RX ファミリ
フラッシュ ROM コレクション

要旨

大容量フラッシュメモリや、ユーザシステムでの低速シリアル書き換えにおいては、フラッシュメモリの全書き換えに時間を要します。本アプリケーションノートでは、必要最小限の領域を書き換えることにより、ユーザプログラムにパッチを当てるフラッシュ ROM コレクション動作の方法を説明します。

本アプリケーションノートでは、特に記載がない限り RX113 を例に説明しています。

動作確認デバイス

・RX113 グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様に合わせて変更し、十分評価してください。

目次

1. 仕様

本アプリケーションノートでは、ユーザが作成したパッチプログラムやテーブルデータを、図 1.1 に示すパッチエリア、図 1.2 に示すテーブルデータエリアにあらかじめ書き込んでおくことで、ユーザプログラムを更新（以下、ROM コレクション）する方法を説明します。

本サンプルコードはユーザプログラム実行前に動作させます。ユーザプログラム実行中、パッチを当てる箇所に到達した場合にパッチプログラムを実行し、パッチプログラム実行後にユーザプロプログラムへ戻ります。本アプリケーションノートの動作例を図 1.3 に示します。

ユーザによるパッチプログラムやテーブルデータエリアを書き込みの途中で、瞬時電圧低下などで書き込みに失敗した場合でも、更新前のプログラムを実行することができます。また本サンプルコードは、パッチプログラムを適用している間に瞬時電圧低下などで書き込みが失敗した場合、再起動時に適用をリトライして更新プログラムを適用します。

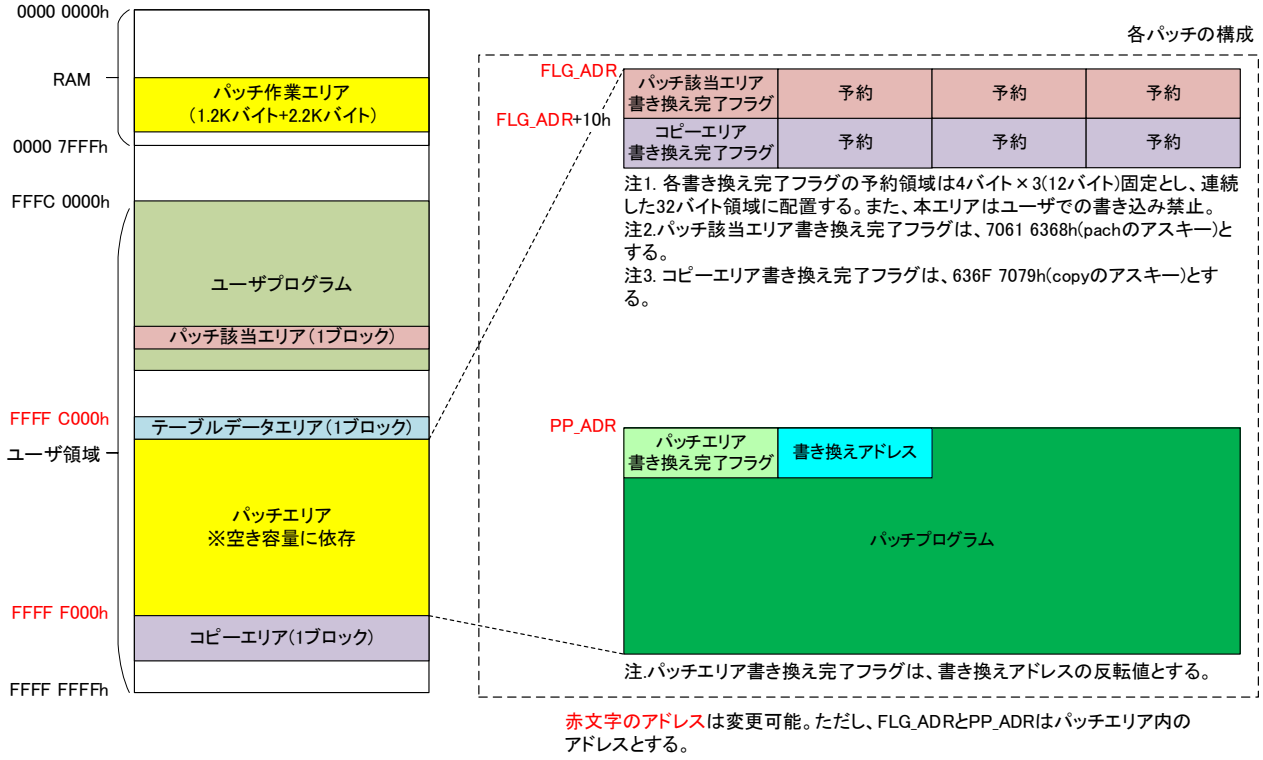


図 1.1　パッチエリア

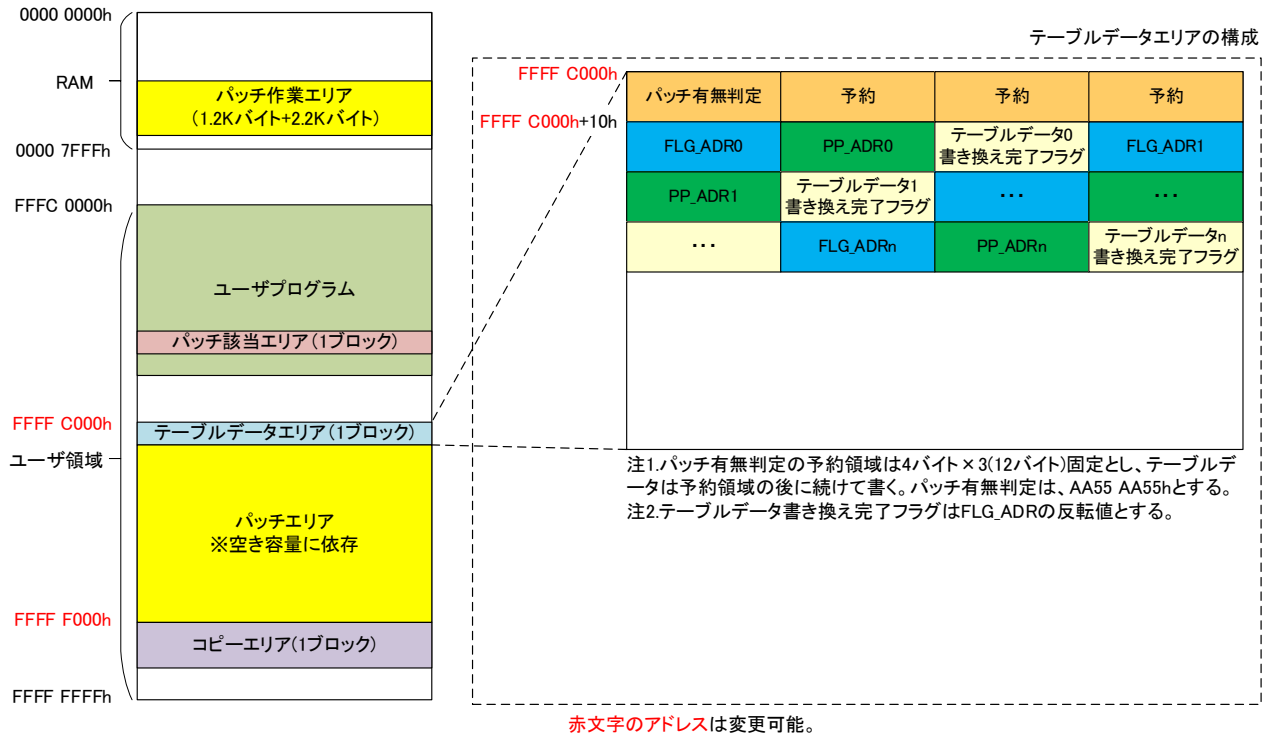


図 1.2　テーブルデータエリア

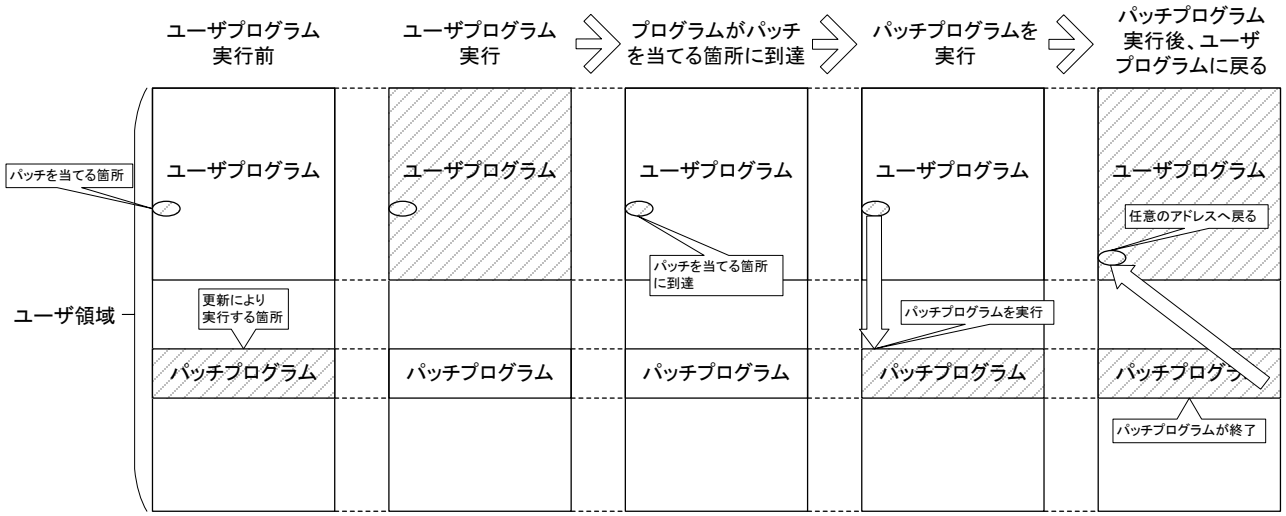


図 1.3　本アプリケーションノートの動作例

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1　動作確認条件

項目	内容
使用マイコン	R5F51138AFP (RX113 グループ)
動作周波数	• メインクロック：16MHz • サブクロック停止 • PLL：32MHz (メインクロック：4 分周 8 逓倍) • HOCO 停止 • システムクロック(ICLK)：32MHz (PLL 1 分周) • FlashIF クロック(FCLK)：32MHz (PLL 1 分周)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e2stdio Version 7.7.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V3.02.00
	コンパイルオプション 統合開発環境のデフォルト設定を使用しています。
iodefine.h のバージョン	Ver1.1
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Starter Kit for RX113 (製品名：R0K505113C010BR)

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノーを以下に示します。併せて参照してください。

- RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)
- RX Family Flash Module Using Firmware Integration Technology (R01AN2184)

上記アプリケーションノートを、本アプリケーションノートのサンプルコードで使用しています。最新版がある場合、最新版に差し替えて使用してください。最新版はルネサスエレクトロニクスホームページで確認および入手してください。

4.　ソフトウェア説明

4.1　動作概要

ROM コレクションする場合、ユーザの仕様に合わせてテーブルデータエリア、パッチエリアを作成してください。なお、本サンプルコードではパッチ有無判定、パッチ該当エリア書き換え完了フラグ、コピーエリア書き換え完了フラグの書き込み上限回数を 4 回としています。

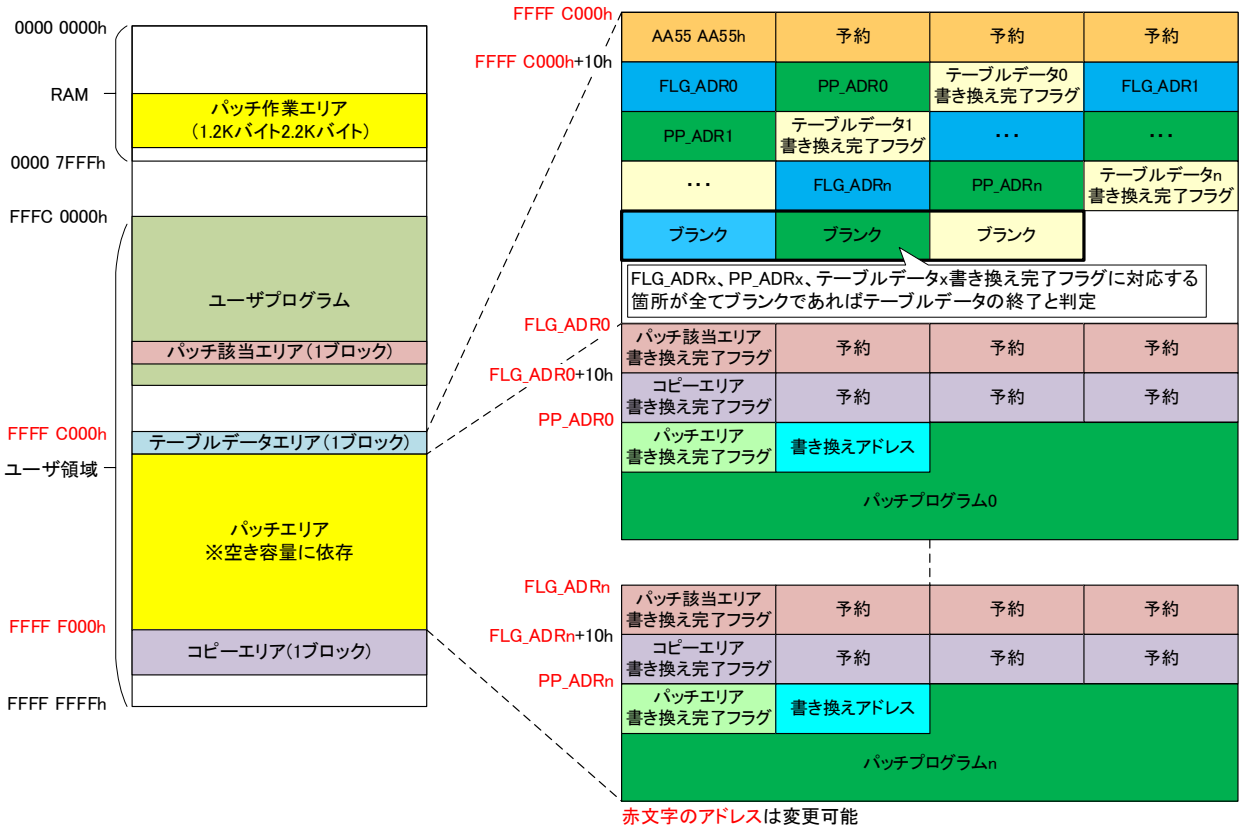


図 4.1　ROM コレクションする場合のテーブルデータエリア

本サンプルコードはユーザプログラム実行前に動作させてください。お客様の作成したパッチエリア、テーブルデータエリアに対応した処理を実行後、ユーザプログラムを実行します。本サンプルコード使用時のフローを図 4.2 に示します。

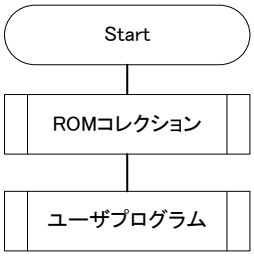


図 4.2　本サンプルコード使用時のフロー

4.1.1　テーブルデータエリア、パッチエリアの確認

本サンプルコードは ROM コレクションする場合、テーブルデータエリア、パッチエリアを図 4.3 に示す手順で確認します。

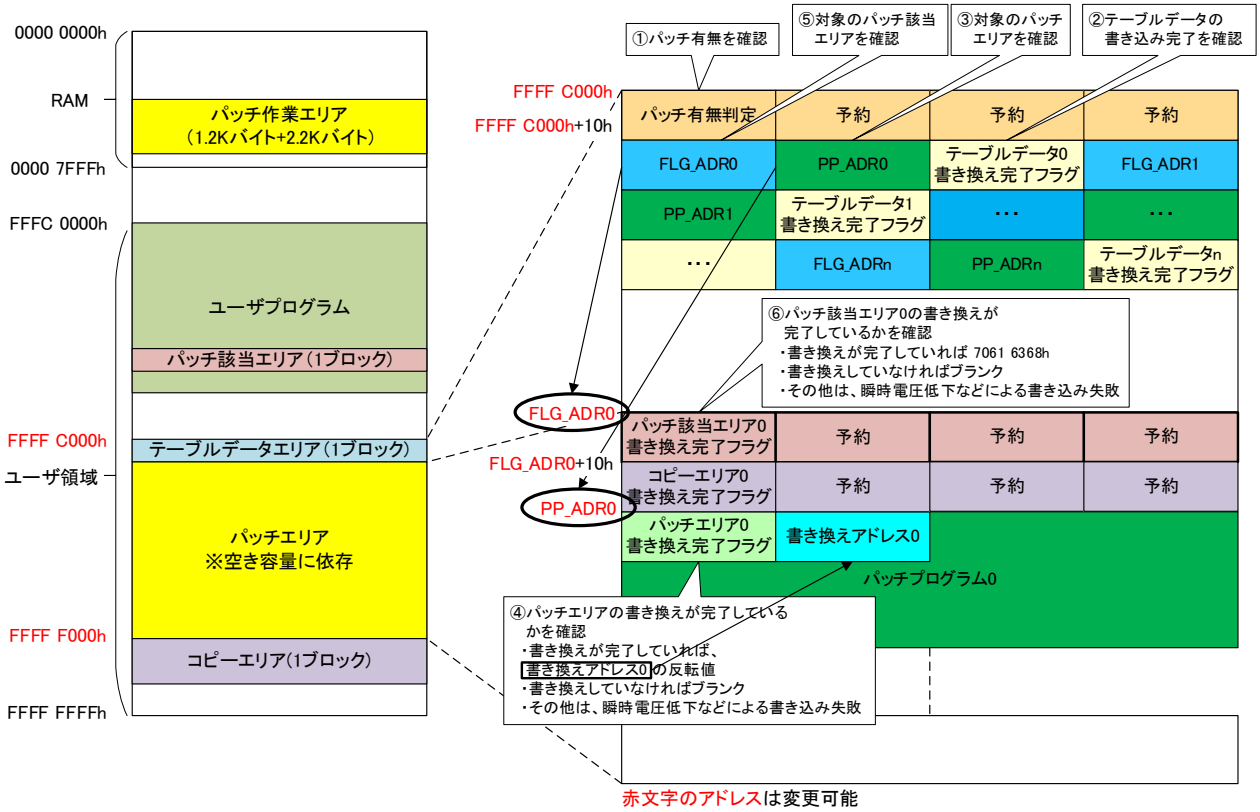


図 4.3　テーブルデータエリア、パッチエリアの確認

4.1.2　ROM コレクションのフロー

ROM コレクションのフローを図 4.4 に示します。

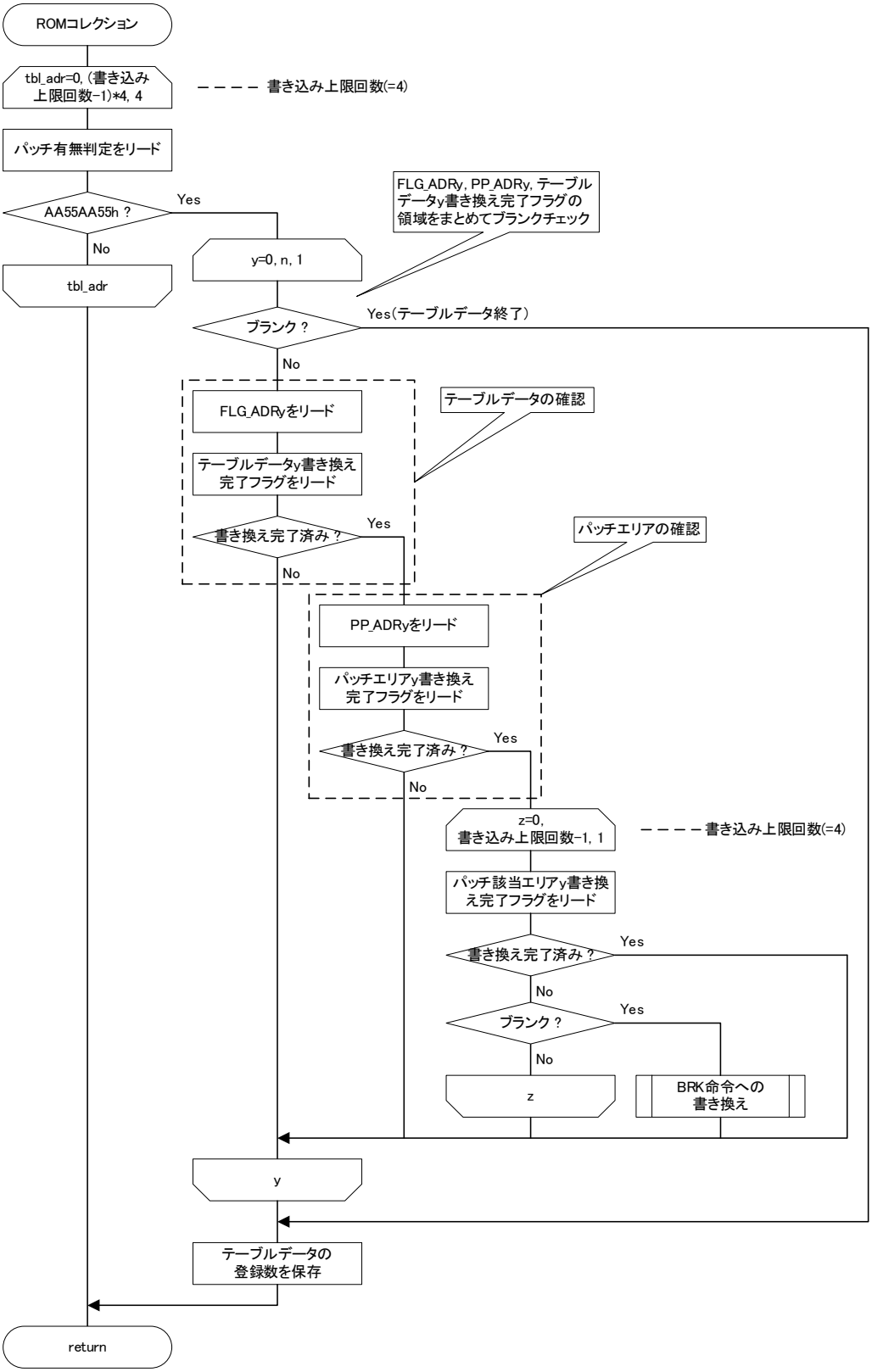


図 4.4　ROM コレクションのフロー

4.1.3　BRK 命令への書き換えフロー

本サンプルコードではパッチプログラムを実行する要因に、ROM コレクションで 00h に書き換えたアドレスを実行したときに発生する BRK 例外処理を使用しています。BRK 命令への書き換えフローを図 4.5 に示します。

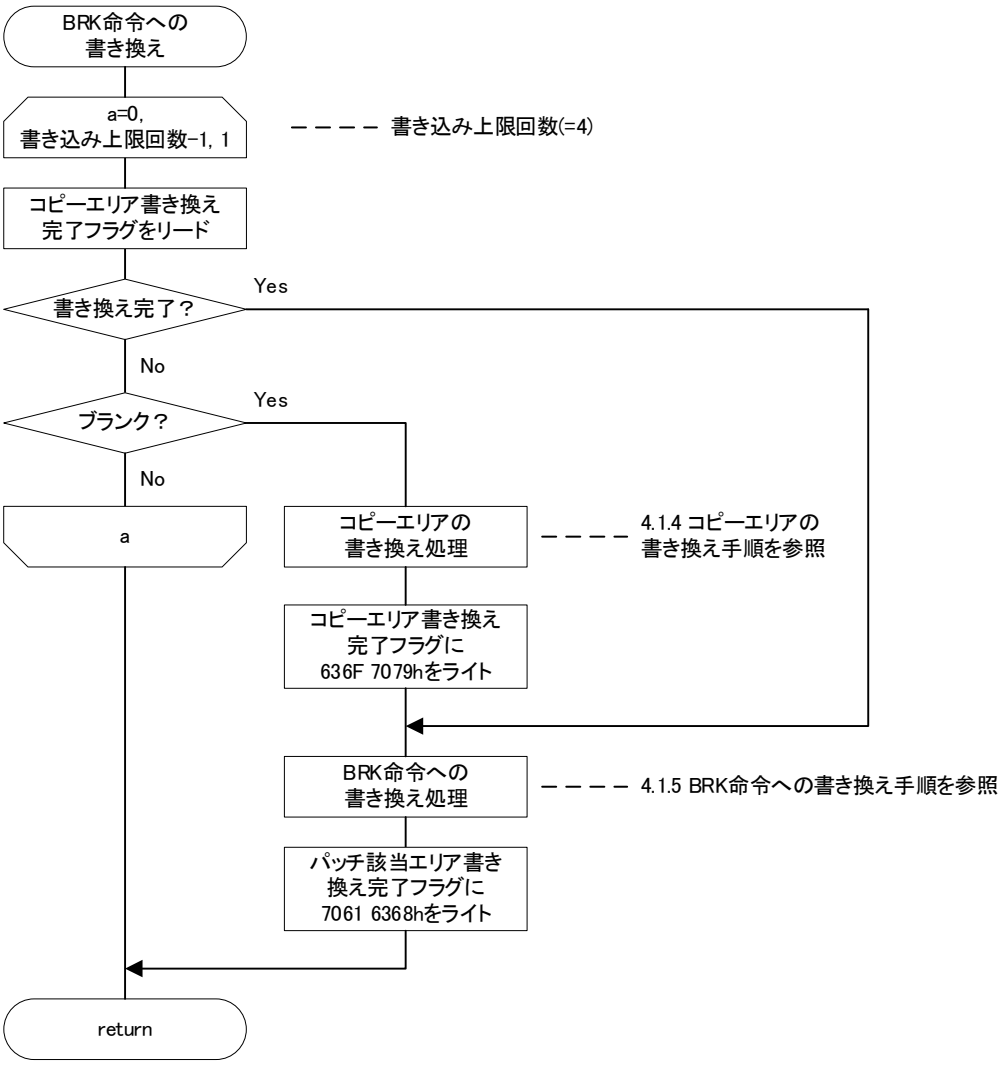


図 4.5　BRK 命令への書き換えフロー

4.1.4　コピーエリアの書き換え手順

本サンプルコードでは、パッチ該当エリアの書き換え中に瞬時電圧低下などで書き込みが失敗する可能性を考慮し、あらかじめ設けたコピーエリアに書き換え前のパッチ該当エリアを退避させます。コピーエリアの書き換え手順を図 4.6 に示します。

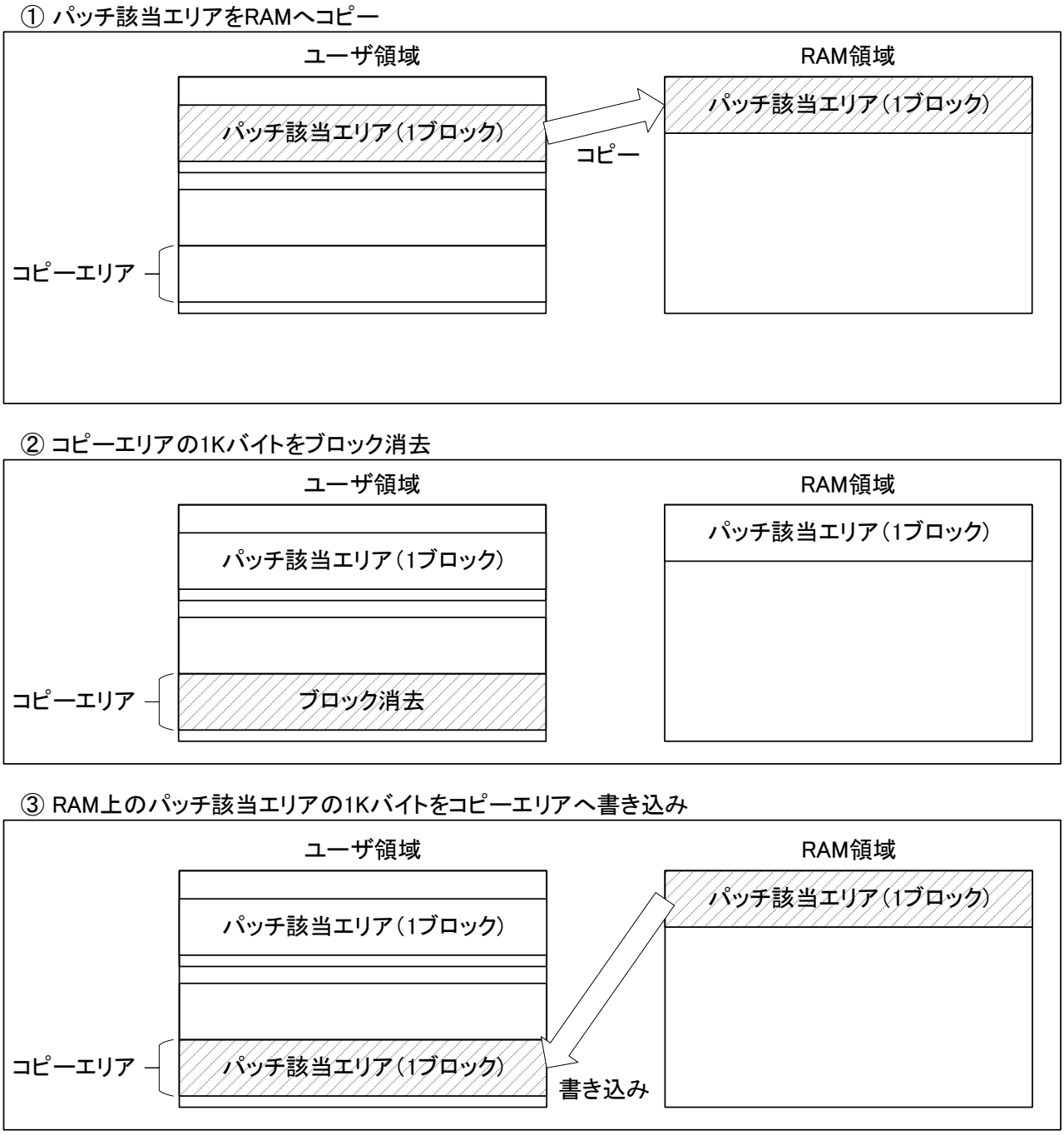


図 4.6　コピーエリアの書き換え手順

4.1.5　BRK 命令への書き換え手順

BRK 命令への書き換え手順を図 4.7 に示します。

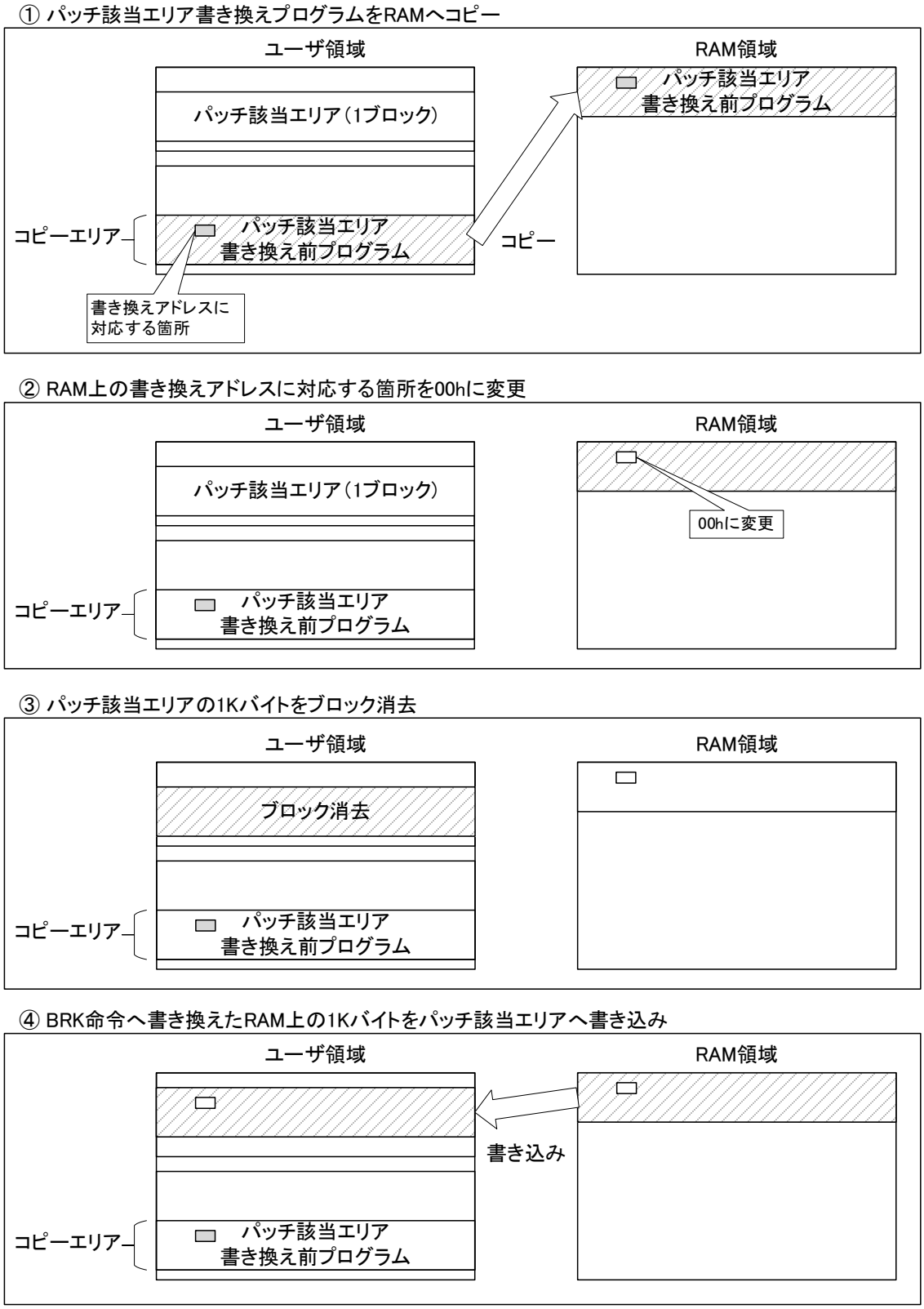


図 4.7　BRK 命令への書き換え手順

4.1.6　予約領域を確保しているフラグの書き込み手順

予約領域を確保しているパッチ該当エリア書き換え完了フラグ、コピーエリア書き換え完了フラグ、パッチ有無判定の書き込みは、対象となる領域がブランクであれば書き込み、ブランクでなければ瞬時電圧低下などによる書き込み失敗と判定し、アドレスを 4h 進めた領域を対象とし、ブランクかどうかを再度確認します。

パッチ該当エリア書き換え完了フラグを例に書き込み手順例を図 4.8 に示します。

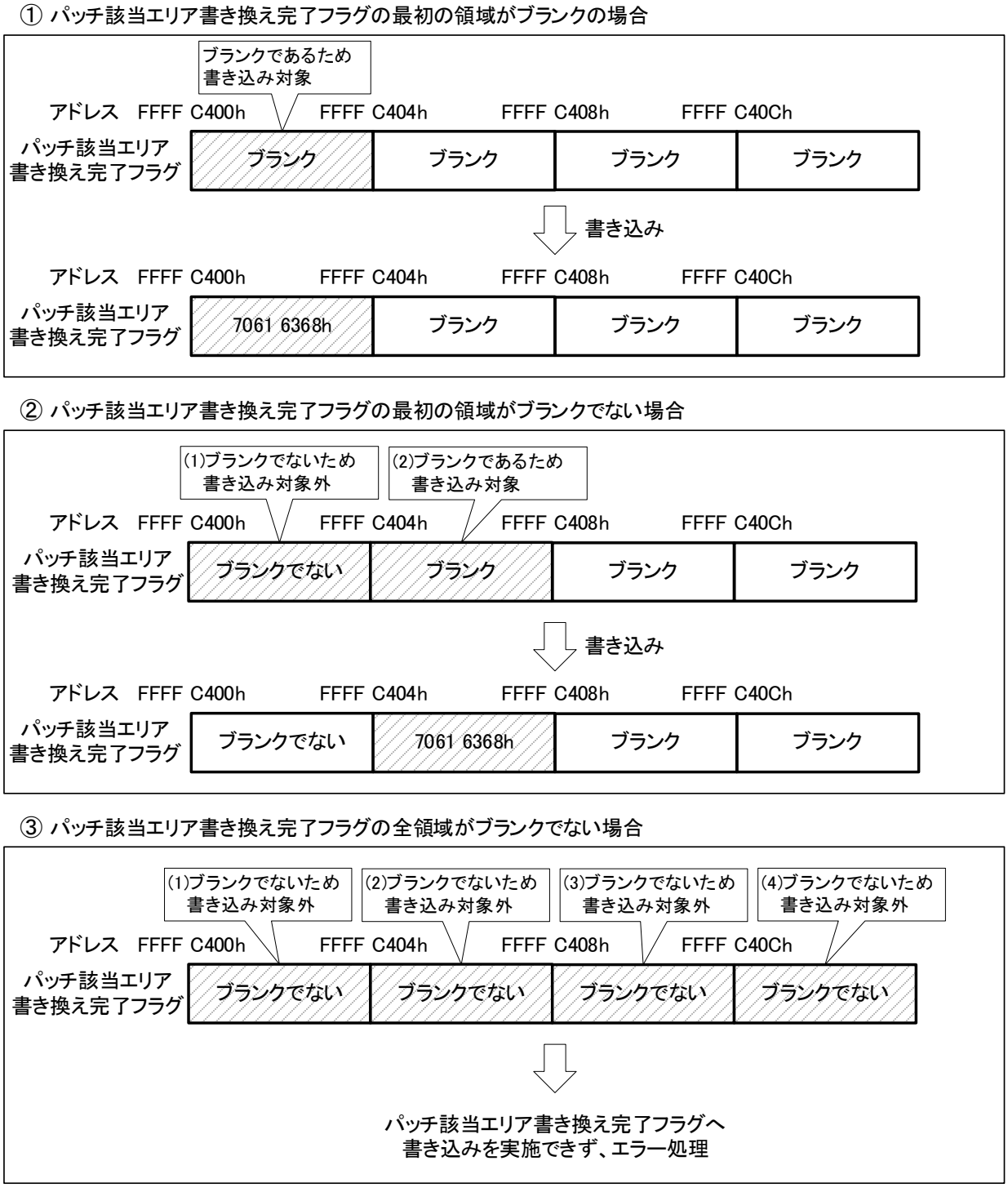
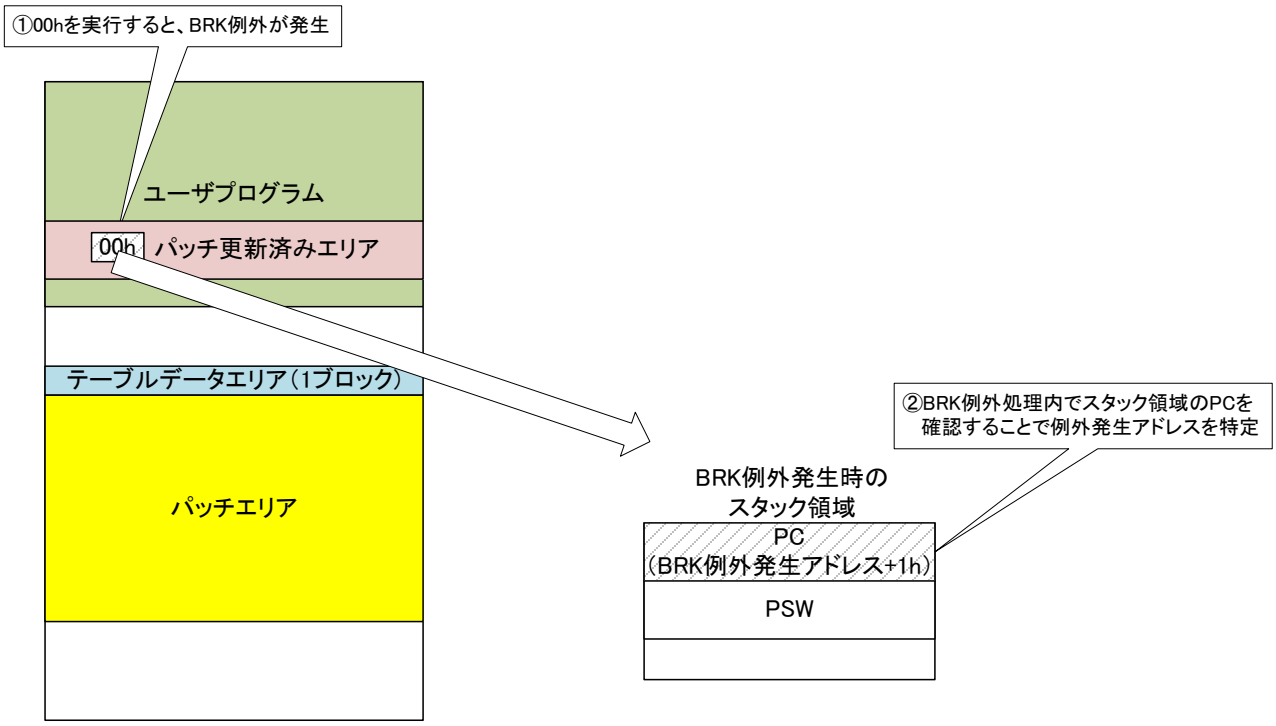


図 4.8　パッチ該当エリア書き換え完了フラグを例に書き込み手順

4.1.7　BRK 例外（無条件トラップ）発生アドレスの特定

本サンプルコードではパッチプログラムを実行する要因に、ROM コレクションで 00h に書き換えたアドレスを実行したときに発生する BRK 例外を使用しています。

ユーザプログラム実行中に書き換えアドレスの対象箇所である 00h を書き込んだアドレスを実行すると、BRK 例外が発生します。BRK 例外処理内でスタック領域の PC（プログラムカウンタ）を確認することで BRK 例外が発生したアドレスを特定します。BRK 例外発生時の動作を図 4.9 に示します。

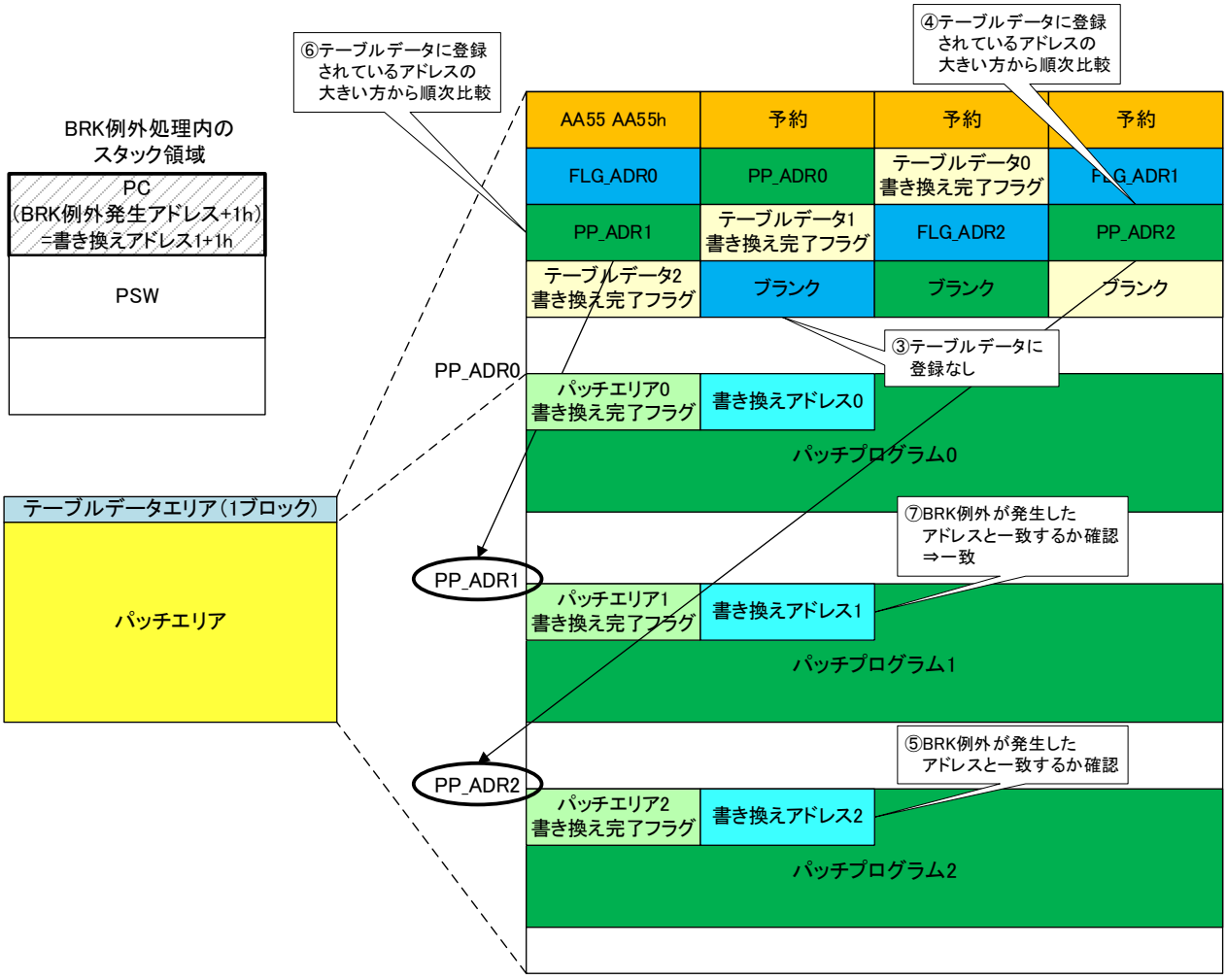


※図 4.10 に続く

図 4.9　BRK 例外発生時の動作

4.1.8　BRK 例外が発生したアドレスから実行するパッチ処理を判定

BRK 例外処理内では、BRK 例外が発生したアドレスとテーブルデータエリア経由で各パッチエリアの書き換えアドレスを比較し、実行するパッチ処理を判定します。なお、テーブルデータへの登録はアドレスの昇順で登録していくことを想定し、アドレスの大きい方から順次比較していきます。この方法により、同一アドレスに複数のパッチが当たっている場合にも最新のパッチプログラムを適用することができます。BRK 例外が発生したアドレスから実行するパッチ処理を判定する動作を図 4.10 に示します。



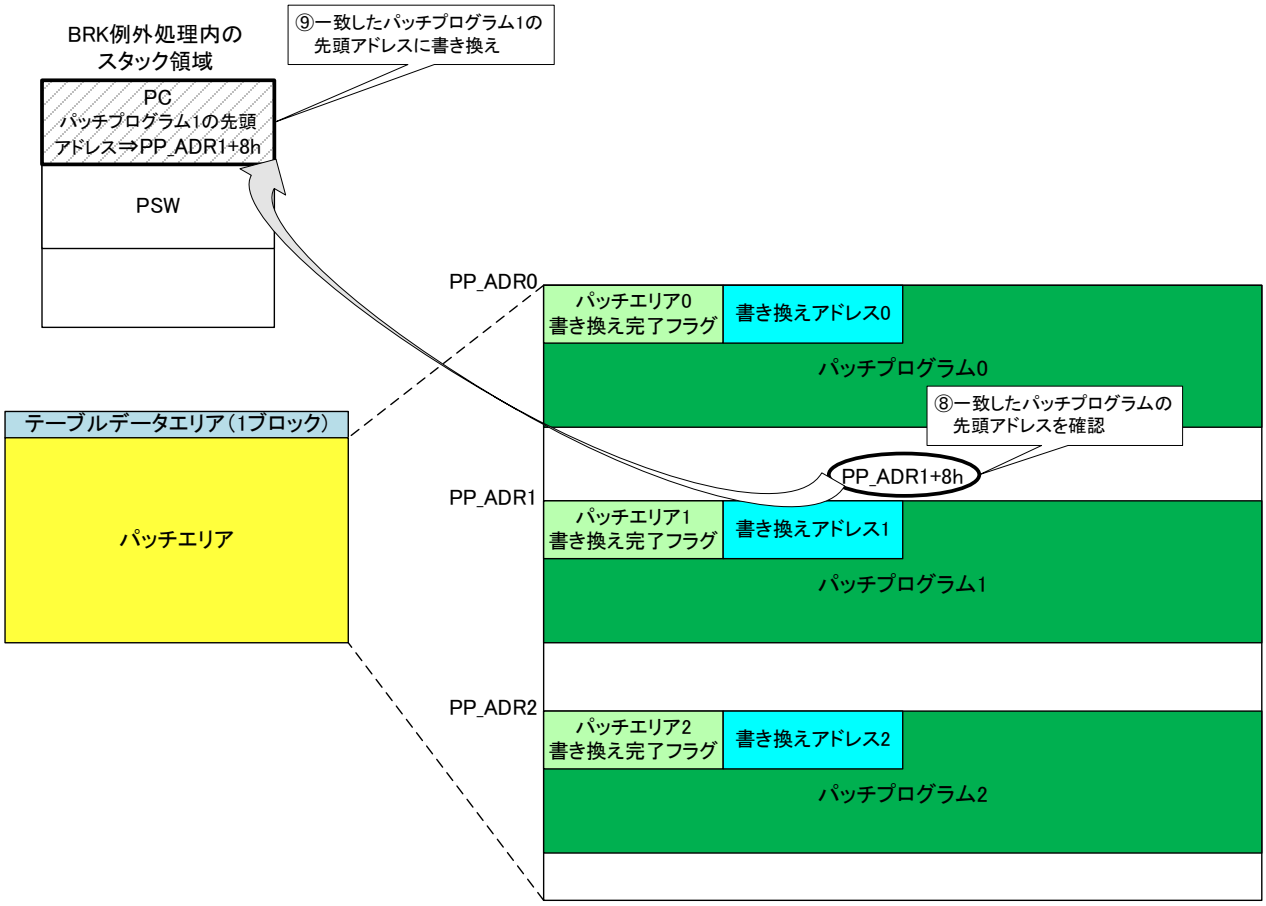
注.書き換えアドレス 0 と書き換えアドレス 1 が同一アドレスであった場合でも、
パッチプログラム 0（古いパッチ）は上記の動作により適用されません

※図 4.11 に続く

図 4.10　BRK 例外が発生したアドレスから実行するパッチ処理を判定する動作

4.1.9　BRK 例外処理からの復帰設定

実行するパッチ処理を判定後、BRK 例外処理内でスタック領域の PC（プログラムカウンタ）に実行するパッチプログラムの先頭アドレスを設定することで、BRK 例外処理後に実行するパッチプログラムへ復帰します。BRK 例外処理からの復帰設定を図 4.11 に示します。



※図 4.12 に続く

図 4.11　BRK 例外処理からの復帰設定

4.1.10 パッチプログラムの実行とユーザプログラムへの復帰

BRK 例外処理後、実行するパッチプログラムへ復帰し、パッチプログラムを実行します。パッチプログラムを実行後、ユーザがあらかじめ記述した命令によりユーザプログラムへ戻ります。パッチプログラムからユーザプログラムへ戻る動作を図 4.12 に示します。

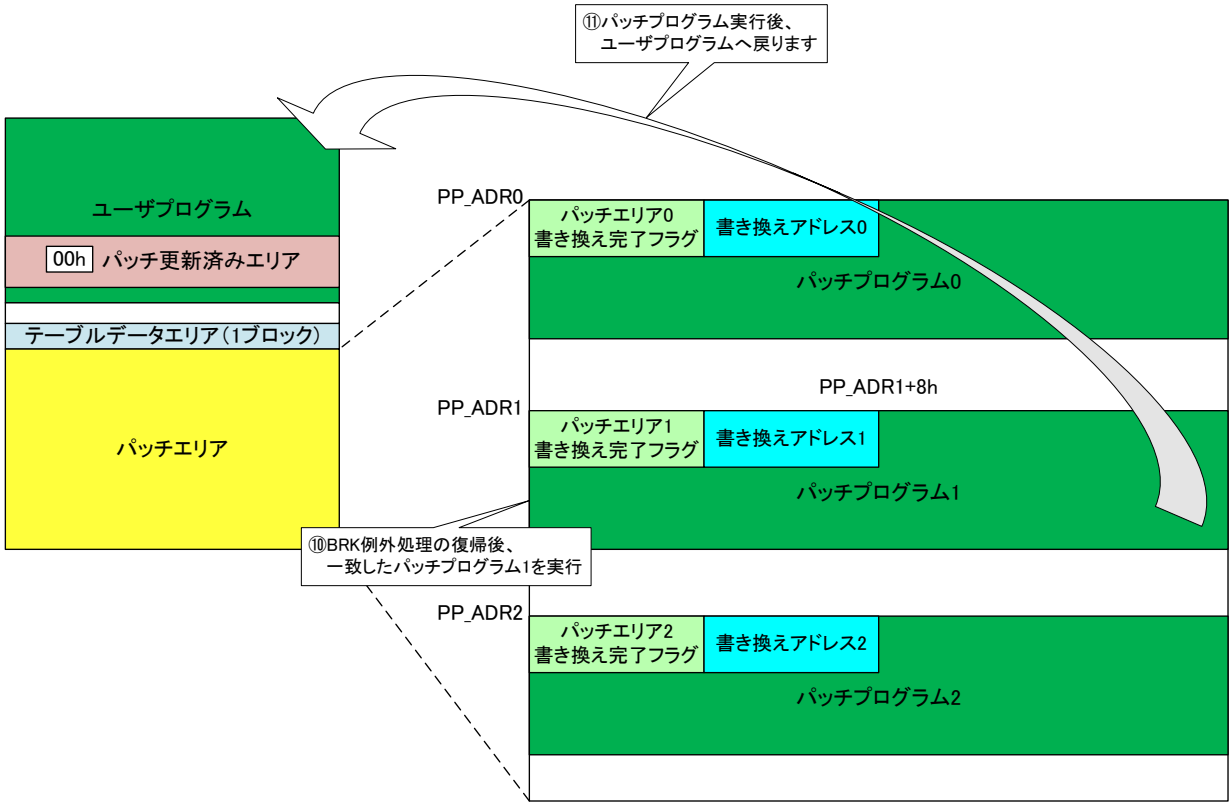


図 4.12　パッチプログラムの実行動作

4.2　必要メモリサイズ

表 4.1 に必要メモリサイズを示します。

表 4.1　必要メモリサイズ

使用メモリ	サイズ	備考
ROM	1028 バイト	r_rom_correct.c モジュール内
RAM	20 バイト	r_rom_correct.c モジュール内
パッチ作業エリア	1300 バイト	r_rom_correct.c モジュール内 サイズはコピーするパッチ該当エリアの容量に依存
	2531 バイト	r_flash_rx モジュール内で使用する容量

必要メモリサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。
r_bsp、r_flash_rx のサイズについては各アプリケーションノートを参照ください。

4.3　ファイル・フォルダ構成

本サンプルコードのソースファイルとフォルダ構成を図 4.13 に示します。なお、統合開発環境で自動生成されるファイルは除きます。

```
＋HardwareDebug                                    [実行ファイルなど]
＋src
    ＋－ smc_gen
    |     ＋－－ r_bsp                              [Renesas Board Support Package]
    |     |       ＋－－ board                      BSP 各RSK用設定
    |     |             ＋－－ rskrx113             RX113用RSK
    |     |             ＋－－ user                 ユーザ設定用
    |     |       ＋－－ doc                        BSP アプリケーションノート
    |     |             ＋－－ en                   英語版
    |     |             ＋－－ jp                   日本語版
    |     |       ＋－－ mcu                        BSP 各MCU情報
    |     |             ＋－－ all                  MCU共通
    |     |             ＋－－ rx113                RX113用
    |     |                   ＋－－ register_access  RX113 レジスタアクセス
    |     ＋－－ r_config                           [API設定ファイル]
    |     ＋－－ r_flash_rx                         [Simple Flash API]
    |             ＋－－ doc                        Simple Flash API アプリケーションノート
    |             ＋－－ ref                        FlashAPI ドライバ設定ファイル
    |             ＋－－ src                        FlashAPI ドライバ
    |                   ＋－－ flash_type_1        Flashタイプ1用
    |                   ＋－－ flash_type_2        Flashタイプ2用
    |                   ＋－－ flash_type_3        Flashタイプ3用
    |                   ＋－－ targets             各種MCU ROM情報
    |                         ＋－－ rx113         RX113用Flash定義
    ＋－－－ r_rom_correct                          [ROMコレクション]
```

図 4.13　ROM コレクションのフォルダ構成

なお、本サンプルコードは以下のパッケージを使用しています。

- r_bsp (Renesas board support package)
- r_flash_rx (RX Family simple flash module)

4.3.1 src フォルダ

ROM コレクションを使用するサンプルソースファイル、およびヘッダファイルが格納されているフォルダです。

表 4.2　使用サンプル

ファイル名	概要	備考
main.c	main 用ソースファイル	
main.h	main 用ヘッダファイル	

4.3.2 src¥r_bsp フォルダ

Renesas Board support package module のソースファイル、およびヘッダファイルが格納されているフォルダです。詳細は、RX ファミリ ボードサポートパッケージモジュール アプリケーションノートを参照してください。

4.3.3 src¥r_config フォルダ

対象 MCU の設定ファイルが格納されているフォルダのファイル一覧を表 4.3 に示します。

表 4.3　ヘッダファイル

ファイル名	概要	備考
r_bsp_config.h	BSP の設定ヘッダファイル	RSK RX113 用
r_flash_rx_config.h	Flash 書き込み設定ファイル	

4.3.4 src¥r_flash_rx フォルダ

フラッシュ FIT モジュールのソースファイル、およびヘッダファイルが格納されているフォルダです。詳細は Flash Module Firmware Integration Technology のアプリケーションノートを参照してください。

4.3.5 src¥r_rom_correct フォルダ

ROM コレクションのソースファイル、およびヘッダファイルが格納されているフォルダです。

表 4.4　ROM コレクション

ファイル名	概要	備考
r_rom_correct.c	ROM コレクション用ソースファイル	
r_rom_correct.h	ROM コレクション用ヘッダファイル	

4.4　オプション設定メモリ

表 4.5 に、サンプルコードで設定しているオプション設定メモリの状態を示します。必要に応じて、ユーザシステムに最適な値を設定してください。

表 4.5　サンプルコードで設定しているオプション設定メモリ

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh-FFFF FF8Ch	FFFF FFFFh	• リセット後、IWDT は停止状態 • タイムアウト期間　2048 サイクル • IWDT クロック分周比　128 分周 • ウィンドウ開始位置/終了位置　指定なし • リセットを許可 • スリープモードのカウント停止有効
OFS1	FFFF FF88h-FFFF FF8Bh	FFFF FFFFh	• 起動時、電圧監視リセット無効 • リセット後、HOCO 発振が無効
MDE	FFFF FF80h-FFFF FF83h	FFFF FFFFh	• リトルエンディアン

4.5　定数一覧

表 4.6 に、サンプルコードで使用する定数を示します。

表 4.6　サンプルコードで使用する定数

定数名	設定値	内容
APPLY_PATCH	AA55 AA55h	パッチ有無判定のパッチありを示すデータ
PATCH_TARGET_REWRITE	7061 6368h	パッチ該当エリア書き換え完了フラグの書き換え完了を示すデータ
COPY_AREA_REWRITE	636F 7079h	コピーエリア書き換え完了フラグの書き換え完了を示すデータ
LMT_WRITE_TIMES	4	パッチ該当エリア書き換え完了フラグ、コピーエリア書き換え完了フラグ、パッチ有無判定の書き込み上限値
REWRITE_FLG_AREA_SIZE	LMT_WRITE_TIMES * 4	パッチ該当エリア書き換え完了フラグ、コピーエリア書き換え完了フラグ、パッチ有無判定の領域サイズ
BRK_CODE	00h	BRK 命令を示すコード
FLASH_RETRY_TIMES	3	フラッシュ関連エラーが発生した場合のリトライ数(注 1)
FLASH_BLOCK_TOP_MASK	FFFF FC00h	フラッシュブロックの先頭アドレスを取得するマスク
TABLE_DATA_AREA_TOP_ADR	FFFF C000h	テーブルデータエリア先頭アドレス
APPLY_PATCH_ADR	TABLE_DATA_AREA_TOP_ADR	パッチ有無判定の先頭アドレス
TABLE_DATA_TOP_ADR	TABLE_DATA_AREA_TOP_ADR + REWRITE_FLG_AREA_SIZE	テーブルデータ先頭アドレス (パッチ有無判定を除いた先頭)
PATCH_AREA_TOP_ADR	TABLE_DATA_AREA_TOP_ADR + FLASH_CF_BLOCK_SIZE	パッチエリア先頭アドレス
COPY_AREA_TOP_ADR	FFFF F000h	コピーエリア先頭アドレス
TABLE_DATA_NUM_MAX	(COPY_AREA_TOP_ADR-TABLE_DATA_TOP_ADR)/(sizeof(table_data_t)	テーブルデータの最大数

注 1.本サンプルコードでフラッシュに関するエラーが発生した場合、全て本設定に応じたリトライ数までエラーが続いた場合にエラーを通知します。

4.6　構造体一覧

図 4.14 にサンプルコードで使用する構造体を示します。

```
typedef struct
{
        uint32_t flg_adr;           /* FLG_ADR */
        uint32_t pp_adr;            /* PP_ADR */
        unit32_t table_rewrite_flg; /* テーブルデータ書き換え完了フラグ */
} table_data_t;

typedef struct
{
        uint32_t patch_rewrite_flg;/* パッチエリア書き換え完了フラグ */
        uint32_t rewrite_adr;       /* 書き換えアドレス */
        uint8_t * ppatch_prg;       /* パッチプログラムの先頭アドレスポインタ */
} patch_data_t;
```

図 4.14　サンプルコードで使用する構造体

4.7　enum 一覧

図 4.15 にサンプルコードで使用する enum を示します。本サンプルコードで使用している RX Family simple flash module の構造体である flash_err_t を包含していますので、合わせて参照ください。

```
typedef enum
{
        ROM_CORRECT_SUCCESS = 0,
        ROM_CORRECT_ERR_BUSY,
        ROM_CORRECT_ERR_ACCESSW,
        ROM_CORRECT_ERR_FAILURE,
        ROM_CORRECT_ERR_CMD_LOCKED,
        ROM_CORRECT_ERR_LOCKBIT_SET,
        ROM_CORRECT_FREQUENCY,
        ROM_CORRECT_ALIGNED,
        ROM_CORRECT_BOUNDARY,
        ROM_CORRECT_OVERFLOW,
        ROM_CORRECT_ERR_BYTES,
        ROM_CORRECT_ERR_ADDRESS,
        ROM_CORRECT_ERR_BLOCKS,
        ROM_CORRECT_ERR_PARAM,
        ROM_CORRECT_ERR_NULL_PTR,
        ROM_CORRECT_TIMEOUT,
        ROM_CORRECT_ERR_TABLE,          /* テーブルデータエリアのエラー */
        ROM_CORRECT_ERR_PATCH,          /* パッチエリアのエラー */
        ROM_CORRECT_ERR_COPY_AREA_FLG,  /* コピーエリア書き換え完了フラグの書き込みエラー */
        ROM_CORRECT_ERR_PATCH_AREA_FLG, /* パッチ該当エリア書き換え完了フラグの書き込みエラー */
        ROM_CORRECT_NOT_APPLY           /* パッチ有無判定に有効データなし */
} rom_correct_err_t;
```

図 4.15　サンプルコードで使用する enum

4.8　変数一覧

表 4.7 にサンプルコードで使用する変数を示します。

表 4.7　static 型変数

型	変数名	内容	使用関数
static uint8_t	table_data_num	テーブルデータの登録数	• R_ROM_Correct • setting_patch

4.9　関数一覧

表 4.8 にサンプルコードで使用する関数を示します。

表 4.8　関数

関数名	説明
R_ROM_Correct	ROM コレクション処理
R_ROM_MakePatch	適用するパッチプログラムへ遷移（BRK 例外処理）
get_brk_address	BRK 例外発生アドレスの取得
set_return_adr	対応するパッチプログラムへの復帰設定
chk_apply_patch	パッチ有無を判定する処理
setting_patch	登録するパッチの設定
chk_patch	登録するパッチの確認
chk_table_data	テーブルデータの確認
chk_patch_area	パッチエリアの確認
chk_rewrite patch_target_area	パッチ該当エリアの書き換え確認
rewrite_patch_target_area	パッチ該当エリアの書き換え処理
chk_rewrite_copy_area	コピーエリアの書き換え確認
rewrite_copy_area	コピーエリアの書き換え処理
rewrite_brk_code	パッチ該当エリアへ BRK 命令の書き換え
call_R_FLASH_Open	エラー時のリトライを含むフラッシュ FIT モジュールの初期化
call_R_FLASH_Erase	エラー時のリトライを含むフラッシュの消去
call_R_FLASH_BlankCheck	エラー時のリトライを含むフラッシュのブランクチェック
call_R_FLASH_Write	エラー時のリトライを含むフラッシュの書き込み

4.10 関数仕様

サンプルコードで使用する関数を示します。

R_ROM_Correct	
概要	ROM コレクション関数
ヘッダ	r_rom_correct.h
宣言	rom_correct_err_t R_ROM_Correct (void)
説明	ROM コレクション処理
引数	なし
リターン値	ROM コレクションの設定結果 ROM_CORRECT_SUCCESS：成功 ROM_CORRECT_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない ROM_CORRECT_ERR_FAILURE：フラッシュのブランクチェック、消去、書き込み失敗 ROM_CORRECT_ERR_BYTES：フラッシュの書き込み/ブランクチェックサイズのエラー ROM_CORRECT_ERR_ADDRESS：無効なアドレス設定 ROM_CORRECT_ERR_BLOCKS：無効なフラッシュブロック No. ROM_CORRECT_ERR_TABLE：テーブルデータエリアの設定エラー ROM_CORRECT_ERR_PATCH：パッチエリアの設定エラー ROM_CORRECT_ERR_WR_COPY_FLAG：コピーエリア書き換え完了フラグの書き込みエラー ROM_CORRECT_ERR_WR_PATCH_FLAG：パッチ該当エリア書き換え完了フラグの書き込みエラー ROM_CORRECT_NOT_APPLY：パッチ有無判定に有効データなし
備考	ROM_CORRECT_NOT_APPLY 以外は、最後に登録されているテーブルデータの設定結果をリターン値で返します。 テーブルデータごとに設定結果の情報が必要であれば、1 つずつテーブルデータとパッチエリアを登録してください。

R_ROM_MakePatch	
概要	適用するパッチプログラムへ遷移（BRK 例外処理）
ヘッダ	r_rom_correction.h
宣言	void R_ROM_MakePatch (void)
説明	BRK 例外が発生したアドレスから適用するパッチプログラムを判定し、適用するパッチプログラムへ遷移します。
引数	なし
リターン値	なし
備考	本関数は BRK 例外関数です

get_brk_address	
概要	BRK 例外発生アドレスの取得
ヘッダ	r_rom_correction.h
宣言	uint32_t get_brk_address (void)
説明	BRK 例外が発生したアドレスを取得します。
引数	なし
リターン値	BRK 例外が発生したアドレス+1
備考	本関数は#pragma inline_asm で宣言しています。 R_ROM_MakePatch 関数を変更し、退避するスタック容量が変更された場合、合わせて本関数も変更する必要があります。

Set_return_adr	
概要	対応するパッチプログラムへの復帰設定
ヘッダ	r_rom_correction.h
宣言	uint32_t set_return_adr (uint32_t return_adr)
説明	BRK 例外処理から復帰するアドレスを設定します。
引数	uint32_t return_adr：適用するパッチプログラムの先頭アドレス
リターン値	なし
備考	本関数は#pragma inline_asm で宣言しています。 R_ROM_MakePatch 関数を変更し、退避するスタック容量が変更された場合、合わせて本関数も変更する必要があります。

chk_apply_patch	
概要	パッチ有無を判定する処理
ヘッダ	r_rom_correction.h
宣言	static rom_correct_err_t chk_apply_patch (void)
説明	パッチ有無判定の確認
引数	なし
リターン値	パッチ有無の確認結果 ROM_CORRECT_SUCCESS：パッチあり ROM_CORRECT_NOT_APPLY：パッチ有無判定に有効データなし
備考	

setting_patch	
概要	登録するパッチの設定
ヘッダ	r_rom_correction.h
宣言	static rom_correct_err_t setting_patch (void)
説明	テーブルデータエリア、パッチエリアに設定されたパッチを登録し、登録しているテーブルデータ数を保存します。
引数	なし
リターン値	実行結果 ROM_CORRECT_SUCCESS：成功 ROM_CORRECT_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない ROM_CORRECT_ERR_FAILURE：フラッシュのブランクチェック、消去、書き込み失敗 ROM_CORRECT_ERR_BYTES：フラッシュの書き込み/ブランクチェックサイズのエラー ROM_CORRECT_ERR_ADDRESS：無効なアドレス設定 ROM_CORRECT_ERR_BLOCKS：無効なフラッシュブロック No. ROM_CORRECT_ERR_TABLE：テーブルデータエリアの設定エラー ROM_CORRECT_ERR_PATCH：パッチエリアの設定エラー ROM_CORRECT_ERR_WR_COPY_FLAG：コピーエリア書き換え完了フラグの書き込みエラー ROM_CORRECT_ERR_WR_PATCH_FLAG：パッチ該当エリア書き換え完了フラグの書き込みエラー
備考	

chk_patch	
概要	登録するパッチの確認
ヘッダ	r_rom_correction.h
宣言	rom_correct_err_t chk_patch (const table_data_t table_data)
説明	テーブルデータエリア、パッチエリアに設定されたパッチを確認します。
引数	const table_data_t table_data：対象となるテーブルデータ
リターン値	実行結果 ROM_CORRECT_SUCCESS：成功 ROM_CORRECT_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない ROM_CORRECT_ERR_FAILURE：フラッシュのブランクチェック、消去、書き込み失敗 ROM_CORRECT_ERR_BYTES：フラッシュの書き込み/ブランクチェックサイズのエラー ROM_CORRECT_ERR_ADDRESS：無効なアドレス設定 ROM_CORRECT_ERR_BLOCKS：無効なフラッシュブロック No. ROM_CORRECT_ERR_TABLE：テーブルデータエリアの設定エラー ROM_CORRECT_ERR_PATCH：パッチエリアの設定エラー ROM_CORRECT_ERR_WR_COPY_FLAG：コピーエリア書き換え完了フラグの書き込みエラー ROM_CORRECT_ERR_WR_PATCH_FLAG：パッチ該当エリア書き換え完了フラグの書き込みエラー
備考	

chk_table_data	
概要	テーブルデータの確認
ヘッダ	r_rom_correction.h
宣言	rom_correct_err_t chk_table_data (const table_data_t table_data)
説明	対象のテーブルデータが正しく設定されているかを確認します。
引数	const table_data_t table_data：対象となるテーブルデータ
リターン値	対象テーブルデータの確認結果 ROM_CORRECT_SUCCESS：対象のテーブルデータが正しい ROM_CORRECT_ERR_TABLE ：対象のテーブルデータが正しくない
備考	

chk_patch_area	
概要	パッチエリアの確認
ヘッダ	r_rom_correction.h
宣言	rom_correct_err_t chk_patch_area (const uint32_t pp_adr)
説明	対象のパッチエリアが正しく設定されているかを確認します。
引数	const uint32_t pp_adr：対象となるテーブルデータの PP_ADR
リターン値	対象パッチエリアの確認結果 ROM_CORRECT_SUCCESS：対象のパッチエリアが正しい ROM_CORRECT_ERR_PATCH ：対象のパッチエリアが正しくない
備考	

chk_rewrite_patch_target_area	
概要	パッチ該当エリアの書き換え確認
ヘッダ	r_rom_correction.h
宣言	rom_correct_err_t chk_rewrite_patch_target_area (const table_data_t table_data)
説明	パッチ該当エリアの書き換えを確認します。
引数	const table_data_t table_data：対象となるテーブルデータ
リターン値	実行結果 ROM_CORRECT_SUCCESS：成功 ROM_CORRECT_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない ROM_CORRECT_ERR_FAILURE：フラッシュのブランクチェック、消去、書き込み失敗 ROM_CORRECT_ERR_BYTES：フラッシュの書き込み/ブランクチェックサイズのエラー ROM_CORRECT_ERR_ADDRESS：無効なアドレス設定 ROM_CORRECT_ERR_BLOCKS：無効なフラッシュブロック No. ROM_CORRECT_ERR_WR_COPY_FLAG：コピーエリア書き換え完了フラグの書き込みエラー ROM_CORRECT_ERR_WR_PATCH_FLAG：パッチ該当エリア書き換え完了フラグの書き込みエラー
備考	

rewrite_patch_target_area	
概要	パッチ該当エリアの書き換え処理
ヘッダ	r_rom_correction.h
宣言	rom_correct_err_t rewrite_patch_target_area (const table_data_t_t table_data)
説明	パッチ該当エリアを書き換えます。
引数	const table_data_t table_data：対象のテーブルデータ
リターン値	実行結果 ROM_CORRECT_SUCCESS：成功 ROM_CORRECT_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない ROM_CORRECT_ERR_FAILURE：フラッシュのブランクチェック、消去、書き込み失敗 ROM_CORRECT_ERR_BYTES：フラッシュの書き込み/ブランクチェックサイズのエラー ROM_CORRECT_ERR_ADDRESS：無効なアドレス設定 ROM_CORRECT_ERR_BLOCKS：無効なフラッシュブロック No. ROM_CORRECT_ERR_WR_COPY_FLAG：コピーエリア書き換え完了フラグの書き込みエラー ROM_CORRECT_ERR_WR_PATCH_FLAG：パッチ該当エリア書き換え完了フラグの書き込みエラー
備考	

chk_rewrite_copy_area	
概要	コピーエリアの書き換え確認
ヘッダ	r_rom_correction.h
宣言	rom_correct_err_t chk_rewrite_copy_area (const table_data_t table_data)
説明	コピーエリアの書き換えを確認します。
引数	const table_data_t table_data：対象となるテーブルデータ
リターン値	実行結果 ROM_CORRECT_SUCCESS：成功 ROM_CORRECT_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない ROM_CORRECT_ERR_FAILURE：フラッシュのブランクチェック、消去、書き込み失敗 ROM_CORRECT_ERR_BYTES：フラッシュの書き込み/ブランクチェックサイズのエラー ROM_CORRECT_ERR_ADDRESS：無効なアドレス設定 ROM_CORRECT_ERR_BLOCKS：無効なフラッシュブロック No. ROM_CORRECT_ERR_WR_COPY_FLAG：コピーエリア書き換え完了フラグの書き込みエラー
備考	

rewrite_copy_area	
概要	コピーエリアの書き換え処理
ヘッダ	r_rom_correction.h
宣言	static flash_err_t rewrite_copy_area (const uint32_t pp_adr)
説明	コピーエリアをパッチ該当エリアのデータに書き換えます。 パッチ該当エリアの書き込み中に瞬時電圧低下などで書き込みが失敗する可能性を考慮し、BRK 命令に書き換えする前のデータをコピーエリアに保存しています。
引数	const uint32_t pp_adr：対象となるテーブルデータの PP_ADR
リターン値	実行結果 FLASH_SUCCESS：成功 FLASH_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない FLASH_ERR_FAILURE：フラッシュの消去、書き込み失敗 FLASH_ERR_BYTES：フラッシュ書き込みサイズのエラー FLASH_ERR_ADDRESS：無効なアドレス設定 FLASH_ERR_BLOCKS：無効なフラッシュブロック No.
備考	

rewrite_brk_code	
概要	パッチ該当エリアへ BRK 命令の書き換え
ヘッダ	r_rom_correction.h
宣言	static flash_err_t rewrite_brk_code (const uint32_t pp_adr)
説明	パッチ該当エリアの書き換えアドレスを BRK 命令に書き換えます。 パッチ該当エリアの書き込み中に瞬時電圧低下などで書き込みが失敗する可能性を考慮し、BRK 命令に書き換えする前のデータはコピーエリアに保存されています。
引数	const uint32_t pp_adr：対象となるテーブルデータの PP_ADR
リターン値	FLASH_SUCCESS：成功 FLASH_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない FLASH_ERR_FAILURE：フラッシュの消去、書き込み失敗 FLASH_ERR_BYTES：フラッシュ書き込みサイズのエラー FLASH_ERR_ADDRESS：無効なアドレス設定 FLASH_ERR_BLOCKS：無効なフラッシュブロック No.
備考	

call_R_FLASH_Open	
概要	エラー時のリトライを含むフラッシュ FIT モジュールの初期化
ヘッダ	r_rom_correction.h
宣言	static flash_err_t call_R_FLASH_Open (void)
説明	フラッシュ FIT モジュールを初期化します。エラー発生時のリトライを含みます。
引数	なし
リターン値	FLASH_SUCCESS：成功 FLASH_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない
備考	

call_R_FLASH_Erase	
概要	エラー時のリトライを含むフラッシュの消去
ヘッダ	r_rom_correction.h
宣言	static flash_err_t call_R_FLASH_Erase (uint32_t block_start_address, uint32_t num_blocks)
説明	フラッシュを消去します（エラー発生時のリトライを含みます）
引数	uint32_t block_start_address：消去対象ブロックの先頭アドレス uint32_t num_blocks：消去対象のブロック数
リターン値	FLASH_SUCCESS：成功 FLASH_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない FLASH_ERR_FAILURE：フラッシュの消去失敗 FLASH_ERR_ADDRESS：無効なアドレス設定 FLASH_ERR_BLOCKS：無効なフラッシュブロック No.
備考	

call_R_FLASH_BlankCheck	
概要	エラー時のリトライを含むフラッシュのブランクチェック
ヘッダ	r_rom_correction.h
宣言	static flash_err_t call_R_FLASH_BlankCheck (uint32_t address, uint32_t num_bytes, flash_res_t *result)
説明	フラッシュのブランクチェックをします（エラー発生時のリトライを含みます）
引数	uint32_t address：ブランクチェックの開始アドレス uint32_t num_bytes：ブランクチェックのバイト数 flash_res_t *result：ブランクチェックの結果ポインタ
リターン値	FLASH_SUCCESS：成功 FLASH_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない FLASH_ERR_FAILURE：フラッシュのブランクチェック失敗 FLASH_ERR_BYTES：フラッシュのブランクチェックサイズのエラー FLASH_ERR_ADDRESS：無効なアドレス設定
備考	

call_R_FLASH_Write	
概要	エラー時のリトライを含むフラッシュの書き込み
ヘッダ	r_rom_correction.h
宣言	static flash_err_t call_R_FLASH_Write (uint32_t src_address , uint32_t dest_address, uint32_t num_bytes)
説明	フラッシュに書き込みします（エラー発生時のリトライを含みます）
引数	uint32_t src_address：書き込むデータが格納されている先頭アドレス uint32_t dest_address：書き込み先の先頭アドレス uint32_t num_bytes：書き込みバイト数
リターン値	FLASH_SUCCESS：成功 FLASH_ERR_BUSY：ソフトウェアコマンドを実行中、またはフラッシュ FIT モジュールが初期化されていない FLASH_ERR_FAILURE：フラッシュの書き込み失敗 FLASH_ERR_BYTES：フラッシュの書き込みサイズのエラー FLASH_ERR_ADDRESS：無効なアドレス設定
備考	

5. 注意事項

5.1 アプリケーションノート使用時の注意事項

- テーブルデータエリア、パッチエリアへは P/E モードによる自己書き換えなどであらかじめユーザにて書き込んでください。また、シリアルプログラマなど、ブートプログラムを用いての書き込みはできません。
- テーブルデータエリア、パッチエリアへ書き込むとき、予約領域、空き領域へは FFh を書かないでください。
- ユーザが使用する R_ROM_Correct 関数は、最後に登録されているテーブルデータの設定結果をリターン値で返します。テーブルデータごとに設定結果の情報が必要であれば、1 つずつテーブルデータとパッチエリアを登録してください。

5.2 テーブルデータエリア、パッチエリアへの書き込み手順例

ユーザにて実施するテーブルデータエリア、パッチエリアへの書き込み手順例を以下に示します。図 5.1 にテーブルデータエリアの書き込み手順を例に示します。また、瞬時電圧低下などによる書き込み失敗時の書き込み手順例について図 5.2 に示します。なお、書き込み前にはブランクチェックを行ってください。

1. 最初にパッチを当てる場合のみ、テーブルデータエリアのパッチ有無判定へ AA55 AA55h を書いてください。2 回目以降にパッチを当てる場合、パッチ有無判定へ AA55 AA55h の書き込みは不要です。
2. テーブルデータエリア、パッチエリアの順に書いてください。
   2-1. テーブルデータ書き込み後、テーブルデータ書き換え完了フラグ（FLG_ADR の反転値）を書いてください。

   2-2. パッチエリア書き込み後、パッチエリア書き換え完了フラグ（書き換えアドレスの反転値）を書いてください。
   2-3. パッチを追加で書く場合、2-1、2-2 を繰り返してください。瞬時電圧低下などにより書き込みを失敗した場合、次の空き領域に書き直してください。ただし、テーブルデータは FLG_ADR、PP_ADR、テーブルデータ書き換え完了フラグの 12 バイトを 1 セットとし、FLG_ADR(m)は FLG_ADR(m-1)の書き込みアドレスに 0Ch を足したアドレスとしてください。

注.　パッチエリア書き込み時に失敗した場合も PP_ADR が変わるため、2-1 からやり直してください。また、書き込み失敗時は消去せず、次の領域に書き込んでください。

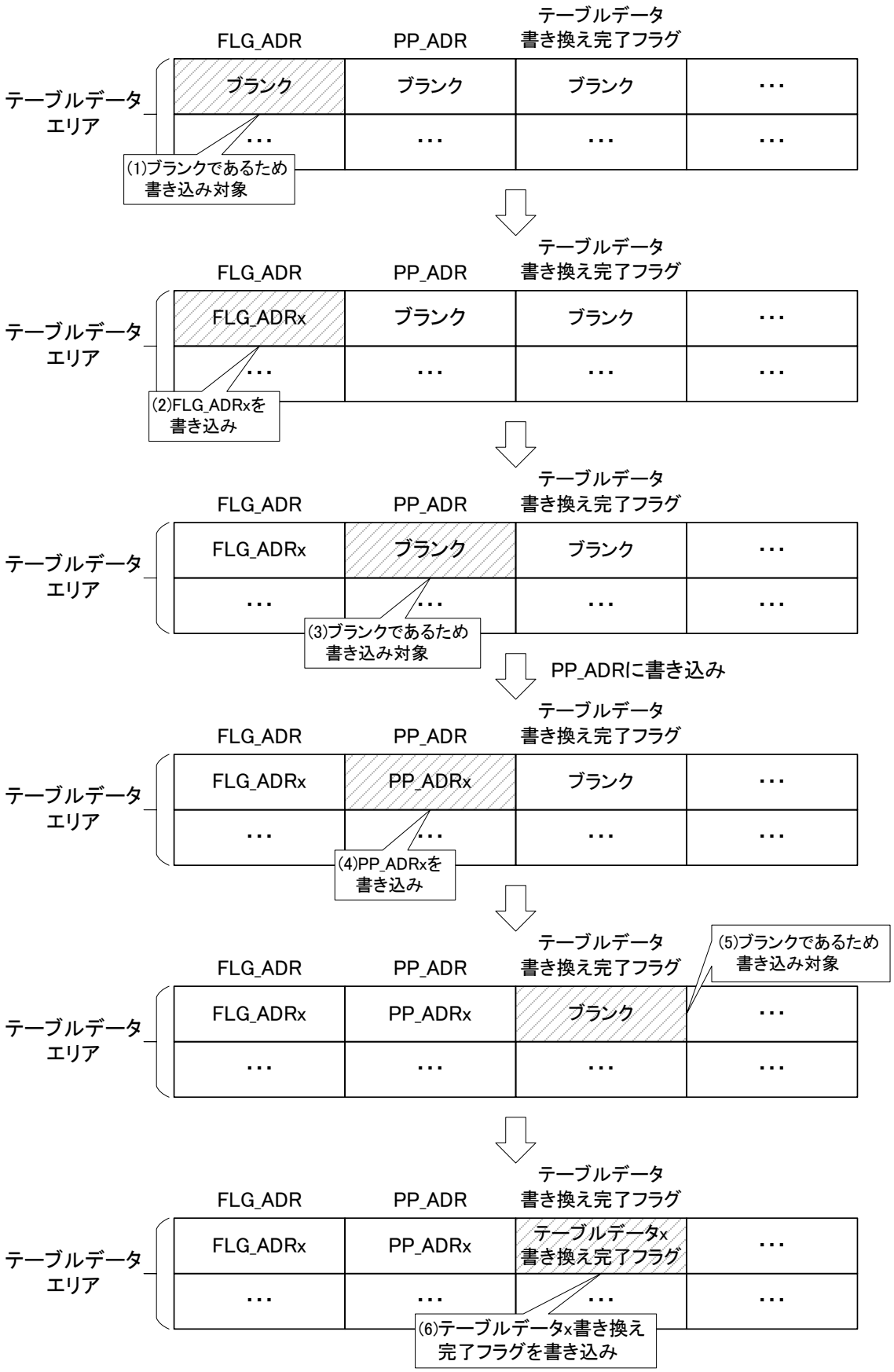


図 5.1　テーブルデータエリアの書き込み手順例

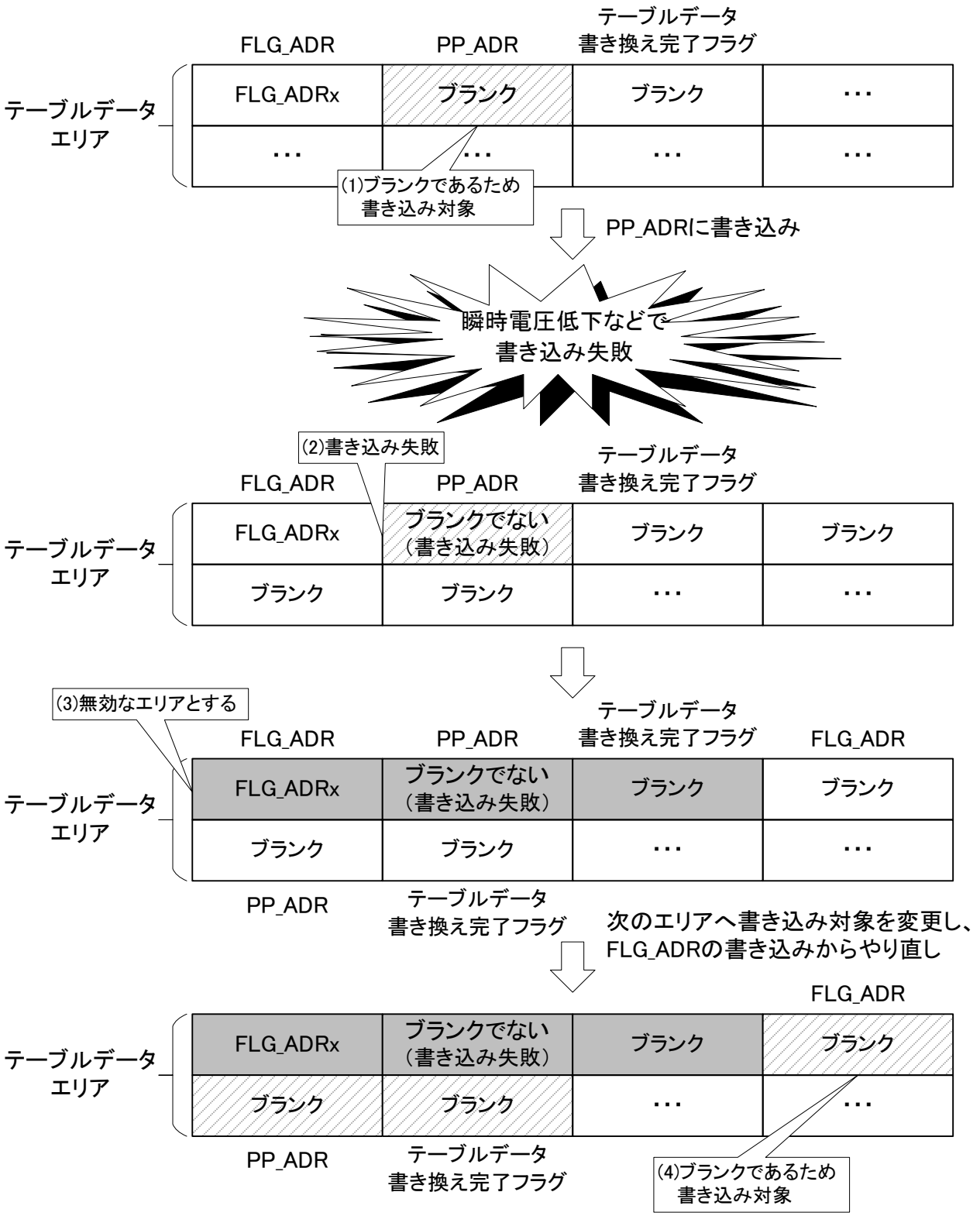


図 5.2　瞬時電圧低下などで書き込みに失敗した場合のテーブルデータエリアの書き込み手順例

5.3　他製品へ仕様を適用する場合の考え方

本アプリケーションノートを他の製品へ適用する場合、製品よってユーザースタックサイズが増える場合があります。図 5.3 および図 5.4 に RX65N でパッチ該当エリアが 32K バイトの例を示します。

また、Flash FIT モジュールを使用する場合、「FLASH_CFG_CODE_FLASH_ENABLE」の設定を”1”に変更し、「RX Family Flash Module Using Firmware Integration Technology (R01AN2184)」の「RAM からコードを実行してコードフラッシュを書き換える場合」を参照しセクションを追加してください。

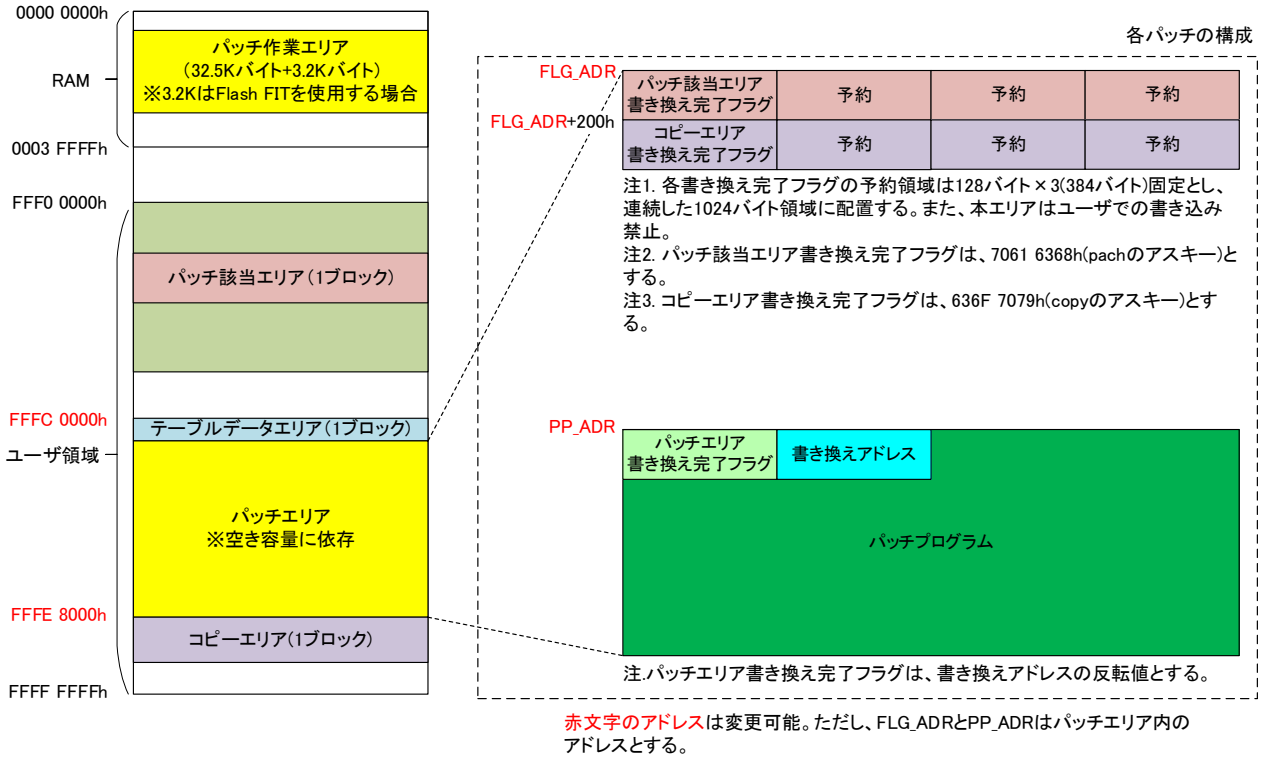


図 5.3　RX65N でパッチ該当エリアが 32K バイトでのパッチエリアの考え方

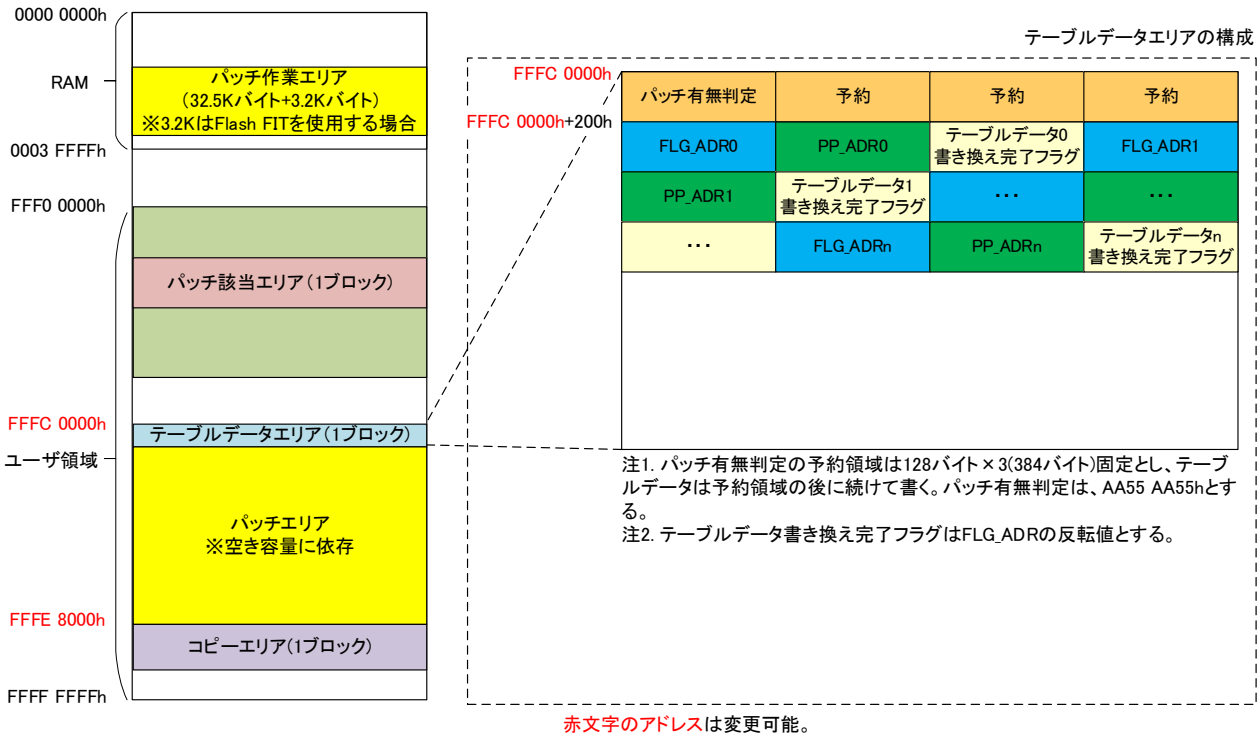


図 5.4　RX65N でパッチ該当エリアが 32K バイトでのテーブルデータエリアの考え方

6.　テーブルデータエリア、パッチエリアのデータ作成方法

ユーザが設定するテーブルデータエリア、パッチエリアのデータ作成方法について、パッチエリアを例に説明します。

6.1　パッチエリアのデータ作成例

図 6.1 に示すパッチエリア 0 を設定する例を示します。

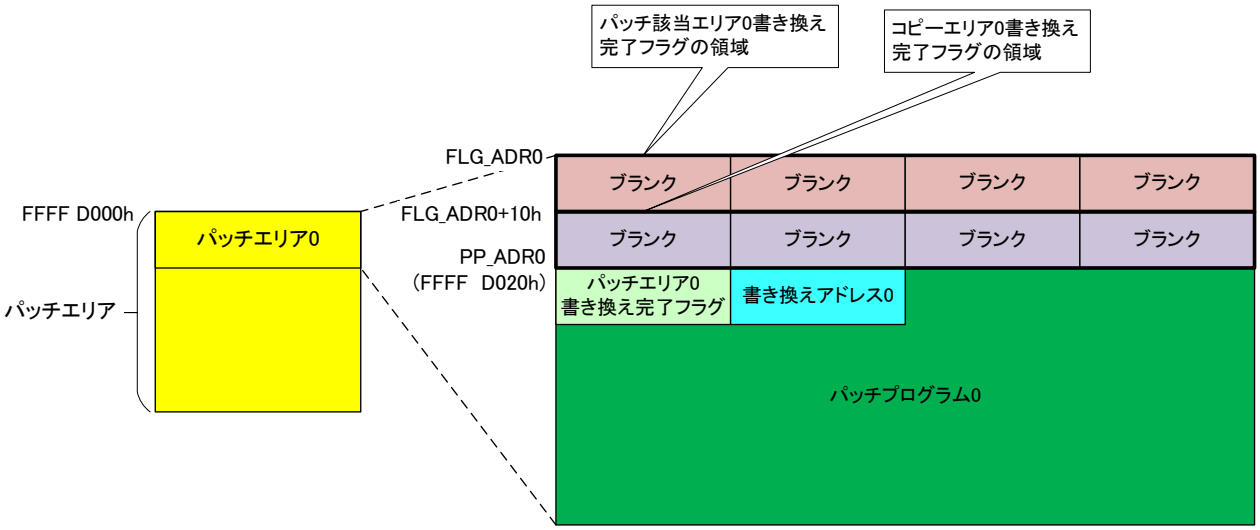


図 6.1　パッチエリアの設定例

#pragma section でパッチエリアの PP_ADR0 を設定し、const 宣言でパッチエリア書き換え完了フラグ、書き換えアドレスを記述後、続けて適用するパッチプログラムを記述します。書き込み手順に関しては、5.2 テーブルデータエリア、パッチエリアへの書き込み手順例を参照ください。

```
#pragma section PP_ADR0

/* パッチエリア書き換え完了フラグ */
const uint32_t g_rewrite_flg0  = XXXXXXXX;
/* 書き換えアドレス */
const uint32_t g_rewrite_adr0 = YYYYYYYY;

void patch_prg0( void )
{
        適用するパッチプログラム処理;
        ・
        ・
        ・
        指定したユーザプログラムへジャンプする処理;
}

#pragma section          /* 該当セクションの終了 */
```

図 6.2　パッチエリアの記述例

記述したセクションに e2 studio 上でアドレスを設定します。メニューバーの”プロジェクト” → “C/C++ Project Setting”をクリックします。

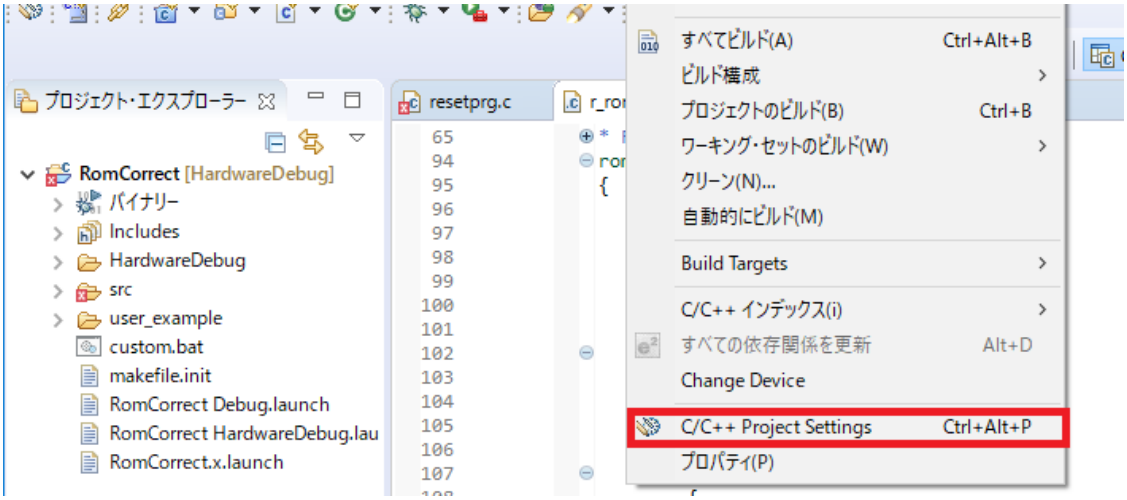


図 6.3　セクションの設定例(1/2)

表示される”プロパティ：RomCorrect”ウィンドウを図 6.4 に示します。C/C++ ビルド → Settings → Linker → セクションにおいて、CPP_ADR0、PPP_ADR0 のセクションを追加し、対応するアドレスを設定して”OK”をクリックします。

その後、ビルドを実行すると、セクション内のデータが設定したアドレスへ配置された実行ファイルが生成されます。

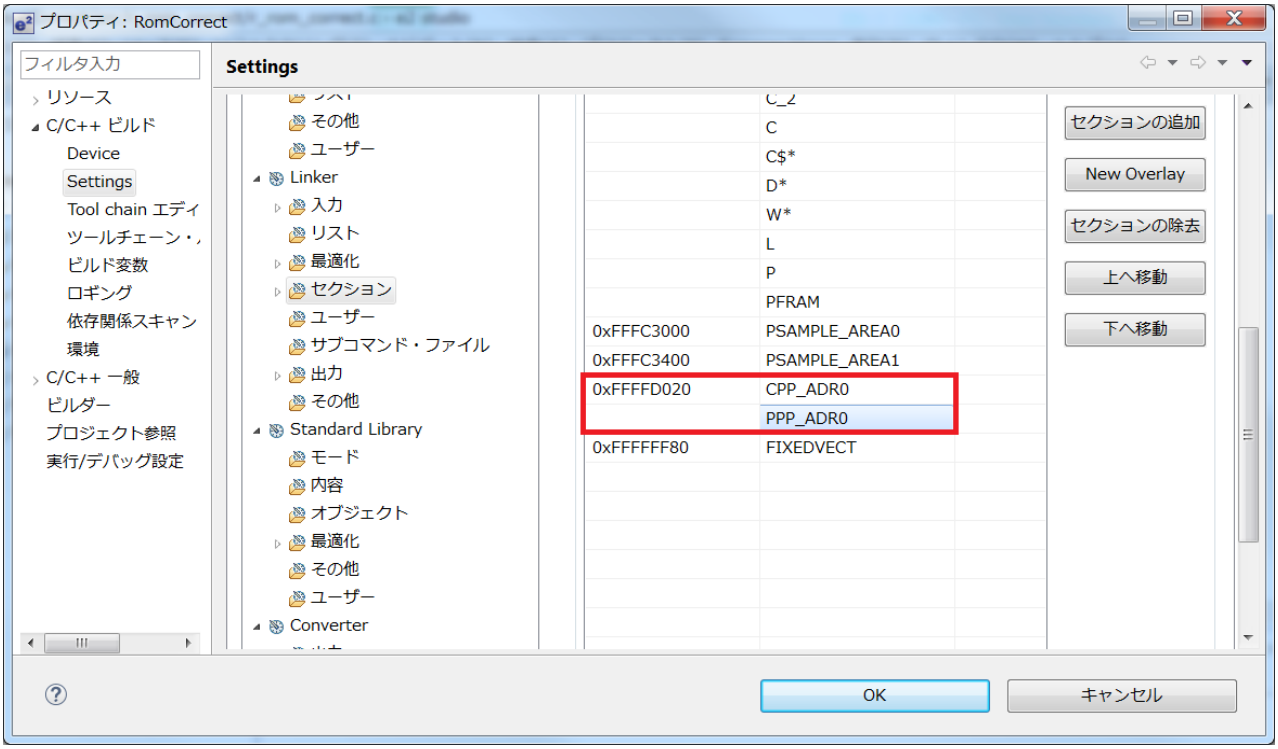


図 6.4　セクションの設定例(2/2)

6.2　パッチエリアのデータを抽出する手順例

指定したメモリ領域からデータを抽出する手順を説明します。

e2 studio とユーザシステムを接続し、プログラムをダウンロード後にメニューバーの”ウィンドウ” → ”ビューの表示” → ”メモリー”をクリックしてメモリーウィンドウを表示します（図 6.5 を参照）。

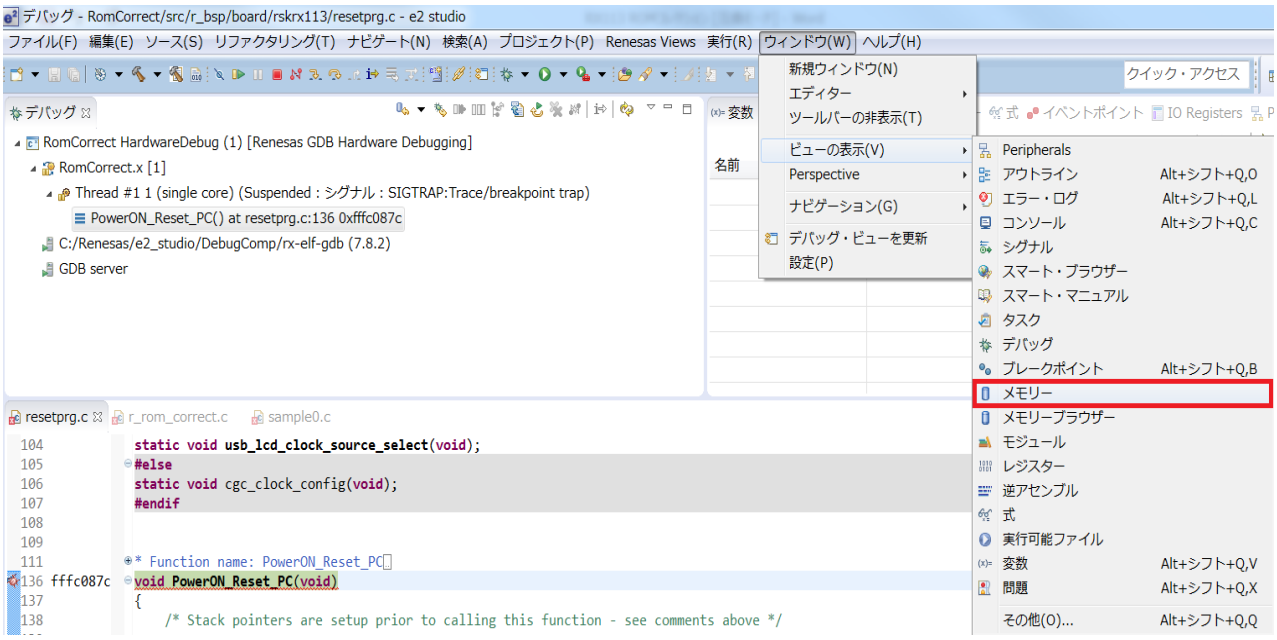


図 6.5　メモリの表示方法

表示される”メモリー”ウィンドウを図 6.6 に示します。”メモリー”ウィンドウで”メモリー・モニターの追加”をクリックします。

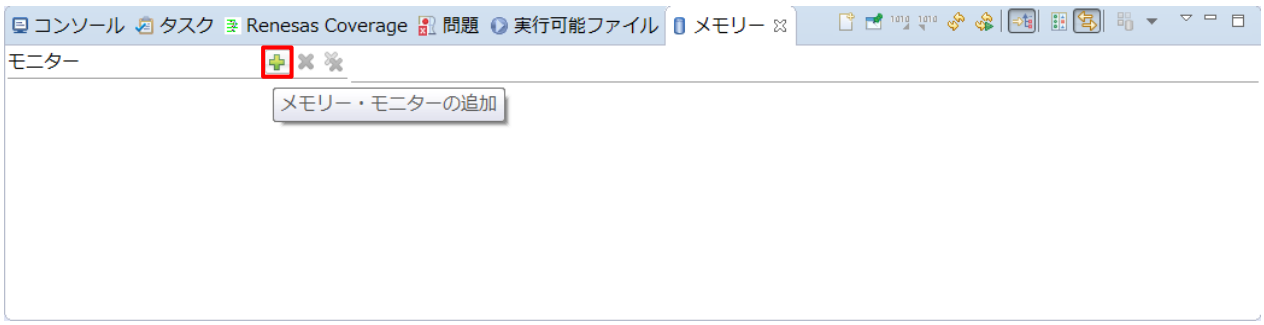


図 6.6　“メモリー”ウィンドウ

表示される”モニター・メモリー”ウィンドウを図 6.7 に示します。表示するメモリ領域のアドレスを入力後、”OK”をクリックします。

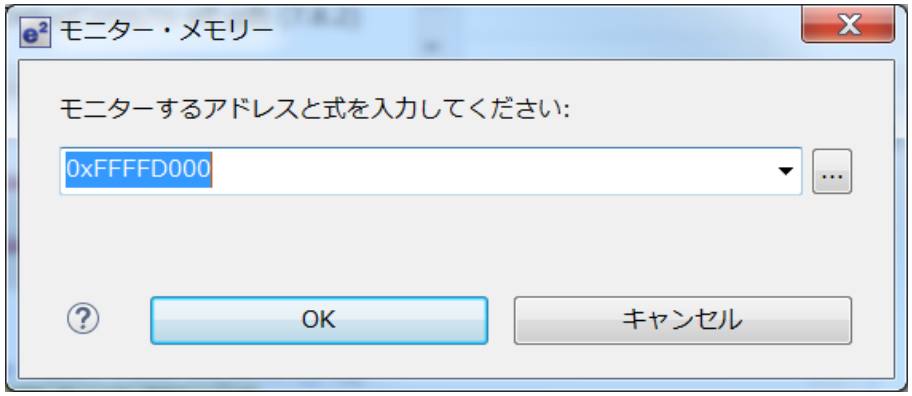


図 6.7　“モニター・メモリー”ウィンドウ

指定したアドレスのメモリ領域が図 6.8 のように表示されます。

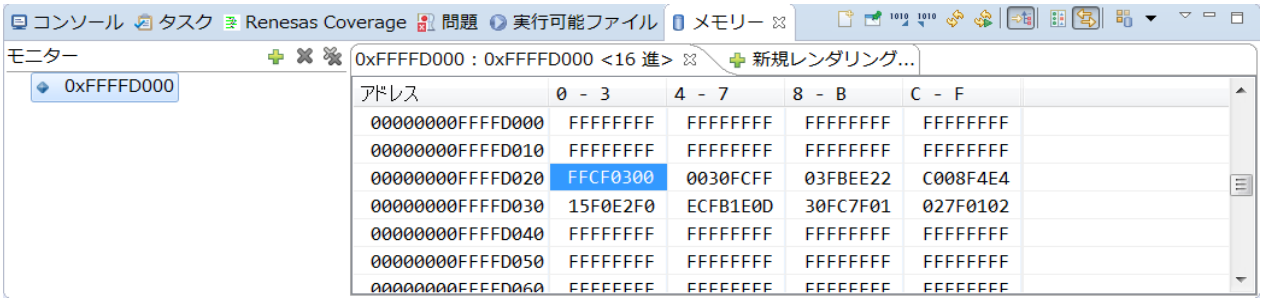


図 6.8　メモリ領域の表示

指定したメモリ領域からデータを抽出する方法を説明します。”メモリー”ウィンドウの”エクスポート”をクリックします（図 6.9 を参照）。

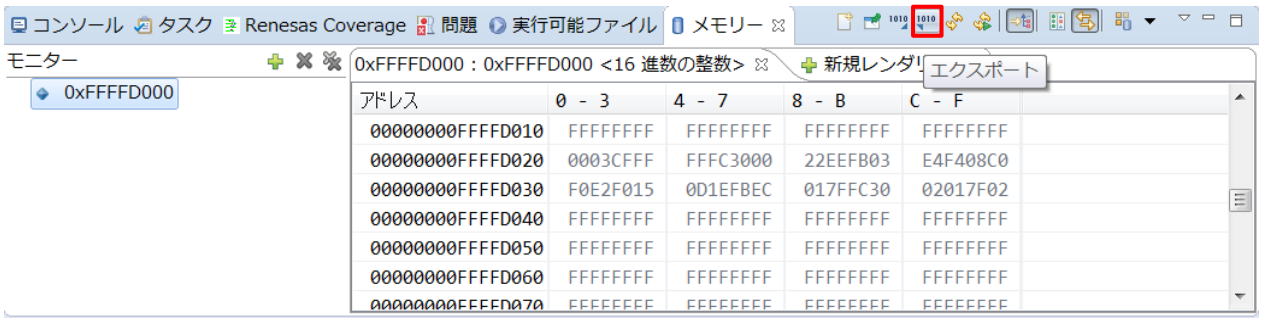


図 6.9　メモリ領域のエクスポート

表示される”メモリーをエクスポート”ウィンドウを図 6.10 に示します。”形式”に保存するファイル形式を選択してください。抽出したいメモリ領域は開始アドレスから終了アドレスの直前までとなり、開始アドレスから長さ（バイト）分です。”ファイル名”に保存するファイルの名前を設定後、”OK”をクリックしてください。

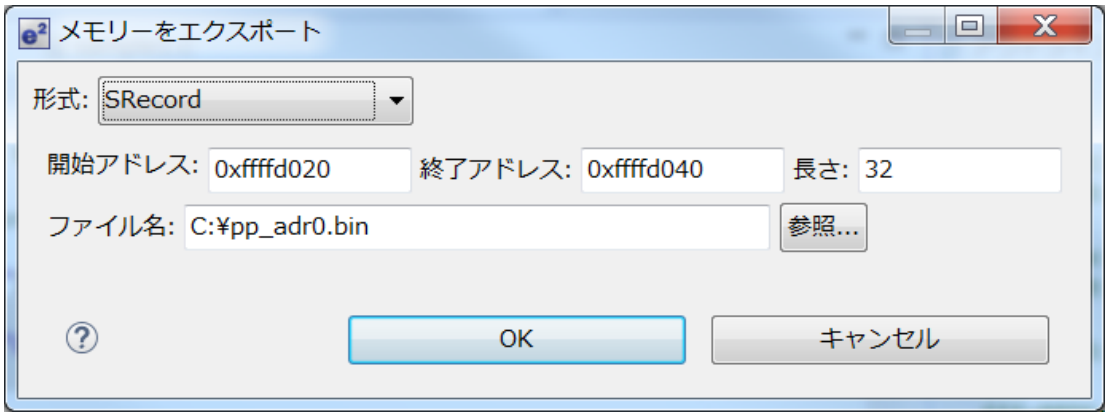


図 6.10　“メモリーをエクスポート”ウィンドウ

7. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX113 グループ ユーザーズマニュアル ハードウェア編（R01UH0448）

（最新版をルネサス エレクトロニクスホームページから入手してください。）

テクニカルアップデート／テクニカルニュース

（最新の情報をルネサス エレクトロニクスホームページから入手してください。）

ユーザーズマニュアル：開発環境

RX ファミリ CC-RX コンパイラユーザーズマニュアル（R20UT3248）

（最新版をルネサス エレクトロニクスホームページから入手してください。）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jan.31.20	ー	初版発行

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、ノイズ幅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。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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

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

(Rev.4.0-1　2017.11)

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。