
V850E2/MN4

R01AN0926JJ0100

Rev.1.00

2012.02.07

UARTJ 制御編

要旨

本アプリケーションノートは、シリアル・インターフェース UARTJ (FIFO 付き) の設定方法、およびサンプルコードの動作概要や使用方法を説明したものです。サンプルコードでは、ボー・レートを 19200bps に設定して、UARTJ1 と UARTJ3 の間でシリアル通信を行い、UARTJ1 からデータを送信して、UARTJ3 が受信します。送信/受信にそれぞれ 16 バイトのユーザ配列メモリ領域を用意しています。

動作確認デバイス

V850E2/MN4 マイクロコントローラ

目次

1. 概要	2
2. 利用環境	5
3. ソフトウェアについて	6
4. アプリケーション例	7

1. 概要

このサンプルコードでは、UARTJ の使用例を示しています。

クロック周波数の選択、ポート入出力の設定など、V850E2/MN4 マイクロコントローラの基本的な初期設定を行います。また、初期設定完了後のメイン処理動作では、UARTJ1 と UARTJ3 の間でシリアル通信を行い、UARTJ1 からデータを送信して、UARTJ3 が受信します。

ソフトウェアのおもな内容は次のとおりです。

各処理の詳細については 4.1 フローチャートで説明します。

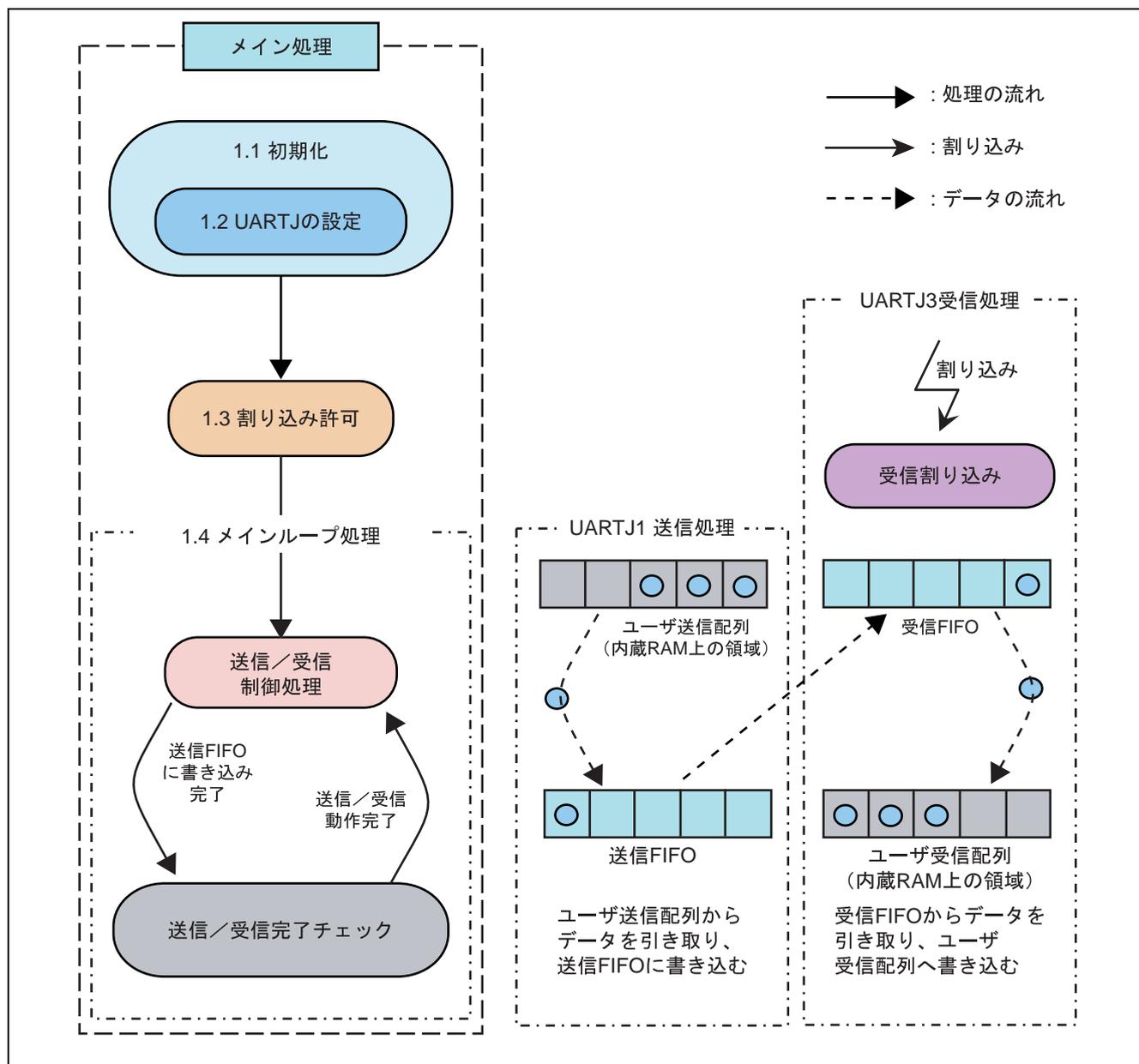


図 1.1 UARTJ のソフトウェア処理の流れ

通信基本仕様は以下のとおりです。

受信用 I/F	UARTJ3
送信用 I/F	UARTJ1
動作モード	FIFO を使って連続送信／受信
転送方向	LSB ファースト
送信／受信時のパリティ	パリティ・ビットなし
送信データ 1 フレーム	8 ビット
ストップ・ビット	1 ビット
ボー・レート	19200bps
送信／受信 FIFO 容量	16 バイト

1.1 初期化

汎用レジスタや機能端子などの初期化処理を行います。

<ポートの設定>

- ポート n・ファンクション・コントロール・エクспанション・レジスタ (PFCE_n)
- ポート n・ファンクション・コントロール・レジスタ (PFC_n)
- ポート n・モード・コントロール・レジスタ (PMC_n)
- ポート n・モード・レジスタ (PM_n)

1.2 UARTJ の設定

UARTJ の動作を制御するために、以下の設定を行います。詳細は、4.2 レジスタの設定 を参照してください。

<UART 制御の設定>

- UARTJ_n 制御レジスタ 0 (URTJ_nCTL0)
- UARTJ_n 制御レジスタ 1 (URTJ_nCTL1)
- UARTJ_n 制御レジスタ 2 (URTJ_nCTL2)

<FIFO 制御の設定>

- UARTJ_n マクロ FIFO 制御レジスタ 0 (URTJ_nFCTL0)
- UARTJ_n マクロ FIFO 制御レジスタ 1 (URTJ_nFCTL1)

1.3 割り込み許可

- EI 命令で割り込みを許可します。

1.4 メインループ処理

メインループ中の処理は、大きく 2 つの処理に分かれており、送信/受信制御処理と送信/受信完了待ちを繰り返します。UARTJ の動作設定完了後、URTJ_nFTX レジスタに送信データを書き込むと送信 FIFO にデータが格納されます。送信 FIFO に格納されたデータは送信シフト・レジスタに転送され、TXD_nF 端子を介して、シリアル通信の送信を開始します。

このサンプルコードでは、送信側 UARTJ1 で 16 バイトデータを送信し、受信側 UARTJ3 で UARTJ1 から送信された 16 バイトデータを受信します。内蔵 RAM 領域には、送信データと受信データの格納用ユーザ送信/受信配列として、それぞれ 16 バイト (=1 バイト (1 データ) ×16) の領域を確保しています。

メインループ動作において、送信/受信割り込みまたはステータス割り込みが発生すると、各割り込み処理に入ります。このサンプルコードでは、以下の時点で割り込みが発生します。各割り込み処理の詳細は第 4 章を参照してください。

- 送信割り込み：送信 FIFO が空になった時
- 受信割り込み：受信 FIFO がフルになった時
- ステータス割り込み：受信または送信中にエラー状態が発生した時

2. 利用環境

この章では、このサンプルコードで使用する場合の回路図および利用環境を説明します。

2.1 回路図

回路図の詳細は V850E2/MN4 ターゲット・ボード QB-V850E2MN4DUAL-TB ユーザーズマニュアル (R20UT0683XJ) を参照してください。

このプログラムでは、UARTJ1 と UARTJ3 の間でシリアル通信を行い、UARTJ1 からデータを送信して、UARTJ3 が受信します。UART1 の端子 TXD1F は P13_4、端子 RXD1F は P13_5 を使用しています。UART3 の端子 TXD3F は P4_6、端子 RXD3F は P4_7 を使用しています。端子間の接続については、TXD1F が RXD3F と、RXD1F が TXD3F と接続しています。

LED1 と LED2 はポート 13 と接続しています。LED1 は P13_7、LED2 は P13_6 を使用しています。

2.2 開発環境

サンプルコードを動作させるために、次の開発ツールをインストールする必要があります。

- CubeSuite+

ルネサスエレクトロニクス社製統合開発環境 CubeSuite+は、コーディング、アセンブル/コンパイル、エミュレータやシミュレータを使ったデバッグ、フラッシュプログラミング等のお客様のアプリケーション開発に必要なあらゆるソフトウェア開発ツールをシームレスかつ簡単に操作することができます。

- MINICUBE

ルネサスエレクトロニクス社製 JTAG 接続方式の汎用型インサーキットエミュレータです。実プロセッサをオンボードの状態でのデバッグでき、透過性の高い安定したエミュレーション機能を提供しています。

MINICUBE と TB ボードを接続するには、変換アダプタが必要です。

3. ソフトウェアについて

この章では、ダウンロードする圧縮ファイルのファイル構成について説明します。

3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名 (ツール構造)	説明	共通ソース ファイル	CubeSuite+用 ファイル
crtE.s	ハードウェア初期化処理		●
V850E2MN4.dir	リンク・ディレクティブ・ファイル		●
uartj.h	変数、関数宣言	●	
main.c	メイン処理	●	
initial.c	ソフトウェア初期化処理	●	
uartj_transmit.c	送信処理	●	
uartj_receive.c	受信処理	●	
interrupt.c	割り込み処理	●	

4. アプリケーション例

この章では、UARTJ の機能の設定について説明します。

4.1 フローチャート

このサンプルコードのフローチャートを、次に示します。

4.1.1 メイン処理フロー

メイン処理フローでは、まずマスクブル割り込みを禁止します。各設定を行ったあと、マスクブル割り込みと送信／受信割り込みの許可を行います。メインループ処理では、送信／受信制御と FIFO 送信／受信完了待ちを繰り返し実行します。

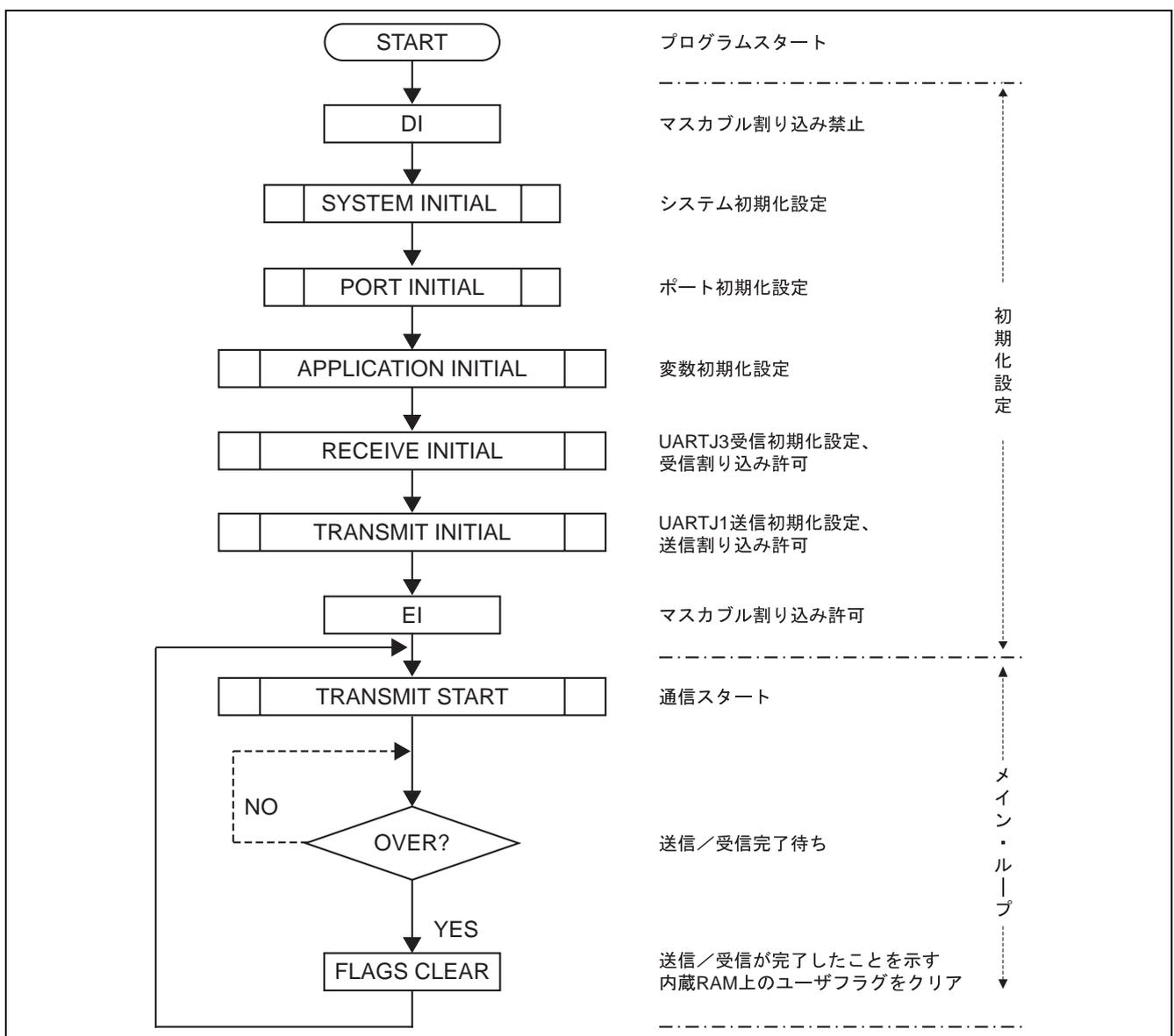


図 4.1 メイン処理フロー

4.1.2 受信割り込み処理

UARTJ のデータ受信に応じて、受信 FIFO のデータを読み出し、データ数が URTJnSLRP[3:0] の設定値と一致した時に割り込みが発生します。

受信割り込みでは、受信した回数分、受信 FIFO からデータを読み出し、ユーザ受信配列に受信データを格納します。全ての受信データをユーザ受信配列に格納後、全部のデータ（16 バイト）をユーザ受信配列に格納したことを示す内蔵 RAM 上のフラグ（受信完了フラグとする）を 1 にセットします。

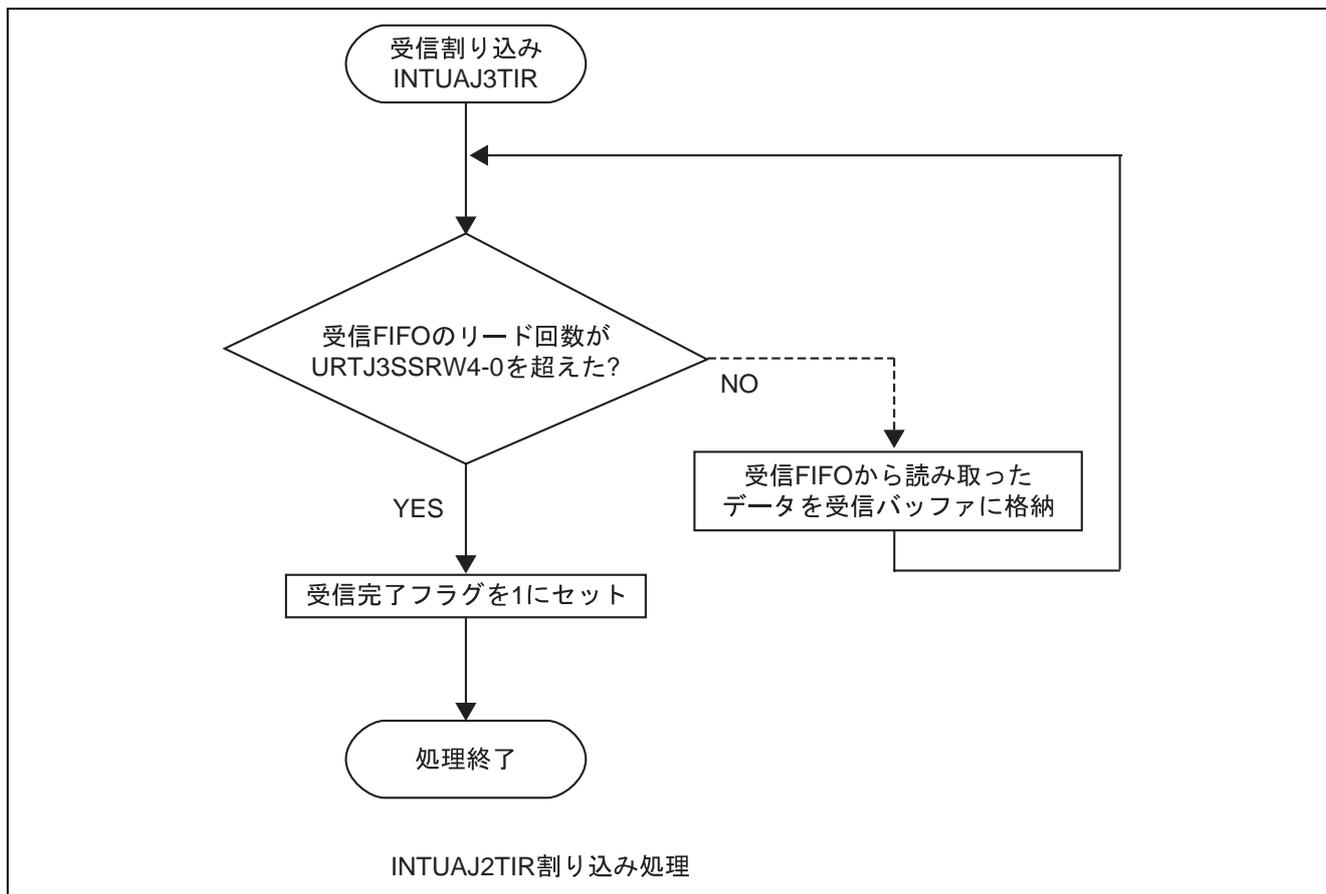


図 4.2 受信割り込み処理

4.1.3 送信割り込み処理

送信 FIFO の格納データ数がポインタ設定ビット URTJ1SLTP3-0 の設定値と一致し、URTJ1FTX へのライトアクセスがない場合に割り込みが発生します。

送信割り込みでは、全部のデータ (16 バイト) を送信 FIFO に転送したことを示す内蔵 RAM 上のフラグ (送信完了フラグとする) を 1 にセットします。

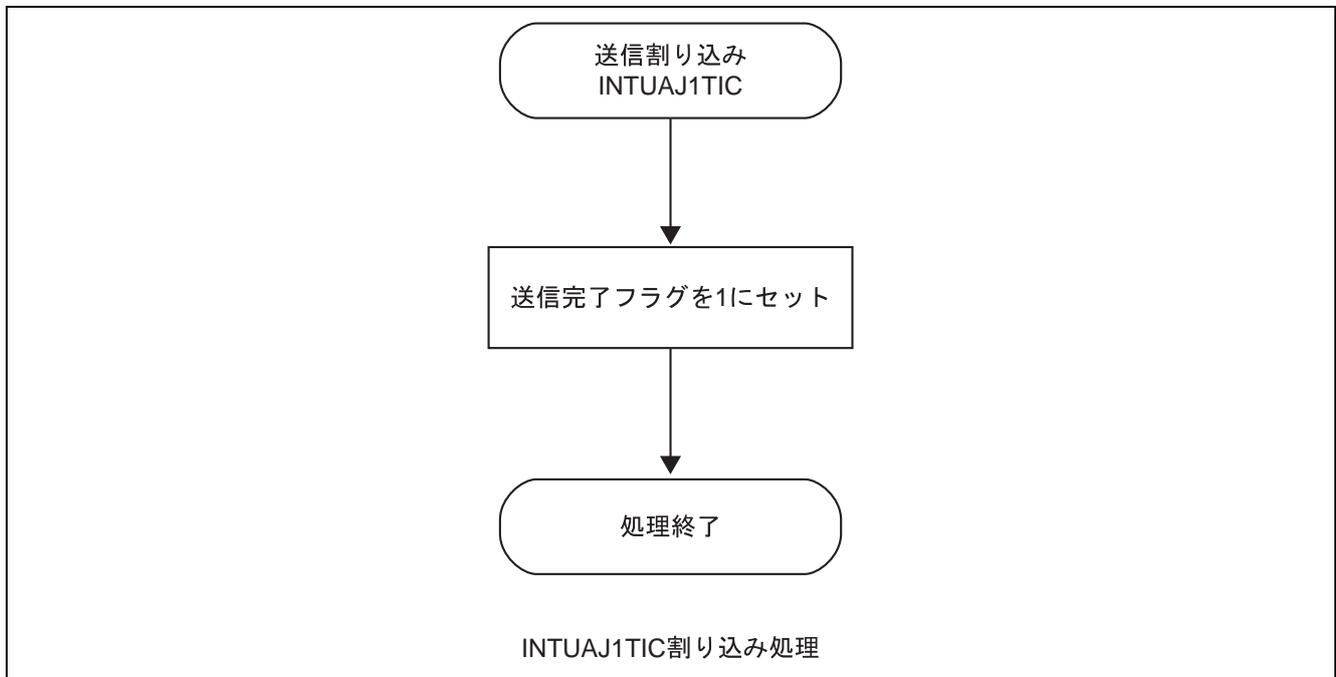


図 4.3 送信割り込み処理

4.1.4 送信／受信動作制御処理

UARTJ の初期化処理完了後、送信データを URTJ1FTX レジスタに書き込むと、送信を開始します。受信側は、送信側からスタート・ビットを受けると受信を開始します。

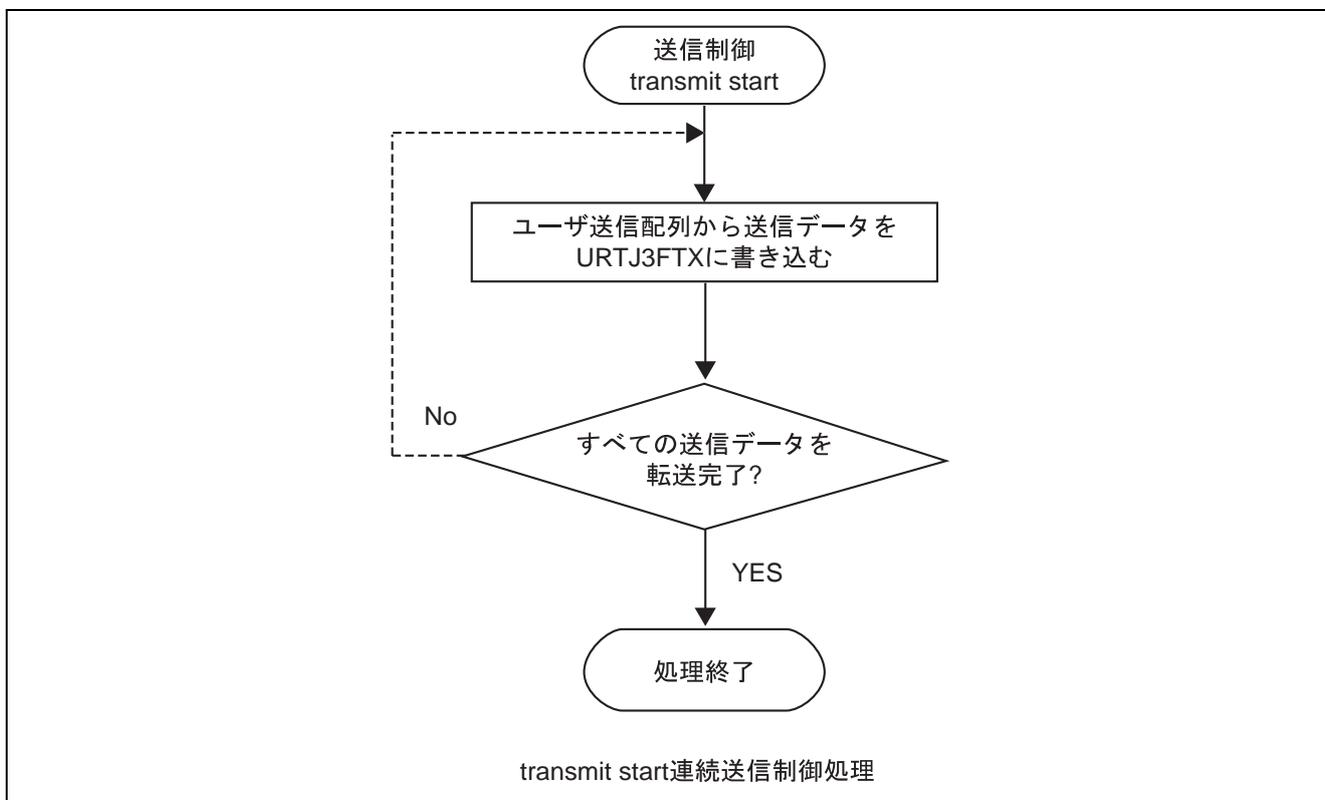


図 4.4 送信/受信動作制御処理

4.1.5 ステータス割り込み処理

送信／受信動作中エラー状態が発生すると、ステータス割り込み要求が発生します。

- データ一貫性エラー
- タイムアウト・エラー
- フレーミング・エラー
- パリティ・エラー
- オーバーラン・エラー
- オーバーフロー・エラー

送信または受信中にエラーを検出するとエラーに応じて $URTJnSTR1$ と $URTJnFSTR1$ レジスタのエラー・フラグが 1 にセットされます。UARTJ のステータス割り込み処理では、送信／受信 FIFO をクリアします。クリア方法は、16 バイトデータ転送に相当する時間を待つことで送信 FIFO が空になります。また、受信 FIFO で全ての受信データを読み出すと、受信 FIFO が空になります。内蔵 RAM 上の送信および受信フラグを 1 にセットします。 $URTJnSTC$ レジスタと $URTJnFSTC$ レジスタを用いて SFR エラー・フラグをクリアすると共に、受信／送信割り込み要求またはステータス割り込み要求をクリアします。そして、UARTJ1 と UARTJ3 動作をいったん停止して、メインループに戻ります（メインループで送信／受信を再開します）。

以下のステータス割り込み処理は UARTJ1 と UARTJ3 で共通です。

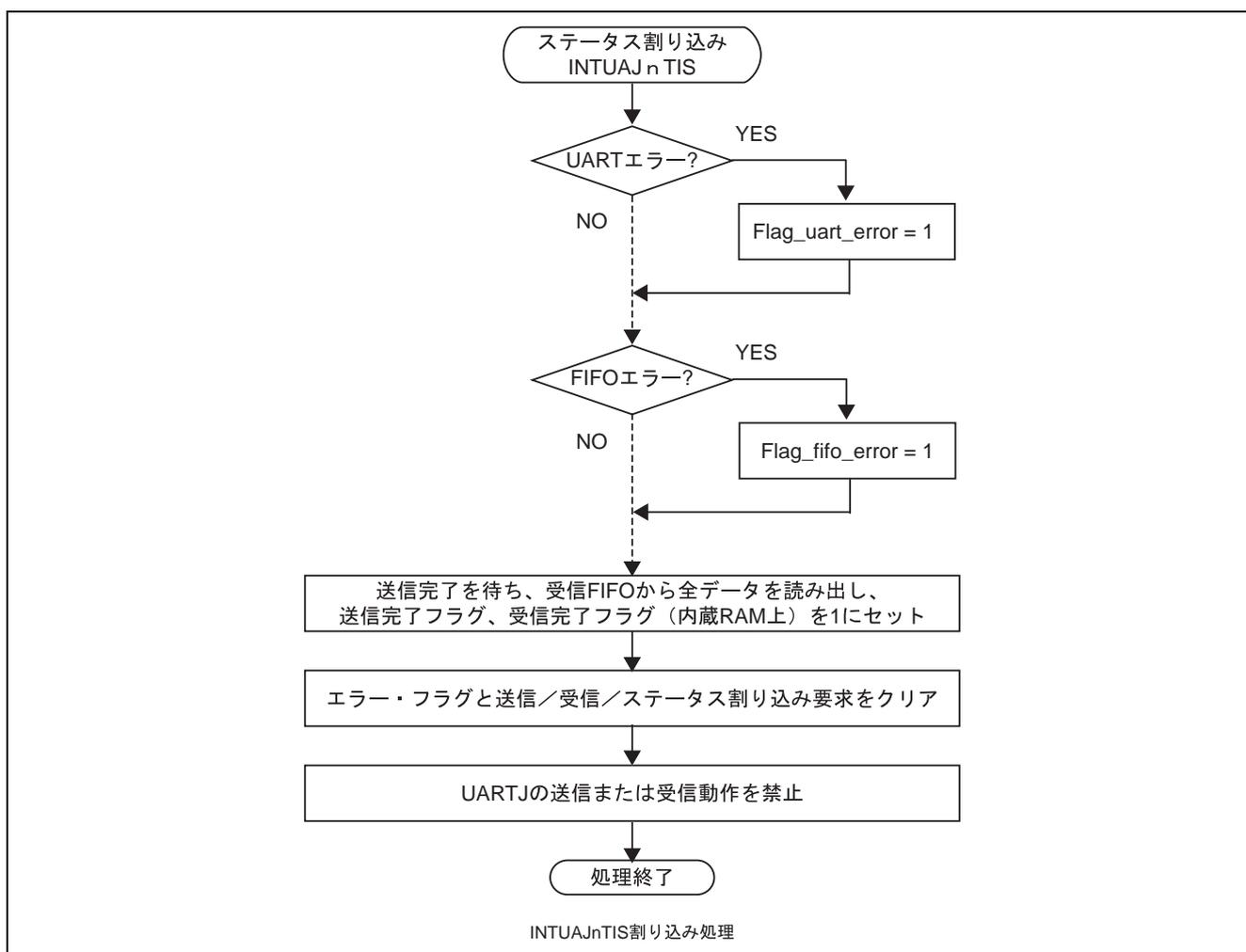


図 4.5 ステータス割り込み処理

4.2 レジスタの設定

この節では、4.1 フローチャートに基づき、各レジスタの設定について説明します。UARTJ の動作を制御するために、以下のレジスタを設定します。

4.2.1 ポート設定

このアプリケーションノートは UARTJ1 と UARTJ3 二つのマクロを使って、シリアル送信/受信動作を行っています。UARTJ1 と UARTJ3 それぞれ対応しているポートを必ず適切に設定してください。

LED はポート 13 と接続しています。LED1 は P13_7、LED2 は P13_6 を使用しています。

マクロ	端子	PMC	PFCE	PFC	PM	対応機能
UARTJ1	TXD1F	1	0	1	0	兼用モード2、出力
	RXD1F	1	0	1	1	兼用モード2、入力
UARTJ3	TXD3F	1	0	1	0	兼用モード2、出力
	RXD3F	1	0	1	1	兼用モード2、入力
PORT	P13_6	0	0	0	0	ポート・モード、出力
	P13_7	0	0	0	0	ポート・モード、出力

設定例：

```

/* P4_6: URTJTA3RXD; alternative mode 2; input*/
/* P4_7: URTJTA3TXD; alternative mode 2; output*/
PFCE4 = 0x0000;
PFC4  = 0x00C0;
PMC4  = 0x00C0;
PM4   = 0x0040;

/* P13_6,7:LEDs; port mode; output*/
/* P13_5:URTJTA1RXD; alternative mode 2; input*/
/* P13_4:URTJTA1TXD; alternative mode 2; output*/
PFCE13 = 0x0000;
PFC13  = 0x0030;
PMC13  = 0x0030;
PM13   = 0x0020;

```

4.2.2 UARTJn 制御レジスタ 2 (URTJnCTL2)

URTJnCTL2 レジスタは、UARTJn マクロがシリアル転送動作で使われるボー・レート指定します。

送信/受信クロックはPCLKを基づいて生成します。PCLKからのクロック信号をプリスケアラで分周し、ボー・レート・ジェネレータで指定のボー・レート・クロックを生成します。

このサンプルコードでは、ボー・レートを19200bpsに設定しています。

アクセス 16ビット単位でリード/ライト可能です。

アドレス <URTJn_base_OS> + 24_H

初期値 EFFF_H どのリセット要因でも初期化されます。

15	14	13	12	11	10	9	8
URTJnPRS[2:0]			0	URTJnBRS[11:8]			
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
URTJnBRS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

表 20-9 URTJnCTL2 レジスタの内容

ビット位置	ビット名	機能																						
15-13	URTJn PRS[2:0]	プリスケアラ・クロック (PRCLK) 分周値 0: PRCLK = PCLK / 2 ⁰ 1: PRCLK = PCLK / 2 ¹ 2: PRCLK = PCLK / 2 ² 3: PRCLK = PCLK / 2 ³ 4: PRCLK = PCLK / 2 ⁴ 5: PRCLK = PCLK / 2 ⁵ 6: PRCLK = PCLK / 2 ⁶ 7: PRCLK = PCLK / 2 ⁷																						
11-0	URTJn BRS[11:0]	ボー・レート・クロック (BRCLK) 分周値 <table border="1"> <thead> <tr> <th>URTJn BRS[11:0]</th><th>送信/受信 BRCLK</th><th>BF 受信クロック</th></tr> </thead> <tbody> <tr> <td>000_H</td><td rowspan="5">PRCLK / (2 x 4)</td><td rowspan="5">PRCLK / 4</td></tr> <tr><td>001_H</td></tr> <tr><td>002_H</td></tr> <tr><td>003_H</td></tr> <tr><td>004_H</td></tr> <tr> <td>005_H</td><td>PRCLK / (2 x 5)</td><td>PRCLK / 5</td></tr> <tr> <td>...</td><td>PRCLK / (2 x URTJnBRS[11:0])</td><td>PRCLK / URTJnBRS[11:0]</td></tr> <tr> <td>FFE_H</td><td>PRCLK / (2 x 4094)</td><td>PRCLK / 4094</td></tr> <tr> <td>FFF_H</td><td>PRCLK / (2 x 4095)</td><td>PRCLK / 4095</td></tr> </tbody> </table>	URTJn BRS[11:0]	送信/受信 BRCLK	BF 受信クロック	000 _H	PRCLK / (2 x 4)	PRCLK / 4	001 _H	002 _H	003 _H	004 _H	005 _H	PRCLK / (2 x 5)	PRCLK / 5	...	PRCLK / (2 x URTJnBRS[11:0])	PRCLK / URTJnBRS[11:0]	FFE _H	PRCLK / (2 x 4094)	PRCLK / 4094	FFF _H	PRCLK / (2 x 4095)	PRCLK / 4095
URTJn BRS[11:0]	送信/受信 BRCLK	BF 受信クロック																						
000 _H	PRCLK / (2 x 4)	PRCLK / 4																						
001 _H																								
002 _H																								
003 _H																								
004 _H																								
005 _H	PRCLK / (2 x 5)	PRCLK / 5																						
...	PRCLK / (2 x URTJnBRS[11:0])	PRCLK / URTJnBRS[11:0]																						
FFE _H	PRCLK / (2 x 4094)	PRCLK / 4094																						
FFF _H	PRCLK / (2 x 4095)	PRCLK / 4095																						

注意 このレジスタは、UARTJn 動作禁止時 (URTJnCTL0.URTJnPW = 0) にのみ書き込み可能です。

図 4.6 URTJnCTL2 レジスタのフォーマット

設定例 :

```
URTJnCTL2 = 0x60D9;
```

```
/* Assume that PCLK is set to 66.667 MHz */
```

```
/* PRSCLK (1/8), baud rate clock (2*217) */
```

```
/* 19200≈PCLK/ (8*2*217) */
```

4.2.3 UARTJn 制御レジスタ 0 (URTJnCTL0)

URTJnCTL0 レジスタは、UARTJn マクロの基本的なシリアル転送動作を制御します。

サンプルコードでは、まずそれぞれの動作を禁止し、各レジスタ制御の設定を行った後、UARTJn の動作を許可します。

アクセス 8/1 ビット単位でリード/ライト可能です。

アドレス <URTJn_base_USER> + 00_H

初期値 00_H どのリセット要因でも初期化されます。

7	6	5	4	3	2	1	0
URTJn PW	URTJn TXE	URTJn RXE	0	0	0	0	URTJn SLDC
R/W	R/W	R/W	R	R	R	R	R/W

表 20-7 URTJnCTL0 レジスタの内容

ビット位置	ビット名	機能
7	URTJnPW	UARTJn イネーブル 0: UARTJn の動作を停止させます。 1: UARTJn の動作を許可します。 このビットを変更すると、すべての送信ユニットと受信ユニットが初期化されます。
6	URTJnTXE	送信動作イネーブル 0 送信動作を禁止します。 1 送信動作を許可します。 • 送信を開始するには、URTJnPW をセット (1) したあと、URTJnTXE をセット (1) します。送信を停止するには、URTJnTXE をクリア (0) したあと、URTJnPW を 0 にクリア (0) します (URTJnTXE と URTJnPW ビットを同時にクリア (0) することも可能です)。 • 送信ユニットを初期化するには、URTJnTXE をクリア (0) し、プリスケアラ・クロックの 2 周期分の時間を経過してから、再度 URTJnTXE をセット (1) します (プリスケアラ・クロックについては (3) 「URTJnCTL2 — UARTJn 制御レジスタ 2」を参照してください)。
5	URTJnRXE	受信許可 0: 受信動作を禁止 1: 受信動作を許可 • 受信を許可するには、URTJnPW をセット (1) したあと、URTJnRXE をセット (1) します。受信を停止するには、URTJnRXE をクリア (0) したあと、URTJnPW をクリア (0) します (URTJnRXE と URTJnPW ビットを同時にクリア (0) することも可能です)。 • 受信ユニットを初期化するには、URTJnRXE を 0 にクリア (0) し、プリスケアラ・クロックの 2 周期分の時間を経過してから、再度 URTJnRXE をセット (1) します。URTJnRXE がセット (1) されてからプリスケアラ・クロックの 2 周期分の時間を経過してから、受信が許可されます。URTJnRXE がセット (1) されてから 4 プリスケアラ・クロック・サイクルが経過すると、URTJnTRXD 信号の立ち上がりエッジの検出が可能になります (プリスケアラ・クロックについては (3) 「URTJnCTL2 — UARTJn 制御レジスタ 2」を参照してください)。
0	URTJnSLDC	データチェックの有効/無効 0: 一貫性チェック無効 1: 一貫性チェック有効 データ送信時にデータ一貫性エラー・チェックを有効にするかどうかを選択します。 セット (1) した場合、送信データと受信データの比較を行います。比較の結果不一致が検出されると、URTJnSTR1.URTJnDCE がセット (1) され、ステータス割り込み要求 INTUAJnTIS が発生します。 このビットは送信開始時のみ参照されます。したがって、このビットの値を送信処理中に変更しても、後続の送信処理は送信開始時の設定に応じて実行されます。

図 4.7 URTJnCTL0 レジスタのフォーマット (1/2)

- 注意
1. UARTJn が次の状態をすべて満たす場合、送信を禁止してください。
 - 送受信許可時 (URTJnCTL0.URTJnPW = URTJnRXE = URTJnTXE = 1)
 - データ一貫性チェックが有効 (URTJnCTL0.URTJnSLDC = 1)
 - 送信中、または送信完了状態
 受信許可状態に保つには、次の手順に従ってください。
 - 保留されている送信がないことを確認します (URTJnSTR0.URTJnSSBT = URTJnSST = 0)。
 - 保留されている受信がないことを確認します (URTJnSTR0.URTJnSSBR = URTJnSSR = 0)。
 - URTJnCTL0.URTJnTXE = 0 を設定して送信を禁止します。
 この手順を実行するのは、URTJnCTL0.URTJnTXE をクリア (0) することによってデータ一貫性エラー・フラグ URTJnSTR1.URTJnDCE がクリア (0) されるためです。上記手順を実行することによって、データ転送中またはデータ転送完了後に送信を禁止したときにデータ一貫性エラーが発生しなくなります。
 2. UARTJn が次の状態をすべて満たすと場合、受信を禁止してください。
 - 送受信許可時 (URTJnCTL0.URTJnPW = URTJnRXE = URTJnTXE = 1)
 - データ一貫性チェックが有効 (URTJnCTL0.URTJnSLDC = 1)
 - 送信中、または送信完了状態
 送信許可状態に保つには、次の手順に従ってください。
 - 保留されている送信がないことを確認します (URTJnSTR0.URTJnSSBT = URTJnSST = 0)。
 - 保留されている受信がないことを確認します (URTJnSTR0.URTJnSSBR = URTJnSSR = 0)。
 - URTJnCTL0.URTJnRXE を 0 に設定して受信を禁止します。
 この手順を実行するのは、URTJnCTL0.URTJnTXE をクリアするとデータ一貫性エラー・フラグ URTJnSTR1.URTJnDCE がクリアされて無効になるためです。上記手順を実行することによって、すでに送信されたデータに対するデータ一貫性エラーは報告されなくなります。
 3. 次の状態をすべて満たす場合、データの送信を開始しないでください。
 - データ一貫性チェックが有効 (URTJnCTL0.URTJnSLDC = 1)
 - BF 受信許可時 (URTJnSTR0.URTJnSSBR = 1)
 - 受信中の BF 検出禁止時 (URTJnCTL1.URTJnSLBM = 0)
 上記の場合、BF の受信が完了したときにデータ一貫性エラーが発生します。このとき、ステータス割り込み INTUAJnTIS が発生し、BF の受信完了は報告されません (URTJnSTR1.URTJnBSF は 0 を保持)。したがって、BF の受信完了は認識されません。

図 4.8 URTJnCTL0 レジスタのフォーマット (2/2)

設定例 :

```

URTJnPW = 1;          /* Enable UARTJn */
URTJnTXE = 1;        /* Enable transmission */
URTJnRXE = 1;        /* Enable reception */
URTJnBYTE = 0x00;    /* Disable UARTJn */

```

4.2.4 UARTJn 制御レジスタ 1 (URTJnCTL1)

URTJnCTL1 レジスタは、UARTJn マクロがシリアル・データ転送で使われるデータ・フレームの形式を指定します。

このサンプルコードでは、BF 関連の送信/受信関連動作は使用していません。

アクセス 16 ビット単位でリード/ライト可能です。

アドレス <URTJn_base_OS> + 20_H

初期値 5002_H どのリセット要因でも初期化されます。

15	14	13	12	11	10	9	8
URTJn SLBM	URTJn BLG[2:0]			0	0	0	URTJn CLG
R/W	R/W	R/W	R/W	R	R	R	R/W
7	6	5	4	3	2	1	0
URTJn SLP[1:0]	URTJn TDL	URTJn RDL	0	URTJn SLG	URTJn SLD	URTJn SLIT	
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

表 20-8 URTJnCTL1 レジスタの内容 (1/3)

ビット位置	ビット名	機能																																				
15	URTJnSLBM	BF 受信モード選択 0: データ受信中の BF の受信を禁止します。 1: データ受信中の BF の受信を許可します。 • このビットは、受信禁止時 (URTJnCTL0.URTJnPW = 0 または URTEnCTL0.URTErRXE = 0) にのみ変更できます。																																				
14 ~ 12	URTJnBLG[2:0]	送信中の BF ビット長 <table border="1"> <thead> <tr> <th>URTJnBLG2</th> <th>URTJnBLG1</th> <th>URTJnBLG0</th> <th>BF 長</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>1</td> <td>13 ビット</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>14 ビット</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>15 ビット</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>16 ビット</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>17 ビット</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>18 ビット</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>19 ビット</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>20 ビット</td> </tr> </tbody> </table> これらのビットは、送信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnTXE = 0) にのみ変更できます。	URTJnBLG2	URTJnBLG1	URTJnBLG0	BF 長	1	0	1	13 ビット	1	1	0	14 ビット	1	1	1	15 ビット	0	0	0	16 ビット	0	0	1	17 ビット	0	1	0	18 ビット	0	1	1	19 ビット	1	0	0	20 ビット
URTJnBLG2	URTJnBLG1	URTJnBLG0	BF 長																																			
1	0	1	13 ビット																																			
1	1	0	14 ビット																																			
1	1	1	15 ビット																																			
0	0	0	16 ビット																																			
0	0	1	17 ビット																																			
0	1	0	18 ビット																																			
0	1	1	19 ビット																																			
1	0	0	20 ビット																																			
8	URTJnCLG	受信/送信データ・ビット長 0: 7 ビット 1: 8 ビット • LIN フォーマットで送信/受信を行うときは、URTJnCLG をセット (1) してください。 • このビットは、送受信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnRXE = 0, URTJnCTL0.URTJnTXE = 0) にのみ変更できます。																																				

図 4.9 URTJnCTL1 レジスタのフォーマット (1/3)

表 20-8 URTJnCTL1 レジスタの内容 (2/3)

ビット位置	ビット名	機能																						
7, 6	URTJnSLP[1:0]	<p>パリティ・ビット選択</p> <table border="1"> <thead> <tr> <th rowspan="2">URTJnSLP1</th> <th rowspan="2">URTJnSLP0</th> <th colspan="2">動作</th> </tr> <tr> <th>送信</th> <th>受信</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>パリティ・ビットなしで出力</td> <td>パリティなしで受信</td> </tr> <tr> <td>0</td> <td>1</td> <td>0パリティを出力 (0に固定)</td> <td>パリティを判定しない</td> </tr> <tr> <td>1</td> <td>0</td> <td>奇数パリティを出力</td> <td>奇数パリティとして判定を行う</td> </tr> <tr> <td>1</td> <td>1</td> <td>偶数パリティを出力</td> <td>偶数パリティとして判定を行う</td> </tr> </tbody> </table> <ul style="list-style-type: none"> 受信中に「パリティを判定しない」を選択すると、パリティ・チェックは行われません。したがって、URTJnSTR1.UTREnPE ビットがセット (1) されないため、エラー割り込みは出力されません。 LIN フォーマットで送信/受信を行うときは、URTJnSLP[1:0] を 00_B にセットしてください。 これらのビットは、送受信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnRXE, URTJnCTL0.URTJnTXE = 0) にのみ変更できません。 	URTJnSLP1	URTJnSLP0	動作		送信	受信	0	0	パリティ・ビットなしで出力	パリティなしで受信	0	1	0パリティを出力 (0に固定)	パリティを判定しない	1	0	奇数パリティを出力	奇数パリティとして判定を行う	1	1	偶数パリティを出力	偶数パリティとして判定を行う
URTJnSLP1	URTJnSLP0	動作																						
		送信	受信																					
0	0	パリティ・ビットなしで出力	パリティなしで受信																					
0	1	0パリティを出力 (0に固定)	パリティを判定しない																					
1	0	奇数パリティを出力	奇数パリティとして判定を行う																					
1	1	偶数パリティを出力	偶数パリティとして判定を行う																					
5	URTJnTDL	<p>送信データ・レベル制御</p> <p>0: 送信データ出力反転無効 1: 送信データ出力反転有効</p> <ul style="list-style-type: none"> このビットを使って URTJnTTXD 端子の出力レベルを反転させることができます。このビットをセット (1) すると、URTJnCTL0.URTJnPW と URTJnCTL0.URTJnTXE の値に関係なく、ただちに URTJnTTXD の出力レベルを反転します。したがって、動作禁止時に URTJnTDL をセット (1) すると、URTJnTTXD はロウ・レベルを出力します。 このビットは、送信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnTXE = 0) にのみ変更できます。 																						
4	URTJnRDL	<p>受信データ・レベル制御</p> <p>0: 受信データ入力反転無効 1: 受信データ入力反転有効</p> <ul style="list-style-type: none"> このビットを使って URTJnTRXD 端子の入力レベルを反転させることができます。このビットをセット (1) すると、URTJnCTL0.URTJnPW と URTJnCTL0.URTJnRXE の値に関係なく、ただちに URTJnTRXD の入力レベルを反転します。したがって、動作禁止時に URTJnRDL をセット (1) すると、URTJnTRXD にはロウ・レベルが入力されます。 このビットは、受信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnRXE = 0) にのみ変更できます。 																						

図 4.10 URTJnCTL1 レジスタのフォーマット (2/3)

表 20-8 URTJnCTL1 レジスタの内容 (3/3)

ビット位置	ビット名	機能
2	URTJnSLG	送信データのストップ・ビット数の選択 0 1ビット 1 2ビット <ul style="list-style-type: none"> データまたは BF の受信時、ストップ・ビット長は常に 1 として処理されます。 このビットは、送信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnTXE = 0) にのみ変更できます。
1	URTJnSLD	転送方向選択 0: MSB ファースト転送 1: LSB ファースト転送 <ul style="list-style-type: none"> LIN フォーマットで送信/受信を行うときは、URTJnSLD をセット (1) してください。 このビットは、送受信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnRXE, URTJnCTL0.URTJnTXE = 0) にのみ変更できます。
0	URTJnSLIT	送信割り込み要求 (INTUAJnTIT) 発生タイミング選択 0: 送信データが送信シフト・レジスタに格納され、送信を開始する際に INTUAJnTIT を発生します。 1: 送信完了時に INTUAJnTIT を発生します。 <ul style="list-style-type: none"> このビットは、送信禁止時 (URTJnCTL0.URTJnPW = 0 または URTJnCTL0.URTJnTXE = 0) にのみ変更できます。

図 4.11 URTJnCTL1 レジスタのフォーマット (3/3)

設定例 :

```

URTJnCTL1 = 0x5103;      /* Disable BF reception during data reception */
                          /* Receive/transmit data bit length: 8 bits */
                          /* No parity */
                          /* Transmit data output: No inversion, receive data input: No inversion */
                          /* Transmit data stop bit count: 1 bit */
                          /* Transfer direction select: LSB first transfer */
                          /* Transmit interrupt request generated at the end of transmission */

```

4.2.5 FIFO 制御レジスタ 0 (URTJnFCTL0)

URTJnFCTL0 レジスタは、受信割り込み要求 (INTUAJnTIR) と送信割り込み要求 (INTUAJnTIT) が発生したときの受信 FIFO と送信 FIFO の格納データ数を指定します。

URTJnFCTL0. URTJnSLRP[3:0]を設定することで、受信 FIFO の格納データ数が設定した値に達したときに受信割り込み要求が発生します。URTJnFCTL0. URTJnSLTP[3:0]を設定することで、送信 FIFO の格納データ数が設定した値に達したときに送信割り込み要求が発生します。

受信 FIFO と送信 FIFO の格納データ数はボー・レートに応じて適切な値を設定することが必要です。サンプルコードではそれぞれ 16 バイトに指定しています。

アクセス 16 ビット単位でリード/ライト可能です。

アドレス <URTJn_base_USER> + 80_H

初期値 0F00_H どのリセット要因でも初期化されます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	URTJnSLRP[3:0]				0	0	0	0	URTJnSLTP[3:0]			
R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

表 20-14 URTJnFCTL0 レジスタの内容

ビット位置	ビット名	機能
11-8	URTJnSLRP[3:0]	受信 FIFO レベル割り込み設定 URTJnSLRP[3:0] では、受信割り込み要求 INTUAJnTIR を発生するときの受信 FIFO ポインタのステータスを指定します。 !URTJnFSTR0.URTJnSSRW[4:0] の値 = (10 _H - URTJnSLRP[3:0] の値) のとき、つまり、10 _H - URTJnSLRP[3:0] 個の読み出し可能なデータ・ワードが受信 FIFO に残っているとき INTUAJnTIR は発生します。
3~0	URTJnSLTP[3:0]	送信 FIFO レベル割り込み設定 URTJnSLTP[3:0] では、送信割り込み要求 INTUAJnTIT が発生するときの送信 FIFO ポインタのステータスを指定します。 INTUAJnTIT は以下の場合に発生します。 URTJnFSTR0.URTJnSSTW[4:0] の値 = (10 _H - URTJnSLTP[3:0] の値) のとき、つまり、10 _H - URTJnSLTP[3:0] 個の数の書き込み可能な空きワードが送信 FIFO に残ります。

図 4.12 URTJnFCTL0 レジスタのフォーマット

設定例：

```
URTJnFCTL0 = 0x0000;          /* Receive FIFO interrupt level: 16 bytes */
                               /* Transmit FIFO interrupt level: 16 bytes */
```

4.2.6 FIFO 制御レジスタ 1 (URTJnFCTL1)

URTJnFCTL1 レジスタは、受信タイムアウトの検出を制御します。

受信 FIFO が空いていない、かつ一定の時間にわたって受信データが受信 FIFO に格納されていないか、受信 FIFO からデータが読み出されていない場合にタイムアウト・エラーが発生します。

アクセス 8/1 ビット単位でリード/ライト可能です。

アドレス <URTJn_base_OS> + 1020_H

初期値 3F_H どのリセット要因でも初期化されます。

7	6	5	4	3	2	1	0
0	0	URTJnSLRT[5:0]					
R	R	R/W	R/W	R/W	R/W	R/W	R/W

表 20-15 URTJnFCTL1 レジスタの内容

ビット位置	ビット名	機能
5-0	URTJnSLRT[5:0]	受信タイムアウト制御 00 _H : タイムアウトの検出を禁止します。 01 _H -3F _H : タイムアウト時間 = ポー・レート・クロック BRCLK の (URTJnSLRT[5:0] x 8) 周期

図 4.13 URTJnFCTL1 レジスタのフォーマット

設定例 :

```
URTJnFCTL1 = 0x3F;          /* Detect timeout */
```

4.2.7 UARTJn ステータス・クリア・レジスタ 0 (URTJnSTC)

データー貫性エラー、パリティ・エラー、フレーミング・エラー、オーバーラン・エラーを発生する共に URTJnSTR1 中の対応エラー・フラグがセットされます。URTJnSTC レジスタを用いて、URTJnSTR1 の各エラー・フラグをクリアすることができます。

このサンプルコードでは INTUAJnTIS が検出された場合、URTJnSTC レジスタと URTJnFSTC レジスタを用いて、エラー・フラグをクリアします。

アクセス 8/1 ビット単位でリード/ライト可能です。
このレジスタを読み出すと、常に 00_H が返ります。

アドレス <URTJn_base_USER> + 10_H

初期値 00_H どのリセット要因でも初期化されます。

7	6	5	4	3	2	1	0
0	0	0	URTJn CLBS	URTJn CLDC	URTJn CLP	URTJn CLF	0
R	R	R	R/W	R/W	R/W	R/W	R/W

表 20-13 URTJnSTC レジスタの内容

ビット位置	ビット名	機能
4	URTJnCLBS	BF 受信成功フラグ・クリア 0: 0 の書き込みは無視されます。 1: 1 を書き込むと URTJnSTR1.URTJnBSF がクリア (0) されます。
3	URTJnCLDC	データー貫性エラー・フラグ・クリア 0: 0 の書き込みは無視されます。 1: 1 を書き込むと URTJnSTR1.URTJnDCE がクリア (0) されます。 このビットをセット (1) することで URTJnDCE をクリア (0) すると、保留されているデータや BF の送信要求はすべて無視されます。
2	URTJnCLP	パリティ・エラー・フラグ・クリア 0: 0 の書き込みは無視されます。 1: 1 を書き込むと URTJnSTR1.URTJnPE がクリア (0) されます。
1	URTJnCLF	フレーミング・エラー・フラグ・クリア 0: 0 の書き込みは無視されます。 1: 1 を書き込むと URTJnSTR1.URTJnFE がクリア (0) されます。

図 4.14 URTJnSTC レジスタのフォーマット

設定例 :

```
URTJnSTC = 0x1f;          /* Clear receive error flags */
                          /* Clear transmit/receive data consistency error flag */
                          /* Clear receive parity error flag */
                          /* Clear receive framing error flag */
                          /* Clear receive overrun error flag */
```

4.2.8 FIFO ステータス・クリア・レジスタ (URTJnFSTC)

タイムアウト・エラー、オーバーフロー・エラー、オーバーラン・エラーが発生した場合、URTJnFSTR1 中の対応エラー・フラグがセットされます。URTJnFSTC レジスタを用いて、URTJnFSTR1 中の各エラー・フラグをクリアすることができます。送信/受信 FIFO ポインタをクリアすることもできます。

このサンプルコードでは INTUAJnTIS が検出された場合、URTJnSTC レジスタと URTJnFSTC レジスタを用いて、エラー・フラグをクリアします。

アクセス 8/1 ビット単位でリード/ライト可能です。
本レジスタを読み出すと、常に 00_H が返ります。

アドレス <URTJn_base_USER> + 8C_H

初期値 00_H どのリセット要因でも初期化されます。

7	6	5	4	3	2	1	0
URTJn CLTM	URTJn CLTO	URTJn CLRO	0	0	0	URTJn CLTP	URTJn CLRP
R/W	R/W	R/W	R	R	R	R/W	R/W

表 20-18 URTJnFSTC0 レジスタの内容

ビット位置	ビット名	機能
7	URTJn CLTM	タイムアウト・エラー・フラグ・クリア 0: 読み出し値は常に 0 になり、0 の書き込みは無視されます。 1: 1 を書き込むと URTJnFSTR1.URTJnTMOE = 0 に設定されます。
6	URTJn CLTO	送信 FIFO オーバフロー・エラー・フラグ・クリア 0: 読み出し値は常に 0 になり、0 の書き込みは無視されます。 1: 1 を書き込むと URTJnFSTR1.URTJnTOFE = 0 に設定されます。
5	URTJn CLRO	受信 FIFO オーバラン・エラー・フラグ・クリア 0: 読み出し値は常に 0 になり、0 の書き込みは無視されます。 1: 1 を書き込むと URTJnFSTR1.URTJnROVE = 0 に設定されます。
1	URTJn CLTP	送信 FIFO ポインタ・クリア 0: 読み出し値は常に 0 になり、0 の書き込みは無視されます。 1: 1 を書き込むと送信 FIFO ポインタが 00 _H に設定され、以下の状態になります。 - URTJnFSTR0.URTJnSSTW[4:0] = 00 _H (送信 FIFO ポインタ) - URTJnFSTR1.URTJnTOFE = 0 (送信 FIFO オーバフロー・エラーなし) - URTJnFSTR1.URTJnSSTF = 0 (送信 FIFO フルではない) - URTJnFSTR1.URTJnSSTE = 1 (送信 FIFO エンプティ)
0	URTJn CLRP	受信 FIFO ポインタ・クリア 0: 読み出し値は常に 0 になり、0 の書き込みは無視されます。 1: 1 を書き込むと受信 FIFO ポインタが 00 _H に設定され、以下の状態になります。 - URTJnFSTR0.URTJnSSRW[4:0] = 00 _H (受信 FIFO ポインタ) - URTJnFSTR1.URTJnROVE = 0 (受信 FIFO オーバラン・エラーなし) - URTJnFSTR1.URTJnSSRF = 0 (受信 FIFO フルではない) - URTJnFSTR1.URTJnSSRE = 1 (受信 FIFO エンプティ)

図 4.15 URTJnFSTC レジスタのフォーマット

設定例 :

```
URTJnFSTC = 0xe3;      /* Clear receive error flags */  
                        /* Clear timeout error flag */  
                        /* Clear transmit FIFO overflow error flag */  
                        /* Clear receive FIFO overrun error flag */  
                        /* Clear transmit FIFO pointer */  
                        /* Clear receive FIFO pointer */
```

4.3 関数仕様

この節で、サンプルコードを使用する関数の仕様について説明します。

4.3.1 メイン (main.c)

【関数名】	main()
【機能】	各初期化処理関数を呼び出したあと、永久ループに入る
【引数】	なし
【戻り値】	なし
【起動方法】	ハードウェア初期化完了したあと自動的にメイン関数に入る
【使用 SFR】	なし
【call 関数】	なし
【変数】	flag_transmit_over, flag_receive_over
【ファイル名】	main.c
【注意事項】	なし

4.3.2 初期化処理 (initial.c)

【関数名】	port_initial()
【機能】	ポートのモード設定を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	PFCE4, PFC4, PMC4, PM4, PFCE13, PFC13, PMC13, PM13
【call 関数】	main()
【変数】	なし
【ファイル名】	initial.c
【注意事項】	なし

【関数名】	cg_initial()
【機能】	特殊クロック周波数制御レジスタの初期化設定を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	SFRCTL3
【call 関数】	main()
【変数】	なし
【ファイル名】	initial.c
【注意事項】	なし

【関数名】 hbus_initial()
【機能】 AHB バスの初期化設定を行う
【引数】 なし
【戻り値】 なし
【起動方法】 コールする
【使用 SFR】 ETARCFG0, ETARADRS0, ETARMASK0
【call 関数】 main()
【変数】 なし
【ファイル名】 initial.c
【注意事項】 なし

【関数名】 board_initial()
【機能】 LED 初期状態の設定を行う
【引数】 なし
【戻り値】 なし
【起動方法】 コールする
【使用 SFR】 P13
【call 関数】 main()
【変数】 なし
【ファイル名】 initial.c
【注意事項】 なし

【関数名】 ram_initial()
【機能】 受信バッファと各フラグ初期状態の設定を行う
【引数】 なし
【戻り値】 なし
【起動方法】 コールする
【使用 SFR】 なし
【call 関数】 main()
【変数】 txdata_urtj1[], rxdata_urtj3[], flag_transmit_over, flag_receive_over, flag_fifo_error, flag_uart_error
【ファイル名】 initial.c
【注意事項】 なし

【関数名】 wait()
【機能】 一定ステップを待つ
【引数】 int number
【戻り値】 なし
【起動方法】 引数設定でコールする
【使用 SFR】 なし
【call 関数】 main(), int_urtj1ire(), int_urtj3ire()
【変数】 なし
【ファイル名】 initial.c
【注意事項】 なし

【関数名】	display()
【機能】	各フラグの状態により、LED を制御する
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	P13
【call 関数】	int_urtj3ir(),int_urtj3ire(), int_urtj1ire(),int_urtj1ic()
【変数】	flag_transmit_over, flag_receive_over, flag_error
【ファイル名】	initial.c
【注意事項】	なし

4.3.3 受信 (uartj_receive.c)

【関数名】	urtj3_receive_initial()
【機能】	UARTJ3 マクロの初期化設定を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	URTJ3CTL0, URTJ3CTL1, URTJ3CTL2, URTJ3FCTL0, URTJ3FCTL1, ICURTJ3IR, ICURTJ3IS
【call 関数】	main()
【変数】	なし
【ファイル名】	uartj_receive.c
【注意事項】	なし

4.3.4 送信 (uartj_transmit.c)

【関数名】	urtj1_transmit_initial()
【機能】	UARTJ1 マクロの初期化設定を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	URTJ1CTL0, URTJ1CTL1, URTJ1CTL2, URTJ1FCTL0, URTJ1FCTL1, ICURTJ1IC, ICURTJ1IS
【call 関数】	main()
【変数】	なし
【ファイル名】	uartj_transmit.c
【注意事項】	なし

【関数名】	urtj1_transmit_start()
【機能】	UARTJ1 マクロの送信動作を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	URTJ1FTX
【call 関数】	main()
【変数】	txdata_urtj1[]
【ファイル名】	uartj_transmit.c
【注意事項】	なし

4.3.5 割り込み (interrupt.c)

【関数名】	int_urtj1ic ()
【機能】	UARTJ1 マクロの送信割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、INTUAJ1IC 要求がある
【使用 SFR】	なし
【call 関数】	なし
【変数】	flag_transmit_over
【ファイル名】	interrupt.c
【注意事項】	なし

【関数名】	int_urtj1ire ()
【機能】	UARTJ1 マクロのステータス割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、INTUAJ1IS 要求がある
【使用 SFR】	URTJ1STR1, URTJ1FSTR1, URTJ1STC, URTJ1FSTC, URTJ1CTL0, URTJ3FSTR0, URTJ3STC, URTJ3FSTC, URTJ3CTL0, ICURTJ1IC, ICURTJ1IS, ICURTJ3IR, ICURTJ3IS, P13
【call 関数】	なし
【変数】	flag_uart_error, flag_fifo_error, flag_receive_over, flag_transmit_over rxdata_urtj3[]
【ファイル名】	interrupt.c
【注意事項】	なし

【関数名】	int_urtj3ir ()
【機能】	UARTJ3 マクロの受信割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、INTUAJ3IR 要求がある
【使用 SFR】	URTJ3FSTR0, URTJ3FRX
【call 関数】	なし
【変数】	rxdata_urtj3[], flag_receive_over
【ファイル名】	interrupt.c
【注意事項】	なし

【関数名】	int_urtj3ire ()
【機能】	UARTJ3 マクロのステータス割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、INTAJ3IS 要求がある
【使用 SFR】	URTJ3STR1, URTJ3FSTR1, URTJ1STC, URTJ1FSTC, URTJ1CTL0, URTJ3FSTR0, URTJ3STC, URTJ3FSTC, URTJ3CTL0, ICURTJ1IC, ICURTJ1IS, ICURTJ3IR, ICURTJ3IS, P13
【call 関数】	なし
【変数】	flag_uart_error, flag_fifo_error, flag_receive_over, flag_transmit_over, rxdata_urtj3[]
【ファイル名】	interrupt.c
【注意事項】	なし

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/>

お問い合わせ先

<http://japan.renesas.com/inquiry>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.02.07	—	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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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て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用途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>