

RL78/G23

R01AN7733JJ0100

Rev. 1.00

2025.04.11

シリアル・アレイ・ユニット (SAU)

(簡易 IIC による EEPROM 制御)

要旨

本アプリケーションノートでは、シリアル・アレイ・ユニット (SAU) の簡易 IIC 機能を用いた EEPROM の RL78/G23 制御方法を説明します。簡易 IIC 機能を使用して IIC バス接続による EEPROM の読みだしと書き込みを割り込み処理により実現しています。

対象デバイス

RL78/G23

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	11
3. 関連アプリケーションノート	11
4. ハードウェア説明	12
4.1 ハードウェア構成例	12
4.2 使用端子一覧	12
5. ソフトウェア説明	13
5.1 動作概要	13
5.2 オプション・バイトの設定一覧	19
5.3 定数一覧	19
5.4 変数一覧	21
5.5 関数一覧	23
5.6 関数仕様	25
5.7 フローチャート	34
5.7.1 メイン関数	34
5.7.2 EEPROMの選択	37
5.7.3 バス開放関数	38
5.7.4 ストップ・コンディション発行関数	39
5.7.5 バス開放処理関数	40
5.7.6 EEPROM書き込み処理	41
5.7.7 EEPROMアドレス・チェック処理	43
5.7.8 スレーブ・アドレス算出処理	44
5.7.9 書き込みデータのページ分割処理	45
5.7.10 スタート・コンディション発行処理	46
5.7.11 EEPROM書き込み完了待ち処理	47
5.7.12 EEPROM読み出し処理	48
5.7.13 EEPROM読み出し完了待ち処理	50
5.7.14 スレーブ・アドレス送信完了処理	50
5.7.15 上位アドレス送信完了処理	51
5.7.16 リスタート処理	51
5.7.17 データ受信開始処理	52
5.7.18 データ受信処理	53
5.7.19 最終データ受信処理	54
5.7.20 データ送信開始処理	54
5.7.21 データ送信処理	55
5.7.22 次ページ書き込み開始処理	56
5.7.23 SCLダミー・クロック出力処理	57
5.7.24 SCL立ち上げ処理	57
5.7.25 SCL立ち下げ処理	58
5.7.26 ACK確認処理	58
5.7.27 SCLロウ・レベル時間待ち処理	59
5.7.28 SCLハイ・レベル時間待ち処理	59
5.7.29 INTIICr割り込み処理	60
5.7.30 INTTM02割り込み処理	63
6. サンプルコード	64
7. 参考ドキュメント	64
改訂記録	65

1. 仕様

本アプリケーションノートは、シリアル・アレイ・ユニットを簡易 IIC 機能で使用して IIC バス接続による EEPROM の読み出しと書き込みを行います。

- ・ EEPROM へのアクセスは構造体にパラメータを設定して、関数を呼び出すことで処理します。
- ・ API としての動作を意識して、EEPROM の制御はできる限り割り込みで処理します。
- ・ 対象とする EEPROM は 2K~512K ビットから 1 種類を選択できるものとします。(デフォルトでは 16K ビットの R1EX24016A を選択しておきます。メイン関数もこれに合わせたテスト処理を行います。)
- ・ 選択された EEPROM に対応したアクセス制御を行います。
- ・ EEPROM の読み出し中にリセットがかかり、EEPROM が SDA 信号にロウを出力した状態で再開したときに、バスが占有された状態からのバスの開放対応も行います。
- ・ 使用する簡易 IIC のチャンネルはチャンネル 03 を標準としますが、簡単に指定変更することができるように考慮します。

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

図 1.2 から図 1.8 に簡易 IIC 機能を用いた IIC 通信のタイミング・チャートを示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
シリアル・アレイ・ユニット	簡易 IIC 機能を利用して、IIC マスタ送受信を行う。 (SCL11 端子と SDA11 端子を使用)

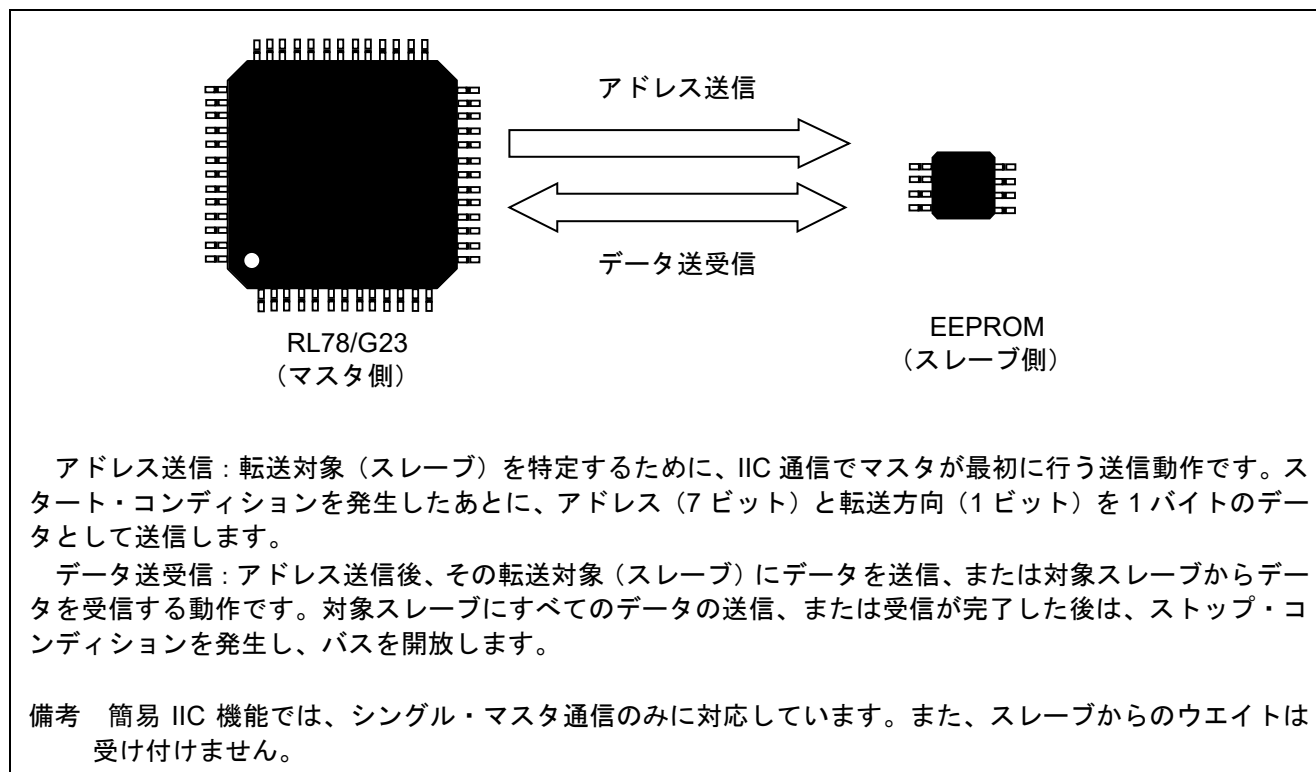


図 1.1 IIC 通信の概要

(1) マスタ→スレーブ通信 1 (スタート・コンディション～アドレス～データ)

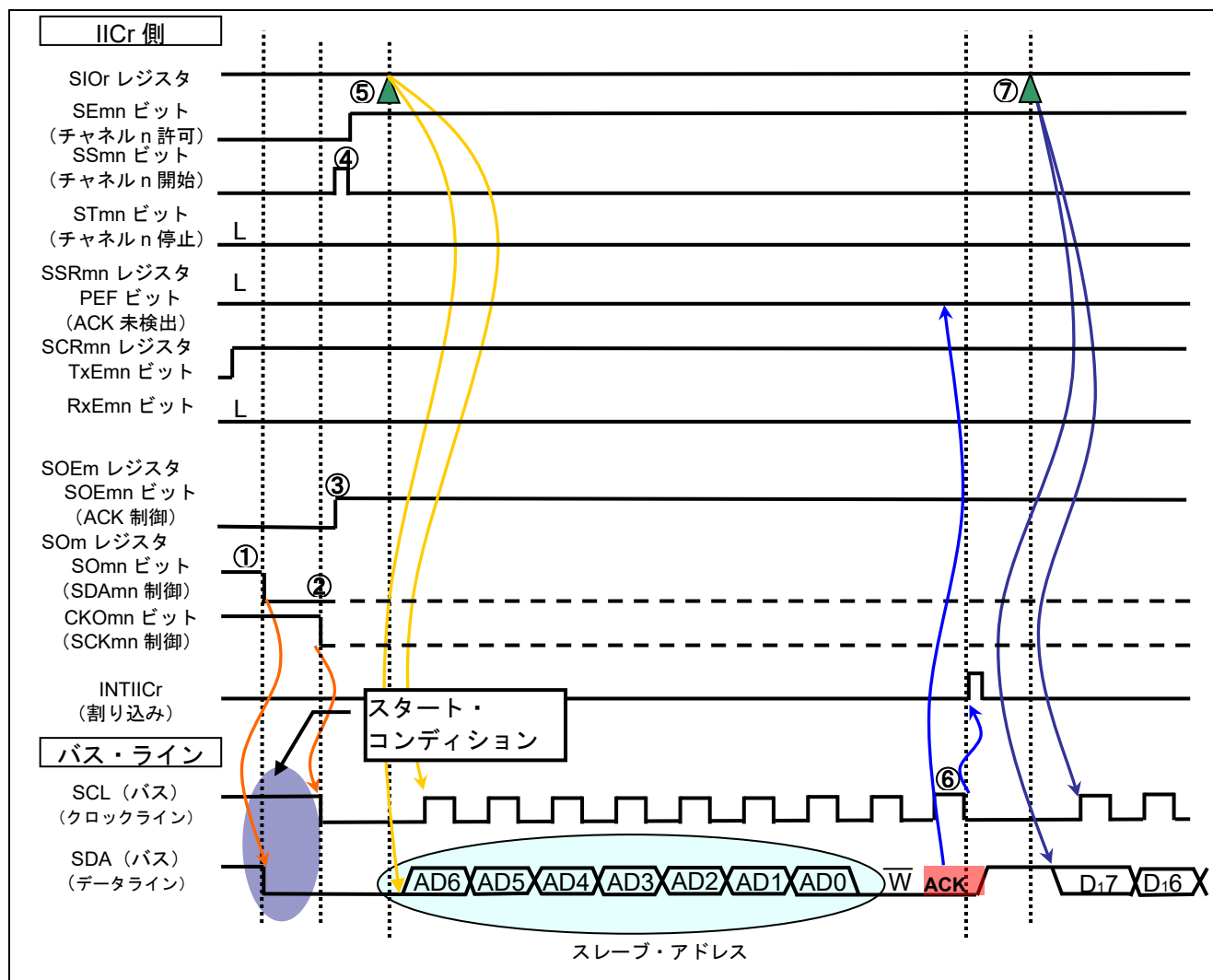


図 1.2 IIC 通信タイミング・チャート (マスター→スレーブ通信例) (1/4)

- ① IIC_{mn} の初期設定 (ここで、SCR_{mn} レジスタは TxEmn=1、RxEmn=0 に設定しておきます) が完了したら、SO_{mn} レジスタの SO_{mn} ビット=0 に設定して SDA 信号を立ち下げることによってスタート・コンディションを発行します。
- ② スタート・コンディションのホールド時間 (標準モードで 4.0μs、ファースト・モードで 0.6μs) 経過したら、SO_{mn} レジスタの CKO_{mn} ビット=0 に設定して SCL 信号を立ち下げます。
- ③ 通信のために SOE_{mn} レジスタの SOE_{mn} ビット=1 に設定して出力を許可します。
- ④ SS_{mn} レジスタの SS_{mn} ビット=1 にしてチャンネル n を動作許可状態にします。
- ⑤ SIO_r レジスタにスレーブのアドレスを書き込むと、通信が起動します。
- ⑥ スレーブ・アドレスの送信が完了すると INTIICr が発生します。
- ⑦ スレーブからの ACK 応答を SSR_{mn} レジスタの PEF ビットで確認し、PEF ビット=0 ならば、SIO_r レジスタに送信データを書き込みます。PEF ビット=1 ならば、送信を中止します。

(2) マスタ→スレーブ通信 2 (アドレス～データ～データ)

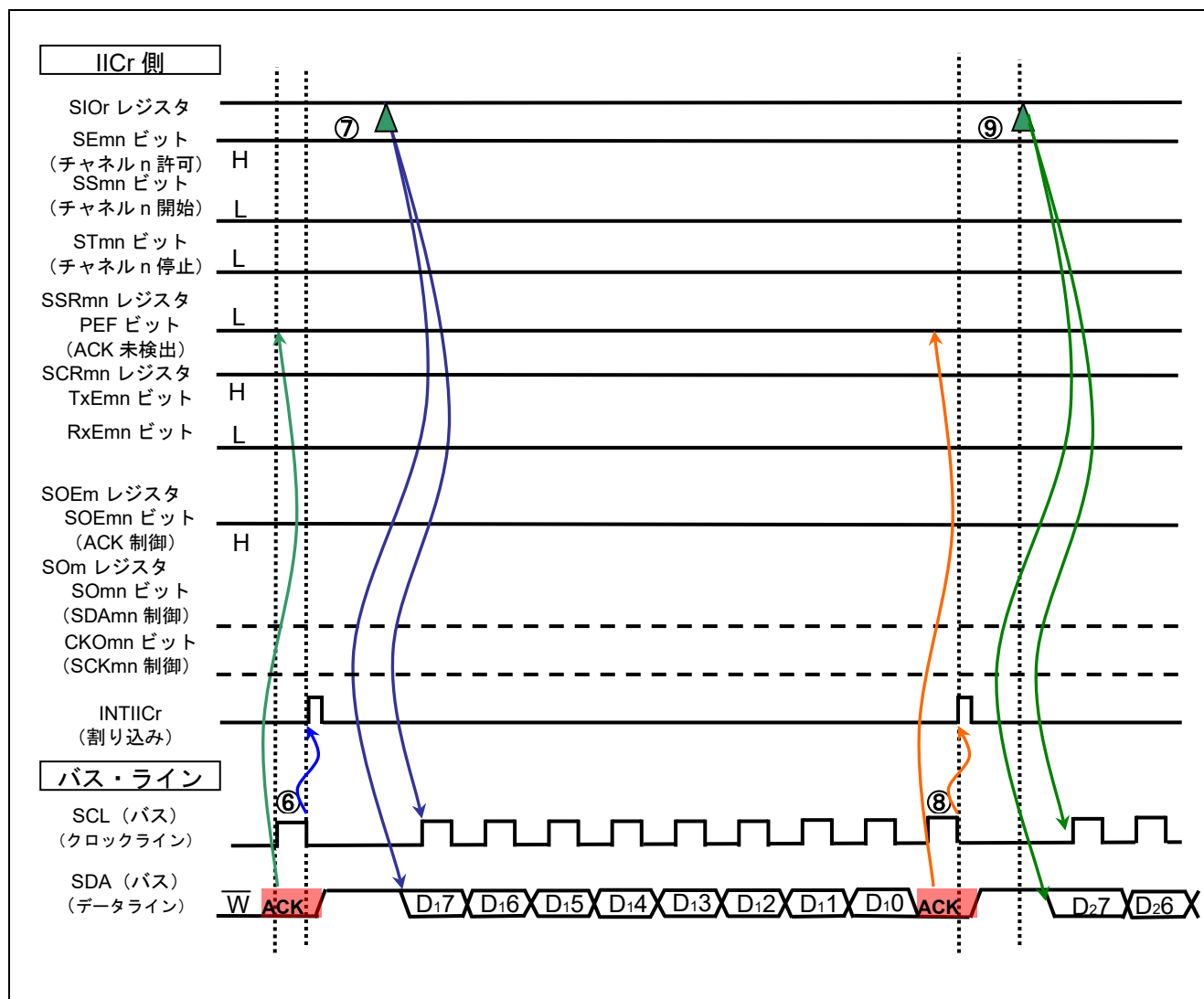


図 1.3 IIC 通信タイミング・チャート (マスタ→スレーブ通信例) (2/4)

- ⑥ スレーブ・アドレスの送信が完了すると INTIICr が発生します。
- ⑦ スレーブからの ACK 応答を SSRmn レジスタの PEF ビットで確認し、PEF ビット=0 ならば、SIOr レジスタに送信データを書き込みます。PEF ビット=1 ならば、送信を中止します。
- ⑧ データ転送が完了すると INTIICr が発生します。
- ⑨ スレーブからの ACK 応答を SSRmn レジスタの PEF ビットで確認し、PEF ビット=0 ならば、SIOr レジスタに送信データを書き込みます。PEF ビット=1 ならば、送信を中止します。

(3) マスタ→スレーブ通信 3 (データ～データ～ストップ・コンディション)

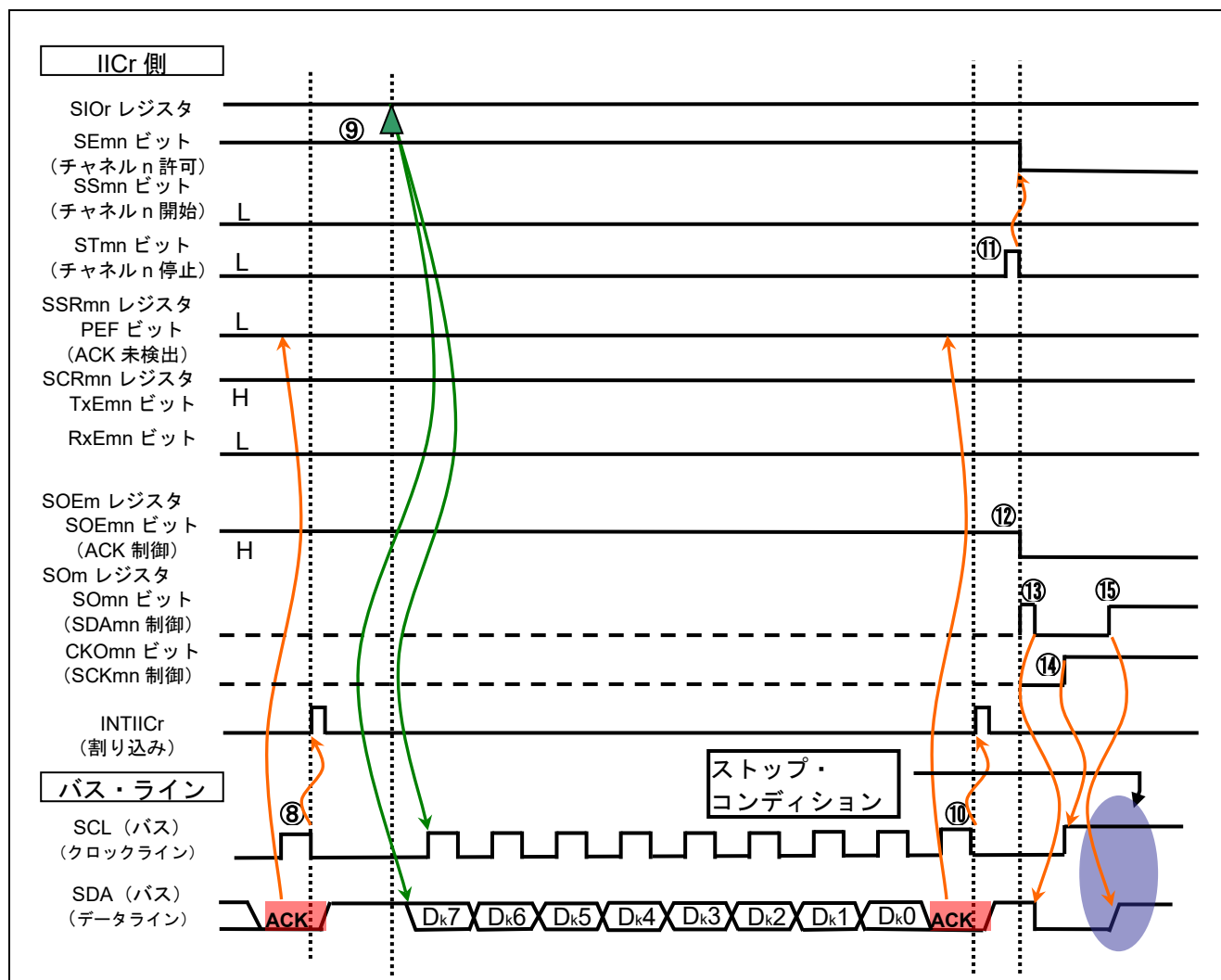


図 1.4 IIC 通信タイミング・チャート (マスター→スレーブ通信例) (3/4)

- ⑧ データ転送が完了すると INTIICr が発生します。
- ⑨ スレーブからの ACK 応答を SSR_{mn} レジスタの PEF ビットで確認し、PEF ビット=0 ならば、SIO_r レジスタに送信データを書き込みます。PEF ビット=1 ならば、送信を中止します。
- ⑩ データ転送が完了すると INTIICr が発生します。
- ⑪ ST_m レジスタの ST_{mn} ビット=1 にしてチャンネル n を動作禁止状態にします。
- ⑫ ストップ・コンディション発行のために SOE_m レジスタの SOE_{mn} ビット=0 に設定して出力を停止します。
- ⑬ ストップ・コンディション発行準備で SDA を立ち下げるために SO_m レジスタの SO_{mn} ビット=0 に設定します。
- ⑭ ストップ・コンディション発行準備で SCL を立ち上げるために SO_m レジスタの CKO_{mn} ビット=1 に設定します。
- ⑮ ストップ・コンディションのセットアップ時間が経過したら、SO_m レジスタの SO_{mn} ビット=1 に設定することで、ストップ・コンディションが発行されます。

(4) マスタ→スレーブ通信 4 (データ～リスタート・コンディション～アドレス)

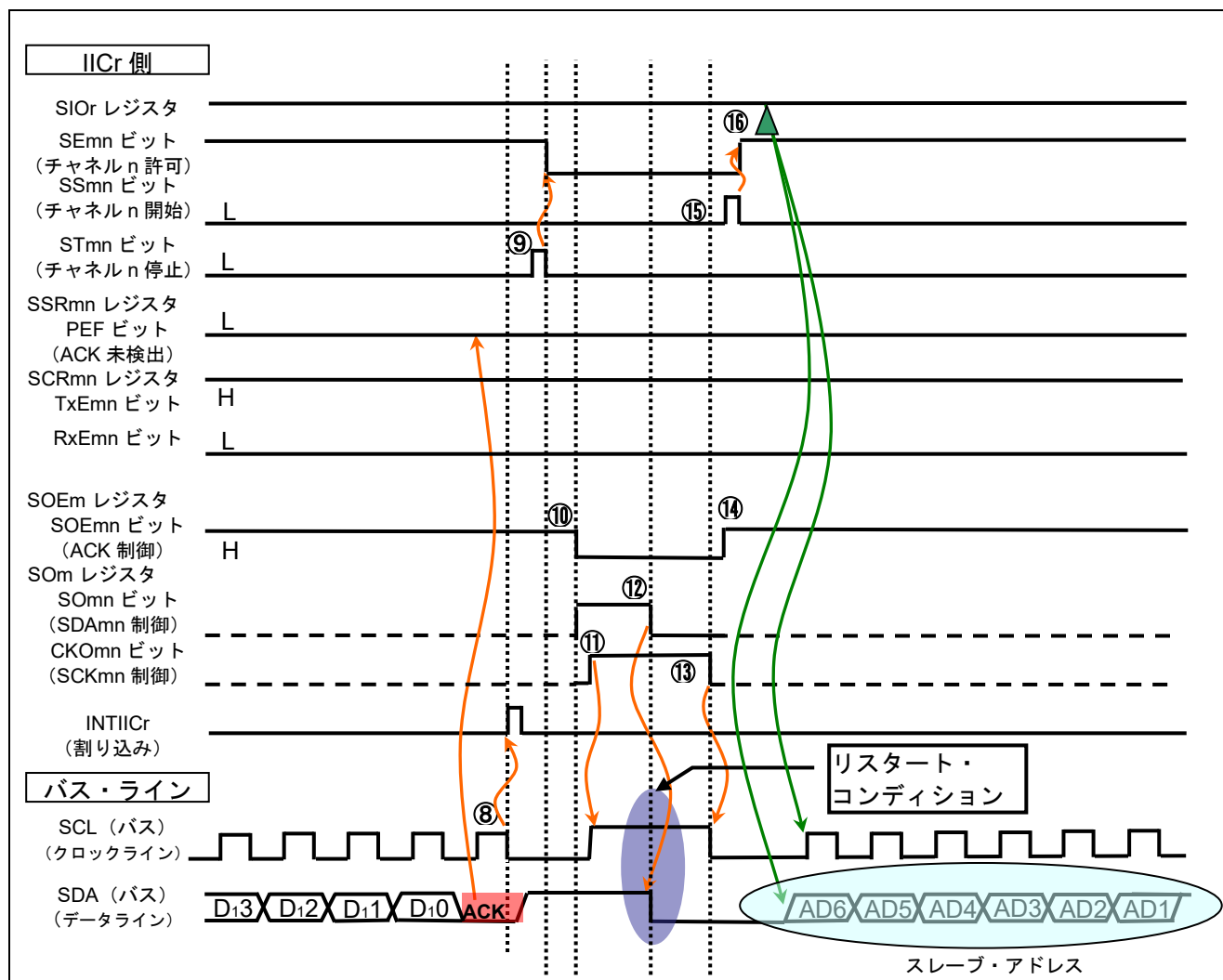


図 1.5 IIC 通信タイミング・チャート (マスタ→スレーブ通信例) (4/4)

- ⑧ データ転送が完了すると INTIICr が発生し、スレーブからの ACK 応答を SSRmn レジスタの PEF ビットで確認します。
- ⑨ STm レジスタの STmn ビット=1 にしてチャネル n を動作禁止状態にします。
- ⑩ リスタート・コンディション発行のために SOEm レジスタの SOEmn ビット=0 に設定して出力を停止します。
- ⑪ リスタート・コンディション発行準備で SDA に続いて SCL を立ち上げるために SOm レジスタの CKOm ビット=1 に設定します。
- ⑫ SOm レジスタの SOMn ビット=0 に設定することで、リスタート・コンディションが発行されます。
- ⑬ スタート・コンディションのホールド時間が経過したら、SOm レジスタの CKOm ビット=0 に設定して SCL 信号を立ち下げます。
- ⑭ 通信のために SOEm レジスタの SOEmn ビット=1 に設定して出力を許可します。
- ⑮ SSm レジスタの SSmn ビット=1 にしてチャネル n を動作許可状態にします。
- ⑯ SIOr レジスタにスレーブのアドレスを書き込むと、通信が起動します。

備考 この処理は EEPROM の読みだし時にマスタ→スレーブ通信で EEPROM のセルのアドレスを指定し、指定したアドレスから読み出すときに使用します。

(5) スレーブ→マスタ通信 1 (スタート・コンディション～アドレス～データ)

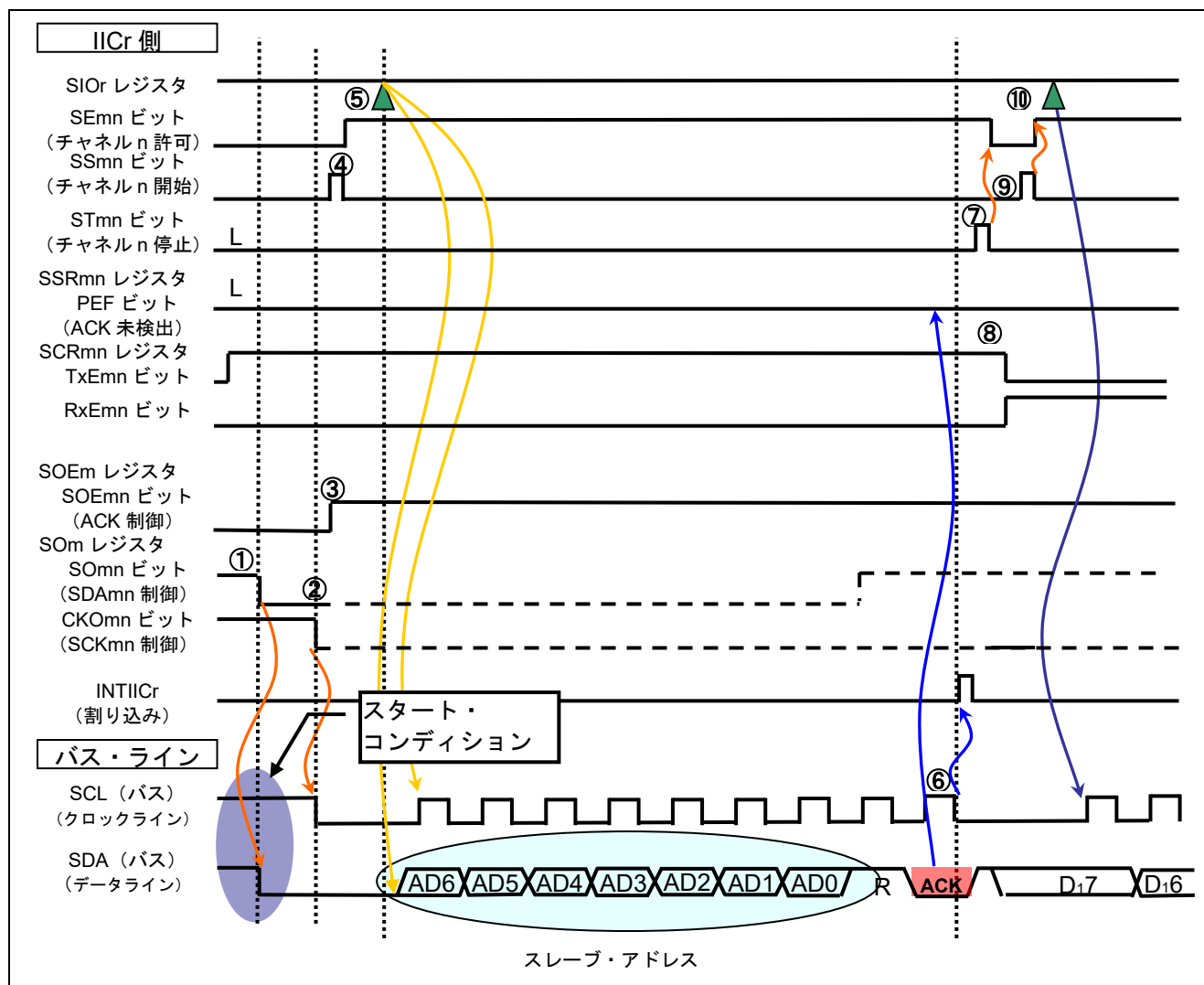


図 1.6 IIC 通信タイミング・チャート (スレーブ→マスタ通信例) (1/3)

- ① IICmn の初期設定 (ここで、SCRmn レジスタは TxEmn=1, RxEmn=0 に設定しておきます) が完了したら、SOmn レジスタの SOmn ビット=0 に設定して SDA 信号を立ち上げることでスタート・コンディションを発行します。
- ② スタート・コンディションのホールド時間 (標準モードで 4.0μs、ファースト・モードで 0.6μs) 経過したら、SOmn レジスタの CKOmn ビット=0 に設定して SCL 信号を立ち下げます。
- ③ 通信のために SOEm レジスタの SOEmn ビット=1 に設定して出力を許可します。
- ④ SSmn レジスタの SSmn ビット=1 にしてチャンネル n を動作許可状態にします。
- ⑤ SIOr レジスタにスレーブのアドレスを書き込むと、通信が起動します。
- ⑥ スレーブ・アドレスの送信が完了すると INTIICr が発生します。スレーブからの ACK 応答を SSRmn レジスタの PEF ビットで確認します。
- ⑦ 通信方向を切り替えるために IICmn の動作を禁止状態にします。
- ⑧ SCRmn レジスタを TxEmn=0, RxEmn=1 に変更します。
- ⑨ IICmn を動作許可状態にします。
- ⑩ SIOr レジスタにデミーデータ (0FFH) を書き込んで受信を起動します。

(6) スレーブ→マスタ通信 2 (アドレス～データ～データ)

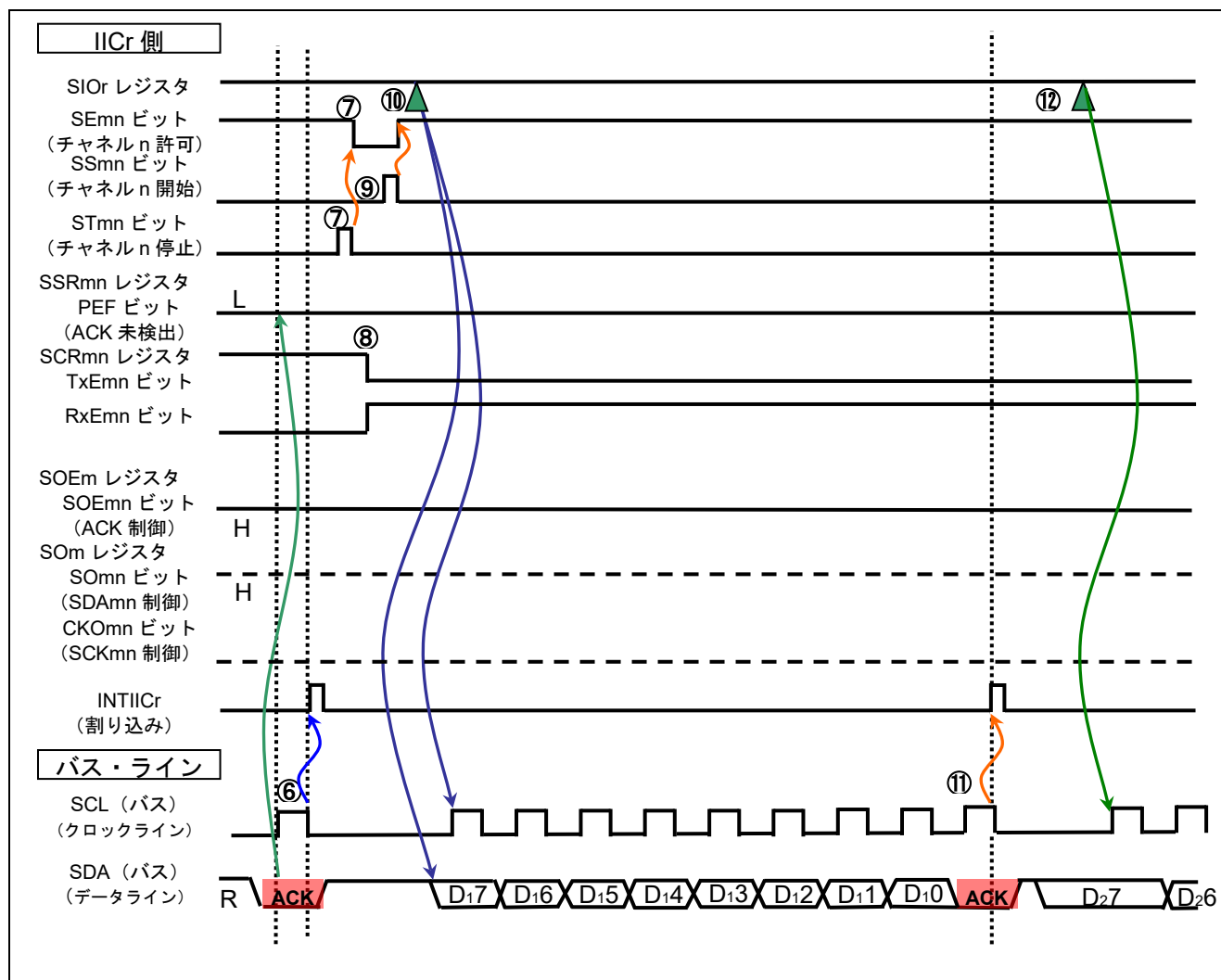


図 1.7 IIC 通信タイミング・チャート (スレーブ→マスタ通信例) (2/3)

- ⑥ スレーブ・アドレスの送信が完了すると INTIICr が発生します。スレーブからの ACK 応答を SSRmn レジスタの PEF ビットで確認します。
- ⑦ 通信方向を切り替えるために IICmn の動作を禁止状態にします。
- ⑧ SCRmn レジスタを TxEmn=0、RxEmn=1 に変更します。
- ⑨ IICmn を動作許可状態にします。
- ⑩ SIOr レジスタにデミーデータ (0FFH) を書き込んで受信を起動します。
- ⑪ SOEmn ビット=1 なので、SCL の 9 クロック目で ACK 応答を行い、受信完了の INTIICr が発生します。
- ⑫ 次のデータ (最後のデータではない) を受信するために SIOr レジスタにデミーデータ (0FFH) を書き込んで受信を起動します。

(7) スレーブ→マスタ通信 3 (データ～データ～ストップ・コンディション)

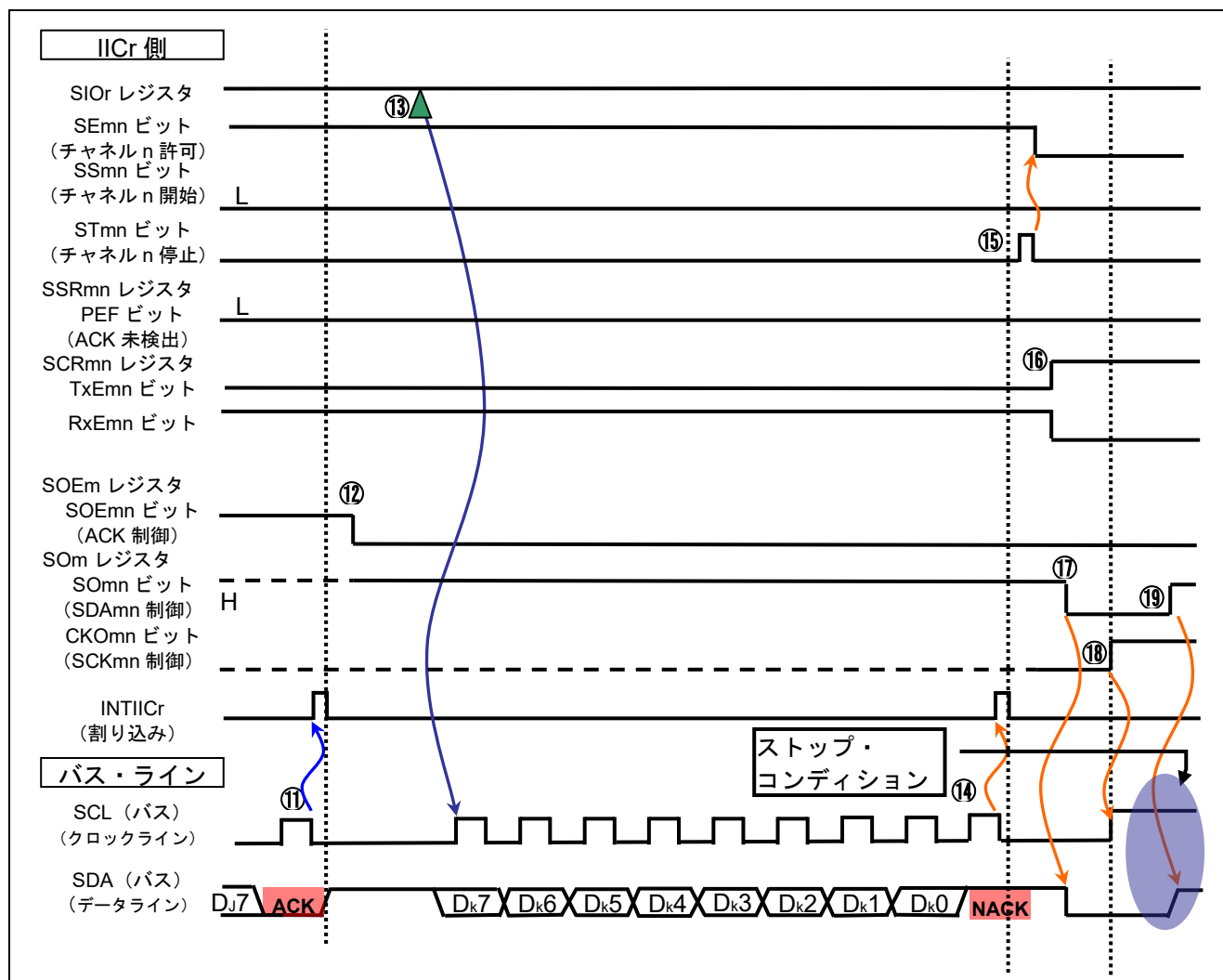


図 1.8 IIC 通信タイミング・チャート (スレーブ→マスタ通信例) (3/3)

- ⑪ SOEmn ビット=1 なので、SCL の 9 クロック目で ACK 応答を行い、受信完了の INTIICr が発生します。
- ⑫ 最後の受信データに NACK 応答するために SOEm レジスタの SOEmn ビット=0 に設定してシリアル出力を禁止します。
- ⑬ SIOr レジスタにデミーデータ (0FFH) を書き込んで受信を起動します。
- ⑭ SOEmn ビット=0 なので、SCL の 9 クロック目で NACK 応答を行い、受信完了の INTIICr が発生します。
- ⑮ STmn レジスタの STmn ビット=1 にしてチャンネル n を動作禁止状態にします。
- ⑯ 次の通信に備えて、SCRmn レジスタを初期設定と同じ状態に設定します。
- ⑰ ストップ・コンディション発行準備で SDA を立ち下げるために SOmn レジスタの SOmn ビット=0 に設定します。
- ⑱ ストップ・コンディション発行準備で SCL を立ち上げるために SOmn レジスタの CKOmn ビット=1 に設定します。
- ⑲ ストップ・コンディションのセットアップ時間が経過したら、SOmn レジスタの SOmn ビット=1 に設定することで、ストップ・コンディションが発行されます。

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/G23 (R7F100GLG)
動作周波数	● 高速オンチップオシレータ (HOCO) クロック : 32 MHz ● CPU/周辺ハードウェア・クロック : 32MHz
動作電圧	● 3.3V ● LVD0 動作 (VLVD0) : リセット・モード 立ち上がり時 TYP. 1.90V 立ち下がり時 TYP. 1.86V
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ V8.13.00
C コンパイラ (CS+)	ルネサス エレクトロニクス製 CC-RL V1.15.00
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 V2025-01 (25.1.0)
C コンパイラ (e ² studio)	ルネサス エレクトロニクス製 CC-RL V1.15.00
統合開発環境 (IAR)	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 V5.10.3
コンパイラ (IAR)	IAR Systems 製 IAR C/C++ Compiler for Renesas RL78 V5.10.3.2716
使用ボード	RL78/G23 Fast Prototyping Board (RTK7RLG230CLG000BJ) + EEPROM (R1EX24016)

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/G23 複数スレーブ・アドレス対応 I2C (マスタ) (R01AN5825J) アプリケーションノート
- RL78/G23 複数スレーブ・アドレス対応 I2C (スレーブ) (R01AN5670J) 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。

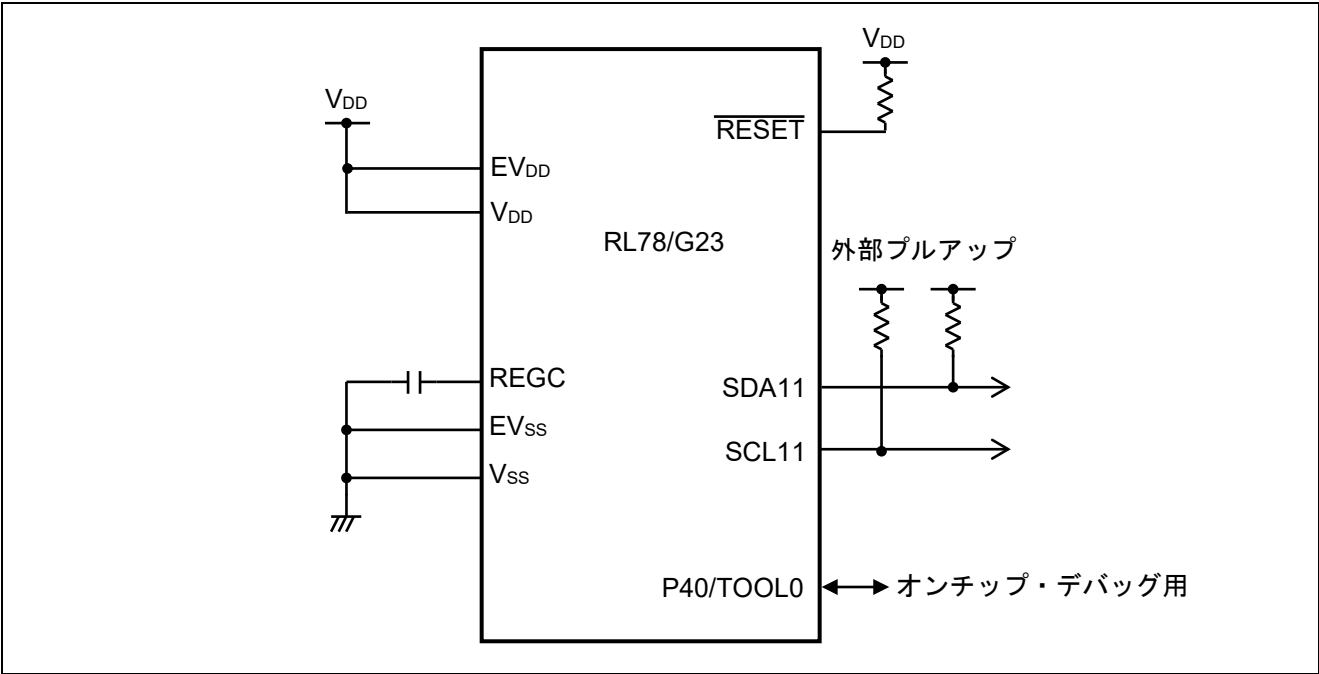


図 4.1 ハードウェア構成

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（入力専用ポートは個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい）。

- 2 EV_{SS} で始まる名前の端子がある場合には V_{SS} に、 EV_{DD} で始まる名前の端子がある場合には V_{DD} にそれぞれ接続してください。
- 3 V_{DD} は $LVD0$ にて設定したリセット解除電圧（ V_{LVD0} ）以上にしてください。

4.2 使用端子一覧

表 4.1に使用端子と機能を示します。

表 4.1 使用端子と機能

端子名	入出力	内容
P30/SCL11	入出力	IIC11 のシリアル・クロック出力端子
P50/SDA11	入出力	IIC11 のシリアル・データ送受信端子

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、シリアル・インタフェース IIC11 による、IIC 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）動作を行います。制御は API としての使用を考慮して、できるだけ割り込みで処理を行っています。

- (1) シリアル・アレイ・ユニット 0 のチャンネル 3 を簡易 IIC で初期設定を行います。

＜設定条件＞

- 動作クロックを CK00 (32MHz) に設定します。
- 動作モードを簡易 IIC に設定します。
- 転送完了割り込みに設定します。
- データとクロックの位相をタイプ 1 に設定します。
- データ 8 ビット長、ストップ・ビット 1 ビット、パリティなし、MSB ファースト転送に設定します。
- 転送クロックをファースト・モードの 381kHz に設定します。
- SO03、CKO03 を 1 に設定します。
- P30/SCL11 端子を転送クロック出力用に、P50/SDA11 端子をデータ送信／データ受信用に設定します。

- (2) タイマ・アレイ・ユニットのチャンネル 2 を書き込み完了確認用に 100 μ s のインターバル・タイマに設定します。

- (3) 使用する EEPROM (16K ビット) のパラメータを処理用パラメータの構造体にコピーします。

- (4) ストップ・コンディション発行し、バスを開放状態にします。

- (5) 256 バイトの書き込むデータ (インクリメント・パターン) を作成します。

- (6) アクセス・パラメータ (構造体 g_PARAI) を設定します。

- (7) 256 バイトのデータを EEPROM の 0x400 番地から書き込みます。

- (8) 書き込んだデータの前後に 1 バイト分を含めて読みだします。

- (9) 16 バイトの同一データ (0xkk) を準備します。(kk=00、11、22、・・・、77)

- (10) EEPROM の 0xk00 番地から 16 バイトに書き込みます。

- (11) k の値を 0~7 に変化させて、(9)~(10)を繰り返します。

- (12) (9)~(10)で書き込んだデータの前後 8 バイトを含めた 32 バイトを読み出します。

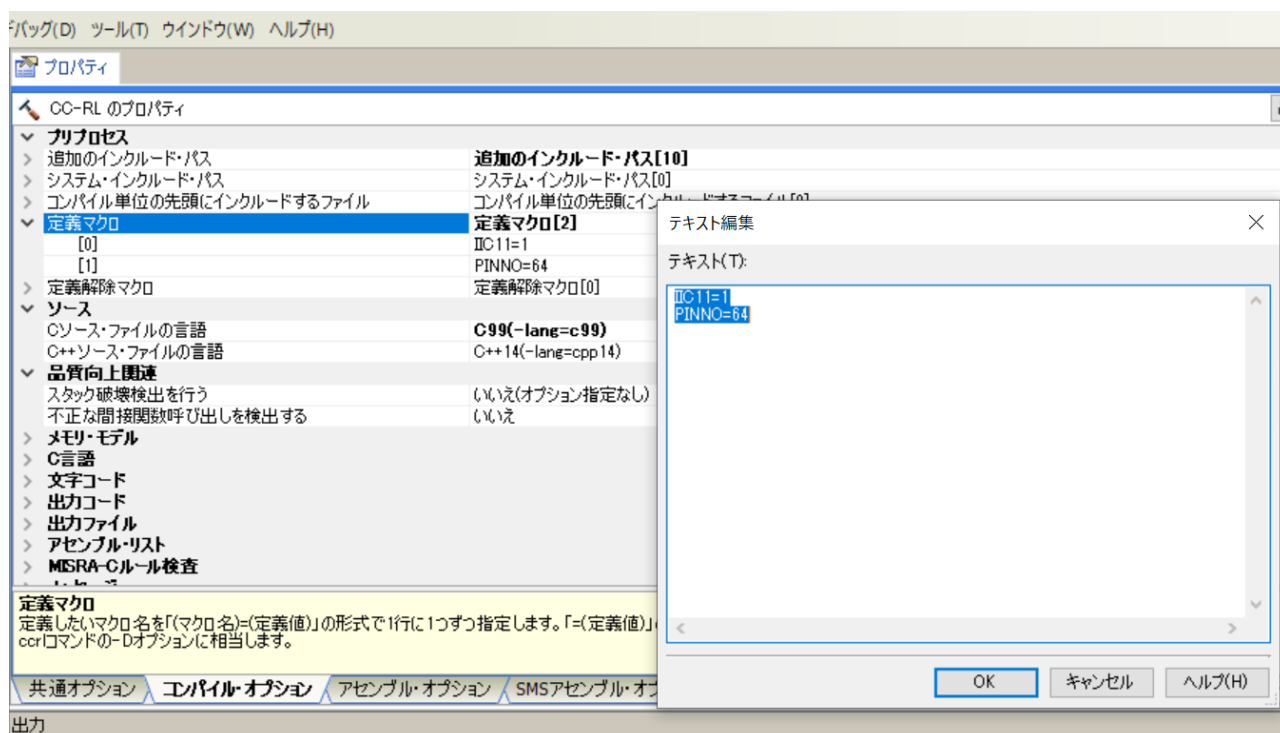
注意 本サンプルコードは、RL78/G23 の簡易 IIC11 を利用した IIC バスによる EEPROM (R1EX24016) の制御例を示すものです。使用するチャネルや EEPROM を変更した場合には十分に評価してご利用ください。

備考 本プロジェクトでは、「コンパイラ・オプション」の「定義マクロ」機能を使用して使用する簡易 IIC チャンネルとデバイスのピン数を指定しています。以下は CS+環境での設定方法です。

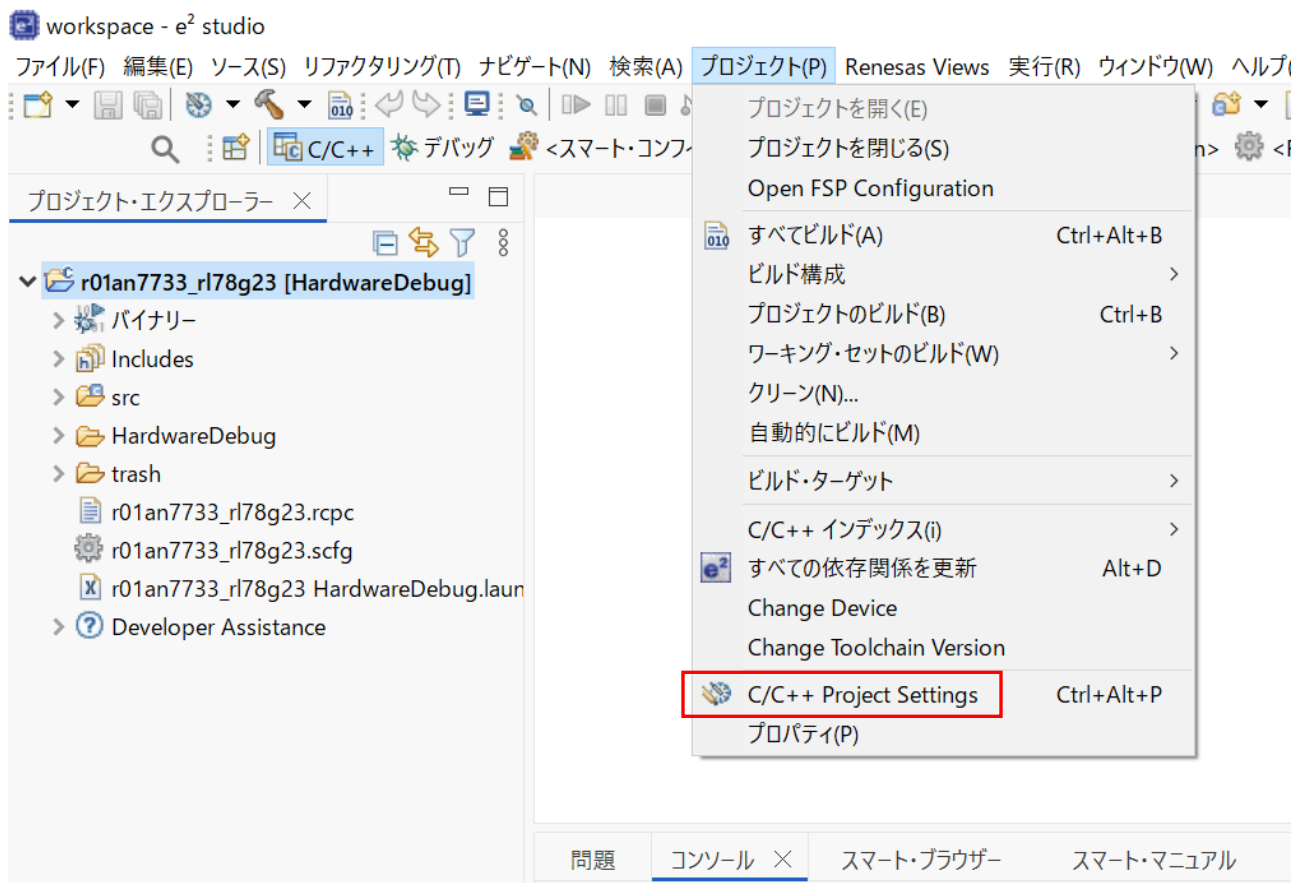
定義マクロの[0]は使用する簡易 IIC のチャンネルとして、IIC00 を選択しているものです。例えば、チャンネル 20 を使用する場合には、ここを「IIC20=1」と変更してください。

定義マクロの[1]は使用する製品のピン数を指定しています。定義マクロの[0]で指定したチャネルが対象の製品に存在するかのチェックに使用しています。

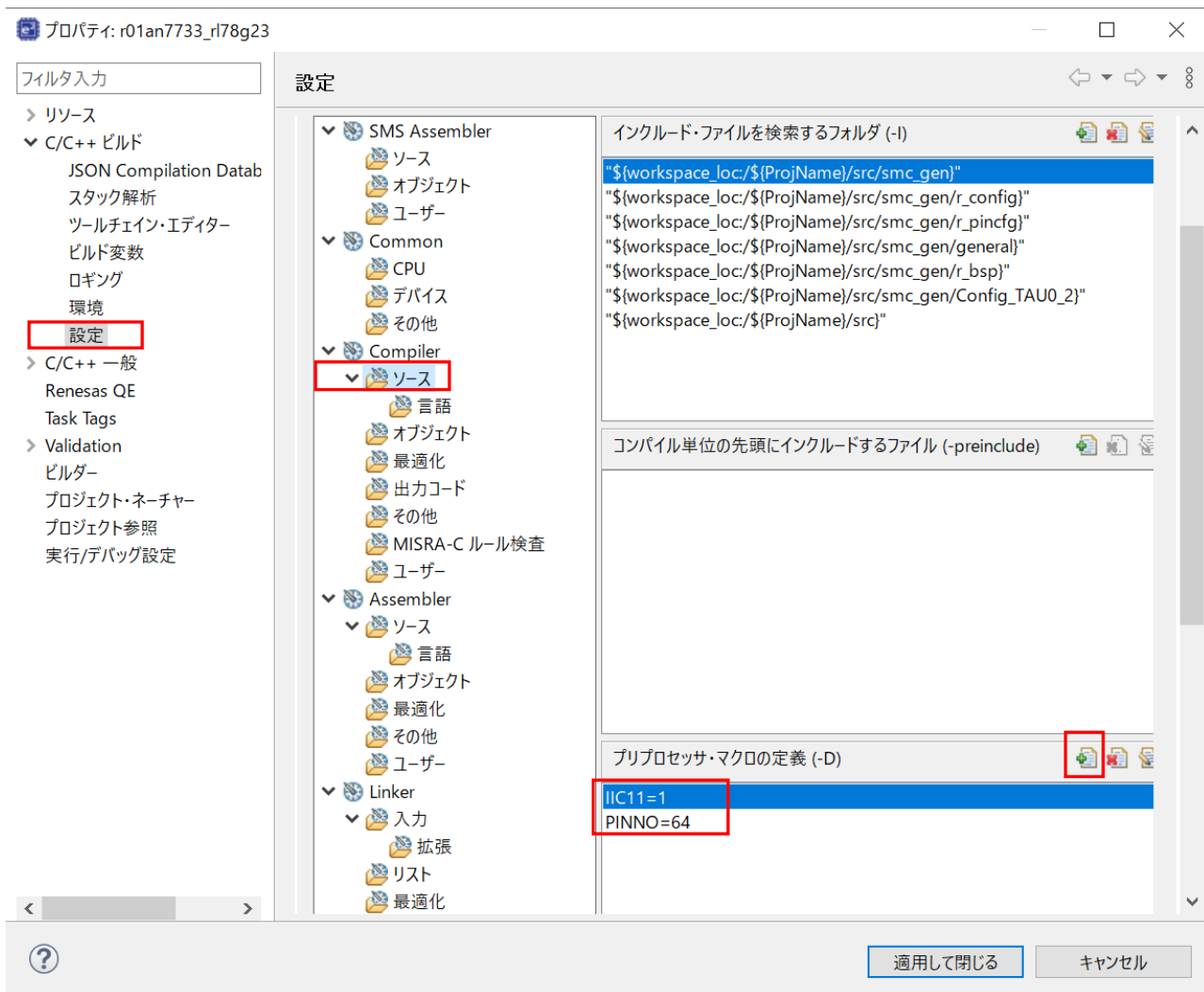
これらを変更するには[定義マクロ]の右端のボタンをクリックしてテキスト編集画面を開いて、そこで編集します。



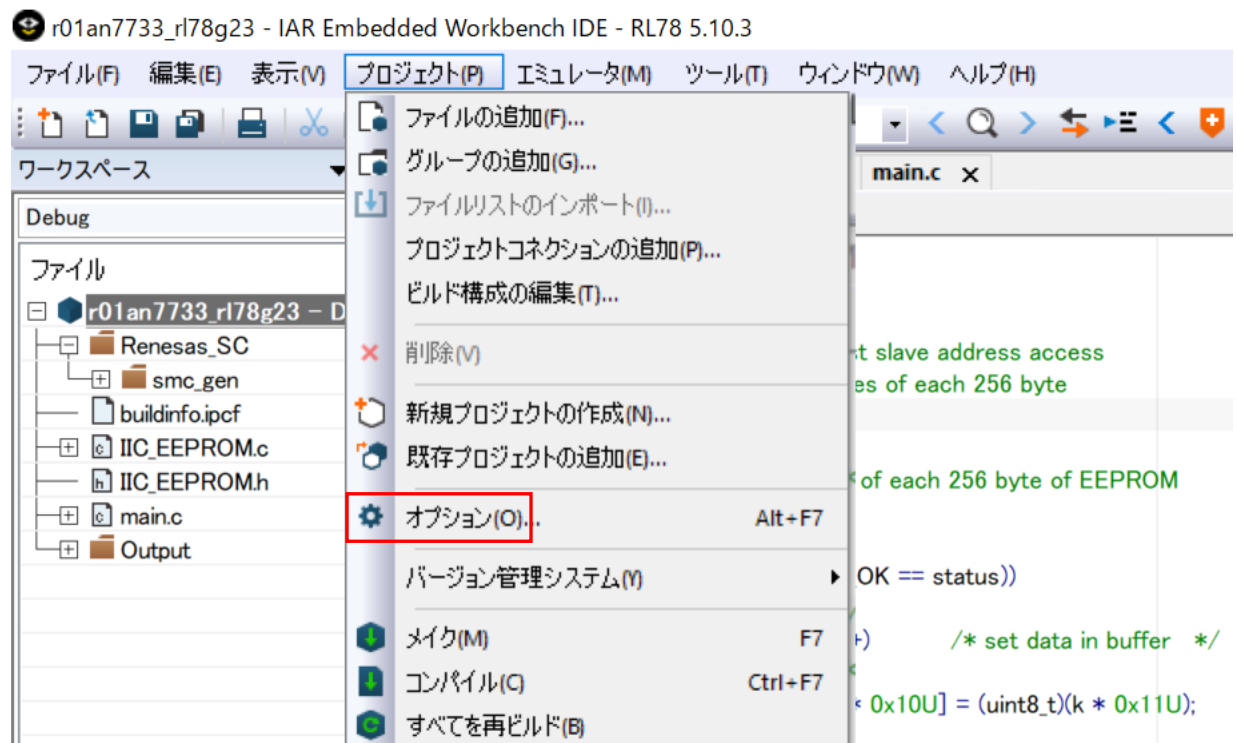
e2studio 環境では以下の手順で設定します。まず、「C/C++ Project Settings」をクリックします。



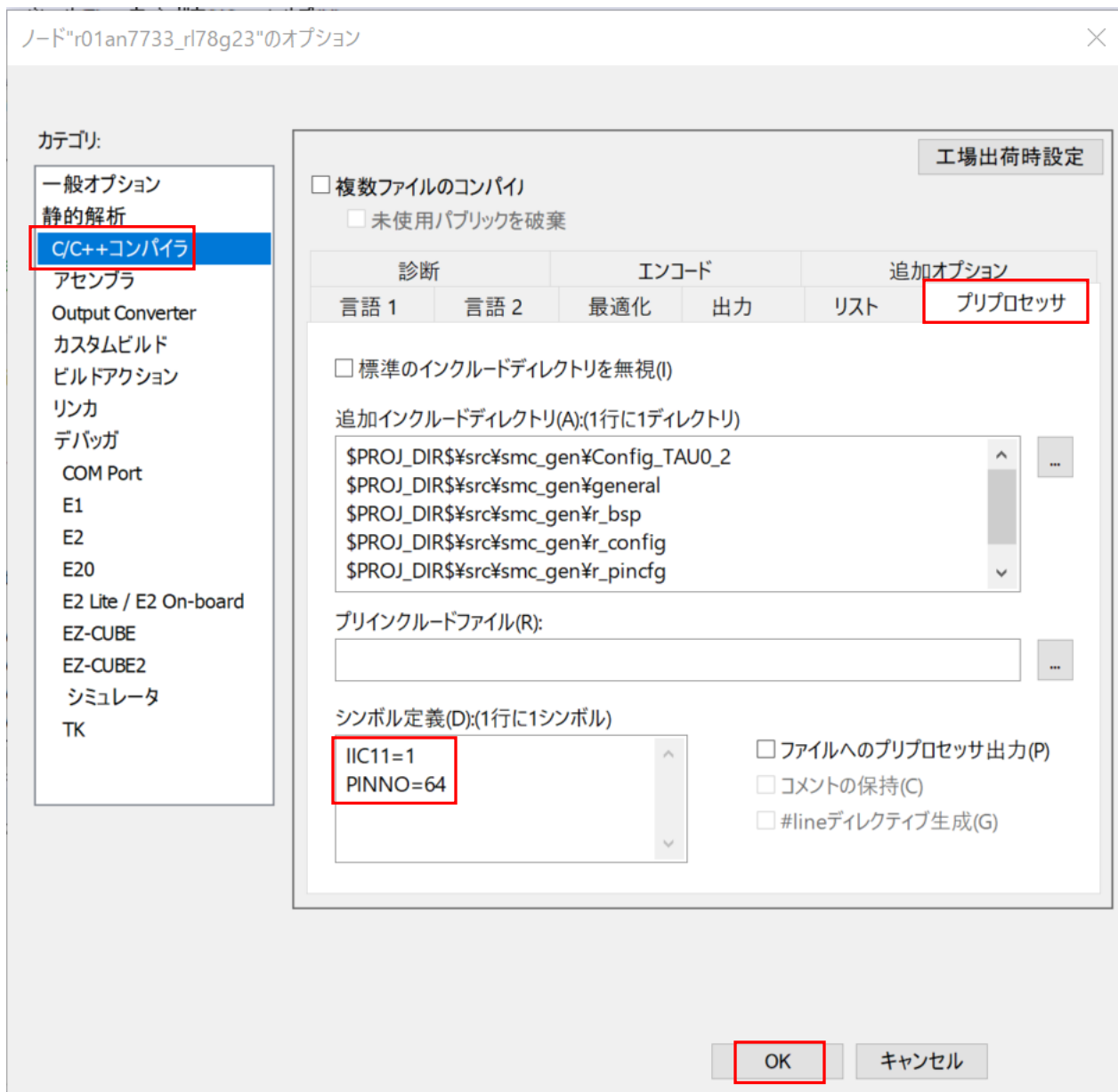
次に、「C/C++ ビルド」の「設定」を選択します。「ツール設定」のタブを開き、「Compiler」の「ソース」を選択します。そして「定義マクロ」の欄にCS+の定義マクロ機能で設定した定義を設定します。最後に「追加」ボタンをクリックすることで、定義を追加することができます。



IAR 環境では以下の手順で設定します。まず、「プロジェクト」の「オプション」をクリックします。



次に、「C/C++ コンパイラ」の「プリプロセッサ」を選択します。そして「シンボル定義」の欄にCS+の定義マクロ機能で設定した定義を設定します。最後に「OK」ボタンをクリックすることで、定義を追加することができます。



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
000C0H/040C0H	01101110B	ウォッチドッグ・タイマ 動作停止 (リセット解除後、カウント停止)
000C1H/040C1H	11111110B	LVD0 リセット・モード 検出電圧：立ち上がり 1.90V／立下り 1.86V
000C2H/040C2H	11101000B	HS モード、HOCO : 32MHz
000C3H/040C3H	10000101B	オンチップ・デバッグ動作許可

5.3 定数一覧

表 5.2、5.3 にサンプルコードで使用する定数を示します。また、表 5.4 に使用するチャネル番号を指定するマクロ名の一覧を示します。

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
PAGE_16	0x000F	2K~16K ビット EEPROM のページのデータ数
PAGE_32	0x001F	32K、64K ビット EEPROM のページのデータ数
PAGE_64	0x003F	128K、256K ビット EEPROM のページのデータ数
PAGE_128	0x007F	512K ビット EEPROM のページのデータ数
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)
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
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
BUS_ERR	0x60	バスが開放状態 (SDA がハイ) にならない
SVAMSK	0b11111110	スレーブ・アドレスのビット 0 マスク用データ
SCLLOWW	0x05	SCL のロウ・レベル時間計測用データ
SCLHIGHW	0x02	SCL のハイ・レベル時間計測用データ
RETRYCNT	0x09	SCL のダミー・クロックのパルス数

表 5.3 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	

表 5.4 サンプルコードで使用するマクロ名

マクロ名	設定値 (チャンネル 3)	内容
SAUmEN	SAU0EN	使用する周辺イネーブル・レジスタ
SPSm	SPS0	シリアル・クロック選択レジスタ
SMRmn	SMR03	シリアル・モード・レジスタ
SCRmn	SCR03	シリアル通信動作設定レジスタ
SDRmn	SDR03	シリアル・データ・レジスタ
SIOr	SIO03	シリアル・データ・レジスタ (送受信データ用)
SSRmn	SSR03	シリアル・ステータス・レジスタ
SIRmn	SIR03	シリアル・フラグ・クリア・トリガ・レジスタ
SSmL	SS0L	シリアル・チャンネル開始レジスタ
STmL	ST0L	シリアル・チャンネル停止レジスタ
TRGONn	0b00001000	トリガ用ビット
SOEmL	SOE0L	シリアル出力許可レジスタ
SOEON	TRGONn	シリアル出力許可レジスタ設定用ビット
SOEOFF	(uint8_t)(~SOEON)	シリアル出力許可レジスタクリア用ビット
SOm	SO0	シリアル出力レジスタ
SDAHIGH	TRGONn	SDA 立ち上げ用ビット
SDALOW	~SDAHIGH	SDA 立ち下げ用ビット
SCLHIGH	TRGONn × 0x100	SCL 立ち上げ用ビット
SCLLOW	~SCLHIGH	SCL 立ち下げ用ビット
IICIFr	IICIF11	割り込み要求フラグ・レジスタ
IICMKr	IICMK11	割り込み要求マスク・レジスタ
PM_IICr	PM5	IICr 用ポート・モード・レジスタ (SDA 信号用を指定)
PM_SDAr	PM5_bit.no0	SDA 用ポート・モード・レジスタ・ビット
PM_SCLr	PM3_bit.no0	SCL 用ポート・モード・レジスタ・ビット
POM_IICr	POM5	ポート出力モード・レジスタ
P_IICr	P3	IICr 用ポート
P_SDAr	P5_bit.no0	SDA 用ポート
P_SCLr	P3_bit.no0	SCL 用ポート
SDAINMODE	0b00000001	ポート・モード・レジスタ入力指定用ビット
SDAOUTMODE	0b11111100	ポート・モード・レジスタ出力指定用ビット
SDASCLON	(uint8_t)(~SDAOUTMODE)	ポート設定用ビット
IICPR0r	IICPR011	割り込み優先度設定レジスタ
IICPR1r	IICPR111	

5.4 変数一覧

表 5.5 にグローバル変数を示します。

なお、g_eeprom_type 以下はモジュール内グローバルな変数です。

表 5.5 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_IICr_Tx_addr1() R_IICr_Tx_addr2() R_IICr_Rx_RST() R_IICr_RxData_ST() R_IICr_RxData() R_IICr_Rx_LastData() R_EEPROM_W() R_EEPROM_wait_write() R_IICr_TxDataST() R_IICr_TxData() R_EEPROM_next_page() IINTIICr()
uint8_t 配列 (256)	g_data_bufferW1	書き込みデータバッファ	main()
uint8_t 配列 (256)	g_data_bufferW2	書き込みデータバッファ	main()
uint8_t 配列 (512)	g_data_bufferR1	読みだしデータバッファ	main()

Type	Variable Name	Contents	Function Used
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 eepromaddr; uint8_t *bufferaddr; uint16_t number;	g_PARAA	EEPROM アクセス用パラメータ	R_EEPROM_R() R_IICr_Tx_addr1() R_IICr_Tx_addr2() R_IICr_Rx_RST() R_IICr_RxData_ST() R_IICr_RxData() R_IICr_Rx_LastData() R_EEPROM_W() R_IICr_TxDataST() R_IICr_TxData() R_EEPROM_Devid() SINTTM02() R_EEPROM_wait_write()
構造体 eeprom_paraA16	g_PARAC	EEPROM アクセス指定用パラメータのコピー	check_EEPROM_Addr() R_EEPROM_R() R_EEPROM_W() R_IICr_TxData() get_slave_Addr() R_EEPROM_Devid()
構造体 eeprom_info uint16_t page_size; uint16_t rom_size; uint8_t addr_mask; uint8_t mask2;	EEPROM_Info	使用するEEPROMのパラメータ保持用 (処理用)	R_device_select() R_IICr_Tx_addr1() get_slave_Addr() R_EEPROM_Devid()

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
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 の読みだし完了を待ちます。
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	EEPROM サイズが 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_EEPROM_Device	書き込み時に書き込みデータがページに収まるように 1 回の書き込みパラメータを補正します。
R_IICr_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_IICr_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_IICr_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_IICr_RxData_ST	データ受信のため、Tx E→0、Rx E→1 に設定してデータ受信を起動します。読み出すデータが 1 バイトの時には NACK 応答に設定して受信を起動します。
R_IICr_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。
R_IICr_Rx_LastData	最後の受信データをバッファに格納し、次の通信に備えて Tx E→1、Rx E→0 に設定し、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_IICr_TxDataST	EEPROM への書き込みデータの送信を開始します。
R_IICr_TxData	データ送信完了し、ページへの書き込みデータが全て送り終わったら、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。全データの処理が完了していれば、処理を終了します。残りのデータがあれば、次のページ書き込みの準備をして、書き込み完了待ちのタイマを起動します。
R_EEPROM_next_page	EEPROM での書き込みが完了したので、タイマを停止し、EEPROM の次ページのセルのアドレスの送信を行います。

関数名	概要
R_IICr_StartCond	SOM レジスタを操作してスタート・コンディションを発行し、IICr の動作を許可します。
R_IICr_StopCond	ストップ・コンディションを発行し、バスが開放されたかを確認します。バスが開放されていない場合は、SCL にダミー・クロックを出力し、再度ストップ・コンディションを発行します。
R_IICr_send_Stop	IICr を動作禁止にし、SOM レジスタを操作してストップ・コンディションを発行します。
R_IICr_wait_bus	SCL 信号に 9 パルスのダミー・クロックを出力して SDA 信号が開放 (ハイになる) か確認します。
R_IICr_SCL_pulse	SCL 信号にダミー・クロックを出力します。
R_IICr_SCL_high	SCL 信号を立ち上げ、ハイ・レベル幅の時間待ちます。
R_IICr_SCL_low	SCL 信号を立ち下げ、ロウ・レベル幅の時間待ちます。
R_IICr_NACK	スレーブからの ACK/NACK 応答を確認します。
R_IICr_SCL_Time	SCL 信号のロウ期間を待ちます。
R_IICr_SCL_highTime	SCL 信号のハイ期間を待ちます。
IINTIICr	IICr の転送完了割り込みで、スレーブの ACK 応答を確認し、次の処理に割り振ります。
INTTM02	TM02 のインターバル割り込み処理で、書き込み完了を確認するためのスレーブ・アドレス送信を行います。
R_IICr_Init	IICr の初期設定を行います。

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(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_EEPROM_R

概要	EEPROM からの読みだし開始要求処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	void R_EEPROM_R(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(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(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_EEPROM_Devide

概要	EEPROM 書き込み時のページ分割処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_EEPROM_Devide(void);
説明	使用する EEPROM のページにあわせて、書き込みデータを分割します。その回で書き込むパラメータを構造体 g_PARAA に設定し、残りのデータの情報を構造体 g_PARAC に設定します。
引数	なし -
リターン値	なし
備考	構造体 g_PARAC のメンバ eepromaddr、bufferaddr が次の書き込みパラメータ、number が次回以降のデータ数

[関数名] R_IICr_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_IICr_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	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_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_IICr_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	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_Rx_RST

概要	受信モードでリスタート処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_Rx_RST(void);
説明	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。
引数	なし -
リターン値	なし
備考	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_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_IICr_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、IICr を送信 (TxE=1、RxE=0) から受信 (TxE=0、RxE=1) に変更して受信を起動 (SIOr にダミー・データ書き込み) します。読み出すデータが 1 バイトの時には NACK 応答に設定して受信を起動します。
引数	なし -
リターン値	なし
備考	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_RxData

概要	データ受信処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_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。読み出すデータが 1 バイトの時には NACK 応答に設定して受信を起動します。
引数	なし -
リターン値	なし
備考	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_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_IICr_Rx_LastData (void);
説明	受信したデータをバッファに格納し、IICrの動作を停止します。次の通信に備えてTxE→1、RxE→0に設定し、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	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_TxDataST

概要	データ送信開始処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_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	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_TxData

概要	データ送信処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_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。ページへの書き込みデータが全て送り終わったなら、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。全データの処理が完了していれば、処理を終了します。残りのデータがあれば、次のページ書き込みの準備をして、書き込み完了待ちのタイマを起動します。
引数	なし -
リターン値	なし
備考	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_EEPROM_next_page

概要	書き込み待ちが完了し、次ページのアドレスの指定処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_EEPROM_next_page(void);
説明	ページへの書き込み完了で、書き込み待ちタイマを停止し、EEPROMの次ページのセルのアドレスの送信を行います。
引数	なし -
リターン値	なし
備考	INTIICmn 処理 (関数名 : IINTIICr) で使用。書き込み完了チェックで、スレーブ・アドレス送信へのACK応答で起動されます。

[関数名] R_IICr_StartCond

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_IICr_StartCond(void);
説明	IICr の動作を一旦禁止してスタート・コンディションの発行を行い、IICr の動作を許可します。
引数	なし -
リターン値	なし
備考	INTIICmn 処理 (関数名 : IINTIICr) で使用

[関数名] R_IICr_StopCond

概要	バス開放処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	MD_STATUS R_IICr_StopCond(void);
説明	IICr のストップ・コンディションを生成します。バスが開放できないときは、SCL にダミー・クロックを9クロック出力し、SDA がハイになったことを確認し、再度ストップ・コンディションを発行します。
引数	なし -
リターン値	[I2C_OK]の場合 : バス開放完了 [BUS_ERR]の場合 : バスが開放できなかった
備考	なし

[関数名] R_IICr_send_Stop

概要	ストップ・コンディション発行処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_send_Stop (void);
説明	IICr を動作禁止にし、SOm レジスタを操作してストップ・コンディションを発行します。
引数	なし -
リターン値	なし
備考	内部でストップ・コンディションを発行する場合に使用します。

[関数名] R_IICr_wait_bus

概要	バス開放確認処理
ヘッダ	r_cg_macrodriver.h r_cg_userdefine.h
宣言	static MD_STATUS R_IICr_wait_bus(void);
説明	IICr を停止した状態で、SCL に9パルスのダミー・クロックを発生して SDA 信号を確認します。
引数	なし -
リターン値	[I2C_OK]の場合 : バスを開放できた。 [BUS_ERR]の場合 : バスを開放できない (SDA 信号がハイにならない)
備考	

[関数名] R_IICr_SCL_pulse

概要	SCL 信号のダミー・クロック出力処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_SCL_pulse(void);
説明	S0m レジスタを操作して SCL にロウのパルスを出力します。
引数	なし -
リターン値	なし
備考	なし

[関数名] R_IICr_SCL_high

概要	SCL 信号を立ち上げ処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_SCL_high(void);
説明	S0m レジスタを操作して SCL を立ち上げ、SCL のハイ・レベル期間待ちます。
引数	なし -
リターン値	なし
備考	なし

[関数名] R_IICr_SCL_low

概要	SCL 信号を立ち下げ処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_SCL_low(void);
説明	S0m レジスタを操作して SCL を立ち下げ、SCL のロウ・レベル期間待ちます。
引数	なし -
リターン値	なし
備考	なし

[関数名] R_IICr_NACK

概要	スレーブからの ACK/NACK 応答確認処理
ヘッダ	r_cg_userdefine.h
宣言	static MD_STATUS R_IICr_NACK(void);
説明	SSRmn レジスタのビット 1（PEF ビット=NACK）を戻します。
引数	なし -
リターン値	[0x00]の場合 : ACK 応答あり [0x02]の場合 : ACK 応答なし
備考	ステータス・フラグはチェックするだけで、クリアしません。

[関数名] R_IICr_SCL_Time

概要	SCL 信号のロウ期間待ち処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_SCL_Time(void);
説明	SCL 信号のロウ期間を待ちます。
引数	なし -
リターン値	なし
備考	

[関数名] R_IICr_SCL_highTime

概要	SCL 信号のハイ期間待ち処理
ヘッダ	r_cg_userdefine.h
宣言	static void R_IICr_SCL_highTime(void);
説明	SCL 信号のハイ期間を待ちます。
引数	なし -
リターン値	なし
備考	

[関数名] IINTIICr

概要	IICr 転送完了割り込み
ヘッダ	r_cg_userdefine.h
宣言	static void __near IINTIICr(void);
説明	IINTIICr で起動され、通信の状況に応じて必要な処理を起動します。EEPROM の書き込み完了待ち以外で、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	

[関数名] INTTM02

概要	100 μ s インターバル・タイマ完了割り込み
ヘッダ	r_cg_userdefine.h
宣言	static void __near INTTM02 (void);
説明	書き込み完了確認のためにスタート・コンディションとスレーブ・アドレスを送信します。
引数	なし -
リターン値	なし
備考	結果の確認は IINTIICr で行います。

備考

5.7 フローチャート

5.7.1 メイン関数

図 5.7～図 5.9 にメイン関数のフローチャートを示します。メイン関数は API の使用例として 16K ビット EEPROM への読み書きのテストを行っています。

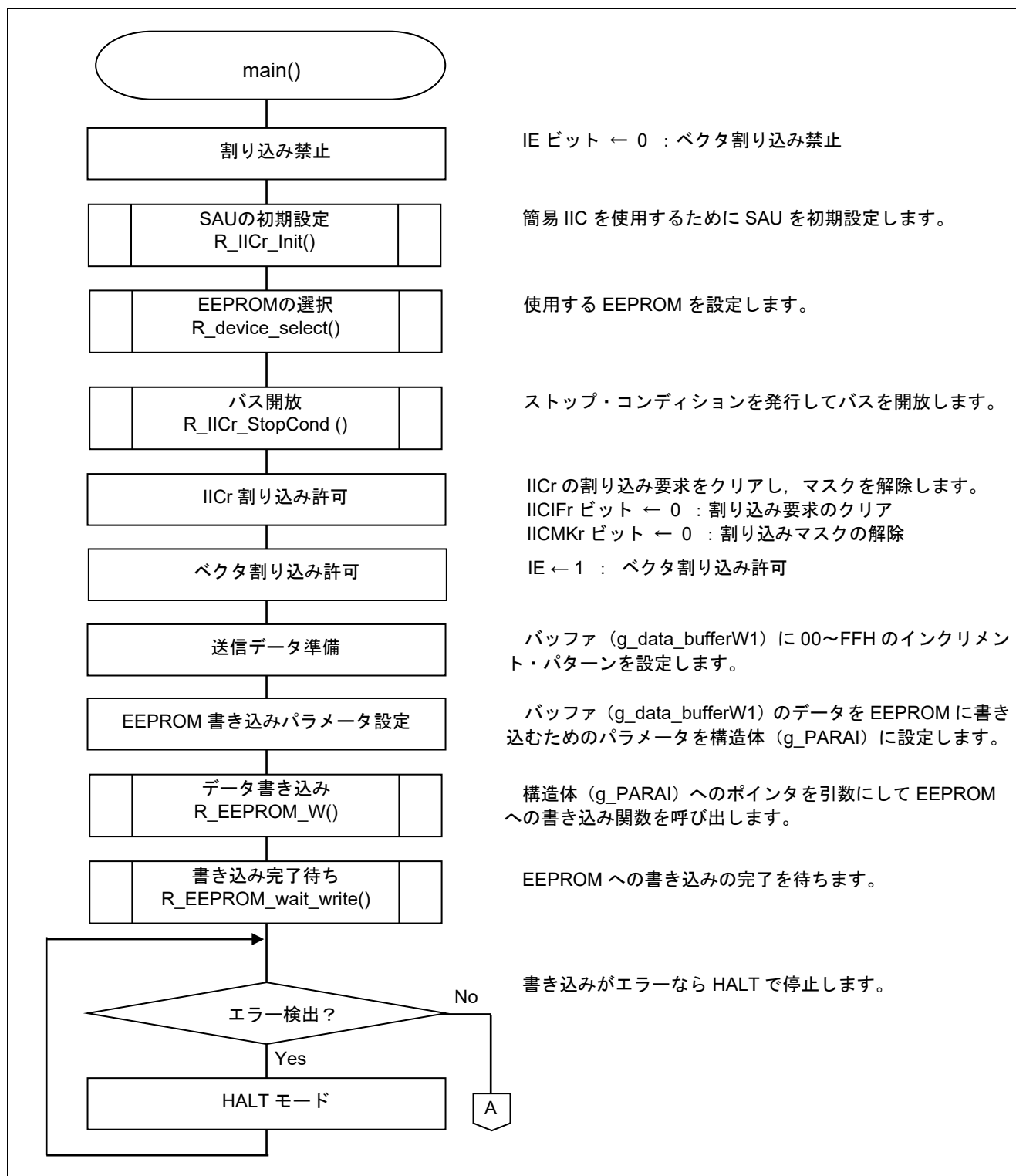


図 5.7 メイン関数 (1/3)

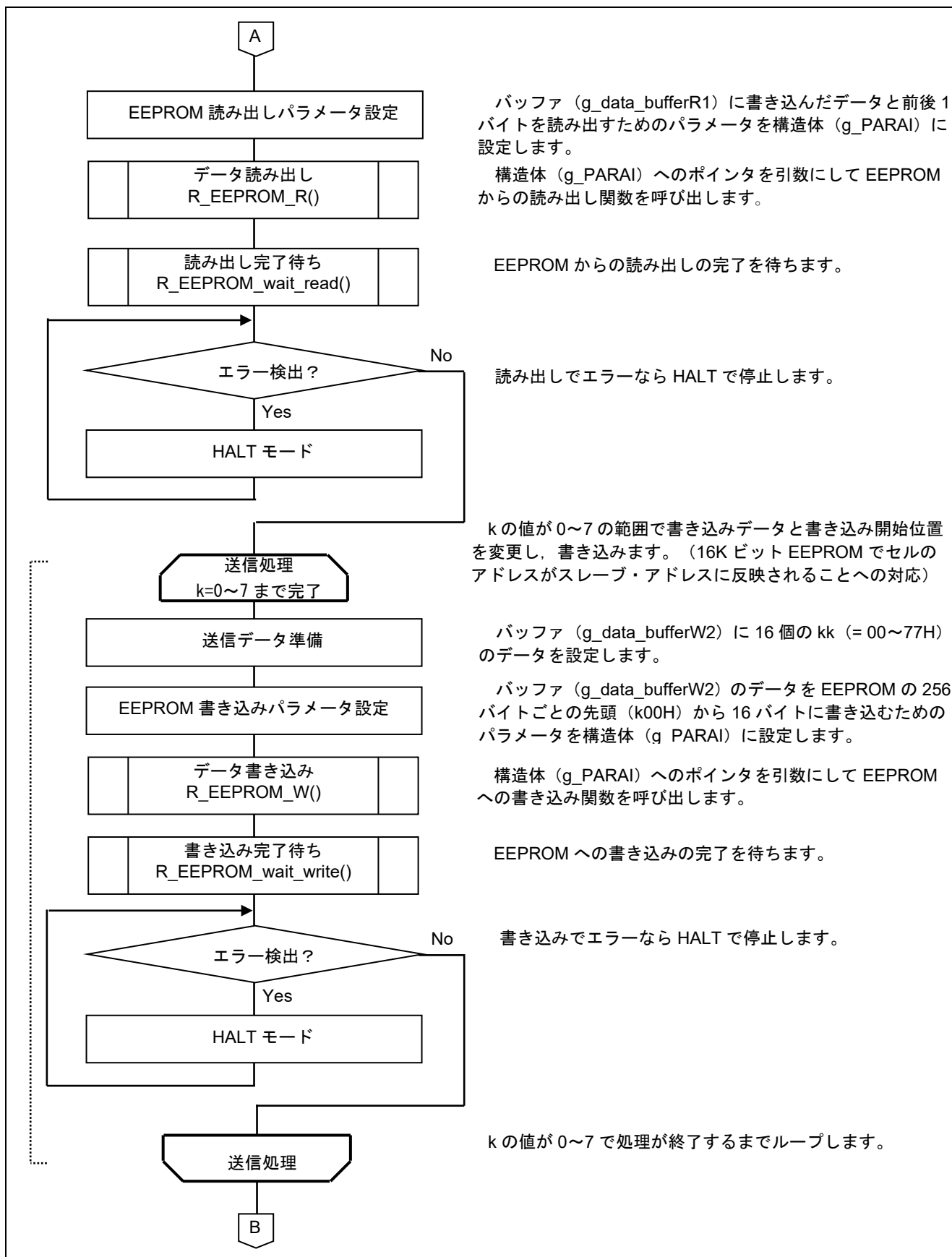


図 5.8 メイン関数 (2/3)

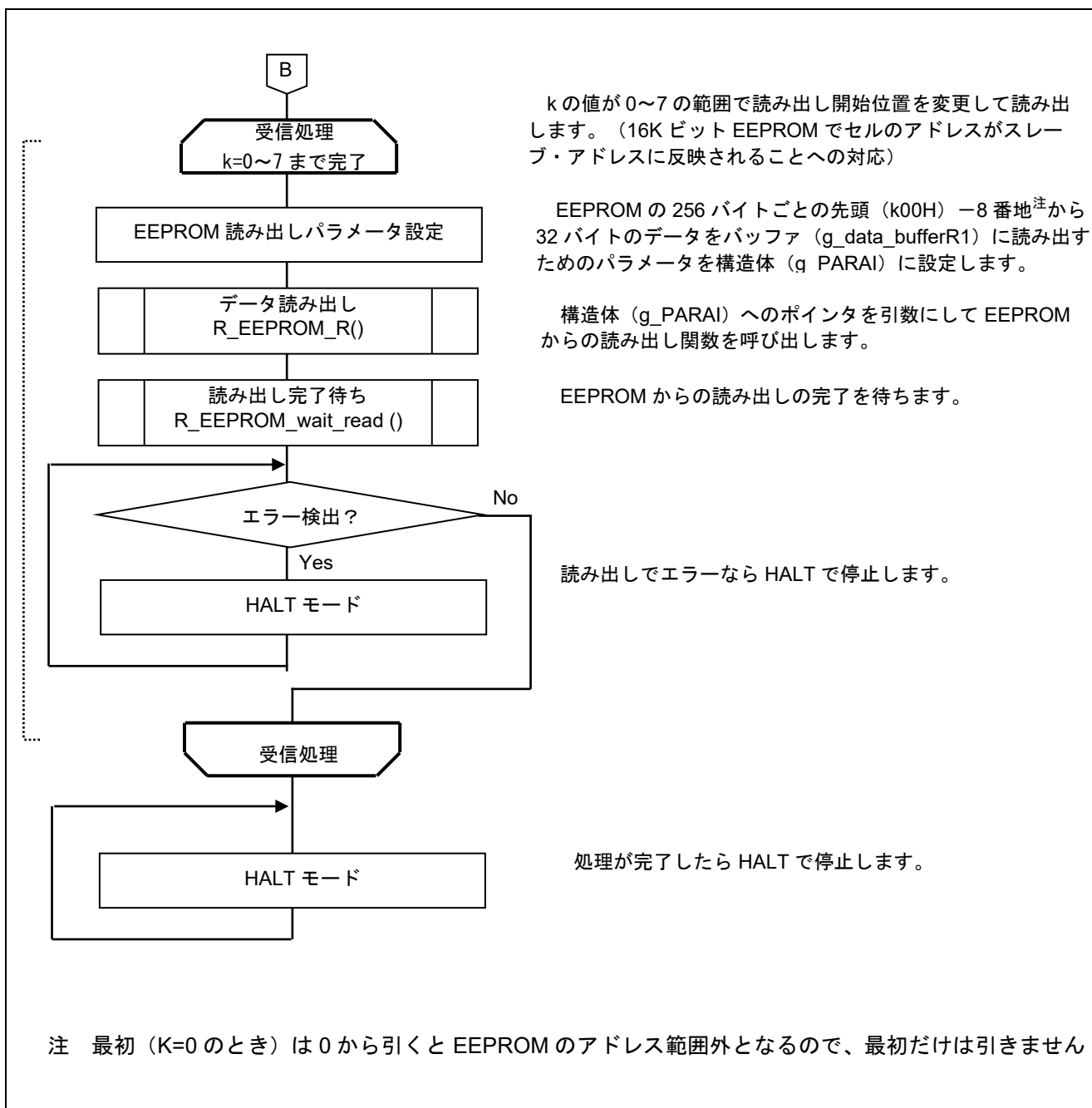


図 5.9 メイン関数 (3/3)

5.7.2 EEPROM の選択

図 5.10 に EEPROM の選択のフローチャートを示します。

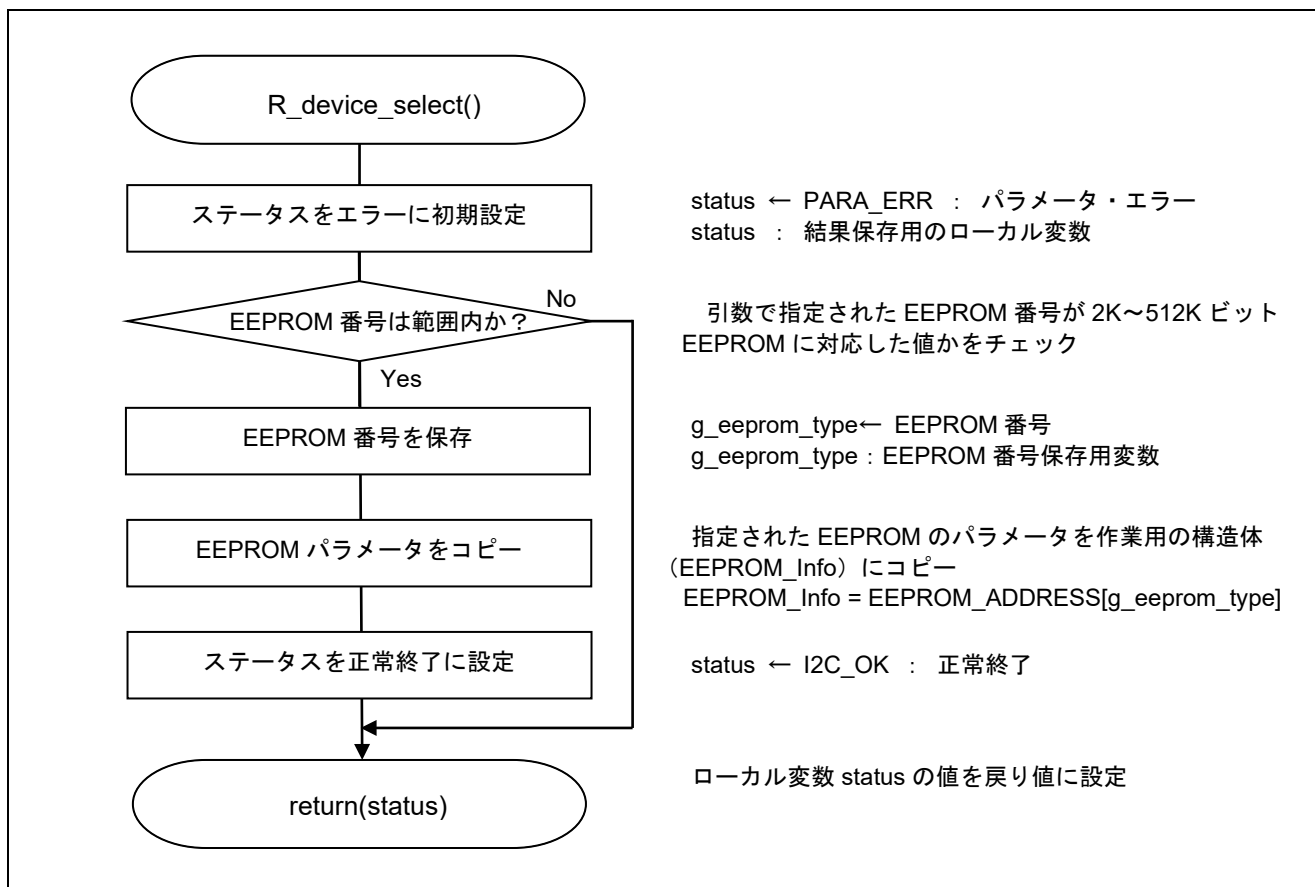


図 5.10 EEPROM の選択

5.7.3 バス開放関数

図 5.11 にバス開放関数のフローチャートを示します。

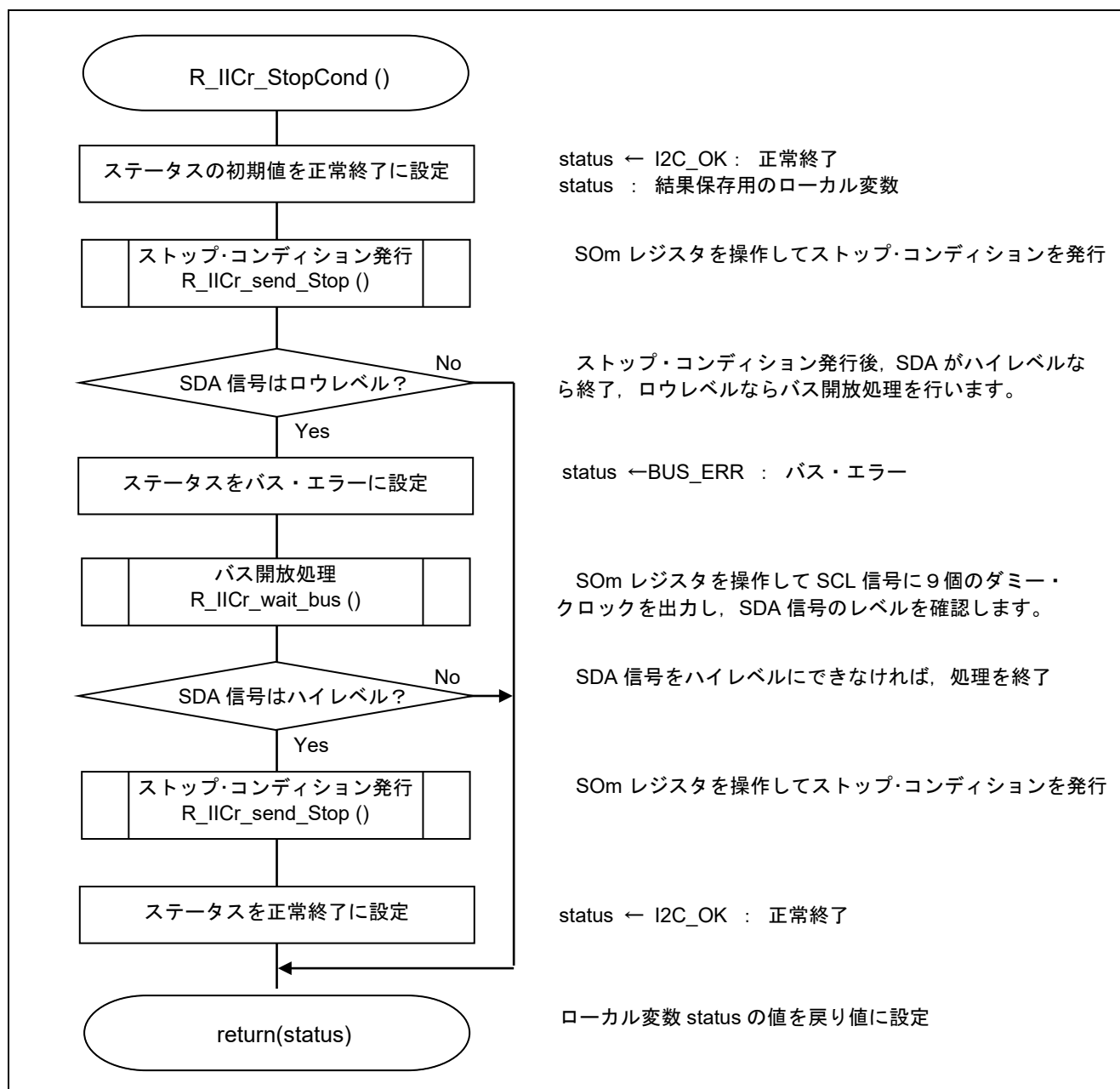


図 5.11 バス開放関数

5.7.4 ストップ・コンディション発行関数

図 5.12 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。

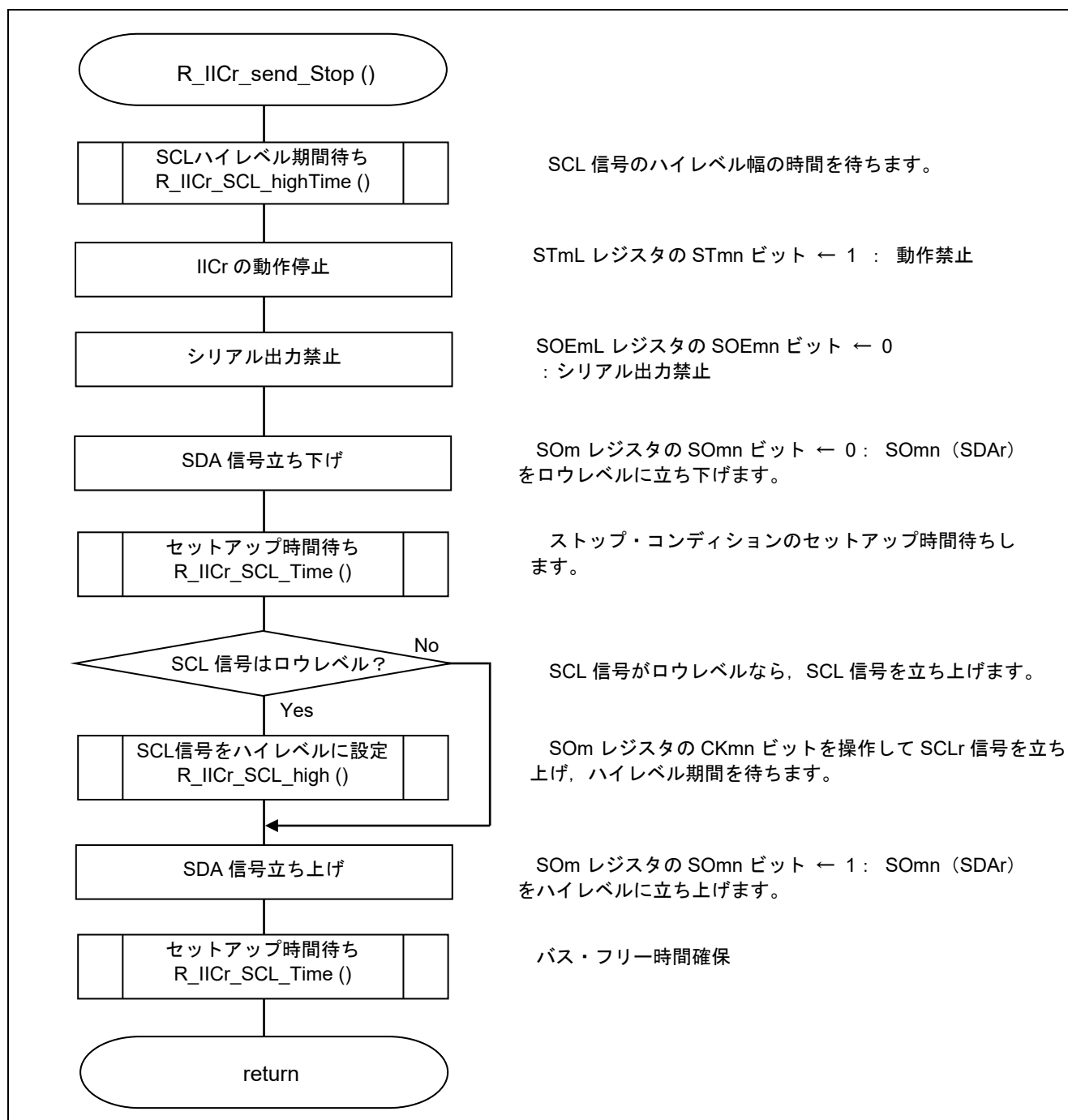


図 5.12 ストップ・コンディション発行関数

5.7.5 バス開放処理関数

図 5.13 にバス開放処理関数のフローチャートを示します。

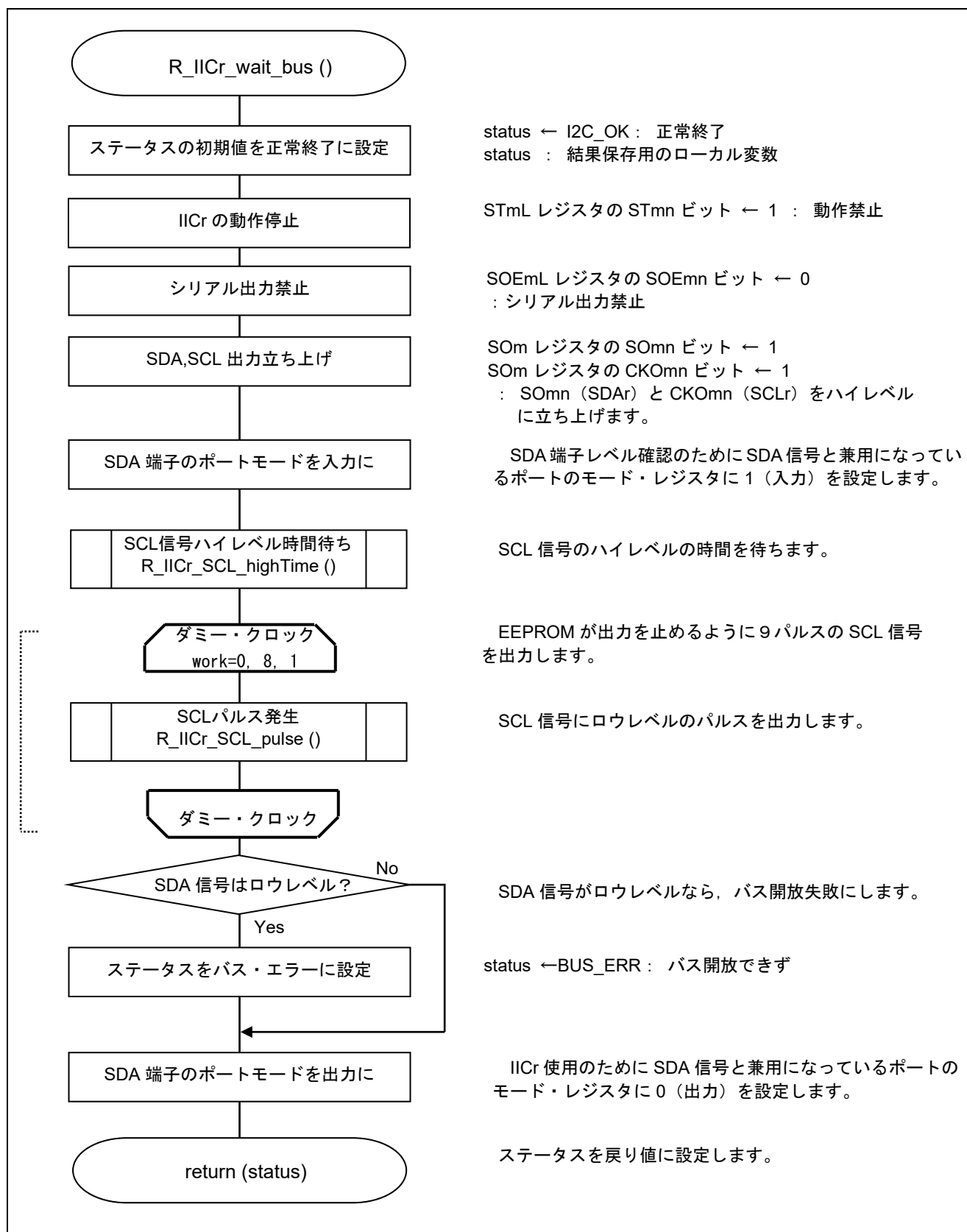


図 5.13 バス開放処理関数

5.7.6 EEPROM 書き込み処理

図 5.14 に EEPROM 書き込み処理の状態遷移図、図 5.15 にフローチャートを示します。

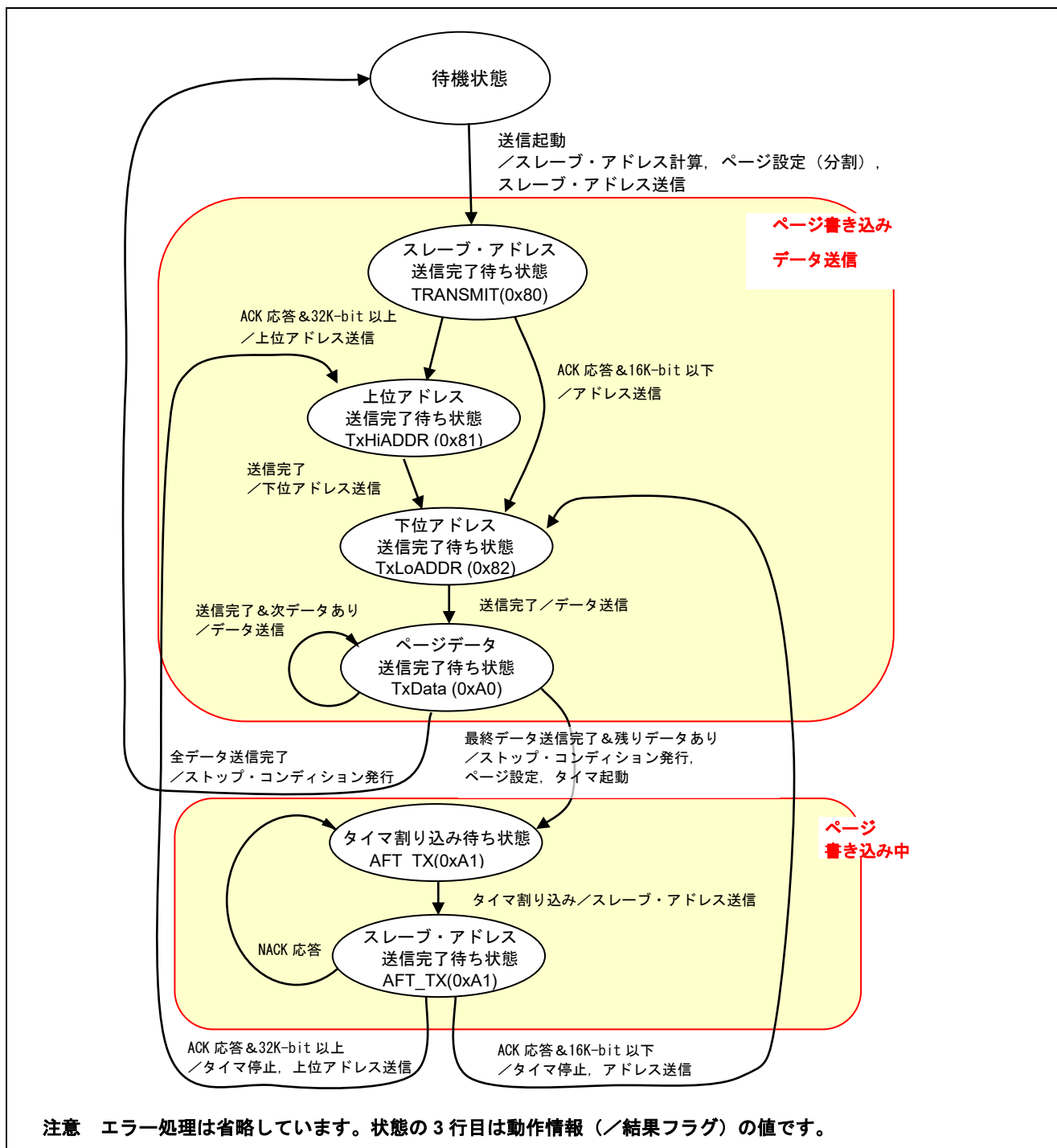


図 5.14 EEPROM 書き込み処理の状態遷移図

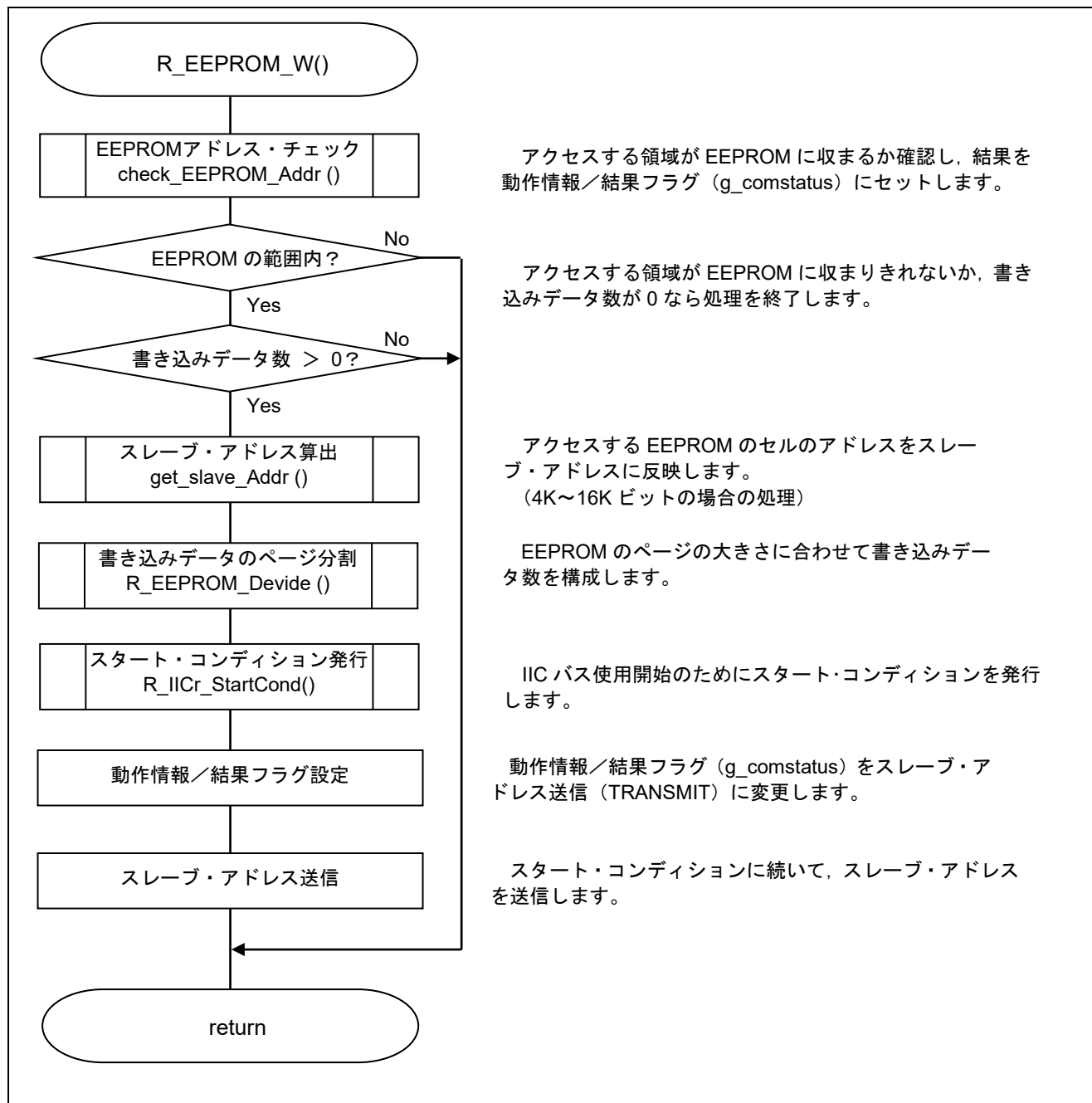


図 5.15 EEPROM 書き込み処理

5.7.7 EEPROM アドレス・チェック処理

図 5.16 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。

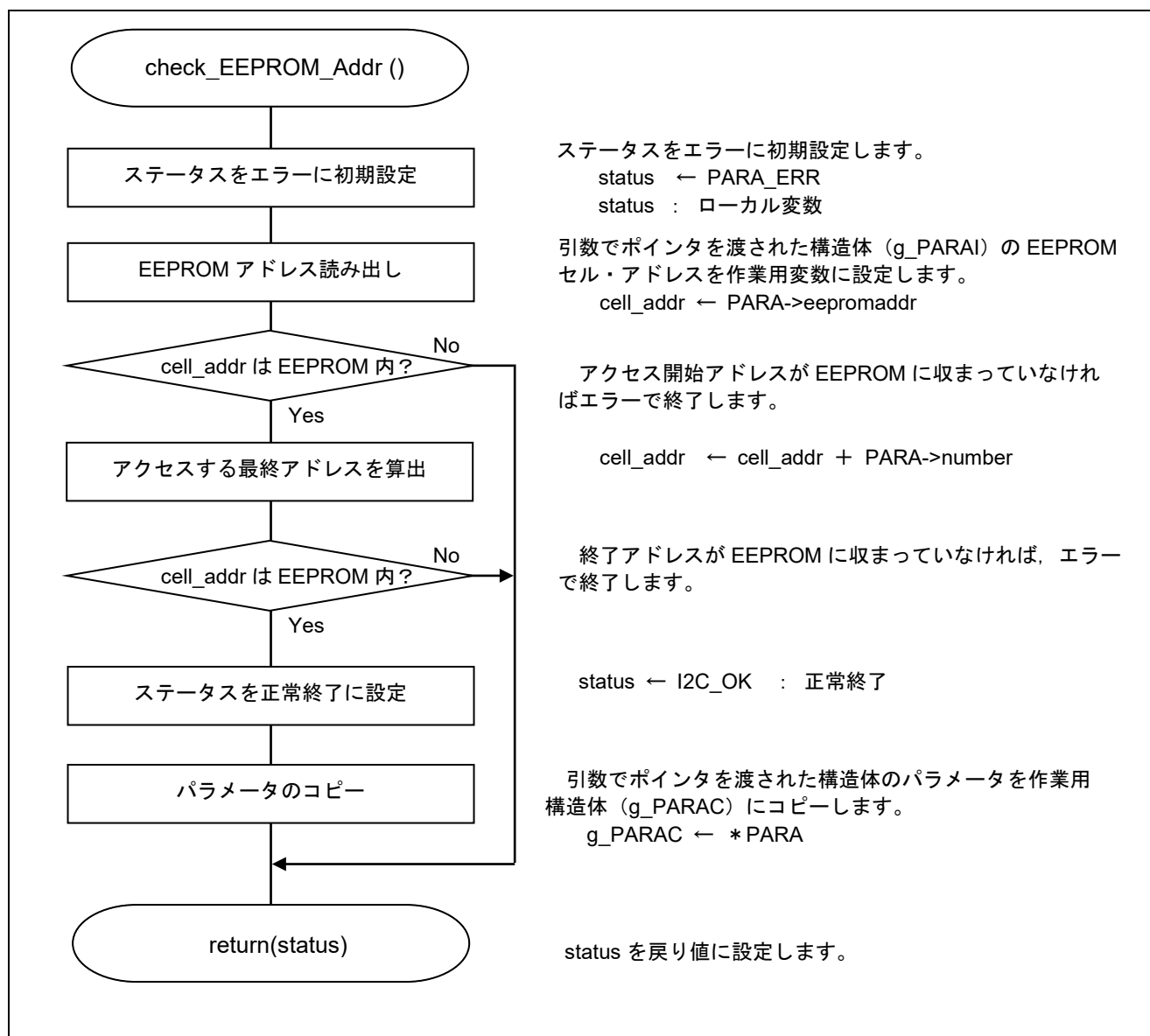


図 5.16 EEPROM アドレス・チェック処理

5.7.8 スレーブ・アドレス算出処理

図 5.17 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。

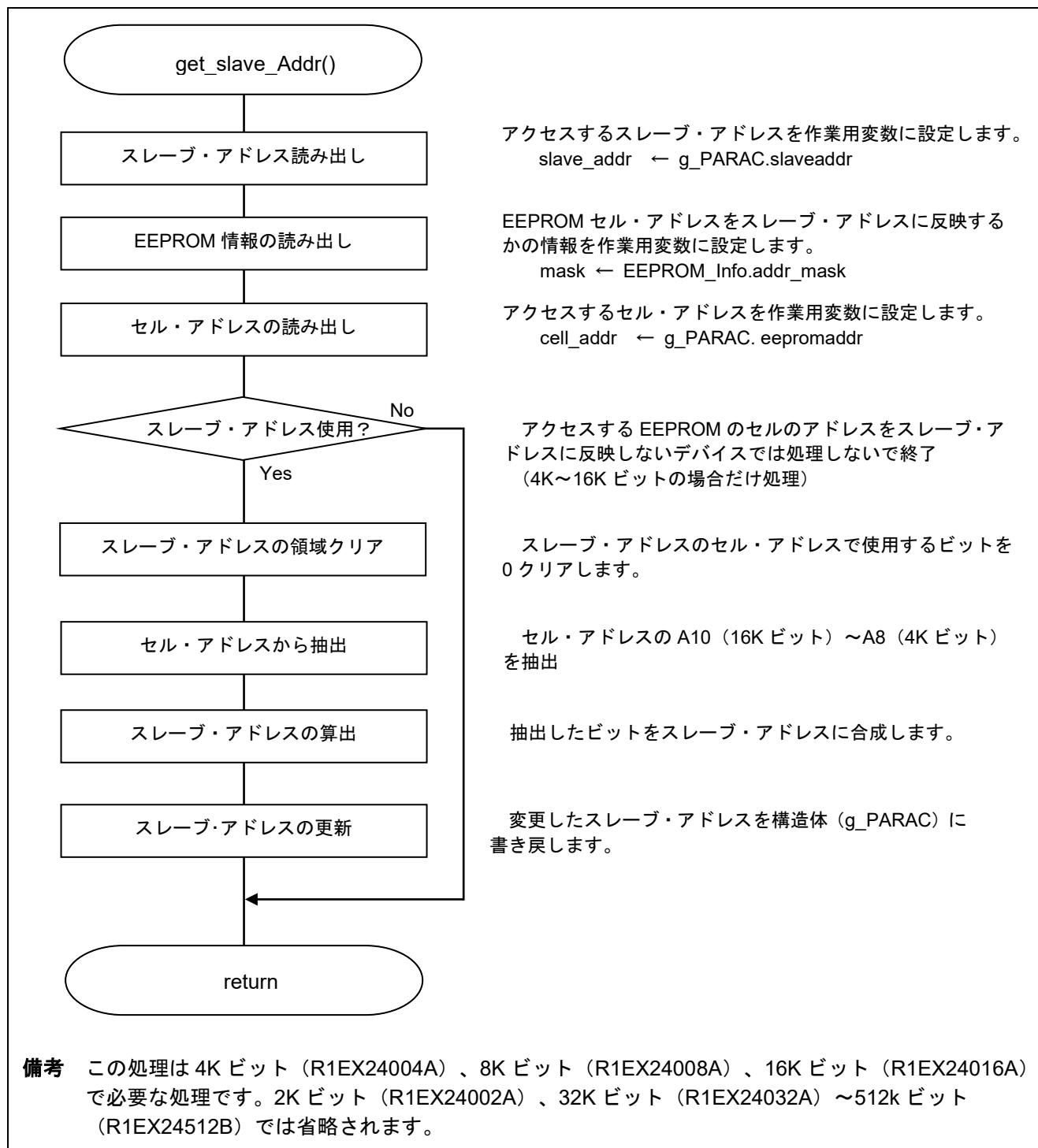


図 5.17 スレーブ・アドレス算出処理

5.7.9 書き込みデータのページ分割処理

図 5.18 に書き込みデータのページ分割処理のフローチャートを示します。

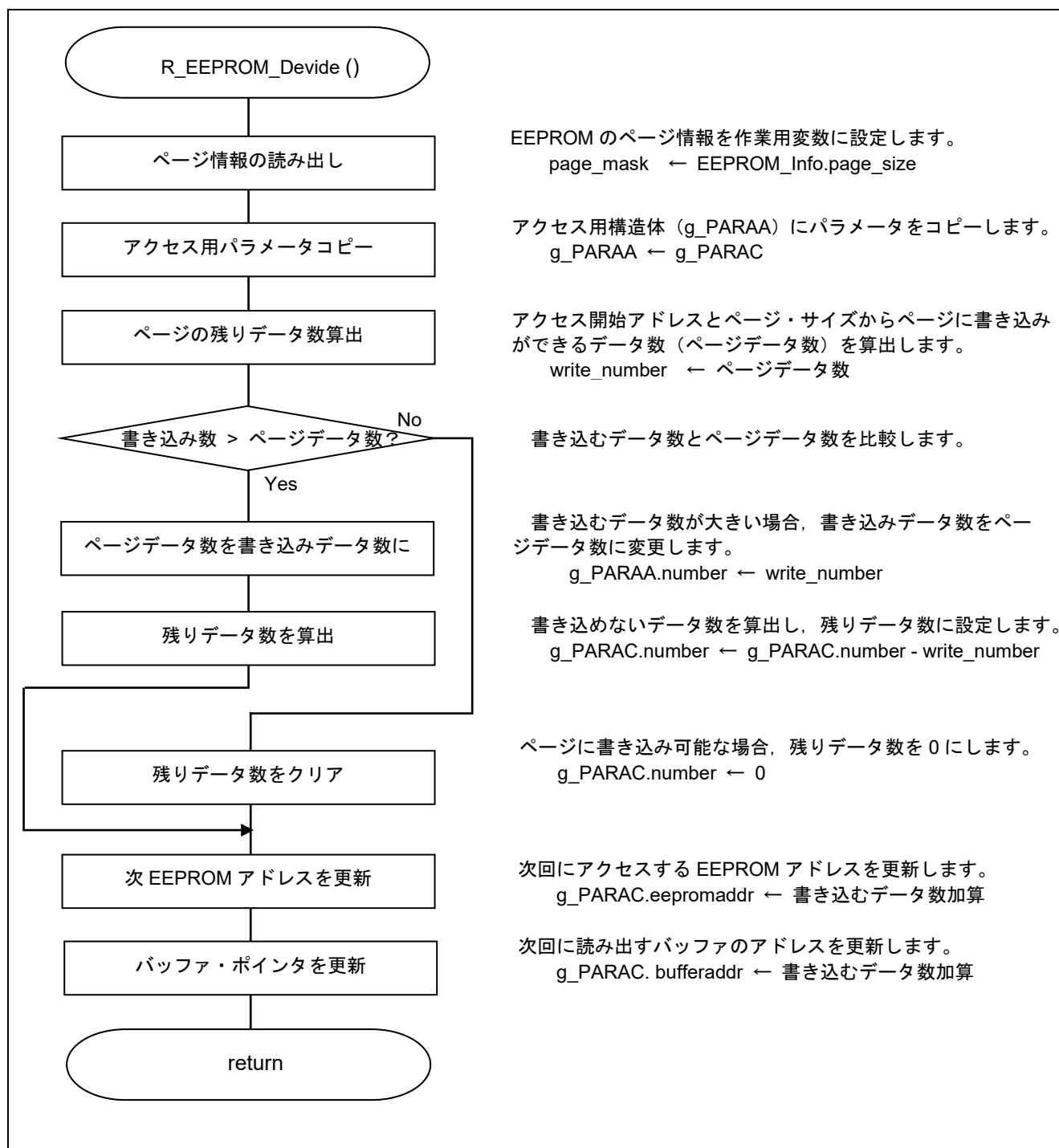


図 5.18 書き込みデータのページ分割処理

5.7.10 スタート・コンディション発行処理

図 5.19 にスタート・コンディション発行処理のフローチャートを示します。

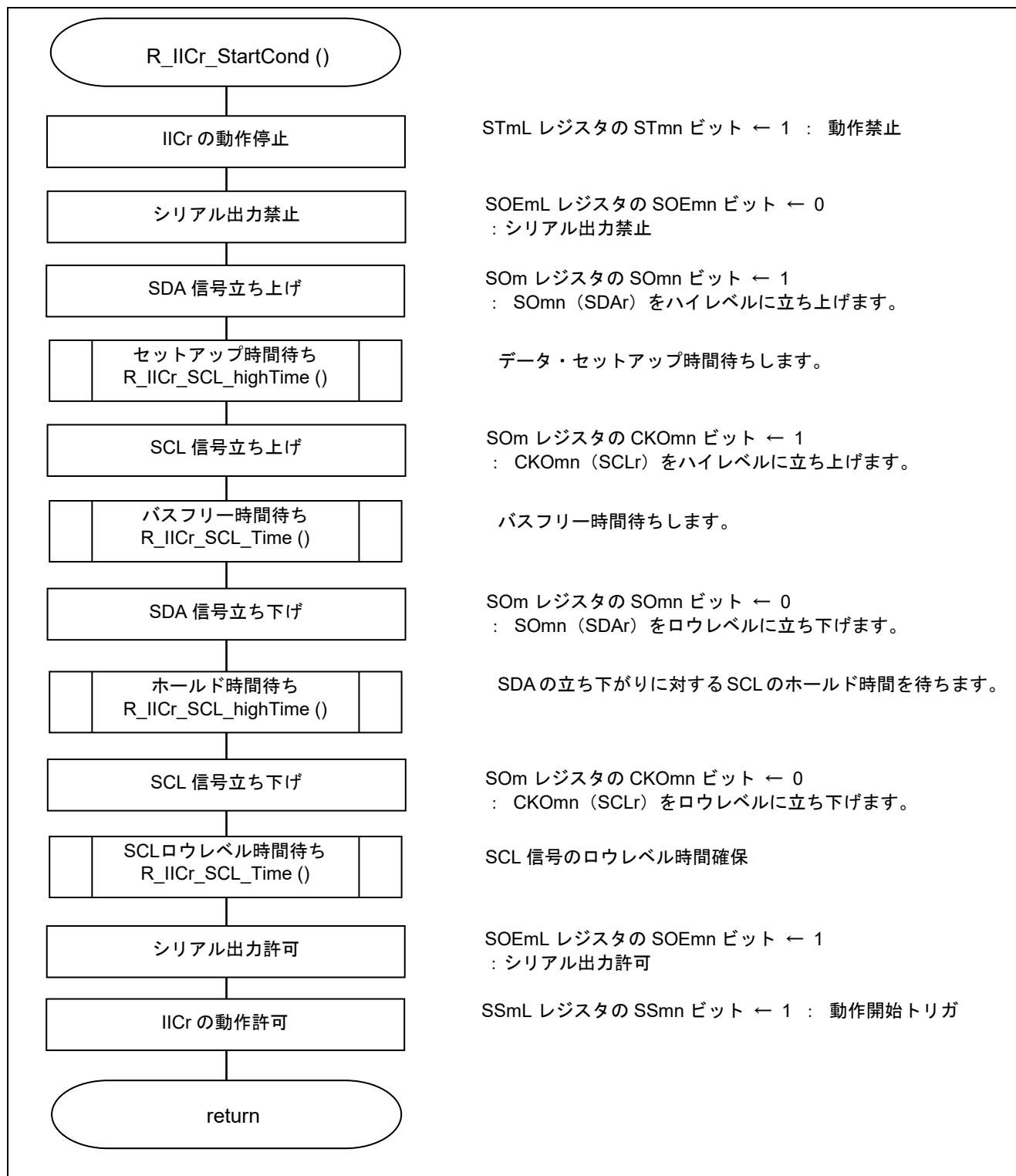


図 5.19 スタート・コンディション発行処理

5.7.11 EEPROM 書き込み完了待ち処理

図 5.20 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。

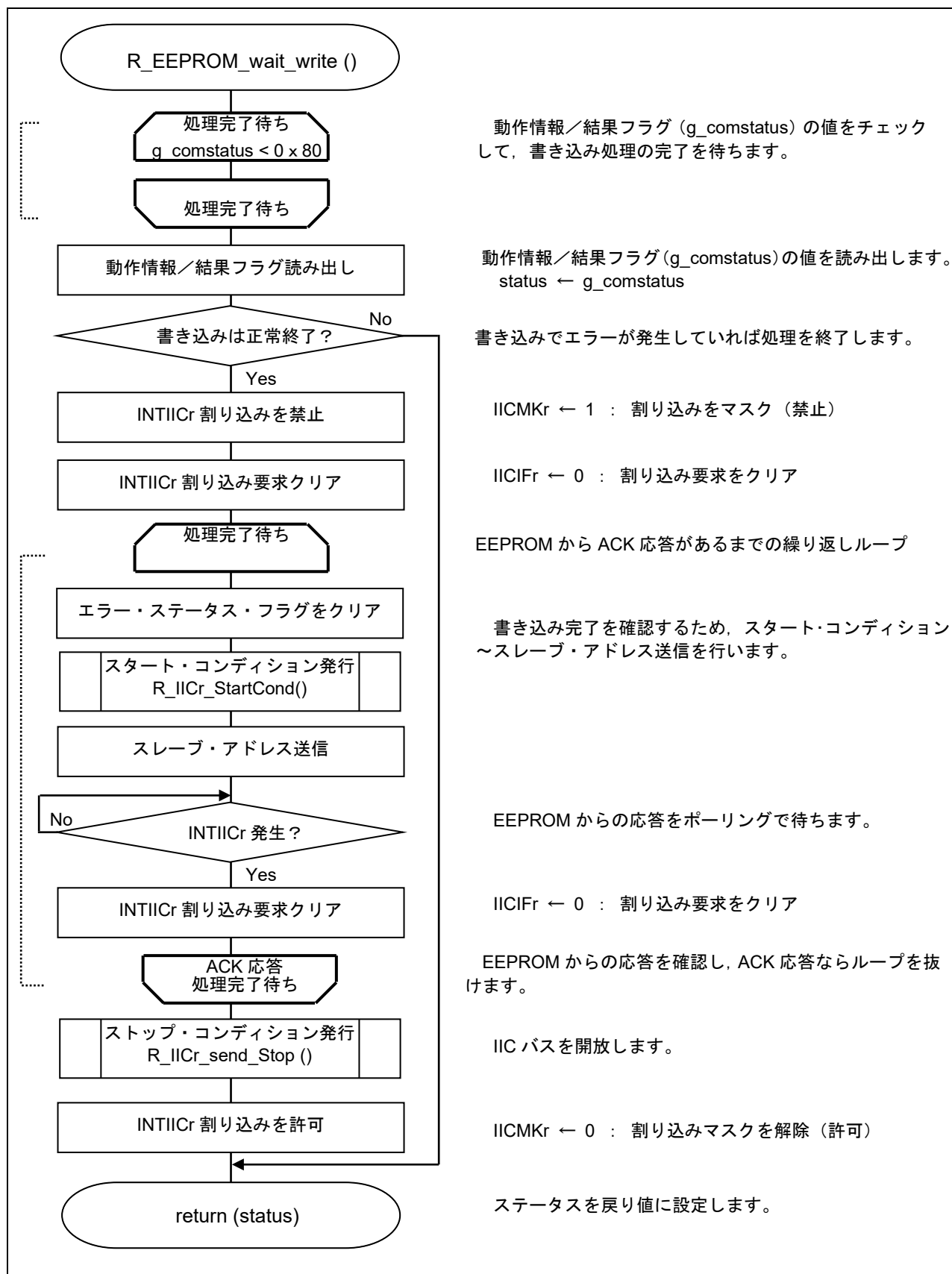


図 5.20 EEPROM 書き込み完了待ち処理

5.7.12 EEPROM 読み出し処理

図 5.21 に EEPROM 読み出し処理の状態遷移図、図 5.22 にフローチャートを示します。

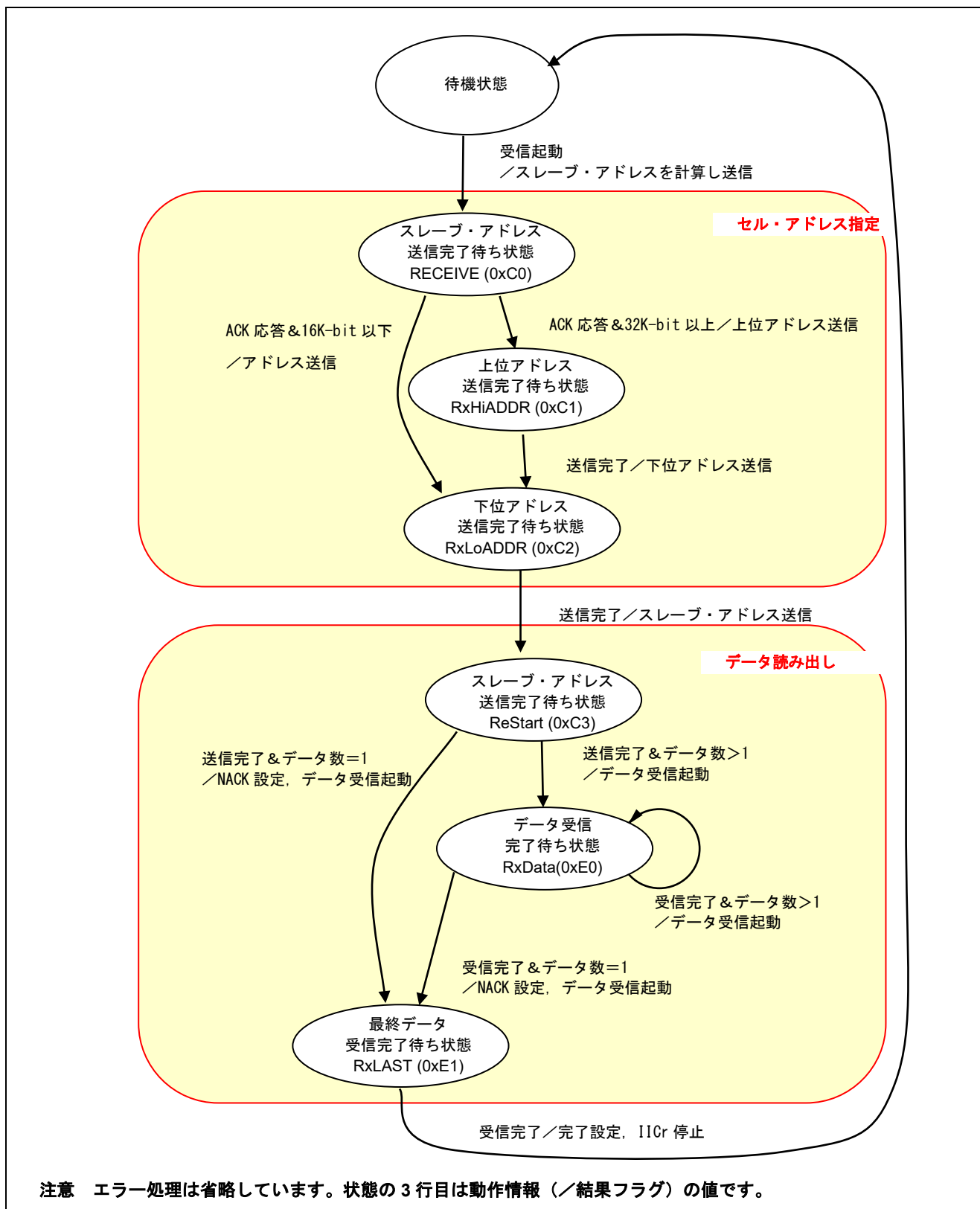


図 5.21 EEPROM 読み出し処理状態遷移図

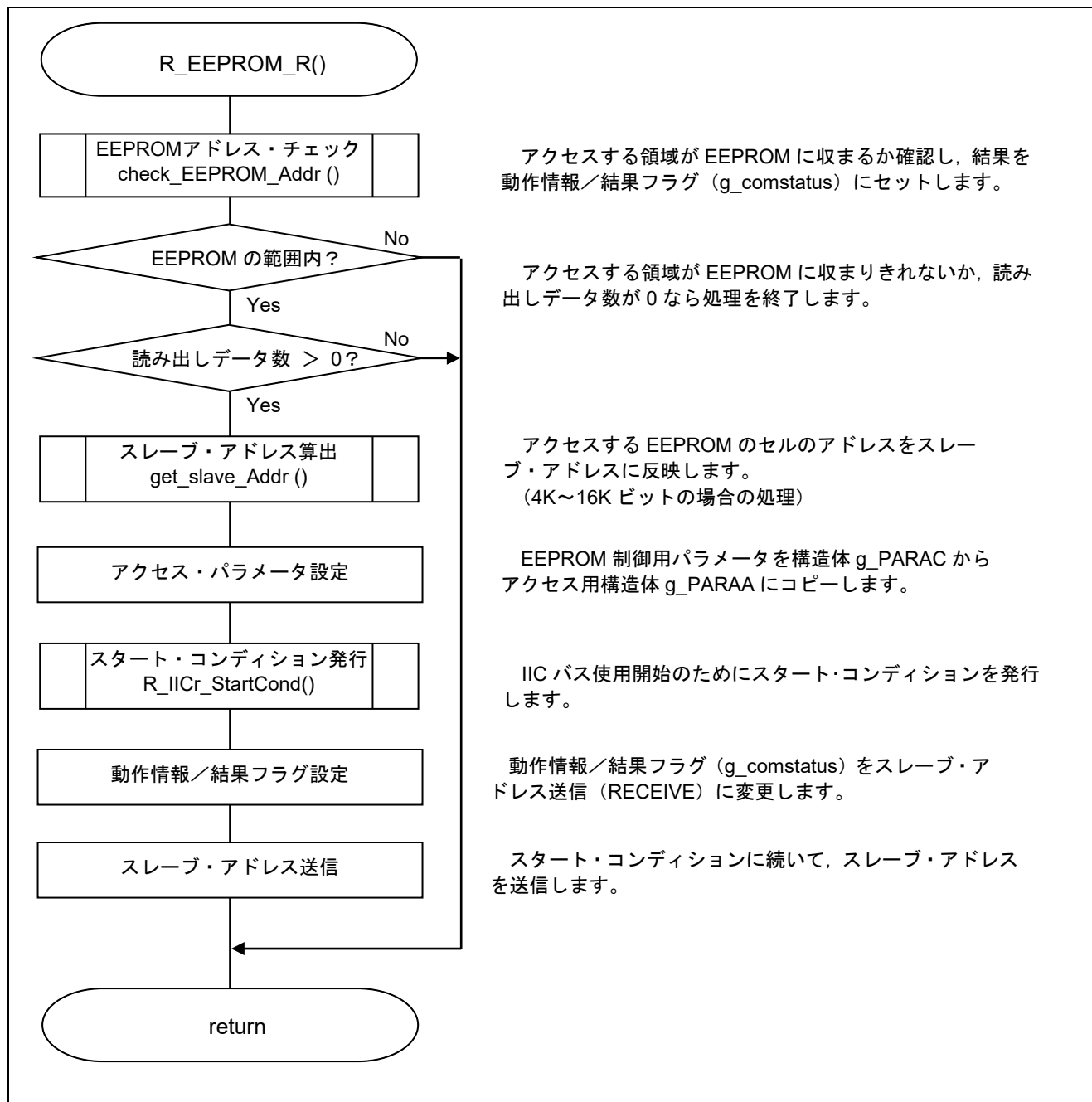


図 5.22 EEPROM 読み出し処理

5.7.13 EEPROM 読み出し完了待ち処理

図 5.23 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。

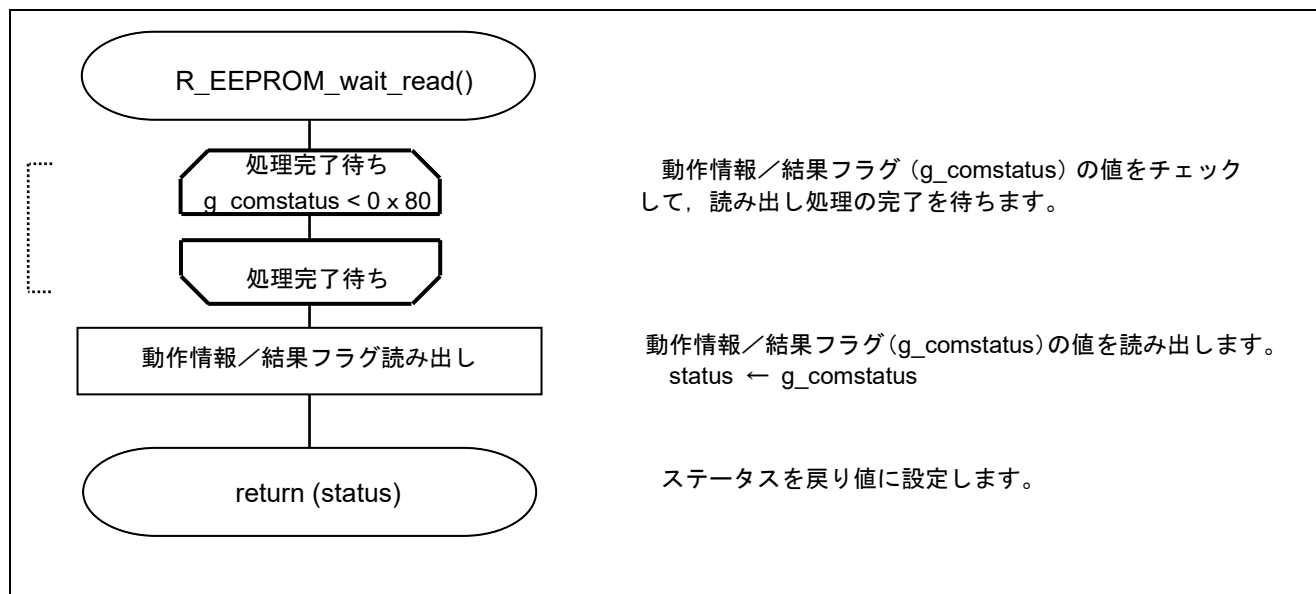


図 5.23 EEPROM 読み出し完了待ち処理

5.7.14 スレーブ・アドレス送信完了処理

図 5.24 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。

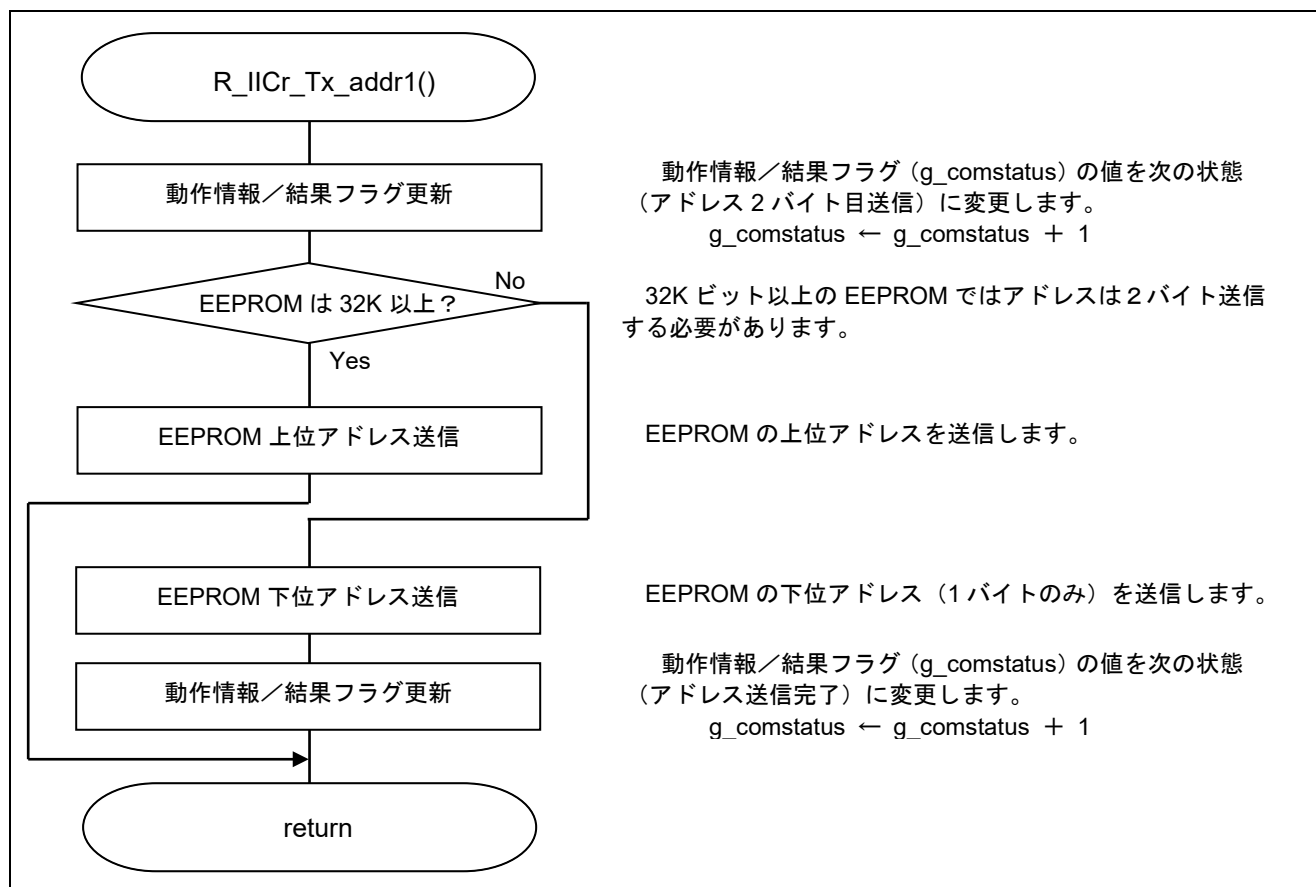


図 5.24 スレーブ・アドレス送信完了処理

5.7.15 上位アドレス送信完了処理

図 5.25 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。

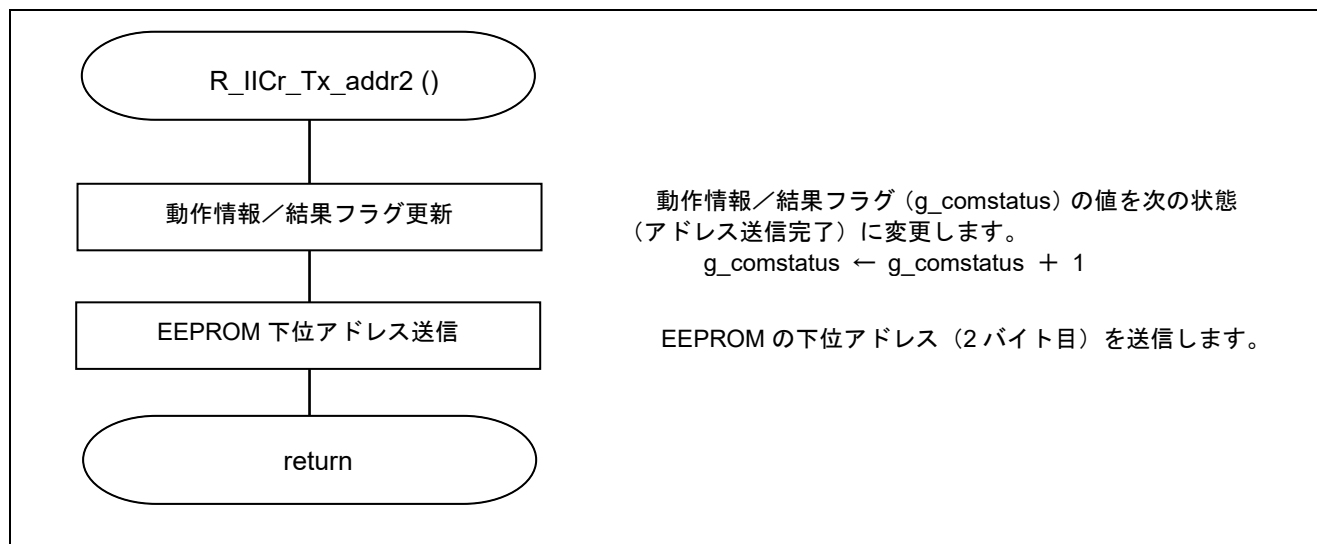


図 5.25 上位アドレス送信完了処理

5.7.16 リスタート処理

図 5.26 にリスタート処理のフローチャートを示します。

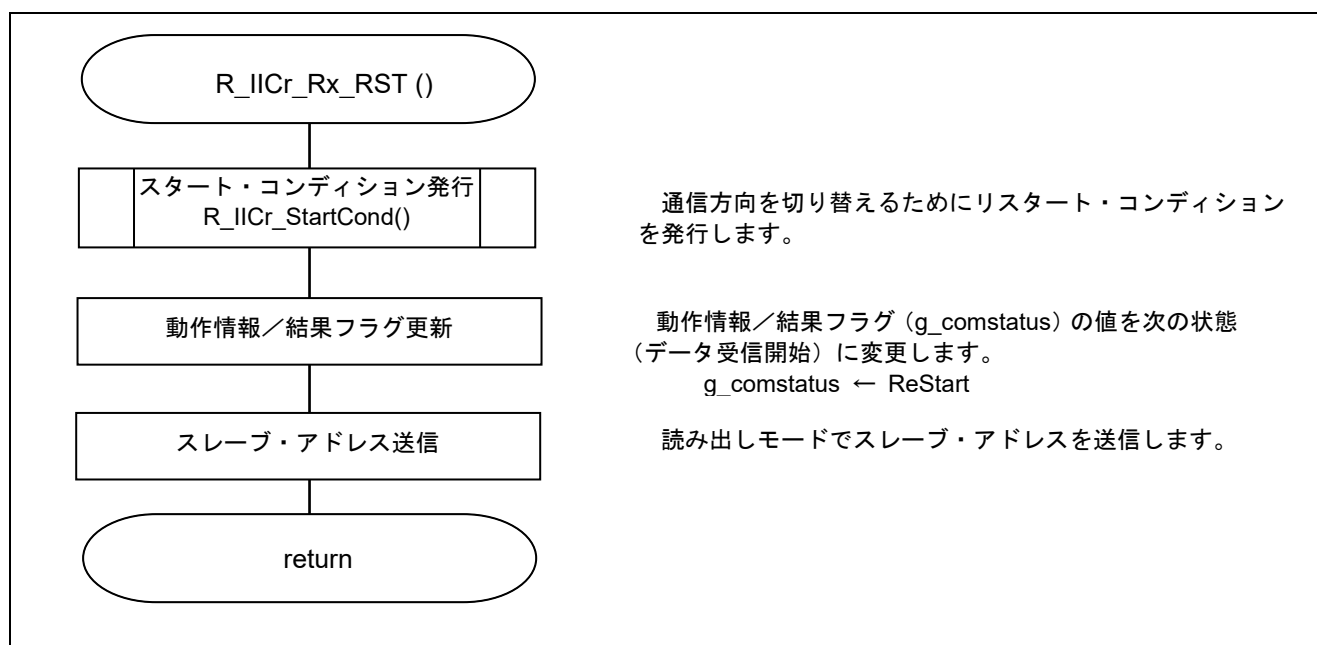


図 5.26 リスタート処理

5.7.17 データ受信開始処理

図 5.27 にデータ受信開始処理のフローチャートを示します。

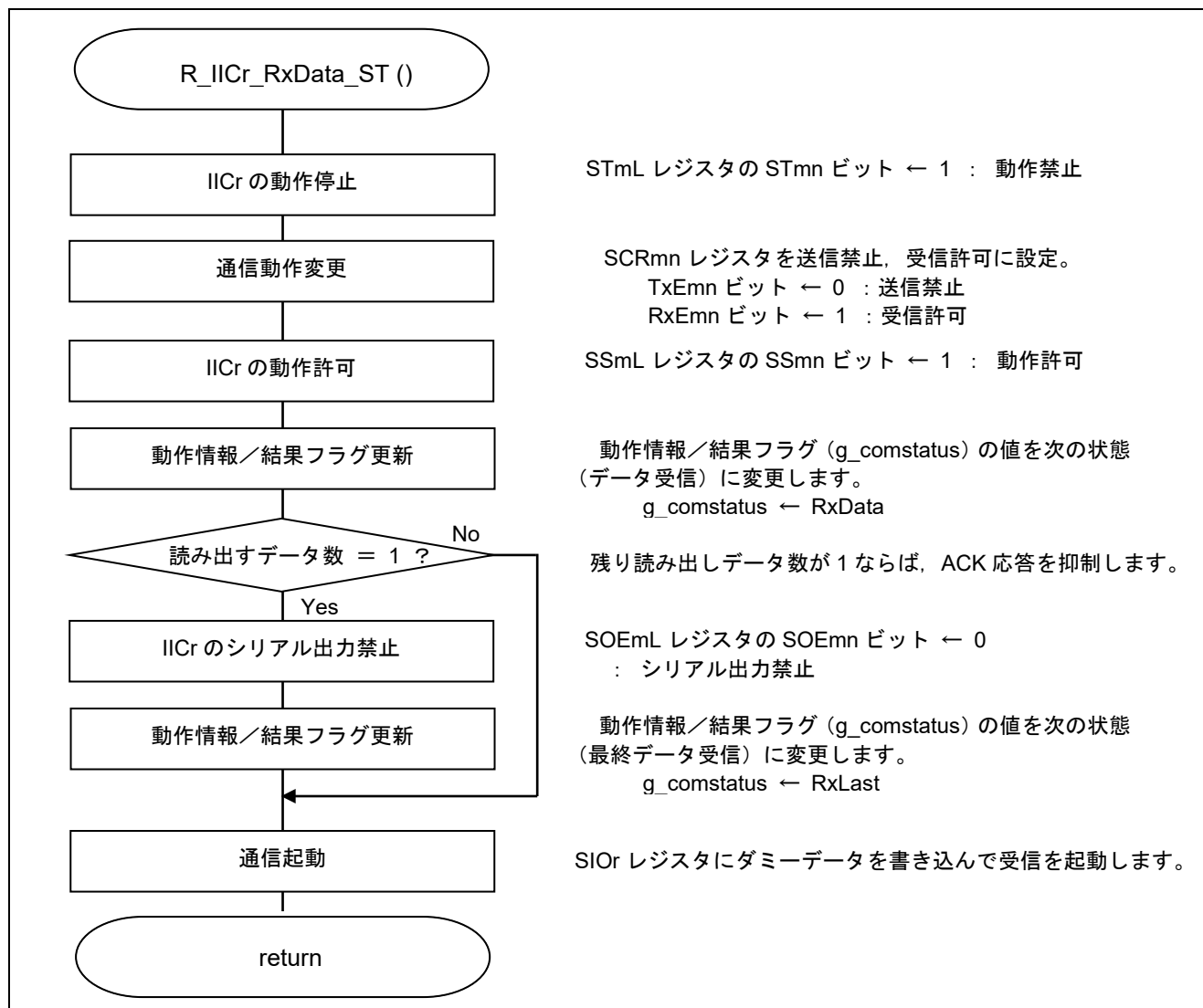


図 5.27 データ受信開始処理

5.7.18 データ受信処理

図 5.28 にデータ受信処理のフローチャートを示します。

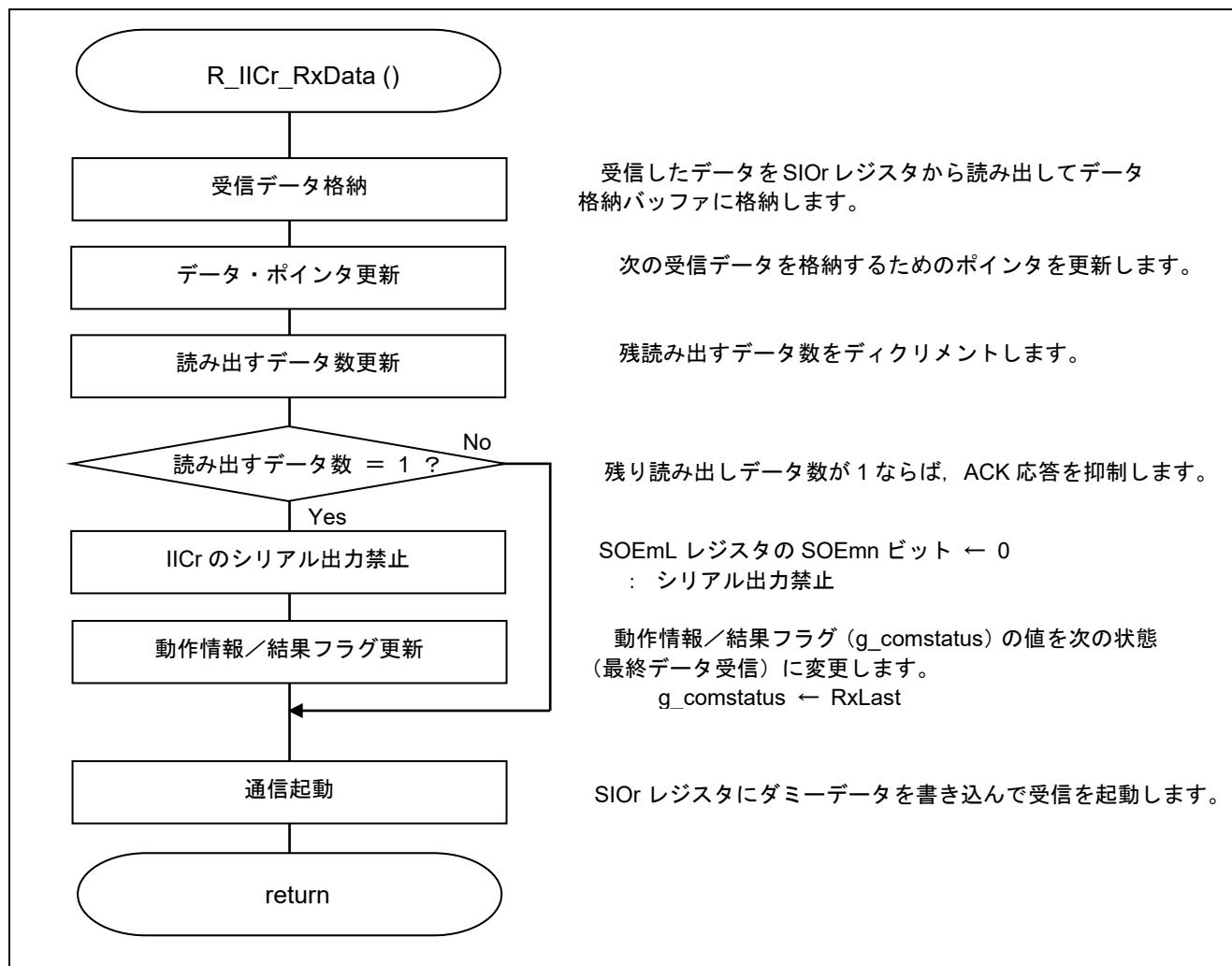


図 5.28 データ受信処理

5.7.19 最終データ受信処理

図 5.29 に最終データ受信処理のフローチャートを示します。

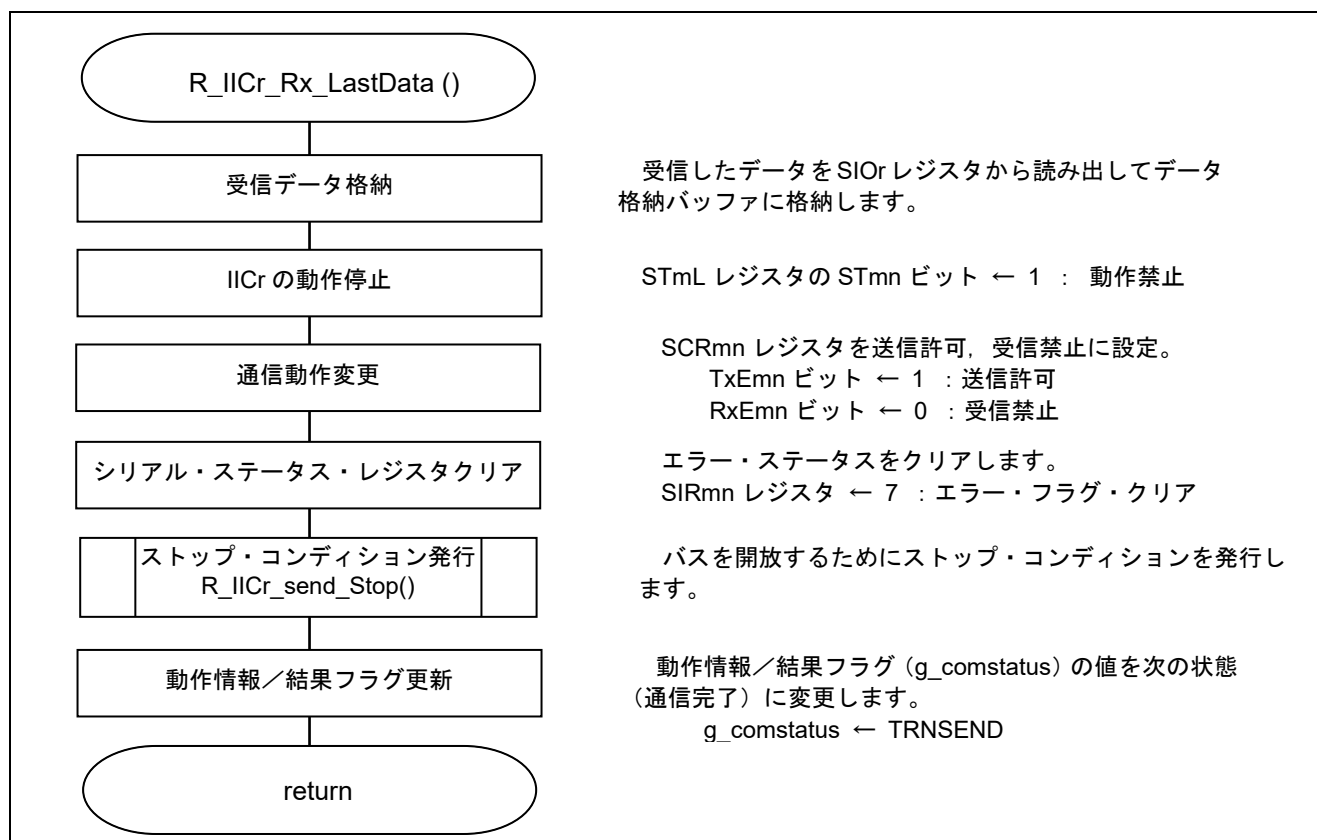


図 5.29 最終データ受信処理

5.7.20 データ送信開始処理

図 5.30 にデータ送信開始処理のフローチャートを示します。

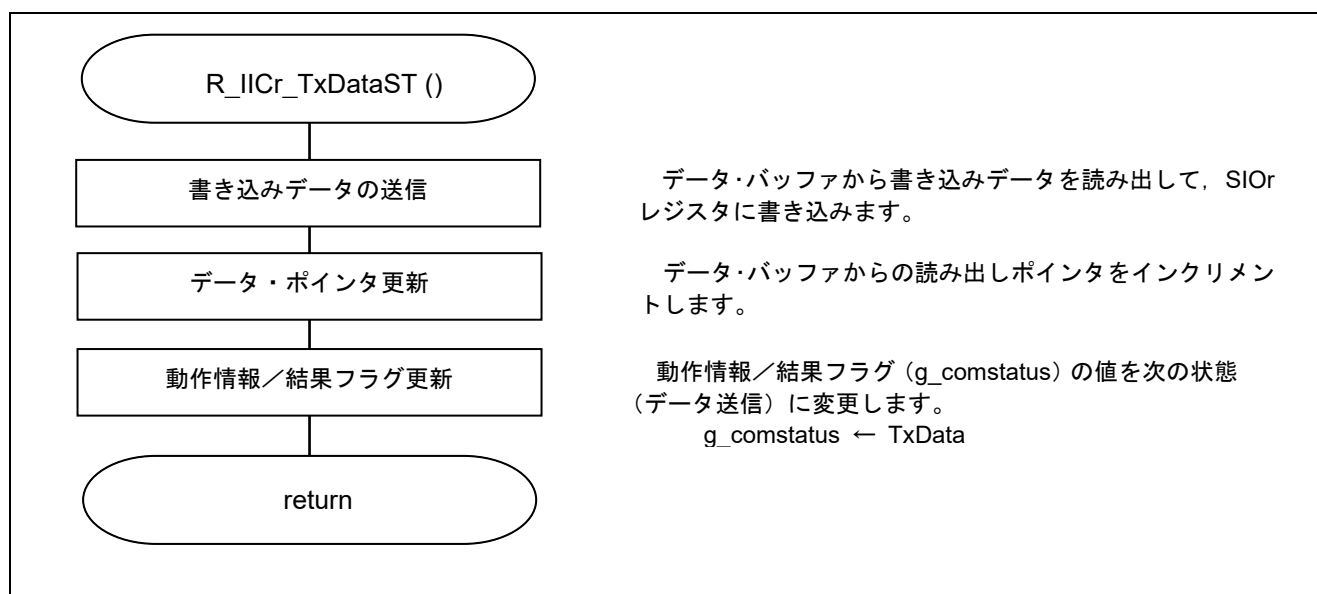


図 5.30 データ送信開始処理

5.7.21 データ送信処理

図 5.31 にデータ送信処理のフローチャートを示します。

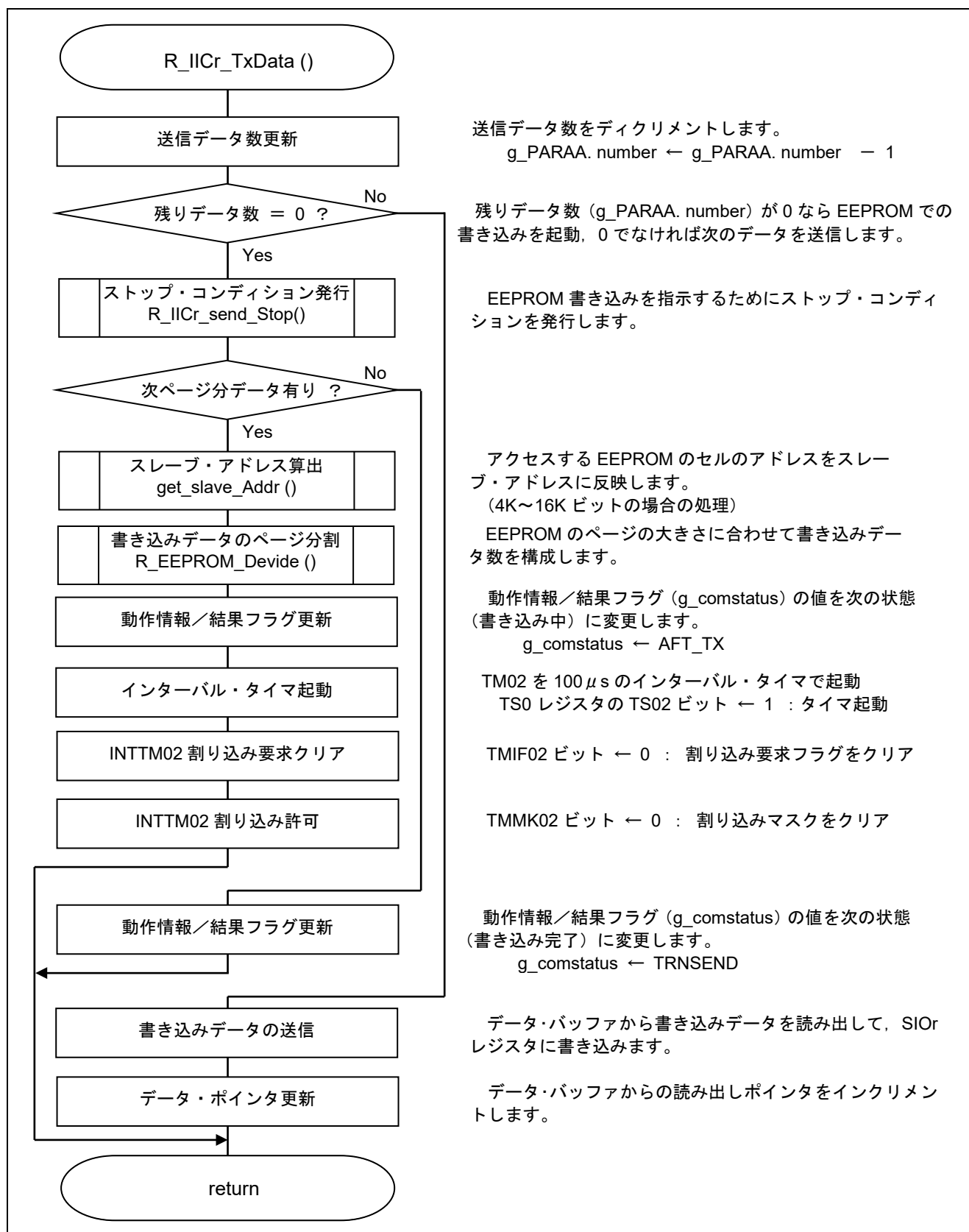


図 5.31 データ送信処理

5.7.22 次ページ書き込み開始処理

図 5.32 に次ページ書き込み開始処理のフローチャートを示します。

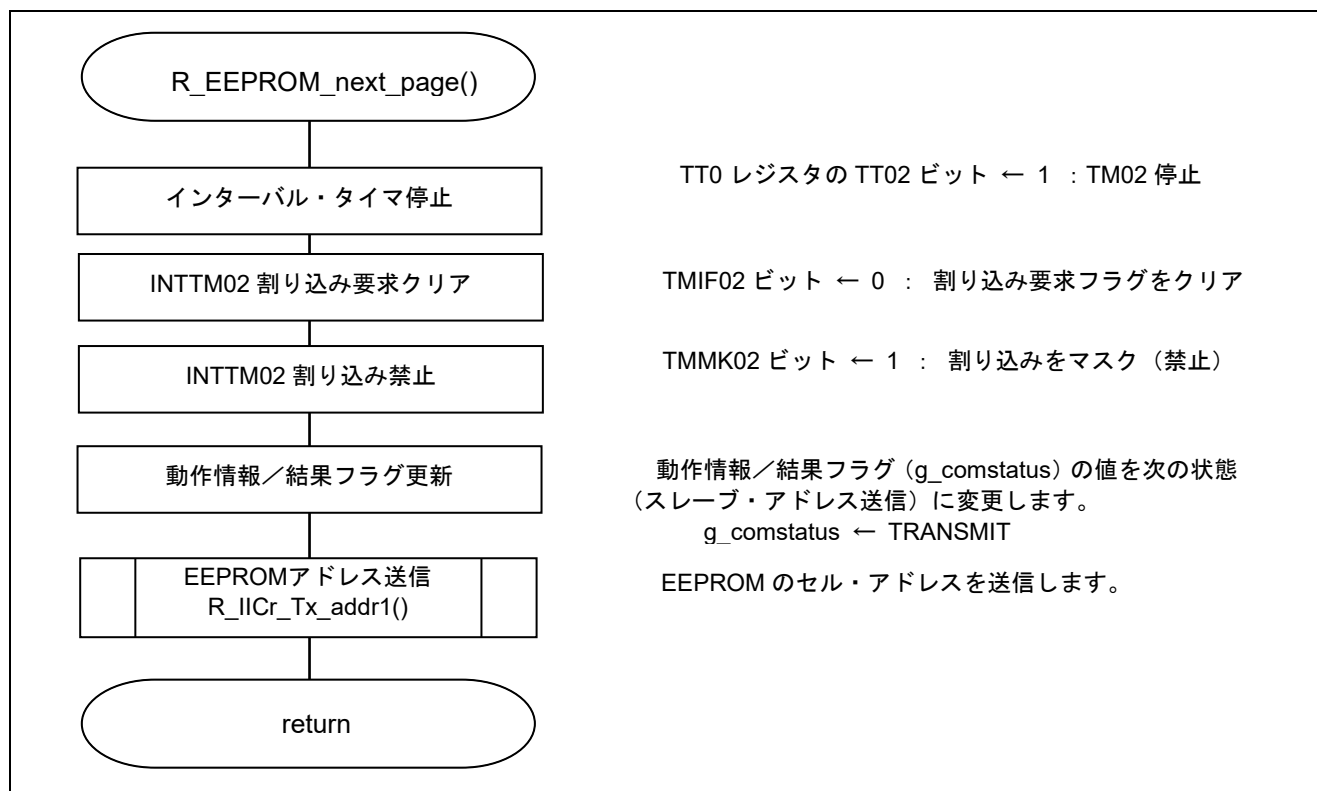


図 5.32 次ページ書き込み開始処理

5.7.23 SCL ダミー・クロック出力処理

図 5.33 に SCL ダミー・クロック出力処理のフローチャートを示します。

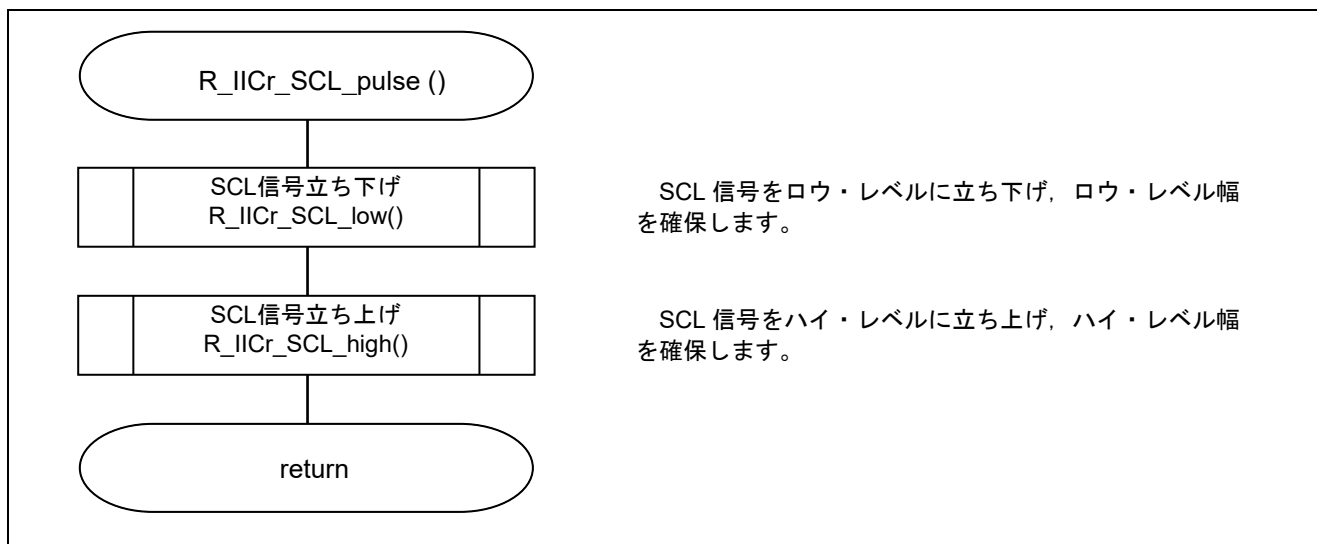


図 5.33 SCL ダミー・クロック出力処理

5.7.24 SCL 立ち上げ処理

図 5.34 に SCL 立ち上げ処理のフローチャートを示します。

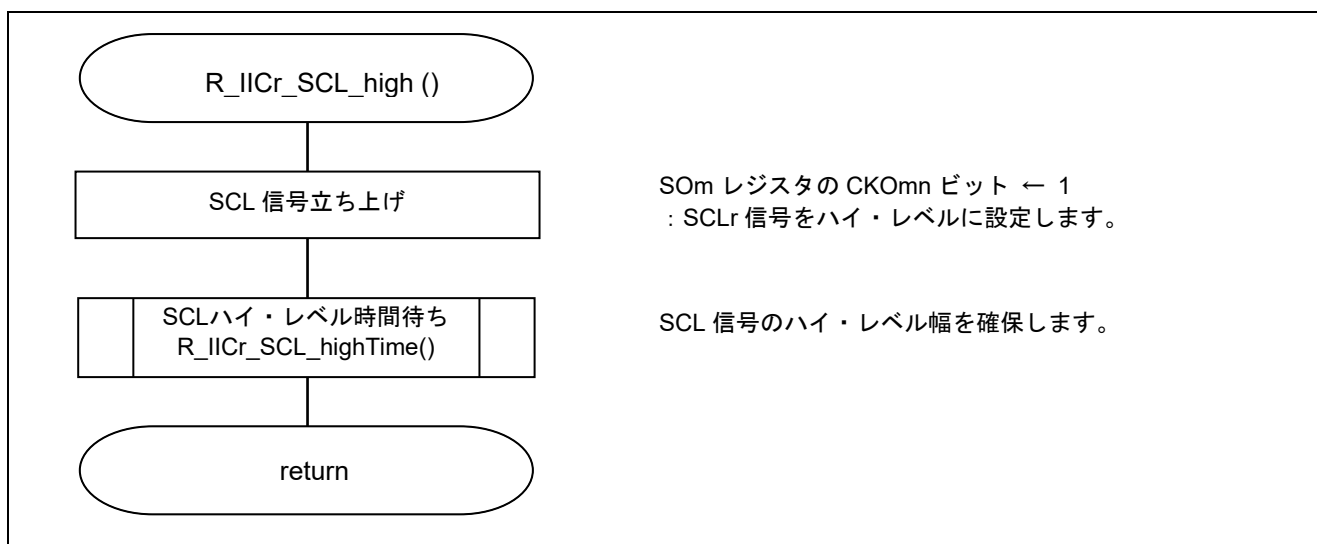


図 5.34 SCL 立ち上げ処理

5.7.25 SCL 立ち下げ処理

図 5.35 に SCL 立ち下げ処理のフローチャートを示します。

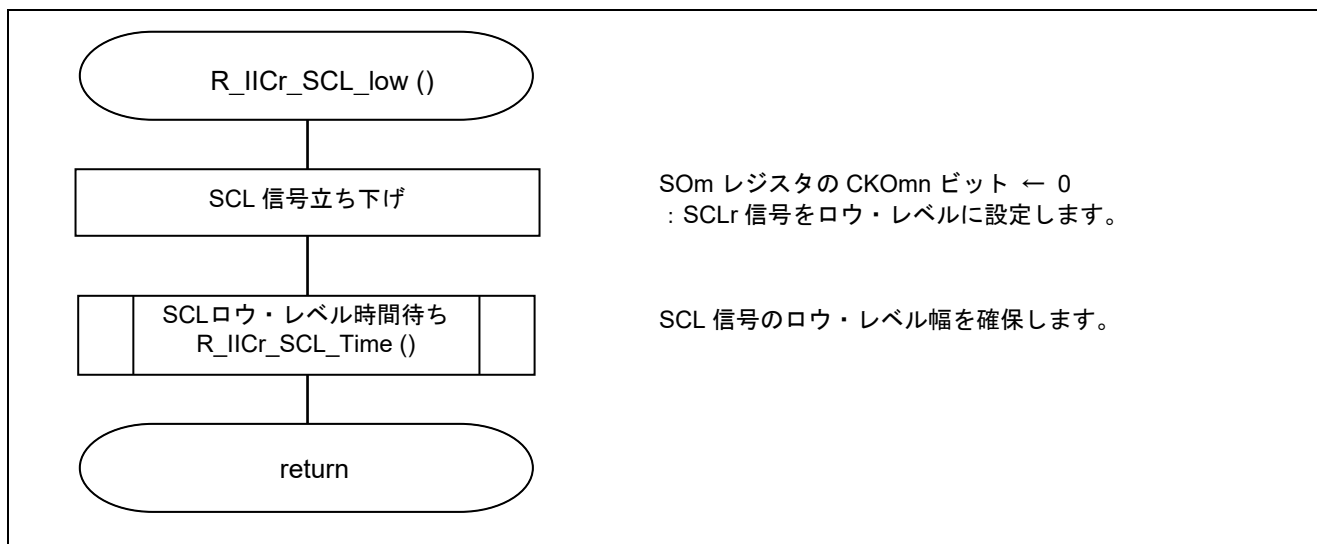


図 5.35 SCL 立ち下げ処理

5.7.26 ACK 確認処理

図 5.36 に ACK 確認処理のフローチャートを示します。

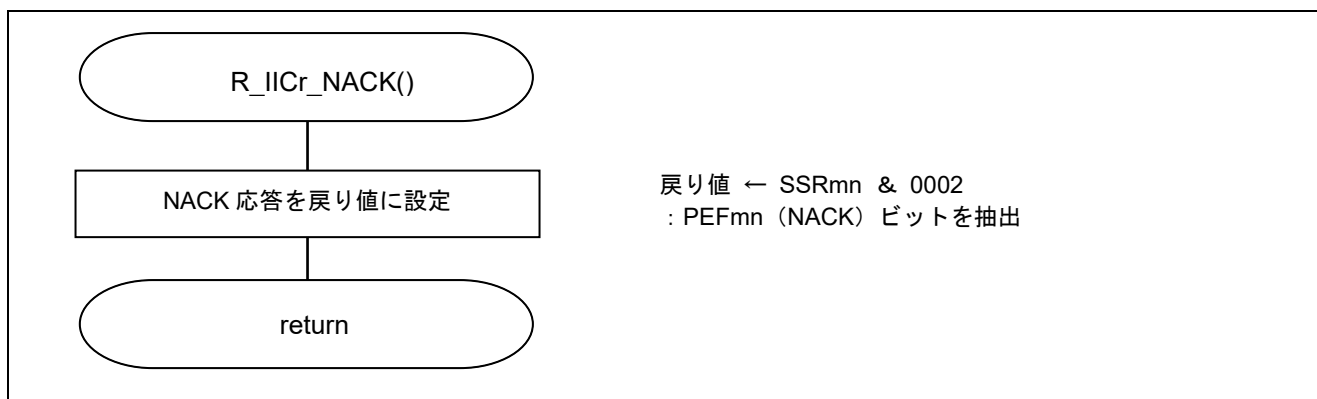


図 5.36 ACK 応答確認処理

5.7.27 SCL ロウ・レベル時間待ち処理

図 5.37 に SCL ロウ・レベル時間待ち処理のフローチャートを示します。

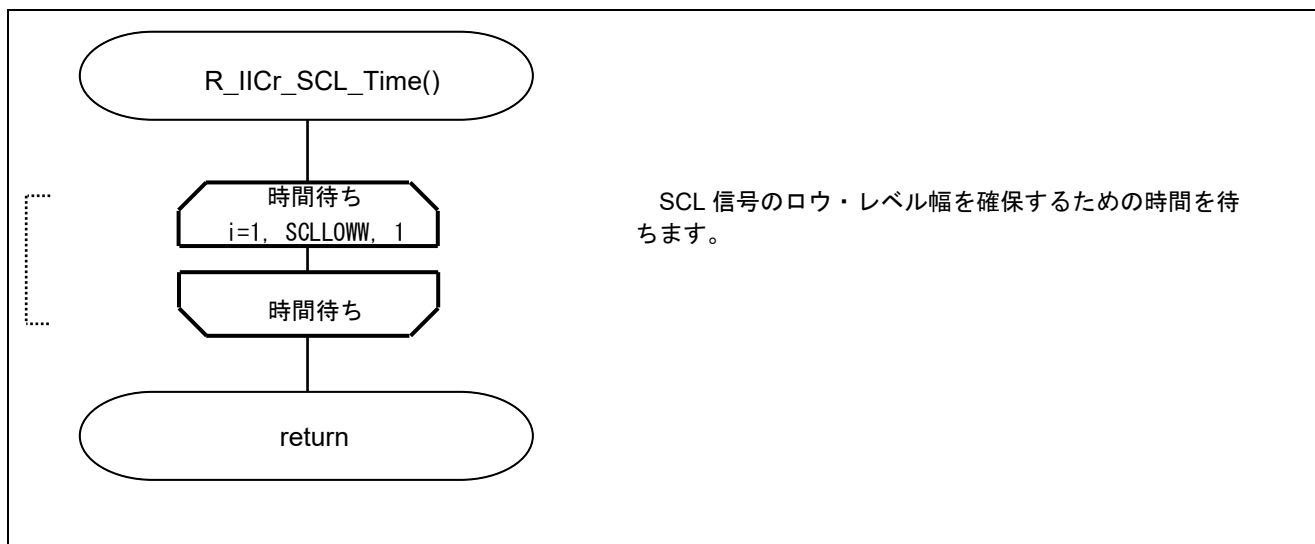


図 5.37 SCL ロウ・レベル時間待ち処理

5.7.28 SCL ハイ・レベル時間待ち処理

図 5.38 に SCL ハイ・レベル時間待ち処理のフローチャートを示します。

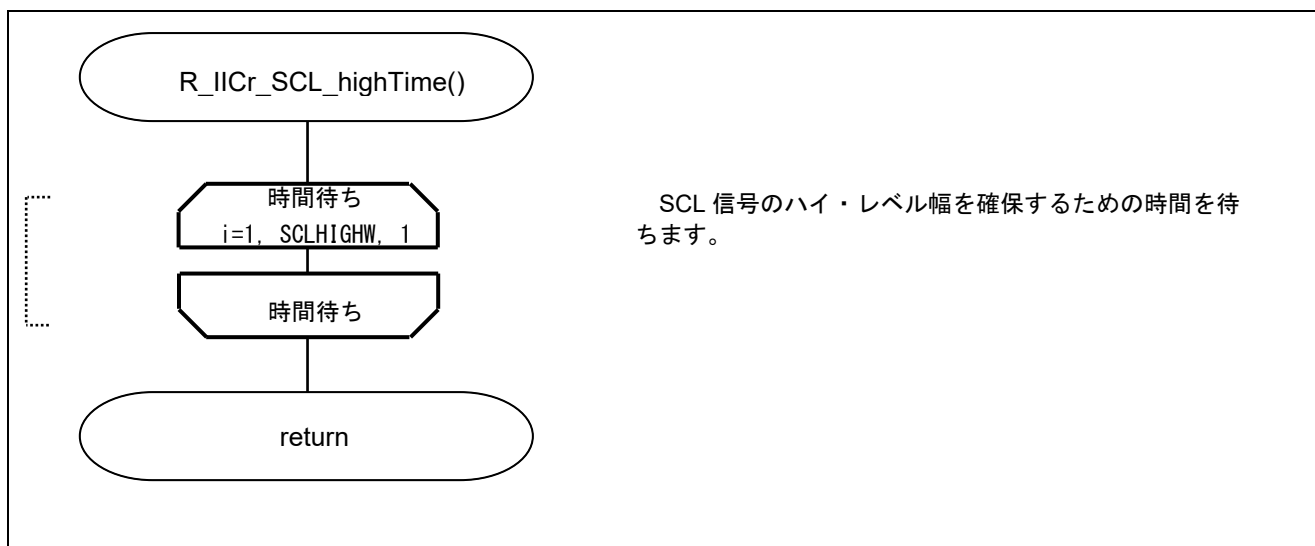


図 5.38 SCL ハイ・レベル時間待ち処理

5.7.29 INTIICr 割り込み処理

図 5.39～図 5.41 に INTIICr 割り込み処理のフローチャートを示します。

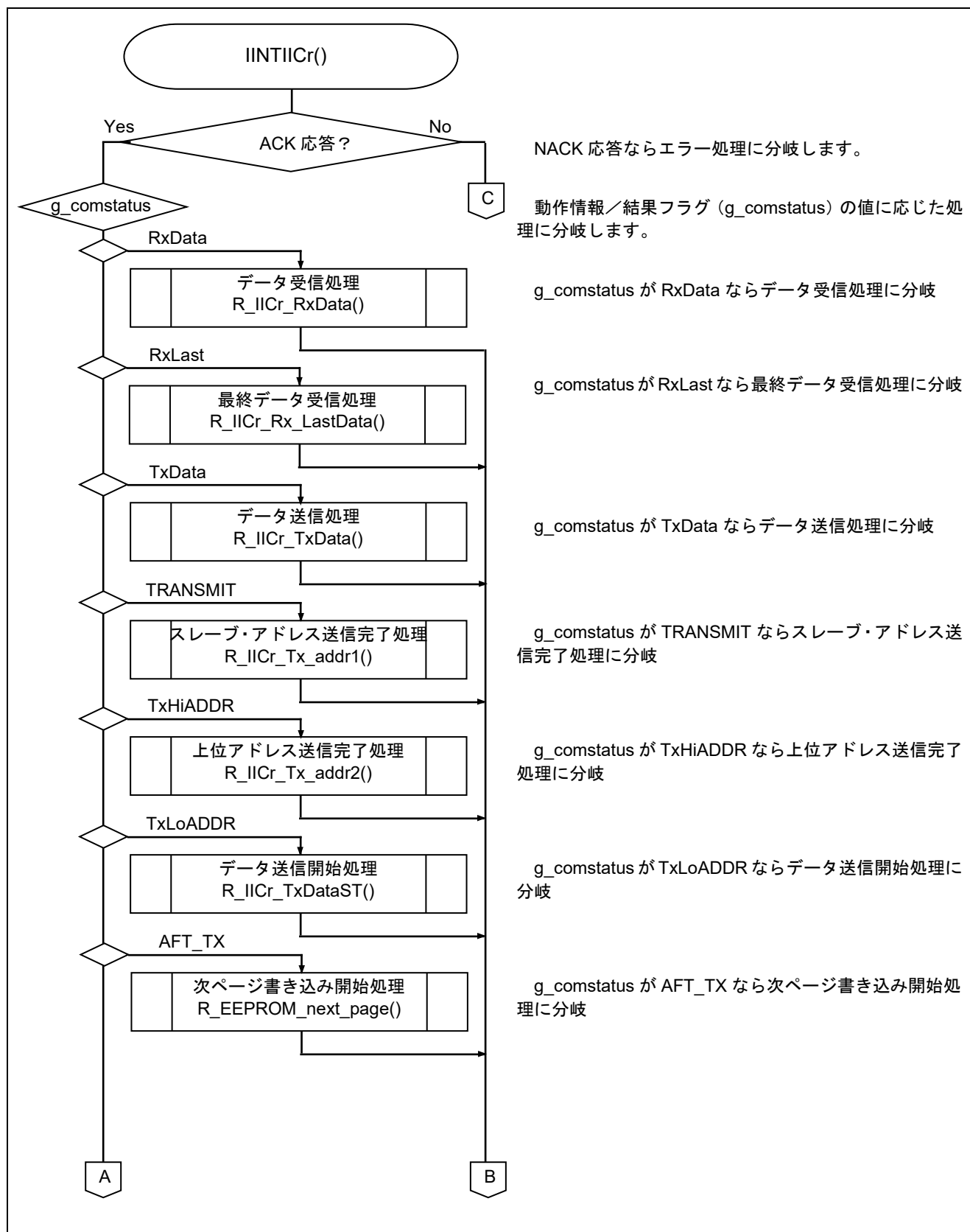


図 5.39 INTIICr 割り込み処理 (1/3)

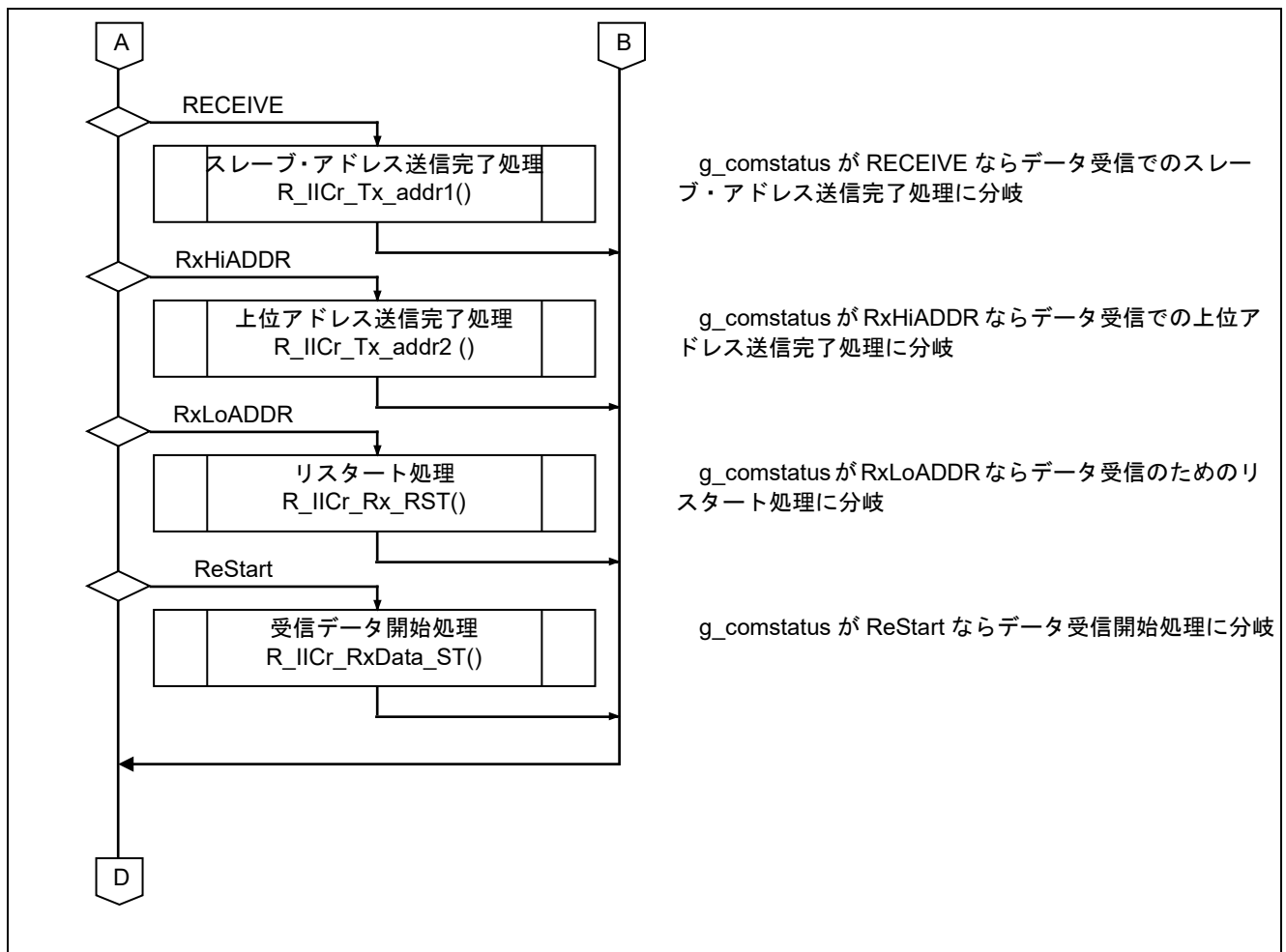


図 5.40 INTIICr 割り込み処理 (2/3)

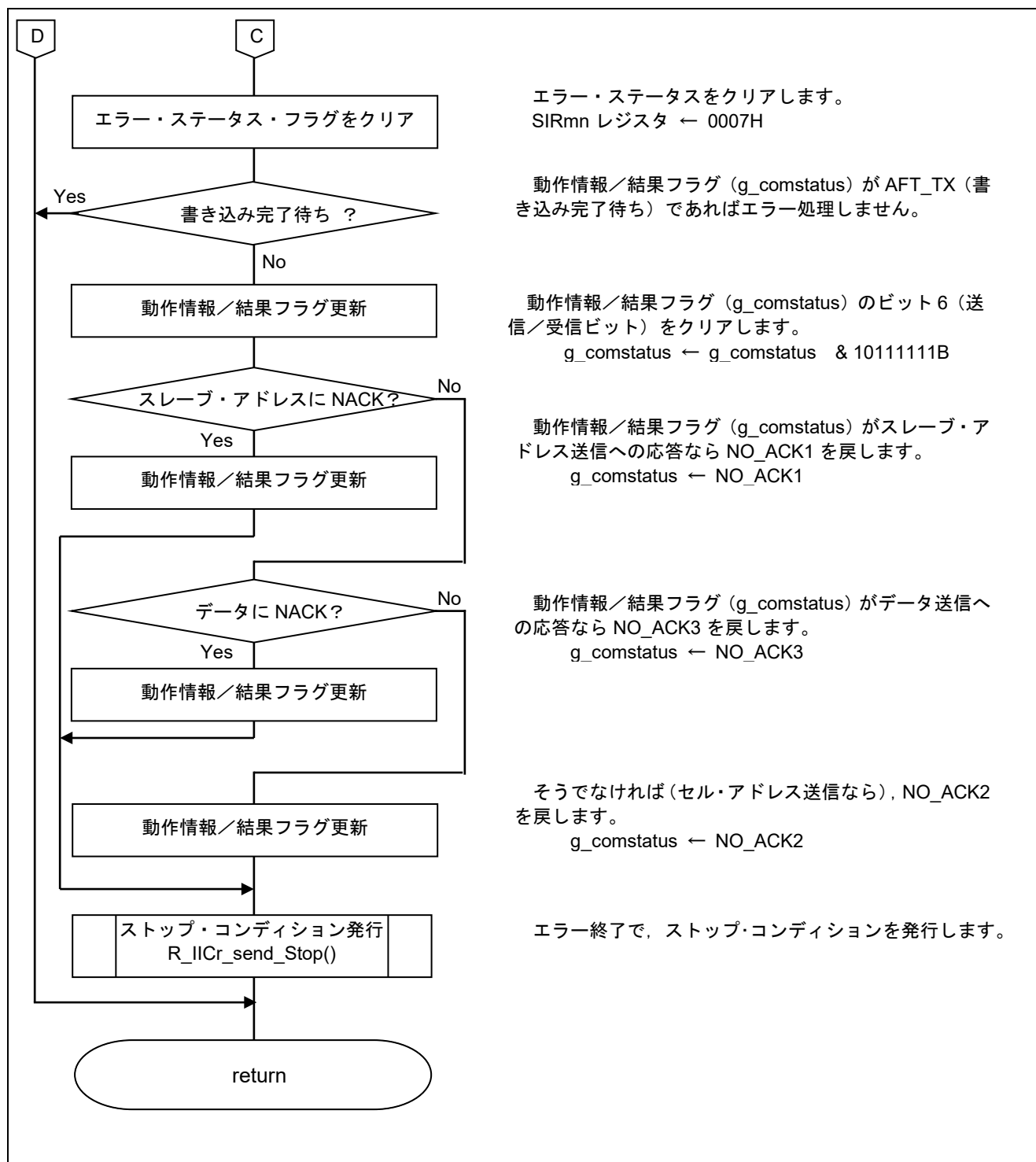


図 5.41 INTIICr 割り込み処理 (3/3)

5.7.30 INTTM02 割り込み処理

図 5.42 に INTTM02 割り込み処理のフローチャートを示します。

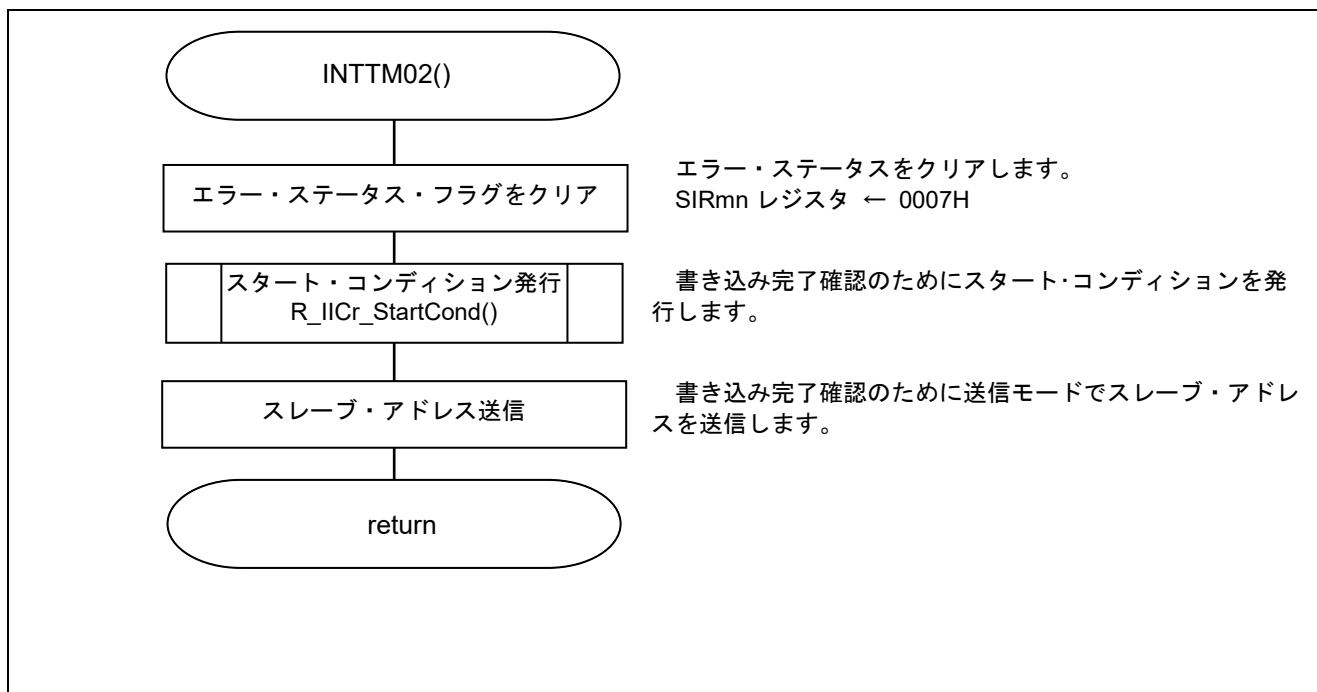


図 5.42 INTTM02 割り込み処理

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/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896J)

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。)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2025.4.11	—	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イやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違っていると、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、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

- 当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
 13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 14. 本資料に記載されている内容または当社製品についてご不明点がございましたら、当社の営業担当者までお問合せください。

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

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

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

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。