

---

## R8C/56E グループ

R01AN1288JJ0100

Rev.1.00

## I<sup>2</sup>C バスシングルマスタ制御プログラム (マスタ送信/受信)

---

2012.11.30

### 要旨

本アプリケーションノートでは、R8C/56E グループの I<sup>2</sup>C バスインタフェースを使用した I<sup>2</sup>C バスシングルマスタ制御プログラム(マスタ送信/受信)について説明しています。

### 対象デバイス

R8C/56E グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

## 目次

1. 仕様.....	3
2. 動作確認条件.....	4
3. ソフトウェア説明.....	5
3.1 動作概要.....	6
3.2 定数一覧.....	10
3.3 構造体と共用体一覧.....	10
3.4 変数一覧.....	11
3.5 関数一覧.....	11
3.6 関数仕様.....	12
3.7 フローチャート.....	16
3.7.1 メイン処理.....	16
3.7.2 システムクロック設定処理.....	17
3.7.3 I <sup>2</sup> C_0 初期設定処理.....	18
3.7.4 マスタ制御開始処理.....	20
3.7.5 I <sup>2</sup> C バスインタフェース割り込み処理.....	22
3.7.6 停止条件検出処理.....	23
3.7.7 マスタ送信処理.....	24
3.7.8 マスタ受信処理.....	25
3.7.9 マスタ制御終了処理.....	26
4. サンプルコード.....	27
5. 参考ドキュメント.....	27

## 1. 仕様

マスタモードで送受信を行います。

使用条件は下記の通りです。I<sup>2</sup>C バスの通信プロトコルに準拠しています。

<使用条件>

- スレーブアドレス長 : 7 ビット
- 転送レート : 約 357kHz(Standard-mode、Fast-mode 対応)
- シングルマスタ通信(マルチマスタ未対応)
- リスタートコンディション生成未対応

表 1.1に使用する周辺機能と用途を、図 1.1にブロック図を示します。

表1.1 使用する周辺機能と用途

周辺機能	用途
クロック同期形シリアルインタフェース (I <sup>2</sup> C バスインタフェースモード)	データを送受信する

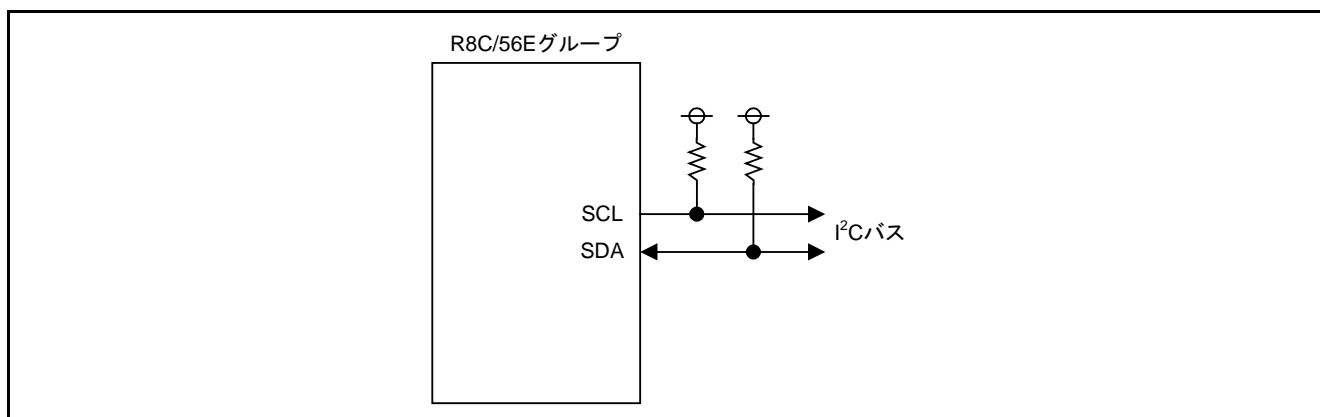


図1.1 ブロック図

## 2. 動作確認条件

本アプリケーションノートのサンプルコードは、表 2.1 の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	R8C/56E グループ
動作周波数	<ul style="list-style-type: none"><li>• XIN クロック : 20MHz</li><li>• システムクロック : 20MHz</li><li>• CPU クロック : 20MHz</li></ul>
動作電圧	5.0V(2.7V~5.5V で動作可能)
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09
C コンパイラ	ルネサスエレクトロニクス製 M16C Series, R8C Family C Compiler V.5.45 Release 01 コンパイルオプション -D__UART0__ -c -finfo -dir "\$(CONFIGDIR)" -R8C (総合開発環境のデフォルト設定を使用しています)

### 3. ソフトウェア説明

マスタ送信で3バイトを送信後、マスタ受信で3バイトを受信します。マスタ送信とマスタ受信を、交互に繰り返します。

#### <設定条件>

- シリアルクロック入出力は、P3\_5/SCL 端子を使用します。
- シリアルデータ入出力は、P3\_7/SDA 端子を使用します。
- I<sup>2</sup>C バスインタフェースモードをマスタモードで使用します。
- チャンネルは I<sup>2</sup>C\_0 を使用します。
- 転送クロックは、f1/56 を使用します(転送レートは、約 357kHz に設定します)。
- ウェイトの挿入時間は、ウェイトなしを設定します(データとアクノリッジを連続して転送)。
- 転送フォーマットは、MSB ファーストを使用します。
- SDA デジタル遅延値は、3×f1 サイクルを使用します。
- アクノリッジの判定は、受信アクノリッジビット(ACKBR ビット)を使用します。
- 受信データフル割り込み要求を使用します。
- 送信終了割り込み要求を使用します。
- 停止条件検出割り込み要求を使用します。
- 送信データエンプティ割り込み要求は使用しません。
- NACK 受信割り込み要求およびアービトレーションロスト/オーバランエラー割り込み要求は使用しません。

#### <転送レートの計算式>

- 転送レート = SICR1\_0 レジスタの CKS0~CKS3 ビットによる設定  
= 20MHz(f1) ÷ 56  
= 357.142 kHz

#### <使用上の注意事項>

本プログラムを使用する場合の注意事項を以下に示します。

- I<sup>2</sup>C バス機能を使用する場合、SITDR\_0 レジスタおよび SIRDR\_0 レジスタは 8 ビット単位でアクセスしてください。
- 統合開発環境(High-performance Embedded Workshop)が自動的に生成するレジスタ定義ファイルでは、これらのレジスタが unsigned short で定義されているため、union byte\_def、unsigned short 等への変更が必要です。

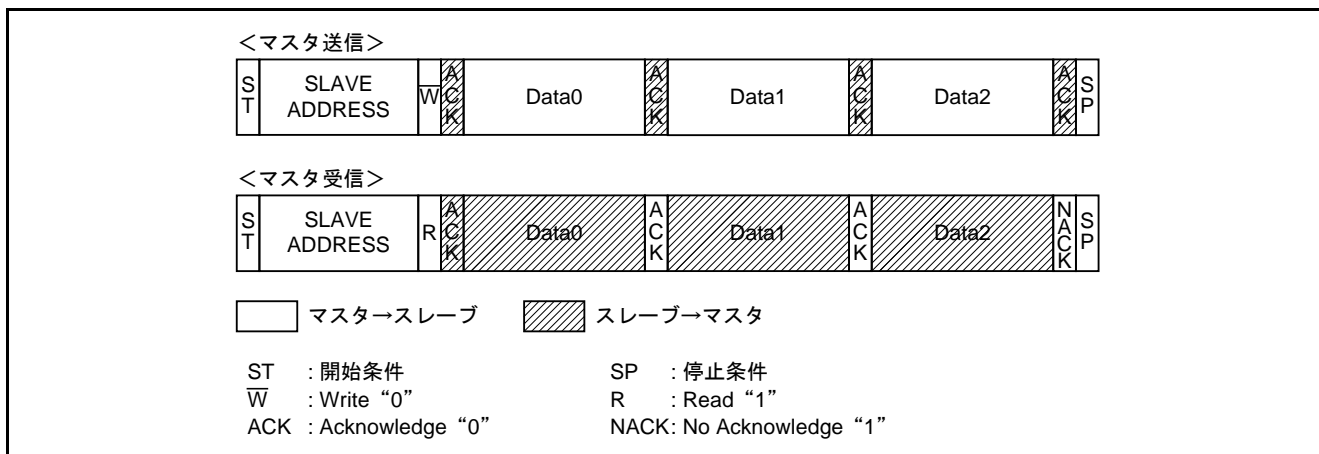


図3.1 転送フォーマット

### 3.1 動作概要

処理の概要を以下に説明します。各番号は、図 3.2～図 3.5に記載している番号に対応しています。

#### (1) 初期設定

システムクロックおよびI<sup>2</sup>Cバスインタフェース関連SFRの初期設定と、使用する変数の初期化を行います。

#### (2) マスタ制御開始

開始条件を生成します。

I<sup>2</sup>Cバスインタフェース割り込み(送信終了割り込み要求)を許可に設定し、スレーブアドレスを送信します。

#### (3) I<sup>2</sup>Cバスインタフェース割り込み(送信終了割り込み要求)

SCLクロックの9ビット目の立ち上がりで、割り込みが発生します。

<マスタ送信時>

- ACK/NACKの判定をし、次バイトの送信データを設定します。

<マスタ受信時>

- ACK/NACKの判定をし、ACKの場合は、次バイトの送信データを設定します。
- 通信を終了するときは、送信終了割り込み要求と受信データフル割り込み要求を禁止に設定します。停止条件を生成し、停止条件検出割り込み要求を許可に設定します。

#### (4) I<sup>2</sup>Cバスインタフェース割り込み(受信データフル割り込み要求)

マスタ受信の時、SCLクロックの9ビット目の立ち上がりで、割り込みが発生します。

次バイトのACK/NACKを設定し、受信データを読み出します。

通信を終了するときは、送信終了割り込み要求と受信データフル割り込み要求を禁止に設定します。その後、停止条件を生成し、停止条件検出割り込み要求を許可に設定します。

#### (5) I<sup>2</sup>Cバスインタフェース割り込み(停止条件検出割り込み要求)

停止条件を検出した時、割り込みが発生します。

停止条件検出割り込み要求を禁止に設定します。

マスタ受信時は、最終受信データを読み出します。

スレーブ受信モードに設定し、I<sup>2</sup>Cバスインタフェース割り込みを禁止に設定します。

図 3.2に概略フローチャート、図 3.3～図 3.5にタイミング図を示します。

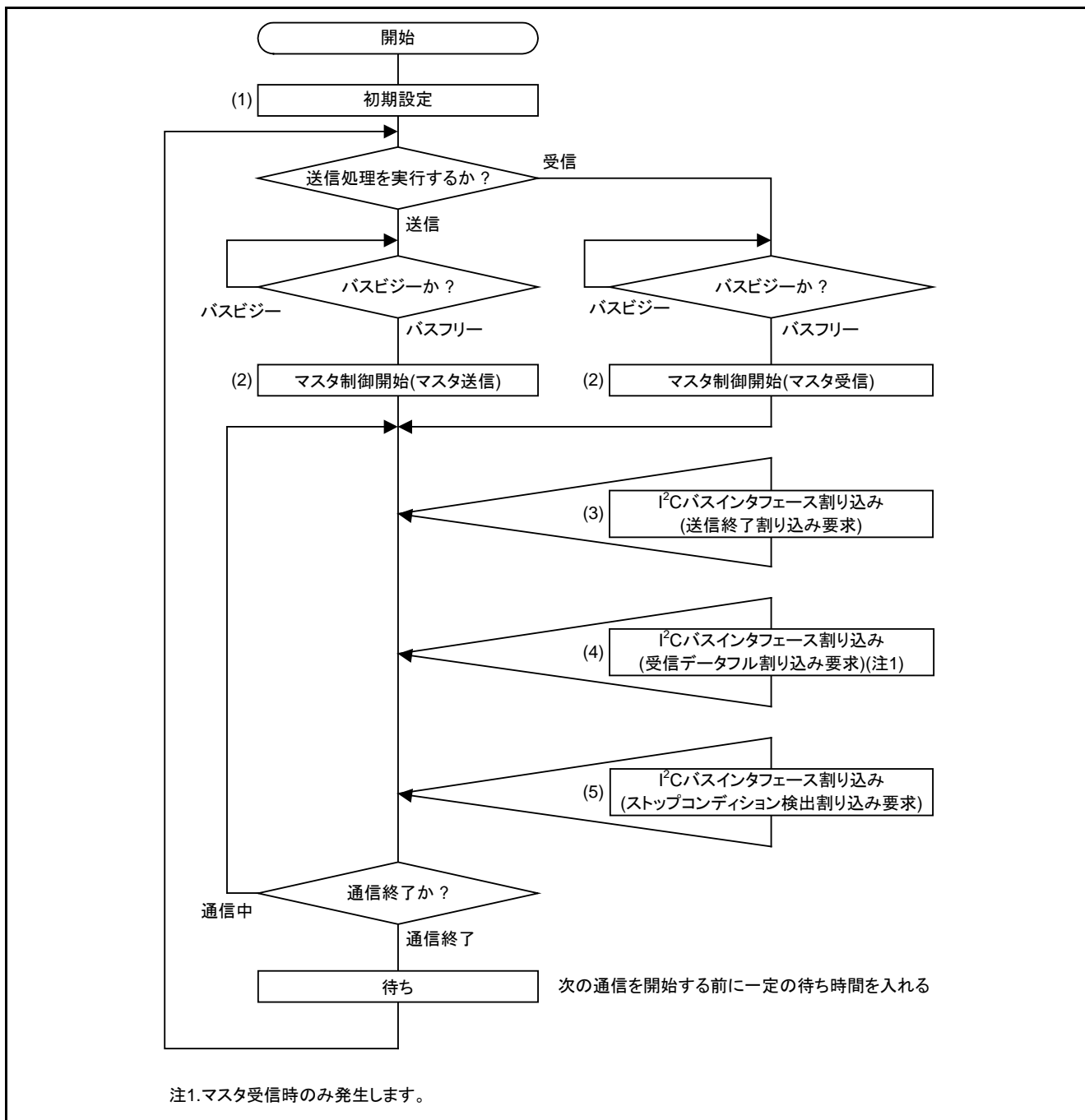


図3.2 概略フローチャート

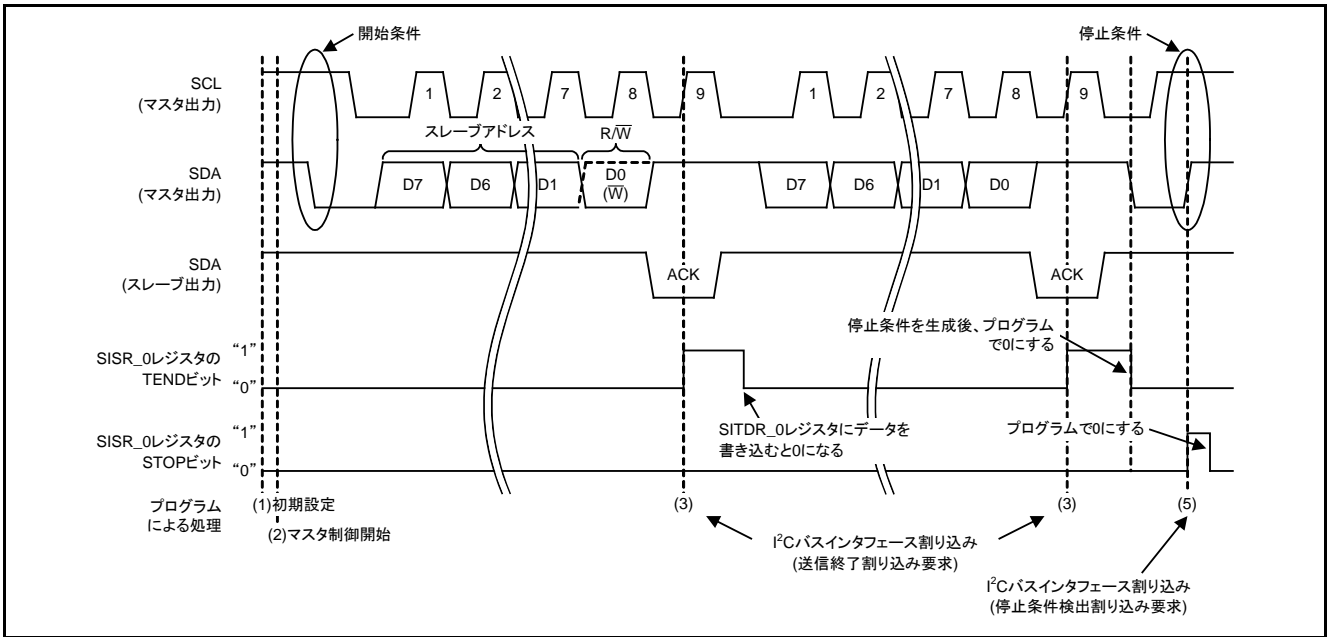


図3.3 マスタ送信のタイミング図

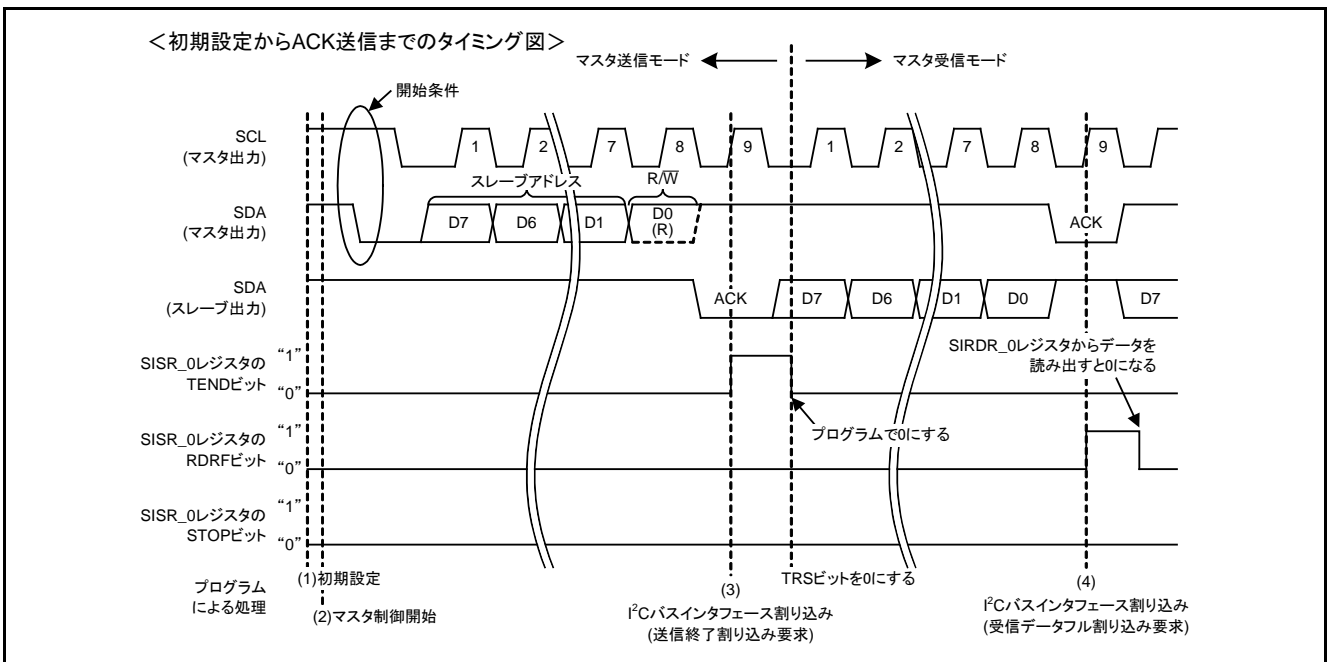


図3.4 マスタ受信のタイミング図(1)



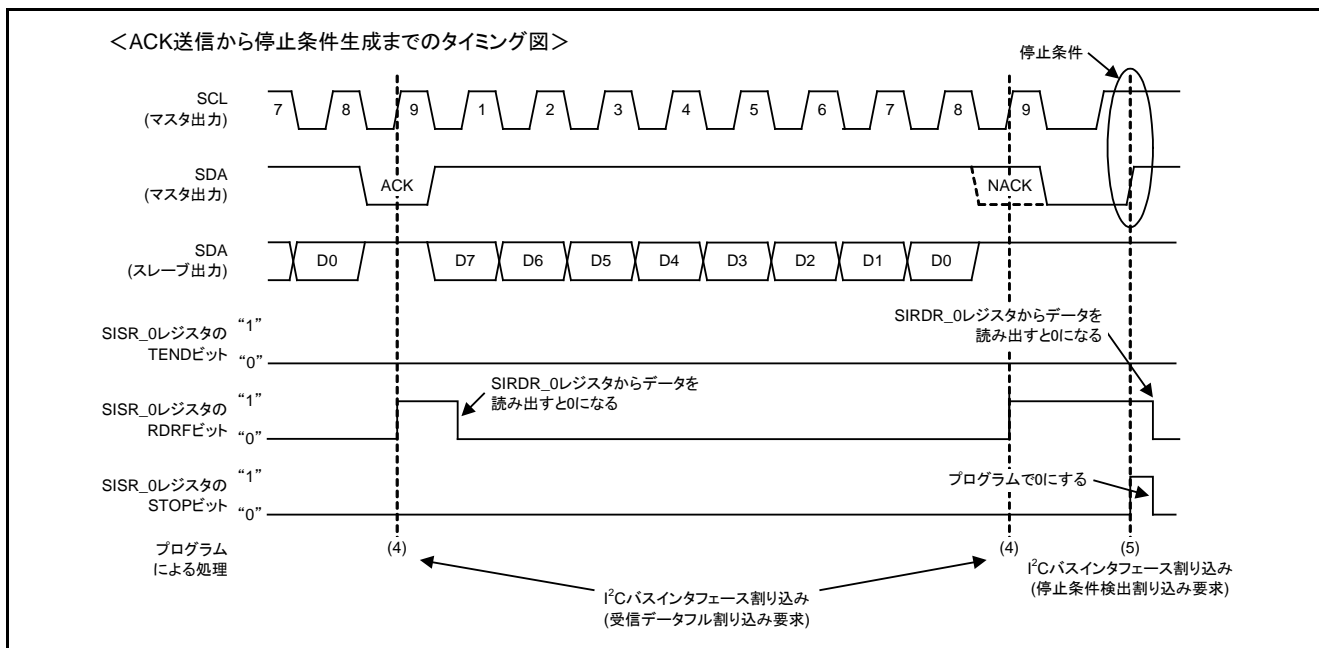


図3.5 マスタ受信のタイミング図(2)

### 3.2 定数一覧

表 3.1にサンプルコードで使用する定数を示します。

表3.1 サンプルコードで使用する定数

定数名	設定値	内容
DEVICE_ADDRESS	0b01010101	スレーブアドレス(7ビット(b6-b0)を使用)
READ	1	マスタ受信
WRITE	0	マスタ送信
LENGTH	3	送信・受信データのバイト数
BUFSIZE	255	送信・受信データのバッファ数
PD_IIC	0x2E7	方向レジスタのアドレス
PD_IIC_INIT	0b01011111	方向レジスタの設定値 PD3_5/SCL : 入力 PD3_7/SDA : 入力
IIC_SP_ON	1	停止条件生成
IIC_SP_OFF	0	停止条件未生成

### 3.3 構造体と共用体一覧

図 3.6にサンプルコードで使用する構造体と共用体一覧を示します。

```

typedef union{
    struct{
        unsigned char b0:1;
        unsigned char b1:1;
        unsigned char b2:1;
        unsigned char b3:1;
        unsigned char b4:1;
        unsigned char b5:1;
        unsigned char b6:1;
        unsigned char b7:1;
    }bit;
    unsigned char all;
}byte_dt;

static byte_dt iic_str1;          /*-- Device address(b7-b1) + R/W(b0) --*/
#define iic_slave_addr iic_str1.all /* Slave Address */
#define iic_rw iic_str1.bit.b0     /* 0: Write (Master Transmit)      1: Read (Master Receive) */

static byte_dt iic_str2;        /* -- Status --*/
#define iic_status iic_str2.all /* All status */
#define iic_start iic_str2.bit.b0 /* 1: During communication      0: Communication end */
#define iic_err_par iic_str2.bit.b1 /* 1: Parameter error           0: not error */
#define iic_err_nack iic_str2.bit.b2 /* 1: NACK detection error      0: not error */
#define iic_err_addr iic_str2.bit.b3 /* 1: Address not match error    0: not error */

static unsigned char iic_length; /* Transfer data length */
static unsigned char iic_index; /* Index of transmit/receive byte number */
static unsigned char far *iic_pointer; /* Pointer of buffer */

```

図3.6 サンプルコードで使用する構造体と共用体一覧

### 3.4 変数一覧

表 3.2にグローバル変数を示します。

表3.2 グローバル変数

型	変数名	内容	使用関数
unsigned char	iic_tx[BUFSIZE]	送信バッファ	main
unsigned char	iic_rx[BUFSIZE]	受信バッファ	main

### 3.5 関数一覧

表 3.3に関数を示します。

表3.3 関数一覧

関数名	概要
mcu_init	システムクロック設定処理
iic_init	I <sup>2</sup> C_0 初期設定処理
iic_master_start	マスタ制御開始処理
_ssuic	I <sup>2</sup> C バスインタフェース割り込み処理
stp_int	停止条件検出処理
master_trn_int	マスタ送信処理
master_rcv_int	マスタ受信処理
iic_master_end	マスタ制御終了処理

### 3.6 関数仕様

サンプルコードの関数仕様を示します。

main	
概要	メイン処理
ヘッダ	iic.h
宣言	void main(void)
説明	下記の処理を行います。 <ul style="list-style-type: none"> <li>システムクロックとI<sup>2</sup>Cバスインタフェースの初期設定関数を呼び出します。</li> <li>初期設定後、マスタ送信とマスタ受信を交互に繰り返します。iic_master_start関数を呼び出してマスタ制御を開始後、iic_master_end関数を呼び出してマスタ制御の終了を待ちます。</li> </ul>
引数	なし
リターン値	なし

mcu_init	
概要	システムクロック設定処理
ヘッダ	なし
宣言	void mcu_init(void)
説明	クロックの設定を行います。
引数	なし
リターン値	なし

iic_init	
概要	I <sup>2</sup> C_0 初期設定処理
ヘッダ	なし
宣言	void iic_init(unsigned char ini)
説明	I <sup>2</sup> C_0 を使用するための初期設定を行います。
引数	unsigned char ini            0 : I <sup>2</sup> C_0 モジュール無効 1 : I <sup>2</sup> C_0 モジュール有効
リターン値	なし

iic_master_start									
概要	マスタ制御開始処理								
ヘッダ	なし								
宣言	unsigned char iic_master_start( unsigned char addr, unsigned char rw, unsigned char far *buf, unsigned char len)								
説明	<p>マスタ制御を開始するための処理を行います。この関数を実行する前に iic_init 関数を実行して I<sup>2</sup>C_0 モジュールを有効にしてください。</p> <p>下記の処理を行います。</p> <ul style="list-style-type: none"> <li>関数の先頭で、全ステータスを初期化し、パラメータチェックを行います。引数が範囲外の場合、パラメータエラーフラグを 1 に設定し、リターン値 0xFF を返します。パラメータエラーの場合、マスタ制御開始処理は行いません。</li> <li>次にバスの状態を確認します。バスビジーの場合、リターン値は 0 です。マスタ制御開始処理は行いません。バスフリーの場合、リターン値は 1 です。マスタ制御開始処理を行います。通信中フラグを 1 に設定した後、開始条件を生成し、スレーブアドレスを送信します。</li> </ul>								
引数	<table> <tr> <td>unsigned char addr</td> <td>0x00~0x7F : スレーブアドレス</td> </tr> <tr> <td>unsigned char rw</td> <td>0x00 : マスタ送信 0x01 : マスタ受信</td> </tr> <tr> <td>unsigned char far *buf</td> <td>送信または受信バッファへのポインタ</td> </tr> <tr> <td>unsigned char len</td> <td>0x01~0xFF : 転送データ長</td> </tr> </table>	unsigned char addr	0x00~0x7F : スレーブアドレス	unsigned char rw	0x00 : マスタ送信 0x01 : マスタ受信	unsigned char far *buf	送信または受信バッファへのポインタ	unsigned char len	0x01~0xFF : 転送データ長
unsigned char addr	0x00~0x7F : スレーブアドレス								
unsigned char rw	0x00 : マスタ送信 0x01 : マスタ受信								
unsigned char far *buf	送信または受信バッファへのポインタ								
unsigned char len	0x01~0xFF : 転送データ長								
リターン値	<table> <tr> <td>unsigned char</td> <td>0 : バスビジー 1 : バスフリー 0xFF : パラメータエラー</td> </tr> </table>	unsigned char	0 : バスビジー 1 : バスフリー 0xFF : パラメータエラー						
unsigned char	0 : バスビジー 1 : バスフリー 0xFF : パラメータエラー								

_ssuic	
概要	I <sup>2</sup> C バスインタフェース割り込み処理
ヘッダ	なし
宣言	void _ssuic(void)
説明	<p>SCL クロックの 9 ビット目の立ち上がり、もしくは停止条件検出で割り込みが発生します。</p> <p>停止条件検出の場合、stp_int 関数を呼び出します。</p> <p>停止条件未検出の場合、マスタ送信時は master_trn_int 関数を、マスタ受信時は master_rcv_int 関数を呼び出します。</p> <p>通信を終了するときは、停止条件を生成し、停止条件検出割り込み要求を許可に設定します。</p>
引数	なし
リターン値	なし

---

stp_int	
概要	停止条件検出処理
ヘッダ	なし
宣言	static void stp_int(void)
説明	I <sup>2</sup> C バスインタフェース割り込み処理から呼び出される関数です。通信中に変更した I <sup>2</sup> C バスインタフェース関連 SFR を初期設定時の状態に戻し、通信中フラグを 0 に設定します。
引数	なし
リターン値	なし

---



---

master_trn_int	
概要	マスタ送信処理
ヘッダ	なし
宣言	static unsigned char master_trn_int(void)
説明	I <sup>2</sup> C バスインタフェース割り込み処理から呼び出される関数です。 次の場合、リターン値は IIC_SP_OFF です。 <ul style="list-style-type: none"> <li>ACK 検出、かつ最終バイト以外(次の送信を開始します)。</li> </ul> 次の場合、リターン値は IIC_SP_ON です。 <ul style="list-style-type: none"> <li>NACK 検出時(NACK 検出エラーフラグを 1 に設定します)。</li> <li>最終バイト送信完了時</li> </ul>
引数	なし
リターン値	unsigned char                    IIC_SP_ON(0) : 停止条件を生成する IIC_SP_OFF(1) : 停止条件を生成しない

---



---

master_rcv_int	
概要	マスタ受信処理
ヘッダ	なし
宣言	static unsigned char master_rcv_int(void)
説明	I <sup>2</sup> C バスインタフェース割り込み処理から呼び出される関数です。 1 バイト目(スレーブアドレス)送信後に、マスタ受信モード設定し、受信データフル割り込み要求を許可にします。 次の場合、リターン値は IIC_SP_OFF です。 <ul style="list-style-type: none"> <li>最終バイト以外。</li> </ul> 次の場合、リターン値は IIC_SP_ON です。 <ul style="list-style-type: none"> <li>NACK 検出時(NACK 検出エラーフラグを 1 に設定します)。</li> <li>最終バイト受信完了時</li> </ul>
引数	なし
リターン値	unsigned char                    IIC_SP_ON(0) : 停止条件を生成する IIC_SP_OFF(1) : 停止条件を生成しない

---

---

**iic\_master\_end**

---

概要	マスタ制御終了処理	
ヘッダ	なし	
宣言	unsigned char iic_master_end(void)	
説明	メイン処理から呼び出される関数です。マスタ制御の終了状態をユーザに知らせます。通信中の場合は、0を返し、通信終了の場合は1を返します。必要に応じて終了時の処理を追加してください。	
引数	なし	
リターン値	unsigned char	0 : 通信中 1 : 通信終了

3.7 フローチャート

3.7.1 メイン処理

図 3.7にメイン処理のフローチャートを示します。

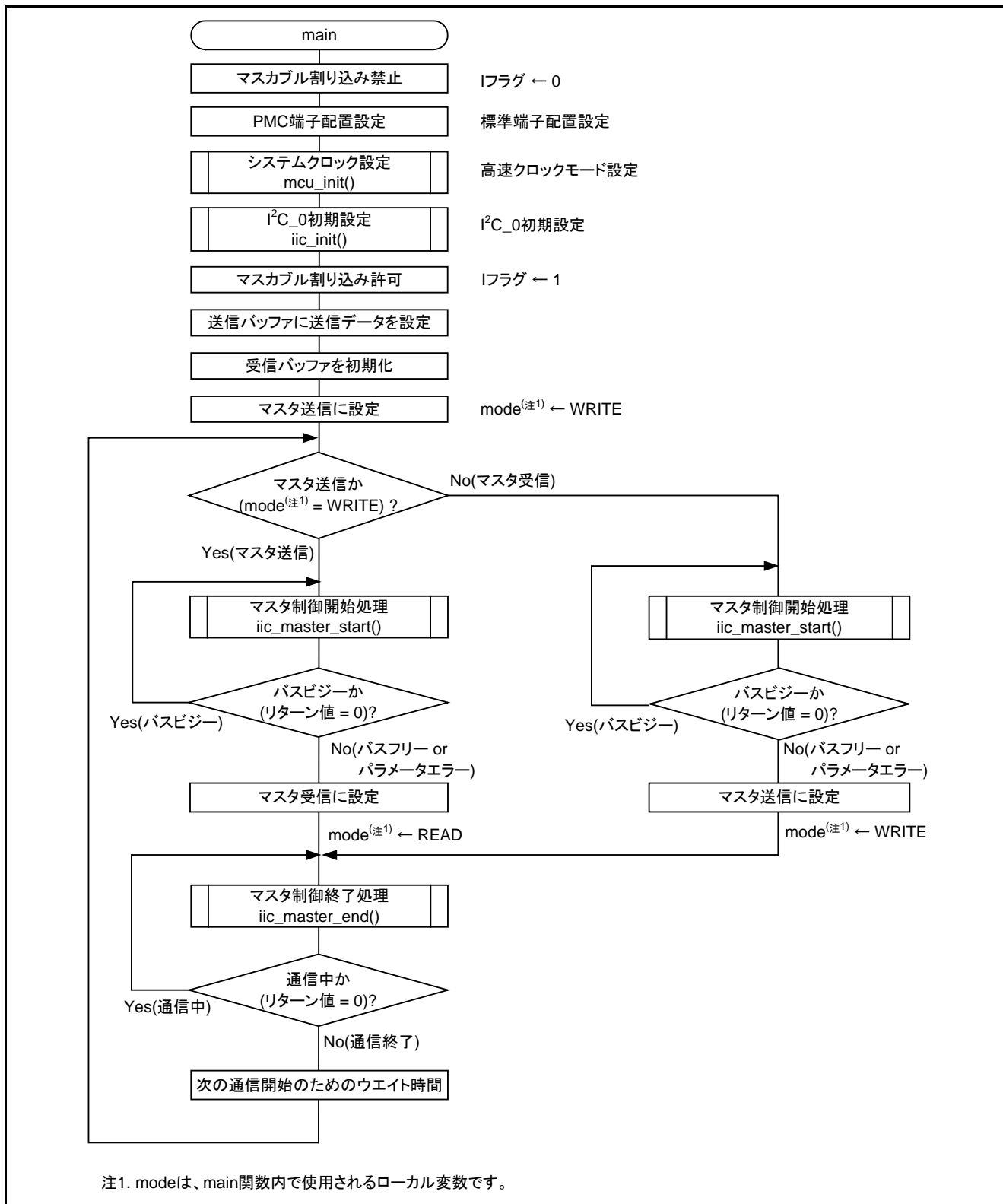


図3.7 メイン処理



## 3.7.2 システムクロック設定処理

図 3.8にシステムクロック設定処理のフローチャートを示します。

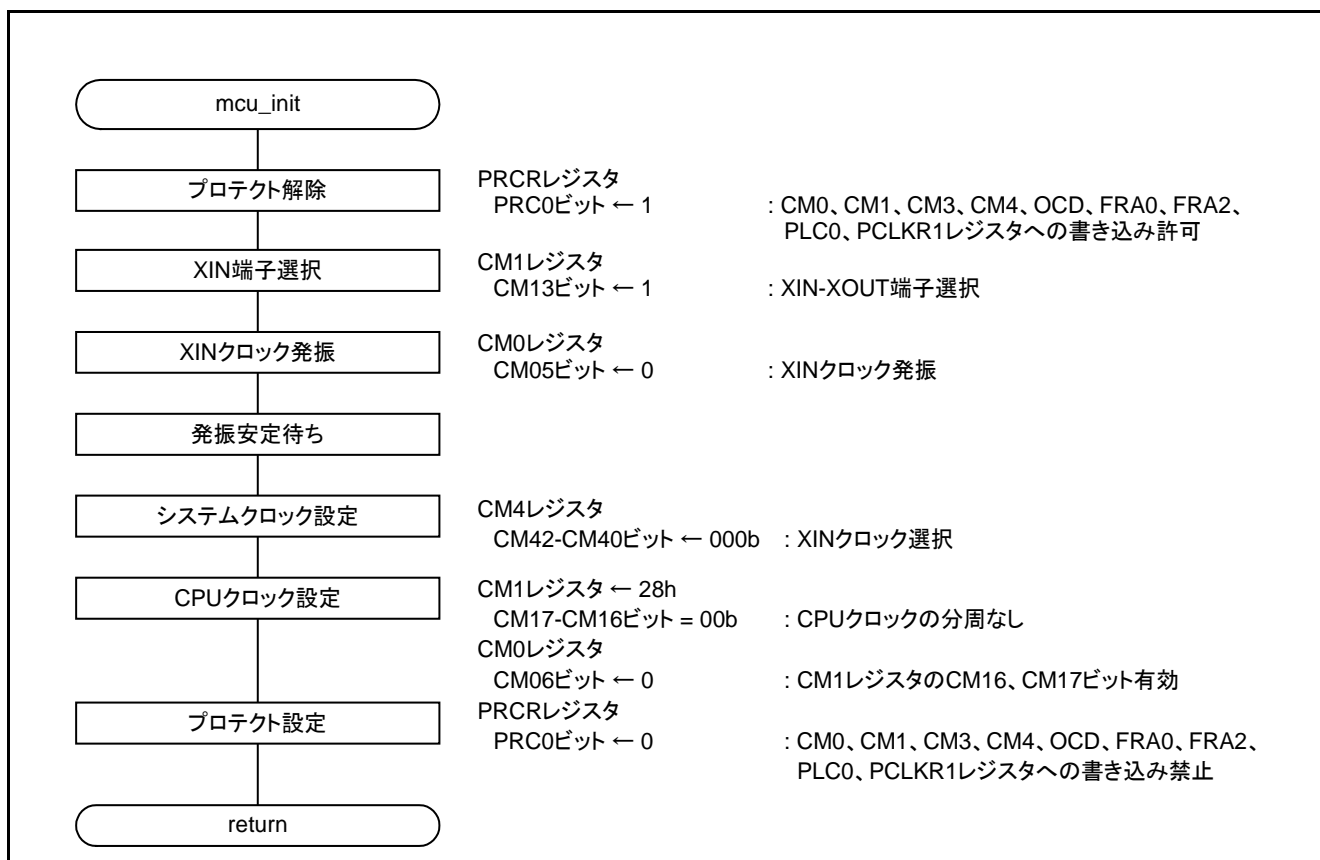


図3.8 システムクロック設定処理

3.7.3 I<sup>2</sup>C\_0 初期設定処理

図 3.9、図 3.10に I<sup>2</sup>C\_0 初期設定処理のフローチャートを示します。

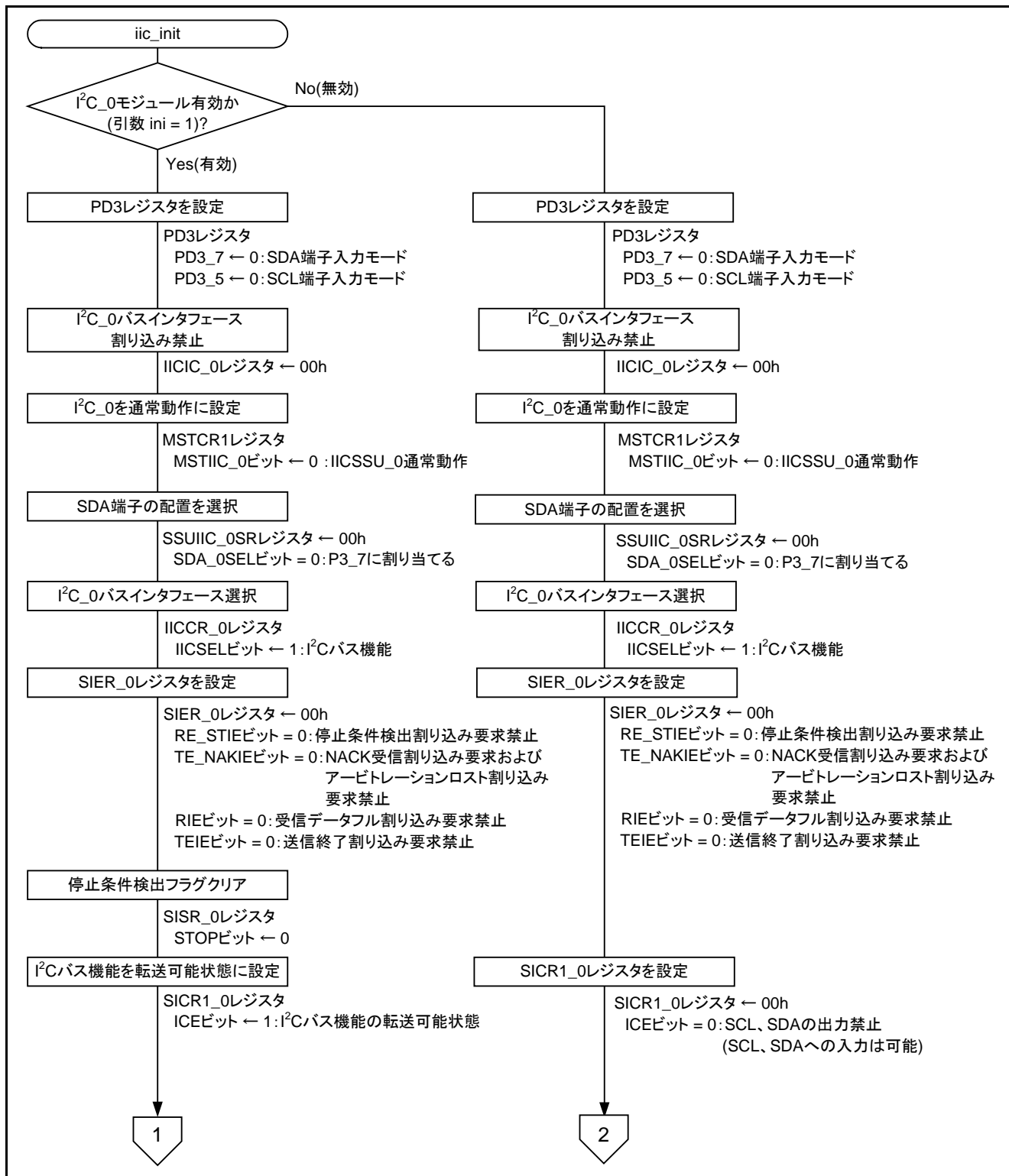


図3.9 I<sup>2</sup>C\_0 初期設定処理(1)

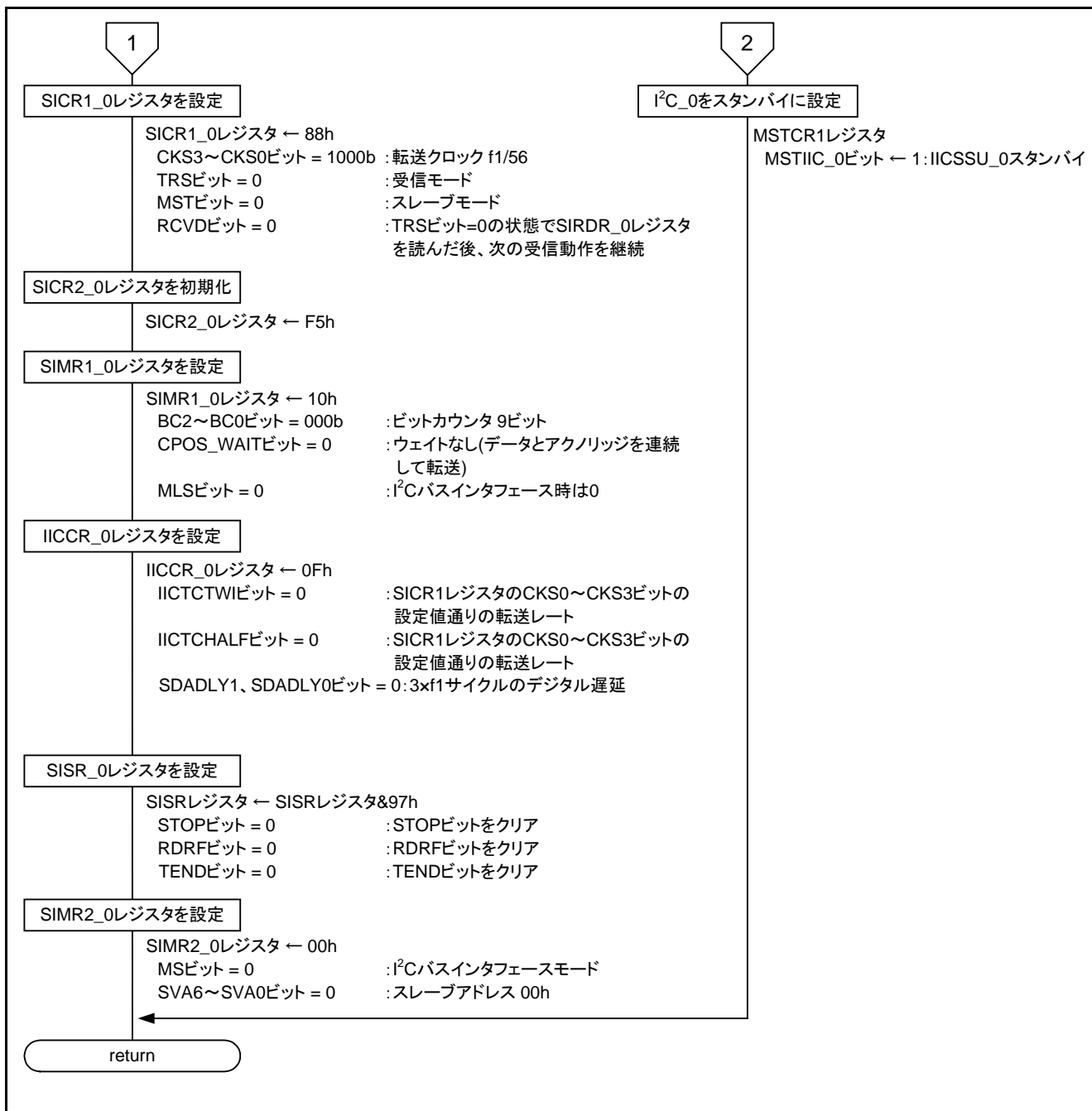


図3.10 I<sup>2</sup>C\_0 初期設定処理(2)

## 3.7.4 マスタ制御開始処理

図 3.11、図 3.12にマスタ制御開始のフローチャートを示します。

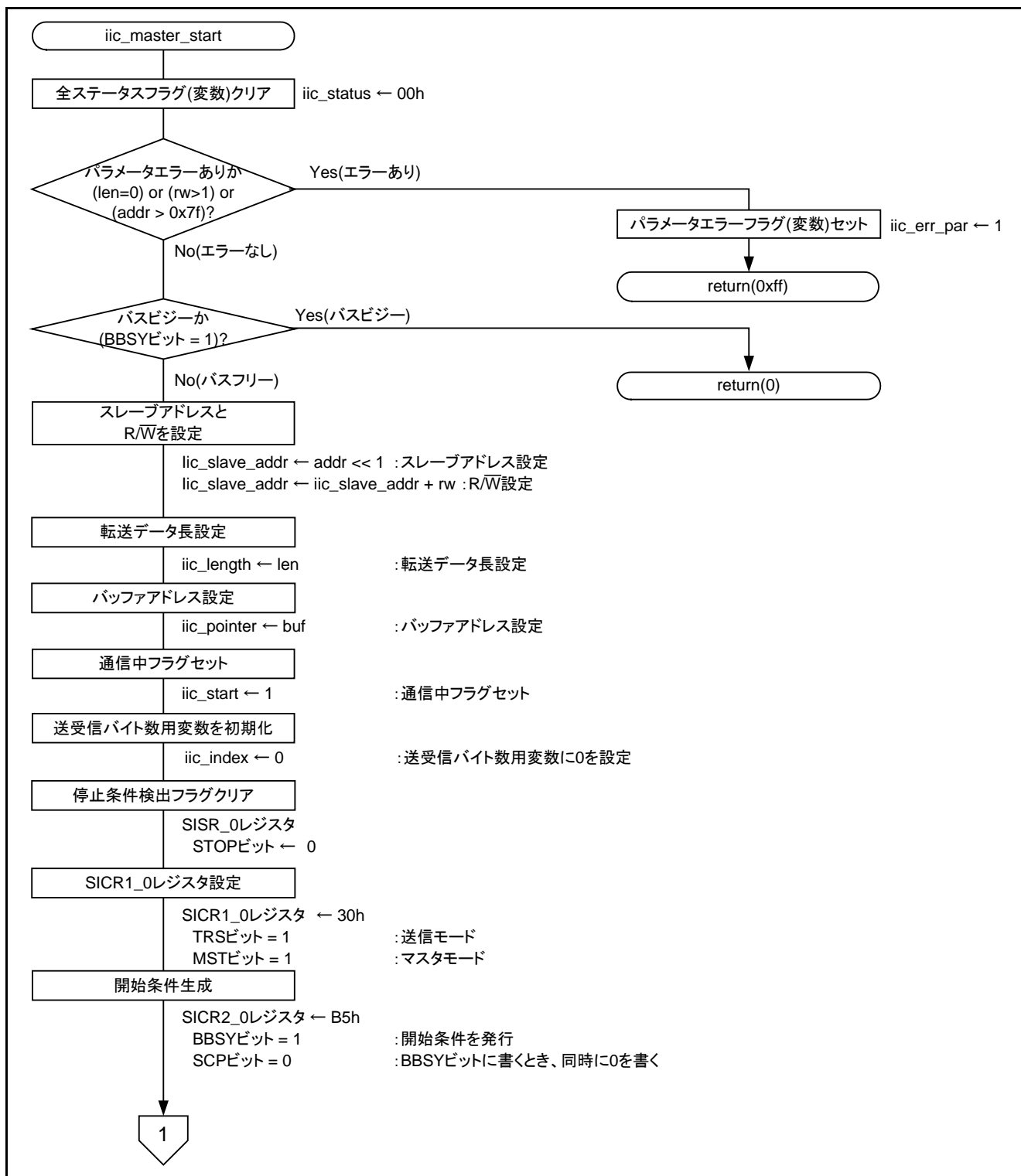


図3.11 マスタ制御開始処理(1)

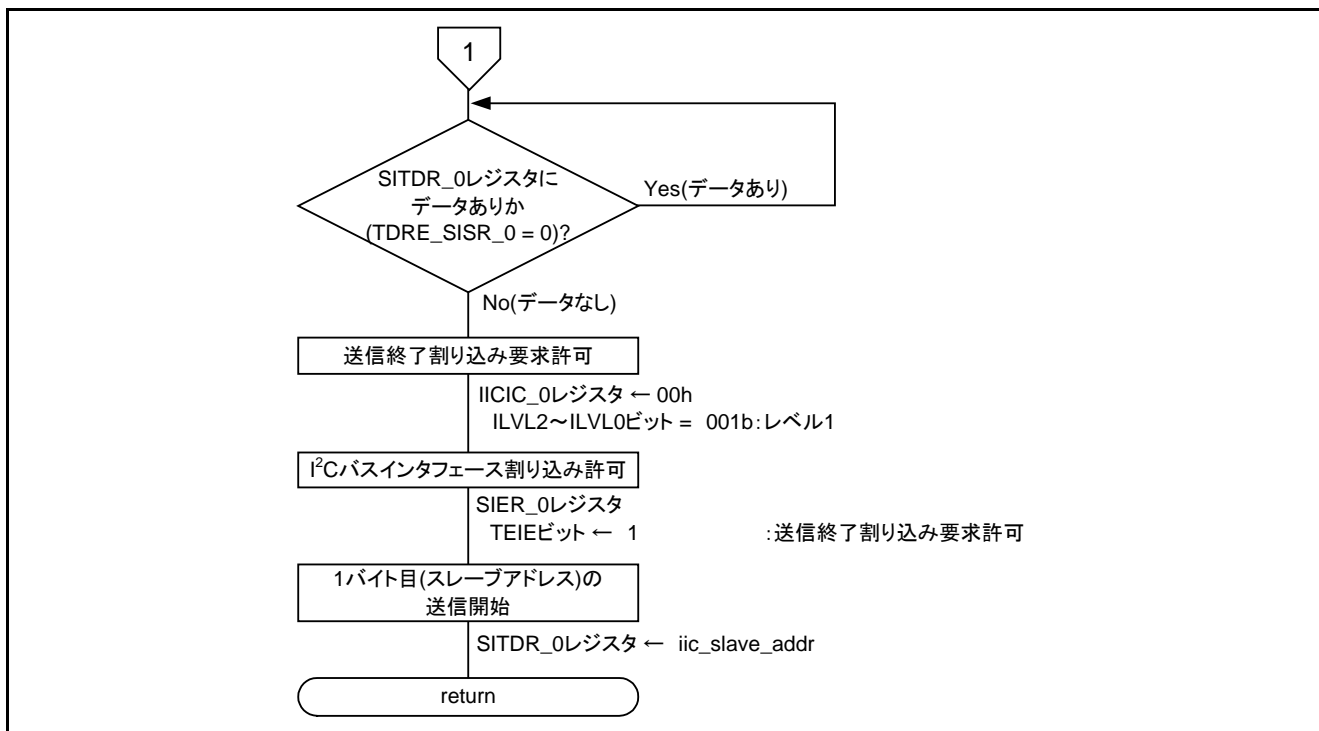


図3.12 マスタ制御開始処理(2)

3.7.5 I<sup>2</sup>C バスインタフェース割り込み処理

図 3.13にI<sup>2</sup>C バスインタフェース割り込み処理のフローチャートを示します。

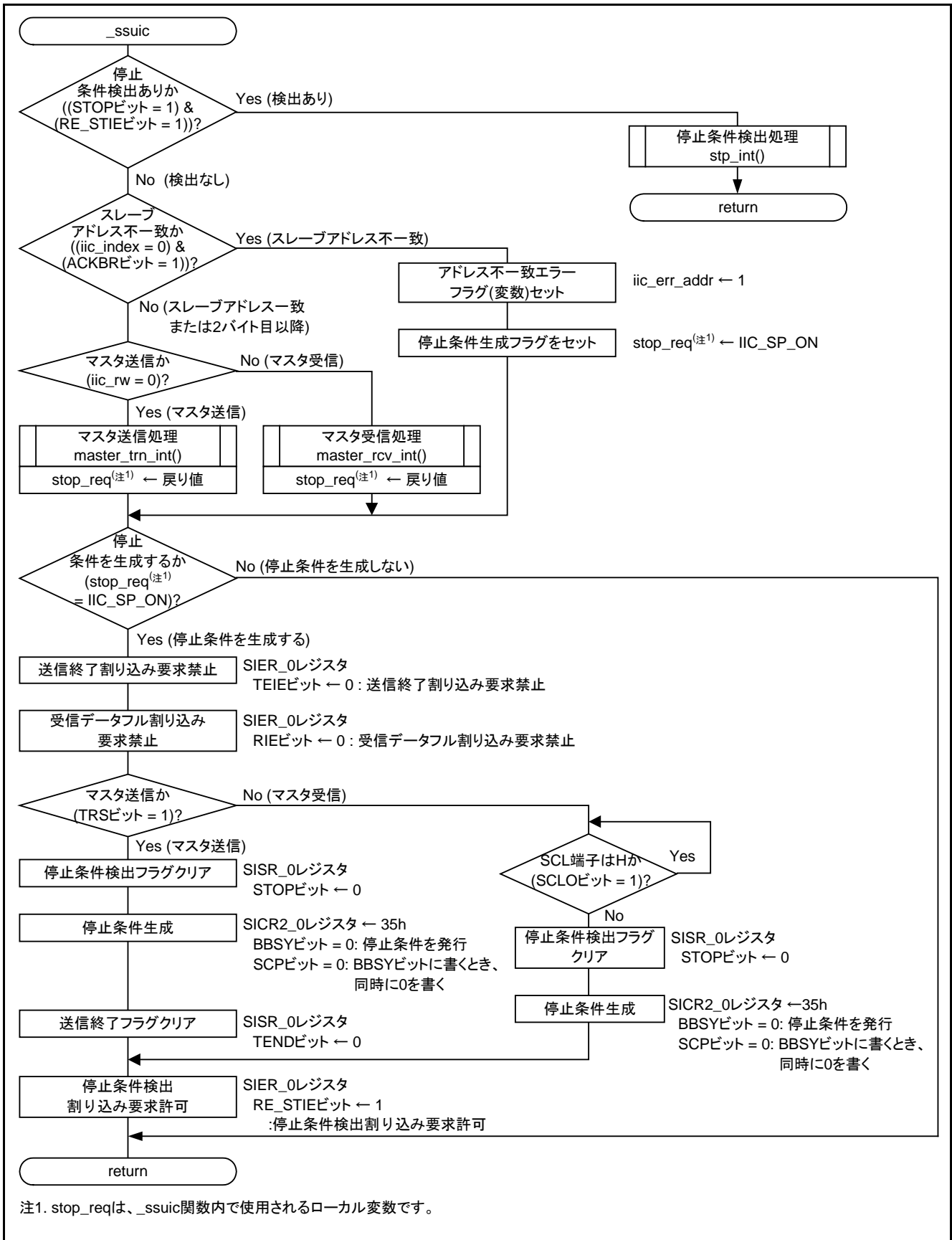


図3.13 I<sup>2</sup>C バスインタフェース割り込み処理

3.7.6 停止条件検出処理

図 3.14に停止条件検出処理のフローチャートを示します。

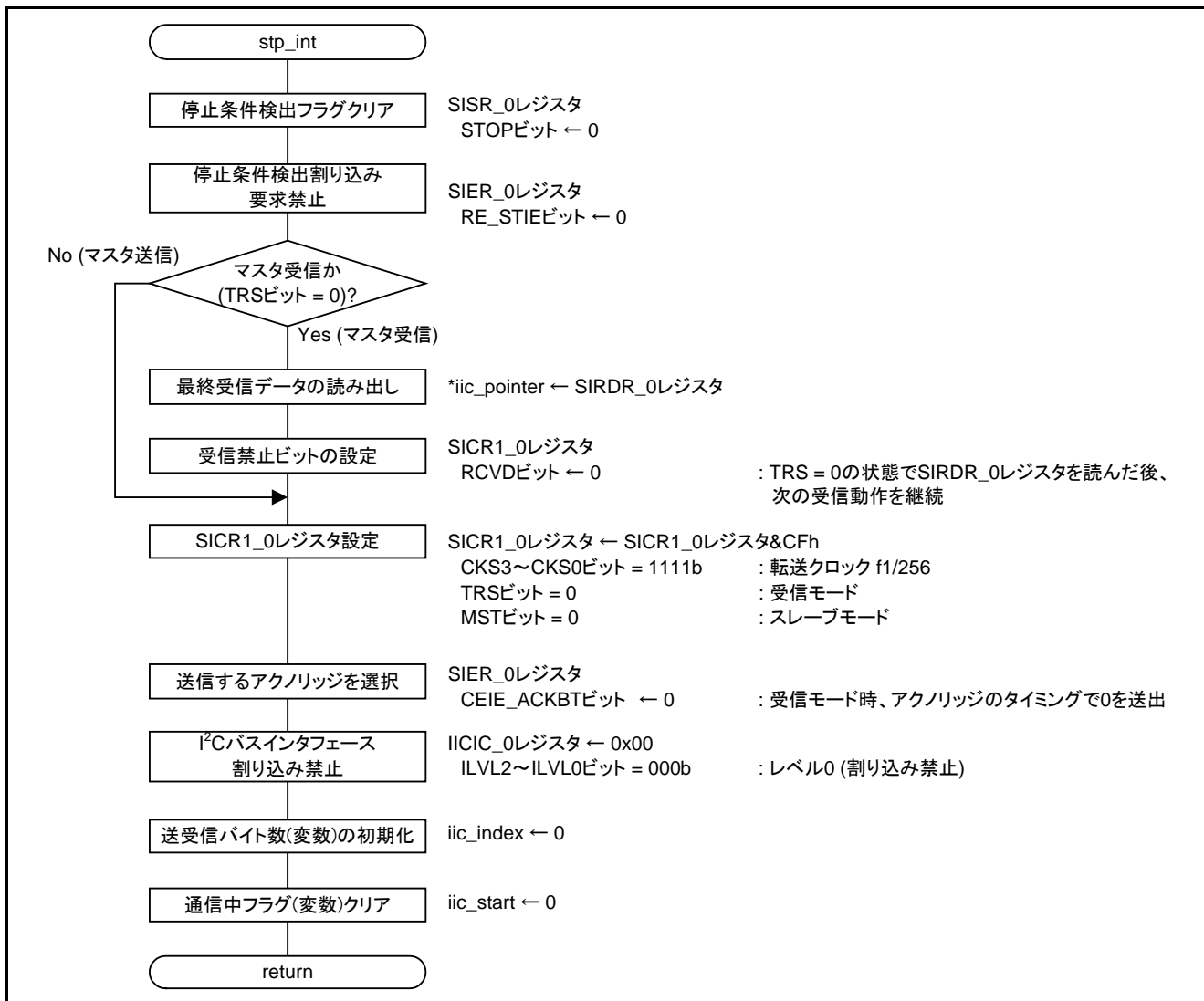


図3.14 停止条件検出処理

## 3.7.7 マスタ送信処理

図 3.15にマスタ送信処理のフローチャートを示します。

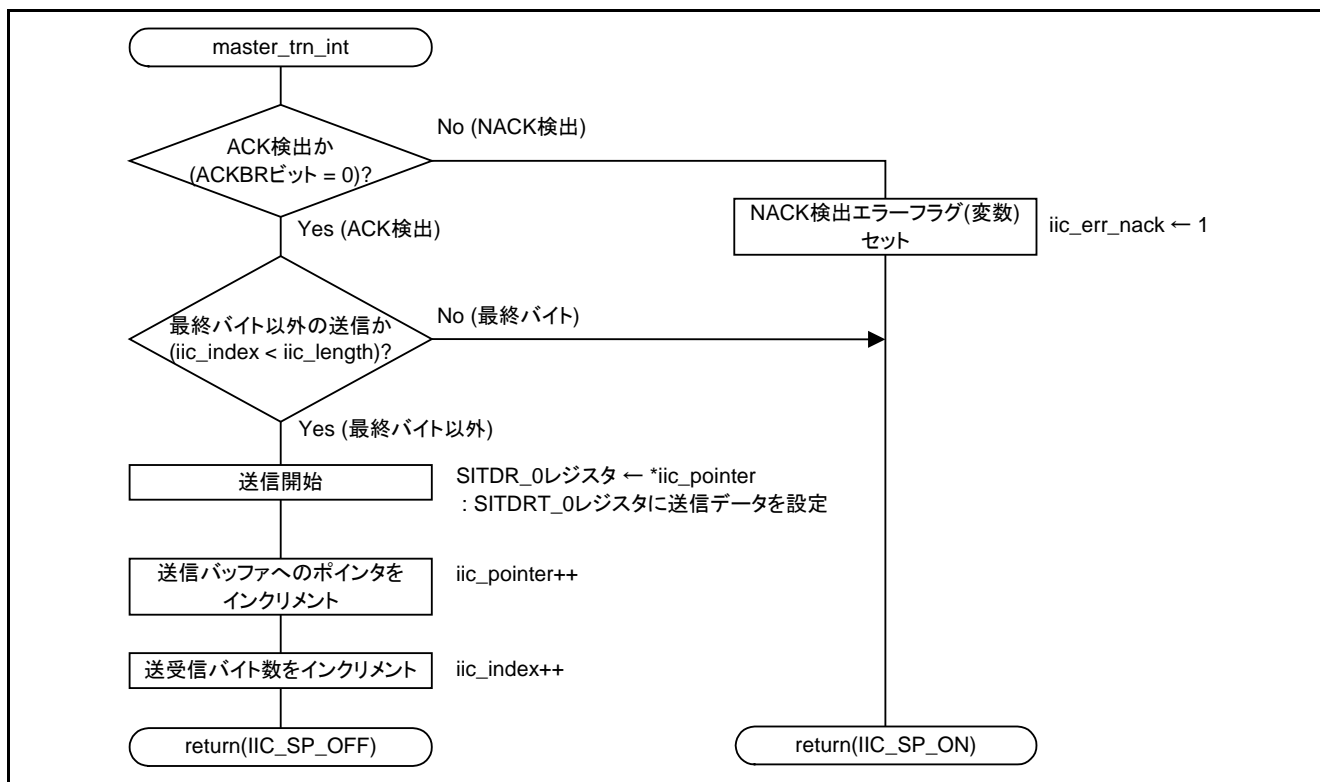


図3.15 マスタ送信処理



3.7.8 マスタ受信処理

図 3.16にマスタ受信処理のフローチャートを示します。

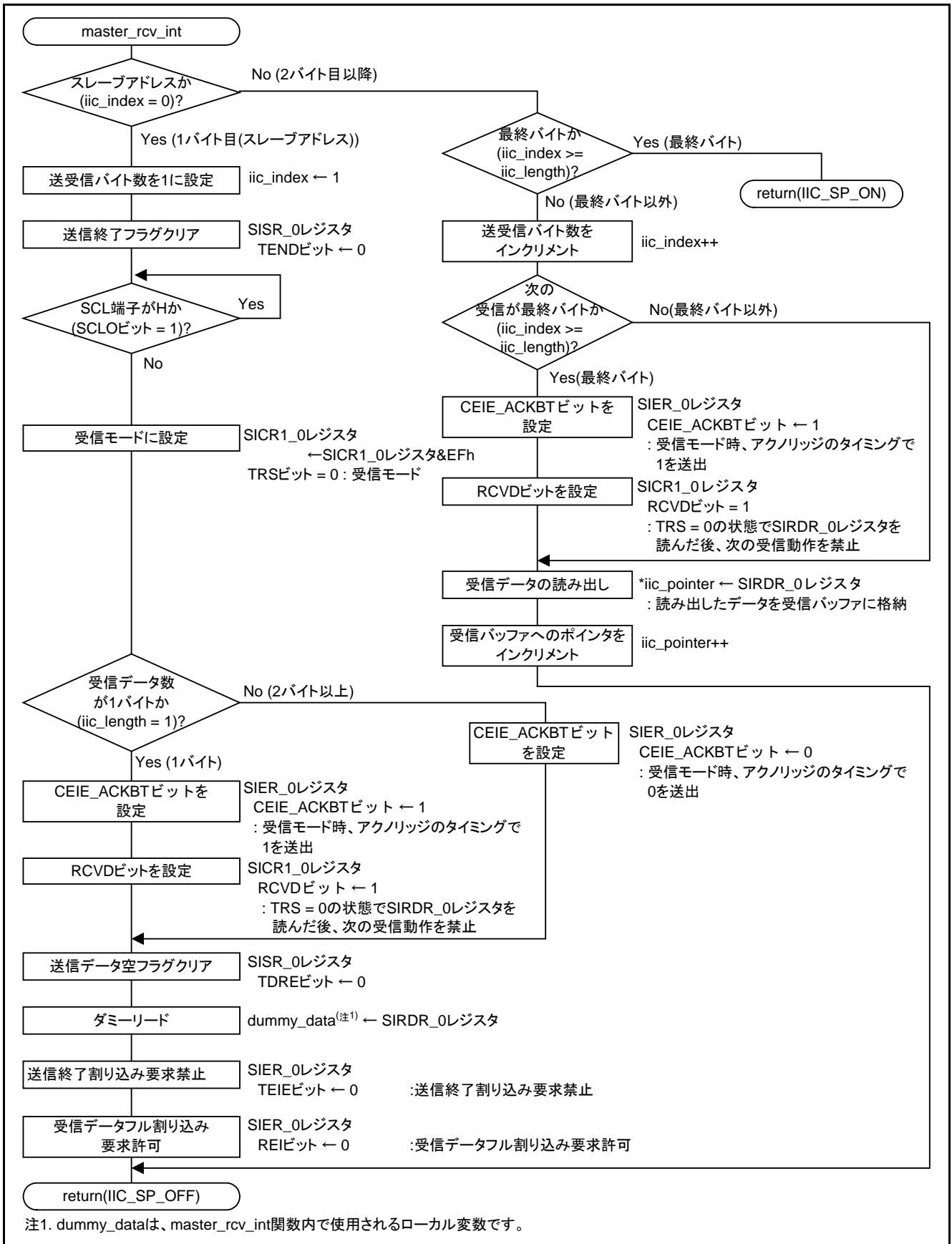


図3.16 マスタ受信処理

## 3.7.9 マスタ制御終了処理

図 3.17にブロック図のフローチャートを示します。

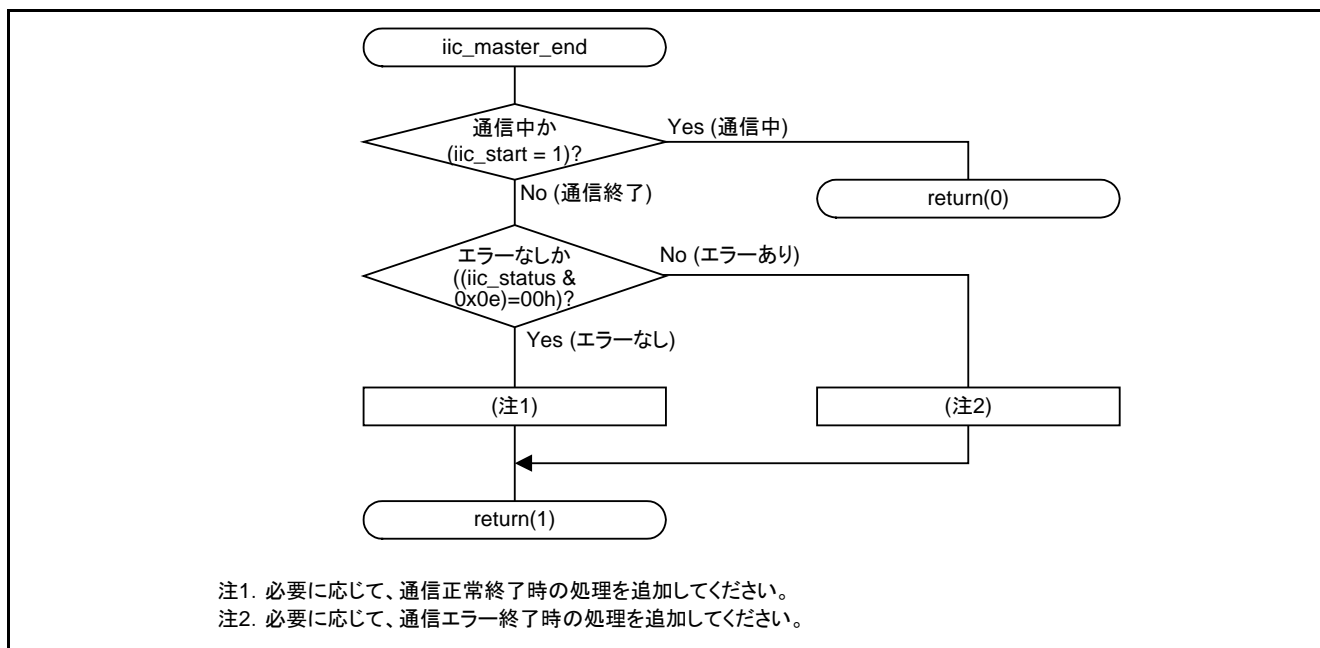


図3.17 マスタ制御終了処理

#### 4. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

#### 5. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

R8C/56E グループユーザーズマニュアル ハードウェア編

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

#### 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/contact/>

## 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（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

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

### 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、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、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て、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は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、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。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。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、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情報に使用しないで行ってください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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/contact/>