RE01 1500KB グループ、256KB グループ USART クロック同期通信サンプルコード(using CMSIS Driver Package)

CMSIS Driver Package USART サンプルコード

要旨

本アプリケーションノートでは RE01 1500KB グループ、および RE01 256KB グループ CMSIS Driver Package を使用したサンプルコードについて説明致します。サンプルコードは同梱されたプロジェクトをご参照ください。

下記に本サンプルコードの概要を示します。

表　サンプルコードの概要

サンプルコードの動作概要	主となる周辺機能使用する	主として使用するドライバ
USART ドライバを使用し、クロック同期通信を行います。	USART を使用	R_USART

対象デバイス

RE01 1500KB グループ

RE01 256KB グループ

ご注意

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

関連ドキュメント

RE01 1500KB、256KB グループ CMSIS Package を用いた開発スタートアップガイド(R01AN4660)

目次

1. 仕様

1.1 プロジェクト説明

本アプリケーションノートには以下のサンプルコードプロジェクトが同梱されています。

RE01 1500KB グループ用サンプルコードプロジェクト：an4700_cmsis_usart_sci_re

RE01 256KB グループ用サンプルコードプロジェクト：r01an4700_cmsis_usart_sci_re_256kb

an4700_cmsis_usart_sci_re は、Evaluation Kit RE01 1500KB 上で動作を確認したプロジェクトです。このプロジェクトの設定は Evaluation Kit RE01 1500KB に実装されている R7F0E015D2CFB に合わせています。

r01an4700_cmsis_usart_sci_re_256kb は、Evaluation Kit RE01 256KB 上で動作を確認したプロジェクトです。このプロジェクトの設定は Evaluation Kit RE01 256KB に実装されている R7F0E01182CFP に合わせています。

その他のデバイスの場合は、プロジェクトの設定でデバイスを変更してご使用ください。

1.2 使用端子

表 1-1、表 1-2 にサンプルコードが使用する端子を示します。

表 1-1 RE01 1500KB グループ用サンプルコードで使用する端子

使用端子	用途
P009	LED0
P809	TXD3
P808	RXD3
P810	SCK3
P501	TXD2
P502	RXD2
P503	SCK2

表 1-2 RE01 256KB グループ用サンプルコードで使用する端子

使用端子	用途
P210	LED0
P012	TXD3
P808	RXD3
P013	SCK3
P102	TXD2
P101	RXD2
P100	SCK2

1.3 フォルダ構成

サンプルコード、およびサンプルコードで使用しているドライバの、フォルダ構成を示します。

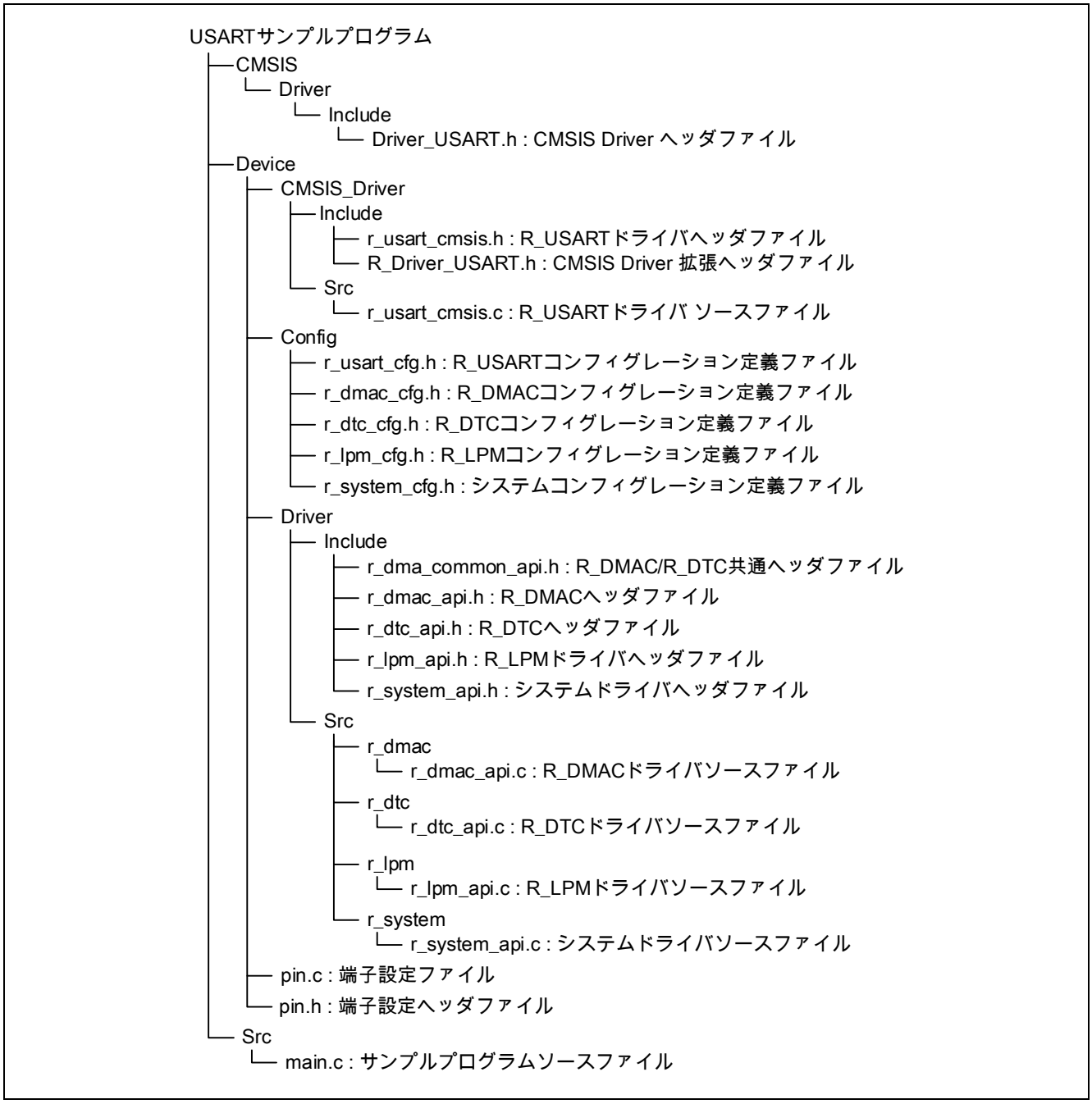


図 1.1 フォルダ構成

1.4 ファイル構成

表 1-3 にサンプルコードで追加・変更したファイルを示します。

表 1-3 サンプルコードで追加・変更したファイル

ファイル名	処理・設定概要	備考
main.c	メイン処理	
pin.c	I/O ポート設定	SCI3、SCI2 の割り当て端子の変更
r_system_cfg.h	システム設定	SCI3 送信、SCI2 受信割り込みの NVIC 登録

1.5 オプション設定メモリ

表 1-4 に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。必要に応じて、お客様のシステムに最適な値を設定してください。

表 1-4 サンプルコードで使用するオプション設定メモリ

シンボル	アドレス	設定値	内容
AWS	0100A164h～0100A167h	FFFF FFFFh	アクセスウィンドウ設定無し
OSIS	0100A150h～0100A15Fh	FFFF FFFFh	ID コードプロテクト無し（ALL FFh）
SECMPUxxx	00000408h~0000043Bh	FFFF FFFFh	MPU 無効
OFS1	00000404h~00000407h	FFFF FFFFh	リセット後、電圧監視 0 リセット無効 リセット後、HOCO 発振無効
OFS0	00000400h~00000403h	FFFF FFFFh	IWDT 自動起動無効 WDT 自動起動無効

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記(表 2-1、表 2-2)の条件で動作を確認しています。

表 2-1 RE01 1500KB グループ用サンプルコード動作確認条件

項目		内容
使用マイコン		R7F0E015D2CFB 144pin
動作周波数	システムクロックに PLL を選択	• メインクロック: 32MHz • PLL: 64MHz (メインクロック 4 分周 8 逓倍) • システムクロック(ICLK): 64MHz (PLL) • 周辺モジュールクロック A (PCLKA): 64MHz (PLL 分周なし) • 周辺モジュールクロック B (PCLKB): 32MHz (PLL 2 分周)
動作電圧		• 3.3V
統合開発環境	IAR	IAR Embedded Workbench for ARM Version 8.32.1 C コンパイラ：IAR C/C++ Compiler for ARM Version 8.32.1
	e² studio	Renesas e² studio Version 7 C コンパイラ：GCC ARM Embedded Version 6.3.1.20170620 GNU 6-2017-q2-update
デバッガ		Segger J-Link OB
ターゲットボード		Evaluation Kit RE01 1500KB （型名：RTK70E015DSXXXXXBE）
CMSIS Driver Package のバージョン		Rev1.00
サンプルコードのバージョン		Rev1.00

表 2-2 RE01 256KB グループ用サンプルコード動作確認条件

項目		内容
使用マイコン		R7F0E01182CFP 100pin
動作周波数	システムクロックに HOCO を選択	• HOCO: 64MHz • システムクロック(ICLK): 64MHz (HOCO) • 周辺モジュールクロック A (PCLKA): 64MHz (HOCO 分周なし) • 周辺モジュールクロック B (PCLKB): 32MHz (HOCO 2 分周)
動作電圧		• 3.3V
統合開発環境	IAR	IAR Embedded Workbench for ARM Version 8.40.2 C コンパイラ：IAR C/C++ Compiler for ARM Version 8.40.2
	e² studio	Renesas e² studio 2020-07 C コンパイラ：GCC ARM Embedded Version 6.3.1.20170620 GNU 6-2017-q2-update
デバッガ		Segger J-Link OB
ターゲットボード		Evaluation Kit RE01 256KB （型名：RTK70E0118CXXXXXBJ）
CMSIS Driver Package のバージョン		Rev1.00
サンプルコードのバージョン		Rev1.03

3. ソフトウェア説明

本サンプルコードは、R_USART ドライバを使用して、SCI3 をマスタ送信モード、SCI2 をスレーブ受信モードに設定します。

サンプルコードの動作を以下に示します。

- リセット解除後、SCI3、SCI2 の初期設定を行います。
- SCI2 のスレーブ受信、SCI3 のマスタ送信を開始します。
- SCI3 の送信完了割り込み、SCI2 の受信完了割り込みが発生すると、送信データと受信データを比較します。
- 送信データと受信データが一致すれば、LED0 を点灯します。その後、SCI2 の受信、SCI3 の送信を再度開始します。

表 3-1 サンプルプログラムの動作情報(SCI3)

項目	設定値
マスタスレーブ	マスタモード
転送モード	クロック同期通信
転送クロック	100kbps
CTS 機能	無効

表 3-2 サンプルプログラムの動作情報(SCI2)

項目	設定値
マスタスレーブ	スレーブモード
転送モード	クロック同期通信
転送クロック	100kbps
CTS 機能	無効

3.1 システム構成図

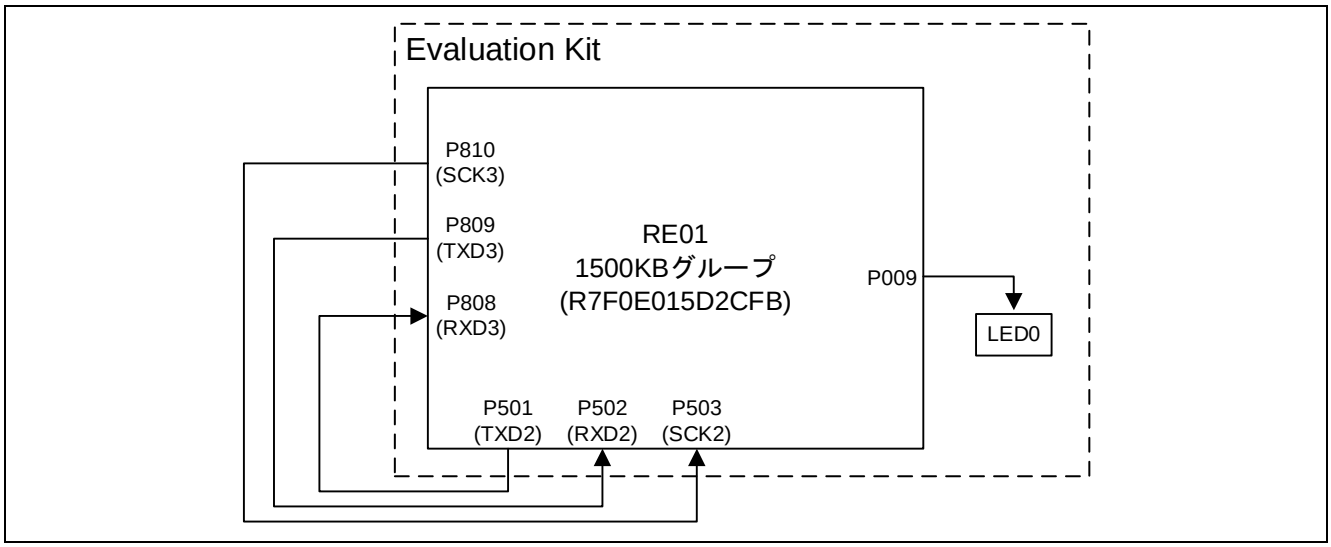


図 3.1　RE01 1500KB グループ用サンプルコード システム構成図

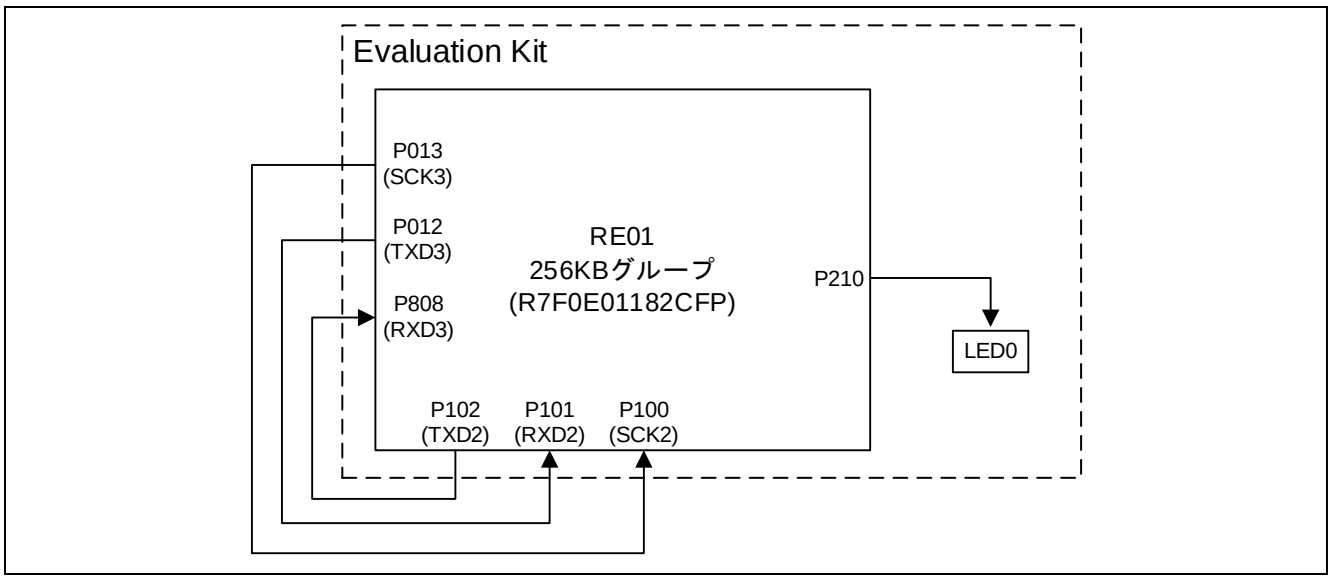


図 3.2　RE01 256KB グループ用サンプルコード システム構成図

3.2 ドライバ設定変更

3.2.1 RE01 1500KB グループ用サンプルコード ドライバ設定変更内容

表 3-3 RE01 1500KB グループサンプルコード ドライバ変更箇所(1/4)

項目	変更箇所	変更内容

CTS2 を未使用に変更	[pin.c]
	R_SCI_Pinset_CH2()関数
		● 以下の部分をコメントアウト
		// PFS->P103PFS_b.ASEL = 0U;
		// PFS->P103PFS_b.ISEL = 0U;
		// PFS->P103PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P103PFS_b.PMR = 1U;

TXD2 の設定をデフォルト(P102)から P501 に変更	[pin.c]
	R_SCI_Pinset_CH2()関数
		● 以下の部分をコメントアウト
		// PFS->P102PFS_b.ASEL = 0U;
		// PFS->P102PFS_b.ISEL = 0U;
		// PFS->P102PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P102PFS_b.PMR = 1U;

		● 以下の部分をコメントアウト解除 PFS->P501PFS_b.ASEL = 0U; PFS->P501PFS_b.ISEL = 0U; PFS->P501PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P501PFS_b.PMR = 1U;

RXD2 の設定をデフォルト(P101)から P502 に変更	[pin.c]
	R_SCI_Pinset_CH2()関数
		● 以下の部分をコメントアウト
		// PFS->P101PFS_b.ASEL = 0U;
		// PFS->P101PFS_b.ISEL = 0U;
		// PFS->P101PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P101PFS_b.PMR = 1U;

		● 以下の部分をコメントアウト解除 PFS->P502PFS_b.ASEL = 0U; PFS->P502PFS_b.ISEL = 0U; PFS->P502PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P502PFS_b.PMR = 1U;

表 3-4　RE01 1500KB グループサンプルコード ドライバ変更箇所(2/4)

項目	変更箇所	変更内容

SCK2 の設定をデフォルト(P100)から P503 に変更	[pin.c]
	R_SCI_Pinset_CH2()関数
		● 以下の部分をコメントアウト
		// PFS->P100PFS_b.ASEL = 0U;
		// PFS->P100PFS_b.ISEL = 0U;
		// PFS->P100PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P100PFS_b.PMR = 1U;
		● 以下の部分をコメントアウト解除
		PFS->P503PFS_b.ASEL = 0U;
		PFS->P503PFS_b.ISEL = 0U;
		PFS->P503PFS_b.PSEL =
		R_PIN_PRV_SCI_PSEL_04;
		PFS->P503PFS_b.PMR = 1U;

CTS3 を未使用に変更	[pin.c]
	R_SCI_Pinset_CH3()関数
		● 以下の部分をコメントアウト
		// PFS->P207PFS_b.ASEL = 0U;
		// PFS->P207PFS_b.ISEL = 0U;
		// PFS->P207PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P207PFS_b.PMR = 1U;

TXD3 の設定をデフォルト(P413)から P に 809 変更	[pin.c]
	R_SCI_Pinset_CH3()関数
		● 以下の部分をコメントアウト
		// PFS->P413PFS_b.ASEL = 0U;
		// PFS->P413PFS_b.ISEL = 0U;
		// PFS->P413PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P413PFS_b.PMR = 1U;
		● 以下の部分をコメントアウト解除
		PFS->P809PFS_b.ASEL = 0U;
		PFS->P809PFS_b.ISEL = 0U;
		PFS->P809PFS_b.PSEL =
		R_PIN_PRV_SCI_PSEL_04;
		PFS->P809PFS_b.PMR = 1U;

表 3-5　RE01 1500KB グループサンプルコード ドライバ変更箇所(3/4)

項目	変更箇所	変更内容

RXD3 の設定をデフォルト(P412)から P808 に変更	[pin.c]/
	R_SCI_Pinset_CH3()関数
		● 以下の部分をコメントアウト
		// PFS->P412PFS_b.ASEL = 0U;
		// PFS->P412PFS_b.ISEL = 0U;
		// PFS->P412PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P412PFS_b.PMR = 1U;

		● 以下の部分をコメントアウト解除 PFS->P808PFS_b.ASEL = 0U; PFS->P808PFS_b.ISEL = 0U; PFS->P808PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P808PFS_b.PMR = 1U;

SCK3 の設定をデフォルト(P411)から P810 に変更	[pin.c]
	R_SCI_Pinset_CH3()関数
		● 以下の部分をコメントアウト
		// PFS->P411PFS_b.ASEL = 0U;
		// PFS->P411PFS_b.ISEL = 0U;
		// PFS->P411PFS_b.PSEL
		= R_PIN_PRV_SCI_PSEL_04;
		// PFS->P411PFS_b.PMR = 1U;

		● 以下の部分をコメントアウト解除 PFS->P810PFS_b.ASEL = 0U; PFS->P810PFS_b.ISEL = 0U; PFS->P810PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P810PFS_b.PMR = 1U;

表 3-6　RE01 1500KB グループサンプルコード ドライバ変更箇所(4/4)

項目	変更箇所	変更内容
NVIC への SCI3 受信データフル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI3_RXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER0)
NVIC への SCI3 送信データエンプティ割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI3_TXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER1)
NVIC への SCI3 受信エラー割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI3_ERI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER3)
NVIC への SCI2 受信データフル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI2_RXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER12)
NVIC への SCI2 送信データエンプティ割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI2_TXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER5)
NVIC への SCI2 受信エラー割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI2_ERI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER7)

3.2.2 RE01 256KB グループ用サンプルコード ドライバ設定変更内容

表 3-7 RE01 256KB グループサンプルコード ドライバ変更箇所(1/2)

項目	変更箇所	変更内容
TXD2 を P102 に設定	[pin.c] R_SCI_Pinset_CH2()関数	● 以下の部分をコメントアウト解除 PFS->P102PFS_b.PMR = 0U; PFS->P102PFS_b.ASEL = 0U; PFS->P102PFS_b.ISEL = 0U; PFS->P102PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P102PFS_b.PMR = 1U;
RXD2 を P101 に設定	[pin.c] R_SCI_Pinset_CH2()関数	● 以下の部分をコメントアウト解除 PFS->P101PFS_b.PMR = 0U; PFS->P101PFS_b.ASEL = 0U; PFS->P101PFS_b.ISEL = 0U; PFS->P101PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P101PFS_b.PMR = 1U;
CLK2 を P100 に設定	[pin.c] R_SCI_Pinset_CH2()関数	● 以下の部分をコメントアウト解除 PFS->P100PFS_b.PMR = 0U; PFS->P100PFS_b.ASEL = 0U; PFS->P100PFS_b.ISEL = 0U; PFS->P100PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P100PFS_b.PMR = 1U;
TXD3 を P012 に設定	[pin.c] R_SCI_Pinset_CH3()関数	● 以下の部分をコメントアウト解除 PFS->P012PFS_b.PMR = 0U; PFS->P012PFS_b.ASEL = 0U; PFS->P012PFS_b.ISEL = 0U; PFS->P012PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04; PFS->P012PFS_b.PMR = 1U;
RXD3 を P808 に設定	[pin.c] R_SCI_Pinset_CH3()関数	● 以下の部分をコメントアウト解除 PFS->P808PFS_b.PMR = 0U; PFS->P808PFS_b.ASEL = 0U; PFS->P808PFS_b.ISEL = 0U; PFS->P808PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_05; PFS->P808PFS_b.PMR = 1U;

表 3-8　RE01 256KB グループサンプルコード ドライバ変更箇所(2/2)

項目	変更箇所	変更内容
CLK3 を P013 に設定	[pin.c] R_SCI_Pinset_CH3()関数	● 以下の部分をコメントアウト解除 PFS->P013PFS_b.PMR = 0U; PFS->P013PFS_b.ASEL = 0U; PFS->P013PFS_b.ISEL = 0U; PFS->P013PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_05; PFS->P013PFS_b.PMR = 1U;
NVIC への SCI3 受信データフル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI3_RXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER0)
NVIC への SCI3 送信データエンプティ割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI3_TXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER1)
NVIC への SCI3 受信エラー割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI3_ERI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER3)
NVIC への SCI2 受信データフル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI2_RXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER12)
NVIC への SCI2 送信データエンプティ割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI2_TXI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER5)
NVIC への SCI2 受信エラー割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SCI2_ERI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER7)

3.3 関数一覧

サンプルコードで追加した関数について説明します。

main	
概　要	メイン処理
ヘッダ	なし
宣　言	void main(void)
説　明	システムの初期設定関数を呼び出します。その後、USART の動作設定を行い、クロック同期式通信を開始します。
引　数	なし
リターン値	なし

system_init	
概　要	システム初期処理
ヘッダ	なし
宣　言	static void system_init(void)
説　明	セクション初期化、システム初期化、R_LPM ドライバ初期化、および IO 電源供給設定関数を呼び出します
引　数	なし
リターン値	なし

sci3_callback		
概　要	SCI3 転送完了コールバック処理	
ヘッダ	なし	
宣　言	static void sci3_callback(uint32_t　event)	
説　明	SCI3 送信完了後、送信完了フラグを Set します。	
引　数	uint32_t　event	コールバック発生要因 ARM_USART_EVENT_SEND_COMPLETE　送信完了
リターン値	なし	

sci2_callback		
概　要	SCI2 転送完了コールバック処理	
ヘッダ	なし	
宣　言	static void sci2_callback(uint32_t　event)	
説　明	SCI2 受信完了後、受信完了フラグを Set します。	
引　数	uint32_t　event	コールバック発生要因 ARM_USART_EVENT_RECEIVE_COMPLETE　受信完了
リターン値	なし	

3.4 定数一覧

表 3-9 に定数一覧を示します。

表 3-9 サンプルコードで使用する定数(ユーザ変更可)

定数名	設定値	内容
SCI_TRANSFER_RATE	100000U	USART 転送クロック
SCI_DATA_SIZE	5	USART 転送データサイズ

3.5 フローチャート

図 3.3 にメイン処理のフローチャートを示します。

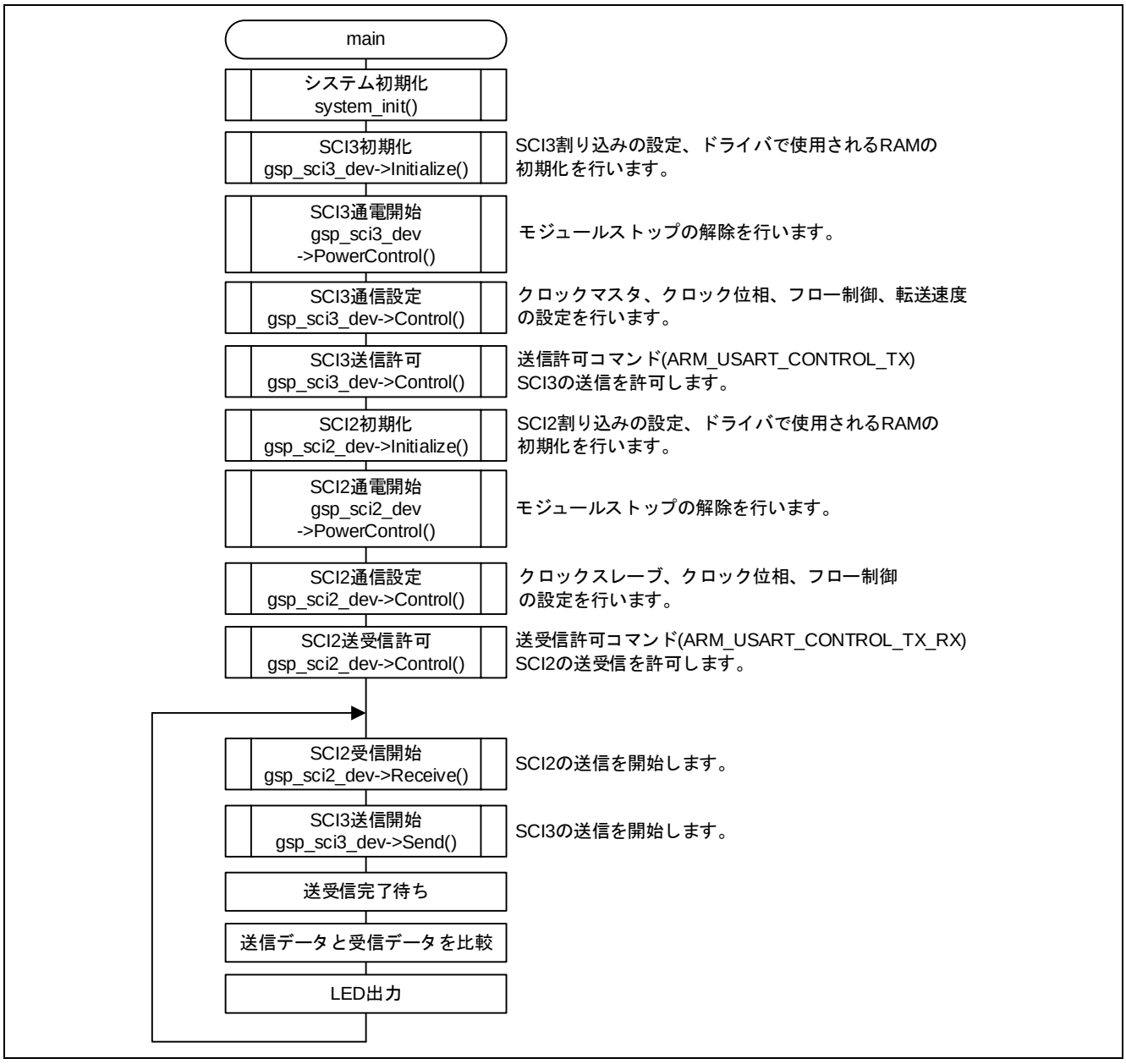


図 3.3 メイン処理

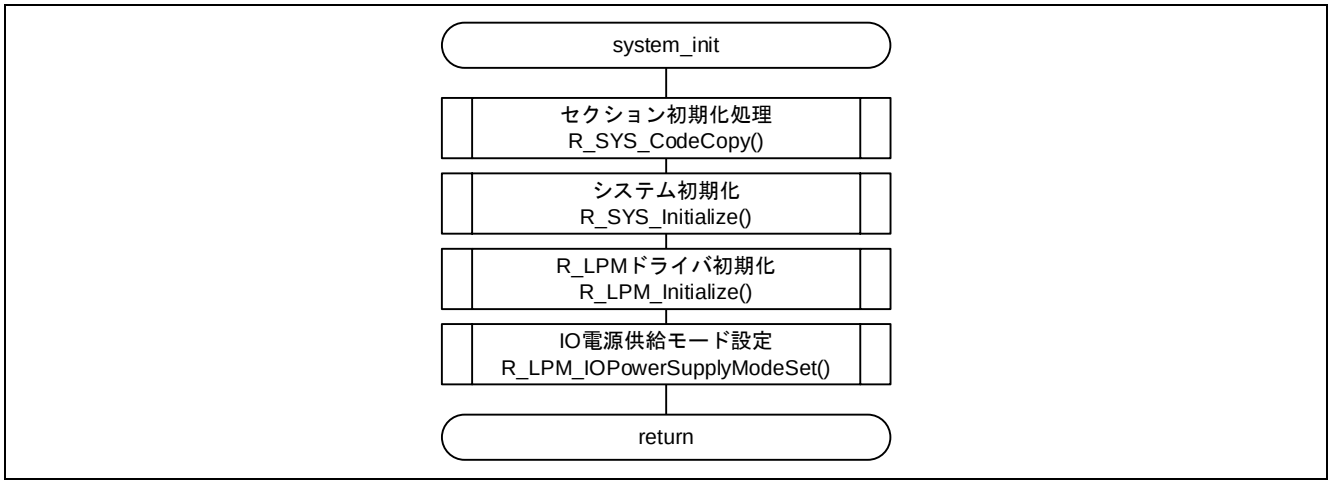


図 3.4　システム初期処理

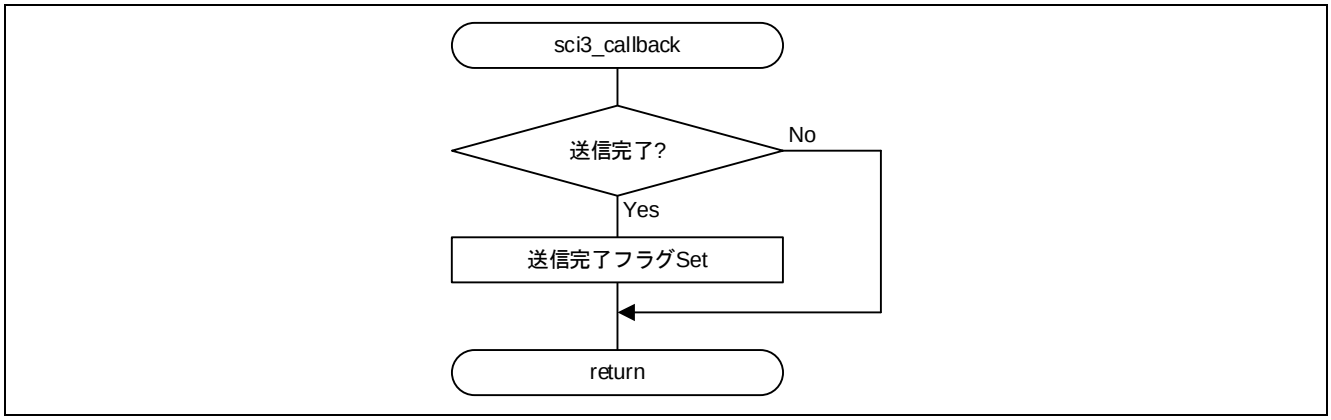


図 3.4　SCI3 転送完了コールバック処理

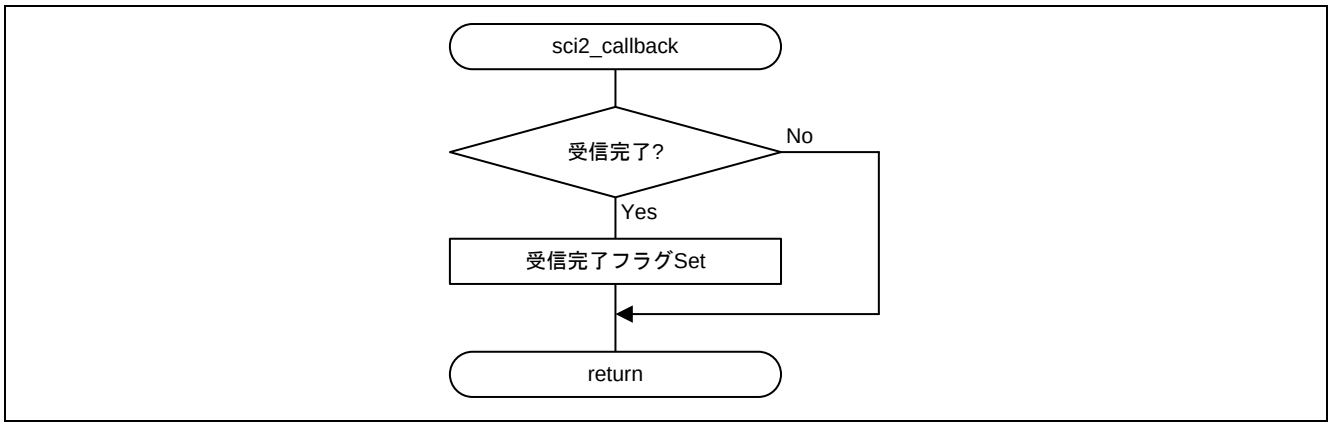


図 3.5　SCI2 転送完了コールバック処理

4. ドライバの API 仕様

4.1 外部仕様書

本ドライバには API の外部仕様を記したドキュメントを同包しています。

Documents フォルダの直下にある Driver Specification フォルダに格納されています。

5. R_USART ドライバを使用する上での注意事項

本章では R_USART ドライバに関する主だった注意点を紹介します。すべての注意点を紹介しきれていません。

注意点について『4.1 外部仕様書』をご参照ください。

5.1 DMAC/DTC を使用した USART 通信について

DMAC/DTC を使用して送信または受信を行う場合は、r_usart_cfg.h の送信/受信制御設定の設定値を変更してください。

送信/受信制御方法の設定定義を表 5-1 に、送信/受信制御方法の定義を表 5-2 に示します。

表 5-1 送信/受信制御方法の設定定義（n=0～5、9）

定義	初期値	内容
SCIn_TRANSMIT_CONTROL	SCI_USED_INTERRUPT	SCIn の送信制御（初期値：割り込み）
SCIn_RECEIVE_CONTROL	SCI_USED_INTERRUPT	SCIn の受信制御（初期値：割り込み）

表 5-2 送信/受信制御方法の定義

定義	値	内容
SCI_USED_INTERRUPT	(0)	送信/受信制御に割り込みを使用
SCI_USED_DMAC0	(1<<0)	送信/受信制御に DMAC0 を使用
SCI_USED_DMAC1	(1<<1)	送信/受信制御に DMAC1 を使用
SCI_USED_DMAC2	(1<<2)	送信/受信制御に DMAC2 を使用
SCI_USED_DMAC3	(1<<3)	送信/受信制御に DMAC3 を使用
SCI_USED_DTC	(1<<15)	送信/受信制御に DTC を使用

5.2 NVIC への割り込み登録について

USART を使用する場合は r_system_cfg.h にて NVIC への登録を行ってください。

使用用途に対する NVIC の登録定義を表 5-3 に、NVIC への割り込み登録例を図 5.1 示します。

表 5-4 使用用途に対する NVIC の登録定義

モード	使用用途	NVIC 登録定義	備考
調歩同期	送信のみで使用	SYSTEM_CFG_EVENT_NUMBER_SCIn_TXI	(注 1)
	受信のみで使用	SYSTEM_CFG_EVENT_NUMBER_SCIn_RXI	(注 2)
		SYSTEM_CFG_EVENT_NUMBER_SCIn_ERI	
	送受信で使用	SYSTEM_CFG_EVENT_NUMBER_SCIn_TXI	(注 1)
		SYSTEM_CFG_EVENT_NUMBER_SCIn_RXI	(注 2)
		SYSTEM_CFG_EVENT_NUMBER_SCIn_ERI	
クロック同期/スマートカード	送信のみで使用	SYSTEM_CFG_EVENT_NUMBER_SCIn_TXI	(注 1)
	送受信、または受信のみで使用	SYSTEM_CFG_EVENT_NUMBER_SCIn_TXI	(注 1)
		SYSTEM_CFG_EVENT_NUMBER_SCIn_RXI	(注 2)
		SYSTEM_CFG_EVENT_NUMBER_SCIn_ERI	

注1. 送信制御に DMACm(m=0～3)を使用する場合は、SYSTEM_CFG_EVENT_NUMBER_DMACm_INT を NVIC 登録してください。

注2. 受信制御に DMACm(m=0～3)を使用する場合は、SYSTEM_CFG_EVENT_NUMBER_DMACm_INT を NVIC 登録してください。

```
・・・

#define SYSTEM_CFG_EVENT_NUMBER_GPT_UVWEDGE
        (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)      /*!< Numbers 0/4/8/12/16/20/24/28 only */
#define SYSTEM_CFG_EVENT_NUMBER_SCI0_RXI
           (SYSTEM_IRQ_EVENT_NUMBER0)             /*!< Numbers 0/4/8/12/16/20/24/28 only */
#define SYSTEM_CFG_EVENT_NUMBER_SCI0_AM
            (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 0/4/8/12/16/20/24/28 only */
・・・

#define SYSTEM_CFG_EVENT_NUMBER_GPT2_CCMPB
         (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)      /*!< Numbers 1/5/9/13/17/21/25/29 only */
#define SYSTEM_CFG_EVENT_NUMBER_SCI0_TXI
           (SYSTEM_IRQ_EVENT_NUMBER1)             /*!< Numbers 1/5/9/13/17/21/25/29 only */
#define SYSTEM_CFG_EVENT_NUMBER_SPI0_SPTI
          (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)     /*!< Numbers 1/5/9/13/17/21/25/29 only */
・・・
#define SYSTEM_CFG_EVENT_NUMBER_GPT2_UDF
           (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)    /*!< Numbers 3/7/11/15/19/23/27/31 only */
#define SYSTEM_CFG_EVENT_NUMBER_SCI0_ERI
           (SYSTEM_IRQ_EVENT_NUMBER3)             /*!< Numbers 3/7/11/15/19/23/27/31 only */
#define SYSTEM_CFG_EVENT_NUMBER_SPI0_SPEI
          (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)     /*!< Numbers 3/7/11/15/19/23/27/31 only */
・・・
```

図 5.1 NVIC への割り込み登録例(SCI0 使用時)

5.3 クロック同期、スマートカード通信にて受信を使用する場合の設定について

クロック同期で送信・受信を同時に行う場合は、USART の Control 関数を用いて以下の手順で TE、RE の許可を行ってください。H/W の制限で TE、RE を同時に許可する必要があります。以下の手順に従わない場合、先に設定した側のみ設定されます。

受信のみを行う場合も、ダミーデータの書き込みを行うため同様の手順で行って下さい。

クロック同期にて受信を使用する場合の設定例を図 5.2 に示します。

```
#include " R_Driver_USART.h"

static void usart_callback(uint32_t event);

// USART driver instance ( SCI0 )
extern ARM_DRIVER_USART Driver_USART0;
static ARM_DRIVER_USART *gsp_sci0_dev = &Driver_USART0;

// Receive data
static uint8_t rx_data[6];

main()
{
    uint32_t arg;
    /* クロック同期 マスタモード */
    arg = ARM_USART_MODE_SYNCHRONOUS_MASTER |
          ARM_USART_CPOL0 | ARM_USART_CPHA0 |
          ARM_USART_FLOW_CONTROL_NONE;

    (void)gsp_sci0_dev->Initialize(usart_callback);              /* USART ドライバ初期化 */
    (void)gsp_sci0_dev->PowerControl(ARM_POWER_FULL);    /* USART のモジュールストップ解除 */
    (void)gsp_sci0_dev->Control(arg, 100000);                 /* クロック同期マスタモード
(100kbps) */
    (void)gsp_sci0_dev->Control(ARM_USART_CONTROL_TX_RX,1);  /* 送受信許可 */

    (void) gsp_sci0_dev->Receive(rx_data, 6);  /* 受信開始 */
    while(1);
}

/******************************************************************************
* callback function
******************************************************************************/
static void usart_callback(uint32_t event)
{
    if (0 != (event & ARM_USART_EVENT_RECEIVE_COMPLETE))
    {
        /* 正常に受信完了した場合の処理を記述 */
    }
    else
    {
        /* 通信異常が発生した場合の処理を記述 */
    }
}
```

図 5.2 クロック同期にて受信を使用する場合の設定例

5.4 端子設定について

本ドライバで使用する端子は、Control 関数で送信または受信が許可状態になったときに設定されます。使用する端子は、pin.c の R_SCI_Pinset_CHn、R_SCI_Pinclr_CHn (n=0～5,9)関数内を編集して設定します。

SCI0 を調歩同期にて使用、TXD0 端子を P703 に、RXD0 端子を P702 に設定、CTS0、SCK0 未使用とする場合の端子設定編集例を図 5.3～図 5.5 に示します。

```
/***********************************************************************//**
* @brief This function sets Pin of SCI0.
***********************************************************************/
/* Function Name : R_SCI_Pinset_CH0 */
void R_SCI_Pinset_CH0(void)  // @suppress("API function naming") @suppress("Function length")
{
    /* Disable protection for PFS function (Set to PWPR register) */
    R_SYS_RegisterProtectDisable(SYSTEM_REG_PROTECT_MPC);

//    /* CTS0 : P107 */
//    PFS->P107PFS_b.PMR  = 0U;
//    PFS->P107PFS_b.ASEL = 0U;

      ...

    /* When using SCI in I2C mode, set the pin to NMOS Open drain. */
////    PFS->P013PFS_b.NCODR = 1U;
////    PFS->P013PFS_b.PCODR = 0U;
//    PFS->P013PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04;
//    PFS->P013PFS_b.PMR  = 1U;

/* P703 を TXD0 用端子に設定 */
    /* TXD0 : P703 */
    PFS->P703PFS_b.ASEL = 0U;
    PFS->P703PFS_b.ISEL = 0U;

    /* When using SCI in I2C mode, set the pin to NMOS Open drain. */
//    PFS->P703PFS_b.NCODR = 1U;
//    PFS->P703PFS_b.PCODR = 0U;
    PFS->P703PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04;
    PFS->P703PFS_b.PMR  = 1U;

//    /* RXD0 : P105 */
//    PFS->P105PFS_b.PMR  = 0U;
//    PFS->P105PFS_b.ASEL = 0U;

      ...
```

図 5.3 端子設定編集例(1/3)

```
    /* When using SCI in I2C mode, set the pin to NMOS Open drain. */
////    PFS->P014PFS_b.NCODR = 1U;
////    PFS->P014PFS_b.PCODR = 0U;
//    PFS->P014PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04;
//    PFS->P014PFS_b.PMR  = 1U;

/* P702 を RXD0 用端子に設定*/
    /* RXD0 : P702 */
    PFS->P702PFS_b.ASEL = 0U;
    PFS->P702PFS_b.ISEL = 0U;

    /* When using SCI in I2C mode, set the pin to NMOS Open drain. */
//    PFS->P702PFS_b.NCODR = 1U;
//    PFS->P702PFS_b.PCODR = 0U;
    PFS->P702PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04;
    PFS->P702PFS_b.PMR  = 1U;

//    /* SCK0 : P104 */
//    PFS->P104PFS_b.PMR  = 0U;
//    PFS->P104PFS_b.ASEL = 0U;

      ...

//    /* SCK0 : P104 */
//    PFS->P104PFS_b.PMR  = 0U;
//    PFS->P104PFS_b.ASEL = 0U;
//    PFS->P104PFS_b.ISEL = 0U;
//    PFS->P104PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04;
//    PFS->P104PFS_b.PMR  = 1U;

    /* SCK0 : P015 */
//    PFS->P015PFS_b.ASEL = 0U;
//    PFS->P015PFS_b.ISEL = 0U;
//    PFS->P015PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04;
//    PFS->P015PFS_b.PMR  = 1U;

    /* SCK0 : P700 */
//    PFS->P700PFS_b.ASEL = 0U;
//    PFS->P700PFS_b.ISEL = 0U;
//    PFS->P700PFS_b.PSEL = R_PIN_PRV_SCI_PSEL_04;
//    PFS->P700PFS_b.PMR  = 1U;

    /* Enable protection for PFS function (Set to PWPR register) */
    R_SYS_RegisterProtectEnable(SYSTEM_REG_PROTECT_MPC);

}/* End of function R_SCI_Pinset_CH0() */
```

図 5.4 端子設定編集例(2/3)

```
/*****************************************************************************//**
* @brief This function clears the pin setting of SCI0.
*****************************************************************************/
/* Function Name : R_SCI_Pinclr_CH0 */
void R_SCI_Pinclr_CH0(void)  // @suppress("API function naming")
{
    /* Disable protection for PFS function (Set to PWPR register) */
    R_SYS_RegisterProtectDisable(SYSTEM_REG_PROTECT_MPC);

//    /* SCK0 : P104 */
//    PFS->P104PFS &= R_PIN_PRV_CLR_MASK;

    /* SCK0 : P015 */
//    PFS->P015PFS &= R_PIN_PRV_CLR_MASK;

    /* SCK0 : P700 */
//    PFS->P700PFS &= R_PIN_PRV_CLR_MASK;

//    /* RXD0 : P105 */
//    PFS->P105PFS &= R_PIN_PRV_CLR_MASK;

    /* RXD0 : P014 */
//    PFS->P014PFS &= R_PIN_PRV_CLR_MASK;

/* P702 を RXD0 用端子に設定*/
    /* RXD0 : P702 */
    PFS->P702PFS &= R_PIN_PRV_CLR_MASK;

//    /* TXD0 : P106 */
//    PFS->P106PFS &= R_PIN_PRV_CLR_MASK;

    /* TXD0 : P013 */
//    PFS->P013PFS &= R_PIN_PRV_CLR_MASK;

/* P703 を TXD0 用端子に設定 */
    /* TXD0 : P703 */
    PFS->P703PFS &= R_PIN_PRV_CLR_MASK;

//    /* CTS0 : P107 */
//    PFS->P107PFS &= R_PIN_PRV_CLR_MASK;

    /* CTS0 : P500 */
//    PFS->P500PFS &= R_PIN_PRV_CLR_MASK;

    /* CTS0 : P704 */
//    PFS->P704PFS &= R_PIN_PRV_CLR_MASK;

    /* Enable protection for PFS function (Set to PWPR register) */
    R_SYS_RegisterProtectEnable(SYSTEM_REG_PROTECT_MPC);
}/* End of function R_SCI_Pinclr_CH0() */
```

図 5.5 端子設定編集例(3/3)

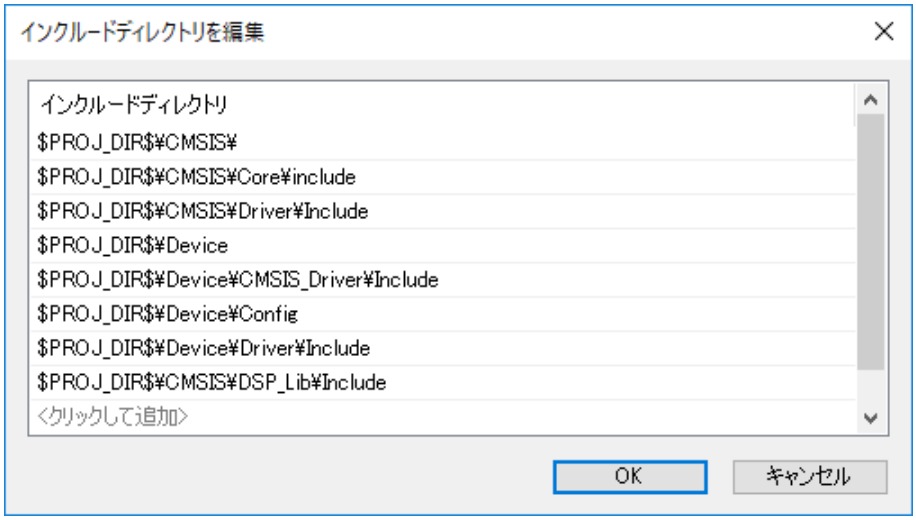
6. トラブルシューティング

6.1 ビルドエラーが発生する

A-1）インクルードディレクトリが設定されていますか？

EWARM をご使用の場合、下記の例の様にインクルードディレクトリを設定することを推奨します。

IDE のオプション [C/C++コンパイラ] -> [プリプロセッサ]から設定ができます。



6.2 CMSIS ドライバの API をコールすると HardFault Error が発生する

A) API の RAM 展開ができていない可能性があります。

RAM 上に配置した API をコールする前に R_SYS_CodeCopy 関数にて API を RAM 展開しているか確認してください。詳細は関連ドキュメント No. R01AN4660 をご参照ください。

6.3 API を呼び出しているが周辺機能が動作しない

A) API の設定が問題無くできていますか？

API の戻り値を確認し、エラー値が返っていないかをご確認ください。

特に r_system_cfg.h の割り込み設定がされていないことでエラー値が返っている事例が多く発生しています。詳細は関連ドキュメント No. R01AN4660 をご参照ください。

6.4 API の戻り値は正常であるが、周辺機能から端子出力が行われない

A) 端子設定は正しいでしょうか？

Pin.c の中にある関数にて端子設定が正しく行えているか確認してください。

詳細は関連ドキュメント No. R01AN4660 をご参照ください。

6.5 周辺機能の入力または出力が期待通り動作しない

A)周辺機能を初期設定する前に VOCR レジスタの設定が行えているか確認してください。

詳細は関連ドキュメント No. R01AN4660 をご参照ください。

7. サンプルコード

サンプルコードは、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。

8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RE01 1500KB グループ ユーザーズマニュアル　ハードウェア編 R01UH0796
RE01 256KB グループ ユーザーズマニュアル　ハードウェア編 R01UH0894

（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。）

RE01 1500KB, 256KB CMSIS Package スタートアップガイド

RE01 1500KB、256KB グループ CMSIS パッケージを用いた開発スタートアップガイド R01AN4660

（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／テクニカルニュース

（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。）

ユーザーズマニュアル：開発環境

（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。）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Aug.19.2019	-	初版発行
1.01	Jan.16.2020	-	RE01 256KB グループを追加
1.02	Mar.19.2020	3,6,8 - プログラム (256KB)	RE01 256KB グループのターゲットボードを Evaluation Kit RE01 256KB に変更 誤記修正 CMSIS Driver Package を差し替え - RE01 256KB: CMSIS Driver Package Rev.0.80
1.03	Jun.01.2020	3 6 プログラム (256KB)	サンプルコードプロジェクト名を変更 動作確認条件を更新 CMSIS Driver Package を差し替え - RE01 256KB: CMSIS Driver Package Rev.1.00

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. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   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、将来の拡張機能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。
8. 製品間の相違について
   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うと、フラッシュメモリ、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、ノイズ幅射量などが異なる場合があります。型名が違う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・システムの設計において、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て生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因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に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   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、産業用ロボット等

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 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・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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が直接的、間接的に支配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。

(Rev.4.0-1　2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。