RL78/G11

中速オンチップ・オシレータでの UART 通信の実現 CC-RL

要旨

本アプリケーションノートでは、RL78/G11 の中速オンチップ・オシレータを利用した UART 通信方法を説明します。UART 通信で求められる周波数精度を有する高速オンチップ・オシレータを用いて、中速オンチップ・オシレータの発振周期を定期的に測定します。その測定結果に基づいて UART 通信のボーレートを補正することで、中速オンチップ・オシレータを利用した UART 通信を行います。

対象デバイス

RL78/G11

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 補正処理の考え方 8
5.3 オプション・バイトの設定一覧 10
5.4 定数一覧 10
5.5 変数一覧 11
5.6 関数一覧 12
5.7 関数仕様 13
5.8 フローチャート 17
5.8.1 初期設定 17
5.8.2 周辺機能初期設定 18
5.8.3 入出力ポートの設定 19
5.8.4 CPU クロック初期設定 19
5.8.5 タイマ・アレイ・ユニット初期設定 20
5.8.6 8 ビット・インターバル・タイマ初期設定関数 22
5.8.7 シリアル・アレイ・ユニット初期設定 23
5.8.8 UART0 初期設定 24
5.8.9 メイン関数 26
5.8.10 メイン・ユーザー初期設定 28
5.8.11 UART0 受信ステータス初期化 29
5.8.12 UART0 動作開始関数 30
5.8.13 UART0 データ送信 31
5.8.14 UART0 動作停止 32
5.8.15 UART0 受信完了割り込み 33
5.8.16 UART0 受信エラー割り込み 34
5.8.17 UART0 送信完了割り込み 34
5.8.18 8 ビット・インターバル・タイマ割り込み 35
5.8.19 TAU チャネル 3 カウント完了割り込み 37

6. サンプルコード 38

7. 参考ドキュメント 38

1. 仕様

[仕様の説明]

表 1.1 に使用する周辺機能と用途を、図 1.1 に本アプリケーションの概要を示します。

表 1.1　使用する周辺機能と用途

周辺機能	用途
シリアル・アレイ・ユニット	UART 通信
タイマ・アレイ・ユニット	中速オンチップ･オシレータ周波数の分周
8 ビット・インターバル・タイマ	中速オンチップ・オシレータ周波数の測定

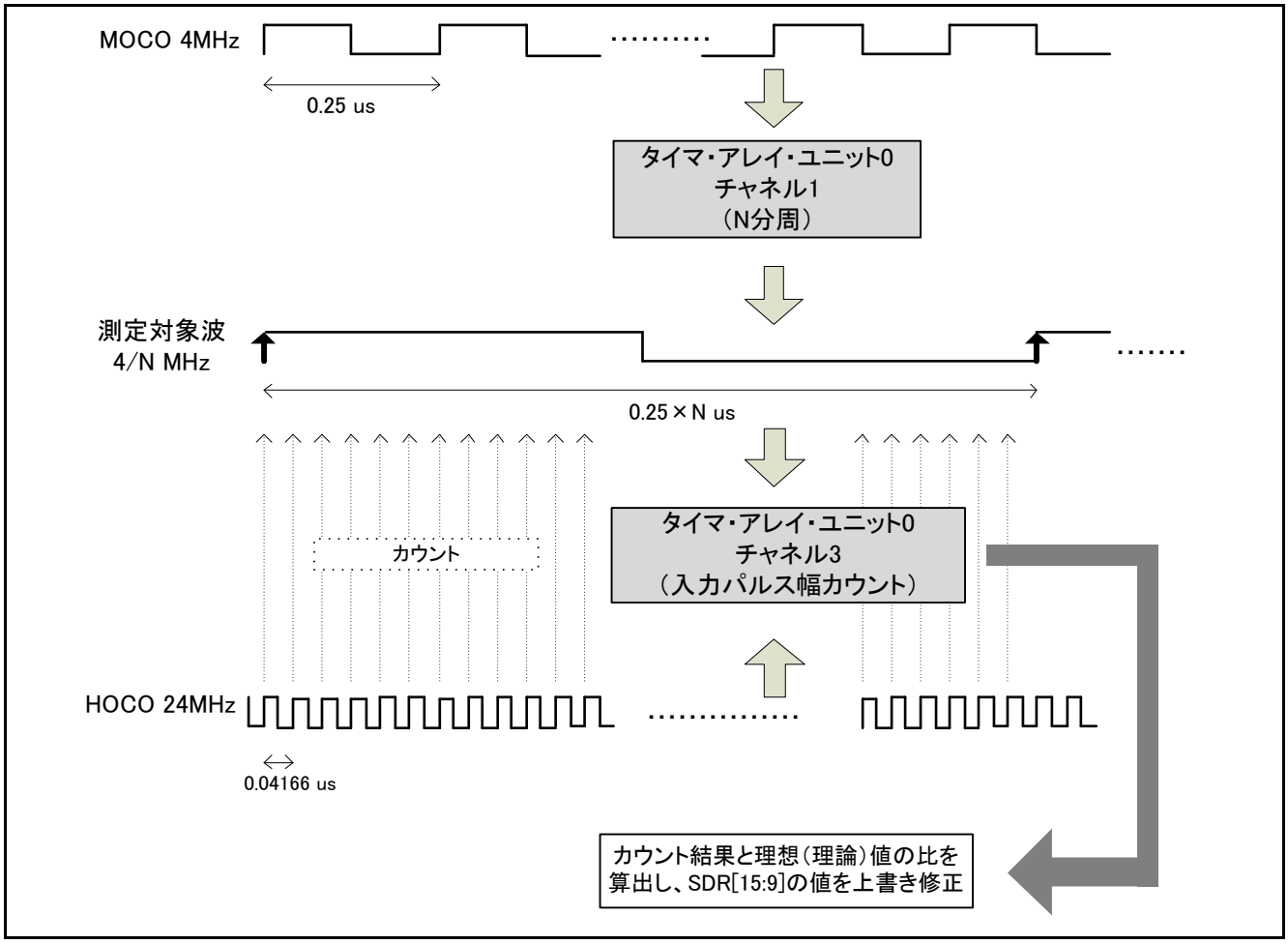

図 1.1　本アプリケーションの概要

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/G11 (R5F1056A)
動作周波数	・中速オンチップ・オシレータ・クロック(f_{IM})：4MHz　（UART 通信時） ・高速オンチップ・オシレータ・クロック(f_{IH})：24MHz　（MOCO 周波数測定時）
動作電圧	3.3V (1.6V～3.6V で動作可能) LVD 動作(V_{LVD})：リセット・モード 2.81V（2.76V～2.87V）
統合開発環境(CS+)	ルネサス エレクトロニクス製 CS+ for CC E8.10.00
C コンパイラ(CS+)	ルネサス エレクトロニクス製 CC-RL V1.12.01
統合開発環境(e^2studio)	ルネサス エレクトロニクス製 e2 studio V2023-01 (23.1.0)
C コンパイラ(e^2studio)	ルネサス エレクトロニクス製 CC-RL V1.12.01

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（R01AN2575JJ）
RL78/G13　シリアル･アレイ･ユニット（UART 通信）（R01AN2517JJ）

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。

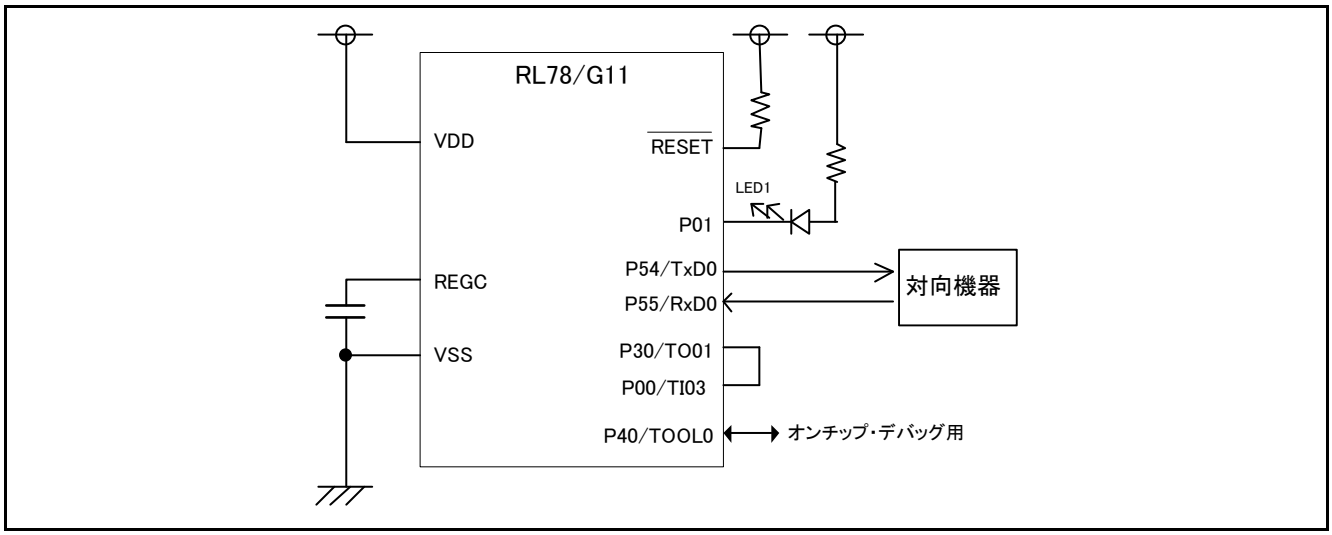

図 4.1　接続例

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

2　EVss で始まる名前の端子がある場合には Vss に、EVDD で始まる名前の端子がある場合には VDD にそれぞれ接続してください。

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

4.2 使用端子一覧

表 4.1 に使用端子と機能を示します。

表 4.1　使用端子と機能

端子名	入出力	内容
P01	出力	LED の点灯制御
P30/TO01	出力	MOCO の分周波形出力
P00/TI03	入力	周波数測定用の波形入力

5. ソフトウェア説明

5.1 動作概要

本サンプルコードでは、対向機器から受信したデータに対応したデータを対向機器に送信します。エラーが発生した場合は、そのエラーに対応したデータを対向機器に送信します。受信データと送信データの対応表を表 5.1 と表 5.2 に示します。

表 5.1　受信データと送信データの対応

受信データ	応答（送信）データ
T（54H）	O（4FH）、 K（4BH）、 ”CR”（0DH）、 ”LF”（0AH）
t（74H）	o（6FH）、 k（6BH）、 ”CR”（0DH）、 ”LF”（0AH）
上記以外	U（55H）、 C（43H）、 ”CR”（0DH）、 ”LF”（0AH）

表 5.2　エラー検出時の送信データの対応

発生したエラー	応答（送信）データ
パリティ・エラー	P（50H）、 E（45H）、 ”CR”（0DH）、 ”LF”（0AH）
フレーミング・エラー	F（46H）、 E（45H）、 ”CR”（0DH）、 ”LF”（0AH）
オーバーラン・エラー	O（4FH）、 E（45H）、 ”CR”（0DH）、 ”LF”（0AH）

(1)　UART の初期設定を行います。

＜UART 設定条件＞

- SAU0 チャネル 0、1 を UART として使用します。
- データ出力は P54/TxD0 端子、データ入力は P55/RxD0 端子を使用します。
- データ長は 8 ビットを使用します。
- データ転送方向設定は LSB ファーストを使用します。
- パリティ設定は偶数パリティを使用します。
- 受信データ・レベル設定は標準を使用します。
- 転送レートは 9600bps を使用します。
- 受信完了割り込み(INTSR0)、送信完了割り込み(INTST0)、エラー割り込み(INTSRE0)を使用します。
- INTSR0、INTST0、INTSRE0 の割り込み優先順位は低優先（レベル 3）を使用します。

(2)　8 ビット・インターバル・タイマの初期設定を行います。

＜8 ビット・インターバル・タイマ設定条件＞

- 8 ビット・インターバル・タイマ割り込み（INTIT00）の割り込み優先順位は低優先（レベル３）を使用します。
- 8 ビット・カウント・モードで使用します。
- 1 分周を f_{IL}/64 に設定します。
- コンペア値を 0xE9 とし、約 1 秒ごとに INTIT00 が発生するよう設定します。

(3)　タイマ・アレイ・ユニットの初期設定を行います。

＜チャネル 1 設定条件＞

- 動作クロックは中速オンチップ・オシレータ（MOCO）4MHz
- 16 ビット・タイマとして動作
- ソフトウェア・トリガ・スタート
- MOCO の有効エッジ：立ち下がり
- インターバル・タイマ・モード／方形波出力として使用
- カウント開始時にタイマ割り込みを発生しない
- 正論理出力

＜チャネル 3 設定条件＞

- 動作クロックは高速オンチップ・オシレータ（HOCO）24MHz
- 単独チャネル動作
- TI03 端子の有効エッジをスタート・トリガ、キャプチャ・トリガとする。
- TI03 端子の有効エッジ：立ち上がりエッジ
- 入力パルス間隔測定モードとして使用
- 正論理出力

(4)　シリアル・チャネル開始レジスタで UART 通信待機状態にした後、HALT 命令を実行します。受信完了割り込み(INTSR0)、エラー割り込み(INTSRE0)の発生により処理を行います。

- INTSR0 発生時は、受信データを取り込み、受信データに対応したデータを送信します。INTSRE0 発生時は、エラー処理を行い、そのエラーに対応したデータを送信します。これらの送信応答処理の間は、8 ビット・インターバル・タイマは動作停止にしておきます。データ送信後は、再び HALT 命令を実行して、受信完了割り込み(INTSR0)、エラー割り込み(INTSRE0)を待ちます。
- 8 ビット・インターバル・タイマ割り込みなど、UART と関係のない要因で HALT から通常動作へ抜け出た場合は、処理を何もせずに HALT 命令に戻ります。

(5)　8 ビット・インターバル・タイマ割り込みにて、ボーレート補正を行います。

- ボーレート補正処理を示す LED を点灯します。
- UART0 の動作を停止します。
- HOCO を発振させ、発振安定時間を待ちます。
- CPU クロック源を MOCO から HOCO に切り替えます。
- TAU0 チャネル 1、チャネル 3 を動作開始し、チャネル 3 の完了割り込みが 2 回発生するまで HALT 状態で待機します。
- 2 回目のチャネル 3 の完了割り込みが発生したら TAU0、チャネル 1 とチャネル 3 の動作を停止します。
- 測定結果から逆算して、ボーレート補正値を SDR レジスタに書き込みます
- UART 通信に備えて CPU クロックを HOCO（24MHz）から MOCO（4MHz）に戻します。
- HOCO の発振を停止します。
- UART0 を動作開始します。
- ボーレート補正処理を示す LED を消灯します。

5.2 補正処理の考え方

補正処理の具体的な方法について詳細を説明します。

(1) 測定対象波の生成

図 1.1 に示したように、中速オンチップ・オシレータ（MOCO）を N 分周して測定対象波を生成します。N の値はできるだけ大きくしたほうが測定の精度が上がりますが、タイマ・アレイ・ユニット 0 チャネル 3 によるカウント結果が最終的に 16 ビットの範囲に収まる必要があります。ここで、MOCO は 4MHz±12%、HOCO は 24MHz±1%ですから、それぞれの Max 値、Typ 値、Min 値の組み合わせにおける測定（カウント）結果を**表 5.3** に示します。

表 5.3 MOCO を N 分周した場合の測定対象波の測定結果

		MOCO (4 MHz±12%)		
		min: 3.52 MHz	Typ: 4.00 MHz	Max: 4.42 MHz
HOCO (24 MHz±1%)	Max: 24.24 MHz	(24.24 / 3.52) N = **6.886N**	(24.24 / 4.00) N = **6.060N**	(24.24 / 4.42) N = **5.484N**
	Typ: 24.00 MHz	(24.00 / 3.52) N = **6.818N**	(24.00 / 4.00) N = **6.000N**	(24.00 / 4.42) N = **5.430N**
	min: 23.76 MHz	(23.76 / 3.52) N = **6.750N**	(23.76 / 4.00) N = **5.940N**	(23.76 / 4.42) N = **5.376N**

表 5.3 に示したように、MOCO が 3.52MHz、HOCO が 24.24MHz のときが、測定結果のカウント値が最も大きくなるケースです。したがって、N は以下の式によって算出できます。

(24.24 / 3.52) N < 2^16 -1

*N < 65535 * 3.52 / 24.24*

N < 9516.6

∴ N (Max) = 9516

ここで、TDR01 レジスタに設定するのは測定対象波の周期の半分です。そのため、設定値は以下のようになります。r_it8bit0_channel0_interrupt 関数をご確認ください。

TDR01 に設定する値 = N/2-1

= 4757 (0x1295)

(2) ボーレート補正

本アプリケーションの UART は MOCO 駆動で f_{CLK} = 4MHz、また f_{MCK} = CK00 = f_{CLK} / 2 = 2MHz の設定です。目標ボーレートは 9600bps ですから、SDR レジスタの上位 7 ビットに入れる値 SDR[15:9]は以下の式で求まります。

$$SDR[15:9] + 1 = f_{MCK} / (2 * 9600)$$
$$= 2\ MHz / (2 * 9600)$$

さらに、MOCO の周波数精度を考慮すると、以下の式になります。

$$SDR[15:9] + 1 = 2\ MHz * (理想カウント値 / CH3\ のキャプチャ値) / (2 * 9600)$$
$$= 2\ MHz * (6.000*N / CH3\ のキャプチャ値) / (2 * 9600)$$
$$= 2\ MHz * (6 * 9516 / CH3\ のキャプチャ値) / (2 * 9600)$$
$$= 5947500 / (CH3\ のキャプチャ値)$$

演算を簡単にするため、右辺の分母と分子をそれぞれ 256（= 2^8）で割ると、最終的に以下の式になります。本サンプルプログラムでは、r_it8bit0_channel0_interrupt 関数にて記述されております。左辺値 SDR[15:9] + 1 をローカル変数 k として記述していますので、ご確認下さい。

$$= 23232 / (CH3\ のキャプチャ値の上位\ 8\ ビット)$$

5.3 オプション・バイトの設定一覧

表 5.4 に、オプション・バイト設定を示します。

表 5.4 オプション・バイト設定

アドレス	設定値	内容
000C0H/010C0H	1110 1111B	ウォッチドッグ・タイマ動作禁止 （リセット解除後、カウント停止）
000C1H/010C1H	0111 1111B	LVD リセット・モード 2.81V(2.76V～2.87V)
000C2H/010C2H	1110 0000B	HS モード、HOCO：24MHz
000C3H/010C3H	1000 0100B	オンチップ・デバッグ許可

5.4 定数一覧

表 5.5 に**サンプルコードで使用する定数**を示します。

表 5.5 サンプルコードで使用する定数

定数名	設定値	内容
MessageOK[4]	"OK¥r¥n"	“T”を受信時の返信メッセージ
Messageok[4]	"ok¥r¥n"	“t”を受信時の返信メッセージ
MessageUC[4]	"UC¥r¥n"	“T” または ”t” 以外を受信した時の返信メッセージ
MessageFE[4]	"FE¥r¥n"	フレーミング・エラー時の返信メッセージ
MessagePE[4]	"PE¥r¥n"	パリティ・エラー時の返信メッセージ
MessageOE[4]	"OE¥r¥n"	オーバーラン・エラー時の返信メッセージ

5.5 変数一覧

表 5.6 にグローバル変数を示します。

表 5.6 グローバル変数

型	変数名	内容	使用関数
uint8_t	g_uart0_rx_buf	受信データ・バッファ	main
uint8_t*	gp_uart0_tx_address	送信データ・ポインタ	R_UART0_Send、 R_uart0_interrupt_send
uint16_t	g_uart0_tx_count	送信データ数カウンタ	R_UART0_Send、 r_uart0_interrupt_send
uint8_t*	gp_uart0_rx_address	受信データ・ポインタ	R_UART0_Receive、 r_uart0_interrupt_receive、 r_uart0_interrupt_error
uint16_t	g_uart0_rx_count	受信データ数カウンタ	R_UART0_Receive、 r_uart0_interrupt_receive
uint16_t	g_uart0_rx_length	受信データ数	R_UART0_Receive、 r_uart0_interrupt_receive
uint8_t	g_valid_measure	TAU0 チャネル 3 の割り込み発生回数	r_it8bit0_channel0_interrupt r_tau0_channel3_interrupt
MDSTATUS	g_uart0_tx_end	UART 送信処理終了フラグ	main
uint8_t	g_uart0_after_adjustment	8 ビット・インターバル・タイマ割り込み関数の経過直後フラグ	main r_it8bit0_channel0_interrupt
uint32_t	g_tau0_ch3_width	方形波 1 周期の測定結果	r_tau0_channel3_interrupt r_it8bit0_channel0_interrupt
uint8_t	g_uart0_rx_err	UART 通信ステータス	main r_uart0_callback_receiveend r_uart0_callback_error

5.6 関数一覧

表 5.7 に関数一覧を示します。

表 5.7 関数一覧

関数名	概要
main	メイン関数
R_MAIN_UserInit	メイン・ユーザー初期化関数
R_UART0_Receive	UART0 受信ステータス初期化関数
R_UART0_Start	UART0 動作開始関数
R_UART0_Send	UART0 データ送信関数
R_UART0_Stop	UART0 動作停止関数
r_uart0_interrupt_receive	UART0 受信完了割り込み関数
r_uart0_interrupt_error	UART0 受信エラー割り込み関数
r_uart0_interrupt_send	UART0 送信完了割り込み関数
r_it8bit0_channel0_interrupt	8 ビット・インターバル･タイマ割り込み関数
r_tau0_channel3_interrupt	TAU チャネル 3 カウント完了割り込み関数
R_TAU0_Channel1_Start	TAU チャネル 1 動作開始関数
R_TAU0_Channel3_Start	TAU チャネル 3 動作開始関数
R_TAU0_Channel1_Stop	TAU チャネル 1 動作開始停止
R_TAU0_Channel3_Stop	TAU チャネル 3 動作開始停止
R_IT8Bit0_Channel0_Start	8 ビット・インターバル･タイマ動作開始関数
R_IT8Bit0_Channel0_Stop	8 ビット・インターバル･タイマ動作停止関数

5.7　関数仕様

サンプルコードの関数仕様を示します。

[関数名]　main	
概　要	メイン関数
ヘッダ	r_cg_macrodriver.h、r_cg_cgc.h、r_cg_port.h、r_cg_tau.h、r_cg_it8bit.h、r_cg_sau.h
宣　言	ー
説　明	メイン・ユーザ初期化関数を実行後、HALT 状態で UART 受信の割り込みを待つ。 UART 受信完了割り込みが発生したら、その応答として UART 送信の処理に進む。 HALT 解除の理由が UART 受信ではなかった場合は、UART 送信処理に進まず、再び HALT 状態に戻る。
引　数	なし
リターン値	なし

[関数名]　R_MAIN_UserInit	
概　要	メイン・ユーザー初期化関数
ヘッダ	r_cg_macrodriver.h、r_cg_cgc.h、r_cg_port.h、r_cg_tau.h、r_cg_it8bit.h、r_cg_sau.h
宣　言	void R_MAIN_UserInit(void);
説　明	UART0, LED の初期設定を行った後、UART0 および 8 ビット・インターバル・タイマの動作を許可する。最後に、EI 命令で割り込みを許可する。
引　数	なし
リターン値	なし

[関数名]　R_UART0_Receive	
概　要	UART0 受信ステータス初期化関数
ヘッダ	r_cg_macrodriver.h、r_cg_sau.h
宣　言	MD_STATUS R_UART0_Receive(uint8_t * const rx_buf, uint16_t rx_num);
説　明	UART0 の受信ステータスを設定します。
引　数	uint8_t* const rx_buf　：受信データバッファのアドレス uint16_t rx_num　：受信データバッファのサイズ
リターン値	[MD_OK]の場合：受信設定完了 [MD_ARGERROR]の場合：受信設定失敗

[関数名]　R_UART0_Start	
概　要	UART0 動作開始関数
ヘッダ	r_cg_macrodriver.h、r_cg_sau.h
宣　言	void R_UART0_Start(void);
説　明	シリアル･アレイ･ユニット 0・チャネル 0、1 の動作を許可し、UART を通信待機状態にします。
引　数	なし
リターン値	なし

[関数名]　R_UART0_Send	
概　要	UART0 データ送信関数
ヘッダ	r_cg_macrodriver.h、r_cg_sau.h
宣　言	MD_STATUS R_UART0_Send(uint8_t * const tx_buf, uint16_t tx_num);
説　明	UART0 送信の初期設定を行い、データ送信を開始します。
引　数	uint8_t* const tx_buf　　：送信データバッファのアドレス uint16_t tx_num　　：送信データバッファのサイズ
リターン値	[MD_OK]の場合：送信設定完了 [MD_ARGERROR]の場合：送信設定失敗

[関数名]　R_UART0_Stop	
概　要	UART0 動作停止関数
ヘッダ	r_cg_macrodriver.h、r_cg_sau.h
宣　言	void R_UART0_Stop(void);
説　明	シリアル･アレイ･ユニット 0・チャネル 0、1 を動作停止させ、UART 通信を停止します。
引　数	なし
リターン値	なし

[関数名]　r_uart0_interrupt_receive	
概　要	UART0 受信完了割り込み関数
ヘッダ	r_cg_macrodriver.h、r_cg_sau.h
宣　言	static void __near r_uart0_interrupt_receive(void)
説　明	受信したデータを RAM に格納して、アドレスと受信回数値を更新します。
引　数	なし
リターン値	なし

[関数名]　r_uart0_interrupt_error	
概　要	UART0 受信エラー割り込み関数
ヘッダ	r_cg_macrodriver.h、r_cg_sau.h
宣　言	static void __near r_uart0_interrupt_error(void)
説　明	受信データを RAM に格納して、r_uart0_callback_error 関数に対し、検出したエラーに対応した応答をさせます。
引　数	なし
リターン値	なし

[関数名]　r_uart0_interrupt_send	
概　要	UART0 送信完了割り込み関数
ヘッダ	r_cg_macrodriver.h、r_cg_sau.h
宣　言	static void __near r_uart0_interrupt_send(void)
説　明	データを送信してポインタとカウンタを更新します。送信データが残っていない場合は、送信完了処理をします。
引　数	なし
リターン値	なし

[関数名] r_it8bit0_interrupt	
概　要	8 ビット・インターバル·タイマ割り込み関数
ヘッダ	r_cg_macrodriver.h、r_cg_it8bit.h、r_cg_sau.h、r_cg_tau.h
宣　言	static void __near r_it8bit0_channel0_interrupt(void)
説　明	定期的に（本アプリケーションでは 1 秒毎）に行われるボーレート補正を実施する関数です。まず UART 通信を無効にし、タイマ・アレイ・ユニット 0・チャネル 1 およびチャネル 3 を起動します。チャネル 1 で生成された測定対象方形波のパルス幅をチャネル 3 で測定し、その測定結果から逆算して、UART のボーレートが理想値に最も近づくよう、クロックの分周値を修正します。そのあと UART 通信を有効にし、再び UART 通信待機状態へと戻します。
引　数	なし
リターン値	なし

[関数名] r_tau0_channel3_interrupt	
概　要	TAU チャネル 3 カウント完了割り込み関数
ヘッダ	r_cg_macrodriver.h、r_cg_tau.h
宣　言	static void __near r_tau0_channel3_interrupt(void)
説　明	タイマ・アレイ・ユニット・チャネル 3 によるパルス幅測定の結果を、グローバル変数に格納します。
引　数	なし
リターン値	なし

[関数名] R_TAU0_Channel1_Start	
概　要	TAU チャネル 1 動作開始関数
ヘッダ	r_cg_macrodriver.h、r_cg_tau.h
宣　言	void R_TAU0_Channel1_Start(void);
説　明	タイマ・アレイ・ユニット・チャネル 1 の動作を許可し、方形波出力を開始します。
引　数	なし
リターン値	なし

[関数名] R_TAU0_Channel3_Start	
概　要	TAU チャネル 3 動作開始関数
ヘッダ	r_cg_macrodriver.h、r_cg_tau.h
宣　言	void R_TAU0_Channel3_Start(void);
説　明	タイマ・アレイ・ユニット・チャネル 3 の動作を許可し、パルス幅測定開始します。
引　数	なし
リターン値	なし

[関数名]	R_TAU0_Channel1_Stop
概　要	TAU チャネル 1 動作停止関数
ヘッダ	r_cg_macrodriver.h、r_cg_tau.h
宣　言	void R_TAU0_Channel1_Stop(void);
説　明	タイマ・アレイ・ユニット・チャネル 1 の動作を停止します。
引　数	なし
リターン値	なし

[関数名]	R_TAU0_Channel3_Stop
概　要	TAU チャネル 3 動作停止関数
ヘッダ	r_cg_macrodriver.h、r_cg_tau.h
宣　言	void R_TAU0_Channel3_Stop(void);
説　明	タイマ・アレイ・ユニット・チャネル 3 の動作を停止します。
引　数	なし
リターン値	なし

[関数名]	R_IT8Bit0_Channel0_Start
概　要	8 ビット・インターバル･タイマ動作開始関数
ヘッダ	r_cg_macrodriver.h、r_cg_it8bit.h
宣　言	void R_IT8Bit0_Channel0_Start(void);
説　明	8 ビット・インターバル･タイマ動作を許可にし、カウントを開始します。
引　数	なし
リターン値	なし

[関数名]	R_IT8Bit0_Channel0_Stop
概　要	8 ビット・インターバル･タイマ動作停止関数
ヘッダ	r_cg_macrodriver.h、r_cg_it8bit.h
宣　言	void R_IT8Bit0_Channel0_Stop(void);
説　明	8 ビット・インターバル･タイマ動作を停止します。
引　数	なし
リターン値	なし

5.8 フローチャート

図 5.1 に本アプリケーションノートの全体フローを示します。

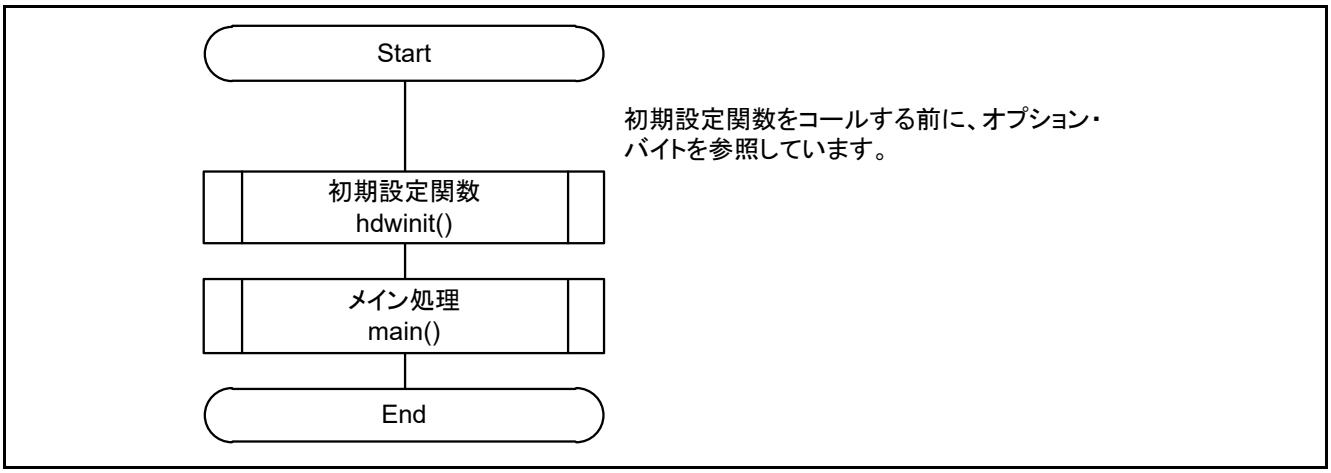

図 5.1 全体フロー

5.8.1 初期設定

図 5.2 に初期設定関数のフローチャートを示します。

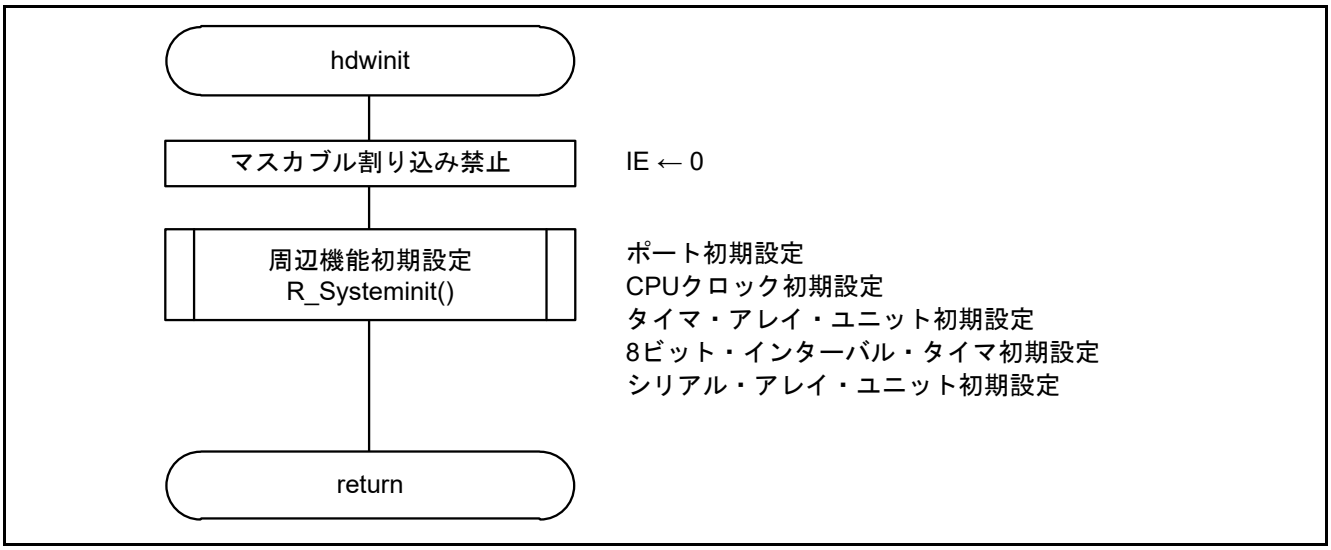

図 5.2 初期設定関数

5.8.2 周辺機能初期設定

図 5.3 に周辺機能初期設定関数のフローチャートを示します。

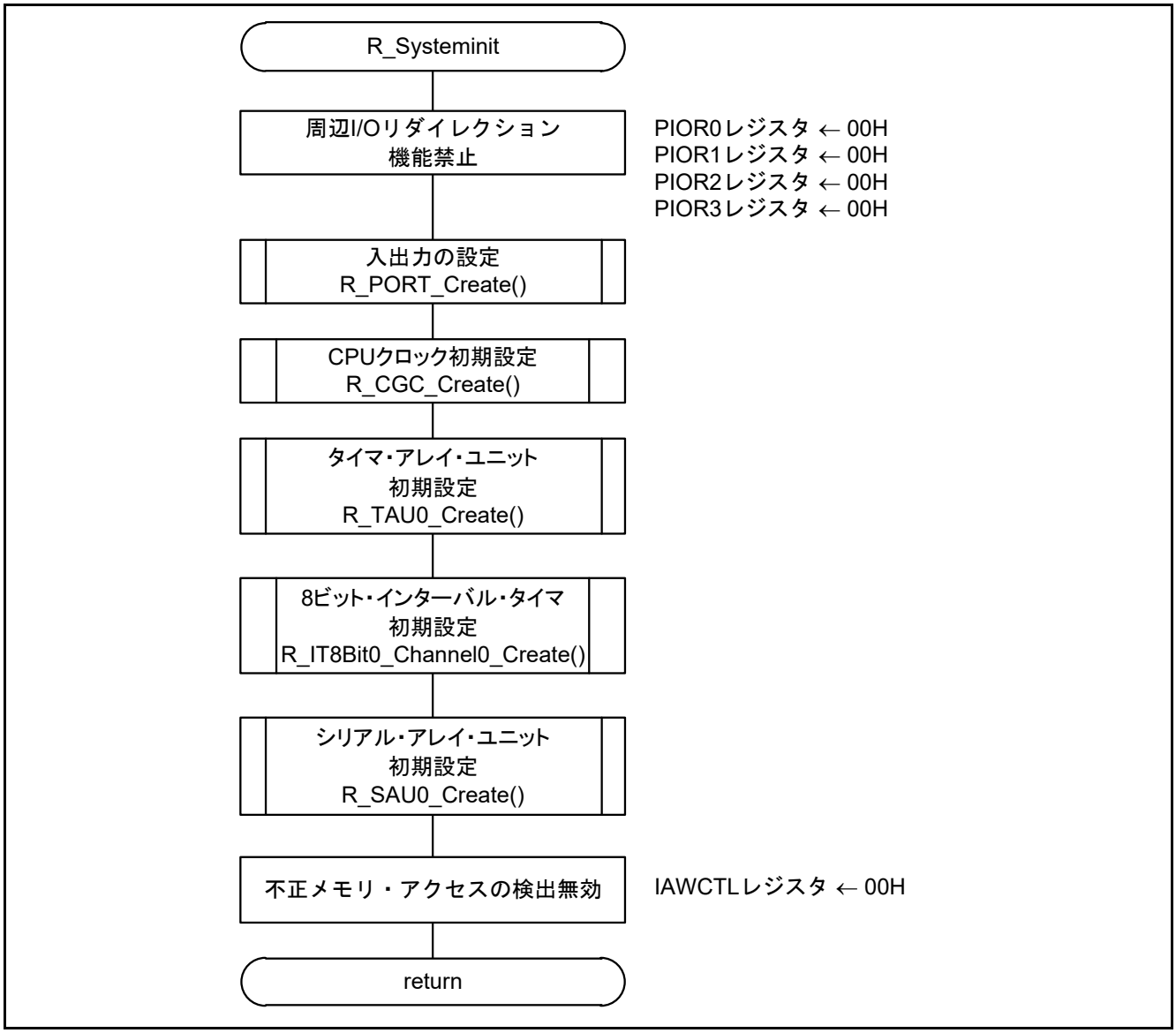

図 5.3 周辺機能初期設定関数

5.8.3　入出力ポートの設定

図 5.4 に入出力ポートのフローチャートを示します。

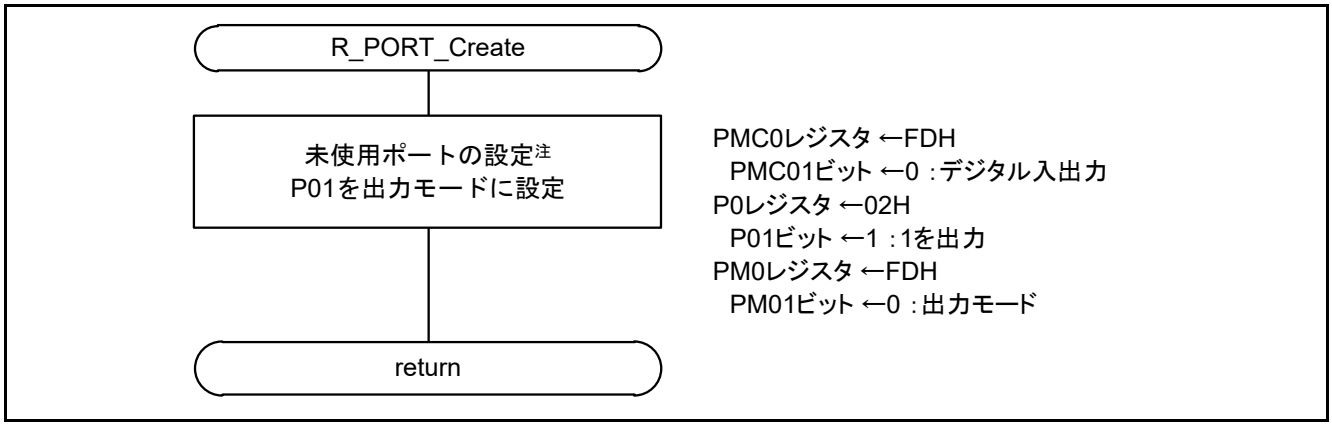

図 5.4　入出力ポートの設定

注　未使用ポートの設定については、RL78/G13 初期設定(R01AN2575J) 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 VDD 又は VSS に接続してください。

5.8.4　CPU クロック初期設定

図 5.5 に CPU クロック初期設定関数のフローチャートを示します。

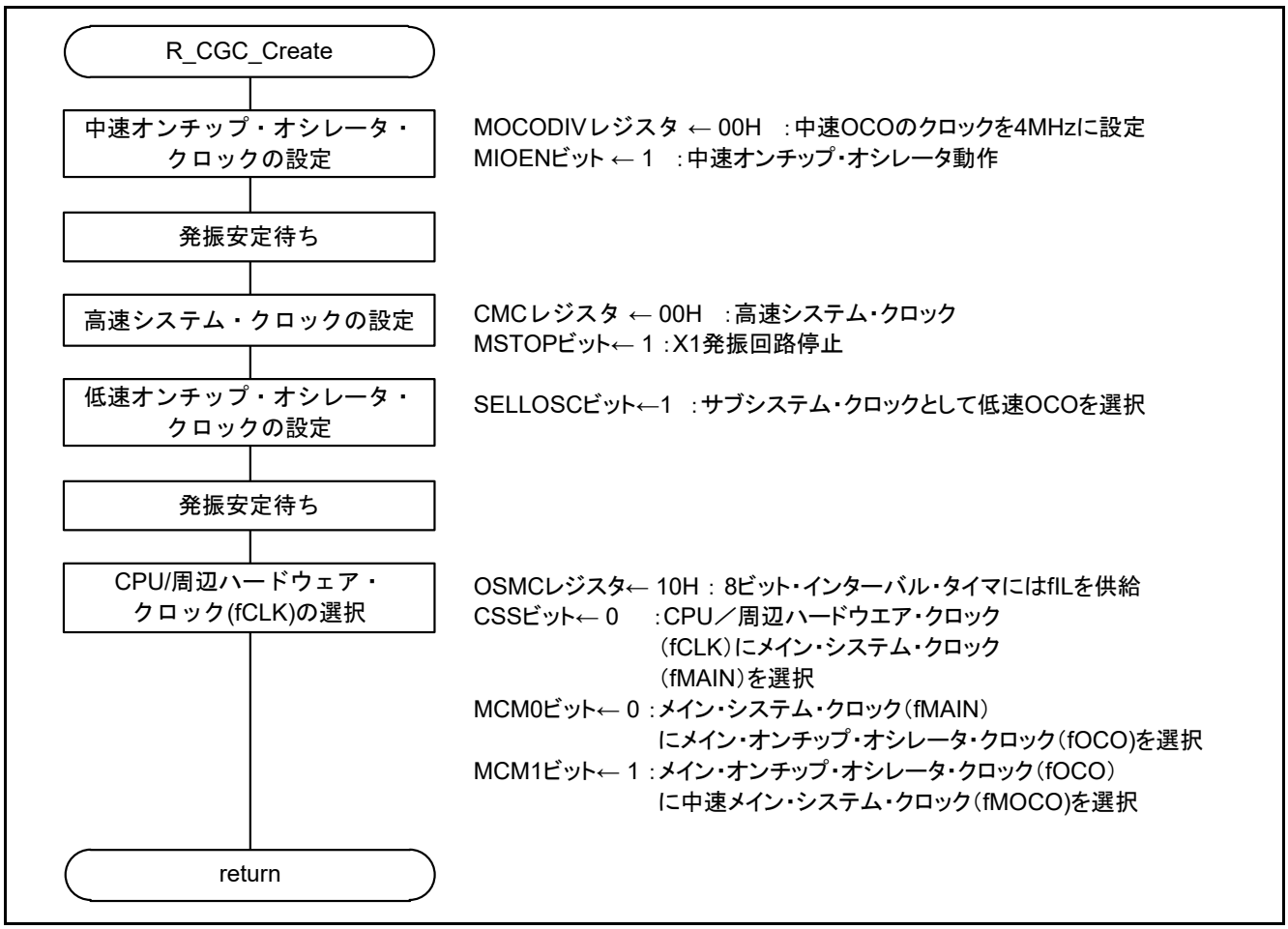

図 5.5　CPU クロック初期設定関数

5.8.5　タイマ・アレイ・ユニット初期設定

図 5.6、図 5.7 にタイマ･アレイ･ユニット初期設定のフローチャートを示します。

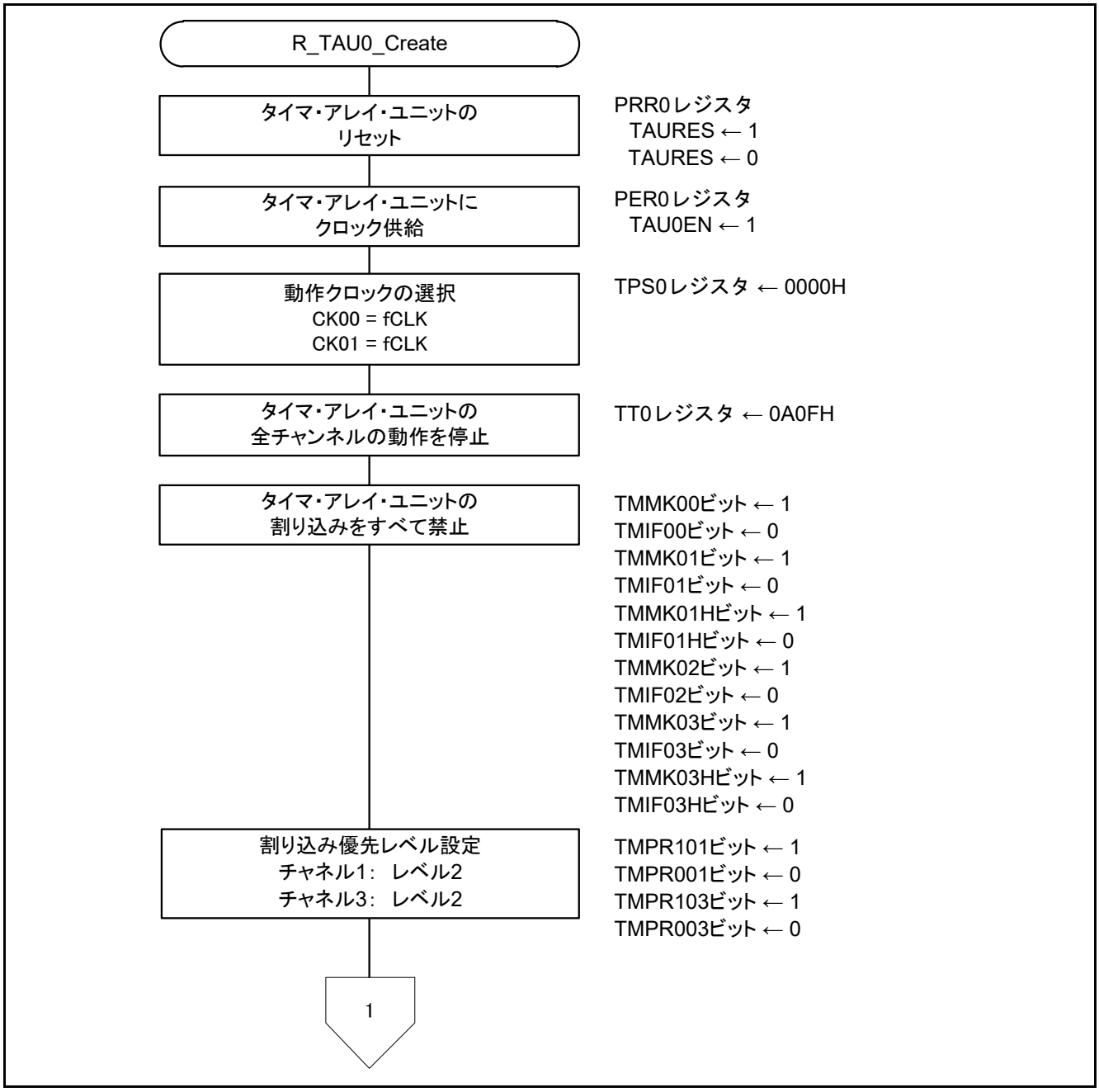

図 5.6　タイマ･アレイ･ユニット初期設定（1/2）

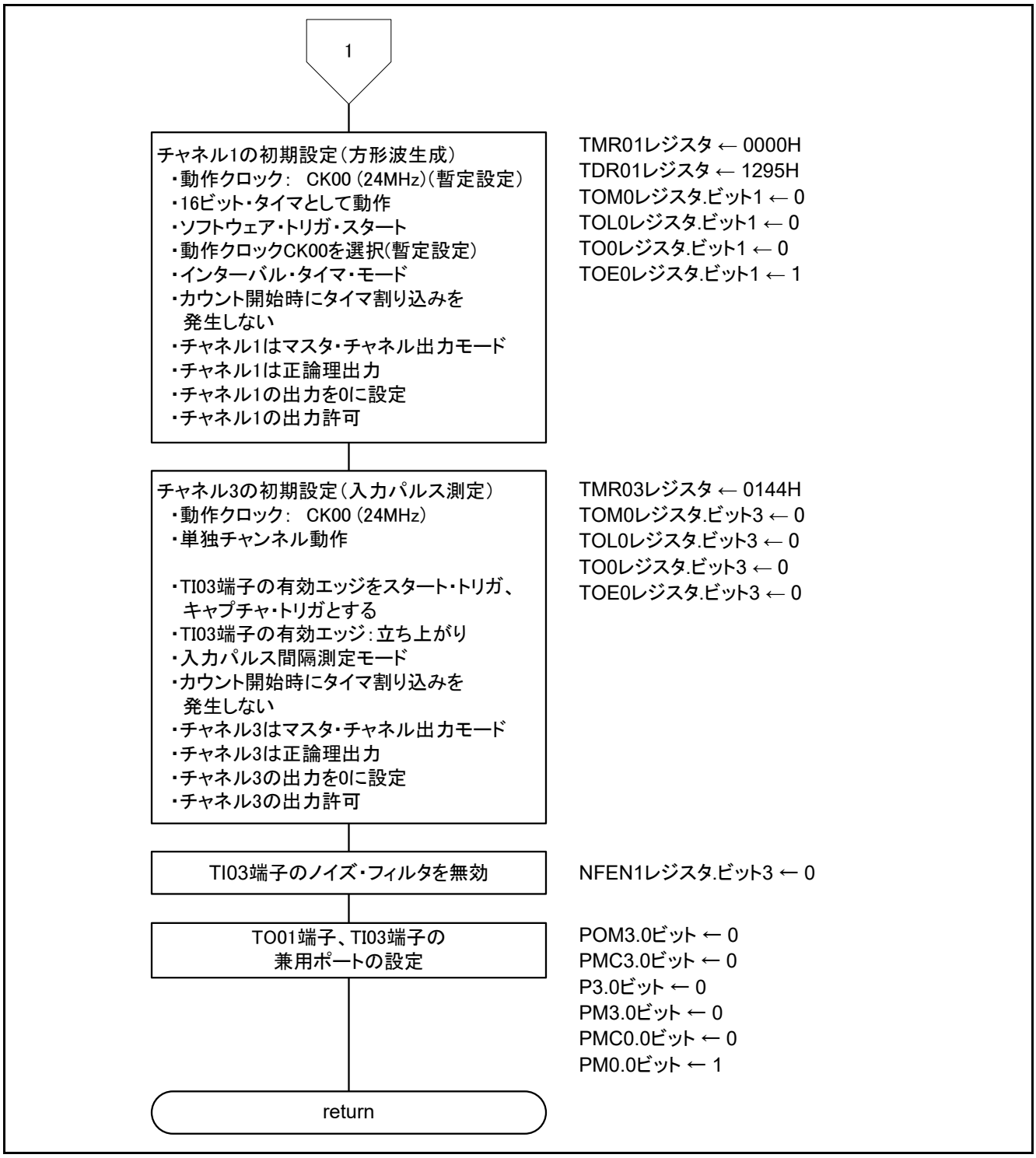

図 5.7 タイマ・アレイ・ユニット初期設定 (2/2)

5.8.6 8 ビット・インターバル・タイマ初期設定関数

図 5.8 に 8 ビット・インターバル・タイマ初期設定関数のフローチャートを示します。

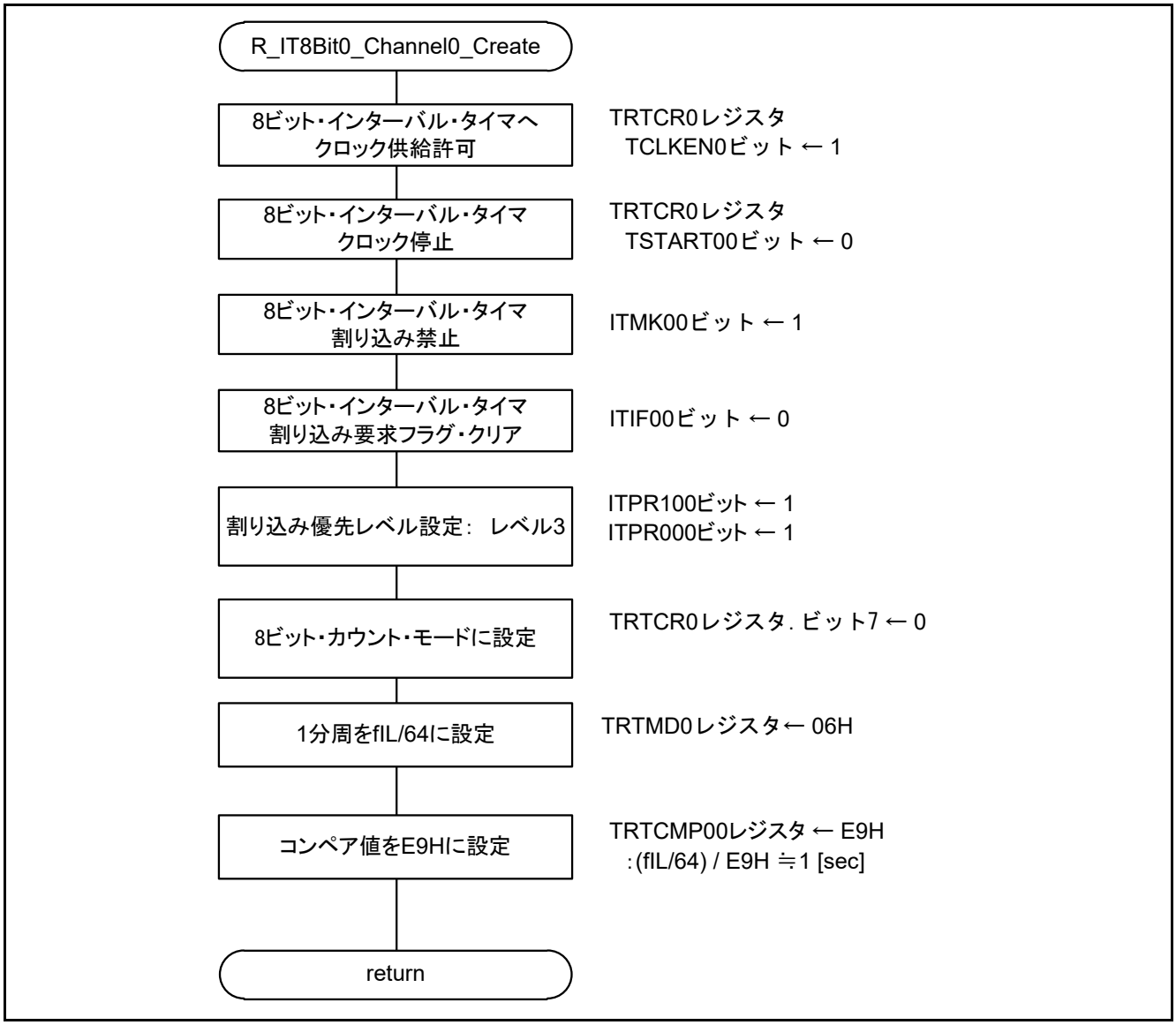

図 5.8 8 ビット・インターバル・タイマ初期設定関数

5.8.7　シリアル・アレイ・ユニット初期設定

図 5.9 にシリアル･アレイ･ユニット初期設定関数のフローチャートを示します。

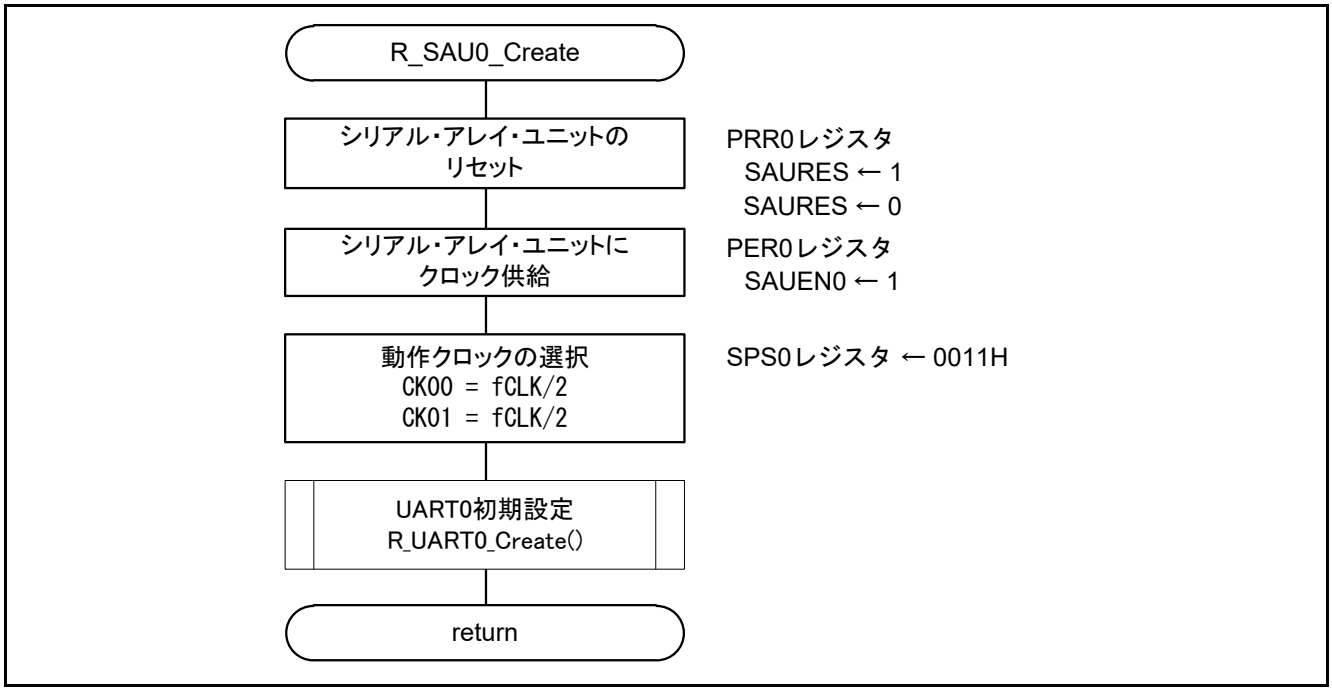

図 5.9　シリアル･アレイ･ユニット初期設定関数

5.8.8　UART0 初期設定

図 5.10、図 5.11 に UART0 初期設定のフローチャートを示します。

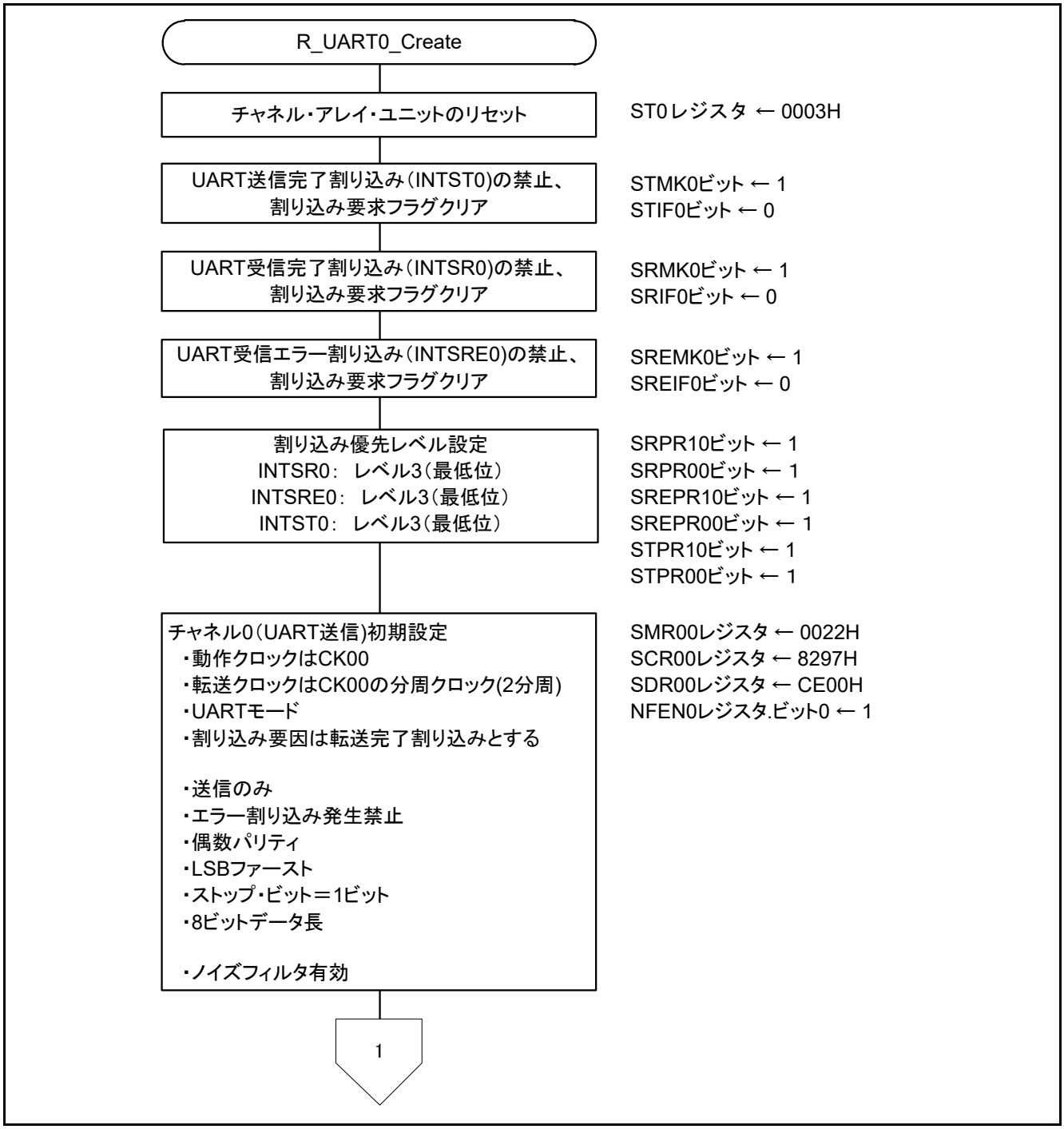

図 5.10　UART0 初期設定

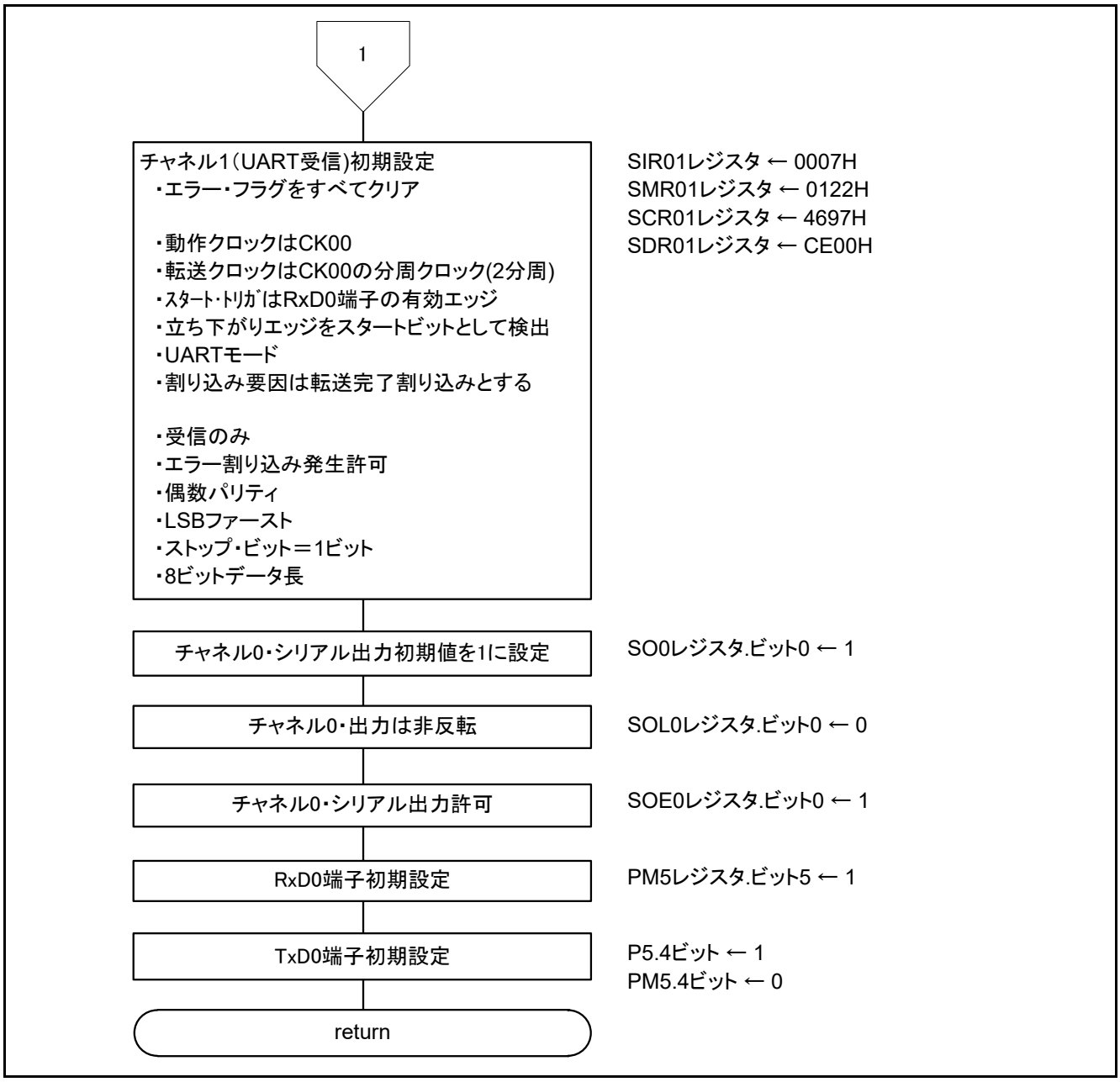

図 5.11　UART0 初期設定

5.8.9 メイン関数

図 5.12、図 5.13 にメイン関数のフローチャートを示します。

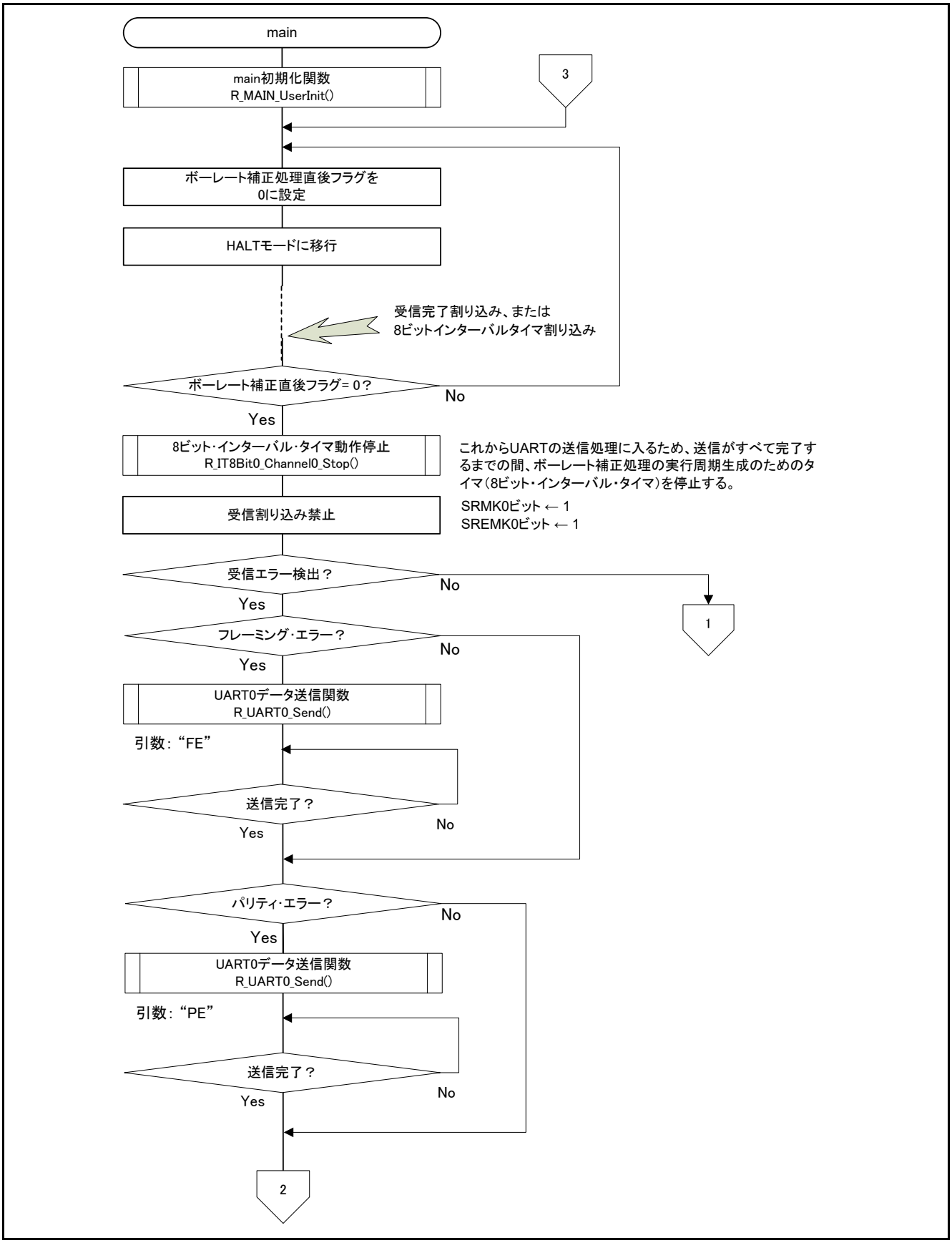

図 5.12 メイン関数（1/2）

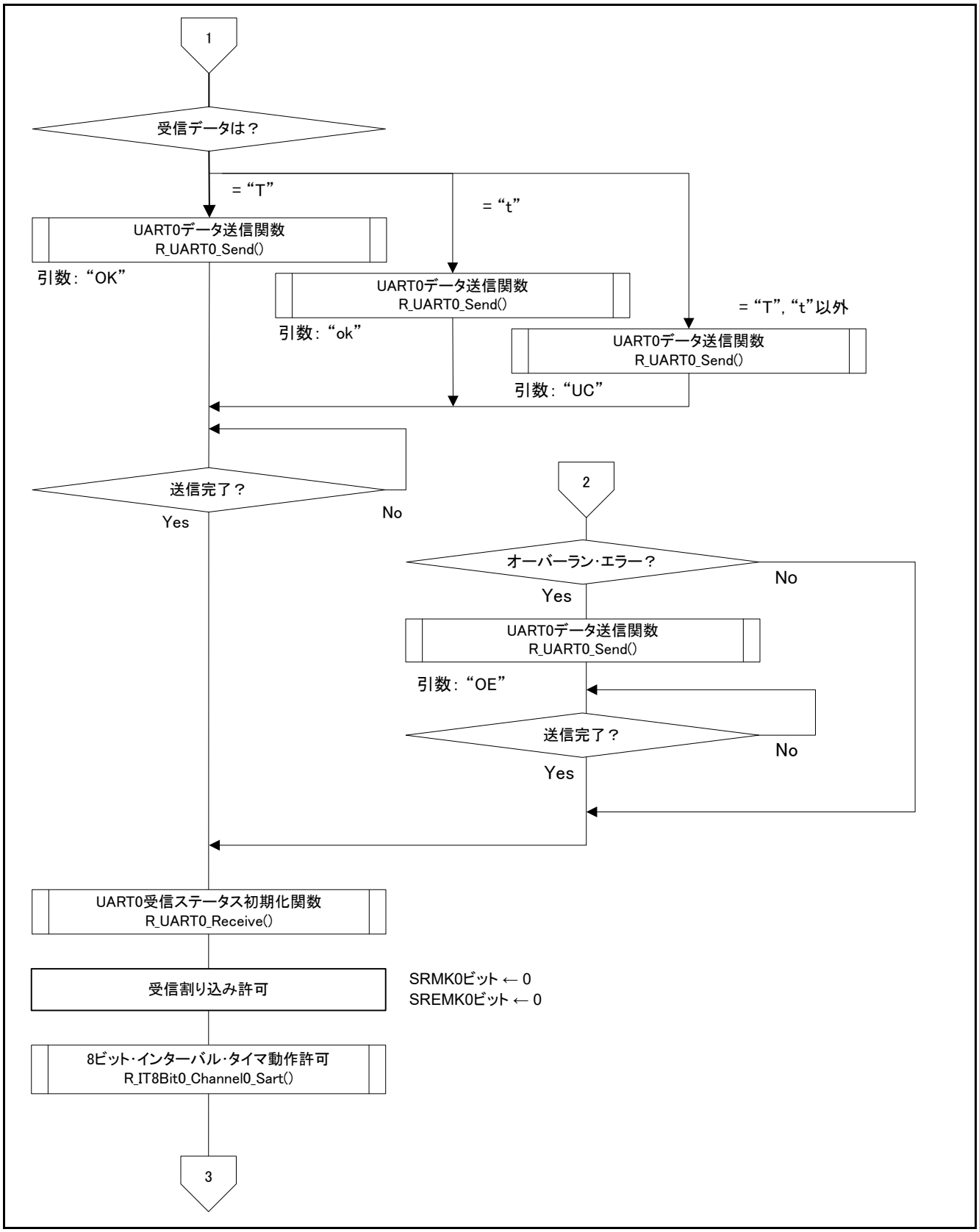

図 5.13　メイン処理（2/2）

5.8.10　メイン・ユーザー初期設定

図 5.14 にメイン・ユーザー初期設定関数のフローチャートを示します。

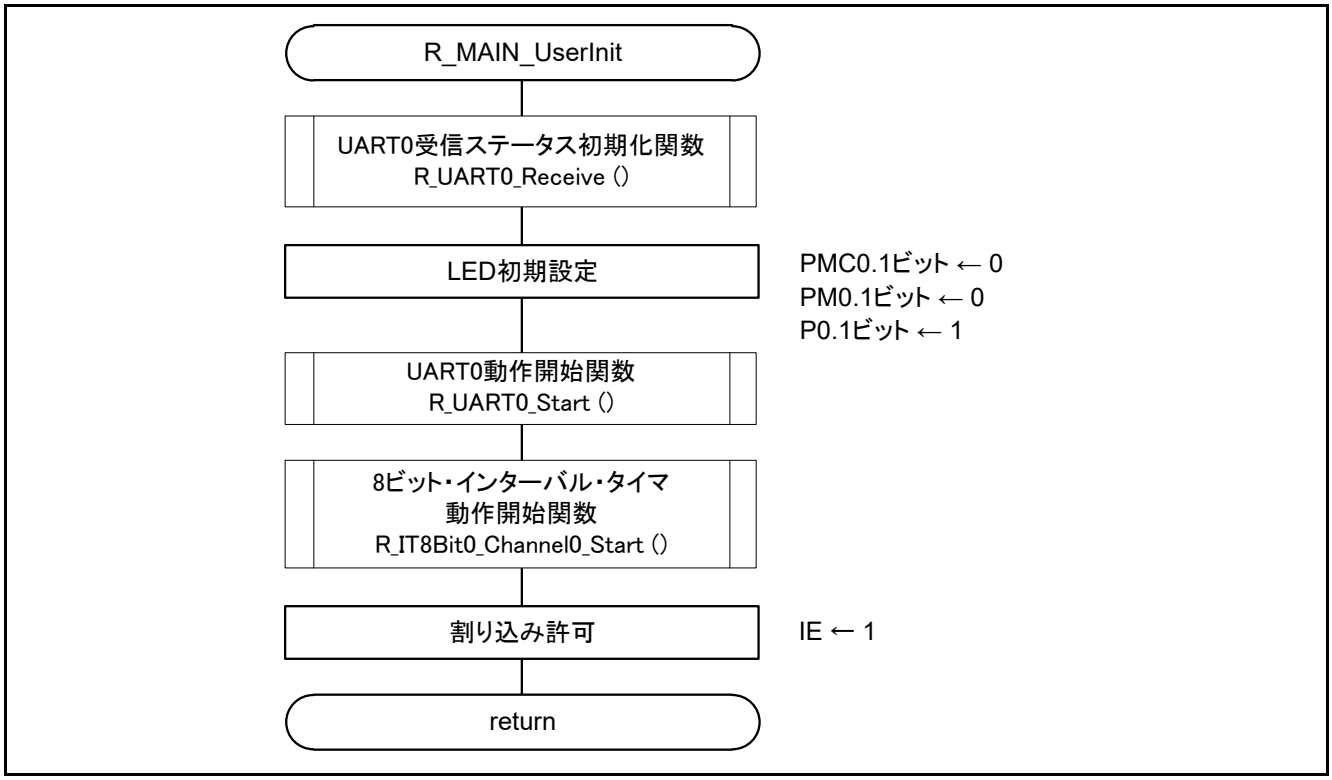

図 5.14　メイン・ユーザー初期設定関数

5.8.11 UART0 受信ステータス初期化

図 5.15 に UART0 受信ステータス初期化関数のフローチャートを示します。

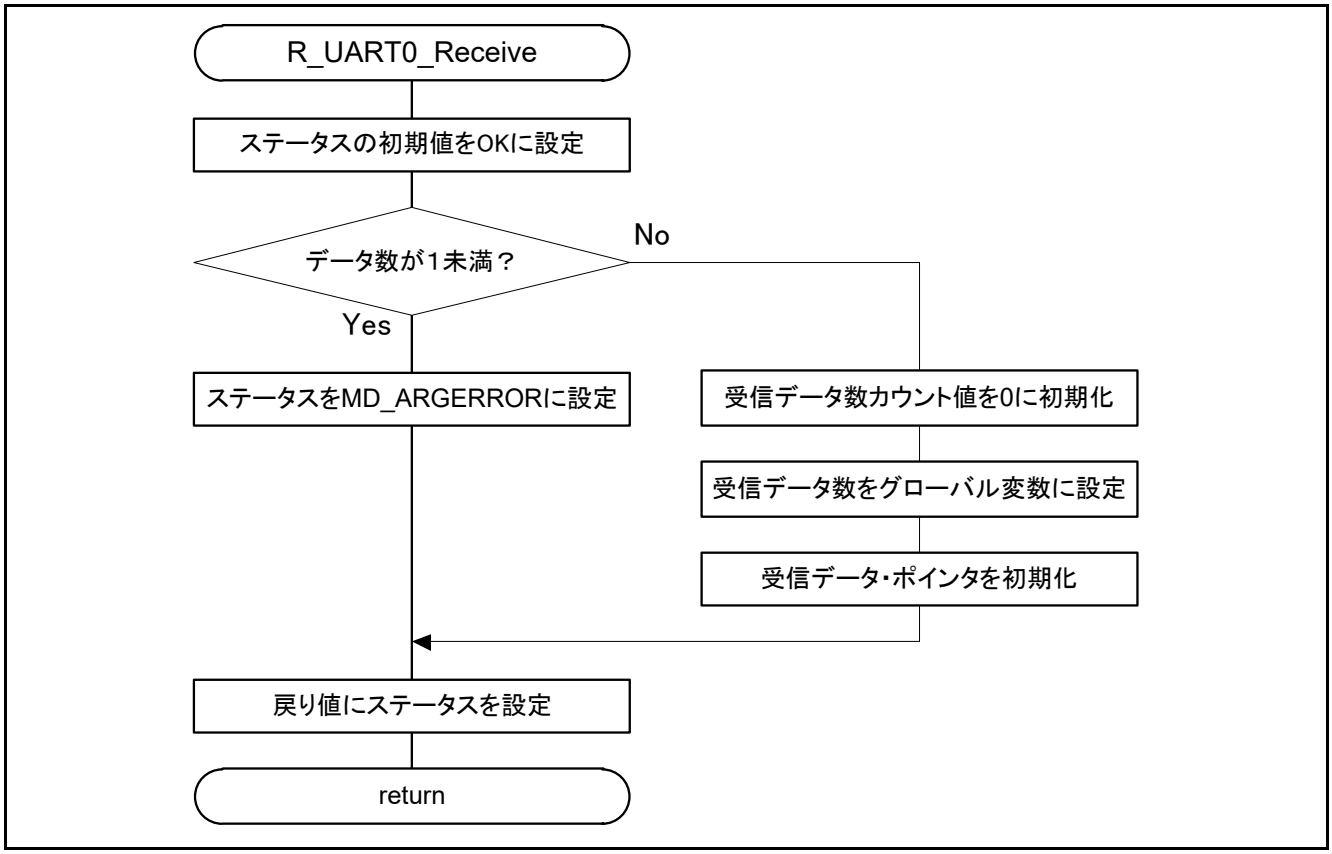

図 5.15 UART0 受信ステータス初期化関数

5.8.12 UART0 動作開始関数

図 5.16 に UART0 動作開始関数のフローチャートを示します。

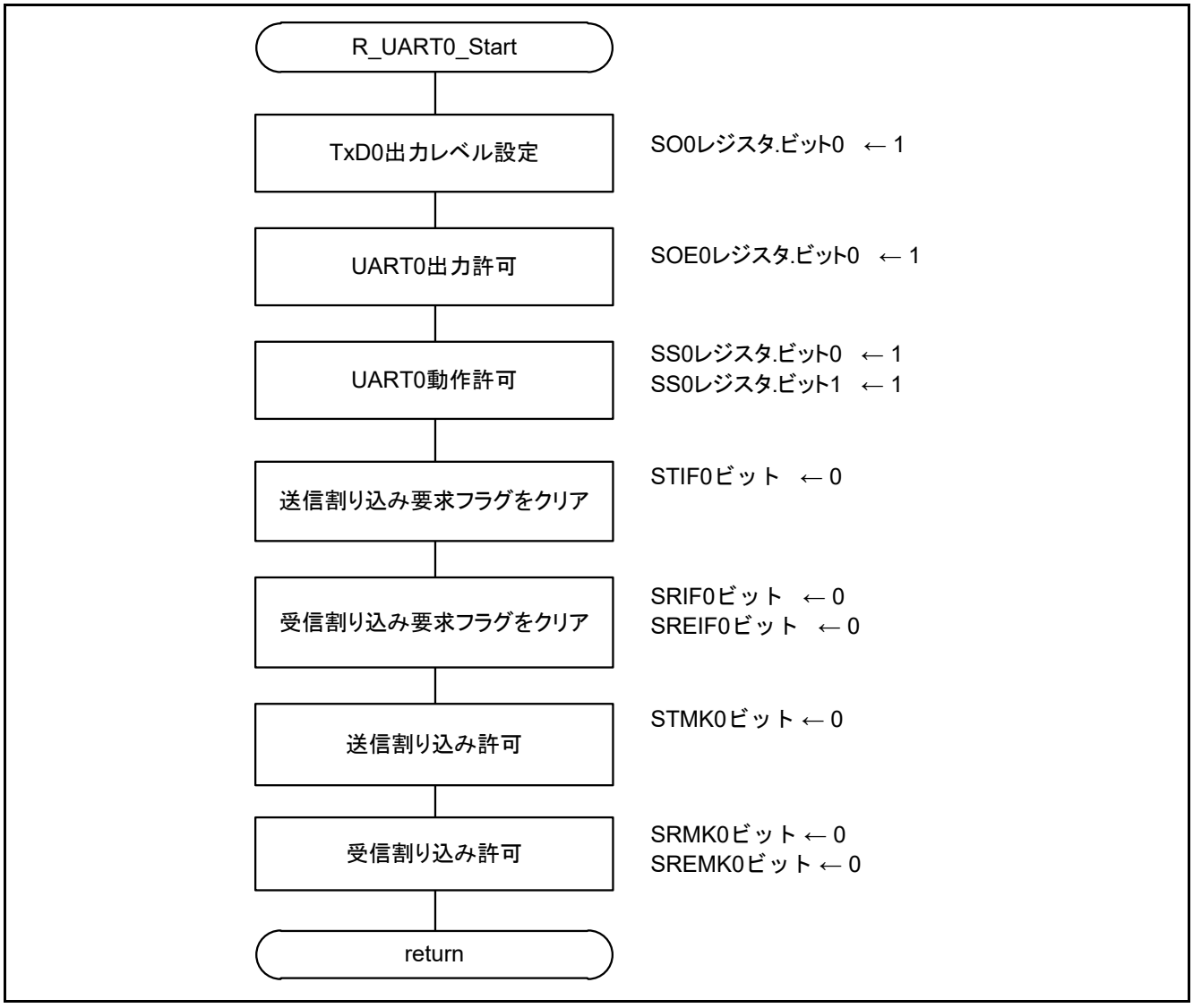

図 5.16 UART0 動作開始関数

5.8.13 UART0 データ送信

図 5.17 に UART0 データ送信関数のフローチャートを示します。

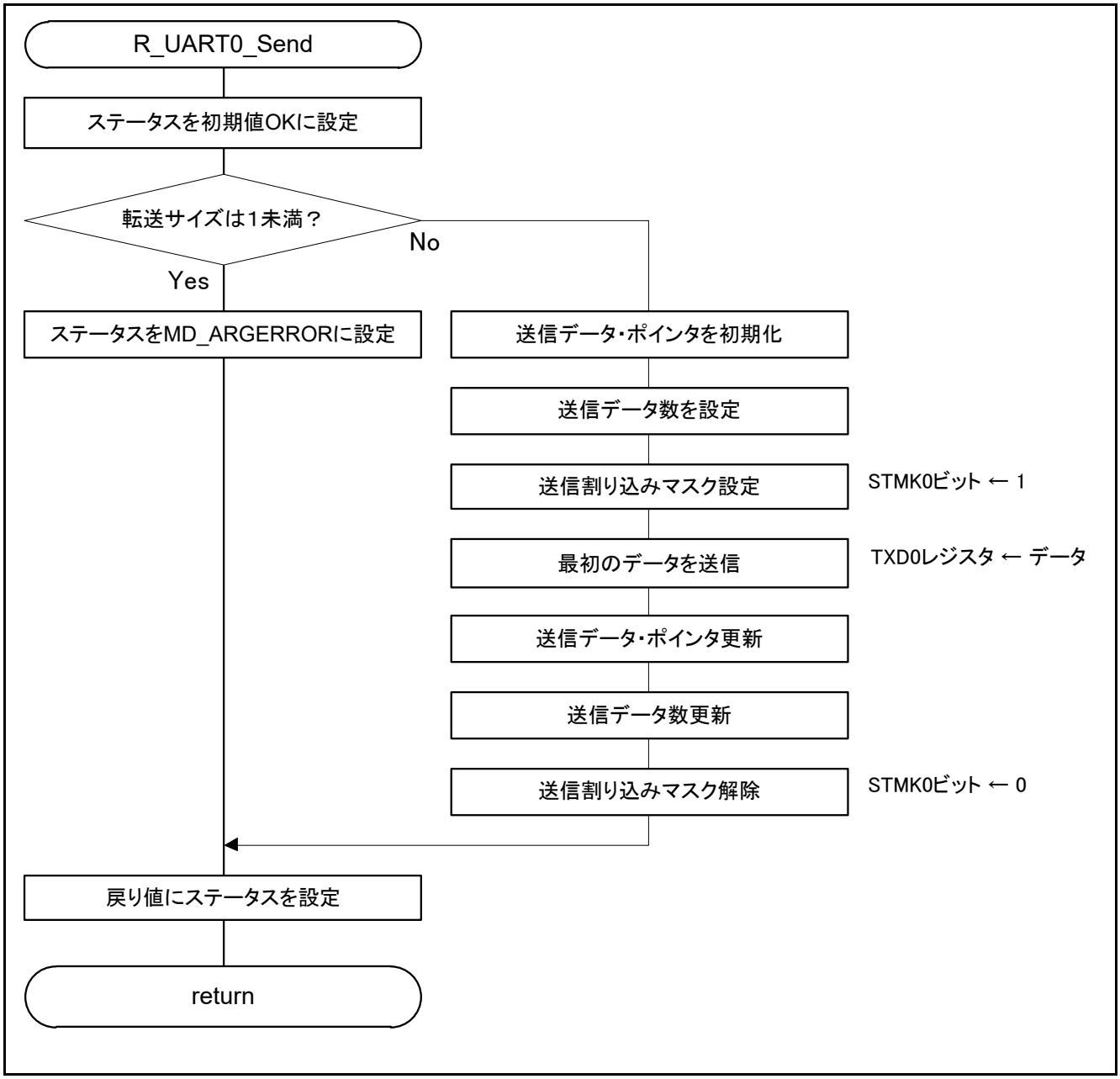

図 5.17 UART0 データ送信関関数

5.8.14 UART0 動作停止

図 5.18 に UART0 動作停止関数のフローチャートを示します。

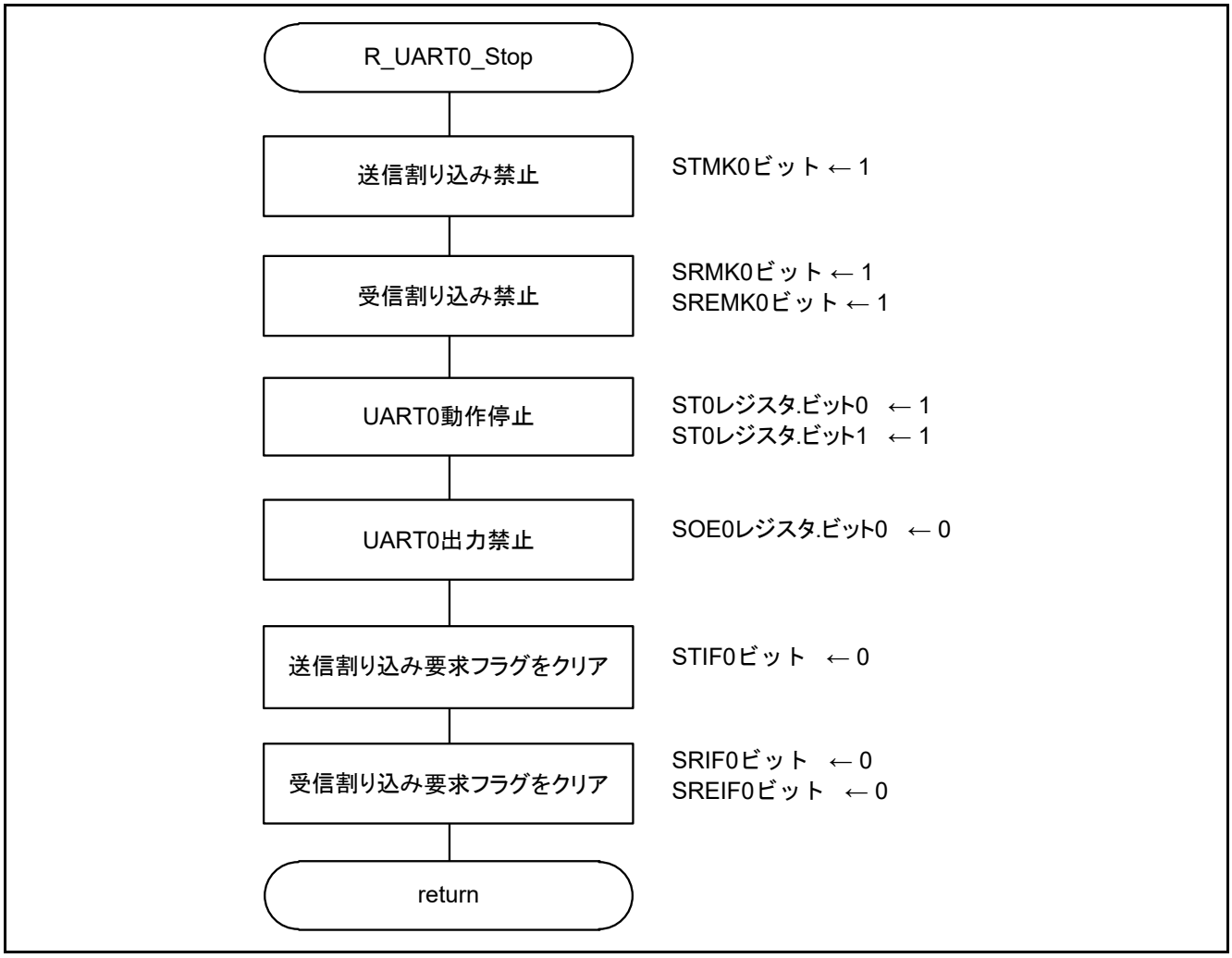

図 5.18 UART0 動作停止関数

5.8.15 UART0 受信完了割り込み

図 5.19 に UART0 受信完了割り込み関数のフローチャートを示します。

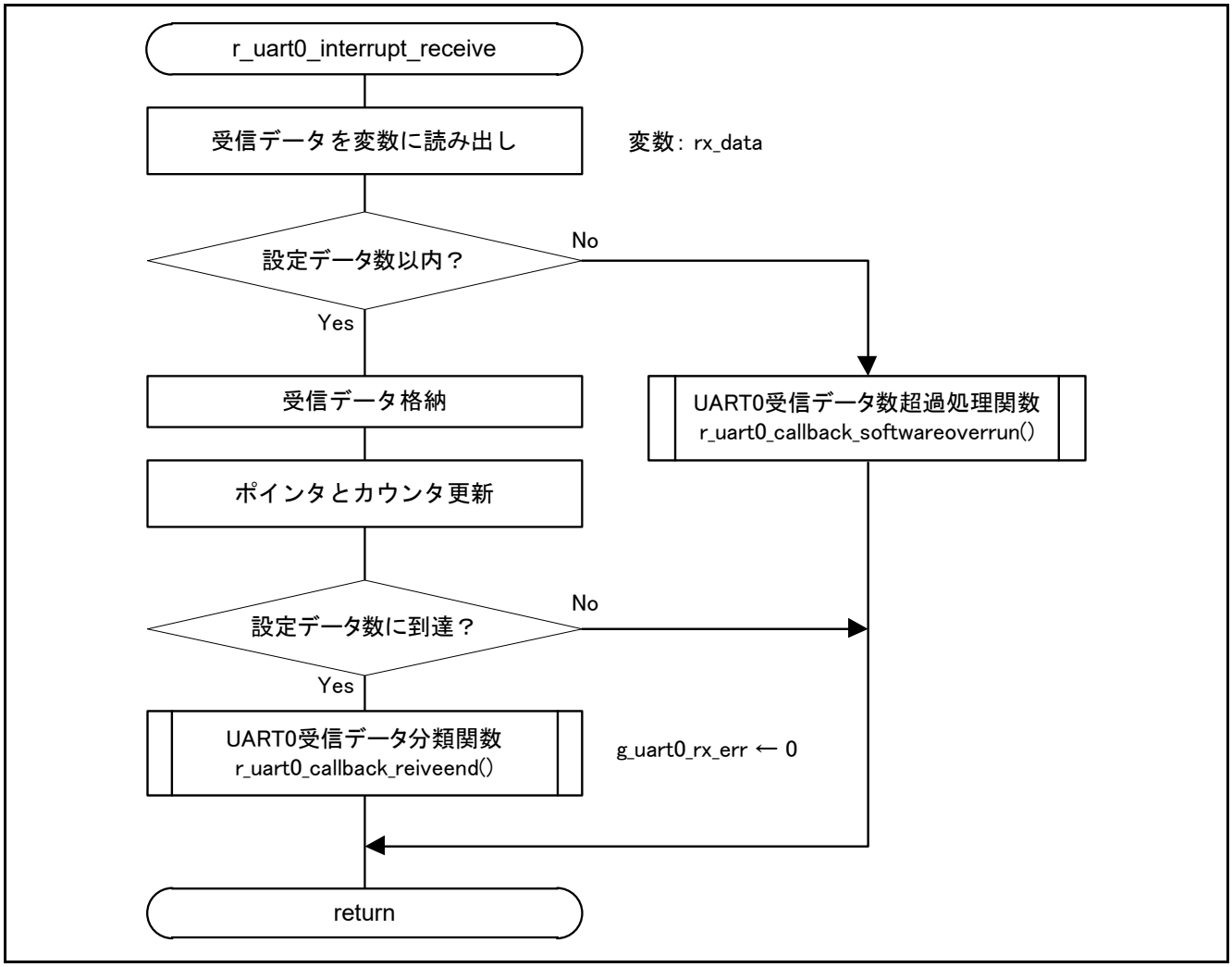

図 5.19 UART0 受信完了割り込み関数

5.8.16 UART0 受信エラー割り込み

図 5.20 に UART0 受信エラー割り込み関数のフローチャートを示します。

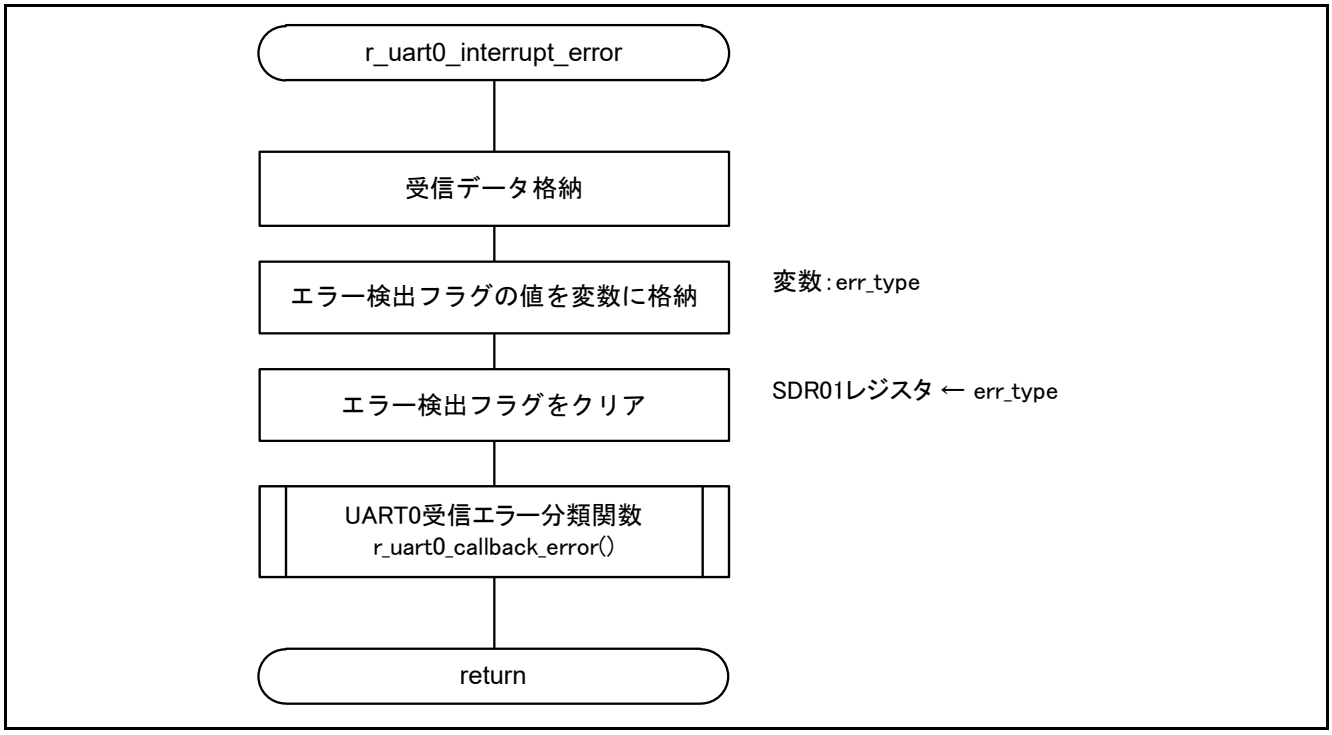

図 5.20 UART0 受信エラー割り込み関数

5.8.17 UART0 送信完了割り込み

図 5.21 に UART0 送信完了割り込み関数のフローチャートを示します。

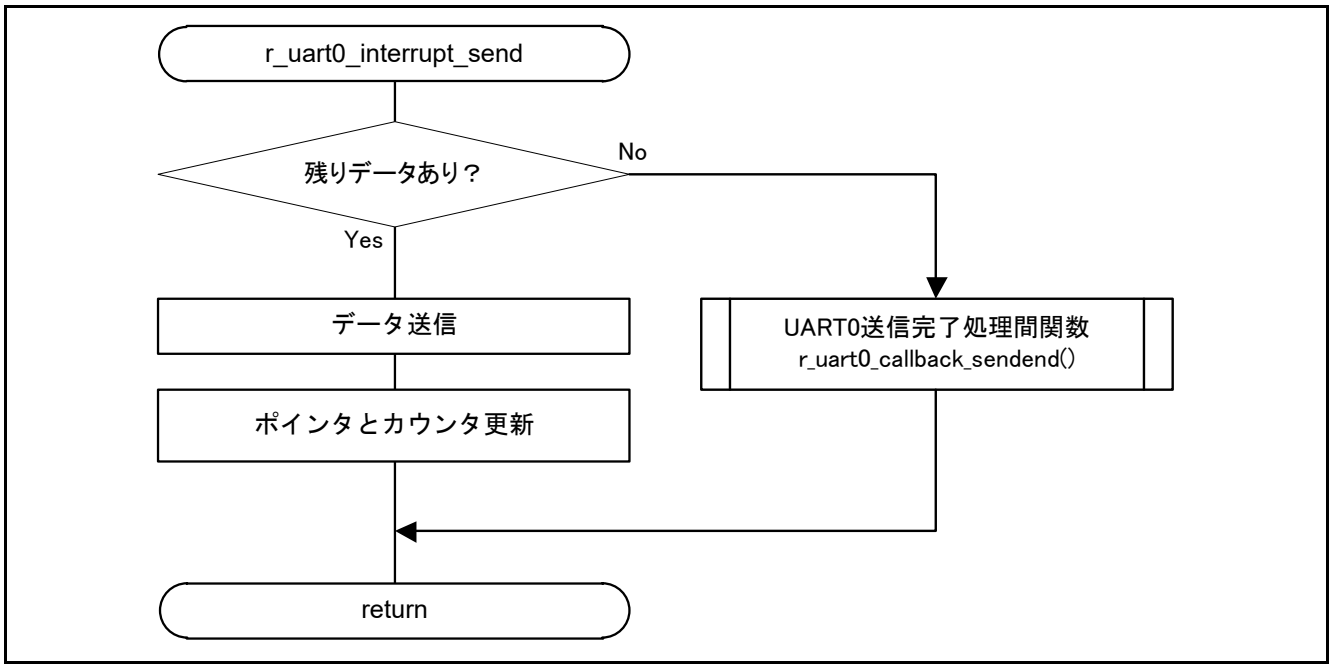

図 5.21 UART0 送信完了割り込み関数

5.8.18 8 ビット・インターバル・タイマ割り込み

図 5.22、図 5.23 に 8 ビット・インターバル・タイマ割り込み関数のフローチャートを示します。

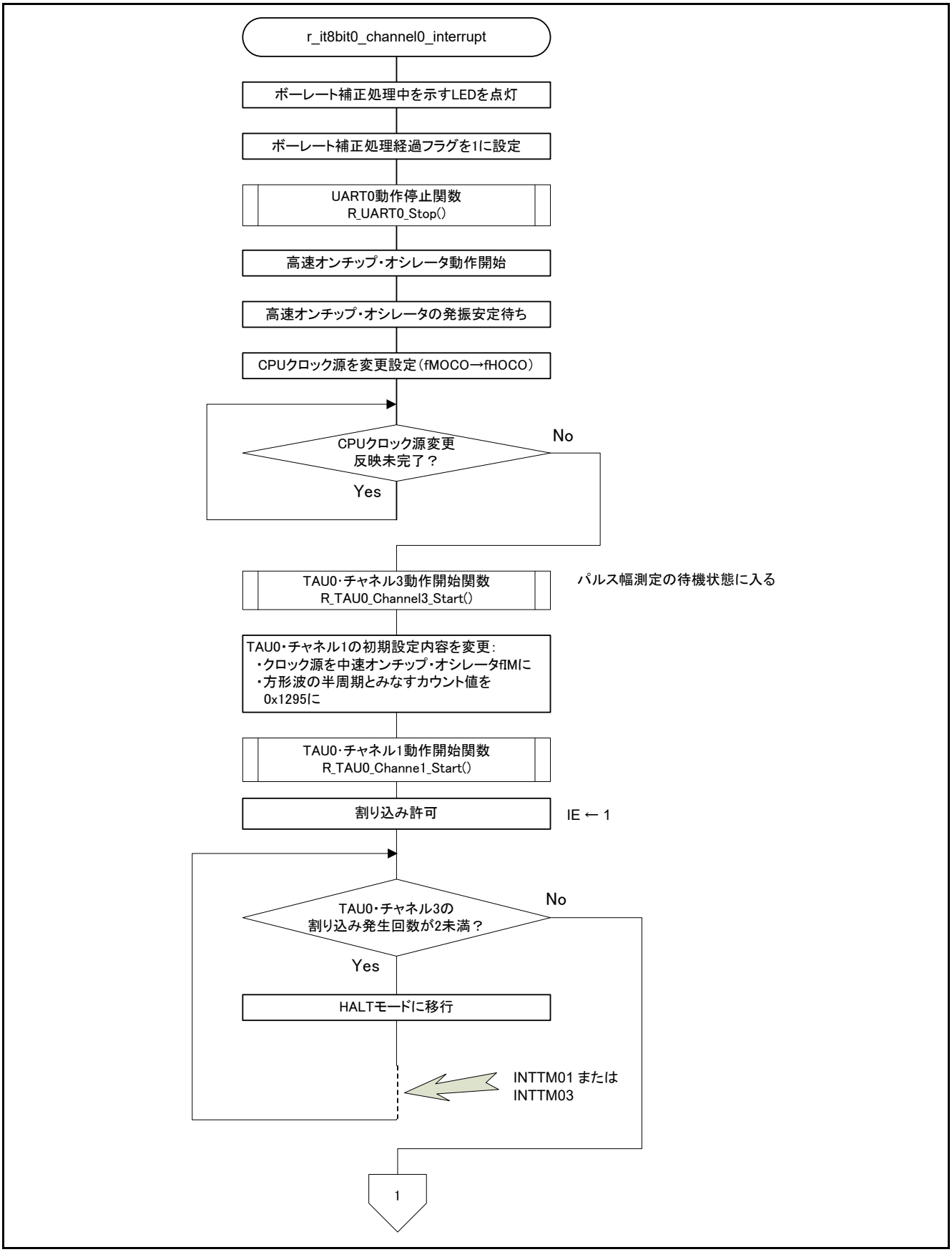

図 5.22 8 ビット・インターバル・タイマ割り込み関数（1/2）

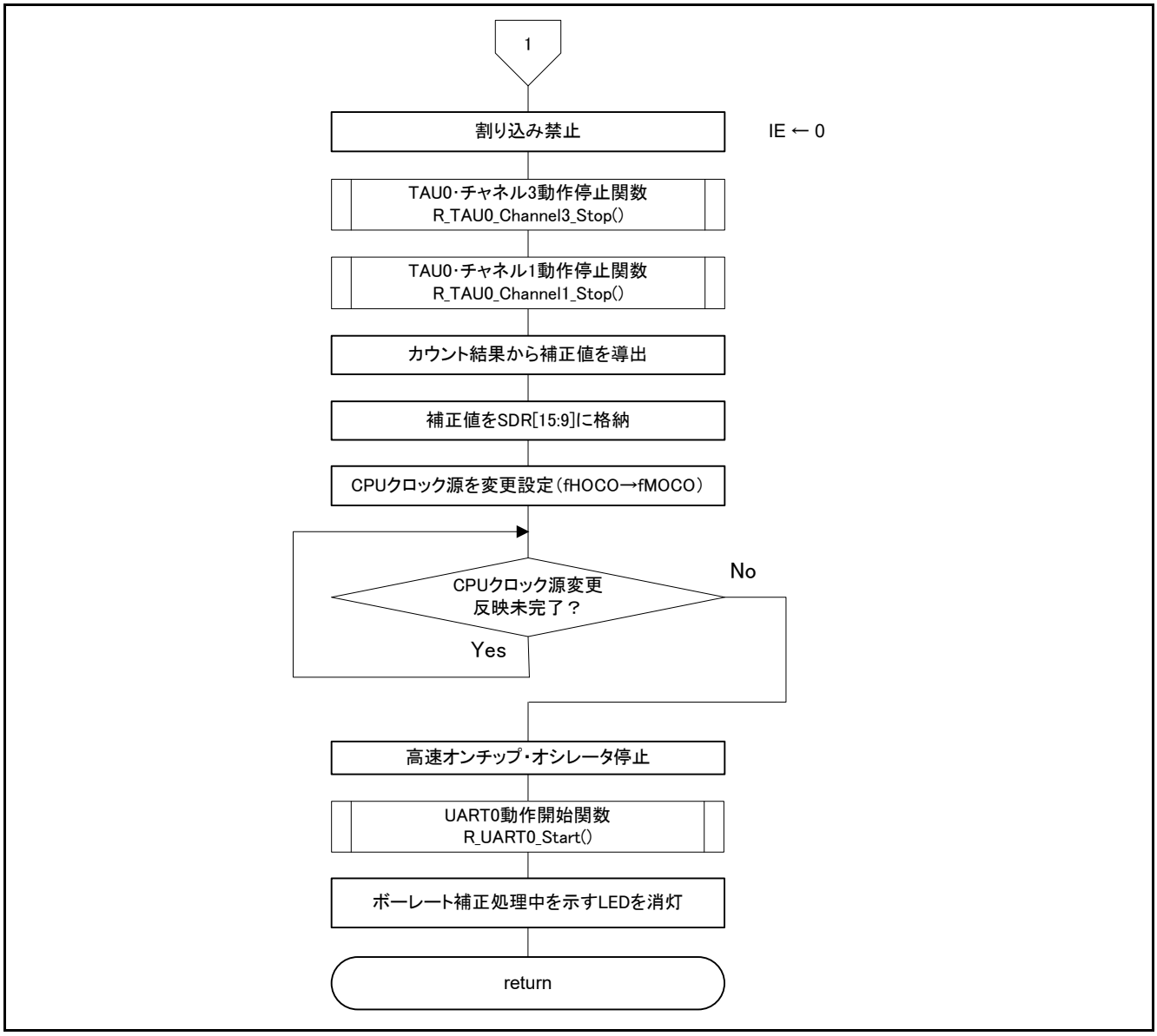

図 5.23　8 ビット・インターバル・タイマ割り込み関数（2/2）

5.8.19 TAU チャネル 3 カウント完了割り込み

図 5.24 に TAU チャネル 3 カウント完了割り込み関数のフローチャートを示します。

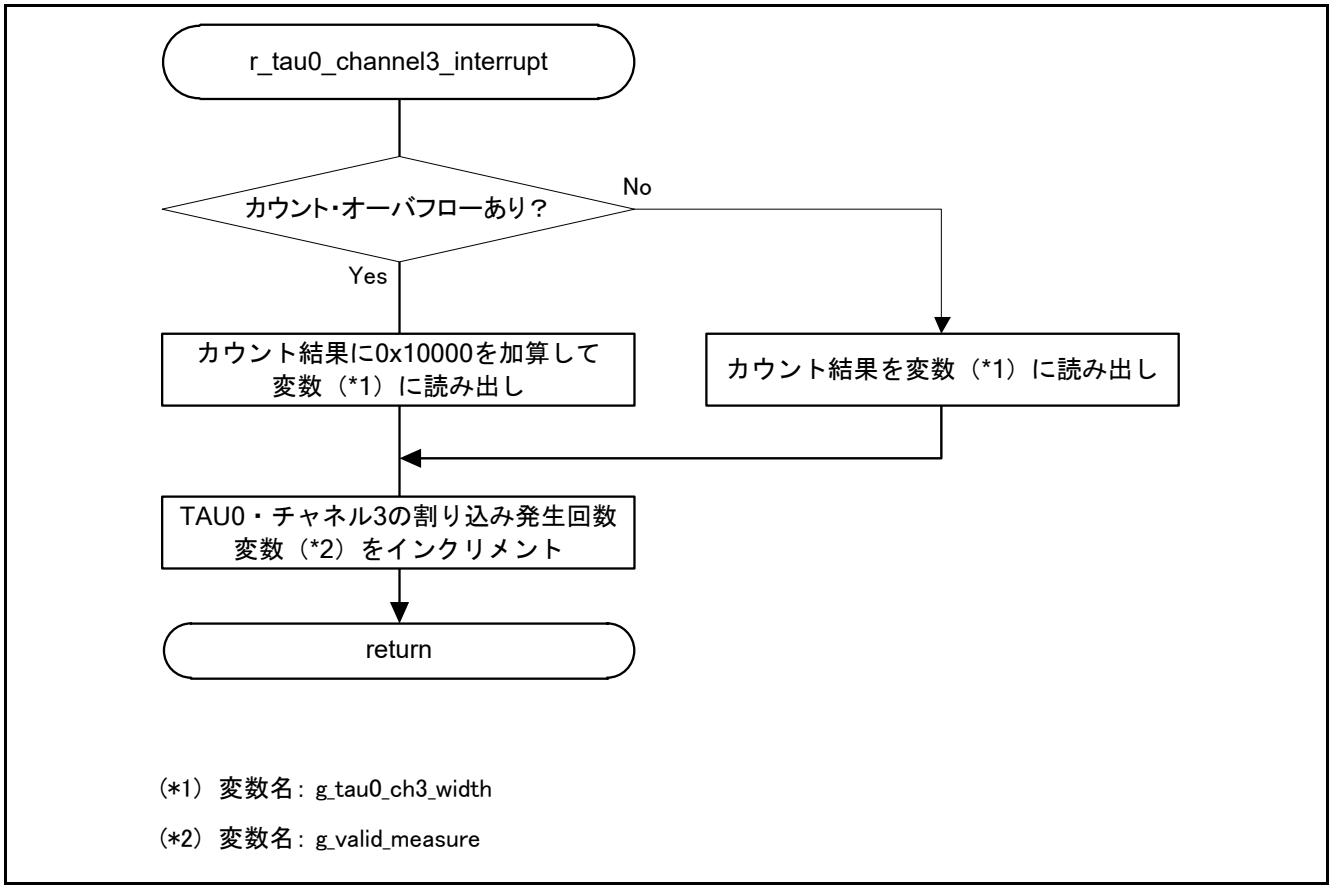

図 5.24 TAU チャネル 3 カウント完了割り込み関数

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/G11 ユーザーズマニュアル ハードウェア編 (R01UH0637J)
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/contact/

改訂記録	RL78/G11 中速オンチップ・オシレータでの UART 通信の実現 CC-RL[RL78][G11]　アプリケーションノート　[タイトル]

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2016.10.31	—	初版発行
1.01	2017.01.12	7	ソフトウェアの変更に伴い、動作概要の修正
		13-15	宣言と引数を修正
		26,27	メイン関数のフローチャートを修正
		35,36	8 ビット・インターバル・タイマ割り込み関数のフローチャートを修正
1.02	2023.08.04	5	動作確認条件更新

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. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がV_{IL}（Max.）からV_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）からV_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   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、将来の拡張機能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。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、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、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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。
6. 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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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。
9. 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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。
10. 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。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。
11. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 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。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配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。

(Rev.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。