

---

# RL78/G13

R01AN1362JJ0120

Rev. 1.20

2016.06.01

## フラッシュ・データ・ライブラリ Type04

---

### 要旨

本アプリケーションノートでは、フラッシュ・データ・ライブラリ Type04（フラッシュ・データ・ライブラリを使用して、データ・フラッシュ・メモリへの書き込みや読み出しを行う方法を説明します。

### 対象デバイス

RL78/G13

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	4
1.1	フラッシュ・データ・ライブラリ概要	5
1.2	フラッシュ・データ・ライブラリのハードウェア環境	5
1.2.1	データ・フラッシュ・メモリについて	7
1.3	フラッシュ・データ・ライブラリのソフトウェア環境	8
1.3.1	セルフ RAM	8
1.3.2	レジスタ・バンク	8
1.3.3	スタック、データ・バッファ	8
1.4	フラッシュ・データ・ライブラリ取得方法	9
2.	動作確認条件	9
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	動作概要	12
5.2	ファイル構成	15
5.3	オプション・バイトの設定一覧	16
5.4	リンク・ディレクティブ・ファイル	17
5.5	定数一覧	17
5.6	変数一覧	18
5.7	関数一覧	19
5.8	関数仕様	20
5.9	フローチャート	31
5.9.1	初期設定関数	32
5.9.2	システム初期化関数	33
5.9.3	入出力ポートの設定	34
5.9.4	CPUクロックの設定	35
5.9.5	TAU0の設定	36
5.9.6	インターバル・タイマの設定	37
5.9.7	外部割り込み入力の設定	38
5.9.8	メイン処理	39
5.9.9	INTP 動作開始処理	41
5.9.10	INTP1 動作開始処理	42
5.9.11	INTP1 外部割り込み	42
5.9.12	INTP2 動作開始処理	43
5.9.13	INTP2 外部割り込み	43
5.9.14	INTP4 動作開始処理	44
5.9.15	INTP4 外部割り込み	44
5.9.16	インターバル・タイマ動作開始処理	45
5.9.17	インターバル・タイマ割り込み	46
5.9.18	インターバル・タイマ動作停止処理	47
5.9.19	フラッシュ・データ・ライブラリ開始処理	48
5.9.20	データ読み出しコマンド処理	49
5.9.21	スイッチ押下状態のクリア	50
5.9.22	LCD 上段に表示する文字列の更新処理	51
5.9.23	LCD 下段に表示する文字列の更新処理	52
5.9.24	スイッチ状態の取得	53
5.9.25	スイッチ押下状態別処理	54
5.9.26	書き込み値インクリメント処理	56
5.9.27	書き込み対象アドレス変更処理	57

5.9.28書き込み実行処理.....	58
5.9.29ブランクチェックコマンド処理.....	60
5.9.30ブロック消去コマンド処理.....	61
5.9.31データ書き込みコマンド処理.....	62
5.9.32ベリファイコマンド処理.....	63
5.9.33長押し検出処理.....	64
5.9.34INTP 動作停止処理.....	66
5.9.35INTP1 動作停止処理.....	66
5.9.36INTP2 動作停止処理.....	67
5.9.37INTP4 動作停止処理.....	67
5.9.38TAU0 チャンネル0 動作開始処理.....	68
5.9.39TAU0 チャンネル0 動作停止処理.....	68
5.9.40データ・フラッシュ初期化処理.....	69
6. サンプルコード.....	70
7. 参考ドキュメント.....	70

## 1. 仕様

本アプリケーションノートでは、フラッシュ・データ・ライブラリの使用方法を説明します。

LCD に書き込み対象アドレス、書き込み値、読み出し値を表示します。3つのスイッチにより操作が可能です、それぞれ書き込み値の変更、書き込み対象アドレスの変更、データ・フラッシュへの書き込みが行われます。また、書き込み値の変更、書き込み対象アドレスの変更を行う2つのスイッチを1秒間同時長押しすることによってデータ・フラッシュの消去が行われます。書き込み値、書き込み対象アドレスを変更する度にLCDの表示も更新され、読み出し値は、電源投入後、書き込み対象アドレス変更時、書き込み完了後もしくは消去完了後に更新されます。

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
ポート入出力	LCD に文字列を表示する LED0 の点灯/消灯
インターバル・タイマ	チャタリング回避のウェイト時間の生成
外部割り込み入力(INTP1)	書き込み値の変更 データ・フラッシュの消去 (INTP2 との同時長押し)
外部割り込み入力(INTP2)	書き込み対象アドレスの変更 データ・フラッシュの消去 (INTP1 との同時長押し)
外部割り込み入力(INTP4)	書き込み実行

図 1.1 に LCD 表示イメージを示します。

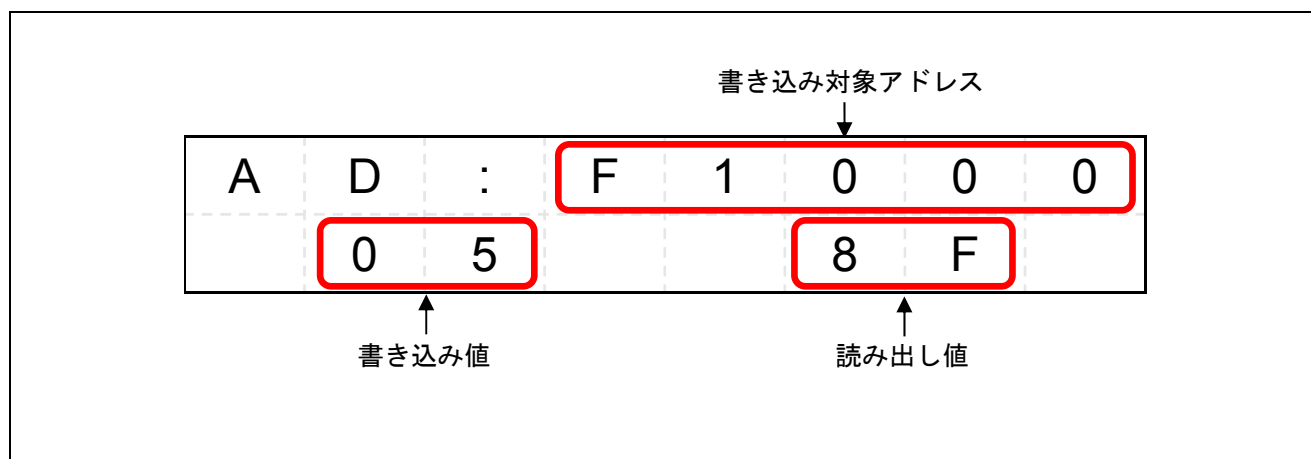


図 1.1 LCD 表示イメージ

## 1.1 フラッシュ・データ・ライブラリ概要

フラッシュ・データ・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、データ・フラッシュ・メモリへの操作を行うためのソフトウェア・ライブラリです。

フラッシュ・データ・ライブラリはユーザ・プログラムから呼び出すことにより、データ・フラッシュ・メモリの書き換えや読み出しを実行します。フラッシュ・データ・ライブラリのハードウェア環境、ソフトウェア環境の注意事項は、「RL78 ファミリー フラッシュ・データ・ライブラリ Type04 ユーザーズマニュアル (R01US0049J)」を参照してください。

## 1.2 フラッシュ・データ・ライブラリのハードウェア環境

RL78 マイクロコントローラのフラッシュ・データ・ライブラリ Type04 はシーケンサを使用し、データ・フラッシュ・メモリの書き換え制御を実行します。データ・フラッシュ・メモリへの制御をシーケンサが行ってくれるので、データ・フラッシュ・メモリ制御中にユーザ・プログラムを動作させることが可能です。この事を BGO(バック・グラウンド・オペレーション)と言います。

データ・フラッシュ・メモリの書き換え中はデータ・フラッシュ・メモリを参照できなくなりますが、コード・フラッシュ・メモリの参照は可能なため、割り込み処理やユーザ・プログラム、およびフラッシュ・データ・ライブラリ Type04 は通常通り ROM 上に配置して使用することが可能です。

図 1.2 にデータ・フラッシュ・メモリの書き換え状態を、図 1.3 にデータ・フラッシュ・メモリの書き換え制御例を示します。

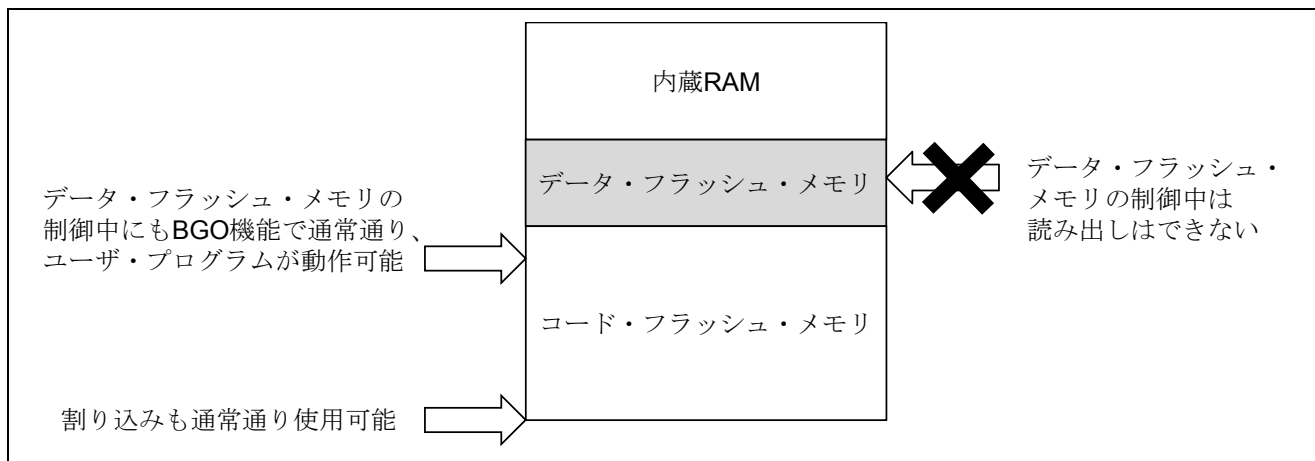


図 1.2 データ・フラッシュ・メモリの書き換え状態

RL78 マイクロコントローラのシーケンサに該当処理の実行要求を行ったのち、制御を直ちにユーザ・プログラムに戻します。データ・フラッシュ・メモリの制御の結果については、ユーザ・プログラムからステータス・チェック関数(PFDL\_Handler 関数)を呼び出し、データ・フラッシュ・メモリの制御状態を確認する処理が必要となります。

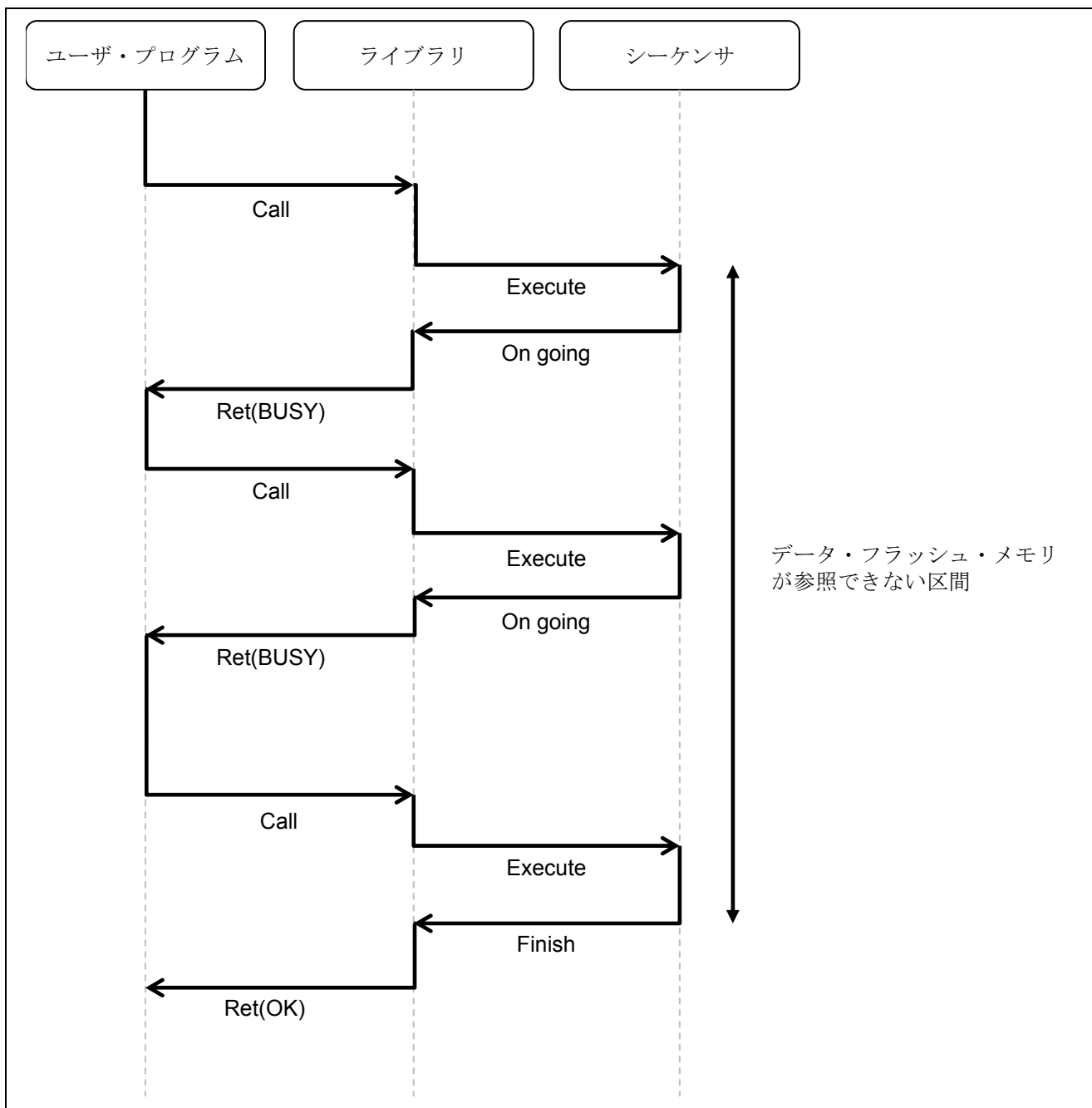


図 1.3 データ・フラッシュ・メモリの書き換え制御例

1.2.1 データ・フラッシュ・メモリについて

RL78/G13(R5F100LE)のデータ・フラッシュ・メモリの構成を以下に記載します。

RL78 マイクロコントローラは、フラッシュ・メモリが 1K バイト単位でブロック分割されています。フラッシュ・データ・ライブラリでは、このブロックを単位としてデータ・フラッシュ・メモリに対し、消去処理を行います。読み込みや書き込み、ブランク・チェック、内部ベリファイは開始アドレスと実行サイズを指定して実行します。

図 1.4 にデータ・フラッシュ・メモリのブロック位置とブロック番号を示します。

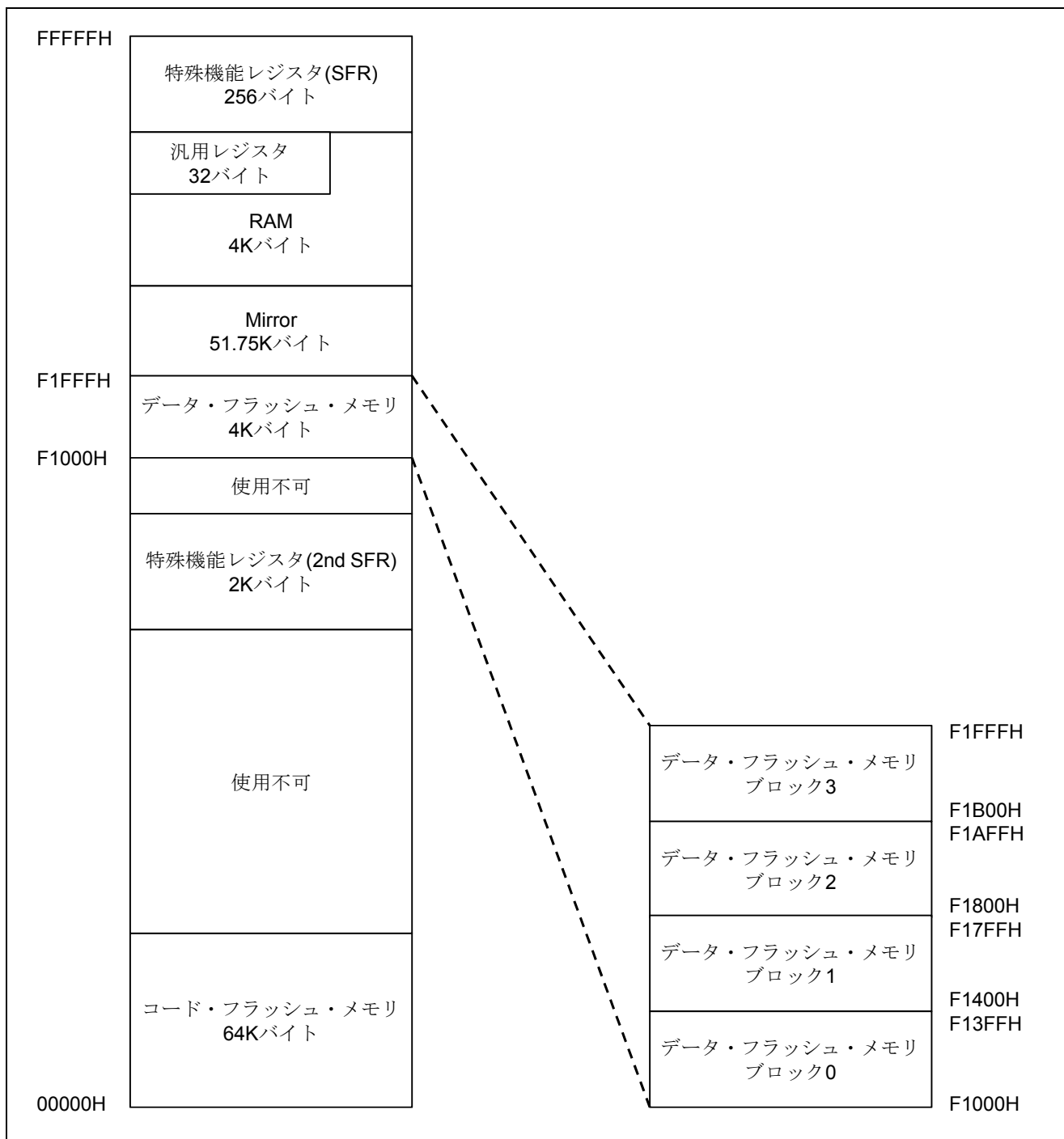


図 1.4 データ・フラッシュ・メモリのブロック位置とブロック番号

### 1.3 フラッシュ・データ・ライブラリのソフトウェア環境

フラッシュ・データ・ライブラリ Type04では、該当プログラムをユーザ領域に配置するため、使用するライブラリの容量のプログラム領域を消費し、フラッシュ・データ・ライブラリ Type04自身は、CPU、スタック、データ・バッファを使用します。

#### 1.3.1 セルフ RAM

フラッシュ・データ・ライブラリ Type04 は、ワークエリアとして 1K バイトの RAM 領域を使用する場合があります。使用する場合、この領域をセルフ RAM と呼びます。ライブラリ内で定義されているため、ユーザによる設定は不要です。

フラッシュ・データ・ライブラリ関数を呼び出すことにより、セルフ RAM 領域のデータが書き換わりません。

#### 1.3.2 レジスタ・バンク

フラッシュ・データ・ライブラリ Type04 はユーザが選択しているレジスタ・バンクの汎用レジスタ、ES/CS レジスタ、SP 及び PSW を使用します。

#### 1.3.3 スタック、データ・バッファ

フラッシュ・データ・ライブラリ Type04 は、シーケンサを使用してデータ・フラッシュ・メモリへの書き込みを行います。事前の設定や制御を行うために CPU を使用します。このため、フラッシュ・データ・ライブラリ Type04 を使用するためには、ユーザ・プログラムで指定されているスタックも必要となります。

**注意** ユーザが指定するアドレスに、スタック、データ・バッファを配置するためには、リンク・ディレクティブを使用します。

- ・スタック

ユーザ・プログラムで使用するスタックに加え、フラッシュ・データ・ライブラリ関数で必要となるスタックの容量を事前に確保し、フラッシュ・データ・ライブラリ Type04 を実行する場合におけるスタック処理で、ユーザ使用 RAM が破壊されないように配置する必要があります。スタックの指定可能範囲はセルフ RAM および FFE20H-FFEFFFH 以外の内蔵 RAM となります。

- ・データ・バッファ

データ・バッファの用途は以下の通りです。

- フラッシュ・データ・ライブラリ Type04 の内部処理におけるワーク領域
- 書き込み実行時、書き込むデータを配置する領域
- 読み出し実行時、取得する読み込みデータを配置する領域

なお、データ・バッファの先頭アドレスの指定可能範囲は、スタックと同様、セルフ RAM 及び FFE20H-FFEFFFH 以外の内蔵 RAM となります。



## 1.4 フラッシュ・データ・ライブラリ取得方法

コンパイルを実行する前に、最新版のデータ・フラッシュ・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。

”incr178”フォルダに”pfdl.h”、”pfdl.inc”、”pfdl\_types.h”をコピーする。  
”librl78”フォルダに”pfdl.lib”をコピーする。

データ・フラッシュ・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/G13 (R5F100LEA)
動作周波数	<ul style="list-style-type: none"> <li>● 高速オンチップ・オシレータ・クロック : 32MHz</li> <li>● CPU/周辺ハードウェア・クロック : 32MHz</li> </ul>
動作電圧	5.0V (2.9V~5.5V で動作可能) LVD 動作 (VLVD) : 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
C コンパイラ	ルネサス エレクトロニクス製 CA78K0R V1.72
使用ボード	Renesas Starter Kit for RL78/G13 (R0K50100LS000BE)
フラッシュ・データ・ライブラリ (Type, Ver)	FDLRL78 Type04, Ver1.04 <sup>注</sup>

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/G13 初期設定 (R01AN0451J) 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。

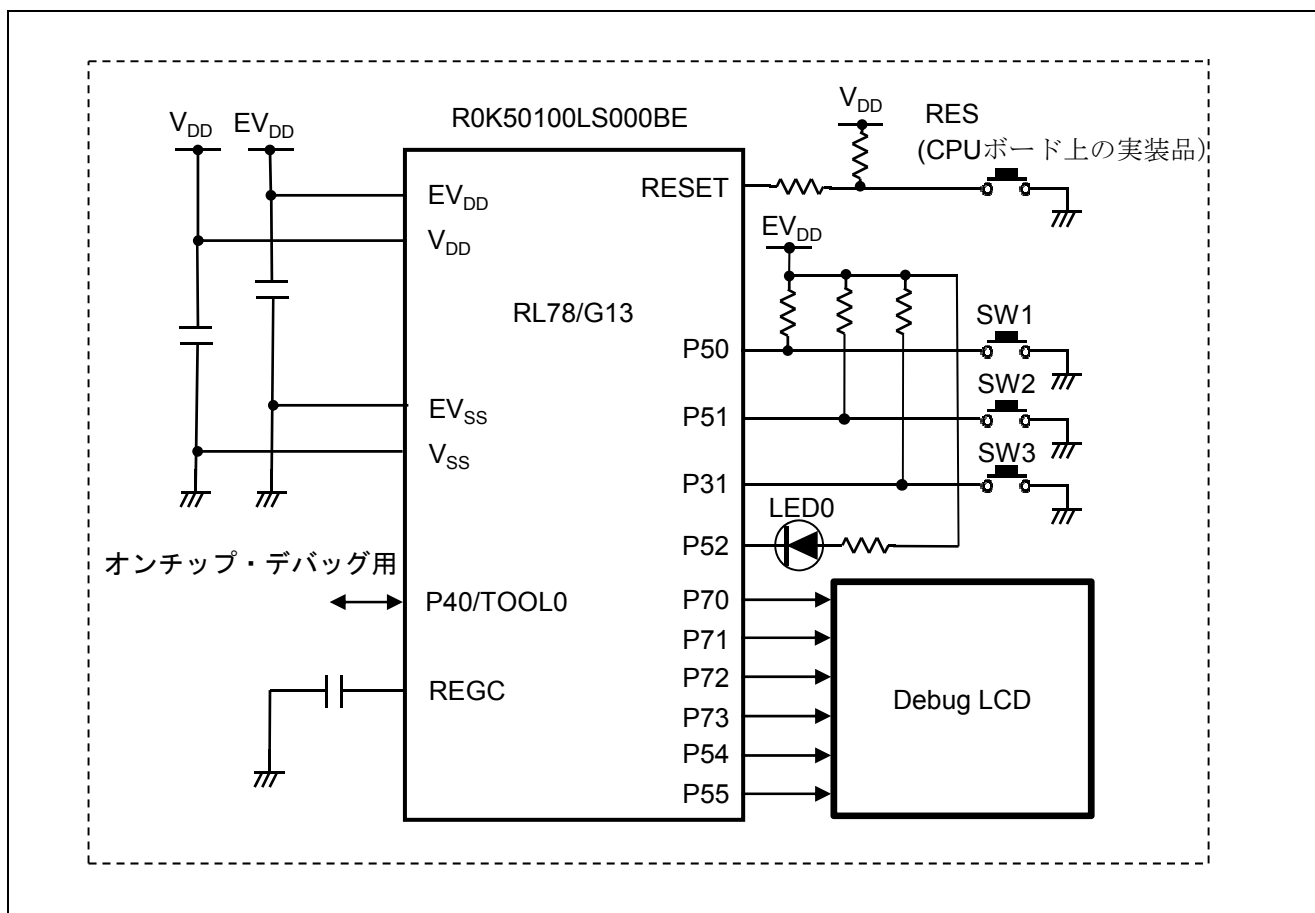


図 4.1 ハードウェア構成例

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  $V_{DD}$  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
P31/TI03/TO03/INTP4	入力	書き込み実行
P50/INTP1/SI11/SDA11	入力	「書き込み対象アドレス」切り換え データ・フラッシュ消去 (INTP2 との同時長押し)
P51/INTP2/SO11	入力	「書き込み値」インクリメント データ・フラッシュ消去 (INTP1 との同時長押し)
P52	出力	フラッシュ・アクセス中 (書き込み/消去) を示す LED の 点灯/消灯
P54	出力	Debug LCD 制御
P55	出力	Debug LCD 制御
P70/KR0/SCK21/SCL21	出力	Debug LCD 制御
P71/KR1/SI21/SDA21	出力	Debug LCD 制御
P72/KR2/SO21	出力	Debug LCD 制御
P73/KR3/SO01	出力	Debug LCD 制御

## 5. ソフトウェア説明

### 5.1 動作概要

本アプリケーションノートでは、フラッシュ・データ・ライブラリの使用方法を説明します。

LCD に書き込み対象アドレス、書き込み値、読み出し値を表示します。3つのスイッチにより操作が可能で、それぞれ書き込み値の変更、書き込み対象アドレスの変更、データ・フラッシュへの書き込みが行われます。また、書き込み値の変更、書き込み対象アドレスの変更を行う2つのスイッチを1秒間同時長押しすることによってデータ・フラッシュの消去が行われます。書き込み値、書き込み対象アドレスを変更する度にLCDの表示も更新され、読み出し値は、電源投入後、書き込み対象アドレス変更時、書き込み完了後もしくは消去完了後に更新されます。

#### (1) TAU0 チャンネル0の設定を行います

<設定条件>

- TAU0 チャンネル0を使用します。
- 動作クロックは500kHzを使用します。
- スタート・トリガは、ソフトウェア・トリガ・スタートのみ有効とします。
- 有効エッジは、立ち下がりエッジを使用します。
- 動作モードはインターバル・タイマ・モードを使用します。
- カウント・スタートと割り込みの設定は、「カウント開始時にタイマ割り込みを発生しない」。
- タイマ割り込み(INTTM00)を使用します。
- 割り込みタイミングを100msに設定します。

#### (2) 12ビット・インターバル・タイマの設定を行います

<設定条件>

- 動作クロックは15kHzを使用します。
- 割り込み優先順位をレベル2に設定します。
- インターバル・タイムを10msに設定します。

#### (3) 外部割り込み入力の設定を行います

- 割り込み優先順位をレベル1に設定します。
- 有効エッジは、立下りエッジを使用します。

#### (4) LCDの初期化を行います。

#### (5) INTPの動作を開始します。

- INTP1, INTP2, INTP4 端子のエッジ検出割り込み処理を許可します。
- INTP1, INTP2, INTP4 の割り込みを許可します。

#### (6) フラッシュ・データ・ライブラリの初期設定を行います。

- フラッシュ・データ・ライブラリで使用する RAM の初期化を行います。
  - 初期設定途中でエラーになった場合は、LCD に"ERROR!"と表示し、以降の処理は行いません。
- (7) データ・フラッシュの内容を読み出します。
- ブロック 0 の先頭アドレスにあるデータを読み出します。
- (8) フラッシュ・データ・ライブラリを停止させ、HALT モードへの移行が可能な状態にします。
- (9) スイッチ押下状態をクリアします。
- (10) LCD に書き込むデータを上段、下段共に更新します。
- (11) LCD の上段に書き込み対象アドレス、下段に書き込み値、及び読み出し値を表示します。
- (12) スイッチが何も押されていないならば、HALT モードに移行して、スイッチ入力を待ちます。
- (13) スイッチによる外部割り込みが入ると、HALT モードから復帰し、チャタリング回避のために以下の処理を行います。
- INTP1、INTP2、INTP4 のいずれかの割り込みハンドラでインターバル・タイマのカウンタ動作を開始します。
  - インターバル・タイマの割り込みが発生するまで待ちます。
  - インターバル・タイマの割り込みハンドラでスイッチ状態の確認をします。  
具体的には、P31、P50、P51 の入力レベルを確認します。
  - P31、P50、P51 のいずれかが"0"であればスイッチ押下されたと判定し、スイッチ押下確認フラグをセットします。
  - P31、P50、P51 が全て"1"であればスイッチが押下されていないと判定し、スイッチ押下確認フラグをクリアして(9)に戻ります。
- (14) どのスイッチが押されているかを判別します。
- (15) スイッチの押下状態別に処理を行います。
- SW1 のみが押されていれば、書き込み値をインクリメントします。
  - SW2 のみが押されていれば、書き込み対象アドレスを変更します。
  - SW3 のみが押されていれば、フラッシュ・アクセス中を示す LED0 を点灯し、選択しているアドレスに書き込み値を書き込みます。
    - 書き込み後に読み出しを行い、書き込み値と読み出し値の比較を行って LED0 を消灯します。
    - 書き込み値と読み出し値が異なる場合は、LCD に"ERROR!"と表示し、以降の処理は行いません。

- SW1 と SW2 が 1 秒間同時長押しされていれば、フラッシュ・アクセス中を示す LED0 を点灯し、データ・フラッシュを初期化します。
  - 正しく初期化されなかった場合は、LCD に"ERROR!"と表示し、以降の処理は行いません。
  - 初期化後、LED0 を消灯し、選択しているアドレスのデータを読み出します。

(16) (9)に戻ります。

## 5.2 ファイル構成

表 5.1 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 5.1 ファイル構成

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_INTCStart R_MAIN_ClearSwitchFlag R_MAIN_GetSwitchStatus R_MAIN_DetectLongPush R_MAIN_IncrementValue R_MAIN_SwitchProcess R_MAIN_UpdateStringUpper R_MAIN_UpdateStringDowner
r_pfdl.c	フラッシュ・データ・ライブラリ実行処理	R_FDL_Init R_FDL_BlankCheck R_FDL_Erase R_FDL_Verify R_FDL_Read R_FDL_Write R_FDL_ChangeAddress R_FDL_ExecuteWrite R_FDL_ClearDataFlash

### 5.3 オプション・バイトの設定一覧

表 5.2 にオプション・バイト設定一覧を示します。

表 5.2 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	11101000B	HS モード、HOCO クロック : 32MHz
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/G13 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。



## 5.4 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・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。

```

*****
;
; Redefined RAM area
*****
;
; -----
; Redefined default data segment RAM
; -----
MEMORY RAM      : ( 0FF30AH, 000B16H )
; -----
; Define new memory entry for saddr area
; -----
MEMORY RAM_SADDR : ( 0FFE20H, 0001E0H )

```

ライブラリの制限領域を標準 RAM 領域として使用しないよう設定

## 5.5 定数一覧

表 5.3 にサンプルコードで使用する定数を示します。

表 5.3 サンプルコードで使用する定数

定数名	設定値	内容
SW_ON	1	スイッチ押下確認
SW_OFF	0	スイッチ押下確認クリア
ON_SW_1	0x01	スイッチ 1 押下状態
ON_SW_2	0x02	スイッチ 2 押下状態
ON_SW_3	0x04	スイッチ 3 押下状態
OFF_SW_ALL	0x00	スイッチ押下状態全クリア
FLASH_START_ADDRESS	0xF1000	データ・フラッシュの開始アドレス
TARGET_BLOCK	0	書き込み対象ブロック <sup>注</sup>
BLOCK_SIZE	0x400	1 ブロックのサイズ(byte)
WRITE_SIZE	1	書き込みデータサイズ(byte)
MAX_VALUE	0xFF	データ・フラッシュ書き込み値の最大値
MAX_ADDRESS	(TARGET_BLOCK +1) * BLOCK_SIZE -1	データ・フラッシュ書き込みアドレスの最大値
PFDL_NG	1	フラッシュ・データ・ライブラリ処理異常終了
FDL_FRQ	32	周波数設定[MHz]
FDL_VOL	0x00	電圧モード(フルスピードモード)
LCD_SIZE	8	LCD の最大表示文字数

注 TARGET\_BLOCK の有効範囲は 0 ~ 3 です。これ以外の値に設定した場合は、ビルド時にエラーとなります。TARGET\_BLOCK の設定値と書き込み対象ブロックの関係を以下に記載します。

- 0 : 書き込み対象ブロックがデータ・フラッシュのブロック 0 (アドレス 0xF1000 ~ 0xF13FF)
- 1 : 書き込み対象ブロックがデータ・フラッシュのブロック 1 (アドレス 0xF1400 ~ 0xF17FF)
- 2 : 書き込み対象ブロックがデータ・フラッシュのブロック 2 (アドレス 0xF1800 ~ 0xF1BFF)
- 3 : 書き込み対象ブロックがデータ・フラッシュのブロック 3 (アドレス 0xF1C00 ~ 0xF1FFF)

## 5.6 変数一覧

表 5.4 にグローバル変数を示します。

表 5.4 グローバル変数

Type	Variable Name	Contents	Function Used
uint8_t	g_sw_push	スイッチ押下確認フラグ	r_main r_it_interrupt R_MAIN_ClearSwitchFlag
uint8_t	g_it_flag	インターバル・タイマ割り込み発生フラグ	r_main r_it_interrupt
uint8_t	g_read_value	読み出し値	R_FDL_Read R_MAIN_UpdateStringDowner
uint8_t	g_write_value	書き込み値	R_FDL_Read R_FDL_Write R_MAIN_Increment R_MAIN_UpdateStringDowner
uint16_t	g_write_address	書き込み対象アドレス	R_FDL_BlankCheck R_FDL_Read R_FDL_Write R_FDL_ChangeAddress R_MAIN_UpdateStringUpper

## 5.7 関数一覧

表 5.5 に関数一覧を示します。

表 5.5 関数一覧

関数名	概要
R_MAIN_INTCStart	INTP 動作開始処理
R_INTC1_Start	INTP1 動作開始処理
r_intc1_interrupt	INTP1 外部割り込み
R_INTC2_Start	INTP2 動作開始処理
r_intc2_interrupt	INTP2 外部割り込み
R_INTC4_Start	INTP4 動作開始処理
r_intc4_interrupt	INTP4 外部割り込み
R_IT_Start	インターバル・タイマ動作開始処理
r_it_interrupt	インターバル・タイマ割り込み
R_IT_Stop	インターバル・タイマ動作停止処理
R_FDL_Init	フラッシュ・データ・ライブラリ開始処理
R_FDL_Read	データ読み出しコマンド処理
R_MAIN_ClearSwitchFlag	スイッチ押下状態のクリア
R_MAIN_UpdateStringUpper	LCD 上段に表示する文字列の更新
R_MAIN_UpdateStringDowner	LCD 下段に表示する文字列の更新
R_MAIN_GetSwitchStatus	スイッチ状態の取得
R_MAIN_SwitchProcess	スイッチ押下状態別処理
R_MAIN_IncrementValue	書き込み値インクリメント処理
R_FDL_ChangeAddress	書き込み対象アドレス変更処理
R_FDL_ExecuteWrite	書き込み実行処理
R_FDL_BlankCheck	ブランクチェックコマンド処理
R_FDL_Erase	ブロック消去コマンド処理
R_FDL_Write	データ書き込みコマンド処理
R_FDL_Verify	ベリファイコマンド処理
R_MAIN_DetectLongPush	長押し検出処理
R_MAIN_INTCStop	INTP 動作停止処理
R_INTC1_Stop	INTP1 動作停止処理
R_INTC2_Stop	INTP2 動作停止処理
R_INTC4_Stop	INTP4 動作停止処理
R_TAU0_Channel0_Start	TAU0 チャンネル 0 動作開始処理
R_TAU0_Channel0_Stop	TAU0 チャンネル 0 動作停止処理
R_FDL_ClearDataFlash	データ・フラッシュ初期化処理

## 5.8 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\_MAIN\_INTCStart

---

概要	INTP 動作開始処理
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h
宣言	void R_MAIN_INTCStart(void)
説明	INTP 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\_INTC1\_Start

---

概要	INTP1 動作開始処理
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
宣言	void R_INTC1_Start(void)
説明	INTP1 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\_intc1\_interrupt

---

概要	INTP1 外部割り込み
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_it.h r_cg_userdefine.h
宣言	__interrupt void r_intc1_interrupt(void)
説明	インターバル・タイマを動作させます。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_INTC2\_Start

---

概要	INTP2 動作開始処理
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
宣言	void R_INTC2_Start(void)
説明	INTP2 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\_intc2\_interrupt

---

概要	INTP2 外部割り込み
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_it.h r_cg_userdefine.h
宣言	__interrupt void r_intc2_interrupt(void)
説明	インターバル・タイマを動作させます。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_INTC4\_Start

---

概要	INTP4 動作開始処理
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
宣言	void R_INTC4_Start(void)
説明	INTP4 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\_intc4\_interrupt

---

概要	INTP4 外部割り込み
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_it.h r_cg_userdefine.h
宣言	__interrupt void r_intc4_interrupt(void)
説明	インターバル・タイマを動作させます。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_IT\_Start

---

概要	インターバル・タイマ動作開始処理
ヘッダ	r_cg_macrodriver.h r_cg_it.h r_cg_userdefine.h
宣言	void R_IT_Start(void)
説明	インターバル・タイマ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\_it\_interrupt

---

概要	インターバル・タイマ割り込み
ヘッダ	r_cg_macrodriver.h r_cg_it.h r_cg_userdefine.h
宣言	__interrupt void r_it_interrupt(void)
説明	インターバル・タイマの動作を停止します。 SW1～SW3 のいずれかのスイッチが押下状態であれば、スイッチ押下状態 g_sw_push を”1”にセットします。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_IT\_Stop

---

概要	インターバル・タイマ動作停止処理
ヘッダ	r_cg_macrodriver.h r_cg_it.h r_cg_userdefine.h
宣言	void R_IT_Stop(void)
説明	インターバル・タイマ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\_FDL\_Init

---

概要	フラッシュ・データ・ライブラリ開始処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	void R_FDL_Init(void)
説明	フラッシュ・データ・ライブラリ Type04 で使用する RAM の初期化、開始を行います。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_FDL\_Read

---

概要	データ読み出しコマンド処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	void R_FDL_Read(void)
説明	データ読み出しコマンドを実行し、読み出しデータ格納変数 g_read_value に格納します。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_MAIN\_ClearSwitchFlag

---

概要	スイッチ押下状態のクリア
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskr178g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h
宣言	void R_MAIN_ClearSwitchFlag(void)
説明	スイッチの押下状態確認フラグ g_sw_push をクリアします。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_MAIN\_UpdateStringUpper

---

概要	LCD 上段に表示される文字列の更新	
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h	
宣言	void R_MAIN_UpdateStringUpper(int8_t upper_string[LCD_SIZE + 1])	
説明	LCD 上段に表示する文字列を g_write_address の値に更新します。	
引数	upper_string[LCD_SIZE + 1]	LCD 上段に表示される文字列
リターン値	なし	
備考	なし	

## [関数名] R\_MAIN\_UpdateStringDowner

---

概要	LCD 下段に表示される文字列の更新	
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h	
宣言	void R_MAIN_UpdateStringDowner(int8_t downer_string[LCD_SIZE + 1])	
説明	LCD 下段に表示する文字列を g_write_value、g_read_value の値に更新します。	
引数	downer_string[LCD_SIZE + 1]	LCD 下段に表示される文字列
リターン値	なし	
備考	なし	



**[関数名] R\_MAIN\_GetSwitchStatus**


---

概要	スイッチ状態の取得
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskr78g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h
宣言	uint8_t R_MAIN_GetSwitchStatus(void)
説明	SW1、SW2、SW3 の状態を取得します。
引数	なし
リターン値	スイッチ押下状態 : sw_status (初期値 = 0) ● SW 押下なし : sw_status ● SW1 押下 : sw_status + ON_SW_1 ● SW2 押下 : sw_status + ON_SW_2 ● SW3 押下 : sw_status + ON_SW_3
備考	なし

**[関数名] R\_MAIN\_SwitchProcess**


---

概要	スイッチ押下状態別処理
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskr78g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h
宣言	uint8_t R_MAIN_SwitchProcess(uint8_t sw_status)
説明	3つのスイッチの押下状態によって処理を分岐します。
引数	sw_status                                 スイッチ押下状態
リターン値	● 正常終了 : PFDL_OK ● 異常終了 : PFDL_NG
備考	なし

## [関数名] R\_MAIN\_IncrementValue

---

概要	書き込み値インクリメント処理
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskr178g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h
宣言	void R_MAIN_IncrementValue(void)
説明	データ・フラッシュへの書き込み値 g_write_value をインクリメントします。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_FDL\_ChangeAddress

---

概要	書き込み対象アドレス変更処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	void R_FDL_ChangeAddress(void)
説明	書き込み対象アドレスを変更し、変更後のアドレスのデータを読み出して読み出しデータ格納変数 g_read_value に格納します。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_FDL\_ExecuteWrite

---

概要	書き込み実行処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	uint8_t R_FDL_ExecuteWrite(void)
説明	データ・フラッシュの書き込み対象アドレスに書き込み値を書き込みます。
引数	なし
リターン値	● 正常終了 : PFDL_OK ● 異常終了 : PFDL_NG
備考	なし

## [関数名] R\_FDL\_BlankCheck

---

概要	ブランクチェックコマンド処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	uint8_t R_FDL_BlankCheck(void)
説明	対象のアドレスがブランク状態かどうかをチェックします。
引数	なし
リターン値	<ul style="list-style-type: none"> <li>● 正常終了 : PFDL_OK</li> <li>● アイドル状態 : PFDL_IDLE</li> <li>● ブランク・チェック・エラー : PFDL_ERR_MARGIN</li> </ul>
備考	なし

## [関数名] R\_FDL\_Erase

---

概要	ブロック消去コマンド処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	uint8_t R_FDL_Erase(void)
説明	ブロック全体のデータを消去します。
引数	なし
リターン値	<ul style="list-style-type: none"> <li>● 正常終了 : PFDL_OK</li> <li>● アイドル状態 : PFDL_IDLE</li> <li>● 消去エラー : PFDL_ERR_ERASE</li> </ul>
備考	なし

## [関数名] R\_FDL\_Write

---

概要	データ書き込みコマンド処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	uint8_t R_FDL_Write(void)
説明	データ・フラッシュにデータを書き込みます。
引数	なし
リターン値	<ul style="list-style-type: none"> <li>● 正常終了 : PFDL_OK</li> <li>● アイドル状態 : PFDL_IDLE</li> <li>● 書き込みエラー : PFDL_ERR_WRITE</li> </ul>
備考	なし

## [関数名] R\_FDL\_Verify

---

概要	ベリファイコマンド処理
ヘッダ	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
宣言	uint8_t R_FDL_Verify(void)
説明	書き込んだ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	<ul style="list-style-type: none"> <li>● 正常終了 : PFDL_OK</li> <li>● アイドル状態 : PFDL_IDLE</li> <li>● 内部ベリファイ・エラー : PFDL_ERR_MARGIN</li> </ul>
備考	なし

## [関数名] R\_MAIN\_DetectLongPush

---

概要	長押し検出処理
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskri78g13def.h pfdl.h pfdl_types.h stdlib.h string.h r_cg_userdefine.h
宣言	uint8_t R_MAIN_DetectLongPush(void)
説明	スイッチの長押しを検出します。
引数	なし
リターン値	<ul style="list-style-type: none"> <li>● 長押し検出 : SW_ON</li> <li>● 長押し不検出 : SW_OFF</li> </ul>
備考	なし

## [関数名] R\_MAIN\_INTCStop

---

概要	INTP 動作停止処理
ヘッダ	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h fdl.h fdl_descriptor.h eel.h eel_descriptor.h eel_user_types.h lcd.h stdlib.h string.h r_cg_userdefine.h
宣言	void R_MAIN_INTCStop(void)
説明	INTP1、INTP2、INTP4 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\_INTC1\_Stop

---

概要	INTP1 動作停止処理
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
宣言	void R_INTC1_Stop(void)
説明	INTP1 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\_INTC2\_Stop

---

概要	INTP2 動作停止処理
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
宣言	void R_INTC2_Stop(void)
説明	INTP2 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\_INTC4\_Stop

---

概要	INTP4 動作停止処理
ヘッダ	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
宣言	void R_INTC4_Stop(void)
説明	INTP4 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\_TAU0\_Channel0\_Start

---

概要	TAU0 チャンネル 0 動作開始処理
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	タイマ・アレイ・ユニット 0 チャンネル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\_Stop

---

概要	TAU0 チャンネル 0 動作停止処理
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	タイマ・アレイ・ユニット 0 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] R\_FDL\_ClearDataFlash

---

概要	データ・フラッシュ初期化処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h pfdl.h pfdl_types.h
宣言	uint8_t R_FDL_ClearDataFlash(void)
説明	データ・フラッシュの内容を初期化します。
引数	なし
リターン値	● 正常終了 : PFDL_OK ● 異常終了 : PFDL_NG
備考	なし

## 5.9 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。

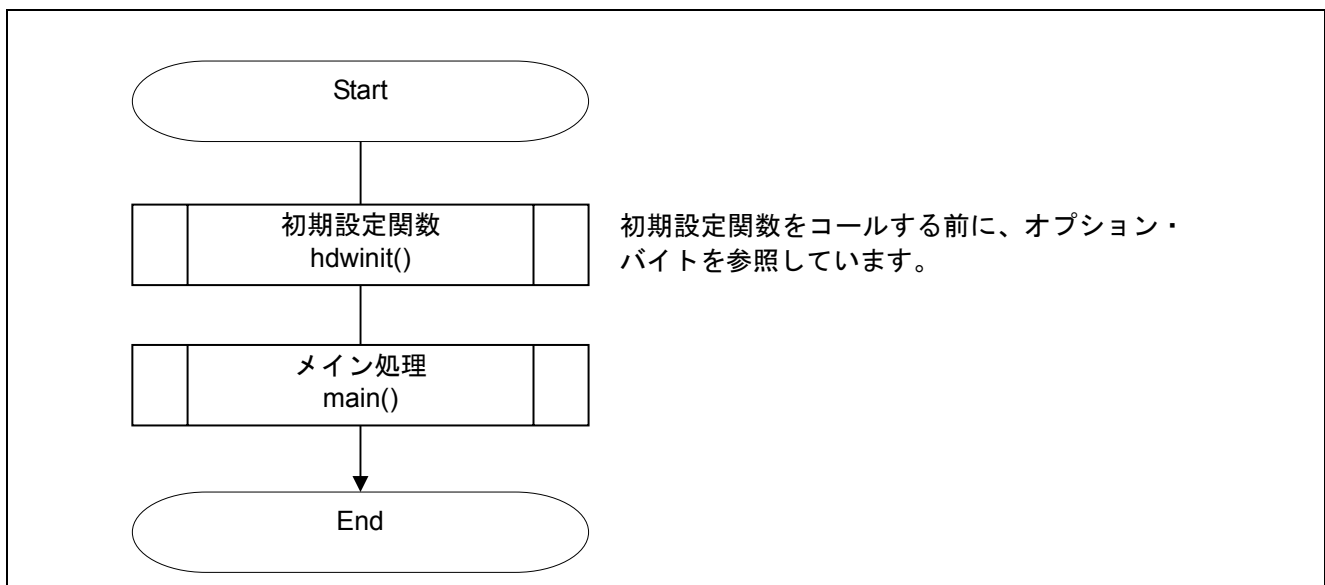


図 5.1 全体フロー

## 5.9.1 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。

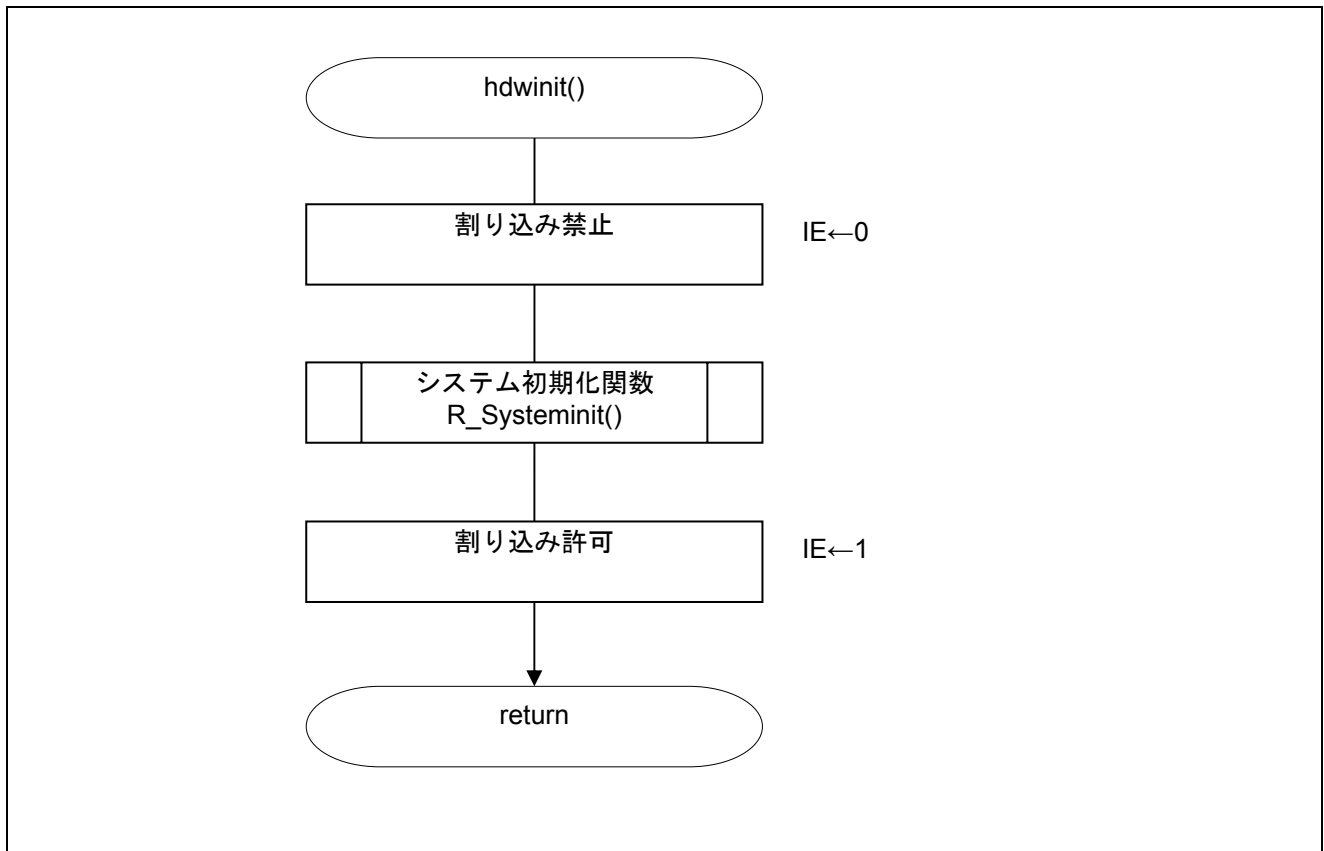


図 5.2 初期設定関数



## 5.9.2 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。

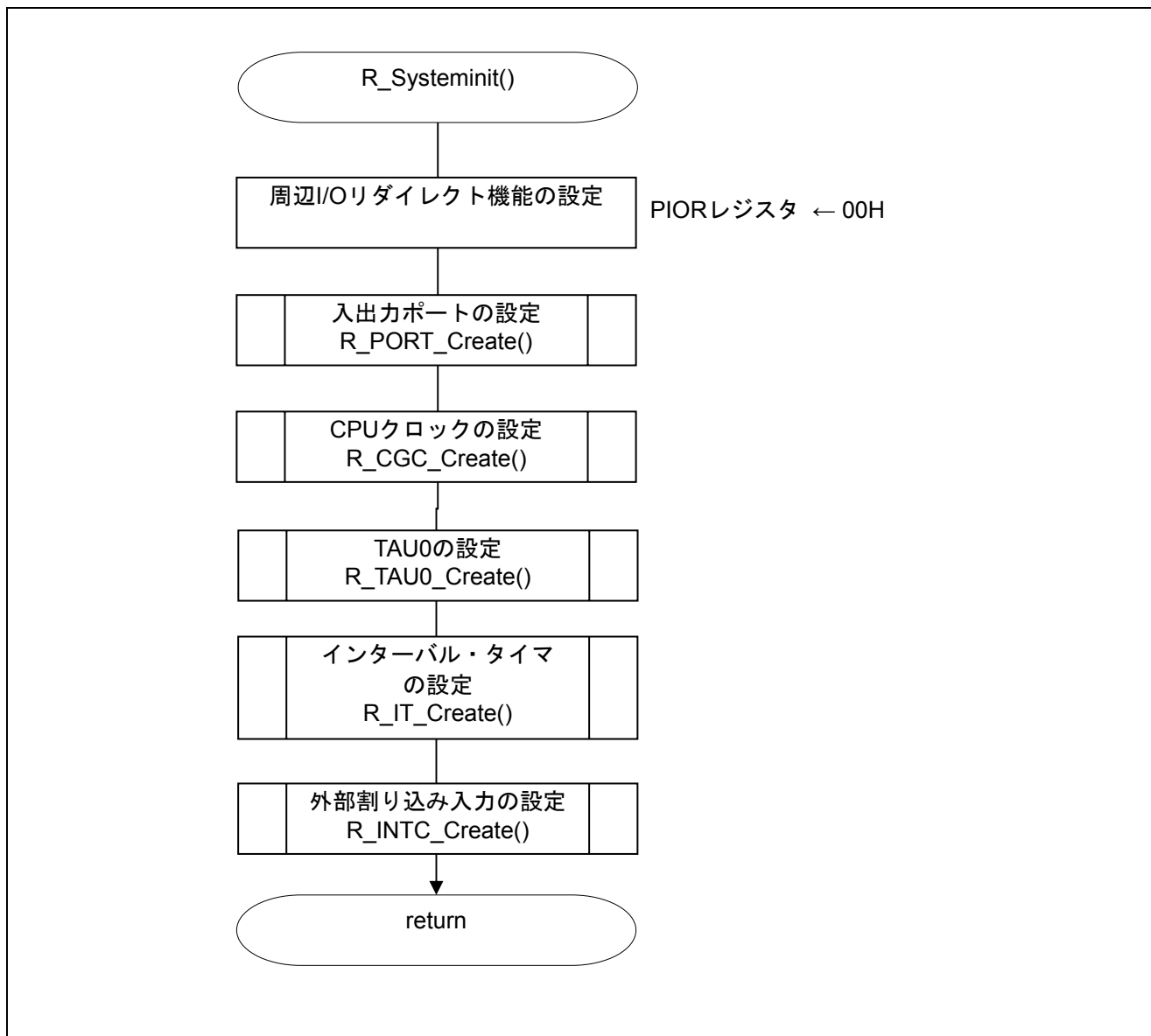


図 5.3 システム初期化関数

## 5.9.3 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。

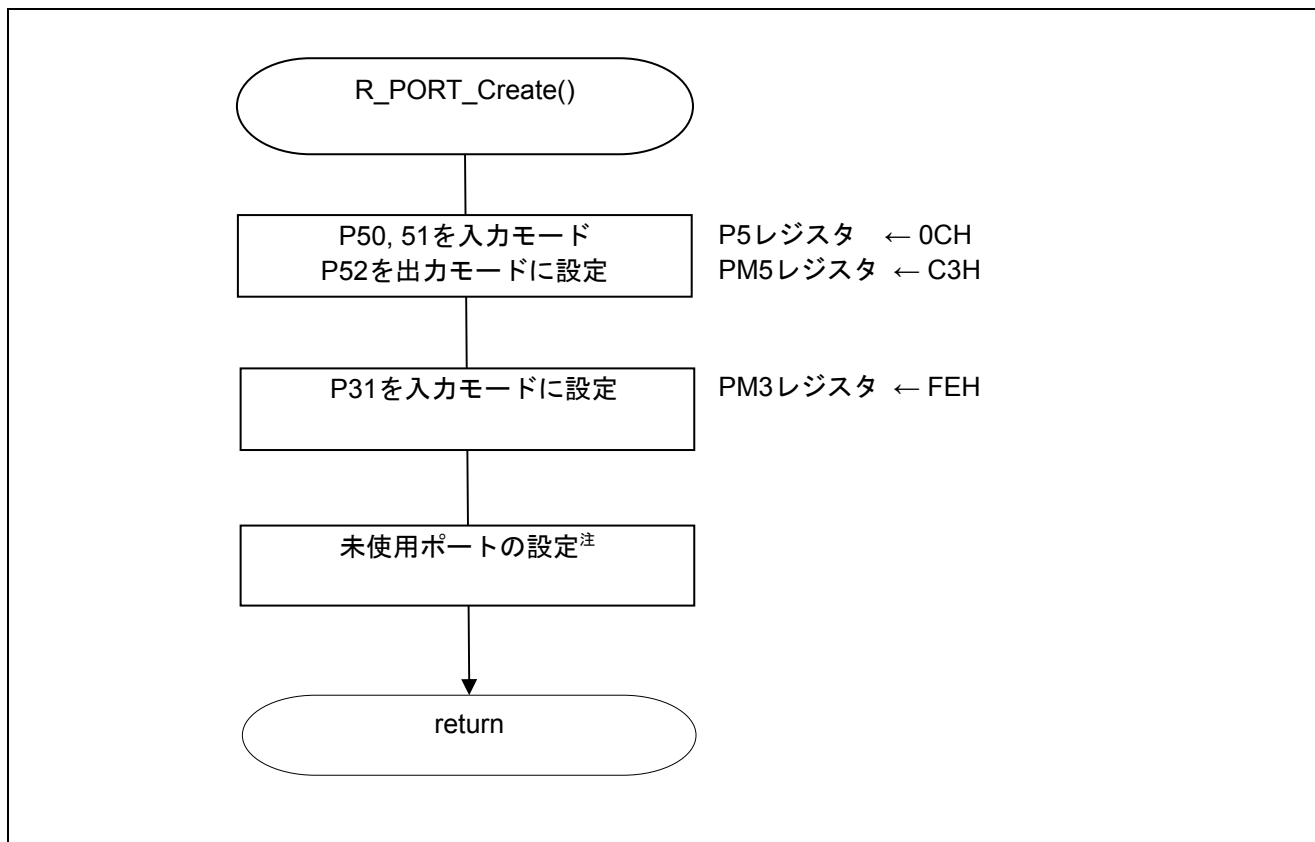


図 5.4 入出力ポートの設定

注 未使用ポートの設定については、RL78/G13 初期設定 (R01AN0451J) 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。

## 5.9.4 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。

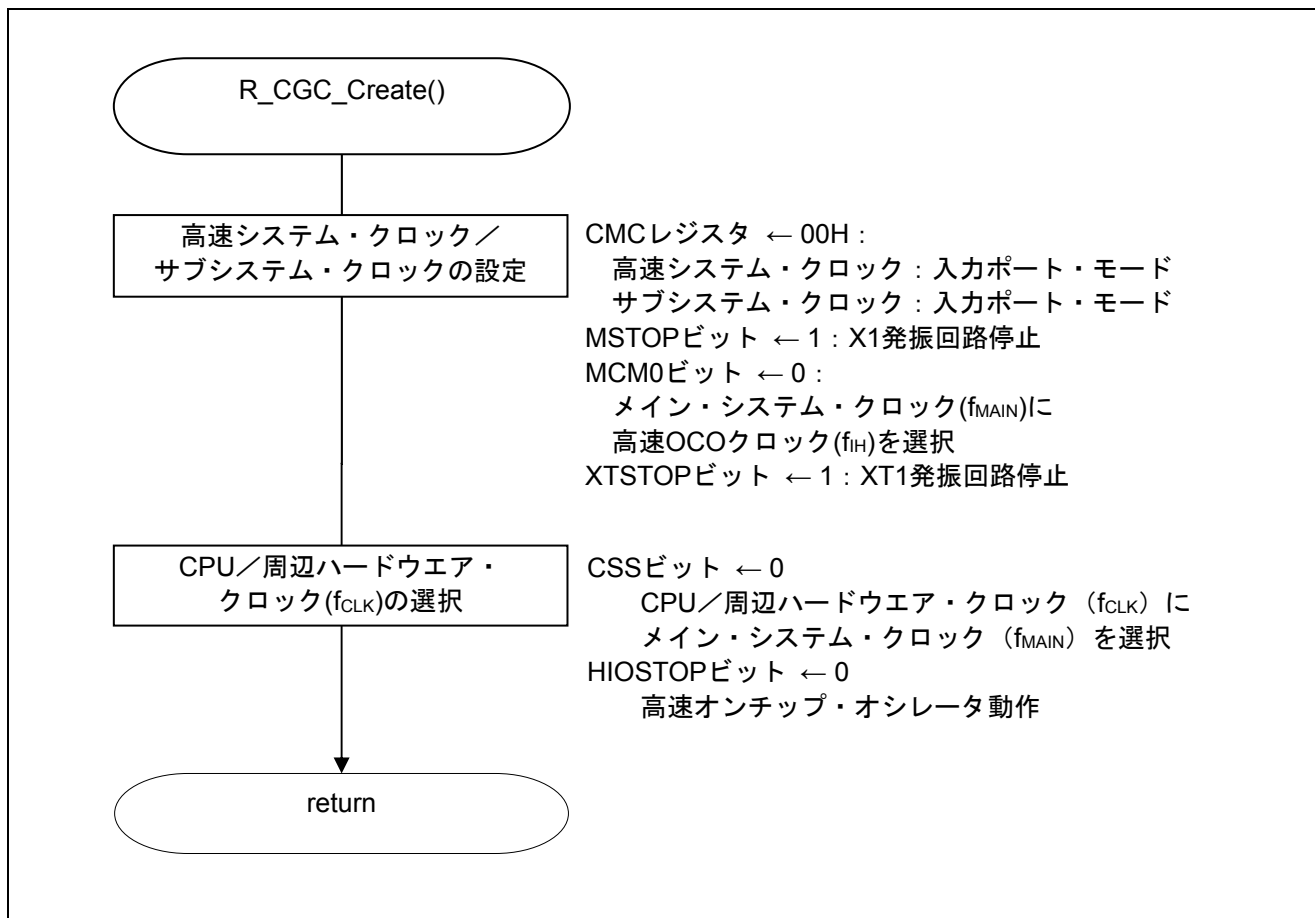


図 5.5 CPUクロックの設定

注意 CPUクロックの設定 (R\_CGC\_Create()) については、RL78/G13 初期設定 (R01AN0451J) 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。

## 5.9.5 TAU0 の設定

図 5.6 に TAU0 の設定のフローチャートを示します。

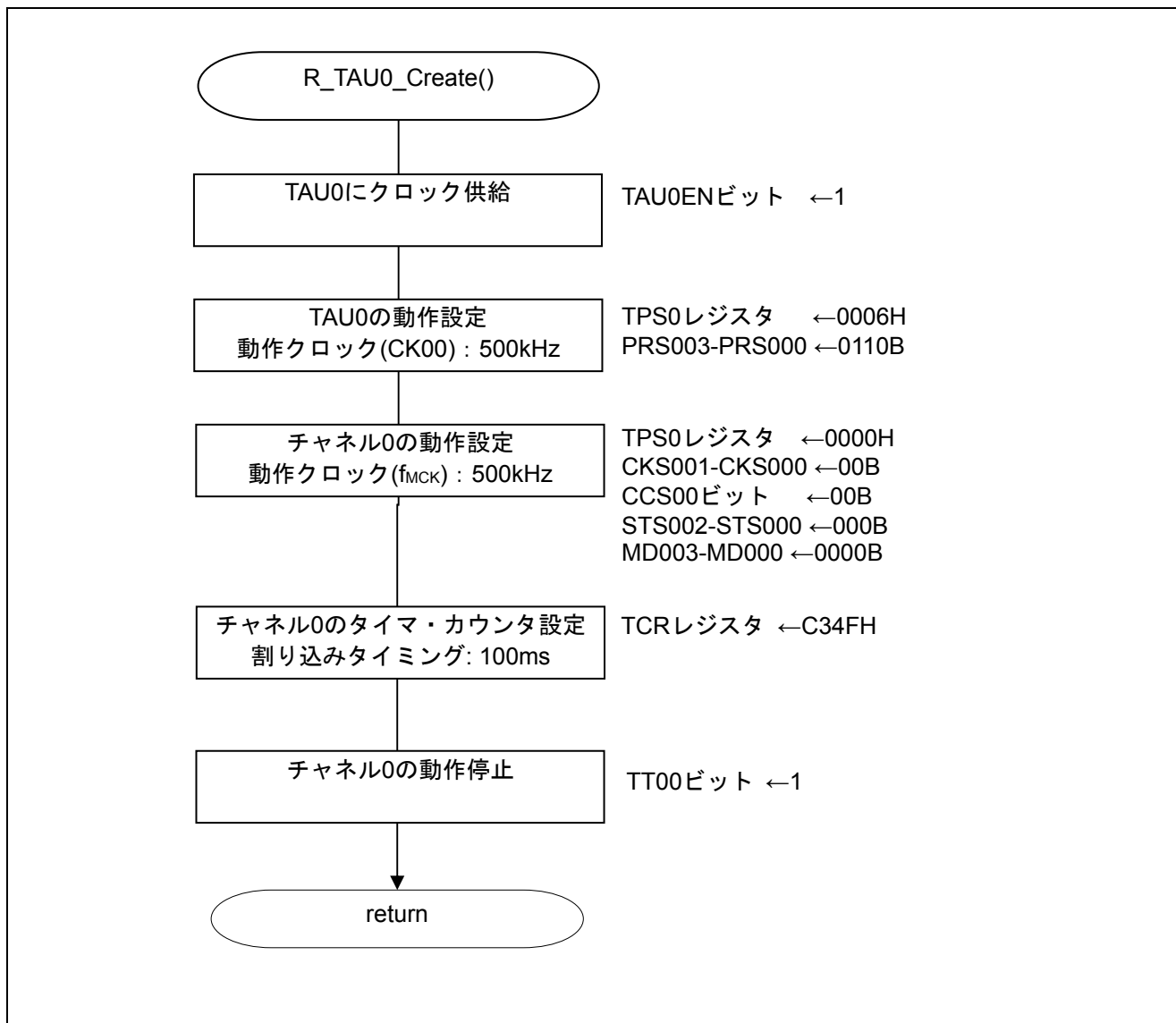


図 5.6 TAU0 の設定

## 5.9.6 インターバル・タイマの設定

図 5.7 にインターバル・タイマの設定のフローチャートを示します。

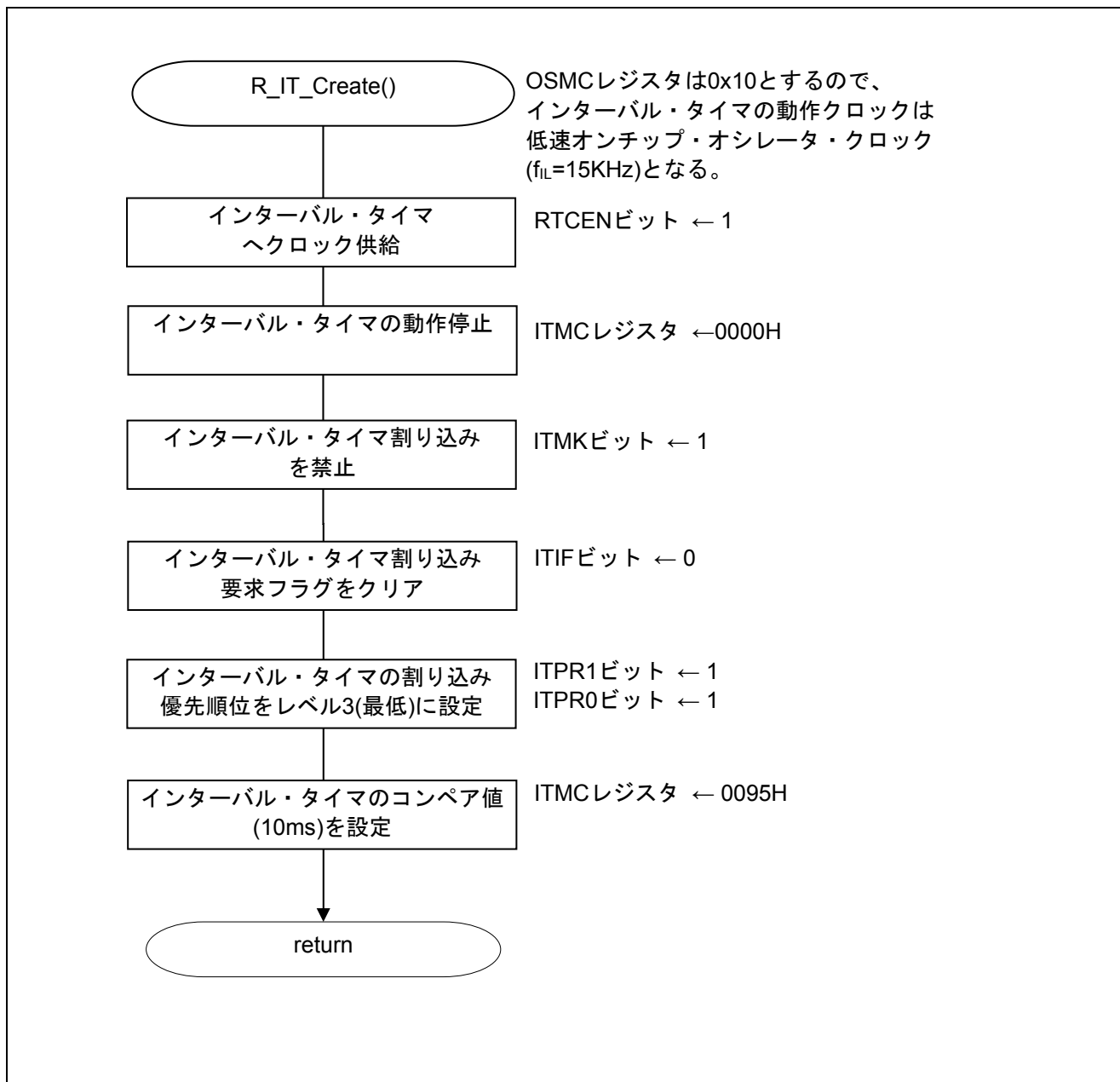


図 5.7 インターバル・タイマの設定

## 5.9.7 外部割り込み入力の設定

図 5.8 に外部割り込み入力の設定のフローチャートを示します。

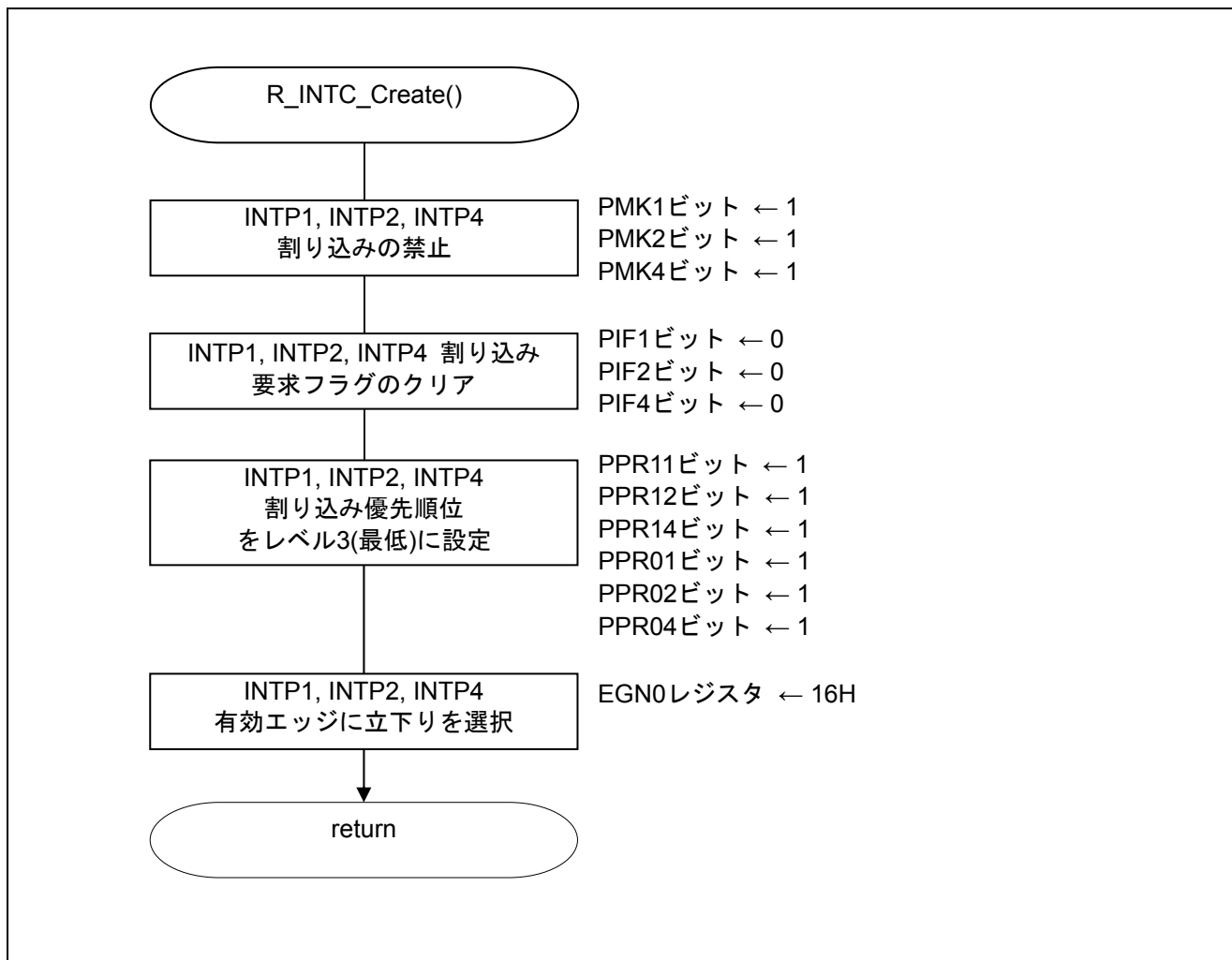


図 5.8 外部割り込み入力の設定

5.9.8 メイン処理

図 5.9 と図 5.10 にメイン処理のフローチャートを示します。

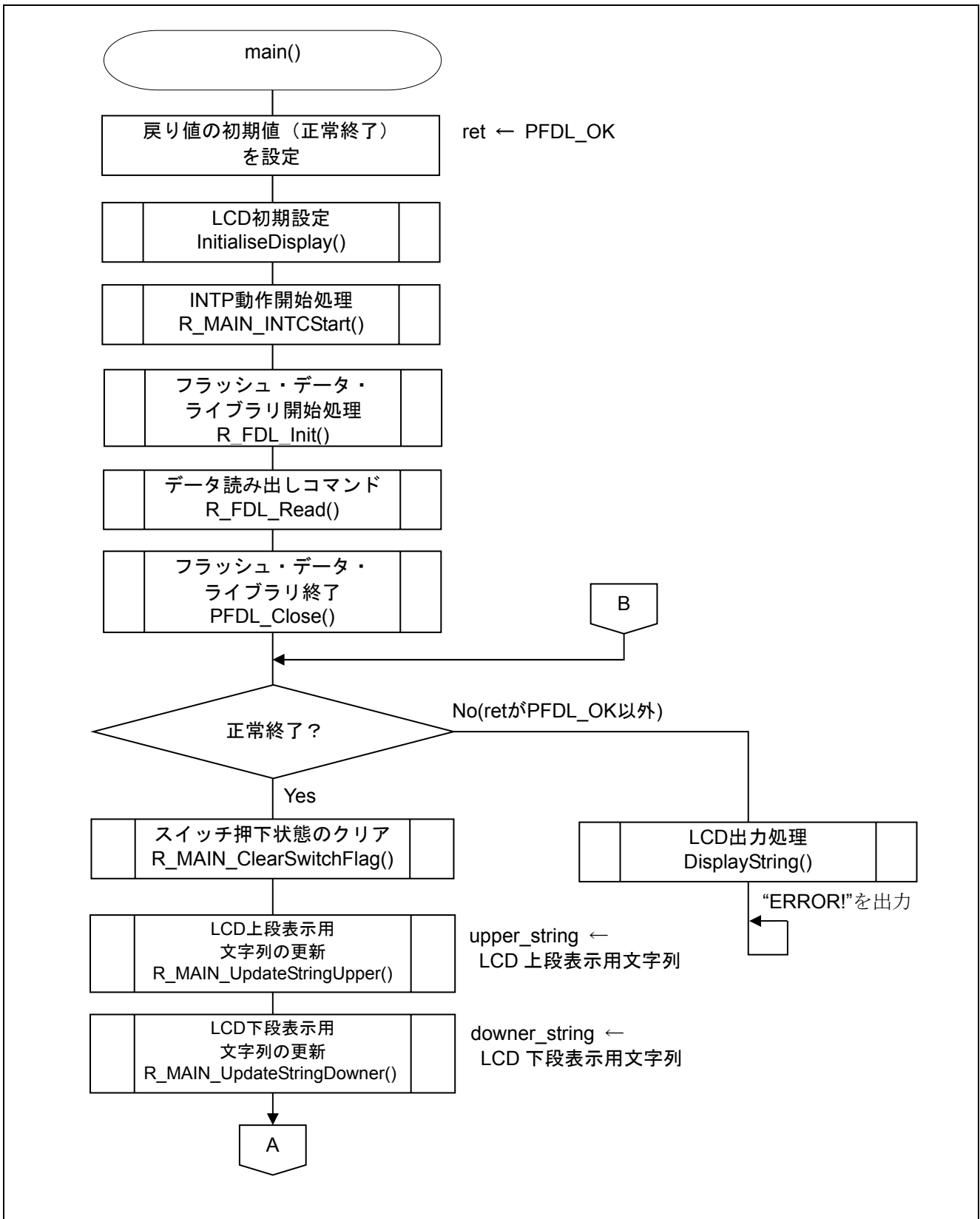


図 5.9 メイン処理(1/2)

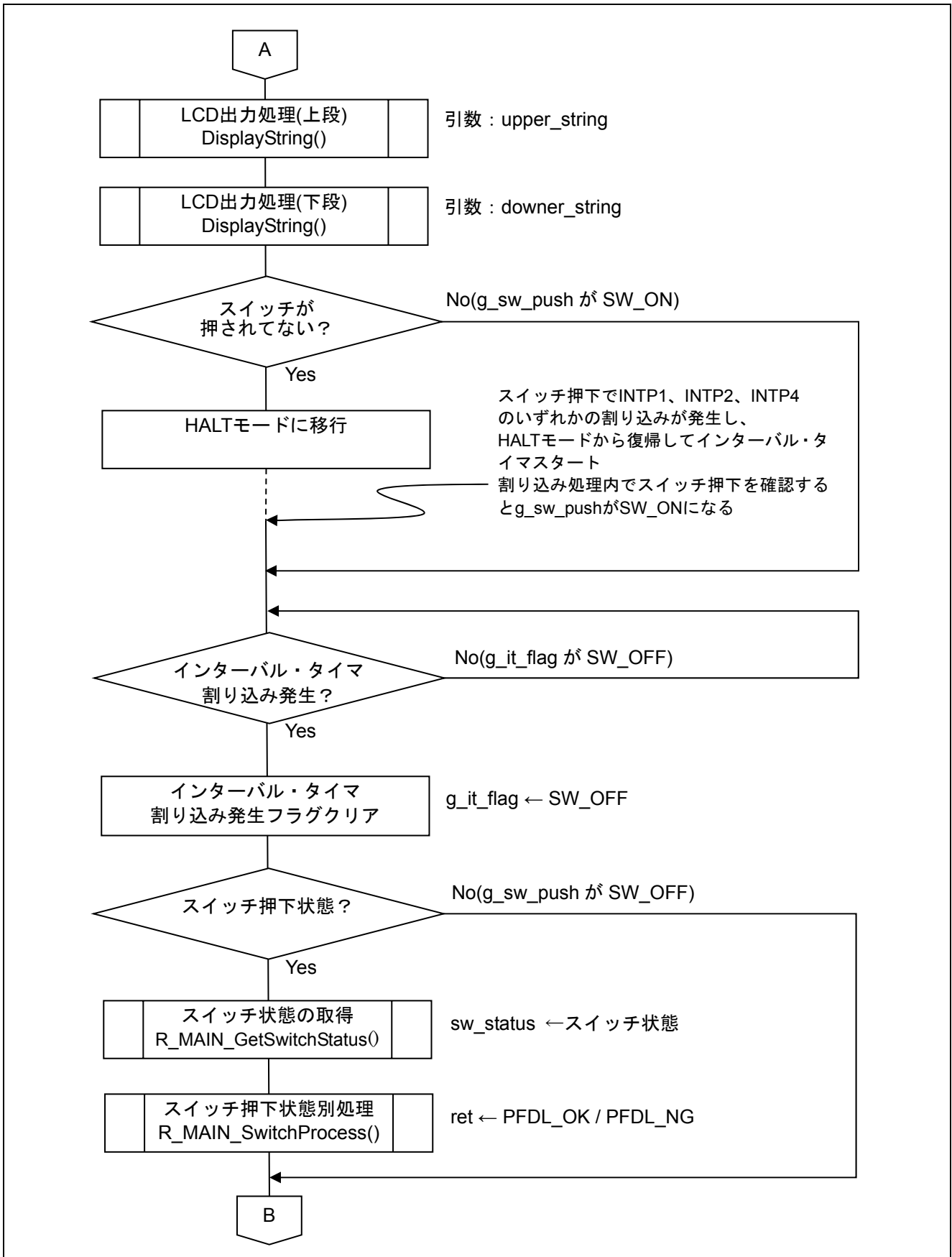


図 5.10 メイン処理(2/2)



## 5.9.9 INTP 動作開始処理

図 5.11 に INTP 動作開始処理のフローチャートを示します。

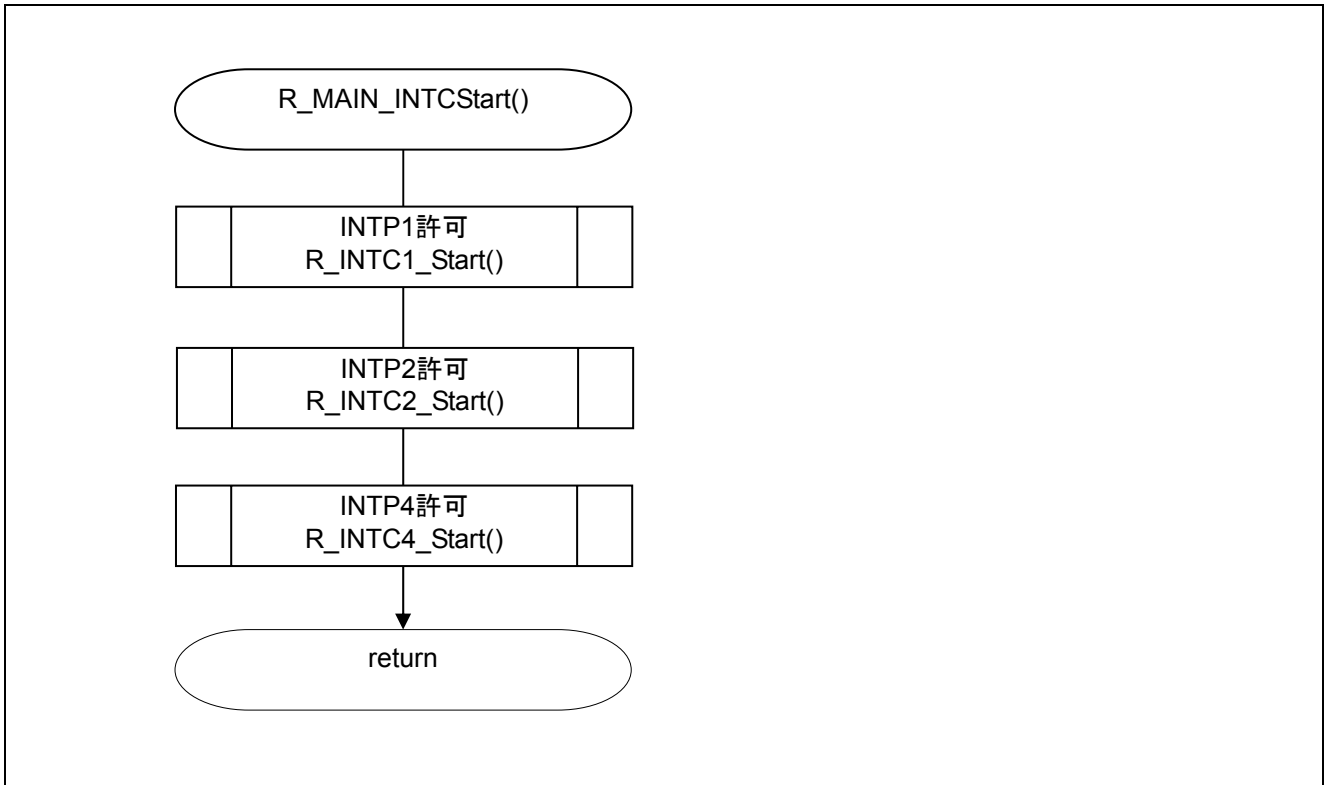


図 5.11 INTP 動作開始処理

## 5.9.10 INTP1 動作開始処理

図 5.12 に INTP1 動作開始処理のフローチャートを示します。

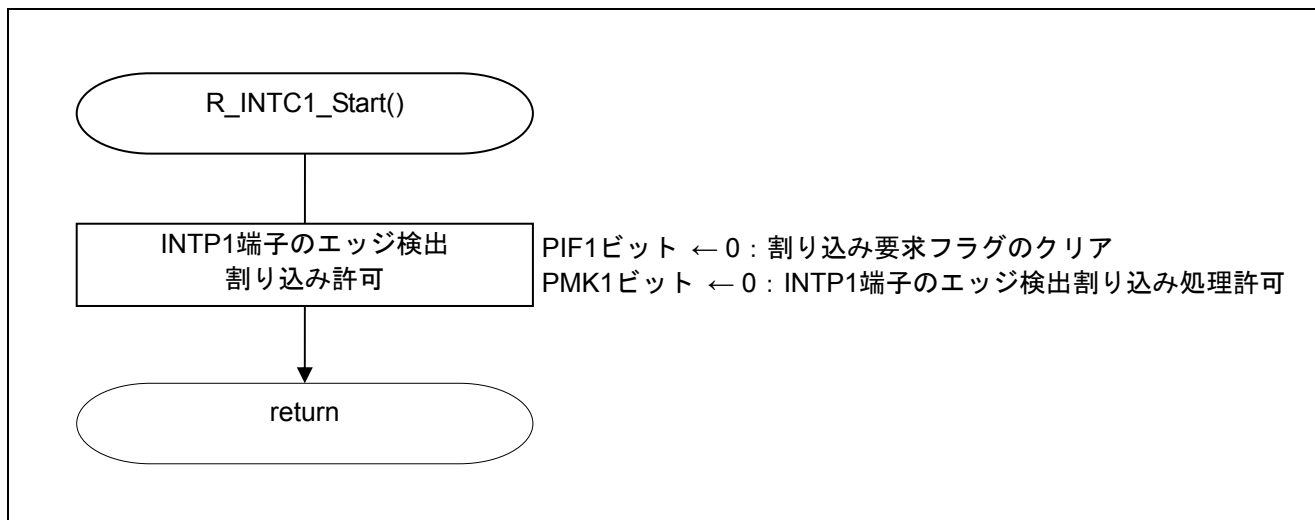


図 5.12 INTP1 動作開始処理

## 5.9.11 INTP1 外部割り込み

図 5.13 に INTP1 外部割り込みのフローチャートを示します。

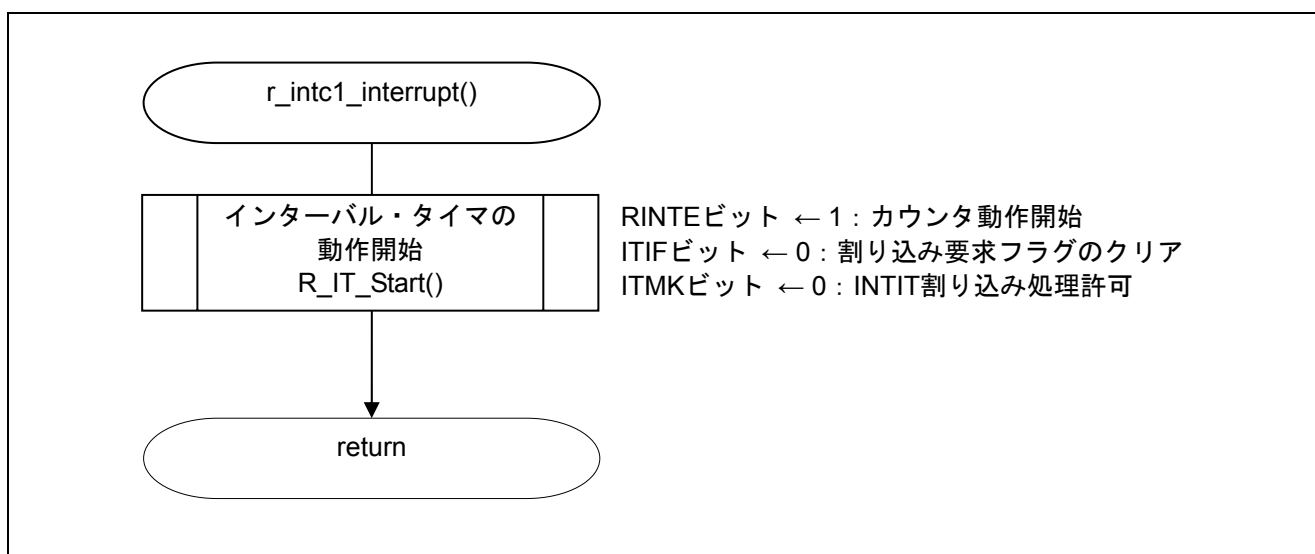


図 5.13 INTP1 外部割り込み

## 5.9.12 INTP2 動作開始処理

図 5.14 に INTP2 動作開始処理のフローチャートを示します。

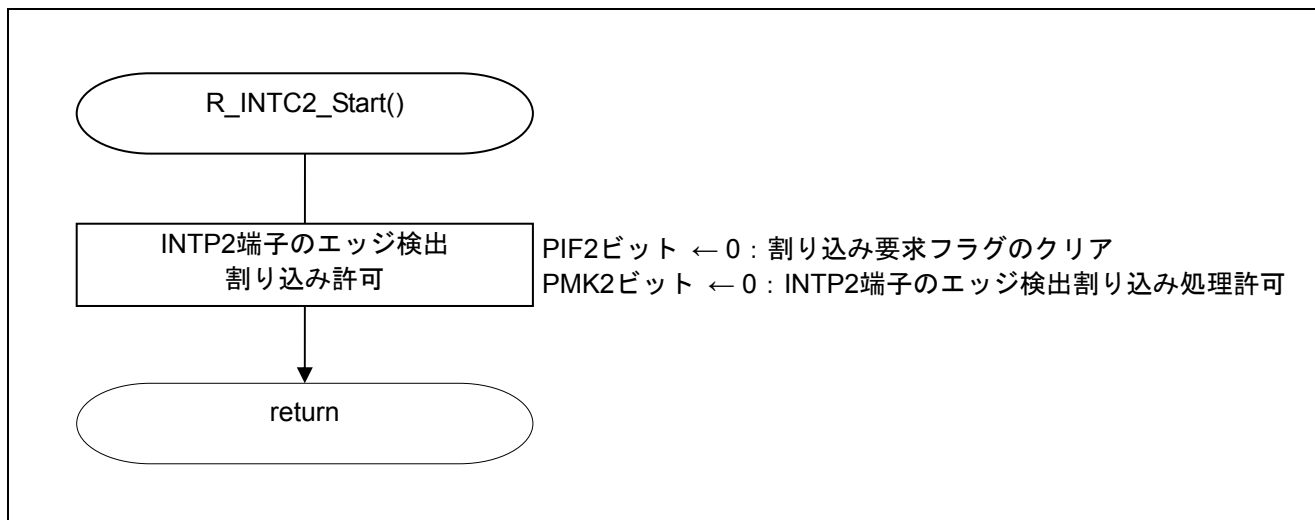


図 5.14 INTP2 動作開始処理

## 5.9.13 INTP2 外部割り込み

図 5.15 に INTP2 外部割り込みのフローチャートを示します。

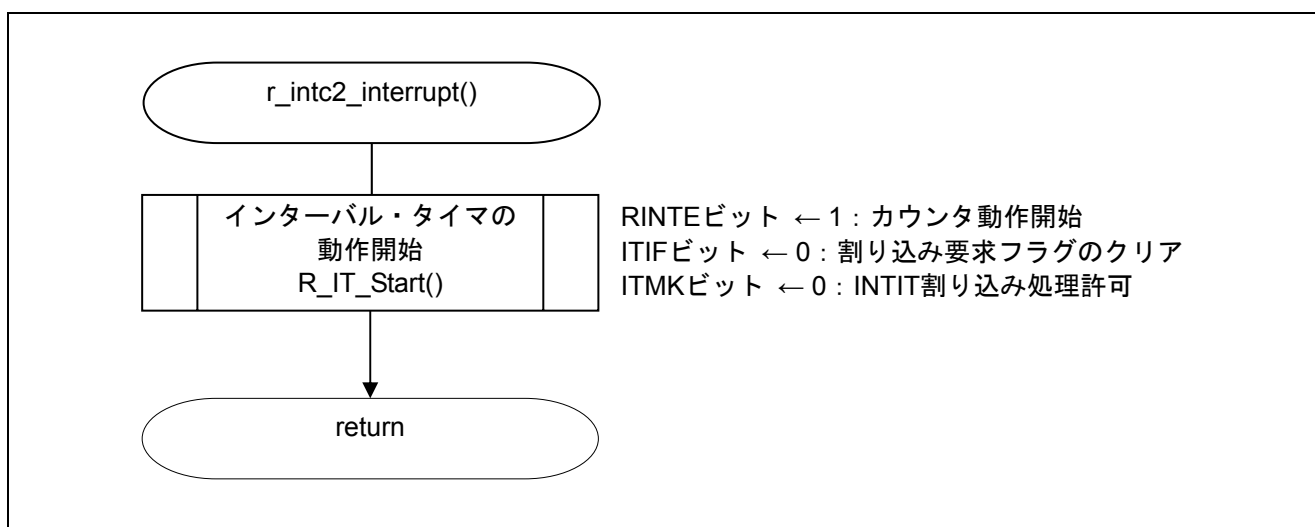


図 5.15 INTP2 外部割り込み

## 5.9.14 INTP4 動作開始処理

図 5.16 に INTP4 動作開始処理のフローチャートを示します。

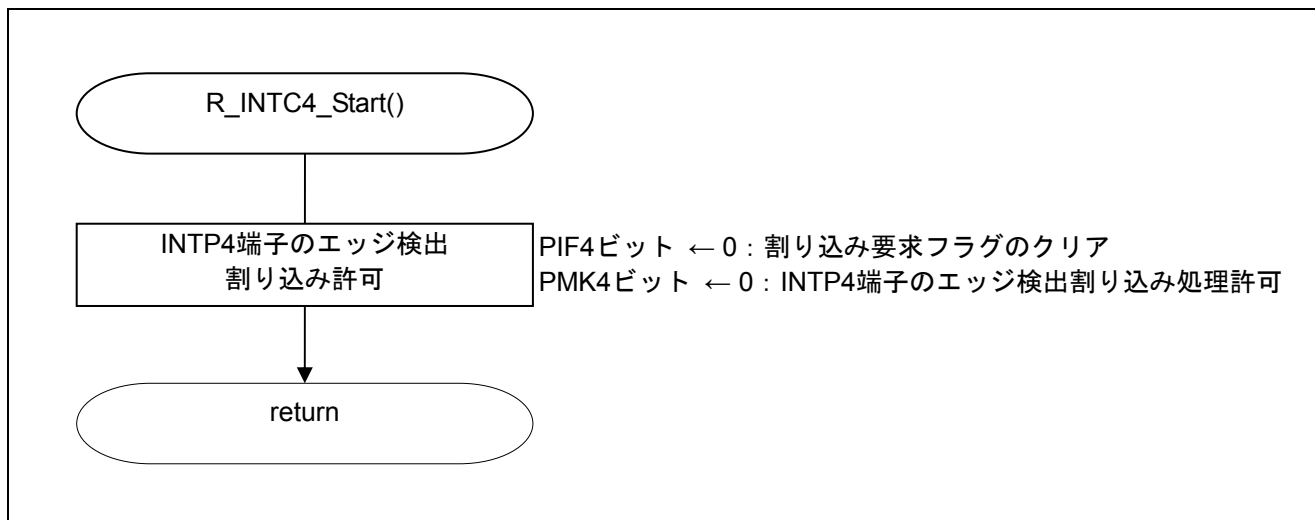


図 5.16 INTP4 動作開始処理

## 5.9.15 INTP4 外部割り込み

図 5.17 に INTP4 外部割り込みのフローチャートを示します。

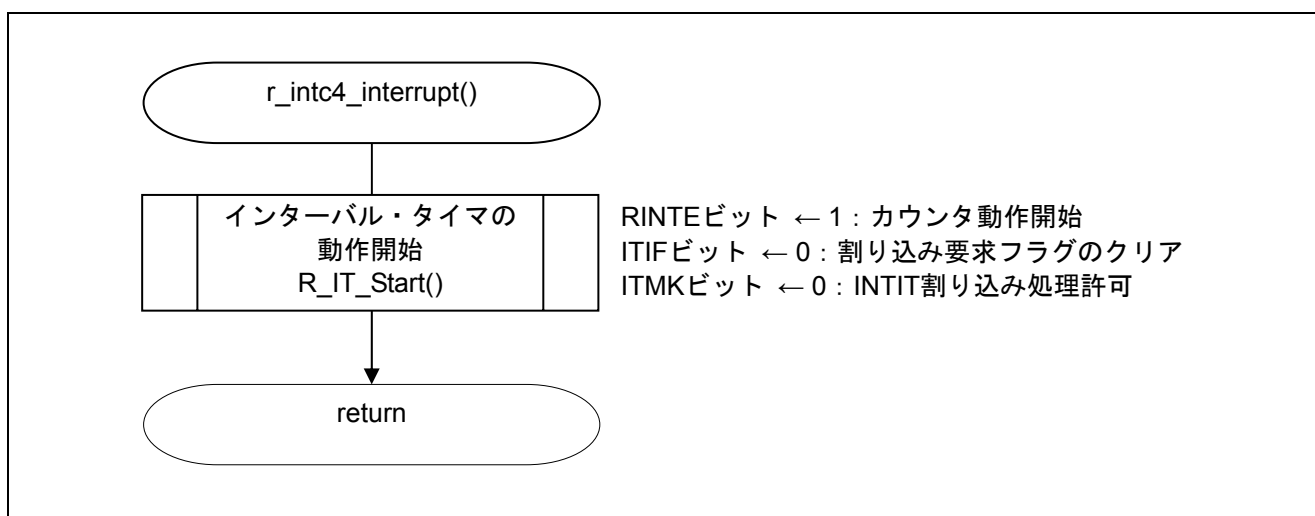


図 5.17 INTP4 外部割り込み

## 5.9.16 インターバル・タイマ動作開始処理

図 5.18 にインターバル・タイマ動作開始処理のフローチャートを示します。

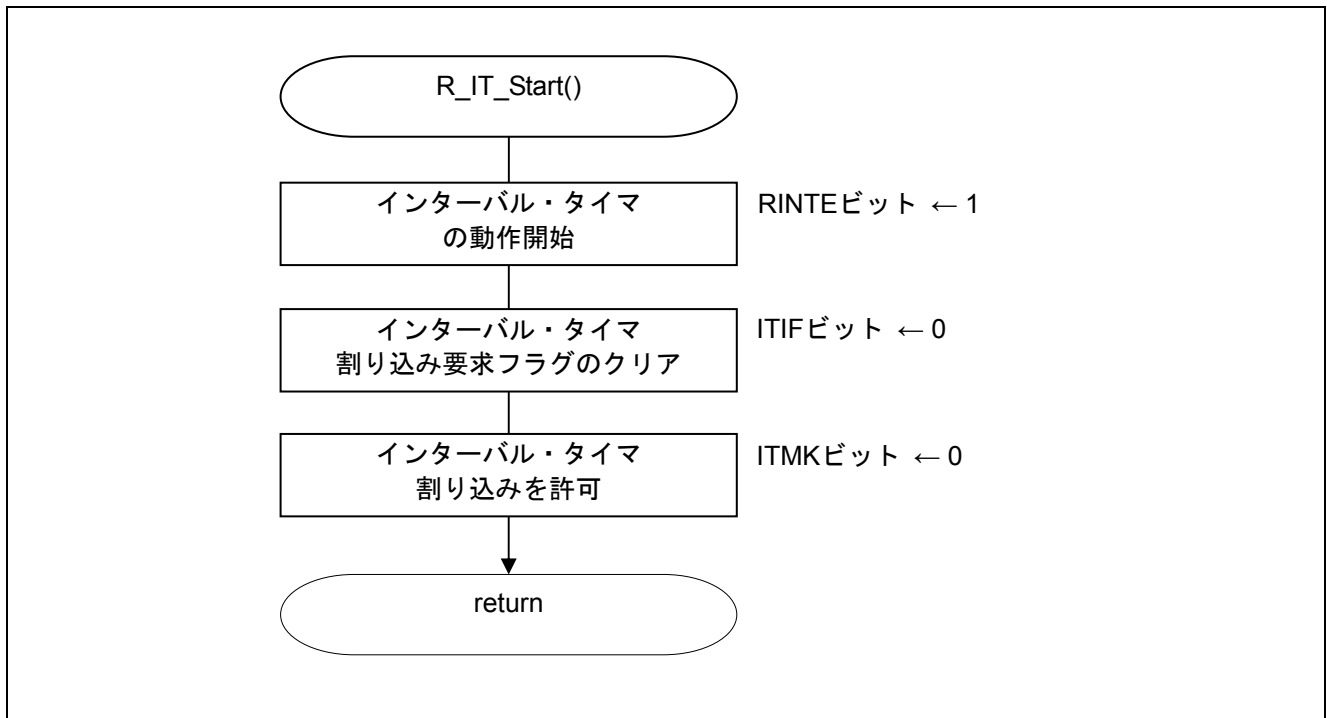


図 5.18 インターバル・タイマ動作開始処理

5.9.17 インターバル・タイマ割り込み

図 5.19 にインターバル・タイマ割り込みのフローチャートを示します。

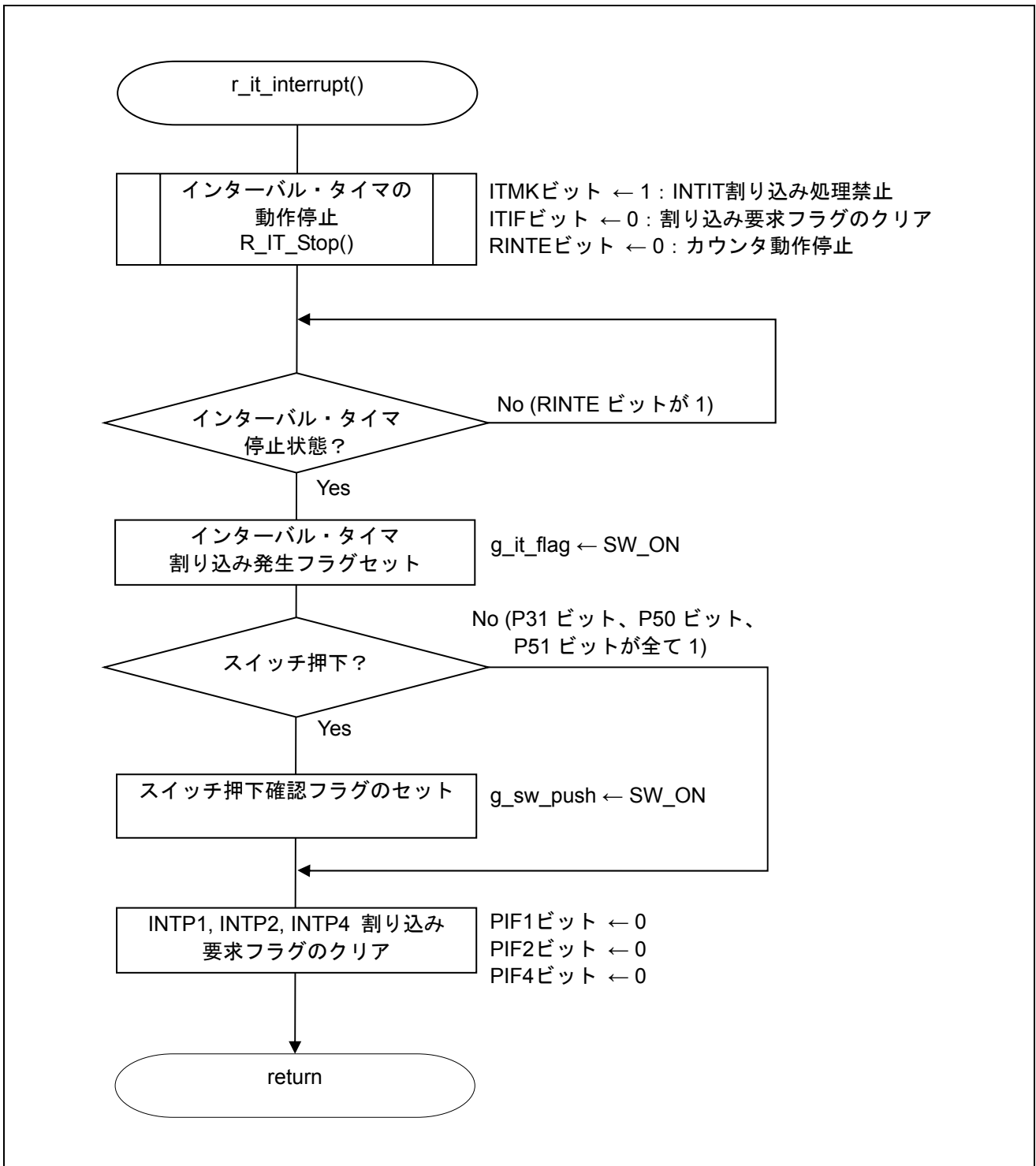


図 5.19 インターバル・タイマ割り込み

## 5.9.18 インターバル・タイマ動作停止処理

図 5.20 にインターバル・タイマ動作停止処理のフローチャートを示します。

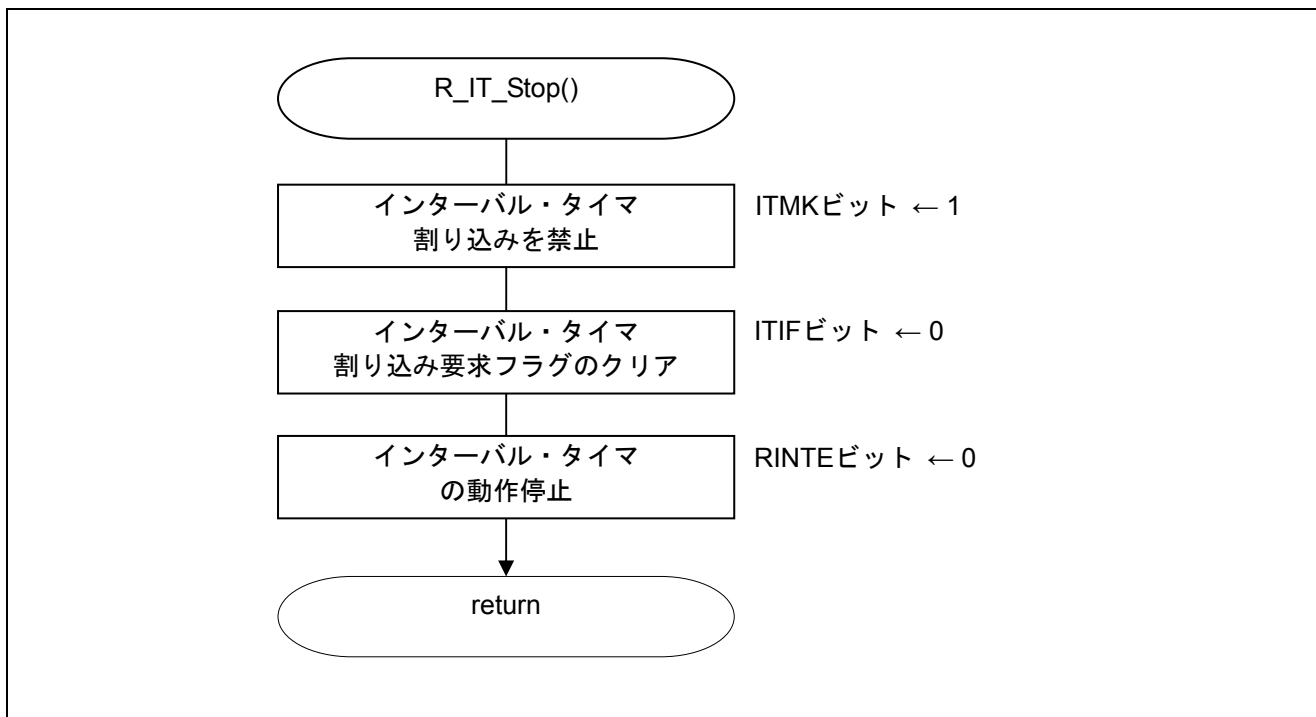


図 5.20 インターバル・タイマ動作停止処理

## 5.9.19 フラッシュ・データ・ライブラリ開始処理

図 5.21 にフラッシュ・データ・ライブラリ開始処理のフローチャートを示します。

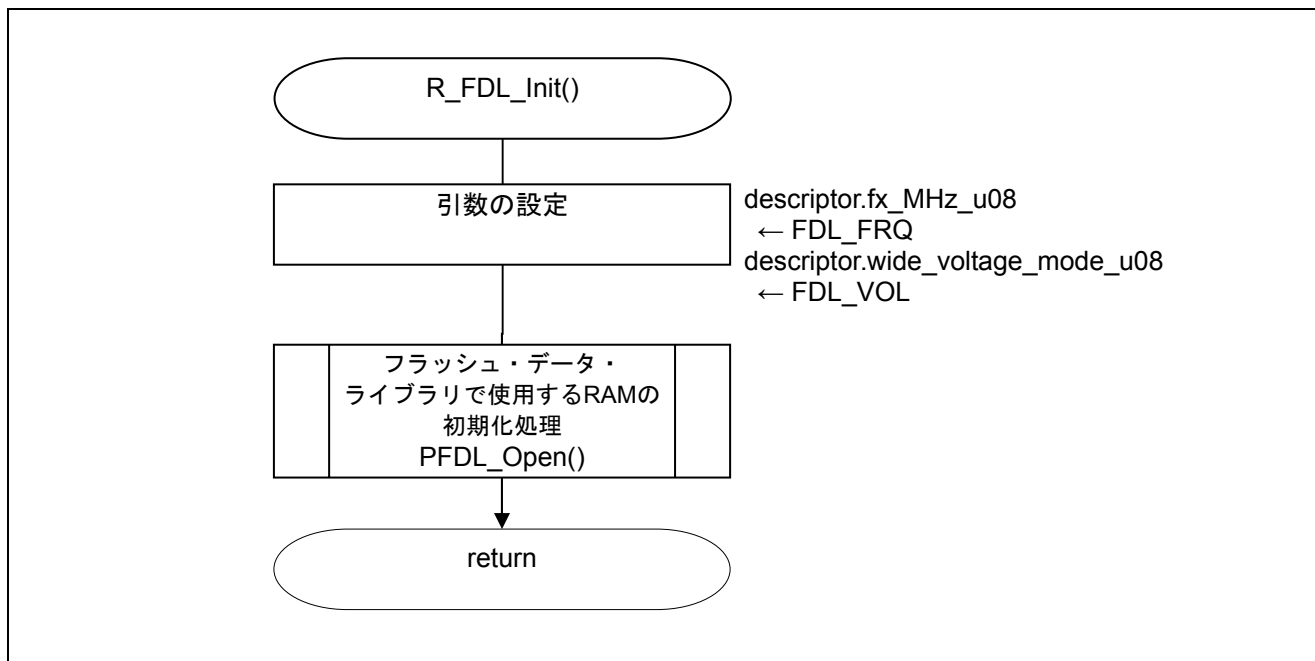


図 5.21 フラッシュ・データ・ライブラリ開始処理



5.9.20 データ読み出しコマンド処理

図 5.22 にデータ読み出しコマンド処理のフローチャートを示します。

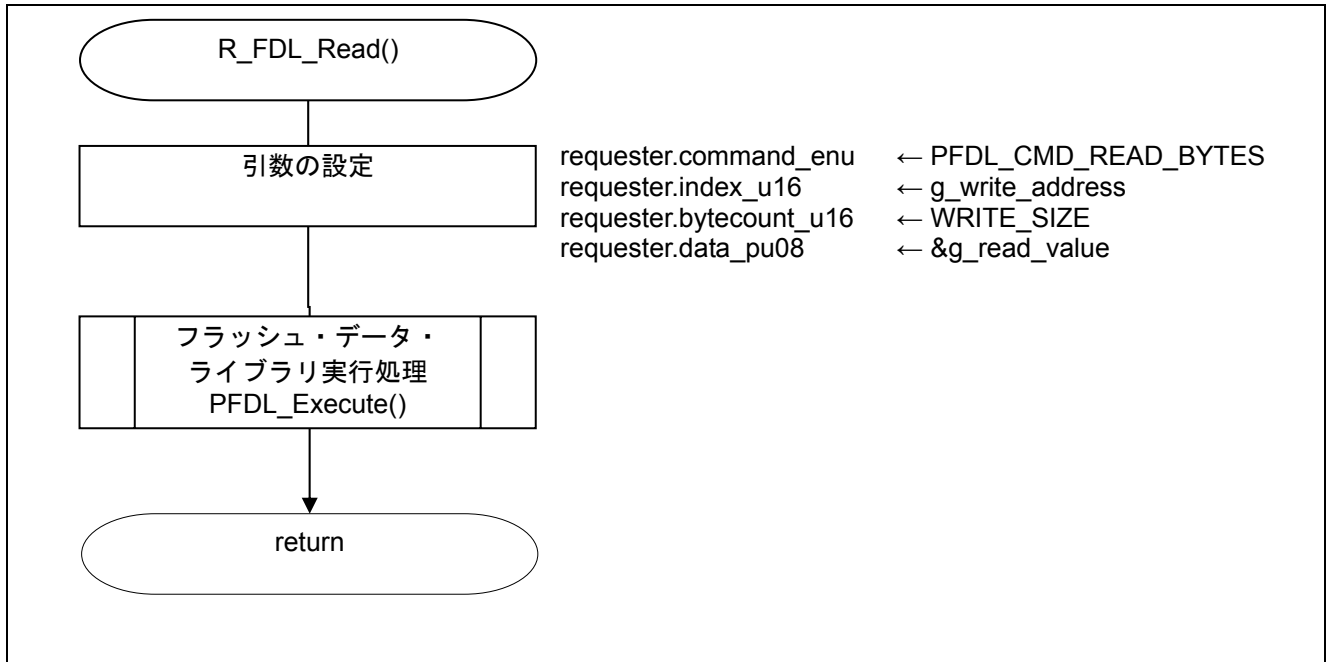


図 5.22 データ読み出しコマンド処理

## 5.9.21 スイッチ押下状態のクリア

図 5.23 にスイッチ押下状態のクリアのフローチャートを示します。

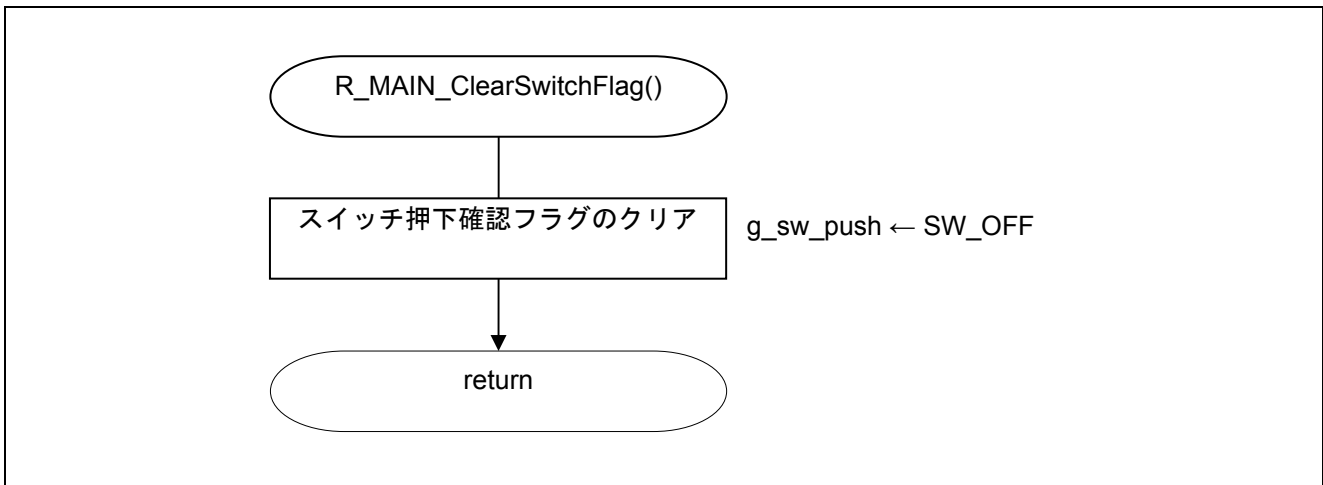


図 5.23 スイッチ押下状態のクリア

## 5.9.22 LCD 上段に表示する文字列の更新処理

図 5.24 に LCD 上段に表示する文字列の更新処理のフローチャートを示します。

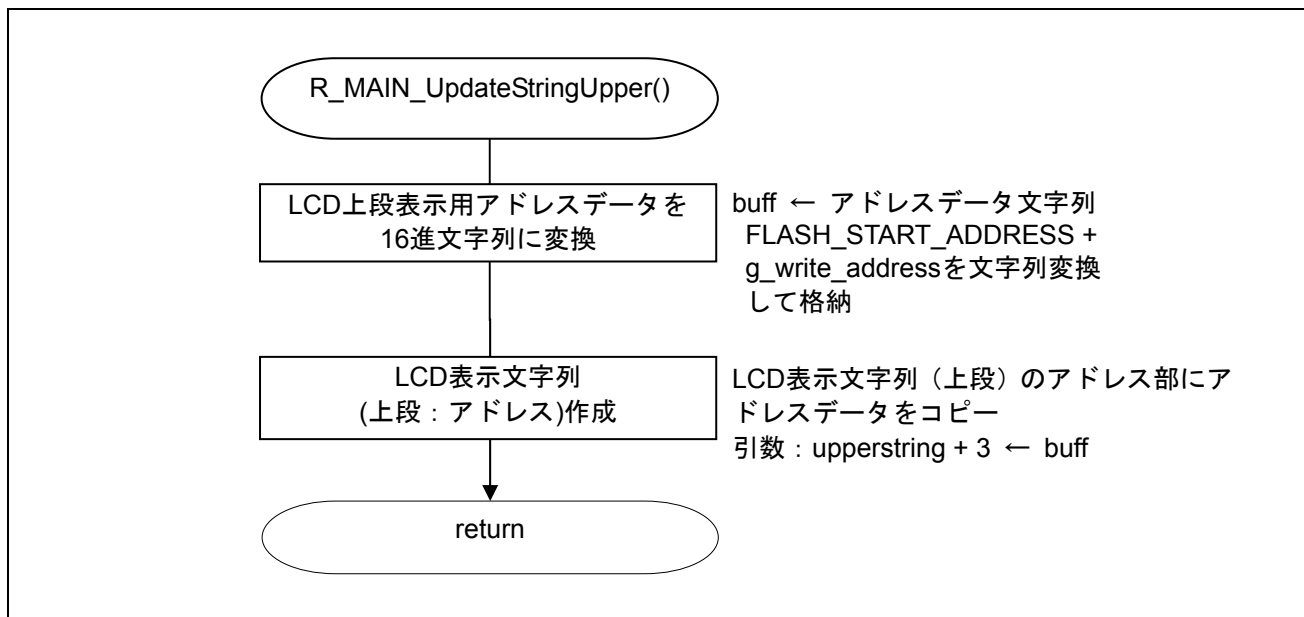


図 5.24 LCD 上段に表示する文字列の更新処理

5.9.23 LCD 下段に表示する文字列の更新処理

図 5.25 に LCD 下段に表示する文字列の更新処理のフローチャートを示します。

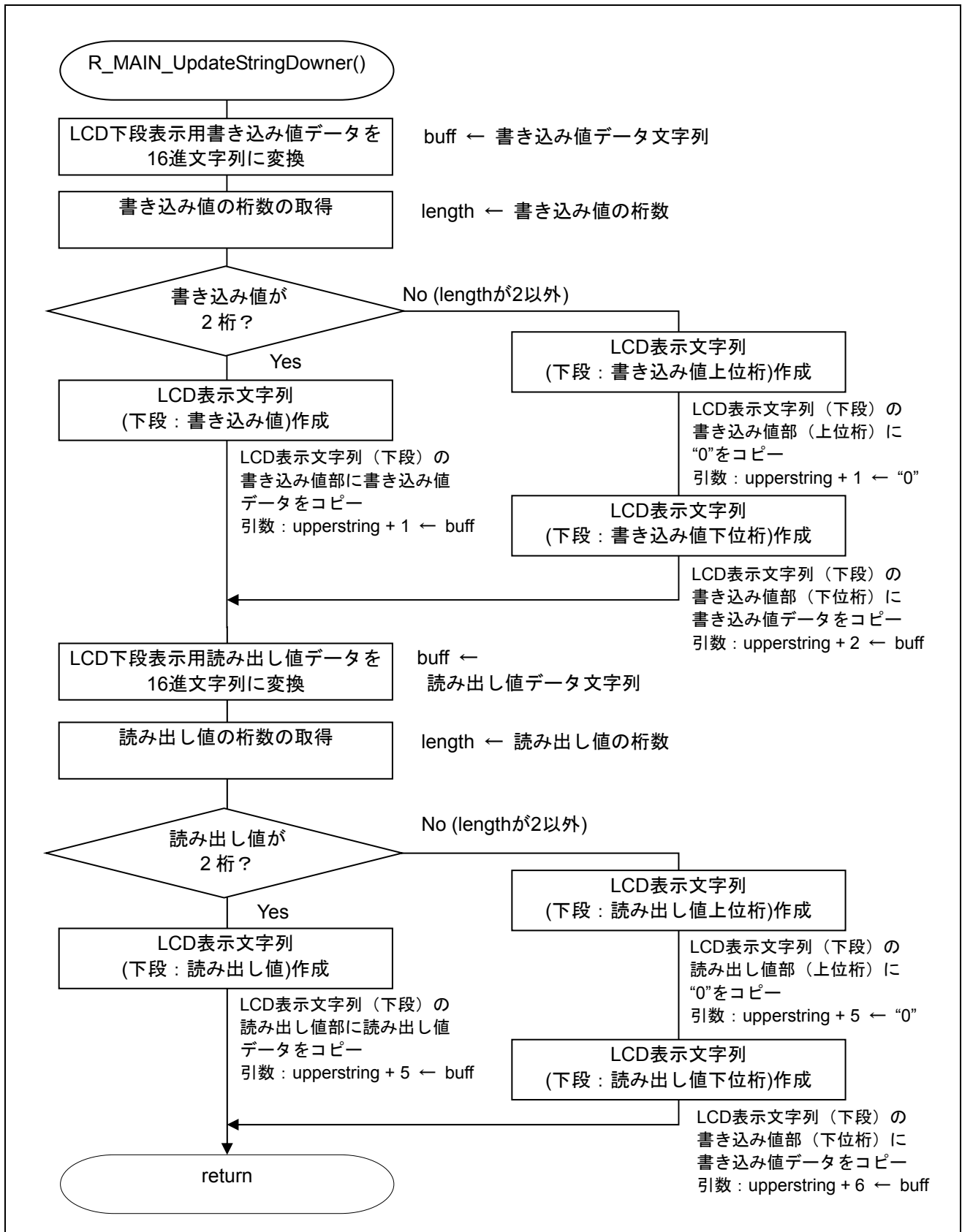


図 5.25 LCD 下段に表示する文字列の更新処理

## 5.9.24 スイッチ状態の取得

図 5.26 にスイッチ状態の取得のフローチャートを示します。

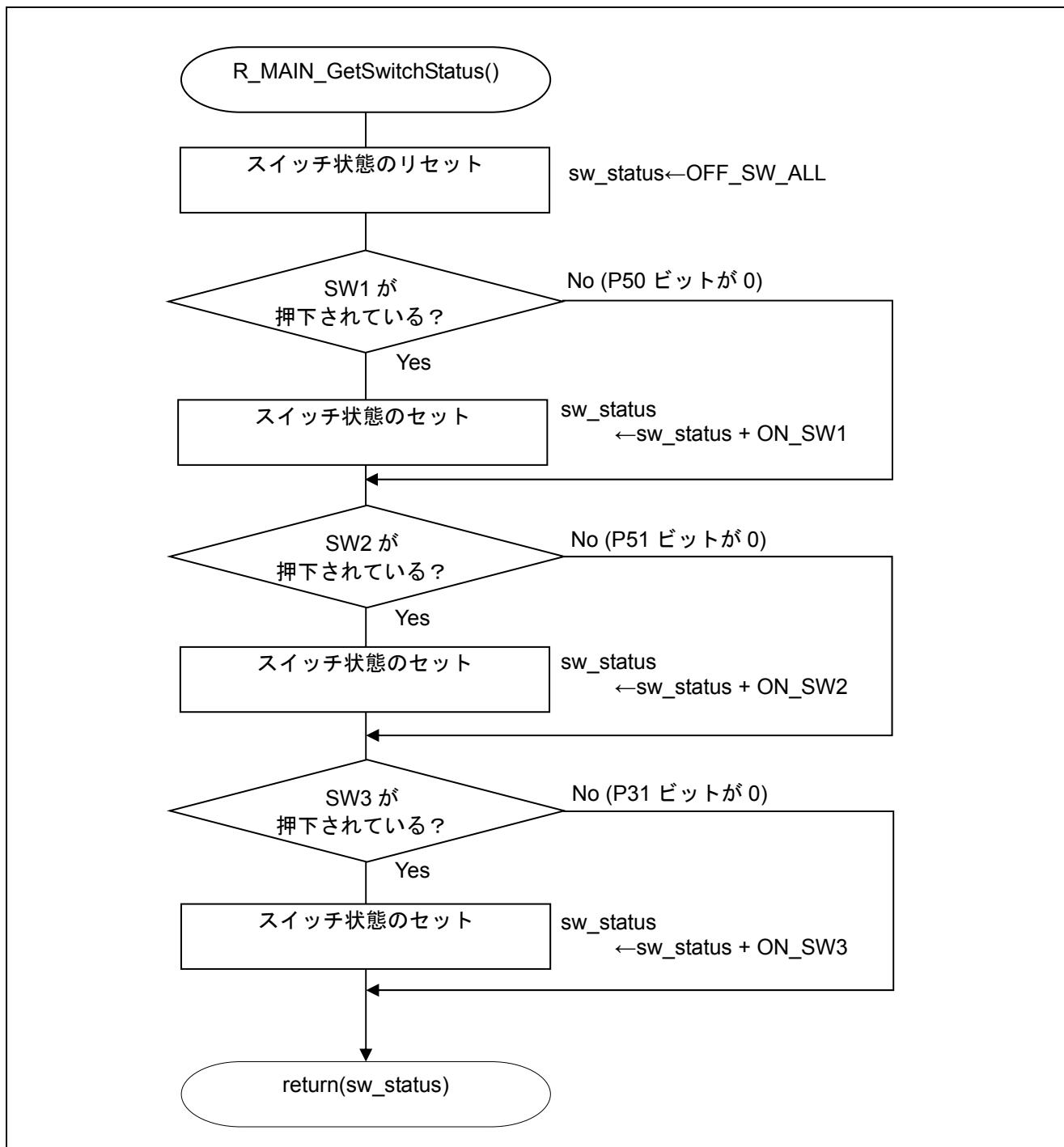


図 5.26 スイッチ状態の取得

5.9.25 スイッチ押下状態別処理

図 5.27 と図 5.28 にスイッチ押下状態別処理のフローチャートを示します。

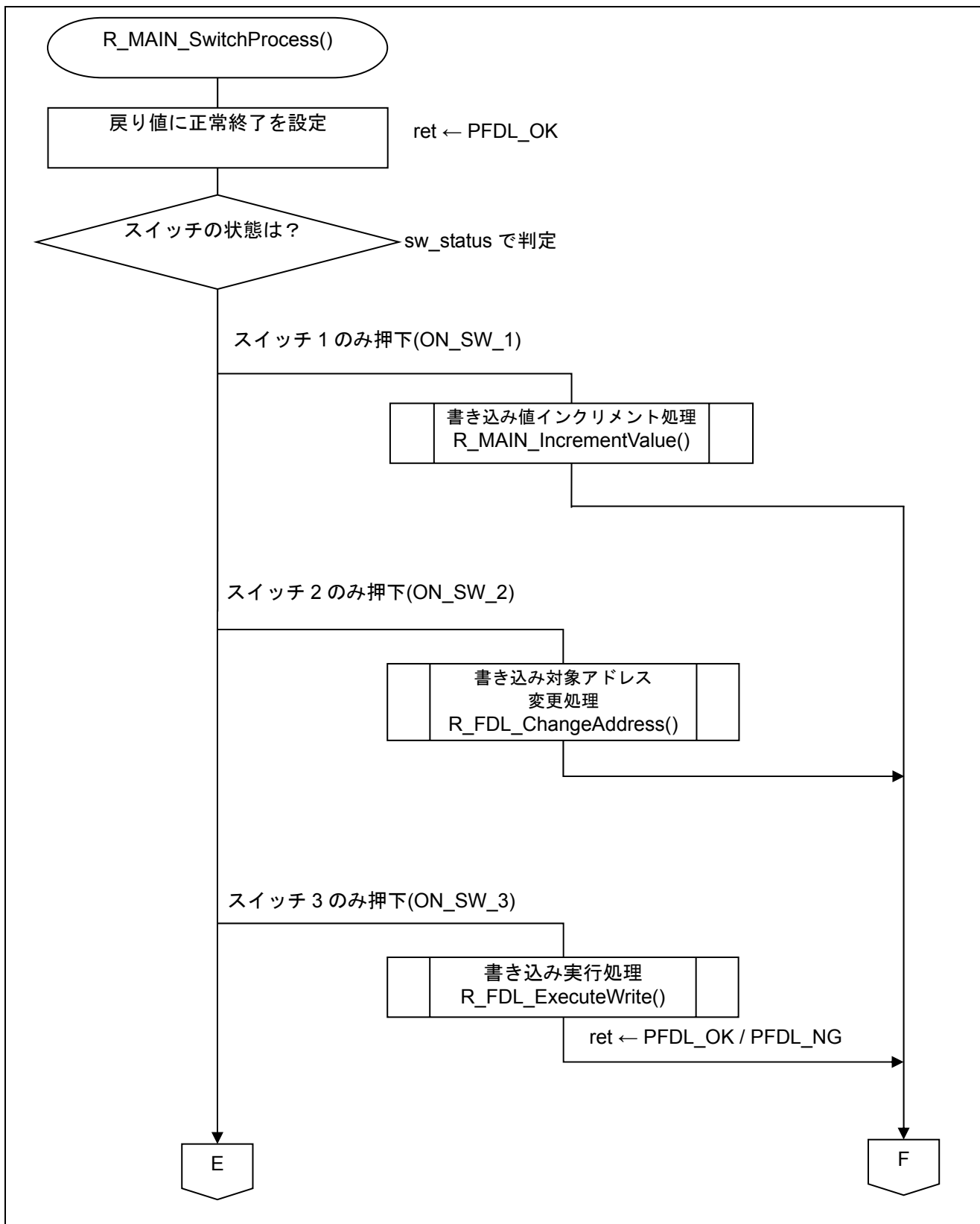


図 5.27 スイッチ押下状態別処理(1/2)

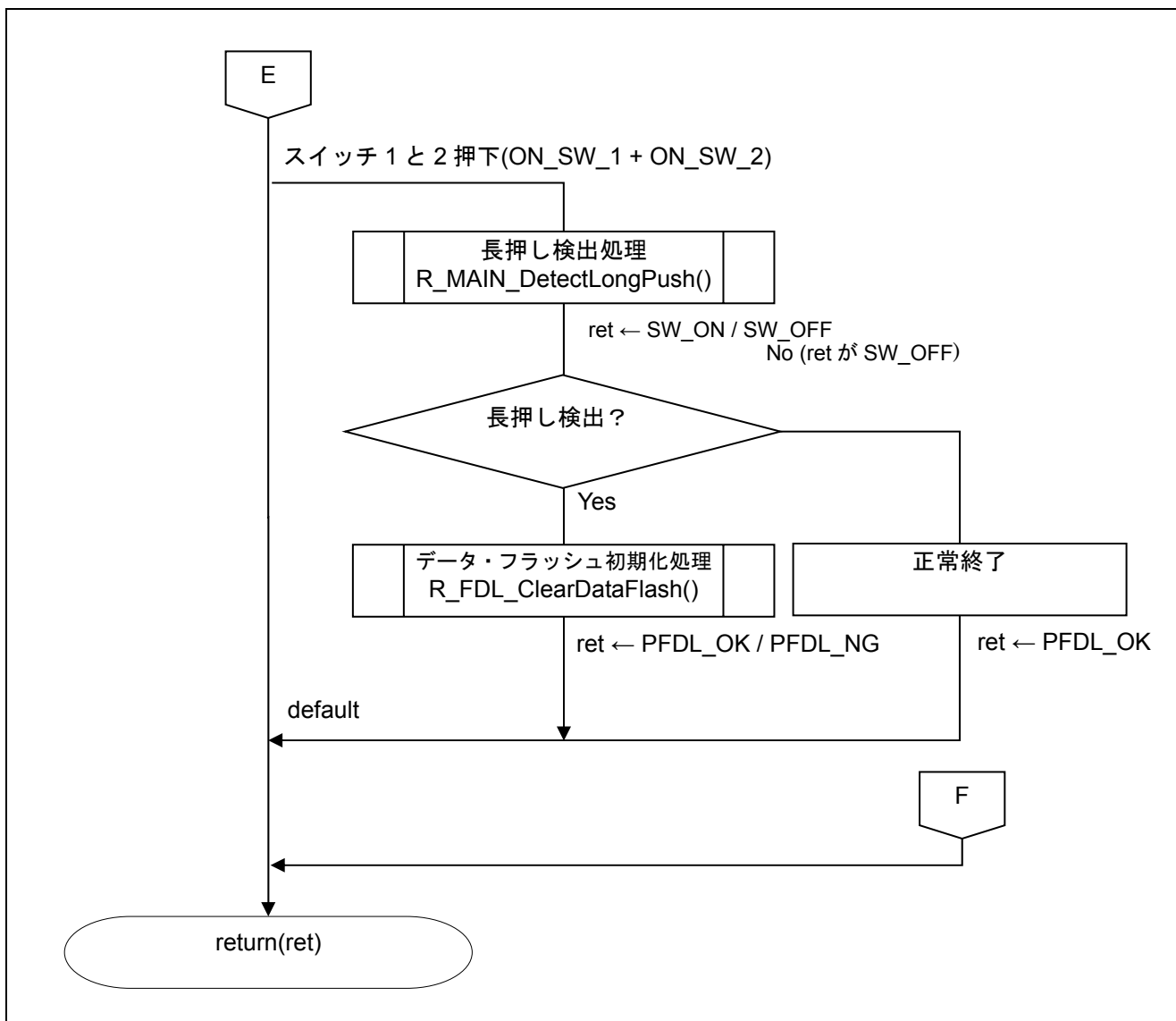


図 5.28 スイッチ押下状態別処理(2/2)

## 5.9.26 書き込み値インクリメント処理

図 5.29 に書き込み値インクリメント処理のフローチャートを示します。

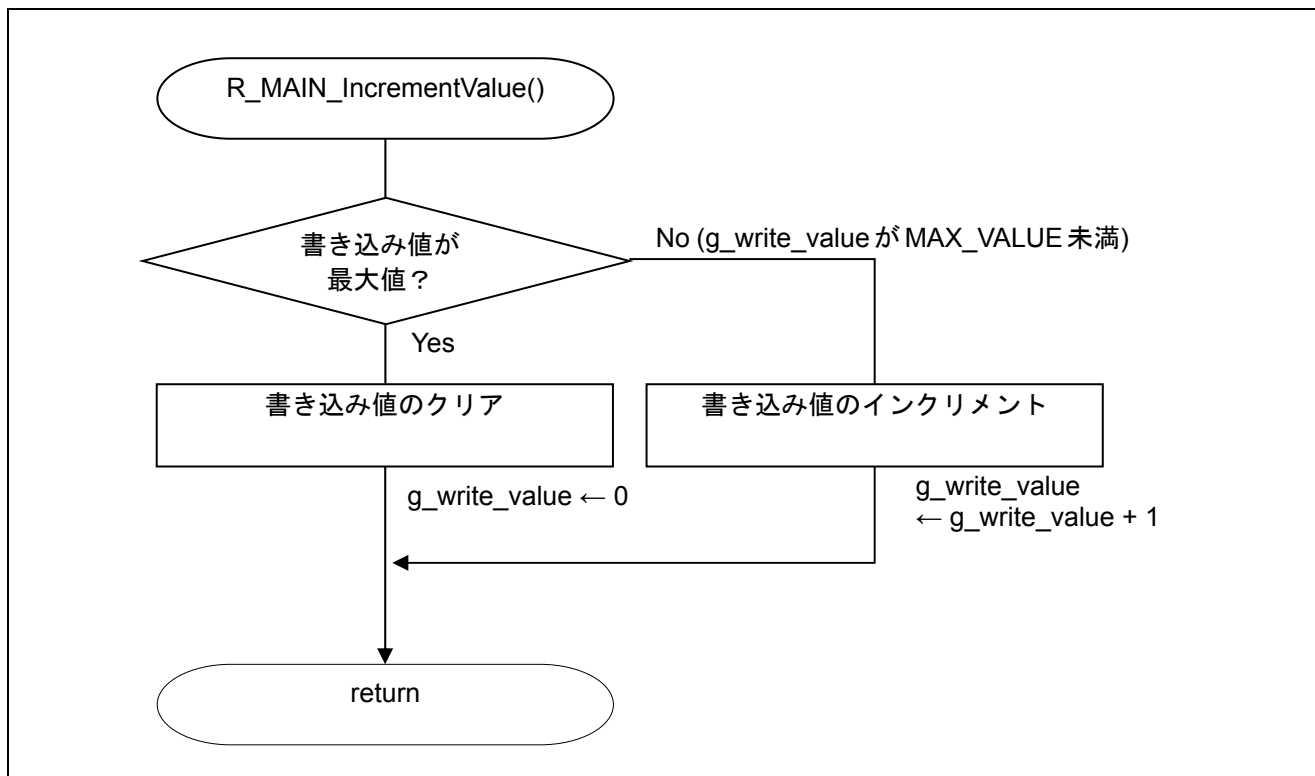


図 5.29 書き込み値インクリメント処理



5.9.27 書き込み対象アドレス変更処理

図 5.30 に書き込み対象アドレス変更処理のフローチャートを示します。

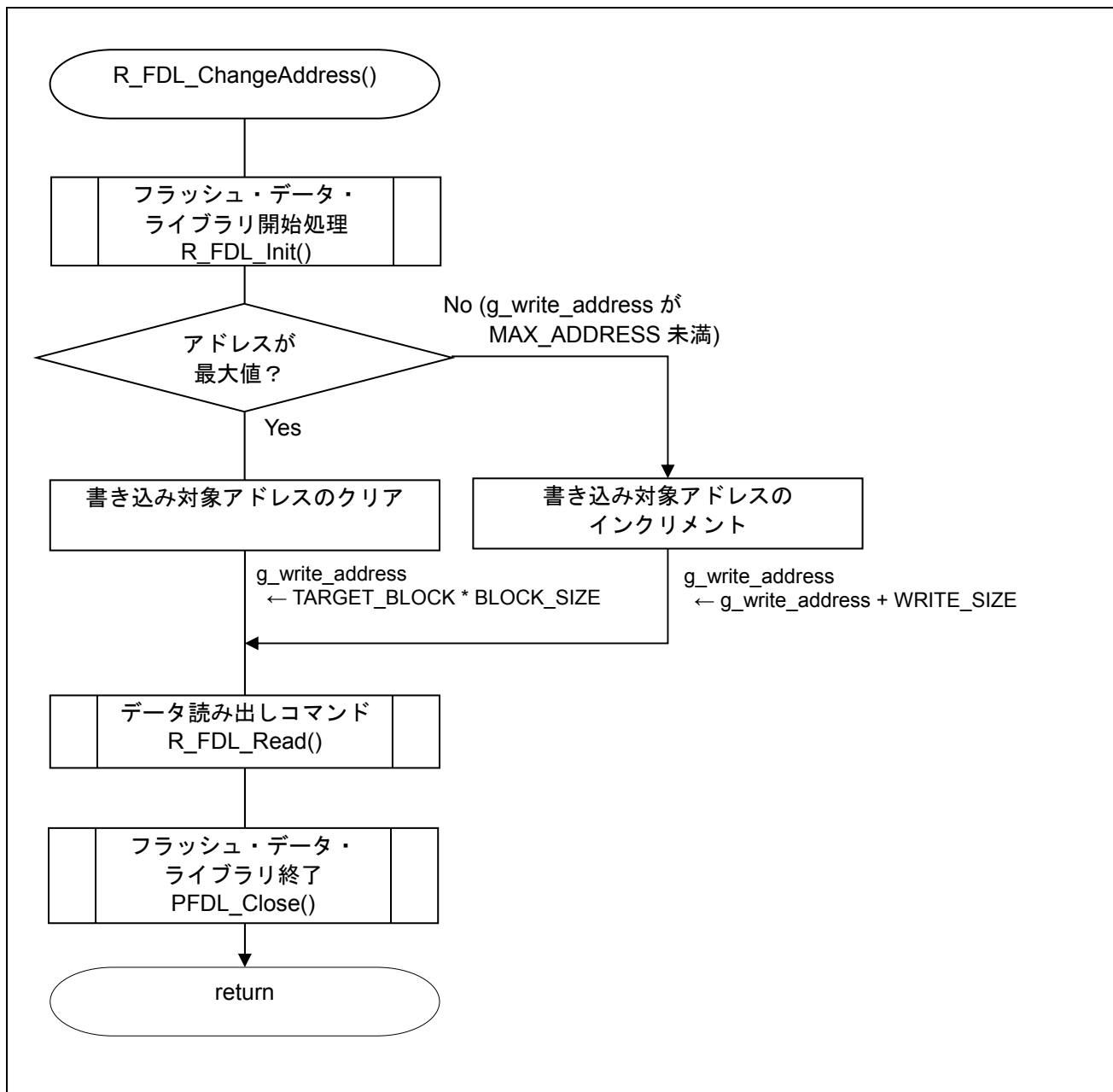


図 5.30 書き込み対象アドレス変更処理

5.9.28 書き込み実行処理

図 5.31 に書き込み実行処理(1/2)、図 5.32 に書き込み実行処理(2/2)のフローチャートを示します。

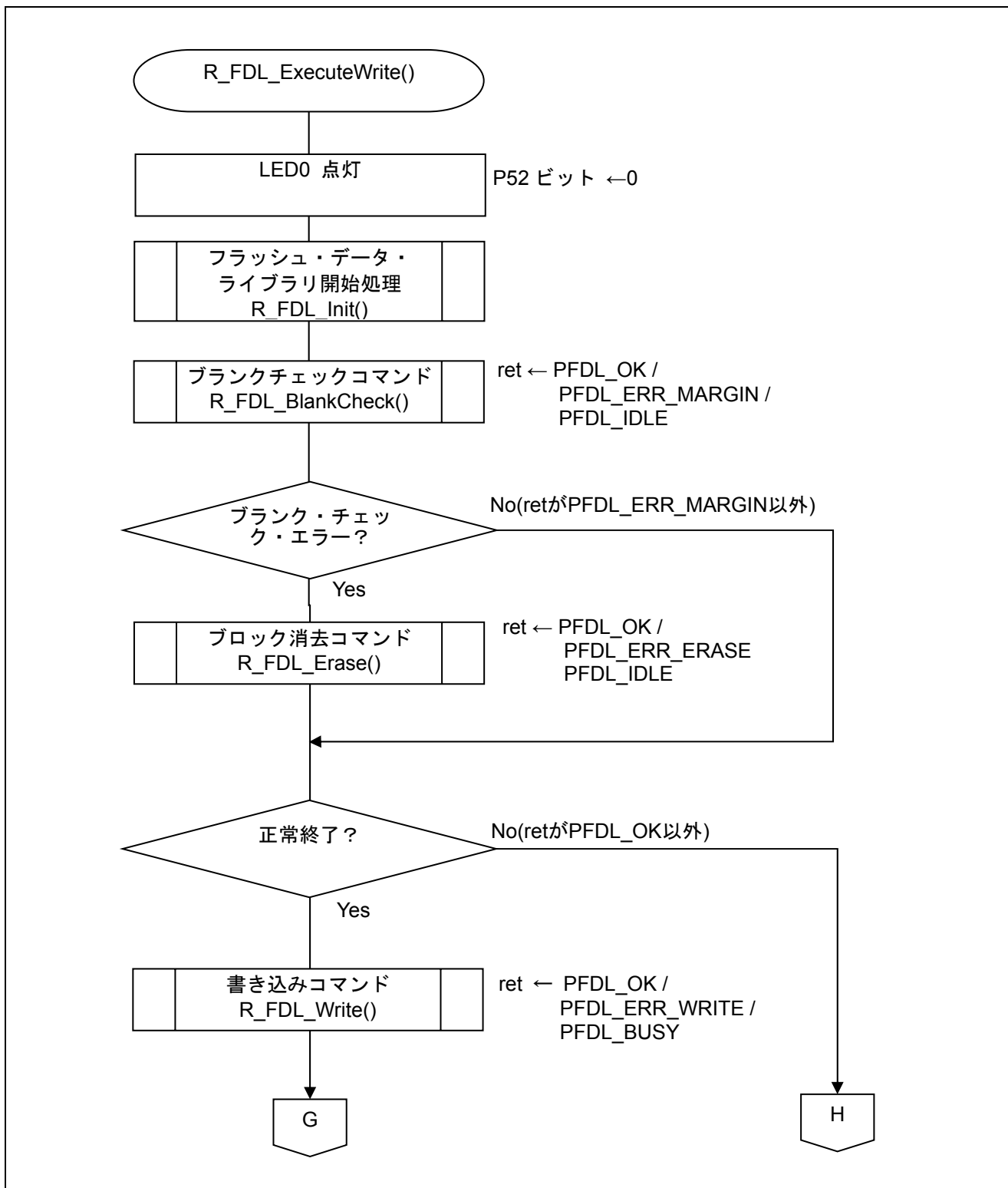


図 5.31 書き込み実行処理(1/2)

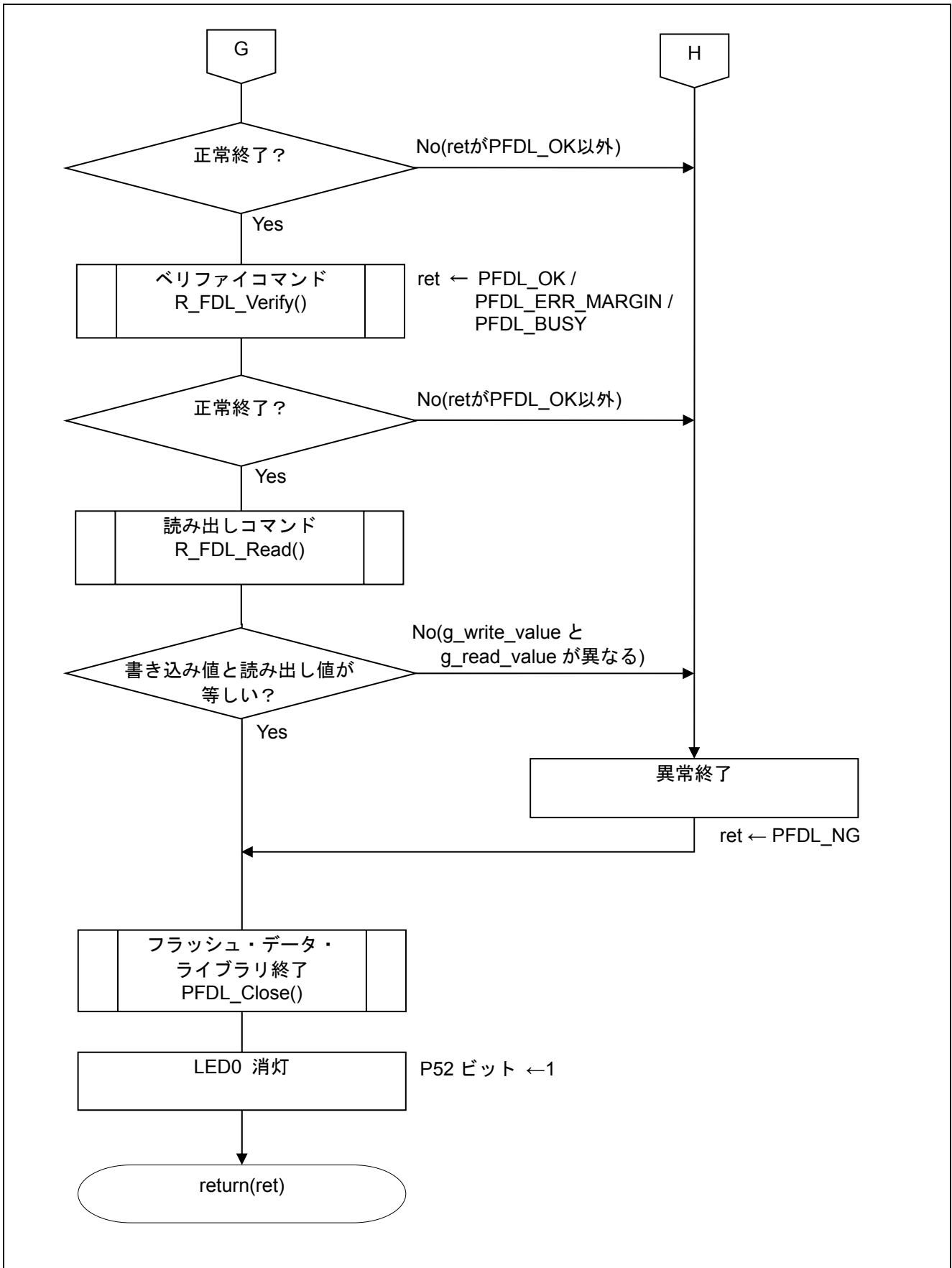


図 5.32 書き込み実行処理(2/2)

5.9.29 ブランクチェックコマンド処理

図 5.33 にブランクチェックコマンド処理のフローチャートを示します。

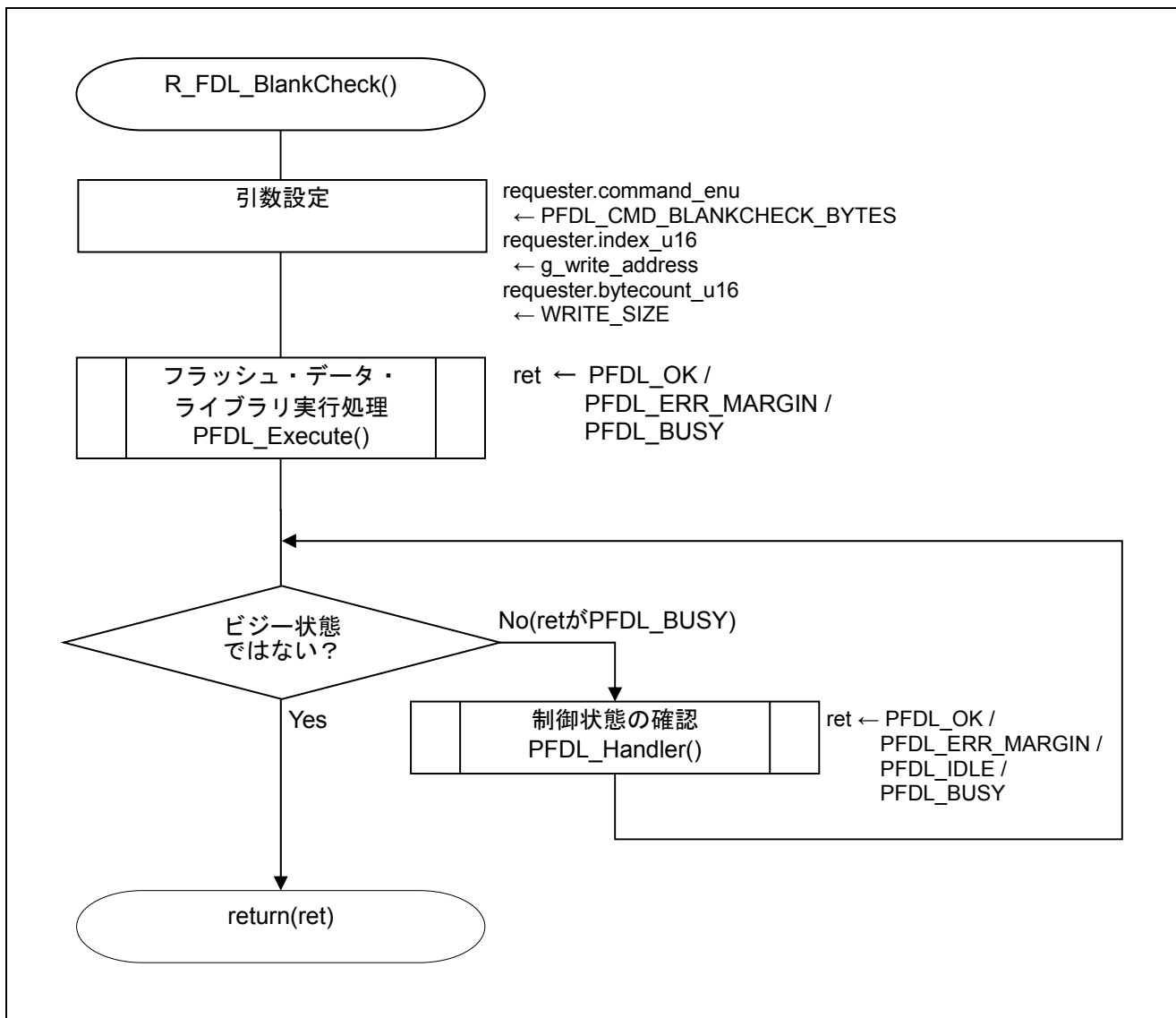


図 5.33 ブランクチェックコマンド処理

5.9.30 ブロック消去コマンド処理

図 5.34 にブロック消去コマンド処理のフローチャートを示します。

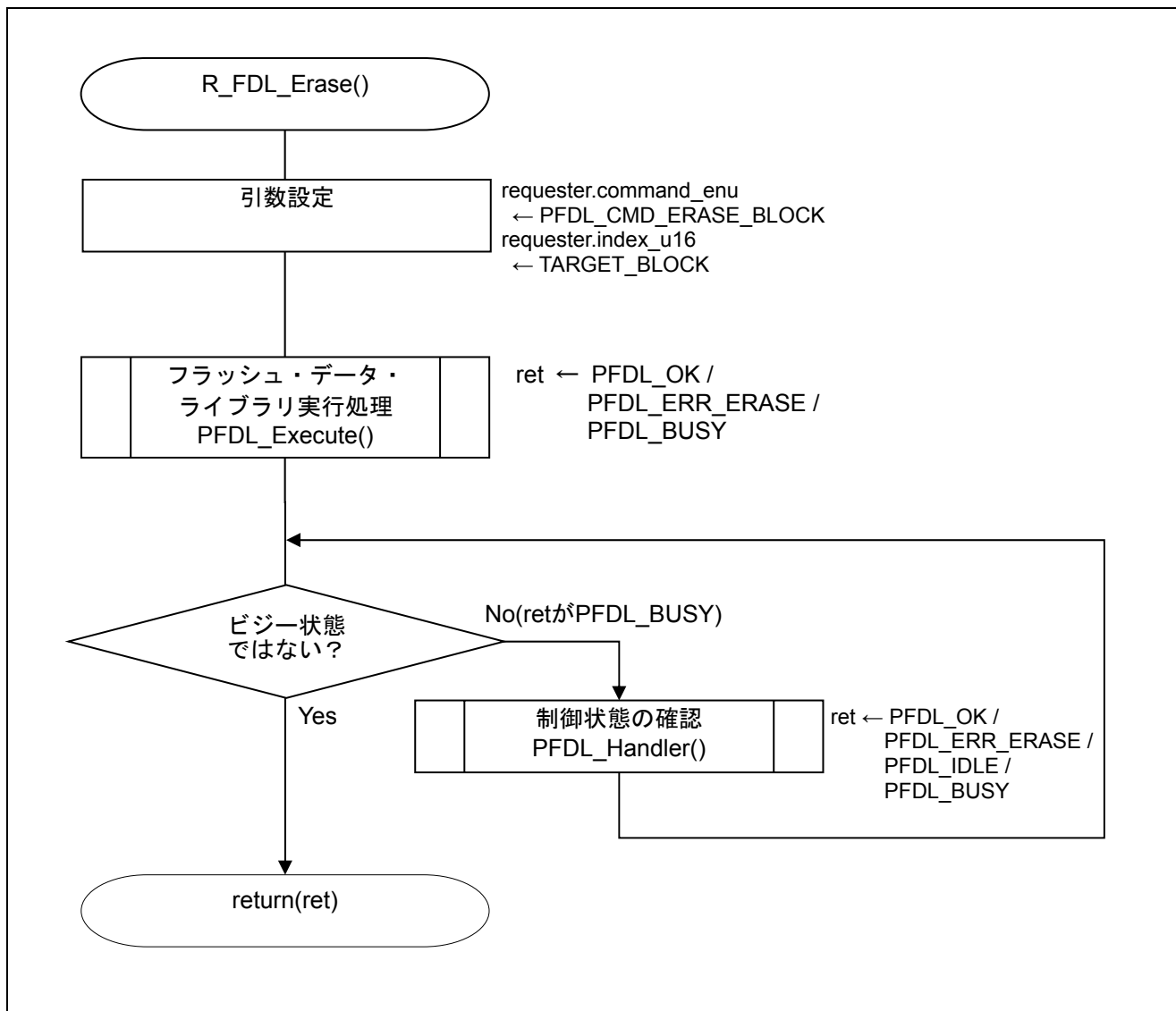


図 5.34 ブロック消去コマンド処理

5.9.31 データ書き込みコマンド処理

図 5.35 にデータ書き込みコマンド処理のフローチャートを示します。

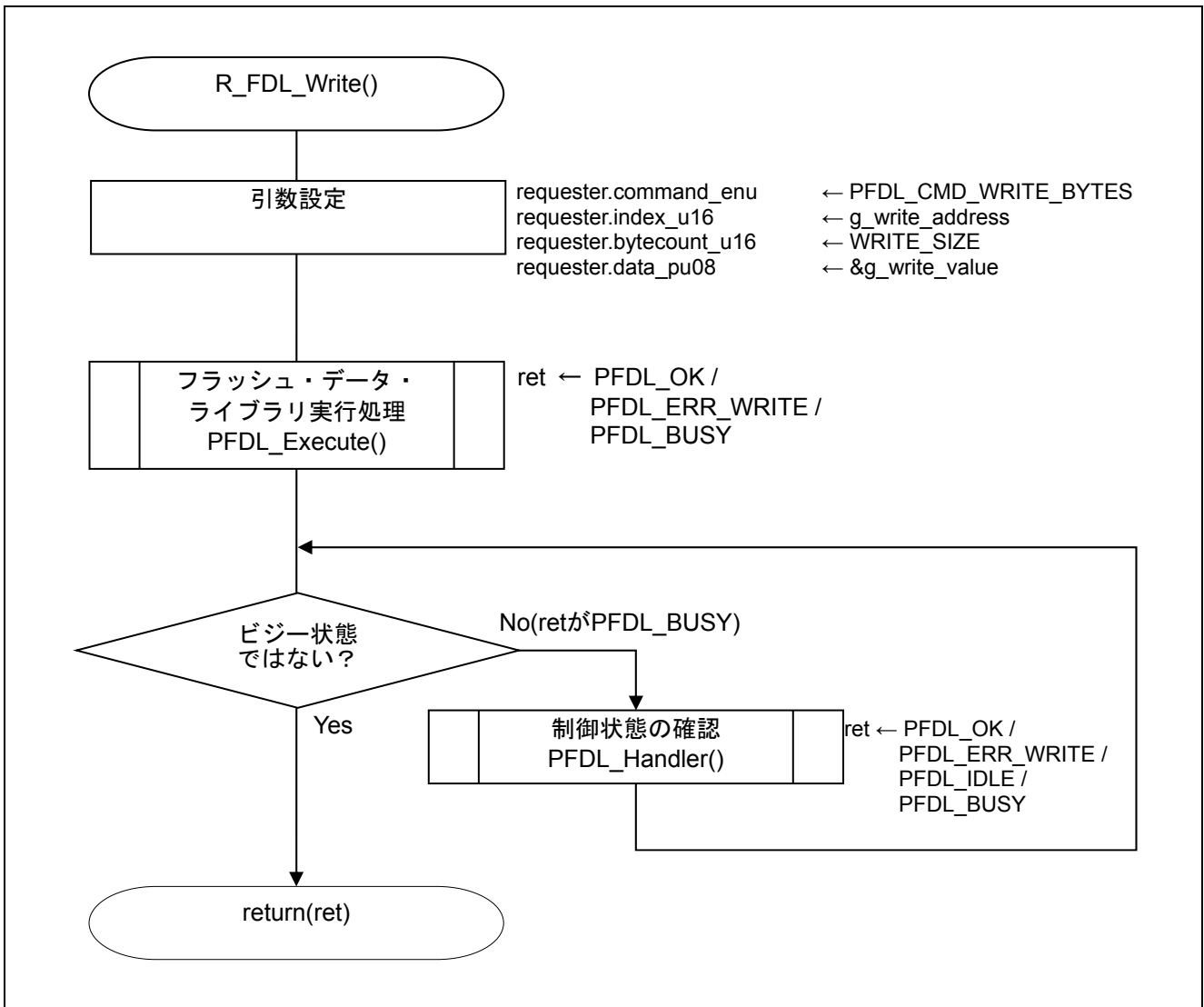


図 5.35 データ書き込みコマンド処理

5.9.32 ベリファイコマンド処理

図 5.36 にベリファイコマンド処理のフローチャートを示します。

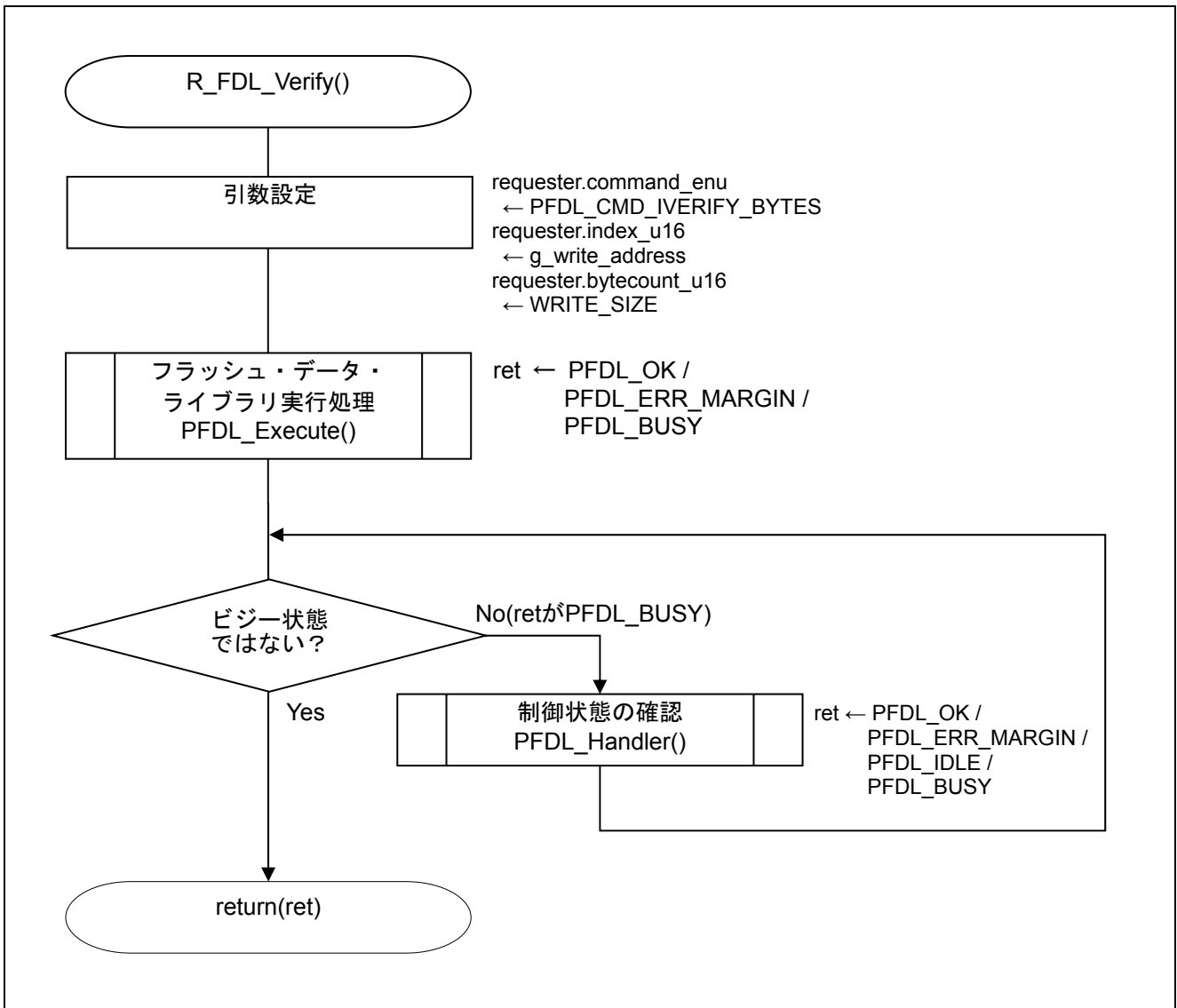


図 5.36 ベリファイコマンド処理

5.9.33 長押し検出処理

図 5.37 に長押し検出処理(1/2)、図 5.38 に長押し検出処理(2/2)のフローチャートを示します。

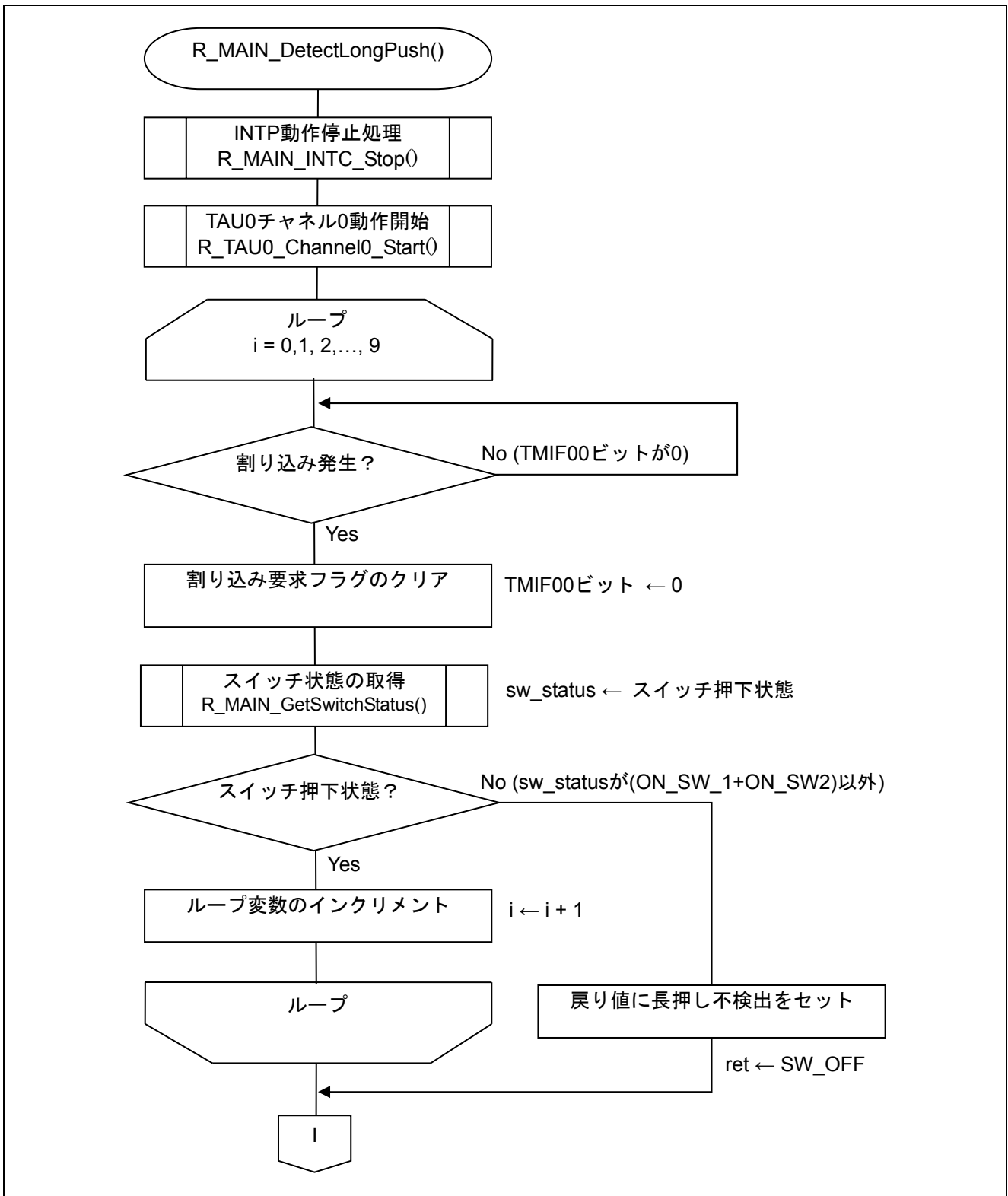


図 5.37 長押し検出処理(1/2)



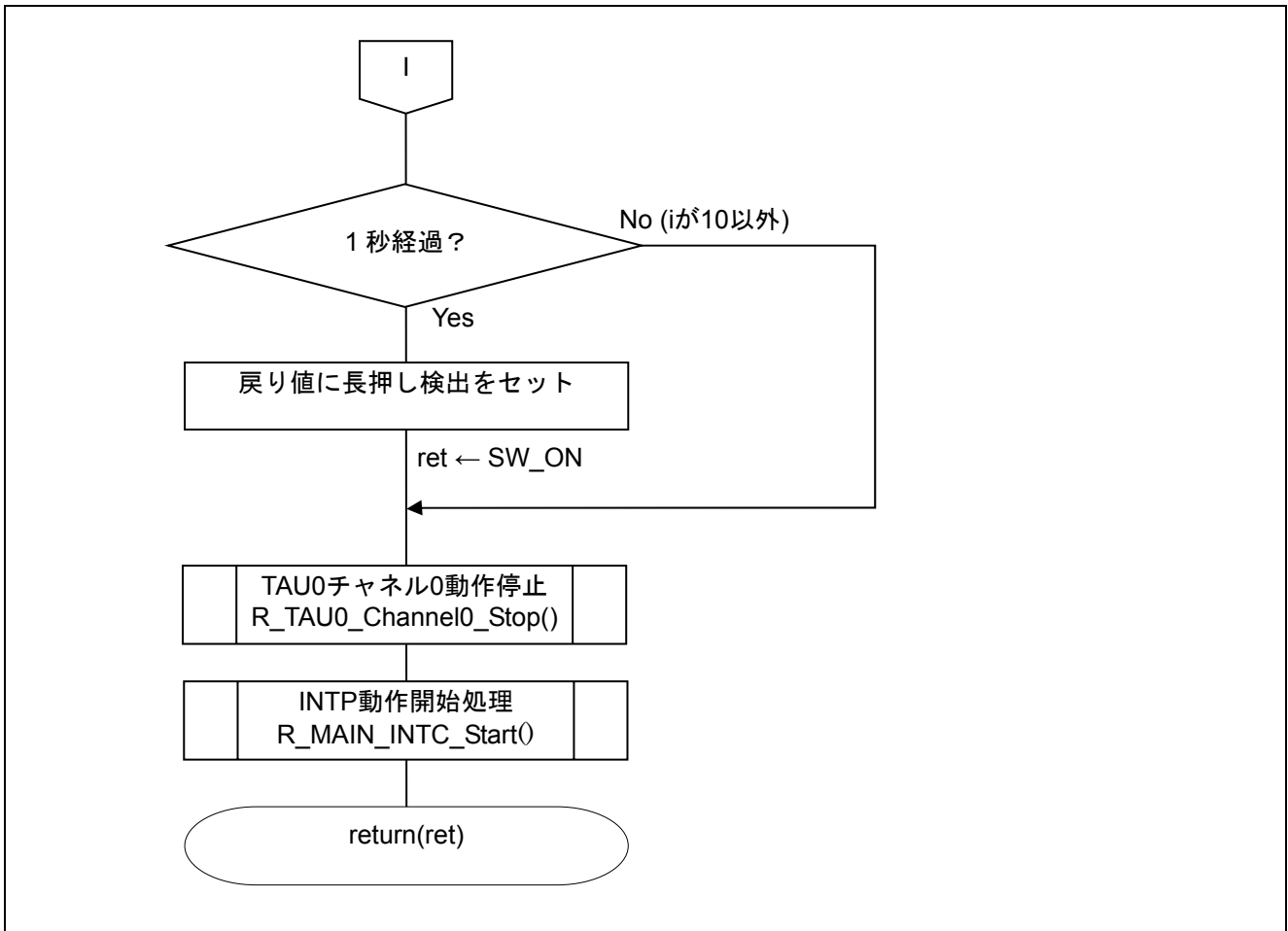


図 5.38 長押し検出処理(2/2)

5.9.34 INTP 動作停止処理

図 5.39 に INTP 動作停止処理のフローチャートを示します。

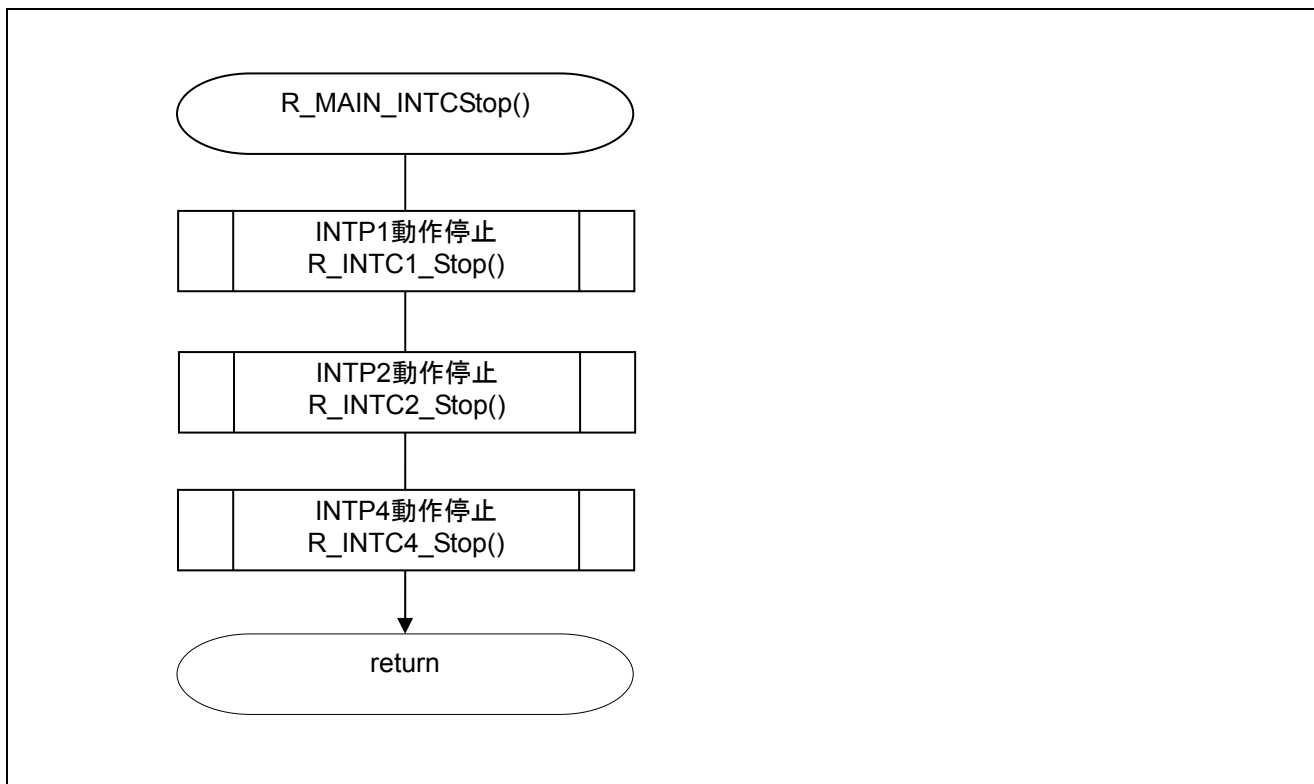


図 5.39 INTP 動作停止処理

5.9.35 INTP1 動作停止処理

図 5.40 に INTP1 動作停止処理のフローチャートを示します。

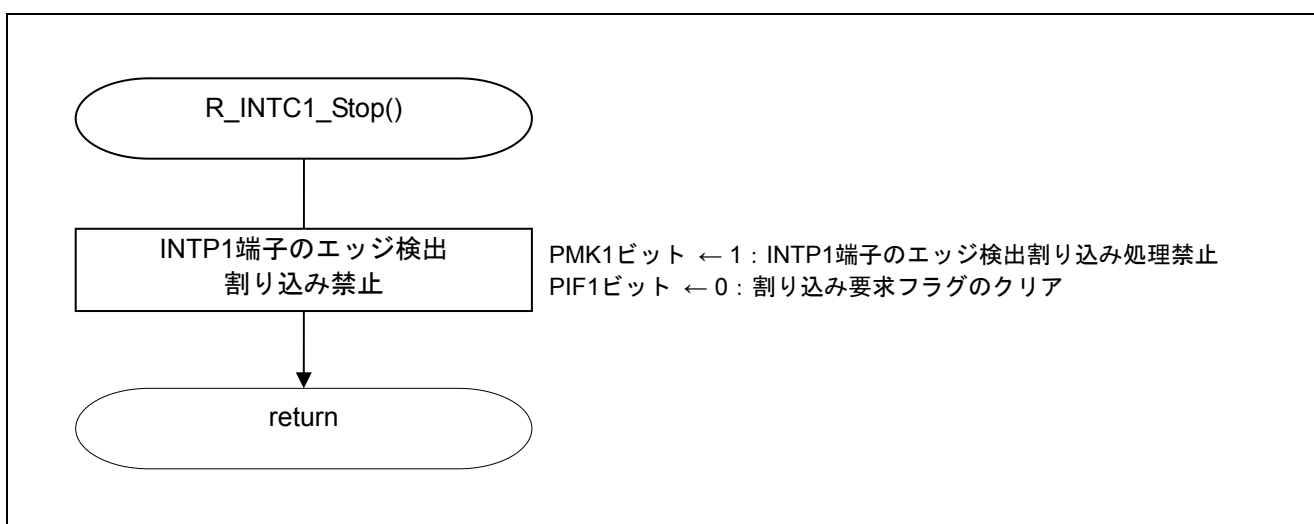


図 5.40 INTP1 動作停止処理

## 5.9.36 INTP2 動作停止処理

図 5.41 に INTP2 動作停止処理のフローチャートを示します。

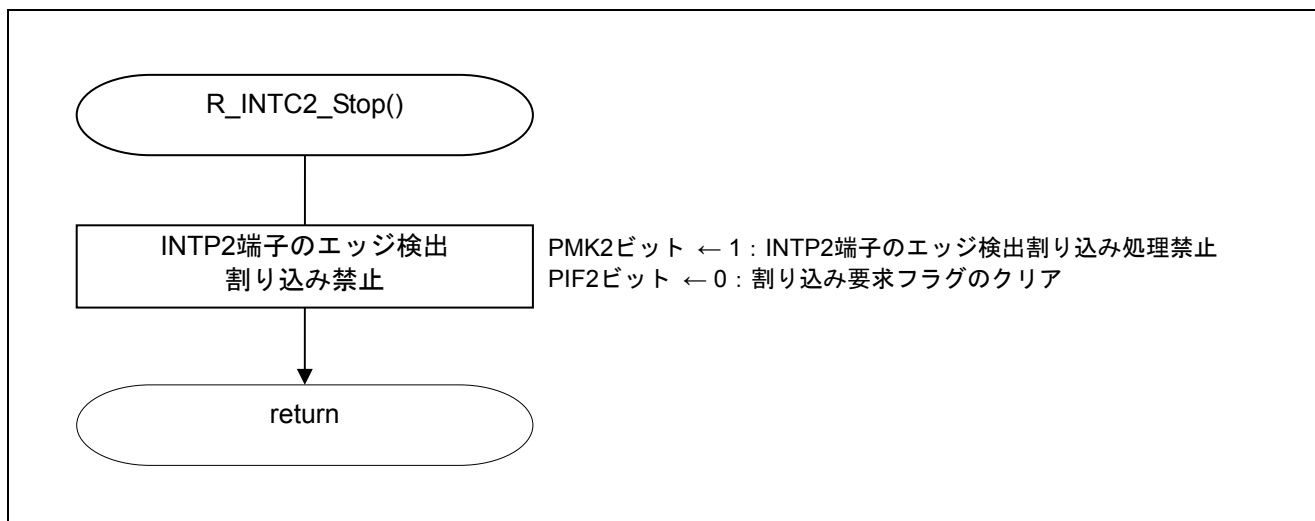


図 5.41 INTP2 動作停止処理

## 5.9.37 INTP4 動作停止処理

図 5.42 に INTP4 動作停止処理のフローチャートを示します。

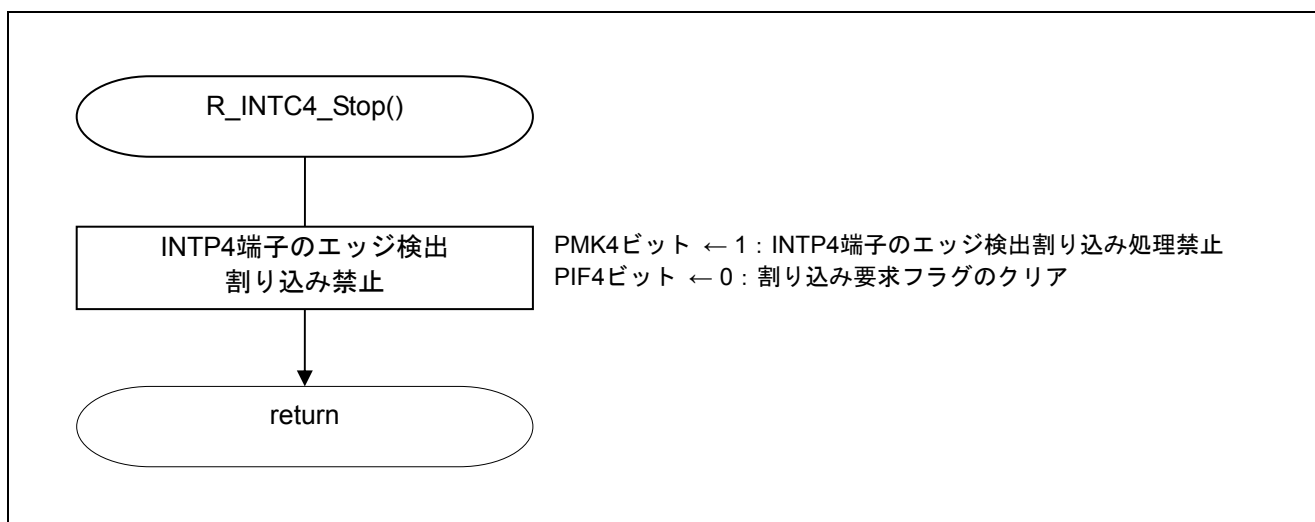


図 5.42 INTP4 動作停止処理

## 5.9.38 TAU0 チャンネル0 動作開始処理

図 5.43 に TAU0 チャンネル0 動作開始処理のフローチャートを示します。

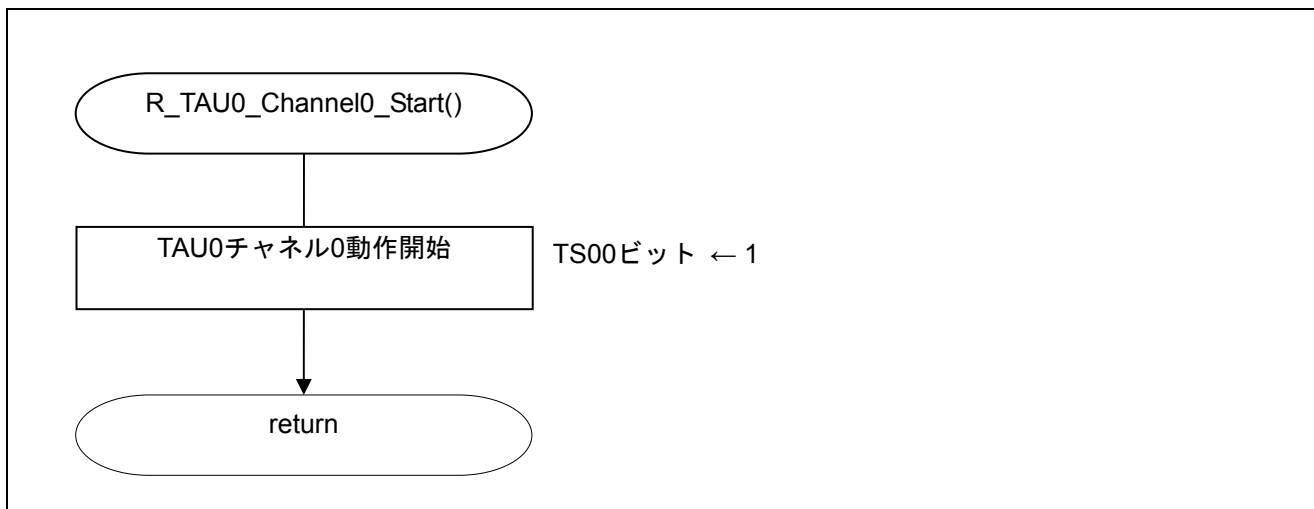


図 5.43 TAU0 チャンネル0 動作開始処理

## 5.9.39 TAU0 チャンネル0 動作停止処理

図 5.44 に TAU0 チャンネル0 動作停止処理のフローチャートを示します。

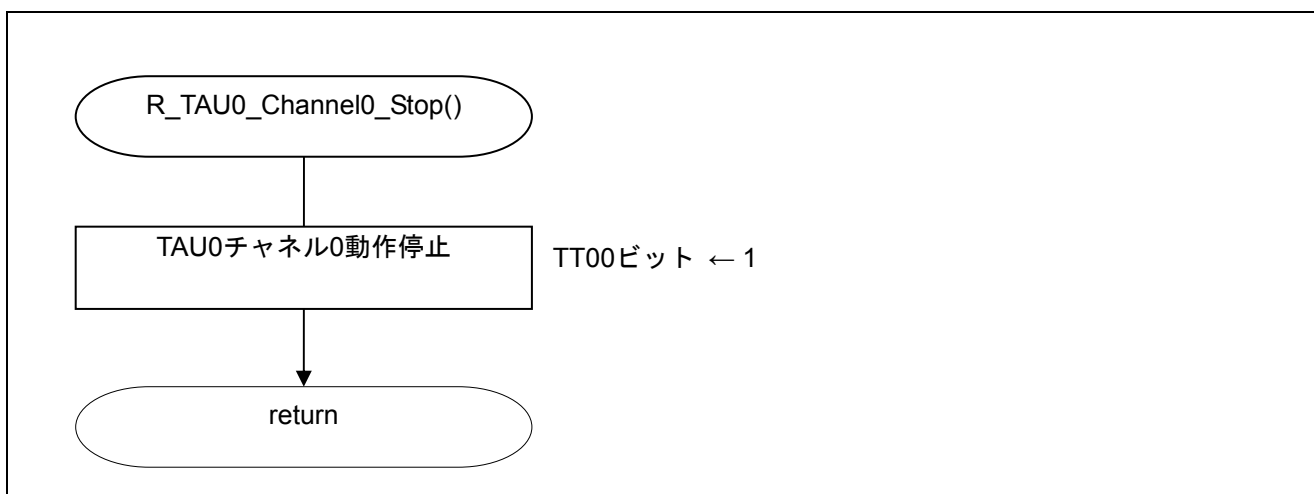


図 5.44 TAU0 チャンネル0 動作停止処理

5.9.40 データ・フラッシュ初期化処理

図 5.45 にデータ・フラッシュ初期化処理のフローチャートを示します。

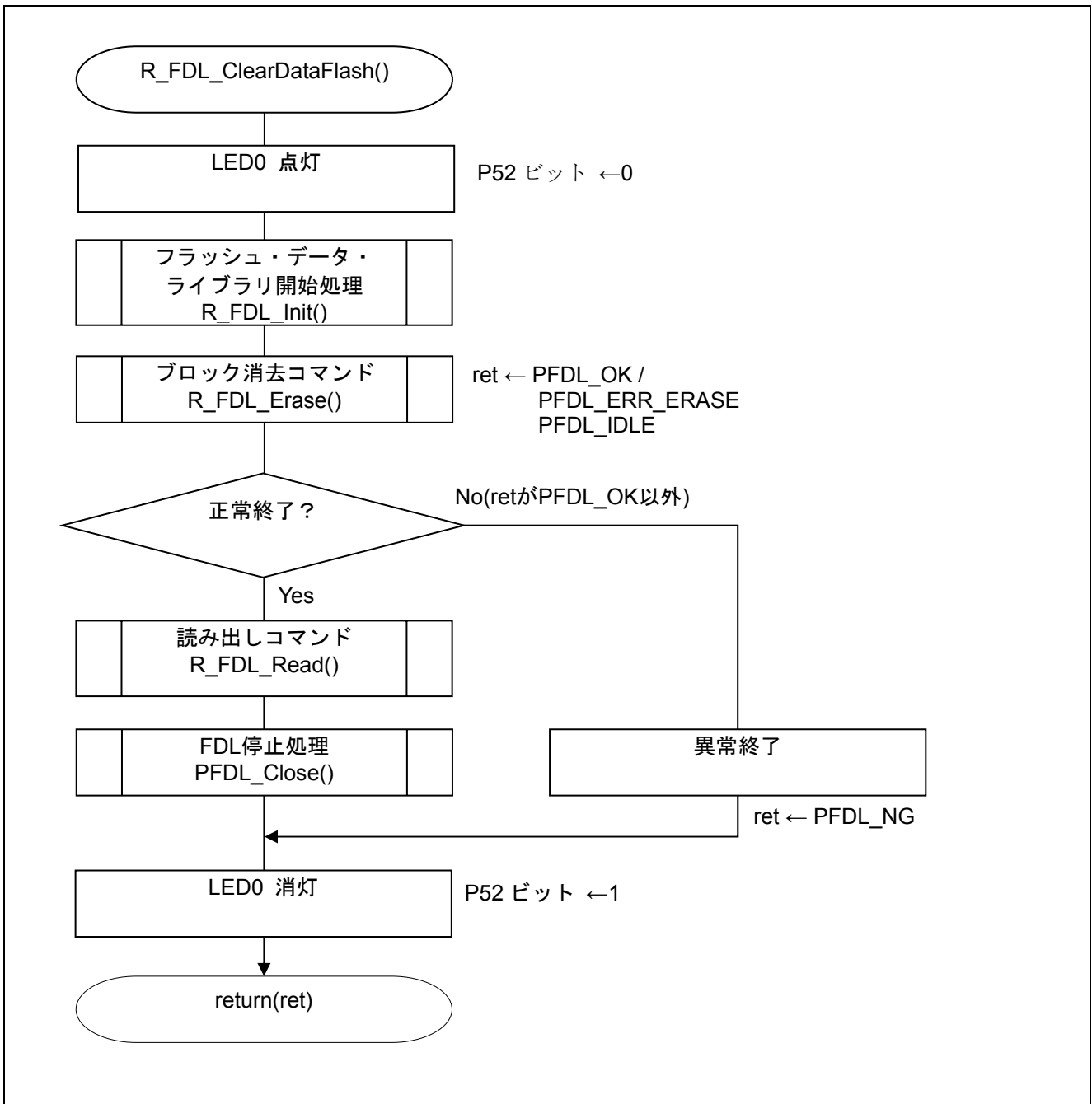


図 5.45 データ・フラッシュ初期化処理

## 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/G13 ユーザーズマニュアル ハードウェア編 (R01UH0146J)

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・データ・ライブラリ Type04 ユーザーズマニュアル (R01US0049J)

(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/G13 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	2013.03.01	—	初版発行
1.10	2015.06.10	5	参考ドキュメント修正
		9	関連アプリケーションノートを削除
		34,37,38	図 5.4、図 5.7、図 5.8 を修正
		70	参考ドキュメントを追加
1.20	2016.06.01	9	1.4 フラッシュ・データ・ライブラリ取得方法を修正

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ものではありませんが、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。



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/>