

## RX62T グループ

DTC のブロック転送による CRC コード算出と

R01AN0635JJ0100

Rev.1.00

DTC のデータ転送による CRC コード付き調歩同期式シリアル通信

2011.07.15

### 要旨

本アプリケーションノートでは、データトランスファコントローラ (DTC) のブロック転送を用いた、CRC 演算器 (CRC) へのシリアルデータ転送例を紹介します。なお、本アプリケーションノートでは、シリアルコミュニケーションインターフェース (SCI) のデータ転送にも DTC を使用し、CRC コード付き調歩同期式シリアル通信を行います。

### 対象デバイス

#### RX62T グループ

RX62T グループと同様の I/O レジスタ (周辺装置制御レジスタ) を持つ他の RX ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートのご使用に際しては十分な評価を行ってください。

### 目次

|                   |    |
|-------------------|----|
| 1. 仕様 .....       | 2  |
| 2. 動作確認環境 .....   | 4  |
| 3. 使用機能 .....     | 4  |
| 4. 動作説明 .....     | 5  |
| 5. ソフトウェア説明 ..... | 11 |
| 6. 参考ドキュメント ..... | 26 |

## 1. 仕様

本アプリケーションノートでは、CRC コード付き送信データの作成、CRC コード付き受信データの CRC 演算および、CRC コード付きシリアル通信に DTC 転送を使用した 130 バイトの送受信を行います。

- CRC 生成多項式は  $X^{16}+X^{15}+X^2+1$  を使用し、LSB ファースト通信用 CRC コード生成を行います。
- SCI のチャンネル 0 を使用し、通信フォーマットは 8 ビット長、1 ストップビット、パリティなしです。
- 通信ビットレートは 38400bps とします。
- DTC は、ICU のソフトウェア割り込み (SWINT)、SCI0 の送信データエンプティ割り込み (TXI0)、SCI0 の受信データフル割り込み (RXI0) を使用します。
- 内蔵 RAM には、あらかじめ CRC 演算器の初期化データ (H'82) を用意しておきます。
- CRC 初期化データ (H'82) を CRC 演算器に転送することで、CRC 生成多項式の設定、LSB ファースト通信用 CRC コード生成の設定、CRC データ出力レジスタ (CRCDOR) のクリアを行います。
- 内蔵 RAM の送信データ格納領域には、あらかじめ 128 バイトの送信データを用意しておきます。
- 128 バイトの送信データは、H'00、H'01、H'02、・・・、H'7D、H'7E、H'7F の連続したデータとします。

### (1) 送信動作

CPU で SWINT 割り込みを発生させ、DTC を起動します。

DTC 転送で、CRC 演算器の初期化、CRC コードの生成、送信データへの CRC コード追加 (計 130 バイト) を行います。

SWINT 割り込みによる DTC 転送終了後、CPU の SWINT 割り込みで SCI0 の送信データエンプティ割り込み (TXI0) を許可し、DTC を起動します。

DTC 転送で、CRC コード付き送信データ (計 130 バイト) を送信します。

送信完了後、DTC (送信) の再設定を行います。

に戻り、送信動作を繰り返します。

### (2) 受信動作

SCI0 の受信データフル割り込み (RXI0) を許可します。

RXI0 割り込みにより、DTC を起動します。

DTC 転送で、CRC コード付き受信データ (計 130 バイト) を内蔵 RAM に転送します。

130 バイト分のデータ転送終了後、チェーン転送で、CRC 演算器の初期化、CRC 演算結果算出、CRC 演算結果の退避を行います。

RXI0 割り込みによる DTC 転送終了後、CPU の RXI0 割り込みで CRC エラー判定を行います。

CRC エラーの場合は受信動作を停止し、ポート 71 に接続された LED を点灯させ、エラー発生を通知します。CRC エラーがなければ DTC (受信) の再設定を行います。

に戻り、受信動作を繰り返します。

図 1 に本アプリケーションノートにおける仕様を示します。

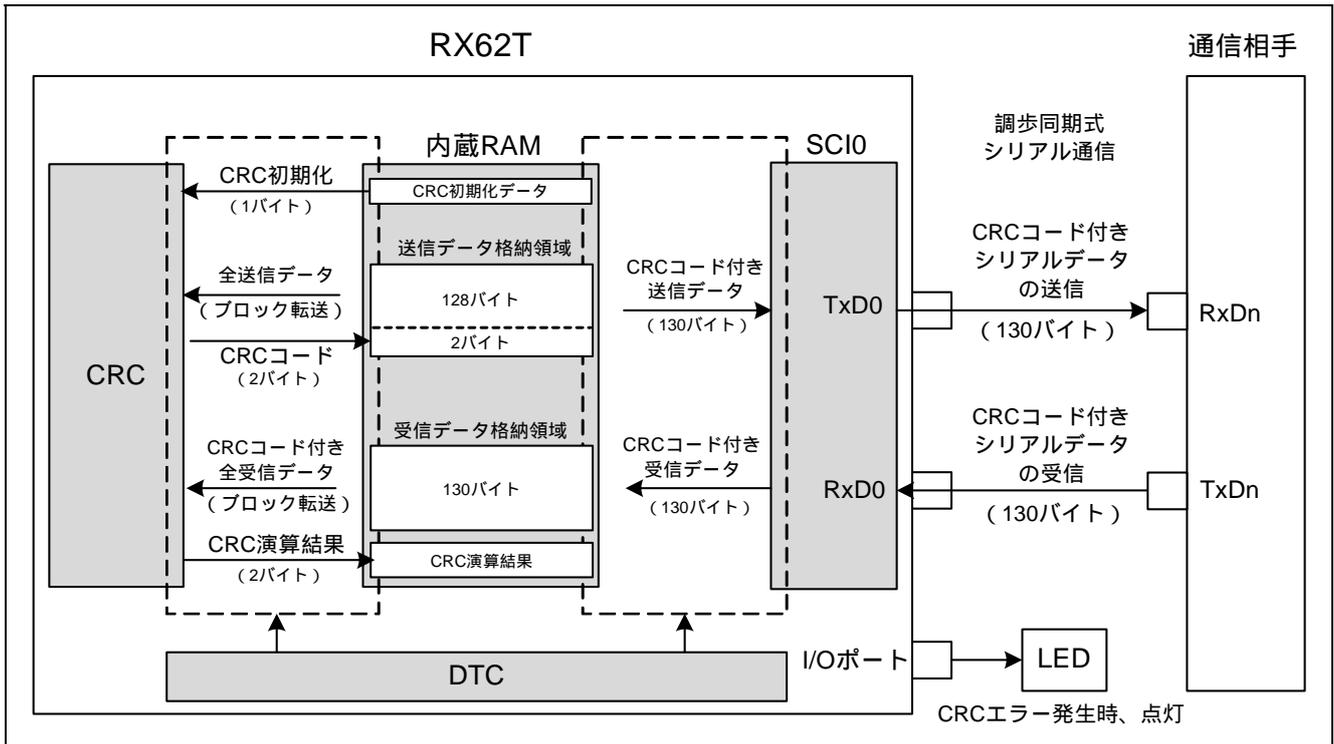


図 1 仕様

## 2. 動作確認環境

動作確認を行った環境を表 1 に示します。

表 1 動作確認環境

| 項目                 | 内容                                    |
|--------------------|---------------------------------------|
| デバイス               | RX62T ( R5F562TAADFP )                |
| ボード                | Renesas Starter Kit (R0K5562T0S000BE) |
| 電源電圧               | 5.0V                                  |
| 入力クロック             | 12.5MHz (ICLK=100MHz, PCLK=50MHz)     |
| 動作温度               | 室温                                    |
| HEW                | Version 4.08.00.011                   |
| Toolchain          | RX Standard Toolchain (V.1.0.1.0)     |
| Debugger/Emulator  | E20 エミュレータ                            |
| Debugger component | RX E1/E20 SYSTEMV.1.01.00             |

## 3. 使用機能

- クロック発生回路
- 消費電力低減機能
- I/O ポート
- 割り込みコントローラ (ICU)
- シリアルコミュニケーションインタフェース (SCI)
- CRC 演算器 (CRC) \*
- データトランスファコントローラ (DTC)

【注】 \* CRC 演算器は、CRC 演算中または CRC 演算結果を退避するまでの間に、CRCDIR レジスタに他のデータがライトされると (例えば 2 つ以上の異なるバスマスタによる CRC 演算器を使用した場合に) 演算結果に影響しますのでご注意ください。本 APN の場合、DTC で CRC 演算器にアクセス中に DMAC から CRC 演算器をアクセスした場合、演算結果が破壊されますので、DMAC で CRC 演算器をアクセスしないようにしてください。なお複数の DTC による CRC 演算器のアクセスは問題ありません。

詳細は「RX62T グループ ユーザーズマニュアル ハードウェア編」を参照してください。

## 4. 動作説明

### 4.1 動作モードの設定

本アプリケーションノートでは、モード端子を MD1=1、MD0=1 に設定し動作モードをシングルチップモードに、システムコントロールレジスタ 0 (SYSCR0) の ROME ビットを 1 に設定し内蔵 ROM を有効に設定しています。

本アプリケーションノートにおける動作モードの設定を表 2 に示します。

表 2 動作モードの設定

| モード端子 |     | SYSCR0 レジスタ | 動作モード      | 内蔵 ROM |
|-------|-----|-------------|------------|--------|
| MD1   | MD0 | ROME        |            |        |
| 1     | 1   | 1           | シングルチップモード | 有効     |

【注】 SYSCR0 レジスタの ROME ビットの初期値は、SYSCR0.ROME=1 のため、プログラム中で SYSCR0 レジスタの設定は行っていません。

### 4.2 クロックの設定

本アプリケーションノートで使用している評価ボードには 12.5MHz の水晶発振子が搭載されています。

そのため、本アプリケーションノートでは、システムクロック (ICLK)、周辺モジュールクロック (PCLK) をそれぞれ、8 週倍 (100MHz)、4 週倍 (50MHz) に設定しています。

### 4.3 エンディアンの設定

本アプリケーションノートは、ビッグエンディアン/リトルエンディアンの両方に対応しています。ハードウェアによるエンディアンの設定を表 3 に示します。

表 3 エンディアン設定 (ハードウェア)

| MDE 端子 | エンディアン    |
|--------|-----------|
| 0      | リトルエンディアン |
| 1      | ビッグエンディアン |

コンパイラオプションのマイコンオプションによるエンディアンの設定を表 4 に示します。

表 4 エンディアン設定 (コンパイラオプション)

| マイコンオプション       | エンディアン    |
|-----------------|-----------|
| endian = little | リトルエンディアン |
| endian = big    | ビッグエンディアン |

【注】 プログラムのコンパイラオプションで選択したエンディアンに合わせて、MDE 端子によるエンディアンを設定してください。

#### 4.4 ビットオーダの設定

本アプリケーションノートのビットオーダは、ライトおよびレフトの両方に対応しています。コンパイラオプションのマイコンオプションによるビットオーダの設定を表 5 に示します。

表 5 ビットオーダ設定 (コンパイラオプション)

| マイコンオプション         | ビットオーダ                                  |
|-------------------|---|
| bit_order = right | ビットフィールドのメンバの並び順を下位ビットから割り付け (オプション省略時) |
| bit_order = left  | ビットフィールドのメンバの並び順を上位ビットから割り付け            |

【注】 本アプリケーションノートでは、ビットフィールドは I/O レジスタ定義ファイル (iodefine.h) で使用しています。I/O レジスタ定義ファイルでは、#pragma bit\_order 拡張子で left を指定しており、ビットフィールドのメンバの並び順を上位ビットから割り付けています。

【注】 コンパイラオプションの bit\_order と #pragma bit\_order 拡張子の両方が指定されている場合は、#pragma bit\_order 拡張子の指定が優先されるため、コンパイラオプションの bit\_order の指定に関係なく、I/O レジスタ定義ファイルで定義されたビットフィールドのメンバの並び順は上位ビットから割り付けられます。

#### 4.5 SCI の設定

本アプリケーションノートでは、SCI のチャンネル 0 を使用し調歩同期式シリアルデータの送受信を行います。SCI の設定条件を表 6 に示します。

表 6 SCI の設定条件

| 使用チャンネル | SCI 0  |
|---------|--|
| 通信モード   | 調歩同期式モード   |
| 割り込み    | <ul style="list-style-type: none"> <li>• 受信エラー割り込み (ERI0)</li> <li>• 受信データフル割り込み (RXI0)</li> <li>• 送信データエンプティ割り込み (TXI0)</li> <li>• 送信完了割り込み (TEI0)</li> </ul> |
| 通信速度    | 38400bps (PCLK = 50MHz)  |
| データ長    | 8 ビットデータ   |
| ストップビット | 1 ストップビット  |
| パリティ    | なし   |

#### 4.6 CRC の設定

本アプリケーションノートでは、CRC 演算器を使用し 2 バイトの CRC コード生成を行います。CRC の設定条件を表 7 に示します。

表 7 CRC の設定条件

|        |                        |
|--------|------------------------|
| 生成多項式  | $X^{16}+X^{15}+X^2+1$  |
| CRC 演算 | LSB ファースト通信用 CRC コード生成 |

## 4.7 DTC の設定

本アプリケーションノートでは、SCI のデータ転送、CRC コード付き送信データの作成、CRC コード付き受信データの CRC 演算に DTC を使用しています。DTC の設定条件（受信）を表 8 に、DTC の設定条件（送信）を表 9 に示します。

表 8 DTC の設定条件（受信）

| 条件           | CRC コード付き<br>受信データの<br>SCI 転送       | CRC 初期化                    | CRC コード付き<br>受信データの<br>CRC 転送   | 演算結果退避                              |
|--------------|-------------------------------------|----------------------------|---------------------------------|-------------------------------------|
| 転送モード        | ノーマル転送モード                           | ノーマル転送モード                  | ブロック転送モード                       | ノーマル転送モード                           |
| 転送回数         | 130 回                               | 1 回                        | 1 回                             | 1 回                                 |
| ブロックサイズ      | -                                   | -                          | 130 バイト                         | -                                   |
| 転送データ<br>サイズ | バイト (Byte)                          | バイト (Byte)                 | バイト (Byte)                      | ワード (Word)                          |
| 転送データ内容      | CRC コード付き<br>受信データ<br>(130 バイト)     | CRC 初期化データ<br>(1 バイト)      | CRC コード付き<br>受信データ<br>(130 バイト) | CRC 演算結果<br>(2 バイト)                 |
| 転送元          | SCI0 のレシーブ<br>データレジスタ<br>(SCI0.RDR) | 内蔵 RAM                     | 内蔵 RAM                          | CRC データ出力<br>レジスタ<br>(CRCDOR)       |
| 転送先          | 内蔵 RAM                              | CRC コントロール<br>レジスタ (CRCCR) | CRC データ入力<br>レジスタ (CRCDIR)      | 内蔵 RAM                              |
| 転送元アドレス      | 転送元は固定                              | 転送元は固定                     | 転送後に転送元アド<br>レスをインクリメン<br>ト     | 転送元は固定                              |
| 転送先アドレス      | 転送後に転送先アド<br>レスをインクリメン<br>ト         | 転送先は固定                     | 転送先は固定                          | 転送先は固定                              |
| 起動要因         | 受信データフル割り<br>込み (RXI0) で起動          | 受信データの SCI 転<br>送後に実行      | CRC 初期化後に実行                     | 受信データの CRC<br>転送後に実行                |
| チェーン         | 許可                                  | 許可                         | 許可                              | 禁止                                  |
| 割り込み         | なし                                  | なし                         | なし                              | 指定したデータ転送<br>終了後、CPU に対し<br>て割り込み許可 |

表 9 DTC の設定条件 (送信)

| 条件           | CRC 初期化                            | 送信データの<br>CRC 転送              | CRC 演算結果追加                          | CRC コード付き<br>送信データの SCI 転送             |
|--------------|------------------------------------|-------------------------------|-------------------------------------|--|
| 転送モード        | ノーマル転送モード                          | ブロック転送モード                     | ノーマル転送モード                           | ノーマル転送モード                              |
| 転送回数         | 1 回                                | 1 回                           | 1 回                                 | 130 回                                  |
| ブロックサイズ      | -                                  | 128 バイト                       | -                                   | -                                      |
| 転送データ<br>サイズ | バイト (Byte)                         | バイト (Byte)                    | ワード (Word)                          | バイト (Byte)                             |
| 転送データ内容      | CRC 初期化データ<br>(1 バイト)              | 送信データ<br>(128 バイト)            | CRC 演算結果<br>(2 バイト)                 | CRC コード付き<br>送信データ<br>(130 バイト)        |
| 転送元          | 内蔵 RAM                             | 内蔵 RAM                        | CRC データ出力レ<br>ジスタ (CRCDOR)          | 内蔵 RAM                                 |
| 転送先          | CRC コントロール<br>レジスタ (CRCCR)         | CRC データ入力<br>レジスタ<br>(CRCDIR) | 内蔵 RAM                              | SCI0 のトランスミッ<br>トデータレジスタ<br>(SCI0.TDR) |
| 転送元アドレス      | 転送元は固定                             | 転送後に転送元アド<br>レスをインクリメン<br>ト   | 転送元は固定                              | 転送後に転送先アドレ<br>スをインクリメント                |
| 転送先アドレス      | 転送先は固定                             | 転送先は固定                        | 転送先は固定                              | 転送先は固定                                 |
| 起動要因         | ICU ソフトウェア割<br>り込み (SWINT) で<br>起動 | CRC 初期化後に実<br>行               | 送信データの CRC<br>転送後に実行                | 送信データエンプティ<br>割り込み (TXI0) で起<br>動      |
| チェーン         | 許可                                 | 許可                            | 禁止                                  | 禁止                                     |
| 割り込み         | なし                                 | なし                            | 指定したデータ転送<br>終了後、CPU に対し<br>て割り込み許可 | 指定したデータ転送終<br>了後、CPU に対して割<br>り込み許可    |

## 4.8 動作詳細

## 4.8.1 受信動作

SCI0 の受信データフル割り込み (RXI0) 要求待ち。

RXI0 割り込み要求で DTC を起動します。

DTC で、CRC コード付き受信データ (計 130 バイト) を SCI0 のレシーブデータレジスタ (RDR) から内蔵 RAM に転送します。

130 バイト分のデータ転送終了後、チェーン転送で、内蔵 RAM の CRC 初期化データ (H'82) を CRC コントロールレジスタ (CRCCR) に転送し、CRC 演算器を初期化します。

CRC 演算器の初期化後、チェーン転送で、内蔵 RAM の CRC コード付き受信データ (計 130 バイト) を CRC データ入力レジスタ (CRCDIR) にブロック転送し、CRC 演算結果を算出します。

CRC データ出力レジスタ (CRCDOR) に算出された CRC 演算結果をチェーン転送で内蔵 RAM に退避します。

DTC 転送終了後、CPU への割り込み要求が発生します (RXI0 割り込み要求)。

RXI0 割り込みルーチン内で内蔵 RAM の CRC 演算結果をリードし、CRC エラー判定を行います。CRC エラーの場合は受信動作を停止し、ポート 71 に接続された LED を点灯させ、エラー発生を通知します。CRC エラーがなければ DTC (受信) の再設定を行い、 に戻ります。

図 2 に受信動作のブロック図を示します。

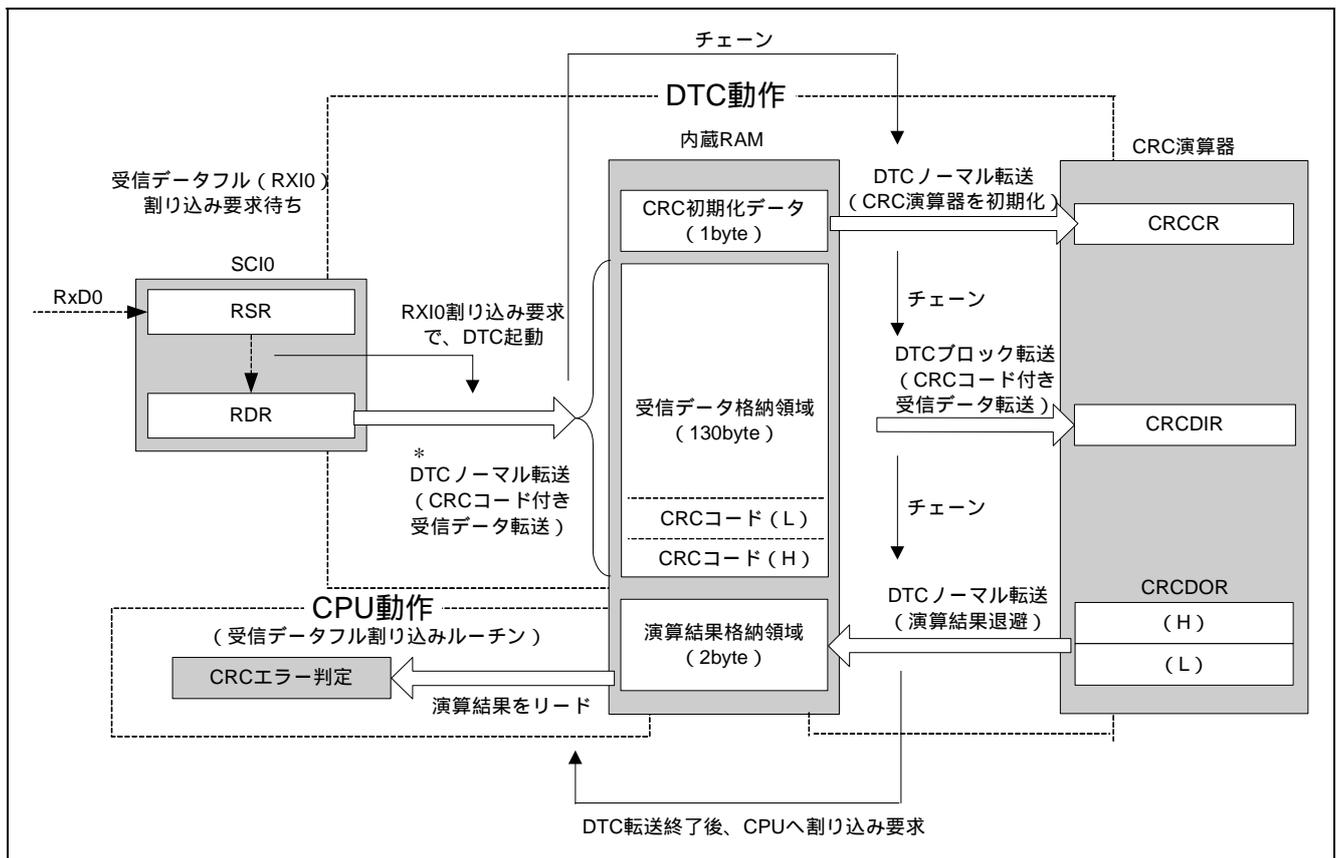


図 2 受信動作のブロック図

【注】 \* RX62T グループでは、DTC を通信機能と組み合わせて使用する場合、IR フラグの自動クリアまでの間に次の転送要求が発生すると、転送要求の消失が発生するので注意が必要です。詳細は、RX62T グループ ユーザーズマニュアル ハードウェア編「11.7 使用上の注意事項」を参照してください。

## 4.8.2 送信動作

ICU のソフトウェア割り込み起動レジスタ (SWINTR) の SWINT ビットに 1 をライトし、ソフトウェア割り込み (SWINT) 要求を発行します。

SWINT 割り込み要求により DTC を起動します。

DTC で、内蔵 RAM の CRC 初期化データ (H'82) を CRCCR に転送し、CRC 演算器を初期化します。

CRC 演算器の初期化後、チェーン転送で内蔵 RAM の送信データ (128 バイト) を CRCDIR にブロック転送し、2 バイトの CRC コードを生成します。

CRCDOR に生成された CRC コードをチェーン転送で送信データに追加し、130 バイトの CRC コード付き送信データを作成します。

DTC 転送終了後、CPU への割り込み要求が発生します (ICU の SWINT 割り込み要求)。

ICU の SWINT 割り込みルーチン内で、モードモニタレジスタ (MDMONR) の MDE ビットをリードし、MDE ビットが 1 (ビッグエンディアン) であれば、CRC コード (H) が送信データ格納領域の最終バイトに来るように CRC コード (L) と CRC コード (H) を入れ替えます。

SCI0 の送信データエンプティ割り込み (TXI0) を許可し、DTC を起動します。

DTC で、CRC コード付き送信データ (計 130 バイト) を SCI0 のトランスミットデータレジスタ (TDR) に転送します。130 バイト分のデータ転送が完了すると DTC (送信) の再設定を行い、 に戻ります。

図 3 に送信動作のブロック図を示します。

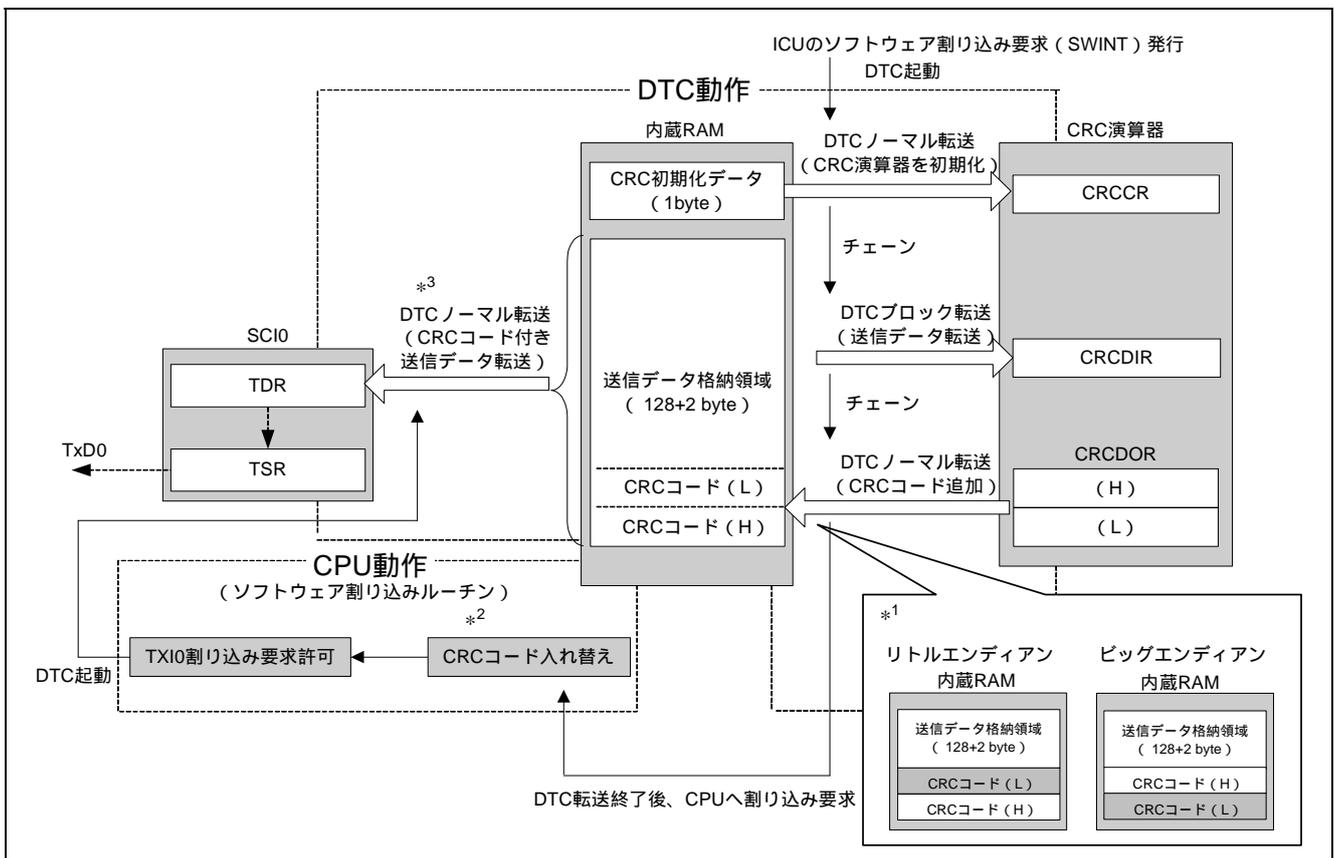


図 3 送信動作のブロック図

【注】 \*1 CRCDOR は 16 ビットレジスタであるため、DTC で内蔵 RAM にデータ転送するとエンディアン設定の影響を受け、リトルエンディアン時は CRC コード (H) が送信データ格納領域の最終バイトに、ビッグエンディアン時は CRC コード (L) が送信データ格納領域の最終バイトに配置されます。

\*2 エンディアン設定がビッグエンディアンのとき、CRC コードが反転して配置されるため CRC コードを入れ替える必要があります。本アプリケーションノートでは、CRC コード (H) が送信データ格納領域の最終バイトに来るように内蔵 RAM の CRC コード (L) と CRC コード (H) を入れ替え

る処理を実施しています。なお、本アプリケーションノートのサンプルソースをリトルエンディアンで使用する場合、本処理は不要です。

\*3 RX62T グループでは、DTC を通信機能と組み合わせて使用する場合、IR フラグの自動クリアまでの間に次の転送要求が発生すると、転送要求の消失が発生するので注意が必要です。

詳細は、RX62T グループ ユーザーズマニュアル ハードウェア編「11.7 使用上の注意事項」を参照してください。

## 5. ソフトウェア説明

### 5.1 定数一覧

表 10 にサンプルコードで使用する定数の一覧を示します。

表 10 定数一覧

| 定数名           | 設定値   | 内容                   |
|---------------|-------|----------------------|
| DTC_CNT       | 130   | DTC の転送回数            |
| BLOCK_SIZE_TX | 128   | DTC 転送 (送信) のブロックサイズ |
| BLOCK_SIZE_RX | 130   | DTC 転送 (受信) のブロックサイズ |
| BIG_ENDIAN    | 1     | ビッグエンディアン            |
| NON_CRC_ERR   | 0000h | CRC エラーなし            |
| SET           | 1     | フラグセット               |
| CLEAR         | 0     | フラグクリア               |

### 5.2 構造体 / 共用体一覧

図 4、図 5 にサンプルコードで使用する構造体 / 共用体を示します。

```

struct st_ram_data{
    union {
        unsigned char ALL[130];          /* CRCコード付きシリアルデータ */
        struct {
            unsigned char DATA[128];    /* シリアルデータ */
            unsigned short CRC_CODE;     /* CRCコード */
        }MEMBER;
    }BUFF;
};

```

図 4 サンプルコードで使用する構造体 / 共用体 (CRC コード付きシリアルデータ)

```

#pragma bit_order left
#pragma unpack

struct st_dtc_full{
  union{
    unsigned long LONG;
    struct{
      unsigned long MRA_MD :2; /* MRA.MDビット */
      unsigned long MRA_SZ :2; /* MRA.SZビット */
      unsigned long MRA_SM :2; /* MRA.SMビット */
      unsigned long :2;
      unsigned long MRB_CHNE :1; /* MRB.CHNEビット */
      unsigned long MRB_CHNS :1; /* MRB.CHNSビット */
      unsigned long MRB_DISEL :1; /* MRB.DISELビット */
      unsigned long MRB_DTS :1; /* MRB.DTSビット */
      unsigned long MRB_DM :2; /* MRB.DMビット */
      unsigned long :2;
      unsigned long :16;
    }BIT;
  }MR;
  void * SAR; /* SARレジスタ */
  void * DAR; /* DARレジスタ */
  struct{
    unsigned long CRA:16; /* CRAレジスタ */
    unsigned long CRB:16; /* CRBレジスタ */
  }CR;
};

#pragma bit_order
#pragma packoption

```

図 5 サンプルコードで使用する構造体 / 共用体 (DTC 転送情報)

【注】 「#pragma unpack」で DTC 転送情報のアライメント数を 4 としています。

### 5.3 変数一覧

表 11 にサンプルコードで使用する変数の一覧を示します。

表 11 変数一覧

| 型名             | 変数名              | 内容                              | 使用関数                        |
|----------------|------------------|---------------------------------|-----------------------------|
| unsigned char  | send_end_flag    | 送信終了フラグ 0: 送信中<br>1: 送信終了       | main, int_sci_tei0          |
| unsigned char  | receive_end_flag | 受信終了フラグ 0: 受信中/受信待ち<br>1: 送信終了  | main, int_sci_rxi0          |
| unsigned char  | crc_error_flag   | CRC エラーフラグ 0: エラーなし<br>1: エラー発生 | main, int_sci_rxi0          |
| unsigned char  | CRC_RESET_DATA   | CRC 初期化データ                      | main, dtc_send, dtc_receive |
| st_dtc_full    | dtc_tx_crc[3]    | DTC 転送情報 (送信: CRC 転送)           | dtc_send                    |
| st_dtc_full    | dtc_tx_sci       | DTC 転送情報 (送信: SCI 転送)           | dtc_send                    |
| st_dtc_full    | dtc_rx[4]        | DTC 転送情報 (受信)                   | dtc_receive                 |
| st_ram_data    | ram_tx           | CRC コード付き送信データ                  | main, dtc_send              |
| st_ram_data    | ram_rx           | CRC コード付き受信データ                  | main, dtc_receive           |
| unsigned short | CRC_RESULT       | CRC 演算結果                        | dtc_receive                 |
| void*          | dtc_table[256]   | DTC ベクタテーブル*                    | dtc_init                    |

【注】 \* アドレス 0x00000000 に配置しています。

## 5.4 関数一覧

表 12 にサンプルコードで使用する関数一覧を示します。

表 12 関数一覧

| 関数名             | 機能                           |
|-----------------|------------------------------|
| HardwareSetup   | 初期化処理、クロック設定、モジュールストップ状態解除   |
| main            | メイン処理<br>ICU の初期設定、割り込みレベル設定 |
| sci0_init       | SCI の初期設定、転送クロックの設定          |
| dtc_init        | DTC の初期設定                    |
| dtc_receive     | DTC 転送情報の配置（受信）              |
| dtc_send        | DTC 転送情報の配置（送信）              |
| crc_error       | CRC エラー表示                    |
| byte_reverse_16 | WORD データの上位バイトと下位バイト入れ替え     |
| int_icu_sw      | ICU ソフトウェア割り込み               |
| int_sci_txi0    | 送信データエンプティ割り込み               |
| int_sci_tei0    | 送信完了割り込み                     |
| int_sci_rxi0    | 受信データフル割り込み                  |
| int_sci_eri0    | 受信エラー割り込み                    |

## 5.5 関数仕様

以下にサンプルコードの関数仕様を示します。

---

`sci0_init`

---

|       |                                   |
|-------|-----------------------------------|
| 概要    | SCI0 の初期設定を行います。                  |
| ヘッダ   | <code>iodef.h</code>              |
| 宣言    | <code>void sci0_init(void)</code> |
| 説明    | 設定内容は 4.5 章の表 6 を参照してください。        |
| 引数    | なし                                |
| リターン値 | なし                                |
| 備考    |                                   |

---

`dtc_init`

---

|       |                                  |
|-------|----------------------------------|
| 概要    | DTC の初期設定を行います。                  |
| ヘッダ   | <code>iodef.h</code>             |
| 宣言    | <code>void dtc_init(void)</code> |
| 説明    | 設定内容は 4.7 章の表 8、表 9 を参照してください。   |
| 引数    | なし                               |
| リターン値 | なし                               |
| 備考    |                                  |

---

`dtc_receive`

---

|       |   |
|-------|---|
| 概要    | DTC 転送情報（受信）の設定を行います。                         |
| ヘッダ   | <code>iodef.h</code> , <code>dtc_def.h</code> |
| 宣言    | <code>void dtc_receive(void)</code>           |
| 説明    | 設定内容は 4.7 章の表 8 を参照してください。                    |
| 引数    | なし  |
| リターン値 | なし  |
| 備考    | DTC の初期設定および再設定時（受信）に使用します。                   |

---

dtc\_send

---

|       |                             |
|-------|-----------------------------|
| 概要    | DTC 転送情報（送信）の設定を行います。       |
| ヘッダ   | iodef.h,dtc_def.h           |
| 宣言    | void dtc_send(void)         |
| 説明    | 設定内容は 4.7 章の表 9 を参照してください。  |
| 引数    | なし                          |
| リターン値 | なし                          |
| 備考    | DTC の初期設定および再設定時（送信）に使用します。 |

---

crc\_error

---

|       |                              |
|-------|------------------------------|
| 概要    | CRC エラー表示を行います。              |
| ヘッダ   | iodef.h                      |
| 宣言    | void crc_error(void)         |
| 説明    | ポート 71 に接続されている LED を点灯させます。 |
| 引数    | なし                           |
| リターン値 | なし                           |
| 備考    | なし                           |

---

byte\_reverse\_16

---

|       |  |
|-------|--|
| 概要    | WORD データの上位バイトと下位バイトを入れ替えます。                               |
| ヘッダ   | iodef.h  |
| 宣言    | unsigned short byte_reverse_16(unsigned short rev_data)    |
| 説明    | 引数 rev_data ( unsigned short 型 ) の上位バイトと下位バイトを入れ替えた値を返します。 |
| 引数    | rev_data ( unsigned short 型 )                              |
| リターン値 | (rev_data << 8)   (rev_data >> 8)                          |
| 備考    | CRC コードの入れ替えに使用します。  |

---

int\_icu\_sw

---

|       |  |
|-------|--|
| 概要    | SWINT 割り込み関数   |
| ヘッダ   | iodef.h,vect.h   |
| 宣言    | void int_icu_sw (void)   |
| 説明    | <ul style="list-style-type: none"><li>MDMONR の MDE ビットをリードし、MDE ビットが 1 (ビッグエンディアン) の場合は byte_reverse_16 関数を呼び出します。</li><li>TXI0 割り込みを許可 (IEN = 1) します。</li></ul> |
| 引数    | なし   |
| リターン値 | なし   |
| 備考    | なし   |

---

int\_sci\_txi0

---

|       |   |
|-------|---|
| 概要    | TXI0 割り込み関数   |
| ヘッダ   | iodef.h,vect.h  |
| 宣言    | void int_sci_txi0(void)   |
| 説明    | <ul style="list-style-type: none"><li>TXI0 割り込みを禁止 (IEN = 0) します。</li><li>TEI0 割り込みを許可 (IEN = 1) します。</li></ul> |
| 引数    | なし  |
| リターン値 | なし  |
| 備考    | なし  |

---

int\_sci\_tei0

---

|       |  |
|-------|--|
| 概要    | TEI0 割り込み関数  |
| ヘッダ   | iodef.h,vect.h   |
| 宣言    | void int_sci_tei0(void)  |
| 説明    | <ul style="list-style-type: none"><li>TEI0 割り込みを禁止 (IEN = 0) します。</li><li>送信終了フラグ (send_end_flag) をセットします。</li></ul> |
| 引数    | なし   |
| リターン値 | なし   |
| 備考    | なし   |

---

`int_sci_rxi0`

---

|       |  |
|-------|--|
| 概要    | RXI0 割り込み関数  |
| ヘッダ   | <code>iodefine.h,vect.h</code>   |
| 宣言    | <code>void int_sci_rxi0(void)</code>   |
| 説明    | <ul style="list-style-type: none"><li>・RXI0 割り込みを禁止 ( IEN = 0 ) します。</li><li>・CRC エラー判定を行い、CRC エラーの場合は CRC エラーフラグ ( <code>crc_error_flag</code> ) をセットします。</li><li>・受信終了フラグ ( <code>receive_end_flag</code> ) をセットします。</li></ul> |
| 引数    | なし   |
| リターン値 | なし   |
| 備考    | なし   |

---

`int_sci_eri0`

---

|       |                                      |
|-------|--------------------------------------|
| 概要    | ERI0 割り込み関数                          |
| ヘッダ   | <code>iodefine.h,vect.h</code>       |
| 宣言    | <code>void int_sci_eri0(void)</code> |
| 説明    | 受信エラー処理を行います ( 本サンプルコードでは除く ) 。      |
| 引数    | なし                                   |
| リターン値 | なし                                   |
| 備考    | なし                                   |

5.6 処理フロー

図 6～図 19 にサンプルコードの処理フローを示します。

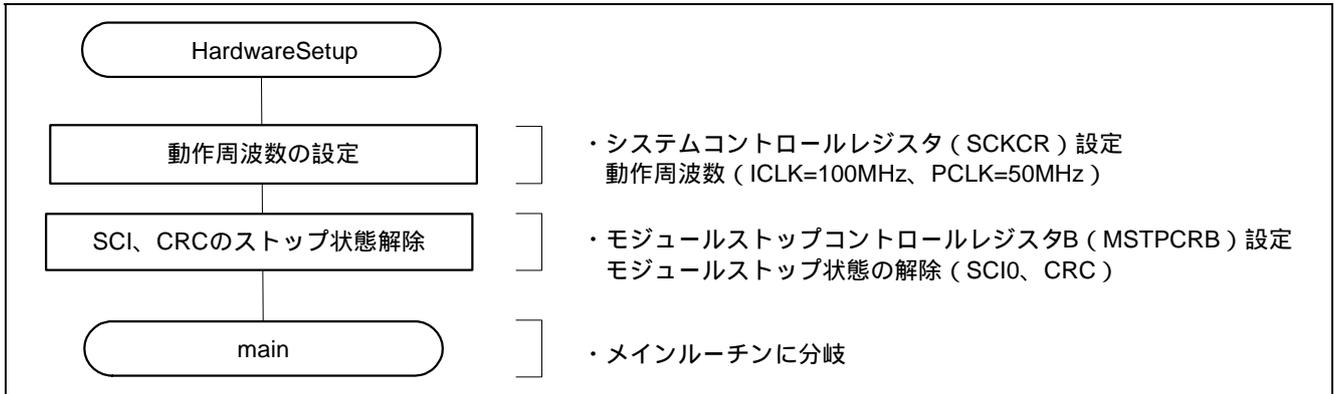


図 6 初期化処理

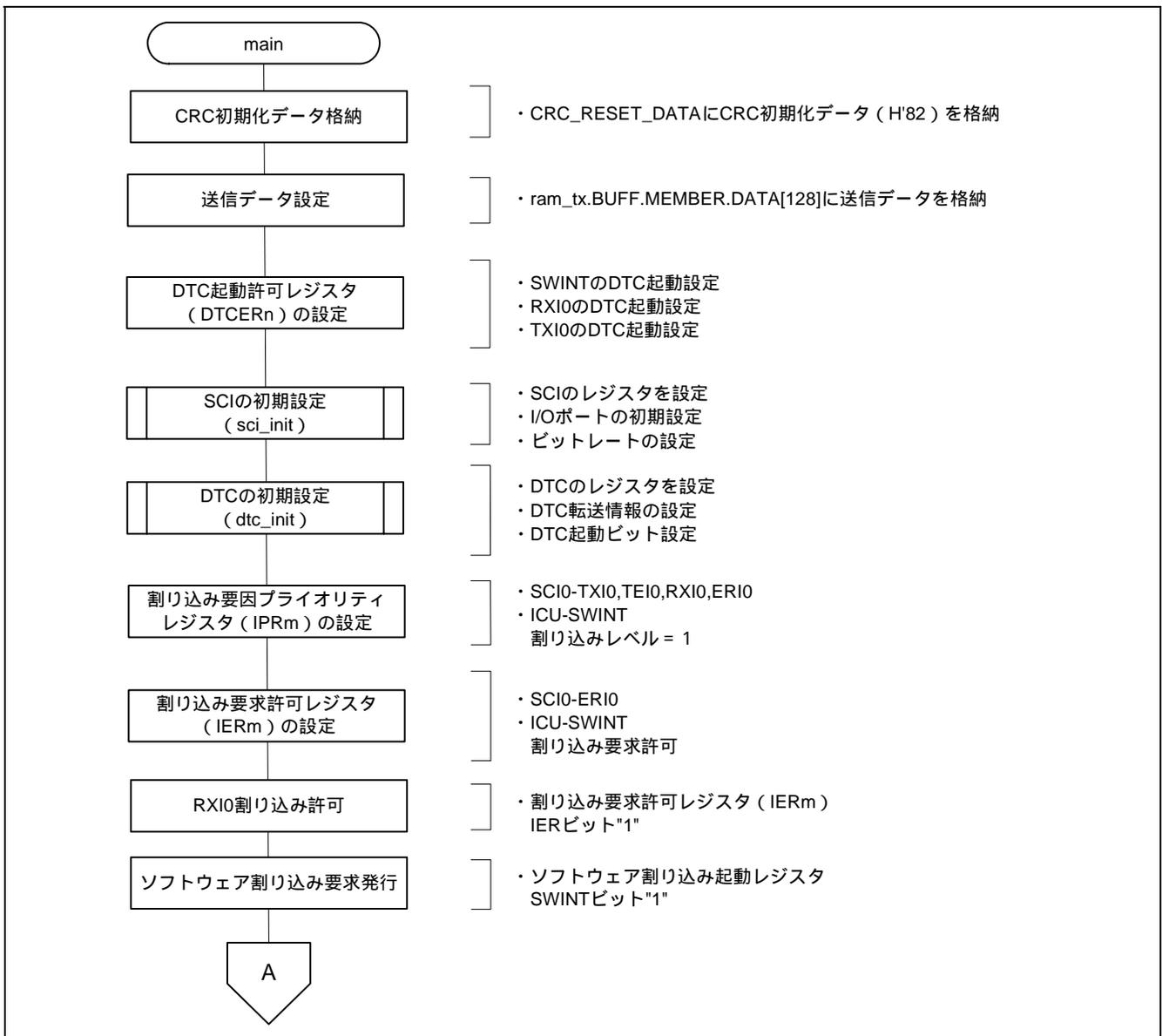


図 7 メイン処理 (1)

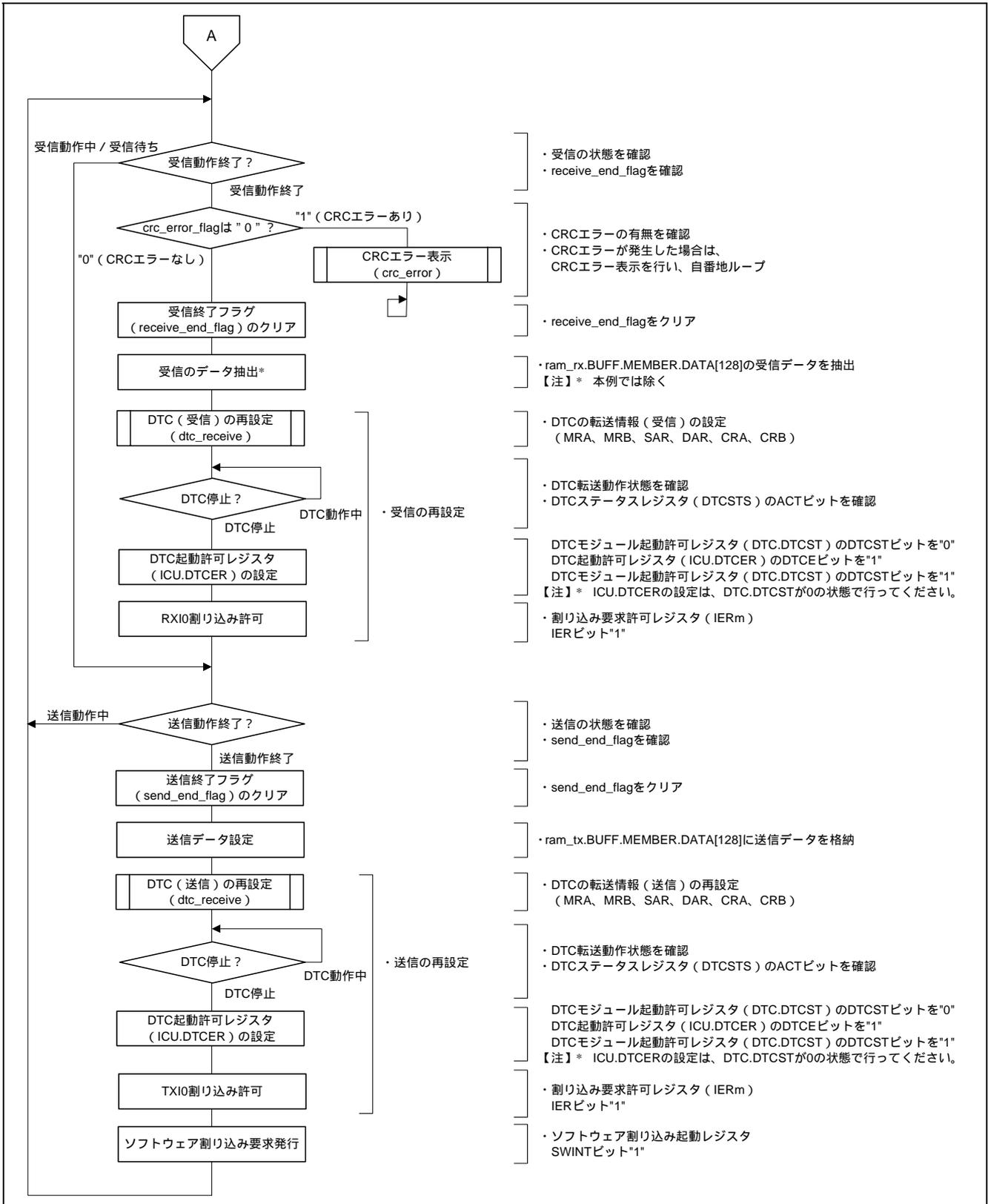


図 8 メイン処理 (2)

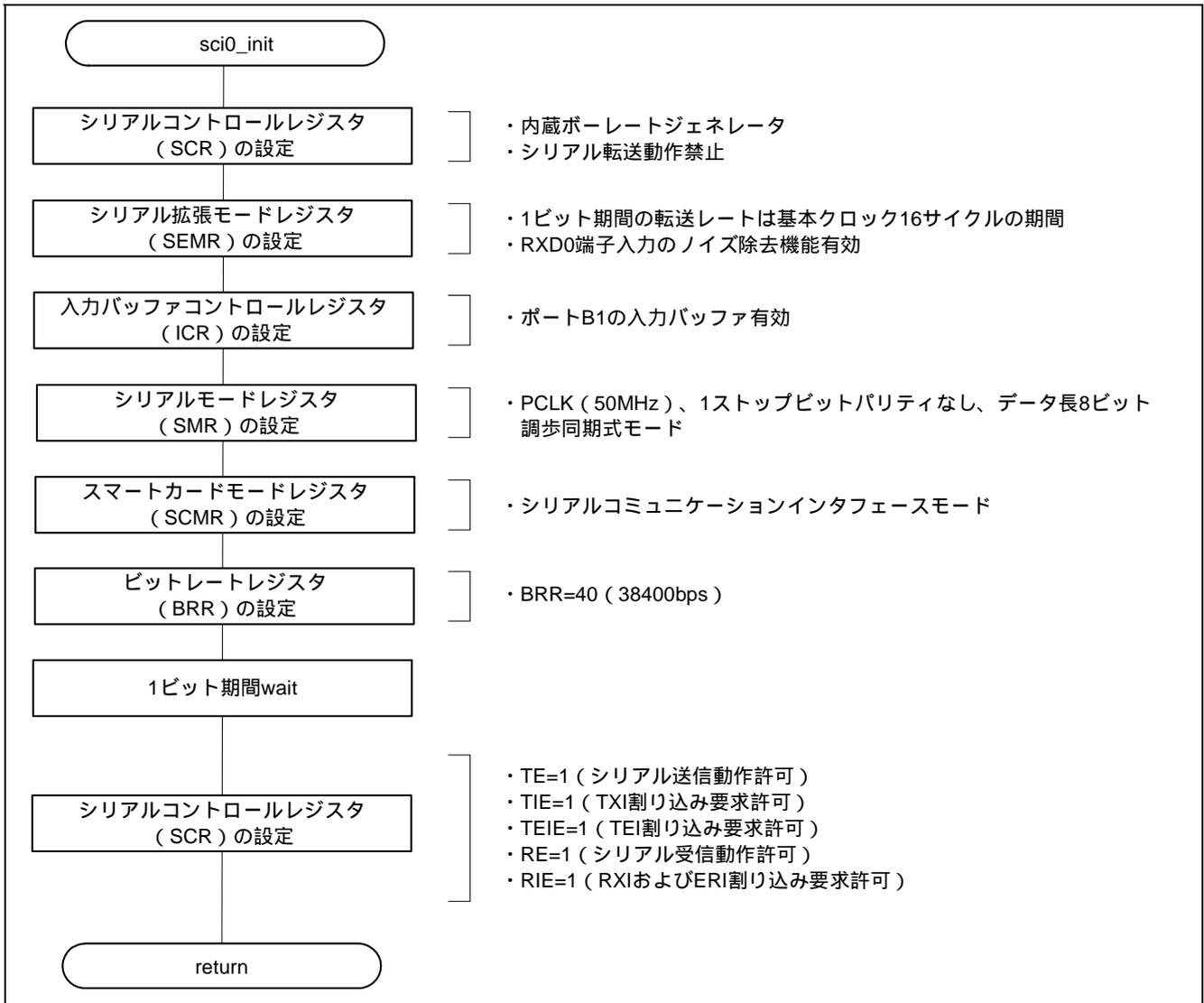


図 9 SCI の初期設定

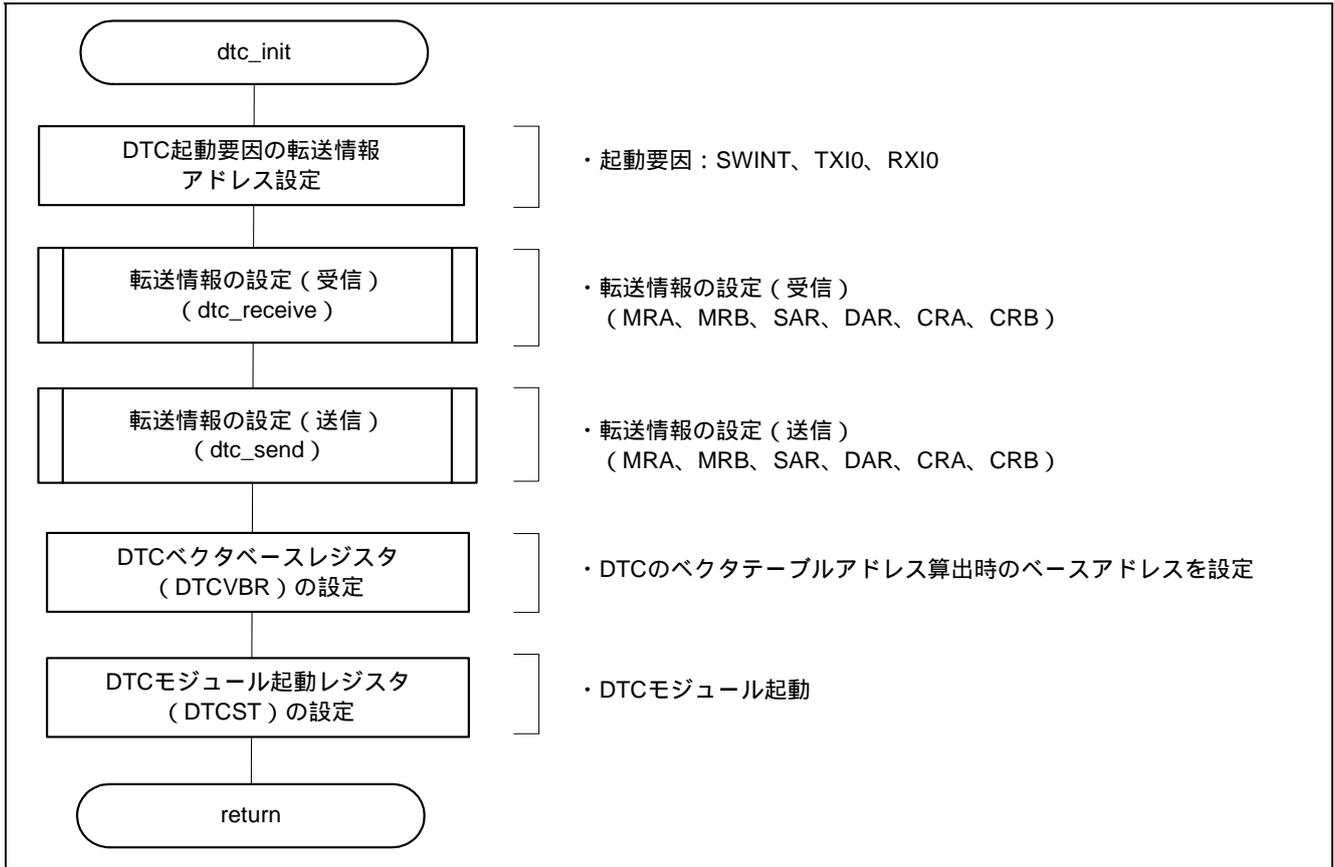


図 10 DTC 初期設定

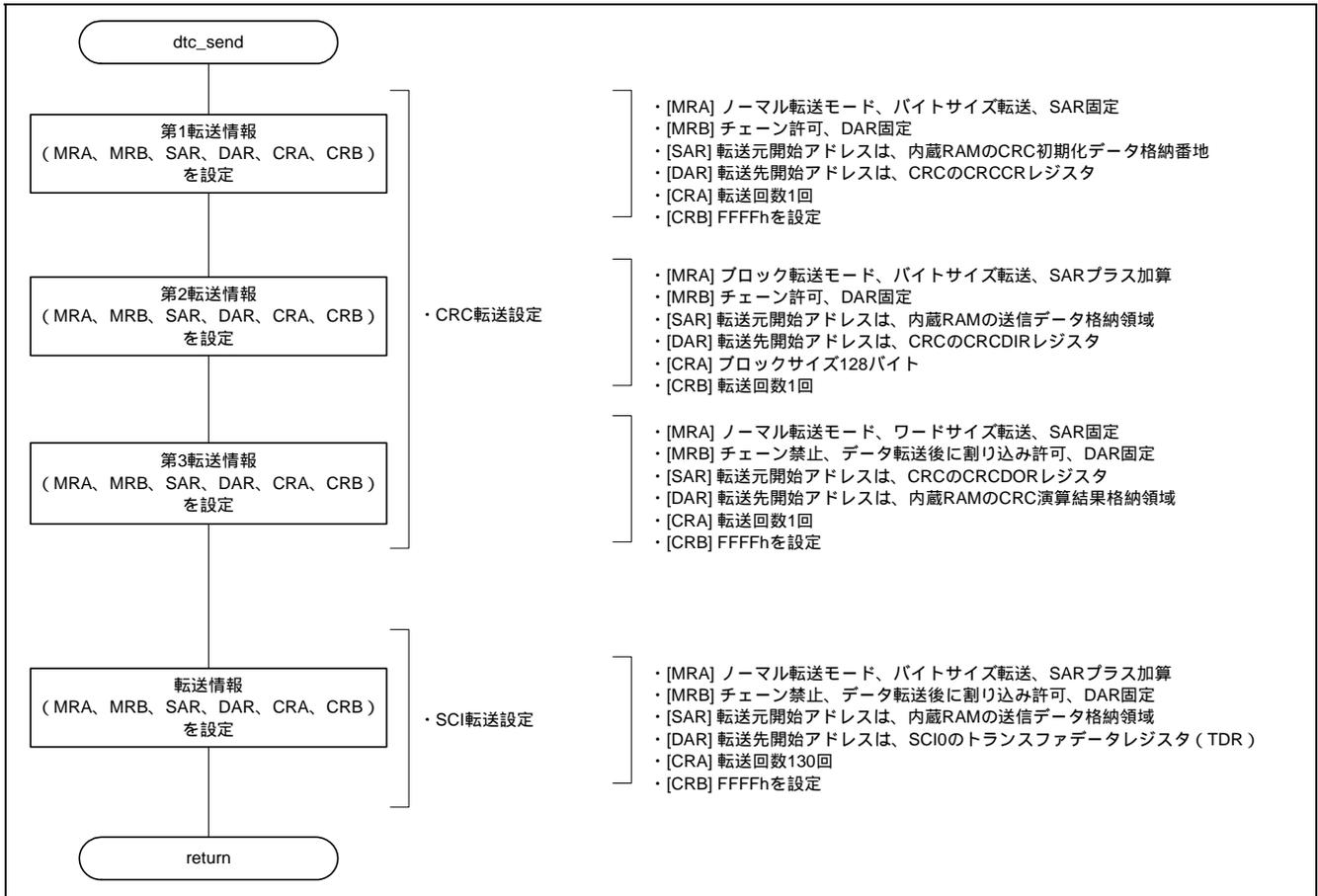


図 11 送信の転送情報配置処理

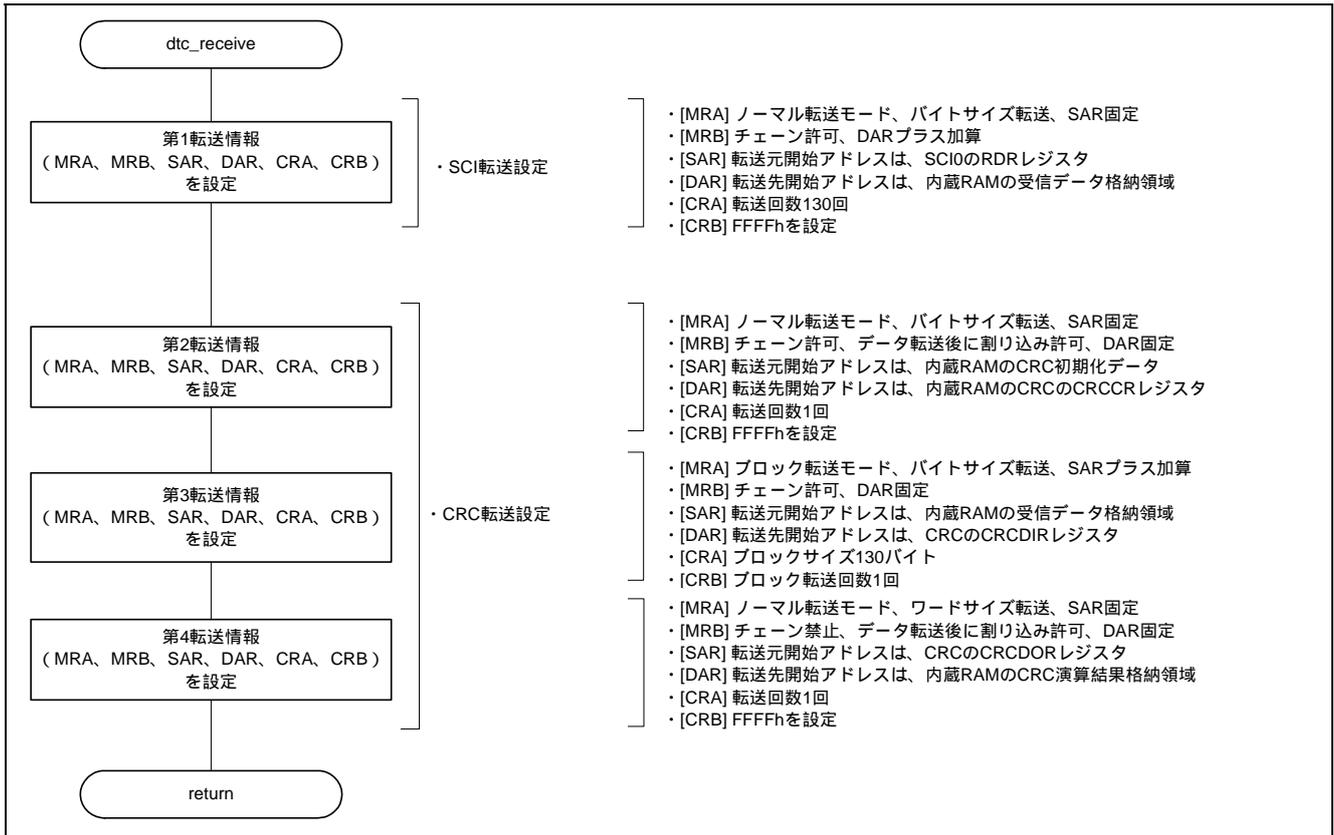


図 12 受信の転送情報配置処理

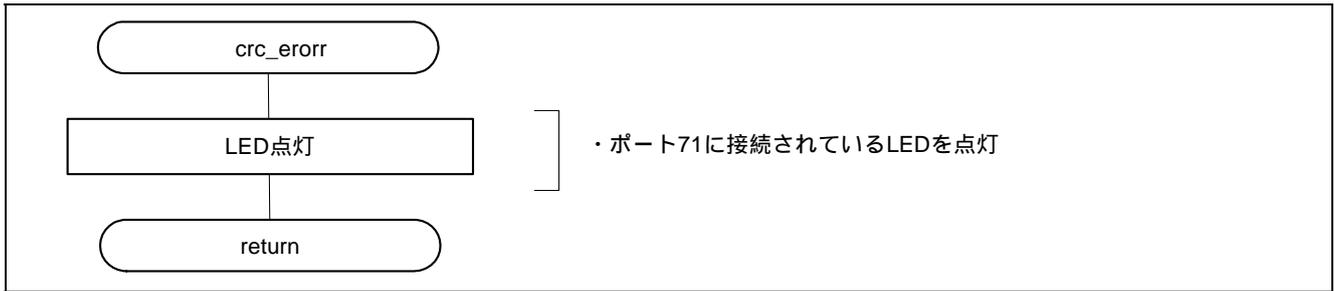


図 13 CRC エラー表示

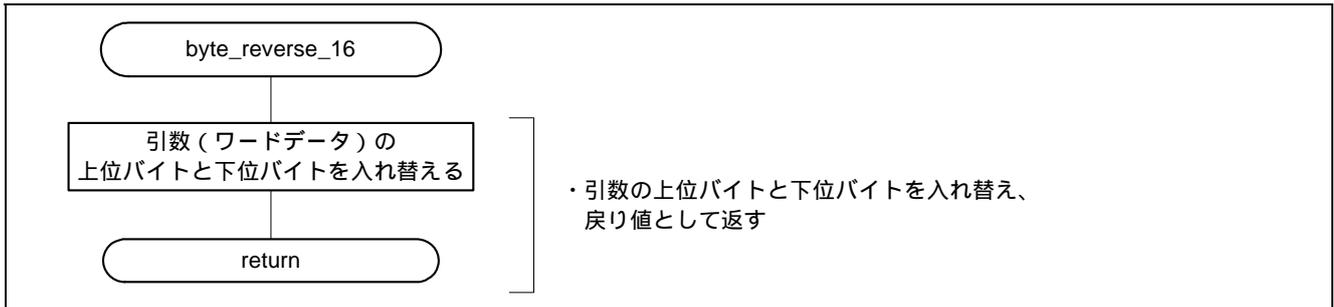


図 14 ワードデータのバイト並び替え

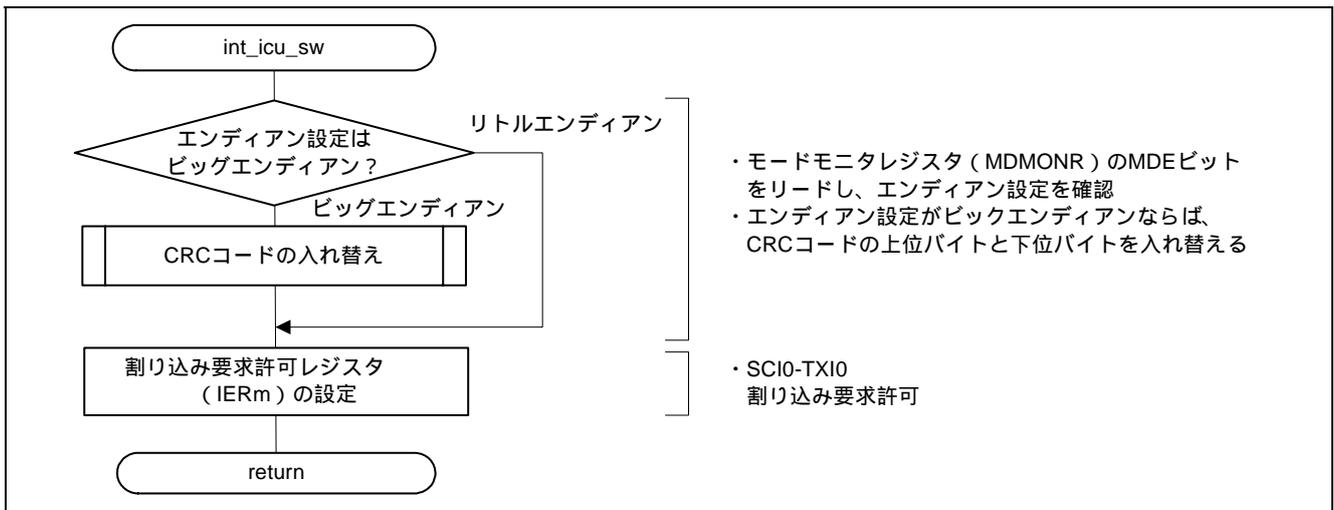


図 15 ソフトウェア割り込み

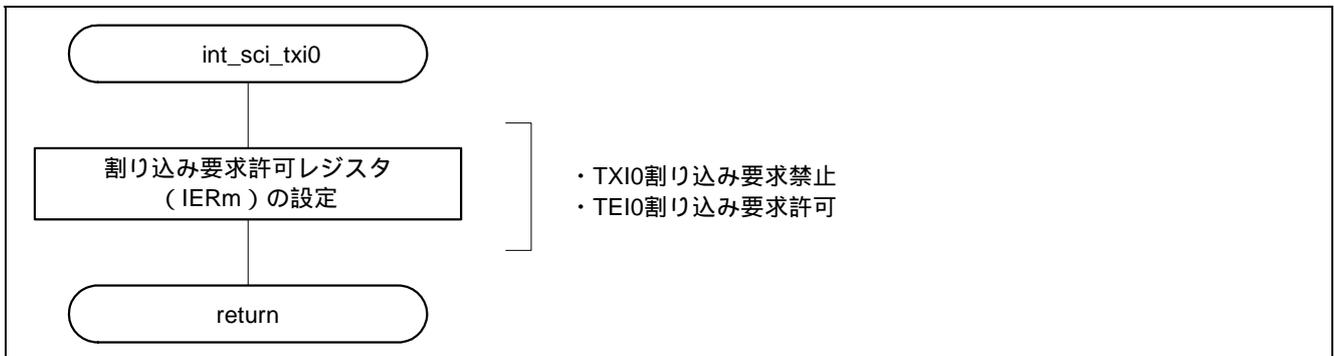


図 16 送信データエンプティ割り込み

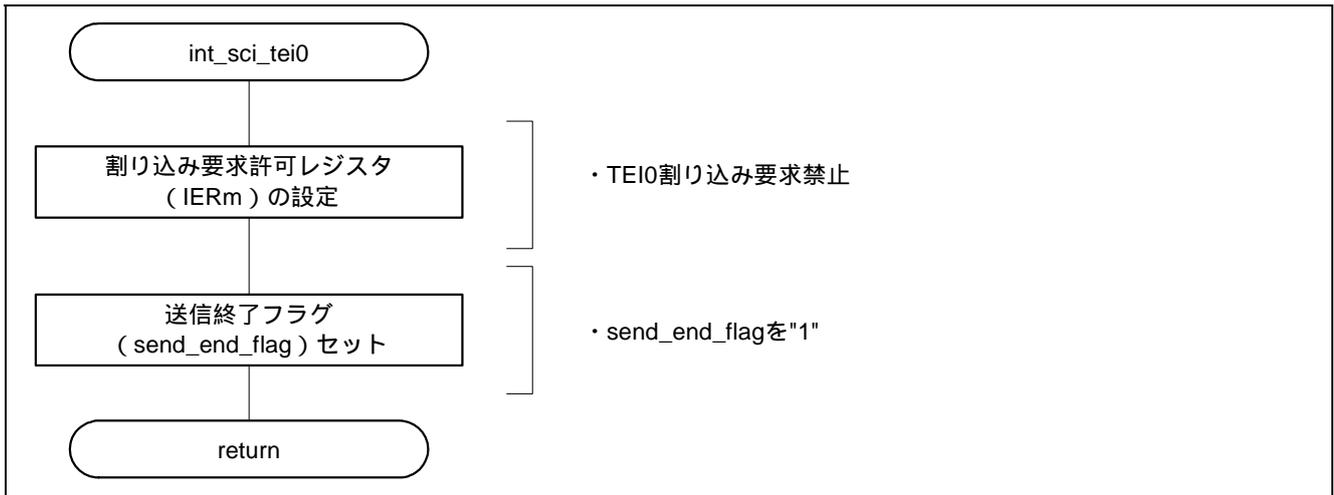


図 17 送信完了割り込み

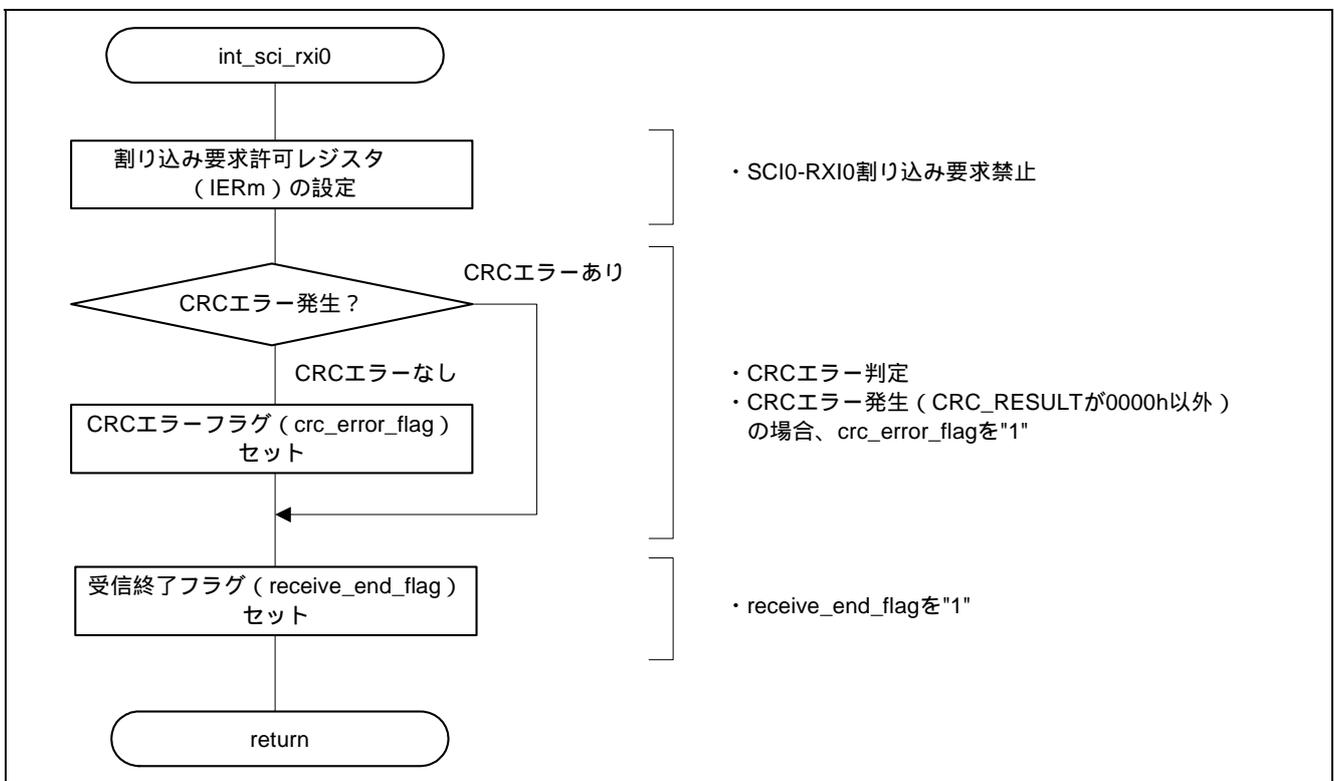


図 18 受信データエンプティ割り込み



図 19 受信エラー割り込み

## 6. 参考ドキュメント

- ハードウェアマニュアル  
RX62T グループ ユーザーズマニュアル ハードウェア編 Rev.1.00  
(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。)
- ソフトウェアマニュアル  
RX ファミリ ユーザーズマニュアル ソフトウェア編 Rev.1.00  
(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。)
- 開発環境マニュアル  
RX ファミリ用 C/C++コンパイラパッケージ ユーザーズマニュアル V.1.0.1.0  
(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  
(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。)

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 | 2011.07.15 | —    | 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. 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）がありません。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>