

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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

SH7764 グループ

I²C バスインタフェース シングルマスタ送受信例

要旨

本アプリケーションノートは、SH7764 の I²C バスインタフェース (IIC) のシングルマスタでの送信および受信例を紹介します。EEPROM にデータをライト/リードアクセスする動作にて説明します。

動作確認デバイス

SH7764 (ルネサステクノロジ製 R0K507764E001BR)

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラム “iic.c”.....	16
4. 参考ドキュメント.....	25

1. はじめに

1.1 仕様

マスタデバイスを SH7764、スレーブデバイスを EEPROM として、データの送受信を行います。

FAST-MODE (SCL 周波数 : Max 400kHz) にて接続しています。(346.15kHz に設定しています)

【注】 EEPROM の仕様を満足するように設定してください。

1.2 使用機能

I²C バスインタフェース (IIC)

1.3 適用条件

- マイコン: SH7764
- 動作周波数:

CPU クロック	324MHz
SuperHyway クロック	108MHz
周辺クロック	54MHz
バスクロック	108MHz
- 統合開発環境 ルネサステクノロジ製
- C コンパイラ ルネサステクノロジ製 SuperH RISC engine ファミリ
C/C++ コンパイラパッケージ Ver.9.03 Release 00
- コンパイルオプション High-performance Embedded Workshop でのデフォルト設定

```
-cpu=sh4a -endian=little -include="$(WORKSPDIR)¥inc"
-object="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -optimize=0
-gbr=auto -chgincpath -errorpath
-global_volatile=0 -opt_range=all -infinite_loop=0
-del_vacant_loop=0 -struct_alloc=1 -nologo
```

1.4 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。合わせて参照してください。

SH7764 グループ 初期設定例 (RJJ05B1508)

2. 応用例の説明

本応用例では、I²C バスインタフェース (IIC) を使用し、マスタデバイスの SH7764 がスレーブデバイスの EEPROM ヘデータを書き込んだ後に、書き込んだデータを読み出します。

IIC についての詳細は、「SH7764 グループ ハードウェアマニュアル」を参照してください。

2.1 IIC の動作概要

2.1.1 特長

I²C バスインタフェース (IIC) には次のような特長があります。

- I²C バスインタフェース方式に対応
- マルチマスタ対応
- 7ビット/10ビットアドレス互換マスタ
- 7ビットのスレーブアドレス
- ファースト・モード対応
- SCL クロック周波数可変

図 1 に IIC のブロック図を示します。

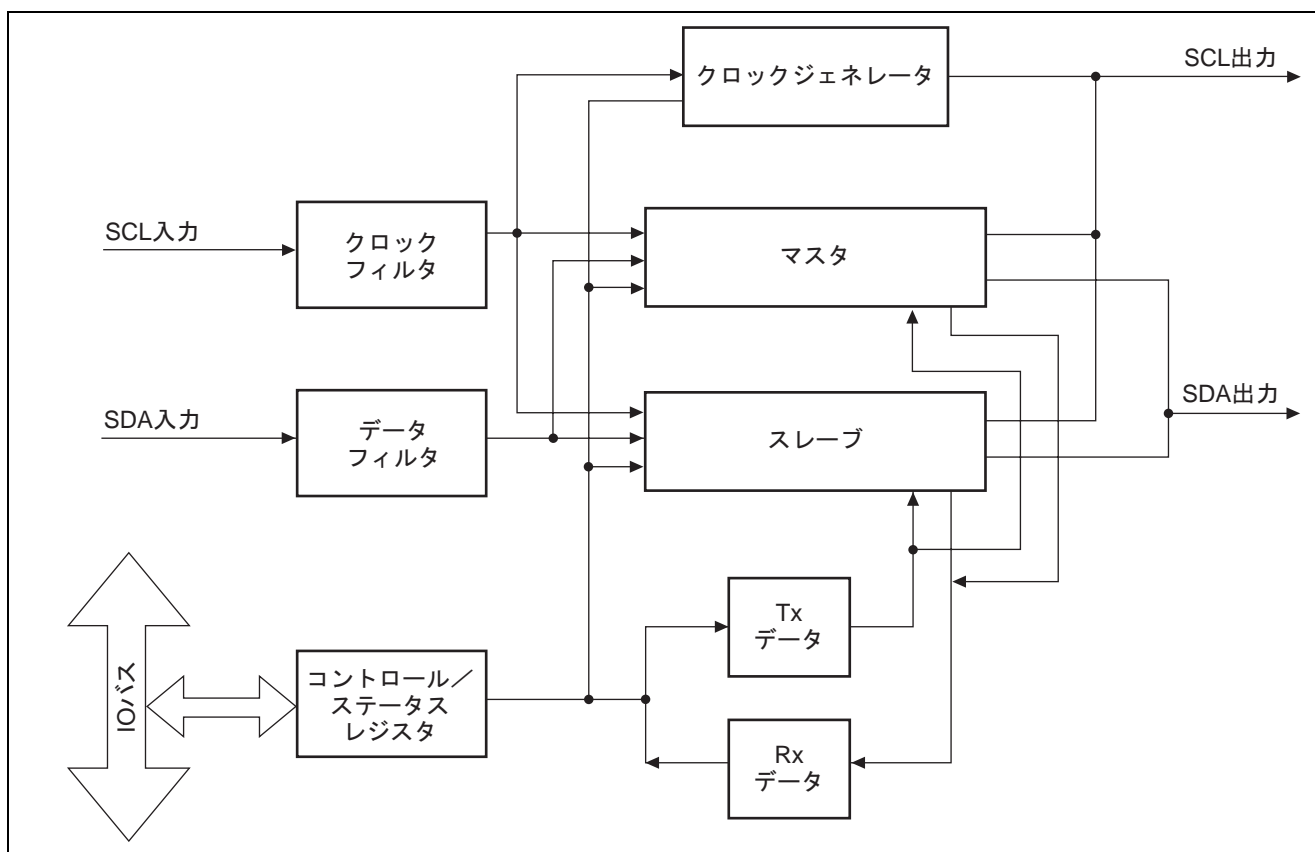


図 1 IIC のブロック図

2.1.2 7ビットアドレスの通信フォーマット

図2にマスタデバイスからスレーブデバイスへのデータ転送フォーマット（マスタデータ送信フォーマット）を示します。図3に、1回の転送中にデータ転送の方向が変わる複合転送フォーマットを示します。

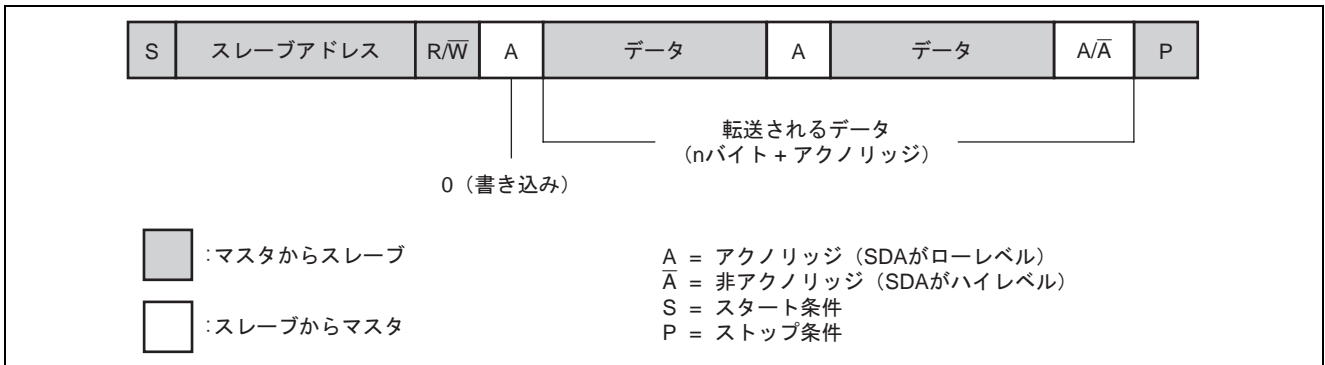


図2 マスタデータ送信フォーマット

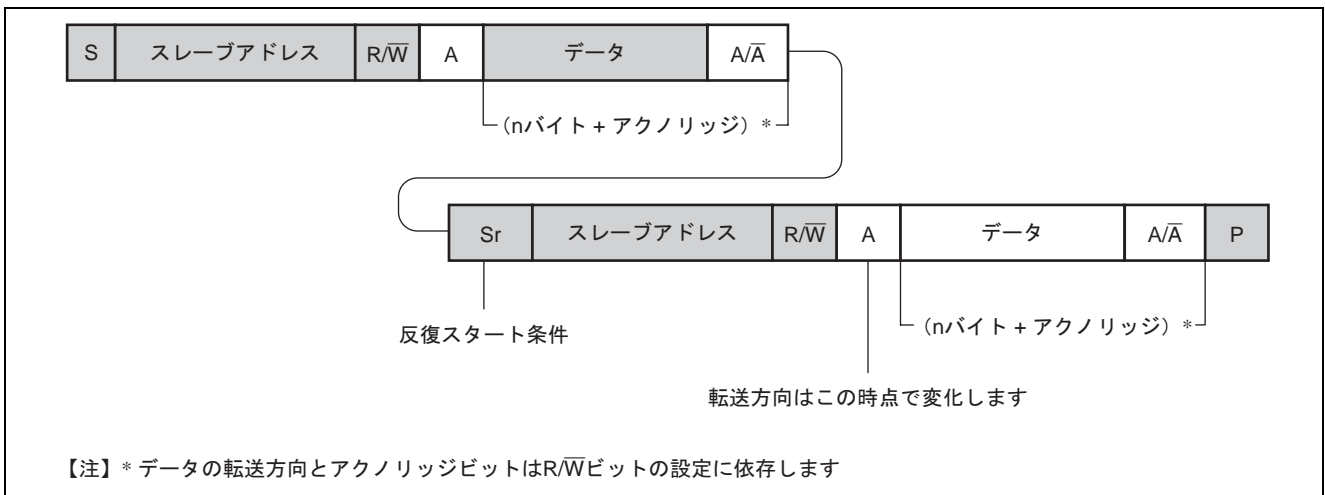


図3 マスタ転送の複合転送フォーマット

2.2 端子接続例

図4に本応用例における端子接続例を示します。

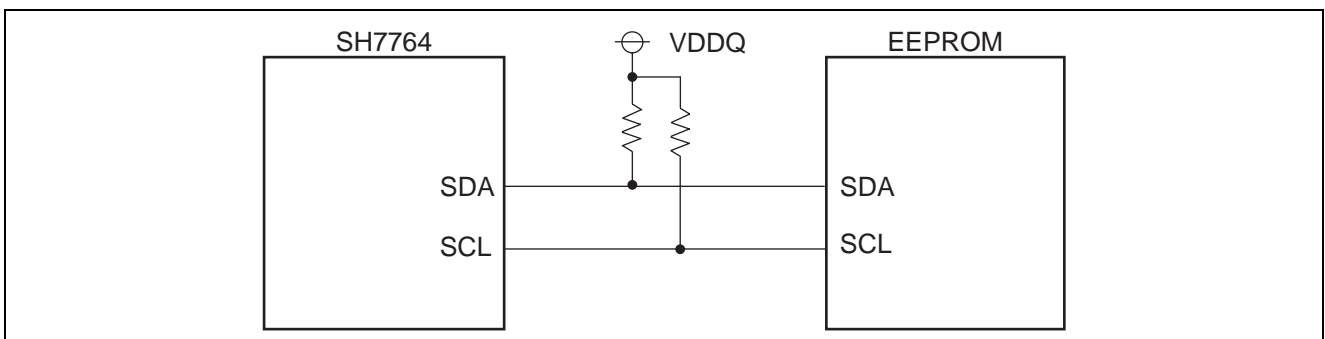


図4 I²C バスインタフェース端子接続例

2.3 参考プログラムの仕様

ここでは参考プログラムの仕様と各処理のフローチャートを説明します。

2.3.1 仕様

- SH7764 の IIC をマスタ、EEPROM をスレーブとし 10 バイトのライト/リードアクセスを行います。
- マイコンリセット後の IIC の初期化時に強制的にバスを開放します。(H/W のノイズ対策を考慮)
- IIC の送信動作、受信動作ともに、通信中に非アクノリッジ (NACK) を検出した場合は、通信を中断して転送動作をリトライ (5 回まで) します。

本応用例では、デバイスコード“B'1010”、デバイスアドレス“B'000”を使用しています。EEPROM のメモリアドレスは 1 バイトで指定しています。

図 5 に、EEPROM への 10 バイト書き込み時の通信フォーマットを、図 6 に EEPROM からの 10 バイト読み出し時の通信フォーマットを示します。

【注】 EEPROM の仕様を満足するように制御してください。

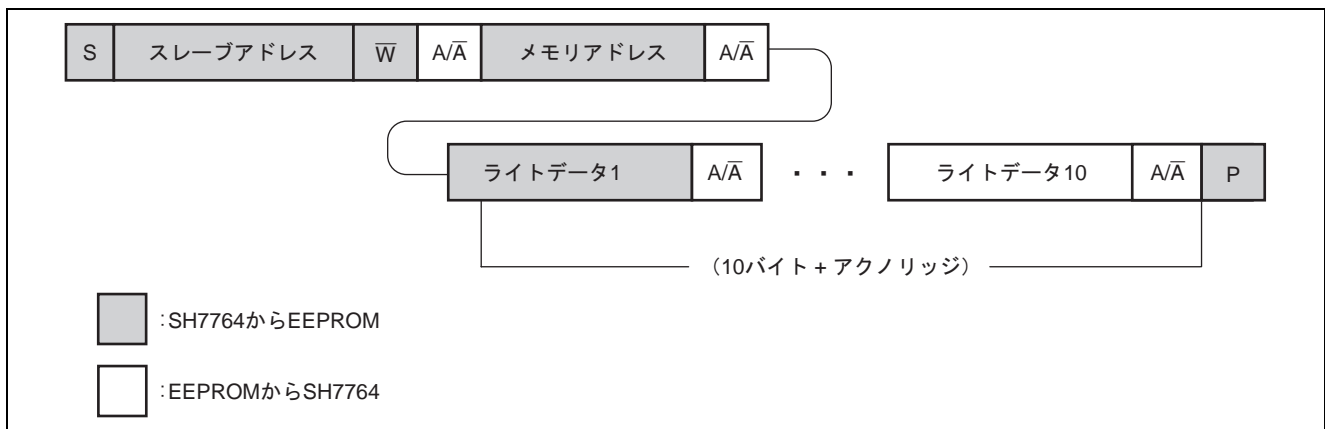


図 5 EEPROM への 10 バイト書き込み時の通信フォーマット

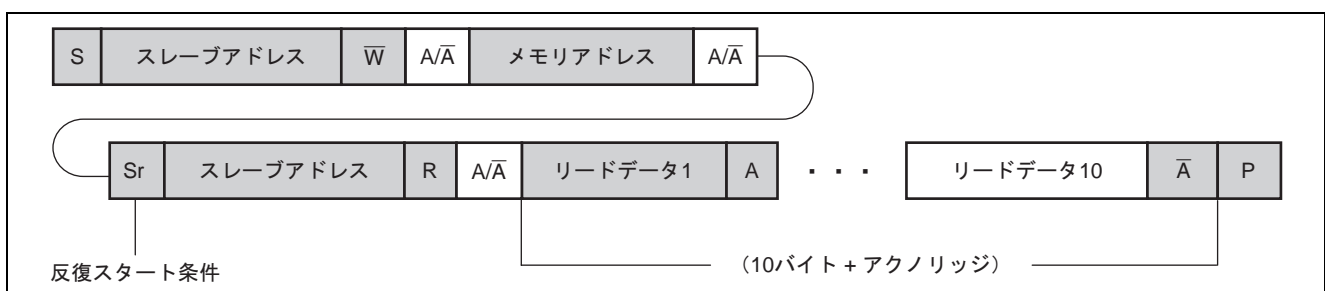


図 6 EEPROM からの 10 バイト読み出し時の通信フォーマット

2.3.2 参考プログラムメインフロー

図7に参考プログラムのメインフローを示します。

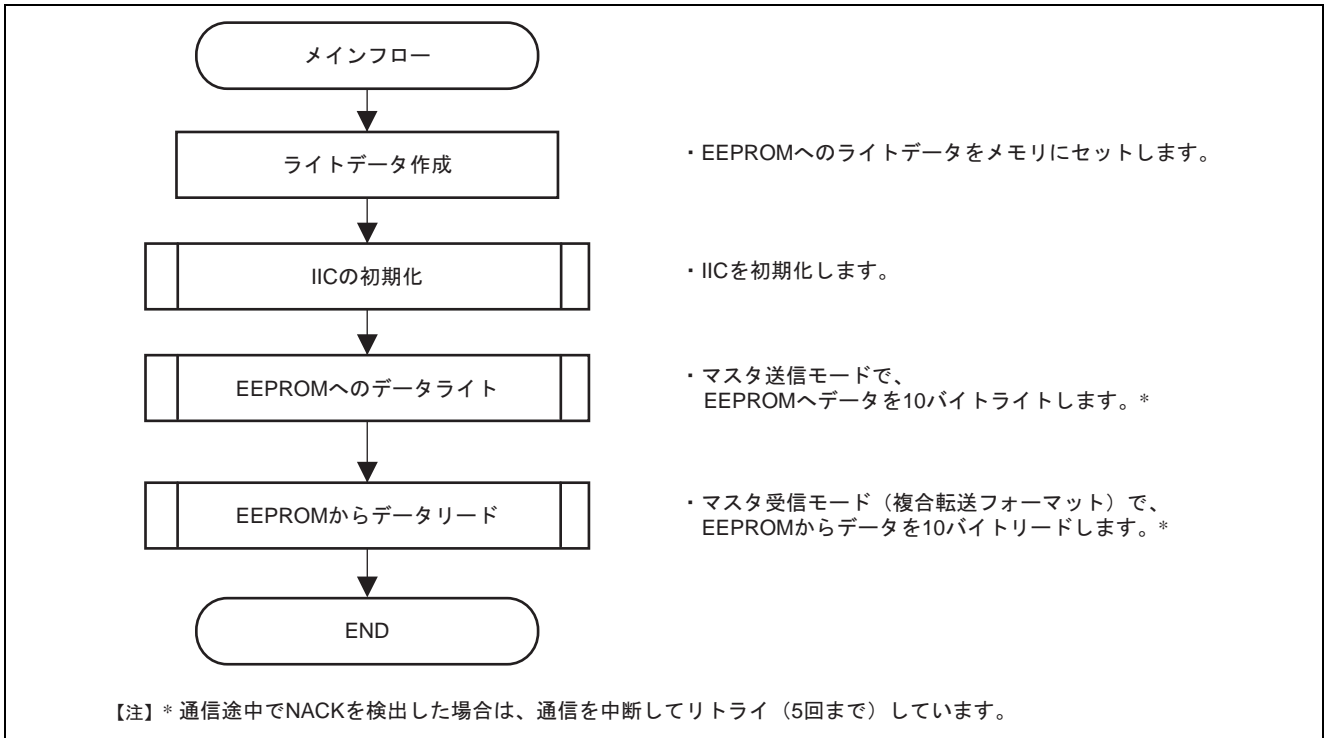


図7 参考プログラムのメインフロー

2.3.3 IICの初期化

図8にIICの初期化フローを示します。

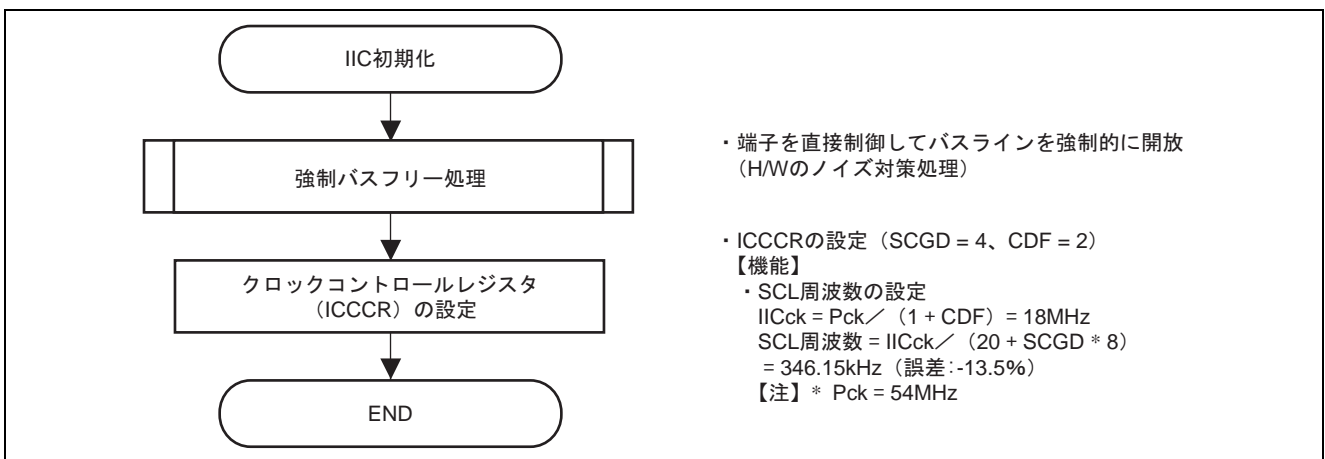


図8 IICの初期化フロー

2.3.4 EEPROM へのデータライト

図9に IIC を使用した EEPROM へのデータライト例を示します。2 バイト以上の連続したデータライトが可能です。

【注】 EEPROM の仕様を満足するように制御してください。

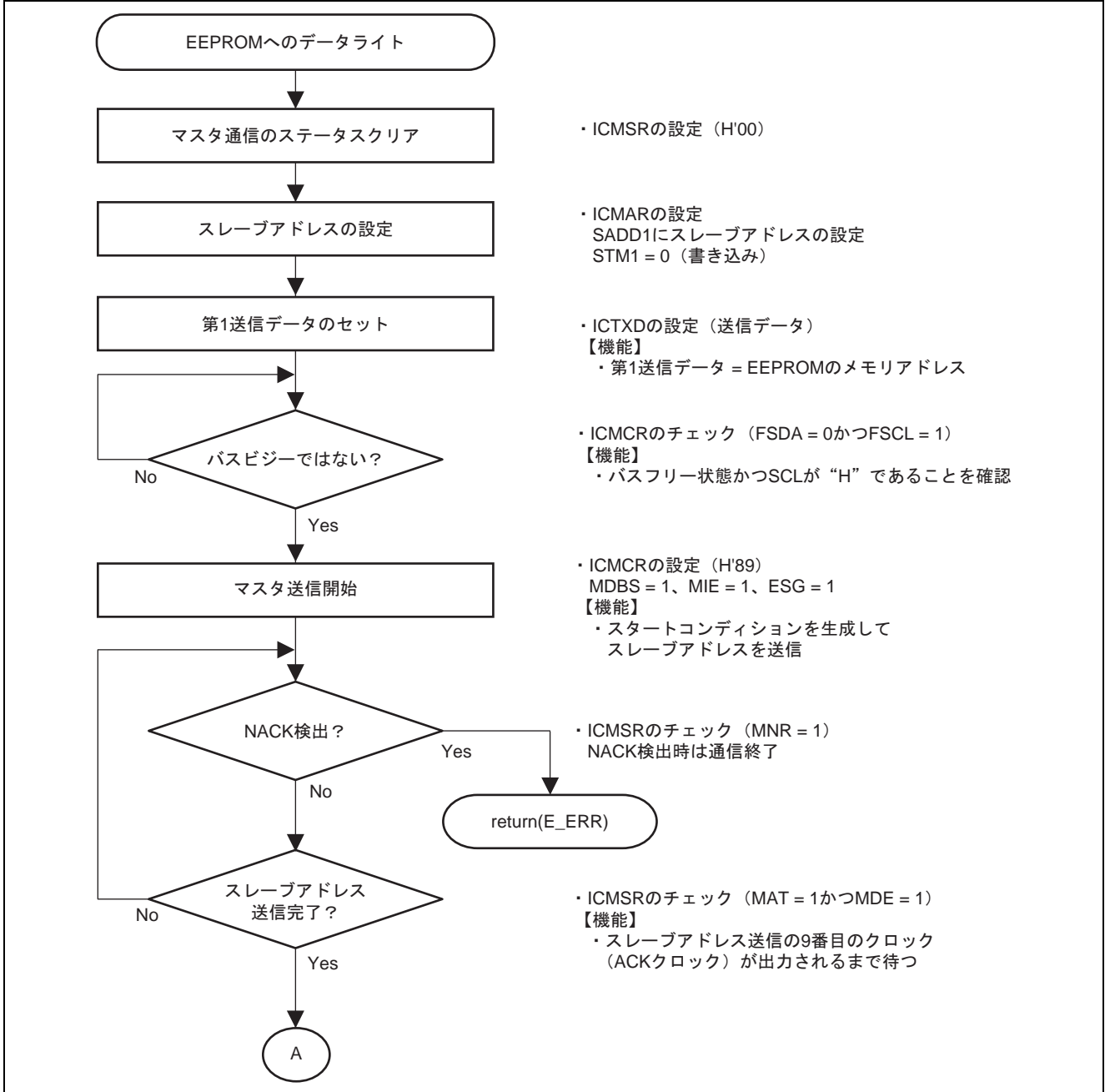


図9 EEPROM へのデータライト (1)

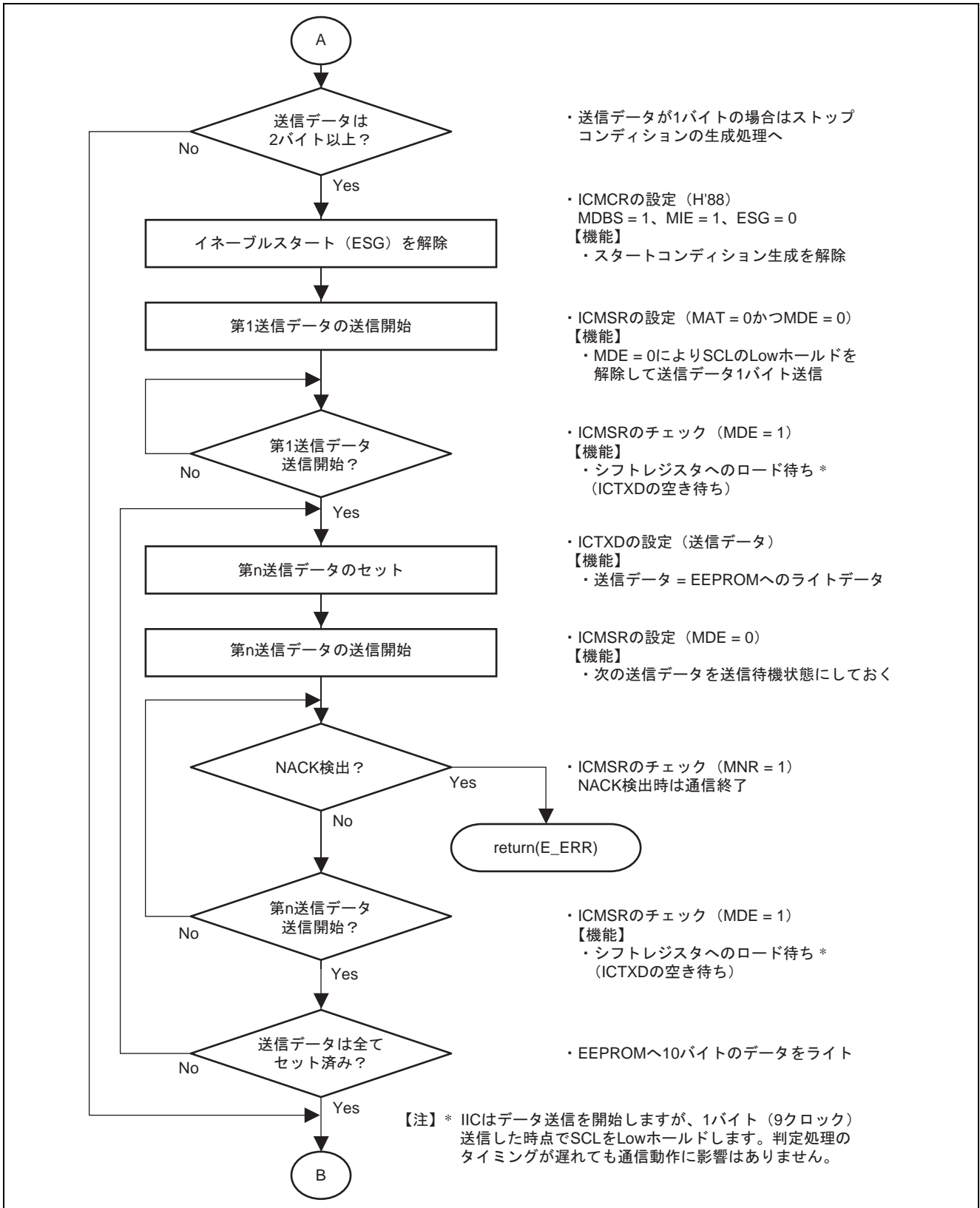


図 9 EEPROM へのデータライト (2)

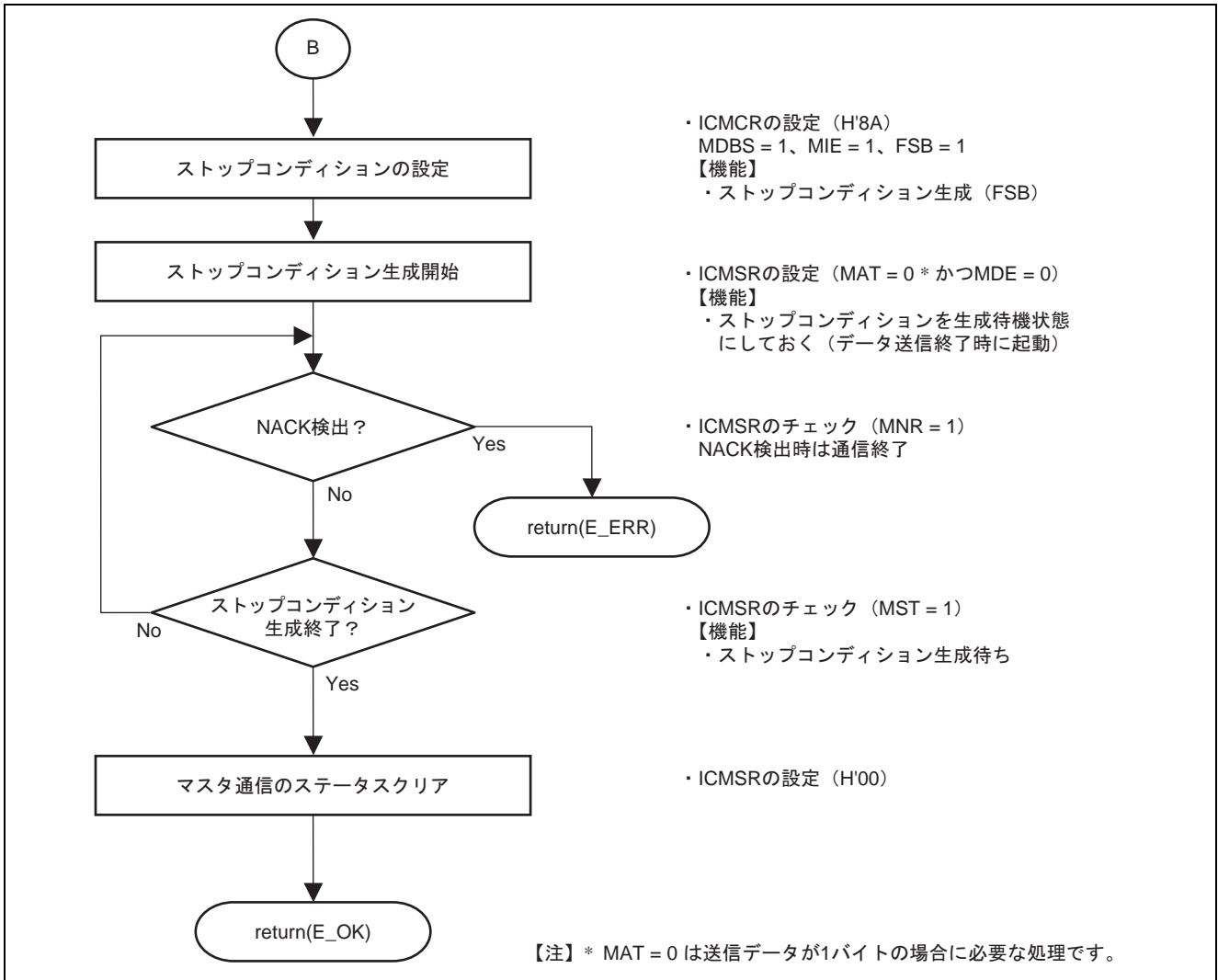


図 9 EEPROM へのデータライト (3)

2.3.5 EEPROM からのデータリード

図 10 に IIC を使用した EEPROM からのデータリード例を示します。2 バイト以上の連続したデータリードが可能です。

【注】 EEPROM の仕様を満足するように制御してください。

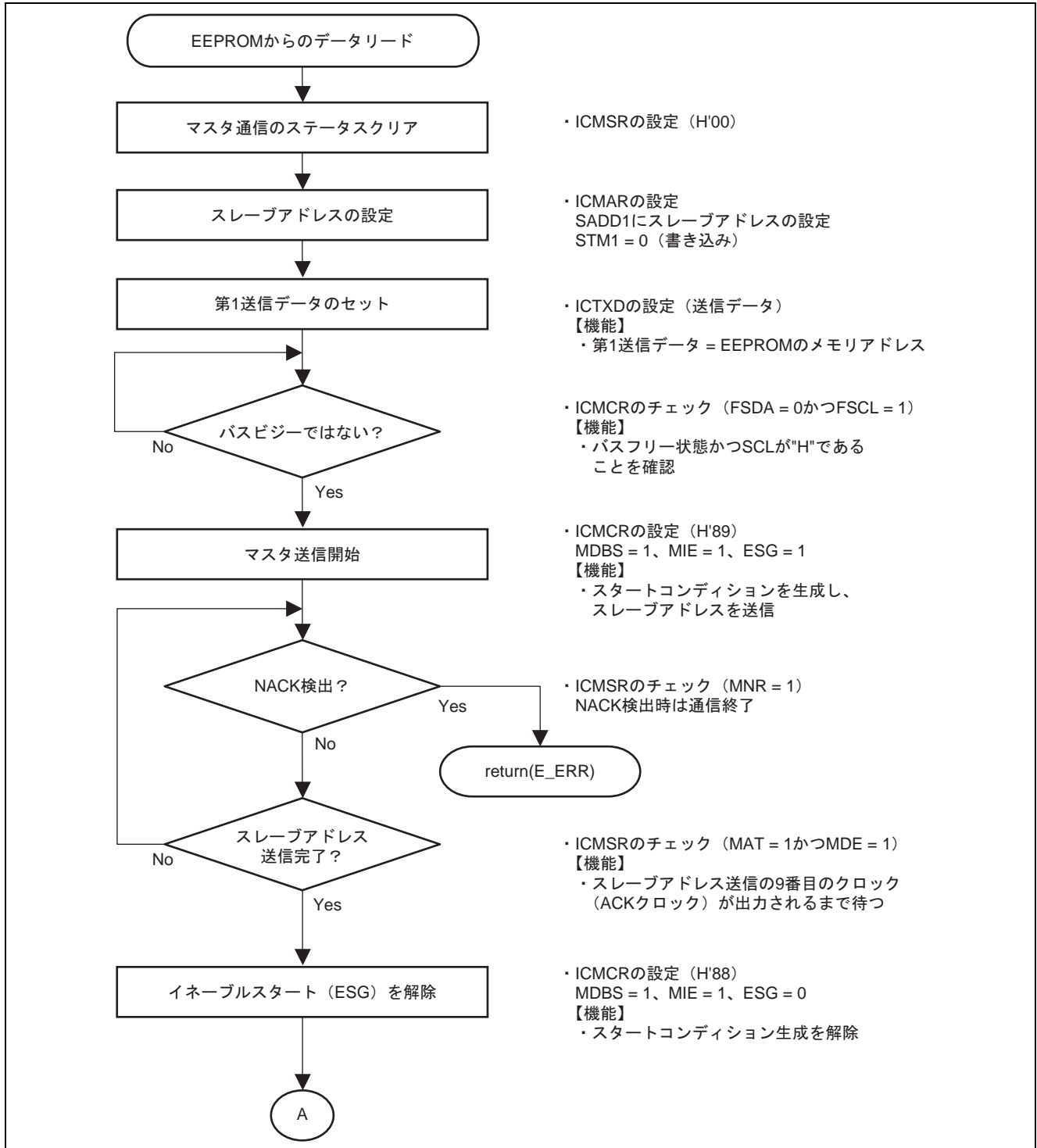


図 10 EEPROM からのデータリード (1)

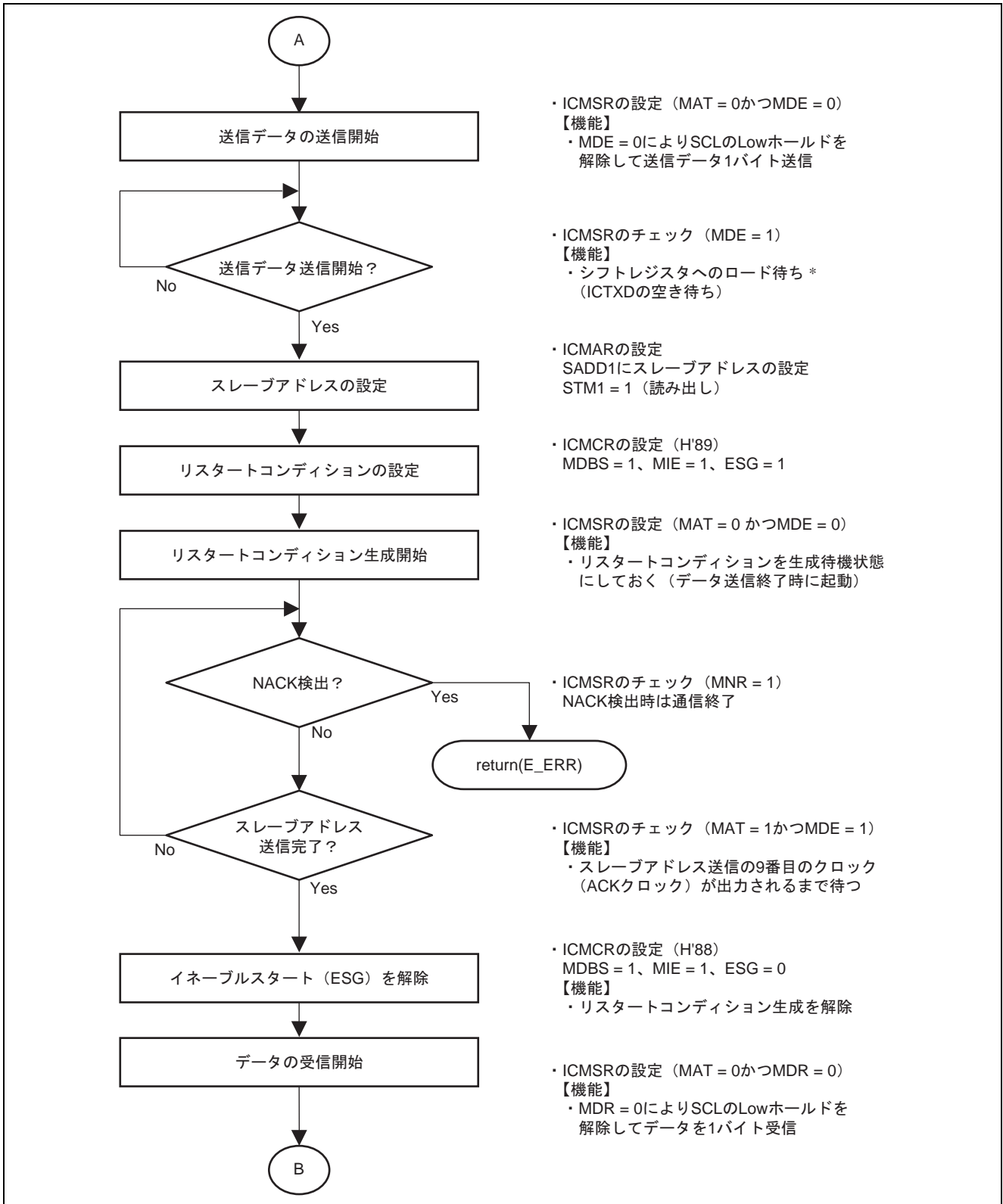


図 10 EEPROM からのデータリード (2)

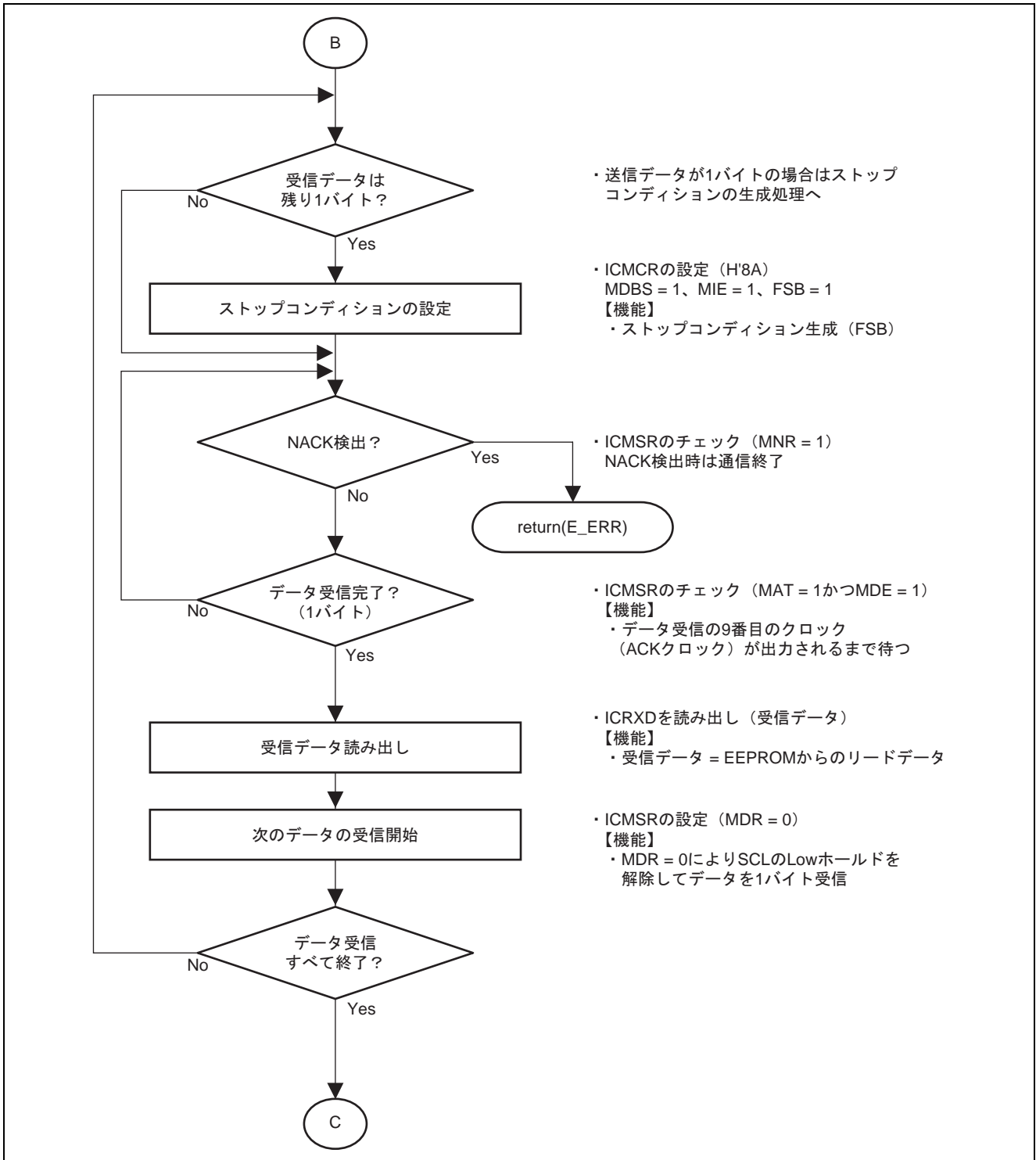


図 10 EEPROM からのデータリード (3)

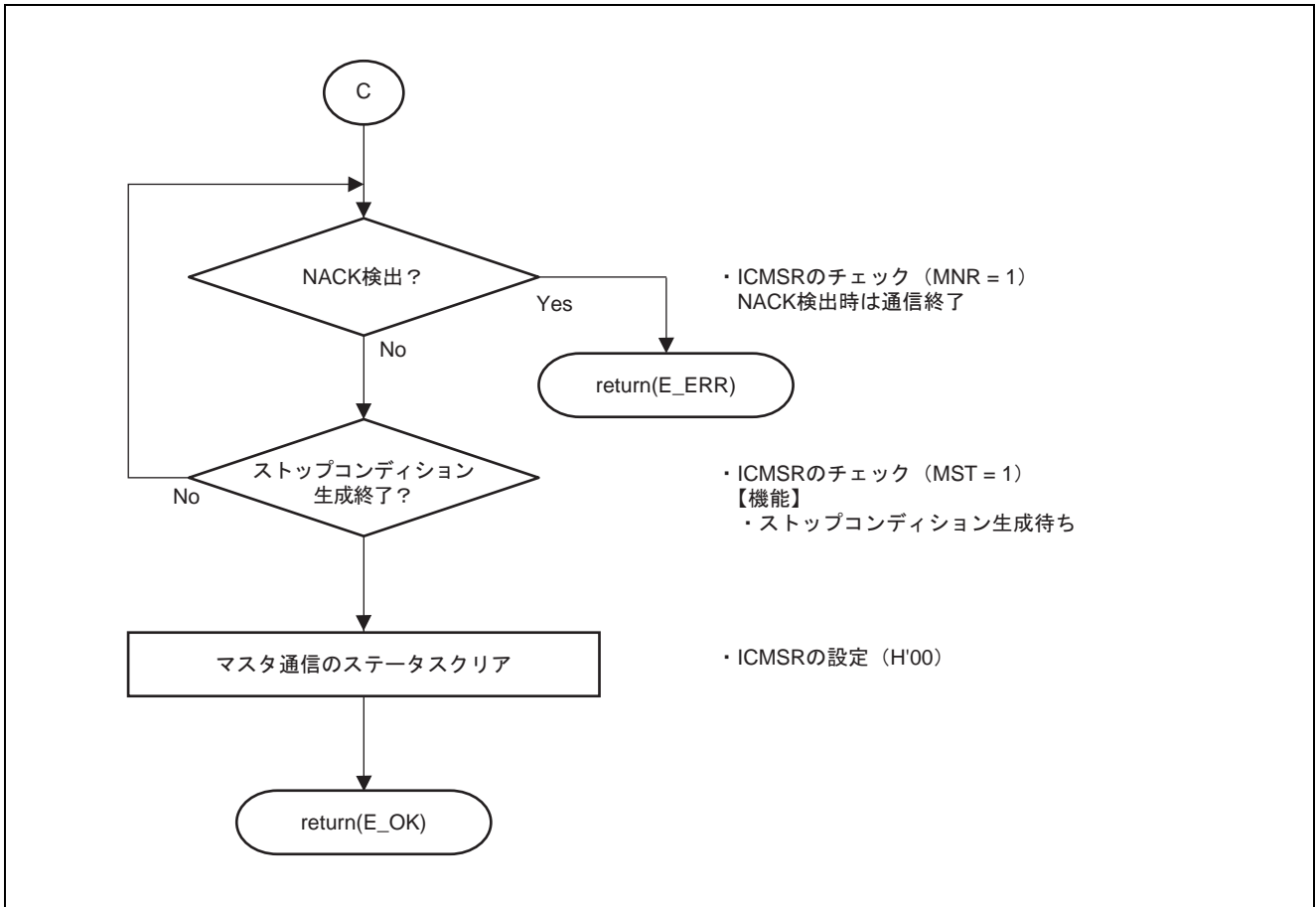


図 10 EEPROM からのデータリード (4)

2.3.6 通信中断処理

図 11 に通信中断の処理フローを示します。

SH7764 の I²C バスインタフェース (IIC) は、マスタ送受信中のスレーブアドレス送信に対して非アクリッジ (NACK) を検出すると、自動的にスレーブアドレスを再送します。このリトライ処理は、スレーブデバイスがアクリッジ (ACK) を応答するまで繰り返します。

通信を中断する場合は、IIC の動作を停止させてからバスを開放します。IIC の動作を停止する際は、正常系の動作だけでなく、スレーブアドレスの自動再送処理も停止してください。

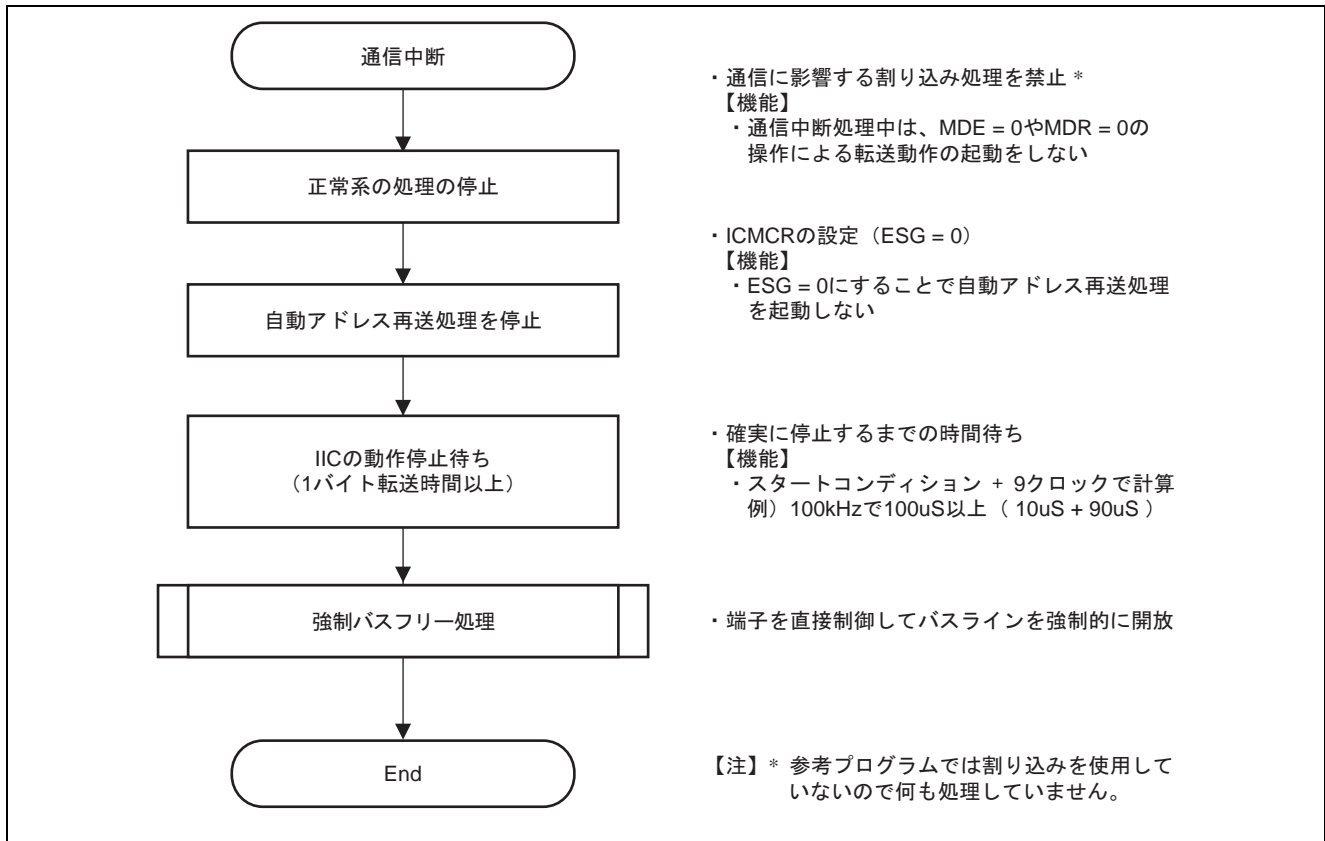


図 11 NACK 時の通信中断処理

2.3.7 強制バスフリー処理

図 12 に強制バスフリーの処理フローを示します。

IIC がマスタの場合に、強制的にストップコンディションを生成してバスラインを開放します。ノイズ等の影響にてバスラインがビジー状態に陥った場合も、本フローに従って、バスラインを開放しています。また、NACK を検出した場合も本フローに従って、バスラインを開放してから、通信をリトライしています。

IIC がバスを開放しても、通信相手デバイスがバスを Low ホールドしている場合があります。その場合は、相手デバイスをハードウェアリセットする等の対策が必要になります。

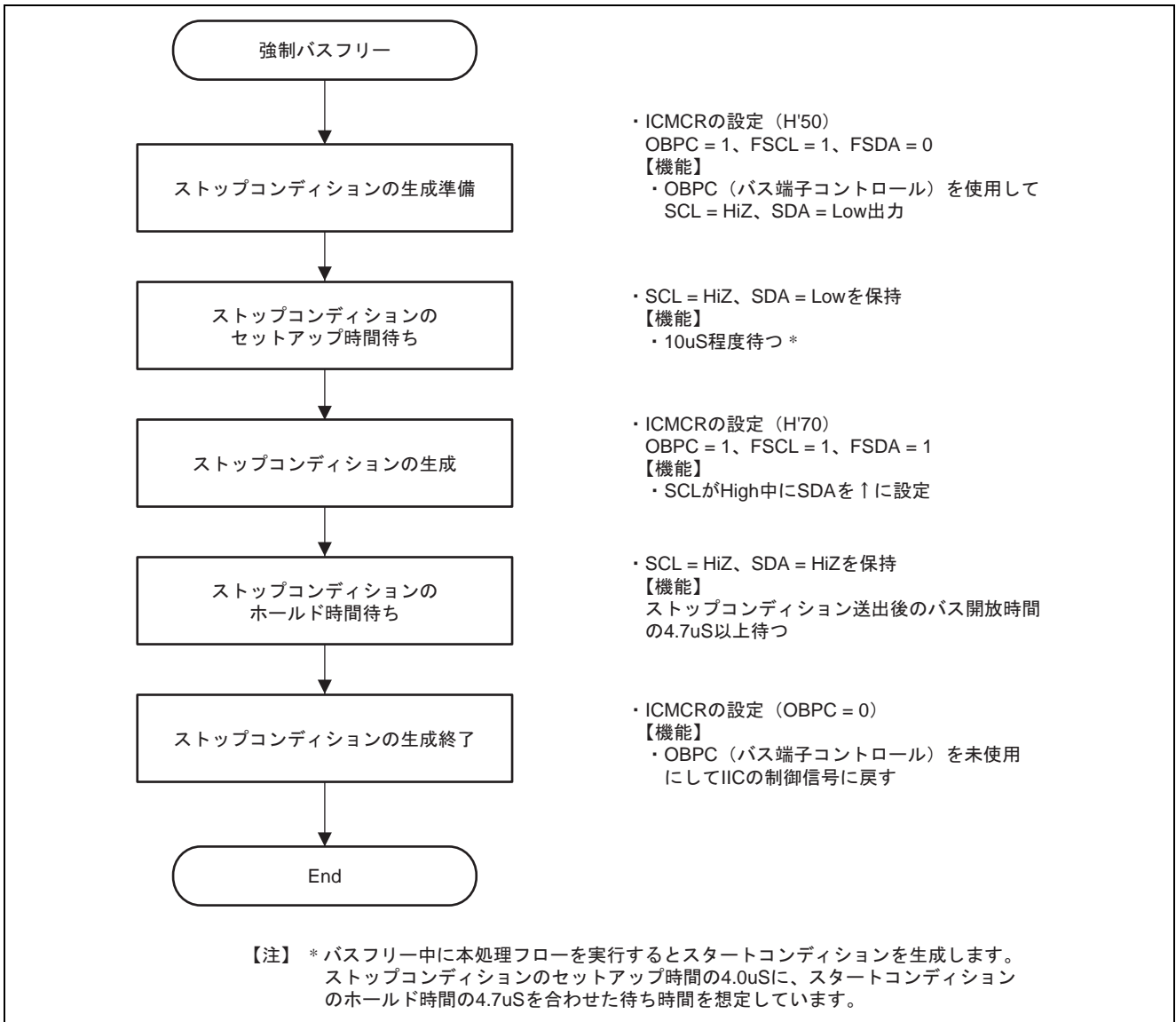


図 12 強制バスフリー

3. 参考プログラム “iic.c”

3.1 サンプルプログラムリスト “マクロ定義”

```

1.  /*"FILE COMMENT"***** Technical reference data *****
2.  *
3.  *      System Name : SH7764 Sample Program
4.  *      File Name   : iic.c
5.  *      Abstract    : I2C シングルマスタ送受信
6.  *      Version     : 1.00.00
7.  *      Device      : SH7764
8.  *      Tool-Chain  : High-performance Embedded Workshop (Ver.4.05.01).
9.  *                  : C/C++ compiler package for the SuperH RISC engine family
10. *                  : (Ver.9.03 Release00).
11. *      OS          : none
12. *      H/W Platform: R0K507764E001BR
13. *      Disclaimer  :
14. *                  <注意事項>
15. *                  本サンプルプログラムはすべて参考資料であり、
16. *                  その動作を保証するものではありません。
17. *                  本サンプルプログラムはお客様のソフトウェア開発時の
18. *                  技術参考資料としてご利用ください。
19. *
20. *      The information described here may contain technical inaccuracies or
21. *      typographical errors. Renesas Technology Corporation and Renesas Solutions
22. *      assume no responsibility for any damage, liability, or other loss rising
23. *      from these inaccuracies or errors.
24. *      Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
25. *      AND Renesas Solutions Corp. All Rights Reserved
26. *
27. *      History     : Sep.01,2009 Ver.1.00.00
28. * "FILE COMMENT END"*****/
29. #include "iodefine.h"
30. /* ==== マクロ定義 ==== */
31. #define DEVICE_CODE    0xA0
32. #define DEVICE_ADDR    0x00
33. #define EEPROM_MEM_ADDR0x0000
34. #define RETRY_MAX      5
35. #define E_OK           0
36. #define E_ERR          -1
37. /* ==== 関数プロトタイプ宣言 ==== */
38. void iic_main(void);
39. void iic_initialize(void);
40. long iic_eeeprom_write(unsigned char d_code,
41.                          unsigned char d_addr,
42.                          unsigned char w_addr,
43.                          unsigned short w_size,
44.                          unsigned char * w_buf);
45. long iic_eeeprom_read(unsigned char d_code,
46.                        unsigned char d_addr,
47.                        unsigned char r_addr,
48.                        unsigned short r_size,
49.                        unsigned char * r_buf);
50. void iic_retry_func(void);
51. void iic_nack_end(void);
52. void iic_iic_bus_free(void);
53. void delay(unsigned long value);
    
```

3.2 サンプルプログラムリスト “メイン処理”

```

54. /*"FUNC COMMENT"*****
55. * ID      :
56. * Outline : I2C メイン処理
57. *-----
58. * Include :
59. *-----
60. * Declaration : void iic_main(void);
61. *-----
62. * Function  : IIC のマスタ送信モードを用いて、EEPROM とデータを送受信します
63. *-----
64. * Argument  : void
65. *-----
66. * Return Value: void
67. /*"FUNC COMMENT END"*****/
68. void iic_main(void)
69. {
70.     unsigned char wrData[10];
71.     unsigned char rdData[10];
72.     int retry_cnt, i;
73.
74.     for(i = 0; i < 10; i++)
75.         wrData[i] = i;
76.
77.     /* ==== IIC モジュール初期化 ==== */
78.     iic_initialize();
79.
80.     /* ==== EEPROM へのライト ==== */
81.     for(retry_cnt = 0; retry_cnt < RETRY_MAX; retry_cnt++){
82.         if(E_OK == iic_eeprom_write(DEVICE_CODE, DEVICE_ADDR, 0x0000, sizeof(wrData), wrData))
83.             break;
84.
85.         iic_retry_func();
86.     }
87.
88.     /* ==== EEPROM からリード ==== */
89.     for(retry_cnt = 0; retry_cnt < RETRY_MAX; retry_cnt++){
90.         if(E_OK == iic_eeprom_read(DEVICE_CODE, DEVICE_ADDR, 0x0000, sizeof(rdData), rdData))
91.             break;
92.
93.         iic_retry_func();
94.     }
95. }

```

3.3 サンプルプログラムリスト “IIC 初期化”

```

96. /*"FUNC COMMENT"*****
97. * ID      :
98. * Outline : IICの初期化
99. *-----
100. * Include :
101. *-----
102. * Declaration : void iic_initialize(void);
103. *-----
104. * Function   : 強制バスフリー後、シリアルクロックを設定します。
105. *-----
106. * Argument   : void
107. *-----
108. * Return Value: void
109. *"FUNC COMMENT END"*****/
110. void iic_initialize(void)
111. {
112.     iic_iic_bus_free();
113.
114.     /* ---- シリアルクロック設定 ---- */
115.     IIC.ICCCR.BIT._SCGD = 0x04;
116.     IIC.ICCCR.BIT._CDF = 0x02;
117.     /* I2C-clk 400kHz(346.15kHz) */
118. }
    
```

3.4 サンプルプログラムリスト “EEPROM のライト”

```

119. /*"FUNC COMMENT"*****
120. * ID      :
121. * Outline : EEPROM へのデータライト
122. * -----
123. * Include :
124. * -----
125. * Declaration : int iic_eeprom_write(unsigned char d_code,
126. *      :          unsigned char d_addr,
127. *      :          unsigned char w_addr,
128. *      :          unsigned short w_size,
129. *      :          unsigned char * w_buf);
130. * -----
131. * Description : デバイスコード d_code, デバイスアドレス d_addr で指定した
132. *      : EEPROM へ w_buf で指定した領域のデータを w_size バイト分
133. *      : 書き込みます。EEPROM のメモリアドレス
134. *      : は w_addr で指定します。
135. * -----
136. * Argument   : unsigned char d_code : デバイスコード
137. *      : unsigned char d_addr : デバイスアドレス
138. *      : unsigned char w_addr : 書き込み開始アドレス
139. *      : unsigned short w_size : 書き込みデータサイズ
140. *      : unsigned char* w_buf  : 書き込みデータ格納先
141. * -----
142. * Return Value: 正常      : E_OK
143. *      : NACK 検出   : E_ERR
144. *"FUNC COMMENT END"*****/
145. long iic_eeprom_write(unsigned char d_code,
146.      unsigned char d_addr,
147.      unsigned char w_addr,
148.      unsigned short w_size,
149.      unsigned char * w_buf)
150. {
151.     unsigned int i;
152.
153.     /* ---- マスタステータスクリア ---- */
154.     IIC.ICMSR.BYTE = 0x00;
155.
156.     /* ---- スレーブアドレス設定 ---- */
157.     IIC.ICMAR.BYTE = (d_code | d_addr) & 0xFE;
158.     /* bit0 (STM) = 0 送信 */
159.
160.     /* ---- 第1送信データセット ---- */
161.     IIC.ICRXD_ICTXD = w_addr;
162.
163.     /* ---- バスビジーチェック ---- */
164.     while(1){
165.         if((IIC.ICMCR.BIT._FSDA == 0) && (IIC.ICMCR.BIT._FSCL == 1))
166.             break;
167.     }
168.
169.     /* ---- マスタ送信開始 ---- */
170.     IIC.ICMCR.BYTE = 0x89;
171.
172.     while(1){
173.         /* ---- NACK チェック ---- */
174.         if(IIC.ICMSR.BIT._MNR)

```

```

175.         return E_ERR;
176.         /* スレーブアドレス送信完了? */
177.         if(IIC.ICMSR.BIT._MDE && IIC.ICMSR.BIT._MAT)
178.             break;
179.     }
180.
181.     /* 送信データは 2 バイト以上? */
182.     if(w_size){
183.         /* ---- イネーブルスタートを解除 ---- */
184.         IIC.ICMCR.BYTE = 0x88;
185.         /* ---- 第 1 送信データの送信開始 ---- */
186.         IIC.ICMSR.BIT._MAT = 0;
187.         IIC.ICMSR.BIT._MDE = 0;
188.         /* 第 1 送信データ送信開始待ち */
189.         while(1){
190.             if(IIC.ICMSR.BIT._MDE)
191.                 break;
192.         }
193.         for(i = 0; i < w_size; i++){
194.             /* ---- 第 n 送信データセット ---- */
195.             IIC.ICRXD_ICTXD = *w_buf;
196.             w_buf++;
197.             /* 第 n 送信データ送信開始 ---- */
198.             IIC.ICMSR.BIT._MDE = 0;
199.
200.             while(1){
201.                 /* ---- NACK チェック ---- */
202.                 if(IIC.ICMSR.BIT._MNR)
203.                     return E_ERR;
204.                 /* 第 n 送信データ送信開始チェック */
205.                 if(IIC.ICMSR.BIT._MDE)
206.                     break;
207.             }
208.         }
209.     }
210.
211.     /* ---- ストップコンディション設定 ---- */
212.     IIC.ICMCR.BYTE = 0x8A;
213.     /* ---- ストップコンディション生成開始 ---- */
214.     IIC.ICMSR.BIT._MDE = 0;
215.     IIC.ICMSR.BIT._MAT = 0;
216.
217.     while(1){
218.         /* ---- NACK チェック ---- */
219.         if(IIC.ICMSR.BIT._MNR)
220.             return E_ERR;
221.         /* ---- ストップコンディション生成待ち ---- */
222.         if(IIC.ICMSR.BIT._MST)
223.             break;
224.     }
225.
226.     /* ---- マスタステータスクリア ---- */
227.     IIC.ICMSR.BYTE = 0x00;
228.
229.     return E_OK;
230. }
    
```

3.5 サンプルプログラムリスト “EEPROM のリード”

```

231. /*"FUNC COMMENT"*****
232. * ID      :
233. * Outline : EEPROM のデータリード
234. * -----
235. * Include :
236. * -----
237. * Declaration : int iic_eeprom_read(unsigned char d_code,
238. *      :          unsigned char d_addr,
239. *      :          unsigned char r_addr,
240. *      :          unsigned short r_size,
241. *      :          unsigned char * r_buf);,
242. * -----
243. * Description : デバイスコード d_code, デバイスアドレス d_addr で指定した
244. *      : EEPROM から r_buf で指定した領域に w_size バイト分
245. *      : データを読み込みます。EEPROM のメモリアドレス
246. *      : は r_addr で指定します。
247. * -----
248. * Argument   : unsigned char d_code : デバイスコード
249. *      : unsigned char d_addr : デバイスアドレス
250. *      : unsigned char r_addr : 読み込み開始アドレス
251. *      : unsigned short r_size : 読み込みデータサイズ
252. *      : unsigned char* r_buf  : 読み込みデータ格納先
253. * -----
254. * Return Value: 正常      : E_OK
255. *      : NACK 検出   : E_ERR
256. * "FUNC COMMENT END"*****/
257. long iic_eeprom_read(unsigned char d_code,
258.      unsigned char d_addr,
259.      unsigned char r_addr,
260.      unsigned short r_size,
261.      unsigned char * r_buf)
262. {
263.     unsigned int i;
264.
265.     /* ---- マスタステータスクリア ---- */
266.     IIC.ICMSR.BYTE = 0x00;
267.
268.     /* ---- スレーブアドレス設定 ---- */
269.     IIC.ICMAR.BYTE = (d_code | d_addr) & 0xFE;
270.
271.     /* ---- 送信データセット ---- */
272.     IIC.ICRXD_ICTXD = r_addr;
273.
274.     /* ---- バスビジーチェック ---- */
275.     while(1){
276.         if((IIC.ICMCR.BIT._FSDA == 0) && (IIC.ICMCR.BIT._FSCL == 1))
277.             break;
278.     }
279.
280.     /* ---- マスタ送信開始 ---- */
281.     IIC.ICMCR.BIT._MDBS = IIC.ICMCR.BIT._MIE = IIC.ICMCR.BIT._ESG = 1;
282.
283.     while(1){
284.         /* ---- NACK チェック ---- */
285.         if(IIC.ICMSR.BIT._MNR)
286.             return E_ERR;

```

```

287.
288.     /* スレーブアドレス送信完了待ち */
289.     if(IIC.ICMSR.BIT._MDE && IIC.ICMSR.BIT._MAT)
290.         break;
291.     }
292.
293.     /* ----- イネーブルスタートを解除 ----- */
294.     IIC.ICMCR.BYTE = 0x88;
295.
296.     /* ----- 送信データの送信開始 ----- */
297.     IIC.ICMSR.BYTE = 0x00;
298.
299.     /* ----- ICTXD 空き待ち ----- */
300.     while(1){
301.         if(IIC.ICMSR.BIT._MDE)
302.             break;
303.     }
304.
305.     /* ----- スレーブアドレス設定 ----- */
306.     IIC.ICMAR.BIT._STM1 = 1;
307.     /* bit0(STM) = 1 受信 */
308.
309.     /* ----- リスタートコンディション設定 ----- */
310.     IIC.ICMCR.BYTE = 0x89;
311.
312.     /* リスタートコンディション生成開始 ----- */
313.     IIC.ICMSR.BYTE = 0x00;
314.
315.     while(1){
316.         /* ----- NACK チェック ----- */
317.         if(IIC.ICMSR.BIT._MNR)
318.             return E_ERR;
319.
320.         /* スレーブアドレス送信待ち */
321.         if(IIC.ICMSR.BIT._MDR && IIC.ICMSR.BIT._MAT)
322.             break;
323.     }
324.
325.     /* ----- イネーブルスタートを解除 ----- */
326.     IIC.ICMCR.BYTE = 0x88;
327.
328.     /* ----- データ受信開始 ----- */
329.     IIC.ICMSR.BYTE = 0x00;
330.
331.     for(i = 0; i < r_size; i++){
332.         if(i == (r_size - 1))
333.             /* ----- ストップコンディションの設定 ----- */
334.             IIC.ICMCR.BYTE = 0x8A;
335.         while(1){
336.             /* ----- NACK チェック ----- */
337.             if(IIC.ICMSR.BIT._MNR)
338.                 return E_ERR;
339.
340.             /* データ受信待ち(1バイト) */
341.             if(IIC.ICMSR.BIT._MDR)
342.                 break;
343.         }
    
```



```

344.
345.     /* ---- 受信データ読み出し ---- */
346.     *r_buf = IIC.ICRXD_ICTXD;
347.     r_buf++;
348.
349.     /* ---- 次のデータ受信開始 ---- */
350.     IIC.ICMSR.BYTE = 0x00;
351. }
352.
353. while(1){
354.     /* ---- NACK チェック ---- */
355.     if(IIC.ICMSR.BIT._MNR)
356.         return E_ERR;
357.
358.     /* ストップコンディション生成待ち */
359.     if(IIC.ICMSR.BIT._MST)
360.         break;
361. }
362.
363. /* ---- マスタステータスクリア ---- */
364. IIC.ICMSR.BYTE = 0x00;
365.
366. return E_OK;
367. }
    
```

3.6 サンプルプログラムリスト “通信中断処理”

```

368. /*"FUNC COMMENT"*****
369. * ID      :
370. * Outline : NACK 時の通信中断処理
371. *-----
372. * Include :
373. *-----
374. * Declaration : void iic_nack_end(void);
375. *-----
376. * Description : NACK を受信した際、通信中断してバスを開放します。
377. *-----
378. * Argument  : void
379. *-----
380. * Return Value: void
381. *"FUNC COMMENT END"*****/
382. void iic_nack_end(void)
383. {
384.     /* ---- 自動アドレス再送処理を停止 ---- */
385.     IIC.ICMCR.BIT._ESG = 0;
386.
387.     /* ---- IIC の動作停止待ち (1 バイト転送時間以上待機) ---- */
388.     delay(10000);
389.
390.     /* ---- 強制バスフリー処理 ---- */
391.     iic_iic_bus_free();
392. }
    
```

3.7 サンプルプログラムリスト “強制バスフリー”

```

393. /*"FUNC COMMENT"*****
394. * ID      :
395. * Outline : 強制バスフリー処理
396. *-----
397. * Include :
398. *-----
399. * Declaration : void iic_iic_bus_free(void);
400. *-----
401. * Description : 強制的にストップコンディションを生成してバスを開放します。
402. *-----
403. * Argument  : void
404. *-----
405. * Return Value: void
406. *"FUNC COMMENT END"*****/
407. void iic_iic_bus_free(void)
408. {
409.     /* ---- ストップコンディションの生成準備 ---- */
410.     IIC.ICMCR.BYTE = 0x50;
411.     /* ---- 10us 待機 (4.7uS+4.0uS 以上) ---- */
412.     delay(1000);
413.     /* ---- ストップコンディションの生成 (SDA に HiZ をセット) ---- */
414.     IIC.ICMCR.BYTE = 0x70;
415.     /* ---- 5us 待機 (4.7uS 以上) ---- */
416.     delay(500);
417.     /* ---- ストップコンディションの生成終了 (SCL, SDA の出力を停止) ---- */
418.     IIC.ICMCR.BIT._OBPC = 0;
419. }
    
```

4. 参考ドキュメント

- The I²C-Bus Specification (Version 2.1), January 2000, Phillips Semiconductor.
- ハードウェアマニュアル
SH7764 ハードウェアマニュアル
(最新版をルネサス テクノロジホームページから入手してください。)
- ソフトウェアマニュアル
SH-4A ソフトウェアマニュアル
(最新版をルネサス テクノロジ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 テクノロジホームページ

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

お問合せ先

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

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.01.21	—	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。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事事務の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認いただきますとともに、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したのですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊社製品の故障または誤動作が生じた場合も人身事故、火災事故、社会的損害などを生じさせないよう、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特にマイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
 - 1 1. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
 - 1 2. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
 - 1 3. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会ください。

D039444