RX63N グループ、RX631グループ

USB ペリフェラル CDC によるフラッシュブートローダ

R01AN1710JJ0100
Rev.1.00
2014.01.06

要旨

本アプリケーションノートでは、RX63N グループ、RX631グループの USB2.0 ホスト／ファンクションモジュールを、ファンクション動作で使用し、USB 経由で内蔵フラッシュメモリの書き換えを行う方法(USB によるフラッシュブートローダ)について説明します。

USB によるフラッシュブートローダの特長を以下に示します。

- PC からのコマンド送信による対象デバイス制御
  対象デバイスのフラッシュ消去／書き込み／ブランクチェック／ターゲットプログラムの実行を、PC からのコマンドにより行います。
- Motorola S フォーマットのプログラムを書き込み可能
- USB 規格 2.0 のフルスピード転送に対応
- USB コミュニケーションデバイスクラス仕様の Abstract Control Model に準拠

対象デバイス

・RX63N グループ	177、176 ピン版	ROM 容量：768KB～2MB
・RX63N グループ	145、144 ピン版	ROM 容量：768KB～2MB
・RX63N グループ	100 ピン版	ROM 容量：768KB～2MB
・RX631 グループ	177、176 ピン版	ROM 容量：256KB～2MB
・RX631 グループ	145、144 ピン版	ROM 容量：256KB～2MB
・RX631 グループ	100 ピン版	ROM 容量：256KB～2MB

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あわせて変更し、十分評価してください。

目次

1.　仕様

USB によるフラッシュブートローダでは、ホスト PC で起動するターミナルソフトからマイコンに対してコマンドを送信し、マイコンのフラッシュメモリの書き換えを行います。

表 1.1に使用する周辺機能と用途を、図 1.1に使用例を示します。

表1.1　使用する周辺機能と用途

周辺機能	用途
ROM (コード格納用フラッシュメモリ)	ROM P/E モードによる 内蔵フラッシュメモリの書き換え
USB2.0 ホスト／ファンクションモジュール	ホスト PC との通信用

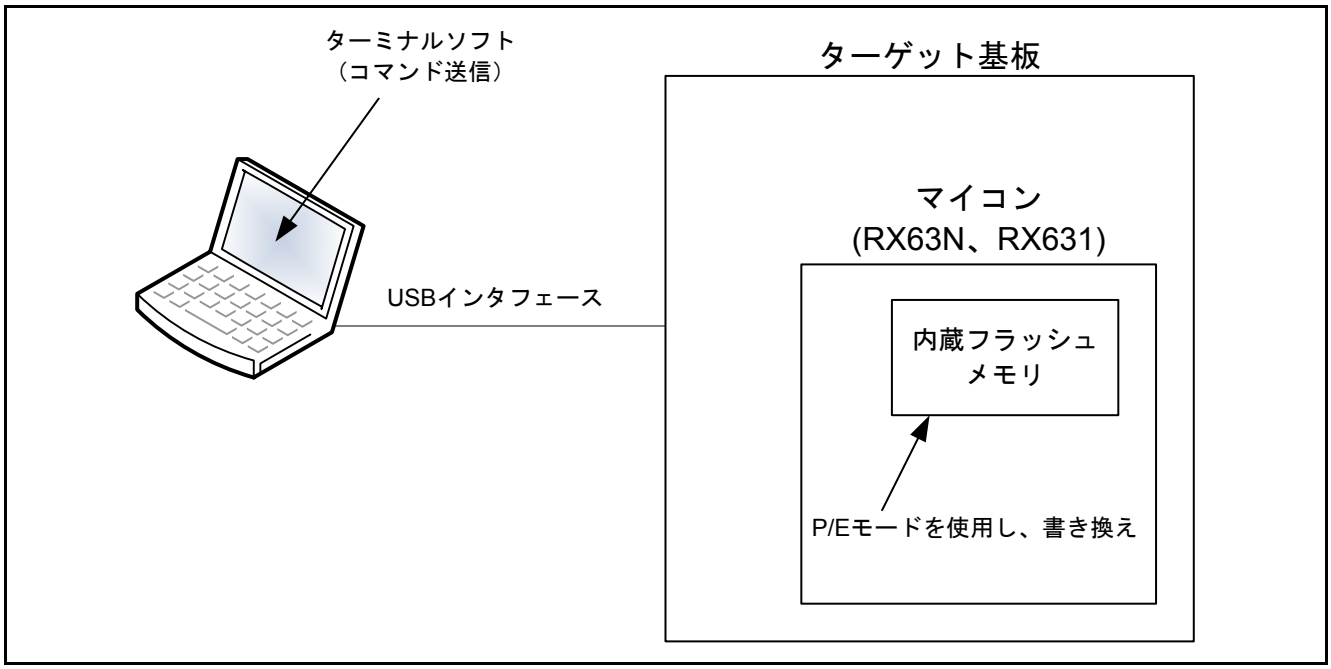

図1.1　使用例

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	R5F563NBDDFC (RX63N グループ)
動作周波数	• メインクロック: 12MHz • PLL: 192MHz (メインクロック 1 分周 16 逓倍) • システムクロック(ICLK): 96MHz (PLL 2 分周) • FlashIF クロック(FCLK): 48MHz (PLL 4 分周) • 外部バスクロック(BCLK):24MHz (PLL8 分周) • 周辺モジュールクロック B (PCLKB): 48MHz (PLL 4 分周) • USB クロック(UCLK): 48MHz (PLL 4 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 High-performance Embedded Workshop Version 4.09.01
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 V.1.02 Release 01
	コンパイルオプション --cpu=rx600 -bit_order=left -include="$(WORKSPDIR)¥WorkSpace¥ANSI" -include="$(WORKSPDIR)¥WorkSpace¥CDCFW¥include" -include="$(WORKSPDIR)¥WorkSpace¥HwResourceForUSB¥inc" -include="$(WORKSPDIR)¥WorkSpace¥HwResourceForUSB¥USBHW" -include="$(WORKSPDIR)¥WorkSpace¥HwResourceForUSB¥USBHW¥DEF" -include="$(WORKSPDIR)¥WorkSpace¥HwResourceForUSB¥USBHW¥REG" -include="$(WORKSPDIR)¥WorkSpace¥HwResourceForUSB¥USRCFG" -include="$(WORKSPDIR)¥WorkSpace¥SmplMain¥APL" -include="$(WORKSPDIR)¥WorkSpace¥USBSTDFW¥include" -include="$(WORKSPDIR)¥WorkSpace¥FLASH" -include="$(WORKSPDIR)¥WorkSpace¥FLASH¥src" -include="$(WORKSPDIR)¥WorkSpace¥r_bsp" -define=USB_FW_PP=USB_FW_NONOS_PP,USBC_DEBUGLCD_PP -output=obj="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -nostuff -nologo
iodefine.h のバージョン	Version 1.6A
エンディアン	リトルエンディアン
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 RX63N (製品型名: R0K50563NC000BE)

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トを以下に示します。併せて参照してください。

- RX600 & RX200 シリーズ RX 用シンプルフラッシュ API　Rev.2.40　（R01AN0544JU0240_RX）
- Renesas USB MCU and USB ASSP USB Basic Host and Peripheral firmware Rev.2.10 （R01AN0512JJ0210_USB）
- Renesas USB MCU and USB ASSP USB Peripheral Communications Device Class Driver (PCDC) Rev.2.10　（R01AN0273JJ0210_USB）
- RX63N グループ、RX631グループ　初期設定例　Rev.1.10（R01AN1245JJ0110_RX63N）

上記アプリケーションノートの関数を、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。Rev は本アプリケーションノート作成時点のものです。
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.　ハードウェア説明

4.1　　使用端子一覧

表 4.1に使用端子と機能を示します。

使用端子は 176 ピン版の製品を想定しています。176 ピン版未満の製品を使用する場合は、使用する製品に合わせて端子を選択してください。

表4.1　使用端子と機能

端子名	入出力	内容
USB1_DP	入出力	ポート 1　USB 内蔵トランシーバ D+ 入出力端子 USB バス の D+端子に接続
USB1_DM	入出力	ポート 1　USB 内蔵トランシーバ D- 入出力端子 USB バスの D- 端子に接続
USB1_VBUS	入力	ポート 1 USB ケーブル接続モニタ端子 USB バスの VBUS に接続。ファンクション動作時の VBUS の接続／切断を検出することが可能
USB1_DPUPE	出力	ポート 1 ファンクション動作時の USB D+信号の 1.5kΩ プルアップ抵抗の制御信号

注：　USB Peripheral Communication Device Class Driver で使用しているその他の端子設定(スイッチや SCI 等)については、USB Peripheral Communication Device Class Driver のアプリケーションノートを参照してください。

5.　ソフトウェア説明

5.1　動作概要

サンプルコードは、ホスト PC からコマンドデータを受信し、そのコマンドに対応した動作(メニュー表示、ブランクチェック、消去、書き込み、または書き込んだプログラム(ターゲットプログラム)を実行)を行います。ホスト PC からのコマンド送信はターミナルソフトで行います。サンプルコードが書き換える対象は、ユーザ領域の一部(ターゲットエリア)のみとなります。サンプルコード自身が使用している領域(FFFE 0000h ～ FFFF FFFFh)の書き換えは行えません。

図 5.1にメモリ配置を示します。

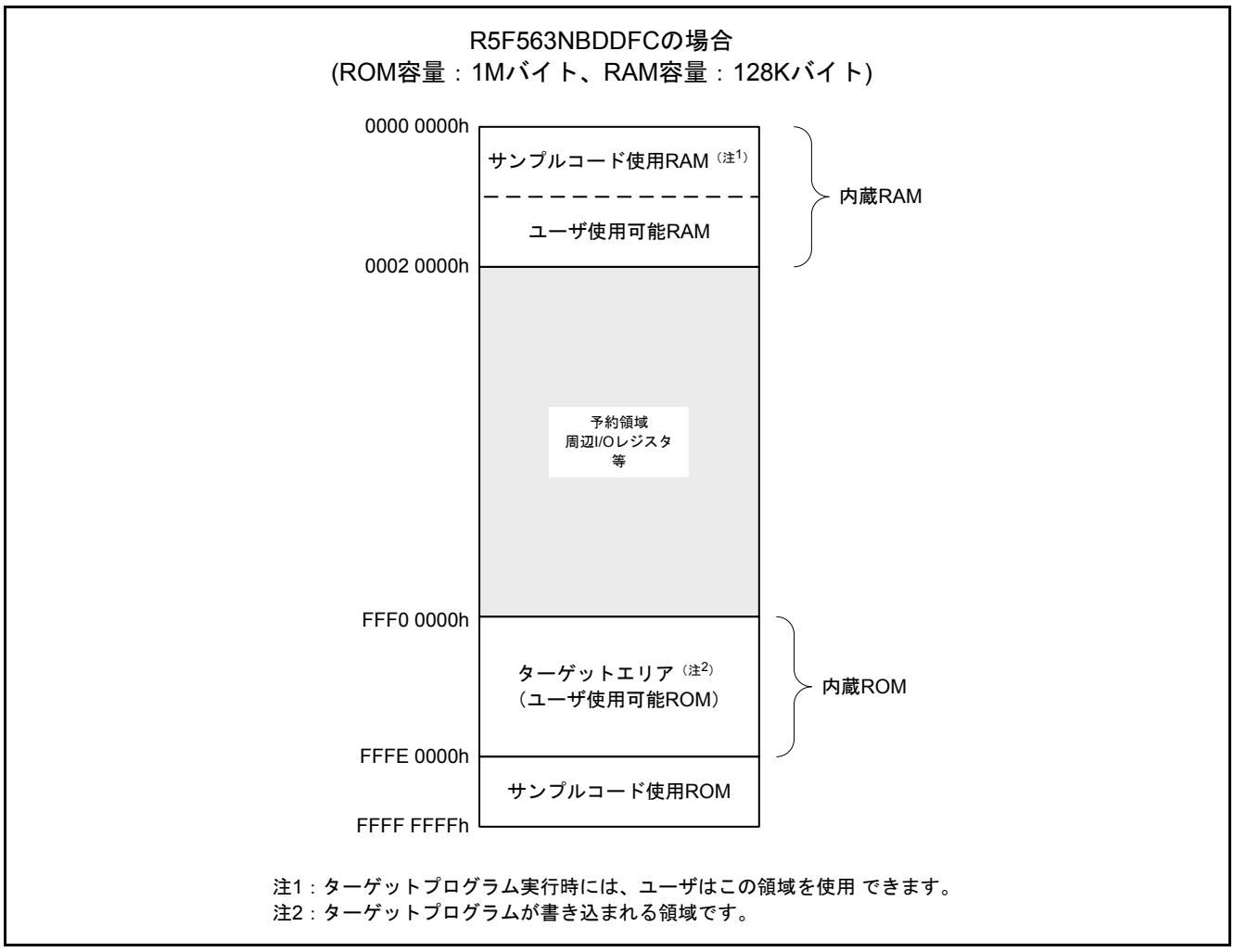

図5.1　メモリ配置

サンプルコードでは、以下の(1)～(3)の順にフラッシュメモリ書き換えを行います。

(1) リセット解除後、PC と接続されていない、かつ“FFFD FFFCh”番地(ターゲットリセットベクタ)が FFFF FFFFh でない場合、ターゲットプログラムを実行します。
PC と接続されていれば、マイコンに任意のデータが送られてくるまで(ターミナルソフトを起動し、キーボードのいずれかのキーを押すまで)、一定間隔でメッセージ(“Press any key”)を PC へ送信します。(図 5.2)

(2) PC から任意のデータが送られてくると、PC へメニューを送信し、コマンドデータの受信を待ちます。(図 5.3)

(3) コマンドを受信すると、そのコマンドに対応した動作(メニュー表示、ブランクチェック、消去、書き込み、ターゲットプログラム実行)を行います。

図5.2　キー入力待ち画面例

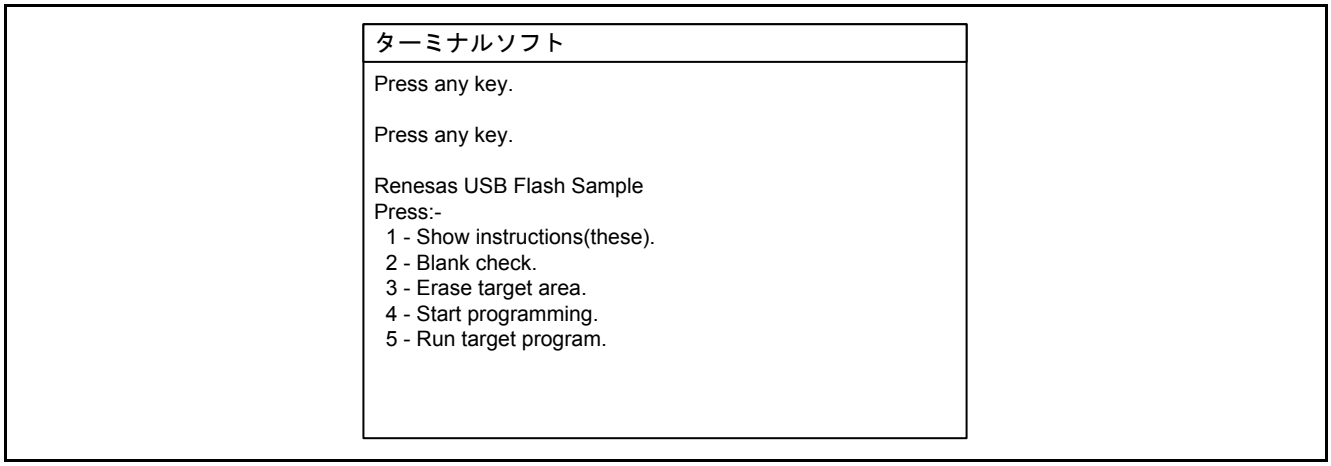

図5.3　コマンド入力画面例

5.2　コマンド一覧

表 5.1にコマンド一覧を示します。コマンド一覧は「メニュー表示」を実行することでターミナルソフトに表示されます。

表5.1　コマンド一覧

PC で入力するキー	ASCII コード	内容
1	31h	メニュー表示
2	32h	ブランクチェック
3	33h	消去
4	34h	書き込み
5	35h	ターゲットプログラム実行

5.3　ターゲットプログラムの実行開始位置

サンプルコードは、マイコンのリセット解除後に USB が未接続だった場合、またはコマンドによってターゲットプログラム実行を選択された場合、ターゲットプログラムを実行します。このときサンプルコードは、アドレス“FFFD FFFCh”に書かれているアドレス番地からプログラムを実行します。つまり、ターゲットプログラムにとってのリセットベクタが“FFFD FFFCh”(ターゲットリセットベクタ)になります。

ターゲットプログラムでは、あらかじめターゲットリセットベクタに開始アドレスを格納してください。

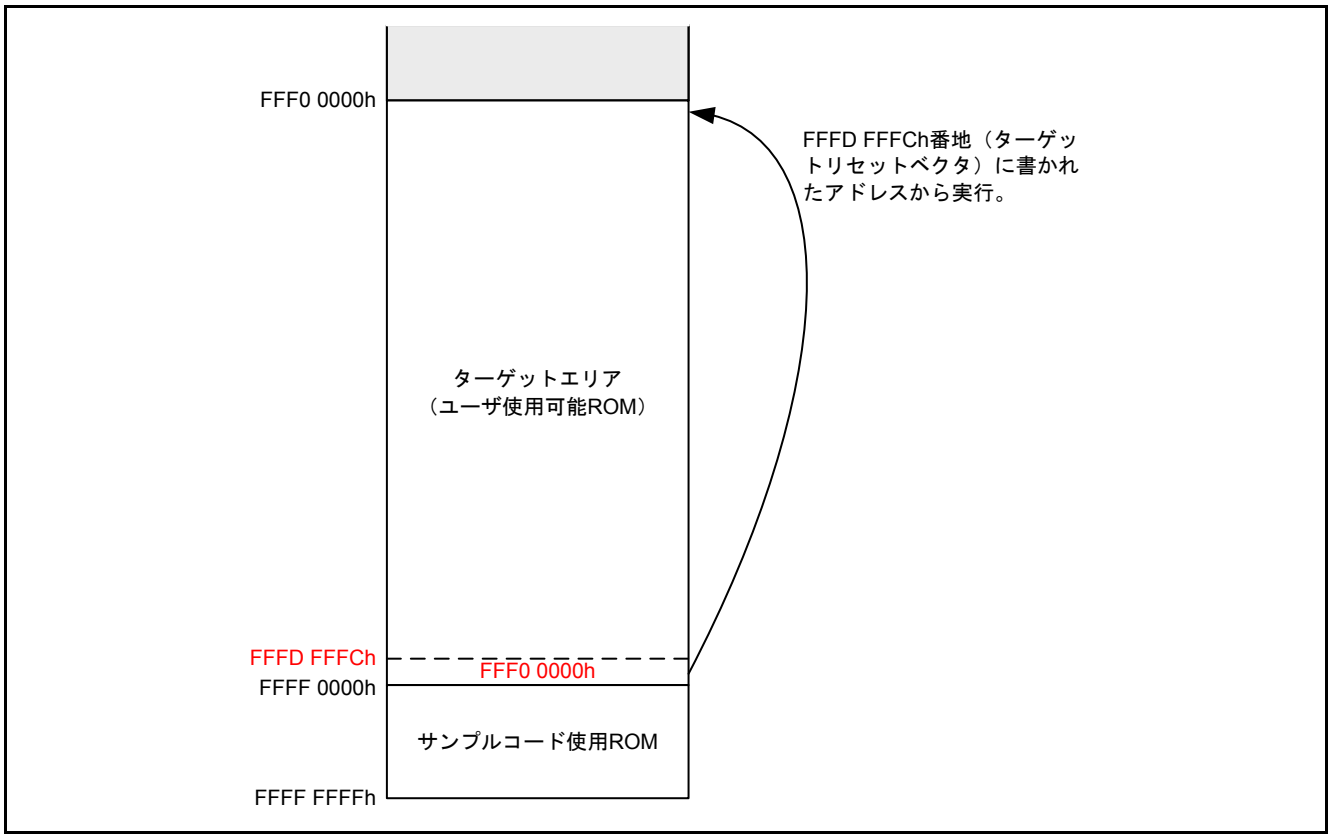

図5.4　ターゲットリセットベクタ

5.4　サンプルコードのモード

図 5.5にサンプルコードのモード遷移図を示します。

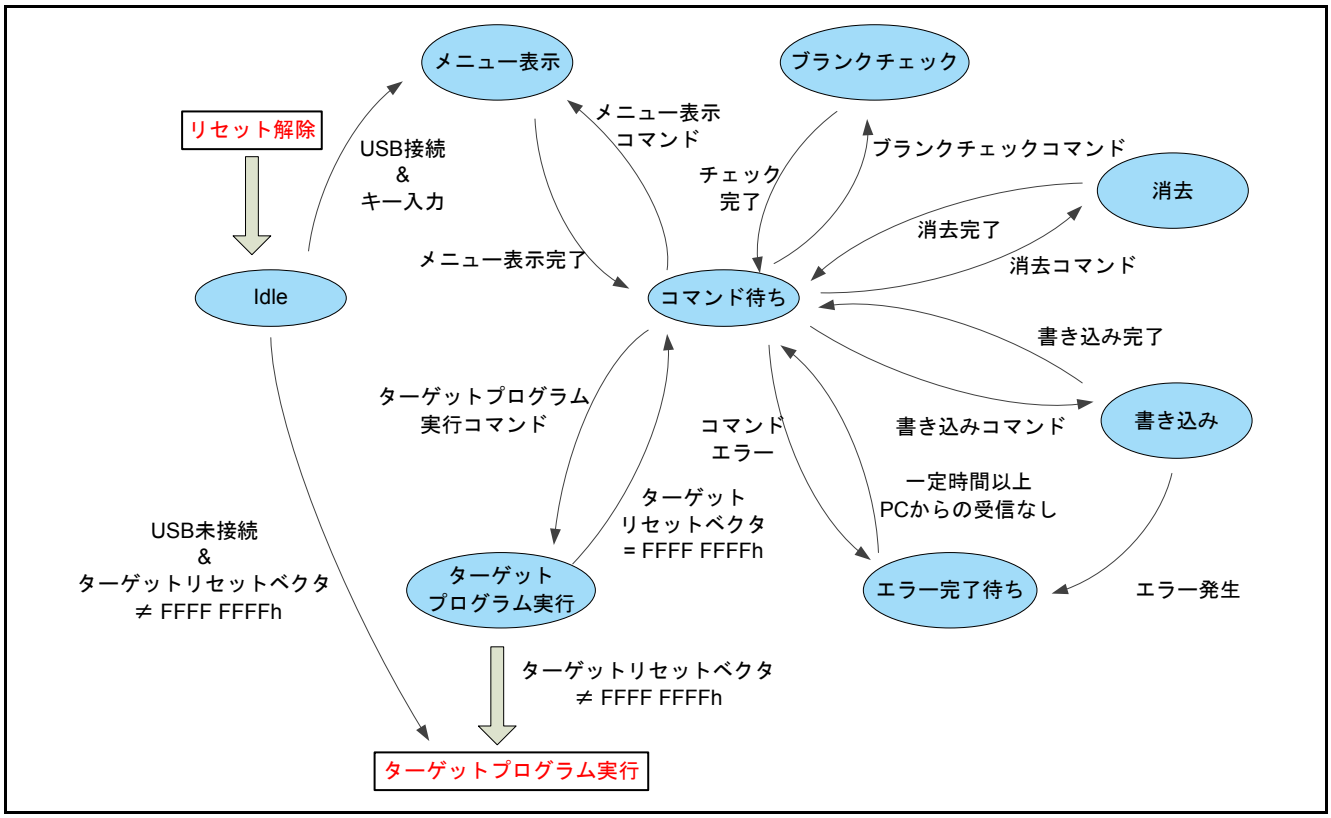

図5.5　サンプルコードのモード遷移図

5.4.1　Idle モード

リセット解除後に Idle モードに遷移します。USB が未接続だった場合、かつターゲットリセットベクタの値が FFFF FFFFh 以外だった場合にターゲットプログラムを実行します。

USB が接続されていた場合は、一定間隔で PC からのキー入力を促すメッセージを送信します。PC のキーが入力されると(任意のデータをマイコンが受信すると)メニュー表示モードに遷移します。

5.4.2　メニュー表示モード

メニューを表示します。表示が完了すると、コマンド待ちモードに遷移します。

5.4.3　コマンド待ちモード

PC からのコマンドを待ちます。何らかのコマンドを受けると、そのコマンドに従いモードを遷移します。

5.4.4　ブランクチェックモード

ターゲットエリアのブランクチェックを行います。ブランクチェックが完了すると、その結果を表示したのち、コマンド待ちモードに遷移します。

5.4.5　消去モード

シンプルフラッシュ API を使用して、ターゲットエリアを消去します。消去が完了すると、その結果を表示したのち、コマンド待ちモードに遷移します。

5.4.6　書き込みモード

シンプルフラッシュ API を使用して、ターゲットエリアへの書き込みを行うモードです。マイコンはコマンドを受けたのち、mot ファイルの受信待ちになります。PC から mot ファイルを ASCII コードで送信してください。書き込みコマンドを受けた後の動作を以下に示します。

(1) Motorola S フォーマットの最初のデータである‘S’(ASCII コード)を受信するまで待ちます。
このとき、‘S’以外のデータはすべて破棄します。

(2) ‘S’を受信したのち、Motorola S フォーマットに従ってデータを受信し、書き込みを行います。
いったん‘S’のデータを受信すると､それ以降に Motorola S フォーマットと異なるフォーマットのデータを受信した場合、エラー完了待ちモードに遷移します。また、チェックサムエラー等のエラーが発生した場合も、エラー完了待ちモードに遷移します。各エラーとその表示メッセージについては、「5.10 メッセージテーブル一覧」を参照ください。

(3) 書き込みが完了したら、その結果をメッセージで表示したのち、コマンド待ちモードに遷移します。

注 1：　Motorola S フォーマットの最後には改行コードが付加されますが、サンプルコードはチェックサムまでを Motorola S フォーマットとして扱います。したがって、改行コードは上記(1)の判定ですべて破棄されます。また、サンプルコードは書き込み完了時に、余分に受信したすべてのデータを破棄していますが、改行コードがデータ破棄よりも遅いタイミングで受信された場合、マイコンはそのデータをコマンドとして受信するため、コマンドエラーメッセージを表示する場合があります。

5.4.7　エラー完了待ちモード

「5.2 コマンド一覧」に記載されていないコマンドをマイコンが受信した場合、または書き込みモードでエラーが発生した場合に、エラー完了待ちモードに遷移します。エラー完了待ちモードでは、PC からデータを受信しない状態が一定時間続いた場合、コマンド待ちモードに遷移します。

これは、受信した mot ファイルにエラーがあった場合、マイコンは書き込みを中断しますが、ターミナルソフトからは続けてデータが送られてくる可能性があるためです。エラーが発生した直後にコマンド待ちモードに遷移してしまうと、それらのデータをコマンドとして受け取ってしまうため、一度エラー完了待ちモードに遷移し、PC からデータが送られて来なくなるのを待ちます。なお、エラー完了待ちモード中に PC から受信したデータはすべて破棄されます。

5.4.8　ターゲットプログラム実行モード

USB を停止し、ターゲットプログラムを実行します。ただし、ターゲットリセットベクタの値が FFFF FFFFh だった場合は、エラーメッセージを表示した後、コマンド待ちモードに遷移します。

5.5　書き込み時のデータの流れ

ターゲットプログラム書き込み時における、マイコン内部のデータの流れを図 5.6に示します。

受信時：

(1)　PC から受信したデータを受信用リングバッファに転送します。
(2)　Motorola S フォーマットの 1 レコードを MotS 用バッファ(ASCII)にコピーします。
(3)　Motorola S フォーマットのヘッダ部分を解析すると同時に、ASCII コードのデータを Binary データに変換し、MotS 用バッファ(Binary)に格納します。
(4)　書き込み用バッファにデータを格納します。
RX63N、RX631 グループのユーザ領域への書き込み単位は 256byte です。書き込みデータが 256byte 揃うまで(1)～(4)を繰り返します。また、書き込みデータ数の合計が 256byte を超えた場合は、超えた分のデータを一時保存しておき、次回の 256byte 書き込み時に使用します。
(5)　準備された書き込みデータ(256byte)を、シンプルフラッシュ API を使用してフラッシュメモリへ書き込みます。

送信時：

(6)　シンプルフラッシュ API からの戻り値から、書き込み結果を判定します。
(7)　書き込み結果に対応するメッセージを、送信用リングバッファに格納します。
(8)　USB Peripheral Communication Device Class Driver を使用して、PC へメッセージを送信します。

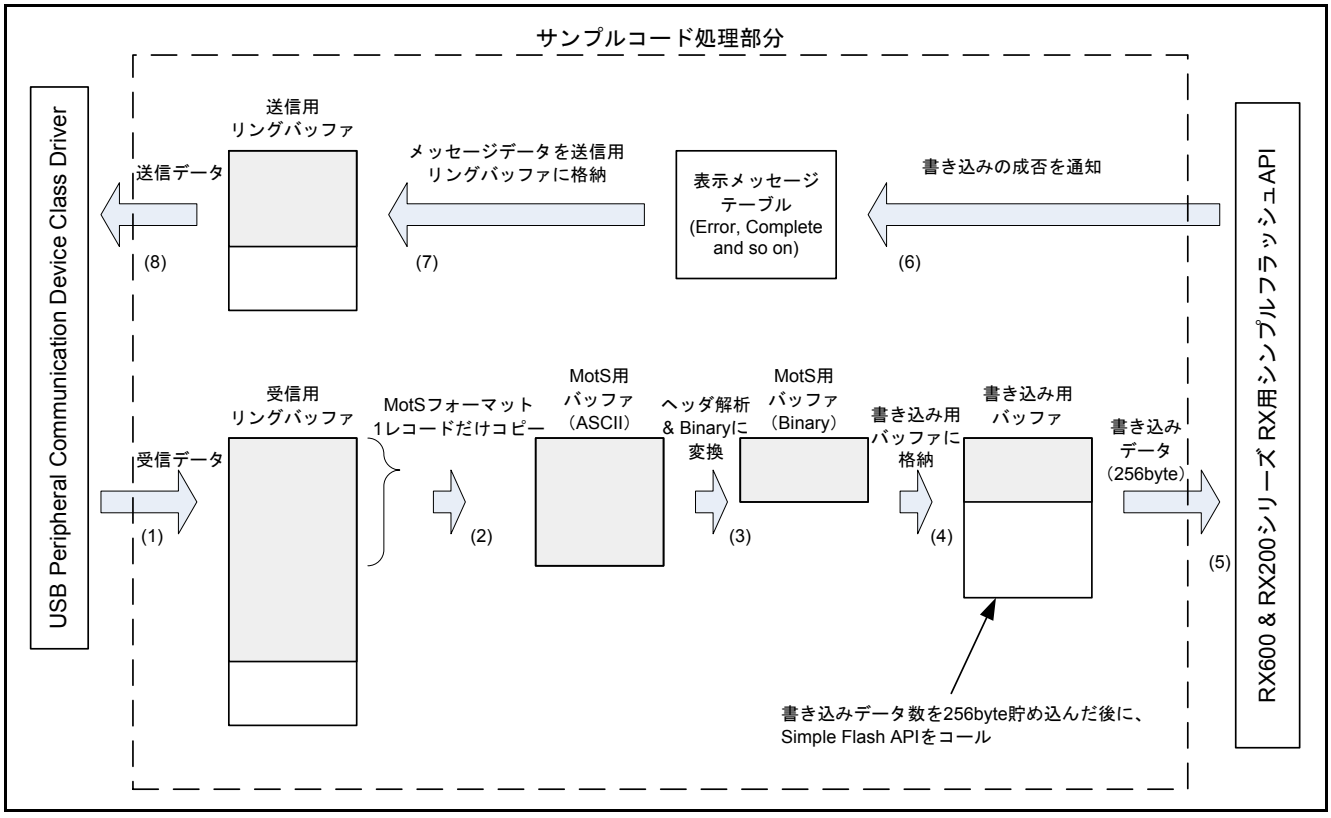

図5.6　書き込み時のデータの流れ

図 5.7に書き込み時のデータ構造を示します。

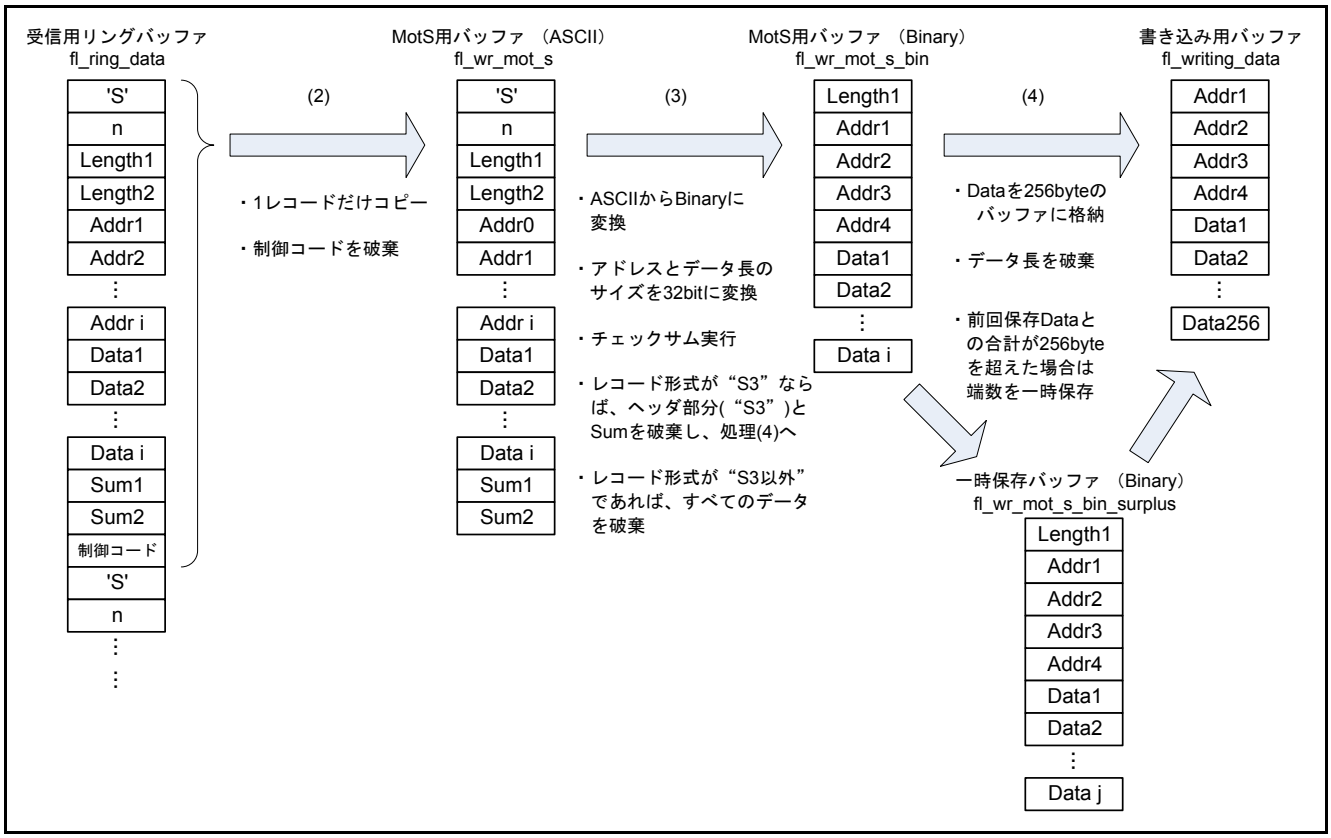

図5.7　書き込み時のデータ構造

5.6　ROM 容量の変更

サンプルコードは 1M バイトの ROM 容量の使用を前提としています。ROM 容量が 1M バイト以外のマイコンを使用する場合は、ファイル"r_Flash_main.h"内の define"FL_END_BLOCK_NUM"を、使用する容量に合わせて変更してください。

表 5.2にターゲットエリア ROM 容量一覧を示します。

表5.2　ターゲットエリア ROM 容量一覧

ROM 容量	ターゲットエリア ROM 容量	ターゲットエリア 開始アドレス	ターゲットエリア ブロック番号
2M	1920K	FFE0 0000h	EB14～EB69
1.5M	1408K	FFE8 0000h	EB14～EB61
1M	896K	FFF0 0000h	EB14～EB53
768K	640K	FFF4 0000h	EB14～EB45
512K	384K	FFF8 0000h	EB14～EB37
384K	256K	FFFA 0000h	EB14～EB29
256K	128K	FFFC 0000h	EB14～EB21

5.7　ファイル構成

表 5.3にファイル構成を示します。

表5.3　ファイル構成

ファイル名	概要	備考
r_flash_api_rx.c	RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のプログラム	詳細は RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のアプリケーションノートを参照してください。
r_flash_api_rx63n.h	RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のプログラムの外部参照用ヘッダファイル	詳細は RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のアプリケーションノートを参照してください。
r_flash_api_rx_private.h	RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のプログラムのヘッダファイル	詳細は RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のアプリケーションノートを参照してください。
r_flash_api_rx_config.h	RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のプログラムの外部参照用ヘッダファイル	詳細は RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のアプリケーションノートを参照してください。
r_flash_api_rx_if.h	RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のプログラムの外部参照用ヘッダファイル	詳細は RX600 & RX200 シリーズ RX 用シンプルフラッシュ API のアプリケーションノートを参照してください。
r_init_stop_module.c	リセット後に動作している周辺機能の停止	
r_init_stop_module.h	r_init_stop_module.c のヘッダファイル	
r_init_non_existent_port.c	存在しないポートの初期設定	
r_init_non_existent_port.h	r_init_non_existent_port.c のヘッダファイル	
r_init_clock.c	クロック初期設定	
r_init_clock.h	r_init_clock.c のヘッダファイル	
r_Flash_main.c	フラッシュ書き換えデータ処理	
r_Flash_main.h	フラッシュ書き換えデータ処理の外部参照用ヘッダファイル	
r_Flash_buff.c	USB との送受信用バッファ関連処理	
r_Flash_buff.h	USB との送受信用バッファ関連処理の外部参照用ヘッダファイル	
TrgtPrgDmmy.c	ターゲットプログラム用の領域を確保するためのダミープログラム	
r_bps フォルダ内のファイル	RX600 & RX200 シリーズ RX 用シンプルフラッシュ API で使用する r_bps パッケージのプログラム	
その他ファイル	USB Peripheral Communication Device Class Driver のプログラム	詳細はルネサス USB デバイス USB Peripheral Communication Device Class Driver および USB Basic Firmware のアプリケーションノートを参照してください。

5.8　定数一覧

表 5.4にサンプルコードで使用する定数を示します。ただし、USB Peripheral Communication Device Class Driver、シンプルフラッシュ API で使用しているものは除きます。

表5.4　サンプルコードで使用する定数

定数名	設定値	内容
FL_CMD_DATA_SHOW_INST	31h	コマンド値。ASCII“1”
FL_CMD_DATA_BLNK_CHECK	32h	コマンド値。ASCII“2”
FL_CMD_DATA_ERASE_TRGT_AREA	33h	コマンド値。ASCII“3”
FL_CMD_DATA_PRG_TRGT_AREA	34h	コマンド値。ASCII“4”
FL_CMD_DATA_RUN_TRGT_AREA	35h	コマンド値。ASCII“5”
FL_RINGBUFF_SIZE	1024	USB からのデータ受信用リングバッファサイズ
FL_RINGBUFF2_SIZE	256	USB へのデータ送信用リングバッファサイズ
FL_USB_RCV_BLANK_SIZE	64	USB が一度に送信するデータ数
FL_SEND_END_CODE	00h	メッセージテーブルの End Code
FL_MOTS_ADDR_SIZE	4	Motorola S フォーマットのアドレスのバッファサイズ
FL_MOTS_SUM_SIZE	1	Motorola S フォーマットのチェックサムのバッファサイズ
FL_START_BLOCK_NUM	14	ターゲットエリアの最初のブロック
FL_END_BLOCK_NUM	53	ターゲットエリアの最後のブロック
FL_TARGET_REST_VECT_ADDR	FFFD FFFCh	ターゲットリセットベクタ
FL_USB_UNCONNECT_WAIT_PERIOD	10000h	USB 未接続時のターゲットプログラム実行までの wait 時間
FL_IDLE_MESSAGE_OUTPUT_PERIOD	10000h	アイドル時のメッセージ表示間隔の時間
FL_ERROR_WAIT_PERIOD	10000h	エラー完了待ち時間(このカウンタ値の間、USB から何も受信しなければ、エラー完了待ちを終了)

5.9　構造体/共用体一覧

図 5.8にサンプルコードで使用する構造体/共用体を示します。ただし、USB Peripheral Communication Device Class Driver、シンプルフラッシュ API で使用しているものは除きます。

```
/* buffer for mot S format data */
typedef struct {
    uint8_t type[2]; /* "S0", "S1" and so on */
    uint8_t len[2]; /* "0-255" */
    uint8_t addr_data_sum[512];
} Fl_prg_mot_s_t;

/* buffer for write data
    (this data is the converted data from mot S format data) */
typedef struct {
    uint8_t len;
    uint32_t addr;
    uint8_t data[256];
} Fl_prg_mot_s_binary_t;

/* buffer for writing flash */
typedef struct {
    uint32_t addr;
    uint8_t data[256];
} Fl_prg_writing_data_t;
```

図5.8　サンプルコードで使用する構造体/共用体

5.10　メッセージテーブルー覧

表 5.5にサンプルコードで使用するメッセージを示します。なお、各メッセージに含まれている改行コードは省略します。

表5.5　サンプルコードで使用するメッセージ

メッセージ	内容
Press any key.	アイドル時に一定間隔で表示されます
Renesas USB Flash Sample Press:- 1 - Show instructions(these). 2 - Blank check. 3 - Erase target area. 4 - Start programming. 5 - Run target program.	メニュー
Target area is blank.	ブランクチェック後、ターゲットエリアがブランクだった場合に表示されます
Target area is NOT blank.	ブランクチェック後、ターゲットエリアがブランクでなかった場合に表示されます
Erase target area...	消去中に表示されます
Erase complete.	消去が完了すると表示されます
ERROR!!! - Erase is failed.	消去が失敗すると表示されます
Please send a mot file.	書き込みを開始すると表示されますこのメッセージが表示された後、mot ファイルを送信してください
Program complete.	書き込みが完了すると表示されます
Program failed.	書き込みが失敗すると表示されます
ERROR!!! - Verify error.	ベリファイエラーが発生すると表示されます
Run target program.	ターゲットプログラムを実行すると表示されます
ERROR!!! - Target reset vector(0xFFFDFFFC) is 0xFFFFFFFF.	ターゲットプログラム実行しようとした時に、ターゲットベクタが FFFF FFFFh だった場合に表示されます
ERROR!!! - Command error. Please press a number from 1 to 5.	正しくないコマンドが入力されたときに表示されます
ERROR!!! - Mot file format is NOT correct.	正しくないターゲットプログラムが送信されたときに表示されます
ERROR!!! - Check sum error.	ターゲットプログラム(Motorola S フォーマット)のチェックサムでエラーが発生したときに表示されます
Please wait for instructions to be shown.	エラーが発生し、かつ続けてデータが送られてくる状態のときに表示されます。一定時間データが送信されてこなければ、コマンド待ち状態に遷移します。エラー発生後も連続してデータが送られてくる場合、ピリオドを一定間隔で表示します
ピリオド "."	エラー発生時に一定間隔で表示されます
改行コード"¥r¥n"	各メッセージの改行に使用します

5.11　関数一覧

表 5.6にサンプルコードで使用する関数を示します。ただし、USB Peripheral Communication Device Class Driver、シンプルフラッシュ API で使用しているものは除きます。

表5.6　サンプルコードで使用する関数

関数名	概要
R_INIT_StopModule	リセット後に動作している周辺機能の停止
R_INIT_NonExistentPort	存在しないポートの初期設定
R_INIT_Clock	クロック初期設定
R_FI_Rewrite_process	書き換え処理のメインとなる関数
R_FI_Idle	リセット解除後、キーを押されるまでの処理
R_FI_AnalyzeCMD	コマンド解析
R_FI_EraseTrgtArea	消去
R_FI_PrgTrgtArea	書き込み
R_FI_RunTrgtPrg	ターゲットプログラム実行
R_FI_ErrorWait	エラー完了待ち
R_FI_cmd_ShowInst	メニュー表示
R_FI_cmd_BlankCheckStart	ブランクチェック
R_FI_cmd_EraseStart	消去開始処理
R_FI_cmd_PrgStart	書き込み開始処理
R_FI_cmd_RunTrgtPrgStart	実行開始処理
R_FI_Blnk_BlankCheck	ブランクチェック(シンプルフラッシュ API の例に従った処理)
R_FI_Ers_EraseFlash	消去(シンプルフラッシュ API の例に従った処理)
R_FI_Prg_StoreMotS	Motorola S フォーマットデータ格納
R_FI_Prg_ProcessForMotS_data	Motorola S フォーマット解析および Binary データへ変換
R_FI_Prg_MotS_AsciiToBinary	Motorola S フォーマットデータ ASCII - Binary 変換
R_FI_Prg_MakeWriteData	書き込みデータ作成
R_FI_Prg_WriteData	ターゲットプログラム書き込み (シンプルフラッシュ API の例に従った処理)
R_FI_Prg_ClearMotSVariables	Motorola S フォーマットデータ関連の変数クリア
R_FI_Run_StopUSB	USB 停止
R_FI_RcvDataString	USB 受信データ格納
R_FI_SetSendData	USB 送信データセット
R_FI_SetDisplayMsgData	表示用メッセージデータセット
R_FI_RingCheckBlank	USB 受信データを格納するためのバッファのブランクチェック
R_FI_Ring2CheckData	USB 送信データの格納するためのバッファのブランクチェック
R_FI_USB_NonConnect_Run	USB 未接続時のターゲットプログラム実行処理
R_FI_RingEnQueue	USB 受信データ格納バッファへの書き込み
R_FI_RingDeQueue	USB 受信データ格納バッファからの読み出し
R_FI_RingClear	USB 受信データ格納バッファクリア
R_FI_RingCheck	USB 受信データ格納バッファデータ数チェック
R_FI_Ring2EnQueue	USB 送信データ格納バッファへの書き込み
R_FI_Ring2DeQueue	USB 送信データ格納バッファからの読み出し
R_FI_Ring2Clear	USB 送信データ格納バッファクリア
R_FI_Ring2Check	USB 送信データ格納バッファデータ数チェック
R_FI_AsciiToHexByte	ASCII コードを Binary データへ変換

5.12　関数仕様

サンプルコードの関数仕様を示します。

R_INIT_StopModule	
概　要	リセット後に動作している周辺機能の停止
ヘッダ	r_init_stop_module.h
宣　言	void R_INIT_StopModule(void)
説　明	モジュールストップ状態へ遷移する設定を行います。
引　数	なし
リターン値	なし
備　考	サンプルコードでは、モジュールストップ状態への遷移は行っていません。 本関数の詳細は、アプリケーションノート「RX63N グループ、RX631 初期設定例 Rev.1.10」を参照してください。

R_INIT_NonExistentPort	
概　要	存在しないポートの初期設定
ヘッダ	r_init_non_existent_port.h
宣　言	void R_INIT_NonExistentPort(void)
説　明	176 ピン未満の製品に対して、存在しないポートの端子に対応するポート方向レジスタの初期設定を行います。
引　数	なし
リターン値	なし
備　考	サンプルコードでは、176 ピン版(PIN_SIZE=176)に設定しています。 本関数をコールした後に、存在しないポートを含む PDR、PODR レジスタへバイト単位で書き込む場合、存在しないポートの方向制御ビットには“1”、ポート出力データ格納ビットには“0”を設定してください。 本関数の詳細は、アプリケーションノート「RX63N グループ、RX631 初期設定例 Rev.1.10」を参照してください。

R_INIT_Clock	
概　要	クロック初期設定
ヘッダ	r_init_clock.h
宣　言	void R_INIT_Clock(void)
説　明	クロックの初期設定を行います。
引　数	なし
リターン値	なし
備　考	サンプルコードでは、システムクロックを PLL とし、サブクロックを使用しない処理を選択しています。 また、以下の設定を変更して使用しています。 ①　BCLK の分周比を 4 分周から 8 分周に変更しています。 ②　USB クロックを未使用から 4 分周に変更しています。 ③　BCLK 端子出力を分周なしから 2 分周に変更しています。 本関数の詳細は、アプリケーションノート「RX63N グループ、RX631グループ初期設定例 Rev.1.10」を参照してください。

R_FI_Rewrite_process	
概　要	書き込み処理メイン
ヘッダ	r_Flash_main.h
宣　言	void R_FI_Rewrite_process(void)
説　明	・書き込み処理のモードによって、各モードの処理関数を呼び出します。
引　数	なし
リターン値	なし

R_FI_Idle	
概　要	Idle モード処理
ヘッダ	なし
宣　言	static void R_FI_Idle(void)
説　明	・USB が何らかのデータを受信するまで(PC からのキー入力があるまで)、一定間隔で"Press any key"というメッセージを表示します。
引　数	なし
リターン値	なし

R_FI_AnalyzeCMD	
概　要	コマンド解析
ヘッダ	なし
宣　言	static FI_SMPL_command_t R_FI_AnalyzeCMD(void)
説　明	・ 受信用リングバッファにデータがあった場合、そのデータの 1byte 目をコマンドとして解析します。 ・ 1byte 目以外のすべての受信データを破棄します。
引　数	なし
リターン値	・ データ未受信：FL_CMD_NONE ・ メニュー表示コマンド受信：FL_CMD_SHOW_INST ・ ブランクチェックコマンド受信：FL_CMD_BLNK_CHECK ・ 消去コマンド受信：FL_CMD_ERASE_TRGT_AREA ・ 書き込みコマンド受信：FL_CMD_PRG_TRGT_AREA ・ ターゲットエリア実行コマンド受信：FL_CMD_RUN_TRGT_AREA ・ 上記以外受信：FL_CMD_ERROR
備　考	

R_FI_EraseTrgtArea	
概　要	消去モード処理
ヘッダ	r_Flash_main.h
宣　言	static void R_FI_EraseTrgtArea(void)
説　明	・ ターゲットエリアを消去する関数を呼び出します。 ・ 消去結果をメッセージ表示します。
引　数	なし
リターン値	なし
備　考	

R_FI_PrgTrgtArea	
概　要	書き込みモード処理
ヘッダ	なし
宣　言	static void R_FI_PrgTrgtArea(void)
説　明	・ 受信用リングバッファにデータがあった場合、Motorola S フォーマットとして格納する関数を呼び出します。 ・ Motorola S フォーマットを一つ分受信したら、ヘッダ解析、バイナリデータへの変換を行う関数を呼び出します。 ・ バイナリデータへの変換が完了した場合、そのデータを書き込み用バッファに格納する関数を呼び出します。
引　数	なし
リターン値	なし
備　考	

R_FI_RunTrgtPrg	
概　要	実行モード処理
ヘッダ	なし
宣　言	static void R_FI_RunTrgtPrg(void)
説　明	・ ターゲットリセットベクタが FFFF FFFFh 以外だった場合、USB を停止させた後、ターゲットプログラムを実行します。 ・ ターゲットリセットベクタが FFFF FFFFh だった場合、ターゲットベクタエラーのメッセージを表示します。
引　数	なし
リターン値	なし
備　考	

R_FI_ErrorWait	
概　要	エラー完了待ちモード処理
ヘッダ	なし
宣　言	static void R_FI_ErrorWait(void)
説　明	・ 受信用リングバッファが空の状態で、一定時間以上経過した場合、コマンド待ちモードに遷移します。 ・ 受信用リングバッファに何らかのデータがあった場合、そのデータを破棄した後、再度一定時間の経過を開始します。
引　数	なし
リターン値	なし
備　考	

R_FI_cmd_ShowInst	
概　要	メニュー表示
ヘッダ	なし
宣　言	static void R_FI_cmd_ShowInst(void)
説　明	・ メニューを表示します。
引　数	なし
リターン値	なし
備　考	

R_FI_cmd_BlankCheckStart	
概　要	ブランクチェック開始
ヘッダ	なし
宣　言	static void R_FI_cmd_BlankCheckStart(void)
説　明	・ ターゲットエリアのブランクチェックを行う関数をコールします。 ・ ブランクチェックの結果をメッセージで表示します。
引　数	なし
リターン値	なし
備　考	

R_FI_cmd_EraseStart	
概　要	ターゲットエリア消去開始
ヘッダ	なし
宣　言	static void R_FI_cmd_EraseStart(void)
説　明	・ ターゲットエリアの消去を開始します。 ・ 消去開始のメッセージを表示します。
引　数	なし
リターン値	なし
備　考	

R_FI_cmd_PrgStart	
概　要	ターゲットエリア書き込み開始
ヘッダ	なし
宣　言	static void R_FI_cmd_PrgStart(void)
説　明	・ ターゲットエリアへの書き込みを開始します。 ・ 書き込み開始メッセージを表示します。
引　数	なし
リターン値	なし
備　考	

R_Fl_cmd_RunTrgtPrgStart	
概　要	ターゲットプログラム実行開始
ヘッダ	なし
宣　言	static void R_Fl_cmd_RunTrgtPrgStart(void)
説　明	・ ターゲットベクタが FFFF FFFFh だった場合､ターゲットベクタエラーのメッセージを送信します。 ・ ターゲットベクタが FFFF FFFFh でなかった場合、ターゲットプログラム実行のメッセージを表示した後、ターゲットプログラム実行モードに遷移します。
引　数	なし
リターン値	なし
備　考	

R_Fl_Blnk_BlankCheck	
概　要	ブランクチェック
ヘッダ	なし
宣　言	Fl_API_SMPL_rtn_t R_Fl_Blnk_BlankCheck(void)
説　明	・ ターゲットエリアのブランクチェックを行います。
引　数	なし
リターン値	・ ターゲットエリアがブランクの場合：FLASH_API_SAMPLE_OK ・ ターゲットエリアがブランクでない場合：FLASH_API_SAMPLE_NG
備　考	

R_Fl_Ers_EraseFlash	
概　要	ターゲットエリア消去
ヘッダ	なし
宣　言	Fl_API_SMPL_rtn_t R_Fl_Ers_EraseFlash(void)
説　明	・ ターゲットエリアを消去します。
引　数	なし
リターン値	・ 消去が正常に完了した場合：FLASH_API_SAMPLE_OK ・ 消去が正常に完了しなかった場合：FLASH_API_SAMPLE_NG
備　考	・ 消去中の割り込みによる ROM アクセスを防ぐため、プロセッサステータスワード(PSW)のプロセッサ割り込み優先レベル(IPL)を変更します。

R_FI_Prg_StoreMotS	
概　要	Motorola S フォーマットデータ格納
ヘッダ	なし
宣　言	static FI_API_SMPL_rtn_t R_FI_Prg_StoreMotS(uint8_t)
説　明	・ 引数で受け取ったデータを Motorola S フォーマットデータとして 1byte ずつ格納します。 ・ 最初に'S'(ASCII コード)を受け取るまでは、すべてのデータを破棄します。
引　数	・ 第一引数：mot_data　　：Motorola S フォーマットデータ
リターン値	・ 一つ分の Motorola S フォーマットデータ('S'からチェックサムまで)を格納した場合：FLASH_API_SAMPLE_OK ・ 一つ分の Motorola S フォーマットデータを格納していない場合：FLASH_API_SAMPLE_NG
備　考	・ 本関数は、Motorola S フォーマットデータを 1byte ずつ繰り返し引数で渡して使用します。 ・ チェックサムの確認は行いません。

R_FI_Prg_ProcessForMotS_data	
概　要	Motorola S フォーマットヘッダ解析、および Binary 変換
ヘッダ	なし
宣　言	static void R_FI_Prg_ProcessForMotS_data(void)
説　明	・ Motorola S フォーマットのヘッダ解析し、Binary 変換する関数をコールします。 ・ Motorola S フォーマットと異なるデータがあった場合、エラーメッセージを表示します。
引　数	なし
リターン値	なし
備　考	

R_FI_Prg_MotS_AsciiToBinary	
概　要	Motorola S フォーマットデータ ASCII - Binary 変換
ヘッダ	なし
宣　言	static FI_API_SMPL_rtn_t R_FI_Prg_MotS_AsciiToBinary (FI_prg_mot_s_t *, FI_prg_mot_s_binary_t *)
説　明	・ ASCII コードの Motorola S フォーマットのデータを Binary データに変換します。 ・ 変換した Binary データのチェックサムを確認します。 ・ Motorola S フォーマットと異なるデータがあった場合、エラーメッセージを表示します。 ・ チェックサムエラーが発生した場合、エラーメッセージを表示します。
引　数	・ 第一引数：*tmp_mot_s　　：ASCII コードの Motorola S フォーマットのデータのポインタ ・ 第二引数：*tmp_mot_s_binary　　：Binary 変換されたデータを格納する変数のポインタ
リターン値	・ 正常に変換が完了した場合：FLASH_API_SAMPLE_OK ・ 正常に変換が完了しなかった場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_Prg_MakeWriteData	
概　要	ターゲットエリアへの書き込みデータ作成
ヘッダ	なし
宣　言	static FI_API_SMPL_rtn_t R_FI_Prg_MakeWriteData(void)
説　明	・ 256byte ずつに区切られたデータを作成します。
引　数	なし
リターン値	・ 256byte の書き込みデータ作成が完了した場合：FLASH_API_SAMPLE_OK ・ 256byte の書き込みデータ作成が未完了の場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_Prg_WriteData	
概　要	ターゲットエリアへの書き込み
ヘッダ	なし
宣　言	static FI_API_SMPL_rtn_t R_FI_Prg_WriteData(void)
説　明	・ ターゲットエリアに書き込みを行います。 ・ 書き込んだデータのベリファイを行います。 ・ 書き込みに失敗した場合、エラーメッセージを表示します。 ・ ベリファイエラーが発生した場合、エラーメッセージを表示します。
引　数	なし
リターン値	・ 書き込みが正常に完了した場合：FLASH_API_SAMPLE_OK ・ 書き込みが正常に完了しなかった場合：FLASH_API_SAMPLE_NG
備　考	・ 書き込み中の割り込みによる ROM アクセスを防ぐため、プロセッサステータスワード(PSW)のプロセッサ割り込み優先レベル(IPL)を変更します。

R_FI_Prg_ClearMotSVariables	
概　要	Motorola S フォーマット用変数クリア
ヘッダ	なし
宣　言	static void R_FI_Prg_ClearMotSVariables(void)
説　明	・ Motorola S フォーマット用変数をクリアします。
引　数	なし
リターン値	なし
備　考	

R_FI_Run_StopUSB	
概　要	USB 停止
ヘッダ	iodefine.h、r_usb_usrconfig.h
宣　言	static void R_FI_Run_StopUSB(void)
説　明	・ USB を停止します。
引　数	なし
リターン値	なし
備　考	

R_Fl_RcvDataString

概　要	USB 受信データ格納
ヘッダ	R_Flash_main.h
宣　言	Fl_API_SMPL_rtn_t R_Fl_RcvDataString(void *, uint16_t)
説　明	・USB が受信したデータを受信用リングバッファに格納します。
引　数	・第一引数：*tranadr　：USB が受信したデータが格納されているバッファのポインタ ・第二引数：length　：USB が受信したデータ数
リターン値	・格納が完了した場合：FLASH_API_SAMPLE_OK ・受信用リングバッファがいっぱいだった場合：FLASH_API_SAMPLE_NG
備　考	

R_Fl_SetSendData

概　要	USB 送信データ格納
ヘッダ	R_Flash_main.h
宣　言	uint16_t R_Fl_SetSendData(void *, uint16_t)
説　明	・USB が送信するデータを送信用バッファに格納します。
引　数	・第一引数：*tranadr　：送信用バッファのポインタ ・第二引数：length_lim　：送信用バッファのデータ格納限界数
リターン値	・送信用バッファに格納されたデータ数
備　考	

R_Fl_SetDisplayMsgData

概　要	メッセージ表示
ヘッダ	なし
宣　言	static Fl_API_SMPL_rtn_t R_Fl_SetDisplayMsgData(Fl_disp_tbl_num_t)
説　明	・指定されたメッセージを送信用リングバッファに格納します。
引　数	・第一引数：table_num　：表示するメッセージの番号
リターン値	・格納が完了した場合：FLASH_API_SAMPLE_OK ・送信用リングバッファがいっぱいだった場合：FLASH_API_SAMPLE_NG
備　考	

R_Fl_RingCheckBlank

概　要	受信用リングバッファの空き容量確認
ヘッダ	R_Flash_main.h
宣　言	Fl_API_SMPL_rtn_t R_Fl_RingCheckBlank(void)
説　明	・受信用リングバッファに USB の受信 1 回分(64byte)の空きがあるかを確認します。
引　数	なし
リターン値	・空きがあった場合：FLASH_API_SAMPLE_OK ・空きがなかった場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_Ring2CheckData	
概　要	送信用リングバッファのデータ確認
ヘッダ	R_Flash_main.h
宣　言	FI_API_SMPL_rtn_t R_FI_Ring2CheckData(void)
説　明	・送信用データバッファにデータがあるかを確認します。
引　数	なし
リターン値	・データがあった場合：FLASH_API_SAMPLE_OK ・データがなかった場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_USB_NonConnect_Run	
概　要	USB 未接続時のターゲットプログラム実行
ヘッダ	R_Flash_main.h
宣　言	void R_FI_USB_NonConnect_Run(void)
説　明	・USB を停止し、ターゲットプログラムを実行します。
引　数	なし
リターン値	なし
備　考	本関数は USB が未接続の場合に呼び出されます。

R_FI_RingEnQueue	
概　要	受信用リングバッファへのデータ格納
ヘッダ	r_Flash_buff.h
宣　言	FI_API_SMPL_rtn_t R_FI_RingEnQueue(uint8_t)
説　明	・受信用リングバッファへデータを格納します。
引　数	・第一引数：enq_data　　：格納データ
リターン値	・格納完了の場合：FLASH_API_SAMPLE_OK ・バッファフルの場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_RingDeQueue	
概　要	受信用リングバッファからのデータ読み出し
ヘッダ	r_Flash_buff.h
宣　言	FI_API_SMPL_rtn_t R_FI_RingDeQueue(uint8_t *)
説　明	・受信用リングバッファからデータを読み出します。
引　数	・第一引数：*deq_data　　：読み出したデータを格納するバッファのポインタ
リターン値	・正常にデータを読み出した場合：FLASH_API_SAMPLE_OK ・読み出すデータがなかった場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_RingClear	
概　要	受信用リングバッファクリア
ヘッダ	r_Flash_buff.h
宣　言	FI_API_SMPL_rtn_t R_FI_RingClear(void)
説　明	・ 受信用リングバッファをクリアします。
引　数	なし
リターン値	・ 常に FLASH_API_SAMPLE_OK を返します。
備　考	

R_FI_RingCheck	
概　要	受信用リングバッファのデータ数確認
ヘッダ	r_Flash_buff.h
宣　言	uint32_t R_FI_RingCheck(void)
説　明	・ 受信用リングバッファのデータ数を確認します。
引　数	なし
リターン値	・ 受信データ数を返します。
備　考	

R_FI_Ring2EnQueue	
概　要	送信用リングバッファへのデータ格納
ヘッダ	r_Flash_buff.h
宣　言	FI_API_SMPL_rtn_t R_FI_Ring2EnQueue(uint8_t)
説　明	・ 送信用リングバッファへデータを格納します。
引　数	・ 第一引数：enq_data　　：格納データ
リターン値	・ 格納完了の場合：FLASH_API_SAMPLE_OK ・ バッファフルの場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_Ring2DeQueue	
概　要	送信用リングバッファからのデータ読み出し
ヘッダ	r_Flash_buff.h
宣　言	FI_API_SMPL_rtn_t R_FI_Ring2DeQueue(uint8_t *)
説　明	・送信用リングバッファからデータを読み出します。
引　数	・第一引数：*deq_data　　：読み出したデータを格納するバッファのポインタ
リターン値	・正常にデータを読み出した場合：FLASH_API_SAMPLE_OK ・ 読み出すデータがなかった場合：FLASH_API_SAMPLE_NG
備　考	

R_FI_Ring2Clear	
概　要	送信用リングバッファクリア
ヘッダ	r_Flash_buff.h
宣　言	FI_API_SMPL_rtn_t R_FI_Ring2Clear(void)
説　明	・送信用リングバッファをクリアします。
引　数	なし
リターン値	・常に FLASH_API_SAMPLE_OK を返します。
備　考	

R_FI_Ring2Check	
概　要	受信用リングバッファのデータ数確認
ヘッダ	r_Flash_buff.h
宣　言	uint32_t R_FI_Ring2Check(void)
説　明	・送信用リングバッファのデータ数を確認します。
引　数	なし
リターン値	・送信データ数を返します。
備　考	

R_FI_AsciiToHexByte	
概　要	ASCII コードから Binary データへの変換
ヘッダ	r_Flash_buff.h
宣　言	uint8_t R_FI_AsciiToHexByte(uint8_t, uint8_t)
説　明	・2byte の ASCII コードデータを 1byte の Binary データへ変換します。
引　数	・第一引数：in_upper　：ASCII コードデータ(上位) ・第二引数：in_lower　：ASCII コードデータ(下位)
リターン値	・Binary に変換されたデータを返します。
備　考	

5.13　フローチャート

5.13.1　USB メイン処理

図 5.9、図 5.10にUSB メイン処理のフローチャートを示します。

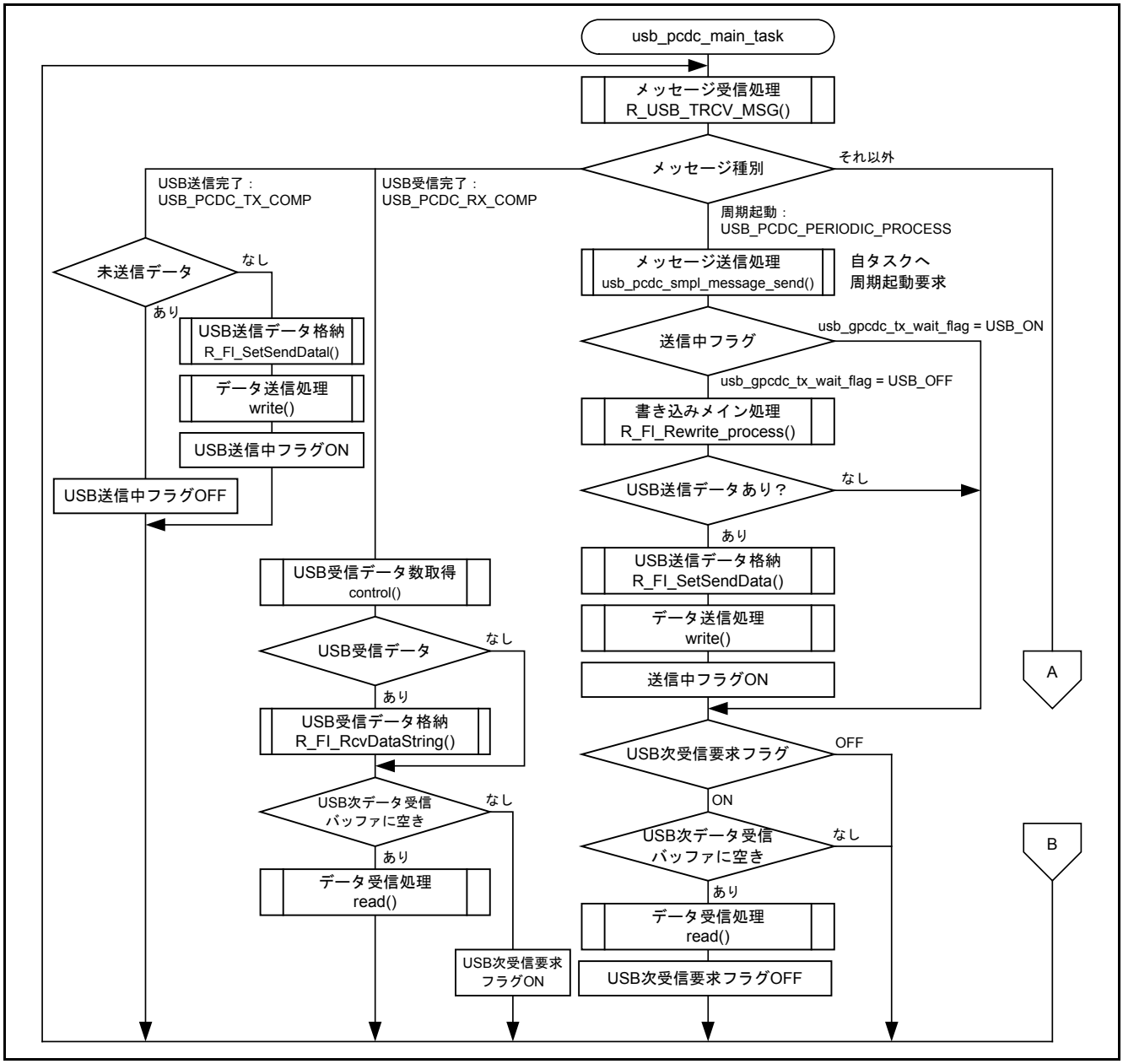

図5.9　USB メイン処理(1/2)

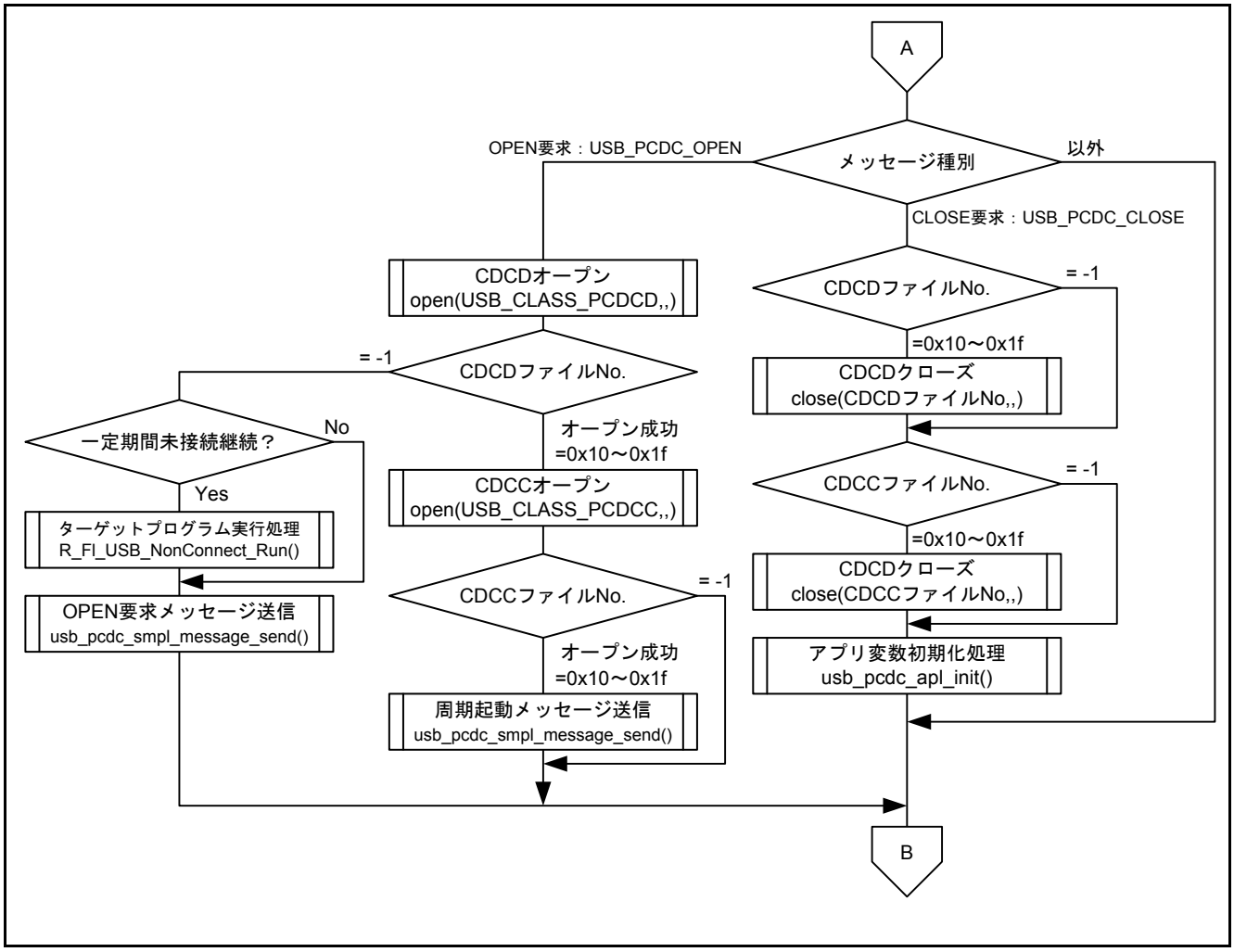

図5.10　USB メイン処理(2/2)

5.13.2　書き込みメイン処理

図 5.11に書き込みメイン処理のフローチャートを示します。

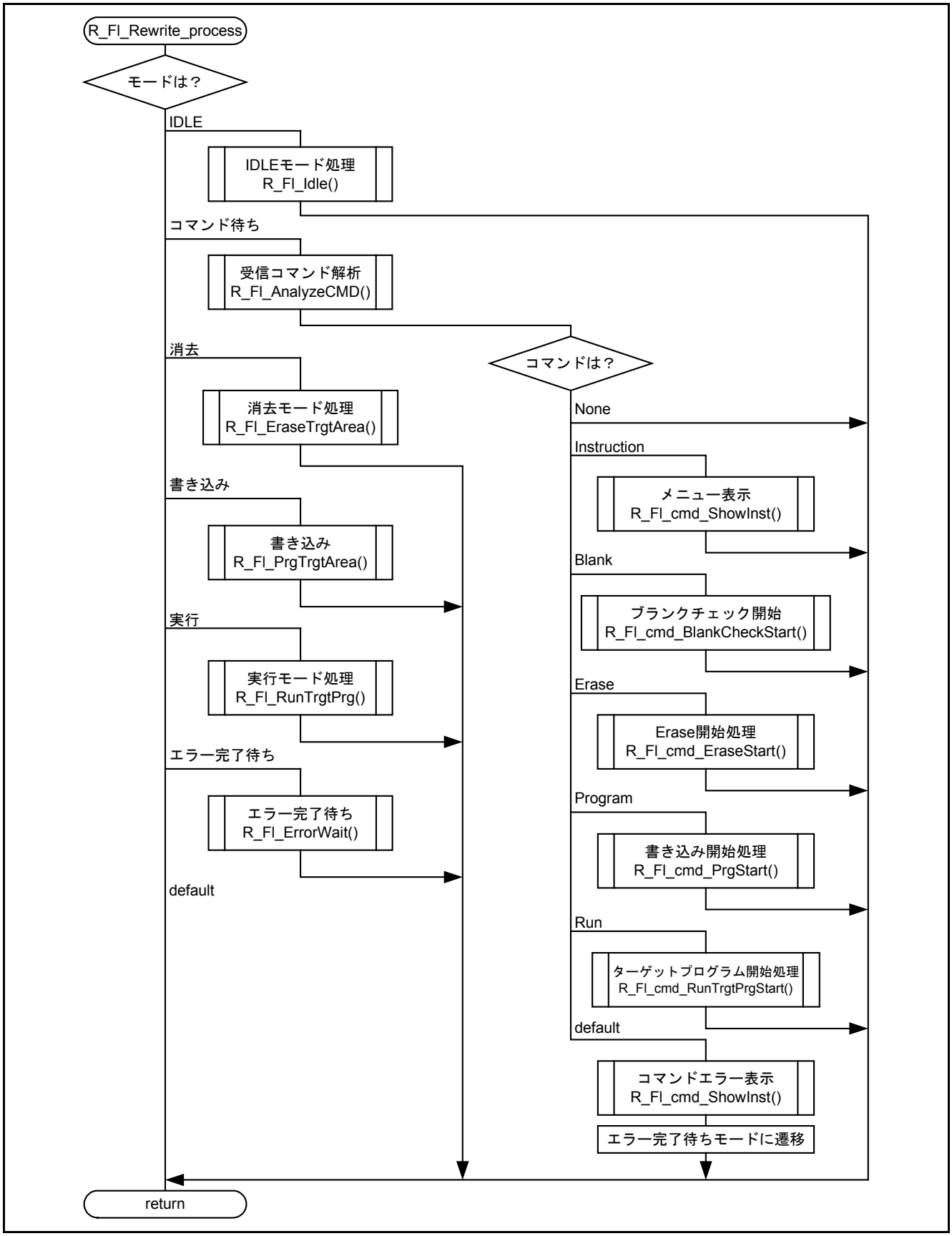

図5.11　書き込みメイン処理

5.13.3　Idle モード処理

図 5.12にIdle モード処理のフローチャートを示します。

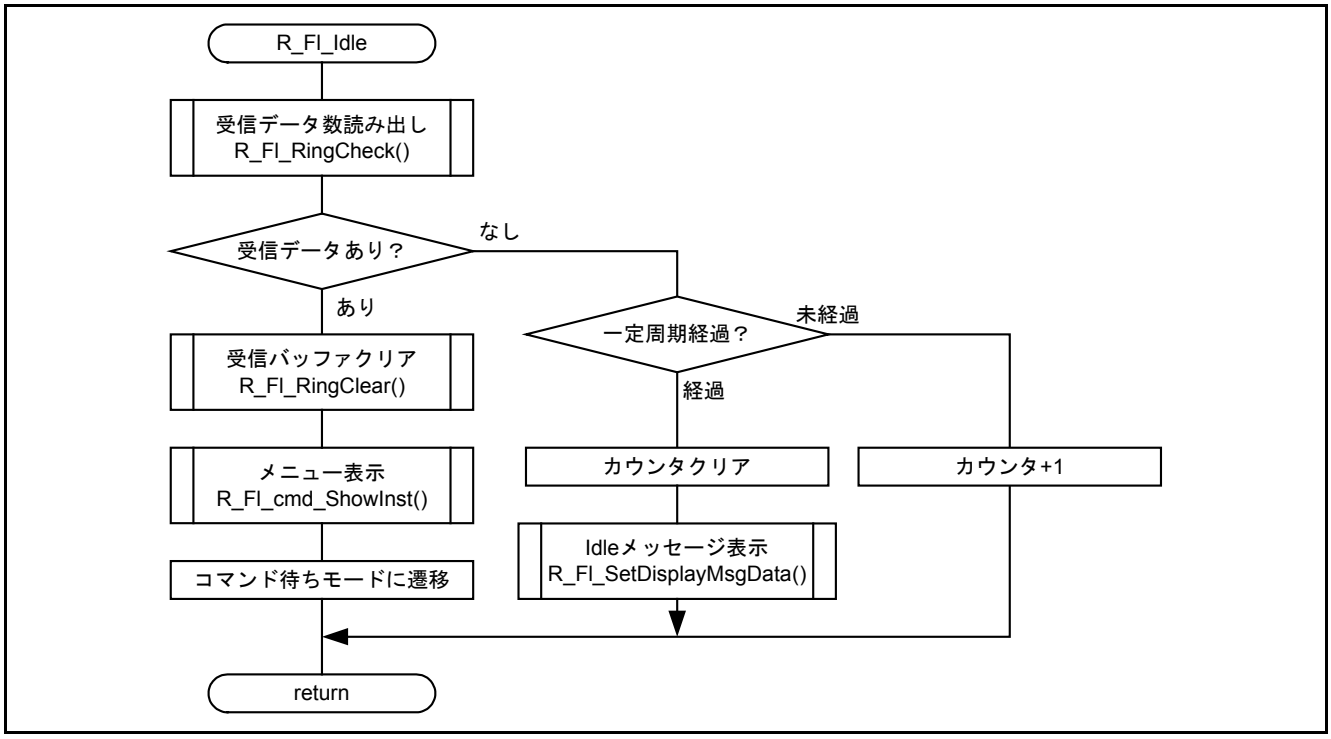

図5.12　Idle モード処理

5.13.4　コマンド解析処理

図 5.13にコマンド解析処理のフローチャートを示します。

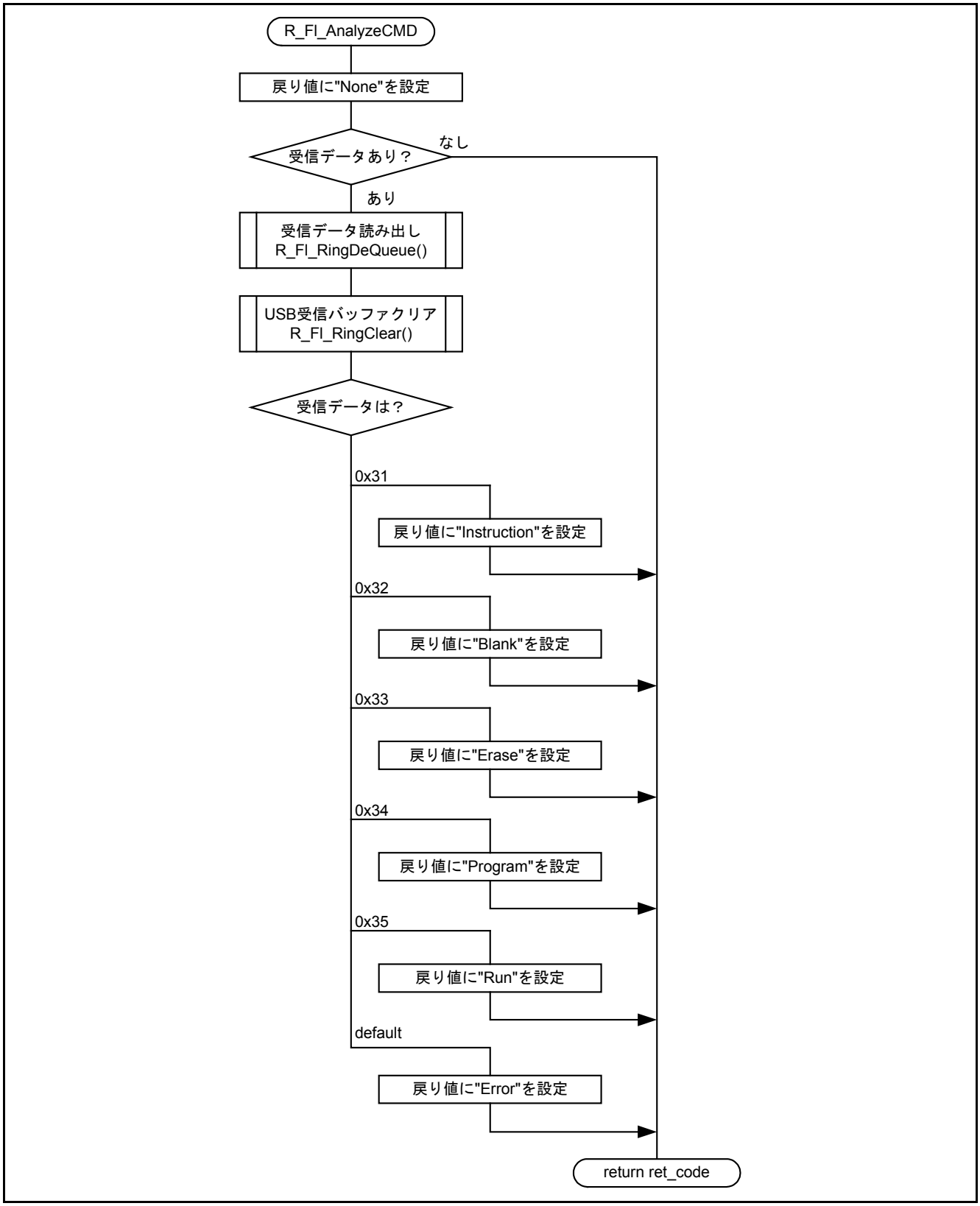

図5.13　コマンド解析処理

5.13.5　消去モード処理

図 5.14に消去モード処理のフローチャートを示します。

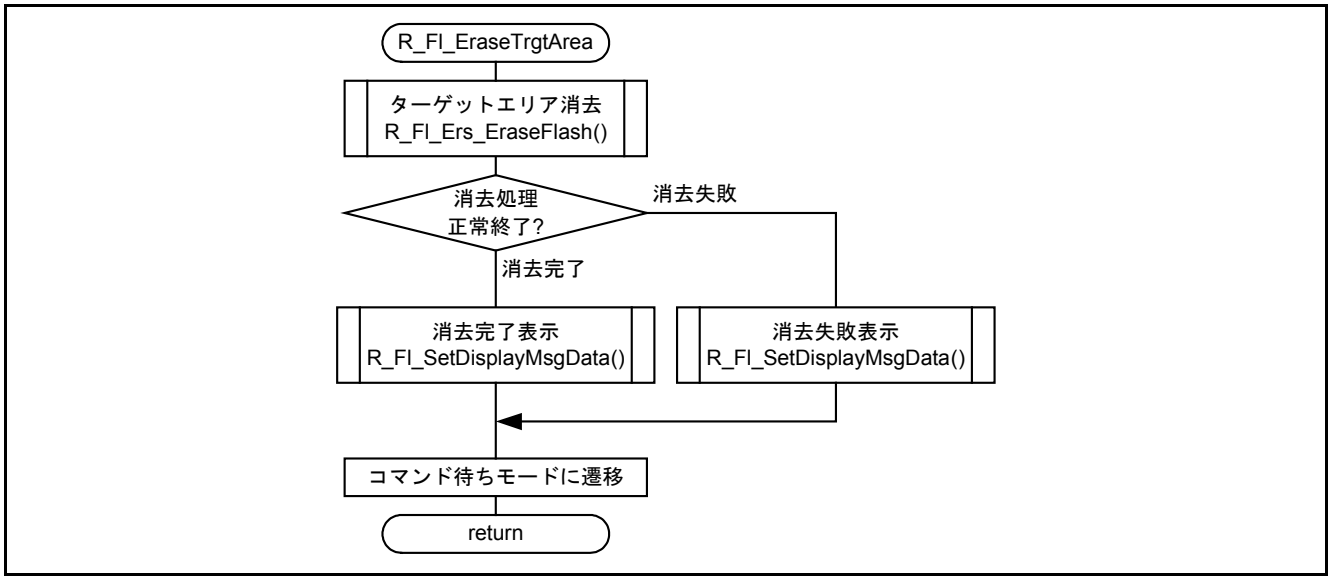

図5.14　消去モード処理

5.13.6　書き込みモード処理

図 5.15に書き込みモード処理のフローチャートを示します。

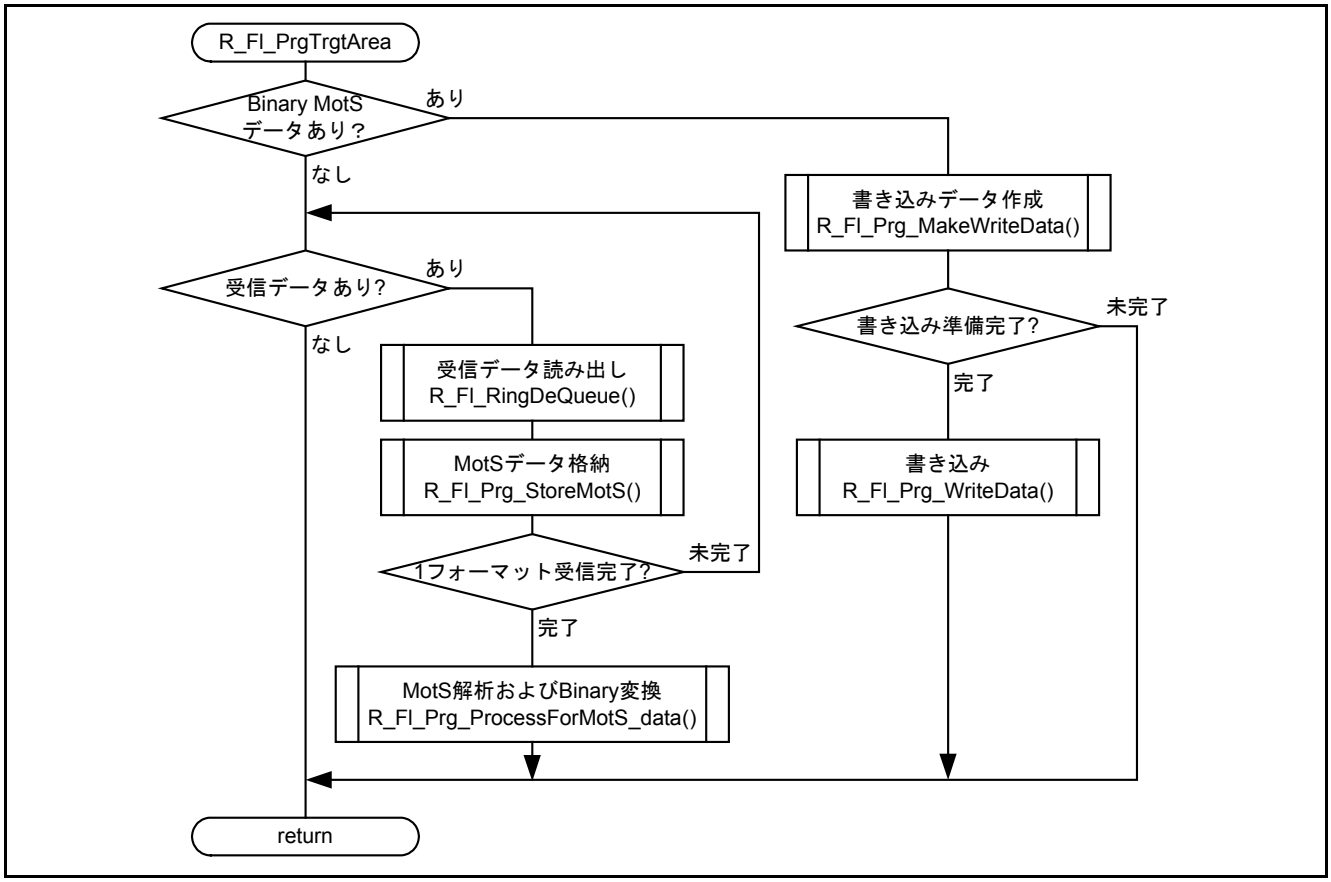

図5.15　書き込みモード処理

5.13.7　実行モード処理

図 5.16に実行モード処理のフローチャートを示します。

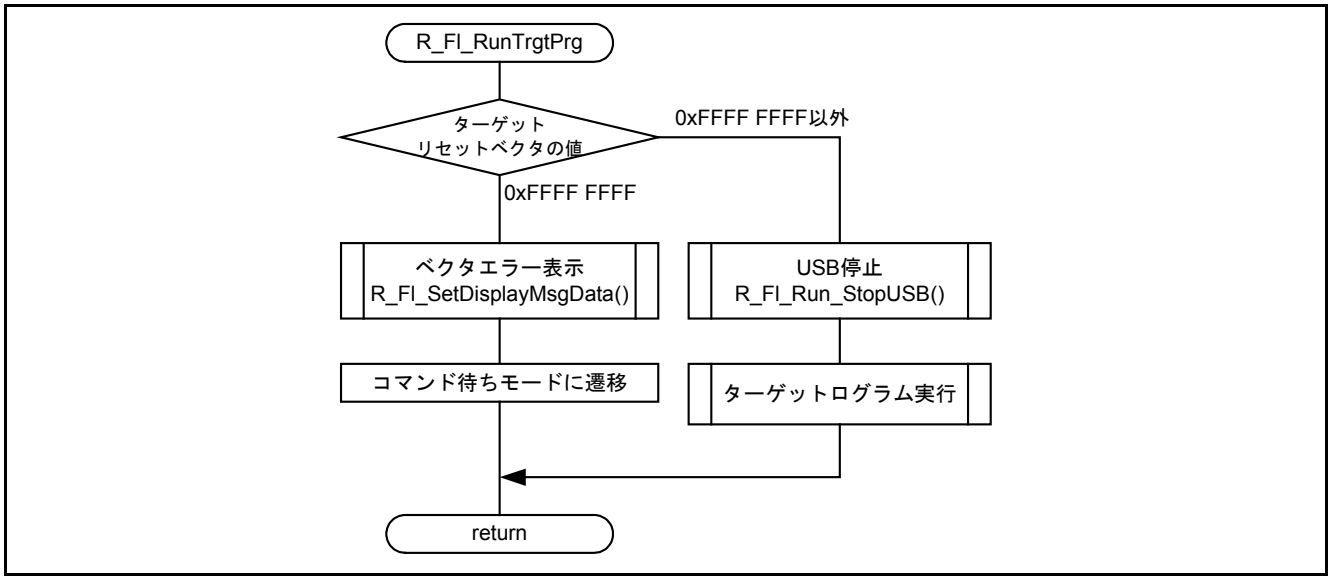

図5.16　実行モード処理

5.13.8　エラー完了待ちモード処理

図 5.17にエラー完了待ちモード処理のフローチャートを示します。

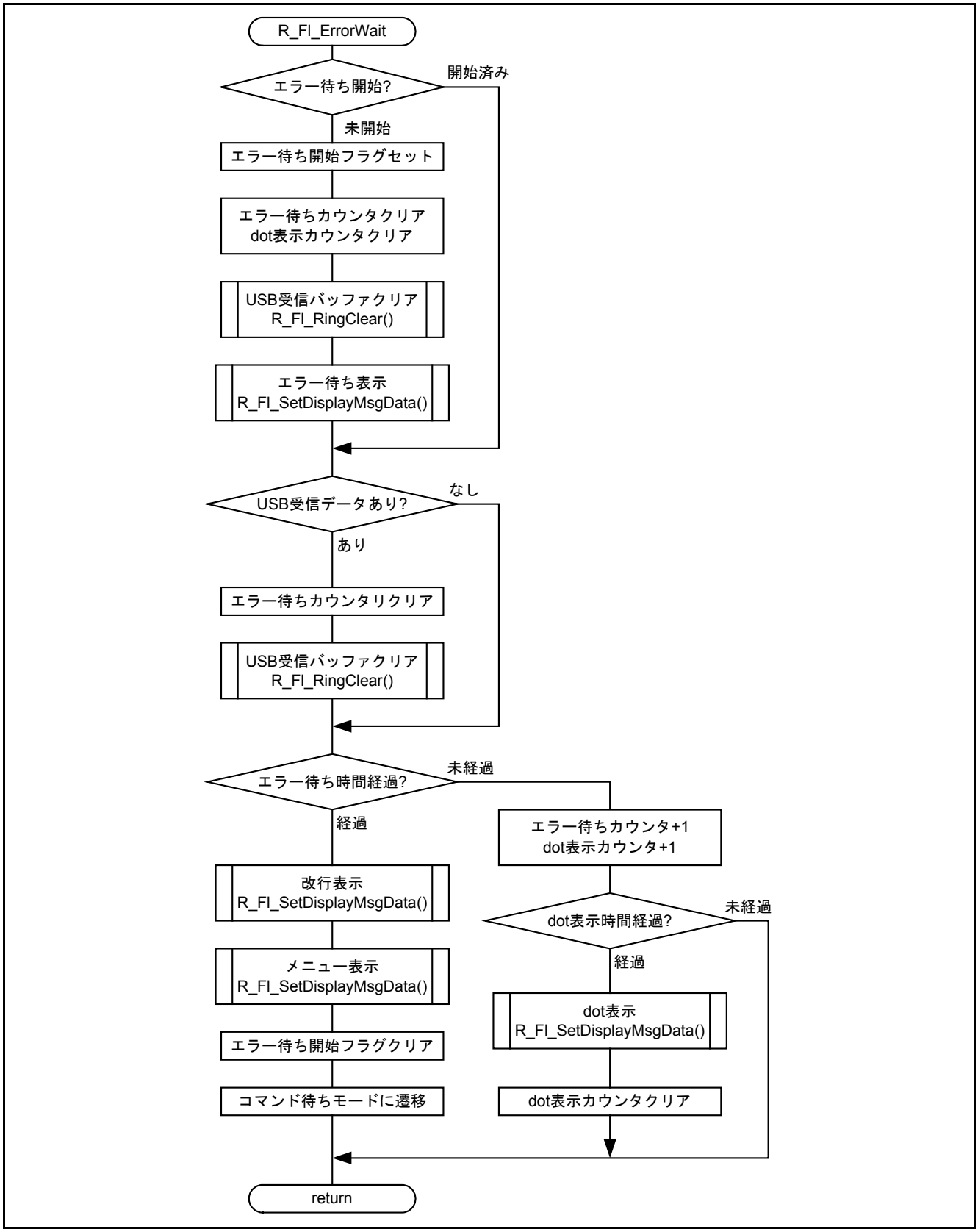

図5.17　エラー完了待ちモード処理

5.13.9　メニュー表示

図 5.18にメニュー表示のフローチャートを示します。

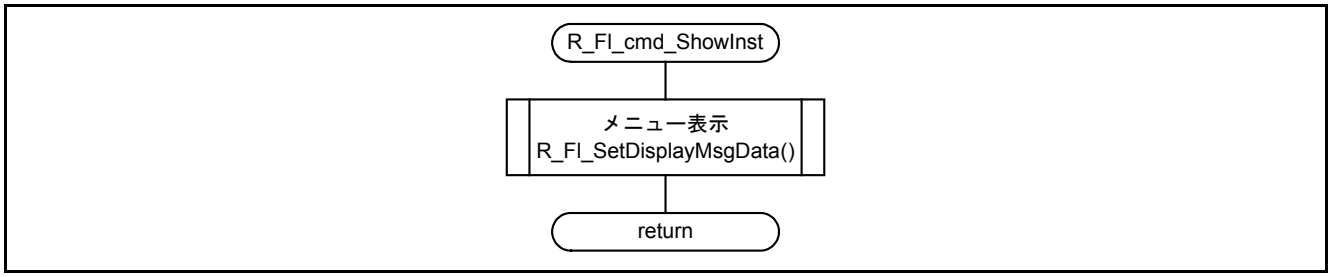

図5.18　メニュー表示

5.13.10　ブランクチェック開始

図 5.19にブランクチェック開始のフローチャートを示します。

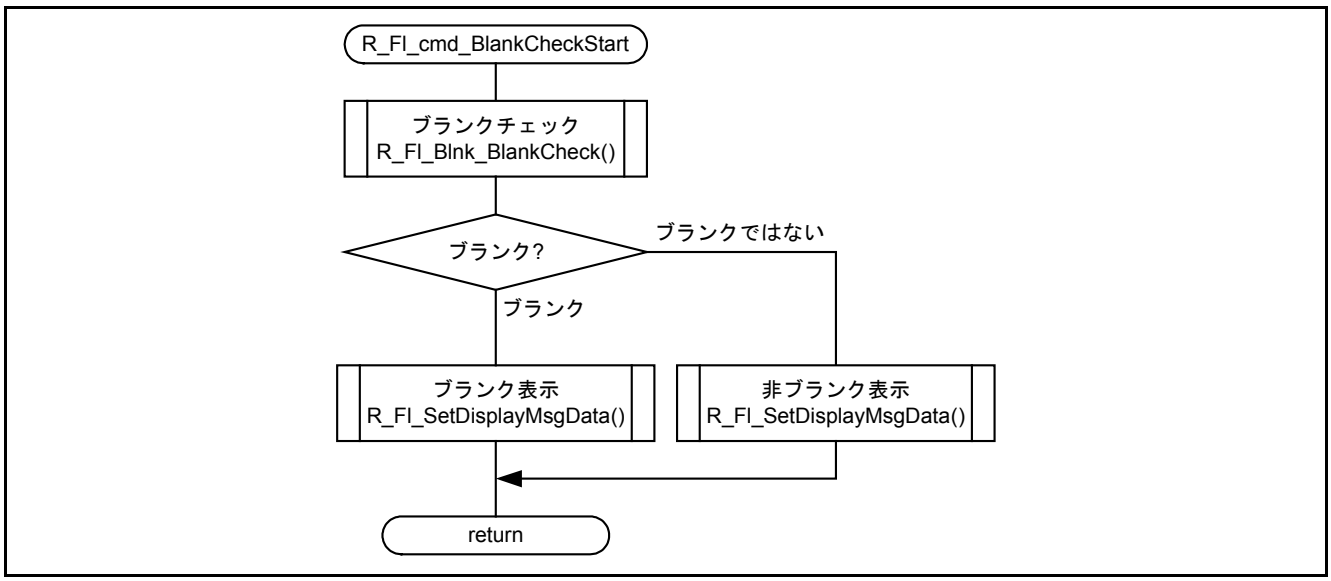

図5.19　ブランクチェック開始

5.13.11　消去開始

図 5.20に消去開始のフローチャートを示します。

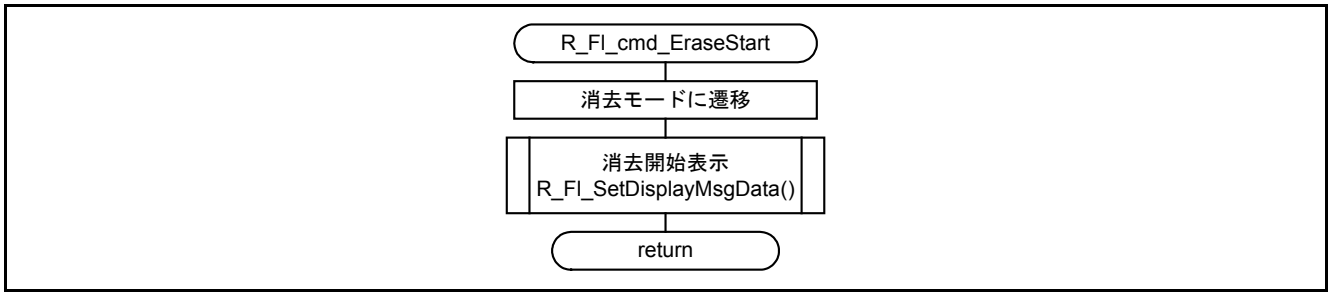

図5.20　消去開始

5.13.12　書き込み開始

図 5.21に書き込み開始のフローチャートを示します。

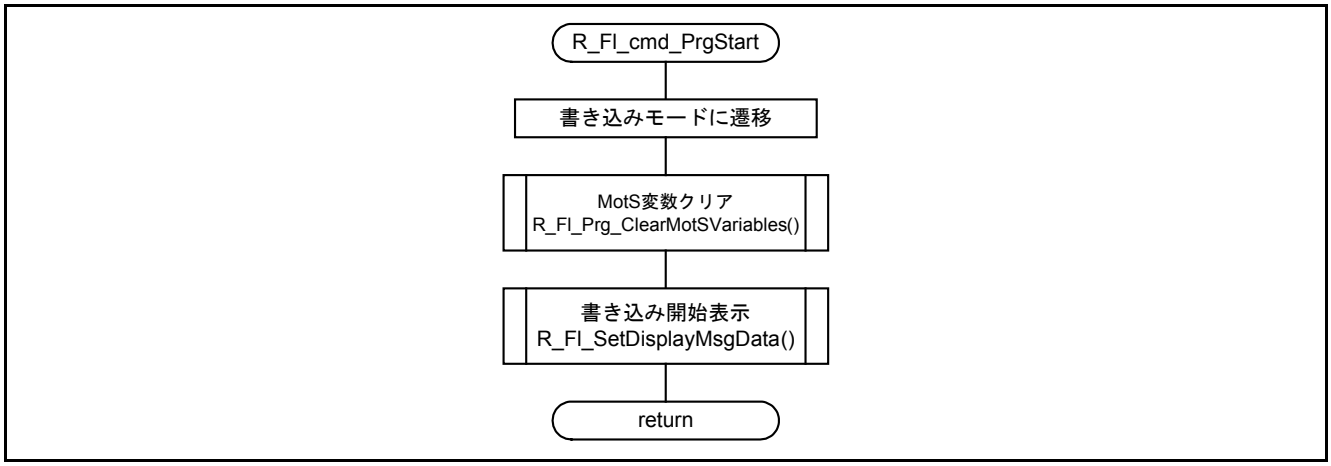

図5.21　書き込み開始

5.13.13　ターゲットプログラム実行開始

図 5.22にターゲットプログラム実行開始のフローチャートを示します。

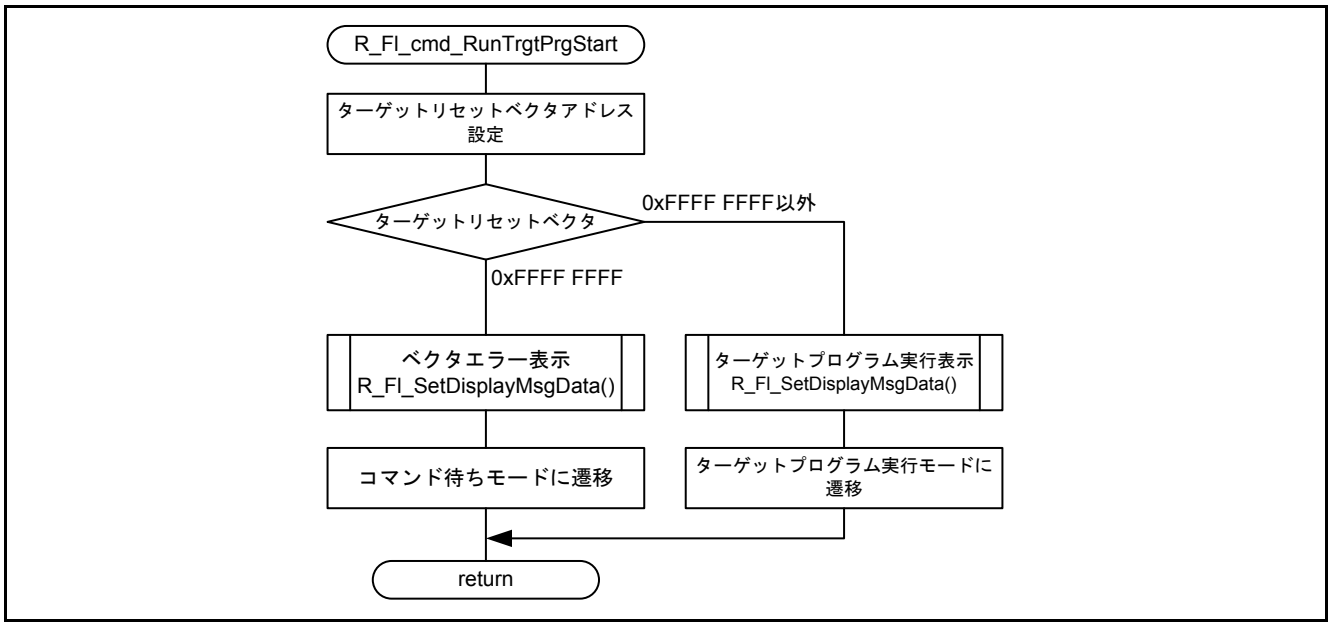

図5.22　ターゲットプログラム実行開始

5.13.14　ブランクチェック

図 5.23にブランクチェックのフローチャートを示します。

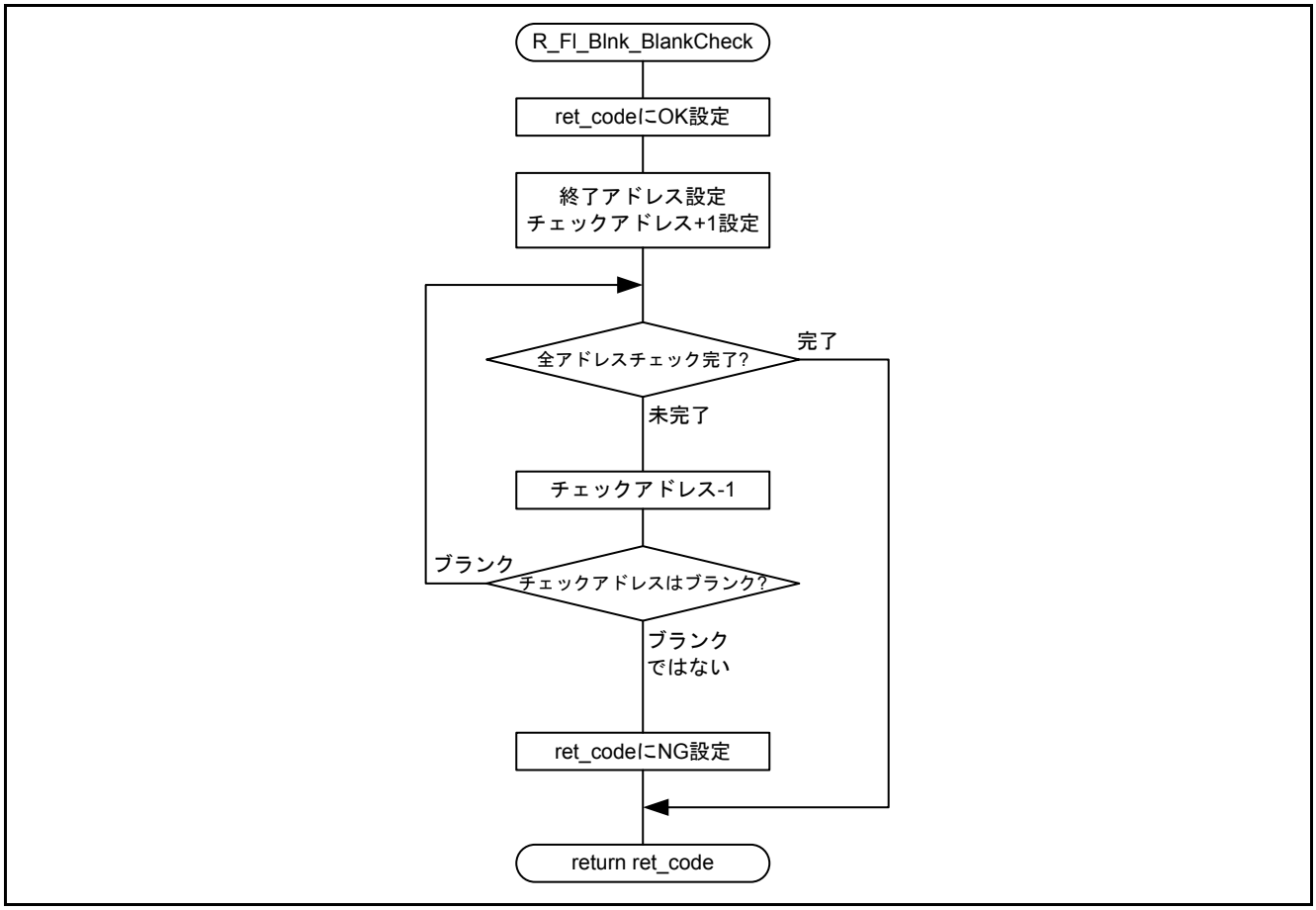

図5.23　ブランクチェック

5.13.15　ターゲットエリア消去

図 5.24にターゲットエリア消去のフローチャートを示します。

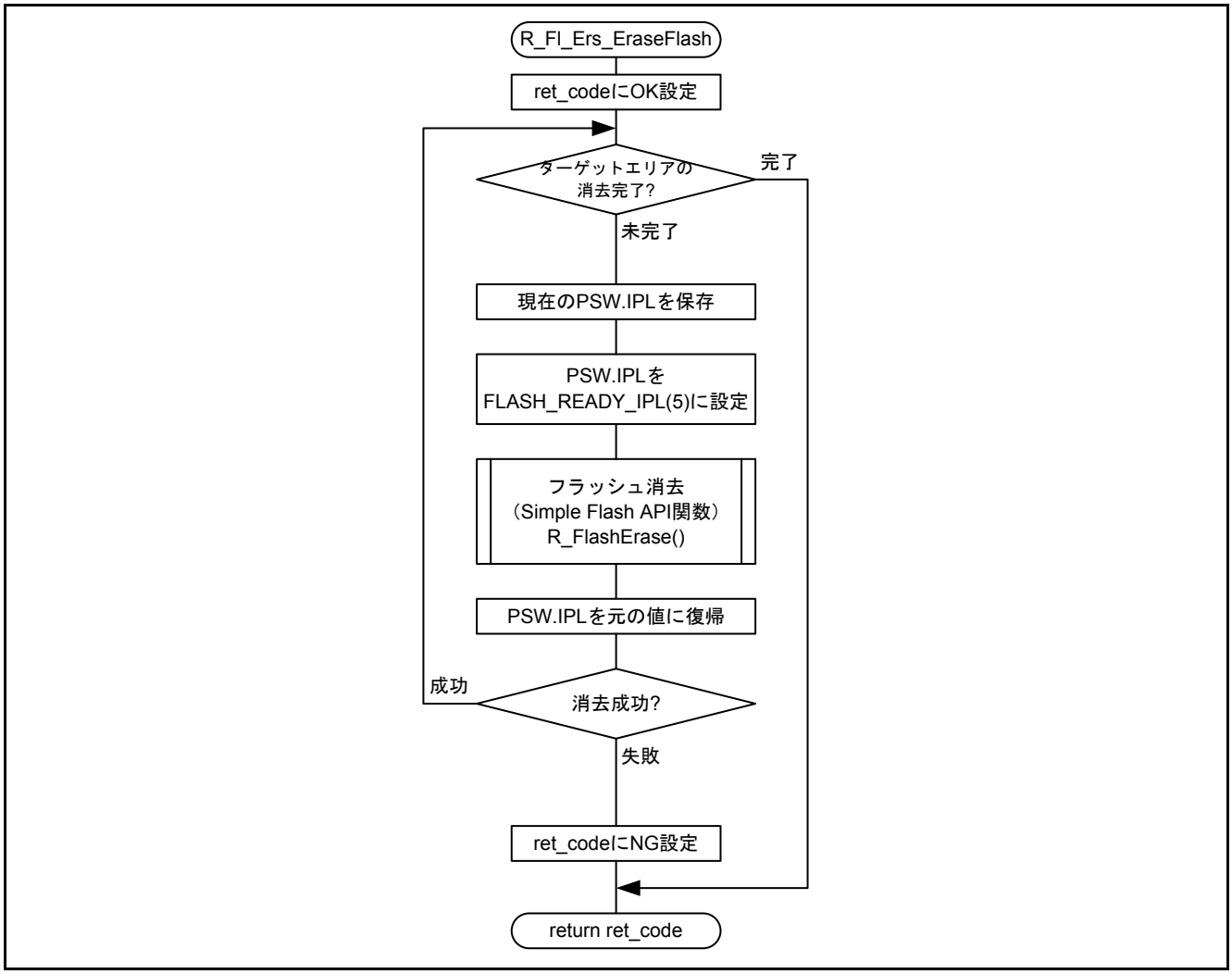

図5.24　ターゲットエリア消去

5.13.16　Motorola S フォーマットデータ格納

図 5.25にMotorola S フォーマットデータ格納のフローチャートを示します。

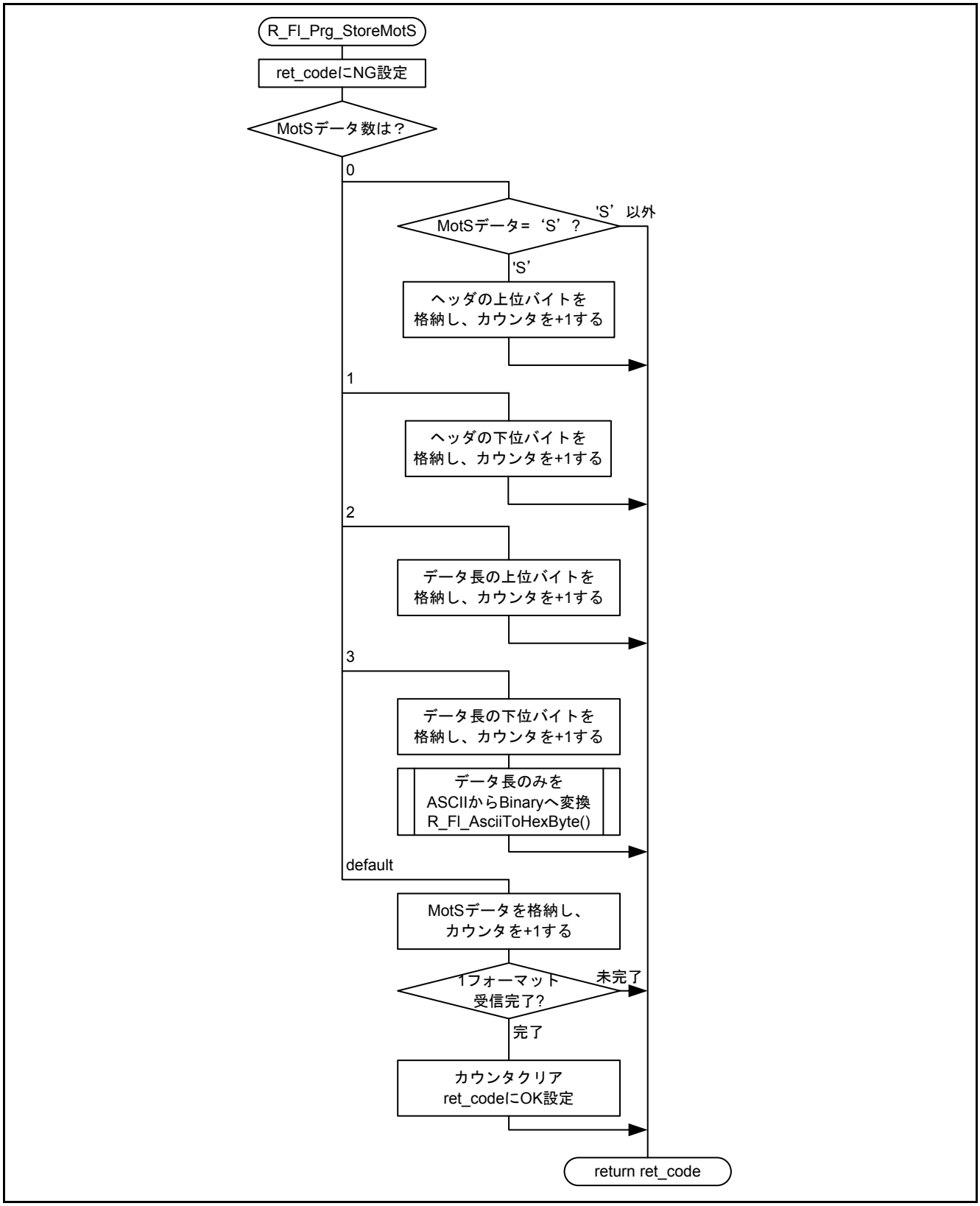

図5.25　Motorola S フォーマットデータ格納

5.13.17　Motorola S フォーマットヘッダ解析、および Binary 変換

図 5.26、図 5.27にMotorola S フォーマットヘッダ解析、および Binary 変換のフローチャートを示します。

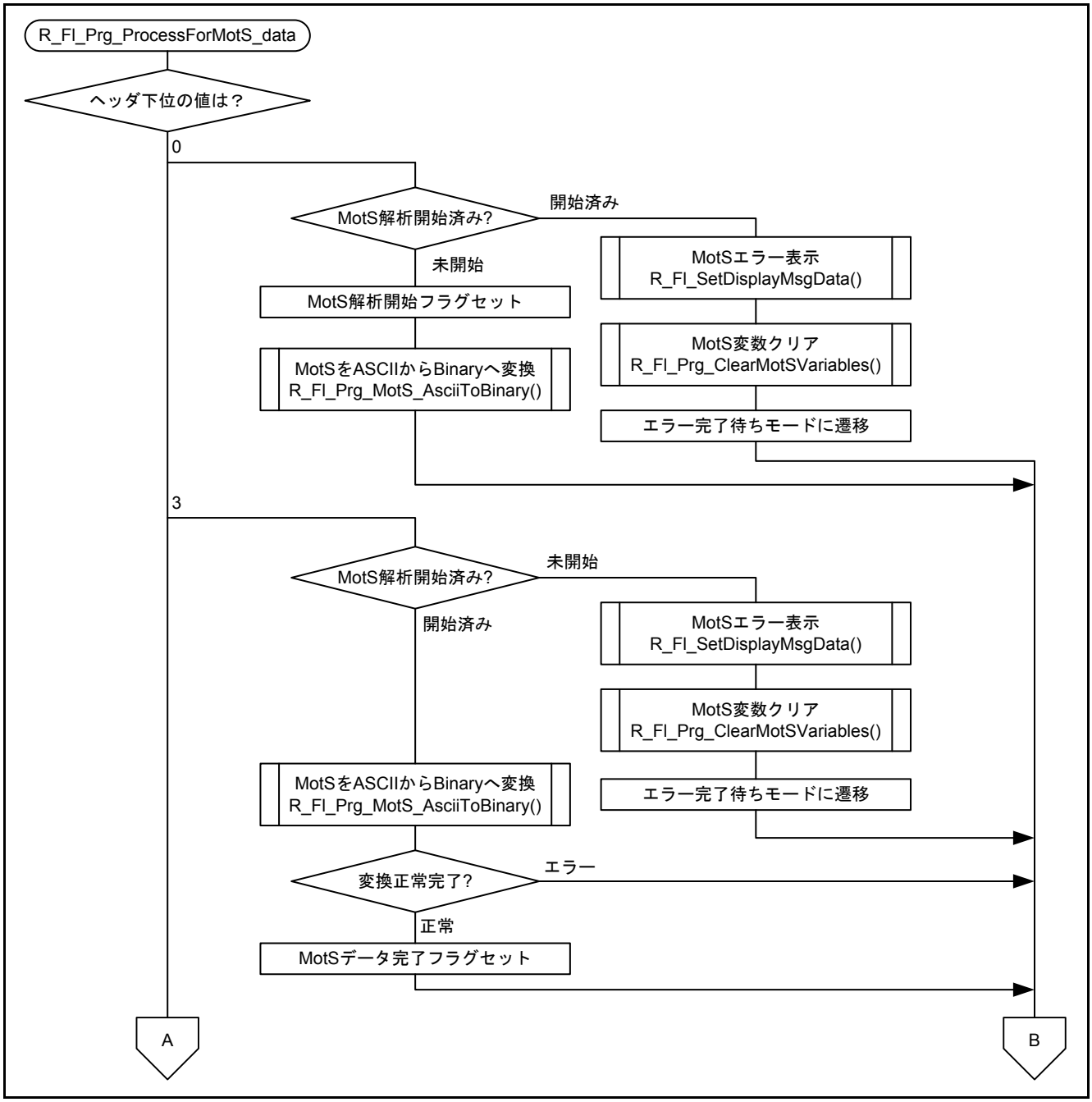

図5.26　Motorola S フォーマットヘッダ解析、および Binary 変換(1/2)

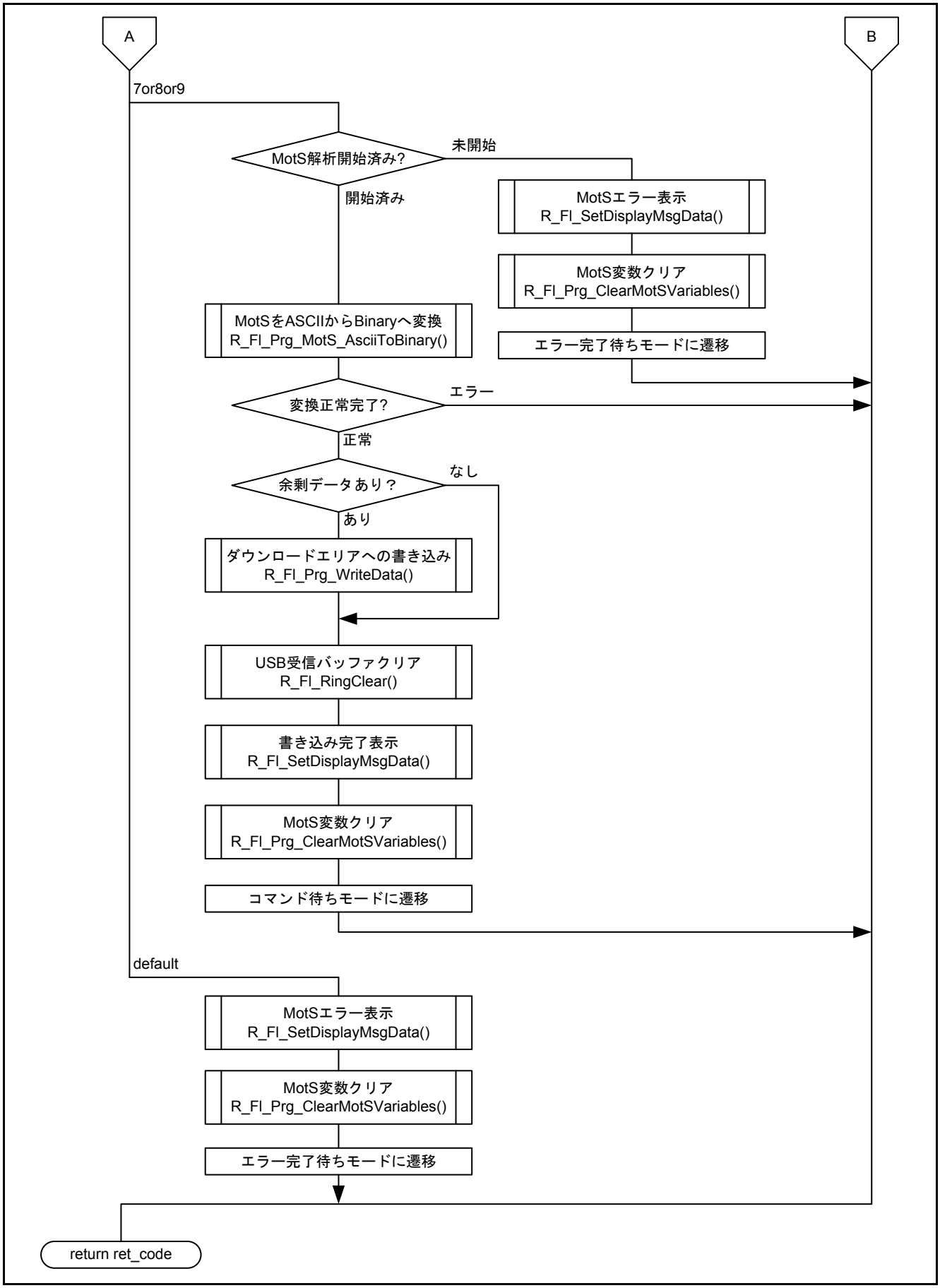

図5.27　Motorola S フォーマットヘッダ解析、および Binary 変換(2/2)

5.13.18　Motorola S フォーマットデータ ASCII - Binary 変換

図 5.28にMotorola S フォーマットデータ ASCII - Binary 変換のフローチャートを示します。

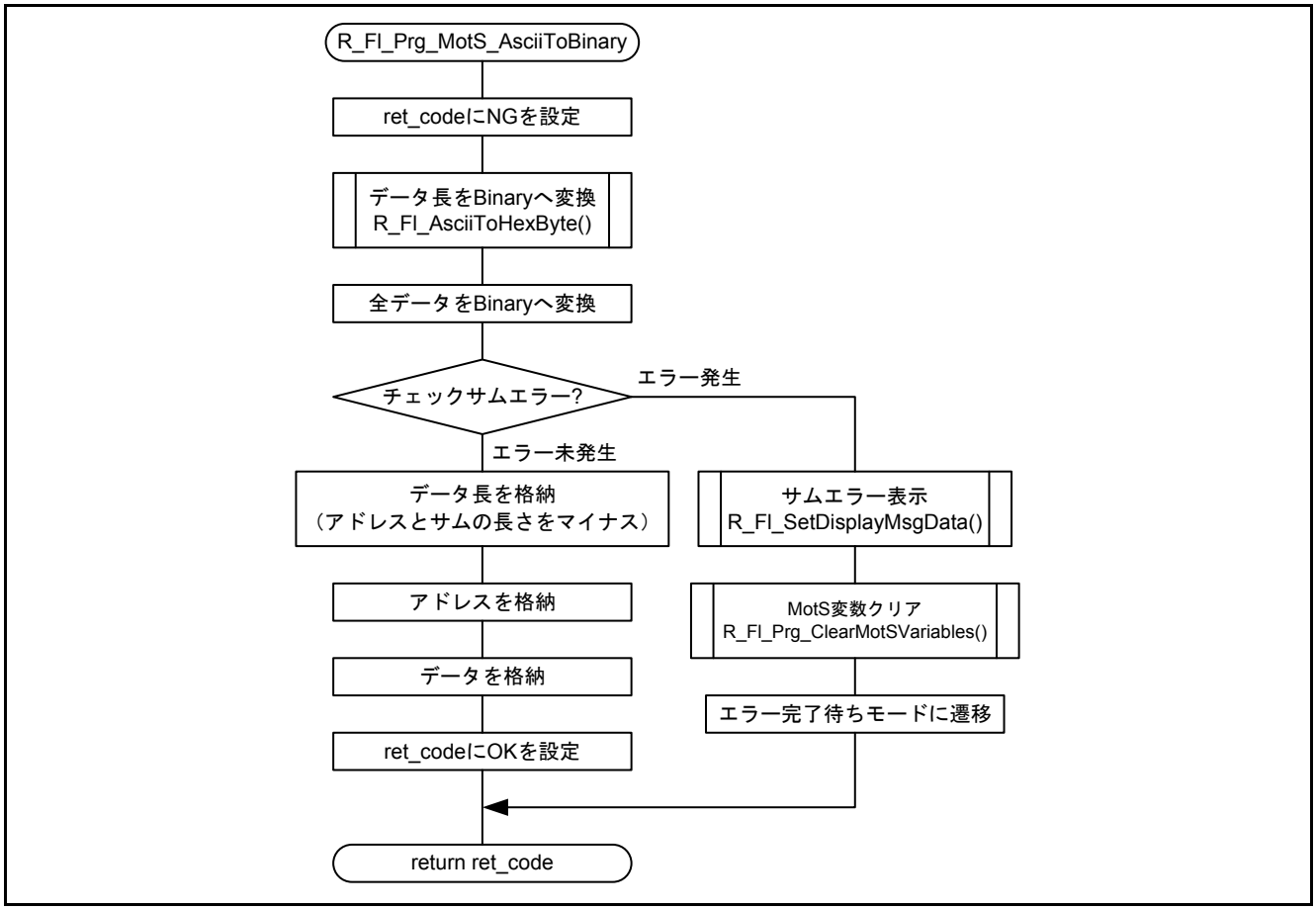

図5.28　Motorola S フォーマットデータ ASCII - Binary 変換

5.13.19　ターゲットエリアへの書き込みデータ作成

図 5.29にターゲットエリアへの書き込みデータ作成のフローチャートを示します。

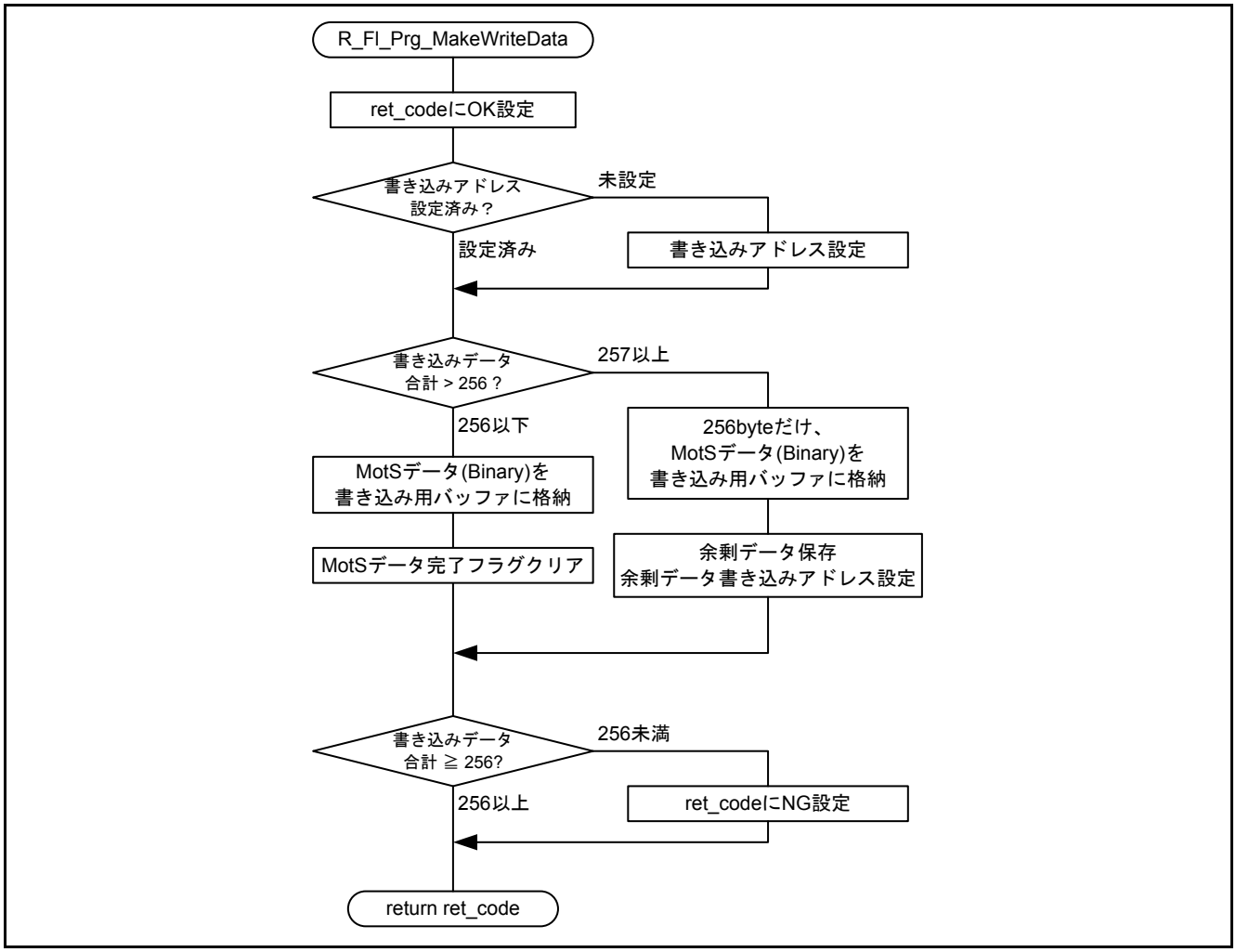

図5.29　ターゲットエリアへの書き込みデータ作成

5.13.20　ターゲットエリアへの書き込み

図 5.30にターゲットエリアへの書き込みのフローチャートを示します。

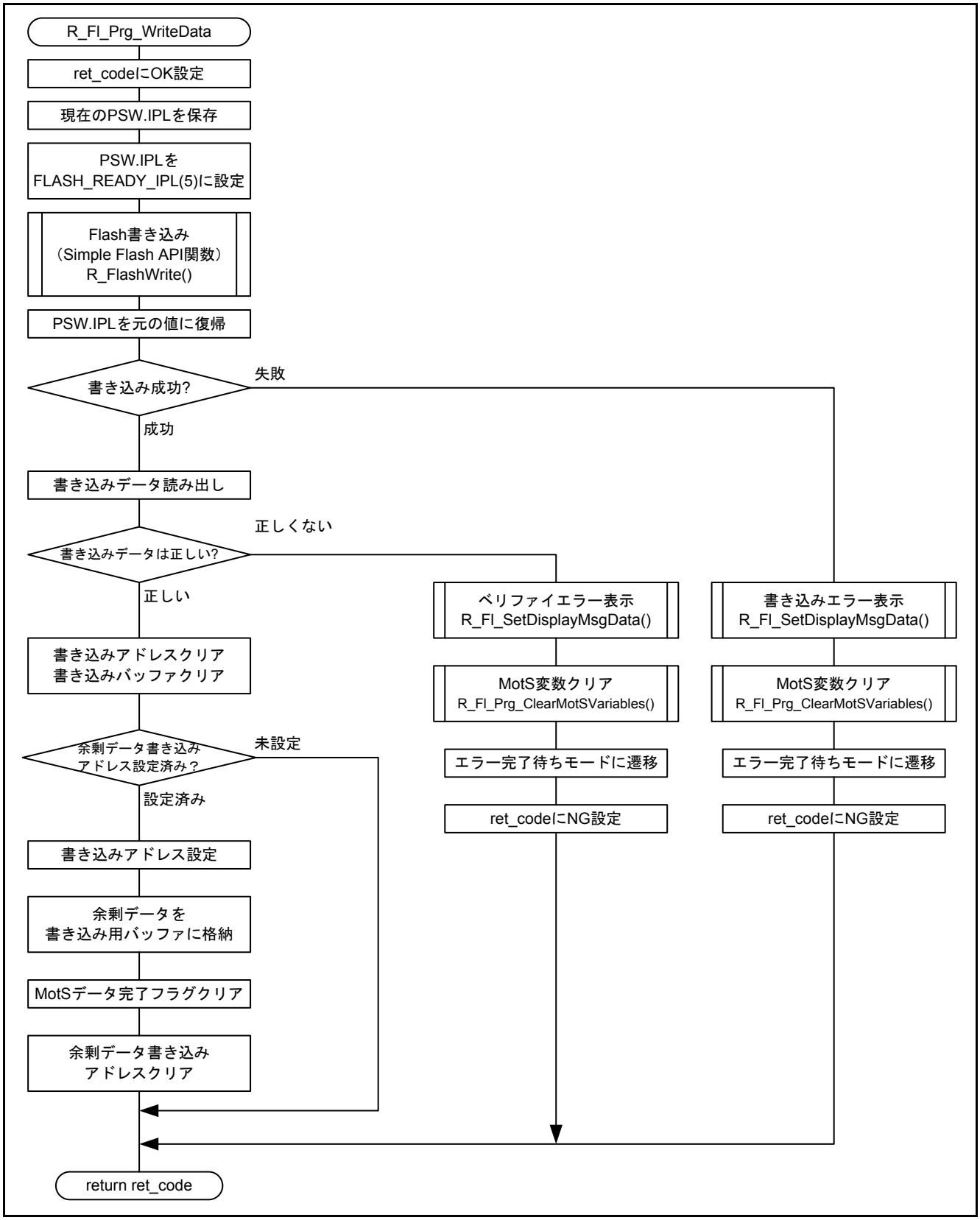

図5.30　ターゲットエリアへの書き込み

5.13.21　Motorola S フォーマット用変数クリア

図 5.31にMotorola S フォーマット用変数クリアのフローチャートを示します。

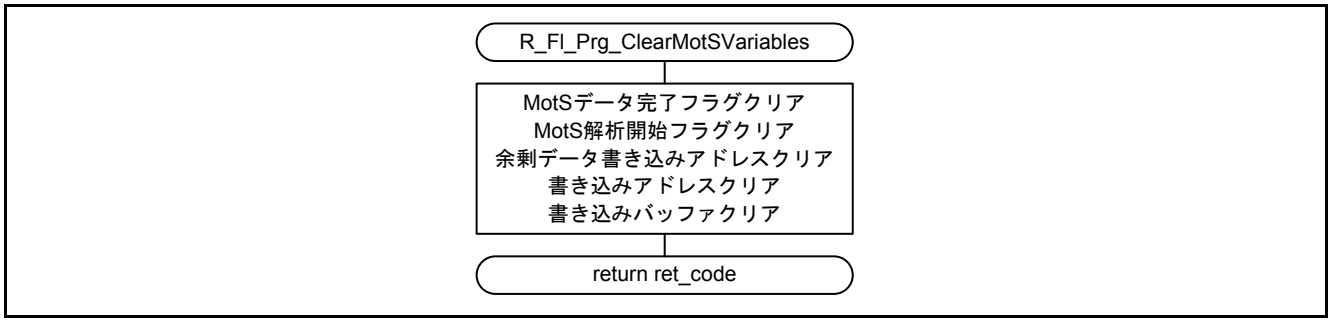

図5.31　Motorola S フォーマット用変数クリア

5.13.22　USB 停止

図 5.32にUSB 停止のフローチャートを示します。

図5.32　USB 停止

5.13.23　USB 受信データ格納

図 5.33にUSB 受信データ格納のフローチャートを示します。

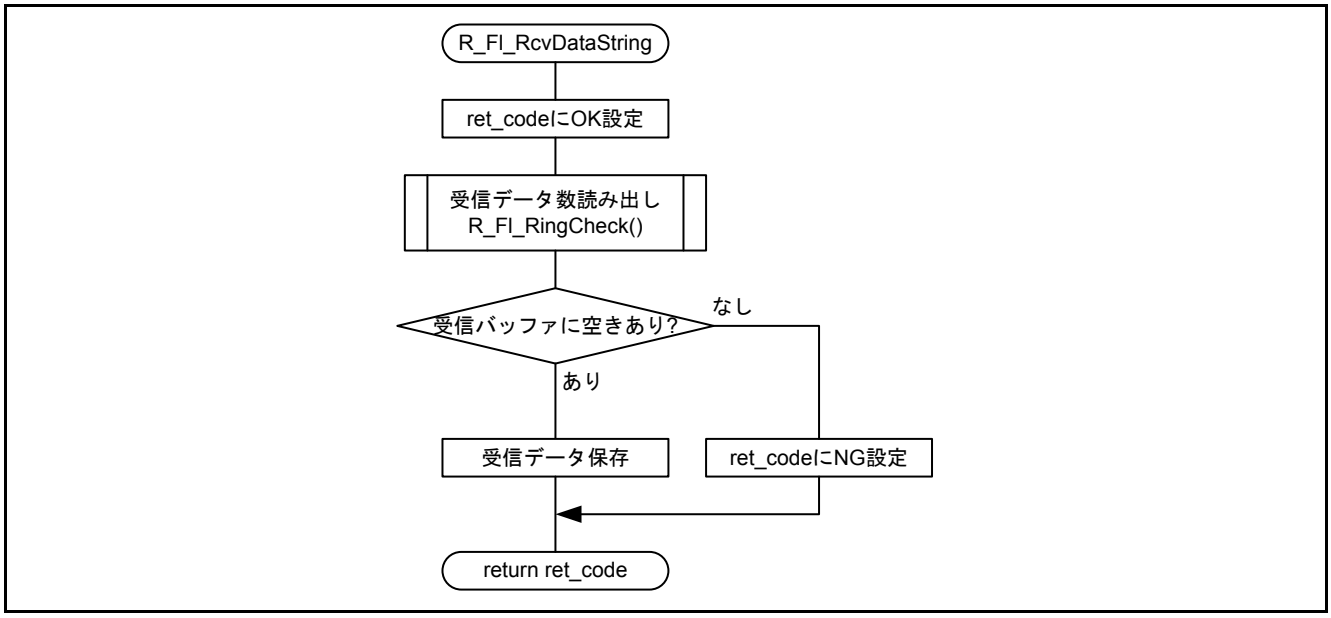

図5.33　USB 受信データ格納

5.13.24　USB 送信データ格納

図 5.34にUSB 送信データ格納のフローチャートを示します。

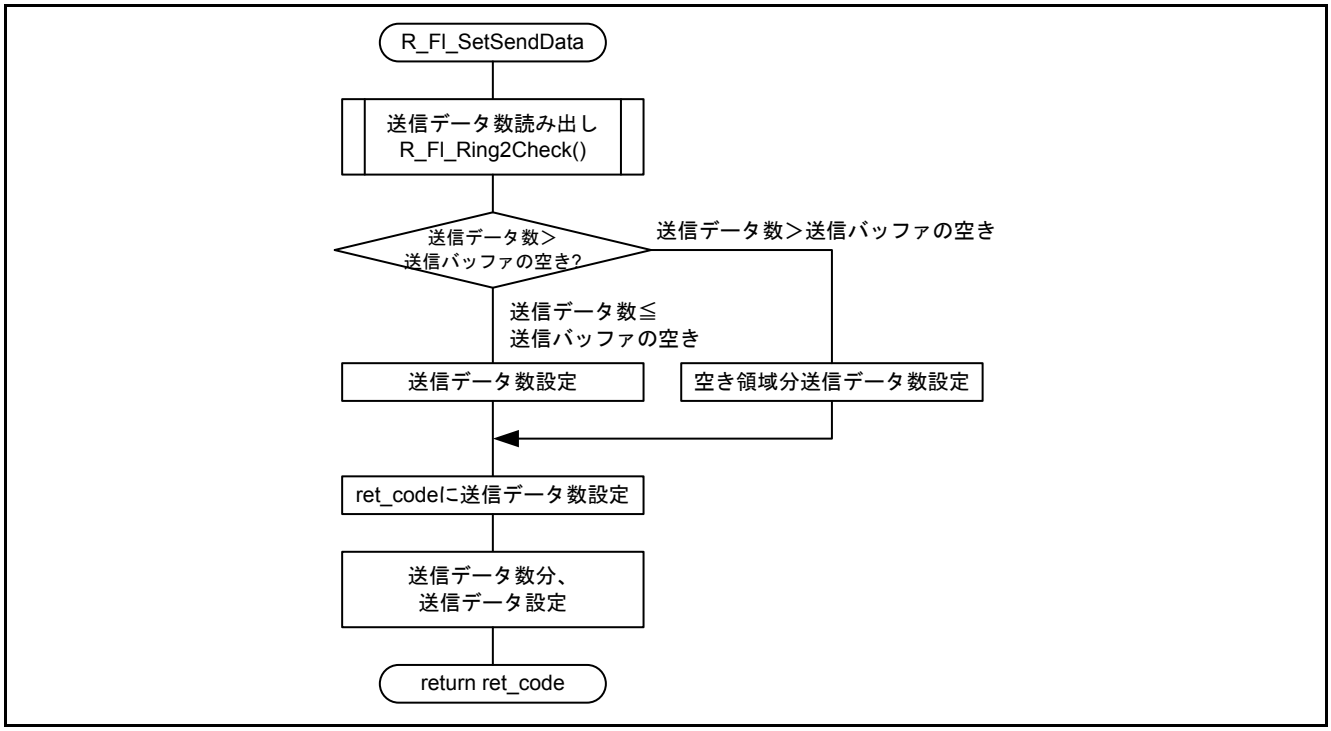

図5.34　USB 送信データ格納

5.13.25　メッセージ表示

図 5.35にメッセージ表示のフローチャートを示します。

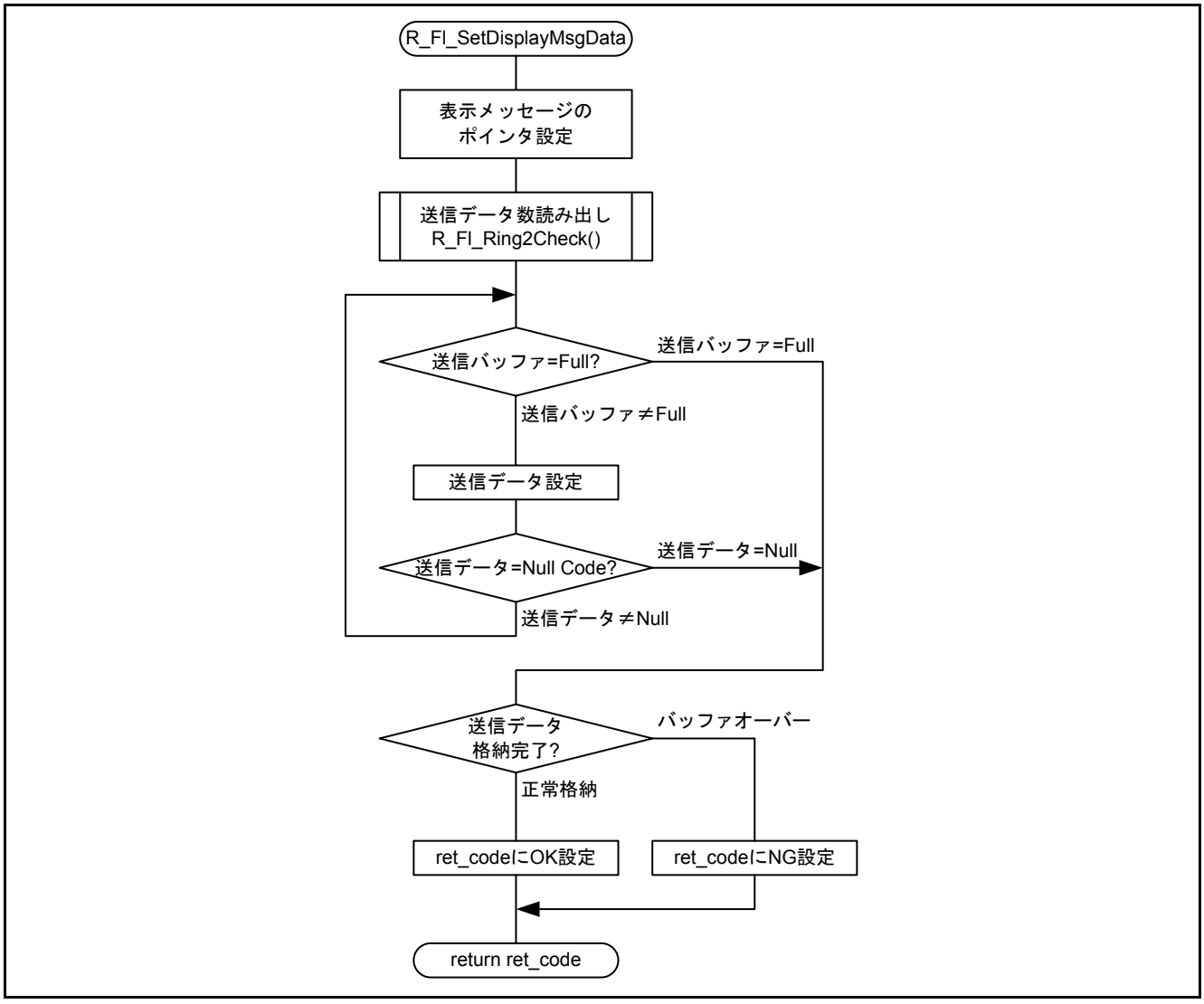

図5.35　メッセージ表示

5.13.26　受信用リングバッファの空き容量確認

図 5.36に受信用リングバッファの空き容量確認のフローチャートを示します。

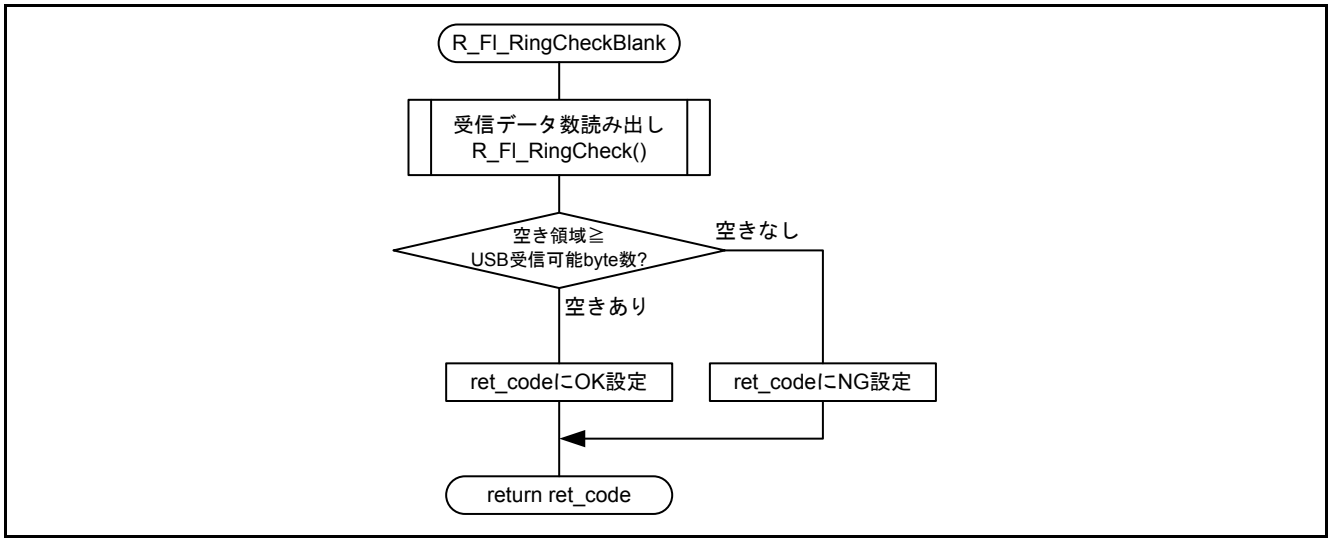

図5.36　受信用リングバッファの空き容量確認

5.13.27　送信用リングバッファのデータ確認

図 5.37に送信用リングバッファのデータ確認のフローチャートを示します。

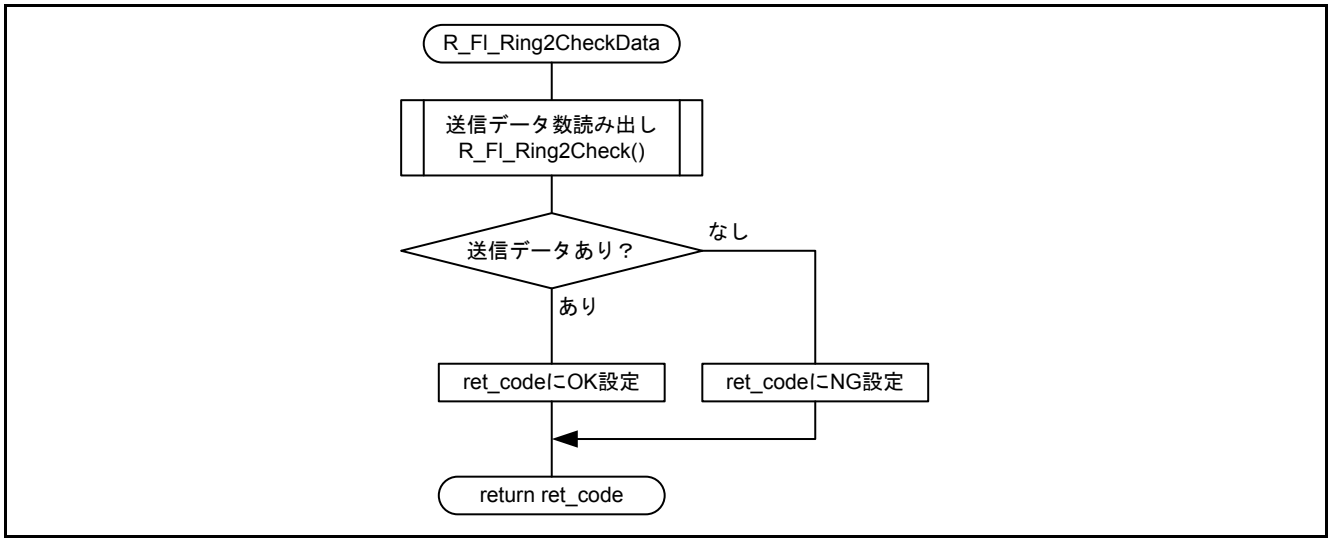

図5.37　送信用リングバッファのデータ確認

5.13.28　USB 未接続時のターゲットプログラム実行

図 5.38にUSB 未接続時のターゲットプログラム実行のフローチャートを示します。

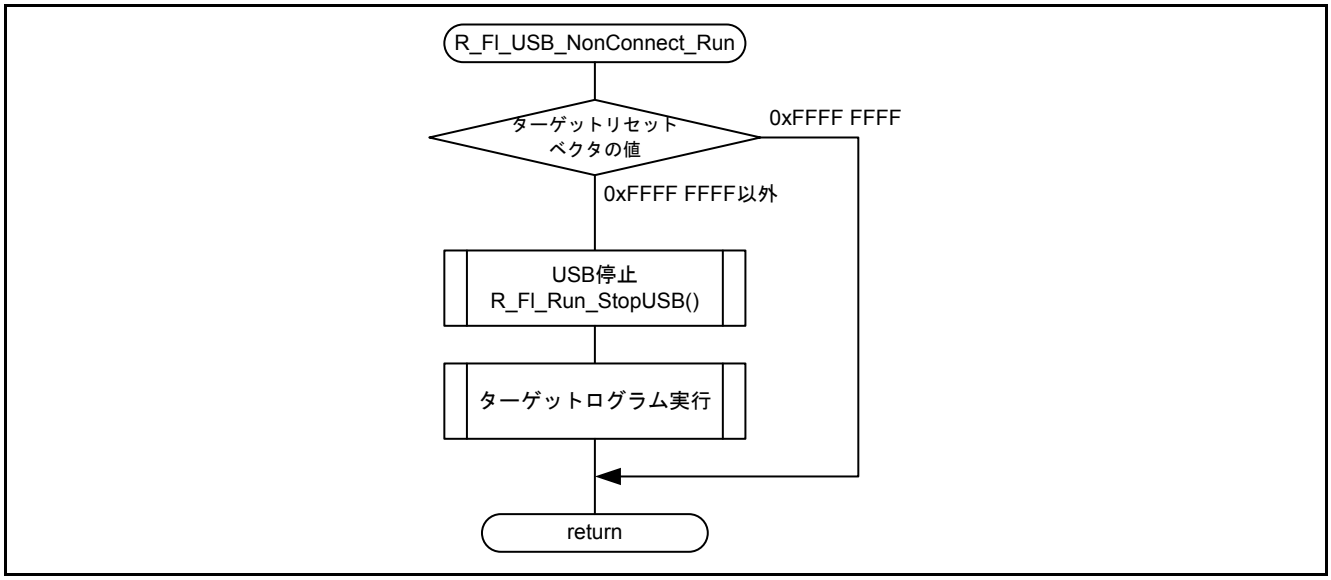

図5.38　USB 未接続時のターゲットプログラム実行

5.13.29　受信用リングバッファへのデータ格納

図 5.39に受信用リングバッファへのデータ格納のフローチャートを示します。

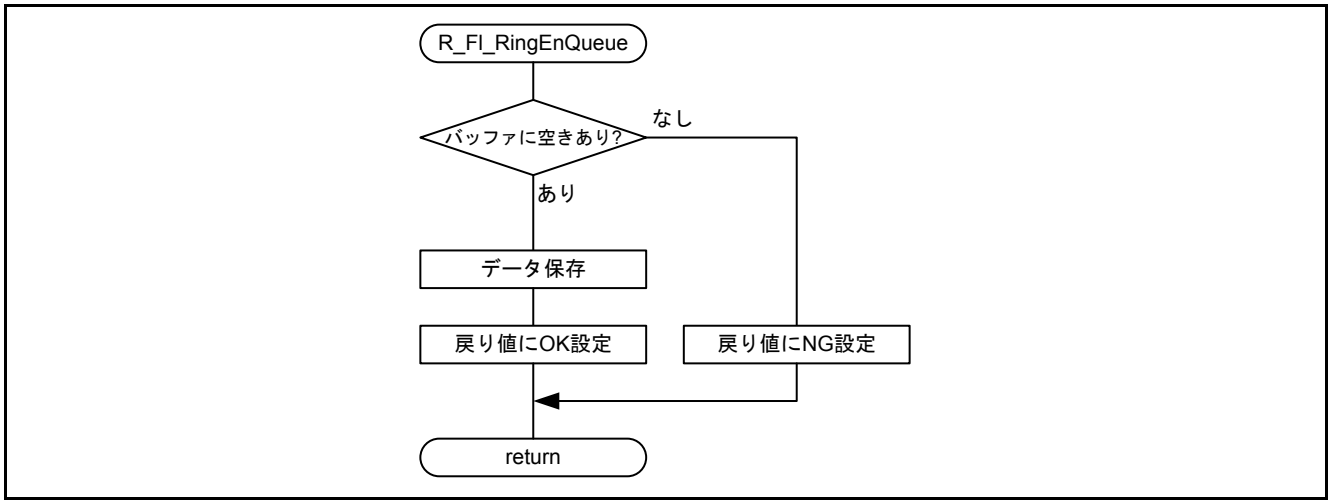

図5.39　受信用リングバッファへのデータ格納

5.13.30　受信用リングバッファからのデータ読み出し

図 5.40に受信用リングバッファからのデータ読み出しのフローチャートを示します。

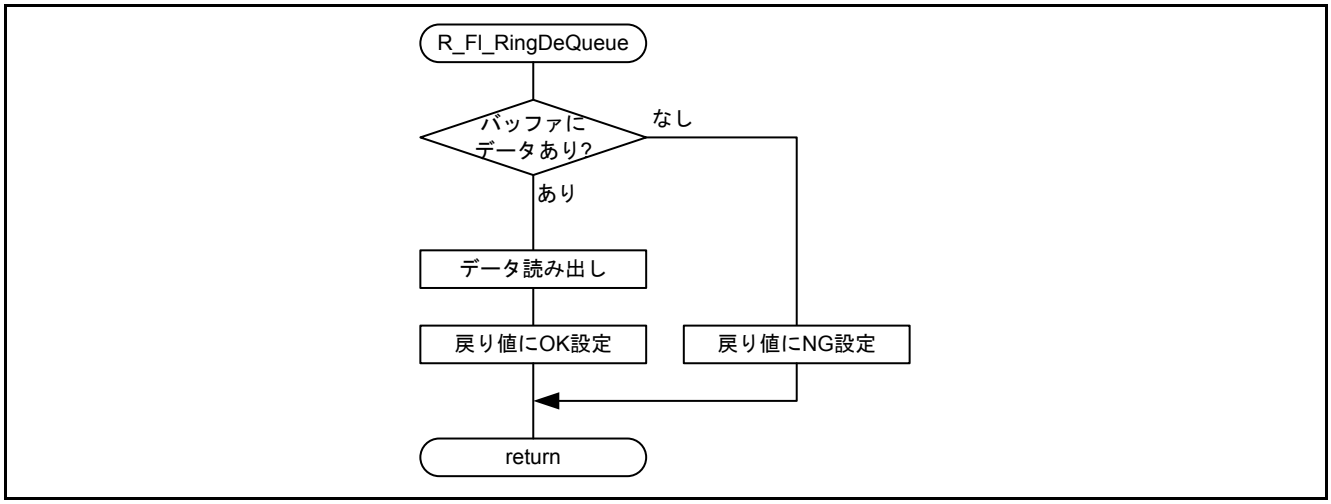

図5.40　受信用リングバッファからのデータ読み出し

5.13.31　受信用リングバッファクリア

図 5.41に受信用リングバッファクリアのフローチャートを示します。

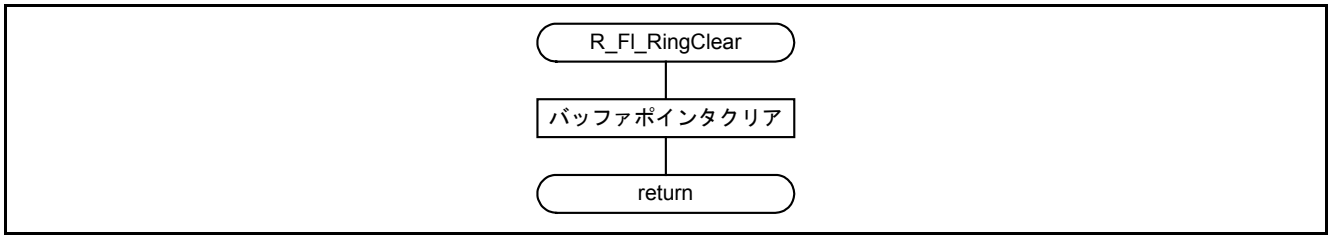

図5.41　受信用リングバッファクリア

5.13.32　受信用リングバッファのデータ数確認

図 5.42に受信用リングバッファのデータ数確認のフローチャートを示します。

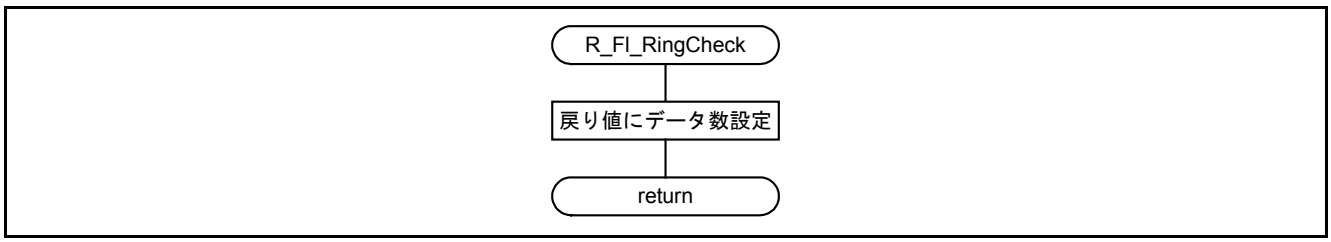

図5.42　受信用リングバッファのデータ数確認

5.13.33　送信用リングバッファへのデータ格納

図 5.43に送信用リングバッファへのデータ格納のフローチャートを示します。

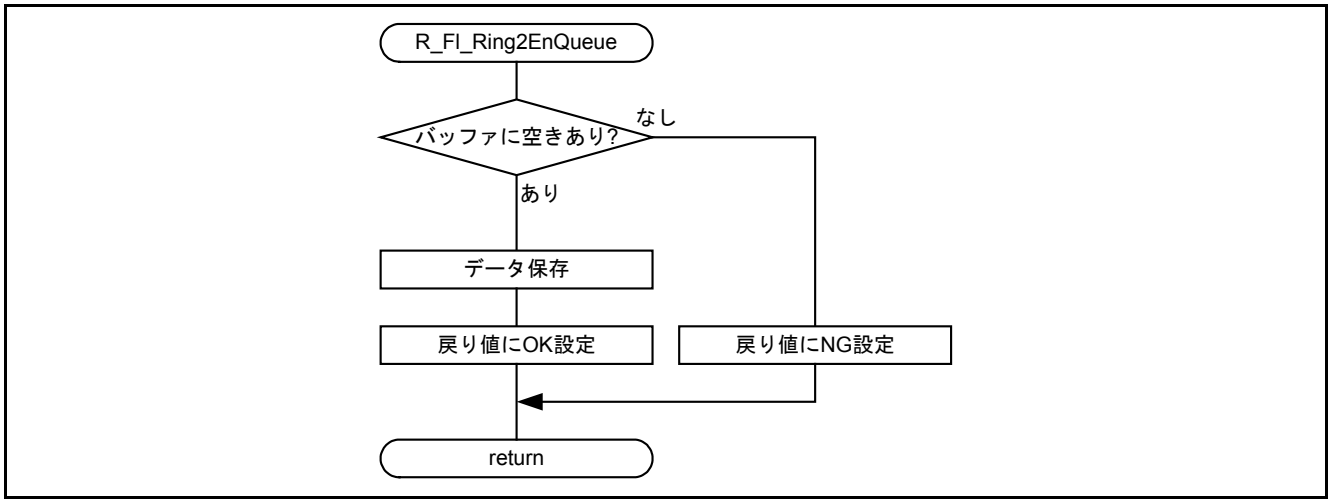

図5.43　送信用リングバッファへのデータ格納

5.13.34　送信用リングバッファからのデータ読み出し

図 5.44に送信用リングバッファからのデータ読み出しのフローチャートを示します。

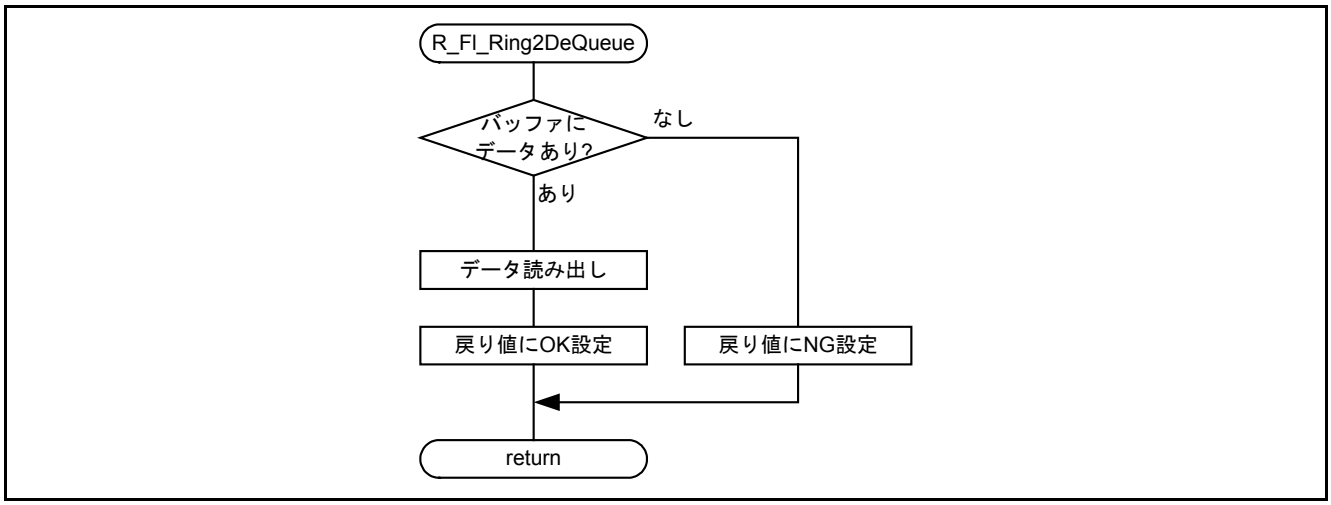

図5.44　送信用リングバッファからのデータ読み出し

5.13.35　送信用リングバッファクリア

図 5.45に送信用リングバッファクリアのフローチャートを示します。

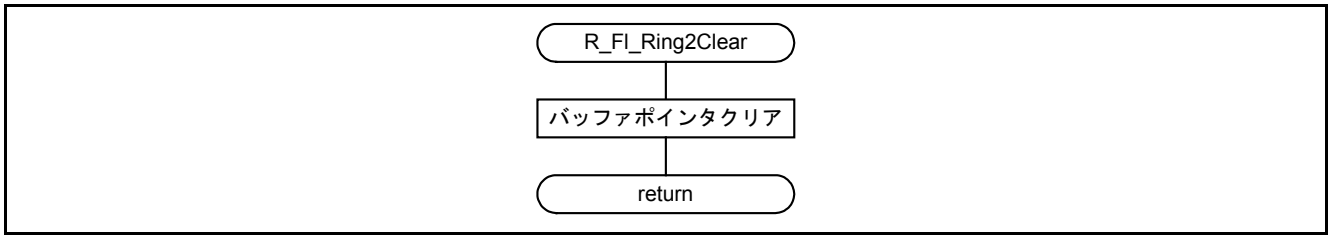

図5.45　送信用リングバッファクリア

5.13.36　送信用リングバッファのデータ数確認

図 5.46に送信用リングバッファのデータ数確認のフローチャートを示します。

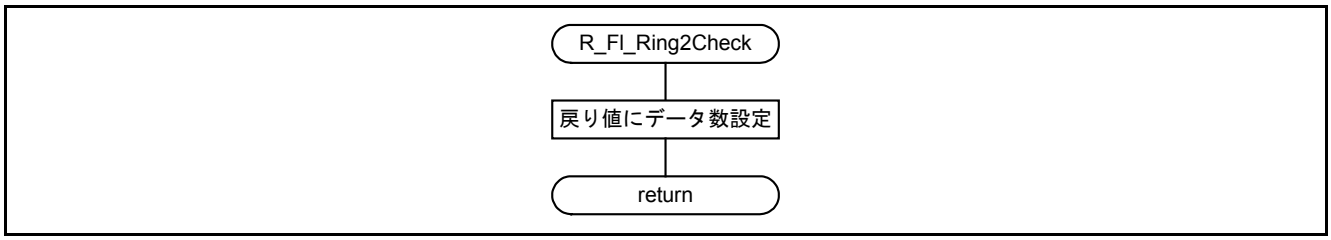

図5.46　送信用リングバッファのデータ数確認

5.13.37　ASCII コードから Binary データへの変換

図 5.47にASCII コードから Binary データへの変換のフローチャートを示します。

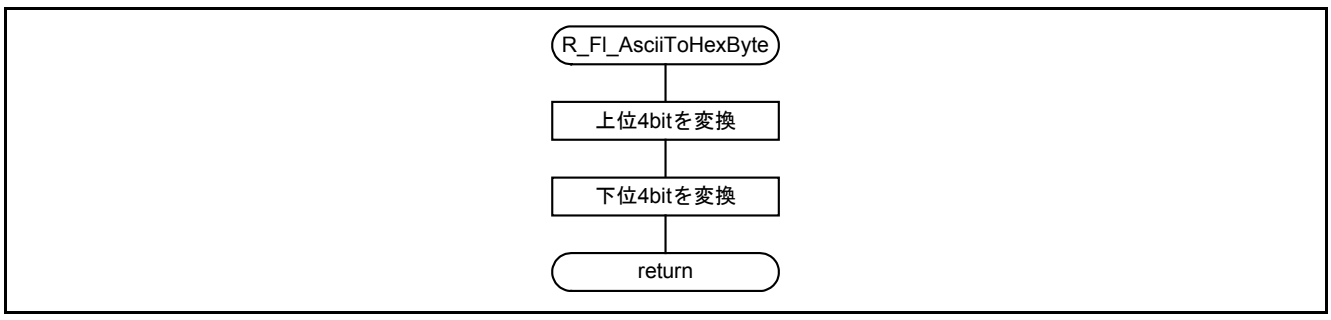

図5.47　ASCII コードから Binary データへの変換

6. 使用方法

(1) サンプルコードの Vender ID および Product ID を変更します。
ファイル" r_usb_pcdc_descriptor.c "の中で定義されている" USB_VENDORID"と" USB_PRODUCTID" をご使用になる環境に合わせて変更してください。変更の詳細は USB Peripheral Communication Device Class Driver および USB Basic Firmware のアプリケーションノートを参照してください。

```
/***********************************************************************
User define macro definitions
***********************************************************************/
#define USB_VENDORID    0x0000
#define USB_PRODUCTID   0x0002
```

図6.1　サンプルコードの Vender ID、Product ID (ファイル" r_usb_pcdc_descriptor.c")

(2) USB ドライバの Vender ID および Product ID を変更します。
フォルダ"usb_driver"の中にあるファイル" CDC_Demo.inf"または"CDC_Demo_Win7.inf"の"VID_0000" と"PID_0002"を変更します。変更値は上記サンプルコードの Vender ID、Product ID と同じである必要があります。変更の詳細は USB Peripheral Communication Device Class Driver および USB Basic Firmware のアプリケーションノートを参照してください。

```
[Manufacturer]
%STRING_MAUNUFACTURER%=Model

[Model]
%STRING_MODEL%=CDC, USB\VID_0000&PID_0002
```

図6.2　USB driver の Vender ID、Product ID(ファイル"CDC_Demo.inf")

```
[DeviceList]
%DESCRIPTION%=DriverInstall, USB\VID_0000&PID_0002

[DeviceList.NTamd64]
%DESCRIPTION%=DriverInstall, USB\VID_0000&PID_0002
```

図6.3　USB driver の Vender ID、Product ID(ファイル"CDC_Demo_Win7.inf")

(3) サンプルコードをすべてビルドし、マイコンに書き込みます。
(4) マイコンへ電源投入後、PC とマイコンを USB で接続します。(注 1)
(5) Windows の USB ドライバインストール画面で、フォルダ"usb_driver"の中にあるファイル "CDC_Demo.inf"または"CDC_Demo_Win7.inf"を選択し、ドライバをインストールします。すでに USB ドライバがインストールされている場合は、この手順は不要です。
(6) ターミナルソフトを起動し、マイコンとの通信を開始します。
(7) 以降はターミナルソフトに表示されるメッセージに従い操作します。

注 1：　ターゲットリセットベクタに FFFF FFFFh 以外の値が書かれていた場合、かつ一定時間 USB が接続されない状態が続くと、ターゲットプログラムが実行されます。ターゲットプログラム実行時には、USB を停止していますのでご注意ください。USB のドライバインストール時には、USB の接続に時間がかかる可能性がありますので、ターゲットリセットベクタには FFFF FFFFh を書いておいてください。

7. ターゲットプログラムの例

本アプリケーションノートのファイルには、ターゲットプログラムの例(UsrPrgSample.zip)が同梱されています。「2. 動作確認条件」に示されているボードの LED を順に点灯していくプログラムです。ターゲットリセットベクタの設定やセクションの設定の参考にしてください。なお、ターゲットプログラムは、1M バイトの ROM 容量の使用を前提としています。

8. 注意事項

8.1 書き込み速度

ご使用になるターミナルソフトによっては、書き込みが非常に遅くなる場合があります。もし書き込みが非常に遅い場合には、ご使用になられているターミナルソフトを変更してお試しください。

8.2 書き込み、消去中の USB 切断

ターゲットエリアへの書き込み、消去中は、USB を抜き差ししないでください。

8.3 HEW の設定

サンプルコードは、フラッシュ書き換え時に、ROM 上のコードを RAM に転送して実行します。これら設定の詳細は RX600&RX200 シリーズ RX 用シンプルフラッシュ API のアプリケーションノートの「2.10 プロジェクトにミドルウェアを加えるには」と「2.12 フラッシュ API コードを RAM 上に置くには」を参照してください。

8.4 USB の Vender ID と Product ID

サンプルコードを使用する際、USB の Vender ID と Product ID を変更する必要があります。詳細は「6. 使用方法」および USB Peripheral Communication Device Class Driver、USB Basic Firmware のアプリケーションノートを参照してください。

8.5 固定ベクタテーブルの割り込み

サンプルコードは、固定ベクタテーブルに配置された割り込みの内、リセットのみに対応しています。それ以外の固定ベクタテーブルの割り込みを使用する場合は、サンプルコードを変更してご使用ください。

8.6 ターゲットプログラムのリセットベクタ

サンプルコードを使用して書き込むターゲットプログラムの実行開始位置は、ターゲットリセットベクタ(FFFD FFFCh)の値で決定されます。したがって、ターゲットプログラムは、リセットベクタが FFFD FFFCh に配置されるように設定してください。詳細は「5.3. ターゲットプログラムの実行開始位置」を参照してください。

また、ターゲットプログラムの例については「7. ターゲットプログラムの例」を参照してください。

8.7 Motorola S フォーマット

サンプルコードが対応している Motorola S フォーマットは S0、S3、S7 フォーマットのみとなります。また、アドレスは昇順にのみ対応しています。降順またはアドレスが前後する形の mot ファイルは送信しないでください。

8.8　while(1)の処理

サンプルコードでは、USB への送信バッファがオーバフローした場合、while(1)でデッドロックします。

8.9　USB 通信中におけるプログラムの停止

PC 上のターミナルソフトとマイコンをつないだ状態で、マイコンをリセットした場合、再度マイコンを実行しても正常に通信できない可能性があります。その場合、ターミナルソフトを一度閉じた後、改めてマイコンと PC を接続し直してください。

8.10　エンディアン

本サンプルコードはリトルエンディアンのみの対応となります。

8.11　RX 用シンプルフラッシュ API からの変更点

サンプルコードは、RX 用シンプルフラッシュ API のプログラムを流用しています。RX 用シンプルフラッシュ API の仕様については RX 用シンプルフラッシュ API のアプリケーションノートを参照してください。以下に変更箇所を示します。

RX 用のシンプルフラッシュ API から変更したファイルは“r_flash_api_rx600_config.h”、“r_bsp_config.h”です。

・ ファイル“r_flash_api_rx600_config.h”内の変更箇所：

① フラッシュ書き込み/消去中の割り込みによる ROM アクセスを防ぐため、フラッシュ書き込み/消去時のプロセッサステータスワード(PSW)のプロセッサ割り込み優先レベル(IPL)を、以下のマクロ定義にて指定した値に変更します。本アプリケーションノートでは“5”に設定しています。

マクロ定義：#define FLASH_READY_IPL 5

② シンプルフラッシュ API の設定を変更しています。

変更前：//#define FLASH_API_RX_CFG_FLASH_TO_FLASH
#define FLASH_API_RX_CFG_IGNORE_LOCK_BITS
#define FLASH_API_RX_CFG_COPY_CODE_BY_API

変更後：#define FLASH_API_RX_CFG_FLASH_TO_FLASH
//#define FLASH_API_RX_CFG_IGNORE_LOCK_BITS
//#define FLASH_API_RX_CFG_COPY_CODE_BY_API

・ ファイル“r_bsp_config.h”内の変更箇所：

(ア) シンプルフラッシュ API の r_bsp/board/rskrx63n フォルダに格納されているファイルを使用しています。

(イ) シンプルフラッシュ API の設定を変更しています。

変更前：#define BSP_CFG_PCKA_DIV (4)
#define BSP_CFG_IEBCK_DIV (8)
#define BSP_CFG_BCLK_OUTPUT (0)

変更後：#define BSP_CFG_PCKA_DIV (2)
#define BSP_CFG_IEBCK_DIV (2)
#define BSP_CFG_BCLK_OUTPUT (2)

8.12　USB Peripheral Communication Device Class Driver からの変更点

サンプルコードは USB Peripheral Communication Device Class Driver の ANSI インタフェースを使用したプログラムを流用しています。USB Peripheral Communication Device Class Driver の仕様については USB Peripheral Communication Device ClassDriver および USB Basic Firmware のアプリケーションノートを参照してください。

8.12.1　変更箇所

USB Peripheral Communication Device Class Driver から変更したファイルは"r_usb_pcdc_apl.c"、"dbsct_pcdc.c"、"rx_mcu.c"の三つです。

・ ファイル"r_usb_pcdc_apl.c"内の変更箇所：

① インクルードファイルを追加しています。

追加：#include "r_Flash_main.h"

#include " r_Flash_buff.h"

②上記以外の変更箇所は、#ifdef R_FLASH_USB で示しています。

・ ファイル"dbsct_pcdc.c"内の変更箇所：

コメント"// Flash table"で示しています。

・ファイル"rx_mcu.c"内の変更箇所：

① インクルードファイルを追加しています。

追加：#include "r_init_clock.h"

#include "r_init_non_existent_port.h"

#include "r_init_stop_module.h"

② CPU の初期設定(usb_cpu_McuInitialize 関数)で、RX63N グループ、RX631グループ 初期設定例の関数(R_INIT_StopModule、R_INIT_NonExistentPort、R_INIT_Clock)を使用するように変更しています。

8.12.2　追加ファイル

USB Peripheral Communication Device Class Driver に追加したファイルについては「5.7　ファイル構成」を参照してください。

8.12.3　追加セクション

表 8.1に追加セクションを示します。

表8.1　追加セクション

セクション名	概要
B_flash_api_sec	RAM 上で動作する Flash 書き換えコードの変数のセクション
R_flash_api_sec	
RPFRAM	RAM 上で動作する Flash 書き換えコード用セクション
TRGT_DMMY_FIXEDVECT	ターゲットプログラム固定ベクタ用セクション

8.12.4　インクルードファイルディレクトリ

"WorkSpace¥FLASH"、および"WorkSpace¥r_bsp"をインクルードファイルディレクトリに追加しています。

8.12.5　リンカの設定

ROM から RAM へ MAP するリンカの設定を追加しています。

- Rom"PFRAM"を RPFRAM へ MAP しています。
- Rom"D_flash_api_sec"を"R_flash_api_sec"へ MAP しています。

8.13　RX63N グループ、RX631 グループ初期設定例からの変更点

サンプルコードは、RX63N グループ、RX631グループ初期設定例のプログラムを流用しています。RX63N グループ、RX631グループ初期設定例の仕様についてはRX63N グループ、RX631グループ初期設定例のアプリケーションノートを参照してください。以下に変更箇所を示します。

RX63N グループ、RX631グループ初期設定例から変更したファイルは“r_init_clock.c”です。

① BCLK の分周比を 4 分周から 8 分周に変更しています。

変更前：
```
SYSTEM.SCKCR.LONG = 0x21C21211;
while (0x21C21211 != SYSTEM.SCKCR.LONG)
```
変更後：
```
SYSTEM.SCKCR.LONG = 0x21C31211;
while (0x21C31211 != SYSTEM.SCKCR.LONG)
```

② USB クロックを未使用から 4 分周に変更しています。

変更前：
```
SYSTEM.SCKCR2.WORD = 0x0012;
```
変更後：
```
SYSTEM.SCKCR2.WORD = 0x0032;
```

③ BCLK 端子出力を分周なしから 2 分周に変更しています。

変更前：
```
SYSTEM.BCKCR.BYTE = 0x00;
while (0x00 != SYSTEM.BCKCR.BYTE)
```
変更後：
```
SYSTEM.BCKCR.BYTE = 0x01;
while (0x01 != SYSTEM.BCKCR.BYTE)
```

9. 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。

10. 参考ドキュメント

- ユーザーズマニュアル：ハードウェア
  RX63N グループ、RX631グループ ユーザーズマニュアル ハードウェア編 Rev.1.70（R01UH0040JJ）
  （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 C/C++コンパイラパッケージ V.1.01 ユーザーズマニュアル Rev.1.00（R20UT0570JJ）
  （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。）

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
  http://japan.renesas.com
- お問合せ先
  http://japan.renesas.com/contact/

改訂記録	RX63N グループ、RX631グループ　アプリケーションノート USB ペリフェラル CDC によるフラッシュブートローダ

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.01.06	—	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. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。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。

4. クロックについて

【注意】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。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの責任を負うものではありません。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. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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）、交通用信号機器、
   防災・防犯装置、各種安全装置等
   当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、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。お客様が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. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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。

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その総株主の議決権の過半数を直接または間接に保有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。

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

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

ルネサス エレクトロニクス株式会社　〒100-0004　千代田区大手町2-6-2（日本ビル）

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