

RX220 グループ、RX21A グループ

RIIC による EEPROM との通信

R01AN1955JJ0100
Rev.1.00
2014.05.07

要旨

本アプリケーションノートでは、RX220 グループ、RX21A グループの RIIC (I²C バスインターフェース) を用いた EEPROM との通信 (シングルマスタ) について説明します。

対象デバイス

RX220 グループ、RX21A グループ

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	2
2. 動作確認条件	4
3. 関連アプリケーションノート	5
4. ハードウェア説明	6
5. ソフトウェア説明	7
6. サンプルコード	46
7. 使用上のご注意	46
8. 参考ドキュメント	47

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。

表 1.1 に使用する周辺機能と用途、表 1.2 に RIIC 設定、表 1.3 に RX220 で使用する EEPROM 仕様、表 1.4 に RX21A で使用する EEPROM 仕様、図 1.1 に RX220 接続図、図 1.2 に RX21A 接続図を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
RIIC	マスタ送信、マスタ受信

表 1.2 RIIC 設定

項目	内容
動作周波数	内部標準クロック(IIC): 20MHz
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"> SCLn 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(IIC):約 3.277ms)

表 1.3 RX220 で使用する EEPROM 仕様

項目	内容
型名	R1EX24016ASAS0A
容量	16K (2-kword × 8-bit)
スレーブアドレス	スレーブアドレス: 1010xxxx bit0 は R/W ビット、bit1 ~ bit3 は EEPROM 上の page を示します。 詳しくは EEPROM の仕様書をご確認ください。
ライトプロテクト	常に解除 <ul style="list-style-type: none"> WP 端子: Low

表 1.4 RX21A で使用する 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: 1010xxxx bit0 は R/W ビット、bit1 ~ bit2 は EEPROM 上の page を示します。 詳しくは EEPROM の仕様書をご確認ください。
ライトプロテクト	常に解除 <ul style="list-style-type: none"> WP 端子: Low

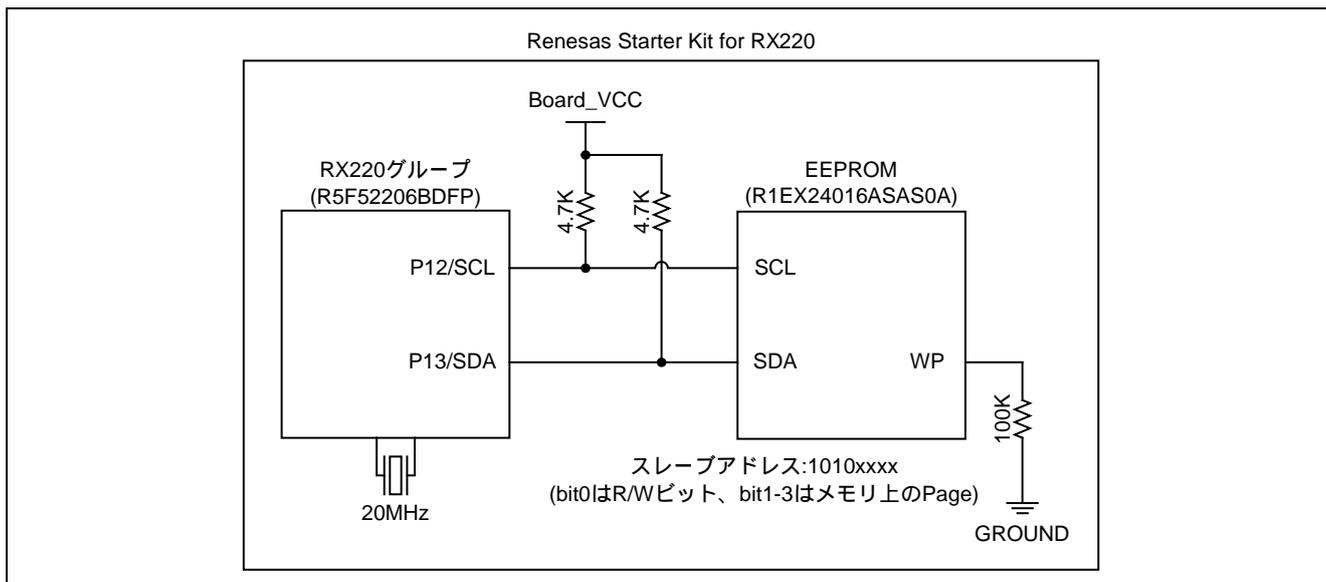


図 1.1 RX220 接続図

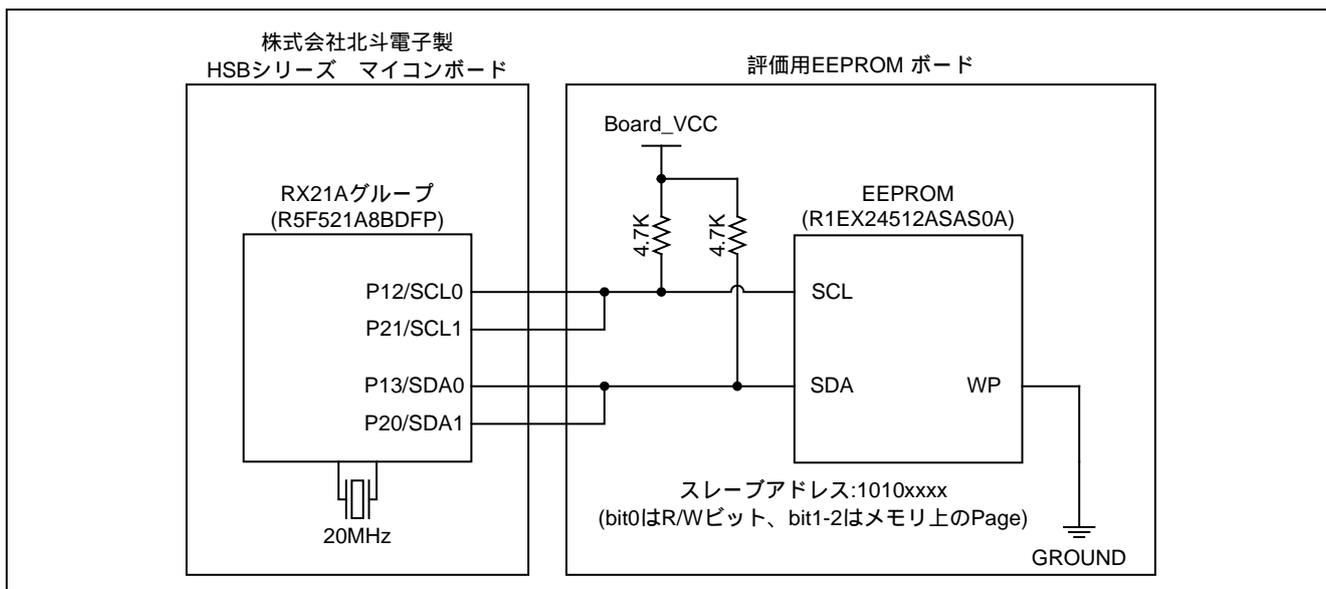


図 1.2 RX21A 接続図

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 RX220 の動作確認条件

項目	内容
使用マイコン	R5F52206BDFP(RX220 グループ)
動作周波数	<ul style="list-style-type: none"> ● メインクロック: 20MHz ● システムクロック(ICLK): 20MHz (メインクロック 1 分周) ● 周辺モジュールクロック B (PCLKB): 20MHz (メインクロック 1 分周) ● 外部バスクロック(BCLK): 20MHz(メインクロック 1 分周)
動作電圧	5.0V(E1 エミュレータから供給)
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.01.007
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=rx200 -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	Version1.0A
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 1.00
使用ボード	Renesas Starter Kit for RX220 (製品型名: R0K505220S000BE)

表 2.2 RX21A の動作確認条件

項目	内容
使用マイコン	R5F521A8BDFP (RX21A グループ)
動作周波数	<ul style="list-style-type: none"> メインクロック: 20MHz システムクロック (ICLK): 20MHz (メインクロック 1 分周) 周辺モジュールクロック B (PCLKB): 20MHz (メインクロック 1 分周) 外部バスクロック (BCLK): 20MHz (メインクロック 1 分周)
動作電圧	3.3V (E1 エミュレータから供給)
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.01.007
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=rx200 -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 1.0
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 1.00
使用ボード	株式会社北斗電子製 HSB シリーズ マイコンボード (製品型名:HSBRX21AP-B) 評価用 EEPROM ボード (北斗電子製 HSB シリーズ マイコンボードには EEPROM が接続されていません。動作を確認する際には、EEPROM を別途ご用意ください。)

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RX220 グループ 初期設定例 Rev.1.00(R01AN1494JJ0100_RX220)
- RX21A グループ 初期設定例 Rev.1.00(R01AN1486JJ0100_RX21A)

上記アプリケーションノートの初期設定関数を、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。

本アプリケーションノートのサンプルコードでは、RX21A のクロック設定をデフォルトから変更していません。(システムクロックを PLL からメインクロックに変更)

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 使用端子一覧

表 4.1、表 4.2にRX220 の使用端子と機能、RX21A の使用端子と機能を示します。

表 4.1 RX220 の使用端子と機能

端子名	入出力	内容
P12/SCL	入出力	RIIC0 シリアルクロック入出力
P13/SDA	入出力	RIIC0 シリアルデータ入出力

表 4.2 RX21A の使用端子と機能

端子名	入出力	内容
P12/SCL0	入出力	RIIC0 シリアルクロック入出力
P13/SDA0	入出力	RIIC0 シリアルデータ入出力
P21/SCL1	入出力	RIIC1 シリアルクロック入出力
P20/SDA1	入出力	RIIC1 シリアルデータ入出力

5. ソフトウェア説明

5.1 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、バスを開放します。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 5.1 に EEPROM への書き込み信号例を示します。

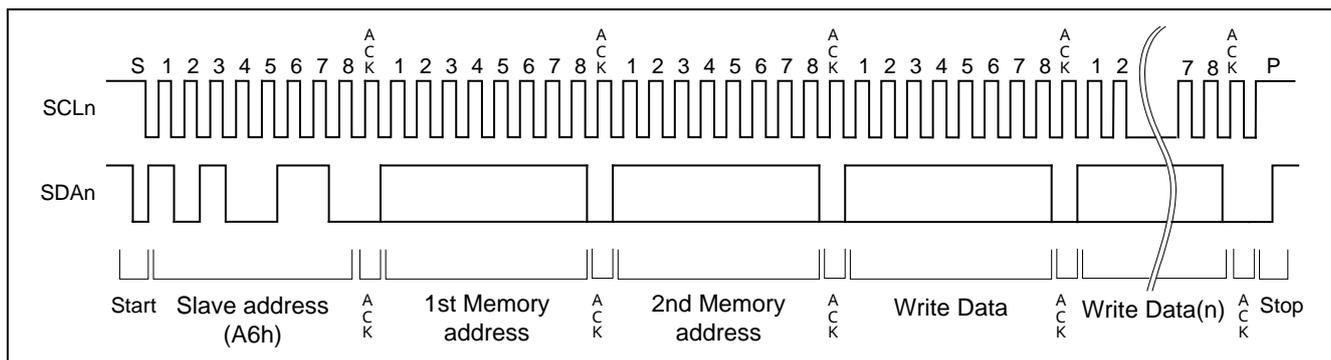


図 5.1 EEPROM への書き込み信号例

5.2 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 5.2 に EEPROM からの読み込み信号例を示します。

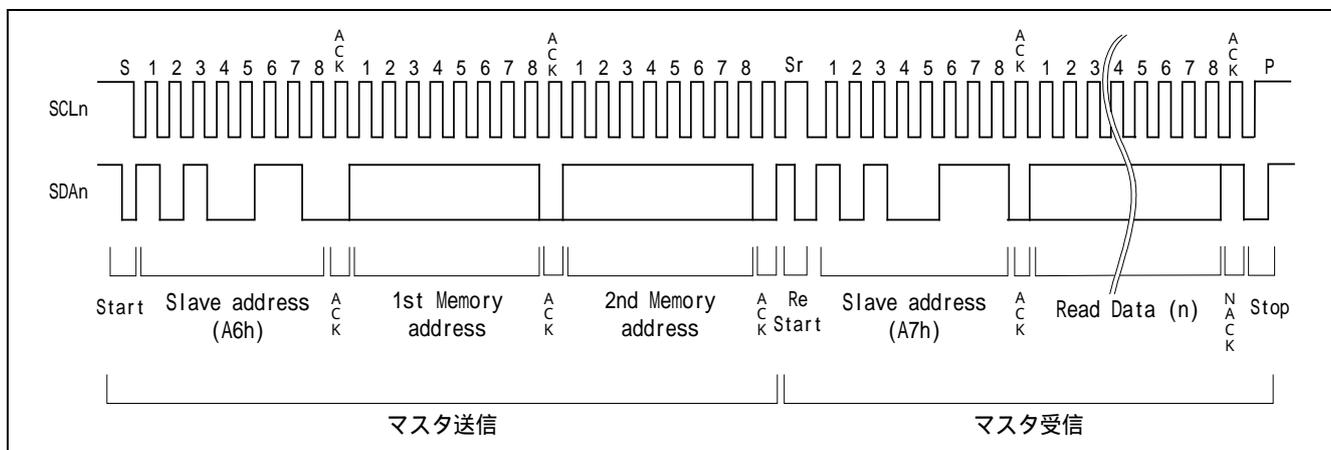


図 5.2 EEPROM からの読み込み信号例

5.3 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 5.3にAcknowledge Polling 信号例を示します。

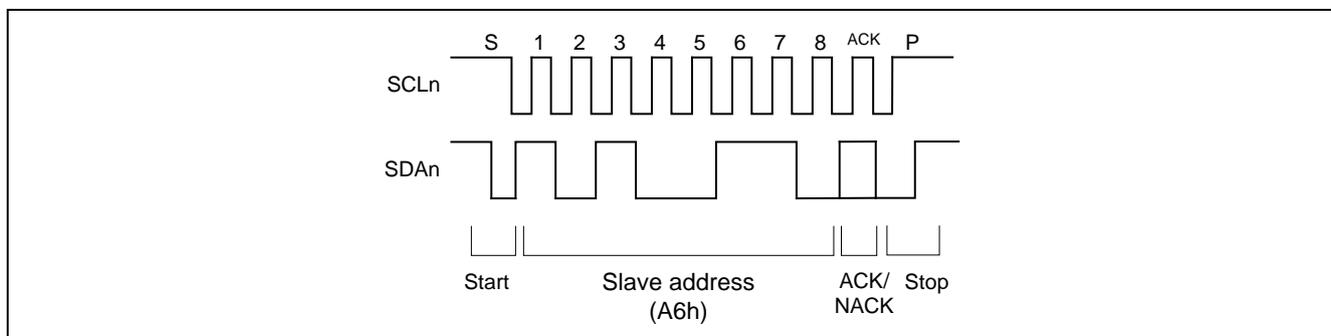


図 5.3 Acknowledge Polling 信号例

5.4 ファイル構成

表 5.1にサンプルコードで使用するファイルを示します。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 5.1 サンプルコードで使用するファイル

対象デバイス	ファイル名	概要	備考
共通	main.c	メイン処理	
	iic_eeprom.c	シングルマスタ送受信	
	iic_eeprom.h	シングルマスタ送受信のヘッダファイル	
	iic_eeprom_cfg.h	シングルマスタ送受信のコンフィグレーションヘッダファイル	
RX220	r_init_stop_module.c	リセット後に動作している周辺機能の停止	
	r_init_stop_module.h	r_init_stop_module.c のヘッダファイル	
	r_init_non_existent_port.c	存在しないポートの初期設定	
	r_init_non_existent_port.h	r_init_non_existent_port.c のヘッダファイル	
	r_init_clock.c	クロック初期設定	
	r_init_clock.h	r_init_clock.c のヘッダファイル	
RX21A	r_init_stop_module.c	リセット後に動作している周辺機能の停止	
	r_init_stop_module.h	r_init_stop_module.c のヘッダファイル	
	r_init_non_existent_port.c	存在しないポートの初期設定	
	r_init_non_existent_port.h	r_init_non_existent_port.c のヘッダファイル	
	r_init_clock.c	クロック初期設定	
	r_init_clock.h	r_init_clock.c のヘッダファイル	

5.5 オプション設定メモリ

表 5.2 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 5.2 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、IWDTC は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh ~ FFFF FF88h	FFFF FFFFh	リセット後、電圧監視 0 リセット無効 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 5.3、表 5.4、表 5.5 にサンプルコードで使用する定数を示します。

表 5.3 サンプルコードで使用する定数(main.c)

定数名	設定値	内容
TARGET_SALVE_ADDRESS	0xA6	EEPROM アドレス
EEPROM_ADDRESS_LENGTH	2u	EEPROM アドレス長

表 5.4 サンプルコードで使用する定数(iic_eeprom_cfg.h で RIIC0 チャンネル選択時)

定数名	設定値	内容
RIICn	RIIC0	RIIC チャンネル:RIIC0
MSTP_RIICn	MSTP(RIIC0)	RIIC0 モジュールストップ設定ビット
IEN_RIICn_EEIn	IEN(RIIC0,EEI0)	RIIC0.EEI0 割り込み要求許可ビット
IEN_RIICn_RXIn	IEN(RIIC0,RXI0)	RIIC0.RXI0 割り込み要求許可ビット
IEN_RIICn_TXIn	IEN(RIIC0,TXI0)	RIIC0.TXI0 割り込み要求許可ビット
IEN_RIICn_TEln	IEN(RIIC0,TEI0)	RIIC0.TEI0 割り込み要求許可ビット
IPR_RIICn_EEIn	IPR(RIIC0,EEI0)	RIIC0.EEI0 割り込み優先レベル設定ビット
IPR_RIICn_RXIn	IPR(RIIC0,RXI0)	RIIC0.RXI0 割り込み優先レベル設定ビット
IPR_RIICn_TXIn	IPR(RIIC0,TXI0)	RIIC0.TXI0 割り込み優先レベル設定ビット
IPR_RIICn_TEln	IPR(RIIC0,TEI0)	RIIC0.TEI0 割り込み優先レベル設定ビット
IR_RIICn_EEIn	IR(RIIC0,EEI0)	RIIC0.EEI0 割り込みステータスフラグ
IR_RIICn_RXIn	IR(RIIC0,RXI0)	RIIC0.RXI0 割り込みステータスフラグ
IR_RIICn_TXIn	IR(RIIC0,TXI0)	RIIC0.TXI0 割り込みステータスフラグ
IR_RIICn_TEln	IR(RIIC0,TEI0)	RIIC0.TEI0 割り込みステータスフラグ
SCLn_RIICn_PDR	PORT1.PDR.BIT.B2	P12 端子方向制御ビット
SDAn_RIICn_PDR	PORT1.PDR.BIT.B3	P13 端子方向制御ビット
SCLn_RIICn_PMR	PORT1.PMR.BIT.B2	P12 端子モード制御ビット
SDAn_RIICn_PMR	PORT1.PMR.BIT.B3	P13 端子モード制御ビット
SCLn_RIICn_PFS	MPC.P12PFS.BIT.PSEL	P12 端子機能制御レジスタ
SDAn_RIICn_PFS	MPC.P13PFS.BIT.PSEL	P13 端子機能制御レジスタ
PSEL_SETTING	(0x0F)	端子機能選択ビット設定値: SCL0、SDA0

表 5.5 サンプルコードで使用する定数(r_iic_eeprom_cfg.h で IIC1 チャンネル選択時)

定数名	設定値	内容
RIICn	RIIC1	IIC チャンネル:RIIC1
MSTP_RIICn	MSTP(RIIC1)	RIIC1 モジュールストップ設定ビット
IEN_RIICn_EEIn	IEN(RIIC1,EEI1)	RIIC1.EEI1 割り込み要求許可ビット
IEN_RIICn_RXIn	IEN(RIIC1,RXI1)	RIIC1.RXI1 割り込み要求許可ビット
IEN_RIICn_TXIn	IEN(RIIC1,TXI1)	RIIC1.TXI1 割り込み要求許可ビット
IEN_RIICn_TEln	IEN(RIIC1,TEI1)	RIIC1.TEI1 割り込み要求許可ビット
IPR_RIICn_EEIn	IPR(RIIC1,EEI1)	RIIC1.EEI1 割り込み優先レベル設定ビット
IPR_RIICn_RXIn	IPR(RIIC1,RXI1)	RIIC1.RXI1 割り込み優先レベル設定ビット
IPR_RIICn_TXIn	IPR(RIIC1,TXI1)	RIIC1.TXI1 割り込み優先レベル設定ビット
IPR_RIICn_TEln	IPR(RIIC1,TEI1)	RIIC1.TEI1 割り込み優先レベル設定ビット
IR_RIICn_EEIn	IR(RIIC1,EEI1)	RIIC1.EEI1 割り込みステータスフラグ
IR_RIICn_RXIn	IR(RIIC1,RXI1)	RIIC1.RXI1 割り込みステータスフラグ
IR_RIICn_TXIn	IR(RIIC1,TXI1)	RIIC1.TXI1 割り込みステータスフラグ
IR_RIICn_TEln	IR(RIIC1,TEI1)	RIIC1.TEI1 割り込みステータスフラグ
SCLn_RIICn_PDR	PORT2.PDR.BIT.B1	P21 端子方向制御ビット
SDAn_RIICn_PDR	PORT2.PDR.BIT.B0	P20 端子方向制御ビット
SCLn_RIICn_PMR	PORT2.PMR.BIT.B1	P21 端子モード制御ビット
SDAn_RIICn_PMR	PORT2.PMR.BIT.B0	P20 端子モード制御ビット
SCLn_RIICn_PFS	MPC.P21PFS.BIT.PSEL	P21 端子機能制御レジスタ
SDAn_RIICn_PFS	MPC.P20PFS.BIT.PSEL	P20 端子機能制御レジスタ
PSEL_SETTING	(0x0F)	端子機能選択ビット設定値: SCL1、SDA1

5.7 構造体/共用体一覧

図 5.4 に関数 IIC_EeWrite と関数 IIC_RandomRead の引数用構造体を示します。また、表 5.6 に構造体 "iic_api_t" のメンバー一覧を示します。

```

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    *pRWDData;    /* Pointer for Data buffer */
};
typedef struct str_iic_api_t iic_api_t;

```

図 5.4 関数 IIC_EeWrite と関数 IIC_RandomRead の引数用構造体

表 5.6 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。

5.8 変数一覧

表 5.7にstatic 型変数を示します。

表 5.7 static 型変数

型	変数名	内容	使用関数
static uint8_t	trm_eeprom_adr[EEPROM_ADDRESS_LENGTH]	EEPROM スレーブアドレス格納バッファ(書き込み時)	SampleEepromWrite
static uint8_t	rcv_eeprom_adr[EEPROM_ADDRESS_LENGTH]	EEPROM スレーブアドレス格納バッファ(読み込み時)	SampleEepromRead
static uint8_t	trm_buff[256]	送信データバッファ	SampleEepromWrite
static uint8_t	rcv_buff[256]	受信データバッファ	SampleEepromRead
static iic_api_t	iic_buff_prm[2]	関数 IIC_EeWrite と関数 IIC_RandomRead の引数用構造体	SampleEepromWrite SampleEepromRead
static iic_api_t	iic_buff	関数 IIC_EeWrite と関数 IIC_RandomRead の引数用構造体 (関数 IIC_EeWrite と関数 IIC_RandomRead 兼用)	IIC_EeWrite IIC_RandomRead iic_eei_int_sp iic_eei_int_st iic_rxi_int_eeread iic_txi_int_eewrite iic_txi_int_eeread
static enum riic_internal_mode_t	iic_mode	内部モード	IIC_Create IIC_EeWrite IIC_RandomRead iic_eei_int_sp IIC_RXI_interrupt IIC_TXI_interrupt IIC_TEI_interrupt
static enum riic_status_t	iic_status	IIC ステータス	IIC_Create IIC_EeWrite IIC_RandomRead IIC_GetStatus iic_eei_int_sp iic_eei_int_nack
static unit32_t	iic_trm_cnt	内部 IIC 送信カウンタ	IIC_Create IIC_EeWrite IIC_RandomRead iic_eei_int_sp iic_txi_int_eewrite iic_txi_int_eeread
static unit32_t	iic_rcv_cnt	内部 IIC 受信カウンタ	IIC_Create IIC_RandomRead iic_eei_int_sp iic_rxi_int_eeread

5.9 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_EeWrite”と関数”IIC_RandomRead”のリターン値は列挙型で宣言されます。IIC ステータス一覧とIIC ステータス遷移図をそれぞれ表 5.8、図 5.5 に示します。また、IIC バスステータスを表 5.9 に、内部モード一覧を表 5.10 に、関数”IIC_EeWrite”と関数”IIC_RandomRead”のリターン値を表 5.11 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 5.8 IIC ステータス一覧(enum riic_status_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_EeWrite 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。

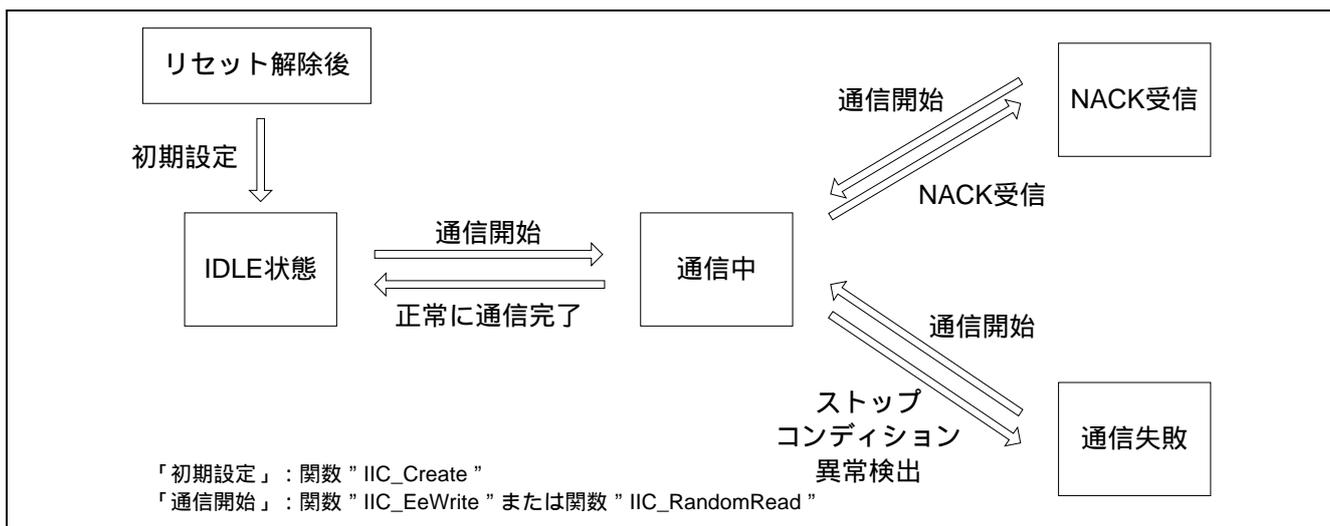


図 5.5 IIC ステータス遷移図

表 5.9 IIC バスステータス(enum riic_bus_status_t)

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

表 5.10 内部モード一覧(enum riic_internal_mode_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_EE_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_EE_WRITE	EEPROM ライトモード。 関数 IIC_EeWrite で通信を開始するとこのモードに遷移します。

表 5.11 関数"IIC_EeWrite"と関数"IIC_RandomRead"のリターン値(enum riic_ee_fnc_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。)

5.10 関数一覧

表 5.12に関数を示します。

表 5.12 関数

関数名	概要
main	メイン処理
R_INIT_StopModule	リセット後に動作している周辺機能の停止
R_INIT_NonExistentPort	存在しないポートの初期設定
R_INIT_Clock	クロック初期設定
port_init	ポート初期設定
peripheral_init	周辺初期設定
CpuIntCreate	CPU 割り込み設定
SampleEepromWrite	EEPROM への書き込み処理例
SampleEepromRead	EEPROM からの読み込み処理例
IICAckPolling	Acknowledge Polling
IIC_Create	IIC 処理設定
IIC_Destroy	IIC 終了処理
IIC_EeWrite	EEPROM への書き込み開始処理
IIC_RandomRead	EEPROM からの読み込み開始処理
IIC_GetStatus	IIC のステータスチェック
IIC_EEI_interrupt	通信エラー/イベント発生割り込み
iic_eei_int_timeout	タイムアウト検出割り込み
iic_eei_int_al	アービトラクションロスト検出割り込み
iic_eei_int_sp	ストップコンディション検出割り込み
iic_eei_int_st	スタートコンディション検出割り込み
iic_eei_int_nack	NACK 検出割り込み
IIC_RXI_interrupt	受信データフル割り込み
iic_rxi_int_eeread	EEPROM からの読み込み処理(マスタ受信部分)
IIC_TXI_interrupt	送信データエンプティ割り込み
iic_txi_int_eewrite	EEPROM への書き込み処理
iic_txi_int_eeread	EEPROM からの読み込み処理(マスタ送信部分)
IIC_TEI_interrupt	送信終了割り込み
iic_tei_int_eewrite	EEPROM 書き込み処理後の送信終了処理
iic_tei_int_eeread	EEPROM 読み込み処理後の送信終了処理
iic_gen_clk_sp	異常発生時のストップコンディション生成処理
iic_error	エラー処理
Excep_RIICn_EEIn	RIICn.EEIn 割り込み処理
Excep_RIICn_RXIn	RIICn.RXIn 割り込み処理
Excep_RIICn_TXIn	RIICn.TXIn 割り込み処理
Excep_RIICn_TEin	RIICn.TEin 割り込み処理

5.11 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、ポート初期設定、周辺機能初期設定を行います。その後、EEPROM への書き込み処理例、Acknowledge Polling、EEPROM からの読み込み処理例、IIC 処理終了の関数を呼び出します。
引数	なし
リターン値	なし
R_INIT_StopModule	
概要	リセット後に動作している周辺機能の停止
ヘッダ	r_init_stop_module.h
宣言	void R_INIT_StopModule(void)
説明	モジュールストップ状態に遷移する設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、モジュールストップ状態への遷移は行っていません。 本関数の詳細は、アプリケーションノート「RX220 グループ 初期設定 Rev.1.00」、アプリケーションノート「RX21A グループ 初期設定例 Rev.1.00」を参照してください。
R_INIT_NonExistentPort	
概要	存在しないポートの初期設定
ヘッダ	r_init_non_existent_port.h
宣言	void R_INIT_NonExistentPort(void)
説明	100 ピン未満の製品に対して、存在しないポートに端子に対応するポート方向レジスタ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、100 ピン版(PIN_SIZE=100)に設定しています。 本関数をコールした後に、存在しないポートを含む PDR、PODR レジスタへバイト単位で書き込む場合、存在しないポートの方向制御ビットには“1”、ポート出力データ格納ビットには“0”を設定してください。 本関数の詳細は、アプリケーションノート「RX220グループ 初期設定例 Rev.1.00」、アプリケーションノート「RX21A グループ 初期設定例 Rev.1.00」を参照してください。

R_INIT_Clock	
概要	クロック初期設定
ヘッダ	r_init_clock.h
宣言	void R_INIT_Clock(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メインクロックとし、サブクロックを使用しない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「RX220グループ 初期設定例 Rev.1.00」、アプリケーションノート「RX21A グループ 初期設定例 Rev.1.00」を参照してください。

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

peripheral_init	
概要	周辺機能初期設定
ヘッダ	なし
宣言	static void peripheral_init(void)
説明	使用する周辺機能の初期設定を行います。
引数	なし
リターン値	なし

CpuIntCreate	
概要	CPU 割り込み設定
ヘッダ	なし
宣言	void CpuIntCreate(void)
説明	RIIC の CPU 割り込みの初期設定を行います。
引数	なし
リターン値	なし

SampleEepromWrite	
概要	EEPROM への書き込み処理例
ヘッダ	iic_eeprom.h
宣言	static void SampleEepromWrite(void)
説明	マスタ送信を使用して EEPROM への書き込みを行います。
引数	なし
リターン値	なし

SampleEepromRead

概要	EEPROM からの読み込み処理例
ヘッダ	iic_eeprom.h
宣言	static void SampleEepromRead(void)
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。
引数	なし
リターン値	なし

IICAckPolling

概要	Acknowledge Polling
ヘッダ	iic_eeprom.h
宣言	bool IICAckPolling(uint8_t in_addr1, uint8_t in_num, uint32_t in_len)
説明	EEPROM が書き換え中か否かを判断します。
引数	uint8_t in_addr1 :スレーブアドレス(0x00-0xFE) uint8_t in_num :Acknowledge Polling の繰り返し回数 uint32_t in_len :Acknowledge Polling 間のインターバル時間の長さ
リターン値	true :ACK 応答あり false :ACK 応答なし

IIC_Create

概要	IIC 処理設定
ヘッダ	iic_eeprom.h
宣言	void IIC_Create(void)
説明	RIIC の初期設定、転送ビットレート設定、割り込み設定、タイムアウト設定を行います。
引数	なし
リターン値	なし

IIC_Destroy

概要	IIC 終了処理
ヘッダ	なし
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

IIC_EeWrite

概要	EEPROM への書き込み開始処理	
ヘッダ	iic_eeprom.h	
宣言	enum riic_ee_fnc_t IIC_EeWrite(iic_api_t data1)	
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	:スレーブアドレス メモリアドレスカウンタ メモリアドレス格納バッファポインタ データカウンタ データ格納バッファポインタ
リターン値	RIIC_OK	:正常に通信が開始された場合
	RIIC_BUS_BUSY	:I ² C バスがビジーだった場合
	RIIC_MODE_ERROR	:RIIC が通信中の場合

IIC_RandomRead

概要	EEPROM からの読み込み開始処理	
ヘッダ	iic_eeprom.h	
宣言	enum riic_ee_fnc_t IIC_RandomRead(iic_api_t data1)	
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	:スレーブアドレス メモリアドレスカウンタ メモリアドレス格納バッファポインタ データカウンタ データ格納バッファポインタ
リターン値	RIIC_OK	:正常に通信が開始された場合
	RIIC_BUS_BUSY	:I ² C バスがビジーだった場合
	RIIC_MODE_ERROR	:RIIC が通信中の場合
	RIIC_PRM_ERROR	:引数の値が正しくなかった場合

IIC_GetStatus

概要	IIC のステータスチェック	
ヘッダ	iic_eeprom.h	
宣言	void IIC_GetStatus(enum riic_status_t *data1, enum riic_bus_status_t *data2)	
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 riic_status_t *data1	:IIC ステータス
	Enum riic_bus_status_t *data2	:IIC バスステータス
リターン値	なし	

IIC_EEI_interrupt	
概要	通信エラー/イベント発生割り込み
ヘッダ	なし
宣言	void IIC_EEI_interrupt(void)
説明	検出した割り込み要求の関数を呼び出します。
引数	なし
リターン値	なし

iic_eei_int_timeout	
概要	タイムアウト検出割り込み
ヘッダ	なし
宣言	static void iic_eei_int_timeout(void)
説明	異常発生時のストップコンディション生成処理関数を呼び出します。
引数	なし
リターン値	なし

iic_eei_int_al	
概要	アービトレーションロスト検出割り込み
ヘッダ	なし
宣言	static void iic_eei_int_al(void)
説明	異常発生時のストップコンディション生成処理関数を呼び出します。
引数	なし
リターン値	なし

iic_eei_int_sp	
概要	ストップコンディション検出割り込み
ヘッダ	iic_eeprom.h
宣言	static void iic_eei_int_sp(void)
説明	通信を終了し、内部モードを「IDLE」にします。
引数	なし
リターン値	なし

iic_eei_int_st	
概要	スタートコンディション検出割り込み
ヘッダ	iic_eeprom.h
宣言	static void iic_eei_int_st(void)
説明	EEPROM のスレーブアドレスを送信します。本割り込みは、EEPROM からデータを読み出す時のみ使用します。
引数	なし
リターン値	なし

iic_eei_int_nack	
概要	NACK 検出割り込み
ヘッダ	iic_eeprom.h
宣言	static void iic_eei_int_nack(void)
説明	ストップコンディションの発行を要求します。
引数	なし
リターン値	なし

IIC_RXI_interrupt	
概要	受信データフル割り込み
ヘッダ	iic_eeprom.h
宣言	void IIC_RXI_interrupt(void)
説明	内部モードを確認し、関数を呼び出します。
引数	なし
リターン値	なし

iic_rxi_int_eeread	
概要	EEPROM からの読み込み処理(マスタ受信部分)
ヘッダ	iic_eeprom.h
宣言	static void iic_rxi_int_eeread(void)
説明	受信データをバッファに格納します。
引数	なし
リターン値	なし

IIC_TXI_interrupt	
概要	送信データエンプティ割り込み
ヘッダ	iic_eeprom.h
宣言	void IIC_TXI_interrupt(void)
説明	内部モードを確認し、関数を呼び出します。
引数	なし
リターン値	なし

iic_txi_int_eewrite	
概要	EEPROM への書き込み処理
ヘッダ	iic_eeprom.h
宣言	static void iic_txi_int_eewrite(void)
説明	スレーブアドレス、メモリアドレス、書き込みデータを順に送信します。
引数	なし
リターン値	なし

iic_txi_int_eeread	
概要	EEPROM からの読み込み処理(マスタ送信部分)
ヘッダ	iic_eeeprom.h
宣言	static void iic_txi_int_eeread(void)
説明	スレーブアドレス、メモリアドレスを順に送信します。
引数	なし
リターン値	なし

IIC_TEI_interrupt	
概要	送信終了割り込み
ヘッダ	iic_eeeprom.h
宣言	void IIC_TEI_interrupt(void)
説明	内部モードを確認し、関数を呼び出します。
引数	なし
リターン値	なし

iic_tei_int_eewrite	
概要	EEPROM 書き込み処理後の送信終了処理
ヘッダ	なし
宣言	static void iic_tei_int_eewrite(void)
説明	ストップコンディションの発行を要求します。
引数	なし
リターン値	なし

iic_tei_int_eeread	
概要	EEPROM 読み込み処理後の送信終了処理
ヘッダ	なし
宣言	static void iic_tei_int_eeread(void)
説明	リスタートコンディションの発行を要求します。
引数	なし
リターン値	なし

iic_gen_clk_sp	
概要	異常発生時のストップコンディション生成処理
ヘッダ	なし
宣言	static void iic_gen_clk_sp(void)
説明	異常発生時にストップコンディションの発行を要求します。
引数	なし
リターン値	なし

iic_error	
概要	エラー処理
ヘッダ	なし
宣言	static void iic_error (enum riic_err_code_t error_code)
説明	通常この関数は呼び出されません。呼び出された場合は無限ループします。
引数	enum riic_err_code_t error_code :IIC のエラーコード
リターン値	なし

Excep_RIICn_EEIn	
概要	RIICn.EEIn 割り込み処理(レベル割り込み)
ヘッダ	なし
宣言	static void Excep_RIICn_EEIn(void)
説明	通信エラー/イベント発生割り込み関数を呼び出します。
引数	なし
リターン値	なし

Excep_RIICn_RXIn	
概要	RIICn.RXIn 割り込み処理(エッジ割り込み)
ヘッダ	なし
宣言	static void Excep_RIICn_RXIn(void)
説明	受信データフル割り込み関数を呼び出します。
引数	なし
リターン値	なし

Excep_RIICn_TXIn	
概要	RIICn.TXIn 割り込み処理(エッジ割り込み)
ヘッダ	なし
宣言	static void Excep_RIICn_TXIn(void)
説明	送信データエンティ割り込み関数を呼び出します。
引数	なし
リターン値	なし

Excep_RIICn_TEIn	
概要	RIICn.TEIn 割り込み処理(レベル割り込み)
ヘッダ	なし
宣言	static void Excep_RIICn_TEIn(void)
説明	送信終了割り込み関数を呼び出します。
引数	なし
リターン値	なし

5.12 フローチャート

本アプリケーションノートのフローチャートは、RIIC0 チャンネルを選択した場合のフローチャートです。RX220 の RIIC チャンネル数は 1 チャンネル(RIIC0)なので、チャンネル選択はできません。RX21A は RIIC0、RIIC1 の 2 チャンネルから選択できます。

5.12.1 メイン処理

図 5.6にメイン処理のフローチャートを示します。

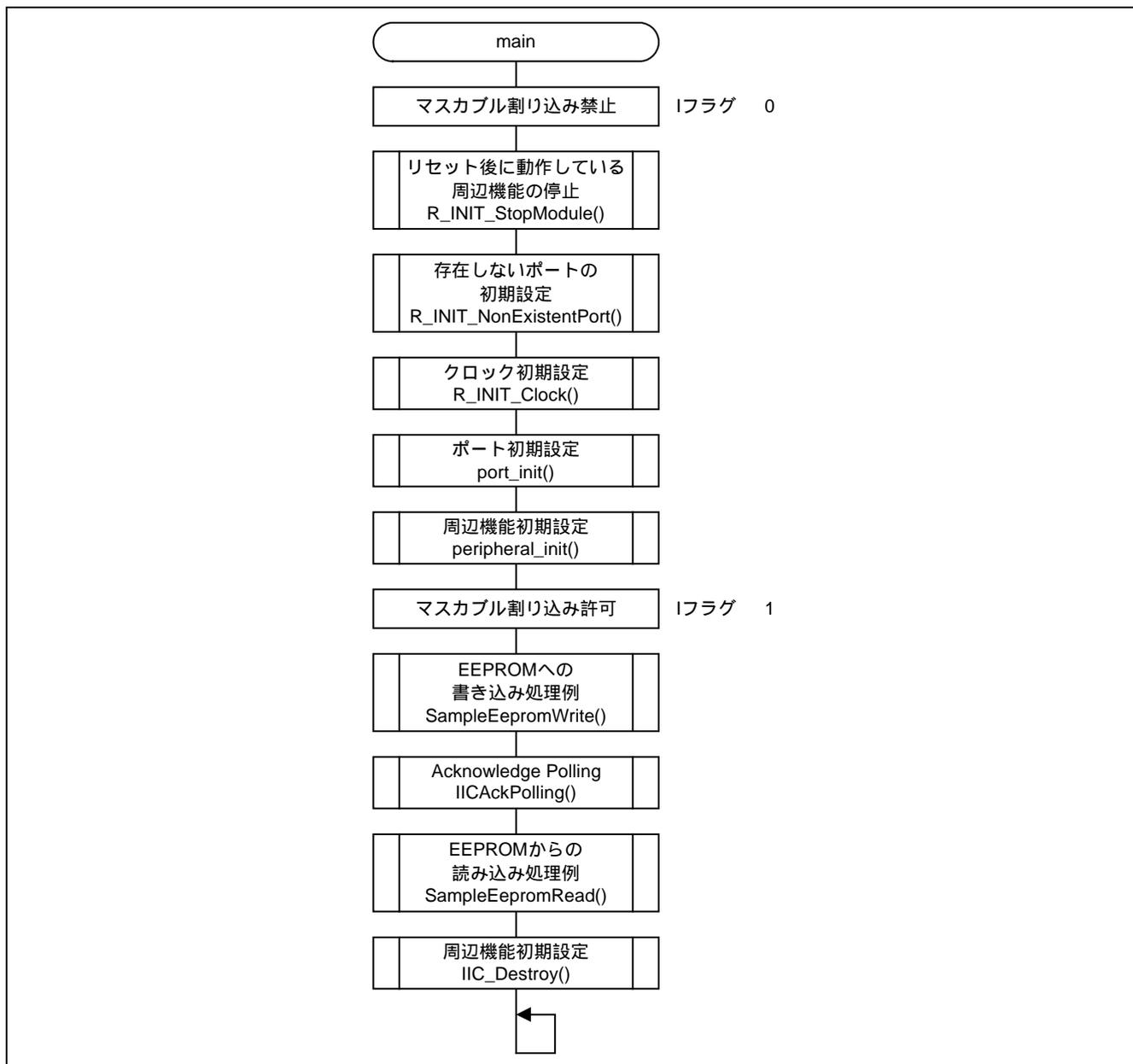


図 5.6 メイン処理

5.12.2 ポート初期設定

図 5.7にポート初期設定のフローチャートを示します。

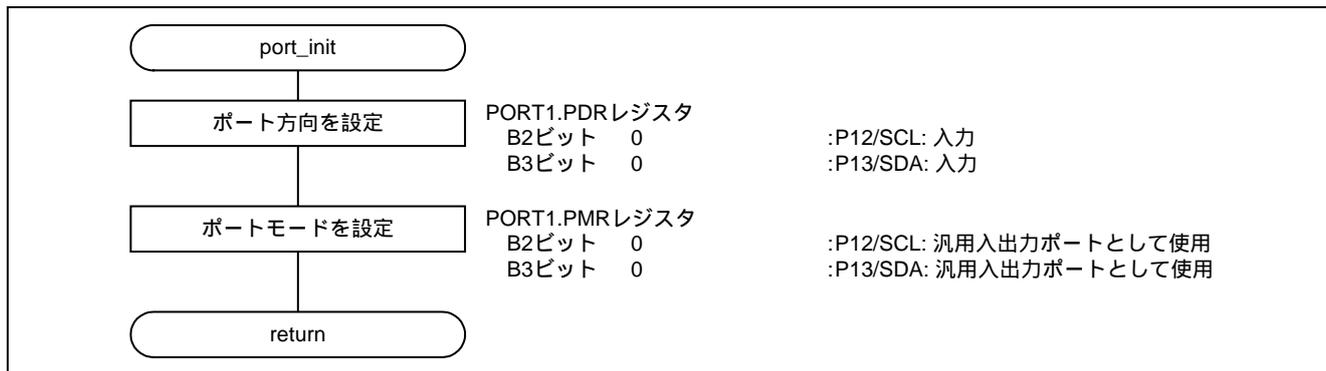


図 5.7 ポート初期設定

5.12.3 周辺機能初期設定

図 5.8に周辺機能初期化設定のフローチャートを示します。

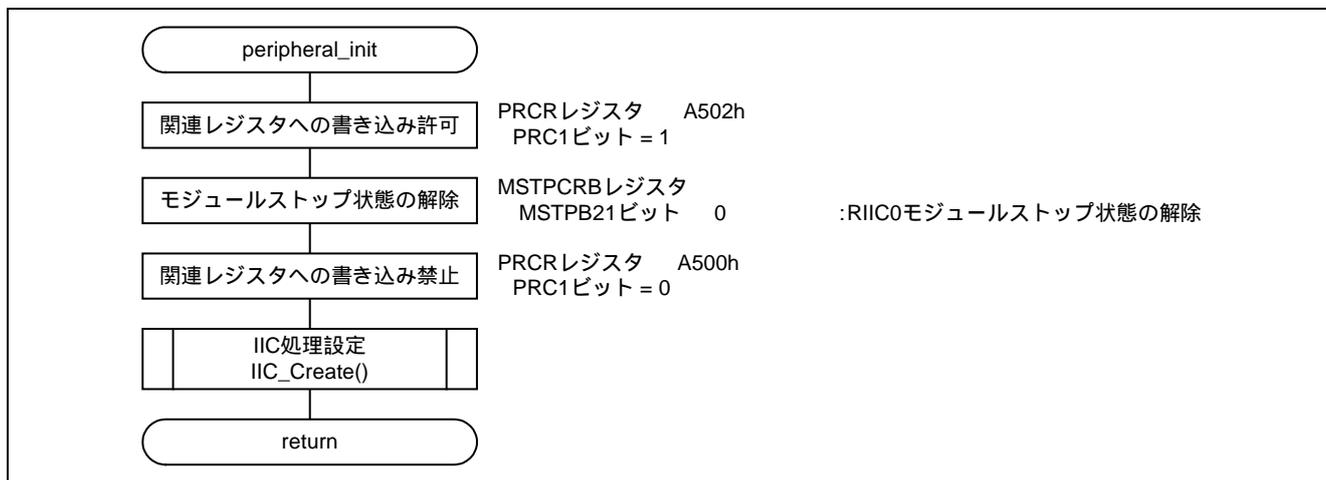


図 5.8 周辺機能初期化設定

5.12.4 CPU 割り込み設定

図 5.9、図 5.10にCPU 割り込み設定のフローチャートを示します。

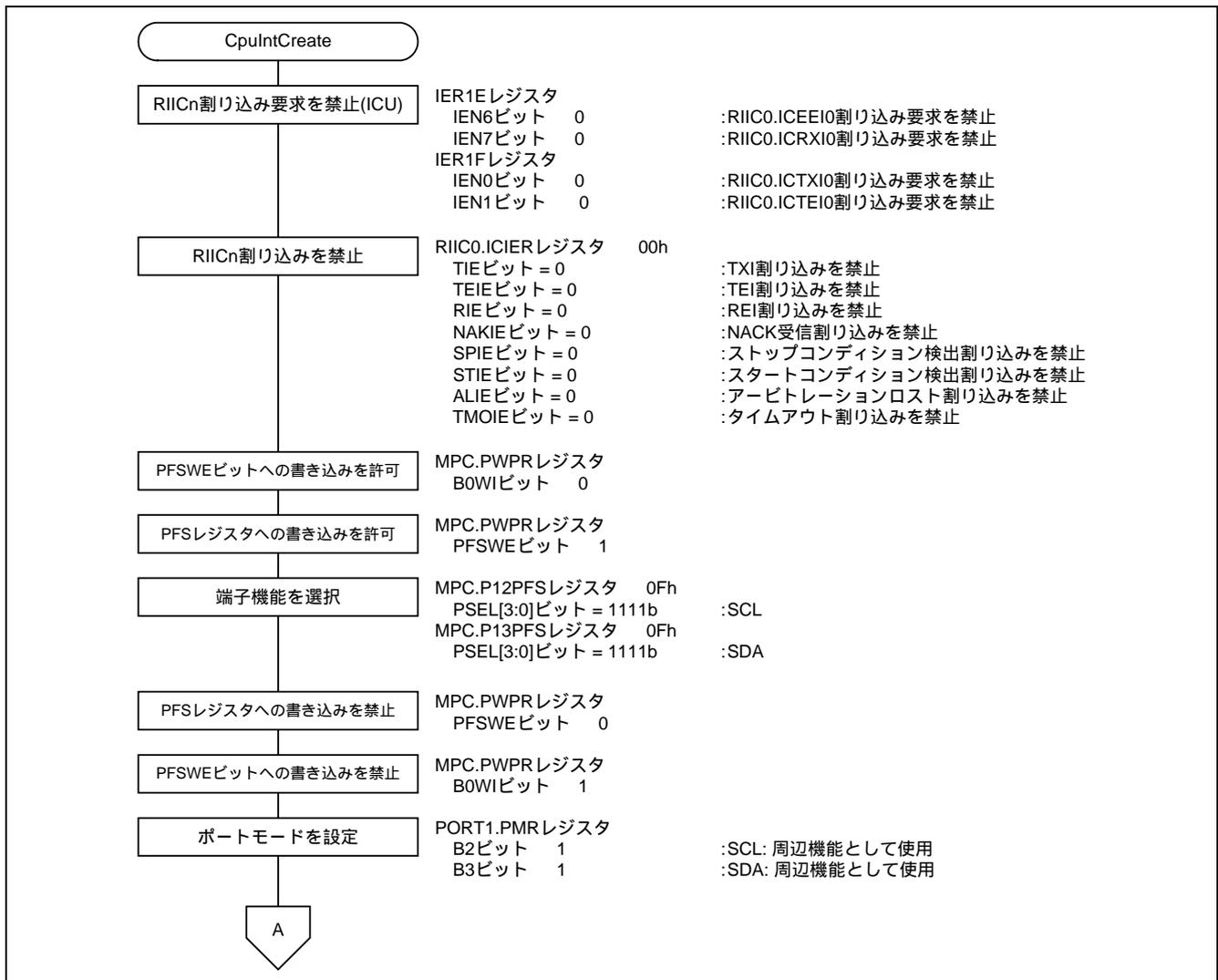


図 5.9 CPU 割り込み設定(1/2)

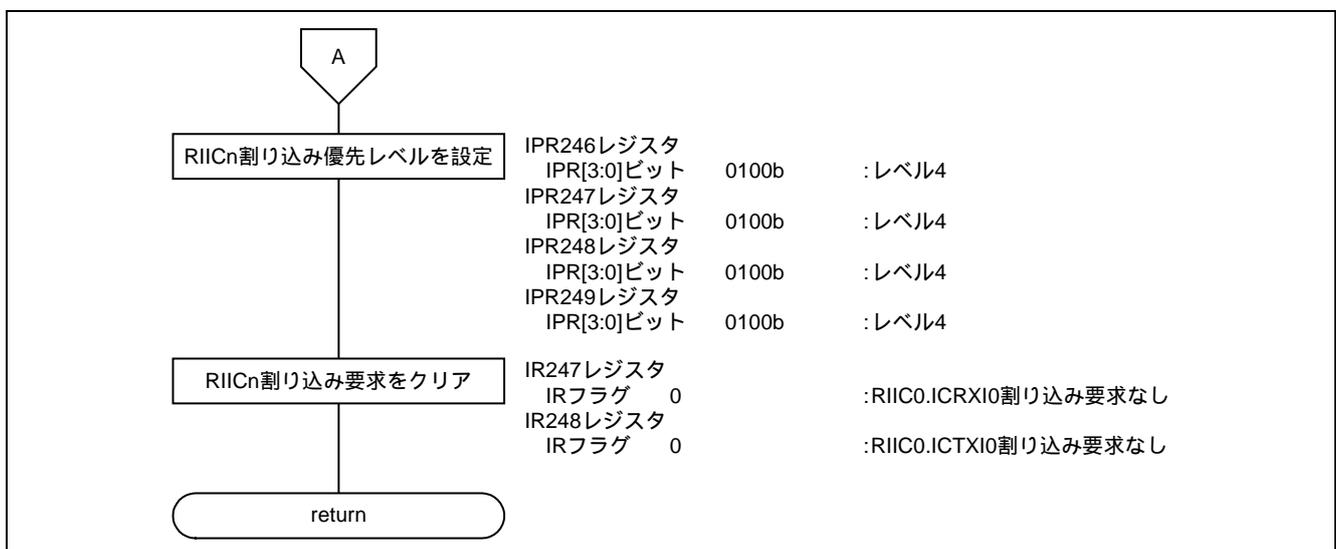


図 5.10 CPU 割り込み設定(2/2)

5.12.5 EEPROM への書き込み処理例

図 5.11にEEPROM への書き込み処理例のフローチャートを示します。

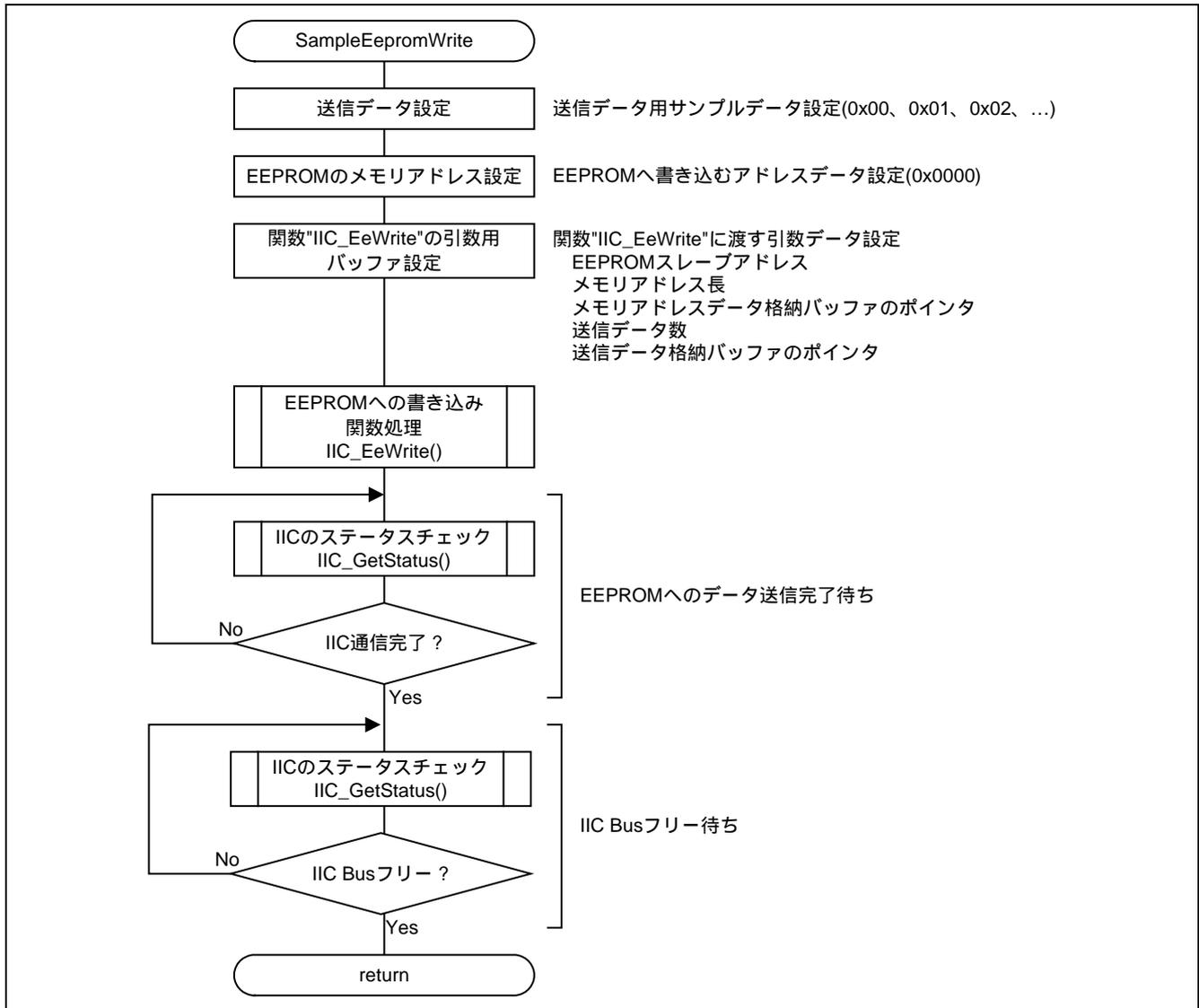


図 5.11 EEPROM への書き込み処理例

5.12.6 EEPROM からの読み込み処理例

図 5.12にEEPROM からの読み込み処理例のフローチャートを示します。

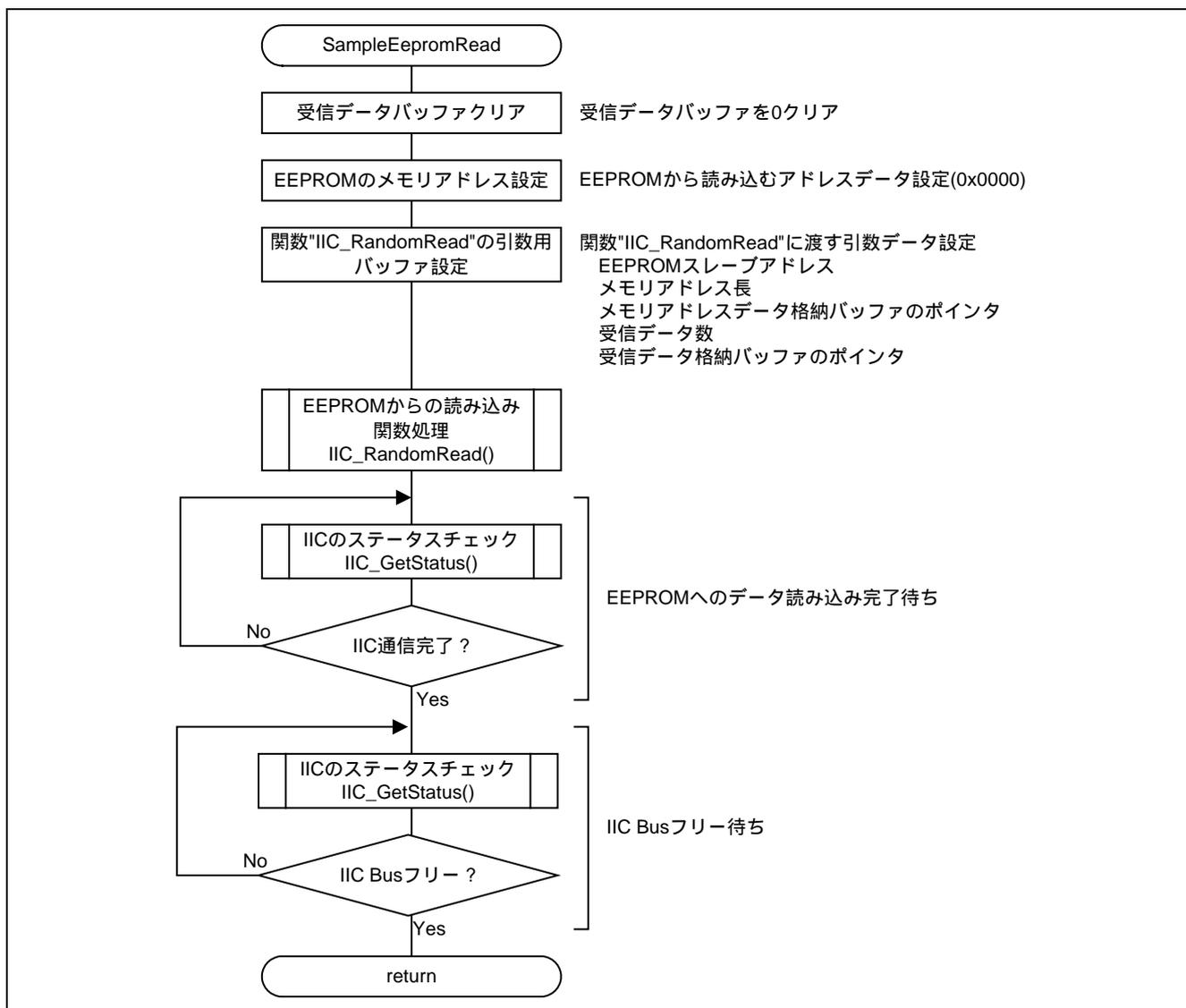


図 5.12 EEPROM からの読み込み処理例

5.12.7 Acknowledge Polling

図 5.13にAcknowledge Pollingのフローチャートを示します。

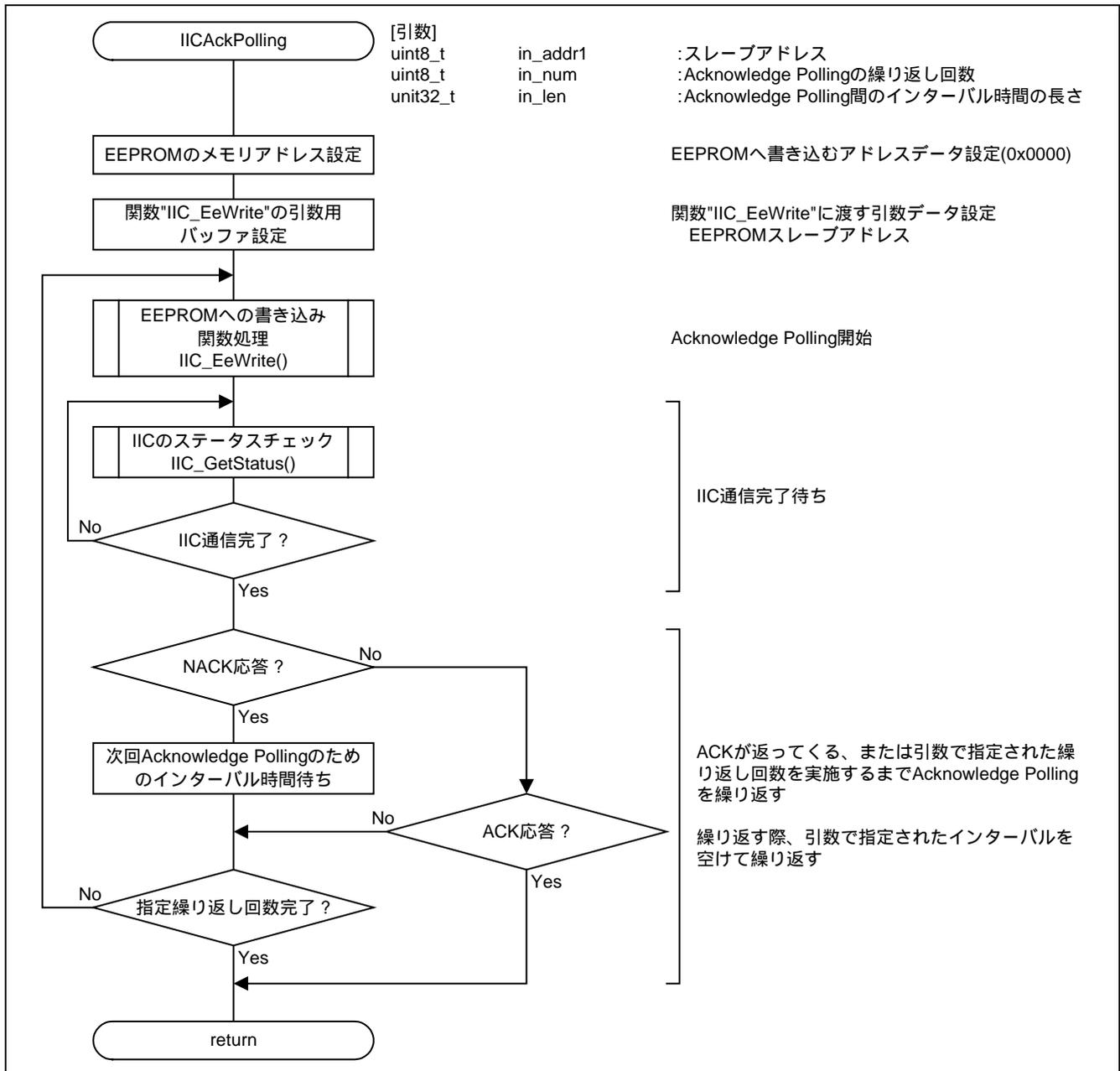


図 5.13 Acknowledge Polling

5.12.8 IIC 処理設定

図 5.14にIIC 処理設定のフローチャートを示します。

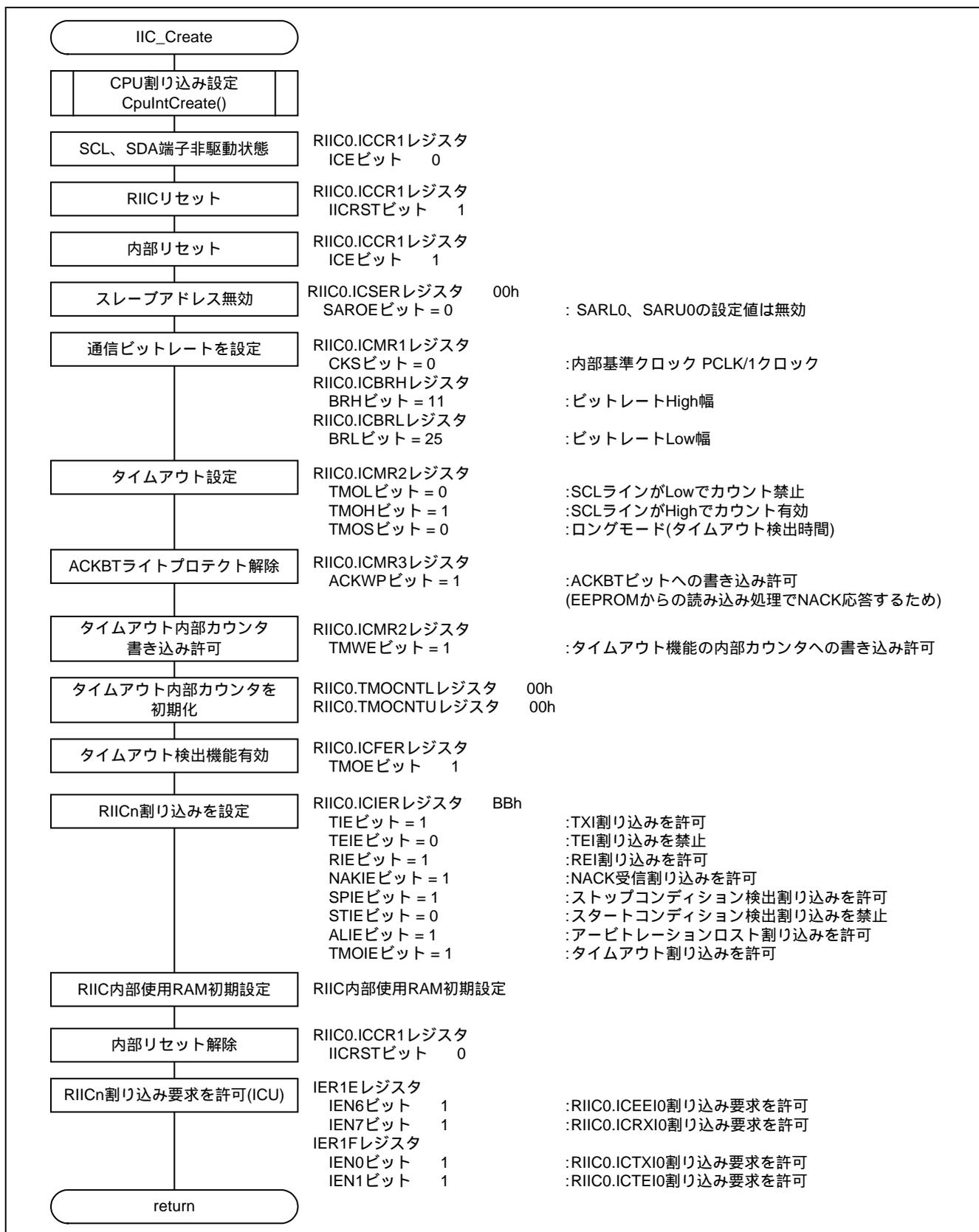


図 5.14 IIC 処理設定

5.12.9 IIC 終了処理

図 5.15にIIC 終了処理のフローチャートを示します。

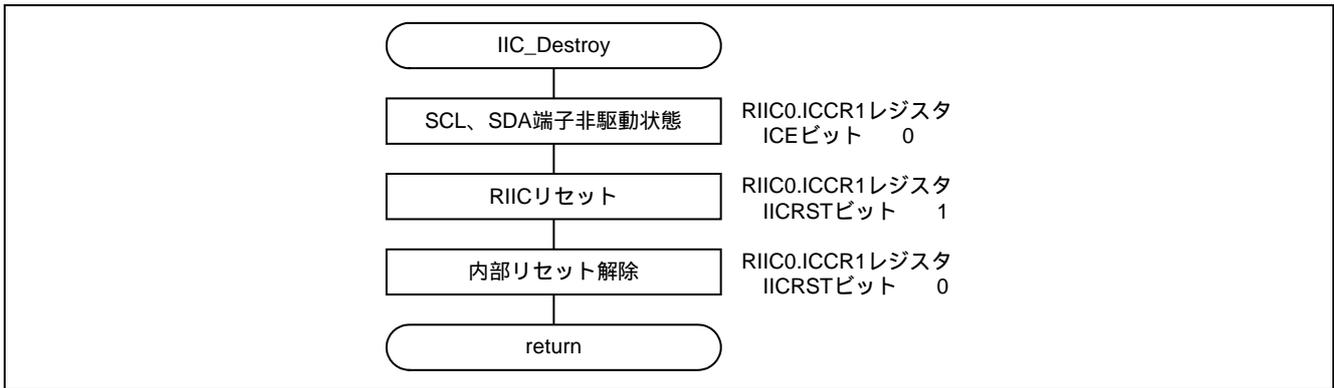


図 5.15 IIC 終了処理

5.12.10 EEPROM への書き込み開始処理

図 5.16にEEPROM への書き込み開始処理のフローチャートを示します。

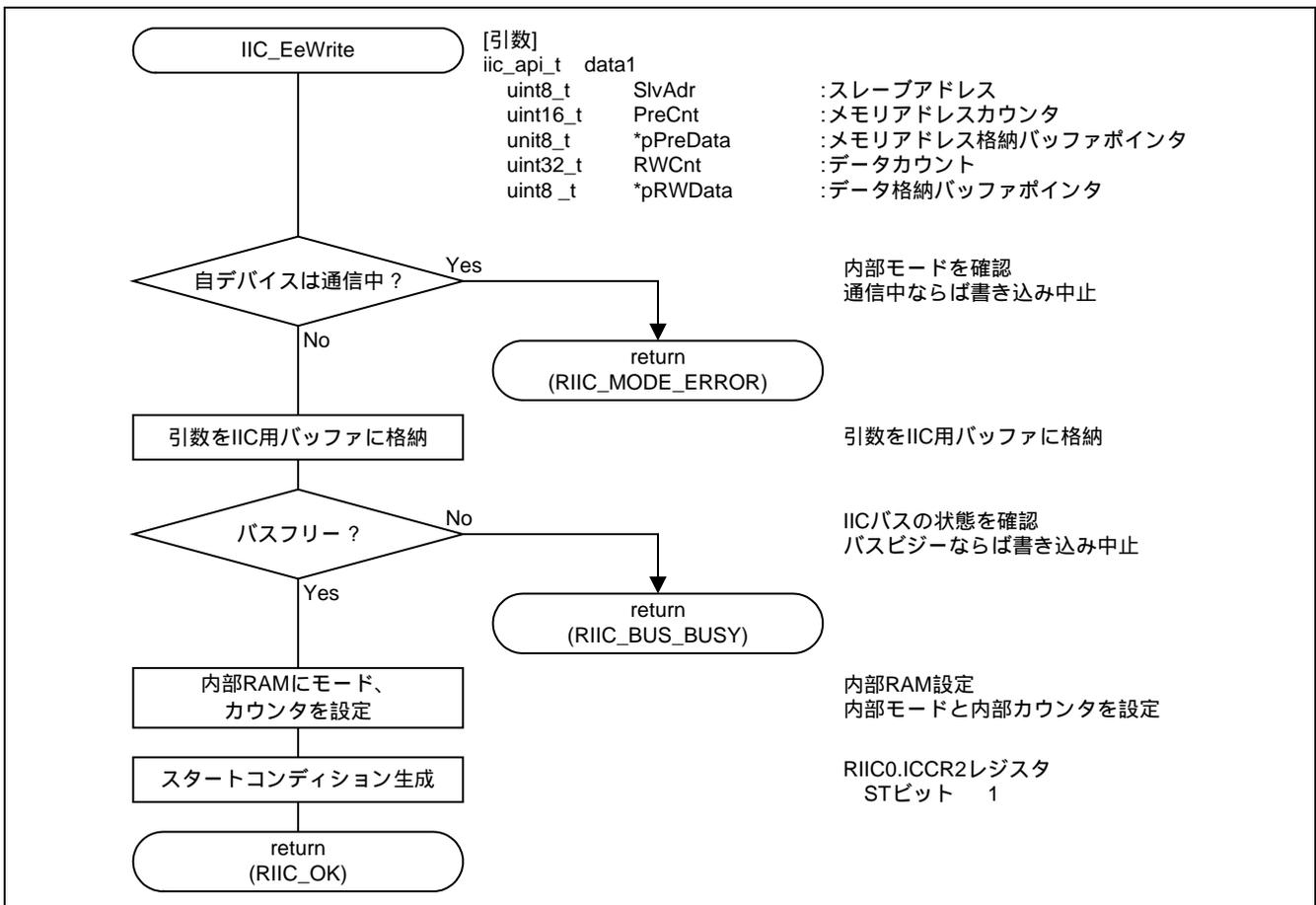


図 5.16 EEPROM への書き込み開始処理

5.12.11 EEPROM からの読み込み開始処理

図 5.17にEEPROM からの読み込み開始処理のフローチャートを示します。

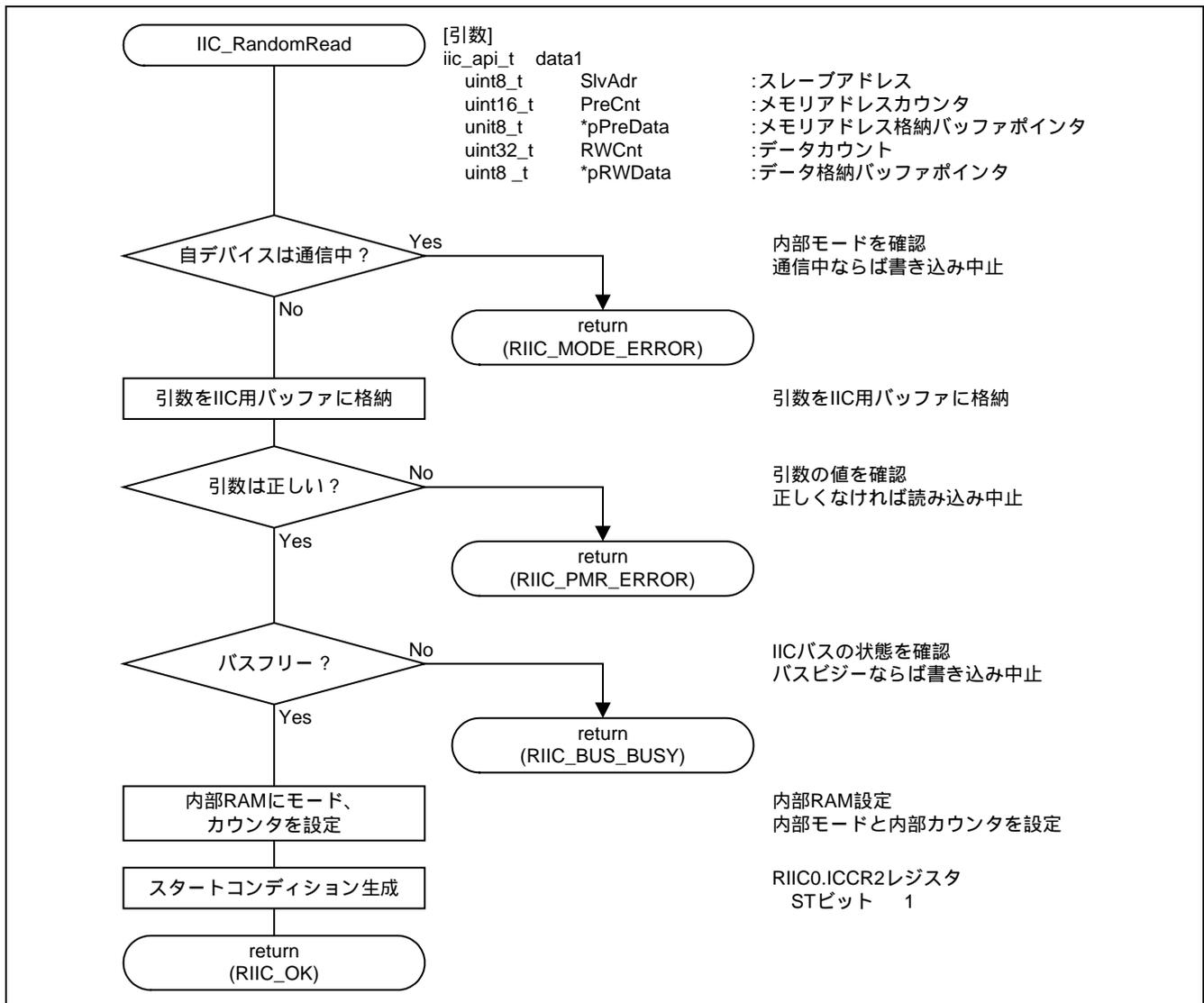


図 5.17 EEPROM からの読み込み開始処理

5.12.12 IIC のステータスチェック

図 5.18にIIC のステータスチェックのフローチャートを示します。

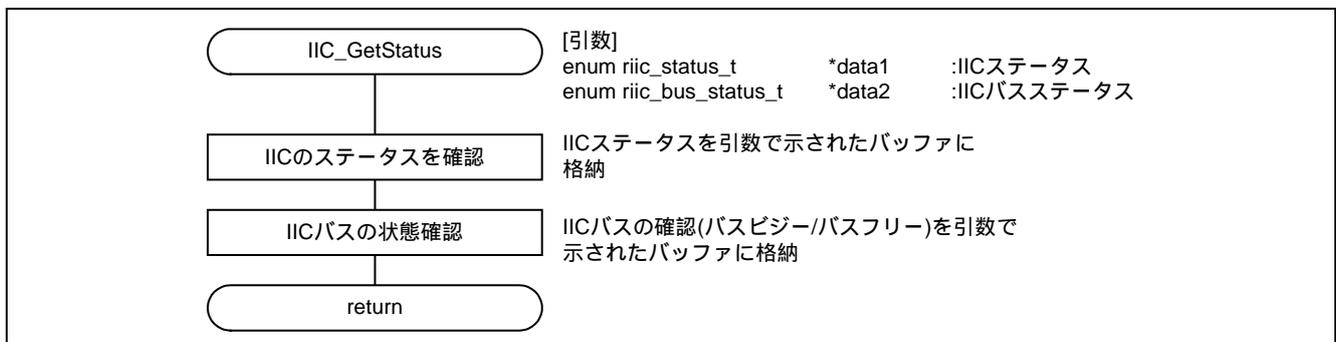


図 5.18 IIC のステータスチェック

5.12.13 通信エラー/イベント発生割り込み

図 5.19に通信エラー/イベント発生割り込みのフローチャートを示します。

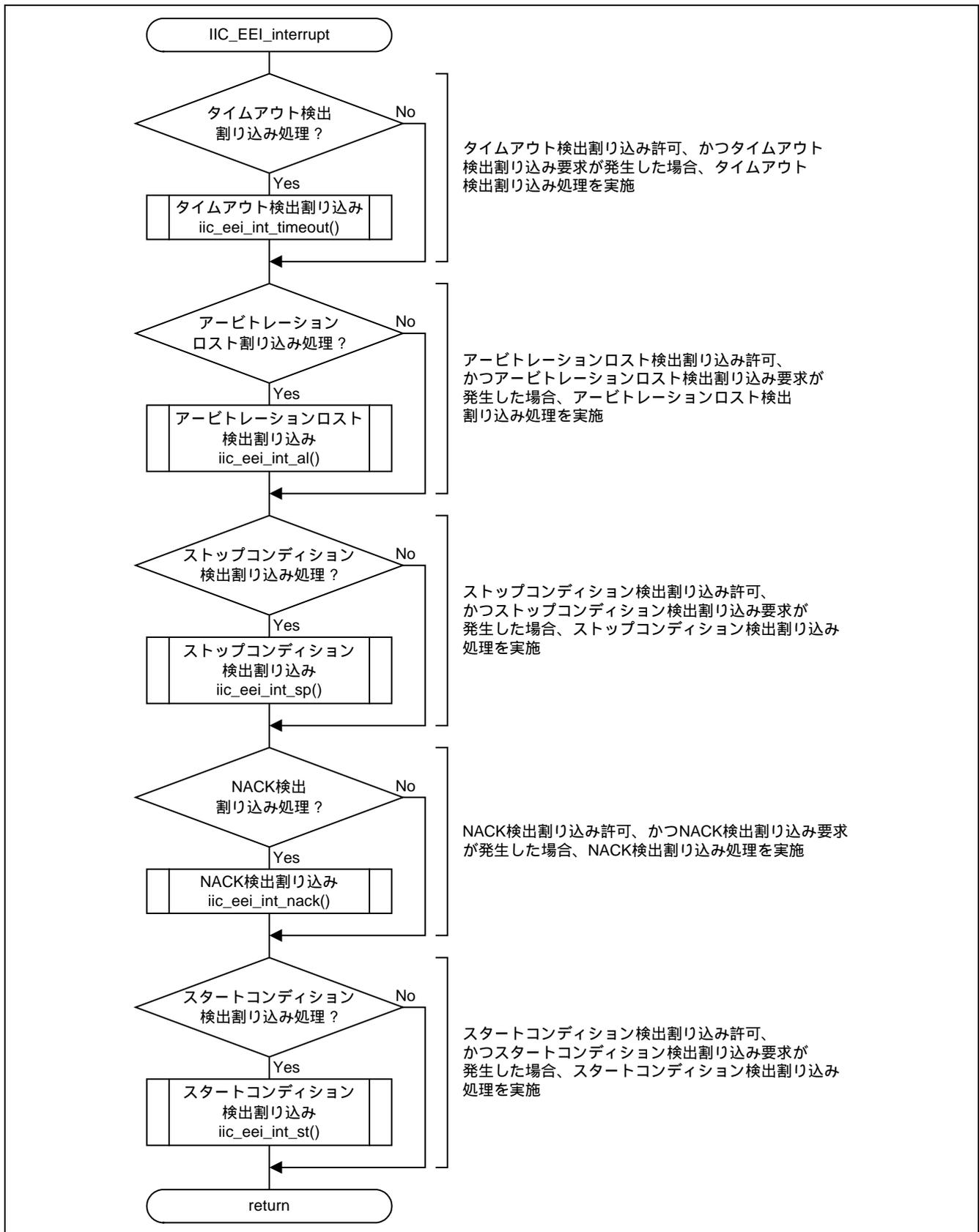


図 5.19 通信エラー/イベント発生割り込み

5.12.14 タイムアウト検出割り込み

図 5.20にタイムアウト検出割り込みのフローチャートを示します。

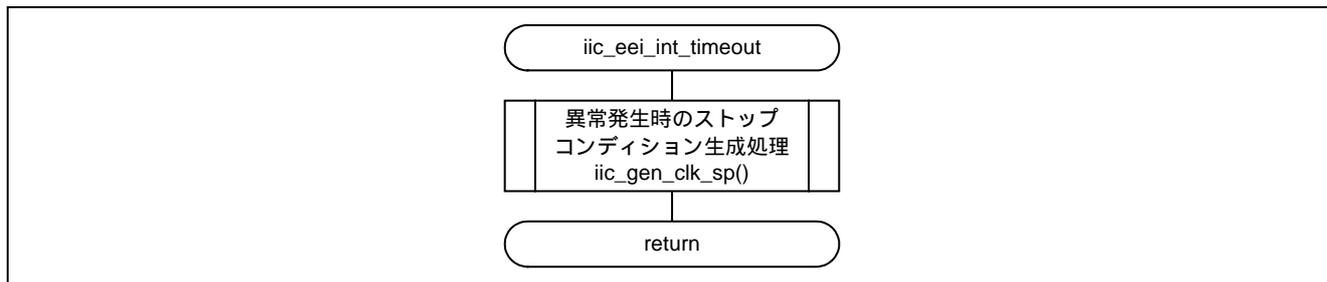


図 5.20 タイムアウト検出割り込み

5.12.15 アービトレーションロスト検出割り込み

図 5.21にアービトレーションロスト検出割り込みのフローチャートを示します。

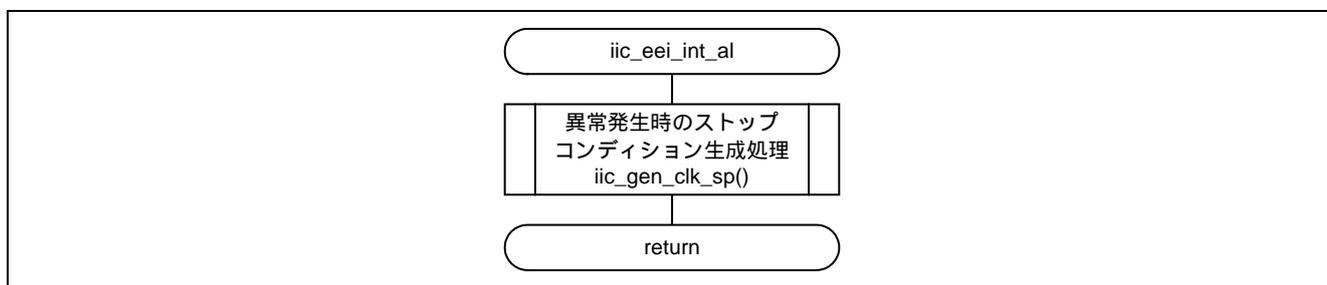


図 5.21 アービトレーションロスト検出割り込み

5.12.16 ストップコンディション検出割り込み

図 5.22、図 5.23 にストップコンディション検出割り込みのフローチャートを示します。

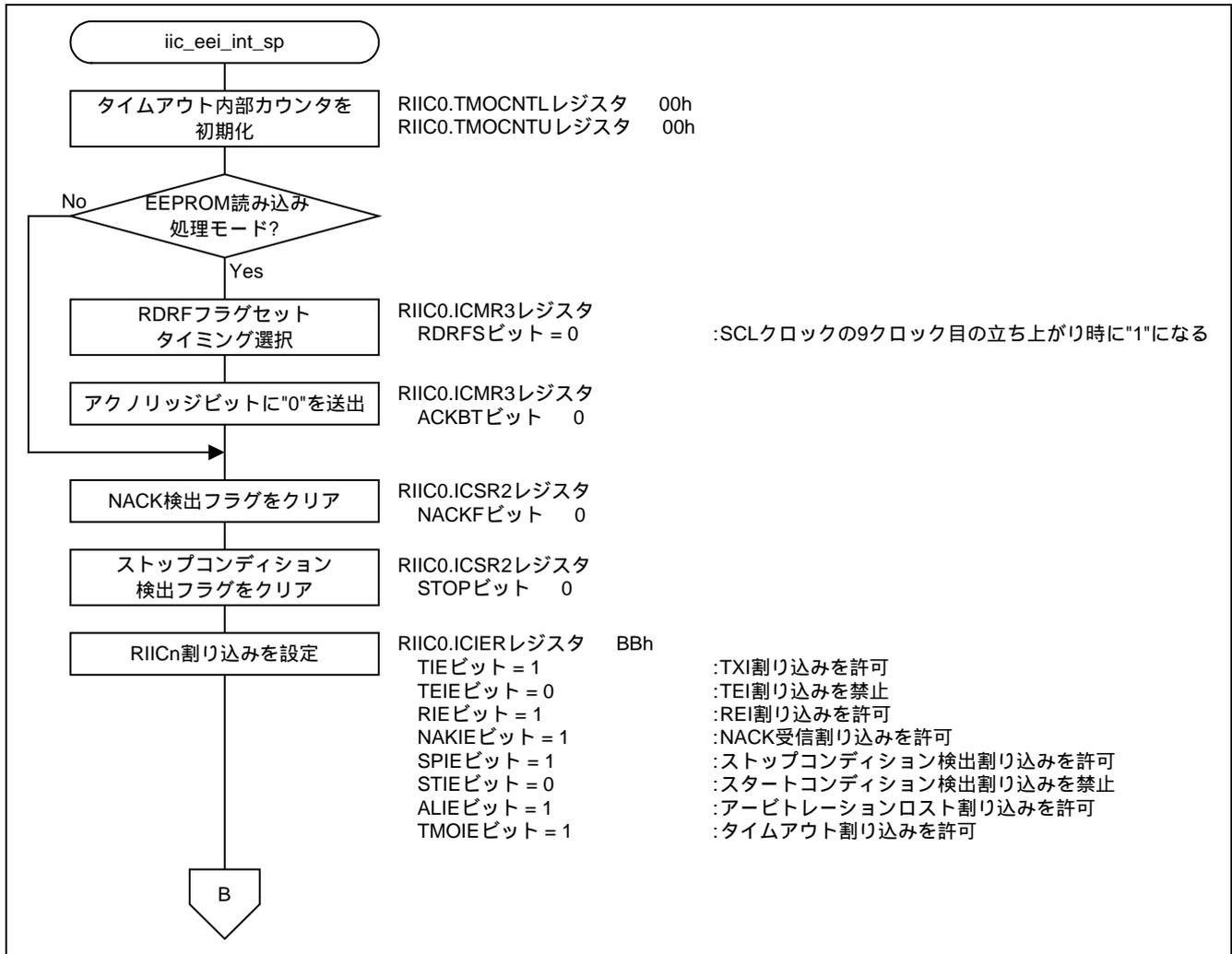


図 5.22 ストップコンディション検出割り込み(1/2)

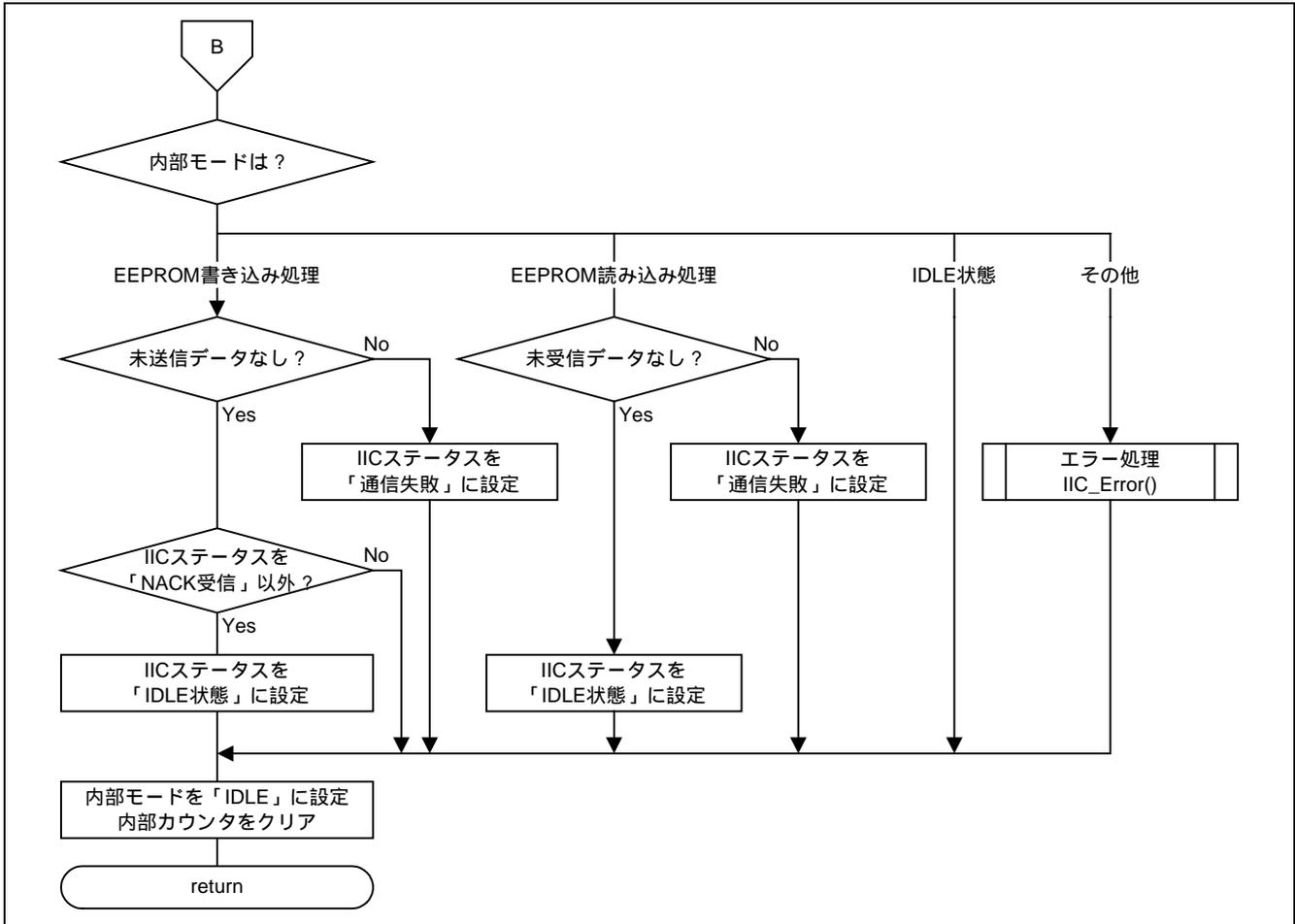


図 5.23 ストップコンディション検出割り込み(2/2)

5.12.17 スタートコンディション検出割り込み

図 5.24にスタートコンディション検出割り込みのフローチャートを示します。

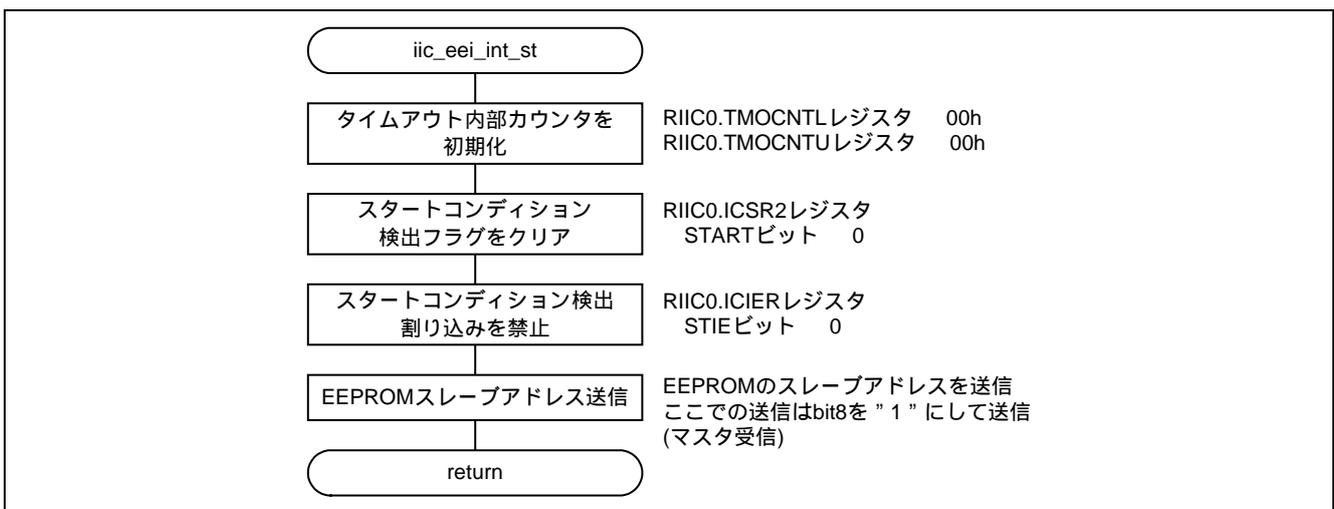


図 5.24 スタートコンディション検出割り込み

5.12.18 NACK 検出割り込み

図 5.25にNACK 検出割り込みのフローチャートを示します。

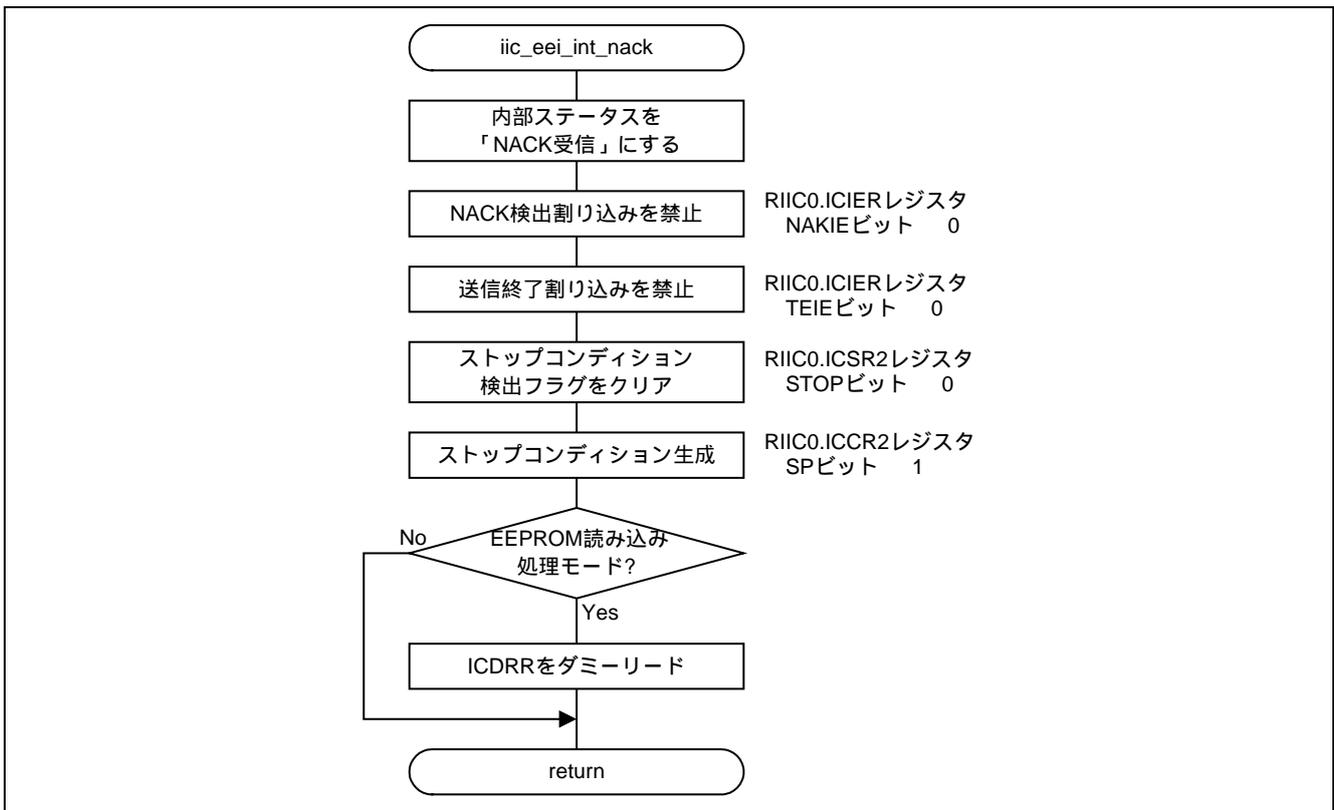


図 5.25 NACK 検出割り込み

5.12.19 受信データフル割り込み

図 5.26に受信データフル割り込みのフローチャートを示します。

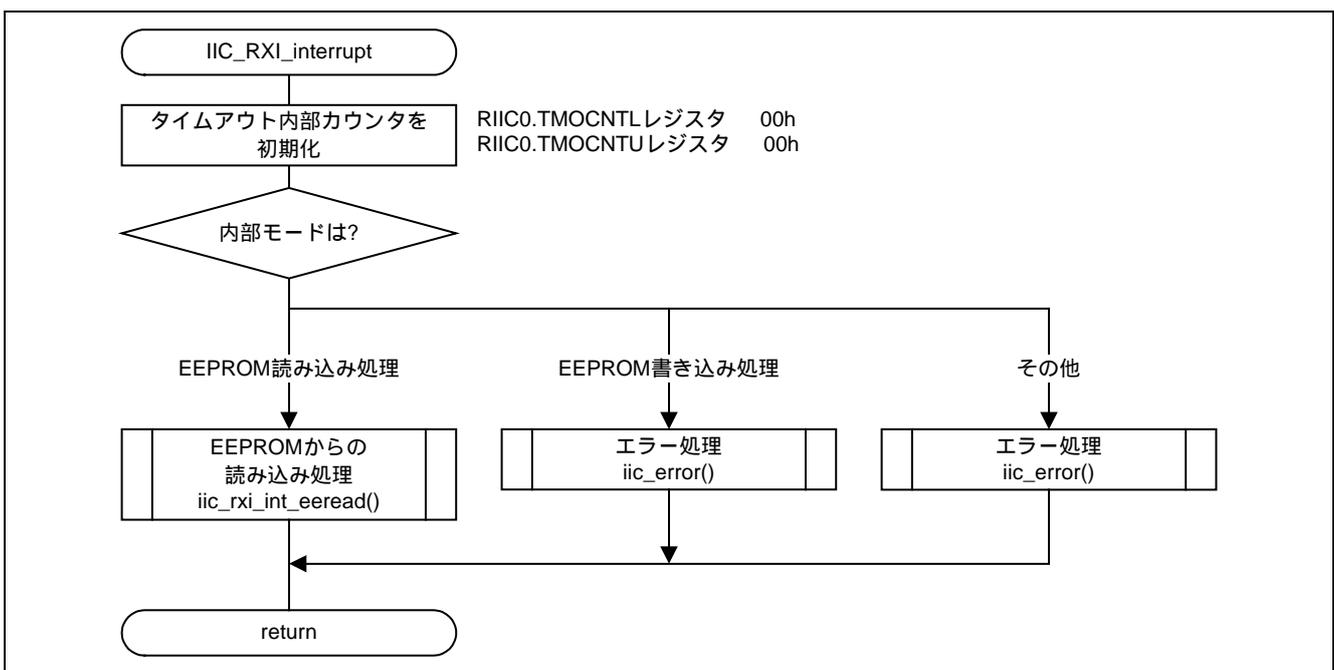


図 5.26 受信データフル割り込み

5.12.20 EEPROM からの読み込み処理 (マスタ受信部分)

図 5.27にEEPROM からの読み込み処理 (マスタ受信部分) のフローチャートを示します。

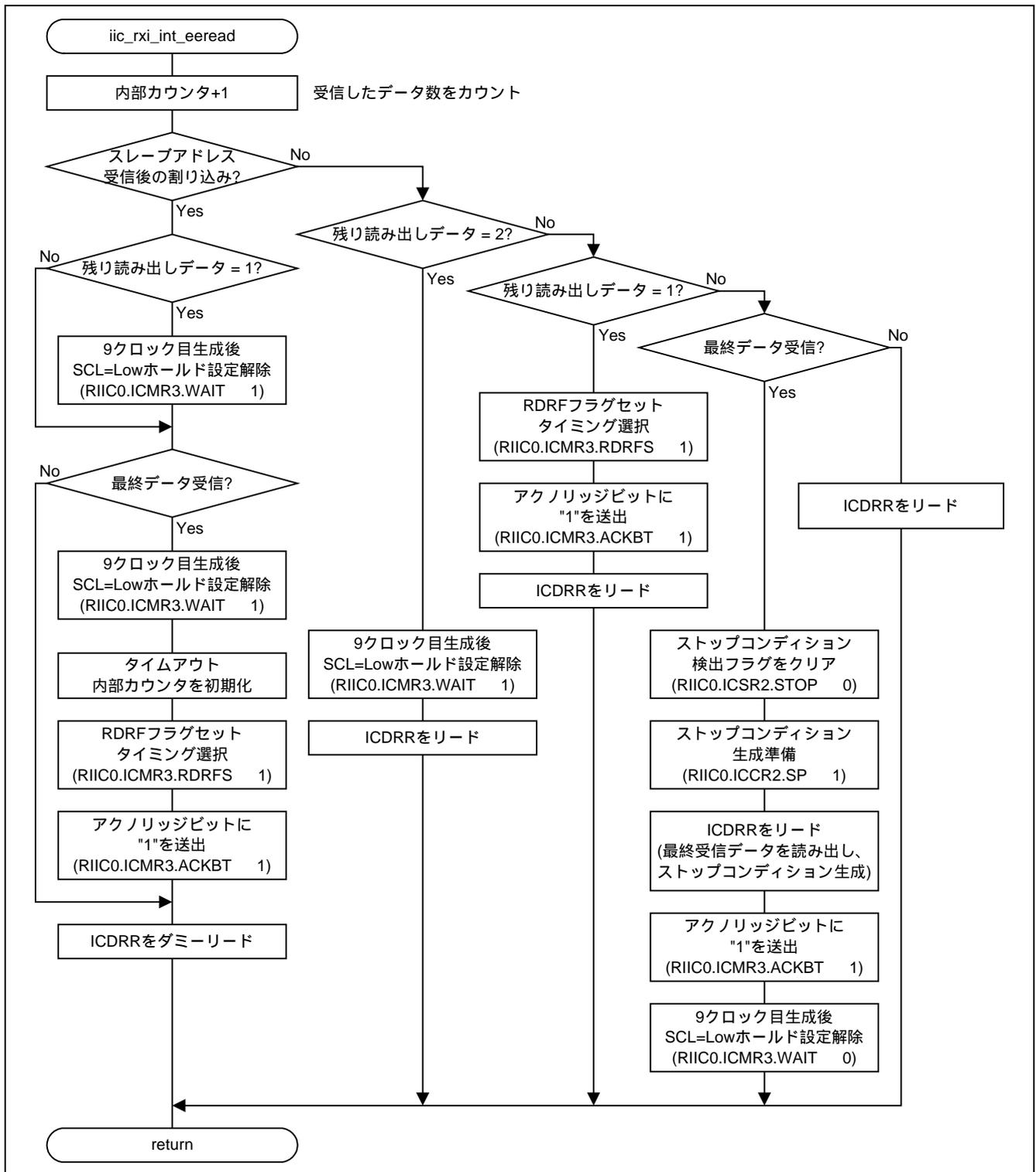


図 5.27 EEPROM からの読み込み処理 (マスタ受信部分)

5.12.21 送信データエンプティ割り込み

図 5.28に送信データエンプティ割り込みのフローチャートを示します。

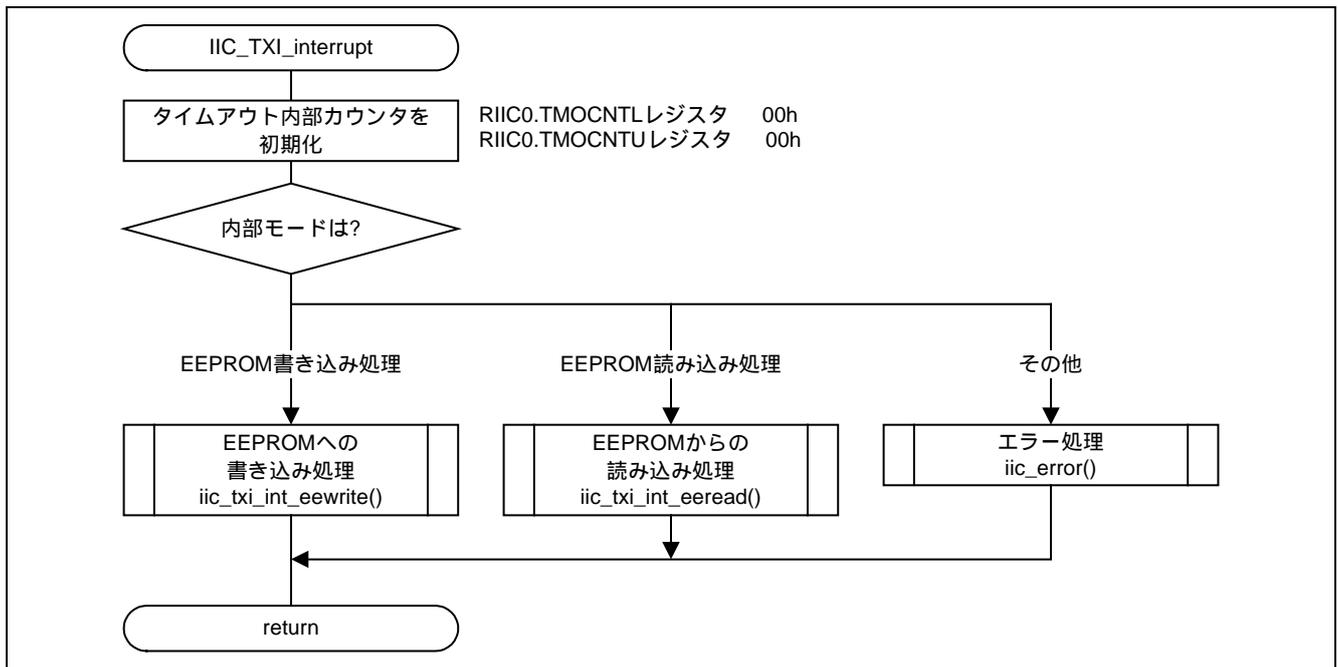


図 5.28 送信データエンプティ割り込み

5.12.22 EEPROM への書き込み処理

図 5.29にEEPROM への書き込み処理のフローチャートを示します。

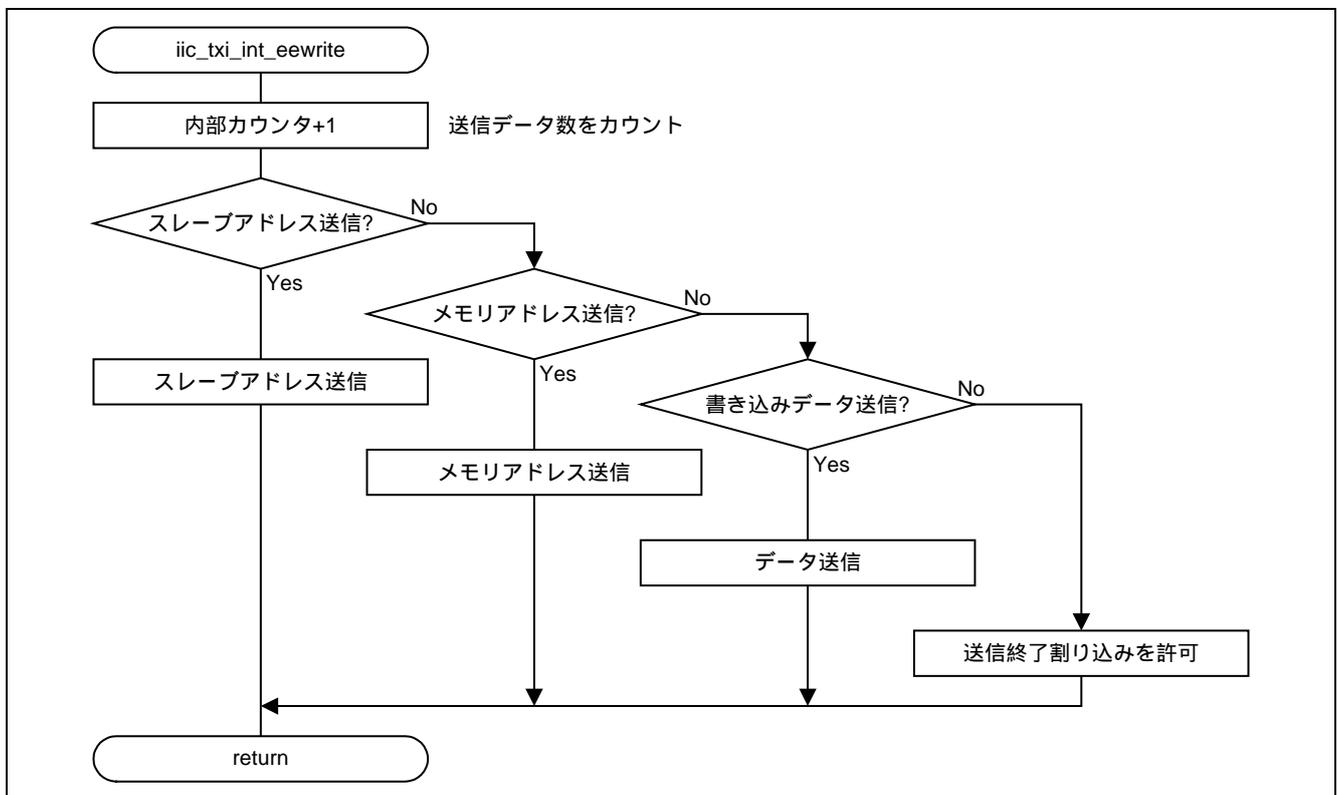


図 5.29 EEPROM への書き込み処理

5.12.23 EEPROM からの読み込み処理（マスタ送信部分）

図 5.30にEEPROM からの読み込み処理（マスタ送信部分）のフローチャートを示します。

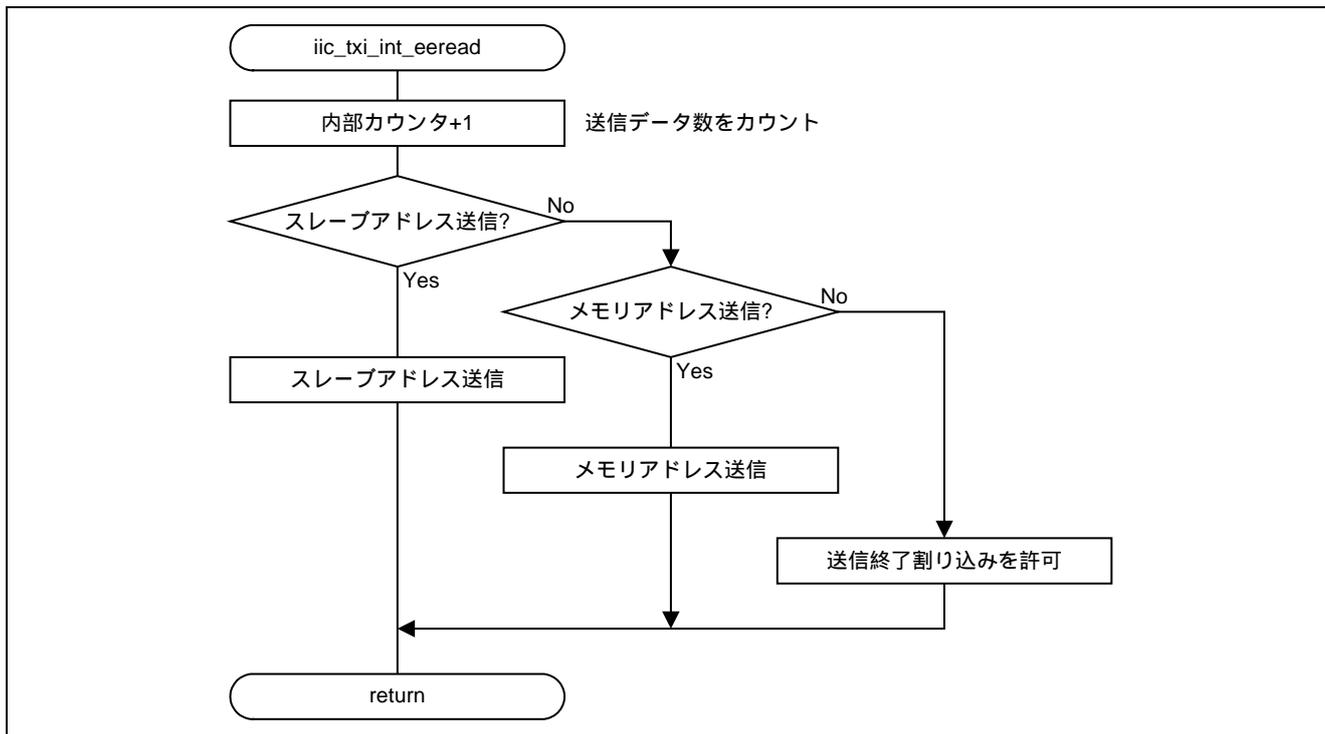


図 5.30 EEPROM からの読み込み処理（マスタ送信部分）

5.12.24 送信終了割り込み

図 5.31に送信終了割り込みのフローチャートを示します。

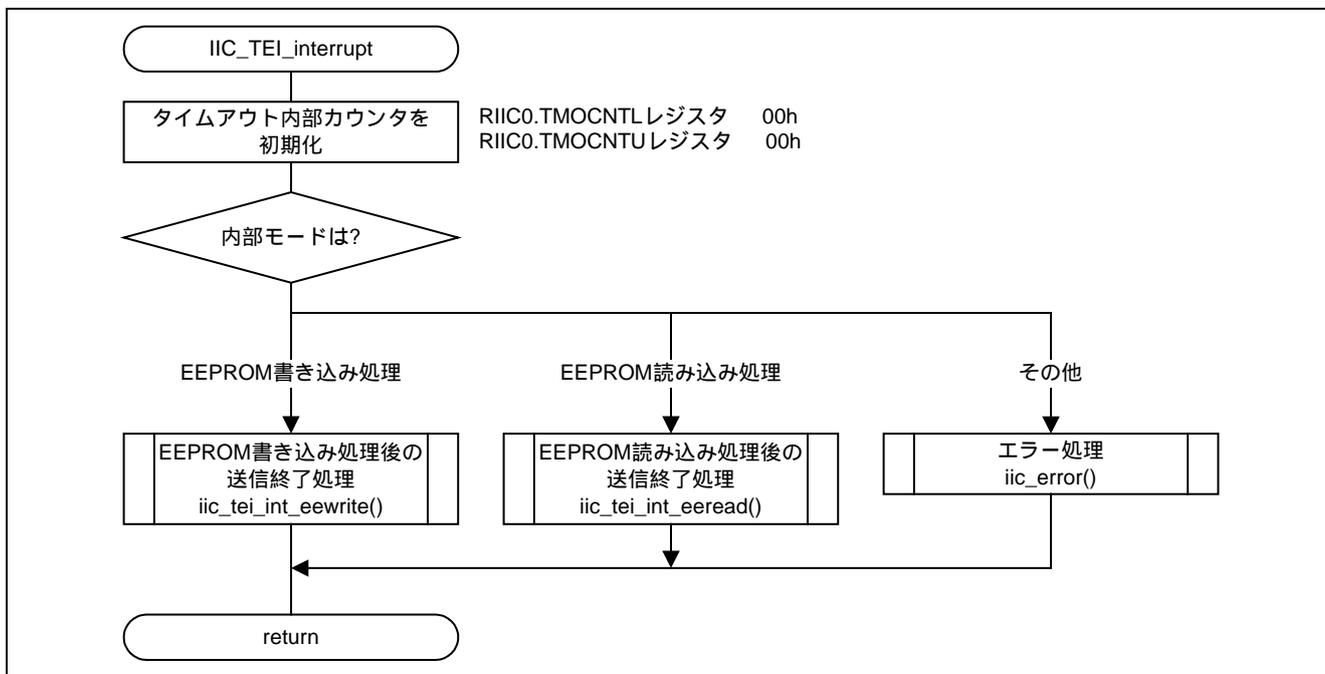


図 5.31 送信終了割り込み

5.12.25 EEPROM 書き込み処理後の送信終了処理

図 5.32にEEPROM 書き込み処理後の送信終了処理のフローチャートを示します。

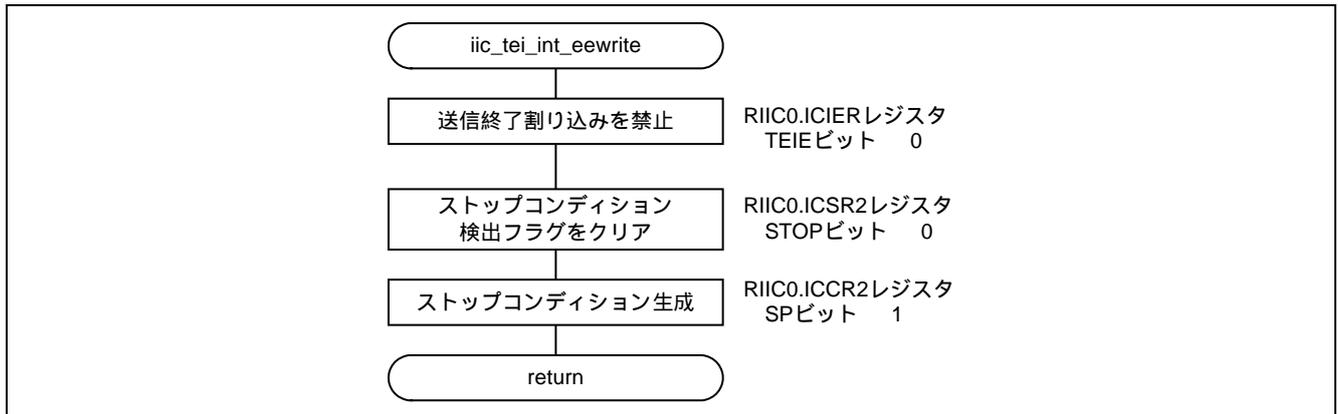


図 5.32 EEPROM 書き込み処理後の送信終了処理

5.12.26 EEPROM 読み込み処理後の送信終了処理

図 5.33にEEPROM 読み込み処理後の送信終了処理のフローチャートを示します。

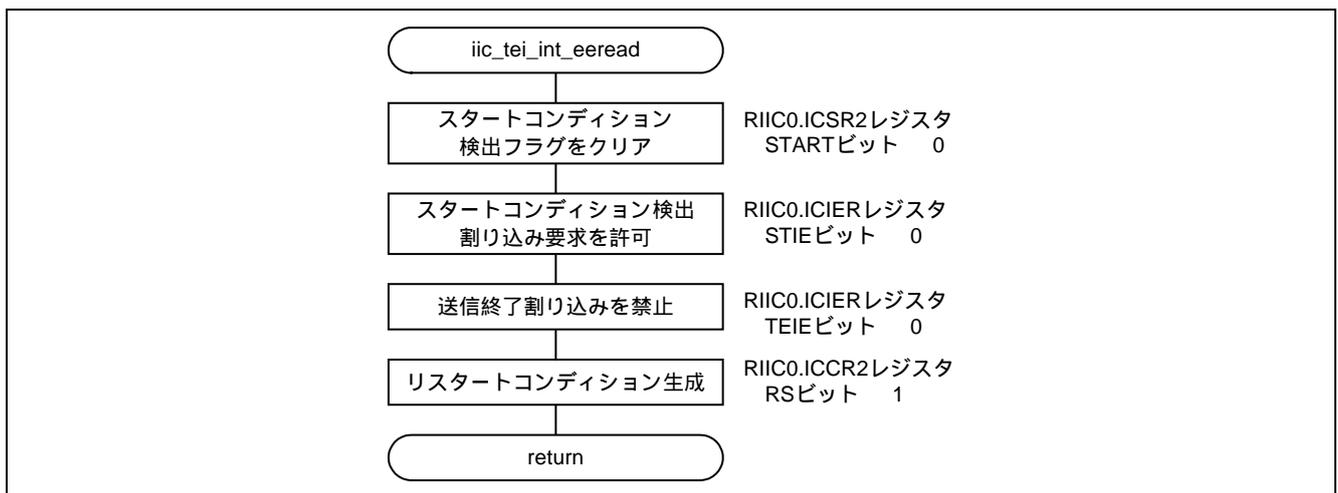


図 5.33 EEPROM 読み込み処理後の送信終了処理

5.12.27 異常発生時のストップコンディション生成処理

図 5.34、図 5.35に異常発生時のストップコンディション生成処理のフローチャートを示します。

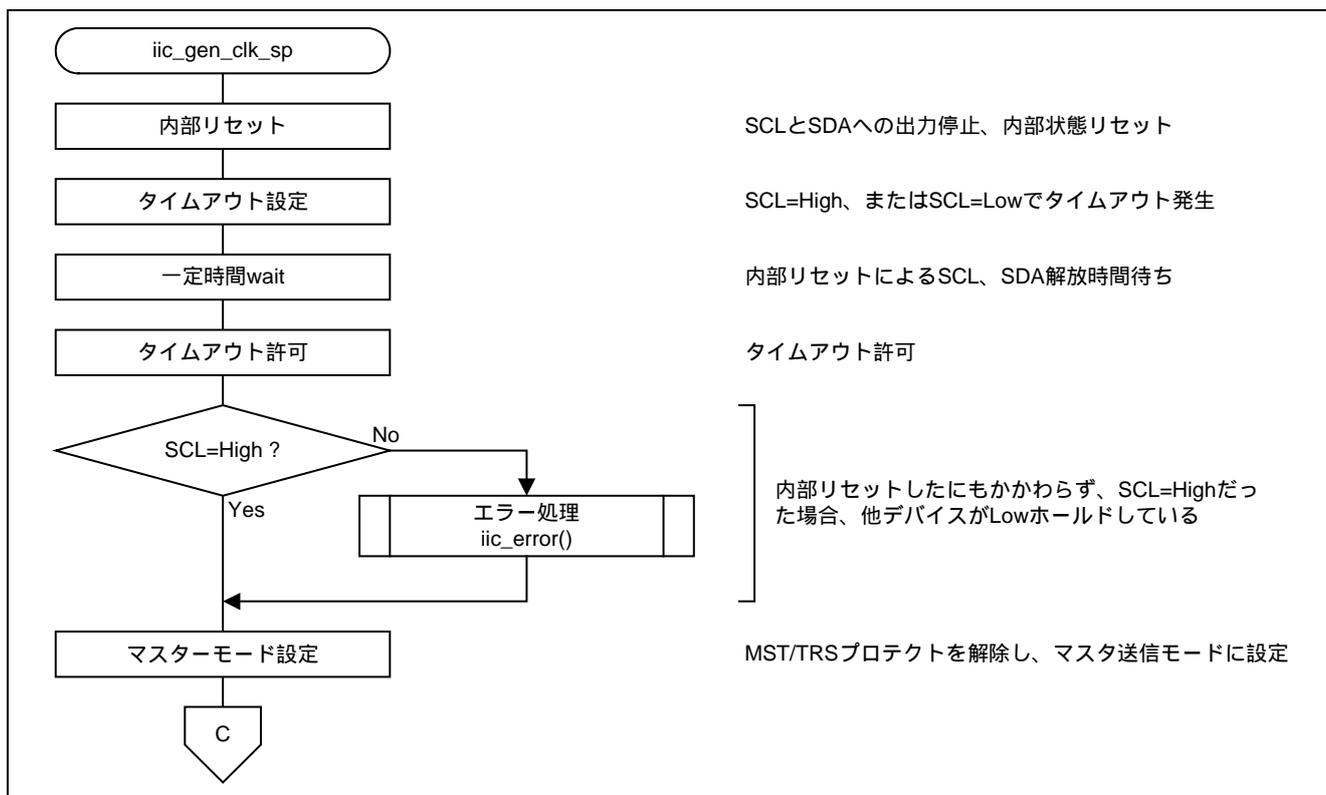


図 5.34 異常発生時のストップコンディション生成処理(1/2)

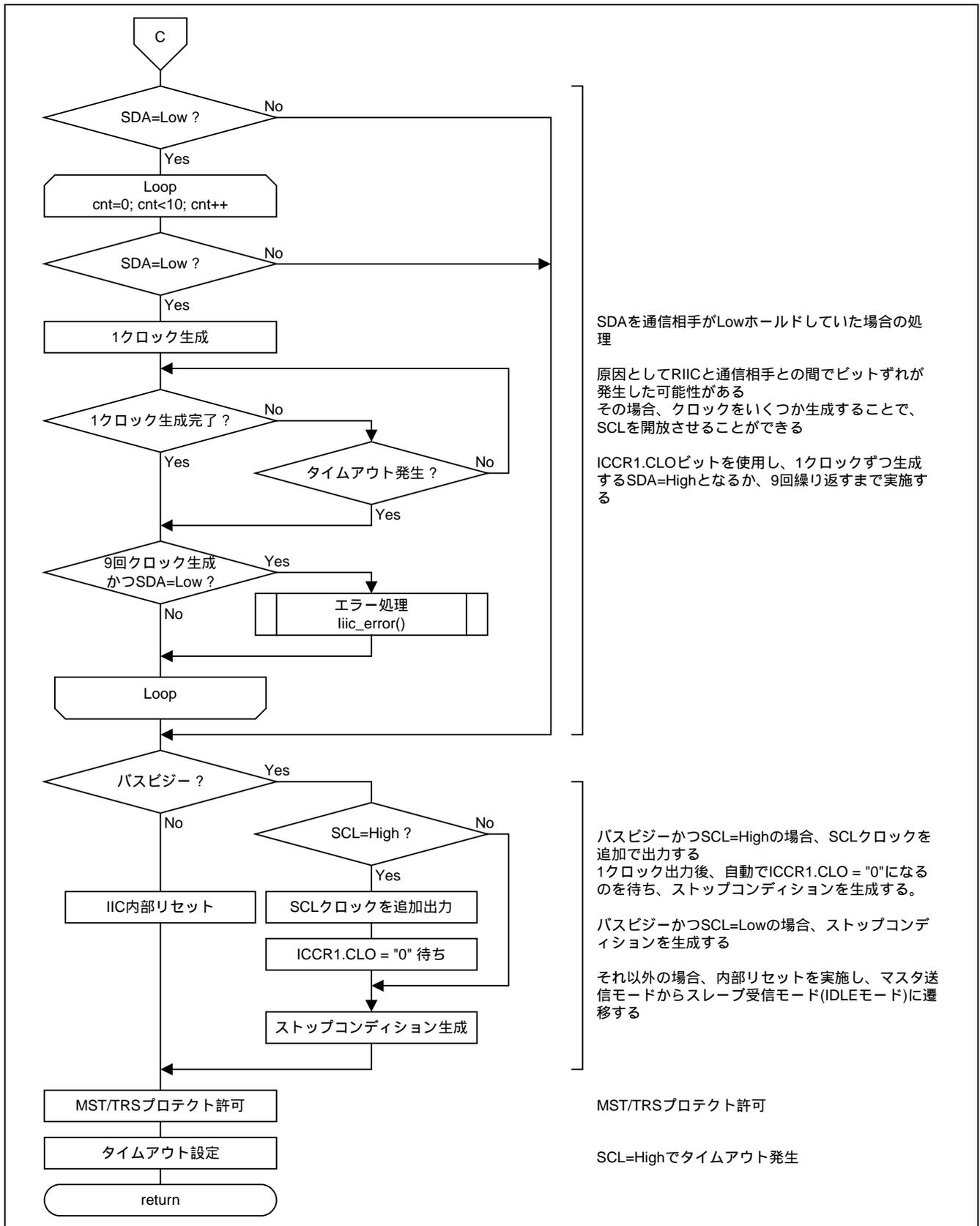


図 5.35 異常発生時のストップコンディション生成処理(2/2)

5.12.28 エラー処理

図 5.36にエラー処理のフローチャートを示します。

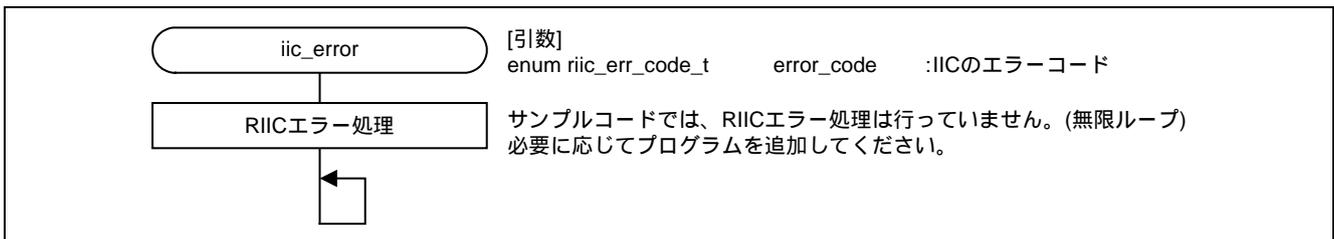


図 5.36 エラー処理

5.12.29 RIICn.EEIn 割り込み処理 (レベル割り込み)

図 5.37にRIICn.EEIn 割り込み処理 (レベル割り込み) のフローチャートを示します。

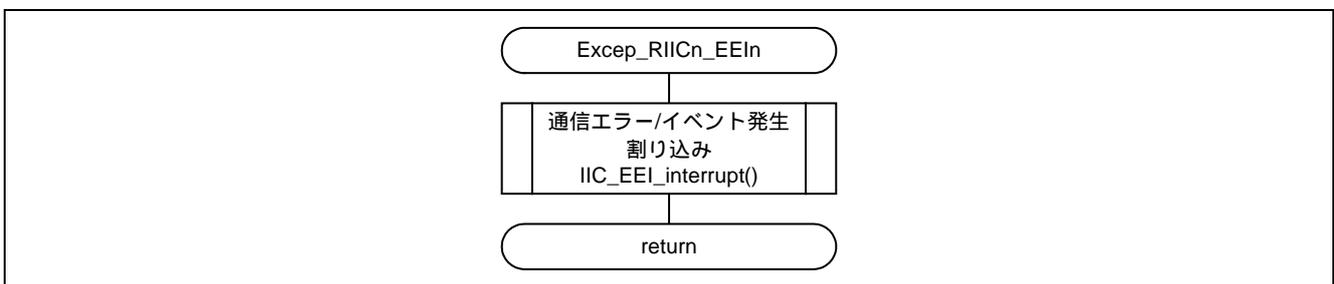


図 5.37 RIICn.EEIn 割り込み処理 (レベル割り込み)

5.12.30 RIICn.RXIn 割り込み処理 (エッジ割り込み)

図 5.38にRIICn.RXIn 割り込み処理 (エッジ割り込み) のフローチャートを示します。

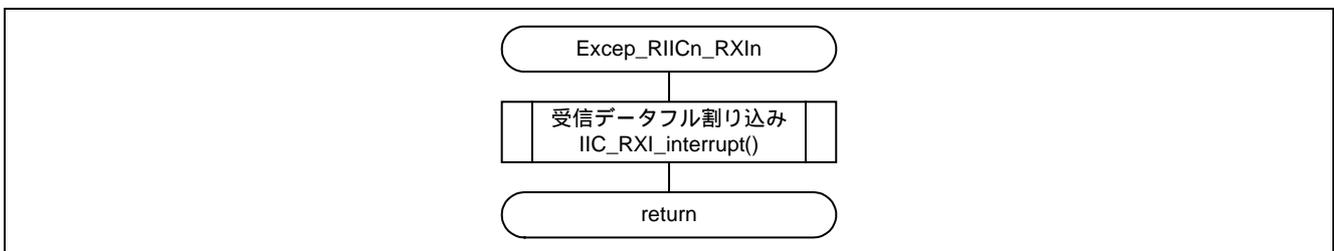


図 5.38 RIICn.RXIn 割り込み処理 (エッジ割り込み)

5.12.31 RIICn.TXIn 割り込み処理 (エッジ割り込み)

図 5.39にRIICn.TXIn 割り込み処理 (エッジ割り込み) のフローチャートを示します。

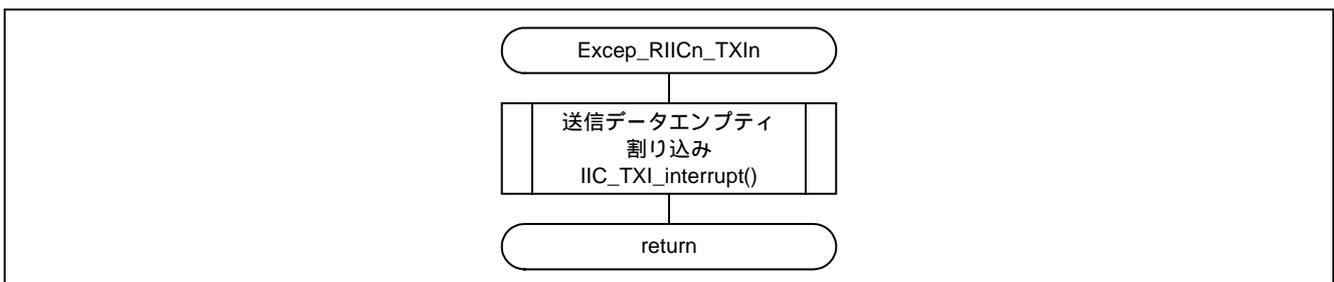


図 5.39 RIICn.TXIn 割り込み処理 (エッジ割り込み)

5.12.32 RIICn.TEIn 割り込み処理 (レベル割り込み)

図 5.40 にRIICn.TEIn 割り込み処理 (レベル割り込み) のフローチャートを示します。

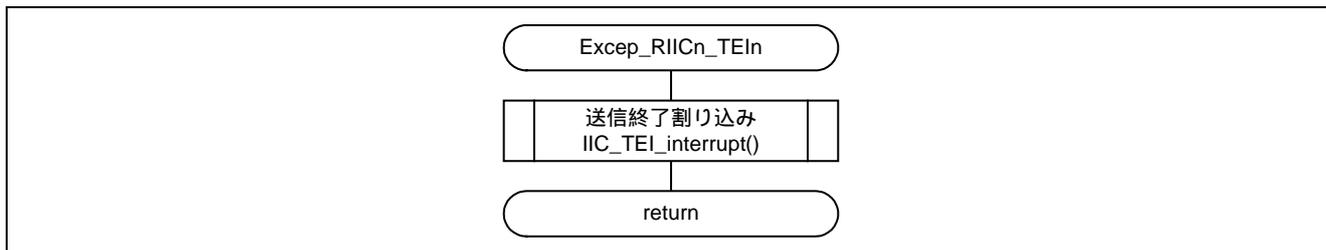


図 5.40 RIICn.TEIn 割り込み処理 (レベル割り込み)

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. 使用上のご注意

7.1 サンプルコード使用上のご注意

本アプリケーションノートのサンプルコードでは、RX220 グループまたは RX21A グループのデバイスを選択できます。デバイス選択の際には、下記の設定を行ってください。

1. High-performance Embedded Workshop のワークスペースウィンドウの[project]タブで、使用するデバイスのプロジェクトをアクティブプロジェクトに設定します。
アクティブプロジェクトを設定する方法は、最新の High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

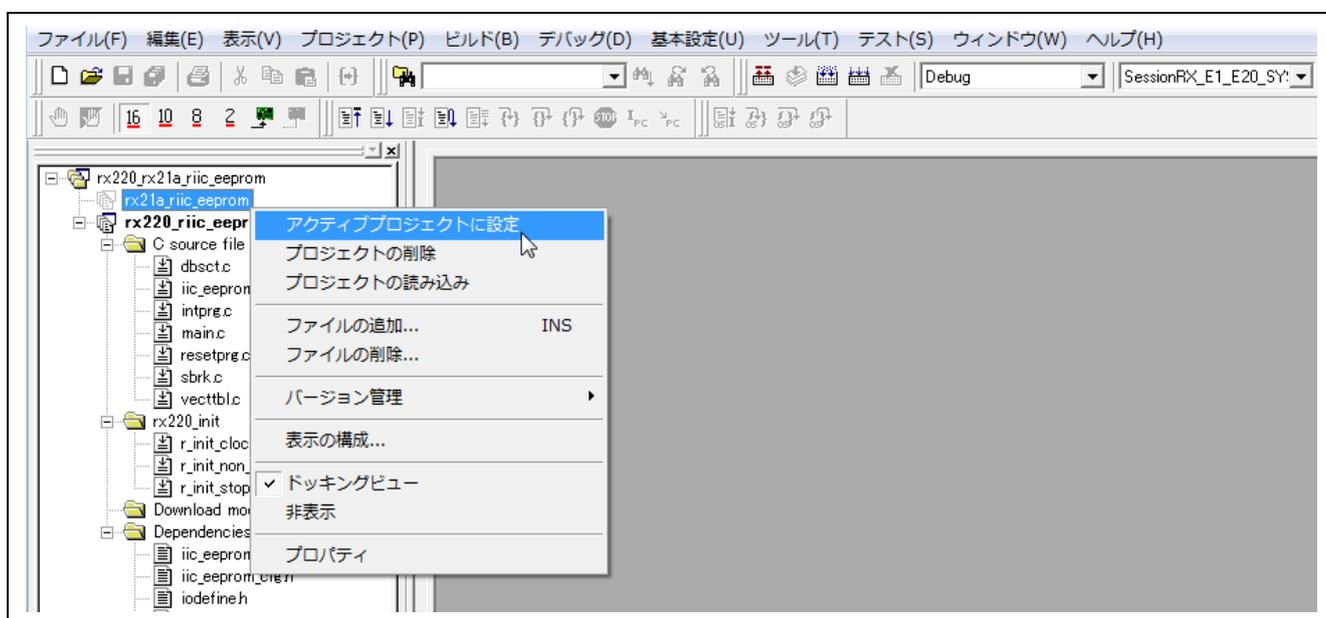


図 7.1 アクティブプロジェクトの設定方法

2. 使用するデバイスをコンフィグレーションファイル(iic_eeprom_cfg.h)で選択します。使用するデバイスのコードをアンコメント、使用しないデバイスのコードをコメントアウトしてください。

7.2 ボード使用上のご注意

本アプリケーションノート記載の使用ボードで、サンプルコードの動作を確認するには、下記の点にご注意ください。

使用ボード: 株式会社北斗電子製 HSB シリーズ マイコンボード (製品型名:HSBRX21AP-B)

株式会社北斗電子製 HSB シリーズ マイコンボードには、EEPROM が接続されていません。動作を確認するには、EEPROM を別途ご用意ください。

8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX220 グループ ユーザーズマニュアル ハードウェア編 Rev.1.00(R01UH0292JJ)

RX21A グループ ユーザーズマニュアル ハードウェア編 Rev.1.00(R01UH0292JJ)

(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。)

High-performance Embedded Workshop V.4.09 ユーザーズマニュアル Rev.1.00(R20UT0372JJ)

(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。)

データシート：EEPROM

R1EX24016ASAS0A データシート V.1.00 Rev.1.00 (rjj03c270_r1ex24016axxs0abs)

R1EX24512ASAS0A データシート V.1.00 Rev.1.00 (rjj03c249_r1ex24512axxs0abs)

(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/>

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	2014.05.07	—	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. 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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、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

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>