

SH7216 グループ

I²Cバスインタフェース 3 EEPROMリードライト例

R01AN0062JJ0101
Rev.1.01
2012.02.10

要旨

本アプリケーションノートでは、SH7216 のI²Cバスインタフェース 3 (IIC3) のマスタ送信動作およびマスタ受信動作を使用してEEPROMをリードライトアクセスする例について説明しています。

動作確認デバイス

SH7216

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラム例.....	17
4. 参考ドキュメント.....	31

1. はじめに

1.1 仕様

- マスタデバイスを SH7216、スレーブデバイスを EEPROM として、EEPROM ヘデータをライトします。
- 同様に EEPROM からデータをリードします。
- 転送レートを 195kHz に設定しています。

【注】 EEPROMの電気的特性を満足するように転送レートを設定してください。

1.2 使用機能

- I²Cバスインタフェース 3 (IIC3)

1.3 適用条件

マイコン	SH7216
動作周波数	内部クロック : 200MHz バスクロック : 50MHz 周辺クロック : 50MHz
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Ver.4.07.00
C コンパイラ	ルネサスエレクトロニクス製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.03 Release00
コンパイルオプション	High-performance Embedded Workshop でのデフォルト設定 (-cpu=sh2afpu -fpu=single -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

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。

- SH7216 グループ 初期設定例

1.5 "L"アクティブ端子 (信号) の表記について

端子名 (信号名) 末尾の # は "L" アクティブ端子 (信号) であることを示します。

2. 応用例の説明

参考プログラムでは、I²Cバスインタフェース 3 (IIC3)を使用し、マスタデバイスのSH7216 がスレーブデバイスのEEPROMへデータをライトした後、続けてリードします。

2.1 使用機能の動作概要

I²Cバスインタフェース 3 (IIC3)は、フィリップス社が提唱するI²Cバス (Inter IC Bus) インタフェース方式に準拠しており、サブセット機能を備えています。ただしI²Cバスを制御するレジスタの構成が一部フィリップス社のものと異なりますので注意してください。

SH7216 のI²Cバスインタフェース 3 (IIC3)には以下に示す特長があります。

- I²Cバスフォーマットまたはクロック同期式シリアルフォーマットを選択可能
- 連続送信/受信可能
シフトレジスタ、送信データレジスタ、受信データレジスタがそれぞれ独立しているため、連続送信/受信が可能

表 1にフォーマット別の特長を示します。図 1にIIC3 のブロック図を示します。

IIC3 についての詳細は、「SH7214 グループ、SH7216 グループ ユーザーズマニュアル ハードウェア編 I²Cバスインタフェース 3」の章を参照してください。

表1 フォーマット別の特長

フォーマット	特長
I ² Cバス フォーマット	<ul style="list-style-type: none"> ● マスタモードでは開始条件、停止条件を自動生成 ● 受信時、アクノリッジの出力レベルを選択可能 ● 送信時、アクノリッジビットを自動ロード ● ビット同期/ウェイト機能内蔵 マスタモードではビットごとに SCL の状態をモニタして自動的に同期を取ります。転送準備ができていない場合には、SCL をローレベルにして待機させます。 ● 割り込み要因：6 種類 <ol style="list-style-type: none"> ① 送信データエンプティ (スレーブアドレス一致時を含む) ② 送信終了 ③ 受信データフル (スレーブアドレス一致時を含む) ④ アービトレーションロスト ⑤ NACK 検出 ⑥ 停止条件検出 ● 送信データエンプティ割り込みと受信データフル割り込みにより、ダイレクトメモリアクセスコントローラ (DMAC) およびデータトランスファコントローラ (DTC) を起動させてデータの転送を行うことができます。 ● バスを直接駆動可能 SCL、SDA の 2 端子は、バス駆動機能選択時 NMOS オープンドレイン出力
クロック同期式 シリアル フォーマット	<ul style="list-style-type: none"> ● 割り込み要因：4 種類 <ol style="list-style-type: none"> ① 送信データエンプティ ② 送信終了 ③ 受信データフル ④ オーバランエラー ● 送信データエンプティ割り込みと受信データフル割り込みにより、ダイレクトメモリアクセスコントローラ (DMAC) およびデータトランスファコントローラ (DTC) を起動させてデータの転送を行うことができます。

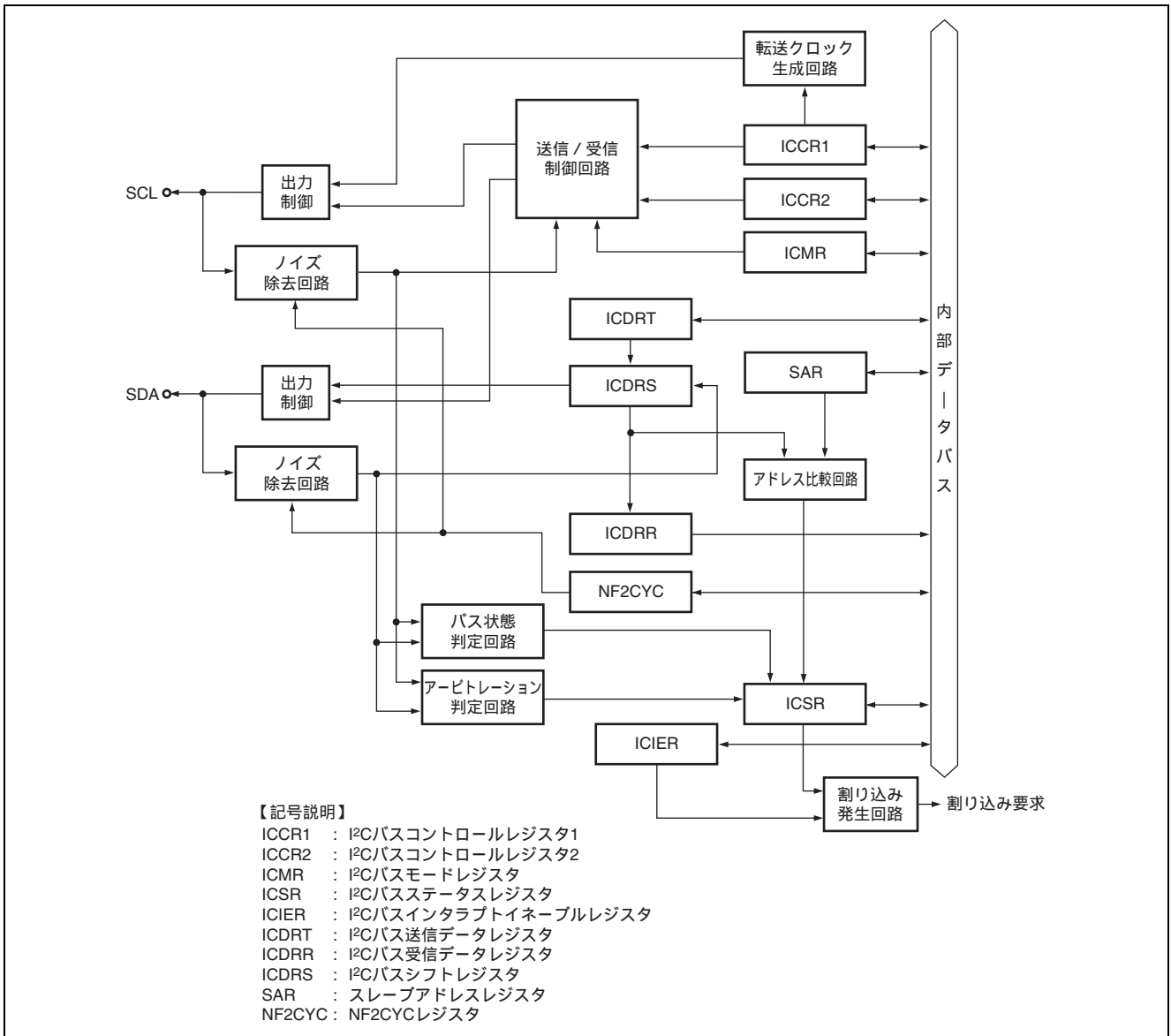


図1 IIC3 のブロック図

2.2 使用機能の設定手順

ここでは、IIC3 の初期設定手順について説明します。転送レートはEEPROMの電気的特性を満足するように設定してください。参考プログラムではPφ/256 を選択しています。図 2にIIC3 の初期設定フロー例を示します。なお、各レジスタ設定の詳細は、「SH7214 グループ、SH7216 グループ ユーザーズマニュアル ハードウェア編」を参照してください。

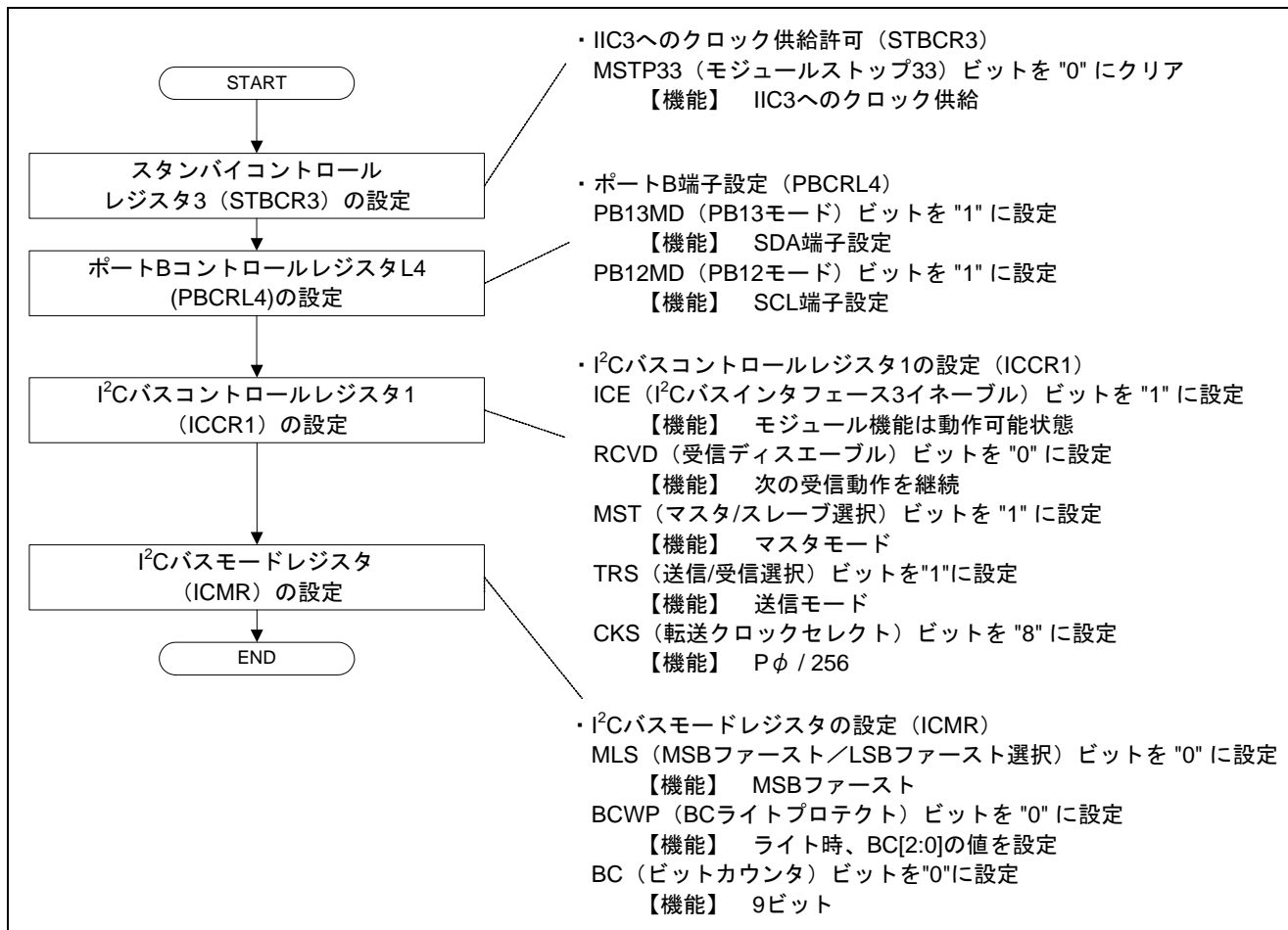


図2 IIC3 の初期設定フロー例

2.3 参考プログラムの動作

参考プログラムでは、IIC3 をマスタ送信モードに設定し、10 バイトのページライトを行います。その後、IIC3 をマスタ受信モードに設定し 10 バイトのシーケンシャルリードを行います。

デバイスコードについては EEPROM のデータシートを確認してください。参考プログラムでは、デバイスコード "B'1010" を使用します。

参考プログラムでは、デバイスアドレス "B'000" を使用します。デバイスアドレスについては、EEPROM のデータシートを確認してください。

メモリアドレスは EEPROM の書き込みまたは読み込み開始アドレスを示し、ライトまたはリードする度に EEPROM 側でアドレスがインクリメントされます。図 3 にページライト動作図、図 4 にシーケンシャルリード動作図を示します。また、図 5 に参考プログラムの動作環境を示します。

参考プログラムは、ルネサス エレクトロニクス製 EEPROM (型名: R1EX24064ASA00A) で動作を確認しています。

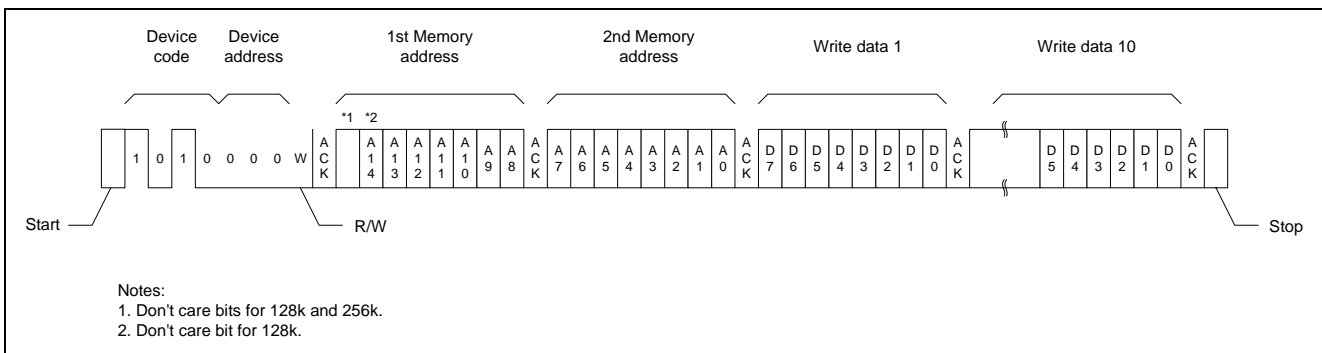


図3 ページライト動作図

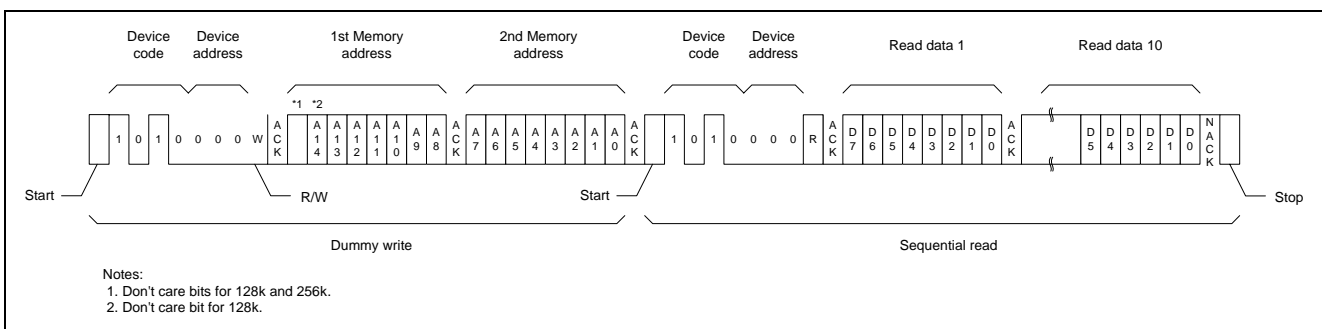


図4 シーケンシャルリード動作図

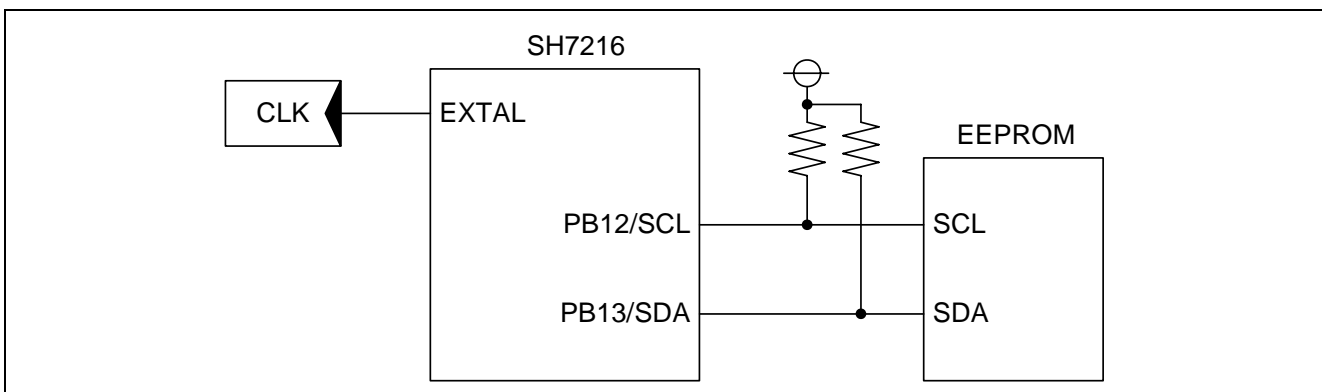


図5 参考プログラムの動作環境

2.4 参考プログラムの処理手順

表 2 に参考プログラムのレジスタ設定を示します。また、表 3 に参考プログラムで使用しているマクロ定義を示します。図 6～図 13 に参考プログラムの処理フローを示します。

表2 参考プログラムのレジスタ設定（初期設定時）

レジスタ名	アドレス	設定値	機能
スタンバイコントロール レジスタ 3 (STBCR3)	H'FFFE 0408	H'00	MSTP33="0": IIC3 は動作
I ² C バスコントロールレジスタ 1 (ICCR1)	H'FFFE E000	H'B8	ICE = "1": SCL/SDA はバス駆動状態 RCVD = "0": 次の受信動作を継続 MST = "1"、TRS = "1": マスタ送信モード CKS = "B'1000": 転送レート Pφ/256
I ² C バスモードレジスタ (ICMR)	H'FFFE E002	H'30	MLS = "0": MSB ファースト BCWP = "0": ライト時、BC の値を設定 BC = "B'000": 9 ビット

表3 参考プログラムで使用しているマクロ定義

マクロ定義	設定値	機能
EEPROM_MEM_ADDR	H'0000	EEPROM 開始アドレス
DEVICE_CODE	H'A0	デバイスコード
DEVICE_ADDR	H'00	デバイスアドレス
IIC_DATA_WR	H'00	ライトコード
IIC_DATA_RD	H'01	リードコード
IIC3_DATA	10	データ転送サイズ
E_OK	0	正常終了
E_ERR	-1	エラー終了

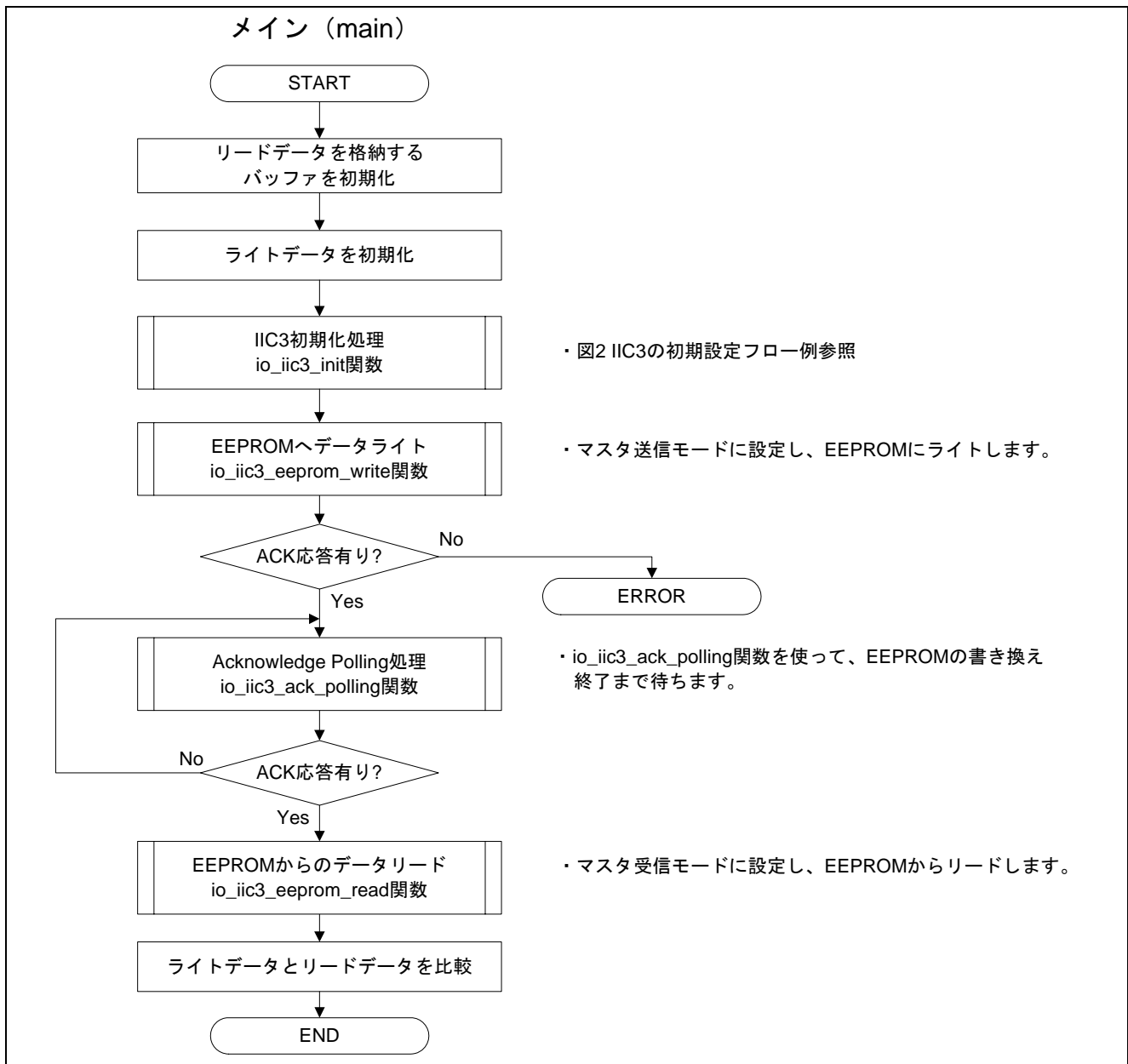


図6 参考プログラムの処理フロー (1)

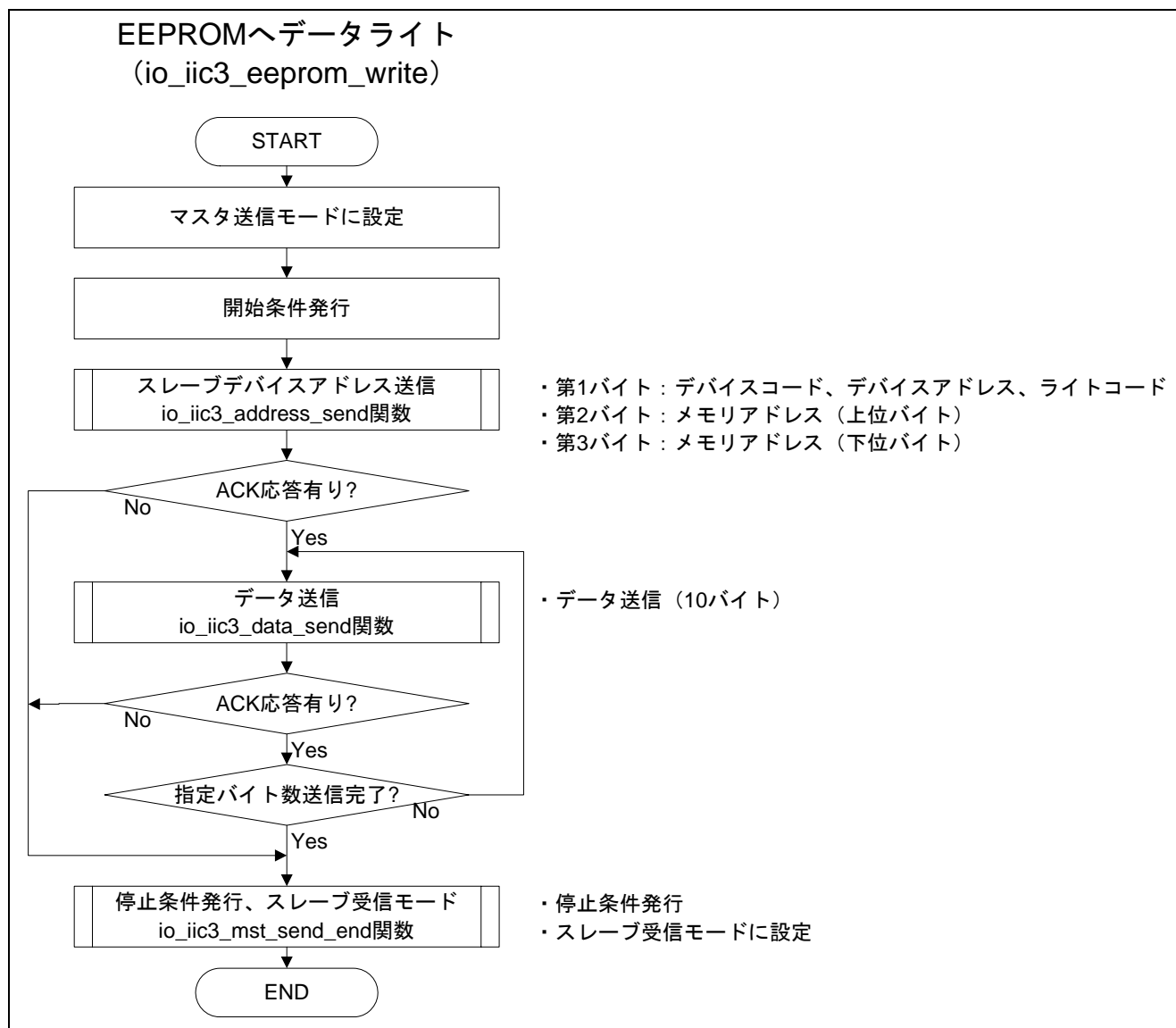


図7 参考プログラムの処理フロー（2）

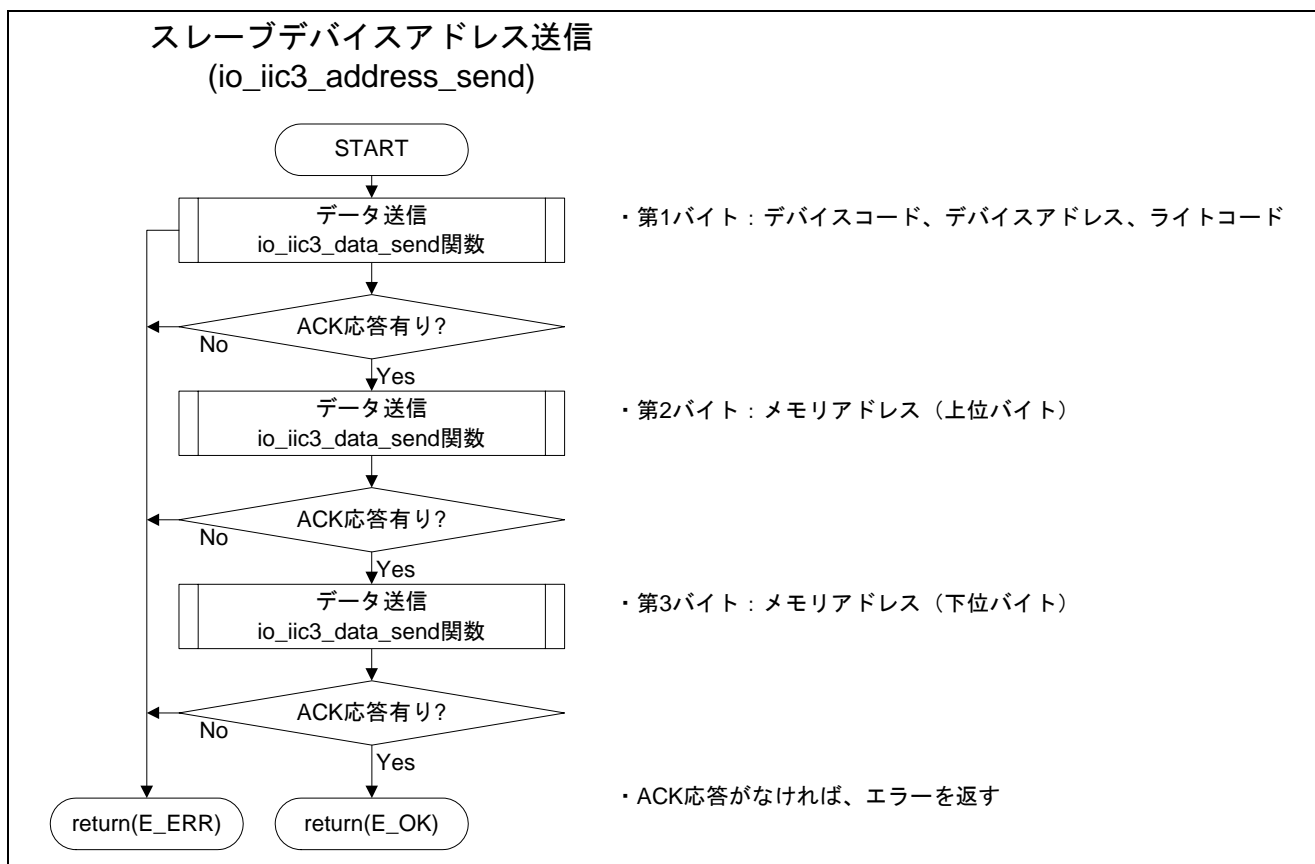


図8 参考プログラムの処理フロー (3)

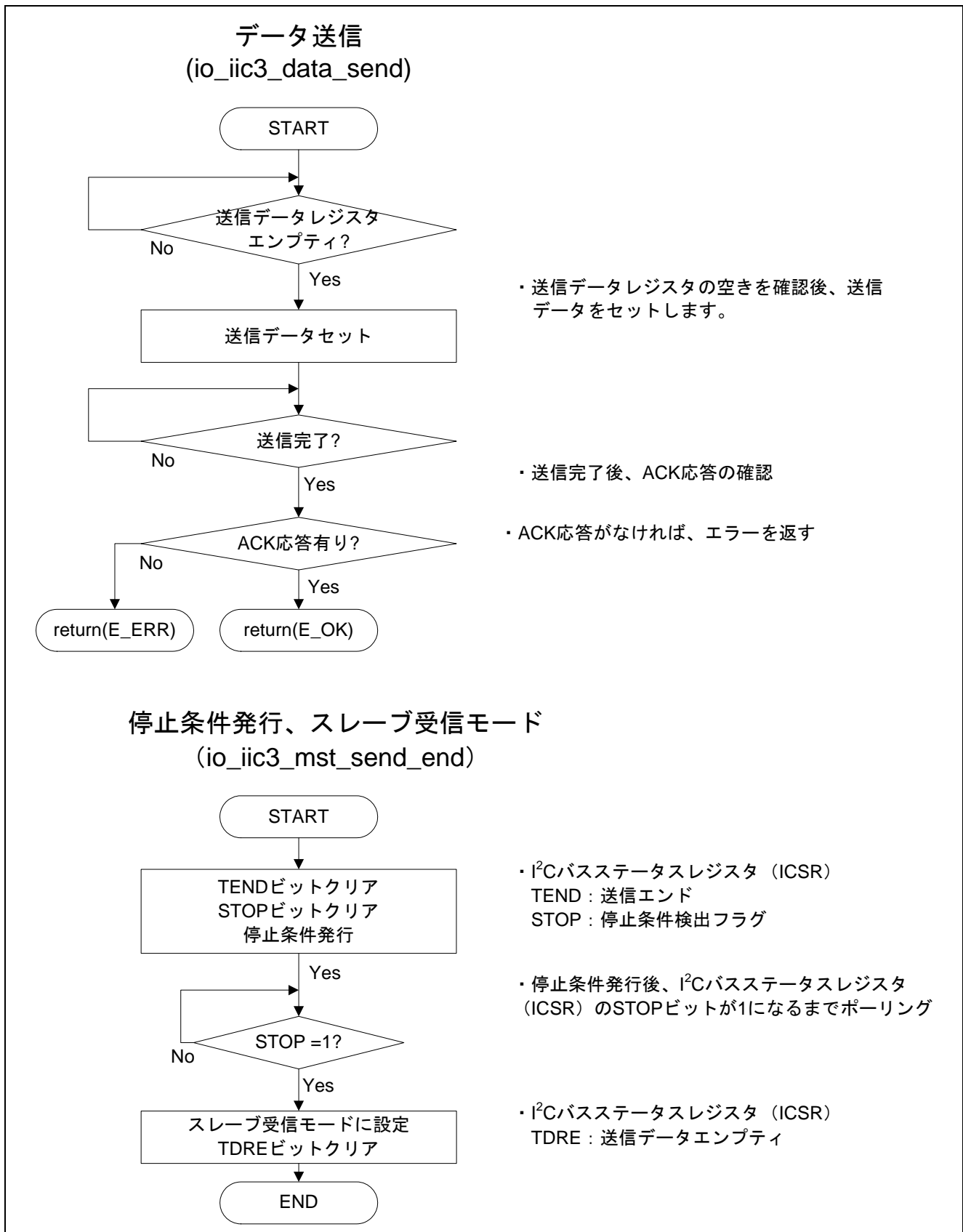


図9 参考プログラムの処理フロー (4)

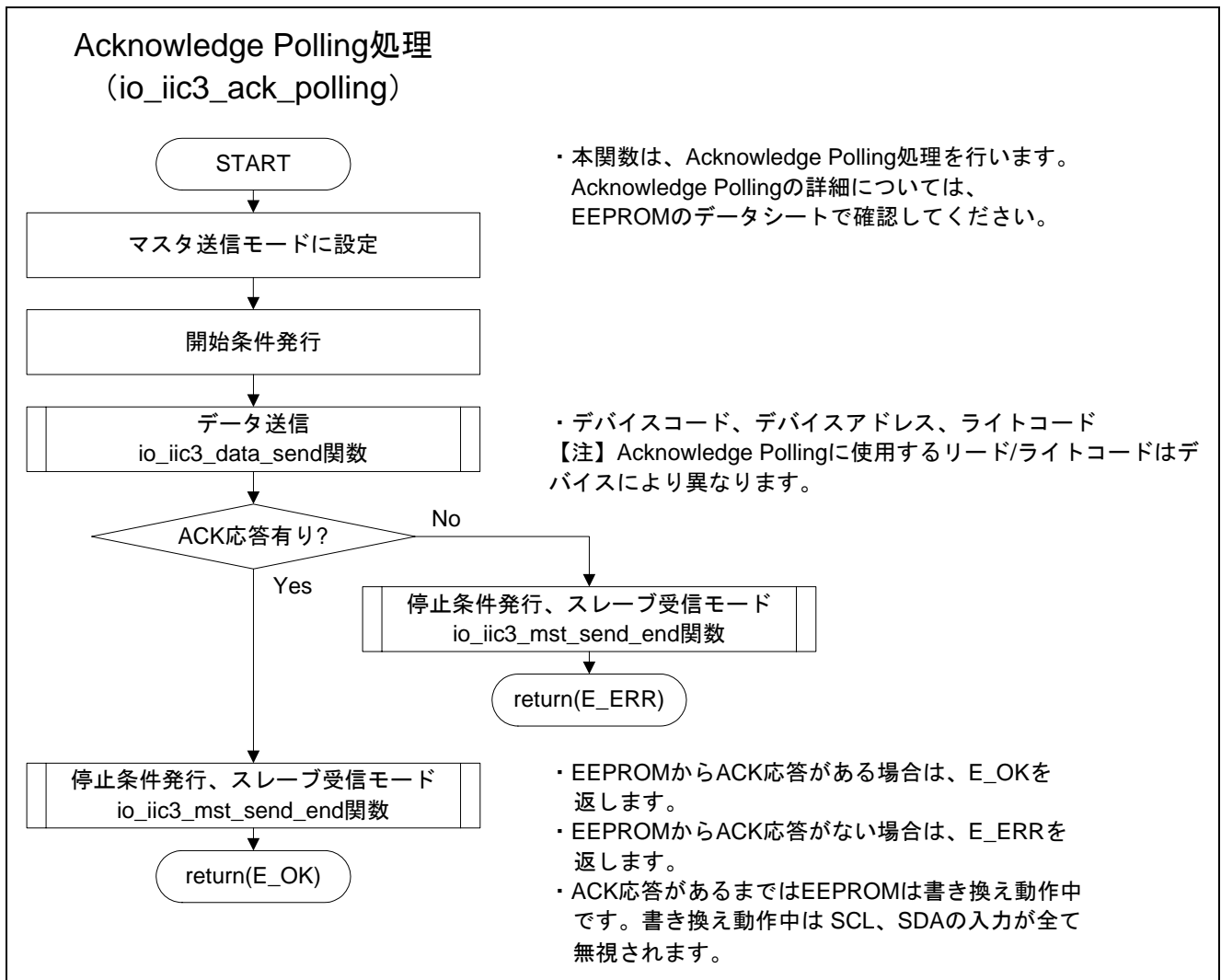


図10 参考プログラムの処理フロー (5)

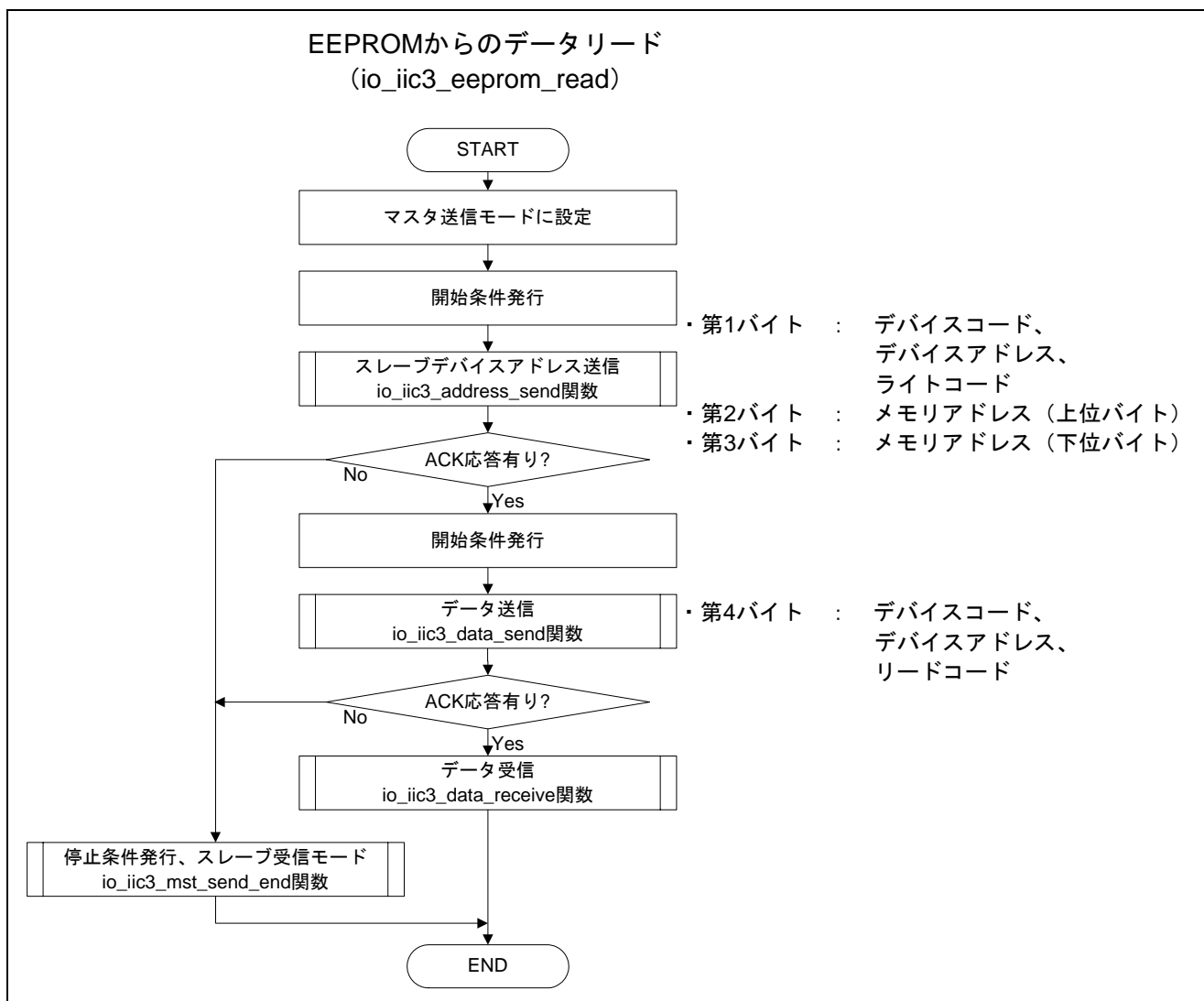
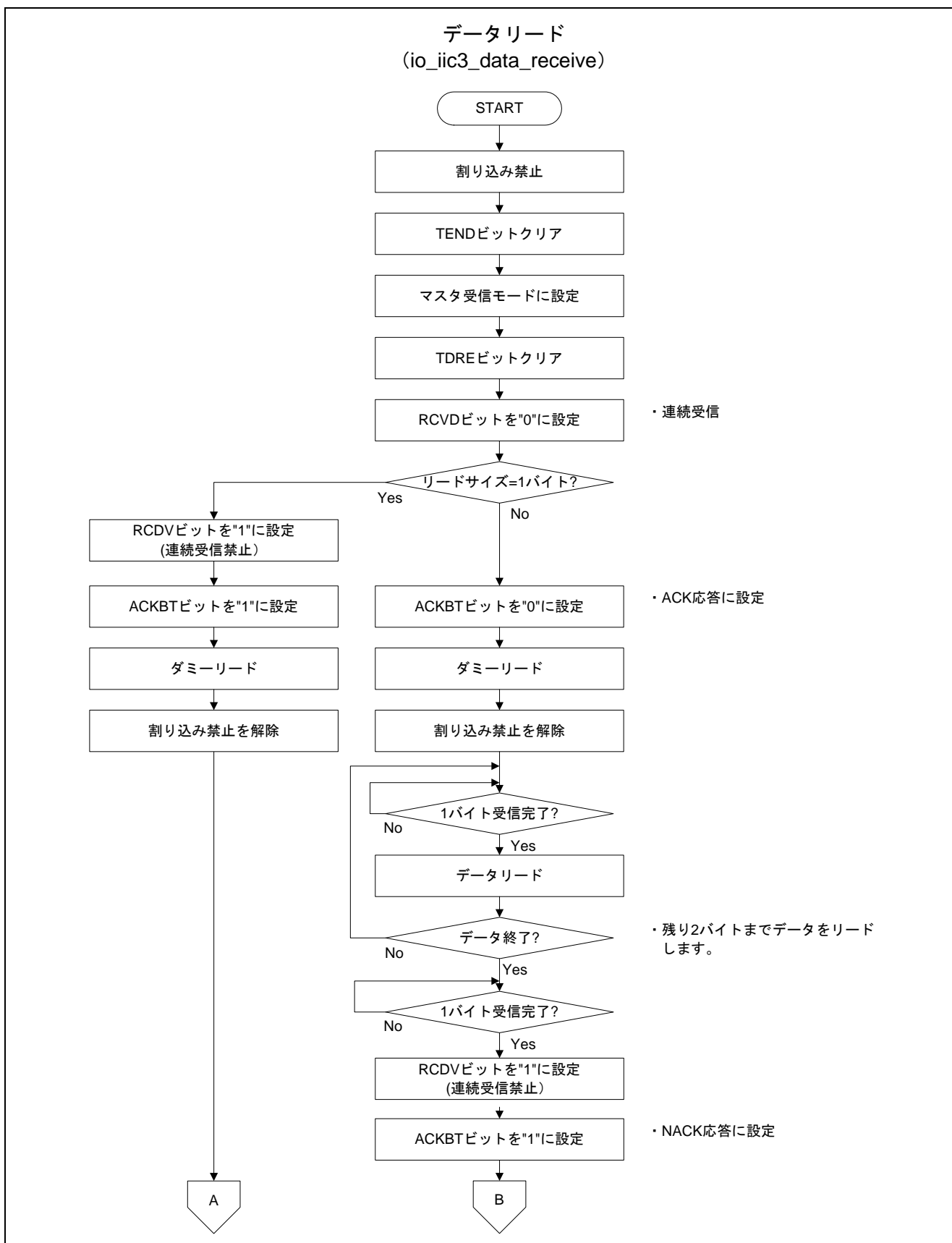


図11 参考プログラムの処理フロー (6)



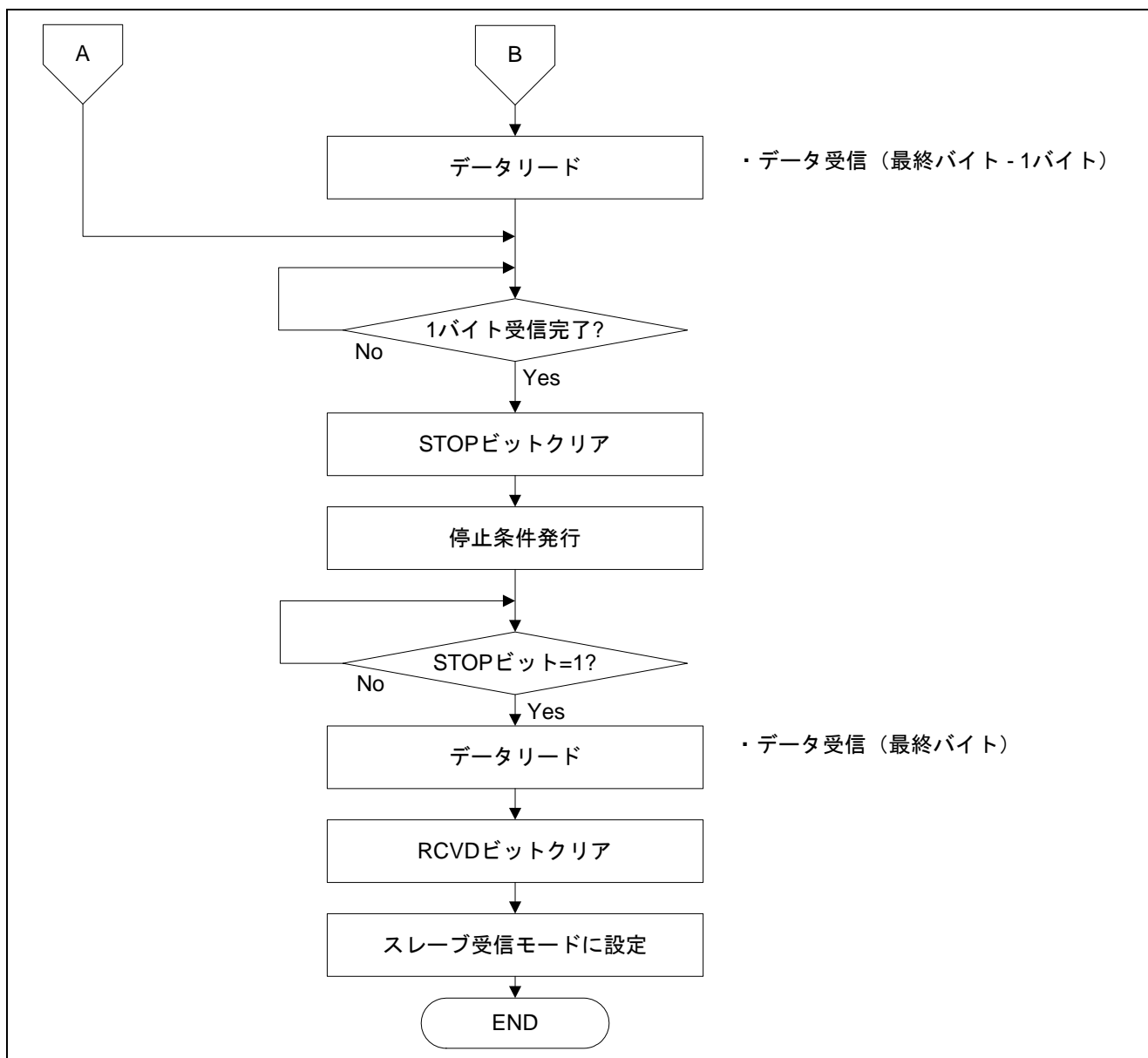


図13 参考プログラムの処理フロー (8)

2.5 マスタ受信モード時の注意

8クロック目の立ち下がり付近でI²Cバス受信データレジスタ (ICDRR) をリードすると、受信データが取得できない場合があります。

また、受信バッファフルかつ8クロック目の立ち下がり付近でI²Cバスコントローラレジスタの受信ディスエーブルビット (RCVD) を1に設定すると、停止条件の発行ができなくなる場合があります。

マスタ受信モードでICDRRをリードする処理は8クロックの立ち上がりまでに行ってください。

2.6 マスタ受信モード、ACKBT 設定時の注意

マスタ受信モード動作時、連続転送している最終データの8つ目のSCLが立ち下がる前にACKBTを設定してください。スレーブ送信側デバイスがオーバーランする可能性があります。

2.7 IICRST 使用上の注意

I²Cバス動作中に、ICCR1レジスタのICEビットに0をライトもしくはICCR2レジスタのIICRSTビットに1をライトすると、ICCR2レジスタのBBSYビットとICSRレジスタのSTOPビットは不定となります。

本注意事項の詳細についてはルネサス テクニカルアップデート (発行番号: TN-MC*-A022A/J) を参照してください。

3. 参考プログラム例

3.1 サンプルプログラムリスト"main.c (1) "

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corp. and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corp. and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
29 *  "FILE COMMENT" ***** Technical reference data *****
30 *  System Name : SH7216 Sample Program
31 *  File Name   : main.c
32 *  Abstract    : IIC3 EEPROM リードライト例
33 *  Version     : 1.00.00
34 *  Device      : SH7216
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              : (Ver.9.03 Release00).
38 *  OS          : None
39 *  H/W Platform: R0K572167 (CPU board)
40 *  Description :
41 *****/
42 *  History     : Jun.30,2010 Ver.1.00.00
43 *  "FILE COMMENT END" *****/
44 #include <machine.h>
45 #include <stdio.h>
46 #include "iodefine.h"
47
```

3.2 サンプルプログラムリスト"main.c (2) "

```
48  /* ==== symbol definition ==== */
49  #define EEPROM_MEM_ADDR 0x0000
50  #define DEVICE_CODE 0xA0 /* EEPROM device code  :b'1010 */
51  #define DEVICE_ADDR 0x00 /* EEPROM device address:b'000 */
52  #define IIC_DATA_WR 0x00 /* Data write code      :b'0 */
53  #define IIC_DATA_RD 0x01 /* Data read code       :b'1 */
54  #define IIC3_DATA 10
55
56  #define E_OK 0
57  #define E_ERR -1
58
59  /* ==== RAM allocation variable declaration ==== */
60  unsigned char ReadData[IIC3_DATA];
61  unsigned char WriteData[IIC3_DATA];
62
63  /* ==== prototype declaration ==== */
64  void main(void);
65  int io_iic3_init(void);
66  int io_iic3_eeprom_read(unsigned char d_code,unsigned char d_adr,unsigned short r_adr,
67                          unsigned int r_size,unsigned char* r_buf);
68  int io_iic3_eeprom_write(unsigned char d_code,unsigned char d_adr,
69                           unsigned short w_adr,unsigned int w_size,unsigned char* w_buf);
70  int io_iic3_address_send(unsigned char* data);
71  int io_iic3_data_receive(unsigned char* r_buf,unsigned int r_size);
72  int io_iic3_data_send(unsigned char data);
73  void io_iic3_mst_send_end(void);
74  int io_iic3_ack_polling(void);
75
```

3.3 サンプルプログラムリスト"main.c (3) "

```

76  /*"FUNC COMMENT"*****
77  * ID      :
78  * Outline : サンプルプログラムメイン
79  *-----
80  * Include :
81  *-----
82  * Declaration : void main(void);
83  *-----
84  * Description : IIC3 のマスタ送信モードを用いて、EEPROM にデータをライトします。
85  *              : IIC3 のマスタ受信モードを用いて、EEPROM からデータをリードします。
86  *-----
87  * Argument  : void
88  *-----
89  * Return Value : void
90  *-----
91  * Note      : None
92  *"FUNC COMMENT END"*****/
93  void main(void)
94  {
95      int i,ack;
96
97      /* ==== データ格納場所クリア ==== */
98      for(i=0;i<IIC3_DATA;i++){
99          ReadData[i] = 0x00;
100     }
101     /* ==== 書き込みデータ作成 ==== */
102     for(i=0;i<IIC3_DATA;i++){
103         WriteData[i] = IIC3_DATA+i;
104     }
105
106     /* ==== IIC3 初期設定 ==== */
107     io_iic3_init();
108
109     /* ==== IIC3 マスタ送信 ==== */
110     ack = io_iic3_eeprom_write(DEVICE_CODE, /* デバイスコード */
111                               DEVICE_ADDR, /* デバイスアドレス */
112                               0x0000, /* 書き込み開始アドレス */
113                               sizeof(WriteData), /* 書き込みデータサイズ */
114                               WriteData); /* データ格納場所 */
115
116     /* ==== 書き込みエラーチェック ==== */
117     if( ack != E_OK ){
118         while(1){
119             /* error */
120         }
121     }
122     /* ==== Acknowledge Polling ==== */
123     while(io_iic3_ack_polling() != E_OK){
124         /* EEPROM の内部書き換えが完了するまで待つ */
125     }
126

```

3.4 サンプルプログラムリスト"main.c (4) "

```
127  /* ==== IIC3 マスタ受信 ==== */
128  io_iic3_eeprom_read(DEVICE_CODE, /* デバイスコード */
129                      DEVICE_ADDR, /* デバイスアドレス */
130                      0x0000, /* 読み出し開始アドレス */
131                      sizeof(ReadData), /* 読み出しデータサイズ */
132                      ReadData); /* データ格納場所 */
133
134  /* ==== 結果比較 ==== */
135  for(i=0; i<IIC3_DATA; i++){
136      if( WriteData[i] != ReadData[i] ){
137          while(1){
138              /* error */
139          }
140      }
141  }
142  while(1){
143      /* Loop */
144  }
145  }
146
```

3.5 サンプルプログラムリスト"main.c (5) "

```

147  /*"FUNC COMMENT"*****
148  * ID      :
149  * Outline : IIC3 モジュール初期設定
150  *-----
151  * Include : iodefne.h
152  *-----
153  * Declaration : int io_iic3_init(void);
154  *-----
155  * Description : IIC3 の初期設定を行います。
156  *-----
157  * Argument   : void
158  *-----
159  * Return Value : E_OK
160  *-----
161  * Note       : None
162  *"FUNC COMMENT END"*****/
163  int io_iic3_init(void)
164  {
165
166      /* ---- STBCR3 ---- */
167      STB.CR3.BIT._IIC3 = 0; /* IIC3 動作 */
168
169      /* ---- PORT ---- */
170      PFC.PBCRL4.BIT.PB13MD = 6; /* SDA select */
171      PFC.PBCRL4.BIT.PB12MD = 6; /* SCL select */
172
173
174      /* ----IIC31 module operation enabled ---- */
175      IIC3.ICCR1.BIT.ICE = 1u; /* IIC3 モジュール動作可能 */
176      IIC3.ICCR1.BIT.RCVD = 0u; /* 次の受信動作を継続 */
177      IIC3.ICCR1.BIT.MST = 1u; /* マスタ選択 */
178      IIC3.ICCR1.BIT.TRIS = 1u; /* 送信選択 */
179      IIC3.ICCR1.BIT.CKS = 8u; /* 転送クロックレート : Pφ /256(195kHz) */
180
181      /* ---IIC bus mode register (ICMR) setting --- */
182      IIC3.ICMR.BYTE = 0x30u;
183          /*
184              bit7      : MLS:0 ----- MSB first
185              bit6      : Reserve:0 ----- Reserve bit
186              bit5-4    : Reserve:1 ----- Reserve bit
187              bit3      : BCWP:0----- Unsetting
188              bit2-0    : BC0:0, BC1:0,BC0:0----- IIC format 9-bit
189          */
190
191      return(E_OK);
192  }
193

```

3.6 サンプルプログラムリスト"main.c (6) "

```

194  /*"FUNC COMMENT"*****
195  * ID      :
196  * Outline : EEPROMからのデータリード
197  *-----
198  * Include : iodefine.h
199  *-----
200  * Declaration : int io_iic3_eeprom_read(unsigned char d_code,
201  *      :      unsigned char d_adr,
202  *      :      unsigned short r_adr,
203  *      :      unsigned int r_size,
204  *      :      unsigned char* r_buf);
205  *-----
206  * Description : デバイスコードd_code、デバイスアドレスd_adrで指定した
207  *      : EEPROMからr_sizeバイト分データを読み出し、r_bufで指定した
208  *      : 領域に読み出したデータを格納します。EEPROMのメモリアドレス
209  *      : はr_adrで指定します。
210  *-----
211  * Argument  : unsigned char d_code : デバイスコード
212  *      : unsigned char d_adr : デバイスアドレス
213  *      : unsigned short r_adr : 読み込み開始アドレス
214  *      : unsigned int r_size : 読み込みデータサイズ
215  *      : unsigned char* r_buf : 読み込みデータ格納先
216  *-----
217  * Return Value : ACK 応答有り : E_OK
218  *      : ACK 応答無し : E_ERR
219  *-----
220  * Note      : None
221  *"FUNC COMMENT END"*****/
222  int io_iic3_eeprom_read(unsigned char d_code,unsigned char d_adr,unsigned short r_adr,
223      unsigned int r_size,unsigned char* r_buf)
224  {
225      int ack = E_OK;
226      unsigned char send[4];
227
228      send[0] = (unsigned char)(d_code|((d_adr & 0x7)<<1)|IIC_DATA_WR);
229      send[1] = (unsigned char)((r_adr>>8) & 0x00ff);
230      send[2] = (unsigned char)(r_adr & 0x00ff);
231      send[3] = (unsigned char)(d_code|((d_adr & 0x7)<<1)|IIC_DATA_RD);
232
233      while(IIC3.ICCR2.BIT.BBSY == 1u){
234          /* バス解放待ち */
235      }
236      IIC3.ICCR1.BYTE |= 0x30u; /* マスタ送信モードに設定 */
237      IIC3.ICCR2.BYTE = ((IIC3.ICCR2.BYTE & 0xbfu)|0x80u); /* 開始条件発行 */
238
239      ack = io_iic3_address_send(send); /* 第1,2,3バイト送信 */
240

```

3.7 サンプルプログラムリスト"main.c (7) "

```
241     if(ack == E_OK){
242         /* 指定したデバイスから ACK 応答有り */
243         IIC3.ICCR2.BYTE=((IIC3.ICCR2.BYTE & 0xbf0) | 0x80); /* 開始条件発行 */
244         ack = io_iic3_data_send(send[3]); /* 第4バイト送信 */
245         if(ack == E_OK){
246             io_iic3_data_receive(r_buf,r_size); /* データ受信 */
247         }
248         else{
249             io_iic3_mst_send_end();
250         }
251     }
252     else{
253         /* 指定したデバイスから ACK 応答無し */
254         io_iic3_mst_send_end();
255     }
256     return(ack);
257 }
258
```

3.8 サンプルプログラムリスト"main.c (8) "

```

259  /*"FUNC COMMENT"*****
260  * ID      :
261  * Outline : マスタ受信モード
262  *-----
263  * Include : #include "iodefine.h"
264  *-----
265  * Declaration : int io_iic3_data_receive(unsigned char* r_buf,
266  *      :      unsigned int r_size);
267  *-----
268  * Description : マスタ受信モードに設定し、r_size で指定したバイト数分の
269  *      :      受信を行い、受信データを r_buf に格納します。
270  *      :      指定数の受信完了後、スレーブ受信モードに切り替えます。
271  *-----
272  * Argument  : unsigned char* r_buf : 読み込みデータ格納先
273  *      :      unsigned int r_size : 読み込みデータサイズ
274  *-----
275  * Return Value : 常に E_OK
276  *-----
277  * Note      : None
278  *"FUNC COMMENT END"*****/
279  int io_iic3_data_receive(unsigned char* r_buf,unsigned int r_size)
280  {
281      int cnt, mask;
282      unsigned char dummy;
283
284      mask = get_imask();
285      set_imask(15);          /* ↓↓↓↓ 割り込み禁止 ↓↓↓↓ */
286
287      /* ==== マスタ受信モード(連続受信)を設定 ==== */
288      IIC3.ICSR.BIT.TEND = 0u;      /* TEND クリア */
289      IIC3.ICCR1.BIT.MST = 1u;      /* マスタモード */
290      IIC3.ICCR1.BIT.TRS = 0u;      /* 受信モード */
291      IIC3.ICSR.BIT.TDRE = 0u;      /* TDRE クリア */
292      IIC3.ICCR1.BIT.RCVD = 0u;     /* 連続受信 */
293
294      /* ==== 受信開始 (1 バイトのみの場合) ==== */
295      if(r_size == 1){              /* 1 バイト受信時 */
296          IIC3.ICCR1.BIT.RCVD = 1u; /* 連続受信禁止 */
297          IIC3.ICIER.BIT.ACKBT = 1u; /* アクノリッジ設定"H" */
298          dummy = IIC3.ICDRR;       /* ダミーリード */
299          set_imask(mask);          /* ↑↑↑↑ 割り込み許可 ↑↑↑↑ */
300      }
301      /* ==== 受信開始 (2 バイト以上の場合) ==== */
302      else{
303          IIC3.ICIER.BIT.ACKBT = 0u; /* アクノリッジ設定"L" */
304          dummy = IIC3.ICDRR;       /* ダミーリード */
305          set_imask(mask);          /* ↑↑↑↑ 割り込み許可 ↑↑↑↑ */
306

```


3.9 サンプルプログラムリスト"main.c (9) "

```
307     /* ==== 2バイト残してデータリード ==== */
308     cnt = r_size;
309     while( cnt > 2 ){
310         /* ---- 1バイト受信完了待ち ---- */
311         while(IIC3.ICSR.BIT.RDRF == 0u){
312             /* wait */
313         }
314         /* ---- データリード ---- */
315         *r_buf++ = IIC3.ICDRR;
316         cnt--;
317     }
318
319     /* ==== 最終バイト-1の受信完了待ち ==== */
320     while(IIC3.ICSR.BIT.RDRF == 0u){
321         /* wait */
322     }
323     /* ==== 最終バイトの受信準備 ==== */
324     IIC3.ICCR1.BIT.RCVF = 1u;          /* 連続受信禁止 */
325     IIC3.ICIER.BIT.ACKBT = 1u;       /* アクノリッジを"H"に設定 */
326
327     /* ==== 最終バイト-1のデータリード ==== */
328     *r_buf++ = IIC3.ICDRR;
329 }
330
331 /* ==== 最終バイト受信完了待ち ==== */
332 while(IIC3.ICSR.BIT.RDRF == 0u){
333     /* wait */
334 }
335
336 /* ==== 停止条件を発行 ==== */
337 /* ---- 発行開始 ---- */
338 IIC3.ICSR.BIT.STOP = 0u;          /* STOP フラグクリア */
339 IIC3.ICCR2.BYTE &= 0x3fu;       /* 停止条件発行 */
340
341 /* ---- 完了待ち ---- */
342 while(IIC3.ICSR.BIT.STOP == 0u){
343     /* wait */
344 }
345
346 /* ==== 最終バイトのデータをリード ==== */
347 *r_buf = IIC3.ICDRR;          /* 最終バイト */
348
349 /* ==== スレーブ受信モードに戻す ==== */
350 IIC3.ICCR1.BIT.RCVF = 0u;       /* RCVF をクリア */
351 IIC3.ICCR1.BYTE &= 0xcfu;       /* スレーブ受信モード */
352
353 return(E_OK);
354 }
355
```

3.10 サンプルプログラムリスト "main.c (10) "

```

356  /*"FUNC COMMENT"*****
357  * ID      :
358  * Outline : EEPROM へのデータライト
359  *-----
360  * Include : iodef.h
361  *-----
362  * Declaration : int io_iic3_eeprom_write(unsigned char d_code,
363  *      :      unsigned char d_adr,
364  *      :      unsigned short w_adr,
365  *      :      unsigned int w_size,
366  *      :      unsigned char* w_buf);
367  *-----
368  * Description : デバイスコード d_code、デバイスアドレス d_adr で指定した EEPROM へ
369  *      : w_buf で指定した領域のデータを w_size バイト分書き込みます。
370  *      : EEPROM のメモリアドレスは w_adr で指定します。
371  *-----
372  * Argument   : unsigned char d_code ; I : デバイスコード
373  *      : unsigned char d_adr ; I : デバイスアドレス
374  *      : unsigned short w_adr ; I : 書き込み開始アドレス
375  *      : unsigned int w_size ; I : 書き込みデータサイズ
376  *      : unsigned char* w_buf ; O : 書き込みデータ格納先
377  *-----
378  * Return Value : ACK 応答有り : E_OK
379  *      : ACK 応答無し : E_ERR
380  *-----
381  * Note       : None
382  *"FUNC COMMENT END"*****
383  int io_iic3_eeprom_write(unsigned char d_code,unsigned char d_adr,unsigned short w_adr,
384      unsigned int w_size,unsigned char* w_buf)
385  {
386      int ack = E_OK;
387      int i;
388      unsigned char send[3];
389
390      send[0] = (unsigned char)(d_code|((d_adr & 0x7)<<1)|IIC_DATA_WR);
391      send[1] = (unsigned char)((w_adr>>8) & 0x00ff);
392      send[2] = (unsigned char)(w_adr & 0x00ff);
393
394      while(IIC3.ICCR2.BIT.BBSY == 1u){
395          /* バス解放待ち */
396      }
397      IIC3.ICCR1.BYTE |= 0x30u; /* マスタ送信モードに設定 */
398      IIC3.ICCR2.BYTE = ((IIC3.ICCR2.BYTE & 0xbfu)|0x80u); /* 開始条件発行 */
399
400      ack = io_iic3_address_send(send); /* 第 1,2,3 バイト送信 */
401

```

3.11 サンプルプログラムリスト "main.c (11) "

```

402     if(ack == E_OK){
403         /* 指定したデバイスから ACK 応答有り */
404         for(i=0;i<w_size;i++){
405             ack = io_iic3_data_send(*w_buf++);          /* データ送信 */
406             if(ack == E_ERR){
407                 break;
408             }
409         }
410         io_iic3_mst_send_end();
411     }
412     else{
413         /* 指定したデバイスから ACK 応答無し */
414         io_iic3_mst_send_end();
415     }
416     return(ack);
417 }
418
419 /*"FUNC COMMENT"*****
420 * ID          :
421 * Outline     : スレーブデバイスアドレス送信
422 *-----
423 * Include     :
424 *-----
425 * Declaration : int io_iic3_address_send(unsigned char* data);
426 *-----
427 * Description : data で指定したスレーブデバイスのアドレス (1 バイト)
428 *              : およびメモリアドレス (2 バイト) の送信を行います。
429 *-----
430 * Argument    : unsigned char* data ; I : 送信データ
431 *-----
432 * Return Value : ACK 応答有り : E_OK
433 *              : ACK 応答無し : E_ERR
434 *-----
435 * Note        : None
436 *"FUNC COMMENT END"*****
437 int io_iic3_address_send(unsigned char* data)
438 {
439     int ack;
440
441     ack = io_iic3_data_send(*data++);          /* スレーブデバイスのアドレス */
442     if(ack == E_ERR){
443         return(ack);
444     }
445     ack = io_iic3_data_send(*data++);          /* 1st メモリアドレス */
446     if(ack == E_ERR){
447         return(ack);
448     }
449     ack = io_iic3_data_send(*data);           /* 2nd メモリアドレス */
450     if(ack == E_ERR){
451         return(ack);
452     }
453     return(ack);
454 }

```

3.12 サンプルプログラムリスト"main.c (12) "

```
455
456 /*"FUNC COMMENT"*****
457 * ID      :
458 * Outline : 1バイトデータ送信
459 *-----
460 * Include : iodef.h
461 *-----
462 * Declaration : int io_iic3_data_send(unsigned char data);
463 *-----
464 * Description : dataを送信します。処理手順は以下の通りです。
465 *              : 1.ICDRT エンプティ待ち
466 *              : 2.送信データセット
467 *              : 3.送信完了確認
468 *              : 4.ACK 応答確認
469 *-----
470 * Argument  : unsigned char data : 送信データ
471 *-----
472 * Return Value : ACK 応答有り : E_OK
473 *              : ACK 応答無し : E_ERR
474 *-----
475 * Note      : None
476 *"FUNC COMMENT END"*****/
477 int io_iic3_data_send(unsigned char data)
478 {
479     int ack;
480
481     while(IIC3.ICSR.BIT.TDRE == 0u){
482         /* ICDRT のエンプティ待ち */
483     }
484     IIC3.ICDRT = data;
485
486     while(IIC3.ICSR.BIT.TEND == 0u){
487         /* 送信完了待ち */
488     }
489     if(IIC3.ICIER.BIT.ACKBR == 0u){
490         ack = E_OK;
491     }
492     else{
493         ack = E_ERR;
494     }
495     return(ack);
496 }
497
```

3.13 サンプルプログラムリスト"main.c (13) "

```
498  /*"FUNC COMMENT"*****
499  * ID      :
500  * Outline : 停止条件発行
501  *-----
502  * Include : iodef.h
503  *-----
504  * Declaration : void io_iic3_mst_send_end(void);
505  *-----
506  * Description : 停止条件を発行し、スレーブ受信モードに切り替えます。
507  *-----
508  * Argument  : void
509  *-----
510  * Return Value : void
511  *-----
512  * Note      : None
513  *"FUNC COMMENT END"*****/
514 void io_iic3_mst_send_end(void)
515 {
516     IIC3.ICSR.BIT.TEND = 0u; /* TEND フラグクリア */
517     IIC3.ICSR.BIT.STOP = 0u; /* STOP フラグクリア */
518     IIC3.ICCR2.BYTE &= 0x3fu; /* 停止条件発行 */
519
520     while(IIC3.ICSR.BIT.STOP == 0u){
521         /* バス解放待ち */
522     }
523
524     IIC3.ICCR1.BYTE &= 0xcfu; /* スレーブ受信モード */
525     IIC3.ICSR.BIT.TDRE = 0u; /* TDRE をクリア */
526 }
527
```

3.14 サンプルプログラムリスト"main.c (14) "

```
528  /*"FUNC COMMENT"*****
529  * ID      :
530  * Outline : Acknowledge Polling
531  *-----
532  * Include : iodef.h
533  *-----
534  * Declaration : io_iic3_ack_polling
535  *-----
536  * Description : 本関数は、EEPROM が書き換え中か否かを判定します。
537  *              : EEPROM は、内部で書き換え中の場合入力コマンドを無視し、
538  *              : ACK を返しません。本関数で EEPROM が書き換え中ではないこと
539  *              : を確認してから EEPROM にアクセスしてください。
540  *              : Acknowledge Polling 時に送信する Read/Write コードは、
541  *              : 使用するデバイスにより異なります。使用する EEPROM の
542  *              : データシートで確認してください。
543  *-----
544  * Argument : void
545  *-----
546  * Return Value : E_OK : NOT_BUSY
547  *              : E_ERR : BUSY (EEPROM は書き換え中)
548  *-----
549  * Note      : None
550  *"FUNC COMMENT END"*****/
551  int io_iic3_ack_polling(void)
552  {
553      int ack = E_OK;
554      unsigned char send = (unsigned char)(DEVICE_CODE|((DEVICE_ADDR & 0x7)<<1)|IIC_DATA_WR);
555
556      while(IIC3.ICCR2.BIT.BBSY == 1u){
557          /* バス解放待ち */
558      }
559      IIC3.ICCR1.BYTE |= 0x30u; /* マスタ送信モードに設定 */
560      IIC3.ICCR2.BYTE = ((IIC3.ICCR2.BYTE & 0xbf0)|0x80u); /* 開始条件発行 */
561
562      ack = io_iic3_data_send(send);
563
564      io_iic3_mst_send_end(); /* 停止条件発行 */
565
566      return(ack);
567  }
568  /* End of File */
569
```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A、SH2A-FPU ソフトウェアマニュアル Rev.3.00
(最新版をルネサスエレクトロニクスのホームページから入手してください。)
- ハードウェアマニュアル
SH7214 グループ、SH7216 グループ ユーザーズマニュアル ハードウェア編 Rev.2.00
(最新版をルネサスエレクトロニクスの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/inquiry>

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.08.20	—	初版発行
1.01	2012.02.10	16	マスタ受信モード時の注意の誤記修正 正) 8クロックの立ち上がりまでに行ってください 誤) 8クロックの立ち下がりまでに行ってください

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

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

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

(03)5201-5307

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