

V850ES/Jx3-L

I²C バス

EEPROM 制御編

R01AN1551JJ0100

Rev. 1.00

2013.03.28

要旨

本アプリケーションノートでは、I²C バス機能を用いた外付け EEPROM の制御方法を説明します。I²C バス接続した EEPROM へ 4 バイト (ワード) 長データの書き込みと読み出しを行います。また、書き込みと読み出しのデータ比較を行い、比較結果に対応した LED 表示制御を行います。

対象デバイス

V850ES/JC3-L

V850ES/JE3-L

V850ES/JF3-L

V850ES/JG3-L

V850ES/JG3-L USB

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
2.1	CubeSuite+ コード生成機能を使用するときの注意事項	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
5.1.1	EEPROM 書き込み処理	7
5.1.2	EEPROM 読み出し処理	8
5.2	オプション・バイトの設定一覧	9
5.3	定数一覧	9
5.4	変数一覧	11
5.5	関数一覧	12
5.6	関数仕様	13
5.7	フローチャート	18
5.7.1	IIC00 初期設定関数	19
5.7.2	メイン関数	20
5.7.3	EEPROM の選択	22
5.7.4	IIC0 スタート関数	22
5.7.5	スタート・コンディション発行関数	23
5.7.6	ストップ・コンディション発行関数	23
5.7.7	EEPROM 書き込み処理	24
5.7.8	EEPROM アドレス・チェック処理	25
5.7.9	スレーブ・アドレス算出処理	26
5.7.10	EEPROM 書き込み完了待ち処理	27
5.7.11	EEPROM 読み出し処理	28
5.7.12	EEPROM 読み出し完了待ち処理	29
5.7.13	スレーブ・アドレス送信完了処理	29
5.7.14	上位アドレス送信完了処理	30
5.7.15	リスタート処理	30
5.7.16	データ受信開始処理	31
5.7.17	データ受信処理	31
5.7.18	最終データ受信処理	32
5.7.19	データ送信開始処理	32
5.7.20	データ送信処理	33
5.7.21	MD_INTIIC0 割り込み処理	34
6.	サンプルコード	37
7.	参考ドキュメント	37

1. 仕様

本アプリケーションノートは、I²C バス機能を使用して、I²C バス接続した外付け EEPROM に書き込みと読み出しを行います。また書き込みデータと読み出しデータの比較結果に応じた LED 表示制御を行います。

- ・ I²C バス機能での通信制御は、API としての動作を意識して、割り込み関数で処理します。
- ・ 対象とする EEPROM は、2K~512K ビット製品から選択できるよう、複数の I²C 通信用制御パラメータを準備しています。本サンプルコードでは、16K ビットの R1EX24016A を選択し、その通信用制御パラメータを用いて、通信処理を実行します。
- ・ I²C 通信により、指定した EEPROM 内部アドレスに 4 バイトのデータを書き込みます。次に書き込んだ 4 バイトデータを読み出します。EEPROM へのアクセスは、4 バイトを 1 ブロックとしたブロック単位で扱います。
- ・ 最後に書き込みデータと読み出しデータが一致しているか比較を行います。データが一致している場合はポート制御により LED1 を点灯表示します。データが一致しない場合は LED2 を点灯表示します。

表 1.1 に使用する周辺機能と用途を、図 1.1 に I²C 通信の概要を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
I ² C00	I ² C マスタ送受信 (SCL00 端子と SDA00 端子を使用)

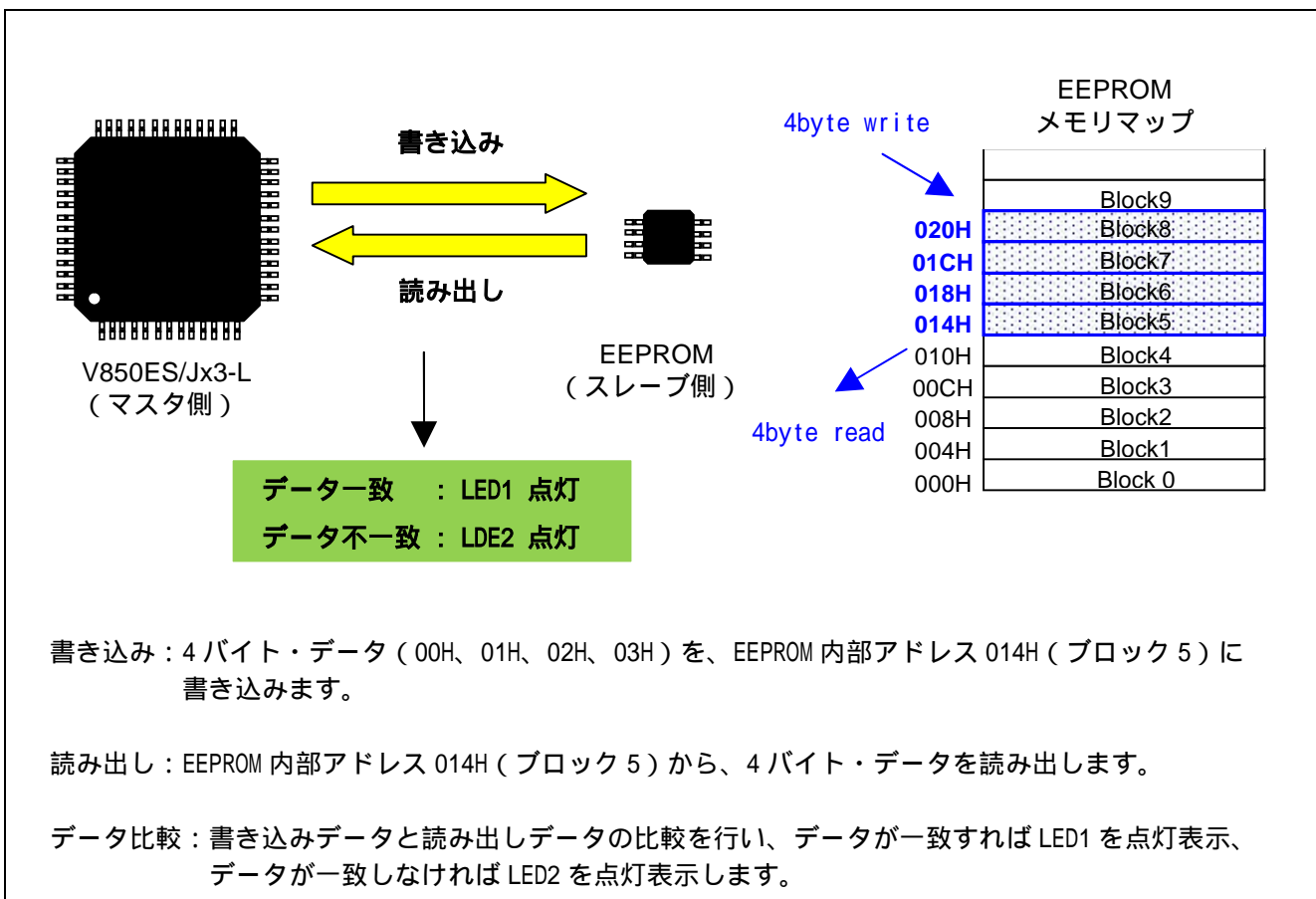


図 1.1 I²C 通信の概要

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	V850ES/JG3-L USB (μ PD70F3796GC)
動作周波数	<ul style="list-style-type: none"> ● CPU クロック : 16 MHz ● 周辺クロック : 16 MHz ● メイン・クロック発振周波数 : 6 MHz
動作電圧	3.3V (2.7V~3.6V) (CPU クロックが 16MHz 動作のとき)
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V1.03.00
C コンパイラ	ルネサス エレクトロニクス製 CA850 V3.50
使用ボード	QB-V850ESJG3LUSB-TB + 外付け EEPROM (R1EX24016)

注意 本サンプルコードは、V850ES/Jx3-L デバイスに対応していますが、動作確認条件と異なるデバイスを選択される場合は、コード生成機能などを利用して、動作周波数、動作電圧などデバイスに適した条件でご使用ください。

2.1 CubeSuite+ コード生成機能を使用するときの注意事項

本サンプルコードでは、コード生成機能を利用していますが、一部のソースファイルを追加/削除しています。本プロジェクトに対して、コード生成を実行した場合は、以下の操作を行ってください。

- ・「CG_serial_user.c」をプロジェクトからはずしてください。
- ・「IIC_EEPROM.c」と「CG_ik.dir」をプロジェクトに追加してください(既存のファイルを追加)。

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。

V850ES/JG3-L 初期設定 (U19479JJ) 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

図 4.1 に本アプリケーションノートで使用するハードウェア構成例を示します。

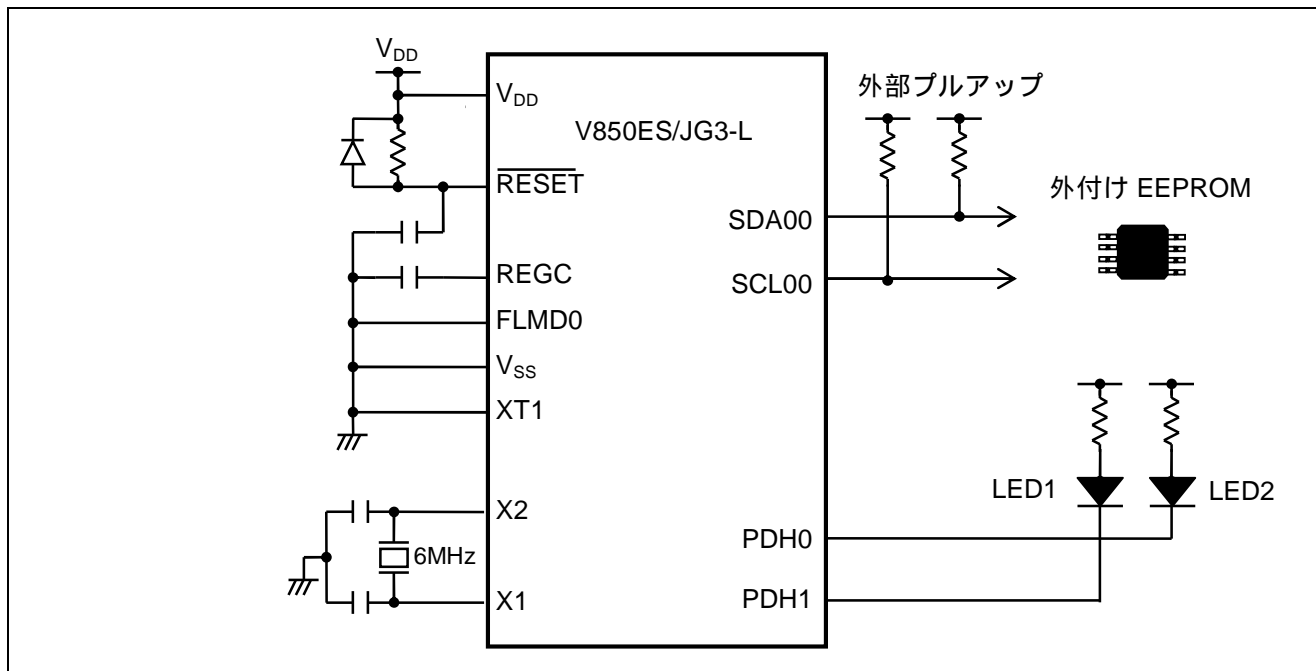


図 4.1 ハードウェア構成

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。

- 2 EV_{DD} 端子、 AV_{REF0} 端子は V_{DD} と同電位にしてください。
- 3 EV_{SS} 端子は GND と同電位にしてください。
- 4 REGC はコンデンサ（推奨値：4.7 μ F）を介し、GND に接続してください。
- 5 FLMD0 端子は、通常動作モード時は GND に接続してください。

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
P39/SCL00	入出力	I ² C00 のシリアル・クロック出力端子
P38/SDA00	入出力	I ² C00 のシリアル・データ送受信端子
PDH1	入出力	LED1 点灯表示用（比較データ一致）
PDH0	入出力	LED2 点灯表示用（比較データ不一致）

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、I²C00 による I²C マスタ送受信機能を使用して EEPROM の制御(書き込み、読み出し)を行います。また EEPROM への書き込み / 読み出しデータの比較を行い、結果を LED 表示します。

< 設定条件 >

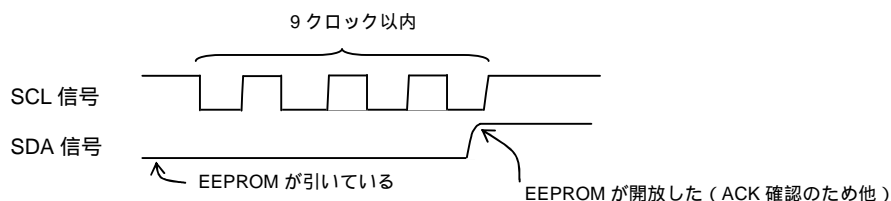
- I²C0 分周クロックを動作許可、 $f_{xx} / 2$ に設定します。
- 動作モードを高速モードに、またデジタル・フィルタ・オンに設定します。
- 転送クロックを 333kHz に設定します。
- 自局アドレスを 80H に設定します。
- 動作許可後にスタート・コンディションを生成可能に設定します。
- ストップ・コンディション検出による割り込み要求発生を禁止に設定します。
- 9 クロック目の立ち下がりで割り込み要求発生に設定します。
- アクノリッジを許可に設定します。
- P39/SCL00 端子を転送クロック出力用に、P38/SDA00 端子をデータ送信 / 受信用に設定します。

- (1) I²C00 の初期設定を行います。
- (2) EEPROM への書き込み処理ウエイト用に、タイマ TMM0 を 100 μ s のインターバル設定とします。
- (3) 使用する EEPROM (16K ビット) の通信用制御パラメータを指定します。
- (4) I²C00 を動作許可にします。
- (5) 4 バイトの書き込みデータ (00H、01H、02H、03H) を作成します。
- (6) EEPROM アクセス用パラメータ(構造体 g_PARA1)にスレーブ・アドレス(A0H)と、アクセス先の EEPROM 内アドレス(ブロック 5 (014H))を設定します。
- (7) EEPROM の書き込み処理を行います。
- (8) 書き込んだデータを EEPROM から読み出し処理を行います。
- (9) 書き込み / 読み出しデータの比較を行い、結果(一致 / 不一致)を LED 点灯表示します。

注意 本サンプルコードは、V850ES/JG3-L の I²C00 を利用した I²C バスによる EEPROM (R1EX24016) の制御例を示すものです。使用するチャンネルや EEPROM を変更した場合には十分に評価してご利用ください。

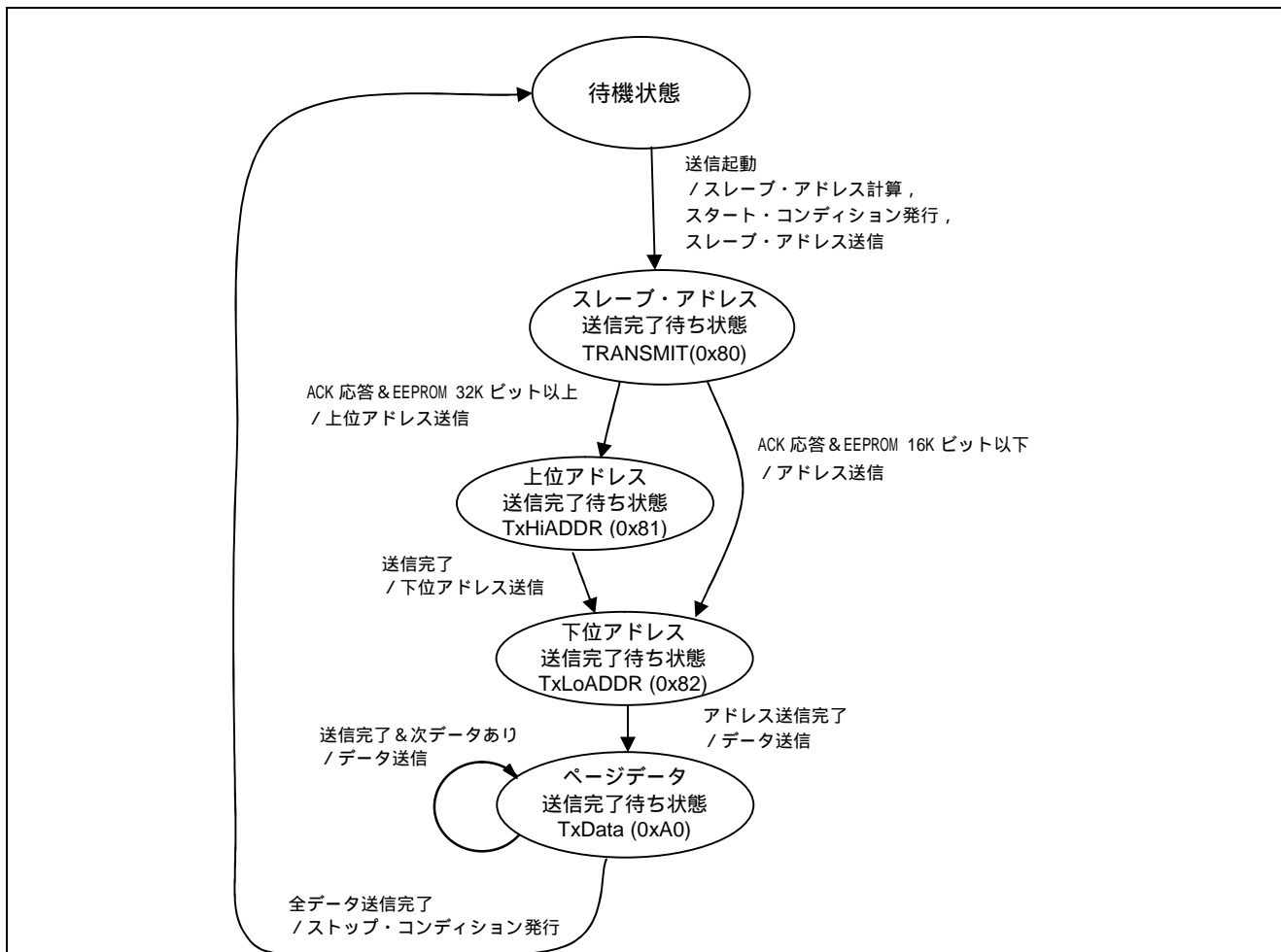
コラム I²C バスを使用して、最初にストップ・コンディションを発行してバスを解放状態にする場合、スレーブとなる EEPROM が SDA 信号をロウに引いていると、ストップ・コンディションの発行が行えないため注意が必要です。

このような場合は、SCL 信号をプログラムで制御して疑似的な I²C バス・クロックを最大 9 クロック分発生させることで、正常な EEPROM の SDA 信号ドライブを止め、SDA 信号をハイにすることが可能です。



5.1.1 EEPROM 書き込み処理

図 5.1 に EEPROM 書き込み処理の状態遷移図を示します。各状態は変数 g_comstatus の値 (TRANSMIT(0x80) など) で定義されています。I²C 通信では、転送終了割り込み関数内で g_comstatus の値を参照することで現在の状態を判断し、次の処理を実行していきます。



注意 エラー処理は省略しています。各状態の 3 行目は g_comstatus の値です。

I²C 通信では割り込み関数内で、現在の状態を判断して次の処理を実行していきます。

転送終了割り込み関数	
g_comstatus = TxHiADDR	→ 下位アドレス送信 (g_comstatus TxLoADDR)
TxLoADDR	→ データ送信 (g_comstatus TxData)
TxData	

図 5.1 EEPROM 書き込み処理の状態遷移図

5.1.2 EEPROM 読み出し処理

図 5.2 に EEPROM 読み出し処理の状態遷移図を示します。EEPROM 書き込み処理と同様に、各状態は変数 `g_comstatus` の値(`RECEIVE(0xC0)` など)で定義されています。読み出し処理では、セル・アドレス指定(EEPROM に対して、読み出したい EEPROM 内部アドレスを通知する)とデータ読み出し(転送方向をマスタ受信として、リスタート・コンディションを発行する)の 2 種類に分かれます。

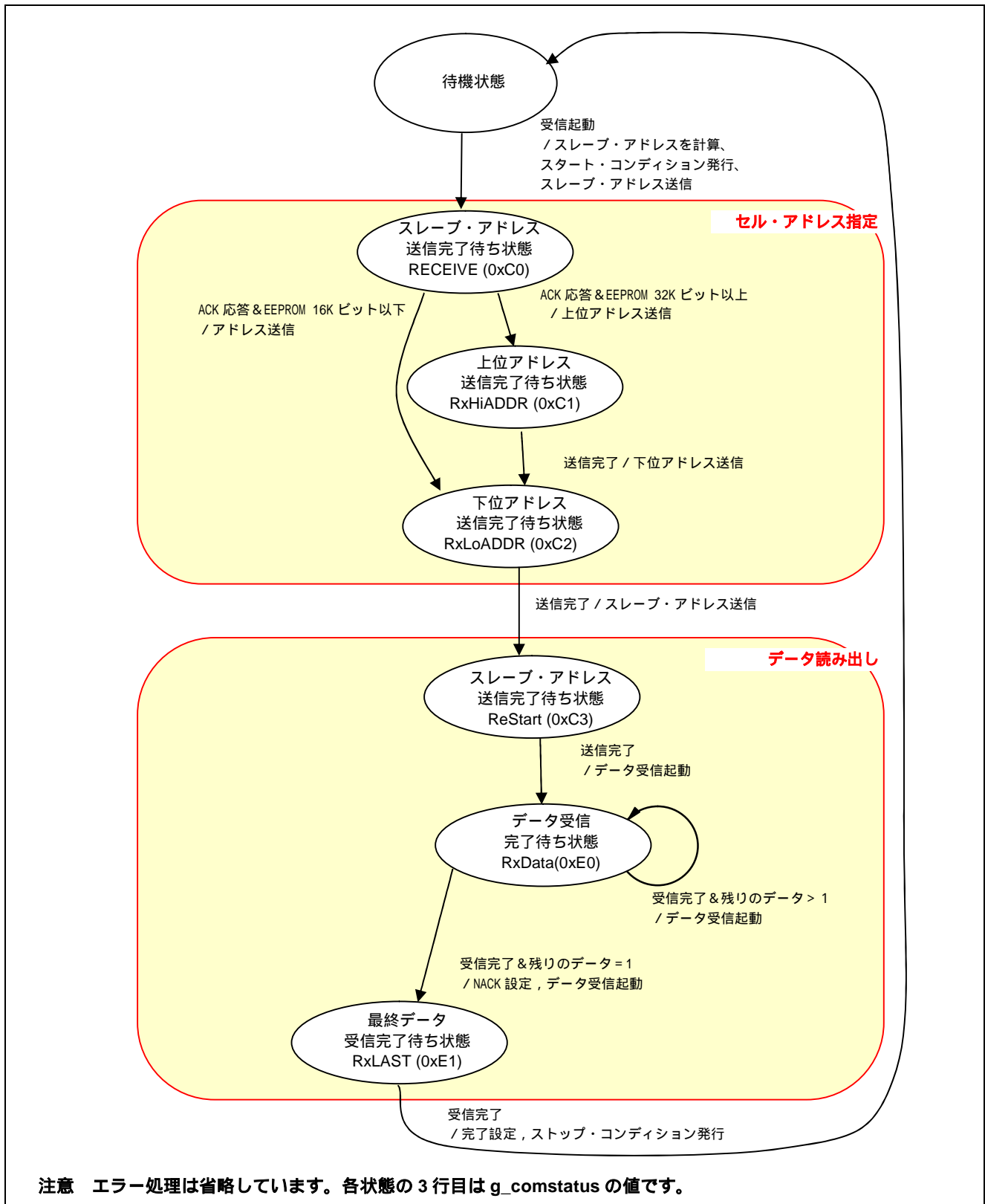


図 5.2 EEPROM 読み出し処理状態遷移図

5.2 オプション・バイトの設定一覧

表 5.1 にオプション・バイト設定を示します。

表 5.1 オプション・バイト設定

アドレス	設定値	内容
0000007AH	00000110B	内蔵発振器のソフトウェアによる停止不可 (発振安定時間 10.92ms ($f_x = 6\text{MHz}$))
0000007BH -0000007FH	00000000B	007BH-007FH 番地には 0x00 を設定してください。

5.3 定数一覧

表 5.2 にサンプルコードで使用する定数を示します。

表 5.2 サンプルコードで使用する定数 (1/2)

定数名	設定値	内容
BLOCK_SIZE	4	EEPROM アクセス時のデータ・ブロック単位 (バイト)
MEMORY_2K	0x0001	2K ビット EEPROM の容量 (256 バイト単位)
MEMORY_4K	0x0002	4K ビット EEPROM の容量 (256 バイト単位)
MEMORY_8K	0x0004	8K ビット EEPROM の容量 (256 バイト単位)
MEMORY_16K	0x0008	16K ビット EEPROM の容量 (256 バイト単位)
MEMORY_32K	0x0010	32K ビット EEPROM の容量 (256 バイト単位)
MEMORY_64K	0x0020	64K ビット EEPROM の容量 (256 バイト単位)
MEMORY_128K	0x0040	128K ビット EEPROM の容量 (256 バイト単位)
MEMORY_256K	0x0080	256K ビット EEPROM の容量 (256 バイト単位)
MEMORY_512K	0x0100	512K ビット EEPROM の容量 (256 バイト単位)
BLOCK_2K	右記参照	MEMORY_2K * 256 / BLOCK_SIZE
BLOCK_4K	右記参照	MEMORY_4K * 256 / BLOCK_SIZE
BLOCK_8K	右記参照	MEMORY_8K * 256 / BLOCK_SIZE
BLOCK_16K	右記参照	MEMORY_16K * 256 / BLOCK_SIZE
BLOCK_32K	右記参照	MEMORY_32K * 256 / BLOCK_SIZE
BLOCK_64K	右記参照	MEMORY_64K * 256 / BLOCK_SIZE
BLOCK_128K	右記参照	MEMORY_128K * 256 / BLOCK_SIZE
BLOCK_256K	右記参照	MEMORY_256K * 256 / BLOCK_SIZE
BLOCK_512K	右記参照	MEMORY_512K * 256 / BLOCK_SIZE
ADDR0BIT	0b00000000	スレーブ・アドレスをセル・アドレスで使用しない
ADDR1BIT	0b00000001	A8 をスレーブ・アドレスのビット 1 で指定
ADDR2BIT	0b00000011	A9, 8 をスレーブ・アドレスのビット 2, 1 で指定
ADDR3BIT	0b00000111	A10 ~ 8 をスレーブ・アドレスのビット 3 ~ 1 で指定
I2C_OK	0x00	正常完了
PARA_ERR	0x20	パラメータ・エラー
COMP_ERR	0x21	書き込み / 読み出しデータの比較一致エラー
NO_ACK1	0x40	スレーブ・アドレスへの ACK 応答なし
NO_ACK2	0x41	EEPROM アドレスへの ACK 応答なし
NO_ACK3	0x42	送信データへの ACK 応答なし
SVAMSK	0b11111110	スレーブ・アドレスのビット 0 マスク用データ

表 5.2 サンプルコードで使用する定数 (2 / 2)

定数名	設定値	内容
R1EX24002A	0x00	使用する EEPROM の指定を行うための定数 列挙型の定数 eeprom_name で定義しており, EEPROM の パラメータを eeprom_info 型の構造体 EEPROM_ADDRESS から参照するために使用
R1EX24004A	0x01	
R1EX24008A	0x02	
R1EX24016A	0x03	
R1EX24032A	0x04	
R1EX24064A	0x05	
R1EX24128B	0x06	
R1EX24256B	0x07	
R1EX24512B	0x08	

5.4 変数一覧

表 5.3 にグローバル変数を示します。

表 5.3 グローバル変数

Type	Variable Name	Contents	Function Used
構造体 eeprom_paraA16	g_PARAI	EEPROM アクセス指定用パラメータ	main() check_EEPROM_Addr()
uint8_t	g_comstatus	動作情報 / 結果フラグ	main() check_EEPROM_Addr() R_EEPROM_R() R_EEPROM_wait_read() R_IIC0_Tx_addr1() R_IIC0_Tx_addr2() R_IIC0_Rx_RST() R_IIC0_RxData_ST() R_IIC0_RxData() R_IIC0_Rx_LastData() R_EEPROM_W() R_EEPROM_wait_write() R_IIC0_TxDataST() R_IIC0_TxData() MD_INTIIC0()
uint8_t 配列 (BLOCK_SIZE)	g_data_bufferW	書き込みデータバッファ	main()
uint8_t 配列 (BLOCK_SIZE)	g_data_bufferR	読み出しデータバッファ	main()
uint8_t	g_data_counter	データ用カウンタ	R_IIC0_TxDataST() R_IIC0_TxData() R_IIC0_RxData_ST() R_IIC0_RxData() R_IIC0_Rx_LastData()
uint8_t	g_celladdr	EEPROM 内部アドレス	
uint8_t	g_eeprom_type	使用する EEPROM 番号	R_device_select()
構造体 eeprom_paraA16 uint8_t slaveaddr; uint16_t block_num;	g_PARAA	EEPROM アクセス用パラメータ	R_EEPROM_R() R_IIC0_Tx_addr1() R_IIC0_Tx_addr2() R_IIC0_Rx_RST() R_IIC0_RxData_ST() R_IIC0_RxData() R_IIC0_Rx_LastData() R_EEPROM_W() R_IIC0_TxDataST() R_IIC0_TxData()
構造体 eeprom_paraA16	g_PARAC	作業用 EEPROM アクセス用パラメータ (g_PARAA のコピー)	check_EEPROM_Addr() R_EEPROM_R() R_EEPROM_W() R_IIC0_TxData() get_slave_Addr()
構造体 eeprom_info uint16_t rom_size; uint16_t total_block uint8_t addr_mask;	EEPROM_Info	使用する EEPROM のパラメータ保持用 (処理用)	R_device_select() R_IIC0_Tx_addr1() get_slave_Addr()

5.5 関数一覧

表 5.4 に関数一覧を示します。

表 5.4 関数一覧

関数名	概要
R_device_select	使用する EEPROM を指定します。
R_EEPROM_R	引数で渡されたポインタで示された構造体からアクセス用パラメータに従って EEPROM からデータを読み出します。
R_EEPROM_wait_read	EEPROM の読み出し完了を待ちます。
R_EEPROM_W	引数で渡されたポインタで示された構造体からアクセス用パラメータに従って EEPROM にデータを書き込みます。
R_EEPROM_wait_write	EEPROM への書き込み完了を待ちます。
check_EEPROM_Addr	指定されたパラメータが EEPROM の容量を超えないかをチェックします。容量内であれば、アクセス用パラメータをコピーします。
get_slave_Addr	4K~16K ビットの場合にセルのアドレスの上位をスレーブ・アドレスに組み込みます。
R_IIC0_Tx_addr1	スレーブ・アドレスの送信完了処理を行い、EEPROM のセルのアドレスを送信します。
R_IIC0_Tx_addr2	EEPROM のセルのアドレスが 2 バイトのときに、下位アドレスを送信します。
R_IIC0_Rx_RST	EEPROM のセルのアドレスを送信完了後にデータ読みだしのために受信モードでリスタートします。
R_IIC0_RxData_ST	データ受信のため、ダミーデータを IIC0 レジスタに書き込み、データ受信を起動します。
R_IIC0_RxData	受信したデータをバッファに格納し、次のデータ受信を起動します。また最後のデータ受信前にアクノリッジを禁止設定にします。
R_IIC0_Rx_LastData	最後の受信データをバッファに格納し、ストップ・コンディションを発行して処理を終了します。
R_IIC0_TxDataST	EEPROM への書き込みデータの送信を開始します。
R_IIC0_TxData	データ送信完了し、次の書き込みデータある場合は、データ送信を起動します。全ての書き込みデータが送り終われば、ストップ・コンディションを発行し、EEPROM に書き込みを指示します。
R_IIC00_StartCondition	スタート・コンディションを発行し、スタート・コンディション検出を待ちます。
R_IIC00_StopCondition	ストップ・コンディションを発行し、ストップ・コンディション検出を待ちます。
MD_INTIIC0	IIC00 の転送終了割り込みで、スレーブの ACK 応答を確認し、通信状態に合わせて次の処理に割り振ります。
R_IIC00Start	IIC00 の動作許可をします。

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] R_device_select

概要	使用する EEPROM を指定します。
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	MD_STATUS R_device_select(enum eeprom_name);
説明	引数で指定された EEPROM のパラメータを構造体 EEPROM_Info にコピーします。
引数	[EEPROM の名前] 列挙型定数 eeprom_name に登録された名前
リターン値	[I2C_OK]の場合 : 正常終了 [PARA_ERR]の場合 : 指定した名前の間違い
備考	なし

[関数名] R_IIC00_Start

概要	IIC00 の動作許可
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	Void R_IIC00_Start(void)
説明	IIC00 の動作許可を行います。
引数	なし
リターン値	なし
備考	なし

[関数名] R_EEPROM_R

概要	EEPROM からの読み出し開始要求処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	void R_EEPROM_R(struct eeprom_paraA16 *PARA);
説明	引数のポインタで示された構造体のパラメータで指定した読みだしを行います。
引数	*PARA eeprom_paraA16 型の構造体へのポインタ
リターン値	なし
備考	読みだし完了待ちと結果は R_EEPROM_wait_read で行います。

[関数名] R_EEPROM_wait_read

概要	EEPROM からの読みだし完了待ち処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	MD_STATUS R_EEPROM_wait_read (void);
説明	R_EEPROM_R で起動した読みだしの完了待ちを行います。
引数	なし
リターン値	[I2C_OK]の場合 : 読みだし正常終了 : [PARA_ERR]の場合 : 指定されたパラメータが EEPROM の範囲外 [NO_ACK1]の場合 : スレーブ・アドレスへの ACK 応答なし [NO_ACK2]の場合 : EEPROM アドレスへの ACK 応答なし
備考	

[関数名] R_EEPROM_W

概要	EEPROM への書き込み開始処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	void R_EEPROM_W(struct eeprom_paraA16 *PARA);
説明	引数のポインタで示された構造体のパラメータで指定した書き込みを行います。
引数	*PARA eeprom_paraA16 型の構造体へのポインタ
リターン値	なし
備考	書き込み完了待ちと結果は R_EEPROM_wait_write で行います

[関数名] R_EEPROM_wait_write

概要	EEPROM への書き込み完了待ち処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	MD_STATUS R_EEPROM_wait_write(void);
説明	R_EEPROM_W で起動した EEPROM への書き込みの完了を待ちます。
引数	なし -
リターン値	[I2C_OK]の場合 : 書き込み正常終了 [PARA_ERR]の場合 : 指定されたパラメータが EEPROM の範囲外 [NO_ACK1]の場合 : スレーブ・アドレスへの ACK 応答なし [NO_ACK2]の場合 : EEPROM アドレスへの ACK 応答なし [NO_ACK3]の場合 : 送信データへの ACK 応答なし
備考	なし

[関数名] check_EEPROM_Addr

概要	EEPROM のアクセス領域 (ブロック番号) のチェック処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	static MD_STATUS check_EEPROM_Addr(struct eeprom_paraA16 *PARA);
説明	EEPROM をアクセスするパラメータから読み / 書きする領域が EEPROM の中に納まっているかをチェックします。
引数	*PARA eeprom_paraA16 型の構造体へのポインタ
リターン値	[I2C_OK]の場合 : アクセスする領域は EEPROM 内に収まっている。 [PARA_ERR]の場合 : アクセスする領域は EEPROM に収まりきれない。
備考	なし

[関数名] get_slave_Addr

概要	EEPROM のスレーブ・アドレス算出処理
ヘッダ	r_cg_userdefine.h
宣言	static void get_slave_Addr(void);
説明	4K ~ 16K ビット EEPROM でセルのアドレスの上位 1 ~ 3 ビットでスレーブ・アドレスを修飾します。
引数	なし -
リターン値	なし
備考	構造体 g_PARAC のメンバ eepromaddr の値によりメンバ slaveaddr を修飾します。

[関数名] R_IIC0_Tx_addr1

概要	EEPROM のアドレス送信処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_Tx_addr1(void);
説明	32K ビット以上の EEPROM でセルの上位アドレスを送信します。16K ビット以下では 1 バイトのセルのアドレスを送信します。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC0_Tx_addr2

概要	EEPROM のセルの下位アドレスの送信処理。
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_Tx_addr2(void);
説明	32K ビット以上の EEPROM で、アドレスの下位バイトを送信します。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC0_Rx_RST

概要	EEPROM 読み出しのためのリスタート処理。
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_Rx_RST(void);
説明	データ読み出しのために、受信モードでリスタートします。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC0_RxData_ST

概要	データ受信開始処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_RxData_ST(void);
説明	読み出しモードでのスレーブ・アドレス送信完了で、受信を起動 (IIC0 レジスタにダミー・データ書き込み) します。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC0_RxData

概要	データ受信処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_RxData (void);
説明	受信したデータをバッファに格納し、次データの受信を起動します。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC0_Rx_LastData

概要	最終データ受信完了処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_Rx_LastData (void);
説明	受信したデータをバッファに格納し、ストップ・コンディションを発行して処理を終了します。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC0_TxDataST

概要	データ送信開始処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_TxDataST (void);
説明	EEPROM アドレス送信完了で、データ送信を開始します。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC0_TxData

概要	データ送信処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC0_TxData (void);
説明	1 バイトのデータ転送が完了したら、次のデータを送信します。全ての書き込みデータが送信完了した場合は、ストップ・コンディションを発行し、EEPROM に書き込みを指示します。残りの書き込みデータがあれば、データ送信をします。
引数	なし -
リターン値	なし
備考	MD_INTIIC0 割り込み処理で使用

[関数名] R_IIC00_StartCondition

概要	スタート・コンディション発行処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IIC00_StartCondition(void);
説明	スタート・コンディションの発行を行い、スタート・コンディションの検出を待ちます。
引数	なし -
リターン値	なし
備考	なし

[関数名] R_IIC00_StopCondition

概要	ストップ・コンディション生成
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	Void R_IIC00_StopCondition(void)
説明	IIC0 のストップ・コンディションの発行と、ストップ・コンディションの検出を待ちます。
引数	なし
リターン値	なし
備考	なし

[関数名] MD_INTIIC0

概要	IIC0 転送終了割り込み
ヘッダ	r_cg_userdefine.h
宣言	__interrupt void MD_INTIIC0(void);
説明	INTIIC0 割り込み要求で起動され、通信の状況に応じて必要な処理を実行します。EEPROM の読み出し完了以外で、スレーブからの NACK 応答を検出したときには処理の状態に応じて g_comstatus にエラー・フラグを設定します。
引数	なし
リターン値	なし
備考	

5.7 フローチャート

図 5.3 に本アプリケーションノートの全体フローを示します。

備考 初期設定関数については、IIC00 以外の関数については説明を省略しています。

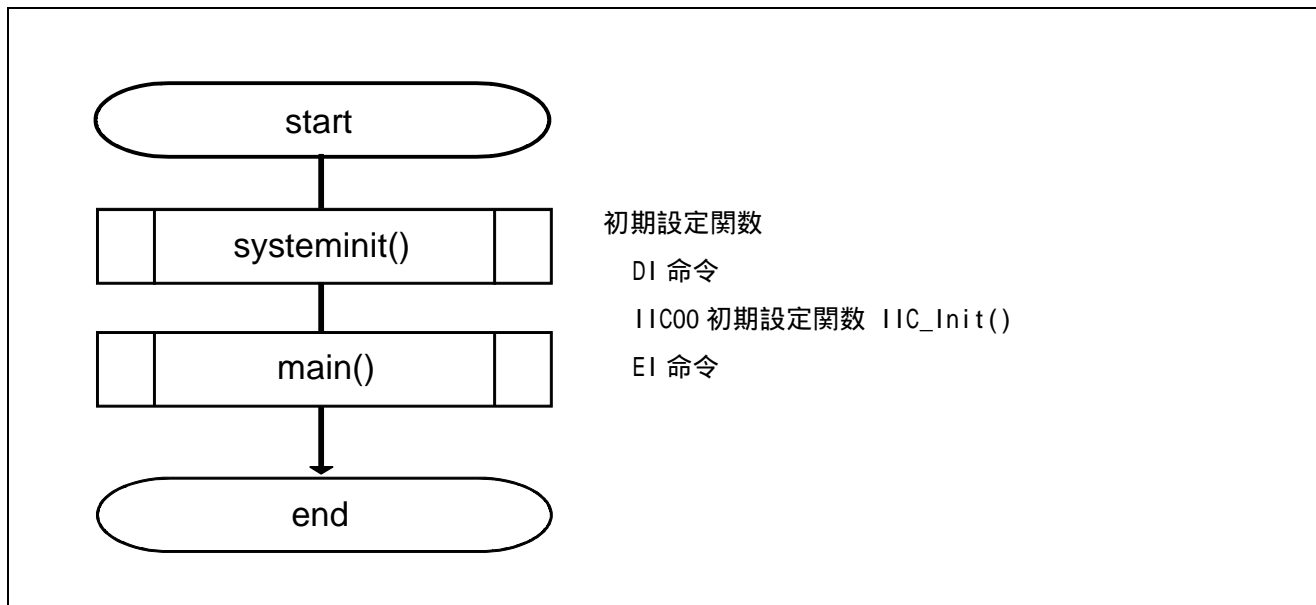


図 5.3 全体フロー

5.7.1 IIC00 初期設定関数

図 5.4 に IIC00 初期設定関数のフローチャートを示します。

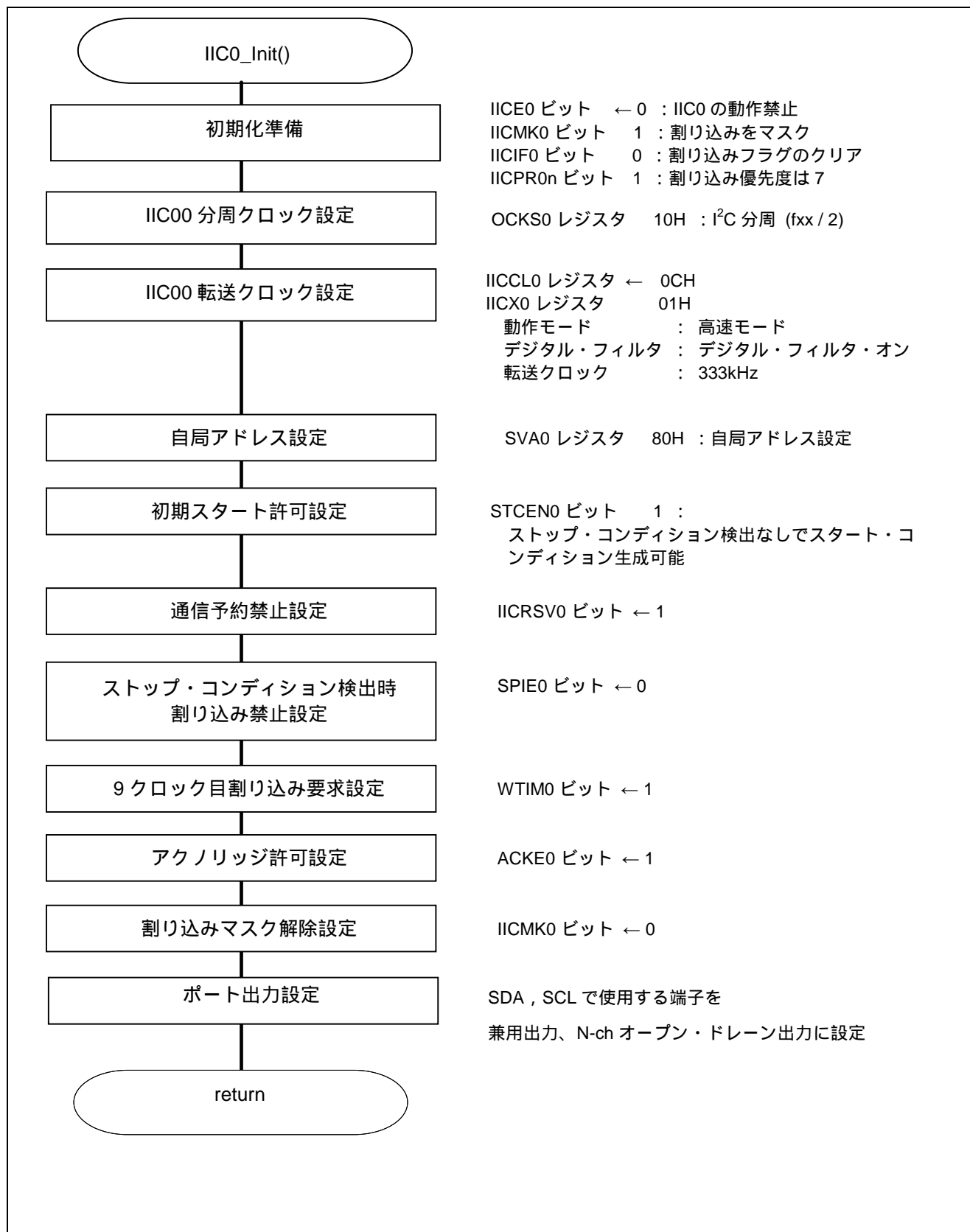


図 5.4 IIC0 の設定

5.7.2 メイン関数

図 5.5, 図 5.6 にメイン関数のフローチャートを示します。メイン関数では 16K ビット EEPROM への読み書きのテストを行っています。

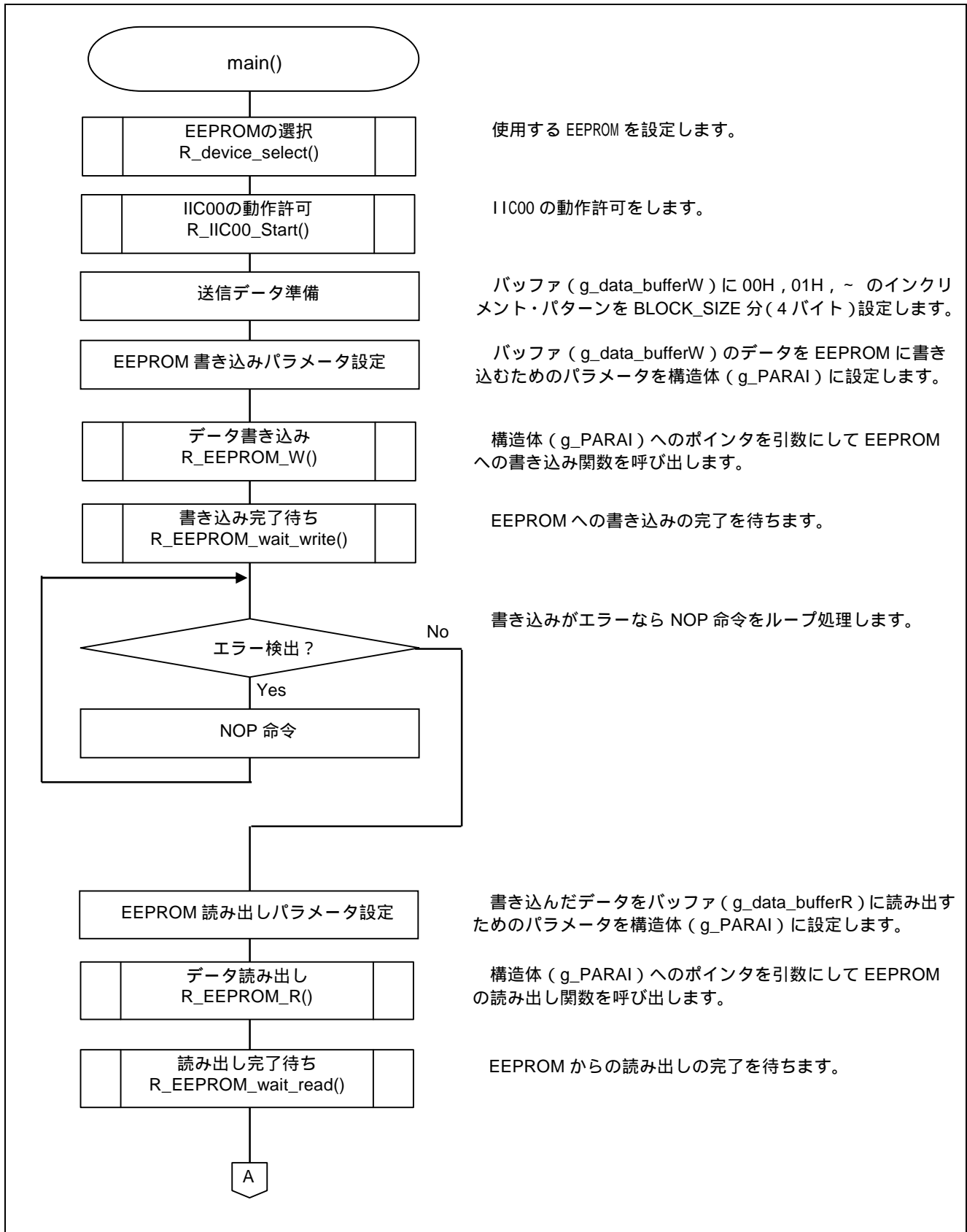


図 5.5 メイン処理 (1/2)

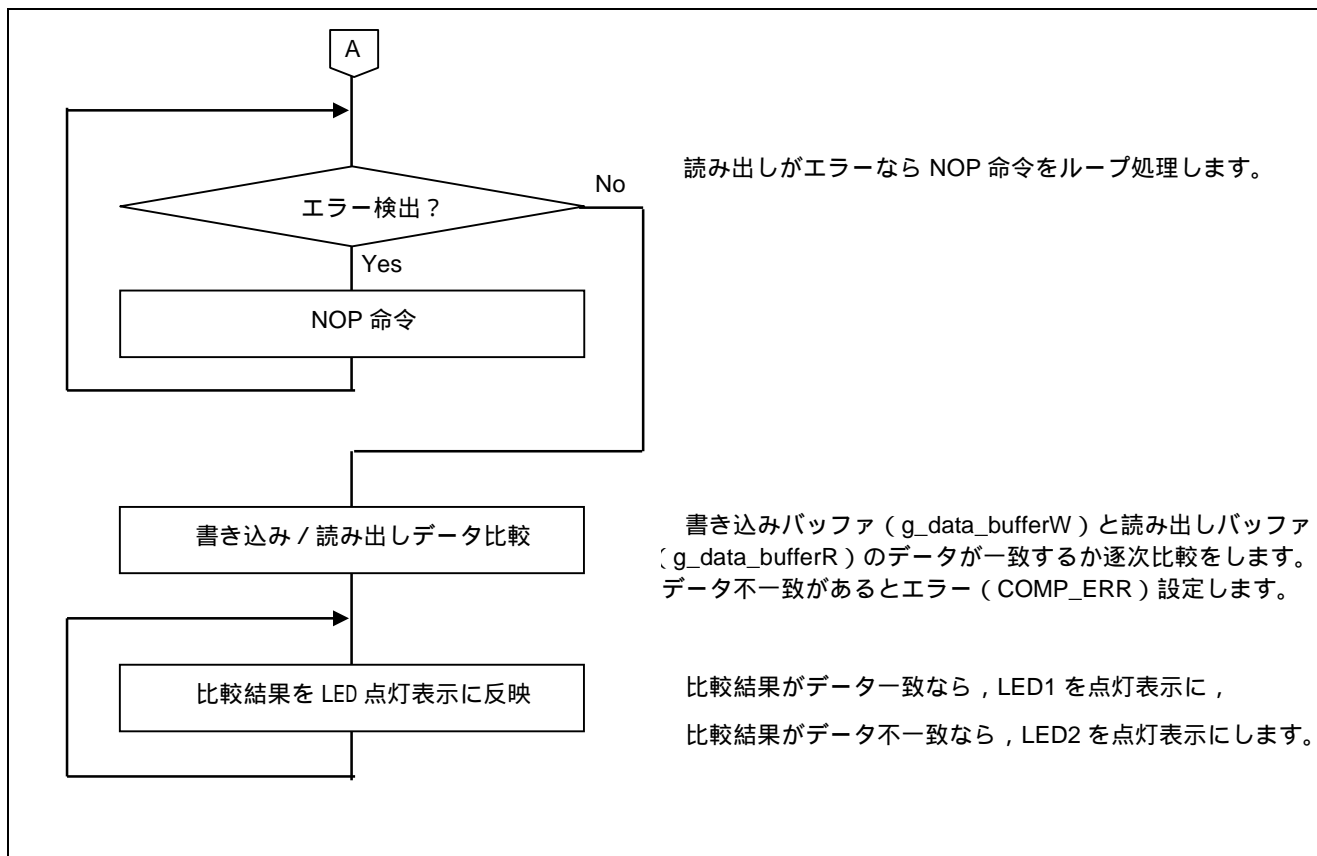


図 5.6 メイン関数 (2/2)

5.7.3 EEPROM の選択

図 5.7 に EEPROM の選択のフローチャートを示します。

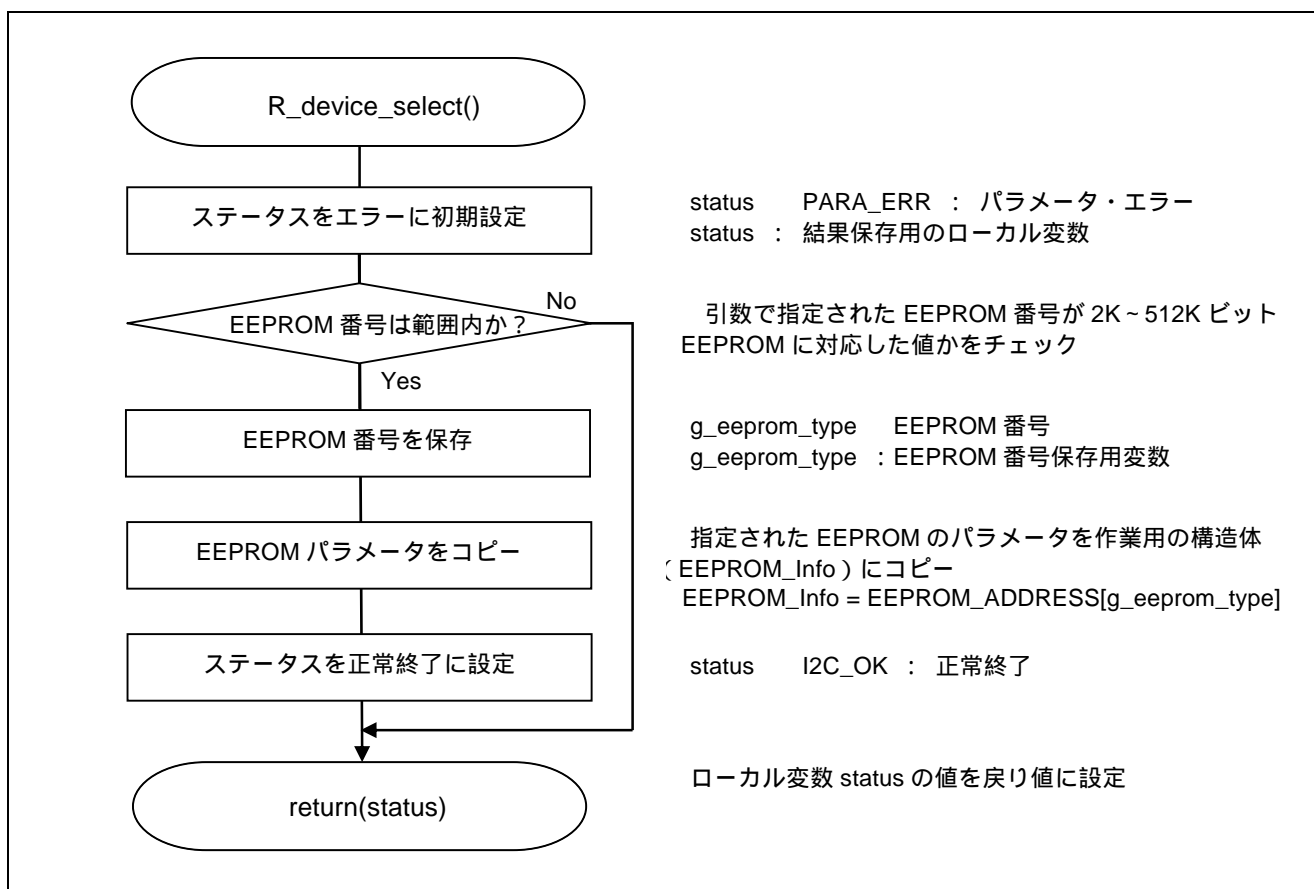


図 5.7 EEPROM の選択

5.7.4 IIC0 スタート関数

図 5.8 に IIC0 スタート関数のフローチャートを示します。

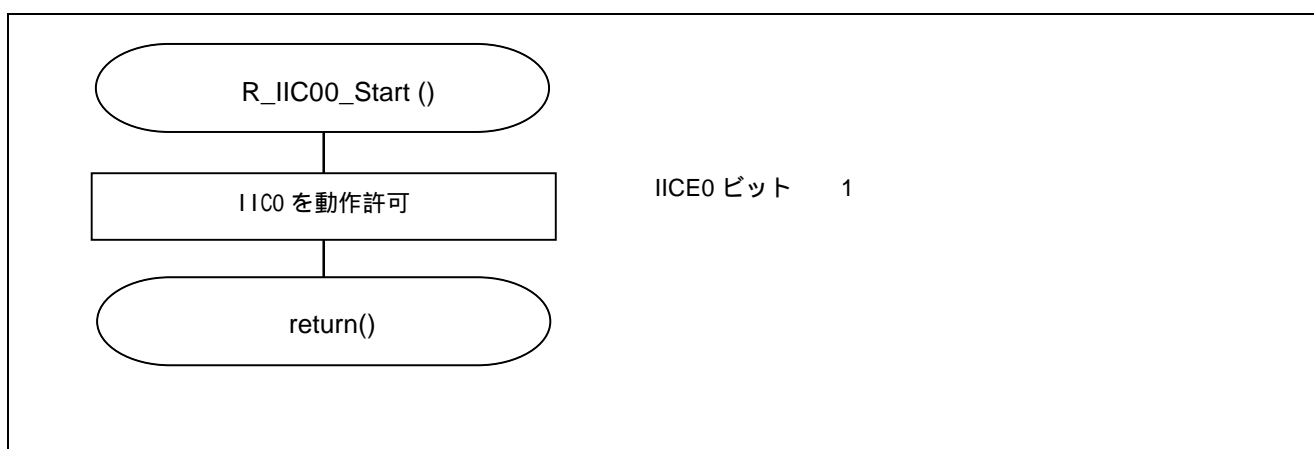


図 5.8 IIC0 スタート発行関数

5.7.5 スタート・コンディション発行関数

図 5.9 にスタート・コンディション発行関数のフローチャートを示します。

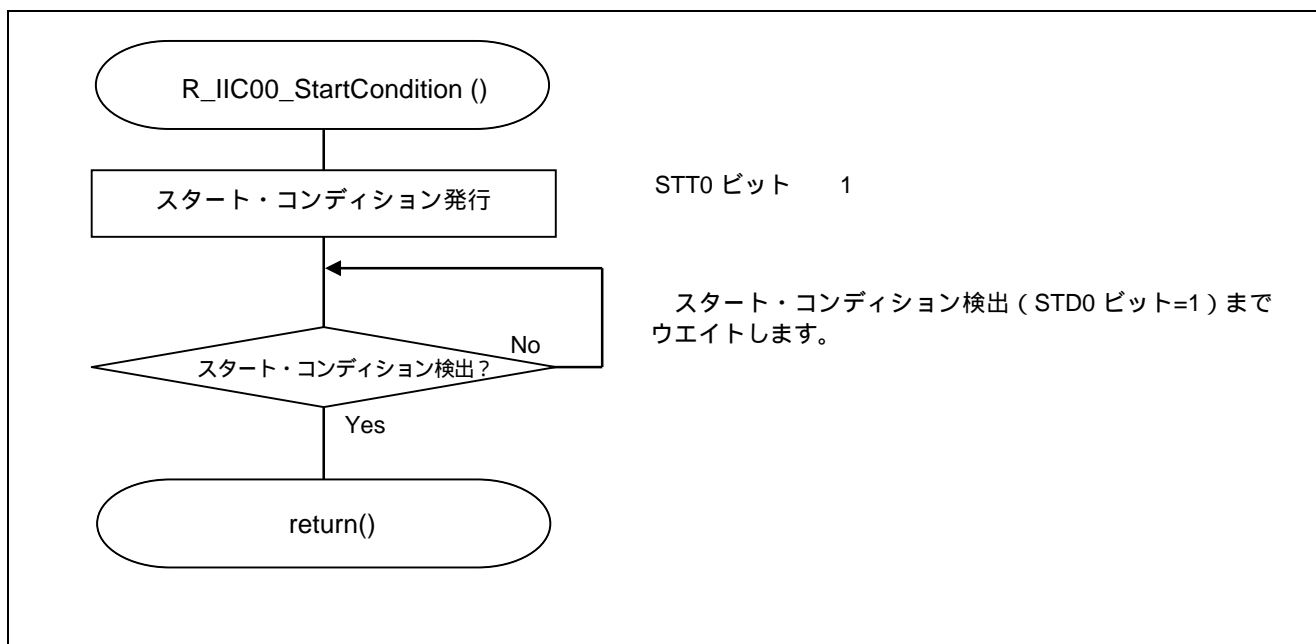


図 5.9 スタート・コンディション発行関数

5.7.6 ストップ・コンディション発行関数

図 5.9 にストップ・コンディション発行関数のフローチャートを示します。

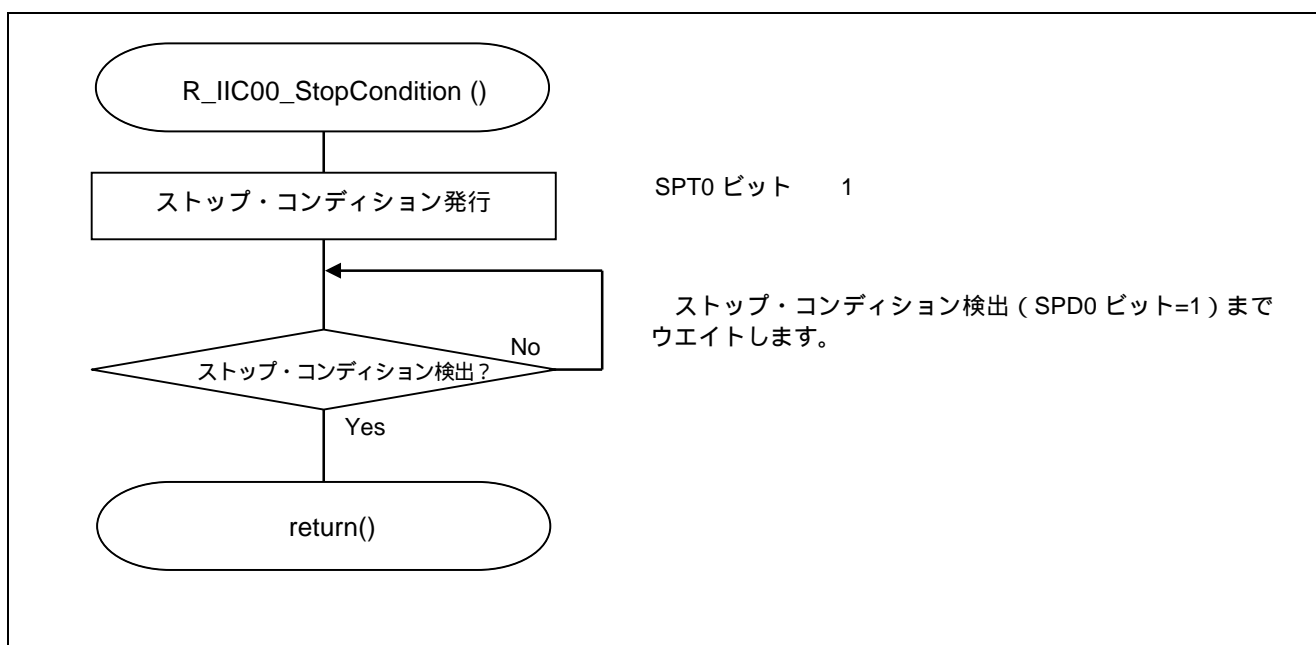


図 5.9 ストップ・コンディション発行関数

5.7.7 EEPROM 書き込み処理

図 5.10 に EEPROM 書き込み処理のフローチャートを示します。

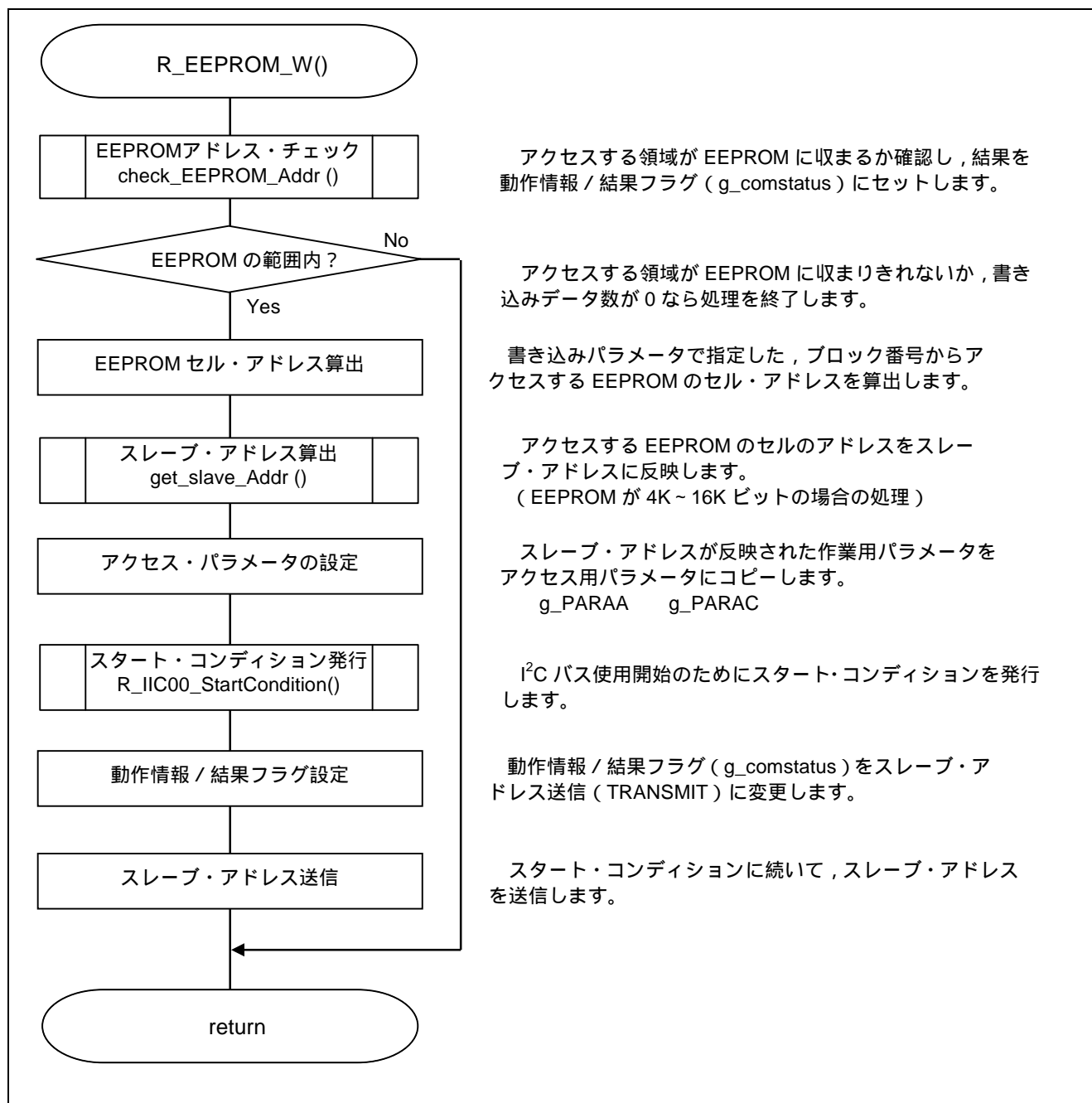


図 5.10 EEPROM 書き込み処理

5.7.8 EEPROM アドレス・チェック処理

図 5.11 に EEPROM アドレス・チェック処理のフローチャートを示します。

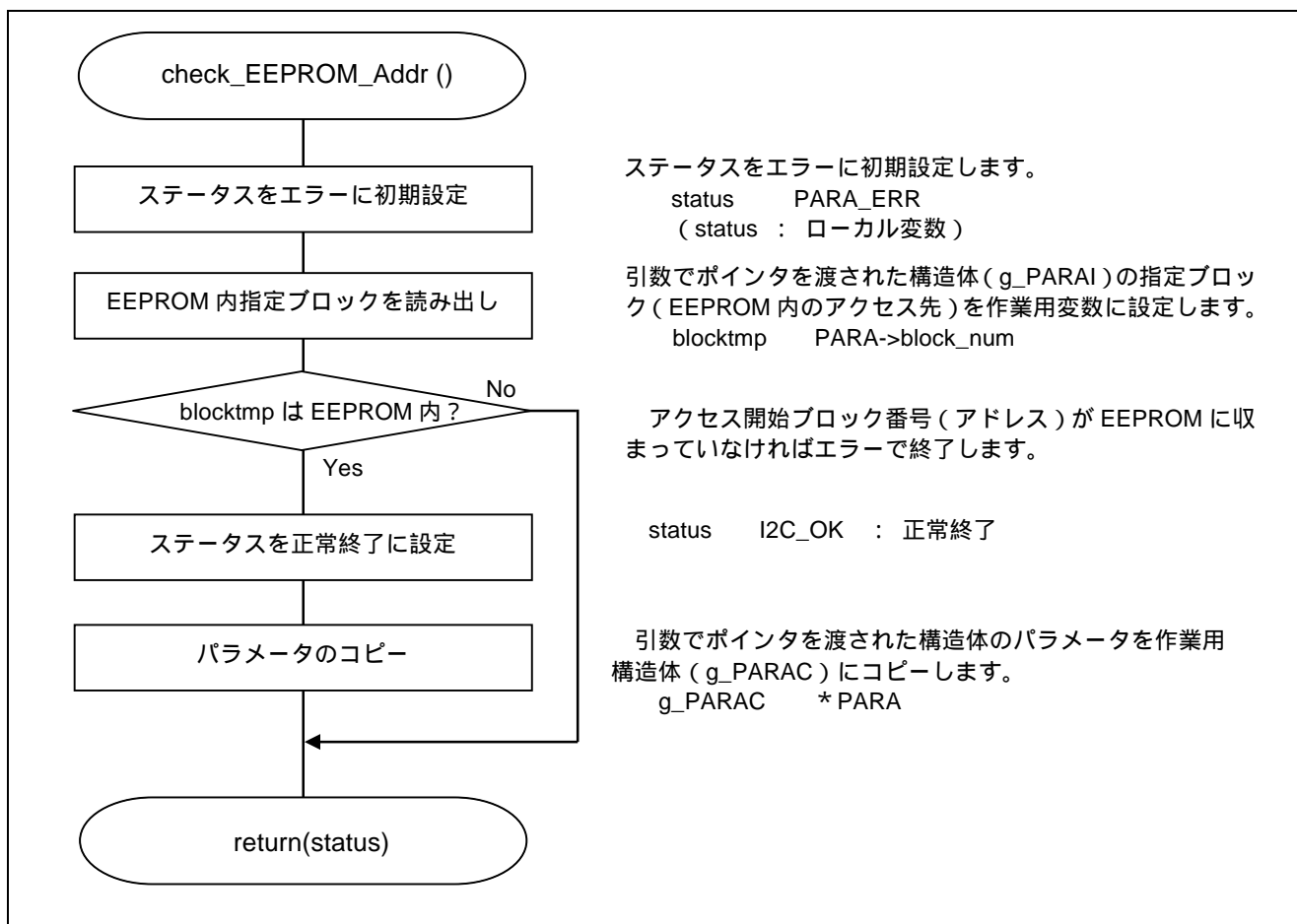


図 5.11 EEPROM アドレス・チェック処理

5.7.9 スレーブ・アドレス算出処理

図 5.12 にスレーブ・アドレス算出処理のフローチャートを示します。

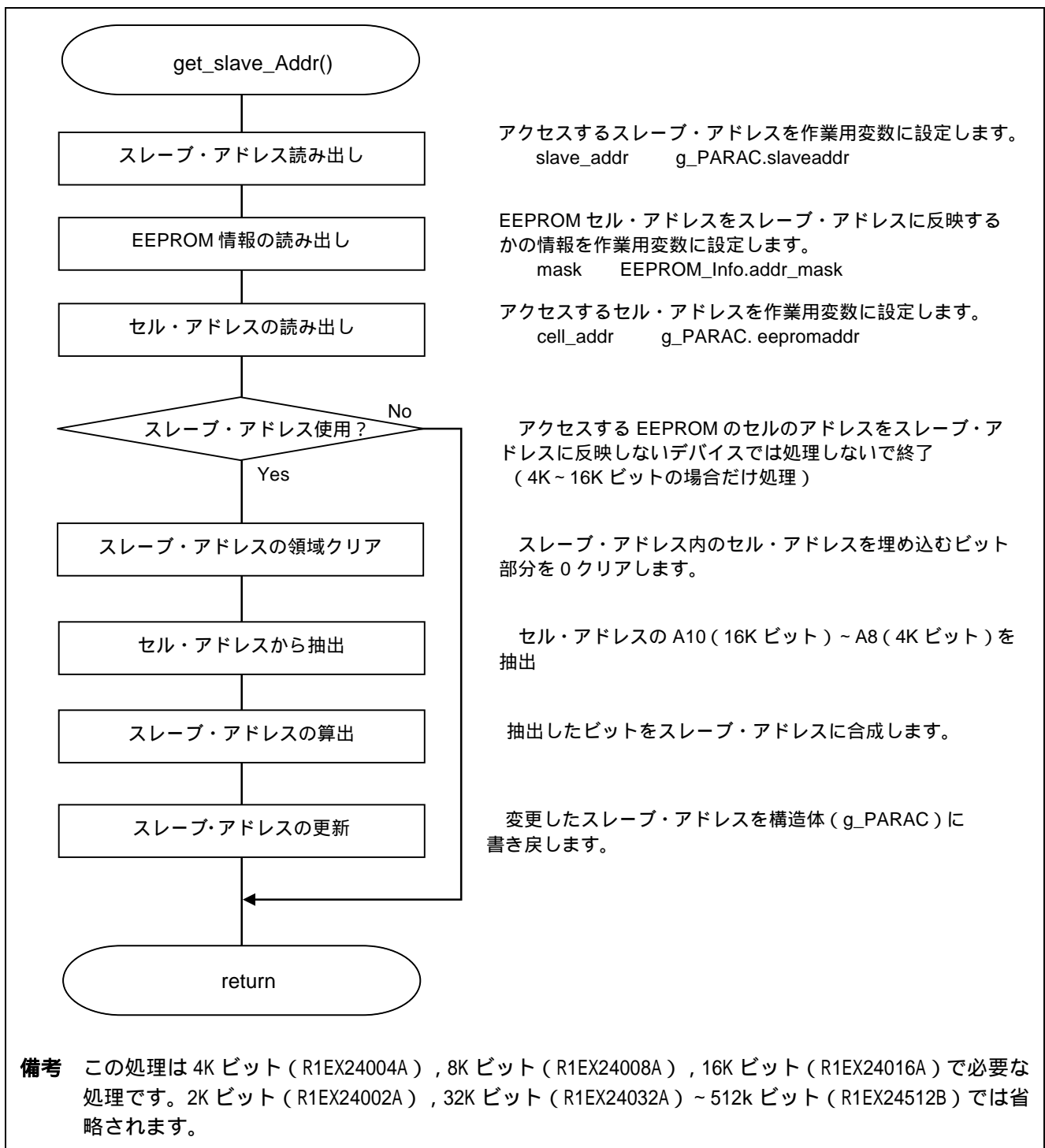


図 5.12 スレーブ・アドレス算出処理

5.7.10 EEPROM 書き込み完了待ち処理

図 5.13 に EEPROM 書き込み完了待ち処理のフローチャートを示します。

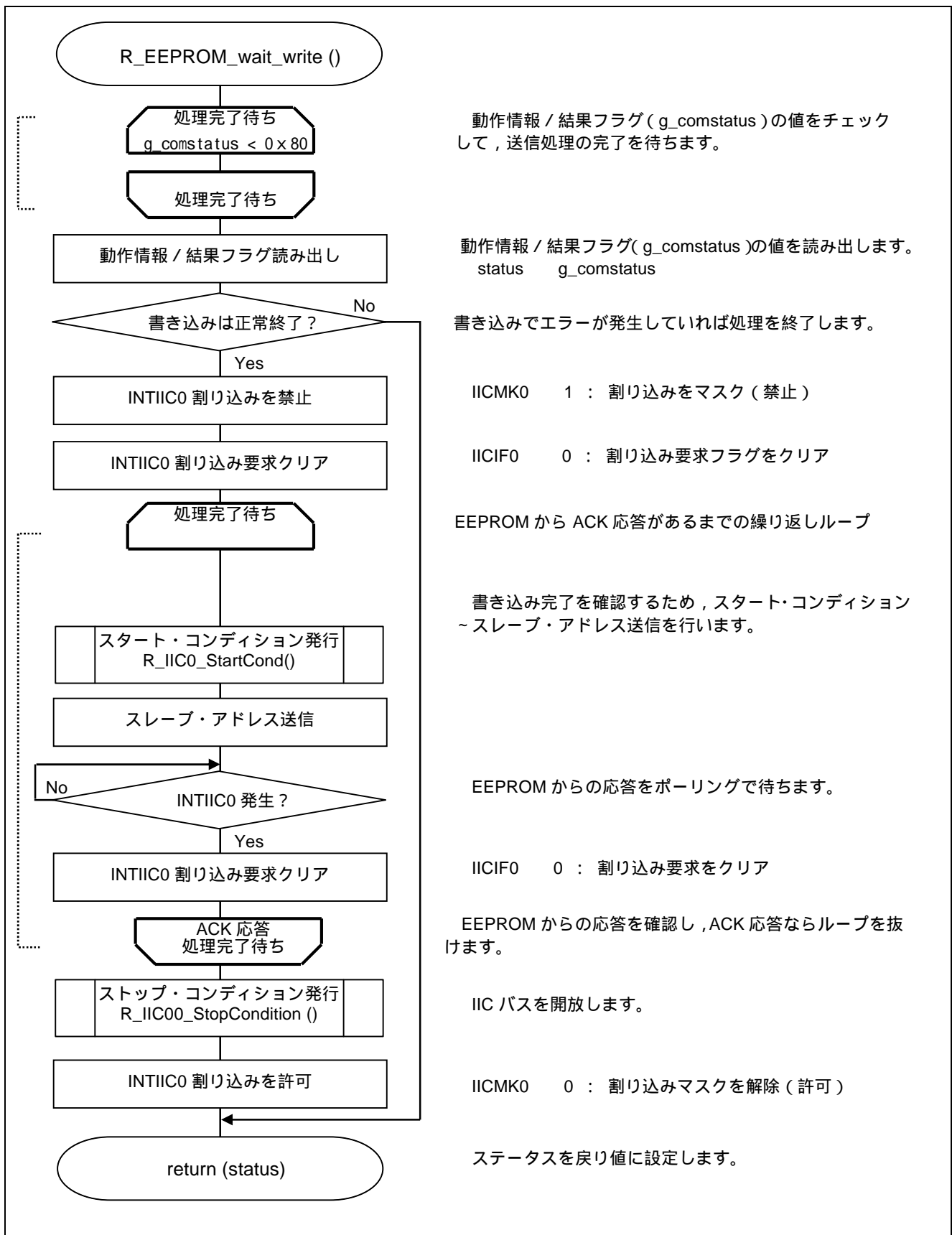


図 5.13 EEPROM 書き込み完了待ち処理

5.7.11 EEPROM 読み出し処理

図 5.14 に EEPROM 読み出し処理のフローチャートを示します。

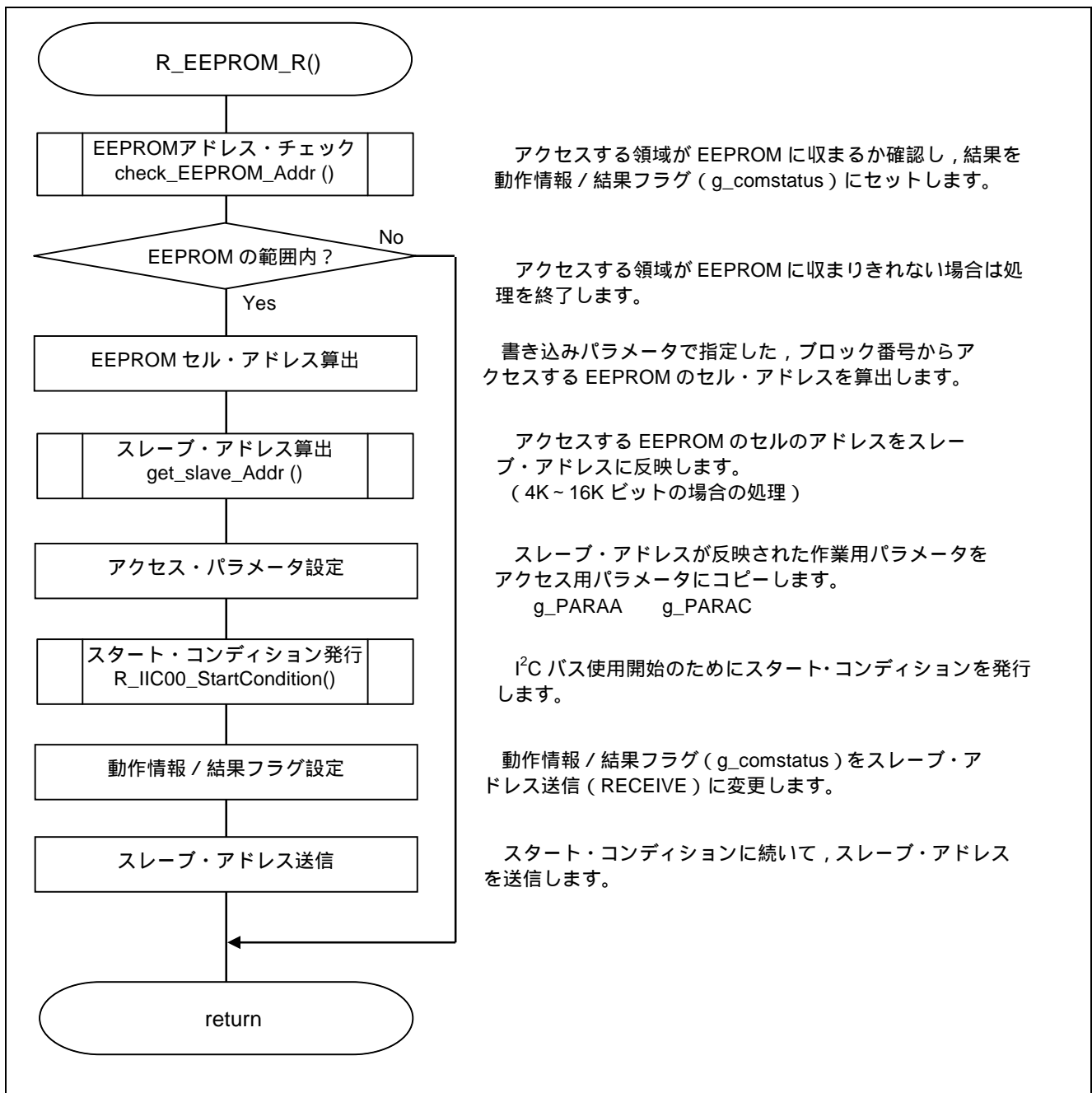


図 5.14 EEPROM 読み出し処理

5.7.12 EEPROM 読み出し完了待ち処理

図 5.15 に EEPROM 読み出し完了待ち処理のフローチャートを示します。

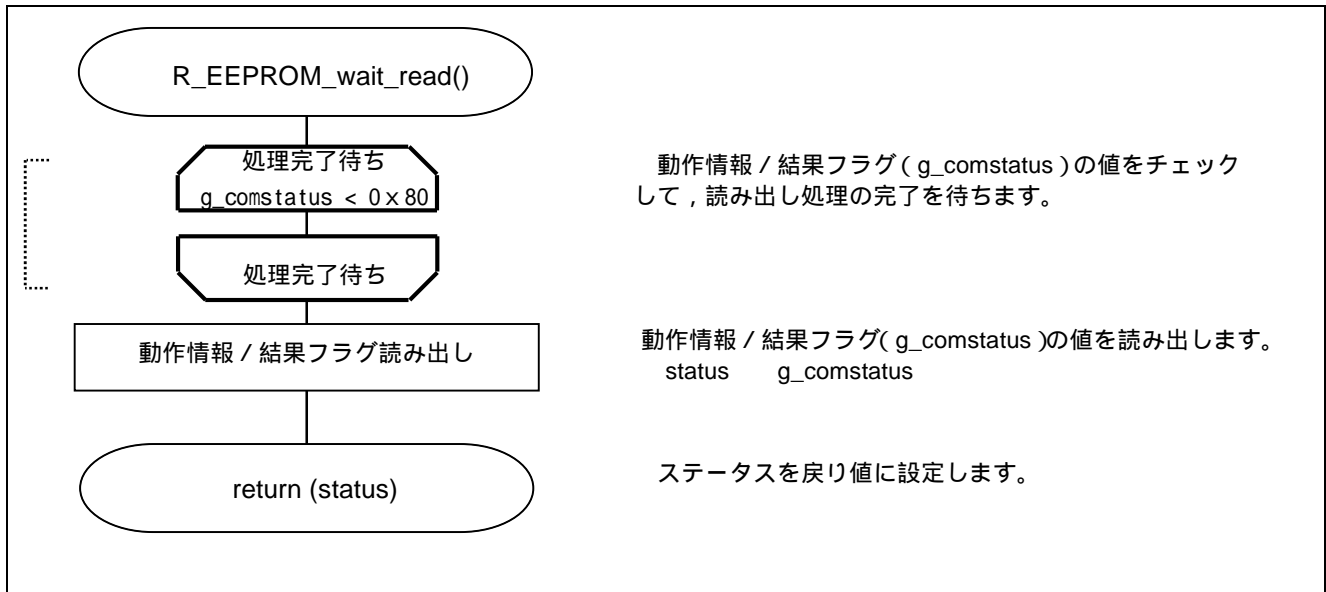


図 5.15 EEPROM 読み出し完了待ち処理

5.7.13 スレーブ・アドレス送信完了処理

図 5.16 にスレーブ・アドレス送信完了処理のフローチャートを示します。

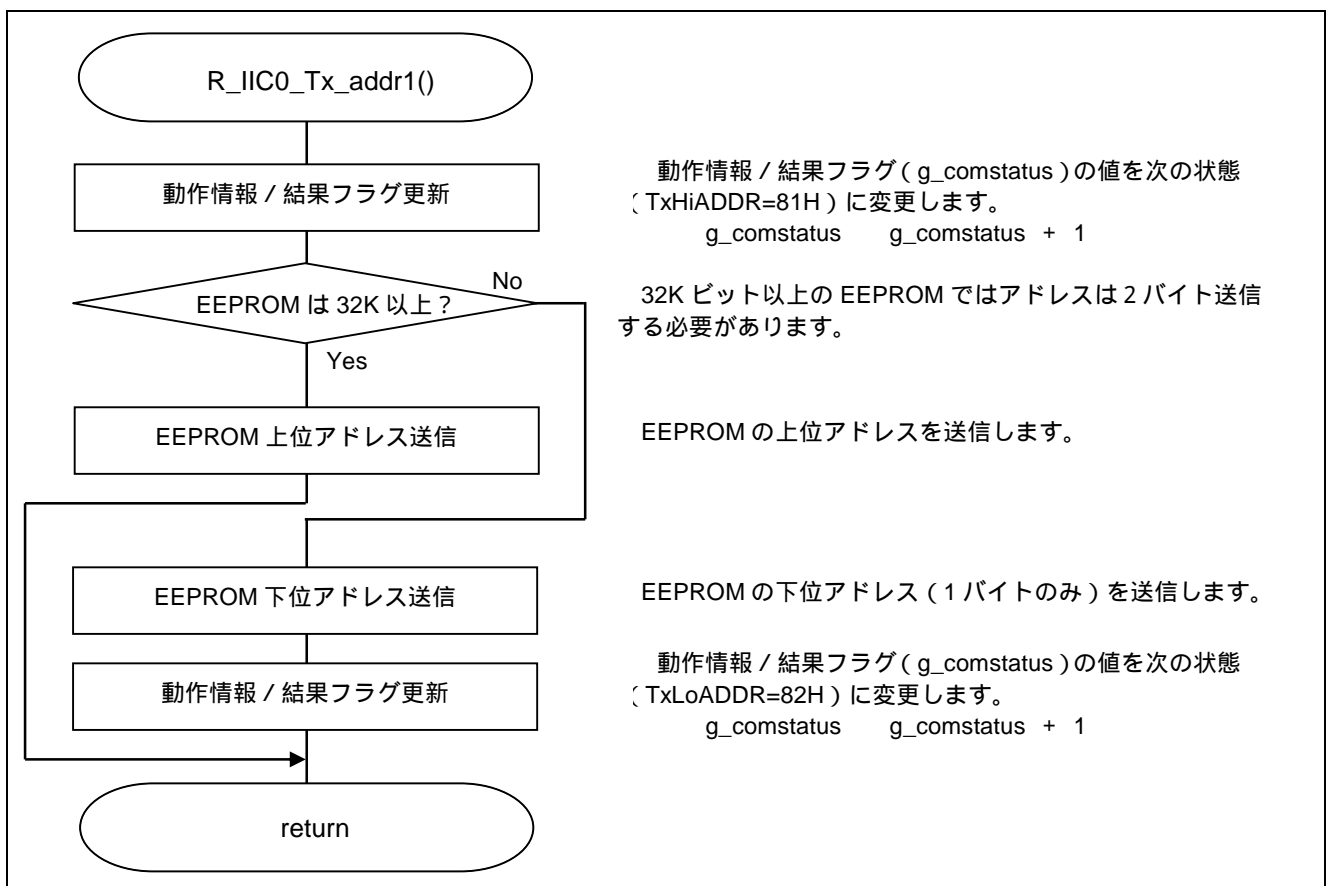


図 5.16 スレーブ・アドレス送信完了処理

5.7.14 上位アドレス送信完了処理

図 5.17 に上位アドレス送信完了処理のフローチャートを示します。

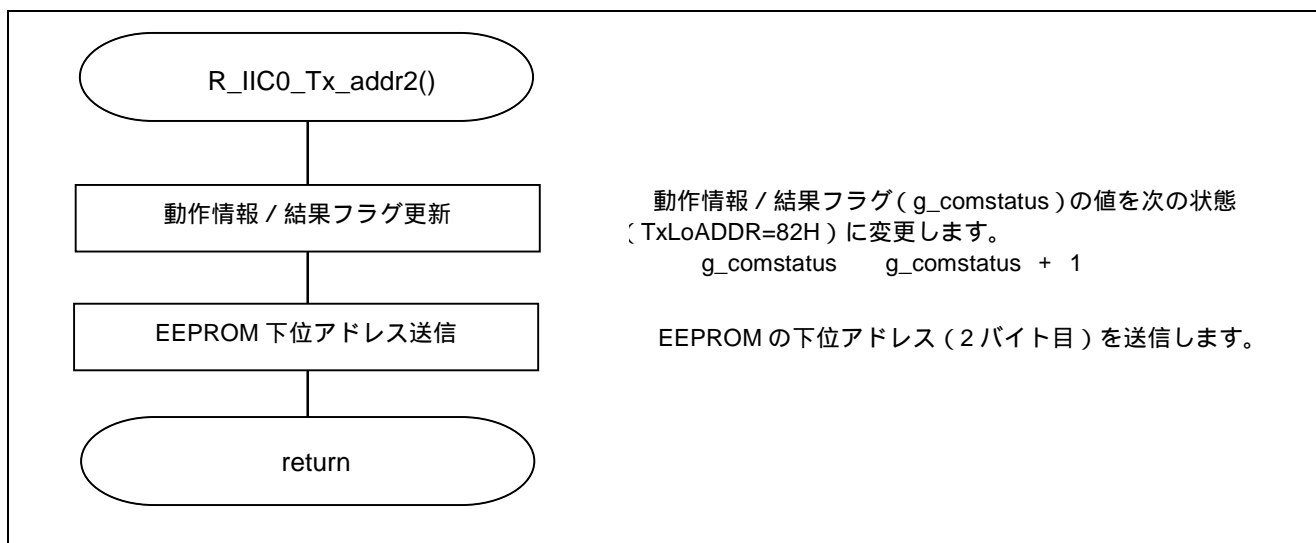


図 5.17 上位アドレス送信完了処理

5.7.15 リスタート処理

図 5.18 にリスタート処理のフローチャートを示します。

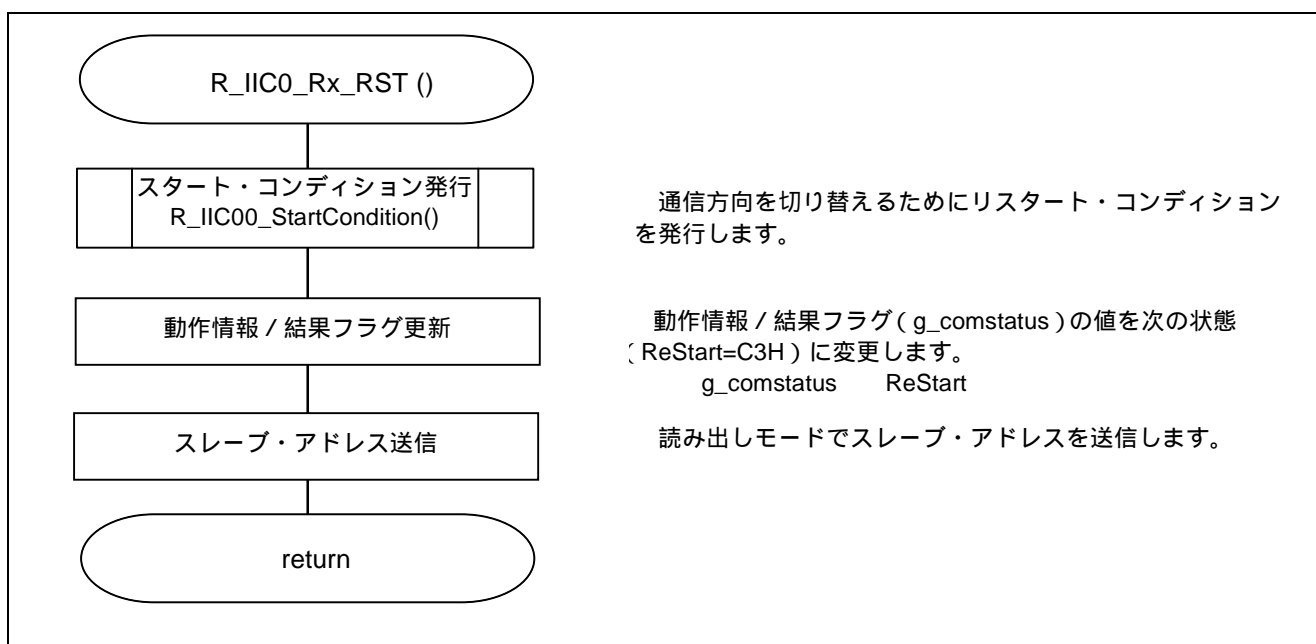


図 5.18 リスタート処理

5.7.16 データ受信開始処理

図 5.19 にデータ受信開始処理のフローチャートを示します。

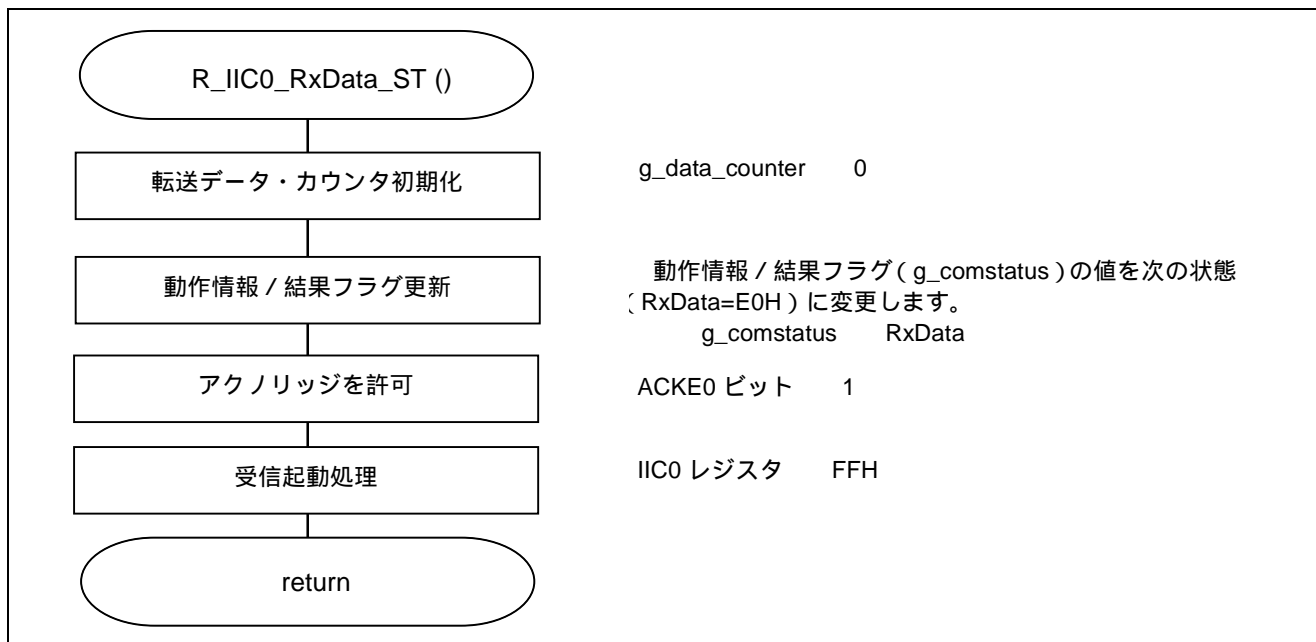


図 5.19 データ受信開始処理

5.7.17 データ受信処理

図 5.20 にデータ受信処理のフローチャートを示します。

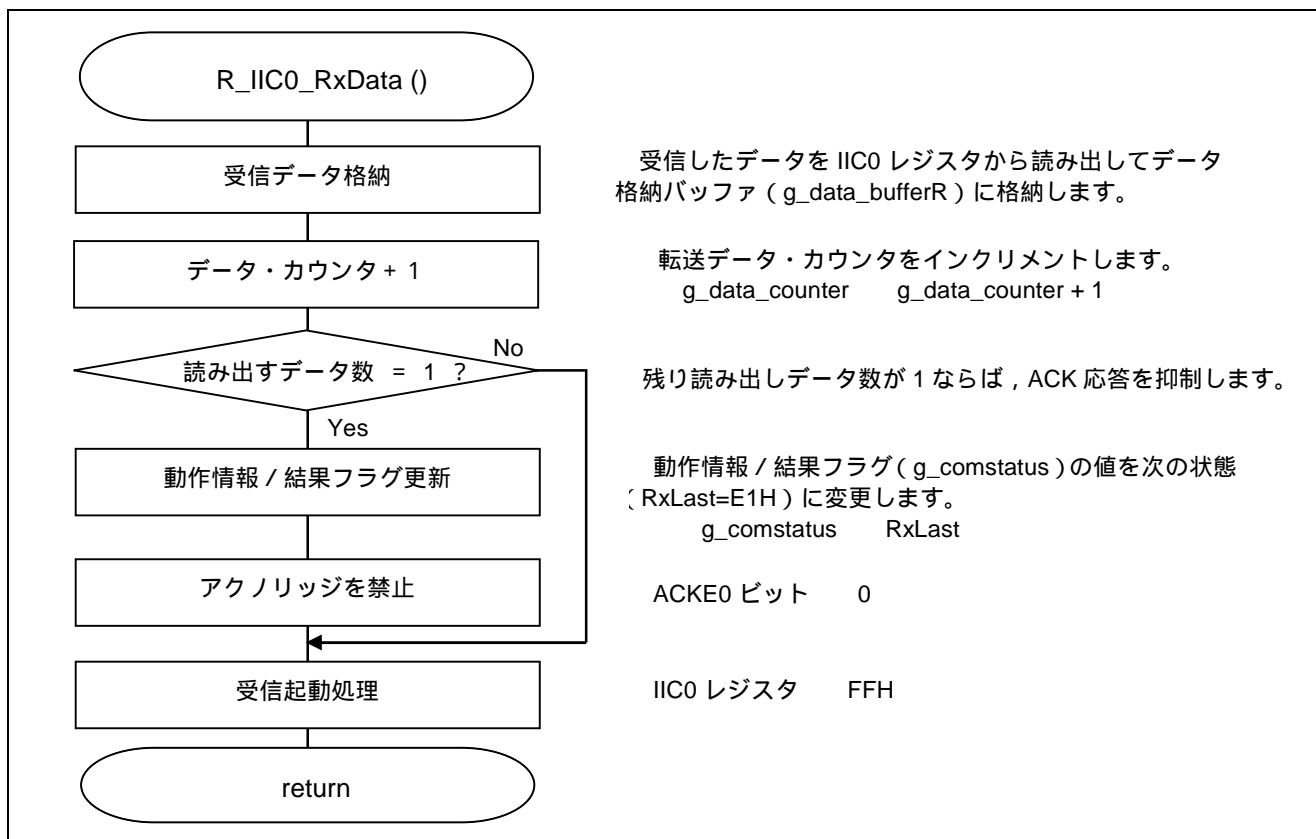


図 5.20 データ受信処理

5.7.18 最終データ受信処理

図 5.21 に最終データ受信処理のフローチャートを示します。

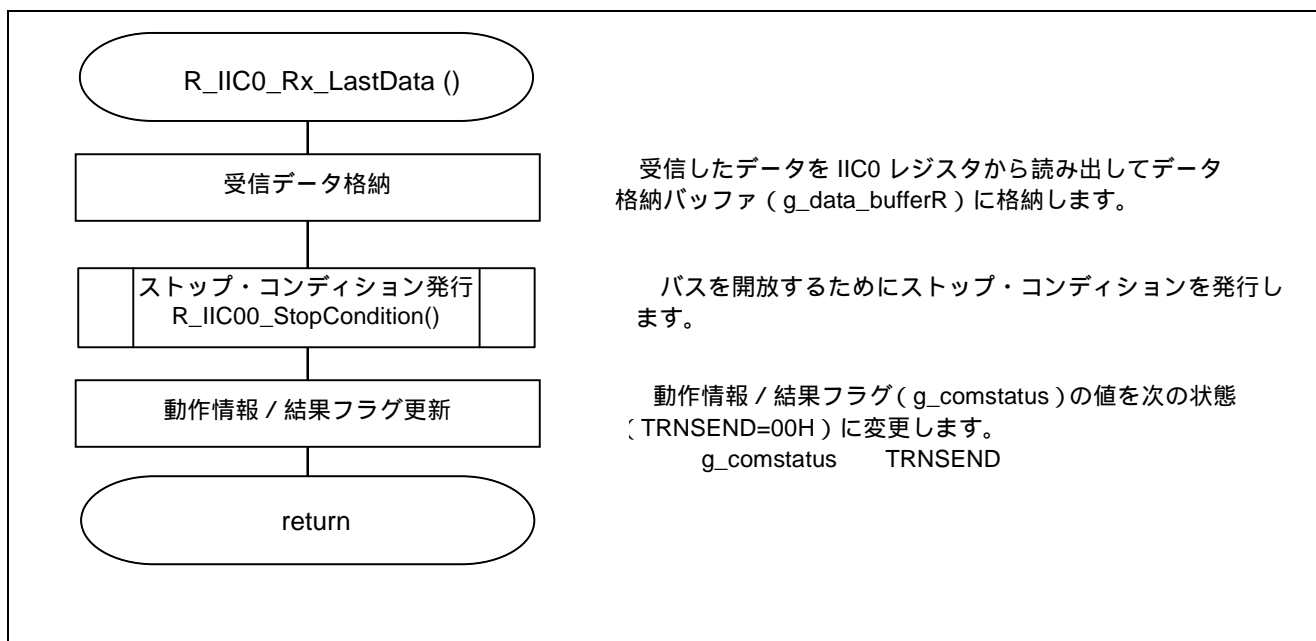


図 5.21 最終データ受信処理

5.7.19 データ送信開始処理

図 5.22 にデータ送信開始処理のフローチャートを示します。

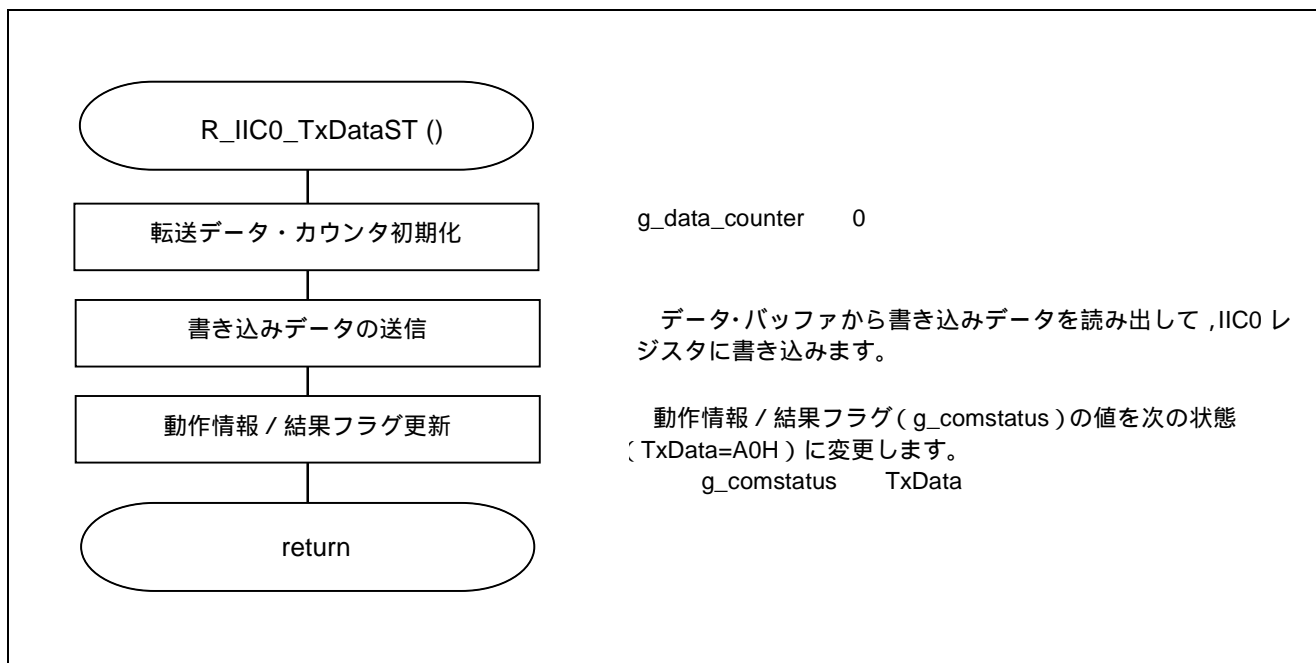


図 5.22 データ送信開始処理

5.7.20 データ送信処理

図 5.23 にデータ送信処理のフローチャートを示します。

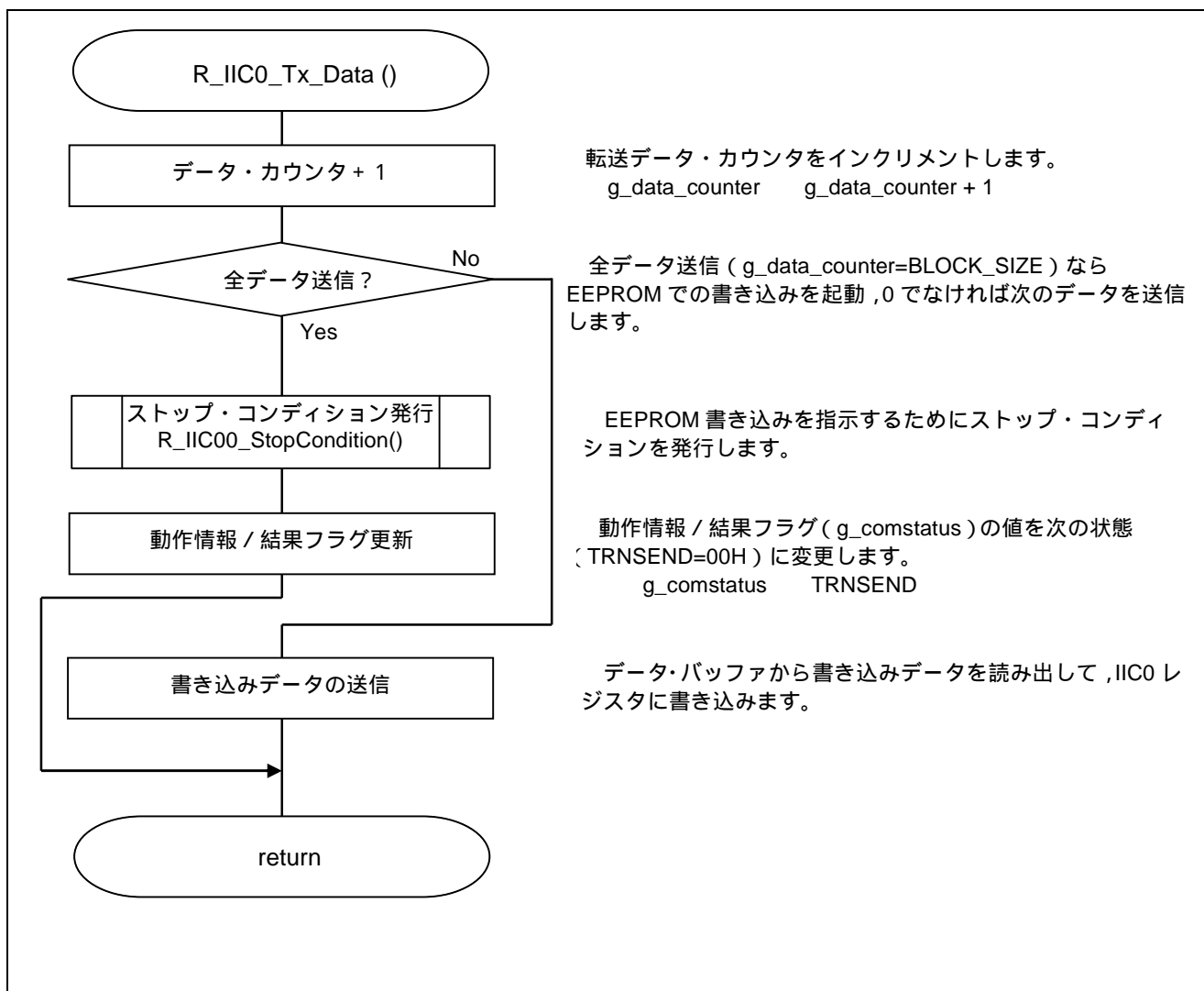


図 5.23 データ送信処理

5.7.21 MD_INTIIC0 割り込み処理

図 5.24 ~ 図 5.41 に INTIIC0 割り込み処理のフローチャートを示します。

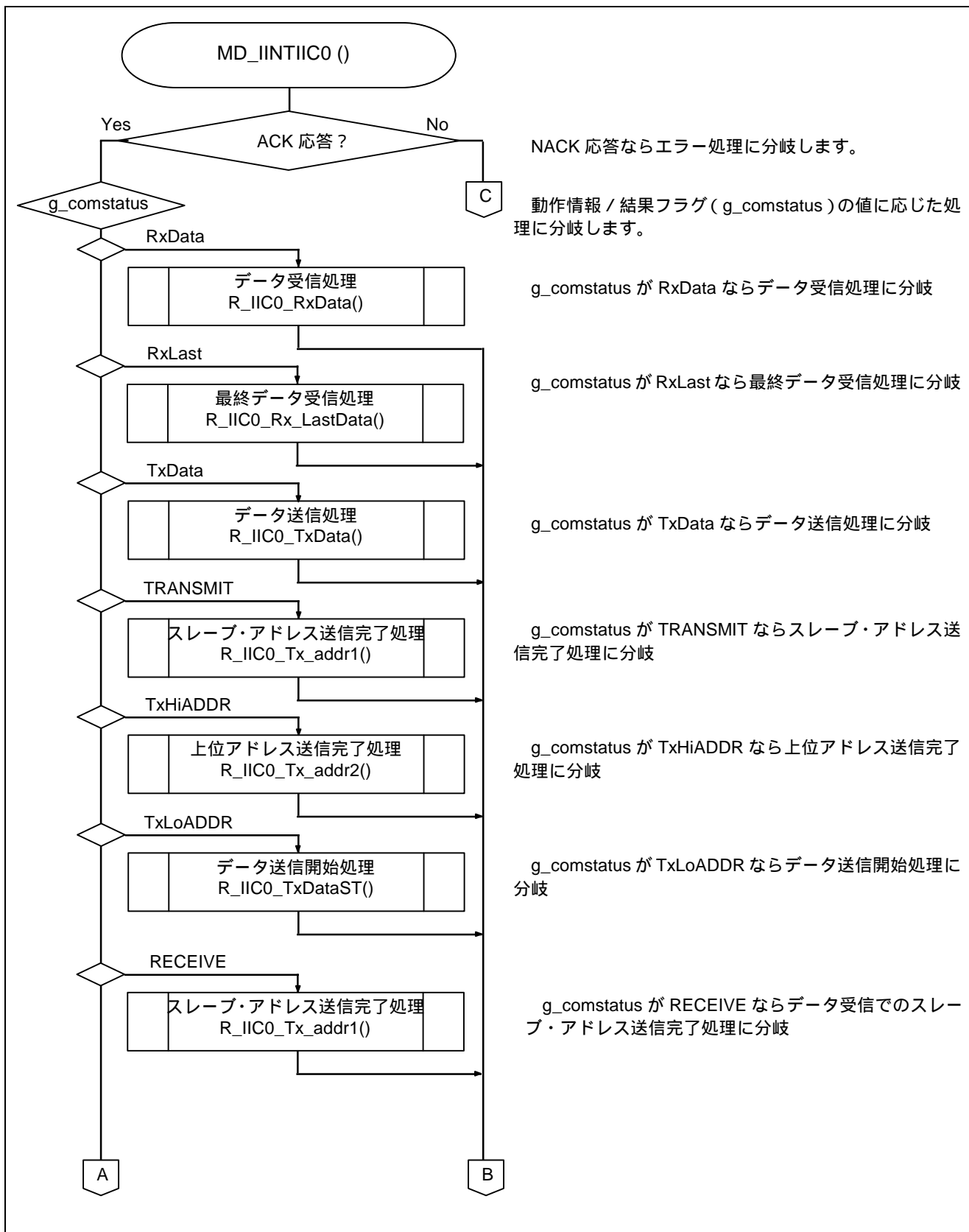


図 5.24 MD_INTIIC0 割り込み処理 (1/3)

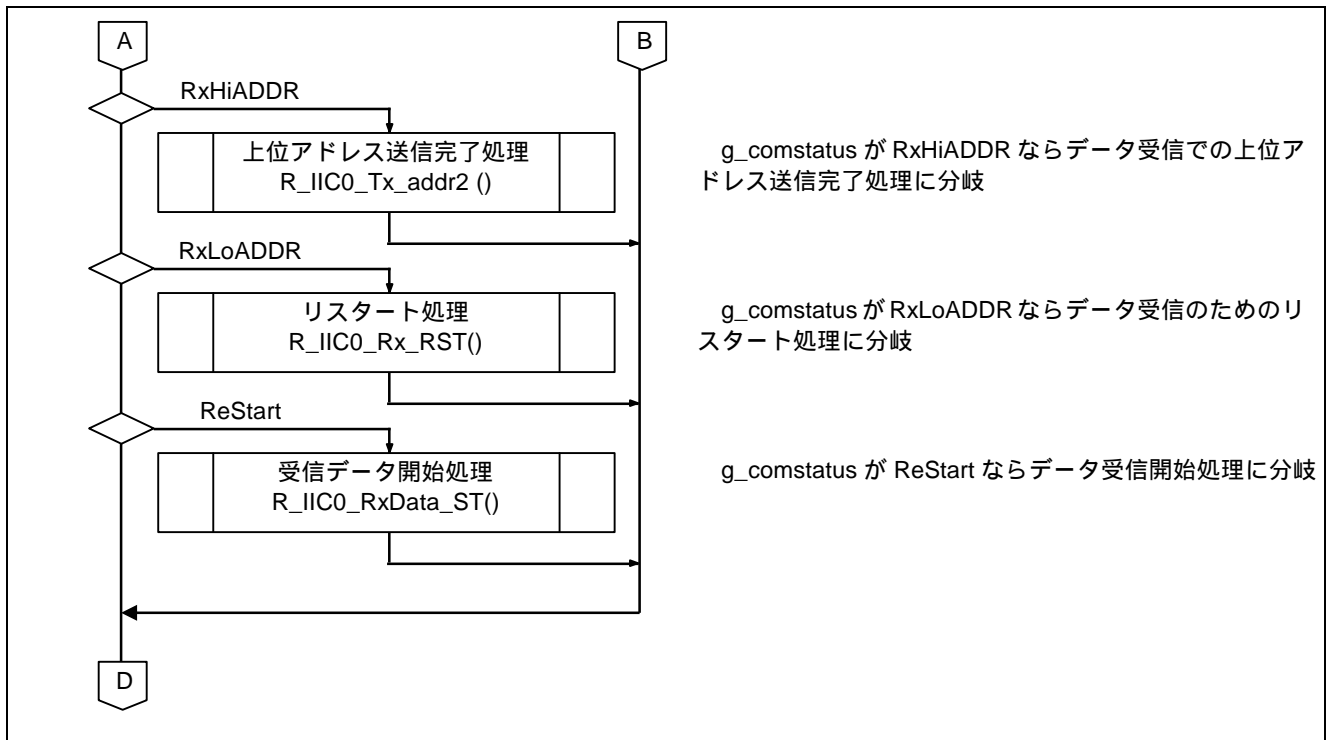


図 5.25 MD_INTIIC0 割り込み処理 (2/3)

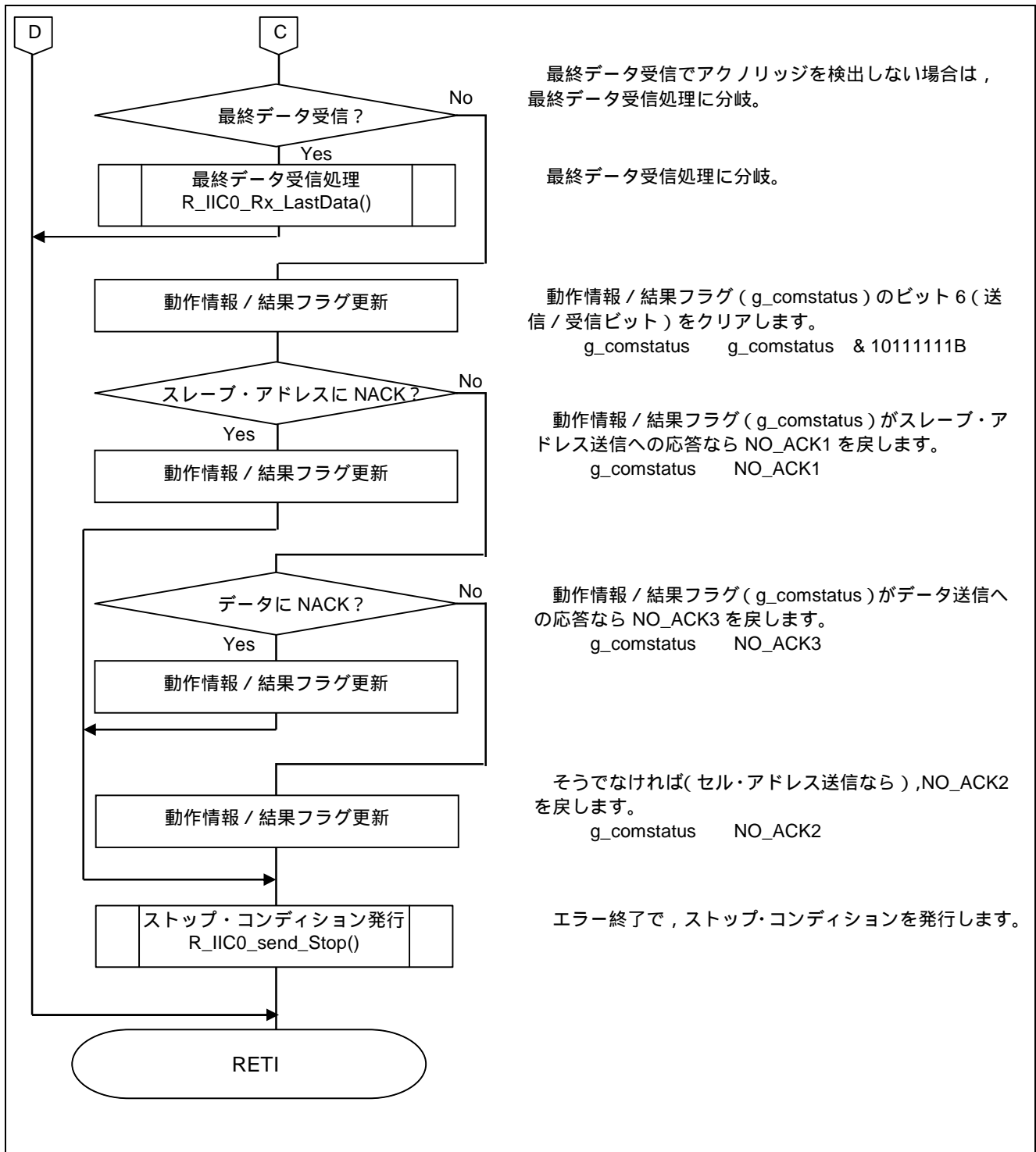


図 5.26 MD_INTIIC0 割り込み処理 (3/3)

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

V850ES/JC3-L, JE3-L ユーザーズマニュアル ハードウェア編 (R01UH0018JJ)

V850ES/JF3-L ユーザーズマニュアル ハードウェア編 (R01UH0017JJ)

V850ES/JG3-L ユーザーズマニュアル ハードウェア編 (R01UH0165JJ)

V850ES/JG3-L (USB コントローラ内蔵製品) ユーザーズマニュアル ハードウェア編 (R01UH0001JJ)
(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	V850ES/Jx3-L I ² バス EEPROM 制御
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.3.28	—	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

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、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事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・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/contact/>