

RXファミリ

既存の周辺機能を応用した低消費電力モード中の通信方法

要旨

本アプリケーションノートはRXファミリに搭載されているSCIやSPI等と外部端子割込み(IRQn)を連携させ低消費電力モード中に外部IOと通信可能な機能を実現する方法を説明します。

本アプリケーションノートの対象は、以下の対象デバイスです。本アプリケーションノートの内容をRX66Nグループ以外のマイコンに適応する場合は、対象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。

対象デバイス

RXファミリデバイス

動作確認デバイス

RX66Nグループ

目次

1. 概要	3
1.1 背景	3
1.2 低消費電力モード中の通信方法	3
1.3 システム構成	4
1.3.1 LEDによる状態表示	4
1.3.2 Target Board for RX66Nの改造	5
1.3.3 コマンド一覧	5
1.3.4 ターミナルソフト(TeraTerm)の使用方法	6
2. 動作確認条件	10
3. 動作環境	11
3.1 接続方法	11
3.2 サンプルプログラムの書き込み方法	12
4. システムの説明	15
4.1 低消費電力モード中の通信方法の動作説明	15
5. ソフトウェアの説明	16
5.1 構成	16
5.1.1 ファイルの構成	16
5.1.2 定数一覧	16
5.1.3 変数一覧	16
5.1.4 関数一覧	17
5.1.5 関数仕様	18
5.2 フローチャート	20
5.2.1 メイン関数	20
5.2.2 Lpmode関数	21
5.2.3 R_Config_SCI12_Create_UserInit関数	22
5.2.4 r_Config_ICU_irq7_interrupt関数	22
5.2.5 r_Config_SCI12_callback_receiveerror関数	23
5.2.6 r_Config_SCI12_callback_receiveend関数	23
6. 参考ドキュメント	24
改訂記録	25

1. 概要

1.1 背景

バッテリー駆動のアプリケーションなどでは、一般的に常に低消費電力モードに滞在し、定期的もしくは外部IOなどからのデータ受信によりCPUを復帰させ処理を実行します。その処理終了後は再度低消費電力モードに遷移することでシステムのバッテリー消費を少なくし、より長期間サービスを提供したいというニーズがあり、RXファミリに搭載されているSCIやSPI等と外部端子割込み(IRQn)を連携させ低消費電力モード中に外部IOとの通信方法を説明します。

1.2 低消費電力モード中の通信方法

本サンプルプログラムでは、RXファミリに広く内蔵されているSCI(シリアルコミュニケーションインタフェース)とIRQn(外部割込み端子)を応用して実現します。

1.3 システム構成

図 1-1にシステム構成図を示します。また、表 1-1 使用する周辺機能と用途を示します。

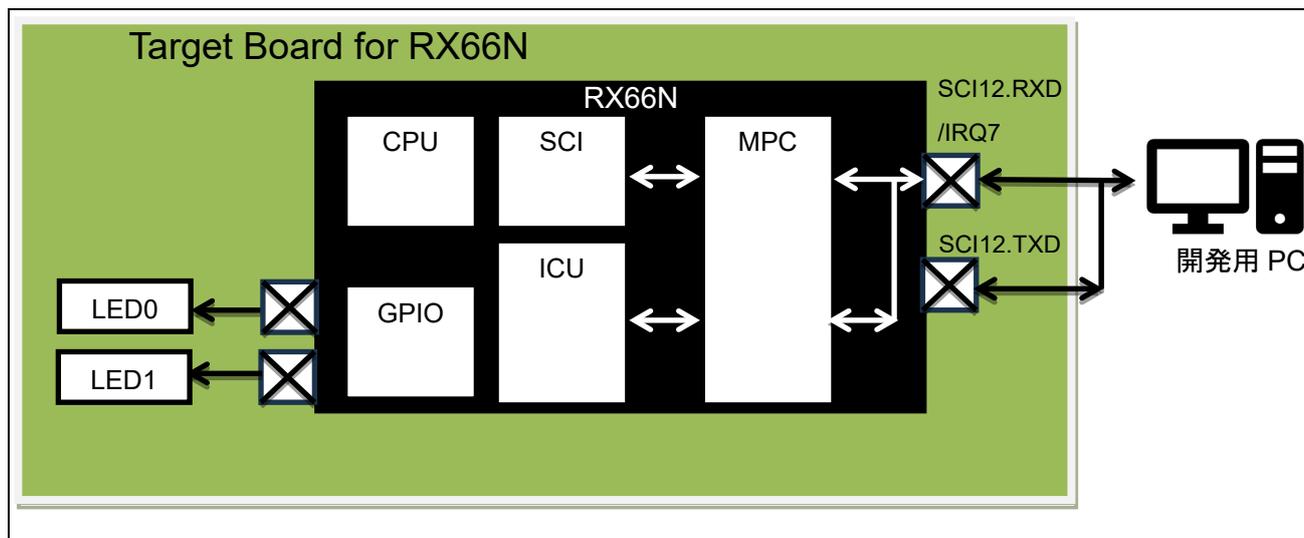


図 1-1 システム構成図

表 1-1 使用する周辺機能と用途

周辺機能	用途
ICU	IRQ 割り込み受付、CPU へ割り込み
SCI12	UART 通信インターフェース
MPC	低消費電力モード遷移時:ポート E2 を IRQ7 低消費電力モード復帰時:ポート E2 を SCI12.RXD へ切り替え
GPIO	ポート E2:SCI12.RXD/IRQ7 ポート E1:SCI12.TXD ポート 32:LED0 の点灯/滅灯制御 ポート 33:LED1 の点灯/滅灯制御

1.3.1 LED による状態表示

LED0 でRX66Nグループの状態を確認可能です。

表 1-2 LED0 によるRX66Nグループの状態

LED0	状態
OFF	低消費電力モード中
ON	通常動作中

LED1 は、外部 IO からの LED 点灯、滅灯コマンドで点灯、滅灯します。

表 1-3 LED1 によるコマンド実行結果

LED1	状態
OFF	LED 滅灯コマンドで滅灯
ON	LED 点灯コマンドで点灯

1.3.2 Target Board for RX66Nの改造

開発用 PC との接続に使用する USB シリアル変換ケーブルとTarget Board for RX66Nを接続するため、Target Board for RX66Nにジャンパケーブル(オス⇄オス)×2 本を実装する必要があります。CN3 の 76 ピンと 77 ピンにジャンパケーブルを 1 本ずつ実装してください。

図 1-2にジャンパケーブルの実装位置を示します。

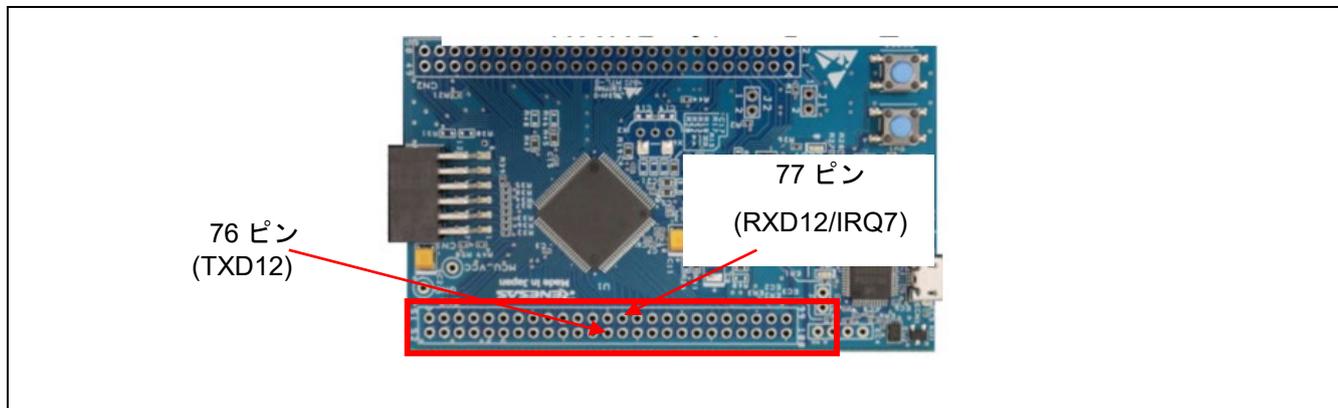


図 1-2 ジャンパケーブル実装位置

なお、径が細い(0.5mm 未満)ジャンパ線だと後述の USB シリアル変換ケーブルにしっかり接続できない可能性があるので使用しないでください。

1.3.3 コマンド一覧

RX66Nグループと外部 IO との通信はノイズや意図しないデータ受信による起動などの誤動作を排除するため、コマンド形式とします。

表 1-4に使用するコマンドの一覧、図 1-3にコマンドフォーマットを示します。

表 1-4 コマンドの一覧

No.	コマンド	動作
1	CMON	LED1 を点灯
2	CMOF	LED1 を滅灯
3	CMRE	外部 IO への再送要求
4	CMRV	外部 IO への受信完了通知

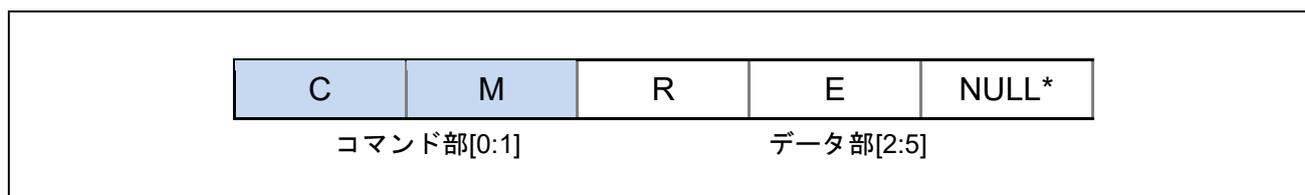


図 1-3 コマンドフォーマット

*コマンド通信時、その終了を表すため、必ず最後に NULL を入れています。

1.3.4 ターミナルソフト(TeraTerm)の使用方法

本サンプルプログラムでは、RX66Nグループと外部IOに見立てたPCとの通信にTeraTermを使います。以下にTeraTermの使用方法を示します。なお、TeraTermはお客様で事前にPCにインストールをお願いします。

- a TeraTermを起動しポートを選択します
「シリアル(E)」を選択し、ご利用の環境に合った「ポート(R)」を選択し、「OK」をクリックします。図1-4では、COM8を使用する場合です。



図 1-4 TeraTermの起動、ポートの設定

- b TeraTermの端末設定を行います。
設定>端末を選択し、下図の赤枠の設定に変更します。

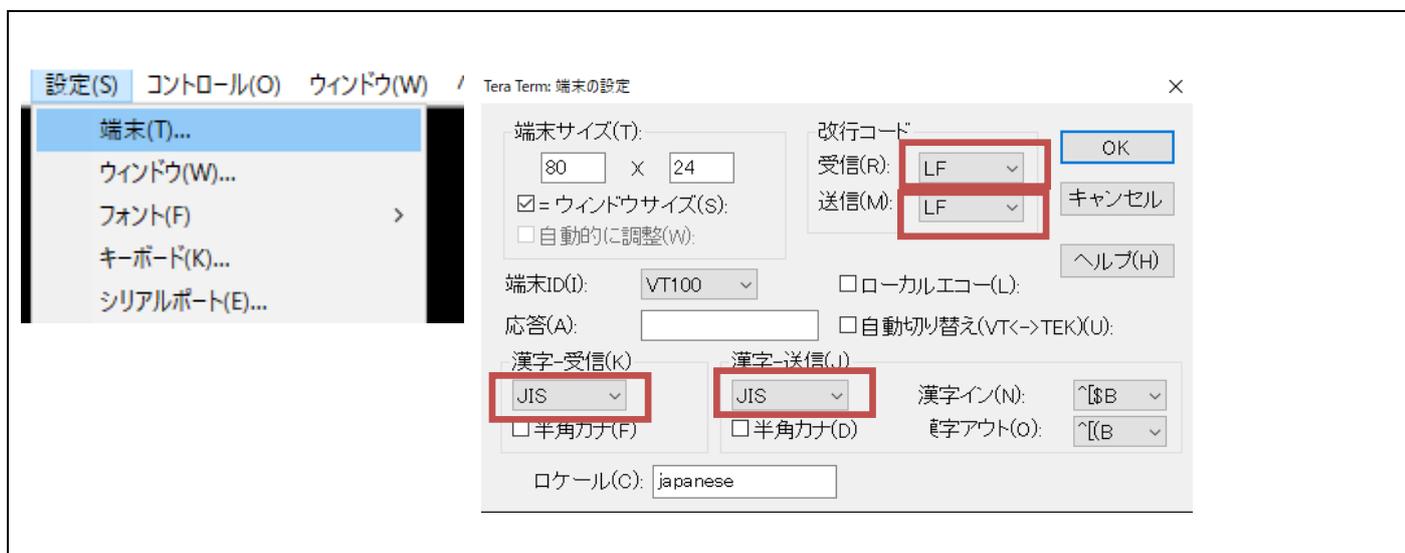


図 1-5 TeraTermの端末設定

- c TeraTerm でのシリアル設定を行います
設定>シリアルポートを開き下図の設定に変更します。

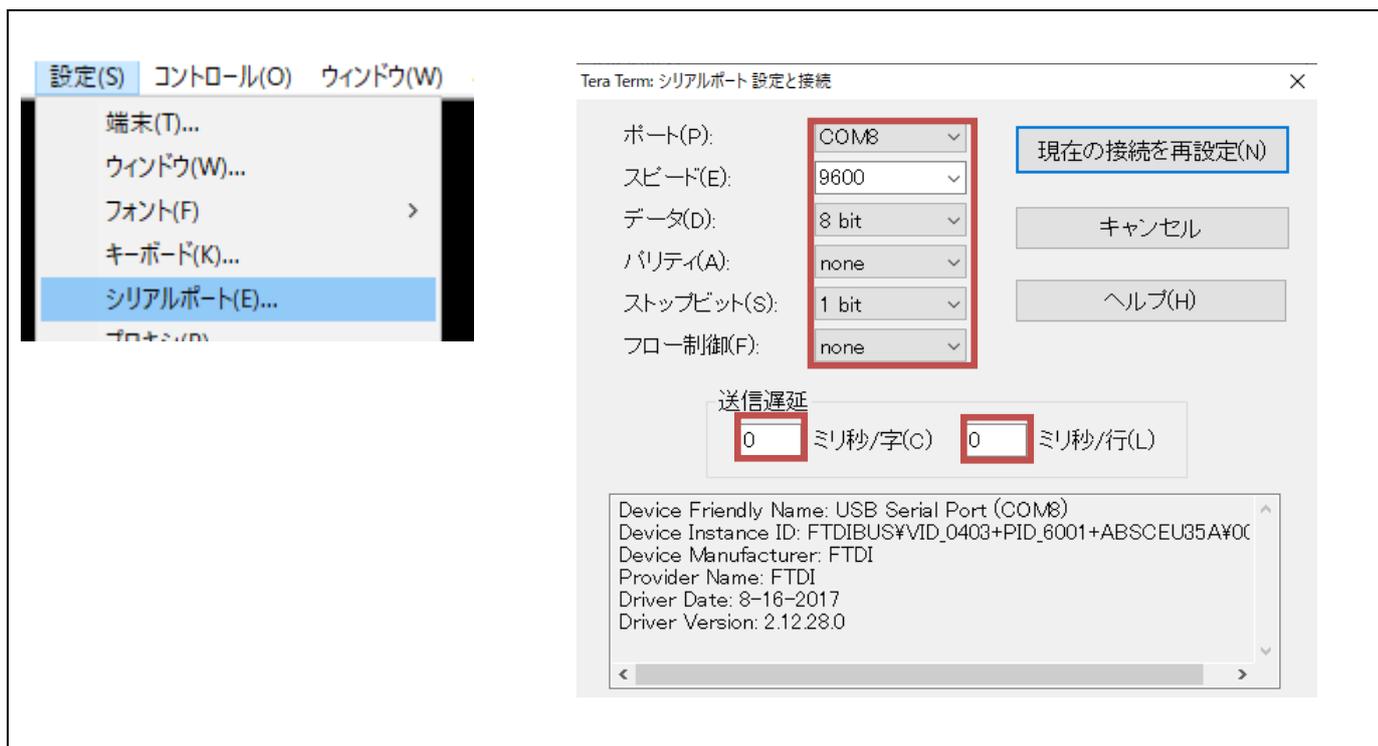


図 1-6 シリアル通信の設定

- d コントロール(O)>マクロ (M) を選択し

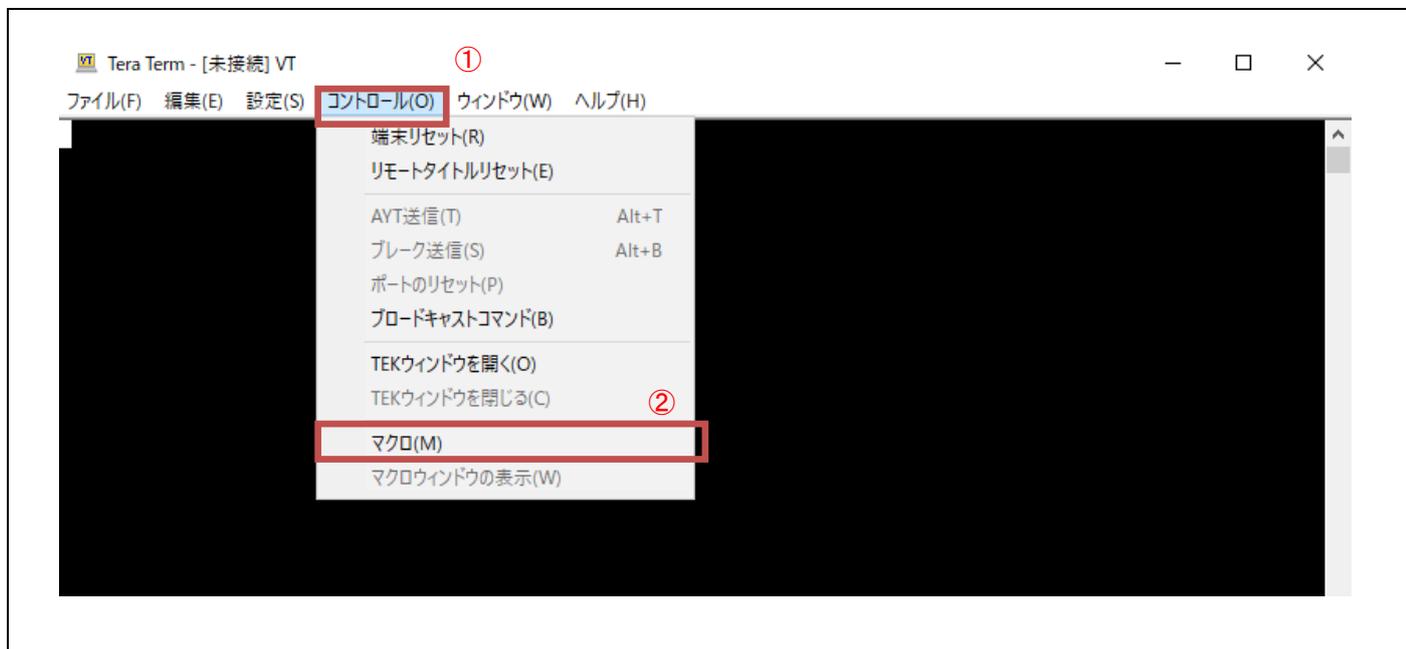


図 1-7 マクロの起動

- e 立ち上がったエクスプローラウィンドウから“send_test.ttl”を開きます。
この“send_test.ttl”は本サンプルコードに同梱されています。

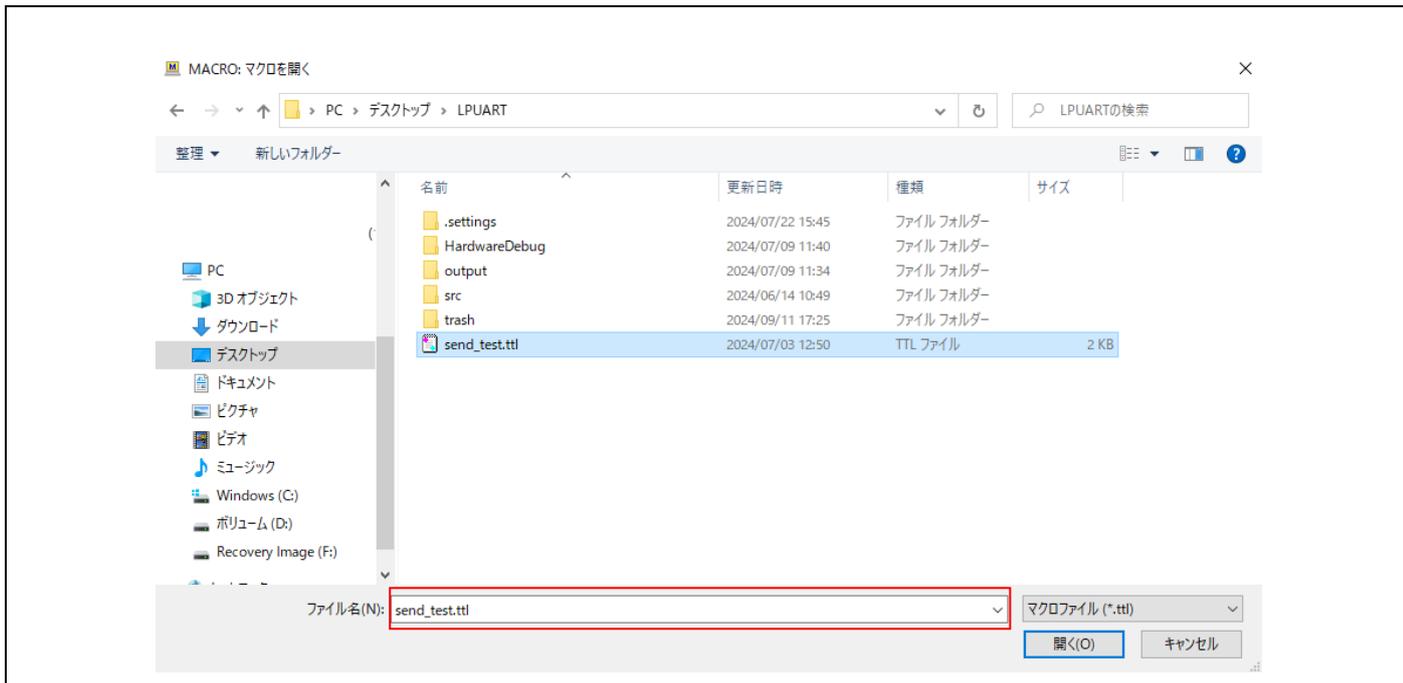


図 1-8 マクロの選択

- f コマンド入力用のウィンドウが立ち上がるので以下のいずれかのコマンドを入力してください。
- ・ CMON :LED1 を点灯させるコマンドを送信
 - ・ CMOF :LED1 を滅灯させるコマンドを送信
- “OK” をクリック、もしくは ENTER キーを押下すると入力したコマンドが送信されます。

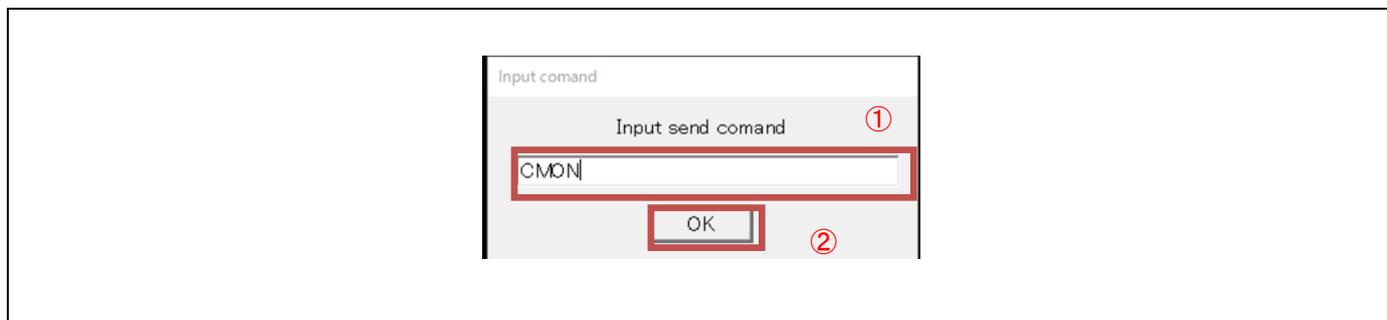


図 1-9 コマンドの送信

- g RX66Nグループが再送されたコマンドを受信すると、受信コマンド(CMRV)が送信され、以下のポップアップが表示されます。

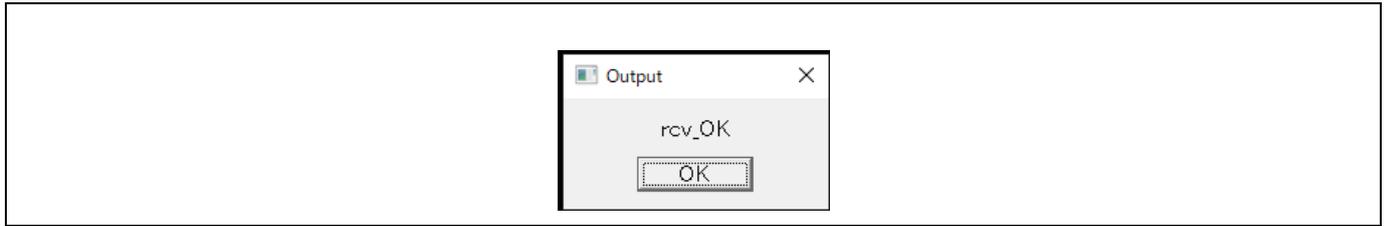


図 1-10 RX66Nグループからの受信完了通知

注:再送要求(CMRV)が来た場合、TeraTerm は最初の送信コマンドを覚えているので、TeraTerm が自律で最初のコマンドを再送します。よって、再度コマンドを入力する必要はありません。

2. 動作確認条件

本アプリケーションノートのサンプルプログラムは、以下に示す条件で動作を確認しています。

表 2-1 動作確認環境

項目	内容
使用 MCU	R5F566NNHDFP (Target Board for RX66N 搭載)
動作周波数	メインクロック : 停止 PLL : 240MHz (HOCO x 1/1 x 15) HOCO : 16MHz LOCO : 停止 システムクロック (ICLK) : 120MHz (PLL x 1/2) 周辺モジュールクロック A (PCLKA) : 120MHz (PLL x 1/2) 周辺モジュールクロック B (PCLKB) : 60MHz (PLL x 1/4) 周辺モジュールクロック C (PCLKC) : 60MHz (PLL x 1/4) 周辺モジュールクロック D (PCLKD) : 60MHz (PLL x 1/4) FlashIF クロック (FCLK) : 60MHz (PLL x 1/4)
動作電圧	3.3V
総合開発環境	ルネサスエレクトロニクス e ² studio Version 2024-07.0
C コンパイラ ^注	ルネサスエレクトロニクス C/C++ Compiler Package for RX Family V3.06.00
RX スマート・コンフィグレータ	V2.22.0
ボードサポートパッケージ (r_bsp)	V7.50
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
サンプルコードバージョン	V1.00
使用ボード	Target Board for RX66N (型名 : RTK5RX66N0C00000BJ)
エミュレータ	E2-Lite

注 本プロジェクトで指定するツールチェーン(C コンパイラ) と同一のバージョンがインストールされていない場合は、ツールチェーンが選択されない状態になり、エラーが発生します。
エラーが発生した場合は、プロジェクトの設定画面でツールチェーンの選択状態を確認してください。

設定方法は、FAQ 3000404 を参照してください。

[FAQ 3000404 :インポートしたプロジェクトをビルドすると「PATH でプログラム"make"が見つかりません」エラーになる\(e² studio\)](#)

3. 動作環境

本アプリケーションのサンプルプログラムの書き込みには、Target Board for RX66Nと開発用 PC を USB で接続し、書き込む必要があります。

ここでは、Target Board for RX66N搭載のオンチップデバッガを使って書き込みします。

表 3-1にサンプルプログラムの書き込み環境、図 3-2に接続方法を示します。

表 3-1 サンプルプログラムの書き込み環境

項目	内容
開発環境	<ul style="list-style-type: none"> 開発用 Windows PC e²studio Version 2024-07.0
デバッグツール	<ul style="list-style-type: none"> E2 Lite(RX)
ボード	<ul style="list-style-type: none"> Target Board for RX66N (型名 : RTK5RX66N0C00000BJ)
ケーブル	<ul style="list-style-type: none"> USB シリアル変換ケーブル(type-A ⇄ シリアルコネクタ)* ジャンパケーブル(オス ⇄ オス)

*本ケーブルは、実装したジャンパーケーブル(オス⇄オス)と接続するので、シリアルコネクタ側がメスになっているものをご用意ください。



図 3-1 USB シリアル変換ケーブル (ご参考)

3.1 接続方法

- CN3 の 76 ピン (RXD12/IRQ7) と 77 ピン(TXD12)に実装したジャンパケーブルと USB シリアル変換ケーブルのシリアルコネクタを接続します。
- USB ケーブルの type-A 側を開発用 PC の USB ポートに接続します。これでオンチップデバッグエミュレータに接続されます。

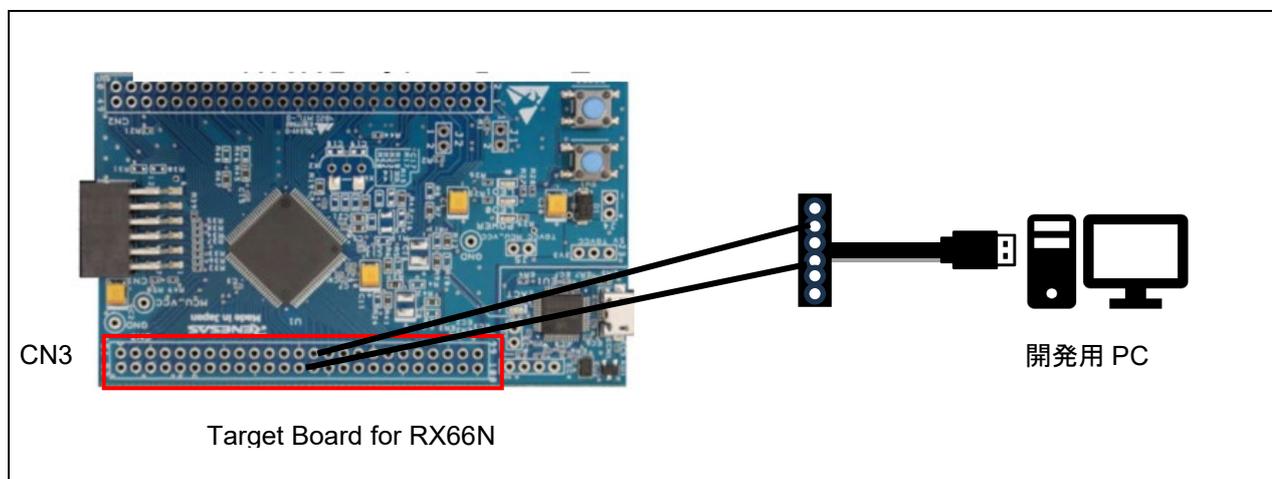


図 3-2 Target Board for RX66N 接続図

3.2 サンプルプログラムの書き込み方法

接続が完了したら、以下手順に従いプロジェクトの書き込みを行います。

a. プロジェクトをインポートします。

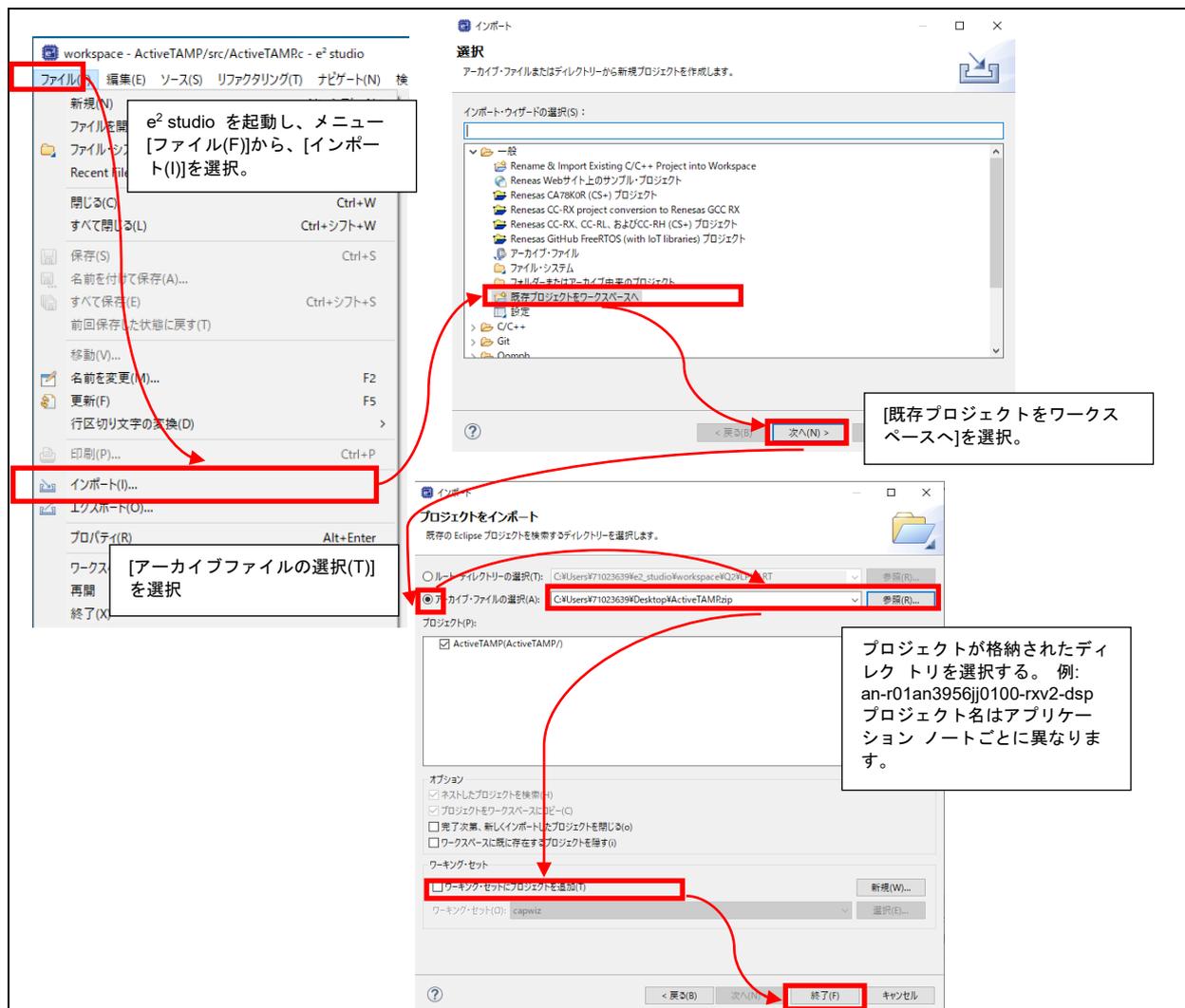


図 3-3 プロジェクトのインポート

- b. プロジェクトをビルド後、e² studio のルールとしてデバッグボタンを押します。これで書き込みが開始します

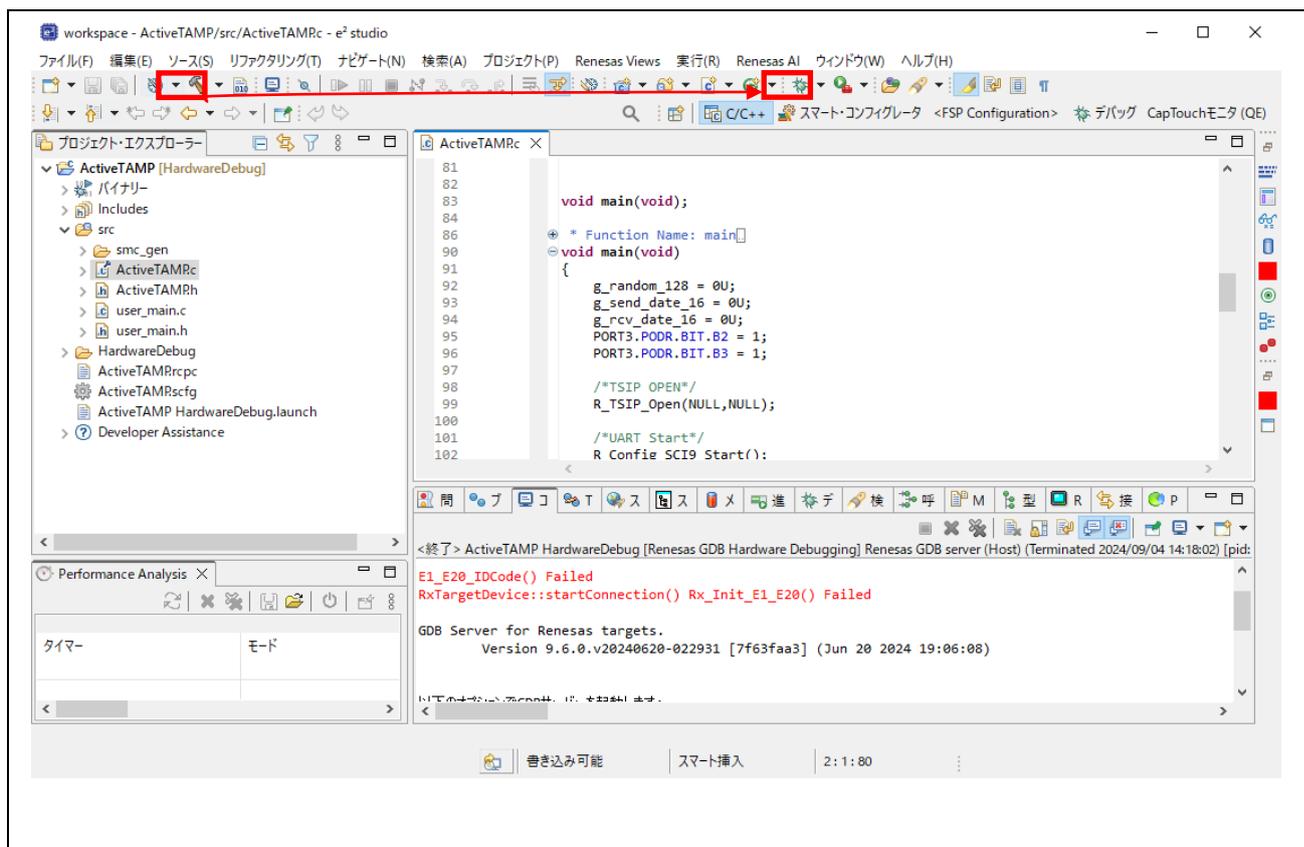


図 3-4 ビルド、デバッグ

- c. 書き込み後、自動的にFINE ブートモードで起動するため終了を押下して、e² studio とオンチップデバッグエミュレータを切断します。

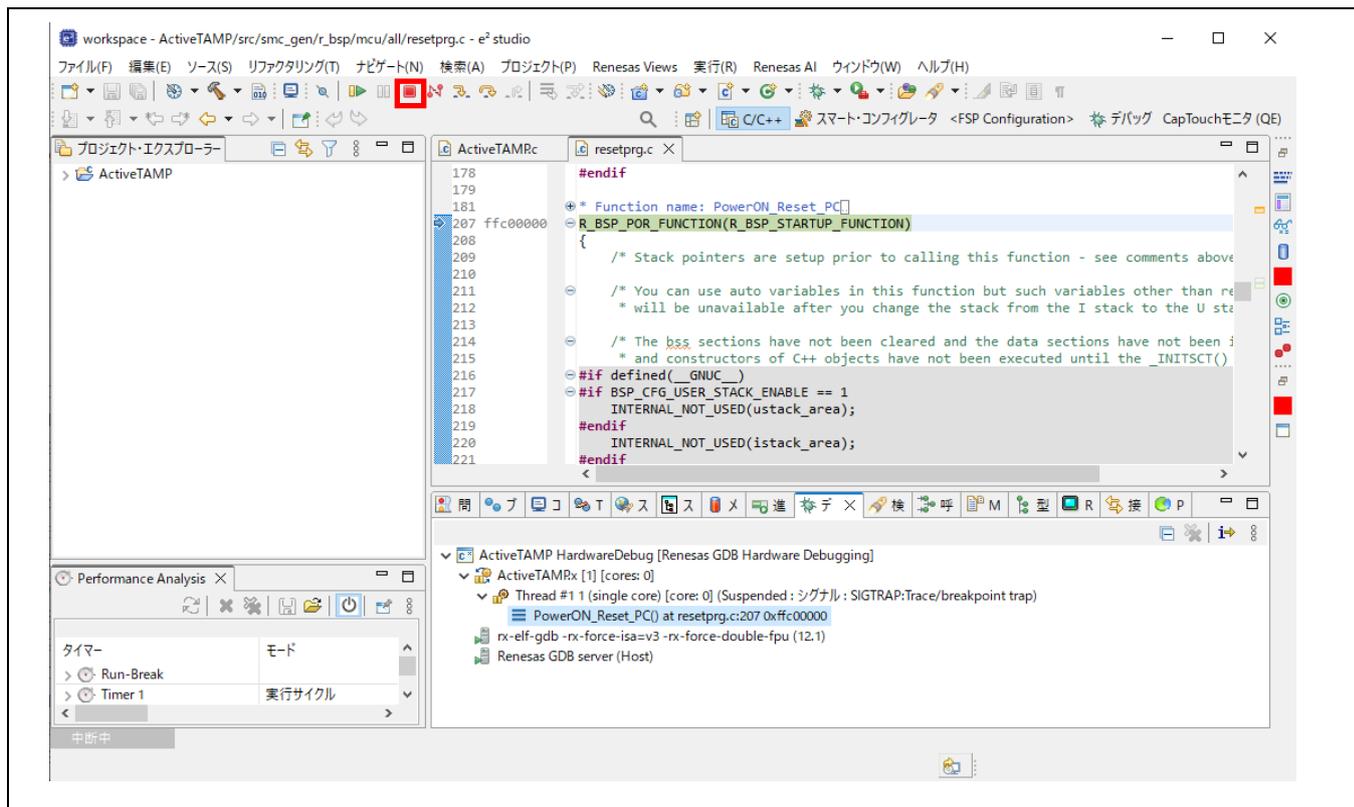


図 3-5 エミュレータの切断

- d. エミュレータリセットヘッダ(J6)を接続するとデバッグを必要としないシングルチップモードとして起動し、書き込まれたプロジェクトを実行します。

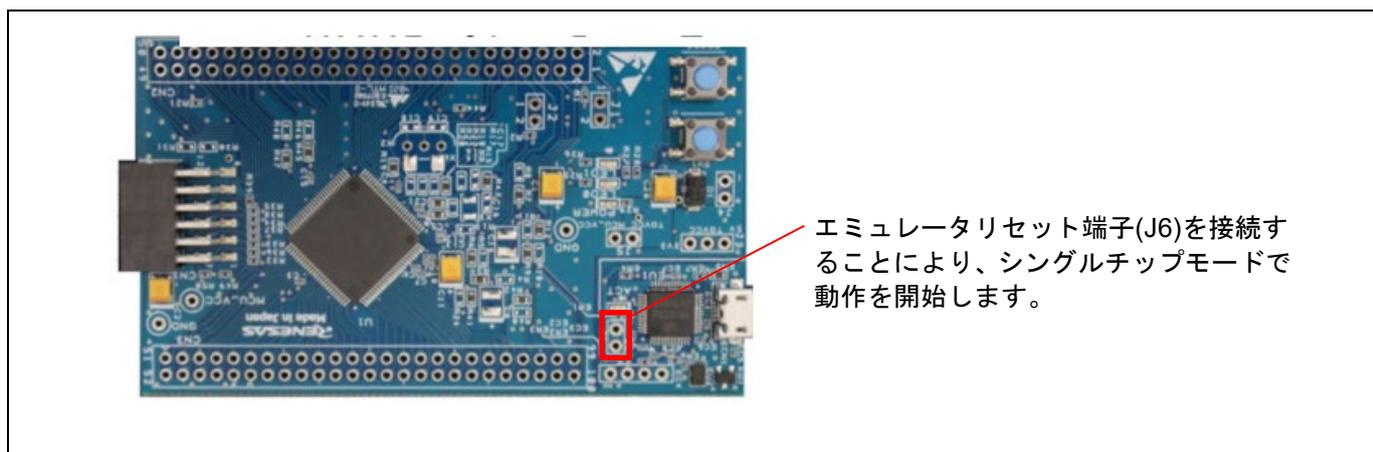


図 3-6 動作モードの切り替え

4. システムの説明

4.1 低消費電力モード中の通信方法の動作説明

本サンプルプログラムでは、外部 IO にターミナルソフトをインストールした PC を使って低消費電力モード(ソフトウェアスタンバイ)中のRX66Nグループと UART 通信を行います(ここでは CMON コマンドを発行)。

電源を投入すると、RX66Nグループは PE2 を IRQ7 にしてソフトウェアスタンバイモードへ遷移します。

外部 IO からの UART 通信(RX66Nグループには IRQ7 割り込み)により、ソフトウェアスタンバイモードから復帰します。復帰後、RX66Nグループは SCI の送受信許可、PE2 を IRQ7 から SCI12.RXD に設定し、再送要求コマンドを発行します。外部 IO は再送要求コマンド受領後、Target Board for RX66N上の LED1 を点灯させるコマンドを再発行し、LED1 を点灯させソフトウェアスタンバイモードに遷移します。

その後、RX66Nグループは LED1 を滅灯し、再度ソフトウェアスタンバイモードに遷移します。

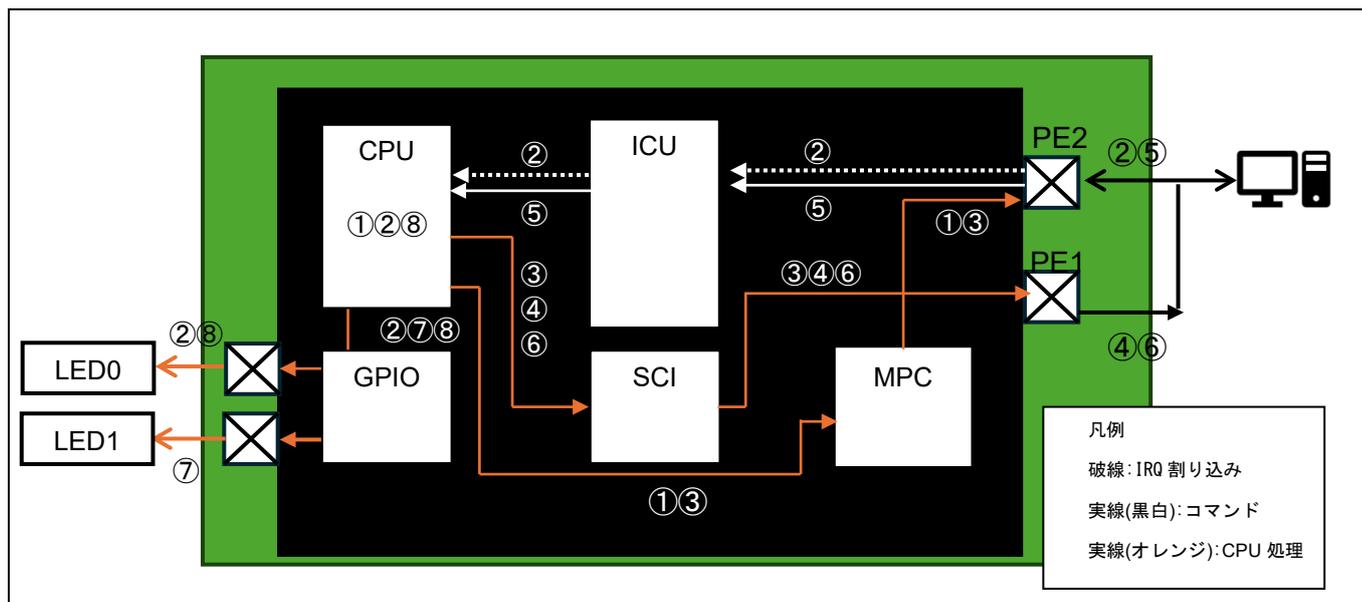


図 4-1 低消費電力モード中の通信方法の動作

①	電源投入	RX66Nグループは PE2 を IRQ7 にして、ソフトウェアスタンバイモードへ遷移します
②	LED 点灯指示	PC からの LED 点灯コマンド(CMON) を受信します。 RX66Nグループとしてこれは IRQ7 割り込みなので CPU が復帰します 復帰により通常状態となったので状態を示す LED1 を点灯します(通常状態)
③	SCI 送受信許可と PE2 端子機能変更	PC からの再送に備え、 CPU が SCI の送受信許可を設定します。 PE2 を IRQ7⇒SCI12.RXD へ変更します。
④	再送要求	RX66Nグループから PC へ再送要求コマンド(CMRE)を送信します
⑤*	LED 点灯指示再送	PC の TeraTerm から LED 点灯指示(CMON)を自動で再送します
⑥	受信完了	CPU が受信完了コマンド(CMRV)を外部 IO へ送信します
⑦	LED 点灯	CPU が LED1 を点灯します
⑧	ソフトウェアスタンバイモードへ遷移	LED1 を滅灯します PE2 を SCI12.RXD⇒IRQ7 に変更します 再びソフトウェアスタンバイモードへ遷移します

*:上記③で再送指示後、再送信されなかった場合(ここでは CMON)、RX66Nグループは外部 IO からの再送信を待ち続けます。本サンプルプログラムでは自律で再送しますが、システムで応用する場合、必ず再送信してください。

5. ソフトウェアの説明

5.1 構成

5.1.1 ファイルの構成

表 5-1に本サンプルプログラムのファイルを示します。なお、スマート・コンフィグレータのコード生成機能によって生成されたソースコードをそのまま使用しているファイルは除きます。

表 5-1 サンプルコードで使用するファイル

ファイル名	概要	備考
main.c	メイン処理	-
main.h	main.c のヘッダファイル	-
Config_SCI12_user.c	SCI12 のピンアサイン処理	スマート・コンフィグレータによって生成されたファイルにピンアサインの処理を追加
Config_ICU_user.c	IRQ7 のピンアサイン処理	スマート・コンフィグレータによって生成されたファイルにピンアサインの処理を追加

5.1.2 定数一覧

表 5-2にサンプルプログラムで使用する定数を示します。

表 5-2 定数一覧

定数名	設定値	内容
WAKEUP_DELAY	15000L	通常動作モード遷移後の発振安定待機時間
WAKEUP_DELAY_UNITS	1000000L	1us 単位での待ち時間計算値
LED_ON	0	LED 出力データ:点灯
LED_OF	1	LED 出力データ:消灯
LED0	PORT3.PODR.BIT.B2	LED0 出力データ格納ビット
LED1	PORT3.PODR.BIT.B3	LED1 出力データ格納ビット

5.1.3 変数一覧

表 5-3にサンプルプログラムで使用するグローバル変数を示します。

表 5-3 変数一覧

型	変数名	内容	使用関数
uint8_t	rcv_ok[5U]	受信完了コマンド	main
uint8_t	resend[5U]	再送要求コマンド	main
const char	ledcmd1[5U]	受信コマンド判別用データ(LED 点灯命令)	main
const char	ledcmd2[5U]	受信コマンド判別用データ(LED 滅灯命令)	main
char	cmd1[2]	コマンド識別用控え	main
char	cast_rcv[5U]	uint8_t 型から char 型への型変換に使用	main
uint8_t	rcvend	受信完了イベント発生フラグ	main

5.1.4 関数一覧

表 5-4に関数を示します。なお、スマート・コンフィグレータのコード生成機能によって生成されたソースコードをそのまま使用している関数については省略しています。

表 5-4 関数一覧

関数名	概要
main	メイン処理
lpmode	ソフトウェアスタンバイモードでの通信待機処理
r_Config_SCI12_Create_UserInit	SCI12 用の端子設定処理
r_Config_SCI12_callback_receiveerror	通信エラー時のコールバック処理
r_Config_SCI12_callback_receiveend	受信完了時のコールバック処理
r_Config_ICU_irq7_interrupt	IRQ7 の割り込み禁止処理

5.1.5 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	初期設定後、ソフトウェアスタンバイモードへの遷移を行います。 また、ホストからのコマンドによる LED1 の点灯、滅灯処理を行います。
引数	なし
リターン値	なし
備考	-

lpmode

概要	ソフトウェアスタンバイモードでの通信待機処理
ヘッダ	main.h
宣言	void lpmode(void)
説明	PE2 を IRQ7 にピンアサインし、ソフトウェアスタンバイモードへ遷移します。 このとき、PC からの通信により CPU は起動します。 また、CPU 復帰後は、PE2 を SCI12 の RXD12 へ変更、再送要求コマンドの送信、 再送されてくるコマンドの待機処理を行います。
引数	なし
リターン値	なし
備考	-

r_Config_SCI12_Create_UserInit

概要	SCI12 用の端子設定処理
ヘッダ	Config_SCI12.h
宣言	void R_Config_SCI12_Create_UserInit(void)
説明	PE2 端子を SCI12 用にピンアサインを変更し、SCI12 の通信を開始できる状態に します。
引数	なし
リターン値	なし
備考	-

r_Config_SCI12_callback_receiveerror

概要	通信エラー時のコールバック処理
ヘッダ	Config_SCI12.h
宣言	static void r_Config_SCI12_callback_receiveerror(void);
説明	通信エラー発生時、再送要求コマンドの送信処理を行います。
引数	なし
リターン値	なし
備考	-

r_Config_SCI12_callback_receiveend

概要	受信完了時のコールバック処理
ヘッダ	Config_SCI12.h
宣言	static void r_Config_SCI12_callback_receiveend(void);
説明	受信完了時、受信完了フラグを立て、受信完了コマンドの送信処理を行います。
引数	なし
リターン値	なし
備考	-

r_Config_ICU_irq7_interrupt

概要	IRQ7用の割り込み禁止処理
ヘッダ	なし
宣言	static void r_Config_ICU_irq7_interrupt(void)
説明	ソフトウェアスタンバイモードからCPUを起動させるための割り込み処理です。 本処理内では、本割り込み(IRQ7)の禁止処理を行います。
引数	なし
リターン値	なし
備考	-

5.2 フローチャート

ここでは本サンプルプログラムの各関数の動作をフローチャートで説明します。

なお、図 5-1内で示される「各モジュールの初期化」はスマート・コンフィグレータにより、R_Config_<モジュール名称>_Create関数に生成されます(例外:TSIPはR_TSIP_Open関数に生成)。また、R_Config_<モジュール名称>_Create_UserInit関数はR_Config_<モジュール名称>_Create関数によりコールされます。

5.2.1 メイン関数

メイン関数のフローを図 5-1に示します。

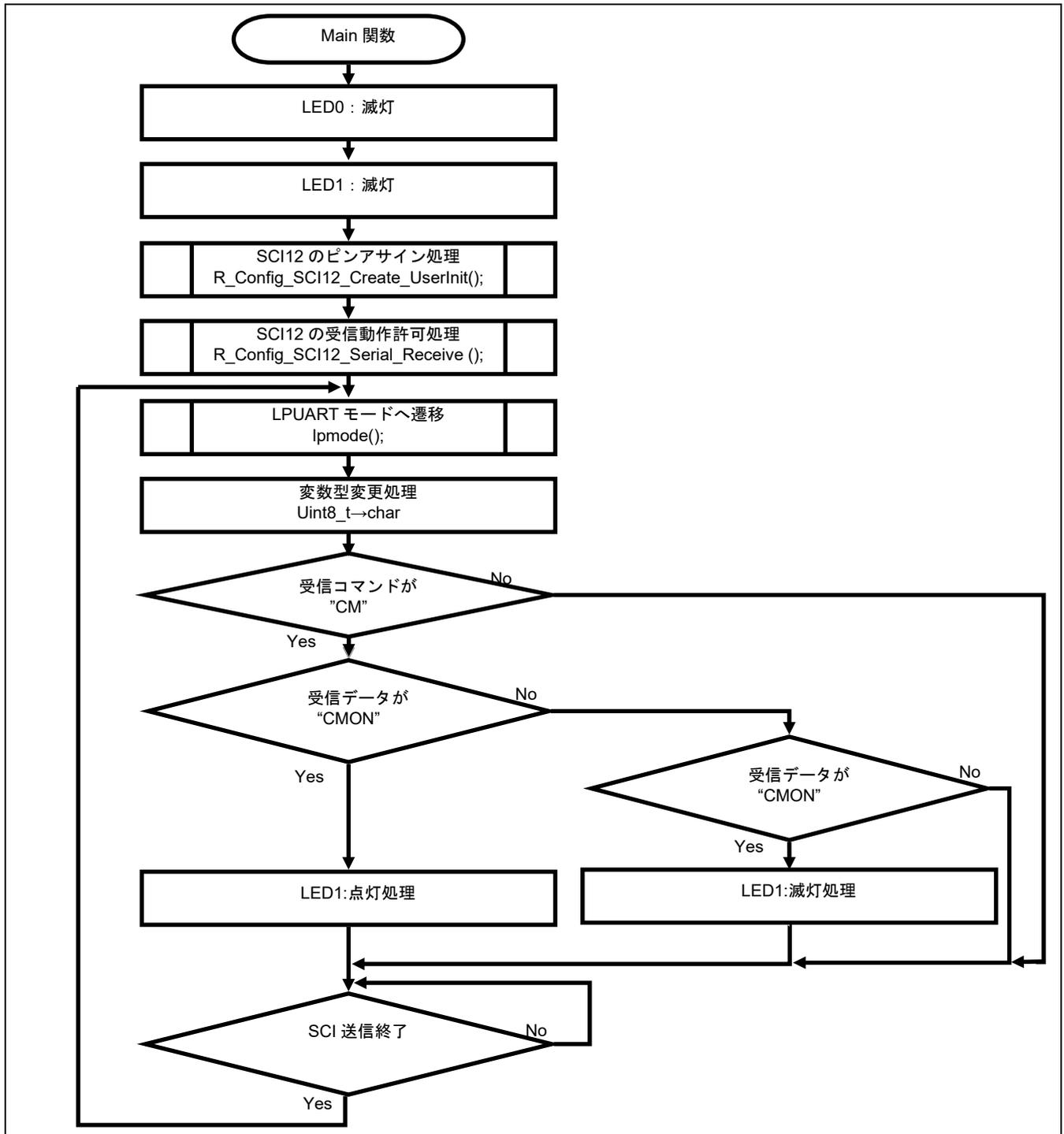


図 5-1 メイン関数 フロー

5.2.2 Lpmode 関数

Lpmode 関数のフローを図 5-2に示します。

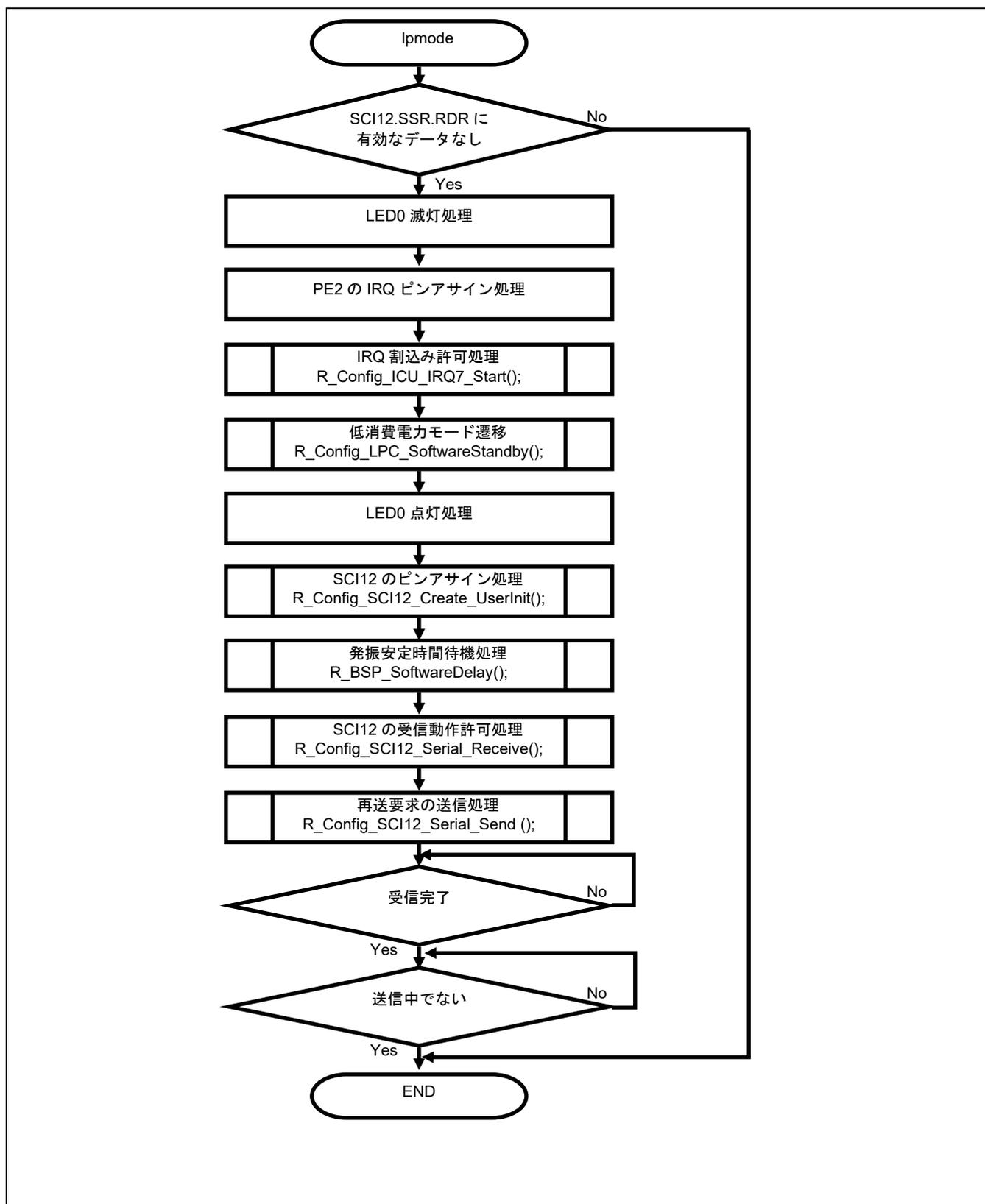


図 5-2 lpmode 関数のフロー

5.2.3 R_Config_SCI12_Create_UserInit 関数

R_Config_SCI12_Create_UserInit のフローを図 5-3に示します。

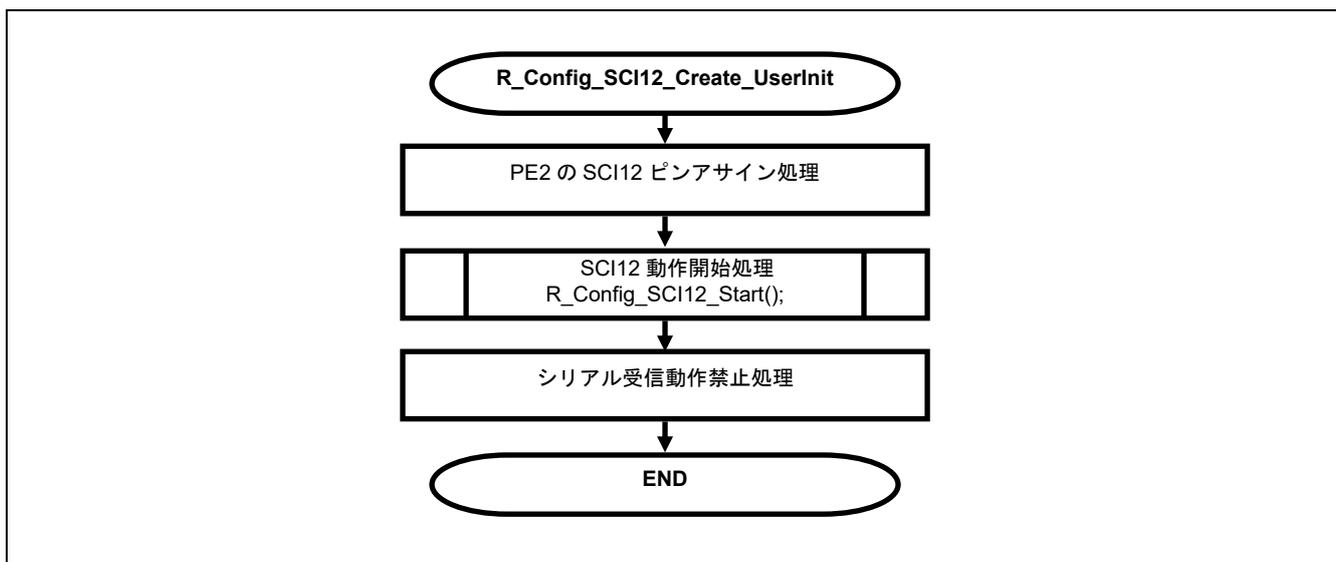


図 5-3 R_Config_SCI12_Create_UserInit のフロー

5.2.4 r_Config_ICU_irq7_interrupt 関数

r_Config_ICU_irq7_interrupt のフローを図 5-4に示します。

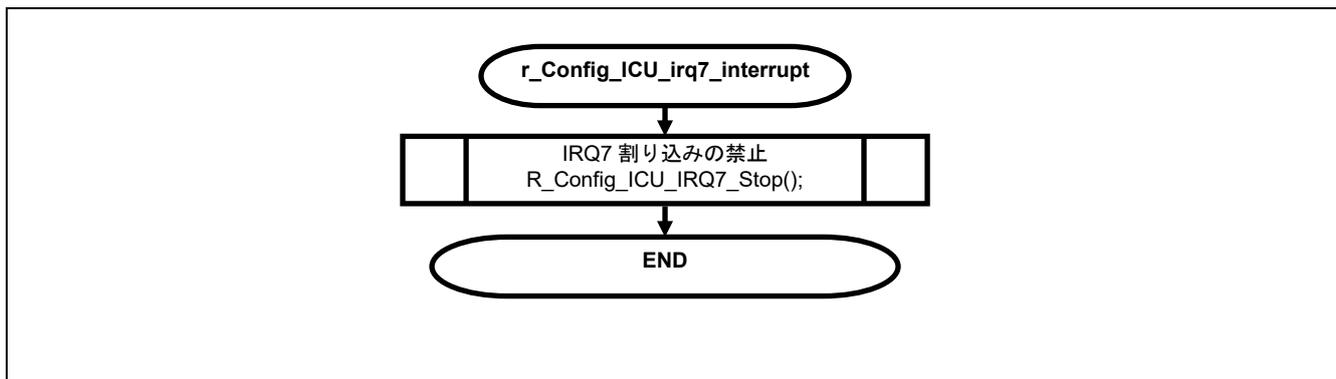


図 5-4 r_Config_ICU_irq7_interrupt のフロー

5.2.5 r_Config_SCI12_callback_receiveerror 関数

r_Config_SCI12_callback_receiveerro のフローを図 5-5に示します。

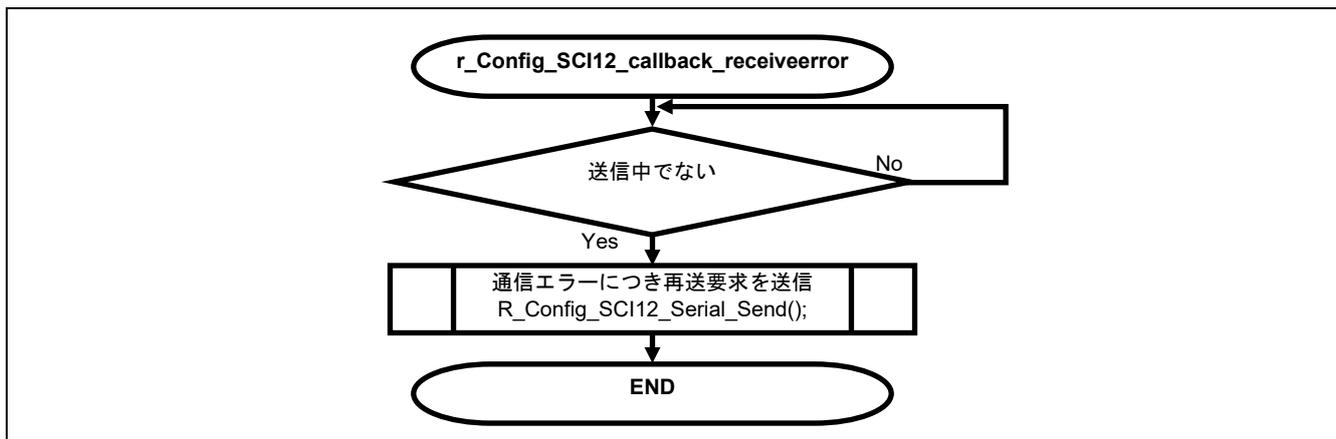


図 5-5 r_Config_SCI12_callback_receiveerror のフロー

5.2.6 r_Config_SCI12_callback_receiveend 関数

r_Config_SCI12_callback_receiveend のフローを図 5-6に示します。

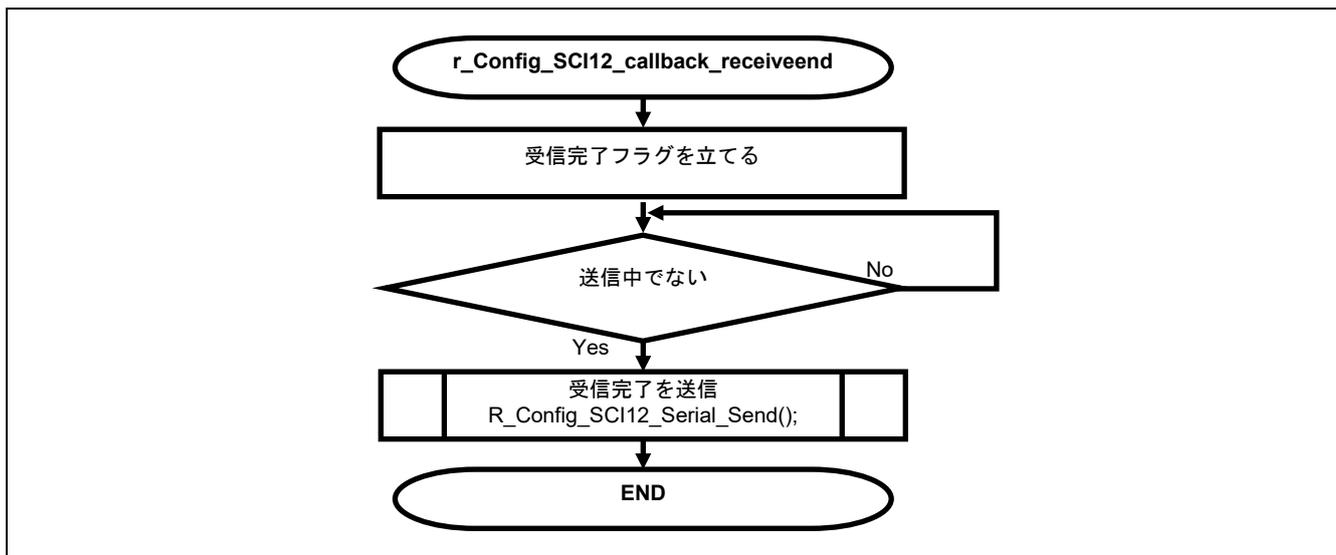


図 5-6 r_Config_SCI12_callback_receiveend のフロー

6. 参考ドキュメント

[RX66Nグループ ユーザーズマニュアル ハードウェア編 \(renesas.com\)](#)

[Target Board for RX66N ユーザーズマニュアル Rev.1.00 \(renesas.com\)](#)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2025.1.10	-	初版発行

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、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「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。

5. クロックについて

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。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

8. 製品間の相違について

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、フラッシュメモリ、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、ノイズ幅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。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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。
6. 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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因またはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。
 10. 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。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. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
 13. 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。
 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 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が直接的、間接的に支配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。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/