 接続ソリューションは、現在のフェーズではクラウドの CA が生成したデバイス鍵とデバイス証明書を使用しています。これらのプロジェクトをアップデートし、デバイス証明書を生成する際に、Renesas がサポートしているサードパーティ CA を使用する予定です。

SBM 用インタフェースのサポート
標準的なサードパーティ CA 向けのサポートを追加した後、この機能を既存の Synergy Secure Boot Manager (セキュアブートマネージャ、SBM) に統合する予定です。

鍵インジェクションのサポート
鍵インジェクション（Key Injection）とは、セキュアな環境で MCU の外部で鍵を生成し、次にその鍵を MCU に導入する（inject）ためのセキュリティ機能です。このサポートについては、別途、文書化する予定です。

7. 参考資料（References）

- SSP ユーザーズマニュアル (renesassynergy.com/ssp)
- Secure Boot Manager (セキュアブートマネージャ：SBM) アプリケーションプロジェクト (https://www.renesas.com/us/en/software/D6002619.html)
  - Secure Data-at-rest (保存データ（データアットレスト）のセキュリティ) アプリケーション

8. 既知の問題と制限（Known Issues and Limitations）

このホストアプリケーションのテストは、Windows (7/10) PC でのみ実施しました。

9. 付録（Appendix）

9.1 用語集（Glossary）

用語	意味
認証局 (Certificate Authority：CA)	ポリシーベースのルールに従ってデジタル証明書を発行する存もの（entity）です。CA として使用できるのは、クラウド内に位置するパブリック CA またはプライベート CA、あるいは社内（on-premises）CA (ローカル CA) です。社内 CA は通常、1 台のセキュア機器上でホストされています。
デバイス証明書 (Device Certificate)	個別の Synergy デバイスを一意に 識別する証明書です。この証明書はデジタル署名されており、証明書が既知の出所から到着したものであることや、変更 (改ざん) されていないこと、そのデバイスが信頼されている (信頼されたアプリケーションファームウェアとともに、Secure Boot Manager (セキュアブートマネージャ) がインストールされている) ことを伝えます。
信頼の基点 (Root of Trust)	信頼の基点とは、非常に信頼性が高いハードウェア、ファームウェア、ソフトウェアの各コンポーネントの組み合わせであり、これらは特定のクリティカルなセキュリティ機能を実行します。 (https://csrc.nist.gov/projects/hardware-roots-of-trust)
SCE	セキュア暗号化エンジン（Secure Crypto Engine）の略称です。MCU 内にあるモジュールの 1 つであり、効率的で低消費電力な暗号化アクセラレーション TRNG (True Random Number Generation 真の乱数生成) を実現するほか、分離用暗号化鍵（isolating cryptographic key）の作成と分離（isolation）を実施します。
PKI	公開鍵基盤（Public Key Infrastructure）の略称です。デジタル証明書の作成、管理、配布、使用、保存、取り消しを行うために必要とされる、一連の役割、ポリシー、手続きのことです。公開鍵暗号化を通じてセキュア ID を管理する目的で、通常は PKI を使用します。
鍵ペア (Key Pair)	非対称鍵（asymmetric key）は、公開鍵と秘密鍵のペアという形で生成されます。秘密鍵は特定の団体または個人のみが秘密のまま保持するもので、その団体または個人の身元を伝える目的で使用できます。公開鍵は自由に配布することができ、秘密鍵と一意に関連付けられています。
セキュアコード (Secure Code)	内部フラッシュのセキュア領域内に存在している 1 つまたはグループの関数です。セキュア領域は、MPU が定義し、動作させます。これらのセキュア関数は、

用語	意味
	セキュアデータ領域と、セキュアではないデータ領域の両方にアクセスできます。
セキュアではないコード (Non-Secure code)	内部フラッシュのセキュアではない領域内に存在している 1 つまたはグループの関数です。これらのセキュアではないコードは、セキュア領域にアクセスできません。これらのコードがアクセスできるのは、セキュアではない領域のみです。
HRK	隠されたルート鍵 (Hidden Root Key) の略称です。これは、Synergy MCU 内に格納された一意（unique）な鍵です。
SBM	Secure Boot Manager (セキュアブートマネージャ) の略称です。Synergy MCU 上でアプリケーションバイナリのダウンロード、ブート、更新をセキュアに実施するソフトウェア集合体を指します。
チャレンジ文字列 (Challenge String)	ホストアプリケーションがランダムに生成した文字列です。ホストアプリケーションはこの文字列を使用して、ターゲットが秘密鍵を所有していることを確認します。
ユニーク ID (Unique ID)	識別用の値（identification value）であり、個別の Synergy MCU にとって一意であるほか、Synergy MCU の内部に格納されています。SCE は鍵をラッピングするときにユニーク ID を使用します。
チャレンジ応答文字列 (Challenge Response String)	チャレンジ文字列に対する応答です。チャレンジ応答文字列とは、チャレンジデータに対する署名であり、受信側の秘密鍵を使用してチャレンジ文字列に署名する方法で作成します。

Web サイトおよびサポート

以下の URL で、Synergy プラットフォームの詳細の確認、関連するドキュメントのダウンロード、サポートの活用ができます。

Synergy ソフトウェア	renesassynergy.com/software
Synergy ソフトウェアパッケージ	renesassynergy.com/ssp
ソフトウェアアドオン	renesassynergy.com/addons
ソフトウェア用語集	renesassynergy.com/softwareglossary
開発ツール	renesassynergy.com/tools
Synergy ハードウェア	renesassynergy.com/hardware
マイクロコントローラ	renesassynergy.com/mcus
MCU 用語集	renesassynergy.com/mcuglossary
パラメトリック検索	renesassynergy.com/parametric
キット	renesassynergy.com/kits
Synergy ソリューション Gallery	renesassynergy.com/solutionsgallery
パートナープロジェクト	renesassynergy.com/partnerprojects
アプリケーションプロジェクト	renesassynergy.com/applicationprojects
セルフサービスサポートリソース:	
ドキュメント	renesassynergy.com/docs
ナレッジベース	renesassynergy.com/knowledgebase
フォーラム	renesassynergy.com/forum
トレーニング	renesassynergy.com/training
ビデオ	renesassynergy.com/videos
Web チケット	renesassynergy.com/support

改訂履歴

Rev.	発行日	説明	
		ページ	ポイント
1.00	2019.4.18	-	・初版 ・英語版（R11AN0348EU0100, Rev.1.00, 2018.10.24発行）を翻訳

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
   家電、工作機械、パーソナル機器、産業用ロボット等

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、
   金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：https://www.renesas.com/contact/