# RX62N、RX621 グループ

## M3S-T4-Tiny による Ethernet フラッシュブートローダ

R01AN0891JJ0100
Rev.1.00
2012.03.13

### 要旨

本アプリケーションノートでは、M3S-T4-Tiny（TCP/IP プロトコルスタック）を使用した Ethernet 経由のデータ転送により、ホスト PC から転送された S タイプフォーマットファイルでマイコン内蔵フラッシュメモリの書き換えを行う「M3S-T4-Tiny による Ethernet フラッシュブートローダ」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ライブラリを使用しています。

- Ethernet 経由でのデータ転送：
  「RX ファミリ M3S-T4-Tiny：導入ガイド」Rev.1.02 (R20AN0051JJ0102)
- 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)

本アプリケーションコードの特長を以下に示します。

- PC 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むことが可能
  ホスト PC 上で動作するアプリケーション（ホスト PC 用サンプルプログラム）を用いて、S タイプフォーマットファイルを Ethernet 経由にて転送し、マイコン内蔵フラッシュメモリの消去および書き込みを行います。
- 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。

- Ethernet 仕様
  トランスポート層のプロトコルには TCP を使用しています。

### 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

本アプリケーションノートのサンプルコードは、Ethernet ケーブルで接続するホスト PC および RX62N-RSK で動作します。

RX62N-RSK 上のスイッチ（SW3）を押していない状態でリセットを解除すると、Ethernet ケーブルで接続されたホスト PC（ホスト PC 用サンプルプログラムを使用します）と RX62N-RSK が TCP/IP プロトコルによる通信を行い、ホスト PC 内の S タイプフォーマットのプログラムを RX62N-RSK にデータ転送し、マイコン内蔵フラッシュメモリに書き込みます。なお、サンプルコードが書き換える領域は、ユーザマットの一部のみとなり、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「5.3　動作概要」を参照してください。

RX62N-RSK 上のスイッチ（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。

内蔵フラッシュメモリへの書き込み結果は、RX62N-RSK 上の発光ダイオード（LED0～LED3）に表示します。表示の内容は「5.6.　サンプルコードの LED 表示」を参照してください。

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<br>内蔵フラッシュメモリの書き換え |
| ETHERC イーサネットコントローラ | ホスト PC との通信用 |
| EDMAC イーサネットコントローラ用 DMA コントローラ | ホスト PC との通信におけるデータ送受信制御用 |
| CMT コンペアマッチタイマ | TCP/IP プロトコルスタック（M3S-T4-Tiny）の時間管理用 |

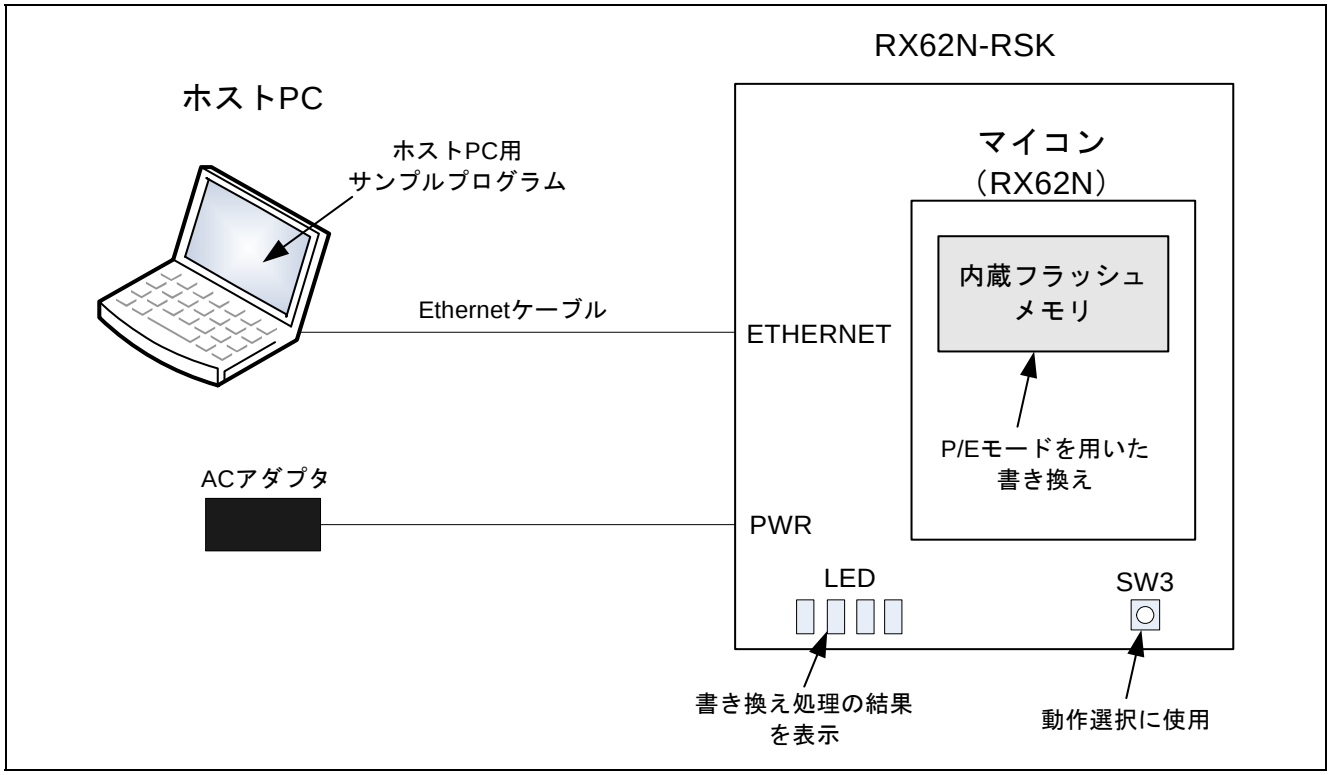


図1.1　使用例

## 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、表 2.1および表 2.2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 | • EXTAL：12MHz<br>• ICLK：96MHz<br>• PCLK：48MHz<br>• BCLK：24MHz<br>• SDCLK：24MHz |
| 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<br>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<br>RX Standard Toolchain Version 1.1.0.0 |
| | -cpu=rx600<br>-include="$(PROJDIR)¥src¥bsp","$(PROJDIR)¥src¥FlashAPI",<br>"$(PROJDIR)¥src¥driver","$(PROJDIR)¥src¥t4¥lib",<br>"$(PROJDIR)¥src¥user_app"<br>-output=obj="$(CONFIGDIR)¥$(FILELEAF).obj"<br>-debug<br>-nologo<br>*1 |
| プロセッサモード | スーパーバイザモード |
| 動作モード | シングルチップモード |
| エンディアン | リトルエンディアン／ビッグエンディアン |
| 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 |

【注】 *1 ビッグエンディアンの場合は、"-endian=big"を追加設定してください。

表2.2　動作確認条件（ホスト PC 用サンプルプログラム）

| 項目 | 内容 |
|---|---|
| ハードウェア | PC/AT 互換機（Ethernet インタフェース要） |
| オペレーティングシステム | Microsoft Windows XP Professional Service Pack 3 |
| 使用ツール | コマンドプロンプト（cmd.exe） |

## 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。あわせて参照してください。

- 「RX ファミリ M3S-T4-Tiny: 導入ガイド」Rev.1.02 (R20AN0051JJ)
- 「RX600 シリーズ RX600 用のシンプルフラッシュ API」Rev.2.20 (R01AN0544JU)

# 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 |
|---|---|---|
| ET_MDC | 出力 | ET_MDIO による情報転送用の参照クロック信号<br>PHY の MDC 端子に接続します |
| ET_MDIO | 入出力 | STA と PHY-LSI との間で管理情報を交換するための双方向信号<br>PHY の MDIO 端子に接続します |
| ET_LINKSTA | 入力 | PHY-LSI からのリンクステータス入力<br>PHY の LINK/PHYAD1 端子に接続します |
| ET_TX_CLK | 入力 | 送信クロック信号<br>PHY の TX_CLK 端子に接続します |
| ET_ETXD0 | 出力 | 4bit の送信データ<br>PHY の TXD0 端子に接続します |
| ET_ETXD1 | 出力 | 4bit の送信データ<br>PHY の TXD1 端子に接続します |
| ET_ETXD2 | 出力 | 4bit の送信データ<br>PHY の TXD2 端子に接続します |
| ET_ETXD3 | 出力 | 4bit の送信データ<br>PHY の TXD3 端子に接続します |
| ET_TX_EN | 出力 | 送信許可信号<br>PHY の TX_EN 端子に接続します |
| ET_TX_ER | 出力 | 送信中のエラーを PHY-LSI に通知<br>PHY の TX_ER 端子に接続します |
| ET_COL | 入力 | 衝突検出信号<br>PHY の COL 端子に接続します |
| ET_CRS | 入力 | キャリア検出信号<br>PHY の CRS 端子に接続します |
| ET_RX_CLK | 入力 | 受信クロック信号<br>PHY の RX_CLK 端子に接続します |
| ET_ERXD0 | 入力 | 4bit の受信データ<br>PHY の RXD0 端子に接続します |
| ET_ERXD1 | 入力 | 4bit の受信データ<br>PHY の RXD1 端子に接続します |
| ET_ERXD2 | 入力 | 4bit の受信データ<br>PHY の RXD2 端子に接続します |
| ET_ERXD3 | 入力 | 4bit の受信データ<br>PHY の RXD3 端子に接続します |
| ET_RX_DV | 入力 | 有効な受信データが ET_ERXD3～ET_ERXD0 上にあることを示す信号<br>PHY の RX_DV 端子に接続します |
| ET_RX_ER | 入力 | 受信エラー端子データ受信中に発生したエラー状態を示す信号<br>PHY の RX_ER 端子に接続します |

| 端子名 | 入出力 | 内容 |
|---|---|---|
| MDE | 入力 | モード端子<br>MDE 端子の制御によりエンディアンを変更します |
| MD0 | 入力 | モード端子<br>MD0 端子の制御により動作モードを変更します |
| MD1 | 入力 | モード端子<br>MD1 端子の制御により動作モードを変更します |
| P07 | 入力 | サンプルコードの動作選択用端子 |
| 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 |

# 5.　ソフトウェア説明

## 5.1　　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、

Ethernet での通信に

「RX ファミリ M3S-T4-Tiny: 導入ガイド」を、

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

「RX600 シリーズ RX600 用のシンプルフラッシュ API」を

使用しています。

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

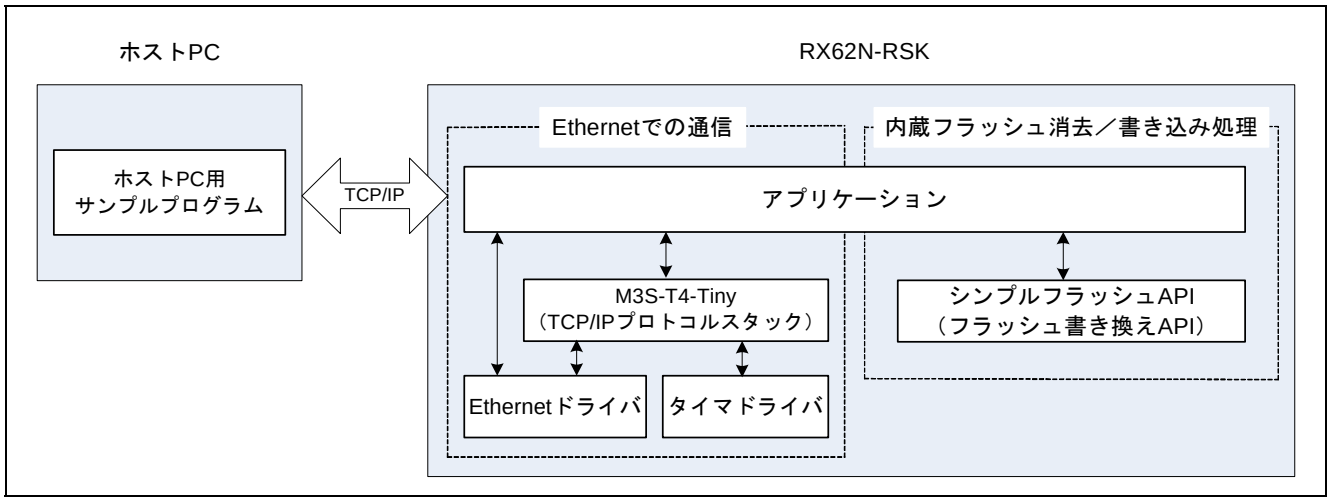


図5.1　サンプルコードのソフトウェア構成

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 | M3S-T4-Tiny 関数を呼び出し、ホスト PC から Ethernet 経由で S タイプフォーマットファイルを受信します。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、書き込みを行います。 |
| M3S-T4-Tiny | TCP/IP プロトコルスタックです。 |
| Ethernet ドライバ | イーサネットコントローラ（ETHERC）とイーサネット用 DMA コントローラ（EDMAC）を使用したドライバです。 |
| タイマドライバ | コンペアマッチタイマ（CMT）を使用したドライバです。 |
| シンプルフラッシュ API | 内蔵フラッシュメモリの消去／書き込みを行う API です。 |
| ホスト PC 用サンプルプログラム | ホスト PC 用サンプルプログラムです。<br>ホスト PC と RX62N-RSK との間で、Ethernet を用いた TCP/IP プロトコルによる通信を行い、ホスト PC 内の S タイプフォーマットファイルを RX62N-RSK に送信します。<br>詳細は、「7.　ホスト PC 用サンプルプログラムの例」を参照してください。 |

## 5.2　サンプルコードのフォルダ構成

図 5.2にサンプルコードのフォルダ構成を示します。

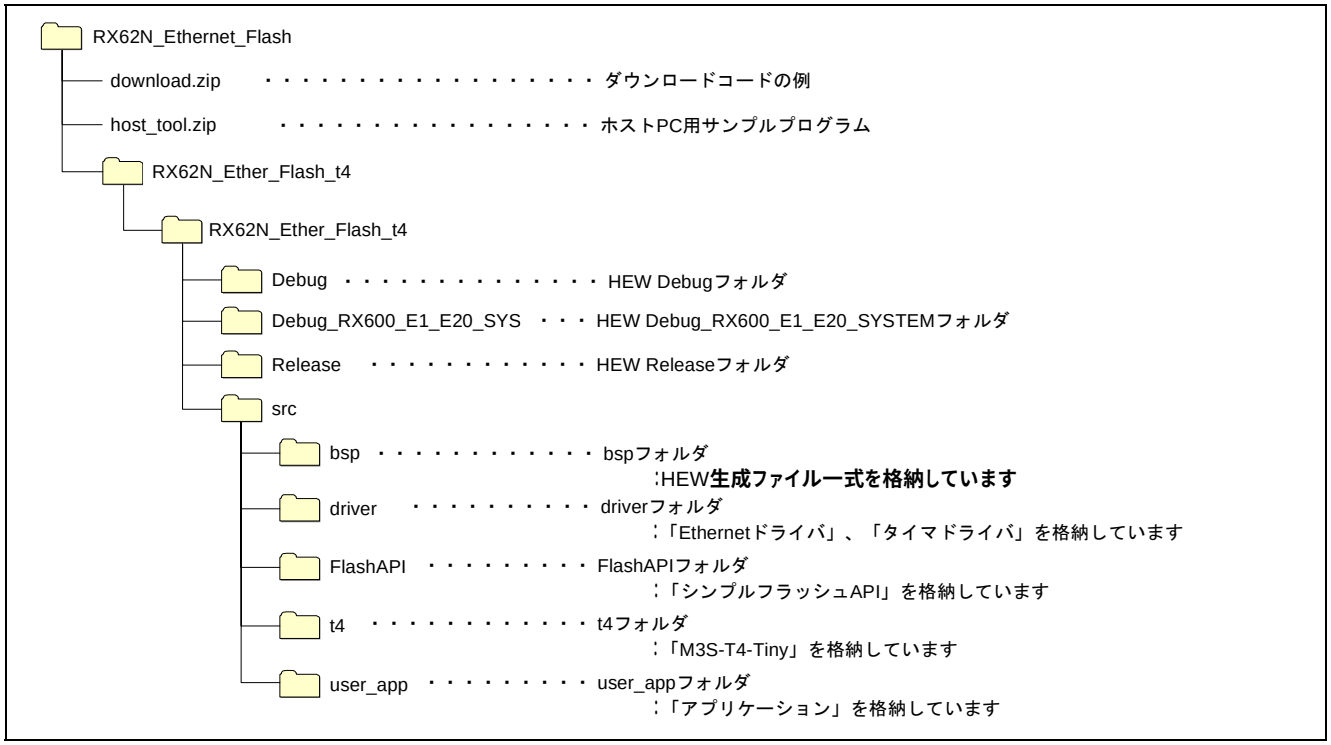


図5.2　サンプルコードのフォルダ構成

## 5.3　動作概要

### 5.3.1　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 RX62N-RSK のスイッチ SW3（マイコン P07 端子）の状態を確認します。このときスイッチ SW3 を押していない状態（マイコン P07 端子=H）であれば、M3S-T4-Tiny による Ethernet フラッシュブートローダを実行し、Ethernet 経由で内蔵フラッシュメモリを書き換えます。また、スイッチ SW3 を押した状態（マイコン P07 端子=L）であればダウンロードコードを実行します。

図 5.3にリセット解除後の動作を示します。

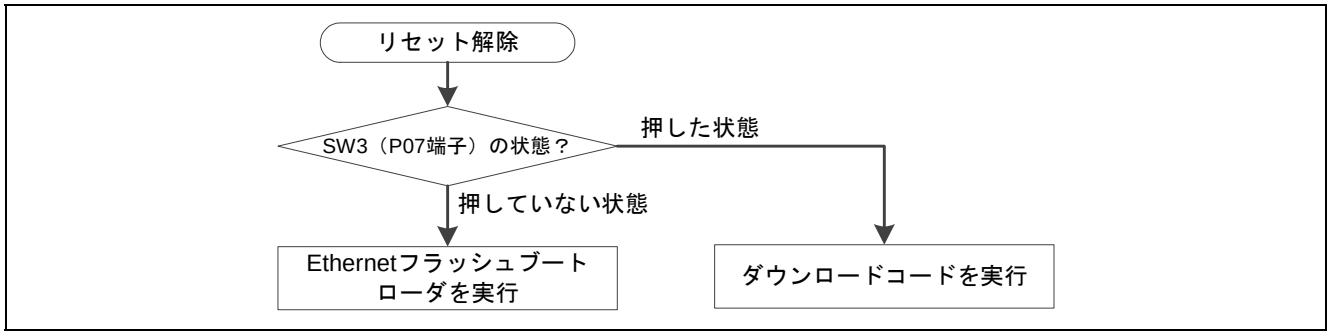


図5.3　リセット解除後の動作

### 5.3.2　書き換え対象

M3S-T4-Tiny による Ethernet 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（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（FFFF A000h～FFFF FFFFh）の書き換えは行いません。

図 5.4にメモリ配置を示します。

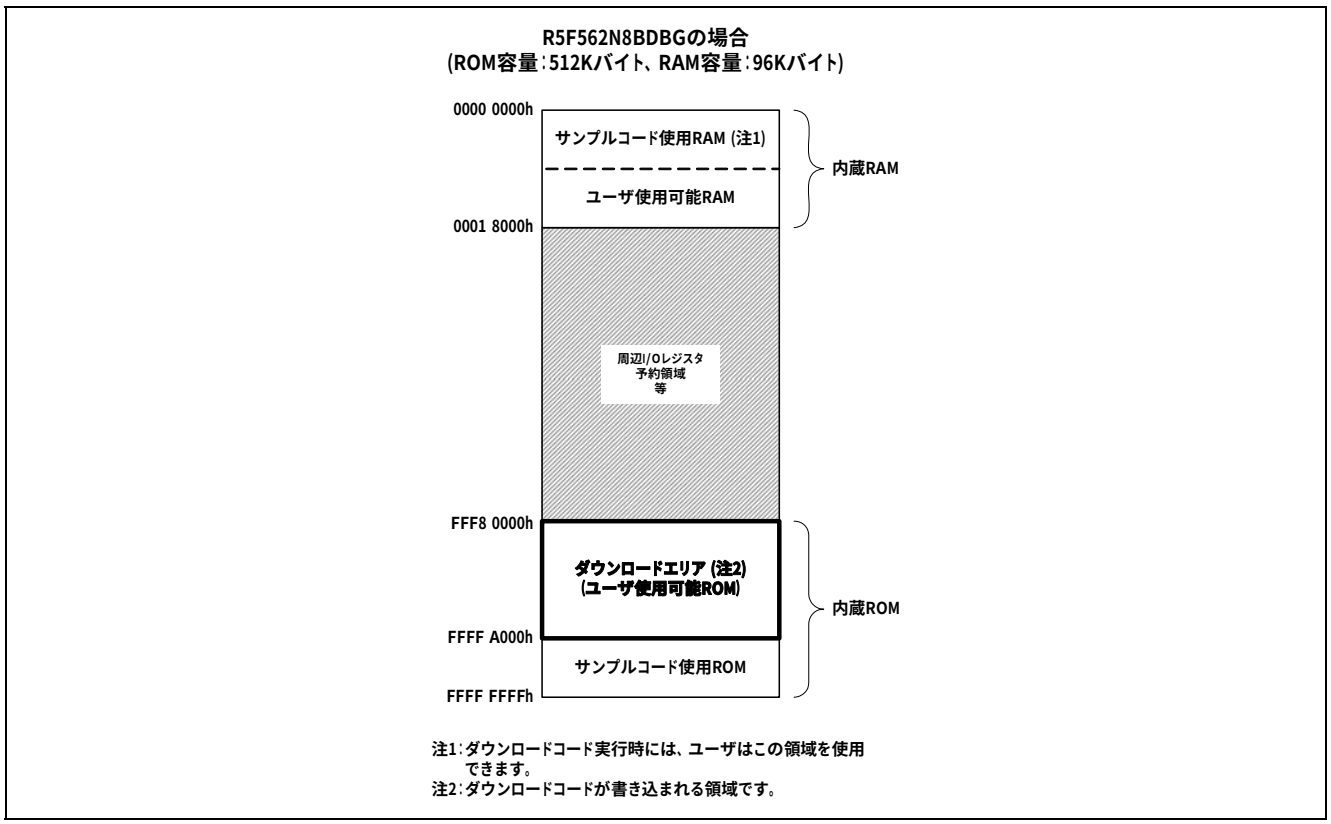


図5.4　メモリ配置

### 5.3.3　M3S-T4-Tiny による Ethernet フラッシュブートローダの動作

M3S-T4-Tiny による Ethernet 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 5.5にダウンロードエリアの書き換え動作を示します。

＜M3S-T4-Tiny による Ethernet フラッシュブートローダを実行するための前準備＞

① PC の IP アドレスとサブネットマスクを以下のように設定し、PC と RX62N-RSK を Ethernet ケーブルで接続してください。
　IP アドレス ：192.168.0.2
　サブネットマスク：255.255.255.0
② PC にてコマンドプロンプトを開き、ホスト PC 用サンプルプログラム（RX62N-test_client.exe）のある場所まで移動してください。

＜M3S-T4-Tiny による Ethernet フラッシュブートローダの実行＞

① 前準備を実行後、図 5.6のように RX62N-RSK の IP アドレス「192.168.0.3」とポート番号「1024」、書き込む S タイプフォーマットファイル（ダウンロードコード）の名前「任意のファイル名」を引数に指定してホスト PC 用サンプルプログラム（RX62N-test_client.exe）を実行してください。
② RX62N-RSK のリセット解除を行うと、PC と RX62N-RSK 間での TCP 接続が確立して、ホスト PC 用サンプルプログラムにてファイルサイズの確認の後、RX62N-RSK にファイルサイズを送信します。
③ ファイルサイズを受信した RX62N-RSK は、ダウンロードエリアを消去し、消去完了通知をホスト PC に送信します。
④ 消去完了通知を受信したホスト PC は S タイプフォーマットデータを RX62N-RSK に送信します。
⑤ RX62N-RSK は S タイプフォーマットデータを最大 1,400byte ずつ受信し、データの解析処理を行った後、256byte 単位でダウンロードエリアに書き込みます。
⑥ S タイプフォーマットのエンドレコード（S7 or S8 or S9 レコード）を検出するまで、④、⑤の処理を繰り返します。
⑦ RX62N-RSK 上の LED に書き込み結果を表示し、書き換え処理の正常終了または異常終了をホスト PC に送信します。
⑧ 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、ホスト PC に図 5.7のような結果を表示します。

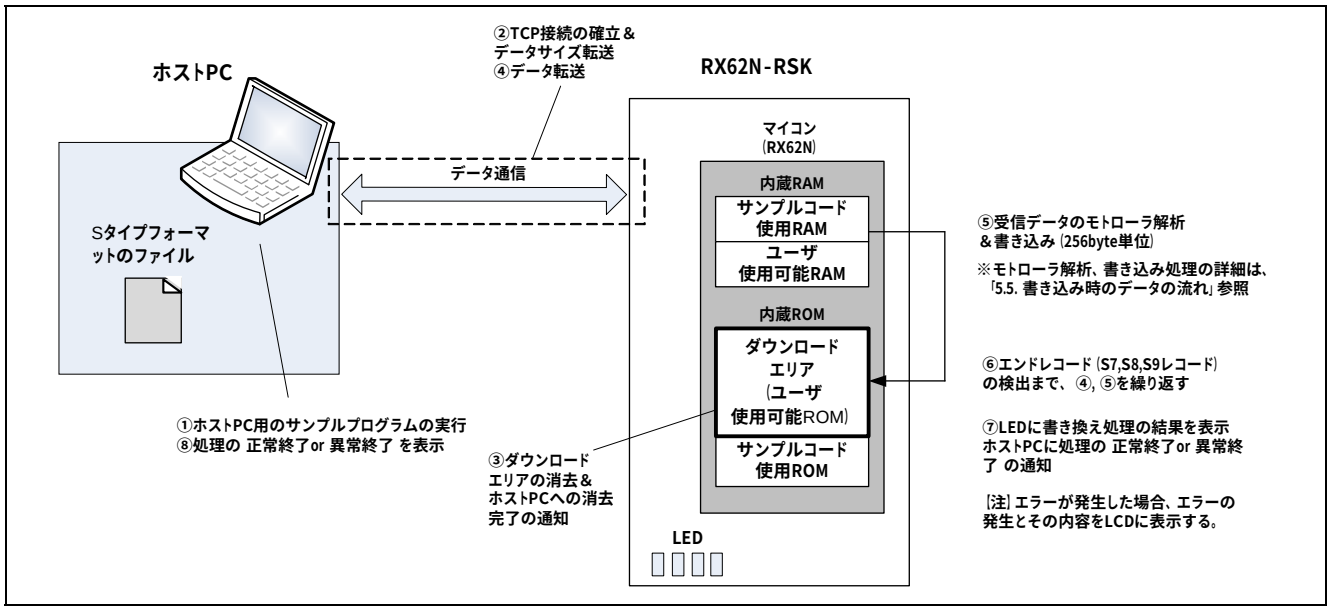


図5.5　ダウンロードエリアの書き換え動作

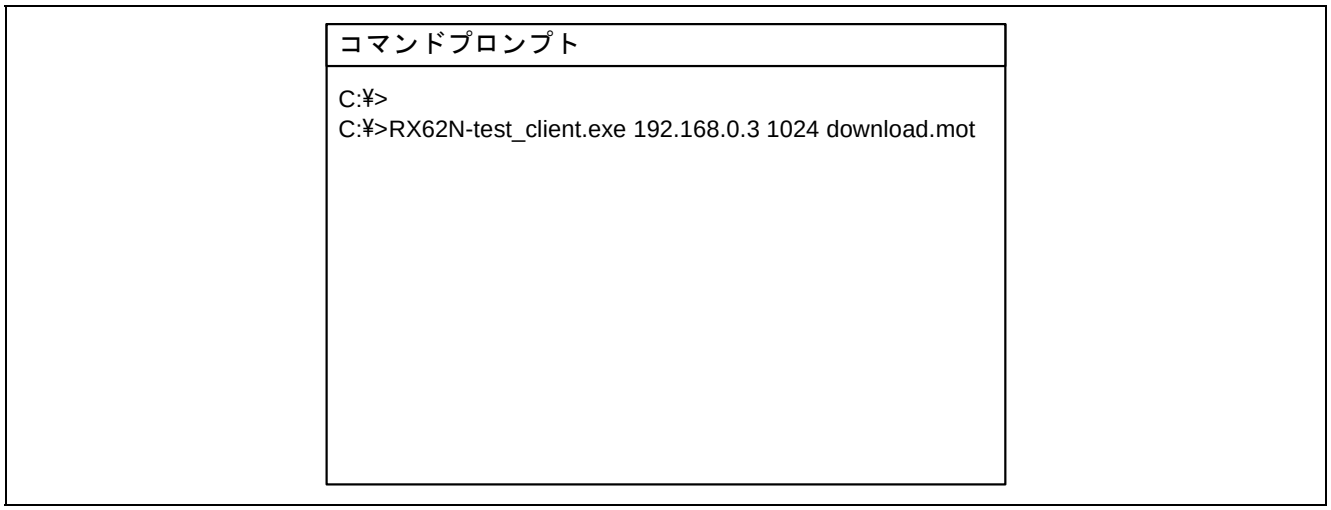


図5.6　ホスト PC 用サンプルプログラム 引数入力

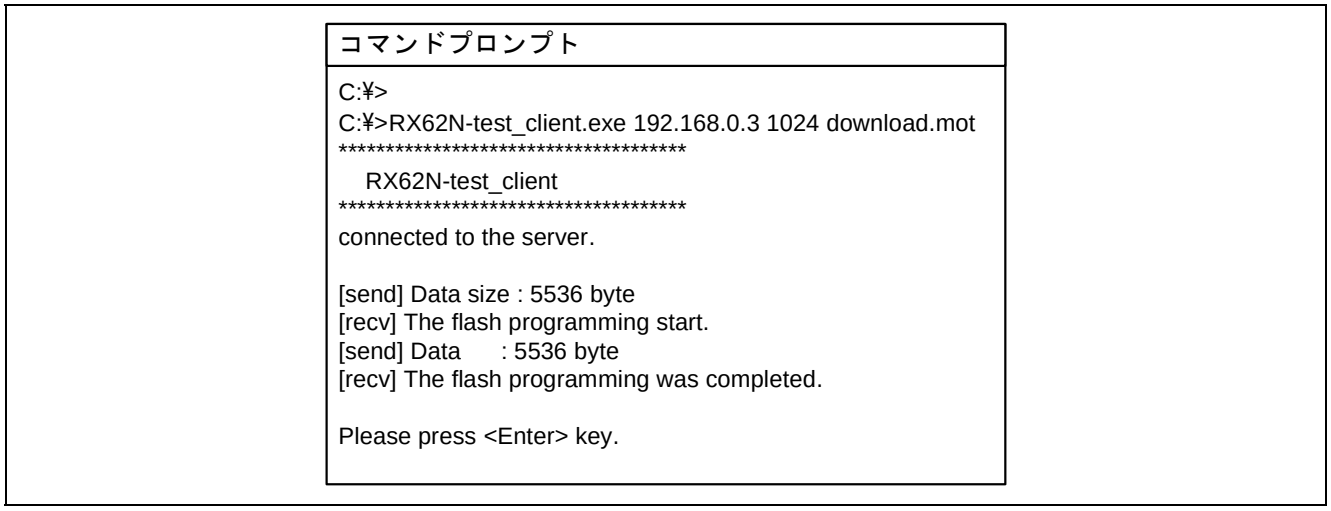


図5.7　ホスト PC 用サンプルプログラム 実行結果

### 5.3.4　M3S-T4-Tiny による Ethernet フラッシュブートローダの動作

ホスト PC を含めた全体のフローを、図 5.8および図 5.9に示します。なお、ホスト PC と RX62N-RSK 間の矢印は、TCP による通信を示します。

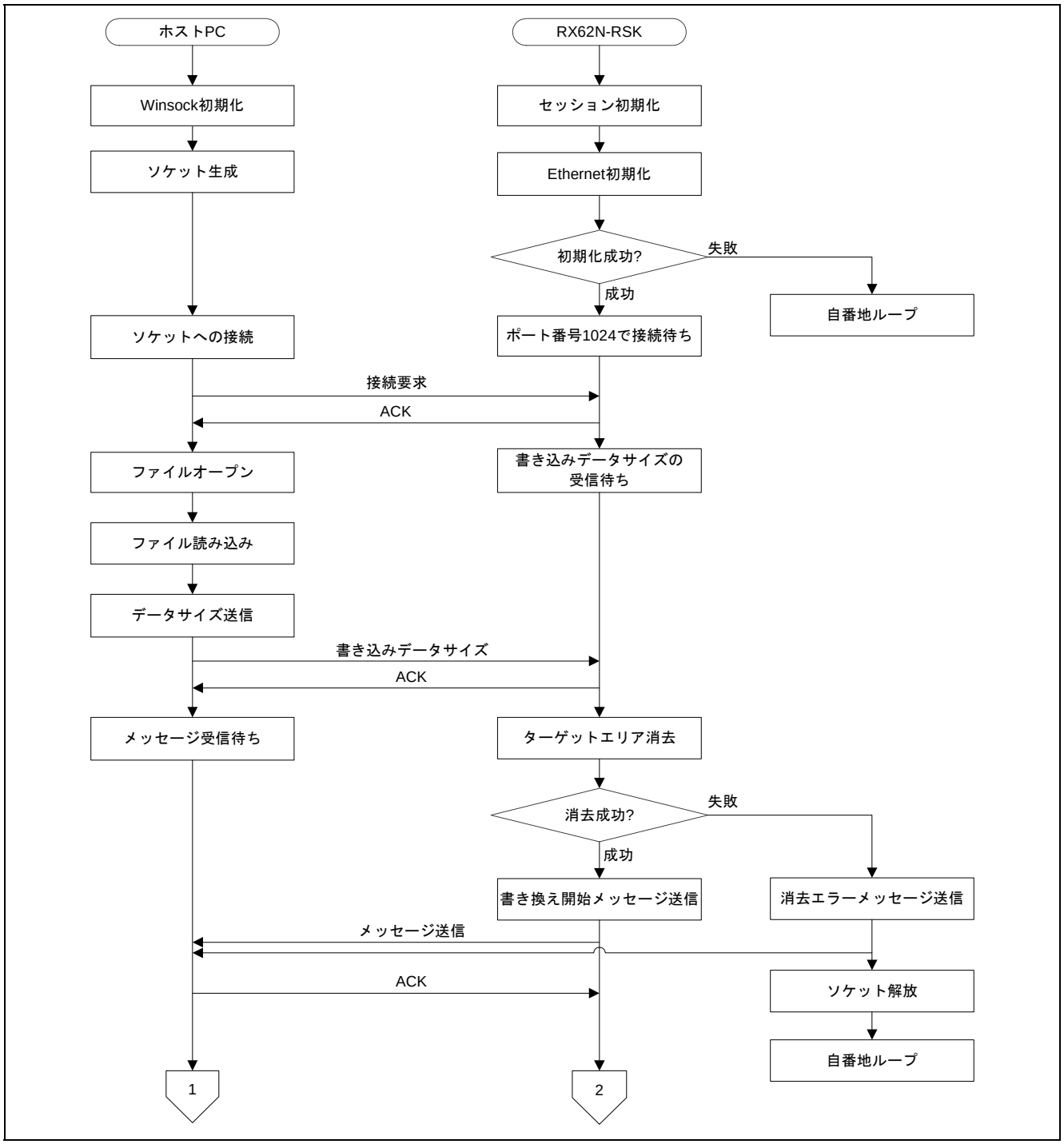


図5.8　全体のフロー図

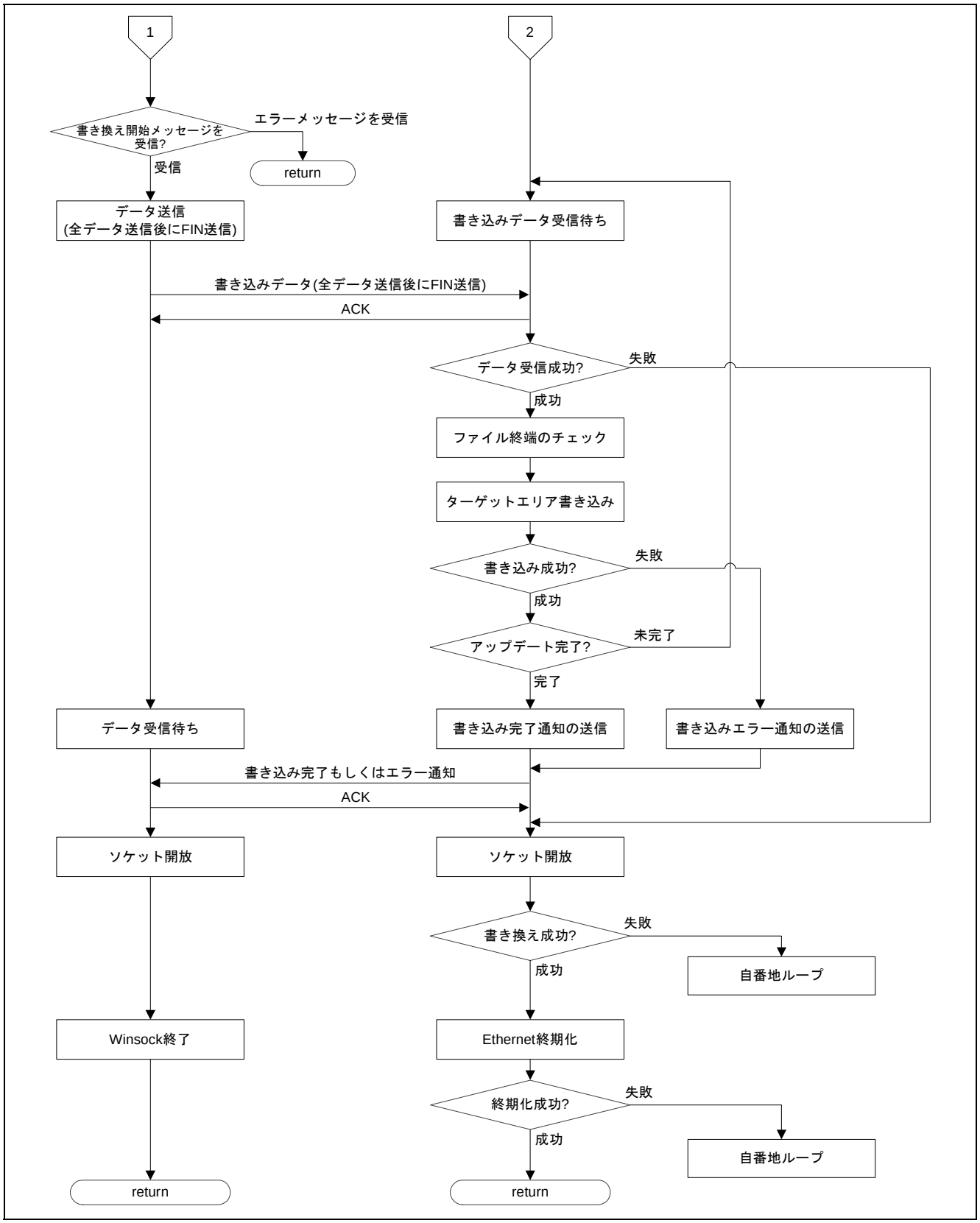


図5.9　全体のフロー図（続き）

## 5.4　ダウンロードコードの実行

サンプルコードは、マイコンのリセット解除後、スイッチ SW3 を押した状態（マイコン P07 端子=L）である場合、ダウンロードコードを実行します。

### 5.4.1　ダウンロードコードの実行開始位置

サンプルコードは、アドレス"FFFF 9FFCh"に書かれているアドレスを関数呼び出しすることでダウンロードコードを実行します。したがって、ダウンロードコードには、"FFFF 9FFCh"に開始アドレスを格納してください。

図 5.10にダウンロードコードの実行開始位置を示します。

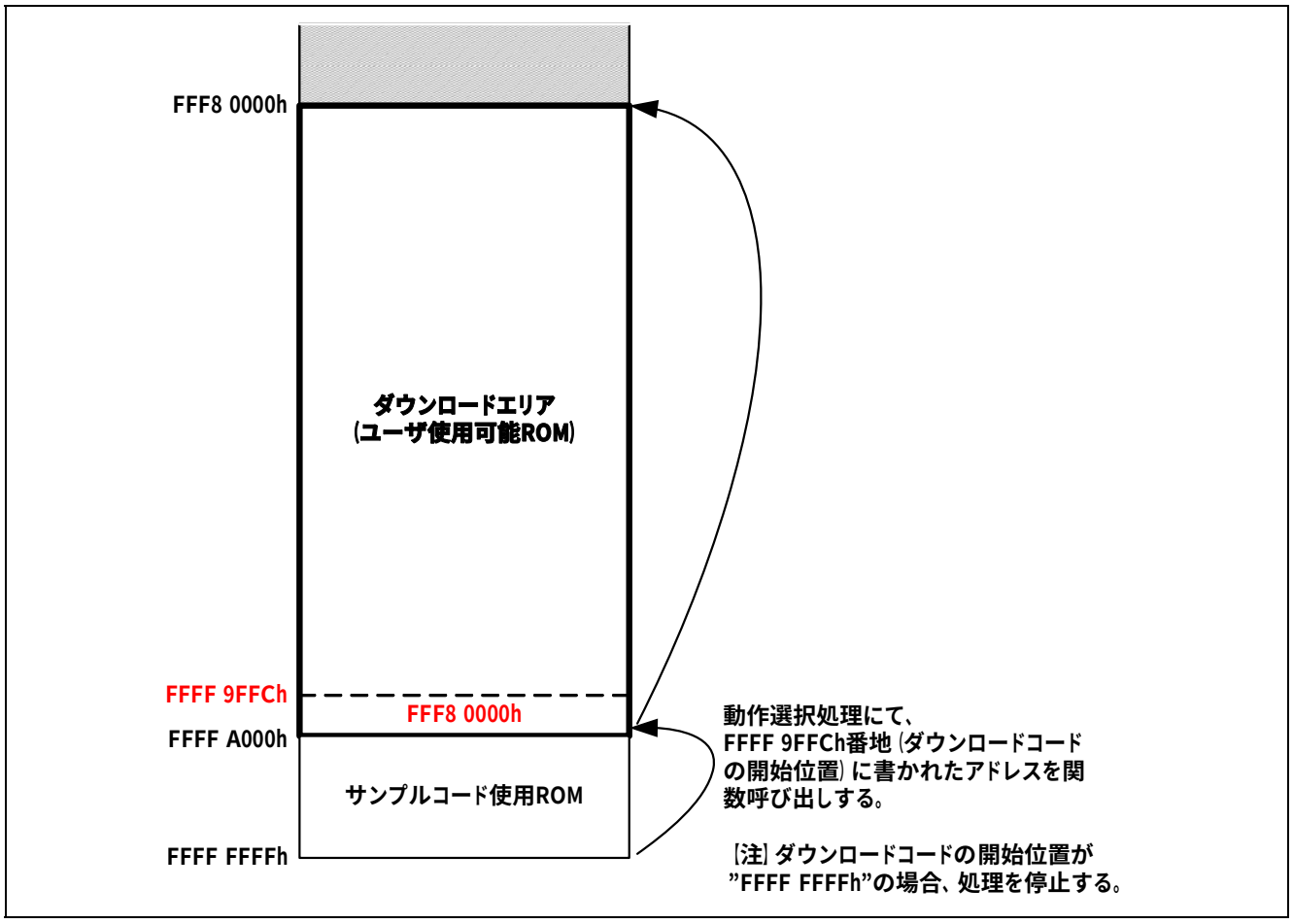


図5.10　ダウンロードコードの実行開始位置

【注】　ダウンロードコードの開始位置に何も書かれていない場合（ダウンロードコードの開始位置が"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。

## 5.5　書き込み時のデータの流れ

ダウンロードコードの書き込み時におけるマイコン内部のデータの流れを、図 5.11に示します。

① Ethernet 経由にて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 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。

③ 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における S タイプフォーマットの解析仕様は、「8.　S タイプフォーマット」を参照してください。

④ 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を 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②～④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。

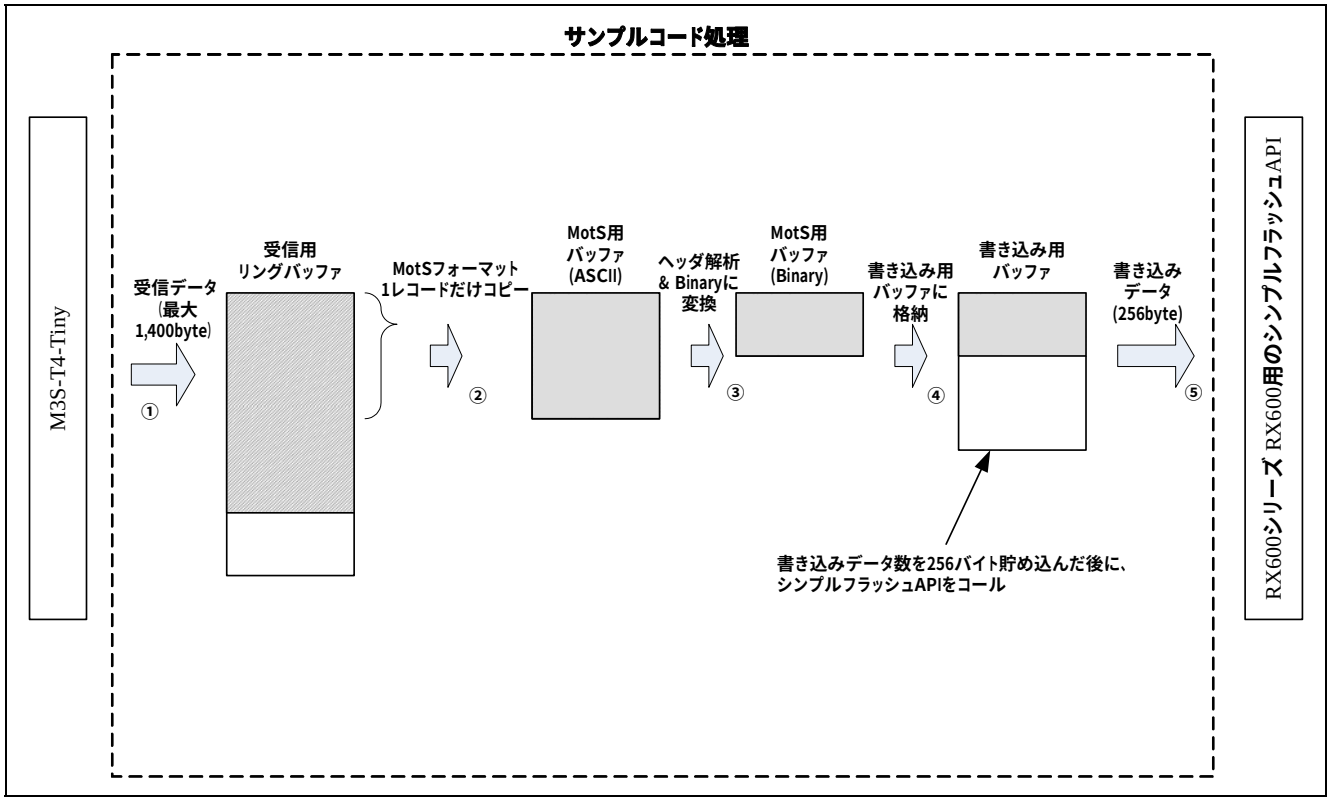


図5.11　書き込み時のデータの流れ

図 5.12に書き込み時のデータ構造を示します。

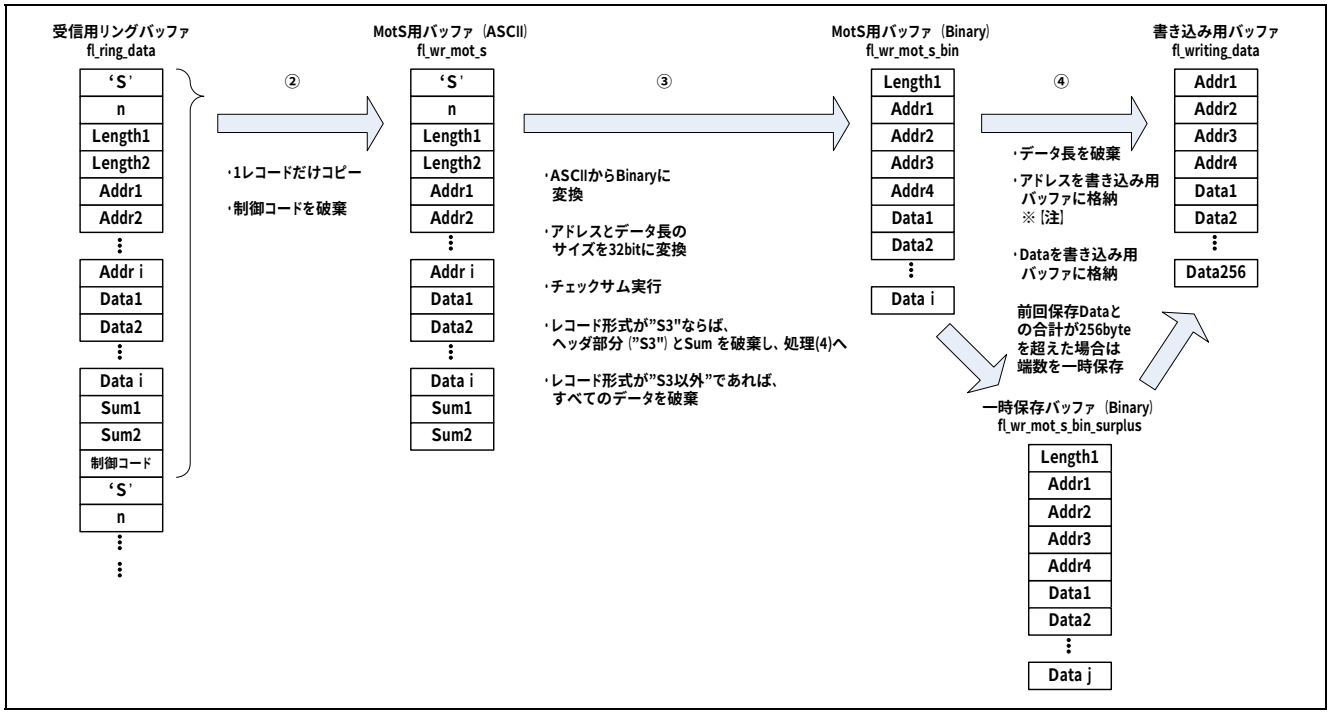


図5.12　書き込み時のデータ構造

【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、「5.13.12　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。

## 5.6　サンプルコードの LED 表示

サンプルコードでは、内蔵フラッシュメモリへの書き込み結果を RX62N-RSK 上の発光ダイオード（LED）に表示します。なお、ダウンロード処理の実行中は、LED（LED0～LED3）は消灯状態となっています。

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 | | | | 内容 |
|---|---|---|---|---|
| LED3 | LED2 | LED1 | LED0 | |
| ○ | ○ | ○ | ○ | 内蔵フラッシュメモリの書き換えに成功した場合に表示されます。（書き換え成功） |
| ● | ● | ● | ○ | Ethernet 初期化に失敗した場合に表示されます。（イーサネット初期化エラー） |
| ● | ● | ○ | ● | Ethernet 終期化に失敗した場合に表示されます。（イーサネット終期化エラー） |
| ● | ● | ○ | ○ | Ethernet 接続に失敗した場合に表示されます。（イーサネット接続エラー） |
| ● | ○ | ● | ● | Ethernet 受信に失敗した場合に表示されます。（イーサネット受信エラー） |
| ● | ○ | ● | ○ | Ethernet 切断に失敗した場合に表示されます。（イーサネット切断エラー） |
| ● | ○ | ○ | ● | 内蔵フラッシュメモリの消去に失敗した場合に表示されます。（イレーズエラー） |
| ● | ○ | ○ | ○ | 内蔵フラッシュメモリの書き込みに失敗した場合に表示されます。（ライトエラー） |
| ○ | ● | ● | ● | 内蔵フラッシュメモリ書き込み後のベリファイ確認に失敗した場合に表示されます。（ベリファイエラー） |
| ○ | ● | ● | ○ | ファイルの終端まで処理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がなかった場合に表示されます。（ファイルエンドエラー） |
| ○ | ● | ○ | ● | ダウンロードコード実行時、ダウンロードコードの開始位置が"FFFFFFFFh"だった場合に表示されます。（ダウンロードコード未書き込みエラー） |
| ○ | ● | ○ | ○ | S タイプフォーマットのチェックサム検査で異常があった場合に表示されます。（チェックサムエラー）「8.　S タイプフォーマット」を参照してください。 |
| ○ | ○ | ● | ● | サンプルコードが非対応となる S タイプフォーマットだった場合に表示されます。（フォーマットエラー）「8.　S タイプフォーマット」を参照してください。 |
| ○ | ○ | ● | ○ | ダウンロードエリア外への書き込みデータを検出した場合に表示されます。（アドレスエラー）「8.　S タイプフォーマット」を参照してください。 |

## 5.7　必要メモリサイズ

表 5.3に必要メモリサイズを示します。

表5.3　必要メモリサイズ

| 使用メモリ | サイズ | 備考 |
|---|---|---|
| ROM | 24,318byte | サンプルコードは FFFF A000h ～FFFF FFFFh 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 － 24576byte」となります。 |
| RAM | 40,796byte | 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す。 |

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。

## 5.8　ファイル構成

表 5.4にサンプルコードで使用するファイルを示します。

なお、総合開発環境で自動生成されるファイル、ダウンロードコードの例およびホスト PC 用サンプルプログラムは除きます。

表5.4　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 | Ethernet との受信用リングバッファ関連処理 | |
| r_Flash_buff.h | Ethernet との受信用リング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 | |
| main.c | main 関数 | |
| その他ファイル | 「RX ファミリ M3S-T4-Tiny: 導入ガイド」のプログラム | 詳細は「RX ファミリ M3S-T4-Tiny: 導入ガイド」のアプリケーションノートを参照してください。 |

## 5.9　定数一覧

表 5.5にサンプルコードで使用する定数を示します。

表5.5　サンプルコードで使用する定数

| 定数名 | 設定値 | 内容 |
|---|---|---|
| FL_T4_API_TIMEOUT | 1000 | M3S-T4-Tiny 関数のタイムアウト時間 |
| FL_INPUT_BUFSIZE | 1400 | Ethernet からのデータ受信用バッファサイズ |
| FL_RINGBUFF_SIZE | 1400 | Ethernet 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 | 6 | 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 | FFFF9FFFh | ダウンロードエリアの最後尾アドレス |
| FL_RCV_BLANK_SIZE | 1400 | リングバッファの補充可能サイズ |

## 5.10　構造体/共用体一覧

図 5.13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5.13　サンプルコードで使用する構造体／共用体

## 5.11　関数一覧

表 5.6に関数を示します。ただし、シンプルフラッシュ API、TCP/IP プロトコルスタック、Ethernet ドライバで使用しているものは除きます。

表5.6　関数

| 関数名 | 概要 |
|---|---|
| R_Fl_Mode_Entry | 動作選択処理 |
| R_Fl_Ether_Sample_Init | Ethernet の初期化 |
| R_Fl_Ether_Sample_Quit | Ethernet の終期化 |
| R_Fl_Flash_Update | 書き換え処理のメインとなる関数 |
| R_Fl_EraseTrgtArea | 消去処理 |
| R_Fl_Ers_EraseFlash | ダウンロードエリア消去 |
| R_Fl_PrgramTrgtArea | ダウンロードエリアへの書き込み |
| R_Fl_Prg_PrgramFlash | 書き込み処理 |
| R_Fl_Prg_StoreMotS | S タイプフォーマットデータ格納 |
| R_Fl_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_Fl_Prg_MotS_AsciiToBinary | S タイプフォーマットデータの ASCII - Binary 変換 |
| R_Fl_Prg_MakeWriteData | ダウンロードエリアへの書き込みデータ作成 |
| R_Fl_Prg_WriteData | ダウンロードエリアへの書き込み |
| R_Fl_Prg_ClearMotSVariables | S タイプフォーマットデータ関連の変数クリア |
| R_Fl_RcvDataString | Ethernet 受信データ格納 |
| R_Fl_RingCheckBlank | Ethernet 受信データ格納用リングバッファの空き容量確認 |
| R_Fl_RingInit | Ethernet 受信データ格納用リングバッファの初期化 |
| R_Fl_RingEnQueue | Ethernet 受信データ格納用リングバッファへのデータ格納 |
| R_Fl_RingDeQueue | Ethernet 受信データ格納用リングバッファからのデータ読み出し |
| R_Fl_RingCheck | Ethernet 受信データ格納用リングバッファのデータ数確認 |
| R_Fl_AsciiToHexByte | ASCII コードを Binary データへ変換 |
| R_Fl_LED_Ini | LED 初期設定処理 |
| R_Fl_LED_Fnc | LED 消灯／点灯処理 |

## 5.12　関数仕様

サンプルコードの関数仕様を示します。

| R_Fl_Mode_Entry | |
|---|---|
| 概要 | 動作選択処理 |
| ヘッダ | r_Flash_main.h |
| 宣言 | void R_Fl_Mode_Entry(void) |
| 説明 | • 動作の選択を行います。<br>• LED の初期設定処理を行います。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | • RX62N-RSK のスイッチ SW3 を押した状態であるとダウンロードコードを実行します。SW3 を押していない状態であると、本関数のリターン後にM3S-T4-Tiny による Ethernet フラッシュブートローダの処理に移ります。 |

| R_Fl_Ether_Sample_Init | |
|---|---|
| 概要 | Ethernet の初期化 |
| ヘッダ | r_Flash_main.h |
| 宣言 | Fl_API_SMPL_rtn_t R_Fl_Ether_Sample_Init(void) |
| 説明 | • LAN コントローラの初期化／起動を行う関数をコールします。<br>• M3S-T4-Tiny の初期化を行う関数をコールします。<br>• RX62N-RSK 上の LED 表示を初期化します。 |
| 引数 | なし |
| リターン値 | • 初期化が正常に完了した場合：FLASH_API_SAMPLE_OK<br>• 初期化が正常に完了しなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_Ether_Sample_Quit | |
|---|---|
| 概要 | Ethernet の終期化 |
| ヘッダ | r_Flash_main.h |
| 宣言 | Fl_API_SMPL_rtn_t R_Fl_Ether_Sample_Quit(void) |
| 説明 | • M3S-T4-Tiny の終了処理を行う関数をコールします。<br>• LAN コントローラの停止を行う関数をコールします。 |
| 引数 | なし |
| リターン値 | • 終期化が正常に完了した場合：FLASH_API_SAMPLE_OK<br>• 終期化が正常に完了しなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_Flash_Update | |
|---|---|
| 概要 | 書き換え処理メイン |
| ヘッダ | r_Flash_main.h |
| 宣言 | Fl_API_SMPL_rtn_t R_Fl_Flash_Update(void) |
| 説明 | • M3S-T4-Tiny 関数をコールし、ホスト PC から S タイプフォーマットファイルを受信します。<br>• 受信した S タイプフォーマットファイルで内蔵フラッシュメモリを書き換える関数をコールします。 |
| 引数 | なし |
| リターン値 | • 書き換え処理が正常に完了した場合：FLASH_API_SAMPLE_OK<br>• 書き換え処理が正常に完了しなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_EraseTrgtArea | |
|---|---|
| 概要 | 消去処理 |
| ヘッダ | なし |
| 宣言 | static Fl_API_SMPL_rtn_t R_Fl_EraseTrgtArea(void) |
| 説明 | • ダウンロードエリアを消去する関数を呼び出します。 |
| 引数 | なし |
| リターン値 | • 消去処理が正常に完了した場合：FLASH_API_SAMPLE_OK<br>• 消去処理が正常に完了しなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_Ers_EraseFlash | |
|---|---|
| 概要 | ダウンロードエリア消去 |
| ヘッダ | なし |
| 宣言 | static Fl_API_SMPL_rtn_t R_Fl_Ers_EraseFlash(void) |
| 説明 | • ダウンロード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<br>• 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_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 Fl_API_SMPL_rtn_t R_Fl_PrgramTrgtArea(void) |
| 説明 | • 書き込み処理を行う関数を呼び出します。 |
| 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<br>• 書き込みが正常に完了しなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_Prg_PrgramFlash | |
|---|---|
| 概要 | 書き込み処理 |
| ヘッダ | なし |
| 宣言 | static Fl_API_SMPL_rtn_t R_Fl_Prg_PrgramFlash(void) |
| 説明 | • 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。<br>• 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。<br>• ファイル終端に達した場合は、S タイプフォーマットのエンドレコードを受信しているかの確認を行います。（受信していない場合は、リターン値として FLASH_API_SAMPLE_NG を返します） |
| 引数 | なし |
| リターン値 | • 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<br>• ダウンロードエリアへの書き込みが完了しなかった場合：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 Fl_API_SMPL_rtn_t R_Fl_Prg_StoreMotS(uint8_t) |
| 説明 | • 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。<br>• 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　　：S タイプフォーマットデータ |
| リターン値 | • 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<br>• 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 | • 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。<br>• 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 Fl_API_SMPL_rtn_t R_Fl_Prg_ProcessForMotS_data(void) |
| 説明 | • 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。<br>• 書き込みバッファへデータを格納する関数をコールします。<br>• ダウンロードエリアへデータを書き込む関数をコールします。 |
| 引数 | なし |
| リターン値 | • 正常に完了した場合：FLASH_API_SAMPLE_OK<br>• S タイプフォーマットと異なるデータがあった場合：FLASH_API_SAMPLE_NG |
| 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 | • ASCII コードの S タイプフォーマットデータを Binary データに変換します。<br>• 変換した Binary データのチェックサムを確認します。 |
| 引数 | • 第一引数：*tmp_mot_s　：ASCII コードの S タイプフォーマットデータのポインタ<br>• 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<br>• チェックサムエラーが発生した場合：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 | • 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<br>• 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_Fl_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 Fl_API_SMPL_rtn_t R_Fl_Prg_WriteData(void) |
| 説明 | • ダウンロードエリア内への書き込みであるかを確認します。<br>• ダウンロードエリアに書き込みを行います。<br>• 書き込んだデータのベリファイを行います。<br>• 書き込みに失敗した場合、エラー関数をコールします。 |
| 引数 | なし |
| リターン値 | • 書き込みが正常に完了した場合：FLASH_API_SAMPLE_OK<br>• 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_Fl_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_Fl_Prg_ClearMotSVariables(void) |
| 説明 | • 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_Fl_RcvDataString | |
|---|---|
| 概要 | Ethernet 受信データ格納 |
| ヘッダ | なし |
| 宣言 | static Fl_API_SMPL_rtn_t R_Fl_RcvDataString(void *, uint16_t) |
| 説明 | • Ethernet にて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　：Ethernet にて受信したデータが格納されているバッファのポインタ<br>• 第二引数：length　：Ethernet にて受信したデータ数 |
| リターン値 | • 格納が完了した場合：FLASH_API_SAMPLE_OK<br>• 受信用リングバッファに空きがなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_RingCheckBlank | |
|---|---|
| 概要 | Ethernet 受信データ格納用リングバッファの空き容量確認 |
| ヘッダ | r_Flash_buff.h |
| 宣言 | Fl_API_SMPL_rtn_t R_Fl_RingCheckBlank(void) |
| 説明 | • Ethernet 受信データ格納用リングバッファに、Ethernet 経由で受信するデータ 1 回分（1,400byte）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<br>• 空きがなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_RingInit | |
|---|---|
| 概要 | Ethernet 受信データ格納用リングバッファの初期化 |
| ヘッダ | r_Flash_buff.h |
| 宣言 | void R_Fl_RingInit(void) |
| 説明 | • Ethernet 受信データ格納用リングバッファを初期化します。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | |

| R_Fl_RingEnQueue | |
|---|---|
| 概要 | Ethernet 受信データ格納用リングバッファへのデータ格納 |
| ヘッダ | r_Flash_buff.h |
| 宣言 | Fl_API_SMPL_rtn_t R_Fl_RingEnQueue(uint8_t) |
| 説明 | • Ethernet 受信データ格納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　：格納するデータ |
| リターン値 | • 格納完了の場合：FLASH_API_SAMPLE_OK<br>• バッファフルの場合：FLASH_API_SAMPLE_NG |
| 備考 | |

| R_Fl_RingDeQueue | |
|---|---|
| 概要 | Ethernet 受信データ格納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 | Fl_API_SMPL_rtn_t R_Fl_RingDeQueue(uint8_t *) |
| 説明 | • Ethernet 受信データ格納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<br>• 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 | Ethernet 受信データ格納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 | • Ethernet 受信データ格納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 | • 格納済みの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 | • 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）<br>• 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 | |

| R_Fl_LED_Ini | |
|---|---|
| 概要 | LED 初期設定処理 |
| ヘッダ | なし |
| 宣言 | static void R_Fl_LED_Ini(void) |
| 説明 | • RX62N-RSK 上の LED の初期設定処理を行います。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | |

| R_Fl_LED_Fnc | |
|---|---|
| 概要 | LED 消灯／点灯処理 |
| ヘッダ | なし |
| 宣言 | static Fl_API_SMPL_rtn_t R_Fl_LED_Fnc(uint8_t) |
| 説明 | • RX62N-RSK 上の LED の消灯／点灯処理を行います。<br>詳細は「5.6.　サンプルコードの LED 表示」を参照してください。 |
| 引数 | • 第一引数：in_data　　：LED の消灯／点灯設定値 |
| リターン値 | • 正常に完了した場合：FLASH_API_SAMPLE_OK<br>• 正常に完了しなかった場合：FLASH_API_SAMPLE_NG |
| 備考 | • 引数 in_data は各 bit がそれぞれの LED の消灯／点灯の設定値となっており、対応する LED は下記となります。なお、消灯する場合は「0」を、点灯する場合は「1」を各 bit に設定してください。<br>bit[0]：LED0、bit[1]：LED1、bit[2]：LED2、bit[3]：LED3 |

## 5.13　フローチャート

### 5.13.1　動作選択処理

図 5.14に動作選択処理のフローチャートを示します。

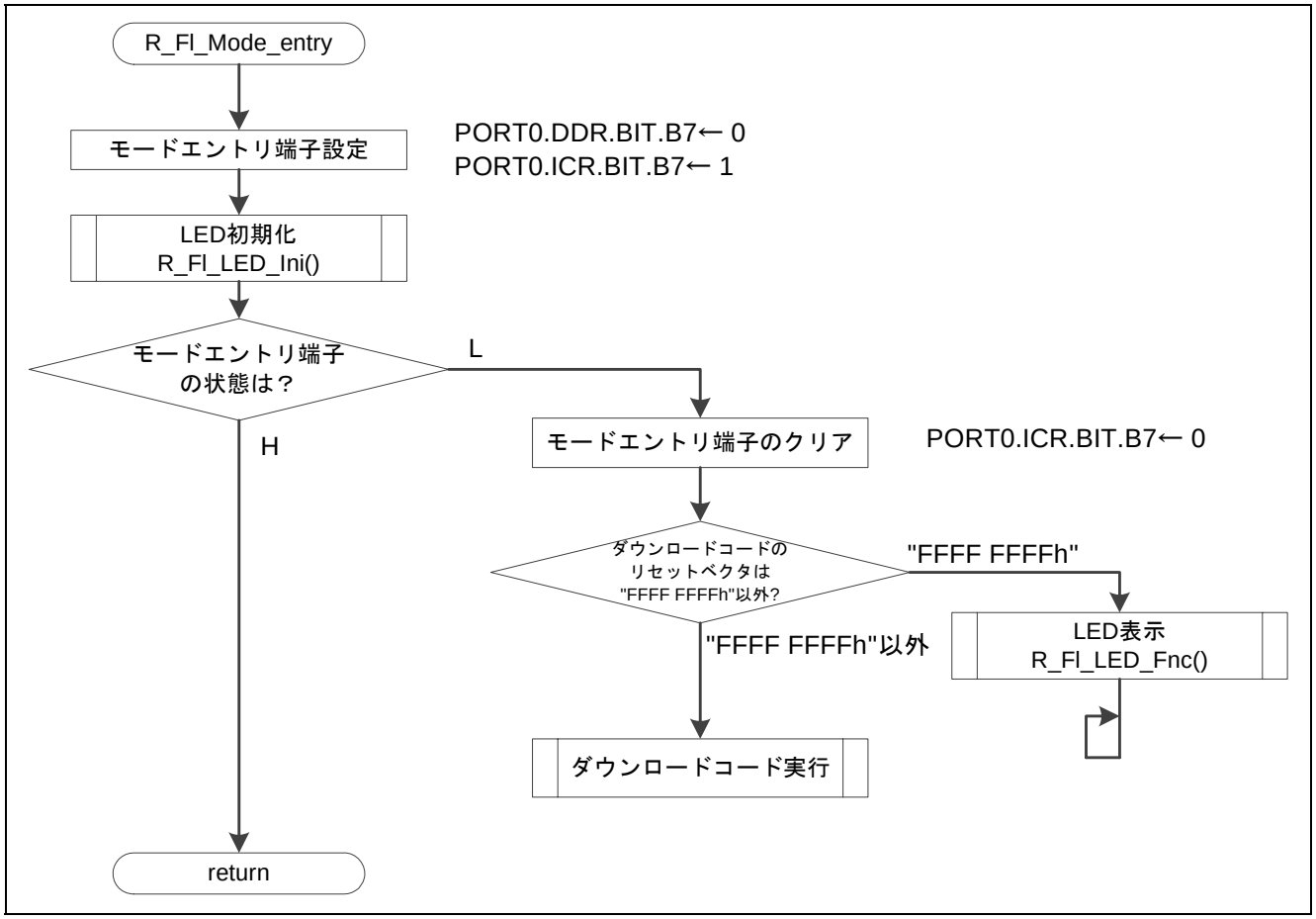


図5.14　動作選択処理

### 5.13.2　Ethernet 初期化処理

図 5.15にEthernet 初期化処理のフローチャートを示します。

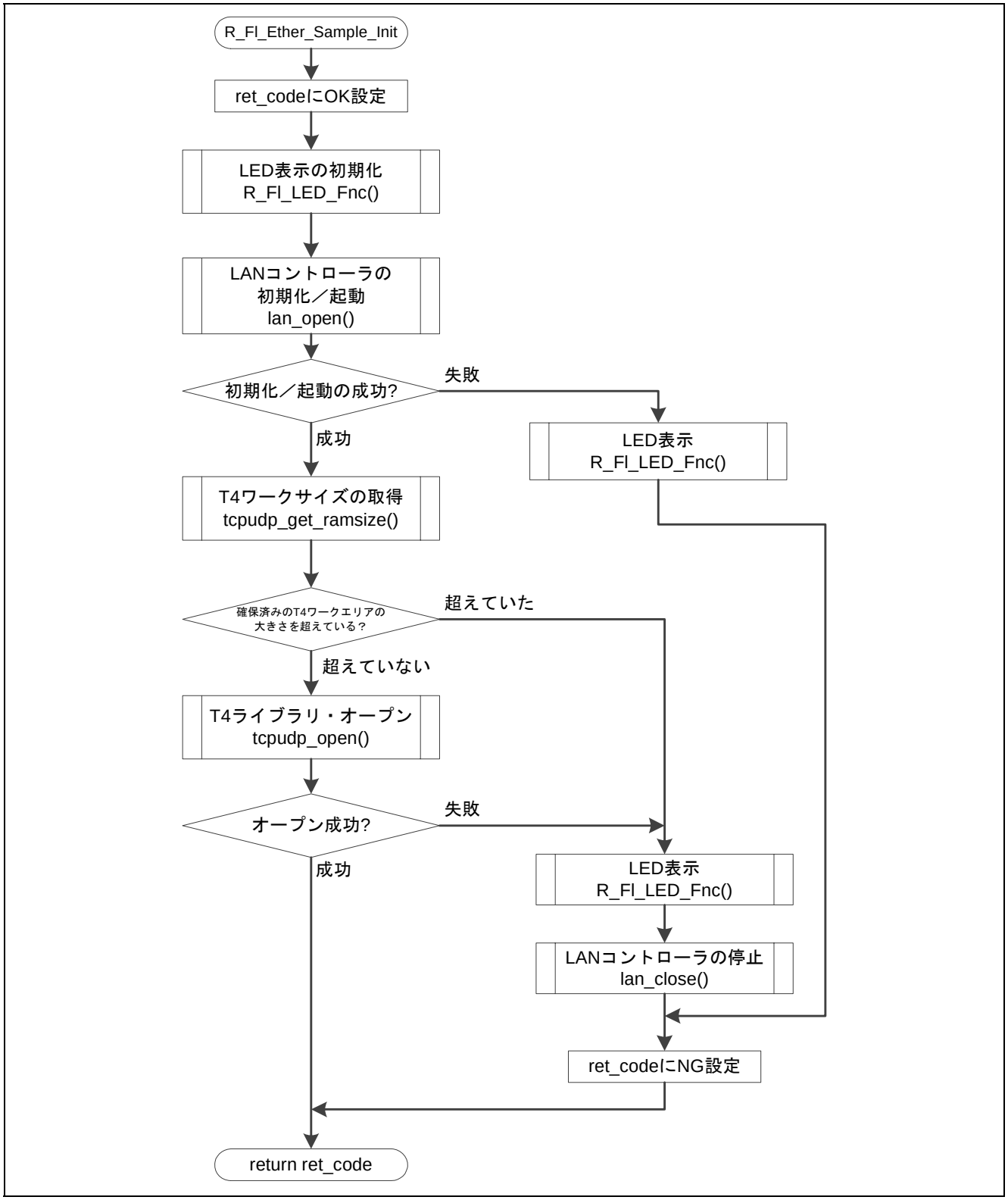


図5.15　Ethernet 初期化処理

### 5.13.3　Ethernet 終期化処理

図 5.16にEthernet 終期化処理のフローチャートを示します。

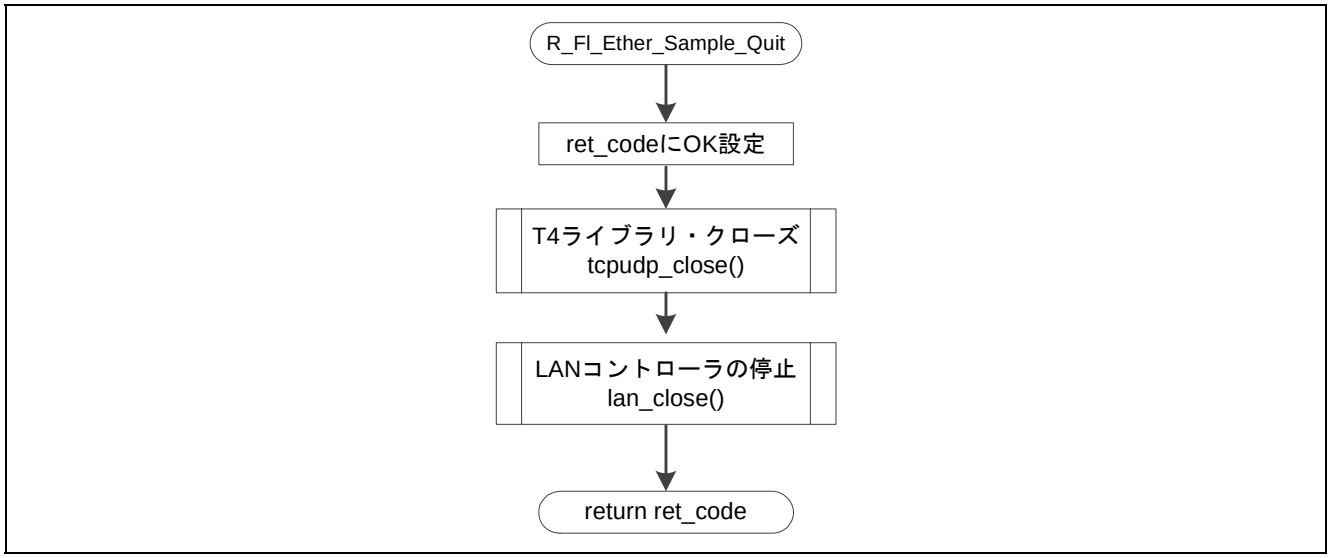


図5.16　Ethernet 終期化処理

### 5.13.4　書き換え処理メイン

図 5.17および図 5.18に書き換え処理メインのフローチャートを示します。

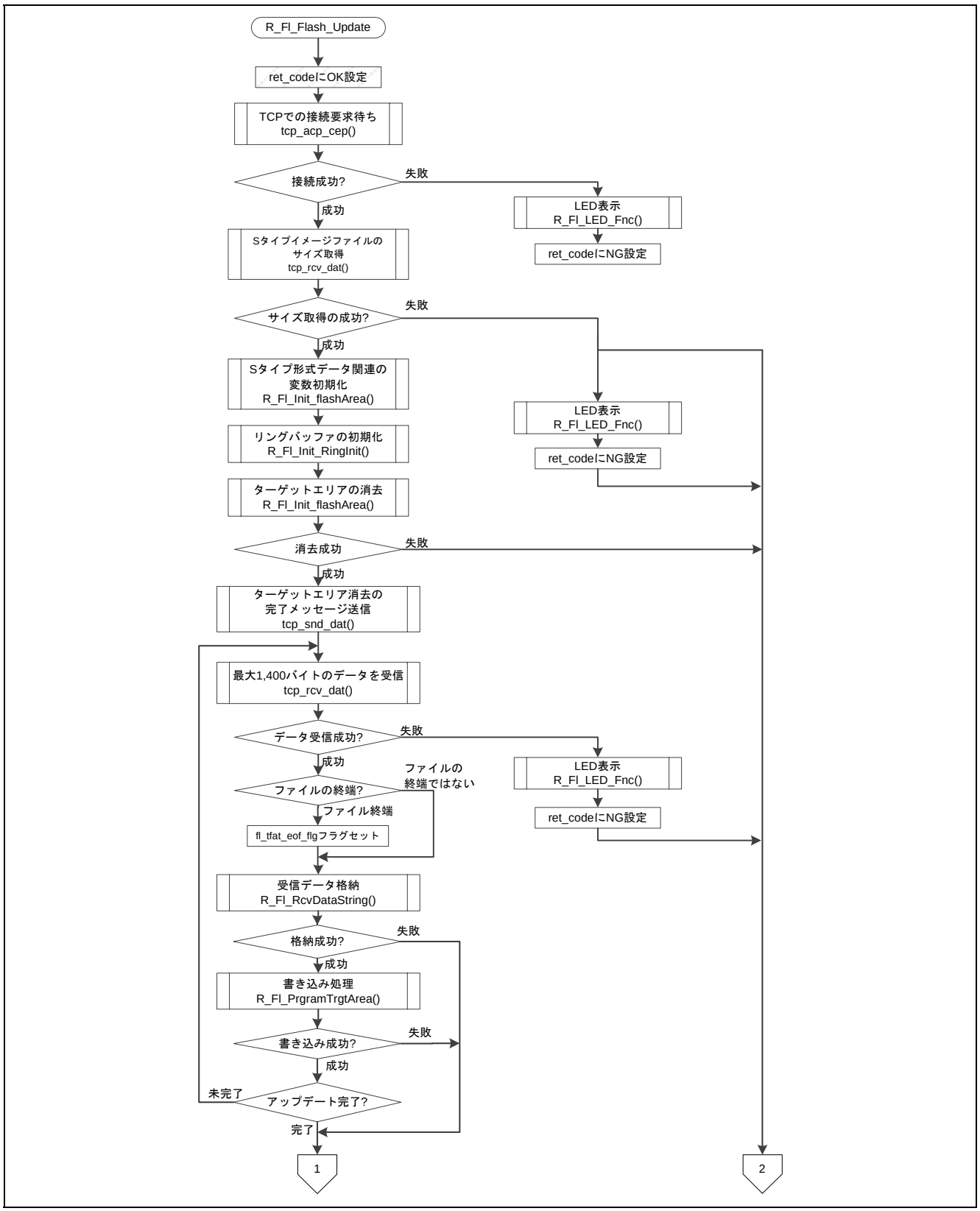


図5.17　書き換え処理メイン

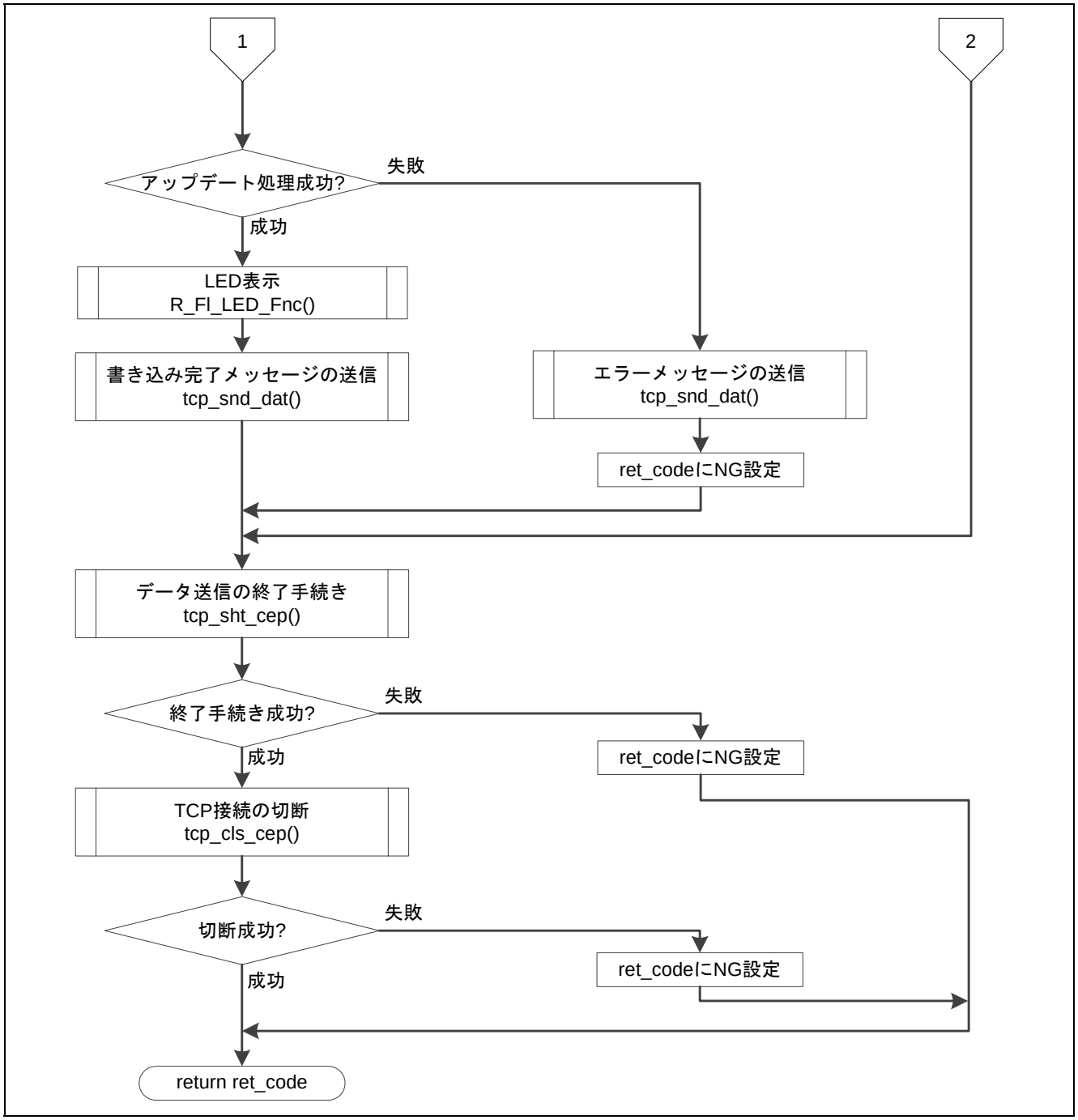


図5.18　書き換え処理メイン（続き）

### 5.13.5　消去処理

図 5.19に消去処理のフローチャートを示します。

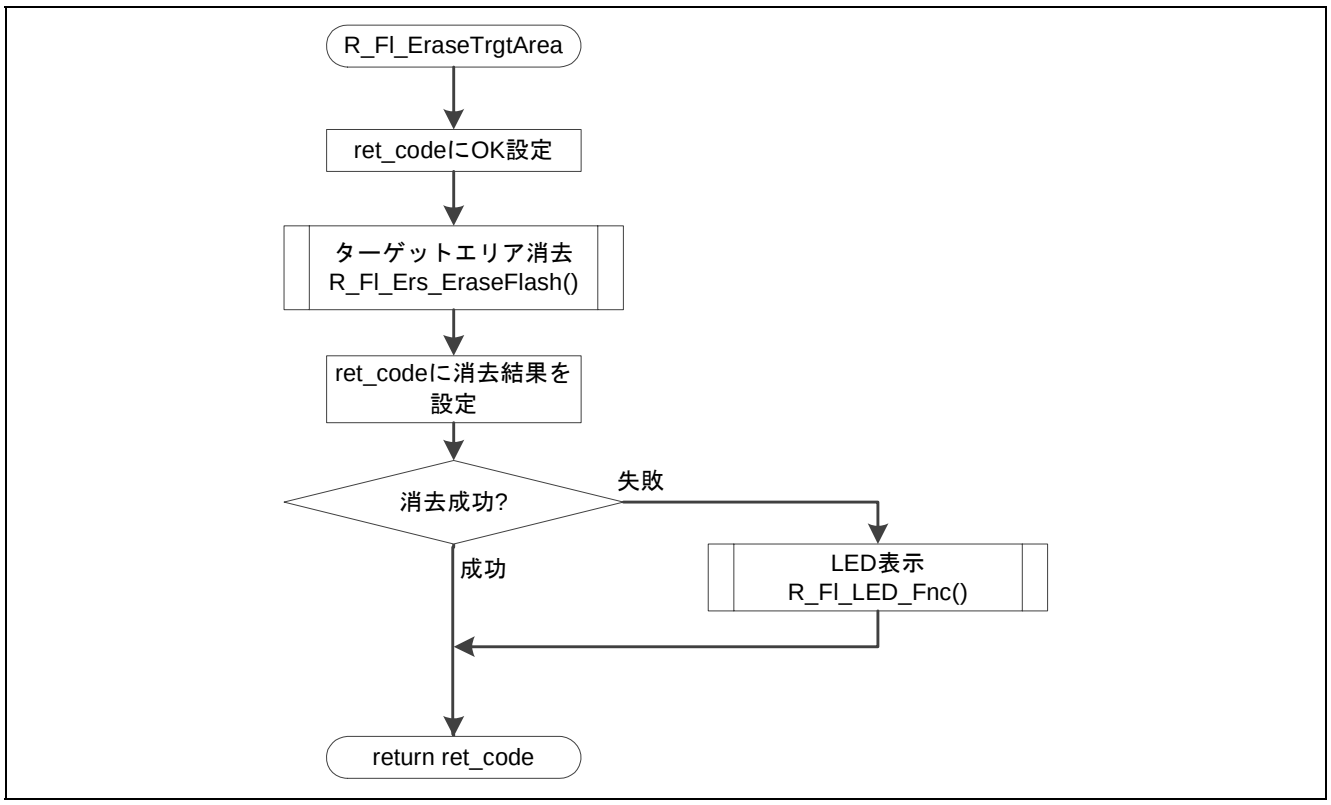


図5.19　消去処理

### 5.13.6　ダウンロードエリア消去

図 5.20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。

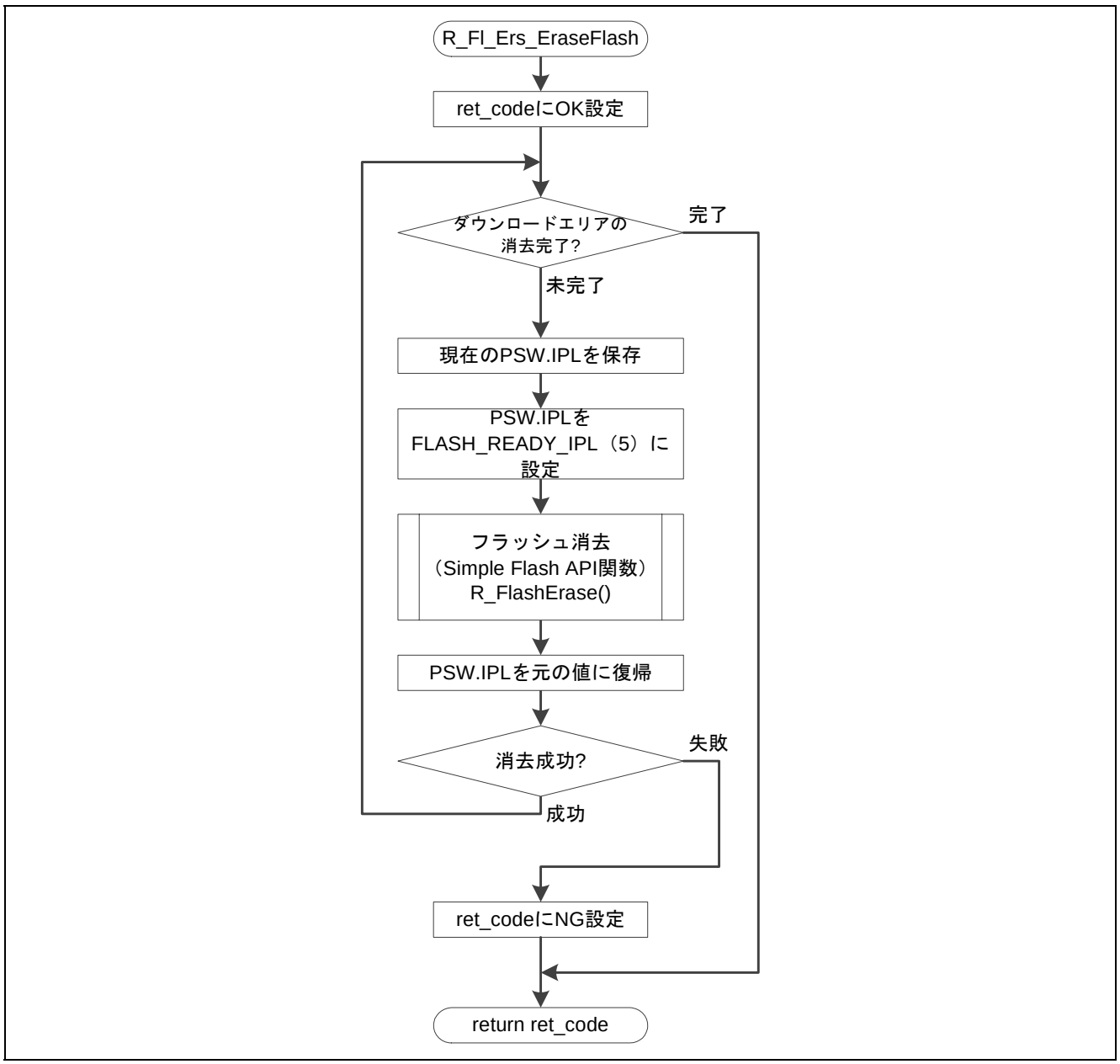


図5.20　ダウンロードエリア消去

### 5.13.7　書き込み処理

図 5.21に書き込み処理のフローチャートを示します。

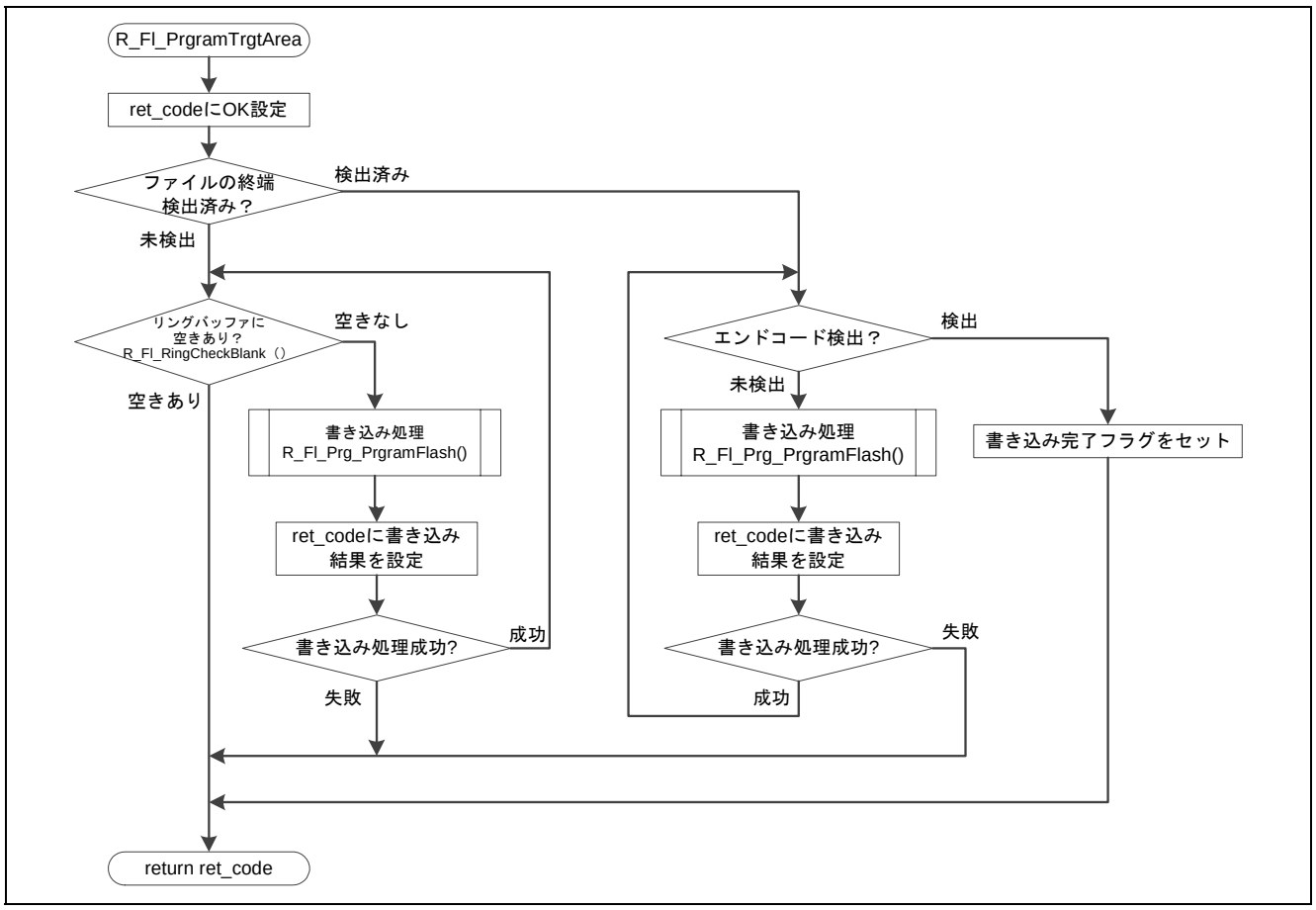


図5.21　書き込み処理

### 5.13.8　ダウンロードエリア書き込み

図 5.22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。

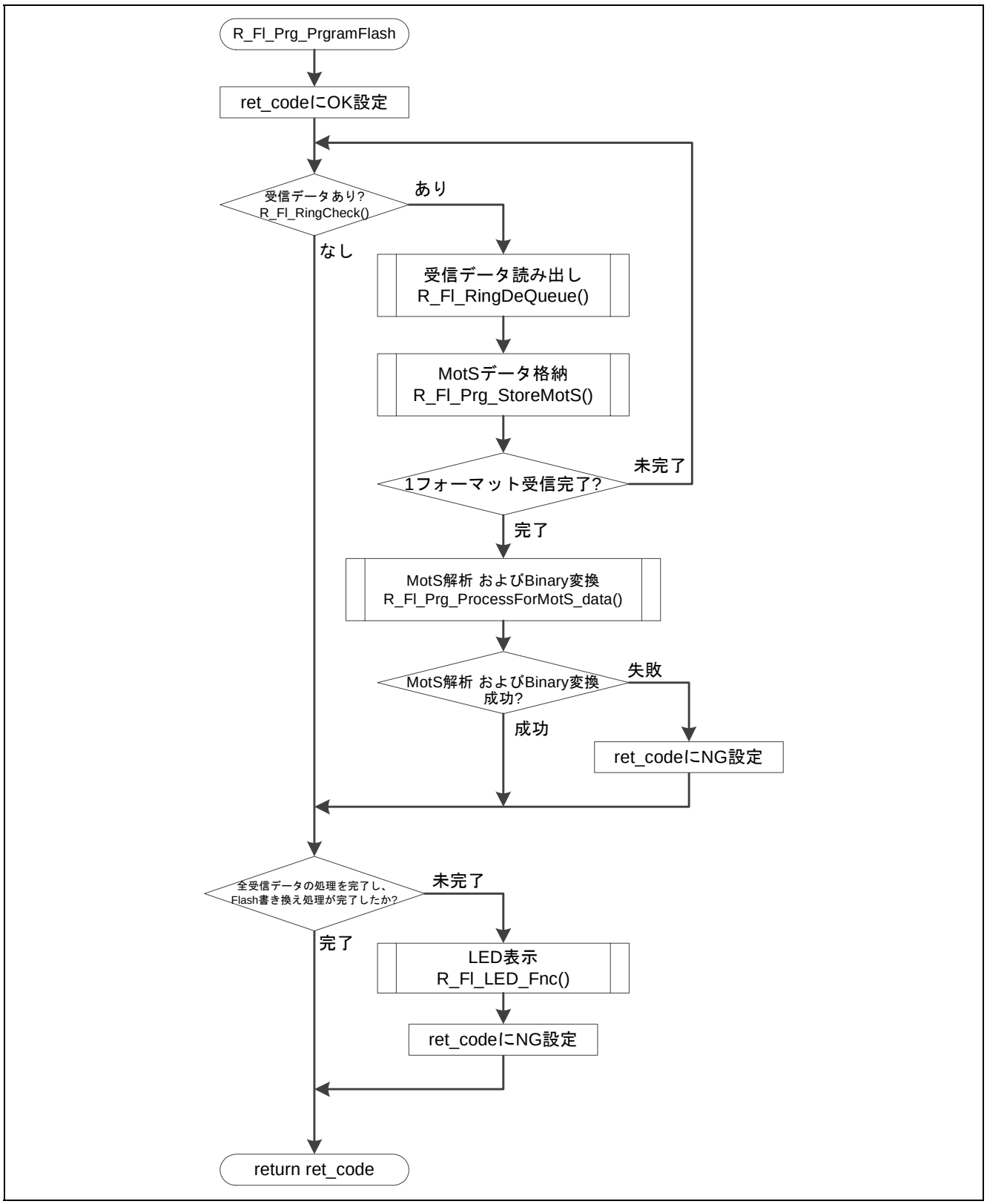


図5.22　ダウンロードエリア書き込み

### 5.13.9　S タイプフォーマットデータ格納

図 5.23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。

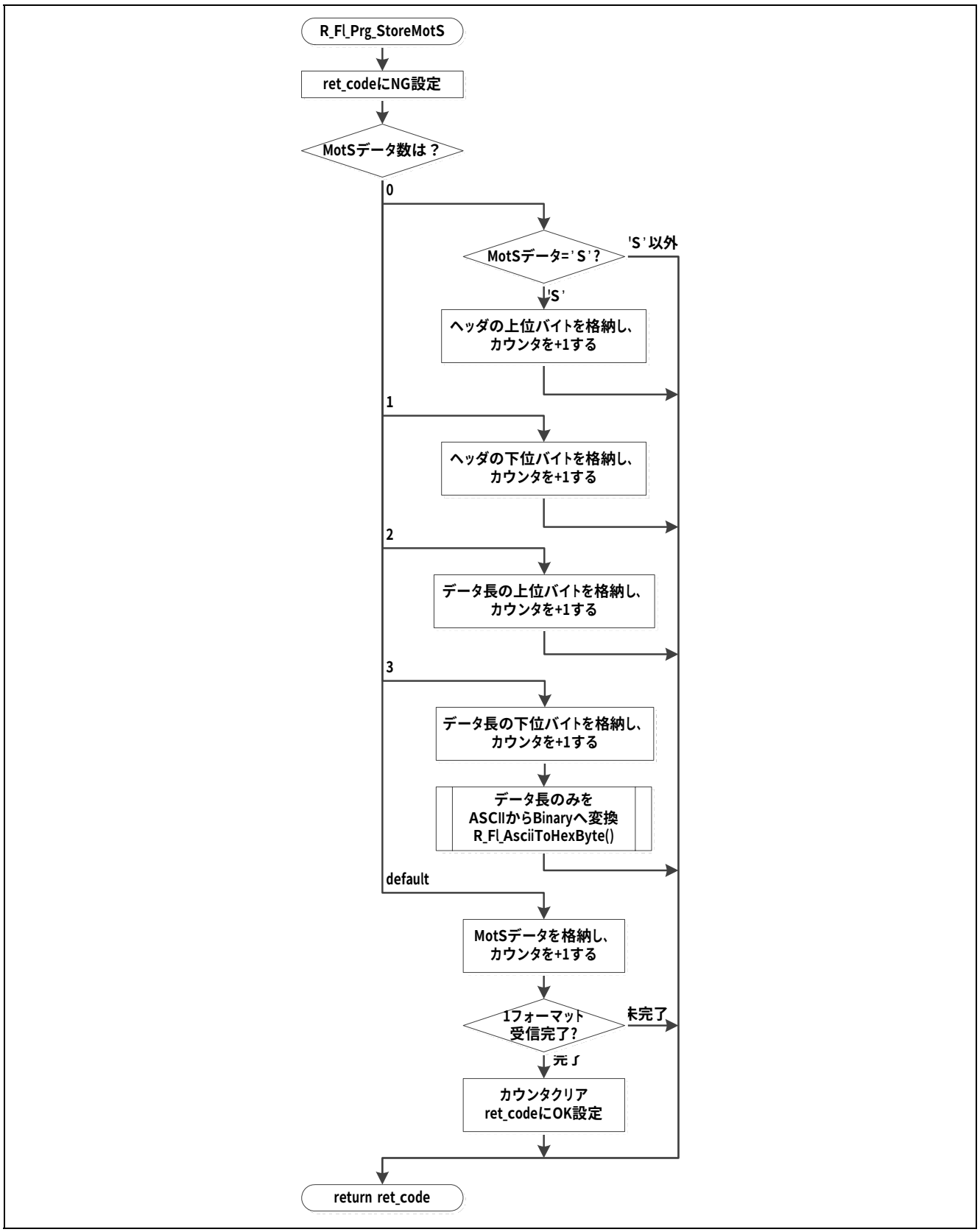


図5.23　S タイプフォーマットデータ格納

### 5.13.10　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 5.24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。

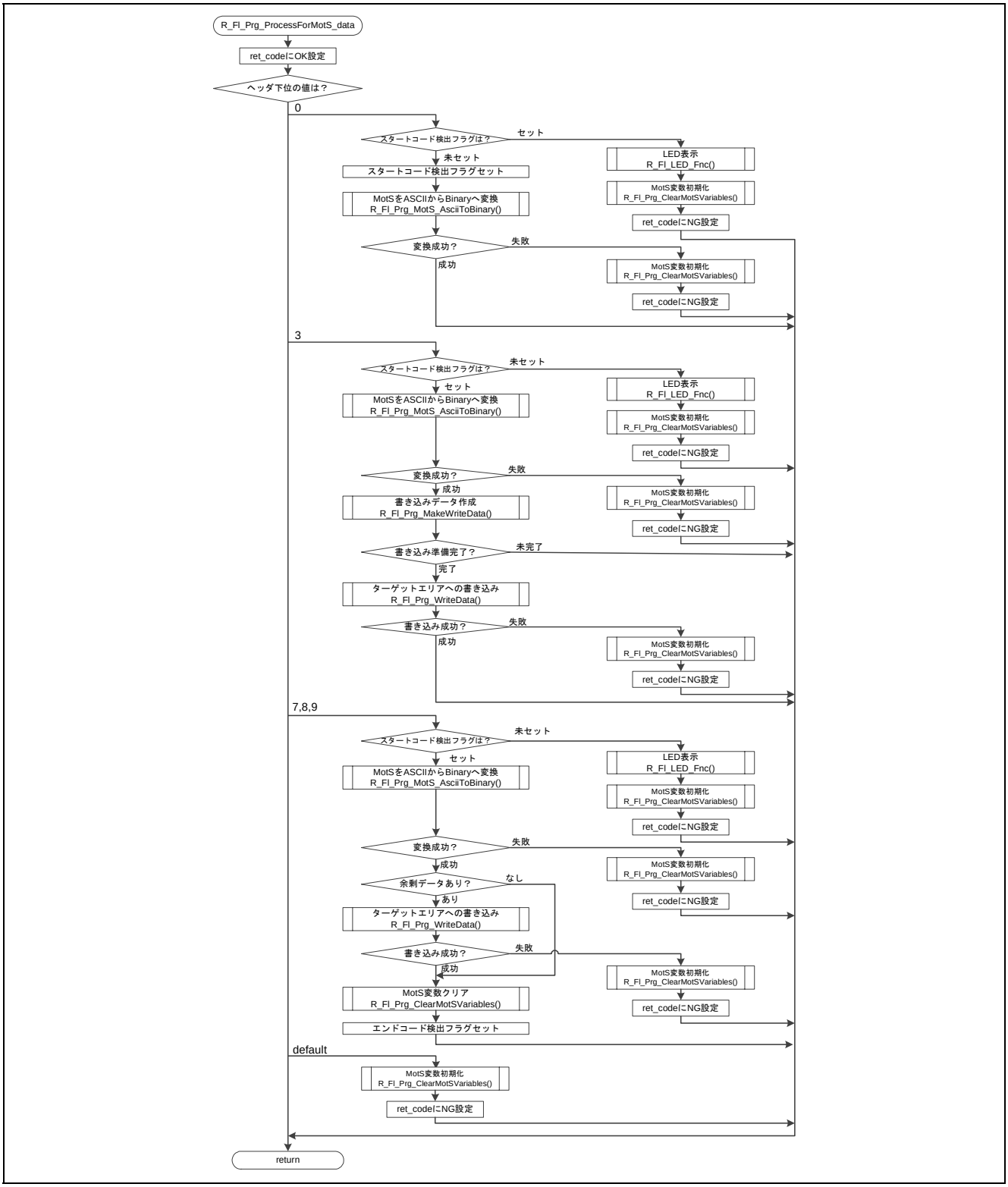


図5.24　S タイプフォーマットヘッダ解析、および Binary 変換、書き込み

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

図 5.25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。

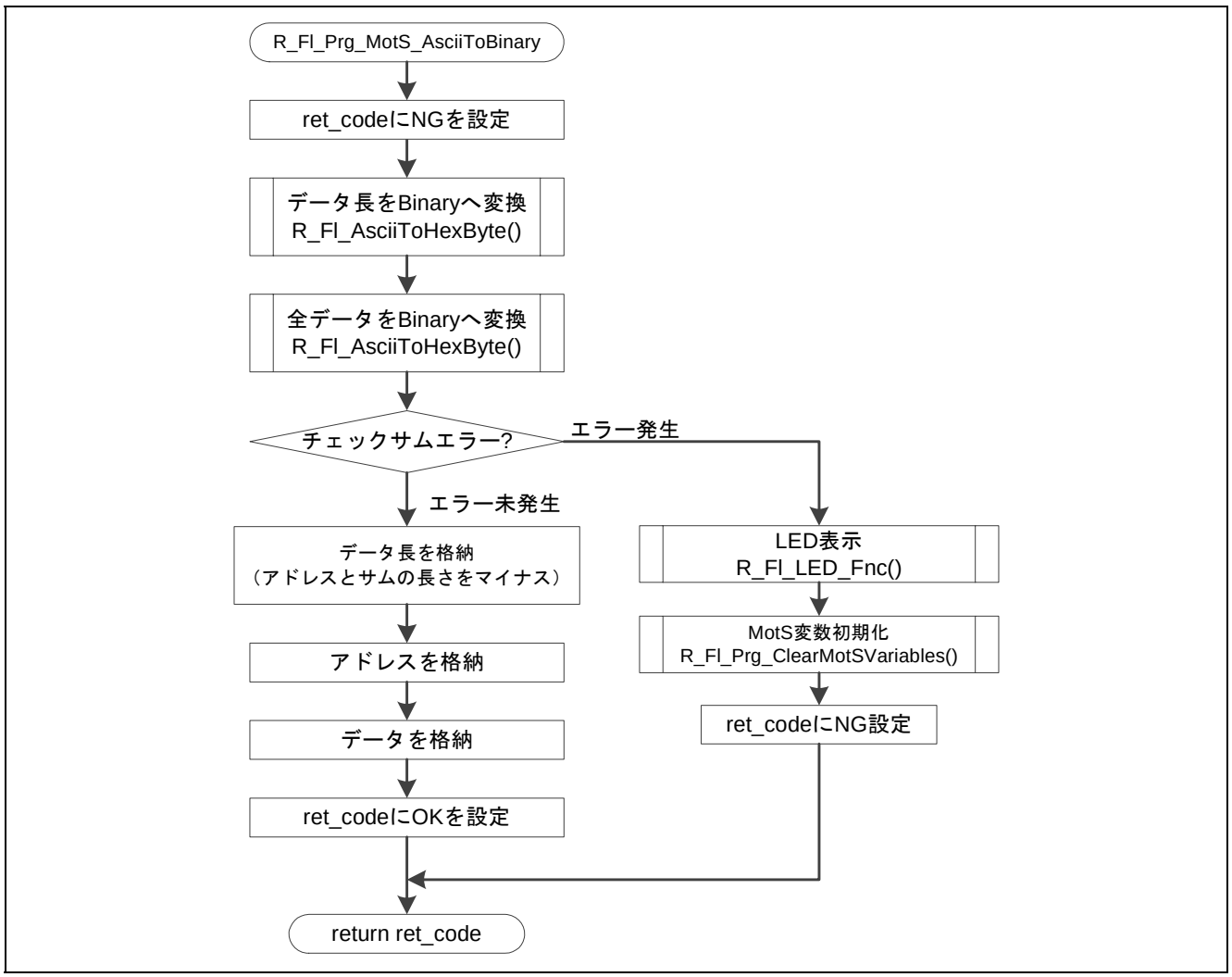


図5.25　S タイプフォーマットデータ ASCII - Binary 変換

### 5.13.12　ダウンロードエリアへの書き込みデータ作成

図 5.26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。

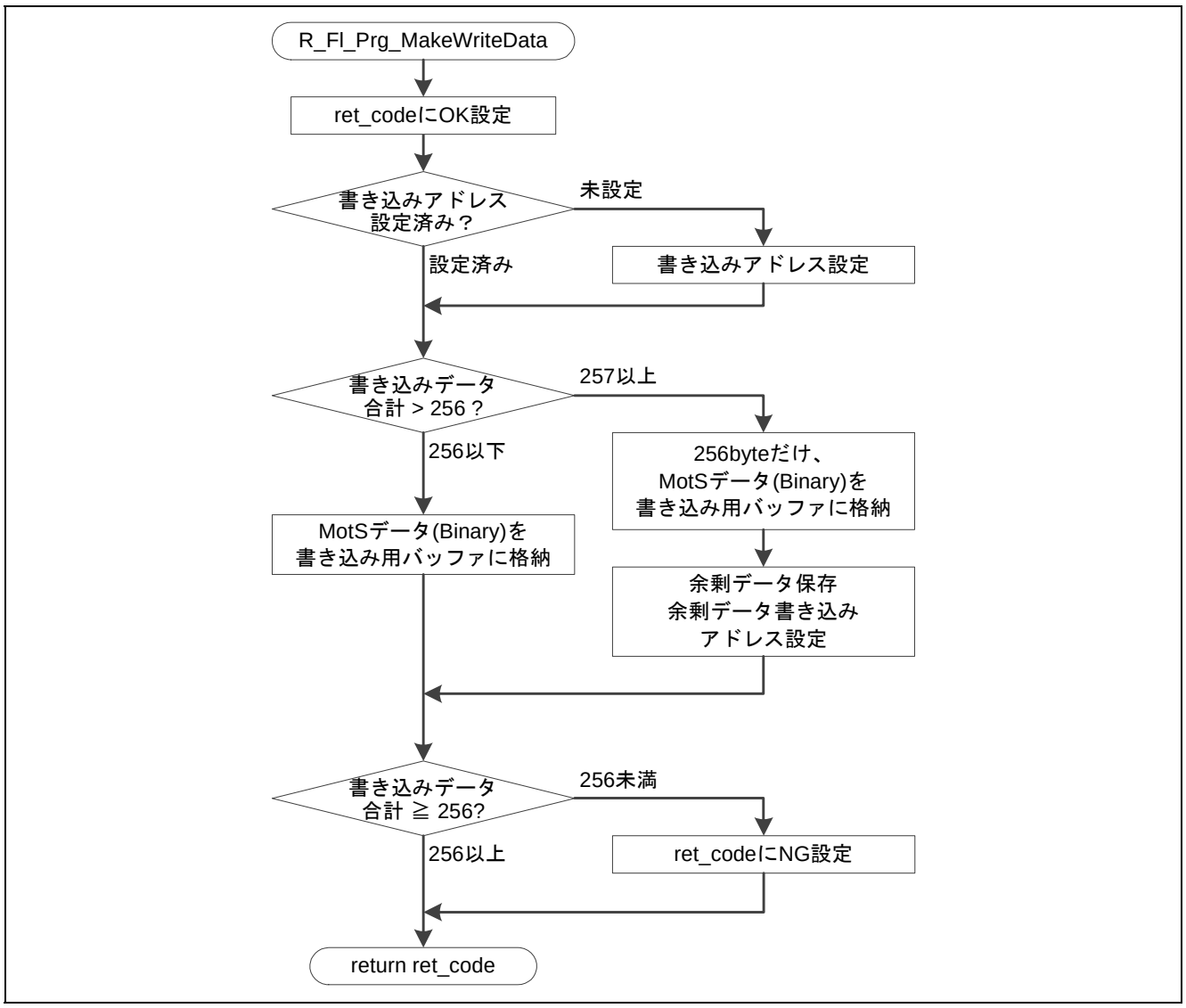


図5.26　ダウンロードエリアへの書き込みデータ作成

### 5.13.13　ダウンロードエリアへの書き込み

図 5.27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。

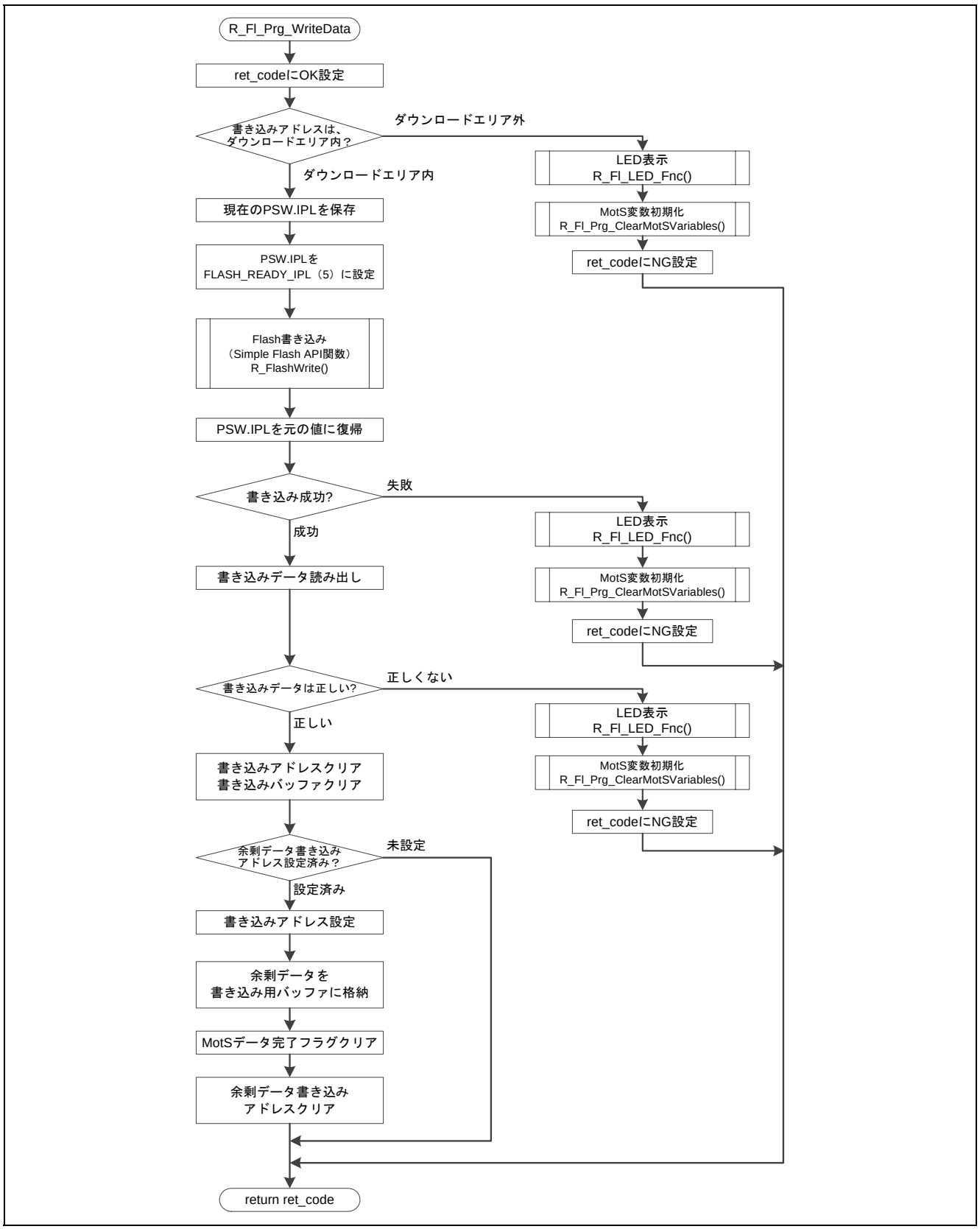


図5.27　ダウンロードエリアへの書き込み

### 5.13.14　S タイプフォーマットデータ関連の変数クリア

図 5.28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。

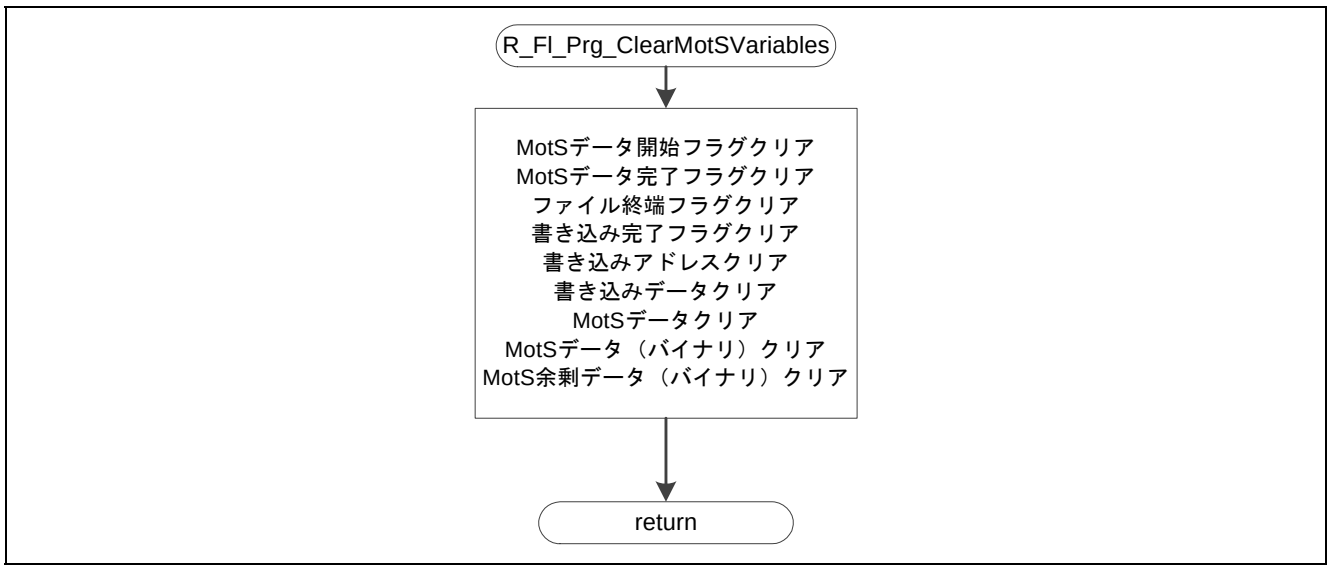


図5.28　S タイプフォーマットデータ関連の変数クリア

### 5.13.15　Ethernet 受信データ格納

図 5.29にEthernet 受信データ格納のフローチャートを示します。

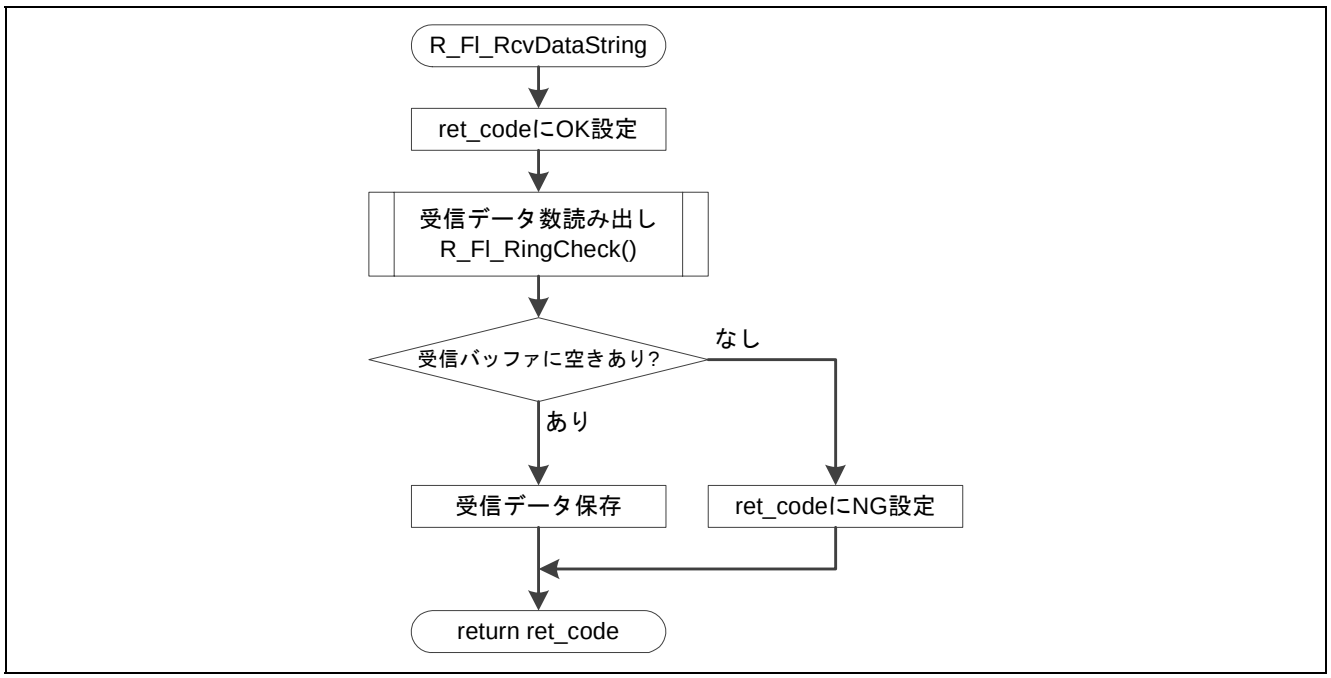


図5.29　Ethernet 受信データ格納

### 5.13.16　Ethernet 受信データ格納用リングバッファの空き容量確認

図 5.30にEthernet 受信データ格納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。

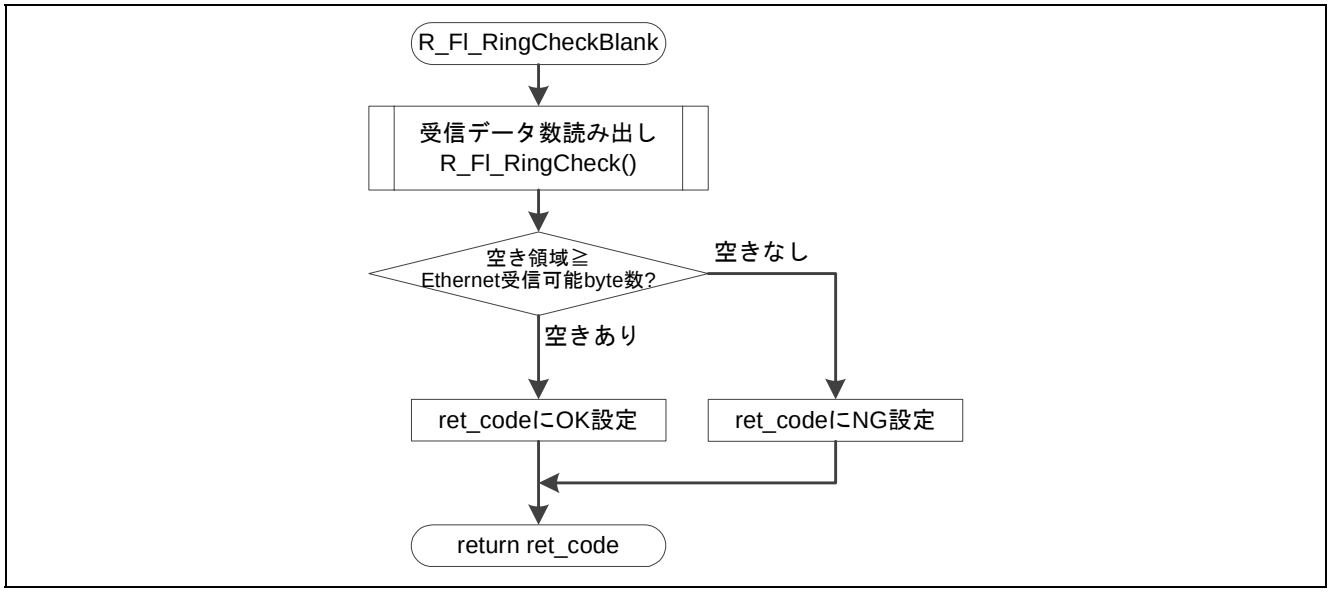


図5.30　Ethernet 受信データ格納用リングバッファの空き容量確認

### 5.13.17　Ethernet 受信データ格納用リングバッファの初期化

図 5.31にEthernet 受信データ格納用リングバッファの初期化のフローチャートを示します。

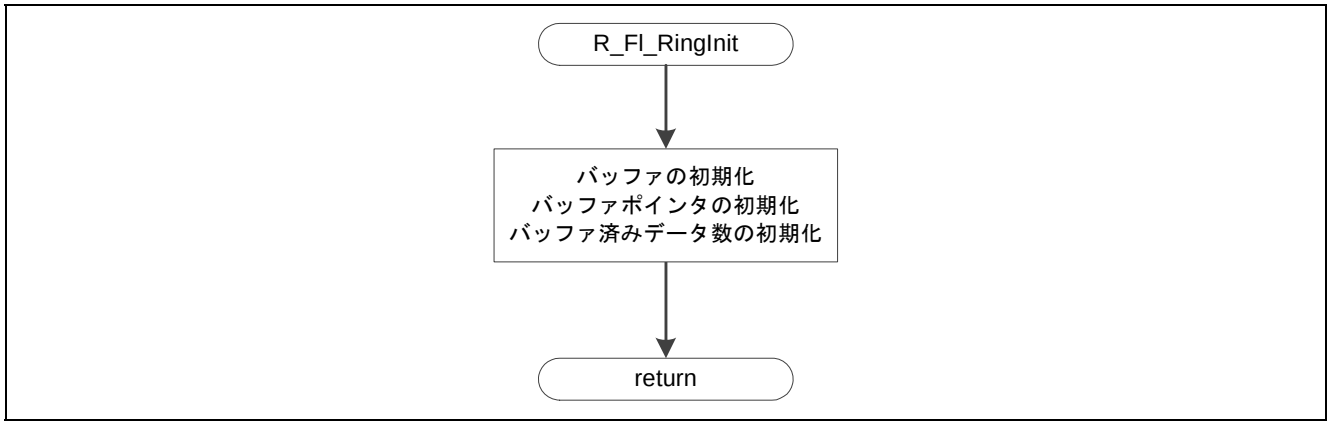


図5.31　Ethernet 受信データ格納用リングバッファの初期化

### 5.13.18　Ethernet 受信データ格納用リングバッファへのデータ格納

図 5.32にEthernet 受信データ格納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。

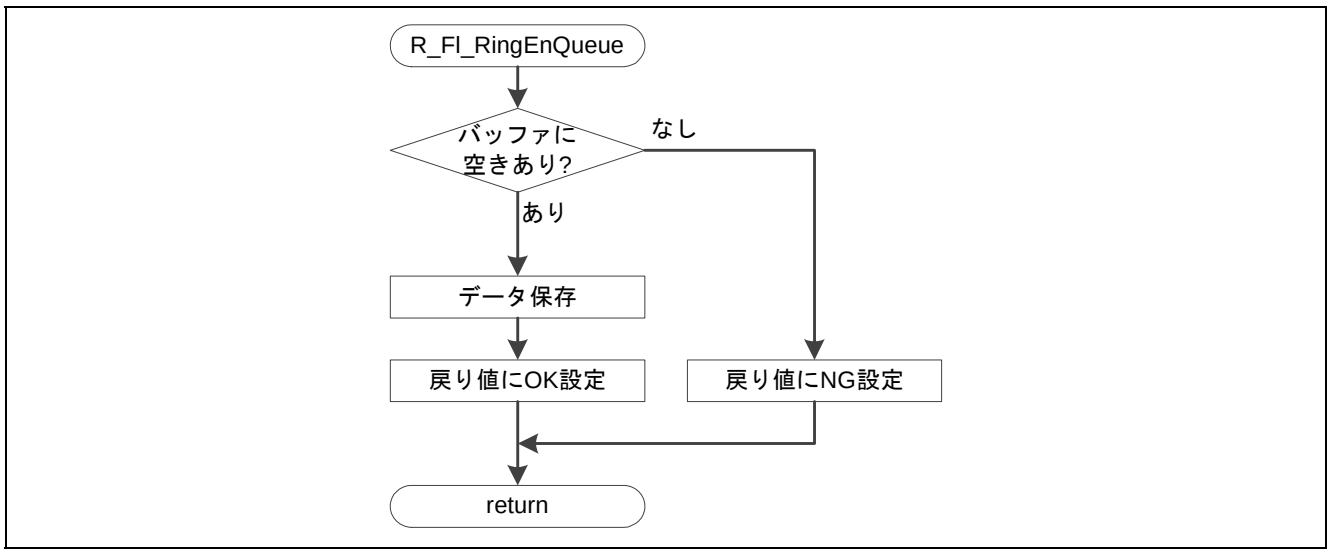


図5.32　Ethernet 受信データ格納用リングバッファへのデータ格納

### 5.13.19　Ethernet 受信データ格納用リングバッファからのデータ読み出し

図 5.33にEthernet 受信データ格納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。

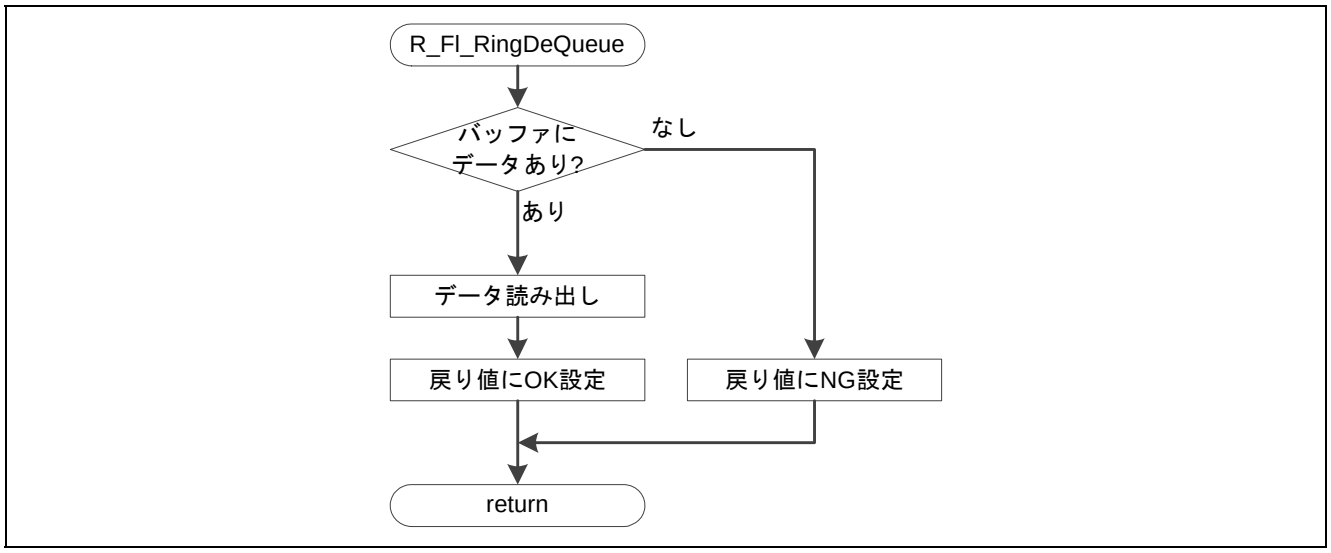


図5.33　Ethernet 受信データ格納用リングバッファからのデータ読み出し

### 5.13.20　Ethernet 受信データ格納用リングバッファのデータ数確認

図 5.34にEthernet 受信データ格納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。

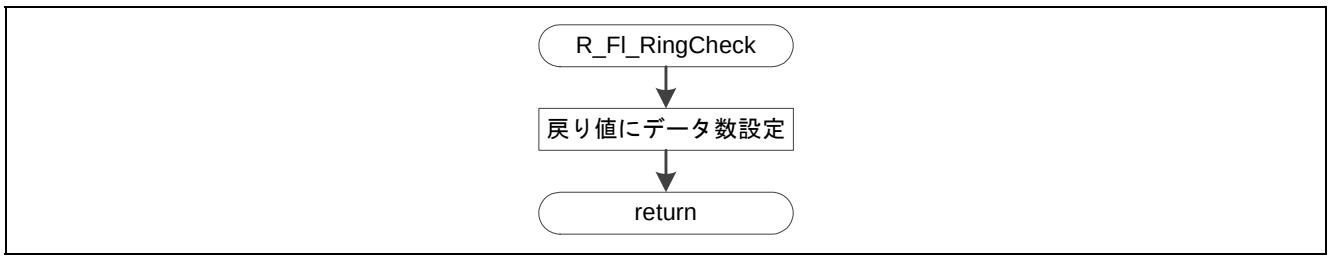


図5.34　Ethernet 受信データ格納用リングバッファのデータ数確認

### 5.13.21　ASCII コードから Binary データへの変換

図 5.35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。

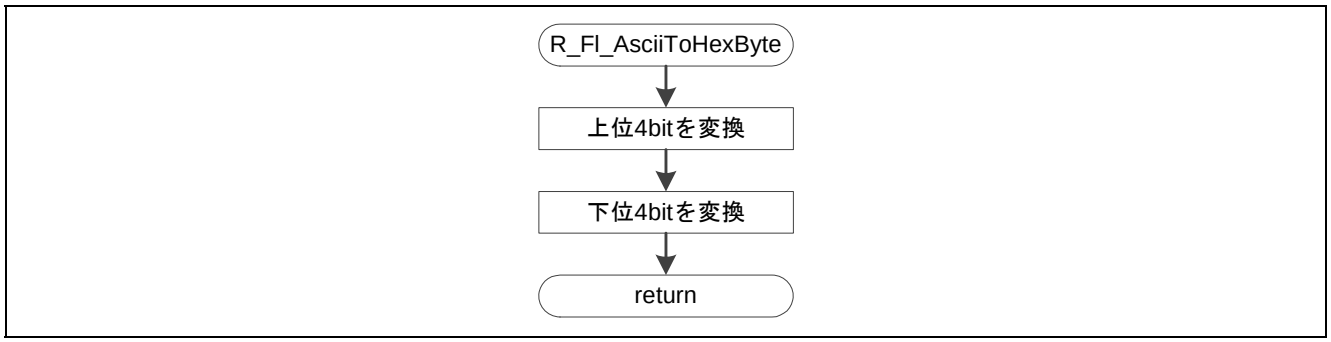


図5.35　ASCII コードから Binary データへの変換

### 5.13.22　LED 初期設定処理

図 5.36にLED 初期設定処理のフローチャートを示します。

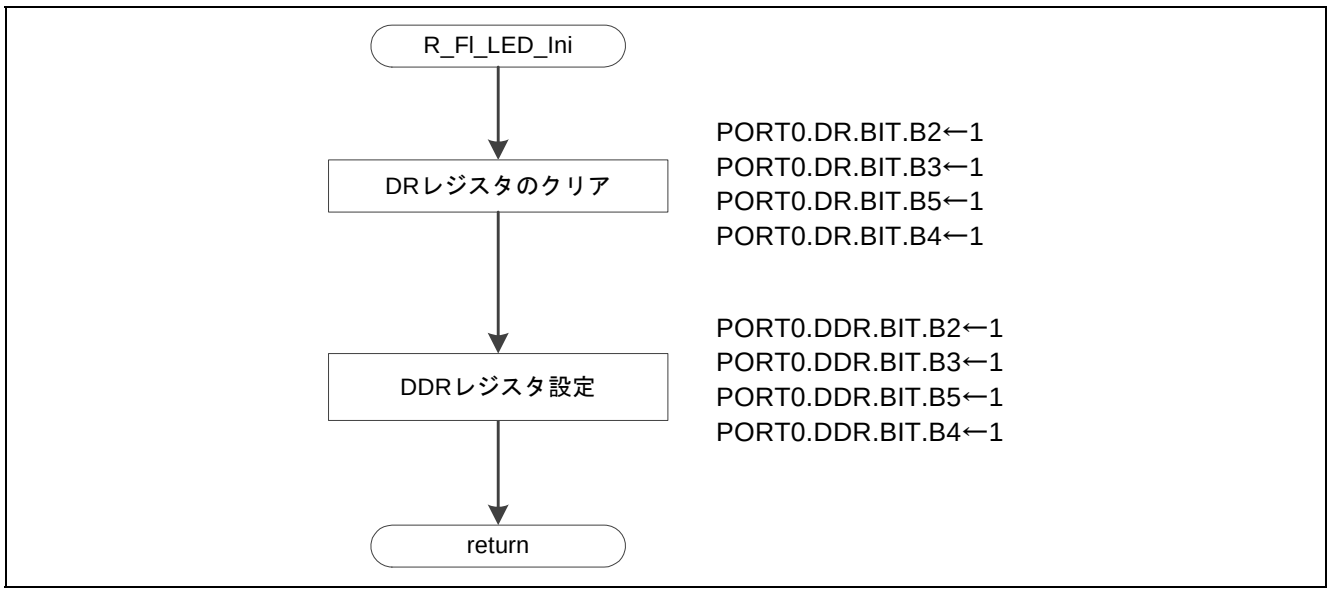


図5.36　LED 初期設定処理

### 5.13.23　LED 消灯／点灯処理

図 5.37にLED 消灯／点灯処理のフローチャートを示します。

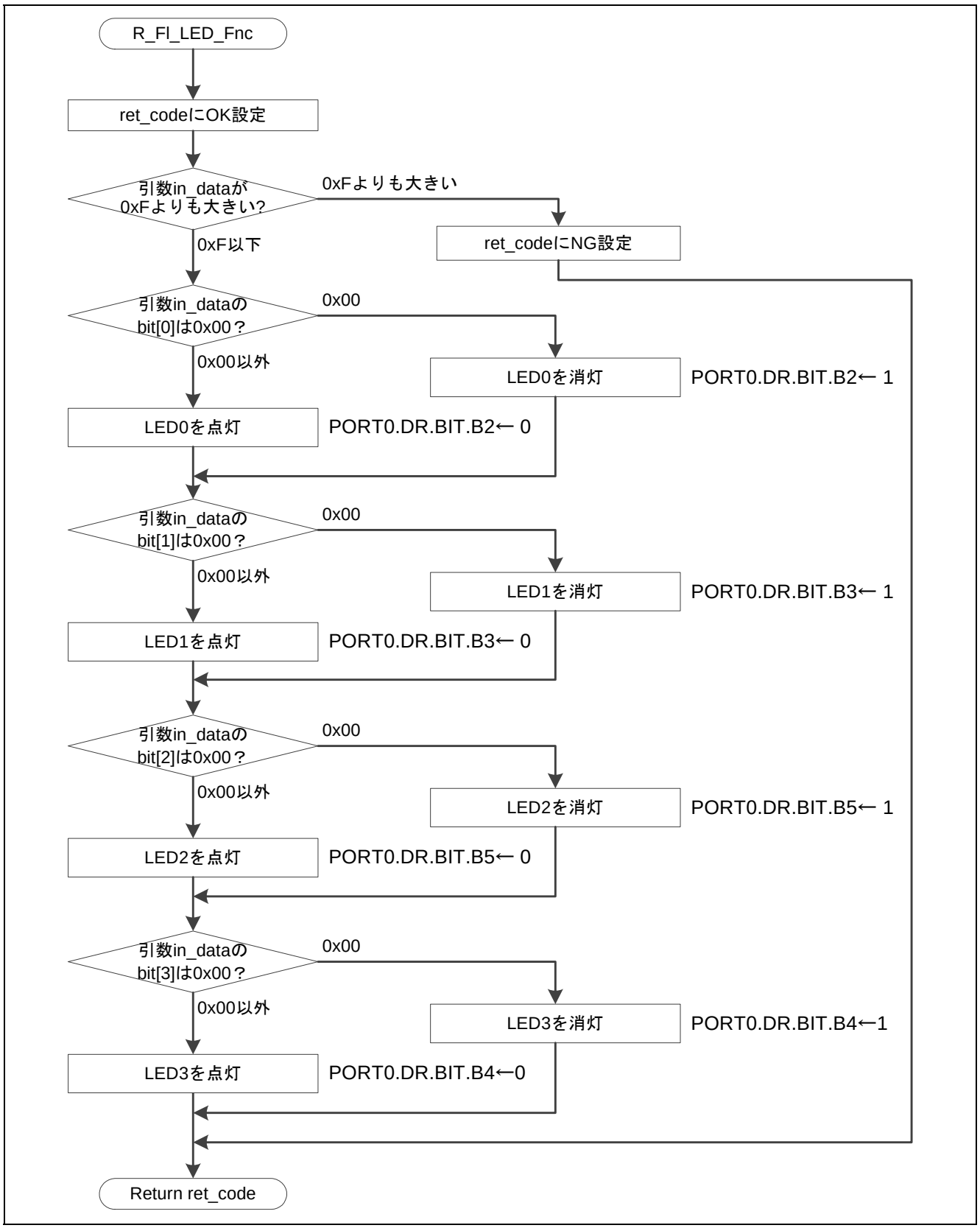


図5.37　LED 消灯／点灯処理

## 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）が付属しています。

「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.　ホスト PC 用サンプルプログラムの例

本アプリケーションノートには、ホスト PC 用サンプルプログラムのソースコードと実行ファイル（host_tool.zip）が付属しています。

ホスト PC 用サンプルプログラムは、TCP/IP モデルのアプリケーション層に相当するプログラムです。

本プログラムはコンソールアプリケーションで、RX62N-RSK の IP アドレスとポート番号、書き込む S タイプフォーマットのファイル名を指定して起動します。そして、指定した IP アドレスとポート番号に対して接続を試み、S タイプフォーマットファイルのデータを TCP で RX62N-RSK に送信します。なお、本プログラムでは、S タイプフォーマットファイルの最大サイズを 2,048Kbyte に設定しています。

本プログラムでは、TCP によるデータ転送を実現するために Winsock ライブラリを使用します。

# 8. S タイプフォーマット

サンプルコードが対応する S タイプフォーマットについて説明します。

## 8.1 レコード形式

図 8.1にサンプルコードが対応するレコード形式を示します。

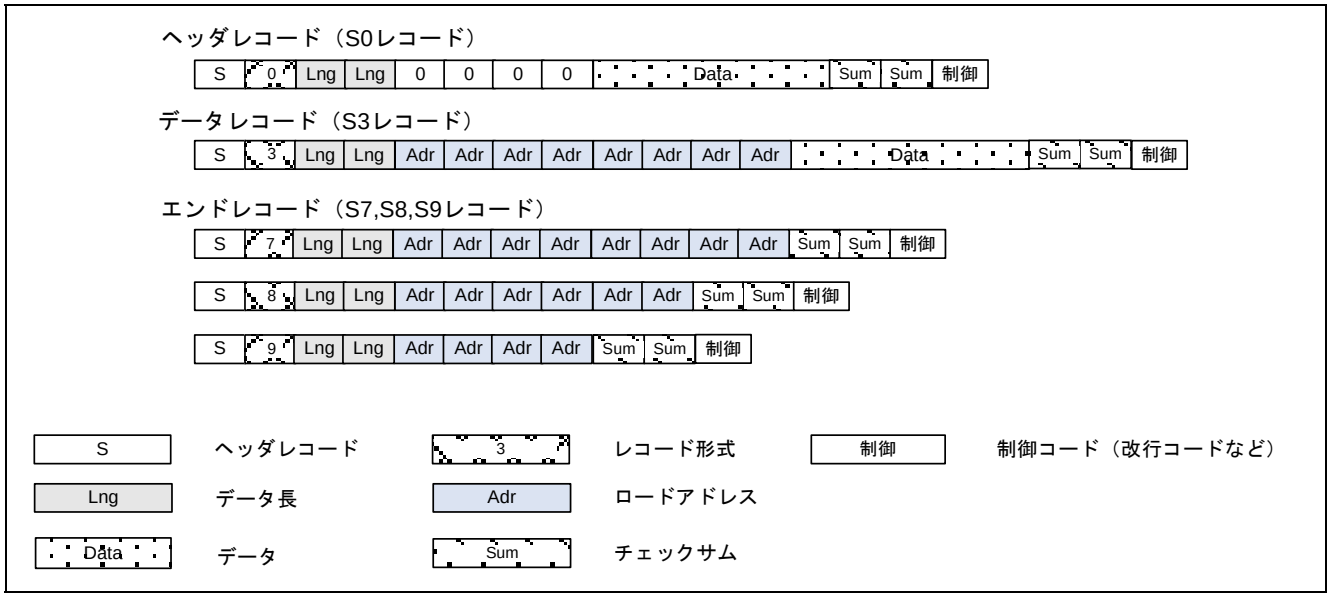


図8.1　サンプルコードが対応するレコード形式

## 8.2 レコード構成

図 8.2にサンプルコードが対応するレコード構成を示します。図 8.2の順序以外で構成された S タイプフォーマットには対応していません。

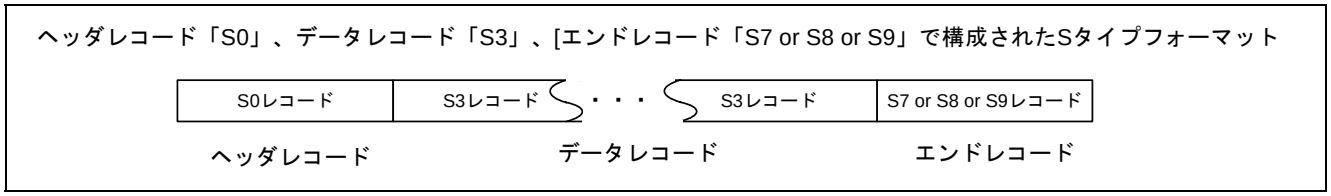


図8.2　サンプルコードが対応するレコード構成

## 8.3 ロードアドレス

サンプルコードでは、ロードアドレスが昇順の S タイプフォーマットのみ対応しています。降順およびアドレスが前後する S タイプフォーマットファイルは使用しないでください。

## 8.4 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

サンプルコードでは、受信した S タイプフォーマットのレコードごとにチェックサムの検査を行います。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

サンプルコードでは、受信した S タイプフォーマットが以下の条件のいずれかを満たした場合、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）またはエンドレコード（S7、S8、S9）を検出した場合。

図 8.3にフォーマットエラーの検出条件を示します。

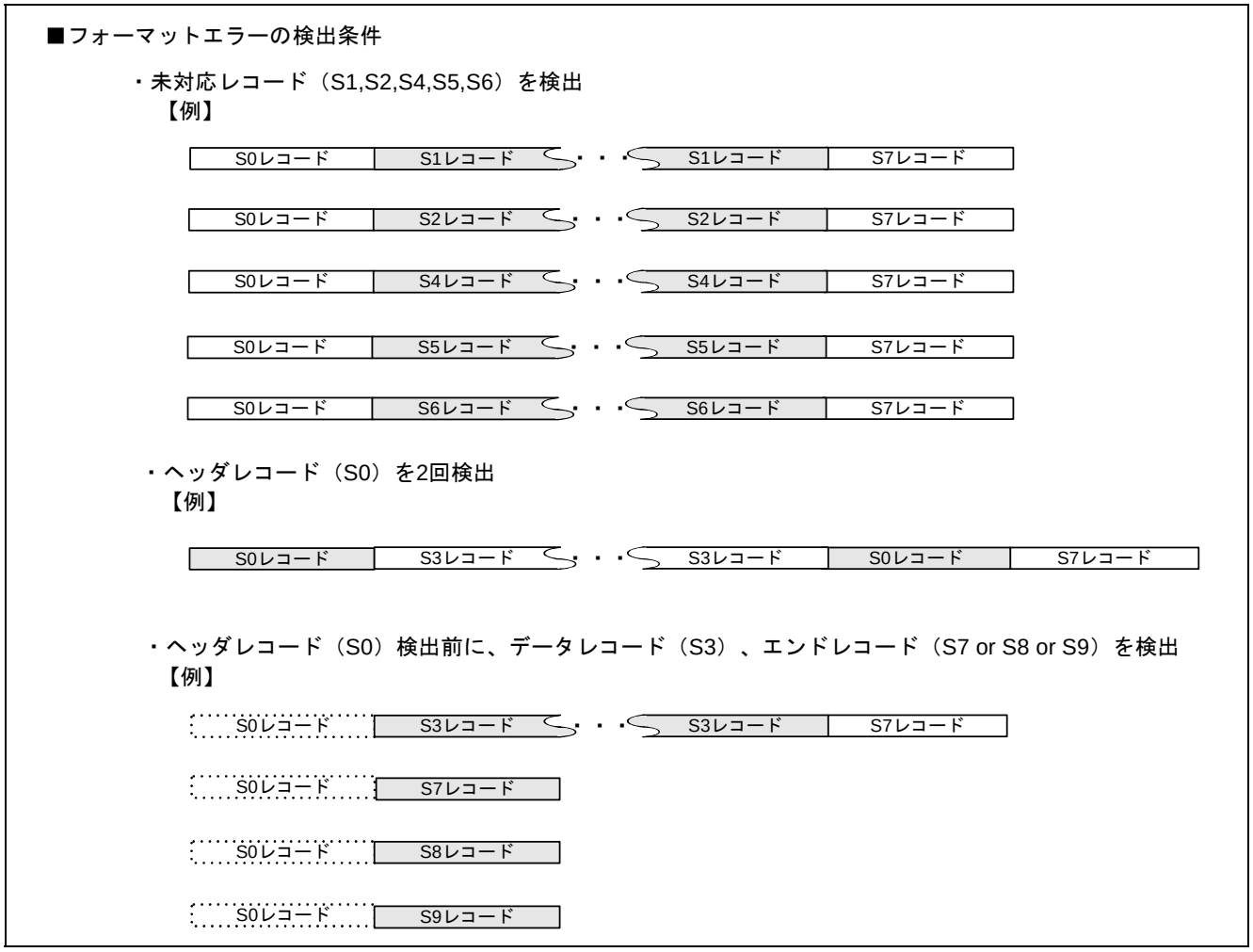


図8.3　フォーマットエラーの検出条件

(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。

# 9.　注意事項

## 9.1　消去／書き込み中の Ethernet ケーブル挿抜

ダウンロードエリアの消去／書き込み中は、Ethernet ケーブルを抜き差ししないでください。

## 9.2　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。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。

## 9.3　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（FFFF 9FFCh）の値で決定されます。したがって、ダウンロードコードは、リセットベクタが FFFF 9FFCh に配置されるように設定してください。詳細は「5.3.　動作概要」を参照してください。

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

## 9.4　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 512Kbyte です。

ROM 容量が 384Kbyte または 256Kbyte のマイコンを使用する場合は、ファイル"r_Flash_main.h"内の#define ディレクティブ"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 9.1にROM 容量一覧を示します。

表9.1　ROM 容量一覧

| 製品型名 | ROM 容量 | ダウンロードエリア<br>ROM 容量 | ダウンロードエリア<br>開始アドレス | ダウンロードエリア<br>ブロック番号 |
|---|---|---|---|---|
| R5F562x8 | 512K | 488K | FFF8 0000h | EB6～EB37 |
| R5F562x7 | 384K | 360K | FFFA 0000h | EB6～EB29 |
| R5F562x6 | 256K | 232K | FFFC 0000h | EB6～EB21 |

## 9.5　動作モード

本サンプルコードは、シングルチップモードのみに対応します。

RX62N-RSK の SW4 Pin1 および SW4 Pin2 は、OFF（MD0=High、MD1=High）に設定してください。

## 9.6　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。

### 9.6.1　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、次の設定を行ってください。また、RX62N-RSK の SW4 Pin3 を ON（MDE=Low）に設定してください。

1. HEW のツールバー[ビルド]→[RX Standard Toolchain]から RX Standard Toolchain を起動します。
2. タブ[CPU]のエンディアンから「Little-endian データ」を選択し、OK をクリックします。
3. HEW ワークスペースウィンドウのプロジェクトツリーにて、M3S-T4-Tiny ライブラリファイル "T4_Library_rx600_ether_big.lib"の右クリックメニュー「ビルドから除外」を選択し、同 "T4_Library_rx600_ether_little.lib"の右クリックメニュー「ビルドから除外の解除」を選択します。
4. 再度ビルドします。

### 9.6.2　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、次の設定を行ってください。また、RX62N-RSK の SW4 Pin3 を OFF（MDE=High）に設定してください。

1. HEW のツールバー[ビルド]→[RX Standard Toolchain]から RX Standard Toolchain を起動します。
2. タブ[CPU]のエンディアンから「Big-endian データ」を選択し、OK をクリックします。
3. HEW ワークスペースウィンドウのプロジェクトツリーにて、M3S-T4-Tiny ライブラリファイル "T4_Library_rx600_ether_little.lib"の右クリックメニュー「ビルドから除外」を選択し、同 "T4_Library_rx600_ether_big.lib"の右クリックメニュー「ビルドから除外の解除」を選択します。
4. 再度ビルドします。

## 9.7　ホスト PC 用サンプルプログラムの変更

本アプリケーションノートには、ホスト PC 用サンプルプログラムのソースファイルと実行ファイル（host_tool.zip）が付属しています。

ホスト PC 用サンプルプログラムを変更する場合は、お客様の責任において変更してください。

なお、弊社では統合開発環境として、

Microsoft Visual C++ 2010（Microsoft Visual Studio 2010 Professional）を

使用しています。

## 9.8　「RX600 シリーズ RX600 用のシンプルフラッシュ API」の変更点

本アプリケーションノートは、「RX600 シリーズ RX600 用のシンプルフラッシュ API」のサンプルコードを使用しています。「RX600 シリーズ RX600 用のシンプルフラッシュ API」の仕様については、「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。

### 9.8.1　使用したファイル

使用したファイルは、“ r_flash_api_rx600.c ”、“ r_flash_api_rx600.h ”、“ r_flash_api_rx600_private.h ”、“ r_flash_api_rx600_config.h ”および“ mcu_info.h ”です。

### 9.8.2　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：

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 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：

①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。

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 BCLK_HZ　　(12000000)

変更後：#define BCLK_HZ　　(24000000)

## 9.9　「RX ファミリ M3S-T4-Tiny: 導入ガイド」の変更点

本アプリケーションノートは、「RX ファミリ M3S-T4-Tiny: 導入ガイド」のサンプルコードおよびライブラリを使用しています。「RX ファミリ M3S-T4-Tiny」の仕様については、「RX ファミリ M3S-T4-Tiny: 導入ガイド」のアプリケーションノートを参照してください。

### 9.9.1　使用したファイル

使用したファイルは、"r_t4_itcpip.h"、"T4_Library_rx600_ether_big.lib"、"T4_Library_rx600_ether_little.lib"、"config_tcpudp.c"、"phy.c"、"phy.h"、"r_ether.c"、"r_ether.h"、"reg_accsess.h"、"t4_driver.c"、"timer.c"および"timer.h"です。

なお、T4 コンフィグレーションファイル"config_tcpudp.c"は、「RX ファミリ M3S-T4-Tiny: 導入ガイド」のプロジェクト[TCP ブロッキングコール]のファイルを使用しています。

### 9.9.2　変更箇所

使用したファイルのうち、変更したファイルはありません。

## 9.10　HEW 生成ファイルの変更点

本アプリケーションノートは、HEW 生成ファイルの一部を変更しています。

### 9.10.1　変更箇所

変更したファイルは、"dbsct.c"、"hwsetup.c"、"intprg.c"、"resetprg.c"、"stacksct.h"および"vect.h"です。

- ファイル"dbsct.c"内の変更箇所（2 箇所）：

(1) D セクションから R セクションへのコピー処理を行うセクションを追加

```
変更前：{ sectop("D_1"), secend("D_1"), sectop("R_1") }
変更後：{ sectop("D_1"), secend("D_1"), sectop("R_1") },
        { sectop("PFRAM"), secend("PFRAM"), sectop("RPFRAM") }
```

(2) B セクションの 0 クリア処理を行うセクションを追加

```
変更前：{ sectop("B_1"), secend("B_1") }
変更後：{ sectop("B_1"), secend("B_1") },
        { sectop("B_RX_DESC"), secend("B_RX_DESC") },
        { sectop("B_TX_DESC"), secend("B_TX_DESC") },
        { sectop("B_RX_BUFF_1"), secend("B_RX_BUFF_1") },
        { sectop("B_TX_BUFF_1"), secend("B_TX_BUFF_1") },
        { sectop("B_ETH_BUFF"), secend("B_ETH_BUFF") },
        { sectop("B_flash_api_sec"), secend("B_flash_api_sec") },
        { sectop("B_flash_api_sec_2"), secend("B_flash_api_sec_2") },
        { sectop("B_flash_api_sec_1"), secend("B_flash_api_sec_1") }
```

- ファイル"hwsetup.c"内の変更箇所（4 箇所）：

(1) インクルードするファイルの追加

```
変更前：なし
変更後：#include "r_ether.h"
```

(2) システムクロックの設定

```
変更前：なし
変更後：/* CPG setting */
        io_set_cpg();
```

(3) I/O ポートの設定

```
変更前：なし
変更後：/* Setup the port pins */
        ConfigurePortPins();
```

(4) 周辺モジュールのモジュールストップ状態の解除
変更前：なし
変更後：/* Enables peripherals */
EnablePeripheralModules();

- ファイル"intprg.c"内の変更箇所（1 箇所）：
(1) CMTU0_CMT0 の変更
変更前：void Excep_CMTU0_CMT0(void){ }
変更後：void Excep_CMTU0_CMT0(void){ timer_interrupt(); }

- ファイル"resetprg.c"内の変更箇所（1 箇所）：
(1) 動作選択処理関数を追加
変更前：なし
変更後：/* **** Mode entry **** */
R_Fl_Mode_Entry();

- ファイル"stacksct.h"内の変更箇所（1 箇所）：
(1) スタックサイズの変更
変更前：#pragma stacksize su=0x100
変更後：#pragma stacksize su=0x300

- ファイル"vect.h"内の変更箇所（1 箇所）：
(1) EHER EINT のコメント化
変更前：#pragma interrupt (Excep_ETHER_EINT(vect=32))
void Excep_ETHER_EINT(void);
変更後：//#pragma interrupt (Excep_ETHER_EINT(vect=32))
//void Excep_ETHER_EINT(void);

### 9.10.2　追加セクション

表 9.2に追加セクションを示します。

表9.2　追加セクション

| セクション名 | 概要 |
|---|---|
| RPFRAM | RAM 上で動作する Flash 書き換えコードの初期化データ領域（変数領域）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 |
| B_RX_DESC | Ethernet ドライバ（受信デスクリプタ）の未初期化データ領域用セクション |
| B_TX_DESC | Ethernet ドライバ（送信デスクリプタ）の未初期化データ領域用セクション |
| B_ETH_BUFF | Ethernet 受信バッファおよび M3S-T4-Tiny ワークメモリの未初期化データ領域用セクション |
| B_ETH_BUFF_1 | Ethernet 受信バッファおよび M3S-T4-Tiny ワークメモリの未初期化データ領域用セクション（Align = 1） |
| B_RX_DESC_1 | Ethernet ドライバ（受信デスクリプタ）の未初期化データ領域用セクション（Align = 1） |
| B_TX_DESC_1 | Ethernet ドライバ（送信デスクリプタ）の未初期化データ領域用セクション（Align = 1） |
| B_flash_api_sec | Flash 書き換えコードの未初期化データ領域用セクション |
| B_flash_api_sec_2 | Flash 書き換えコードの未初期化データ領域用セクション（Align = 2） |
| B_flash_api_sec_1 | Flash 書き換えコードの未初期化データ領域用セクション（Align = 1） |

### 9.10.3　インクルードファイルディレクトリ

インクルードファイルのディレクトリ設定を追加しています。

- "$(PROJDIR)¥src¥bsp"をインクルードファイルディレクトリに追加しています。
- "$(PROJDIR)¥src¥FlashAPI"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。
- "$(PROJDIR)¥src¥driver"をインクルードファイルディレクトリに追加しています。
- "$(PROJDIR)¥src¥t4¥lib"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。
- "$(PROJDIR)¥src¥user_app"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。

### 9.10.4　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。

## 10. 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。

## 11. 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 ファミリ用 C/C++コンパイラパッケージ　V.1.01 Release00
  RX ファミリ C/C++コンパイラパッケージ V1.01 ユーザーズマニュアル　Rev.1.00
  （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「RX ファミリ M3S-T4-Tiny: 導入ガイド」Rev.1.02 (R20AN0051JJ0102)
  （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

## 改訂記録

| Rev. | 発行日 | 改訂内容 | |
|---|---|---|---|
| | | ページ | ポイント |
| 1.00 | 2012.03.13 | — | 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、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、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きますとともに、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用途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。
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、海底中継機器、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て、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・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/




Colophon 1.1