

RX62N、RX621 グループ

USB ホストフラッシュブートローダ

R01AN0892JJ0100
Rev.1.00
2012.03.28

要旨

本アプリケーションノートでは、RX62N、RX621 グループの USB2.0 ホスト / ファンクションモジュールを使用し、シングルチップモードでホスト動作させ、USB 経由で内蔵フラッシュメモリの書き換えを行う「USB ホストフラッシュブートローダ」について説明します。

なお、本アプリケーションノートでは以下のアプリケーションノートのサンプルコードおよびドライバを使用しています。

- 内蔵フラッシュメモリの消去 / 書き込み :
「RX600 シリーズ RX600 用のシンプルフラッシュ API」 Rev.2.20 (R01AN0544JU0220)
- USB 通信 :
「ルネサス USB デバイス USB Basic Firmware」 Rev.1.10 (R01AN0512JJ0110)
「ルネサス USB デバイス USB Host Mass Storage Class Driver」 Rev.1.10 (R01AN0513JJ0110)
- FAT ファイルシステム :
「M3S-TFAT-Tiny : FAT ファイルシステムソフトウェア」 Rev.1.00 (R20AN0038JJ0100)

本アプリケーションノートの特長を以下に示します。

- USB メモリに格納した S タイプフォーマットのプログラムを書き込み可能。
S タイプフォーマットのプログラムを格納した、USB マスストレージデバイス (USB メモリ) の接続を認識すると、マイコン内蔵フラッシュメモリを消去し、プログラムを書き込みます。
- 書き込んだプログラムの実行が可能。
マイコン内蔵フラッシュメモリに書き込んだ S タイプフォーマットのプログラムを実行することができます。
- USB 仕様
USB2.0 規格のフルスピード転送に対応しています。
USB マスストレージクラスの Bulk-Only Transport (BOT) に対応しています。
USB マスストレージサブクラスの SFF-8070i (ATAPI) および SCSI に対応しています。

対象デバイス

RX62N、RX621 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あわせて変更し、十分評価してください。

目次

1. 仕様	3
2. 動作確認条件	4
3. 関連アプリケーションノート	4
4. ハードウェア説明	5
5. ソフトウェア説明	6
6. ダウンロードコードの例	42
7. S タイプフォーマット	43
8. 注意事項	45
9. サンプルコード	50
10. 参考ドキュメント	50

1. 仕様

本アプリケーションノートのサンプルコードは、RX62N-RSK 上で動作します。

RX62N-RSK に接続されたスイッチ (SW3) を押ししていない状態でリセットを解除すると、接続された USB メモリ内の S タイプフォーマットのプログラム (ファイル名: download.mot) を内蔵フラッシュメモリに書き込みます。書き込み終了後、SW3 を押した状態でリセットを解除すると、内蔵フラッシュメモリに書き込んだプログラム (以降: ダウンロードコード) を実行します。

なお、サンプルコードが書き換える領域はユーザマツの一部のみとなります。サンプルコードが使用している領域の書き換えは行いません。詳細は「5.1 動作概要」を参照ください。

内蔵フラッシュメモリへの書き込み状況および結果は、RX62N RSK に接続された、LED、LCD に表示します。表示の内容は「5.5. サンプルコードの LED、LCD 表示」を参照ください。

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	USB メモリとの通信

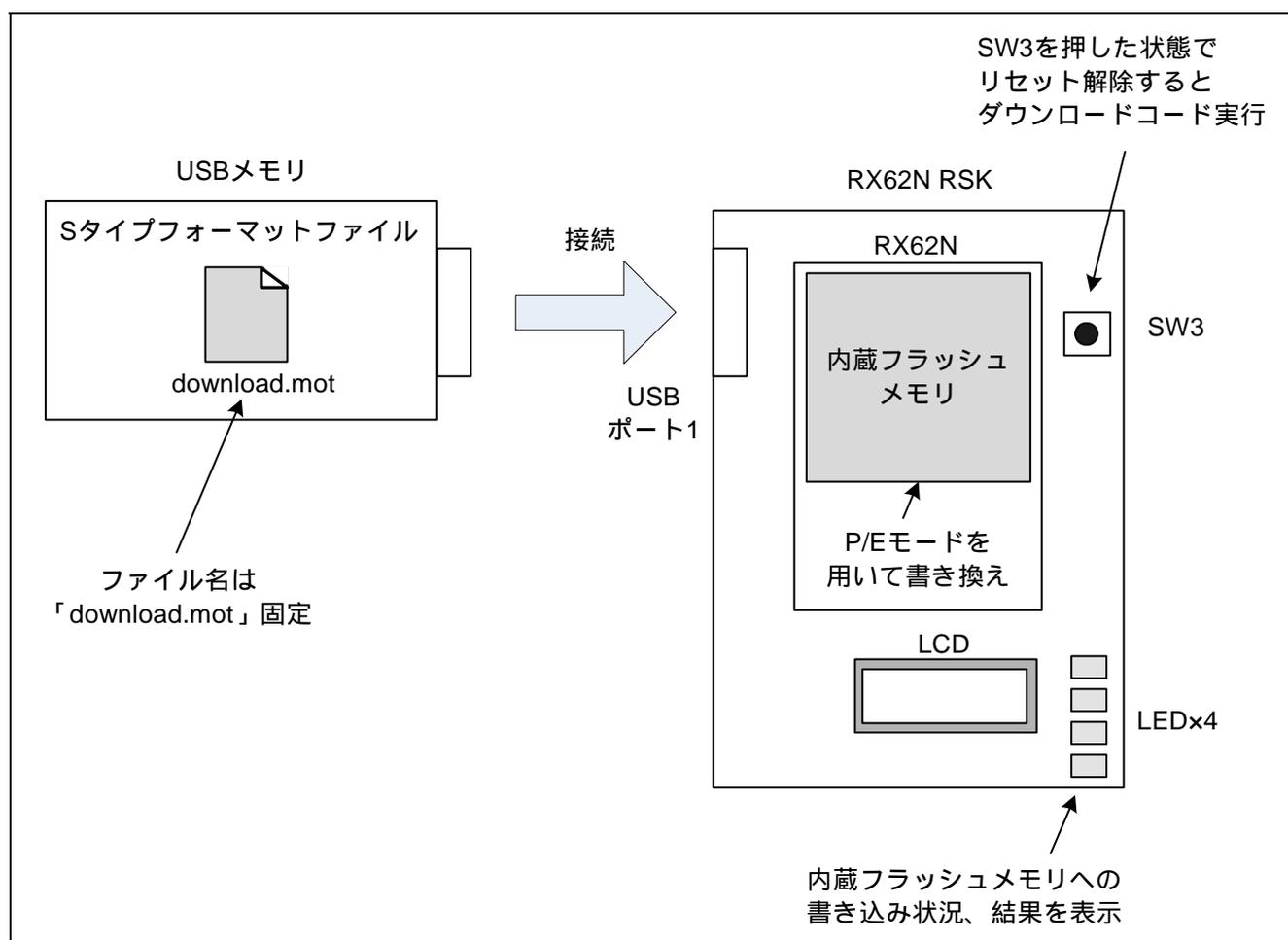


図1.1 使用例

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	RX62N グループ (R5F562N8BDBG)
使用デバイス	R5F562N8BDBG
動作周波数	<ul style="list-style-type: none"> • EXTAL : 12MHz • ICLK : 48MHz • BCLK : 48MHz • PCLK : 24MHz • UCLK : 48MHz
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 4.09.00.007
C コンパイラ	ルネサスエレクトロニクス製 RX Standard Toolchain Version 1.1.0.0 '-cpu=rx600 -include="\$ (WORKSPDIR)¥WorkSpace¥USBCSTDFW¥include", " \$ (WORKSPDIR)¥WorkSpace¥USBSTDFW¥include", " \$ (WORKSPDIR)¥WorkSpace¥USB2STDFW¥include", " \$ (WORKSPDIR)¥WorkSpace¥SmpIMain¥APL", " \$ (WORKSPDIR)¥WorkSpace¥HwResourceForUSB", " \$ (WORKSPDIR)¥WorkSpace¥MSCCFW¥include", " \$ (WORKSPDIR)¥WorkSpace¥MSC2FW¥TFAT¥lib_src", " \$ (WORKSPDIR)¥WorkSpace¥MSC2FW¥include", " \$ (WORKSPDIR)¥WorkSpace¥FLASH" -define=USB2_FUNCSEL_PP=USBC_HOST_PP, USBC_FW_PP=USBC_FW_NONOS_PP,USBC_TFAT_USE_PP=1 -output=obj="\$ (CONFIGDIR)¥\$ (FILELEAF).obj" -debug -nostuff -optimize=0 -nologo
動作モード	シングルチップモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Development Tools (製品型名 : R0K5562N0S000BE) に同梱されている RSK+ RX62N を使用

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。あせて参照してください。

- ルネサス USB デバイス USB Basic Firmware Rev.1.10 (R01AN0512JJ)
- ルネサス USB デバイス USB Host Mass Storage Class Driver Rev.1.10 (R01AN0513JJ)
- RX600 シリーズ RX600 用のシンプルフラッシュ API Rev.2.20 (R01AN0544JU)
- M3S-TFAT-Tiny : FAT ファイルシステムソフトウェア Rev.1.00 (R20AN0038JJ)

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。

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_VBUSEN-B	出力	ポート 1 外部電源チップへの VBUS (5V) の供給許可信号
USB1_OVRCURA	入力	ポート 1 外部オーバカレント検出信号を接続します。また OTG 電源 チップとの接続時には VBUS コンパレータ信号を接続します
P07	入力	サンプルコードのモード選択用端子
P02	出力	LED 接続端子
P03	出力	LED 接続端子
P05	出力	LED 接続端子
P34	出力	LED 接続端子
P84	出力	LCD モジュール制御用端子
P85	出力	LCD モジュール制御用端子
P94	出力	LCD モジュール制御用端子
P95	出力	LCD モジュール制御用端子
P96	出力	LCD モジュール制御用端子
P97	出力	LCD モジュール制御用端子

5. ソフトウェア説明

5.1 動作概要

5.1.1 リセット解除後の動作

サンプルコードは、マイコンのリセット解除後に SW3 (P07) の状態をチェックします。このとき SW3 が押していない状態 (P07 = H) であれば、USB ホストフラッシュブートローダを実行し、USB 経由で内蔵フラッシュメモリを書き換えます。また、SW3 を押した状態 (P07 = L) であればダウンロードコードを実行します。図 5.1 にリセット解除後の動作を示します。

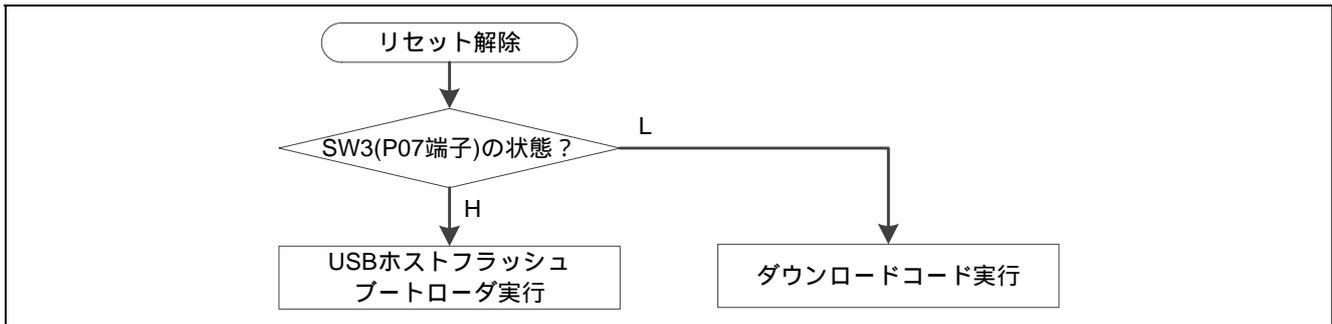


図5.1 リセット解除後の動作

5.1.2 書き換え対象

USB ホストフラッシュブートローダが書き換える対象はユーザマットの一部分 (以降: ダウンロードエリア) のみとなります。サンプルコードが使用している領域 (FFFE4000h ~ FFFFFFFFh) の書き換えは行いません。図 5.2 にメモリ配置を示します。

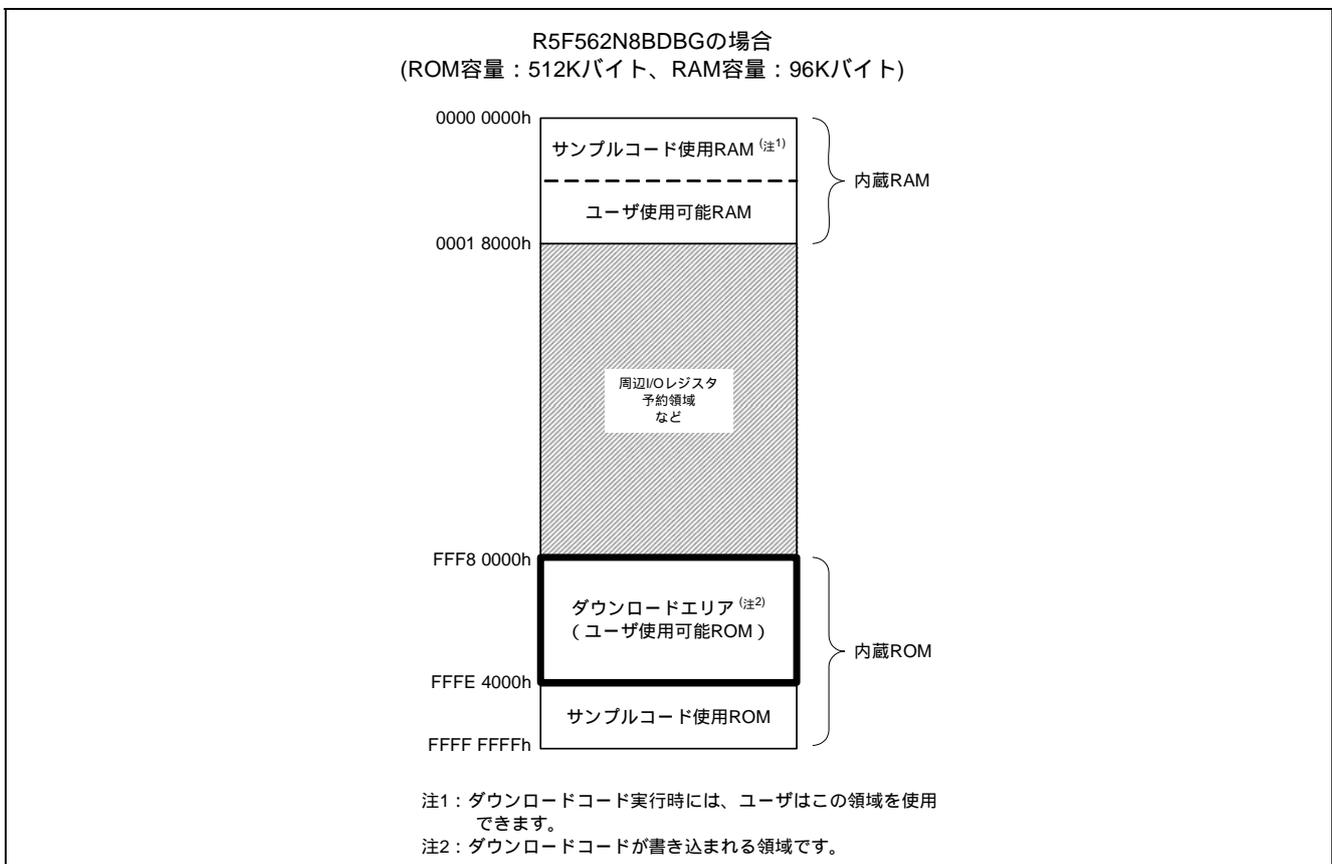


図5.2 メモリ配置

5.1.3 ダウンロードエリアの書き換え

USB ホストフラッシュブートローダは、以下の手順でダウンロードエリアの書き換えを行います。図 5.3 にダウンロードエリアの書き換え動作を示します。

USB デバイスの接続を監視します。

USB の接続を検出すると、接続されたデバイスの情報を取得し、アクセス可能か判定します。

接続された USB デバイス (USB メモリ) へのアクセスが可能ならば、ファイル名「download.mot」の S タイプフォーマットファイルを検索します。

ファイル名「download.mot」の S タイプフォーマットファイルを確認すると、ダウンロードエリアを消去します。

ダウンロードエリアの消去後、USB メモリ内の S タイプフォーマットファイルから 2048byte 分データを読み出し、内蔵 RAM に格納します。

内蔵 RAM へのデータ格納後、データの解析処理を行い、256byte 単位でダウンロードエリアに書き込みます。

全データの書き込みを完了するまで、の処理を繰り返します。

なお、S タイプフォーマットファイルの終端は、エンドレコード (S7、S8、S9 レコード) で判定しています。

正常にダウンロードエリアの消去 / 書き込み処理を完了すると、I/O ポートに接続された LED、LCD で正常終了を知らせます。

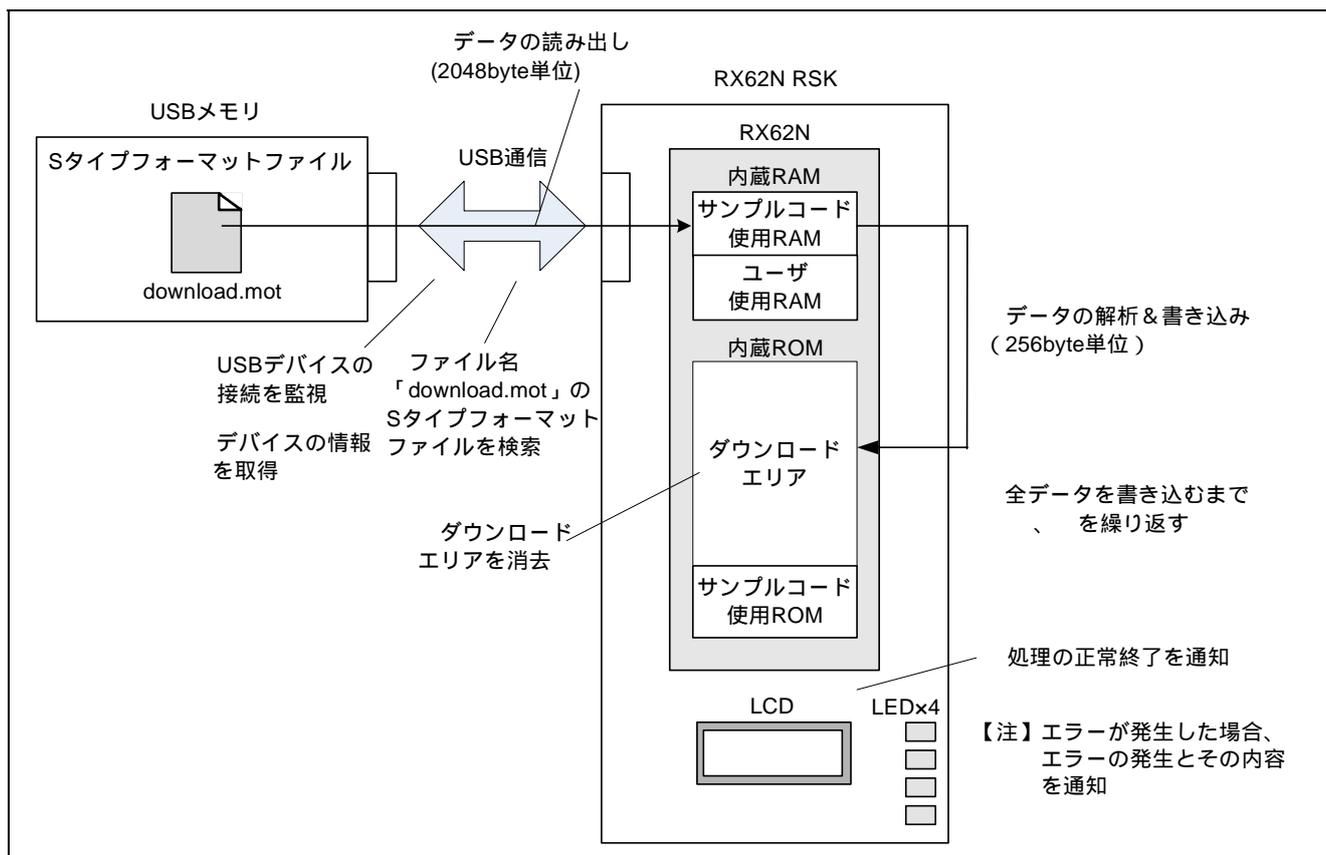


図5.3 ダウンロードエリアの書き換え動作

【注】 サンプルコード実行中にエラーが発生した場合は、LED、LCD でそのエラー内容を通知し、処理を終了します。エラーの発生条件および LED、LCD 表示については「5.5. サンプルコードの LED、LCD 表示」を参照ください。

5.2 ダウンロードコードの実行開始位置

サンプルコードは、マイコンのリセット解除後 SW3 の状態が L である場合、ダウンロードコードを実行します。このときサンプルコードは、アドレス"FFFE 3FFCh"に書かれているアドレス番地から実行します。つまり、ダウンロードコードにとってのリセットベクタが"FFFE 3FFCh"になります。ダウンロードコードでは、あらかじめ"FFFE 3FFCh"に開始アドレスが格納されるようにしておいてください。

図 5.4にダウンロードコードのリセットベクタを示します。

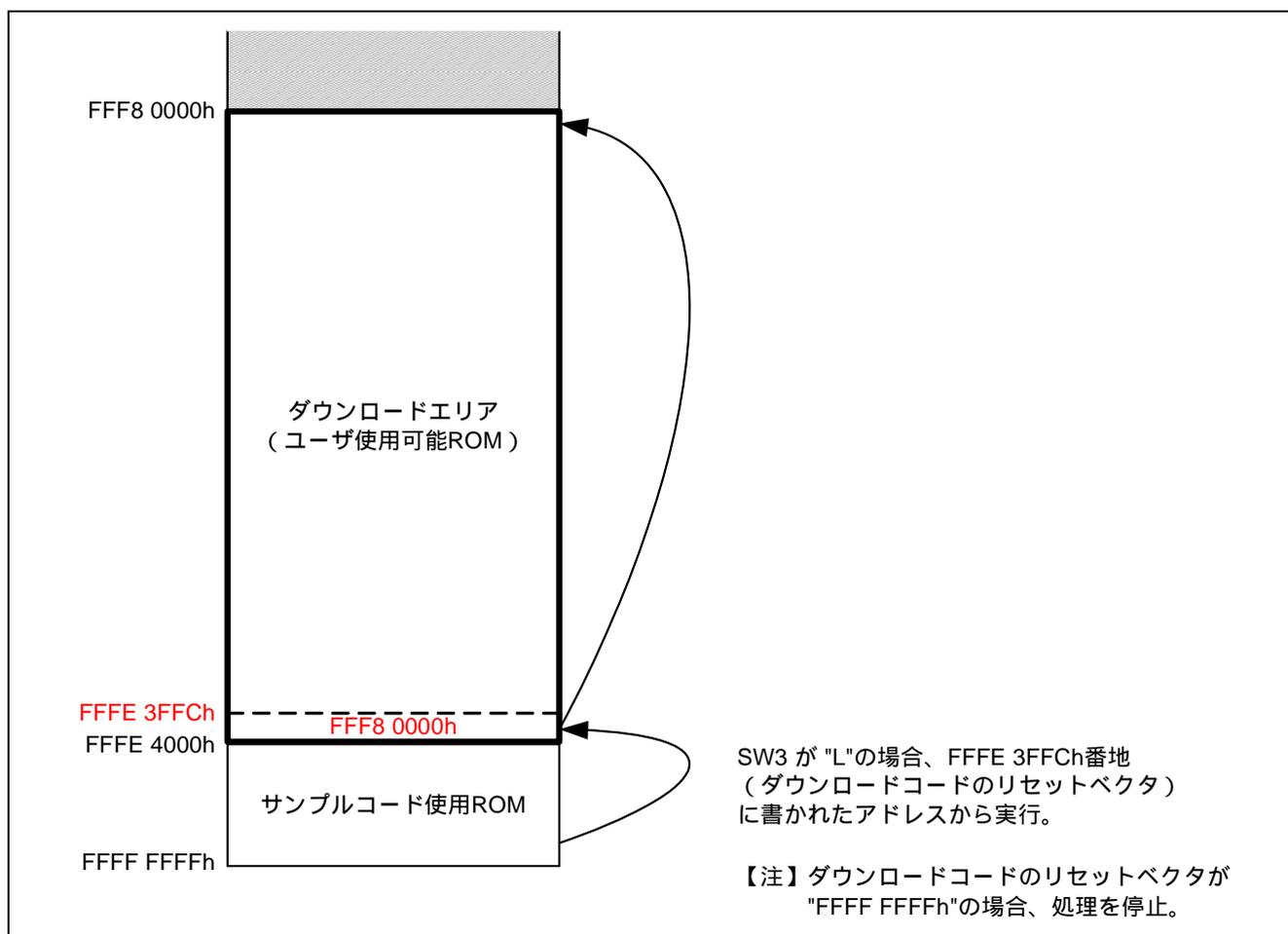


図5.4 ダウンロードコードのリセットベクタ

【注】 ダウンロードコードのリセットベクタに何も書かれていない場合 (ダウンロードコードのリセットベクタが"FFFF FFFFh"の場合) は、while(1)で自番地ループを行い処理を停止します。

5.3 サンプルコードのソフトウェア構成

サンプルコードでは、USB との通信に、

「ルネサス USB デバイス USB Basic Firmware」と

「ルネサス USB デバイス USB Host Mass Storage Class Driver」を、

FAT ファイルシステムとして

「M3S-TFAT-Tiny : FAT ファイルシステムソフトウェア」

を使用しています。

また、内蔵フラッシュメモリの消去処理および書き込み処理には、

「RX600 シリーズ RX600 用のシンプルフラッシュ API」を、

を使用しています。

図 5.5 にサンプルコードのソフトウェア構成を、表 5.1 にソフトウェアの概要を示します。

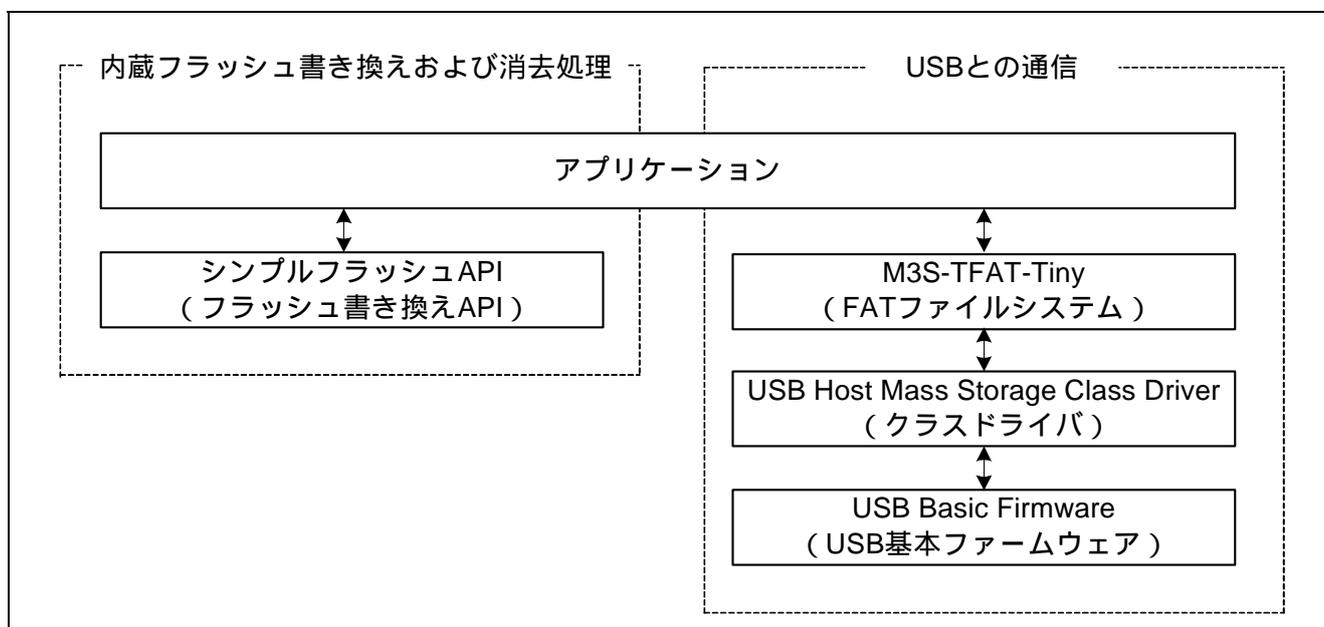


図5.5 サンプルコードのソフトウェア構成

表5.1 ソフトウェアの概要

モジュール名	概要
アプリケーション	FAT ライブラリ関数を使用し、USB メモリ内の S タイプフォーマットのプログラムを読み出します。また、シンプルフラッシュ API 関数を使用して、内蔵フラッシュメモリの消去、書き換えを行います。
RX600 用のシンプルフラッシュ API	内蔵フラッシュメモリの消去 / 書き換えを行う API です。
M3S-TFAT-Tiny	FAT12、FAT16 に対応した FAT ファイルシステムです。
USB Host Mass Storage Class Driver	USB マスストレージクラスの Bulk-Only Transport (BOT) プロトコルに対応したクラスドライバです。
USB Basic Firmware	USB インタフェース制御用のサンプルプログラムです。

5.4 書き込み時のデータの流れ

ダウンロードコード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。

USB ドライバにより取得したデータを受信用リングバッファに転送します。

S タイプフォーマットの 1 レコードを S タイプフォーマット用バッファ (ASCII) にコピーします。

S タイプ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、S タイプフォーマット用バッファ (Binary) に格納します。

なお、本アプリケーションノートで対応する S タイプフォーマットの仕様は、「7. S タイプフォーマット」を参照ください。

書き込み用バッファにデータを格納します。

RX62N、RX621 のユーザマットへの書き込み単位は 256byte となっています。そのため、サンプルコードでは、ユーザマットへの 1 回の書き込みサイズ 256byte とし、書き込みバッファに格納される書き込みデータの合計が 256byte になるまで ~ の処理を繰り返しています。書き込みデータの合計が 256byte を超えてしまった場合は、超えた分のデータを一時保存しておき、次の 256byte 書き込み時に使用しています。

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。

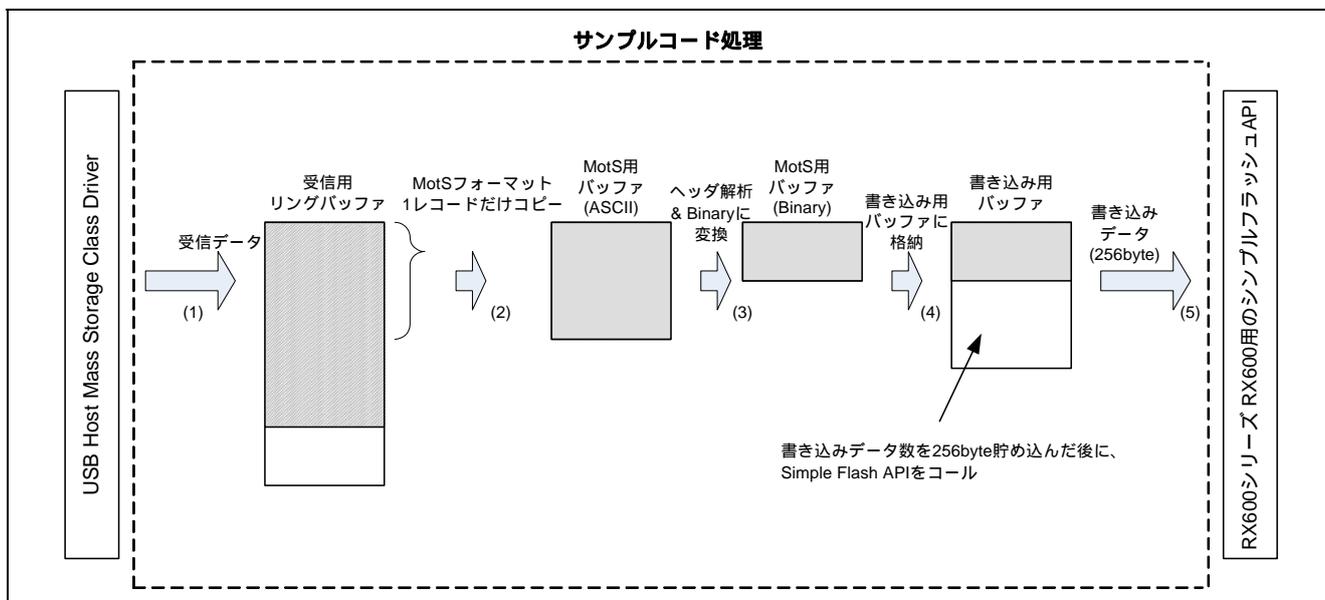


図5.6 書き込み時のデータの流れ

図 5.7に書き込み時のデータ構造を示します。

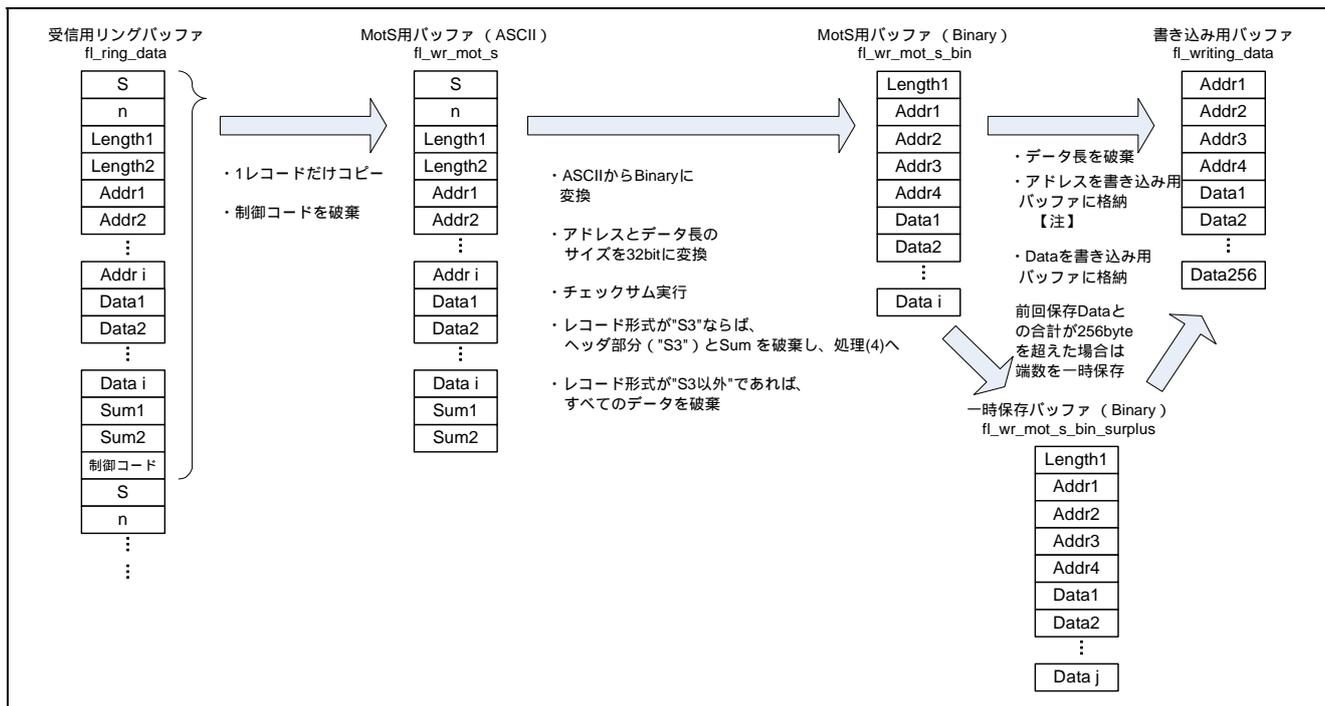


図5.7 書き込み時のデータ構造

【注】 RX62N、RX621 グループの内蔵フラッシュメモリは、書き込み対象の先頭アドレスを 256byte 境界にそろえる必要があるため、サンプルコードでは書き込み用バッファにアドレスを格納する際、書き込みアドレスの先頭が 256byte 境界となるよう処理を行っています。処理の詳細は、「5.12.11 ダウンロードエリアへの書き込みデータ作成」のフローチャートを参照ください。

5.5 サンプルコードの LED、LCD 表示

サンプルコードではプログラムの進行状況および結果を RX62N RSK に接続された LED、LCD に表示します。表 5.2 にサンプルコードの LED 表示一覧を示します。

表5.2 サンプルコードの LED 表示一覧

: 点灯、 : 消灯

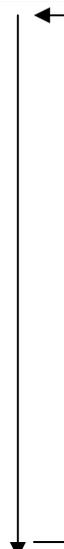
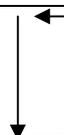
LED 表示					順序	内容
LED3	LED2	LED1	LED0			
					ダウンロード処理の実行中、LED は 2 進カウントアップ表示を行います。 LED 表示は、1 ブロック書き込み終了 (16K byte 書き込み) ごとに更新されます。	
					サンプルコードが正常終了した場合、LED はシフト表示を行います。LED 表示は、500ms 経過ごとに更新されます。	
					サンプルコードが異常終了した場合、LED はブリンク表示を行います。LED 表示は、500ms 経過ごとに更新されます。	

表 5.3にサンプルコードの LCD 表示一覧を示します。

表5.3 サンプルコードの LCD 表示一覧

LCD 表示	内容
FLASH BOOT	リセット解除後、「USB ホストフラッシュブートローダ」が実行された場合に表示されます。
DETACH	リセット解除後、USB が接続されていない場合に表示されます。*
ATTACH	リセット解除後、USB が接続された場合に表示されます。*
PROGRAM UPDATE..	ダウンロード処理の実行中に表示されます。
ERROR!! D OPEN	アクセス可能なドライブが検出されなかった場合に表示されます。(ドライブオープンエラー)
ERROR!! D MOUNT	ドライブのマウント処理に失敗した場合に表示されます。(ドライブマウントエラー)
ERROR!! F OPEN	ファイルオープン処理に失敗した場合に表示されます。(ファイルオープンエラー)
ERROR!! F READ	ファイルに読み出し処理に失敗した場合に表示されます。(ファイルリードエラー)
ERROR!! F CLOSE	ファイルのクローズ処理に失敗した場合に表示されます。(ファイルクローズエラー)
ERROR!! SUM	「7S タイプフォーマット」を参照ください。(チェックサムエラー)
ERROR!! MOTS	「7S タイプフォーマット」を参照ください。(フォーマットエラー)
ERROR!! ERASE	ダウンロードエリアの消去に失敗した場合に表示されます。(イレースエラー)
ERROR!! WRITE	ダウンロードエリアへの書き込みに失敗した場合に表示されます。(書き込みエラー)
ERROR!! ADDRESS	「7S タイプフォーマット」を参照ください。(アドレスエラー)
ERROR!! VERIFY	ダウンロードエリアへ書き込んだデータのベリファイ結果に異常があった場合に表示されます。(ベリファイエラー)
ERROR!! F END	FAT ライブラリでファイルの終端を検出したにもかかわらず、S タイプフォーマットのエンドコードを受信しなかった場合に表示されます。(ファイルエンドエラー)
ERROR!! ILL DET	ドライブオープン処理もしくは、ダウンロード処理実行中に USB のデタッチを検出した場合に表示されます。(イリーガルデタッチエラー)

【注】 * USB 未接続時の「DETACH」表示、USB 接続時の「ATTACH」表示は、「USB Host Mass Storage Class Driver」の仕様です。

5.6 必要メモリサイズ

表 5.4に必要メモリサイズを示します。

表5.4 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	105028byte	サンプルコードは、FFFE4000h ~ FFFFFFFFh に配置されているため書き換え可能な ROM 容量(ダウンロードエリアの容量)は「全体の ROM 容量 - 114688byte」となります。
RAM	28476byte	ダウンロードコード実行時、ユーザはこの領域を使用できません。

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。

5.7 ファイル構成

表 5.5にサンプルコードで使用するファイルを示します。

表5.5 ファイル構成

ファイル名	概要	備考
r_flash_api_rx600.c	RX600 シリーズ RX600 用のシンプルフラッシュ API のプログラム	詳細は RX600 シリーズ RX600 用の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_rx600.h	RX600 シリーズ RX600 用のシンプルフラッシュ API のプログラムの外部参照用インクルードヘッダ	詳細は RX600 シリーズ RX600 用の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_rx600_private.h	RX600 シリーズ RX600 用のシンプルフラッシュ API のプログラムの外部参照用インクルードヘッダ	詳細は RX600 シリーズ RX600 用の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_rx600_config.h	RX600 シリーズ RX600 用のシンプルフラッシュ API のパラメータ設定用インクルードヘッダ	詳細は RX600 シリーズ RX600 用の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。
mcu_info.h	RX600 シリーズ RX600 用のシンプルフラッシュ API のパラメータ設定用インクルードヘッダ	詳細は RX600 シリーズ RX600 用の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_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インクルードヘッダ	
r_Flash_buff.c	USB との受信バッファ関連処理	
r_Flash_buff.h	USB との受信バッファ関連処理の外部参照用インクルードヘッダ	
TrgtPrgDmmy.c	ダウンロードコード用の領域を確保するためのダミープログラム	
その他ファイル	USB Host Mass Storage Class Driver のプログラム	詳細はルネサス USB デバイス USB Host Mass Storage 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 5.6にサンプルコードで使用する定数を示します。

表5.6 サンプルコードで使用する定数

定数名	設定値	内容
FL_RINGBUFF_SIZE	4096	USBからのデータ受信用リングバッファサイズ
FL_MOTS_ADDR_SIZE	4	Sタイプフォーマットデータのアドレスバッファサイズ
FL_MOTS_SUM_SIZE	1	Sタイプフォーマットデータのチェックサムバッファサイズ
FL_START_BLOCK_NUM	10	ダウンロードエリアの最初のブロック
FL_END_BLOCK_NUM	37	ダウンロードエリアの最後のブロック
FL_START_WRITE_ADDRESS	FFF80000h	ダウンロードエリアの先頭アドレス
FL_END_WRITE_ADDRESS	FFFE3FFFh	ダウンロードエリアの最後尾アドレス
FL_USB_RCV_BLANK_SIZE	2048	リングバッファの補充可能サイズ
FL_MODE_ENTRY_WAIT_LCD_PERIOD	100000	モードエントリ時の待ち時間
FL_UPDATE_WAIT_LED_PERIOD	64	ダウンロード処理時のLED表示間隔の時間
FL_ERROR_WAIT_LED_PERIOD	500	エラー処理時のLED表示間隔の時間
FL_DONE_WAIT_LED_PERIOD	500	正常終了処理時のLED表示間隔の時間
FL_UPDATE_FILE_NAME	"download.mot"	ダウンロードコードのファイル名
FL_PORT_MDE	PORT0.PORT.BIT.B7	SW3 接続ポート (P07) の PORT レジスタ
FL_DDR_MDE	PORT0.DDR.BIT.B7	SW3 接続ポート (P07) の DDR レジスタ
FL_ICR_MDE	PORT0.ICR.BIT.B7	SW3 接続ポート (P07) の ICR レジスタ

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。

```
/* 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 関数一覧

表 5.7に関数を示します。ただし、USB Host Mass Storage Class Driver、シンプルフラッシュ API、FAT ファイルシステムソフトウェアで使用しているものは除きます。

表5.7 関数

関数名	概要
R_FI_Mode_Entry	モード選択
R_FI_Flash_Update	書き換え処理のメインとなる関数
R_FI_EraseTrgtArea	消去処理
R_FI_Ers_EraseFlash	ダウンロードエリア消去
R_FI_PrgramTrgtArea	書き込み処理
R_FI_Prg_PrgramTrgtArea	ダウンロードエリアへの書き込み
R_FI_Prg_StoreMotS	S タイプフォーマットデータ格納
R_FI_Prg_ProcessForMotS_data	S タイプフォーマットデータのヘッダ解析、Binary 変換、書き込み
R_FI_Prg_MotS_AsciiToBinary	S タイプフォーマットデータの ASCII -Binary 変換
R_FI_Prg_MakeWriteData	ダウンロードエリアへの書き込みデータ作成
R_FI_Prg_WriteData	ダウンロードエリアへの書き込み
R_FI_Prg_ClearMotSVariables	S タイプ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_Error	エラー処理
R_FI_RingCheckBlank	受信用リングバッファの空き容量確認
R_FI_RingEnQueue	USB 受信データ格納バッファへのデータ格納
R_FI_RingDeQueue	USB 受信データ格納バッファからのデータ読み出し
R_FI_RingCheck	USB 受信データ格納バッファデータ数確認
R_FI_AsciiToHexByte	ASCII コードを Binary データへ変換

5.11 関数仕様

サンプルコードの関数仕様を示します。

R_FI_Mode_Entry	
概要	モードエントリ
ヘッダ	r_Flash_main.h, iodefine.h
宣言	void R_FI_Mode_Entry(void)
説明	<ul style="list-style-type: none"> SW3の状態をチェックします。 SW3=Lであれば「USB ホストフラッシュブートローダ」実行を実行します SW3=Hであれば「ダウンロードコード」を実行します。
引数	なし
リターン値	なし
備考	

R_FI_Flash_Update	
概要	書き換え処理メイン
ヘッダ	r_Flash_main.h
宣言	void R_FI_Flash_Update (void)
説明	<ul style="list-style-type: none"> ダウンロードエリアの消去処理を行う関数をコールします。 FAT ライブラリ関数をコールし、USB メモリ内の S タイプフォーマットファイルからデータを読み出します。 S タイプフォーマットデータの解析/ダウンロードエリアの書き換え処理を行う関数をコールします。 FAT ライブラリ関数の実行に失敗した場合、エラー関数をコールします。
引数	なし
リターン値	なし
備考	

R_FI_EraseTrgtArea	
概要	消去処理
ヘッダ	なし
宣言	static void R_FI_EraseTrgtArea(void)
説明	<ul style="list-style-type: none"> ダウンロードエリアを消去する関数を呼び出します。 ダウンロードエリアの消去に失敗した場合、エラー関数をコールします。
引数	なし
リターン値	なし
備考	

R_FI_Ers_EraseFlash	
概要	ダウンロードエリア消去
ヘッダ	なし
宣言	static FI_API_SMPL_rtn_t R_FI_Ers_EraseFlash(void)
説明	<ul style="list-style-type: none"> ダウンロードエリアを消去します。
引数	なし
リターン値	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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_Fl_PrgramTrgtArea	
概要	ダウンロードエリアへの書き込み
ヘッダ	なし
宣言	static void R_Fl_PrgramTrgtArea (void)
説明	<ul style="list-style-type: none"> 書き込み処理を行う関数を呼び出します。 S タイプフォーマットのエンドレコードを受信せず、ファイル終端に達した場合、エラー関数をコールします。
引数	なし
リターン値	なし
備考	

R_Fl_Prg_PrgramTrgtArea	
概要	書き込み処理
ヘッダ	なし
宣言	static FI_API_SMPL_rtn_t R_Fl_Prg_PrgramFlash(void)
説明	<ul style="list-style-type: none"> 受信用リングバッファにデータがあった場合、S タイプフォーマットのレコード一つ分を格納する関数を呼び出します。 S タイプフォーマットのレコードを一つ分格納したら、ヘッダ解析、バイナリデータへの変換、ダウンロードエリアへの書き込みを行う関数をコールします。
引数	なし
リターン値	<ul style="list-style-type: none"> ダウンロードエリアへの書き込みが完了した場合：FLASH_API_SAMPLE_OK ダウンロードエリアへの書き込みが未完了の場合：FLASH_API_SAMPLE_NG
備考	

R_Fl_Prg_StoreMotS	
概要	S タイプ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_Fl_Prg_StoreMotS(uint8_t)
説明	<ul style="list-style-type: none"> 引数で受け取ったデータを S タイプ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	<ul style="list-style-type: none"> 第一引数：mot_data : S タイプフォーマットデータ
リターン値	<ul style="list-style-type: none"> 一つ分の S タイプ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 S タイプ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	<ul style="list-style-type: none"> 本関数は、S タイプ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_Fl_Prg_ProcessForMotS_data	
概要	S タイプフォーマットレコードのヘッダ解析、Binary 変換、書き込み
ヘッダ	なし
宣言	static void R_Fl_Prg_ProcessForMotS_data(void)
説明	<ul style="list-style-type: none"> • S タイプフォーマットのヘッダ解析を行い、Binary 変換する関数をコールします。 • 書き込みバッファヘデータを格納する関数をコールします。 • ダウンロードエリアヘデータを書き込む関数をコールします。 • S タイプフォーマットと異なるデータがあった場合、エラー関数をコールします。
引数	なし
リターン値	なし
備考	

R_Fl_Prg_MotS_AsciiToBinary	
概要	S タイプフォーマットデータ ASCII - Binary 変換
ヘッダ	なし
宣言	static Fl_API_SMPL_rtn_t R_Fl_Prg_MotS_AsciiToBinary (Fl_prg_mot_s_t *, Fl_prg_mot_s_binary_t *)
説明	<ul style="list-style-type: none"> • ASCII コードの S タイプ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。 • S タイプフォーマットと異なるデータがあった場合、エラー関数をコールします。 • チェックサムエラーが発生した場合、エラー関数をコールします。
引数	<ul style="list-style-type: none"> • 第一引数 : *tmp_mot_s : ASCII コードの S タイプ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	<ul style="list-style-type: none"> • 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_Fl_Prg_MakeWriteData	
概要	ダウンロードエリアへの書き込みデータ作成
ヘッダ	なし
宣言	static Fl_API_SMPL_rtn_t R_Fl_Prg_MakeWriteData(void)
説明	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	ダウンロード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	<ul style="list-style-type: none"> ダウンロード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、エラー関数をコールします。 ベリファイエラーが発生した場合、エラー関数をコールします。
引数	なし
リターン値	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	S タイプフォーマットデータ関連の変数クリア
ヘッダ	なし
宣言	static void R_FI_Prg_ClearMotSVariables(void)
説明	<ul style="list-style-type: none"> S タイプ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	r_Flash_main.h
宣言	void R_FI_Run_StopUSB(void)
説明	<ul style="list-style-type: none"> USB を停止します。
引数	なし
リターン値	なし
備考	

R_FI_RcvDataString	
概要	USB 受信データ格納
ヘッダ	なし
宣言	static FI_API_SMPL_rtn_t R_FI_RcvDataString(void *, uint16_t)
説明	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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_FI_Error	
概要	エラー処理
ヘッダ	r_Flash_main.h
宣言	void R_FI_Error(FI_err_tbl_num_t err_num)
説明	<ul style="list-style-type: none"> • USB 停止関数をコールします。 • LED、LCD にエラー表示を行います。
引数	<ul style="list-style-type: none"> • 第一引数 : err_num : エラー
リターン値	なし
備考	

R_FI_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_buff.h
宣言	FI_API_SMPL_rtn_t R_FI_RingCheckBlank(void)
説明	<ul style="list-style-type: none"> • 受信用リングバッファにファイルリード関数で読み出すデータ 1 回分 (2048byte) の空きがあるかを確認します。
引数	なし
リターン値	<ul style="list-style-type: none"> • 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_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	<ul style="list-style-type: none"> • 受信用リングバッファへデータを格納します。
引数	<ul style="list-style-type: none"> • 第一引数 : enq_data : 格納データ
リターン値	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 受信用リングバッファからデータを読み出します。
引数	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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_FL_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_FL_RingCheck(void)
説明	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 受信データ数を返します。
備考	

R_FL_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_FL_AsciiToHexByte(uint8_t, uint8_t)
説明	<ul style="list-style-type: none"> 2byte の ASCII コードデータを 1byte の Binary データへ変換します。
引数	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Binary に変換されたデータを返します。
備考	

5.12 フローチャート

5.12.1 USB 処理メイン

図 5.9にUSB 処理メインのフローチャートを示します。

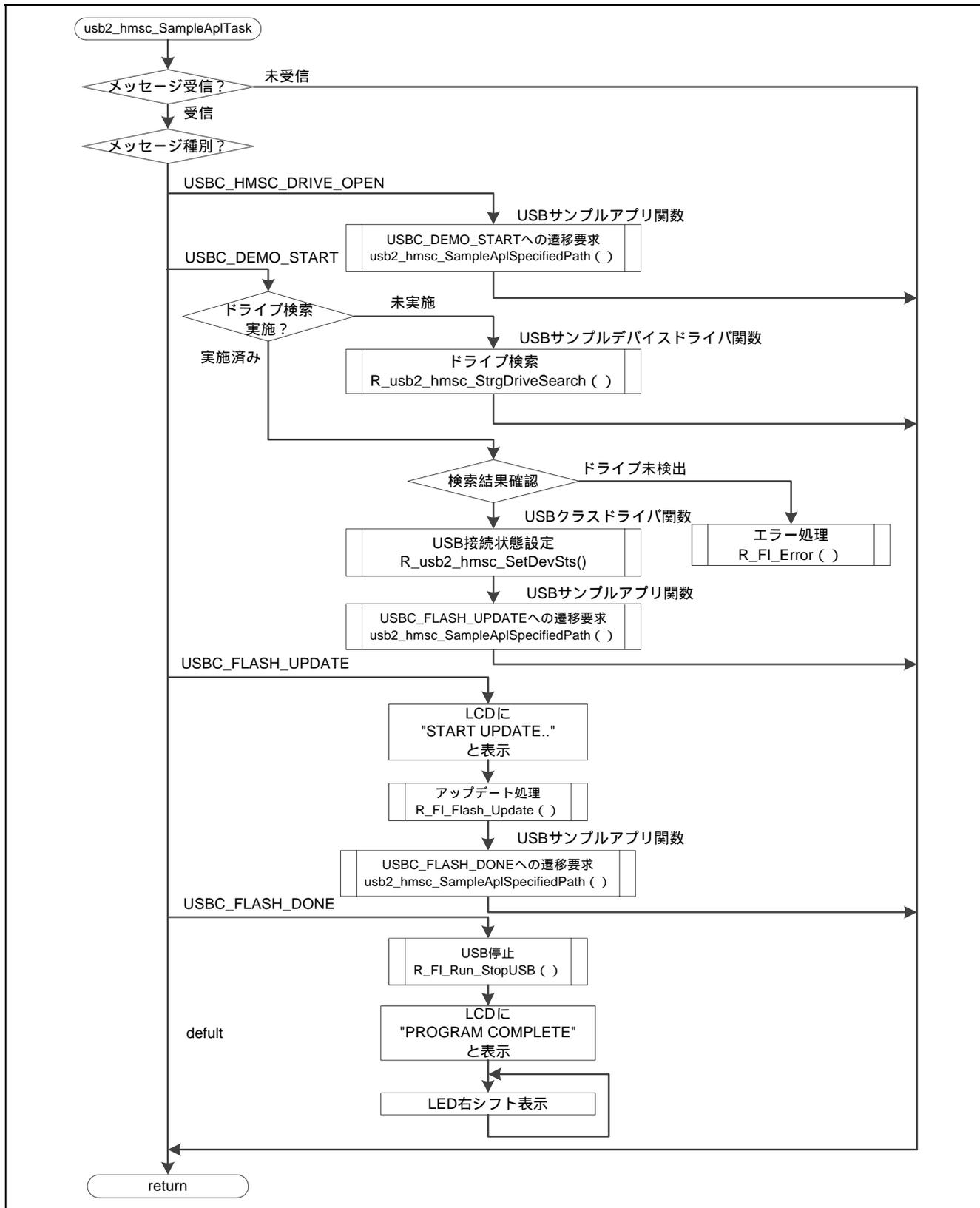


図5.9 USB 処理メイン

5.12.2 モードエントリ

図 5.10にモードエントリのフローチャートを示します。

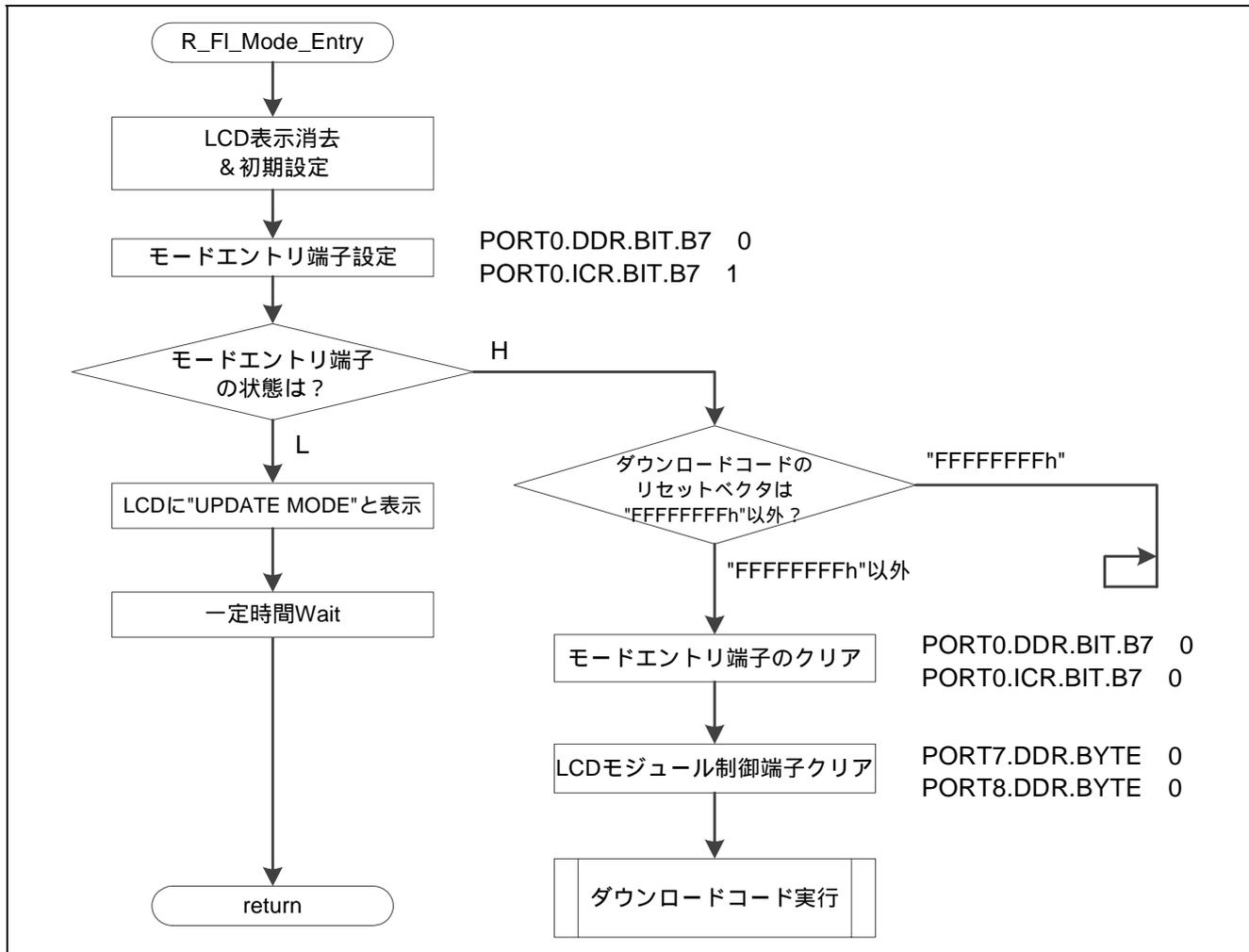


図5.10 モードエントリ

5.12.3 書き換え処理メイン

図 5.11に書き換え処理メインのフローチャートを示します。

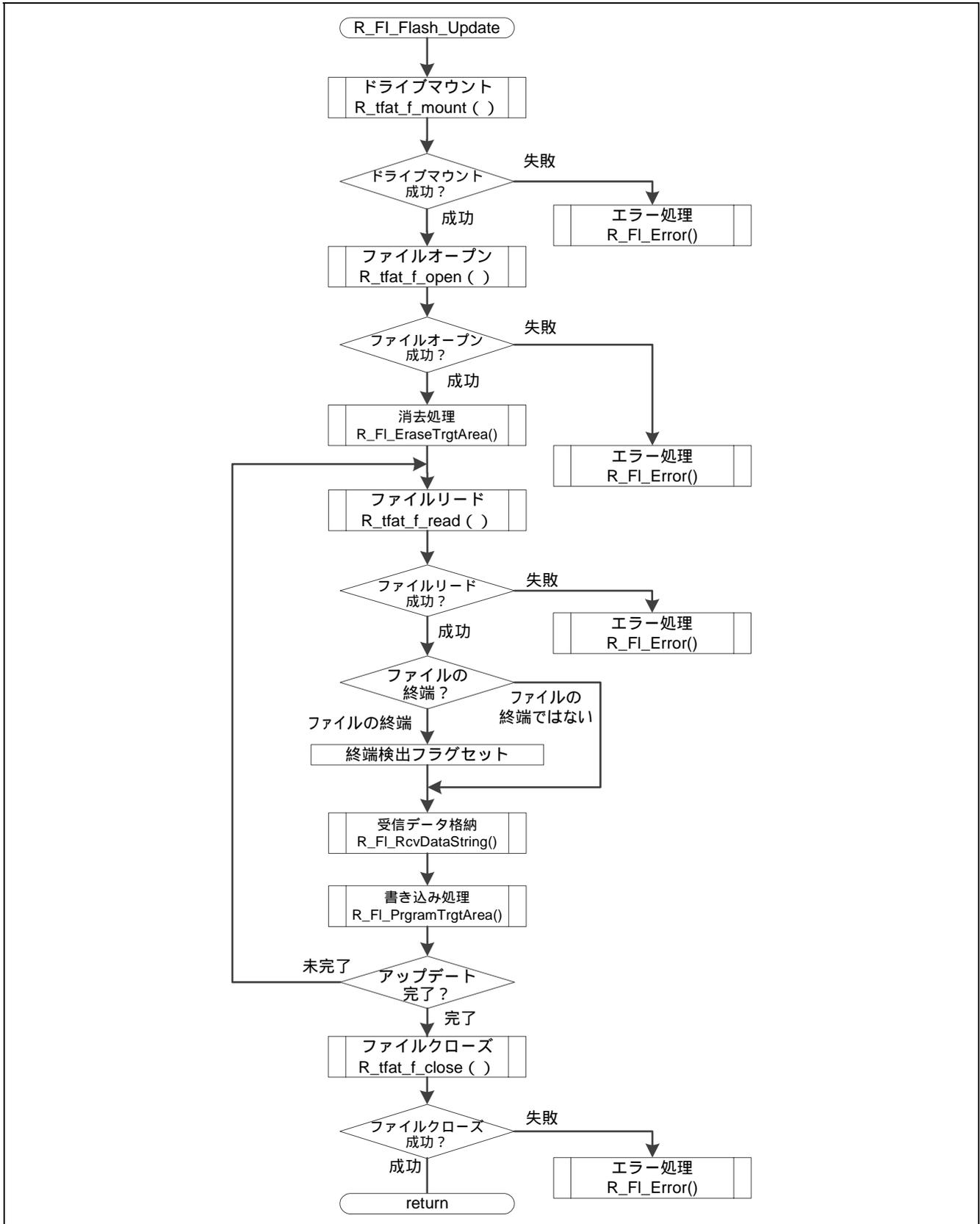


図5.11 書き換え処理メイン

5.12.4 消去処理

図 5.12に消去処理のフローチャートを示します。

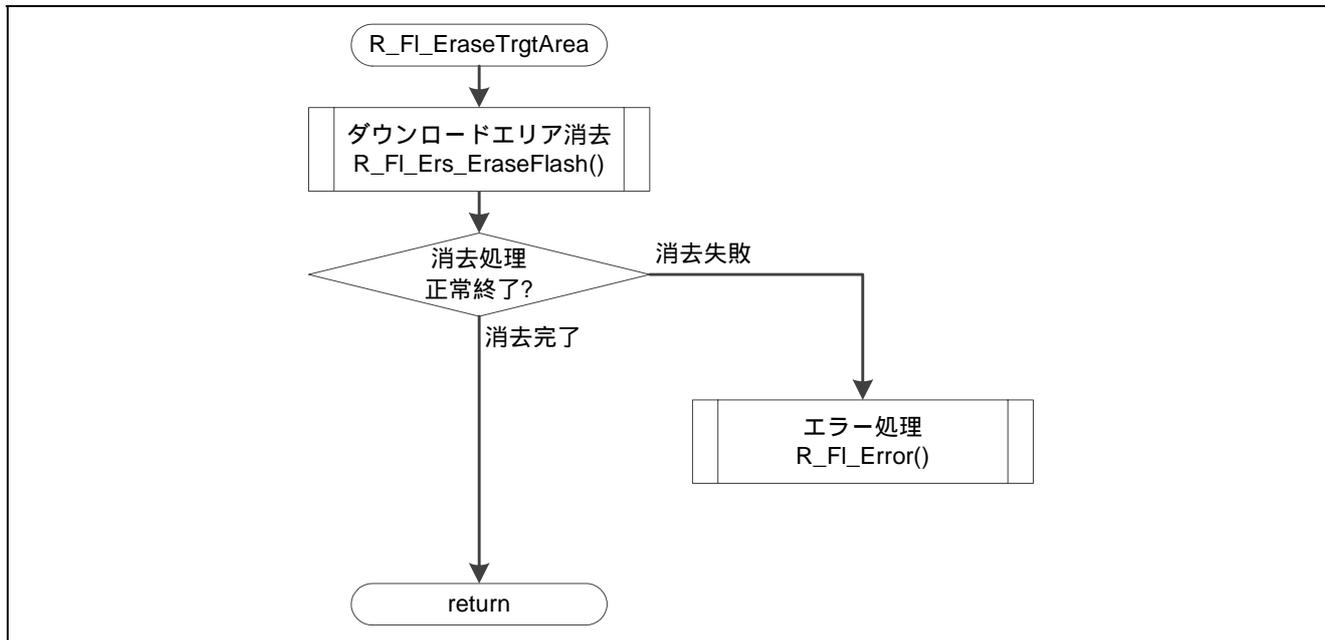


図5.12 消去処理

5.12.5 ダウンロードエリア消去

図 5.13にダウンロードエリア消去のフローチャートを示します。

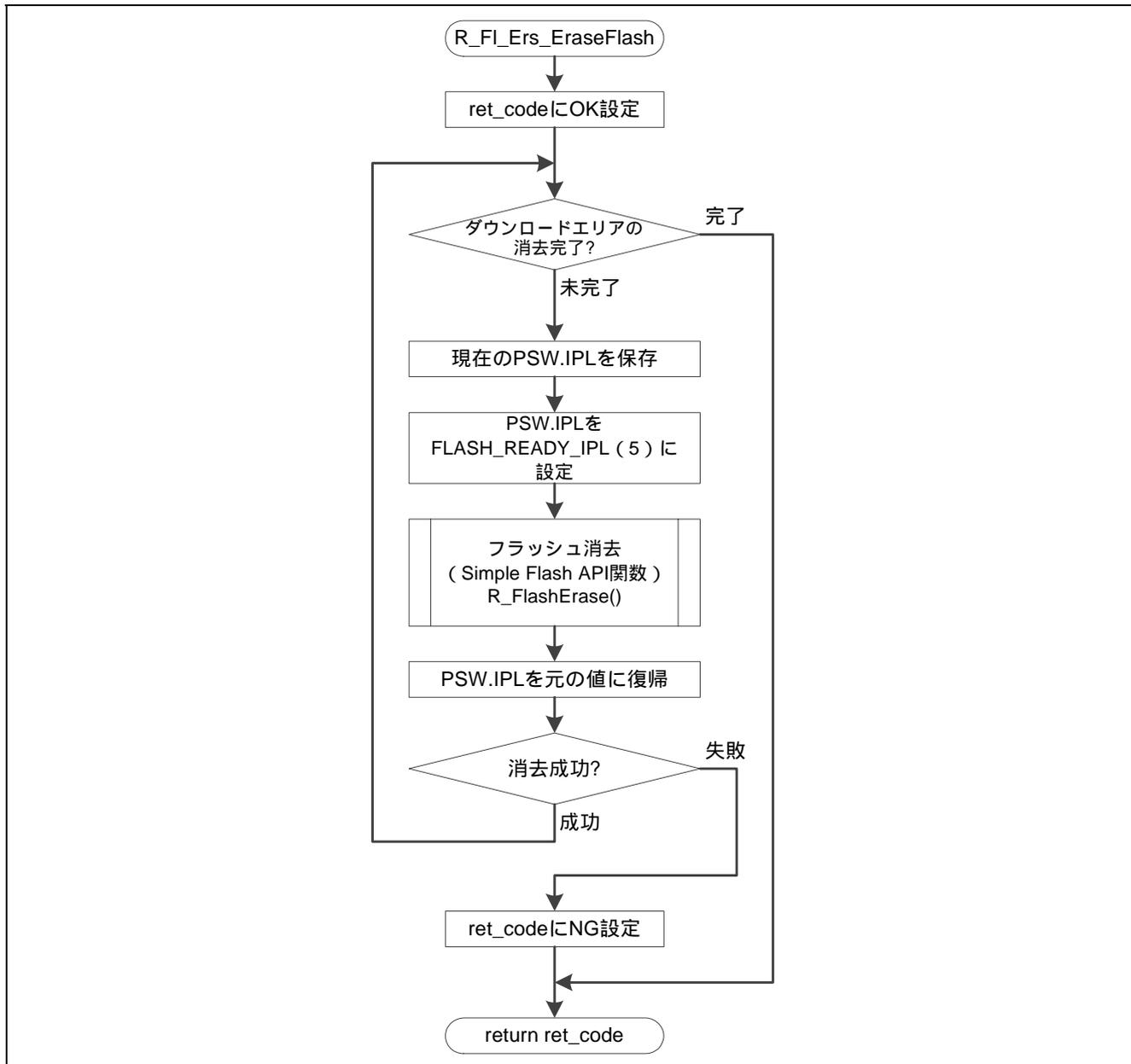


図5.13 ダウンロードエリア消去

5.12.6 書き込み処理

図 5.14に書き込み処理のフローチャートを示します。

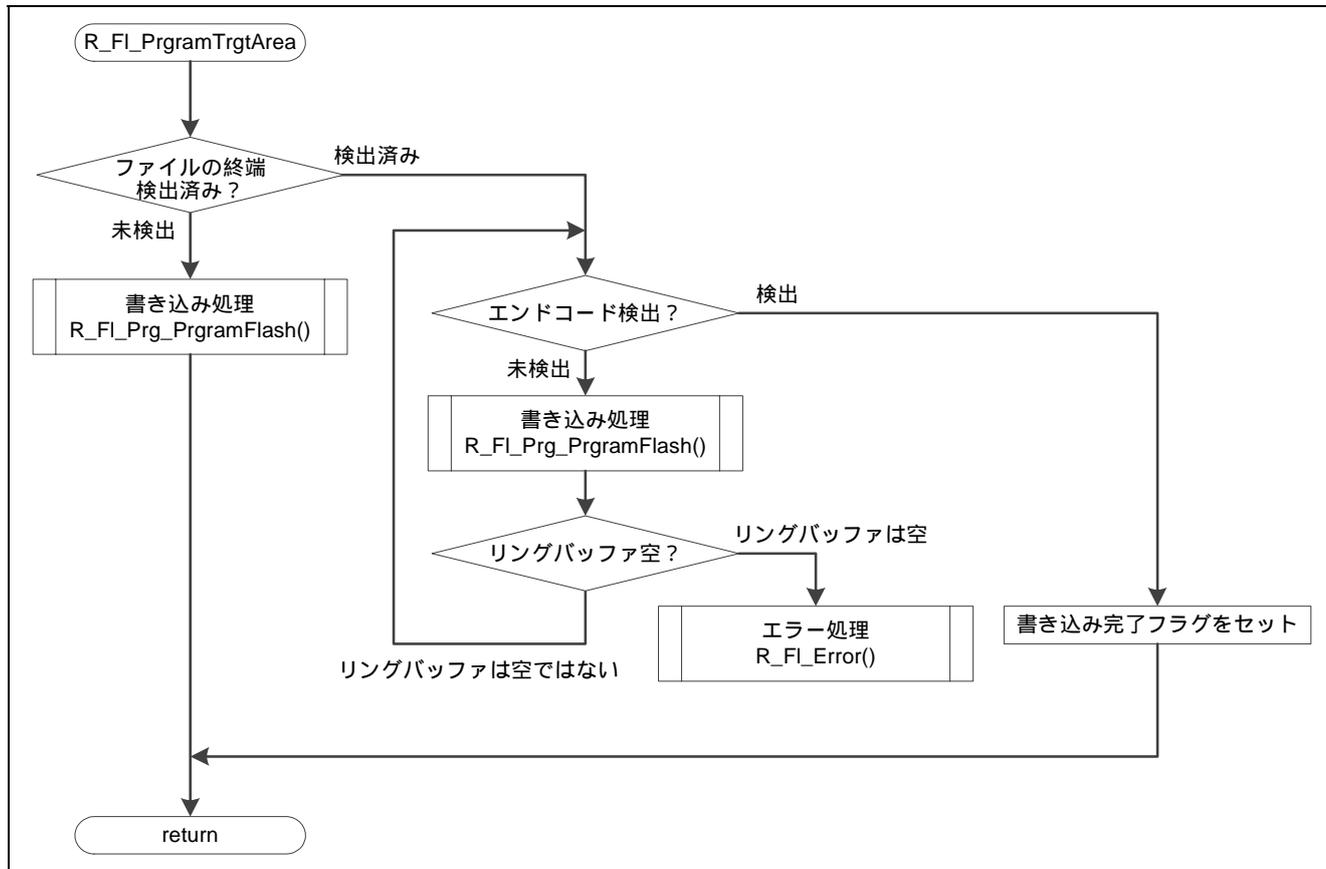


図5.14 書き込み処理

5.12.7 ダウンロードエリア書き込み

図 5.15にダウンロードエリア書き込みのフローチャートを示します。

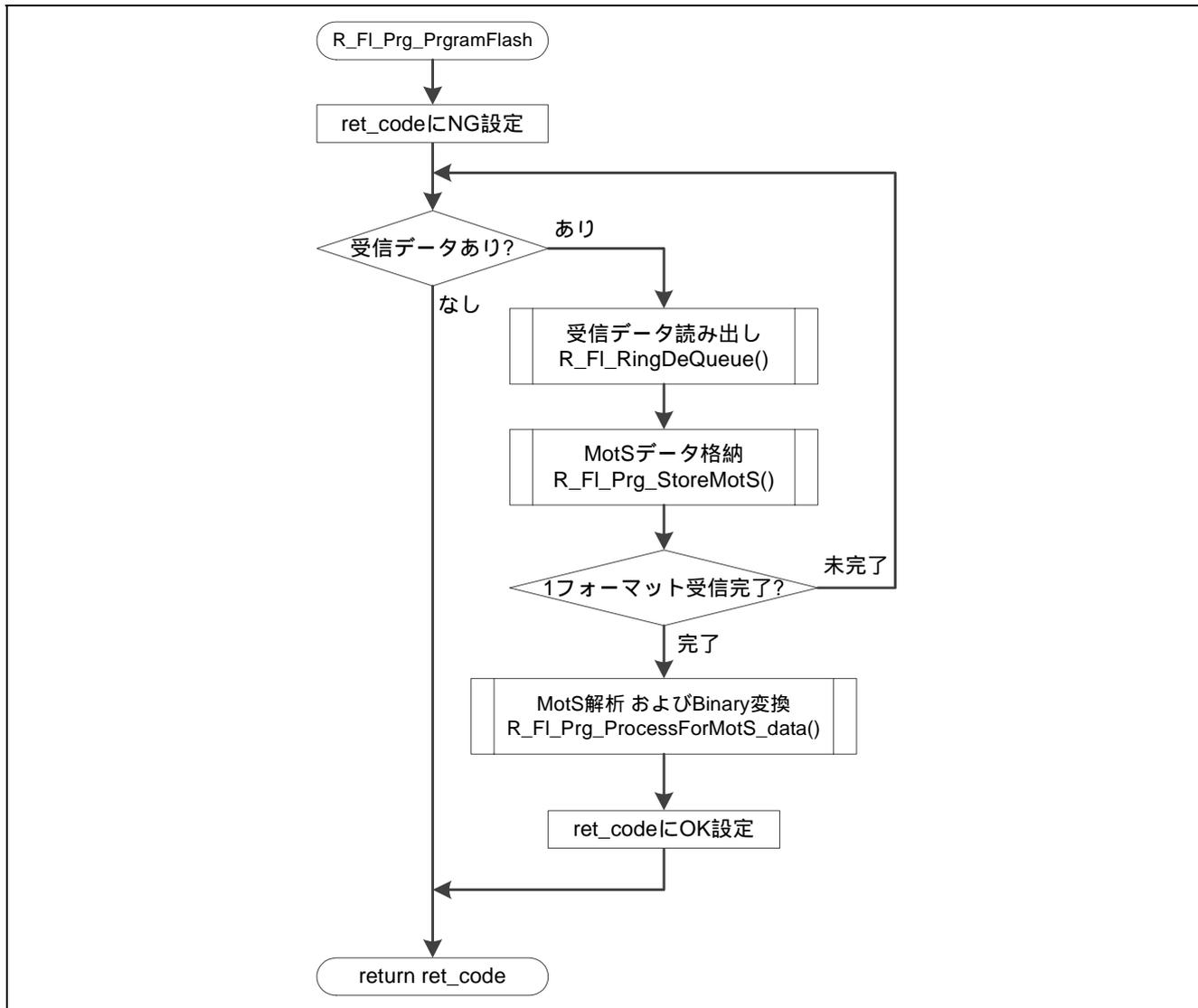


図5.15 ダウンロードエリア書き込み

5.12.8 Sタイプフォーマットデータ格納

図 5.16にSタイプフォーマットデータ格納のフローチャートを示します。

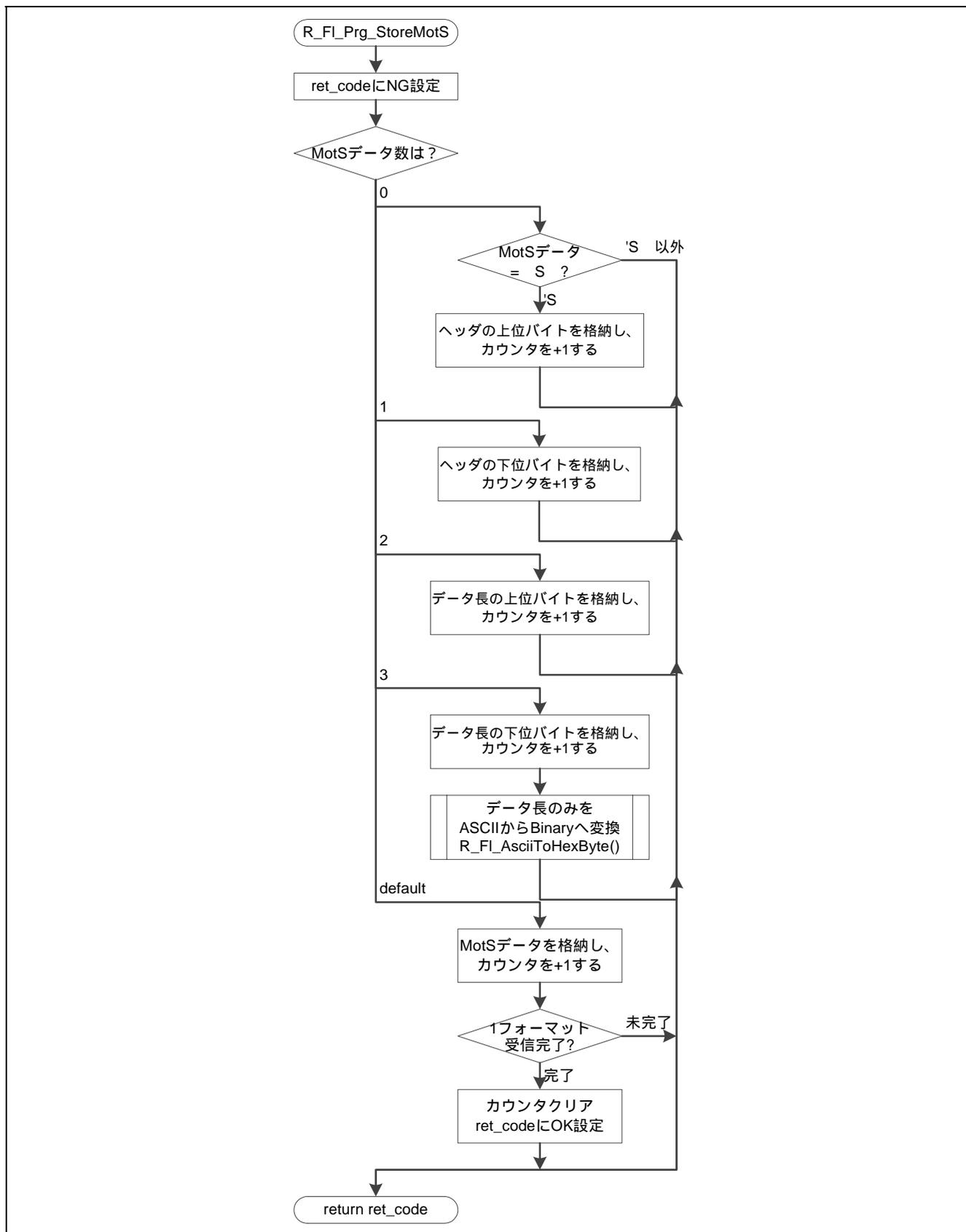


図5.16 Sタイプフォーマットデータ格納

5.12.9 S タイプフォーマットヘッダ解析、および Binary 変換、書き込み

図 5.17にS タイプフォーマットヘッダ解析、および Binary 変換、書き込みのフローチャートを示します。

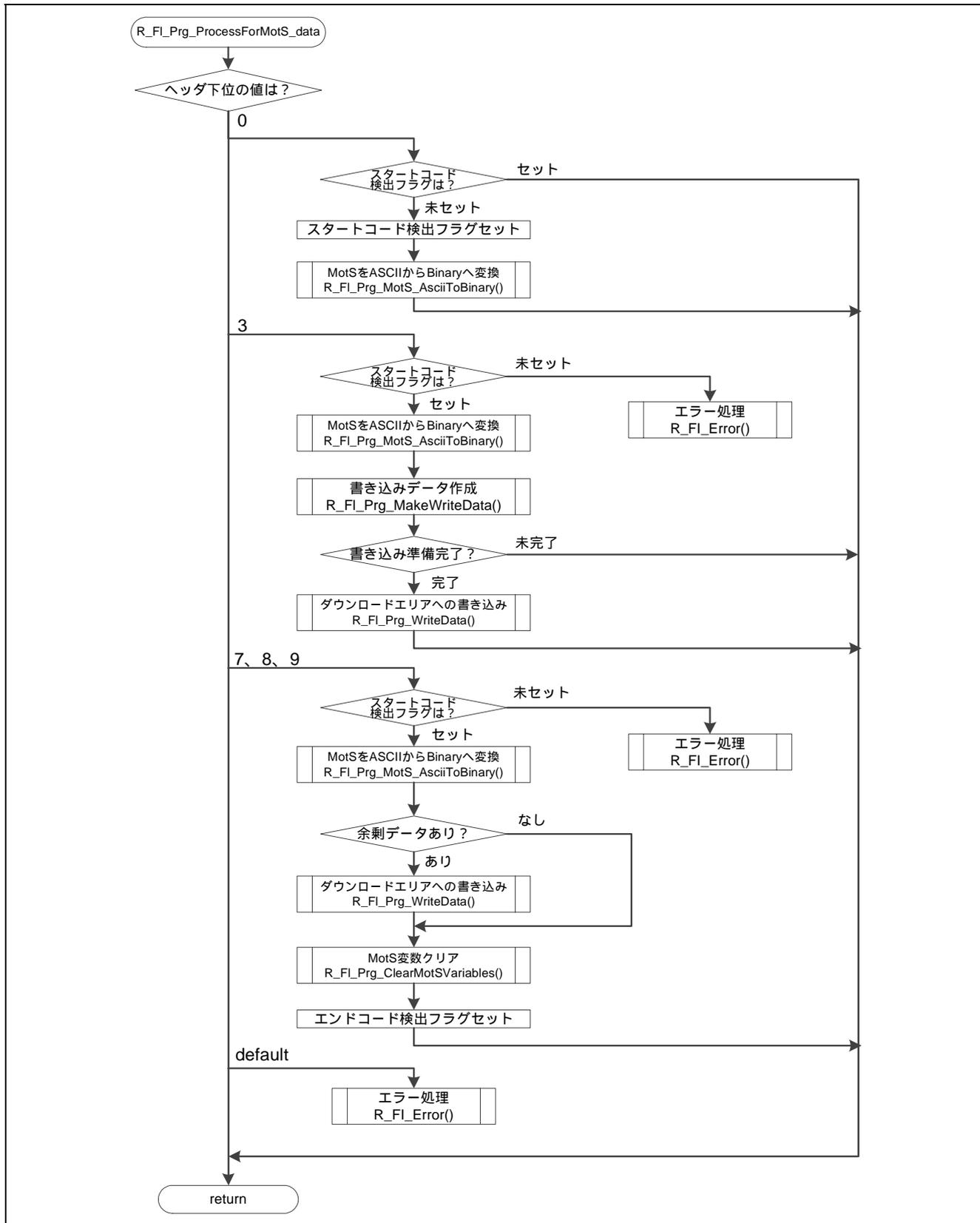


図5.17 S タイプフォーマットヘッダ解析、および Binary 変換、書き込み

5.12.10 S タイプフォーマットデータ ASCII - Binary 変換

図 5.18にS タイプ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。

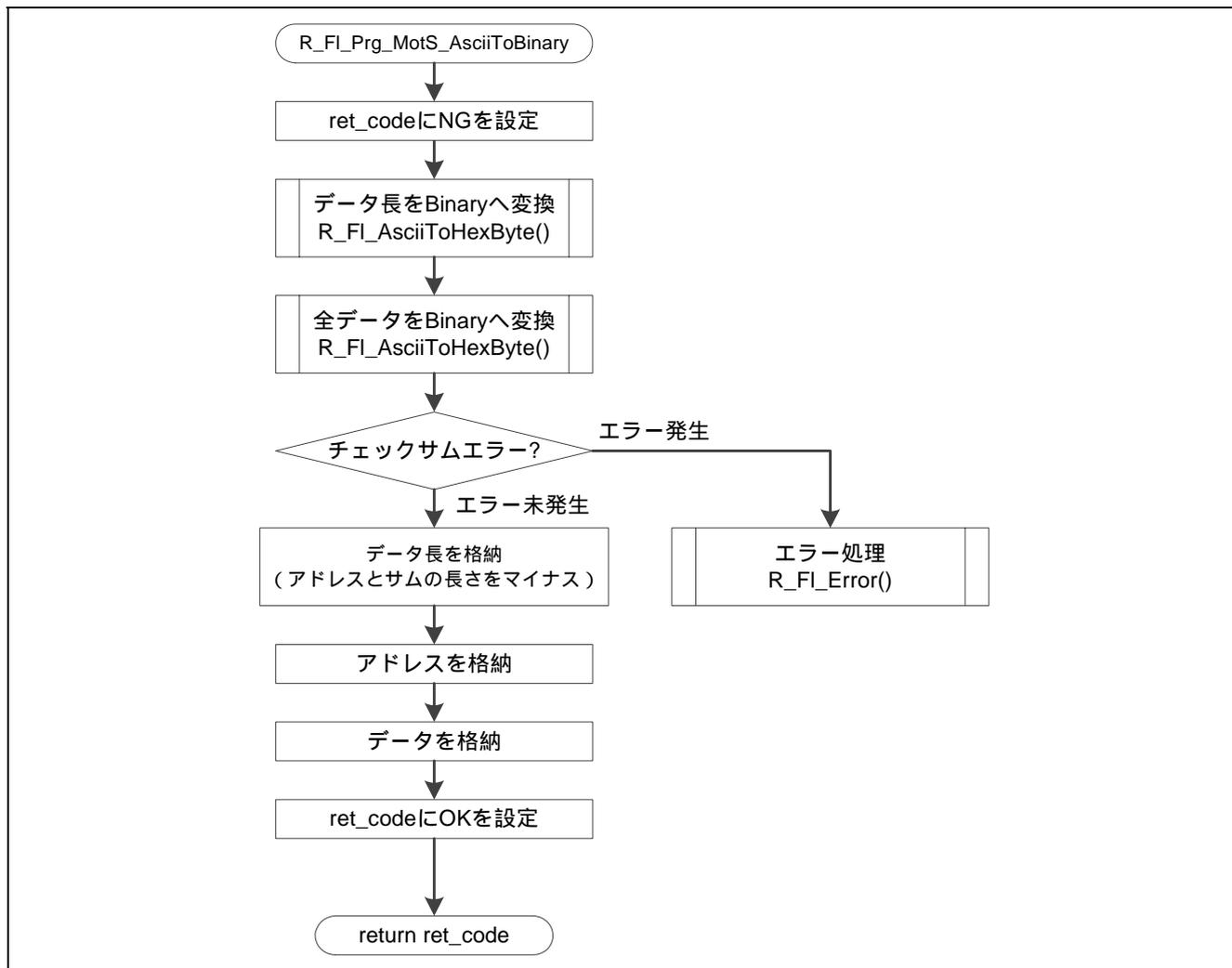


図5.18 S タイプフォーマットデータ ASCII - Binary 変換

5.12.11 ダウンロードエリアへの書き込みデータ作成

図 5.19にダウンロード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。

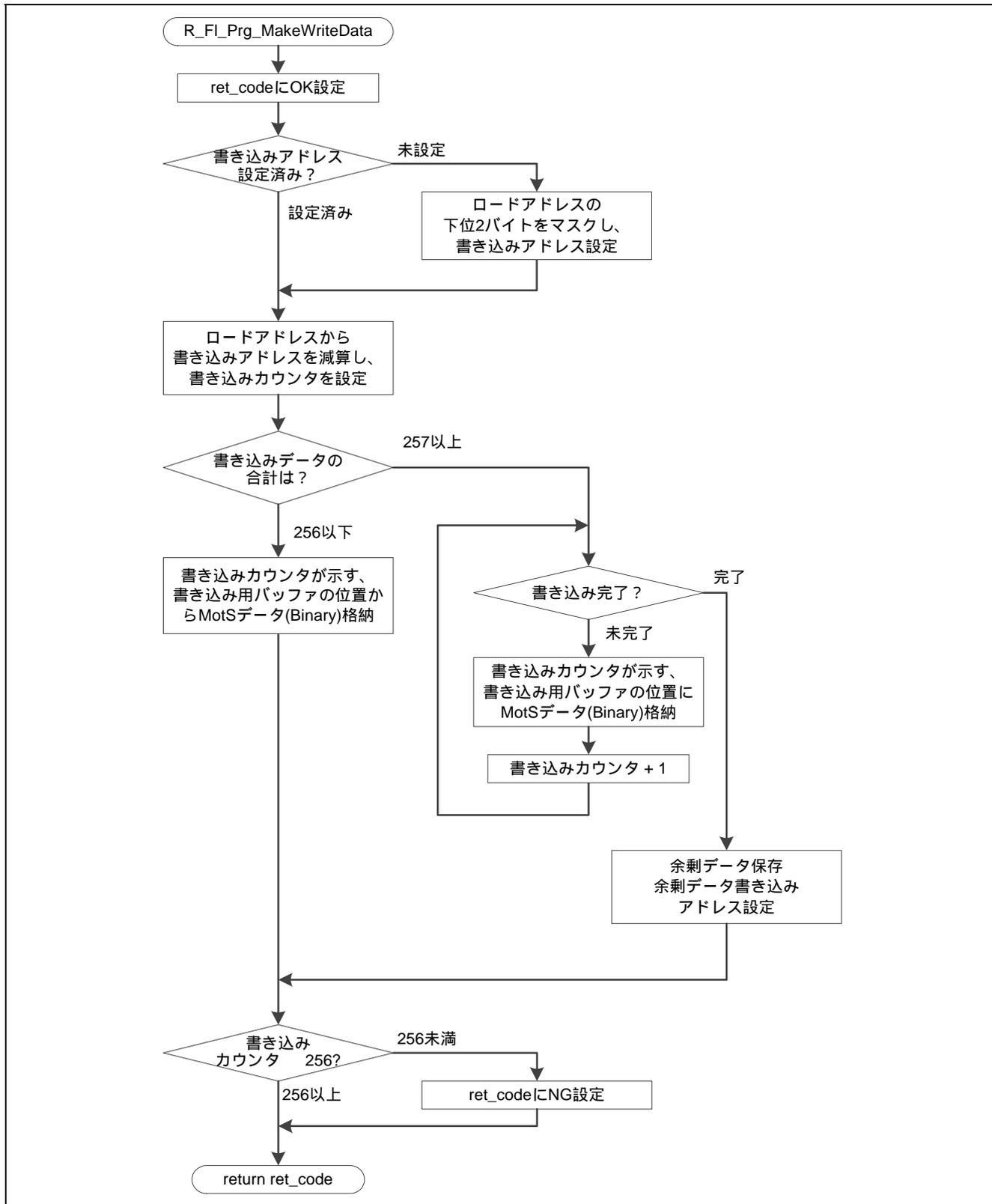


図5.19 ダウンロードエリアへの書き込みデータ作成

5.12.12 ダウンロードエリアへの書き込み

図 5.20にダウンロードエリアへの書き込みのフローチャートを示します。

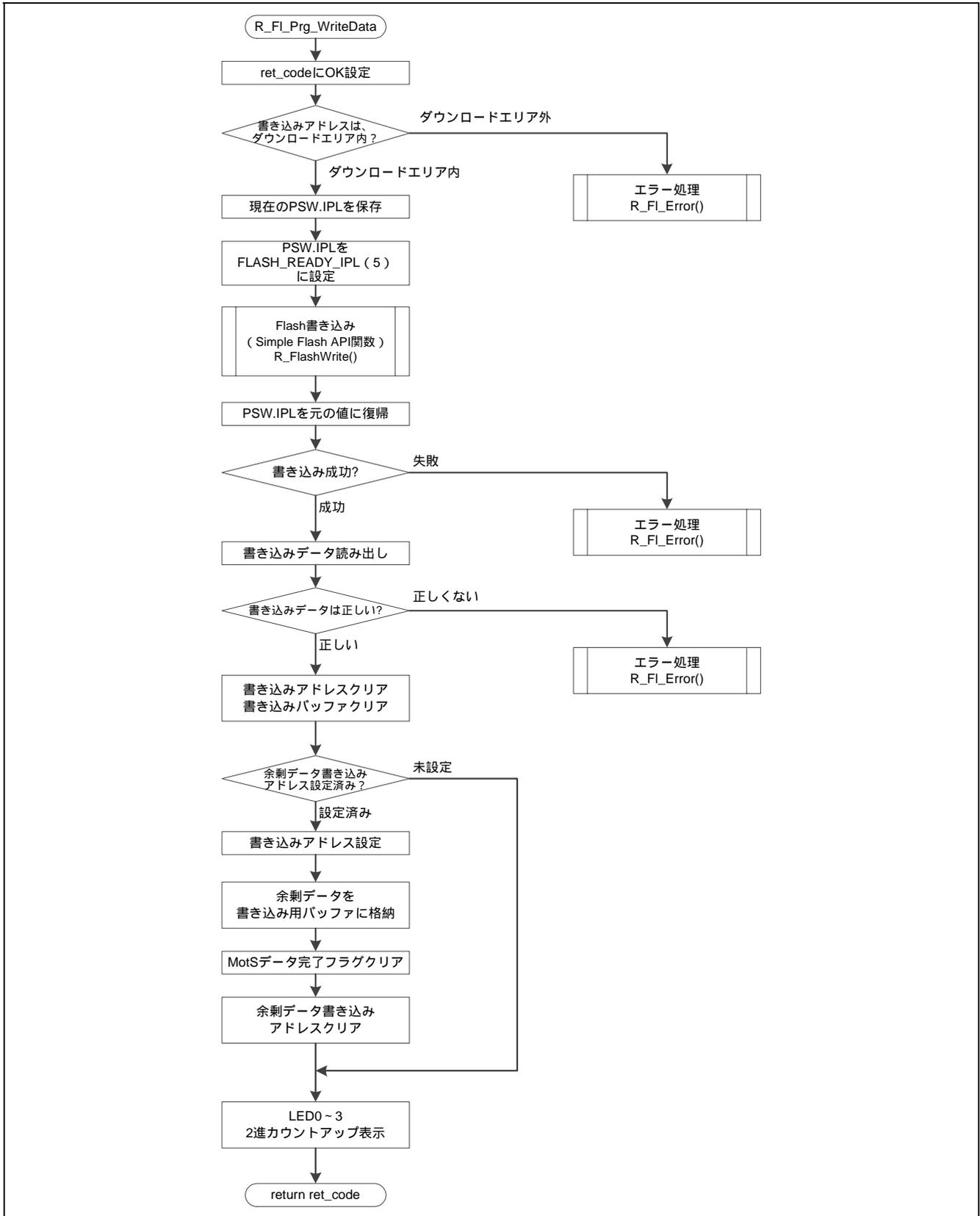


図5.20 ダウンロードエリアへの書き込み

5.12.13 Sタイプフォーマットデータ関連の変数クリア

図 5.21にSタイプフォーマットデータ関連の変数クリアのフローチャートを示します。

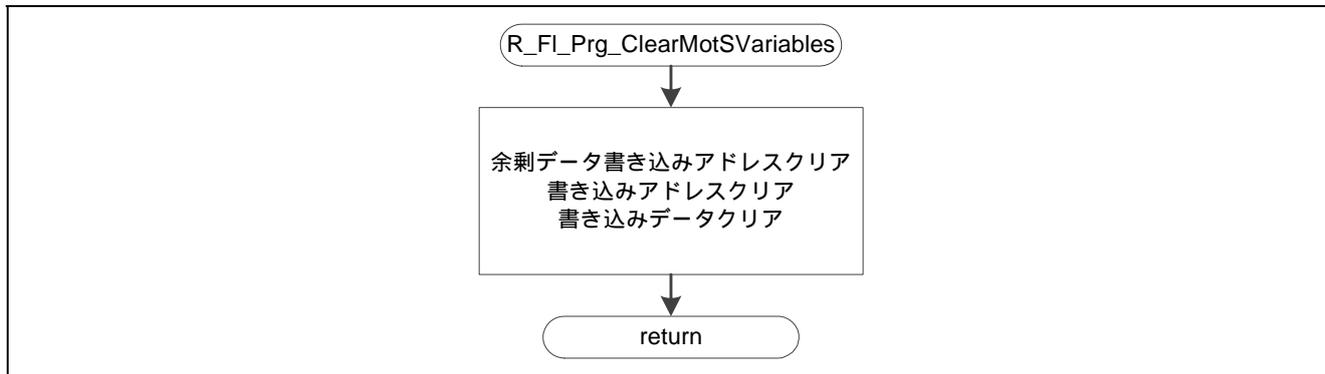


図5.21 Sタイプフォーマットデータ関連の変数クリア

5.12.14 USB 受信データ格納

図 5.22にUSB 受信データ格納のフローチャートを示します。

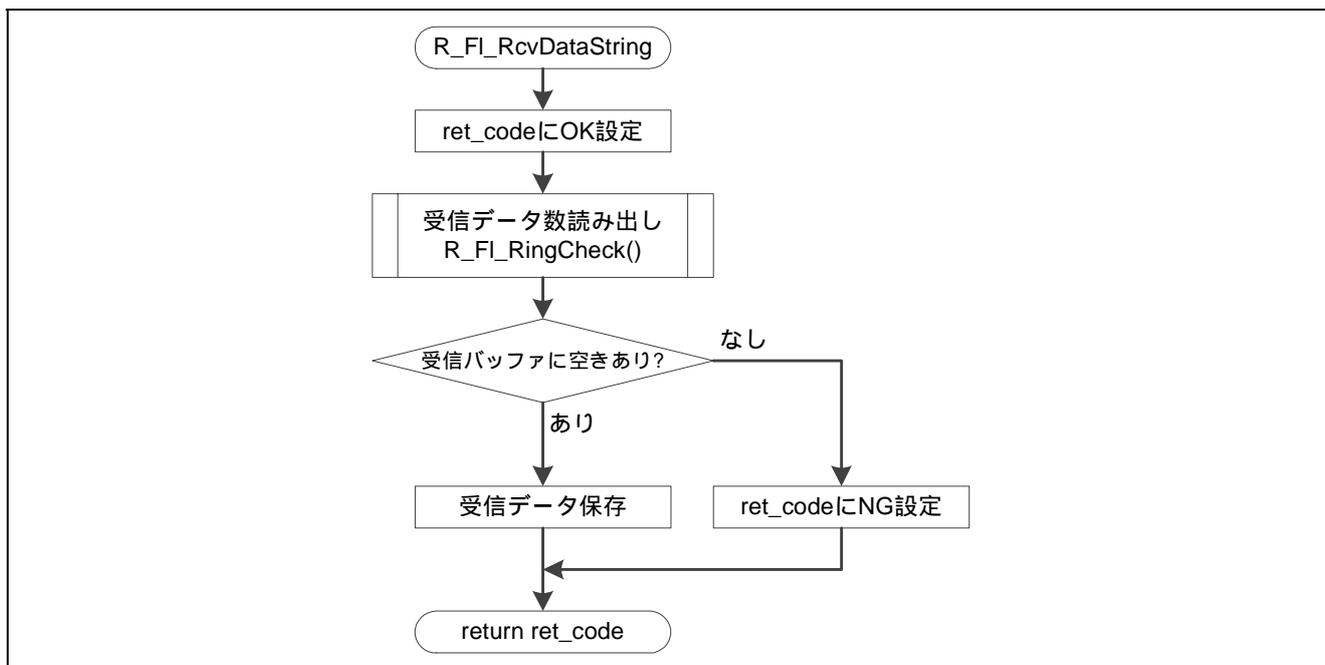


図5.22 USB 受信データ格納

5.12.15 受信用リングバッファの空き容量確認

図 5.23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。

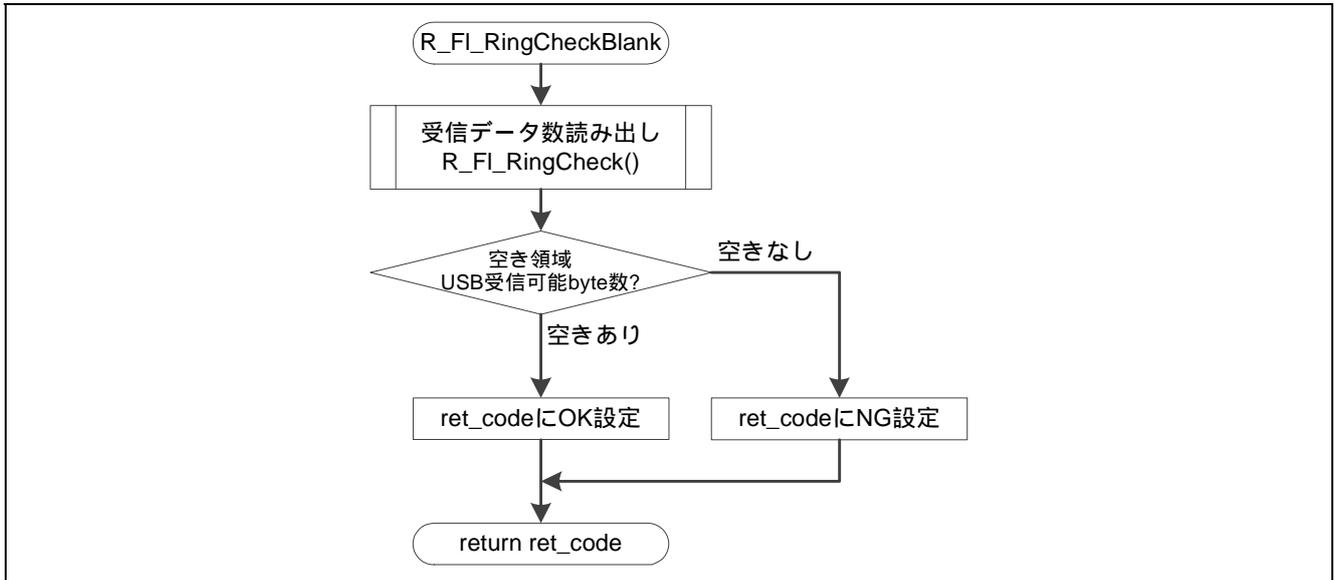


図5.23 受信用リングバッファの空き容量確認

5.12.16 USB 停止

図 5.24にUSB 停止のフローチャートを示します。

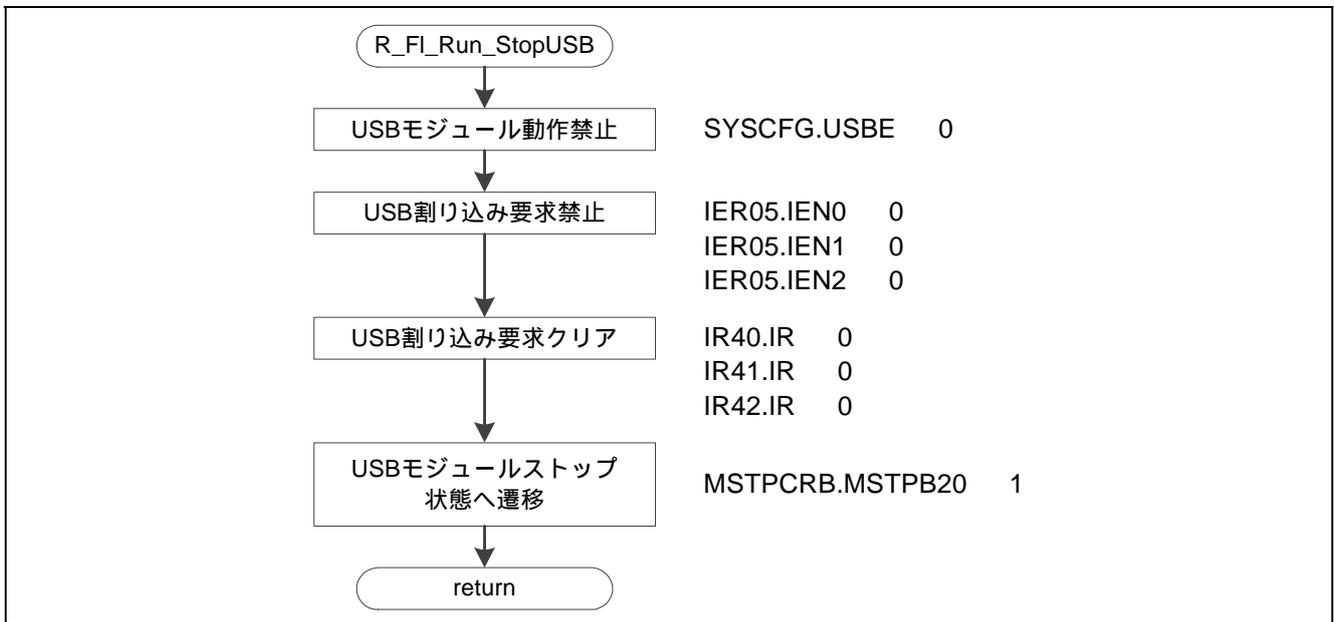


図5.24 USB 停止

5.12.17 エラー処理

図 5.25にエラー処理のフローチャートを示します。

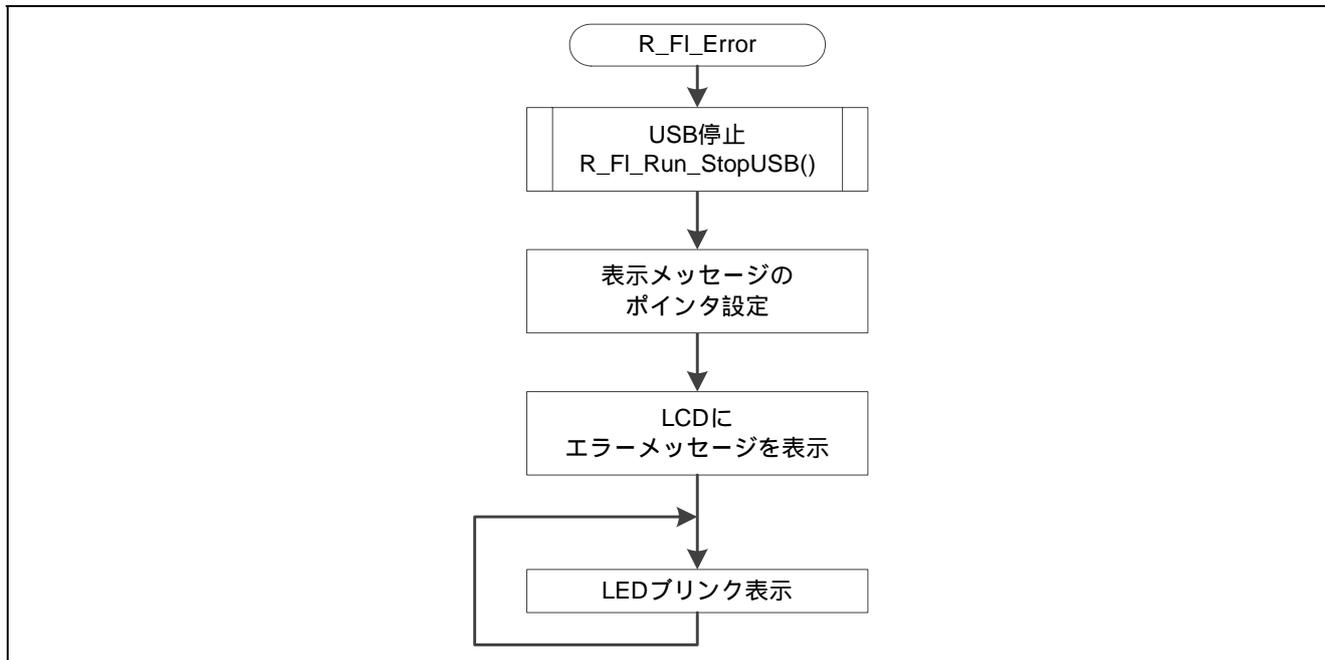


図5.25 エラー処理

5.12.18 受信用リングバッファへのデータ格納

図 5.26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。

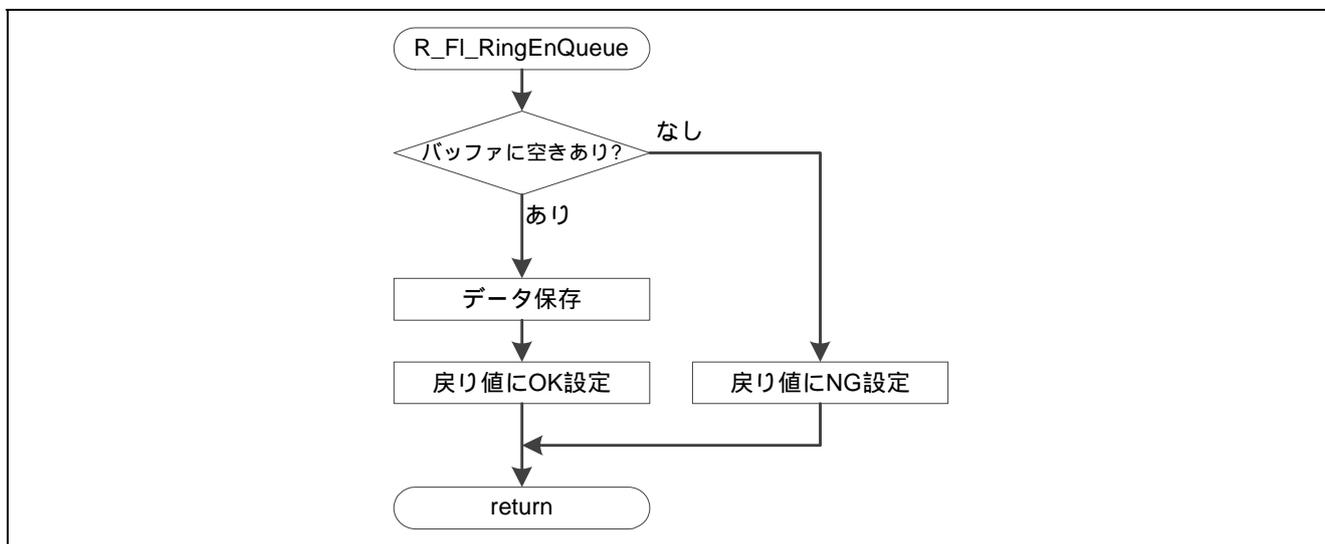


図5.26 受信用リングバッファへのデータ格納

5.12.19 受信用リングバッファからのデータ読み出し

図 5.27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。

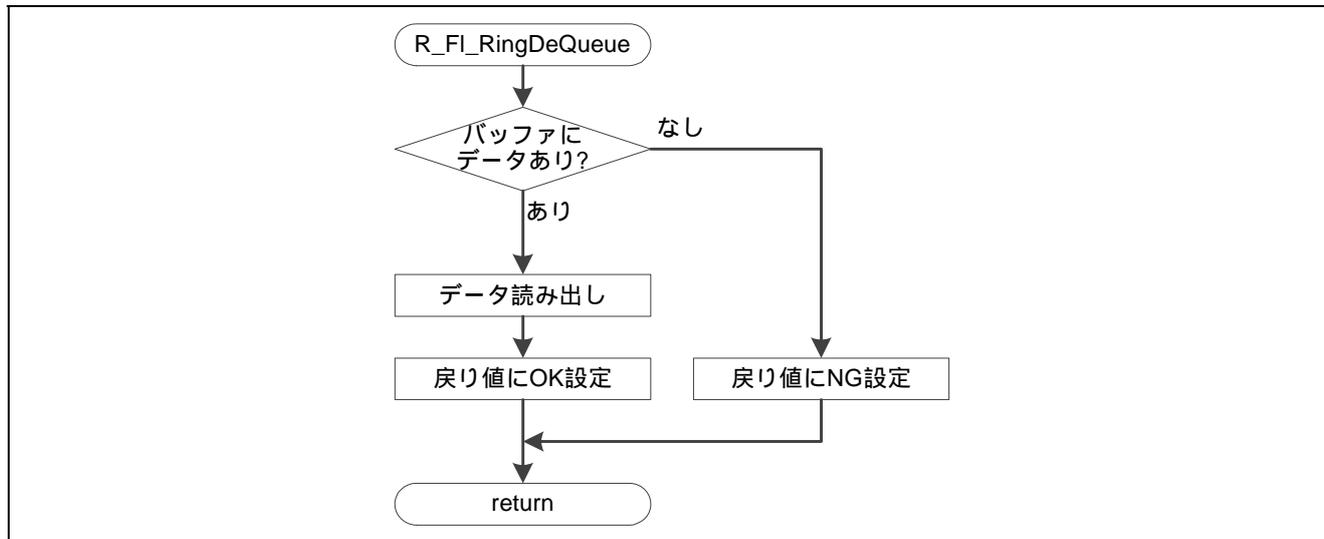


図5.27 受信用リングバッファからのデータ読み出し

5.12.20 受信用リングバッファのデータ数確認

図 5.28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。

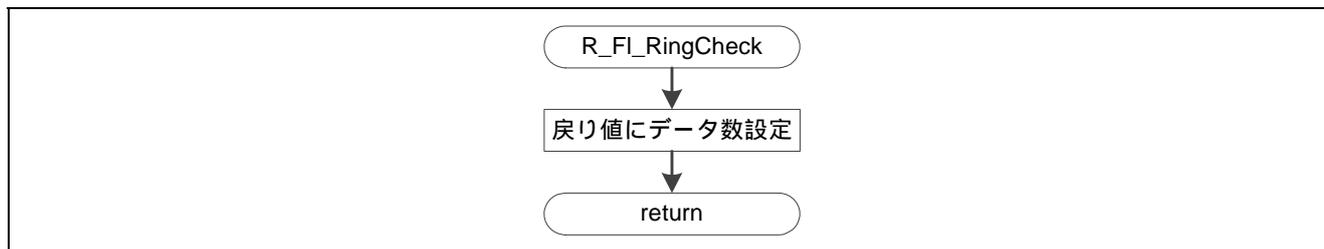


図5.28 受信用リングバッファのデータ数確認

5.12.21 ASCII コードから Binary データへの変換

図 5.29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。

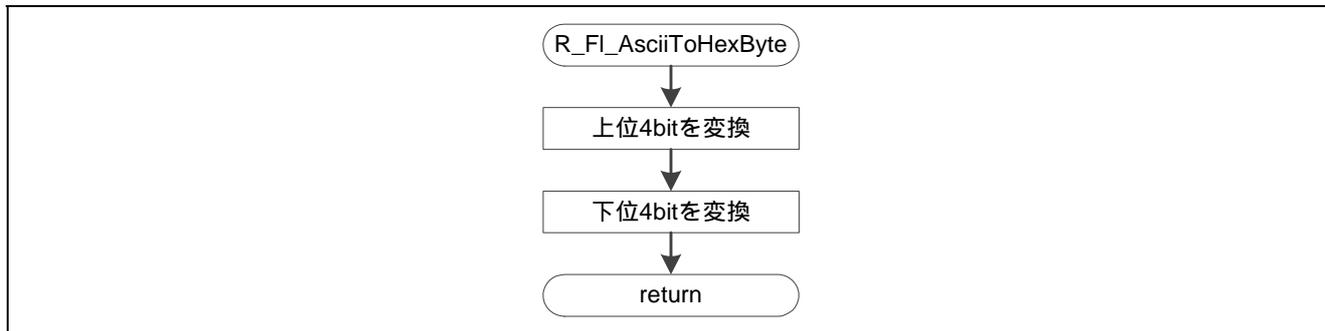


図5.29 ASCII コードから Binary データへの変換

6. ダウンロードコードの例

本アプリケーションノートのファイルには、ダウンロードコードの例(download.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。ダウンロード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、ダウンロードコードは、512Kbyte の ROM 容量の使用を前提としています。

7.4 エラー検出

サンプルコードでは、受信した S タイプフォーマットに異常があった場合、エラーを検出します。

(a) チェックサムエラー

受信した S タイプフォーマットのレコード毎にチェックサムの検査を行い、検査結果に異常があった場合、「チェックサムエラー」を検出します。

(b) フォーマットエラー

S タイプフォーマットが以下の条件に当てはまった場合、「フォーマットエラー」を検出します。

- 未対応レコード (S1、S2、S4、S5、S6) を検出した場合。
- ヘッダレコード (S0) を 2 回検出した場合
- ヘッダレコード (S0) 検出前に、データレコード (S3) もしくは、エンドレコード (S7、S8、S9) を検出した場合。

図 7.3 にフォーマットエラーの検出条件を示します。

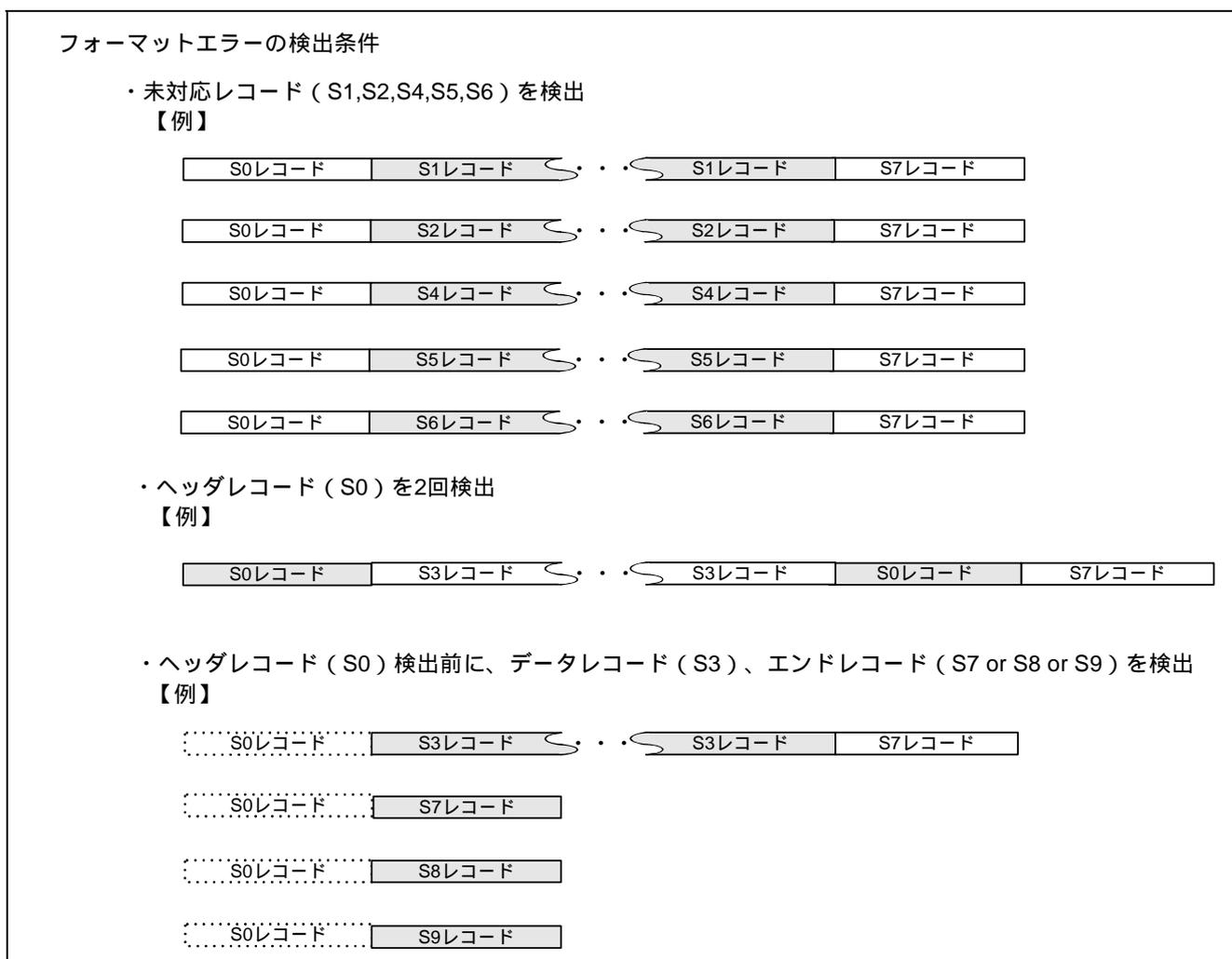


図 7.3 フォーマットエラーの検出条件

(c) アドレスエラー

ダウンロードエリア外への書き込みデータを受信した場合、「アドレスエラー」を検出します。

8. 注意事項

8.1 書き込み、消去中の USB 切断

ダウンロード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。

8.2 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 シリーズ RX600 用の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。

8.3 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。

8.4 ダウンロードコードのリセットベクタ

サンプルコードを使用して書き込むダウンロードコードの実行開始位置は、ダウンロードリセットベクタ (FFFE 3FFCh) の値で決定されます。従って、ダウンロードコードは、リセットベクタが FFFE 3FFCh に配置されるように設定してください。詳細は「5.2. ダウンロードコードの実行開始位置」を参照してください。

また、ダウンロードコードの例については「6. ダウンロードコードの例」を参照してください。

8.5 ROM 容量の変更

サンプルコードが使用しているマイコンの ROM 容量は 512K byte です。ROM 容量が 384K byte または、256K byte 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 8.1 に ROM 容量一覧を示します。

表8.1 ROM 容量一覧

製品型名	ROM 容量	ダウンロードエリア ROM 容量	ダウンロードエリア 開始アドレス	ダウンロードエリア ブロック番号
R5F562x8	512K	400K	FFF8 0000h	EB13 ~ EB37
R5F562x7	384K	272K	FFFA 0000h	EB13 ~ EB29
R5F562x6	256K	144K	FFFC 0000h	EB13 ~ EB21

8.6 while(1)の処理

サンプルコードでは、USB 受信リング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。ご注意ください。

8.7 エンディアン

本アプリケーションノートのサンプルコードは、リトルエンディアン/ビッグエンディアンの両方に対応しています。なお、エンディアンはフラッシュブートローダとダウンロードコードで同じ設定にしてご設計ください。

8.7.1 リトルエンディアン使用時

リトルエンディアンで動作する場合は以下の設定を行ってください。

1. モード端子 (MDE) の設定
RX62N RSK の SW4 Pin3 を ON (MDE=L) に設定してください。
2. コンパイラオプションの設定
コンパイラオプションのエンディアンの設定で"Little-endian データ"を指定してください。
3. ユーザシステム定義ファイル (r_usbc_cDefUsr.h) の変更
"r_usbc_cDefUsr.h"内のマクロ定義"USBC_CPUBYTE_PP"の値を"USBC_BYTE_LITTLE_PP"としてください。
4. TFAT ライブラリファイル (tfat_rx_little_v100.lib、tfat_rx_big_v100.lib) の変更
リンクの入力ライブラリファイルオプションで"WorkSpace¥MSC2FW¥TFAT¥lib¥tfat_rx_little_v100.lib"を指定してください。

8.7.2 ビッグエンディアン使用時

ビッグエンディアンで動作する場合は以下の設定を行ってください。

1. モード端子 (MDE) の設定
RX62N RSK の SW4 Pin3 を OFF (MDE=H) に設定してください。
2. コンパイラオプションの設定
コンパイラオプションのエンディアンの設定で"Big-endian データ"を指定してください。
3. ユーザシステム定義ファイル (r_usbc_cDefUsr.h) の変更
"r_usbc_cDefUsr.h"内のマクロ定義"USBC_CPUBYTE_PP"の値を"USBC_BYTE_BIG_PP"としてください。
4. TFAT ライブラリファイル (tfat_rx_little_v100.lib、tfat_rx_big_v100.lib) の変更
リンクの入力ライブラリファイルオプションで"WorkSpace¥MSC2FW¥TFAT¥lib¥tfat_rx_big_v100.lib"を指定してください。

8.8 RX600 用のシンプルフラッシュ API からの変更点

サンプルコードは、RX600 用のシンプルフラッシュ API のプログラムを流用しています。RX600 用のシンプルフラッシュ API の仕様については RX600 用の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。

8.8.1 変更箇所

RX600 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"、"mcu_info.h" です。

- ファイル "r_flash_api_rx600_config.h" 内の変更箇所：
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

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 IGNORE_LOCK_BITS
        #define COPY_CODE_BY_API
        #define FLASH_API_USE_R_BSP
変更後：##define IGNORE_LOCK_BITS
        ##define COPY_CODE_BY_API
        ##define FLASH_API_USE_R_BSP
```

- ファイル "mcu_info.h" 内の変更箇所：
シンプルフラッシュ API の r_bsp/board/rskrx62n フォルダに格納されているファイルを使用しています。

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 ICLK_HZ      (96000000)
        #define PCLK_HZ      (48000000)
        #define BCLK_HZ      (12000000)
変更後：#define ICLK_HZ      (48000000)
        #define PCLK_HZ      (24000000)
        #define BCLK_HZ      (48000000)
```

8.9 USB Host Mass Storage Class Driver からの変更点

サンプルコードは USB Host Mass Storage Class Driver のプログラムを流用しています。USB Host Mass Storage Class Driver の仕様については USB Host Mass Storage 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。

8.9.1 変更箇所

USB Host Mass Storage Class Driver から変更したファイルは

- r_usb2_HMSC_apl.c
- dbsct_hmsc.c
- RX62NRSK.c
- resetprg.c

の4つです。

- ファイル"r_usb2_HMSC_apl.c"内の変更箇所：
#ifdef R_FLASH_USB で示しています。
- ファイル"dbsct_hmsc.c"内の変更箇所：
コメント"// Flash table"で示しています。
- ファイル"RX62NRSK.c"内の変更箇所：
「usb2_cstd_TargetInit」関数内の IRQ8、9 割り込み設定関数を削除しています。
削除：usbc_cpu_IRQ9_Enable();
削除：usbc_cpu_IRQ8_Enable();
- ファイル "resetprg.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"
 2. 関数"PowerON_Reset_PC"内の INTB の設定箇所を変更しています。
変更前：set_intb(0xFFFF80000);
変更後：set_intb(__sectop("INTERRUPT_VECTOR"));
 3. 関数"PowerON_Reset_PC"内でモードエントリ関数を呼んでいます。
追加：R_Fl_Mode_Entry();

8.9.2 追加ファイル

USB Host Mass Storage 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。

8.9.3 追加セクション

表 8.2 に USB Host Mass Storage Class Driver の追加セクションを示します。

表8.2 追加セクション

セクション名	概要
R_flash_api_sec	RAM 上で動作する Flash 書き換えコードの変数用セクション
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	ダウンロードコード固定ベクタ用セクション

8.9.4 インクルードファイルディレクトリ

"WorkSpace¥FLASH" 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。

8.9.5 マクロ定義 (コンパイラオプション)

- "R_FLASH_USB" をコンパイラオプションのマクロ定義に追加しています。
- "USBC_SDRAM_USE_PP" をコンパイラオプションのマクロ定義から削除しています。

8.9.6 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。

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

- RX62N グループ、RX621 グループ ユーザーズマニュアル ハードウェア編 Rev.1.20
(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。)
- C コンパイラマニュアル
- RX ファミリー C コンパイラパッケージ V.1.10R00
- C コンパイラユーザーズマニュアル Rev.1.10R00
(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。)
- アプリケーションノート RX600 シリーズ RX600 用のシンプルフラッシュ API Rev.2.20 (R01AN0544JU)
(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。)
- アプリケーションノート ルネサス USB デバイス USB Basic Firmware Rev.1.10 (R01AN0512JJ)
(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。)
- アプリケーションノート ルネサス USB デバイス USB Host Mass Storage Class Driver Rev.1.10(R01AN0513JJ)
(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。)
- アプリケーションノート M3S-TFAT-Tiny : FAT ファイルシステムソフトウェア Rev.1.00(R20AN0038JJ)
(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/inquiry>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.03.28	—	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（予約領域）のアクセス禁止

【注意】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、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）がありません。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、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っていると、内部 ROM、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。型名が異なる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. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただけますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている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事務の目的で使用しないでください。また、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に使用することができません。
6. 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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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、当社は、一切その責任を負いません。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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と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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■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/>