

RX63N グループ

DTC を用いた調歩同期式通信

要旨

本サンプルコードでは、データトランスファコントローラ (DTC) を用いた調歩同期式通信の方法について説明します。

対象デバイス

- RX63N

内容

1.	仕様	4
2.	動作確認条件	5
3.	ハードウェア説明.....	6
3.1	使用端子一覧.....	6
3.2	接続信号	6
4.	ソフトウェア説明.....	7
4.1	動作概要	7
4.1.1	送信動作.....	7
4.1.2	受信動作.....	9
4.2	ファイル構成.....	11
4.3	オプション設定メモリ.....	12
4.4	定数一覧	12
4.5	変数一覧	13
4.6	関数一覧	13
4.7	関数仕様	14
4.8	作成する関数のフローチャート.....	16
4.8.1	初期設定.....	16
4.8.2	メイン処理.....	17
4.8.3	SCI0 送信完了通知処理	17
4.8.4	SCI0 受信完了処理	18
4.8.5	SCI2 受信完了処理	19
4.8.6	SCI2 送信完了処理	19
4.8.7	SCI0 受信エラー処理	20
4.8.8	SCI2 受信エラー処理	20
5.	PDG の設定	21
5.1	SYSTEM 設定	23
5.2	SCI0 設定	25
5.3	SCI2 設定	27
5.4	DTC の設定	29
5.5	SYSTEM の端子設定	34
5.6	I/O 設定	35
5.7	ソースの生成.....	36
5.8	CS+への登録.....	37
6.	CS+のプロジェクトに PDG のソースファイルを登録する際の設定.....	39

7.	動作確認方法	42
7.1	ウォッチ式の登録.....	42
7.2	実行	44
8.	参考ドキュメント.....	47

1. 仕様

SCI0 から SCI2 へ調歩同期式通信を用い、送信データエンプティ割り込み要求(TXI0)で DTC を起動し、データを送信します。SCI2 は受信データフル割り込み要求(RXI2)で DTC を起動し、受信したデータを RAM へ格納します。次に TXI2 で DTC を起動し、SCI0 から受信したデータを送り返します。SCI0 は RXI0 で DTC を起動し、受信したデータを RAM へ格納します。SCI0 が SCI2 へ送信したデータと SCI2 が SCI0 へ送り返したデータが一致した場合、LED1 が点灯します。図 1.1 に DTC を用いた調歩同期式データ通信の概要を示します。

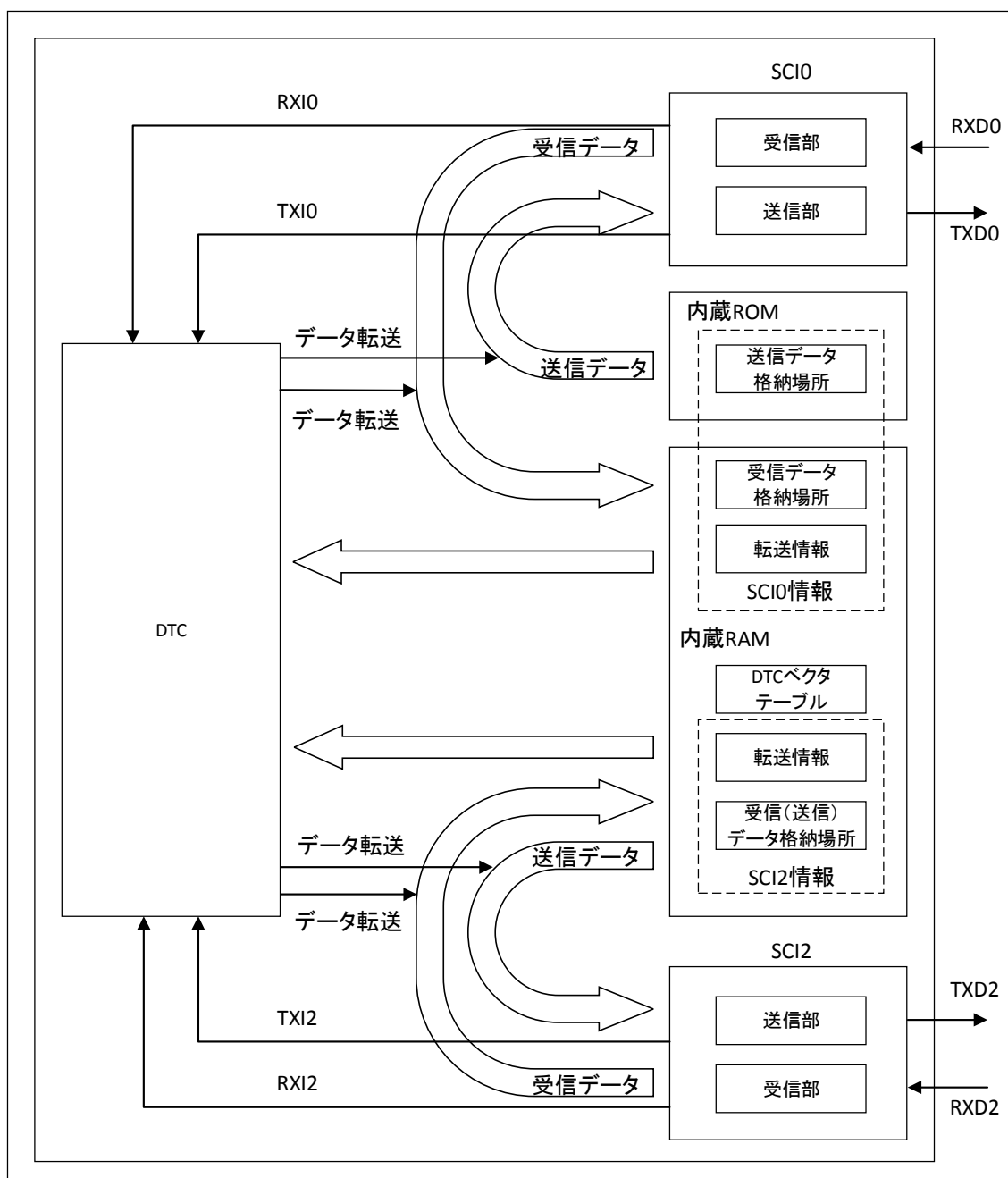


図 1.1 DTC を用いた調歩同期式データ通信の概要

2. 動作確認条件

本サンプルコードは、表 2.1 の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	R5F563NFDDFP (RX63N グループ)
動作周波数	・メインクロック：12MHz ・PLL：192MHz（メインクロック 1 分周 16 通倍） ・システムクロック (ICLK)：96MHz（PLL 2 分周）
ボード電源電圧	5V
マイコン動作電圧	3.3V
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
統合開発環境	ルネサスエレクトロニクス製品 CS+ for CC-RL V5.00.00
エミュレータ	ルネサスエレクトロニクス製 E1 エミュレータ
使用ボード	北斗電子製評価ボード HSBRX63NP (R5F563NFDDFP)

3. ハードウェア説明

3.1 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
P20	出力	TXD0
P21	入力	RXD0
P13	出力	TXD2
P12	入力	RXD2
PD6	出力	LED1

3.2 接続信号

本サンプルコードでは、使用ボードで表 3.2 に示す接続を追加してください。

表 3.2 接続信号一覧

	送信	受信
接続 1	J2 の 9pin (P20/TXD0)	J2 の 3pin (P12/RXD2)
接続 2	J2 の 4pin (P13/TXD2)	J2 の 10pin (P21/RXD0)

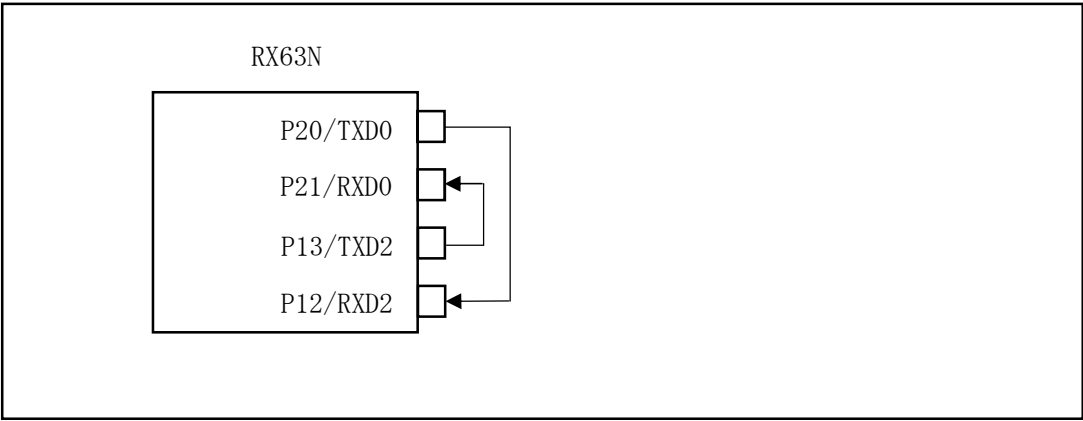


図 3.1 信号接続図

4. ソフトウェア説明

4.1 動作概要

4.1.1 送信動作

図 4.1 に送信動作タイミング図を示した上でそれぞれ図中の番号の動作および処理を示します。使用関数も合わせて参照ください。

(1) 初期設定

SCI0 から SCI2 に送信するために、以下のように初期設定を行います。

DTC(TXI0)の設定概要は次の通りです。詳細は「5.4 DTC の設定」を参照ください。

- ・フルアドレスモード
- ・ノーマルモード
- ・転送回数 13 回
- ・転送データバイトサイズは 1 バイト
- ・転送元アドレスは内蔵 RAM でアドレス更新モードはインクリメント
- ・転送先アドレスは SCI0 トランスミットデータレジスタでアドレス更新モードは固定
- ・SCI 送信データエンプティ割り込みで起動

SCI0 の設定概要は次の通りです。詳細は「5.2 SCI0 設定」を参照ください。

- ・調歩同期式モード
- ・データ長を 8 ビットとし、ストップビットは 1 ビット、LSB ファーストとする
- ・ボーレートは 57600bps とする

(2) 送信動作及び TXI 割り込み要求を許可

TE ビットを 1 にすることによりシリアル送信動作が、TIE ビットを 1 にすることにより TXI 割り込み要求が許可されます。TXI 割り込み要求により DTC が起動し、データ 1 が TDR に転送されます。

(3) データ 1 を TSR に転送

TDR からデータ 1 を TSR に転送します。データは直ちに TXD 端子より送信されます。同時に TXI 割り込み要求により DTC が起動し、データ 2 が TDR に転送されます。

(4) データラストを TDR に転送

DTC が起動し、データラストを TDR に転送します。指定回数の転送が完了したため、CPU へ割り込みが発生し、TXI 割り込み処理 (*1) を実行します。

(*1) : PDG の DTC 項目にある送信完了通知関数

(5) データラストを TSR に転送

TDR にあるデータラストを TSR に転送します。データは直ちに TXD0 端子より送信されます。

(6) データラストの送信完了

TXD0 端子から送信が完了します。

※本サンプルコードでは使用していませんが、TEI を使用する場合は、このタイミングで SCI の送信完了を確認できます。

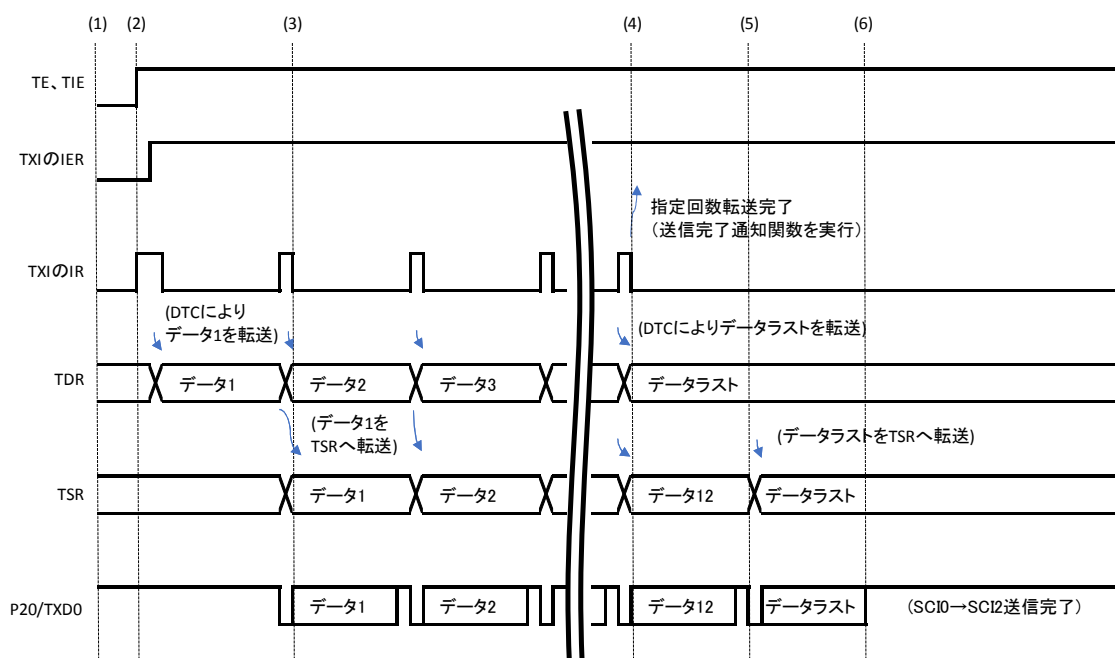


図 4.1 送信動作タイミング図

4.1.2 受信動作

図 4.2 に受信動作タイミング図を示した上でそれぞれ図中の番号の動作および処理を示します。使用関数も合わせて参照ください。

(1) 初期設定

SCI0 からのデータを SCI2 が受信するために、以下のように初期設定を行います。

DTC(RXI2)の設定概要は次の通りです。詳細は「5.4 DTC の設定」を参照ください。

- ・フルアドレスモード
- ・ノーマルモード
- ・転送回数 13 回
- ・転送データバイトサイズは 1 バイト
- ・転送元アドレスは SCI2 レシーブデータレジスタでアドレス更新モードは固定
- ・転送先アドレスは内蔵 RAM でアドレス更新モードはインクリメント
- ・SCI 受信データフル割り込みで起動

SCI2 の設定概要は次の通りです。詳細は「5.3 SCI2 設定」を参照ください。

- ・調歩同期式モード
- ・データ長を 8 ビットとし、ストップビットは 1 ビット、LSB ファーストとする
- ・ボーレートは 57600bps とする

(2) 受信動作及び RXI 割り込み要求を許可

RIE ビットを 1 にすることにより RXI 割り込み要求が、RE ビットを 1 にすることによりシリアル受信動作が許可されます。

(3) データ 1 受信完了

データ 1 の受信が完了すると、RXI 割り込みにより DTC が起動され、受信されたデータ 1 は RDR から受信バッファ (RAM) へ転送されます。

(4) データラスト受信完了

データラストの受信が完了すると、RXI 割り込み要求により DMAC が起動され、受信されたデータは RDR から受信バッファ (RAM) へ転送されます。指定回数の転送が完了したため、CPU へ割り込みが発生し、RXI 割り込み処理 (*2) を実行します。

(*2):受信完了通知関数はこのタイミングで呼ばれます。

4.2 ファイル構成

本サンプルコードを作成するにあたり、編集したファイルを表 4.1 に示します。統合開発環境で自動生成されて編集していないファイル、および 5. PDG の設定で生成されるファイルに関しましては割愛します。

表 4.1 ファイル名一覧

ファイル名	概要	備考
DTC_RX63N.c	メインファイル <ul style="list-style-type: none"> • SCI0 送信/受信処理 • SCI2 送信/受信処理 • LED1 制御 • オプション設定メモリ 	
hwsetup.c	初期設定 <ul style="list-style-type: none"> • 存在しない端子の処理 • クロックの設定 • ポートの設定 • DTC の設定 • SCI0 の設定 • SCI2 の設定 	
resetprg.c	リセット例外処理	HardwareSetup(); のコメントアウトを解除しました

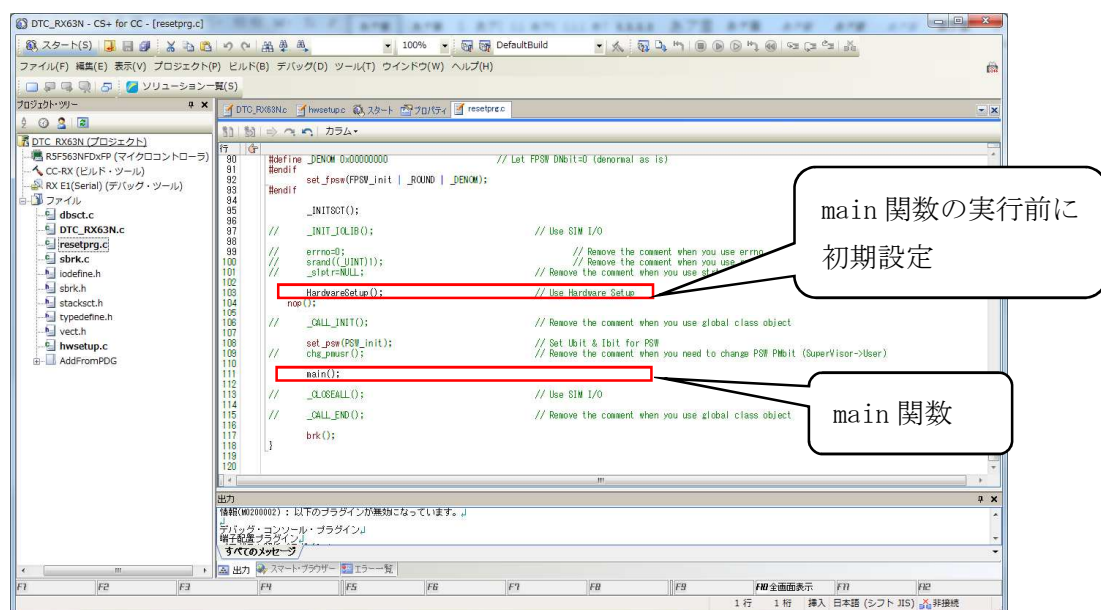


図 4.3 resetprg.c

4.3 オプション設定メモリ

表 4.2 に本サンプルコードで使用するオプション設定メモリの状態を示します。

表 4.2 オプション設定メモリー一覧

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh～FFFF FF8Ch	FFFF FFFFh	リセット後、IWDT は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh～FFFF FF88h	FFFF FFFFh	リセット後、 電圧監視 0 リセット無効 H0C0 (高速オンチップオシレー タ) 発振が無効
MDES	FFFF FF83h～FFFF FF80h	FFFF FFFFh	リトルエンディアン

OFS0 と OFS1 はメインファイルの最後尾に記載しています。

MDES については vecttbl.c ファイル(プロジェクト作成時に自動生成されるファイル)に定義されています。

4.4 定数一覧

表 4.3 に本サンプルコードで使用する定数、表 4.4 に const 型定数を示します。

表 4.3 サンプルコードで使用する定数

定数名	設定値	内容
BUFF_SIZE	13	通信バッファサイズ
VECT_SIZE	256	ベクタテーブルサイズ
DEST_SIZE	16	転送情報保存テーブルサイズ

表 4.4 サンプルコードで使用する const 型定数

型	変数名	内容	使用関数
const unsigned char	send_buf[BUFF_SIZE]	SCI0 の送信データ	main Sci0ReFunc

4.5 変数一覧

表 4.5 に本サンプルコードで使用する変数を示します。

表 4.5 サンプルコードで使用する変数

型	変数名	内容	使用関数
unsigned char	CH0_read_buf[BUFF_SIZE]	SCI0 の受信バッファ	Sci0TrFunc Sci0ReFunc
unsigned char	CH2_read_buf[BUFF_SIZE]	SCI2 の受信バッファ	main Sci2ReFunc

4.6 関数一覧

表 4.6 に関数一覧を掲載します。本サンプルコードで新規作成、もしくは編集した関数のみ記載しています。PDG の設定は「5. PDG の設定」を参照ください。サンプルコードで使っている PDG で生成された関数に関しましては、「RX63N グループ、RX631 グループ Peripheral Driver Generator リファレンスマニュアル」を参照ください。

表 4.6 関数一覧

関数名	概要
main	メイン処理
Sci0TrFunc	SCI0 送信完了通知関数 ※実際には SCI0 送信完了ではなく、最後の送信データを TDR0 へ転送し、送信データエンプティとなるタイミング
Sci0ReFunc	SCI0 受信完了通知関数
Sci2ReFunc	SCI2 受信完了通知関数
Sci2TrFunc	SCI2 送信完了通知関数 ※実際には SCI2 送信完了ではなく、最後の送信データを TDR2 へ転送し、送信データエンプティとなるタイミング
Sci0ErFunc	SCI0 受信エラー関数
Sci2ErFunc	SCI2 受信エラー関数

4.7 関数仕様

本サンプルコードで作成、もしくは編集した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	SCI2 の受信開始 SCI0 の送信開始
引数	なし
リターン値	なし

Sci0TrFunc

概要	SCI0 送信完了通知処理
ヘッダ	なし
宣言	void Sci0TrFunc(void)
説明	SCI0 の送信完了ではないため、処理なし
引数	なし
リターン値	なし

Sci0RdFunc

概要	SCI0 受信完了通知処理
ヘッダ	なし
宣言	void Sci0RdFunc(void)
説明	SCI0 及び SCI2 のシリアル送受信停止 送信データと受信データの比較 LED1 制御
引数	なし
リターン値	なし

Sci2ReFunc

概要	SCI2 受信完了通知処理
ヘッダ	なし
宣言	void Sci2RdFunc(void)
説明	SCI0 及び SCI2 のシリアル送受信停止 SCI0 の受信開始 SCI2 の受信データを送信開始
引数	なし
リターン値	なし

Sci2TrFunc

概要	SCI2 送信完了通知処理
ヘッダ	なし
宣言	void Sci2TrFunc(void)
説明	SCI2 の送信完了ではないため、処理なし
引数	なし
リターン値	なし

Sci0ErFunc

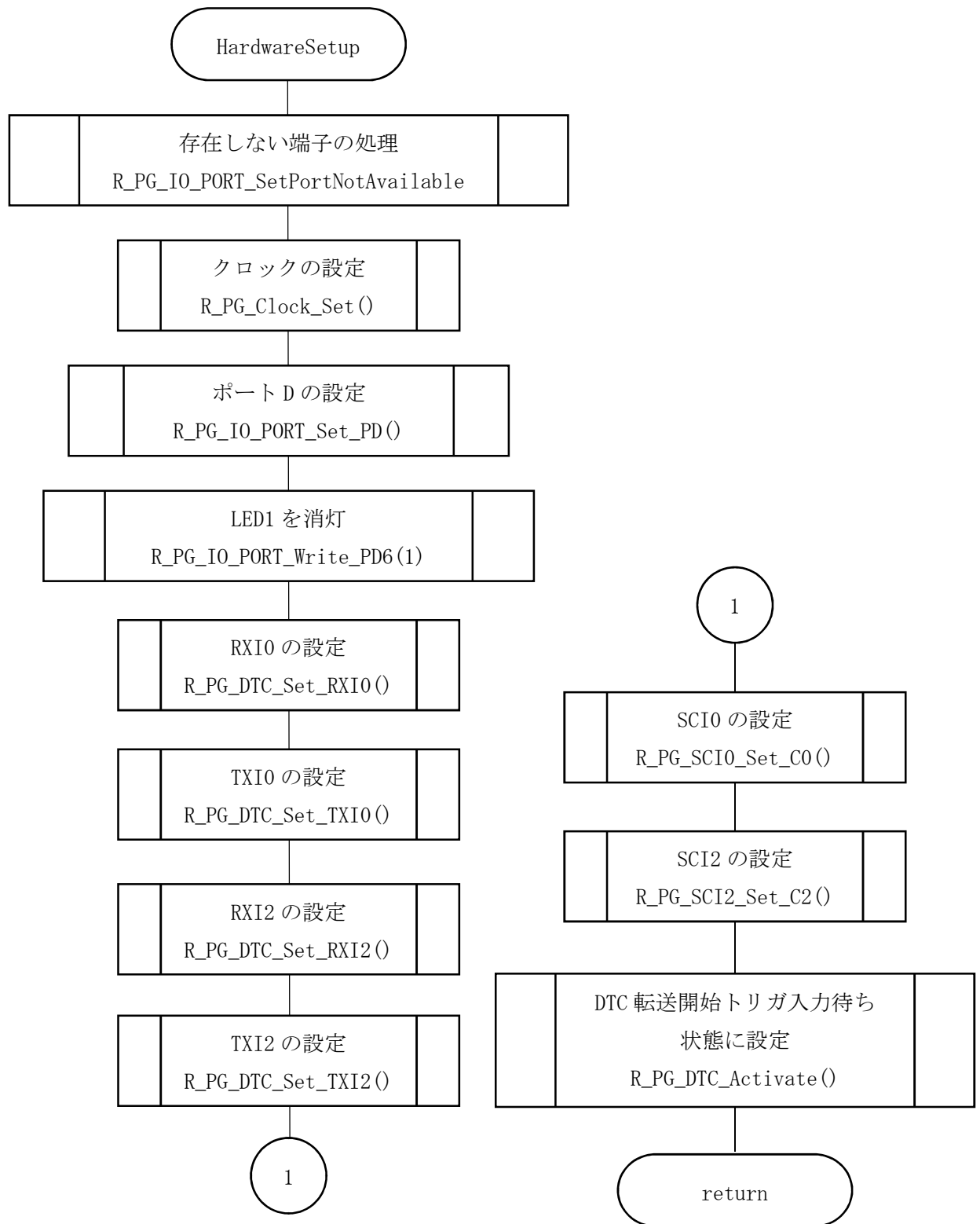
概要	SCI0 受信エラー処理
ヘッダ	なし
宣言	void Sci0ErFunc(void)
説明	処理なし
引数	なし
リターン値	なし

Sci2ErFunc

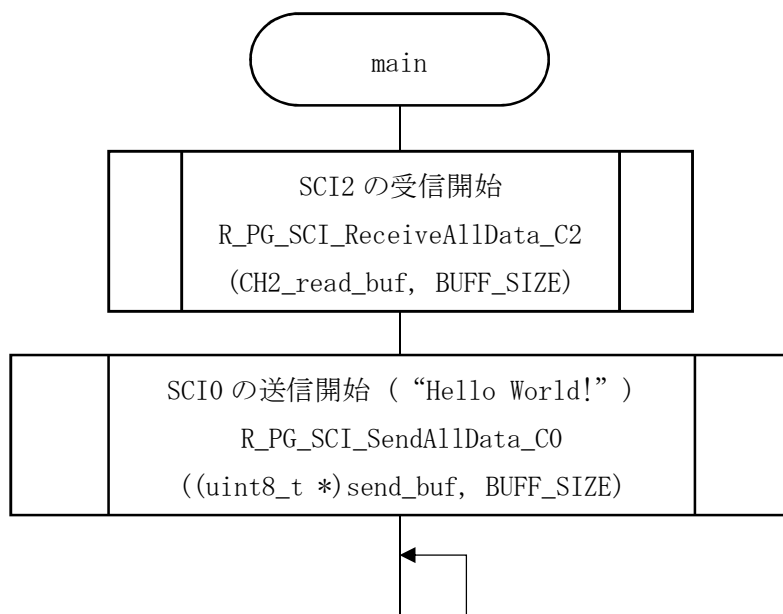
概要	SCI2 受信エラー処理
ヘッダ	なし
宣言	void Sci2ErFunc(void)
説明	処理なし
引数	なし
リターン値	なし

4.8 作成する関数のフローチャート

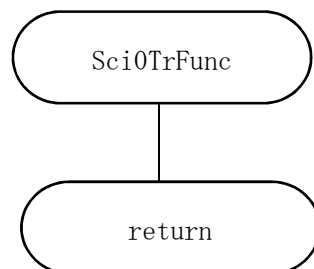
4.8.1 初期設定



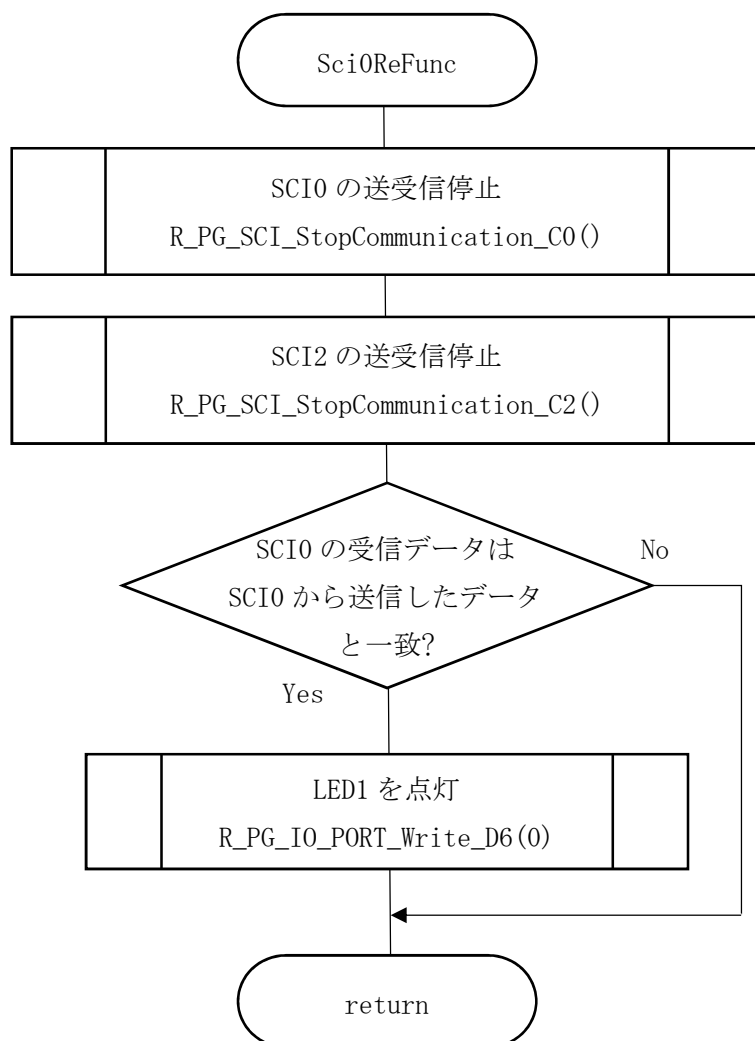
4.8.2 メイン処理



4.8.3 SCI0 送信完了通知処理



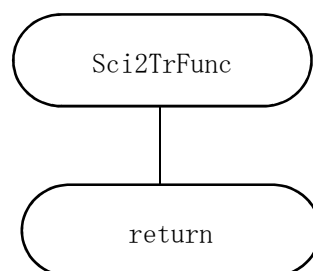
4.8.4 SCI0 受信完了処理



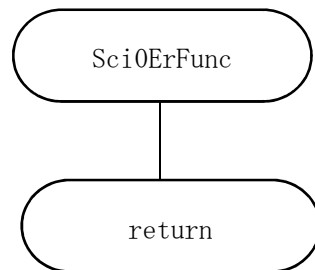
4.8.5 SCI2 受信完了処理



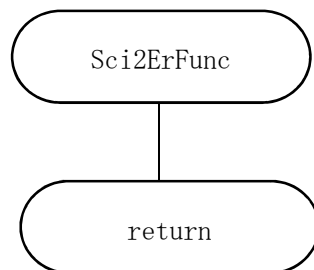
4.8.6 SCI2 送信完了処理



4.8.7 SCI0 受信エラー処理



4.8.8 SCI2 受信エラー処理



5. PDG の設定

本サンプルコードにおける PDG の設定を以下に説明します。本設定において生成されるソースファイルの詳細は”RX63N グループ、RX631 グループ Peripheral Driver Generator リファレンスマニュアル”を参照ください。

Peripheral Driver Generator 2 を起動します。



メニューバーのファイル→プロジェクトの新規作成 をクリックすると、以下のウィンドウが表示されます。プロジェクト名、マイコンのグループ、型を入力し、「OK」をクリックすると、プロジェクトが作成されます。

新規作成

プロジェクト名:
DTC_RX63N

ディレクトリ: ?
c:\renesas\PDG2_proj 参照...

デバイス選択

シリーズ: RX600

グループ: RX63N

型名: R5F563NFDDFP

パッケージ: PLQP0100KB-A

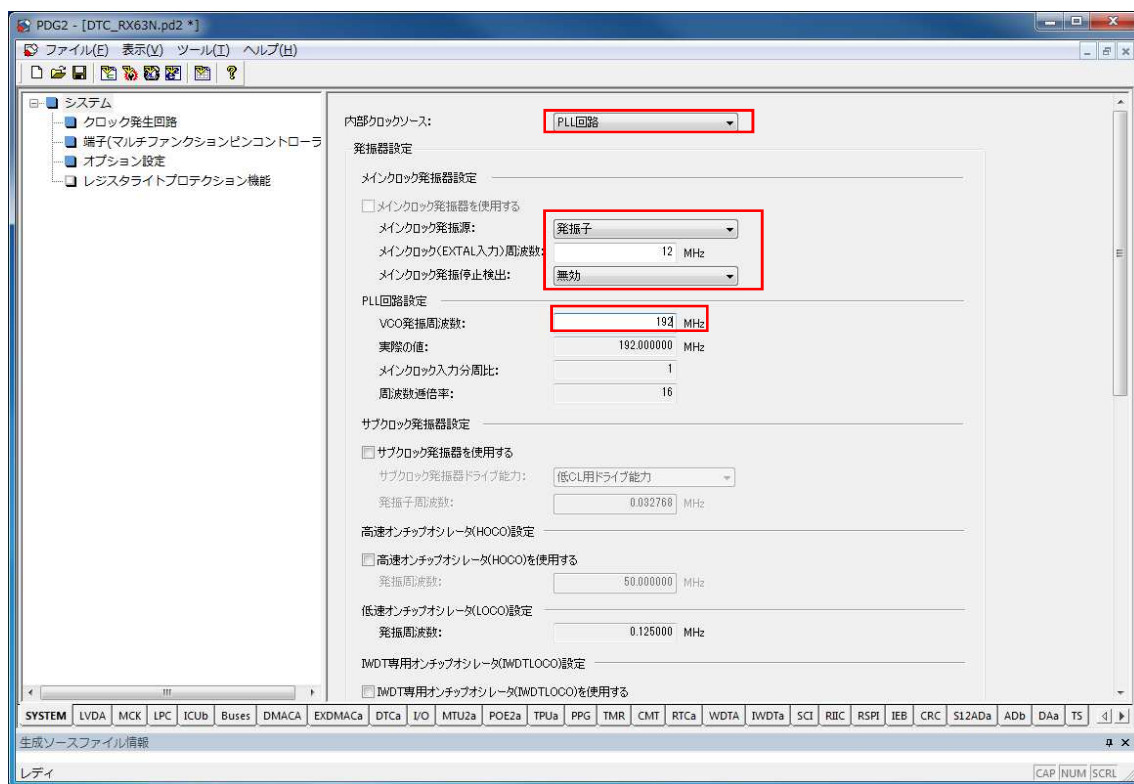
ROM容量: 2M バイト

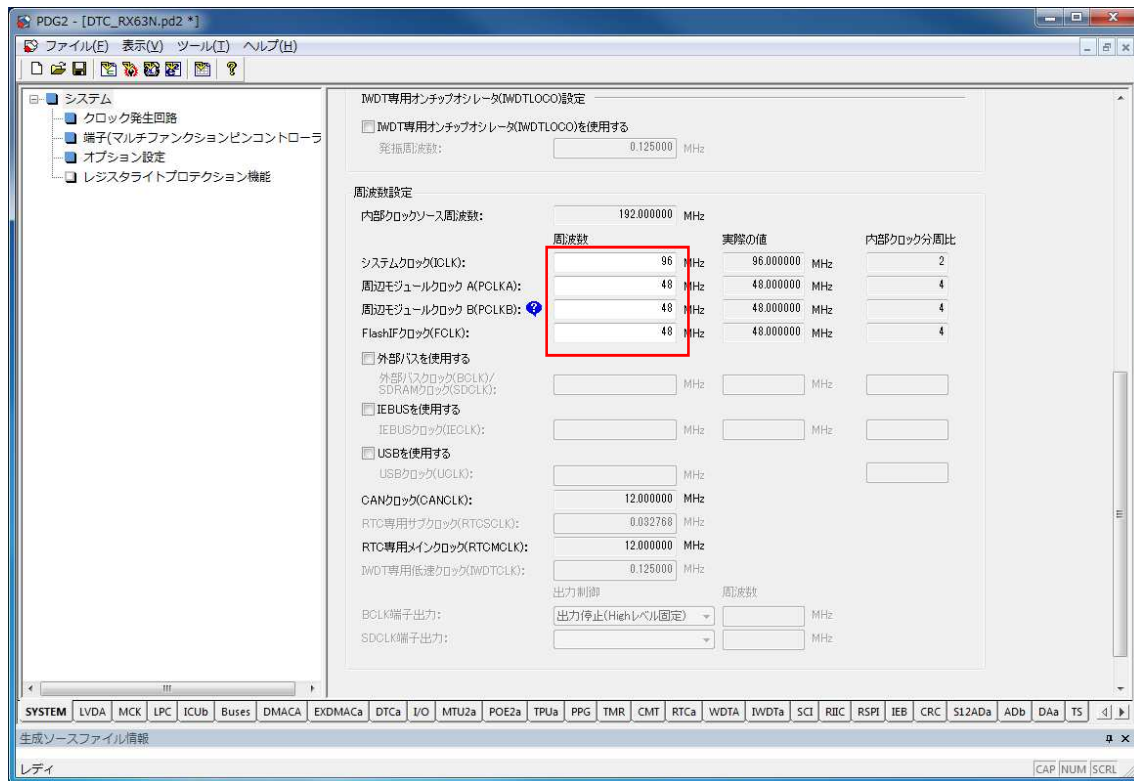
RAM容量: 256K バイト

クリック OK キャンセル

5.1 SYSTEM 設定

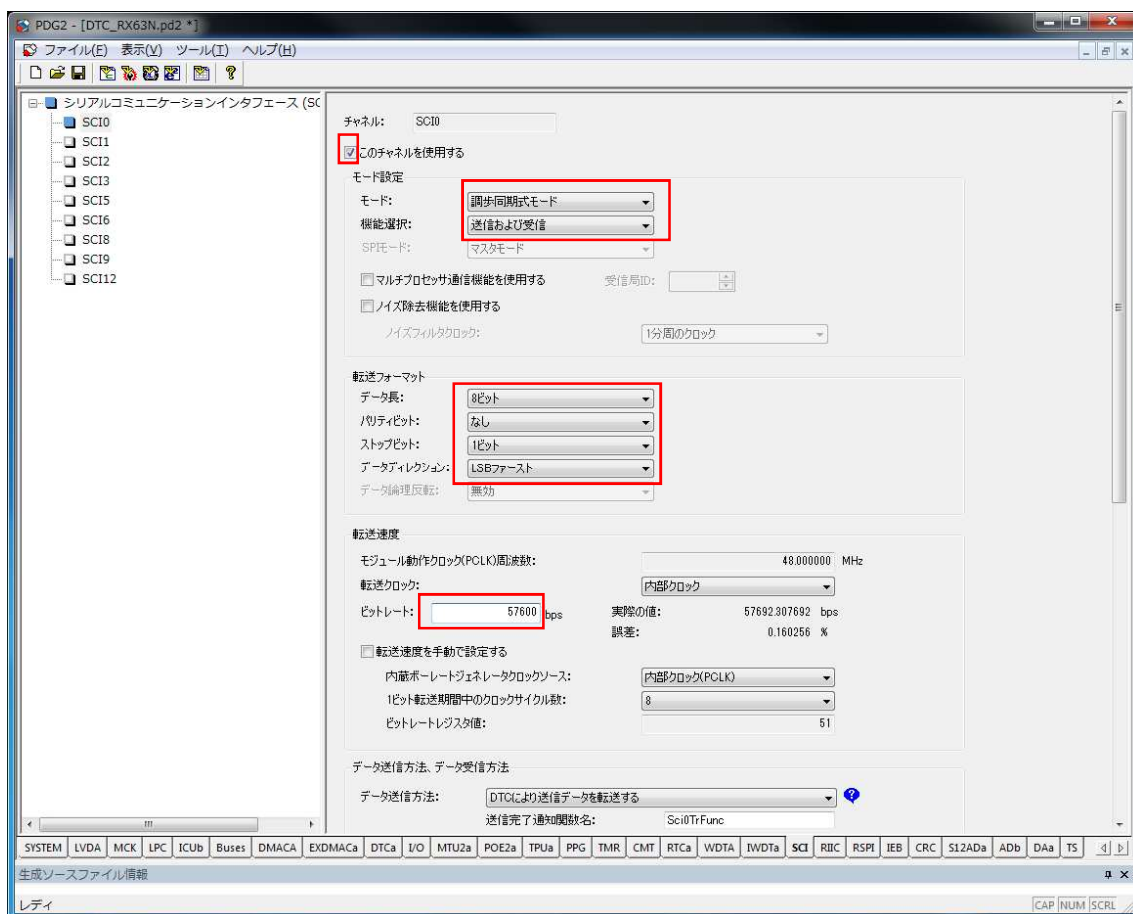
システムタブのクロック発生回路の設定を以下に示します。

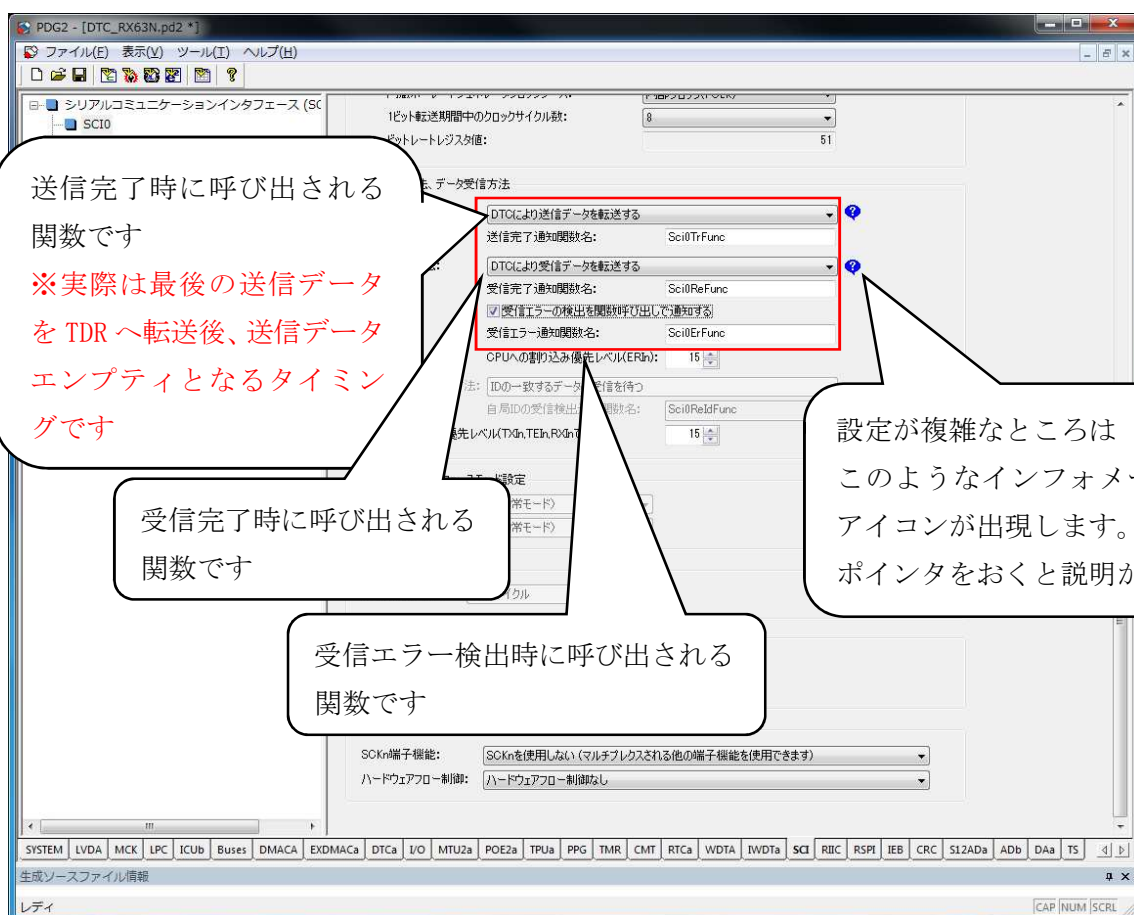




5.2 SCI0 設定

SCI0 の設定を以下に示します。





「DTC により送信データを転送する」のインフォメーションアイコンをポイントしたときは以下が表示されます。

後程 DTC の設定をする際に重要です。設定におけるポイントを赤枠で囲みます。

DTCにより送信するデータを転送する場合、**送信データエンプティ割り込み(TXIn)**をトリガとし、**転送先をトランスミットデータレジスタに指定したDTC転送を設定**してください。関数 `R_PG_SCI_SendAllData_Cn` を呼ぶと、DTC転送を開始しデータを送信します。**転送終了時は通知関数内から `R_PG_SCI_StopCommunication` を呼び出し、送信を停止してください。**

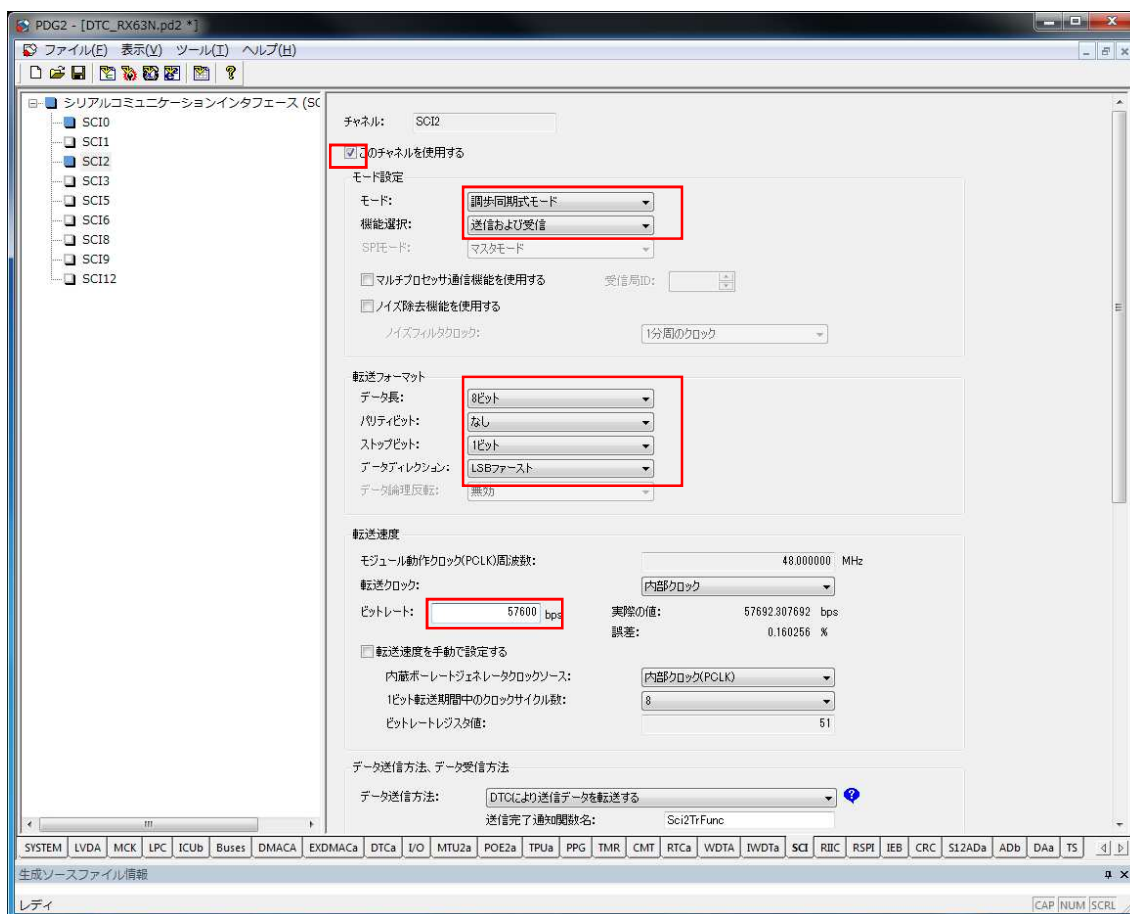
“転送終了時は通知関数内から送信を停止してください”と書いてありますが、通知関数を実行するタイミングは送信完了のタイミングではありません。SCI の送信完了後に `R_PG_SCI_StopCommunication` を呼ぶ必要があります。本サンプルコードでは受信完了時に送信、受信を停止しています。

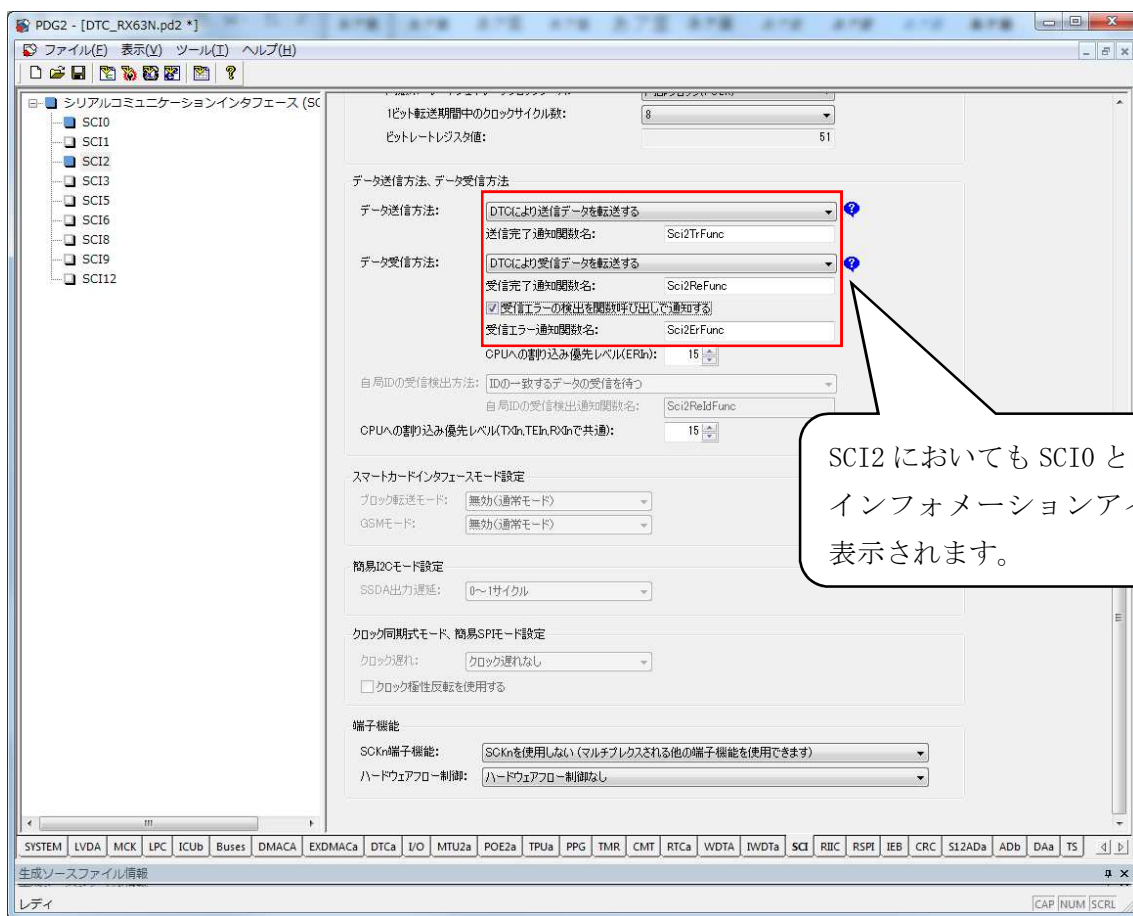
「DTC により受信データを転送する」のインフォメーションアイコンをポイントしたときは以下が表示されます。

DTCにより受信したデータを転送する場合、**受信データフル割り込み(RXIn)**をトリガとしたDTC転送を設定し、**レシーブデータレジスタのアドレスをデータ転送元に指定**してください。関数 `R_PG_SCI_ReceiveAllData_Cn` を呼ぶと、受信を開始しDTC転送を行います。転送終了時は通知関数内から `R_PG_SCI_StopCommunication` を呼び出して下さい。

5.3 SCI2 設定

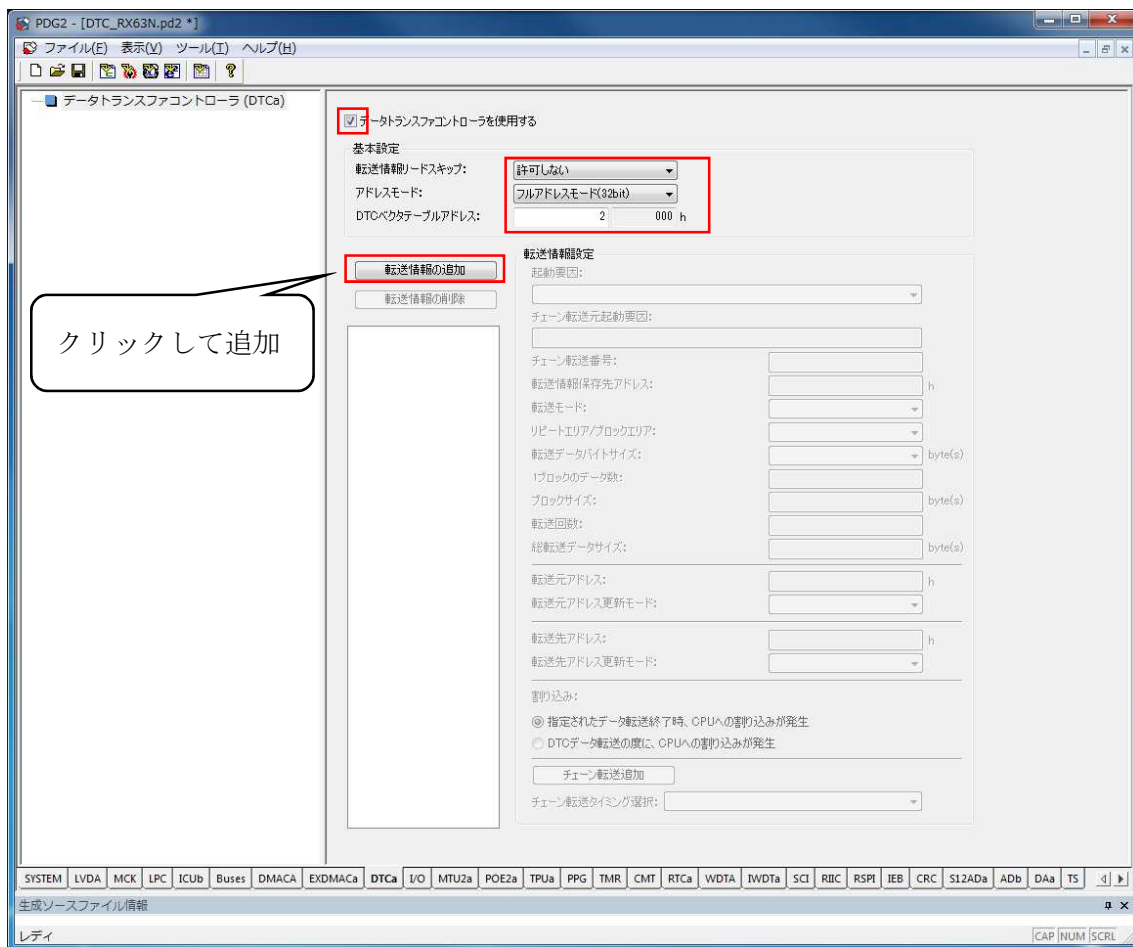
SCI2 の設定を以下に示します。(SCI0 と同様です)





5.4 DTC の設定

DTC の設定を以下に示します。



起動要因が RXI0 (SCI0 受信データフル割り込み) の転送情報設定を下に示します。
 先ほどのインフォメーションアイコンの説明に従います。

プルダウンから選択

RAMエリアから任意の箇所に配置します。

インフォメーションアイコンの説明から転送元アドレスを、SCI0 のレシーブデータレジスタのアドレスに設定

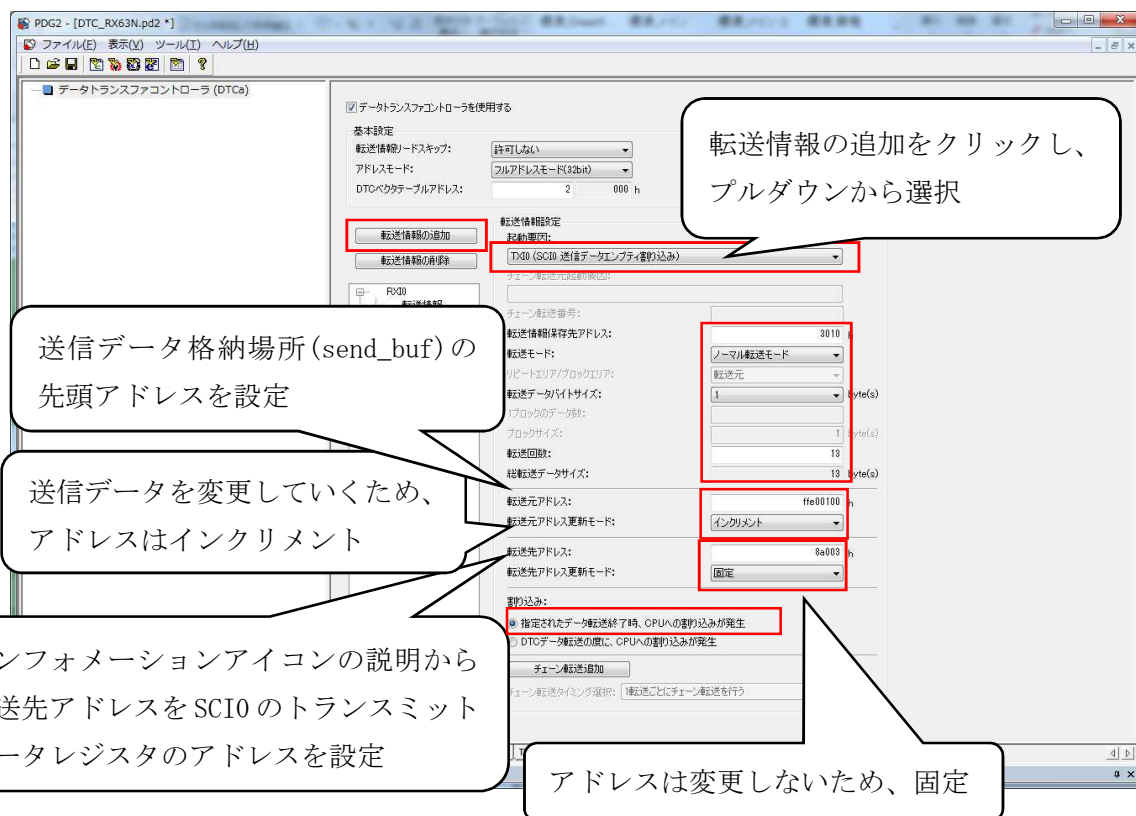
アドレスは変更しないため、固定

受信データ格納先 (CH0_read_buf) の先頭アドレスを設定

格納先を変更していくため、アドレスはインクリメント

起動要因が TXI0(SCI0 送信データエンプティ割り込み)の転送情報設定を以下に示します。

こちら先ほどのインフォメーションアイコンの説明に従います。



起動要因が RXI2 (SCI2 受信データフル割り込み) の転送情報設定を以下に示します。
RXI0 (SCI0 受信データフル割り込み) の転送情報設定と同じ要領です。

転送情報の追加をクリックし、プルダウンから選択

転送元アドレスを、SCI2 のレジスタデータレジスタのアドレスに設定

アドレスは変更しないため、固定

受信データ格納先 (CH2_read_buf) の先頭アドレスを設定

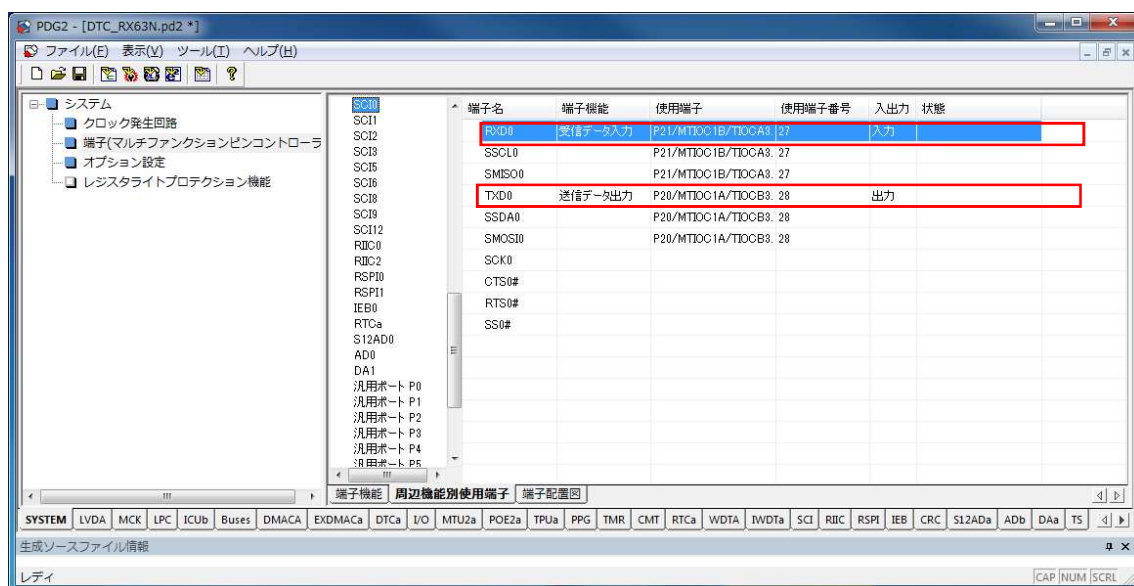
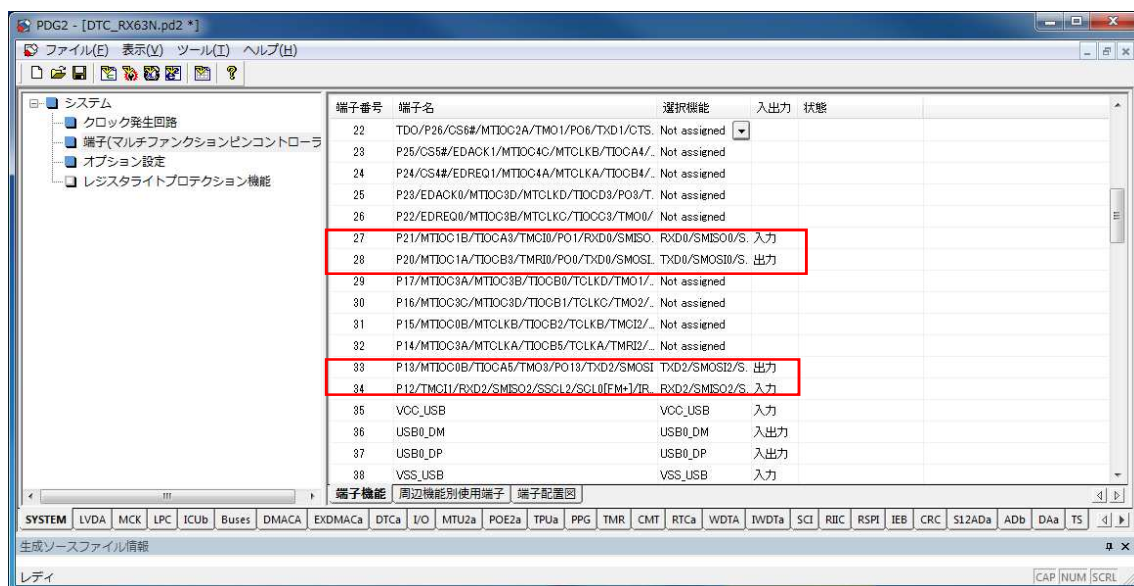
格納先を変更していくため、アドレスはインクリメント

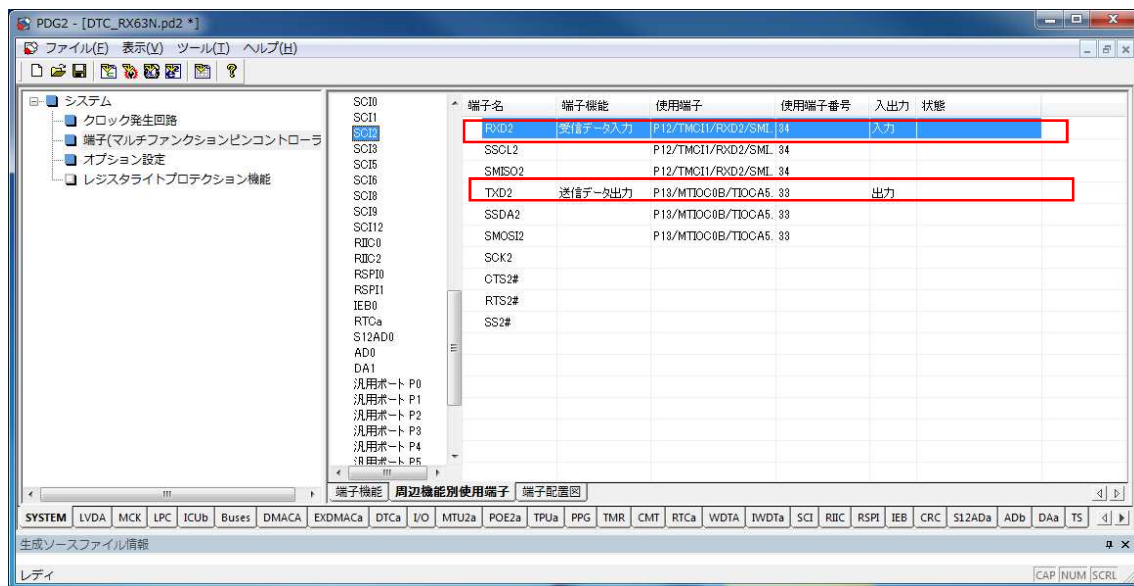
TXI0(SCI0 送信データエンプティ割り込み)の転送情報設定と同じ要領です。



5.5 SYSTEM の端子設定

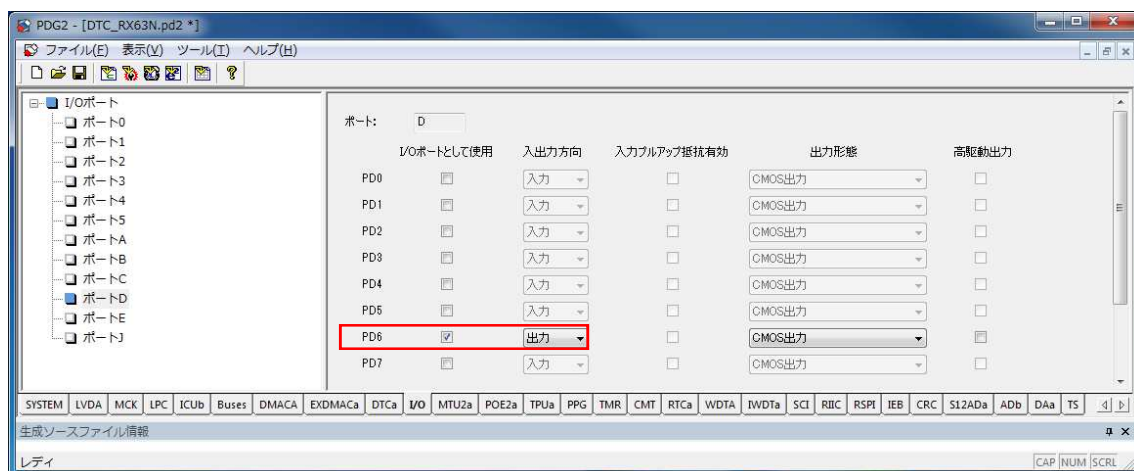
SYSTEM の端子設定を確認します。





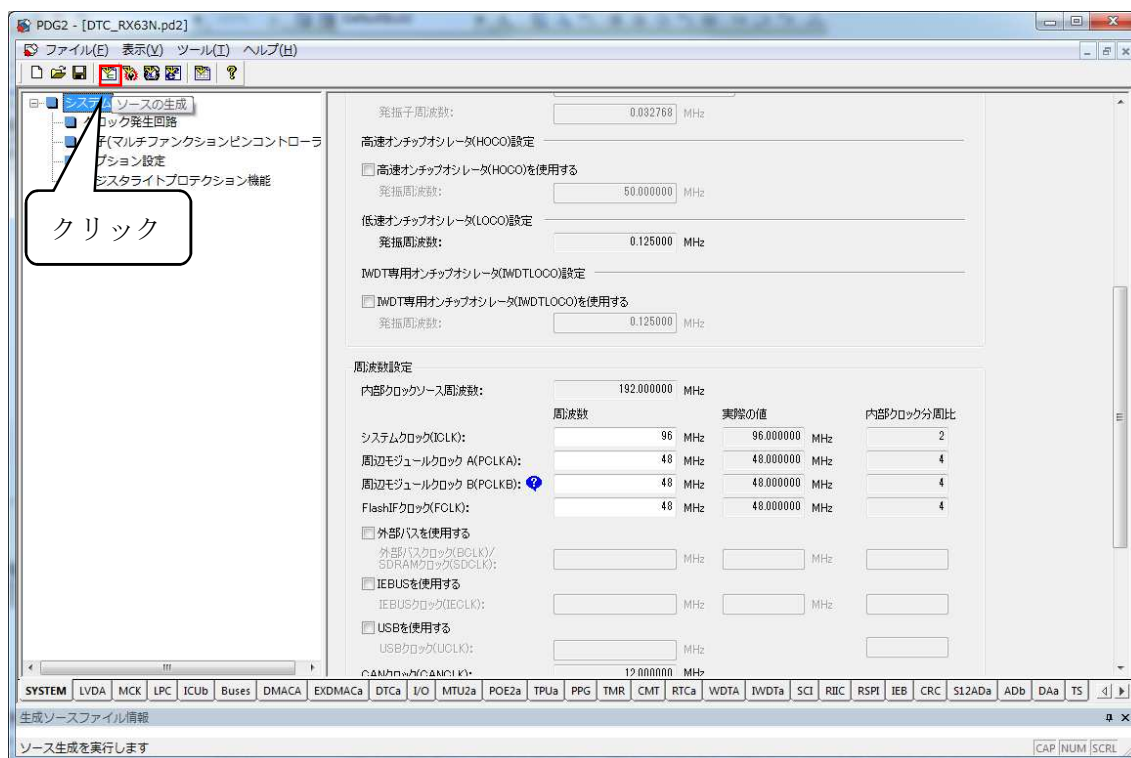
5.6 I/O 設定

I/O の設定を以下に示します。LED1 で使用する PD6 を出力にします。

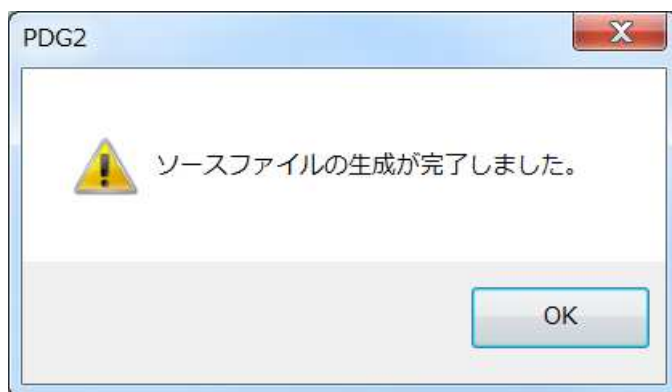


5.7 ソースの生成

以下の GUI をクリックすると、

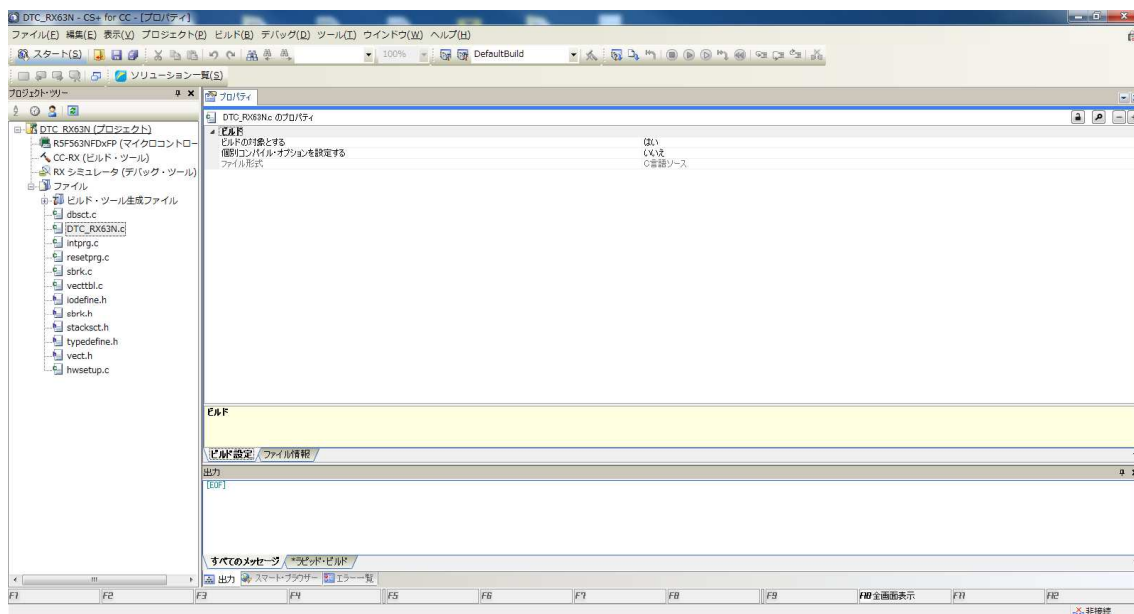


ソースファイルが生成されます。

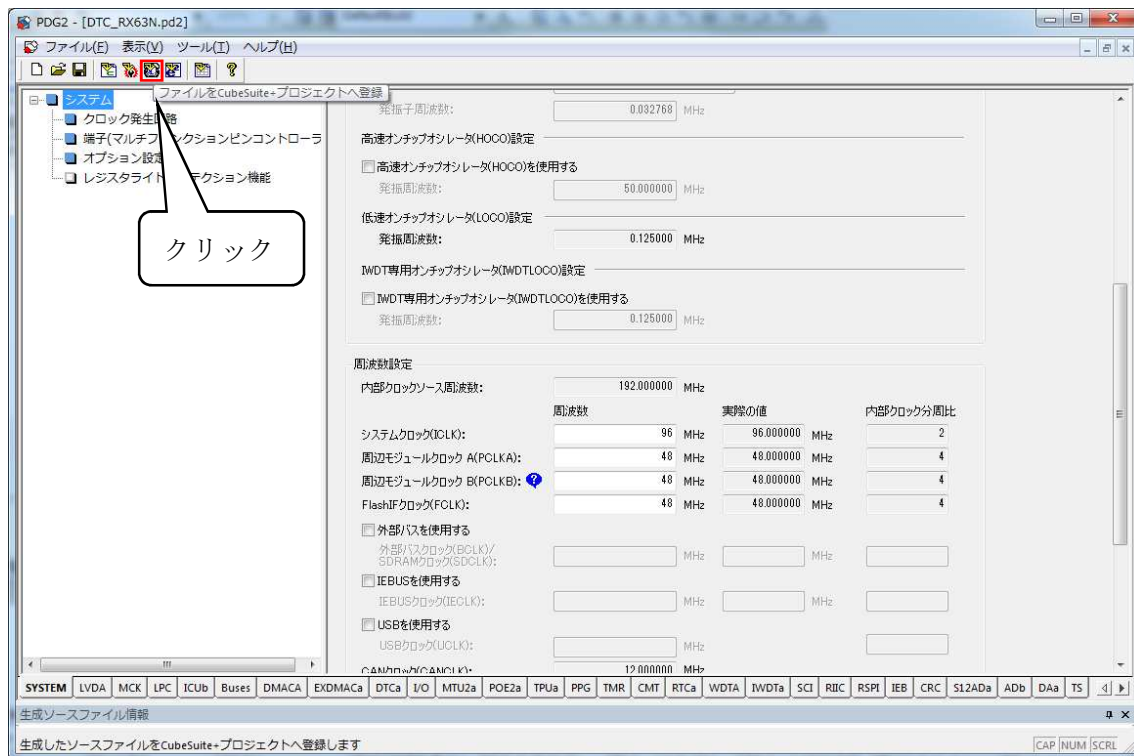


5.8 CS+への登録

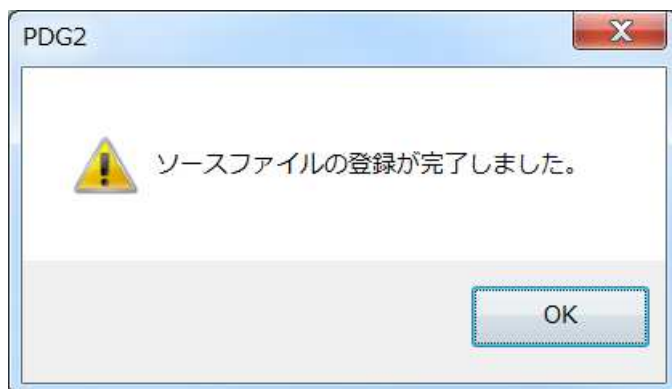
対象の CS+プロジェクトを開きます。



以下の GUI をクリックします

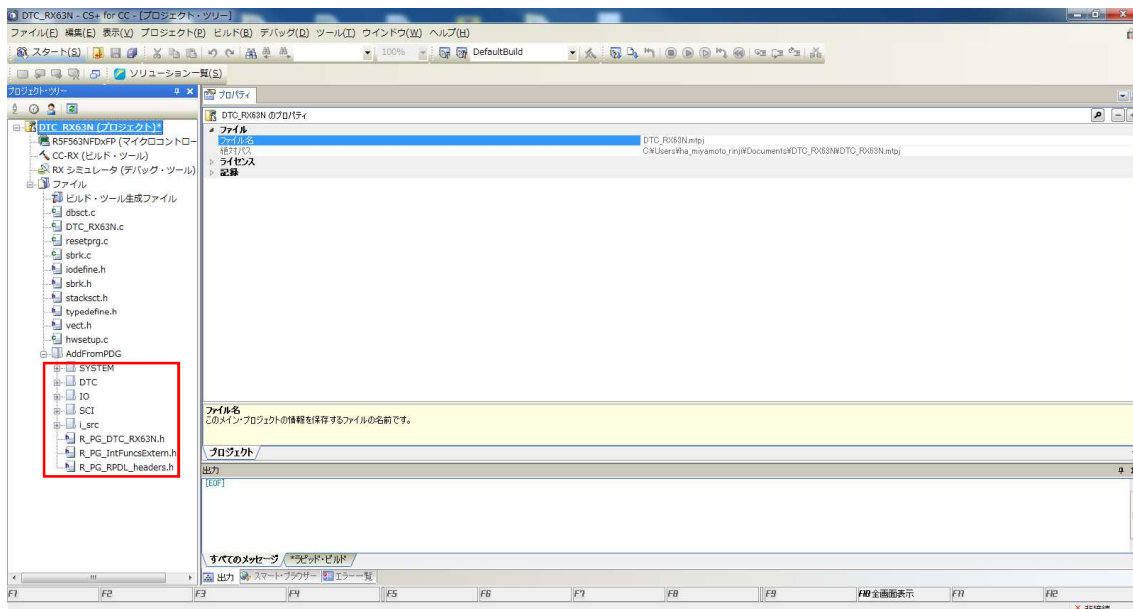


ソースファイルの登録が完了しました。



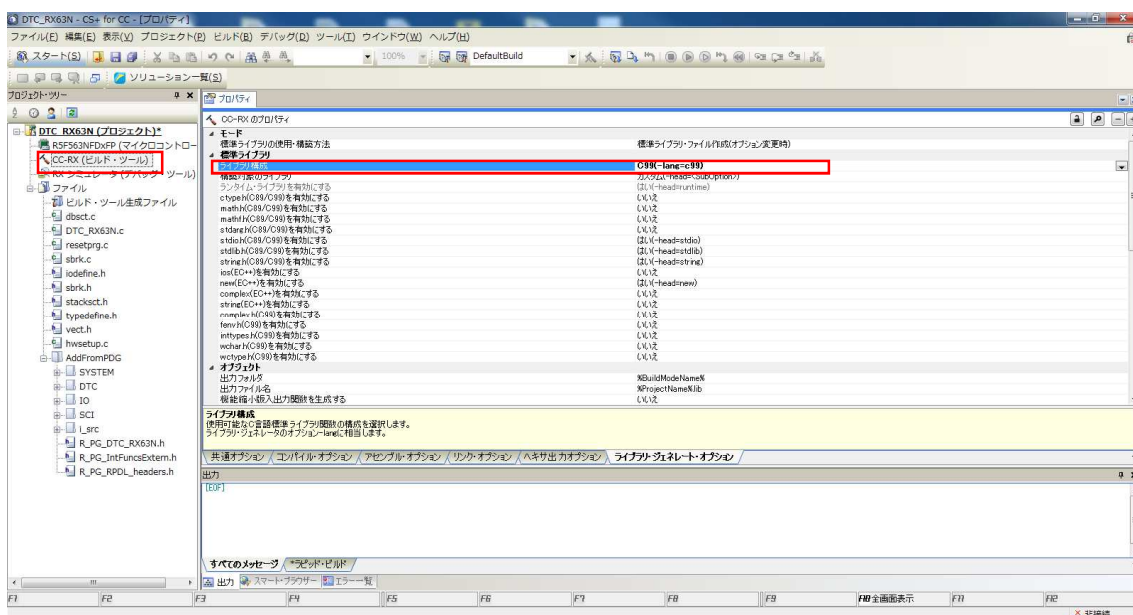
6. CS+のプロジェクトに PDG のソースファイルを登録する際の設定

CS+のプロジェクトに PDG で生成されたソースファイルを登録すると、プロジェクトのファイルに AddFromPDG フォルダが追加されます。

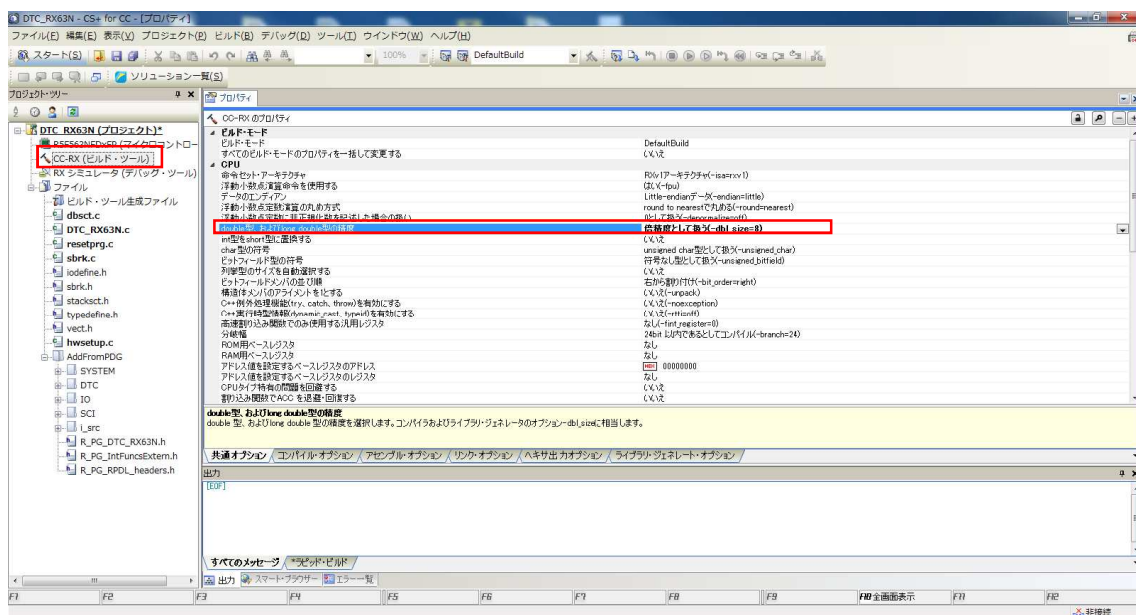


そのままビルドをすると、エラーおよび警告が発生します。解消する設定を以下に示します。

PDG で生成されるソースファイルは bool 変数を使用しています。対応させるため、ビルド・ツールを右クリック→プロパティを表示し、ライブラリ・ジェネレート・オプションタブにある「ライブラリ構成」を”C99(-lang=c99)”に設定します。



PDG で生成されるソースファイルは double 型、および long double 型の精度を倍精度として扱っているため、ビルド・ツールを右クリック→プロパティを表示し、共通オプションタブにある「double 型、および long double 型の精度」を” 倍精度として扱う(-dbl_size=8)” に設定します。

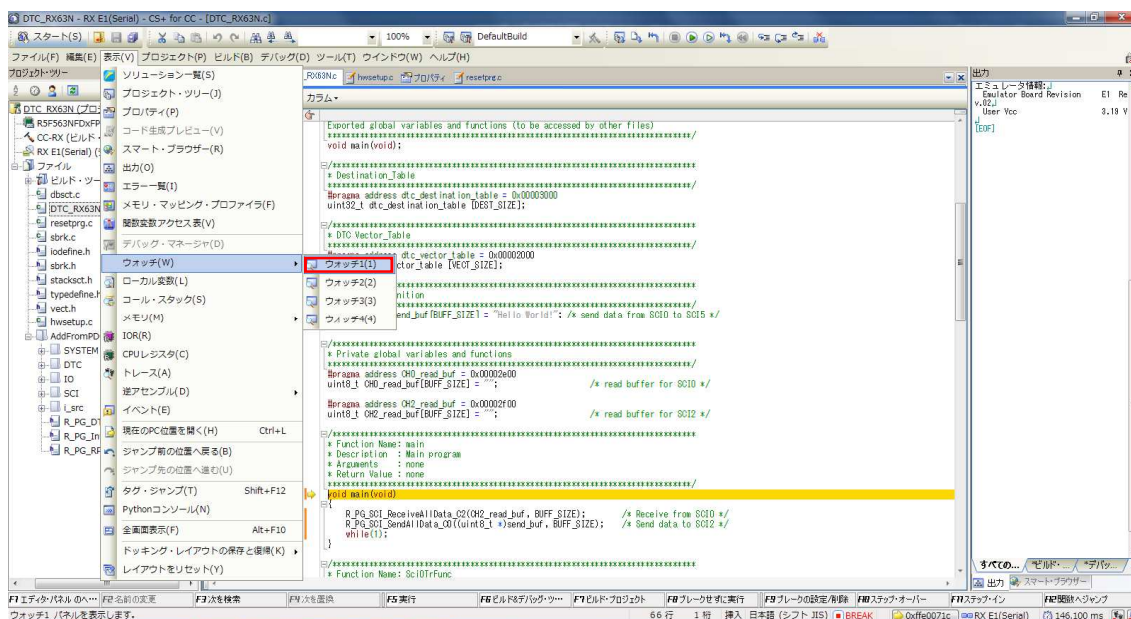


7. 動作確認方法

7.1 ウォッチ式の登録

DTC を用いて SCI0 から SCI2 にデータが送信され、SCI2 の受信データが SCI0 に送り返されているかを確認する 1 つの方法として、CS+に搭載されている機能のウォッチ式を使用しました（エミュレータを使用しない場合は、LED1 の点灯でのみ動作を確認）。ウォッチ式に登録することで対象の変数に格納されているデータを確認することができます。使用方法を以下に示します。

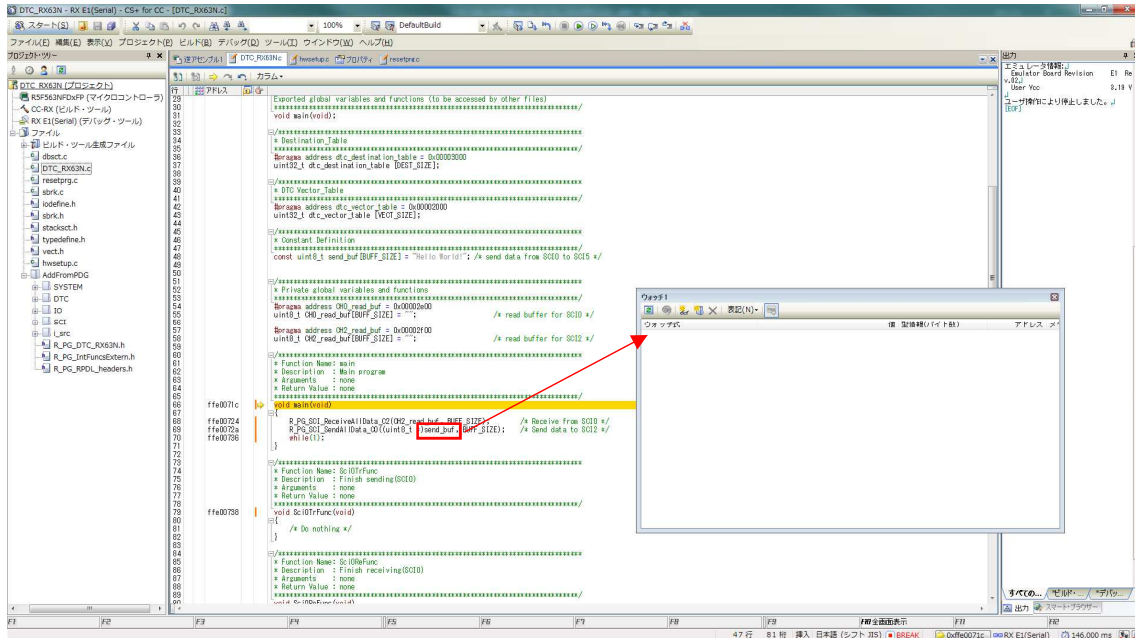
表示タブからウォッチ、そしてウォッチ 1 を選択します。



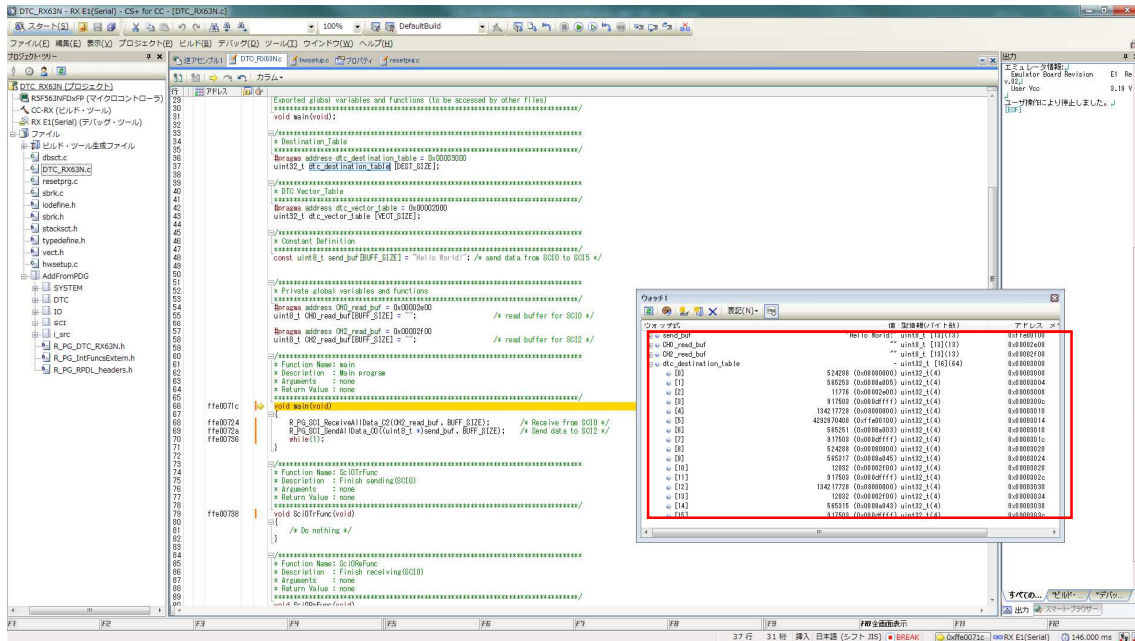
ウォッチ(ウォッチ 1)パネルが表示されました。



ウォッチ 1 が表示されたら main 関数の中の “send_buf” をウォッチパネルにドラッグ&ドロップします。



“CH2_read_buf” と “CH0_read_buf”、さらに “dtc_destinatin_table” も同様にドラッグ&ドロップします。



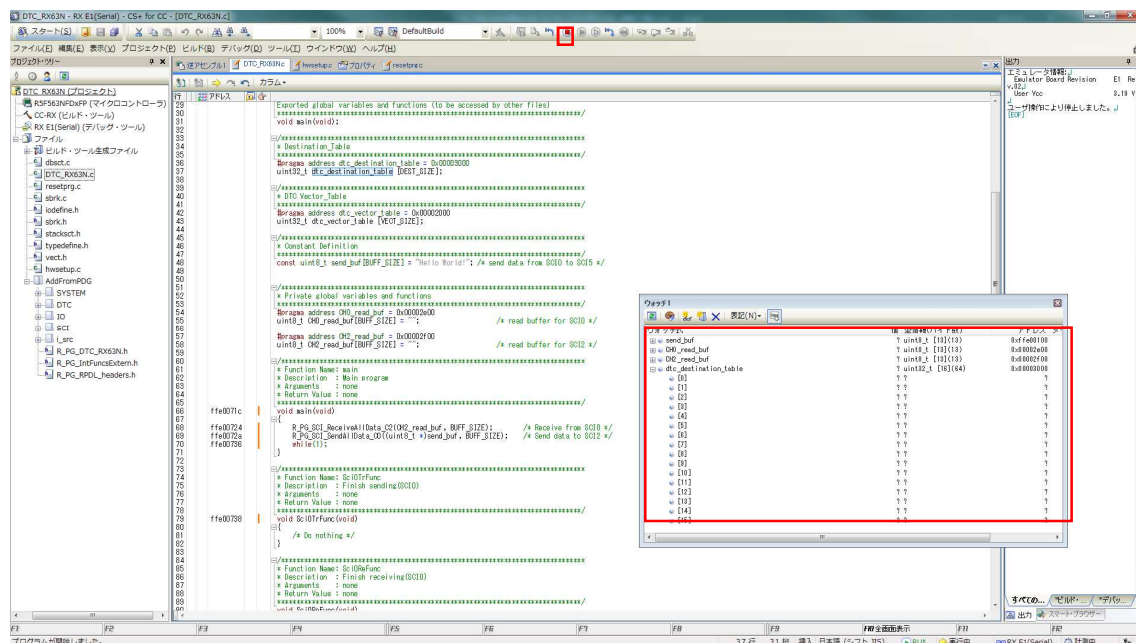
上の図のようにウォッチパネルには“send_buf”にのみ“Hello World!”が表示され、“CH2_read_buf”と“CH0_read_buf”は空欄であることが確認できます。ボード側は、LED1は消灯となっています。この状態で実行前の準備は完了です。

7.2 実行

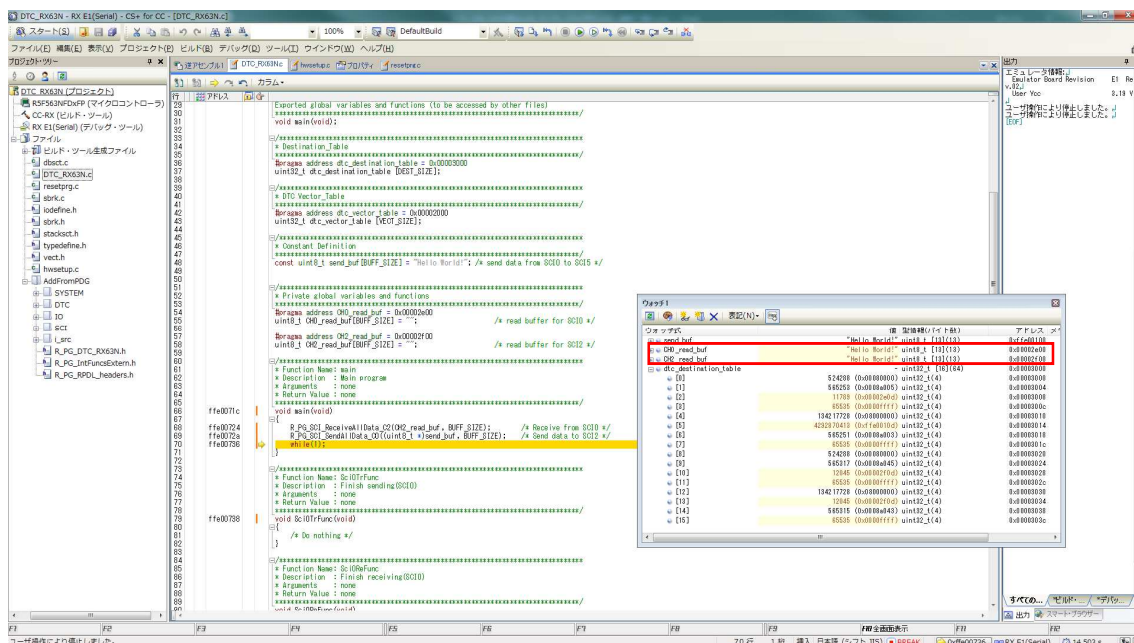
実行ボタンをクリックします。

実行中はウォッチパネルのウォッチ式の値は全て“?”マーク表示です。なお、LED1は点灯しました。

ここで、停止をクリックします。



プログラムを停止すると、ウォッチパネルで“CH2_read_buf”と“CH0_read_buf”は共に“Hello World!”が表示されています。これで SCI2 は SCI0 から“Hello World!”を受信できており、SCI0 は SCI2 から送り返された“Hello World!”を受信できたことが確認できました。



なお、“dtc_destinatin_table”にも変化が見られるため、確認します。実行前と実行後のウォッチを拡大します。

実行前のウォッチは次の通りです。

ウォッチ式	値	型情報(バイト数)	アドレス
send_buf	"Hello World!"	uint8_t 13	0xffe00100
CH0_read_buf	"	uint8_t 13	0x00002e00
CH2_read_buf	"	uint8_t 13	0x00002f00
dtc_destination_table		uint32_t [16](64)	
[0]	524288 (0x00080000)	uint32_t(4)	
[1]	565253 (0x0008a005)	uint32_t(4)	
[2]	11776 (0x00002e00)	uint32_t(4)	
[3]	917503 (0x000dffff)	uint32_t(4)	
[4]	134217728 (0x08000000)	uint32_t(4)	
[5]	4292870400 (0xffe00100)	uint32_t(4)	
[6]	565251 (0x0008a003)	uint32_t(4)	
[7]	917503 (0x000dffff)	uint32_t(4)	
[8]	524288 (0x00080000)	uint32_t(4)	
[9]	565317 (0x0008a045)	uint32_t(4)	
[10]	12032 (0x00002f00)	uint32_t(4)	
[11]	917503 (0x000dffff)	uint32_t(4)	
[12]	134217728 (0x08000000)	uint32_t(4)	
[13]	12032 (0x00002f00)	uint32_t(4)	
[14]	565315 (0x0008a043)	uint32_t(4)	
[15]	917503 (0x000dffff)	uint32_t(4)	

Callouts:

- RX10 転送情報(実行前)
- TX10 転送情報(実行前)
- RX12 転送情報(実行前)
- TX12 転送情報(実行前)

実行後は次の通りです。RX10 の転送情報においては 13 バイト分のデータ受信により転送先アドレスはインクリメントされ、2e00h から 2e0dh になりました。転送回数はデクリメントされるため、dh(13 回)から 0 になりました。TX10、RX12、TX12 それぞれの転送情報も所定の通り更新されており、DTC は正常に機能していることが確認できました。

ウォッチ式	値	型情報(バイト数)	アドレス
send_buf	"Hello World!"	uint8_t 13	0xffe00100
CH0_read_buf	"Hello World!"	uint8_t 13	0x00002e00
CH2_read_buf	"Hello World!"	uint8_t 13	0x00002f00
dtc_destination_table		uint32_t [16](64)	
[0]	524288 (0x00080000)	uint32_t(4)	
[1]	565253 (0x0008a005)	uint32_t(4)	
[2]	11789 (0x00002e0d)	uint32_t(4)	
[3]	65535 (0x0000ffff)	uint32_t(4)	
[4]	134217728 (0x08000000)	uint32_t(4)	
[5]	4292870413 (0xffe0010d)	uint32_t(4)	
[6]	565251 (0x0008a003)	uint32_t(4)	
[7]	65535 (0x0000ffff)	uint32_t(4)	
[8]	524288 (0x00080000)	uint32_t(4)	
[9]	565317 (0x0008a045)	uint32_t(4)	
[10]	12045 (0x00002f0d)	uint32_t(4)	
[11]	65535 (0x0000ffff)	uint32_t(4)	
[12]	134217728 (0x08000000)	uint32_t(4)	
[13]	12045 (0x00002f0d)	uint32_t(4)	
[14]	565315 (0x0008a043)	uint32_t(4)	
[15]	65535 (0x0000ffff)	uint32_t(4)	

Callouts:

- RX10 転送情報(実行後)
転送先アドレスが 2e0dh に
転送回数が 0 に変化した
- TX10 転送情報(実行後)
転送元アドレスが ffe0010dh に
転送回数が 0 に変化した
- RX12 転送情報(実行後)
転送先アドレスが 2f0dh に
転送回数が 0 に変化した
- TX12 転送情報(実行後)
転送元アドレスが 2f0dh に
転送回数が 0 に変化した

8. 参考ドキュメント

RX63N グループ、RX631 グループ ユーザーズマニュアルハードウェア編

RX63N グループ、RX631 グループ Peripheral Driver Generator リファレンスマニュアル

以上