
RL78/G13

R01AN4671JJ0100

Rev.1.00

2019.05.07

ソフトウェア UART CC-RL

要旨

本アプリケーションノートでは、外部割り込みとタイマ・アレイ・ユニットを使用し、ソフトウェアで UART 通信機能（ソフトウェア UART）を実現する方法を説明します。

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 | 3 |
| 1.1 ソフトウェア UART の通信動作の設定 | 3 |
| 1.2 ボー・レート | 4 |
| 1.3 ソフトウェア UART 受信動作 | 5 |
| 1.4 ソフトウェア UART 送信動作 | 6 |
| 1.5 ソフトウェア UART 送受信時の CPU 使用率 | 6 |
| 2. 動作確認条件 | 7 |
| 3. 関連アプリケーションノート | 7 |
| 4. ハードウェア説明 | 8 |
| 4.1 ハードウェア構成例 | 8 |
| 4.2 使用端子一覧 | 8 |
| 5. ソフトウェア説明 | 9 |
| 5.1 動作概要 | 9 |
| 5.2 オプション・バイトの設定一覧 | 13 |
| 5.3 定数一覧 | 13 |
| 5.4 変数一覧 | 14 |
| 5.5 関数一覧 | 15 |
| 5.6 関数仕様 | 16 |
| 5.7 フローチャート | 19 |
| 5.7.1 初期設定関数 | 19 |
| 5.7.2 周辺機能初期設定関数 | 20 |
| 5.7.3 CPU クロック初期設定 | 21 |
| 5.7.4 入出力ポートの設定 | 22 |
| 5.7.5 タイマ・アレイ・ユニット初期設定関数 | 23 |
| 5.7.6 外部割り込み初期設定関数 | 24 |
| 5.7.7 メイン関数 | 25 |
| 5.7.8 メイン・ユーザー初期設定関数 | 27 |
| 5.7.9 ソフトウェア UART データ送信関数 | 28 |
| 5.7.10 ソフトウェア UART 通信動作設定関数 | 29 |
| 5.7.11 ソフトウェア UART 受信ステータス初期化関数 | 30 |
| 5.7.12 外部割り込み許可関数 | 31 |
| 5.7.13 外部割り込み禁止関数 | 31 |
| 5.7.14 パリティ・チェック関数 | 32 |
| 5.7.15 データ順序並び替え関数 | 33 |
| 5.7.16 外部割り込み処理 | 34 |
| 5.7.17 タイマ・アレイ・ユニット チャンネル 0 割り込み処理 | 35 |
| 5.7.18 タイマ・アレイ・ユニット チャンネル 1 割り込み処理 | 36 |
| 6. サンプルコード | 39 |
| 7. 参考ドキュメント | 39 |

1. 仕様

本アプリケーションノートでは、外部割り込みとタイマ・アレイ・ユニットを使用して UART 通信機能をソフトウェアで実現します。対向機器から送られてくる ASCII 文字を解析し、応答処理を行います。

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 |
|----------------------|---------------------|
| 外部割り込み | 受信端子のスタート・ビット検出 |
| タイマ・アレイ・ユニット チャンネル 0 | 送信時のデータ出力タイミングをカウント |
| タイマ・アレイ・ユニット チャンネル 1 | 受信時のラッチ・タイミングをカウント |

1.1 ソフトウェア UART の通信動作の設定

ソフトウェア UART で選択可能な通信動作を表 1.2 に示します。

ソフトウェア UART の通信動作の設定は、`r_cg_userdefine.h` ファイルの 40 行目～51 行目の `#define` 宣言で行います。`#define` 宣言を有効にすることで設定が反映されます。項目ごとに 1 つのみ有効にしてください。

表 1.2 ソフトウェア UART の通信動作

| 項目 | 内容 | 設定方法 |
|-----------|-----------|-------------------------------------|
| データ転送順序 | MSB ファースト | <code>#define MSB_FIRST</code> 有効 |
| | LSB ファースト | <code>#define LSB_FIRST</code> 有効 |
| パリティ | パリティなし | <code>#define NON_PARITY</code> 有効 |
| | 0 パリティ | <code>#define ZERO_PARITY</code> 有効 |
| | 奇数パリティ | <code>#define ODD_PARITY</code> 有効 |
| | 偶数パリティ | <code>#define EVEN_PARITY</code> 有効 |
| データ出力 | データ反転 | <code>#define INVERT</code> 有効 |
| | データ非反転 | <code>#define NON_INVERT</code> 有効 |
| データ長 | 8 ビット | — |
| スタート・ビット長 | 1 ビット | — |
| ストップ・ビット長 | 1 ビット | — |

1.2 ボー・レート

ソフトウェア UART では、受信データのラッチ・タイミングと送信データの出カタイミングをタイマ・アレ・イ・ユニットのインターバル・タイマを使用してカウントします。下記の式からボー・レートを反映させたインターバル時間の設定値 (TDRmn レジスタ) を算出してください。インターバル時間の設定値 (TDRmn レジスタ) は r_cg_userdefine.h ファイルの #define BAUD_RATE に設定してください。

インターバル時間の設定値 (TDRmn レジスタ) = (CPU クロック周波数 / ボー・レート)

また、受信時のスタート・ビットは、外部割り込みの有効エッジの検出からボー・レート/2 の時間が経過したタイミングでラッチします。このため、スタート・ビットのラッチ・タイミングのみインターバル時間が異なります。なお、割り込み処理内でラッチすることから、ラッチ・タイミングに遅延が発生します。本アプリケーションノートでは、ラッチ・タイミングの遅延を調整するため、下記の式のように 67 クロックを引いた値をインターバル時間の設定値 (TDRmn レジスタ) に設定します。インターバル時間の設定値 (TDRmn レジスタ) は r_cg_userdefine.h ファイルの #define BAUD_RATE_START_BIT に設定してください。

インターバル時間の設定値 (TDRmn) = (CPU クロック周波数 / (ボー・レート / 2)) - 67

1.3 ソフトウェア UART 受信動作

ソフトウェア UART の受信動作を説明します。

受信動作は外部割り込みの有効エッジの検出によって開始します。受信端子のラッチ・タイミングにはタイマ・アレイ・ユニットのインターバル・タイマを使用します。ラッチ・タイミングは、入力レベルの変化からボー・レート/2の時間が経過したタイミングになります。ラッチした値は受信データ変数に格納してシフト動作を行います。

ストップ・ビットのラッチが完了するとフレーミング・エラーとパリティ・エラーの判定を行います。エラーが発生していなければ、ソフトウェア UART の通信動作に合わせて受信データを加工し、8ビットの受信データを作成します。8ビットの受信データを RAM に格納して1データの受信動作が完了します。

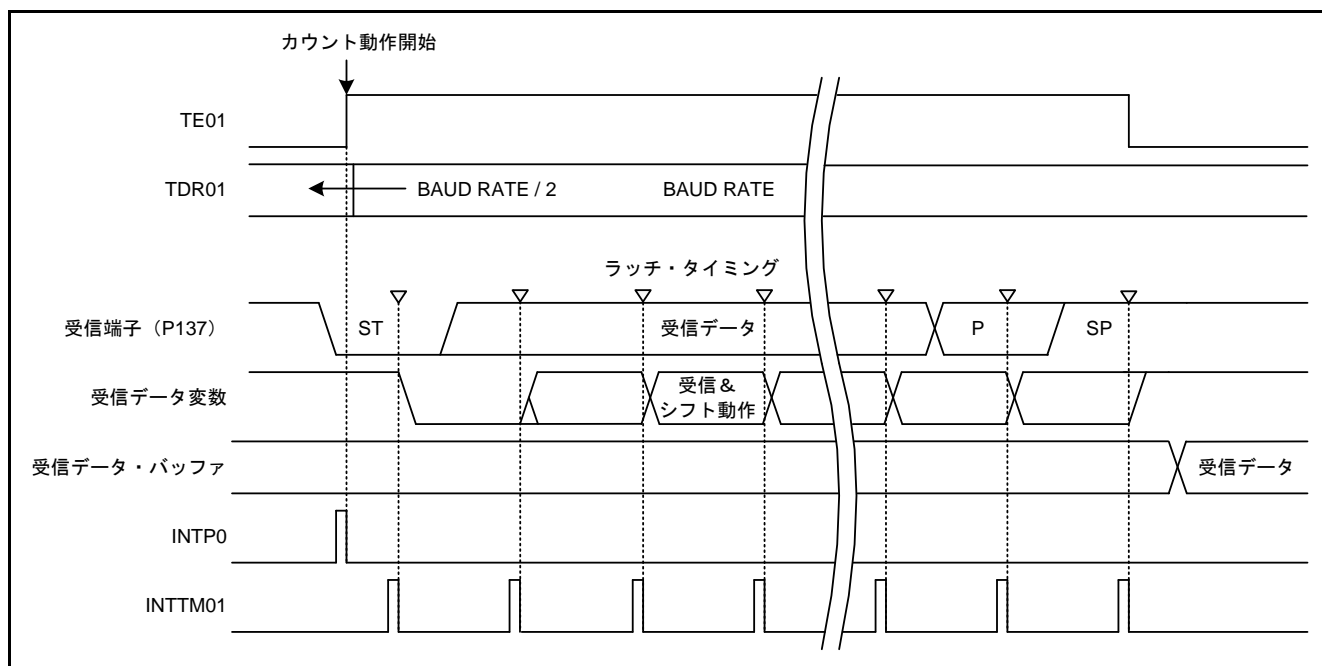


図 1.1 ソフトウェア UART 受信のタイミング・チャート

1.4 ソフトウェア UART 送信動作

ソフトウェア UART の送信動作を説明します。

ソフトウェア UART の通信動作（データ転送順序、パリティ・ビット、データ出力）に合わせ、最下位ビットから順に「スタート・ビット、データ、パリティ・ビット、ストップ・ビット」となる 16 ビットの送信データを作成します。

データの出カタイミングはタイマ・アレイ・ユニットのインターバル・タイマでカウントします。割り込み発生ごとに最下位ビットから順に P130 へ出力します。ストップ・ビットの出力が完了すると 1 データの送信完了となります。

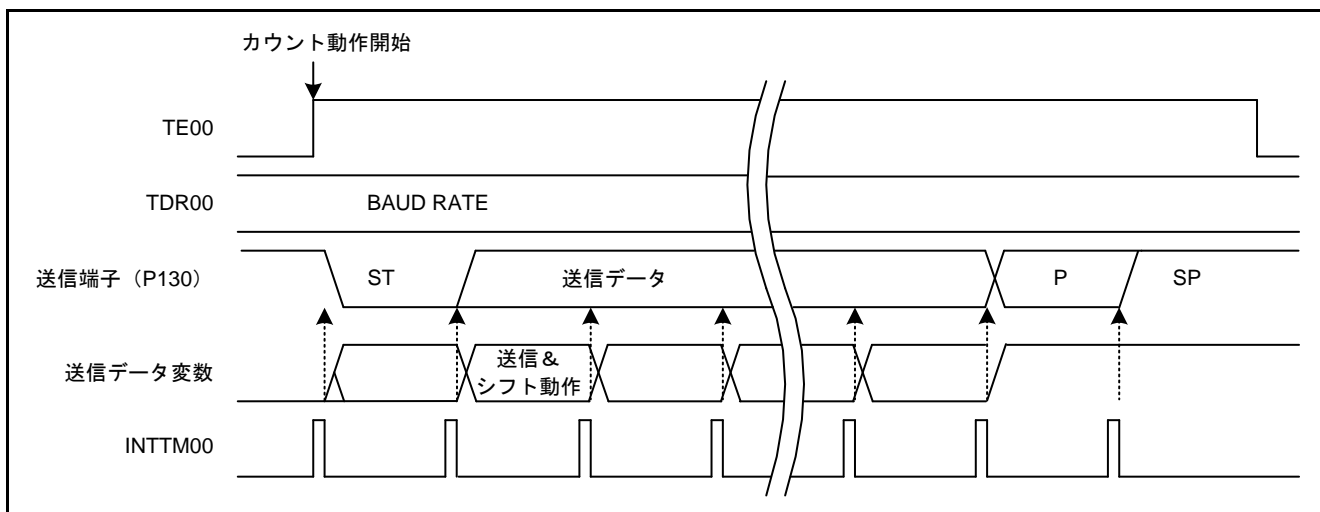


図 1.2 ソフトウェア UART 送信のタイミング・チャート

1.5 ソフトウェア UART 送受信時の CPU 使用率

ソフトウェア UART では、送受信時に CPU を使用します。表 1.3 に送受信時の CPU 使用率を示します。CPU クロックやボー・レートを変更して使用する際は、十分に評価を実施してください。

表 1.3 CPU 使用率

| CPU クロック | ボー・レート | CPU 使用率 (参考値) |
|----------|----------|---------------------|
| 32MHz | 9600bps | 送信 : 約 1%、受信 : 約 1% |
| 32MHz | 19200bps | 送信 : 約 2%、受信 : 約 2% |
| 16MHz | 19200bps | 送信 : 約 4%、受信 : 約 4% |

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 (R5F100LE) |
| 動作周波数 | <ul style="list-style-type: none"> ● 高速オンチップ・オシレータ・クロック (f_{IH}) : 32MHz ● CPU/周辺ハードウェア・クロック : 32MHz |
| 動作電圧 | 3.3V (1.8V~5.5V で動作可能) LVD 動作 (V_{LVD}) : リセット・モード 2.75V (2.70V to 2.81V) |
| 統合開発環境 (CS+) | ルネサス エレクトロニクス製 CS+ for CC V8.01.00 |
| C コンパイラ (CS+) | ルネサス エレクトロニクス製 CC-RL V1.08.00 |
| 統合開発環境 (e ² studio) | ルネサス エレクトロニクス製 e ² studio V7.3.0 |
| C コンパイラ (e ² studio) | ルネサス エレクトロニクス製 CC-RL V1.08.00 |

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 CC-RL (R01AN2575JJ) アプリケーションノート

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

本アプリケーションノートで使用するハードウェアを図 4.1 に示します。

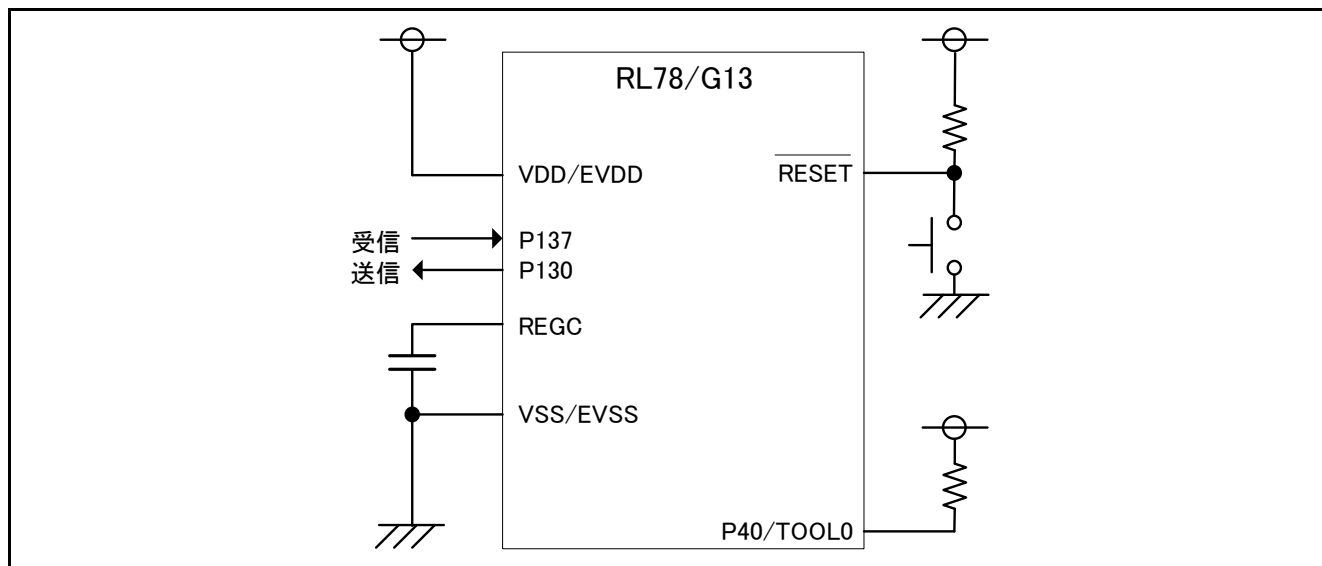


図 4.1 ハードウェア構成例

注意 1 この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

(入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続して下さい)。

4.2 使用端子一覧

使用端子と機能を表 4.1 に示します。

表 4.1 使用端子と機能

| 端子名 | 入出力 | 機能 |
|------------|-----|--------|
| P130 | 出力 | データを出力 |
| P137/INTPO | 入力 | データを入力 |

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、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。通信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) |

詳細は下記①～③に記載します。

① CPU クロックの初期設定を行います。

<CPU クロック設定条件>

- CPU/周辺ハードウェア・クロック (f_{CLK}) に高速オンチップ・オシレータ・クロック 32MHz を設定します。
- 高速システム・クロック端子の動作モードを入力ポート・モードに設定します。
- サブシステム・クロック端子の動作モードを入力ポート・モードに設定します。

② 入出力ポートの初期設定を行います。

<入出力ポート設定条件>

- P137 を入力モードに設定します。
- P130 を出力モードに設定します。

③ タイマ・アレイ・ユニットの初期設定を行います。

<チャンネル 0 の設定条件>

- タイマ動作モードはインターバル・タイマ・モードに設定します。
- タイマ・データ・レジスタ 00 (TDR00) は初期値 104us に設定します。
- タイマ出力許可レジスタは動作禁止に設定します。
- タイマ・チャンネル 0 のタイマ割り込み (INTTM00) を使用します。

<チャンネル 1 の設定条件>

- タイマ動作モードはインターバル・タイマ・モードに設定します。
- タイマ・データ・レジスタ 01 (TDR01) は初期値 104us に設定します。
- タイマ出力許可レジスタは動作禁止に設定します。
- タイマ・チャンネル 1 のタイマ割り込み (INTTM01) を使用します。

④ 外部割り込みの初期設定を行います。

<外部割り込み設定条件>

- INTP0 端子の有効エッジを設定します。

#define NON_INVERT 有効時 : 立ち上がりエッジ

#define INVERT 有効時 : 立ち下がりエッジ

⑤ 送信端子 (P130) の出力レベルの初期値を設定します。

#define NON_INVERT 有効時 : ハイ・レベル出力

#define INVERT 有効時 : ロウ・レベル出力

⑥ 受信ステータスの初期化とポー・レートを設定して、外部割り込みを許可します。

⑦ HALT モードへ移行して、データの受信を待ちます。

⑧ 外部割り込み発生により、データの受信処理を行います。受信処理完了後、外部割り込みを禁止に設定します。

⑨ 受信データに対応した下記データの送信処理を行います。

- “T”を受信 : “OK”を送信
- “t”を受信 : “ok”を送信
- “T”と“t”以外を受信 : “UC”を送信
- フレーミング・エラー発生 : “FE”を送信
- パリティ・エラー発生 : “PE”を送信

⑩ 送信処理の完了を待ちます。

⑪ 受信ステータスを初期化し、外部割り込みを許可します。

⑫ ⑦から⑪を繰り返し実行します。

データ受信処理

- ① 外部割り込み発生によって HALT モードから復帰し、タイマ・アレイ・ユニットのインターバル・タイマのカウンタ動作を開始します。その後、インターバル時間の設定を「ボー・レート/2」から「ボー・レート」に変更します。
- ② インターバル時間（ボー・レート/2）経過後にインターバル・タイマの割り込みが発生します。P137 をラッチしてスタート・ビットの値を受信データ変数に格納します。受信データ変数を 1 ビット左シフトします。
- ③ HALT モードへ移行してインターバル・タイマの割り込みを待ちます。
- ④ インターバル時間（ボー・レート）経過後にインターバル・タイマの割り込みが発生します。P137 をラッチしてデータの値を受信データ変数に格納します。受信データ変数を 1 ビット左シフトします。
- ⑤ ストップ・ビットを受信するまで③と④を繰り返し実行します。
- ⑥ 全ビットの受信が完了後、インターバル・タイマのカウンタ動作を停止します。
- ⑦ #define INVERT 有効時、受信データ変数の値を反転します。
- ⑧ フレーミング・エラーの発生を確認します。フレーミング・エラーが発生していた場合は受信エラー・ステータスにフレーミング・エラーを設定します。
- ⑨ #define EVEN_PARITY または #define ODD_PARITY 有効時、パリティ・エラーの発生を確認します。パリティ・エラーが発生していた場合は受信エラー・ステータスにパリティ・エラーを設定します。
- ⑩ ⑧と⑨で受信エラーが発生していなかった場合、受信データ変数からスタート・ビット、パリティ・ビット、ストップ・ビットを省いた 8 ビットのデータに加工して受信データ・バッファに格納します。#define LSB_FIRST 有効時は 8 ビットのデータのビットの順序を並び替えます。
- ⑪ 受信データ・ポインタと受信データ数カウンタを更新します。
- ⑫ 外部割り込みを許可に設定して、インターバル時間を「ボー・レート / 2」に設定します。その後、HALT モードへ移行します。
- ⑬ 受信データ数変数に設定したデータ数を受信するまで①～⑫を繰り返し実行します。受信が完了すると受信完了フラグをセットし、受信動作を終了します。

データ送信処理

- ① RAM から 8 ビットの送信データを読み出して送信データ・バッファに格納します。
- ② 送信データ変数にストップ・ビットとパリティ・ビットを設定します。パリティ・ビットは#define ZERO_PARITY、#define EVEN_PARITY、#define ODD_PARITY 有効時に設定します。
- ③ #define MSB_FIRST 有効時、送信データ・バッファのビットの順序を並び替えます。
- ④ 送信データ変数に送信データ・バッファの値を加算して、最下位ビットから順に「スタート・ビット、データ、パリティ・ビット、ストップ・ビット」の構成にします。
- ⑤ #define INVERT 有効時、送信データ変数の値を反転します。
- ⑥ タイマ・アレイ・ユニットのインターバル・タイマのカウント動作を開始します。カウント動作開始と同時に割り込みが発生し、P130 に送信データ変数の最下位ビットの値を設定してデータを出力します。送信データ変数を 1 ビット右シフトします。
- ⑦ HALT モードへ移行してインターバル・タイマの割り込みを待ちます。
- ⑧ インターバル時間経過後にインターバル・タイマの割り込みが発生します。P130 に送信データ変数の最下位ビットの値を設定してデータを出力します。送信データ変数を 1 ビット右シフトします。
- ⑨ ストップ・ビットを送信するまで⑦と⑧を繰り返し実行します。
- ⑩ 全ビットの送信完了後、インターバル・タイマのカウント動作を停止します。
- ⑪ 多重割り込みを許可します。
- ⑫ 送信データ・ポインタと送信データ数カウンタを更新します。
- ⑬ ①～⑤の内容を実行します。
- ⑭ インターバル・タイマのカウント動作を開始します。前回送信時のストップ・ビットの期間を確保するため、カウント動作開始時には割り込みを発生させません。
- ⑮ 送信データ数変数に設定したデータ数を送信するまで⑦～⑭を繰り返し実行します。送信が完了すると送信完了フラグをセットし、送信動作を終了します。

5.2 オプション・バイトの設定一覧

表 5.3 に、オプション・バイト設定を示します。

表 5.3 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 TYP. 2.75V 電源立ち上がり時 2.76V~2.87V 電源立ち下がり時 2.70V~2.81V |
| 000C2H/010C2H | 1110 1000B | HS モード、高速オンチップ・オシレータ・ クロック : 32MHz |
| 000C3H/010C3H | 1000 0100B | オンチップ・デバッグ許可 |

5.3 定数一覧

表 5.4 にサンプルコードで使用する定数を示します。

表 5.4 サンプルコードで使用する定数

| 定数名 | 設定値 | 内容 |
|---------------------|-----------|-----------------------|
| LENGTH_8BIT | 0x08 | データのビット数 |
| FRAMING_ERROR | 0x01 | フレーミング・エラー |
| PARITY_ERROR | 0x02 | パリティ・エラー |
| BAUD_RATE_START_BIT | 0x0047 | ラッチ・タイミングの設定値 |
| BAUD_RATE | 0x0115 | ボー・レートの設定値 (9600bps) |
| BIT_TOTAL | 0x10、0x11 | 総ビット数 |
| OUTPUT_INIT | 0x00、0x01 | 送信端子の初期値 |
| ACTIVE_LEVEL | 0x80、0x00 | 受信端子の有効値 |
| g_messageOK[4] | "OK¥r¥n" | "T"を受信時の返信メッセージ |
| g_messageok[4] | "ok¥r¥n" | "t"を受信時の返信メッセージ |
| g_messageUC[4] | "UC¥r¥n" | "T"と"t"以外を受信時の返信メッセージ |
| g_messageFE[4] | "FE¥r¥n" | フレーミング・エラー時の返信メッセージ |
| g_messagePE[4] | "PE¥r¥n" | パリティ・エラー時の返信メッセージ |

5.4 変数一覧

表 5.5 にグローバル変数を示します。

表 5.5 グローバル変数

| 型 | 変数名 | 内容 | 使用関数 |
|----------|---------------------|-------------|---|
| uint8_t | gp_uarte_tx_address | 送信データ・ポインタ | r_uarte_send()、 r_tau0_channel0_interrupt() |
| uint8_t | gp_uarte_rx_address | 受信データ・ポインタ | r_uarte_receive()、 r_tau0_channel1_interrupt() |
| uint8_t | g_uarte_tx_buffer | 送信データ・バッファ | r_uarte_send()、 r_tau0_channel0_interrupt()、 r_uarte_tx_set() |
| uint8_t | g_uarte_rx_buffer | 受信データ・バッファ | main()、R_MAIN_UserInit()、 r_tau0_channel1_interrupt() |
| uint16_t | g_uarte_tx_count | 送信データ数カウンタ | r_uarte_send()、 r_tau0_channel0_interrupt |
| uint16_t | g_uarte_rx_count | 受信データ数カウンタ | r_uarte_receive()、 r_tau0_channel1_interrupt() |
| uint16_t | g_uarte_rx_length | 受信データ数 | r_uarte_receive()、 r_tau0_channel1_interrupt() |
| uint16_t | g_uarte_tx_data | 送信データ | r_tau0_channel0_interrupt()、 r_uarte_tx_set() |
| uint16_t | g_uarte_rx_data | 受信データ | r_tau0_channel1_interrupt() |
| uint8_t | g_uarte_rx_error | 受信エラー・ステータス | main()、r_uarte_receive()、 r_tau0_channel1_interrupt() |
| uint8_t | g_uarte_rx_end | 受信完了フラグ | main()、 r_tau0_channel1_interrupt() |
| uint8_t | g_uarte_tx_end | 送信完了フラグ | main()、r_uarte_send()、 r_tau0_channel0_interrupt() |
| uint8_t | g_tx_bit_count | 送信ビット・カウンタ | r_uarte_send()、 r_tau0_channel0_interrupt() |
| uint8_t | g_rx_bit_count | 受信ビット・カウンタ | r_tau0_channel1_interrupt() |
| uint16_t | g_check | パリティ・ビット比較値 | r_tau0_channel1_interrupt() |
| uint16_t | g_parity | パリティ・ビット受信値 | r_tau0_channel1_interrupt() |
| uint8_t | g_port | ポートからの読み出し値 | r_tau0_channel1_interrupt() |

5.5 関数一覧

表 5.6 に関数一覧を示します。

表 5.6 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_INTC0_Start | 外部割り込み許可関数 |
| R_INTC0_Stop | 外部割り込み禁止関数 |
| r_uarts_send | ソフトウェア UART データ送信関数 |
| r_uarts_receive | ソフトウェア UART 受信ステータス初期化関数 |
| r_parity_check | パリティ・チェック関数 |
| r_bit_swap_8bit | データ順序並び替え関数 |
| r_uarts_tx_set | ソフトウェア UART 通信動作設定関数 |
| r_intc0_interrupt | 外部割り込み処理 |
| r_tau0_channel0_interrupt | タイマ・アレイ・ユニット チャンネル 0 割り込み処理 |
| r_tau0_channel1_interrupt | タイマ・アレイ・ユニット チャンネル 1 割り込み処理 |

5.6 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_intc.h、r_cg_timer.h、 r_cg_userdefine.h |
| 宣言 | — |
| 説明 | メイン・ユーザー初期化関数を実行後、HALT モードに移行してデータの受信を待ちます。1 バイトのデータを受信した後、受信したデータに対応したデータを送信します。 |
| 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_intc.h、r_cg_timer.h、 r_cg_userdefine.h |
| 宣言 | void R_MAIN_UserInit(void); |
| 説明 | P130 に出力の初期値を設定します。次に、ソフトウェア UART のステータスを初期化します。外部割り込みを許可に設定し、マスクブル割り込みを許可します。 |
| 引数 | なし |
| リターン値 | なし |
| [関数名] R_INTC0_Start | |
| 概要 | 外部割り込み許可関数 |
| ヘッダ | r_cg_macrodriver.h、r_cg_intc.h、r_cg_userdefine.h |
| 宣言 | void R_INTC0_Start(void); |
| 説明 | 外部割り込み処理許可に設定します。 |
| 引数 | なし |
| リターン値 | なし |
| [関数名] R_INTC0_Stop | |
| 概要 | 外部割り込み許可関数 |
| ヘッダ | r_cg_macrodriver.h、r_cg_intc.h、r_cg_userdefine.h |
| 宣言 | void R_INTC0_Stop(void); |
| 説明 | 外部割り込み処理禁止に設定します。 |
| 引数 | なし |
| リターン値 | なし |

| | |
|-----------------------|--|
| [関数名] r_arts_send | |
| 概要 | ソフトウェア UART データ送信処理 |
| 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_userdefine.h |
| 宣言 | void r_arts_send(uint8_t * tx_buf, uint16_t tx_num); |
| 説明 | ソフトウェア UART の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 * 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 | なし |
| [関数名] r_arts_receive | |
| 概要 | ソフトウェア UART データ受信処理 |
| 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_userdefine.h |
| 宣言 | void r_arts_receive(uint8_t * rx_buf, uint16_t rx_num); |
| 説明 | ソフトウェア UART の受信の初期設定を行います。 |
| 引数 | uint8_t * 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 | なし |
| [関数名] r_parity_check | |
| 概要 | パリティ・チェック関数 |
| 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_userdefine.h |
| 宣言 | uint16_t r_parity_check(uint16_t data); |
| 説明 | データのパリティ値を求めます。 |
| 引数 | uint16_t data :送受信データ |
| リターン値 | uint16_t :パリティ値 |
| [関数名] r_bit_swap_8bit | |
| 概要 | データ順序並び替え関数 |
| 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_userdefine.h |
| 宣言 | uint8_t r_bit_swap_8bit(uint8_t data); |
| 説明 | データのビットの順序を並び替えます。 |
| 引数 | uint8_t data :送受信データ |
| リターン値 | uint8_t :順序を並び替えたデータ |
| [関数名] r_arts_tx_set | |
| 概要 | ソフトウェア UART 通信動作設定関数 |
| 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_userdefine.h |
| 宣言 | void r_arts_tx_set(void); |
| 説明 | ソフトウェア UART の通信動作を設定します。 |
| 引数 | なし |
| リターン値 | なし |

| | |
|-------------------------|---|
| [関数名] r_intc0_interrupt | |
| 概要 | 外部割り込み処理 |
| ヘッダ | r_cg_macrodriver.h、r_cg_intc.h、r_cg_timer.h、r_cg_userdefine.h |
| 宣言 | #pragma interrupt r_intc0_interrupt(vect=INTP0) |
| 説明 | タイマ・アレイ・ユニット チャンネル 1 のカウントを開始します。外部割り込み禁止に設定します。 |
| 引数 | なし |
| リターン値 | なし |

| | |
|---------------------------------|---|
| [関数名] r_tau0_channel0_interrupt | |
| 概要 | タイマ・アレイ・ユニット チャンネル 0 割り込み処理 |
| ヘッダ | r_cg_macrodriver.h、r_cg_intc.h、r_cg_timer.h、r_cg_userdefine.h |
| 宣言 | #pragma interrupt r_tau0_channel0_interrupt(vect=INTTM00) |
| 説明 | 送信データを P13 レジスタに設定します。1 データ分の送信が完了した場合は、次のデータの送信を開始します。 |
| 引数 | なし |
| リターン値 | なし |

| | |
|---------------------------------|--|
| [関数名] r_tau0_channel1_interrupt | |
| 概要 | タイマ・アレイ・ユニット チャンネル 1 割り込み処理 |
| ヘッダ | r_cg_macrodriver.h、r_cg_intc.h、r_cg_timer.h、r_cg_userdefine.h |
| 宣言 | #pragma interrupt r_tau0_channel1_interrupt(vect=INTTM01,bank=RB2,enable=true) |
| 説明 | 入力レベルを P13 レジスタから読み出します。ストップ・ビットまで受信した場合はフレーミング・エラーとパリティ・エラーの発生を確認します。エラーが発生していない場合は、受信データを受信データ・バッファへ格納します。 |
| 引数 | なし |
| リターン値 | なし |

5.7 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。

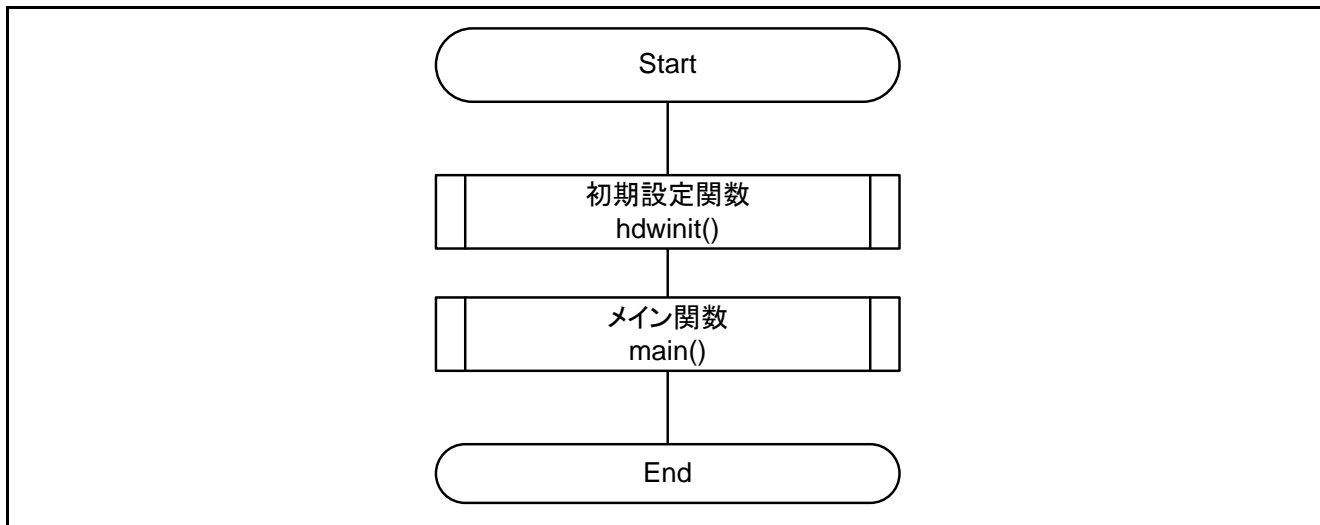


図 5.1 全体フロー

5.7.1 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。

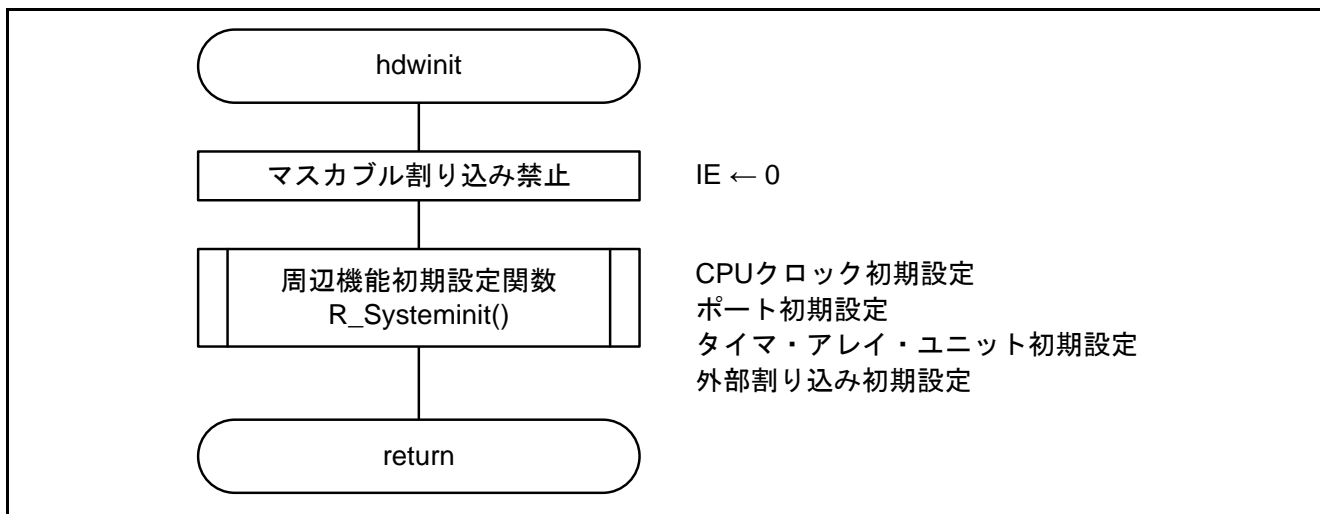


図 5.2 初期設定関数

5.7.2 周辺機能初期設定関数

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。

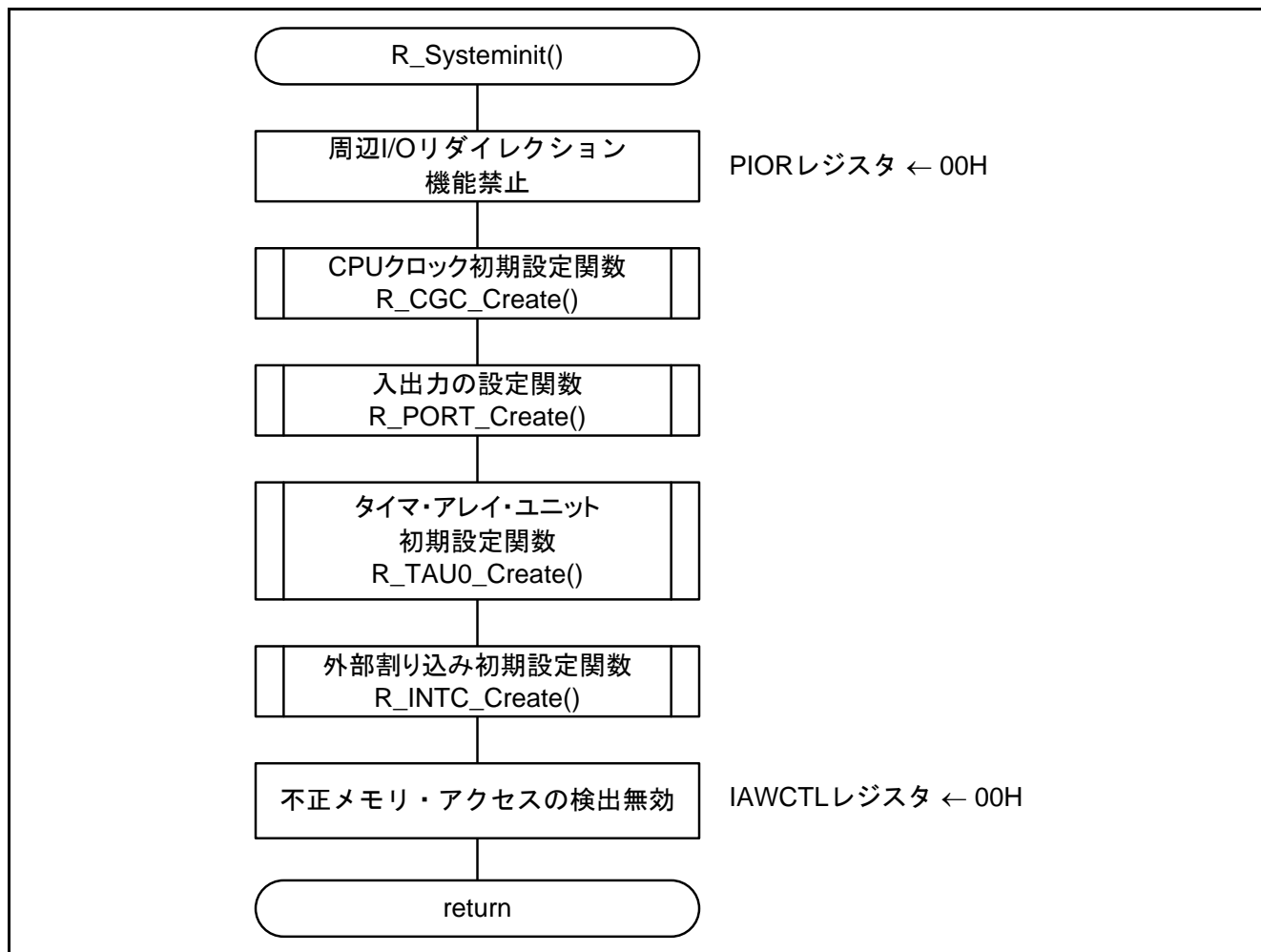


図 5.3 周辺機能初期設定関数

5.7.3 CPU クロック初期設定

図 5.4 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。

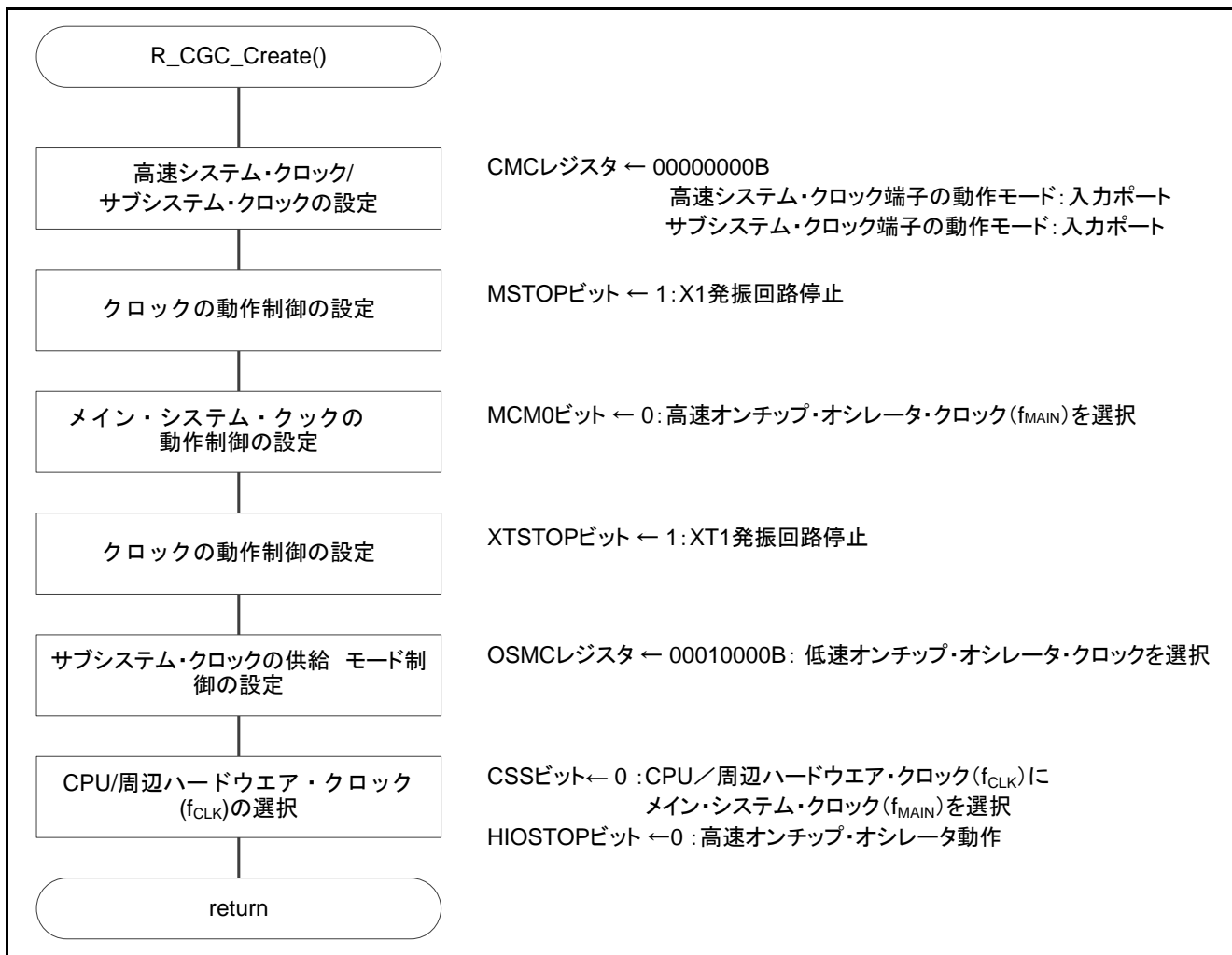


図 5.4 CPU クロック初期設定関数

5.7.4 入出力ポートの設定

図 5.5 に入出力ポートのフローチャートを示します。

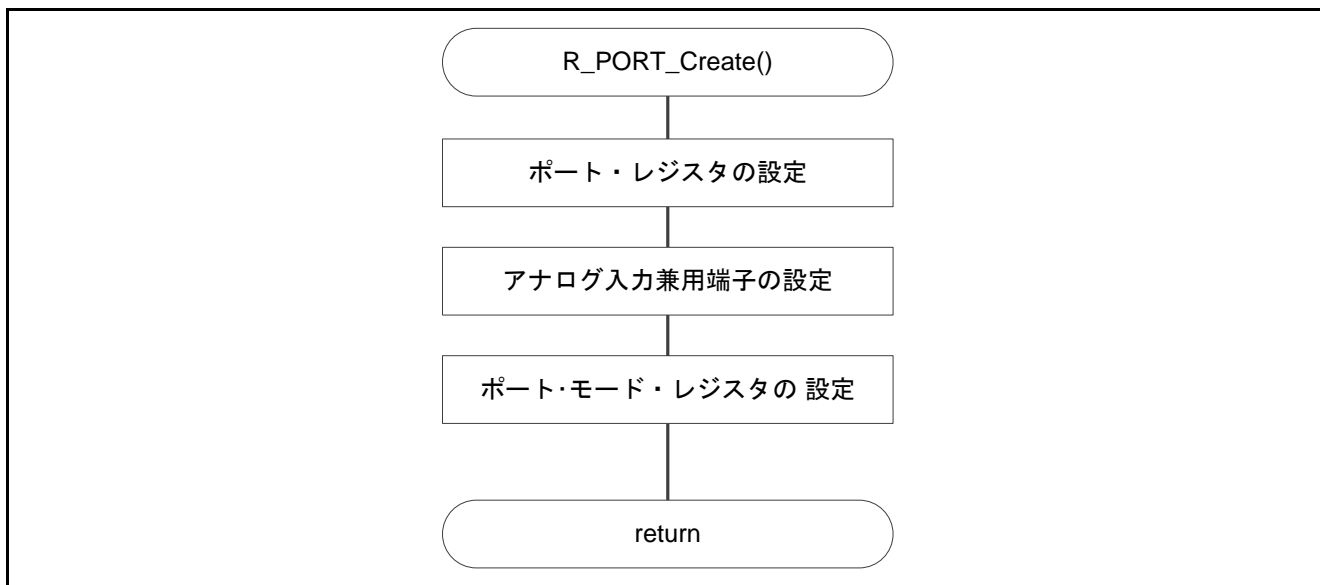


図 5.5 入出力ポートの設定

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。

5.7.5 タイマ・アレイ・ユニット初期設定関数

図 5.6 にタイマ・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。

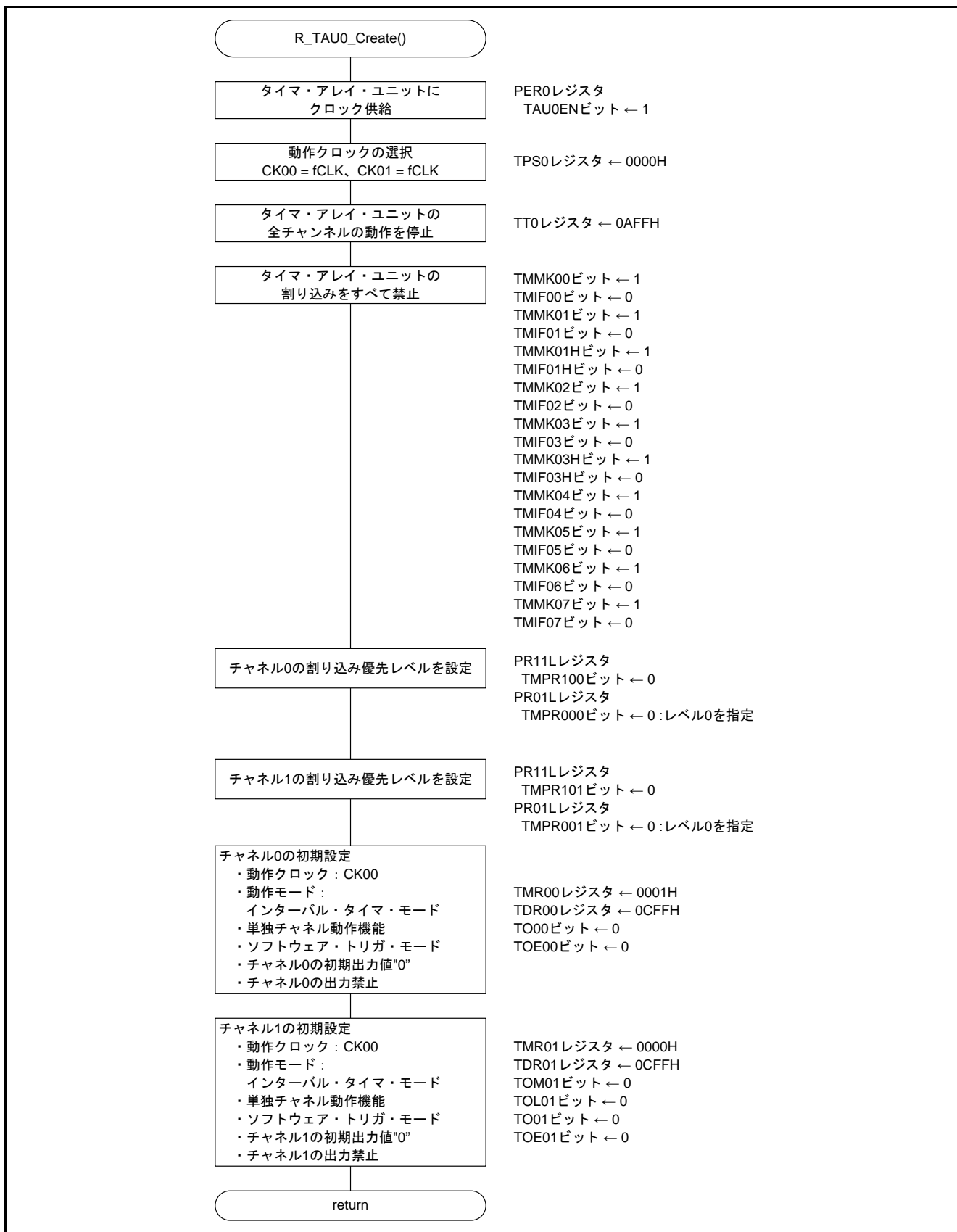


図 5.6 タイマ・アレイ・ユニット初期設定関数

5.7.6 外部割り込み初期設定関数

図 5.7 に外部割り込み初期設定関数のフローチャートを示します。

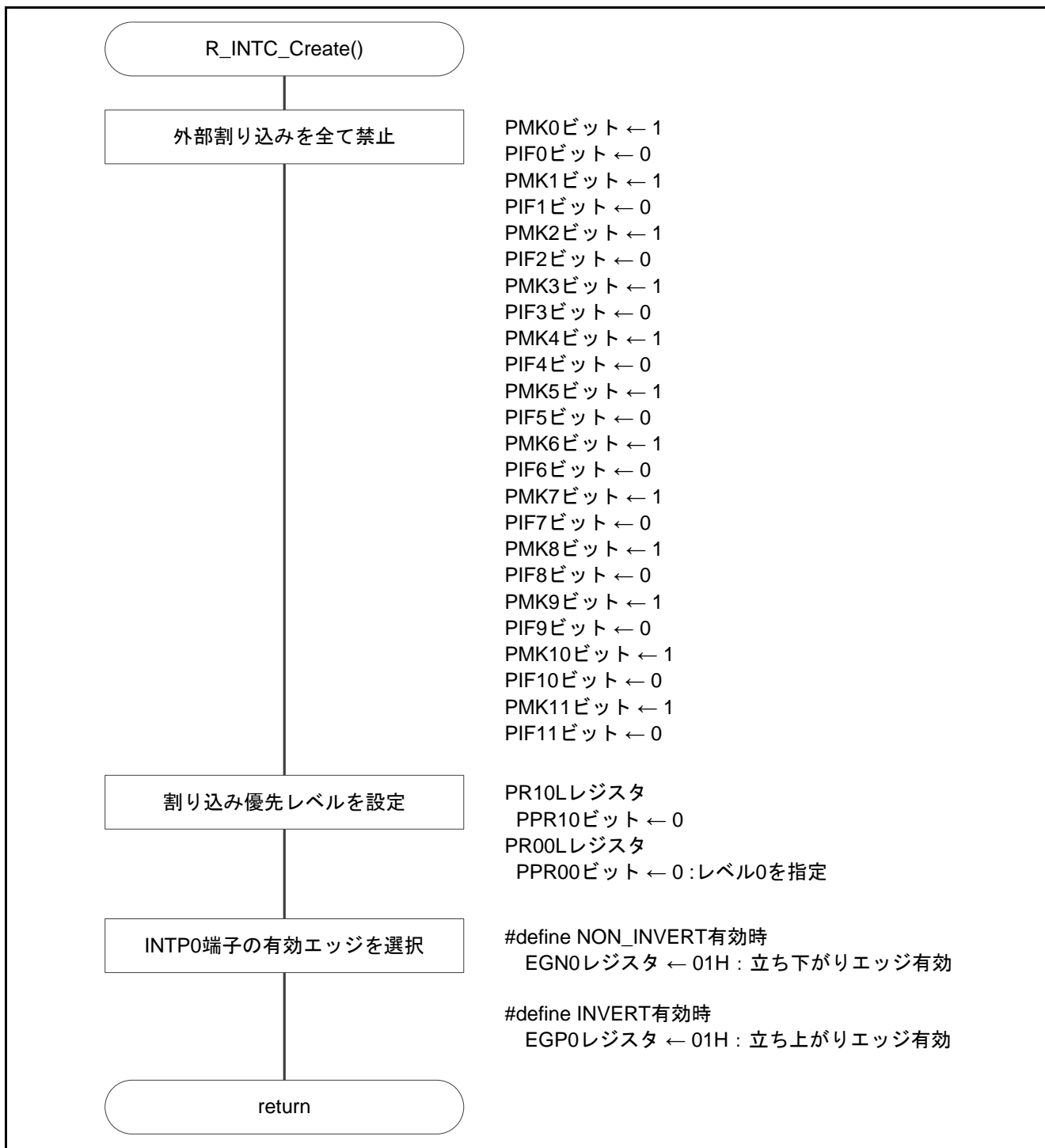


図 5.7 外部割り込み初期設定関数

5.7.7 メイン関数

図 5.8、図 5.9 にメイン関数のフローチャートを示します。

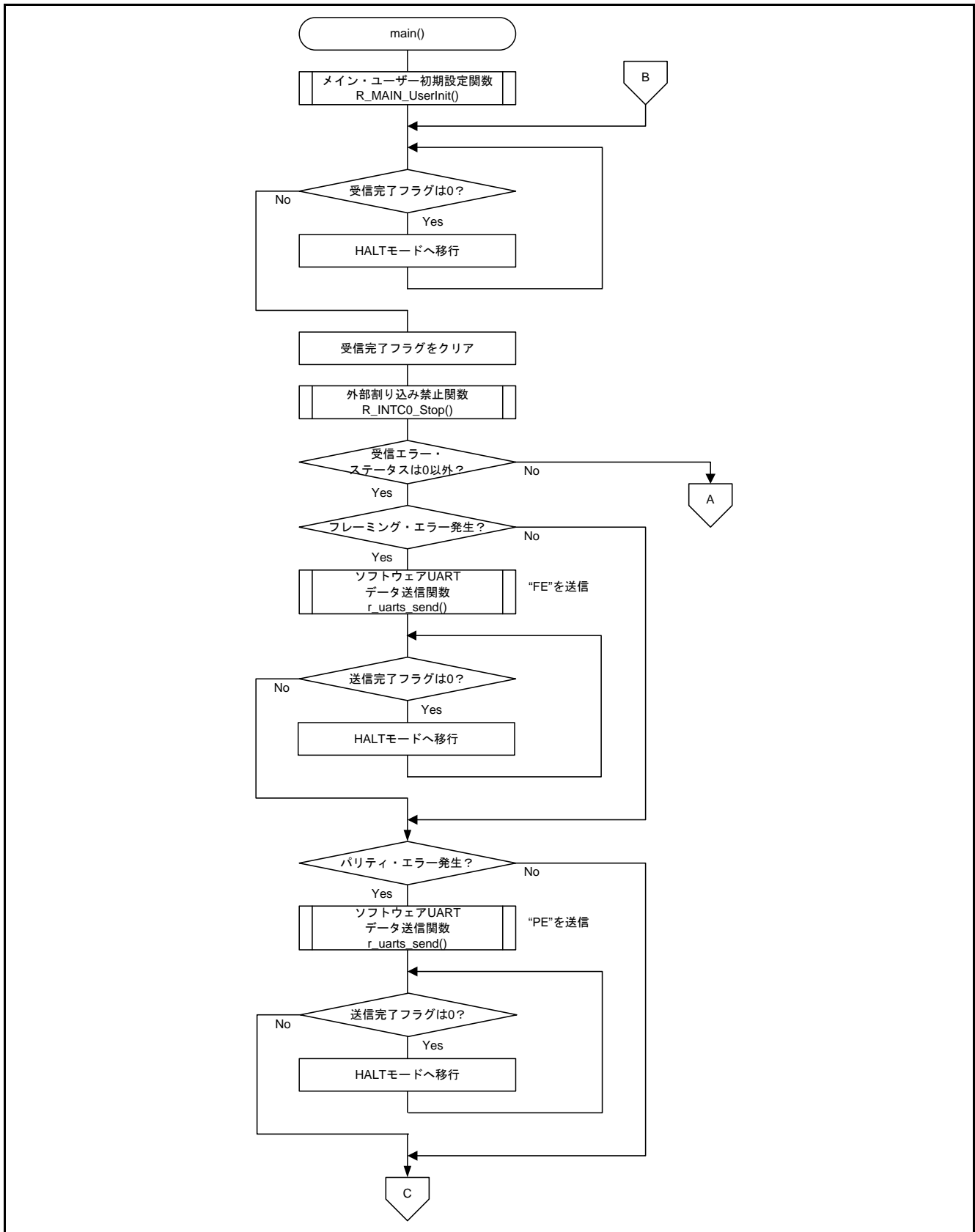


図 5.8 メイン関数 (1/2)

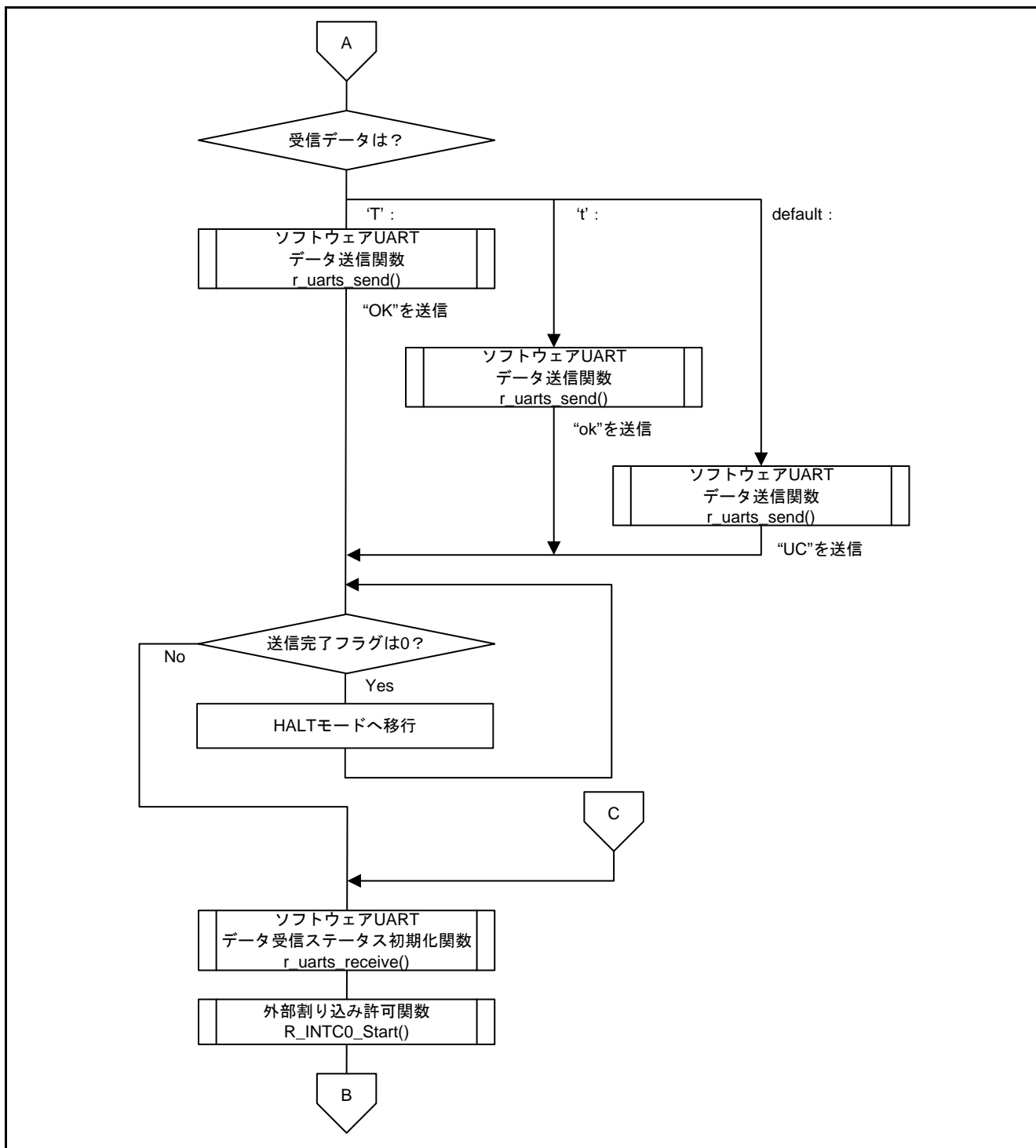


図 5.9 メイン関数 (2/2)

5.7.8 メイン・ユーザー初期設定関数

図 5.10 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。

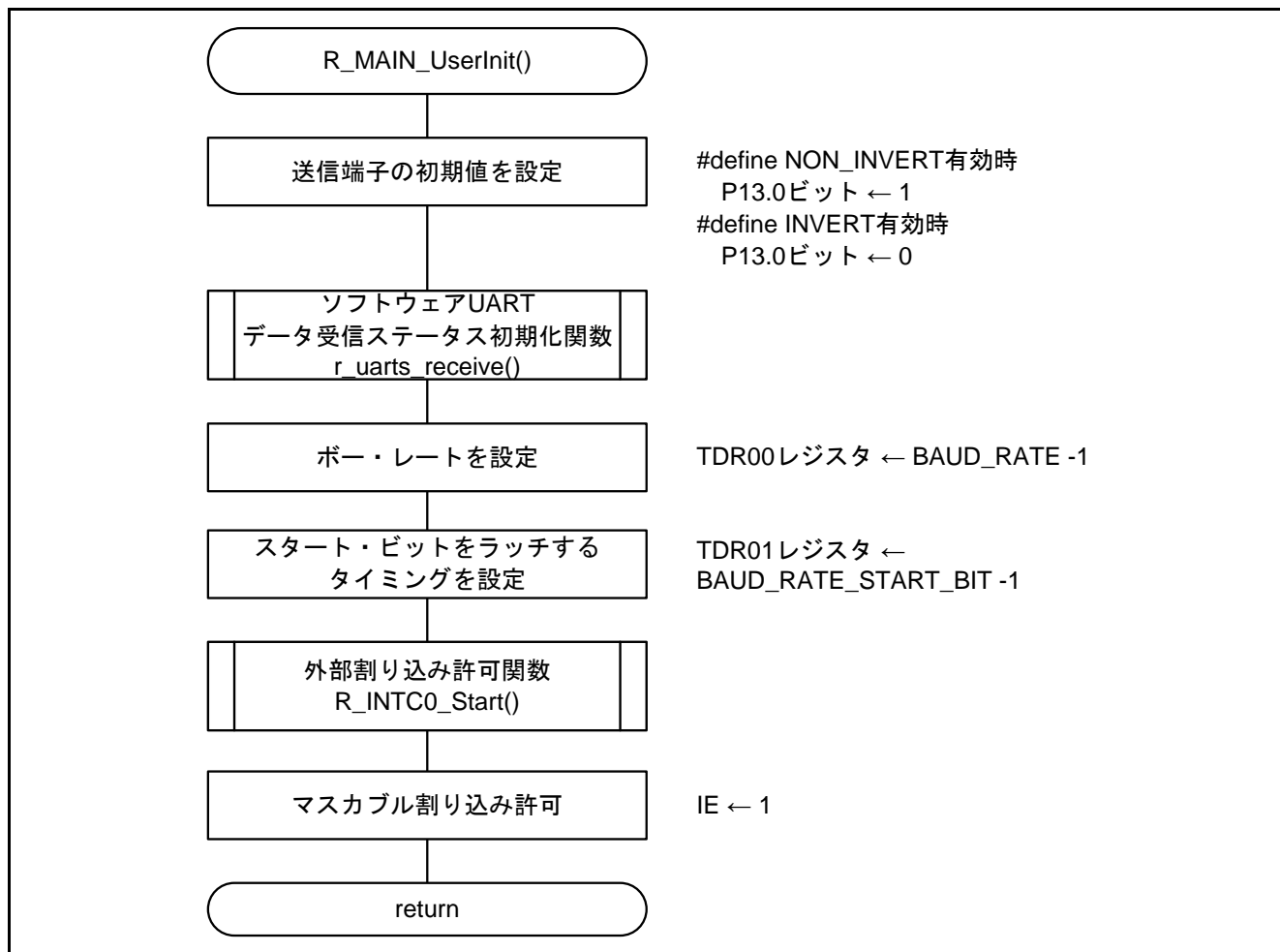


図 5.10 メイン・ユーザー初期設定関数

5.7.9 ソフトウェア UART データ送信関数

図 5.11 にソフトウェア UART データ送信関数のフローチャートを示します。

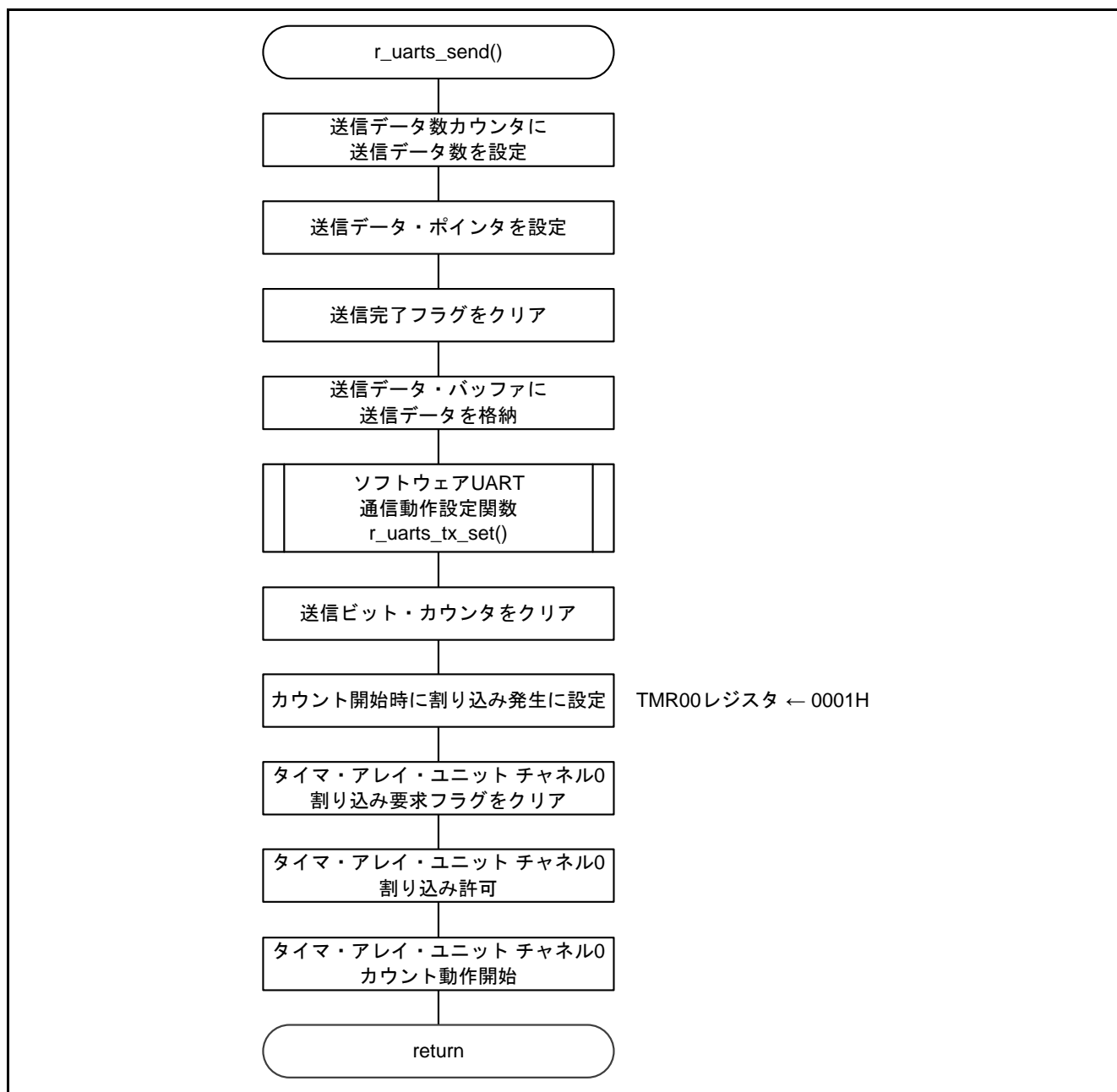


図 5.11 ソフトウェア UART データ送信関数

5.7.10 ソフトウェア UART 通信動作設定関数

図 5.12 にソフトウェア UART 通信動作設定関数のフローチャートを示します。

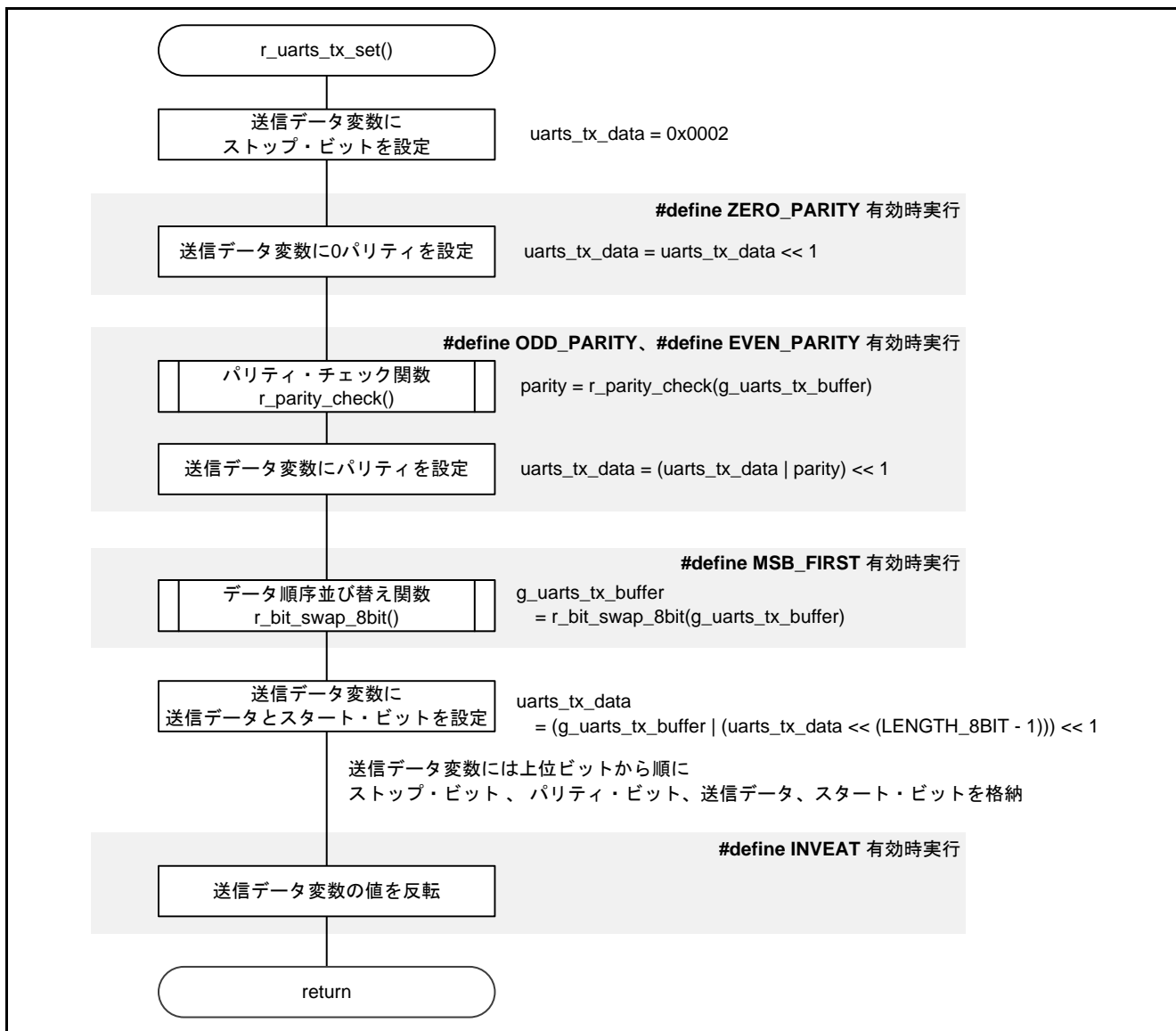


図 5.12 ソフトウェア UART 通信動作設定関数

5.7.11 ソフトウェア UART 受信ステータス初期化関数

図 5.13 にソフトウェア UART 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。

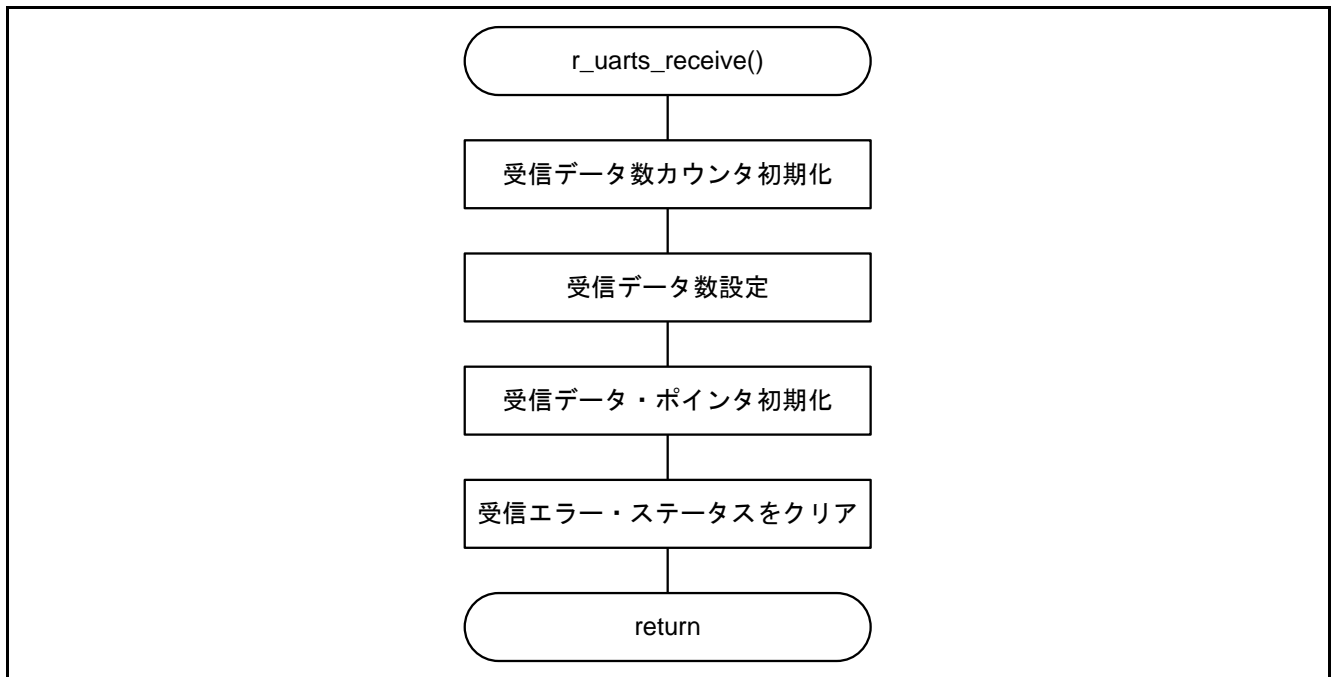


図 5.13 ソフトウェア UART 受信ステータス初期化関数

5.7.12 外部割り込み許可関数

図 5.14 に外部割り込み許可関数のフローチャートを示します。

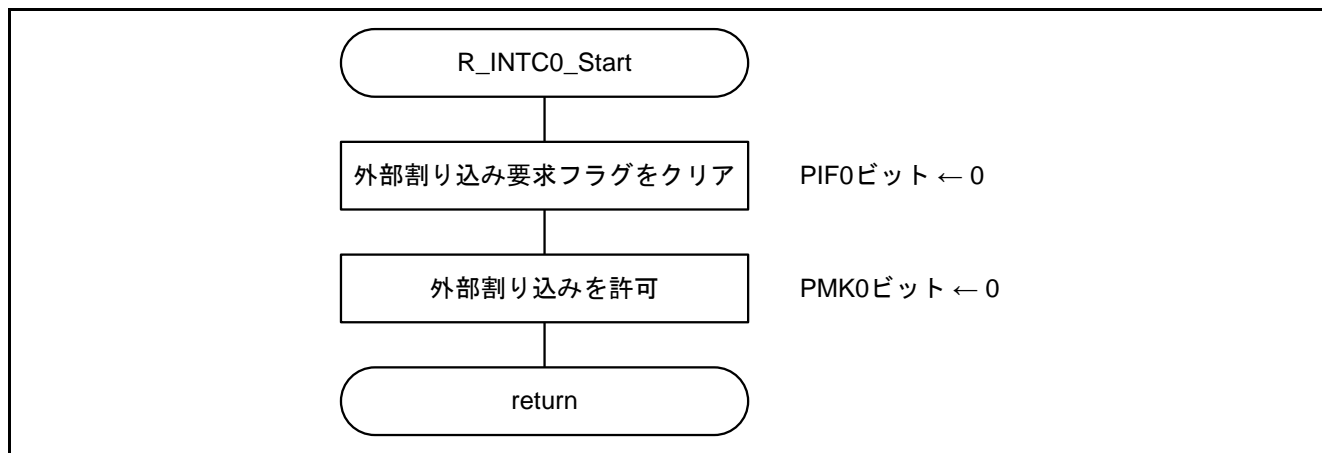


図 5.14 外部割り込み許可関数

5.7.13 外部割り込み禁止関数

図 5.15 に外部割り込み禁止関数のフローチャートを示します。

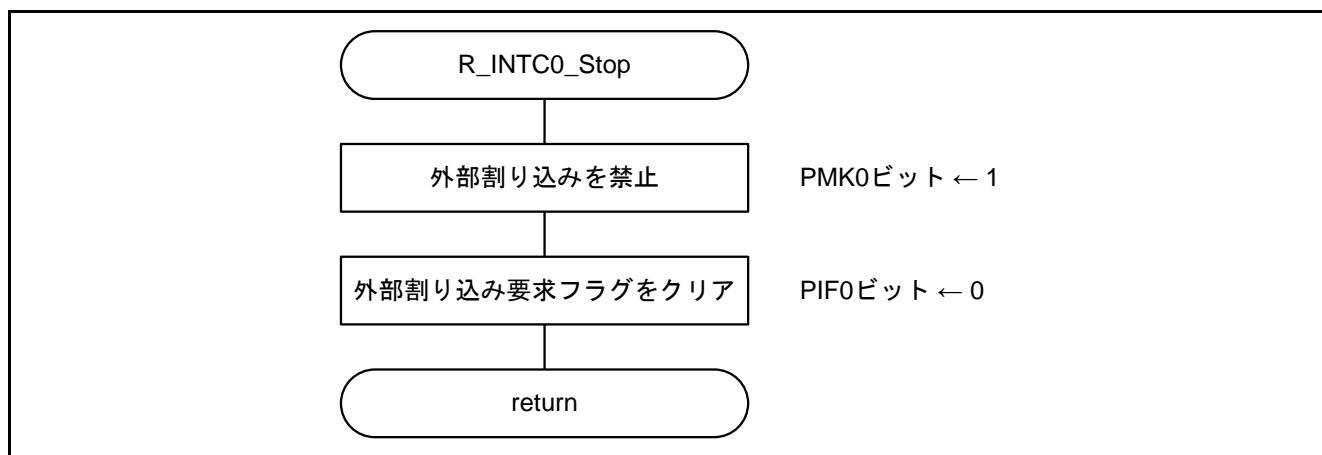


図 5.15 外部割り込み禁止関数

5.7.14 パリティ・チェック関数

図 5.16 にパリティ・チェック関数のフローチャートを示します。

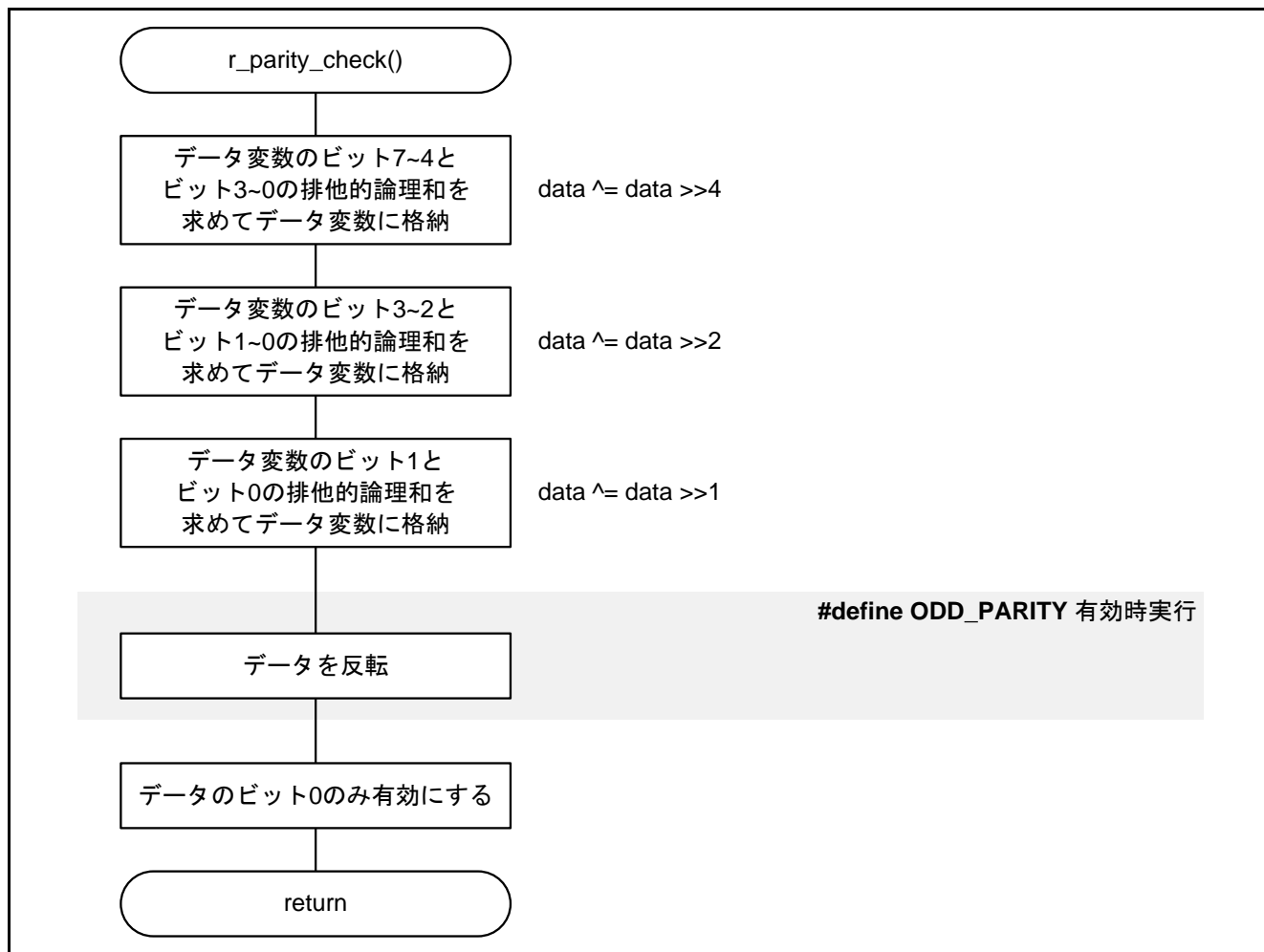


図 5.16 パリティ・チェック関数

5.7.15 データ順序並び替え関数

図 5.17 にデータ順序並び替え関数のフローチャートを示します。

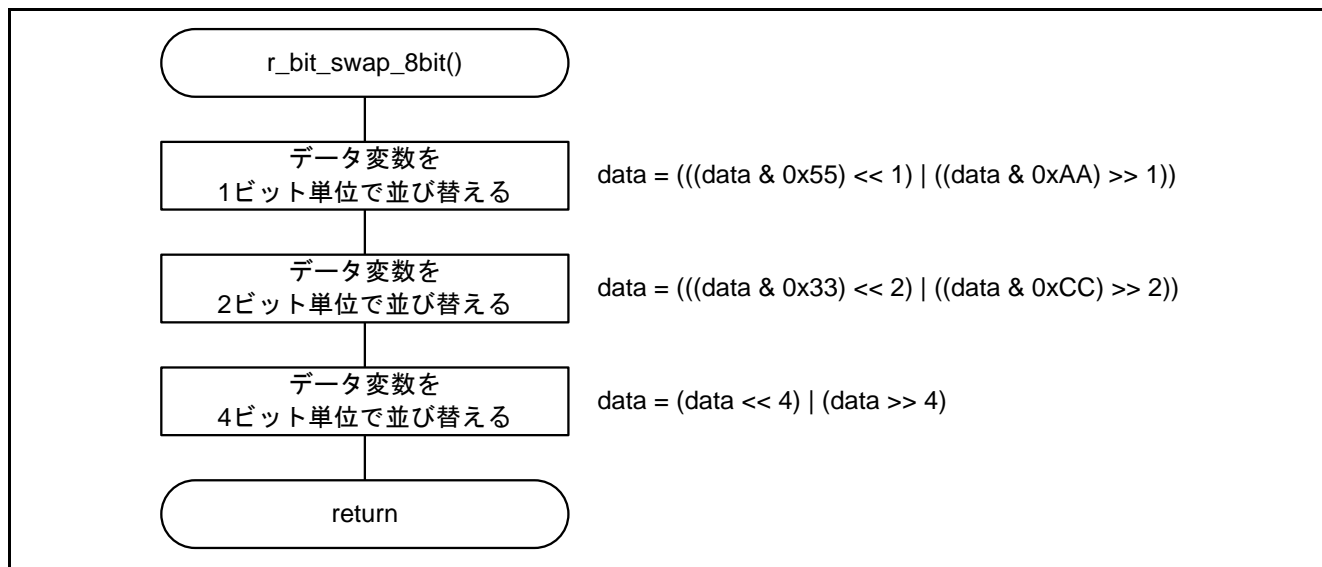


図 5.17 データ順序並び替え関数

5.7.16 外部割り込み処理

図 5.18 に外部割り込み処理のフローチャートを示します。

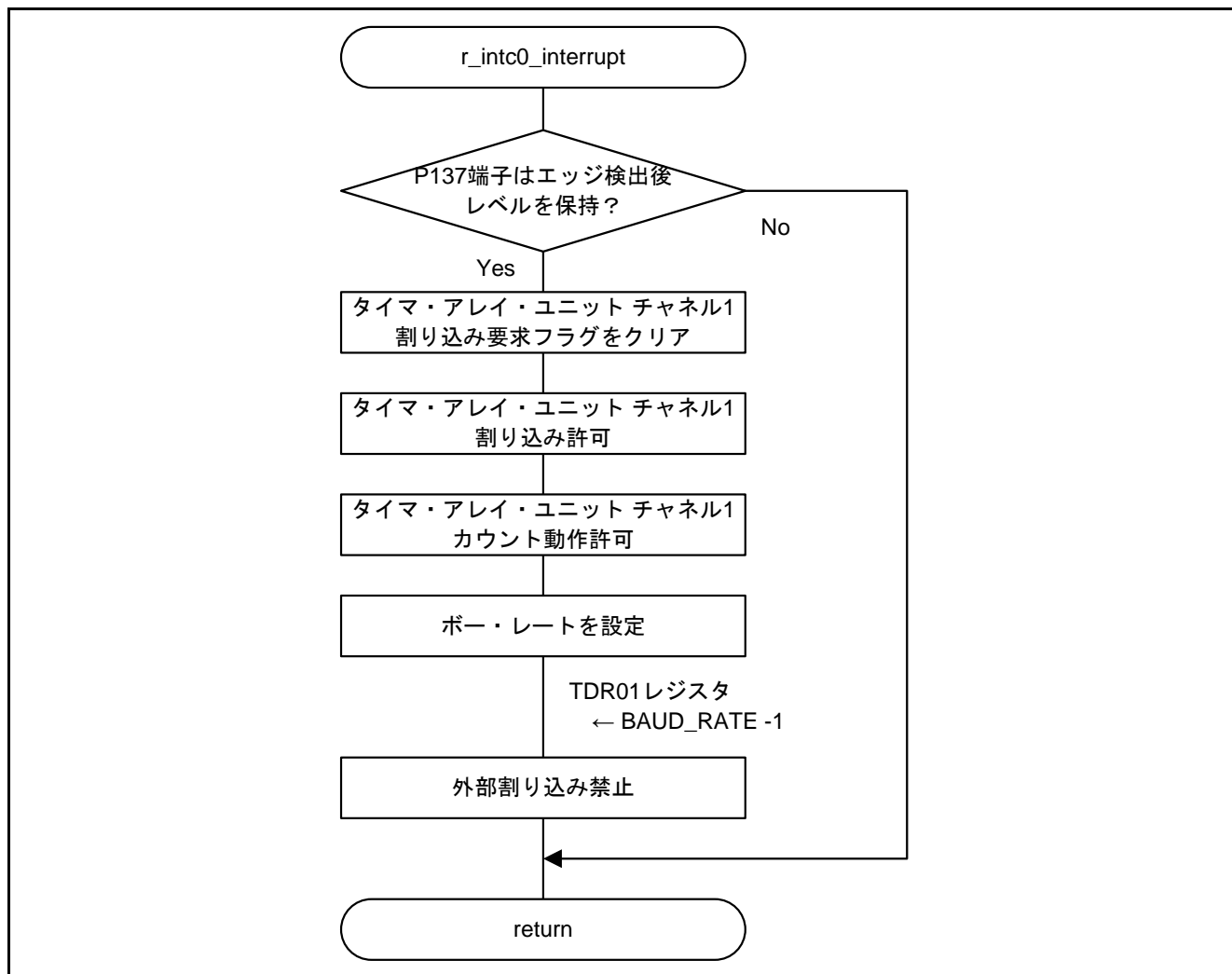


図 5.18 外部割り込み処理

5.7.17 タイマ・アレイ・ユニット チャンネル0 割り込み処理

図 5.19 にタイマ・アレイ・ユニット チャンネル0 割り込み処理のフローチャートを示します。

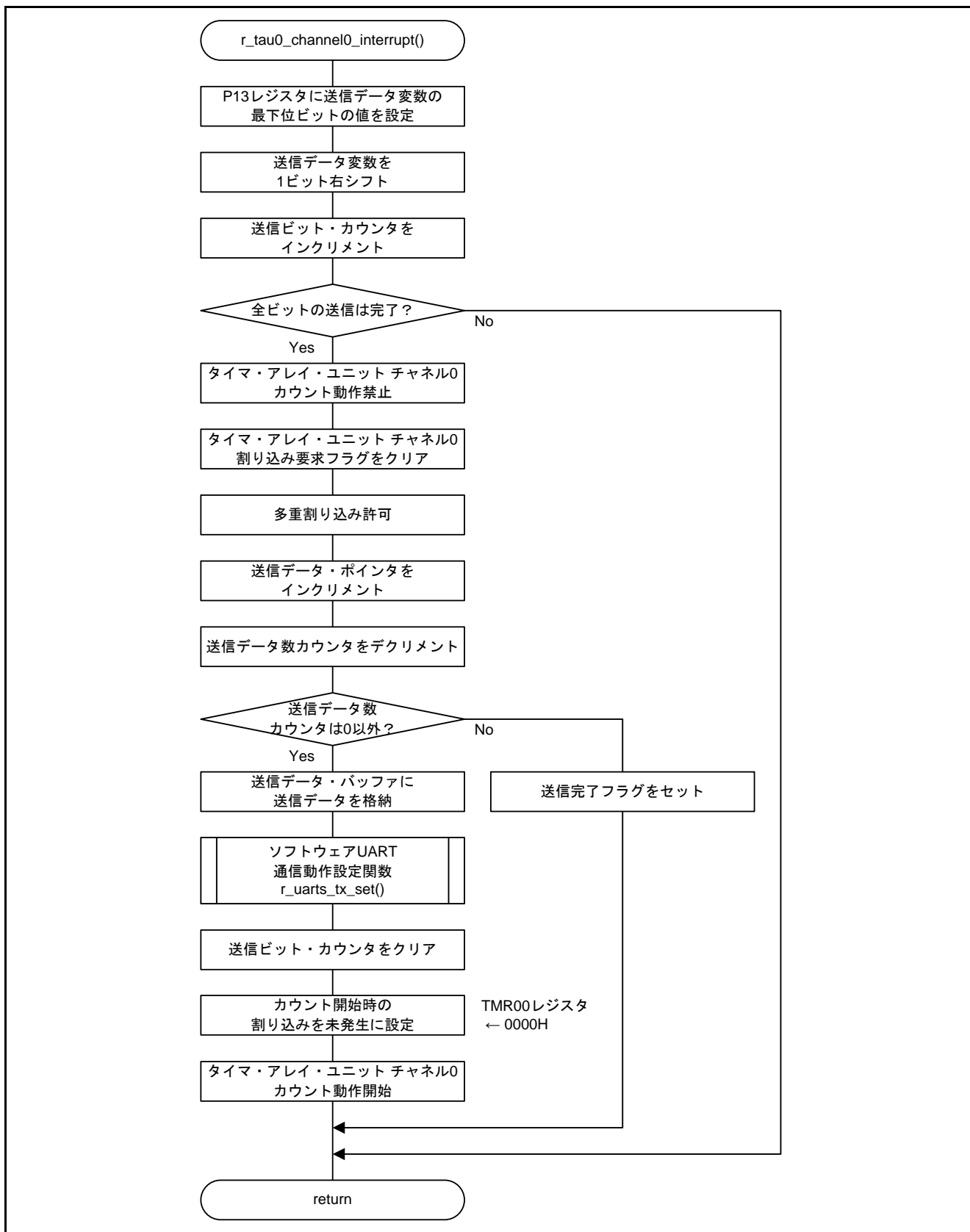


図 5.19 タイマ・アレイ・ユニット チャンネル0 割り込み処理

5.7.18 タイマ・アレイ・ユニット チャンネル 1 割り込み処理

図 5.20~図 5.22 にタイマ・アレイ・ユニット チャンネル 1 割り込み処理のフローチャートを示します。

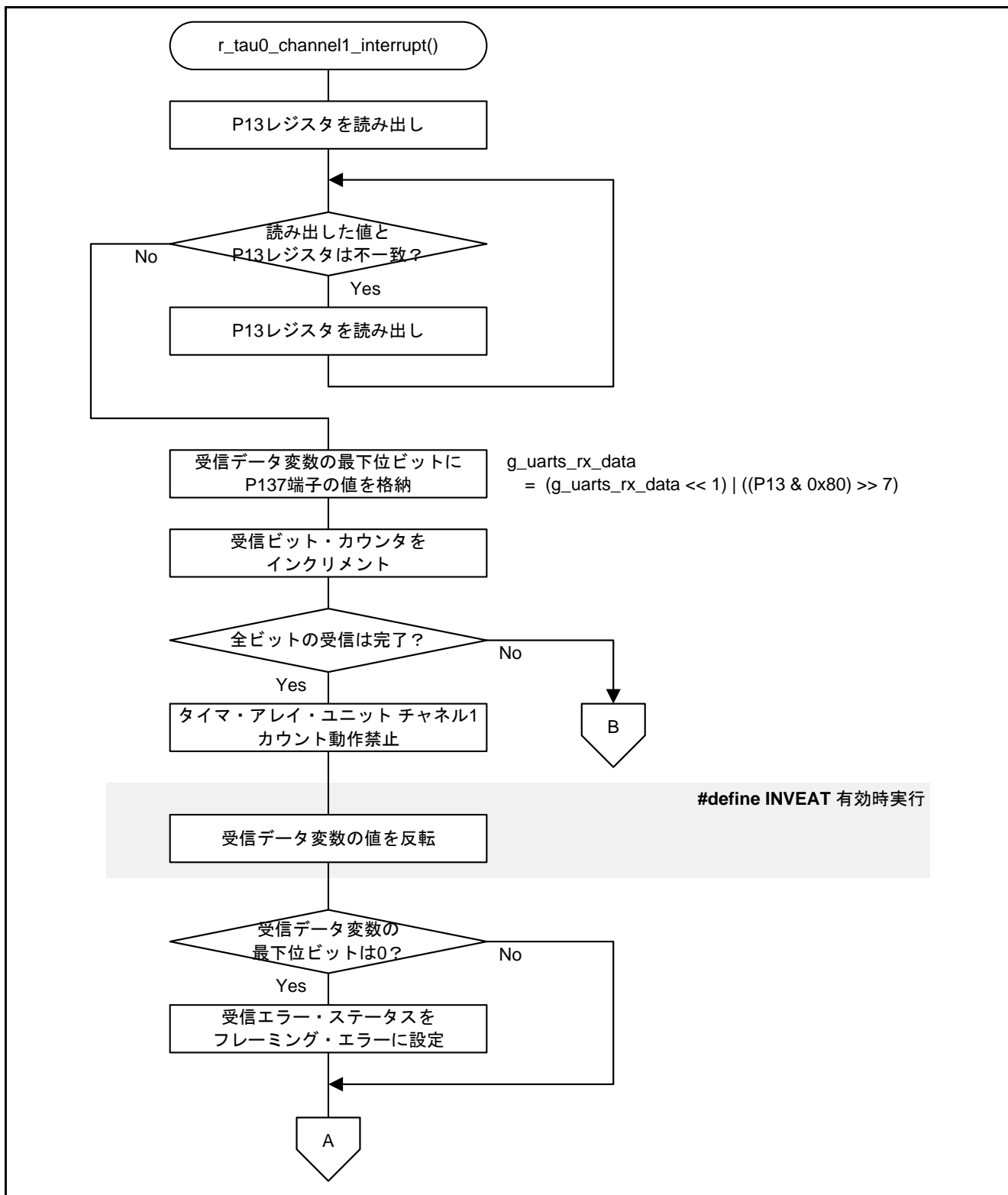


図 5.20 タイマ・アレイ・ユニット チャンネル 1 割り込み処理 (1/3)

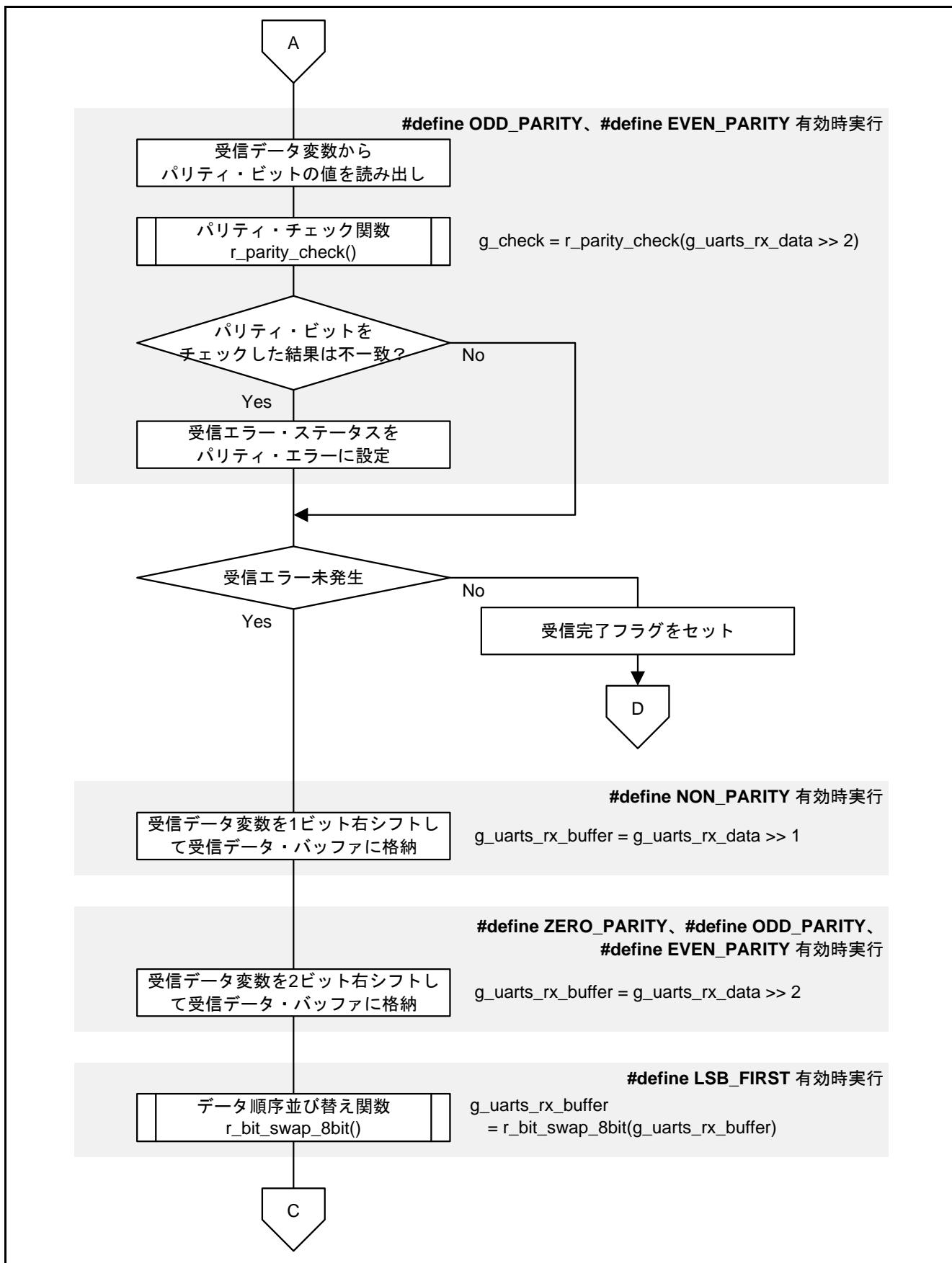


図 5.21 タイマ・アレイ・ユニット チャネル 1 割り込み処理 (2/3)

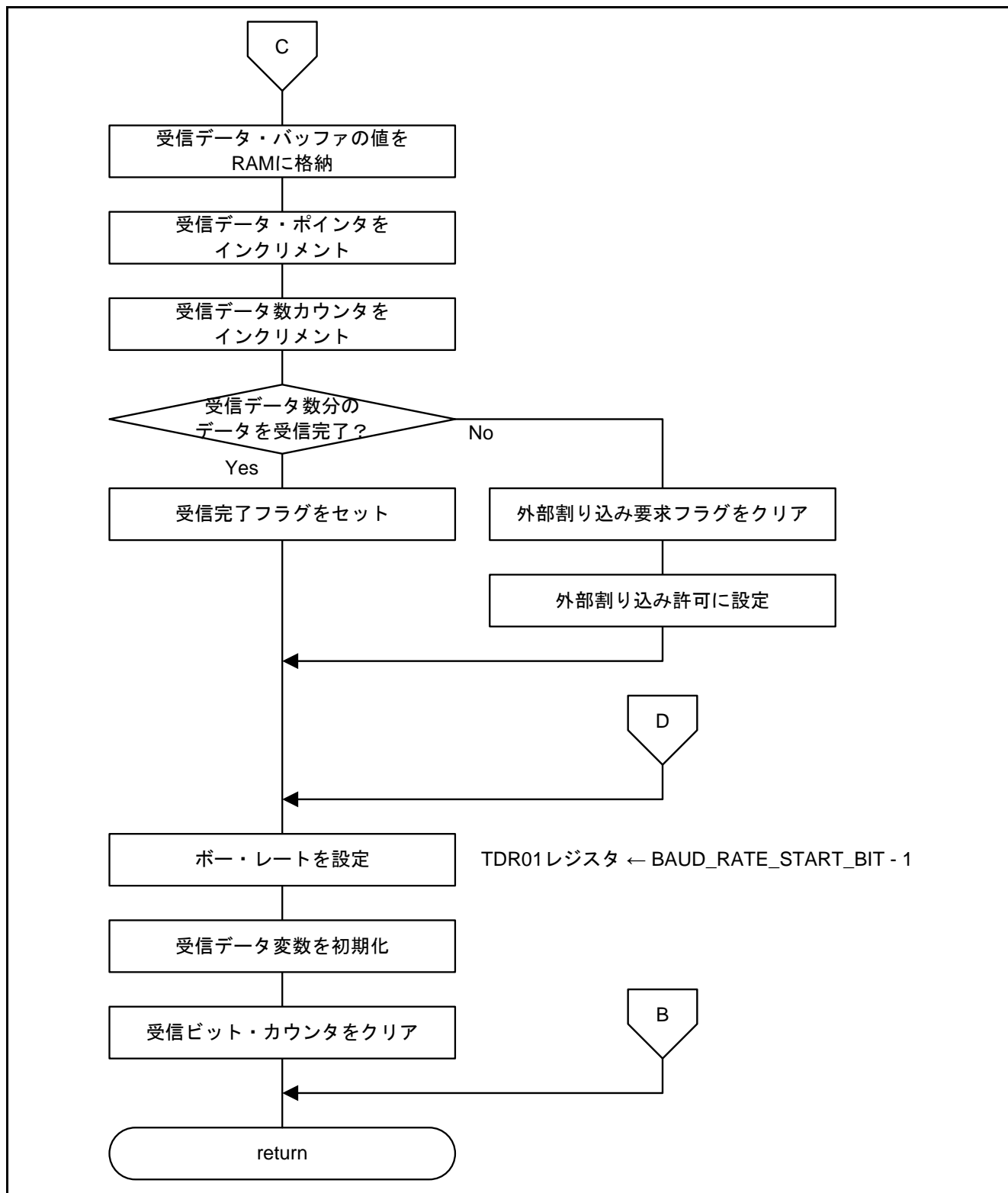


図 5.22 タイマ・アレイ・ユニット チャンネル1 割り込み処理 (3/3)

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)

(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/>

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。

改訂記録

| Rev. | 発行日 | 改訂内容 | |
|------|------------|------|------|
| | | ページ | ポイント |
| 1.00 | 2019.05.07 | — | 初版発行 |
| | | | |

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、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製品の入力がノイズなどに起因して、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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っていると、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。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まれます。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、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、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。
 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。
 6. 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。
 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、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。
 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。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. 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。
 10. 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いたします。
 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。
 12. 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.4.0-1 2017.11)

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。