

RX210 グループ

SCI を用いた調歩同期式通信

要旨

本サンプルコードでは、調歩同期式通信の初期設定から送信、別チャンネルで送信データを受信する方法について説明します。

対象デバイス

- RX210

内容

1.	仕様	3
2.	動作確認条件	3
3.	ハードウェア説明.....	4
3.1	使用端子一覧.....	4
3.2	接続信号	4
4.	ソフトウェア説明.....	5
4.1	動作概要	5
4.2	ファイル構成.....	6
4.3	オプション設定メモリ.....	7
4.4	定数一覧	7
4.5	変数一覧	8
4.6	関数一覧	8
4.7	関数仕様	9
4.8	作成する関数のフローチャート.....	11
4.8.1	初期設定.....	11
4.8.2	メイン処理.....	12
4.8.3	SCI0 送信完了処理	12
4.8.4	SCI0 受信完了処理	12
4.8.5	SCI5 受信完了処理	13
4.8.6	SCI5 送信完了処理	13
5.	PDG の設定	14
5.1	SYSTEM 設定	16
5.2	SCI0 設定	17
5.3	SCI5 設定	19
5.4	SYSTEM の端子設定	21
5.5	I/O 設定	22
5.6	ソースの生成.....	23
5.7	CS+への登録.....	24
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.....	26
7.	動作確認方法	29
7.1	ウォッチ式の登録.....	29
7.2	実行	31
8.	参考ドキュメント.....	33

1. 仕様

SCI0 から SCI5 へ調歩同期式通信でデータを送信します。SCI5 は受信したデータを SCI0 に送り返します。SCI0 が SCI5 へ送信したデータと SCI5 が SCI0 へ送り返したデータが一致した場合、LED (D3) が点灯します。

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	R5F5210BBDFP (RX210 グループ)
動作周波数	・メインクロック : 20MHz ・動作周波数 : 20MHz
ボード電源電圧	5V
マイコン動作電圧	5V
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	スリープモード
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 HSB RX210-100B (R5F5210BBDFP)

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
PA4	出力	TXD5
PA2	入力	RXD5
PH2	出力	D3(LED)

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	J1 の 34pin (P20/TXD0)	J2 の 6pin (PA2/RXD5)
接続 2	J2 の 4pin (PA4/TXD5)	J1 の 33pin (P21/RXD0)

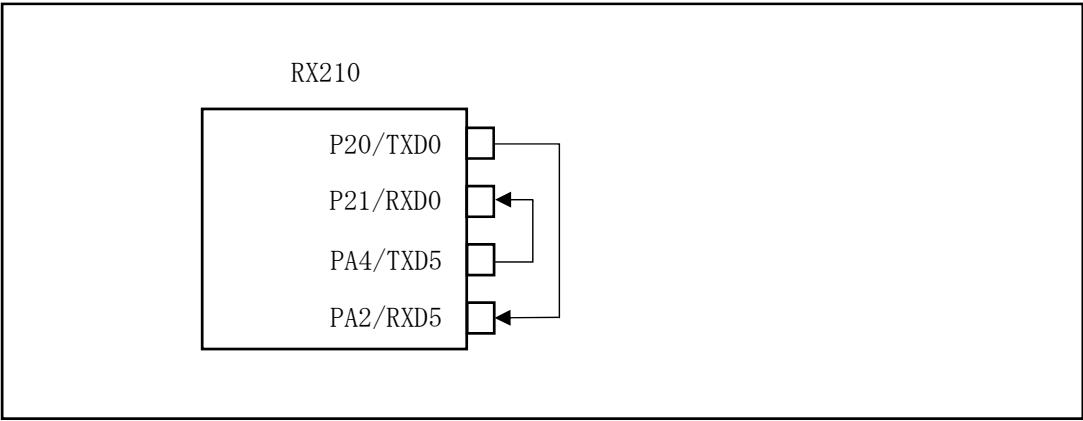


図 3.1 信号接続図

4. ソフトウェア説明

4.1 動作概要

図 4.1 に動作タイミング図を、以下に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、SCI5 の初期設定を行い、TXD0、TXD5 端子は” H” 出力となります。

(2) SCI5 の受信開始

指定データサイズを受信を開始します。

(3) SCI0 の送信開始

SCI0 から” Hello World!” という文字列を送信します。

(4) SCI0 の送信完了および受信開始

指定データサイズを受信を開始します。

(5) SCI5 の受信完了および送信開始

SCI5 の受信データをそのまま SCI5 から送信します。

(6) SCI0 の受信完了

SCI0 で SCI5 へ送信したデータと同じ” Hello World!” が受信できれば、送受信が正しくできていると判断し、D3(LED)が点灯します。

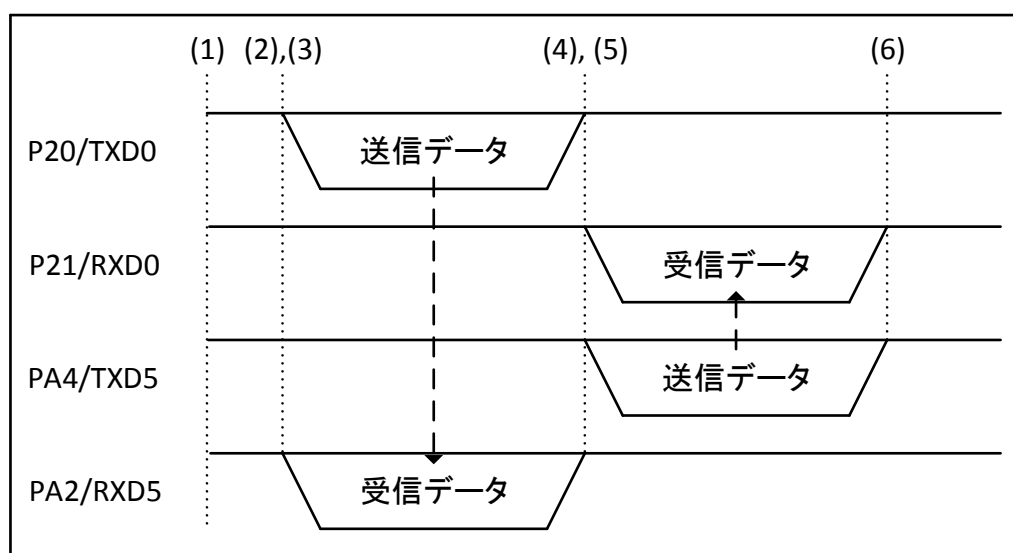


図 4.1 動作タイミング

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
SCI_RX210. c	メインファイル <ul style="list-style-type: none"> ・ SCI0 送信/受信処理 ・ SCI5 送信/受信処理 ・ D3 (LED) 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 ・ SCI0 の設定 ・ SCI5 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

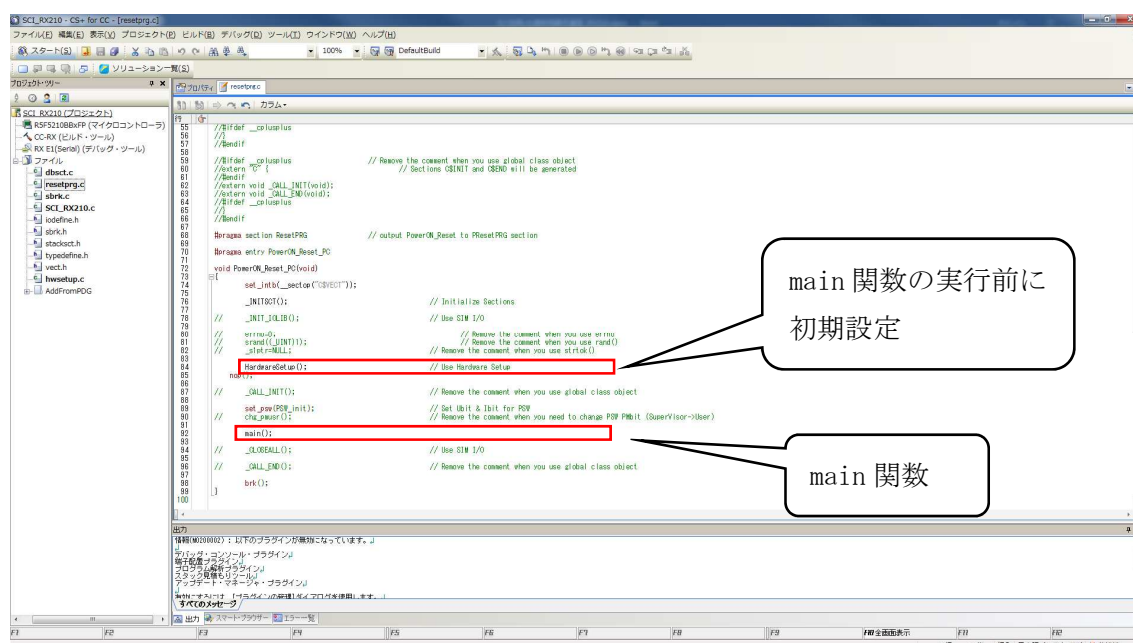


图 4.2 resetprg.c

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

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	CH5_read_buf[BUFF_SIZE]	SCI5 の受信バッファ	main Sci5ReFunc

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用している PDG で生成された関数に関しましては、「RX210 グループ、RX210 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 送信完了関数
Sci0ReFunc	SCI0 受信完了関数
Sci5ReFunc	SCI5 受信完了関数
Sci5TrFunc	SCI5 送信完了関数

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	SCI5 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 送信完了処理
ヘッダ	なし
宣言	void Sci0TrFunc(void)
説明	SCI0 の受信開始
引数	なし
リターン値	なし

Sci0RdFunc

概要	SCI0 受信完了処理
ヘッダ	なし
宣言	void Sci0RdFunc(void)
説明	送信データと受信データの比較
引数	なし
リターン値	なし

Sci5ReFunc

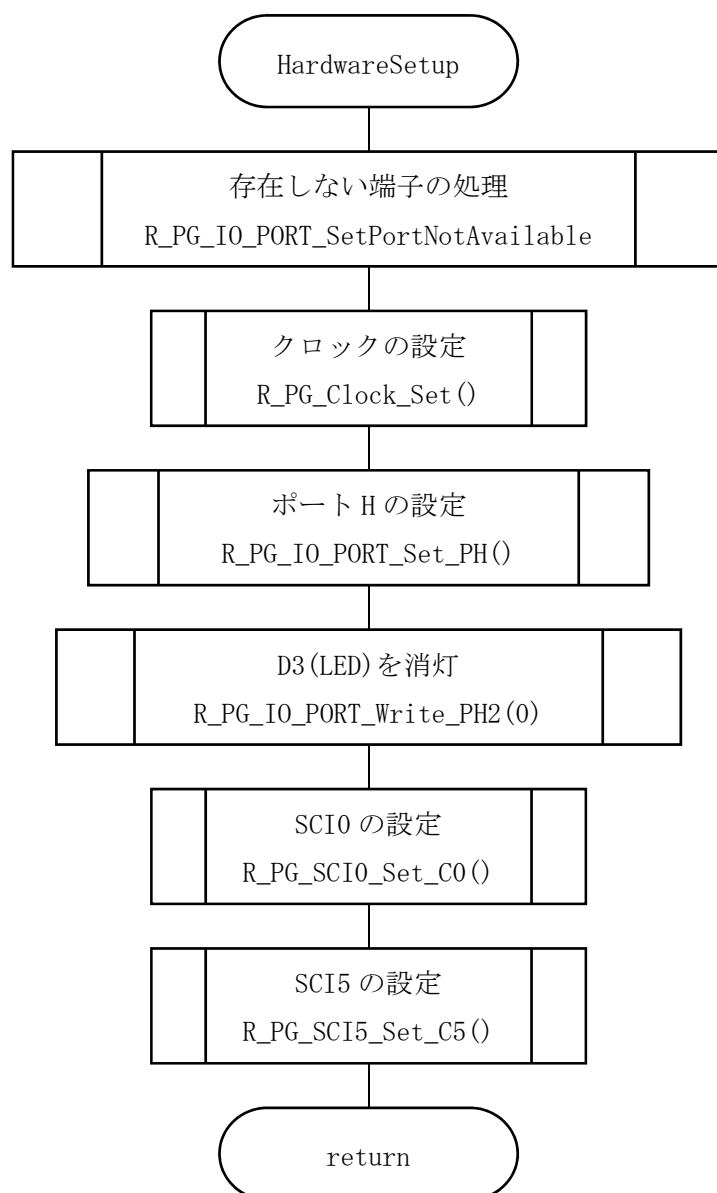
概要	SCI5 受信完了処理
ヘッダ	なし
宣言	void Sci5RdFunc(void)
説明	SCI5 の受信データを送信開始
引数	なし
リターン値	なし

Sci5TrFunc

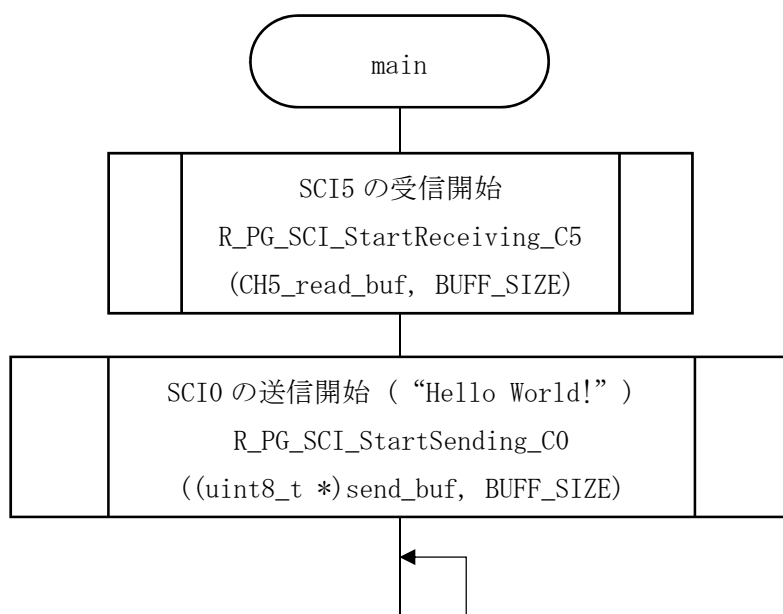
概要	SCI5 送信完了処理
ヘッダ	なし
宣言	void Sci5TrFunc(void)
説明	処理なし
引数	なし
リターン値	なし

4.8 作成する関数のフローチャート

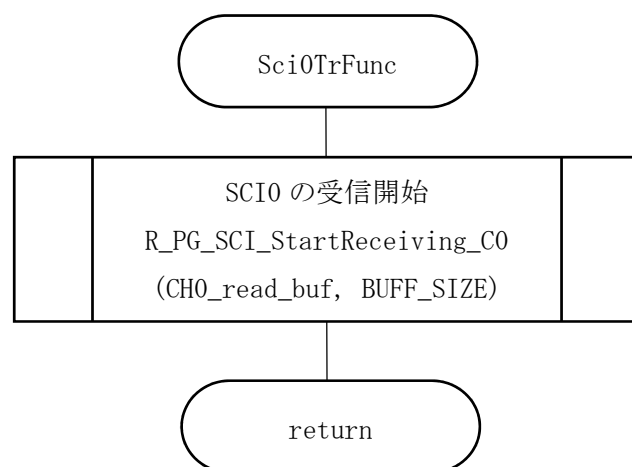
4.8.1 初期設定



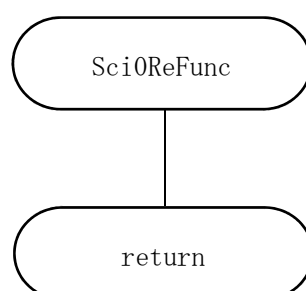
4.8.2 メイン処理



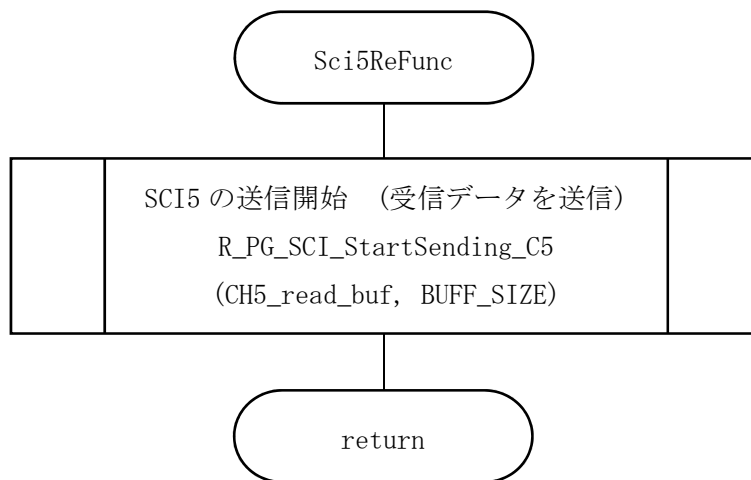
4.8.3 SCI0 送信完了処理



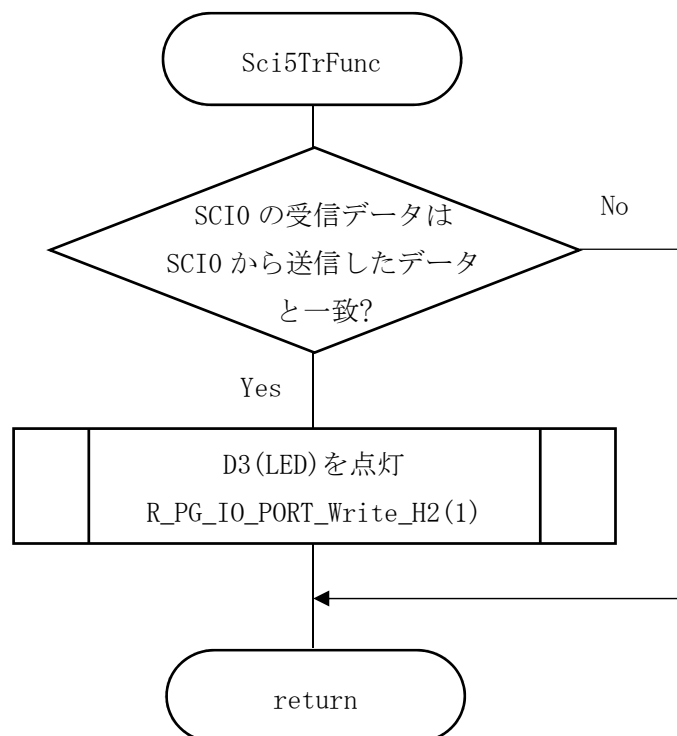
4.8.4 SCI0 受信完了処理



4.8.5 SCI5 受信完了処理



4.8.6 SCI5 送信完了処理



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”RX210 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:
SOI_RX210

ディレクトリ: ?
c:\renesas\PDG2_proj 参照...

デバイス選択

シリーズ: RX200

グループ: RX210

型名: R5F5210BBxFP

パッケージ: PLQP0100KB-A

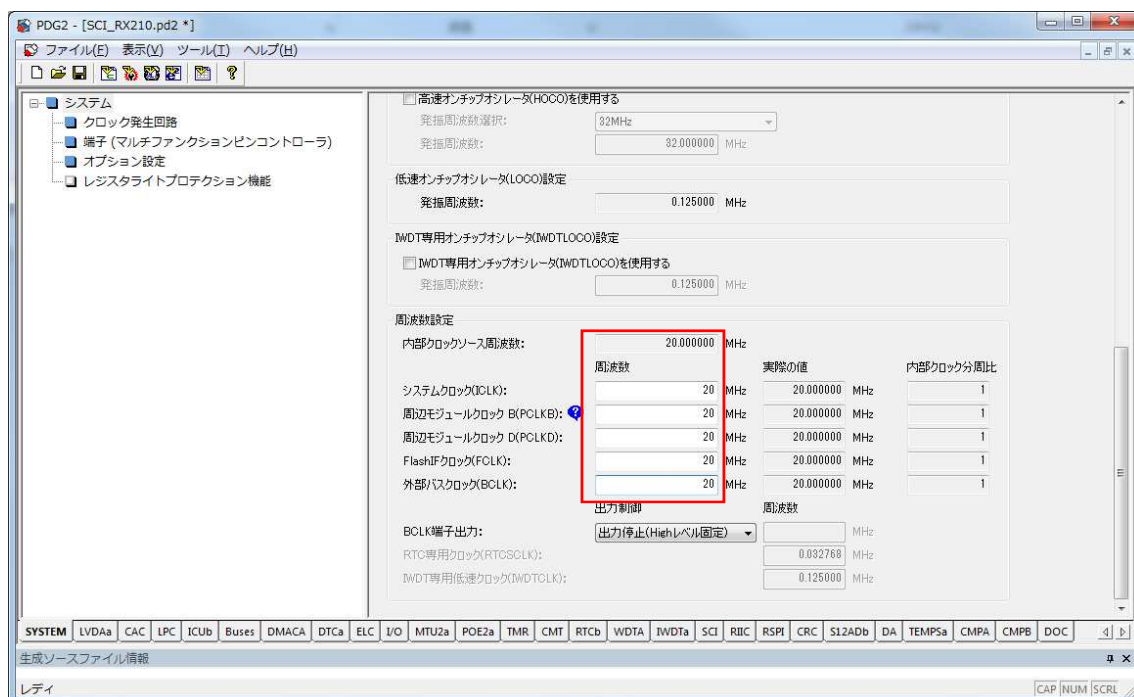
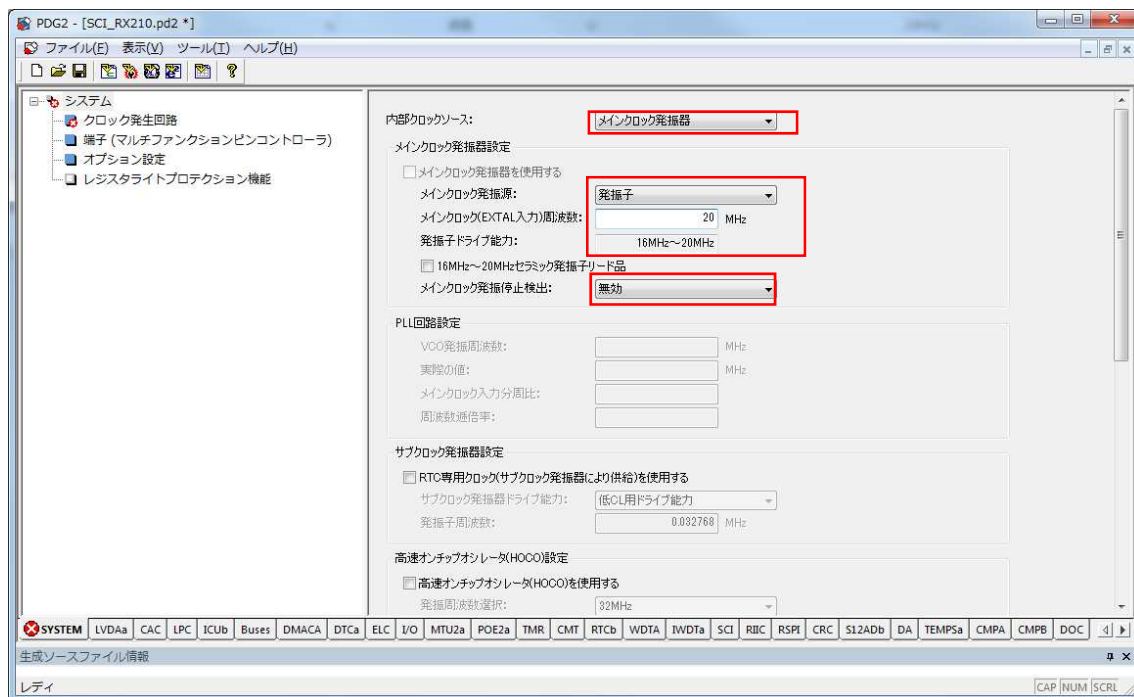
ROM容量: 1M バイト

RAM容量: 96K バイト

クリック OK キャンセル

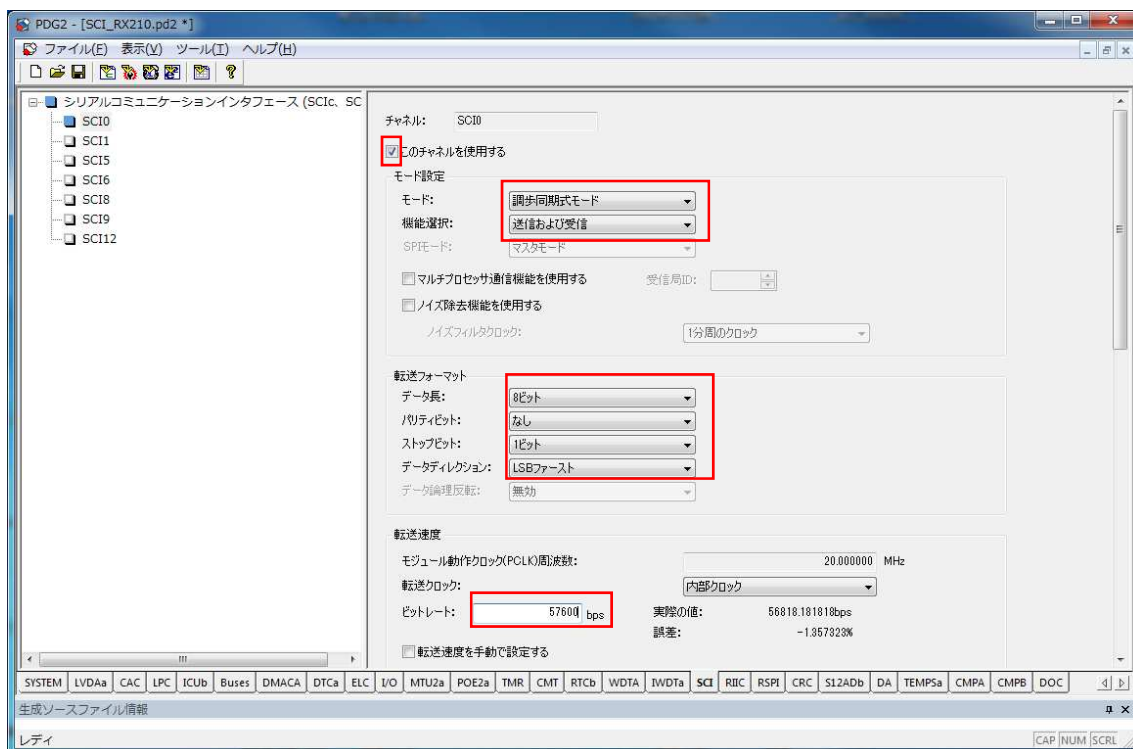
5.1 SYSTEM 設定

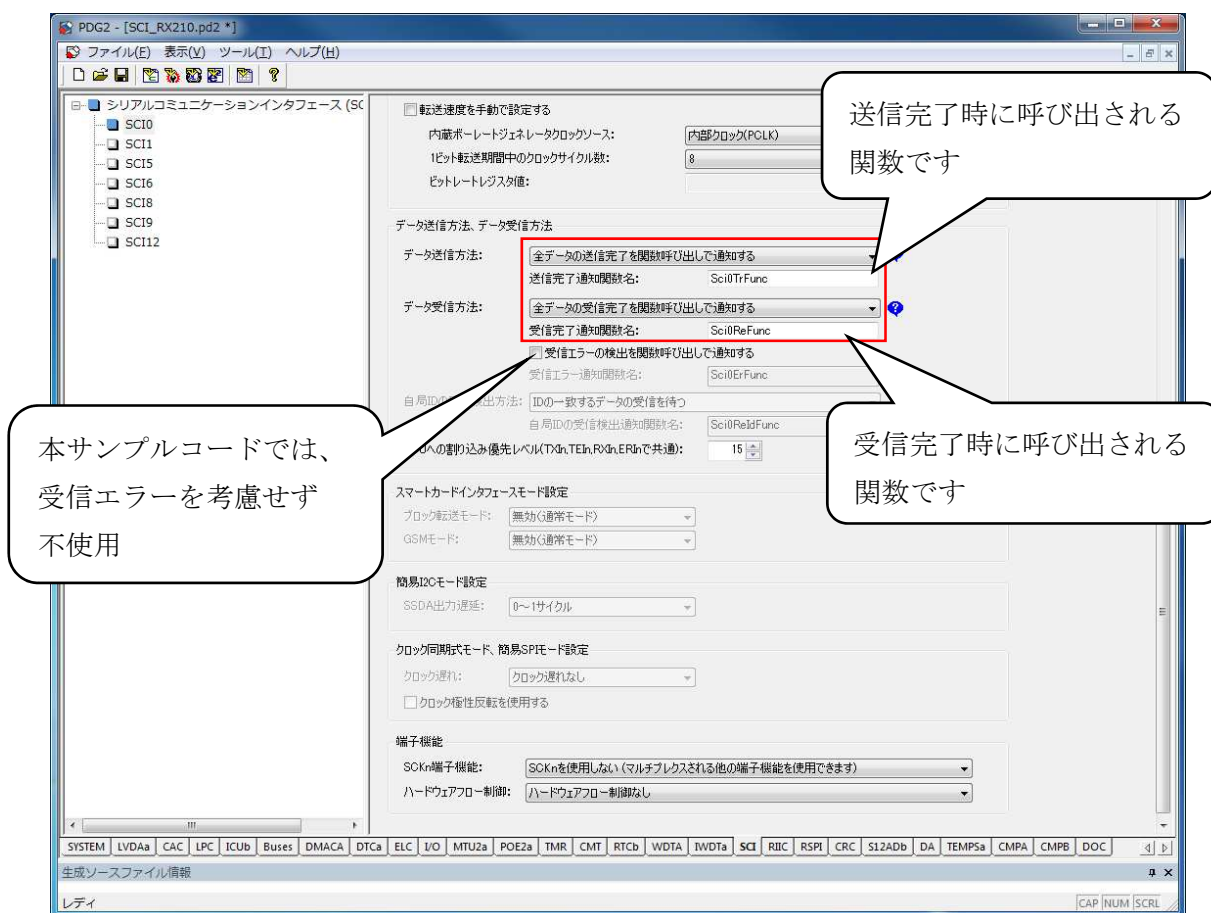
システムタブのクロック発生回路の設定を以下に示します。



5.2 SCI0 設定

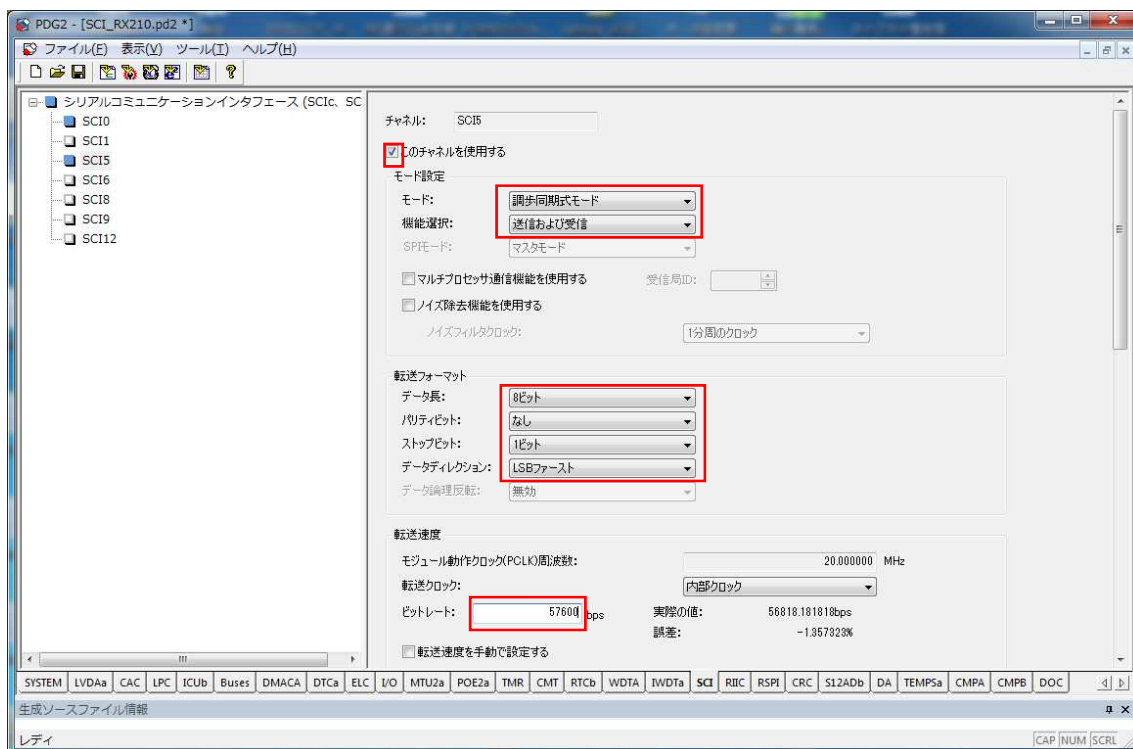
SCI0 の設定を以下に示します。

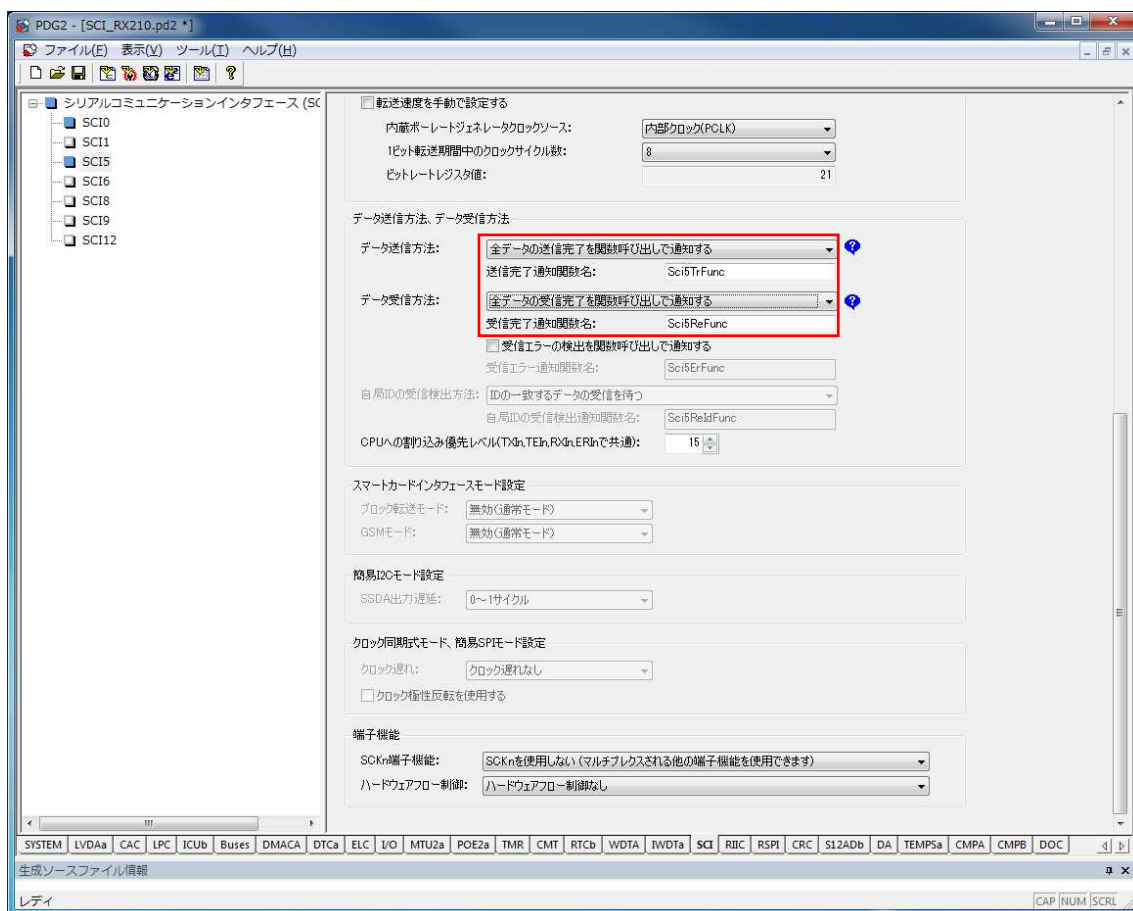




5.3 SCI5 設定

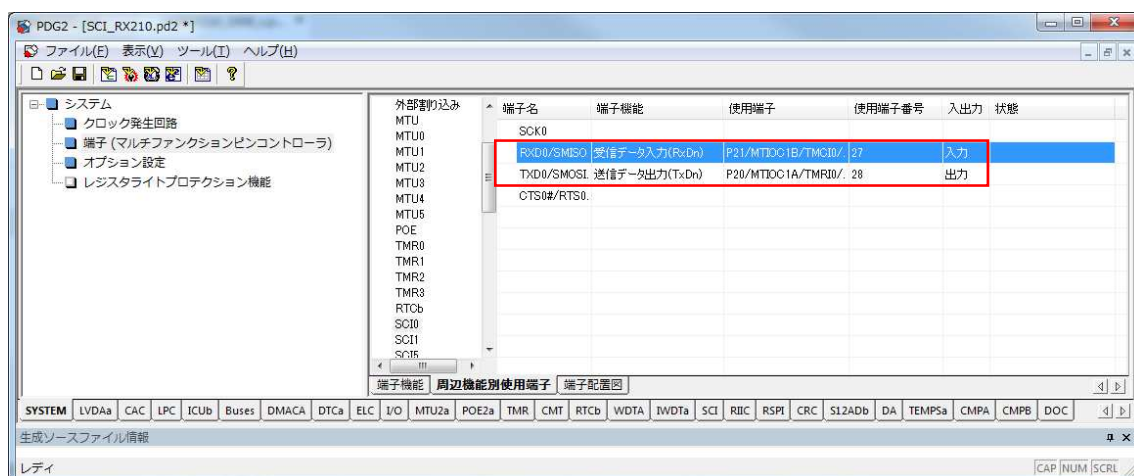
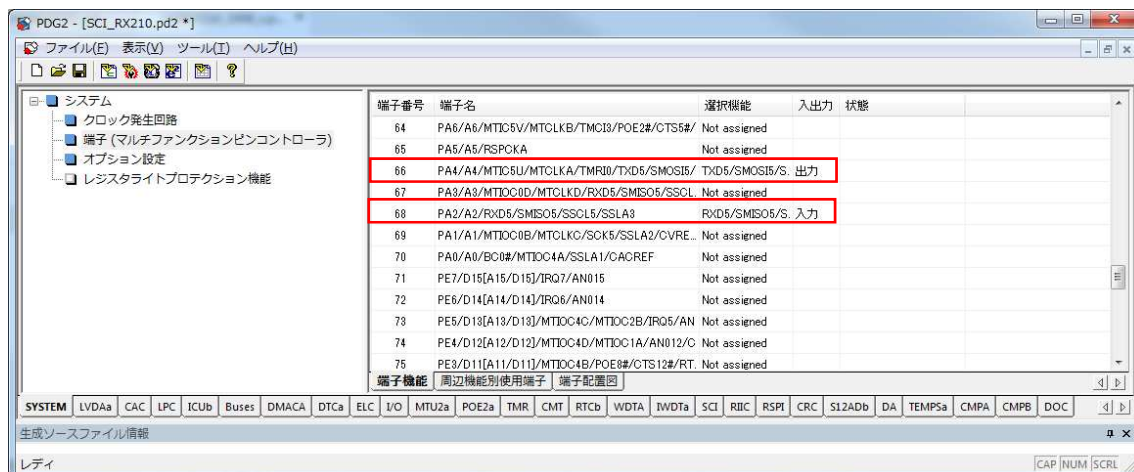
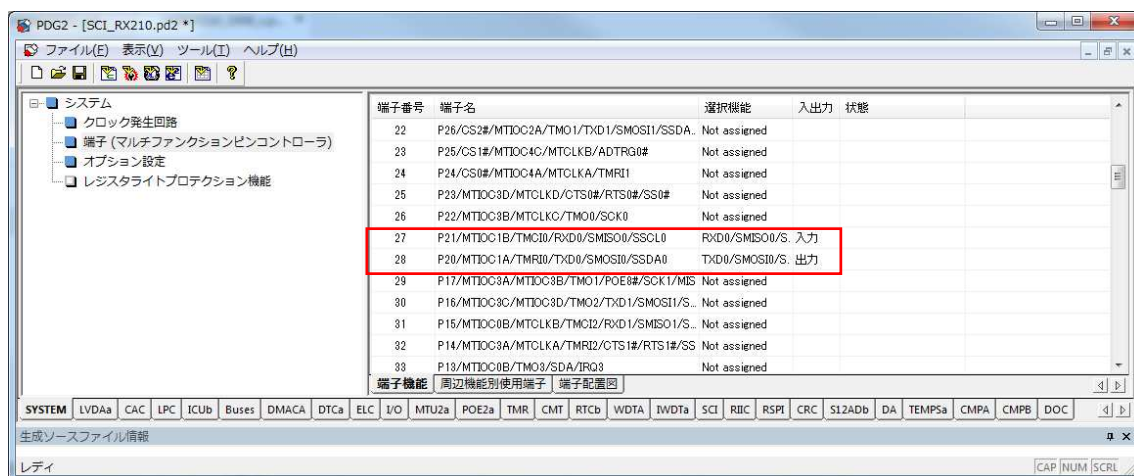
SCI5 の設定を以下に示します。(SCI0 と同様です)

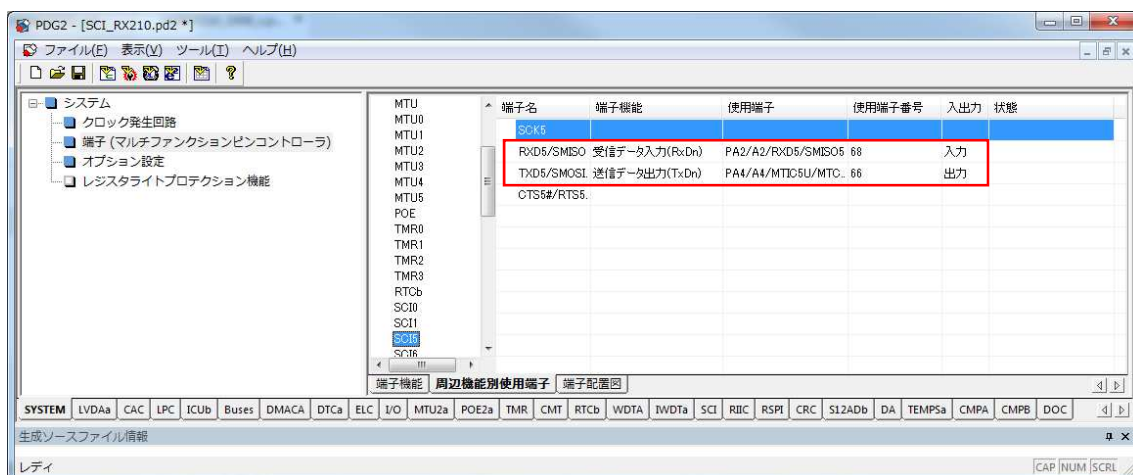




5.4 SYSTEM の端子設定

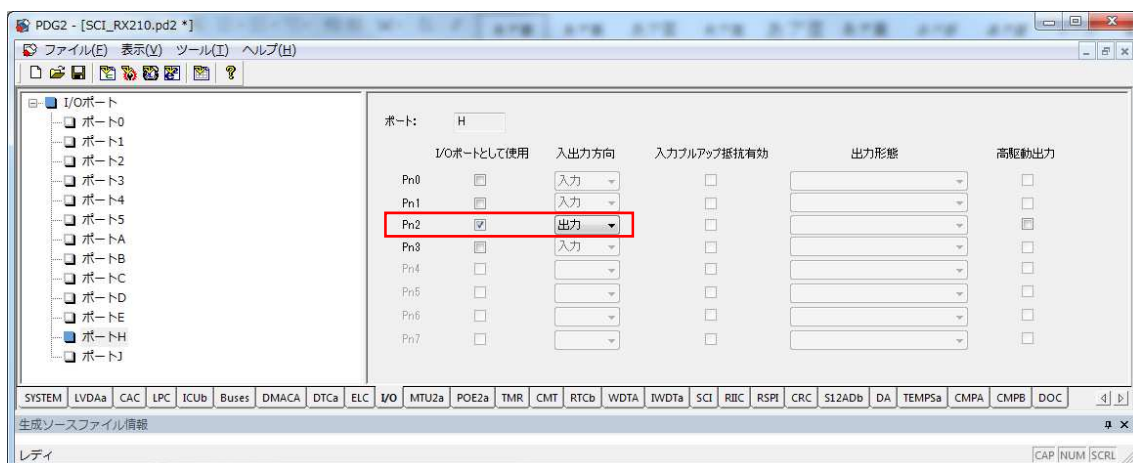
SYSTEM の端子設定を確認します。





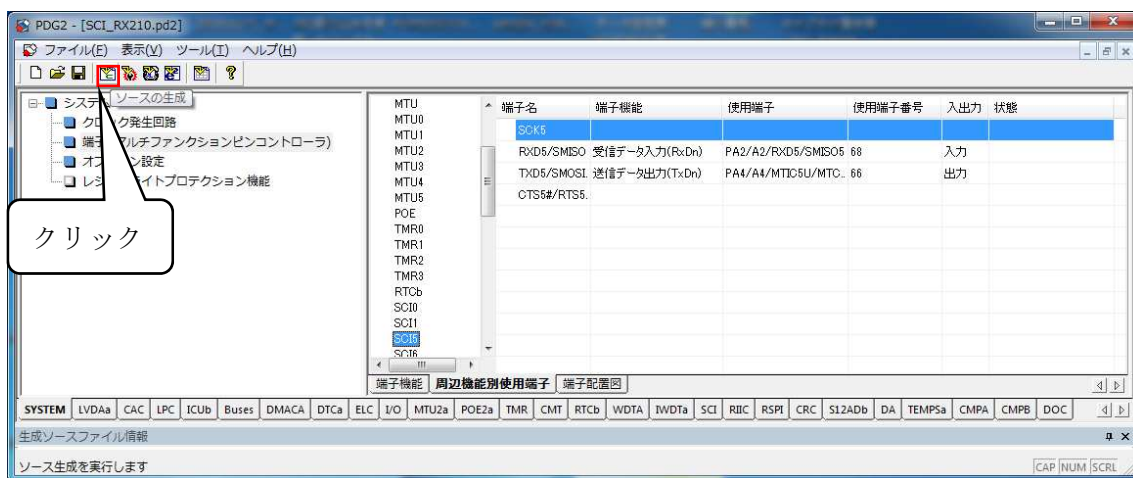
5.5 I/O 設定

I/O の設定を以下に示します。D3(LED)で使用する PH2 を出力にします。

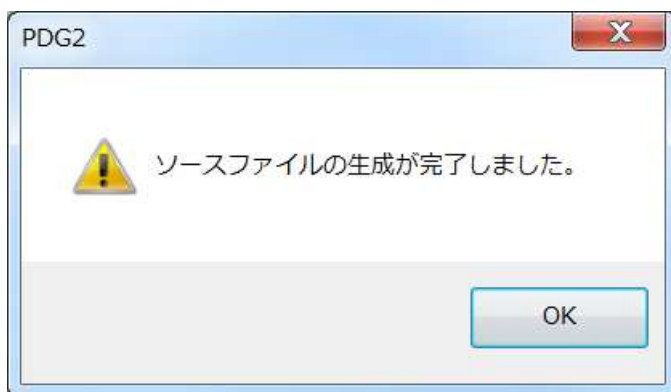


5.6 ソースの生成

以下の GUI をクリックすると、

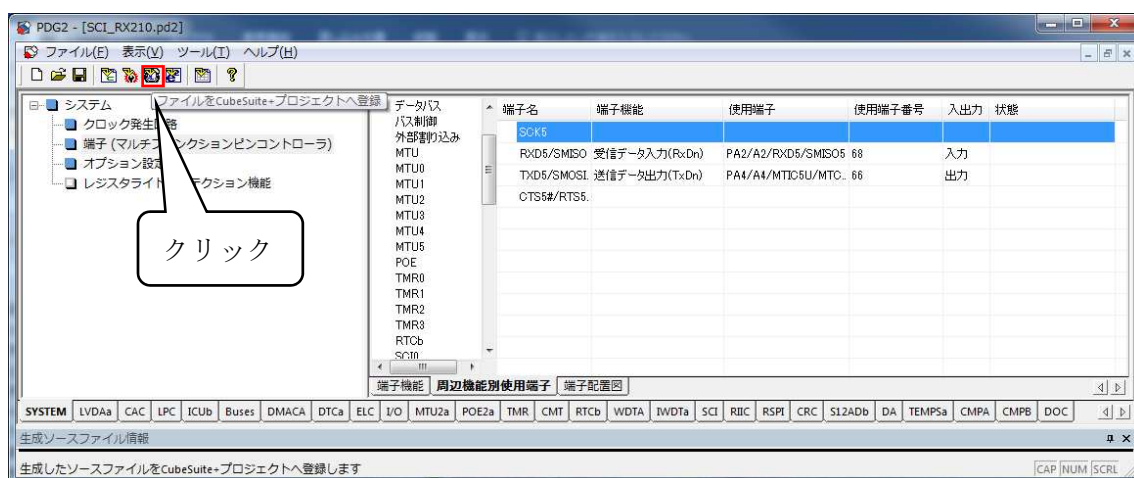


ソースファイルが生成されます。

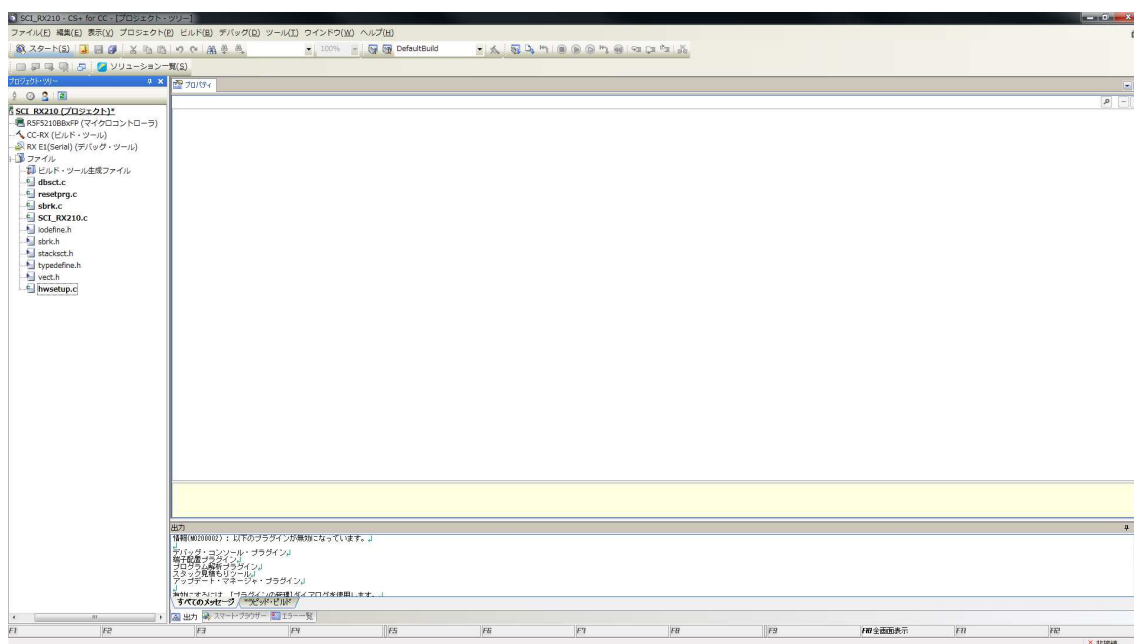


5.7 CS+への登録

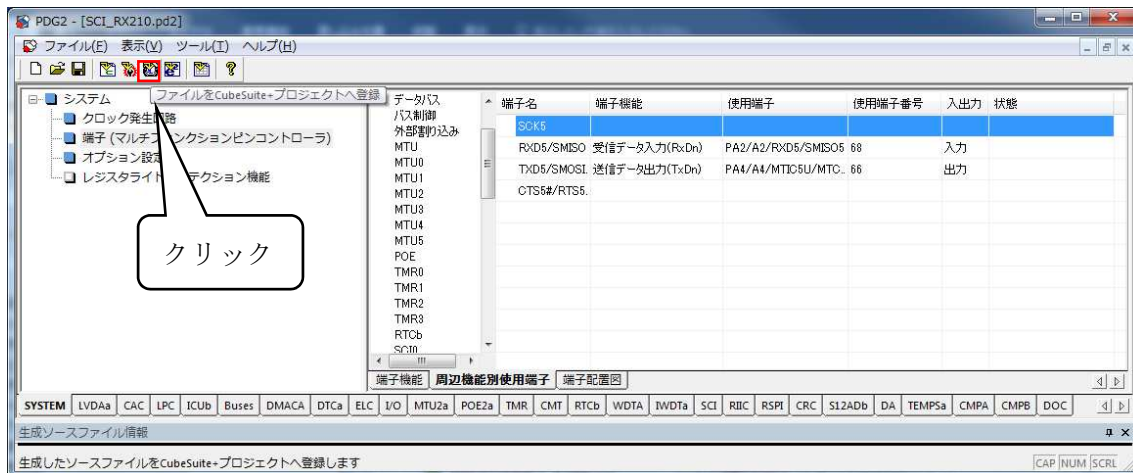
以下の GUI をクリックします。



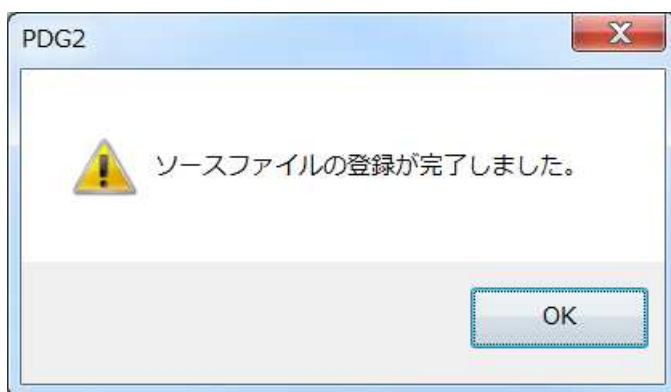
対象の CS+プロジェクトを開きます。



以下の GUI をクリックします

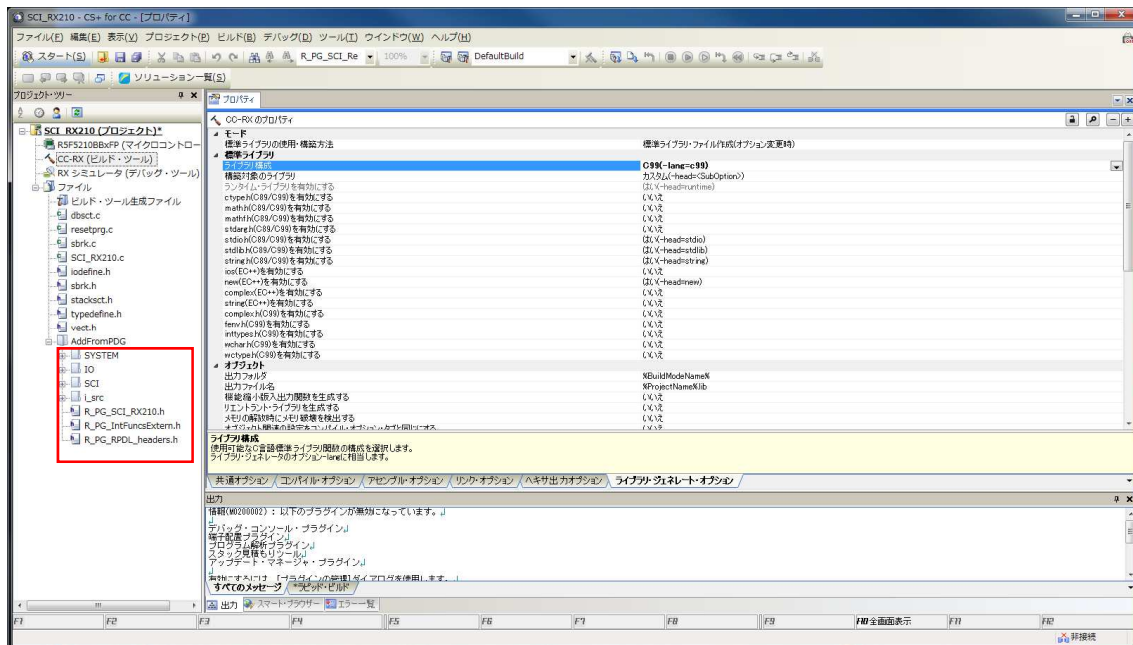


ソースファイルの登録が完了しました。



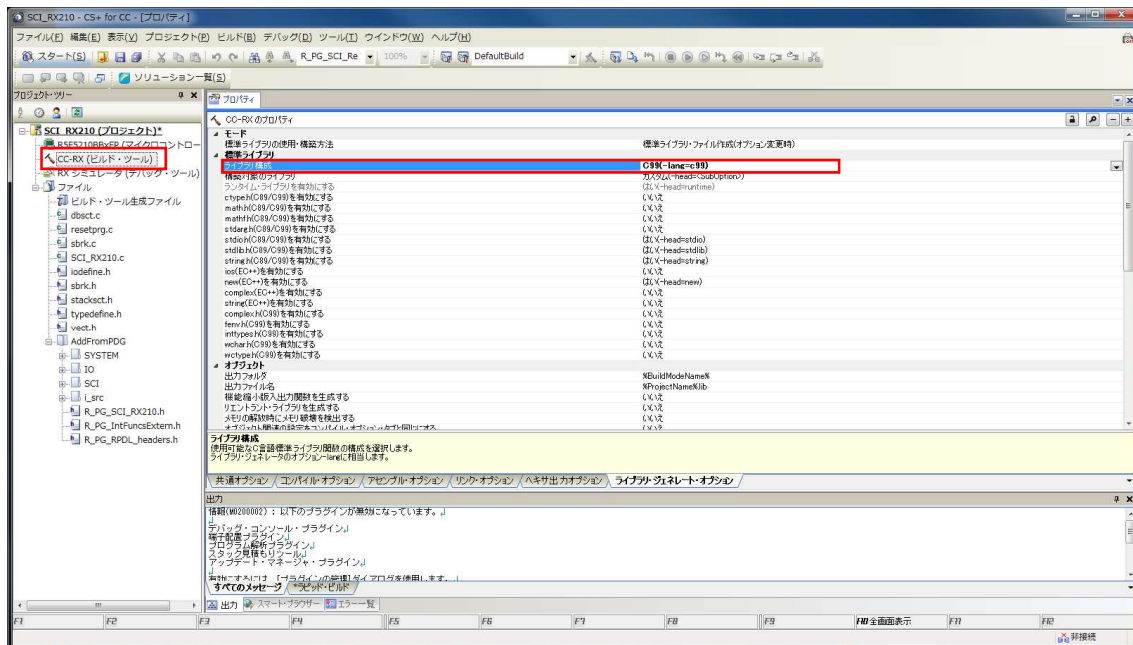
6. CS+のプロジェクトに PDG のソースファイルを登録する際の設定

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。

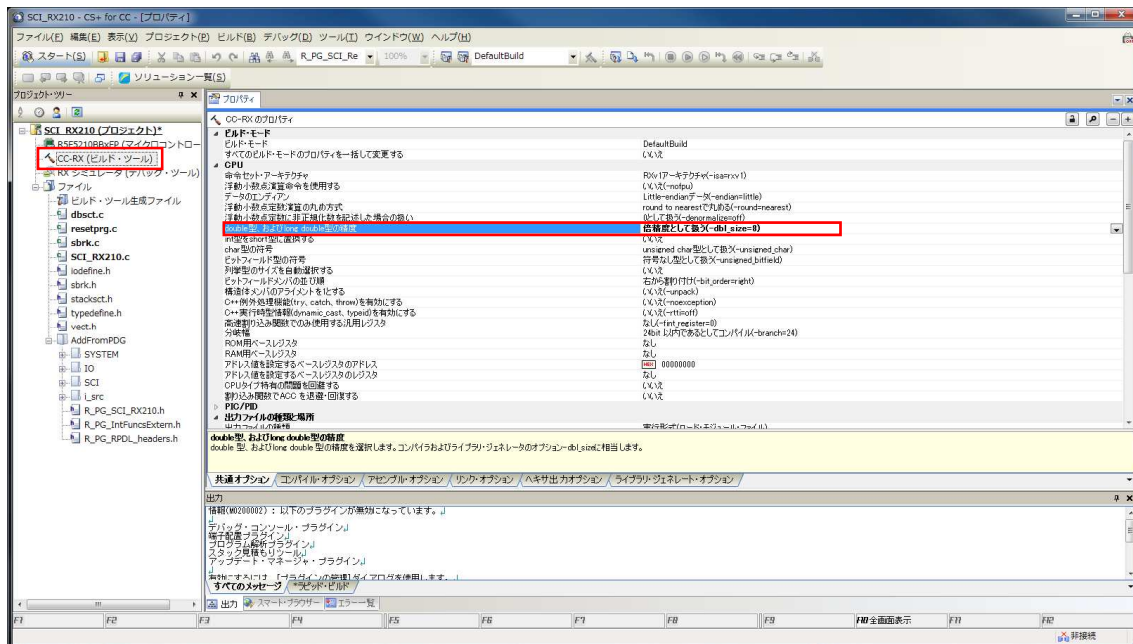


そのままビルドをすると、エラーおよび警告が発生します。解消する設定を以下に示します。

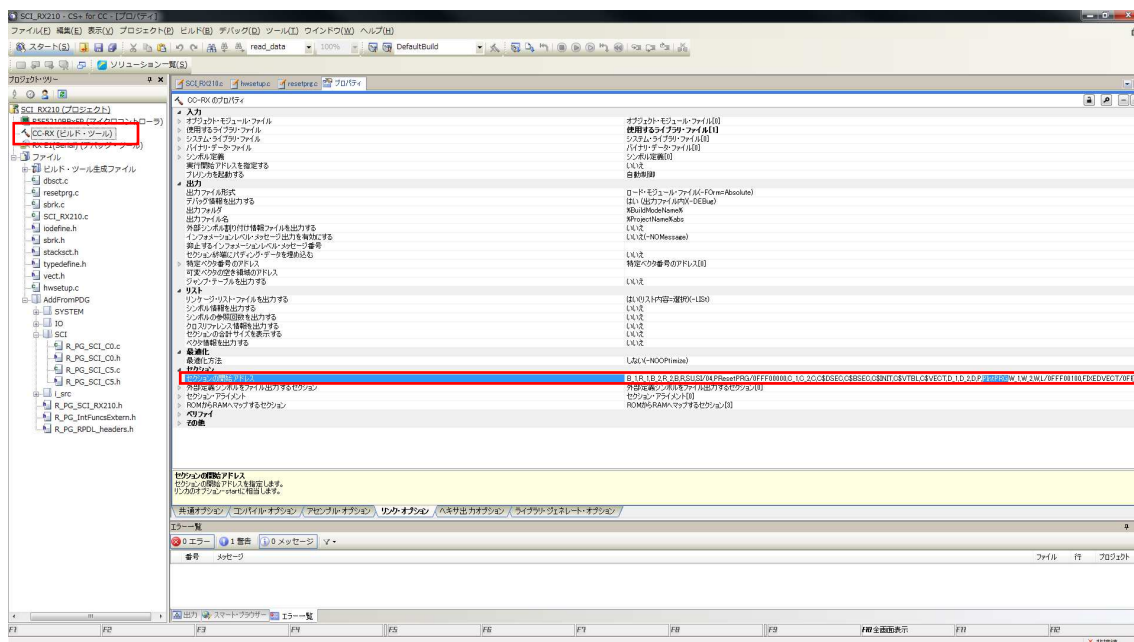
PDG で生成されるソースファイルは bool 変数を使用しています。対応させるため、ビルド・ツールを右クリック→プロパティを表示し、ライブラリ・ジェネレート・オプションタブにある「ライブラリ構成」を”C99(-lang=c99)”に設定します。



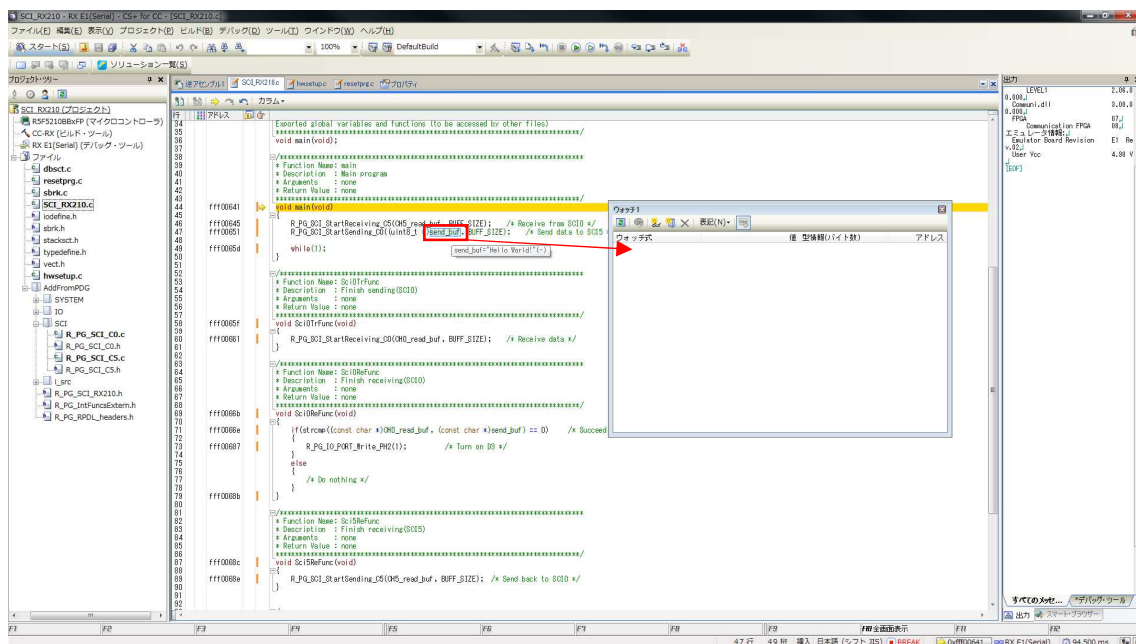
PDG で生成されるソースファイルは double 型、および long double 型の精度を倍精度として扱っているため、ビルド・ツールを右クリック→プロパティを表示し、共通オプションタブにある「double 型、および long double 型の精度」を”倍精度として扱う(-dbl_size=8)”に設定します。



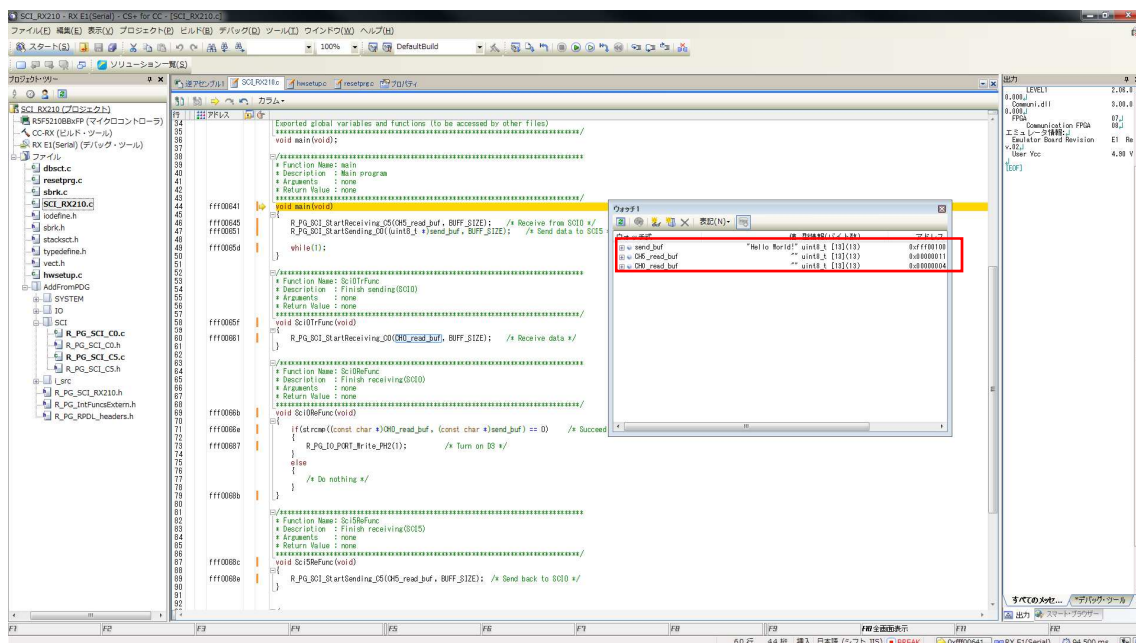
PDGで生成されるソースファイルを登録すると PIntPRG セクションを使用しないため、CS+プロジェクトを生成した際にデフォルトで設定されている PIntPRG セクションを削除します。ビルド・ツールを右クリック→プロパティを表示し、リンクオプションタブにある「セクションの開始アドレス」から” PIntPRG ” を削除します。



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。



“CH5_read_buf” と “CH0_read_buf” も同様にドラッグ&ドロップします。



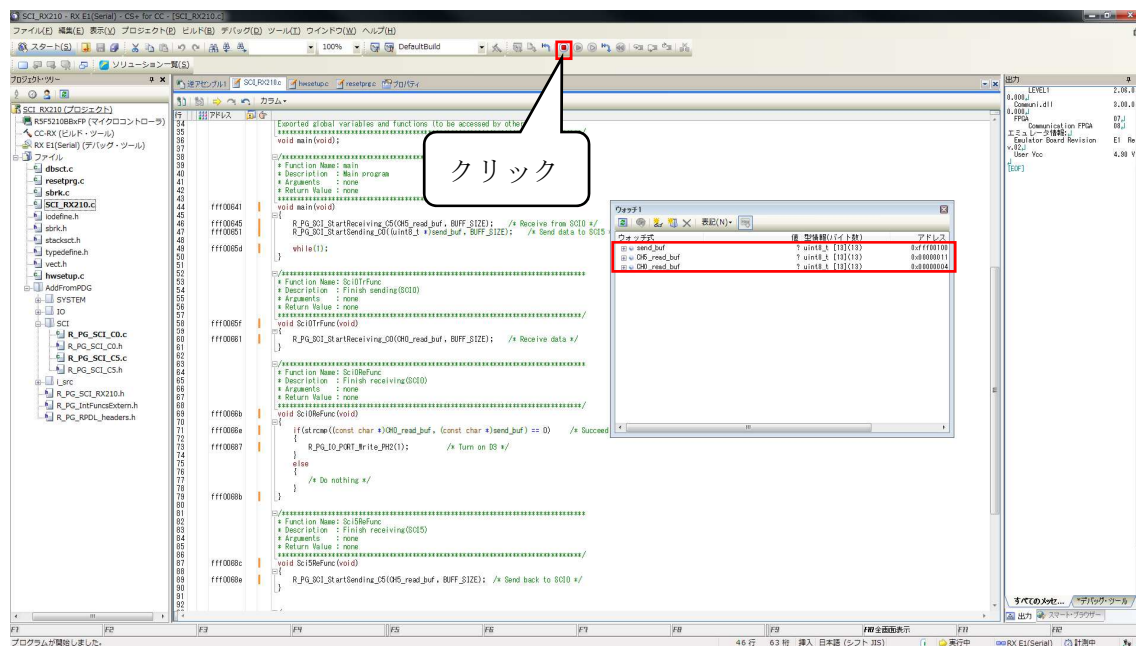
上の図のようにウォッチパネルには “send_buf” にのみ “Hello World!” が表示され、“CH5_read_buf” と “CH0_read_buf” は空欄であることが確認できます。ボード側は、D3(LED)が消灯となっています。この状態で実行前の準備は完了です。

7.2 実行

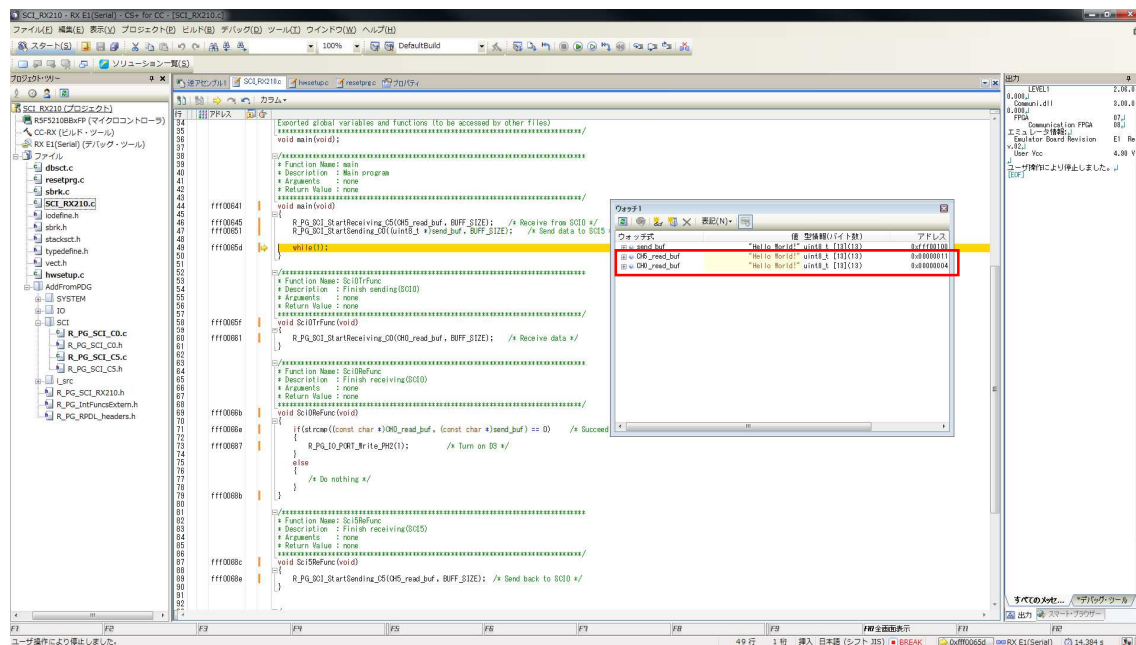
実行ボタンをクリックします。

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、D3(LED)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“CH5_read_buf”と“CH0_read_buf”は共に“Hello World!”が表示されています。これで SCI5 は SCI0 から“Hello World!”を受信できており、SCI0 は SCI5 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。



8. 参考ドキュメント

RX210 グループ ユーザーズマニュアルハードウェア編

RX210 グループ Peripheral Driver Generator リファレンスマニュアル

以上