

RX63N グループと RX210 グループ

RIIC モジュールを用いた I2C 通信

要旨

本サンプルコードでは、RX63N（マスタ）と RX210（スレーブ）を接続し、I2C 通信の初期設定から送受信する方法について説明します。

対象デバイス

- RX63N
- RX210

内容

1. 仕様	3
2. 動作確認条件.....	3
2.1 マスタ	3
2.2 スレーブ	4
3. ハードウェア説明.....	5
3.1 使用端子一覧.....	5
3.1.1 マスタ	5
3.1.2 スレーブ	5
3.2 接続信号.....	6
4. ソフトウェア説明.....	7
4.1 動作概要.....	7
4.2 マスタ (RX63N)	9
4.2.1 ファイル構成	9
4.2.2 定義、定数一覧.....	9
4.2.3 変数一覧.....	10
4.2.4 関数一覧.....	11
4.2.5 関数仕様.....	12
4.2.6 作成する関数のフローチャート	15
4.3 スレーブ (RX210)	29
4.3.1 ファイル構成	29
4.3.2 定義一覧.....	29
4.3.3 変数一覧.....	30
4.3.4 関数一覧.....	31
4.3.5 関数仕様.....	31
4.3.6 作成する関数のフローチャート	34
5. 動作確認方法.....	45
6. 参考ドキュメント.....	47

1. 仕様

RX63N（マスタ）は I2C 通信でデータを RX210（スレーブ）に送信します。RX210（スレーブ）は受信したデータを RX63N（マスタ）に送信します。RX63N（マスタ）はスレーブへ送信したデータと、スレーブから受信したデータが一致した場合、LED1 が点灯します。

2. 動作確認条件

2.1 マスタ

本サンプルコードは、表 2.1 の条件で動作を確認しています。

表 2.1 マスタ動作確認条件

項目	内容
使用マイコン	R5F563NFD DFP (RX63N グループ)
動作周波数	・メインクロック：12MHz ・システムクロック (ICLK)：12MHz（メインクロック 1 分周） ・周辺モジュールクロック B(PCLKB)：12MHz（メインクロック 1 分周）
ボード電源電圧	5V
マイコン動作電圧	3.3V
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
統合開発環境	ルネサスエレクトロニクス製品 e2 studio V6.1.0
エミュレータ	ルネサスエレクトロニクス製 E1 エミュレータ
使用ボード	北斗電子製評価ボード HSBRX63NP(R5F563NFD DFP)

2.2 スレープ

本サンプルコードは、表 2.2 の条件で動作を確認しています。

表 2.2 スレープ動作確認条件

項目	内容
使用マイコン	R5F5210BBDFP (RX210 グループ)
動作周波数	・メインクロック : 20MHz ・システムクロック (ICLK) : 20MHz (メインクロック 1 分周) ・周辺モジュールクロック B(PCLKB) : 20MHz (メインクロック 1 分周)
ボード電源電圧	3.3V
マイコン動作電圧	3.3V
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
統合開発環境	ルネサスエレクトロニクス製品 e2 studio V6.1.0
エミュレータ	ルネサスエレクトロニクス製 E1 エミュレータ
使用ボード	北斗電子製評価ボード HSBRX210-100B(R5F5210BBDFP)

3. ハードウェア説明

3.1 使用端子一覧

3.1.1 マスタ

表 3.1 に RX63N の使用端子と機能を示します。

表 3.1 RX63N の使用端子と機能

端子名	入出力	内容
P12	出力	I2C クロック出力端子(SCL0)
P13	入出力	I2C データ入出力端子(SDA0)
P21	入力	汎用ポート入力(P21)

3.1.2 スレーブ

表 3.2 に RX210 の使用端子と機能を示します。

表 3.2 RX210 の使用端子と機能

端子名	入出力	内容
P12	入力	I2C クロック入力端子(SCL0)
P13	入出力	I2C データ入出力端子(SDA0)
P21	出力	汎用ポート出力(P21)

3.2 接続信号

本サンプルコードで使用する接続信号一覧を表 3.3 に示します。また、RX63N ボード、RX210 ボードの接続図を図 3.1 に示します。

表 3.3 接続信号一覧

	RX63N	RX210
接続 1	J2 の 3pin (P12/SCL0)	J1 の 27pin (P12/SCL0)
接続 2	J2 の 4pin (P13/SDA0)	J1 の 28pin (P13/SDA0)
接続 3	J2 の 10pin (P21)	J1 の 34pin (P21)
接続 4	J1 の 49pin (VCC)	J1 の 47pin (VCC)
接続 5	J1 の 50pin (VSS)	J1 の 50pin (VSS)

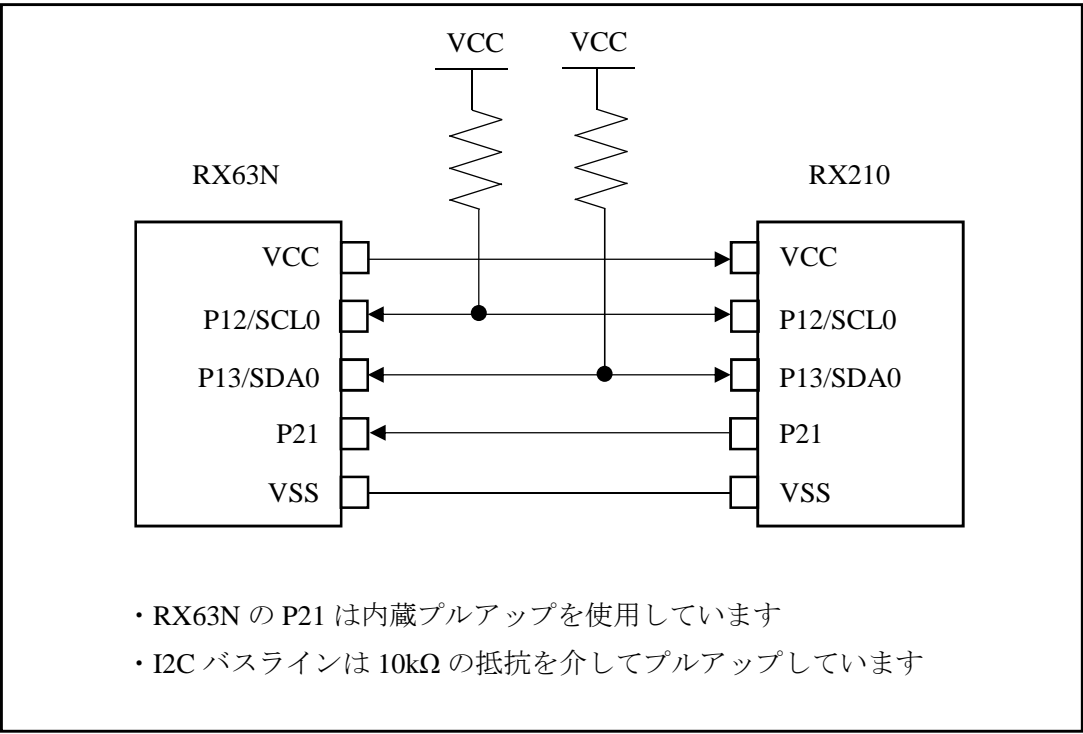


図 3.1 接続図

4. ソフトウェア説明

4.1 動作概要

図 4.1 に RX63N - RIIC0 → RX210 - RIIC0 および RX210 - RIIC0 → RX63N - RIIC0 の動作タイミング図を、以下に図中の番号の動作および処理を示します。使用関数も合わせて参照ください。

(1) 初期設定

RX63N - RIIC0 から RX210 - RIIC0 ヘデータを送信するために、以下の初期設定を行います。

●RX63N – RIIC0

- ・ I2C マスタモード
- ・ 転送速度設定は 100kbps
- ・ ノイズフィルタ設定未使用
- ・ SAD 出力遅延設定未使用

●RX210 – RIIC0

- ・ I2C スレーブモード
- ・ スレーブアドレスは 0x1A
- ・ 転送速度設定は 100kbps
- ・ ノイズフィルタ設定未使用
- ・ SAD 出力遅延設定未使用

(2) RX210 – RIIC0 (スレーブ)の受信開始

受信の開始設定をします。受信設定後、P21 を High → Low → High 出力とし、マスタ側に準備ができたことを通知します。

(3) RX63N – RIIC0 (マスタ)の送信開始

P21 の Low レベルを確認後、送信の開始設定をします。スタートコンディションを発行し、スレーブアドレス(0x1A)に対して、SCL 端子のクロックを出力し、クロックに同期させて送信データ (文字列 "TEST") を送信します。

(4) RX210 – RIIC0 (スレーブ) 受信完了および RX63N – RIIC0 (マスタ) 送信完了

送信データが送信完了すると、転送完了割り込みが発生します。

ストップコンディションが発行され、SCL、SDA 信号が“H”となります。

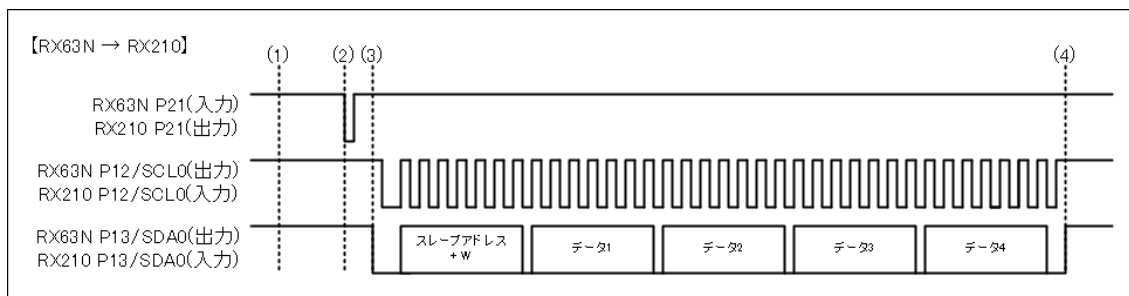


図 4.1 RX63N - RIIC0 → RX210 - RIIC0 の動作タイミング

図 4.2 に RX210 - RIIC0 → RX63N - RIIC0 の動作タイミング図を、以下の図中の番号の動作および処理を示します。使用関数も合わせて参照ください。

(1) RX210 - RIIC0 (スレーブ)の送信開始

送信の開始設定をします。このとき、送信データは RX63N - RIIC0 → RX210 - RIIC0 で受信したデータ (文字列 "TEST") を送信します。設定後、P21 を High → Low 出力とし、マスタ側に準備ができたことを通知します。

(2) RX63N - RIIC0 (マスタ)の受信開始

RX210 - P21 からの信号受信後、受信の開始設定をします。スタートコンディションを発行し、スレーブアドレス(0x1A)に対して、SCL 端子のクロックを出力し、クロックに同期させてデータを受信します。

(3) RX210 - RIIC0 (スレーブ) 送信完了および RX63N - RIIC0 (マスタ) 受信完了

送信データが送信完了すると、転送完了割り込みが発生します。
ストップコンディションが発行され、SCL、SDA 信号が“H”となります。

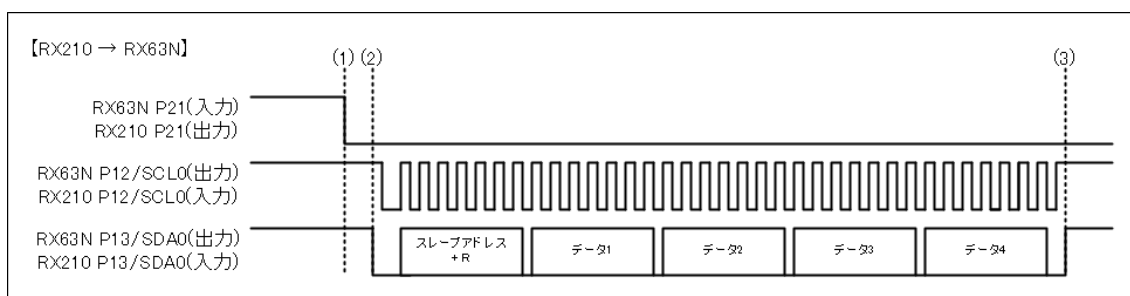


図 4.2 RX210 - RIIC0 → RX63N - RIIC0 の動作タイミング

4.2 マスタ (RX63N)

4.2.1 ファイル構成

本サンプルコードを作成するにあたり、編集したファイルを表 4.1 に示します。

表 4.1 ファイル名一覧

ファイル名	概要	備考
RIIC_RX63N_MASTER.c	メインファイル	

4.2.2 定義、定数一覧

本サンプルコードで使用する定義一覧を表 4.2 に、定数一覧を表 4.3 に示します。

表 4.2 定義一覧

定数名	設定値	内容
MASTER_SENDS_ADR_7_W	0x00U	7bitスレーブアドレス送信
MASTER_SENDS_ADR_7_R	0x01U	7bit スレーブアドレス受信
MASTER_SENDS_DATA	0x05U	マスタデータ送信中
MASTER_SENDS_END	0x06U	マスタ送信完了
MASTER_SENDS_STOP	0x07U	マスタ送信停止
MASTER_RECEIVES_START	0x08U	マスタ受信開始
MASTER_RECEIVES_DATA	0x09U	マスタデータ受信中
MASTER_RECEIVES_STOPPING	0x0AU	マスタ受信停止中
MASTER_RECEIVES_STOP	0x0BU	マスタ受信停止
MASTER_RECEIVE	0x0CU	マスタ受信モード
MASTER_TRANSMIT	0x0DU	マスタ送信モード
MASTER_RECEIVES_RESTART	0x0EU	マスタ受信再開
SLAVE_ADDRES	0x1AU	スレーブのアドレス
BUFFER_SIZE	0x04U	バッファサイズ

表 4.3 定数一覧

型	変数名	設定値	内容
const unsigned char	riic0_tx_buffer[BUFFER_SIZE]	"TEST"	送信バッファ

4.2.3 変数一覧

表 4.4 に本サンプルコードで使用する変数一覧を示します。

表 4.4 変数一覧

型	変数名	内容
unsigned short	riic0_mode_flag	RIIC0 動作モード
unsigned short	riic0_state	RIIC0 状態データ
unsigned char	riic0_slave_address	スレーブアドレス
unsigned char*	riic0_rx_address	受信アドレス
unsigned char*	riic0_tx_address	送信アドレス
unsigned short	riic0_rx_count	受信データカウンタ
unsigned short	riic0_tx_count	送信データカウンタ
unsigned short	riic0_rx_length	受信データ数
unsigned char	riic0_rx_buffer[BUFFER_SIZE]	受信バッファ

4.2.4 関数一覧

表 4.5 に関数一覧を掲載します。本サンプルコードで新規作成、編集した関数のみ記載しています。

表 4.5 関数一覧

関数名	概要
main	メイン処理
cgc_init	CGC の初期設定処理
port_init	ポートの初期設定処理
riic0_master_init	RIIC0 の初期設定処理
riic0_master_send	送信処理
riic0_master_receive	受信処理
r_riic0_transmit_interrupt	送信割り込み
r_riic0_transmitend_interrupt	送信完了割り込み
r_riic0_receive_interrupt	受信割り込み
r_riic0_error_interrupt	エラー、イベント発生割り込み

4.2.5 関数仕様

main

概要	メイン処理
ヘッダ	なし
宣言	void main(void)
説明	CGC、ポート、RIIC0 初期設定関数の呼び出し
引数	なし
リターン値	なし

cgc_init

概要	CGC の初期設定処理
ヘッダ	なし
宣言	void cgc_init(void)
説明	CGC の初期設定処理
引数	なし
リターン値	なし

port_init

概要	ポートの初期設定処理
ヘッダ	なし
宣言	void port_init(void)
説明	ポートの初期設定処理
引数	なし
リターン値	なし

riic0_master_init

概要	RIIC0 の初期設定処理
ヘッダ	なし
宣言	void riic0_master_init(void)
説明	RIIC0 の初期設定処理
引数	なし
リターン値	なし

riic0_master_send

概要	送信処理
ヘッダ	なし
宣言	void riic0_master_send(unsigned short adr, unsigned char * const tx_buf, unsigned short tx_num)
説明	引数で指定されたスレーブアドレスに送信バッファから送信データ数だけ送信する設定をします
引数	adr スレーブアドレス tx_buf 送信バッファ tx_num 送信データ数
リターン値	なし

riic0_master_receive

概要	受信処理
ヘッダ	なし
宣言	void riic0_master_receive(unsigned short adr, unsigned char * const rx_buf, unsigned short rx_num)
説明	引数で指定されたスレーブアドレスから受信バッファに受信データ数だけ受信する設定をします
引数	adr スレーブアドレス const rx_buf 受信バッファ rx_num 受信データ数
リターン値	なし

r_riic0_transmit_interrupt

概要	送信割り込み
ヘッダ	なし
宣言	static void r_riic0_transmit_interrupt(void)
説明	マスタ送信時のスレーブアドレス、データの送信 マスタ受信時のスレーブアドレスの送信
引数	なし
リターン値	なし

r_riic0_transmitend_interrupt

概要	送信完了割り込み
ヘッダ	なし
宣言	static void r_riic0_transmitend_interrupt(void)
説明	送信終了設定
引数	なし
リターン値	なし

r_riic0_receive_interrupt

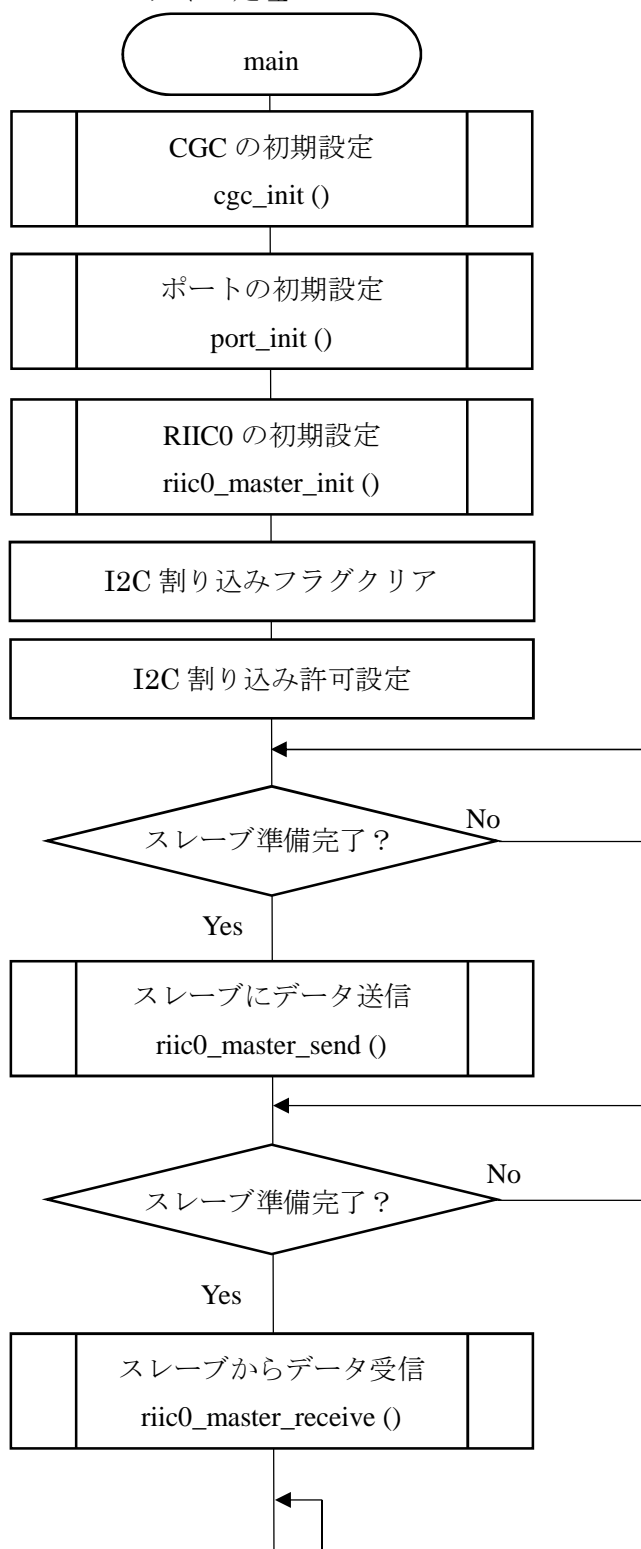
概要	受信割り込み
ヘッダ	なし
宣言	static void r_riic0_receive_interrupt(void)
説明	マスタ受信開始の設定、受信データのリード、マスタ受信終了設定
引数	なし
リターン値	なし

r_riic0_error_interrupt

概要	通信エラー、イベント発生割り込み
ヘッダ	なし
宣言	static void r_riic0_error_interrupt(void)
説明	通信エラー、イベント発生割り込み処理
引数	なし
リターン値	なし

4.2.6 作成する関数のフローチャート

4.2.6.1 メイン処理



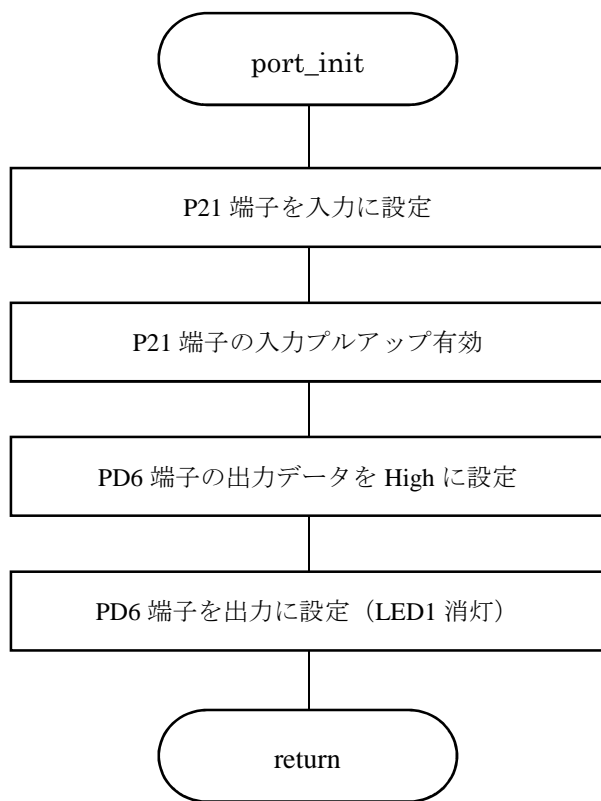
4.2.6.2

CGC の初期設定



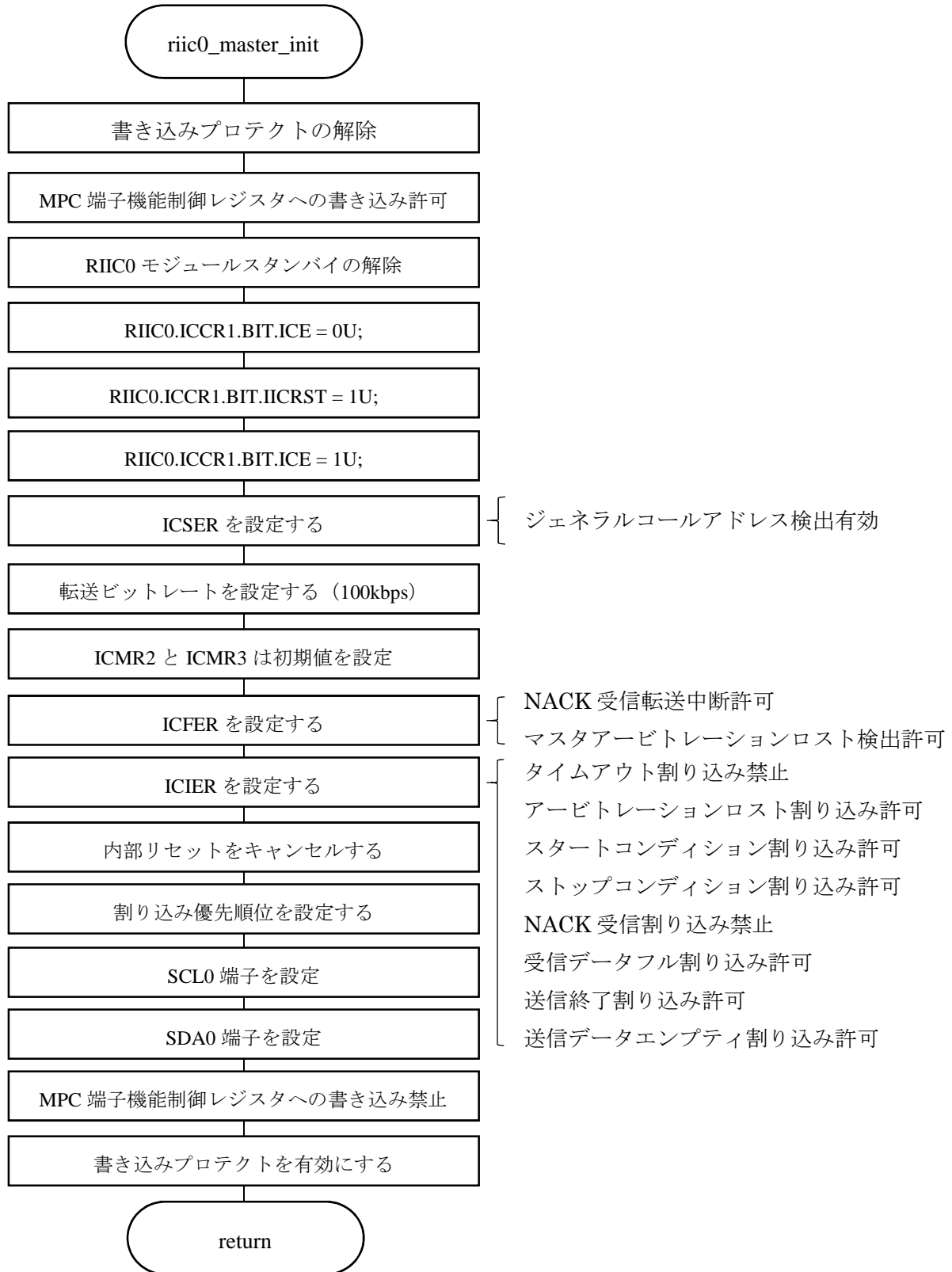
4.2.6.3

ポートの初期設定処理



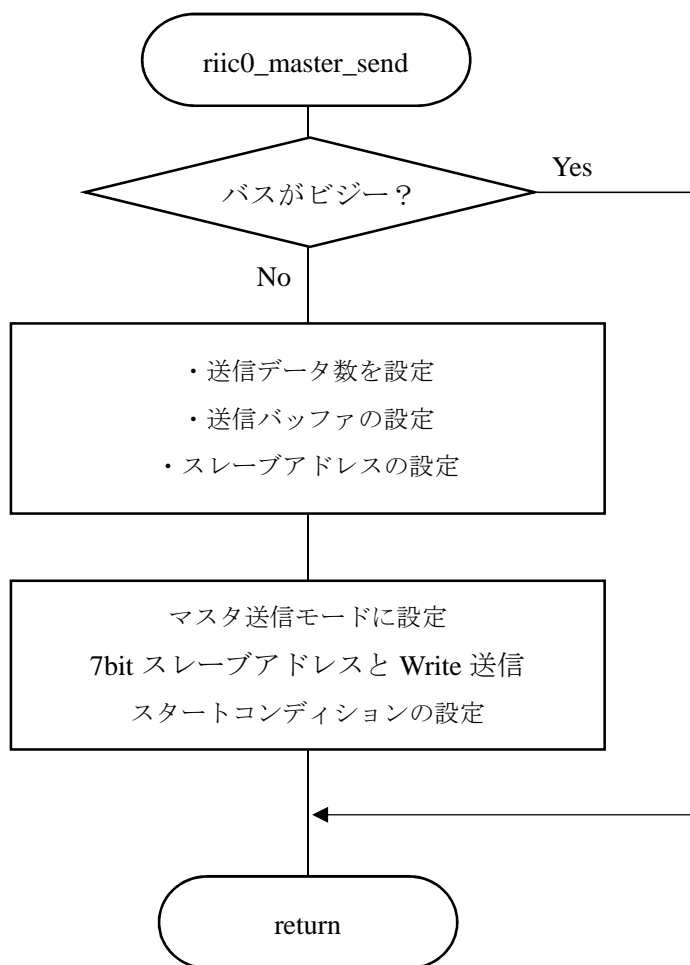
4.2.6.4

RIIC0 の初期設定処理



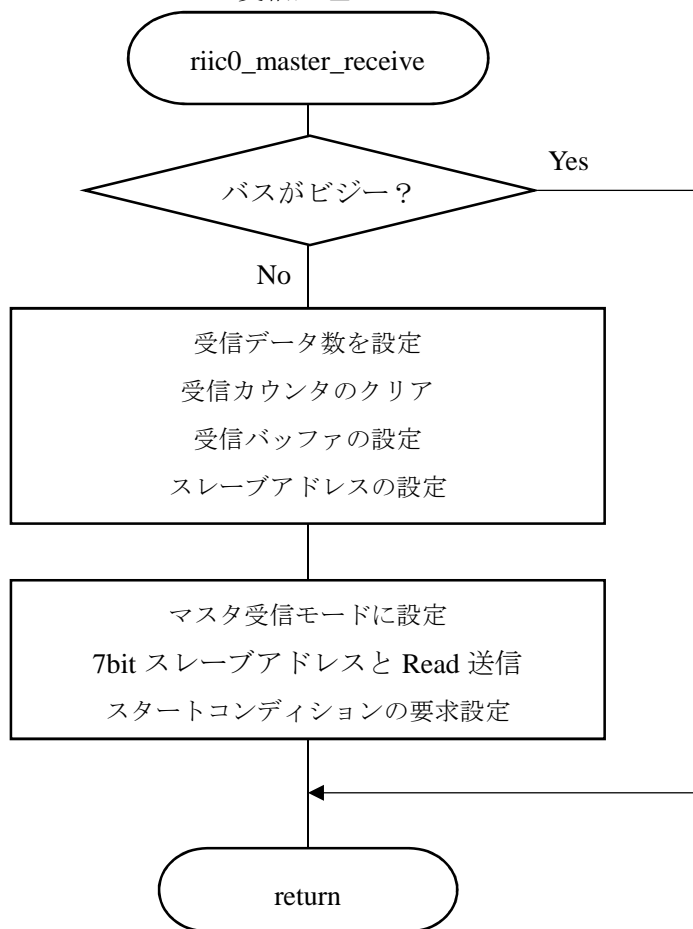
4.2.6.5

送信処理



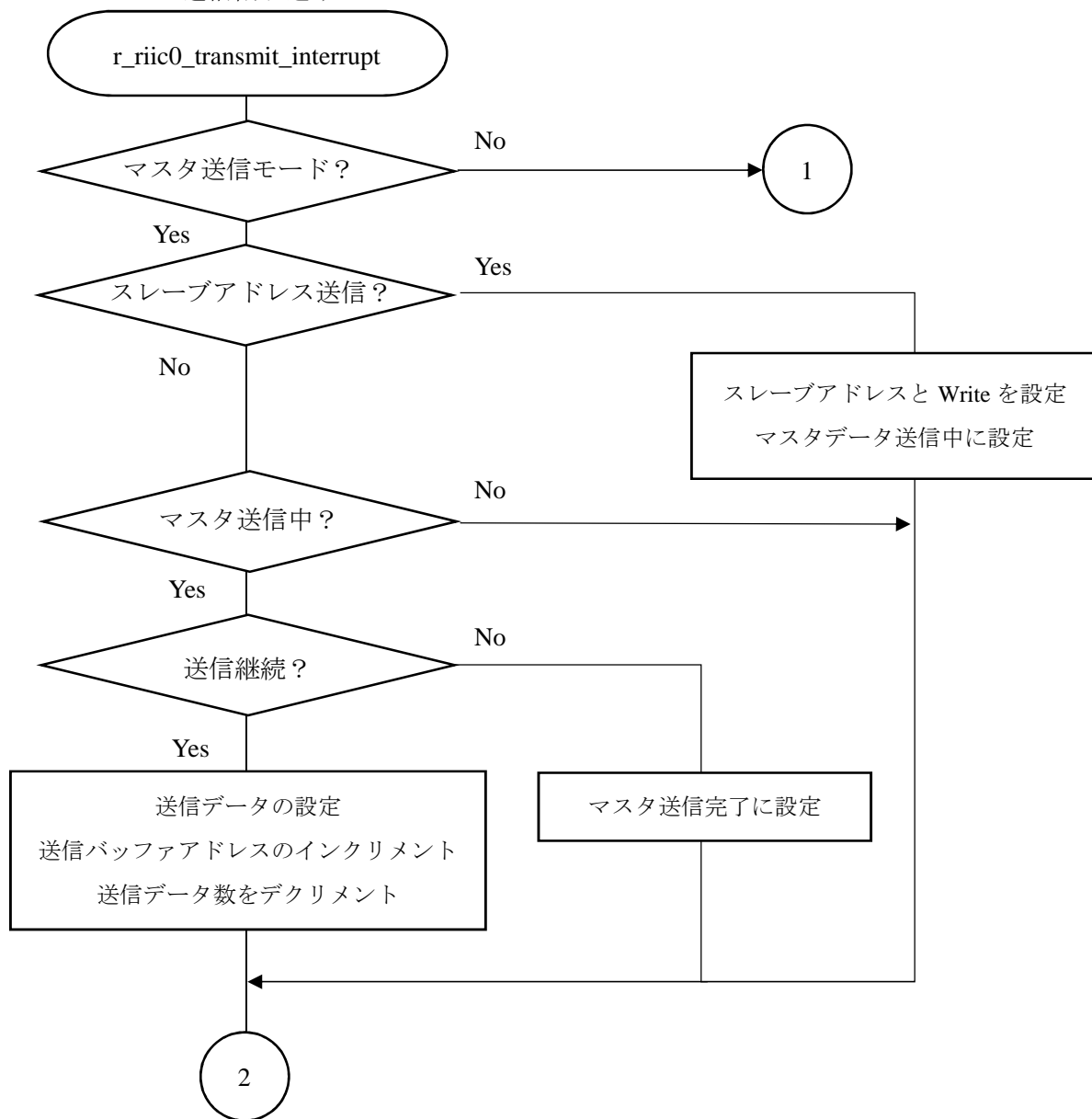
4.2.6.6

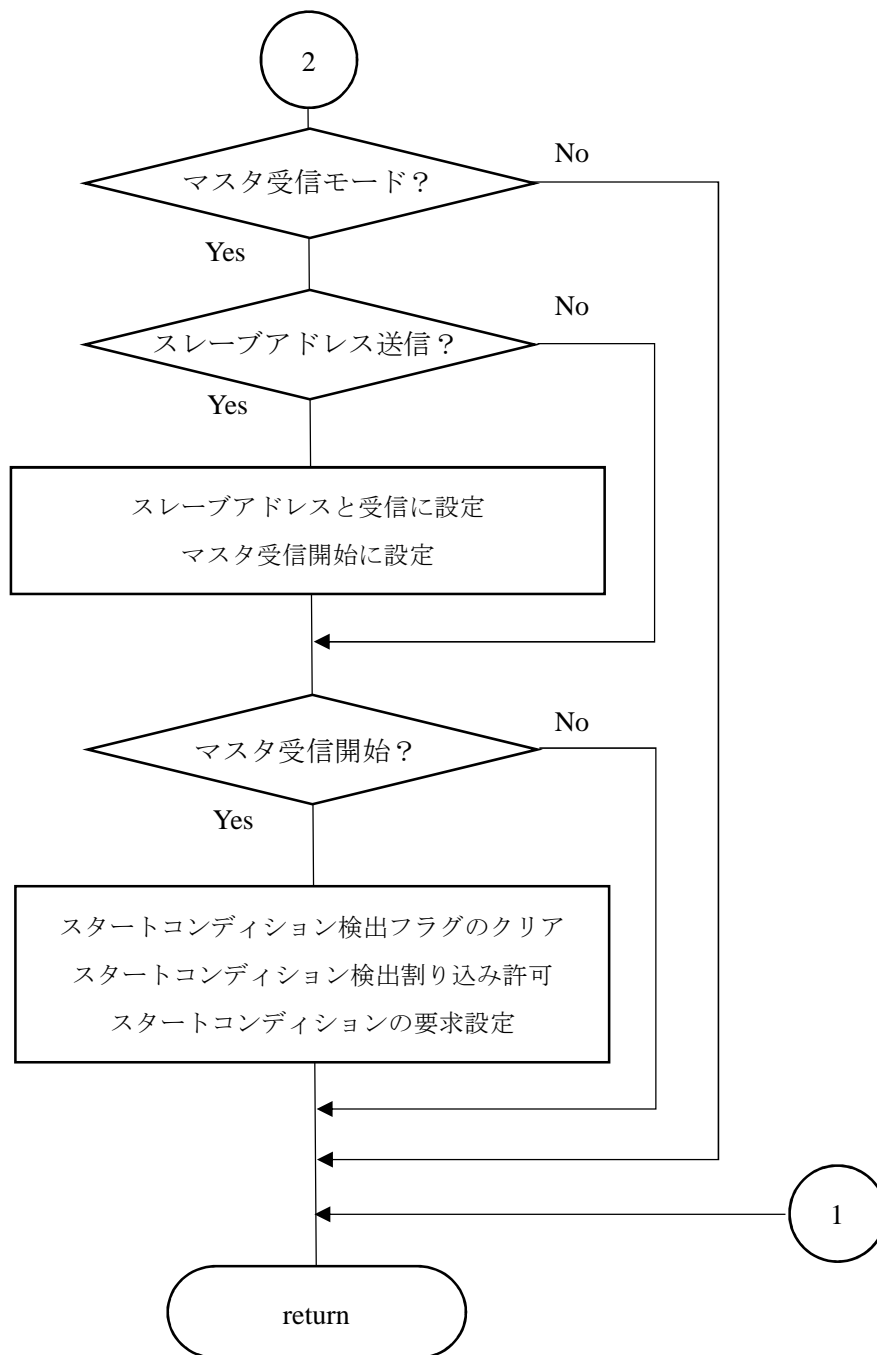
受信処理



4.2.6.7

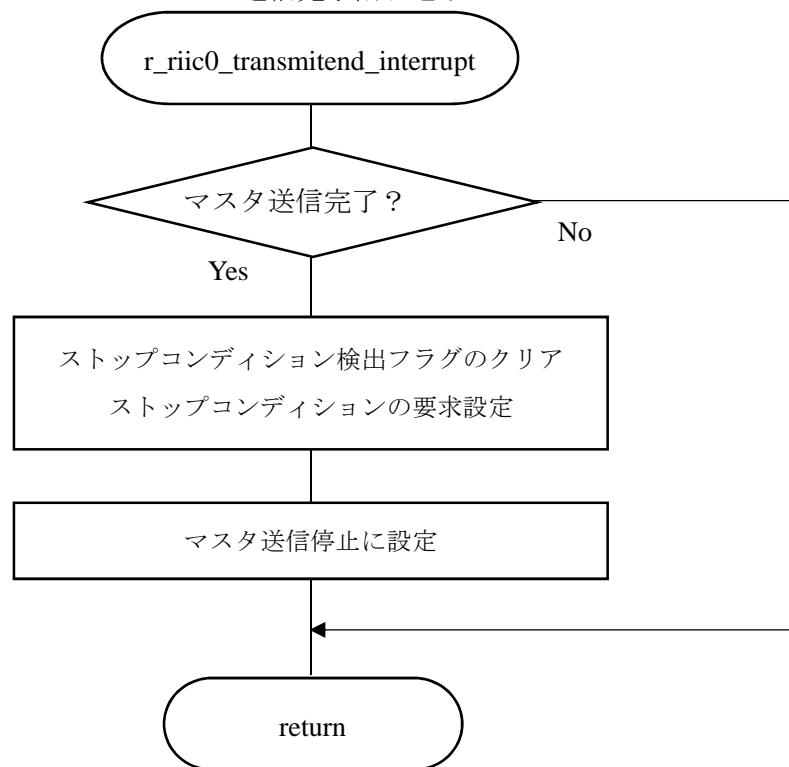
送信割り込み





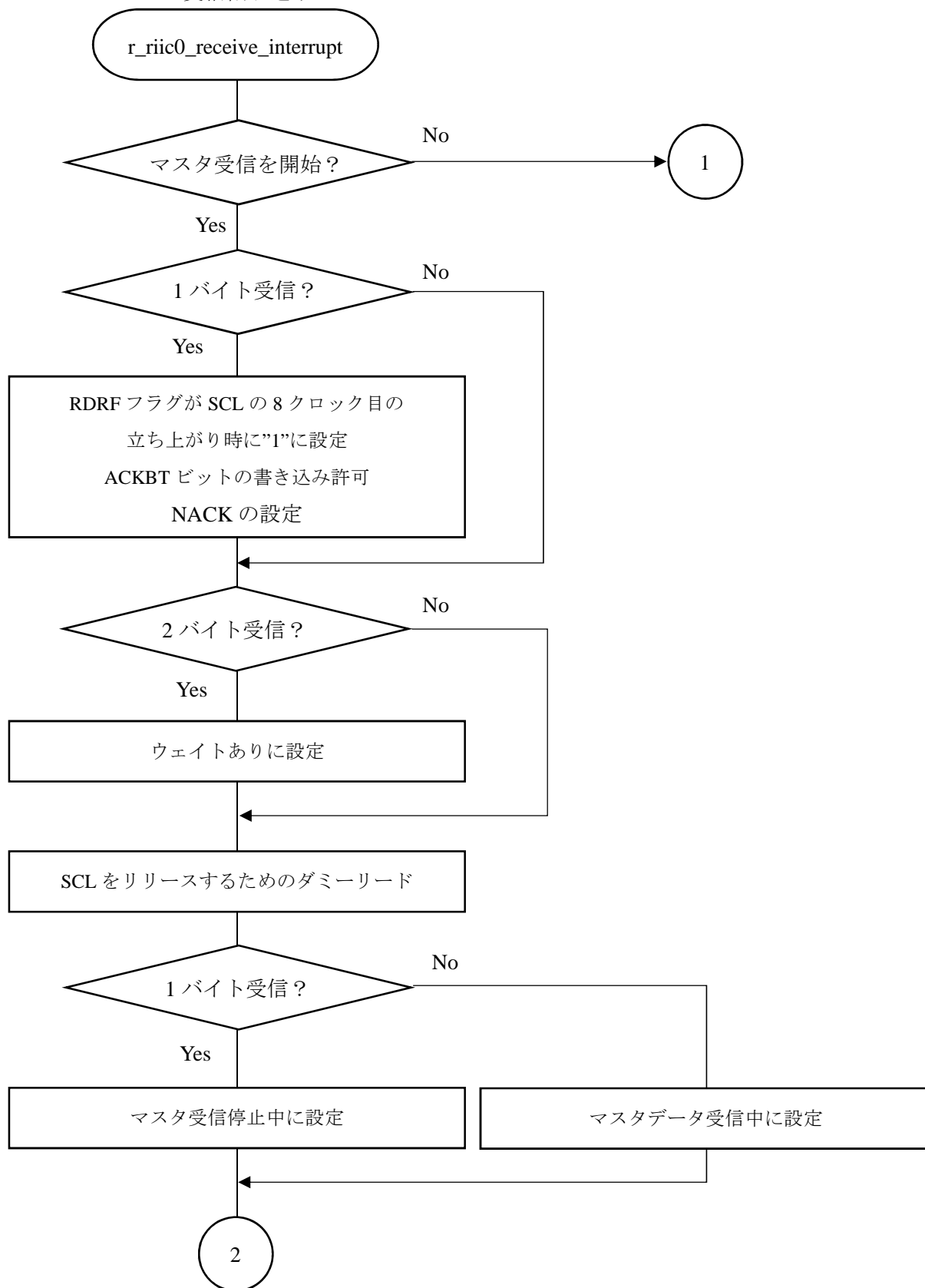
4.2.6.8

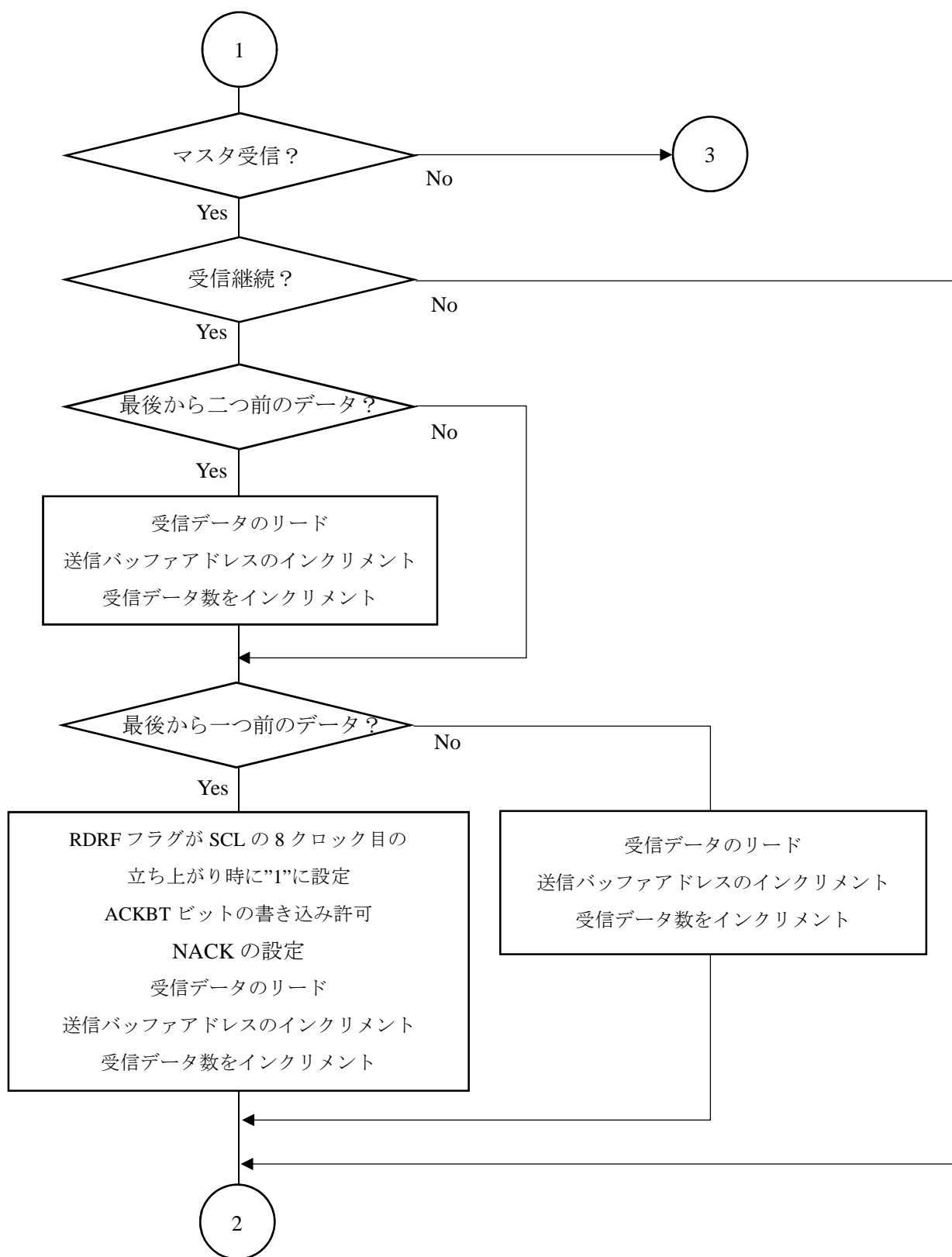
送信完了割り込み

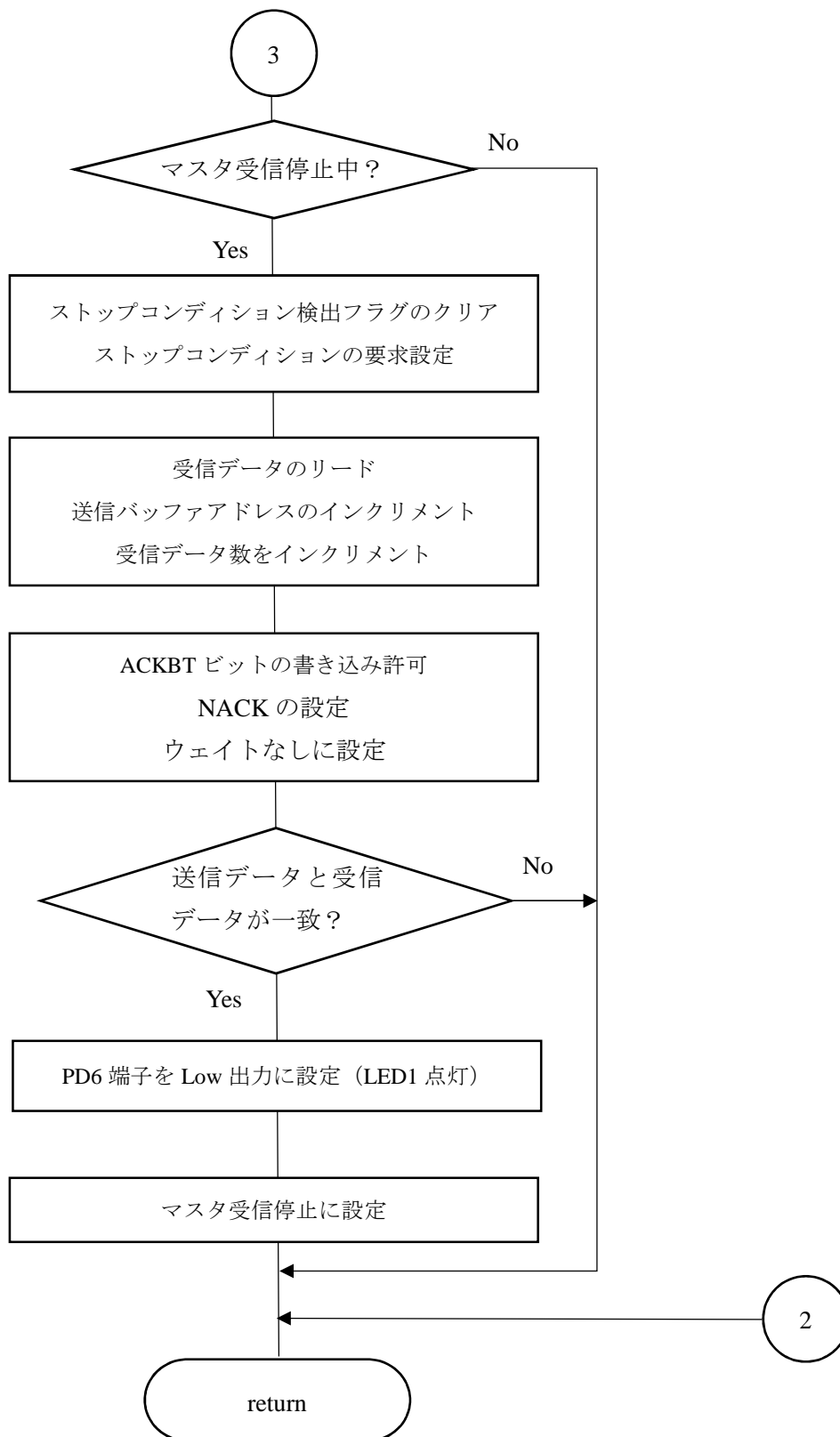


4.2.6.9

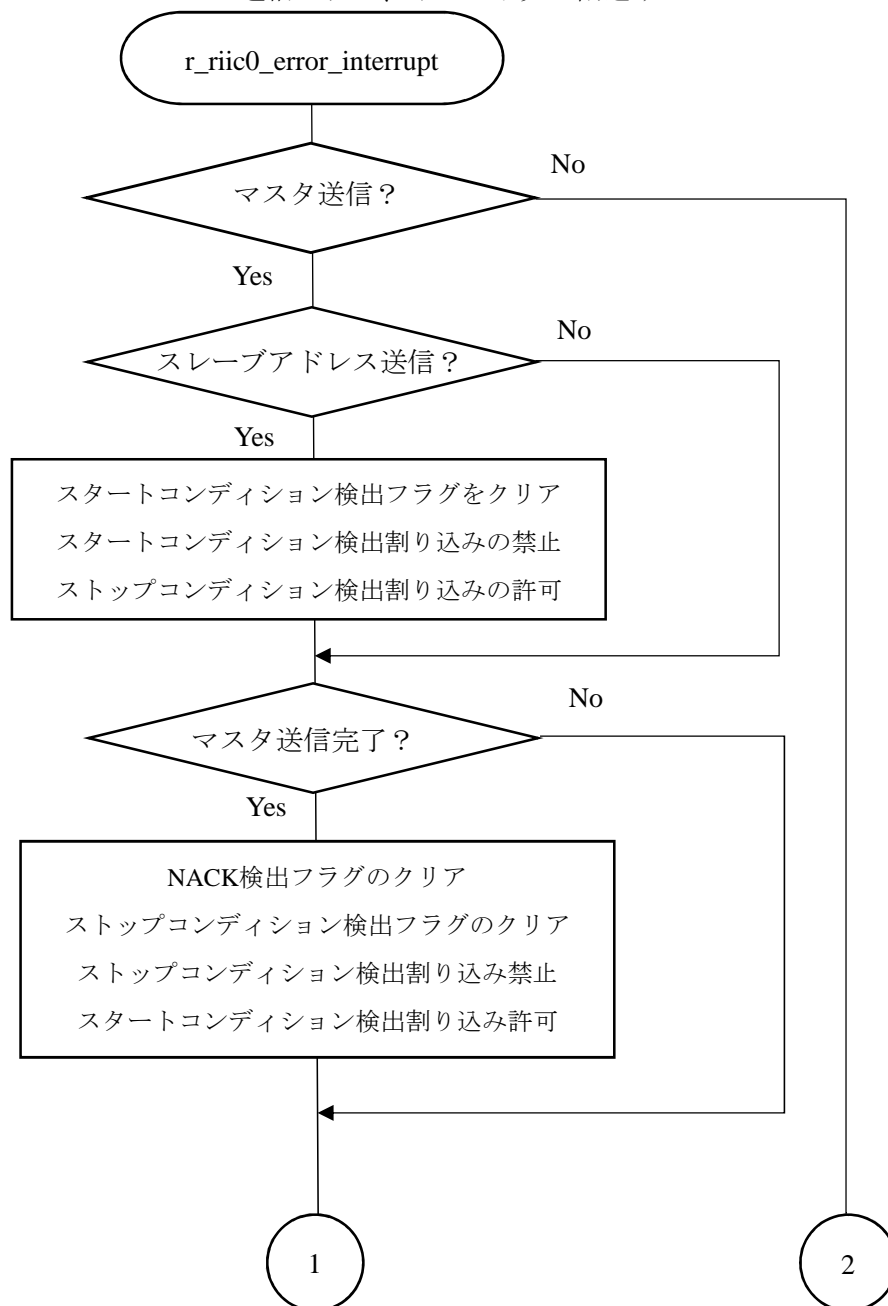
受信割り込み

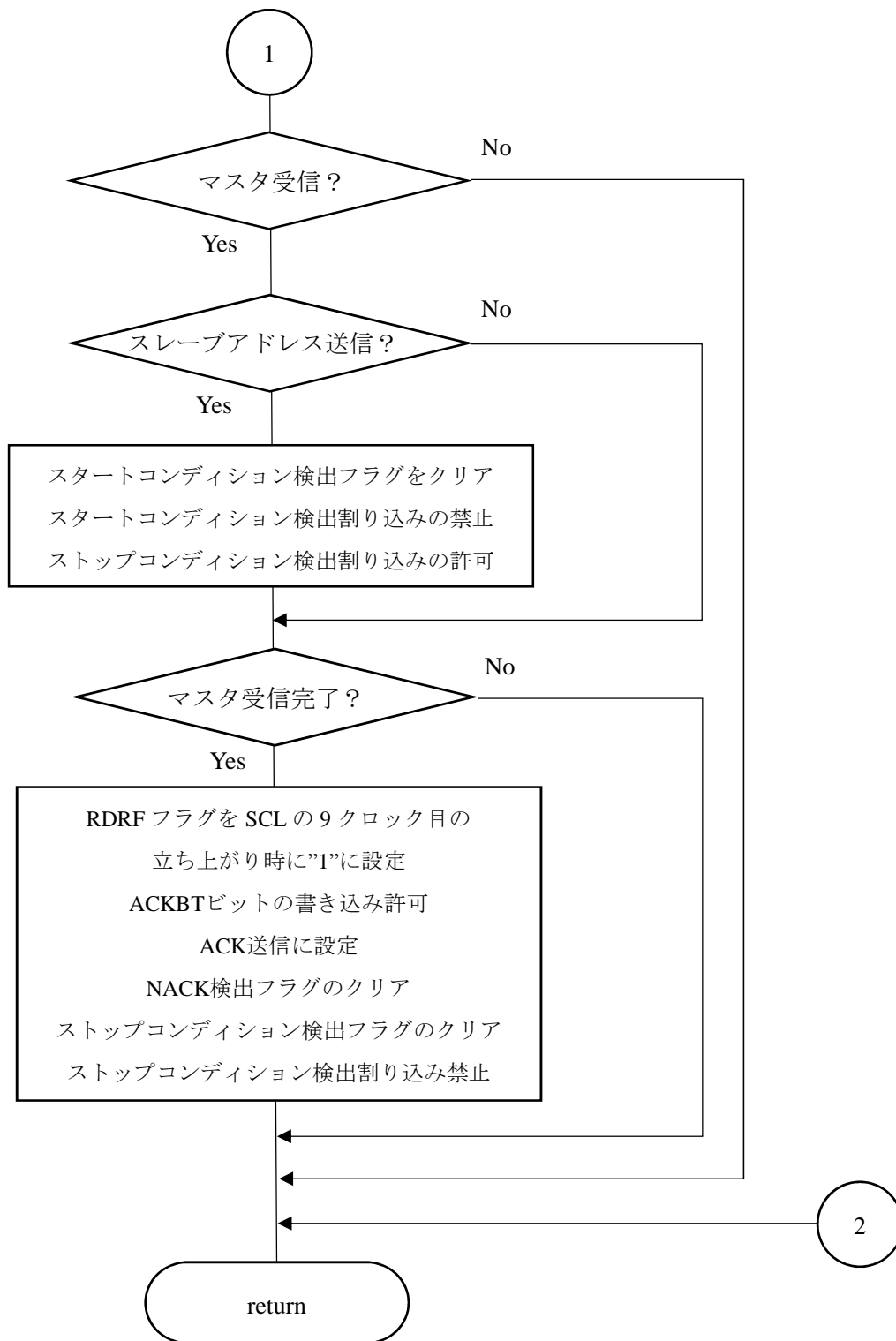






4.2.6.10 通信エラー、イベント発生割込み





4.3 スレーブ (RX210)

4.3.1 ファイル構成

本サンプルコードを作成するにあたり、編集したファイルを表 4.6 に示します。

表 4.6 ファイル名一覧

ファイル名	概要	備考
RIIC_RX210_SLAVE.c	メインファイル	

4.3.2 定義一覧

表 4.7 に本サンプルコードで使用する定義一覧を示します。

表 4.7 定義一覧

定数名	設定値	内容
SLAVE_RECEIVES_DATA	0x00U	スレーブ受信
SLAVE_SENDS_DATA	0x01U	スレーブ送信
SLAVE_SENDS_END	0x02U	スレーブ送信完了
SLAVE_SENDS_STOP	0x03U	スレーブ送信停止
SLAVE_RECEIVES_STOP	0x04U	スレーブ受信停止
SLAVE_WAIT_START_CONDITION	0x05U	スタートコンディション待ち
SLAVE_RECEIVE	0x06U	スレーブ受信
SLAVE_TRANSMIT	0x07U	スレーブ送信
SLAVE_RECEIVE_END	0x08U	スレーブ受信停止
BUFFER_SIZE	0x20U	バッファサイズ
RECEIVE_SIZE	5U	受信サイズ

4.3.3 変数一覧

表 4.8 に本サンプルコードで使用する変数一覧を示します。

表 4.8 変数一覧

型	変数名	内容
unsigned short	riic0_mode_flag	RIIC0 動作モード
unsigned short	riic0_state	RIIC0 状態データ
unsigned char*	riic0_rx_address	受信アドレス
unsigned char*	riic0_tx_address	送信アドレス
unsigned short	riic0_rx_count	受信データカウンタ
unsigned short	riic0_tx_count	送信データカウンタ
unsigned short	riic0_rx_length	受信データ数
unsigned short	riic0_dummy_read_count	ダミーリード用
unsigned char	riic0_buffer[BUFFER_SIZE]	受信（送信）バッファ

4.3.4 関数一覧

表 4.9 に関数一覧を掲載します。本サンプルコードで新規作成、編集した関数のみ記載しています。

表 4.9 関数一覧

関数名	概要
main	メイン処理
cgc_init	CGC の初期設定処理
port_init	ポートの初期設定処理
riic0_slave_init	RIIC0 の初期設定処理
riic0_slave_send	送信処理
riic0_slave_receive	受信処理
r_riic0_transmit_interrupt	送信割り込み
r_riic0_transmitend_interrupt	送信完了割り込み
r_riic0_receive_interrupt	受信割り込み
r_riic0_error_interrupt	エラー、イベント発生割り込み

4.3.5 関数仕様

main	
概要	メイン処理
ヘッダ	なし
宣言	void main(void)
説明	CGC、ポート、RIIC0 初期設定関数の呼び出し
引数	なし
リターン値	なし
cgc_init	
概要	CGC の初期設定処理
ヘッダ	なし
宣言	void cgc_init(void)
説明	CGC の初期設定処理
引数	なし
リターン値	なし

port_init

概要	ポートの初期設定処理
ヘッダ	なし
宣言	void port_init(void)
説明	ポートの初期設定処理
引数	なし
リターン値	なし

riic0_master_init

概要	RIIC0 の初期設定処理
ヘッダ	なし
宣言	void riic0_slave_init(void)
説明	RIIC0 の初期設定処理
引数	なし
リターン値	なし

riic0_master_send

概要	送信処理
ヘッダ	なし
宣言	void riic0_slave_send(unsigned char * const tx_buf, unsigned short tx_num)
説明	送信バッファから送信データサイズだけデータを送信する
引数	tx_buf 送信バッファ tx_num 送信データサイズ
リターン値	なし

riic0_master_receive

概要	受信処理
ヘッダ	なし
宣言	void riic0_slave_receive(unsigned char * const rx_buf, unsigned short rx_num)
説明	受信バッファに受信データサイズだけデータを受信する
引数	const rx_buf 受信バッファ rx_num 受信データサイズ
リターン値	なし

r_riic0_transmit_interrupt

概要	送信割り込み
ヘッダ	なし
宣言	static void r_riic0_transmit_interrupt(void)
説明	スレーブ送信動作
引数	なし
リターン値	なし

r_riic0_transmitend_interrupt

概要	送信完了割り込み
ヘッダ	なし
宣言	static void r_riic0_transmitend_interrupt(void)
説明	スレーブ送信完了処理
引数	なし
リターン値	なし

r_riic0_receive_interrupt

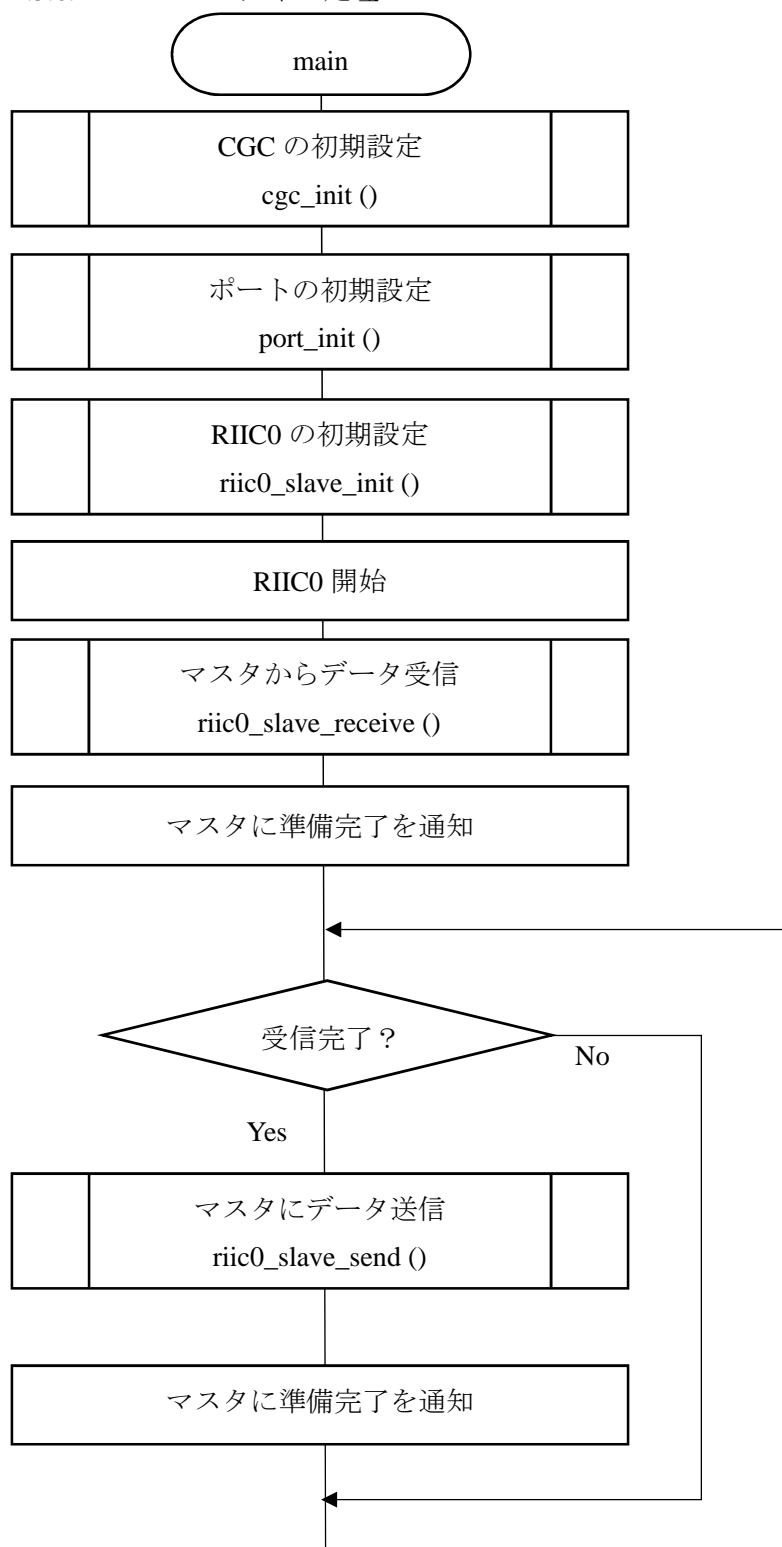
概要	受信割り込み
ヘッダ	なし
宣言	static void r_riic0_receive_interrupt(void)
説明	スレーブ受信処理
引数	なし
リターン値	なし

r_riic0_error_interrupt

概要	通信エラー、イベント発生割り込み
ヘッダ	なし
宣言	static void r_riic0_error_interrupt(void)
説明	通信エラー、イベント発生処理
引数	なし
リターン値	なし

4.3.6 作成する関数のフローチャート

4.3.6.1 メイン処理



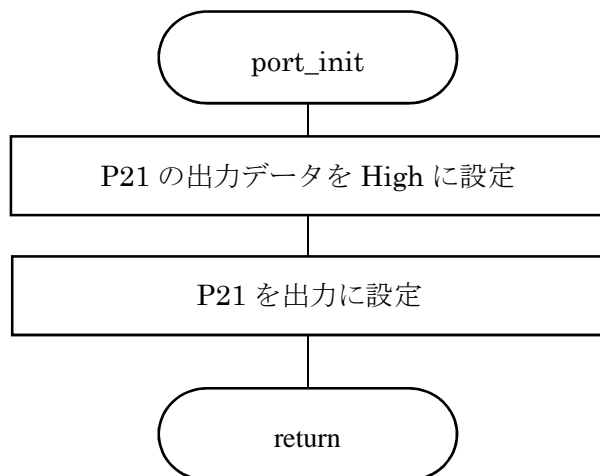
4.3.6.2

CGC の初期設定処理



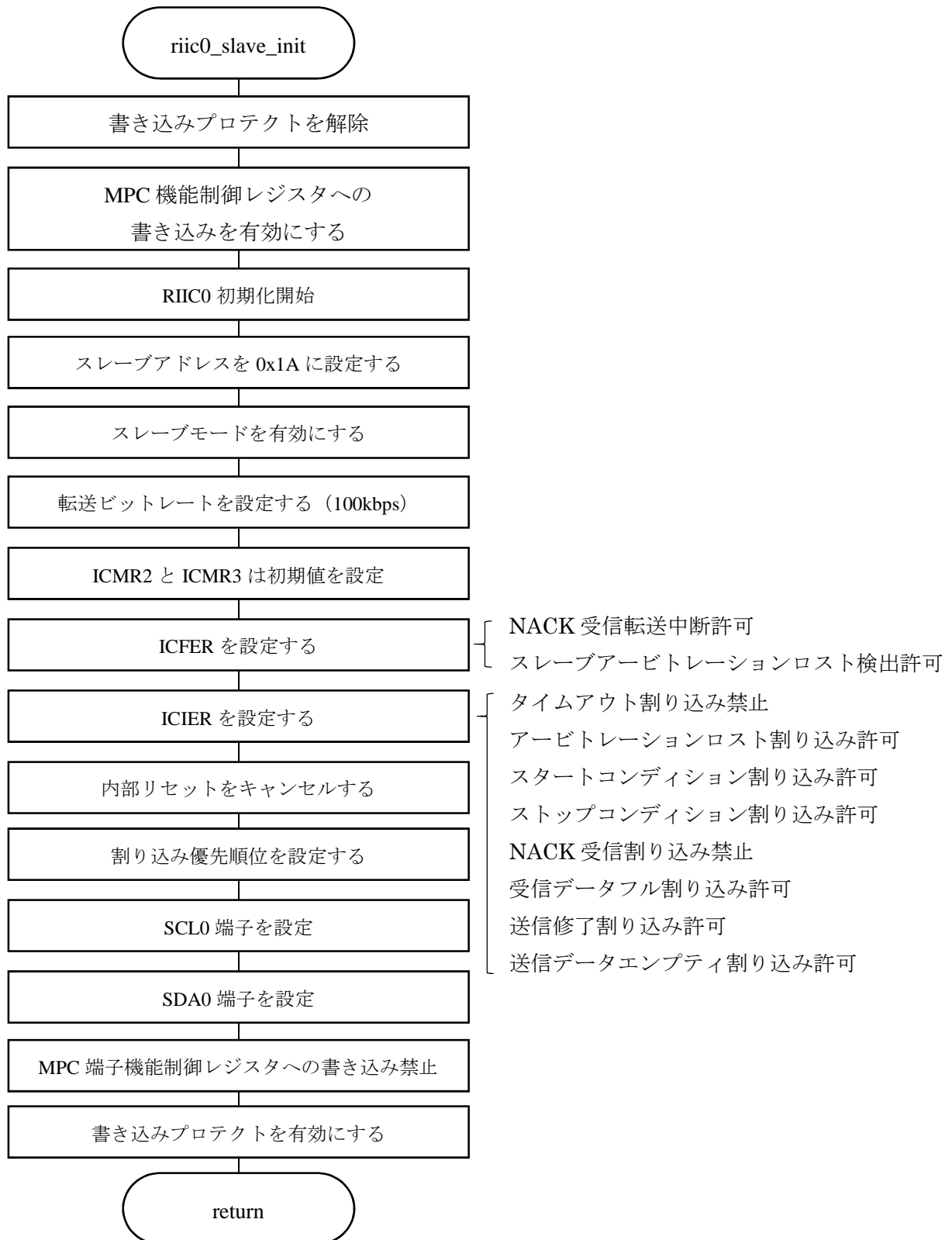
4.3.6.3

ポートの初期設定処理



4.3.6.4

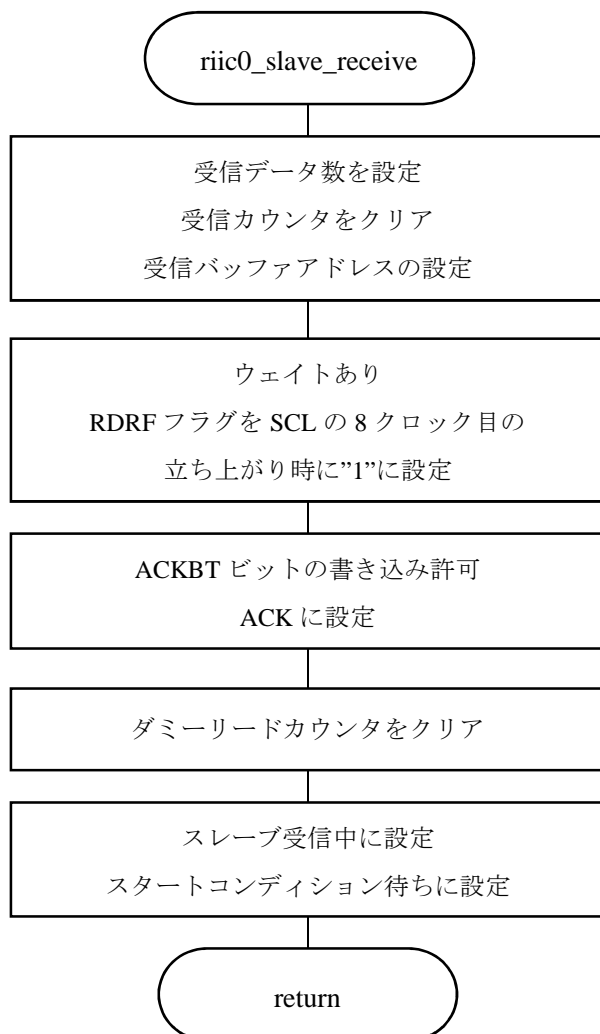
RIIC0 の初期設定処理



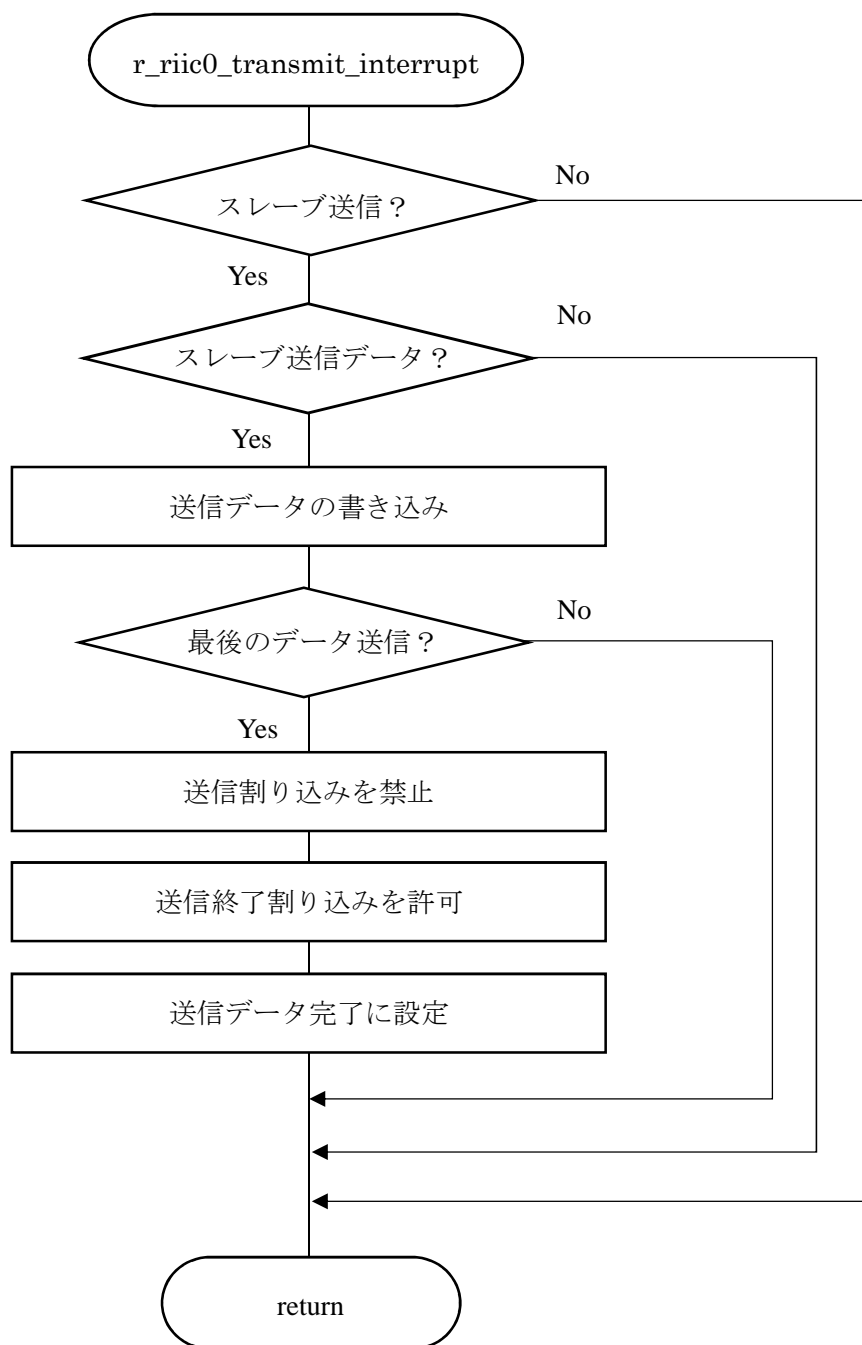
4.3.6.5 スレーブ送信の設定



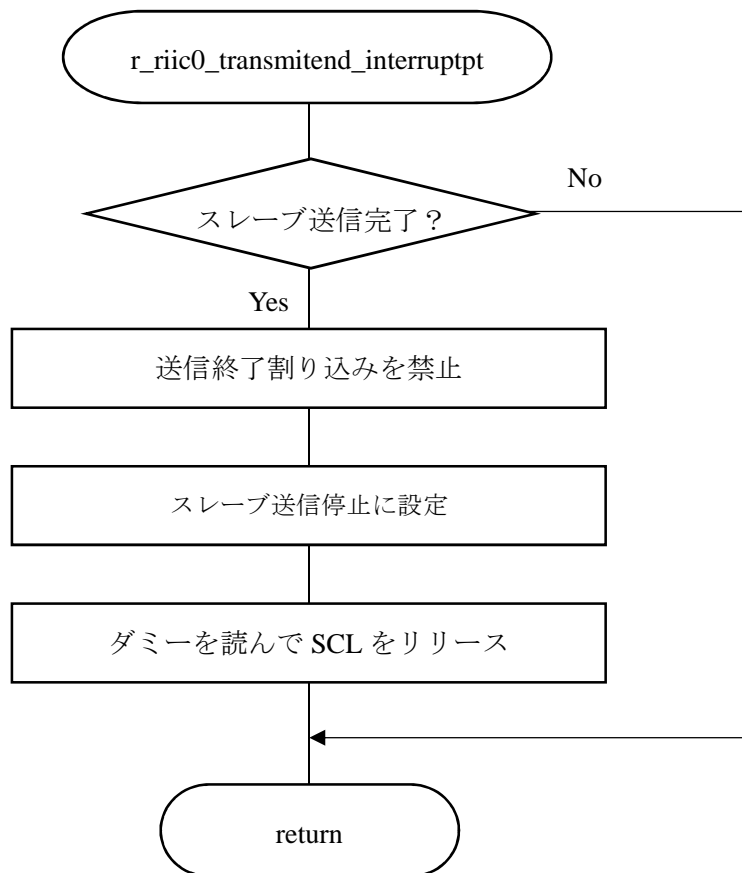
4.3.6.6 スレーブ受信設定



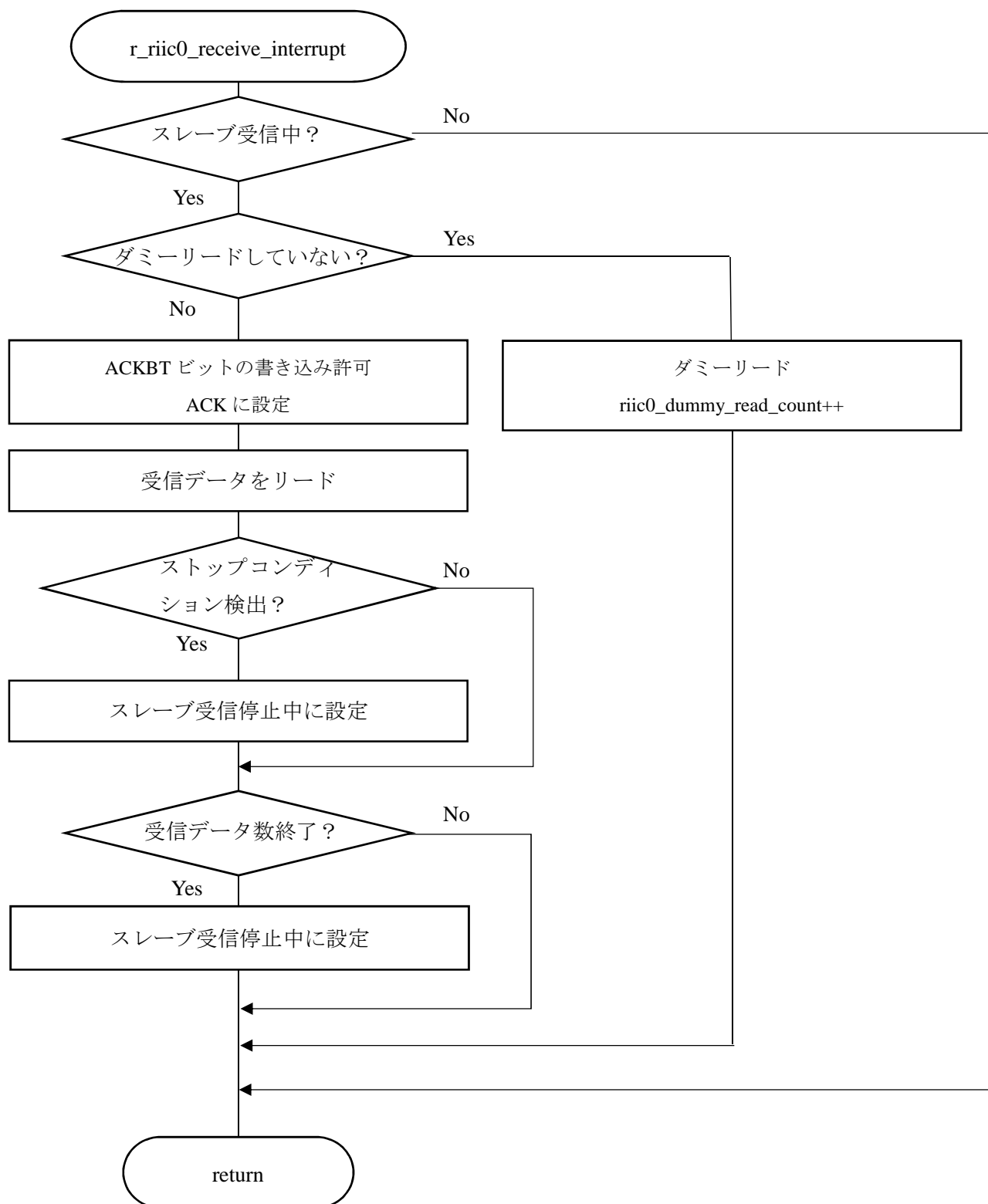
4.3.6.7 送信割り込み



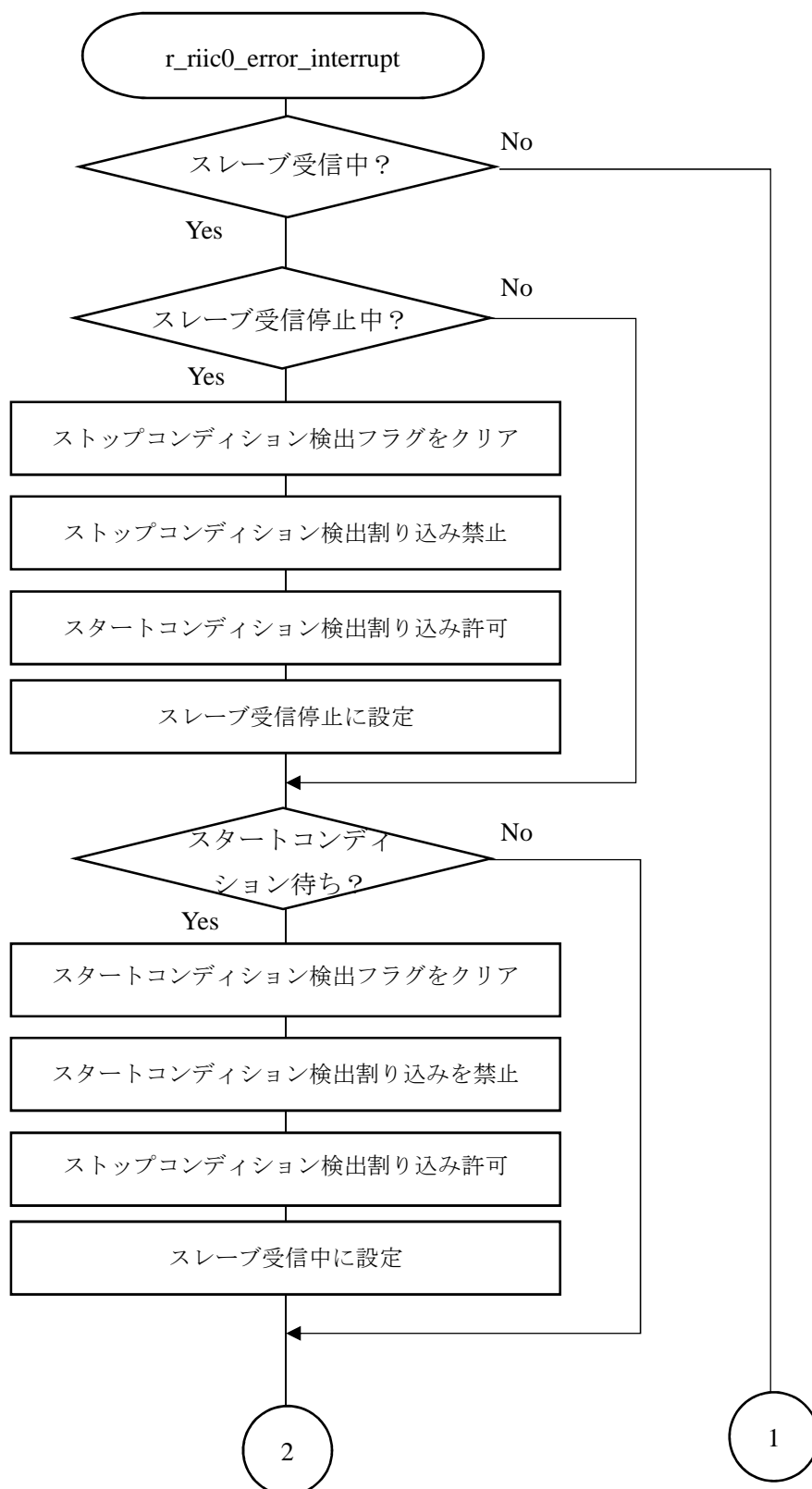
4.3.6.8 送信完了割り込み

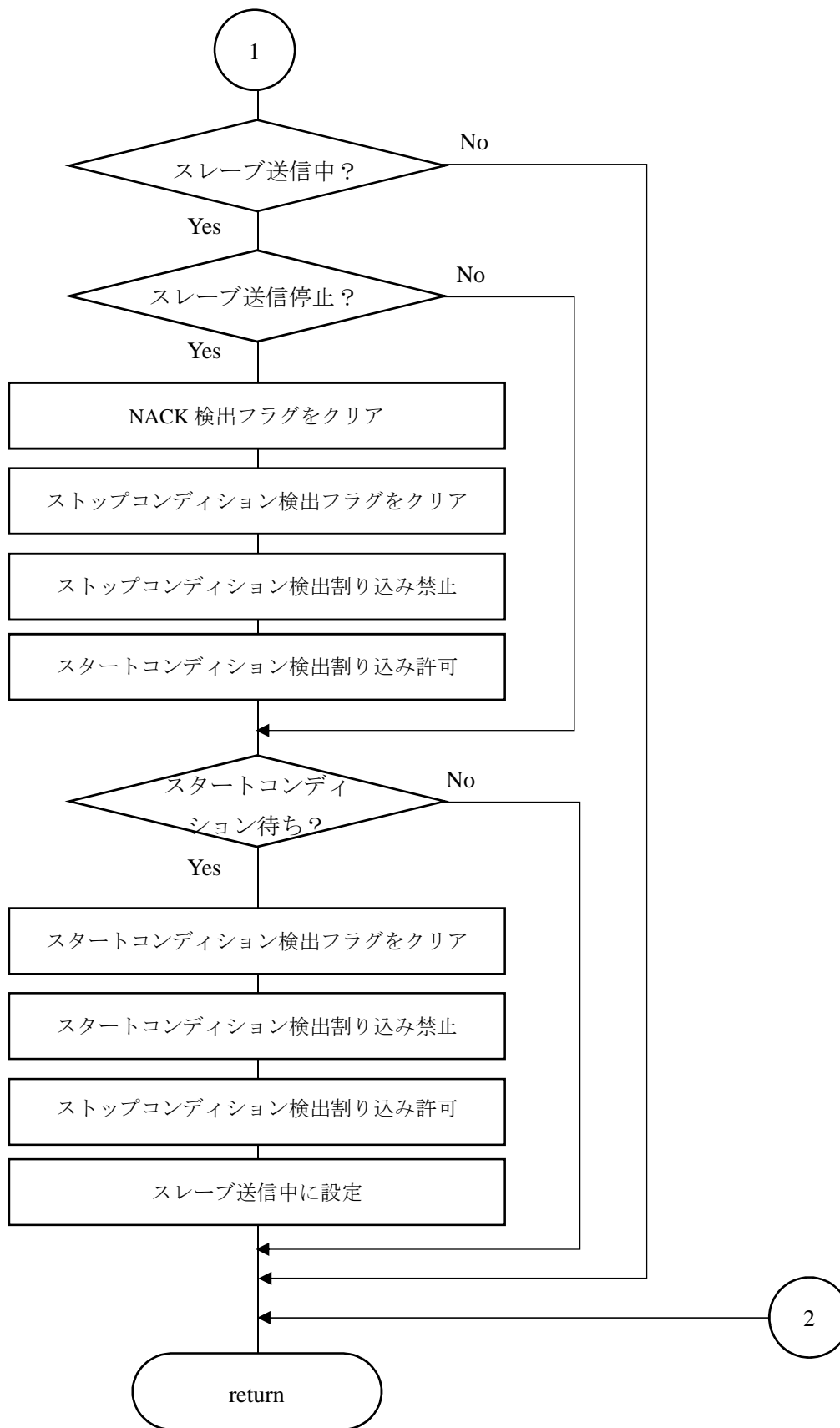


4.3.6.9 受信割り込み



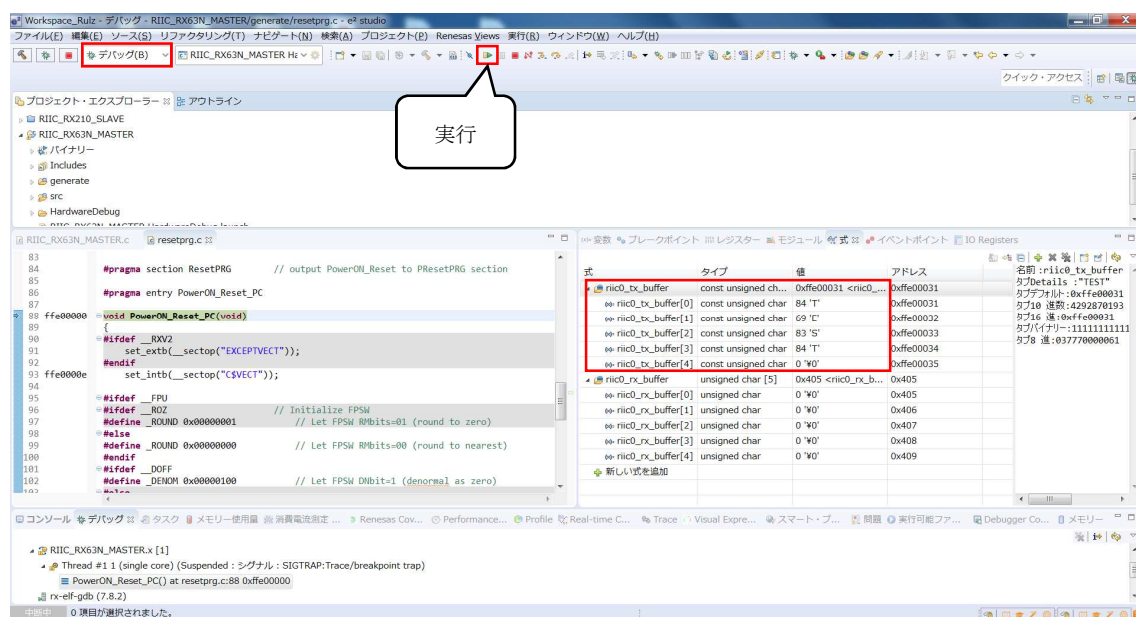
4.3.6.10 通信エラー、イベント発生割り込み



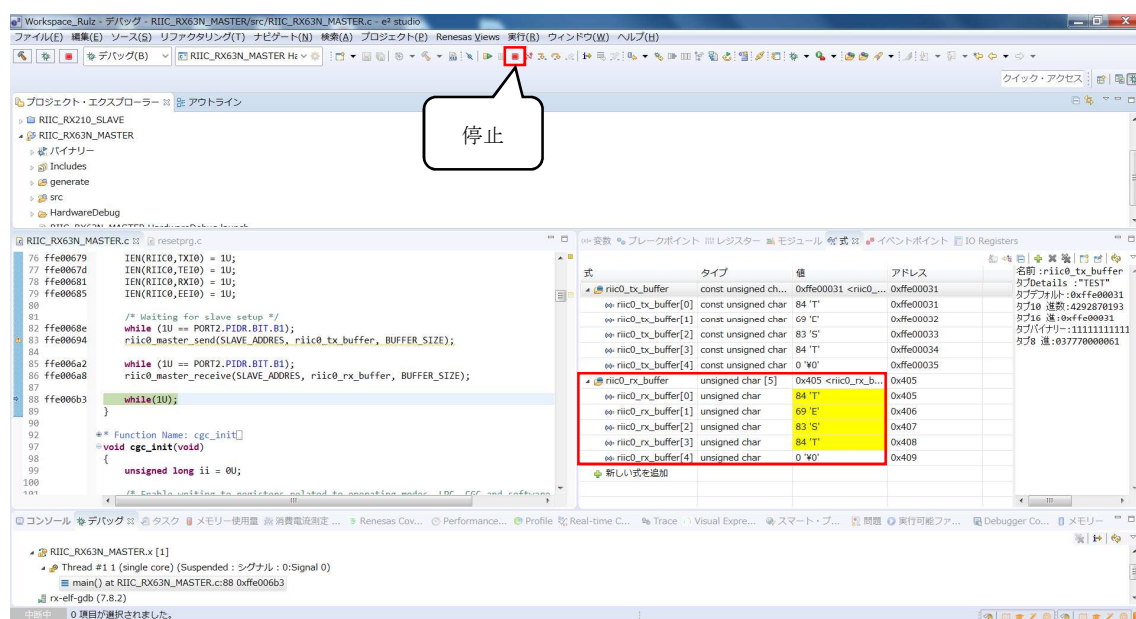


5. 動作確認方法

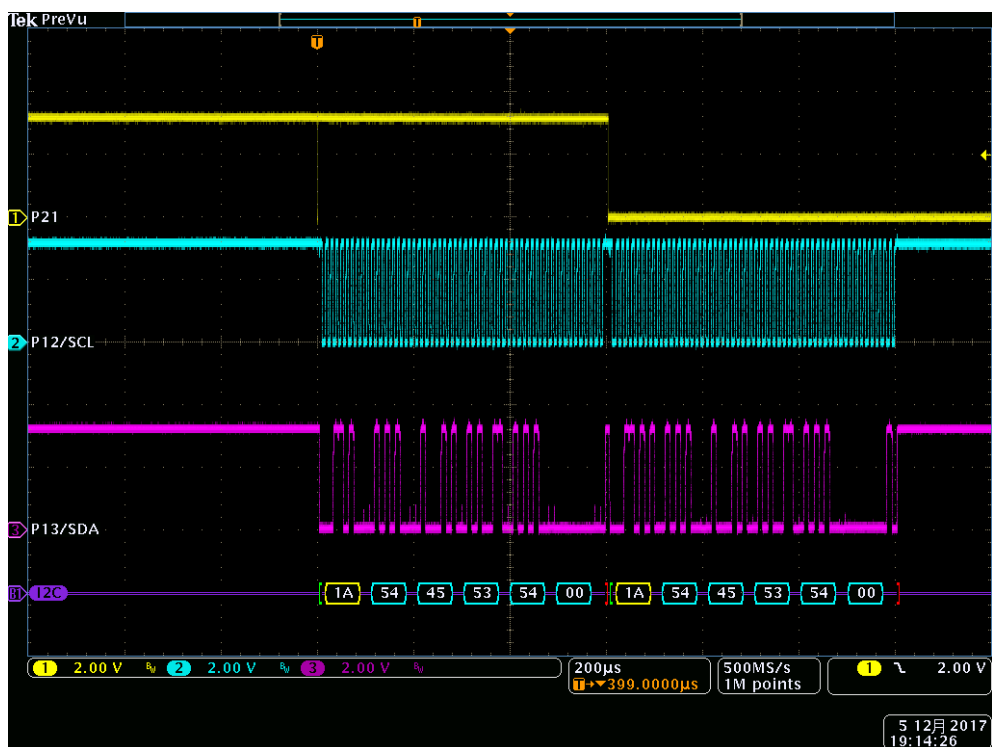
RX63N（マスタ）側にE1エミュレータを接続し、“デバッグ”ボタンをクリックし、プログラムをダウンロードします。送信バッファ（`riic0_tx_buffer`）に“TEST”の文字列が格納されており、受信バッファ（`riic0_rx_buffer`）には全て 00h が格納されています。データを確認後、“実行”ボタンをクリックします。



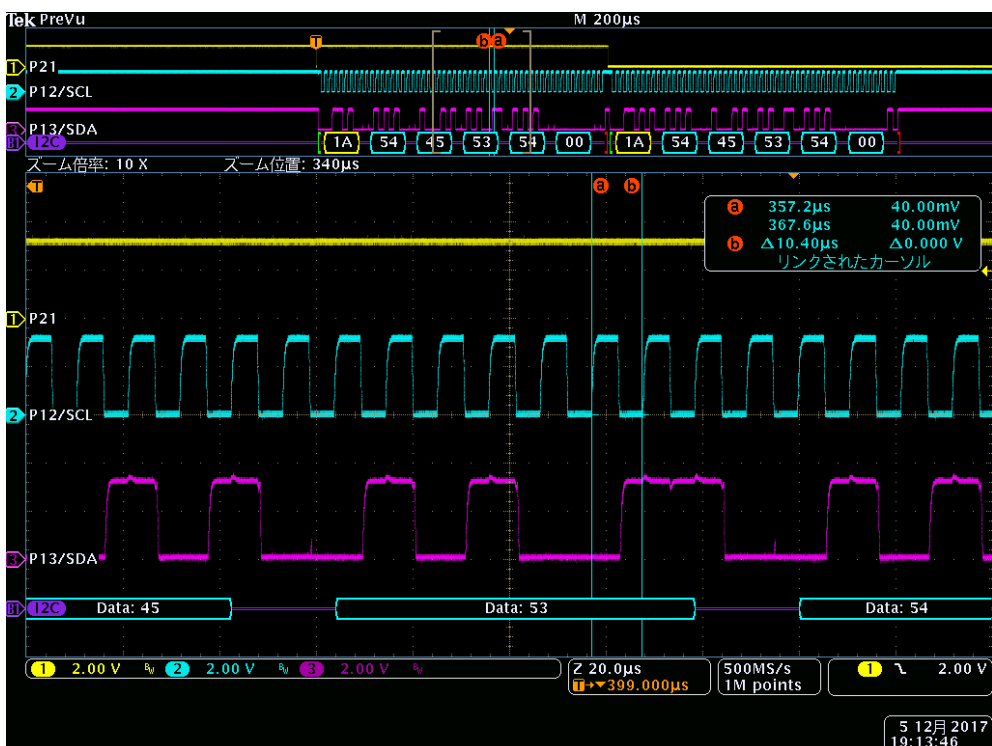
RX210（スレーブ）側を動作させ、“停止”ボタンを押すと、受信バッファ（`riic0_rx_buffer`）に“TEST”の文字列が格納されていることが確認できました。



測定した I2C 波形を以下に示します。また、送受信後に LED1 が点灯することを確認しました。



ズームした波形は以下になります。SCL 端子から 100kHz のクロック出力を確認しました。



6. 参考ドキュメント

RX63N グループ ユーザーズマニュアル ハードウェア編

HSBRX63NP シリーズ HSBRX631P シリーズ 取扱説明書

RX210 グループ ユーザーズマニュアル ハードウェア編

HSBRX210-100B シリーズ 取扱説明書

以上