RL78/G12

R01AN3022JJ0110
Rev. 1.10
2016.06.01

セルフ・プログラミング（UART 受信データ） CC-RL

要旨

本アプリケーションノートでは、セルフ書き込みによるフラッシュ・メモリ・プログラミングの使用方法の概要を説明します。フラッシュ・セルフ・プログラミング・ライブラリ Type01 を使用し、フラッシュ・メモリの書き換えを行います。

尚、本アプリケーションノートのサンプル・プログラムは、書き換え対象をコード・フラッシュの一部（アドレス 0x3BFC ～0x3BFF）に限定し、コード・フラッシュの一部をデータ格納領域として使用します。セルフ・プログラミングの実行方法、および、プログラム・フラッシュの全領域の書き換え方法の詳細については、「RL78/G13 マイクロコントローラ フラッシュ・セルフ・プログラミング実行編 アプリケーションノート（R01AN0718J）」を参照してください。

対象デバイス

RL78/G12

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では、セルフ書き込みによるフラッシュ・メモリ・プログラミングの使用方法を説明します。

コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値を読み出し、その値によって LED の点滅間隔を設定します。その後、送信側からデータ（4byte）を受信し、セルフ書き込みを行ってコード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値を受信データに書き換えます。書き換えが完了すると、再度コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値を読み出し、その値によって LED の点滅間隔を変更します。

表 1.1に使用する周辺機能と用途を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
シリアル・アレイ・ユニット 0 チャネル 0	UART でデータの受信を行う
シリアル・アレイ・ユニット 0 チャネル 1	UART でデータの送信を行う
ポート入出力	LED の点灯／消灯

1.1 フラッシュ・セルフ・プログラミング・ライブラリ概要

フラッシュ・セルフ・プログラミング・ライブラリは、RL78 マイクロコントローラに搭載されたファームウェアを使用し、コード・フラッシュ・メモリ内のデータを書き換えるためのソフトウエアです。

フラッシュ・セルフ・プログラミング・ライブラリをユーザ・プログラムから呼び出すことにより、コード・フラッシュ・メモリの内容を書き換えることができます。

フラッシュ・セルフ・プログラミングを行うためにはフラッシュ・セルフ・プログラミングの初期化処理や、使用する機能に対応する関数を C 言語、アセンブリ言語のどちらかでユーザ・プログラムから実行する必要があります。

1.2　コード・フラッシュ・メモリについて

RL78/G12(R5F1026A)のコード・フラッシュ・メモリの構成を以下に記載します。

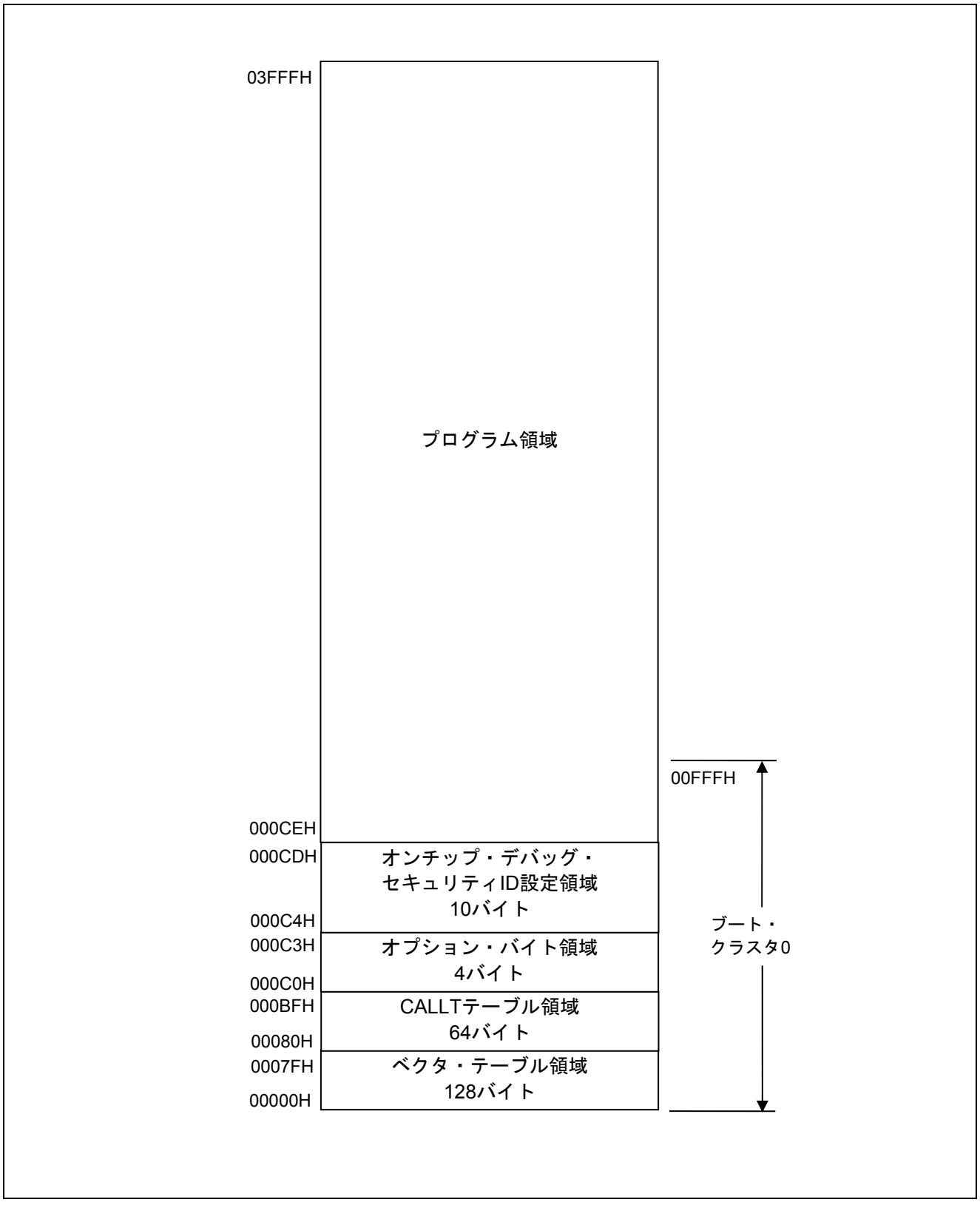

図 1.1　コード・フラッシュ・メモリの構成

RL78/G12 のコード・フラッシュ・メモリの特長を以下に記載します。

表 1.2 コード・フラッシュ・メモリの特長

項目	内容
消去、ベリファイの最小単位	1 ブロック（1024byte）
書き込みの最小単位	1 ワード（4byte）
セキュリティ機能	ブロック消去、書き込み、ブート領域の書き換え禁止設定が可能（出荷時は全て許可）
	フラッシュ・セルフ・プログラミング・ライブラリによりセキュリティ設定変更可能

注意 ブート領域の書き換え禁止以外のセキュリティ設定は、フラッシュ・セルフ・プログラミング時は無効となります。

1.3 フラッシュ・セルフ・プログラミング

RL78/G12 には、フラッシュ・セルフ・プログラミングを行うためのライブラリが用意されています。書き換えプログラムからフラッシュ・セルフ・プログラミング・ライブラリの各関数を呼び出すことでフラッシュ・セルフ・プログラミングを行います。

RL78/G12 のフラッシュ・セルフ・プログラミングはシーケンサ（フラッシュ・メモリ制御用の専用回路）を使用してフラッシュの書き換え制御を行いますが、シーケンサの制御中はコード・フラッシュ・メモリを参照できません。そのため、シーケンサ制御中にユーザ・プログラムを動作させる必要がある場合、コード・フラッシュ・メモリの消去や書き込み、セキュリティ・フラグの設定等を行う時に、フラッシュ・セルフ・プログラミング・ライブラリの一部のセグメントや、書き換えプログラムを RAM に配置して制御を行う必要があります。シーケンサ制御中にユーザ・プログラムを動作させる必要が無い場合は、フラッシュ・セルフ・プログラミング・ライブラリや書き換えプログラムを ROM（コード・フラッシュ・メモリ）上に配置して動作させることが可能です。

1.3.1　フラッシュ書き換え

RL78/G12 にはブート・スワップ機能がありません。フラッシュ・セルフ・プログラミングで、ベクタ・テーブル・データ、プログラムの基本機能、およびフラッシュ・セルフ・プログラミング・ライブラリを配置している領域の書き換え中に、電源の瞬断、外部要因によるリセットの発生などにより書き換えが失敗した場合、書き換え中のデータが破壊され、その後のリセットによるユーザ・プログラムの再スタートや再書き込みができなくなります。

フラッシュ・セルフ・プログラミングでのプログラムの書き換え動作イメージを以下に記載します。フラッシュ・セルフ・プログラミングを行うプログラムは、ブート・クラスタ 0 に配置しています。

本アプリケーションノートのサンプル・プログラムは、書き換え対象をコード・フラッシュの一部（アドレス 0x3BFC ～0x3BFF）に限定し、コード・フラッシュの一部をデータ格納領域として使用します。セルフ・プログラミングの実行方法、および、プログラム・フラッシュの全領域の書き換え方法の詳細については、「RL78/G13 マイクロコントローラ フラッシュ・セルフ・プログラミング実行編 アプリケーションノート（R01AN0718J）」を参照してください。

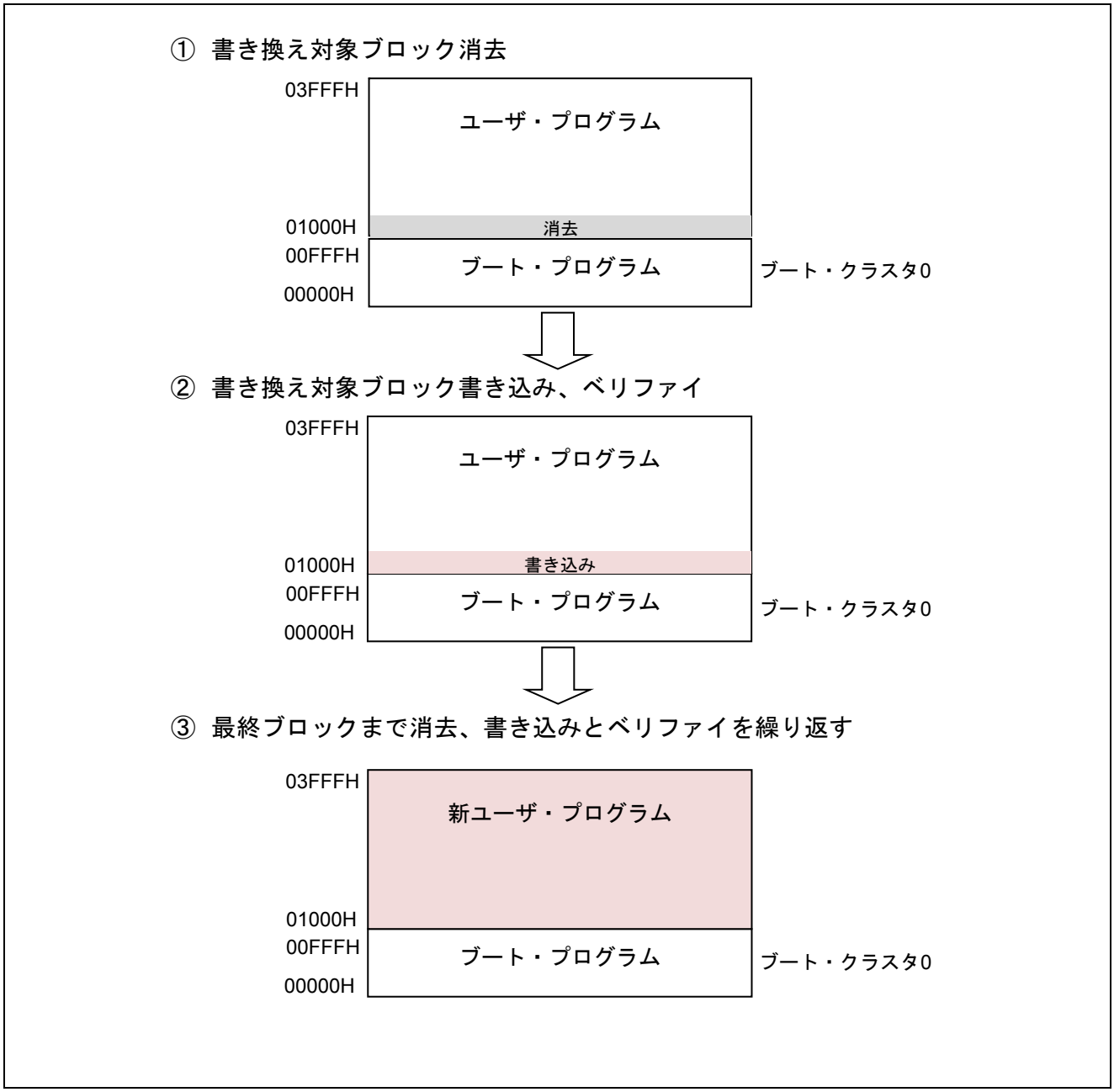

図 1.2　フラッシュ書き換えのイメージ図

1.4 フラッシュ・セルフ・プログラミング・ライブラリ取得方法

コンパイルを実行する前に、最新版のフラッシュ・セルフ・プログラミング・ライブラリをダウンロードして、本サンプルコードの r01an3022_flash フォルダ内の以下のフォルダにライブラリファイルをコピーしてください。

”incrl78”フォルダに”fsl.h”、”fsl.inc”、”fsl_types.h”をコピーする。

”librl78”フォルダに”fsl.lib”をコピーする。

フラッシュ・セルフ・プログラミング・ライブラリは、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。

詳細は、最寄りのルネサス営業または特約店にお問い合わせください。

2. 動作確認条件

本アプリケーションノード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	RL78/G12 (R5F1026A)
動作周波数	● 高速オンチップ・オシレータ・クロック ：24MHz ● CPU／周辺ハードウエア・クロック ：24MHz
動作電圧	5.0V（2.9V～5.5V で動作可能） LVD 動作（V_{LVD}）：リセット・モード 2.81V (2.76V～2.87V)
統合開発環境(CS+)	ルネサス エレクトロニクス製 CS+ for CC V3.03.00
C コンパイラ(CS+)	ルネサス エレクトロニクス製 CC-RL V1.02.00
統合開発環境(e^{2}studio)	ルネサス エレクトロニクス製 e^{2}studio V4.0.2.008
C コンパイラ(e^{2}studio)	ルネサス エレクトロニクス製 CC-RL V1.02.00
使用ボード	RL78/G12 ターゲット・ボード（QB-R5F1026A-TB）
CC-RL コンパイラ用フラッシュ・セルフ・プログラミング・ライブラリ(Type, Ver)	FSLRL78 Type01, Ver2.21 注

注 最新バージョンをご使用／評価の上、ご使用ください。

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。

RL78/G12 初期設定（R01AN2582J）アプリケーションノート

4. ハードウエア説明

4.1 ハードウエア構成例

図 4.1に本アプリケーションノートで使用する**ハードウエア構成例**を示します。

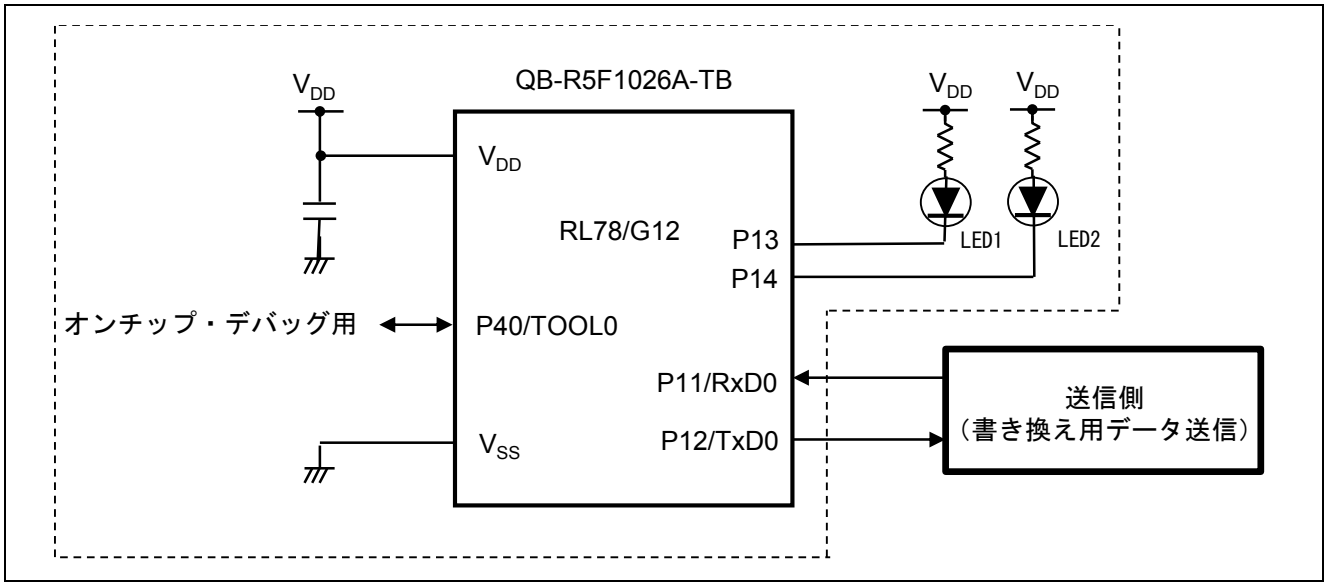

図 4.1 ハードウエア構成例

注意 1 この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい）。

2 V_{DD} は LVD にて設定したリセット解除電圧（V_{LVD}）以上にしてください。

4.2 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	機能
P11/ANI17/SI00/RxD0/SDA00/TOOLRxD	入力	UART シリアル・データ受信用端子
P12/ANI18/SO00/TxD0/TOOLTxD	出力	UART シリアル・データ送信用端子
P13	出力	LED1 の点灯／消灯
P14	出力	LED2 の点灯／消灯

5. ソフトウエア説明

5.1 通信仕様

本アプリケーションノートのサンプル・プログラムは、UART でデータを受信し、フラッシュ・セルフ・プログラミングを行います。送信側からは START コマンド、WRITE コマンド、END コマンドの 3 つのコマンドのいずれかが送信されます。それぞれのコマンドに応じた処理を行い、正常終了の場合には送信側へ正常応答（0x01）を返します。異常終了の場合には応答を返さず、LED1 と LED2 を点灯して、以降の処理は行いません。以下に UART 通信設定と、各コマンドの仕様を記載します。

表 5.1 UART 通信設定

データ・ビット長[bit]	8
データ転送方向	LSB ファースト
パリティ設定	パリティなし
転送レート[bps]	115200

5.1.1 START コマンド

START コマンドを受信するとフラッシュ・セルフ・プログラミングの初期設定を行います。正常終了すると送信側へ正常応答（0x01）を返します。異常終了の場合には応答を返さず、LED1 と LED2 を点灯して、以降の処理は行いません。

START コード (0x01)	データ長 (0x0002)	コマンド (0x02)	データ (なし)	チェックサム (1byte)

5.1.2 WRITE コマンド

WRITE コマンドを受信すると受信したデータをフラッシュ・メモリへ書き込みます。正常終了すると送信側へ正常応答（0x01）を返します。異常終了の場合には応答を返さず、LED1 と LED2 を点灯して、以降の処理は行いません。

START コード (0x01)	データ長 (0x0006)	コマンド (0x03)	データ (4byte)	チェックサム (1byte)

5.1.3 END コマンド

END コマンドを受信すると現在書き込んでいるブロックのベリファイを行います。ベリファイが異常終了の場合には LED1 と LED2 を点灯して、以降の処理は行いません。END コマンドを受信した場合には、正常終了／異常終了に関わらず、送信側への応答は返しません。

START コード (0x01)	データ長 (0x0002)	コマンド (0x04)	データ (なし)	チェックサム (1byte)

※チェックサムは、コマンド部とデータ部のバイト単位の加算値です。

5.1.4 通信シーケンス

本サンプル・プログラムは送信側からのコマンド受信により、以下に示すシーケンスで動作を行います。

(1) 送信側：

START コマンドを送信します。

(2) 本サンプル・プログラム：

LED2 を点灯して「フラッシュ・アクセス中」であることを示し、フラッシュ・セルフ・プログラミングの初期設定を行います、正常終了すると応答（0x01）を返します。

(3) 送信側：

WRITE コマンドとデータ（4byte）を送信します。

(4) 本サンプル・プログラム：

受信したデータ（4byte）をプログラム・フラッシュのアドレス 0x3BFC ～ 0x3BFF に書き込み、正常終了すると応答（0x01）を返します。

(5) 送信側：

END コマンドを送信します。

(6) 本サンプル・プログラム：

現在の書き換え対象ブロックのベリファイを行い、LED2 を消灯して「フラッシュ・アクセス中」でないことを示します。

5.2 動作概要

本アプリケーションノートでは、セルフ書き込みによるフラッシュ・メモリ・プログラミングの使用方法を説明します。

コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値を読み出し、その値によって LED1 の点滅間隔を設定します。その後、送信側からデータ（4byte）を受信し、セルフ書き込みを行ってコード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値を受信データに書き換えます。書き換えが完了すると、再度コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値を読み出し、その値によって LED1 の点滅間隔を変更します。

LED1 は、送信側から受信したデータ(4byte)の平均値（コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF のバイト毎の加算値を 4 で割った値）× 10[ms]の間隔で点滅します。例えば、アドレス 0x3BFC の値が”15”、アドレス 0x3BFD の値が”150”、アドレス 0x3BFE の値が”100”、アドレス 0x3BFF の値が”200”だった場合は、(15 + 150 + 100 + 200) / 4 * 10 = 1162.5 となり、LED1 は 1162.5[ms]の間隔で点滅します。

LED2 の点灯は、「フラッシュ・アクセス中」であることを示します。

(1) 入出力ポートを設定します。

<設定条件>

- LED 点灯制御ポート（LED1、LED2）：P13、P14 を出力ポートに設定します。

(2) SAU0 のチャネル 0、1 の初期設定を行います。

<設定条件>

- SAU0 チャネル 0、1 を UART として使用します。
- データ出力は P12/TxD0 端子、データ入力は P11/RxD0 端子を使用します。
- データ長は 8 ビットを使用します。
- データ転送方向設定は LSB ファーストを使用します。
- パリティ設定はパリティビットなしを使用します。
- 受信データ・レベル設定は標準を使用します。
- 転送レートは 115200bps を使用します。

(3) TAU0 のチャネル 0 の初期設定を行います。

<設定条件>

- TAU0 の動作クロック 0（CK00）を 23.44[KHz]、動作クロック 1（CK01）を 24[MHz]、動作クロック 2（CK02）を 12[MHz]、動作クロック 3（CK03）を 93.75[KHz]に設定します。
- 動作クロックは動作クロック 0（CK00）を選択します。
- スタート・トリガはソフトウエア・トリガ・スタートのみ有効に設定します。
- 動作モードはインターバル・タイマ・モード、カウント開始時にタイマ割り込みを発生しない設定にします。

(4) UART0 の動作を開始し、送信完了割り込みを禁止します。

(5) 割り込みを許可します。

(6)　コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値を読み出し、0x3BFC ～ 0x3BFF の値の平均を計算して LED1 を点灯します。

(7)　読み出し値が 0 より大きかった場合には、LED1 を点滅させるために TAU0 のチャネル 0 のインターバル・タイムを 0x3BFC ～ 0x3BFF の平均値 × 10[ms]に設定し、TAU0 のチャネル 0 の動作を開始します。

(8)　HALT モードに移行して、送信側からの送信データを待ちます。

UART 受信完了割り込み要求か、TAU0 のチャネル 0 の割り込み要求で HALT モードから通常動作に移行します。TAU0 のチャネル 0 の割り込み要求で HALT モードから復帰した場合には、再度 HALT モードに移行します。

(9)　割り込みを禁止します。

(10)　読み出し値が 0 より大きかった場合は、TAU0 のチャネル 0 の動作を停止します。

(11)　送信側から START コマンド（0x02）を受信したら、セルフ・プログラミングの初期設定を行います。

- P14 を Low レベル出力にし、LED2 を点灯して「フラッシュ・アクセス中」であることを示します。
- FSL_Init 関数を呼び出し、フラッシュ・セルフ・プログラミング環境の初期化を行い、以下のように設定します。

  電圧モード　　：フルスピードモード

  CPU の動作周波数　　：24[MHz]

  ステータス・チェック・モード　：ステータス・チェック・インターナル・モード
- FSL_Open 関数を呼び出し、フラッシュ・セルフ・プログラミングを開始（フラッシュ環境の開始）します。
- FSL_PrepareFunctions 関数を呼び出し、RAM 実行が必要なフラッシュ関数（標準書き換え関数）を使用できる状態にします。

(12)　送信側へ正常終了（0x01）を送信します。

(13)　WRITE コマンド（0x03）と書き込みデータ（4byte）を受信します。

(14)　書き込み先アドレスから、書き換え対象ブロックを算出します。

(15)　FSL_BlankCheck 関数を呼び出し、書き換え対象ブロックが書込み済みかどうかを確認します。

(16)　書き換え対象ブロックが書き込み済みの場合は、FSL_Erase 関数を呼び出し、書き換え対象ブロックを消去します。

(17) FSL_Write 関数を呼び出し、書き込み先アドレスに受信データを書き込みます。

(18) 送信側へ正常終了（0x01）を送信します。

(19) END コマンド（0x04）を受信します。

(20) FSL_IVerify 関数を呼び出し、書き換え対象ブロックをベリファイします。

(21) P14 を High レベル出力にし、LED2 を消灯して「フラッシュ・アクセス中」でないことを示します。

(22) (5)へ戻ります。

注意　フラッシュ・セルフ・プログラミングを正常終了することができなかった場合（処理中にエラーが発生した場合）は、LED1 と LED2 を点灯し、以降の処理は行いません。

5.3　ファイル構成

表 5.2に統合開発環境で自動生成されるファイルへの追加関数、追加ファイル一覧を示します。

表 5.2　追加関数、追加ファイル一覧

ファイル名	概要	備考
r_main.c	メインモジュール	追加関数： r_main_led_blink r_main _clear_uart_flag r_main _packet_analyze r_main _self_execute r_main _self_initialize r_main _write_execute
r_cg_serial_user.c	SAU モジュール	追加関数： r_uart0_receive_start r_uart0_send_start

5.4 オプション・バイトの設定一覧

表 5.3にオプション・バイト設定一覧を示します。

表 5.3 オプション・バイト設定一覧

アドレス	設定値	内容
000C0H/010C0H	11101111B	ウォッチドッグ・タイマ 動作停止 (リセット解除後、カウント停止)
000C1H/010C1H	01111111B	LVD リセット・モード 2.81V (2.76V～2.87V)
000C2H/010C2H	11100000B	HS モード、HOCO クロック : 24MHz
000C3H/010C3H	10000100B	オンチップ・デバッグ許可 オンチップ・デバッグ・セキュリティ ID 認証失敗時にフラッシュ・メモリのデータを消去する

RL78/G12 のオプション・バイトは、ユーザ・オプション・バイト（000C0H – 000C2H）とオンチップ・デバッグ・オプション・バイト（000C3H）で構成されています。

電源投入時、またはリセット解除後、自動的にオプション・バイトを参照して、指定された機能の設定が行われます。

5.5 リンク・オプション

リンク・オプションの-start オプションでフラッシュ・セルフ・プログラミングを行うフラッシュ・セルフ・ライブラリを ROM 領域に配置します。

フラッシュ・セルフ・プログラミング・ライブラリ Type01 で設定が必要となるセクションは、”-start オプション”で全て指定してください。

注 リンク・オプションの詳細については、RL78 コンパイラ CC-RL ユーザーズマニュアル (R20UT3123J)を参照して下さい。

5.6 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。

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
LED1	P1_bit.no3	LED1 制御ポート
LED2	P1_bit.no4	LED2 制御ポート
LED_ON	0	LED 点灯
LED_OFF	1	LED 消灯
NORMAL_END	0x00	正常終了
ERROR	0xFF	異常終了
NO_RECIEVE	0x00	コマンド受信状態：未受信
START_CODE	0x01	コマンド受信状態：START コード受信済
PACKET_SIZE	0x02	コマンド受信状態：データ長受信済
START	0x02	START コマンド
WRITE	0x03	WRITE コマンド
END	0x04	END コマンド
FULL_SPEED_MODE	0x00	フラッシュ・セルフ・ライブラリ初期化関数の引数：動作モードをフルスピードモードに設定
FREQUENCY_24M	0x18	フラッシュ・セルフ・ライブラリ初期化関数の引数：RL78/G12 の動作周波数 = 24MHz
INTERNAL_MODE	0x01	フラッシュ・セルフ・ライブラリ初期関数の引数：ステータス・チェック・インターナル・モードに設定
BLOCK_SIZE	0x400	コード・フラッシュの 1 ブロックのサイズ（1024byte）
TXSIZE	0x01	送信側へ送信する応答データのサイズ
RXSIZE	0x06	受信バッファのサイズ
WRITESIZE	0x01	書き込みデータサイズ（ワード）
WRITEADDR	0x3BFC	書き込み開始アドレス
READADDR	0x3BFC	読み出し開始アドレス

5.7 変数一覧

表 5.5に本サンプル・プログラムで使用するグローバル変数を示します。

表 5.5 グローバル変数

Type	Variable Name	Contents	Function Used
uint8_t	g_intsr_flag	UART 受信完了割り込み発生フラグ	main r_main_clear_uart_flag r_uart0_interrupt_receive
uint8_t	g_intsre_flag	UART 受信エラー割り込み発生フラグ	main r_main_clear_uart_flag r_uart0_interrupt_error

5.8 関数一覧

表 5.6に関数一覧を示します。

表 5.6 関数一覧

関数名	概要
R_UART0_Start	UART0 動作開始
r_uart0_interrupt_receive	UART0 受信完了割り込み
r_uart0_interrupt_error	UART0 受信エラー割り込み
r_main_led_blink	読み出し値に応じて LED1 の点滅間隔を変更し、LED1 の点滅開始
R_TAU0_Channel0_Start	TAU0 チャネル 0 の動作開始
r_tau0_channel0_interrupt	TAU0 チャネル 0 割り込み
r_uart0_receive_start	UART0 データ受信
r_main_clear_uart_flag	UART0 受信完了割り込み発生フラグ、UART0 受信エラー割り込み発生フラグのクリア
R_TAU0_Channel0_Stop	TAU0 チャネル 0 の動作停止
r_main_packet_analyze	受信データ解析
r_ main_self_execute	フラッシュ・セルフ・プログラミング実行
r_ main_self_initialize	フラッシュ・セルフ・プログラミング初期設定
r_ main_write_execute	フラッシュ書き換え実行
r_uart0_send_start	UART0 データ送信

5.9 関数仕様

サンプルコードの関数仕様を示します。

[関数名] R_ UART0_Start

概要	UART0 動作開始
ヘッダ	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h
宣言	void R_ UART0_Start(void)
説明	シリアル・アレイ・ユニット 0 のチャネル 0 の動作を開始し、通信待機状態にします。
引数	なし
リターン値	なし
備考	なし

[関数名] r_uart_interrupt_receive

概要	UART0 受信完了割り込み
ヘッダ	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h
宣言	__interrupt void r_uart_interrupt_receive(void)
説明	UART0 受信完了割り込み発生フラグ g_intsr_flag に”1”をセットします。
引数	なし
リターン値	なし
備考	なし

[関数名] r_uart_interrupt_error

概要	UART0 受信エラー割り込み
ヘッダ	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h
宣言	__interrupt void r_uart_interrupt_error(void)
説明	UART0 受信エラー発生フラグ g_intsre_flag に”1”をセットします。
引数	なし
リターン値	なし
備考	なし

[関数名] r_main_led_blink

概要	読み出し値に応じて LED1 の点滅間隔を変更し、LED1 の点滅開始	
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h	
宣言	void r_main_led_blink(float average)	
説明	LED1 の点滅間隔を引数 average × 10[ms]に設定し、LED1 の点滅動作を開始します。	
引数	average	コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値の平均
リターン値	なし	
備考	なし	

[関数名] R_TAU0_Channel0_Start

概要	TAU0 チャネル 0 の動作開始	
ヘッダ	r_cg_macrodriver.h r_cg_timer.h r_cg_userdefine.h	
宣言	void R_TAU0_Channel0_Start(void)	
説明	TAU0 チャネル 0 の動作を開始します。	
引数	なし	
リターン値	なし	
備考	なし	

[関数名] r_tau0_channel0_interrupt

概要	TAU0 チャネル 0 割り込み	
ヘッダ	r_cg_macrodriver.h r_cg_timer.h r_cg_userdefine.h	
宣言	__interrupt void r_tau0_channel0_interrupt(void)	
説明	LED1 の状態（点灯／消灯）を反転します。	
引数	なし	
リターン値	なし	
備考	なし	

[関数名] r_uart0_receive_start

概要	UART0 データ受信	
ヘッダ	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h	
宣言	uint8_t r_uart0_receive_start(uint16_t *rx_length, uint8_t *rx_buf)	
説明	受信データを受信バッファ rx_buf に格納し、受信データ長[byte]を rx_length に格納します。	
引数	rx_length	受信データ長[byte]格納アドレス
	rx_buf	受信データ・バッファのアドレス
リターン値	正常終了：NORMAL_END 異常終了：ERROR	
備考	なし	

[関数名] r_main_clear_uart_flag

概要	UART0 受信完了割り込み発生フラグ、UART0 受信エラー割り込み発生フラグのクリア
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
宣言	void r_main_clear_uart_flag(void)
説明	UART0 受信完了割り込み発生フラグ g_intsr_flag と、UART0 受信エラー割り込み発生フラグ g_intsre_flag を”0”にクリアします。
引数	なし
リターン値	なし
備考	なし

[関数名] R_TAU0_Channel0_Stop

概要	TAU0 チャネル 0 の動作停止
ヘッダ	r_cg_macrodriver.h r_cg_timer.h r_cg_userdefine.h
宣言	void R_TAU0_Channel0_Stop(void)
説明	TAU0 チャネル 0 の動作を停止します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_main_packet_analyze

概要	受信データ解析	
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h	
宣言	uint8_t r_main_packet_analyze(uint16_t rx_length, uint8_t *rx_buf)	
説明	受信したコマンドのパラメータチェック、チェックサムの計算、比較を行い、受信したデータが正しいかどうかを判定します。	
引数	rx_length	受信データ長[byte]
	rx_buf	受信データ・バッファのアドレス
リターン値	START コマンド受信：START WRITE コマンド受信：WRITE END コマンド受信：END コマンドのパラメータ・エラー、チェックサムエラー：ERROR	
備考	なし	

[関数名] r_main_self_execute

概要	フラッシュ・セルフ・プログラミング実行
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_userdefine.h fsl.h fsl_types.h
宣言	void r_main_self_execute(void)
説明	フラッシュ・セルフ・プログラミングを行います。
引数	なし
リターン値	なし
備考	なし

[関数名] r_main_self_initialize

概要	フラッシュ・セルフ・プログラミング初期設定
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_userdefine.h fsl.h fsl_types.h
宣言	uint8_t r_main_self_initialize(void)
説明	フラッシュ・セルフ・プログラミング実行前の初期設定を行います。
引数	なし
リターン値	正常終了：FSL_OK パラメータ・エラー：FSL_ERR_PARAMETER 消去エラー：FSL_ERR_ERASE 内部ベリファイ・エラー：FSL_ERR_IVERIFY 書き込みエラー：FSL_ERR_WRITE フローエラー：FSL_ERR_FLOW
備考	なし

[関数名] r_main_write_execute

概要	フラッシュ書き換え実行	
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_userdefine.h fsl.h fsl_types.h	
宣言	uint8_t r_main_write_execute(uint32_t write_addr)	
説明	フラッシュ書き換えを行います。	
引数	Write_addr	書き換え開始アドレス
リターン値	正常終了：FSL_OK 異常終了：ERROR	
備考	なし	

[関数名] r_uart0_send_start

概要	UART0 データ送信	
ヘッダ	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h	
宣言	uint8_t r_uart0_send_start(uint16_t tx_length, uint8_t *tx_buf)	
説明	tx_buf に格納されているデータを tx_length[byte]送信します。	
引数	tx_length tx_buf	送信データ長[byte] 送信データ・バッファのアドレス
リターン値	正常終了：NORMAL_END パラメータ・エラー(tx_length が 0 以下)：ERROR	
備考	なし	

5.10 フローチャート

図 5.1にサンプルコードの**全体フロー**を示します。

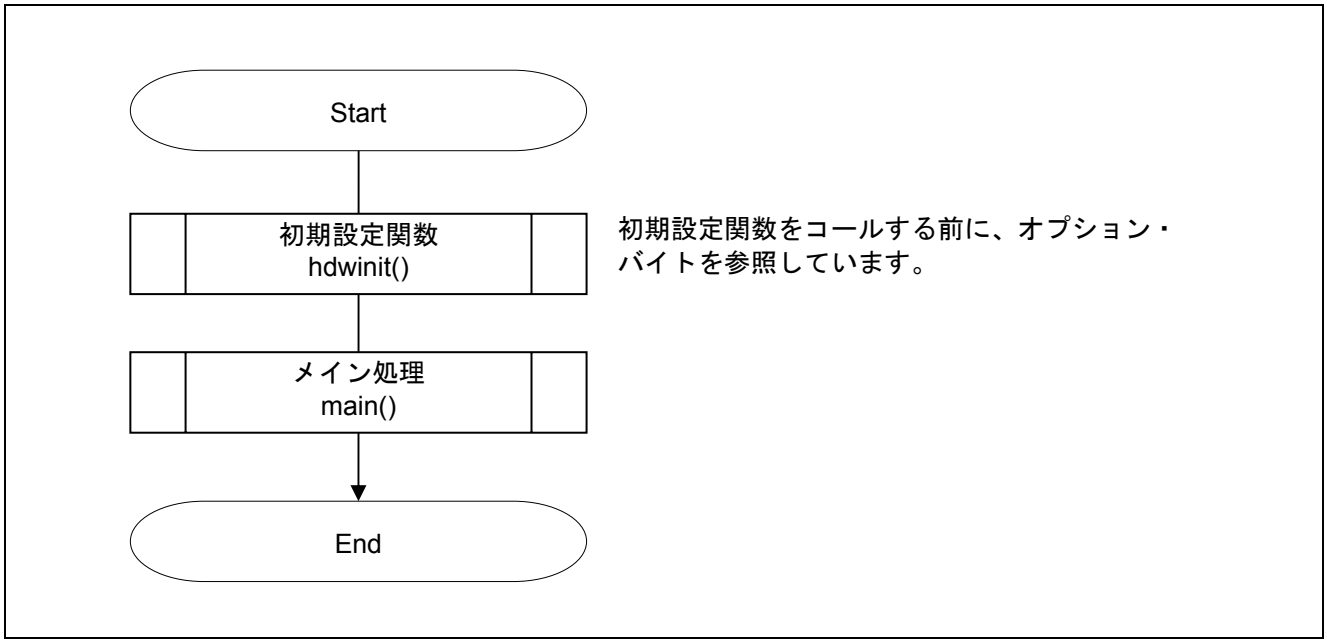

図 5.1 全体フロー

注 初期設定関数の前後でスタートアップ・ルーティンが実行されます。

5.10.1 初期設定関数

図 5.2に**初期設定関数**のフローチャートを示します。

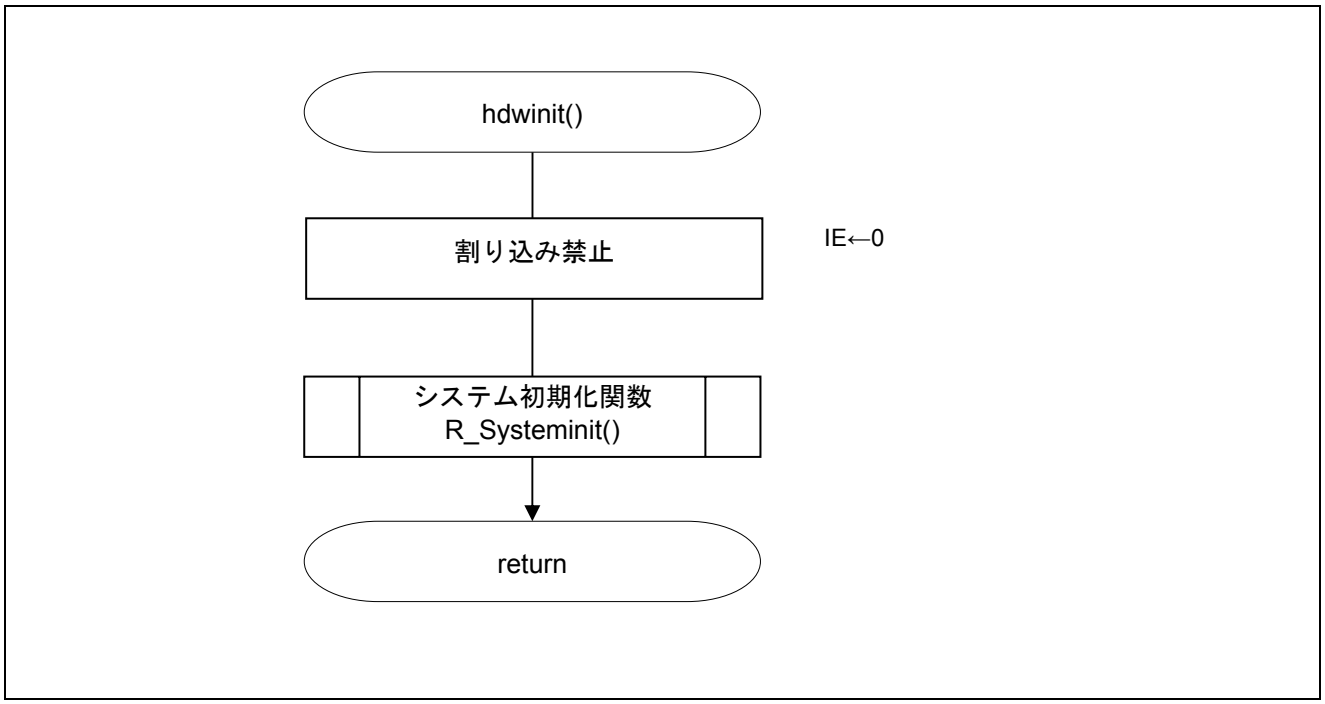

図 5.2 初期設定関数

5.10.2 システム初期化関数

図 5.3にシステム初期化関数のフローチャートを示します。

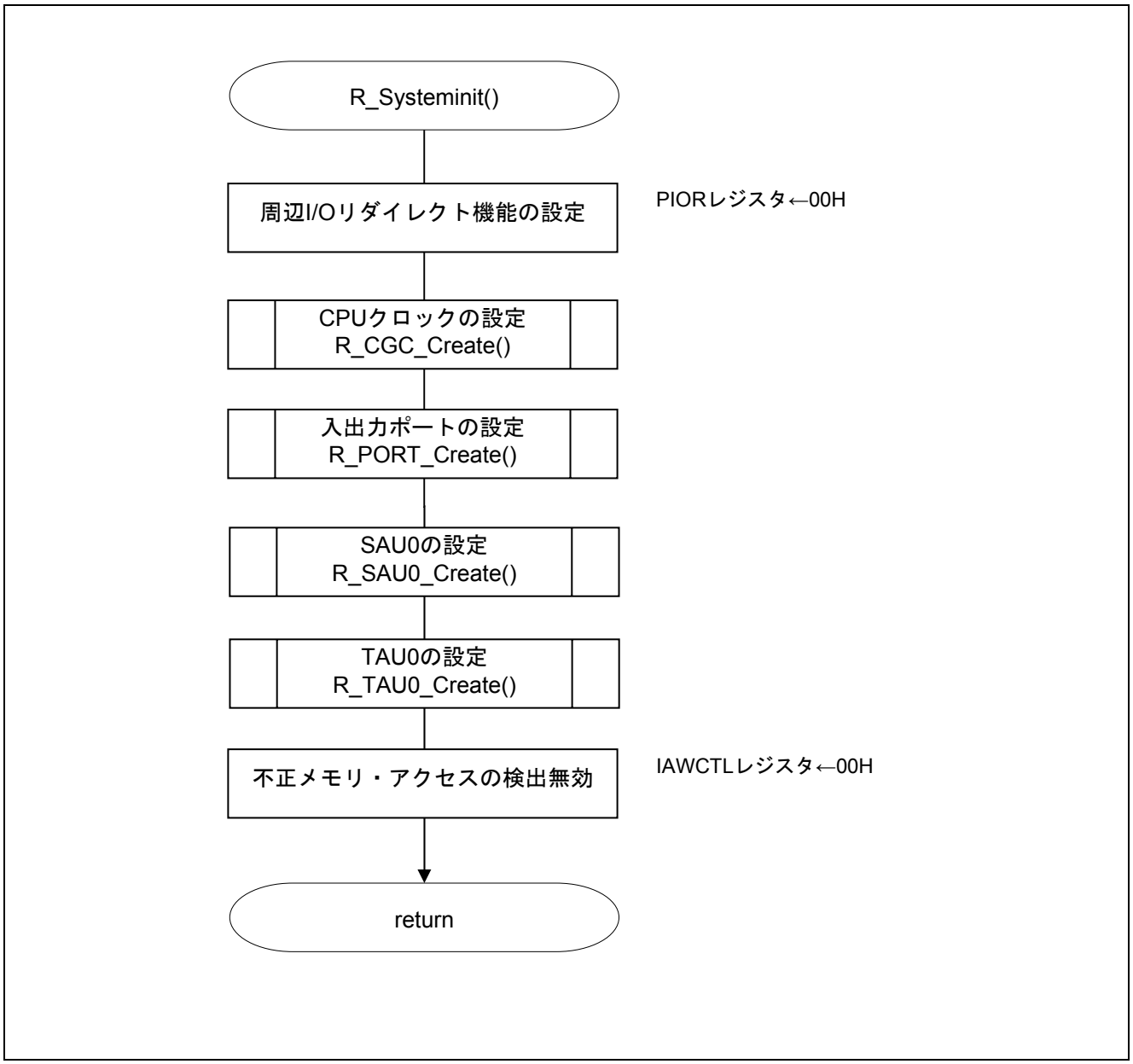

図 5.3 システム初期化関数

5.10.3　入出力ポートの設定

図 5.4に入出力ポートの設定のフローチャートを示します。

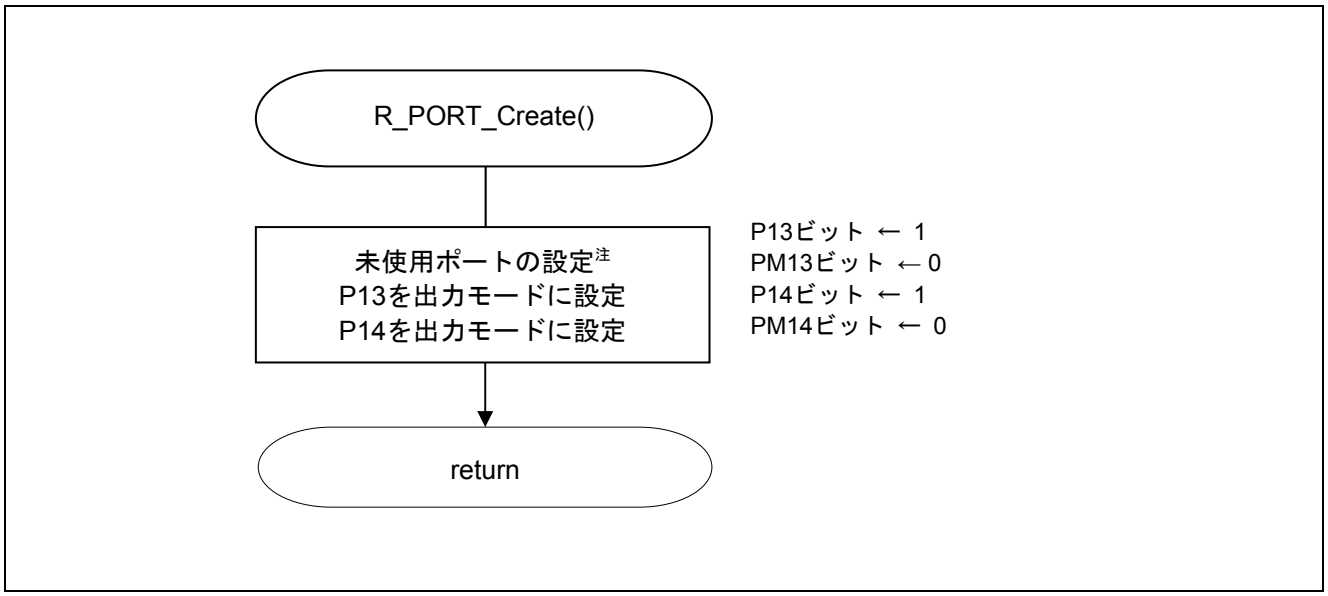

図 5.4　入出力ポートの設定

注　　未使用ポートの設定については、RL78/G12 初期設定（R01AN2582J）アプリケーションノート“フローチャート”を参照して下さい。

注意　　未使用のポートは、端子処理などを適切に行い、電気的特性を満たすように設計してください。
また、未使用の入力専用ポートは個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい。

5.10.4　CPU クロックの設定

図 5.5にCPU クロックの設定のフローチャートを示します。

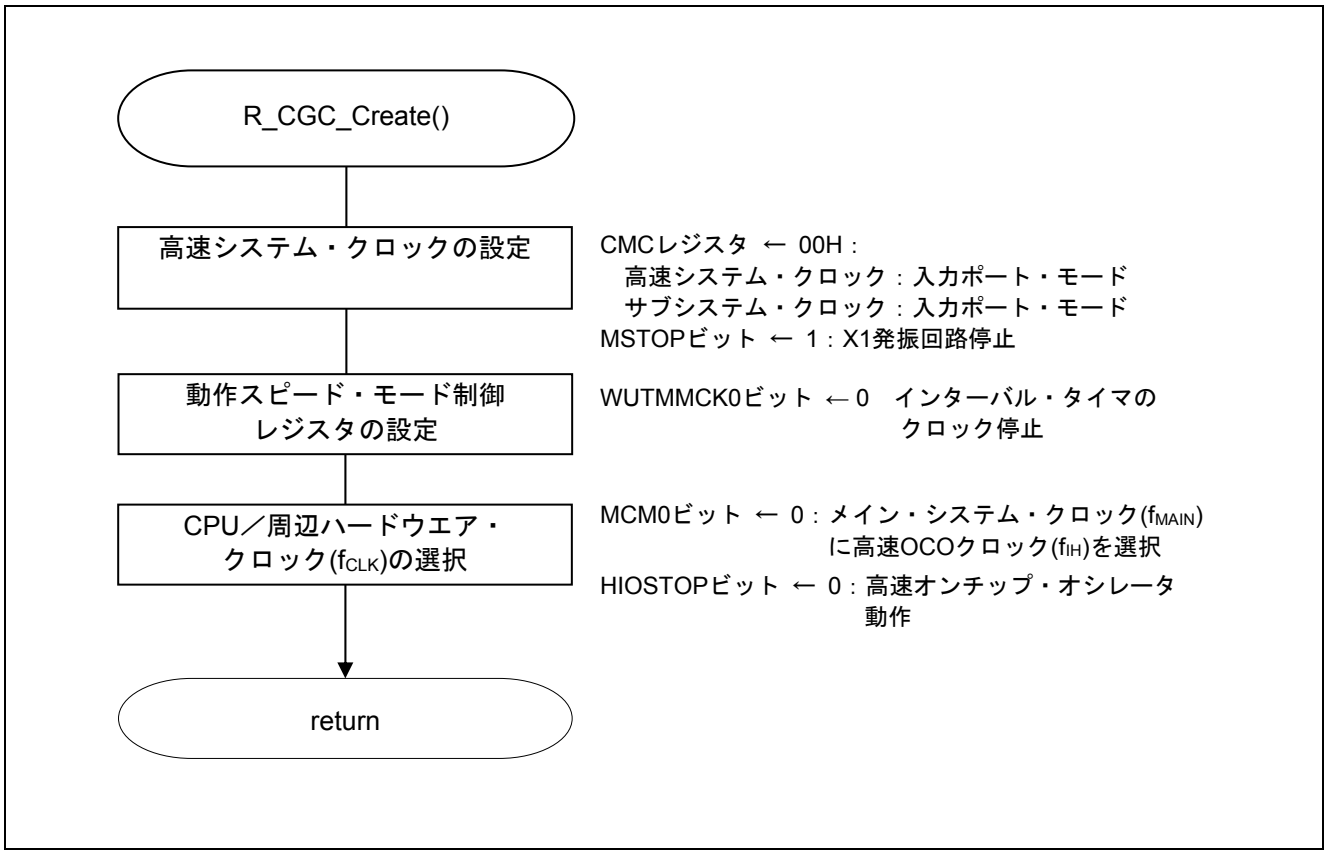

図 5.5　CPU クロックの設定

注意　CPU クロックの設定（R_CGC_Create()）については、RL78/G12 初期設定（R01AN2582J）アプリケーションノート"フローチャート"を参照して下さい。

5.10.5 SAU0 の設定

図 5.6にSAU0 の設定のフローチャートを示します。

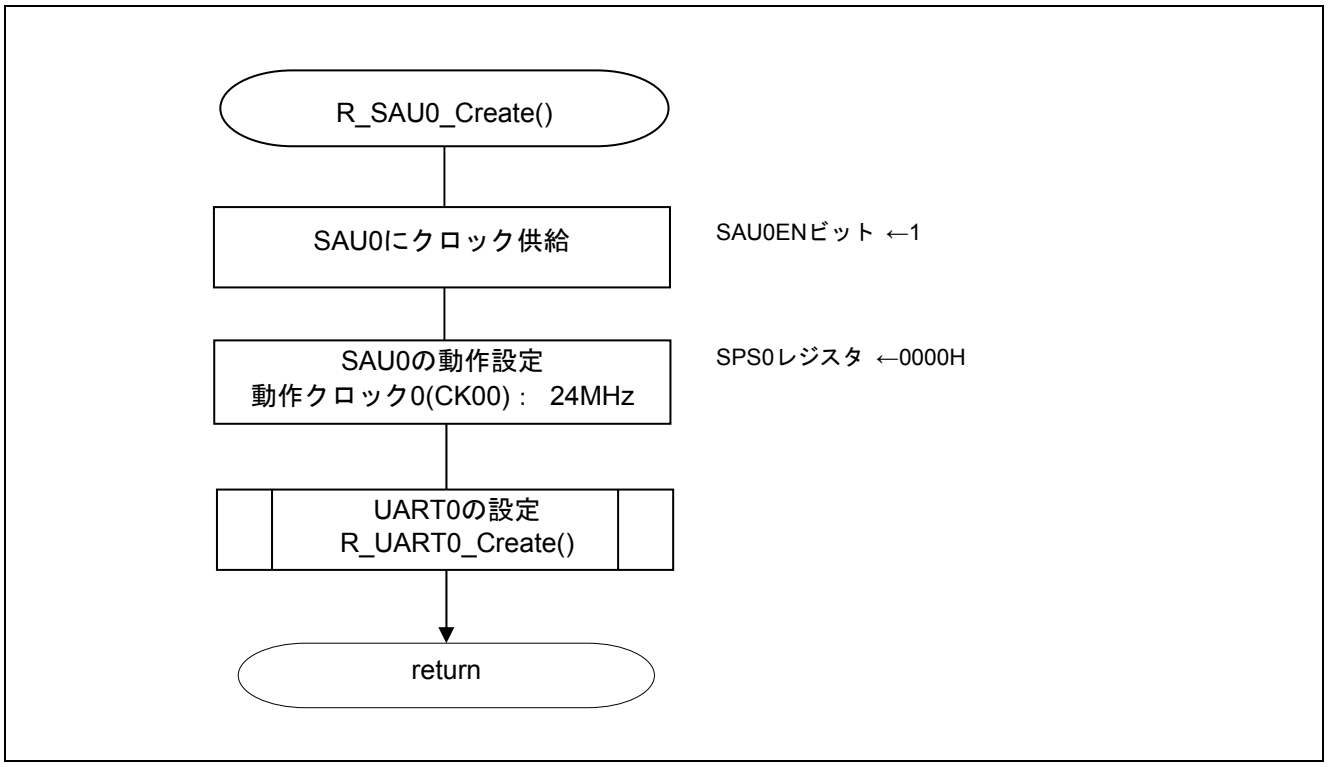

図 5.6 SAU0 の設定

5.10.6　UART0 の設定

図 5.7に UART0 の設定(1/3)、図 5.8 に UART0 の設定(2/3)、図 5.9 に UART0 の設定(3/3)のフローチャートを示します。

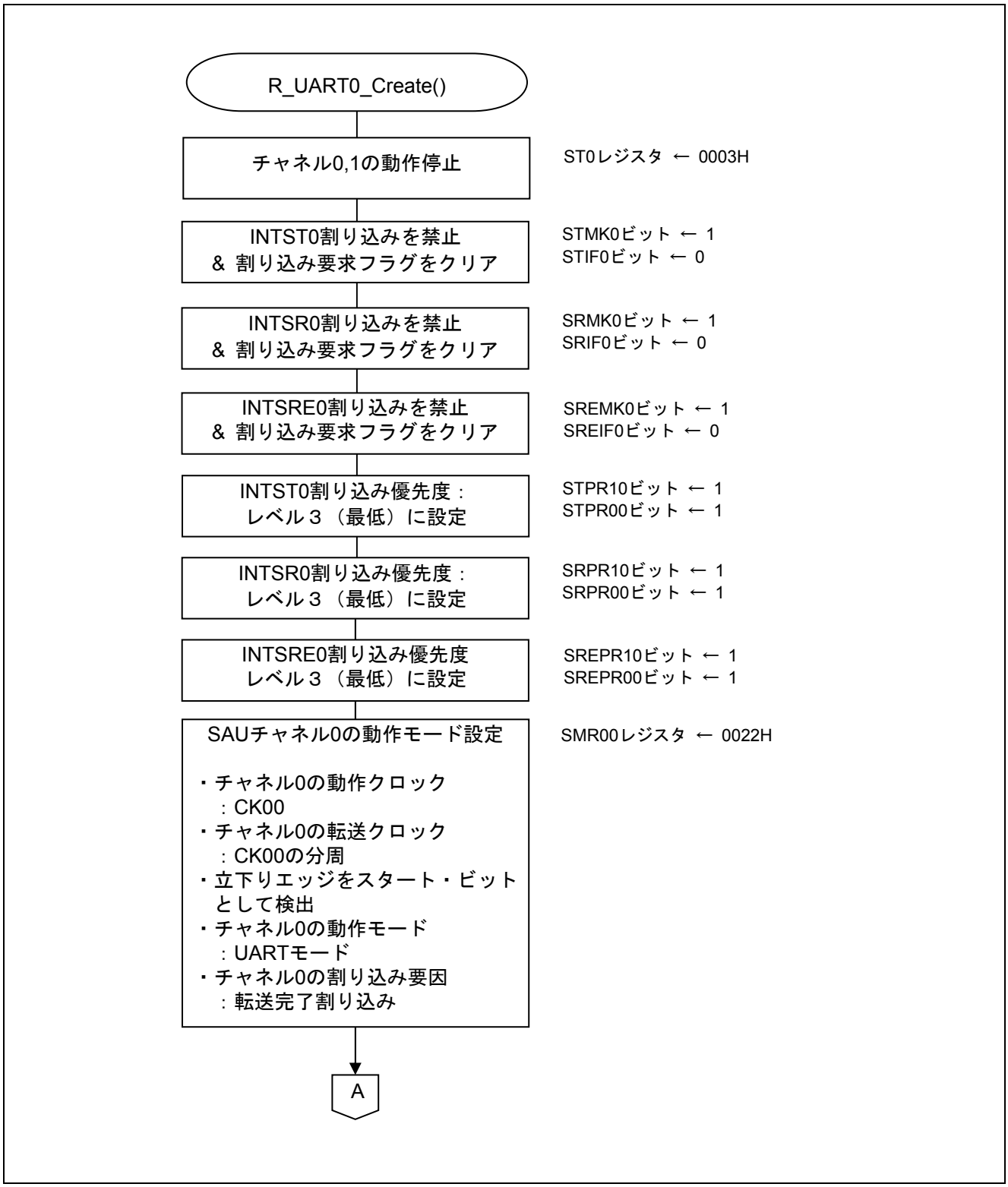

図 5.7　UART0 の設定（1/3）

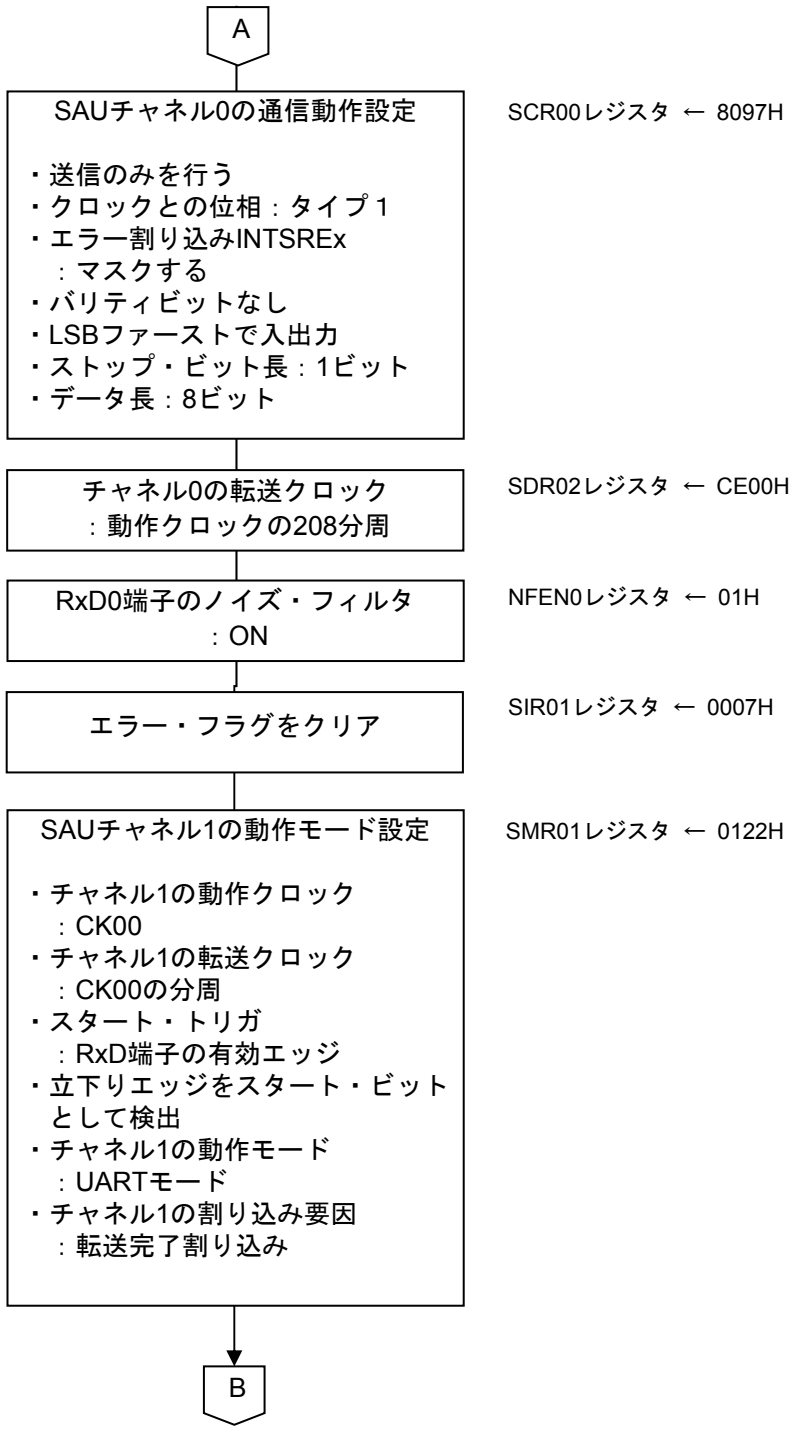

図 5.8　UART0 の設定（2/3）

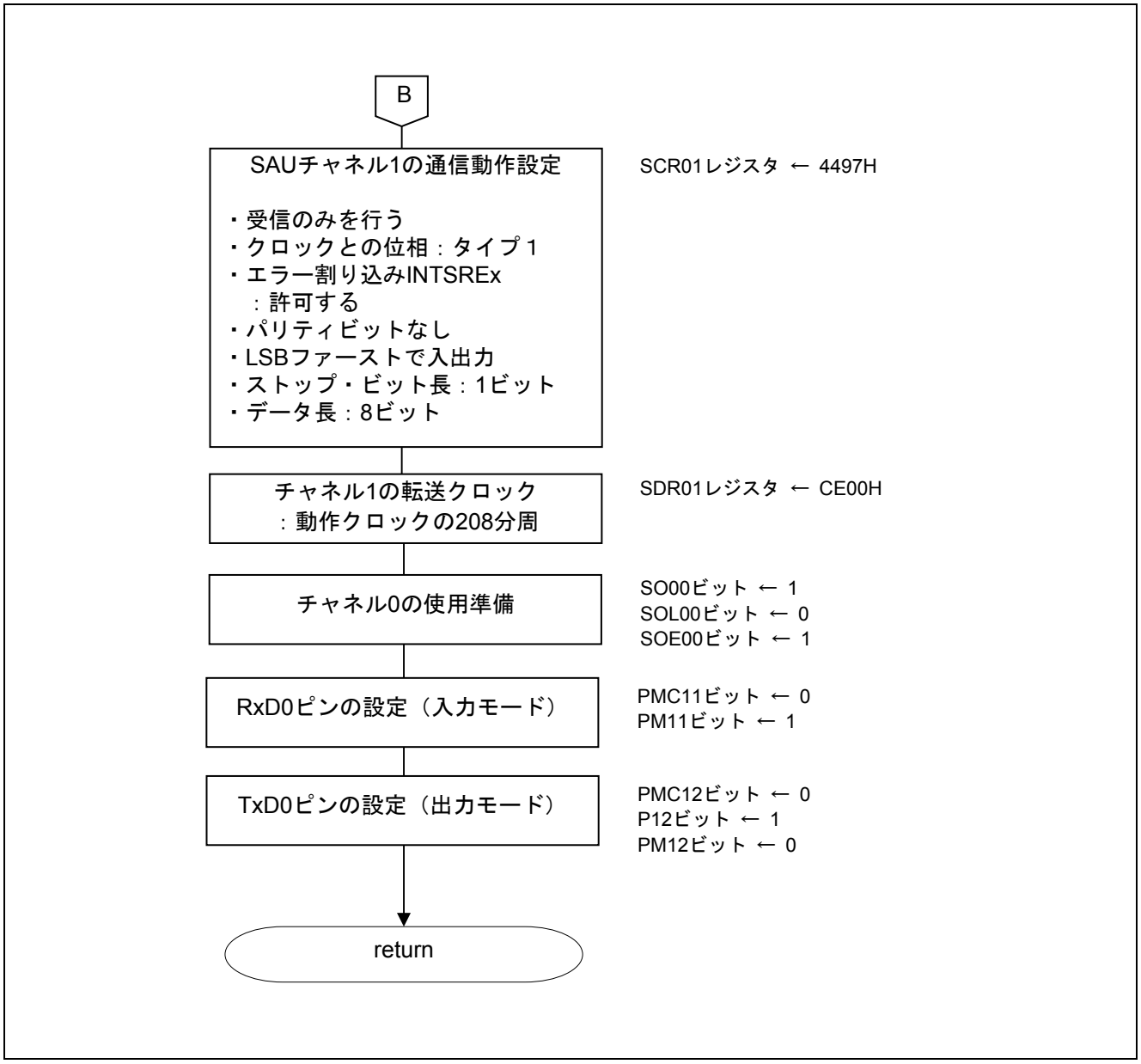

図 5.9 UART0 の設定（3/3）

5.10.7 TAU0 の設定

図 5.10にTAU0 の設定のフローチャートを示します。

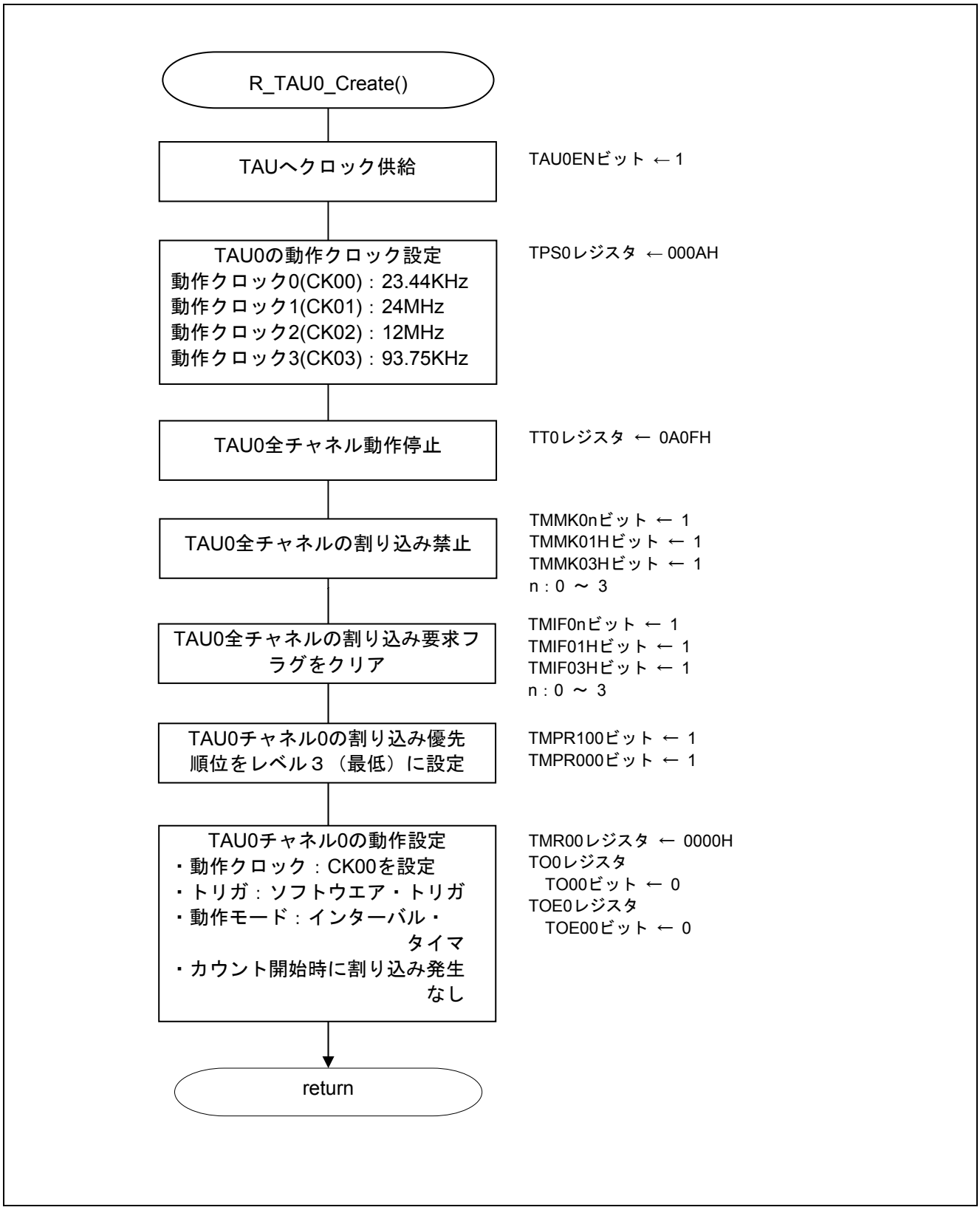

図 5.10 TAU0 の設定

5.10.8 メイン処理

図 5.11にメイン処理(1/2)、図 5.12にメイン処理(2/2)のフローチャートを示します。

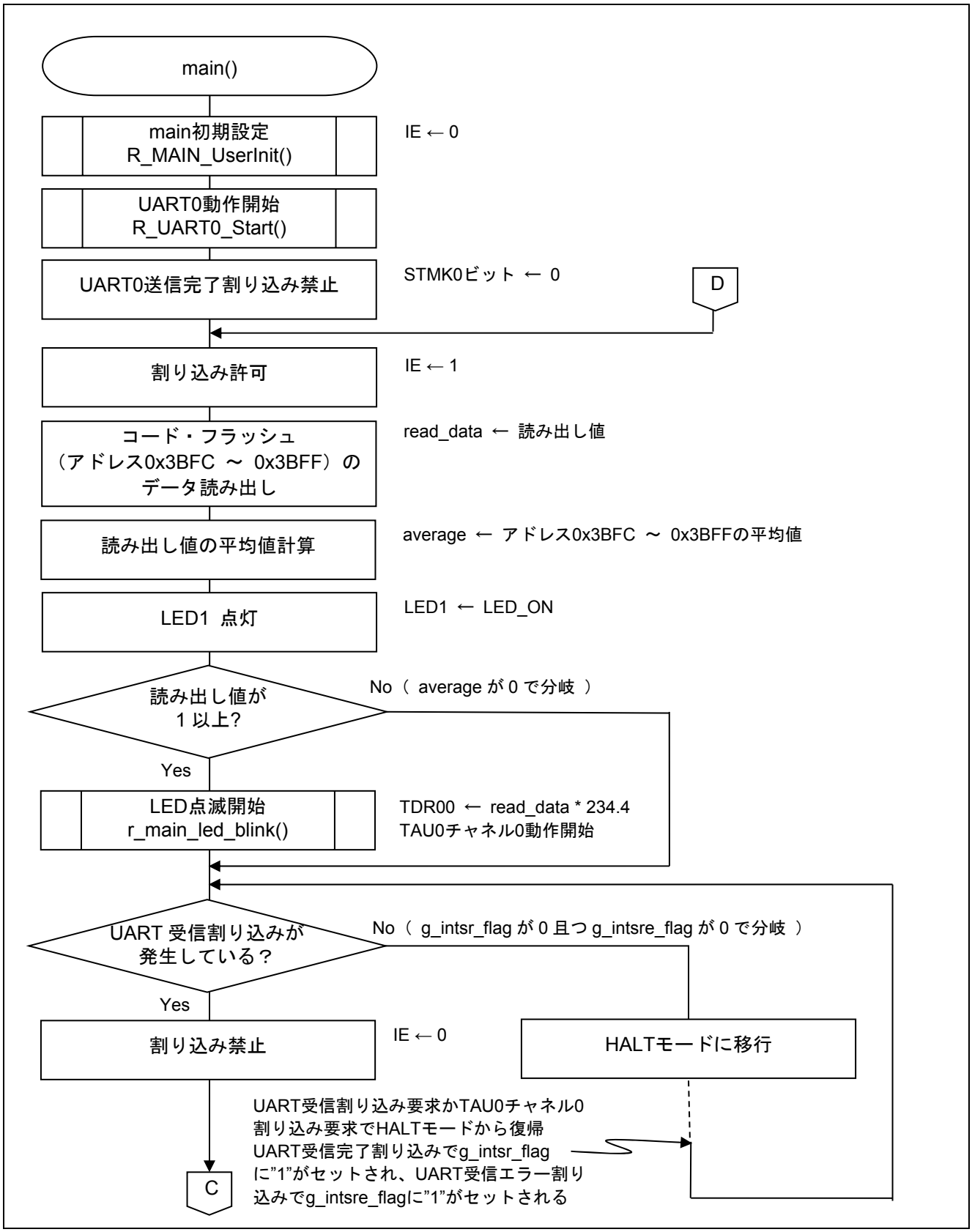

図 5.11 メイン処理(1/2)

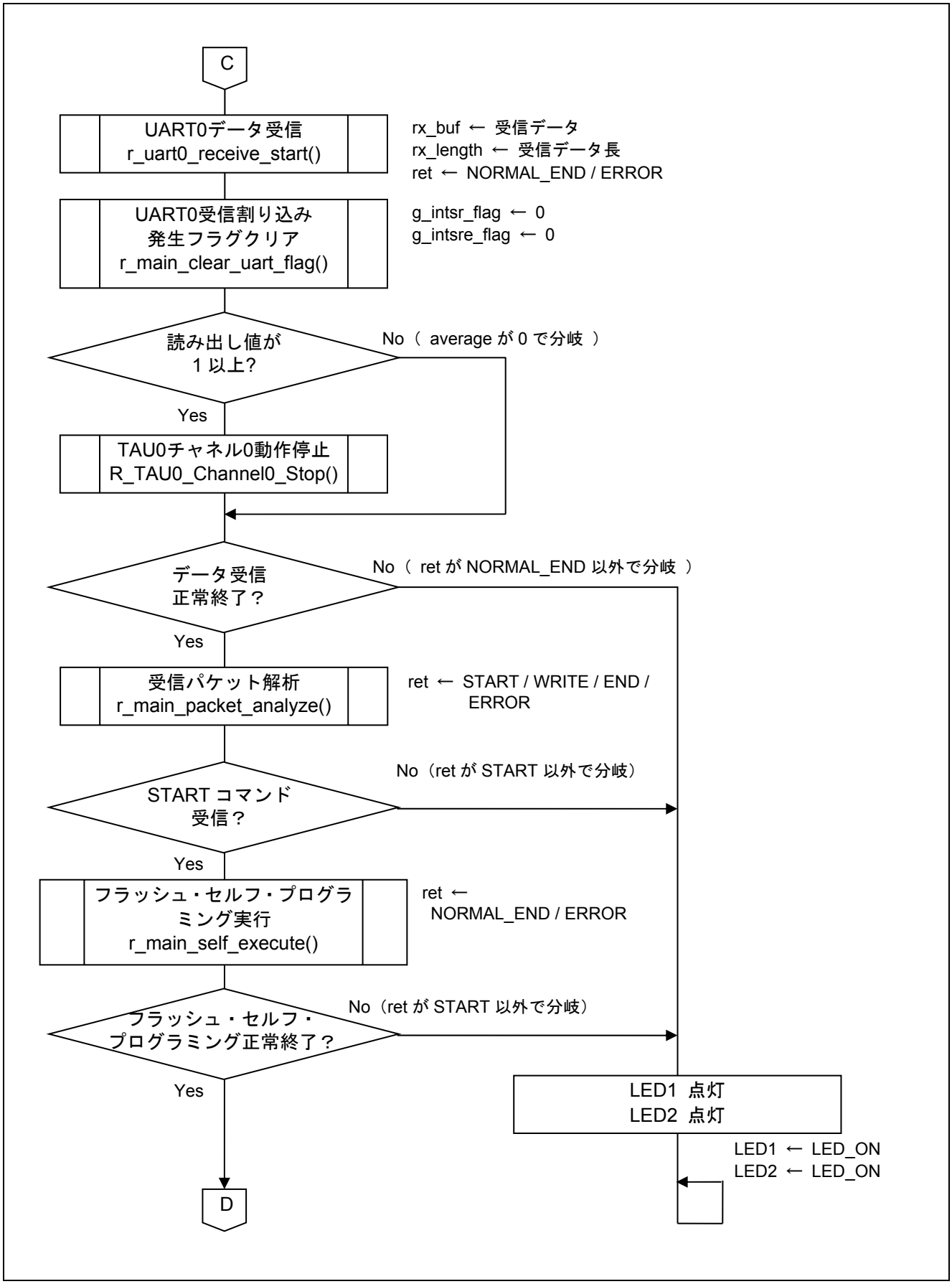

図 5.12 メイン処理(2/2)

5.10.9 メイン初期設定

図 5.13 にメイン初期設定のフローチャートを示します。

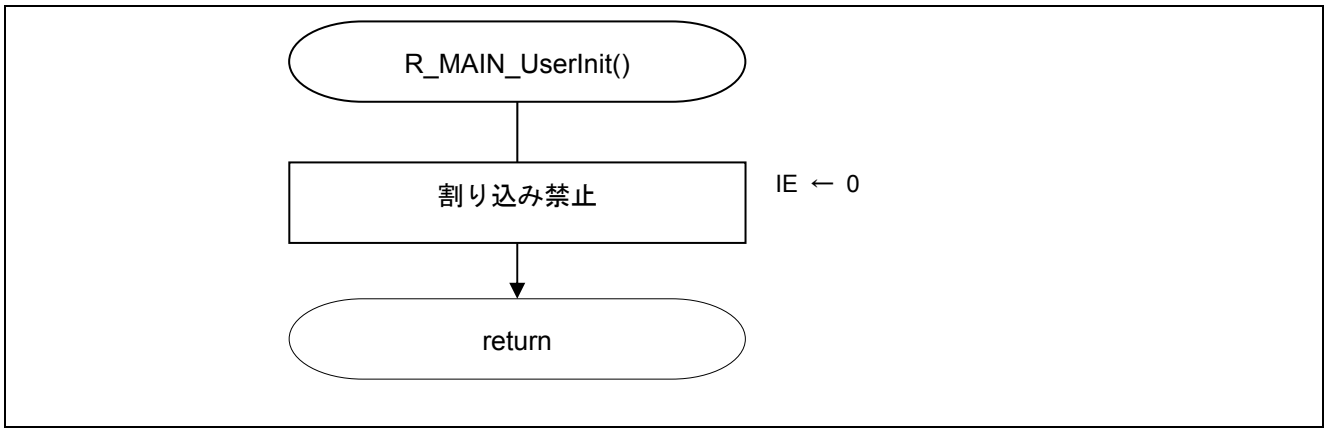

図 5.13 メイン初期設定

5.10.10 UART0 動作開始

図 5.14 にUART0 動作開始のフローチャートを示します。

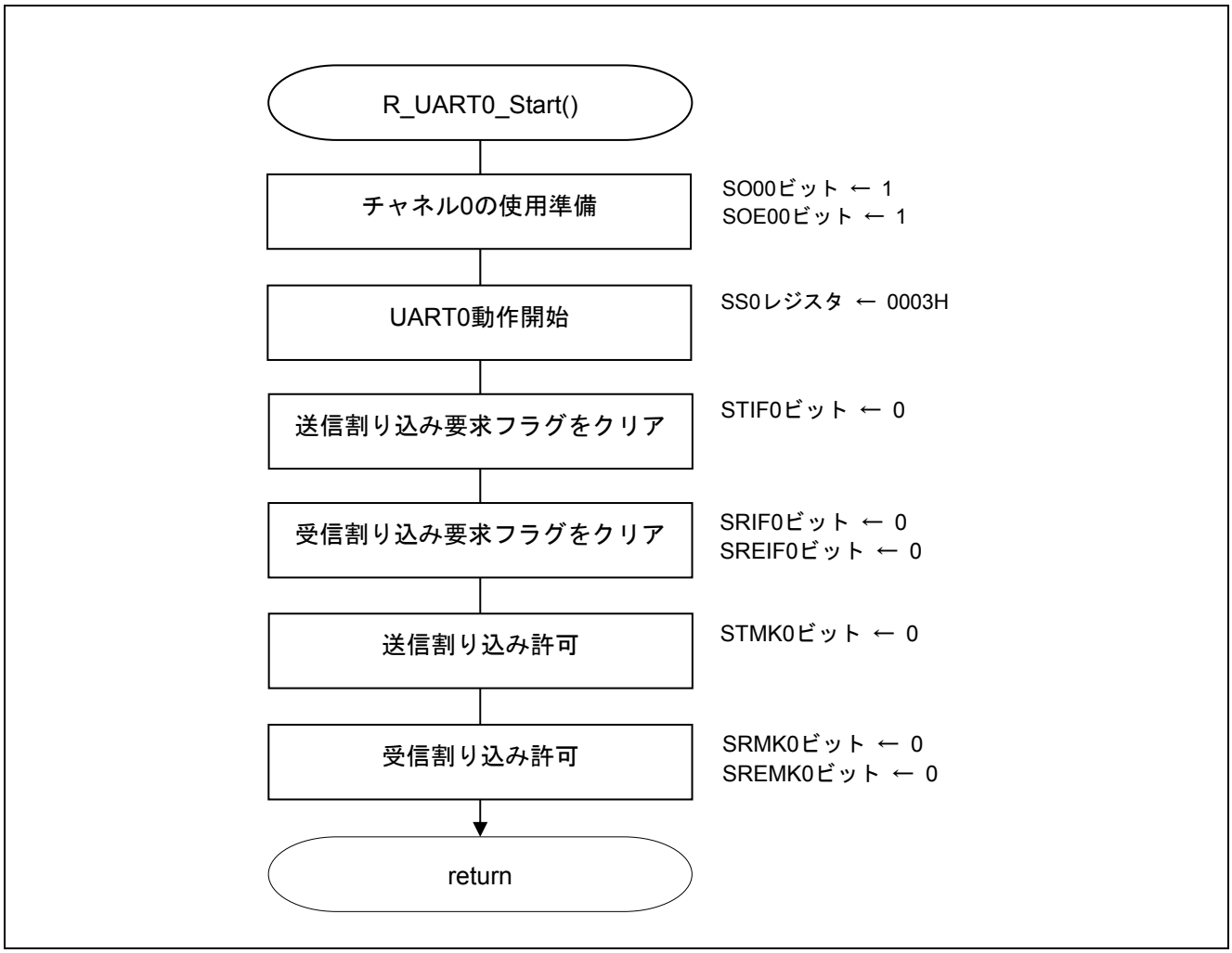

図 5.14 UART0 動作開始

5.10.11 UART0 受信完了割り込み

図 5.15 にUART0 受信完了割り込みのフローチャートを示します。

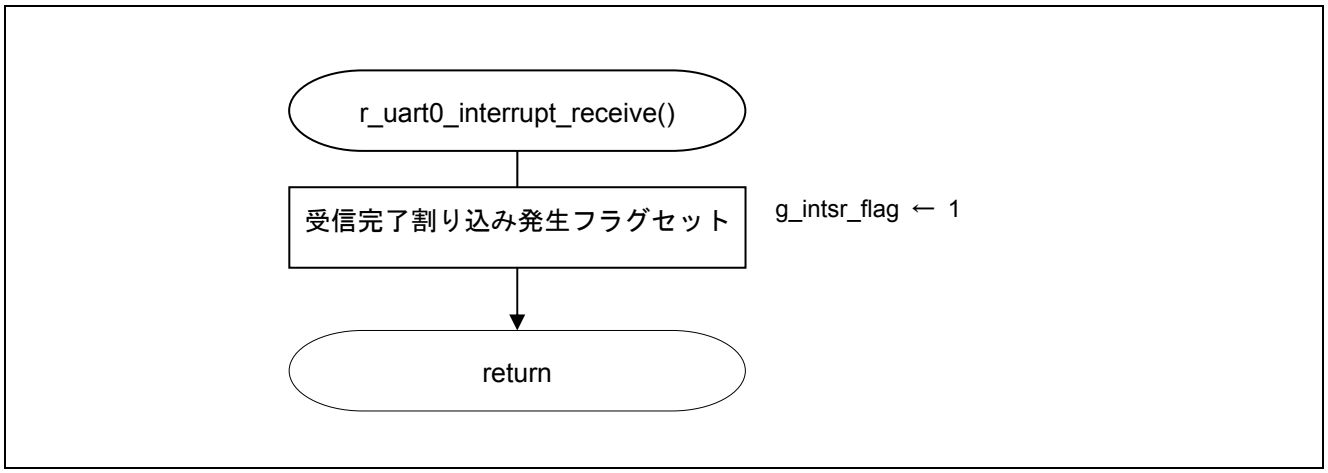

図 5.15 UART0 受信完了割り込み

5.10.12 UART0 受信エラー割り込み

図 5.16 にUART0 受信エラー割り込みのフローチャートを示します。

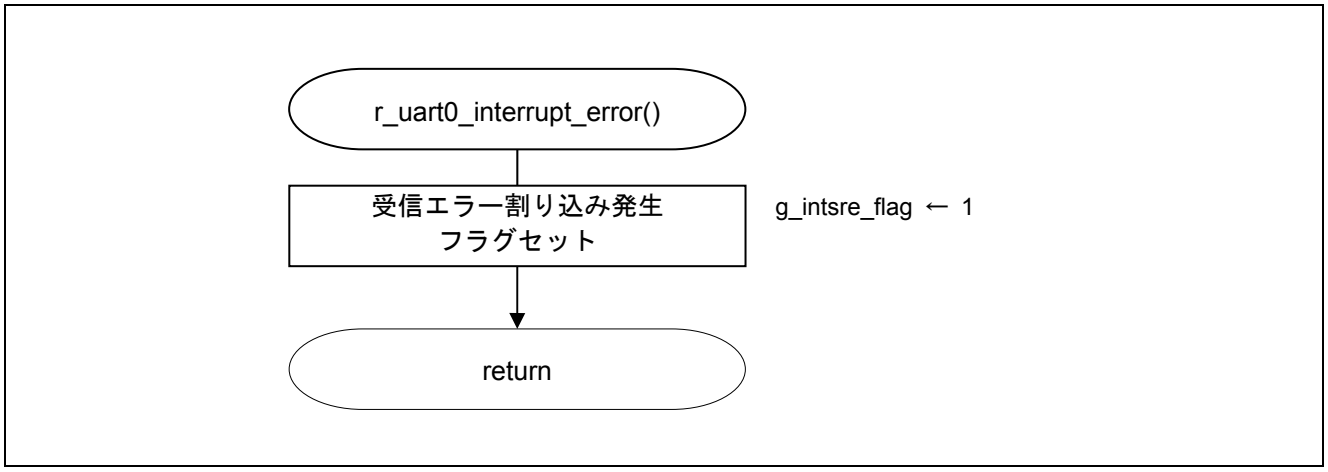

図 5.16 UART0 受信エラー割り込み

5.10.13 LED 点滅開始

図 5.17 にLED 点滅開始のフローチャートを示します。

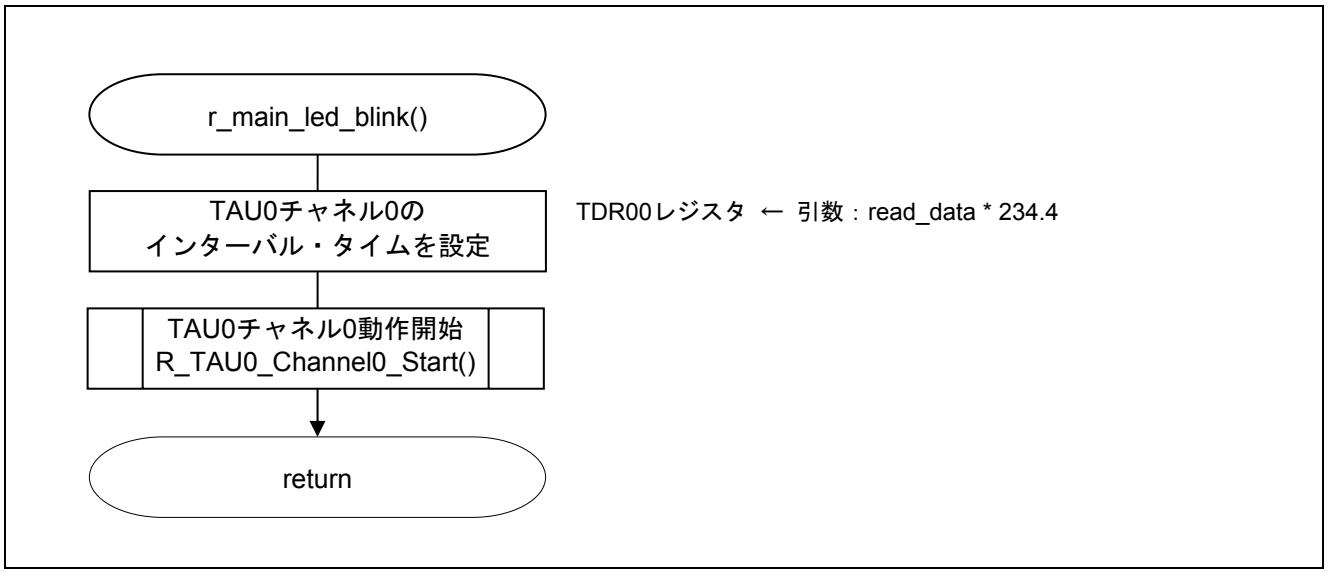

図 5.17 LED 点滅開始

5.10.14 TAU0 チャネル 0 動作開始

図 5.18 にTAU0 チャネル 0 動作開始のフローチャートを示します。

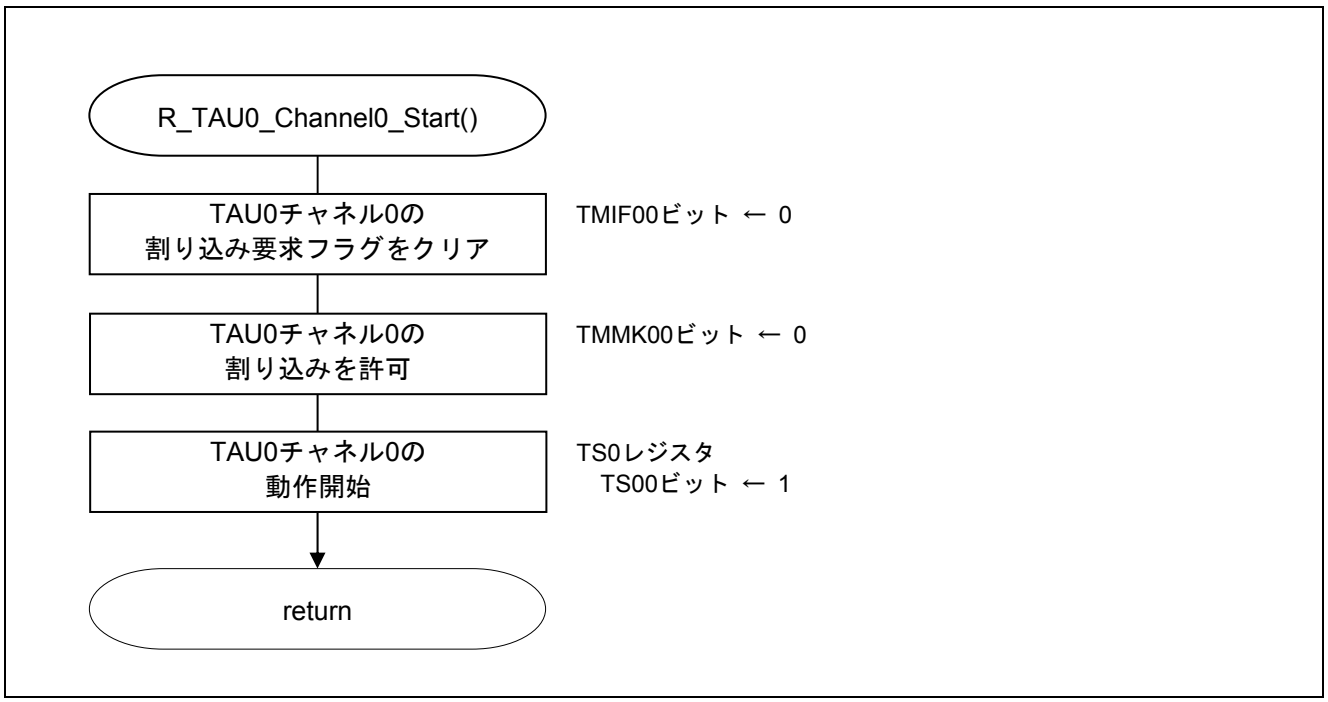

図 5.18 TAU0 チャネル 0 動作開始

5.10.15 TAU0 チャネル 0 割り込み

図 5.19 にTAU0 チャネル 0 割り込みのフローチャートを示します。

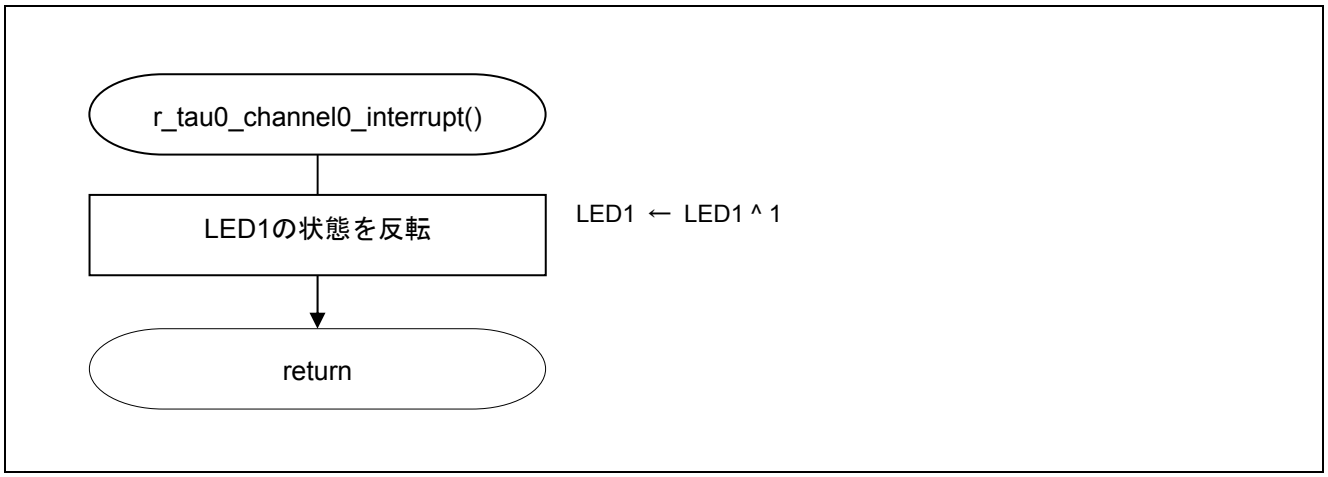

図 5.19 TAU0 チャネル 0 割り込み

5.10.16 UART0 データ受信

図 5.20 にUART0 データ受信(1/2)、図 5.21 にUART0 データ受信(2/2)のフローチャートを示します。

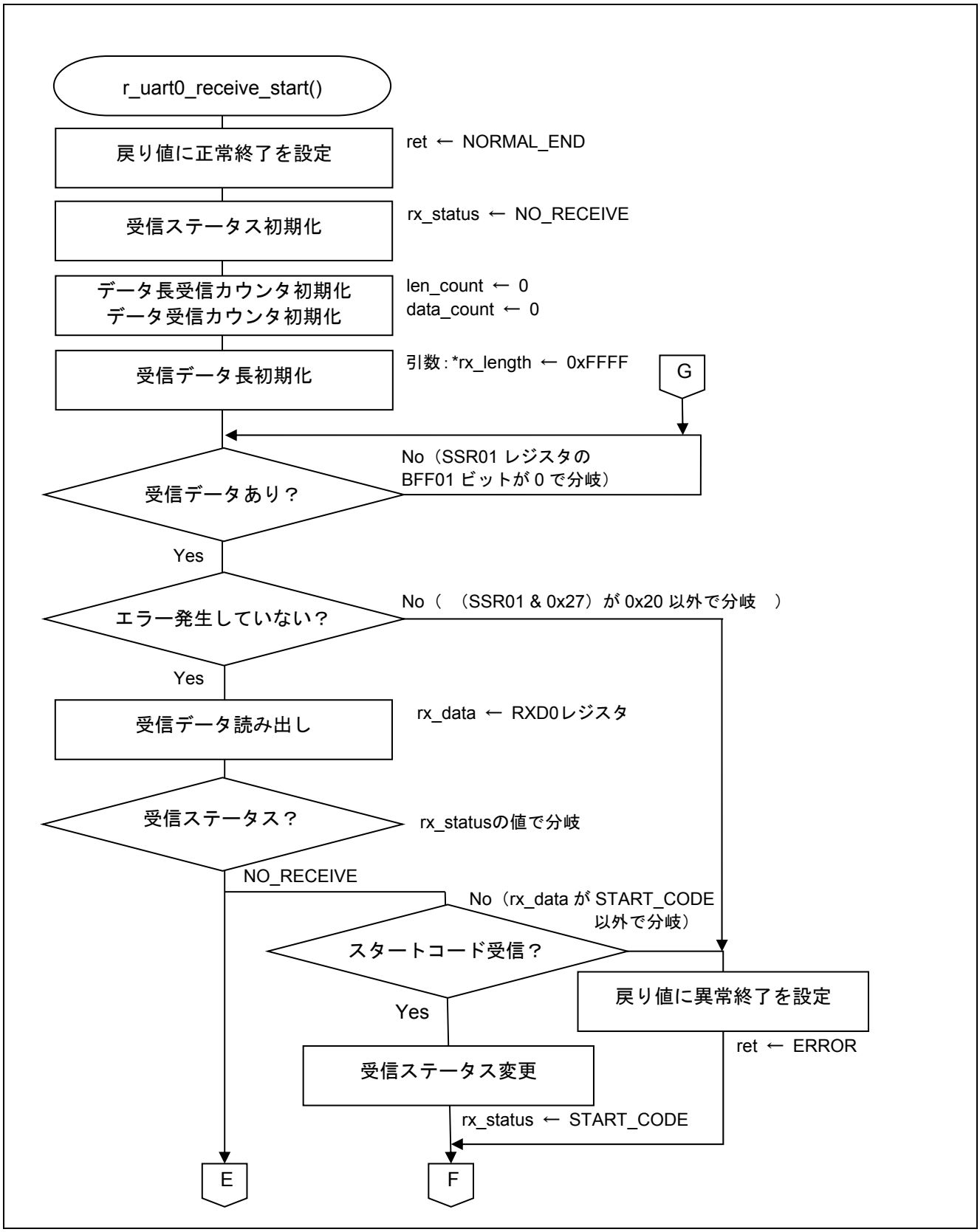

図 5.20 UART0 データ受信(1/2)

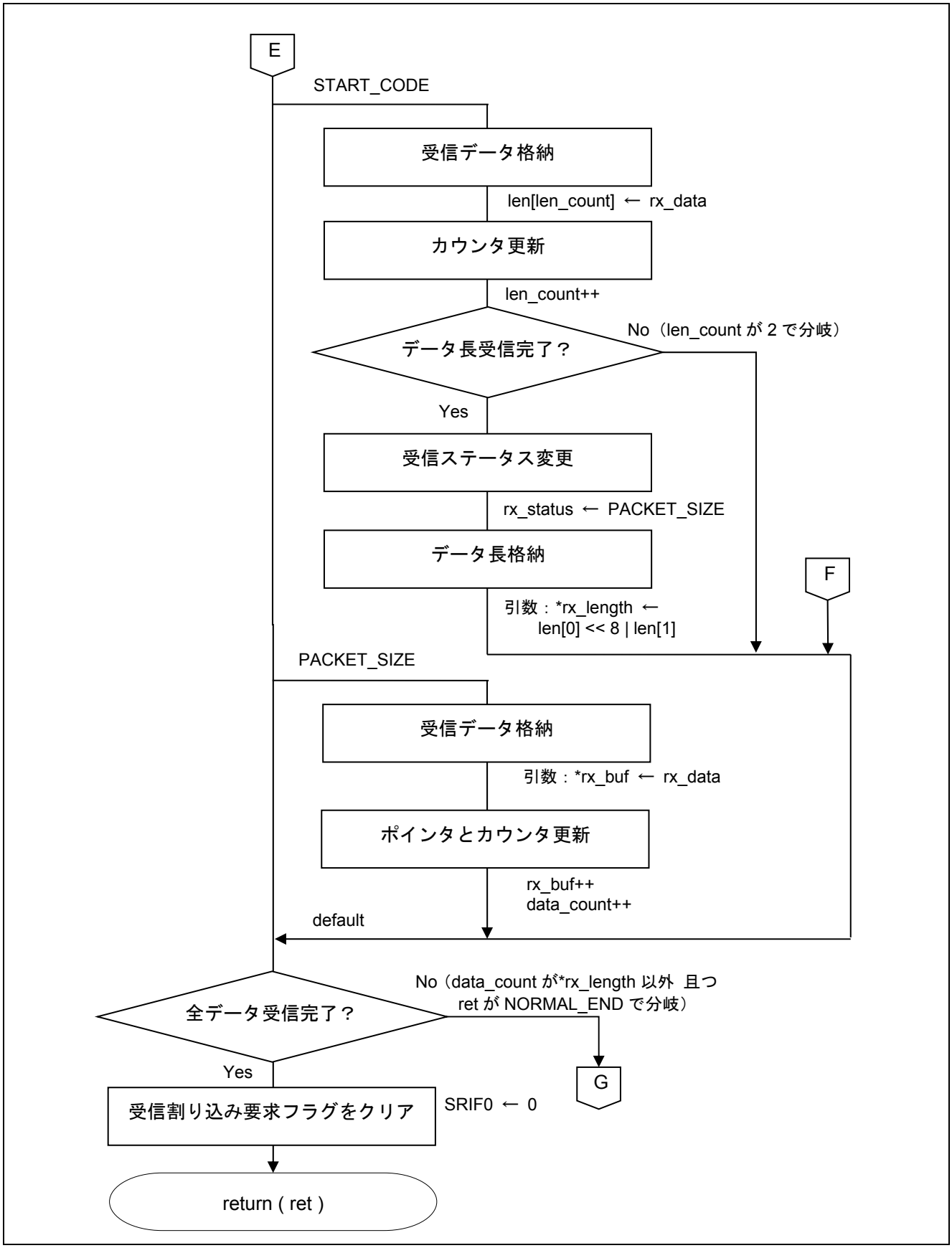

図 5.21 UART0 データ受信(2/2)

5.10.17 UART0 受信割り込み発生フラグクリア

図 5.22 にUART0 受信割り込み発生フラグクリアのフローチャートを示します。

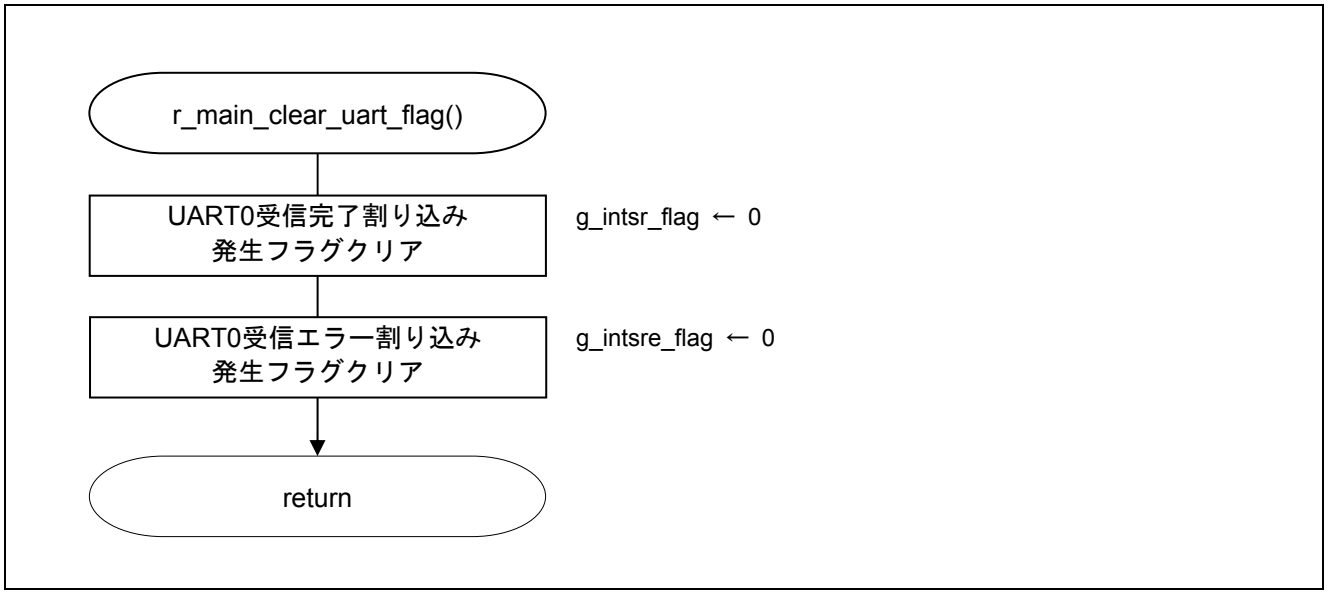

図 5.22 UART0 受信割り込み発生フラグクリア

5.10.18 TAU0 チャネル 0 動作停止

図 5.23 にTAU0 チャネル 0 動作停止のフローチャートを示します。

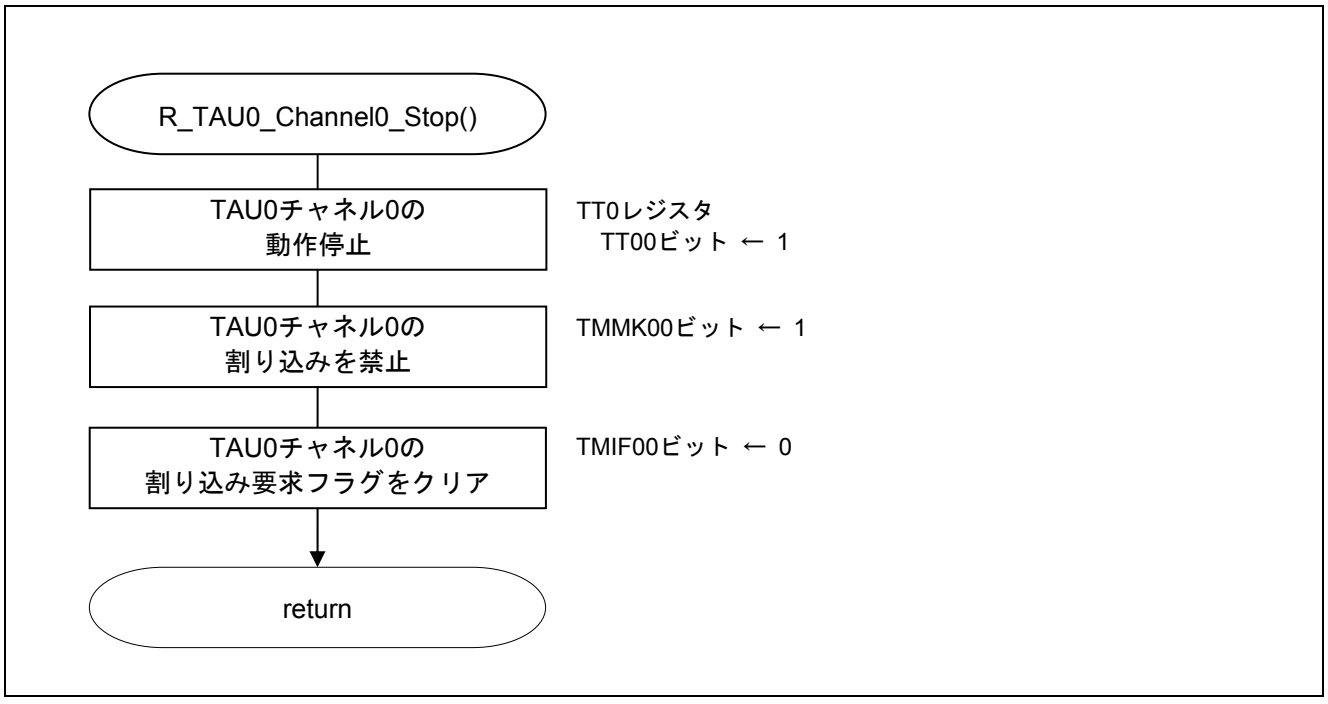

図 5.23 TAU0 チャネル 0 動作停止

5.10.19 受信パケット解析

図 5.24 に受信パケット解析のフローチャートを示します。

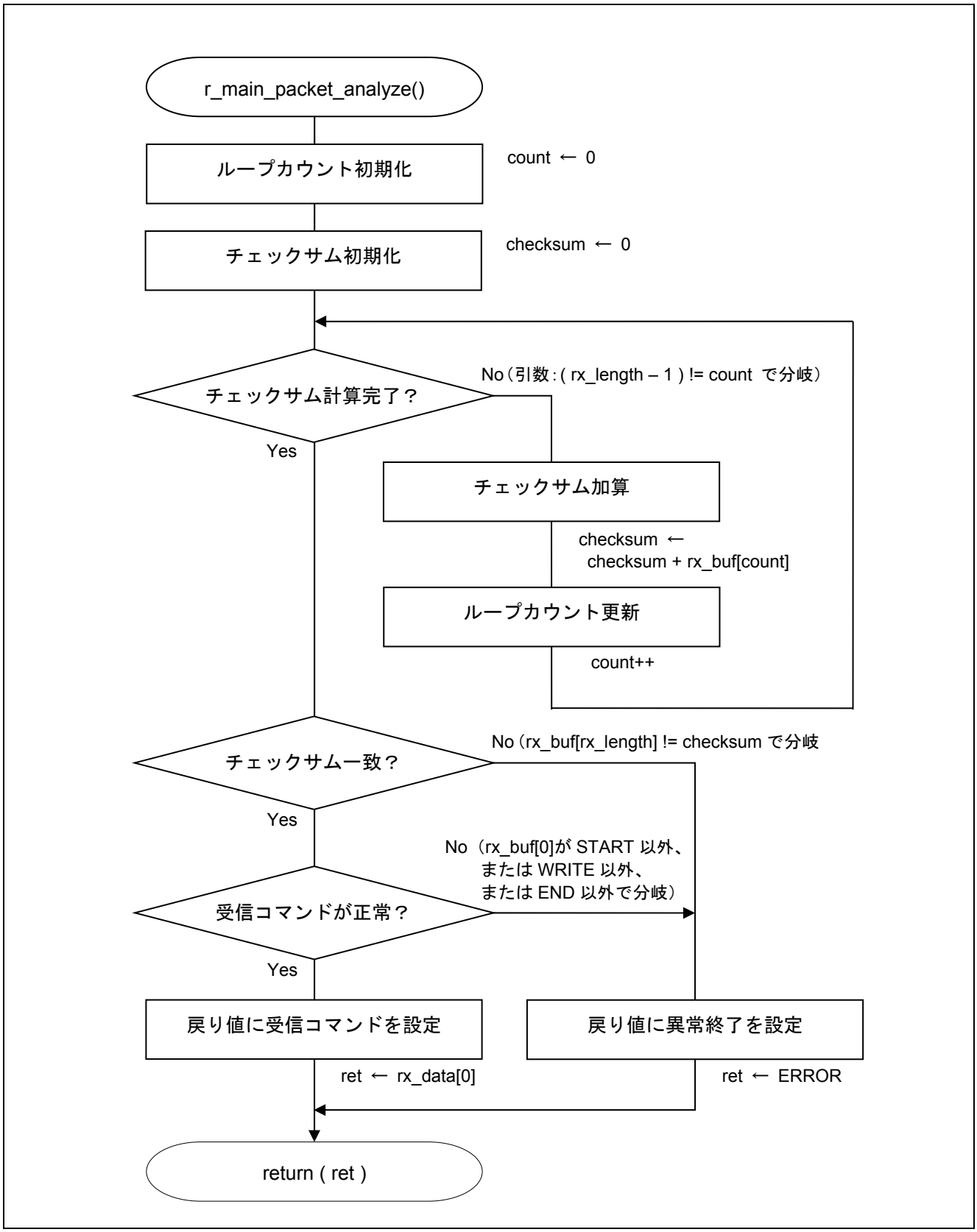

図 5.24 受信パケット解析

5.10.20 フラッシュ・セルフ・プログラミング実行

図 5.25 にフラッシュ・セルフ・プログラミング実行のフローチャートを示します。

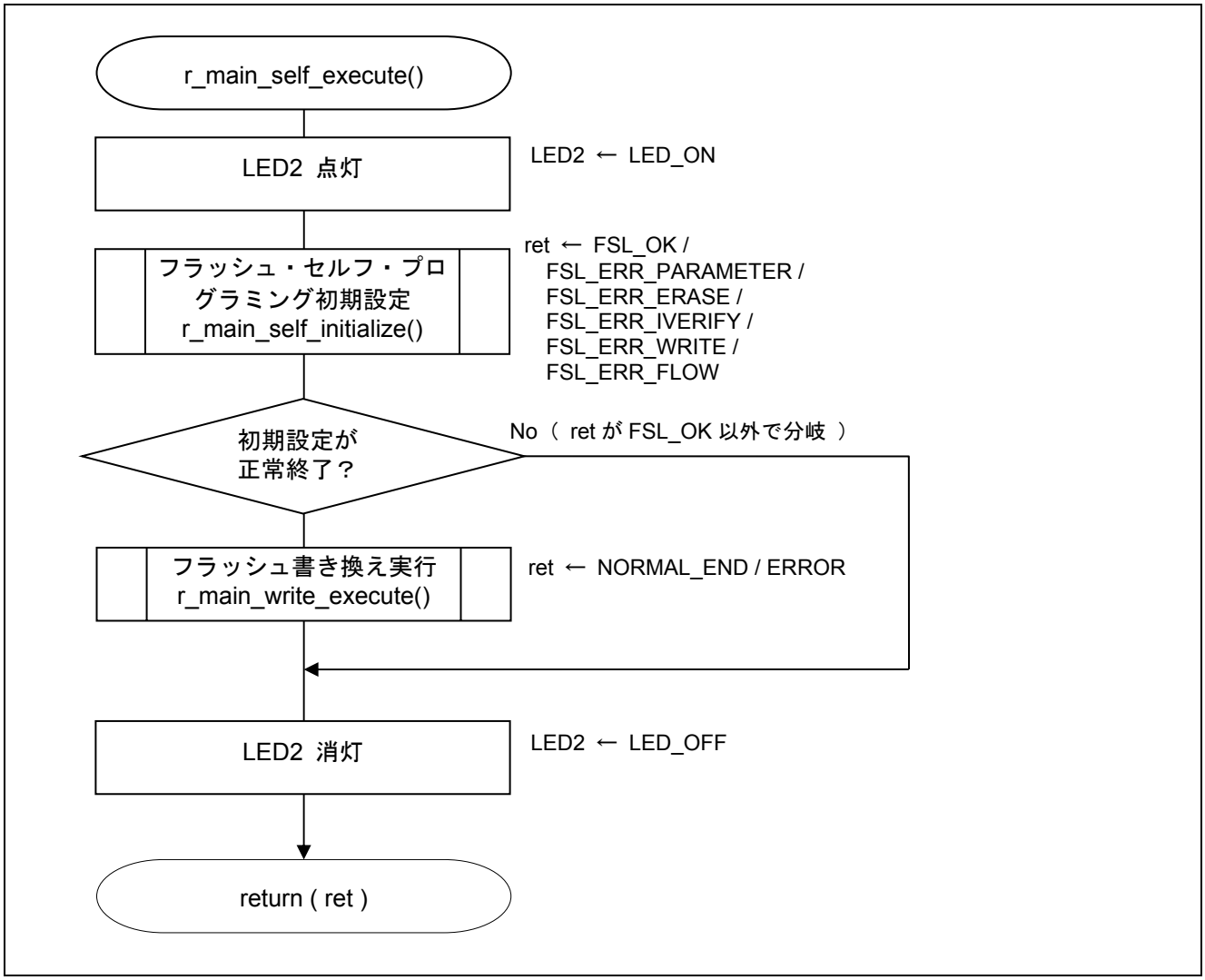

図 5.25 フラッシュ・セルフ・プログラミング実行

5.10.21 フラッシュ・セルフ・プログラミング初期設定

図 5.26 にフラッシュ・セルフ・プログラミング初期設定のフローチャートを示します。

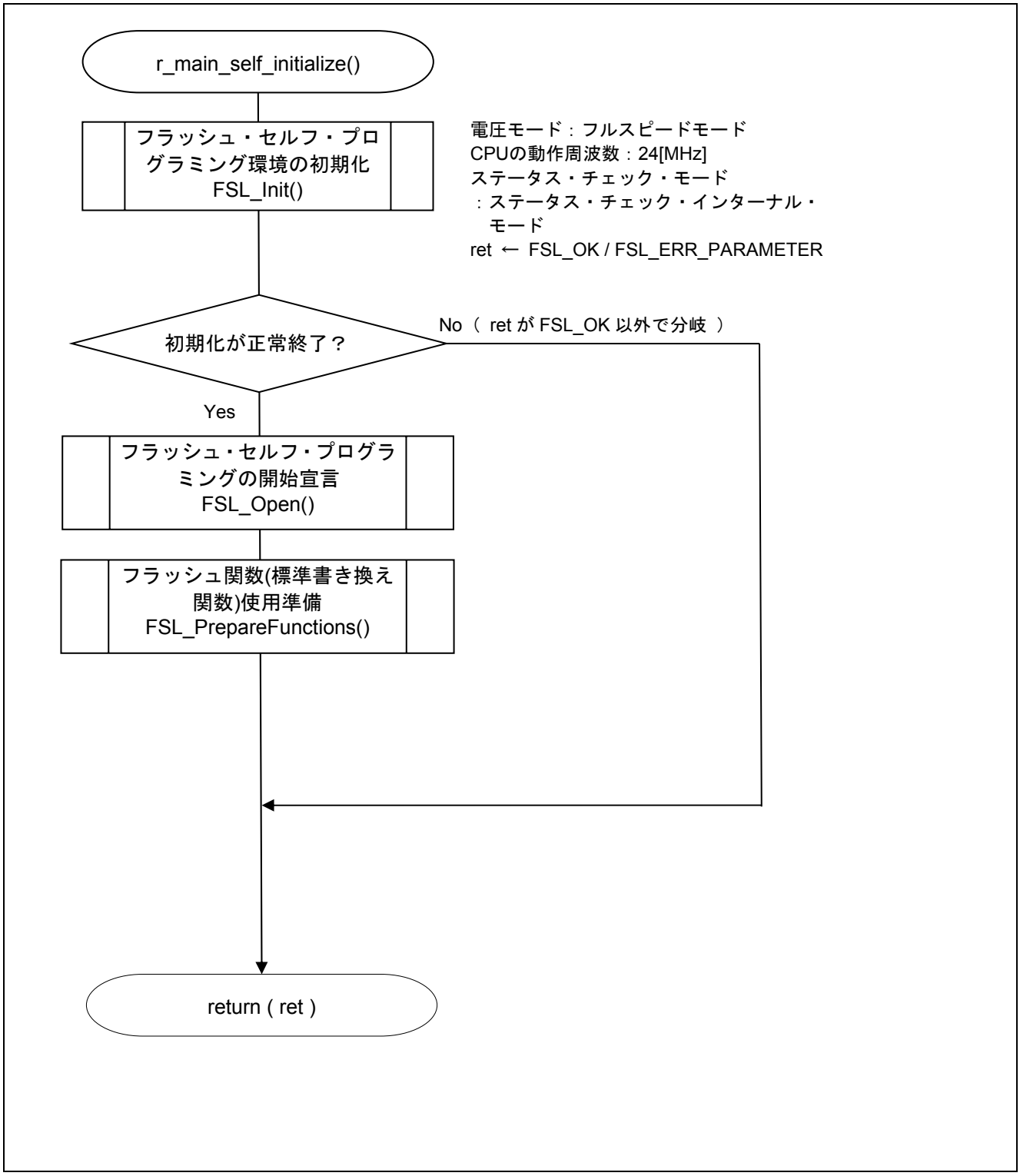

図 5.26 フラッシュ・セルフ・プログラミング初期設定

5.10.22 フラッシュ書き換え実行

図 5.27 にフラッシュ書き換え実行(1/3)、図 5.28 にフラッシュ書き換え実行(2/3)、図 5.29 にフラッシュ書き換え実行(3/3)のフローチャートを示します。

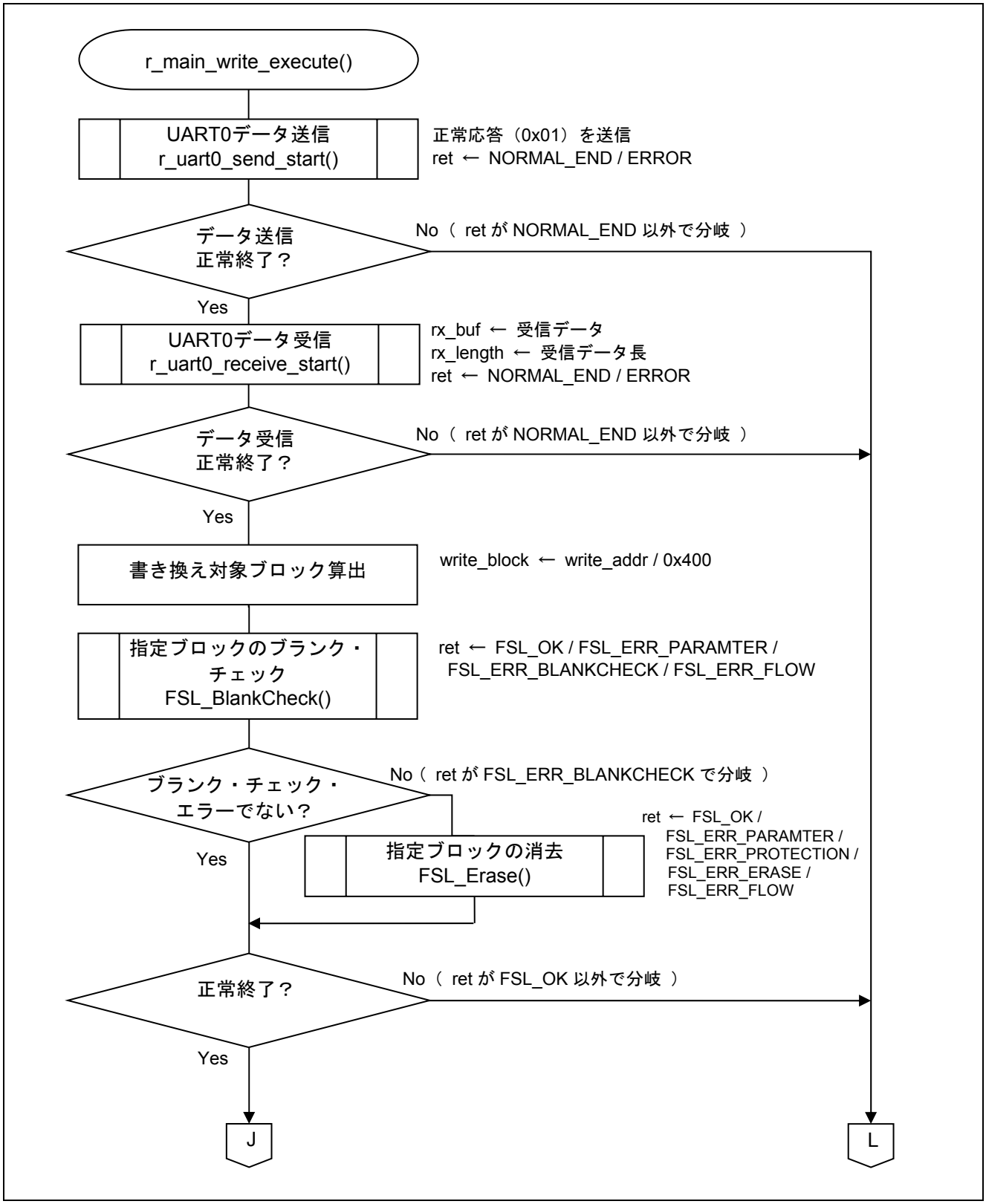

図 5.27 フラッシュ書き換え実行(1/3)

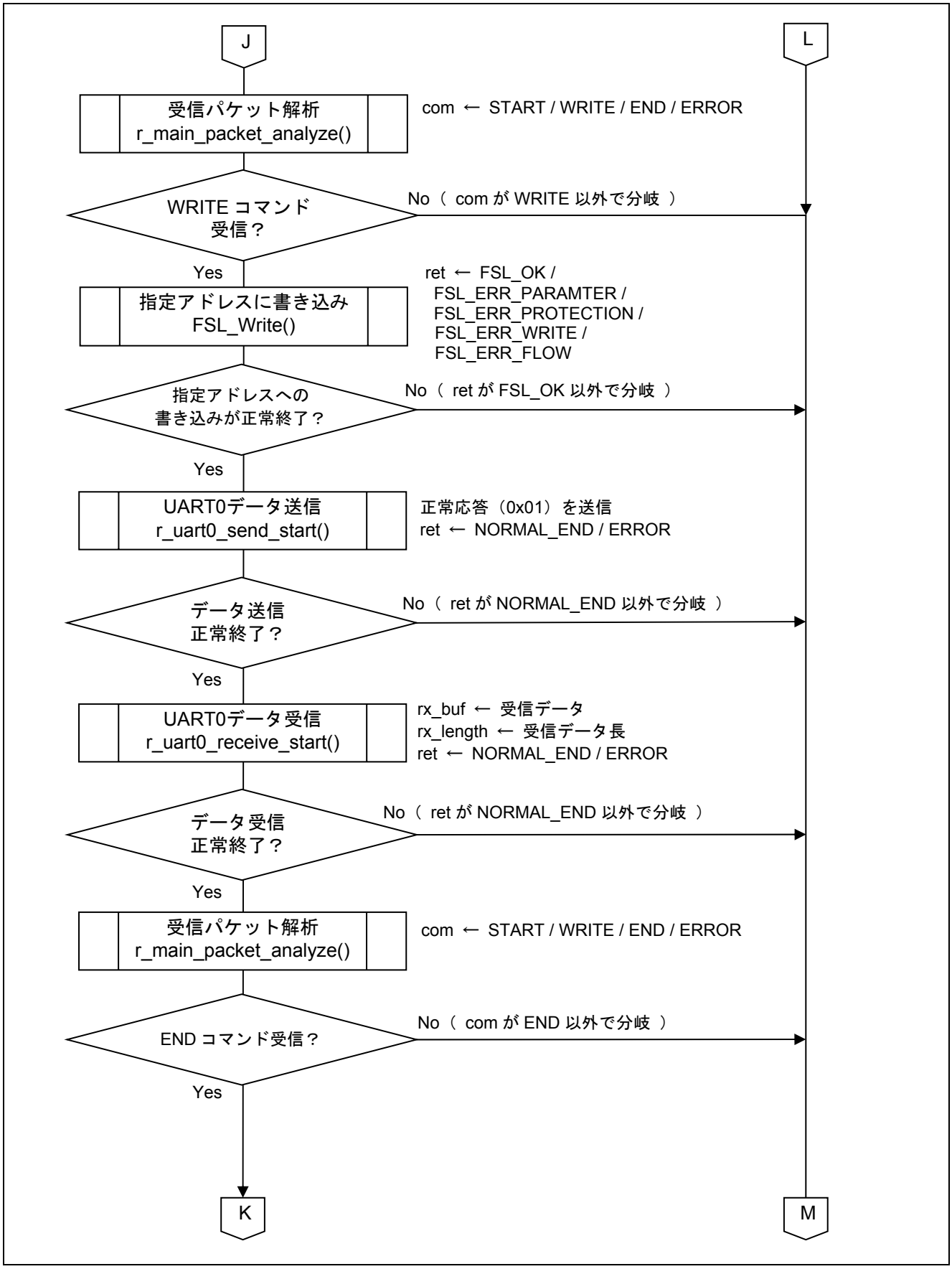

図 5.28 フラッシュ書き換え実行(2/3)

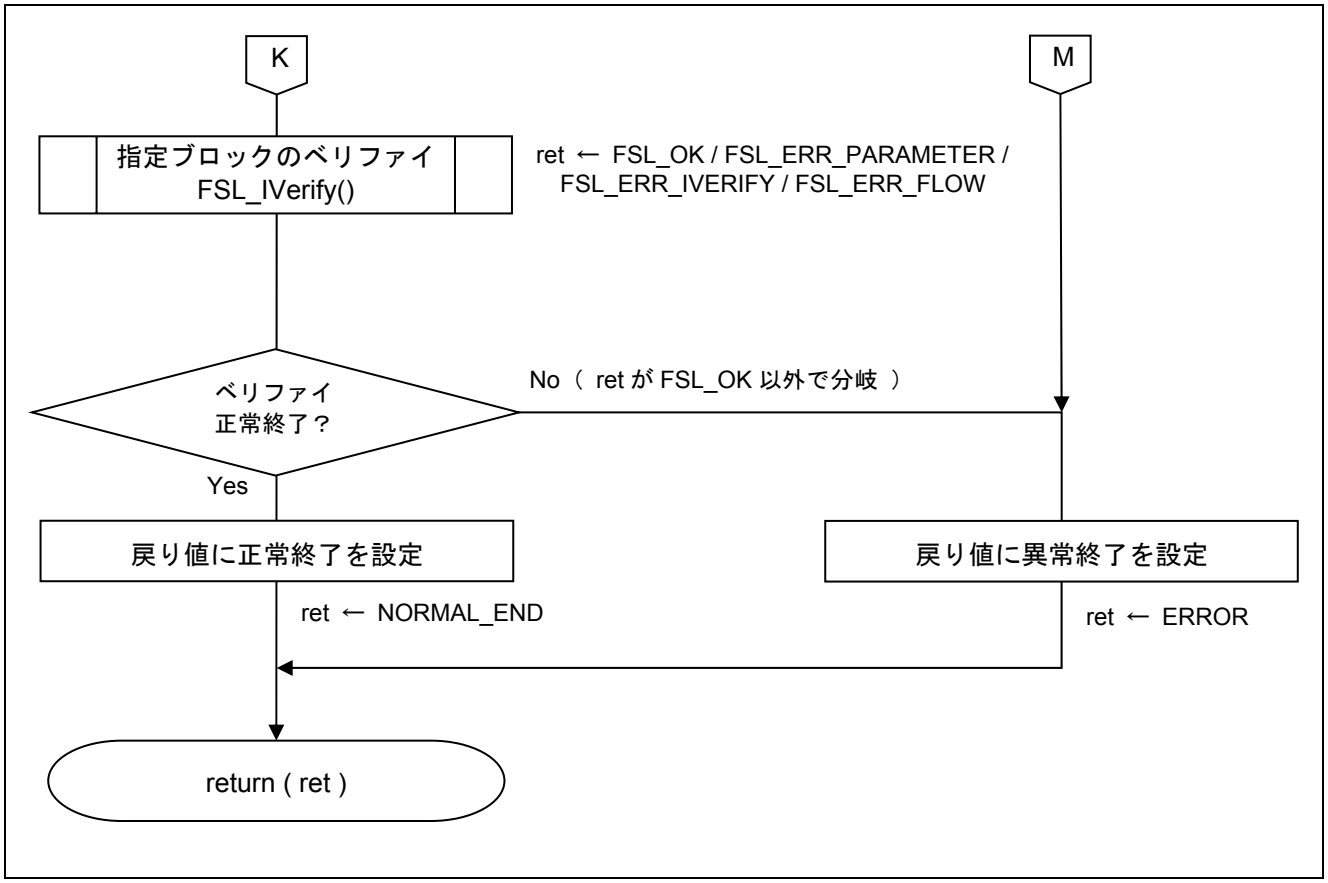

図 5.29 フラッシュ書き換え実行(3/3)

5.10.23 UART0 データ送信

図 5.30 にUART0 データ送信のフローチャートを示します。

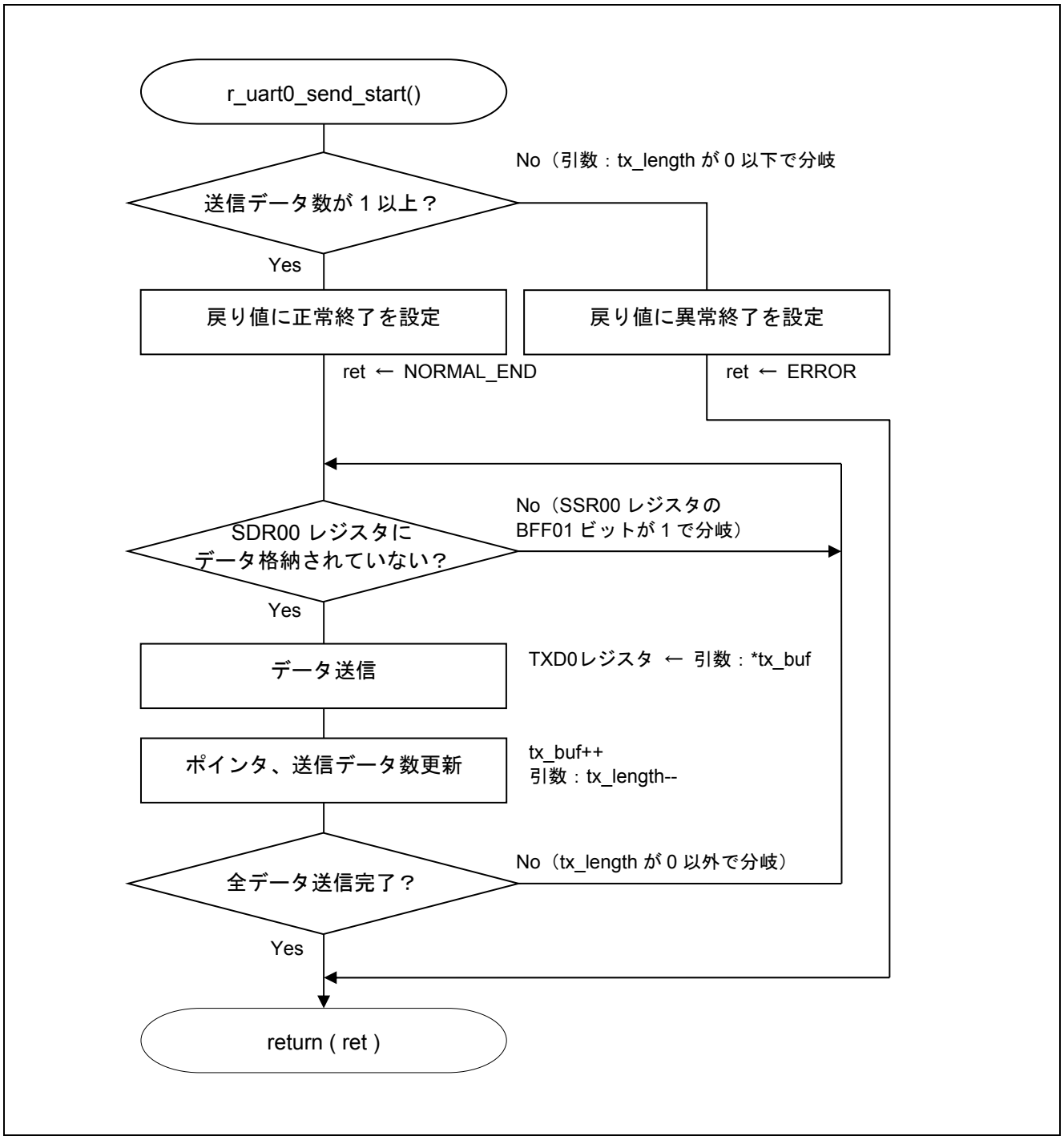

図 5.30 UART0 データ送信

6. 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。

7. 参考ドキュメント

RL78/G12 ユーザーズマニュアル ハードウエア編（R01UH0200J）

RL78 ファミリ ユーザーズマニュアル ソフトウエア編（R01US0015J）

RL78 ファミリ フラッシュ・セルフ・プログラミング・ライブラリ Type01 ユーザーズマニュアル（R01US0050J）

（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。）

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	RL78/G12 セルフ・プログラミング（UART 受信データ） CC-RL

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2015.10.20	—	初版発行
1.10	2016.06.01	8	1.4 フラッシュ・セルフ・プログラミング・ライブラリ取得方法を修正
		53	参考ドキュメントを追加

製品ご使用上の注意事項

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

1. 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　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

■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/