
SH7216 グループ

RJJ06B1079-0101

Rev.1.01

IIC3 マスタ送信

2010.06.25

要旨

本アプリケーションノートは、SH7216 の I²C バスインタフェース 3 (IIC3) のシングルマスタ送信例を掲載しています。

動作確認デバイス

SH7216

目次

1. はじめに	2
2. 概要	3
3. 参考プログラム例	12
4. 参考ドキュメント	18

1. はじめに

1.1 仕様

- マスタデバイス,スレーブデバイスを共に SH7216 として,マスタ送信側からスレーブ受信側へデータを転送します。
- 転送レートは 403kHz に設定しています。

【注】 このサンプルプログラムの動作には SH7216 CPU ボード (R0K572167) を 2 台使用し,スレーブ受信側のソフトウェアは「SH7216 グループ アプリケーションノート IIC3 スレーブ受信 (RJJ06B1080)」に記載したものを使用しています。

1.2 使用機能

- I²C パスインタフェース 3 (IIC3)
- クロックパルス発振器 (CPG)
- 低消費電力モード
- ピンファンクションコントローラ (PFC)

1.3 適用条件

- マイコン: SH7216
- 動作周波数: 内部クロック 200 MHz
バスクロック 100 MHz
周辺クロック 50 MHz
- 統合開発環境: ルネサス エレクトロニクス製
High-performance Embedded Workshop Ver.4.05.01
- 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.9.03.00 Release 01
- 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 -pic=1 -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)
- スレーブ受信モジュール: SH7216 CPU ボード (R0K572167)

1.4 関連アプリケーションノート

- SH7216 グループ アプリケーションノート IIC3 スレーブ受信 (RJJ06B1080)

2. 概要

本プログラムでは、 I^2C バスインタフェース 3 (IIC3) を使用し、マスタデバイスの SH7216 からスレーブデバイスの SH7216 にデータ転送を行います。

2.1 使用機能の動作概要

I^2C バスインタフェース 3 (IIC3) は、フィリップス社が提唱する I^2C バス (Inter IC Bus) インタフェース方式に準拠しており、サブセットの機能を備えています。ただし、 I^2C 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構成と異なりますので注意してください。

SH7216 の I^2C 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。

- I^2C 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です。
- 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 の概念図を示します。

表 1 フォーマット別の特長

フォーマット	特長
I^2C 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。

【注】 IIC3 についての詳細は、「SH7216 グループ ハードウェアマニュアル (RJJ09B0575) I^2C バスインタフェース 3」の章をご参照ください。

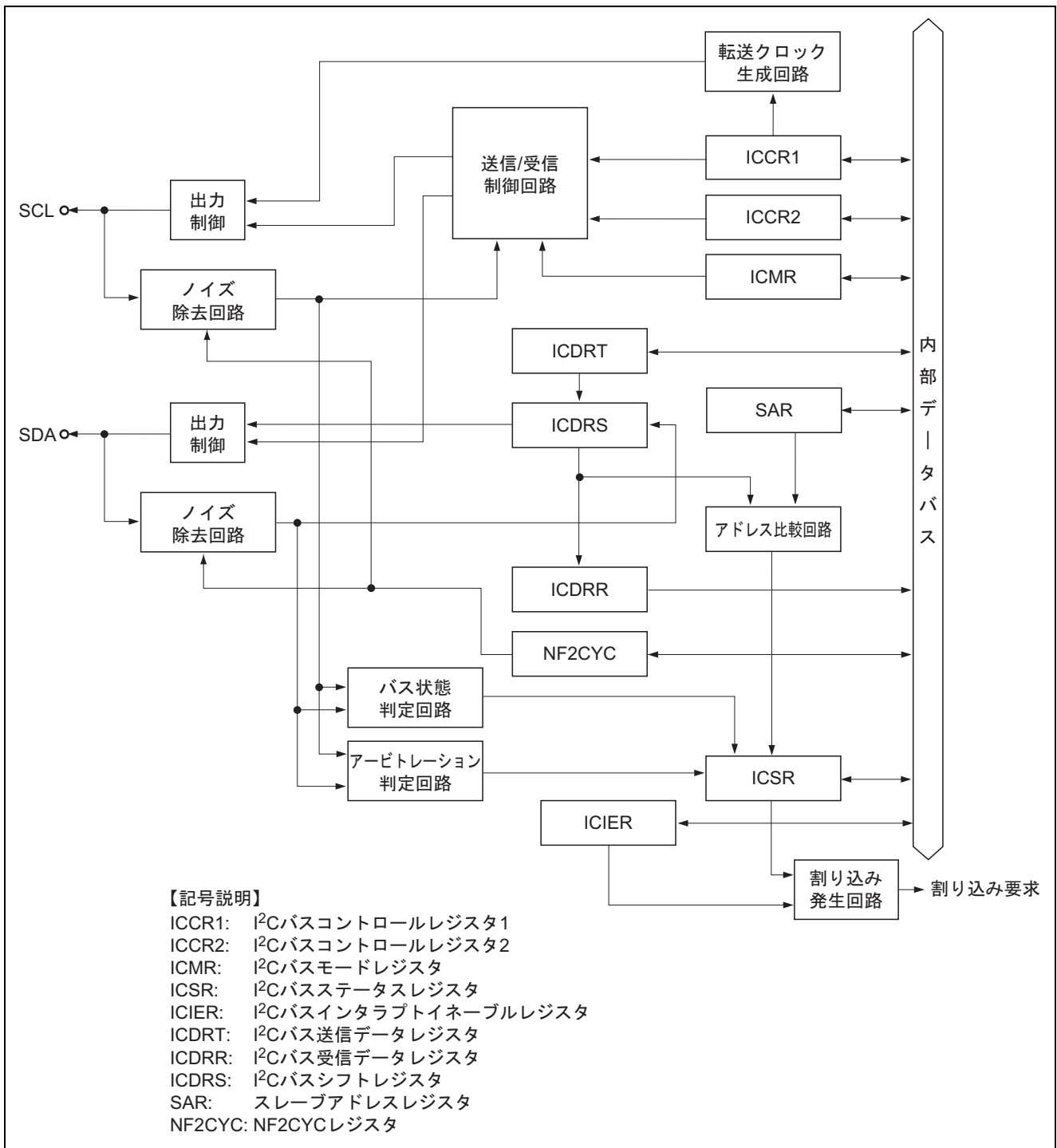


図 1 IIC3 の概念図

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たすように設定してください。参考プログラムでは Pφ/124 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、「SH7216 グループ ハードウェアマニュアル (RJJ09B0575)」を参照ください。

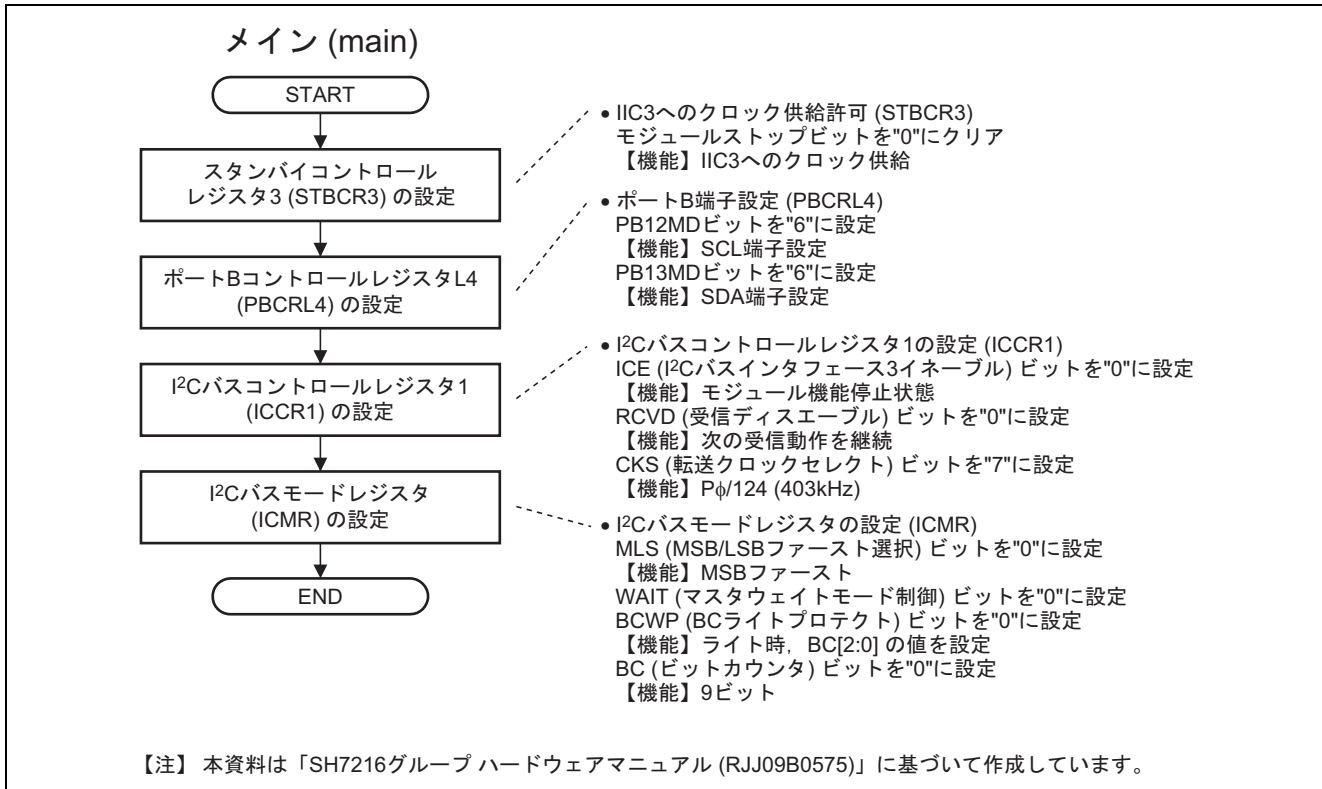


図 2 IIC3 の初期設定フロー例

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。

デバイスコードは"B'1010"，デバイスアドレス"B'000"を使用します。

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。

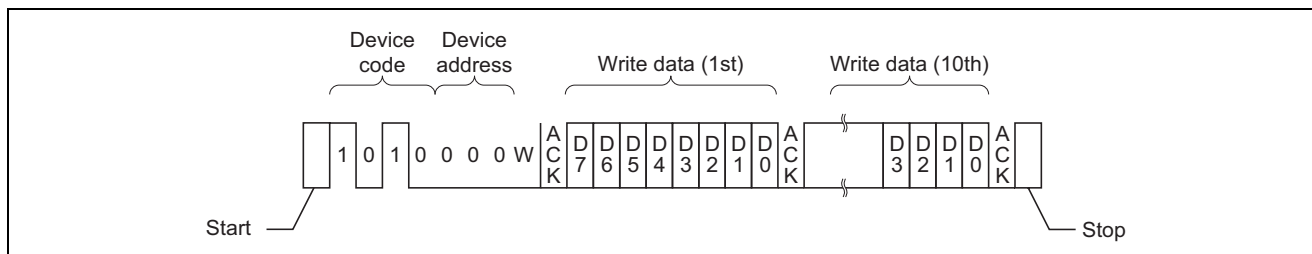


図 3 ページライト動作図

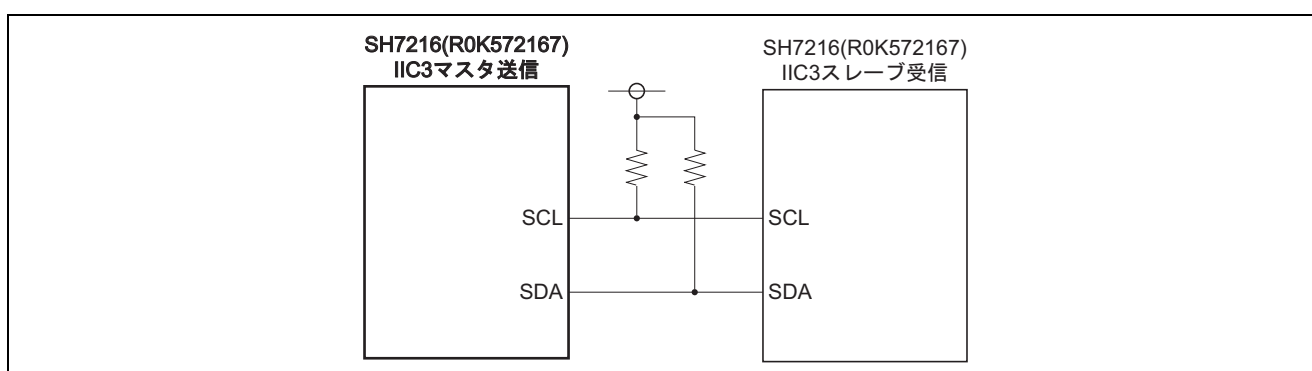


図 4 参考プログラムの動作環境

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8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

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'FFFE0408	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'FFFE0000	H'B7	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'0111": 転送レート Pφ/124
I ² C バスモードレジスタ (ICMR)	H'FFFE0002	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
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

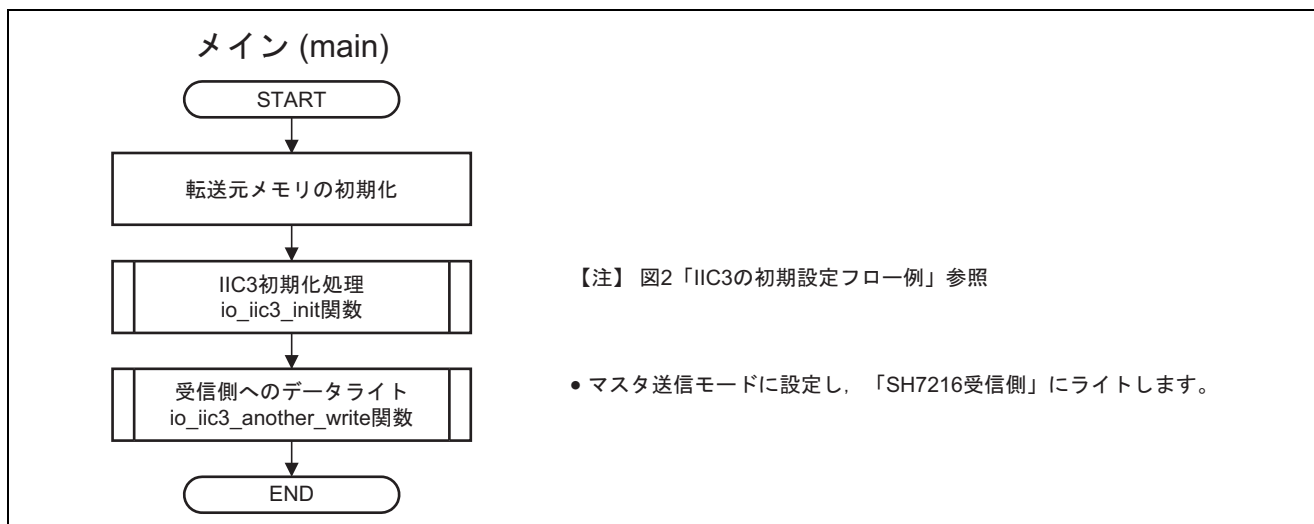


図5 参考プログラムの処理フロー (1)

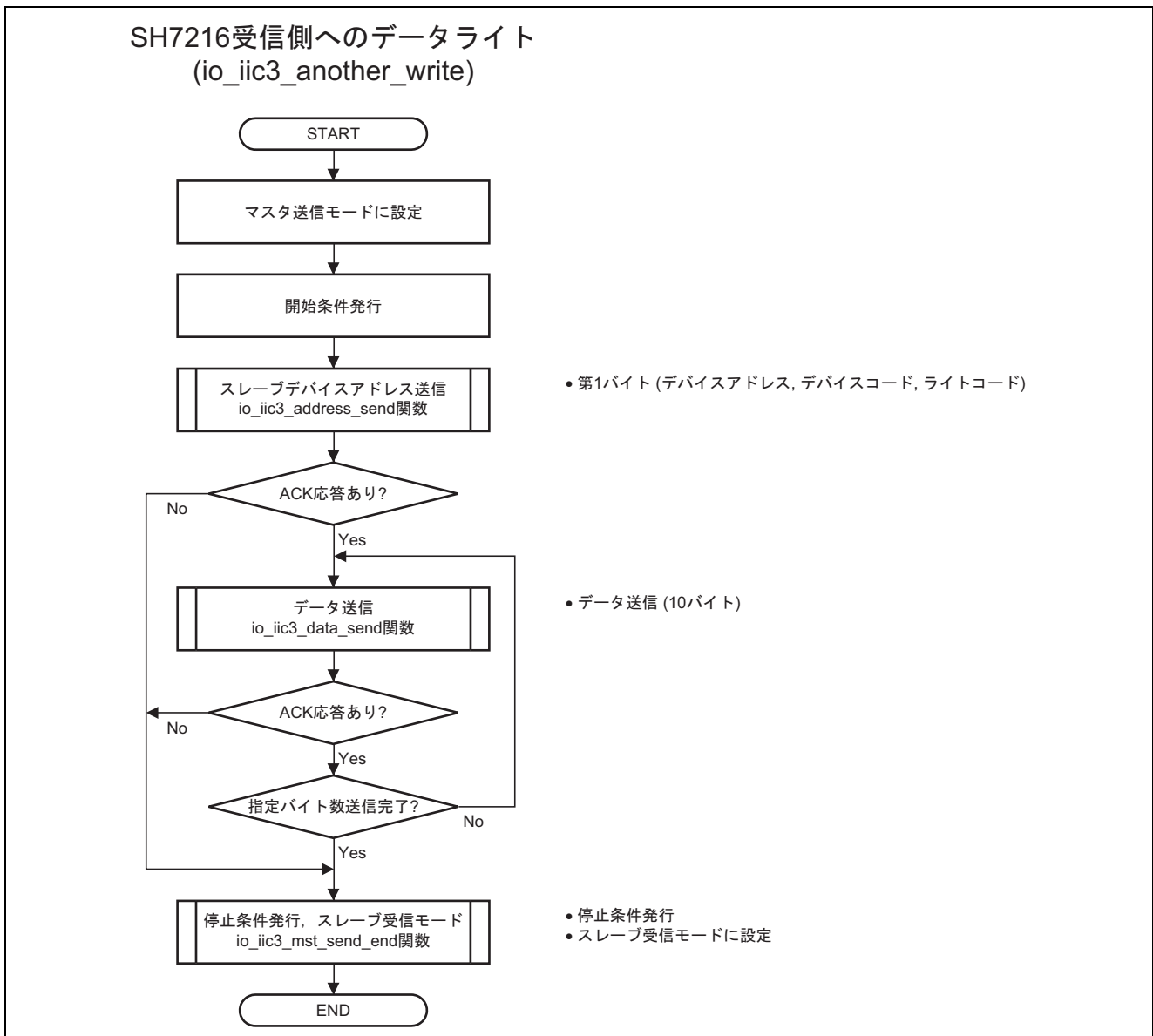


図 6 参考プログラムの処理フロー (2)

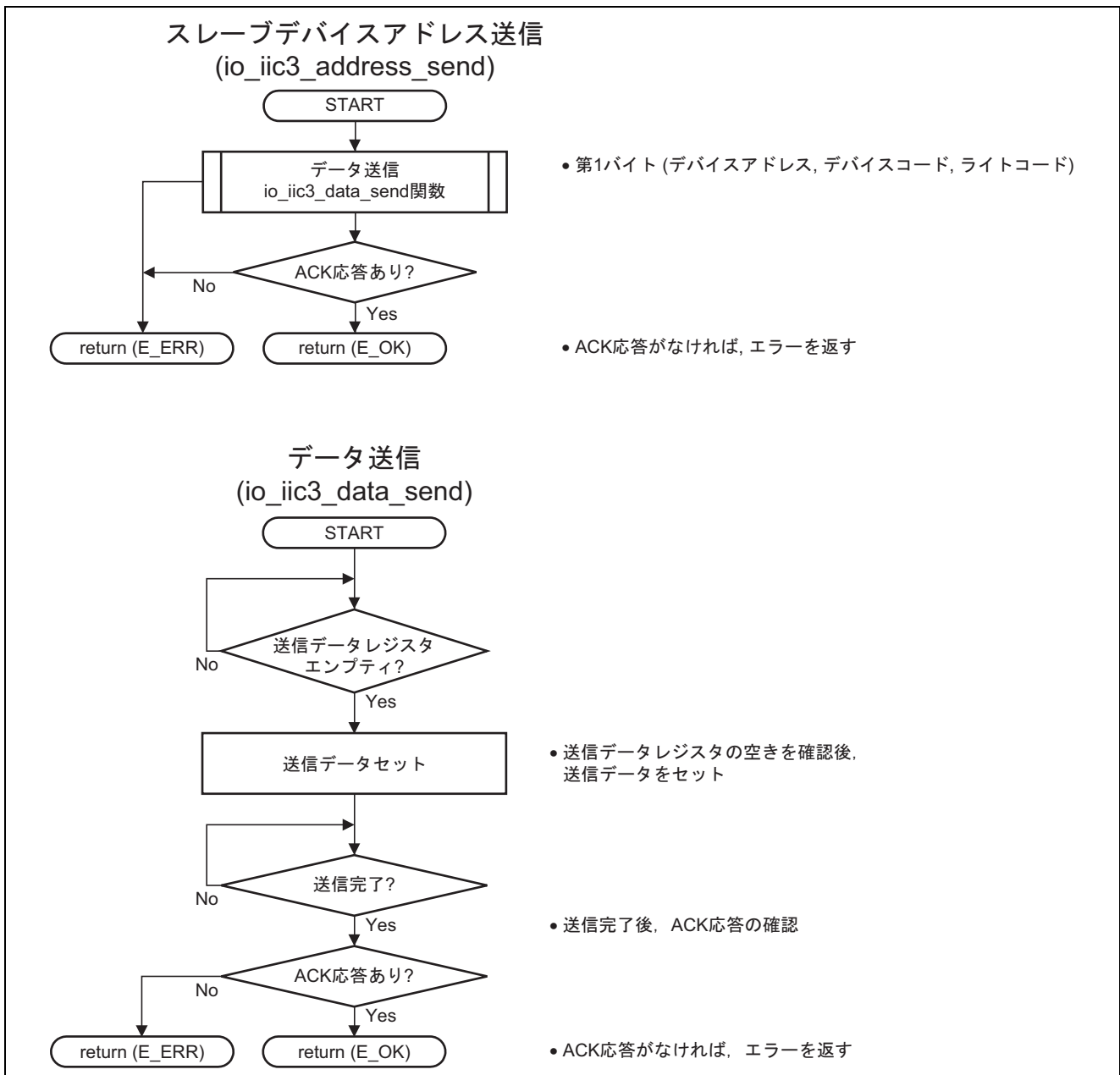


図7 参考プログラムの処理フロー (3)

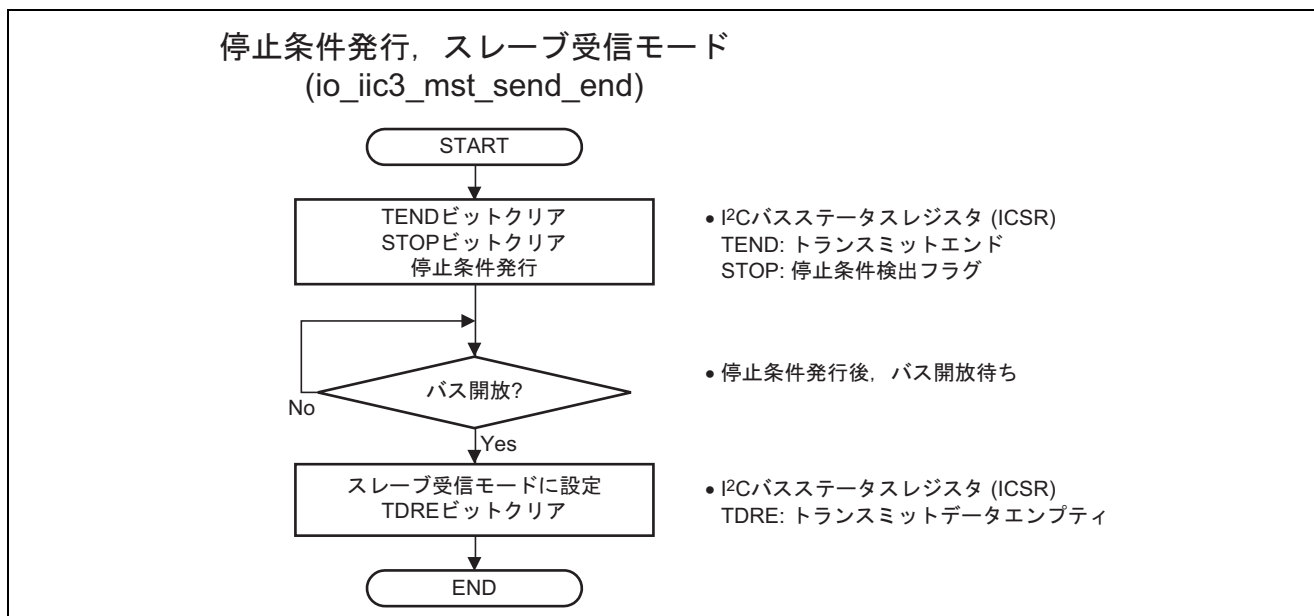


図 8 参考プログラムの処理フロー (4)

3. 参考プログラム例

```

/*****
*   DISCLAIMER
*
*   This software is supplied by Renesas Technology Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Technology Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
*****/
*   Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.
**"FILE COMMENT"***** Technical reference data *****
*   System Name : SH7216 Sample Program
*   File Name   : main.c
*   Abstract    : Sample Program of Compare Match Timer
*   Version     : 1.00.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.05.01).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 :                               (Ver.9.03 Release01).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description :
*   note       : This program sends data to the SH7211 base on the
*   : slave-receive side by using the master-send mode of IIC3.
*****/
*   History    : Jun.8,2009 Ver.1.00.00
**"FILE COMMENT END"*****
#include <machine.h>
#include "iodefine.h"      /* SH7216 iodefine */

/* ==== symbol definition ==== */
#define DEVICE_CODE 0xA0      /* another device code   : b'1010   */
#define DEVICE_ADDR 0x00     /* another device address : b'000   */
#define IIC_DATA_WR 0x00     /* Data write code       : b'0     */
#define IIC_DATA_RD 0x01     /* Data read code        : b'1     */
#define IIC3_DATA 10

#define E_OK 0
#define E_ERR -1

/* ==== RAM allocation variable declaration ==== */
unsigned char WriteData[IIC3_DATA];

```

図9 サンプルプログラムリスト"main.c" (1)

```

/* ==== prototype declaration ==== */
void main(void);
int io_iic3_another_write(unsigned char d_code, unsigned char d_adr,
                          unsigned int w_size, unsigned char *w_buf);
int io_iic3_data_send(unsigned char data);
int io_iic3_address_send(unsigned char *data);
void io_iic3_mst_send_end(void);
int io_iic3_init(void);

void io_led_ini(void);
void io_init_cmt0(void);

/*"FUNC COMMENT"*****
 * Outline      : sumple program main
 *-----
 * Include      :
 *-----
 * Declaration  : void main(void);
 *-----
 * Function     : sumple program main
 *-----
 * Argument     : void
 *-----
 * Return Value : none
 *-----
 * Notice       :
 *"FUNC COMMENT END"*****/
void main(void)
{
    int i;

    /* ==== Writing data creation ==== */
    for(i = 0; i < IIC3_DATA; i++){
        WriteData[i] = IIC3_DATA+i;
    }

    /* ==== IIC3 initialization ==== */
    io_iic3_init();
    /* ==== IIC3 master-send mode ==== */
    io_iic3_another_write(DEVICE_CODE,          /* Device code */
                          DEVICE_ADDR,         /* Device address */
                          sizeof(WriteData),   /* Writing data's size */
                          WriteData);         /* Store the writing data */

    while(1){
        /* Loop */
    }
}

```

図 10 サンプルプログラムリスト"main.c" (2)

```

/*"FUNC COMMENT"*****
 * Outline      : IIC3-module initial setting
 *-----
 * Include      : #include "iodefine.h"
 *-----
 * Declaration  : int io_iic3_init(void);
 *-----
 * Function     : IIC3-module initial setting
 *-----
 * Argument    : void
 *-----
 * Return Value : E_OK
 *-----
 * Notice      :
 *"FUNC COMMENT END"*****/
int io_iic3_init(void)
{
    STB.CR3.BIT._IIC3 = 0x00;      /* Clear the IIC3 module standby mode */

    /* ---- PFC setting ---- */
    PFC.PBCRL4.BIT.PB12MD = 0x06; /* SCL selection */
    PFC.PBCRL4.BIT.PB13MD = 0x06; /* SDA selection */
    PFC.PBPCRL.BIT.PB12PCR = 1;   /* SCL pull-up */
    PFC.PBPCRL.BIT.PB13PCR = 1;   /* SDA pull-up */
    /* ----IIC31 module operation disabled ---- */
    IIC3.ICCR1.BIT.ICE = 0x00;    /* IIC3 transfer disabled state */
    IIC3.ICCR1.BIT.ICE = 0x01;    /* IIC3 transfer enabled */
    IIC3.ICCR1.BIT.RCVD = 0x00;   /* Continue the next receive data */
    IIC3.ICCR1.BIT.CKS = 0x07;    /* Transfer clock rate:P /124(403kHz) */
    /* ---IIC bus mode register (ICMR) setting --- */
    IIC3.ICMR.BYTE = 0x30;
    /*
        bit7  : MLS:0 ----- MSB first
        bit6  : WAIT:0 ----- No WAIT insertion
        bit5-4 : Reserve:1 ----- Reserve bits
        bit3  : BCWP:0 ----- Unsetting
        bit2-0 : BC2:0, BC1:0, BC0:0 ----- IIC bus format: 9 bits
    */

    return(E_OK);
}

```

図 11 サンプルプログラムリスト"main.c" (3)

```

/*"FUNC COMMENT"*****
* Outline      : Write data for slave-side
*-----
* Include      : #include "iodefine.h"
*-----
* Declaration  : int io_iic3_mst_send(unsigned char d_code,
*          :          unsigned char d_adr,
*          :          unsigned int w_size,
*          :          unsigned char *w_buf);
*-----
* Function     : This master-send mode choose slave-side is specified
* : by d_code & d_adr.
*          : Sending data to be written in slave-side is specified by w_buf,
* : and the data is "w_size"BYTE.
*          : (DEVICE_CODE = d_code, DEVICE_ADDR = d_adr,
* : sizeof(WriteData) = w_size, WriteData = *w_buf)
*-----
* Argument     : unsigned char d_code ; Device code
*          : unsigned char d_adr ; Device address
*          : unsigned int w_size ; Writing data size
*          : unsigned char* w_buf ; Store the writing data
*-----
* Return Value : E_OK ; ACK response
*          : E_ERR ; No ACK response
*-----
* Notice      :
* "FUNC COMMENT END"*****/
int io_iic3_another_write(unsigned char d_code, unsigned char d_adr,
                          unsigned int w_size, unsigned char *w_buf)
{
    int ack = E_OK;
    int i;
    unsigned char send[1];

    send[0] = (unsigned char)(d_code | ((d_adr & 0x7) << 1) | IIC_DATA_WR);

    while(IIC3.ICCR2.BIT.BBSY == 0x01){
        /* It waits untill the bus is opened */
    }
    /* It sets it to the master send mode */
    IIC3.ICCR1.BYTE |= 0x30;
    /* Start condition is issued */
    IIC3.ICCR2.BYTE = ((IIC3.ICCR2.BYTE & 0xbf) | 0x80);

    ack = io_iic3_address_send(send);          /* 1st BYTE send */

    if(ack == E_OK){
        /* When there is ack reaction from the specified device */
        for(i = 0; i < w_size; i++){
            ack = io_iic3_data_send(*w_buf++);          /* Data send */
            if(ack == E_ERR){
                break;
            }
        }
        io_iic3_mst_send_end();
    }
    else{
        /* When there is ack reaction from the specified device */
        io_iic3_mst_send_end();
    }
    return(ack);
}

```

図 12 サンプルプログラムリスト"main.c" (4)

```

/*"FUNC COMMENT"*****
 * Outline      : Slave device address send
 *-----
 * Include      :
 *-----
 * Declaration  : int io_iic3_address_send(unsigned char *data);
 *-----
 * Function     : The address of the slave device specified with
 * : data (1byte) is send.
 *-----
 * Argument     : unsigned char *data ; Sending data
 *-----
 * Return Value : E_OK ; ACK response
 *              : E_ERR ; No ACK response
 *-----
 * Notice      :
 *"FUNC COMMENT END"*****/
int io_iic3_address_send(unsigned char *data)
{
    int ack;

    ack = io_iic3_data_send(*data++); /* Address of slave device */
    if(ack == E_ERR){
        return(ack);
    }
    return(ack);
}
/*"FUNC COMMENT"*****
 * Outline      : 1byte data sending
 *-----
 * Include      : #include "iodefine.h"
 *-----
 * Declaration  : int io_iic3_data_send(unsigned char data);
 *-----
 * Function     : Data sending. The processing procedure is as follows.
 * : 1. ICDRT waits for even empty
 * : 2. Sending data set
 * : 3. Data sending completion confirmation
 * : 4. ACK response confirmation
 *-----
 * Argument     : unsigned char data ; Sending data
 *-----
 * Return Value : E_OK ; ACK response
 *              : E_ERR ; No ACK response
 *-----
 * Notice      :
 *"FUNC COMMENT END"*****/
int io_iic3_data_send(unsigned char data)
{
    int ack;
    while(IIC3.ICSR.BIT.TDRE == 0x0){
        /* 1. ICDRT waits for even empty */
    }
    IIC3.ICDRT = data; /* 2. Sending data set */
    while(IIC3.ICSR.BIT.TEND == 0x00){
        /* 3. Data sending completion confirmation */
    }
    /* 4. ACK response confirmation */
    if(IIC3.ICIER.BIT.ACKBR == 0){
        ack = E_OK;
    }
    else{
        ack = E_ERR;
    }
    return(ack);
}

```

図 13 サンプルプログラムリスト"main.c" (5)


```
/*"FUNC COMMENT"*****  
* Outline      : Stop condition issue  
*-----  
* Include      : #include "iodefine.h"  
*-----  
* Declaration  : void io_iic3_mst_send_end(void);  
*-----  
* Function     : The conditions stop is issued, and it switches to  
* : the slave-receive mode.  
*-----  
* Argument     : void  
*-----  
* Return Value : none  
*-----  
* Notice      :  
* "FUNC COMMENT END"*****/  
void io_iic3_mst_send_end(void)  
{  
    IIC3.ICSR.BIT.TEND = 0x00;    /* TEND Flag clear */  
    IIC3.ICSR.BIT.STOP = 0x00;   /* STOP Flag clear */  
    IIC3.ICCR2.BYTE &= 0x3f;    /* The conditions stop is issued */  
  
    while(IIC3.ICSR.BIT.STOP == 0x00){  
        /* It waits untill the bus is opened */  
    }  
  
    IIC3.ICCR1.BYTE &= 0xcf;     /* It switches to the slave-receive mode */  
    IIC3.ICSR.BIT.TDRE = 0x00;  /* TDRE clear */  
}  
/* End of File */
```

図 14 サンプルプログラムリスト"main.c" (6)

4. 参考ドキュメント

- ソフトウェアマニュアル
 - (1) SH-2A, SH2A-FPU ソフトウェアマニュアル (RJJ09B0086)
(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)
- ハードウェアマニュアル
 - (2) SH7216 グループ ハードウェアマニュアル (RJJ09B0575)
(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)

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	2009.07.28	—	初版発行
1.01	2010.06.25	—	FRQCR 設定方法変更に伴うソースプロジェクト修正

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っていると、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>