

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ひ申し上げます。

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、誤りが無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。

SH7262/SH7264 グループ

I²C バスインタフェース 3 シングルマスタ送信 (EEPROM のライト)

要旨

本アプリケーションノートは、SH7262/SH7264のI²Cバスインタフェース3 (IIC3) のシングルマスタ送信を使用してEEPROMにライトアクセスする例について説明します。

動作確認デバイス

SH7262/SH7264

以下、総称して「SH7264」として説明します。

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.....	13
4. 参考ドキュメント.....	23

1. はじめに

1.1 仕様

- マスタデバイスを SH7264、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。
- 転送レートを 391kHz 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 3 (IIC3)

1.3 適用条件

マイコン	SH7262/SH7264
動作周波数	内部クロック : 144 MHz バスクロック : 72 MHz 周辺クロック : 36 MHz
統合開発環境	ルネサステクノロジ製 High-performance Embedded Workshop Ver.4.04.01
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.9.02 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 -object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)
EEPROM	HN58X24128FPIE(128k bit)

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。

- SH7262/SH7264 グループ 初期設定例
- SH7262/SH7264 グループ I²C バスインタフェース 3 シングルマスタ受信 (EEPROM のリード)

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 SH7264 からスレーブデバイスの EEPROM にデータ転送を行います。

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。

SH7264 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 のを示します。

IIC3 についての詳細は、「SH7262 グループ、SH7264 グループグループ ハードウェアマニュアル 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、停止条件の自動生成 • 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) 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) を起動させてデータの転送を行うことができます。

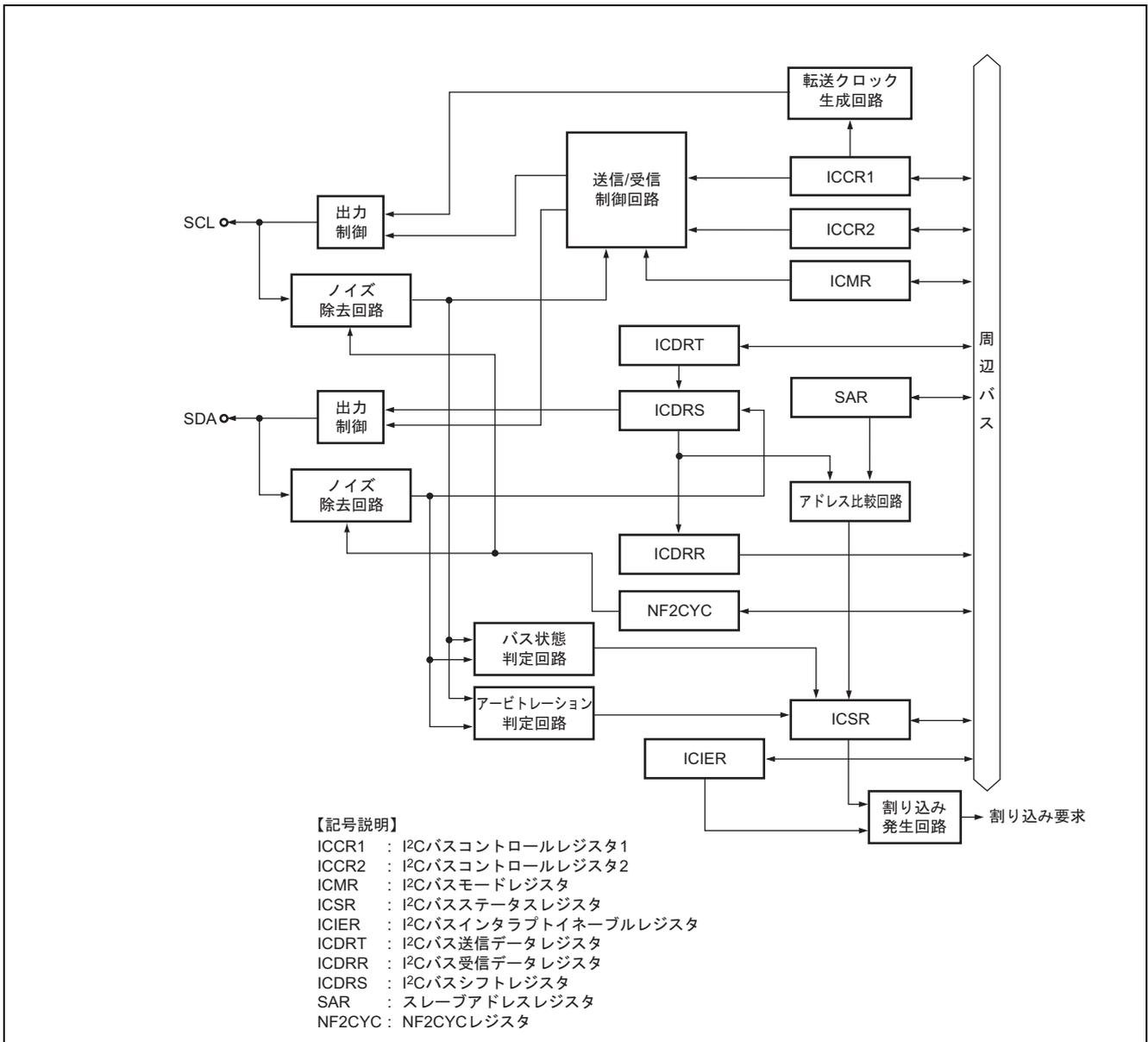


図1 IIC3のブロック図

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。転送レートは外部仕様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/92 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、「SH7262 グループ、SH7264 グループ ハードウェアマニュアル」を参照してください。

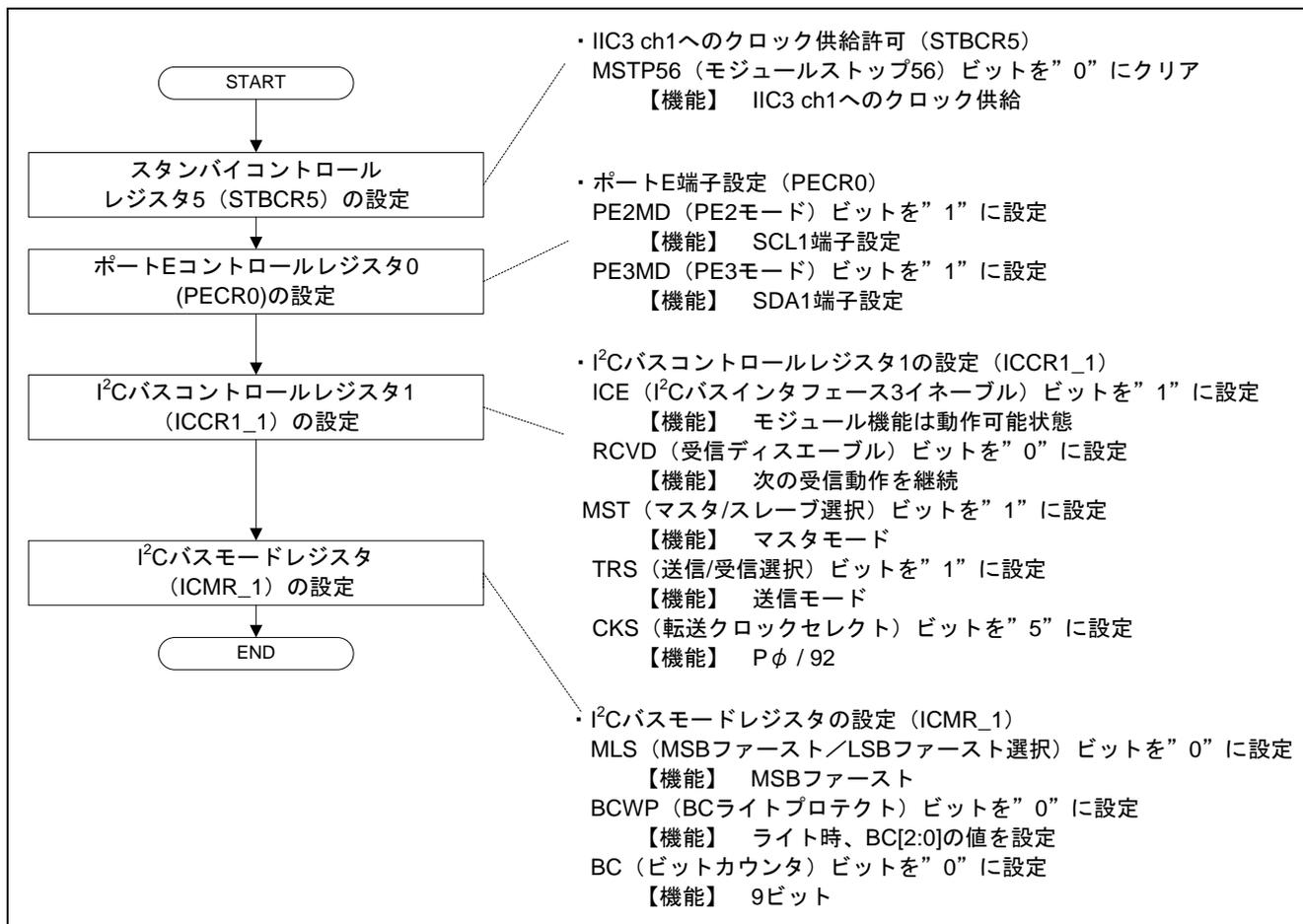


図2 IIC3 の初期設定フロー例

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。

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 の書き込み開始アドレスを示し、ライトする度に 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を、図 4 に 参考プログラムの動作環境を示します。

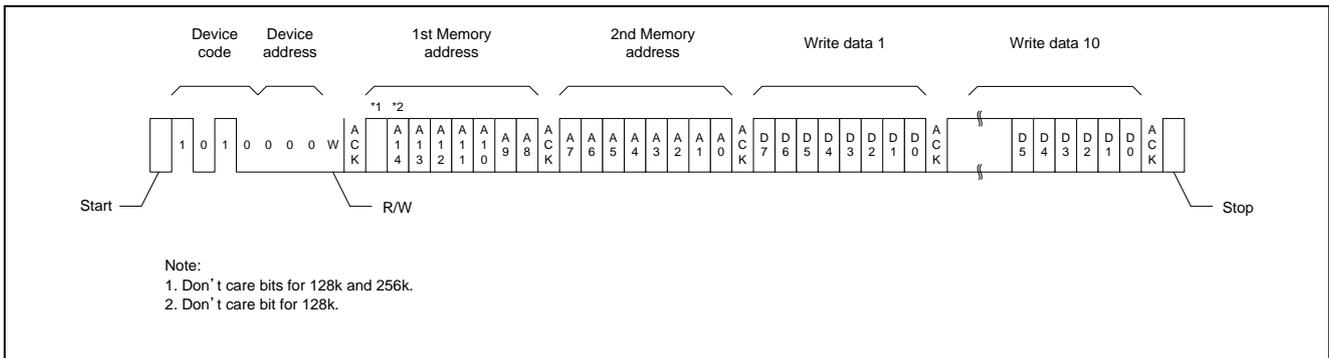


図3 ページライト動作図

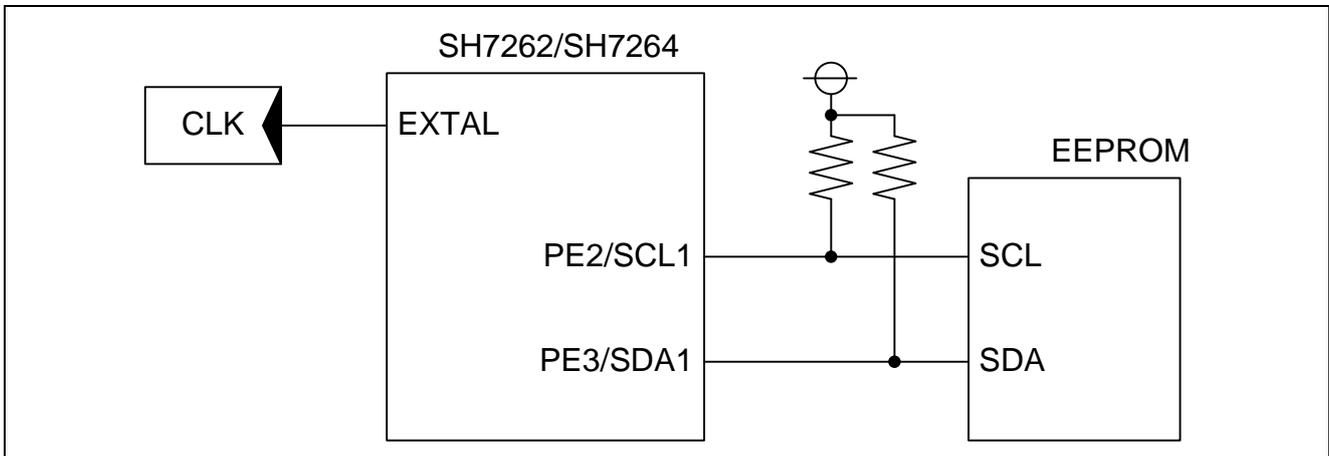


図4 参考プログラムの動作環境

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 5～図 9 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 5 (STBCR5)	H'FFFE 0410	H'00	MSTP56="0": IIC3 の ch1 は動作
I ² C バスコントロールレジスタ 1 (ICCR1_1)	H'FFFE E400	H'B5	ICE = "1": SCL/SDA はバス駆動状態 RCVD = "0": 次の受信動作継続 MST = "1"、TRS = "1": マスタ送信モード CKS = "B'0101": 転送レート Pφ/92
I ² C バスモードレジスタ (ICMR_1)	H'FFFE E402	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

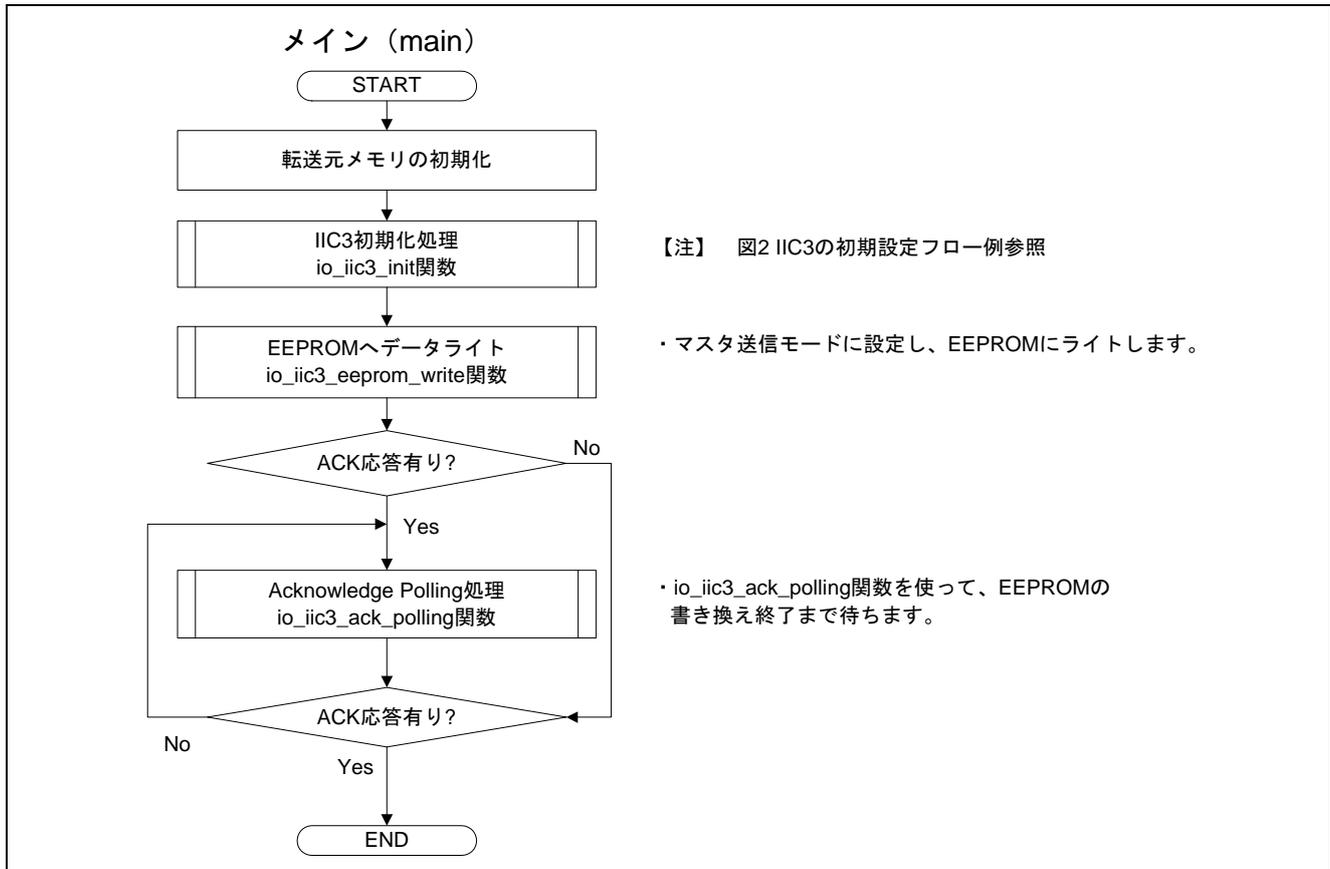


図5 参考プログラムの処理フロー (1)

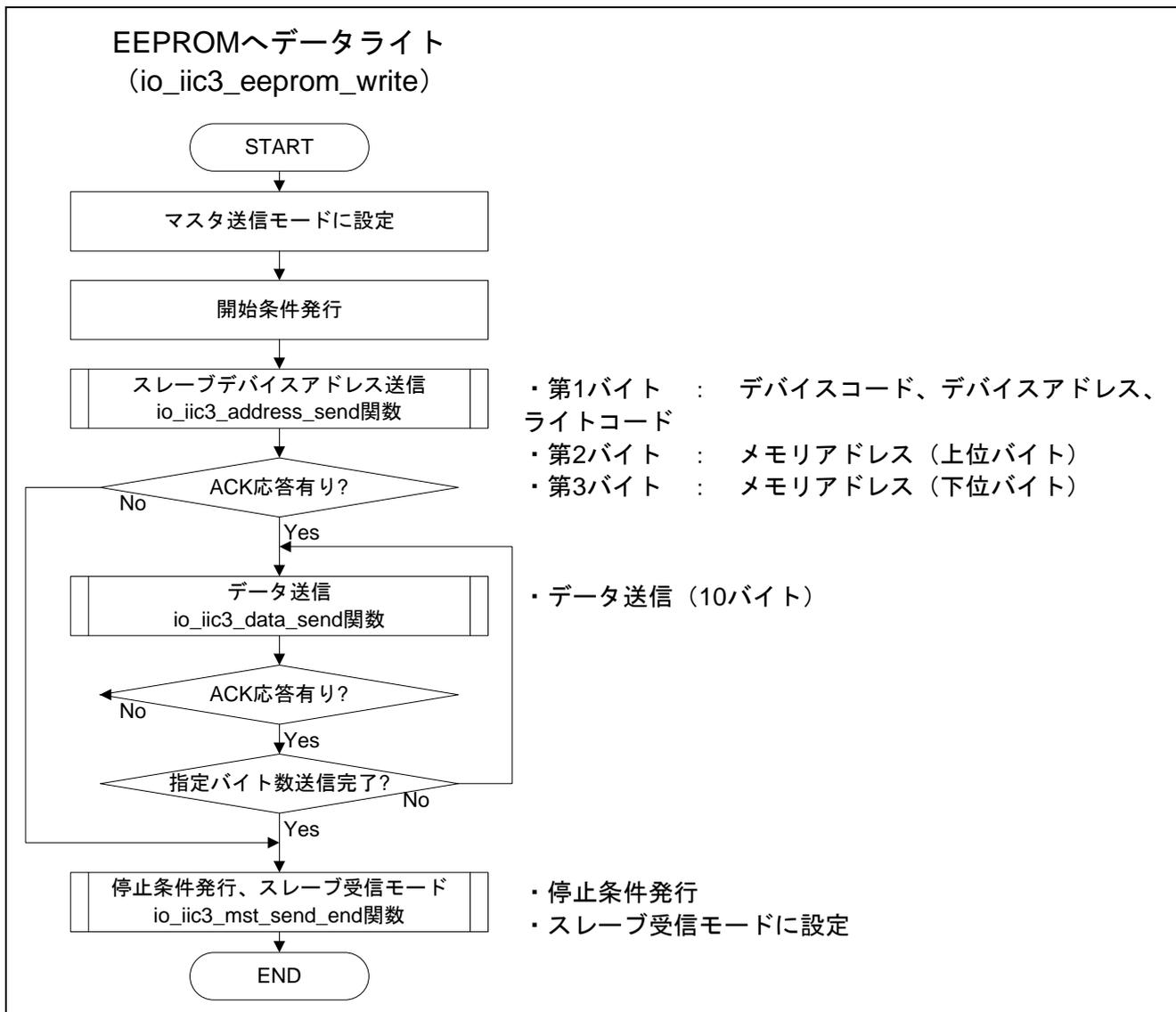


図6 参考プログラムの処理フロー (2)

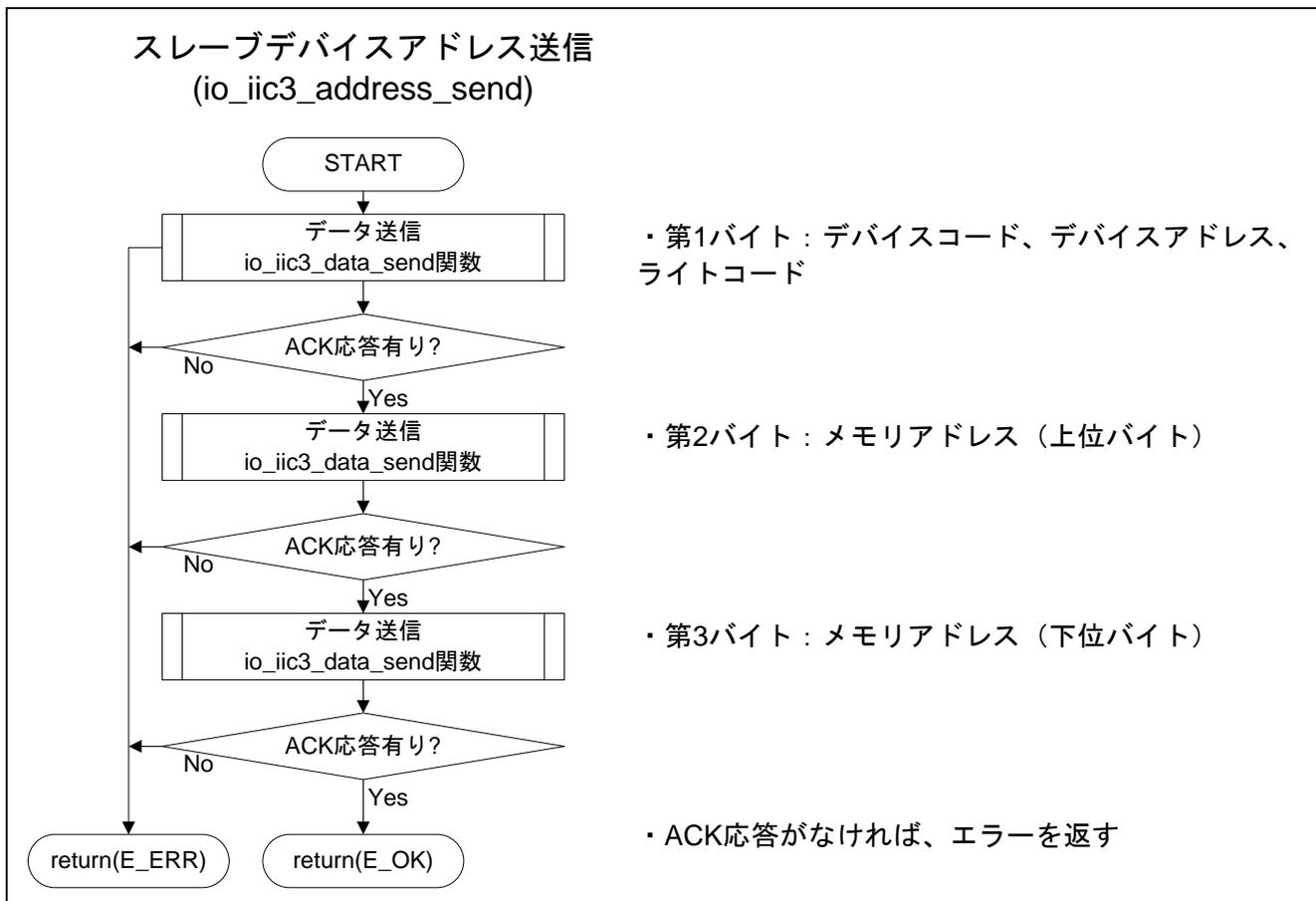


図7 参考プログラムの処理フロー (3)

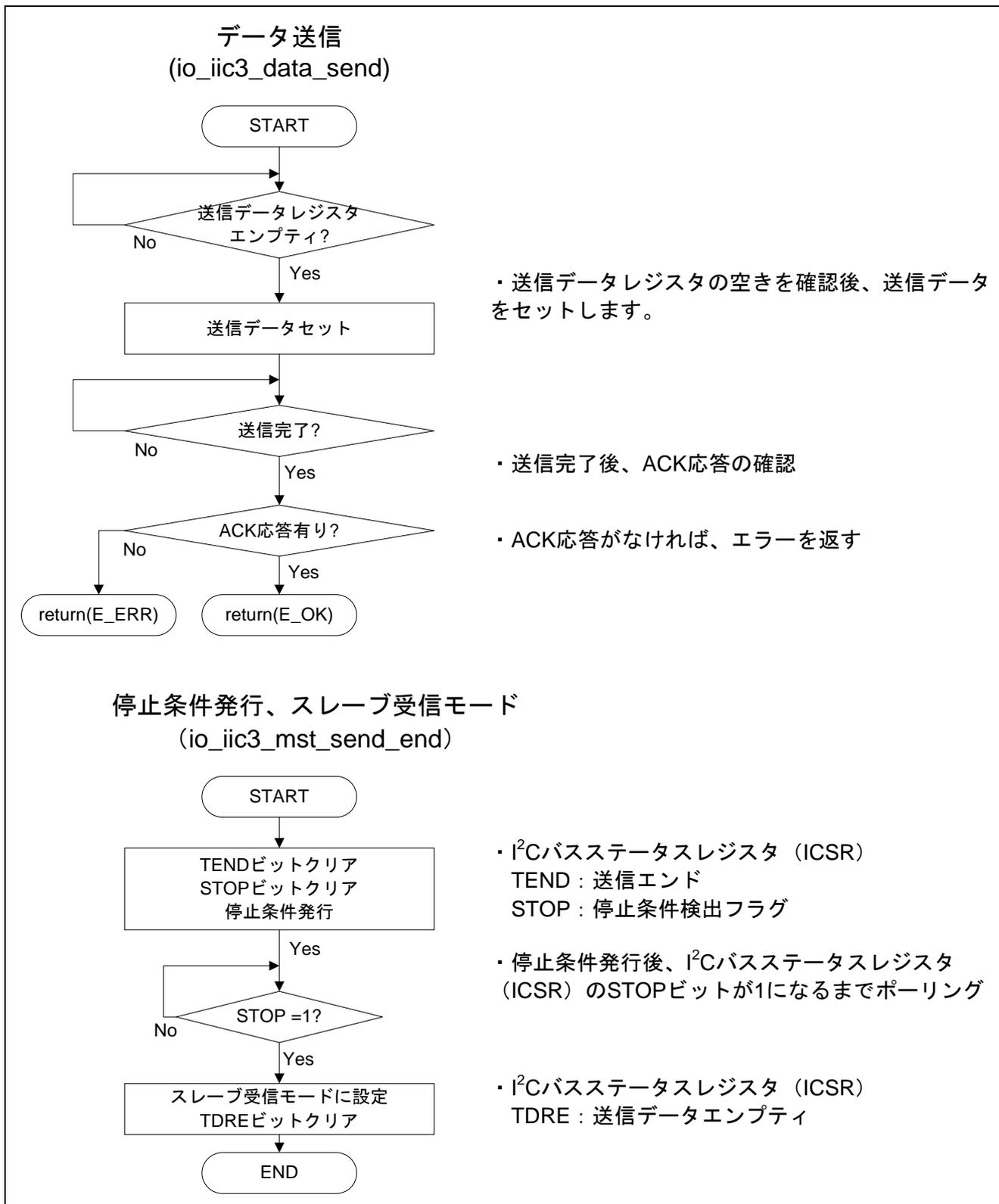


図8 参考プログラムの処理フロー (4)

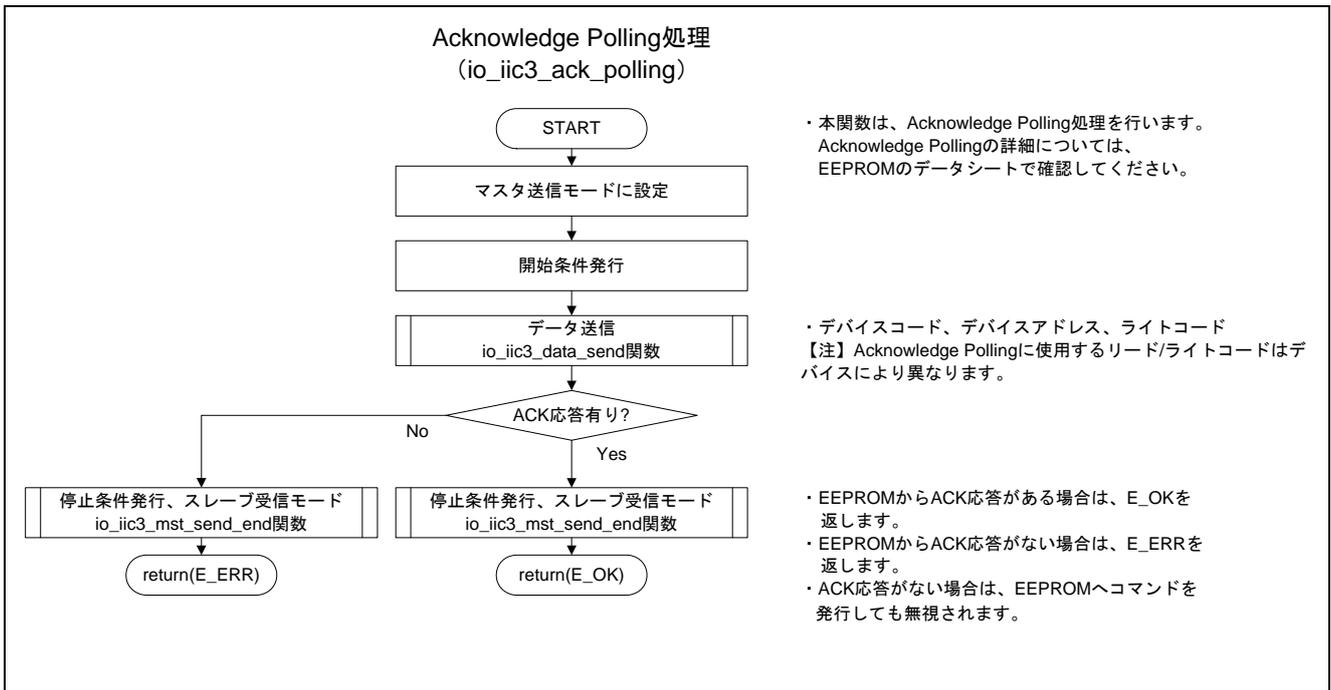


図9 参考プログラムの処理フロー (5)

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  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *      System Name : SH7264 Sample Program
4  *      File Name   : main.c
5  *      Abstract    : IIC3 マスタ送信モードサンプルプログラム
6  *      Version     : 1.00.00
7  *      Device      : SH7264/SH7262
8  *      Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9  *                  : C/C++ compiler package for the SuperH RISC engine family
10 *                  :                               (Ver.9.02 Release00).
11 *      OS          : None
12 *      H/W Platform: M3A-HS64G50(CPU board)
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 *
25 *      Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
26 *      AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *      History     : Jan.23,2009 Ver.1.00.00
29 *"FILE COMMENT END"*****
30 #include <machine.h>
31 #include "iodefine.h"      /* SH7264 iodefine */
32
33 /* ==== symbol definition ==== */
34 #define EEPROM_MEM_ADDR 0x0000
35 #define DEVICE_CODE 0xA0 /* EEPROM device code :b'1010 */
36 #define DEVICE_ADDR 0x00 /* EEPROM device address:b'000 */
37 #define IIC_DATA_WR 0x00 /* Data write code :b'0 */
38 #define IIC_DATA_RD 0x01 /* Data read code :b'1 */
39 #define IIC3_DATA 10
40
41 #define E_OK 0
42 #define E_ERR -1
43
44 /* ==== RAM allocation variable declaration ==== */
45 unsigned char WriteData[IIC3_DATA];
46

```

3.2 サンプルプログラムリスト"main.c (2) "

```

47  /* ==== prototype declaration ==== */
48  void main(void);
49  int io_iic3_eeprom_write(unsigned char d_code,unsigned char d_adr,
50                          unsigned short w_adr,unsigned int w_size,unsigned char* w_buf);
51  int io_iic3_data_send(unsigned char data);
52  int io_iic3_address_send(unsigned char* data);
53  void io_iic3_mst_send_end(void);
54  int io_iic3_init(void);
55  int io_iic3_ack_polling(void);
56

```

3.3 サンプルプログラムリスト"main.c (3) "

```

57  /*"FUNC COMMENT"*****
58  * ID      :
59  * Outline : サンプルプログラムメイン
60  *-----
61  * Include :
62  *-----
63  * Declaration : void main(void);
64  *-----
65  * Description : IIC3 のマスタ送信モードを用いて、EEPROM にデータを送信します。
66  *-----
67  * Argument  : void
68  *-----
69  * Return Value: void
70  *"FUNC COMMENT END"*****/
71 void main(void)
72 {
73     int i,ack;
74
75     /* ==== 書き込みデータ作成 ==== */
76     for(i=0;i<IIC3_DATA;i++){
77         WriteData[i] = IIC3_DATA+i;
78     }
79
80     /* ==== IIC3 初期設定 ==== */
81     io_iic3_init();
82
83     /* ==== IIC3 マスタ送信 ==== */
84     ack = io_iic3_eeeprom_write(DEVICE_CODE, /* デバイスコード */
85                                DEVICE_ADDR, /* デバイスアドレス */
86                                0x0000, /* 書き込み開始アドレス */
87                                sizeof(WriteData), /* 書き込みデータサイズ */
88                                WriteData); /* データ格納場所 */
89
90     if( ack == E_OK ){
91         /* ==== Acknowledge Polling ==== */
92         while(io_iic3_ack_polling() != E_OK){
93             /* EEPROM の内部書き換えが完了するまで待つ */
94         }
95     }
96
97     while(1){
98         /* Loop */
99     }
100 }
101

```

3.4 サンプルプログラムリスト"main.c (4) "

```

102  /*"FUNC COMMENT"*****
103  * ID      :
104  * Outline : IIC3 モジュール初期設定
105  *-----
106  * Include : #include "iodefine.h"
107  *-----
108  * Declaration : int io_iic3_init(void);
109  *-----
110  * Description : 本関数は、IIC3 のチャンネル 1 の初期設定を行っています。
111  *-----
112  * Argument  : void
113  *-----
114  * Return Value: E_OK
115  *"FUNC COMMENT END"*****/
116  int io_iic3_init(void)
117  {
118
119      /* ---- STBCR5 ---- */
120      CPG.STBCR5.BIT.MSTP56 = 0; /* IIC3 チャンネル 1 動作 */
121
122      /* ---- PORT ---- */
123      PORT.PECR0.BIT.PE2MD = 0x01; /* SCL1 select */
124      PORT.PECR0.BIT.PE3MD = 0x01; /* SDA1 select */
125
126
127      /* ----IIC31 module operation enabled ---- */
128      IIC3_1.ICCR1.BIT.ICE = 1u; /* IIC3 モジュール動作可能 */
129      IIC3_1.ICCR1.BIT.RCVD = 0u; /* 次の受信動作を継続 */
130      IIC3_1.ICCR1.BIT.MST = 1u; /* マスタ選択 */
131      IIC3_1.ICCR1.BIT.TRS = 1u; /* 送信選択 */
132
133      IIC3_1.ICCR1.BIT.CKS = 5u; /* 転送クロックレート : Pφ / 92 (391kHz) */
134      /* ---IIC bus mode register (ICMR) setting --- */
135      IIC3_1.ICMR.BYTE = 0x30u;
136          /*
137              bit7      : MLS:0 ----- MSB first
138              bit6-4    : Reserve:1 ----- Reserve bit
139              bit3      : BCWP:0----- Unsetting
140              bit2-0    : BC0:0, BC1:0,BC0:0----- IIC format 9-bit
141          */
142
143      return(E_OK);
144  }
145

```

3.5 サンプルプログラムリスト"main.c (5) "

```

146  /*"FUNC COMMENT"*****
147  * ID      :
148  * Outline : EEPROM へのデータライト
149  *-----
150  * Include : #include "iodefine.h"
151  *-----
152  * Declaration : int io_iic3_eeprom_write(unsigned char d_code,
153  *      :                unsigned char d_adr,
154  *      :                unsigned short w_adr,
155  *      :                unsigned int w_size,
156  *      :                unsigned char* w_buf);
157  *-----
158  * 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
159  *      : EEPROM へ w_buf で指定した領域のデータを w_size バイト分
160  *      : 書き込みます。EEPROM のメモリアドレス
161  *      : は w_adr で指定します。
162  *-----
163  * Argument   : unsigned char d_code : デバイスコード
164  *      : unsigned char d_adr  : デバイスアドレス
165  *      : unsigned short w_adr : 書き込み開始アドレス
166  *      : unsigned int w_size  : 書き込みデータサイズ
167  *      : unsigned char* w_buf : 書き込みデータ格納先
168  *-----
169  * Return Value: ACK 応答有り : E_OK
170  *      : ACK 応答無し : E_ERR
171  *"FUNC COMMENT END"*****/
172  int io_iic3_eeprom_write(unsigned char d_code,unsigned char d_adr,unsigned short w_adr,
173                          unsigned int w_size,unsigned char* w_buf)
174  {
175      int ack = E_OK;
176      int i;
177      unsigned char send[3];
178
179      send[0] = (unsigned char)(d_code|((d_adr & 0x7)<<1)|IIC_DATA_WR);
180      send[1] = (unsigned char)((w_adr>>8) & 0x00ff);
181      send[2] = (unsigned char)(w_adr & 0x00ff);
182
183      while(IIC3_1.ICCR2.BIT.BBSY == 1u){
184          /* バス解放待ち */
185      }
186      IIC3_1.ICCR1.BYTE |= 0x30u; /* マスタ送信モードに設定 */
187      IIC3_1.ICCR2.BYTE = ((IIC3_1.ICCR2.BYTE & 0xbfu)|0x80u); /* 開始条件発行 */
188
189      ack = io_iic3_address_send(send); /* 第1,2,3バイト送信 */
190

```

3.6 サンプルプログラムリスト"main.c (6) "

```

191     if(ack == E_OK){
192         /* 指定したデバイスから ACK 応答有り */
193         for(i=0;i<w_size;i++){
194             ack = io_iic3_data_send(*w_buf++);          /* データ送信 */
195             if(ack == E_ERR){
196                 break;
197             }
198         }
199         io_iic3_mst_send_end();
200     }
201     else{
202         /* 指定したデバイスから ACK 応答無し */
203         io_iic3_mst_send_end();
204     }
205     return(ack);
206 }
207

```

3.7 サンプルプログラムリスト "main.c (7) "

```

208  /*"FUNC COMMENT"*****
209  * ID      :
210  * Outline  : スレーブデバイスアドレス送信
211  *-----
212  * Include  :
213  *-----
214  * Declaration : int io_iic3_address_send(unsigned char* data);
215  *-----
216  * Description : data で指定したスレーブデバイスのアドレス (1 バイト)
217  *              : およびメモリアドレス (2 バイト) の送信を行います。
218  *-----
219  * Argument   : unsigned char* data : 送信データ
220  *-----
221  * Return Value: ACK 応答有り : E_OK
222  *              : ACK 応答無し : E_ERR
223  *"FUNC COMMENT END"*****/
224  int io_iic3_address_send(unsigned char* data)
225  {
226      int ack;
227
228      ack = io_iic3_data_send(*data++);      /* スレーブデバイスのアドレス */
229      if(ack == E_ERR){
230          return(ack);
231      }
232      ack = io_iic3_data_send(*data++);      /* 1st メモリアドレス */
233      if(ack == E_ERR){
234          return(ack);
235      }
236      ack = io_iic3_data_send(*data);        /* 2nd メモリアドレス */
237      if(ack == E_ERR){
238          return(ack);
239      }
240      return(ack);
241  }
242

```

3.8 サンプルプログラムリスト"main.c (8) "

```

243  /*"FUNC COMMENT"*****
244  * ID      :
245  * Outline : 1バイトデータ送信
246  *-----
247  * Include : #include "iodefine.h"
248  *-----
249  * Declaration : int io_iic3_data_send(unsigned char data);
250  *-----
251  * Description : dataを送信します。処理手順は以下の通りです。
252  *              : 1.ICDRT エンプティ待ち
253  *              : 2.送信データセット
254  *              : 3.送信完了確認
255  *              : 4.ACK 応答確認
256  *-----
257  * Argument  : unsigned char data : 送信データ
258  *-----
259  * Return Value: ACK 応答有り : E_OK
260  *              : ACK 応答無し : E_ERR
261  *"FUNC COMMENT END"*****/
262  int io_iic3_data_send(unsigned char data)
263  {
264      int ack;
265
266      while(IIC3_1.ICSR.BIT.TDRE == 0u){
267          /* ICDRT のエンプティ待ち */
268      }
269      IIC3_1.ICDRT = data;
270      while(IIC3_1.ICSR.BIT.TEND == 0u){
271          /* 送信完了待ち */
272      }
273      if(IIC3_1.ICIER.BIT.ACKBR == 0u){
274          ack = E_OK;
275      }
276      else{
277          ack = E_ERR;
278      }
279      return(ack);
280  }
281

```

3.9 サンプルプログラムリスト"main.c (9) "

```

282  /*"FUNC COMMENT"*****
283  * ID      :
284  * Outline : 停止条件発行
285  *-----
286  * Include : #include "iodefine.h"
287  *-----
288  * Declaration : void io_iic3_mst_send_end(void);
289  *-----
290  * 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。
291  *-----
292  * Argument  : void
293  *-----
294  * Return Value: void
295  *"FUNC COMMENT END"*****/
296  void io_iic3_mst_send_end(void)
297  {
298      IIC3_1.ICSR.BIT.TEND = 0u;      /* TEND フラグクリア */
299      IIC3_1.ICSR.BIT.STOP = 0u;     /* STOP フラグクリア */
300      IIC3_1.ICCR2.BYTE &= 0x3fu;   /* 停止条件発行 */
301
302      while(IIC3_1.ICSR.BIT.STOP == 0u){
303          /* バス解放待ち */
304      }
305
306      IIC3_1.ICCR1.BYTE &= 0xcfu;   /* スレーブ受信モード */
307      IIC3_1.ICSR.BIT.TDRE = 0u;   /* TDRE をクリア */
308  }
309

```

3.10 サンプルプログラムリスト "main.c (10) "

```

310  /*"FUNC COMMENT"*****
311  * ID      :
312  * Outline  : Acknowledge Polling
313  *-----
314  * Include  : #include "iodefine.h"
315  *-----
316  * Declaration : io_iic3_ack_polling
317  *-----
318  * Description : 本関数は、EEPROM が書き換え中か否かを判定します。
319  *              : EEPROM は、内部で書き換え中の場合入力コマンドを無視し、
320  *              : ACK を返しません。本関数で EEPROM が書き換え中ではないこと
321  *              : を確認してから EEPROM にアクセスしてください。
322  *              : Acknowledge Polling 時に送信する Read/Write コードは、
323  *              : 使用するデバイスにより異なります。使用する EEPROM の
324  *              : データシートで確認してください。
325  *-----
326  * Argument   : void
327  *-----
328  * Return Value: E_OK   : NOT_BUSY
329  *              : E_ERR  : BUSY (EEPROM は書き換え中)
330  *"FUNC COMMENT END"*****/
331  int io_iic3_ack_polling(void)
332  {
333      int ack = E_OK;
334      unsigned char send = (unsigned char)(DEVICE_CODE|((DEVICE_ADDR & 0x7)<<1)|IIC_DATA_WR);
335
336      while(IIC3_1.ICCR2.BIT.BBSY == 1u){
337          /* バス解放待ち */
338      }
339      IIC3_1.ICCR1.BYTE |= 0x30u;          /* マスタ送信モードに設定 */
340      IIC3_1.ICCR2.BYTE = ((IIC3_1.ICCR2.BYTE & 0xbf) | 0x80u); /* 開始条件発行 */
341
342      ack = io_iic3_data_send(send);
343
344      io_iic3_mst_send_end(); /* 停止条件発行 */
345
346      return(ack);
347  }
348  /* End of File */

```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A/SH-2A-FPU ソフトウェアマニュアル Rev.3.00
(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。)
- ハードウェアマニュアル
SH7262 グループ、SH7264 グループ ハードウェアマニュアル Rev.1.00
(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/>

お問い合わせ先

<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	2009.03.23	—	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、あるいはその他軍用途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。
11. 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。
12. 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。
13. 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