RL78/G23

ハードウェア・フロー制御に対応した UART 通信

要旨

本アプリケーションノートでは、ハードウェア・フロー制御に対応した UART 通信の実現方法を説明します。

動作確認デバイス

RL78/G23

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

本アプリケーションノートでは、ハードウェア・フロー制御に対応した UART 通信を行います。ハードウェア・フロー制御では、UART 送受信の 1 バイトごとにポートを使用して RTS と CTS の信号を制御します。

1.1 ハードウェア・フロー制御の仕様

ハードウェア・フロー制御に対応した UART 通信の構成を図 1-1 に示します。また、使用する端子と用途を表 1-1 に示します。

- RTS の制御

RTS の制御は、UART を使用したデータ受信の 1 バイトごとに実施します。

データ受信を開始したタイミングで、RTS をハイ・レベルに設定します。これにより、対向デバイスに「現在データを受信できない」ことを知らせます。データ受信の開始は、キー割り込みを使用して検出します。

1 バイトの受信が完了した後、受信データを読み出してから RTS をロウ・レベルに戻します。これで、対向デバイスに「データを受信可能」であることを通知します。

- CTS の確認

UART でデータを送信する際、CTS のレベルは 1 バイトごとに確認します。

CTS がロウ・レベルの場合、対向機器がデータを受信可能と判断し、データを送信します。

CTS がハイ・レベルの場合、対向機器がデータを受信できない状態と判断し、データ送信を一時停止します。外部割り込みを使用して CTS がロウ・レベルになったことを検出した後、データを送信します。

図 1-1 フロー制御の構成図

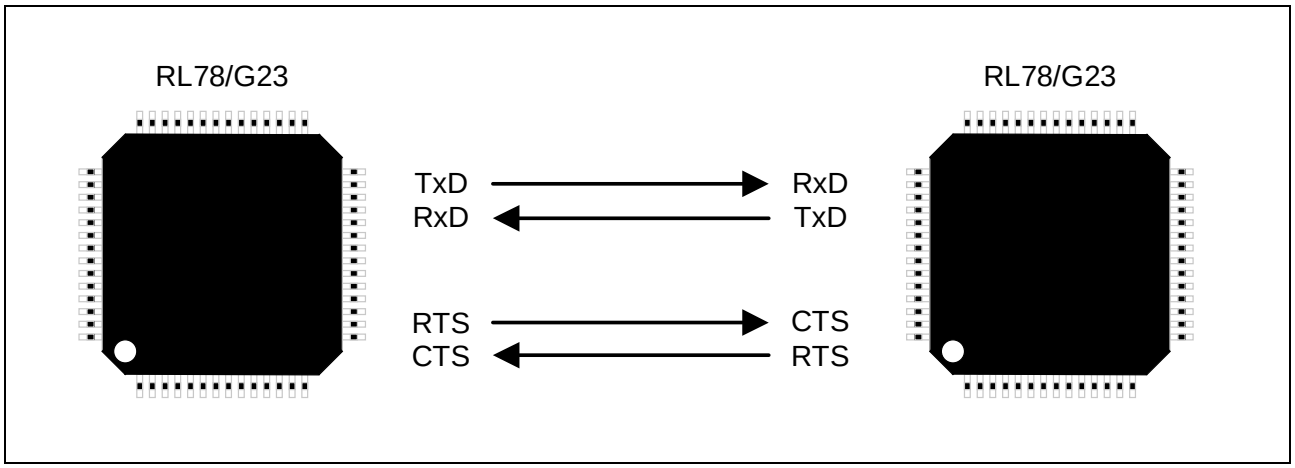


表 1-1 フロー制御に使用する端子と用途

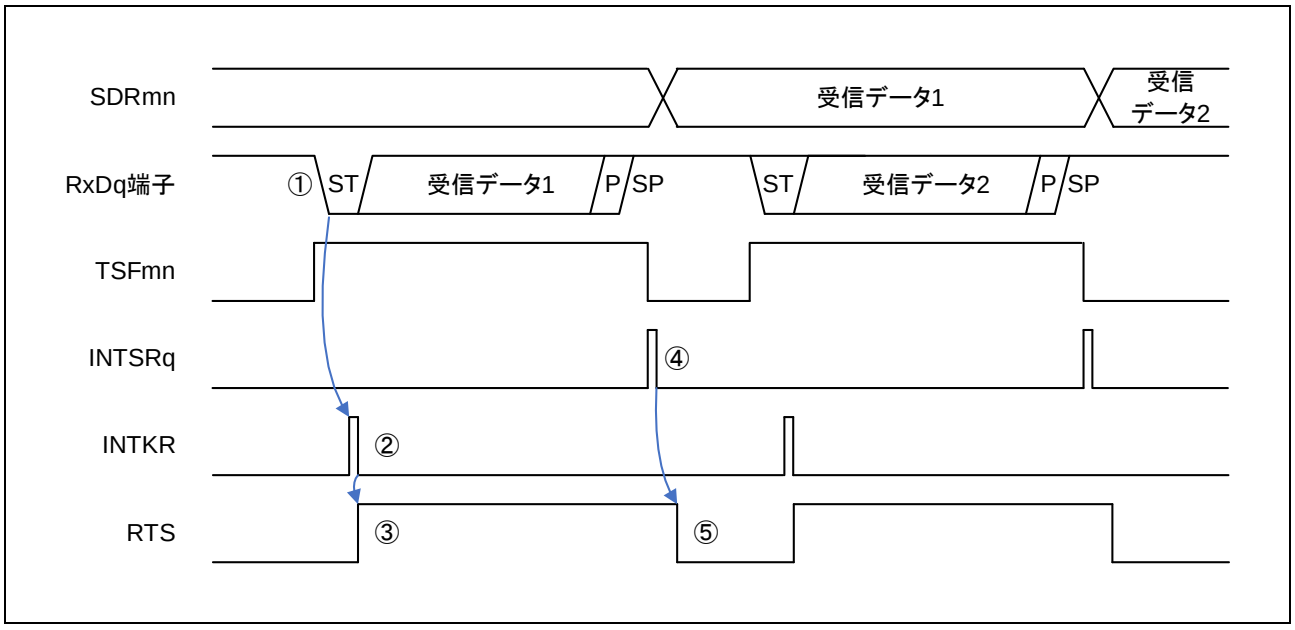
項目	端子	用途
TxD	P77/TS09/KR7/INTP11 /(**TxD2**)	UART のデータ送信に使用します。
RxD	P76/TS08/**KR6**/INTP10 /(**RxD2**)	UART のデータ受信に使用します。 キー割り込みを使用し、データ受信時にスタート・ビットの立ち下がりを検出します。
RTS	**P75**/TS07/KR5/INTP9 /SCK01/SCL01	対向デバイスへ UART 受信の状態を通知します。 ロウ・レベル：受信可能 ハイ・レベル：受信不可能
CTS	**P31**/TS01/EI31/TI03/TO03 /**INTP4**/(PCLBUZ0)	対向デバイスから UART 受信の状態を入力します。 外部割り込みを使用し、CTS がロウ・レベルに変化したことを検出します。

- UART 受信

UART 受信のタイミングチャートを図 1-2 に示します。

UART 受信では、RxDq 端子に割り当てられたキー割り込み機能を使用して、受信データのスタート・ビットを検出します（①, ②）。スタート・ビットを検出すると、RTS をハイ・レベルにします（③）。データ受信が完了すると、INTSRq 割り込みが発生します（④）。この割り込み処理で受信データを SDRmn レジスタから読み出します。その後、RTS をロウ・レベルにします（⑤）。

図 1-2 データ受信時のタイミングチャート



- UART 送信

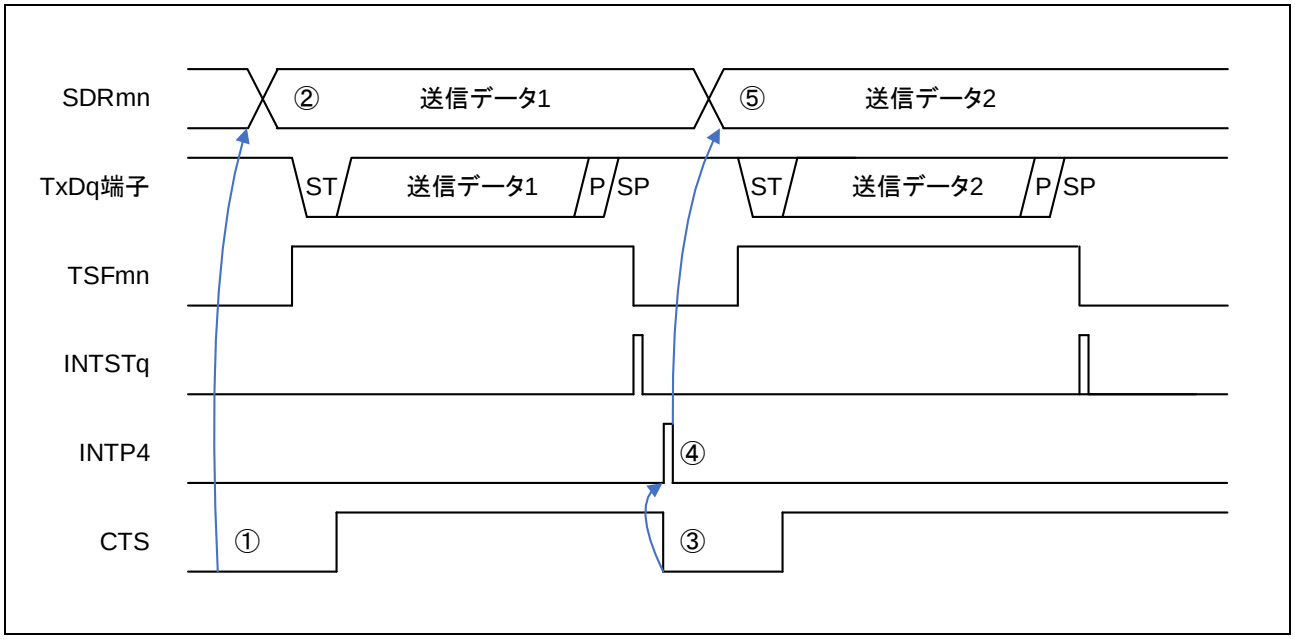
UART 送信のタイミングチャートを図 1-3 に示します。

UART 送信では、まず CTS のレベルを確認します（①）。

CTS がロウ・レベルの場合、送信するデータを SDRmn レジスタに格納し、送信を開始します（②）。

CTS がハイ・レベルの場合、CTS に割り当てられた外部割り込み（INTP4）を動作許可します。CTS がロウ・レベルになると、INTP4 割り込みが発生します（③，④）。割り込みが発生した後、送信データを SDRmn レジスタに格納し、送信を開始します（⑤）。

図 1-3 UART 送信のタイミングチャート



1.2 通信の仕様

本アプリケーションノートでは、図 1-4 に示すデータのフォーマットにしたがって、UART 通信を行います。また、UART 通信の設定を表に示します。

表 1-2 UART 通信設定

項目	設定
データ・ビット長[bit]	8
データ転送順序	LSBファースト
パリティ	偶数パリティ
転送レート[bps]	1000000

図 1-4 データのフォーマット

STX	LEN	データ（可変長）	SUM	ETX
(1 バイト)	（1 バイト）	（最大 255 バイト）	（1 バイト）	（1 バイト）

記号	値	内容
STX	02H	フレームのヘッダ
LEN	-	データ長情報
SUM	-	フレーム内のチェックサム・データ 初期値 00H から計算対象すべてのデータを 1 バイトごとに減算した値（ボローは無視） 計算対象：LEN+データすべて
ETX	03H	フレームのフッタ

フレーム内のチェックサム（SUM）の計算例を次に示します。

次のフレームの場合、チェックサムの計算対象は LEN から D4 です。

SUM の値は、00H（初期値）– 04H – FFH – 80H – 40H – 22H = 1BH（ボロー無視。下位 8 ビットのみ）になります。

STX	LEN	D1	D2	D3	D4	SUM	ETX
02H	04H	FFH	80H	40H	22H	Checksum	03H

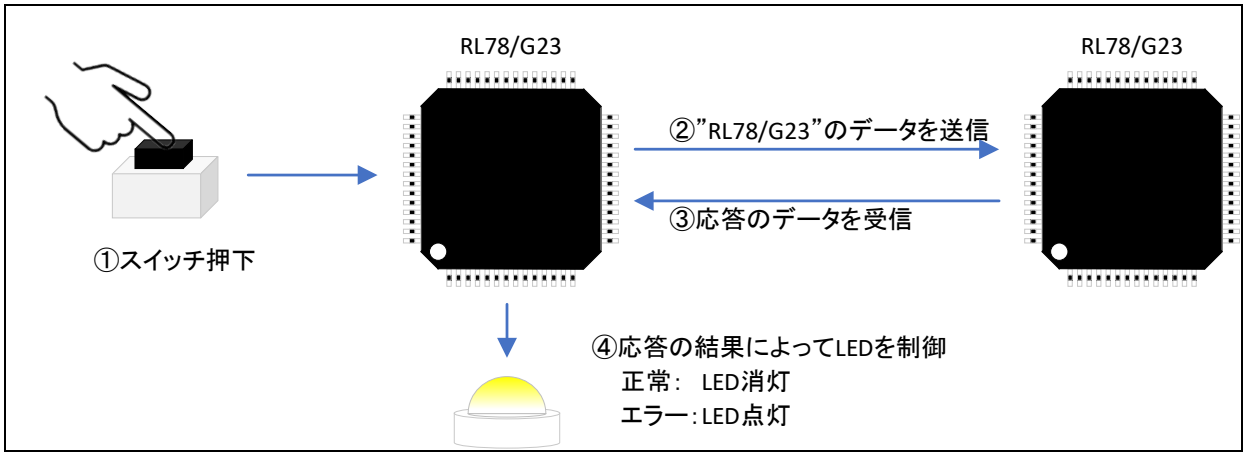
1.3 仕様詳細

本サンプルコードの状態遷移図を図 1-5 に示します。

本サンプルコードでは、初期設定完了後に、スイッチの押下を待ちます。スイッチが押下されると、UART でデータの送信を開始します。すべてのデータを送信すると、対向機器からの応答を待ちます。応答データが正常終了の場合は、再度、スイッチの押下を待ちます。応答データが異常終了またはチェックサムエラー発生の場合は、LED を点灯します。

また、対向機器には本サンプルコードを使用できます。データを受信すると、受信の結果に従って、応答のデータを送信します。

図 1-5 状態遷移図



(1) ポートの初期設定を行います。

＜ポート設定条件＞

- LED2 を制御する P52 をハイ・レベル出力に設定します。
- LED1 を制御する P53 をハイ・レベル出力に設定します。
- RTS を制御する P75 をハイ・レベル出力に設定します。
- CTS を制御する P31 を入力に設定します。

(2) UART2 の初期設定を行います。

＜UART2 設定条件＞

- SAU1 チャネル 0,1 を UART2 として使用します。
- データ入力は P76/RxD2 端子、データ出力は P77/TxD2 端子を使用します。
- 転送モードはシングル転送モードを使用します。
- データ長は 8 ビットを使用します。
- パリティは偶数パリティを使用します。
- データ転送順設定は LSB ファーストを使用します。
- 転送レートは 1Mbps を使用します。

(3) 外部割り込みの初期設定を行います。

＜外部割り込み設定条件＞

- INTP0 を立ち下がりエッジ検出として使用します。
- INTP4 を立ち下がりエッジ検出として使用します。

(4) キー割り込みの初期設定を行います。

＜キー割り込み設定条件＞

- KR6 を立ち下がりエッジ検出として使用します。

(5) 初期設定が完了したら、以下に示す手順で対向機器との通信動作を行います。

① RTS をロウ・レベルに設定します。

② スイッチの押下、または UART によるデータ受信を HALT モードで待ちます。

送信処理（③~⑨）

③ スイッチが押下されると、INTP0 割り込みが発生し、INTP0 割り込み処理を実行します。

INTP0 割り込み処理では、データ送信中にスイッチが無効化されるよう INTP0 割り込みを停止します。送信するデータをフォーマット形式に合わせて送信バッファに格納します。

④ CTS を確認します。

CTS がロウ・レベルの場合、データを送信します（⑥へ移行）。

CTS がハイ・レベルの場合、データ送信を保留し、CTS がロウ・レベルになるのを検知するため INTP4 割り込みを有効にします。

⑤ CTS の変化を検出します。

CTS がロウ・レベルに変化すると、INTP4 割り込みが発生し、INTP4 割り込み処理内で INTP4 割り込みを停止します。その後、データを送信します。

⑥ データ送信完了を HALT モードで待ちます。

⑦ すべてのデータが送信完了するまで、手順④～⑥を繰り返します。

⑧ データ送信が完了したら、INTP0 割り込みを許可してスイッチを有効にします。

⑨ 対向機器からの応答データ受信を HALT モードで待ちます。

受信処理（⑩~⑭）

⑩ 対向機器からのデータ受信を開始すると、スタート・ビットの立ち下がりでキー割り込みが発生します。

⑪ キー割り込み処理で RTS をハイ・レベルに設定します。次に、1 バイトの受信が完了するまでキー割り込みを禁止にします。

⑫ 受信完了割り込みが発生するとデータを読み取り、以下のステータスに基づいて処理を実行します。

- 受信待機中：

  受信データと STX の一致を確認します。一致の場合、受信ステータスを STX 受信完了に設定します。不一致の場合、受信ステータスをエラー発生に設定します。

- STX 受信完了：

  受信データをデータ長の値として変数に格納します。また、チェックサムの計算を開始します。

- データ受信中：

  受信データを受信バッファに格納します。次にチェックサムを計算します。受信データ数とデータ長が一致した場合、受信ステータスをデータ受信完了に設定します。

- データ受信完了：

  受信データと計算したチェックサム値の一致を確認します。一致の場合、受信ステータスをチェックサム受信完了に設定します。不一致の場合、エラー発生に設定します。

- チェックサム受信完了：

  受信データと ETX の一致を確認します。一致の場合、受信ステータスを ETX 受信完了に設定し、応答データに ACK を設定します。不一致の場合、受信ステータスをエラー発生に設定し、応答データに NACK を設定します。

⑬ 受信ステータスがエラー発生、またはすべてのデータ受信完了するまで⑨~⑫を繰り返します。

⑭ エラー発生や応答データにしたがって処理を実行します。

- 受信ステータスがエラー発生の場合、LED1 を点灯し、処理を終了します。
- 応答データが正常終了だった場合、LED2 を点灯し、処理を終了します。
- 応答データが異常終了だった場合、初期化を行い、②へ移行します。

また、②で対向機器からのデータ受信を開始した場合は、⑩~⑬の受信処理を実行します。次に、送信する応答データをフォーマット形式に合わせて送信バッファに格納します。その後、応答データを送信するため、④~⑦の送信処理を実行します。

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では、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

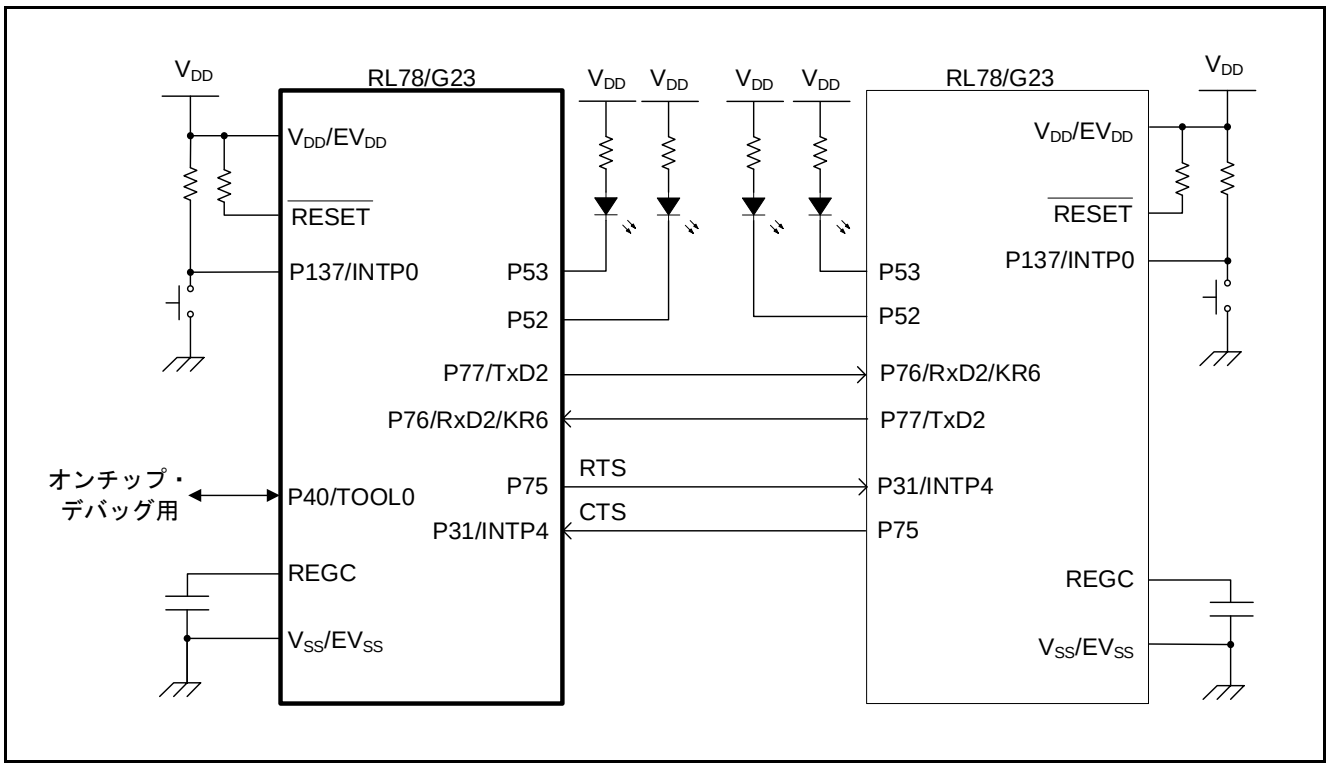
項目	内容
使用マイコン	RL78/G23（R7F100GLG）
動作周波数	● 高速オンチップ・オシレータ（HOCO）クロック： 32MHz ● CPU／周辺ハードウエア・クロック： 32MHz
動作電圧	5.0V（1.8V～5.5V で動作可能） LVD0 動作（VLVD0）：リセット・モード 立ち上がり時 TYP. 1.90V (1.84 V ～ 1.95 V) 立ち下がり時 TYP. 1.86V (1.80 V ～ 1.91 V)
統合開発環境(CS+)	ルネサス エレクトロニクス製 CS+ for CC V8.11.00
コンパイラ(CS+)	ルネサス エレクトロニクス製 CC-RL V1.13.00
統合開発環境(e^2 studio)	ルネサス エレクトロニクス製 e^2 studio 2024-07 (24.7.0)
コンパイラ(e^2 studio)	ルネサス エレクトロニクス製 CC-RL V1.13.00
統合開発環境(IAR)	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 V5.10.3
コンパイラ(IAR)	IAR Systems 製 IAR C/C++ Compiler for Renesas RL78 V5.10.3.2716
スマート・コンフィグレータ	V.1.10.0
ボードサポートパッケージ（r_bsp）	V.1.62
使用ボード	RL78/G23-64p Fast Prototyping Board（RTK7RLG230CLG000BJ）

3. ハードウェア説明

3.1 ハードウェア構成例

図 3-1 に本アプリケーションノートで使用するハードウェア構成例を示します。

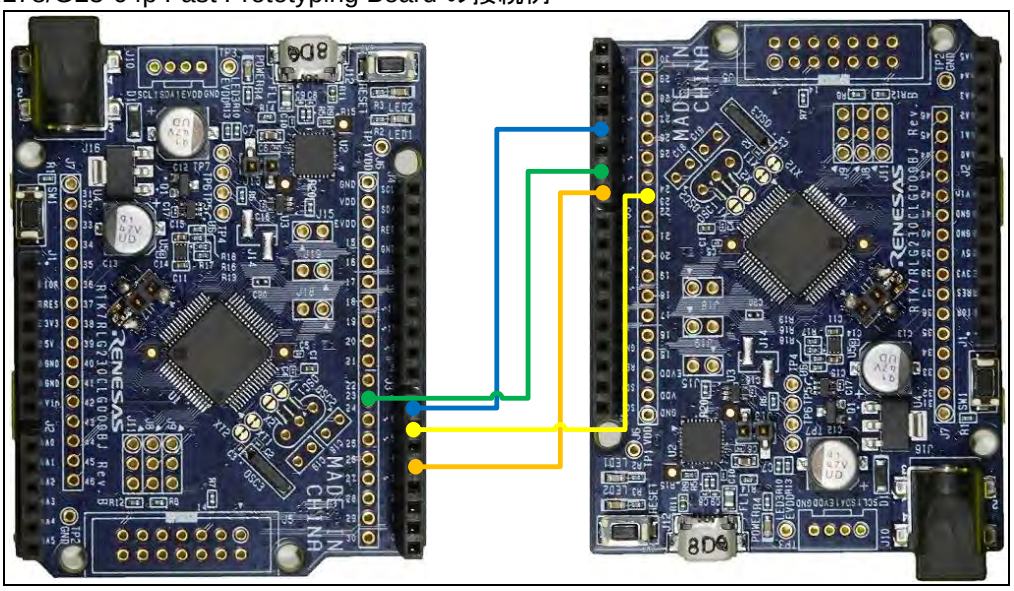
図 3-1 ハードウェア構成例



注意 1 この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続して下さい）。

2 VDD は LVD にて設定したリセット解除電圧（VLVD）以上にしてください。

図 3-2 RL78/G23-64p Fast Prototyping Board の接続例



3.2 使用端子一覧

表 3-1 に使用端子と機能を示します。

表 3-1 使用端子と機能

端子名	入出力	内容
P31/TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	入力	CTS 信号入力
P52/(INTP10)	出力	LED2 制御信号出力
P53/(INTP11)	出力	LED1 制御信号出力
P75/TS07/KR5/INTP9/SCK01/SCL01	出力	RTS 信号出力
P76/TS08/KR6/INTP10/(RxD2)	入力	データ受信
P77/TS09/KR7/INTP11/(TxD2)	出力	データ送信
P137/EI137/INTP0	入力	スイッチ信号入力

注意 本アプリケーションノートは、使用端子のみを端子処理しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

4. ソフトウェア説明

4.1 オプション・バイトの設定一覧

表 4-1 にオプション・バイト設定を示します。

表 4-1 オプション・バイト設定

アドレス	設定値	内容
000C0H	1110 1111B (EFH)	ウォッチドッグ・タイマ 動作停止 （リセット解除後、カウント停止）
000C1H	1111 1110B (FEH)	LVD リセット・モード 検出電圧： 立ち上がり時 TYP. 1.90V (1.84 V ～ 1.95 V) 立ち下がり時 TYP. 1.86V (1.80 V ～ 1.91 V)
000C2H	11101000B (E8H)	HS モード、HOCO : 32MHz
000C3H	10000100B (84H)	オンチップ・デバッグ許可

4.2 定数一覧

表 4-2 にサンプルコードで使用する定数を示します。

表 4-2 サンプルコードで使用する定数

定数名	定義場所	設定値	内容
WAIT	r_cg_userdefine.h	3	受信時、スタート・ビット検出から TSF フラグ判定までの待ち時間の設定値
CTS_HIGH	r_cg_userdefine.h	1	CTS 信号のレベル：ハイ・レベル
CTS_LOW	r_cg_userdefine.h	0	CTS 信号のレベル：ロウ・レベル
RTS_HIGH	r_cg_userdefine.h	1	RTS 信号のレベル：ハイ・レベル
RTS_LOW	r_cg_userdefine.h	0	RTS 信号のレベル：ロウ・レベル
LED_ON	r_cg_userdefine.h	0	LED の点灯値
SEND_START	r_cg_userdefine.h	1	送信ステータス：送信開始
SENDING	r_cg_userdefine.h	2	送信ステータス：送信中
SEND_END	r_cg_userdefine.h	3	送信ステータス：送信完了
RECEIVING_WAIT	r_cg_userdefine.h	0	受信ステータス：受信待機中
STX_RECEIVED	r_cg_userdefine.h	1	受信ステータス：STX 受信完了
RECEIVING_DATA	r_cg_userdefine.h	2	受信ステータス：データ受信中
DATA_RECEIVED	r_cg_userdefine.h	3	受信ステータス：データ受信完了
SUM_RECEIVED	r_cg_userdefine.h	4	受信ステータス：チェックサム受信完了
ETX_RECEIVED	r_cg_userdefine.h	5	受信ステータス：ETX 受信完了
RECEIVING_ERROR	r_cg_userdefine.h	6	受信ステータス：エラー発生
ACK	r_cg_userdefine.h	0x06	受信後の応答：正常終了
SUM_ERROR	r_cg_userdefine.h	0x07	受信後の応答：チェックサムエラー発生
NACK	r_cg_userdefine.h	0x15	受信後の応答：異常終了

4.3 変数一覧

表 4-3 にグローバル変数を示します。

表 4-3 サンプルコードで使用するグローバル変数

Type	Variable Name	Contents	Function Used
uint8_t	g_send_data[]	送信データ	main.c, Config_INTC_user.c, Config_UART2_user.c
uint8_t	g_receive_buffer[]	受信データバッファ	main.c, Config_UART2_user.c
uint8_t	g_send_status	送信ステータス	main.c, Config_INTC_user.c , Config_UART2_user.c
uint8_t	g_receive_status	受信ステータス	main.c, Config_UART2_user.c
uint8_t	g_receive_length	受信したデータのデータ長	main.c, Config_UART2_user.c
uint8_t	g_receive_checksum	受信データのチェックサム値	Config_UART2_user.c
uint8_t	g_receive_count	受信データ数	Config_UART2_user.c
uint8_t	g_receive_response	受信後の応答データ	main.c, Config_UART2_user.c
uint8_t	g_cmd_send_buffer[]	送信データバッファ	Config_UART2_user.c

4.4 関数一覧

表 4-4 に関数一覧を示します。

表 4-4 関数一覧

関数名	概要	ソースファイル
main	メイン処理	main.c
r_main_user_init	初期設定関数	main.c
r_Config_INTC_intp0_interrupt	INTP0 割り込み関数	Config_INTC_user.c
r_Config_INTC_intp4_interrupt	INTP4 割り込み関数	Config_INTC_user.c
r_Config_KR_interrupt	キー割り込み関数	Config_KR_user.c
r_Config_UART2_interrupt_send	UART2 送信完了割り込み関数	Config_UART2_user.c
r_Config_UART2_interrupt_receive	UART2 受信完了割り込み関数	Config_UART2_user.c
r_send_protocol	送信データ格納関数	Config_UART2_user.c
r_uart_send_control	送信制御関数	Config_UART2_user.c
r_uart_receive_control	受信制御関数	Config_UART2_user.c

4.5 関数仕様一覧

サンプルコードの関数仕様を示します。

[関数名] main

概要	メイン処理
ヘッダ	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, Config_KR.h, r_cg_userdefine.h
宣言	void main(void);
説明	初期設定後に HALT モードへ移行します。各割り込みによって HALT モードを解除します。データ受信の状態によって下記を実行します。 ・受信エラーが発生した場合、LED1 を点灯します。 ・受信した応答データが異常終了だった場合、LED2 を点灯します。 ・受信したデータが応答データ以外だった場合（データ長が 2 以上）、対向機器への応答データの送信を開始します。
引数	• なし
リターン値	• なし
備考	なし

[関数名] r_main_user_init

概要	初期設定関数
ヘッダ	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, Config_KR.h, r_cg_userdefine.h
宣言	static void r_main_user_init(void);
説明	スイッチの入力（INTP0）を有効にします。UART 受信を動作許可します。
引数	• なし
リターン値	• なし
備考	なし

[関数名] r_Config_INTC_intp0_interrupt

概要	INTP0 割り込み関数
ヘッダ	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, r_cg_userdefine.h
宣言	#pragma interrupt r_Config_INTC_intp0_interrupt(vect=INTP0)
説明	チャタリング除去のためウエイトします。その後、データ送信を開始します。
引数	• なし
リターン値	• なし
備考	なし

[関数名] r_Config_INTC_intp4_interrupt

概要	INTP4 割り込み関数
ヘッダ	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, r_cg_userdefine.h
宣言	#pragma interrupt r_Config_INTC_intp4_interrupt(vect=INTP4)
説明	データ送信を実施します。
引数	• なし
リターン値	• なし
備考	なし

[関数名] r_Config_KR_interrupt

概要	キー割り込み関数
ヘッダ	r_cg_macrodriver.h, Config_KR.h, r_cg_userdefine.h
宣言	#pragma interrupt r_Config_KR_interrupt(vect=INTKR)
説明	UART 受信時のスタート・ビットの検出を行います。 ウエイト後、UART の通信状態を確認します。通信中であれば、RTS 信号をハイ・レベルにします。
引数	• なし
リターン値	• なし
備考	なし

[関数名] r_Config_UART2_interrupt_send

概要	UART2 送信完了割り込み関数
ヘッダ	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
宣言	#pragma interrupt r_Config_UART2_interrupt_send(vect=INTST2)
説明	送信制御関数を実行します。
引数	• なし
リターン値	• なし
備考	スマート・コンフィグレータで生成したコードを変更しています。

[関数名] r_Config_UART2_interrupt_receive

概要	UART2 受信完了割り込み関数
ヘッダ	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
宣言	#pragma interrupt r_Config_UART2_interrupt_receive(vect=INTSR2)
説明	受信制御関数を実行します。
引数	• なし
リターン値	• なし
備考	スマート・コンフィグレータで生成したコードを変更しています。

[関数名] r_send_protocol

概要	送信データ格納関数
ヘッダ	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
宣言	MD_STATUS r_send_protocol(const uint8_t __far * data, const uint16_t data_len);
説明	送信フォーマットに合わせて STX，データ数，送信データ，チェックサム，ETX を送信バッファに格納します。
引数	const uint8_t __far * data: データの開始アドレス const uint16_t data_len: データの長さ
リターン値	81H：異常終了 00H：正常終了
備考	なし

[関数名] r_uart_send_control

概要	送信制御関数
ヘッダ	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
宣言	void r_uart_send_control(void);
説明	CTS のレベルを確認します。CTS がハイ・レベルの場合、CTS の変化を検出するため、外部割り込み（INTP4）割り込みを許可します。 CTS がロウ・レベルの場合、データ送信を実行します。
引数	• なし
リターン値	• なし
備考	なし

[関数名] r_uart_receive_control

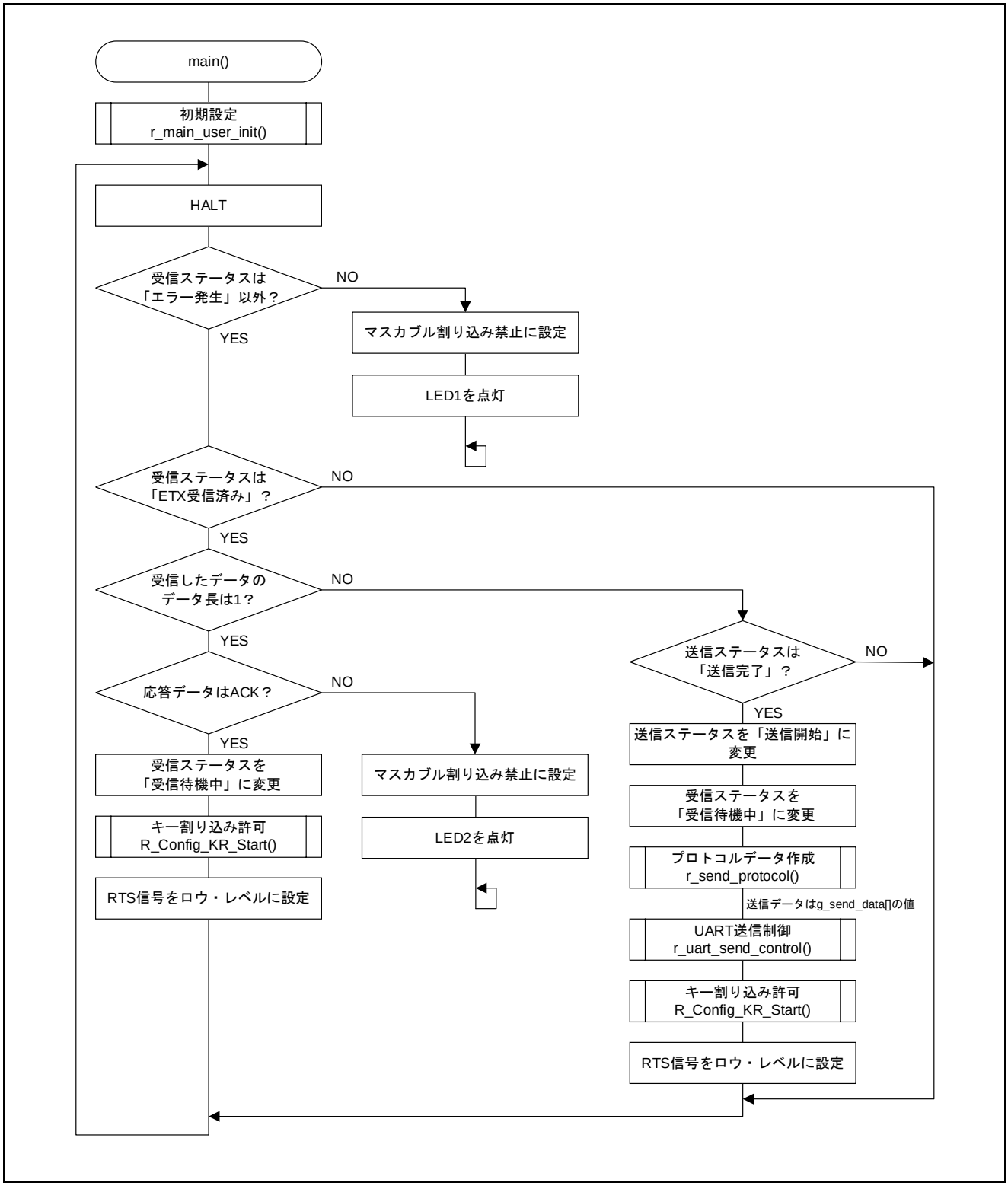
概要	受信制御関数
ヘッダ	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
宣言	void r_uart_receive_control (void);
説明	受信データを読出します。その後、下記のように受信ステータスに応じて処理を実行します。 ・RECEIVING_WAIT 受信データと STX の一致を判定します。 ・STX_RECEIVED チェックサムの計算を開始します。 ・RECEIVING_DATA 受信データの格納およびチェックサムの計算を行います。 ・DATA_RECEIVED 受信データと計算したチェックサム値の一致を判定します。 ・SUM_RECEIVED 受信データと ETX の一致を判定します。
引数	• なし
リターン値	• なし
備考	なし

4.6 フローチャート

4.6.1 メイン処理フローチャート

図 4-1 にメイン処理のフローを示します。

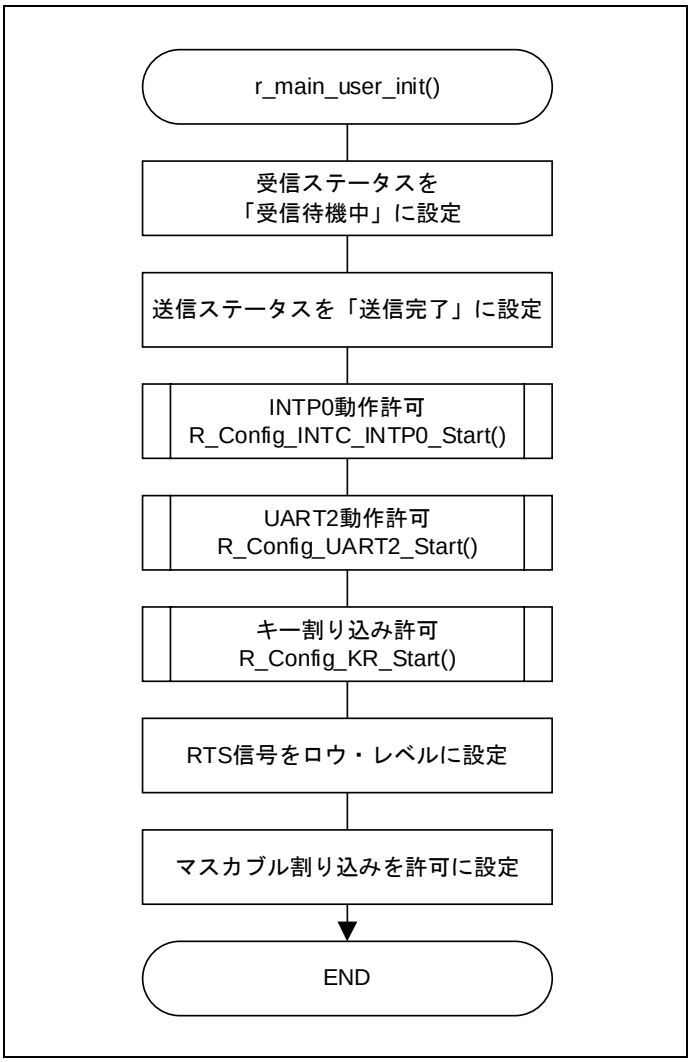
図 4-1 メイン処理



4.6.2 初期設定関数フローチャート

図 4-2 に初期設定関数のフローを示します。

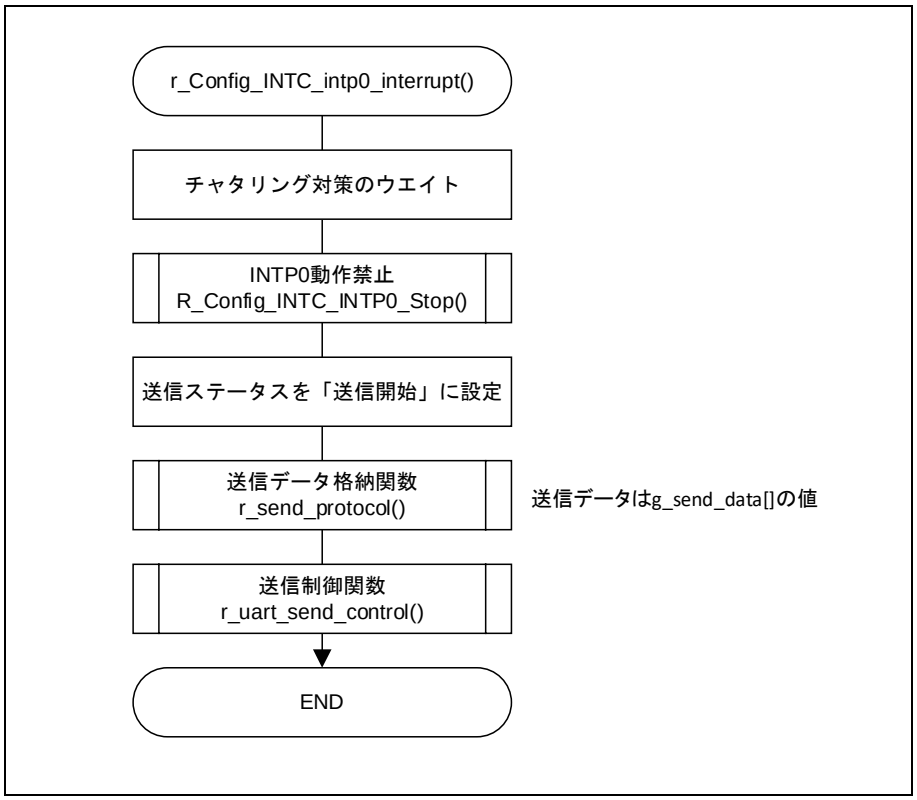
図 4-2 初期設定関数



4.6.3 INTP0 割り込み関数フローチャート

図 4-3 に INTP0 割り込み関数のフローを示します。

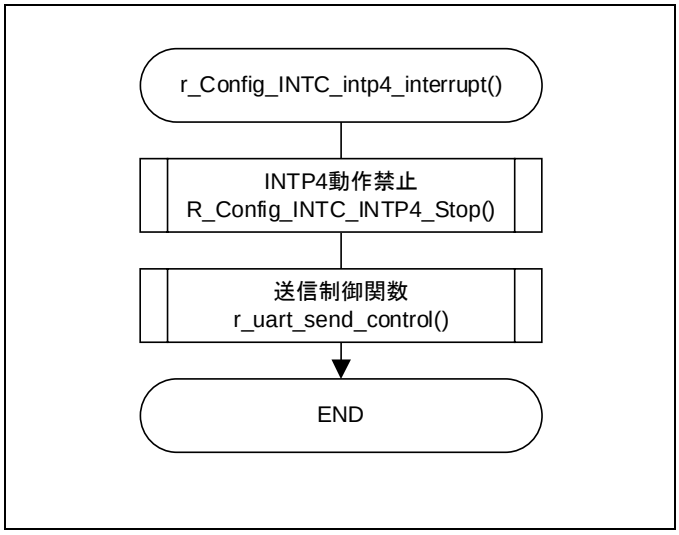
図 4-3 INTP0 割り込み関数



4.6.4 INTP4 割り込み関数フローチャート

図 4-4 に INTP4 割り込み関数のフローを示します。

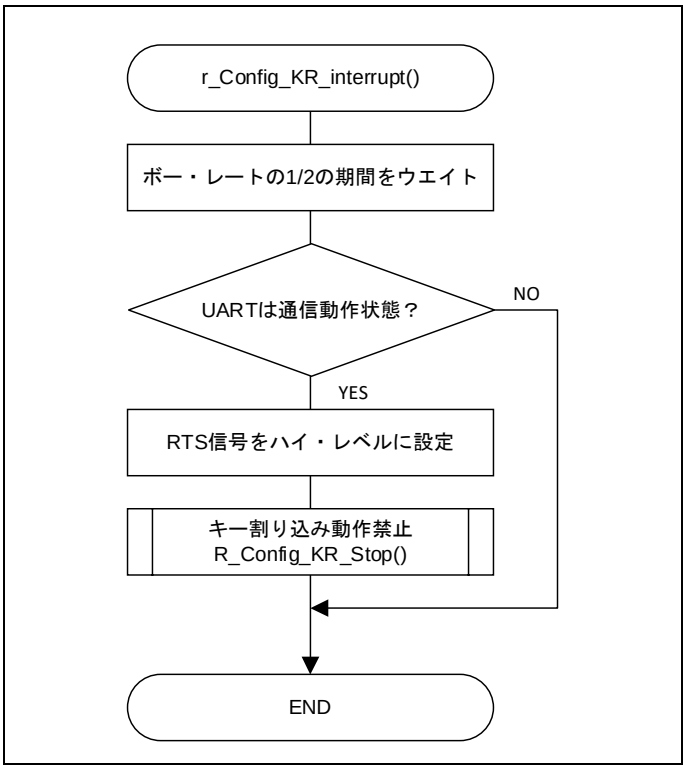
図 4-4 INTP4 割り込み関数



4.6.5 キー割り込み関数フローチャート

図 4-5 にキー割り込み関数のフローを示します。

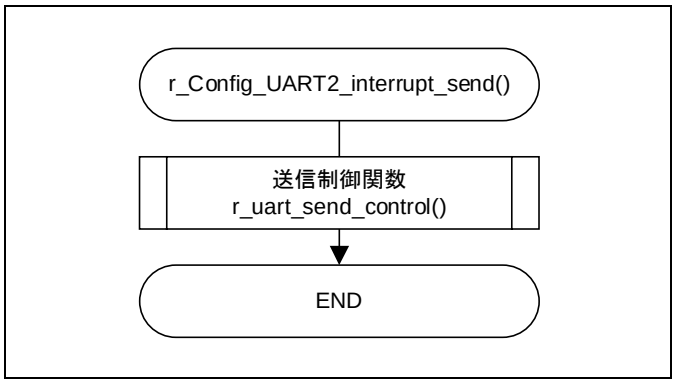
図 4-5 キー割り込み関数



4.6.6 UART2 送信完了割り込み関数フローチャート

図 4-6 に UART2 送信完了割り込み関数のフローを示します。

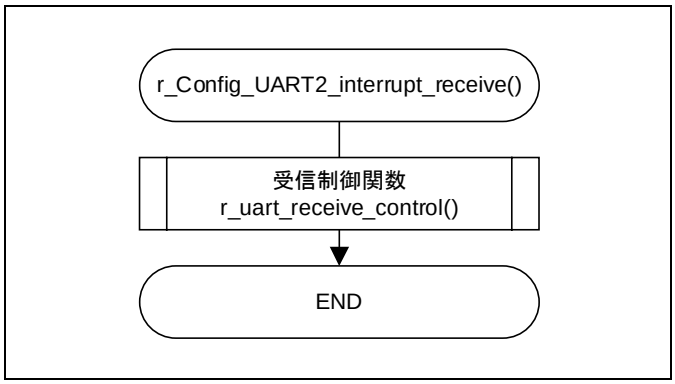
図 4-6 UART2 送信完了割り込み関数



4.6.7 UART2 受信完了割り込み関数フローチャート

図 4-7 に UART2 受信完了割り込み関数のフローを示します。

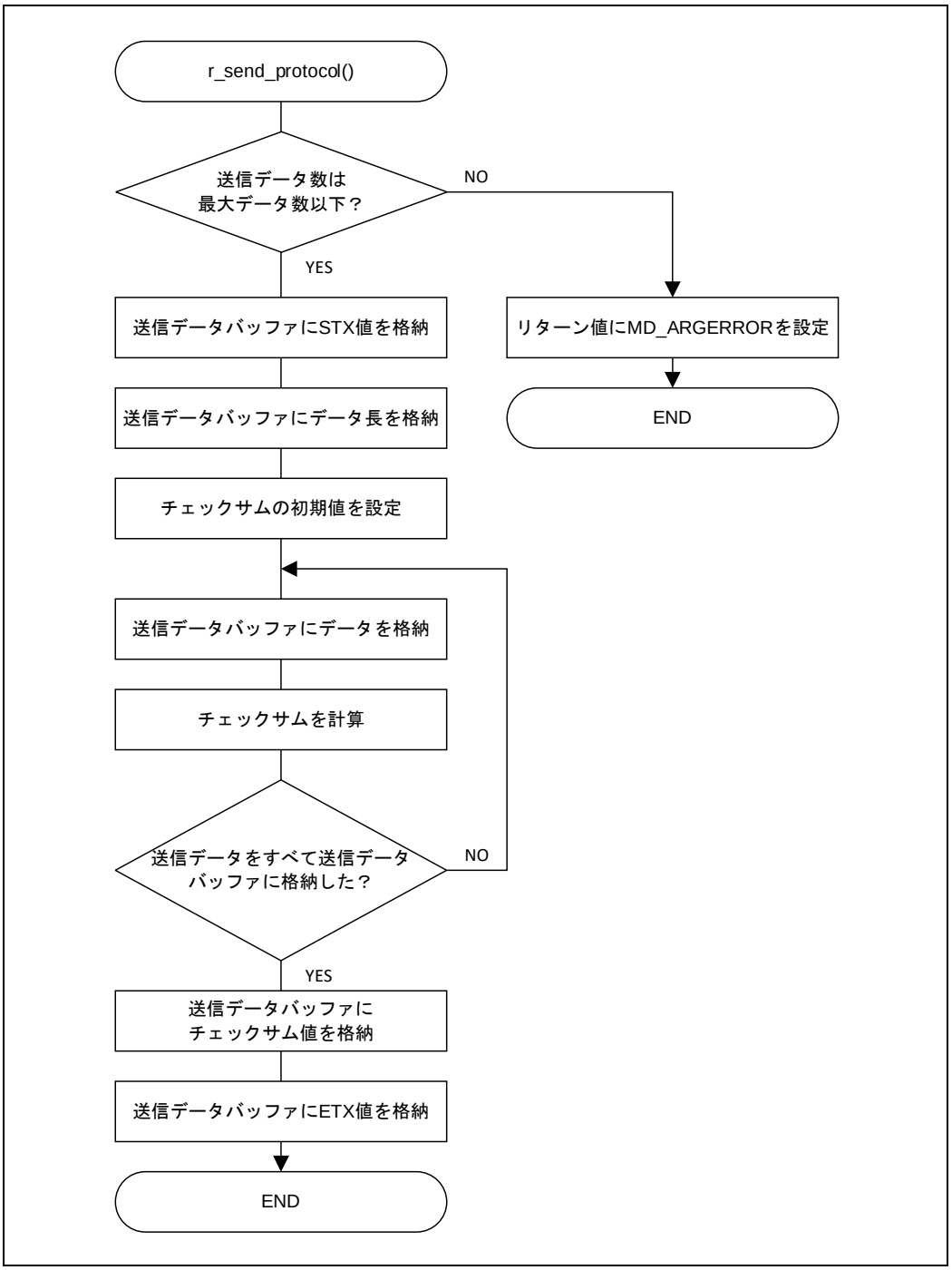
図 4-7 UART2 受信完了割り込み関数



4.6.8 送信データ格納関数フローチャート

図 4-8 に送信データ格納関数のフローを示します。

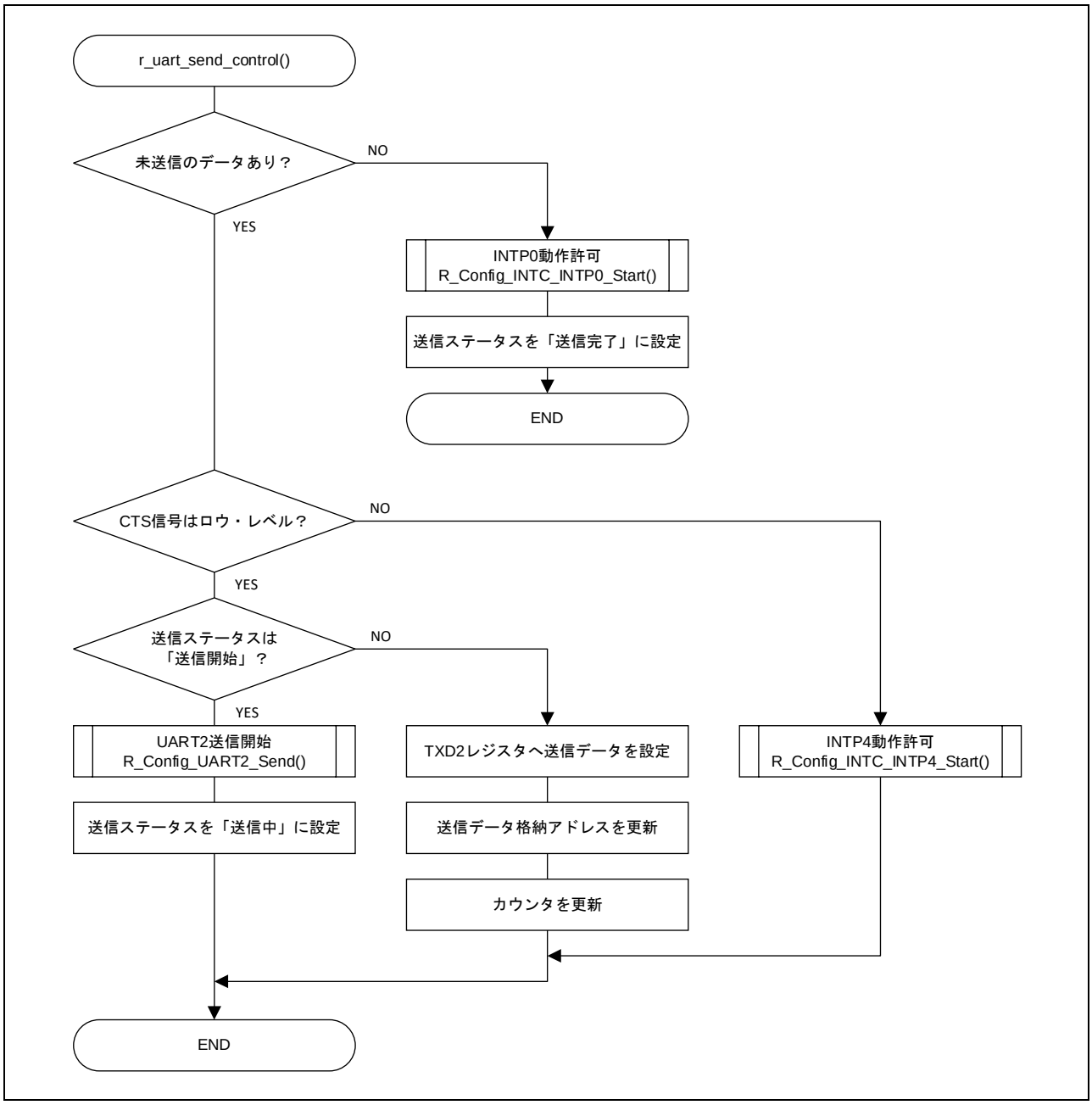
図 4-8 送信データ格納関数



4.6.9 送信制御関数フローチャート

図 4-9 に送信制御関数のフローを示します。

図 4-9 送信制御関数



4.6.10 受信制御関数フローチャート

図 4-10 ~ 図 4-12 に受信制御関数のフローを示します。

図 4-10 受信制御関数 (1/3)

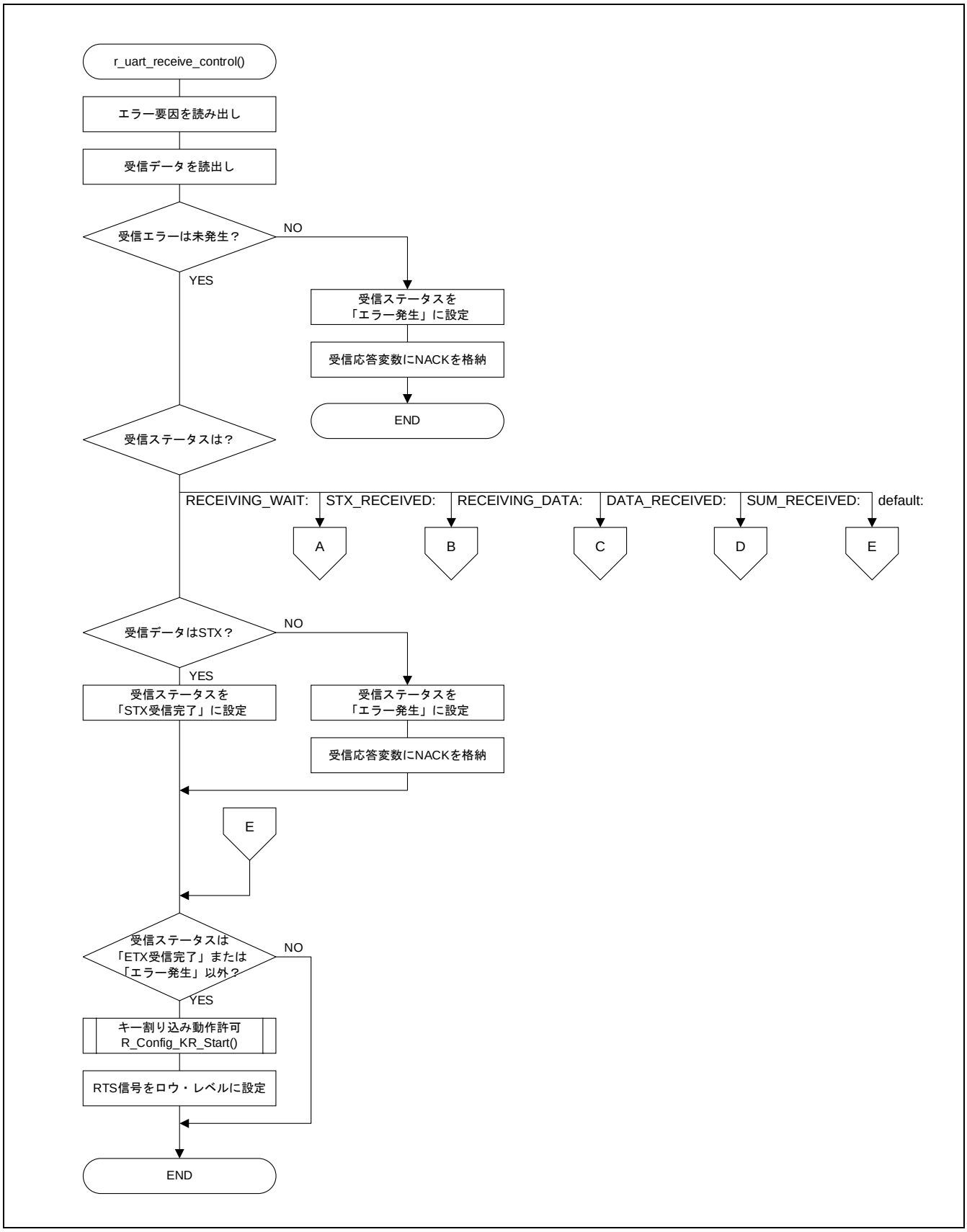


図 4-11 受信制御関数 (2/3)

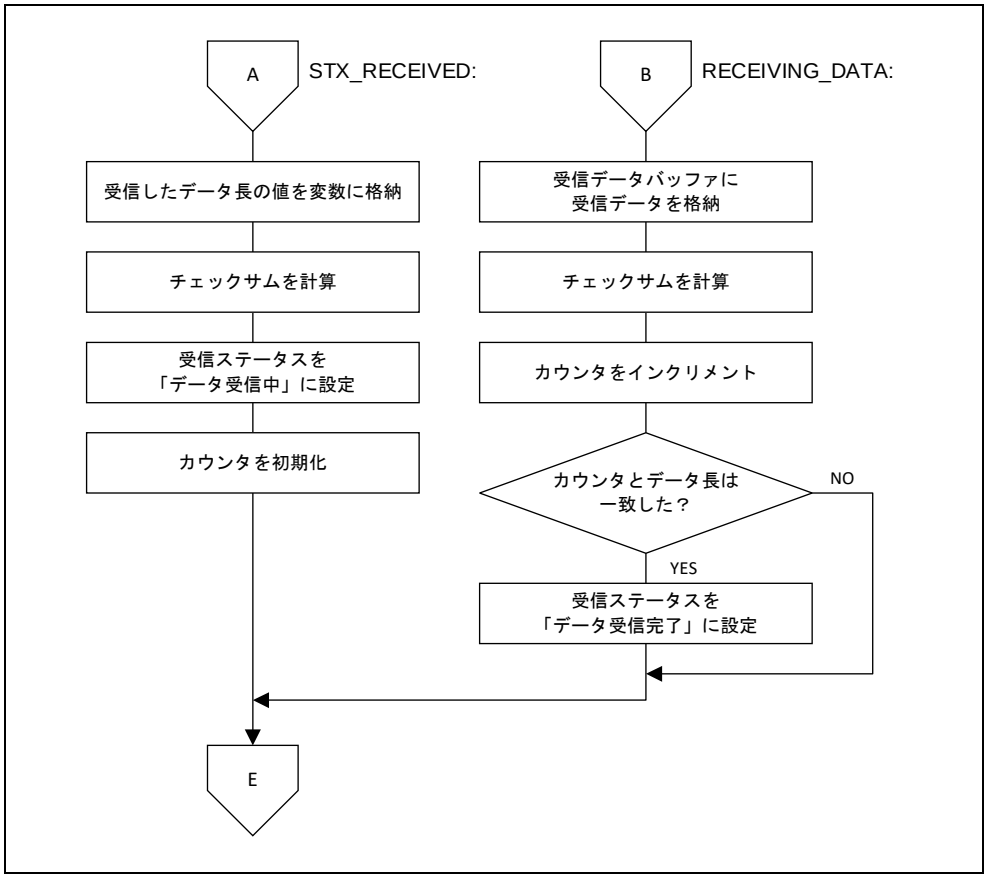
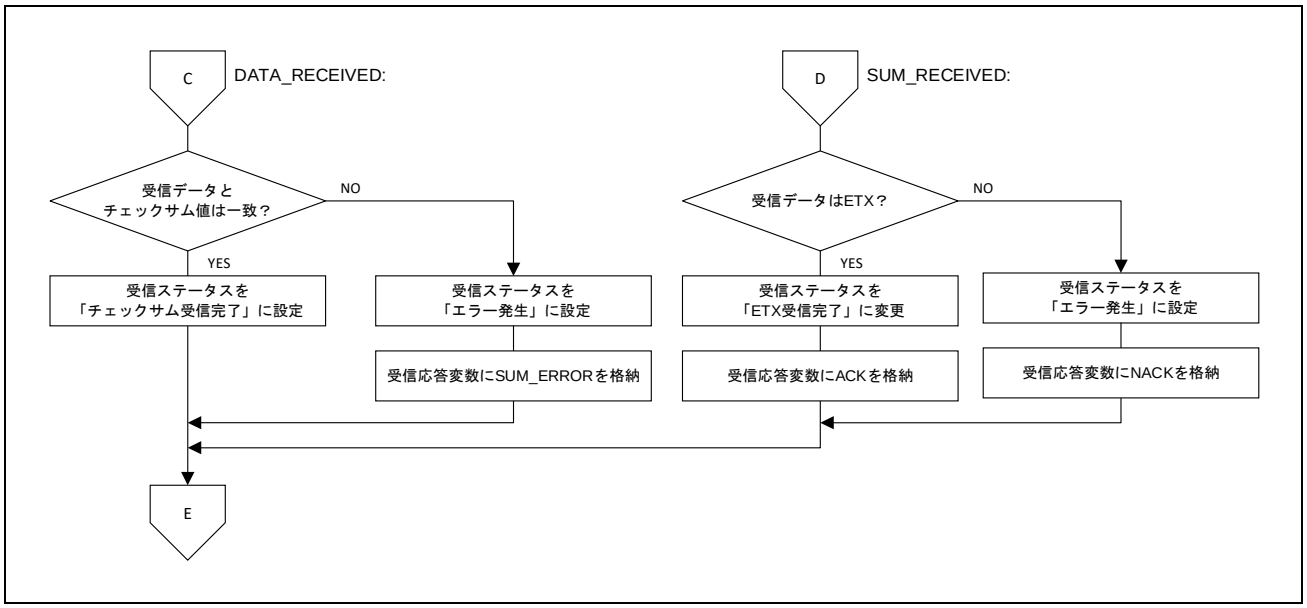


図 4-12 受信制御関数 (3/3)



5. 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。

6. 参考ドキュメント

RL78/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896J)
RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015J)
(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。)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2024.12.6	-	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うと、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も、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、脆弱性問題に起因し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て、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といたします。
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

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/

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。