

---

## RX63N グループ、RX631 グループ

R01AN1713JJ0101

Rev.1.01

### コールドスタート時の RTC 時刻情報の真偽判定例

---

2015.11.06

#### 要旨

本アプリケーションノートでは、リアルタイムクロック(以下、RTC)とバッテリーバックアップ機能を併用した時の使用例を示します。Vcc 端子の電圧が低下してから復帰するまでの期間、VBATT から電源供給ができていたかを判断するために、RTC のクロック設定やレジスタ値を使用して、真偽を行い、継続動作をするか、初期化をするかを判定します。

#### 対象デバイス

- ・ RX63N グループ 177、176 ピン版 ROM 容量 : 768KB~2MB
- ・ RX63N グループ 145、144 ピン版 ROM 容量 : 768KB~2MB
- ・ RX63N グループ 100 ピン版 ROM 容量 : 768KB~2MB
- ・ RX631 グループ 177、176 ピン版 ROM 容量 : 256KB~2MB
- ・ RX631 グループ 145、144 ピン版 ROM 容量 : 256KB~2MB
- ・ RX631 グループ 100 ピン版 ROM 容量 : 256KB~2MB
- ・ RX631 グループ 64 ピン版 ROM 容量 : 256KB~512KB

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
2. 動作確認条件 .....	4
3. 関連アプリケーションノート .....	4
4. ハードウェア説明 .....	5
4.1 ハードウェア構成例 .....	5
4.2 使用端子一覧 .....	5
5. ソフトウェア説明 .....	6
5.1 動作概要 .....	6
5.2 バッテリバックアップ機能 .....	7
5.3 E2 データフラッシュのプログラム/イレーズ .....	8
5.3.1 E2 データフラッシュのイレーズ処理.....	8
5.3.2 E2 データフラッシュのプログラム処理 .....	8
5.3.3 シンプルフラッシュ API の変更点.....	8
5.4 ファイル構成 .....	9
5.5 オプション設定メモリ .....	10
5.6 定数一覧 .....	11
5.7 構造体/共用体一覧 .....	11
5.8 変数一覧 .....	12
5.9 関数一覧 .....	13
5.10 関数仕様 .....	14
5.11 フローチャート.....	18
5.11.1 メイン処理 .....	18
5.11.2 ポート初期設定 .....	19
5.11.3 RTC の初期化を判定.....	20
5.11.4 アラーム時刻情報の有効/無効を設定 .....	22
5.11.5 使用するバックアップした時刻情報を確定 .....	23
5.11.6 RTC 時刻情報の値を確認 .....	24
5.11.7 RTC 時刻情報とバックアップした時刻情報の前後を確認する処理 .....	25
5.11.8 RTC 時刻情報と正常範囲の確認.....	26
5.11.9 RTC 時刻情報の読み出し .....	27
5.11.10 クロック初期設定 .....	28
5.11.11 サブクロックの発振設定 .....	29
5.11.12 RTC の初期化 .....	30
5.11.13 RTC の周期割り込み処理 .....	31
5.11.14 RTC 時刻情報のバックアップ処理.....	32
6. サンプルコード.....	34
7. 参考ドキュメント .....	34

### 1. 仕様

リセットが解除された後、RTC のカウント動作を継続するか、または RTC の初期化をするかをソフトウェアで判定し実行します。

バッテリーバックアップ機能を使用すると、Vcc 端子の電圧が低下しても VBATT からの電源供給により、RTC とサブクロック発振器は動作を継続します。Vcc 端子の電圧の低下が続いて VBATT からの電源供給も停止した場合のみ RTC の初期化が必要です。ただし、Vcc 電圧が低下して RTC への電源供給が Vcc から VBATT に切り替わった後、VBATT 電圧保証範囲を下回った場合は、必ず 1 度 Vcc 端子と VBATT 端子の電圧を 0V まで下げてからパワーオンリセットをしてください。

VBATT の回路は、接続したコンデンサの充電に要する時間は最大 15 分、コンデンサが満充電時に電源供給が可能な期間を最短 10 日の仕様を想定しています。

本アプリケーションノートでは、VBATT から電源供給可能な期間内に Vcc 電圧が復帰してリセット解除されたか、VBATT への充電時間が十分か、およびリセット解除後に RTC レジスタに格納されている値は正常値を保持しているかを確認し判定しています。

Vcc 電圧が保持されている期間は、1 秒周期で RSK の "Debug LCD" に表示している RTC 時刻情報を更新します。また、リセット解除後 1 秒後、VBATT に接続したコンデンサの充電に要する時間 15 分後、RTC 時刻情報の時カウンタ (RHRCNT) を更新した時、E2 データフラッシュに格納している RTC 時刻情報を更新します。

表 1.1 に使用する周辺機能と用途を、図 1.1 に動作例を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
RTCa	時計カウンタ
VBATT	RTC とサブクロック発振器への電源供給
E2 データフラッシュ	RTC 時刻情報のバックアップ

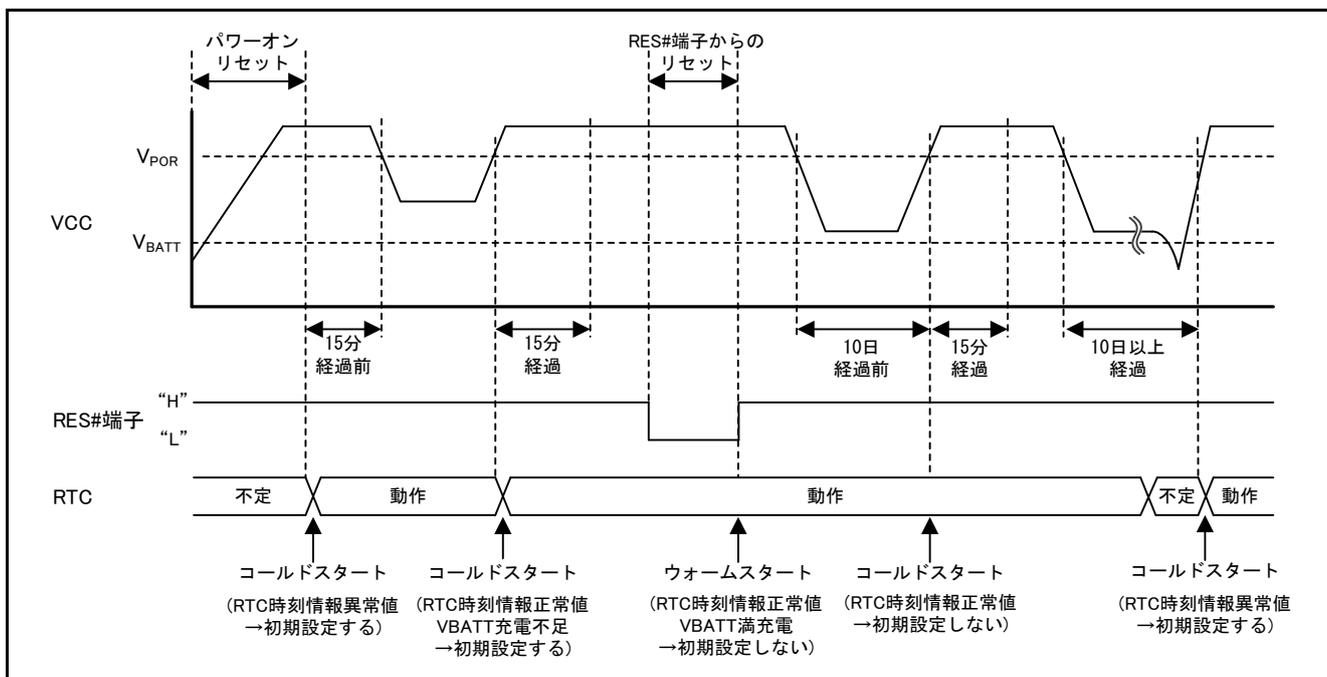


図 1.1 動作例

## 2. 動作確認条件

本アプリケーションノートのサンプルコードは、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	R5F563NBDDFC (RX63N グループ)
動作周波数	<ul style="list-style-type: none"> <li>・ メインクロック: 12MHz</li> <li>・ サブクロック: 32.768kHz</li> <li>・ PLL: 192MHz (メインクロック 1 分周 16 通倍)</li> <li>・ LOCO: 125kHz</li> <li>・ システムクロック (ICLK): 96MHz (PLL 2 分周)</li> <li>・ 周辺モジュールクロック B (PCLKB): 48MHz (PLL 4 分周)</li> </ul>
動作電圧	3.3V、VBATT は V <sub>BATT</sub> 以上
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.01
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.1.02 Release 01 コンパイルオプション -cpu=rx600 -output=obj="\$ (CONFIGDIR)¥\$(FILELEAF).obj" -debug -nologo (統合開発環境のデフォルト設定を使用しています)
iodefine.h のバージョン	Version 1.8
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.01
使用ボード	Renesas Starter Kit+ for RX63N (製品型名: R0K50563NC000BE)

## 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。

RX63N グループ、RX631 グループ 初期設定例 Rev1.10 (R01AN1245JJ)

RX600 & RX200 シリーズ RX 用シンプルフラッシュ API Rev2.40 (R01AN0544JU)

RXFamily Using the Simple Flash API for RX without the r\_bsp Module Rev1.00 (R01AN1890EU)

RX63N High-performance Embedded Workshop 用ルネサススタータキットのサンプルコード Rev1.00 (R01AN1395JG)

RX ファミリ ソフトウェアによるウェイト処理のコーディング例 Rev.1.00 (R01AN1852JJ)

上記アプリケーションノートの初期設定関数、Debug LCD 出力関数、シンプルフラッシュ API 関数、ソフトウェアによるウェイト処理関数を、本アプリケーションノートのサンプルコードで使用しています。Rev は本アプリケーションノート作成時点のものです。

最新版がある場合、最新版に差し替えて使用してください。最新版はルネサスエレクトロニクスホームページで確認および入手してください。

## 4. ハードウェア説明

### 4.1 ハードウェア構成例

図 4.1に接続例を示します。

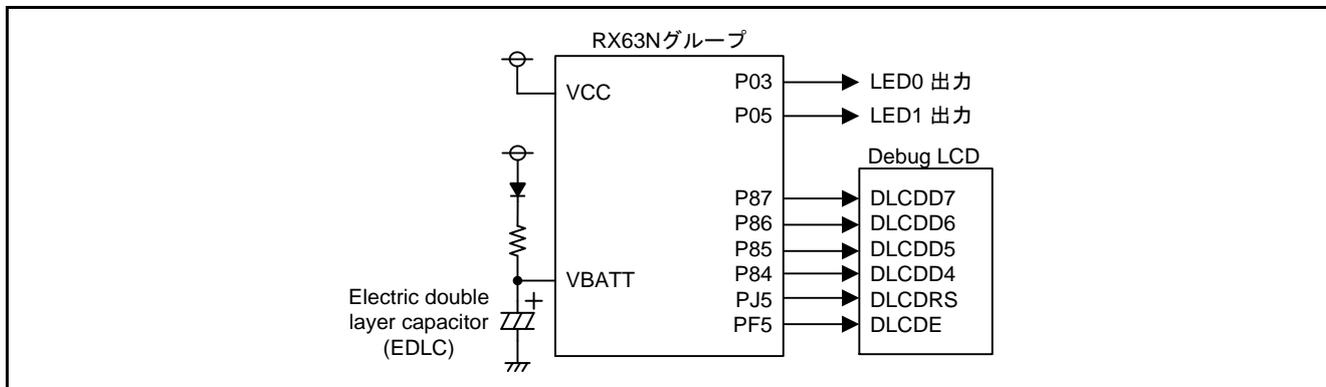


図4.1 接続例

本アプリケーションノートでは、充電時間を最大 15 分、電源保持(放電)時間を最短 10 日としています。

注： 充電時間、電源保持(放電)時間は、VBATT 端子の回路構成によって異なります。充放電時間は、ご使用になるシステムで十分に確認を行ってください。

### 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	内容
P03	出力	LED0 出力(RTC 初期化完了)
P05	出力	LED1 出力(E2 データフラッシュバックアップエラー)
P87	出力	Debug LCD Data bit 7 出力
P86	出力	Debug LCD Data bit 6 出力
P85	出力	Debug LCD Data bit 5 出力
P84	出力	Debug LCD Data bit4 出力
PJ5	出力	Debug LCD Enable 出力
PF5	出力	Debug LCD Register select 出力

## 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、以下のチェック項目で RTC 時刻情報の有効性を判断し、RTC を継続動作するか、RTC の初期化を実行します。図 5.1 に確認手順のフローチャートを示します。

- ・ RTC カウントソース : サブクロック動作
- ・ RTC の制御レジスタ : 初期設定値と同値
- ・ RTC の自動補正設定 : 有効時に初期設定値と同値
- ・ RTC の時刻情報 : 時刻情報として正常値かつ、E2 データフラッシュにバックアップを取った時間から 10 日以内の値
- ・ RTC のアラーム情報 : 時刻情報として正常値かつ、E2 データフラッシュにバックアップを取った時間から 10 日以内の値
- ・ VBATT 回路の充電検出 : コールドスタートかつ、前回の RTC 初期化から 15 分経過

RTC の初期設定値を以下に示します。

- ・ 時間モード : 12 時間モード
- ・ 初期設定時刻 : 2013 年 1 月 1 日(火) 00 時 00 分 00 秒
- ・ アラーム設定 : 無効
- ・ RTCOUT 出力 : 出力禁止
- ・ 自動補正 : 使用しない
- ・ カウントソース : (標準 CL 用)サブクロック
- ・ 時間キャプチャ : 使用しない
- ・ 割り込み : 周期割り込み(PRD)を使用(発生周期: 1 秒)、アラーム割り込み、桁上げ割り込みは禁止。

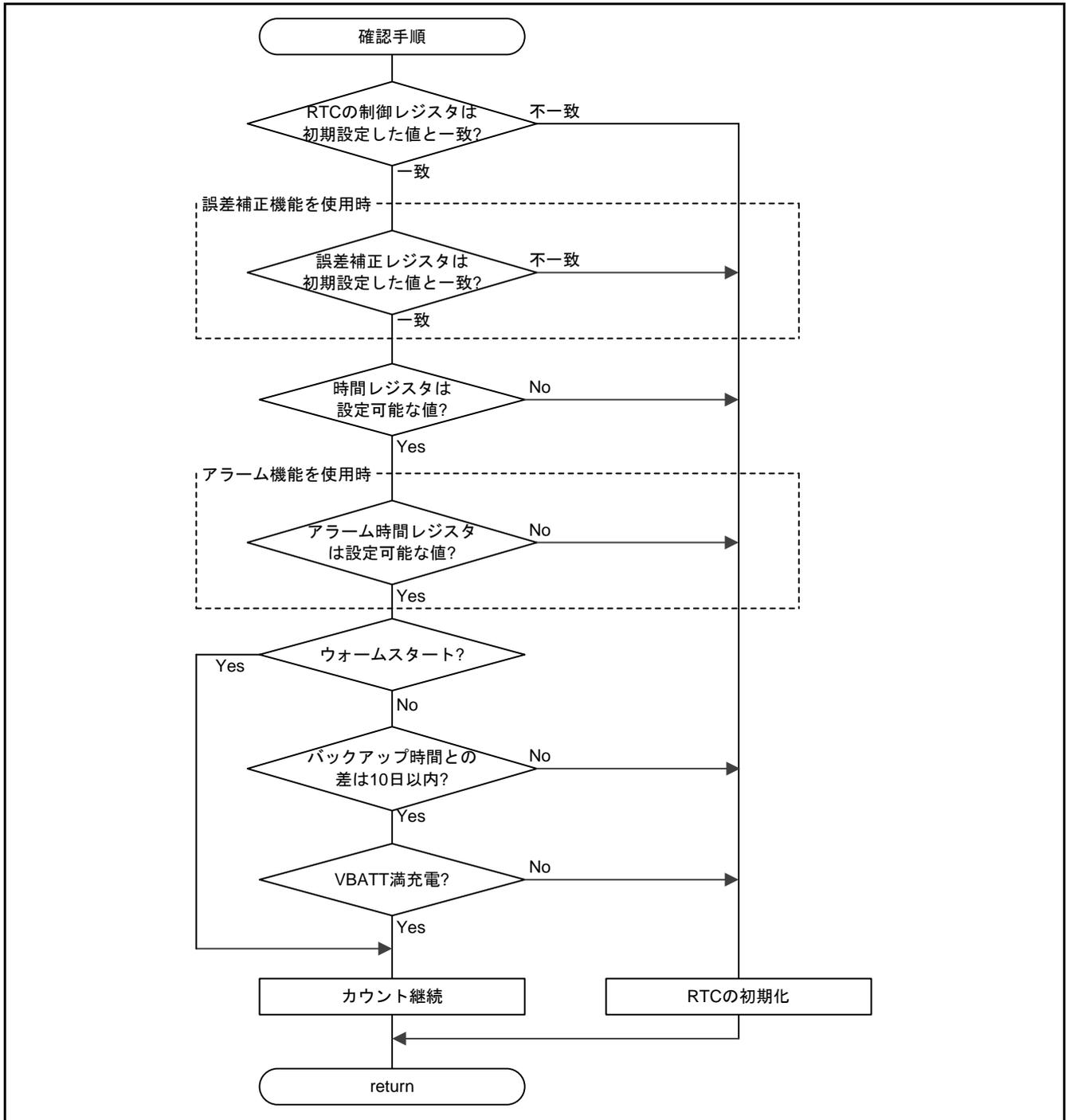


図5.1 確認手順のフローチャート

## 5.2 バッテリバックアップ機能

本アプリケーションノートでは、VBATT 端子に出力電圧  $V_{BATT}$  以上を 10 日以上保持でき、15 分以内で満充電になる回路を接続しています。

なお、充放電時間は接続する回路によって異なります。ご使用されるシステムで十分に確認を行ってください。

### 5.3 E2 データフラッシュのプログラム/イレーズ

本アプリケーションノートでは、E2 データフラッシュの 0010 0000h から 0010 003Fh(64 バイト)を使用します。RTC 時刻情報 10 バイトを格納する 0010 0000h~0010 0009h と 0010 0020h~0010 0029h は交互に上書きされます。64 バイトのうちその他の領域は、データを保持しません。

プログラム/イレーズで使用しているその他の「RX 用シンプルフラッシュ API」については、「RX600 &RX200 シリーズ RX 用シンプルフラッシュ API」を参照してください。

#### 5.3.1 E2 データフラッシュのイレーズ処理

E2 データフラッシュのイレーズ処理は「RX 用シンプルフラッシュ API」で提供される R\_FlashErase Range 関数を使用しています。イレーズエラーは R\_FlashEraseRange 関数からの戻り値か、FlashError 関数でのレジスタ値の確認で判断できます。

#### 5.3.2 E2 データフラッシュのプログラム処理

E2 データフラッシュのプログラムは「RX 用シンプルフラッシュ API」で提供される R\_FlashWrite 関数を使用しています。プログラムエラーは R\_FlashWrite 関数からの戻り値か、FlashError 関数でのレジスタ値の確認で判断できます。

#### 5.3.3 シンプルフラッシュ API の変更点

フラッシュメモリ(ROM)のプログラムから E2 データフラッシュをイレーズ、ライトします。また、イレーズ、ライト実行中も、メインループの処理を実行できるように設定しています。

本アプリケーションノートでは、シンプルフラッシュ API から” r\_flash\_api\_rx\_config.h”を変更しています。

表 5.1に” r\_flash\_api\_rx\_config.h”の変更点を示します。

表 5.1 「r\_flash\_api\_rx\_config.h」の変更点

変更項目	変更内容	変更後のコード
シンプルフラッシュ API の設定変更	ROM の書き込みなし	// #define FLASH_API_RX_CFG_ENABLE_ROM_PROGRAMMING
	書き込むデータは RAM バッファに格納する	#define FLASH_API_RX_CFG_FLASH_TO_FLASH
	E2 データフラッシュの操作をバックグラウンドで実行する	#define FLASH_API_RX_CFG_DATA_FLASH_BGO
	フラッシュ API を RAM に転送しない	// #define FLASH_API_RX_CFG_COPY_CODE_BY_API

注. E2 データフラッシュのプログラム/イレーズ中に、Vcc 電源が遮断されるか、リセットされると、RTC 時刻情報が異常になるため、リセット解除後は RTC も初期状態になります。

## 5.4 ファイル構成

表 5.2 にサンプルコードで使用するファイル、表 5.3 に標準インクルードファイルを、表 5.4、表 5.5 に参照する関連アプリケーションノート関数と設定値を、表 5.6 にシンプルフラッシュ API のモジュールを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表5.2 サンプルコードで使用するファイル

ファイル名	内容
main.c	メイン処理、ポート設定
r_init_clock_an1713.c	クロック初期設定
r_init_clock_an1713.h	r_init_clock_an1713.c のヘッダファイル
rtc_func.c	RTC のレジスタ判定処理と RTC の初期設定、周期割り込み処理
rtc_func.h	rtc_func.c のヘッダファイル
flash_write.c	RTC 時刻情報を E2 データフラッシュにバックアップする処理

表5.3 標準インクルードファイル

参照先	内容
stdbool.h	論理型、および論理値に関するマクロを定義します。
stdint.h	指定した幅の整数型を宣言してマクロを定義します。
machine.h	RX ファミリー用 組み込み関数の形式を定義します。

表5.4 参照する関連アプリケーションノートの関数と設定値

(RX63N グループ、RX631 グループ 初期設定例、RX ファミリー ソフトウェアによるウェイト処理のコーディング例)

ファイル名	関数	設定内容
r_init_stop_module.c	R_INIT_StopModule()	-
r_init_stop_module.h	-	DMAC/DTC、EXDMAC、RAM0、RAM1 はモジュールストップ解除
r_init_non_existent_port.c	R_INIT_NonExistentPort()	-
r_init_non_existent_port.h	-	176 ピン版を指定
r_delay.c	R_DELAY_Us()	ウェイト時間を設定
r_delay.h	-	-

表5.5 参照する関連アプリケーションノートの関数と設定値

(RX63N High-performance Embedded Workshop 用ルネサススタータキットのサンプルコード)

ファイル名	関数	設定内容
lcd.c	Init_LCD() Display_LCD()	-
lcd.h	-	-
rskrx63ndef.h	-	-

表5.6 シンプルフラッシュ API のモジュール

(RX600 & RX200 シリーズ RX 用のシンプルフラッシュ API のアプリケーションノート)

モジュール名	概要
r_flash_api_rx	RX600 & RX200 シリーズ RX 用のシンプルフラッシュ API のプログラム
u_bsp	ユーザ用の Board support package(注)

注. r\_bsp を関連アプリケーションノートの「Using the Simple Flash API for RX without the r\_bsp Module」に従って変更したものです。

## 5.5 オプション設定メモリ

表 5.7にサンプルコードで使用するオプション設定メモリの状態を示します。必要に応じて、お客様のシステムに最適な値を設定してください。

表5.7 サンプルコードで使用するオプション設定メモリ

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh~FFFF FF8Ch	FFFF FFFFh	リセット後、IWDT は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh~FFFF FF88h	FFFF FFFBh	リセット後、電圧監視 0 リセット有効 リセット後、HOCO 発振が無効
MDES	FFFF FF83h~FFFF FF80h	FFFF FFFFh	リトルエンディアン

## 5.6 定数一覧

表 5.8にサンプルコードで使用する定数を示します。

表5.8 サンプルコードで使用する定数

定数名	設定値	内容
RTC_INIT_RES	1	RTC 初期化が必要なりセット
RTC_RUN_RES	0	RTC 初期化が不要なりセット
RTC_BACKUP_INIT	2	RTC 時刻情報のバックアップ要求あり (初期化処理要因、1 秒後のバックアップ要求)
RTC_BACKUP_ACK	1	RTC 時刻情報のバックアップ要求あり (15 分後、1 時間毎のバックアップ要求)
RTC_BACKUP_NACK	0	RTC 時刻情報のバックアップ要求なし
VBATT_FULL_CHARGE	1	VBATT 回路は満充電
VBATT_EMPTY	0	VBATT 回路は充電不足
MASK_DATA_SEC	0x01	秒データチェック処理実行フラグ、フラグ確認データ
MASK_DATA_MIN	0x02	分データチェック処理実行フラグ、フラグ確認データ
MASK_DATA_HR	0x04	時データチェック処理実行フラグ、フラグ確認データ
MASK_DATA_WK	0x08	曜日データチェック処理実行フラグ、フラグ確認データ
MASK_DATA_DAY	0x10	日データチェック処理実行フラグ、フラグ確認データ
MASK_DATA_MON	0x20	月データチェック処理実行フラグ、フラグ確認データ
MASK_DATA_YR	0x40	年データチェック処理実行フラグ、フラグ確認データ
ALL_DATA_CHECK	0x7F	全データチェック処理実行フラグ
ADDRESS_BLOCK_DB0	0x100000	DB0 ブロックの先頭番地
ADDRESS_BLOCK_DB1	0x100020	DB1 ブロックの先頭番地
BUFF_SIZE	10	時刻情報の格納サイズ、書き込みサイズ
WRITE_BUSY	2	書き込みモード(ライト中)
ERASE_BUSY	1	書き込みモード(イレーズ中)
WRITE_READY	0	書き込みモード(準備中)
SELECT_SUB	PATTERN_D	サブクロックの設定パターン選択

## 5.7 構造体/共用体一覧

図 5.2にサンプルコードで使用する構造体/共用体を示します。

```

/* **** Time data **** */
typedef struct
{
    uint8_t    second;        /* Second */
    uint8_t    minute;       /* Minute */
    uint8_t    hour;         /* Hour */
    uint8_t    dayweek;      /* Day of the week */
    uint8_t    day;          /* Day */
    uint8_t    month;        /* Month */
    uint16_t   year;         /* Year */
    uint8_t    charge_VBATT; /* VBATT charge info */
    uint8_t    write_cnt;    /* Backup counter */
} time_bcd_t;
    
```

図5.2 サンプルコードで使用する構造体/共用体

## 5.8 変数一覧

表 5.9にグローバル変数を、表 5.10にstatic 型変数を、表 5.11にconst 型変数を示します。

表5.9 グローバル変数

型	変数名	内容	使用関数
uint8_t	init_RTC	RTC の初期化要求フラグ	main enable_RTC_an1713 CGC_subclk_as_RTC_an1713
uint32_t	time_data	時刻情報の取得バッファ	check_RTC flash_check check_RTC_flash check_RTC_after10days enable_RTC_an1713 Excep_RTC_PRD time_backup
uint8_t	flash_write	RTC 時刻情報のバックアップ要求フラグ	main check_RTC enable_RTC_an1713 Excep_RTC_PRD
int8_t	write_counter	バックアップカウンタ	main check_RTC flash_check check_RTC_flash check_RTC_after10days Excep_RTC_PRD
uint8_t	write_mode	書き込みモード	Excep_RTC_PRD flash_check
uint8_t	lcd_buffer[9]	Debug LCD 用のバッファ	main uint32_ToBCDString
uint8_t	flash_is_ready	フラッシュの処理完了フラグ	time_backup check_RTC Excep_RTC_PRD

表5.10 static 型変数

型	変数名	内容	使用関数
static uint8_t	charge_min_count	VBATT 回路の充電時間(分)カウンタ	check_RTC Excep_RTC_PRD
static uint8_t	check_time_data	時刻情報チェック要否フラグ	check_RTC Set_register_enable flash_check check_RTC_clock
static uint8_t	prog_buff	バックアップする時刻情報	time_backup

表5.11 const 型変数

型	変数名	内容	使用関数
const static uint32_t	backup_address[2] = {ADDRESS_BLOCK_DB0, ADDRESS_BLOCK_DB1};	DB0 ブロックの先頭番地、 DB1 ブロックの先頭番地	check_RTC flash_check check_RTC_flash check_RTC_after10days

## 5.9 関数一覧

表 5.12に関数を示します。

表5.12 関数(1)

関数名	概要	記載ファイル
main	メイン処理	main.c
port_init	ポート出力設定	main.c
check_RTC	RTC の初期化要否を判定	rtc_func.c
Set_register_enable	アラーム時刻情報の有効/無効を設定	rtc_func.c
flash_check	使用するバックアップした時刻情報を確定	rtc_func.c
check_RTC_clock	RTC 時刻情報の値を確認	rtc_func.c
check_RTC_flash	RTC 時刻情報とバックアップした時刻情報の比較	rtc_func.c
check_RTC_after10days	RTC 時刻情報と正常範囲の確認	rtc_func.c
rtc_time_read	RTC 時刻情報の読み出し	rtc_func.c
R_INIT_Clock_an1713	クロック初期設定	r_init_clock_an1713.c
CGC_subclk_as_RTC_an1713	サブクロックの設定 (サブクロックを RTC のカウントソースに使用し、システムクロックには使用しない場合)	r_init_clock_an1713.c
enable_RTC_an1713	RTC の初期化	rtc_func.c
Excep_RTC_PRD	RTC の周期割り込み処理	rtc_func.c
time_backup	RTC 時刻情報のバックアップ	flash_write.c

## 5.10 関数仕様

サンプルコードの関数仕様を示します。

main	
概要	メイン処理
ヘッダ	なし
宣言	void main(void)
説明	初期設定後、RTC 時刻情報のバックアップ要求が発生していれば、バックアップ処理関数を呼び出します。
引数	なし
リターン値	なし
備考	メインループの処理内で E2 データフラッシュ P/E モード中に書き換え禁止のレジスタは書き換えないでください。メインループ処理に別の処理を追加した場合、E2 データフラッシュ P/E モード中は、追加した処理を含むメインループにある処理はすべて P/E モードで動きます。
port_init	
概要	ポート初期設定
ヘッダ	なし
宣言	static void port_init(void)
説明	ポートの初期設定を行います。
引数	なし
リターン値	なし
check_RTC	
概要	RTC の初期化要否を判定する処理
ヘッダ	rtc_func.h
宣言	uint8_t check_RTC(void)
説明	RTC のレジスタ値と enable_RTC_an1713 関数で設定する RTC のレジスタ初期設定値、バックアップした時刻情報、コールド/ウォームスタート、バックアップ時点での VBATT 回路の充電状況を確認し、継続動作をさせるか、初期化処理を実行するかを判断します。
引数	なし
リターン値	RTC_INIT_RES : RTC 時計情報は異常 RTC_RUN_RES : RTC 時計情報は正常
flash_check	
概要	使用するバックアップした時刻情報を確定
ヘッダ	rtc_func.h
宣言	static uint8_t flash_check(void)
説明	DB0, DB1 のバックアップカウンタを確認し、以降で比較に使用するデータを確定させます。もしバックアップカウンタが異常な場合、バックアップした時刻情報を確認して比較に使用するデータを確定させます。
引数	なし
リターン値	RTC_INIT_RES : RTC 時計情報は異常 RTC_RUN_RES : RTC 時計情報は正常

---

Set\_register\_enable

---

概要	アラーム時刻情報の有効/無効を設定する処理	
ヘッダ	rtc_func.h	
宣言	static void Set_register_enable(void)	
説明	アラーム時刻情報の ENB ビットが有効の場合のみレジスタ値を確認するため、RTC のアラームレジスタの ENB ビットを読み出します。	
引数	なし	
リターン値	なし	

---

check\_RTC\_clock

---

概要	RTC 時刻情報の値を確認する処理	
ヘッダ	rtc_func.h	
宣言	static unsigned char check_RTC_clock(void)	
説明	RTC 時刻情報のレジスタ値が正しい値であることを確認する。	
引数	なし	
リターン値	RTC_INIT_RES	: RTC 時計情報は異常
	RTC_RUN_RES	: RTC 時計情報は正常

---

check\_RTC\_flash

---

概要	RTC 時刻情報とバックアップした時刻情報の前後を確認する処理	
ヘッダ	rtc_func.h	
宣言	static unsigned char check_RTC_flash(void)	
説明	RTC 時刻情報とバックアップした時刻情報を比較し、RTC 時刻情報がバックアップした時刻情報よりも後の時間であることを確認する。	
引数	なし	
リターン値	RTC_INIT_RES	: RTC 時計情報は異常
	RTC_RUN_RES	: RTC 時計情報は正常

---

check\_RTC\_after10days

---

概要	RTC 時刻情報と正常範囲の確認処理	
ヘッダ	rtc_func.h	
宣言	static unsigned char check_RTC_after10days(void)	
説明	RTC 時刻情報とバックアップした時刻情報を比較し、RTC 時刻情報が正常な範囲(本アプリケーションノートでは、バックアップした時刻情報から 10 日)以内にあることを確認する。	
引数	なし	
リターン値	RTC_INIT_RES	: RTC 時計情報は異常
	RTC_RUN_RES	: RTC 時計情報は正常

---

<b>rtc_time_read</b>	
概要	RTC 時刻情報の読み出し
ヘッダ	なし
宣言	static void rtc_time_read(void)
説明	RTC 時刻情報を読み出して、RAM の RTC 時刻情報格納領域に格納します。
引数	なし
リターン値	なし

---



---

<b>R_INIT_Clock_an1713</b>	
概要	クロック初期設定
ヘッダ	r_init_clock.h
宣言	void R_INIT_Clock_an1713(void)
説明	クロックの初期設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、システムクロックを PLL とし、サブクロックを RTC のカウントソースに使用する処理を選択しています。 本関数の詳細は、アプリケーションノート「RX63N グループ、RX631 グループ 初期設定例 Rev.1.10」を参照してください。

---



---

<b>CGC_subclk_as_RTC_an1713</b>	
概要	サブクロックの設定
ヘッダ	r_init_clock.h
宣言	void CGC_subclk_as_RTC_an1713 (void)
説明	サブクロックを RTC のカウントソースに使用し、システムクロックには使用しない場合の設定を行います。
引数	なし
リターン値	なし
備考	本関数の詳細は、アプリケーションノート「RX63N グループ、RX631 グループ 初期設定例 Rev.1.10」の設定パターン D を参照してください。

---



---

<b>enable_RTC_an1713</b>	
概要	RTC の初期化
ヘッダ	なし
宣言	static void enable_RTC_an1713 (void)
説明	RTC を使用する場合の初期化(クロック供給の設定と RTC ソフトウェアリセット)を行います。
引数	なし
リターン値	なし

---

Excep_RTC_PRD	
概要	RTC の周期割り込み処理
ヘッダ	rtc_func.h
宣言	static void Excep_RTC_PRD(void)
説明	RTC の周期割り込みで、RTC 時刻情報を読み出して Debug LCD に表示します。 リセット解除から VBATT 回路の充電時間を経過した後に VBATT 回路充電フラグ、バックアップカウンタを更新し、バックアップ要求をセットします。 また、1 時間ごとにバックアップカウンタを更新し、バックアップ要求をセットします。
引数	なし
リターン値	なし
time_backup	
概要	RTC 時刻情報のバックアップ処理
ヘッダ	なし
宣言	void time_backup(uint32_t time_data, uint8_t write_block)
説明	RTC 時刻情報を E2 データフラッシュに格納します。 E2 データフラッシュのブロック 0 あるいはブロック 1 から先頭から 10 バイトに RTC 時刻情報を保存します。保存先は、バックアップカウンタの bit0 をもって判断します。
引数	uint32_t time_data            バックアップする RTC 時刻情報データ uint8_t write_block        バックアップする領域指定データ
リターン値	なし
備考	CPU が E2 データフラッシュの書き込み/消去を実行している間も ROM 領域に配置したプログラムを実行します。

5.11 フローチャート

5.11.1 メイン処理

図 5.3にメイン処理のフローチャートを示します。

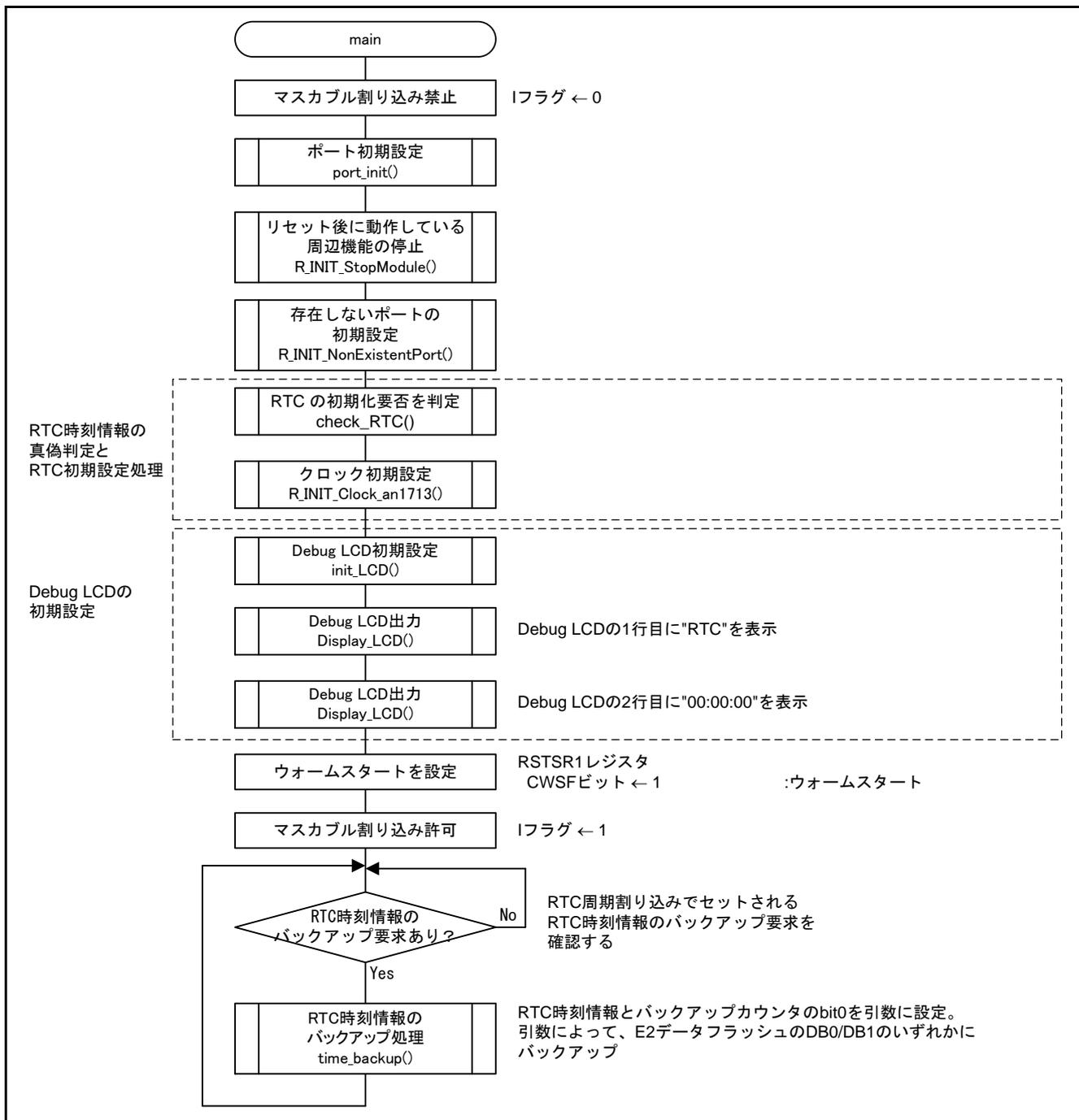


図5.3 メイン処理

5.11.2 ポート初期設定

図 5.4にポート初期設定のフローチャートを示します。

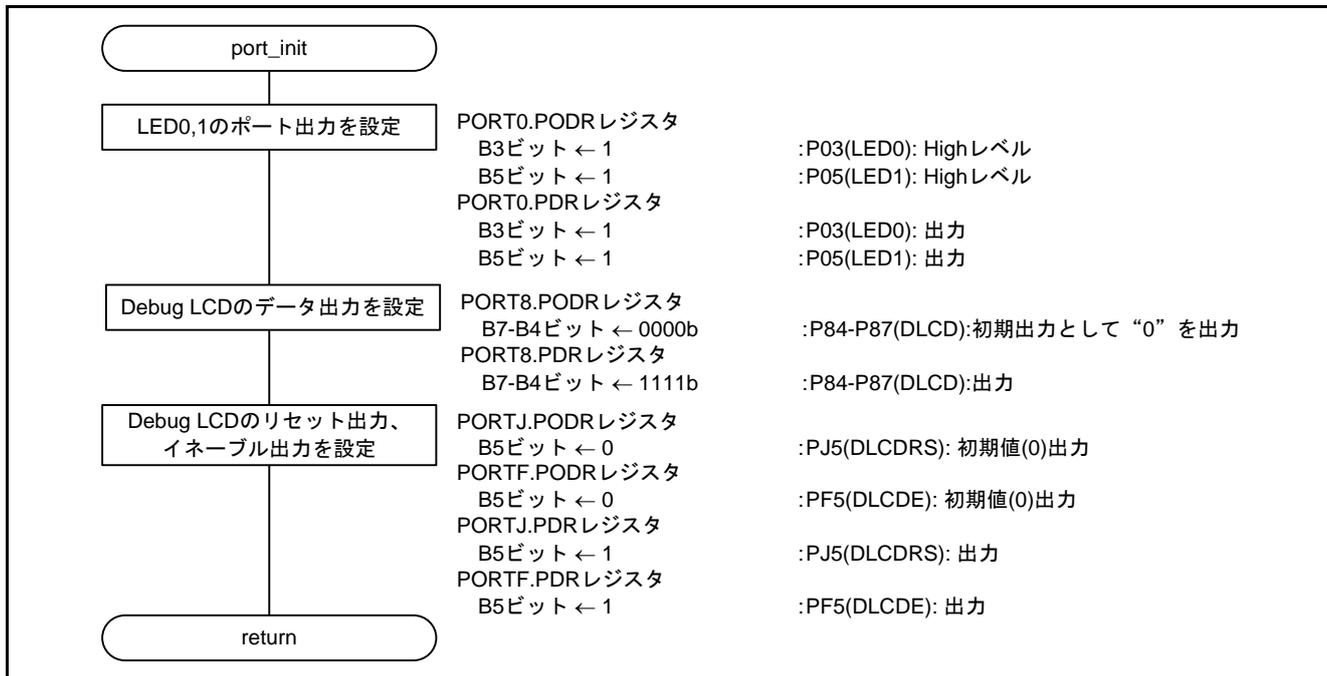


図5.4 ポート初期設定

5.11.3 RTC の初期化を判定

図 5.5、図 5.6にRTC の初期化を判定する処理のフローチャートを示します。

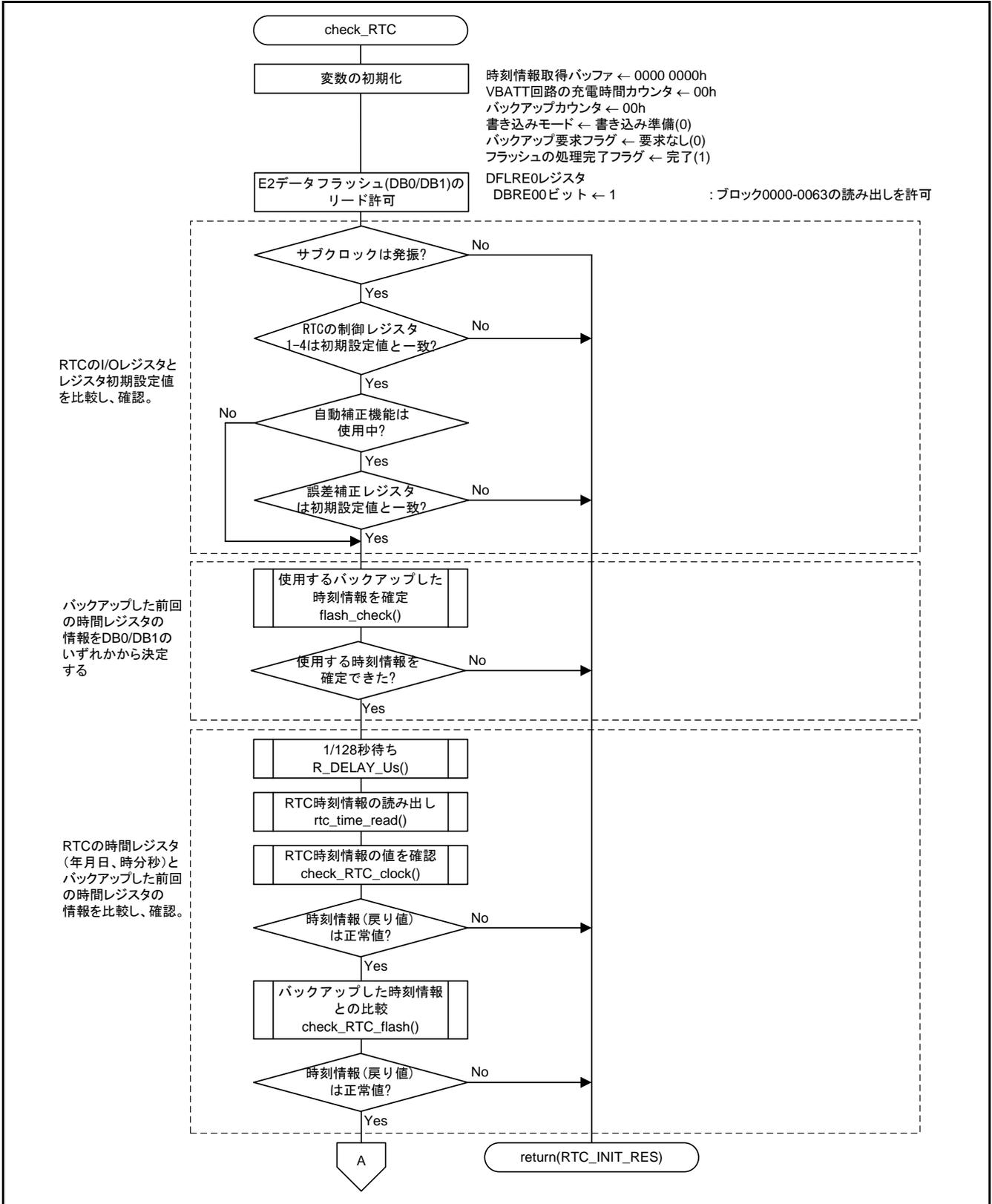


図5.5 RTC の初期化を判定する処理(1)

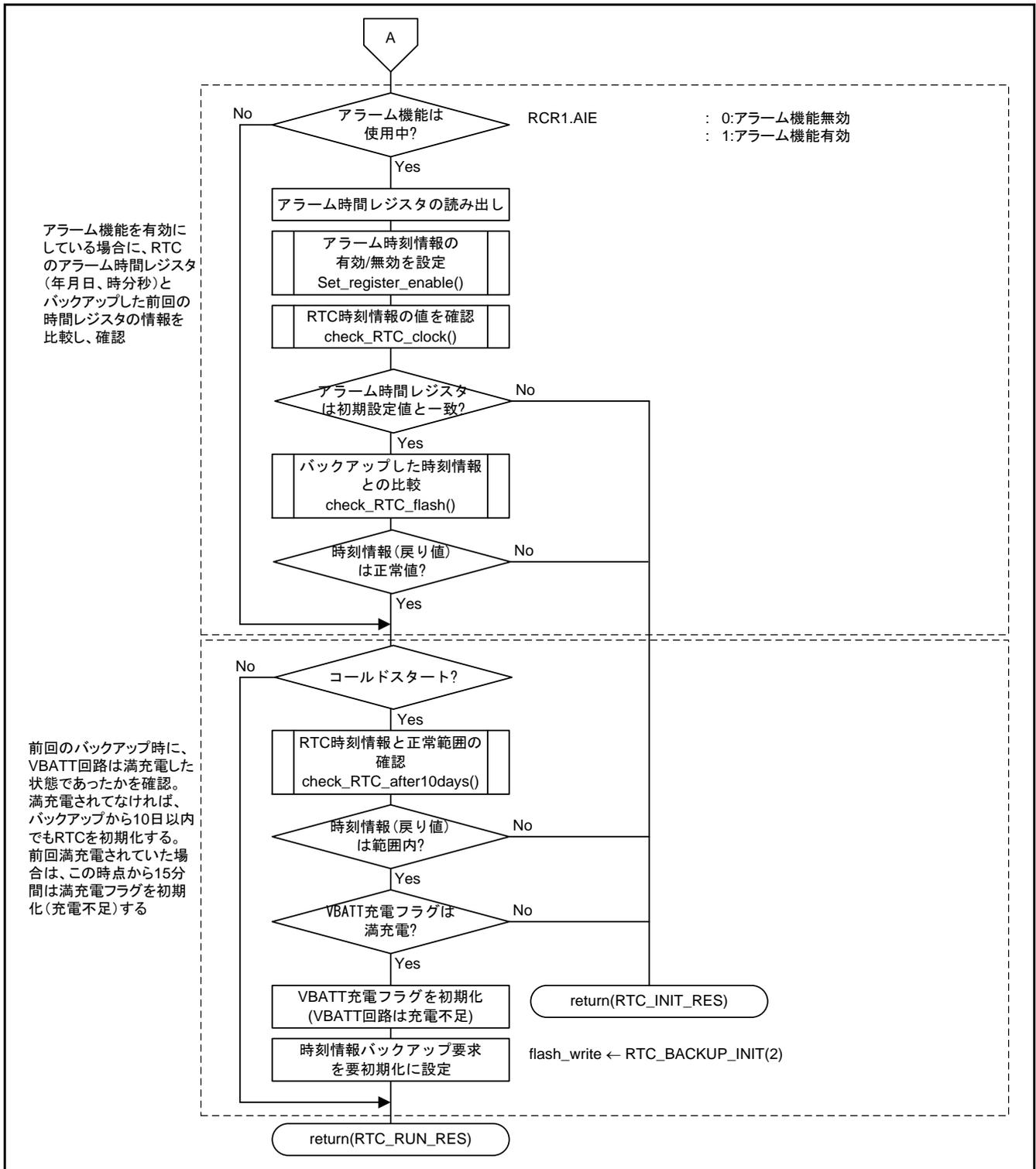


図5.6 RTCの初期化を判定する処理(2)

5.11.4 アラーム時刻情報の有効/無効を設定

図 5.7 にアラーム時刻情報の有効/無効を設定する処理のフローチャートを示します。

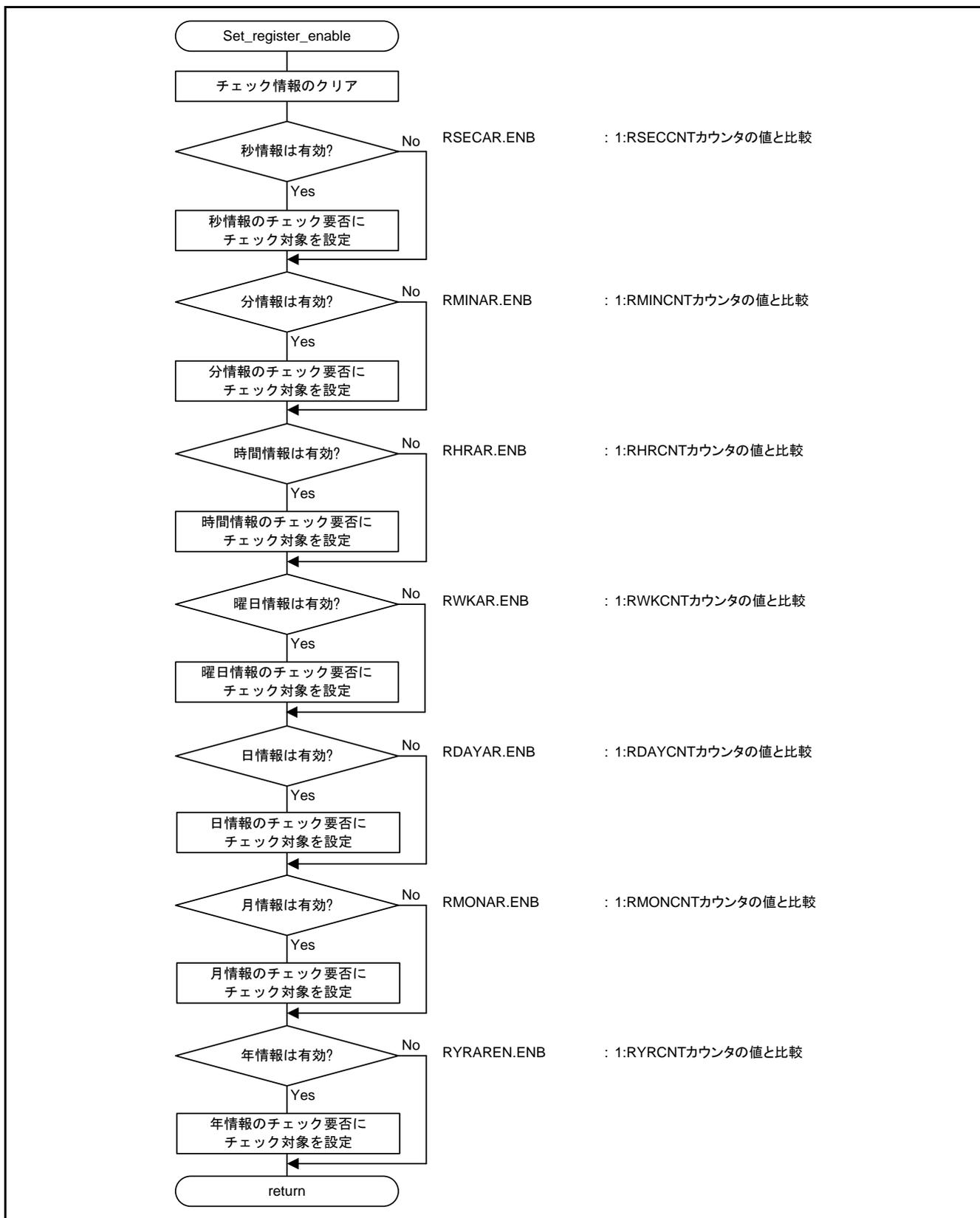


図5.7 アラーム時刻情報の有効/無効を設定する処理

5.11.5 使用するバックアップした時刻情報を確定

図 5.8 にアラーム時刻情報の有効/無効を設定する処理のフローチャートを示します。

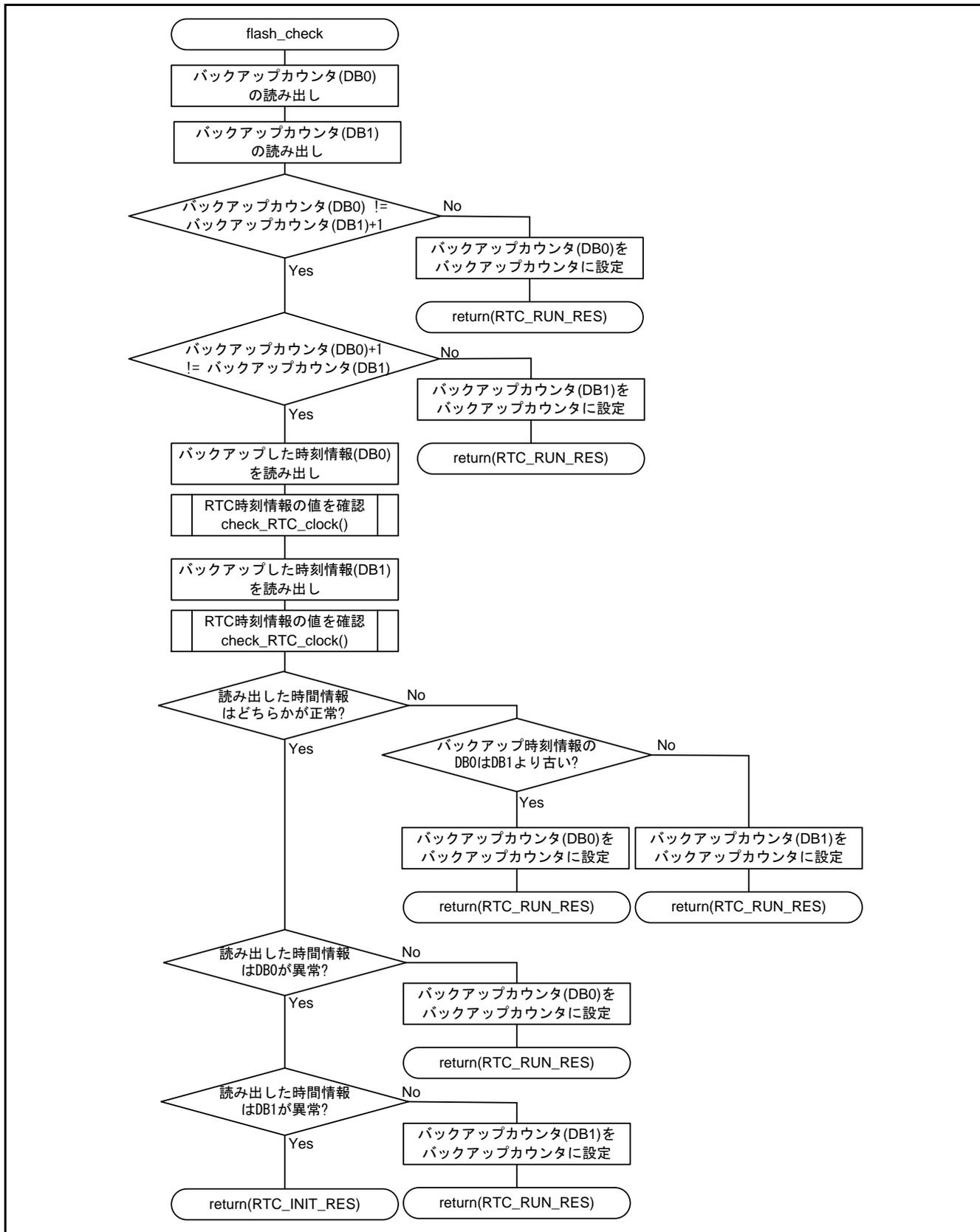


図5.8 使用するバックアップした時刻情報を確定する処理

5.11.6 RTC 時刻情報の値を確認

図 5.9にRTC 時刻情報の値を確認する処理のフローチャートを示します。

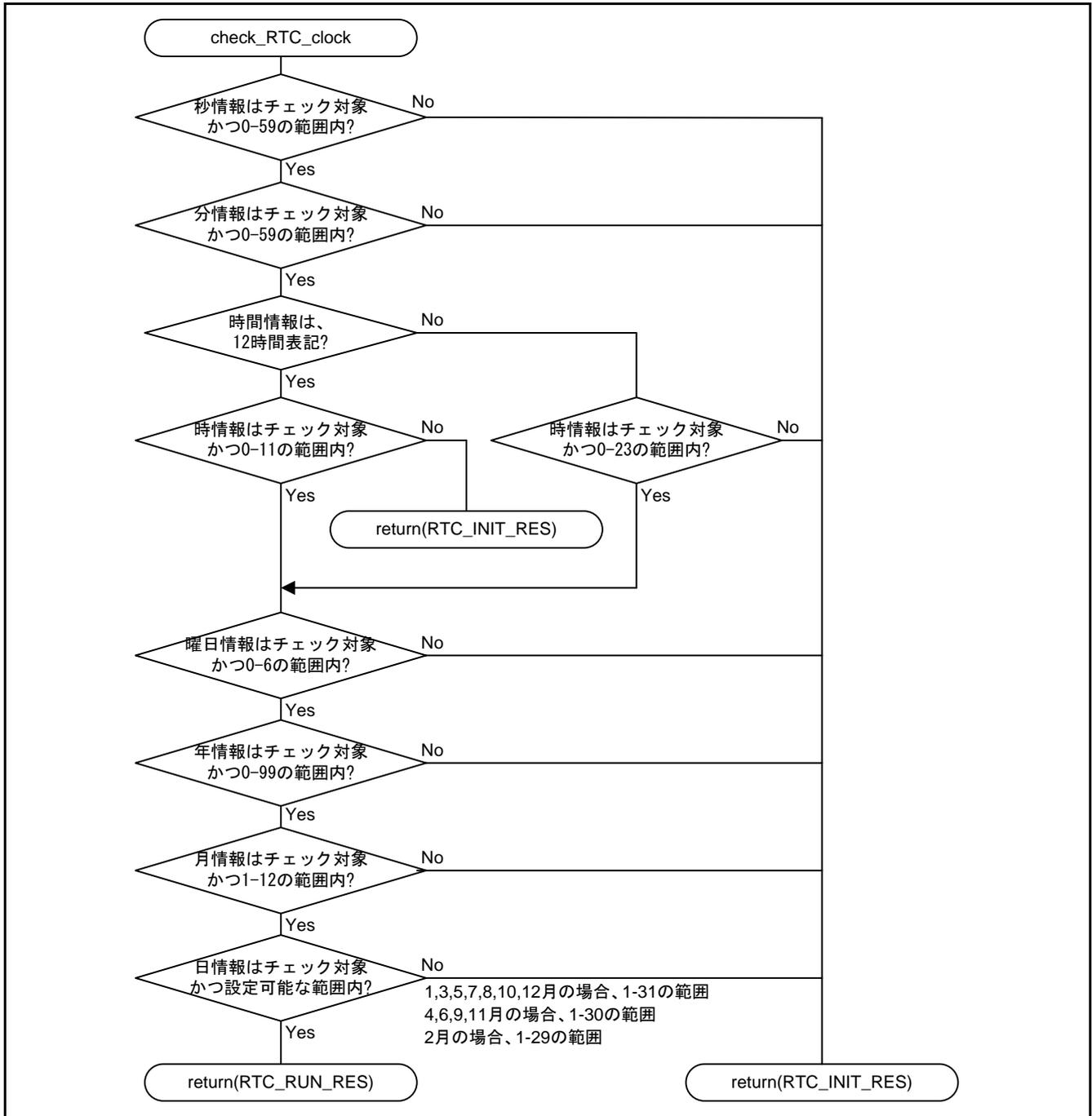


図5.9 RTC 時刻情報の値を確認する処理

5.11.7 RTC 時刻情報とバックアップした時刻情報の前後を確認する処理

図 5.10 にアラーム時刻情報の有効/無効を設定する処理のフローチャートを示します。

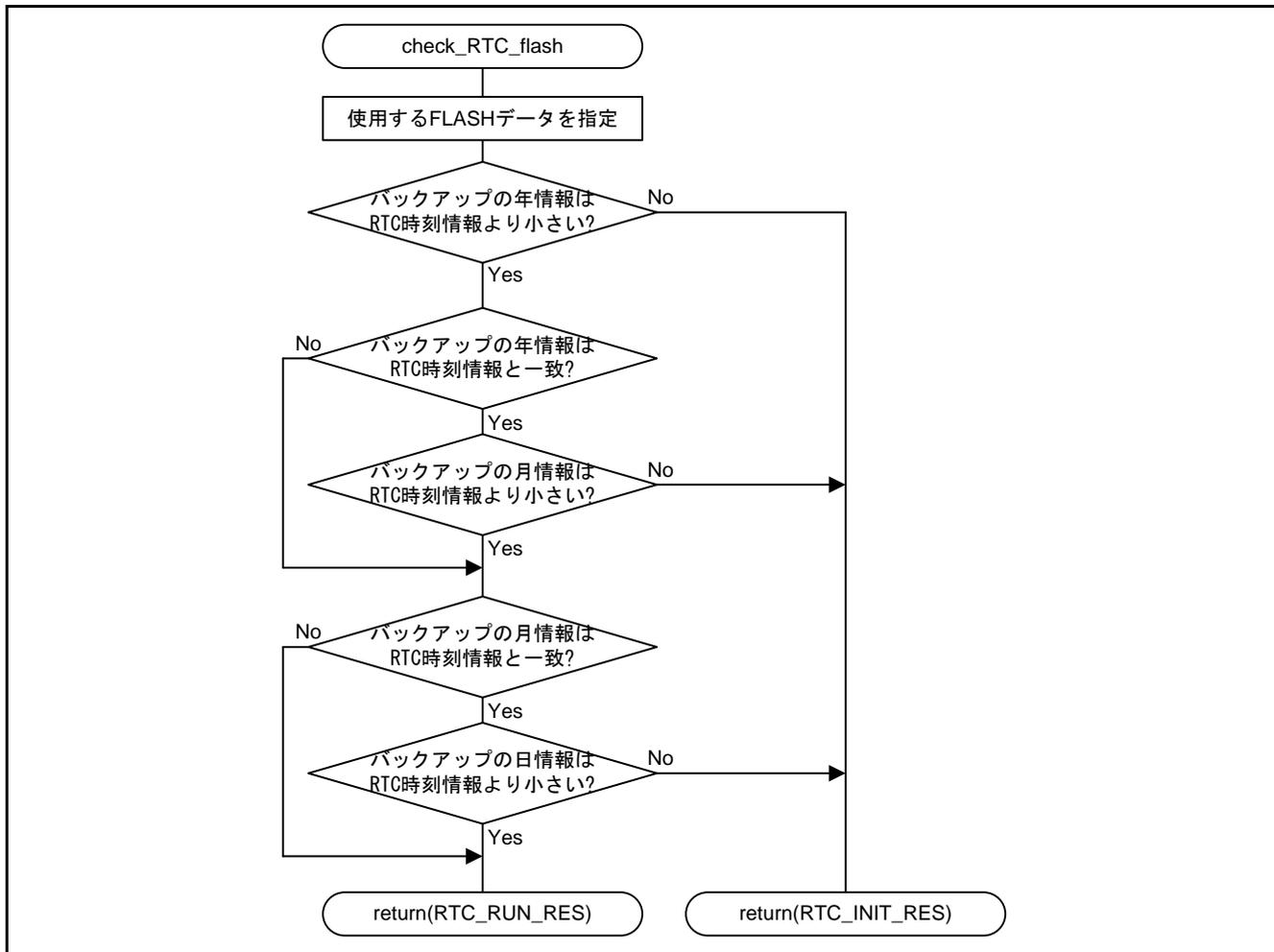


図5.10 RTC 時刻情報とバックアップした時刻情報の前後を確認する処理

5.11.8 RTC 時刻情報と日データの範囲の確認

図 5.11にRTC 時刻情報と正常範囲の確認処理のフローチャートを示します。

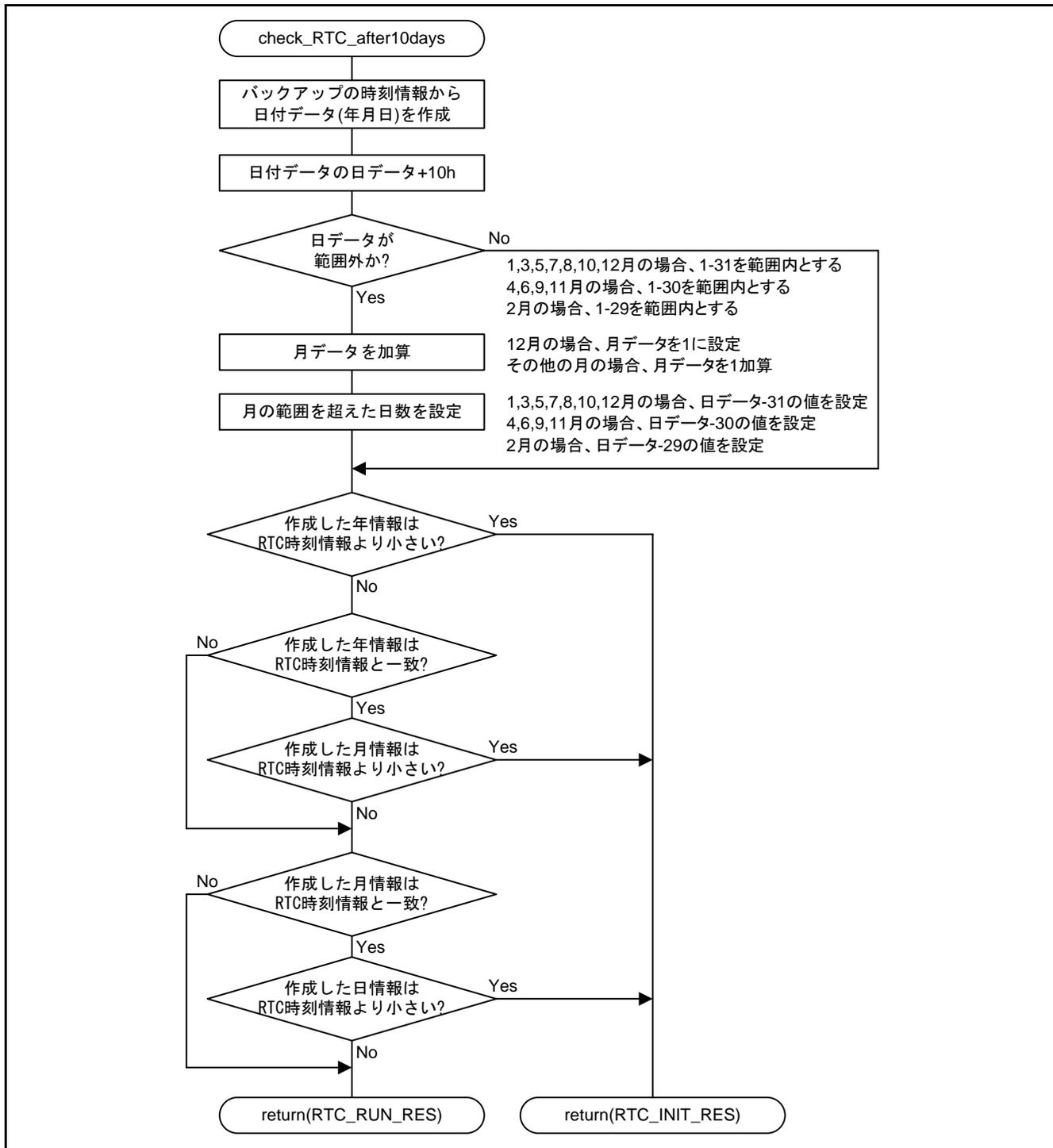


図5.11 RTC 時刻情報と正常範囲の確認処理

5.11.9 RTC 時刻情報の読み出し

図 5.12にRTC 時刻情報の読み出しのフローチャートを示します。

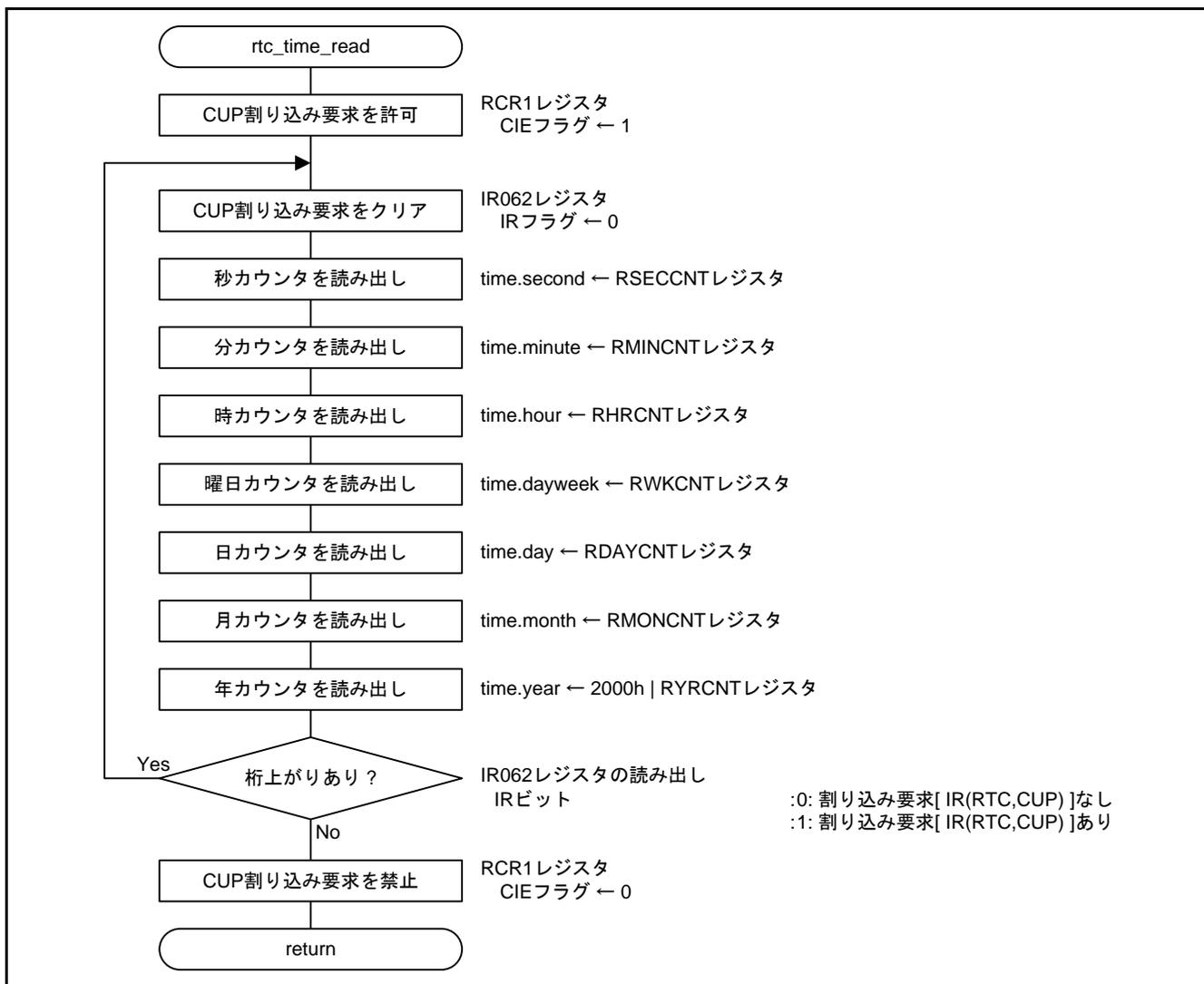


図5.12 RTC 時刻情報の読み出し

5.11.10 クロック初期設定

図 5.13にクロック初期設定のフローチャートを示します。

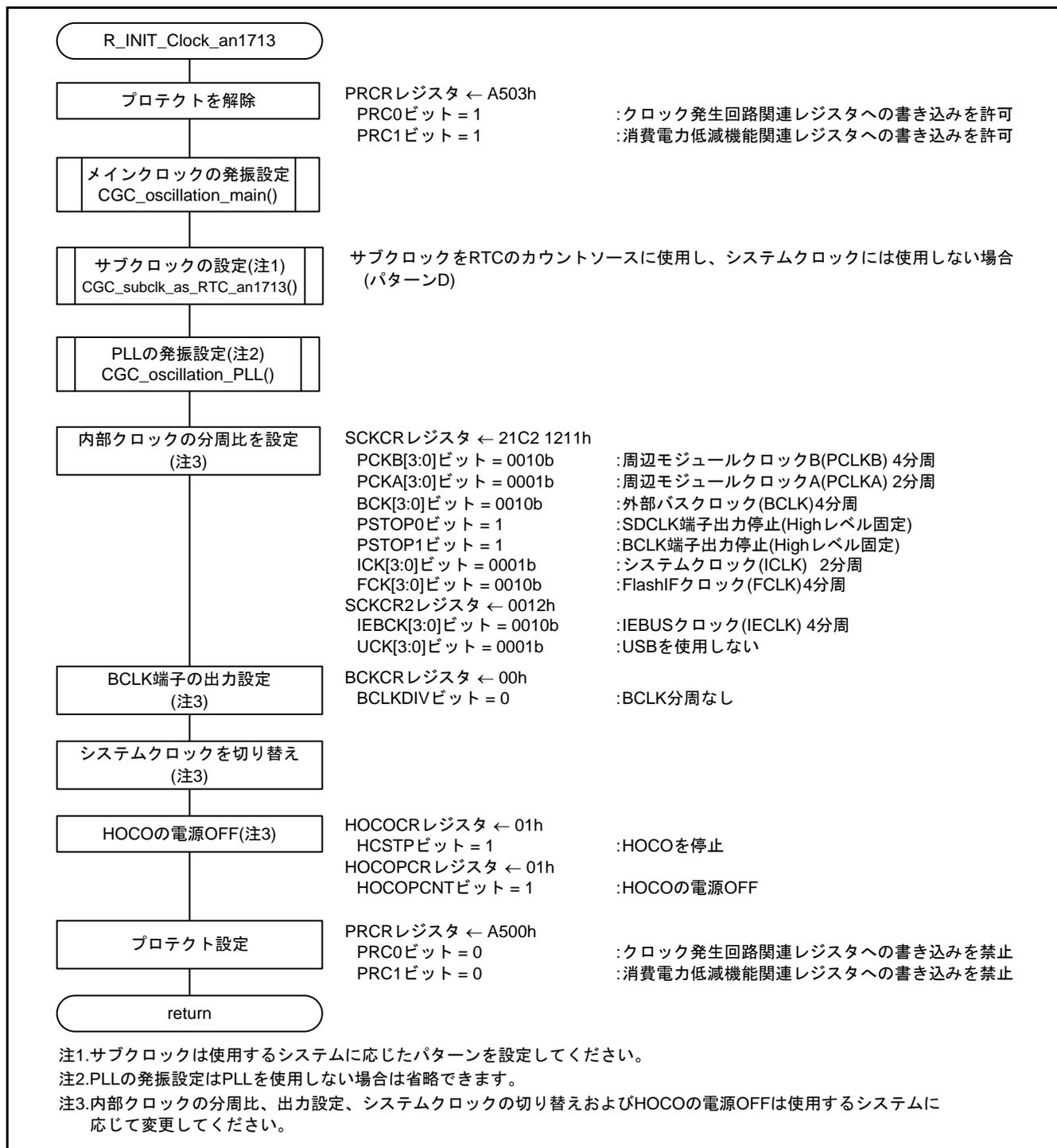


図5.13 クロック初期設定

5.11.11 サブクロックの発振設定

図 5.14にサブクロックの設定のフローチャートを示します。

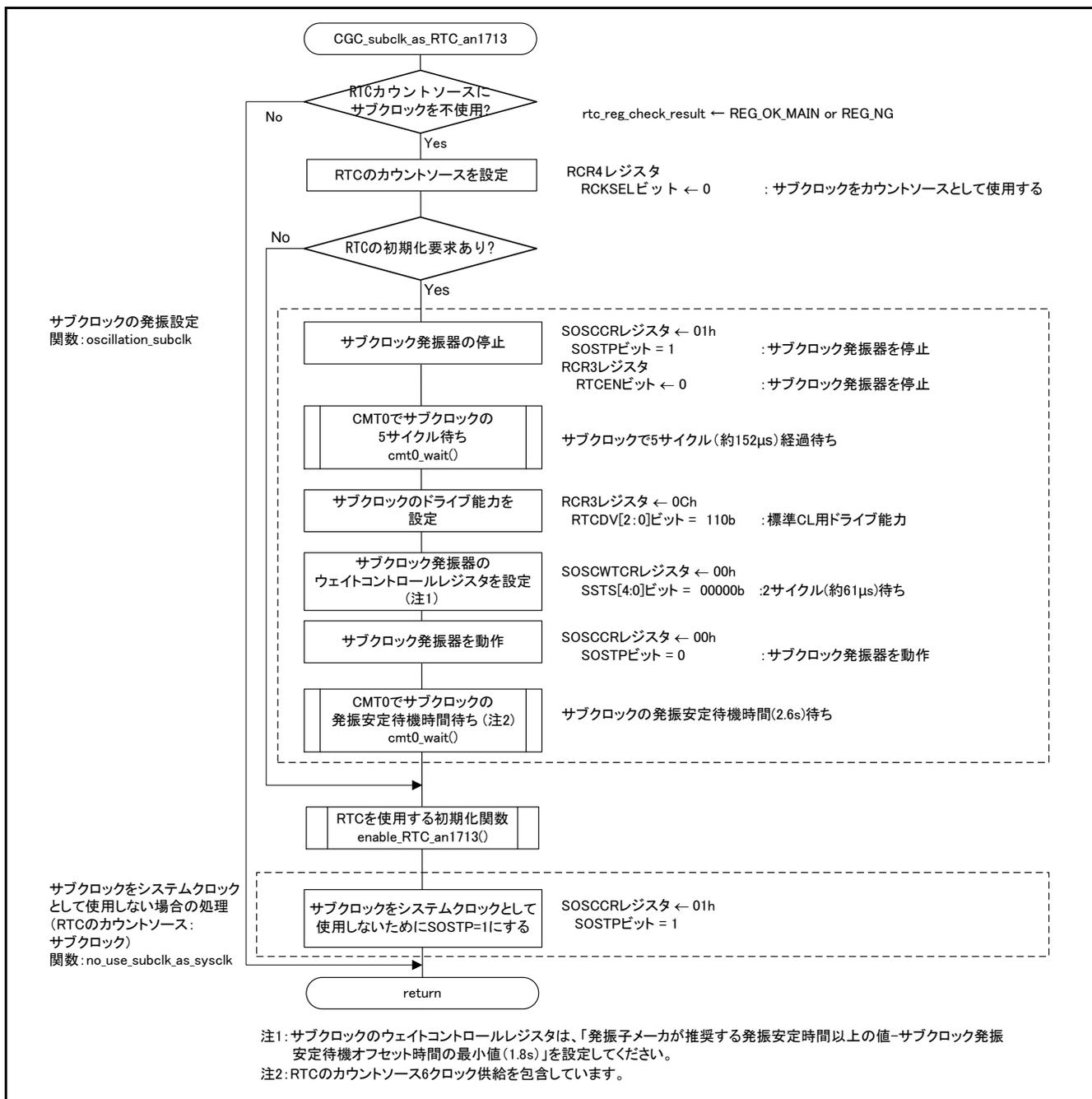


図5.14 サブクロックの設定

5.11.12 RTC の初期化

図 5.15にRTC の初期化処理のフローチャートを示します。

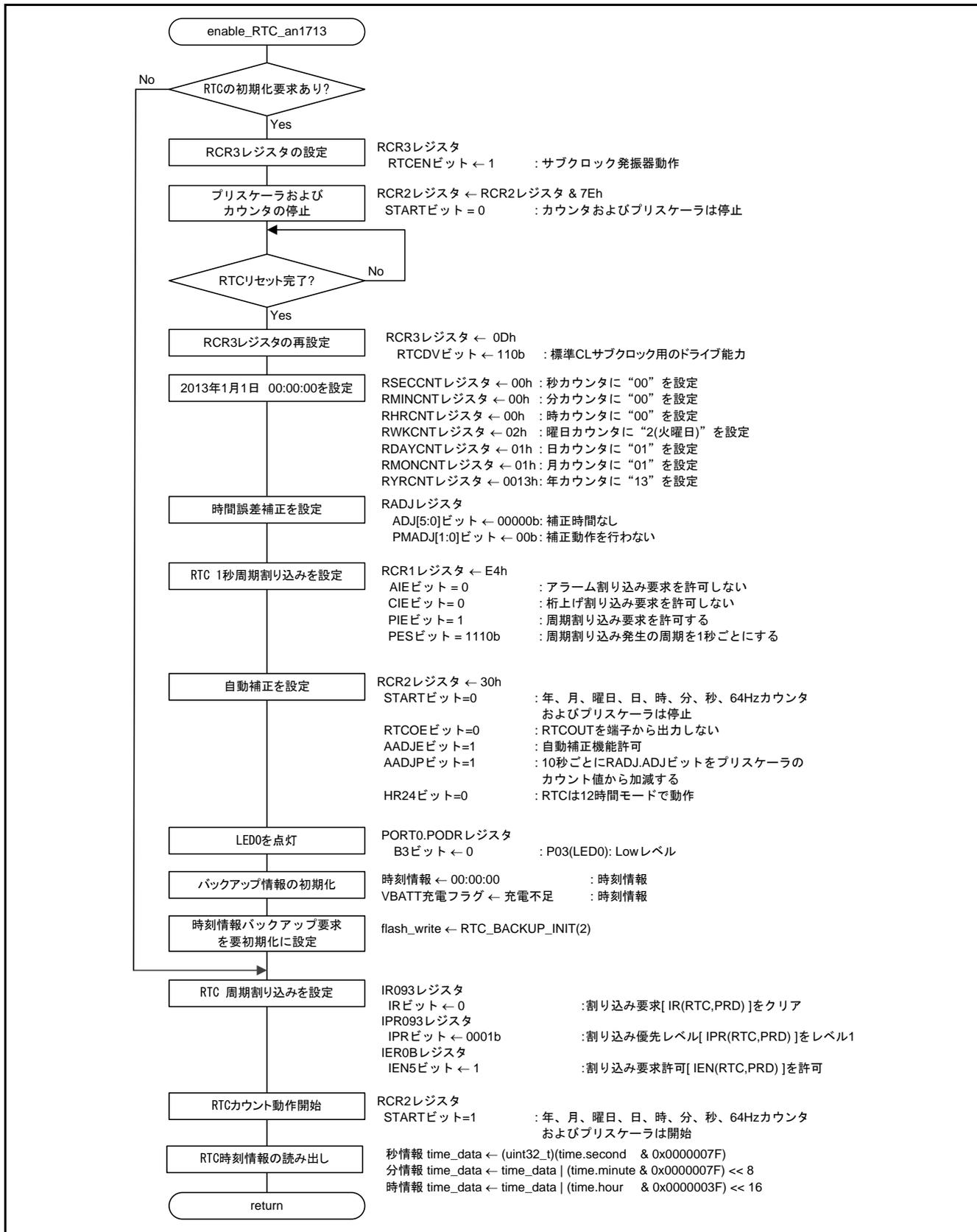


図5.15 RTC の初期化処理

5.11.13 RTC の周期割り込み処理

図 5.16にRTC の周期割り込み処理のフローチャートを示します。

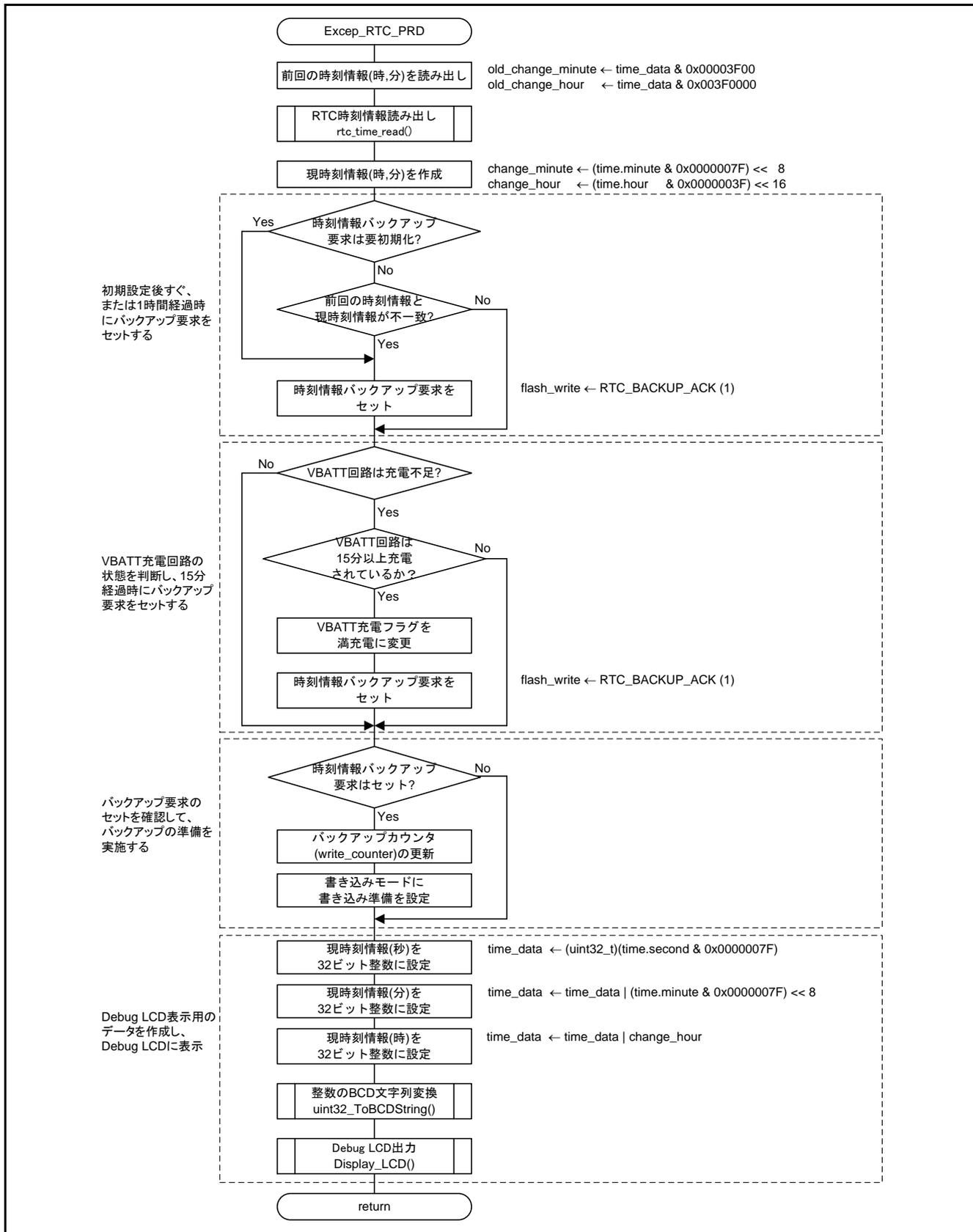


図5.16 RTC の周期割り込み処理

5.11.14 RTC 時刻情報のバックアップ処理

図 5.17、図 5.18にRTC 時刻情報のバックアップ処理のフローチャートを示します。

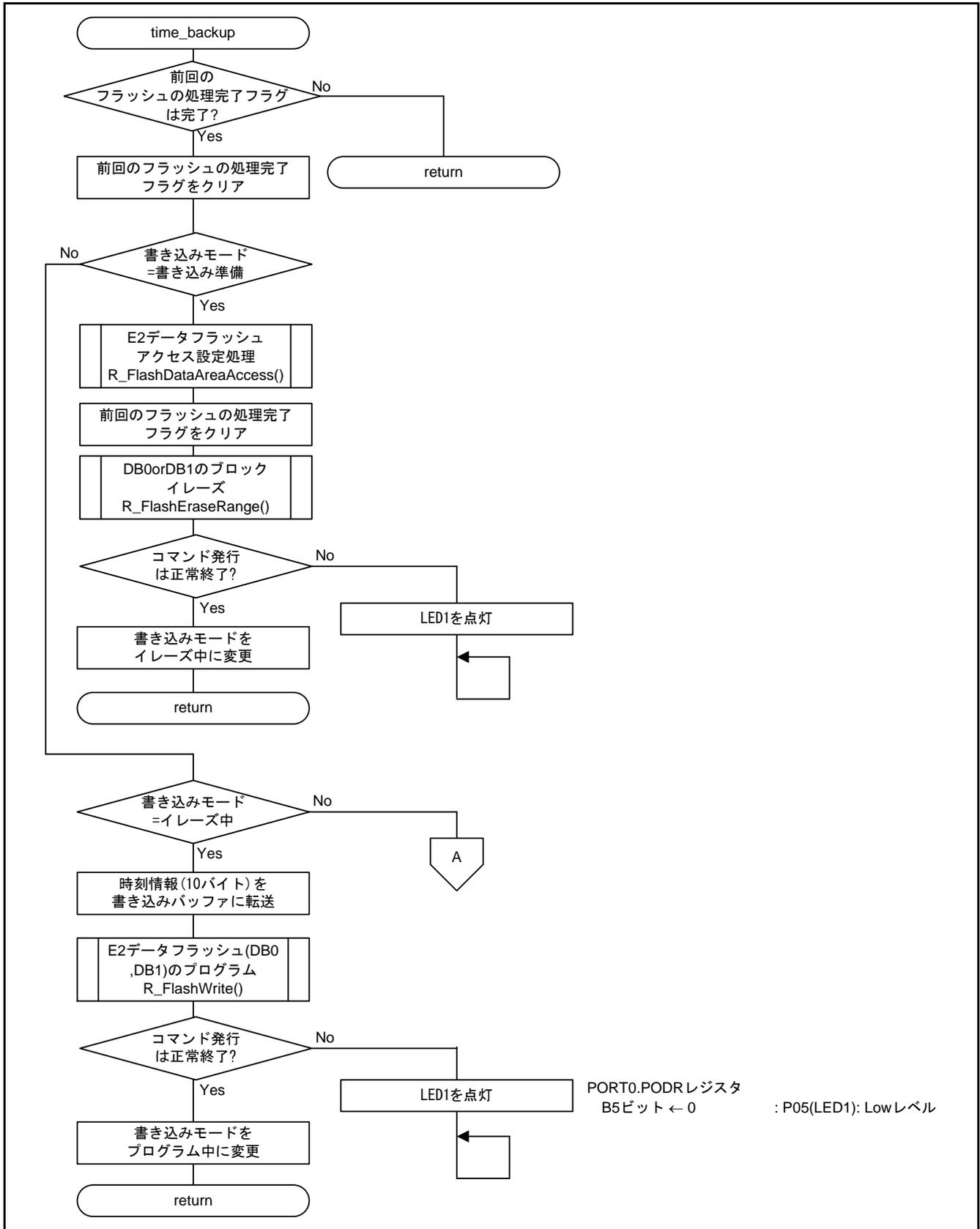


図5.17 RTC 時刻情報のバックアップ処理(1)

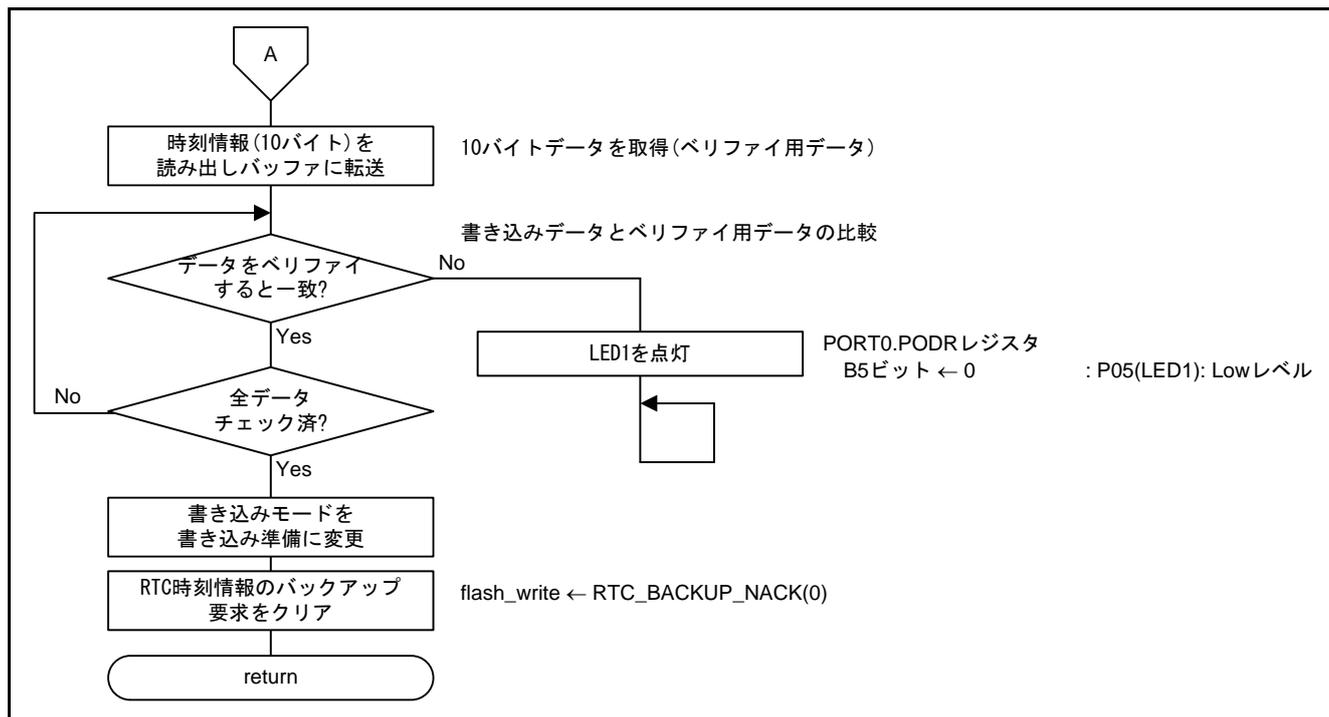


図5.18 RTC 時刻情報のバックアップ処理(2)

## 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

ユーザーズマニュアル：ハードウェア

RX63N グループ、RX631 グループグループ ユーザーズマニュアル ハードウェア編  
Rev.1.80(R01UH0041JJ)  
(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。)

ユーザーズマニュアル：開発環境

RX ファミリー C/C++コンパイラパッケージ V.1.01 ユーザーズマニュアル Rev.1.00(R20UT0570JJ)  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ  
<http://japan.renesas.com>

お問合せ先

<http://japan.renesas.com/contact/>

改訂記録	RX63N グループ、RX631 グループ コールドスタート時の RTC 時刻情報の真偽判定例
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.04.01	—	初版発行
1.01	2015.11.06	3	仕様を修正
		26	図を修正
		プログラム	OFS0 レジスタ、OFS1 レジスタの設定値を修正 リセット後に時間カウンタを読み出すための待ち時間を追加

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）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違えば、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い合わせください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍用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

営業お問い合わせ窓口

<http://www.renesas.com>

営業お問い合わ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。  
総合お問い合わせ窓口：<http://japan.renesas.com/contact/>