

RX62T グループ

R01AN0637JJ0100

Rev.1.00

RIIC による EEPROM との通信

2011.07.15

要旨

本アプリケーションノートではルネサス MCU の RIIC (I²C バスインタフェース) を用いた EEPROM との通信 (シングルマスタ) について説明しています。

対象デバイス

RX62T グループ

RX62T グループと同様の I/O レジスタ (周辺装置制御レジスタ) を持つ他の RX ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等に変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートのご使用に際しては十分な評価を行ってください。

目次

| | |
|-------------------|----|
| 1. 仕様 | 2 |
| 2. 動作確認環境 | 3 |
| 3. 動作説明 | 4 |
| 4. ソフトウェア説明 | 6 |
| 5. 参考ドキュメント | 28 |

1. 仕様

EEPROM と通信を行い、8byte のデータを EEPROM に書き込みます。その後、書き込んだ 8byte のデータを読み込みます。書き込みと読み込みの間では、Acknowledge Polling を行うことで EEPROM への書き込み完了を確認しています。

1.1 接続図

本資料で説明する応用例の接続図を図 1 に示します。

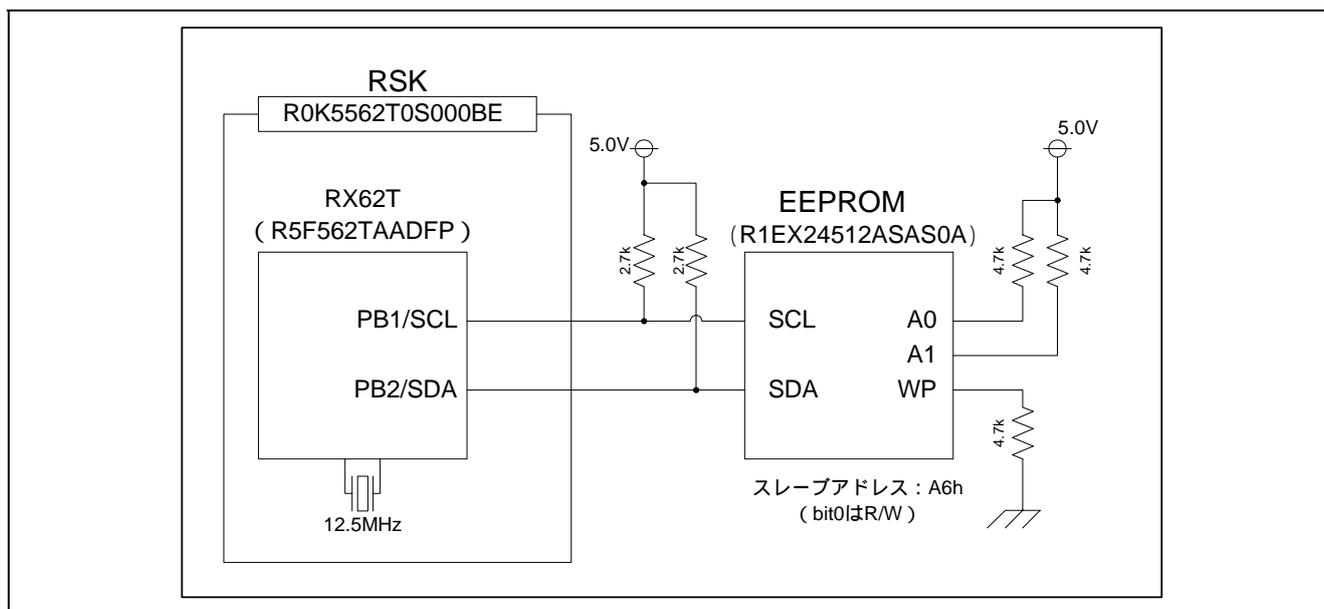


図 1 接続図

1.2 RIIC 設定

本資料で説明する応用例の RIIC の設定を表 1 に示します。

表 1 RIIC 設定

| 項目 | 内容 |
|------------|---|
| 動作周波数 | <ul style="list-style-type: none"> 入力クロック (EXTAL) : 12.5MHz システムクロック (ICLK) : 100MHz 周辺モジュールクロック (PCLK) : 50MHz 内部基準クロック (IICϕ) : 12.5MHz |
| マスタ/スレーブ | シングルマスタ |
| アドレスフォーマット | 7ビットアドレスフォーマット |
| 転送速度 | 400kbps |
| タイムアウト検出 | <ul style="list-style-type: none"> SCL ラインが Low のときカウント ロングモード (16ビットカウンタ (IICϕ)) : 約 5.24288ms |

1.3 EEPROM

本資料で説明する応用例の EEPROM の仕様を表 2 に示します。

表 2 EEPROM 仕様

| 項目 | 内容 |
|----------|---|
| 型名 | R1EX24512ASAS0A |
| 容量 | 512k (64-kword × 8-bit) |
| スレーブアドレス | スレーブアドレス : A6h bit0 は R/W ビット。bit1 と bit2 はそれぞれ A0、A1 端子の値に依存する。 <ul style="list-style-type: none">• A0 端子 : High• A1 端子 : High |
| ライトプロテクト | 常に解除 <ul style="list-style-type: none">• WP 端子 : Low |

2. 動作確認環境

本資料で説明する応用例の動作確認環境を表 3 に示します。

表 3 動作確認環境

| 項目 | 内容 |
|--------------------|---|
| デバイス | RX62T (R5F562TAADFP) |
| ボード | Renesas Starter Kit (R0K5562T0S000BE) |
| 電源電圧 | 5.0V (E1 より供給) |
| 入力クロック | 12.5MHz (ICLK=100MHz、PCLK=50MHz) |
| 動作温度 | 室温 |
| HEW | Version 4.09.00.007 |
| Toolchain | RX Standard Toolchain (V.1.0.2.0) |
| Debugger/Emulator | E1 エミュレータ |
| Debugger component | RX E1/E20 SYSTEM V.1.01.00 |

3. 動作説明

3.1 EEPROM への書き込み

マスタ送信を使用して EEPROM への書き込みを行います。RIIC がスタートコンディション(S)を送信し、続いて EEPROM のスレーブアドレスを送信します。このとき、8 ビット目は R/W ビットとなりますので、書き込み時には"0"を送信します (マスタ送信)。その後、8bit×2 のメモリアドレスと EEPROM へ書き込みたいデータを順次送信します。ここで送信する 8bit×2 のメモリアドレスは、書き込みを行う EEPROM 上のアドレスを示します。全データの送信が完了した後、ストップコンディション(P)を生成し、バスを解放します。なお、本アプリケーションノートでは、書き込みを行うメモリアドレスを 0000h としています。

図 2 に EEPROM への書き込み時の信号例を示します。

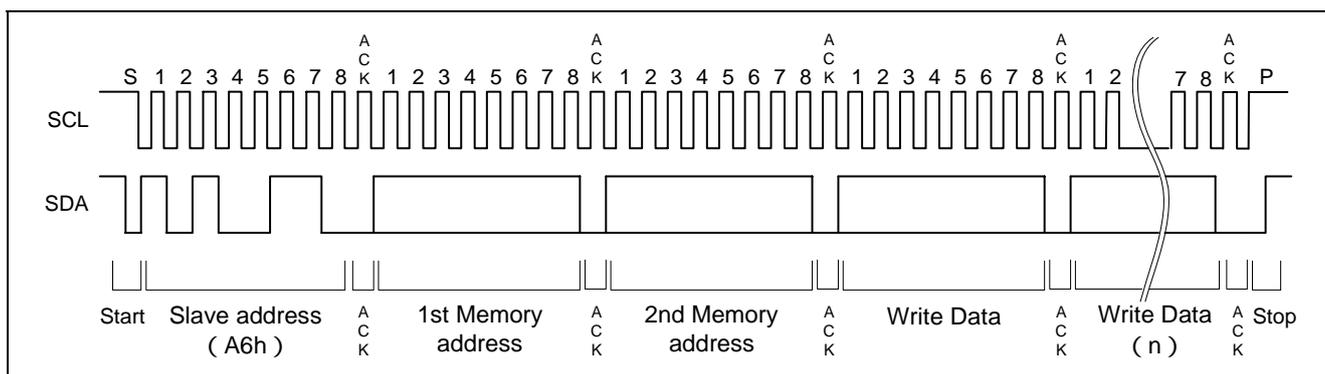


図 2 EEPROM への書き込み信号例

3.2 EEPROM からの読み込み

マスタ送信とマスタ受信 (複合フォーマット)を使用して EEPROM からデータを読み込みます。まず、RIIC がスタートコンディション(S)を送信し、EEPROM のスレーブアドレスと 8bit×2 のメモリアドレスを送信します。このときの EEPROM のスレーブアドレス送信では、R/W ビットとして"0"を送信します (マスタ送信)。その後、リスタートコンディション(Sr)を送信し、再度 EEPROM のスレーブアドレスを送信します。このときの EEPROM のスレーブアドレス送信では、R/W ビットとして"1"を送信します (マスタ受信)。EEPROM のスレーブアドレスを送信した後、続けてクロックを生成することで、EEPROM からデータを読み込みます。読み込み時、RIIC は 1byte 受信するごとに ACK を送信しますが、最終データに限り NACK を返します。そして、その後ストップコンディション(P)を生成します。なお、本アプリケーションノートでは、読み込みを行うメモリアドレスを 0000h としています。

図 3 に EEPROM からの読み込み時の信号例を示します。

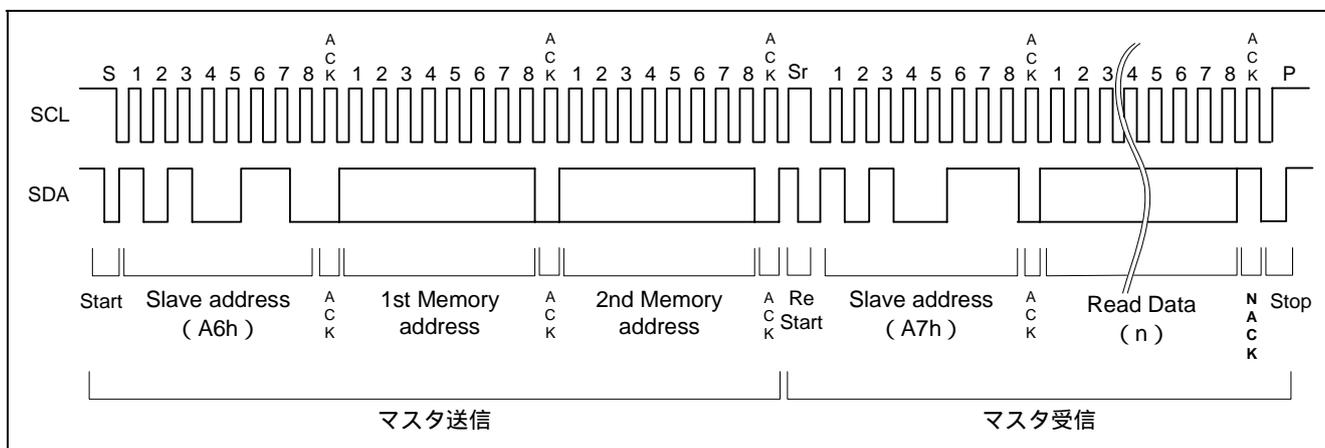


図 3 EEPROM からの読み込み信号例

3.3 Acknowledge Polling

EEPROM が書き換え中か否かを判断する方法として、Acknowledge Polling を使用します。Acknowledge Polling ではスタートコンディションに続いて、EEPROM スレーブアドレス、ストップコンディションを送信します。このとき、EEPROM が書き換え中の場合は ACK クロックに"1"を返してきます (NACK)。逆に、書き換えが完了していた場合は"0"を返してきます (ACK)。これによって、書き換え中か否かを判断します。

図 4 に Acknowledge Polling の信号例を示します。

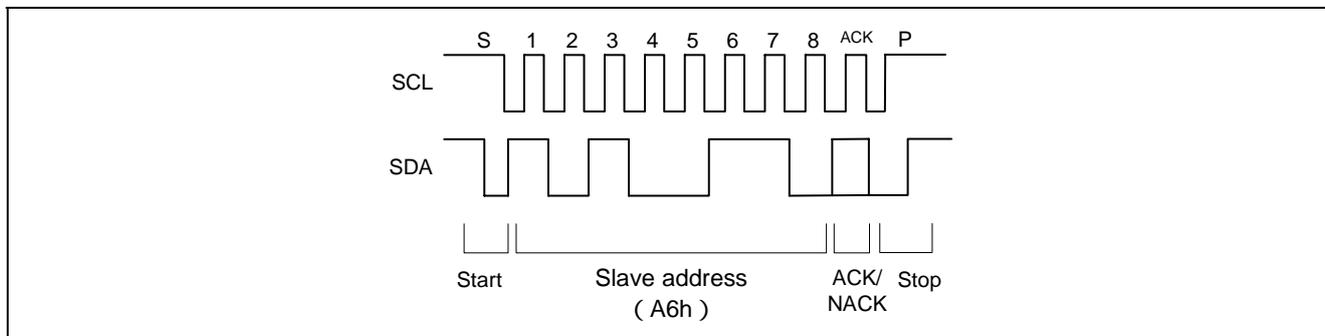


図 4 Acknowledge Polling 信号例

4. ソフトウェア説明

4.1 関数一覧

表 4、表 5 に関数一覧を示します。細字の関数は静的関数です。

表 4 ファイル"main.c"内の関数一覧

| 関数名 | 内容 | 備考 |
|----------------------|---------------------|------|
| main | メイン処理 | 図 7 |
| SampleEepromWrite | EEPROM への書き込み処理例 | 図 11 |
| SampleEepromRead | EEPROM からの読み込み処理例 | 図 12 |
| IICAckPolling | Acknowledge Polling | 図 13 |
| CpuCreate | CPU 初期設定 | 図 8 |
| CpuIntCreate | CPU 割り込み設定 | 図 9 |
| IICPortCreate | IIC ポート設定 | 図 10 |

表 5 ファイル"iic.c"内の関数一覧

| 関数名 | 内容 | 備考 |
|-----------------------|---|------|
| IIC_Create | IIC 初期設定 | 図 14 |
| IIC_Destroy | IIC 終了処理 | 図 15 |
| IIC_EepWrite | EEPROM への書き込み開始処理 | 図 16 |
| IIC_RandomRead | EEPROM からの読み込み開始処理 | 図 17 |
| IIC_GetStatus | IIC のステータスチェック | 図 18 |
| IIC_EEI_Int | 通信エラー / イベント発生割り込み | 図 19 |
| IIC_EEI_IntTimeOut | タイムアウト検出割り込み IIC_EEI_Int 内で呼ばれます | 図 20 |
| IIC_EEI_IntAL | アービトレーションロスト検出割り込み IIC_EEI_Int 内で呼ばれます | 図 21 |
| IIC_EEI_IntSP | ストップコンディション検出割り込み IIC_EEI_Int 内で呼ばれます | 図 22 |
| IIC_EEI_IntST | スタートコンディション検出割り込み IIC_EEI_Int 内で呼ばれます | 図 23 |
| IIC_EEI_IntNack | NACK 検出割り込み IIC_EEI_Int 内で呼ばれます | 図 24 |
| IIC_RXI_Int | 受信データフル割り込み | 図 25 |
| IIC_RXI_IntEepRead | EEPROM からの読み込み処理 (マスタ受信部分) IIC_RXI_Int 内で呼ばれます | 図 26 |
| IIC_TXI_Int | 送信データエンプティ割り込み | 図 27 |
| IIC_TXI_IntEepWrite | EEPROM への書き込み処理 IIC_TXI_Int 内で呼ばれます | 図 28 |
| IIC_TXI_IntEepRead | EEPROM からの読み込み処理 (マスタ送信部分) IIC_TXI_Int 内で呼ばれます | 図 29 |
| IIC_TEI_Int | 送信終了割り込み | 図 30 |
| IIC_TEI_IntEepWrite | EEPROM 書き込み処理後の送信終了処理 IIC_TEI_Int 内で呼ばれます | 図 31 |
| IIC_TEI_IntEepRead | EEPROM 読み込み処理後の送信終了処理 IIC_TEI_Int 内で呼ばれます | 図 32 |
| IIC_GenCikSP | 異常発生時のストップコンディション生成処理 IIC_EEI_IntTimeOut と IIC_EEI_IntAL 内で呼ばれます | 図 33 |
| IIC_Error | エラー処理 | 図 34 |

4.2 変数

4.2.1 構造体

関数 IIC_EepWrite と関数 IIC_RandomRead の引数で使用する構造体を図 5 に示します。また、構造体の各メンバの一覧を表 6 に示します。

```

struct str_IIC_API_T
{
    uint8_t      SlvAdr;          /* Slave Address, Don't set bit0. It's a Read/Write bit */
    uint16_t     PreCnt;         /* Number of Predata */
    uint8_t      *pPreData;     /* Pointer for PreData (Memory Addr of EEPROM) */
    uint32_t     RWCnt;         /* Number of Data */
    uint8_t      *pRWData;     /* Pointer for Data buffer */
};
typedef struct str_IIC_API_T IIC_API_T;

```

図 5 関数 IIC_EepWrite と関数 IIC_RandomRead の引数用構造体

表 6 構造体"IIC_API_T"のメンバー一覧

| 構造体メンバ | 設定可能範囲 | 内容 |
|-----------|----------------------------|--|
| SlvAdr | 00h ~ FEh | スレーブアドレス。 最下位ビットは R/W ビットであるため、常に"0"を設定してください。 |
| PreCnt | 00h ~ FFh | メモリアドレスカウンタ。 本資料で使用する EEPROM の場合、常に"2"。 |
| *pPreData | - | メモリアドレス格納バッファポインタ。 書き込み時：データを書き込む EEPROM 上のアドレス（書き込み先）。 読み込み時：データを読み出す EEPROM 上のアドレス（読み込み元）。 |
| RWCnt | 0000 0000h ~ FFFF FFFFh | データカウンタ。 書き込み時：EEPROM へ書き込むデータ数。 読み込み時：EEPROM から読み出すデータ数。 |
| *pRWData | - | データ格納バッファポインタ。 書き込み時：EEPROM へ書き込むデータの格納元。 読み込み時：EEPROM から読み出すデータの格納先。 |

4.2.2 変数一覧

表 7、表 8 に変数一覧を示します。

表 7 ファイル"main.c"内の変数一覧

| 関数名 | 内容 |
|---------------------------|--|
| uint8_t trm_buff[256] | 送信データバッファ |
| uint8_t rcv_buff[256] | 受信データバッファ |
| uint8_t trm_eeprom_adr[2] | EEPROM スレーブアドレス格納バッファ (書き込み時) |
| uint8_t rcv_eeprom_adr[2] | EEPROM スレーブアドレス格納バッファ (読み込み時) |
| IIC_API_T iic_buff_prm[2] | 関数 IIC_EepWrite と関数 IIC_RandomRead の引数用構造体 |

表 8 ファイル"iic.c"内の変数一覧

| 関数名 | 内容 |
|-----------------------------|---|
| static IIC_API_T iic_buff | 関数 IIC_EepWrite と関数 IIC_RandomRead の引数用構造体 (関数 IIC_EepWrite と関数 IIC_RandomRead 兼用) |
| static int8_t iic_mode | 内部モード |
| static int8_t iic_status | IIC ステータス |
| static uint32_t iic_trm_cnt | 内部 IIC 送信カウンタ |
| static uint32_t iic_rcv_cnt | 内部 IIC 受信カウンタ |

4.2.3 列挙型

IIC ステータス、IIC バスステータス、内部モード、関数"IIC_EepWrite"と関数"IIC_RandomRead"のリターン値は列挙型で宣言されています。IIC ステータス一覧と遷移図をそれぞれ表 9、図 6 に示します。また、IIC バスステータスを表 10 に、内部モード一覧を表 11 に、関数"IIC_EepWrite"と関数"IIC_RandomRead"のリターン値を表 12 にそれぞれ示します。

IIC ステータスは関数"IIC_GetStatus"を呼んだ際に、第一引数のポインタで示されたアドレスに格納されます。内部モードは、本サンプルコードの IIC 関連の関数が内部でのみ使用するモードです。

表 9 IIC ステータス一覧 (enum RiicStatus_t)

| 定義名 | 内容 |
|------------------------------|---|
| RIIC_STATUS_IDLE | IDLE 状態。 関数 IIC_Create で初期設定を行った後、この状態に遷移します。 また、EEPROM への書き込みまたは EEPROM からの読み込みが正しく終了した場合 (ストップコンディション検出後) も、この状態に遷移します。 |
| RIIC_STATUS_ON_COMMUNICATION | 通信中。 関数 IIC_EepWrite または関数 IIC_RandomRead で通信を開始するとこの状態に遷移します。 |
| RIIC_STATUS_NACK | NACK 受信。 NACK を受信するとこの状態に遷移します。 |
| RIIC_STATUS_FAILED | 通信失敗。 EEPROM への書き込みまたは EEPROM からの読み込みが完了する前にストップコンディションを検出するとこの状態に遷移します。 本サンプルコードでは、タイムアウトまたはアービトレーションロストが発生するとストップコンディションを生成するため、この状態に遷移します。 |

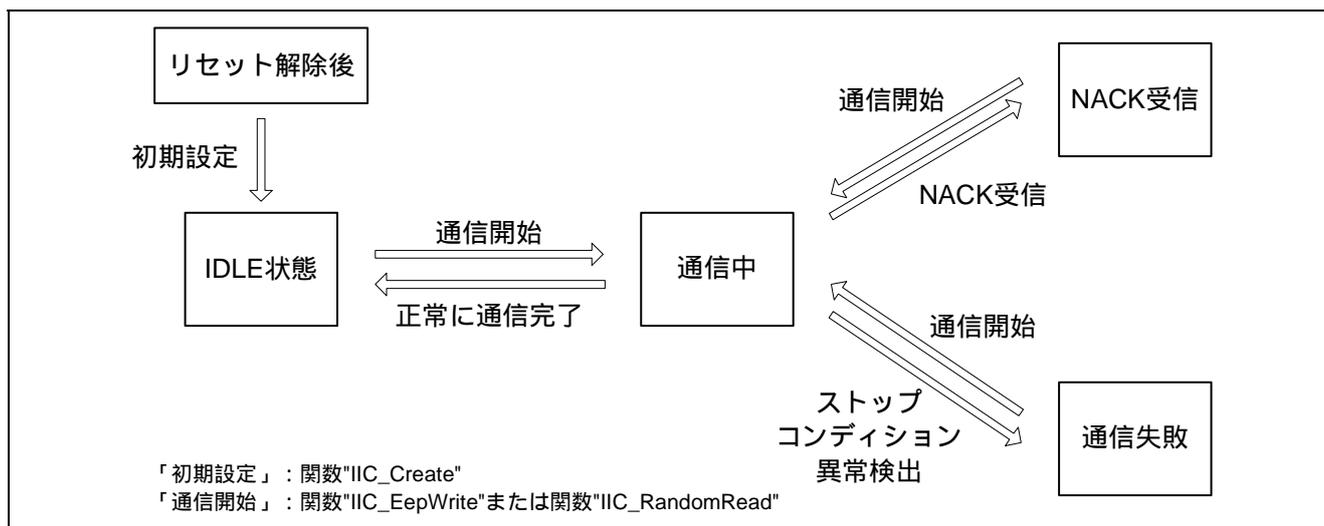


図 6 IIC ステータス遷移図

表 10 IIC バスステータス (enum RiicBusStatus_t)

| 定義名 | 内容 |
|----------------------|------------|
| RIIC_BUS_STATUS_FREE | IIC バスフリー。 |
| RIIC_BUS_STATUS_BBSY | IIC バスビジー。 |

表 11 内部モード一覧 (enum RiicInternalMode_t)

| 定義名 | 内容 |
|--------------------|---|
| IIC_MODE_IDLE | IDLE モード。 関数 IIC_Create による初期設定、またはストップコンディションを検出するとこのモードに遷移します。 |
| IIC_MODE_EEP_READ | EEPROM リードモード。 関数 IIC_RandomRead で通信を開始するとこのモードに遷移します。 |
| IIC_MODE_EEP_WRITE | EEPROM ライトモード。 関数 IIC_EepWrite で通信を開始するとこのモードに遷移します。 |

表 12 関数"IIC_EepWrite"と関数"IIC_RandomRead"のリターン値 (enum RiicEepFnc_t)

| 定義名 | 内容 |
|-----------------|--|
| RIIC_OK | 正常に通信が開始された場合にこの値を返します。 |
| RIIC_BUS_BUSY | I ² C バスがビジーだった場合にこの値を返します。 |
| RIIC_MODE_ERROR | RIIC が通信中の場合にこの値を返します。 |
| RIIC_PRM_ERROR | 引数の値が正しくなかった場合にこの値を返します。 (関数"IIC_RandomRead"のみこの値を使用します。) |

4.3 関数仕様

サンプルコードの RIIC を制御する関数仕様を示します。

| IIC_Create | |
|--------------|---|
| 概要 | RIIC の初期設定を行います。 |
| ヘッダ | r_apn_iic.h |
| 宣言 | void IIC_Create(void) |
| 説明 | 以下の設定を行います。 <ul style="list-style-type: none"> ・転送速度設定：400kbps ・割り込み設定 ・タイムアウト設定 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | |
| IIC_Destroy | |
| 概要 | RIIC を停止します。 |
| ヘッダ | r_apn_iic.h |
| 宣言 | void IIC_Destroy(void) |
| 説明 | RIIC を停止し、RIIC 関連の全レジスタをクリアします。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | 通信中に本関数を実行した場合も、強制的に RIIC を停止します。 |
| IIC_EepWrite | |
| 概要 | EEPROM への書き込みを開始します。 |
| ヘッダ | r_apn_iic.h |
| 宣言 | int8_t IIC_EepWrite(IIC_API_T) |
| 説明 | マスタ送信を使用して EEPROM への書き込みを行います。I ² C バスがビジーだった場合、または RIIC が通信中の場合は、マスタ送信を開始しません。 |
| 引数 | IIC_API_T data1 |
| リターン値 | 正常に通信が開始された場合：RIIC_OK I ² C バスがビジーだった場合：RIIC_BUS_BUSY RIIC が通信中の場合：RIIC_MODE_ERROR |
| 備考 | 引数 IIC_API_T data1 については「4.2.1 構造体」をご参照ください。 リターン値については、「4.2.3 列挙型」をご参照ください。 引数のメンバであるスレーブアドレス (SlvAdr) の bit0 は"0"に設定してください。 |

| IIC_RandomRead | |
|----------------|--|
| 概要 | EEPROM からの読み込みを開始します。 |
| ヘッダ | r_apn_iic.h |
| 宣言 | int8_t IIC_RandomRead(IIC_API_T); |
| 説明 | マスタ送信とマスタ受信 (複合フォーマット) を使用して EEPROM からデータを読み込みます。I ² C バスがビジーだった場合、または RIIC が通信中の場合は、マスタ送信を開始しません。 |
| 引数 | IIC_API_T data1 |
| リターン値 | 正常に通信が開始された場合 : RIIC_OK I ² C バスがビジーだった場合 : RIIC_BUS_BUSY RIIC が通信中の場合 : RIIC_MODE_ERROR 引数の値が正しくなかった場合 : RIIC_PRM_ERROR |
| 備考 | 引数 IIC_API_T data1 については「4.2.1 構造体」をご参照ください。 リターン値については、「4.2.3 列挙型」をご参照ください。 メモリアドレスカウンタとデータカウンタの両方を"0"すると、引数の値が正しくないと判定します。 引数のメンバであるスレーブアドレス (SlvAdr) の bit0 は"0"に設定してください。 |

| IIC_GetStatus | |
|---------------|--|
| 概要 | IIC のステータスを確認します。 |
| ヘッダ | r_apn_iic.h |
| 宣言 | void IIC_GetStatus(enum RiicStatus_t*, enum RiicBusStatus_t*); |
| 説明 | IIC のステータスを第一引数で示された領域に格納します。また、IIC のバスの状態を第二引数で示された領域に格納します。 |
| 引数 | enum RiicStatus_t *data1 enum RiicBusStatus_t *data2 |
| リターン値 | なし |
| 備考 | 引数については、「4.2.3 列挙型」をご参照ください。 |

4.4 フローチャート

以下に各関数のフローチャートを示します。

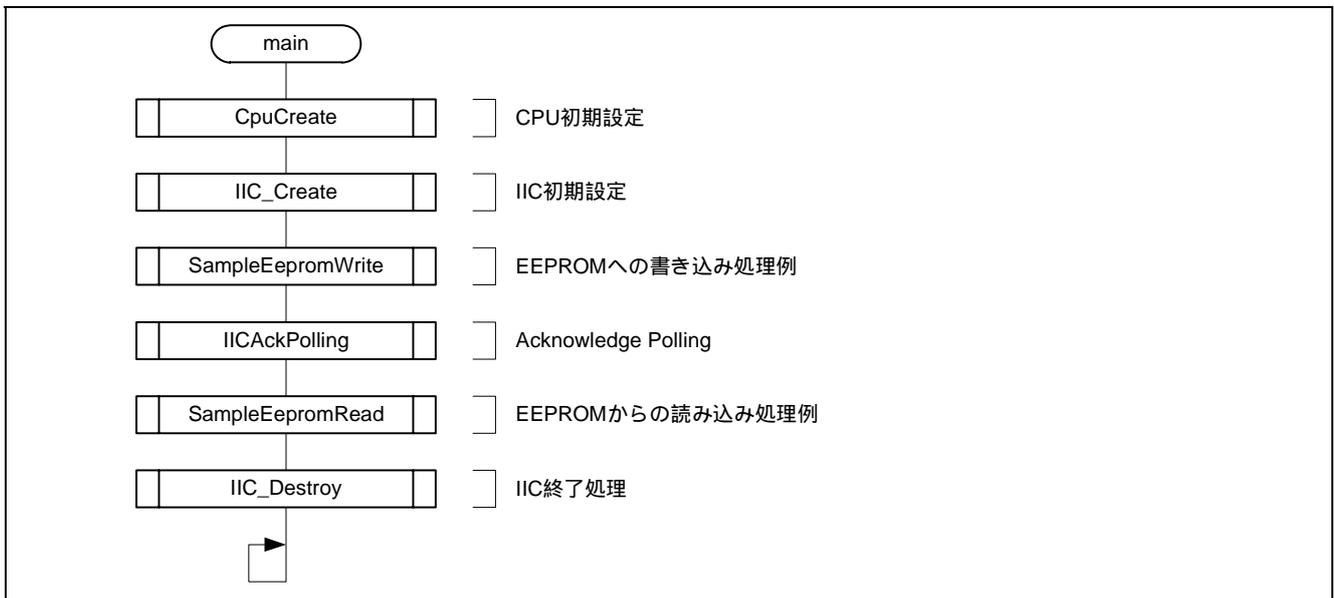


図 7 main 処理

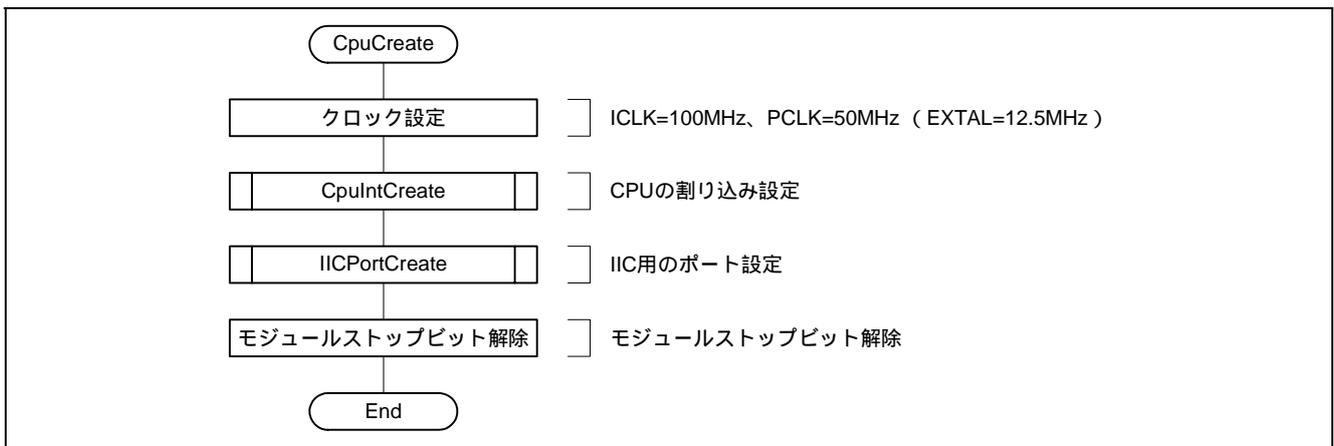


図 8 CPU 初期設定

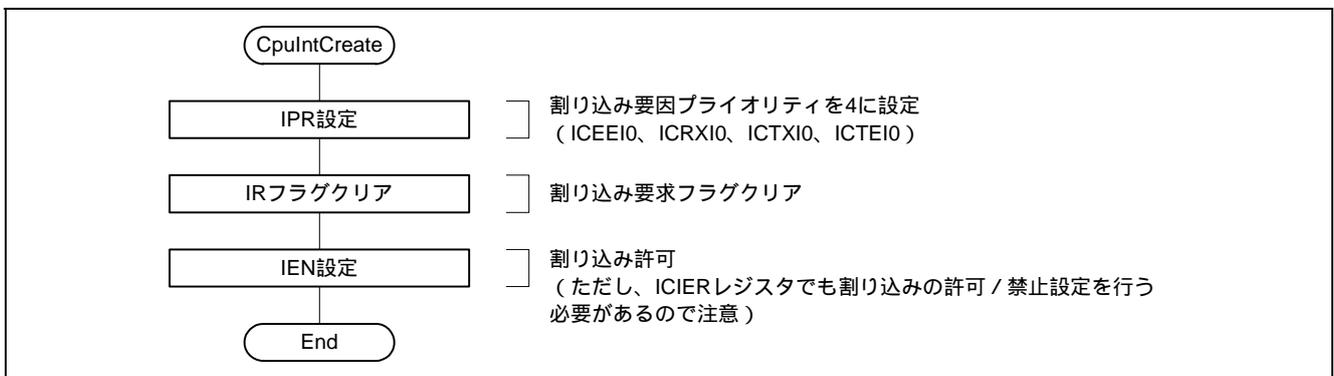


図 9 CPU の割り込み設定

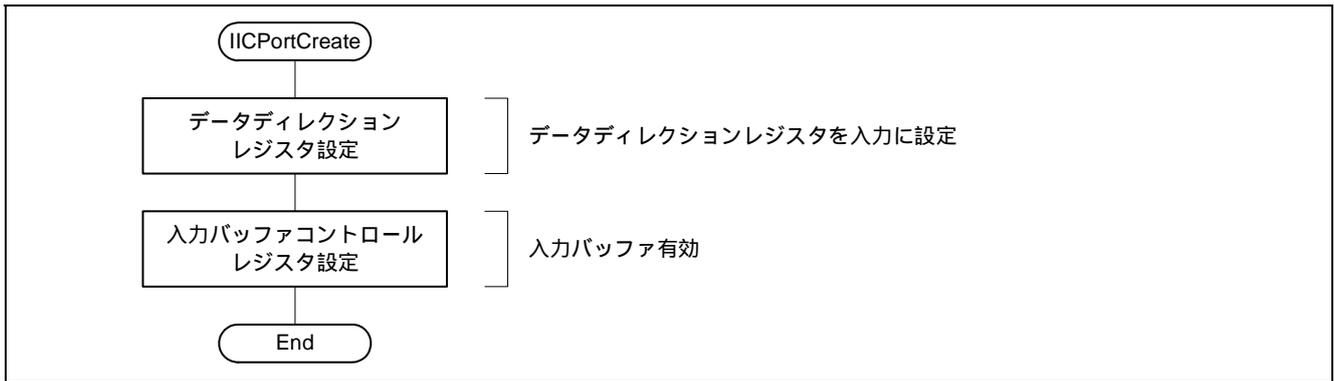


図 10 IIC 用のポート設定

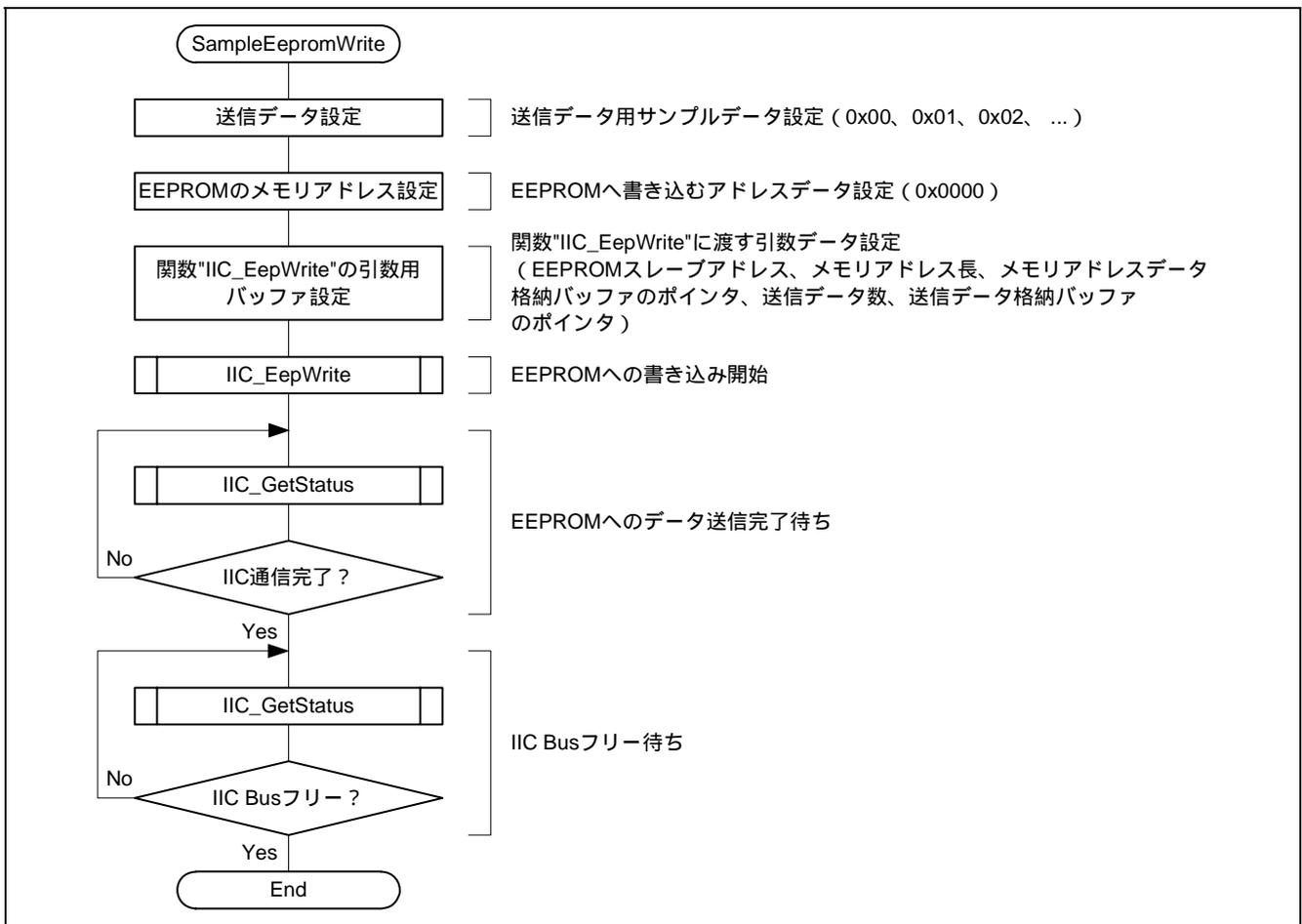


図 11 EEPROM への書き込み処理例

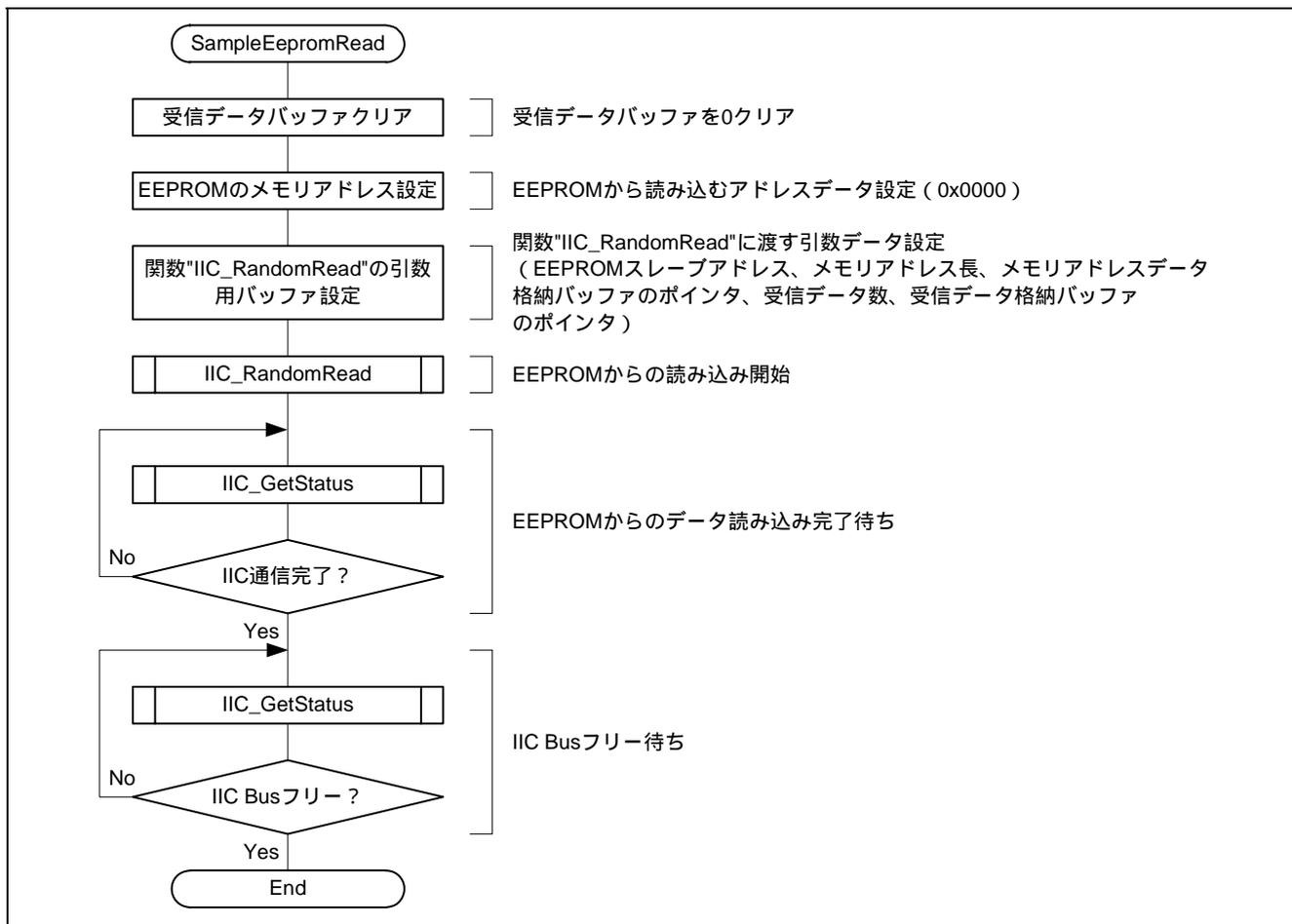


図 12 EEPROM からの読み込み処理例

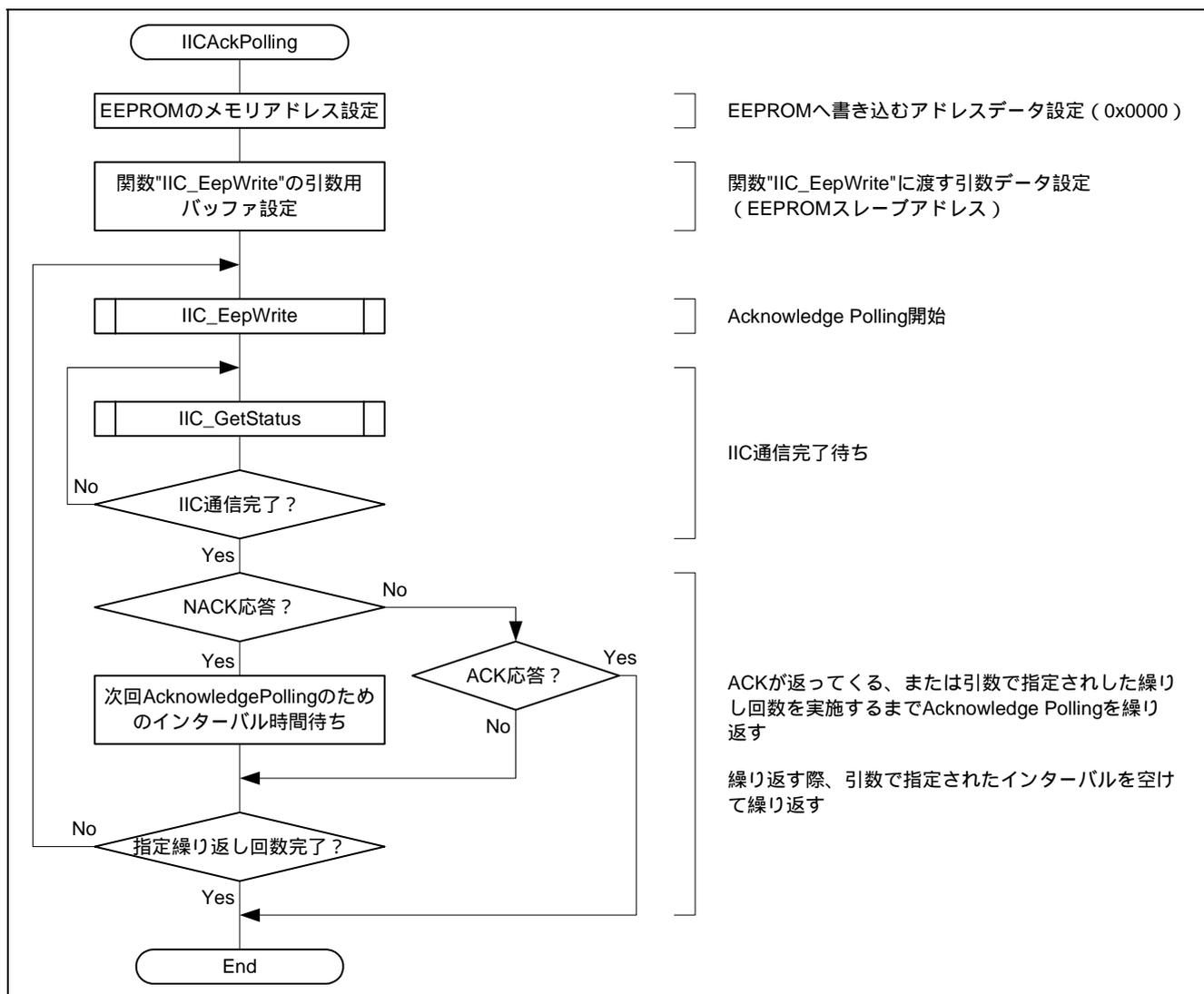


図 13 Acknowledge Polling

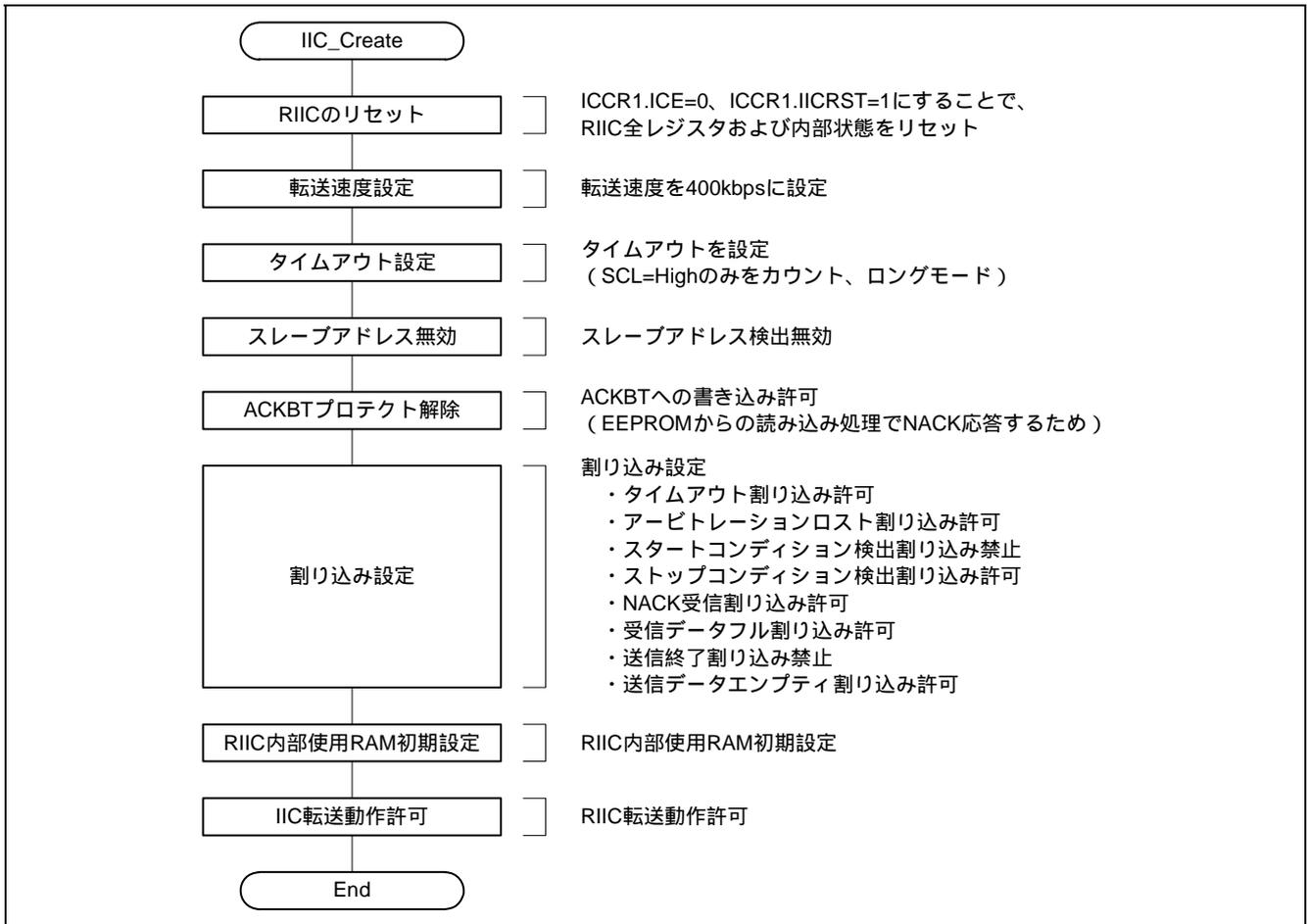


図 14 IIC 初期設定



図 15 IIC 終了処理

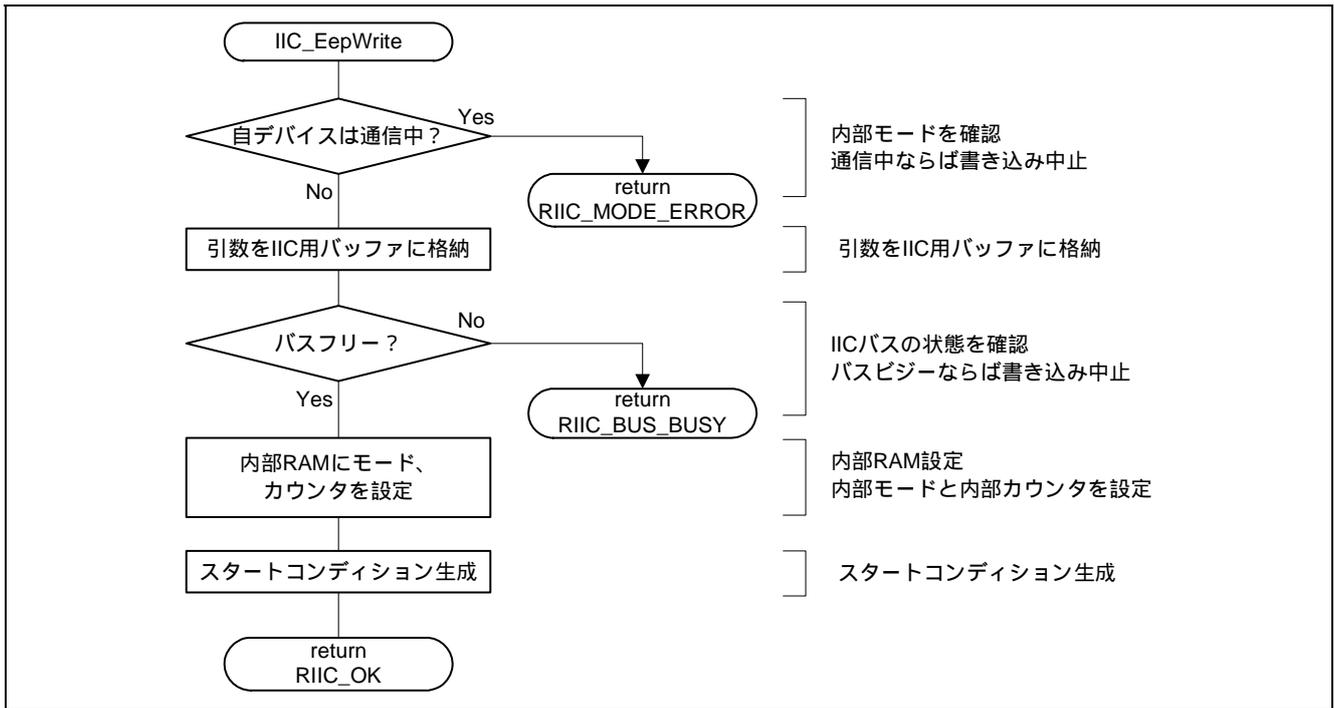


図 16 EEPROM への書き込み開始処理

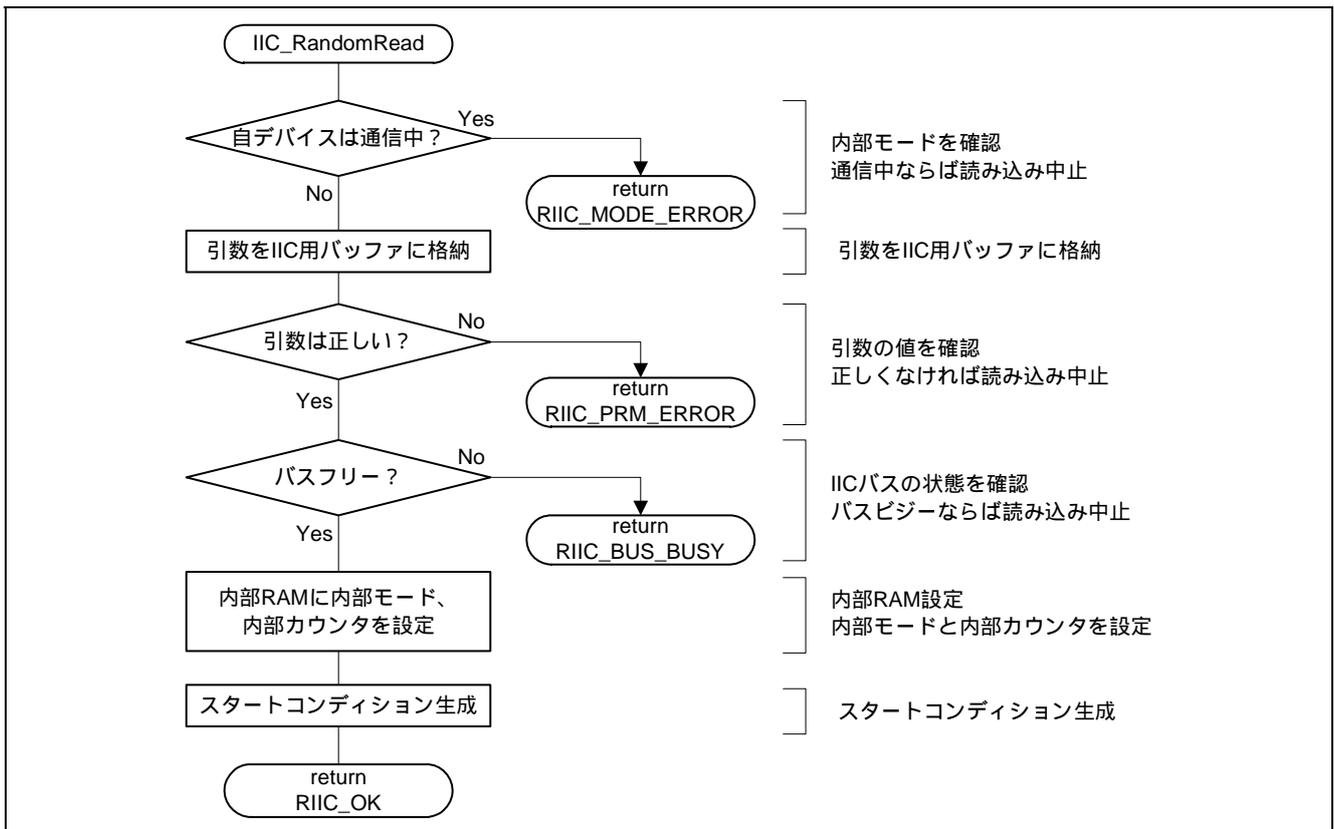


図 17 EEPROM からの読み込み開始処理

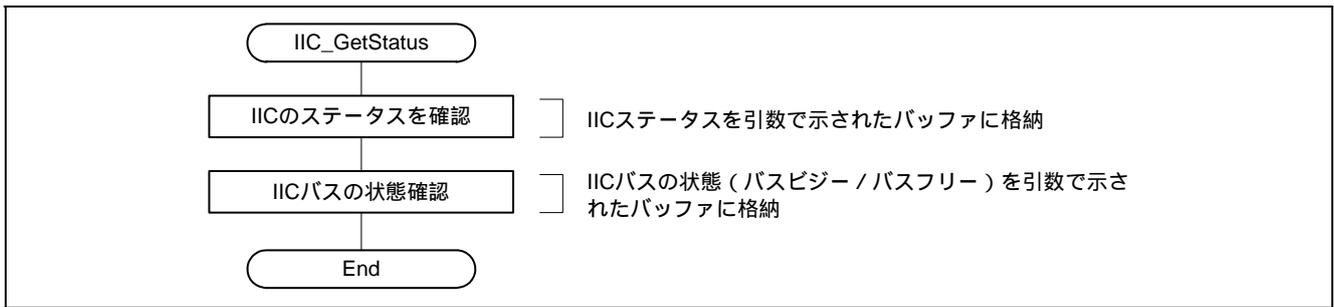


図 18 IIC 状態確認処理

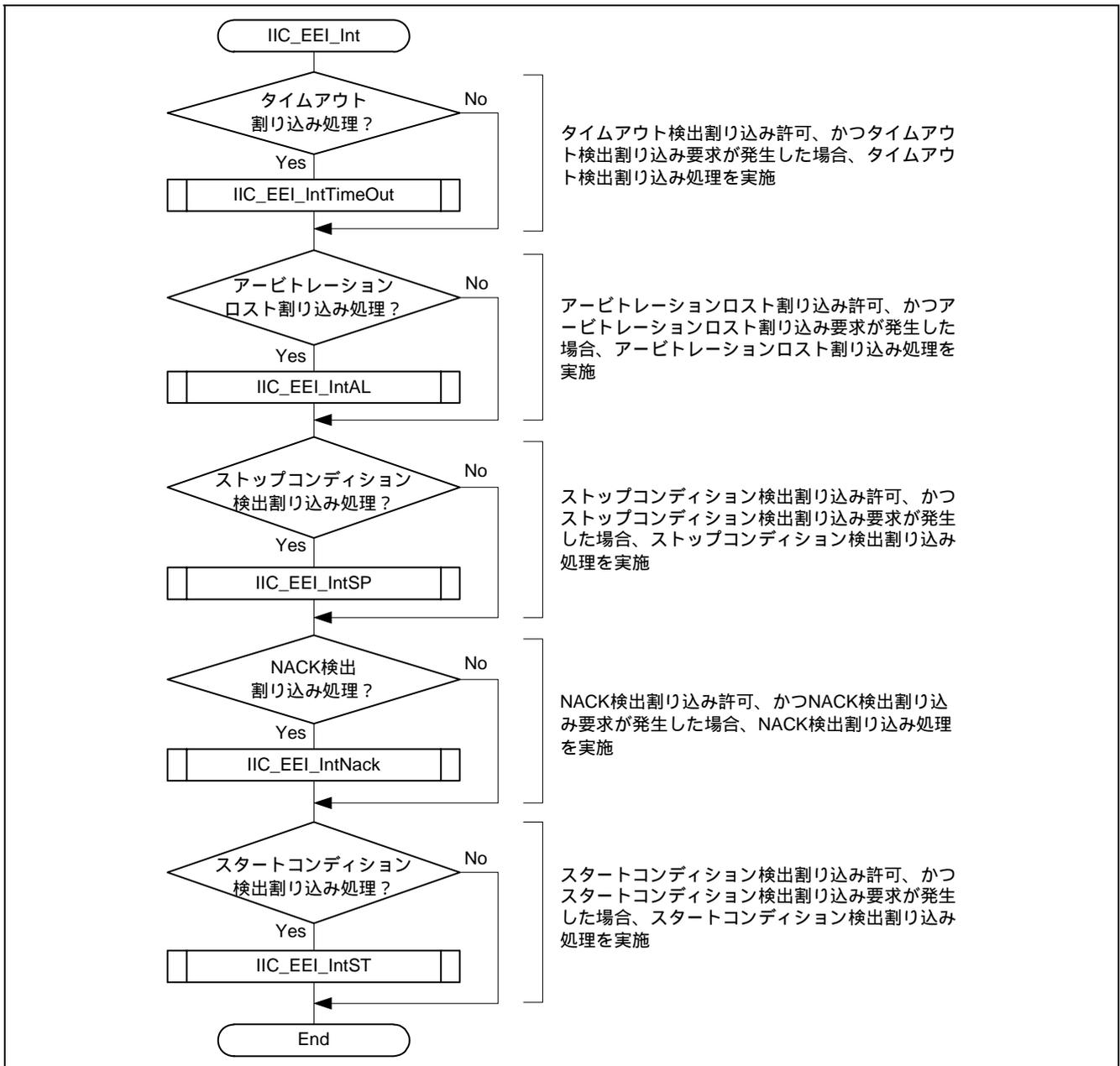


図 19 通信エラー/イベント発生割り込み



図 20 タイムアウト検出割り込み

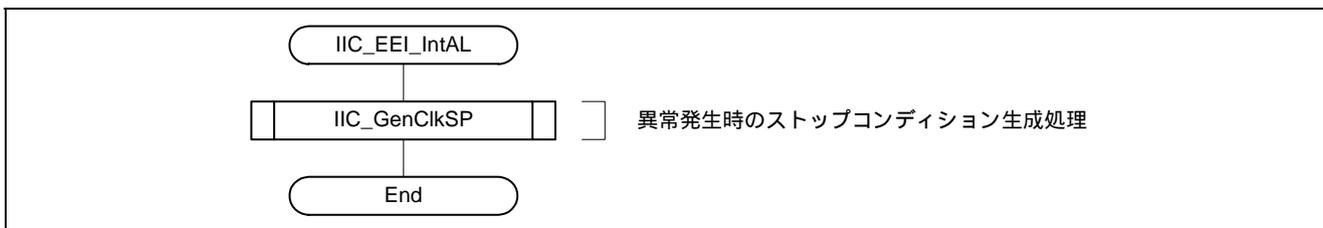


図 21 アービトレーションロスト検出割り込み

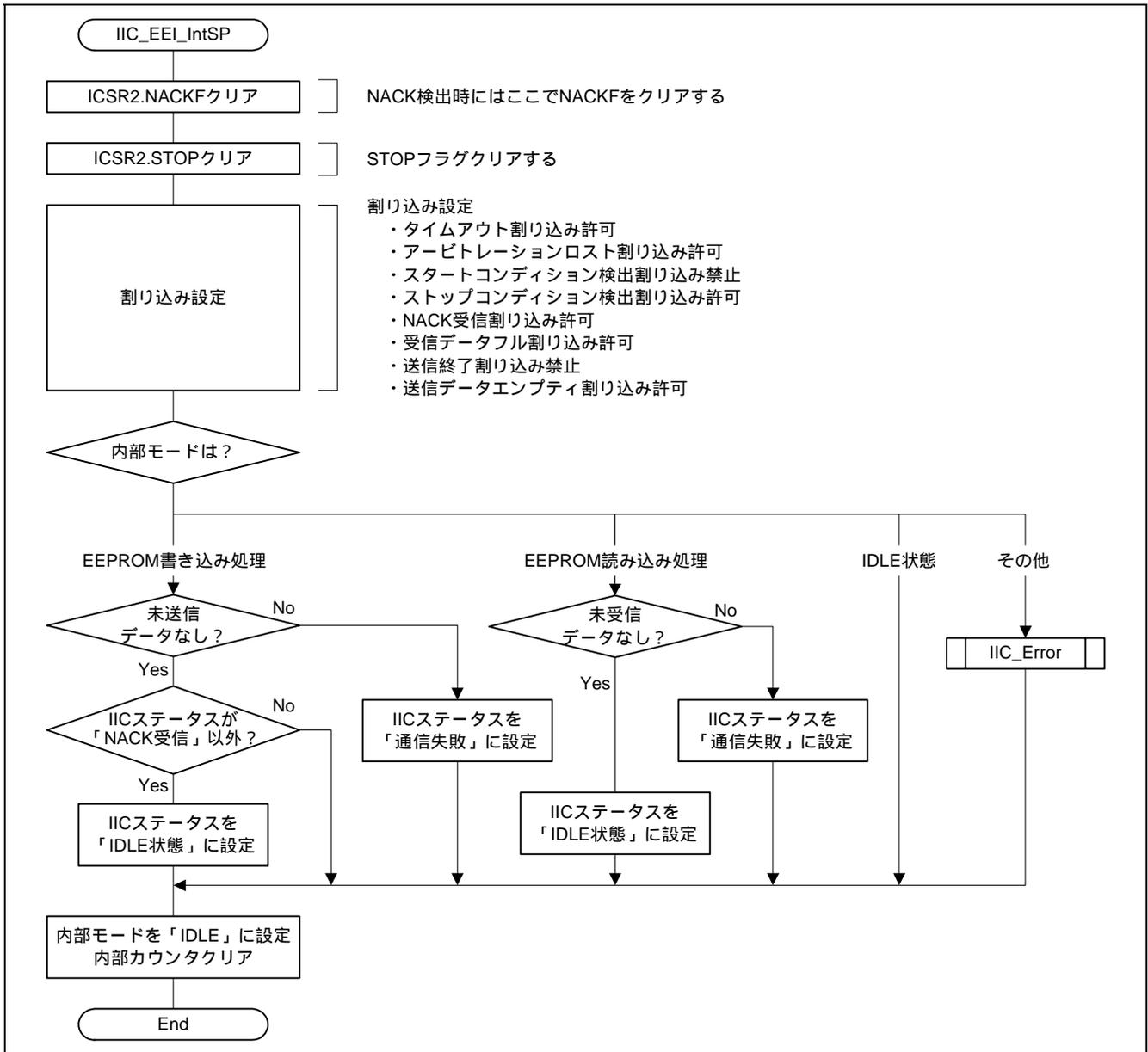


図 22 ストップコンディション検出割り込み

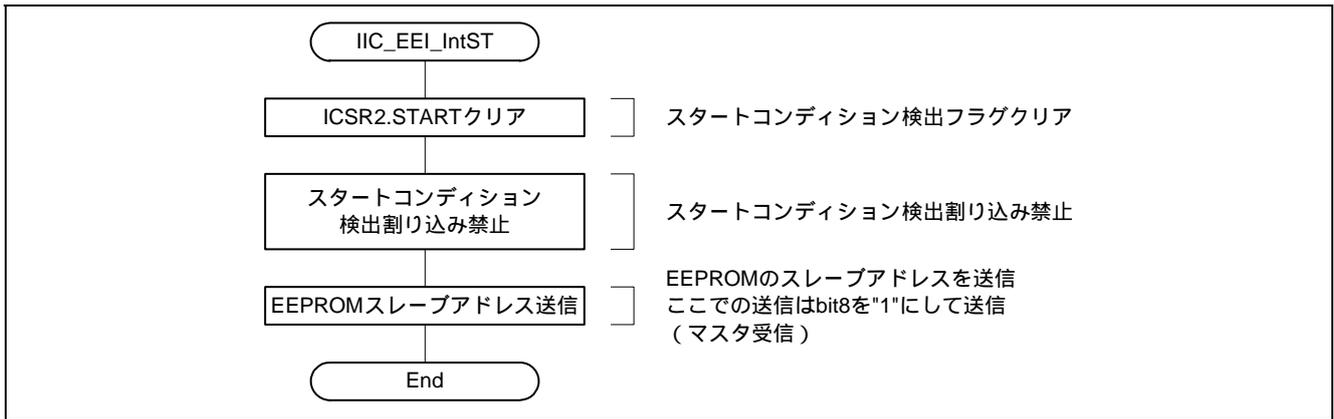


図 23 スタートコンディション検出割り込み

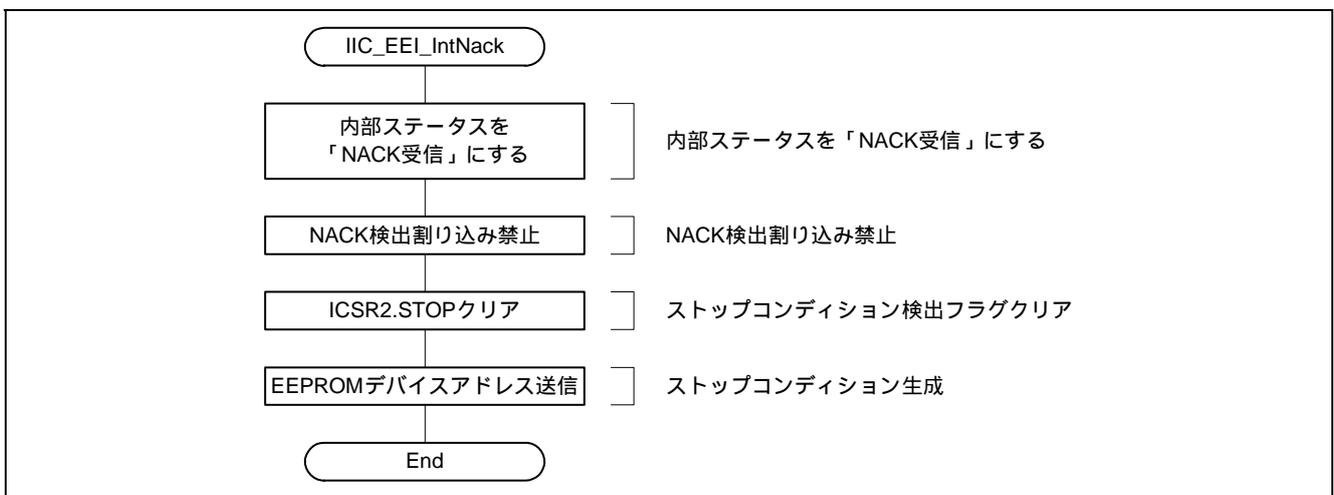


図 24 NACK 検出割り込み

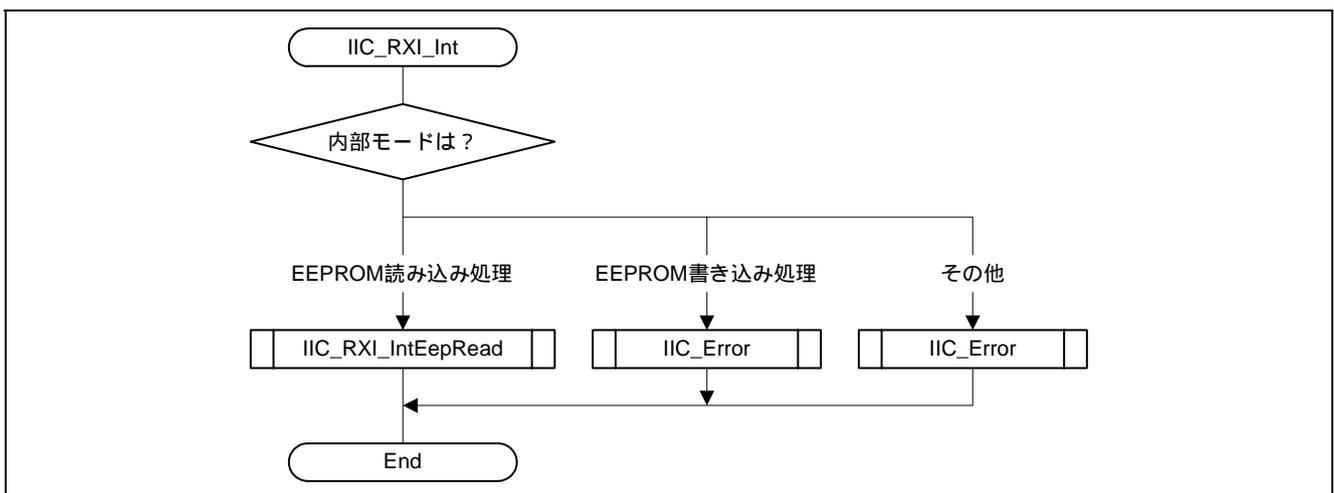


図 25 受信データフル割り込み

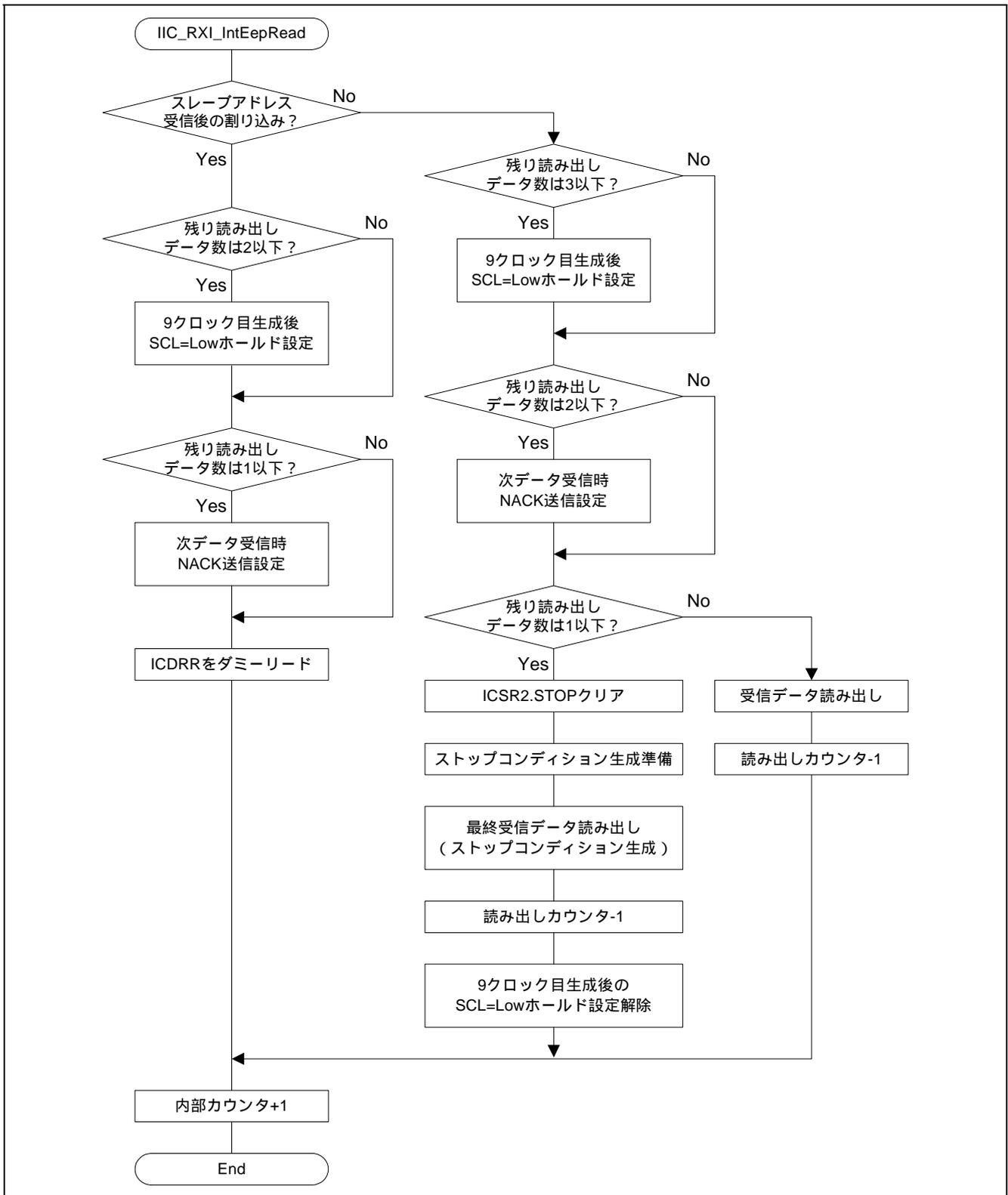


図 26 EEPROM からの読み込み処理 (マスタ受信部分)

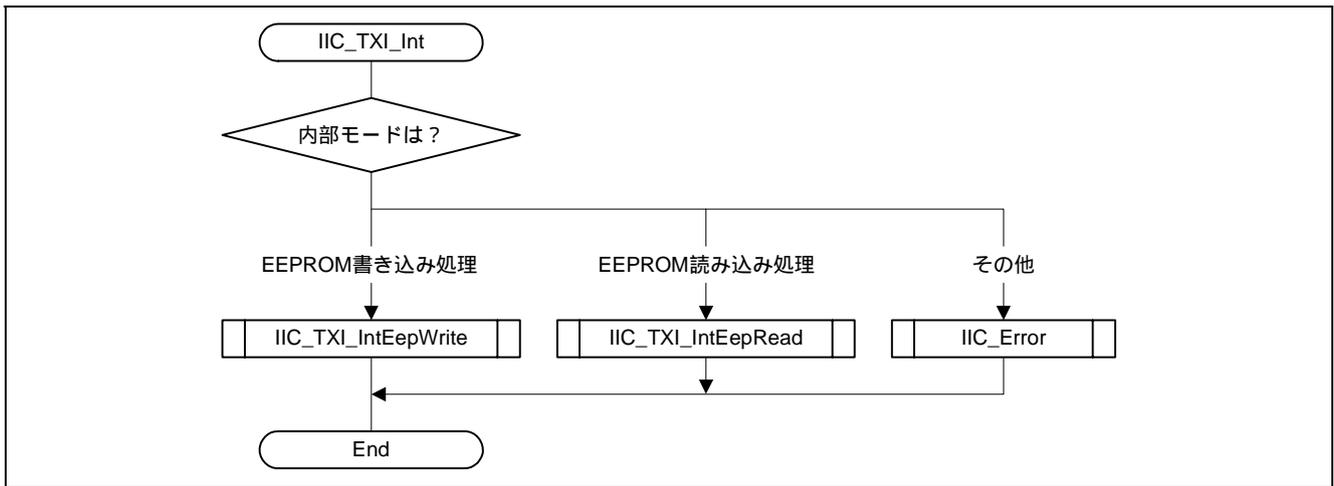


図 27 送信データエンpty割り込み

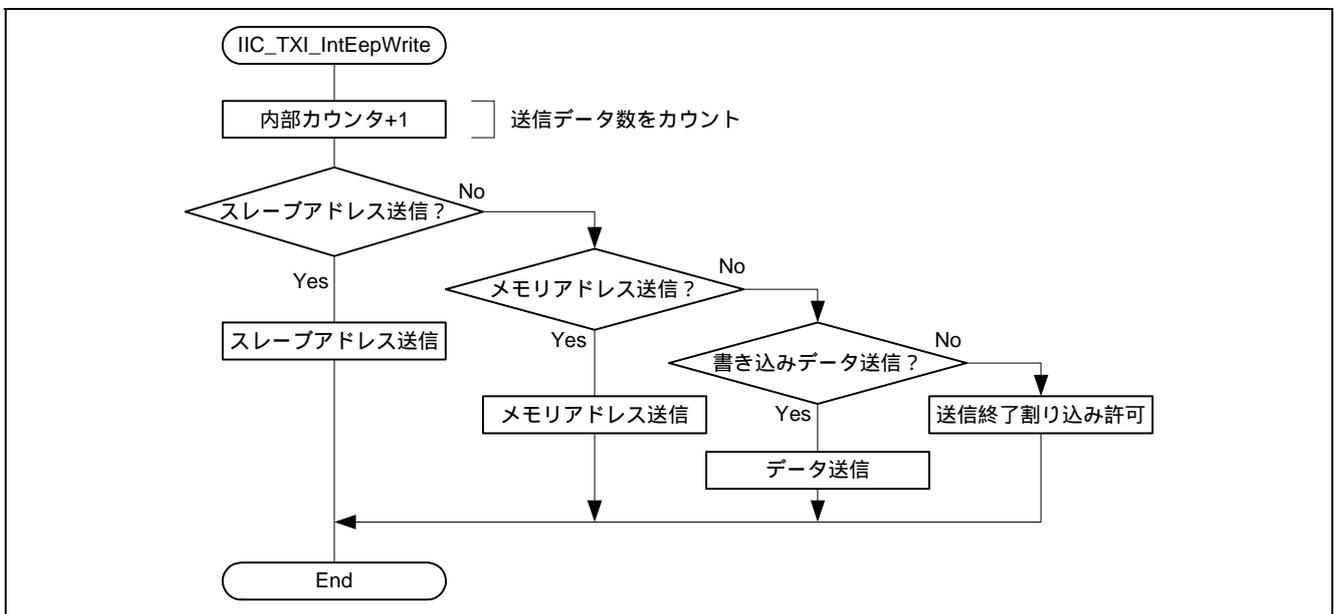


図 28 EEPROM への書き込み処理

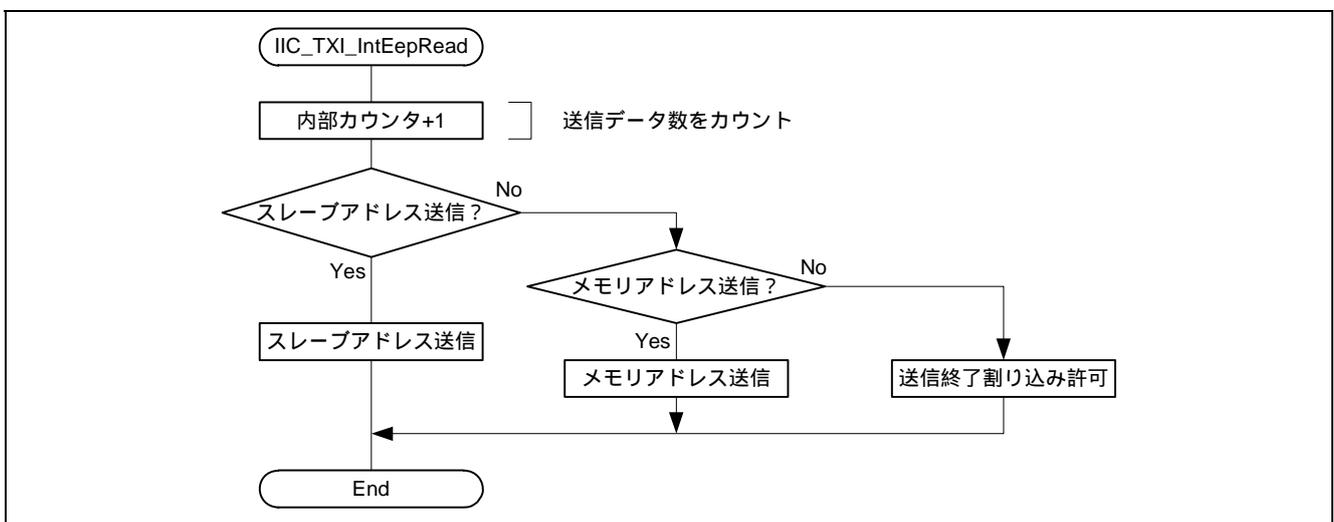


図 29 EEPROM からの読み込み処理 (マスタ送信部分)

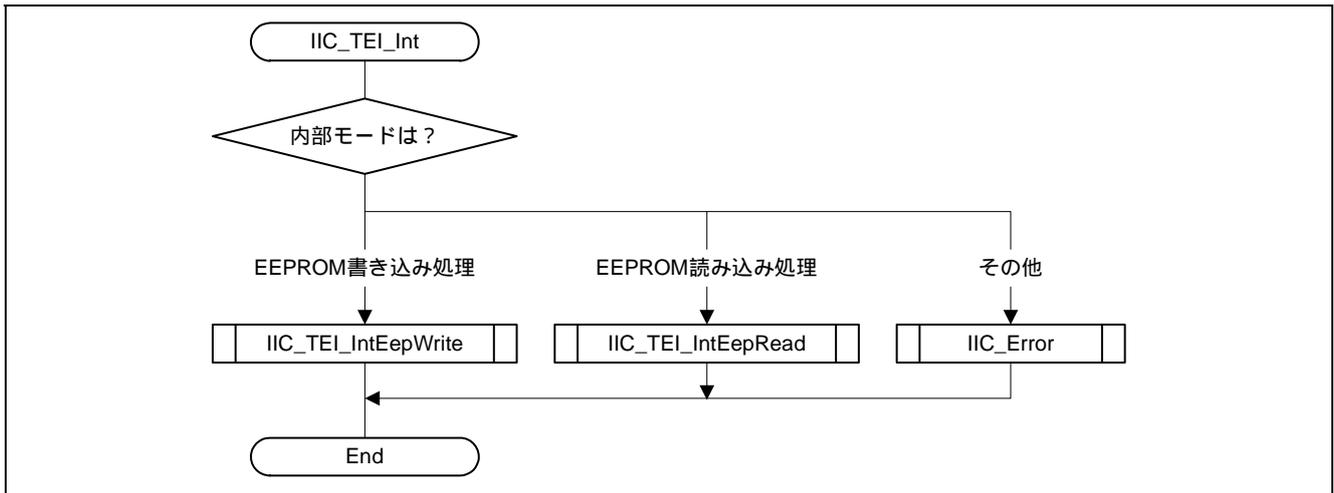


図 30 送信終了割り込み

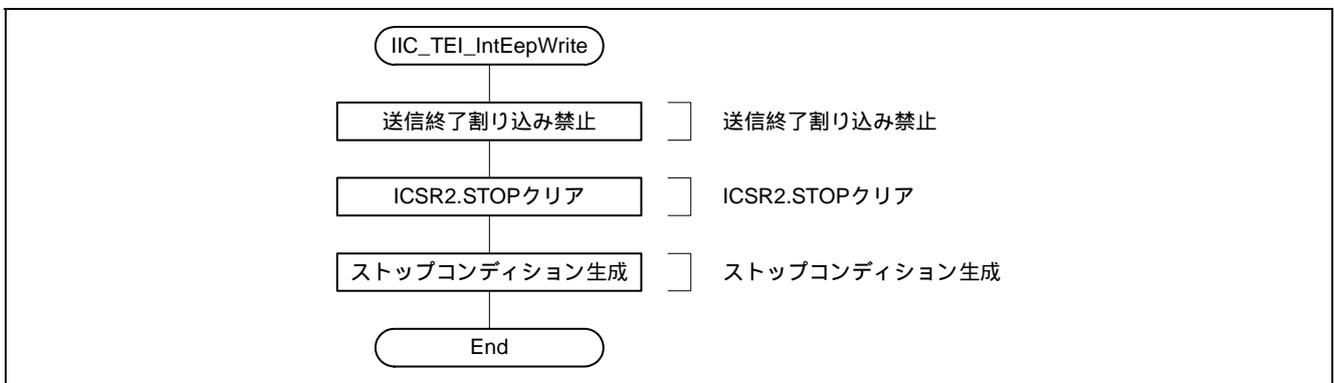


図 31 EEPROM 書き込み処理後の送信終了処理

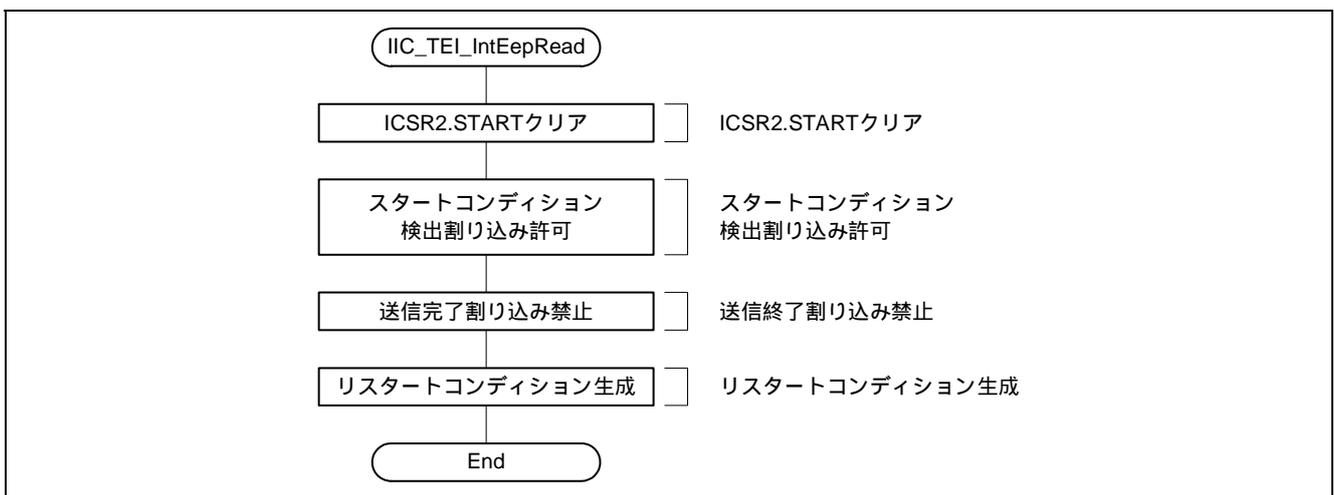
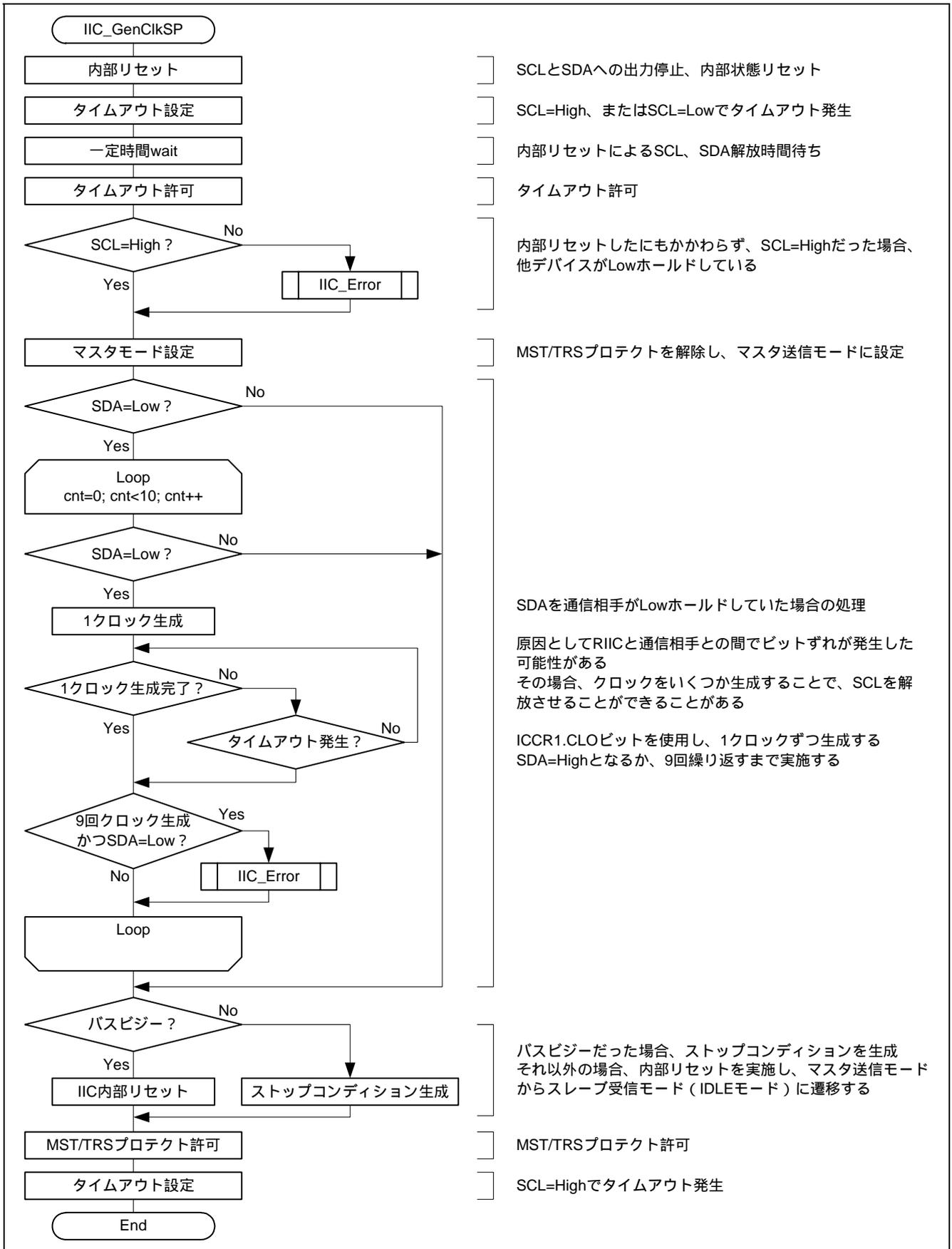


図 32 EEPROM 読み込み処理後の送信終了処理



内部リセットによるSCL、SDA解放時間待ち

SCL=High、またはSCL=Lowでタイムアウト発生

内部リセットによるSCL、SDA解放時間待ち

タイムアウト許可

内部リセットしたにもかかわらず、SCL=Highだった場合、他デバイスがLowホールドしている

MST/TRSプロテクトを解除し、マスタ送信モードに設定

SDAを通信相手がLowホールドしていた場合の処理

原因としてRIICと通信相手との間でビットずれが発生した可能性がある
 その場合、クロックをいくつか生成することで、SCLを解放させることができることがある

ICCR1.CLOビットを使用し、1クロックずつ生成する
 SDA=Highとなるか、9回繰り返すまで実施する

バスビジーだった場合、ストップコンディションを生成
 それ以外の場合、内部リセットを実施し、マスタ送信モードからスレーブ受信モード (IDLEモード) に遷移する

MST/TRSプロテクト許可

SCL=Highでタイムアウト発生

図 33 異常発生時のストップコンディション生成処理

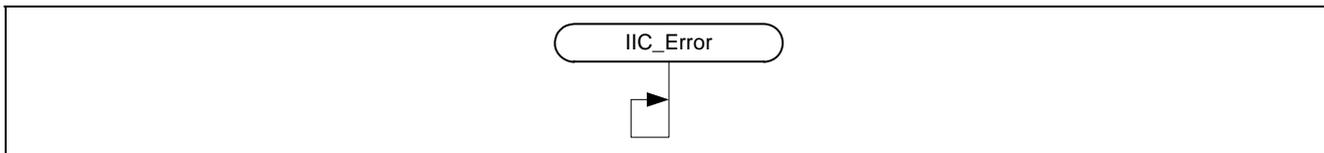


図 34 エラー処理

5. 参考ドキュメント

- ハードウェアマニュアル
RX62T グループ ユーザーズマニュアル ハードウェア編
(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。)
- ソフトウェアマニュアル
RX ファミリ ユーザーズマニュアル ソフトウェア編
(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。)
- 開発環境マニュアル
RX ファミリ用 C/C++コンパイラパッケージ ユーザーズマニュアル
(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
(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/>

お問い合わせ先

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

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 | 2011.07.15 | — | 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. 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. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連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の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている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の目的で使用しないでください。また、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に使用することができません。
6. 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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■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/inquiry>