

R32C/100 シリーズ

UARTi 特殊モード1(I²Cモード)を使用したEEPROM制御

R01AN0759JJ0101

Rev.1.01

2011.09.30

要旨

本アプリケーションノートでは、UART2 特殊モード1(I²Cモード)を使用したEEPROM制御について説明します。

R32C/118グループでは特殊モード1が使用できるチャンネルは7チャンネル(UART0~UART6)あります。UART2以外のチャンネルを使用する場合は、ユーザーズマニュアルハードウェア編を参照して、UARTi (i=0~6) 関連レジスタを変更してください。

対象デバイス

R32C/116グループ

R32C/117グループ

R32C/118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	5
3. 関連アプリケーションノート	5
4. ハードウェア説明	5
4.1 使用端子一覧	5
5. ソフトウェア説明	6
5.1 動作概要	7
5.1.1 ライトモード時の動作概要	7
5.1.2 リードモード時の動作概要	9
5.2 定数一覧	11
5.3 構造体/共用体一覧	12
5.4 変数一覧	13
5.5 関数一覧	13
5.6 関数仕様	14
5.7 フローチャート.....	19
5.7.1 メイン処理.....	19
5.7.2 UART2初期設定処理	21
5.7.3 マスタ制御開始処理	22
5.7.4 スタートコンディション/ストップコンディション検出割り込み処理	23
5.7.5 スタートコンディション検出処理.....	24
5.7.6 ストップコンディション検出処理.....	25
5.7.7 リスタートコンディション検出処理	27
5.7.8 UART2送信割り込み処理.....	28
5.7.9 メモリアドレス送信処理	29
5.7.10 マスタ送信処理	30
5.7.11 マスタ受信処理	31
5.7.12 INT0、INT1初期設定処理	32
5.7.13 INT0割り込み処理.....	32
5.7.14 INT1割り込み処理.....	32
6. サンプルコード.....	33
7. 参考ドキュメント	33

1. 仕様

UART2特殊モード1(I²Cモード)を使用したEEPROM制御を行います。EEPROMにデータの書き込み(ライトモード)、EEPROMからデータの読み出し(リードモード)を行ないます。

EEPROMはルネサスエレクトロニクス製のR1EX24xxxシリーズを使用します。

下記使用条件においてI²C-busの通信プロトコルに準拠しています。

<使用条件>

- 転送レート：約350kbps (注1)
- 転送データ長：1~256バイト(デバイスアドレスワード(注2)、メモリアドレスを含まず)
- 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対応(リードモード時)

注1. ・設定値は378kbps

(クロック同期化機能を有効にすると、ノイズフィルタ幅+U2BRGカウントソースの1~1.5サイクルのサンプリング遅延が発生し、SCLクロックの“H”認識が遅れるため、SCLクロックの“H”幅が伸びます。そのため、SCLクロックの転送レートの設定に対して、実際のSCLクロックは遅くなります。応用例ではクロック同期化機能を有効にしているため、実際の転送レートは約350kbpsになります。(参考値：プルアップ電圧5V、プルアップ抵抗値1kΩ))

・Standard-Mode、Fast-Mode対応

注2. デバイスアドレスワードはデバイスコード4bit、デバイスアドレスコード3bit、Read/Writeコード1bitの3つのコードで構成されています。

表1.1に使用する周辺機能と用途を、図1.1に接続図を、図1.2に通信フォーマットを示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
シリアルインタフェース(UART2)	EEPROMとの通信
INT0割り込み	モードをライトモードに変更
INT1割り込み	モードをリードモードに変更

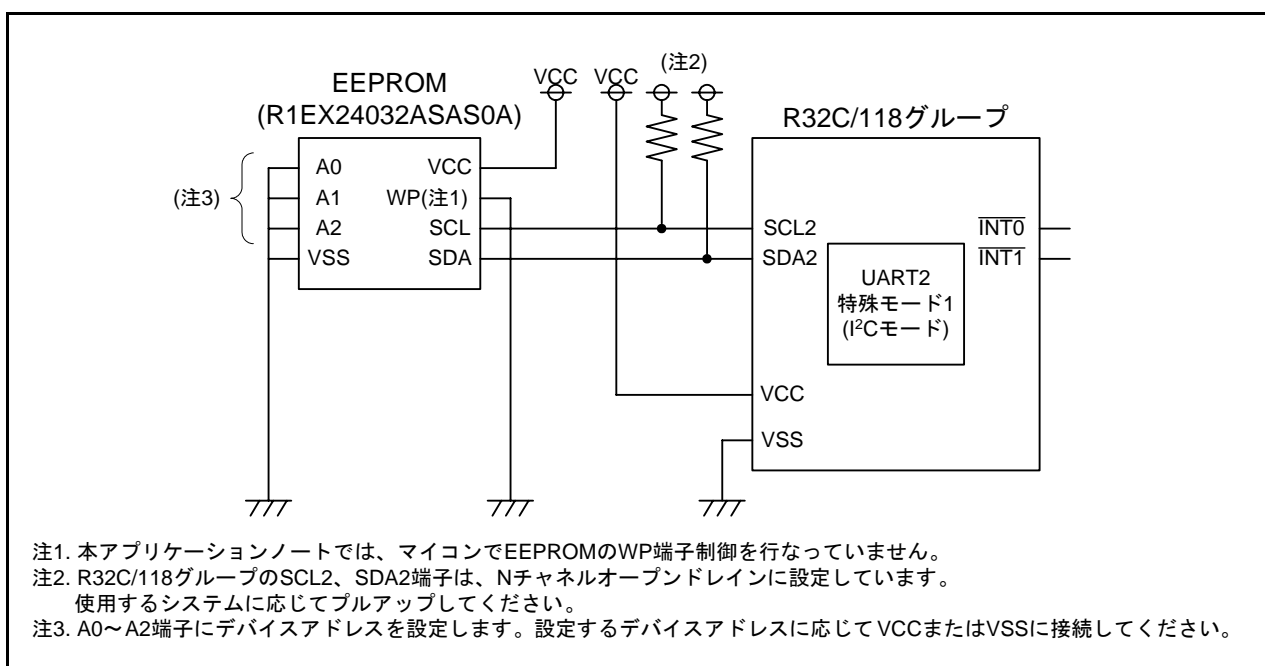


図 1.1 接続図

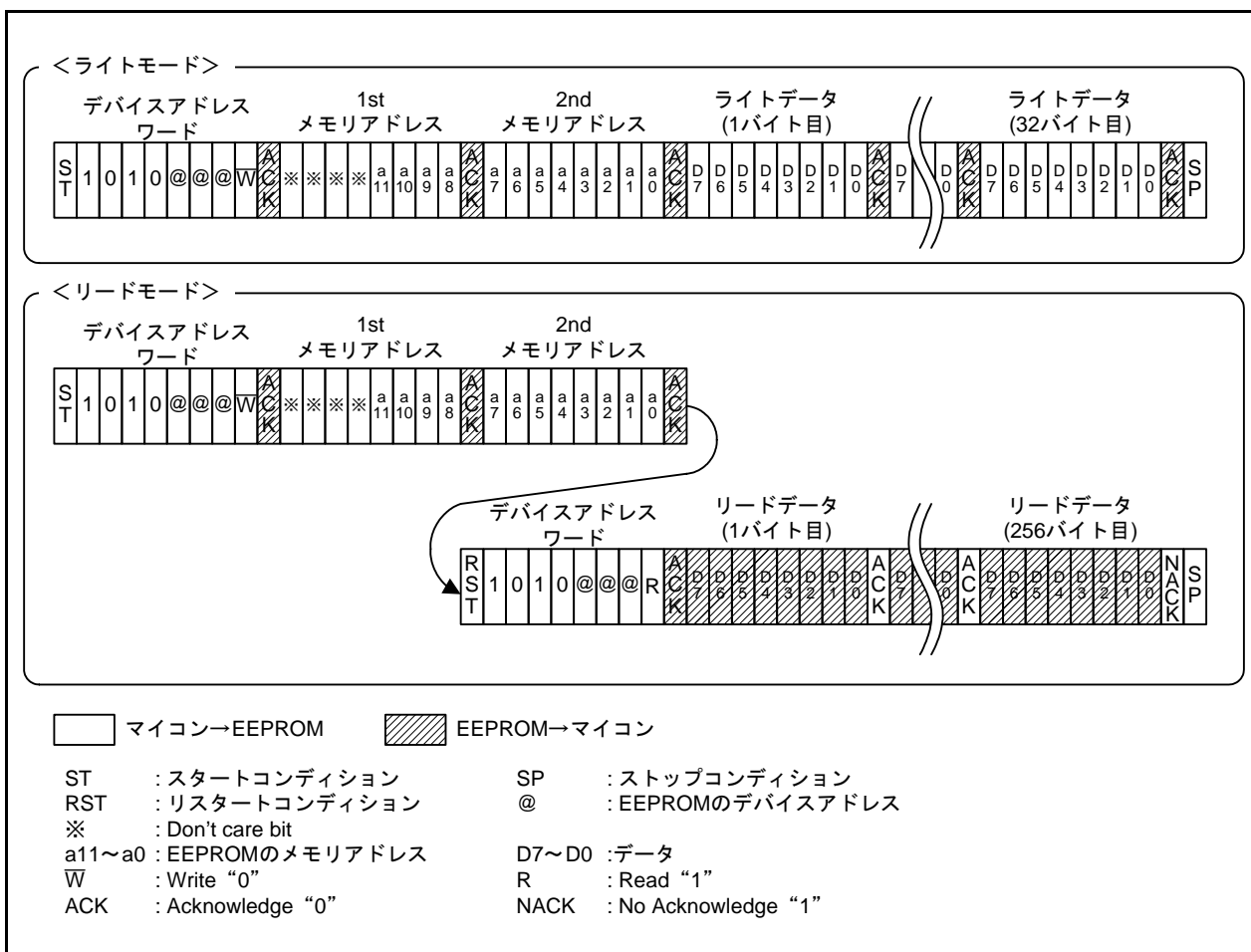


図 1.2 通信フォーマット

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、下記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	R5F64189DFD(R32C/118グループ)
使用デバイス(EEPROM)	R1EX24032ASAS0A
動作周波数	<ul style="list-style-type: none"> ・ XINクロック : 16MHz ・ PLLクロック : 100MHz ・ ベースクロック : 50MHz ・ CPUクロック : 50MHz ・ 周辺バスクロック : 25MHz ・ 周辺機能クロック : 25MHz
動作電圧	5V
統合開発環境	ルネサス エレクトロニクス製 High-performance Embedded Workshop Version 4.08
Cコンパイラ	ルネサス エレクトロニクス製 R32C/100 Series C Compiler V.1.02 Release 01 コンパイルオプション -D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300 -DVECTOR_ADR=0x0FFFFFFBDC -c -finfo -dir "\$(CONFIGDIR)" (統合開発環境のデフォルト設定を使用しています)
動作モード	シングルチップモード
サンプルコードのバージョン	1.00

3. 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。併せて参照してください。

- ・ R32C/100シリーズメインクロック通倍モード設定手順(RJJ05B1259)
- ・ M16Cファミリ、R8Cファミリ UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)
- ・ R32C/100シリーズ UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1587)

4. ハードウェア説明

4.1 使用端子一覧

表 4.1に使用端子と機能を示します。

表 4.1 使用端子と機能

端子名	入出力	内容
P7_0/SDA2	入出力	I ² Cモードのデータ入出力
P7_1/SCL2	出力	I ² Cモードのクロック出力
P8_2/INT0	入力	INT0割り込みの入力
P8_3/INT1	入力	INT1割り込みの入力

5. ソフトウェア説明

R32C/118グループのマイコンを使用し、EEPROMに256バイト書き込み(ライトモード)またはEEPROMから256バイト読み出し(リードモード)を行ないます。ライトモードでは32バイト(1ページ)書き込みを8回繰り返して256バイト書き込みを行ないます。INT0割り込みが発生するとライトモードに変更します。INT1割り込みが発生するとリードモードに変更します。

シリアルインタフェース(UART2)の特殊モード1(I²Cモード)を以下の設定条件で使用します。

<設定条件>

- I²Cモードを使用します。
- 転送クロックは、内部クロックを使用します。
- U2BRGカウンタソースは、f1を使用します。
- SDA2、SCL2端子は、Nチャンネルオープンドレイン出力を使用します。
- 転送フォーマットは、MSBファーストを使用します。
- UART2送信割り込み要因は、送信完了(TXEPT=1)を使用します。
- クロック位相設定はクロック遅れありを選択します。
- SDA2デジタル遅延値は、U2BRGカウンタソースの7~8サイクルを使用します。
- クロック同期化機能を使用します。
- SCL2ウェイト出力機能2は使用しません。
- SDA2出力停止機能は使用しません。
- スタートコンディション/ストップコンディション検出割り込みを使用します。
- UART2送信割り込みを使用します。
- UART2受信割り込みは使用しません。
- 転送レートは、約378kbpsに設定します。

<転送レートの計算式>

$$\begin{aligned}\text{転送レート} &= \text{U2BRGカウンタソース} \div (2 \times (\text{U2BRGレジスタの設定値} + 1)) \\ &= 25\text{MHz}(f1) \div (2 \times (32 + 1)) \\ &\approx 378.788\text{kbps}\end{aligned}$$

5.1 動作概要

5.1.1 ライトモード時の動作概要

- (1) 初期設定
システムクロック、UART2関連SFR、INT割り込み関連SFRの初期設定を行いません。
INT0割り込みが発生すると、RAM上の変数(mode)にライトモードを設定し、スタートコンディションを生成します。
- (2) スタートコンディション/ストップコンディション検出割り込み
スタートコンディション生成終了時、スタートコンディション/ストップコンディション検出割り込みが発生します。U2TBレジスタにデバイスアドレスワードを設定し、送信を開始します。
- (3) UART2送信割り込み
SCLクロックの9ビット目の立ち下がりで、UART2送信割り込みが発生します。
UART2送信割り込み処理内でACKを受信したことを確認してから、U2TBレジスタに1stメモリアドレスを設定します。
- (4) UART2送信割り込み
UART2送信割り込み処理内でACKを受信したことを確認してから、U2TBレジスタに2ndメモリアドレスを設定します。
- (5) UART2送信割り込み
UART2送信割り込み処理内でACKを受信したことを確認してから、U2TBレジスタにライトデータを設定します。32バイト送信するまで(5)を繰り返します。
- (6) UART2送信割り込み
32バイト送信ごとに、UART2送信割り込み処理内でストップコンディションを生成します。
- (7) スタートコンディション/ストップコンディション検出割り込み
ストップコンディション検出時、スタートコンディション/ストップコンディション検出割り込みが発生し、ストップコンディション検出処理を行いません。
- (8) EEPROMへの書き込み待ち
ストップコンディション検出処理内で、約5msのソフトウェアループを使用してEEPROMの書き込み完了を待ちます。その後ライトデータが残っている場合は、スタートコンディションを生成します。

256バイト送信するまで(2)~(8)を繰り返します。

256バイト送信後は、INT0、INT1割り込みの発生待ちになります。

図 5.1にライトモード時のタイミング図を示します。

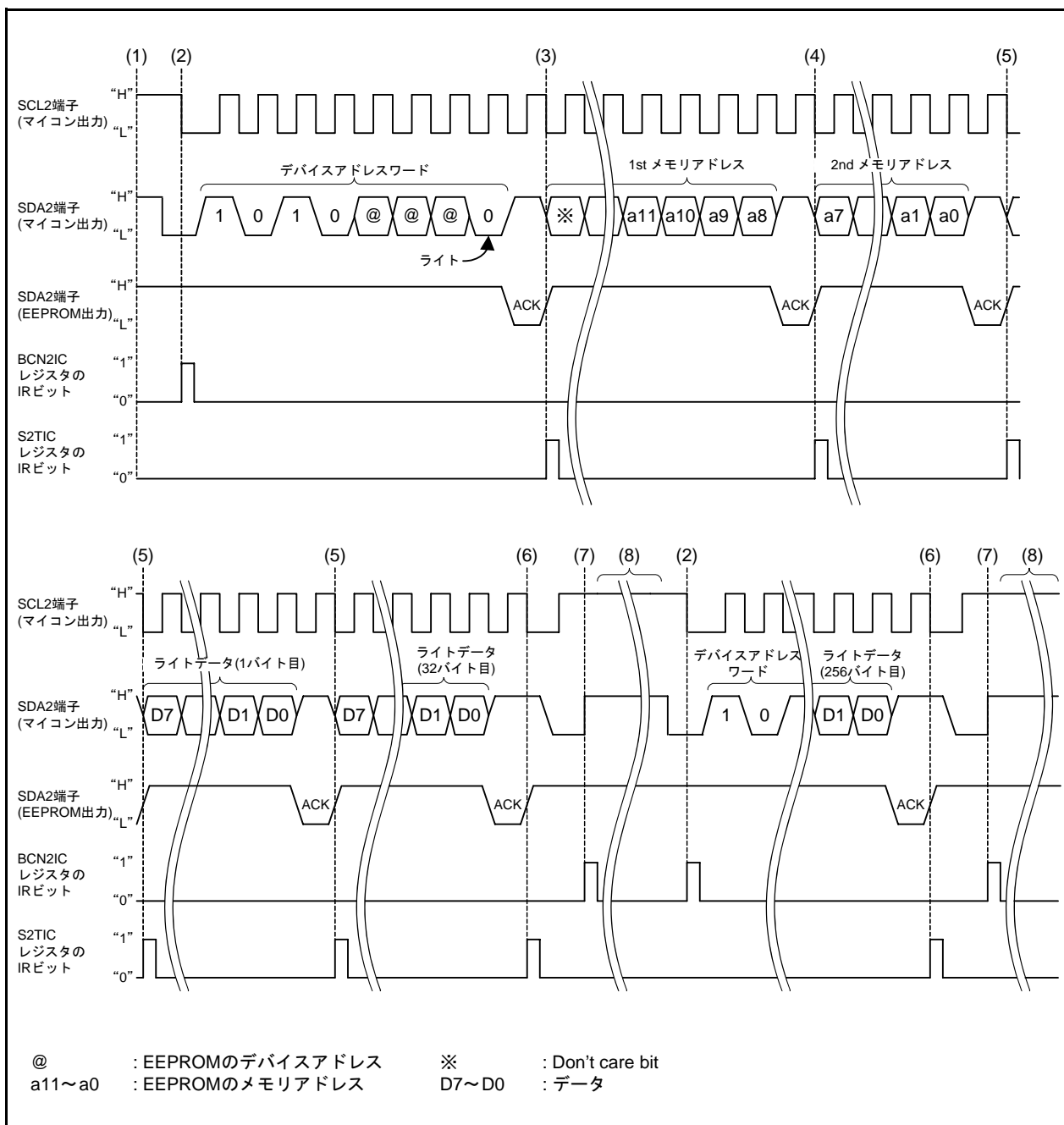


図 5.1 ライトモード時のタイミング図

5.1.2 リードモード時の動作概要

- (1) 初期設定
システムクロック、UART2関連SFR、INT割り込み関連SFRの初期設定を行いません。
INT1割り込みが発生すると、RAM上の変数(mode)にリードモードを設定し、スタートコンディションを生成します。
- (2) スタートコンディション/ストップコンディション検出割り込み
スタートコンディション生成終了時、スタートコンディション/ストップコンディション検出割り込みが発生します。U2TBレジスタにデバイスアドレスワードを設定し、送信を開始します。(このときのデバイスアドレスワードの8ビット目は“0”(ライトコード)です。)
- (3) UART2送信割り込み
SCLクロックの9ビット目の立ち下がり、UART2送信割り込みが発生します。
UART2送信割り込み処理内でACKを受信したことを確認してから、U2TBレジスタに1stメモリアドレスを設定します。
- (4) UART2送信割り込み
UART2送信割り込み処理内でACKを受信したことを確認してから、U2TBレジスタに2ndメモリアドレスを設定します。
- (5) UART2送信割り込み
UART2送信割り込み処理内でリスタートコンディションを生成します。
- (6) スタートコンディション/ストップコンディション検出割り込み
リスタートコンディション生成終了時、スタートコンディション/ストップコンディション検出割り込みが発生します。U2TBレジスタにデバイスアドレスワードを設定し、送信を開始します。(このときのデバイスアドレスワードの8ビット目は“1”(リードコード)です。)
- (7) UART2送信割り込み
次のデータを受信するために、U2TBレジスタにダミーデータとACKデータを設定します。
- (8) UART2送信割り込み
リードデータをU2RBレジスタから読み出し、受信データバッファに書き込みます。
次のデータを受信するために、U2TBレジスタにダミーデータとACKデータを設定します。
254バイト受信するまで(8)を行いません。
- (9) UART2送信割り込み
リードデータをU2RBレジスタから読み出し、受信データバッファに書き込みます。
256バイト目のデータ(最後のデータ)を受信するために、U2TBレジスタにダミーデータとNACKデータを設定します。
- (10) UART2送信割り込み
リードデータをU2RBレジスタから読み出し、受信データバッファに書き込みます。UART2送信割り込み処理内で、ストップコンディションを生成します。
- (11) スタートコンディション/ストップコンディション検出割り込み
ストップコンディション検出時、スタートコンディション/ストップコンディション検出割り込みが発生し、ストップコンディション検出処理を行いません。
INT0、INT1割り込みの発生待ちになります。

図 5.2にリードモード時のタイミング図を示します。

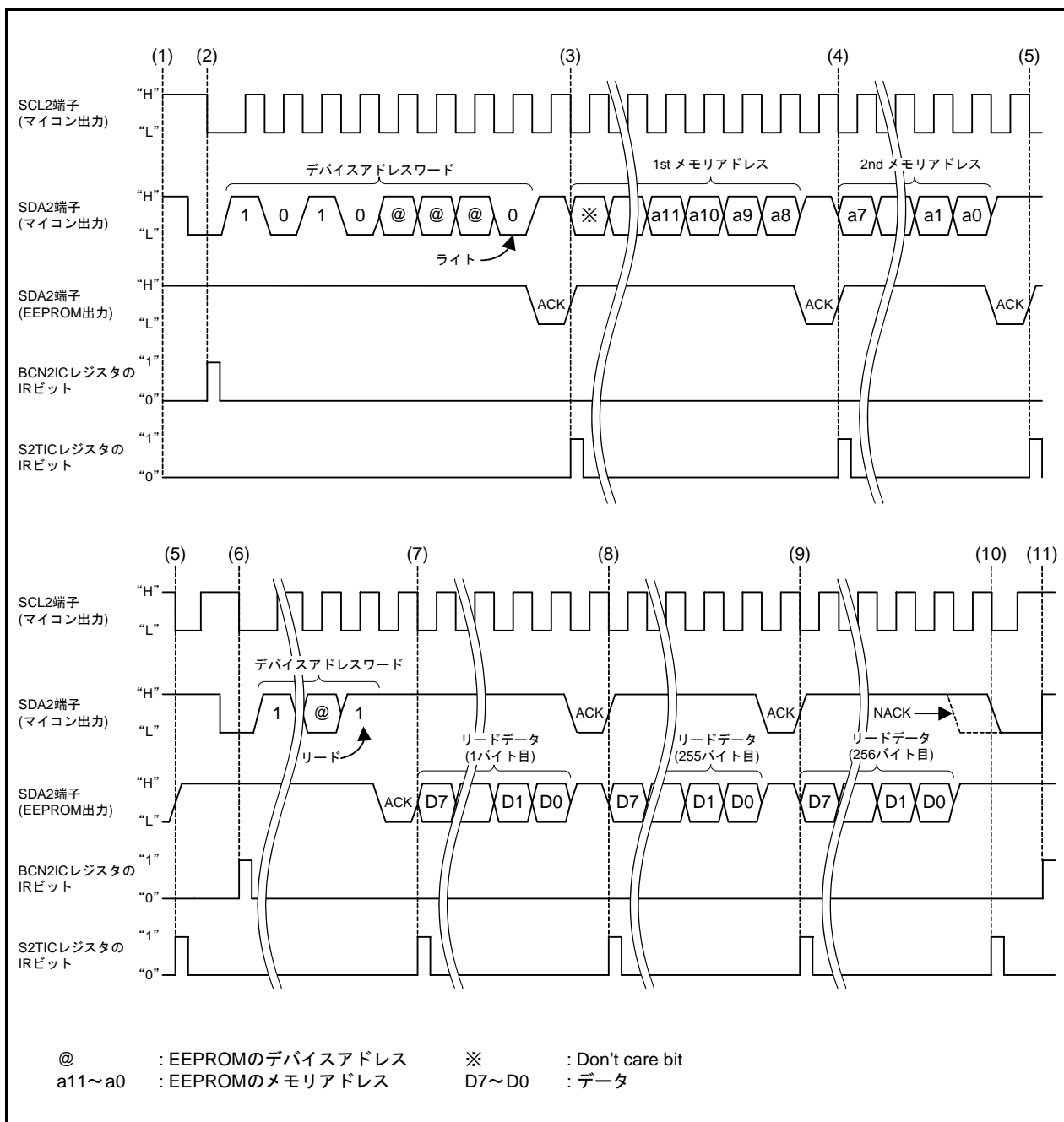


図 5.2 リードモード時のタイミング図

5.2 定数一覧

表 5.1にサンプルコードで使用する定数を示します。

表 5.1 サンプルコードで使用する定数

定数名	設定値	内容
DEVICE_ADDR_WORD	1010 0000b	デバイスアドレスワード b7-b4: デバイスコード、b3-b1: スレーブアドレス b0: R/Wコード
EEPROM_MEM_ADDR	0000h	EEPROMのメモリアドレス
LENGTH	32	1ページの送信サイズ
W_SIZE	256	送信データのサイズ($W_SIZE \leq BUFSIZE$)
R_SIZE	256	受信データのサイズ($R_SIZE \leq BUFSIZE$)
BUFSIZE	256	送受信データバッファサイズ
PAGE	(注1)	送信ページ数
IIC_BRG	(33 - 1)	$25\text{MHz} / (2 \times (32 + 1)) = \text{約}378\text{kbps}$
WAIT_LOOP	50000	EEPROMの書き込み待ち時間用
DISABLE	0	禁止
ENABLE	1	許可
BUSY	0	通信中
RDY	1	通信終了
PAR_ERR	FFh	パラメータエラー
IDLE	FFh	アイドル状態
WRITE	0	R/W(ライトモード)
READ	1	R/W(リードモード)
R_CODE	1	EEPROMのリードコード
W_CODE	0	EEPROMのライトコード
IIC_SP_ON	1	ストップコンディション生成する
IIC_SP_OFF	0	ストップコンディション生成しない
PD_IIC	pd7	PD7_0: SDA2、PD7_1: SCL2
PD_IIC_INIT	0000 0011b	PD7_0: 出力、PD7_1: 出力
PD_IIC_INIT_INPUT	1111 1100b	PD7_0: 入力、PD7_1: 入力

注1. $((W_SIZE + LENGTH - 1) / LENGTH)$

5.3 構造体/共用体一覧

図 5.3 にサンプルコードで使用する構造体/共用体を示します。

```

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;

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;
    unsigned char b8:1;
    unsigned char b9:1;
    unsigned char b10:1;
    unsigned char b11:1;
    unsigned char b12:1;
    unsigned char b13:1;
    unsigned char b14:1;
    unsigned char b15:1;
  }bit;
  struct{
    unsigned char byte0;
    unsigned char byte1;
  }byte;
  unsigned short all;
}word_dt;

byte_dt iic_str1;
#define device_addr_word iic_str1.all /* デバイスアドレスワード */
#define iic_rw iic_str1.bit.b0 /* b7-b4: デバイスコード(fixed) b3-b1: デバイスアドレスコード b0: R/Wコード */
/* 0: Write(Master Transmit) 1: Read(Master Receive) */

byte_dt iic_str2;
#define iic_status iic_str2.all /* ステータス */
/* すべてのステータス */
#define iic_start iic_str2.bit.b0 /* 1: 通信中 0: 通信終了 */
#define iic_err_par iic_str2.bit.b1 /* 1: パラメーターエラー 0: エラーなし */
#define iic_err_nack iic_str2.bit.b2 /* 1: NACK検出エラー 0: エラーなし */

```

図 5.3 サンプルコードで使用する構造体/共用体

5.4 変数一覧

表 5.2 にサンプルコードで使用するグローバル変数を示します。

表 5.2 グローバル変数

型	変数名	内容	使用関数
unsigned short	iic_length	転送データ長	iic_master_start, master_trn_int, master_rcv_int
unsigned char	trn_data[BUFSIZE]	送信データバッファ	main, stp_int
unsigned char	rcv_data[BUFSIZE]	受信データバッファ	main
unsigned short	mem_addr	メモリアドレスバッファ	main, stp_int, mem_addr_trn_int
unsigned char	mode	モードを保存	main, _int0, _int1
unsigned char far	*ptr_buf	バッファのポインタ	iic_master_start, master_trn_int, master_rcv_int
unsigned short	cnt_tr_index	送受信バイト数	sta_int, stp_int, _uart2_trans, master_trn_int, master_rcv_int
unsigned short	cnt_remain_t_index	残り送信データカウンタ	iic_master_start, stp_int
unsigned char	cnt_mem_addr	メモリアドレス送信カウンタ	_start_stop_condition_detection, sta_int, stp_int, _uart2_trans
unsigned char	cnt_page	ページカウンタ	iic_master_start, stp_int

5.5 関数一覧

表 5.3 にサンプルコードで使用する関数を示します。

表 5.3 関数

関数名	概要
main	メイン処理
uart2_init	UART2 初期設定処理
iic_master_start	マスタ制御開始処理
_start_stop_condition_detection	スタートコンディション/ストップコンディション検出割り込み処理
sta_int	スタートコンディション検出処理
stp_int	ストップコンディション検出処理
re_sta_int	リスタートコンディション検出処理
_uart2_trans	UART2 送信割り込み処理
mem_addr_trn_int	メモリアドレス送信処理
master_trn_int	マスタ送信処理
master_rcv_int	マスタ受信処理
int_init	INT0、INT1 初期設定処理
_int0	INT0 割り込み処理
_int1	INT1 割り込み処理

5.6 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	なし
宣言	void main(void)
説明	システムクロックとUART2の初期設定、INT0、INT1初期設定後、INT0、INT1割り込みが発生するのを待ちます。RAM上の変数(mode)の値が、ライトモードの時、EEPROMへのデータの書き込み、リードモードの時、EEPROMからデータの読み出しを行います。
引数	なし
リターン値	なし
備考	

uart2_init

概要	UART2初期設定処理
ヘッダ	なし
宣言	void uart2_init(unsigned char init)
説明	UART2特殊モード1(I ² Cモード)の設定を行います。
引数	●第一引数 : init : I ² Cモード有効/無効
リターン値	なし
備考	

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 short len)
説明	送受信を開始するための処理を行います。 この関数の先頭で全ステータスを初期化し、引数の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、戻り値“PAR_ERR”を返します。パラメータエラーの場合はマスタ制御開始処理は行いません。 次にバス状態を確認します。 ・バスビジーの場合、戻り値“BUSY”を返します。マスタ制御開始処理は行いません。 ・バスフリーの場合、戻り値“RDY”を返します。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、スタートコンディションを生成します。
引数	●第一引数 : addr : デバイスアドレスワード ●第二引数 : rw : ライトモード/リードモード ●第三引数 : *buf : 送信または受信バッファへのポインタ ●第四引数 : len : 転送データ長
リターン値	●バスビジーの場合 : BUSY ●バスフリーの場合 : RDY ●パラメータエラーの場合 : PAR_ERR
備考	

_start_stop_condition_detection

概要	スタートコンディション/ストップコンディション検出割り込み処理
ヘッダ	なし
宣言	void _start_stop_condition_detection(void)
説明	スタートコンディション/ストップコンディション検出時に割り込みが発生します。スタートコンディション生成終了時はsta_int関数を、リスタートコンディション生成終了時はre_sta_int関数を、ストップコンディション検出時はstp_int関数を呼び出します。
引数	なし
リターン値	なし
備考	

sta_int

概要	スタートコンディション検出処理
ヘッダ	なし
宣言	void sta_int(void)
説明	スタートコンディション/ストップコンディション検出割り込み処理から呼び出されます。送受信許可に設定後、デバイスアドレスワードの送信を行います。
引数	なし
リターン値	なし
備考	

stp_int

概要	ストップコンディション検出処理
ヘッダ	なし
宣言	void stp_int(void)
説明	スタートコンディション/ストップコンディション検出割り込み処理から呼び出されます。送受信するデータが残っていない場合は、通信中に変更したUART2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。INT0、INT1割り込みを許可に設定します。
引数	なし
リターン値	なし
備考	

re_sta_int

概要	リスタートコンディション検出処理
ヘッダ	なし
宣言	void re_sta_int(void)
説明	スタートコンディション/ストップコンディション検出割り込み処理から呼び出されます。リードモード時、デバイスアドレスワードの送信を行います。
引数	なし
リターン値	なし
備考	

uart2_trans

概要	UART2送信割り込み処理
ヘッダ	なし
宣言	void uart2_trans(void)
説明	<p>SCLクロックの9ビット目の立ち下がりで割り込みが発生します。この関数の先頭でU2RBレジスタを読み出します。</p> <ul style="list-style-type: none"> ●デバイスアドレスワード送信時 NACKを検出した場合は、ストップコンディションを生成します。 ●メモリアドレス送信前 mem_addr_trn_int関数を呼び出します。 ●メモリアドレス送信後 <ul style="list-style-type: none"> ・ライトモード時：master_trn_int関数を呼び出します。 ・リードモード時(リスタートコンディション生成前)： リスタートコンディションを生成します。 ・リードモード時(リスタートコンディション生成後)： master_rcv_int関数を呼び出します。 <p>通信を終了するときはストップコンディションを生成します。</p>
引数	なし
リターン値	なし
備考	

mem_addr_trn_int

概要	メモリアドレス送信処理
ヘッダ	なし
宣言	unsigned char mem_addr_trn_int(unsigned short rb_data, unsigned char cnt, unsigned char len)
説明	<p>UART2送信割り込み処理から呼び出される関数です。メモリアドレスの送信を行います。</p> <ul style="list-style-type: none"> ●1バイト目のメモリアドレス送信時 1バイト目のメモリアドレスの送信のみを行います。 ●2バイト目のメモリアドレスを送信時 2バイト目のメモリアドレスを送信後、次に送信するメモリアドレスの更新を行います。
引数	<ul style="list-style-type: none"> ●第一引数：rb_data : U2RBレジスタから読み出したデータ ●第二引数：cnt : 送信メモリアドレスのカウント数 ●第三引数：len : 転送データ長
リターン値	<ul style="list-style-type: none"> ●ストップコンディション生成する場合：IIC_SP_ON ●ストップコンディション生成しない場合：IIC_SP_OFF
備考	

master_trn_int

概要	マスタ送信処理
ヘッダ	なし
宣言	unsigned char master_trn_int(unsigned short rb_data)
説明	<p>UART2送信割り込み処理から呼び出される関数です。</p> <p>次の場合、戻り値“IIC_SP_OFF”を返します。</p> <ul style="list-style-type: none"> ・ACK検出、かつ最終バイト以外(次の送信を開始します) <p>次の場合、戻り値“IIC_SP_ON”を返します。</p> <ul style="list-style-type: none"> ・NACK検出時(NACK検出エラーフラグを“1”に設定します) ・最終バイト送信完了時
引数	●第一引数 : rb_data : U2RBレジスタから読み出したデータ
リターン値	<ul style="list-style-type: none"> ●ストップコンディション生成する場合 : IIC_SP_ON ●ストップコンディション生成しない場合 : IIC_SP_OFF
備考	

master_rcv_int

概要	マスタ受信処理
ヘッダ	なし
宣言	unsigned char master_rcv_int(unsigned short rb_data)
説明	<p>UART2送信割り込み処理から呼び出される関数です。</p> <p>引数の値を受信データバッファに格納します(デバイスアドレスワード時は除く)。次が最終バイトであればNACKを、最終バイト以外ならACKを設定して次の受信を開始します。</p> <p>次の場合、戻り値“IIC_SP_OFF”を返します。</p> <ul style="list-style-type: none"> ・最終バイト以外 <p>次の場合、戻り値“IIC_SP_ON”を返します。</p> <ul style="list-style-type: none"> ・最終バイト受信完了時
引数	●第一引数 : rb_data : U2RBレジスタから読み出したデータ
リターン値	<ul style="list-style-type: none"> ●ストップコンディション生成する場合 : IIC_SP_ON ●ストップコンディション生成しない場合 : IIC_SP_OFF
備考	

int_init

概要	INT0、INT1初期設定処理
ヘッダ	なし
宣言	void int_init(void)
説明	INT0、INT1割り込み優先レベルを“1”に設定します。
引数	なし
リターン値	なし
備考	

_int0

概要	INT0割り込み処理
ヘッダ	なし
宣言	void _int0(void)
説明	INT0、INT1割り込み禁止にしてライトモードに設定します。
引数	なし
リターン値	なし
備考	

_int1

概要	INT1割り込み処理
ヘッダ	なし
宣言	void _int1(void)
説明	INT0、INT1割り込み禁止にしてリードモードに設定します。
引数	なし
リターン値	なし
備考	

5.7 フローチャート

5.7.1 メイン処理

図 5.4~図 5.5にメイン処理のフローチャートを示します。

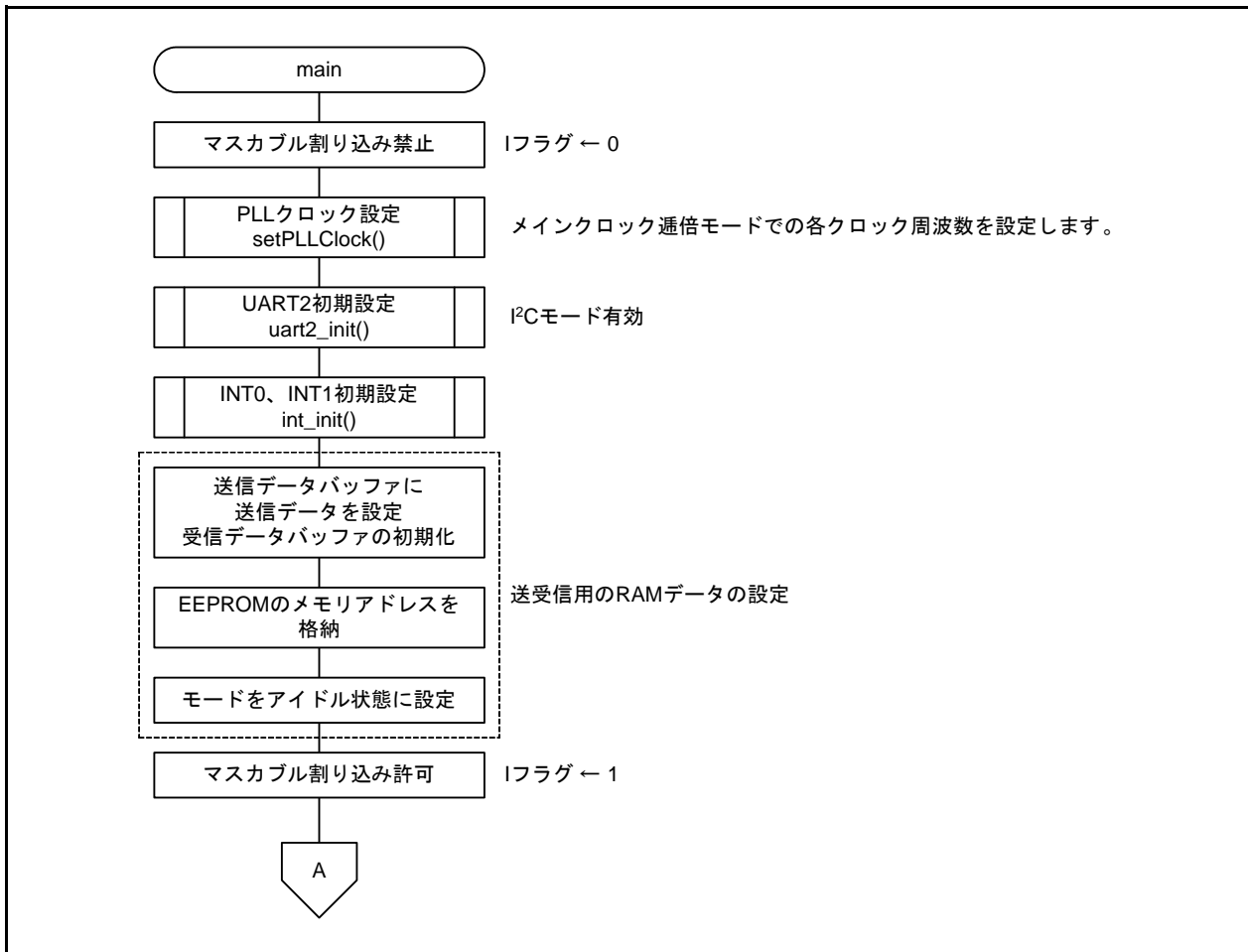


図 5.4 メイン処理(1/2)

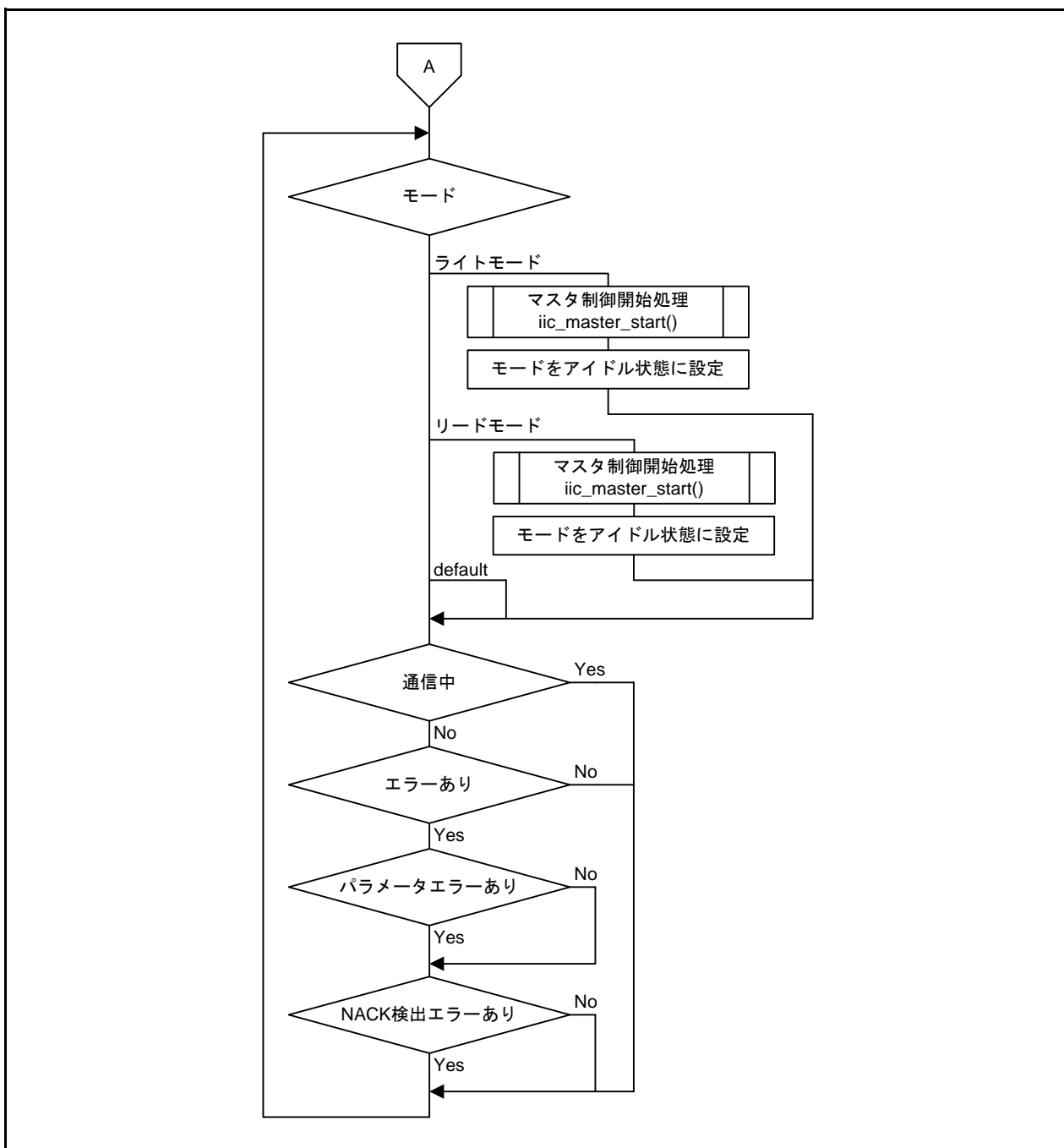


図 5.5 メイン処理 (2/2)

5.7.2 UART2 初期設定処理

図 5.6 に UART2 初期設定処理のフローチャートを示します。

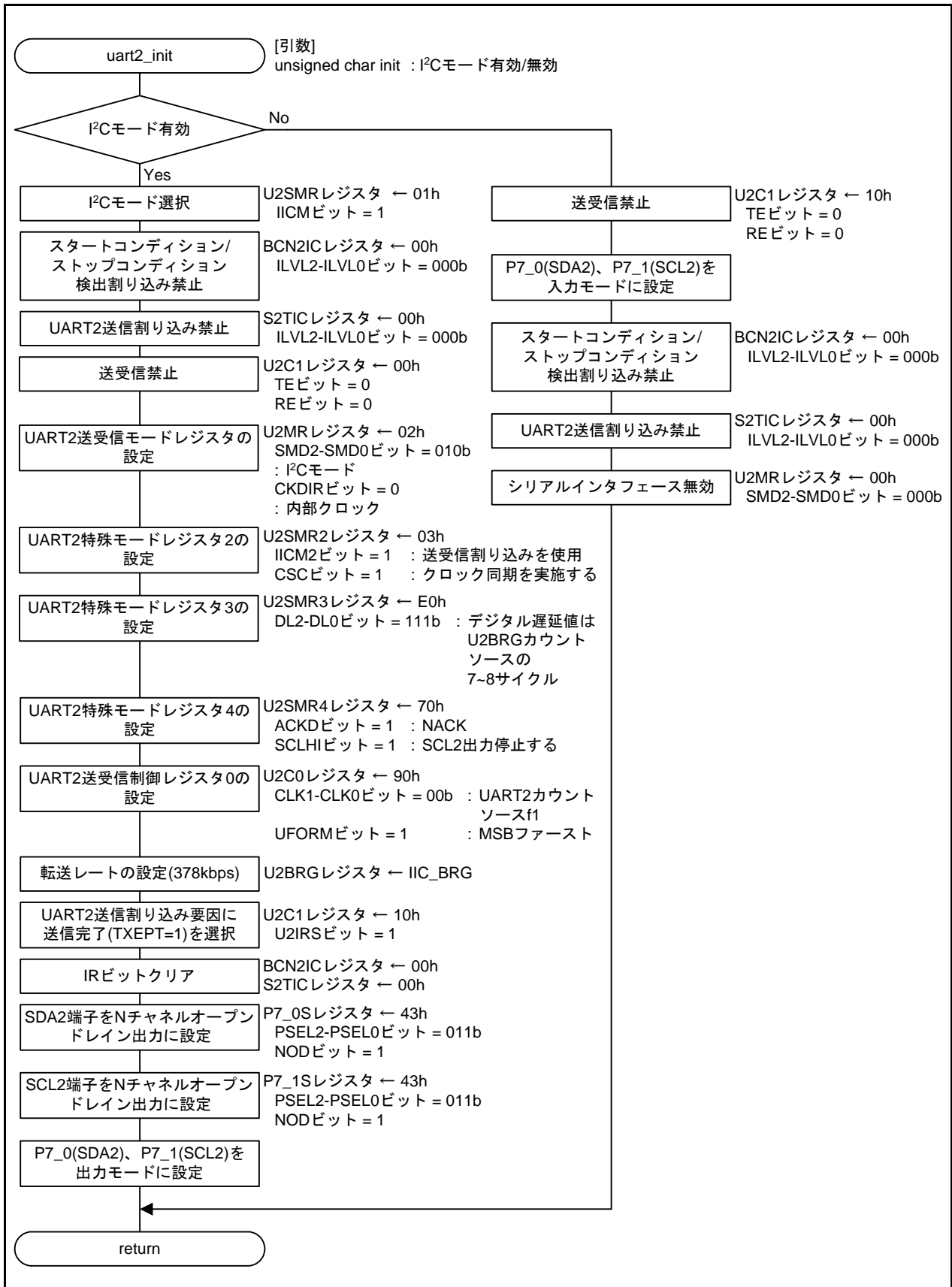


図 5.6 UART2 初期設定処理

5.7.3 マスタ制御開始処理

図 5.7にマスタ制御開始処理のフローチャートを示します。

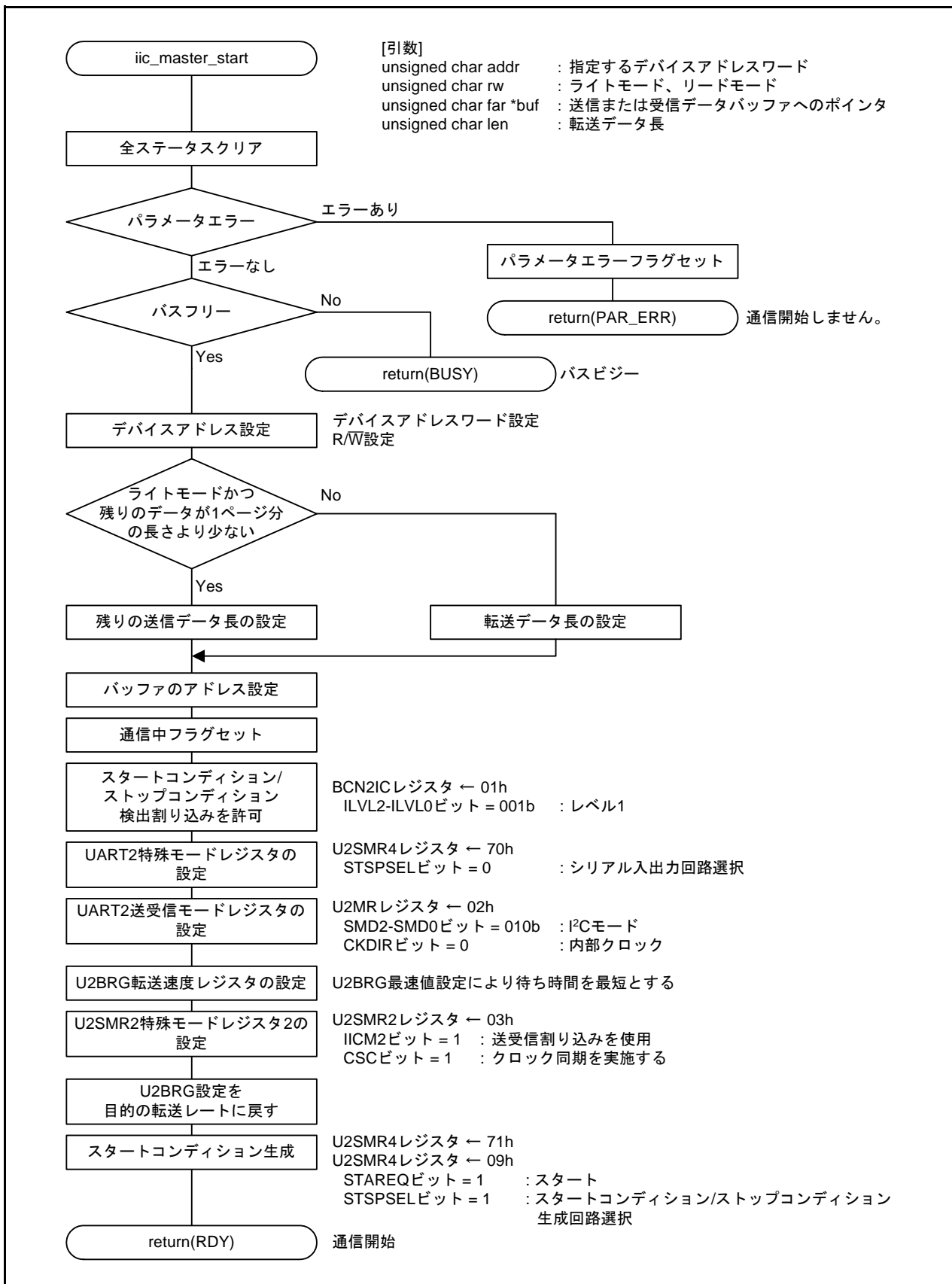


図 5.7 マスタ制御開始処理

5.7.4 スタートコンディション/ストップコンディション検出割り込み処理

図 5.8 にスタートコンディション/ストップコンディション検出割り込み処理のフローチャートを示します。

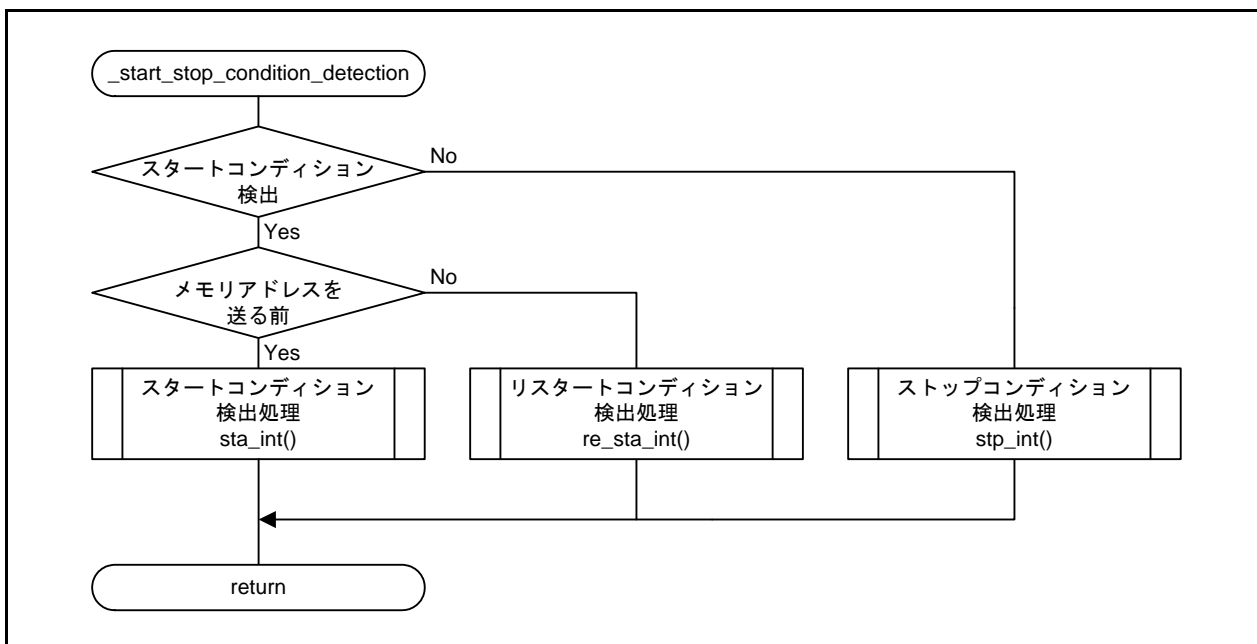


図 5.8 スタートコンディション/ストップコンディション検出割り込み処理

5.7.5 スタートコンディション検出処理

図 5.9にスタートコンディション検出処理のフローチャートを示します。

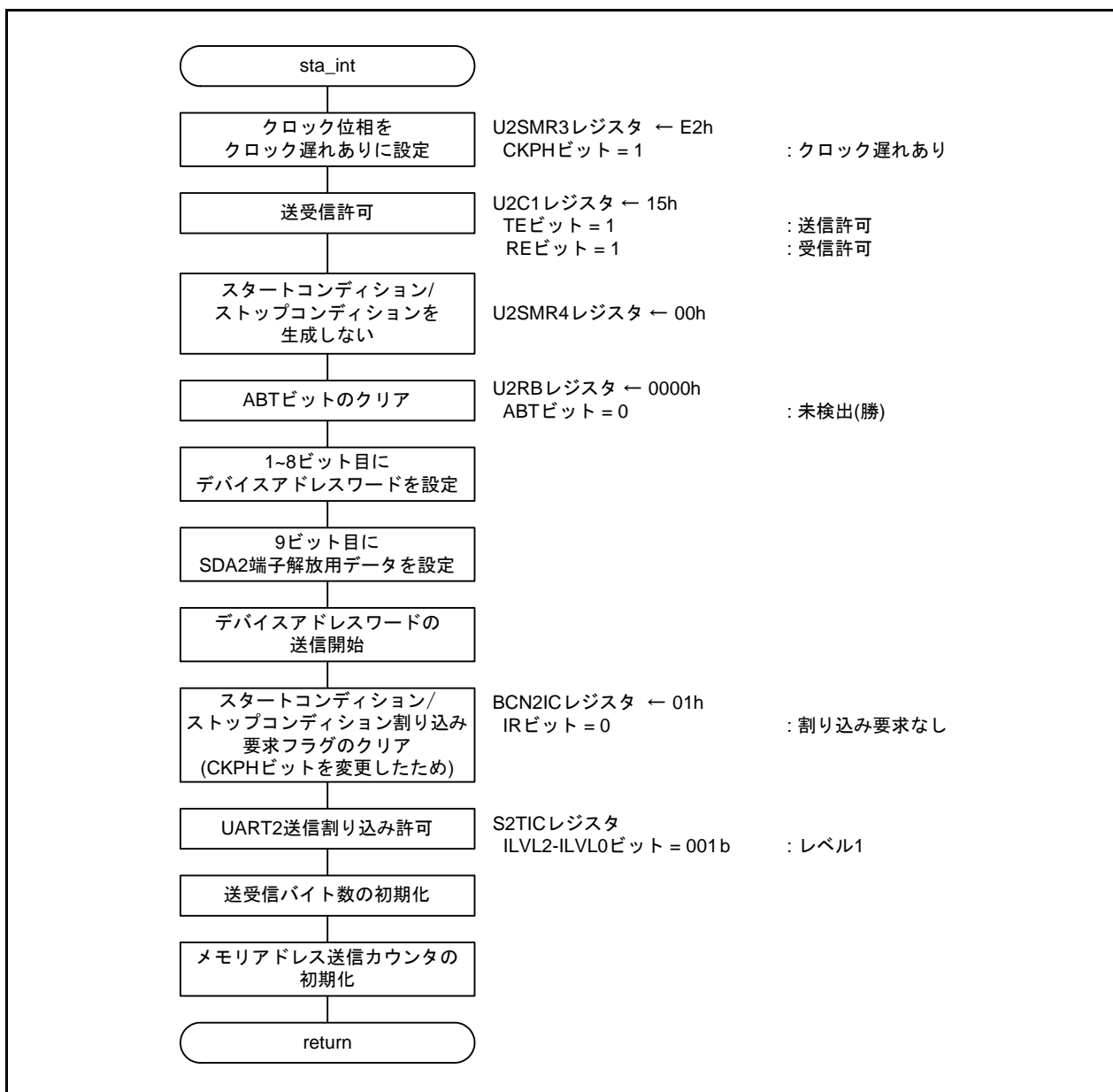


図 5.9 スタートコンディション検出処理

5.7.6 ストップコンディション検出処理

図 5.10~図 5.11 にストップコンディション検出処理のフローチャートを示します。

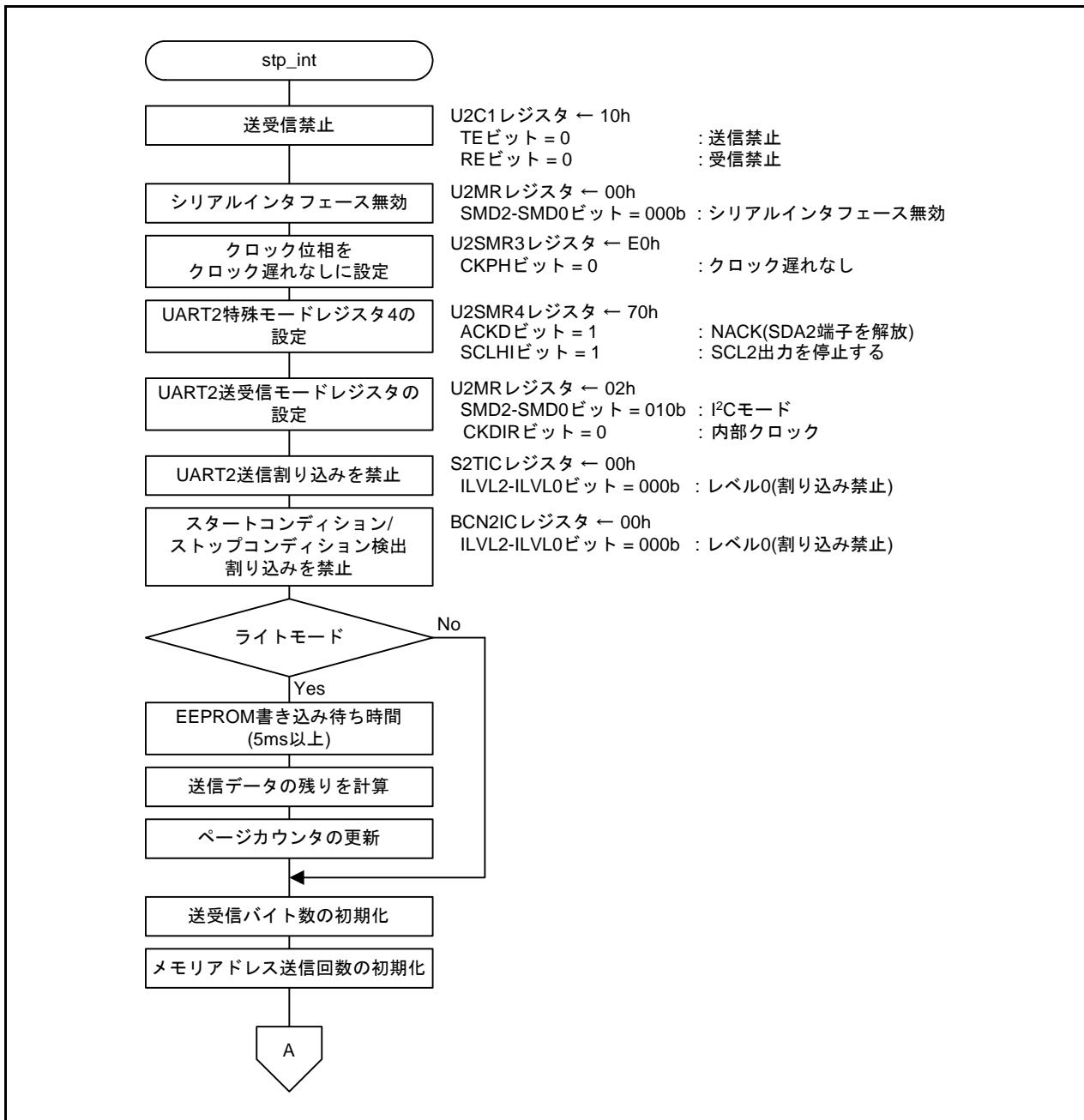


図 5.10 ストップコンディション検出処理(1/2)

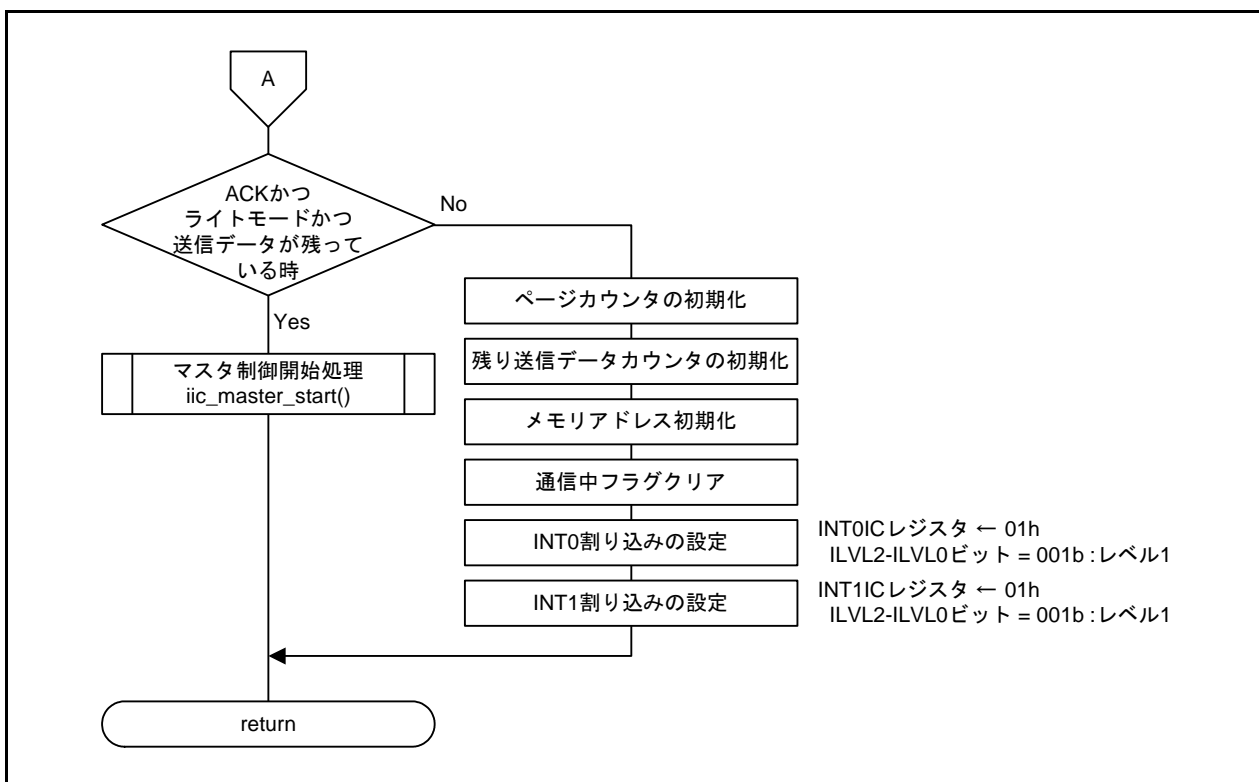


図 5.11 ストップコンディション検出処理(2/2)

5.7.7 リスタートコンディション検出処理

図 5.12にリスタートコンディション検出処理のフローチャートを示します。

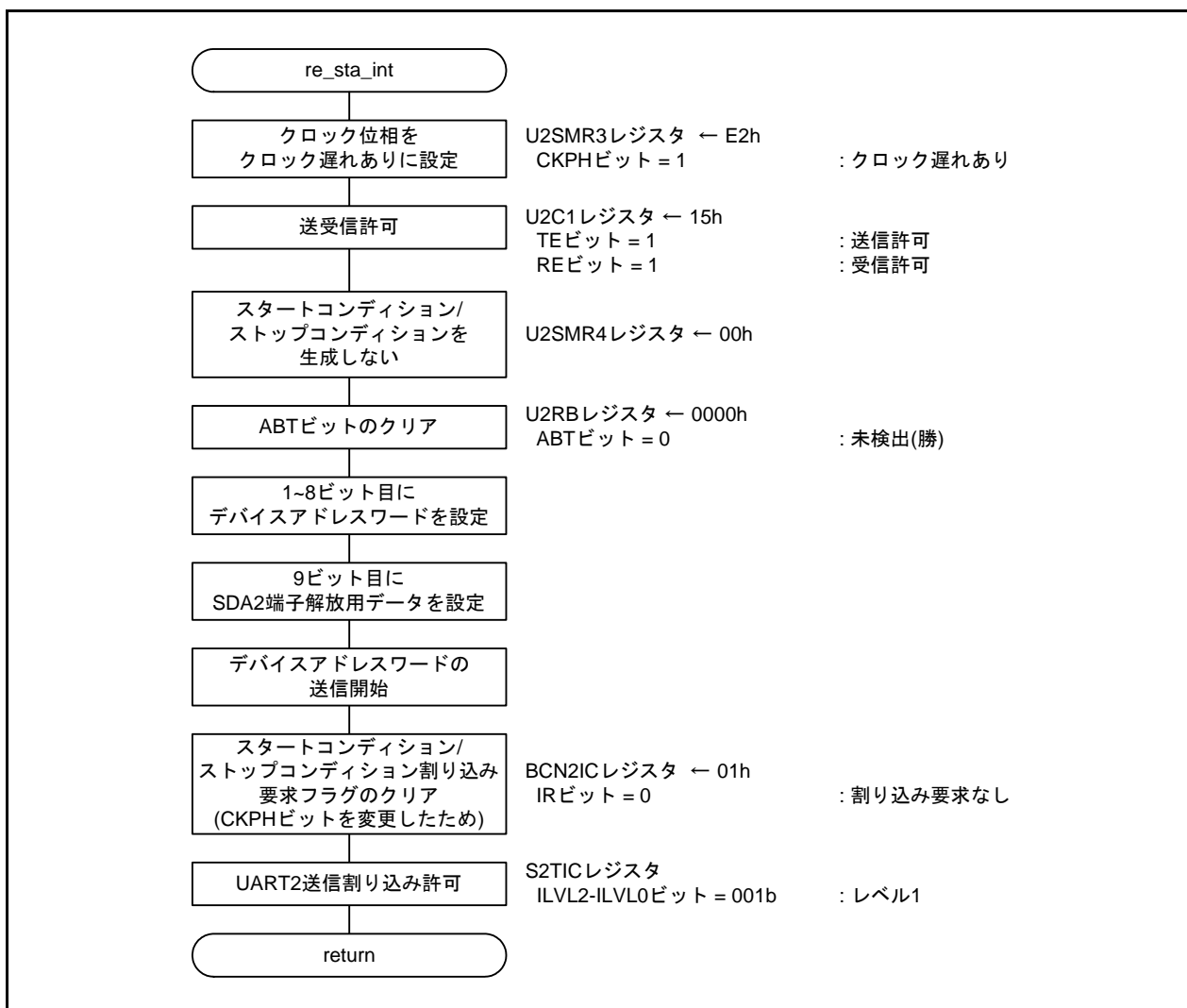


図 5.12 リスタートコンディション検出処理

5.7.8 UART2送信割り込み処理

図 5.13にUART2送信割り込み処理のフローチャートを示します。

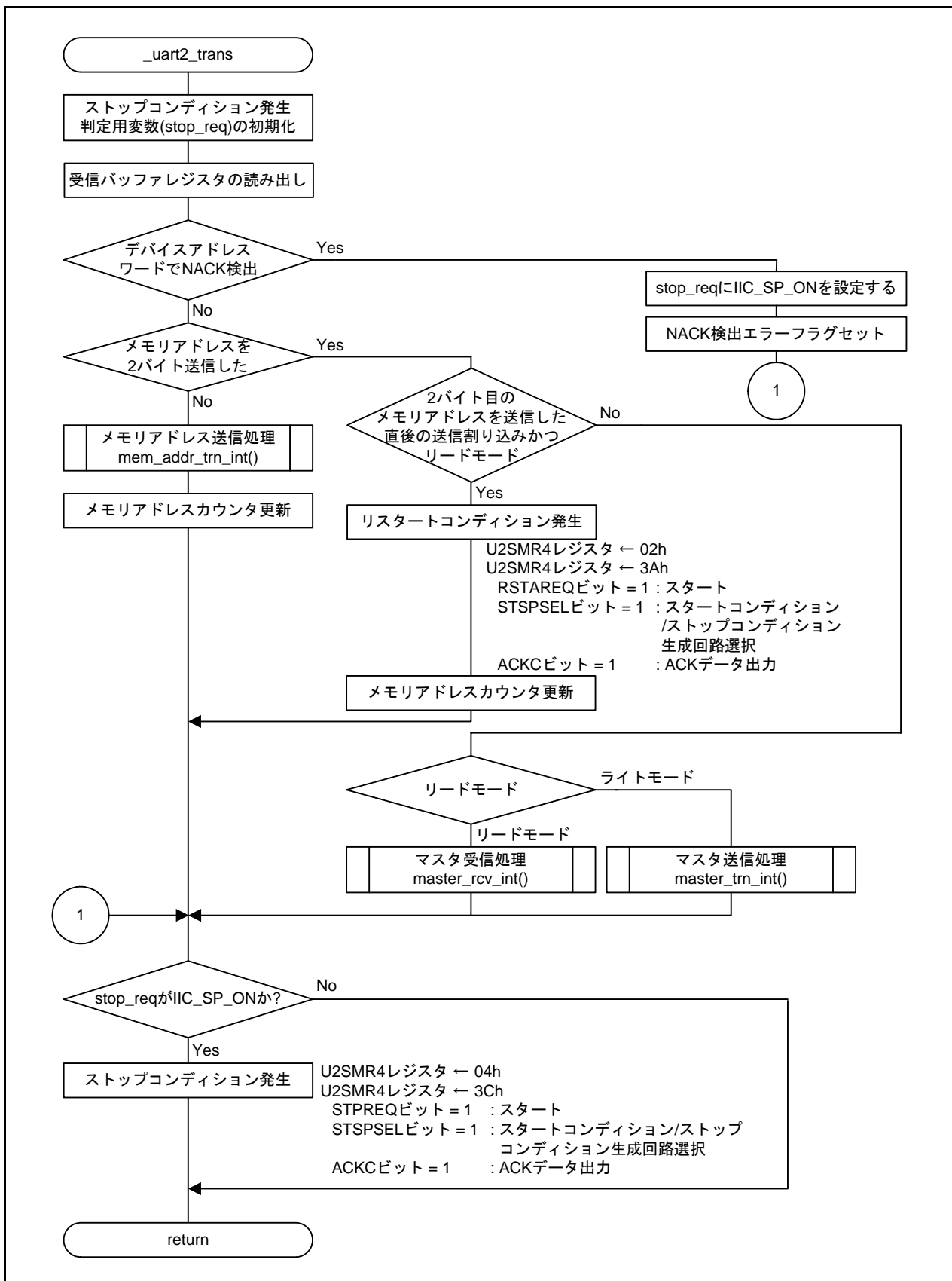


図 5.13 UART2送信割り込み処理

5.7.9 メモリアドレス送信処理

図 5.14にメモリアドレス送信処理のフローチャートを示します。

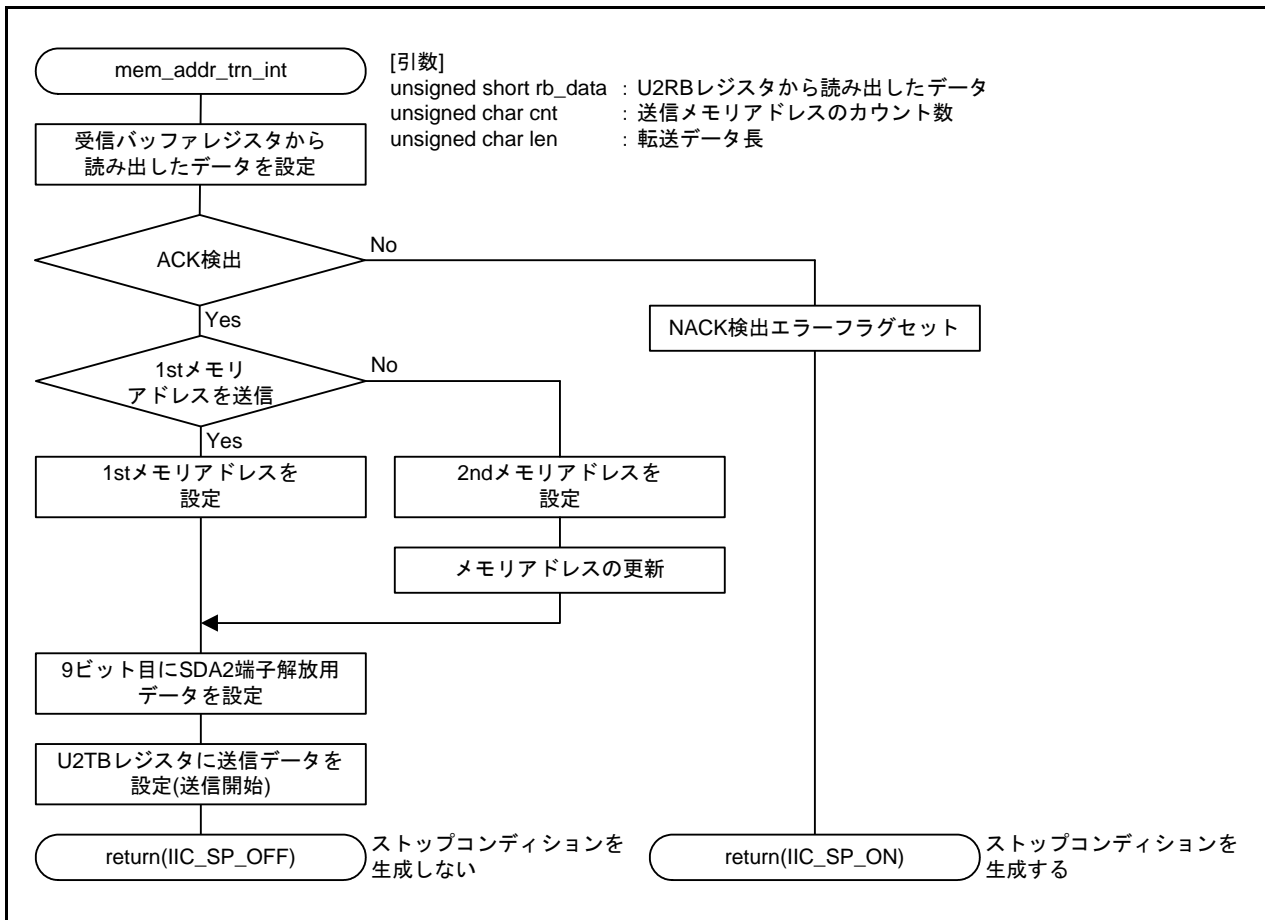


図 5.14 メモリアドレス送信処理

5.7.10 マスタ送信処理

図 5.15 にマスタ送信処理のフローチャートを示します。

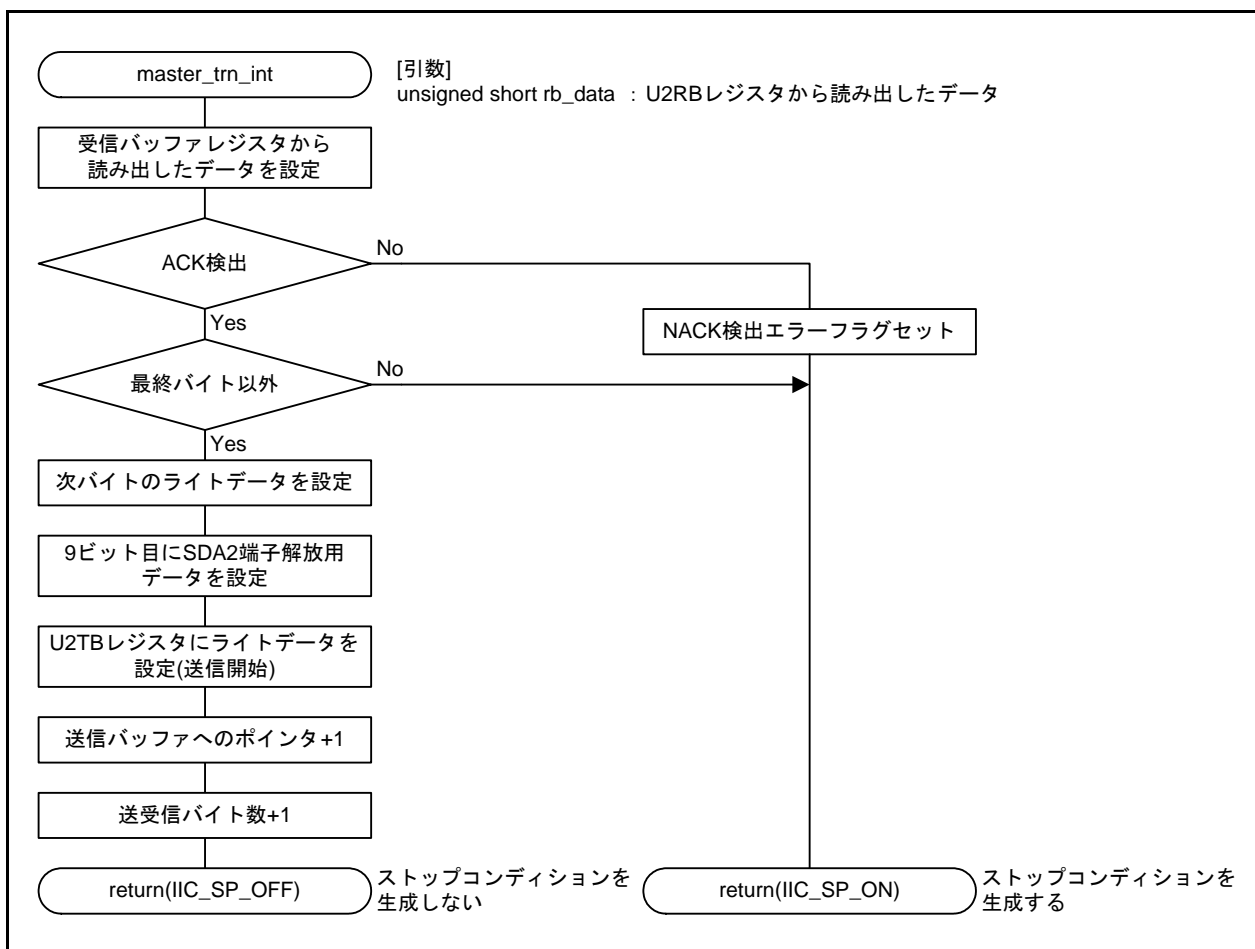


図 5.15 マスタ送信処理

5.7.11 マスタ受信処理

図 5.16にマスタ受信処理のフローチャートを示します。

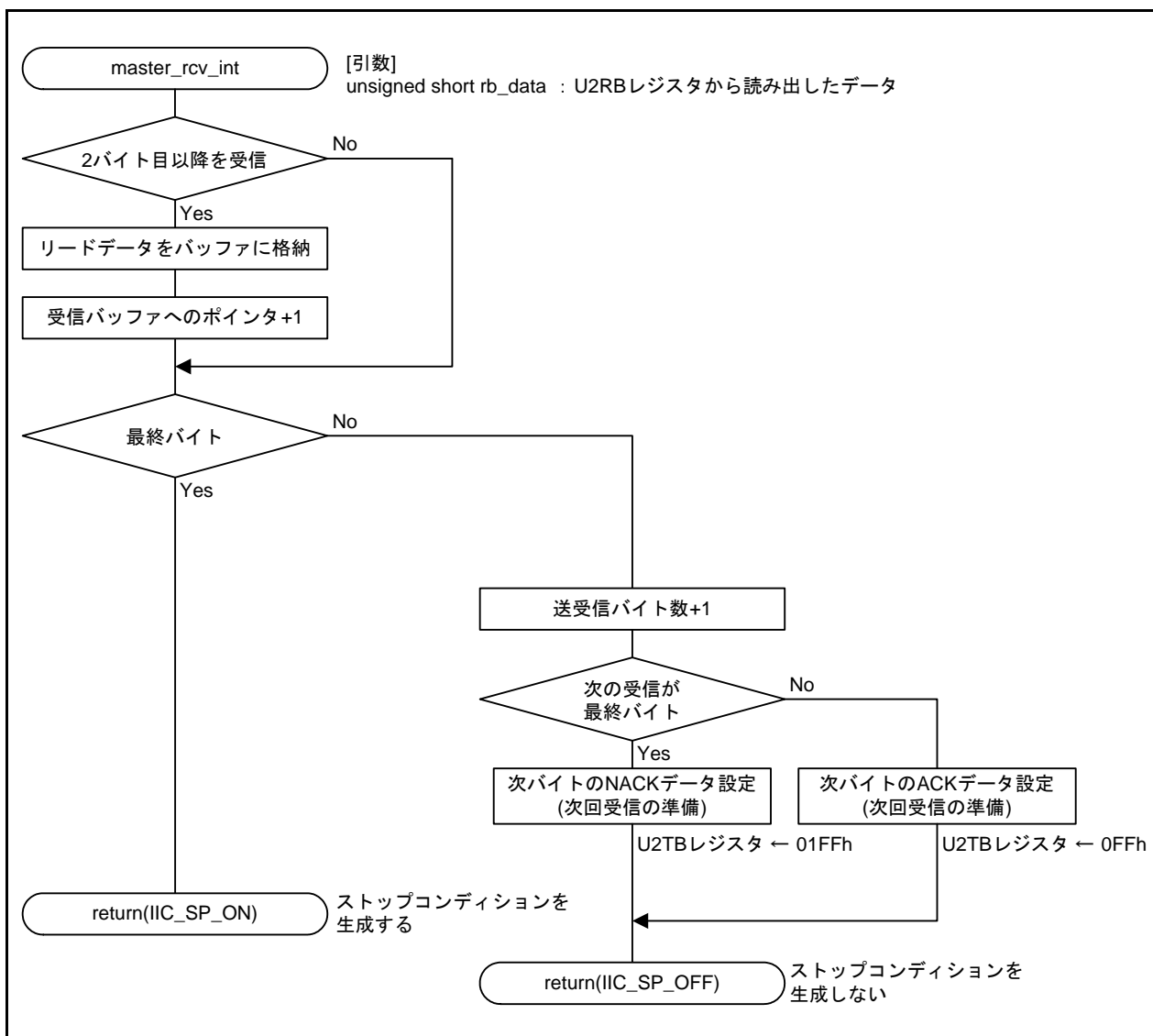


図 5.16 マスタ受信処理

5.7.12 INT0、INT1 初期設定処理

図 5.17にINT0、INT1 初期設定処理のフローチャートを示します。

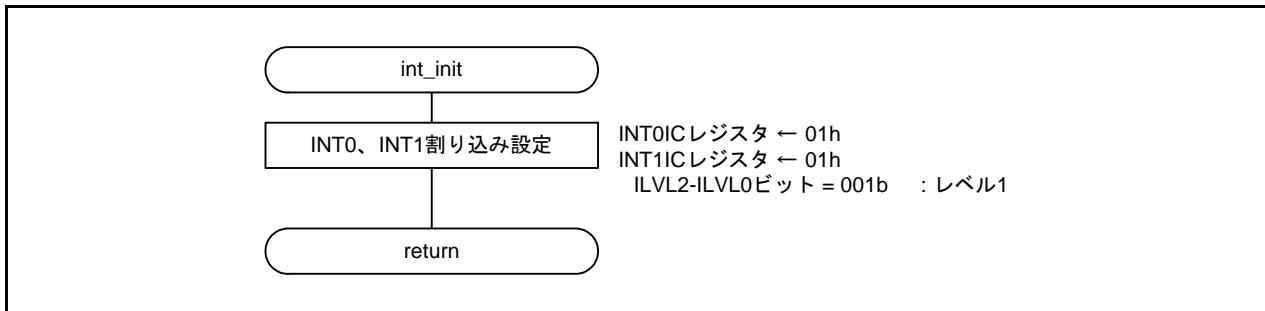


図 5.17 INT0、INT1 初期設定処理

5.7.13 INT0 割り込み処理

図 5.18にINT0 割り込み処理のフローチャートを示します。

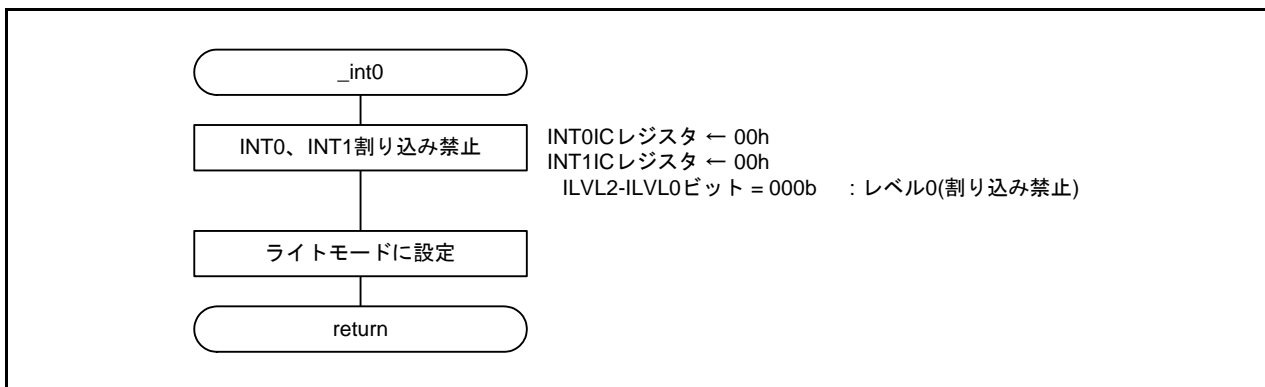


図 5.18 INT0 割り込み処理

5.7.14 INT1 割り込み処理

図 5.19にINT1 割り込み処理のフローチャートを示します。

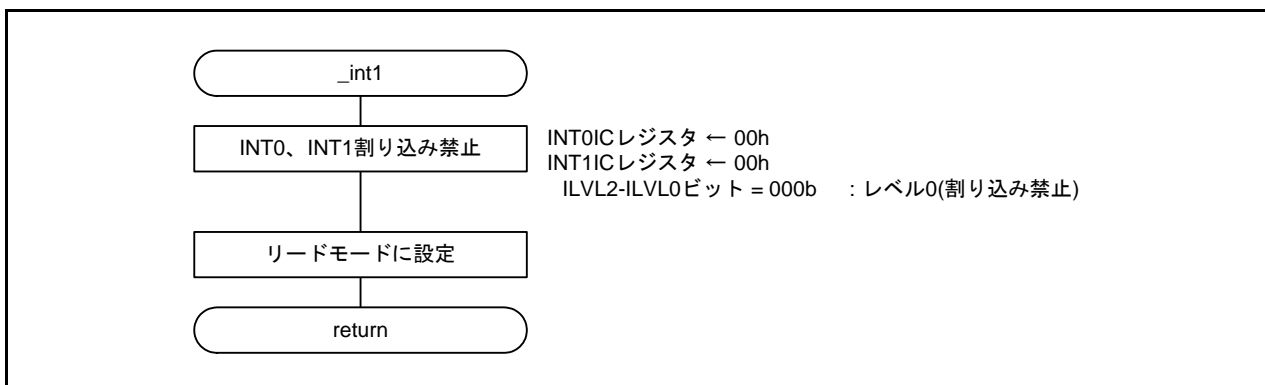


図 5.19 INT1 割り込み処理

6. 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。

7. 参考ドキュメント

R32C/116 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10

R32C/117 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10

R32C/118 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10

(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。)

C コンパイラマニュアル

R32C/100 シリーズ用 C コンパイラパッケージ V.1.02

C コンパイラユーザーズマニュアル Rev.2.00

(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>

改訂記録	R32C/100シリーズ UARTi特殊モード1(I2Cモード)を使用したEEPROM制御
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2011.08.31	-	初版発行
1.01	2011.09.30	-	全般 「スタート/ストップコンディション」 → 「スタートコンディション/ストップコンディション」
		1	要旨 「UARTi」 → 「UART2」、「他の」 → 「UART2以外の」
		7	(5) 「(5)を行います。」 → 「(5)を繰り返します。」 (8) 「EEPROMが~待ちます。」 → 「約5msの~待ちます。」 本文 「(2)~(8)を行います。」 → 「(2)~(8)を繰り返します。」
		11	表5.1 「アドレスデバイスワード」 → 「デバイスアドレスワード」、 「配列のサイズ」 → 「送受信データバッファ」
		15~18	割り込みに関する関数の宣言に、リターン値「void」を追加
		16,29	mem_addr_trn_int 第二引数 unsigned char cnt 「データカウント数」 → 「送信メモリアドレスのカウント数」
		22	5.7.3 マスタ制御開始処理 関数名の誤記修正、 判定「パラメータエラーなし」の分岐「Yes/No」の表現変更、 U2SMR2レジスタ設定のコメント変更。
		25	図5.10 U2C1レジスタREビット←0 : 「受信許可」 → 「受信禁止」
28	図5.13 判定「メモリアドレスを~かつリードモード」 → 「2バイト目のメモリアドレスを~かつリードモード」		

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。

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っていると、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。型名が異なる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情報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。



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>