

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、組み込みシステムにおける保存データのセキュア化 (secure Data at Rest) 設計に関する検討事項を説明し、Synergy MCU のセキュリティ機能を使用して保存データのセキュア化ソリューションを実現する方法を紹介します。保存データのセキュア化とは、変更可能なデバイスまたは変更不可能なデバイスに存在する機密データ (sensitive data) を保護する目的で提供されている機能とサービスのことで、

Synergy MCU は、保存データのセキュア化が実現できるように、CPU とバスマスタ (bus master) を使用したデータの暗号化 (encryption)、認証方式 (authentication schemes)、読み取り/書き込みアクセス (read/write access) および 1 回のみ書き込みアクセス (write-once access) の保護を可能にしています。さらに、Synergy MCU は、非セキュアソフトウェアアクセスによる特定のセキュリティ関連 (security-related) 周辺装置 (peripheral) の制御を無効にするセキュリティ機能を備えています。

Renesas は、セキュアブートマネージャ (Secure Boot Manager、SBM) アプリケーションノートをリリースしました。このセキュアブートローダ (secure boot loader) が MCU デバイスにとって信頼の基点 (root of trust) の役割を果たします。SBM を使用する保存データのアプリケーションのガイドラインは本アプリケーションノートでは、示しません。

内部フラッシュアプリケーション向けのこのアプリケーションノートの使用方法例は、機密データの読み取り保護 (sensitive data read protection)、書き込み保護 (write protection)、読み取り/書き込み保護 (read/write protection)、Write-Once 保護 (write-once protection)、Write-Once と読み取り保護 (write once with read protection) に対応しています。内部 SRAM の使用方法に関しては、このアプリケーションノートは読み取り保護と書き込み保護の使用法例を示します。

このガイドに従って、開発中の製品設計に Synergy が提供する保存データのセキュア化の機能とソリューションを使用して、対象アプリケーションに合わせてセキュアコンポーネントを正しく設定した後、付属しているサンプルアプリケーションコードをひな型として参照してコードを作成できます。このガイドは、Synergy MCU のセキュリティ機能を効率的に使用する目的で、メモリアクセス機能をセットアップできるように、ステップごとの操作フローを掲載しています。ハードウェア機能の詳細と API に関する説明は、ハードウェアユーザーズマニュアルと『Synergy™ ソフトウェアパッケージ (SSP) ユーザーズマニュアル』に掲載されています (参考資料の章を参照)。

## 必須リソース (Required Resources)

### 開発ツールとソフトウェア

- e<sup>2</sup> studio ISDE v6.2.1 またはそれ以降 ([www.renesas.com/synergy/e2studio](http://www.renesas.com/synergy/e2studio))
- Visual Studio 2017 Community Version (<https://visualstudio.microsoft.com/downloads/>)
- Synergy ソフトウェアパッケージ (SSP) 1.5.3 またはそれ以降 ([www.renesas.com/synergy/ssp](http://www.renesas.com/synergy/ssp))
- Segger J-link® USB ドライバ (<https://www.segger.com/downloads/J-Link/>)
- Renesas Flash Programmer V3 (<https://www.renesas.com/jp/ja/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html>)

### ハードウェア

- Renesas Synergy PK-S5D9 キット ([www.renesas.com/synergy/pk-s5d9](http://www.renesas.com/synergy/pk-s5d9))
- Windows 7 または 10 と、Tera Term コンソールまたは類似のアプリケーションが動作している PC
- 2 本の Micro USB ケーブル

## 前提条件と対象読者 (Prerequisites and Intended Audience)

このアプリケーションノートは、Renesas e<sup>2</sup> 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、e<sup>2</sup> 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、S5D9 フラッシュメモリの構造と e<sup>2</sup> studio のリンクスクリプトのカスタマイズ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、Synergy MCU を使用した保存データのセキュア化ソリューション開発に従事するユーザです。

## 目次

1. セキュアデータの概要 (Secure Data Overview) .....	6
1.1 保存機密データのトポロジー (Sensitive Data at Rest Topology) .....	6
1.2 保存データのセキュリティ対策 (Data at Rest Security Measures) .....	6
1.2.1 データ暗号化 (Data Encryption) .....	6
1.2.2 データアクセス制御 (Data Access Control) .....	7
1.2.2.1 読み取り保護 (Read Protection) .....	7
1.2.2.2 書き込み保護 (Write Protection) .....	7
1.2.2.3 読み取り保護と書き込み保護 (Read/Write Protection) .....	7
1.2.2.4 Write-Once 保護 (Write-Once Protection) .....	7
1.2.2.5 Write-Once 保護と読み取り保護 (Write-Once and Read Protection) .....	8
1.3 保存データのセキュア化のリスクプロファイルと攻撃対象領域の分析 (Secure Data at Rest Risk Profile and Attack Surface Analysis) .....	8
1.4 セキュアブートローダの使用時または不使用時の保存データのセキュア化 (Secure Data at Rest with and without Secure Boot Loader) .....	8
2. 保存データのセキュア化に関連する Synergy MCU の機能 (Synergy MCU Features for Secure Data at Rest) .....	9
2.1 Synergy のセキュリティ要素の概要 (Overview of Synergy Security Elements) .....	9
2.2 セキュリティ MPU (Security MPU) .....	9
2.2.1 セキュアデータ領域 (Secure Data Regions) .....	10
2.2.1.1 セキュアプログラム (Secure Program) .....	12
2.2.1.2 セキュアアクセスの監視と保護 (Secure Access Monitor and Protection) .....	12
2.3 フラッシュアクセスウィンドウ (Flash Access Window, FAW) .....	13
2.3.1 コードフラッシュの書き込み保護を実現するためのセキュリティ MPU と FAW の使用方法 (Using the Security MPU and FAW for Code Flash Write Protection) .....	14
2.4 デバッグセキュリティに関する検討事項 (Debugging Security Considerations) .....	16
2.5 ARM MPU、バスマスタ MPU、バススレーブ MPU に関する注記 (Notes on ARM MPU, Bus Master MPU, Bus Slave MPU) .....	16
2.6 その他のセキュリティ要素 (Other Security Elements) .....	16
2.6.1 セキュア暗号化エンジン (SCE) (Secure Crypto Engine (SCE)) .....	16
2.6.1.1 セキュリティアルゴリズム (Security Algorithms) .....	16
2.6.1.2 その他の暗号化セキュリティ機能 (Other Crypto Security Features) .....	16
3. セキュリティ要素の構成 (Configuring the Security Elements) .....	17
3.1 Synergy MCU のオプション設定メモリの概要 (Overview of Synergy MCU Option-Setting Memory) .....	17
3.2 セキュリティ MPU の構成 (Configuring the Security MPU) .....	18
3.2.1 セキュリティ MPU のレジスタの設定 (Setting up the Security MPU Registers) .....	18
3.2.2 特定のメモリ領域へのセキュアコード/データの配置 (Locating Secure Code/Data to a Specific Memory Region) .....	19
3.2.3 セキュリティ MPU のレジスタのリセット (Resetting the Security MPU registers) .....	19

3.3	FAW の構成 (Configuring the FAW) .....	20
3.3.1	FAW 領域の設定 (Setting up the FAW Region) .....	20
3.3.2	FAW 領域のクリア (Clearing the FAW) Regions .....	20
3.4	FAW 領域の永続的なロック (Permanent Locking of the FAW Region) .....	20
3.5	デバッグ向けのセキュリティ制御の設定 (Setting up the Security Control for Debugging) .....	21
3.5.1	OSIS の ID コード設定方法 (Methods of Setting the OSIS ID Code) .....	22
3.5.1.1	Synergy コンフィギュレータと J-Link デバッガの使用法 (Using the Synergy Configurator and J-Link Debugger) .....	22
3.5.1.2	J-Link スクリプトの使用法 (Using a J-Link script) .....	22
3.5.1.3	ファクトリブートローダの使用法 (Using the factory bootloader) .....	22
3.5.2	デバッグ向けの OSIS の ID コードの設定方法 (Method of Setting up the OSIS ID Code for Debugging) ..	22
3.5.3	OSIS の ID コードリセット方法 (Method of Resetting the OSIS ID code) .....	23
4.	セキュリティ MPU と FAW を使用する操作フロー (Operational Flow using Security MPU and FAW) .....	23
4.1	内部フラッシュと SRAM の読み取り保護 (Internal Flash and SRAM Read Protection) .....	23
4.2	内部フラッシュの書き込み保護 (Internal Flash Write Protection) .....	24
4.2.1	操作フロー (Operational Flow) .....	25
4.3	内部フラッシュと SRAM の読み取りと書き込みの保護 (Internal Flash and SRAM Read/Write Protection) .....	25
4.3.1	操作フロー (Operational Flow) .....	25
4.4	内部フラッシュの Write-Once 保護 (Internal Flash Write-Once Protection) .....	26
4.4.1	操作フロー (Operational Flow) .....	26
4.5	内部フラッシュの Write-Once 保護と読み取り保護 (Internal Flash Write-Once and Read Protection) .....	27
4.5.1	操作フロー (Operational Flow) .....	27
4.6	操作に関する注意 (Operation Notes) .....	27
4.6.1	メモリ割り当て (Memory Allocation) .....	27
4.6.1.1	セキュア命令と非セキュア命令の間隔 (Space between Secure and Non-secure Instructions) .....	27
4.6.1.2	メモリミラー領域のセキュア領域割り当て禁止 (Do Not Assign the Memory Mirror Region to the Secure Region) .....	27
4.6.1.3	セキュリティ MPU 領域のスタートアップエリア (Startup Area of Security MPU Regions) .....	27
4.6.2	オプション設定メモリのプログラミングに関する制限 (Limitations on Programming the Option-Setting Memory) .....	28
4.6.3	ファクトリブートローダのアクセシビリティ (Factory Bootloader Accessibility) .....	28
4.6.4	非セキュア関数からのセキュア関数のアクセス (Access Secure Function from Non-Secure Functions) ..	28
4.6.5	セキュリティ MPU 領域へのデバッガアクセス (Debugger Access to the Security MPU Regions) .....	28
5.	セキュリティアプリケーションに対応する e <sup>2</sup> studio プロジェクト: 内部フラッシュと SRAM (Security Application e <sup>2</sup> studio Projects: Internal Flash and SRAM) .....	28
5.1	プロジェクト 1 : e <sup>2</sup> studio プロジェクト : 内部フラッシュと SRAM の読み取りおよび書き込み保護 (Project 1 e <sup>2</sup> studio project: Internal Flash and SRAM Read Write Protection) .....	29
5.1.1	ソフトウェアアーキテクチャの概要 (Software Architecture Overview) .....	29

5.1.2	メモリ割り当ての配置 (Memory Allocation Arrangement) .....	30
5.1.3	機能に関する説明 (Functionality Description) .....	32
5.1.4	セキュア SRAM 領域からのソフトウェアの確立と実行 (Establishing and Running Software from Secure SRAM Region) .....	33
5.1.5	プロジェクトのインポートとビルド (Importing and Building the Project) .....	33
5.1.6	ハードウェアのセットアップ (Hardware Setup) .....	34
5.1.7	セキュア機能の検証 (Verifying the Secure Functionalities) .....	34
5.1.7.1	このアプリケーションノートが提供する機能の概要 (Summary of the functionalities provided in this application project:) .....	39
5.1.8	他の Synergy MCU への移行 (Migrating to other Synergy MCUs) .....	39
5.2	プロジェクト 2 : e <sup>2</sup> studio プロジェクト : セキュリティ MPU と FAW の設定のリセット (Project 2: e <sup>2</sup> studio Project: Reset the Security MPU and FAW setting) .....	39
5.3	プロジェクト 3 : FAW を永続的にロックするための PC アプリケーション (Project 3: PC Application to Permanently Lock the FAW) .....	40
5.4	セキュリティ MPU と FAW における リセット J-Link スクリプトの例 (Example Reset J-Link Script for the Security MPU and FAW) .....	42
5.5	S5D9 で OSIS の ID コードをリセットする J-Link スクリプト (J-Link Scripts for Resetting OSIS ID code for S5D9) .....	43
6.	保存データのセキュア化の次の手順 (Secure Data at Rest Next Steps) .....	46
6.1	セキュアデータの暗号化と認証 (Secure Data Encryption and Authentication) .....	46
6.2	外部ストレージの保存データのセキュア化 (External Storage Secure Data at Rest) .....	46
6.3	セキュリティ MPU のセキュリティ機能の使用例 (Example using the Security MPU Security Functions) .....	46
7.	参考資料 .....	46
	改訂記録 .....	48



## 1. セキュアデータの概要 (Secure Data Overview)

AI、IoT、クラウド接続の登場に伴い、営業機密 (trade secret) や個人のプライバシー (personal privacy) を保護する場合の最優先事項はデジタルデータのセキュリティです。

セキュアデータテクノロジーは、伝送中データ (Data in Transit) と保存データ (Data at Rest.) を含みます。伝送中データ、あるいは移動最中のデータとは、インターネットやプライベートネットワーク (専用網、private network) などを経由して、ある場所から他の場所にアクティブに移動しているデータのことです。伝送中データの保護とは、あるネットワークから他のネットワークに、またはローカルストレージデバイス (local storage device) からクラウドストレージデバイス (cloud storage device) に転送している最中のデータを保護することです。保存データとは、デバイスから他のデバイスへ、またはネットワークから他のネットワークへアクティブに移動していないデータのことです。つまり、ハードディスクドライブ (HDD)、ラップトップ (laptop)、フラッシュドライブ (flash drive)、組み込みメモリ (embedded memory) のいずれかに保存されているデータや、他の方法でアーカイブまたは保存されたデータのことです。保存データの保護は、あるデバイスまたはネットワークに保管されている非アクティブ (inactive) データをセキュア化することを目的としています。このアプリケーションノートは、Synergy MCU を使用する組み込み環境における保存データのセキュリティ設計に注目しています。

保存データを保護するために、主なセキュリティ手法 (security measure) として、データ暗号化 (Data Encryption) とデータアクセス制御 (Data Access Control) を使用します。このアプリケーションノートは、Synergy MCU 向けに Write-Once や読み取り/書き込みのアクセス制御を確立するための詳細な手順とともに、さまざまな保存データのセキュア化アプリケーションに対してこれらのセキュリティ機能を適用するためのガイドラインを提供します。本リリースではデータ暗号化を取り扱っていません。今後のリリースでは取り扱う可能性があります。

セキュアブートマネージャ (Secure Boot Manager, SBM) は、基準となる Synergy セキュアブートソリューションです。(このソリューションへの直接リンクについては、参考資料の章を参照してください。) 保存データのセキュア化について検討する場合、セキュアブートソリューションを使用すると、アプリケーションの設計全体にどのような影響を及ぼすのか考慮する必要があります。SBM を使用した保存データの活用方に関しては、このアプリケーションノートの適用外です。

### 1.1 保存機密データのトポロジー (Sensitive Data at Rest Topology)

組み込みシステムでセキュアデータが存在する場所は、揮発性 (volatile) データストレージ (MCU の内部 SRAM、または外部 SDRAM)、または不揮発性 (non-volatile) データストレージ (MCU の内部フラッシュストレージ、外部 QSPI ストレージ、外部 EEPROM ストレージなど) です。アプリケーションのセキュリティ設計の一部として、データの使用事例に基づき、データのトポロジーを考慮する必要があります。たとえば、医療機器 (medical device) の場合、特定のデータ (5 分ごとに測定した血圧など) は揮発性メモリに格納することがあるのに対し、他の種類のデータ (1 日の平均血圧など) は将来使用できるように不揮発性メモリに格納する必要があります。データの性質について考慮する必要があり、その結果、設計を開始する前にデータのトポロジーを決定することになります。この決定は、データのセキュア化に影響を及ぼすからです。

このアプリケーションノートを最初にリリースする際に、内部 SRAM または内部フラッシュストレージに存在しているデータをセキュア化する方法を示すサンプルコードを利用できるようにしました。このアプリケーションノートの今後のリリースでは、方法論について説明し、外部ストレージ (QSPI、EEPROM など) に存在するデータをセキュア化するためのサンプルプロジェクトを提供します。

### 1.2 保存データのセキュリティ対策 (Data at Rest Security Measures)

暗号化とアクセス制御は、保存データのセキュア化に関する 2 種類の主要な保護方式であり、本アプリケーションノートではこれらの点を説明します。これら 2 種類の方式は、内部ストレージと外部ストレージを含め、揮発性と不揮発性両方のストレージタイプに適用します。このアプリケーションノートを最初にリリースする際に、内部ストレージ (揮発性と不揮発性の両方) に関するアクセス制御もサンプルプロジェクトで取り扱います。

#### 1.2.1 データ暗号化 (Data Encryption)

データ暗号化は、保存データと伝送中データの両方をセキュア化するために広く使用されます。

暗号化を使用する内部データのセキュア化は、小規模な ARM Cortex MCU にとってますます不可欠になってきています。これらのデバイスは、ネットワークや通信のアプリケーションで広く使用されているからです。保存データのセキュア化とは、たいていの場合、データを暗号化すること、またはデータを不正アクセス (unauthorized access) からのデータ保護を目的とした暗号化機能を含むプロトコルが存在していることを意味します。たとえば、暗号化秘密鍵

(encrypted private key)を生成するために、Synergy デバイスの セキュア暗号化エンジン (Secure Crypto Engine, SCE)機能を使用します。

保存データを暗号化する使用例は、外部ストレージ内のデータの暗号化です。組み込みシステムで AES 鍵を使用して、外部ストレージ内にある機密のデータとコードを暗号化できます。認証に成功した時点で、外部のコードデータを暗号化解除し、使用することができます。

## 1.2.2 データアクセス制御 (Data Access Control)

デバイス接続の需要が増大し、組み込みシステムの複雑度が高まるにつれて、潜在的な攻撃対象領域 (attack surface)が増加する結果になっています。セキュアデータへのアクセスを制御すると、攻撃対象領域を実質的に減らし、その結果、システムのセキュリティを向上させることができます。Synergy が提供しているアクセス制御機能を使用できる可能性のあるいくつかの使用事例を、以下で簡単に紹介します。

### 1.2.2.1 読み取り保護 (Read Protection)

フラッシュと SRAM に存在している機密データとコードに対して、読み取り保護プロパティ (property)を設定することが可能です。その場合、読み取り権限 (read permission)を付与されたソフトウェアだけがそれらにアクセスできます。Synergy MCU はセキュリティ MPU (Security MPU) ユニートを搭載しており、読み取り保護を適用した機密領域の確立に役立ちます。

- 2.2 章で、セキュリティ MPU の機能を紹介します。
- 3.2 章で、セキュリティ MPU の設定方法を説明します。
- 4.1 章で、セキュリティ MPU の使用による読み取り保護を実施するための操作フローを説明します。

### 1.2.2.2 書き込み保護 (Write Protection)

機密データを、悪意のある改ざんや消去から保護することは重要です。揮発性と不揮発性どちらのデータも、不正書き込みから保護するために、Synergy デバイス内のメモリオプション設定を使用して書き込み保護することが可能です。フラッシュの書き込み保護を実施する方法は 2 通りあります。

Synergy セキュリティ MPU は、非セキュアソフトウェアによるアクセスに対して、フラッシュの消去アクセスと書き込みアクセスを無効にすることができます。詳細については、2.2 章を参照してください。

Synergy MCU のフラッシュアクセスウィンドウ (Flash Access Window, FAW) は、セキュアソフトウェアと非セキュアソフトウェアによる改ざんから、機密フラッシュデータを保護することができます。

- 2.3 章で、FAW の機能を紹介します。
- 3.3 章で、FAW の設定方法を説明します。
- 4.2 章で、FAW を使用した書き込み保護を実施するための操作フローを説明します。

### 1.2.2.3 読み取り保護と書き込み保護 (Read/Write Protection)

読み取り保護と書き込み保護は、マルウェアや IP 盗難 (IP theft) に対する攻撃対象領域を減らします。内部フラッシュデータの場合、書き込み保護と同様、Synergy デバイスで読み取り保護と書き込み保護を実施する 2 種類の方法があります。

Synergy セキュリティ MPU は、非セキュアソフトウェアに対して、セキュリティ MPU のフラッシュ領域と SRAM 領域への読み取りアクセスと書き込みアクセスを無効にすることができます。

セキュリティ MPU と FAW を組み合わせると、フラッシュ内の機密データを、セキュアソフトウェアと非セキュアソフトウェアによる読み取りと書き込みから保護することができます。このリリースは、この使用事例に対応するサンプルプロジェクトを提供しています。

- 4.3 章で、セキュリティ MPU と FAW の使用による読み取り保護と書き込み保護を実施するための操作フローを説明します。

### 1.2.2.4 Write-Once 保護 (Write-Once Protection)

特定の使用状況で、デバイスの寿命全体にわたって、保存機密データをアクセスや改ざんから保護する必要があることがあります。たとえば、セキュアブートローダは製品の寿命全体にわたって不変であることが求められます。データが内部フラッシュに存在する場合、Write-Once 保護を実施するために、FAW 設定をプログラミング (書き込み) できます。

設計でセキュアブートマネージャを使用することに伴う暗示的な意味に注意を払うことが重要です。最終アプリケーションが Renesas の SBM、つまりセキュアブートマネージャのようなセキュアブートローダを使用する場合、セキュアブートローダが自らのために予約する領域として、Write-Once のデータメモリ領域をそのアプリケーション内で確保するように、特別な注意を払う必要があります。Write-Once のポリシーを実現するために FAW のプロパティを設定する作業は、そのデバイスの寿命全体のうちに 1 回のみ実行できるので、この作業を実施する必要があります。すなわち、FAW ポリシーを 1 回プログラミングした（書き込んだ）後、そのポリシーを変更することはできません。

- 4.4 章で、1 回のみ書き込む(Write-Once)セキュアデータとプログラムの使用方法に関する操作フローを説明します。

### 1.2.2.5 Write-Once 保護と読み取り保護(Write-Once and Read Protection)

Write-Once 保護を実施したデータは、オプションで読み取り保護を実施することもできます。機密データを取り扱う場合、セキュアソフトウェアのみが内容を読み取ることのできる、Write-Once 保護を実施したフラッシュデータに対して、読み取り保護を実施することができます。Synergy MCU の場合、セキュリティ MPU と FAW の両方を使用する方法でこれらの保護を実施できます。

- 4.5 章で、Synergy MCU に対して Write-Once 保護と読み取りの保護を実施するための操作フローを説明します。

## 1.3 保存データのセキュア化のリスクプロファイルと攻撃対象領域の分析 (Secure Data at Rest Risk Profile and Attack Surface Analysis)

組み込み環境で保存データのセキュア化を十分に考慮して設計するために、以下のトピックを熟慮する必要があります。

1. 誰が組み込みシステムで機密データにアクセスできるのか考慮します。
2. CPU バスが機密データにアクセスできるかどうかを考慮します。
3. 他のバスマスタが機密データにアクセスできるかどうかを考慮します。アクセス可能な場合、バスマスタがどの周辺装置に接続するか、またこの周辺装置がどのエンティティ(entity)と通信するかを判断します。
4. デバッグが機密データにアクセスできるかどうかを考慮します。
5. 実施済みのセキュリティポリシーや対策を上書きした結果、アプリケーション自体が機密データに対して偶発的な損傷を及ぼす事態への対策を講じているか、などアプリケーションの設計の堅牢性を考慮します。

攻撃対象領域を減らすと、上記のような状況全般の改善に役立ちます。MCU のメモリ全体をセキュア化する方法は、全体のデータセキュリティ向上に実質的に寄与しない可能性があります。攻撃対象領域が大きくなると、ハッカーが弱点を見つけ出す可能性が大きくなるからです。機密データをセキュア化するための適切な指針は、最小量のデータのみをセキュア化するようにアプリケーションを設計し、戦略的なインターフェースを通じてアクセスを制御することです。

システムのリスクプロファイルと攻撃対象領域に関する分析は、このアプリケーションノートの適用範囲外ですが、攻撃対象領域を減らし、システムのリスクプロファイルを最小化できるように、Synergy が提供するセキュリティ対策を紹介します。

## 1.4 セキュアブートローダの使用時または不使用時の保存データのセキュア化 (Secure Data at Rest with and without Secure Boot Loader)

システムでセキュアブートローダソリューションを使用するかどうかは、保存データのセキュア化を実現する方法に影響を及ぼします。

1.2.2 章の データアクセス制御 (Data Access Control) で既に説明したように、アプリケーションがデバイスの寿命全体にわたって機密データを保護し、Renesas SBM のようなセキュアブートマネージャを使用する必要があることがあります。この場合、次の 2 点を考慮する必要があります。まず、FAW (フラッシュアクセスウィンドウ) のプロパティを適切に設定し、アプリケーション固有の機密データを、FAW 領域を使用して変更不可能にした領域に確実に割り当てることができるようにします。次に、製造のプロビジョニングプロセス実施中に、セキュアブートマネージャをプログラミングする（書き込む）のと同時にその機密データを MCU にプログラミングします。加えて、アプリケーションが読み取り保護のみを使用して機密データを保護する必要がある場合、アプリケーション固有の読み取り保護された機密データをセキュリティ MPU 領域内に確実に割り当てることができるように、セキュリティ MPU 領域を適切に考慮する必要があります。



Synergy デバイスがセキュリティ MPU の設定をユーザフラッシュの最初のブロック (8 KB のフラッシュブロック) 内に割り当てることを理解しておく役に立ちます。その結果、セキュリティ MPU の設定を永続的にロックする必要がある場合、内部フラッシュの最初の 8KB を、Write-Once 領域とする必要があります。SBM は、このような使用事例の 1 つの例です。詳細は、個別のデバイスのユーザーズマニュアルを参照してください。

## 2. 保存データのセキュア化に関連する Synergy MCU の機能 (Synergy MCU Features for Secure Data at Rest)

Synergy MCU は、保存データのセキュア化のニーズを満たす豊富なハードウェア機能セットを搭載しています。Synergy MCU は高いレベルから、以下のデータ保護機能をサポートしています。

- ・ 内部フラッシュ向けの Write-Once アクセス制御
- ・ 内部フラッシュ向けの読み取り/書き込みのアクセス制御
- ・ 内部 SRAM 向けの読み取り/書き込みのアクセス制御
- ・ バスマスタとデバッグからのデータアクセス保護
- ・ 揮発性と不揮発性両方の内部ストレージと外部ストレージのハードウェアデータ暗号化機能のサポート
- ・ 非セキュアプログラムが特定の周辺装置にアクセスすることを禁止するセキュリティ機能

この章は、上記のセキュリティ機能をサポートすることができる複数のハードウェア機能を紹介しますが、動作の詳細については説明しません。これらのハードウェアコンポーネントの設定に関する詳細は、3 章を参照してください。

MCU の内部データアクセス制御機能と、外部データ向けの暗号化サポートを統合した結果、組み込みシステムを対象とする一貫したレベルのセキュリティサポートを実現しています。

### 2.1 Synergy のセキュリティ要素の概要 (Overview of Synergy Security Elements)

以下に、Synergy MCU のセキュリティ要素の一覧を示します。

- ・ セキュリティ MPU
- ・ フラッシュアクセスウィンドウ (FAW)
- ・ セキュア暗号化エンジン (SCE)
- ・ OSIS(OCD/Serial Programmer ID Setting Register) を使用したデバッグ保護 (Debug Protection)

表 1 に、Synergy MCU におけるセキュア要素の利用可能状況を要約します。

表 1. Synergy のセキュリティ要素

MCU の機能	S7	S5	S3	S124 以外の S1	S124
セキュリティ MPU	N/A	利用可能	利用可能	利用可能	N/A
FAW	利用可能	利用可能	利用可能	利用可能	利用可能
SCE	利用可能 (*)	利用可能 (*)	利用可能 (*)	利用可能 (*)	利用可能 (*)
OSIS レジスタ	利用可能	利用可能	利用可能	利用可能	利用可能

注記 \*: Synergy MCU ファミリに関する SCE サポートの詳細は、2.6.1 章を参照してください。

また、本章は、ARM MPU、バスマスタ MPU、バスマスタ MPU の動作が、高レベルから保存データのセキュア化にどのように関連しているかも説明します。

### 2.2 セキュリティ MPU (Security MPU)

Synergy デバイスのセキュリティ MPU 機能は、さまざまなソフトウェアコンポーネントとハードウェアコンポーネントの間で分離を作成する方法で、一連の多様なデータセキュリティポリシーを実現します。セキュアなプログラムとデータ、および非セキュアプログラムとデータの識別は、アドレス位置に基づいています。

リセットベクタ(reset vector)をフェッチ(fetch)する前に、セキュリティ MPU の設定を読み取って適用するので、どのコードを実行するより前にこの設定を適用することになります。

**注記:** セキュリティ MPU のメモリ機能は、S7G2 と S124 を除く各 Synergy デバイスに存在しています。

### 2.2.1 セキュアデータ領域 (Secure Data Regions)

図 1 に、セキュリティ MPU が提供している、利用可能なセキュアデータ領域とセキュアプログラム領域を示します。各領域は、開始アドレスと終了アドレスのペア、およびイネーブルビット (enable bit) によって定義されています。

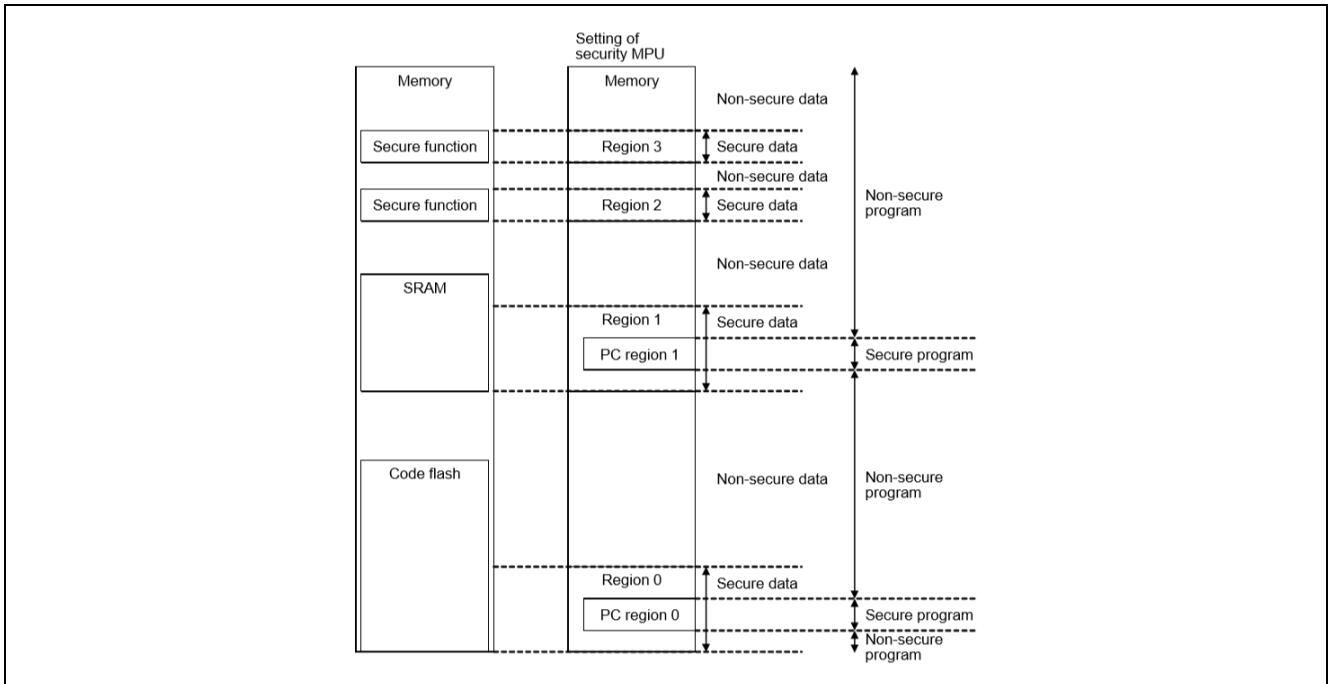


図 1.セキュリティ MPU のセキュア領域

セキュリティ MPU は、図 1 に示すように、4 個のセキュアデータ領域を提供しています。

1 個のセキュアフラッシュデータ領域 (MCU のコードフラッシュ領域内に配置)。セキュリティ MPU の使用方法に関して、セキュアフラッシュデータ領域はセキュアフラッシュデータとセキュアフラッシュプログラムの両方を保持することができます。

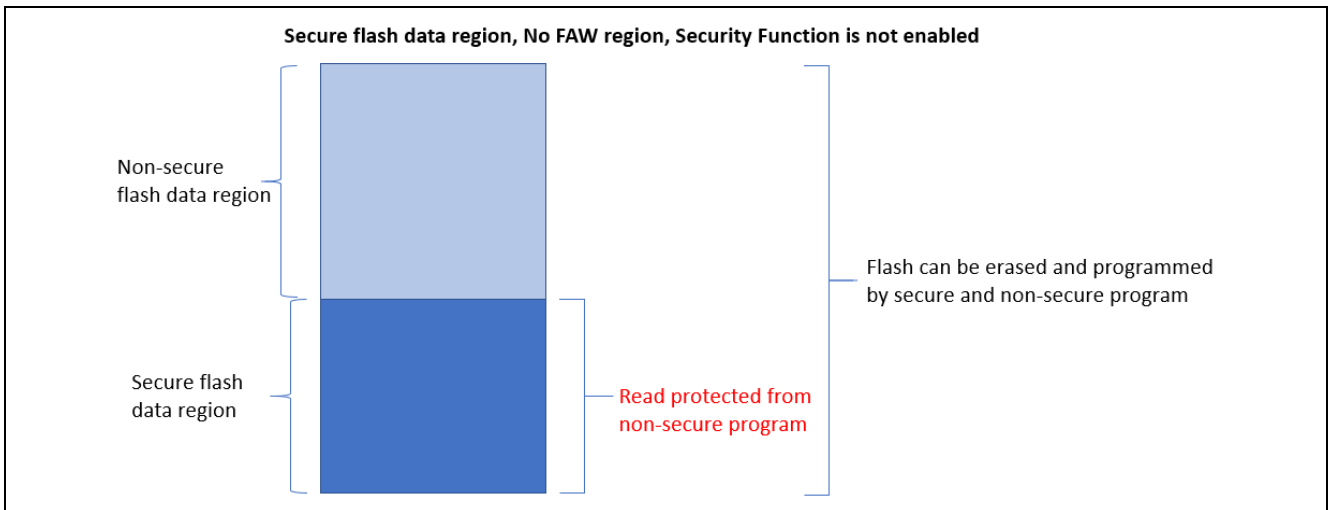


図 2.セキュアフラッシュデータ領域

1 個のセキュア SRAM データ領域。セキュリティ MPU の使用方法に関して、セキュア SRAM データ領域はセキュア SRAM データとセキュア SRAM プログラムの両方を保持することができます。

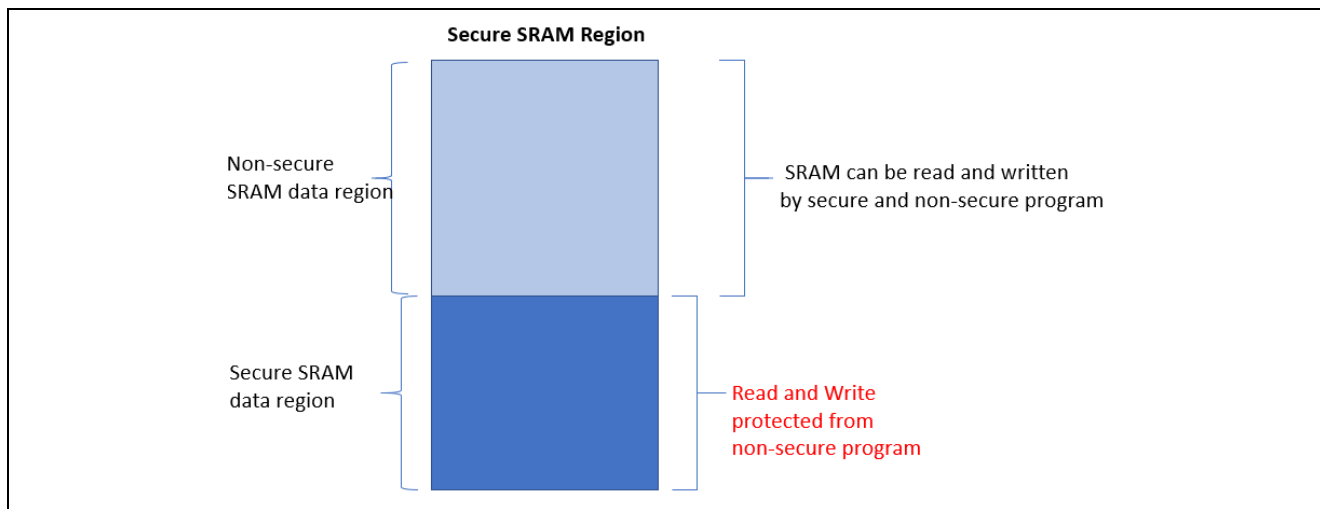


図 3.セキュア SRAM データ領域

2 つのセキュリティ機能領域

- セキュア暗号化エンジン(SCE)用のセキュリティ機能領域: アドレス 0x400C0000 ~ 0x400DFFFF
  - この領域は、Synergy MCU の内部周辺装置バス 7 にマッピングされています
  - このセキュリティ機能を有効にした場合、非セキュアプログラムから SCE を制御することはできません
- フラッシュ (コードとデータ) の消去とプログラミング (書き込み) (E/P) 用のセキュリティ機能: アドレス 0x40100000 ~ 0x407FFFFF
  - この領域は、Synergy MCU の内部周辺装置バス 9 にマッピングされています
  - このセキュリティ機能を有効にした場合、フラッシュは非セキュアプログラムから消去 / プログラムモードに移行することはできません

Table 15.2 Addresses assigned for each bus

Addresses	Bus	Area
0000 0000h to 01FF FFFFh	Memory bus 1, 3	Code flash memory
1FFE 0000h to 1FFF FFFFh	Memory bus 2, 3	SRAMHS
2000 0000h to 2003 FFFFh	Memory bus 4	SRAM0
2004 0000h to 200F FFFFh	Memory bus 5	SRAM1 and Standby SRAM
4000 0000h to 4001 FFFFh	Internal peripheral bus 1	Peripheral I/O registers
4004 0000h to 4005 FFFFh	Internal peripheral bus 3	
4006 0000h to 4007 FFFFh	Internal peripheral bus 4	
4008 0000h to 4009 FFFFh	Internal peripheral bus 5	
400C 0000h to 400D FFFFh	Internal peripheral bus 7	
400E 0000h to 400F FFFFh	Internal peripheral bus 8	Graphic IPs (JPEG, GLCDC, and DRW)
4010 0000h to 407F FFFFh	Internal peripheral bus 9	Flash memory (in P/E*) and data flash memory
6000 0000h to 67FF FFFFh	External bus	QSPI area
8000 0000h to 97FF FFFFh	External bus	CS area and SDRAM area

Note 1. P/E: Programming and erasure.

- S5D9 Usrc's Manual Rev.1.10 (R01UM0004EU0110 Rev.1.10)より引用

図 4.セキュリティ機能領域

### 2.2.1.1 セキュアプログラム (Secure Program)

セキュリティ MPU は、2 つのセキュアプログラム領域を提供しています。各領域は、開始と終了それぞれのプログラムカウンタ (Program Counter、PC) アドレスのペア、およびイネーブルビットによって定義されています。これらを 1 つのセキュアフラッシュプログラム領域と 1 つのセキュア SRAM プログラム領域にすることも可能です。セキュリティの目的で、図 1 に示すように、セキュアフラッシュデータ領域とセキュア SRAM データ領域のそれぞれの中に、セキュアプログラム領域を配置することを推奨します。

**注記:** セキュアフラッシュ領域やセキュア SRAM 領域の外部にセキュアプログラム領域を配置することも可能ですが、この場合はセキュリティホールが発生します。つまり、フラッシュと SRAM の中にあるこのようなセキュアプログラムは、非セキュアソフトウェアから読み取ることができます。

非セキュアプログラムとセキュアプログラムは、セキュア領域と非セキュア領域のそれぞれに対して関数呼び出しを行うことができ、この結果、組み込みシステム設計で効率化を実現できます。

### 2.2.1.2 セキュアアクセスの監視と保護 (Secure Access Monitor and Protection)

セキュリティ MPU は、図 5 に示すように、D コードバス (データアクセス) からの不正な CPU 読み取りに対し、セキュアフラッシュデータ領域とセキュア SRAM データ領域の監視と保護を行います。I コードバス (命令アクセス) は監視の対象ではありません。この結果、非セキュアプログラムがセキュア命令にアクセスすることが可能です。

セキュリティ MPU は、図 5 に示すように、システムバスアクセス (デバッグアクセスなど) に対し、セキュア SRAM データ領域とセキュリティ機能領域の監視と保護を行います。

セキュリティ MPU は、3 つのバスマスタグループからのアクセスに対し、4 つのセキュアデータ領域すべての監視と保護を行います。バスマスタグループの定義については、ハードウェアユーザズマニュアルを参照してください。

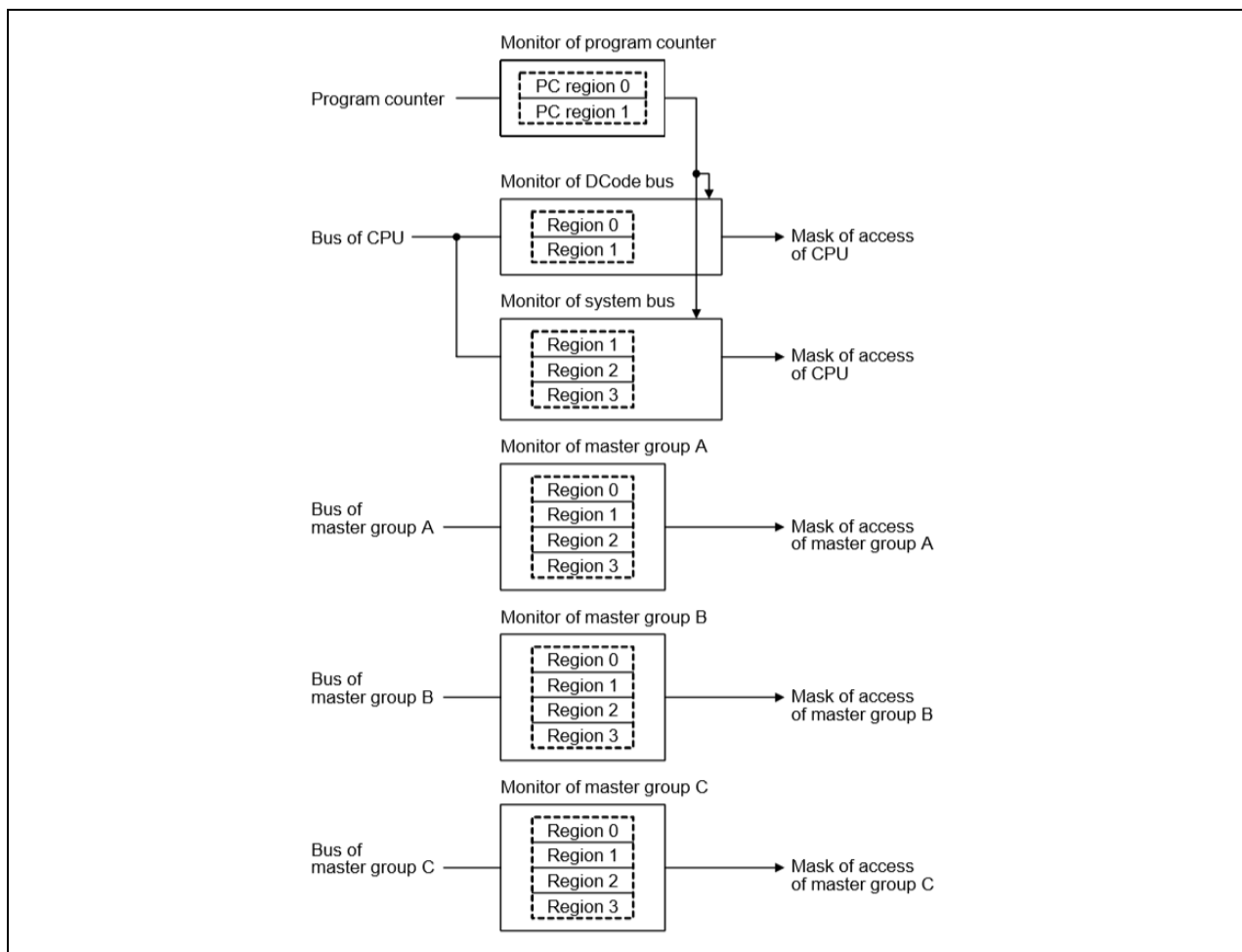


図 5.複数のセキュリティ MPU 領域に関するデータ保護

図 6 の表は、セキュア領域と非セキュア領域に対する CPU アクセスとデバッグアクセスの詳細を示しています。

- ✓ は、左側のバスマスタが上端(行の項目)の領域に対するアクセスを許可されていることを表します
- ✗ は、左側のバスマスタが上端(行の項目)の領域に対するアクセスを許可されていないことを表します

**注記:** 非セキュアプログラムは、SRAMHS 領域以外の SRAM 領域内に配置されているセキュアプログラムをフェッチすることができません。

セキュリティ MPU の設定に関する詳細は、3.2 章で説明します。

PC value	Master access	Slave region 0		Slave region 1				Slave region 2/3	
		FLASH		SRAMHS		Other than SRAMHS		E2P/Flash Cnt./TSIP	
		Sec	Non-sec	Sec	Non-sec	Sec	Non-sec	Sec	Non-sec
Secure	CPU (instruction)	✓	✓	✓	✓	✓	✓	-	-
Non-secure	CPU (instruction)	✓	✓	✓	✓	✗	✓	-	-
Secure	CPU (operand)	✓	✓	✓	✓	✓	✓	✓	✓
Non-secure	CPU (operand)	✗	✓	✗	✓	✗	✓	✗	✓
Secure	Debugger	✗	✓	✗	✓	✗	✓	✗	✓
Non-secure	Debugger	✗	✓	✗	✓	✗	✓	✗	✓
[Don't Care]	DMAC/DTC and others	✗	✓	✗	✓	✗	✓	✗	✓

図 6.CPU アクセスとデバッグアクセスの概要

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

フラッシュアクセスウィンドウ (FAW) は、MCU のフラッシュ空間内で、1 つの連続フラッシュ領域を定義します。この領域内で、セキュアプログラムと非セキュアプログラムの両方に対して、フラッシュの消去動作と書き込み動作が許可されます。このアクセスウィンドウは、プログラムフラッシュ領域内でのみ有効です。このアクセスウィンドウは、セルフプログラミングモード (self-programming mode)、シリアルプログラミングモード (serial programming mode)、およびオンチップデバッグモード (on-chip debug mode) で保護を実現します。

FAW 領域の設定は、フラッシュアクセスウィンドウの開始アドレスおよび終了アドレスとして、フラッシュのブロック境界 (block boundary) を使用します。FAW 制御レジスタ内にある 1 ビット (FSPR ビット) は、クリアされている (0 になっている) 場合、デバイスの寿命全体にわたって、フラッシュアクセスウィンドウの設定に対するいかなる更新も無効にします。内部フラッシュの Write-Once 保護機能は、このビットを使用して再書き込みからの保護を実現します。

DTC/DMAC、EDMAC、GLCDC/DRW(2DG)/JPEG のいずれかが有効になっている間、設定領域に書き込むことはできません。FAW の設定方法の詳細は、3.3 章を参照してください。

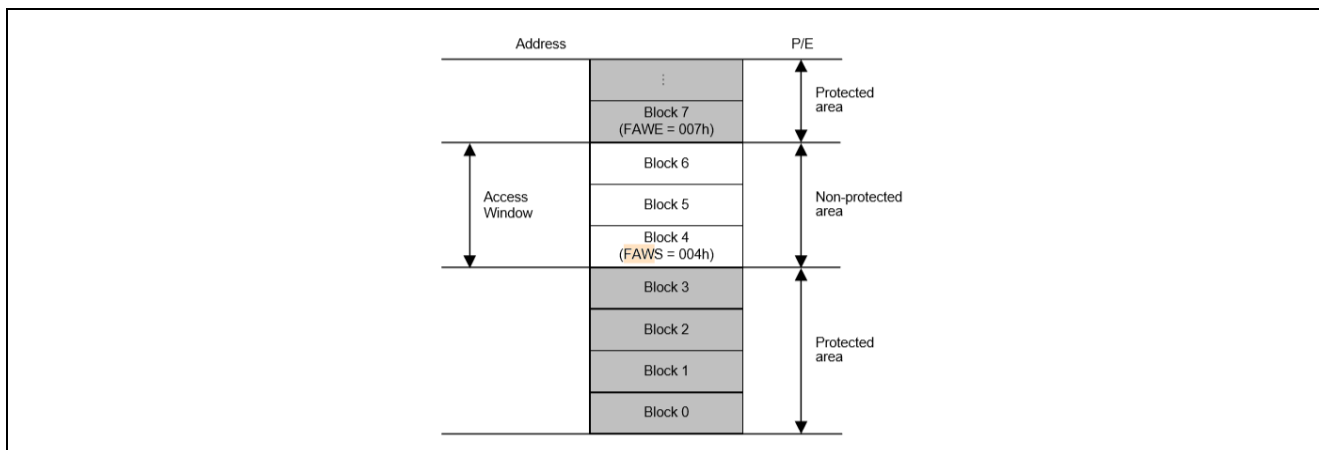


図 7.フラッシュアクセスウィンドウ



### 2.3.1 コードフラッシュの書き込み保護を実現するためのセキュリティ MPU と FAW の使用方法 (Using the Security MPU and FAW for Code Flash Write Protection)

1.2.2 章で説明したように、フラッシュ領域の書き込み保護を実施する方法は 2 通りあります。このアプリケーションノートは、FAW を使用してフラッシュの書き込み保護を実施しています。図 8 に、セキュリティ MPU と FAW を組み合わせて使用し、セキュリティ機能を有効にしていない場合のコードフラッシュの読み取りと、消去およびプログラミング (書き込み) の各アクセス制御を示します。

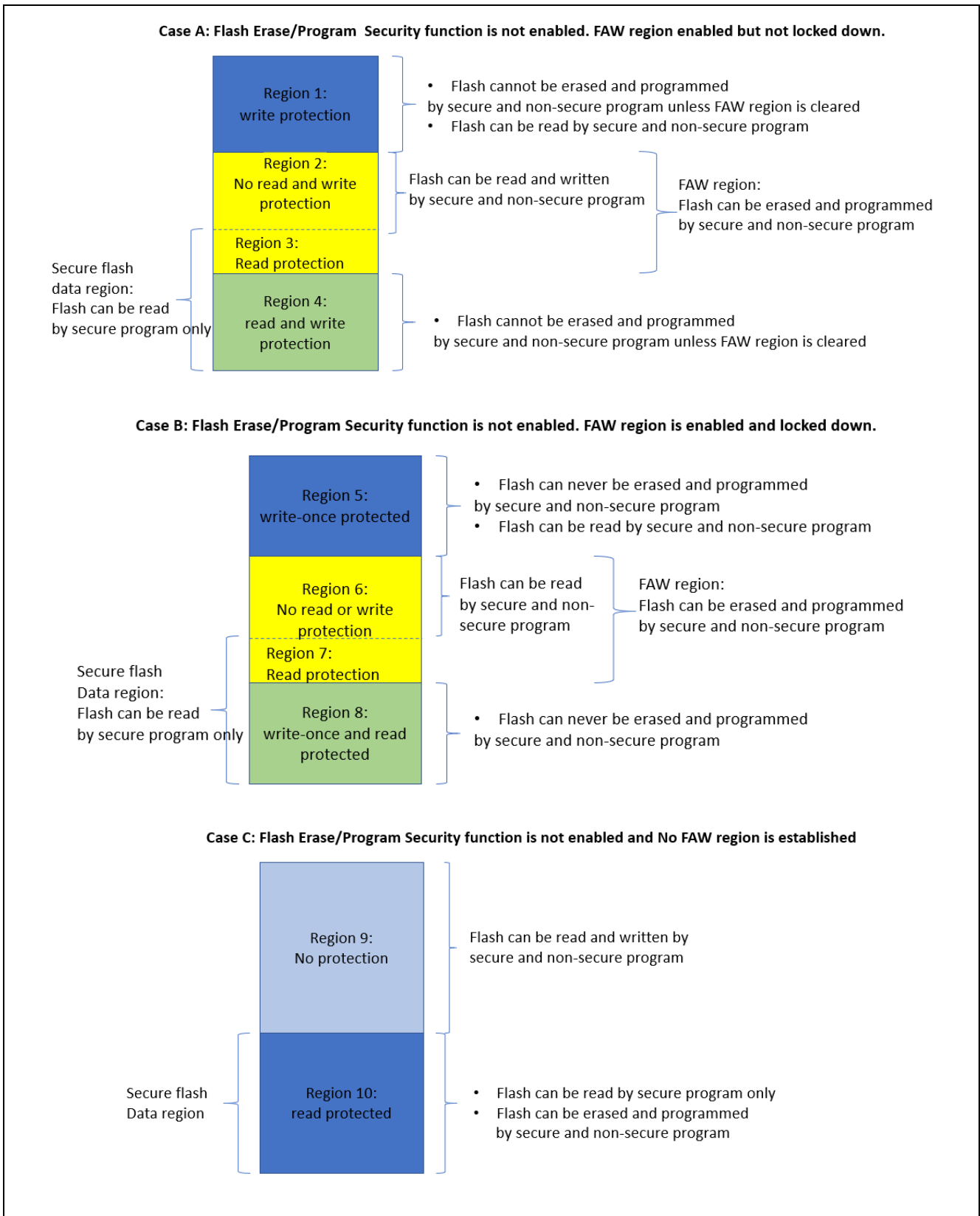


図 8.セキュリティ機能を有効にしない場合のコードフラッシュの読み取りと、消去およびプログラミング (書き込み) の制御

## 2.4 デバッグセキュリティに関する検討事項 (Debugging Security Considerations)

Synergy MCU は、OSIS レジスタを通じた OCD/シリアルプログラマ ID コードの保護機能を提供しています。OSIS レジスタは、OCD/シリアルプログラマの ID コード保護に使用する ID を格納しています。MCU に合わせて OSIS レジスタの ID コードを設定した時点以降、ユーザは OCD/シリアルプログラマに接続する際に、それに一致する ID コードを指定する必要があります。ID コードが一致した場合、デバッグが許可されます。一致しない場合、デバッグプロセスは許可されません。OSIS ID のコード保護の設定に関する詳細は、3.5 章で説明します。

## 2.5 ARM MPU、バスマスタ MPU、バススレーブ MPU に関する注記 (Notes on ARM MPU, Bus Master MPU, Bus Slave MPU)

本章は、MPU、バスマスタ MPU、バススレーブ MPU が保存データの設計にどのように関係しているかを説明します。ARM MPU の定義と設定について理解するには参考資料に示す Arm 社の資料を参照してください。バスマスタ MPU とバススレーブ MPU の定義と設定について理解するには、『Synergy MCU ハードウェアユーザズマニュアル』を参照してください。

これら 3 つの MPU は、これらの MPU が定義した領域に対する意図しないアクセス (inadvertent access) を捕捉 (catch) することを意図していますが、非セキュアプログラムによるレジスタ設定の読み取りや更新に対する保護機能を提供していません。これらの MPU のレジスタ設定は、デバッグまたは非セキュアプログラムによる読み取りから保護されていません。セキュアプログラムと非セキュアプログラムは、正しい手順に従って、レジスタを更新することができます。

さらに、これら 3 つの MPU が定義した MPU 領域は、セキュリティ MPU が提供している保護機能に比べると、同じ水準のセキュリティを提供していません。

- ・ デバッグは保護された領域にアクセスできます
- ・ 書き込み保護のない読み取り保護領域は、セキュアコードや非セキュアコードからの書き込みが可能です
- ・ 読み取り保護のない書き込み保護領域は、セキュアコードや非セキュアコードからの読み取りが可能です
- ・ 読み取り/書き込み保護領域は、セキュアコードや非セキュアコードのどちらからでも、読み取りと書き込みの両方が不可能です

## 2.6 その他のセキュリティ要素 (Other Security Elements)

### 2.6.1 セキュア暗号化エンジン (SCE) (Secure Crypto Engine (SCE))

Synergy MCU が搭載しているセキュア暗号化エンジン (SCE) ハードウェアブロックは、データ暗号化機能と認証機能を提供します。以下に、サポートしている暗号化アルゴリズムと認証アルゴリズムを示します。

#### 2.6.1.1 セキュリティアルゴリズム (Security Algorithms)

- ・ 対称型アルゴリズム (symmetric algorithm) : AES、3DES、ARC4
- ・ 非対称型アルゴリズム (asymmetric algorithm) : RSA、DSA、ECC
- ・ MCU ファミリのサポート状況:
  - S5 と S7 の各 MCU ファミリは、上記の対称型および非対称型アルゴリズムすべてをサポートしています
  - S3 と S1 がサポートしているのは AES アルゴリズムのみです

SCE 構成の詳細は、このアプリケーションノートの範囲外です。動作の詳細は、ハードウェアユーザズマニュアルと SSP ユーザズマニュアルを参照してください。

#### 2.6.1.2 その他の暗号化セキュリティ機能 (Other Crypto Security Features)

- ・ TRNG (True Random Number Generator、真の乱数生成)
- ・ ハッシュ値生成 : SHA1、SHA224、SHA256、GHASH、MD5
- ・ 128 ビットの一意 ID
- ・ MCU ファミリのサポート:
  - S5 と S7 の各 MCU ファミリは、この機能グループ全体をサポートしています
  - S3 は TRNG と GHASH をサポートしています
  - S1 は TRNG のみをサポートしています

### 3. セキュリティ要素の構成 (Configuring the Security Elements)

本章は、保存データのセキュリティの設計を目標のセキュリティレベルで実現するために、セキュリティ MPU、FAW、OSIS ID レジスタをセットアップするための詳細な手順を説明します。また、開発段階でセキュリティ機能の設定を取り消す方法も説明します。さらに、MCU 内でセキュリティ機能をロックするために必要なステップも掲載します。

#### 3.1 Synergy MCU のオプション設定メモリの概要 (Overview of Synergy MCU Option-Setting Memory)

セキュリティ MPU と FAW のレジスタは、以下に示すように、MCU のオプション設定メモリ (option-setting memory) の中に位置しています。オプション設定メモリは、フラッシュのユーザ領域内にあります。これらのレジスタを設定するには、フラッシュのコード領域やデータ領域内にある他の部分の消去やプログラミング (書き込み) とは異なる手順が必要です。図 9 は、S5D9 におけるオプション設定メモリの位置の例です。他の MCU は、これとは異なる位置に FAW レジスタを配置していることがあります。ご使用の MCU に対応する正確な位置については、ハードウェアマニュアルを参照してください。

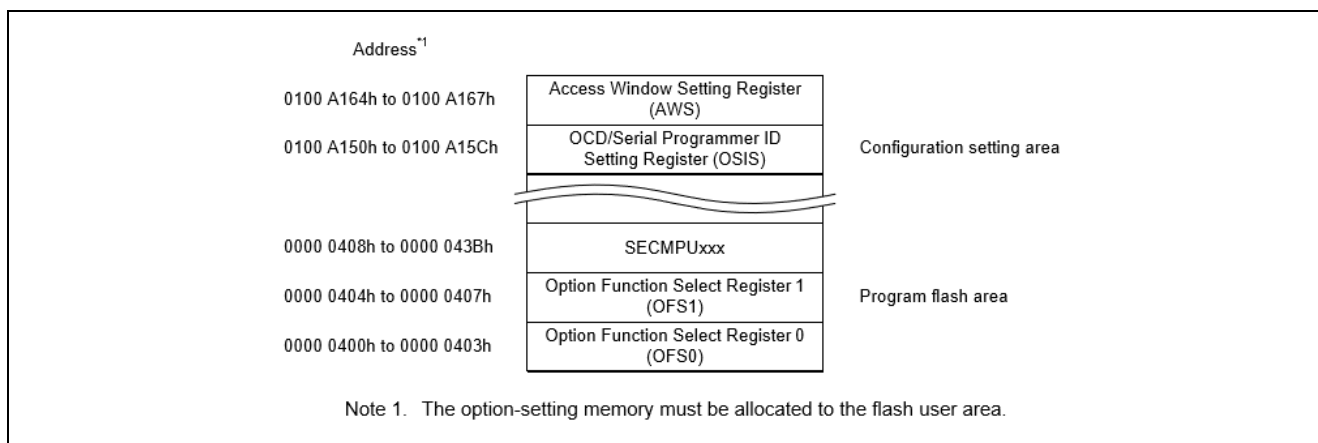


図 9. オプション設定メモリの領域

図 9 に示すように、セキュリティ MPU のレジスタは、フラッシュのプログラム領域のうち最初のセクタ内に割り当てられています。FAW レジスタと OSIS レジスタの設定は、構成の設定領域内に割り当てられています。

SSP v1.5.3 またはそれ以前を使用している場合、セキュリティ MPU と FAW のレジスタを設定するために、以下の方法を使用することを推奨します。

- ・ セキュリティ MPU のレジスタを定義するには、Synergy コンフィギュレータを使用します。3.2.1 章 で、この方法について説明します。
- ・ レジスタ設定のロックが必須ではない状況で、FAW レジスタを構成するには、SSP フラッシュドライバ API を使用します。FAW レジスタの設定をロックするには、ファクトリブートローダ (factory bootloader \*) を使用します。3.3 章で、この方法について説明します。(\* ファクトリブート: SCI/USB ブート)

セキュリティ MPU と FAW のレジスタ設定に関する他の方法は、以下の通りです。

- ・ J-Link スクリプトを使用します。
- ・ MCU のファクトリブートローダと通信し、ファクトリブートローダのユーティリティ機能を活用するには、シリアルポート (SCI/USB) を使用します。3.4 章 で、この方法を使用してロックビット FSPR をクリアつまり 0 に設定し、FAW レジスタの設定を永続的にロックする方法を説明します。
- ・ FAW レジスタをロックするために、セルフプログラミングを実行します。SSP v1.5.3 またはそれ以前は、この機能をサポートしていません。

## 3.2 セキュリティ MPU の構成 (Configuring the Security MPU)

本章は、セキュリティ MPU のレジスタを設定およびリセットするために必要な手順を説明します。

### 3.2.1 セキュリティ MPU のレジスタの設定 (Setting up the Security MPU Registers)

Synergy コンフィギュレータの SSP BSP タブで、ユーザは 4 つのセキュアデータ領域、および 2 つのセキュアプログラム領域を定義し、有効にすることができます。

The screenshot shows the 'Board Support Package Configuration' interface for the S5D9 PK board. The 'Settings' tab is active, displaying a table of Security MPU configurations. A red box highlights the following table:

Property Name	Value
HOCO Oscillation Enable	HOCO oscillation is disabled after
MPU - Enable or disable PC Region 0	Enabled
MPU - PC0 Start	0x00000400
MPU - PC0 End	0x0007FFFF
MPU - Enable or disable PC Region 1	Enabled
MPU - PC1 Start	0x1FFE0000
MPU - PC1 End	0x1FFFFFFF
MPU - Enable or disable Memory Region 0	Enabled
MPU - Memory Region 0 Start	0x400
MPU - Memory Region 0 End	0x000DFFFF
MPU - Enable or disable Memory Region 1	Enabled
MPU - Memory Region 1 Start	0x1FFE0000
MPU - Memory Region 1 End	0x2002FFFF
MPU - Enable or disable Memory Region 2	Disabled
MPU - Memory Region 2 Start	0x40100003
MPU - Memory Region 2 End	0x407FFFFFFF
MPU - Enable or disable Memory Region 3	Disabled
MPU - Memory Region 3 Start	0x400C0000
MPU - Memory Region 3 End	0x400DFFFF

図 10.セキュリティ MPU の構成

以下のリストに、上記のプロパティフィールド (太字で表示) で定義しているすべての領域を示します。



- ・ セキュアフラッシュプログラム
  - [PC0 Start] (PC0 開始) と [PC0 End] (PC0 終了): セキュアフラッシュプログラムに対応するプログラムカウンタ領域
  - [Enable or disable PC Region 0] (PC 領域 0 の有効化または無効化): セキュアフラッシュプログラムを有効または無効にします
- ・ セキュア SRAM プログラム
  - [PC1 Start] (PC1 開始) と [PC1 End] (PC1 終了): セキュア SRAM プログラムに対応するプログラムカウンタ領域
  - [Enable or disable PC Region 1] (PC 領域 1 の有効化または無効化): セキュア SRAM プログラム領域を有効または無効にします
- ・ セキュアフラッシュデータ領域
  - [Memory Region 0 Start] (メモリ領域 0 開始) と [Memory Region 0 End] (メモリ領域 0 終了): セキュアフラッシュデータ領域の開始アドレスと終了アドレス
  - [Enable or disable Memory Region 0] (メモリ領域 0 の有効化または無効化): セキュアフラッシュデータ領域を有効または無効にします
- ・ セキュア SRAM データ領域
  - [Memory Region 1 Start] (メモリ領域 1 開始) と [Memory Region 1 End] (メモリ領域 1 終了): セキュア SRAM データ領域の開始アドレスと終了アドレス
  - [Enable or disable Memory Region 1] (メモリ領域 1 の有効化または無効化): セキュア SRAM データ領域を有効または無効にします
- ・ セキュリティ機能領域 1
  - [Memory Region 2 Start] (メモリ領域 2 開始) と [Memory Region 2 End] (メモリ領域 2 終了): セキュリティ機能 1 の開始アドレスと終了アドレス (内部周辺装置バス 7 または 9 に対応するアドレス空間にする必要があります。以下の説明を参照)
  - [Enable or disable Memory Region 2] (メモリ領域 2 の有効化または無効化): セキュリティ機能 1 を有効または無効にします
- ・ セキュリティ機能領域 2
  - [Memory Region 3 Start] (メモリ領域 3 開始) と [Memory Region 3 End] (メモリ領域 3 終了): セキュリティ機能 2 の開始アドレスと終了アドレス (内部周辺装置バス 7 または 9 に対応するアドレス空間にする必要があります。以下の説明を参照)
  - [Enable or disable Memory Region 3] (メモリ領域 3 の有効化または無効化): セキュリティ機能 2 を有効または無効にします

アプリケーションノートでセキュリティ MPU を使用するための操作フローは、4.1 章を参照してください。

### 3.2.2 特定のメモリ領域へのセキュアコード/データの配置 (Locating Secure Code/Data to a Specific Memory Region)

セキュリティ MPU のレジスタを設定することに加え、セキュアコードとセキュアデータを、意図した領域に割り当てる必要があります。リンカスクリプトをカスタマイズする方法で、この割り当てを実施します。セキュアプログラムと非セキュアプログラムに対応する領域が存在している場合、リンカのデフォルト動作では十分ではありません。セキュリティ MPU を設計するときに、リンカスクリプト内で複数のフラッシュ領域や SRAM 領域を定義する必要があります。

セキュアコード/データを特定のメモリ領域に配置する例は、5.1.2 章を参照してください。

### 3.2.3 セキュリティ MPU のレジスタのリセット (Resetting the Security MPU registers)

セキュリティ MPU のレジスタが FAW 領域内に配置されている場合や、これらのレジスタが FAW 領域外に配置されているが FAW 領域の永続的なロックダウンが実施されていない場合、セキュリティ MPU のレジスタをリセットすることが可能です。

セキュリティ MPU をリセットするには、MCU のフラッシュの最初のブロックに対するフラッシュ消去動作を実施する必要があります。以下に、Synergy e<sup>2</sup> studio プロジェクトを使用してセキュリティ MPU の設定をリセットするための手順を示します。

1. SRAMHS 以外の SRAM 領域から実行するための SRAM プロジェクトを作成します
2. フラッシュ HAL ドライバを開きます
3. セキュリティ MPU のフラッシュ領域が FAW 領域の一部ではない場合、最初に FAW レジスタの設定をクリアします。FAW 領域をクリアする方法については、3.3.2 章を参照してください。

4. フラッシュ HAL ドライバを使用して、フラッシュの最初のブロックを消去します
5. フラッシュ HAL ドライバを閉じます

これら手順の実行については、5.2 章で説明している、e<sup>2</sup> studio のサンプルプロジェクト `secure_data_at_rest_pk_s5d9` を参照してください。

### 3.3 FAW の構成 (Configuring the FAW)

この章は、FAW 領域を設定およびクリアするための手順と、FAW 設定を永続的にロックするための手順を説明します。DTC/DMAC、EDMAC、および GLCDC/ DRW(2DG)/JPEG が有効になっている間、設定領域に書き込むことはできません。SSP v1.5.3 またはそれ以降で DMAC/DTC を無効にするには、DMAC/DTC を使用しているすべてのモジュールを閉じ、次のコードを呼び出します。

```
To disable DMAC/DTC, close all modules using DMAC/DTC and do the following:
ssp_err_t err;
ssp_feature_t ssp_feature = {{{(ssp_ip_t) 0U}}};
ssp_feature.id = SSP_IP_DTC;
err = R_BSP_ModuleStopAlways(&ssp_feature);
```

図 11.FAW 設定の準備

#### 3.3.1 FAW 領域の設定 (Setting up the FAW Region)

FAW を設定するには、DTC/DMAC、EDMAC、および GLCDC/ DRW(2DG)/JPEG を無効にした後、次の SSP FAW API を呼び出します。

```
err = g_flash0.p_api->accessWindowSet(g_flash0.p_ctrl, FAW_START, FAW_END);
```

- ・ `g_flash0.p_ctrl` は、このフラッシュ HAL ドライバの実例です。
- ・ `FAW_START` は、FAW ウィンドウの開始アドレスです。
- ・ `FAW_END` は、アクセスウィンドウが定義した、プログラミング (書き込み) と消去が受け入れられる次のブロックに対応するアドレスです。

**注記:** この API は常に、FAW のロックを無効にすることを意図しています。この API を実行する前に FAW を永続的にロックした場合、この API は予期したとおりの動作をせず、FAW 領域を何も更新できません。

#### 3.3.2 FAW 領域のクリア (Clearing the FAW) Regions

FAW の設定をクリアするには、DTC/DMAC、EDMAC、および GLCDC/ DRW(2DG)/JPEG を無効にした後、次の SSP FAW API を呼び出します。

```
err = g_flash0.p_api->accessWindowClear(g_flash0.p_ctrl);
```

`accessWindowClear` API は、FAW ウィンドウの設定をクリアし、開始アドレスと終了アドレスの両方を 0 に設定する方法で、FAW 制御を実質的に無効にします。このルーチンが機能するのは、FAW の永続的なロックを実施していない場合のみです。

開発段階で FSPR をクリアして 0 に設定する前に FAW の設定をクリアするには、Synergy e<sup>2</sup> studio プロジェクトで以下の手順に従います。

1. SRAMHS 以外の SRAM 領域から実行するための SRAM プロジェクトを作成します
2. フラッシュ HAL ドライバを開きます
3. `accessWindowClear` を使用して FAW の設定をクリアします
4. フラッシュ HAL ドライバを閉じます

### 3.4 FAW 領域の永続的なロック (Permanent Locking of the FAW Region)

FAW 領域を永続的にロックすると、ユーザによる FAW 領域の更新が防止されます。Write-Once フラッシュ領域を確立するには、この手順が必須です。

ユーザが FAW 領域の永続的なロックを考慮する 2 つの使用状況があります。

1. セキュアブートローダなしの使用状況でデバイスをフィールド (市場) に展開した際に、重要なランタイムデータをリアルタイムにロックする目的。

2. セキュアブートローダを使用する状況で、製造段階で重要なデータ/コードをロックする目的。

SSP v1.5.3 およびそれ以前は、FAW レジスタをそれ以降の更新から保護するための FAW ロック機能をサポートしていません。したがって、SSP v1.5.3 およびそれ以前で最初の使用事例をサポートすることはできません。

ファクトリーブートはFAWを設定とクリアする機能があります。製造時に、FAWレジスタを恒久的にロックする目的で、ファクトリーブートが利用できます。このアプリケーションノートで提供される PC プログラム "final\_program\_installer" は、FAW を恒久的にロックする PC プログラムの例です。

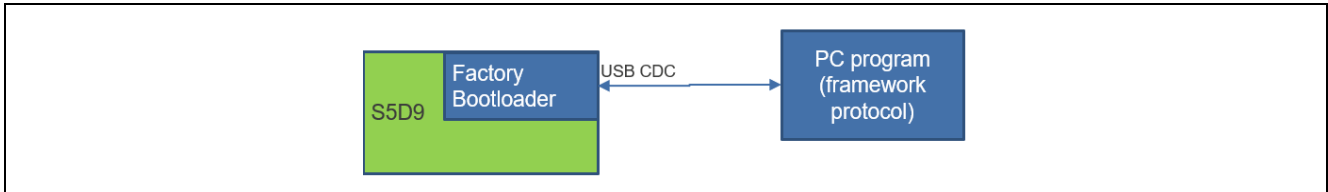


図 12.FAW の永続的なロック

### 3.5 デバッグ向けのセキュリティ制御の設定 (Setting up the Security Control for Debugging)

OCD/シリアルプログラミング ID 設定 (OCD/Serial Programming ID Setting, OSIS) レジスタは、OCD/シリアルプログラムの ID コード保護に使用する ID を格納します。接続時に、OCD/シリアルプログラムは ID の値を渡す必要があります。その結果、MCU はその接続を許可するかどうかを判定できます。

以下に、ID コードを設定するための複数のルールを簡単に示します。詳細は、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。

- ・ OSIS
  - 128 ビットレジスタは、デバッガ/プログラマを認証するための ID を保持しています
  - また、ファクトリーブート(SCI/USB ブート)モードのプログラミング (書き込み) でも、この ID コードを使用します
- ・ Bit[127] = 0 は、デバッグとファクトリプログラミングを完全に無効にします
- ・ Bit[127]=1, Bit[126] = 1 は、ID コードが一致した場合、デバッグとファクトリプログラミングを許可します
  - ユーザフラッシュ領域を消去する All Erase(ALeRASE コマンド)を受け付けます。ただし、FAW ウィンドウが永続的にロックされている場合、フラッシュを消去することはできません。
  - フラッシュブロックに関してセキュリティ MPU が有効になっている場合、ファクトリーブートローダは無効になります。
- ・ Bit[127]=1, Bit[126] = 0 は、ID コードが一致した場合、デバッグとファクトリプログラミングを許可します。

ユーザが OSIS の ID コードを設定できる複数の方法が存在します。

- ・ J-Link デバッガをサポートしている Synergy コンフィギュレータを使用します。
  - J-Link ドライバ V6.34 またはそれ以降が必須であることに注意してください
- ・ Renesas Flash Programming Tool (Renesas フラッシュプログラミングツール、RFP) を使用します
  - RFP のダウンロードリンクは、「参考資料」の章を参照してください
- ・ J-Link スクリプトを使用します
- ・ ファクトリーブートローダを使用します
- ・ セルフプログラミング (現時点で SSP 1.5.3 またはそれ以前はサポートしていません)

### 3.5.1 OSIS の ID コード設定方法 (Methods of Setting the OSIS ID Code)

#### 3.5.1.1 Synergy コンフィギュレータと J-Link デバッガの使用方法 (Using the Synergy Configurator and J-Link Debugger)

Synergy コンフィギュレータで、ユーザは 図 13 に示す SSP プロパティを使用して OSIS の ID コードを設定できます。

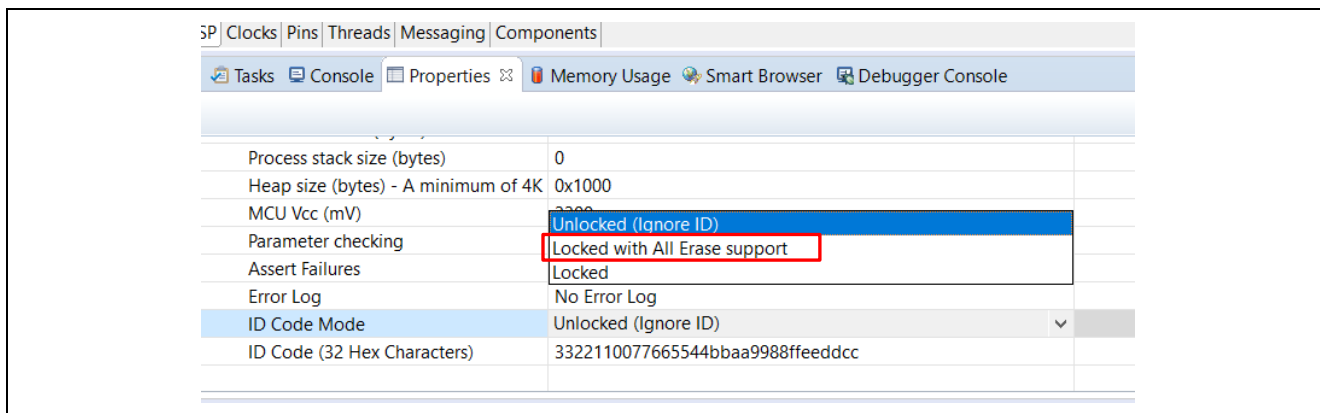


図 13. Synergy コンフィギュレータを使用した OSIS の ID の設定

開発段階では、[ID Code Mode] (ID コードモード) の選択肢として [Locked with All Erase support] (すべての消去をサポートするロック状態) を選択することを推奨します。5.5 章で、ID コードの設定をすべて FF にリセットする際に、フラッシュ全体を消去するサンプルの J-Link スクリプトを掲載しています。

[ID Code Mode] (ID コードモード) で [Locked] (ロック済み) を選択すると、デバイスに対するデバッグとファクトリプログラミングが無効になります。SSP1.5.3 またはそれ以前はセルフプログラミングをサポートしていないので、デバイスに対するデバッグとファクトリプログラミングは永続的に無効になります。

#### 3.5.1.2 J-Link スクリプトの使用方法 (Using a J-Link script)

Segger 社の J-Link を使用する J-Link スクリプトは、OSIS の ID を設定することができます。なおこのアプリケーションノートは、この用法に対応するサンプルスクリプトを提供していません。

#### 3.5.1.3 ファクトリブートローダの使用方法 (Using the factory bootloader)

Renesas SBM は、この使用事例に関係する 1 つの例です。Renesas SBM アプリケーションノートを参照してください。

### 3.5.2 デバッグ向けの OSIS の ID コードの設定方法 (Method of Setting up the OSIS ID Code for Debugging)

e<sup>2</sup> studio で、ユーザは下図のインターフェースを使用して ID コードを設定することができます。

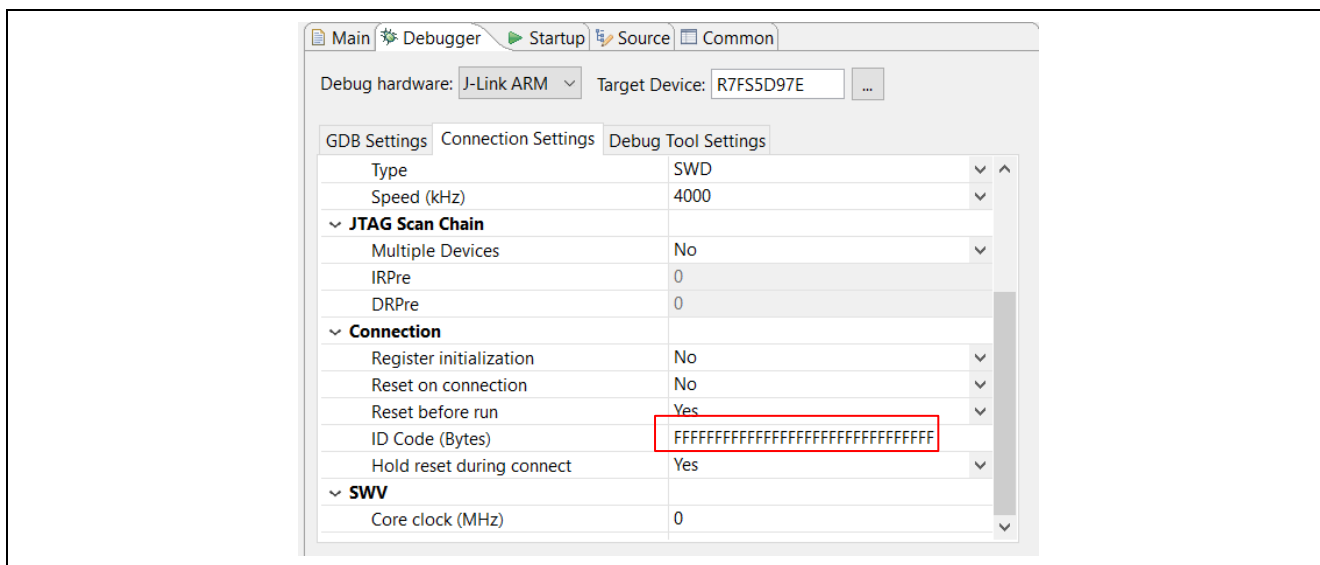


図 14. e<sup>2</sup> studio を使用した ID コードの設定

IAR Embedded Workbench for Renesas Synergy で、ユーザは下図のインターフェースを使用して ID コードを設定することができます。

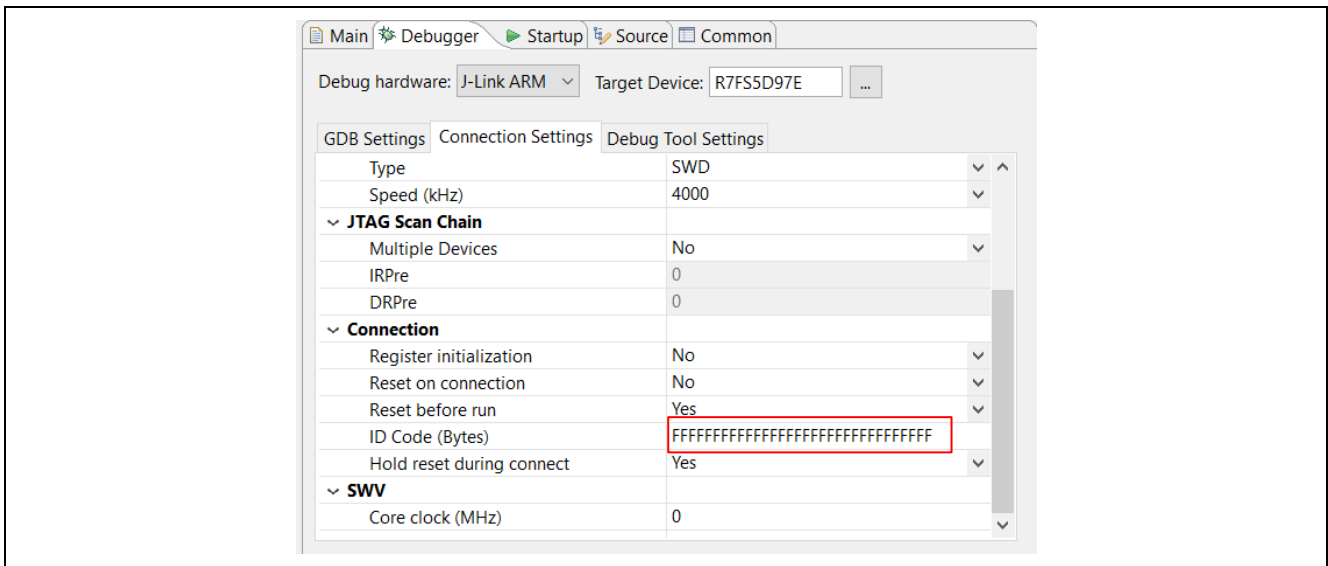


図 15. IAR を使用した ID コードの設定

### 3.5.3 OSIS の ID コードリセット方法 (Method of Resetting the OSIS ID code)

以下に、OSIS の ID コードのリセット方法を示します。

#### J-Link スクリプトの使用方法

このアプリケーションノートは、フラッシュ全体を消去する方法で OSIS の ID コードをリセットするサンプルの J-Link スクリプトを含んでいます。動作の詳細は、5.5 章を参照してください。

#### ファクトリブートローダの使用方法

ユーザは RFP を使用して OSIS の ID をリセットすることができます。

#### セルフプログラミング

セルフプログラミングを実行して OSIS の ID コードをリセットする方法は、今後の SSP のリリースでサポートする予定です。

## 4. セキュリティ MPU と FAW を使用する操作フロー (Operational Flow using Security MPU and FAW)

本章は、Synergy のセキュリティ MPU と FAW を使用するセキュアアプリケーションを設計するための操作フローについて説明します。

### 4.1 内部フラッシュと SRAM の読み取り保護 (Internal Flash and SRAM Read Protection)

セキュリティ MPU でセキュアフラッシュ領域とセキュア SRAM 領域を有効にした場合、これらの領域は読み取り保護されます。

- ・ セキュア SRAM 領域は、非セキュアフラッシュプログラムと非セキュア SRAM プログラムに対して、読み取り保護および書き込み保護されます
- ・ セキュアフラッシュ領域は、非セキュアフラッシュプログラムと非セキュア SRAM プログラムに対して、読み取り保護されます
- ・ バスマスタとデバッガから、セキュア領域へのアクセスは無効になります



セキュリティ MPU の複数の領域を確立するための高レベルの操作フローを、図 16 のフロー図に示します。図 8 の Region 3 (領域 3) と Region 7 (領域 7) は、内部フラッシュに関する読み取り保護の例です。

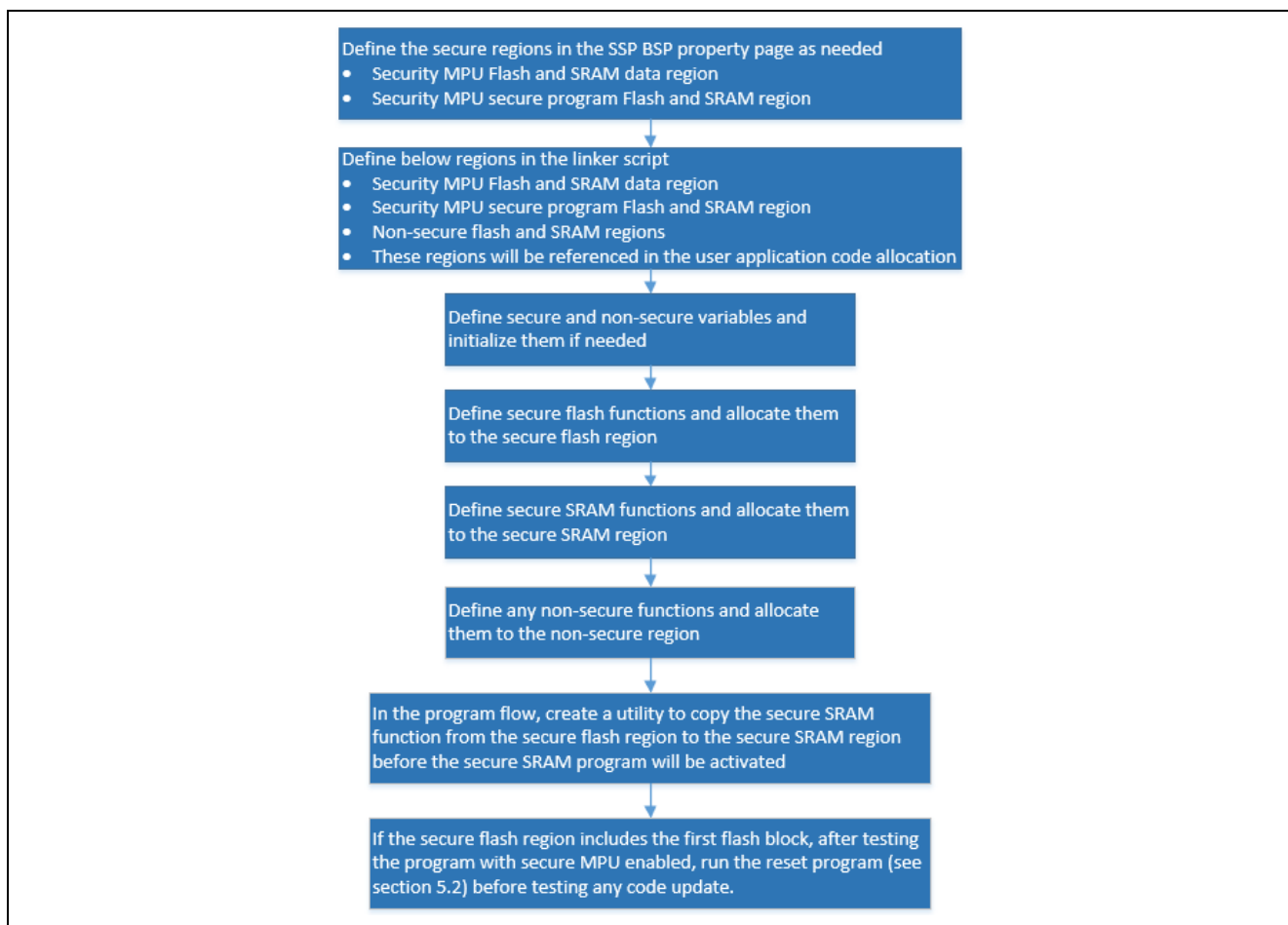


図 16.セキュリティ MPU を使用するための操作フロー

以下に、MCU に合わせてセキュリティ MPU を設定する方法の概要を示します。

- ・ セキュア領域を定義するには、コンフィギュレータの [BSP] タブを使用します。
- ・ セキュア領域と非セキュア領域を定義するには、リンクスクリプトを使用します
- ・ セキュア関数と非セキュア関数を確立します
- ・ セキュアフラッシュ領域から、セキュア SRAM 領域に、セキュア SRAM 関数 (ユーザ関数、定義済みの場合) をコピーします
- ・ システムをテストします

**注記:** セキュリティ MPU を有効にした後、新しいプログラムをデバイスにプログラミングする (書き込む) 場合は常に、MCU でリセットルーチンを実行する必要があります (5.2 章を参照)。

## 4.2 内部フラッシュの書き込み保護 (Internal Flash Write Protection)

4.1 章で説明したように、セキュア SRAM 領域に対してセキュリティ MPU を有効にした時点で、内部 SRAM 領域は書き込み保護されます。内部フラッシュ領域は、2 つの方法で書き込み保護することができます。

- ・ セキュアフラッシュ領域を含めないように、FAW 領域を設定します。その領域全体が (セキュアフラッシュ領域と非セキュアフラッシュ領域のどちらかの一部であってもかまいません)、FAW 領域の中に含まれていない場合、その領域全体はセキュアプログラムと非セキュアプログラムによる消去と書き込みから保護されます。図 8 の Region 1 (領域 1) は、内部フラッシュに関する書き込み保護の例です。
- ・ 内部の周辺装置バス 9 に対応するセキュリティ機能が有効になっている場合、セキュアフラッシュ領域と非セキュアフラッシュ領域を含めたフラッシュ領域全体が、非セキュアフラッシュプログラムと非セキュア SRAM プログラムによる消去とプログラミング (書き込み) から保護されます。

このアプリケーションノートは、最初の方法のみを記述しています。実施例は、5.1 章を参照してください。

#### 4.2.1 操作フロー (Operational Flow)

消去と書き込みを有効にした領域を設定するには、ユーザアプリケーションから SSP API の `accessWindowSet` (3.3.1 章を参照) を呼び出す必要があります。フラッシュ領域の残りの部分は、セキュアプログラムと非セキュアプログラムの両方による消去と書き込みが無効になります。

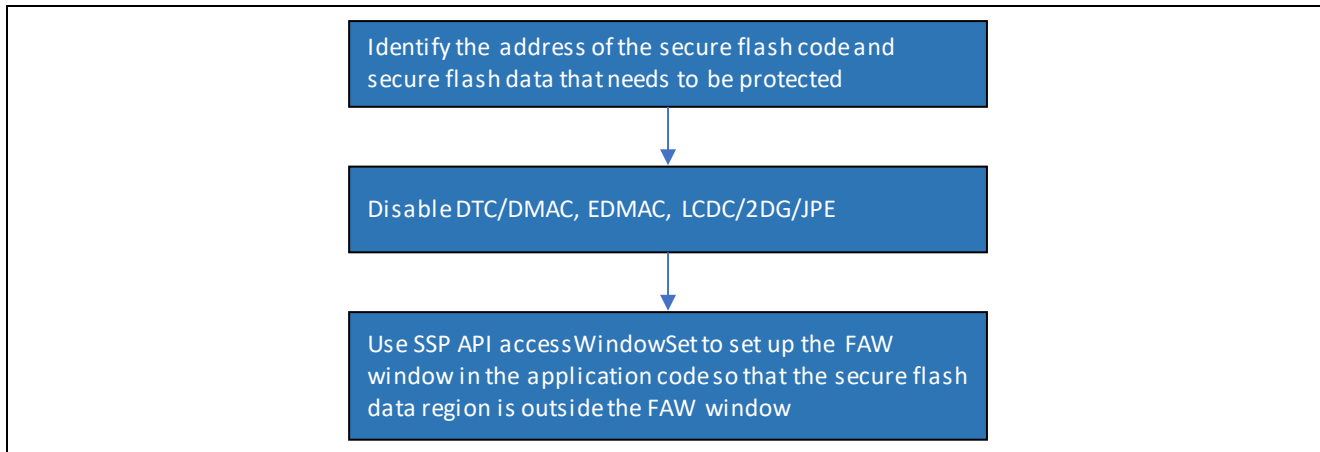


図 17.書き込み保護された内部フラッシュ領域の確立

### 4.3 内部フラッシュと SRAM の読み取りと書き込みの保護 (Internal Flash and SRAM Read/Write Protection)

セキュア SRAM 領域に対してセキュリティ MPU を有効にした時点で、セキュア SRAM は非セキュアフラッシュプログラムと非セキュア SRAM プログラムから読み取り/書き込み保護されます。

書き込み保護されたフラッシュ領域がセキュアフラッシュ領域の一部である場合、このセキュアフラッシュ領域は非セキュアプログラムによる読み取りと書き込みから保護され、セキュアプログラムによる書き込みから保護されます。図 8 の Region 4 (領域 4) は、内部フラッシュに関する読み取りと書き込み保護の例です。

#### 4.3.1 操作フロー (Operational Flow)

セキュア SRAM 領域を設定する方法で内部 SRAM 領域を読み取り保護した場合、その領域は同じプロセスによる書き込みからも保護されます。この操作フローの詳細は、4.1 章を参照してください。

FAW を使用して、書き込み保護された内部フラッシュ領域を設定し、その領域がセキュアフラッシュデータ領域の一部でもある場合、セキュアフラッシュ領域に属するこのセクションは、セキュアプログラムと非セキュアプログラムによる消去と書き込みから保護されます。セキュアフラッシュ領域全体は、非セキュアプログラムから読み取り保護されません。

内部 SRAM と内部フラッシュに関する読み取りと書き込み保護の実施例は、5.1 章を参照してください。

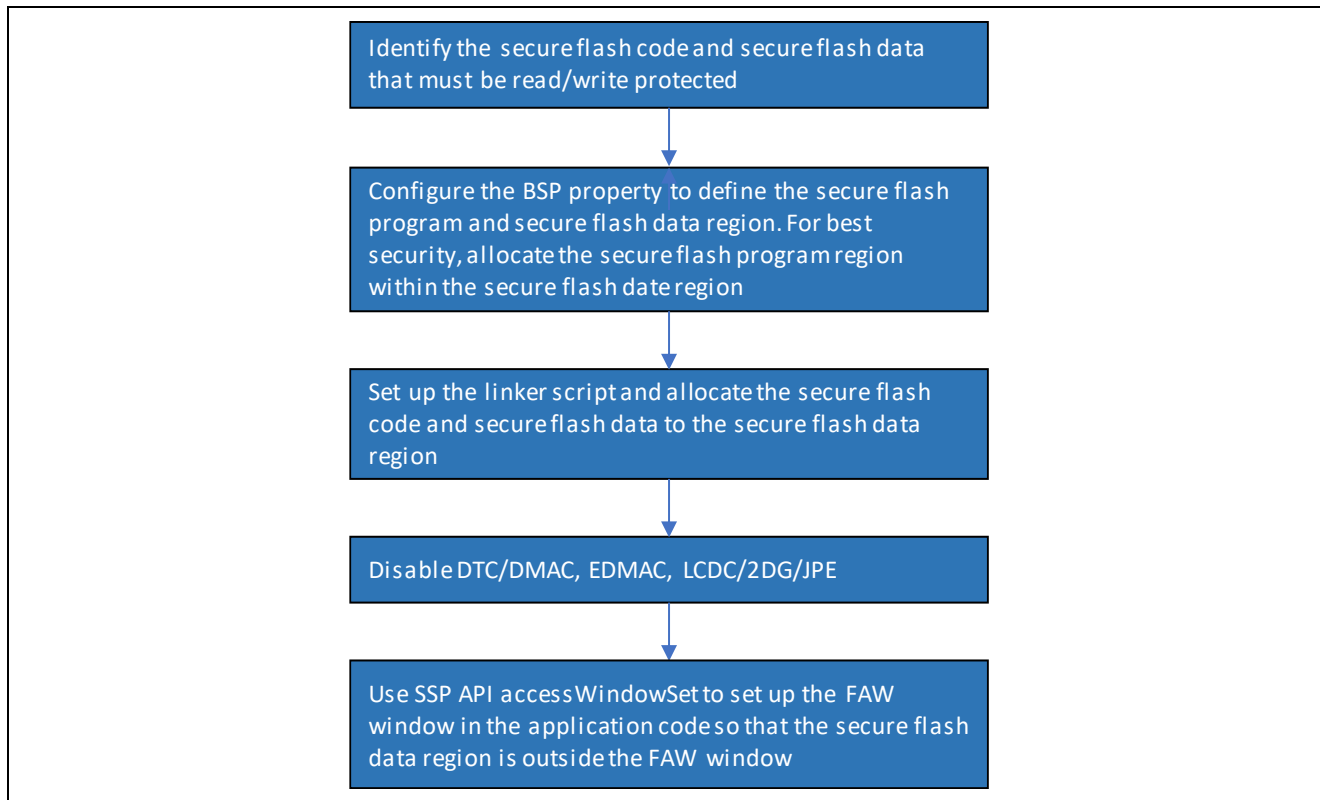


図 18.読み取り/書き込み保護された内部フラッシュ領域の確立

#### 4.4 内部フラッシュの Write-Once 保護 (Internal Flash Write-Once Protection)

FAW レジスタ内の FSPR ビットをクリア (0 に設定) した時点で、FAW ウィンドウ内に含まれていない内部フラッシュ領域は、セキュアプログラムと非セキュアプログラムから Write-Once 保護されます。この Write-Once 保護された領域の内容は、デバイスの寿命全体にわたって変更することができません。図 8 の Region 5 (領域 5) は、内部フラッシュに関する Write-Once 保護の例です。

##### 4.4.1 操作フロー (Operational Flow)

Write-Once 保護を実行する前:

- ・ アプリケーションがロックダウンすることを意図している重要なデータやコードを包括的にテストします
- ・ 他のどの重要なデータやコードも、Write-Once 保護された領域に割り当てる必要はありません

図 19 は、製造段階で操作する場合の操作フローです。Write-Once 保護の実装は、5.3 章を参照してください。

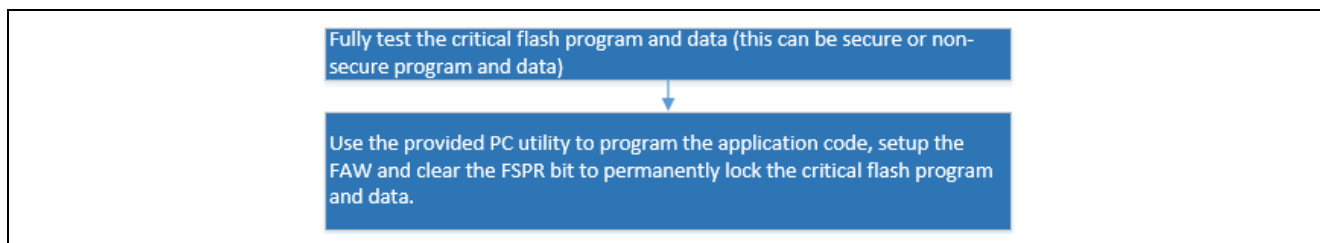


図 19 Write-Once 保護

3.4 章で説明したように、SSP1.5.3 またはそれ以前は、FAW レジスタをロックする方法による Write-Once 保護をサポートしていません。

## 4.5 内部フラッシュの Write-Once 保護と読み取り保護 (Internal Flash Write-Once and Read Protection)

Write-Once 保護と読み取り保護は、セキュアフラッシュプログラムとセキュアフラッシュデータに対して実施することもできます。セキュリティ MPU と FAW 領域を設定する方法でセキュアフラッシュ領域の読み取り/書き込み保護を有効にする場合、ユーザは FAW ウィンドウを永続的にロックすることができます。その場合、どのツールを使用しても対応する領域の消去やプログラミング（書き込み）は実施できなくなり、非セキュアプログラムによる読み取りからも保護されます。ファクトリブートローダ、J-Link スクリプト、セルフプログラミングを使用しても、このフラッシュ領域の消去とプログラミングは不可能です。

図 8 の Region 8 (領域 8) は、内部フラッシュに関する Write-Once 保護と読み取り保護の例です。

### 4.5.1 操作フロー (Operational Flow)

内部セキュアフラッシュアプリケーションの場合、セキュアプログラムと非セキュアプログラムから Write-Once 保護され、非セキュアプログラムから読み取り保護される領域を設定するために、ユーザはこれらの操作手順に従うことができます。

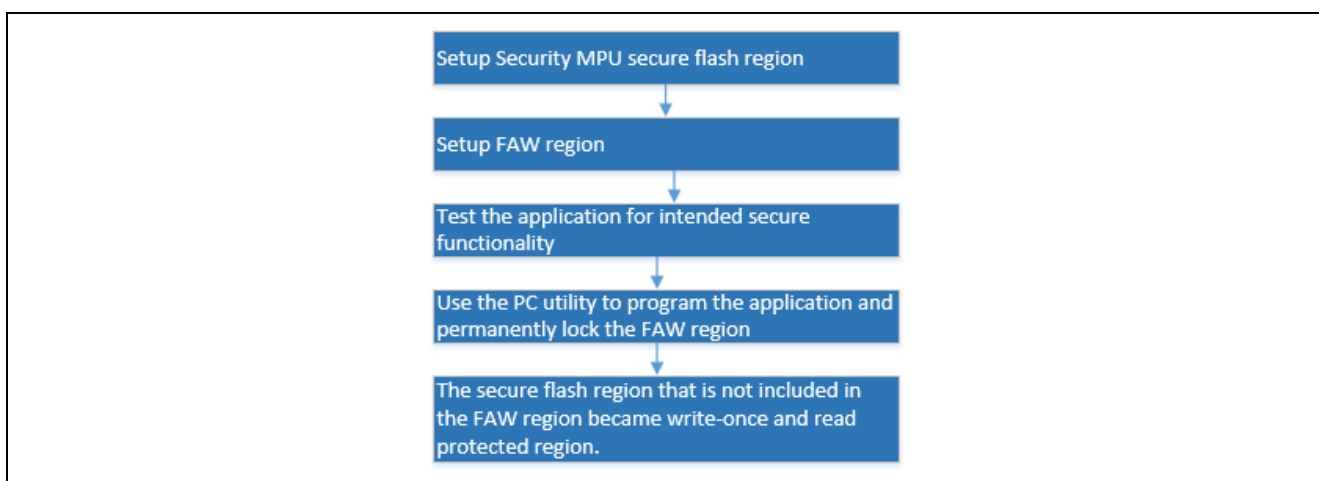


図 20. Write-Once 保護と読み取りの保護

FAW 領域のロックダウンの実施例は、5.3 章を参照してください。

## 4.6 操作に関する注意 (Operation Notes)

### 4.6.1 メモリ割り当て (Memory Allocation)

#### 4.6.1.1 セキュア命令と非セキュア命令の間隔 (Space between Secure and Non-secure Instructions)

非セキュアプログラムの最後の命令と、セキュアプログラムの最初の命令との間に、12 バイトを上回るアドレス空間を確保する必要があります。このギャップ（隙間）を確保せず、非セキュアプログラムの直後にセキュアプログラムを格納した場合、セキュアプログラムが非セキュアプログラムとしてフェッチされる可能性があります。詳細は、ハードウェアユーザズマニュアルを参照してください。

#### 4.6.1.2 メモリミラー領域のセキュア領域割り当て禁止 (Do Not Assign the Memory Mirror Region to the Secure Region)

MMF に対してメモリミラー空間 (memory mirror space) (0200 0000h ~ 027F FFFFh) を設定することは禁止されています。

#### 4.6.1.3 セキュリティ MPU 領域のスタートアップエリア (Startup Area of Security MPU Regions)

セキュアフラッシュデータ領域の開始アドレスエリアを、ベクタテーブルエリアに設定することはできません。

セキュア SRAM データ領域の開始アドレスを、スタックエリアまたはベクタテーブルエリアに設定することはできません。

#### 4.6.2 オプション設定メモリのプログラミングに関する制限 (Limitations on Programming the Option-Setting Memory)

ハードウェアユーザーズマニュアルの「Changing the option-setting memory by self-programming」(セルフプログラミングでオプション設定メモリを変更する場合)の章と「Debugging through an OCD or programming by a flash writer」(OCD によるデバッグ時またはフラッシュライターによってプログラムする場合)の章を参照し、FAW の設定と OSIS の ID 設定に関するオプション設定メモリのプログラミング (書き込み) に関する制限を理解してください。

このアプリケーションノートに含まれるアプリケーションコードは、ユーザーズマニュアルに記載された指針に従っています。

#### 4.6.3 ファクトリブートローダのアクセシビリティ (Factory Bootloader Accessibility)

セキュリティ MPU が有効になっている場合、ファクトリブートローダは利用できません。ファクトリブートローダを再度有効にするには、セキュリティ MPU のレジスタ設定をリセットする必要があります。セキュリティ MPU リセットの実施例は、5.2 章を参照してください。

#### 4.6.4 非セキュア関数からのセキュア関数のアクセス (Access Secure Function from Non-Secure Functions)

非セキュア関数からセキュア関数を呼び出すことができます (2.2 章の図 6 を参照)。

- ・ このリリースのアプリケーションノートは、非セキュア関数からセキュア関数を直接呼び出しています。
- ・ セキュアブートローダ内に配置されているセキュア関数にアクセスする場合、特別な考慮が必要です。
  - セキュア関数のエントリポイント (関数ポインタ) で構成したグループを、セキュアフラッシュデータ領域内に定義します
  - セキュア関数テーブル内に格納されている関数ポインタから、セキュア関数にアクセスします

#### 4.6.5 セキュリティ MPU 領域へのデバッガアクセス (Debugger Access to the Security MPU Regions)

デバッガがデバッグモードで動作している場合、セキュリティ MPU のフラッシュとセキュア SRAM 領域の内容を表示する機能は無効になります。e<sup>2</sup> studio を使用している場合、[memory] (メモリ) ビューからセキュア領域を表示すると、値は「0」と表示されます。

### 5. セキュリティアプリケーションに対応する e<sup>2</sup> studio プロジェクト: 内部フラッシュと SRAM (Security Application e<sup>2</sup> studio Projects: Internal Flash and SRAM)

本章は、セキュリティ MPU と FAW を使用するためのサンプルプロジェクトを紹介します。

- ・ セキュアフラッシュコードとセキュア SRAM コードから、セキュアフラッシュデータとセキュア SRAM データへのアクセス
- ・ 非セキュアフラッシュコードと非セキュア SRAM コードから、セキュアフラッシュデータとセキュア SRAM データへのアクセス

内部フラッシュと SRAM の保存データのセキュア化アプリケーションに関する 3 つのサンプルプロジェクトがあります。

- ・ プロジェクト 1: e<sup>2</sup> studio の Synergy プロジェクト (secure\_data\_at\_rest\_pk\_s5d9.zip):  
機能は以下:
  - セキュアフラッシュデータの読み取り保護
  - セキュアフラッシュデータとセキュアではないフラッシュデータの書き込み保護
  - セキュア SRAM の読み取り保護と書き込み保護
- ・ プロジェクト 2: e<sup>2</sup> studio の Synergy プロジェクト (reset\_flash\_pk\_s5d9.zip):  
機能は以下:
  - セキュリティ MPU のリセット
  - FAW 領域のリセット
- ・ プロジェクト 3: PC ベースの Visual Studio アプリケーション (programInstaller.sln):  
プロジェクト 1 で作成したアプリケーションをダウンロードし、MCU のファクトリブートローダを使用して FAW を永続的にロックするオプションを提供する、したがって、FAW に含まれていない領域の Write-Once 保護を実現しています。

さらに、このアプリケーションノートは、セキュリティ MPU、FAW、OSIS の ID コードそれぞれのリセットを行う、いくつかのサンプル J-Link スクリプトを収録しています。



5.4 章は、FAW 領域を永続的にロックダウンする前に FAW とセキュリティ MPU のレジスタをリセットする 1 個の J-Link スクリプトを説明します。

5.5 章は、e<sup>2</sup> studio で自動生成した blinky プロジェクトで OSIS の ID を設定する手順と、OSIS の ID をリセットするための J-Link スクリプトの使用法を説明します。OSIS の ID をリセットするスクリプトを、blinky プロジェクトと組み合わせて使用し、SSP 1.5.3 を使用して Synergy MCU でセキュリティの処理をデバッグするための開発をすべて示します。

## 5.1 プロジェクト 1:e<sup>2</sup> studio プロジェクト:内部フラッシュと SRAM の読み取りおよび書き込み保護 (Project 1 e<sup>2</sup> studio project: Internal Flash and SRAM Read Write Protection)

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

プロジェクト 1 は、フラッシュと SRAM の読み取り保護と書き込み保護を提示する 4 個のソフトウェアコンポーネントグループを実現します。

- ・ コンソールインタフェース (console Interface)
- ・ セキュアフラッシュコード
- ・ セキュア SRAM コード
- ・ 非セキュアフラッシュコード

このアプリケーションソフトウェア内に、SSP が提供する USB CDC コンソールフレームワークを使用する 1 つのコンソールスレッドが存在しています。このコンソールスレッドは、非セキュアフラッシュ領域内で動作します。このスレッドは COM ターミナルからのユーザ入力を監視し、対応するセキュアフラッシュとセキュア SRAM、および非セキュアフラッシュの各アプリケーションコードをアクティブにします。図 21 を参照してください。

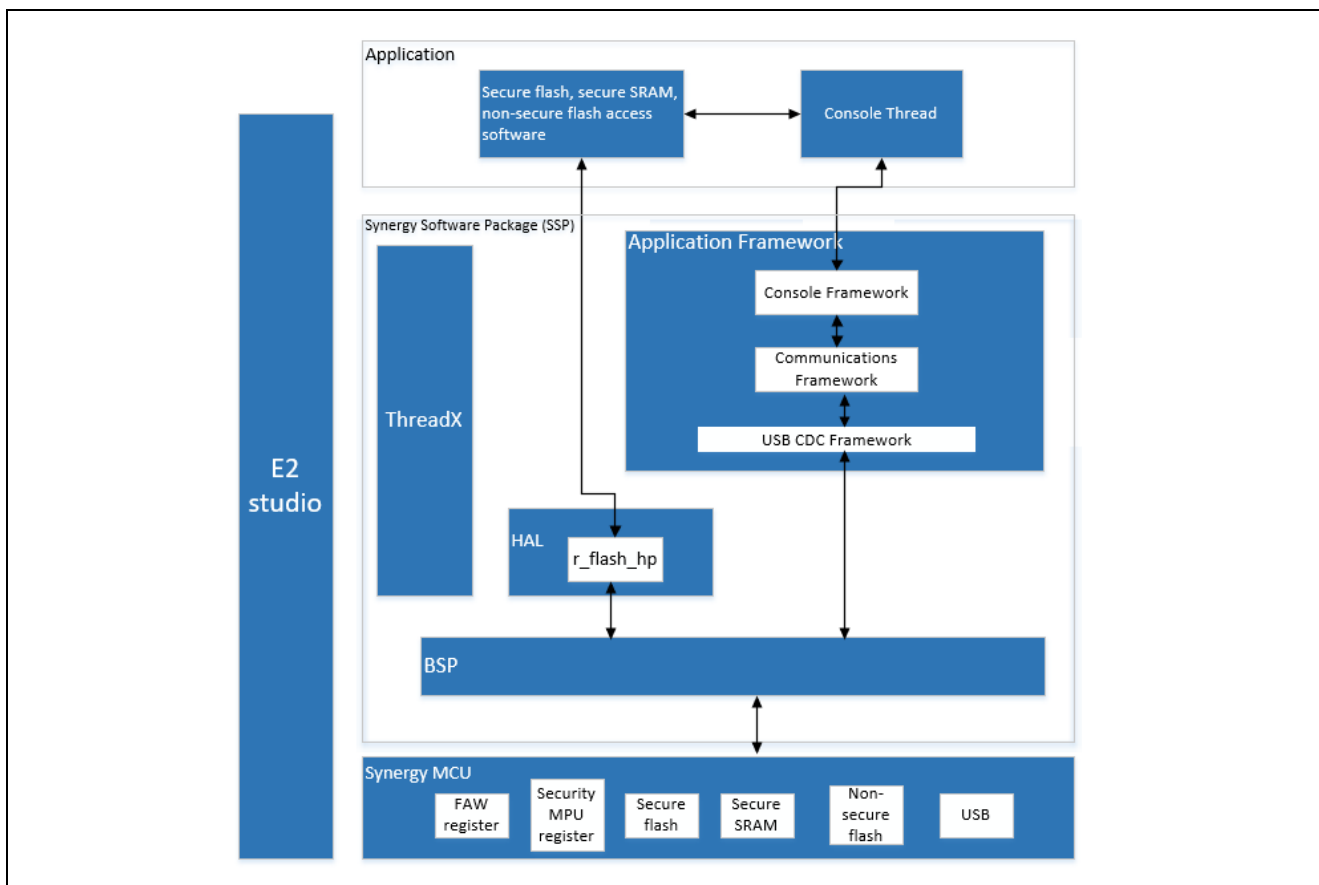


図 21.プロジェクト 1 のソフトウェアブロック図

### 5.1.2 メモリ割り当ての配置 (Memory Allocation Arrangement)

システムのメモリマップを制御するのは、リンクスクリプト `¥script¥S5D9.ld` です。ソフトウェアは `secure_definitions.h` 内で定義されているマクロ (macro) を使用して、さまざまなセキュアコードとセキュアではないコードを割り当てます。

図 22 に、フラッシュソフトウェアコンポーネントに基づくフラッシュメモリの割り当てを示します。

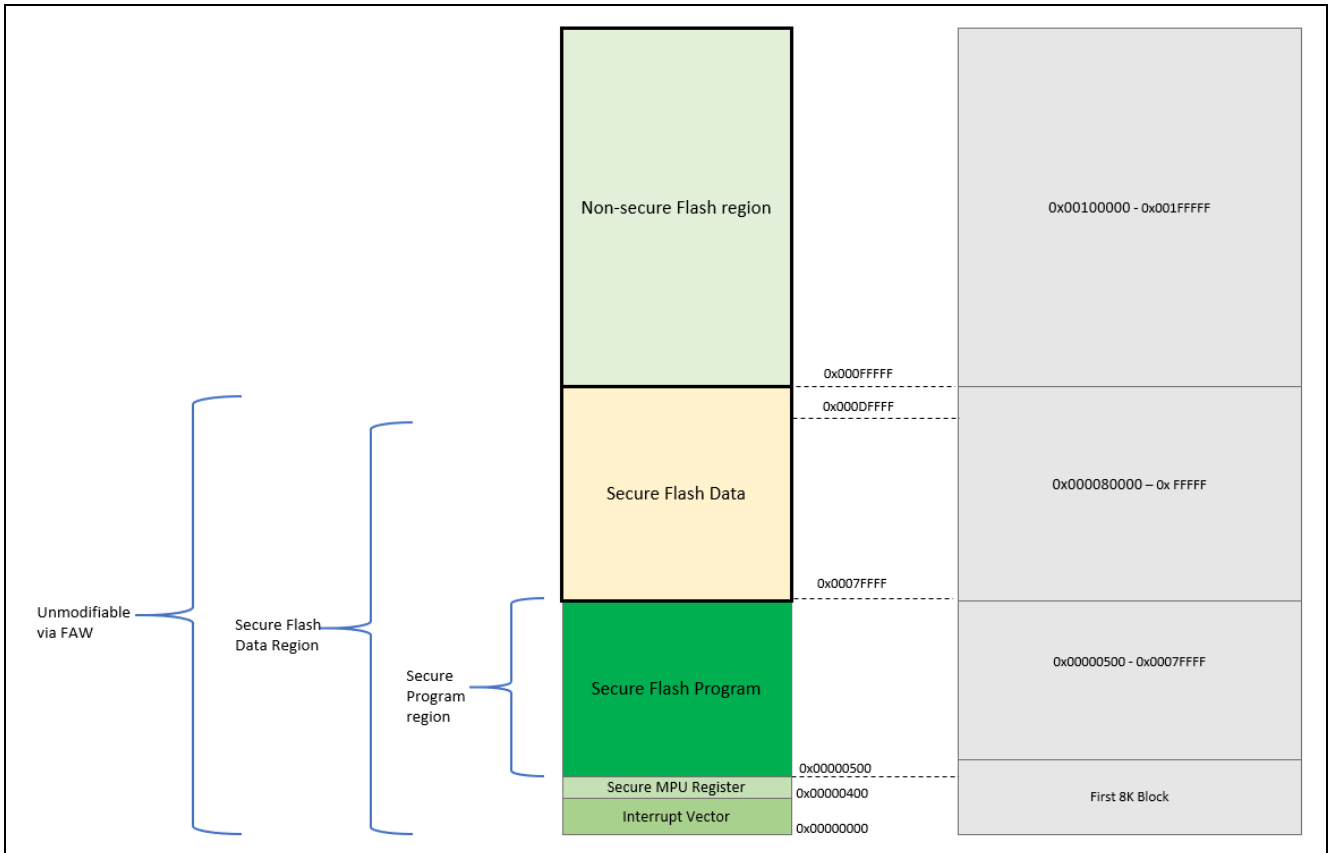


図 22.プロジェクト 1 の内部フラッシュ

図 23 に、さまざまな SRAM ソフトウェアコンポーネントに基づく SRAM メモリの割り当てを示します。

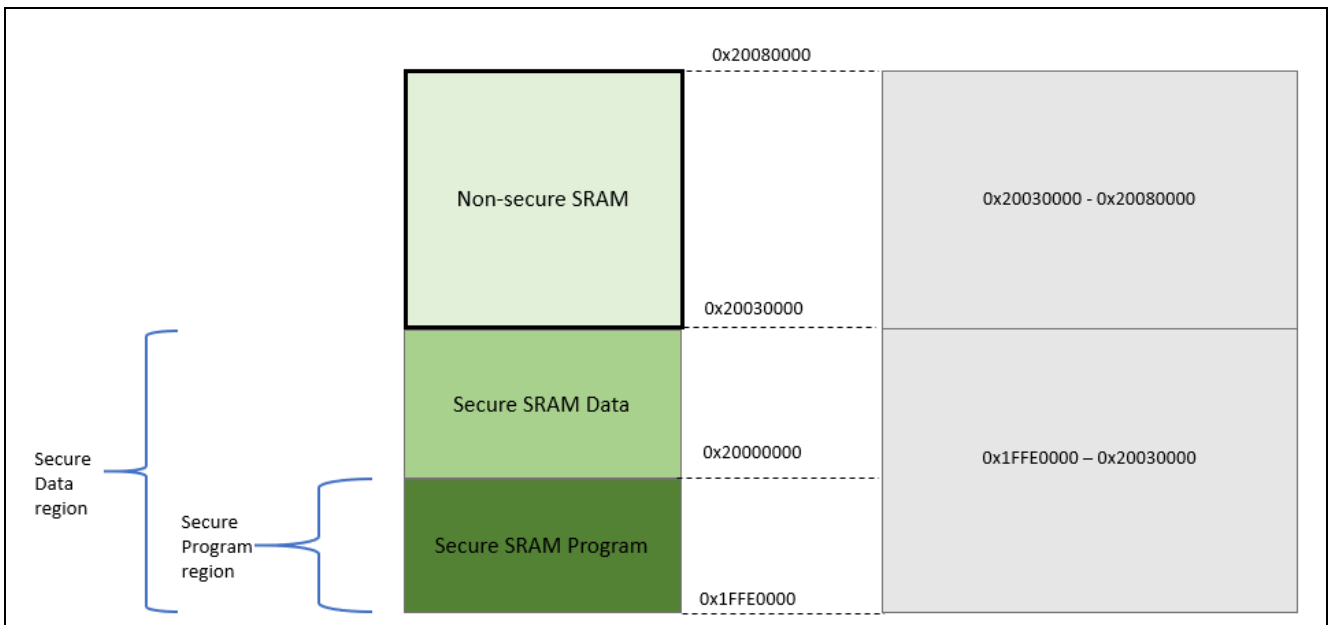


図 23.プロジェクト 1 の内部 SRAM

以下に、リンカスクリプト (¥script¥S5D9.ld) で定義したメモリ領域を示します。

```
/* Linker script to configure memory regions. (メモリ領域を構成するリンカスクリプト) */
MEMORY
{
    VECTOR_TABLE (rx) :ORIGIN = 0x00000000, LENGTH = 0x400 /* vector table, 1024 bytes (ベクタテーブル、1024 バイト) */
    SECURE_PROGRAM (rx) :ORIGIN = 0x00000400, LENGTH = 0x007FC00 /* secure flash program (セキュアフラッシュプログラム) */
    SECURE_DATA (rx) :ORIGIN = 0x00080000, LENGTH = 0x0080000 /* secure flash data (セキュアフラッシュデータ) */
    FLASH (rx) :ORIGIN = 0x00100000, LENGTH = 0x0100000 /* non-secure flash code (セキュアではないフラッシュコード) */
    SECURE_RAM_PROGRAM (rwx) :ORIGIN = 0x1FFE0000, LENGTH = 0x0020000 /* secure SRAM program (セキュア SRAM プログラム) */
    SECURE_RAM (rwx) :ORIGIN = 0x20000000, LENGTH = 0x0030000 /* secure SRAM data (セキュア SRAM データ) */
    RAM (rwx) :ORIGIN = 0x20030000, LENGTH = 0x0050000 /* non-secure SRAM program and data (セキュアではない SRAM プログラムとデータ) */
    DATA_FLASH (rx) :ORIGIN = 0x40100000, LENGTH = 0x0010000 /* 64K */
    QSPI_FLASH (rx) :ORIGIN = 0x60000000, LENGTH = 0x4000000 /* 64M, Change in QSPI section below also (64M、以下の QSPI セクション内も変更) */
    SDRAM (rwx) :ORIGIN = 0x90000000, LENGTH = 0x2000000 /* 32M */
}
```

以下に、secure\_definitions.h で定義したセキュアソフトウェアセクションとセキュアデータセクションが使用するメモリ割り当てマクロを示します。

```
#define SECURE_PROGRAM __attribute__((section (".secure_text")))
#define SECURE_CONST __attribute__((section (".secure_rodata")))
#define SECURE_DATA __attribute__((section (".secure_data")))
#define SECURE_BSS __attribute__((section (".secure_bss")))
#define SECURE_SRAM_PROGRAM __attribute__((section (".secure_sram_program")))
```

以下に、セキュアフラッシュデータとセキュア SRAM データに対する保護を提示する目的で使用する複数のテスト変数を示します。

非セキュアプログラムからの直接の読み取りと書き込みを許可するためのグローバル変数を使用して、セキュアデータエリアに対する読み取りと書き込みをシミュレートします。

表 2. テスト変数

データ型	グローバル変数名	データ型に関するコメント	機能に関するコメント
セキュアフラッシュデータの例	s_dataConst	定数	アクセス関数を用意されていません。非セキュアプログラムからこの変数へのアクセスは有効ではありません。
セキュアな初期化済み SRAM データの例	s_dataInit	初期化済み	アクセス関数を用意されていません。非セキュアプログラムからこの変数へのアクセスは有効ではありません。
セキュアな未初期化 SRAM データの例	s_dataWritten	未初期化	非セキュアフラッシュプログラムからこの変数への書き込み、およびその書き込みの表示は有効ではありません。

### 5.1.3 機能に関する説明 (Functionality Description)

#### (a) コンソールユーザインタフェース

このコードセクションは、CLI の入出力を制御します。

#### (b) セキュアフラッシュプログラム

- ・ セキュアフラッシュプログラムから、セキュア SRAM 領域と非セキュア SRAM 領域に対する読み取りと書き込み
  - すべての SRAM 領域へのアクセスの表示が許可される
- ・ セキュアフラッシュプログラムからセキュアフラッシュデータ領域と非セキュアフラッシュデータ領域に対する読み取り
  - すべてのフラッシュ領域へのアクセスの表示が許可される

#### (c) セキュア SRAM プログラム

- ・ セキュア SRAM プログラムから、セキュア SRAM 領域と非セキュア SRAM 領域に対する読み取りと書き込み
  - すべての SRAM 領域へのアクセスの表示が許可される
- ・ セキュア SRAM プログラムから、セキュアフラッシュ領域と非セキュアフラッシュ領域に対する読み取りと書き込み
  - すべてのフラッシュ領域へのアクセスの表示が許可される

#### (d) 非セキュアフラッシュプログラム

- ・ 非セキュアフラッシュプログラムから、セキュア SRAM 領域と非セキュア SRAM 領域に対する読み取りと書き込み
  - 非セキュア SRAM 領域へのアクセスの表示が許可される
  - セキュア SRAM 領域へのアクセスの表示が許可されません
- ・ 非セキュアフラッシュプログラムからセキュアフラッシュ領域と非セキュアフラッシュ領域に対する読み取りと書き込み
  - FAW を設定する前は、すべてのフラッシュ領域に対する書き込みアクセスが許可されることを示します。これは、内部フラッシュに関する読み取り保護の例です (4.1 章を参照)
  - FAW を設定した後は、FAW 以外の領域に対する書き込みアクセスが許可されないことを示します。これは、内部フラッシュに関する書き込み保護の例です (4.2 章を参照)
    - この FAW 以外の領域がセキュリティ MPU 領域の一部である場合、内部フラッシュに関する読み取り保護と書き込み保護の例になります (4.3 章を参照)。
  - (FSPR ビットをクリアして 0 に設定する形で) FAW の設定をロックした場合:
    - 上記の書き込み保護されたフラッシュ領域は、Write-Once 保護された領域になります (4.4 章を参照)
    - 上記の読み取り保護および書き込み保護されたフラッシュ領域は、Write-Once 保護および読み取り保護された領域になります (4.5 章を参照)

SECURE\_PADDING に対応するマクロ定義に関する特別な注記: このマクロは、4.6.1.1 章で説明したセキュリティ MPU アーキテクチャの要件に従い、セキュアコード実行領域と非セキュアコード実行領域の間にギャップ (隙間) を挿入します。

図 24 に、機能別にグループ分けしたソースファイルを示します。

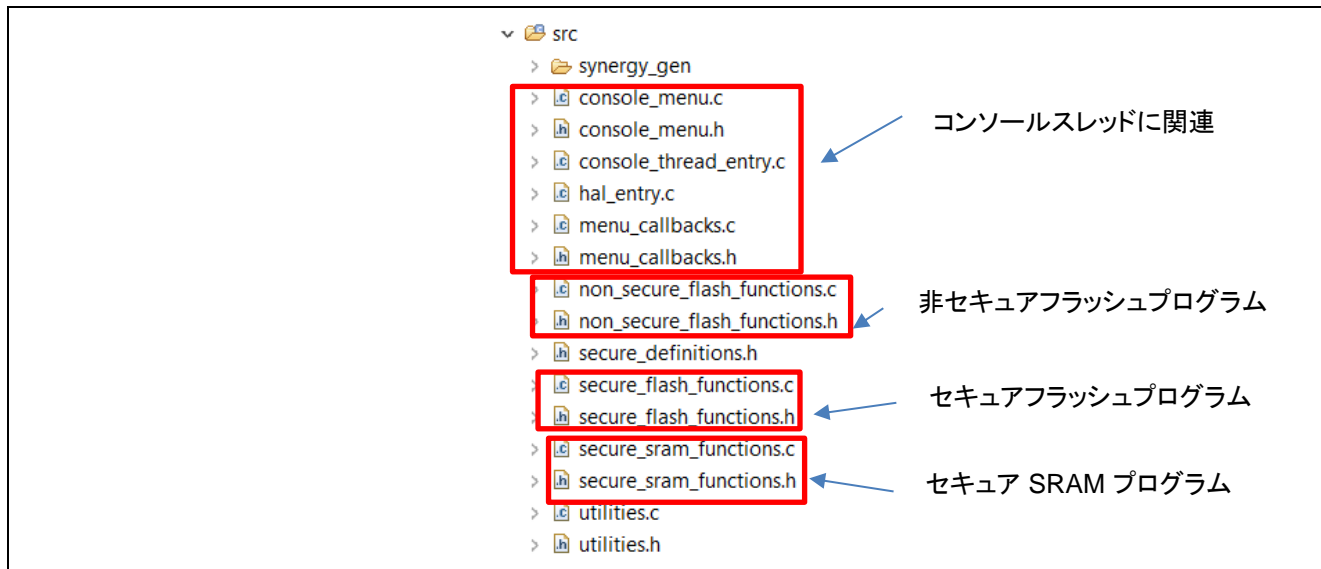


図 24. ソースコード

#### 5.1.4 セキュア SRAM 領域からのソフトウェアの確立と実行 (Establishing and Running Software from Secure SRAM Region)

セキュア SRAM 領域からセキュアプログラムを実行する場合、以下の操作が関係します。

- 5.1.2 章で示したように、リンクスクリプトからセキュア SRAM 領域を設定します
- リンクスクリプト内で、セキュア SRAM アプリケーションコード (.secure\_sram\_program 領域) を、セキュアフラッシュ領域からセキュア SRAM 領域に再配置します

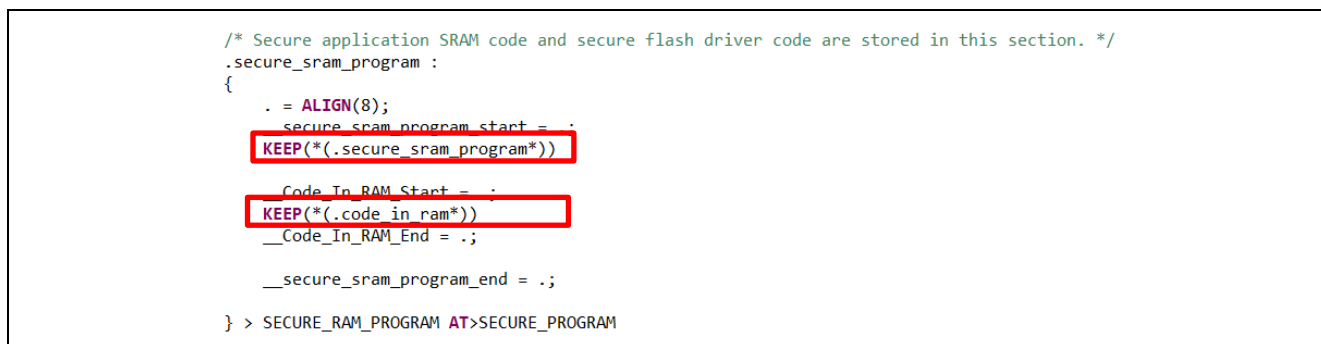


図 25. セキュア SRAM へのコードの割り当て

- アプリケーションコードを使用して、セキュアフラッシュ領域から、セキュア SRAM 領域にセキュア SRAM コードをコピーします。実施については、secure\_flash\_functions.c 内の secure\_sram\_section\_copy 関数を参照してください。
  - .code\_in\_ram という SSP コードセクションは、フラッシュ書き込みルーチンを RAM に再配置します。このアプリケーションノートで、上記のようにこのコードセクションをセキュア SRAM セクション内に割り当てました。

#### 5.1.5 プロジェクトのインポートとビルド (Importing and Building the Project)

secure\_data\_at\_rest\_pk\_s5d9 プロジェクトを e<sup>2</sup> studio にインポートし、このプロジェクトをビルドおよび実行する方法については、このパッケージに付属している『Renesas Synergy™ Project Import Guide』(Renesas Synergy™ プロジェクトインポートガイド) (r11an0023eu0121-synergy-ssp-import-guide.pdf) を参照してください。

このプロジェクトは、USB CDC ドライバである、<https://www.renesas.com/jp/ja/products/synergy/software/add-ons/usb-cdc-drivers.html> を想定しています。このドライバをダウンロードし、インストールしてください。



### 5.1.6 ハードウェアのセットアップ (Hardware Setup)

1. micro USB ケーブルを使用して J5 を PC に接続すると、オンボードデバッガを使用して電力とデバッグ機能を提供できます。
2. micro USB ケーブルを使用して J19 を PC に接続すると、Tera Term などのターミナルプログラムとの通信を実現できます

### 5.1.7 セキュア機能の検証 (Verifying the Secure Functionalities)

ターミナルプログラムを開きます。以下では、例として Tera Term を使用しています。Synergy USB の通信ポートに接続します。

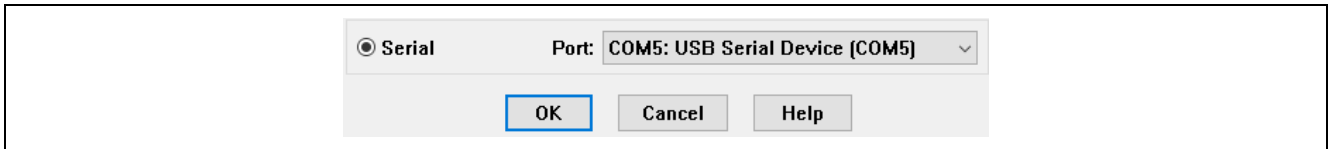


図 26.USB CDC コンソール経由での通信

ステップ 1: 使用可能なメインメニュー項目を表示します。



図 27.メインメニュー

ステップ 2: セキュリティ MPU と FAW のデフォルト設定を表示します (read\_secure\_settings 関数がこのタスクを実行します)。

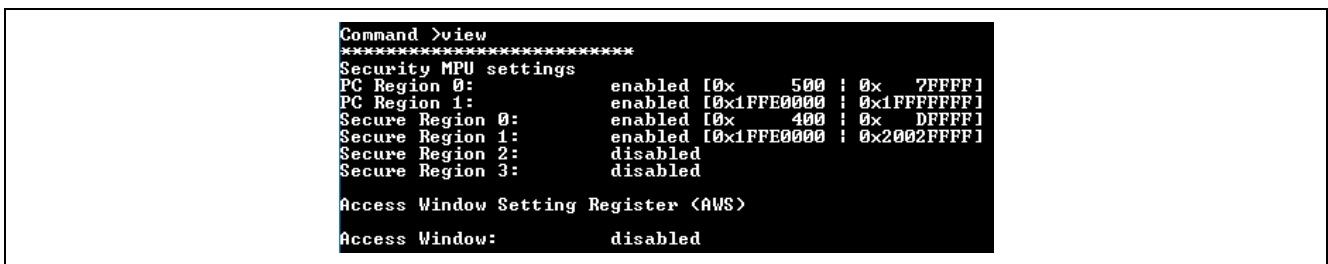


図 28.セキュリティ設定

ステップ 3: セキュアフラッシュ関数 (セキュアフラッシュ領域内で動作する関数) を表示します。これらのセキュアフラッシュ関数を使用して、さまざまなメモリ領域の読み取りと書き込みを行います。

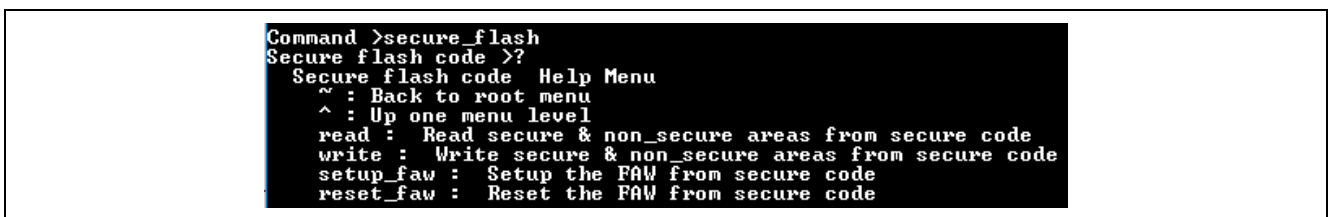


図 29.セキュアフラッシュプログラムの動作に関するメニュー

セキュアフラッシュ関数を使用して、セキュアフラッシュ領域、セキュア SRAM 領域、非セキュアフラッシュ領域、非セキュア SRAM 領域からの読み取りを行い、これらの読み取りアクセスが許可されていることを確認します。secure\_code\_read() 関数は、これらの機能を検証します。

```
Command >secure_flash
Secure flash code >read
*****

Secure code reads...running from secure flash

PASS! secure flash program can read secure flash
PASS! secure flash program can read secure ram
PASS! secure flash program can read non-secure flash
PASS! secure flash program can read non-secure ram
```

図 30.セキュアフラッシュプログラムによる読み取り

セキュアフラッシュ関数を使用して、セキュア SRAM 領域と、非セキュア SRAM 領域への書き込みを行い、これらの書き込みアクセスが許可されていることを確認します。secure\_code\_write() 関数は、これらの機能を実行します。

```
Secure flash code >write
*****

Secure code writes...running from secure flash

PASS! secure flash program can write secure sram data
PASS! secure flash program can write non-secure sram data
```

図 31.セキュアフラッシュプログラムによる書き込み

setup\_faw コマンドを使用して、FAW を設定します。図 32 に、サンプルプロジェクトで設定したデフォルトの FAW 領域を示します。メインメニューに戻り、FAW が設定されていることを確認します。

```
Secure flash code >setup_faw
*****

Setup Flash Access Window...running from flash

PASS secure program => faw is setup from 0x100000 to 0x1FFFFFF

*****
Secure flash code >~
Command >view
*****

Security MPU settings
PC Region 0:          enabled [0x   500 : 0x   7FFFF]
PC Region 1:          enabled [0x1FFE0000 : 0x1FFFFFFF]
Secure Region 0:      enabled [0x   400 : 0x   DFFFF]
Secure Region 1:      enabled [0x1FFE0000 : 0x2002FFFF]
Secure Region 2:      disabled
Secure Region 3:      disabled

Access Window Setting Register (AWS)
Access Window:        enabled [0x 100000 : 0x 200000]
```

図 32.FAW の設定 (0x100000 ~ 0x1FFFFFF)

セキュアフラッシュプログラムから `reset_faw` コマンドを使用して FAW をリセットします。メインメニューに戻り、FAW 領域が無効になっていることを確認します。

```

Command >secure_flash
Secure flash code >reset_faw
*****

Reset Flash Access Window...running from flash

PASS secure program => faw is reset

*****
Secure flash code >~
Command >view
*****
Security MPU settings
PC Region 0:      enabled [0x   500 : 0x  7FFFF]
PC Region 1:      enabled [0x1FFE0000 : 0x1FFFFFFF]
Secure Region 0:  enabled [0x   400 : 0x  DFFFF]
Secure Region 1:  enabled [0x1FFE0000 : 0x2002FFFF]
Secure Region 2:  disabled
Secure Region 3:  disabled

Access Window Setting Register <AWS>
Access Window:    disabled

```

図 33.FAW のリセット

**ステップ 4:**セキュア SRAM 関数 (セキュア SRAM 領域の外部で動作する関数) を表示します。これらのセキュア SRAM 関数を使用して、さまざまなメモリ領域の読み取りと書き込みを行います。

```

Command >secure_sram
Secure sram code >?
Secure sram code Help Menu
~ : Back to root menu
^ : Up one menu level
read : Read secure & non_secure areas from secure sram code
write : Write secure & non_secure areas from secure sram code

```

図 34.セキュア SRAM プログラム

セキュア SRAM 関数を使用して、セキュアフラッシュ領域、セキュア SRAM 領域、非セキュアフラッシュ領域、非セキュア SRAM 領域からの読み取りを行い、これらの読み取りアクセスが許可されていることを確認します。`secure_sram_code_read` 関数は、これらの機能を実行します。

```

Secure sram code >read
*****

Secure sram code reads...running from secure sram

PASS! secure sram program can read secure flash
PASS! secure sram program can read secure ram
PASS! secure sram program can read non-secure flash
PASS! secure sram program can read non-secure ram

```

図 35.セキュア SRAM プログラムによる読み取り

0x100000 ~ 0x1FFFFFF の FAW 領域を有効にし、セキュア SRAM 関数を使用して、セキュアフラッシュ領域、セキュア SRAM 領域、非セキュアフラッシュ領域、非セキュア SRAM 領域への書き込みを行います。secure\_sram\_code\_write 関数は、これらの機能を実行します。

```
Secure sram code >write
*****
Secure sram code writes...running from secure sram
Access Window:      enabled [0x 100000 ! 0x 200000]
PASS! secure sram program can write secure sram data
PASS! secure sram program can write non-secure sram data
PASS! secure sram program cannot write to secure flash region 0xb0000
      which is unmodifiable by FAW
PASS! secure sram program can write to non-secure flash 0x120000
      which is modifiable by FAW
```

図 36 セキュア SRAM プログラムによる書き込み

メイン CLI メニューに戻ります。

```
*****
Secure sram code >~
Command >|
```

図 37.メインメニューに戻る

**ステップ 5:** 非セキュアフラッシュ関数（セキュアフラッシュ領域の外部で動作する関数）を表示します。これらの非セキュアフラッシュ関数を使用して、さまざまなメモリ領域の読み取りと書き込みを行います。

```
Command >non_secure_flash
Non-secure flash code >?
Non-secure flash code Help Menu
~ : Back to root menu
^ : Up one menu level
read : Read secure & non_secure areas from non_secure code
write : Write secure & non_secure areas from non_secure code
```

図 38.非セキュアフラッシュプログラムの動作

非セキュアフラッシュ関数を使用して、セキュアフラッシュ領域、セキュア SRAM 領域、非セキュアフラッシュ領域、非セキュア SRAM 領域からの読み取りを行い、セキュアメモリの読み取りが有効ではないことを確認します。non\_secure\_code\_read 関数は、これらの機能を実行します。

```
Non-secure flash code >read
*****
non-secure code reads...running from non-secure flash
PASS! non-secure program cannot read secure flash
PASS! non-secure program cannot read secure sram
PASS! non-secure program can read non-secure flash
PASS! non-secure program can read non-secure sram
```

図 39.非セキュアフラッシュプログラムからの読み取り動作

0x100000 ~ 0x1FFFFFF の FAW 領域を有効にし、非セキュアフラッシュプログラムを使用して、セキュアフラッシュ領域、セキュア SRAM 領域、非セキュアフラッシュ領域、非セキュア SRAM 領域への書き込みを行います。非セキュアコードによる書き込みを防止する目的で、FAW がセキュアフラッシュメモリを保護していることが表示されます。non\_secure\_code\_write 関数は、これらの機能を実行します。

```
Non-secure flash code >write
*****
non-secure flash code tests...running from non-secure flash
Access Window:          enabled [0x 100000 : 0x 200000]
PASS! non-secure flash program cannot write secure sram data
PASS! non-secure flash program can write to non-secure sram data
PASS! non-secure flash program cannot write to secure flash region 0xb0000
      which is unmodifiable by FAW
PASS! non-secure flash program can write to non-secure flash region 0x120000
      which is modifiable by FAW
```

図 40.非セキュアフラッシュプログラムからの書き込み動作

ステップ 6: セキュア関数のメニューに戻り、FAW の設定をクリアし、view コマンドを使用して動作を確認します。

```
Secure flash code >reset_faw
*****
Reset Flash Access Window...running from flash
PASS secure program => faw is reset
*****
Secure flash code >~
Command >view
*****
Security MPU settings
PC Region 0:          enabled [0x   500 : 0x  7FFFF]
PC Region 1:          enabled [0x1FFE0000 : 0x1FFFFFFF]
Secure Region 0:      enabled [0x   400 : 0x  DFFFF]
Secure Region 1:      enabled [0x1FFE0000 : 0x2002FFFF]
Secure Region 2:      disabled
Secure Region 3:      disabled

Access Window Setting Register <AWS>
Access Window:        disabled
```

図 41.FAW がリセットされたことを確認



### 5.1.7.1 このアプリケーションノートが提供する機能の概要 (Summary of the functionalities provided in this application project:)

- ・ セキュリティ MPU と FAW の設定を使用し、セキュアフラッシュ領域とセキュア SRAM 領域に対する読み取り保護と書き込み保護を実施します。
- ・ デバッガからセキュアデータを秘匿します。デバッガを使用してセキュアデータ領域のメモリを表示すると、値「0」が表示されます。最初の電源の入れなおしの後、デバッガからの保護が有効になります。

Address	0 - 3	4 - 7	8 - B	C - F
0000000000000400	00000000	00000000	00000000	00000000
0000000000000410	00000000	00000000	00000000	00000000
0000000000000420	00000000	00000000	00000000	00000000
0000000000000430	00000000	00000000	00000000	00000000
0000000000000440	00000000	00000000	00000000	00000000
0000000000000450	00000000	00000000	00000000	00000000
0000000000000460	00000000	00000000	00000000	00000000
0000000000000470	00000000	00000000	00000000	00000000
0000000000000480	00000000	00000000	00000000	00000000
0000000000000490	00000000	00000000	00000000	00000000
00000000000004A0	00000000	00000000	00000000	00000000
00000000000004B0	00000000	00000000	00000000	00000000
00000000000004C0	00000000	00000000	00000000	00000000
00000000000004D0	00000000	00000000	00000000	00000000
00000000000004E0	00000000	00000000	00000000	00000000
00000000000004F0	00000000	00000000	00000000	00000000
0000000000000500	00000000	00000000	00000000	00000000
0000000000000510	00000000	00000000	00000000	00000000
0000000000000520	00000000	00000000	00000000	00000000

図 42.セキュアデータ領域をデバッガから表示することはできない

- ・ セキュア SRAM 領域とセキュアフラッシュ領域に対する読み取り保護と書き込み保護を使用して、セキュアフラッシュ領域とセキュア SRAM 領域を実現する方法に関する計画を確立してください。

### 5.1.8 他の Synergy MCU への移行 (Migrating to other Synergy MCUs)

各 MCU グループと、グループ内の各 MCU によって、内部フラッシュと SRAM のサイズが異なることがあります。このアプリケーションを他の製品に移行するには、以下の手順に従ってください。

1. 実際のメモリサイズに合わせて、BSP でセキュリティ MPU の領域設定を変更します
2. [BSP] タブで定義した新しいセキュアフラッシュ領域とセキュア SRAM 領域に注意して、リンカスクリプトを変更します。
3. アプリケーションコード内で、必要に応じて FAW 領域の範囲を変更します。アプリケーションのコードサイズがかなり小さいので、ほとんどの場合、他のアプリケーションコードを変更する必要はありません。

## 5.2 プロジェクト 2: e<sup>2</sup> studio プロジェクト: セキュリティ MPU と FAW の設定のリセット (Project 2: e<sup>2</sup> studio Project: Reset the Security MPU and FAW setting)

この e<sup>2</sup> studio プロジェクト、reset\_flash\_pk\_s5d9 は、SSP の r.flash\_hp HAL ドライバを使用してセキュリティ MPU と FAW の設定をリセットする例を示します。セキュリティ MPU を有効にしたプロジェクトを更新する場合や、新しいプロジェクトをロードする場合、ユーザが毎回このルーチンを実行することを推奨します。

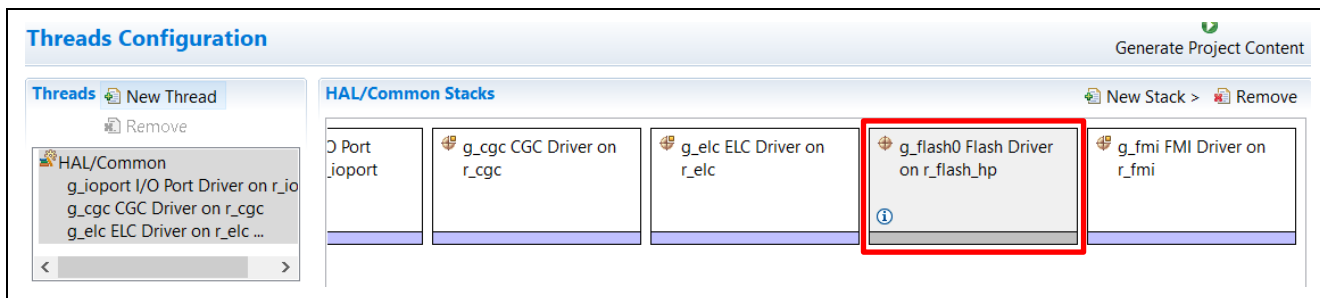


図 43. r\_flash\_hp 上のフラッシュドライバ

フラッシュドライバが構成エリアへの書き込みを実行できるようになる前に、ユーザは DTC/DMAC、EDMAC、GLCDC/DRW(2DG)/JPEG の使用を終了し、図 11.FAW 設定の準備

で説明したようにコードを呼び出す必要があります。これらの準備を終えた後、図 44 に示すルーチン呼び出し、FAW 領域の設定をクリアします。

```
static volatile ssp_err_t err;
SSP_PARAMETER_NOT_USED(err);

//Configure and open the Flash API
err = g_flash0.p_api->open(g_flash0.p_ctrl, g_flash0.p_cfg);
APP_ERR_TRAP(err);

//Remove any prior access window
err = g_flash0.p_api->accessWindowClear(g_flash0.p_ctrl);
APP_ERR_TRAP(err);

//erase SECMPUAC register setting
err = g_flash0.p_api->erase(g_flash0.p_ctrl, secureMPU_reg, 1);
APP_ERR_TRAP(err);

//close flash API
err = g_flash0.p_api->close(g_flash0.p_ctrl);
APP_ERR_TRAP(err);
```

図 44.セキュリティ MPU と FAW の設定のリセット

### 5.3 プロジェクト 3: FAW を永続的にロックするための PC アプリケーション (Project 3: PC Application to Permanently Lock the FAW)

FAW を永続的にロックするための PC 向けユーティリティは、図 12 に示すように、MCU のファクトリブートローダを使用します。PK-S5D9 の場合、ファクトリブートローダモードに移行するには、ジャンパ J1 の位置を 1-2 から 2-3 の位置に変更し、MCU にリセットをかけます。

**通常ブート (Normal Boot):** J1 が位置 1-2 にある場合、S5D9 は内部フラッシュ (ROM) から実行を開始します。これは、既に説明した通常動作モードです。

**ファクトリブートローダモード (Factory Boot Loader Mode):** J1 が位置 2-3 にある場合、S5D9 は USB プログラムモード内のファクトリブートローダを使用してブートします。その結果、USB デバイスインタフェース (CDC クラス) を経由して、プログラムをマイクロコントローラの内部フラッシュに直接ロードすることができます。

このツールを使用するには、以下のステップに従ってください。

**ステップ 1:** デフォルトの評価内容を実行します。これは、デフォルトのプログラムを pk-s5d9 にプログラミング (書き込み) します。¥PC program¥final\_program\_installer\_script の下のデフォルト programInstaller.exe は、FAW をロックしません。このツールは、FAW 領域の設定を行います。

1. Synergy MCU をファクトリブートローダモードに移行します。
2. デバイスマネージャを開き、「Synergy USB Boot」(Synergy USB ブート) に対応する、列挙された COM ポートの番号を見つけます。
3. deploy\_app.bat ファイルを実行し、列挙された COM ポートの番号を引数として指定します。図 45 の出力例を参照してください。

```

C:\ PC program\final_program_installer_script>deploy_app.bat 12

C:\Synergy_SW\iotsg_repository\SSP Applications\Secure Data at Rest\Phase 1\PC
program\final_program_installer_script>REM script to download application and setup FAW
(アプリケーションをダウンロードし、FAW を設定するためのスクリプト)

C:\Synergy_SW\iotsg_repository\SSP Applications\Secure Data at Rest\Phase 1\PC
program\final_program_installer_script>echo off
Parsed SREC file:secure_data_at_rest_pk_s5d9.srec (0x0..0x10D477)
Initialising flash programming connection to port COM12
Established flash programming connection
Disabling flash access window (0x0..0xFFFFFFFF)
Flash access window is now disabled
Erasing flash
Flash erased OK
Writing flash for secure_data_at_rest_pk_s5d9.srec region (bufferOffset=0x0, 0x10D440
bytes)
Flash region written OK
Flash Access Window configured for start=0x100000, end=0x1fffff
Completed OK

```

図 45.FAW 領域をロックせずにアプリケーションをプログラミング (書き込み)

**注記:** アプリケーションを再プログラミング (再書き込み) する前に、リセットルーチンを実行させる必要があります。

**ステップ 2:** 必要な場合、以下の手順に従って FAW 領域のロックに進みます。

1. Visual Studio のプロジェクト、program\_installer.sln  
(%PCprogram%final\_program\_installer\_src%pc%apps%programInstaller%programInstaller.sln) を開きます。
2. プロジェクトのプロパティを開きます。[C/C++] > [Preprocessor] (プリプロセッサ) の下で、  
「ACTIVATE\_THE\_FSPR」という [Preprocessor Definition] (プリプロセッサ定義) を追加します。

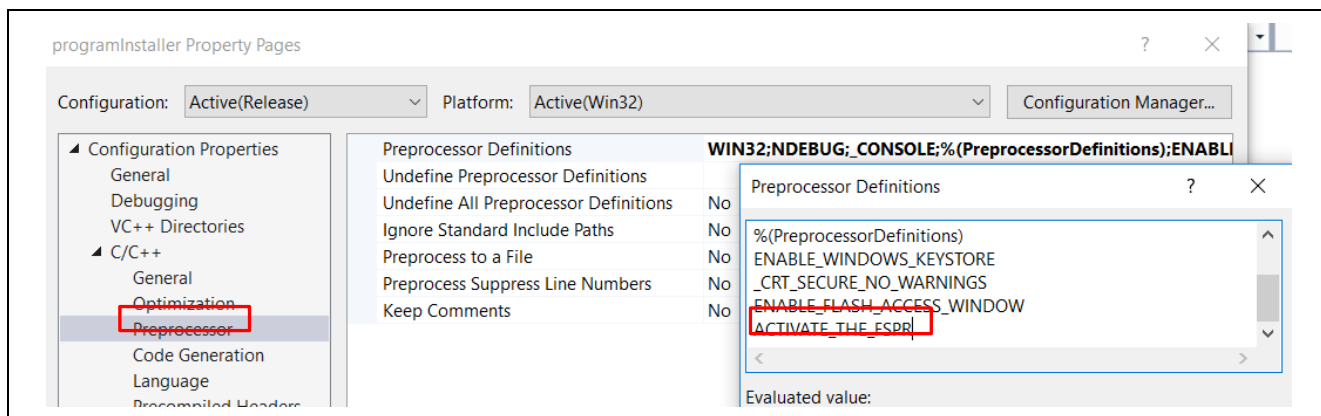


図 46.FAW ロックの設定

3. Program\_Installer プロジェクトをコンパイルします。  
Visual Studio 2017 の Professional または Enterprise バージョンを使用している場合、次のエラーメッセージが表示されます。

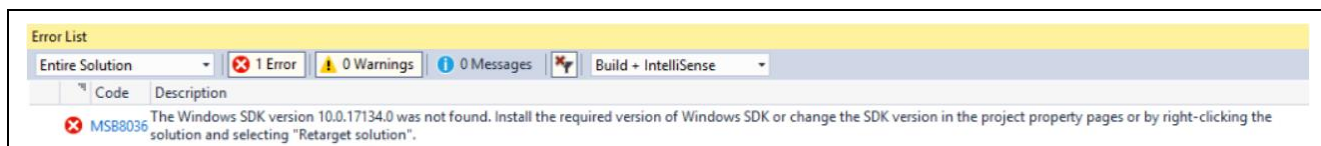


図 47.プログラムインストーラのコンパイル

このエラーメッセージを解決するには、プロジェクトを右クリックし、[Retarget Projects] (プロジェクトの再ターゲット) を選択します。次に、使用している SDK のバージョンを選択します。コンパイルに進みます。

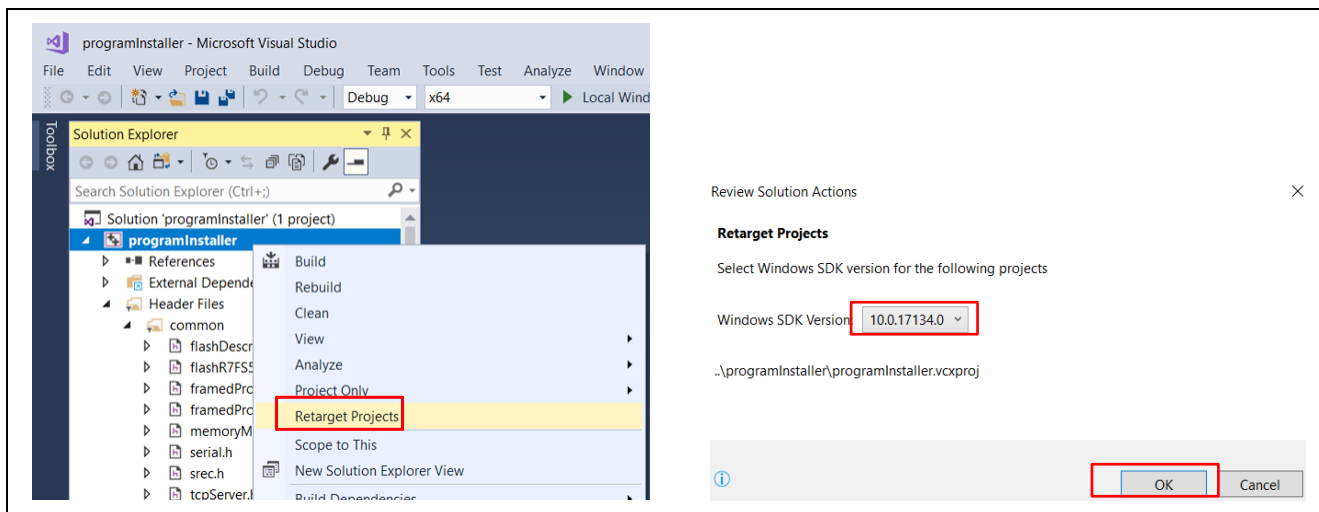


図 48.Zip ファイルのコンテンツ

4. ID コード領域を削除したうえで、アプリケーションノート 1、つまり secure\_data\_at\_rest\_pk\_s5d9 をコンパイルします。これは、このツールに課される制限です。secure\_data\_at\_rest\_pk\_s5d9 に変更を加えない場合、スクリプトフォルダに含まれるデフォルトの .srec ファイルのコピーがあるため、このステップを実行する必要はありません。
5. 生成した secure\_data\_at\_rest\_pk\_s5d9.srec ファイルを、deploy\_app.bat ファイル (¥PCprogram¥final\_program\_installer\_script) と同じフォルダ内に配置します。
6. この章のステップ 1 に掲載した 1 ~ 3 と同じステップを実行して先に進み、最終アプリケーションをプログラミング (書き込み) します。

**警告:** 変更不可能なフラッシュ領域が、現在のイメージ (0x0 ~ 0xFFFFF) で永続的にロックダウンされます。

```
Writing flash for secure_data_at_rest_pk_s5d9.srec region (bufferOffset=0x0, 0x10D440 bytes)
Flash region written OK
Flash Access Window configured for start=0x100000, end=0x1ffffff
Flash access window is now LOCKED
Completed OK
```

図 49.FAW 領域のロック

### 5.4 セキュリティ MPU と FAW における リセット J-Link スクリプトの例 (Example Reset J-Link Script for the Security MPU and FAW)

Reset\_Scripts.zip を解凍し、セキュリティ MPU と FAW 領域のリセット J-Link スクリプト (SMPU\_AWS\_Reset\_Script.zip) を表示します。FSPR ビットをクリアして 0 に設定する (つまり、FAW 領域を永続的にロックする) 前に、このスクリプトを使用してセキュリティ MPU と FAW の設定をリセットすることもできます。SMPU\_AWS\_Reset\_J-Link\_Script.zip を解凍し、図 50 に示す 2 つのファイルを取得します。

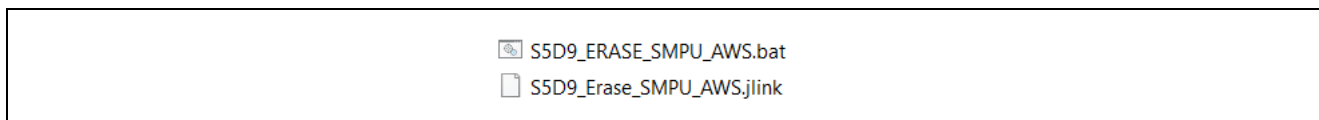
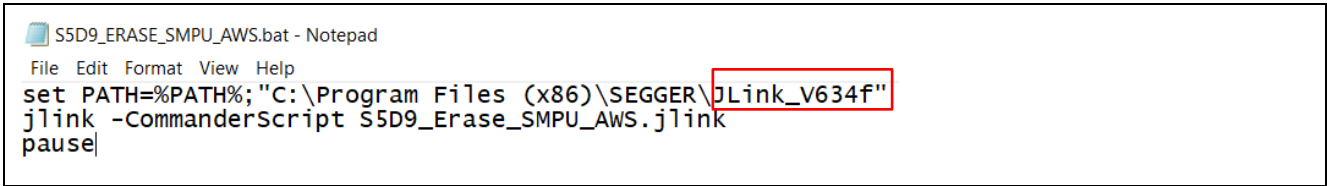


図 50.セキュリティ MPU と FAW のリセットスクリプト

S5D9\_ERASE\_SMPU\_AWS.bat を開き、ユーザの J-Link バージョンに合わせて Segger J-Link バージョンを更新します。



```

S5D9_ERASE_SMPU_AWS.bat - Notepad
File Edit Format View Help
set PATH=%PATH%;"C:\Program Files (x86)\SEGGER\JLink_V634f"
jlink -CommanderScript S5D9_Erase_SMPU_AWS.jlink
pause
  
```

図 51.Segger J-Link バージョンの更新

このスクリプトは、以下の機能を実行します。

1. デバイスをリセットします
2. フラッシュアクセスウィンドウの構成エリア (0x0100A160) すべてを FF に再プログラミング (再書き込み) します
3. デバイスをリセットします
4. コードフラッシュの最初のブロックを消去します
5. デバイスをリセットします

このリセットスクリプトを実行するには、S5D9 を Normal Boot (通常ブート) モードに設定し、コマンドラインウィンドウを開き、この .bat ファイルの配置先を参照し、S5d9\_ERASE\_SMPU\_AWS.bat を実行します。

## 5.5 S5D9 で OSIS の ID コードをリセットする J-Link スクリプト (J-Link Scripts for Resetting OSIS ID code for S5D9)

ソースコード Reset\_Scripts.zip を解凍し、OSIS\_ID\_Resetting\_Script.zip を取り出します。OSIS\_ID\_Resetting\_Script.zip を解凍し、OSIS の ID をリセットするための以下のスクリプトを取得します。

- ・ ID\_Code\_ERASEALL\_S5D9.bat
- ・ IDCode\_ALeRASE\_RV40\_ph2.jlinkscript
- ・ S5D9\_connect.jlink

これらのリセットスクリプトを実行するには、以下のステップに従ってください。

ステップ 1: Segger ツールをまだインストールしていない場合、<https://www.segger.com/downloads/J-Link/> にアクセスし、最新版の J-Link ソフトウェアをダウンロードしてインストールします。

ステップ 2: 新しくインストールした J-LinkARM.dll を

[e<sup>2</sup> studio のインストール先フォルダ]¥DebugComp¥Synergy¥ARM¥Segger にコピーします

**注記:** IDE を使用して OSIS の ID を設定するには、J-Link バージョン V6.34 またはそれ以降が必要です。

ステップ 3: e<sup>2</sup> studio を使用して、blinky プロジェクトを作成します。

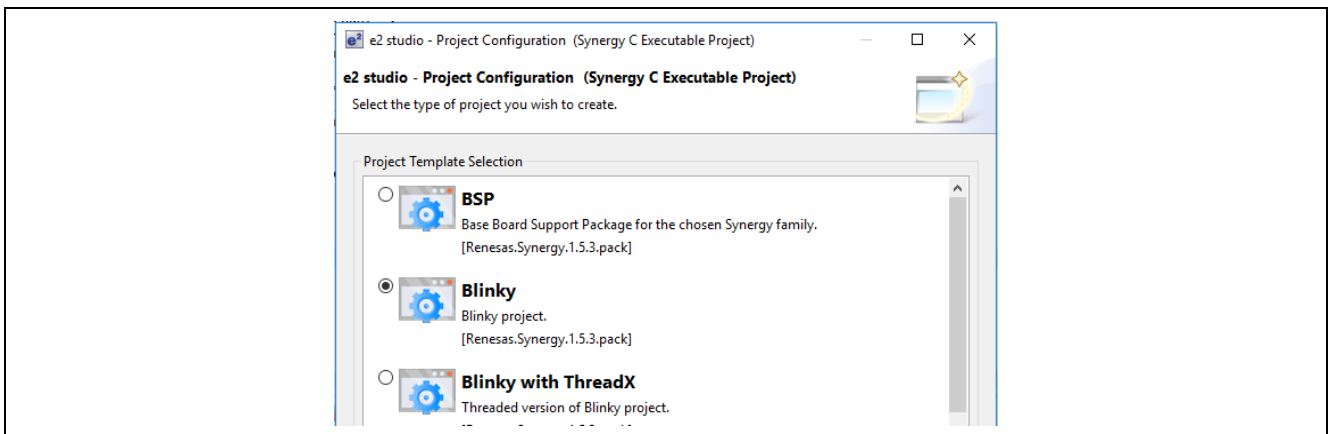


図 52.blinky プロジェクトの作成



ステップ 4: [BSP] タブの下にある [Property] (プロパティ) フィールドで、ID コードを設定します。ビット 127 とビット 126 が両方とも 1 に設定するように注意してください。

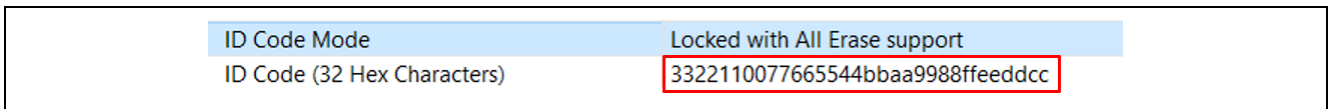


図 53.コンフィギュレータの ID コードの設定

ステップ 5: [Generate Project Content] (プロジェクトコンテンツの生成) をクリックし、プロジェクトをコンパイルします。

ステップ 6: [debug configuration] (デバッグ構成) の [Debugger] (デバッガ) → [Connection Settings] (接続文字列) を開き、以下のように OSIS の ID を入力します。[Apply] (適用) → [Debug] (デバッグ) をクリックします。

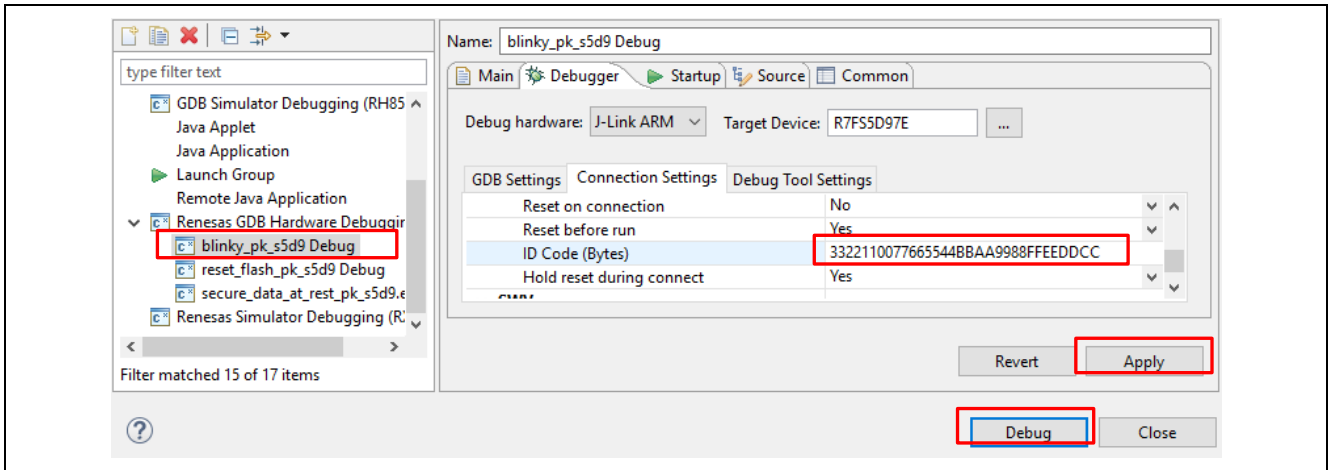


図 54.デバッグ構成の中で、一致する ID コードを設定

ステップ 7: プロジェクトをデバッグします。

ステップ 8: プロジェクトを停止します。

ステップ 9: コマンドプロンプトを開き、上記のスクリプトファイルを展開したフォルダに移動します。

ステップ 10: 既にインストールした J-Link のバージョンに対応するパスを設定します。

```
set PATH=%PATH%;"C:¥Program Files (x86)¥SEGGER¥JLink_V641a"
```

ステップ 11: バッチファイル ID\_Code\_ERASEALL\_S5D9.bat を実行します。以下に示すような出力が表示されます。

```
C:\MyProjects\Data_At_Rest\delivery_01_17\OSIS_ID_Resetting_Script>jlink -JLinkScriptFile IDCode_ALeRASE_RV40_ph2.Jli
nkScript -CommanderScript S5D9_connect.jlink
SEGGER J-Link Commander V6.41a (Compiled Nov 27 2018 14:10:26)
DLL version V6.41a, compiled Nov 27 2018 14:09:51

J-Link Command File read successfully.
Processing script file...

J-Link connection not established yet but required for command.
Connecting to J-Link via USB...O.K.
Firmware: J-Link OB RX621-ARM-SWD V1 compiled Mar 8 2017 13:46:30
Hardware version: V2.10
S/N: 708005016
VTref=3.300V

Selecting SWD as current target interface.

Selecting 4000 kHz as target interface speed

Device "R7FS5D97C" selected.

Connecting to target via SWD
InitTarget() start

Synergy ALeRASE J-Link script InitTarget

ALeRASE CMD
SWO:

DAP-CtrlStat: 0x00000040
DAP-CtrlStat: 0xF0000040
DAP-CtrlStat: 0xC0000040
RESET assert
RESET de-assert
SWO:

DAP-CtrlStat: 0xF0000040
MCUSTAT: 0x00000000
MCUSTAT: 0x00000002
RESET assert
RESET de-assert
DAP-CtrlStat: 0xF0000040
DAP-CtrlStat: 0xF0000040
MCUSTAT: 0x00000001
InitTarget() end
Found SW-DP with ID 0x5BA02477
Scanning AP map to find all available APs
AP[2]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24770011)
AP[1]: APB-AP (IDR: 0x44770002)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E00E000, CID: B105E00D, PID: 000BB00C SCS-M7
ROMTbl[0][1]: E001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[0][2]: E002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[0][3]: E000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[0][4]: E004000, CID: B105900D, PID: 000BB9A1 TPIU
ROMTbl[0][5]: E0041000, CID: B105900D, PID: 000BB925 ETM
ROMTbl[0][6]: E0042000, CID: B105900D, PID: 002BB908 CSTF
ROMTbl[0][7]: E0043000, CID: B105900D, PID: 001BB961 TMC
ROMTbl[0][8]: E0044000, CID: B105F00D, PID: 001BB101 TSG

Synergy ALeRASE J-Link script SetupTarget

MCUSTAT: 0x00000001
MCUCTRL: 0x00000000
SYOCDR: 0x00000080
Memory Address 0: 0xFFFFFFFF
OSIS 1: 0xFFFFFFFF
OSIS 2: 0xFFFFFFFF
OSIS 3: 0xFFFFFFFF
OSIS 4: 0xFFFFFFFF
Cortex-M4 identified.

Script processing completed.
```

図 55.フラッシュ全体の消去と OSIS の ID コードのリセット

ステップ 13: コンフィギュレータを使用し、blinky プロジェクト内で OSIS の ID をすべて FF にリセットして (図 53 参照)、プロジェクトをコンパイルします。

ステップ 14: debugger configuration (デバッガ構成) に対応する OSIS の ID をすべて FF にリセットし (図 54 参照)、デバッグ動作が正常に機能していることを確認します。

## 6. 保存データのセキュア化の次の手順 (Secure Data at Rest Next Steps)

### 6.1 セキュアデータの暗号化と認証 (Secure Data Encryption and Authentication)

このアプリケーションノートの以降のバージョンでは、セキュアデータの暗号化と認証を取り扱います。

### 6.2 外部ストレージの保存データのセキュア化 (External Storage Secure Data at Rest)

このアプリケーションノートの以降のバージョンでは、外部ストレージのデータ暗号化と認証、さらにアクセス制御を取り扱います。

### 6.3 セキュリティ MPU のセキュリティ機能の使用例 (Example using the Security MPU Security Functions)

2.2 章で説明したように、セキュリティ MPU で制御できる 2 つのセキュリティ機能があります。

1 つのセキュリティ機能は、セキュア暗号化エンジン (SCE) へのアクセスを制御します。このセキュリティ機能を有効にした場合、セキュアプログラムのみが SCE にアクセスできます。

もう 1 つのセキュリティ機能は、コードフラッシュとデータフラッシュの消去とプログラミング (書き込み) (E/P) 機能へのアクセスを制御します。このセキュリティ機能を有効にした場合、セキュアプログラムのみが内部フラッシュの E/P を実行できます。

このアプリケーションノートの以降のバージョンでは、これらのセキュリティ機能の使用例を紹介する予定です。

## 7. 参考資料

### 1. MQTT/TLS アプリケーションプロジェクト:

Synergy MQTT/TLS Google Cloud Connectivity Solution - Application Project  
<https://www.renesas.com/jp/ja/software/D6003715.html>

Synergy MQTT/TLS Google クラウド接続ソリューション (参考資料)

<https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0335ju0102-synergy-mqtt-tls-google-cloud-connectivity.pdf>

Synergy MQTT/TLS AWS Cloud Connectivity Solution - Application Project

<https://www.renesas.com/jp/ja/software/D6003098.html>

Synergy MQTT/TLS AWS クラウド接続ソリューション (参考資料)

<https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0336ju0102-synergy-mqtt-tls-aws-cloud-connectivity.pdf>

Synergy MQTT/TLS Azure Cloud Connectivity Solution - Application Project

<https://www.renesas.com/jp/ja/software/D6003716.html>

Synergy MQTT/TLS Azure クラウド接続ソリューション (参考資料)

<https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0337ju0102-synergy-mqtt-tls-azure-cloud-connectivity.pdf>

### 2. セキュアブートマネージャアプリケーションプロジェクト:

<https://www.renesas.com/jp/ja/software/D6003137.html>

### 3. ルネサスフラッシュプログラマ: <https://www.renesas.com/us/en/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html#downloads>

### 4. ARM®v7-M Architecture Reference Manual (ARM DDI 0403D)

### 5. ARM® Cortex®-M4 Processor Technical Reference Manual (ARM DDI 0439D)

ARM® Cortex®-M4 Devices Generic User Guide (ARM DUI 0553A)

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

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019年5月15日	—	初版 英文版(R11AN0359EU0100 Rev.1.00 Jan.31.19)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まれます。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）、交通管制（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なります。
  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)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレスト）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。