

---

# RL78/G13

R01AN1361JJ0110

Rev. 1.10

2016.06.01

## EEPROM エミュレーション・ライブラリ Pack01

---

### 要旨

本アプリケーションノートでは、EEPROM エミュレーション・ライブラリ Pack01（フラッシュ・データ・ライブラリ（以降は、FDL と表記する）と EEPROM エミュレーション・ライブラリ（以降は、EEL と表記する））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 EEPROM エミュレーション概要	5
1.2 EEPROM エミュレーション・ブロックの初期化	9
1.3 EEPROM エミュレーション・ブロックの整理	9
1.4 EEL の初期設定値	10
1.5 EEPROM エミュレーション・ライブラリ取得方法	11
2. 動作確認条件	12
3. 関連アプリケーションノート	12
4. ハードウェア説明	13
4.1 ハードウェア構成例	13
4.2 使用端子一覧	14
5. ソフトウェア説明	15
5.1 動作概要	15
5.2 ファイル構成	18
5.3 オプション・バイトの設定一覧	19
5.4 リンク・ディレクティブ・ファイル	20
5.5 定数一覧	20
5.6 変数一覧	21
5.7 関数一覧	22
5.8 関数仕様	23
5.9 フローチャート	33
5.9.1 初期設定関数	34
5.9.2 システム初期化関数	35
5.9.3 入出力ポートの設定	36
5.9.4 CPUクロックの設定	37
5.9.5 TAU0 の設定	38
5.9.6 インターバル・タイマの設定	39
5.9.7 外部割り込み入力の設定	40
5.9.8 メイン処理	41
5.9.9 INTP 動作開始処理	45
5.9.10 INTP1 動作開始処理	46
5.9.11 INTP1 外部割り込み	46
5.9.12 INTP2 動作開始処理	47
5.9.13 INTP2 外部割り込み	47
5.9.14 INTP4 動作開始処理	48
5.9.15 INTP4 外部割り込み	48
5.9.16 インターバル・タイマ動作開始処理	49
5.9.17 インターバル・タイマ割り込み	50
5.9.18 インターバル・タイマ動作停止処理	51
5.9.19 スイッチ押下状態のクリア	52
5.9.20 EEL 初期設定	53
5.9.21 メンテナンス・モード処理	55
5.9.22 スイッチ状態の取得	56
5.9.23 スイッチ押下状態別処理	57
5.9.24 書き込み値インクリメント処理	58
5.9.25 書き込み対象データ ID 切替処理	59
5.9.26 書き込み実行処理	60
5.9.27 長押し検出処理	62
5.9.28 INTP 動作停止処理	64
5.9.29 INTP1 動作停止処理	65
5.9.30 INTP2 動作停止処理	65

---

5.9.31INTP4 動作停止処理.....	66
5.9.32TAU0 チャンネル0 動作開始処理.....	66
5.9.33TAU0 チャンネル0 動作停止処理.....	67
5.9.34データ・フラッシュ初期化処理.....	68
5.9.35EEL 停止処理.....	70
6. 本サンプル・プログラム.....	71
7. 参考ドキュメント.....	71

## 1. 仕様

本アプリケーションノートでは、EEPROM エミュレーション・ライブラリの使用方法を説明します。

LCD に書き込み値、書き込み対象データ ID、及び各データ ID の現在の値を表示します。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、書き込み対象データ ID 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、書き込み対象データ ID 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、書き込み対象データ ID は変更する度に LCD の表示も更新され、各データ ID の現在の値は、電源投入後、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)	書き込みデータの変更 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)	書き込みデータ ID 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。

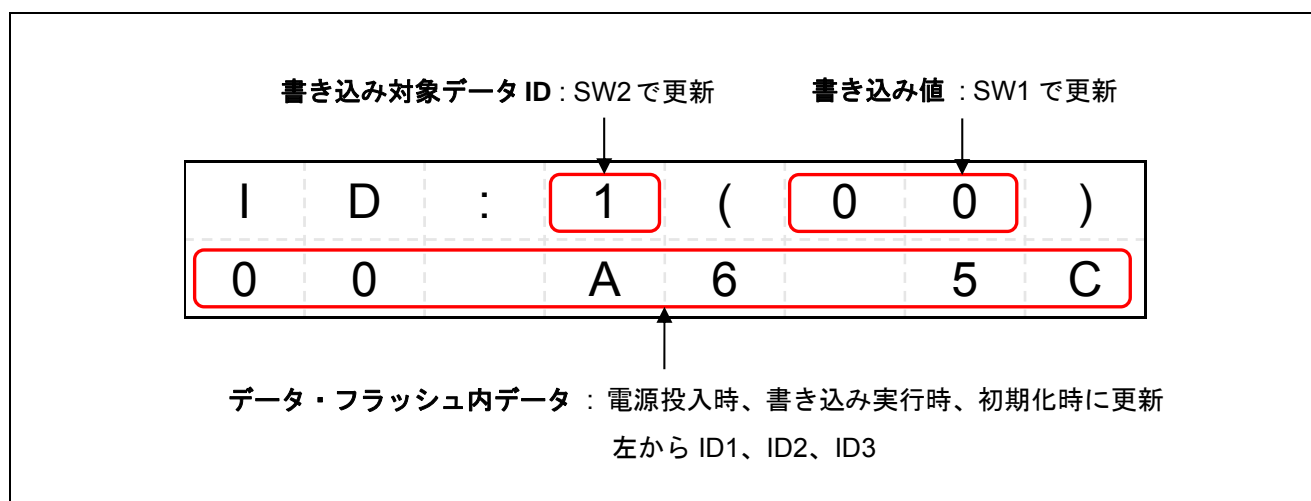


図 1.1 LCD 表示イメージ

## 1.1 EEPROM エミュレーション概要

EEPROM エミュレーションとは、搭載されているフラッシュ・メモリへ EEPROM のようにデータを格納させるための機能です。EEPROM エミュレーションは、FDL と EEL 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実行します。

FDL は、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。EEL は、ユーザ・プログラムから EEPROM エミュレーション機能を実行させるためのソフトウェア・ライブラリです。

ユーザ・プログラムからは、EEL が提供する関数を呼び出す事により、データ・フラッシュ・メモリに関する操作を意識することなく使用することができます。

EEPROM エミュレーション・ライブラリ Pack01 では、1 バイトの識別子（データ ID : 1~255）をデータごとにユーザが割り振り、割り振った識別子ごとに読み出し／書き込みを 1~255 バイトの任意の単位で操作することができます（識別子ごとに割り振れるデータは最大 255 個まで扱うことができます）。

また、データを格納するためのデータ・フラッシュ・メモリは連続した 4 ブロック以上の領域を使用します。このブロックを EEPROM エミュレーション・ブロックと言います。

EEPROM エミュレーションによって書き込まれるデータは、参照用の参照データと、ユーザが指定したユーザ・データに分けられ、参照データはブロックの下位アドレスから、ユーザ・データはブロックの上位アドレスから対象ブロックに書き込みが行われます。

データ・フラッシュ・メモリにユーザ・データを書き込んでいった場合のマップ例を以下に記載します。書き込み順は、ユーザ・データ A→ユーザ・データ B→ユーザ・データ A→ユーザ・データ C とします。

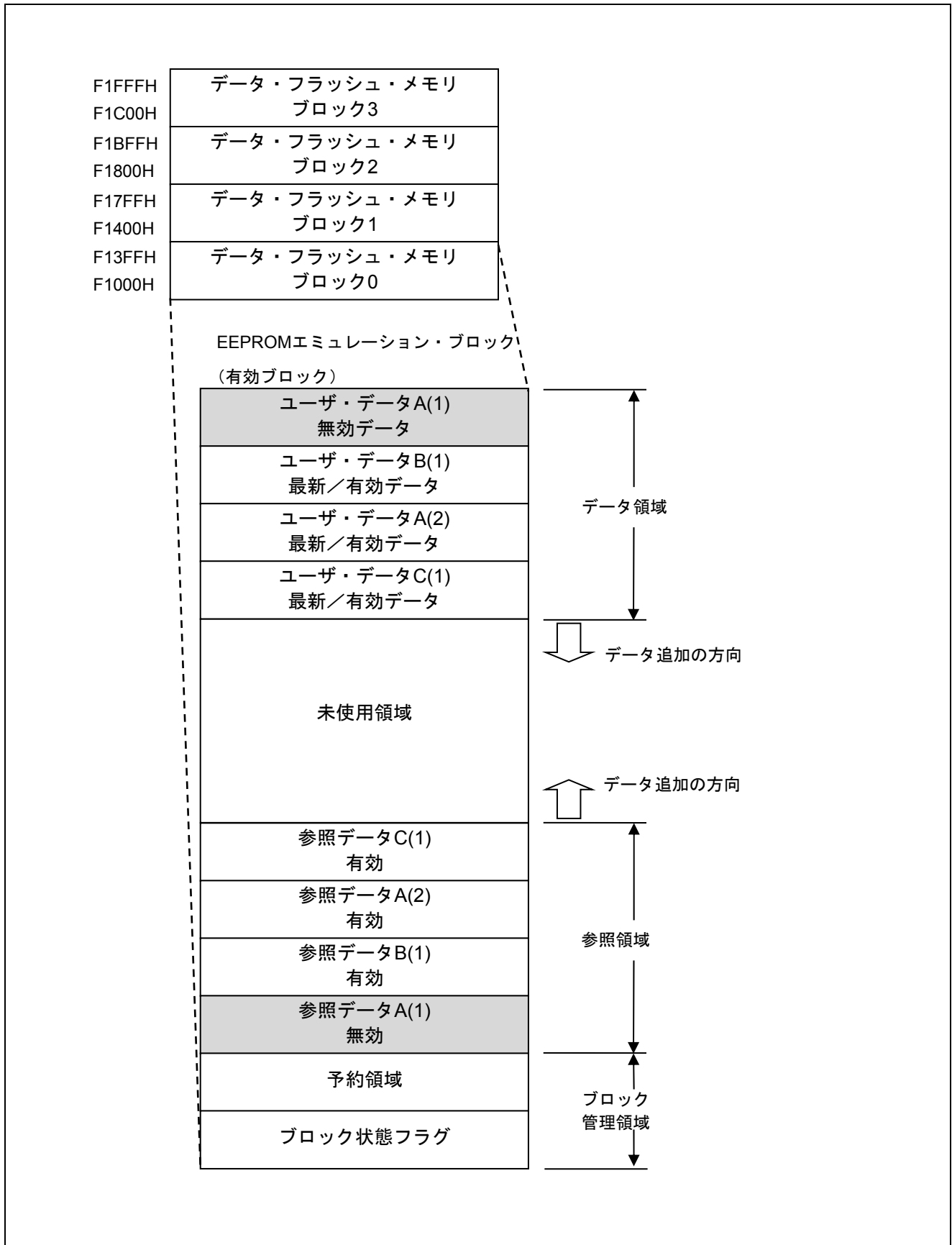


図 1.2 データ書き込みのマップ例

指定したデータを書き込むときに、使用している有効ブロックにデータを書き込むための十分な空き容量がない状態の場合は、有効ブロックを拡張して指定されたデータを書き込みます。

EEPROM エミュレーションはデータ・フラッシュ・メモリのブロック 0 からブロック 3 を順番に有効ブロックや準備ブロックとして拡張し、最後のブロック 3 に到達すると次のブロックを最初のブロック 0 にすることで、ブロックをループさせながら使用します。

書き込み処理のみを継続的に実行した場合、一定のブロック数（残りの準備ブロックが 2 ブロック以下）に達するまで有効ブロックが拡張されていきます。残りの準備ブロックが 2 ブロック以下になった状態でブロックの拡張が発生する場合には、一番古い有効ブロックに存在している有効データを最新の有効ブロックにコピーし、一番古い有効ブロックを消去します。

有効ブロック、準備ブロックの拡張のイメージ図を以下に記載します。

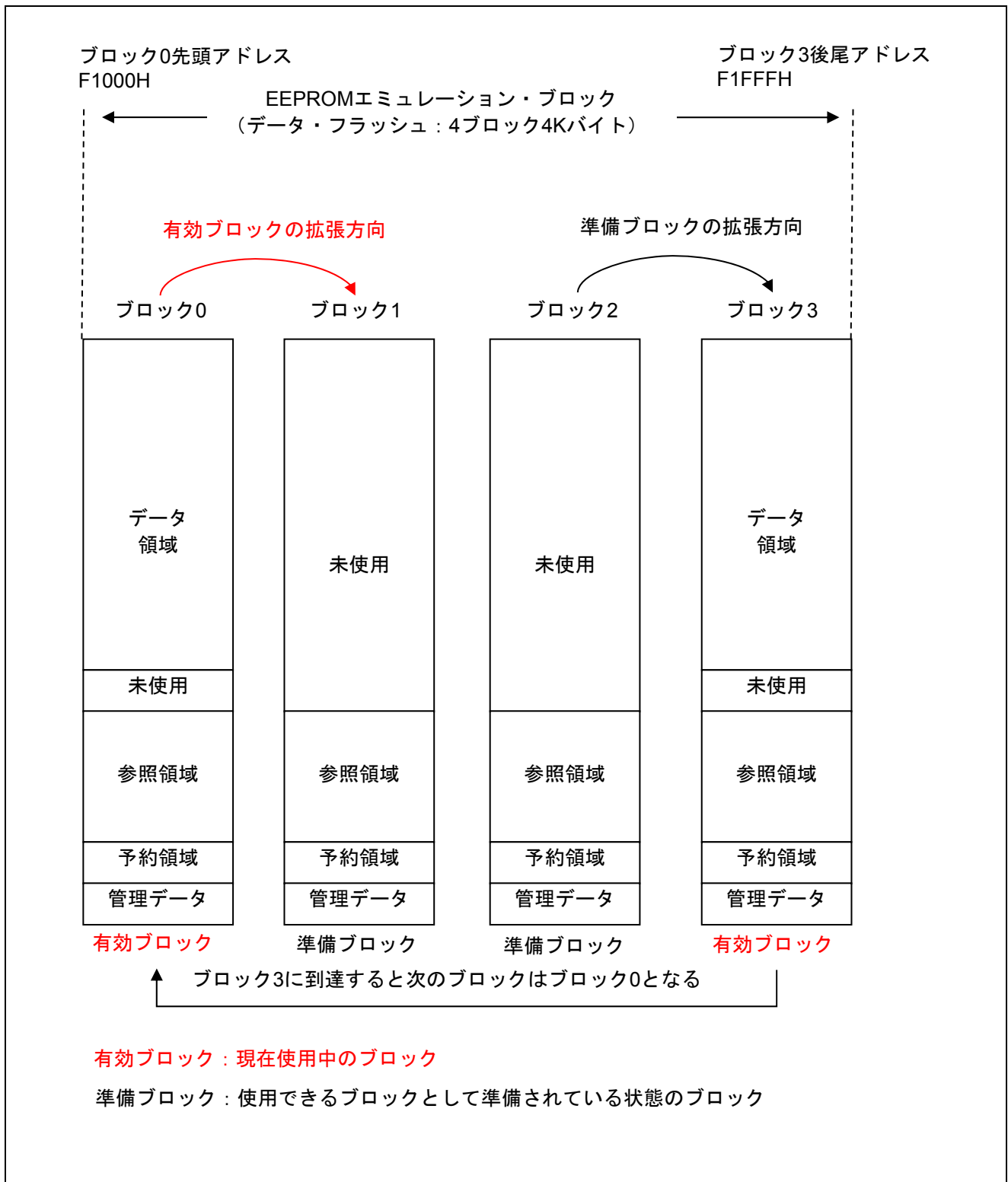


図 1.3 有効ブロック、準備ブロックの拡張のイメージ



データ・フラッシュ・メモリへ書き込むデータの設定項目とその使用範囲を以下に記載します。

設定項目	使用範囲	備考
ユーザ・データ長	1 – 255	-
格納ユーザ・データ数 <sup>注1</sup>	1 – 255	データ種別の数
データ ID の範囲	1 – 255	-
EEPROM エミュレーション・ブロック数 <sup>注2</sup>	4 – 8	-
ユーザ・データの推奨サイズ <sup>注1</sup>	988×総ブロック数×1/4 – 988/2 バイト	書き込みの際に付加される管理用の参照データも含む

注1 ユーザ・データは、全てのユーザ・データが EEPROM エミュレーション・ブロックへ書き込まれるときに必要な合計サイズを総ブロック数の 1/4 以内に収められる状態にする必要があります。そのため、格納するユーザ・データのサイズによって格納ユーザ・データ数の使用範囲は変わります。また、合計サイズも、管理用としてデータごとに付加される参照データ分のサイズも考慮する必要があります。

注2 搭載されているデータ・フラッシュ・メモリの最大ブロック数以上には設定できません。

## 1.2 EEPROM エミュレーション・ブロックの初期化

データ・フラッシュ・メモリを EEL で使用するためには EEPROM エミュレーション・ブロックとして一度初期化を行う必要があります。初期化が必要なブロック状態の場合には EEPROM エミュレーション・ブロック上に有効ブロックが存在しないため、EEL\_CMD\_STARTUP コマンド実行時に「EEL\_ERR\_POOL\_INCONSISTENT」エラーが発生します。このとき、EEPROM エミュレーション・ブロックを使用可能な状態にするために EEL\_CMD\_FORMAT コマンドを実行することによってブロックを初期化する必要があります。

また、データ・フラッシュ・メモリの領域が破壊されてしまった場合や、一度初期化を行った後にユーザ・データの追加等、初期設定値を変更する必要が生じた場合等、EEPROM エミュレーション・ブロックをこれまで使用していた状態のまま継続して使用することができなくなったとき、あるいは、任意のタイミングで初期化を行いたい場合については、任意のタイミングで EEL\_CMD\_FORMAT コマンドを実行する事によって初期化を行うことが可能です。

## 1.3 EEPROM エミュレーション・ブロックの整理

EEPROM エミュレーション・ブロックへの書き込みを行っていくと、有効ブロックを拡張する度に準備ブロックが消費されて行く事になりますが、残りの準備ブロックが2ブロック以下で有効ブロックを拡張する必要が発生した場合には、指定データの書き込みを行う前に有効ブロックの拡張に加えて古いブロックに残っている有効データの移動と、古いブロックの消去処理が行われることとなります。この消去処理を行う必要のあるタイミングで書き込みが発生した場合、書き込みにかかる処理時間に加えてデータの移動、および消去に必要な処理時間も別途発生する事となります。

この追加分の処理時間が許容できない場合は、システムに余裕があるタイミングでブロックのメンテナンスを行うことにより、緊急性の高い書き込みを行う必要が発生したタイミングでデータの移動や消去処理が同時に行われることを回避する事が可能となります。

メンテナンスを行うには、EEL\_CMD\_CLEANUP コマンドの実行によってブロックを整理する方法と、EEL\_Handler関数のメンテナンス・モード処理を実行することによってメンテナンスを実行する方法の二通りがあります。

## 1.4 EEL の初期設定値

EEL の初期設定値として、次に示す項目を必ず設定する必要があります。使用しているマクロ、およびマクロ名は EEL 共通のパラメータとして使用しているため、数値以外は変更しないでください。また、EEPROM エミュレーション・ブロック初期化後 (EEL\_CMD\_FORMAT コマンド実行後) は各値を変更しないでください。変更した場合は、EEPROM エミュレーション・ブロックの再初期化を行ってください。

本アプリケーションノートでは、初期設定値は以下のように設定しています。

### FDL・ユーザー・インクルード・ファイル (fdl\_descriptor.h)

定数名	設定値	備考
FDL_SYSTEM_FREQUENCY	32000000	動作周波数
FDL_WIDE_VOLTAGE_MODE	未定義	データ・フラッシュ・メモリの電圧モード 未定義：フルスピードモード 定義済：ワイドボルテージモード
FAL_POOL_SIZE	4	データ・フラッシュ・メモリのブロック数
EEL_POOL_SIZE	4	データ・フラッシュ・メモリのブロック数

### EEL・ユーザー・インクルード・ファイル (eel\_descriptor.h)

定数名	設定値	備考
EEL_STORAGE_TYPE	'D'	フラッシュ・タイプ (D: データ・フラッシュ)
EEL_VAR_NO	3	格納データ数
EEL_REFRESH_BLOCK_THRESHOLD	1	メンテナンス・モードを実行した場合に整理基準となる有効ブロック数

EEL・ユーザー・プログラム・ファイル (eel\_descriptor.c)

変数名	設定値	備考
eel_descriptor[EEL_VAR_NO + 1][4]	表外へ記載	データの識別子 (データ ID)、およびサイズ
eel_refresh_bth_u08	EEL_REFRESH_BLOCK_THRESHOLD	メンテナンス・モードを実行した場合に整理基準となる有効ブロック数
eel_storage_type_u08	EEL_STORAGE_TYPE	フラッシュ・タイプ (D : データ・フラッシュ)
eel_var_number_u08	EEL_VAR_NO	格納データ数

eel\_descriptor[EEL\_VAR\_NO + 1][4]の設定値を以下に記載します。

データ ID	ワード・サイズ	バイト・サイズ	RAM 参照フラグ
0x01	0x01	0x01	0x01
0x02	0x01	0x01	0x01
0x03	0x01	0x01	0x01
0x00	0x00	0x00	0x00

注意 RAM 参照フラグは、参照設定用のフラグです、登録データには 1 を設定してください。

また、データの最後は終端情報として 0x00 を設定してください。

## 1.5 EEPROM エミュレーション・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 EEPROM エミュレーション・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”eel.h”、”eel\_type.h”、”fdl.h”、”fdl\_types.h”をコピーする。

”libr178”フォルダに”eel.lib”、”fdl.lib”をコピーする。

”descriptor”フォルダに”eel\_descriptor.c”、”eel\_descriptor.h”、”eel\_user\_type.h”、”fdl\_descriptor.c”、

”fdl\_descriptor.h”をコピーする。

EEPROM エミュレーション・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本サンプル・プログラムは、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 (V <sub>LVD</sub> ) : 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 V1.78
使用ボード	Renesas Starter Kit for RL78/G13 (R0K50100LS000BE)
EEPROM エミュレーション・ライブラリ (Type, Ver)	EELRL78 Pack01, Ver1.12 <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

RL78 マイクロコントローラ EEPROM エミュレーション・ライブラリ Pack01 (R01AN0351J) 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。

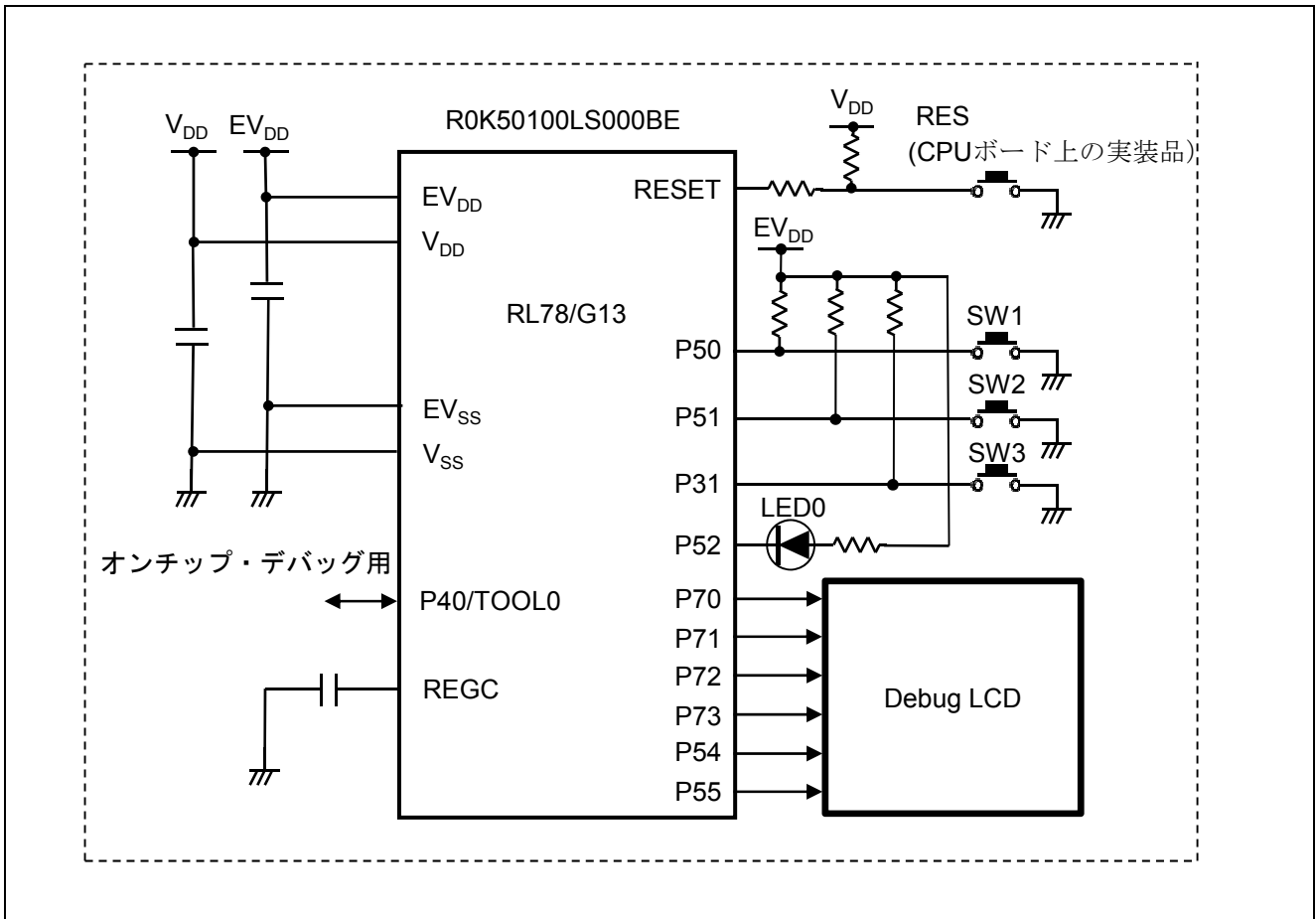


図 4.1 ハードウェア構成例

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	「書き込み対象データ ID」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、EEPROM エミュレーション・ライブラリの使用方法を説明します。

LCD に書き込み値、書き込み対象データ ID、及び各データ ID の現在の値を表示します。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、書き込み対象データ ID 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、書き込み対象データ ID 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、書き込み対象データ ID は変更する度に LCD の表示も更新され、各データ ID の現在の値は、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) EEL の初期設定を行います。

- FDL の初期化を行います。
  - EEL で使用する RAM の初期化を行います。
  - EEL 準備処理を行います。
  - メンテナンス処理を行います。
  - 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。
- ID1～ID3 のデータを読み出します。
- (8) EEL 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 に書き込み値、書き込み対象データ ID、各データ ID の現在の値を出力します。
- (11) スイッチが何も押されていない場合は、HALT モードに移行して、スイッチ入力を待ちます。
- (12) 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・タイマのカウント動作を開始します。
  - 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(8)に戻ります。
- (13) どのスイッチが押されているかを判別します。
- (14) 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、書き込み対象データ ID 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、選択しているデータ ID 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一致の場合は LED0 を消灯します。



- 書き込み値と読み出し値が不一致の場合は、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、ID1 ~ ID3 のデータを読み出します。

(15) (9)に戻ります。

注意 デバイス使用上の注意事項については、RL78/G13 ユーザーズマニュアル ハードウェア編を参照してください。

## 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_ChangeDataID R_MAIN_SwitchProcess
r_eel.c	EEL 実行処理	R_EEL_Initialize R_EEL_Maintenance R_EEL_ExecuteWrite R_EEL_ClearDataFlash R_EEL_Close

### 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。

## 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に本サンプル・プログラムで使用する定数を示します。

表 5.3 本サンプル・プログラムで使用する定数

定数名	設定値	内容
SW_ON	0x01	スイッチ押下確認
SW_OFF	0x00	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
MAX_VALUE	0xFF	データ・フラッシュ書き込み値の最大値
MAX_ID	0x03	データ・フラッシュ書き込みデータ ID の最大値
EEL_NG	0x01	EEL 処理異常終了
EEL_MODE_ENFORCED	0xFF	enforced モード
EEL_MODE_POLLING	0x00	polling モード
LCD_SIZE	0x08	LCD の最大表示文字数

## 5.6 変数一覧

表 5.4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 5.4 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	main R_MAIN_ ClearSwitchFlag r_it_interrupt
uint8_t	g_it_flag	インターバル・タイマ割り込み発生	main R_MAIN_ ClearSwitchFlag r_it_interrupt
uint8_t	g_write_value	書き込み値	main R_MAIN_ IncrementValue R_EEL_ExecuteWrite
uint8_t	g_write_id	書き込み対象データ ID	main R_MAIN_ ChangeDataID R_EEL_ExecuteWrite
uint8_t	g_read_value [EEL_VER_NO]	読み出し値	main R_EEL_ExecuteWrite
int8_t	g_upper_string [LCD_SIZE + 1]	LCD への書き込み文字列(上段)	main
int8_t	g_downer_string [LCD_SIZE + 1]	LCD への書き込み文字列(下段)	r_main R_EEL_ExecuteWrite R_EEL_ClearDataFlash

## 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_MAIN_ClearSwitchFlag	スイッチ押下状態のクリア
R_EEL_Initialize	EEL 初期設定
R_EEL_Maintenance	メンテナンス・モード処理
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_MAIN_ChangeDataID	書き込み対象データ ID 切替処理
R_EEL_ExecuteWrite	書き込み実行処理
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_EEL_ClearDataFlash	データ・フラッシュ初期化処理
R_EEL_Close	EEL 停止処理

## 5.8 関数仕様

本サンプル・プログラム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 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_INTCStart(void)
説明	INTP1、INTP2、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\_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\_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 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_ClearSwitchFlag(void)
説明	スイッチの押下状態をクリアします。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_EEL\_Initialize

---

概要	EEL 初期設定
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h fdl.h fdl_descriptor.h fdl_types.h eel.h eel_descriptor.h eel_user_types.h led.h stdlib.h string.h
宣言	uint8_t R_EEL_Initialize(void)
説明	EEL 初期設定を行い、データ・フラッシュとの送受信を可能にします
引数	sw_status スイッチ押下状態
リターン値	● 正常終了 : EEL_OK ● 異常終了 : EEL_NG
備考	なし

## [関数名] R\_EEL\_Maintenance

---

概要	メンテナンス・モード処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h fdl.h fdl_descriptor.h fdl_types.h eel.h eel_descriptor.h eel_user_types.h led.h stdlib.h string.h
宣言	void R_EEL_Maintenance(void)
説明	ブロックの整理をします。
引数	なし
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 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	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 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	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 SW1、SW2、SW3のスイッチ押下状態
リターン値	<ul style="list-style-type: none"> <li>● 正常終了：EEL_OK</li> <li>● 異常終了：EEL_NG</li> </ul>
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 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_IncrementValue(void)
説明	データ・フラッシュに書き込む値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\_MAIN\_ChangeDataID

---

概要	書き込み対象データ ID 切替処理
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_ChangeDataID(void)
説明	値を書き込むデータ ID を切り替えます。
引数	なし
リターン値	なし
備考	なし

## [関数名] R\_EEL\_ExecuteWrite

---

概要	書き込み実行処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h fdl.h fdl_descriptor.h fdl_types.h eel.h eel_descriptor.h eel_user_types.h led.h stdlib.h string.h
宣言	uint8_t R_EEL_ExecuteWrite(void)
説明	データ・フラッシュの書き込み対象データ ID 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 : EEL_OK ● 異常終了 : EEL_NG
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 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	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	● 長押し検出 : SW_ON ● 長押し不検出 : SW_OFF
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\_EEL\_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 fdl.h fdl_descriptor.h eel.h eel_descriptor.h eel_user_types.h
宣言	uint8_t R_EEL_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 : EEL_OK ● 異常終了 : EEL_NG
備考	なし

## [関数名] R\_EEL\_Close

---

概要	EEL 停止処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h fdl.h fdl_descriptor.h eel.h eel_descriptor.h eel_user_types.h
宣言	uint8_t R_EEL_Close(void)
説明	EEPROM エミュレーションを停止状態にした後、EEPROM エミュレーション終了処理を行います。
引数	なし
リターン値	● 正常終了 : EEL_OK ● 異常終了 : EEL_NG
備考	なし



## 5.9 フローチャート

図 5.1に本サンプル・プログラムの全体フローを示します。

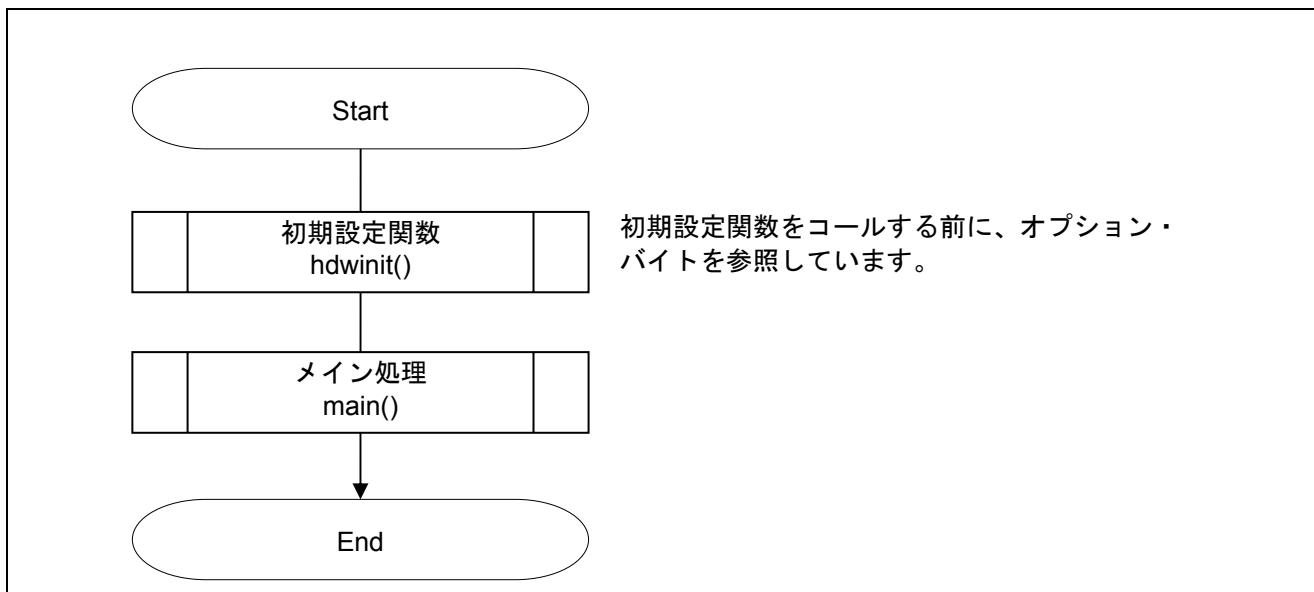


図 5.1 全体フロー

## 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。

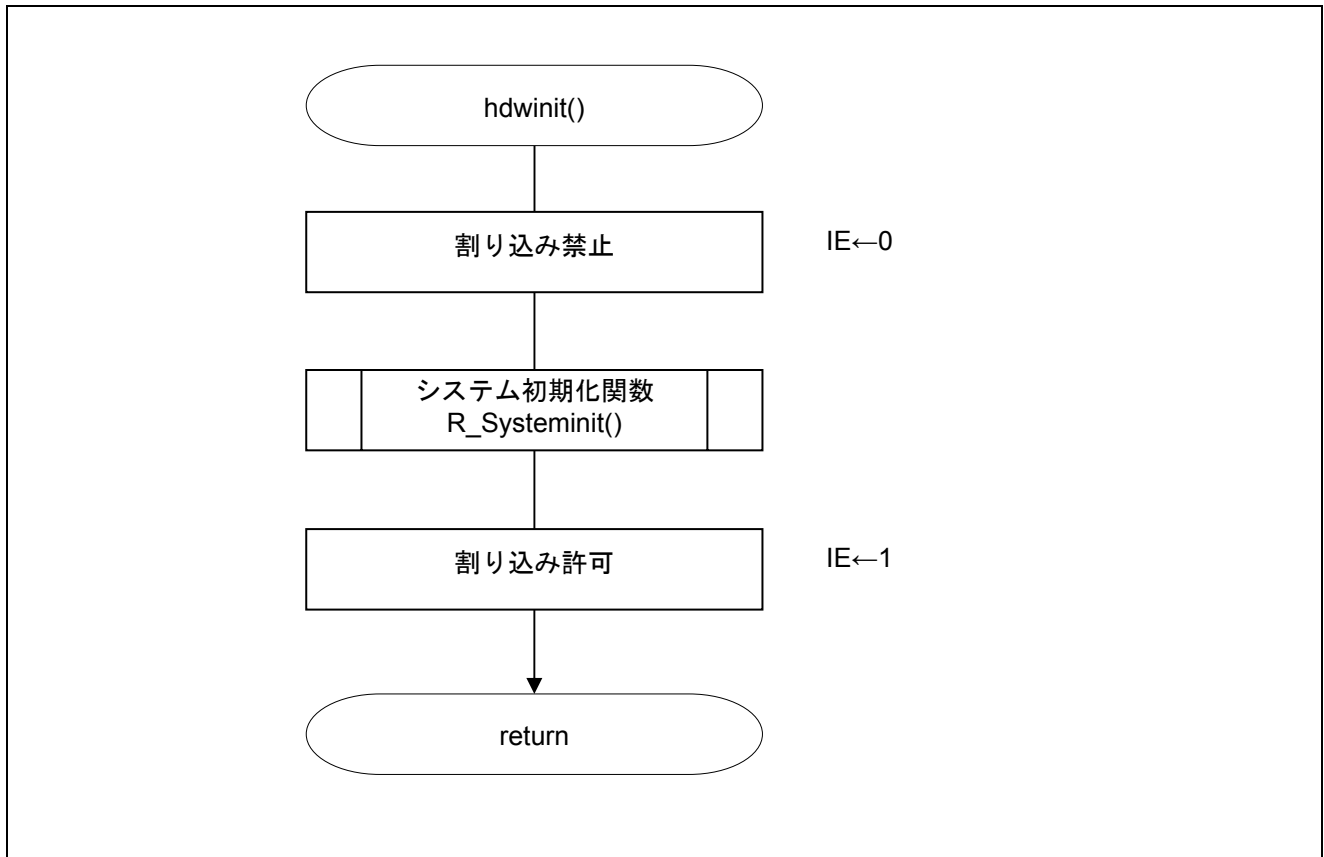


図 5.2 初期設定関数

## 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。

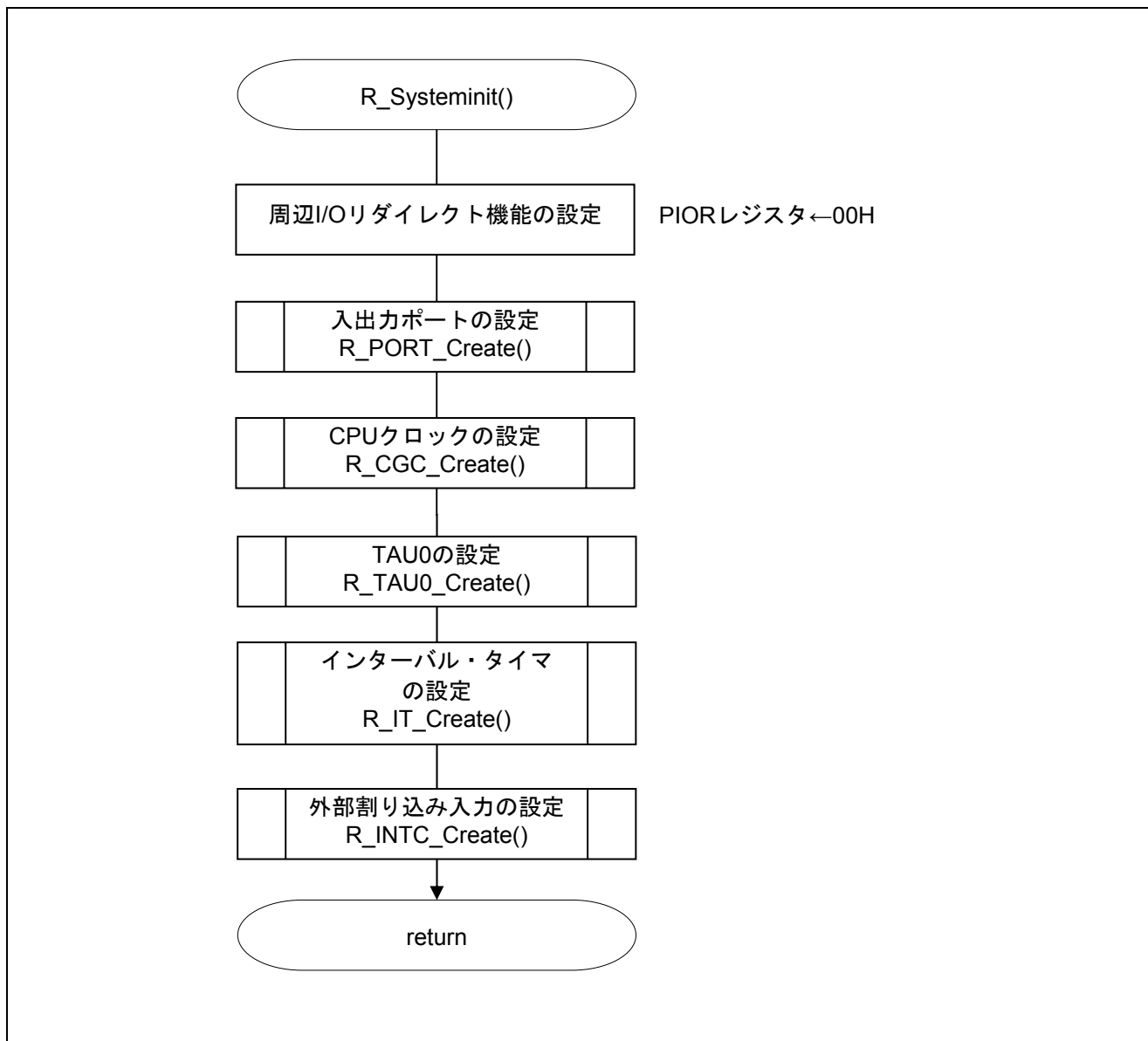


図 5.3 システム初期化関数

### 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。

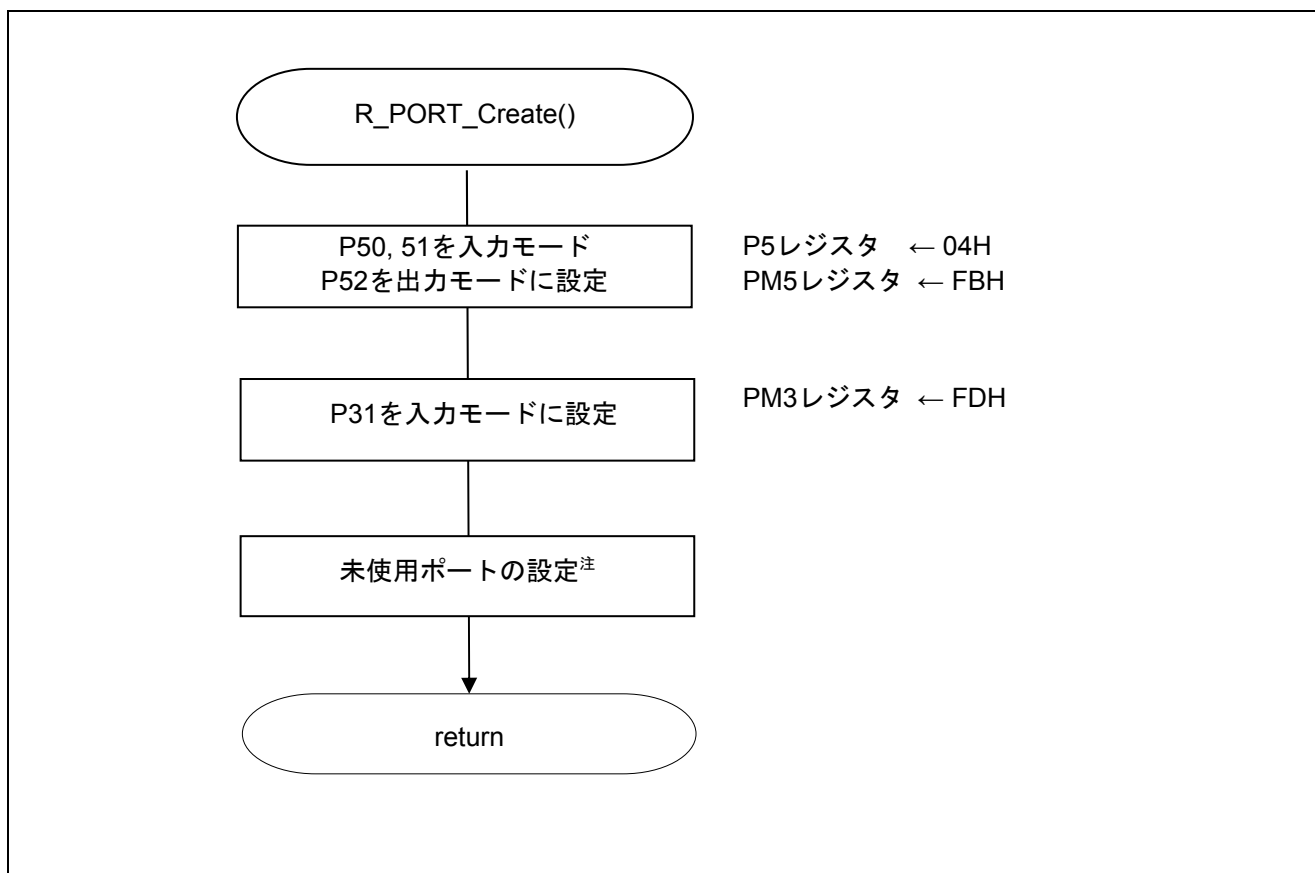


図 5.4 入出力ポートの設定

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。

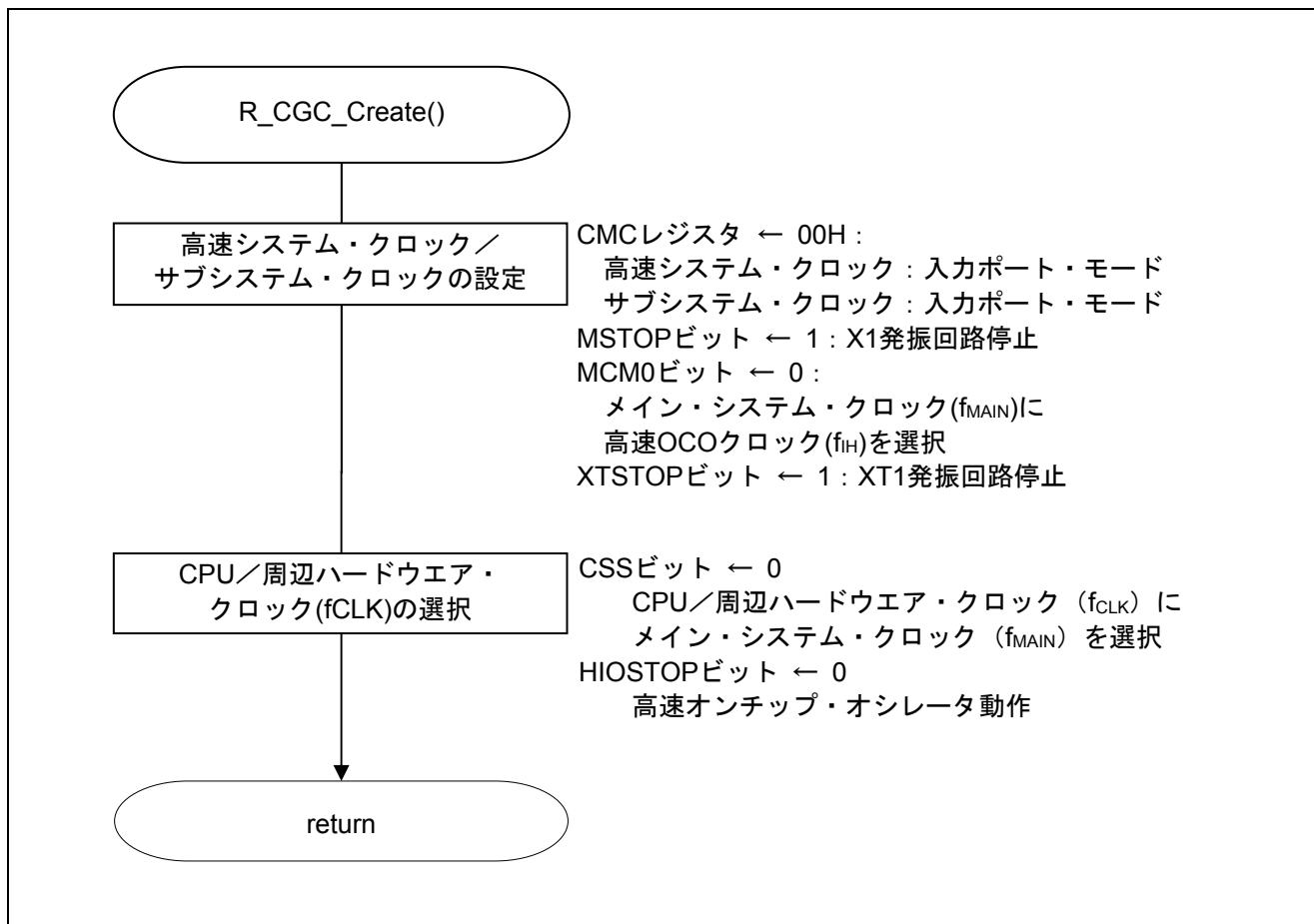


図 5.5 CPU クロックの設定

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。

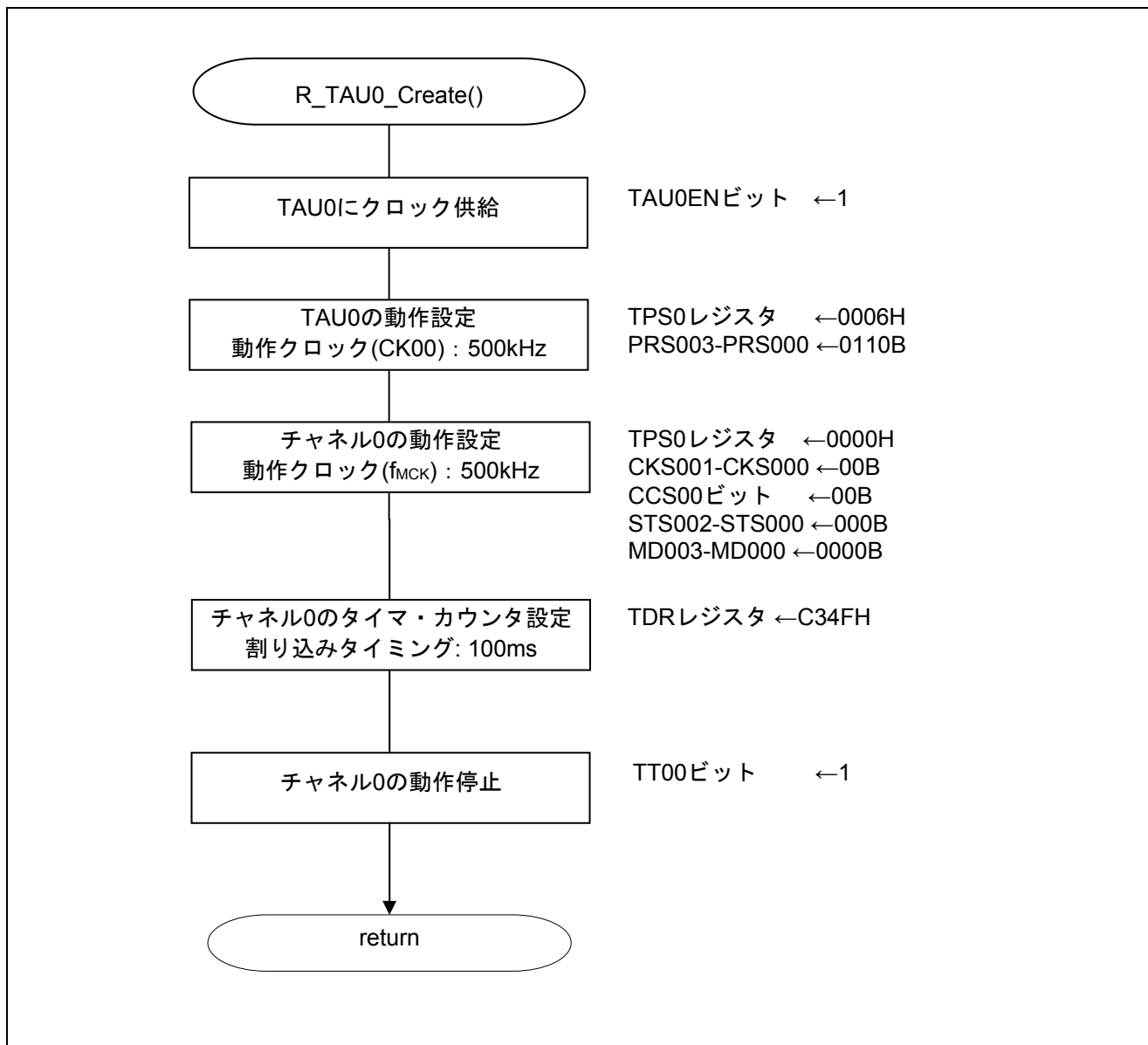


図 5.6 TAU0 の設定

## 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。

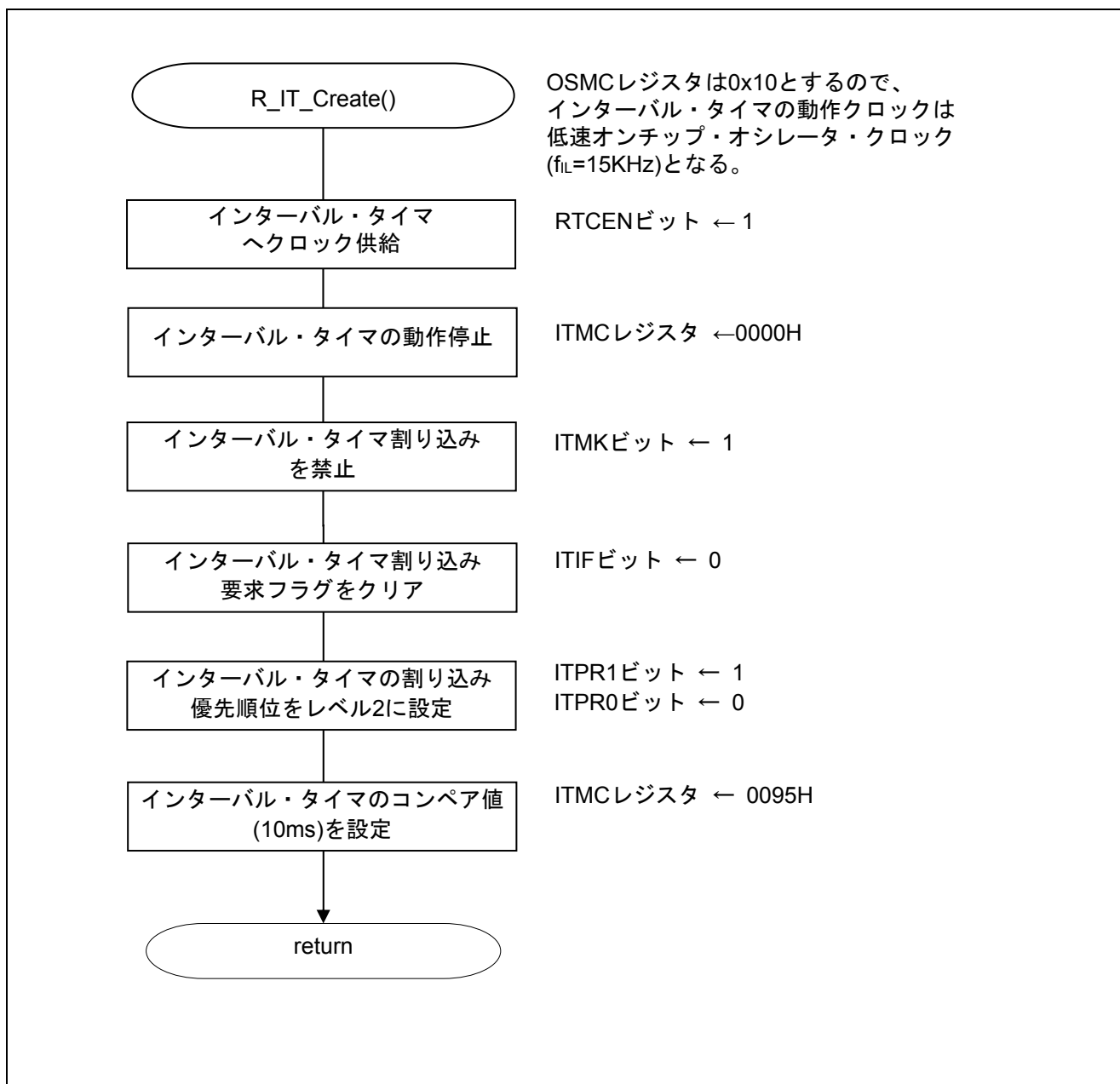


図 5.7 インターバル・タイマの設定

## 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。

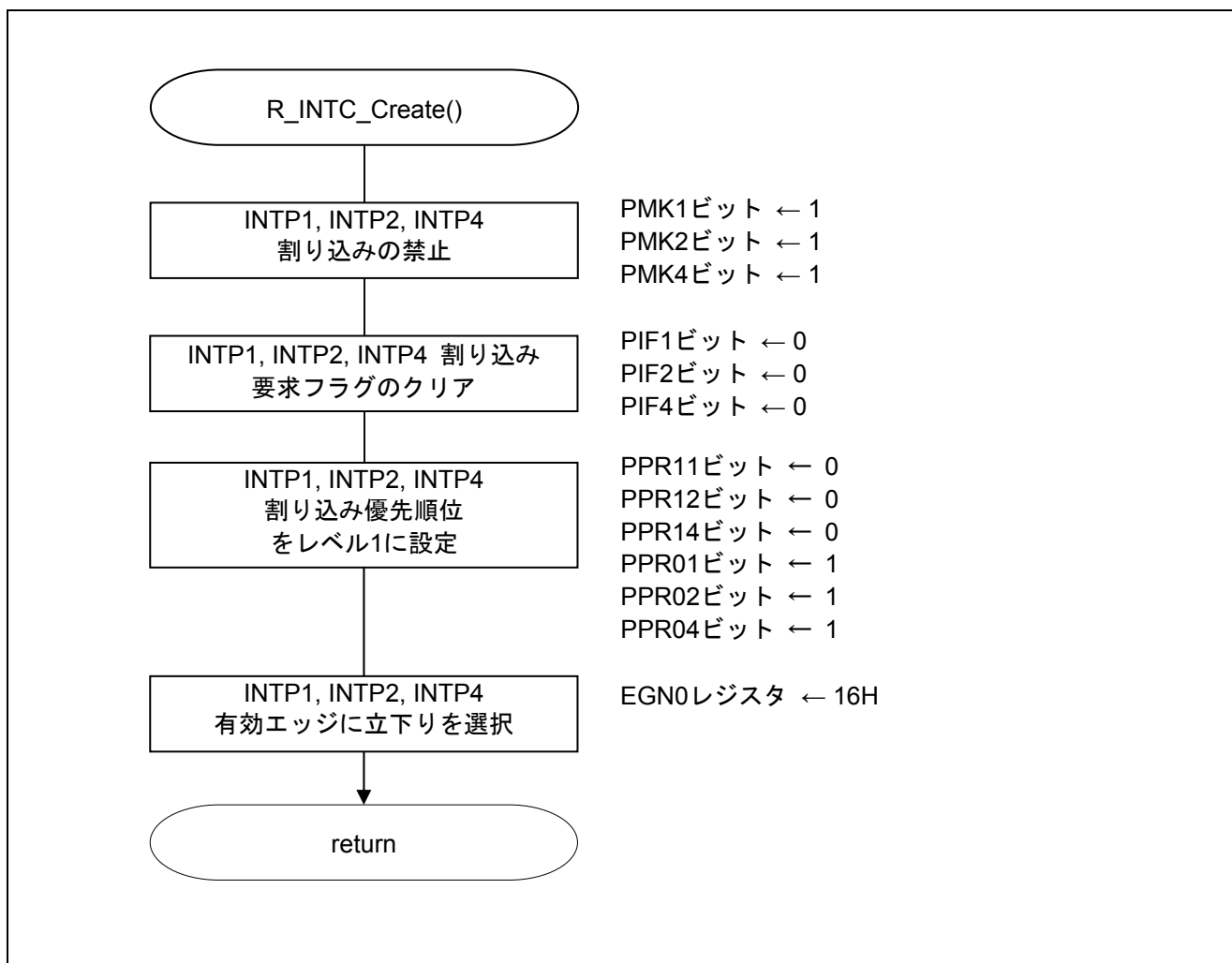


図 5.8 外部割り込み入力の設定



5.9.8 メイン処理

図 5.9から図 5.12にメイン処理のフローチャートを示します。

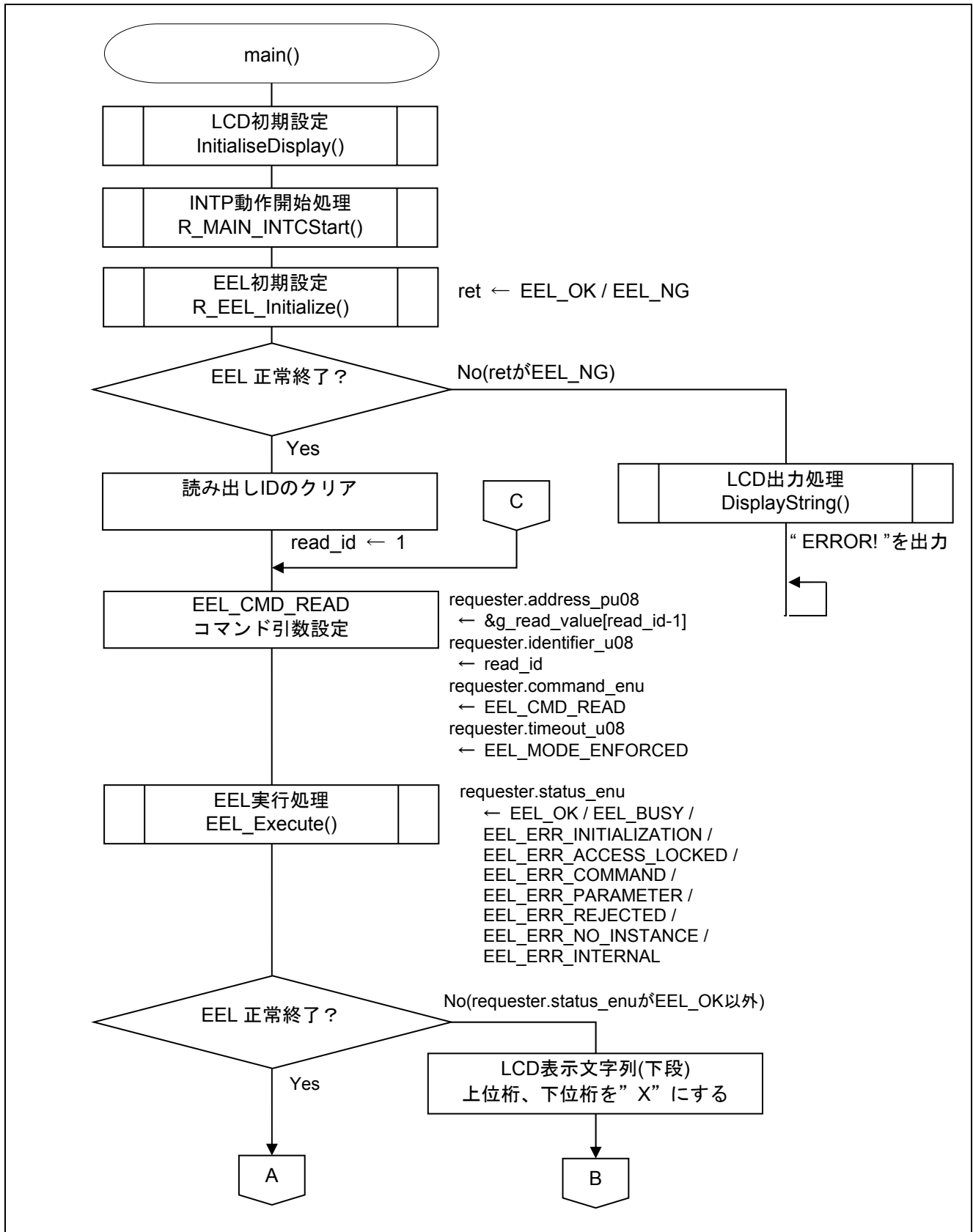


図 5.9 メイン処理(1/4)

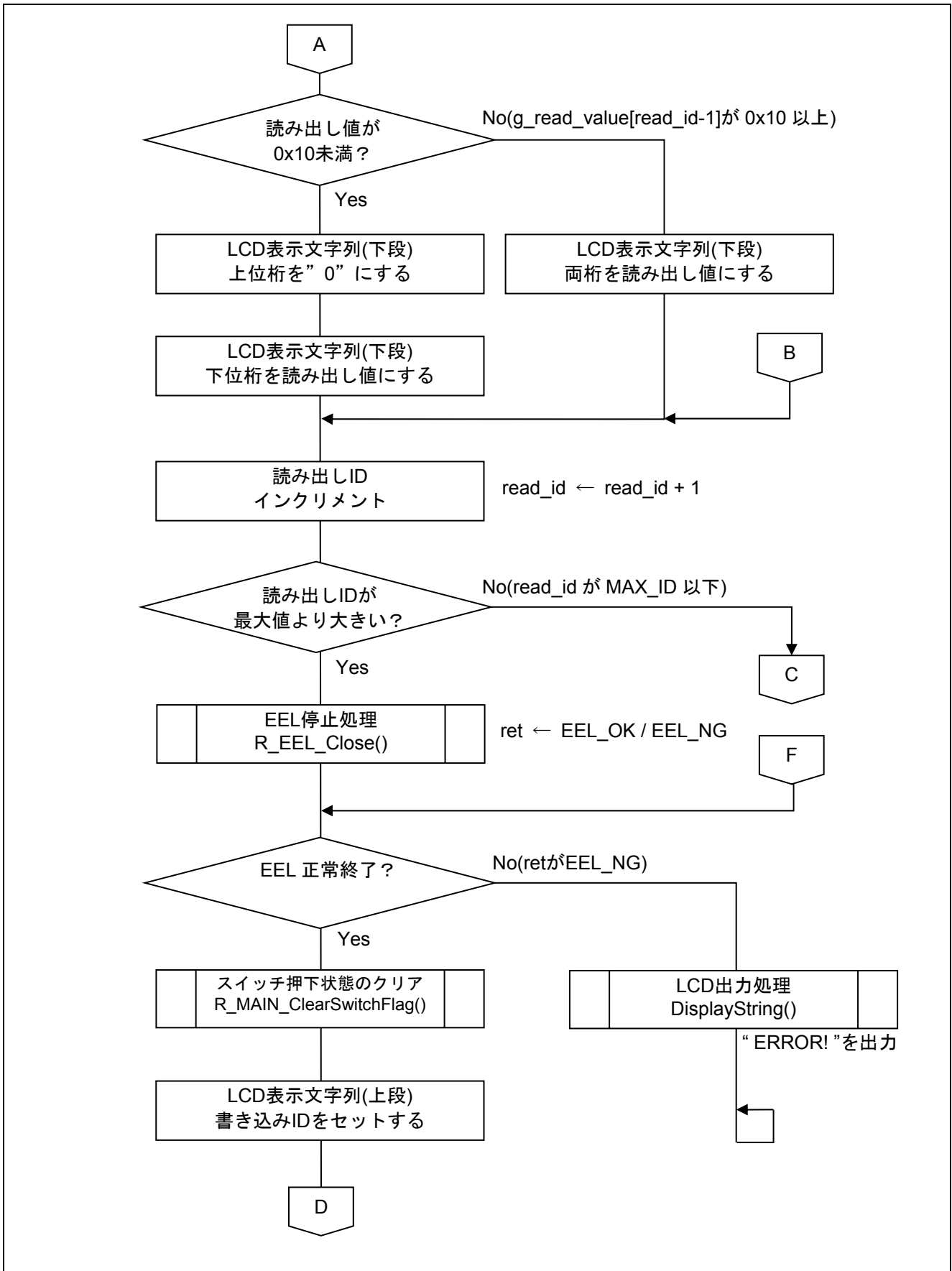


図 5.10 メイン処理(2/4)

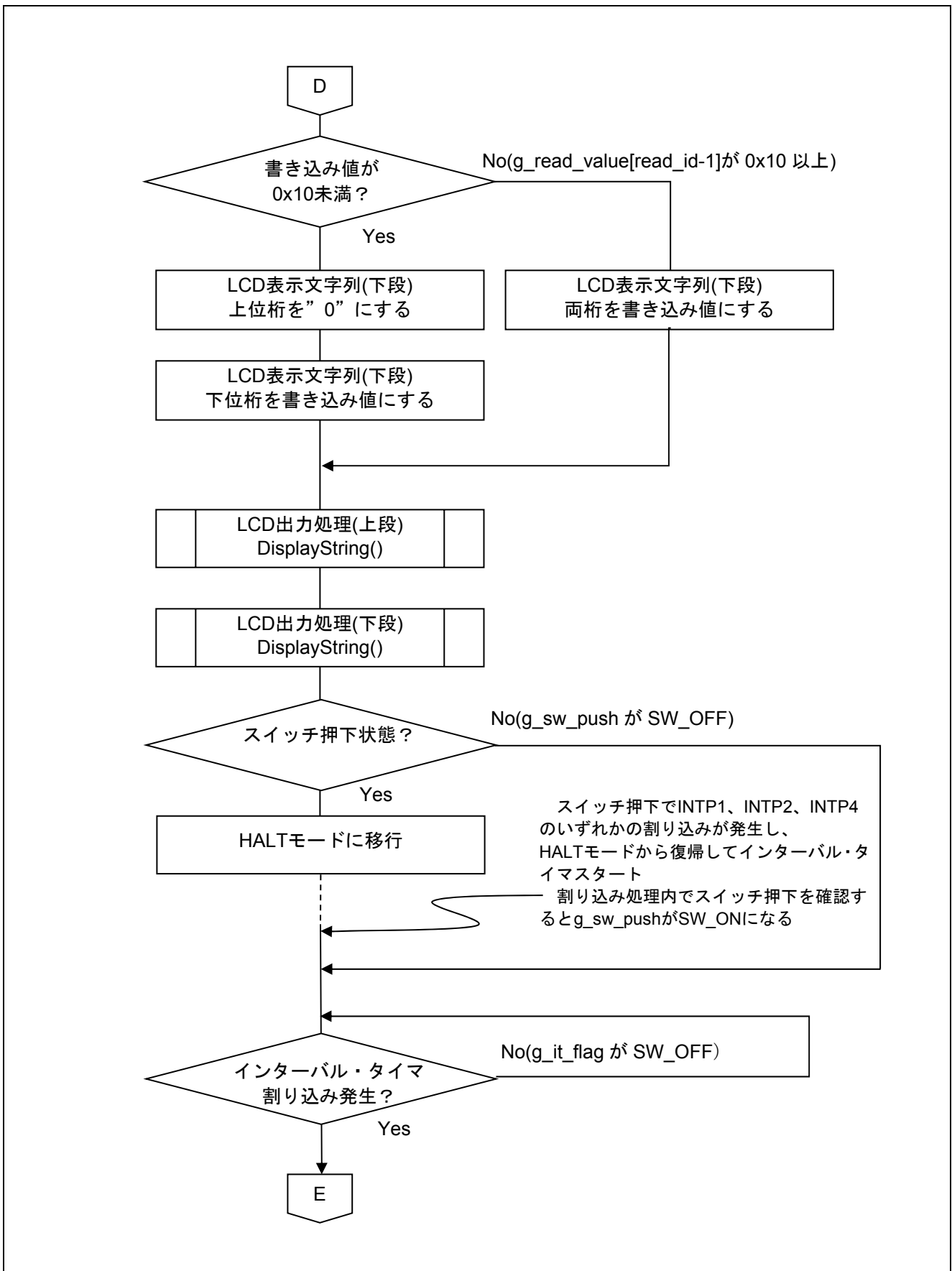


図 5.11 メイン処理(3/4)

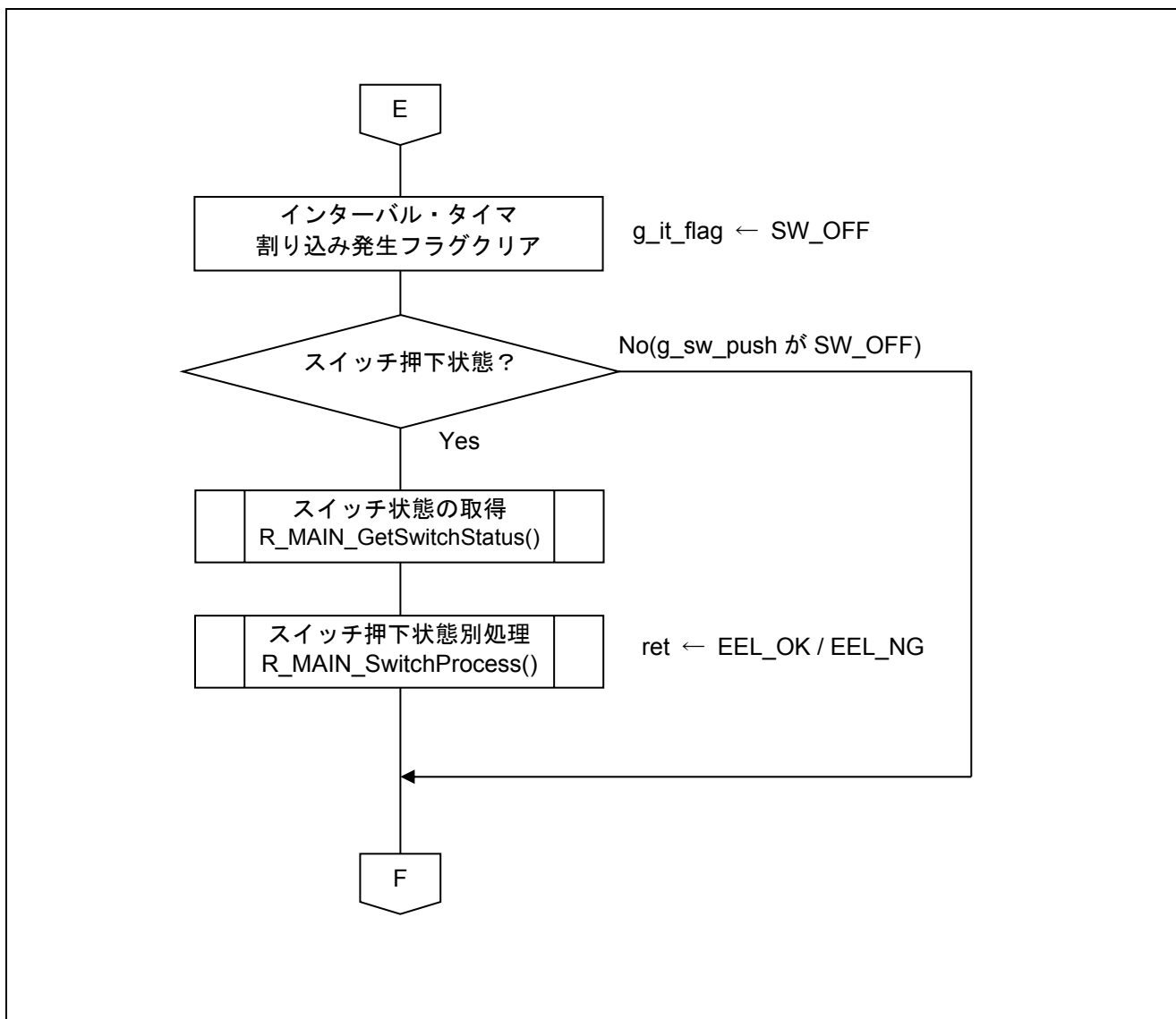


図 5.12 メイン処理(4/4)

## 5.9.9 INTP 動作開始処理

図 5.13にINTP 動作開始処理のフローチャートを示します。

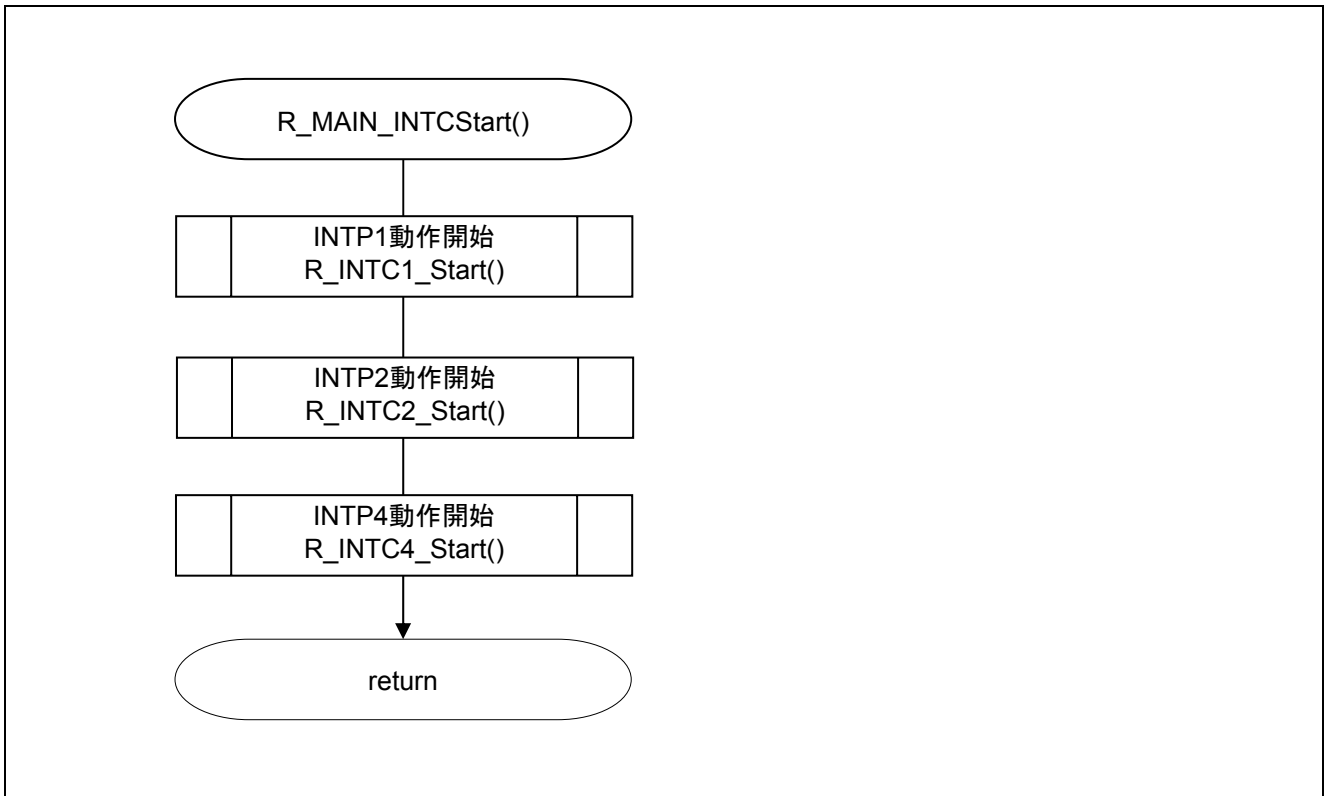


図 5.13 INTP 動作開始処理

## 5.9.10 INTP1 動作開始処理

図 5.14にINTP1 動作開始処理のフローチャートを示します。

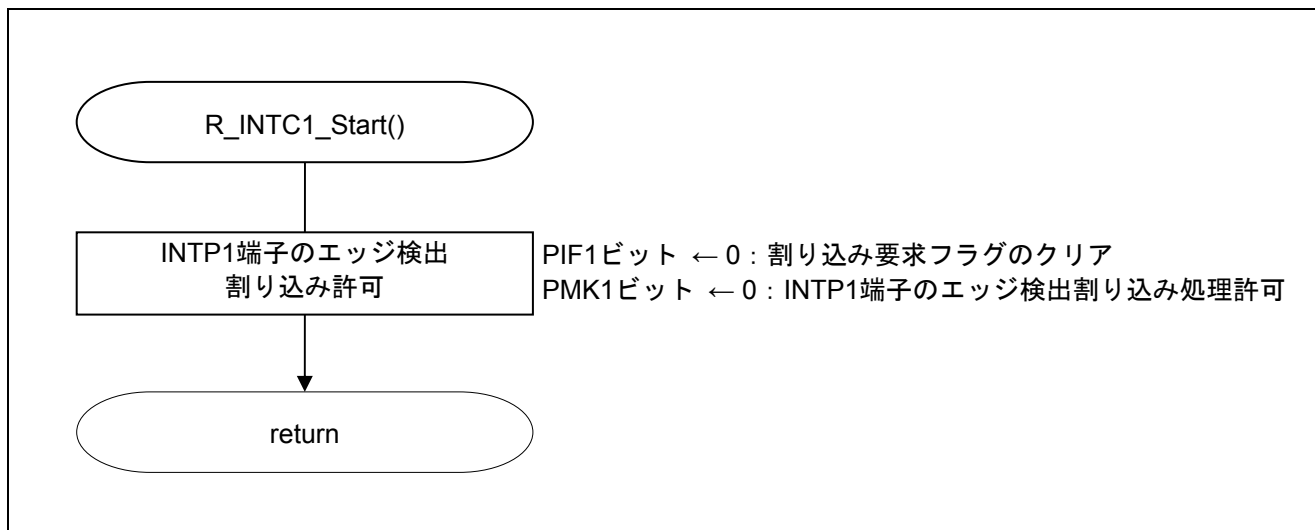


図 5.14 INTP1 動作開始処理

## 5.9.11 INTP1 外部割り込み

図 5.15にINTP1 外部割り込みのフローチャートを示します。

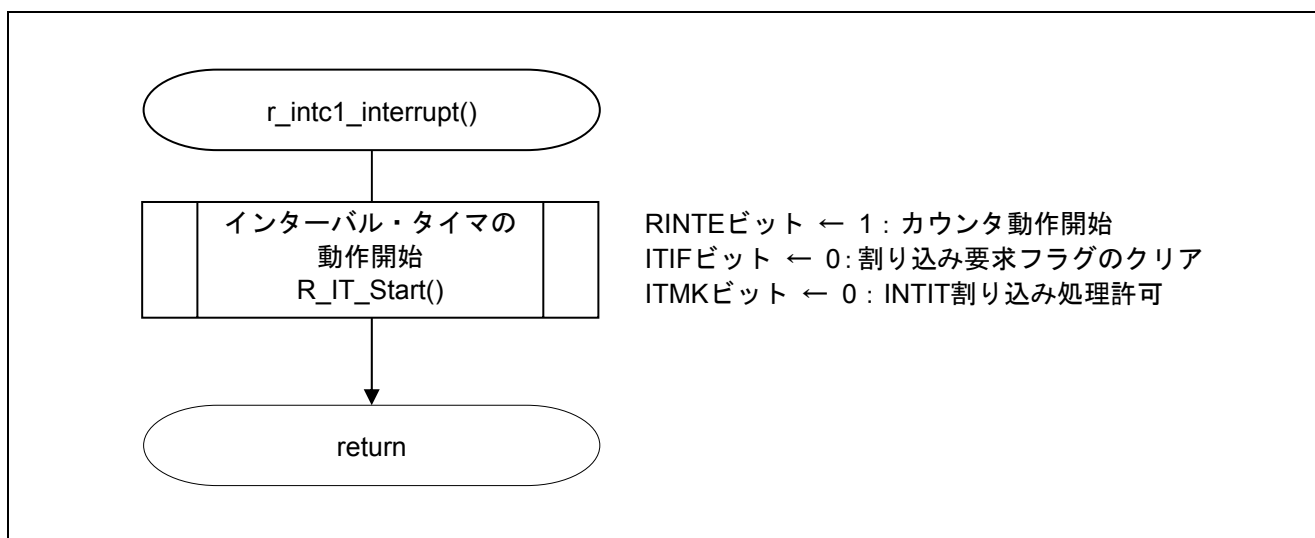


図 5.15 INTP1 外部割り込み

## 5.9.12 INTP2 動作開始処理

図 5.16にINTP2 動作開始処理のフローチャートを示します。

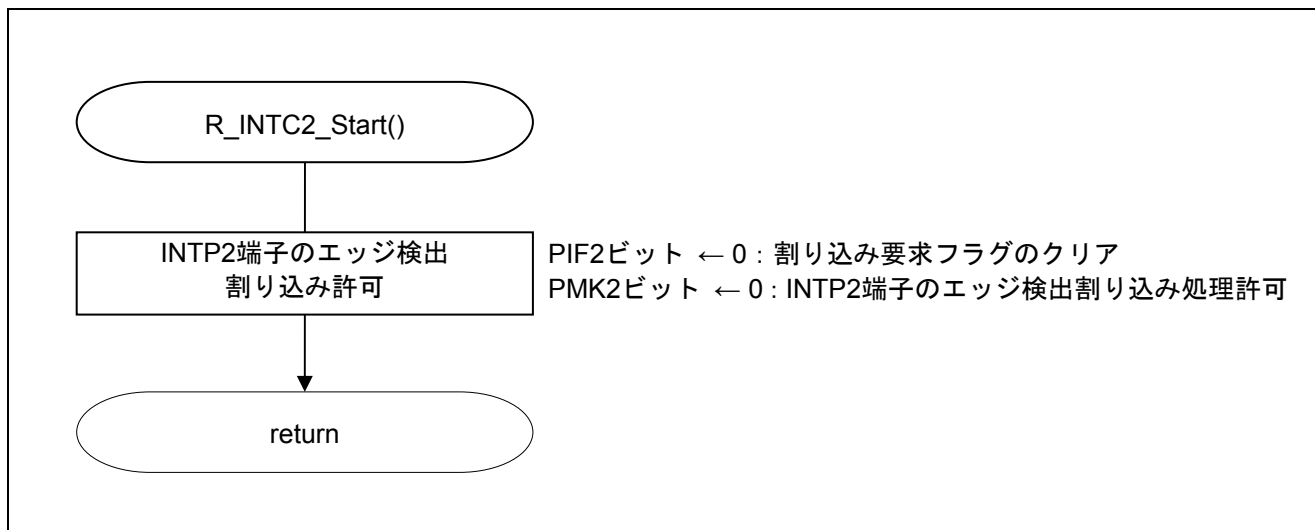


図 5.16 INTP2 動作開始処理

## 5.9.13 INTP2 外部割り込み

図 5.17にINTP2 外部割り込みのフローチャートを示します。

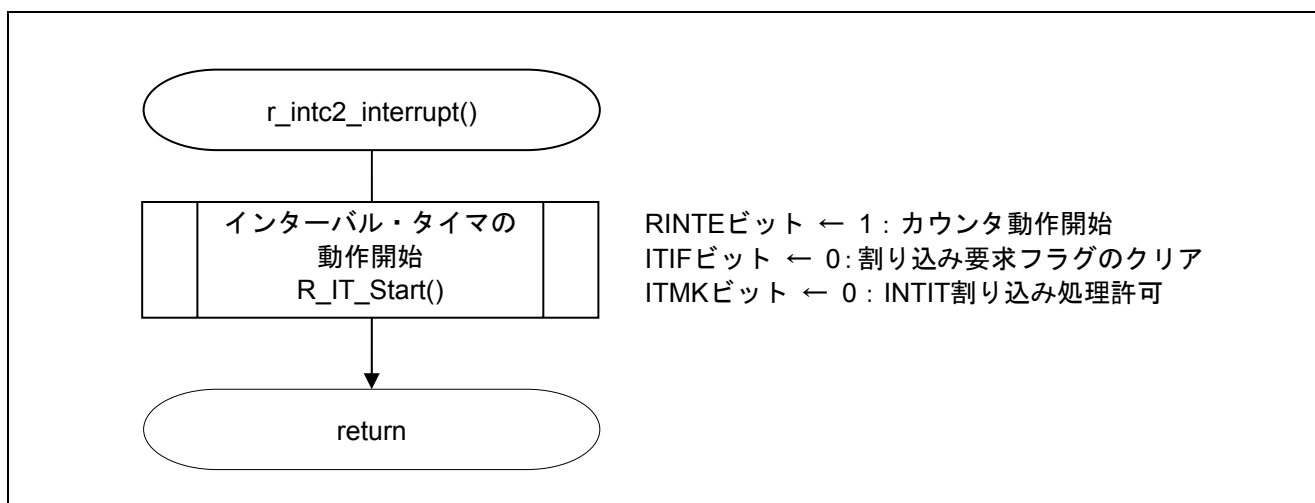


図 5.17 INTP2 外部割り込み

## 5.9.14 INTP4 動作開始処理

図 5.18にINTP4 動作開始処理のフローチャートを示します。

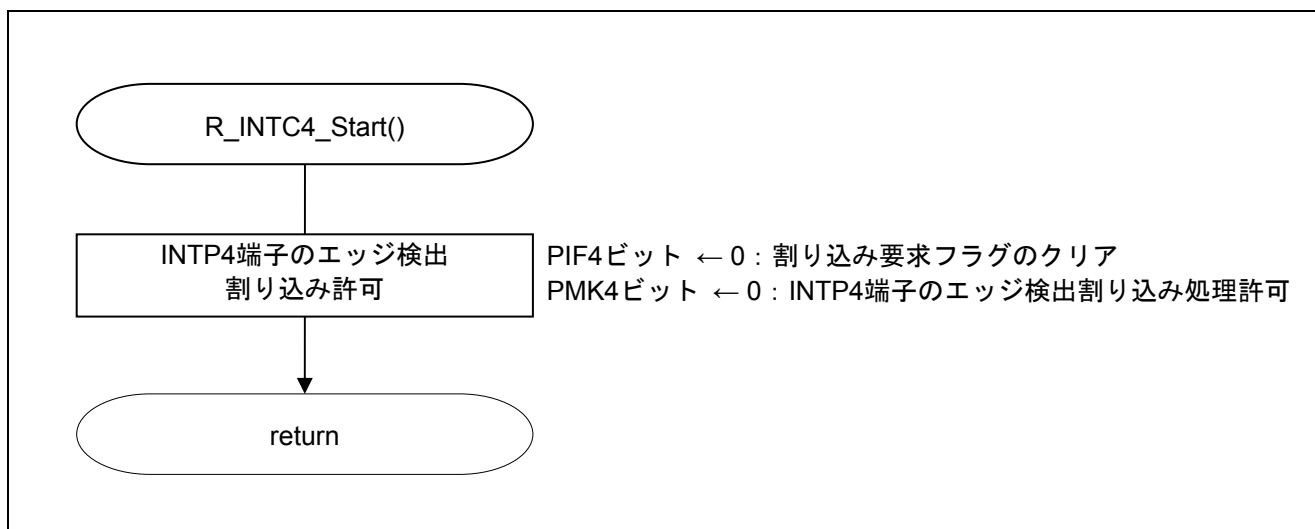


図 5.18 INTP4 動作開始処理

## 5.9.15 INTP4 外部割り込み

図 5.19にINTP4 外部割り込みのフローチャートを示します。

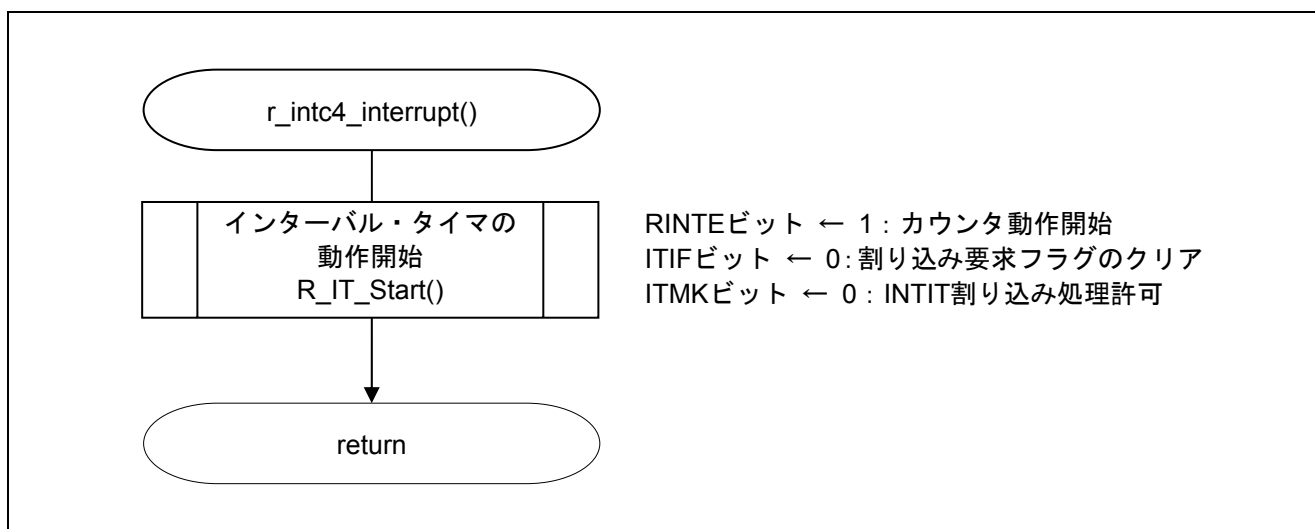


図 5.19 INTP4 外部割り込み



## 5.9.16 インターバル・タイマ動作開始処理

図 5.20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。

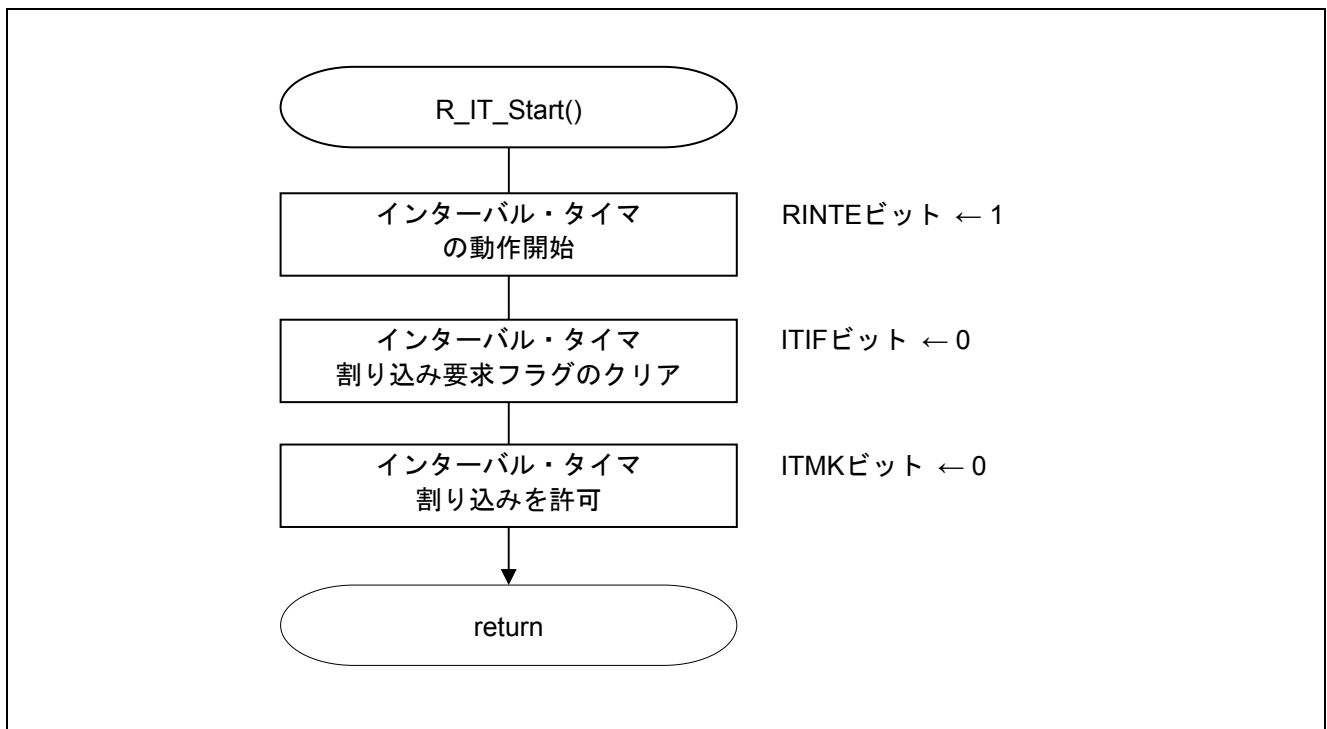


図 5.20 インターバル・タイマ動作開始処理

5.9.17 インターバル・タイマ割り込み

図 5.21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。

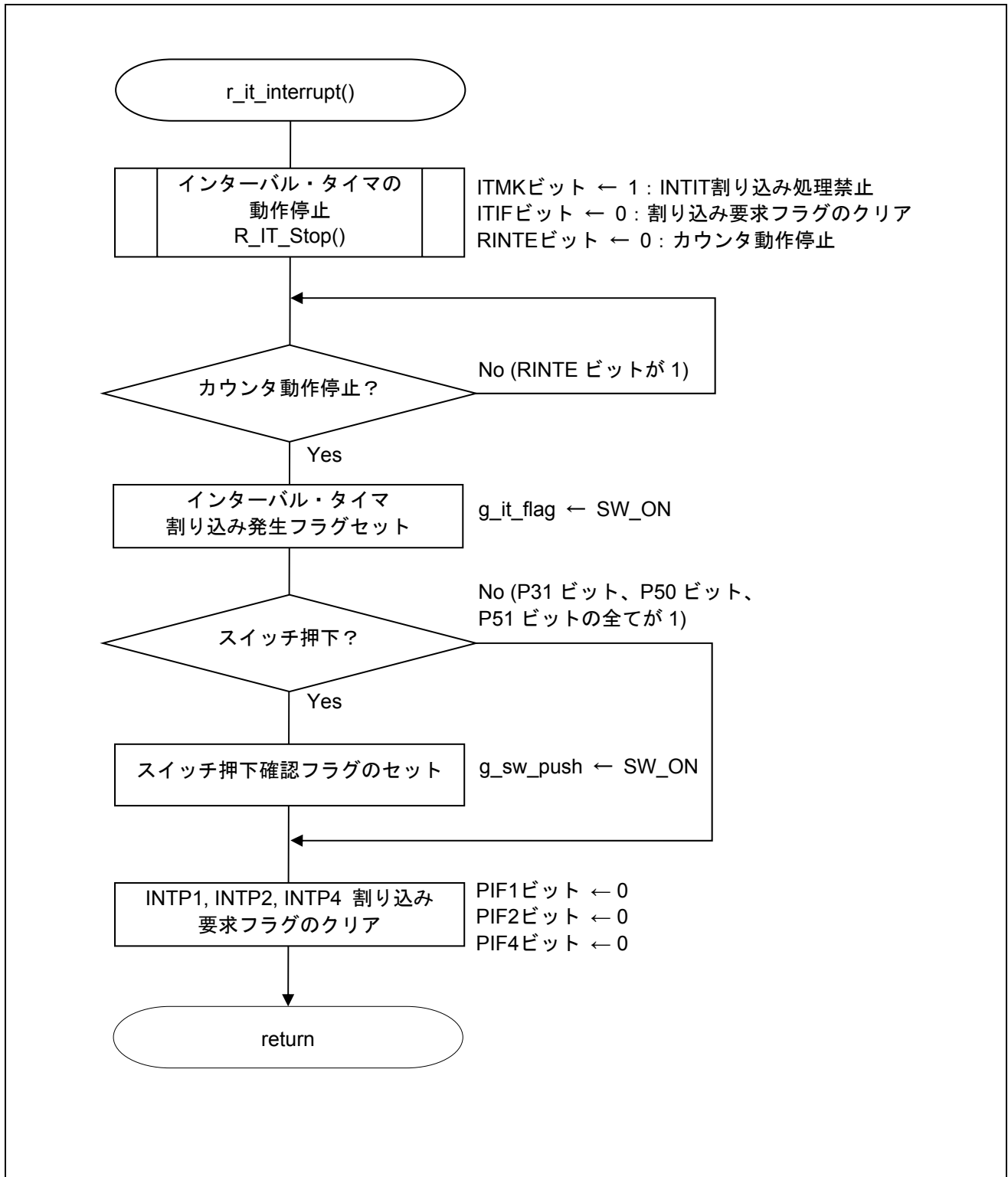


図 5.21 インターバル・タイマ割り込み

## 5.9.18 インターバル・タイマ動作停止処理

図 5.22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。

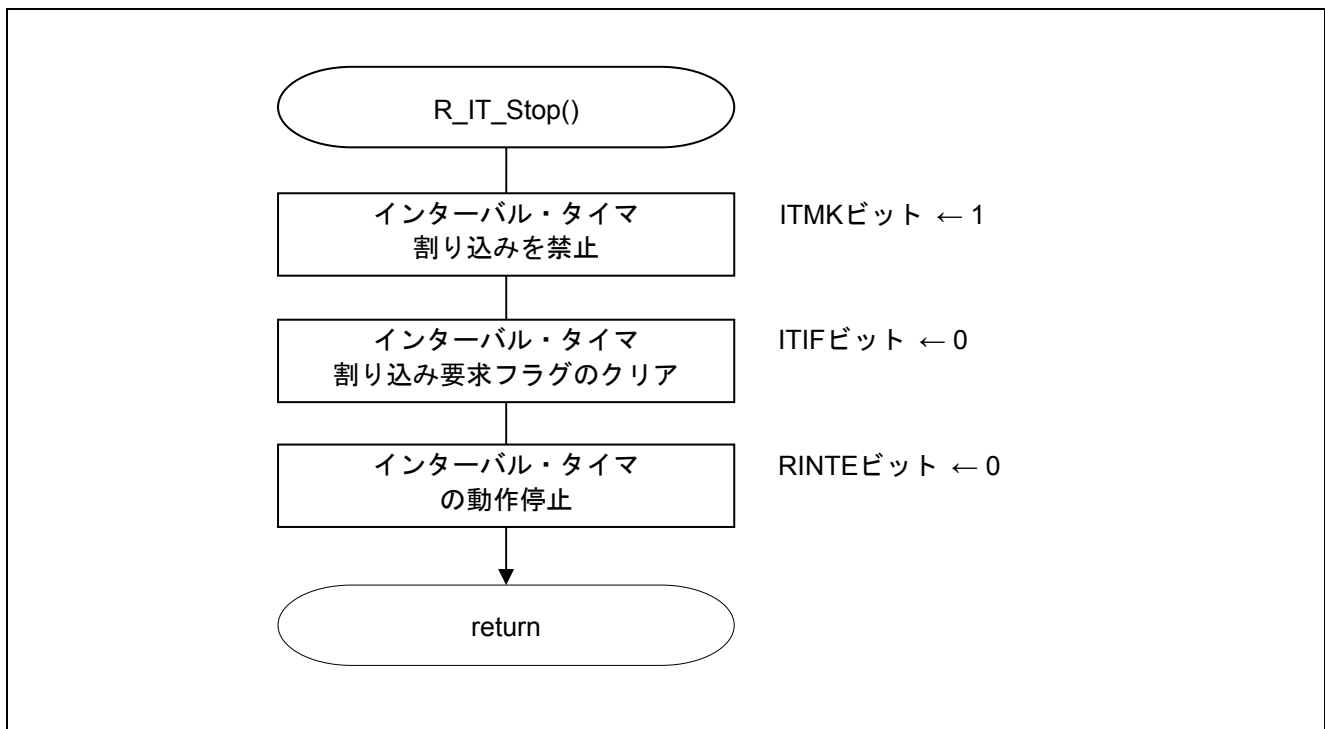


図 5.22 インターバル・タイマ動作停止処理

## 5.9.19 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。

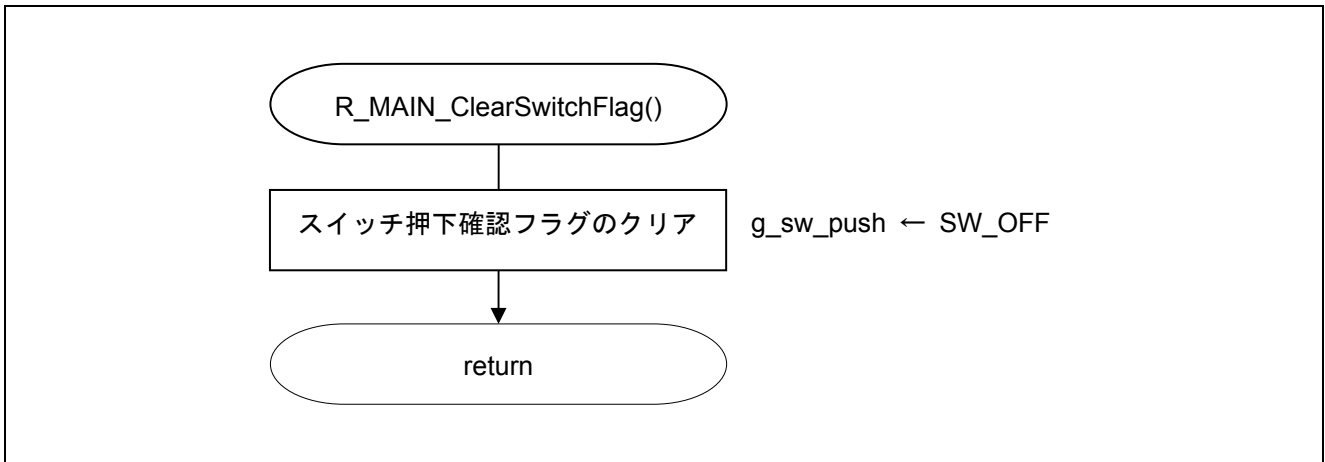


図 5.23 スイッチ押下状態のクリア

5.9.20 EEL 初期設定

図 5.24と図 5.25に EEL 初期設定のフローチャートを示します。

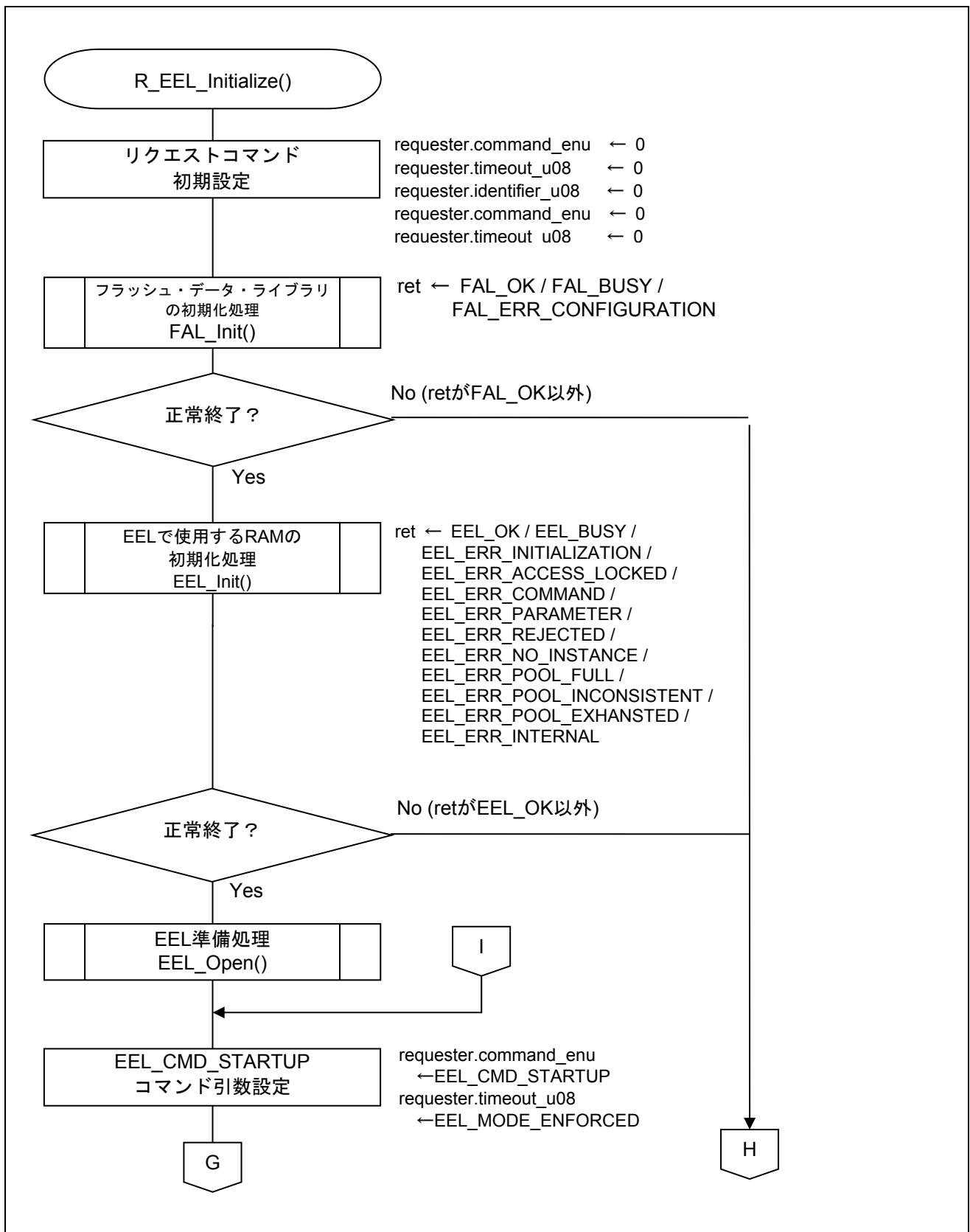


図 5.24 EEL 初期設定(1/2)

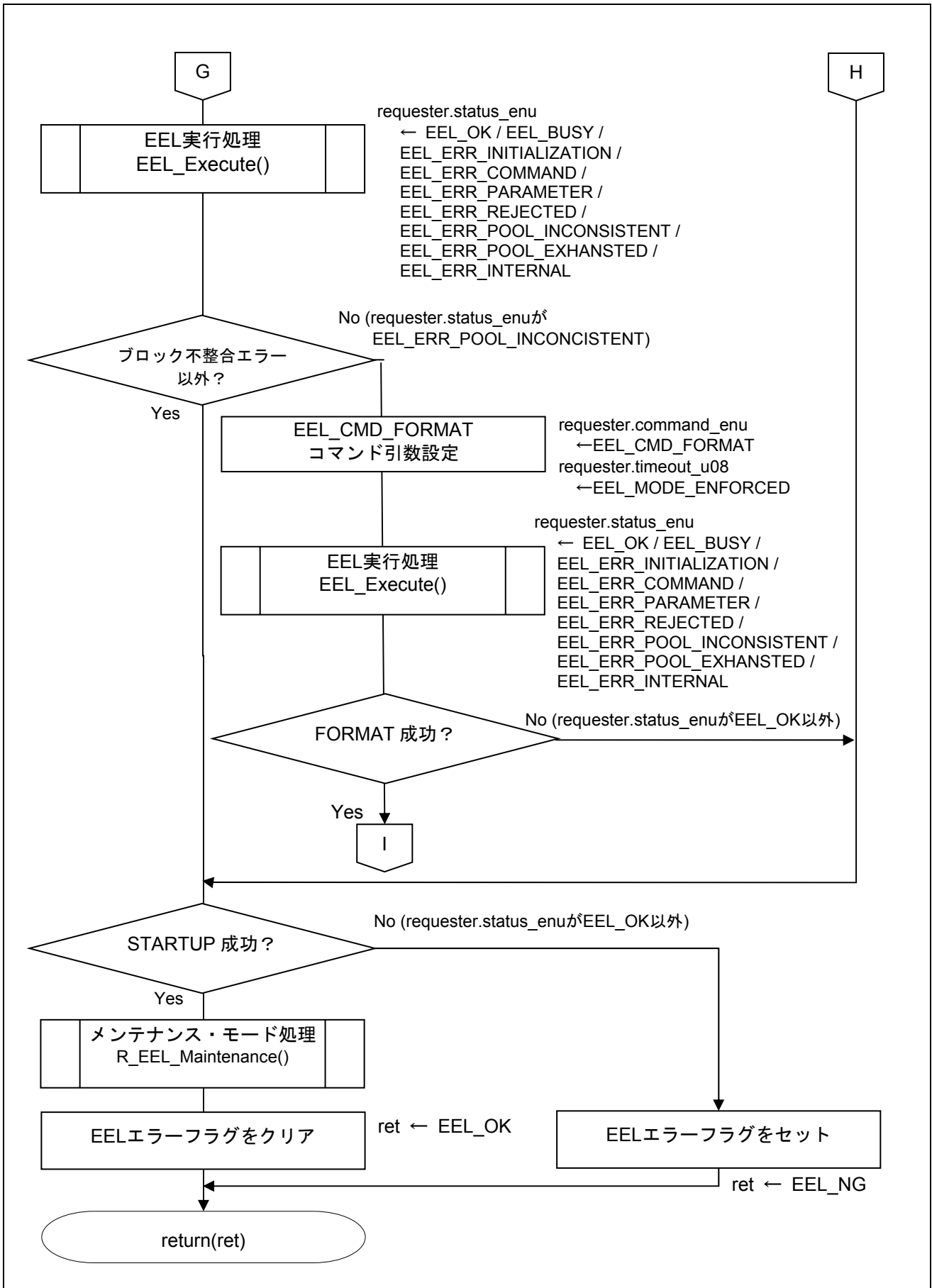


図 5.25 EEL 初期設定(2/2)

5.9.21 メンテナンス・モード処理

図 5.26にメンテナンス・モード処理のフローチャートを示します。

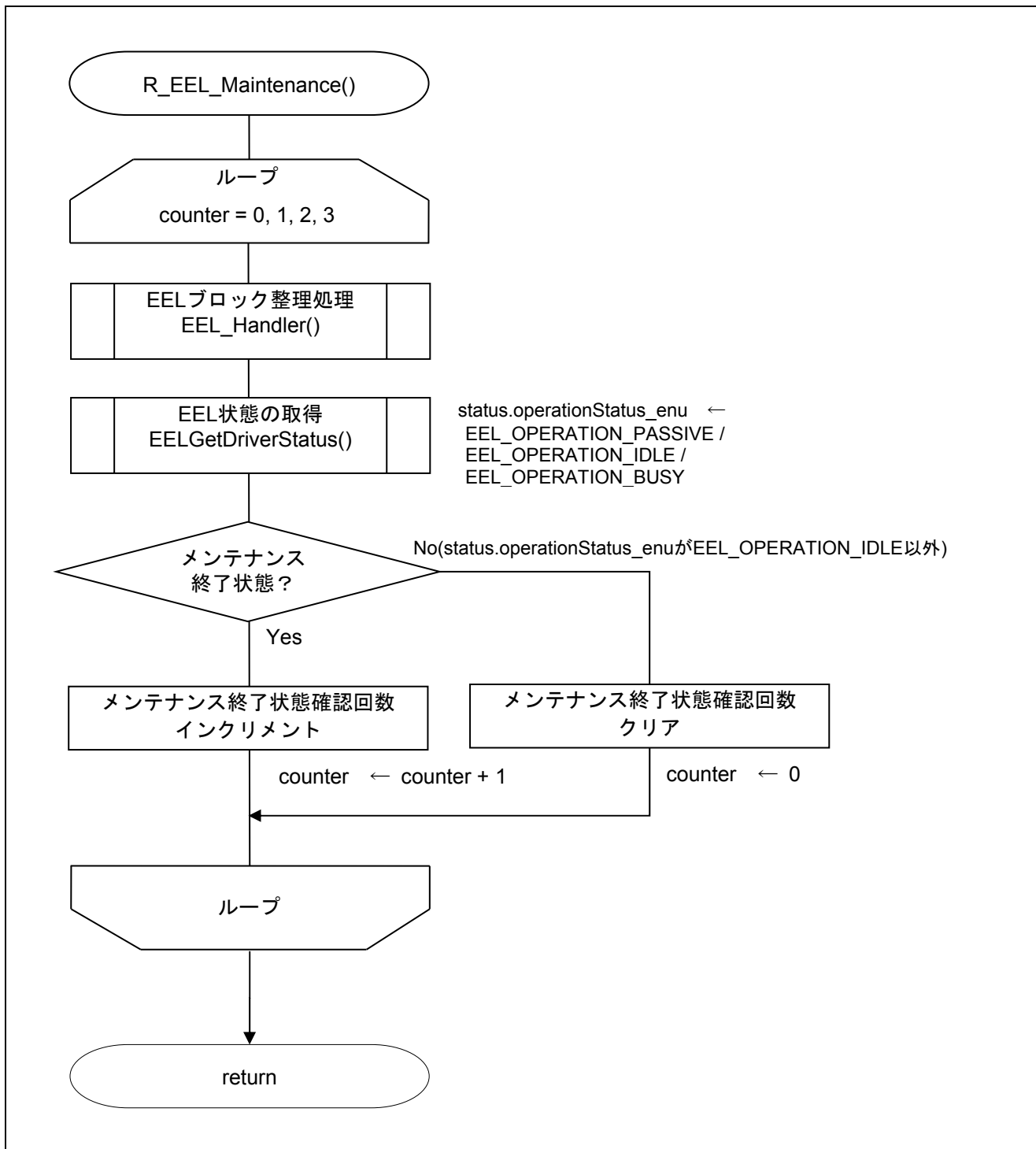


図 5.26 メンテナンス・モード処理

## 5.9.22 スイッチ状態の取得

図 5.27にスイッチ状態の取得のフローチャートを示します。

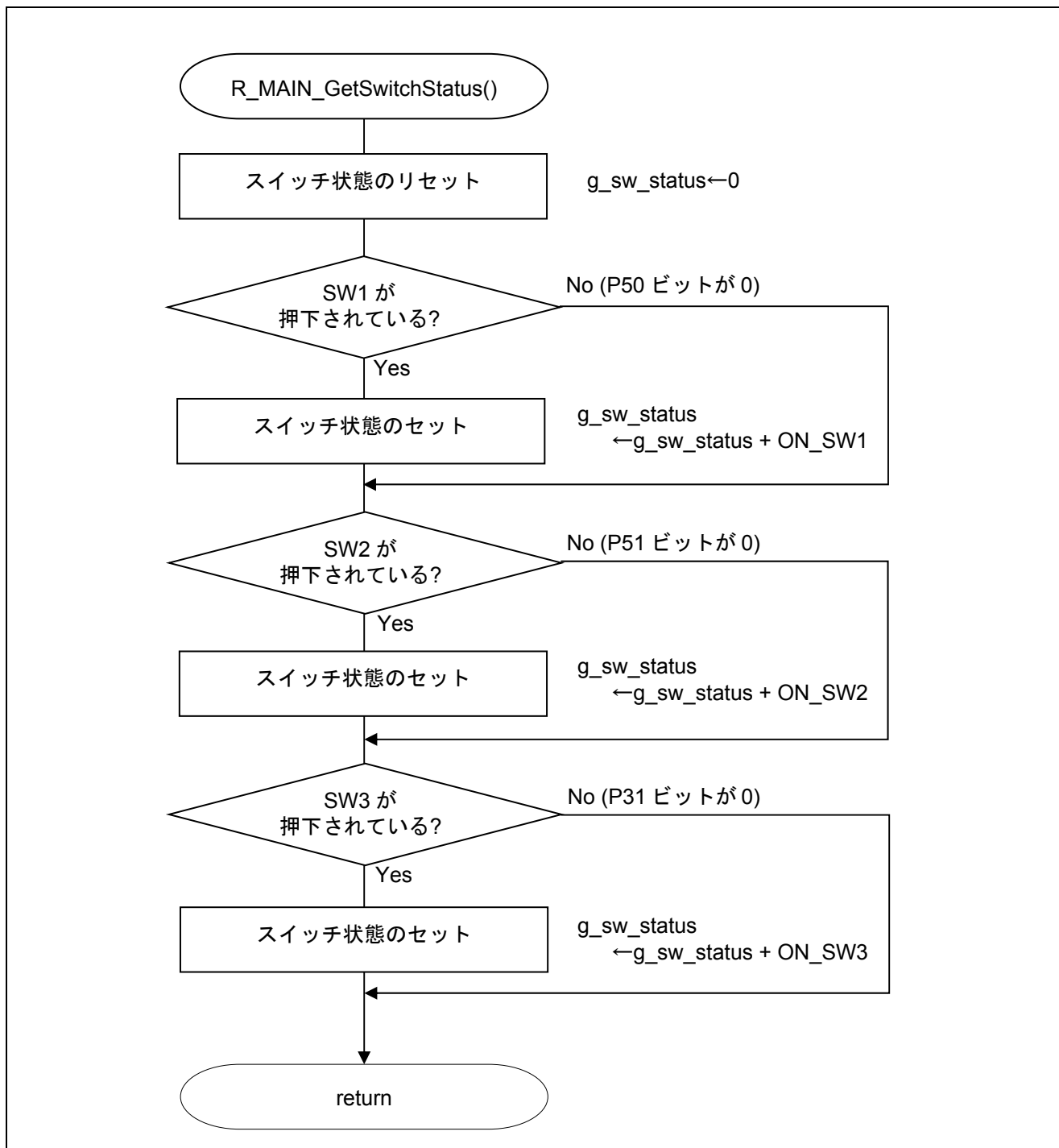


図 5.27 スイッチ状態の取得



5.9.23 スイッチ押下状態別処理

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。

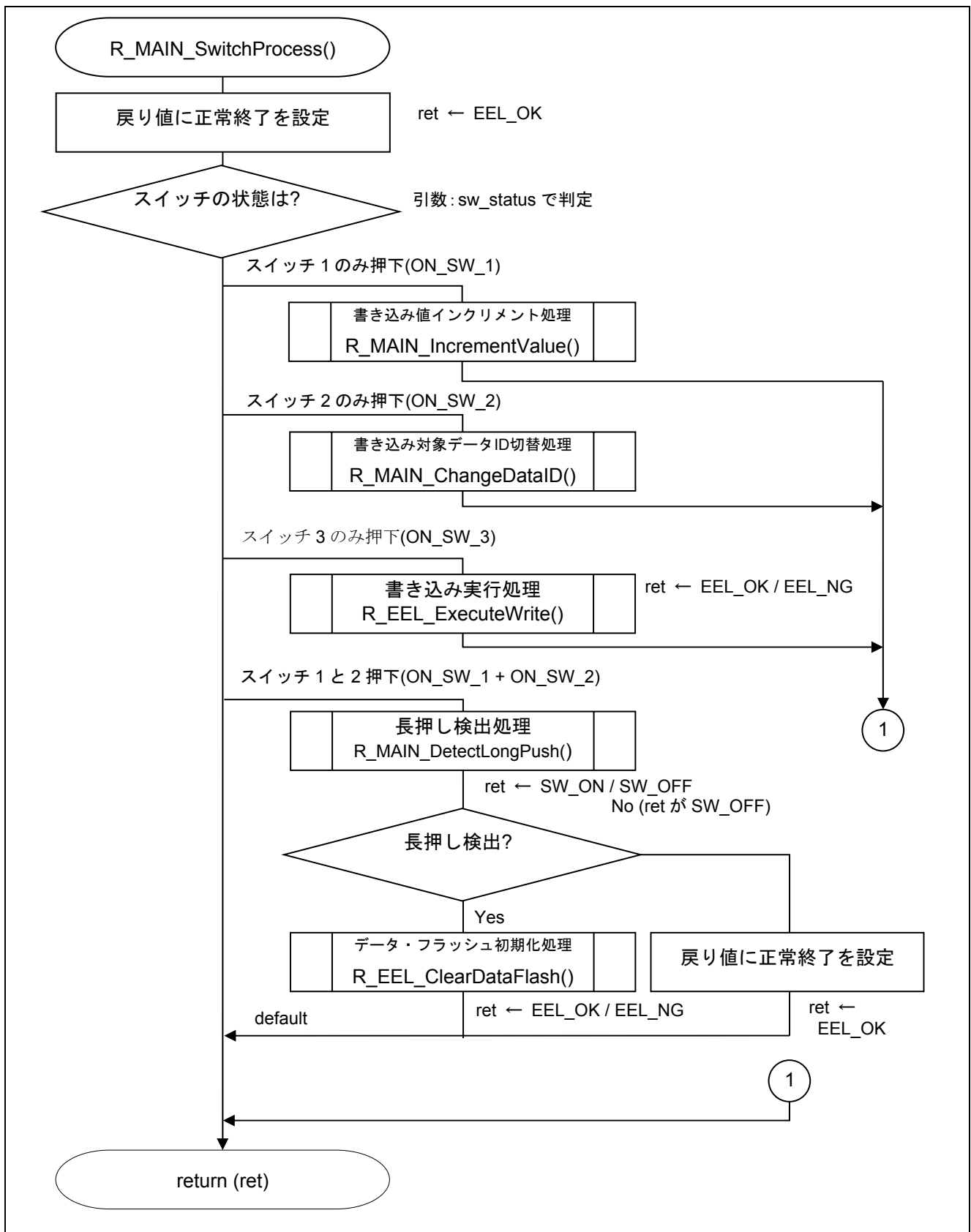


図 5.28 スイッチ押下状態別処理

## 5.9.24 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。

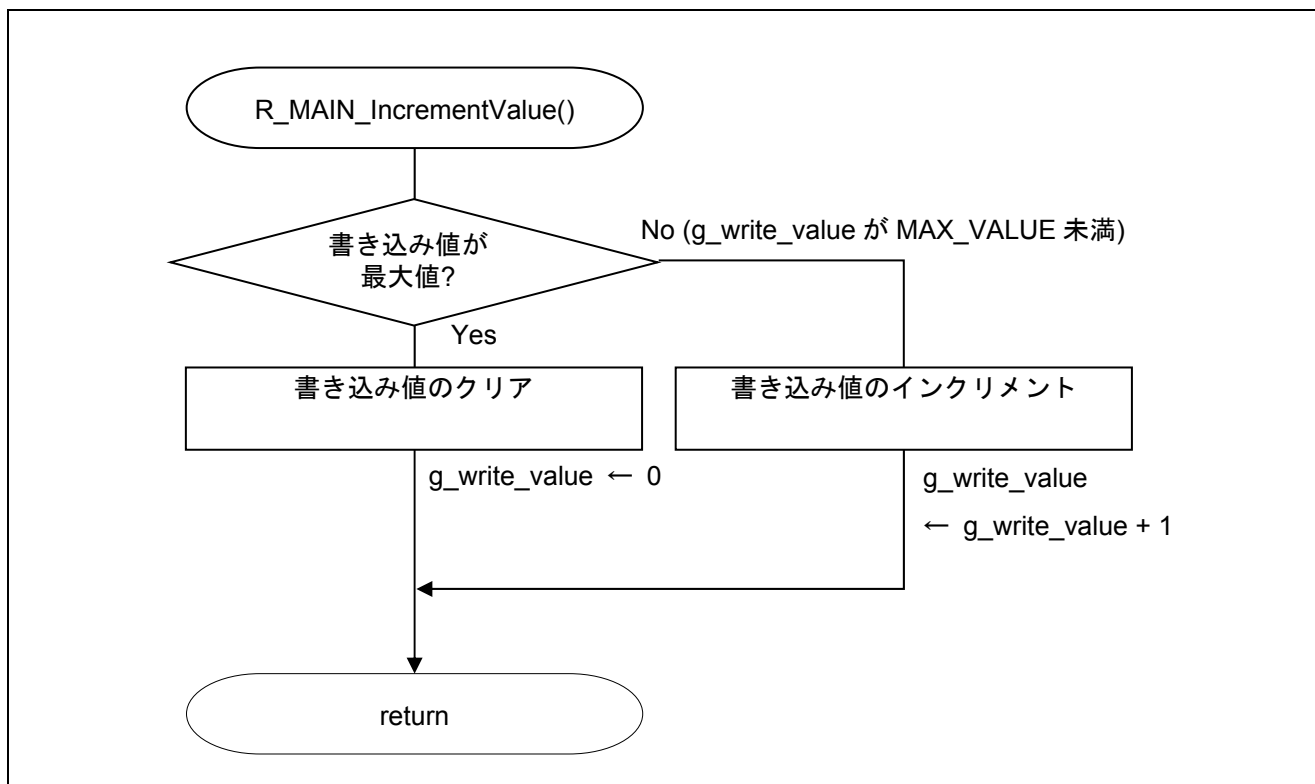


図 5.29 書き込み値インクリメント処理

## 5.9.25 書き込み対象データ ID 切替処理

図 5.30に書き込み対象データ ID 切替処理のフローチャートを示します。

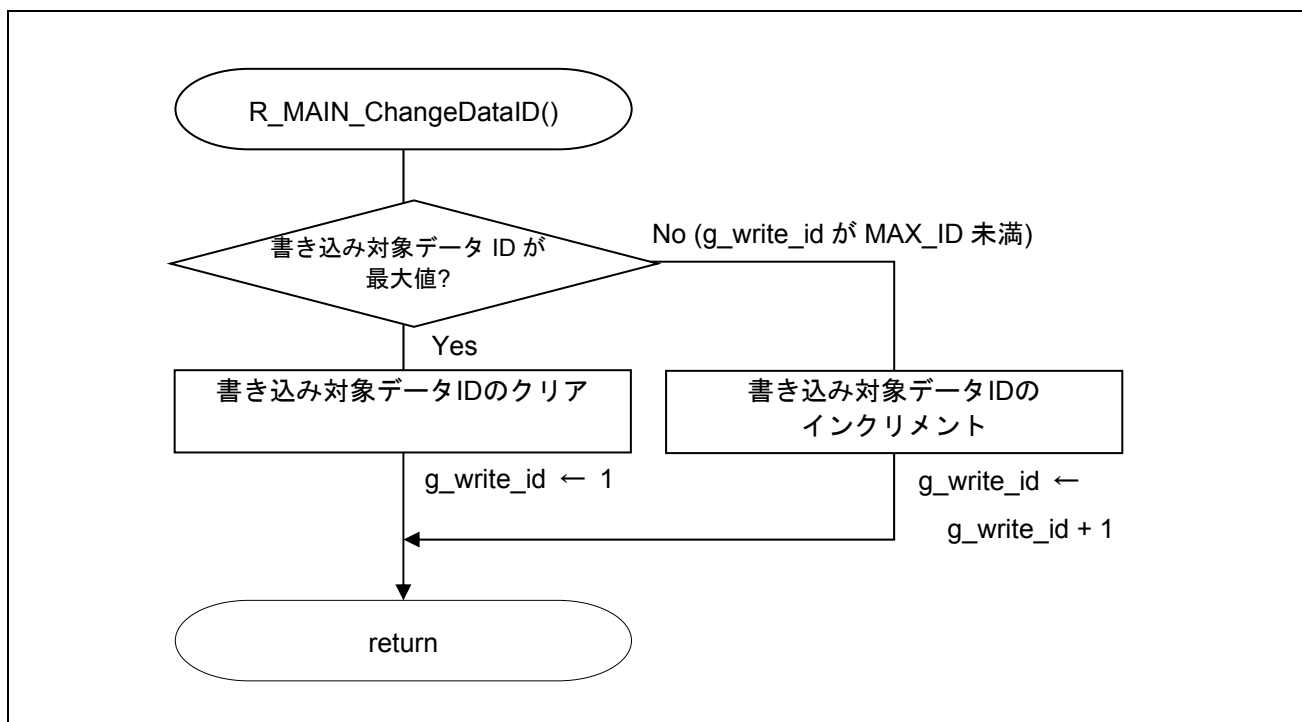


図 5.30 書き込み対象データ ID 切替処理

5.9.26 書き込み実行処理

図 5.31と図 5.32に書き込み実行処理のフローチャートを示します。

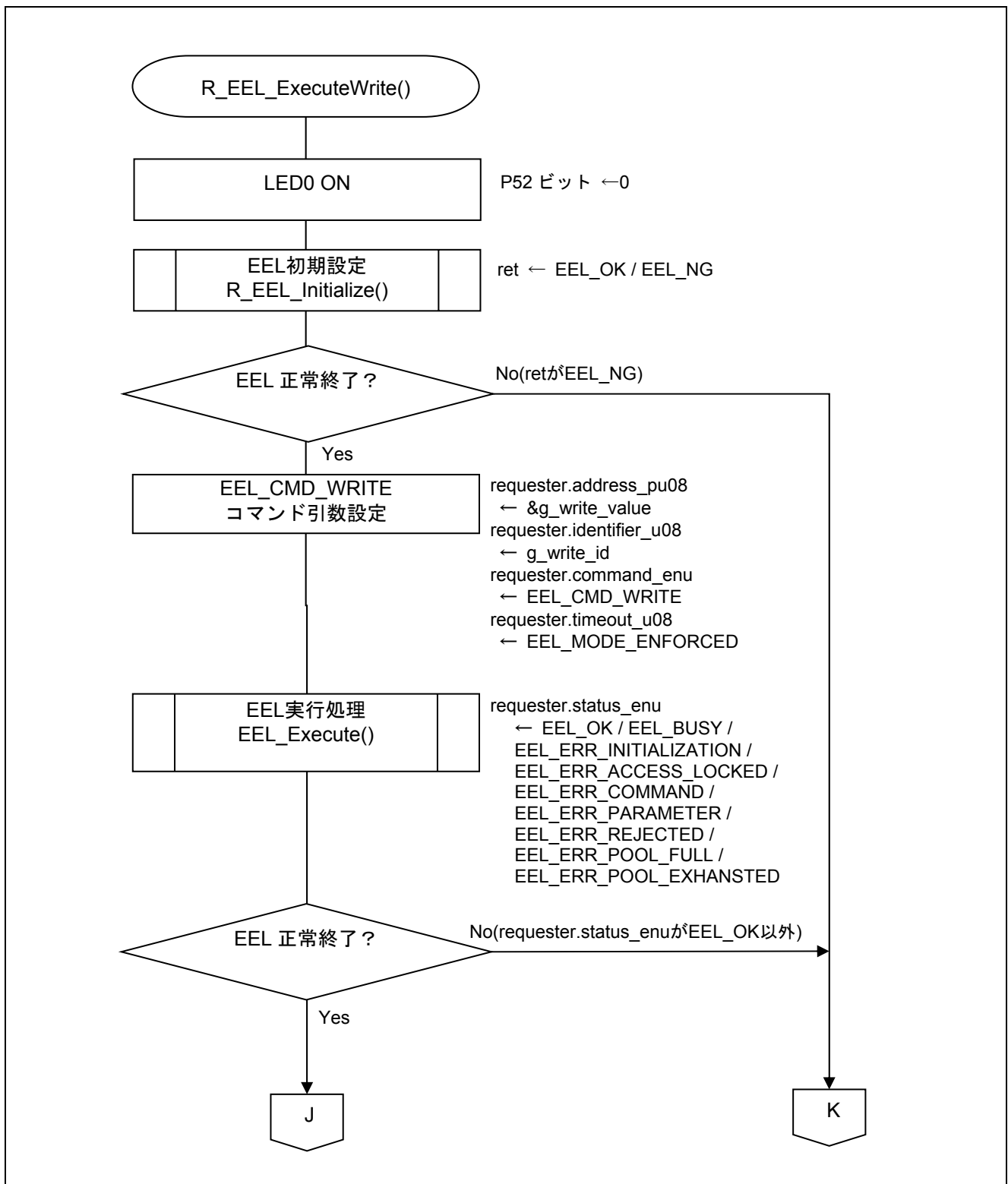


図 5.31 書き込み実行処理(1/2)

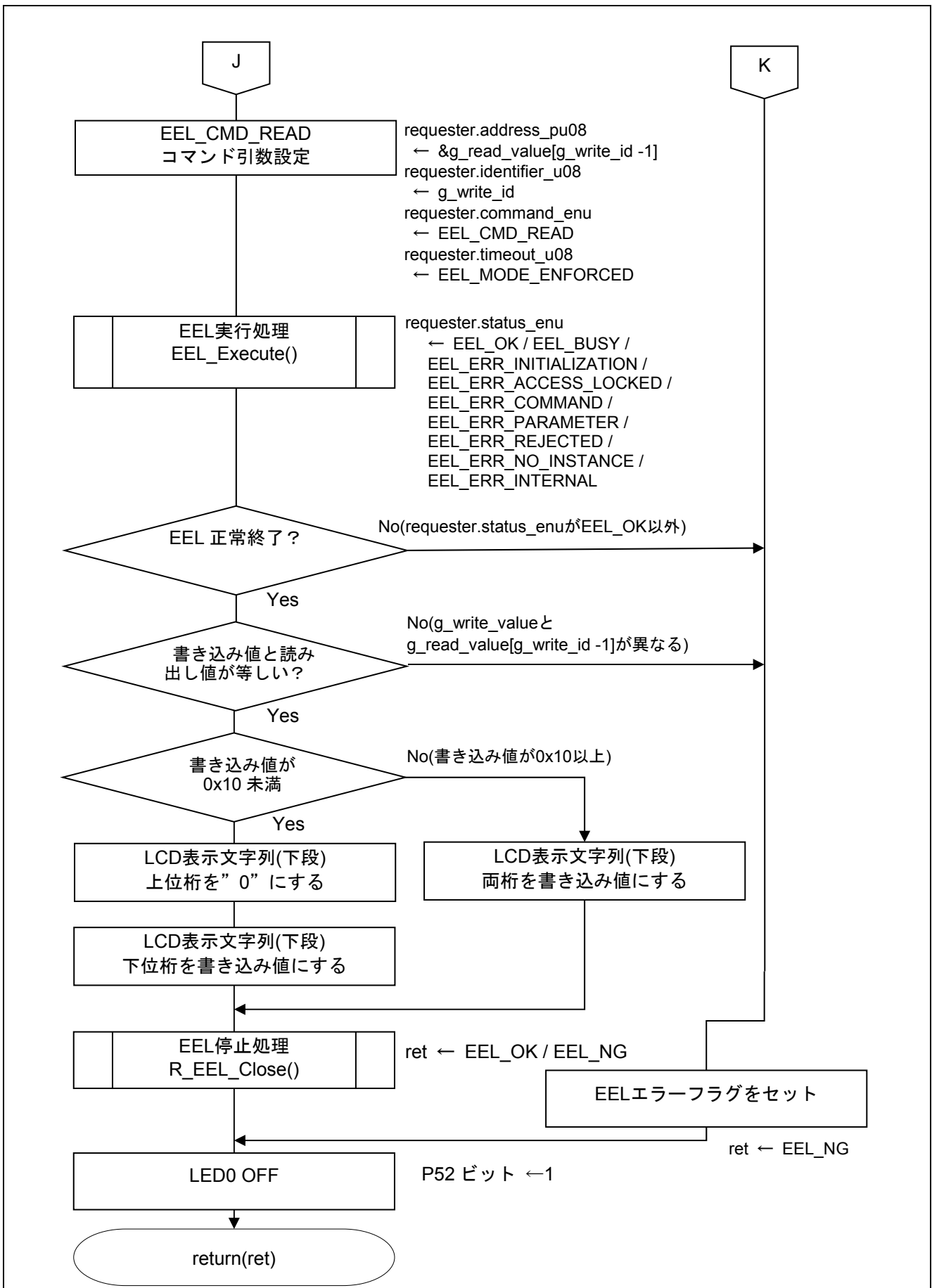


図 5.32 書き込み実行処理(2/2)

5.9.27 長押し検出処理

図 5.33と図 5.34に長押し検出処理のフローチャートを示します。

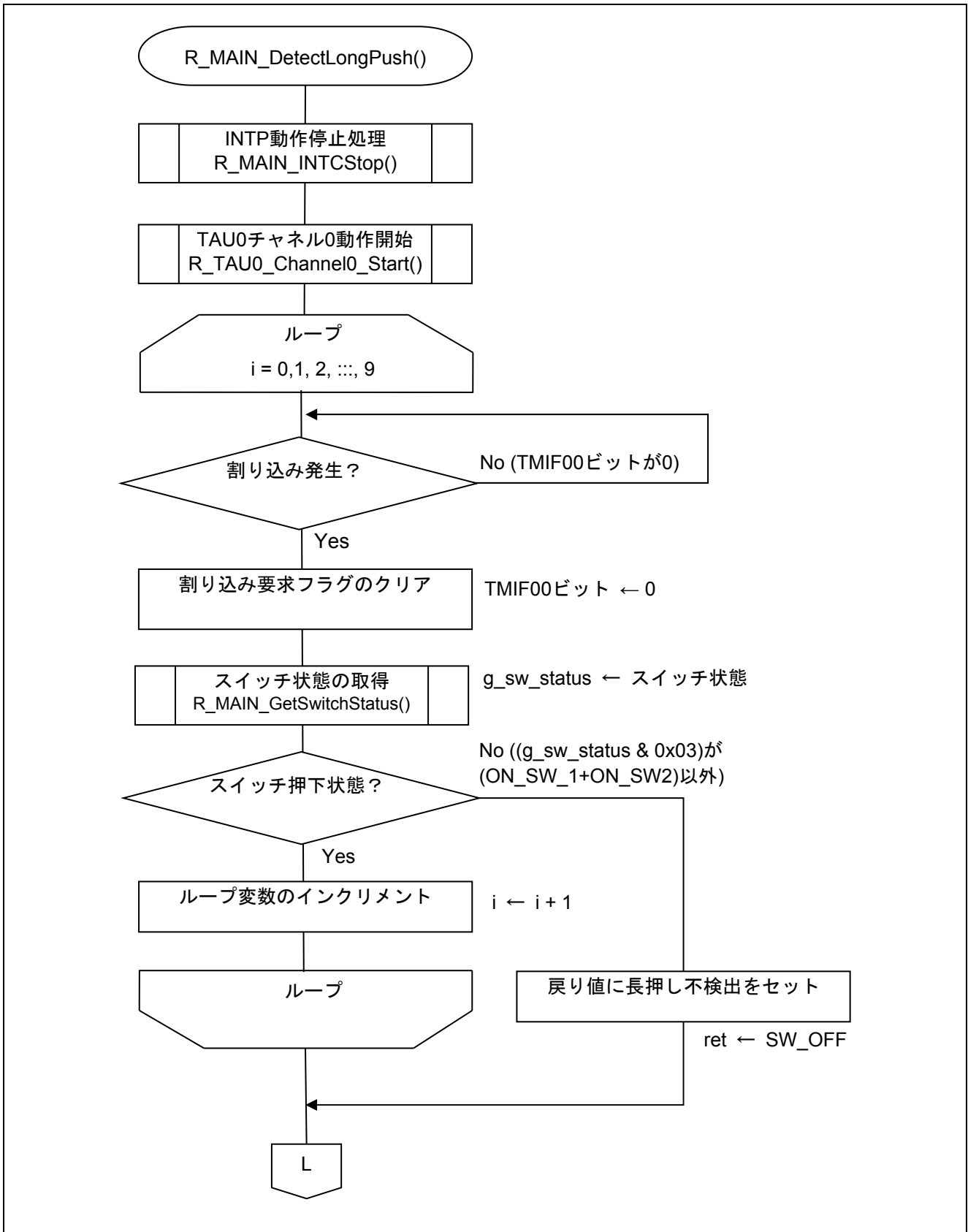


図 5.33 長押し検出処理(1/2)

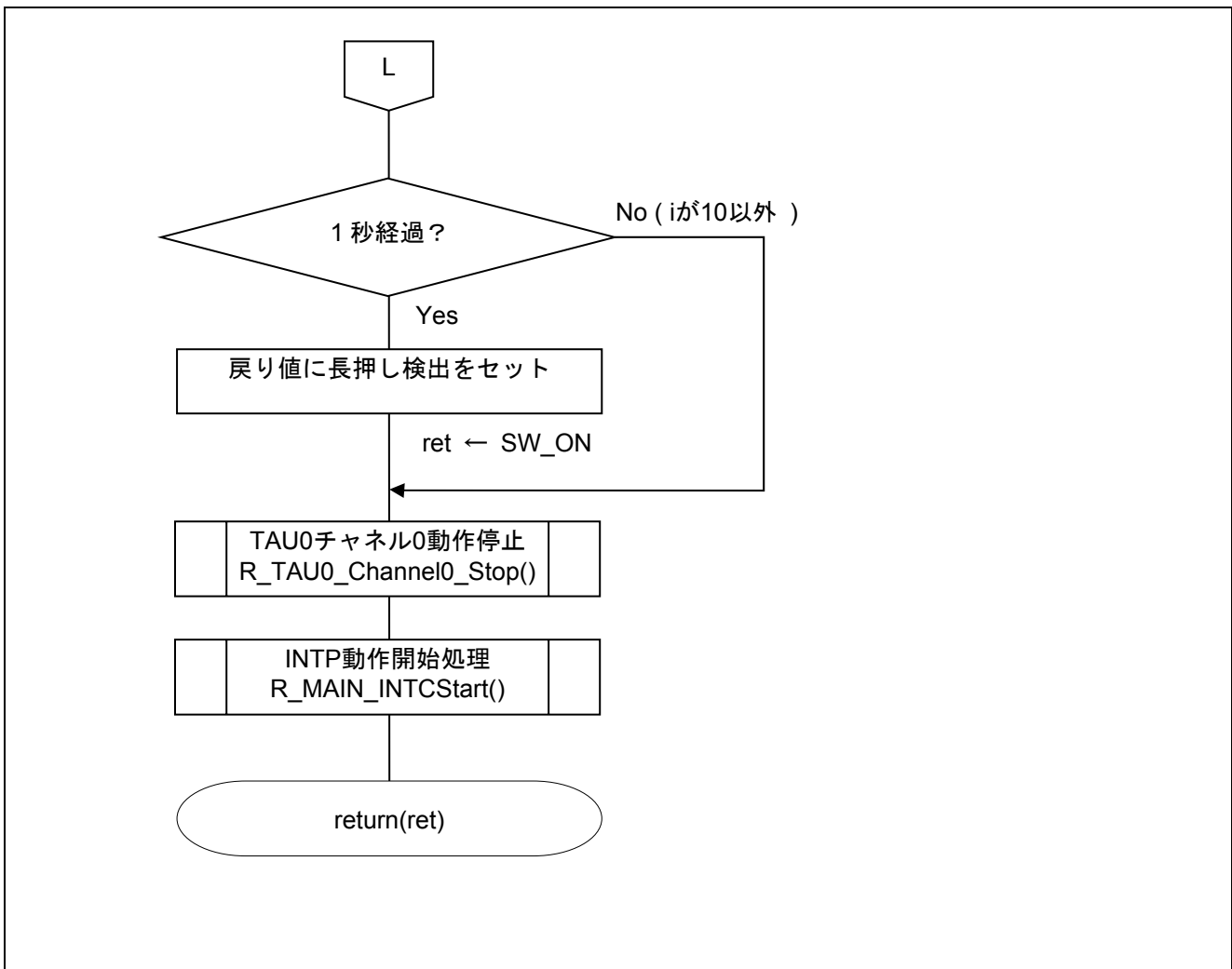


図 5.34 長押し検出処理(2/2)

## 5.9.28 INTP 動作停止処理

図 5.35にINTP 動作停止処理のフローチャートを示します。

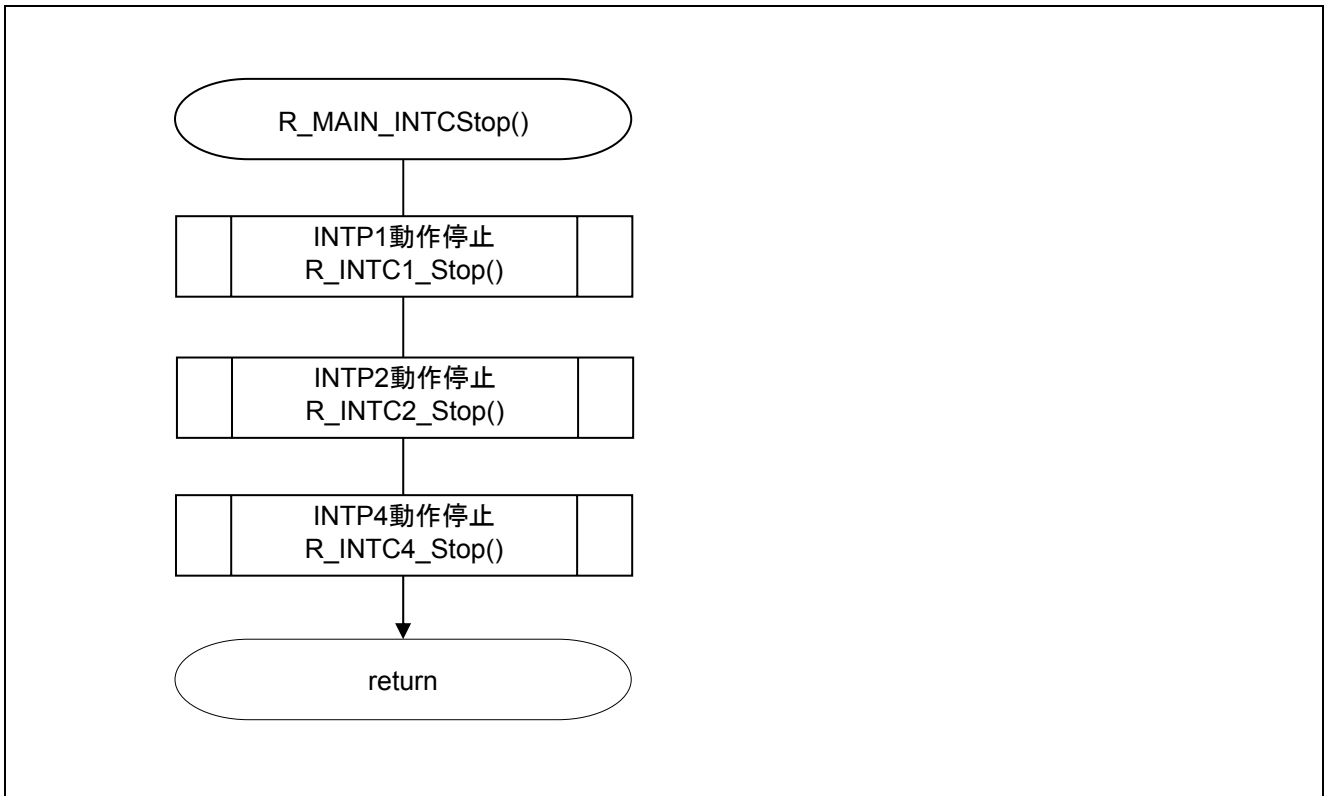


図 5.35 INTP 動作停止処理



## 5.9.29 INTP1 動作停止処理

図 5.36にINTP1 動作停止処理のフローチャートを示します。

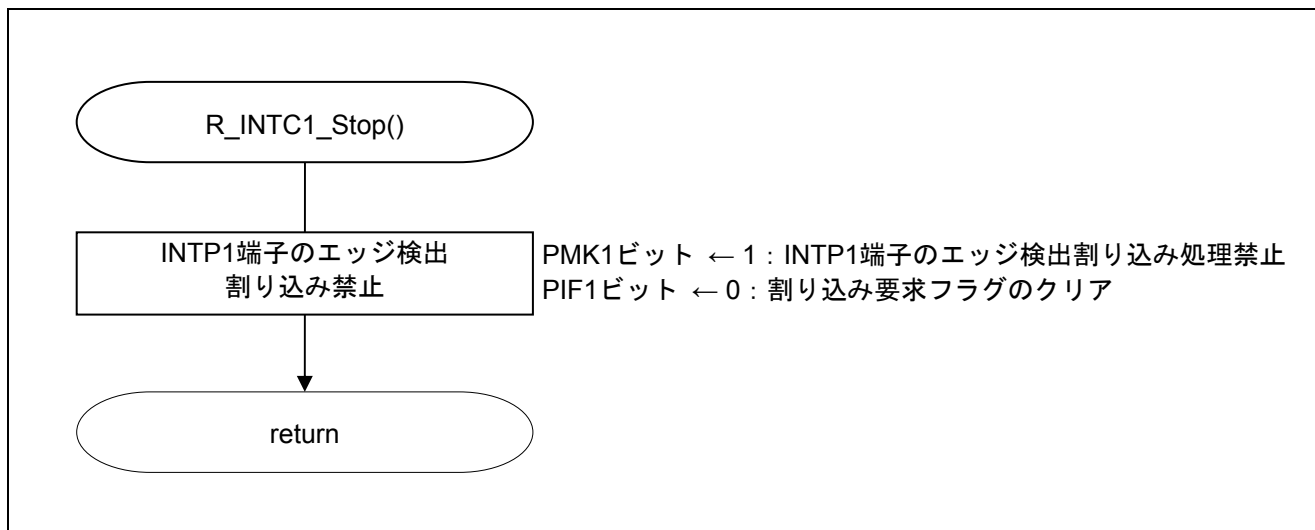


図 5.36 INTP1 動作停止処理

## 5.9.30 INTP2 動作停止処理

図 5.37にINTP2 動作停止処理のフローチャートを示します。

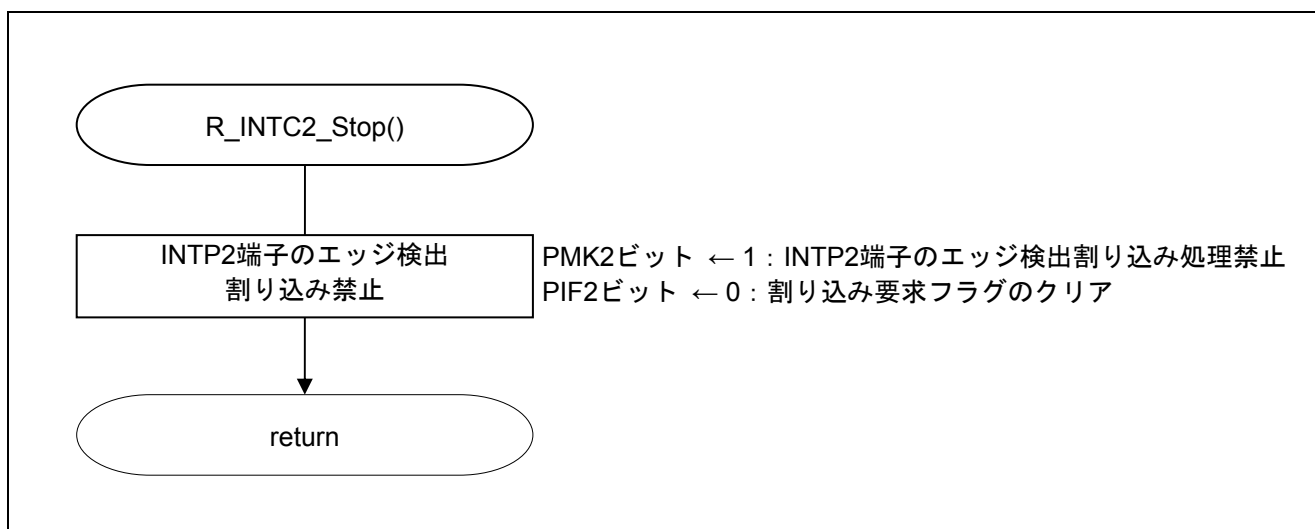


図 5.37 INTP2 動作停止処理

## 5.9.31 INTP4 動作停止処理

図 5.38にINTP4 動作停止処理のフローチャートを示します。

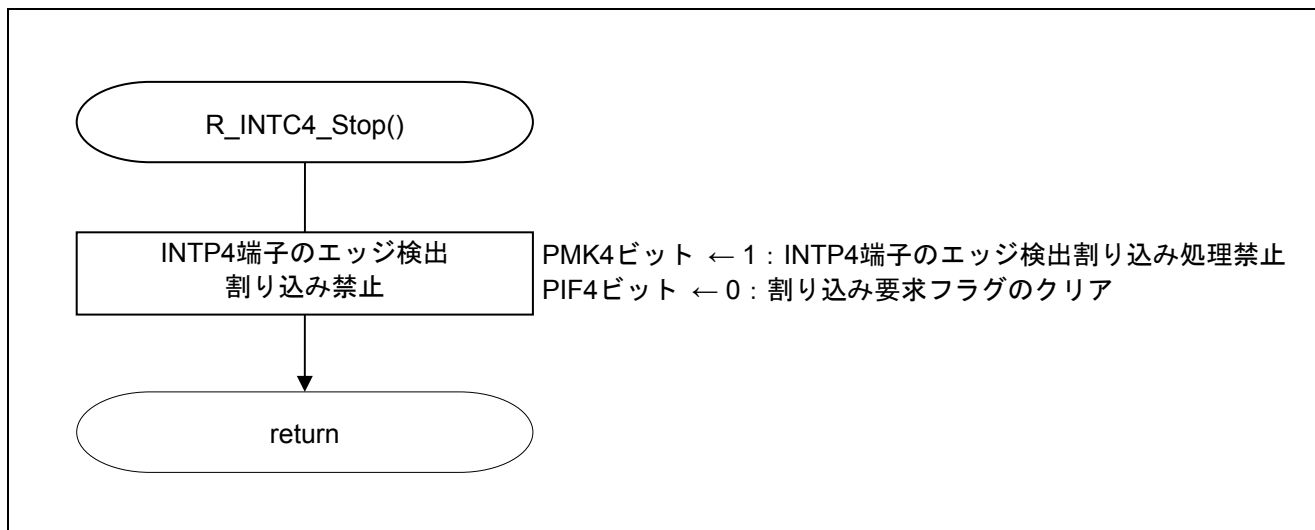


図 5.38 INTP4 動作停止処理

## 5.9.32 TAU0 チャンネル0 動作開始処理

図 5.39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。

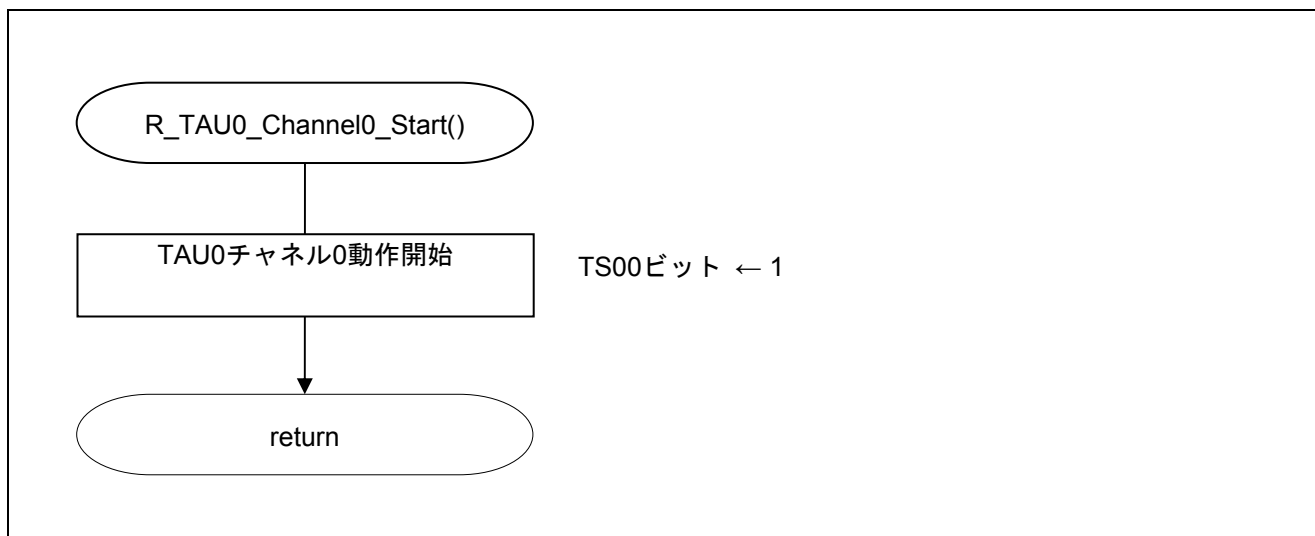


図 5.39 TAU0 チャンネル0 動作開始処理

## 5.9.33 TAU0 チャンネル0 動作停止処理

図 5.40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。

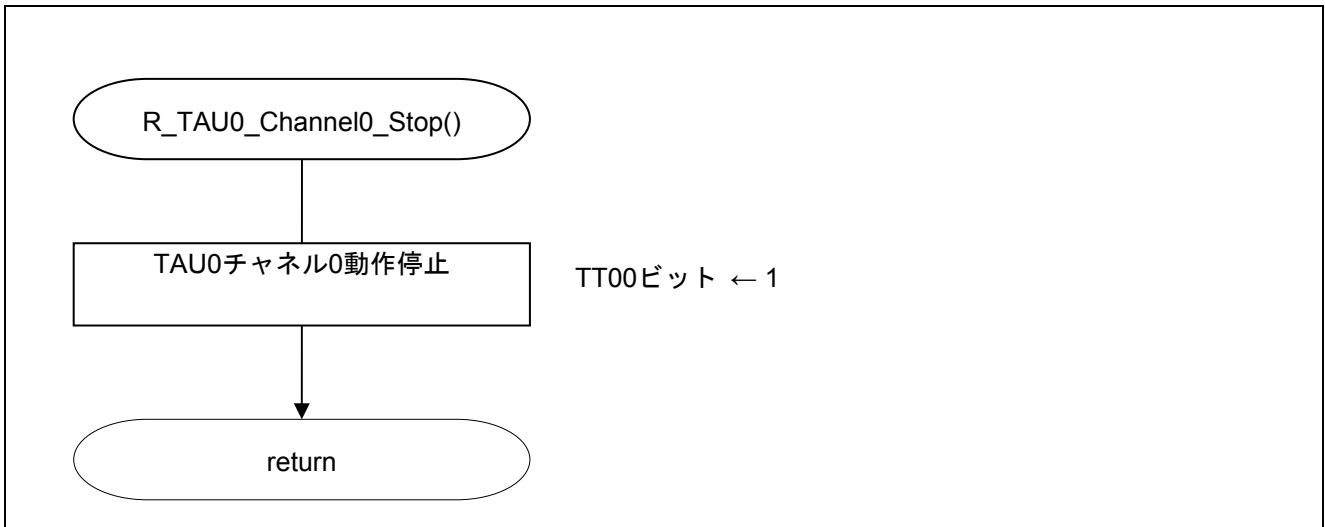


図 5.40 TAU0 チャンネル0 動作停止処理

5.9.34 データ・フラッシュ初期化処理

図 5.41にデータ・フラッシュ初期化処理(1/2)、図 5.42にデータ・フラッシュ初期化処理(2/2)のフローチャートを示します。

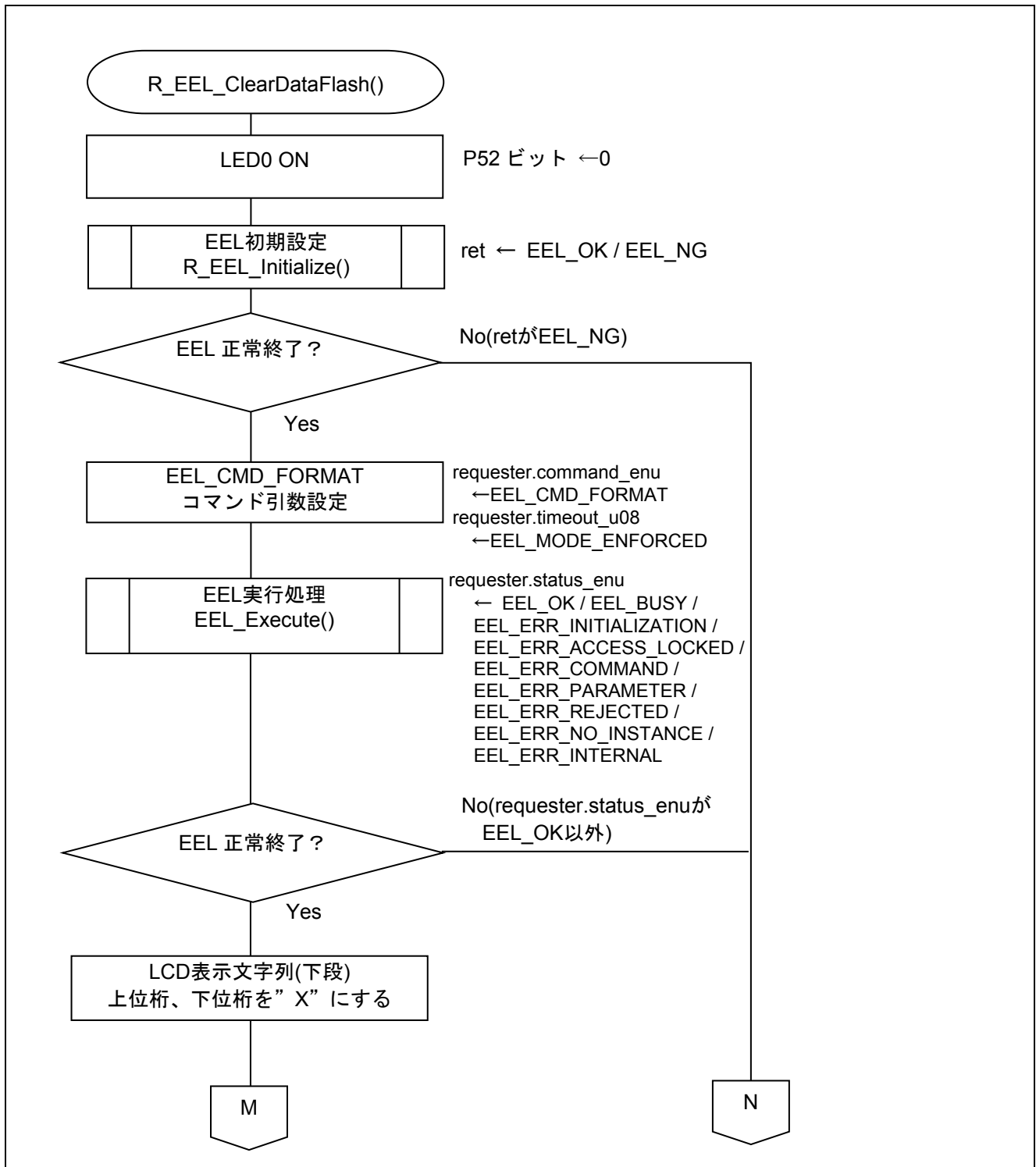


図 5.41 データ・フラッシュ初期化処理(1/2)

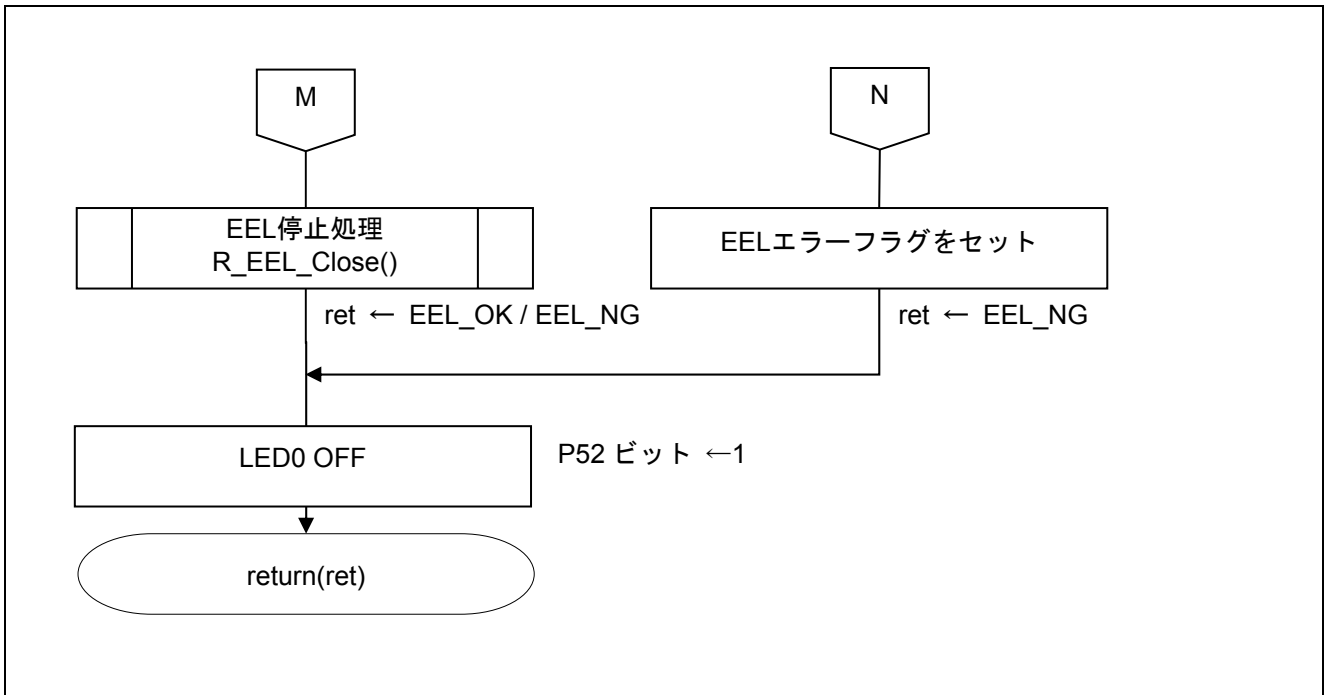


図 5.42 データ・フラッシュ初期化処理(2/2)

5.9.35 EEL 停止処理

図 5.43にEEL 停止処理のフローチャートを示します。

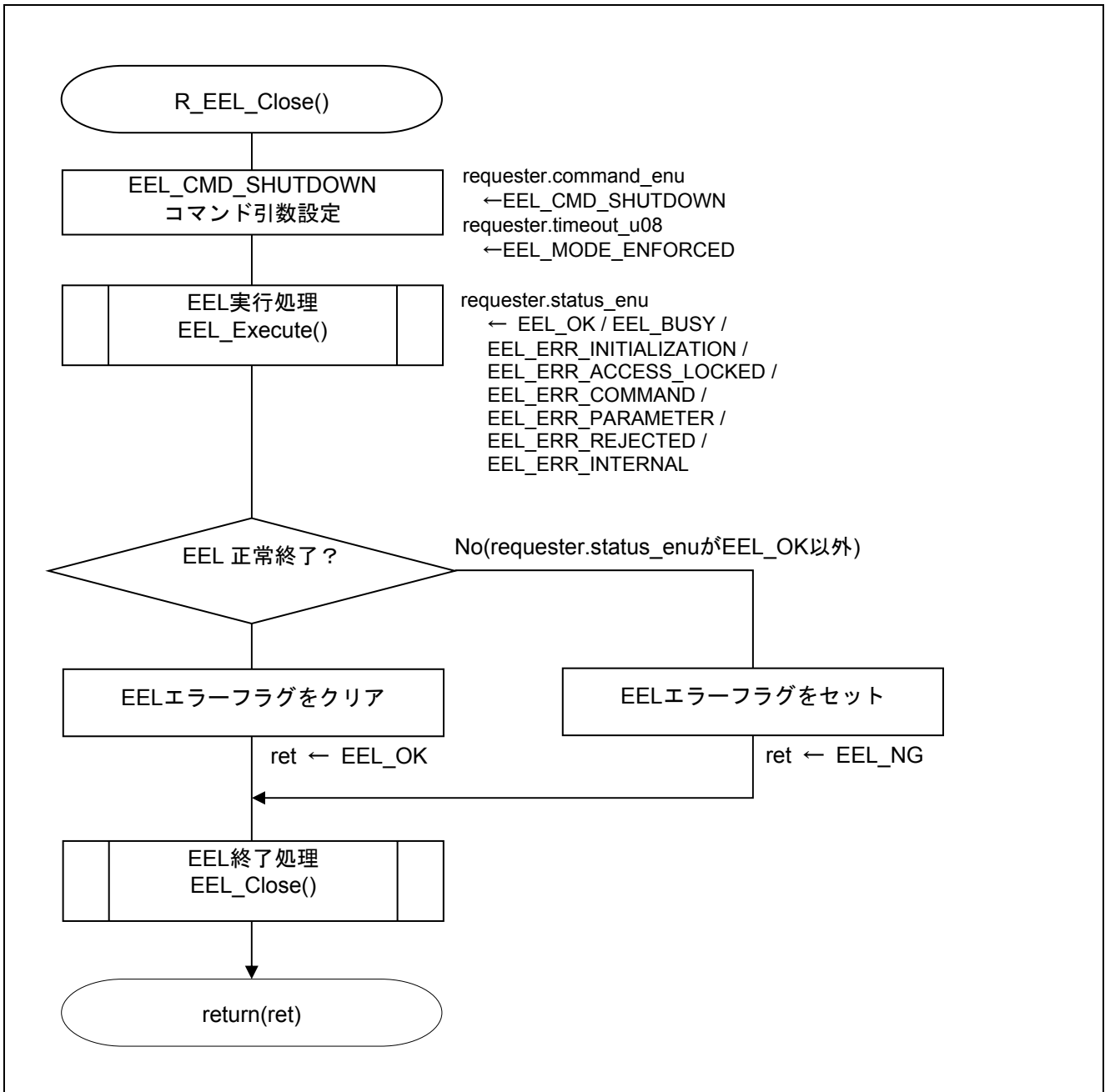


図 5.43 EEL 停止処理

## 6. 本サンプル・プログラム

本サンプル・プログラムは、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)

(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 EEPROM エミュレーション・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	2016.06.01	11	1.4 EEPROM エミュレーション・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、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/>