RX63N 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

・RX63N

内容

1. 仕様

SCI0 から SCI2 へ調歩同期式通信でデータを送信します。SCI2 は受信したデータを SCI0 に送り返します。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。

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 逓倍） ・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)

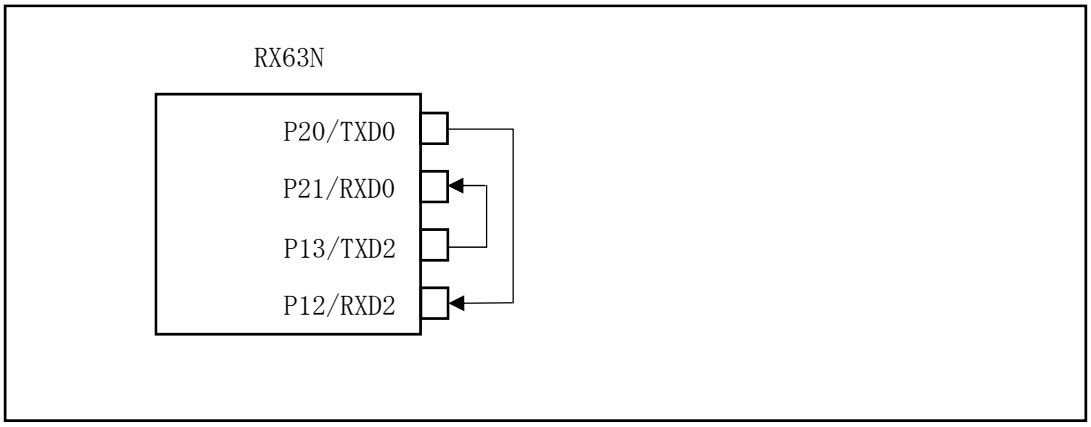


図 3.1 信号接続図

4. ソフトウェア説明

4.1 動作概要

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、SCI2 の初期設定を行い、TXD0、TXD2 端子は”H”出力となります。

(2) SCI2 の受信開始

指定データサイズの受信を開始します。

(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

指定データサイズの受信を開始します。

(5) SCI2 の受信完了および送信開始

SCI2 の受信データをそのまま SCI2 から送信します。

(6) SCI0 の受信完了

SCI0 で SCI2 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、LED1 が点灯します。

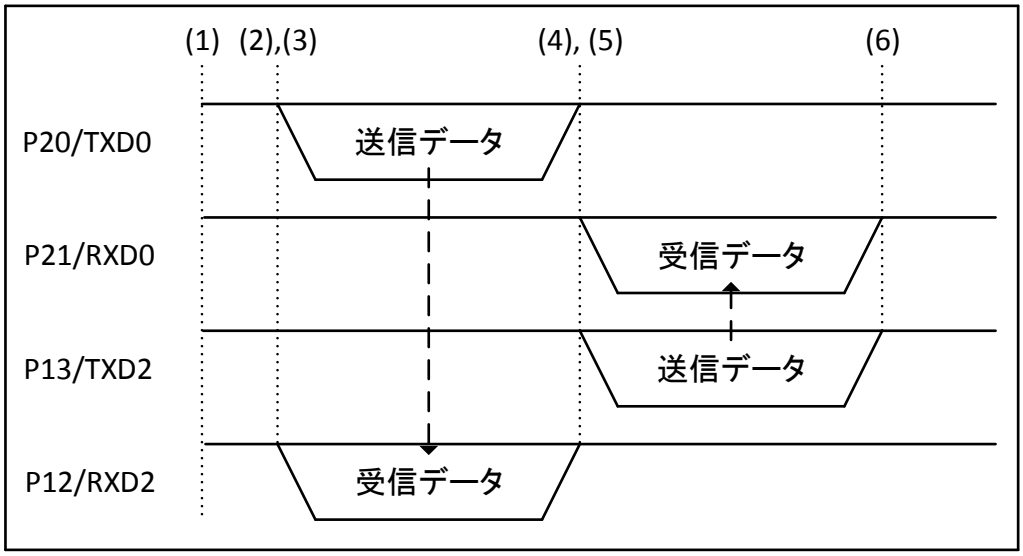


図 4.1 動作タイミング

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_RX63N.c	メインファイル ・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 ・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 ・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

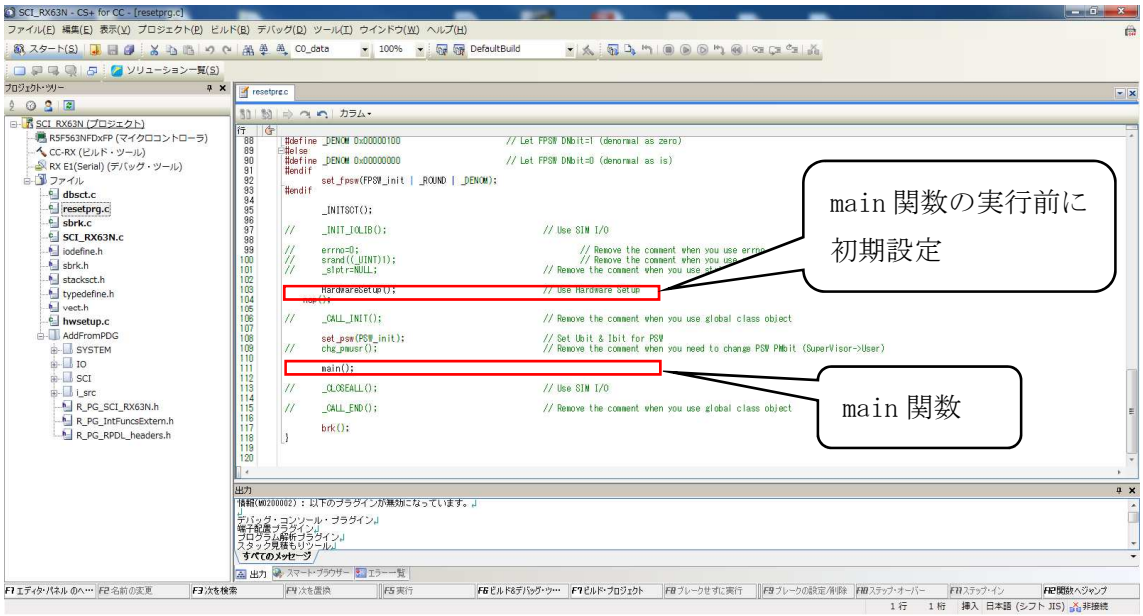


図 4.2　resetprg.c

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 HOCO(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	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用している 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 送信完了関数
Sci0ReFunc	SCI0 受信完了関数
Sci2ReFunc	SCI2 受信完了関数
Sci2TrFunc	SCI2 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	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

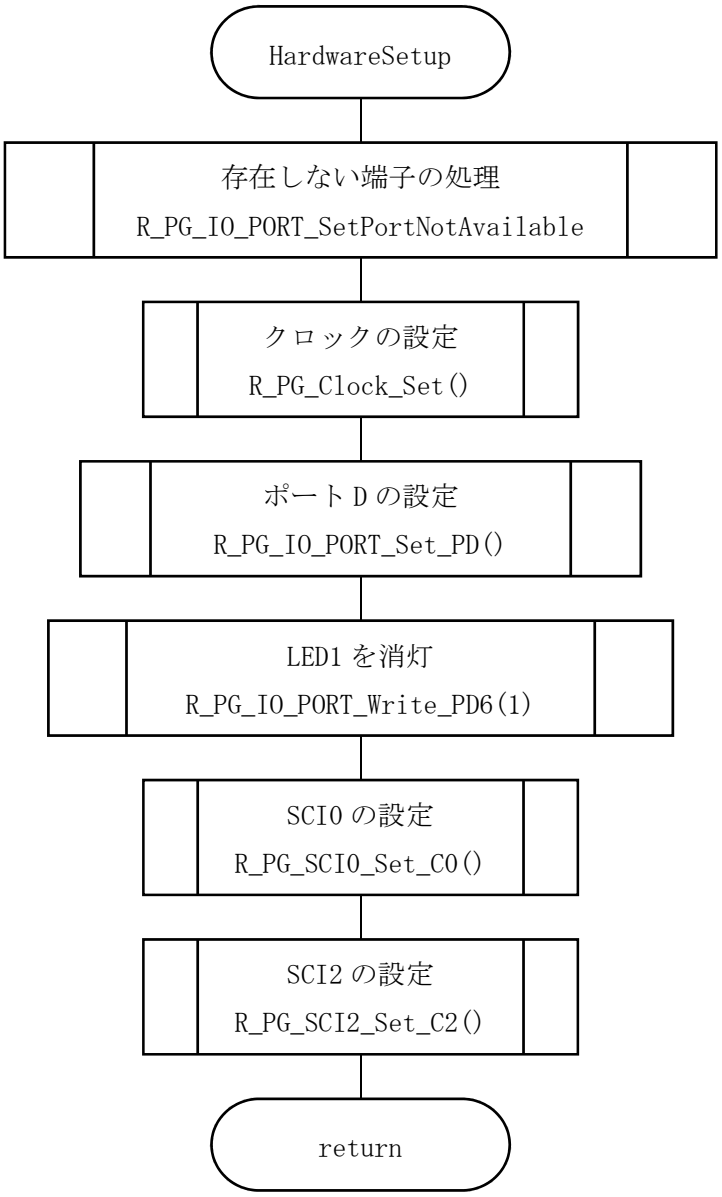
Sci0RdFunc	
概要	SCI0 受信完了処理
ヘッダ	なし
宣言	void Sci0RdFunc(void)
説明	送信データと受信データの比較
引数	なし
リターン値	なし

Sci2ReFunc	
概要	SCI2 受信完了処理
ヘッダ	なし
宣言	void Sci2RdFunc(void)
説明	SCI2 の受信データを送信開始
引数	なし
リターン値	なし

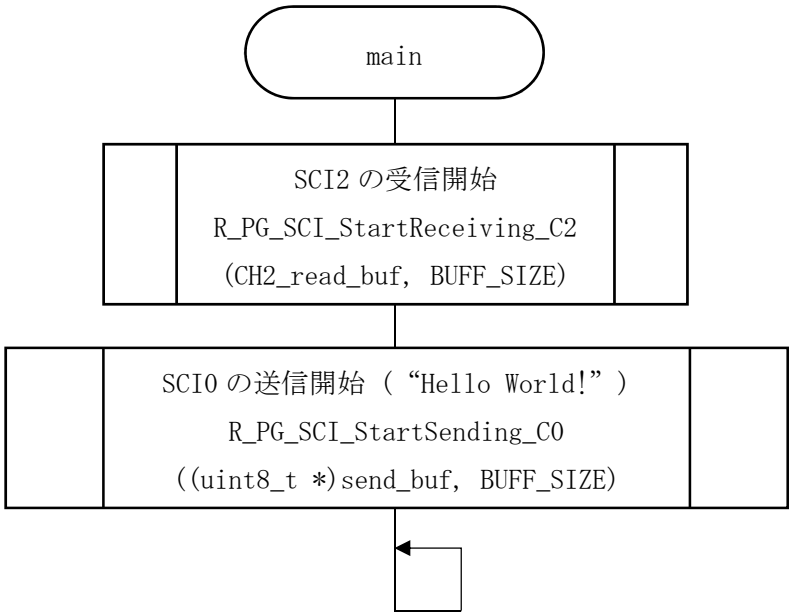
Sci2TrFunc	
概要	SCI2 送信完了処理
ヘッダ	なし
宣言	void Sci2TrFunc(void)
説明	処理なし
引数	なし
リターン値	なし

4.8　　作成する関数のフローチャート

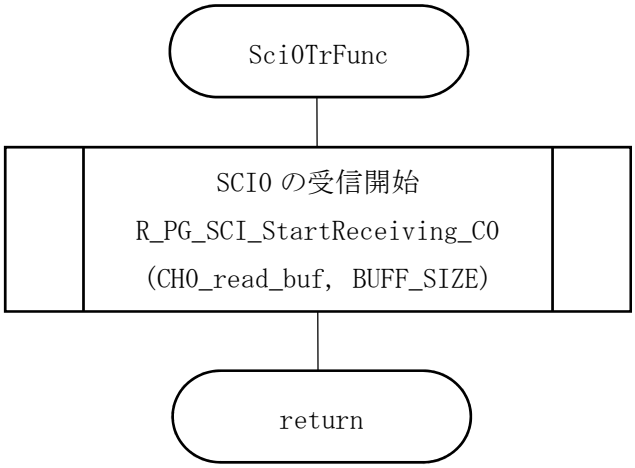
4.8.1　　初期設定



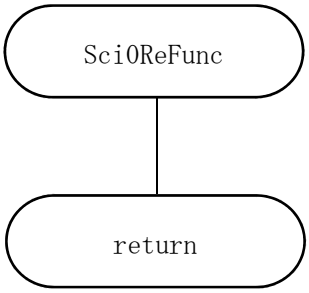
4.8.2　　メイン処理



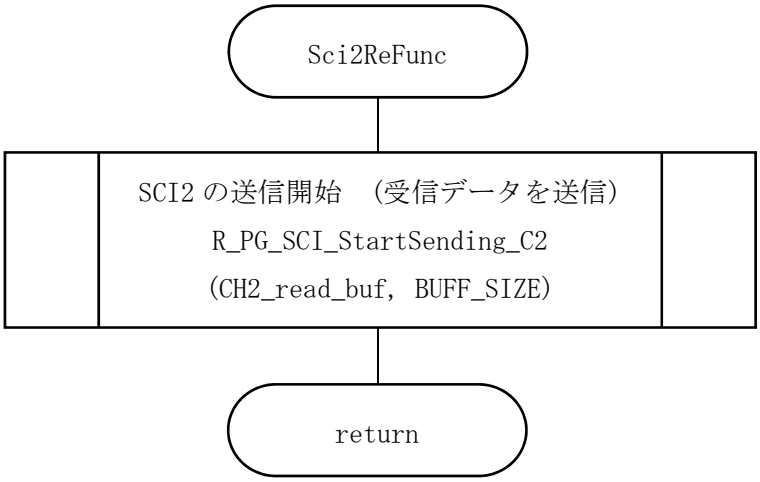
4.8.3　　SCI0 送信完了処理



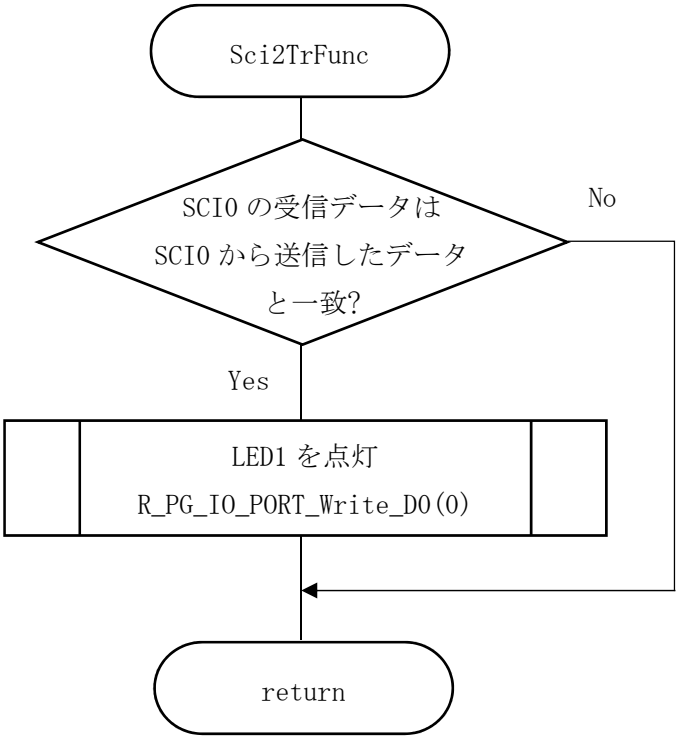
4.8.4　　SCI0 受信完了処理



4.8.5　　SCI2 受信完了処理



4.8.6　　SCI2 送信完了処理

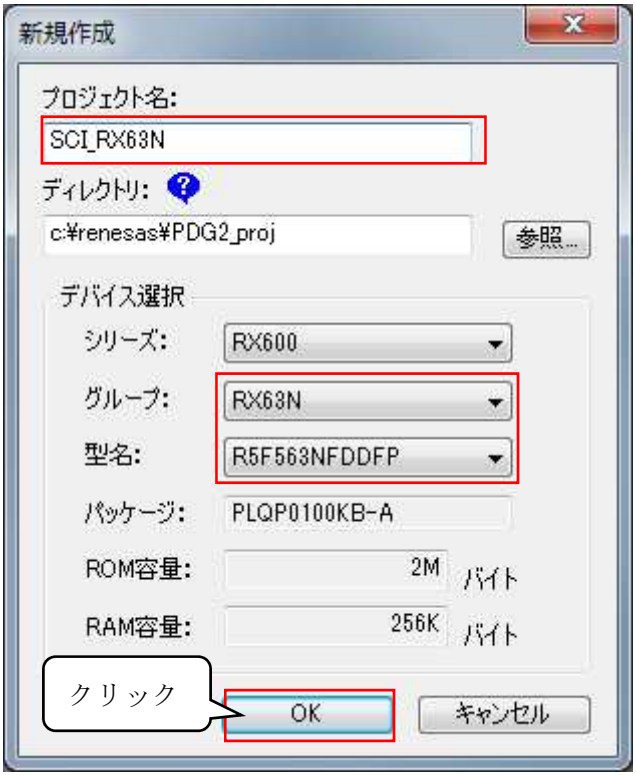


5. PDG の設定

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”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。

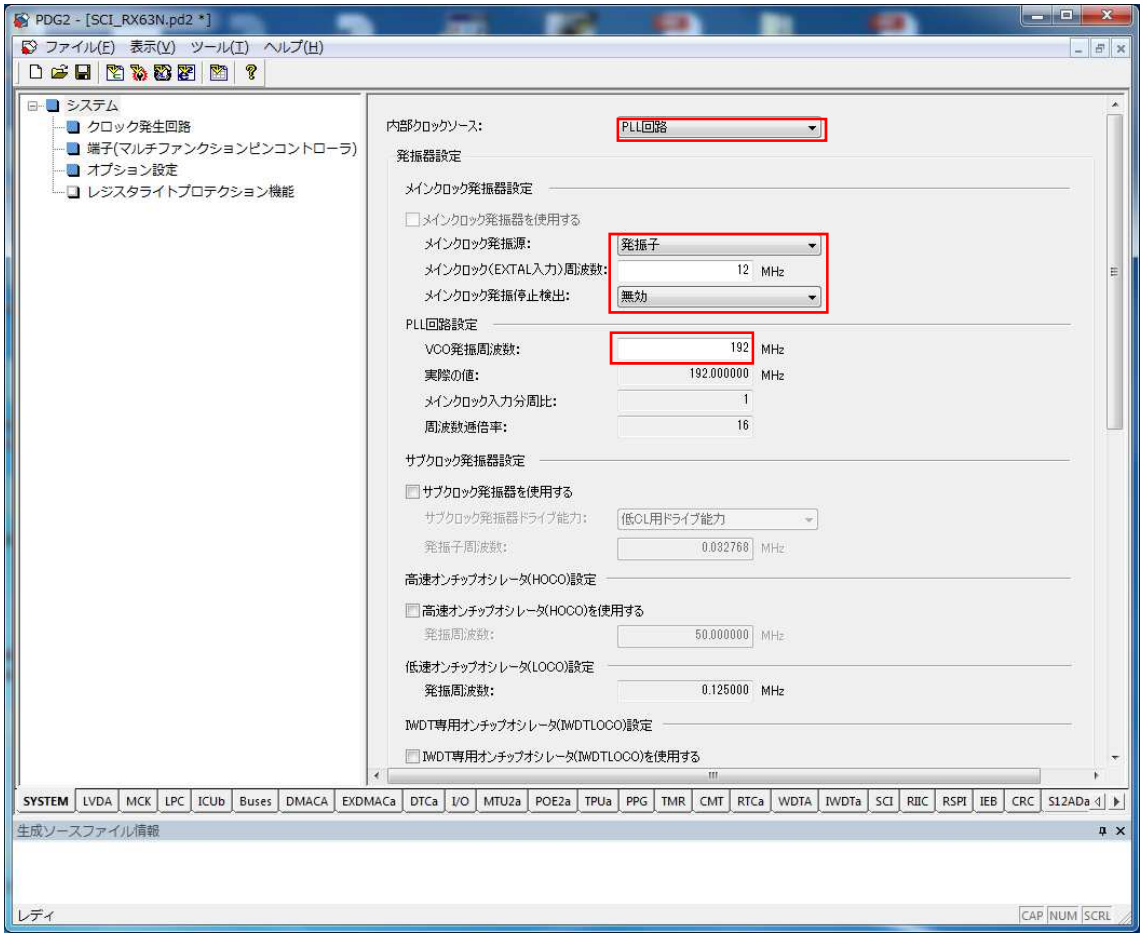
Peripheral Driver Generator 2 を起動します。

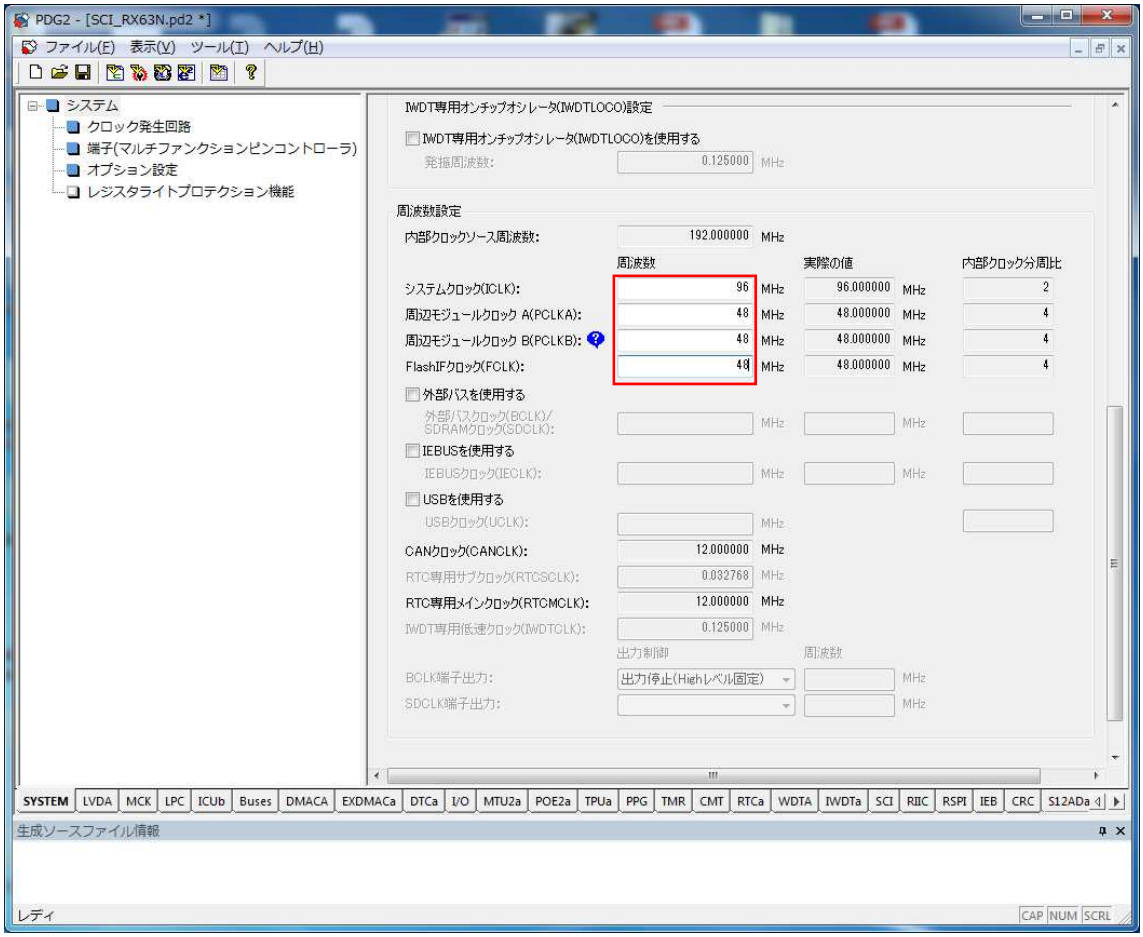
メニューバーのファイル->プロジェクトの新規作成 をクリックすると、以下のウィンドウが表示されます。プロジェクト名、マイコンのグループ、型を入力し、「OK」をクリックすると、プロジェクトが作成されます。



5.1　　SYSTEM 設定

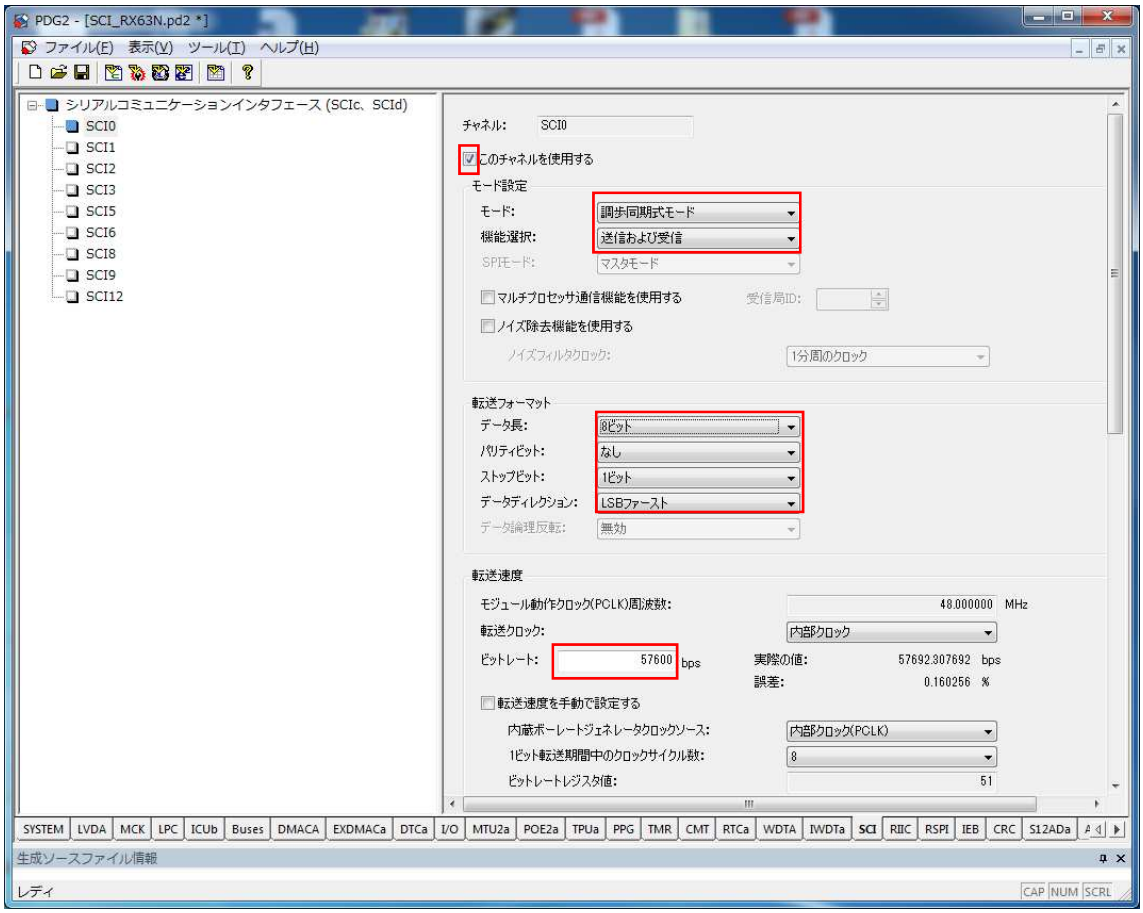
システムタブのクロック発生回路の設定を以下に示します。

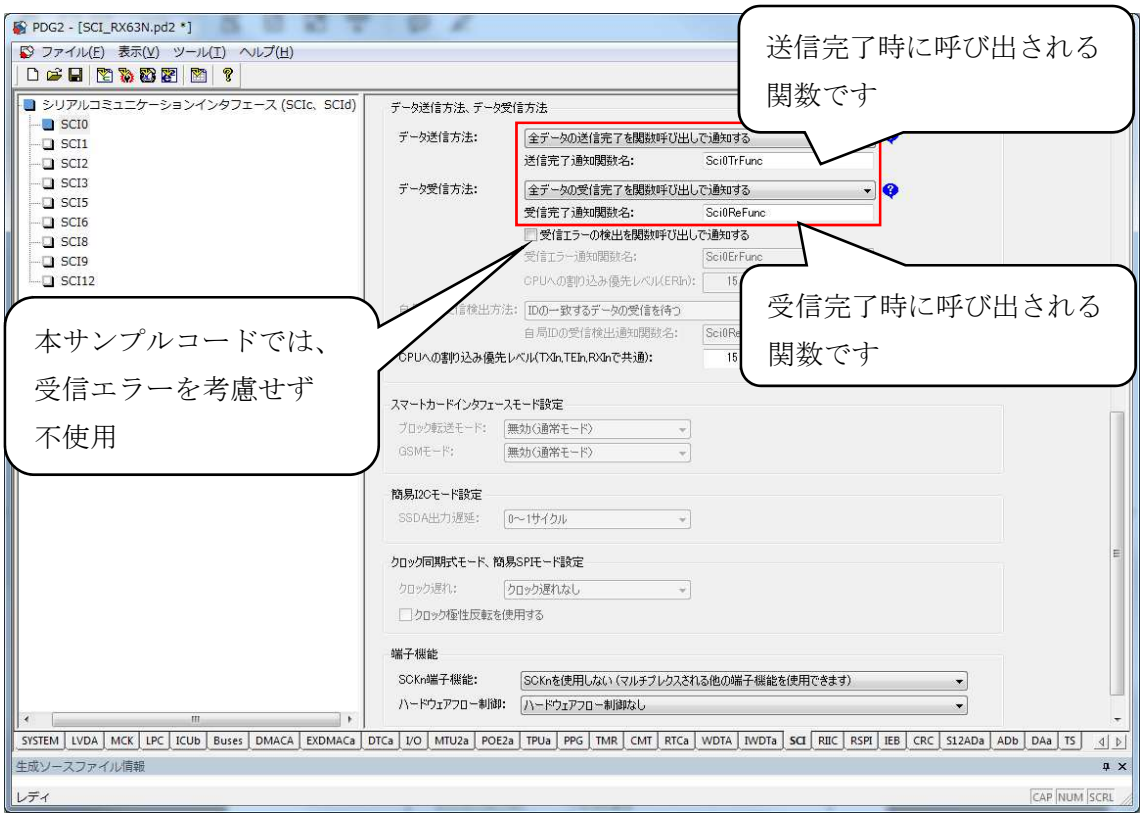



PDG2 - [SCI_RX63N.pd2 *]
ファイル(F) 表示(V) ツール(T) ヘルプ(H)
システム
クロック発生回路
端子(マルチファンクションピンコントローラ)
オプション設定
レジスタライトプロテクション機能
IWDT専用オンチップオシレータ(IWDTLOCO)設定
IWDT専用オンチップオシレータ(IWDTLOCO)を使用する
発振周波数:
0.125000
MHz
周波数設定
内部クロックソース周波数:
192.000000
MHz
周波数
実際の値
内部クロック分周比
システムクロック(ICLK):
96
96.000000
2
周辺モジュールクロック A(PCLKA):
48
48.000000
4
周辺モジュールクロック B(PCLKB):
48
48.000000
4
FlashIFクロック(FCLK):
48
48.000000
4
外部バスを使用する
外部バスクロック(BCLK)/
SDRAMクロック(SDCLK):
IEBUSを使用する
IEBUSクロック(IECLK):
USBを使用する
USBクロック(UCLK):
CANクロック(CANCLK):
12.000000
RTC専用サブクロック(RTCSCLK):
0.032768
RTC専用メインクロック(RTCMCLK):
12.000000
IWDT専用低速クロック(IWDTCLK):
0.125000
出力制御
周波数
BCLK端子出力:
出力停止(Highレベル固定)
SDCLK端子出力:
SYSTEM LVDA MCK LPC ICUb Buses DMACA EXDMACa DTCa I/O MTU2a POE2a TPUa PPG TMR CMT RTCa WDTA IWDTa SCI RIIC RSPI IEB CRC S12ADa
生成ソースファイル情報
レディ
CAP NUM SCRL

5.2　　SCI0 設定

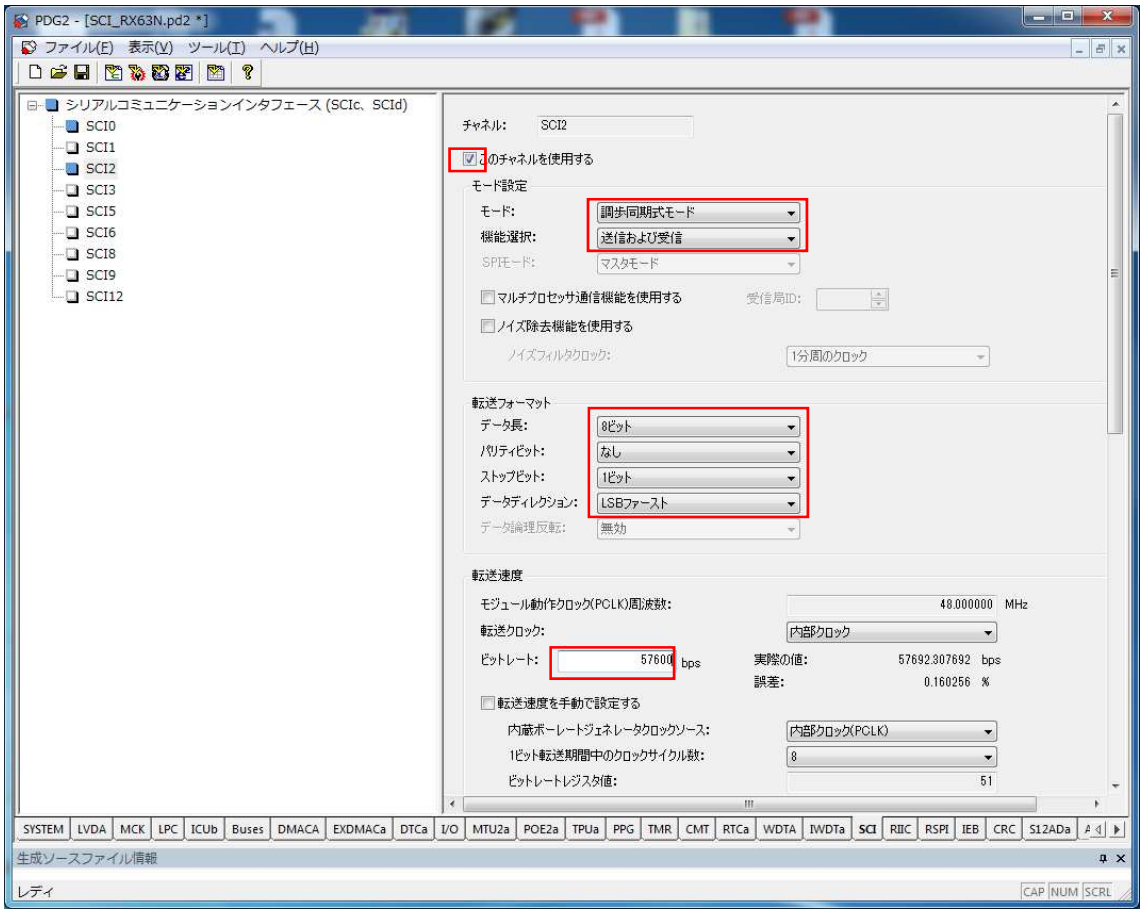
SCI0 の設定を以下に示します。

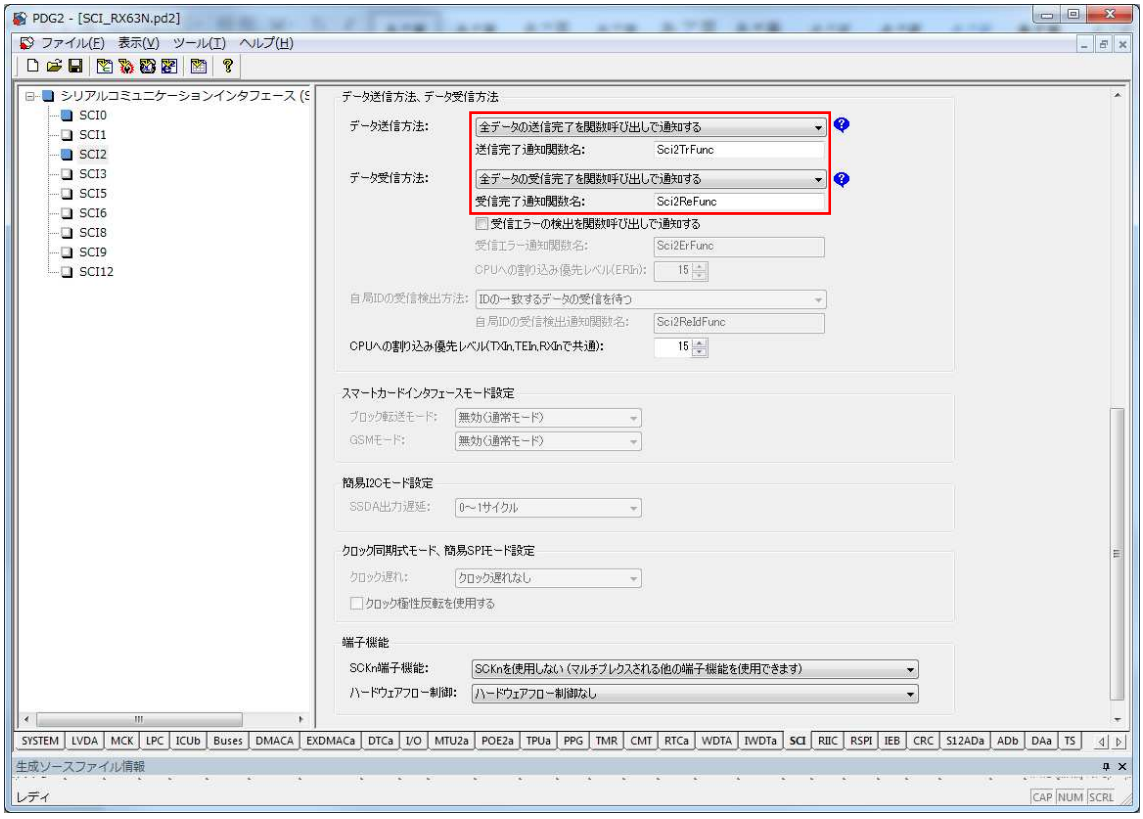



送信完了時に呼び出される関数です
受信完了時に呼び出される関数です
本サンプルコードでは、受信エラーを考慮せず不使用
PDG2 - [SCI_RX63N.pd2 *]
ファイル(F) 表示(V) ツール(T) ヘルプ(H)
シリアルコミュニケーションインタフェース (SCIc、SCId)
SCI0
SCI1
SCI2
SCI3
SCI5
SCI6
SCI8
SCI9
SCI12
データ送信方法、データ受信方法
データ送信方法:
全データの送信完了を関数呼び出しで通知する
送信完了通知関数名:
Sci0TrFunc
データ受信方法:
全データの受信完了を関数呼び出しで通知する
受信完了通知関数名:
Sci0ReFunc
受信エラーの検出を関数呼び出しで通知する
受信エラー通知関数名:
Sci0ErFunc
CPUへの割り込み優先レベル(ERIn):
IDの一致するデータの受信を待つ
自局IDの受信検出通知関数名:
CPUへの割り込み優先レベル(TXIn,TEIn,RXInで共通):
スマートカードインタフェースモード設定
ブロック転送モード:
無効(通常モード)
GSMモード:
無効(通常モード)
簡易I2Cモード設定
SSDA出力遅延:
0～1サイクル
クロック同期式モード、簡易SPIモード設定
クロック遅れ:
クロック遅れなし
クロック極性反転を使用する
端子機能
SCKn端子機能:
SCKnを使用しない (マルチプレクスされる他の端子機能を使用できます)
ハードウェアフロー制御:
ハードウェアフロー制御なし
SYSTEM LVDA MCK LPC ICUb Buses DMACA EXDMACa DTCa I/O MTU2a POE2a TPUa PPG TMR CMT RTCa WDTA IWDTa SCI RIIC RSPI IEB CRC S12ADa ADb DAa TS
生成ソースファイル情報
レディ
CAP NUM SCRL

5.3　　SCI2 設定

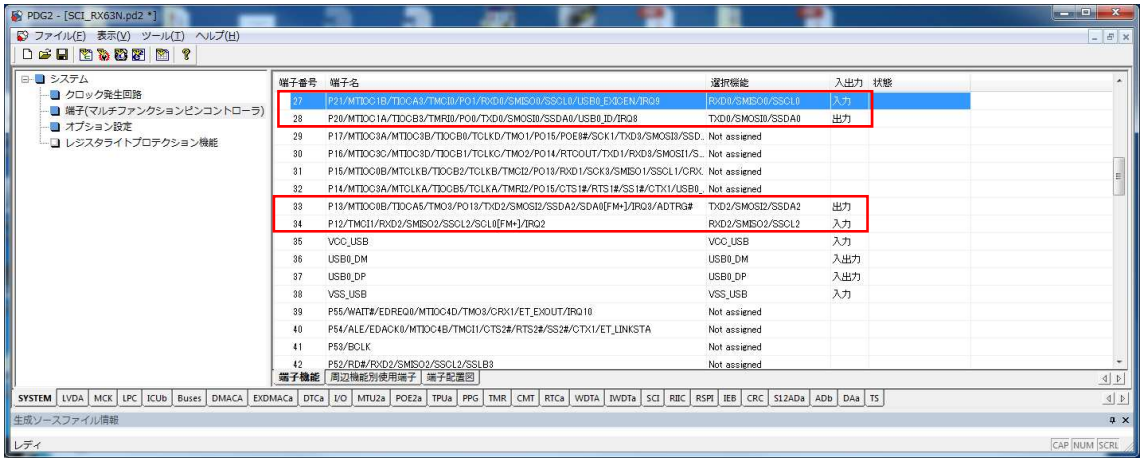
SCI2 の設定を以下に示します。(SCI0 と同様です)

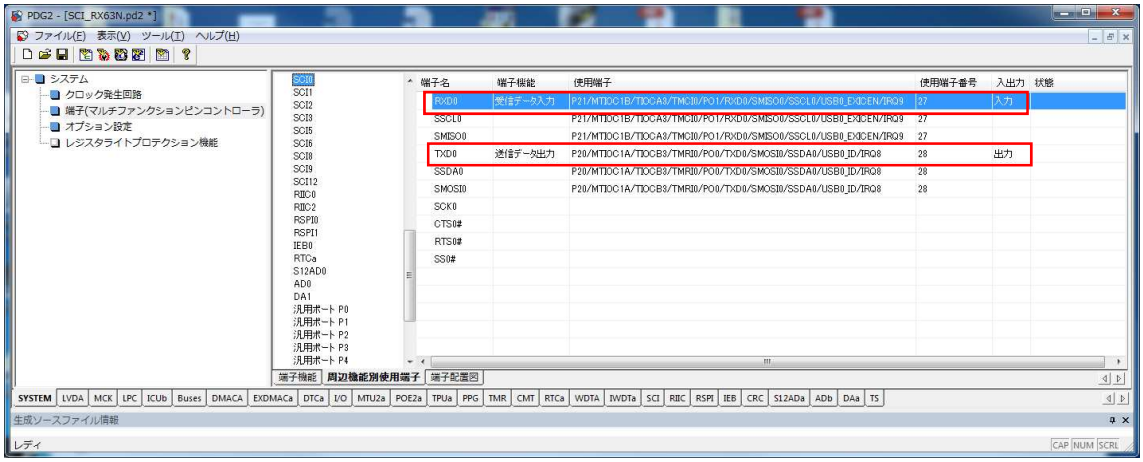


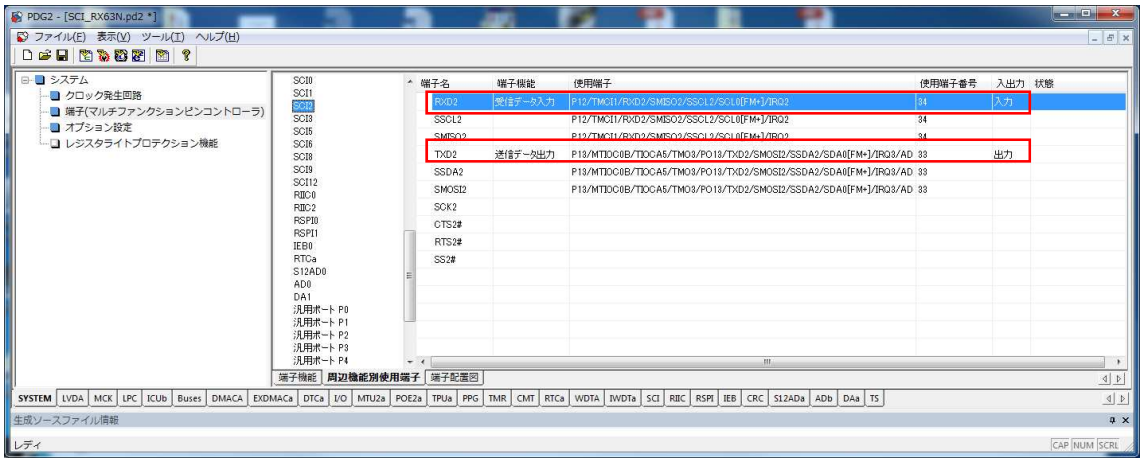

PDG2 - [SCI_RX63N.pd2]
ファイル(F) 表示(V) ツール(T) ヘルプ(H)
シリアルコミュニケーションインタフェース (S
SCI0
SCI1
SCI2
SCI3
SCI5
SCI6
SCI8
SCI9
SCI12
データ送信方法、データ受信方法
データ送信方法:
全データの送信完了を関数呼び出しで通知する
送信完了通知関数名:
Sci2TrFunc
データ受信方法:
全データの受信完了を関数呼び出しで通知する
受信完了通知関数名:
Sci2ReFunc
受信エラーの検出を関数呼び出しで通知する
受信エラー通知関数名:
Sci2ErFunc
CPUへの割り込み優先レベル(ERIn):
15
自局IDの受信検出方法:
IDの一致するデータの受信を待つ
自局IDの受信検出通知関数名:
Sci2ReIdFunc
CPUへの割り込み優先レベル(TXIn,TEIn,RXInで共通):
15
スマートカードインタフェースモード設定
ブロック転送モード:
無効(通常モード)
GSMモード:
無効(通常モード)
簡易I2Cモード設定
SSDA出力遅延:
0～1サイクル
クロック同期式モード、簡易SPIモード設定
クロック遅れ:
クロック遅れなし
クロック極性反転を使用する
端子機能
SCKn端子機能:
SCKnを使用しない（マルチプレクスされる他の端子機能を使用できます）
ハードウェアフロー制御:
ハードウェアフロー制御なし
SYSTEM LVDA MCK LPC ICUb Buses DMACA EXDMACa DTCa I/O MTU2a POE2a TPUa PPG TMR CMT RTCa WDTA IWDTa SCI RIIC RSPI IEB CRC S12ADa ADb DAa TS
生成ソースファイル情報
レディ
CAP NUM SCRL

5.4　　SYSTEM の端子設定

SYSTEM の端子設定を確認します。

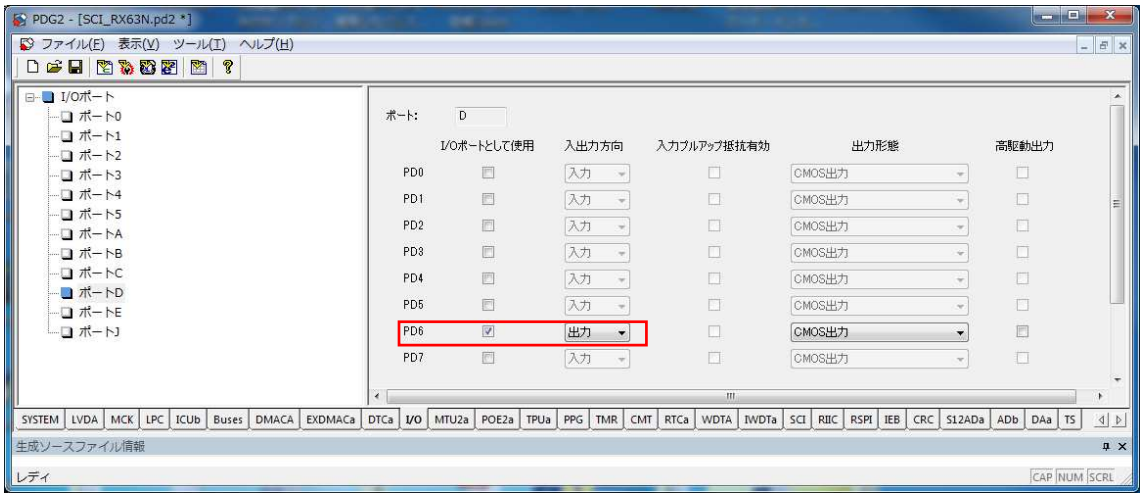






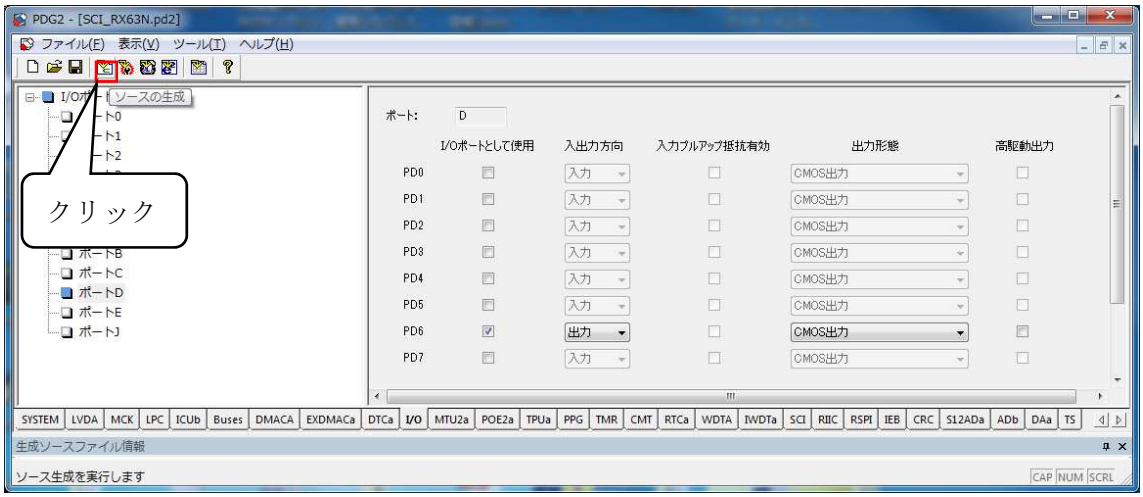
5.5　　I/O 設定

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

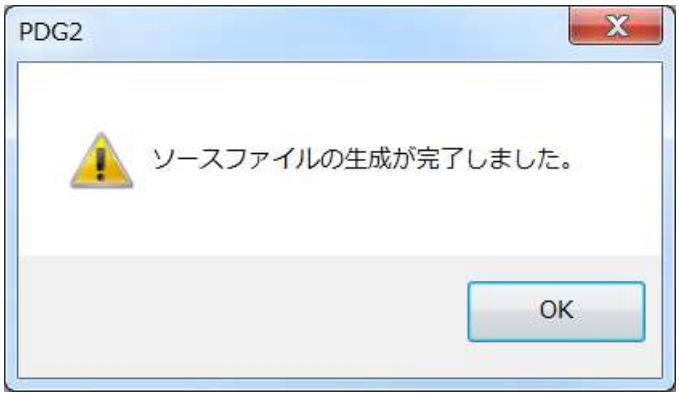


5.6　　ソースの生成

以下の GUI をクリックすると、

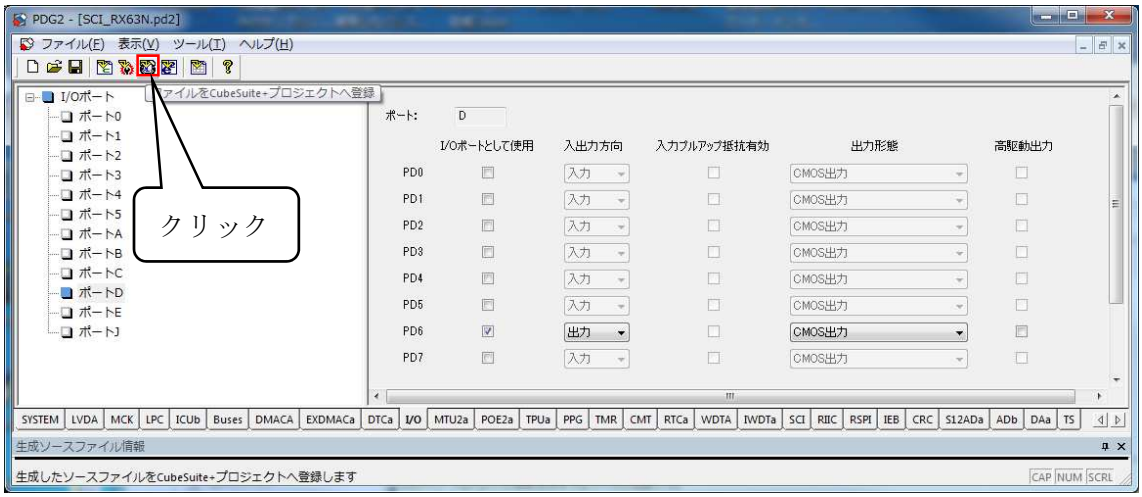


ソースファイルが生成されます。

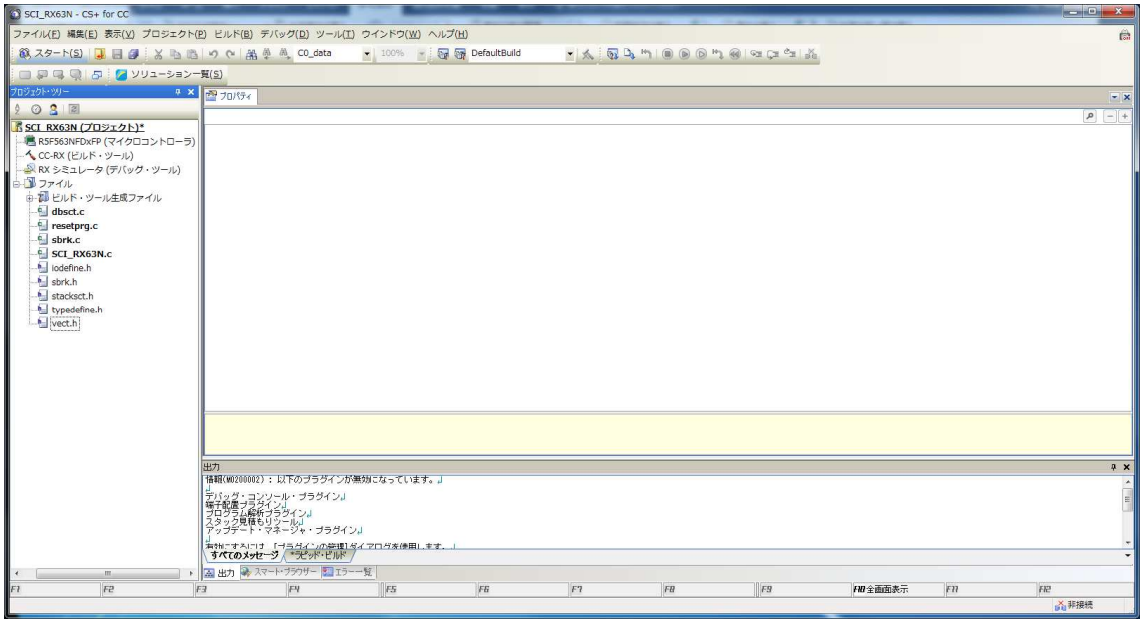


5.7　　CS+への登録

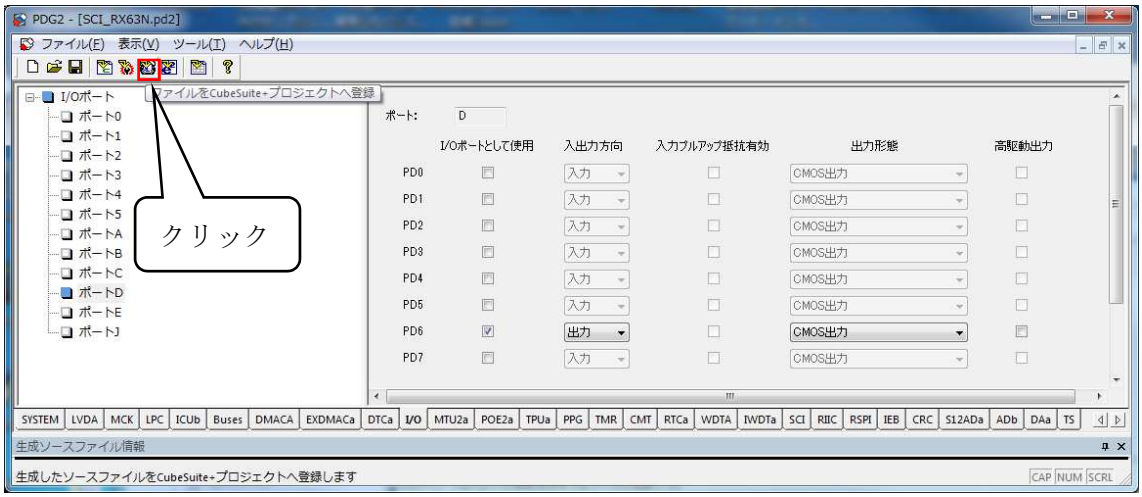
以下の GUI をクリックします。



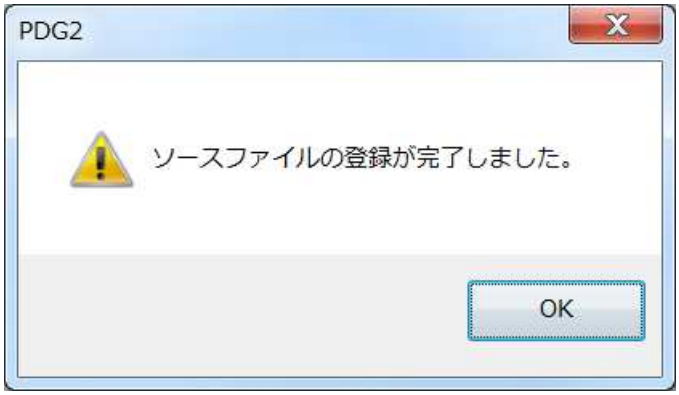
対象の CS+プロジェクトを開きます。



以下の GUI をクリックします

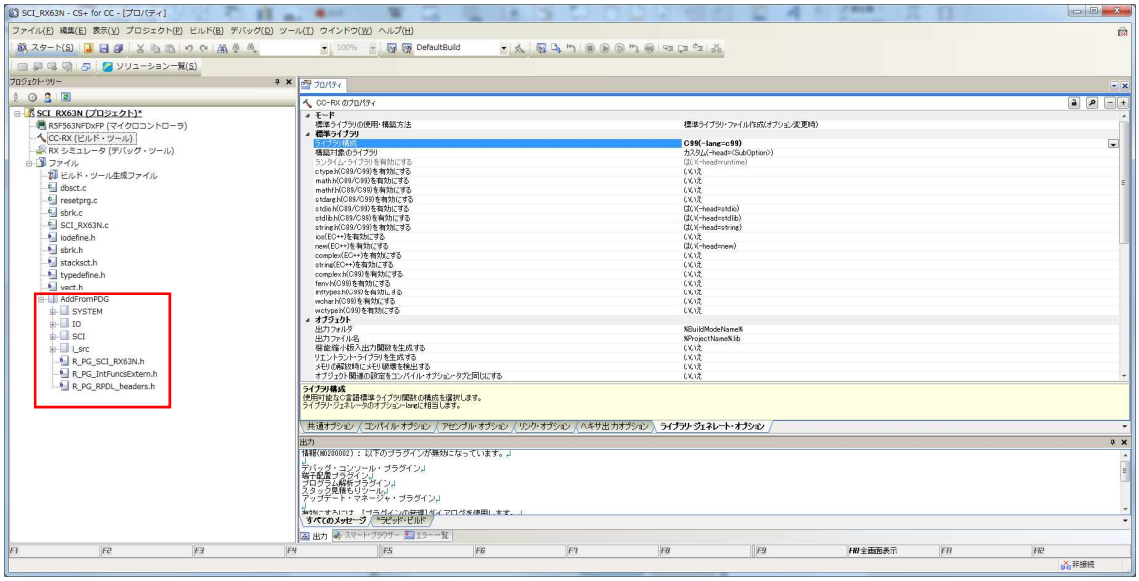


ソースファイルの登録が完了しました。



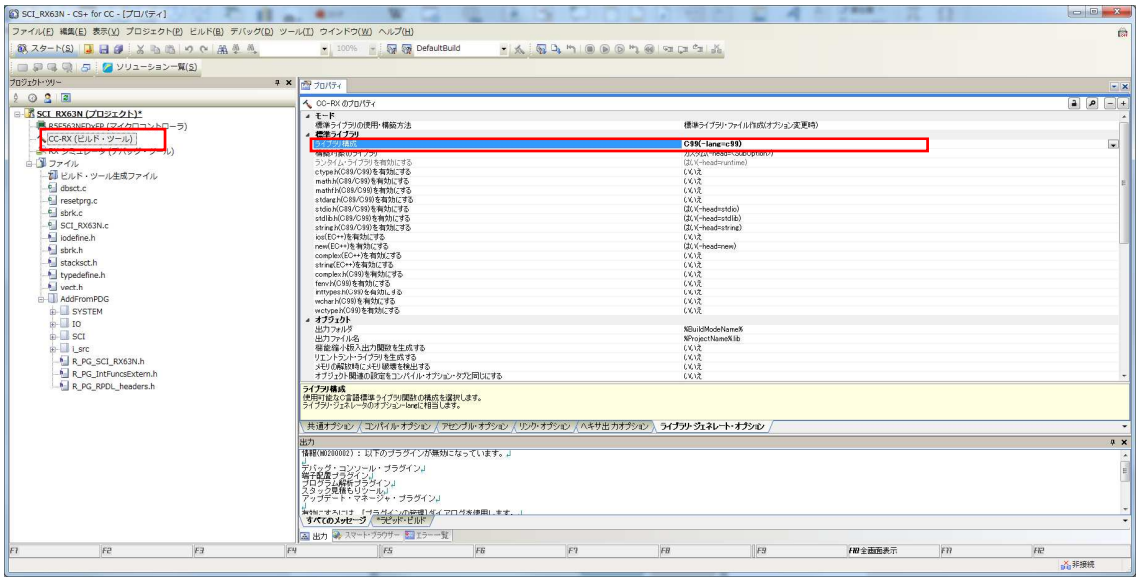
6.　CS+のプロジェクトに PDG のソースファイルを登録する際の設定

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。

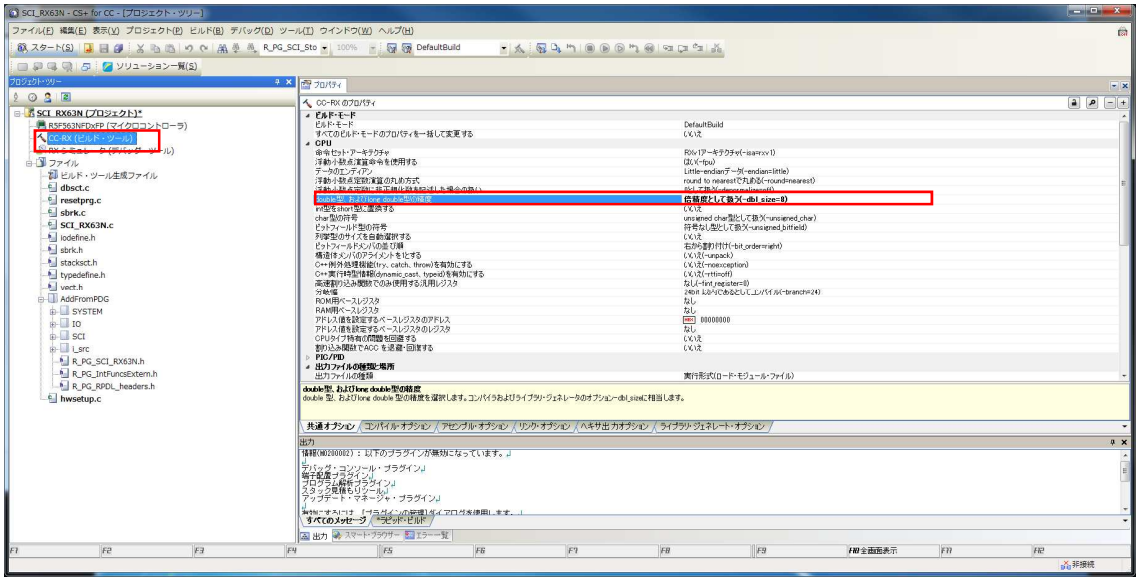


そのままビルドをすると、エラーおよび警告が発生します。解消する設定を以下に示します。

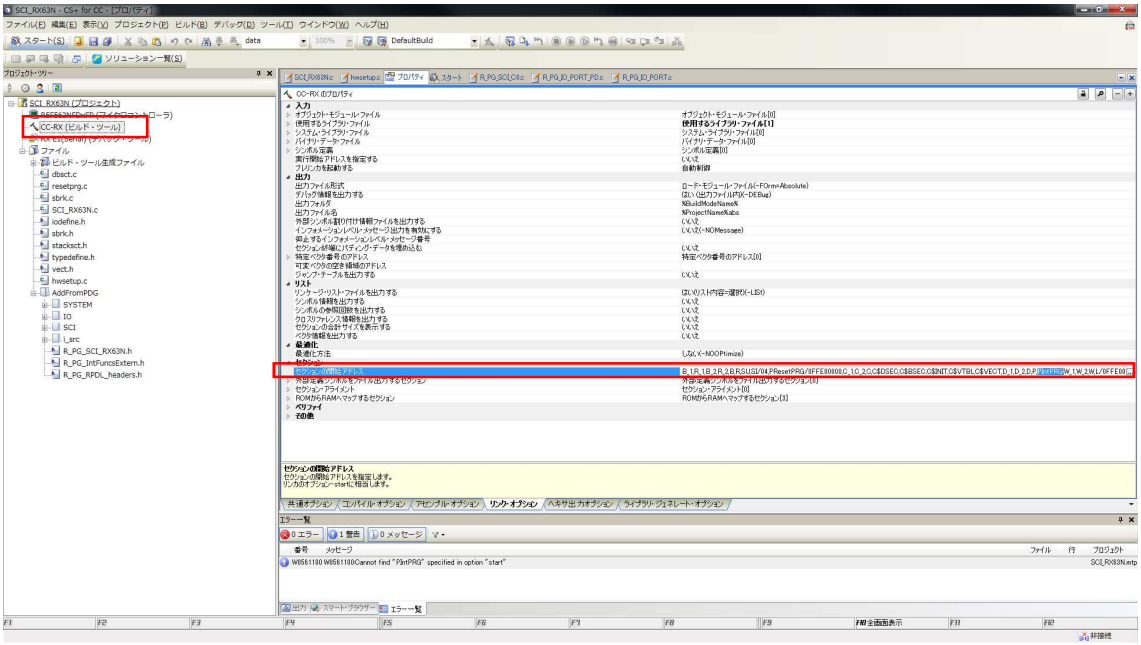
PDG で生成されるソースファイルは bool 変数を使用しています。対応させるため、ビルド・ツールを右クリック->プロパティを表示し、ライブラリ・ジェネレート・オプションタブにある「ライブラリ構成」を”C99(-lang=c99)”に設定します。



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。

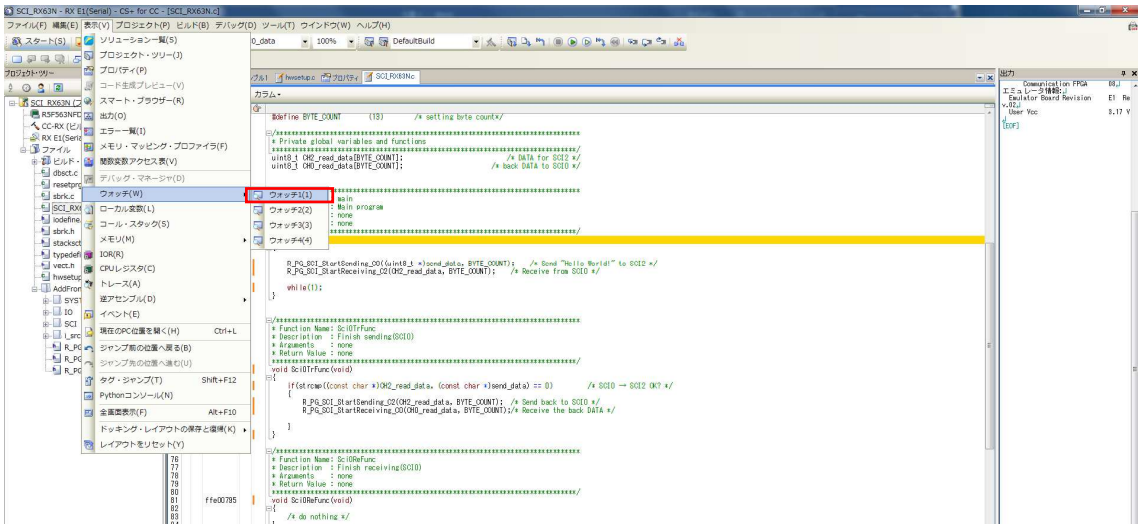


7. 動作確認方法

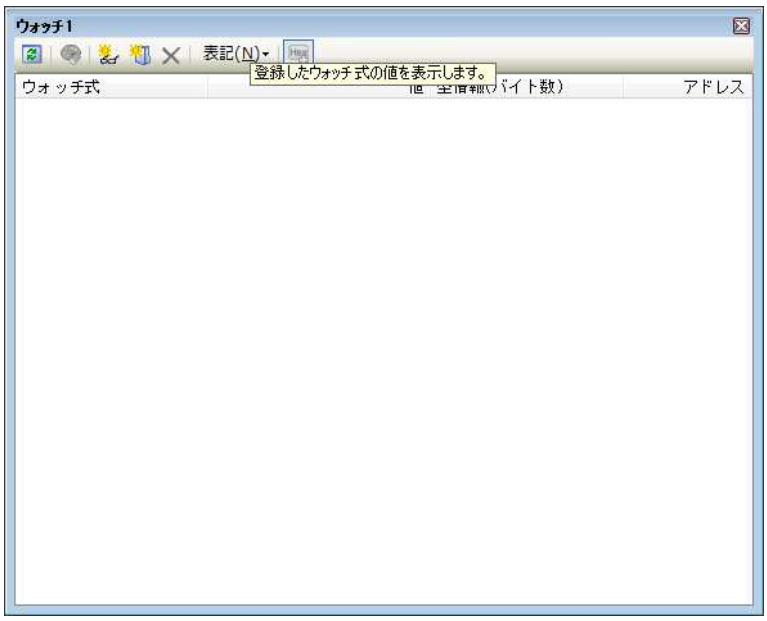
7.1 ウォッチ式の登録

SCI0 から SCI2 にデータが送信され、SCI2 の受信データを 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。

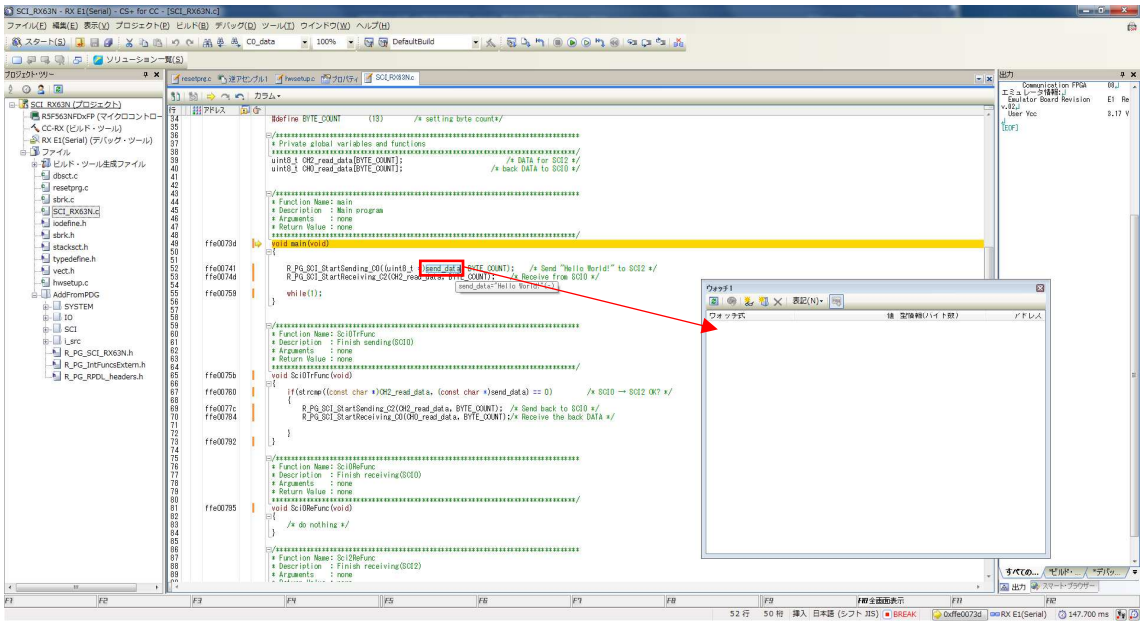
表示タブからウォッチ、そしてウォッチ 1 を選択します。



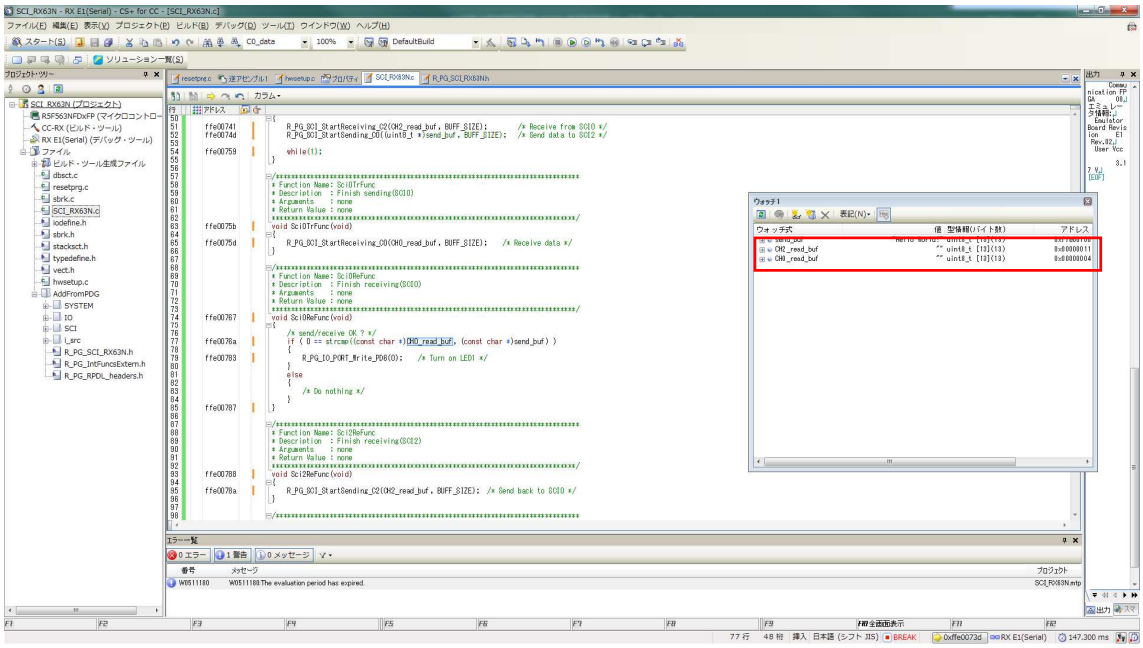
ウォッチ(ウォッチ 1)パネルが表示されました。



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”も同様にドラッグ&ドロップします。



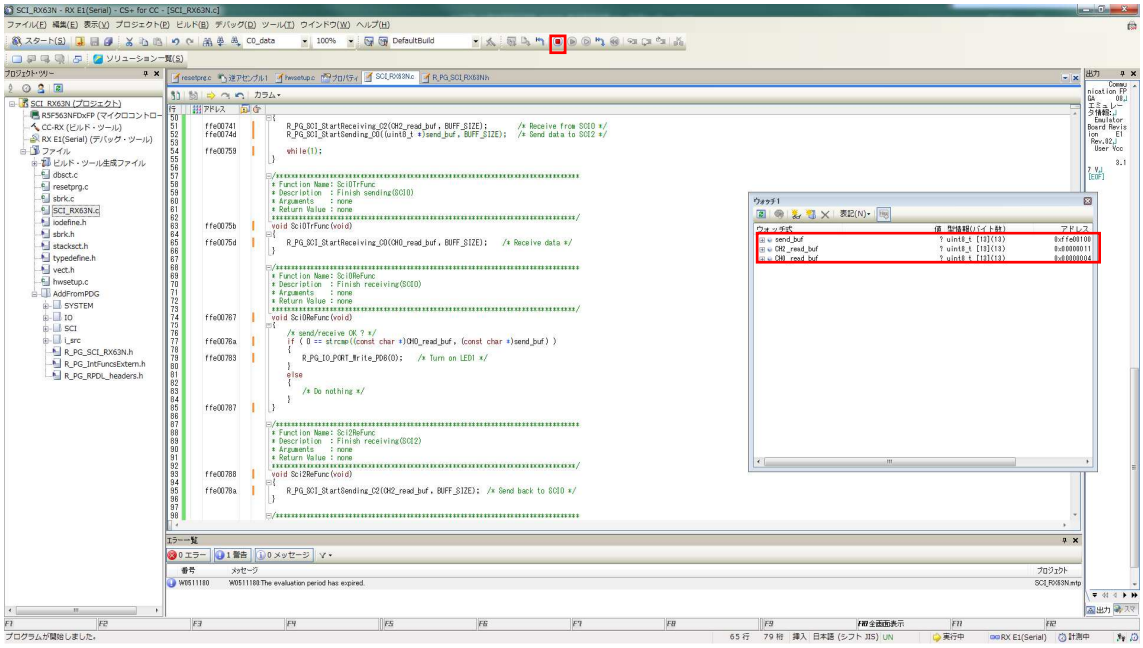
上の図のようにウォッチパネルには“send_buf”にのみ“Hello World!”が表示され、“CH2_read_buf”と“CH0_read_buf”は空欄であることが確認できます。ボード側は、LED1 は消灯となっています。この状態で実行前の準備は完了です。

7.2　　実行

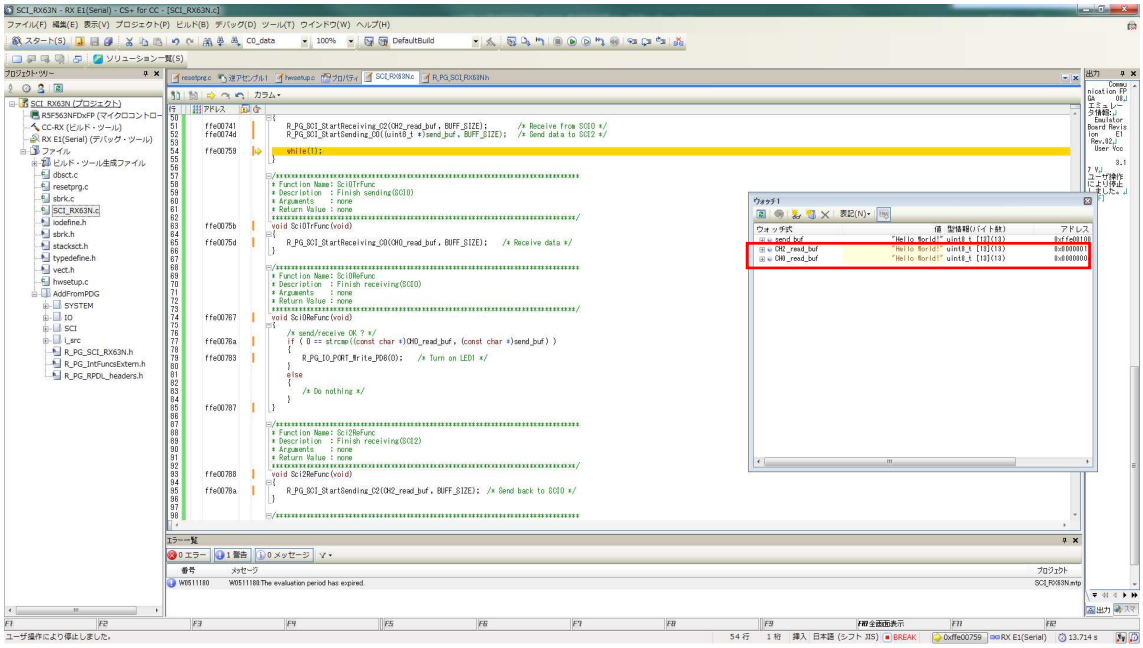
実行ボタンをクリックします。

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!”が受信できたことが確認できました。



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