

RH850/U2A-EVA Group

OTA(Over the Air)を用いたソフトウェア更新

要旨

本アプリケーションノートでは、ルネサスエレクトロニクスの自動車向けシングルチップマイクロコンピュータ RH850/U2A シリーズ(以降、U2A と称す)における OTA(Over the Air)を用いたソフトウェア更新手順について記載しています。

本アプリケーションノートで扱うソフトウェア更新の手順は、更新プログラムの受信、フラッシュメモリへの消去・書き込み、ソフトウェアの切り替えの手順についてです。

外部デバイスとの通信インタフェースとして、RS-CANFD を使用します。

フラッシュメモリの書き換えプログラムはユーザー領域上にあるものとします。

本アプリケーションノートは、OTA によるソフトウェア更新に求められるセキュリティ対策についてはサポートしません。お客様で実使用に即した、セキュリティ対策をしていただく必要があります。

本資料およびプログラムは、RH850/U2A 搭載機能の理解促進を意図するものであり、量産設計を対象とするものではありません。

また、最新のマニュアル、正誤表、テクニカルアップデートや、開発環境の更新を反映しておりません。該当機能を使用される場合には、本プログラムは参考として扱い、最新のドキュメントや開発環境にて、お客様の責任において行ってください。

対象デバイス

- RH850/U2A-EVA Group

対象統合開発環境

CS+(ルネサスエレクトロニクス社製)

バージョン : V8.07.00

デバイスファイル : DR7F702300.DVF

: DR7F702301.DVF

: DR7F702302.DVF

参考文献

デバイスの機能詳細及び電気的特性に関してはユーザーズマニュアル ハードウェア編に記載します。

本アプリケーションノートは以下のマニュアルを参照し作成しております。

RH850/U2A-EVA Group User's Manual (Rev1.20): R01UH0864EJ0120

RH850/U2A-EVA Group User's Manual Flash Memory User's Manual (Rev1.10): R01UH0832EJ0110

使用ソフトウェア

- Renesas Flash Driver RFD28F V.1.00.01 Free package (サンプルコードに同梱)

RFD についての詳細は弊社窓口(www.renesas.com/contact/)までお問い合わせください。

目次

1. OTA(Over The Air)とは	4
2. RH850/U2A の OTA 関連機能.....	5
2.1 U2A のフラッシュメモリ仕様.....	5
2.1.1 Single Map Mode	6
2.1.2 Double Map Mode.....	7
2.2 アドレス変換機能	8
2.2.1 GCFU によるアドレス変換.....	8
2.2.2 Double Map Mode のアドレス変換.....	9
2.3 FCI 機能	10
2.3.1 BGO(Background operation)機能	10
2.3.2 サスペンド機能	10
3. RH850/U2A を用いた OTA 方式	11
3.1 差分書き換え方式(Single map mode).....	11
3.2 バンク切り替え方式(Double map mode).....	12
3.3 Security 対応の注意事項	13
4. OTA 動作例	14
4.1 システム構成.....	14
4.2 CAN FD 通信について	14
4.2.1 CAN FD 通信仕様.....	14
4.2.2 CAN FD コマンド仕様 (* U2A 視点)	15
4.3 Port(LED)について	15
5. Single map mode による OTA 動作例(GCFU を使用したアドレス変換).....	16
5.1 仕様	16
5.1.1 ソフトウェア仕様	16
5.1.2 動作イメージ.....	17
5.1.3 全体シーケンス	18
5.1.4 使用機能.....	20
5.1.5 動作モード	20
5.1.6 メモリマッピングモード	20
5.1.7 RFD コンフィギュレーション	20
5.2 動作手順	21
5.2.1 ① 起動 Bank 切り替え	21
5.2.2 ② BGO 書き込み	24
5.2.3 ③ ID 認証.....	27
5.2.4 ④ Code Flash 消去	28
5.2.5 ⑤ Code Flash 書き込みデータダウンロード	30
5.2.6 ⑥ Code Flash 書き込み.....	31
5.2.7 ⑦ 起動 Bank 情報書き込み関数	33
5.3 メモリ割り付け	34

6. Double Map mode による OTA 動作例(バンク切替方式)	35
6.1 仕様	35
6.1.1 ソフトウェア仕様	35
6.1.2 動作イメージ	37
6.1.3 全体シーケンス	38
6.1.4 使用機能	41
6.1.5 動作モード	41
6.1.6 メモリマッピング	41
6.1.7 RFD コンフィギュレーション	41
6.2 動作手順	42
6.2.1 ① スタートプログラム	42
6.2.2 ② BGO 書き込み	43
6.2.3 ③ ID 認証	46
6.2.4 ④ Code Flash 消去	47
6.2.5 ⑤ Code Flash 書き込みデータダウンロード	49
6.2.6 ⑥ Code Flash 書き込み	50
6.2.7 ⑦ Hardware Property Area の更新	52
6.2.8 ⑧ Switch Area の更新	57
6.2.9 ⑨ TAG Area の更新	60
6.3 メモリ割り付け	62
6.3.1 アドレス配置図	62
7. ソフトウェアの更新中に処理が中断した時の復帰動作	63

1. OTA(Over The Air)とは

OTA とは無線での通信を経由し、データを送受信することを指し、一般的にはソフトウェアの更新などに用いられる技術です。近年、自動車においても、容易な制御ソフトウェアのアップデート手法として注目されています。OTA によるソフトウェアの更新は車両制御を妨げることなく、フラッシュメモリ領域にソフトウェアを書き込むことが求められます。

U2A はバックグラウンドでソフトウェアのアップデートを行う機能をサポートしています。

ソフトウェアの更新には、いくつかのユースケースが想定されます。図 1-1 に、OTA によるソフトウェア更新のユースケースを示します。

Wait OTA は車両の駐車中に更新プログラムを受信し、フラッシュメモリに書き込む方式です。

Semi-non wait OTA は、MCU の外部に外付けのフラッシュメモリを配置し、アプリケーションソフトウェアの通常動作中に受信したデータを外付けのフラッシュメモリに書き込み、車両の駐車中に MCU のフラッシュメモリを書き換えます。

Non wait OTA は、MCU 内部に独立した複数のフラッシュメモリを搭載し、アプリケーションソフトウェアの通常動作中に受信したデータを書き込み、車両の Key-ON 時にソフトウェアが動作するフラッシュメモリの領域を切り替える方式です。

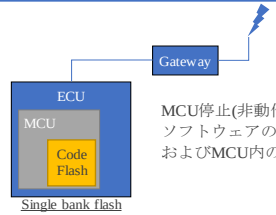
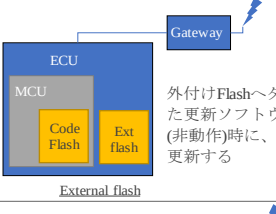
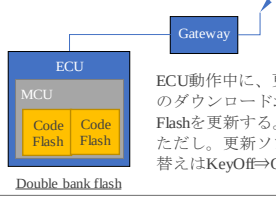
OTA方式	旧プログラムの バックアップ	書き込み失敗時	バックグラウンド 書き込み	MCUへのソフトウェア 書き込みタイミング	イメージ(MCU動作)
Wait OTA	なし	再DL必要	不可	駐車中(key-off)	 Single bank flash
Semi-non wait OTA	なし	再DL不要 (外付けフラッシュの 容量による)	可	駐車中(key-off)	 External flash
Non wait OTA	あり	再DL不要	可	アプリケーション ソフトウェアの 動作中 (Key-on)	 Double bank flash

図 1-1 OTA によるソフトウェア更新のユースケース

U2A は Non wait OTA を実現する、フラッシュメモリのソフトウェア領域を複数搭載する Double map mode をサポートします。また、フラッシュメモリ領域を分割せずに使用する、Single Map mode も搭載します。U2A を Single map mode で使用する場合も、GCFU(Global Calibration Function Unit)を使用することで、バックグラウンドでのソフトウェアのアップデートを実現することができます。

U2A のフラッシュメモリの機能に関する詳細は、[2.1. U2A のフラッシュメモリ仕様](#)を参照ください。

GCFU に関する詳細は、[2.2.1 GCFU によるアドレス変換](#)を参照ください。

2. RH850/U2A の OTA 関連機能

2.1 U2A のフラッシュメモリ仕様

RH850/U2Ax は 3 つの Flash Programming system(FPSYS0, FPSYS1, FPSYS2)^{注1}を搭載しています。各 FPSYS は Flash Memory、Flash sequencer で構成されています。各 Flash sequencer は、Flash Application Command Interface(FACI) を搭載しています。U2A16、U2A8、U2A6 はそれぞれ 16MB、8MB、6MB^{注2}の Code Flash を搭載しています。FPSYS2 は ICUMH 用の Data Flash 領域の FPSYS です。図 2-1 に U2A16 を例に、フラッシュメモリ関連モジュールの構成図を示します。

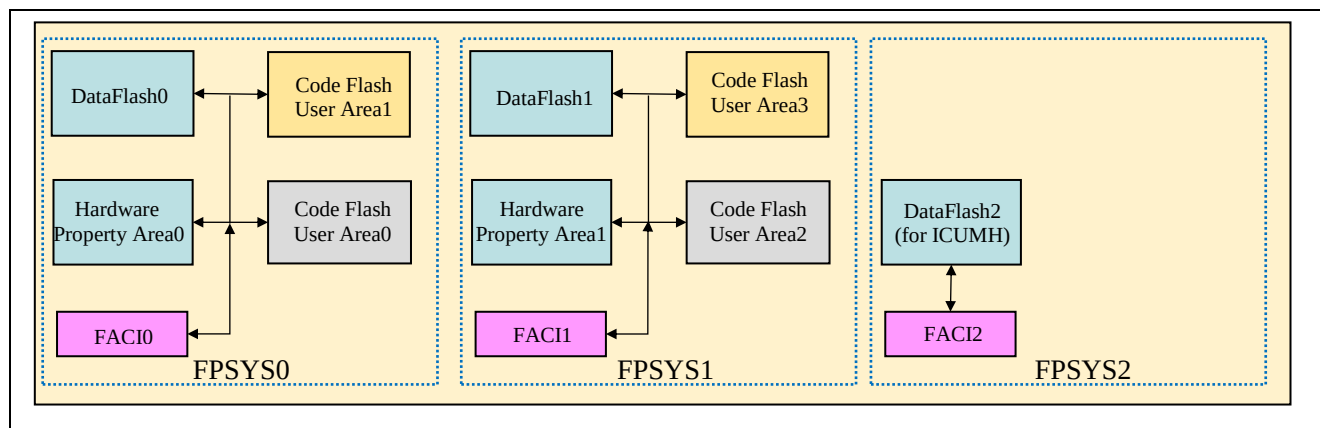


図 2-1 U2A16 のフラッシュメモリ関連モジュールの構成

RH850/U2Ax は 2 種類の Code Flash のマッピングモード(Single map mode/Double Map mode)を搭載しています。Code Flash のマッピングモードは OPBT12.MAPMODE[1:0]で選択することができます。詳細は、RH850/U2A-EVA Group User's Manual :Hardware “51.12.18 OPBT12 — Option Byte 12”を参照ください。

- Single Map Mode

Single map mode では、GCFU による Code Flash の User Area のアドレスのリマップ機能をサポートします。

- Double Map Mode

Double map mode では、Code Flash の User Area のバンクスワップをサポートします。Code Flash の User Area のバンクを 2 つに分割し、片方は有効領域、もう片方は無効領域となります。Code Flash の User Area の有効領域のプログラム実行中に、無効領域への Programming/Erase が可能です。

^{注1} U2A8、U2A6 は FPSYS0,FPSYS2 で構成されています。

^{注2} U2A-EVA は U2A16 mode のときは 16Mbyte, U2A8 mode の 8Mbyte となります。U2A-EVA の場合は U2A16 mode のときは U2A16、U2A8 mode のときは U2A8 にそれぞれ読み替えてください。

2.1.1 Single Map Mode

Single map mode は Code Flash 領域を連続的に配置するモードです。U2A16、U2A8、U2A6 はそれぞれ 16MB、8MB、6MB をソフトウェア動作領域として使用することができます。Single map mode は、部分的なプログラムの更新を行う場合に効率よく Code Flash を使用することができます。図 2-2 に Single map mode 時の Code Flash(User Area)のアドレスマップを示します。

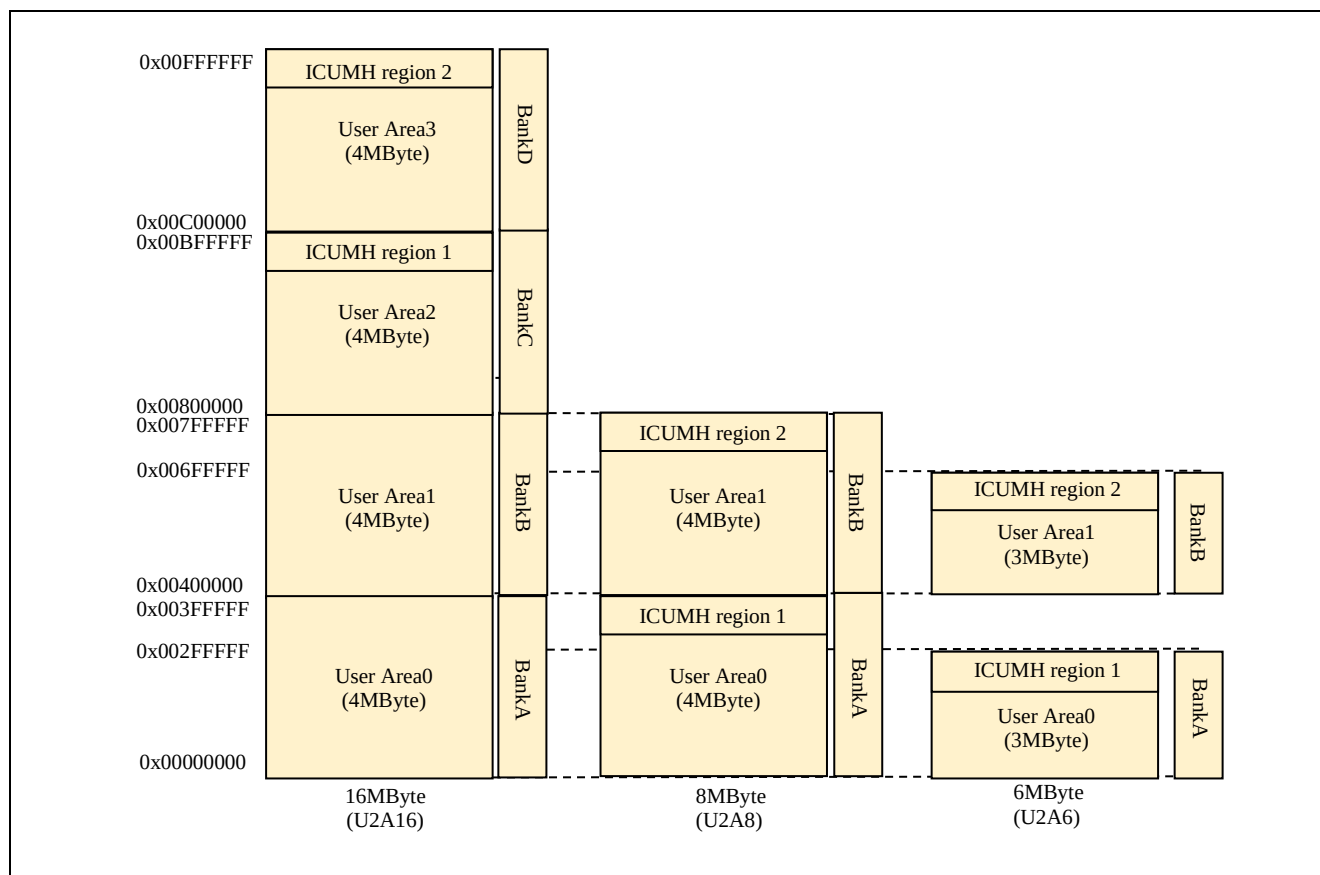


図 2-2 Single map mode の Code Flash (User Area) アドレスマップ^{注1}

^{注1} ICUMH 領域は、オプションバイトにより設定する領域数とアドレスを選択します。ICUMH 無効時は、ICUMH 領域は無効となります。

2.1.2 Double Map Mode

Double map mode は Code Flash 領域を 2 つの領域に分割するモードです。Code Flash は FPSYS 毎に 2 つの領域に分割されてマッピングされており、1 つは有効領域(Valid area)で、もう 1 つは無効領域(Invalid Area)とし、オプションバイトの設定により有効領域と無効領域を入れ替えることができます。U2A16、U2A8 はそれぞれ 8MB、4MB を有効領域として使用することができます。

Double map mode は Code Flash のプログラム全体を更新する場合に適しているモードです。Code Flash を全て二重化しており、有効領域と無効領域のアドレスを入れ替えることで、全てのアクセスマスタから同一のアドレスでアクセスすることができます。プログラム更新後に元のプログラムに戻す場合、無効領域にバックアップデータが存在しているため、容易に元のプログラムに戻すことができます。

Double Map mode 時の Code Flash の有効領域は、OPBT13.DBMAPSW0/1 で選択することができます。有効領域は FPSYS ごとに個別に設定することができます。

それぞれの領域をスワップし有効領域と無効領域を切り替えることにより、有効な領域のアドレスは常に同じアドレスとなります。

有効領域のプログラム実行中に無効領域の書き換えを実行し、有効領域を切り替えることでプログラムの更新を実現します。有効領域の切り替え方法は、[2.2.2 Double Map Mode のアドレス変換](#)を参照ください。

Double map mode 時に無効領域に更新プログラムを書き込む場合、無効領域の全ての領域へ書き込むことを想定しています。

図 2-3 に Double map mode 時の Code Flash(User Area)のアドレスマップを示します。

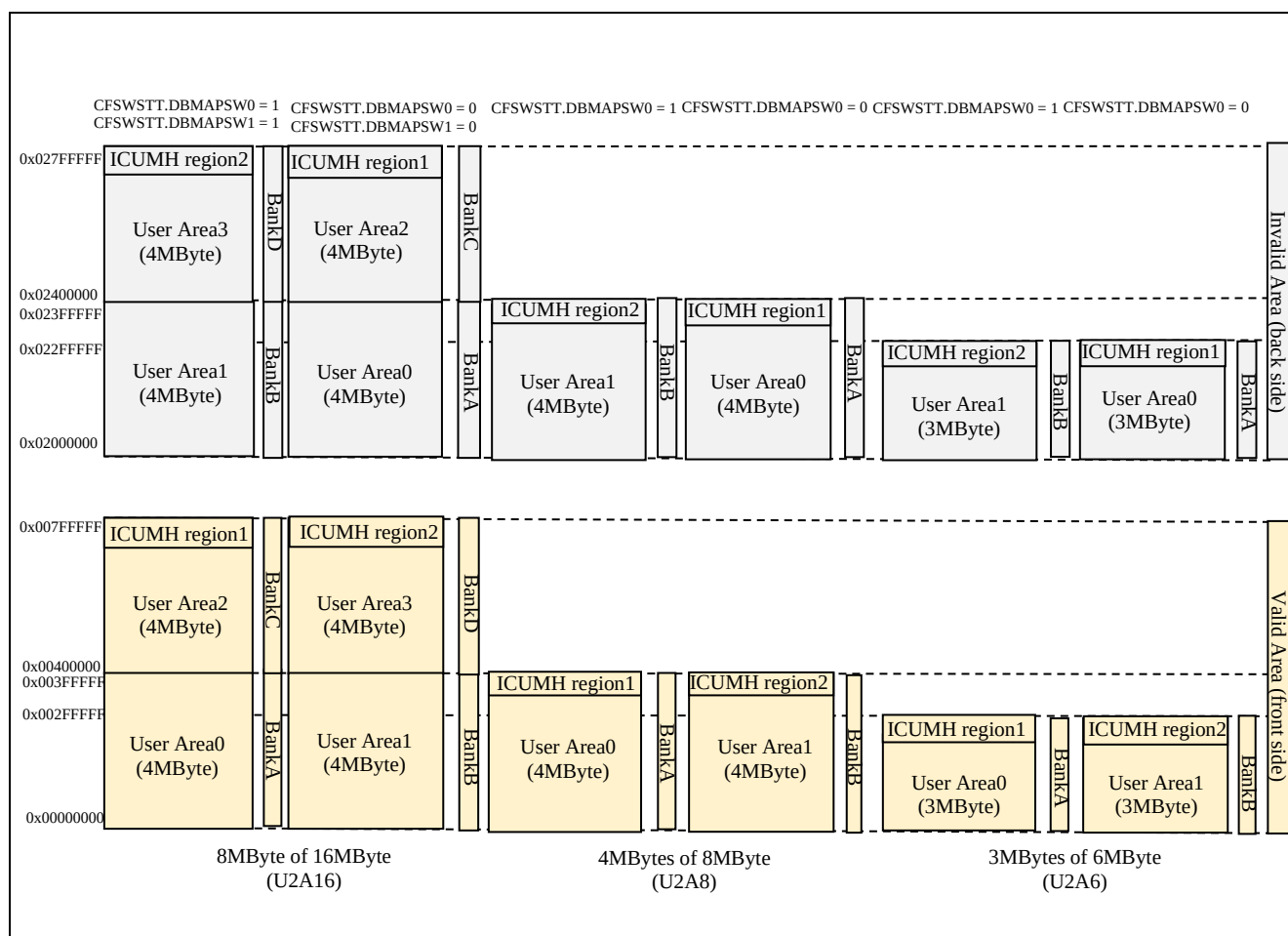


図 2-3 Double map mode の Code Flash(User Area) アドレスマップ^{注1}

^{注1} ICUMH 領域は、オプションバイトにより設定します。ICUMH 無効時は、ICUMH 領域は無効となります。

2.2 アドレス変換機能

2.2.1 GCFU によるアドレス変換

Single Map mode では GCFU によるアドレス変換機能を使用して、更新前・後のプログラムのマッピング切り替えを行います。部分的なプログラムの更新を行う場合に効率よく Code Flash を使用可能です。図 2-4 に GCFU によるアドレス変換を示します。

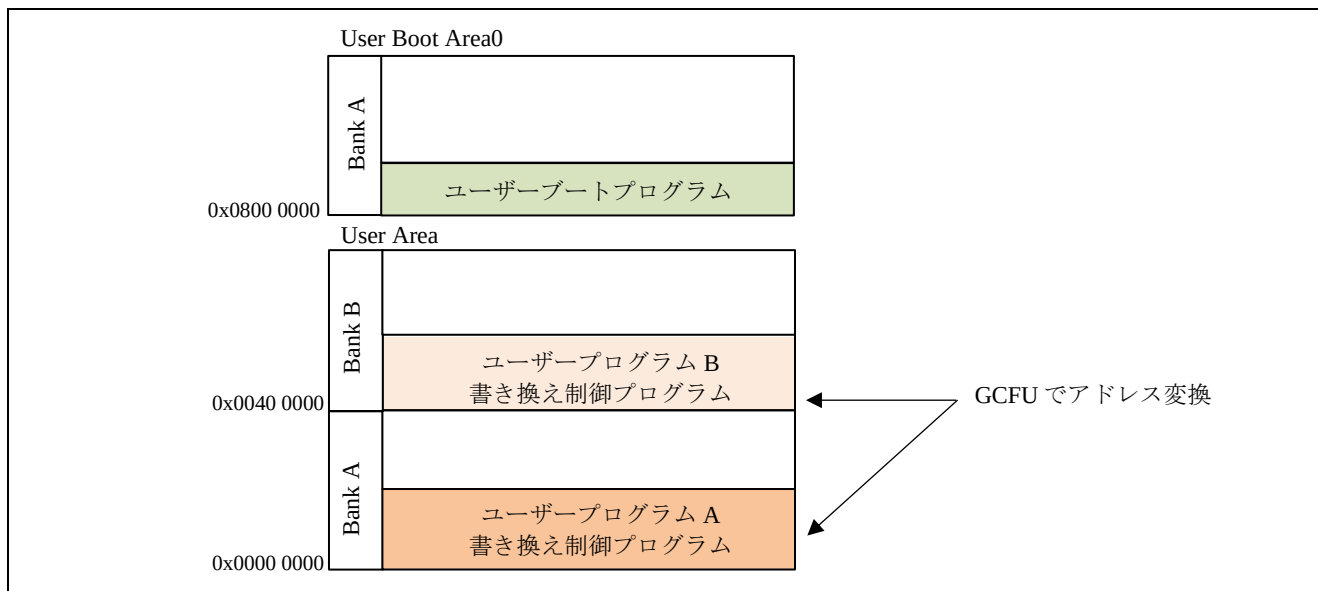


図 2-4 GCFU によるアドレス変換後の Code Flash へのアクセス

CPU からはフラッシュメモリを論理アドレスとしてアクセスし、GCFU によりアクセス先のアドレスを変換します。FACI は GCFU でのアドレス変換に依存しないため、物理アドレスとしてアクセスします。図 2-5 にメモリ関連バス構成図を示します

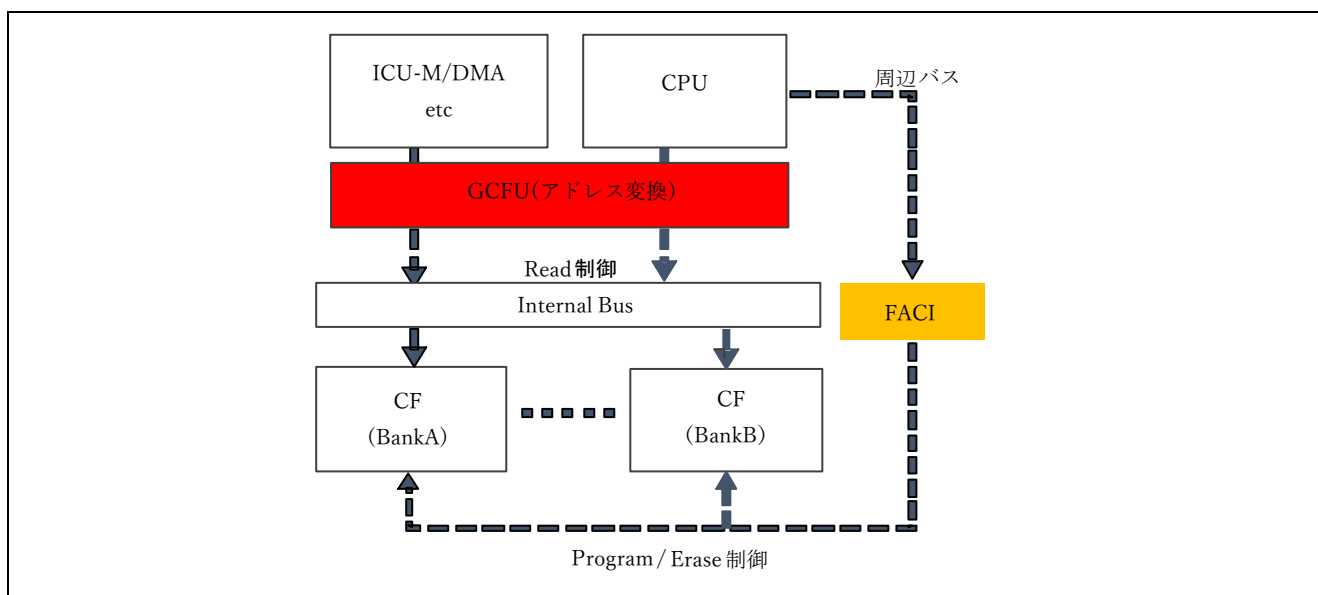


図 2-5 メモリ関連バス構成図

GCFU によるアドレス変換の動作例は 5.1.2 動作イメージを参照ください。

(a) GCFU 使用時の FACI の制約事項

FACI からの Code Flash へのアクセスは、物理アドレスに依存するため、Programing/Erasure 中の Code Flash へのアクセス可否も物理アドレスに依存します。GCFU を使用しアドレス変換した場合も、後述する BGO (Background operation)機能や Multi FPSYS Operation は物理アドレスが基準となるため注意が必要です。詳細 RH850/U2A-EVA Group User's Manual: Hardware Section 51, Flash Memory をご参照ください。

2.2.2 Double Map Mode のアドレス変換

RH850/U2Ax は Data Flash に Double map mode の Code Flash の有効領域設定を含む、ハードウェアのコンフィギュレーション用の領域(Hardware Property Area)を 9 ブロック^{注1}搭載しています。図 2-6 に Hardware Property Area(切り替え関連領域)の概要を示します。

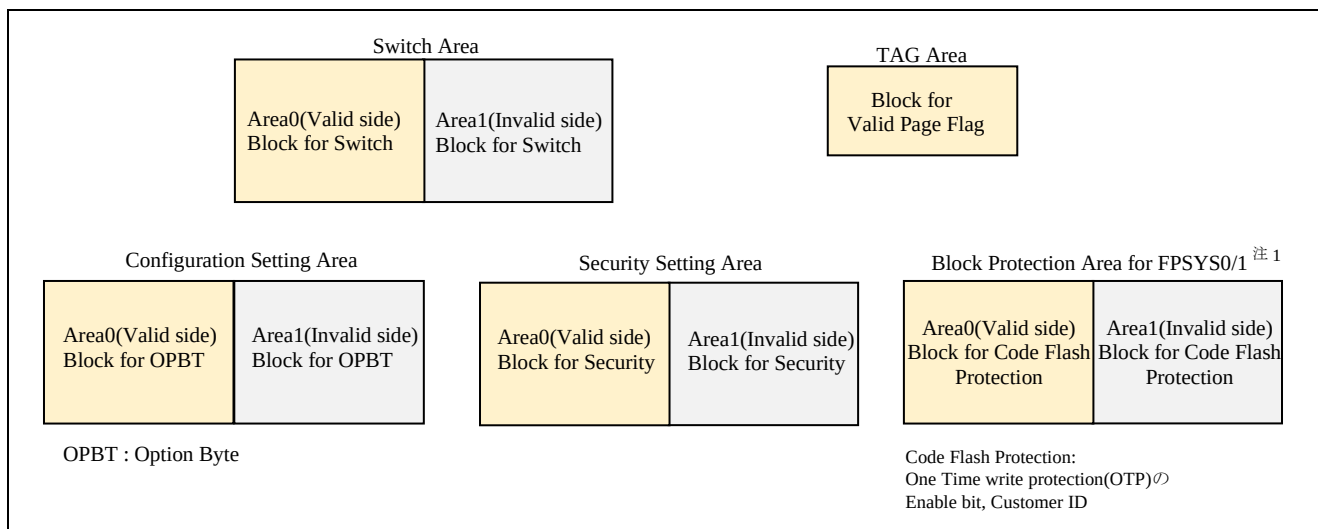


図 2-6 Hardware Property Area(切り替え関連領域)の概要(出荷時)

Configuration Setting Area, Security Setting Area, Block protection Area for FPSYS0/1 ^{注1}はそれぞれ 2 つの領域(Area0/Area1)で構成されており、更新前・更新後の設定を行います。有効領域は front side に、無効領域は back side に配置されます。有効領域は更新できませんが、無効領域は更新可能です。出荷時には Area0 が有効領域として設定されており、Area1 は消去された状態の無効領域です。

Switch Area は 2 つの領域(Area0/Area1)で構成されており、Configuration Setting Area, Security Setting Area, Block protection Area0/1 それぞれの有効領域のスイッチ設定や複数のフラグの領域が配置されています。出荷時には Area0 が有効領域として設定されており、Area1 は無効領域です。

TAG Area は Switch Area の有効領域の設定及びステータスフラグ用の領域です。図 2-7 に Hardware Property Area の切り替えの詳細を示します。

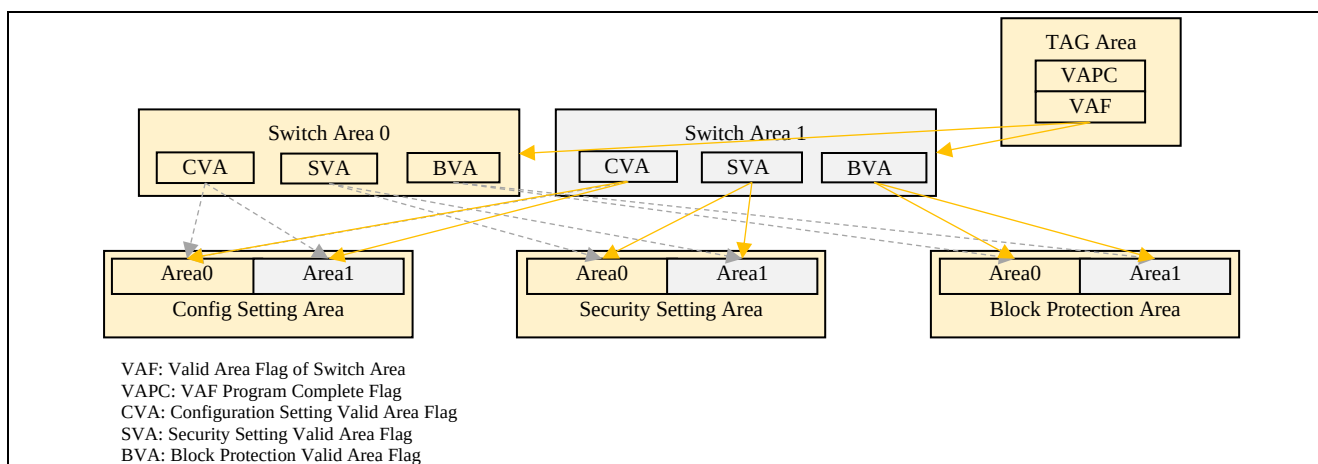


図 2-7 Hardware Property Area 切り替えコンセプト

Switch Area0/1 の有効設定は TAG Area の Valid Area Flag(VAF)によって設定されます。各ハードウェアのコンフィギュレーション設定領域(Configuration Setting Area0/1, Security Setting Area0/1, Block protection Area)は、Switch Area の Valid Area Flag(CVA, SVA, BVA)によって設定されます。各領域は無効領域(Invalid side)のみ、Program/Erase が可能です。

^{注1} Block protection Area for FPSYS1 は U2A16 のみに搭載されている領域です。

2.3 FACI 機能

2.3.1 BGO(Background operation)機能

BGO 機能として、Code Flash の Programing/Erasure 中のサスペンド機能及び、Multi FPSYS Operation をサポートします。

同一 FPSYS 内での Programing/Erasure 中に読み出しが可能なフラッシュメモリは以下の通りです。

- Data Flash の Programing/Erasure の Code Flash の読み出し
- Code Flash の片方の Bank の Programing/Erasure 中の、他方の Bank の読み出し
- Data Flash の Programing/Erasure 中、他の Data Flash 領域の読み出し
- Code Flash の Programing/Erasure 中に Data Flash の読み出しが可能です。

Multi FPSYS Operation

- 異なる FPSYS のフラッシュメモリは同時に、Programing/Erasure が可能です。

図 2-8 にフラッシュメモリ関連モジュール構成図を示します。

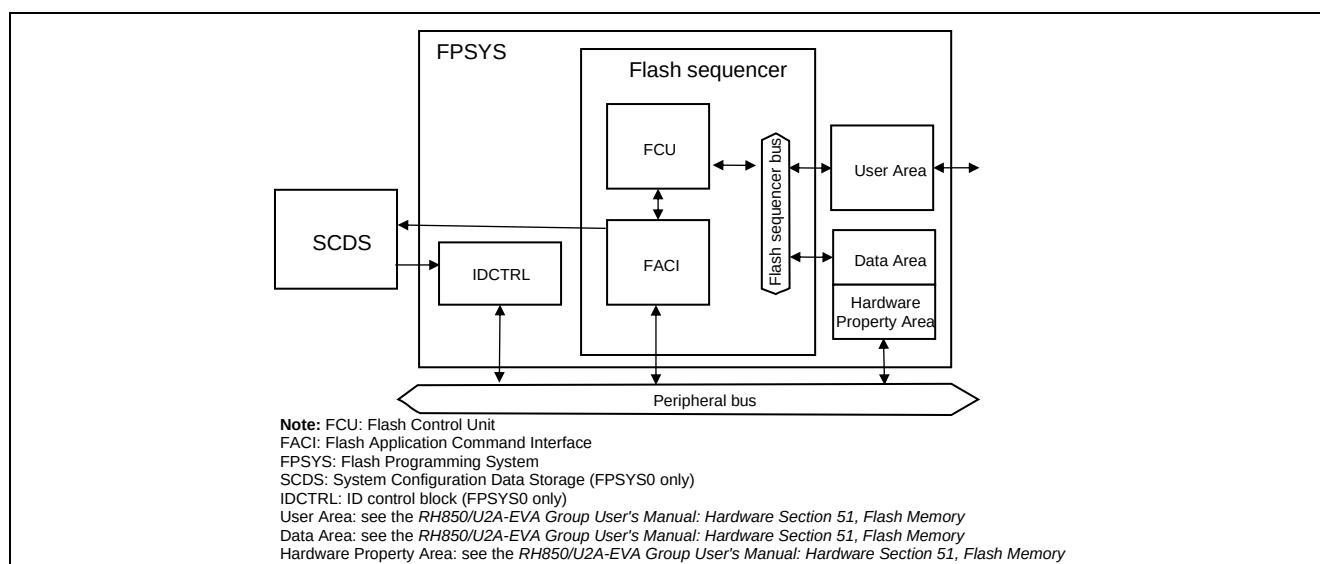


図 2-8 フラッシュメモリ関連モジュール構成図

2.3.2 サスペンド機能

OTA による Code Flash の書き換えは他のアプリケーションのタスクに比べ低優先タスクとして処理されることを想定し、Code Flash 書き換え中に高優先のタスクの割り込みに対応するため、Suspend 機能を備えています。

本アプリケーションノート及びサンプルソフトは、Code Flash 書き換え中のサスペンドはサポートしません。お客様のアプリケーションソフトウェア上で Code Flash を書き換える場合は、Code Flash 書き換え中に他のタスクでフラッシュメモリの書き込み/消去を行う場合は、Code Flash の書き換えをサスペンドし、タスクの処理を行ってください。

3. RH850/U2A を用いた OTA 方式

3.1 差分書き換え方式(Single map mode)

差分コード書き換え方式のイメージを図 3-1 に示します。本節では Single map mode 時の同一 FPSYS 内の Code Flash の下部の Bank を Lower 領域、上部の Bank を Upper 領域として説明します。フラッシュメモリ書き換えプログラム(リプログコード)を配置する領域を Lower、更新するプログラムを配置する領域を Upper とした場合の差分コード書き換えについて説明します。

- ① 更新前
Upper(更新可能)領域のアプリケーション・ソフトウェアへのプログラムフェッチを禁止します。
- ② 更新中
Lower(更新不可)領域のリプログ・コードで、Upper(更新可能)領域のアプリケーション・ソフトウェアの差分コードを書き換えます。
- ③ 更新後
Reset 解除後に、GCFU でアドレスマップを切り替え、更新された Upper 領域の差分コードへのプログラムフェッチを許可します。

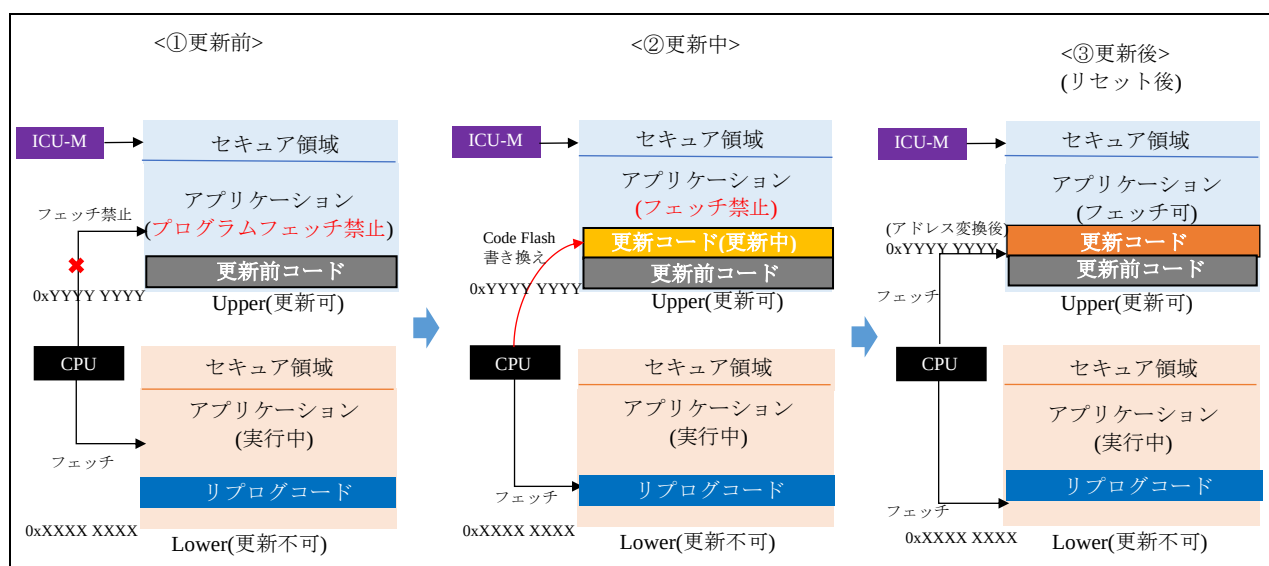


図 3-1 差分コード書き換え方式

差分コード書き換え方式の制限事項

- Code Flash 書き換え中の Bank 内の領域は、Code Flash 書き換え中はアクセス不可になります。従って書き換え中の Code Flash のブロックと同一 Bank のブロックをプログラムフェッチまたはリードすることが必要な場合は、書き換えをサスペンドするか、RAM にコピーしてプログラムフェッチまたはリードを行ってください（RAM 容量の制限により、RAM 実行可能なコードのサイズは限られます）。
- Code Flash の消去・書き換えはブロック単位での実行となるため、更新可能領域に配置するアプリケーションは、差分コード書き換えを考慮し、ブロック毎に機能分割する等の考慮が必要です。
- Code Flash 書き換え中に、同一 Flash System 内の Data Flash を書き換える場合は、FACI のサスペンド機能もしくは強制停止を使用してください。

本アプリケーションノートの動作例、[5. Single map mode による OTA 動作例](#)ではリプログコードを BankA 領域のブロック 0-3、更新コードを BankB のブロック 0-3 に配置します。GCFU を使用し、BankA のブロック 0-3 と BankB のブロック 0-3 のアドレスを切り替えます。ユーザーアプリケーションに差分コード書き換え方式を実装する場合は、ブロック毎に機能分割する等の考慮が必要です。また、本アプリケーションノートは ICU-MH をサポートしません。ユーザーアプリケーションに適用する場合は、必要に応じてセキュア RAM へのコードコピー及び RAM 実行を実装してください。

3.2 バンク切り替え方式(Double map mode)

領域切り替え方式のイメージを図 3-2 に示します。

Double map モード時の同一 FPSYS 内の Code Flash の有効領域(Valid Area)を Lower、更新用領域(Invalid Area)を Upper とした場合の領域切り替えについて説明します。

- ① 更新中
Lower 領域のリプログラムコードで Upper 領域のアプリケーションを書き換えます。
- ② 更新後
アプリケーション更新完了後、Lower 領域と Upper 領域のアドレスマップの切り替え設定を行います。
- ③ 再起動後
マイコンリセットでアドレスマップが切り換わり、Lower 領域の更新アプリケーションを実行します。

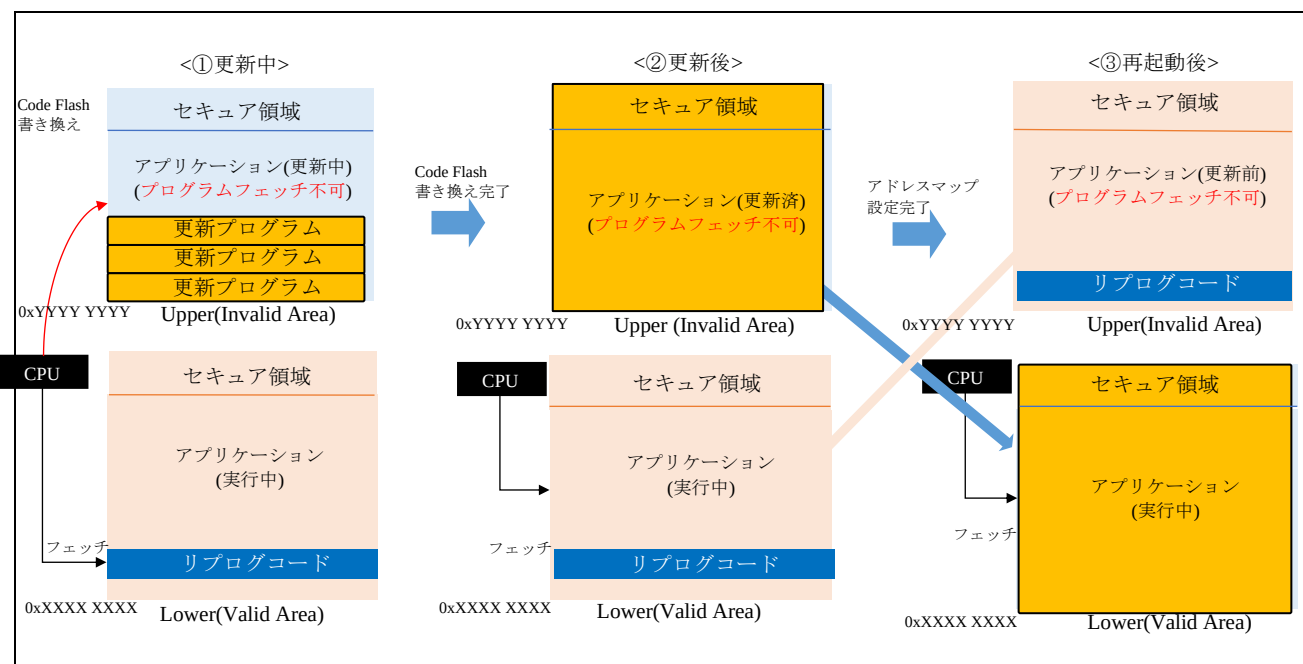


図 3-2 領域切り替え方式

バンク切り替え機能を使った Code Flash 書き換え時の制限事項

- ・ Code Flash 書き換え中に、同一 Flash System 内の Data Flash を書き換える場合は、FACI のサスペンド機能もしくは強制停止を使用してください。

3.3 Security 対応の注意事項

OTA の技術によりアプリケーションソフトウェアの更新を行う場合、サイバー攻撃のリスクへ対処するために堅牢なセキュリティへの対策が求められます。

本アプリケーションノート及びサンプルソフトはフラッシュメモリへのアクセスマスタに **CPU** を使用しています。車載ネットワークのセキュリティ対策を行う場合、ソフトウェアの更新時のフラッシュメモリのアクセスマスタは **ICU-MH** が担うことが想定されます。従って、車載ソフトウェアの更新では、アクセスマスタを **ICU-MH** に変更して実行することが想定されます。^{注1} 図 3-3 に、U2A のアクセスマスタの例を示します。

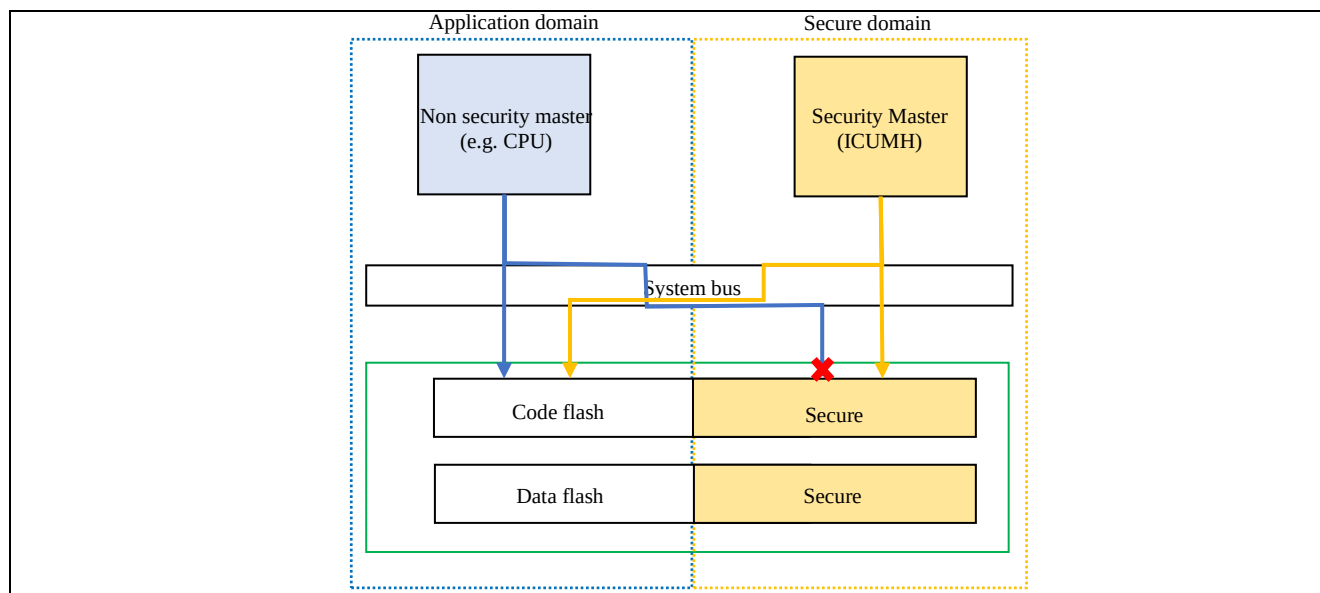


図 3-3 U2A のアクセスマスタの例

また、U2A は FACI やフラッシュメモリを安全に制御するための保護機能を搭載しています。FACI やフラッシュメモリへのアクセスはアクセスマスタによって、保護機能が異なります。^{注1}

本アプリケーションノートでは、外部システムから更新プログラムデータを受信し U2A のフラッシュメモリに書き込みます。セキュリティ対策を講じる場合、外部システムから送信するデータを暗号化する必要があり、データの暗号鍵の認証や改ざんの検知など対策を講じる必要があります。^{注1}

図 3-4 にセキュリティ対策を講じた通信例を示します。

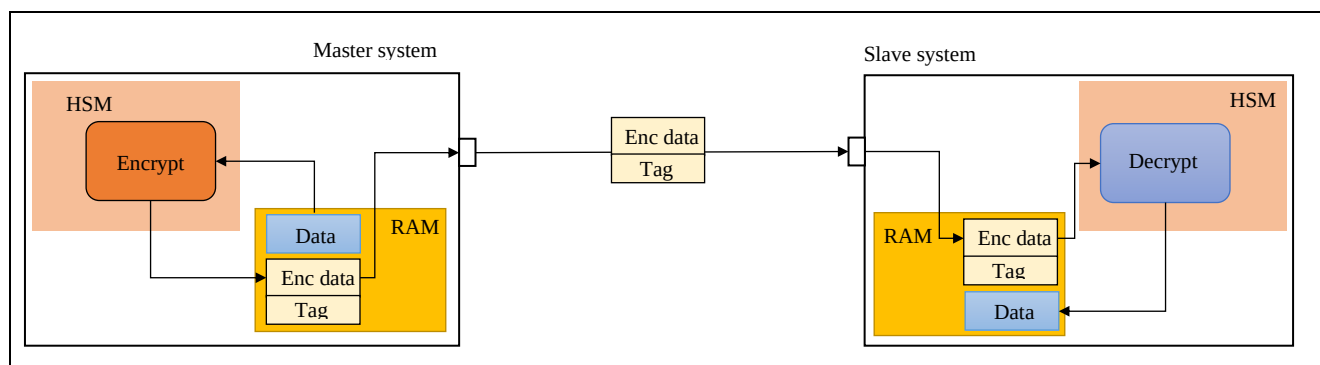


図 3-4 セキュリティ対策を講じた通信例

^{注1} 本アプリケーションノートは、OTA によるソフトウェア更新に求められるセキュリティ対策についてはサポートしません。お客様で実使用に即した、セキュリティ対策をしていただく必要があります。

4. OTA 動作例

本章では、Single map mode 及び Double map mode のサンプルソフトウェアをサポートします。

動作例の詳細は

- [5. Single map mode による OTA 動作例](#)
- [6. Double Map mode による OTA 動作例](#)

を参照ください。

4.1 システム構成

サンプルソフトウェアのシステム構成を図 4-1 に示します。システム構成は各動作例で共通です。

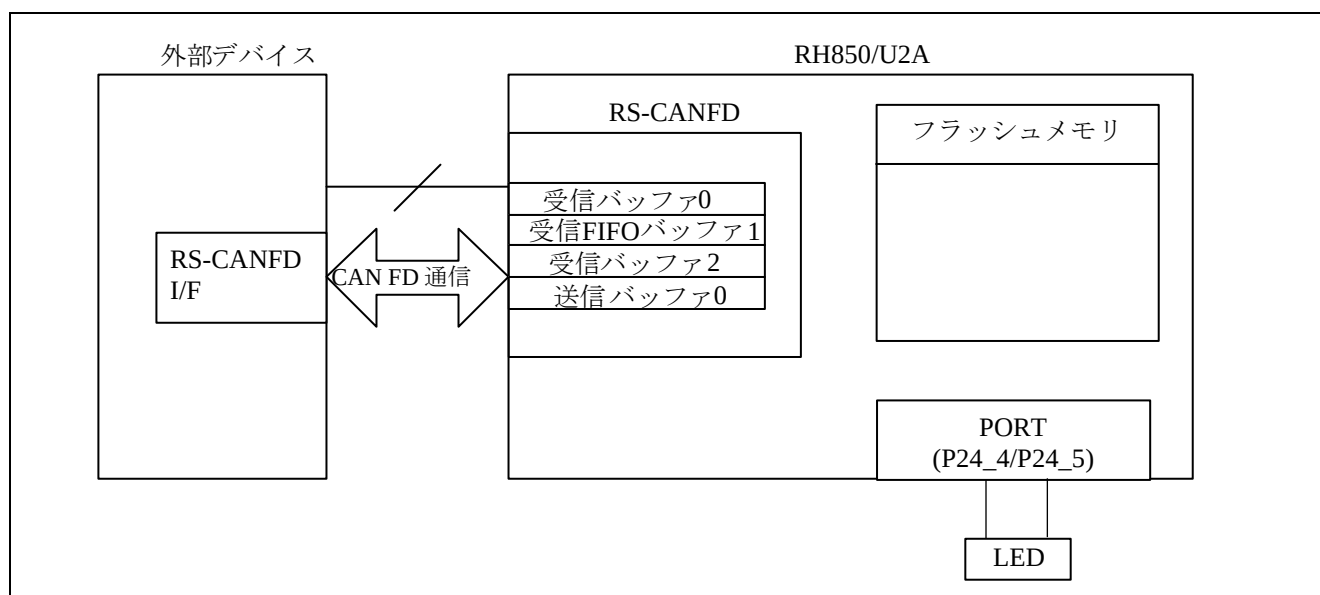


図 4-1 システム構成図

4.2 CAN FD 通信について

CAN FD 通信についての仕様を説明します。通信の仕様は、[5. Single map mode による OTA 動作例](#)、[6. Double Map mode による OTA 動作例](#)で共通となります。

- Code Flash 書き込みに使用されるデータは RS-CANFD(ch0)で通信し、内蔵 RAM に格納されます。
- Code Flash 書き換え対象デバイスは、外部デバイスから CAN FD 通信で送信された特定の ID および、データを受信したときに、対応した Code Flash 書き換え処理を行います。それら特定の ID および、データの組み合わせを本アプリケーションノートでは CAN FD コマンドと呼びます。

4.2.1 CAN FD 通信仕様

- チャンネルは 0 を使用します。
- 通信速度を通常ビットレート 1Mbps、データビットレート 2Mbps に設定します。
- 通信フレームを、CAN FD フレームに設定します。
- 外部デバイスから送信される各 CAN FD コマンドを格納するために、チャンネル 0 の受信ルール数を 2 に設定します。

4.2.2 CAN FD コマンド仕様 (U2A 視点)

- U2A は外部デバイスから書き換え開始コマンドを受信することにより、Code Flash 書き換え処理が開始されます。
- 書き込みデータダウンロードコマンドを外部デバイスから受信することにより、U2A に書き込みデータを受信します。
- U2A から書き込みデータ要求コマンドを外部デバイスに送信することにより、外部デバイスに書き込みデータを要求します。
- U2A が外部デバイスに書き込み終了コマンドを送信することにより、書き換え動作を終了します。

表 4-1 に CAN FD コマンド仕様を示します。

表 4-1 CAN FD コマンド仕様

バッファ	チャネル	コマンド名	送/受	標準 ID	データ長	データ
0	0	書き換え開始コマンド	受信	H'100	1Byte	H'00
1	0	書き込みデータダウンロードコマンド	受信	H'110	64Byte	Code Flash 書き込みデータ 512Byte をダウンロード (64Byte x 8)
1	0	書き込みデータ要求コマンド	送信	H'111	1Byte	H'11
2	0	書き換え終了コマンド	送信	H'121	1Byte	H'22

4.3 Port(LED)について

本アプリケーションノートでは、プログラム実行中に Port(P24_4/P24_5)出力により、U2A に接続した LED を点滅します。Port 出力は、[5. Single map mode](#) による OTA 動作例、[6. Double Map mode](#) による OTA 動作例で共通となります。表 4-2 に Port 出力の仕様を示します。

表 4-2 Port 出力仕様

Bank A 実行中	Bank B 実行中	P24_4	P24_5
書き換えプログラム動作中	-	LED 点滅	-
-	書き換えプログラム動作中	-	LED 点滅

5. Single map mode による OTA 動作例(GCFU を使用したアドレス変換)

5.1 仕様

5.1.1 ソフトウェア仕様

- 本アプリケーションノートは、Single Map mode による OTA 2Bank 構成で、ユーザー領域の書き換えを行います。表 5-1 にメモリ割り付けを示します。
- GCFU (Global Calibration Function Unit)機能を使用して、BankA 起動時は BankA を論理アドレス 0x00000000～0x0000FFFF として動作するように設定します。
- BankB 起動時には BankB を論理アドレス 0x00000000～0x0000FFFF として動作するように設定します。
起動 Bank 情報は Data Flash に格納しており、起動 Bank を判別して GCFU 機能にてアドレス変換を行います。
- 書き換えはプログラムが動作する Bank とは異なる Bank に対して行います。
- Code Flash メモリマッピングモードは Single Map mode、動作モードは User boot mode0、ブート領域は User boot Area0 です。
- フラッシュメモリへのアクセスは Renesas Flash Driver(RFD)を用いて行います。
- 本動作例では、Code Flash の User Area BankC、BankD は書き換え対象外です。

表 5-1 メモリ割り付け

領域	物理アドレス	ブロック	Bank	サイズ	OTA 対象
User boot Area 0	0x0800 0000 – 0x0800 FFFF	User boot 領域 0	BankA	64K バイト	非対象
User Area0	0x0000 0000 – 0x0000 FFFF	ブロック 0-3	BankA	64K バイト	対象
User Area1	0x0040 0000 – 0x0040 FFFF	ブロック 0-3	BankB	64K バイト	対象

5.1.2 動作イメージ

起動する Bank は、次回起動する Bank 情報を Data Flash に格納して判定します。起動する Bank が BankA の場合はアドレス変換せず、BankB の場合はアドレス変換します。起動 Bank による動作を以下に説明します。

(a) BankA 起動

GCFU によるアドレス変換は行いません（物理アドレス＝論理アドレス）。BankA のリプログラムコードでは FACI を制御し BankB の書き換えを行います。

図 5-1 に BankA 起動での書き込み動作を示します。

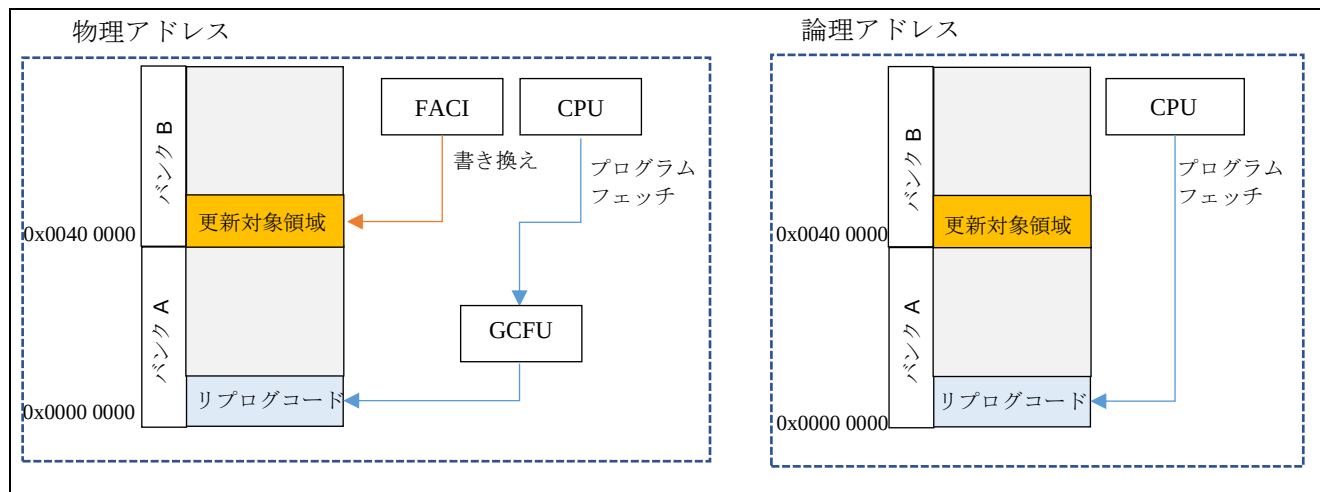


図 5-1 BankA 起動での書き込み動作

(b) BankB 起動

GCFU によりアドレス変換します。（物理アドレス≠論理アドレス）BankB 領域(物理アドレス)のリプログラムコードでは FACI を制御し BankA の書き換えを行います。CPU からのプログラムフェッチは、BankA 起動時と同じアドレスへ行いますが、GCFU によりプログラムフェッチ先のアドレスを BankB に変換します。

図 5-2 に BankB 起動での書き込み動作を示します。

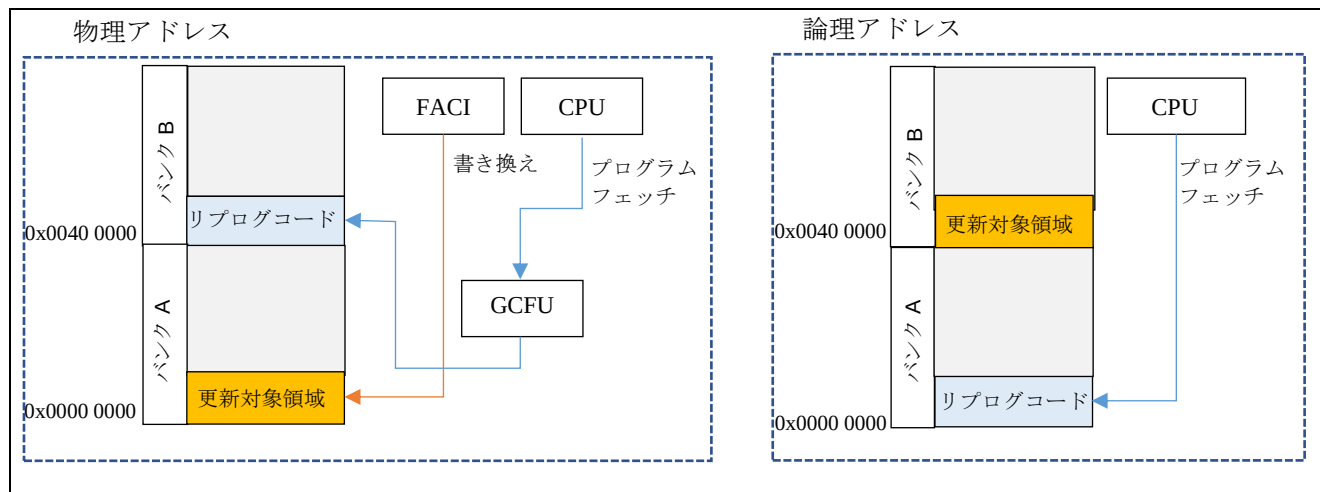


図 5-2 BankB 起動での書き込み動作

5.1.3 全体シーケンス

図 5-3、図 5-4 に全体シーケンスを示します。

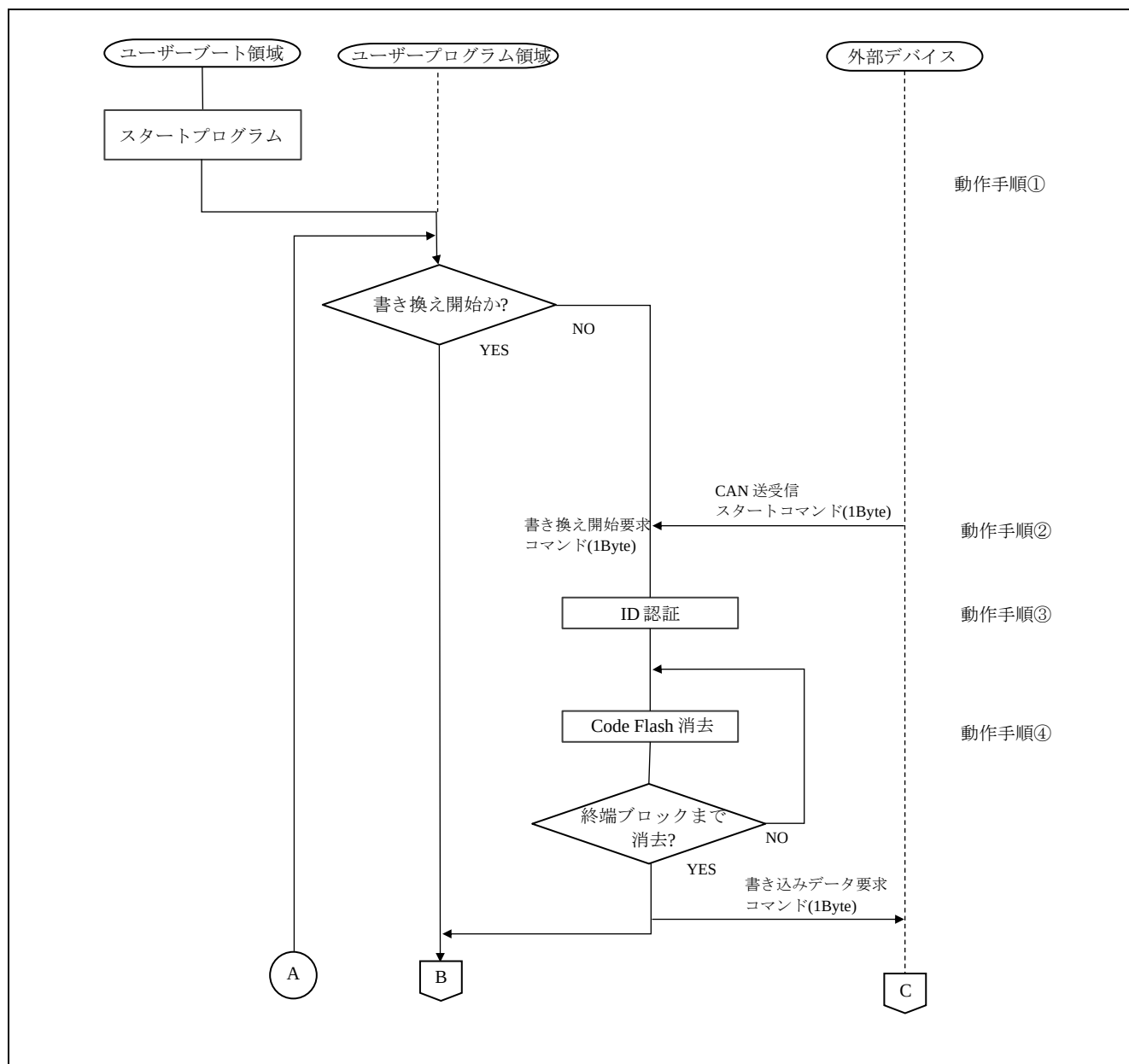


図 5-3 全体シーケンス図(1)

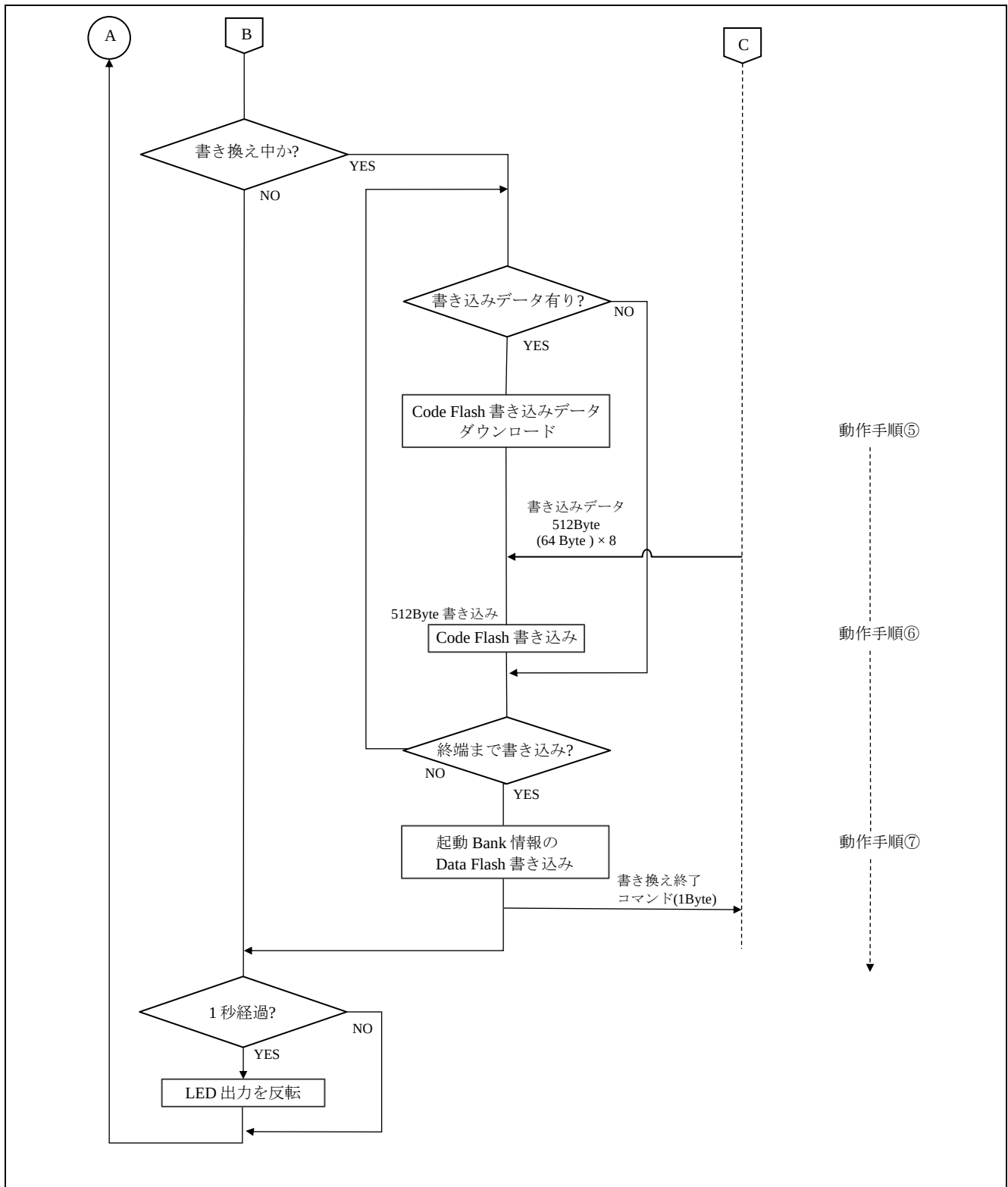


図 5-4 全体シーケンス図(2)

5.1.4 使用機能

- CANFD インタフェース(RS-CANFD)
- FACI
- GCFU
- PORT

5.1.5 動作モード

本アプリケーションノートでは、Code Flash 書き換え時のマイコンの動作モードは User boot mode0で行います。User boot mode0では、CPU0のリセットベクタアドレスは User boot 領域(0x08000000)となります。動作モード選択方法は、モード端子およびオプションバイトで設定します。オプションバイトの設定は、RH850 ファミリ用 Renesas Flash Programmer または、コンフィグレーション設定コマンドを使用し設定します。表 5-2 に動作モード選択を示します。

表 5-2 動作モード選択

端子設定値			オプションバイト設定値		動作モード	起動領域
FLMD0	FLMD1	_TRST	STMSEL1	STMSEL0		
0	X	0	0	1	User boot mode0	User boot Area0

5.1.6 メモリマッピングモード

本動作例では、Code Flash 書き換え時のメモリマッピングは Single Map modeで行います。表 5-3 にメモリマッピング選択を示します。

表 5-3 メモリマッピング選択

オプションバイト設定値		マッピングモード
MAPMODE1	MAPMODE 0	
0	1	Single Map mode

5.1.7 RFD コンフィギュレーション

本動作例で使用する RFD のマクロ定義一覧を、表 5-4 に示します。

表 5-4 マクロ定義一覧

マクロ定義	設定値	概要
R_RFD_VALUE_FORCED_STOP_TIMEOUT	サンプルコード参照	強制的終了コマンドのタイムアウト時間
R_RFD_ERASURE_SUSPENDED_MODE	R_RFD_ERASURE_PRIORITY_MODE	消去優先モードにする
R_RFD_REG_FAEINT_CFAEIE R_RFD_REG_FAEINT_CMDLKIE R_RFD_REG_FAEINT_DFAEIE R_RFD_REG_FAEINT_ECRCTIE	R_RFD_DISABLE	FLERR 割り込みを無効にする
R_RFD_CONTROL_TARGET_DATAFLASH	R_RFD_ENABLE	Data flash にアクセスする
R_RFD_CONTROL_TARGET_CODEFLASH	R_RFD_ENABLE	Code flash にアクセスする
R_RFD_MAPMODE	R_RFD_SINGLE	マップモードは Single map mode
R_RFD_BLOCK_PROTECTION_AREA1	R_RFD_ENABLE	Block protection area1 はアクセス対象
R_RFD_FPMON_CHECK	R_RFD_ENABLE	FPMON チェックを有効にする

5.2 動作手順

動作手順①～⑦は、「5.1.3 全体シーケンス」に対応します。

5.2.1 ① 起動 Bank 切り替え

リセットスタート後、User boot 領域上のスタートプログラムは Data Flash を読み出して、ユーザー領域の起動 Bank を判定し、適切な Bank に切り替えてユーザープログラムにジャンプします。図 5-5 にスタートプログラム動作を示します。

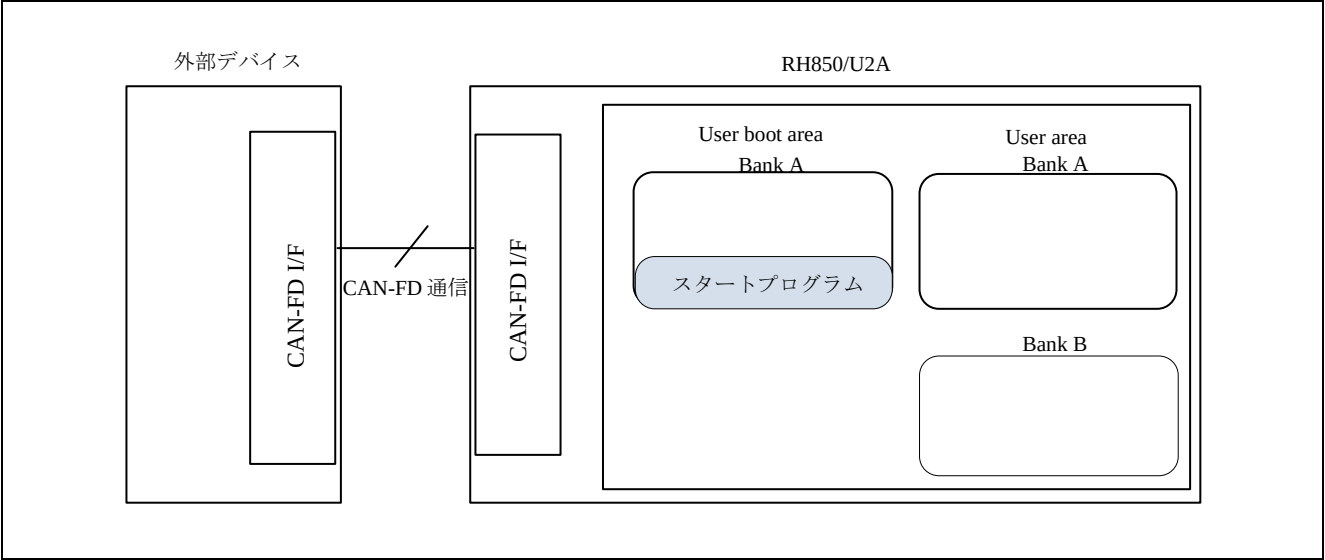


図 5-5 スタートアッププログラム動作

表 5-5 「main_pm0()関数」

関数名	引数	概要
main_pm0()	無し	プログラム開始。ユーザー領域の起動 Bank 切り替え後、ユーザープログラムにジャンプします。

図 5-6 に「main_pm0()関数」のフローチャートを示します。

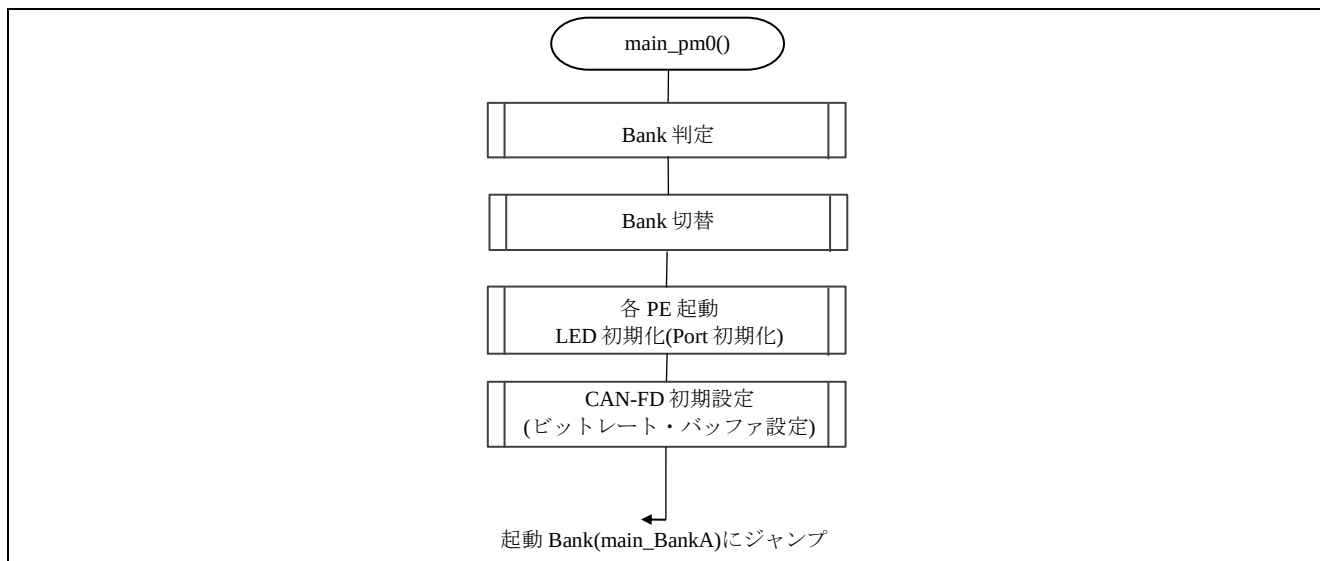


図 5-6 「main_pm0()関数」のフローチャート
「動作手順①」

図 5-7 に「Bank 切替関数」のフローチャートを示します。

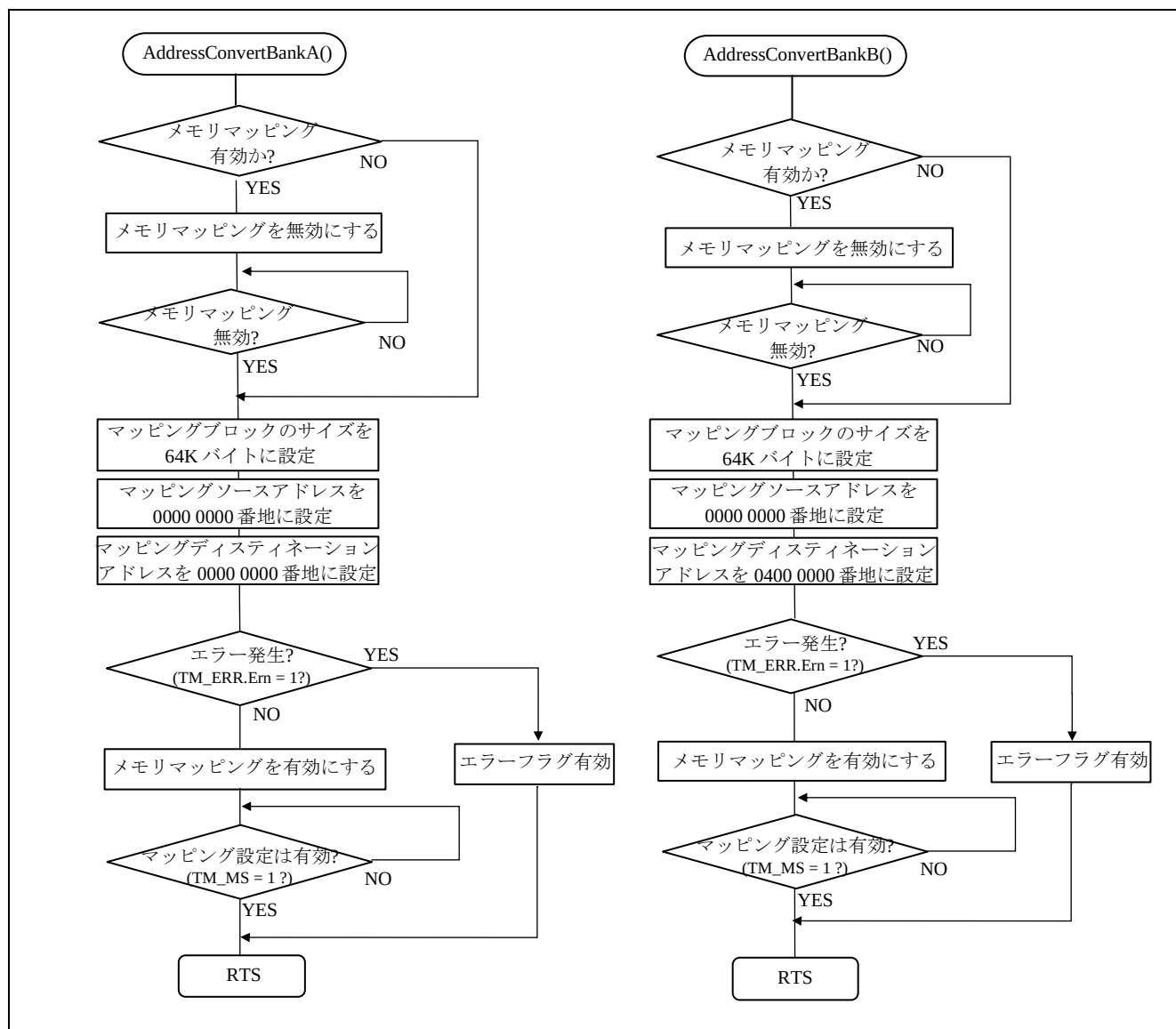


図 5-7 「Bank 切替関数」のフローチャート

5.2.2 ② BGO 書き込み

書き換え対象の Code Flash 領域は起動 Bank とは異なる Bank にすることで、「書き換え制御プログラム」を RAM に転送せずに Code Flash へ書き込むことができます。BankA で起動した場合は BankA の「書き換え制御プログラム」で、BankB の Code Flash 領域を書き換えます。

メインルーチンでは常に LED を点滅するユーザープログラムを実行しており、書き換え開始要求コマンドを受信後に書き換え制御プログラムを並行して OTA を実行します。

図 5-8に Code Flash 消去/書き換え開始動作を示します。

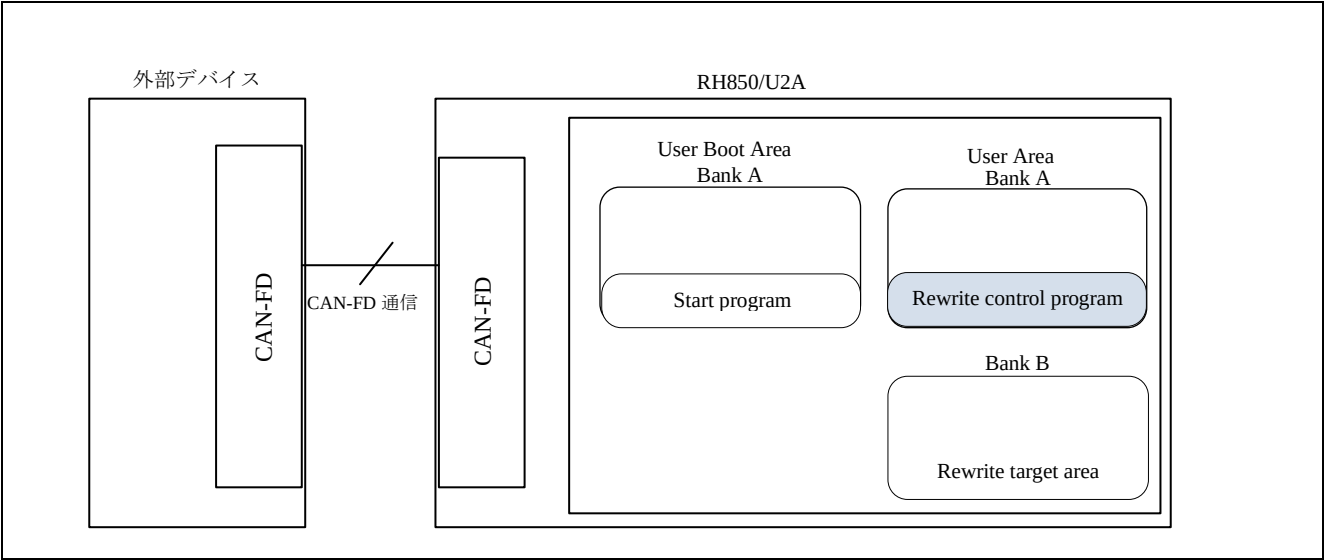


図 5-8 Code Flash 消去/書き換え開始動作

関数説明

表 5-6 「main_BankA()関数」

関数名	引数	概要
main_BankA()	無し	LED の点滅を繰り返し、書き換え開始コマンドを受信後、「書き換え制御プログラム」が実行されます。

図 5-9 に「main_BankA()関数」のフローチャートを示します。

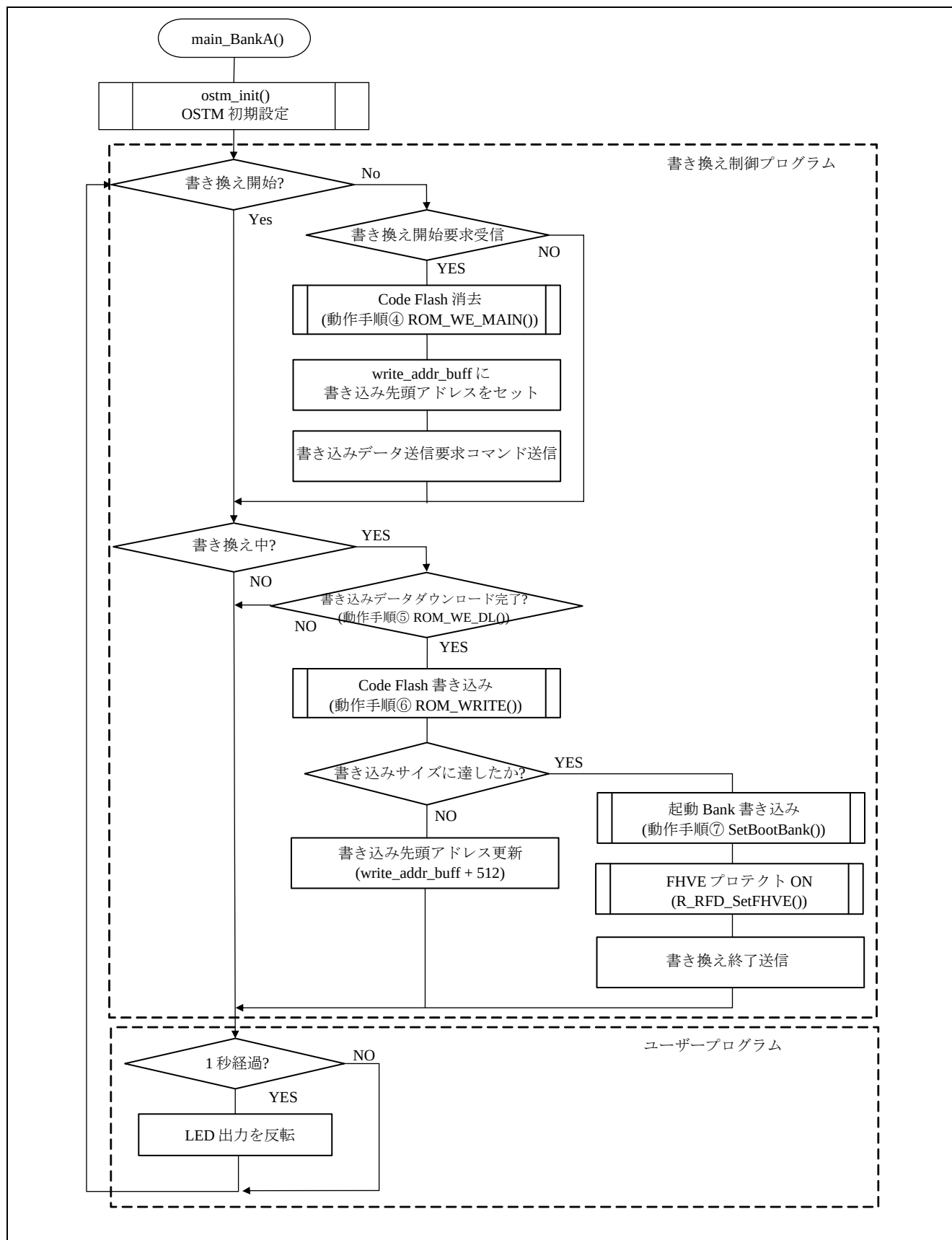


図 5-9 「main_BankA()関数」のフローチャート

「動作手順②」

関数説明

表 5-7 「ROM_WE_MAIN()関数」

関数名	引数	概要
ROM_WE_MAIN()	無し	ID 認証、Code Flash 消去、Code Flash 書き込みデータダウンロード、Code Flash 書き込みの各種関数コールを実施します。

図 5-10 に「ROM_WE_MAIN()関数」のフローチャートを示します。

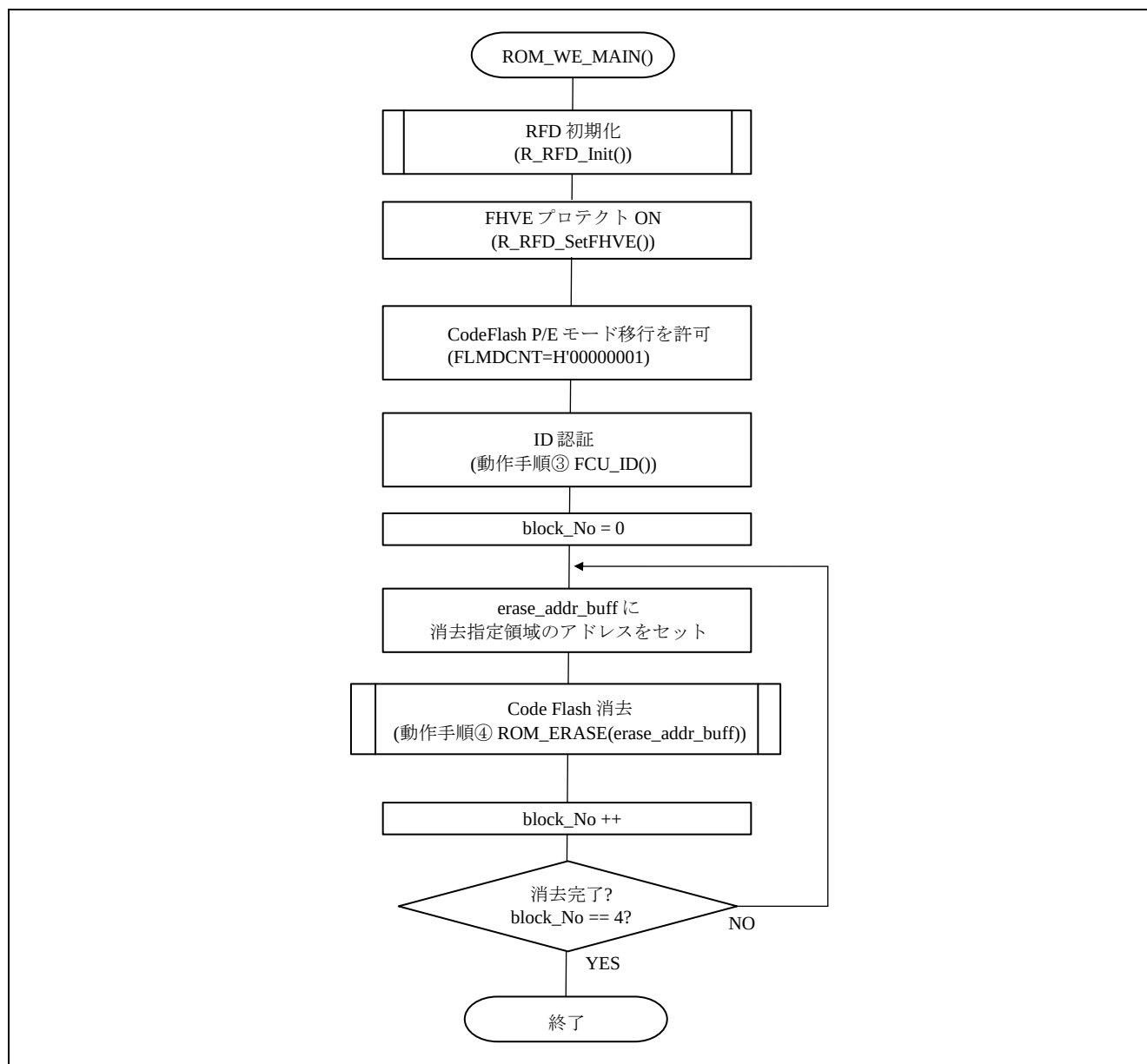


図 5-10 「ROM_WE_MAIN()関数」のフローチャート
「動作手順③、④を含む」

5.2.3 ③ ID 認証

「書き換え制御プログラム」の「ID 認証関数」を実行。Flash メモリの特殊領域にあらかじめ設定した 256 ビットの ID と Customer ID の値を比較することで ID の認証を実行します。

本アプリケーションノートでは、ID 設定は 1 バイト目を"0"、1 バイト目以外を"F"の設定で使用しています。ID 設定を変更する際には RH850 ファミリー用 Renesas Flash Programmer または、コンフィグレーション設定コマンドを使用し設定します。

関数説明

表 5-8 「FCU_ID()関数」

関数名	引数		概要
FCU_ID()	無し		Flash メモリの特殊領域に設定された ID との比較と ID 認証を実行します。
API 名	引数		概要
R_RFD_IDAuth	T_en_IDType	R_RFD_ID_CUSTIDA/B/C	Flash メモリの特殊領域に設定された ID との比較と ID 認証を実行します。
	T_u1*	ID データのポインタ	

図 5-11 に「FCU_ID()関数」のフローチャートを示します。

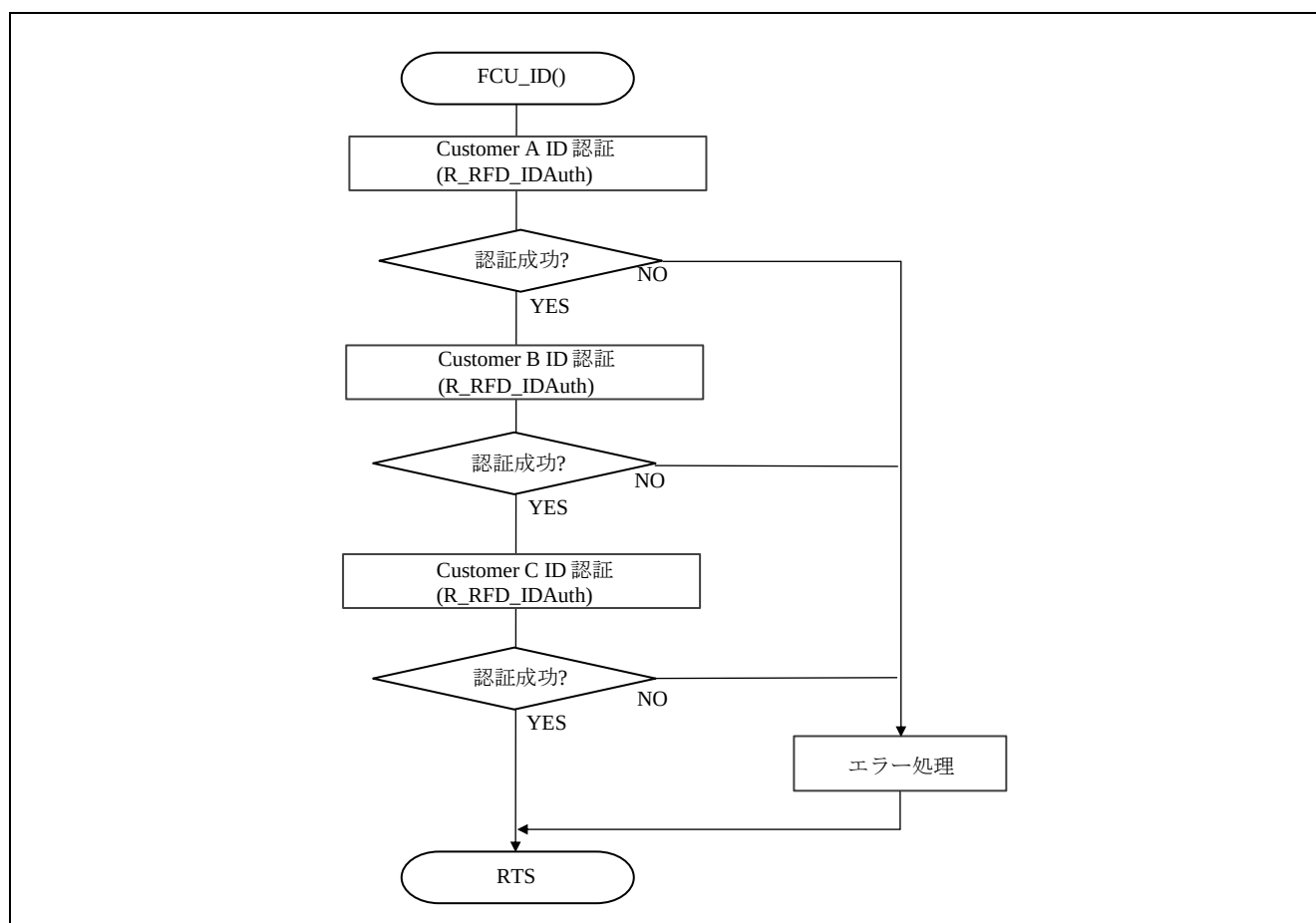


図 5-11 「FCU_ID()関数」のフローチャート

「動作手順③」

5.2.4 ④ Code Flash 消去

ID 認証後、「書き換え制御プログラム」の「Code Flash 消去関数」を実行。

FACI コマンド発行領域にブロックイレーズコマンドを発行し、Code Flash 書き換え指定領域を消去します。

関数説明

表 5-9 「ROM_ERASE()関数」

関数名	引数		概要
ROM_ERASE()	erase_addr	消去する Code Flash 領域の先頭アドレス	Code Flash 書き換え指定領域を消去します
API 名	引数		概要
R_RFD_ShiftToPEMode	T_u2_FACI	R_RFD_FACI0	P/E モードへ遷移
	T_en_FACIMode	R_RFD_MODE_CFPE	
R_RFD_ShiftToReadMode	T_u2_FACI	R_RFD_FACI0	Read モードへ遷移
R_RFD_GetFaciStatus	T_u2_FACI	R_RFD_FACI0	コマンドロックの状態及びエラーチェック
R_RFD_GetFaciSequenceReady	T_u2_FACI	R_RFD_FACI0	FRDY ビットの状態チェック
R_RFD_StatusClear	T_u2_FACI	R_RFD_FACI0	FACI ステータスクリア
R_RFD_ForcedStopAndErrorClear	T_u2_FACI	R_RFD_FACI0	強制終了コマンド発行
R_RFD_EraseCFRequest	T_u4_RfdAddress	消去領域の先頭アドレス	Code Flash 消去コマンド発行

図 5-12 に「ROM_ERASE()関数」のフローチャートを示します。

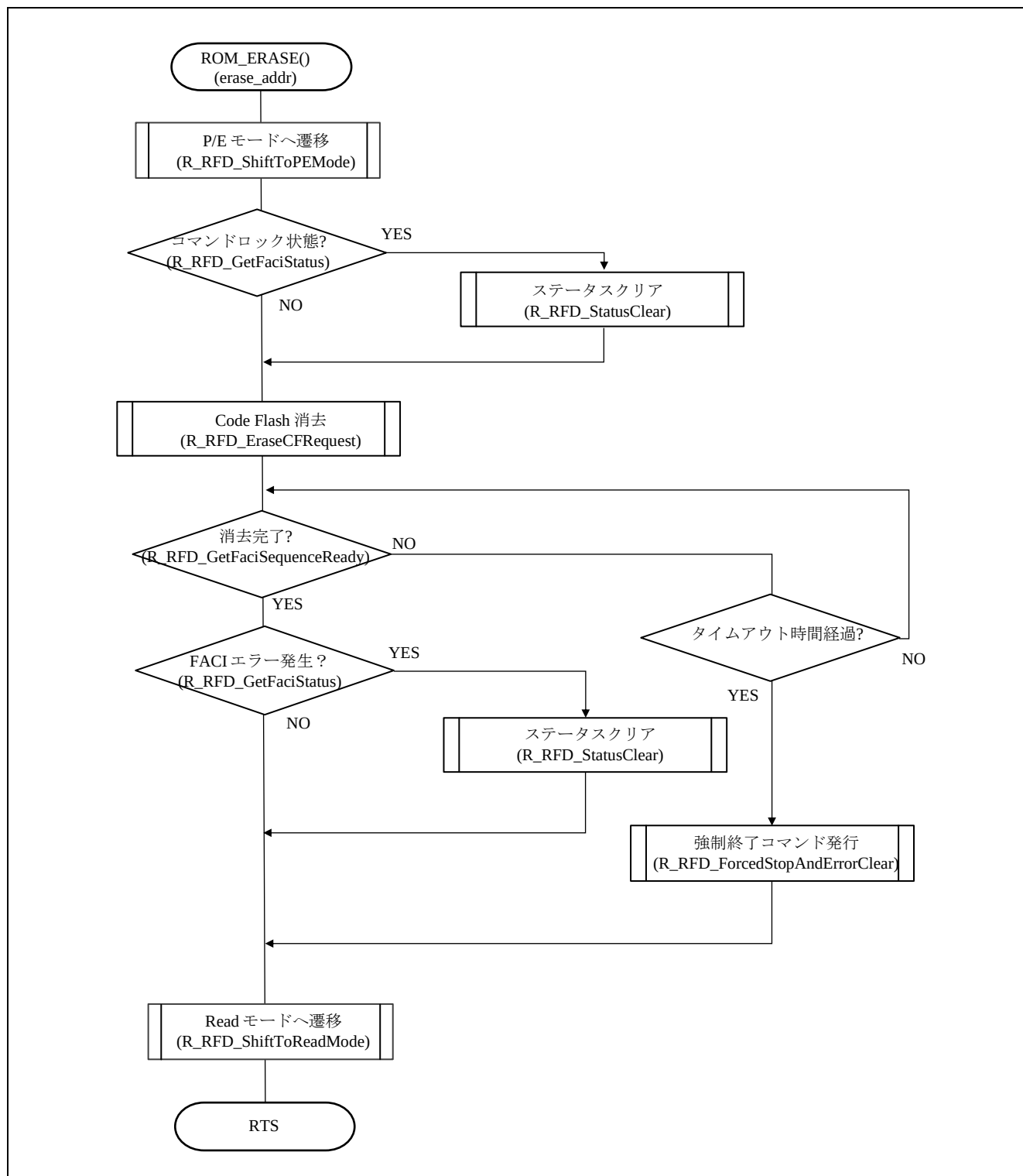


図 5-12 「ROM_ERASE()関数」のフローチャート
「動作手順④」

5.2.5 ⑤ Code Flash 書き込みデータダウンロード

「Code Flash 書き込みデータダウンロード関数」を実行し、書き込みデータ要求コマンドを送信します。書き込みデータ要求コマンドを受信した外部デバイスは、書き込みデータダウンロードコマンドで書き込みデータ 512Byte を U2A へ送信します。「Code Flash 書き込みデータダウンロード関数」では受信した書き込みデータを RAM へ保存します。

関数説明

表 5-10 「ROM_WE_DL()関数」

関数名	引数	概要
ROM_WE_DL()	無し	外部デバイスから書き込みデータをダウンロード

図 5-13 に「ROM_WE_DL()関数」のフローチャートを示します。

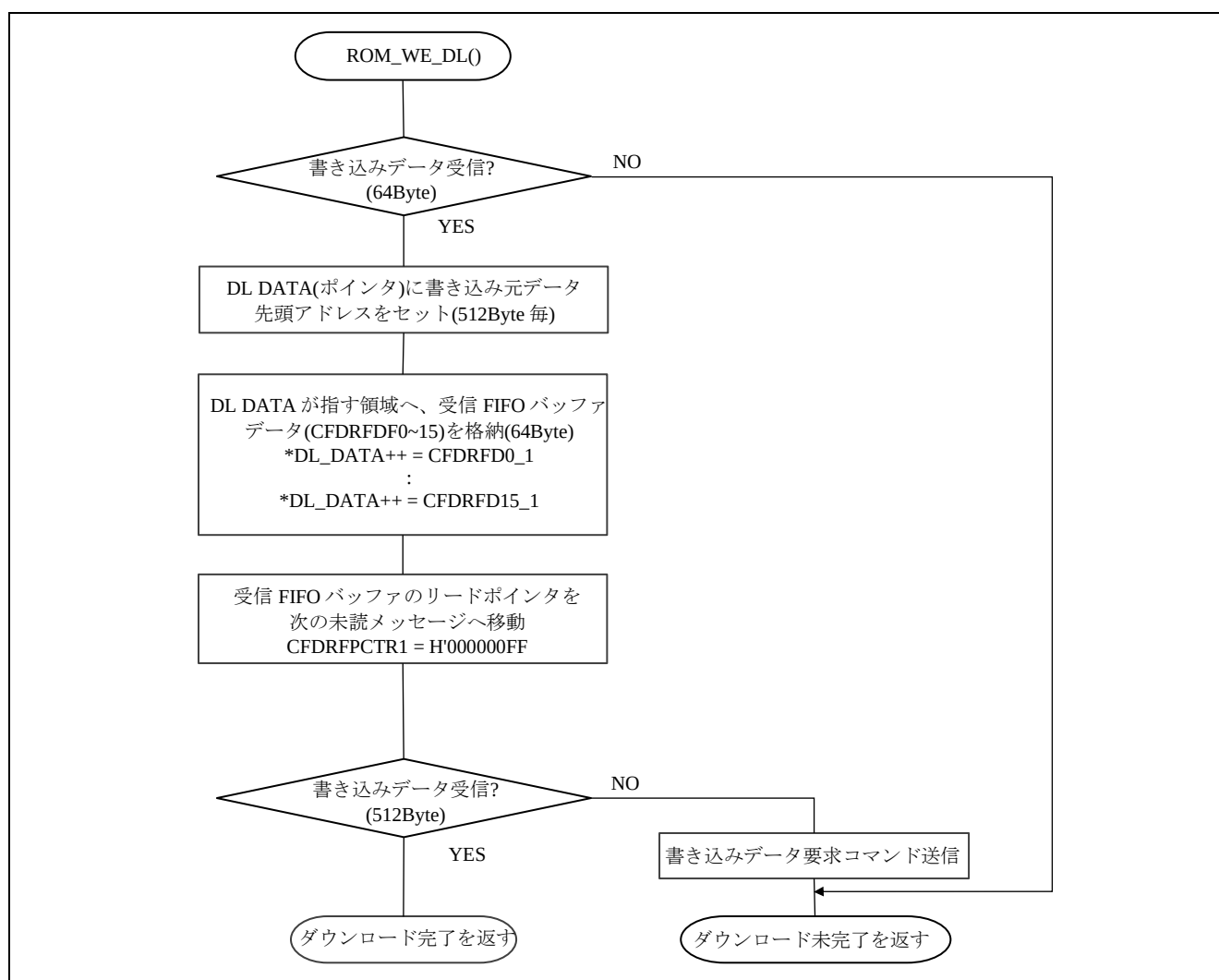


図 5-13 「ROM_WE_DL()関数」のフローチャート
「動作手順⑤」

5.2.6 ⑥ Code Flash 書き込み

CAN 通信で外部デバイスから受信した書き込みデータは、「Code Flash 書き換え関数」を使用して Code Flash に書き込みます。

FACI コマンド発行領域にプログラム・コマンドを発行し、Code Flash 書き換え指定領域に書き込みます。書き込みサイズ（64K バイト）に達した場合、Flash 書き換えを終了します。

関数説明

表 5-11 「ROM_WRITE()関数」

関数名	引数		概要
ROM_WRITE()	write_addr	書き込む Code Flash 領域の先頭アドレス	Code Flash 書き換え指定領域に書き込みます (512Byte 単位)
API 名	引数		概要
R_RFD_ShiftToPEMode	T_u2_FACI	R_RFD_FACI0	P/E モードへ遷移
	T_en_FACIMode	R_RFD_MODE_CFPE	
R_RFD_ShiftToReadMode	T_u2_FACI	R_RFD_FACI0	Read モードへ遷移
R_RFD_GetFaciStatus	T_u2_FACI	R_RFD_FACI0	コマンドロックの状態及びエラーチェック
R_RFD_GetFaciSequenceRead	T_u2_FACI	R_RFD_FACI0	FRDY ビットの状態チェック
R_RFD_StatusClear	T_u2_FACI	R_RFD_FACI0	FACI ステータスクリア
R_RFD_ForcedStopAndErrorClear	T_u2_FACI	R_RFD_FACI0	強制終了コマンド発行
R_RFD_WriteCFRequest	T_u4_RfdAddress	書き込み領域の先頭アドレス	Code Flash 書き込みコマンド発行

図 5-14 に「ROM_WRITE()関数」のフローチャートを示します。

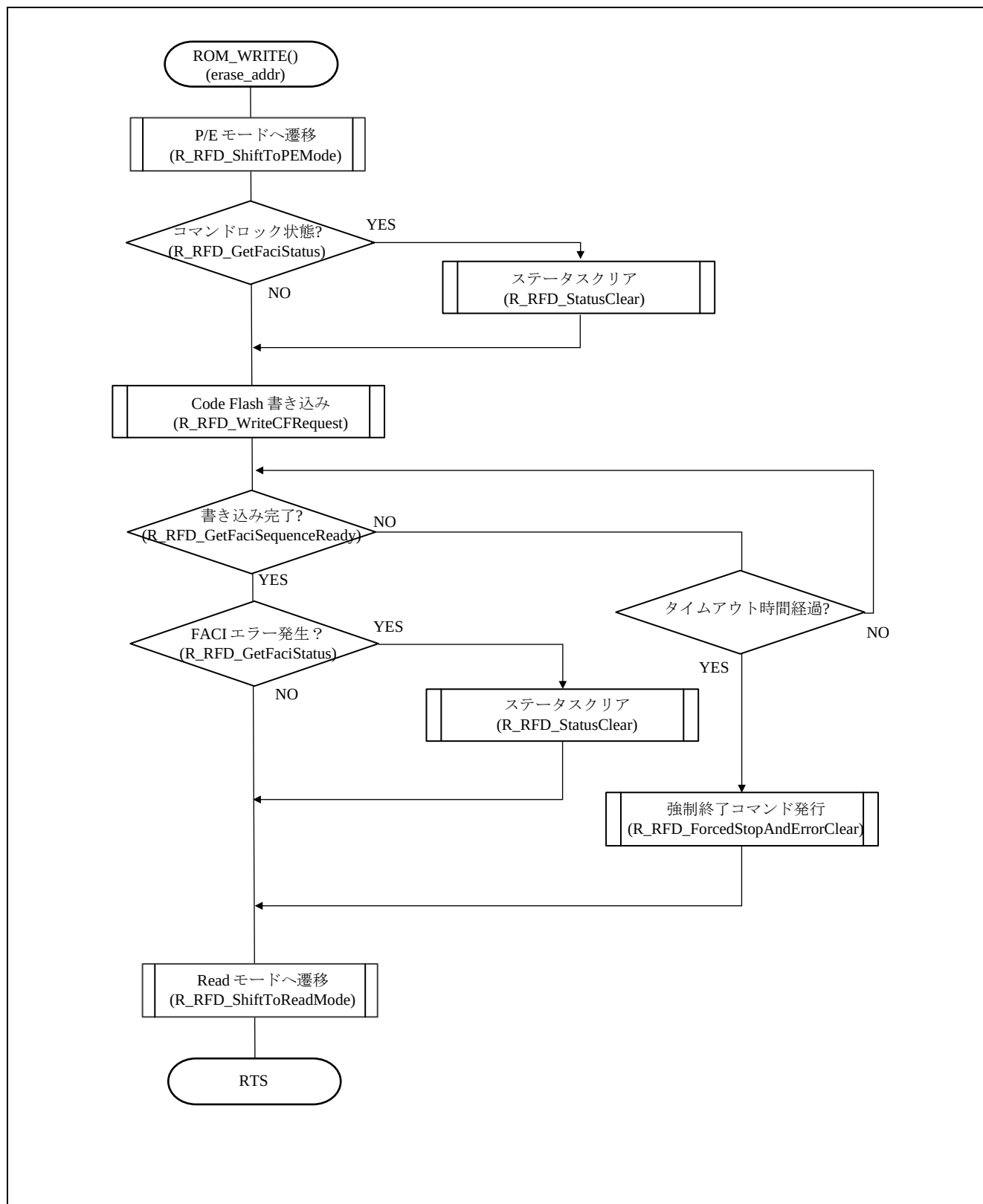


図 5-14 ROM_WRITE()関数」のフローチャート

「動作手順⑥」

5.2.7 ⑦ 起動 Bank 情報書き込み関数

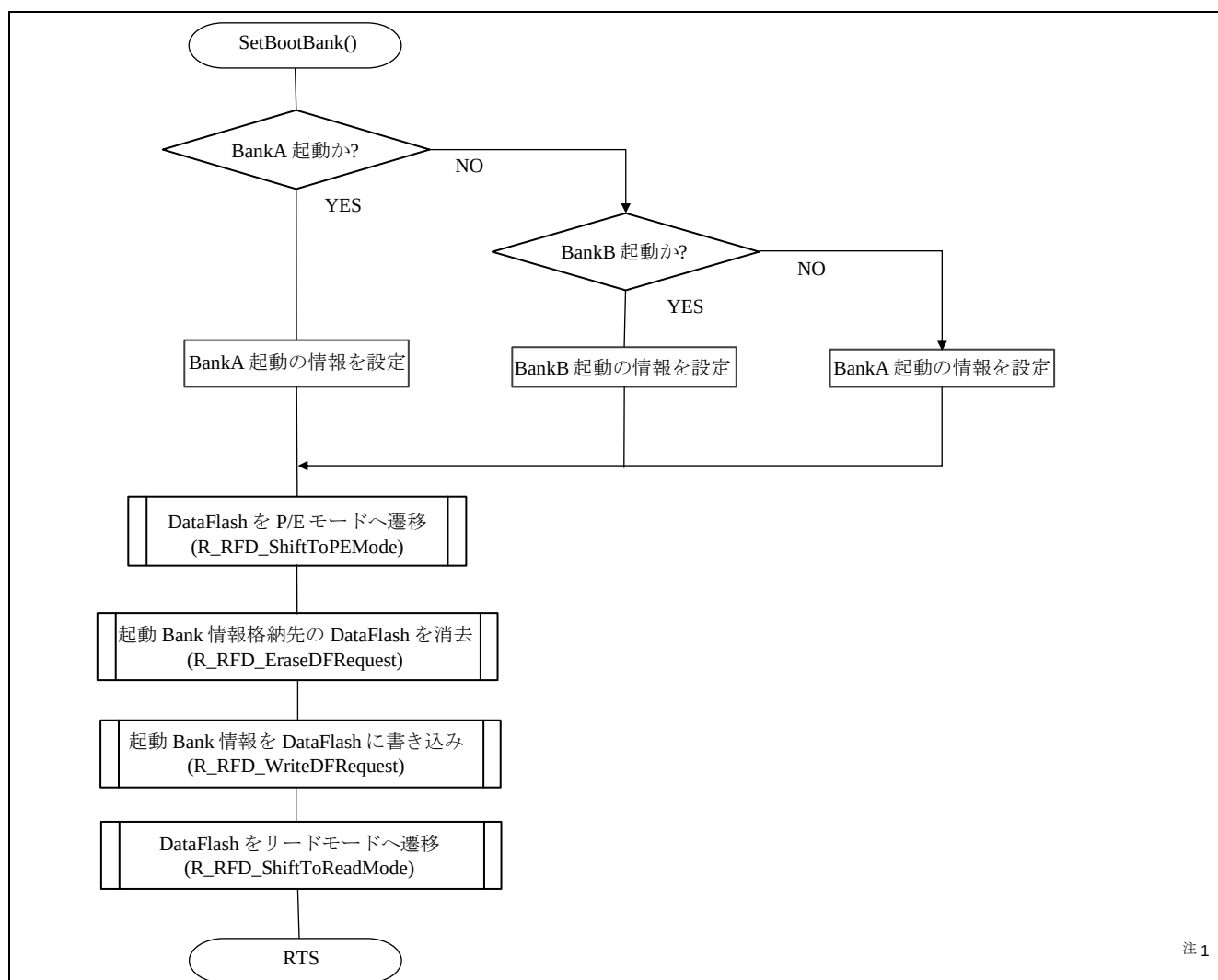
書き換え終了後、次の起動 Bank の情報を Data Flash に書き込みます。

関数説明

表 5-12 「SetBootBank ()関数」

関数名	引数	概要	
SetBootBank ()	無し	次の起動 Bank 情報を判定し、Data Flash に書き込みます。	
API 名	引数		概要
R_RFD_ShiftToPEMode	T_u2_FACI	R_RFD_FACI0	P/E モードへ遷移
	T_en_FACIMode	R_RFD_MODE_DFPE	
R_RFD_ShiftToReadMode	T_u2_FACI	R_RFD_FACI0	Read モードへ遷移
R_RFD_EraseDFRequest	T_u4_RfdAddress	消去領域の先頭アドレス	Data Flash 消去
R_RFD_WriteDFRequest)	T_u4_RfdAddress	書き込み領域の先頭アドレス	Data Flash 書き込み

図 5-15 に「SetBootBank ()関数」のフローチャートを示します。



注 1

図 5-15 「SetBootBank ()関数」のフローチャート

「動作手順⑦」

注 1 S_OPBT4.DPROT=0 に設定する場合は Data Flash ID の認証が必要となります。

5.3 メモリ割り付け

表 5-13 に Single map mode のアドレス配置図(Bank A 有効時)を示します。

表 5-13 Single map mode のアドレス配置図(Bank A 有効時)

領域	アドレス	セクション	アクセス
Code Flash User Area0(Bank A)	H'0000_0000	.BankA.text	PE0
		.BankA.const	
		.FlashControl.text	
		.R_RFD_RODATA_EXTRA.const	
		.R_RFD_CODE_COMMON.text	
		.R_RFD_CODE_COMMON_RAM_NO_BGO.text	
		.R_RFD_CODE_USEROWN_COMMON.text	
		.R_RFD_CODE_DF.text	
		.R_RFD_CODE_CF.text	
		.R_RFD_CODE_CF_RAM_NO_BGO.text	
		.R_RFD_CODE_EXTRA.text	
		.R_RFD_RODATA_VERSION_DF.const	
		.R_RFD_RODATA_VERSION_CF.const	
		.R_RFD_RODATA_VERSION_COMMON.const	
	H'0002_0000	RESET_PE1	PE1
		EIINTTBL_PE1	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.text	
		.data	
Code Flash User Area2(Bank C) ※U2A8,U2A6は非搭載	H'0080_0000	RESET_PE2	PE2
		EIINTTBL_PE2	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.text	
Code Flash User Area3(Bank D) ※U2A8,U2A6は非搭載	H'00C0_0000	RESET_PE3	PE3
		EIINTTBL_PE3	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.text	
Code Flash User boot area0(Bank A)	H'0800_0000	RESET_PE0	PE0
		EIINTTBL_PE0	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.data	
		.text	
LRAM(PE3) ※U2A8,U2A6は非搭載	H'FD60_0000	.text	PE0～PE3
		.data.R	
		.bss	
LRAM(PE2) ※U2A8,U2A6は非搭載	H'FD80_0000	.stack.bss	PE3
		.data.R	
		.bss	
LRAM(PE1)	H'FDA0_0000	.stack.bss	PE2
		.data.R	
		.bss	
LRAM(PE0)	H'FDC0_0000	.stack.bss	PE1
		.data.R	
		.bss	
		.stack.bss	
		.data.R	
Data Flash(Block 0)	H'FF20_0000	.R_RFD_BSS.bss	PE0
		.BankA.bss	
		.data.R	
		.stack.bss	
		.BootBankInfo.const	

6. Double Map mode による OTA 動作例(バンク切替方式)

6.1 仕様

6.1.1 ソフトウェア仕様

- 本動作例は Double Map mode で Code flash に更新プログラムの書き換えを行います。
- Code Flash を有効領域(valid area)と無効領域(invalid area)に分割し、アドレススワップ機能を使用して、有効領域と無効領域を入れ替えます。
- 更新ソフトウェアの書き換えは、Code Flash のプログラムが動作する有効領域とは異なる無効領域に対して行います。
- ソフトウェアの更新の対象となるのは、Code Flash の User Area0/1(Bank A、Bank B)で、後述する Hardware Property Area の設定により、Bank A と Bank B のアドレスを入れ替えます。
- Hardware Property Area の設定により、Code Flash の Extended Area(User boot area, Product info Area, ECC Test Area)の BankA と BankB のアドレスも切り替わります。
- 表 6-1 に Code Flash の Bank A が有効領域のときの、Code Flash のメモリ割り付けを示します。本動作例では、Code Flash の User Area BankC、BankD は書き換え対象外です。
- User boot area は、シリアルプログラミングモードでのみ書き換え可能な領域です。User boot Area の BankA, BankB には、あらかじめ同じデータを書き込んでおく必要があります。Product info Area, ECC Test Area は書き換え不可の領域です。

表 6-1 Code Flash メモリ割り付け(BANK A 有効時)

Code Flash 領域	物理アドレス	ブロック	有効/無効	サイズ	書き換え対象
User Area 0 (U2A16/8)	0x0000 0000 – 0x003F FFFF	ブロック 0-69	有効	4MB	非対象
User Area 0 (U2A6)	0x0000 0000 – 0x002F FFFF	ブロック 0-53		3MB	
User Area 1 (U2A16/8)	0x0200 0000 – 0x023F FFFF	ブロック 0-69	無効	4MB	対象
User Area 1 (U2A6)	0x0200 0000 – 0x022F FFFF	ブロック 0-53		3MB	
User Boot Area 0	0x0800 0000 – 0x0800 FFFF	User boot 領域 0	有効	64KB	非対象
User Boot Area 1	0x0A00 0000 – 0x0A00 FFFF	User boot 領域 1	無効	64KB	非対象
Product info Area 0	0x0803 0000 – 0x0803 7FFF	プロダクト情報領域 0	有効	32KB	非対象
Product info Area 1	0x0A03 0000 – 0x0A03 7FFF	プロダクト情報領域 1	無効	32KB	非対象
ECC Test Area 0	0x0805 0000 – 0x0805 FFFF	ECC テスト領域 0	有効	64KB	非対象
ECC Test Area 1	0x0A05 0000 – 0x0A05 FFFF	ECC テスト領域 1	無効	64KB	非対象

- Data Flash の Hardware Property Area の Configuration setting area, Security setting area, Block protection area0/1, Switch Area は、Code Flash と同様に有効領域(front side)と無効領域(back side)で構成され、有効領域と無効領域を切り替えることが可能です。
- 本動作例では、Hardware Property Area の Configuration setting area, Switch Area, TAG Area を書き換えることにより、Code Flash のアドレス変換を実行します。
- Code Flash の BankA、Bank B の有効/無効は、「Configuration setting area」の OPBT13.DBMAPSW0 により、設定されます。
- 「Configuration setting area」の有効領域は Switch Area の CVA で選択します。「Switch Area」の有効領域は TAG Area の VAF で選択します。本動作例では、「Configuration setting area」の無効領域の OPBT13.DBMAPSW0 を書き換え、「Switch Area」の CVA 及び「TAG Area」の VAF の設定により「Configuration setting Area」の有効領域を切り替えます。
- 上記の設定完了後、リセットにより Code Flash の BankA, BankB のアドレス変換が実行されます。
- 本動作例では、「Security setting area」「Block protection area0/1」の有効領域の切り替えはサポートしません。「Security setting area」「Block protection area0/1」の書き込み・消去などの処理を行う場合、処理手順は「Configuration setting Area」の書き込み・消去と同様であり、書き換え対象の領域のみ変更が必要です。

表 6-2 メモリ割り付け(Hardware Property Area(Area 0 有効時))

Data Flash 領域	物理アドレス	ブロック	有効/無効	サイズ	書き換え対象
Hardware Property Area0	0xFF32 0000-0xFF32 07FF	Extended Data Area	有効	2KB	非対象
	0xFF32 0800-0xFF32 0FFF	Configuration Setting Area0	有効	2KB	非対象
	0xFF32 1000-0xFF32 17FF	Configuration Setting Area1	無効	2KB	対象
	0xFF32 1800-0xFF32 1FFF	Security Setting Area0	有効	2KB	非対象
	0xFF32 2000-0xFF32 27FF	Security Setting Area1	無効	2KB	非対象
	0xFF32 2800-0xFF32 2FFF	Block Protection Area for FPSYS0 Area0	有効	2KB	非対象
	0xFF32 3000-0xFF32 37FF	Block Protection Area for FPSYS0 Area1	無効	2KB	非対象
	0xFF32 5000-0xFF32 67FF	Erase Counter Area for User Area 0, User Boot Area0	有効	2KB×3	非対象
	0xFF32 6800-0xFF32 7FFF	Erase Counter Area for User Area 1, User Boot Area1	無効	2KB×3	非対象
	0xFF37 3800-0xFF37 37FF	Switch Area 0	有効	2KB	非対象
	0xFF37 4000-0xFF37 47FF	Switch Area 1	無効	2KB	対象
	0xFF37 4800-0xFF37 4FFF	TAG Area	有効	2KB	対象
Hardware Property Area1	0xFF34 0000-0xFF34 0FFF	Block Protection Area for FPSYS1 Area0	有効	2KB	非対象
	0xFF34 0800-0xFF34 0FFF	Block Protection Area for FPSYS1 Area1	無効	2KB	
	0xFF34 1000-0xFF34 27FF	Erase Counter Area for User Area 2, User Boot Area0	有効	2KB×3	
	0xFF34 2800-0xFF34 3FFF	Erase Counter Area for User Area 3, User Boot Area1	有効	2KB×3	

6.1.2 動作イメージ

図 6-1 にプログラムの動作イメージを示します。

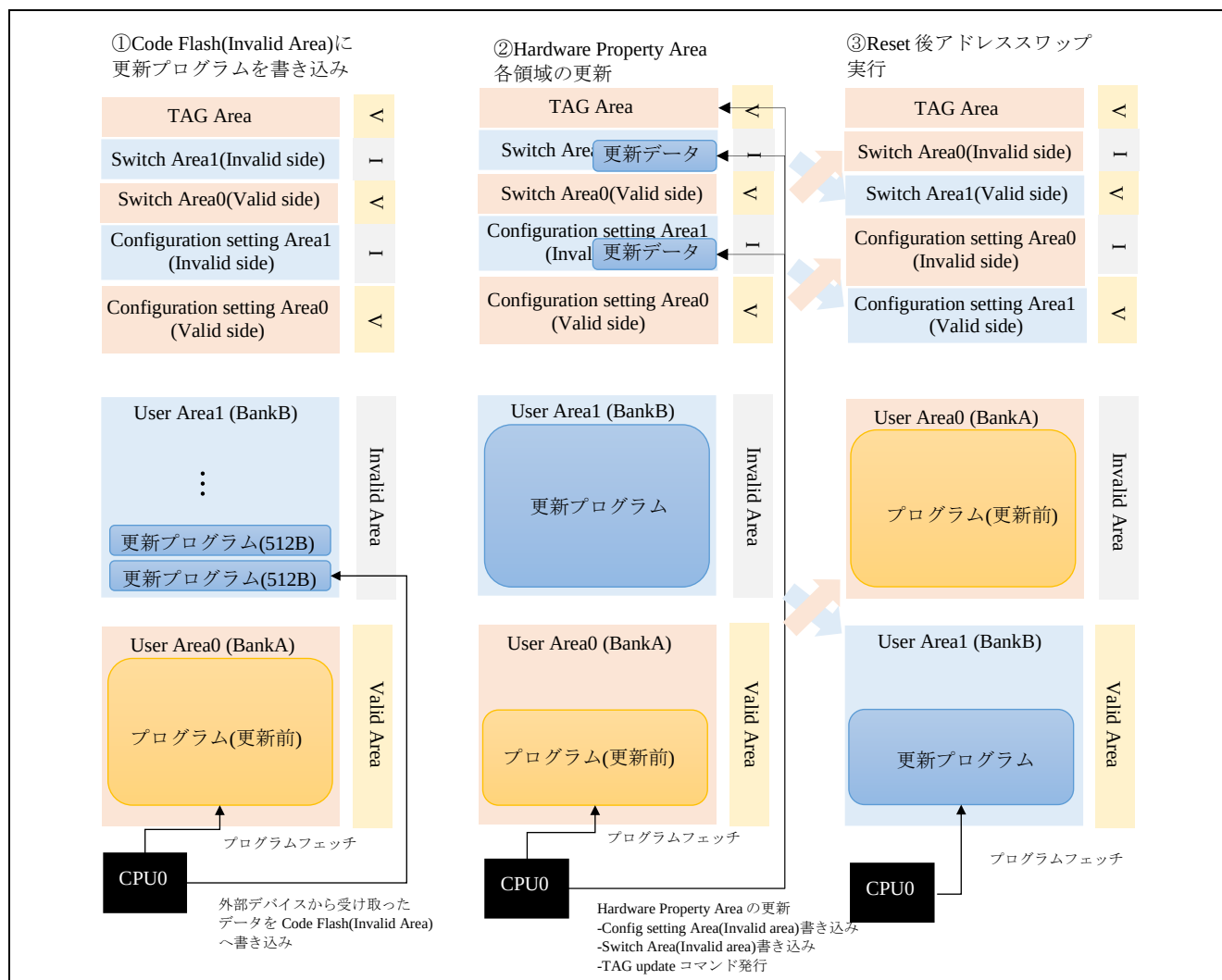


図 6-1 動作イメージ

6.1.3 全体シーケンス

図 6-2、図 6-3、図 6-4 に全体シーケンスを示します。

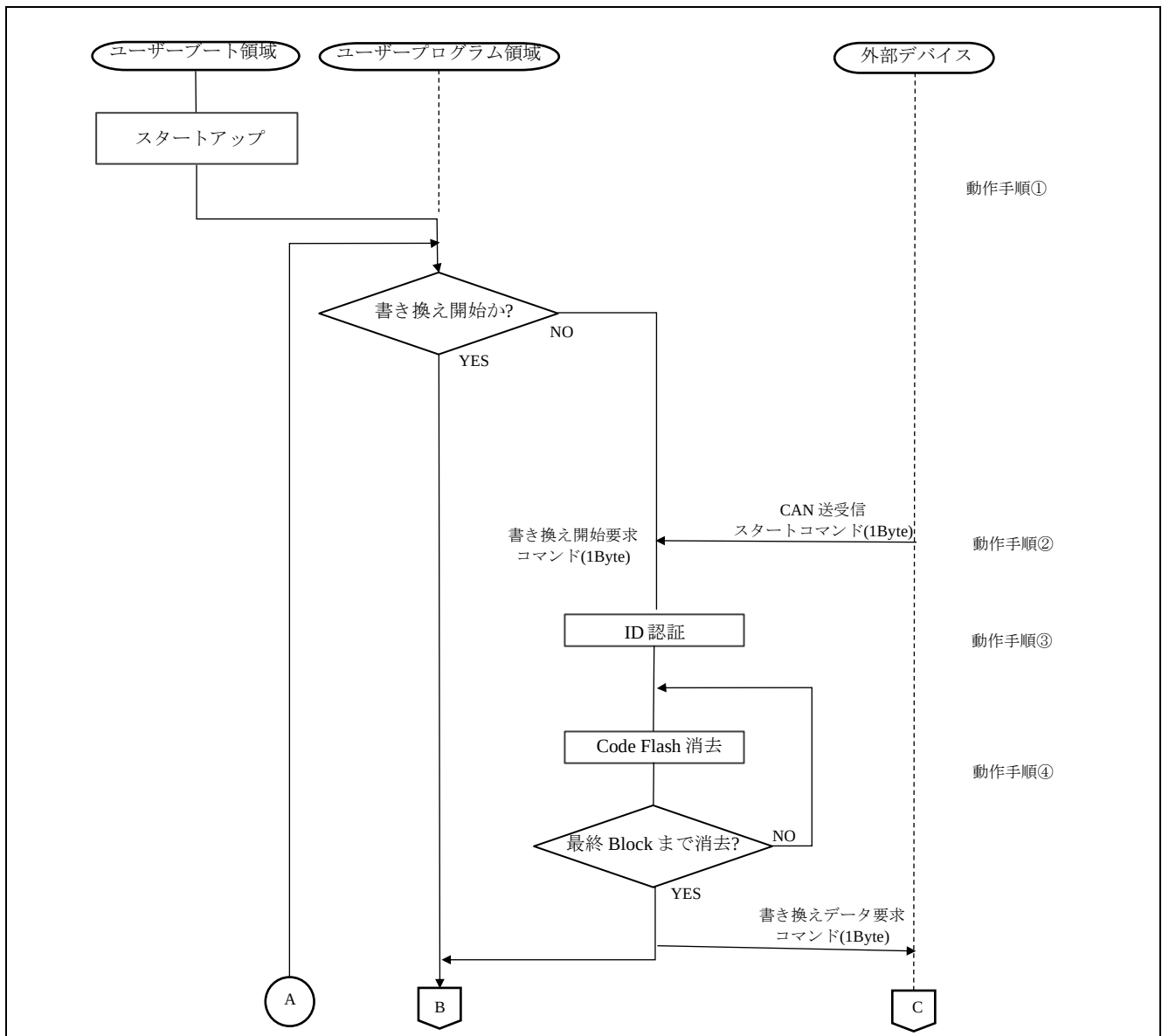


図 6-2 全体シーケンス図(1)

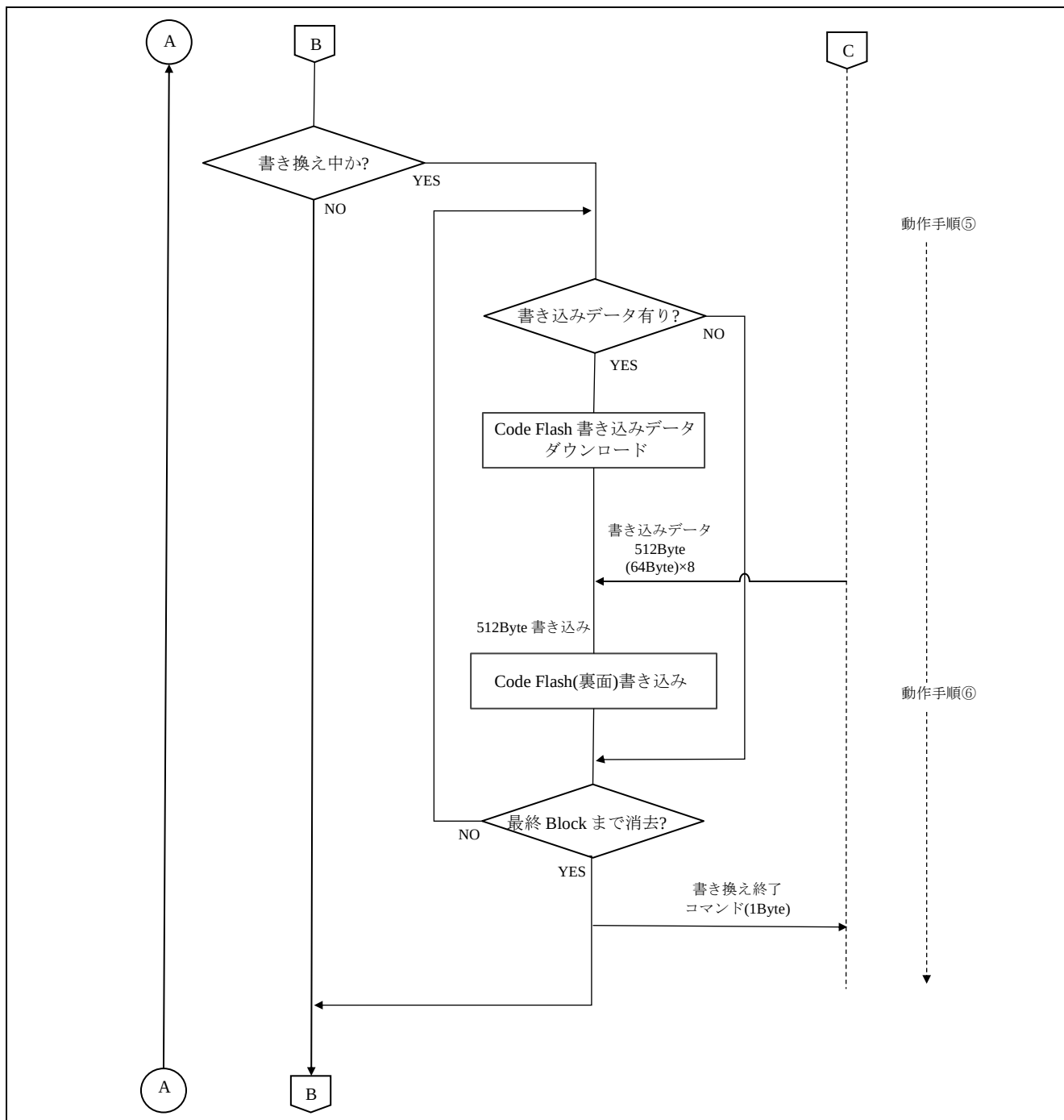


図 6-3 全体シーケンス図(2)

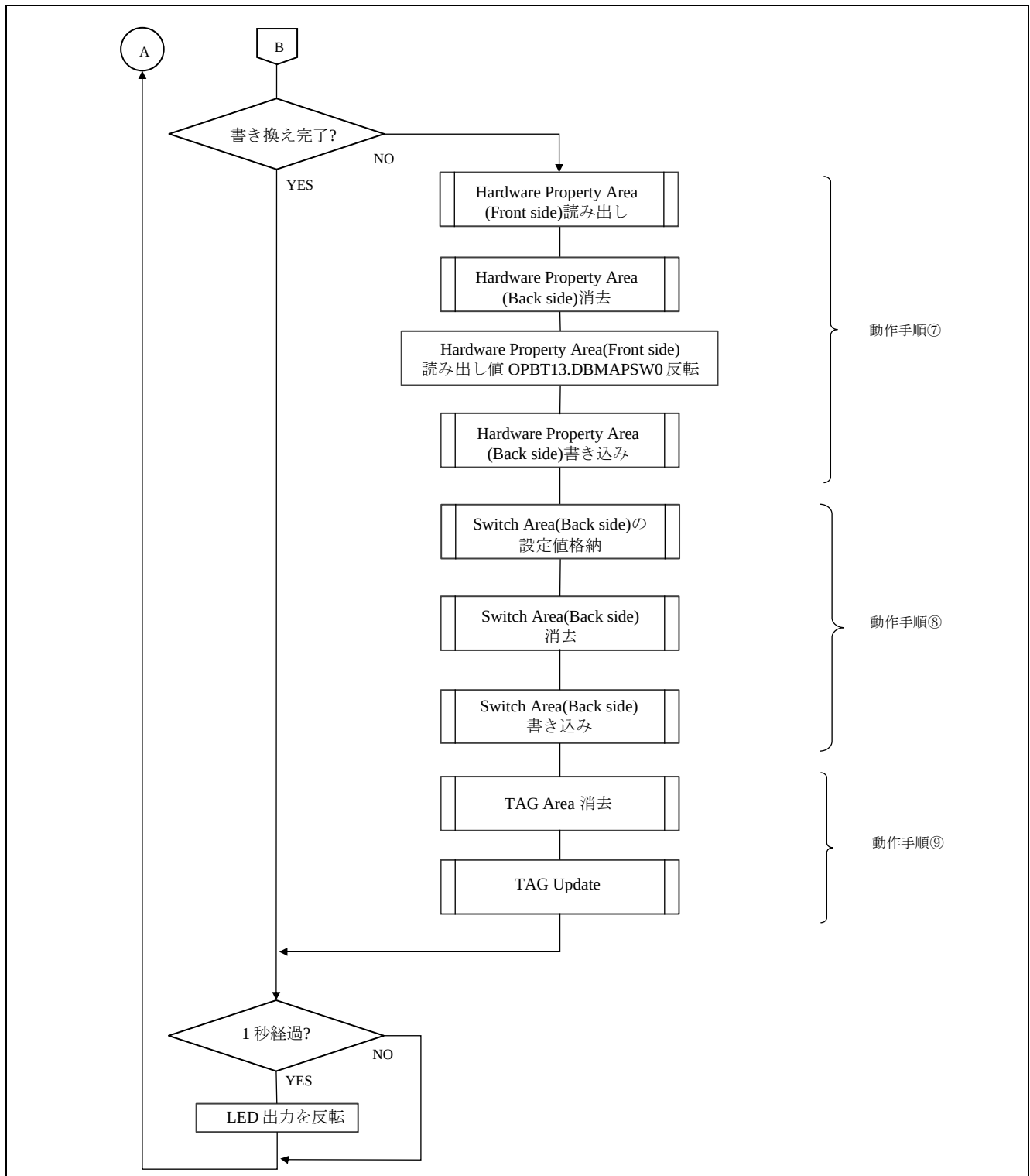


図 6-4 全体シーケンス図(3)

6.1.4 使用機能

- ・ CANFD インタフェース(RS-CANFD)
- ・ FSCI
- ・ PORT
- ・ OSTM

6.1.5 動作モード

本アプリケーションノートでは、Code Flash 書き換え時のマイコンの動作モードは User boot mode0 で行います。User boot mode0 では、CPU0 のリセットベクタアドレスは User boot 領域(0x08000000)となります。

動作モード選択方法は、モード端子で設定します。オプションバイトの設定は、RH850 ファミリ用 Renesas Flash Programmer を使用し設定します。表 6-3 に動作モード選択を示します。

表 6-3 動作モード選択

端子設定値			オプションバイト設定値		動作モード	起動領域
FLMD0	FLMD1	_TRST	STMSEL1	STMSEL0		
0	X	0	0	1	User boot mode0	User boot 領域

6.1.6 メモリマッピング

本アプリケーションノートでは、Code Flash 書き換え時のメモリマッピングは Double map mode で行います。表 6-4 にメモリマッピング選択を示します。

表 6-4 メモリマッピング選択

オプションバイト設定値		メモリマッピング
MAPMODE1	MAPMODE 0	
0	0	Double map mode

6.1.7 RFD コンフィギュレーション

本アプリケーションノートで使用する RFD のマクロ定義一覧を、表 6-5 マクロ定義一覧に示します。

表 6-5 マクロ定義一覧

マクロ定義	設定値	概要
R_RFD_VALUE_FORCED_STOP_TIMEOUT	サンプルコード参照	強制的終了コマンドのタイムアウト時間
R_RFD_ERASURE_SUSPENDED_MODE	R_RFD_ERASURE_PRIORITY_MODE	消去優先モードにする
R_RFD_REG_FAEINT_CFAEIE R_RFD_REG_FAEINT_CMDLKIE R_RFD_REG_FAEINT_DFAEIE R_RFD_REG_FAEINT_ECRCTIE	R_RFD_DISABLE	FLERR 割り込みを無効にする
R_RFD_CONTROL_TARGET_DATAFLASH	R_RFD_ENABLE	Data flash にアクセスする
R_RFD_CONTROL_TARGET_CODEFLASH	R_RFD_ENABLE	Code flash にアクセスする
R_RFD_MAPMODE	R_RFD_DOUBLE	マップモードは Double map mode
R_RFD_BLOCK_PROTECTION_AREA1	R_RFD_ENABLE	Block protection area1 はアクセス対象
R_RFD_FPMON_CHECK	R_RFD_ENABLE	FPMON チェックを有効にする

6.2 動作手順

動作手順①～⑨は、「6.1.3 全体シーケンス」に対応します。

6.2.1 ① スタートプログラム

リセットスタート後、User boot 領域上のスタートプログラムを実行します。図 6-5 に スタートプログラム動作を示します。

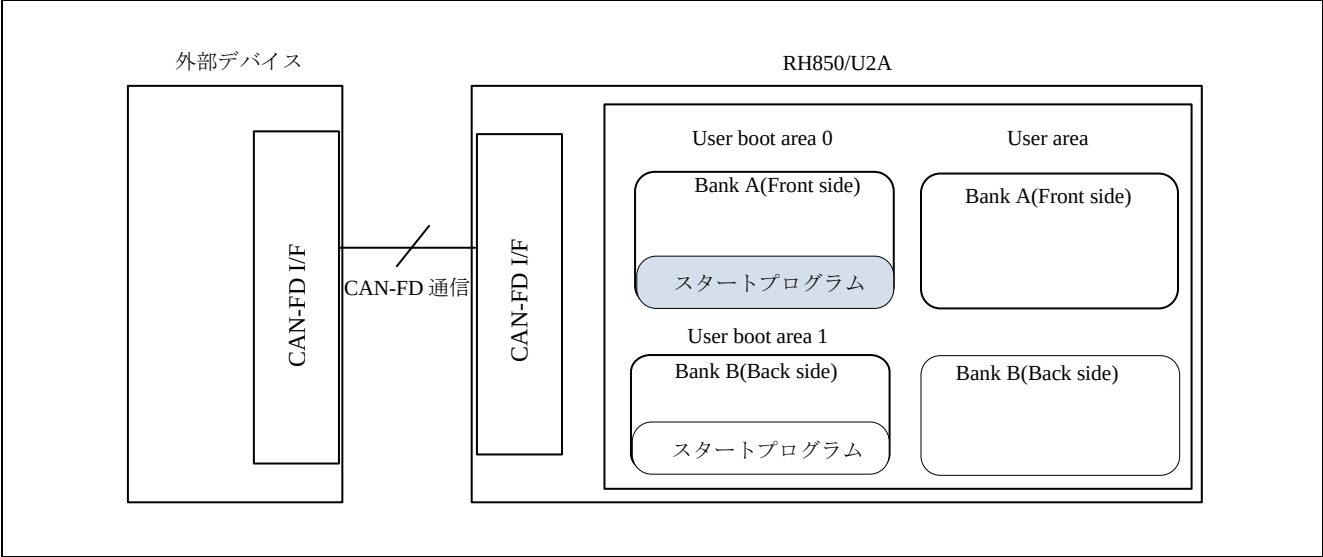


図 6-5 スタートアッププログラム動作

表 6-6 「main_pm0()関数」

関数名	引数	概要
main_pm0()	無し	プログラムを開始し、ユーザープログラムにジャンプします。

図 6-6 に「main_pm0()関数」のフローチャートを示します。

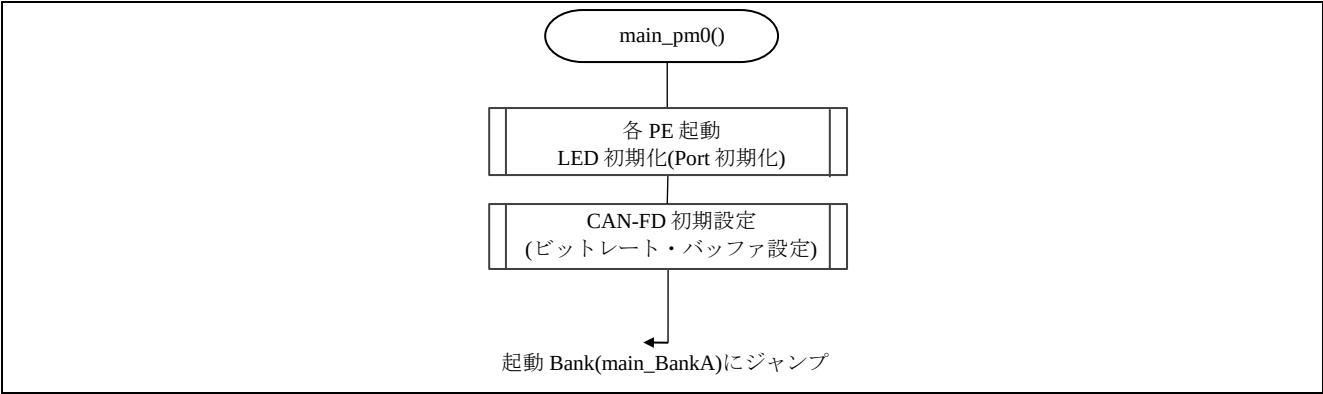


図 6-6 「main_pm0()関数」のフローチャート
「動作手順①」

6.2.2 ② BGO 書き込み

Code Flash の無効領域が書き換え対象の領域です。

図 6-7 に Bank A が有効領域時の Code Flash 消去/書き込み動作を示します。

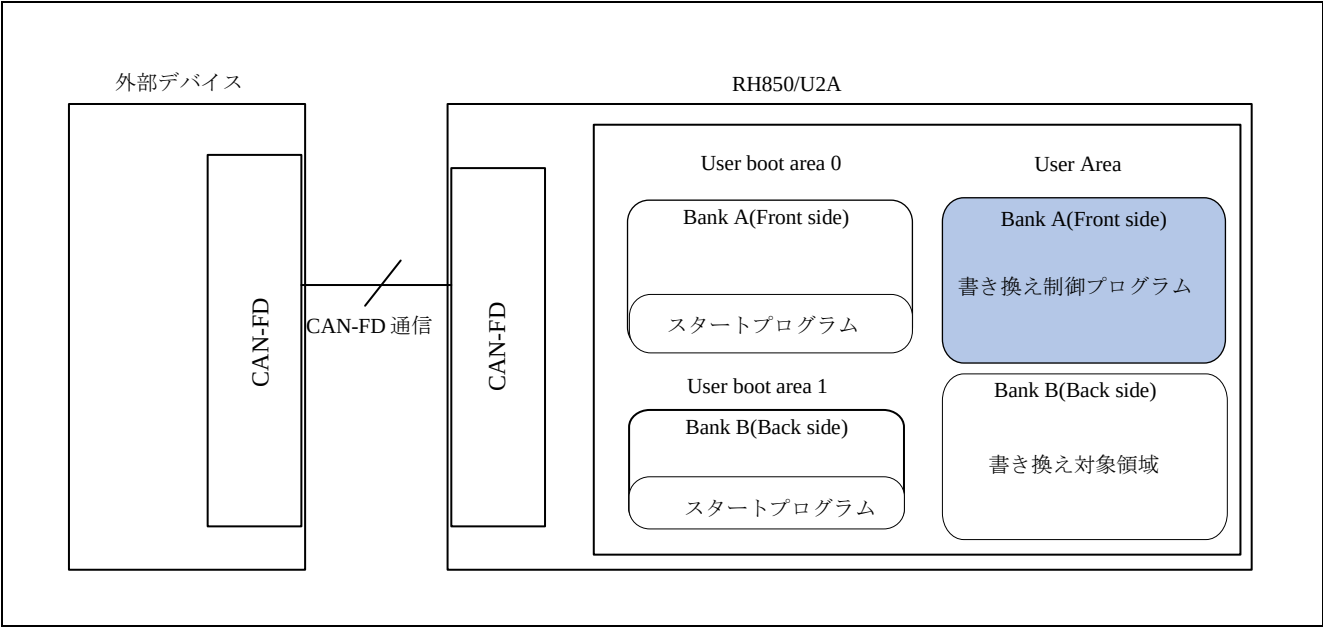


図 6-7 Code Flash 消去/書き換え開始動作

関数説明

表 6-7 「main_BankA()関数」

関数名	引数	概要
main_BankA()	無し	LED の点滅を繰り返し、書き換え開始コマンドを受信後、「書き換え制御プログラム」が実行されます。

図 6-8 に「main_BankA()関数」のフローチャートを示します。

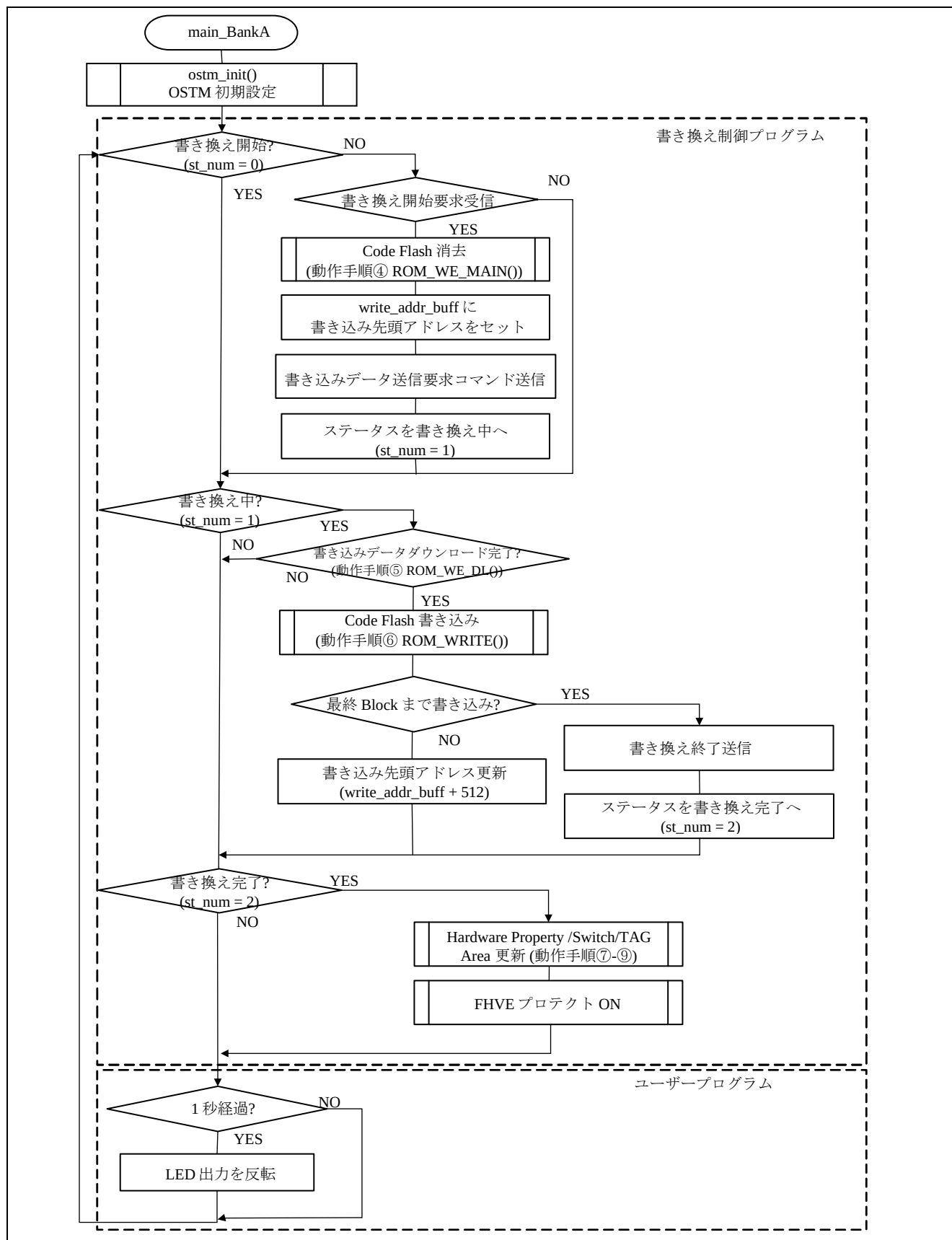


図 6-8 「main_BankA()関数」のフローチャート

「動作手順②」

関数説明

表 6-8 「ROM_WE_MAIN()関数」

関数名	引数	概要
ROM_WE_MAIN()	無し	ID 認証、Code Flash 消去、Code Flash 書き込みデータダウンロード、Code Flash 書き込みの各種関数コールを実施します。

図 6-9 に「ROM_WE_MAIN()関数」のフローチャートを示します。

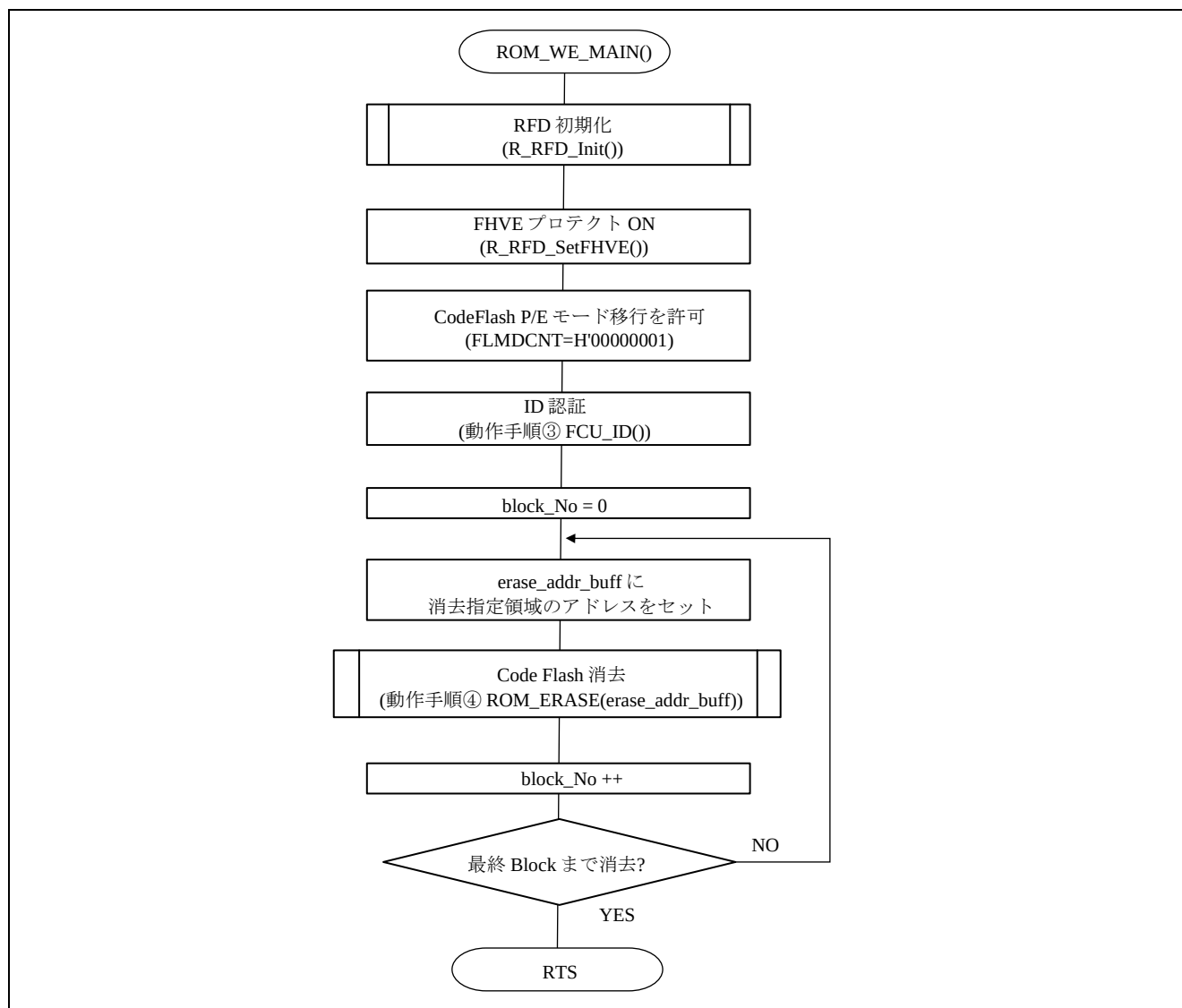


図 6-9 「ROM_WE_MAIN()関数」のフローチャート
「動作手順③、④を含む」

6.2.3 ③ ID 認証

「書き換え制御プログラム」の「ID 認証関数」は Flash メモリの特権領域にあらかじめ設定した 256 ビットの ID と Customer ID の値を比較することで ID の認証を実行します。

本アプリケーションノートでは、ID 設定は 1 バイト目を"0"、1 バイト目以外を"F"の設定で使用しています。ID 設定を変更する際には RH850 ファミリー用 Renesas Flash Programmer または、コンフィグレーション設定コマンドを使用し設定します。

関数説明

表 6-9 「FCU_ID()関数」

関数名	引数		概要
FCU_ID()	無し		Flash メモリの特権領域に設定された ID との比較と ID 認証を実行します。
API 名	引数		概要
R_RFD_IDAuth	T_en_IDType	R_RFD_ID_CUSTIDA/B/C	Flash メモリの特権領域に設定された ID との比較と ID 認証を実行します。
	T_u1*	ID データのポインタ	

図 6-10 に「FCU_ID()関数」のフローチャートを示します。

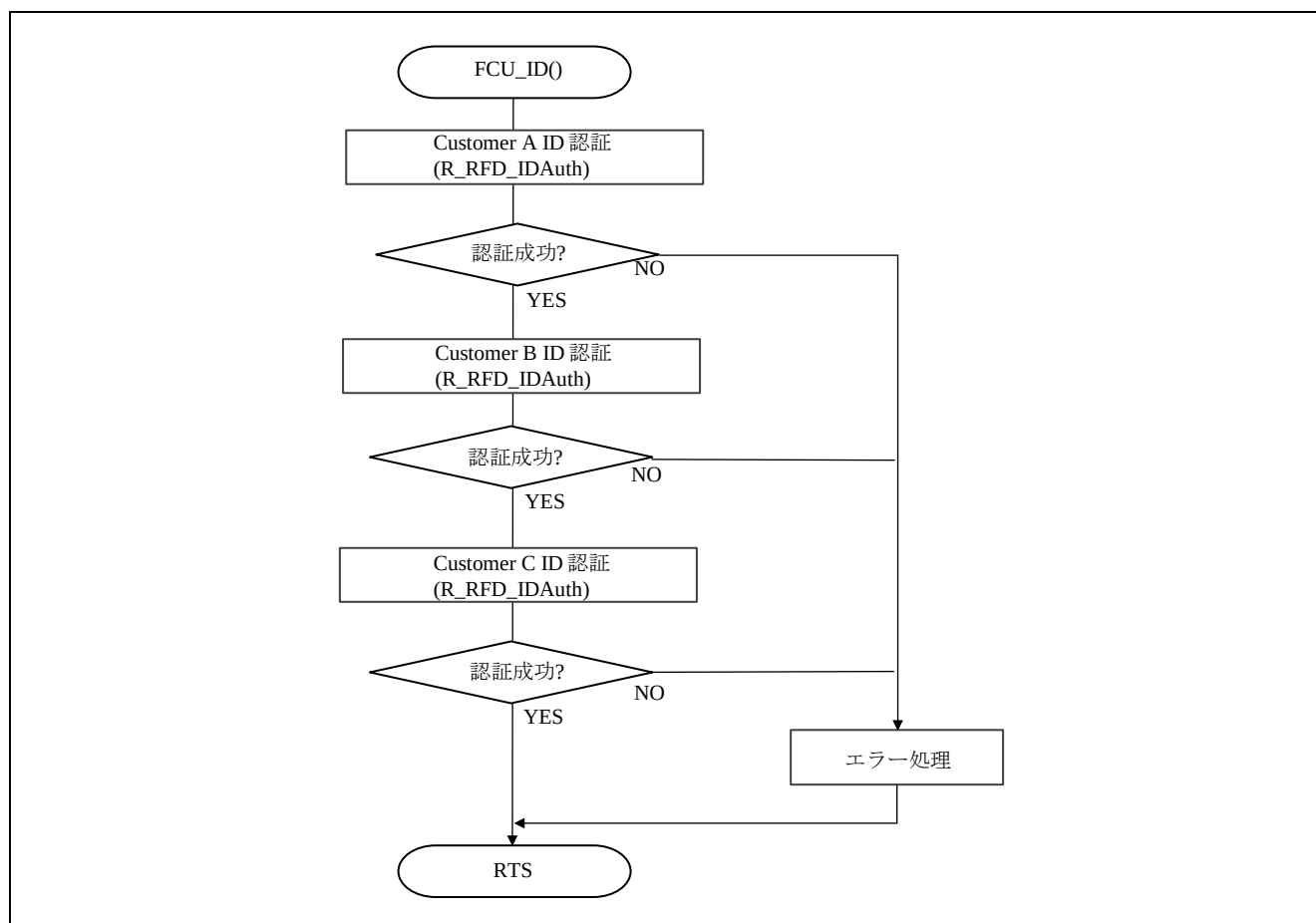


図 6-10 「FCU_ID()関数」のフローチャート

「動作手順③」

6.2.4 ④ Code Flash 消去

ID 認証後、「書き換え制御プログラム」の「Code Flash 消去関数」を実行。

FACI コマンド発行領域にブロックイレーズコマンドを発行し、Code Flash 書き換え指定領域を消去します。

関数説明

表 6-10 「ROM_ERASE()関数」

関数名	引数		概要
ROM_ERASE()	erase_addr	消去する Code Flash 領域の先頭アドレス	Code Flash 書き換え指定領域を消去します
API 名	引数		概要
R_RFD_ShiftToPEMode	T_u2_FACI	R_RFD_FACI0	P/E モードへ遷移
	T_en_FACIMode	R_RFD_MODE_CFPE	
R_RFD_ShiftToReadMode	T_u2_FACI	R_RFD_FACI0	Read モードへ遷移
R_RFD_GetFaciStatus	T_u2_FACI	R_RFD_FACI0	コマンドロックの状態及びエラーチェック
R_RFD_GetFaciSequenceReady	T_u2_FACI	R_RFD_FACI0	FRDY ビットの状態チェック
R_RFD_StatusClear	T_u2_FACI	R_RFD_FACI0	FACI ステータスクリア
R_RFD_ForcedStopAndErrorClear	T_u2_FACI	R_RFD_FACI0	強制終了コマンド発行
R_RFD_EraseCFRequest	T_u4_RfdAddress	消去領域の先頭アドレス	Code Flash 消去コマンド発行

図 6-11 に ROM_ERASE()関数」のフローチャートを示します。

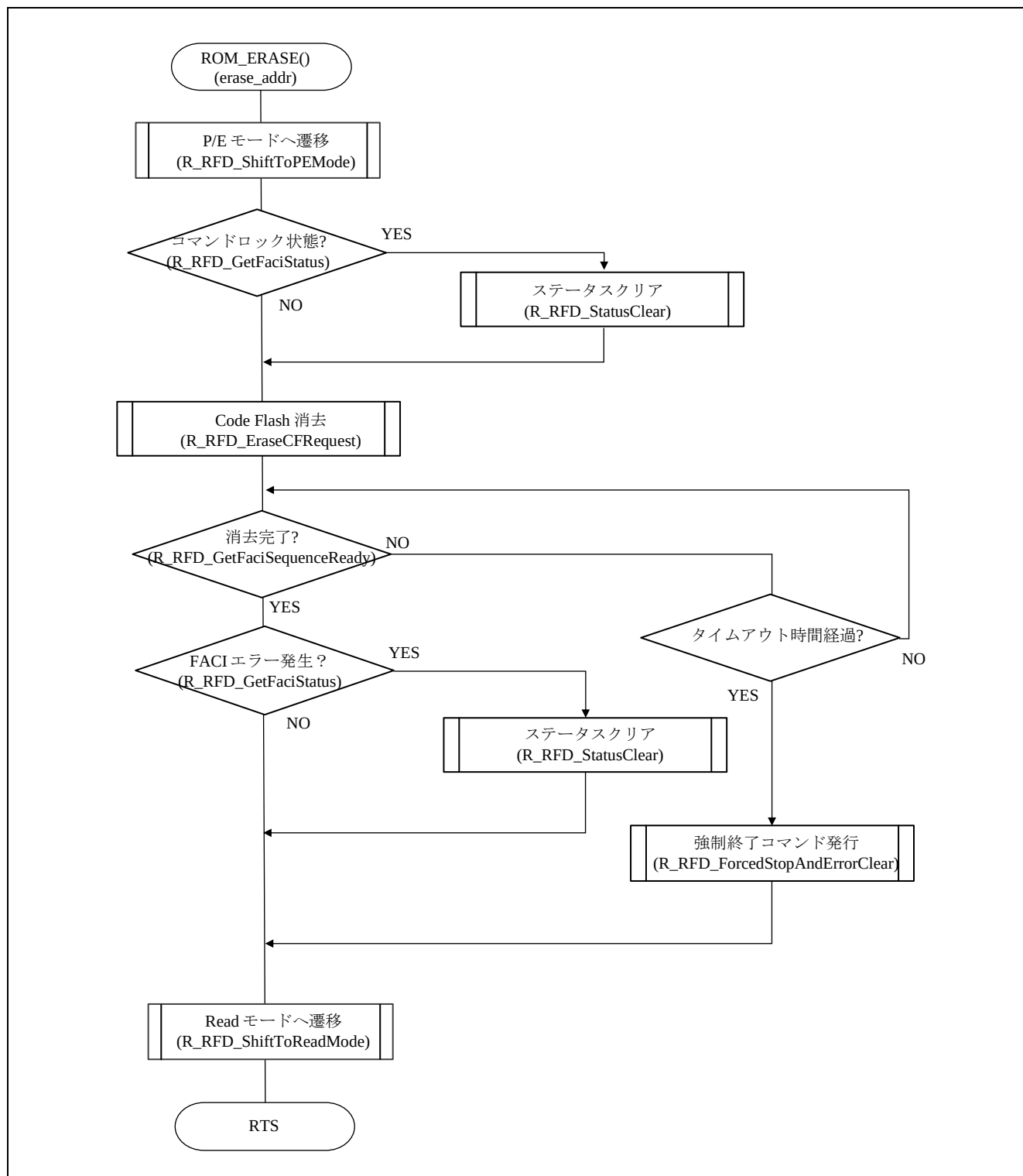


図 6-11 「ROM_ERASE()関数」のフローチャート
「動作手順④」

6.2.5 ⑤ Code Flash 書き込みデータダウンロード

「Code Flash 書き込みデータダウンロード関数」を実行し、書き込みデータ要求コマンドを送信します。書き込みデータ要求コマンドを受信した外部デバイスは、書き込みデータダウンロードコマンドで書き込みデータ 512Byte をマイコンへ送信します。「Code Flash 書き込みデータダウンロード関数」では受信した書き込みデータを RAM へ保存します。

関数説明

表 6-11 「ROM_WE_DL()関数」

関数名	引数	概要
ROM_WE_DL()	無し	外部デバイスから書き込みデータをダウンロード

図 6-12 に「ROM_WE_DL()関数」のフローチャートを示します。

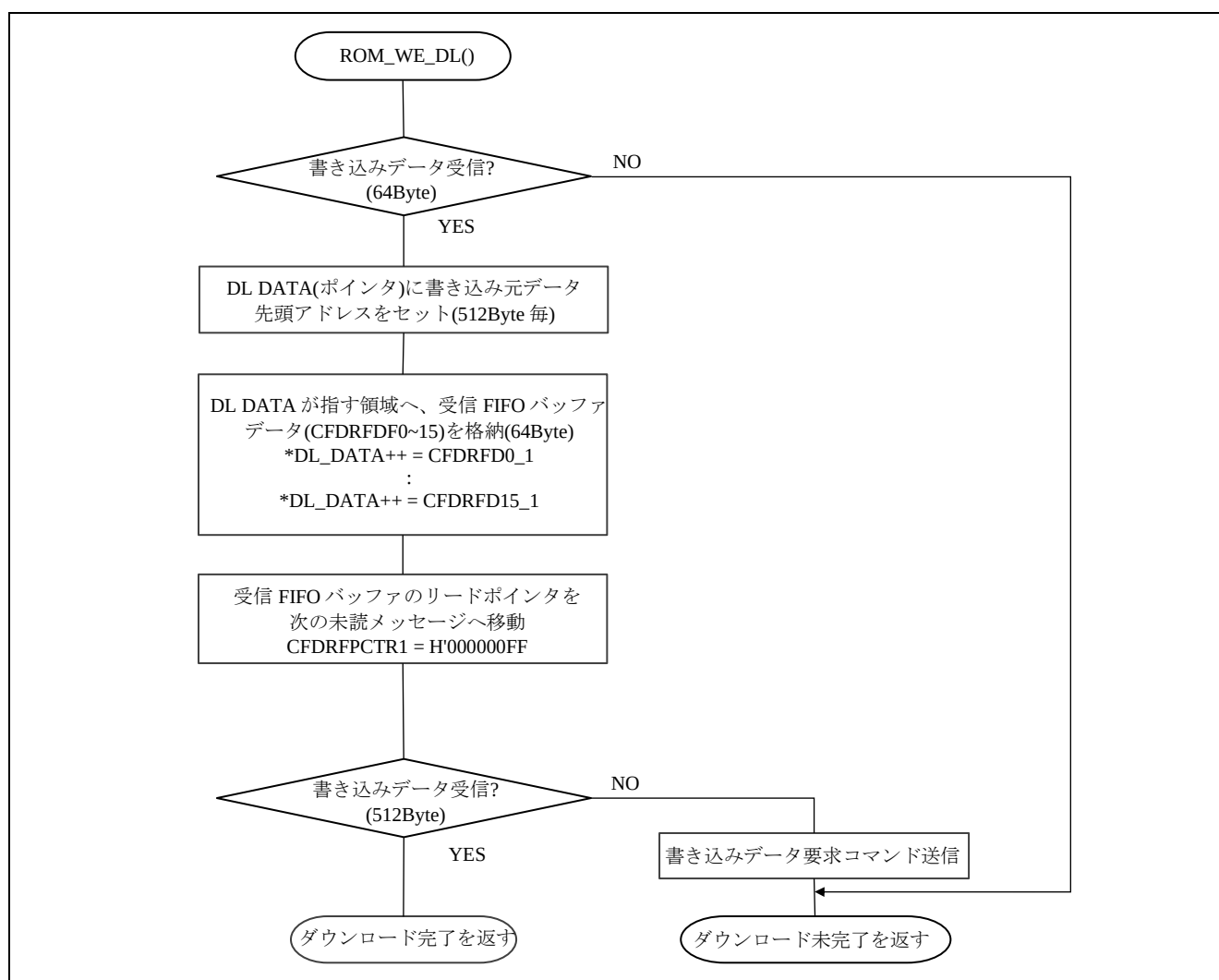


図 6-12 「ROM_WE_DL()関数」のフローチャート
「動作手順⑤」

6.2.6 ⑥ Code Flash 書き込み

CAN 通信で外部デバイスから受信した書き込みデータは、「Code Flash 書き換え関数」を使用して Code Flash に書き込みます。

FACI コマンド発行領域にプログラム・コマンドを発行し、Code Flash 書き込み指定領域に書き込みます。書き込みサイズ（64K バイト）に達した場合、Flash 書き換えを終了します。

関数説明

表 6-12 「ROM_WRITE()関数」

関数名	引数		概要
ROM_WRITE()	write_addr	書き込む Code Flash 領域の先頭アドレス	Code Flash 書き換え指定領域に書き込みます (512Byte 単位)
API 名	引数		概要
R_RFD_ShiftToPEMode	T_u2_FACI	R_RFD_FACI0	P/E モードへ遷移
	T_en_FACIMode	R_RFD_MODE_CFPE	
R_RFD_ShiftToReadMode	T_u2_FACI	R_RFD_FACI0	Read モードへ遷移
R_RFD_GetFaciStatus	T_u2_FACI	R_RFD_FACI0	コマンドロックの状態及びエラーチェック
R_RFD_GetFaciSequenceRead	T_u2_FACI	R_RFD_FACI0	FRDY ビットの状態チェック
R_RFD_StatusClear	T_u2_FACI	R_RFD_FACI0	FACI ステータスクリア
R_RFD_ForcedStopAndErrorClear	T_u2_FACI	R_RFD_FACI0	強制終了コマンド発行
R_RFD_WriteCFRequest	T_u4_RfdAddress	書き込み領域の先頭アドレス	Code Flash 書き込みコマンド発行

図 6-13 に「ROM_WRITE()関数」のフローチャートを示します。

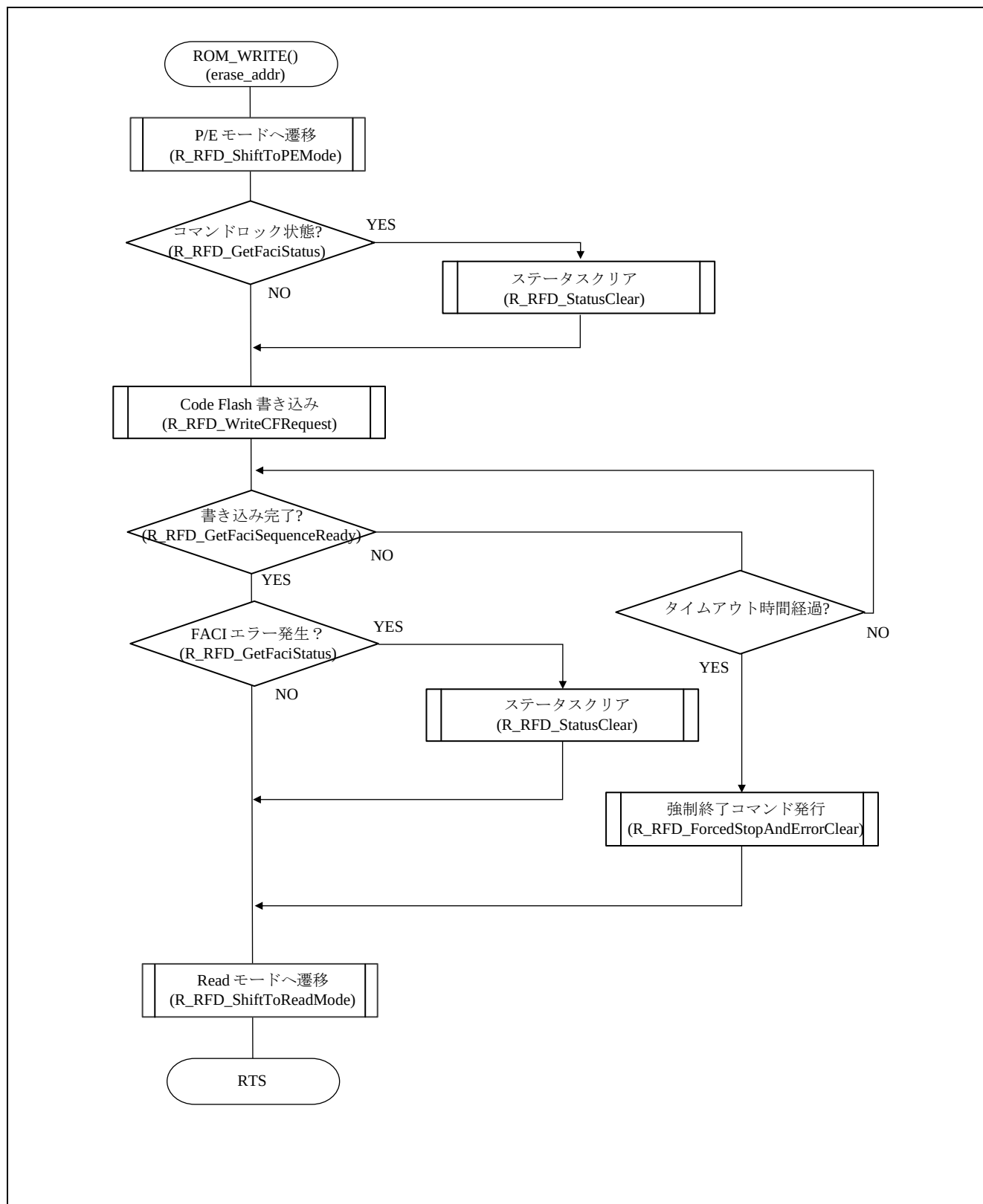


図 6-13 「ROM_WRITE()関数」のフローチャート

「動作手順⑥」

6.2.7 ⑦ Hardware Property Area の更新

「Sample_PropertyAreaControl」関数を実行することで、Data Flash の Hardware Property Area の書き換え、Switch Area の書き換え、TAG Area の書き換えによりバンクスワップを実行します。なお、本関数で扱う Hardware Property Area は「Configuration setting Area」のみです。「Security setting area」「Block protection area0/1」を操作する場合は処理手順は同様であり、対象の領域のみの変更となります。「Security setting area」「Block protection area0/1」を操作する場合はそれぞれに応じた処理を行ってください。

表 6-13 「Sample_PropertyAreaControl ()関数」

関数名	引数		概要
Sample_PropertyAreaControl	N/A	-	バンクスワップを実行します
API 名	引数		概要
R_RFD_ShiftToPEMode	T_u2_FACI	R_RFD_FACI0	P/E モードへ遷移
	T_en_FACIMode	R_RFD_MODE_DFPE	
R_RFD_ShiftToReadMode	T_u2_FACI	R_RFD_FACI0	Read モードへ遷移
R_RFD_GetFaciStatus	T_u2_FACI	R_RFD_FACI0	コマンドロックの状態及びエラーチェック
R_RFD_GetFaciSequenceReady	T_u2_FACI	R_RFD_FACI0	FRDY ビットの状態チェック
R_RFD_StatusClear	T_u2_FACI	R_RFD_FACI0	FACI ステータスクリア
R_RFD_ErasePropertyRequest	T_u4_RfdAddress	消去領域の先頭アドレス	Config setting area 消去コマンド発行
R_RFD_WritePropertyRequest	T_u4_RfdAddress	書き込み領域の先頭アドレス	Config setting area 書き込みコマンド発行
	T_pu4_RfdBuffer	書き込みデータ格納領域の先頭アドレス	
R_RFD_EraseSwitchRequest	N/A	-	Switch area 消去コマンド発行
R_RFD_WriteSwitchRequest	T_pu4_RfdBuffer	書き込みデータ格納領域の先頭アドレス	Switch area 書き込みコマンド発行
R_RFD_EraseTagRequest	N/A	-	TAG area 消去コマンド発行
R_RFD_UpdateTagRequest	N/A	-	TAG update コマンド発行
R_RFD_ForcedStopAndErrorClear	T_u2_FACI	R_RFD_FACI0	強制終了コマンド発行

(1) Configuration Setting Area (Front side)読み出し

Configuration setting area の Front side のデータを読み出し、Configuration setting area の Back side への書き込みデータを作成します。

図 6-14 に「Sample_PropertyAreaControl()関数」の Configuration Setting Area (Front side)読み出しのフローチャートを示します。

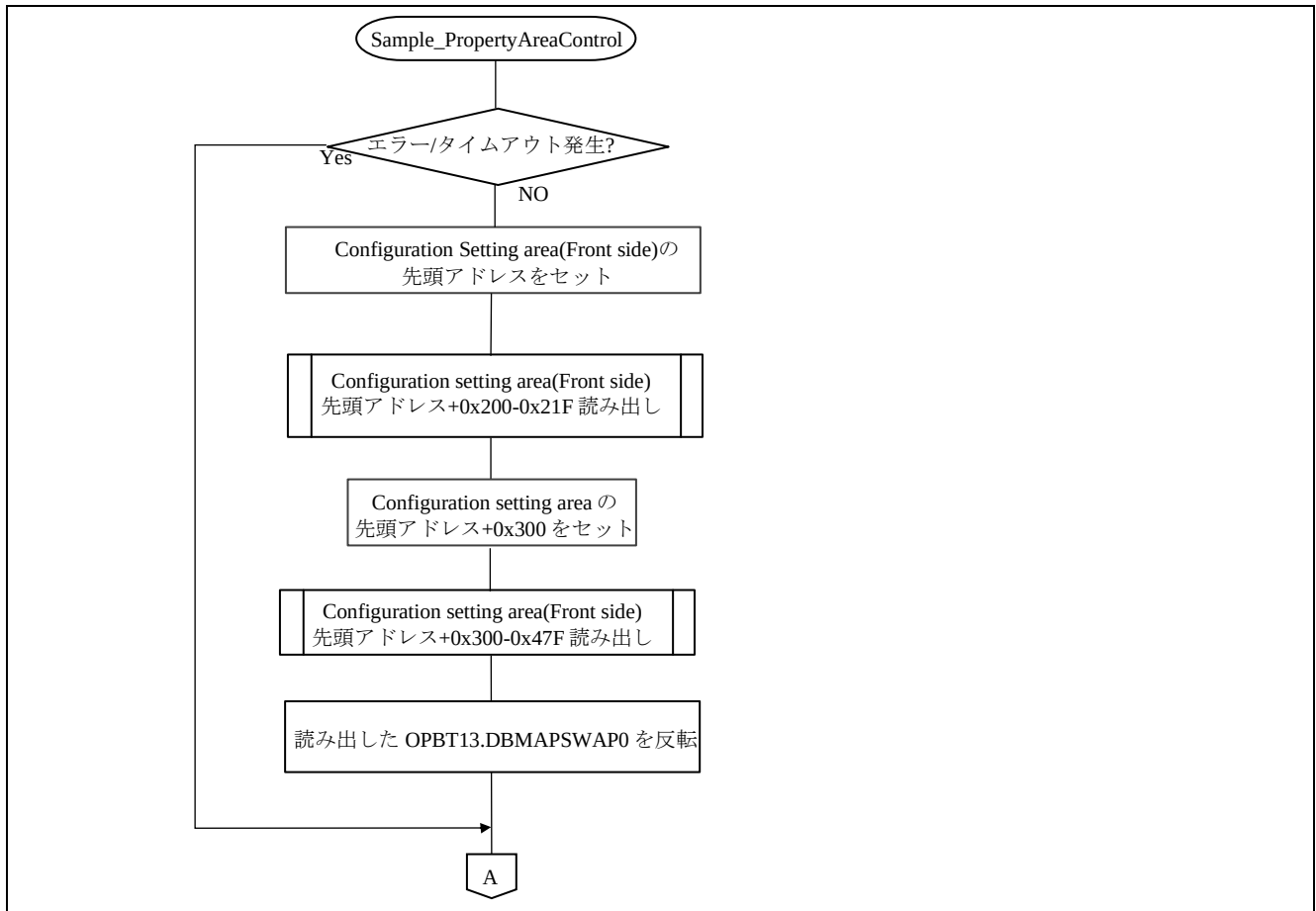


図 6-14 Configuration Setting Area (Front side)読み出しのフローチャート

(2) Configuration Setting Area (Back side)消去

Configuration setting area の Back side のデータ消去を実行します。

図 6-15 に「Sample_PropertyAreaControl()関数」の Configuration setting area の Back side 消去のフローチャートを示します。

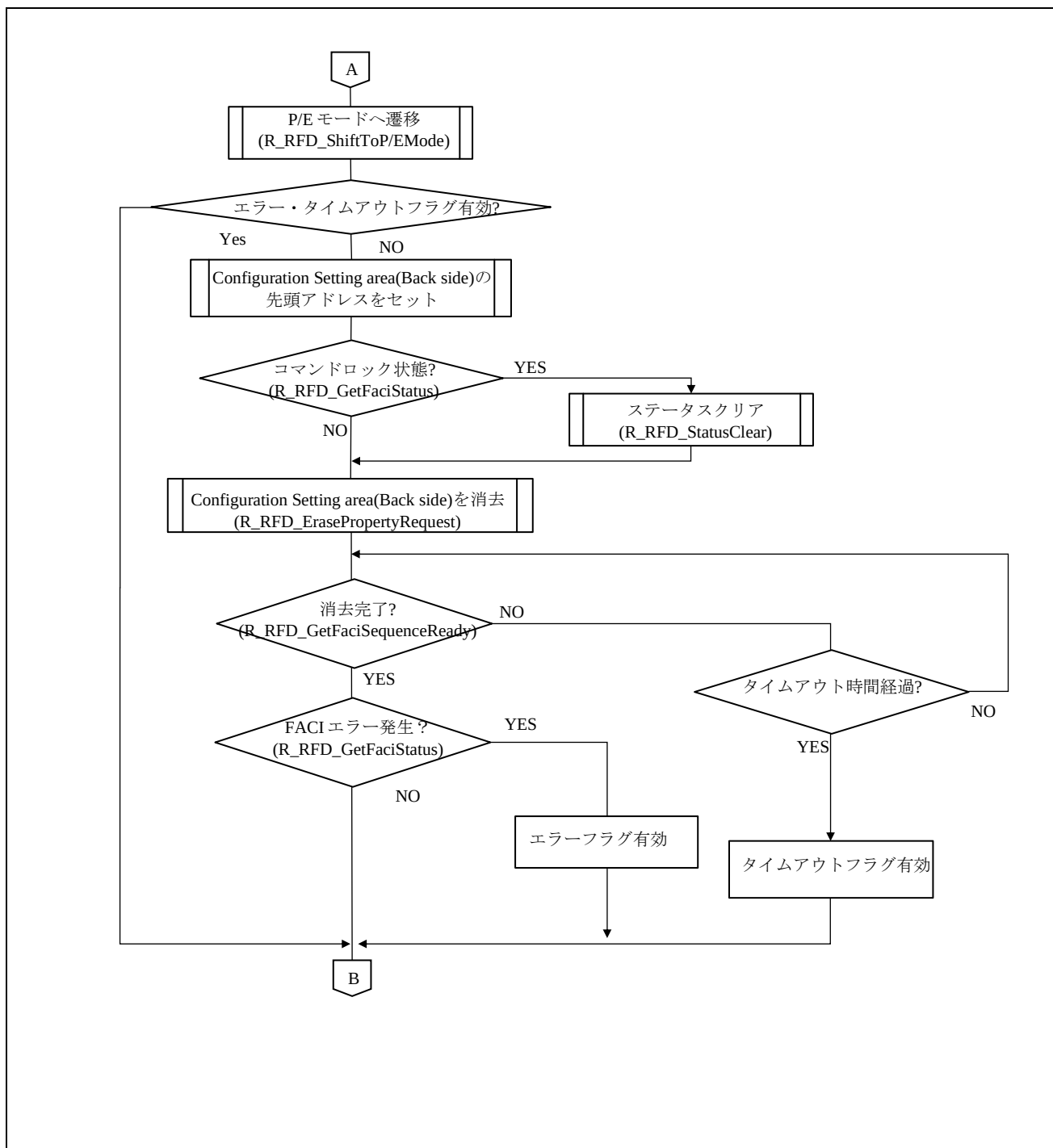


図 6-15 Configuration setting area の Back side 消去のフローチャート

(3) Configuration Setting Area (Back side)書き込み

Configuration setting area の Back side のデータ書き込みを実行します。

図 6-16、図 6-17 に「Sample_PropertyAreaControl()関数」の Configuration setting area(Back side)書き込みのフローチャートを示します。

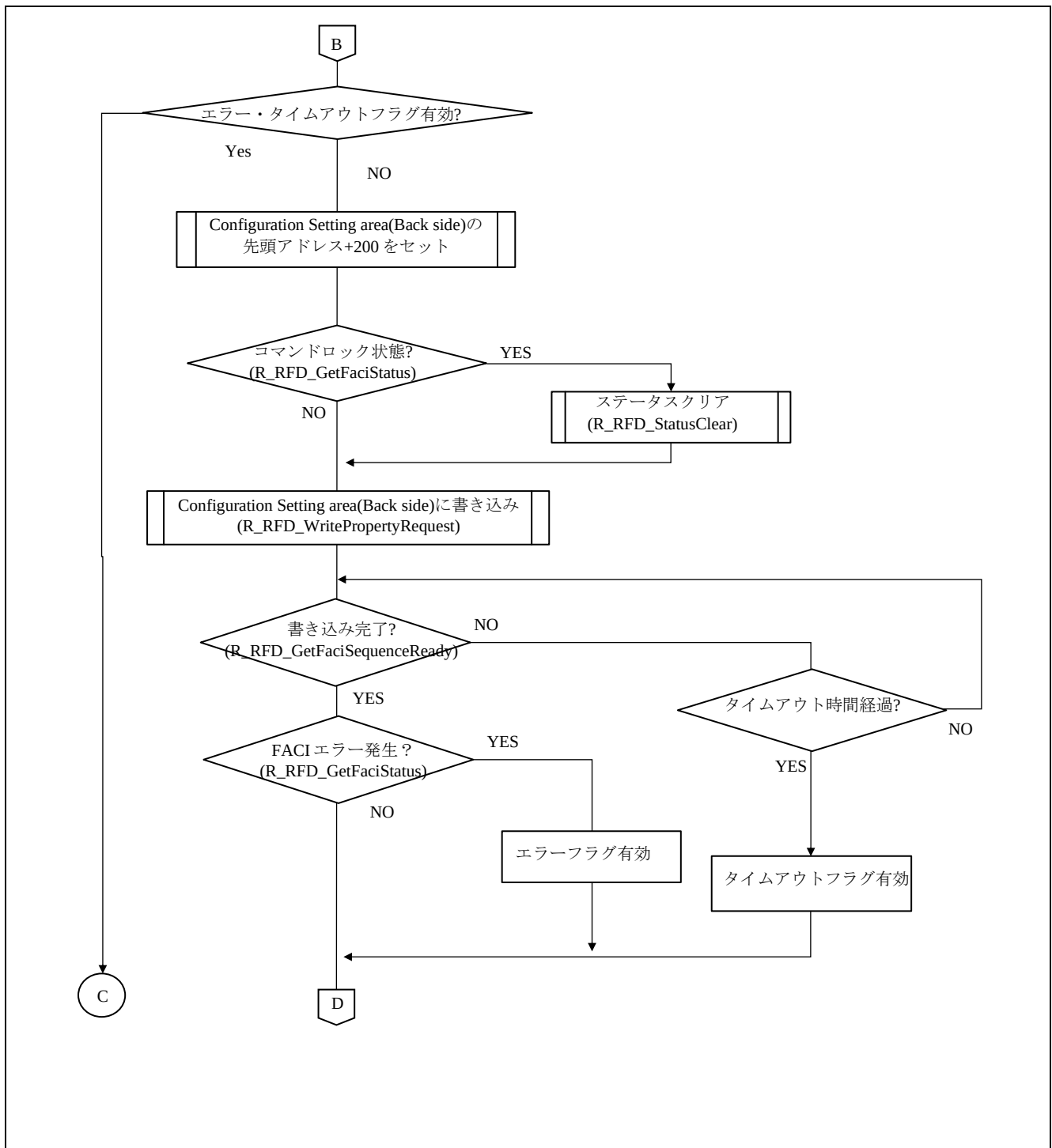


図 6-16 Configuration setting area(Back side)書き込みのフローチャート(1)

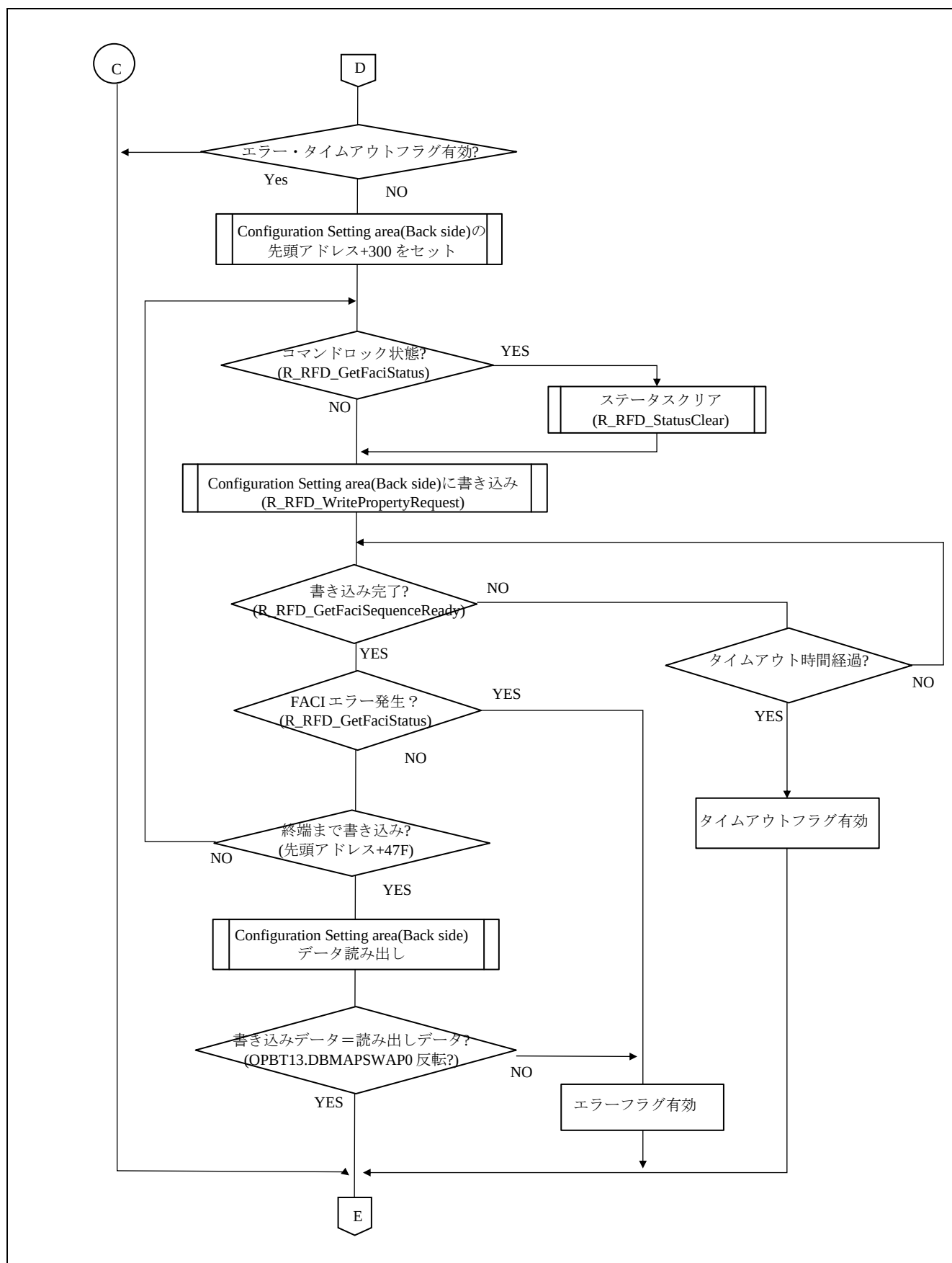


図 6-17 Configuration setting area(Back side)書き込みのフローチャート(2)

6.2.8 ⑧ Switch Area の更新

(1) Switch Area(Back side)の設定値格納

FSWASTAT_0レジスタを読み出し、Configuration Setting Area, Security Setting Area, Block Protection Area0/1の有効領域のステータスを確認し、それぞれの領域の有効にする領域を設定します。本アプリケーションノートでは、Configuration Setting Areaのみ有効にする領域を反転します。「Security setting area」「Block protection area0/1」を操作する場合の処理手順は同様であり、対象の領域のみの変更となります。「Security setting area」「Block protection area0/1」を操作する場合はそれぞれに応じた処理を行ってください。

図 6-18 に「Sample_PropertyAreaControl()関数」の Switch Area(Back side)の設定値格納のフローチャートを示します。

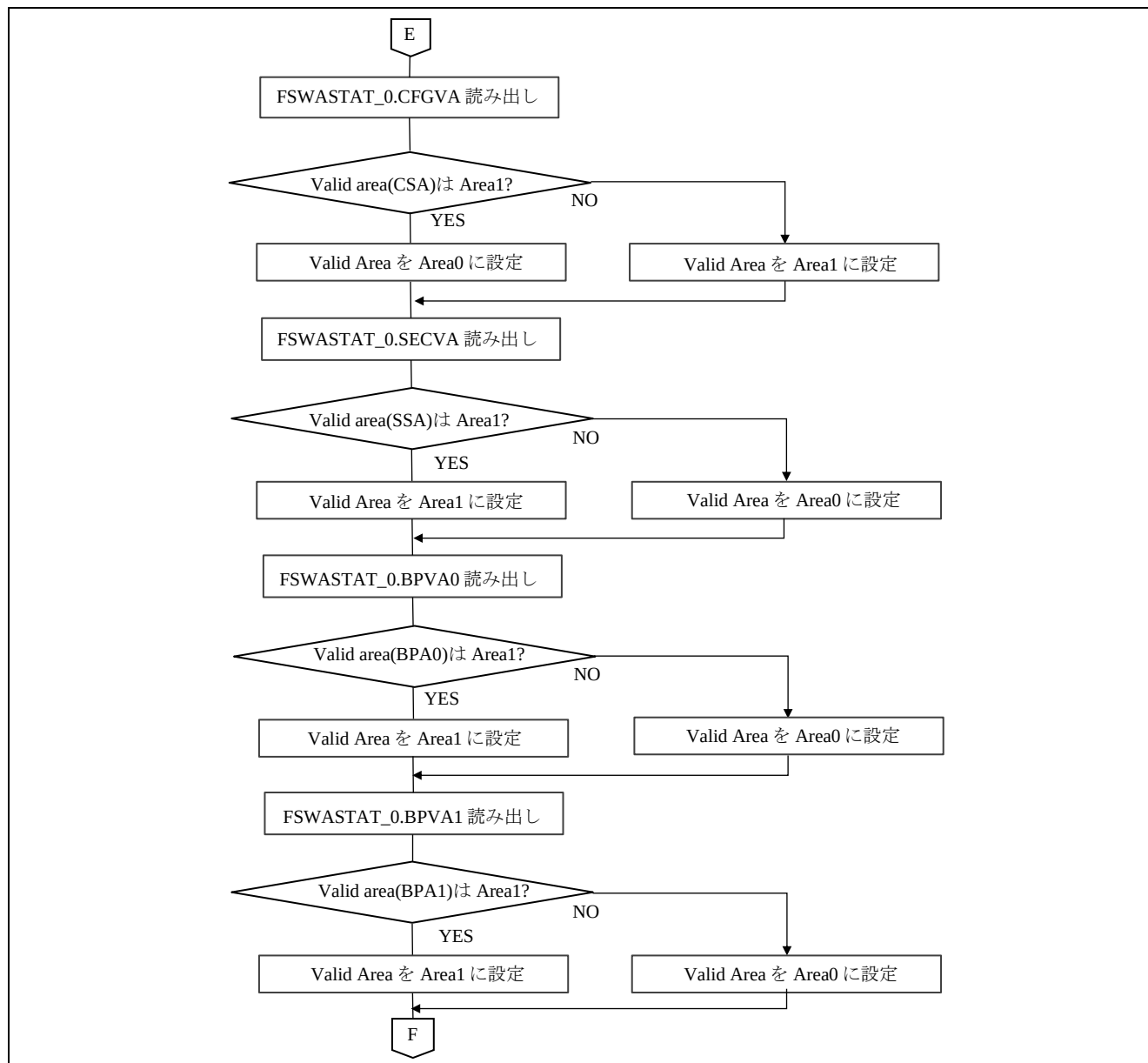


図 6-18 Switch Area(Back side)の設定値格納のフローチャート

(2) Switch Area(Back side)消去

Switch area の Back side の消去を実行します。

図 6-19 に「Sample_PropertyAreaControl()関数」の Switch area(Back side)消去のフローチャートを示します。

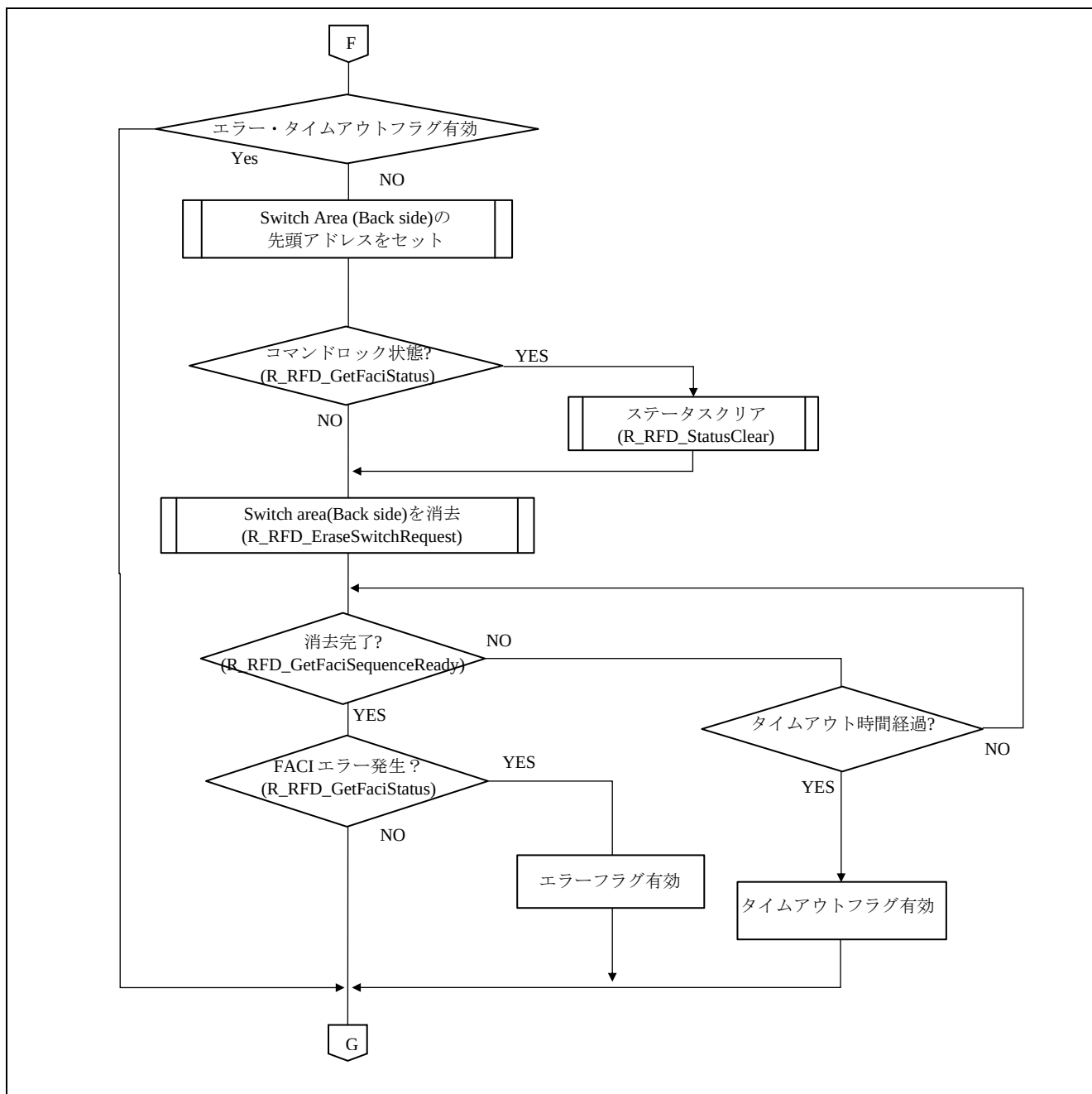


図 6-19 Switch area(Back side)消去のフローチャート

(3) Switch Area(Back side)書き込み

Switch area の Back side の書き込みを実行します。

図 6-20 に「Sample_PropertyAreaControl()関数」の Switch area(Back side)書き込みのフローチャートを示します。

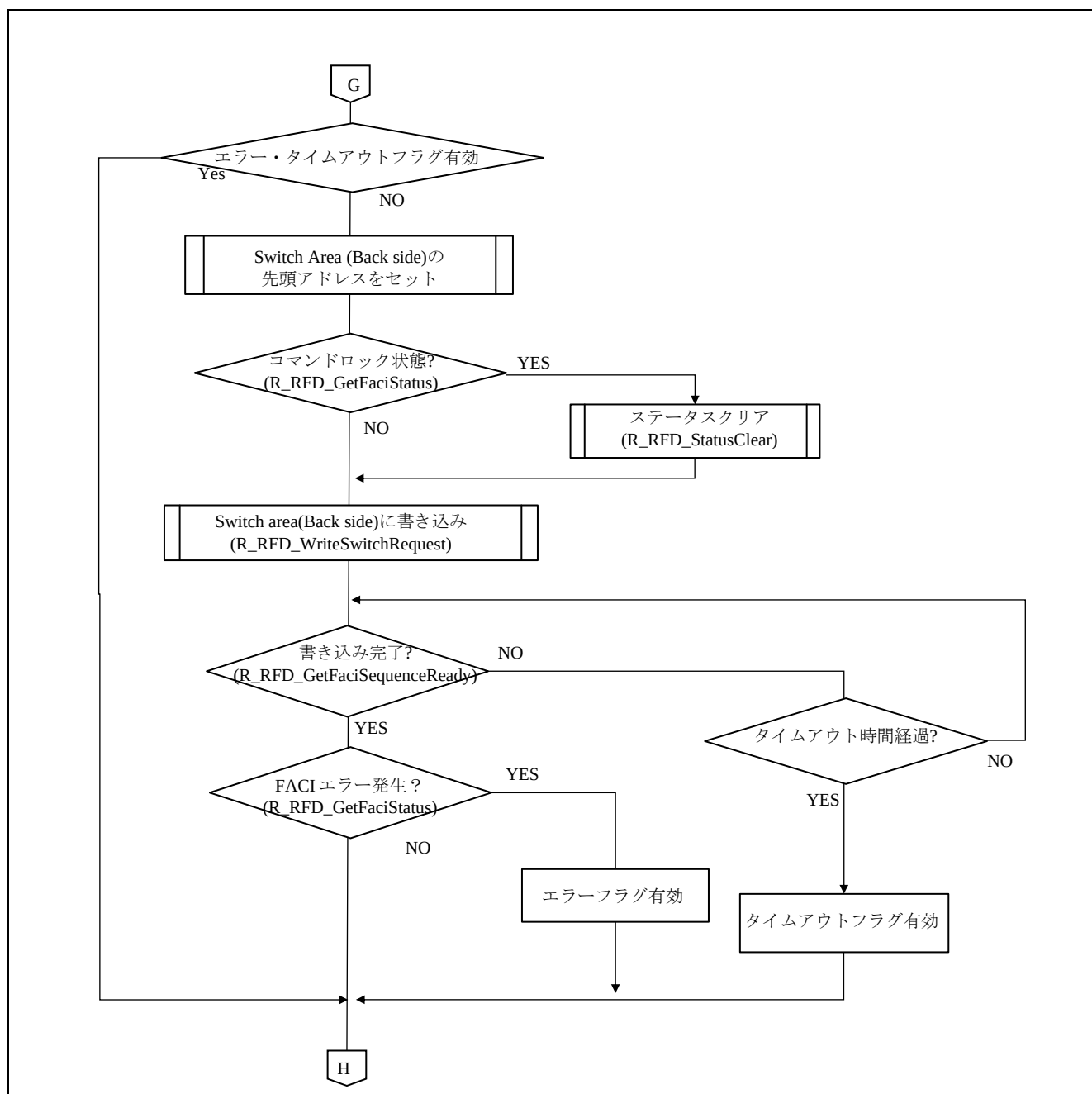


図 6-20 Switch area(Back side)書き込みのフローチャート

6.2.9 ⑨ TAG Area の更新

(1) TAG Area 消去

TAG Area のデータ消去を実行します。

図 6-21 に「Sample_PropertyAreaControl()関数」の TAG Area の消去フローチャートを示します。

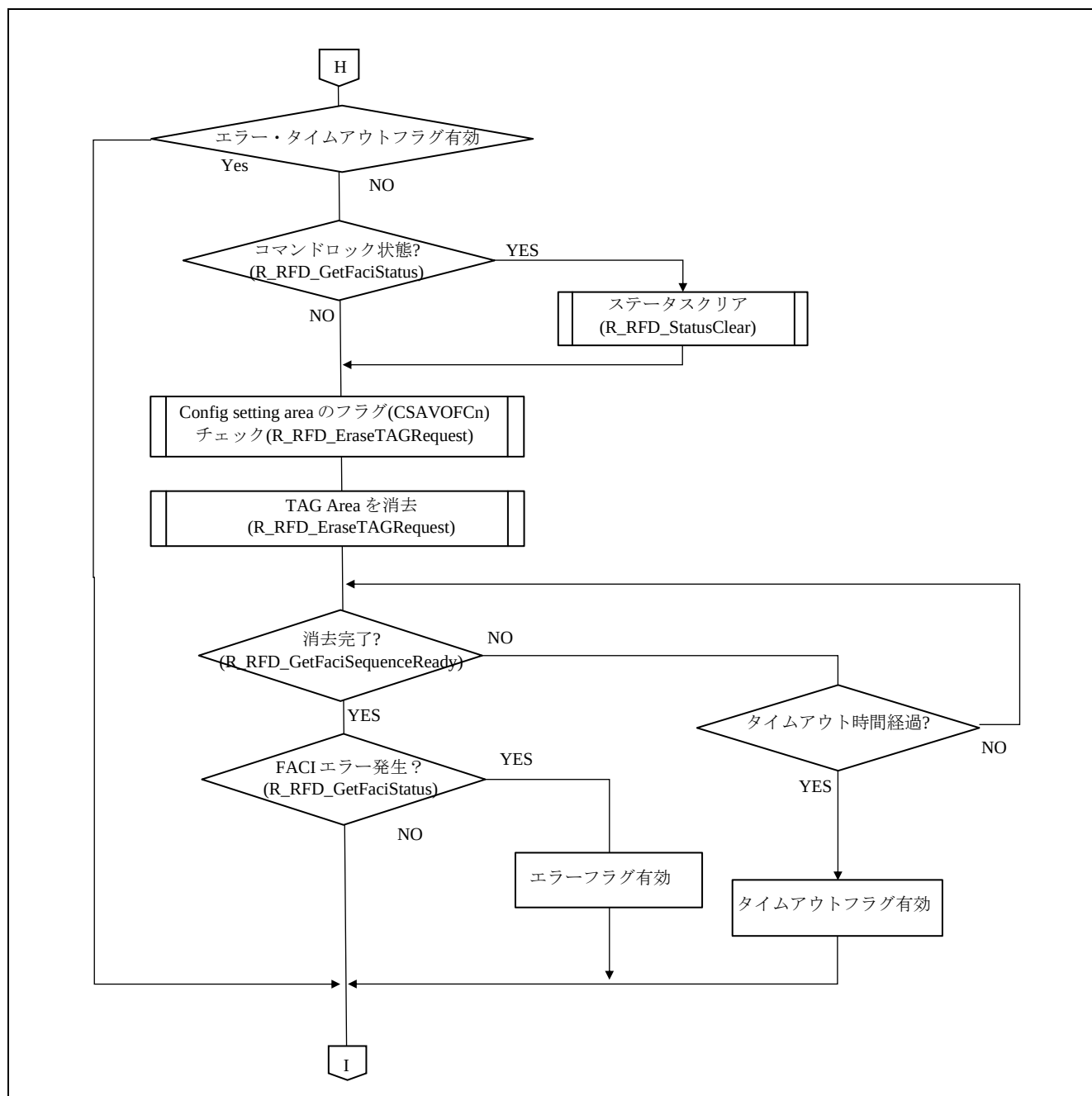


図 6-21 TAG Area の消去のフローチャート

(2) TAG Update

TAG Area の Valid Area Flag(VAF) を書き変えることにより、有効領域を Front side(valid) から Back side(invalid)に切り替えるコマンドを発行します。

図 6-22 に「Sample_PropertyAreaControl()関数」の TAG update のフローチャートを示します。

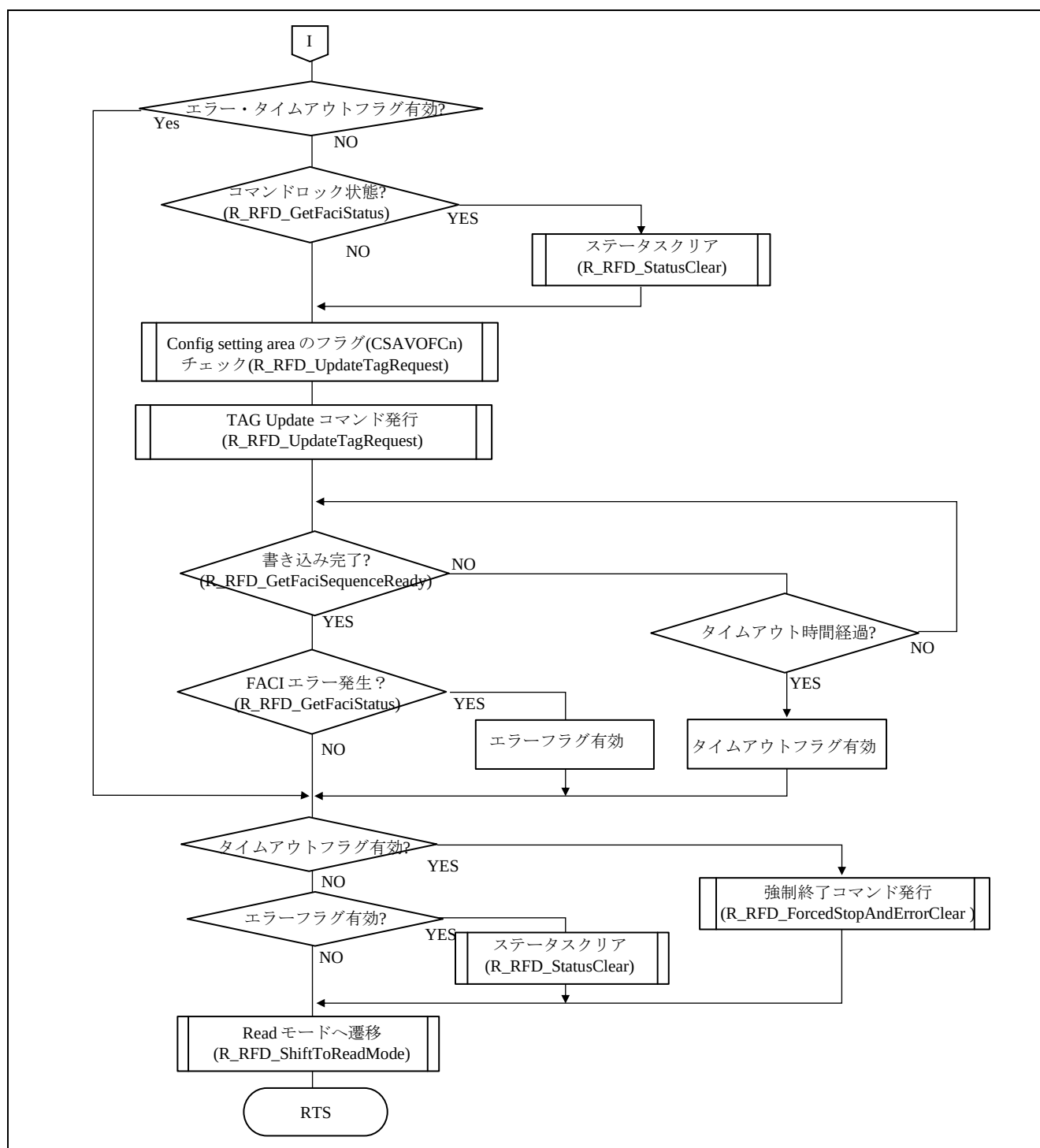


図 6-22 TAG update のフローチャート

6.3 メモリ割り付け

6.3.1 アドレス配置図

表 6-14 に Double map mode のアドレス配置図を示します。

表 6-14 Double map mode のアドレス配置図

フラッシュ領域	アドレス	セクション	アクセス
CodeFlash User Area0/1 (Bank A/B)	H0000_0000	.BankA.text	PE0
		.BankA.const	
		.FlashControl.text	
		.R_RFD_RODATA_EXTRA.const	
		.R_RFD_CODE_COMMON.text	
		.R_RFD_CODE_COMMON_RAM_NO_BGO.text	
		.R_RFD_CODE_USEROWN_COMMON.text	
		.R_RFD_CODE_DF.text	
		.R_RFD_CODE_CF.text	
		.R_RFD_CODE_CF_RAM_NO_BGO.text	
		.R_RFD_CODE_EXTRA.text	
		.R_RFD_RODATA_VERSION_DF.const	
		.R_RFD_RODATA_VERSION_CF.const	
		.R_RFD_RODATA_VERSION_COMMON.const	
	H0000_8000	RESET_PE1	PE1
		EHINITBL_PE1	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.text	
		.data	
CodeFlash User Area2 (Bank C) ※U2A8,U2A6は非搭載	H0040_0000	RESET_PE2	PE2
		EHINITBL_PE2	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.text	
		.data	
	H0042_0000	RESET_PE3	PE3
		EHINITBL_PE3	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.text	
		.data	
CodeFlash User Boot Area0/1 (Bank A/B)	H0800_0000	RESET_PE0	PE0
		EHINITBL_PE0	
		.const	
		.INIT_DSEC.const	
		.INIT_BSEC.const	
		.text.cmn	
		.data	
		.text	
	H0800_4000	.text	PE0～PE3
LRAM (PE3) ※U2A8,U2A6は非搭載	HFD60_0000	.data.R	PE3
		.bss	
		.stack.bss	
LRAM (PE2) ※U2A8,U2A6は非搭載	HFD80_0000	.data.R	PE2
		.bss	
		.stack.bss	
LRAM (PE1)	HFD A0_0000	.data.R	PE1
		.bss	
		.stack.bss	
LRAM (PE0)	HFDC0_0000	.bss	PE0
		.BankA.bss	
		.data.R	
		.stack.bss	
		.R_RFD_BSS.bss	

7. ソフトウェアの更新中に処理が中断した時の復帰動作

Code Flash の書き換え中及び Hardware Property Area の書き換え中に、処理が中断(瞬断、リセット等)した場合は、消去からやり直すことが必須となります。また、ソフトウェアの更新中に、処理が中断した場合の意図しない中断の検出及び、フラッシュメモリの有効領域切り替え後の妥当性(意図した Bank で起動していること)の確認のため、Data Flash 上のユーザーフラグによる進捗の管理を推奨します。

Code Flash の書き換え中に瞬断した場合は、Blank Checking では瞬断した際の消去状態/書き込み状態はチェックできないため、Data Flash 上にユーザーフラグを配置し進捗管理を推奨します。瞬断した場合に、Code Flash の書き換えがどこまで進んだかのユーザーフラグにより、書き換えを消去からやり直すことを推奨します。

Hardware Property Area の書き換え中に瞬断した場合にも、Switch Area の Valid Area Flag(CVA,SVA, BVAn)や TAG Area の Valid Area Flag(VAPC)には頼れないケースがあり、この場合にも Data Flash 上でユーザーフラグを配置し進捗管理を推奨します。

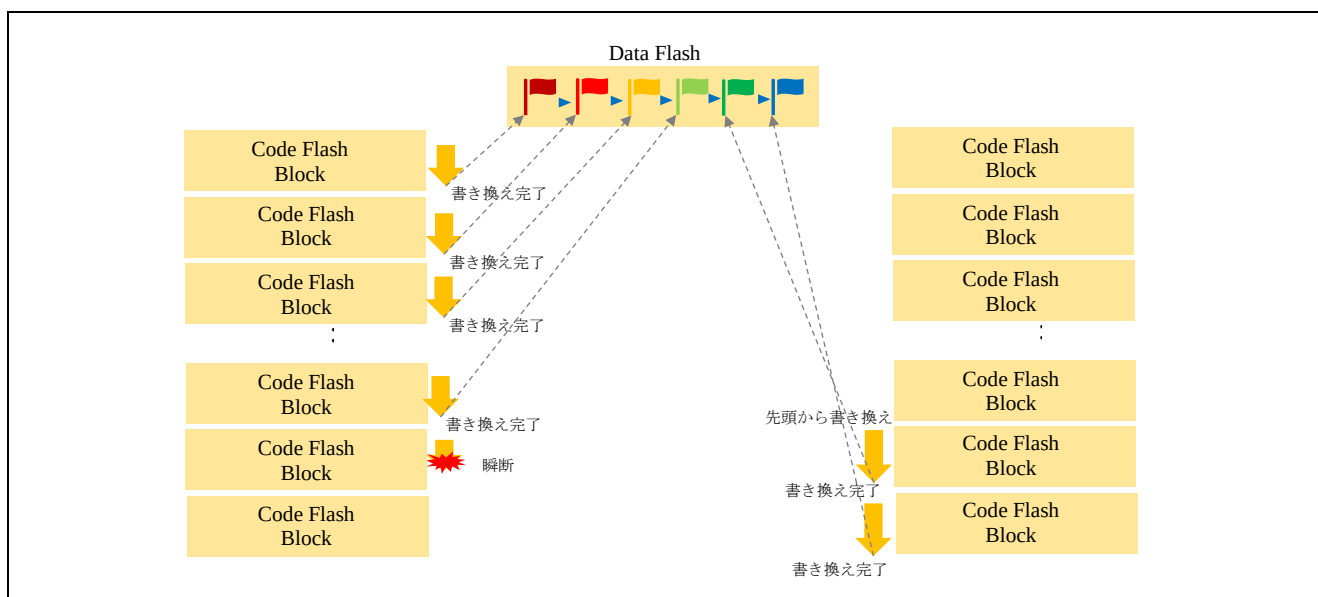


図 7-1 ユーザーフラグイメージ

改訂履歴

リビジョン	日付	改訂履歴	
		ページ	概要
1.00	2021.08.05	全頁	初版発行
1.10	2022.04.01	全頁	対象とする製品に RH850/U2A6 を追加

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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しします。
13. 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。
14. 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. 5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

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/

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。