

RL78/G12

R01AN1463JJ0110

Rev. 1.10

2016.06.01

セルフ・プログラミング(IICA 通信方式)

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を説明します。

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	3
1.1	フラッシュ・データ・ライブラリ概要	3
1.2	コード・フラッシュ・メモリについて	4
1.3	フラッシュ・データ・ライブラリのソフトウェア環境	5
1.3.1	フラッシュ書き換え	6
1.3.2	フラッシュ・シールド・ウインドウ	7
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	8
2.	動作確認条件	8
3.	関連アプリケーションノート	9
4.	ハードウェア説明	10
4.1	ハードウェア構成例	10
4.2	使用端子一覧	11
5.	ソフトウェア説明	12
5.1	通信仕様	12
5.1.1	START コマンド	12
5.1.2	WRITE コマンド	12
5.1.3	END コマンド	12
5.1.4	通信シーケンス	13
5.2	動作概要	14
5.3	ファイル構成	17
5.4	オプション・バイトの設定一覧	18
5.5	リンク・ディレクティブ・ファイル	19
5.6	定数一覧	20
5.7	変数一覧	21
5.8	関数(サブルーチン)一覧	22
5.9	関数(サブルーチン)仕様	23
5.10	フローチャート	26
5.10.1	CPU 初期化	27
5.10.2	入出力ポートの初期設定	28
5.10.3	クロックの発生回路の初期設定	29
5.10.4	IICA0 の初期設定	30
5.10.5	TAU0 の初期設定	31
5.10.6	メイン処理	32
5.10.7	LED 点滅開始	34
5.10.8	TAU0 チャンネル 0 動作開始	34
5.10.9	TAU0 チャンネル 0 動作停止	35
5.10.10	TAU0 チャンネル 0 割り込み	35
5.10.11	IICA0 割り込み処理	36
5.10.12	IICA0 データ転送方向チェック	37
5.10.13	IICA0 データ受信	38
5.10.14	IICA0 受信割り込み発生フラグクリア	40
5.10.15	受信データ解析	41
5.10.16	フラッシュ・セルフ・プログラミング実行	42
5.10.17	フラッシュ・セルフ・プログラミング初期設定	43
5.10.18	フラッシュ書き換え実行	45
6.	サンプルコード	47
7.	参考ドキュメント	47

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・インタフェース IICA	シングルマスタ・システムでの IIC スレーブ送受信を行う。(SCLA0 端子と SDAA0 端子を使用)
タイマ・アレイ・ユニット 0 チャンネル 0	LED の点滅間隔生成
ポート入出力	LED の点灯／消灯

1.1 フラッシュ・データ・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ェアです。

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。

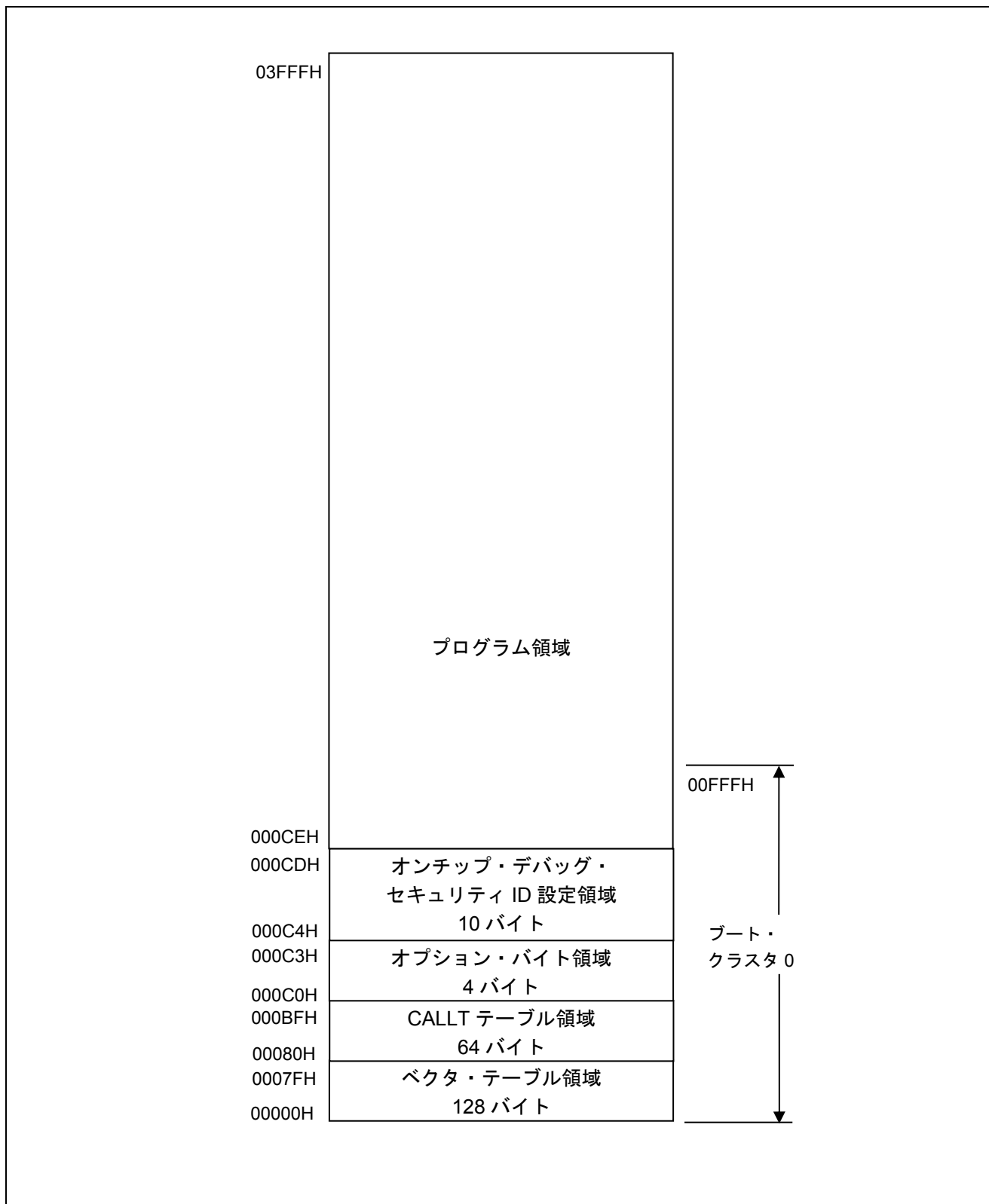


図 1.1 コード・フラッシュ・メモリの構成

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・シールド・ウインドウにより、指定したウインドウ範囲以外 の書き込みおよび消去をフラッシュ・セルフ・プログラミング時のみ 禁止にすることが可能
	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とフラッシュ・シールド・ウインドウ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・データ・ライブラリのソフトウェア環境

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。

フラッシュ・セルフ・プログラミング中は、コード・フラッシュ・メモリを参照できません。そのため、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。

1.3.1 フラッシュ書き換え

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。

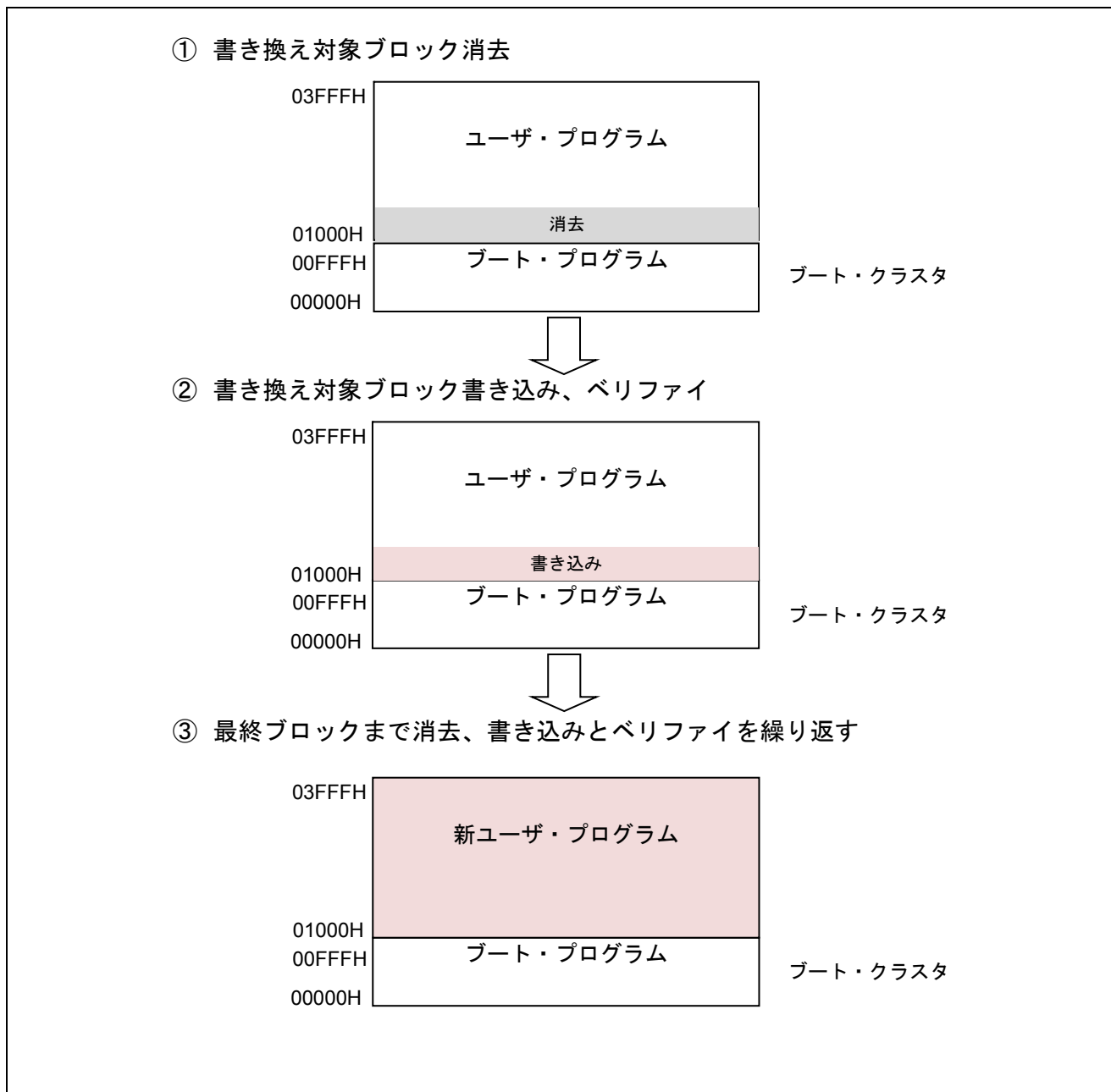


図 1.2 フラッシュ書き換えのイメージ図

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・プログラミングでブート領域を書き換えているときに、電源の遮断、外部要因のリセット発生など書き換えを妨げる要因を発生させないでください。書き換えが妨げられると、ブート領域のデータが壊れて、リセットによるプログラムの再スタートや再書き込みができなくなる場合があります。

1.3.2 フラッシュ・シールド・ウインドウ

フラッシュ・シールド・ウインドウはフラッシュ・セルフ・プログラミング時のセキュリティ機能の一つで、指定してウインドウ範囲以外の書き込み、及び消去をフラッシュ・セルフ・プログラミング時のみ禁止に設定する機能です。

以下に、スタート・ブロックが 04H、エンド・ブロックが 0AH の場合のイメージ図を記載します。

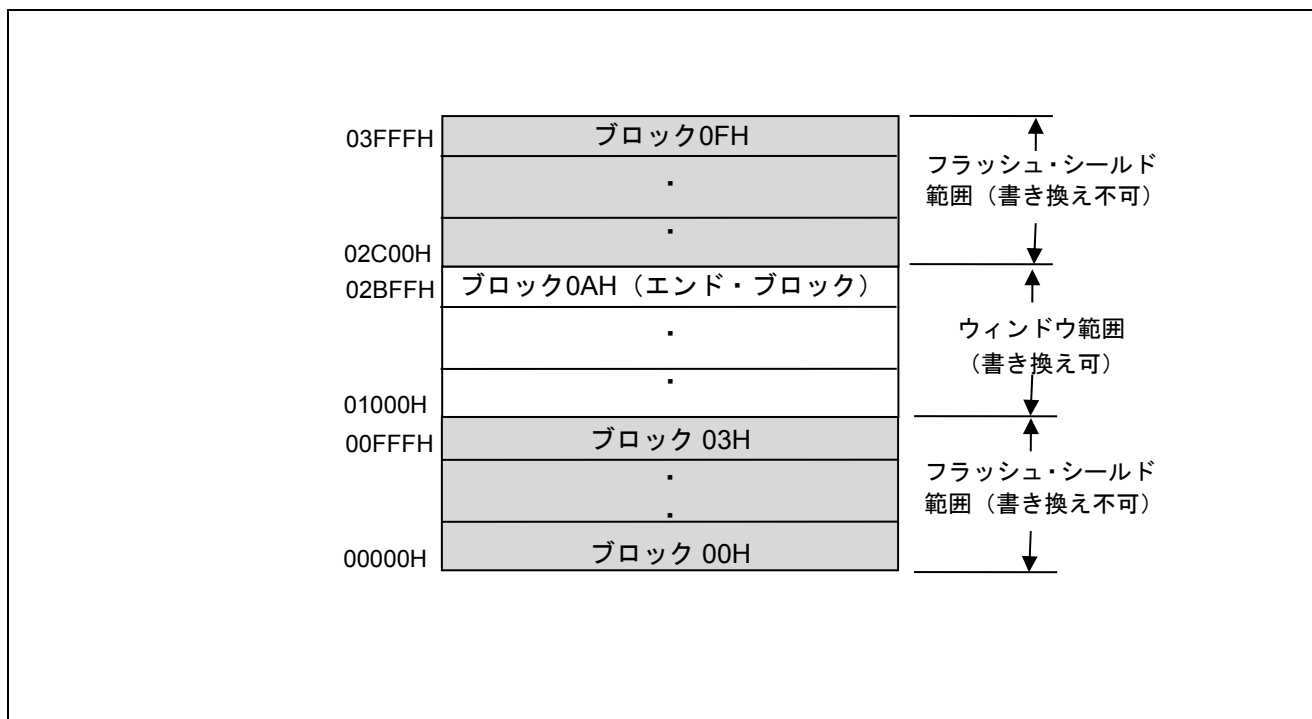


図 1.3 フラッシュ・シールド・ウインドウのイメージ図

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 Workspace 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。

”incl78”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	RL78/G12 (R5F1026A)
動作周波数	<ul style="list-style-type: none"> ● 高速オンチップ・オシレータ・クロック : 24MHz ● CPU/周辺ハードウェア・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	ルネサス エレクトロニクス製 CS+ V3.02.00
アセンブラ	ルネサス エレクトロニクス製 CA78K0R V1.72
使用ボード	RL78/G12 ターゲット・ボード (QB-R5F1026A-TB)
フラッシュ・セルフ・プログラミング・ライブラリ (Type, Ver)	FSLRL78 Type01, Ver2.20 ^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 (R01AN1030J) アプリケーションノート
- RL78 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・ライブラリ Type01 (R01AN0350J) アプリケーションノート
- RL78/G13 フラッシュ・セルフ・プログラミング実行編 (R01AN0718J) アプリケーションノート
- RL78/G12 シリアル・インタフェース IICA (スレーブ送受信) (R01AN1372J) アプリケーションノート
- RL78/G12 シリアル・インタフェース IICA (マスタ送受信) (R01AN1371J) アプリケーションノート

4. ハードウェア説明

4.1 ハードウェア構成例

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ェア構成例を示します。

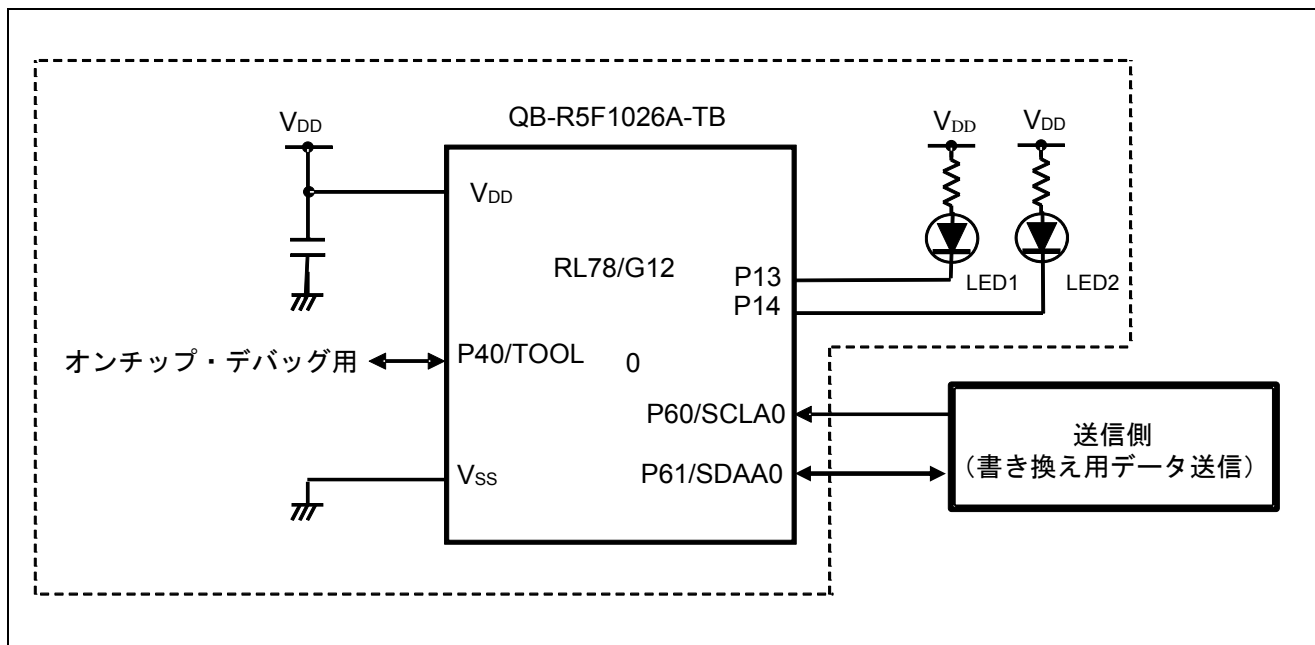


図 4.1 ハードウェア構成例

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 VDD 又は VSS に接続して下さい）。

2 VDD 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
P13	出力	LED1 の点灯／消灯
P14	出力	LED2 の点灯／消灯
P60	入出力	IICA0 のシリアル・クロック入出力端子
P61	入出力	IICA0 のシリアル・データ送受信端子

5. ソフトウェア説明

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、IIC 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 IIC バスのウェイトを解除して次のコマンドを受信します。異常終了の場合には IIC バスをウェイト状態のまま、LED1 と LED2 を点灯して、以降の処理は行いません。以下に IIC 通信設定と、各コマンドの仕様を記載します。

表 5.1 IIC 通信設定

自局アドレス	0xA0
動作モード	標準モード (100 KHz)

5.1.1 START コマンド

START コマンドを受信すると IIC バスをウェイト状態にして、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 IIC バスのウェイトを解除します。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。

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 IIC バスをウェイト状態にして、受信したデータをフラッシュ・メモリへ書き込み、1 ブロックの書き込み毎にベリファイを行います。正常終了すると IIC バスのウェイトを解除します。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。

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 IIC バスをウェイト状態にして、現在書き込んでいるブロックのベリファイを行います。ベリファイが正常終了の場合はブート・フラグを反転後にリセットを発生させ、ブート・スワップを行います。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。

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 IIC バスのウェイトを解除します。

(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 :

IIC バスをウェイト状態にして、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 IIC バスのウェイトを解除します。

(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 :

IIC バスをウェイト状態にして、現在の書き換え対象ブロックのバリファイを行い、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) $\times 10$ [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) シリアル・インタフェース IICA の設定を行います

<設定条件>

- 動作モードを標準モードに設定します。
- 転送クロックを 100kHz に設定します。
- 自局アドレスを 0xA0 に設定します。
- 9 クロック目で割り込みが発生するように設定します。
- ストップ・コンディション割り込み禁止に設定します。
- P60/KR4/SCLA0 端子を転送クロック入出力用に、P61/KR5/SDAA0 端子をデータ送信/受信用に設定します。

(3) TAU0 のチャンネル 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 24MHz、動作クロック 2 (CK02) を 12MHz、動作クロック 3 (CK03) を 93.75kHz 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・トリガはソフトウェア・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) 割り込みを許可します。

- (5) 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。
- (6) 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 のチャンネル 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 のチャンネル 0 の動作を開始します。
- (7) 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
TAU0 のチャンネル 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。
- (8) IICA0 転送完了割り込み要求で HALT モードから通常動作に移行します。
- (9) 割り込みを禁止します。
- (10) 送信側からアドレスと転送方向を受信したら、IIC バスをウェイト状態にして転送方向をチェックします。
- マスタがスレーブにデータを送信する場合、受信完了割り込み要求フラグをクリアして、IIC バスのウェイトを解除します。
 - マスタがスレーブからデータを受信する場合、LED1 と LED2 を点灯し、以降の処理は行いません。
- (11) LED 点滅動作中の場合は、TAU0 のチャンネル 0 の動作を停止して LED 点滅動作を停止します
- (12) 送信側から START コマンド (0x02) を受信したら、IIC バスをウェイト状態にし、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 : 24MHz
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。
 - FSL_GetFlashShieldWindow 関数を呼び出し、フラッシュ・シールド・ウインドウの開始ブロックと終了ブロックを取得します。
 - フラッシュ・シールド・ウインドウの開始ブロックが 0 以外、または終了ブロックが 15 以外の場合は、FSL_SetFlashShieldWindow 関数を呼び出し、フラッシュ・シールド・ウインドウの開始ブロックを 0、終了ブロックを 15 に設定します。
- (13) IIC バスのウェイトを解除し、送信側へ送信許可状態を通知します。

- (14) WRITE コマンド (0x03) と書き込みデータ (4byte) を受信します。
- (15) IIC バスをウェイト状態にし、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。
- (16) FSL_BlankCheck 関数を呼び出し、書き換え対象ブロックが書き込み済みかどうかを確認します。
- (17) 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。
- (18) 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。
- (19) IIC バスのウェイトを解除し、送信側へ送信許可状態を通知します。
- (20) END コマンド (0x04) を受信します。
- (21) 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。
- (22) 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。
- (23) IIC バスのウェイトを解除し、送信側へ送信許可状態を通知します。
- (24) (4)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 にファイル構成を示します。ここでは、フラッシュ・セルフ・プログラミング・ライブラリ (FSLRL78 Type01, Ver2.10) に付属のファイル以外で、プロジェクトへ追加したファイルを記載します。

表 5.2 ファイル構成

ファイル名	概要	備考
r_main.asm	メイン処理	サブルーチン : main SLEDBLINK IINTTM00 IINTIICA0 SCHKDIRIICA0 SRECVIICA0 SCLRIICAFLAG SPACKETANALYZE SFSLEXECUTE SFSLINIT SFSLWRITEEXECUTE
r_init.asm	初期設定処理	サブルーチン : SINIPOINT SINICLK SSTARTIICA0 SINITAU SSTARTINTV0 SSTOPINTV0
opt.asm	オプション・バイト設定	
r_lk_dr	リンク・ディレクティブ・ファイル	-

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	10000101B	オンチップ・デバッグ許可

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。

オプション・バイトの設定は CS+ の「リンク・オプション」タグの「デバイス」パネルの「ユーザ・オプション・バイト値」で指定します。「ユーザ・オプション・バイトを設定する」を「はい(-gb)」に設定してください。

5.5 リンク・ディレクティブ・ファイル

リンク・ディレクティブ・ファイルによって、フラッシュ・セルフ・プログラミングを行う書き換えプログラム、フラッシュ・セルフ・ライブラリはブロック 0~3 (ブート・クラスタ 0) に配置されます。また、フラッシュ・セルフ・ライブラリで使用する RAM 領域を使用しないように設定を行います。

本サンプル・プログラムで使用するリンク・ディレクティブ・ファイルの概要を以下に記載します。

<pre> ***** ; Redefined ROM area ***** ; Define new memory entry for boot cluster 0 MEMORY ROM : (000000H, 003800H) ; Define new memory entry for write-data area MEMORY OCDROM : (003E00H, 000200H) ; Merge XMAIN segment MERGE XMAIN := ROM ; Library(fsl.lib) segment ***** ; Merge FSL_FCD segment MERGE FSL_FCD := ROM ; Merge FSL_FECD segment MERGE FSL_FECD := ROM ; Merge FSL_RCD segment MERGE FSL_RCD := ROM ; Merge FSL_BCD segment MERGE FSL_BCD := ROM ; Merge FSL_BECD segment MERGE FSL_BECD := ROM ***** ; Redefined RAM area ***** ; Define new memory entry for self-RAM MEMORY SELFRAM : (0FF900H, 00038AH) ; Redefined default data segment RAM MEMORY RAM : (0FFD20H, 00100H) ; Define new memory entry for STACK-RAM MEMORY RAM_STK : (0FFC8AH, 00096H) ; Define new memory entry for saddr area MEMORY RAM_SADDR : (0FFE20H, 0001E0H) ; Merge DMAIN segment MERGE DMAIN := RAM ; Merge DSTK segment MERGE DSTK := RAM_STK </pre>	<div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>コード領域の定義 コードをアドレス 0x0000~0x37FF に配置 アドレス 0x3800~0x3BFF はデータ格納領域として、コード配置しない</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>OCD モニター領域の定義</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>コード・セグメント配置アドレスの定義</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>フラッシュ・セルフ・ライブラリを ブート領域に配置</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>フラッシュ・セルフ・ライブラリの使用領域 を標準 RAM 領域として使用しないよう定義</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>標準 RAM 領域の定義</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>スタック RAM 領域の定義</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>フラッシュ・セルフ・ライブラリの使用領域 を標準 RAM 領域として使用しないよう定義</p> </div> <div style="border: 1px solid red; padding: 5px;"> <p>各データ・セグメント配置アドレスの定義</p> </div>
--	---

5.6 定数一覧

表 4.4 にサンプルコードで使用する定数を示します。

表 4.4 サンプルコードで使用する定数

定数名	設定値	内容
BAUDRATE	100	kbps 単位の転送速度 (標準モードの 100kbps)
FHOCO	24	MHz 単位の HOCO 発振周波数 (24MHz)
OPCLK	FHOCO / 2	MHz 単位の IICA0 動作クロック (12MHz)
DIICWL	$(47 * OPCLK + 9) / 10$	IICWL0 レジスタ設定値 (57)
RISETIME	100	信号立ち上がり時間 (100ns)
FALLTIME	100	信号立ち下がり時間 (100ns)
WIDTHHIGH	$5300 - (RISETIME + FALLTIME)$	SCLA0 ハイ・レベル幅 (5100ns)
DIICWH	$(WIDTHHIGH * OPCLK + 999) / 1000$	IICWH0 レジスタ設定値 (62)
CSLFADDR	0xA0	スレーブのアドレス
BUFSIZE	9	データ用バッファのサイズ
NORMAL_END	0x00	正常終了
ERROR	0xFF	異常終了
NO_RECEIVE	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_CMD	0x02	START コマンド
WRITE_CMD	0x03	WRITE コマンド
END_CMD	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・ライブラリ初期化関数の引数: ステータス・チェック・インターナル・モードに設定
START_BLOCK_NUM	0x00	フラッシュ・シールド・ウインドウの開始ブロック番号
END_BLOCK_NUM	0x0F	フラッシュ・シールド・ウインドウの終了ブロック番号
BLOCK_SIZE	0x400	コード・フラッシュの 1 ブロックのサイズ (1024byte)
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
WRITEBLOCK	WRITEADDR / 400H	書き込み開始ブロックアドレス

注 対象アドレスを変更する場合は 0x3800~0x3BFC の範囲内で変更してください。

5.7 変数一覧

表 5.5 に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
8 ビット×9 配列	RRCVBUF	データ用バッファ	main SRECVIICA0 SPACKETANALYZE
8 ビット×2 配列	RLEN	データ長受信用バッファ	SRECVIICA0
16 ビット	RRCVLG	受信データサイズ	SRECVIICA0 SPACKETANALYZE
8 ビット	RINTIICFLAG	IICA 受信割り込みフラグ	main, IINTIICA0, SCLRIICAFLAG
8 ビット	RRECVSTATUS	IICA 受信ステータス	SRECVIICA0
8 ビット	RLENCOUNT	データ長カウンタ	SRECVIICA0
8 ビット	RDATACOUNT	データカウンタ	SRECVIICA0
8 ビット	RRECVDATA	1Byte 分の受信データ	SRECVIICA0
8 ビット	RREADAVE	コード・フラッシュのアドレス 0x3BFC ～ 0x3BFF の値の平均	main, SLEDBLINK
8 ビット×8 配列	RARG	FSL サブルーチン引数用配列	SFSLINIT, SFSLWRITEEXECUTE

5.8 関数(サブルーチン)一覧

表 5.6 に関数(サブルーチン)一覧を示します。

表 5.6 関数(サブルーチン)一覧

関数名	概要
SINIPOINT	ポートの初期設定を行います。
SSTARTIICA0	IICA0 の初期設定を行い、通信待機状態にします。
SINITAU	TAU の初期設定を行います。
SLEDBLINK	LED 点滅処理を行います。
SSTARTINTV0	TAU0 のタイマ・カウントを開始します。
SSTOPINTV0	TAU0 のタイマ・カウントを停止します。
IINTTM00	TAU0 の割り込み処理を行います。
IINTIICA0	INTIICA0 の割り込み処理ルーチンです。
SCHKDIRIICA0	IICA0 通信の向きをチェックします
SRECVIICA0	IICA0 のデータ受信を行います
SCLRIICAFLAG	IICA0 受信完了割り込み発生フラグ、IICA0 受信エラー割り込み発生フラグのクリアを行います。
SPACKETANALYZE	受信データの解析を行います。
SFSLEXECUTE	フラッシュ・セルフ・プログラミングを実行します。
SFSLINIT	フラッシュ・セルフ・プログラミングの初期化を行います。
SFSLWRITEEXECUTE	フラッシュ書き換え処理を実行します。

5.9 関数(サブルーチン)仕様

サンプルコードの関数(サブルーチン)仕様を示します。

[関数名] SINIPORT

概要	入出力ポートの初期設定
説明	P13 と P14 を出力設定にします。
引数	なし
リターン値	なし
備考	なし

[関数名] SSTARTIICA0

概要	IICA0 の初期設定
説明	IICA0 を標準モードのスレーブ・アドレス 0xA0 で初期設定を行います。
引数	なし
リターン値	なし
備考	なし

[関数名] IINTIICA0

概要	IICA0 の割り込み処理
説明	INTIICA0 を受け付けて処理するルーチンです。実際の通信処理はサブルーチン SINTIICA0 で処理します
引数	なし
リターン値	なし
備考	なし

[関数名] SCHKDIRIICA0

概要	IICA0 データ転送方向チェック
説明	データの転送方向のチェックをします。
引数	なし
リターン値	C レジスタ ● 正常終了 : NORMAL_END ● 転送方向エラー(転送方向がスレーブ→マスタ) : ERROR
備考	なし

[関数名] SRECVIICA0

概要	IICA0 データ受信
説明	受信データを受信バッファ RRCVBUF に格納し、受信データ長[byte]を RRCVLG に格納します。
引数	なし
リターン値	C レジスタ ● 正常終了 : NORMAL_END ● 異常終了 : ERROR
備考	なし

[関数名] SCLRICAFLAG

概要	IICA0 受信割り込み発生フラグクリア
説明	IICA0 受信完了割り込み発生フラグ RINTIICFLAG 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] SPACKETANALYZE

概要	受信データ解析
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	なし
リターン値	C レジスタ <ul style="list-style-type: none"> ● START コマンド受信 : START_CMD ● WRITE コマンド受信 : WRITE_CMD ● END コマンド受信 : END_CMD ● 異常終了 : ERROR
備考	なし

[関数名] SINITAU

概要	TAU0 の初期設定
説明	TAU の初期設定を行います。
引数	なし
リターン値	なし
備考	なし

[関数名] IINTTM00

概要	TAU0 チャンネル 0 割り込み
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] SSTARTINTV0

概要	TAU0 チャンネル 0 の動作開始
説明	TAU0 チャンネル 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] SSTOPINTV0

概要	TAU0 チャンネル 0 の動作停止
説明	TAU0 チャンネル 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] SLEDBLINK

概要	LED 点滅開始
説明	LED1 の点滅間隔を RREADAVE × 10[ms]に設定し、LED1 の点滅動作を開始します。
引数	なし
リターン値	なし
備考	なし

[関数名] SFSLEXECUTE

概要	フラッシュ・セルフ・プログラミング実行
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	C レジスタ <ul style="list-style-type: none"> ● 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] SFSLINIT

概要	フラッシュ・セルフ・プログラミング初期設定
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	C レジスタ <ul style="list-style-type: none"> ● 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] SFSLWRITEEXECUTE

概要	フラッシュ書き換え実行
説明	コード・フラッシュのデータを書き換えます。
引数	なし
リターン値	C レジスタ <ul style="list-style-type: none"> ● 正常終了 : NORMAL_END ● 異常終了 : ERROR
備考	なし

5.10 フローチャート

図 4.1 にサンプルコードの全体フローを示します。

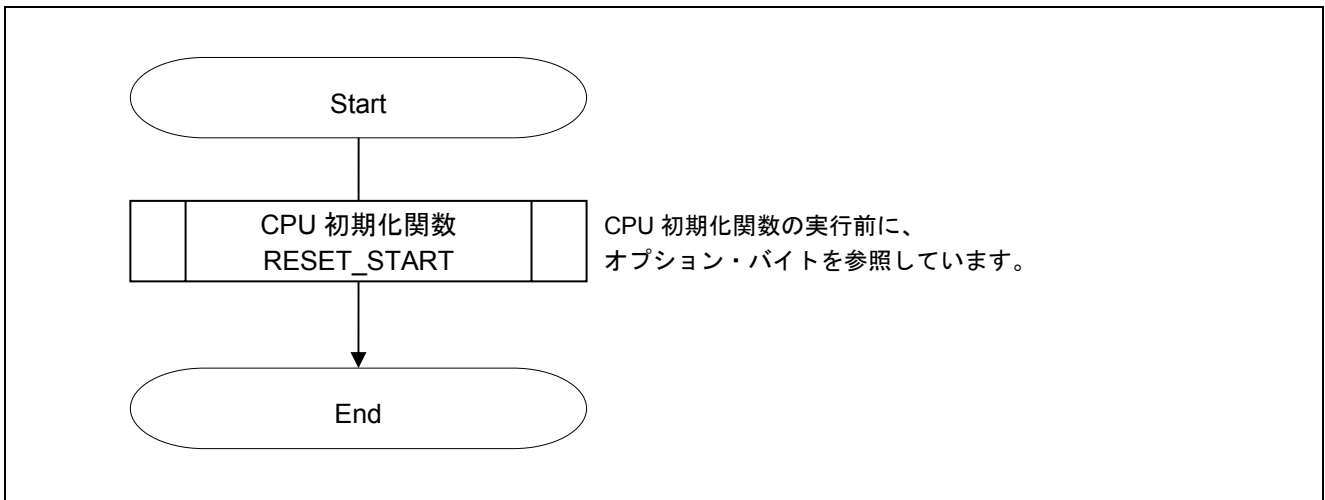


図 4.1 全体フロー

5.10.1 CPU 初期化

図 4.2 に CPU 初期化のフローチャートを示します。

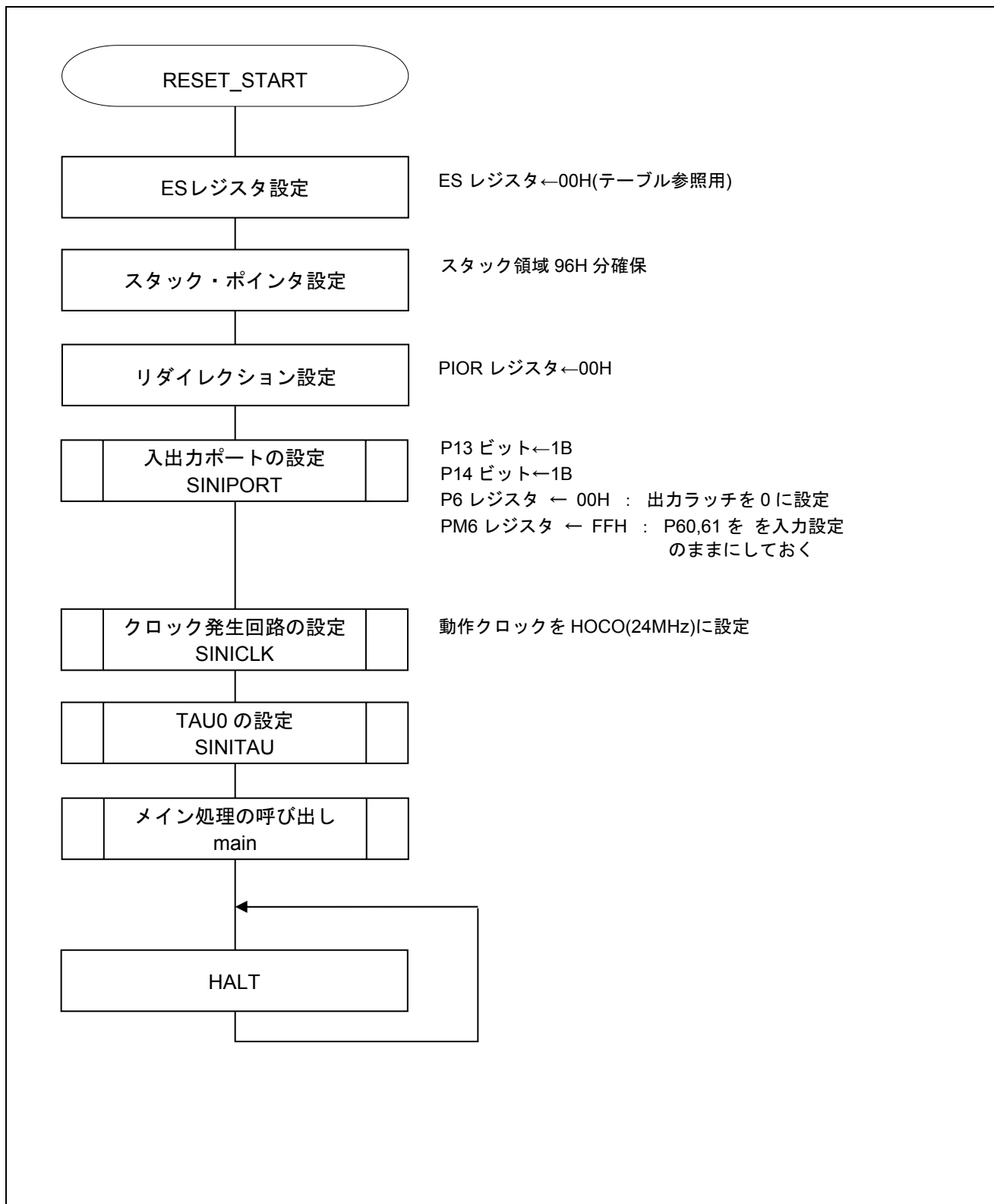


図 4.2 CPU 初期化

5.10.2 入出力ポートの初期設定

図 4.3 に入出力ポートの初期設定フローチャートを示します。

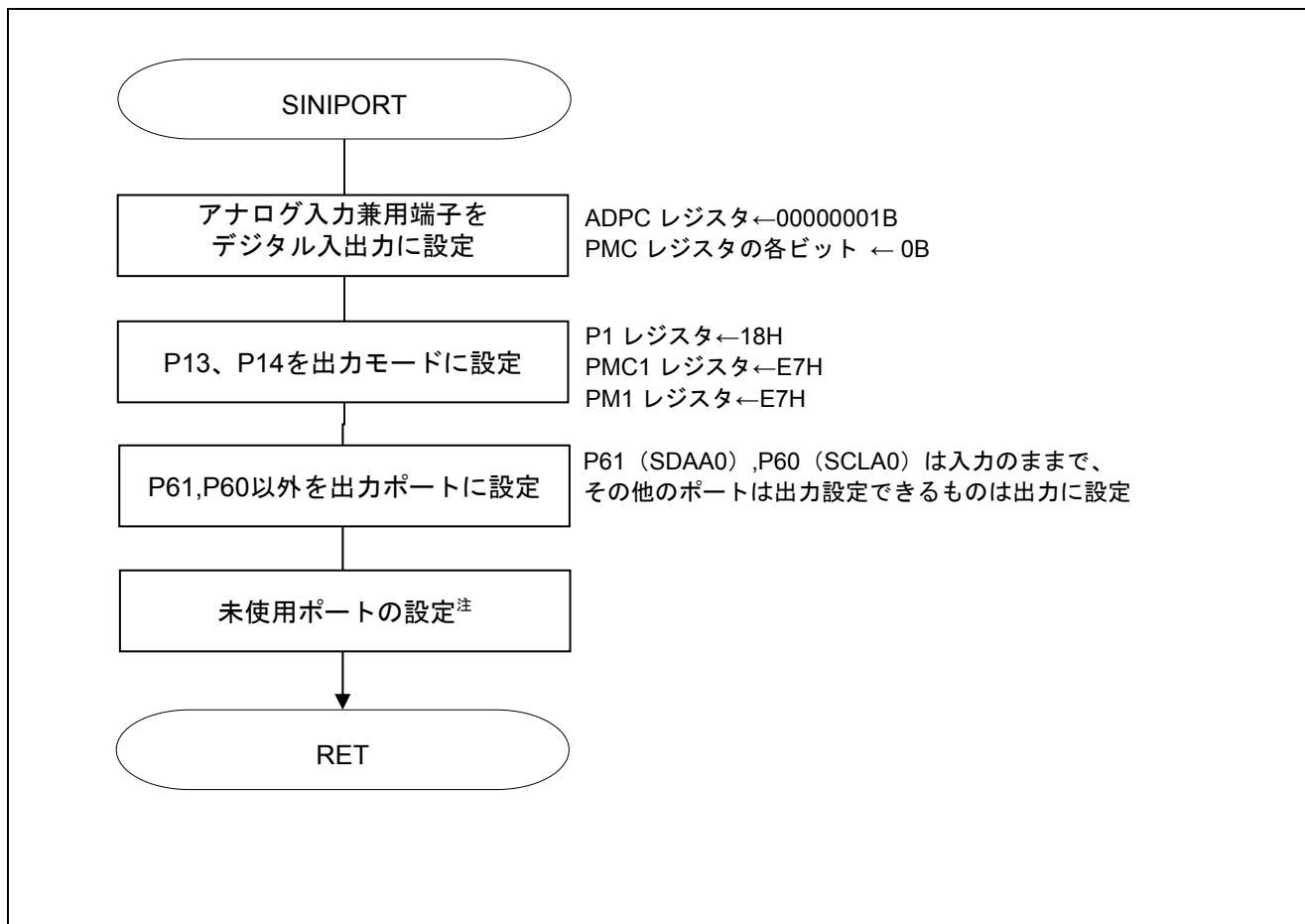


図 4.3 入出力ポートの初期設定

注 未使用ポートの設定については、RL78/G12 初期設定 (R01AN1030J) 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。

5.10.3 クロックの発生回路の初期設定

図 4.4 にクロック発生回路の初期設定のフローチャートを示します。

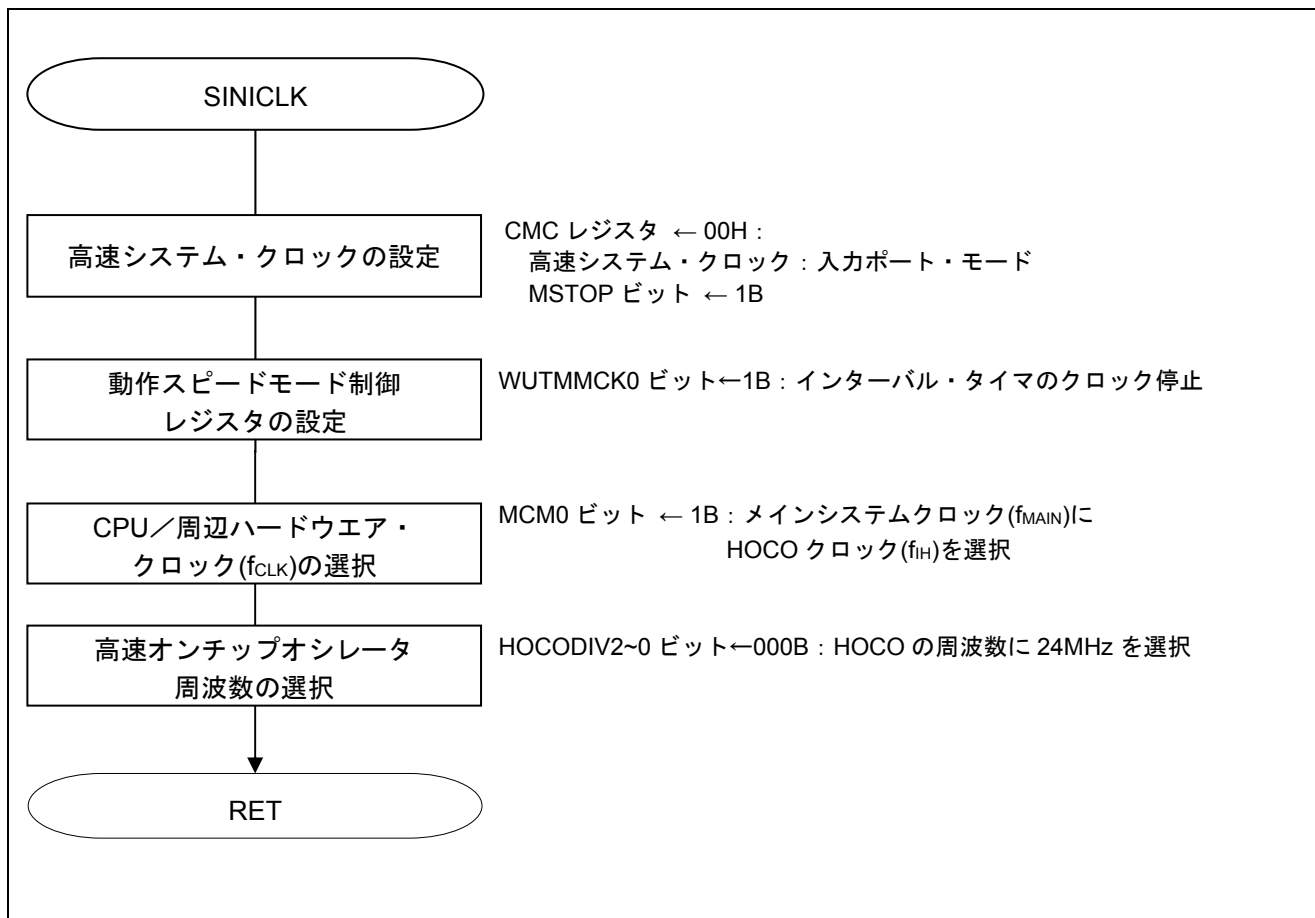


図 4.4 クロック発生回路の初期設定

注意 CPU クロックの設定 (SINICKL) については、RL78/G12 初期設定 (R01AN1030J) 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。

5.10.4 IICA0 の初期設定

図 4.5 に IICA0 の初期設定のフローチャートを示します。

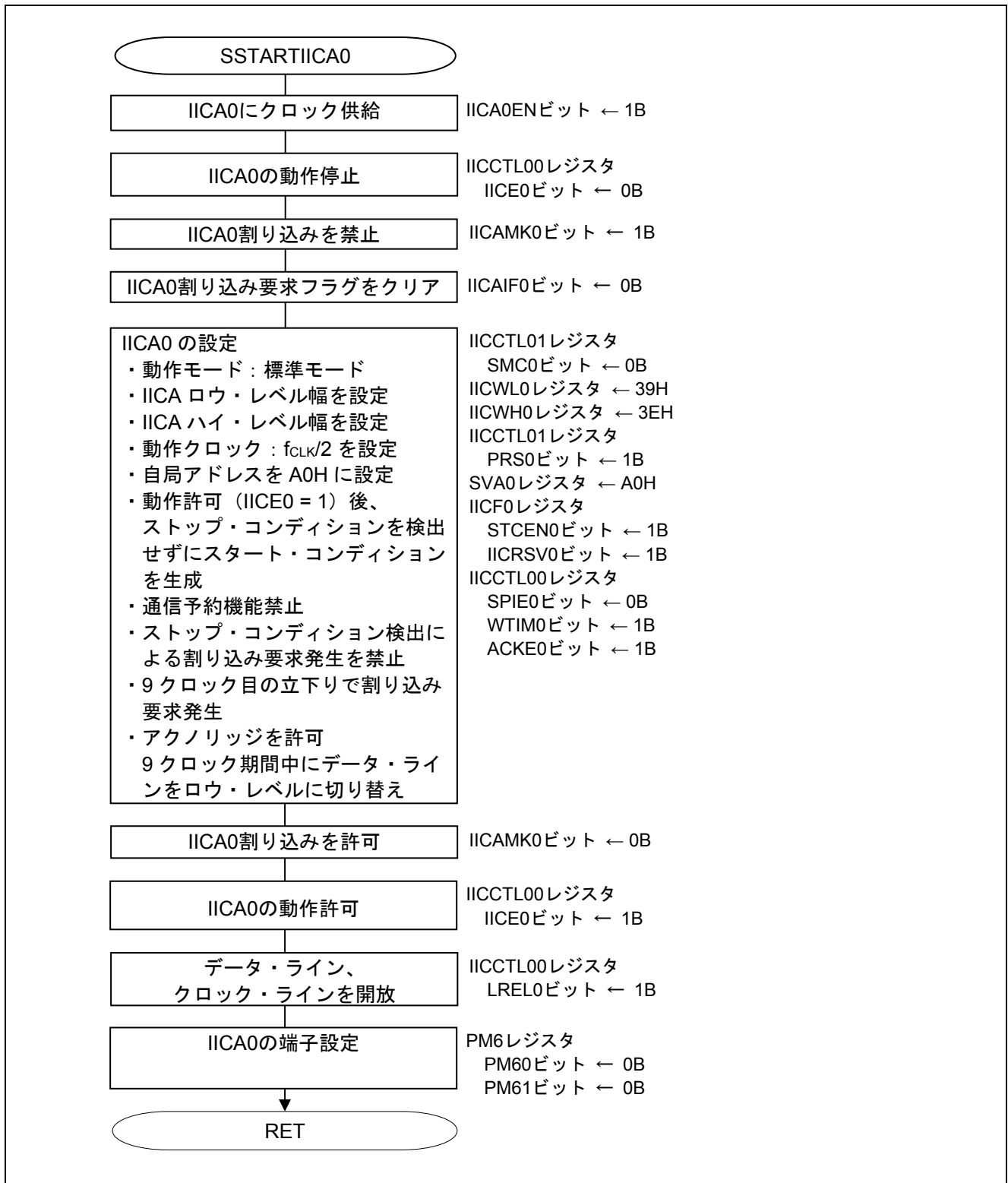


図 4.5 IICA0 の初期設定

5.10.5 TAU0 の初期設定

図 4.6 に TAU0 の初期設定のフローチャートを示します。

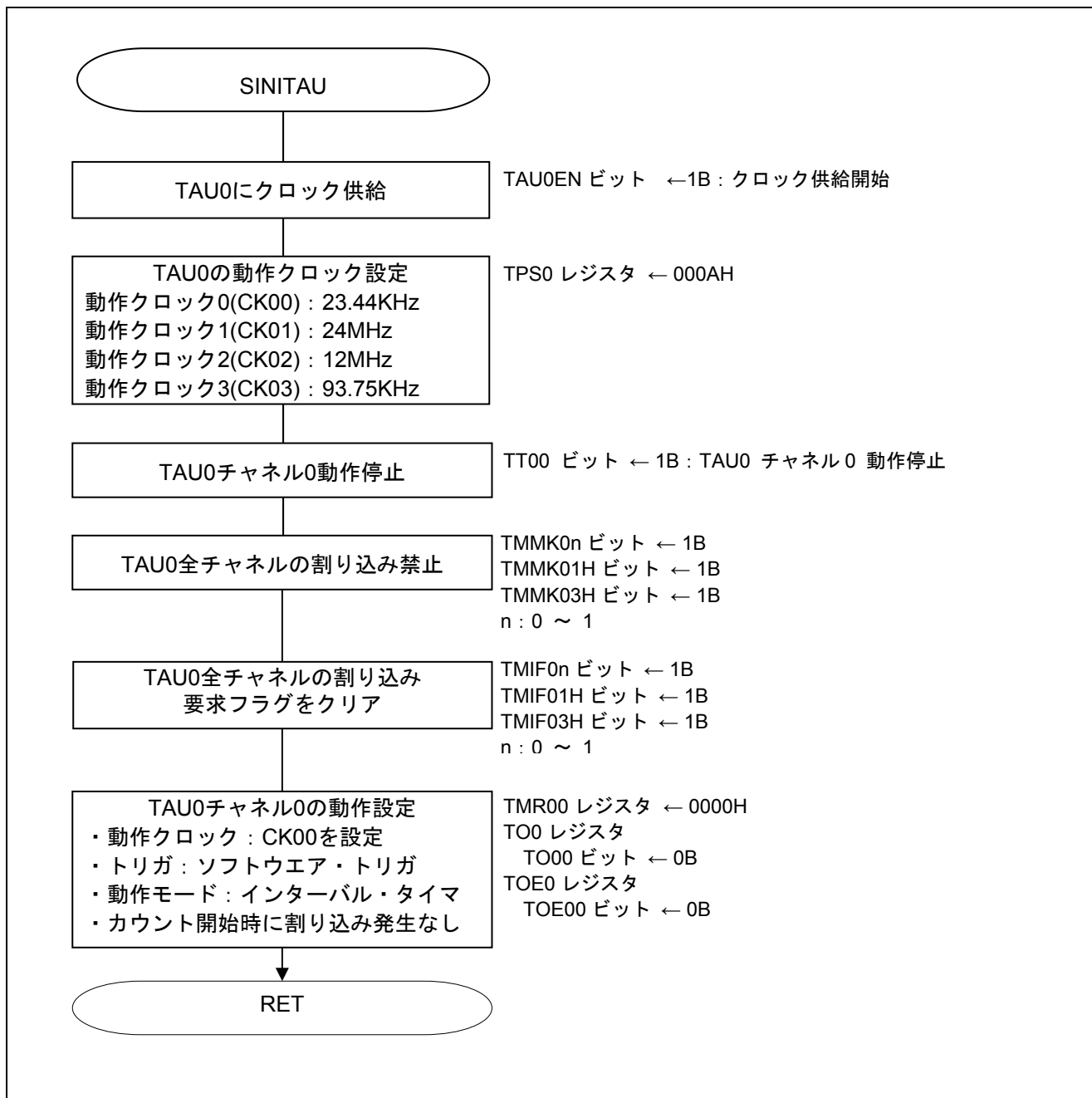


図 4.6 TAU0 の初期設定

5.10.6 メイン処理

図 4.7 にメイン処理(1/2)、図 4.8 にメイン処理(2/2) のフローチャートを示します。

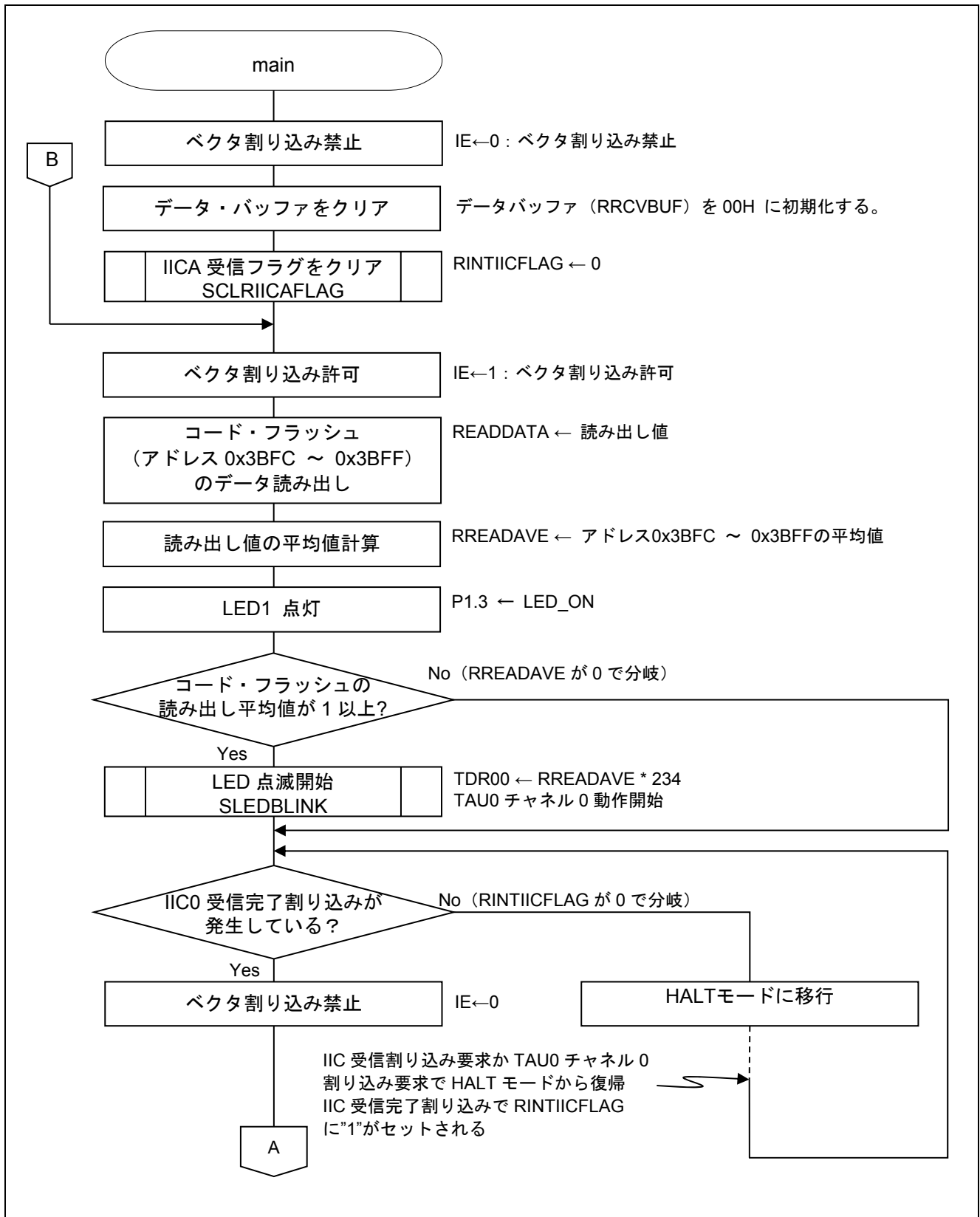


図 4.7 メイン処理(1/2)

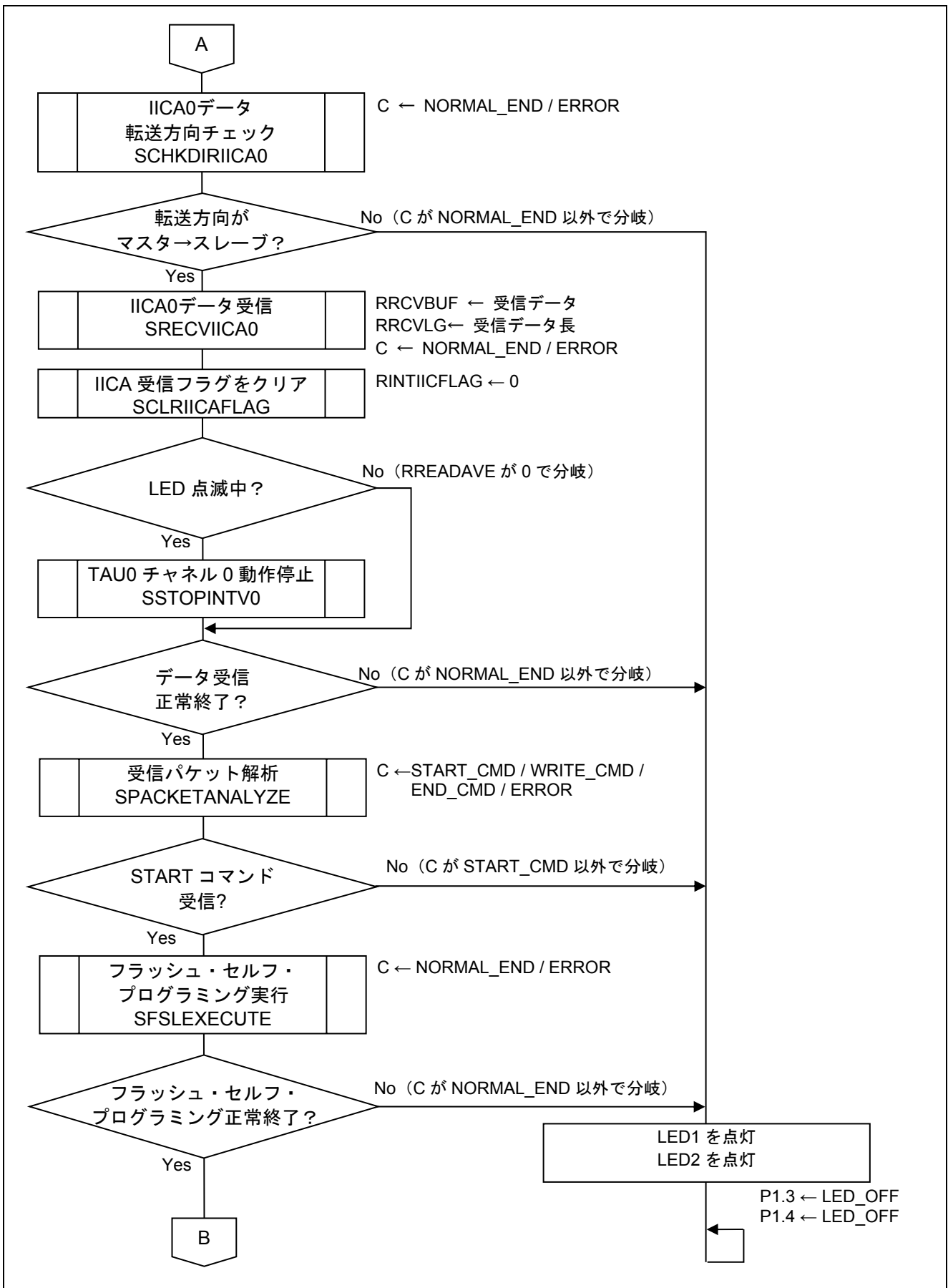


図 4.8 メイン処理(2/2)

5.10.7 LED 点滅開始

図 4.9 に LED 点滅開始のフローチャートを示します。

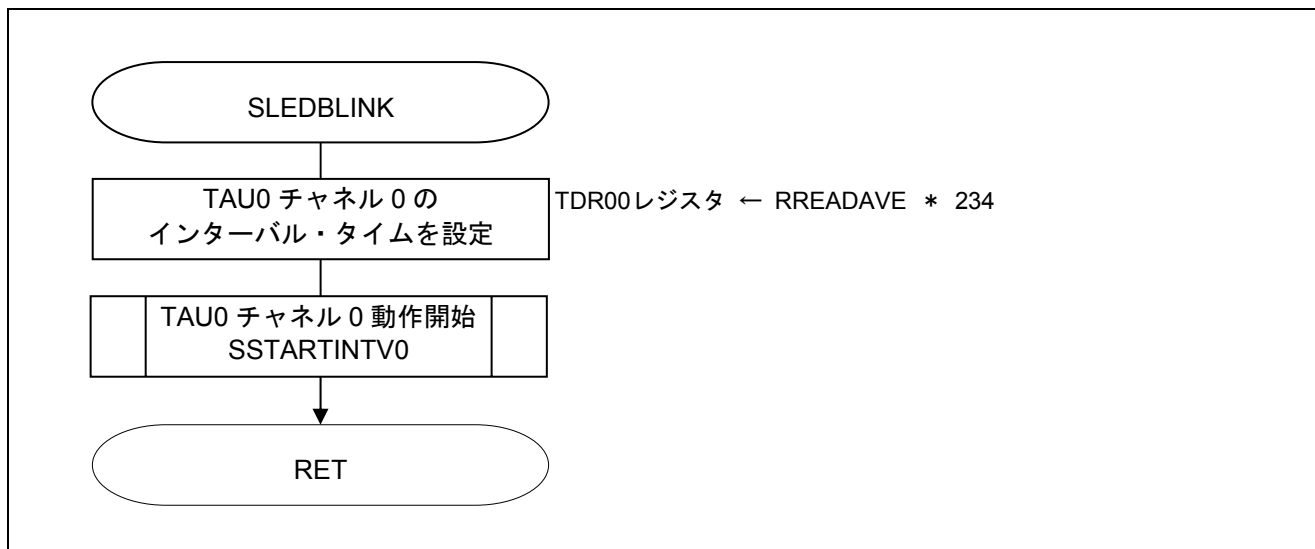


図 4.9 LED 点滅開始

5.10.8 TAU0 チャンネル 0 動作開始

図 4.10 に TAU0 チャンネル 0 動作開始のフローチャートを示します。

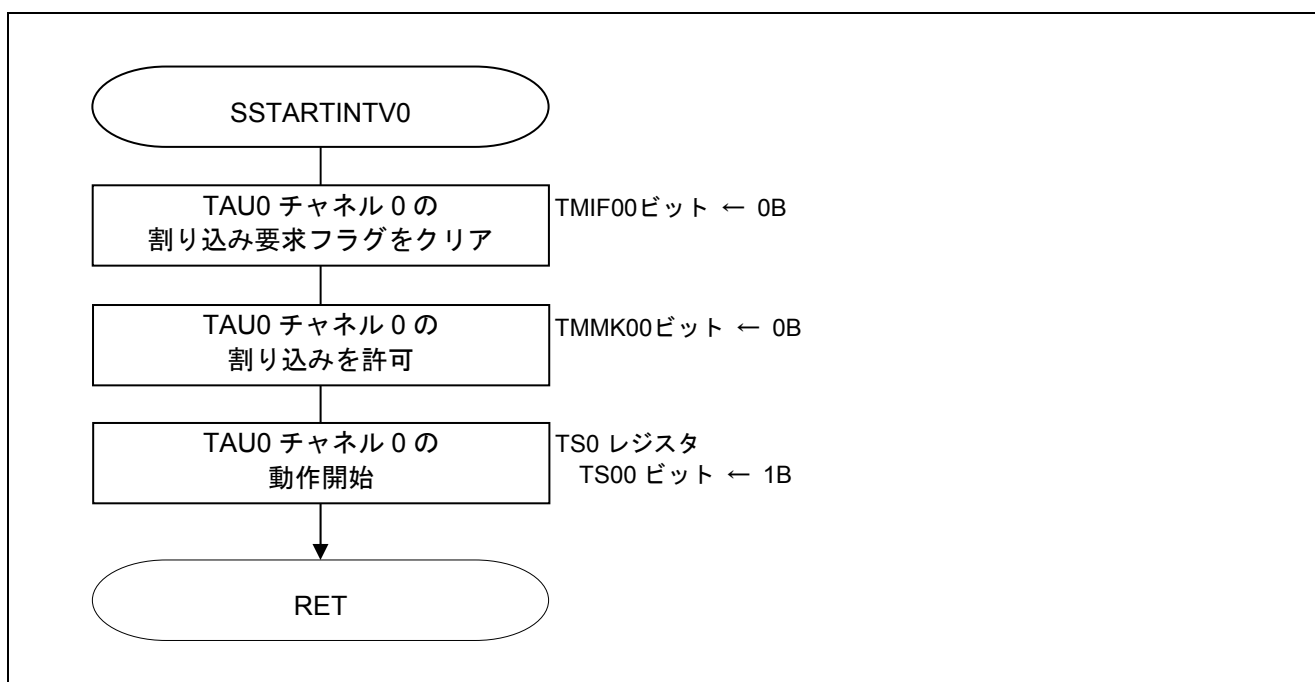


図 4.10 TAU0 チャンネル 0 動作開始

5.10.9 TAU0 チャンネル 0 動作停止

図 4.11 に TAU0 チャンネル 0 動作停止のフローチャートを示します。

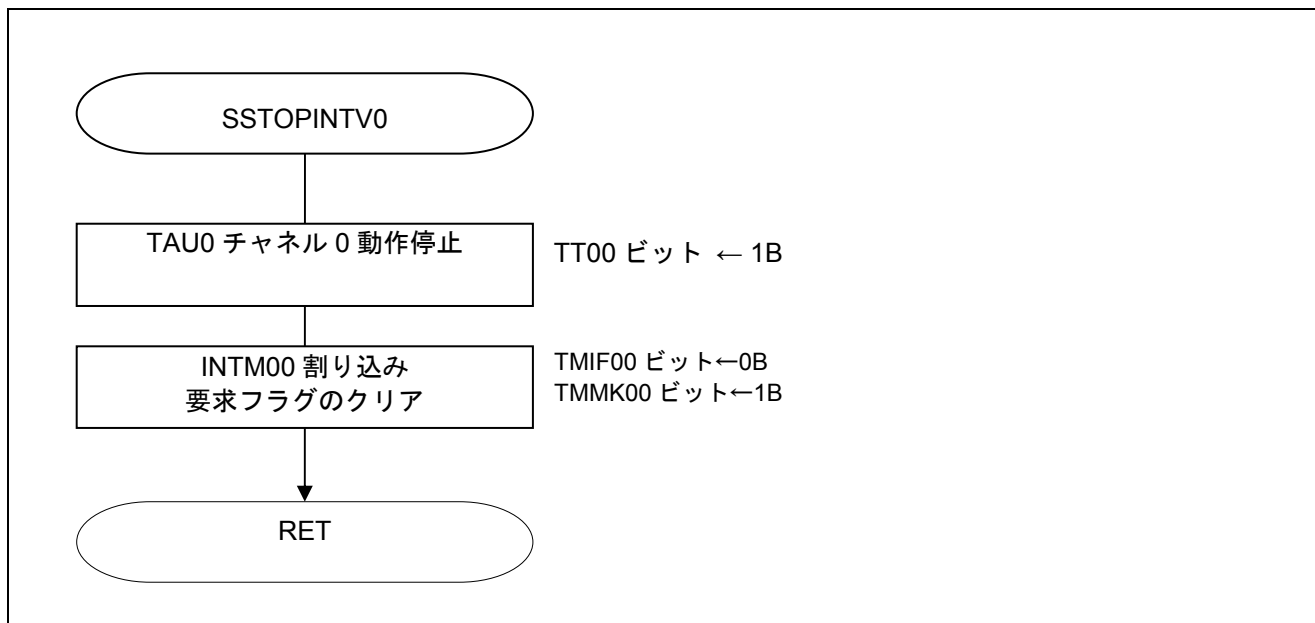


図 4.11 TAU0 チャンネル 0 動作停止

5.10.10 TAU0 チャンネル 0 割り込み

図 4.12 に TAU0 チャンネル 0 割り込みのフローチャートを示します。

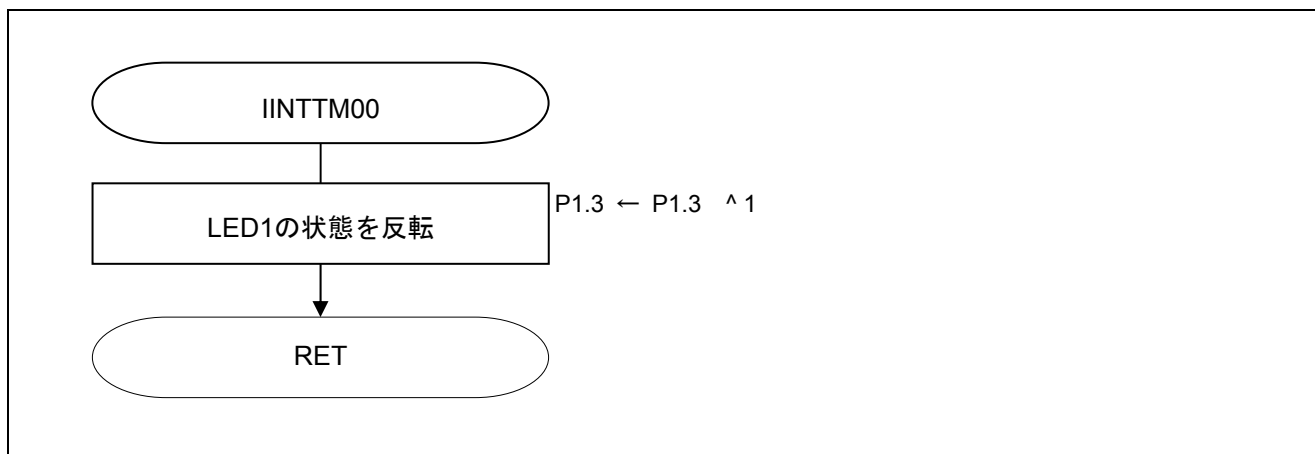


図 4.12 TAU0 チャンネル 0 割り込み

5.10.11 IICA0 割り込み処理

図 4.13 に IICA0 割り込み処理のフローチャートを示します。

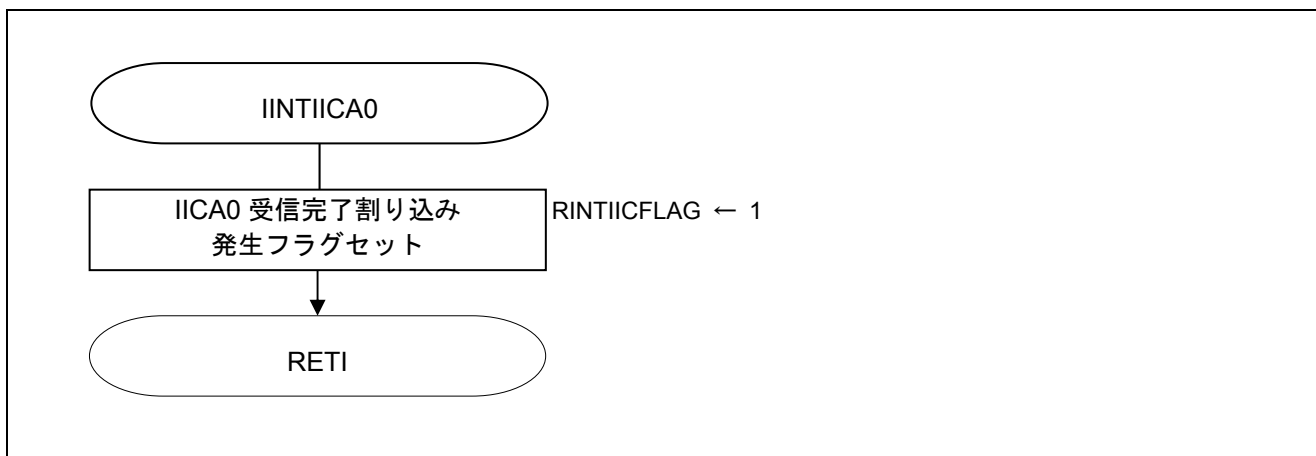


図 4.13 IICA0 割り込み処理

5.10.12 IICA0 データ転送方向チェック

図 4.14 に IICA0 データ転送方向チェックのフローチャートを示します。

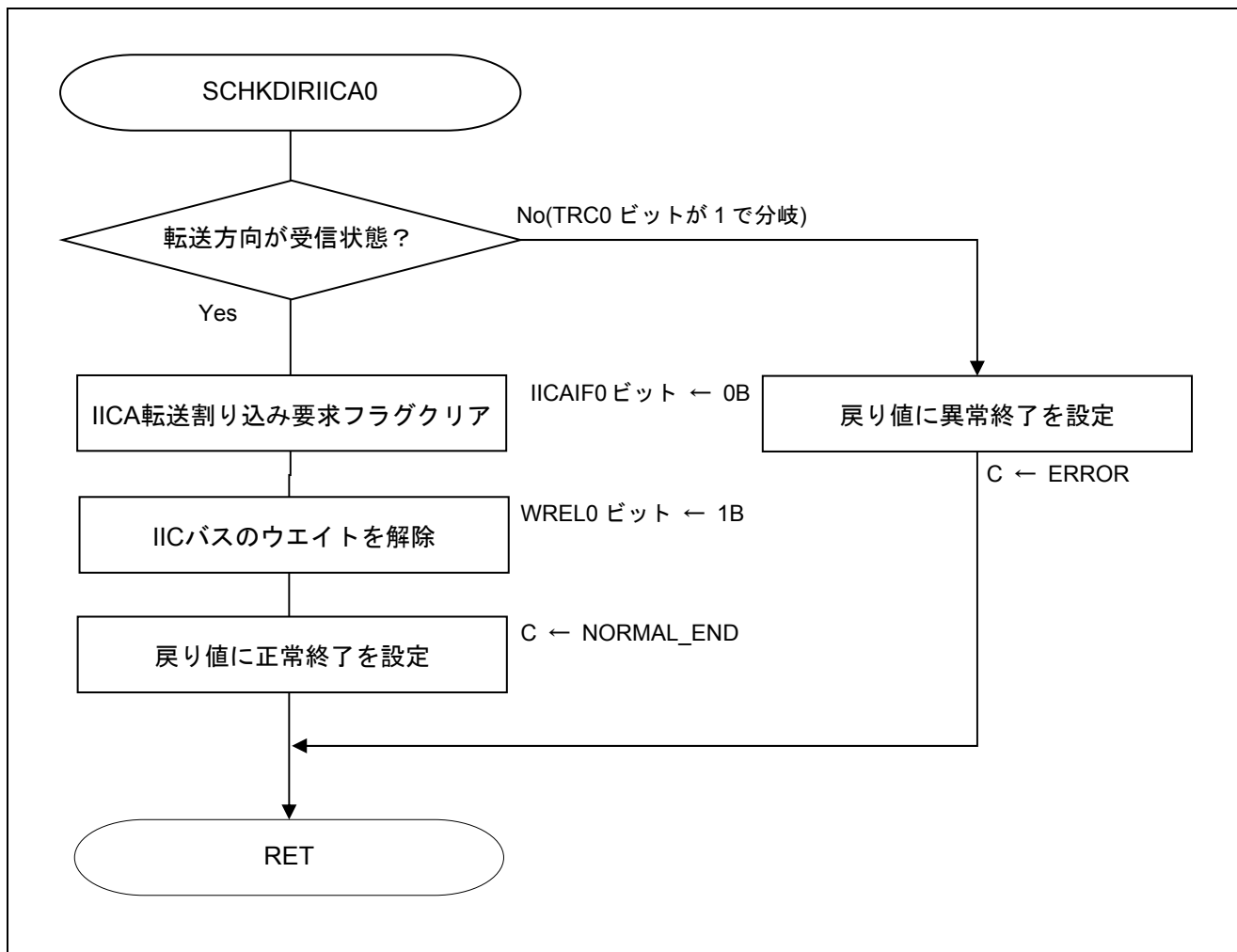


図 4.14 IICA0 データ転送方向チェック

5.10.13 IICA0 データ受信

図 4.15 に IICA0 データ受信 (1/2)を、図 4.16 に IICA0 データ受信(2/2)を示します。

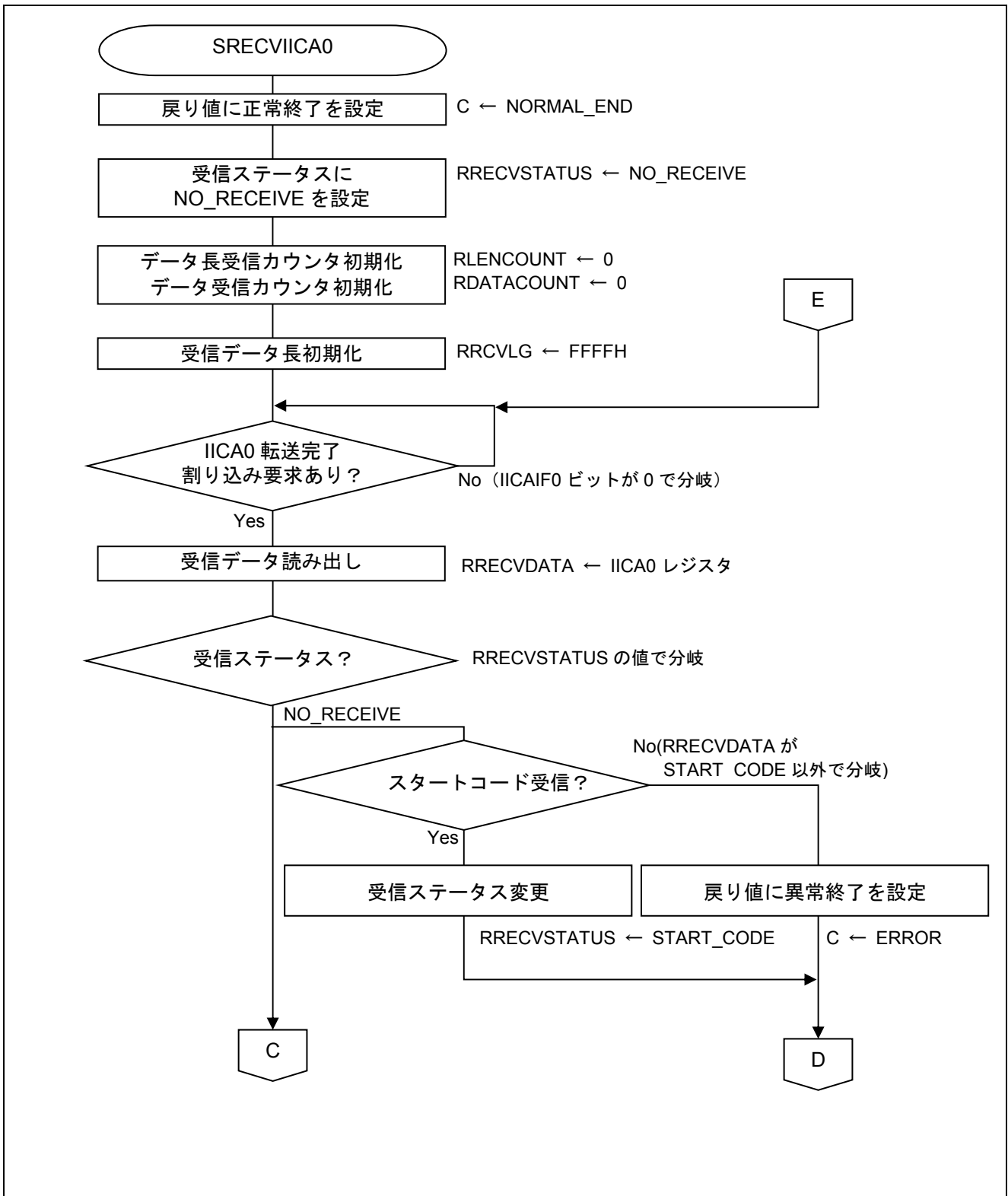


図 4.15 IICA0 データ受信(1/2)

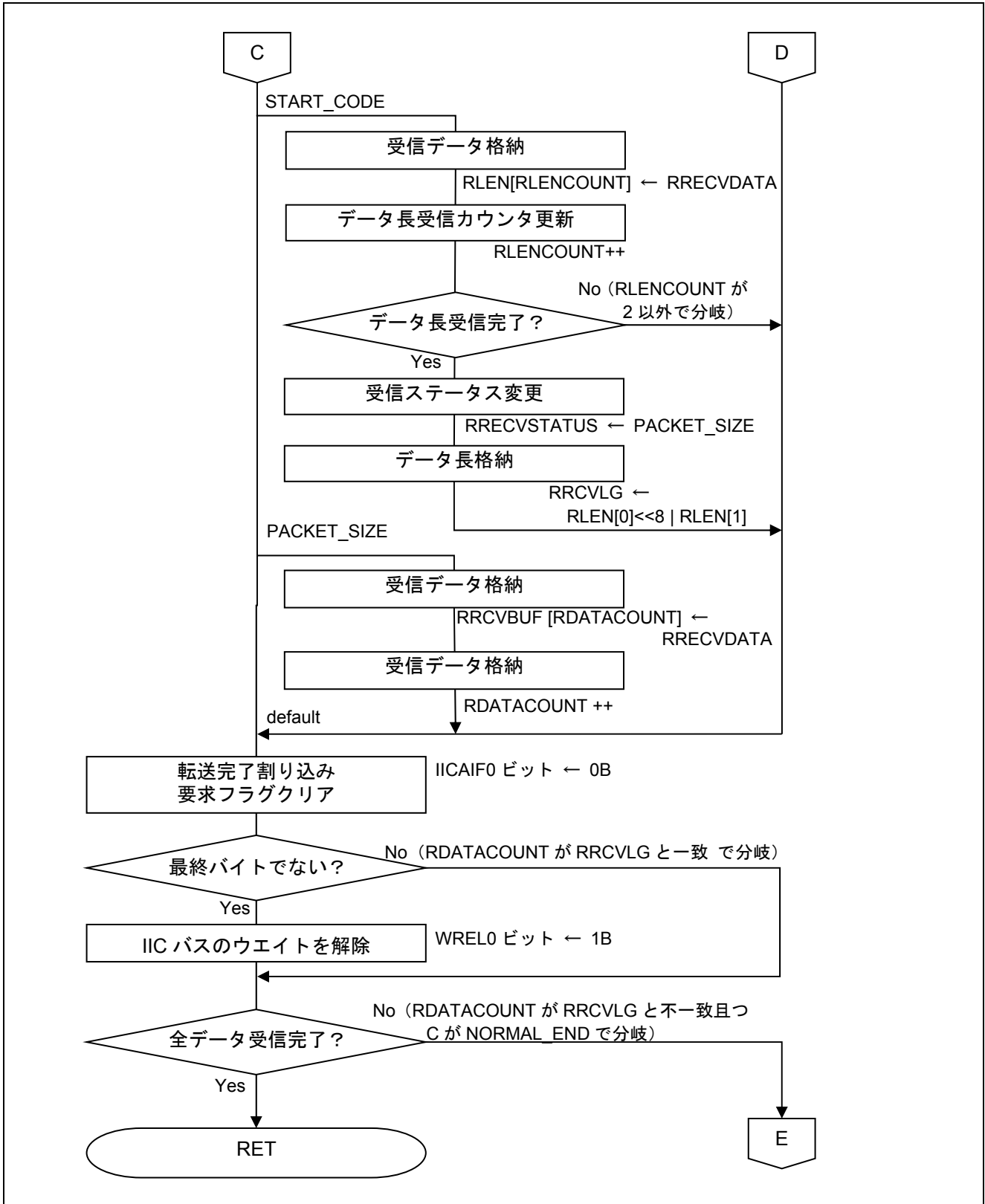


図 4.16 IICA0 データ受信(2/2)

5.10.14 IICA0 受信割り込み発生フラグクリア

図 4.17 に IICA0 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。

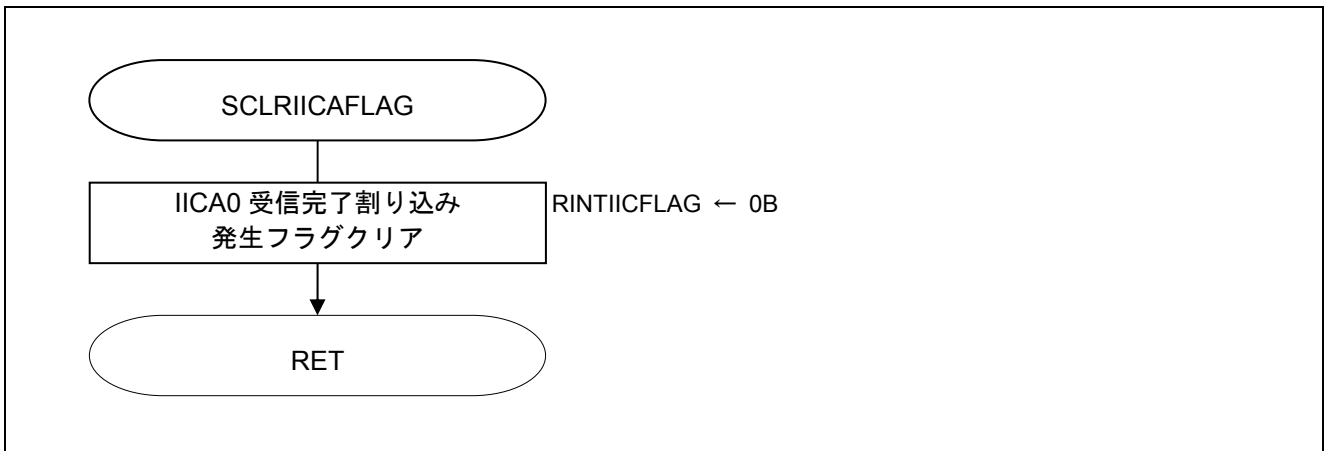


図 4.17 IICA0 受信割り込み発生フラグクリア

5.10.15 受信データ解析

図 4.18 受信データ解析のフローチャートを示します。

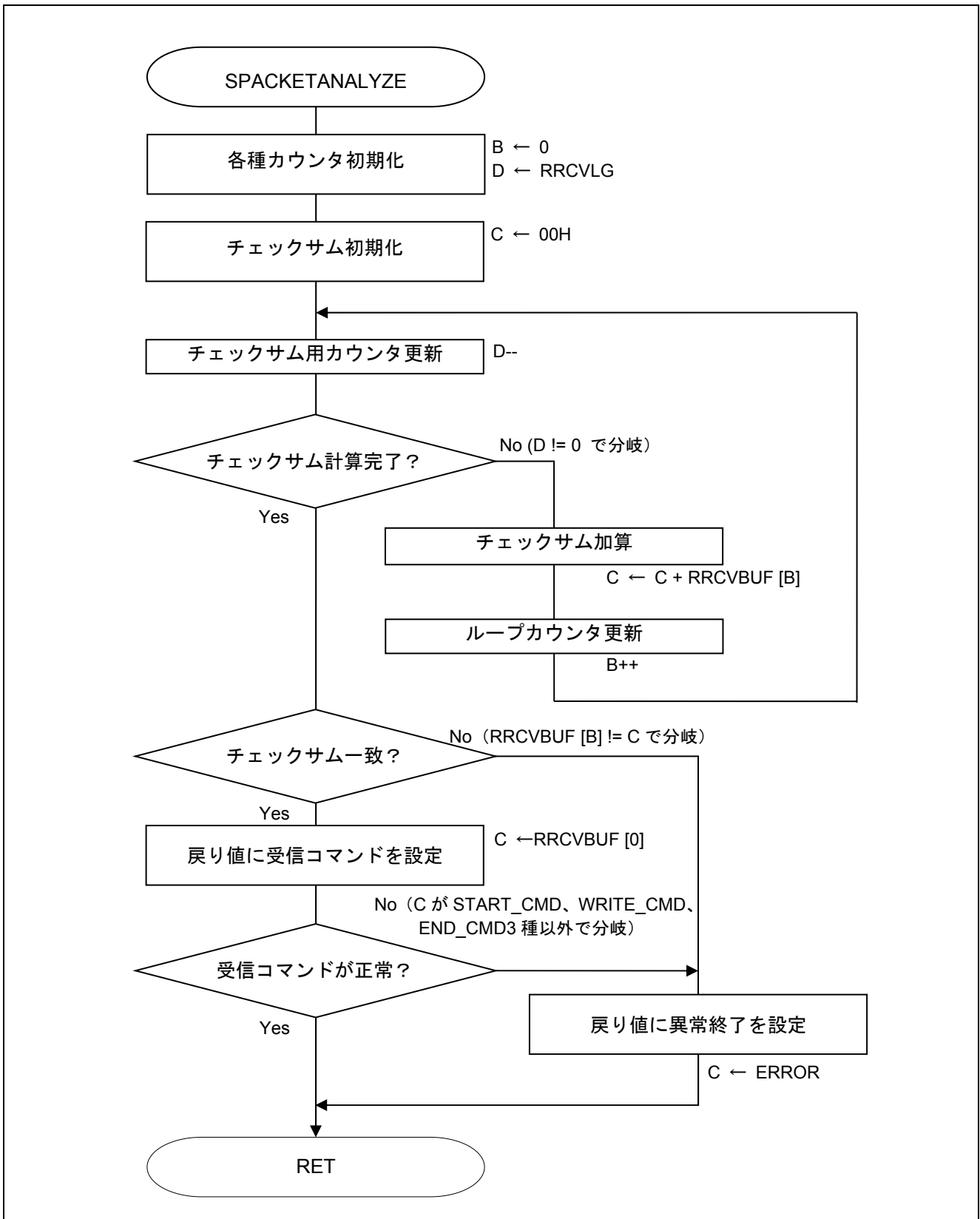


図 4.18 受信データ解析

5.10.16 フラッシュ・セルフ・プログラミング実行

図 4.19 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。

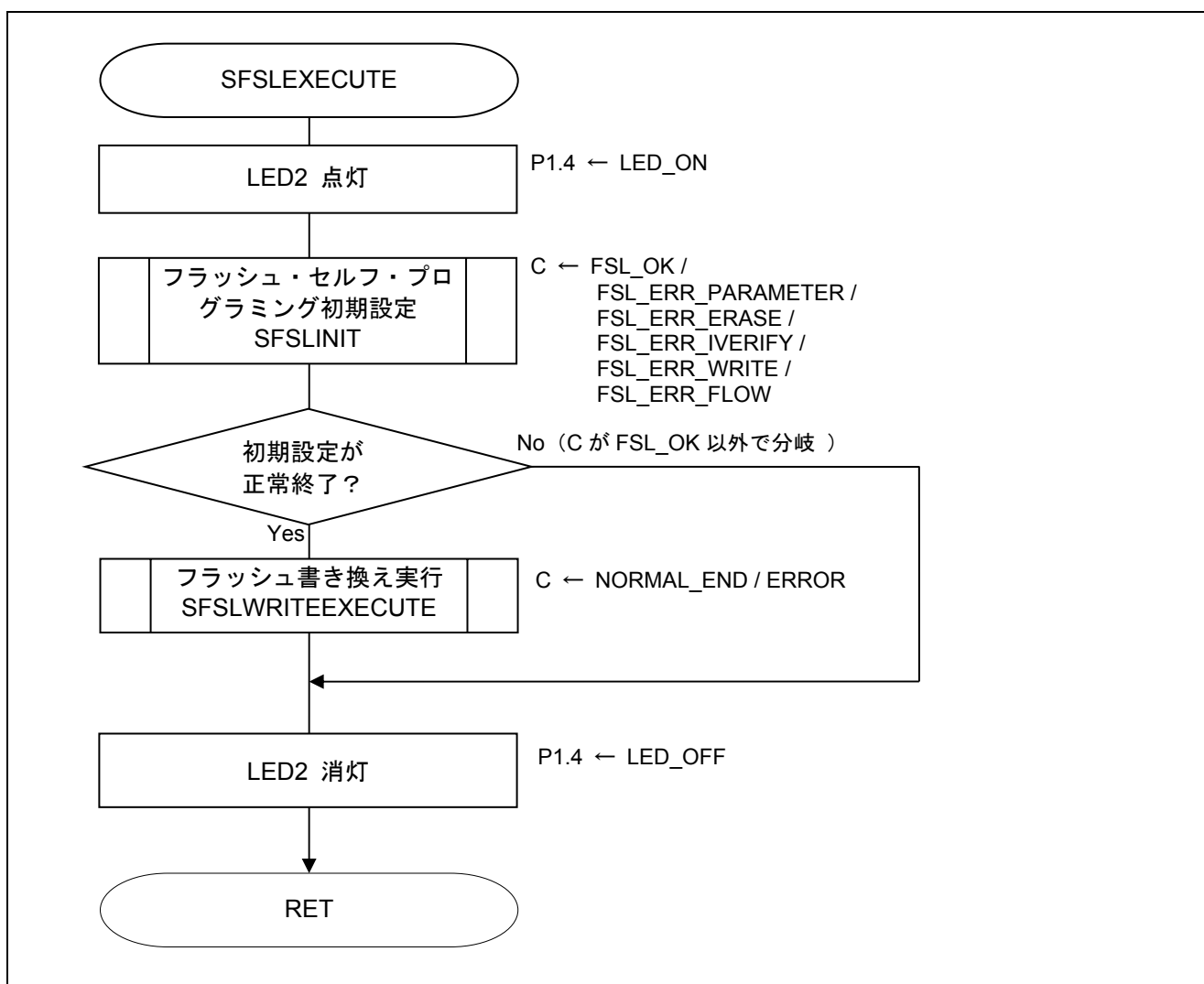


図 4.19 フラッシュ・セルフ・プログラミング実行

5.10.17 フラッシュ・セルフ・プログラミング初期設定

図 4.20 にフラッシュ・セルフ・プログラミング初期設定(1/2)、図 4.21 にフラッシュ・セルフ・プログラミング初期設定(2/2)のフローチャートを示します。

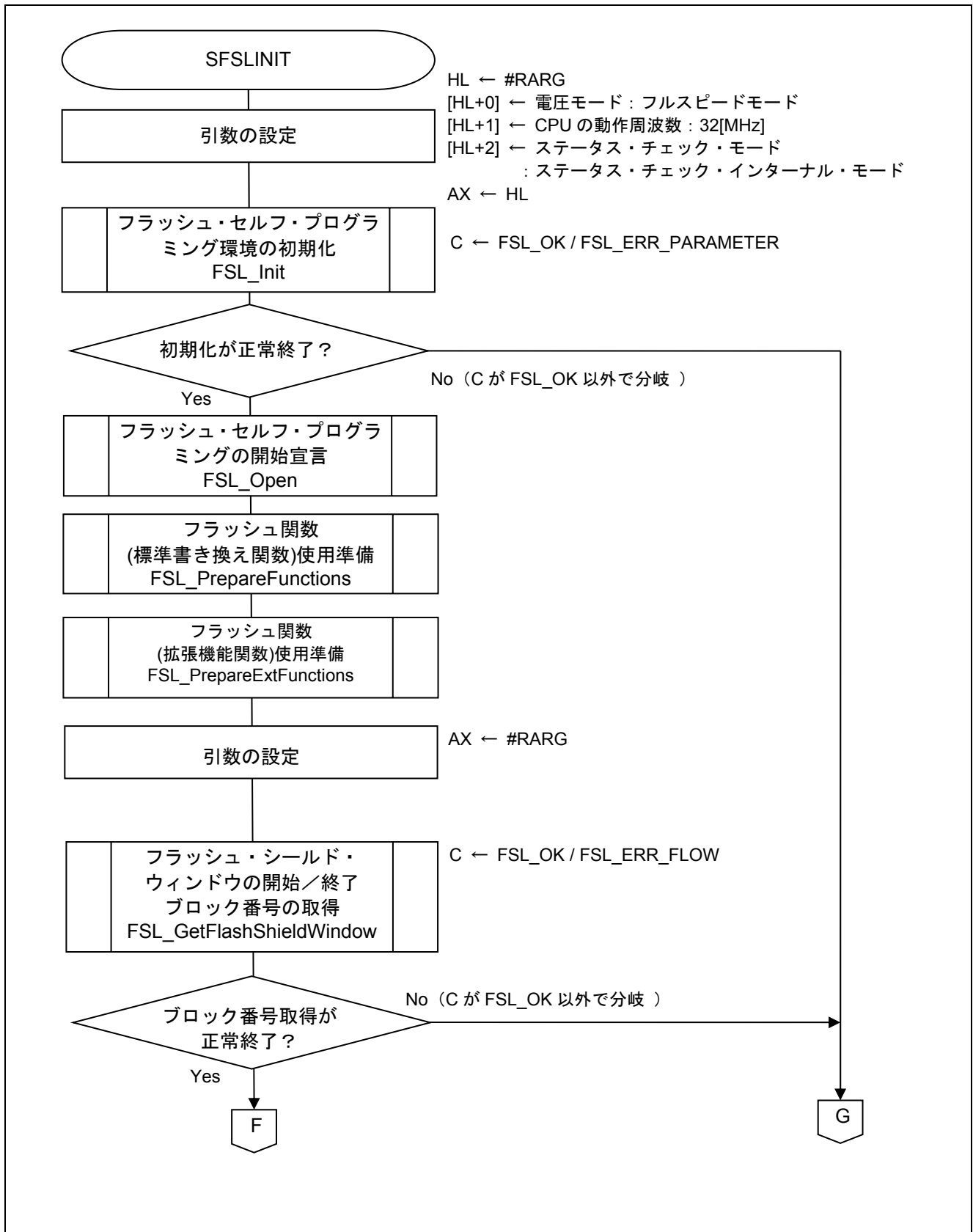


図 4.20 フラッシュ・セルフ・プログラミング初期設定(1/2)

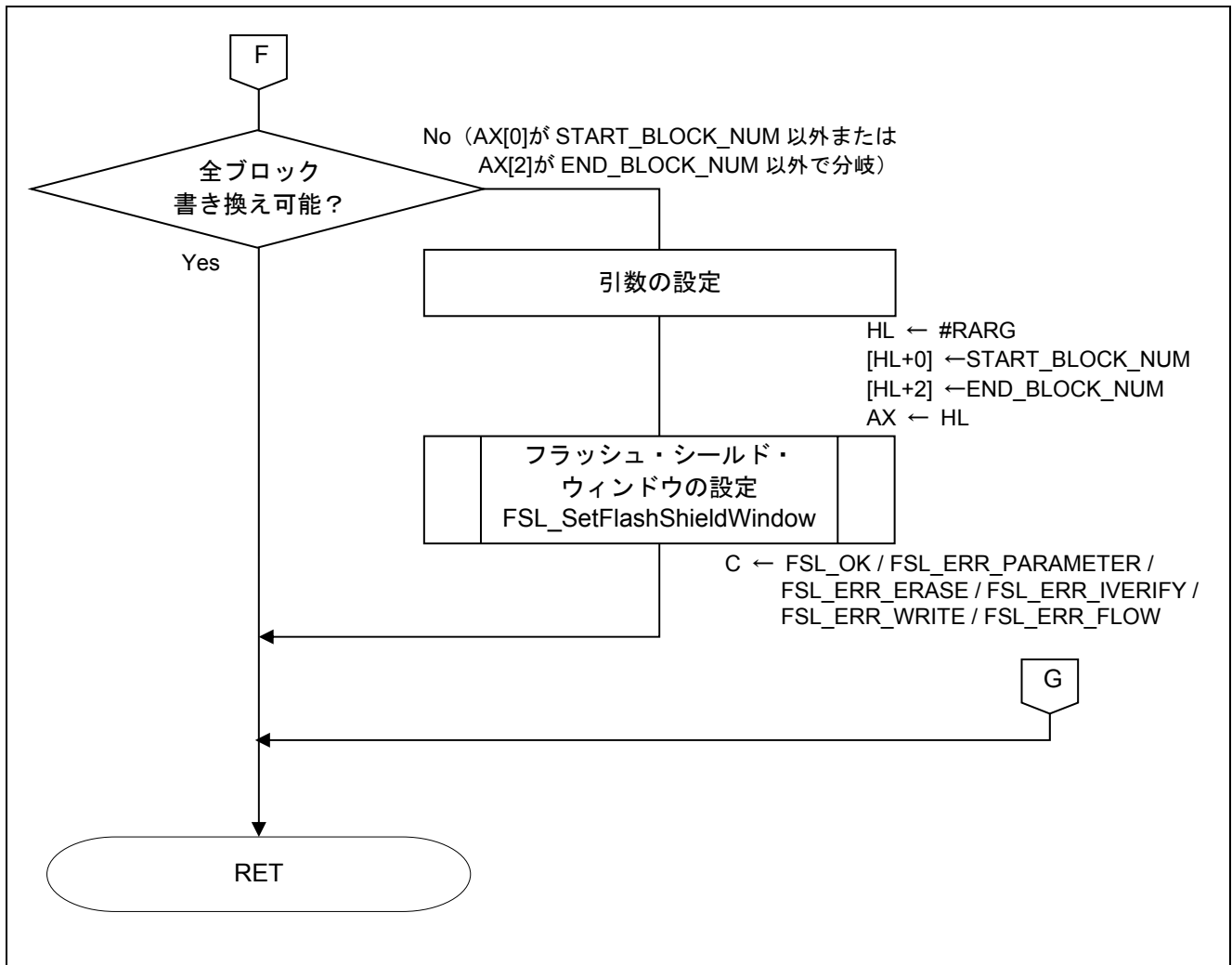


図 4.21 フラッシュ・セルフ・プログラミング初期設定(2/2)

5.10.18 フラッシュ書き換え実行

図 4.22 にフラッシュ書き換え実行(1/2)、図 4.23 にフラッシュ書き換え実行(2/2)のフローチャートを示します。

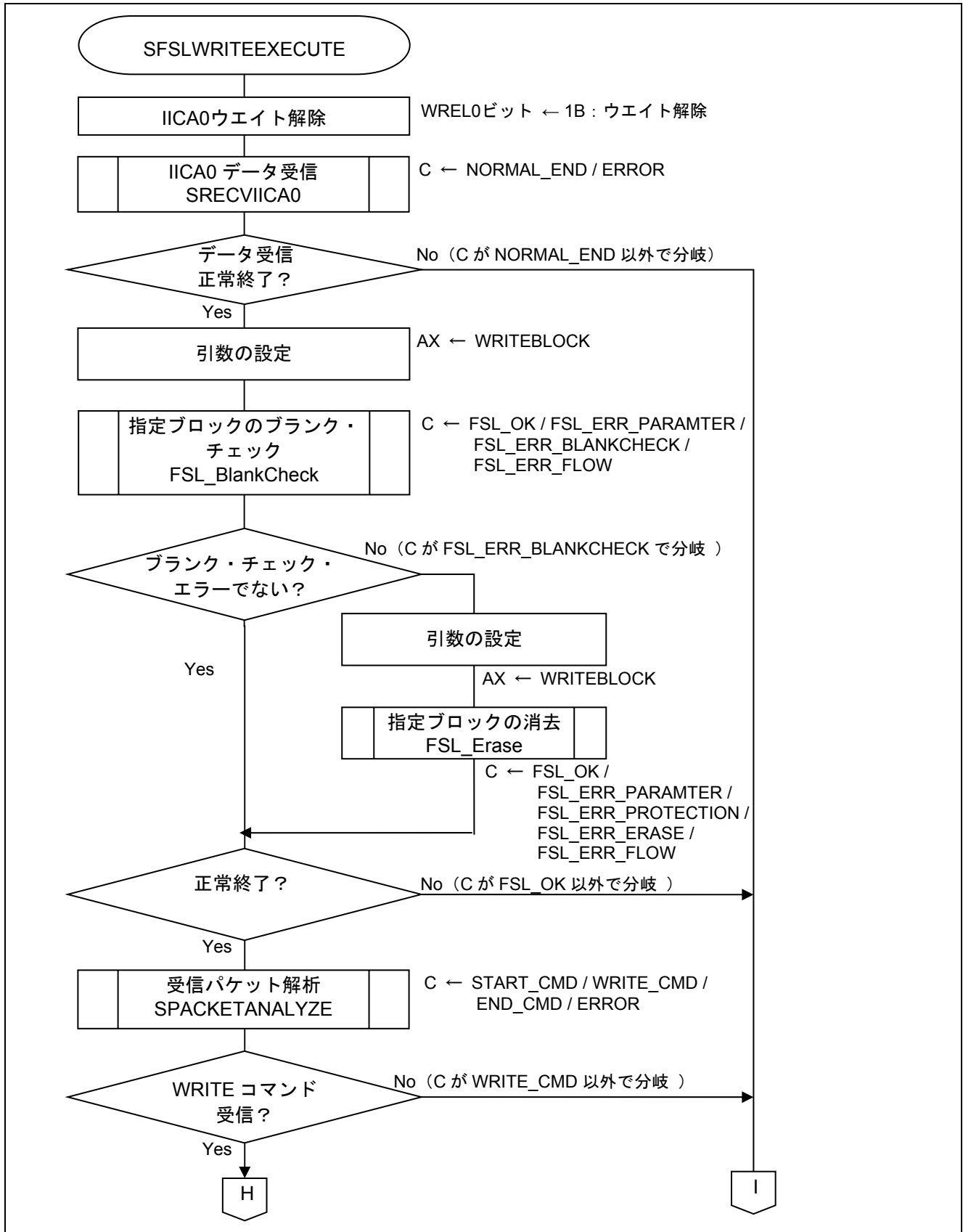


図 4.22 フラッシュ書き換え実行(1/2)

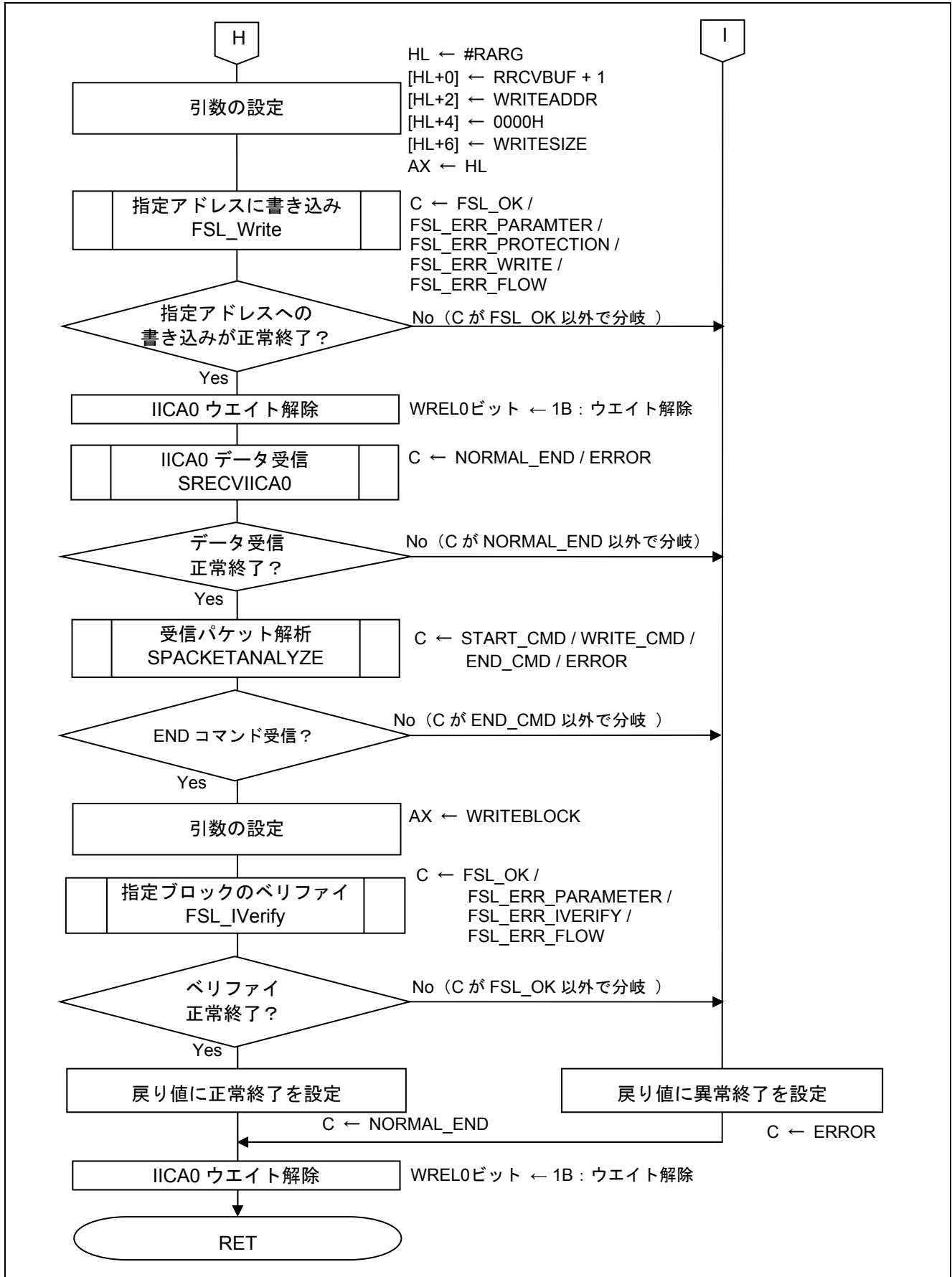


図 4.23 フラッシュ書き換え実行(2/2)

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 ハードウェア編 (R01UH0200J)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (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 フラッシュ・データ・ライブラリ
------	-----------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.02.14	—	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
		47	参考ドキュメントを追加

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。

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、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いません。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、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。



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/>