

RE01 1500KB グループ、256KB グループ SPI マスタ/スレーブ通信 サンプルコード(using CMSIS Driver Package)

CMSIS Driver Package SPI 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	主として使用するドライバ
SPI ドライバを使用し、ROM データを RAM に転送します。	SPI を使用	R_SPI

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	3
1.1 プロジェクト説明	3
1.2 使用端子	3
1.3 フォルダ構成	4
1.4 ファイル構成	5
1.5 オプション設定メモリ	5
2. 動作確認条件	6
3. ソフトウェア説明	7
3.1 システム構成図	8
3.2 ドライバ設定変更	9
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	9
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	11
3.3 関数一覧	13
3.4 定数一覧	14
3.5 フローチャート	14
4. DTC、DMAC を使用した SPI 通信について	16
4.1 SPI0 の DMAC 設定	16
4.2 SPI1 の DTC 設定	16
5. ドライバの API 仕様	17
5.1 外部仕様書	17
6. R_SPI ドライバを使用する上での注意事項	17
6.1 DMAC/DTC を使用した SPI 通信について	17
6.2 NVIC への割り込み登録について	18
6.3 SS 端子をソフトウェア制御で出力する場合	19
6.4 端子設定について	20
6.5 スレーブモードかつ CPHA0 での通信再開について	23
7. トラブルシューティング	24
7.1 ビルドエラーが発生する	24
7.2 CMSIS ドライバの API をコールすると HardFault Error が発生する	24
7.3 API を呼び出しているが周辺機能が動作しない	24
7.4 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	24
7.5 周辺機能の入力または出力が期待通り動作しない	24
8. サンプルコード	25
9. 参考ドキュメント	25
改訂記録	26

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 : r01an4698_cmsis_spi_re_1500kb

RE01 256KB グループ用サンプルコードプロジェクト : r01an4698_cmsis_spi_re_256kb

r01an4698_cmsis_spi_re_1500kb 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。

r01an4698_cmsis_spi_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

以下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
P501	MOSIA_C
P502	RSPCKA_C
P012	SSLA0_B
P609	MOSIB_B
P607	RSPCKB_B
P610	SSLB0_B

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
P010	MOSIA_B
P011	RSPCKA_B
P014	SSLA0_B
P110	MOSIB_A
P108	RSPCKB_A
P106	SSLB0_A

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。

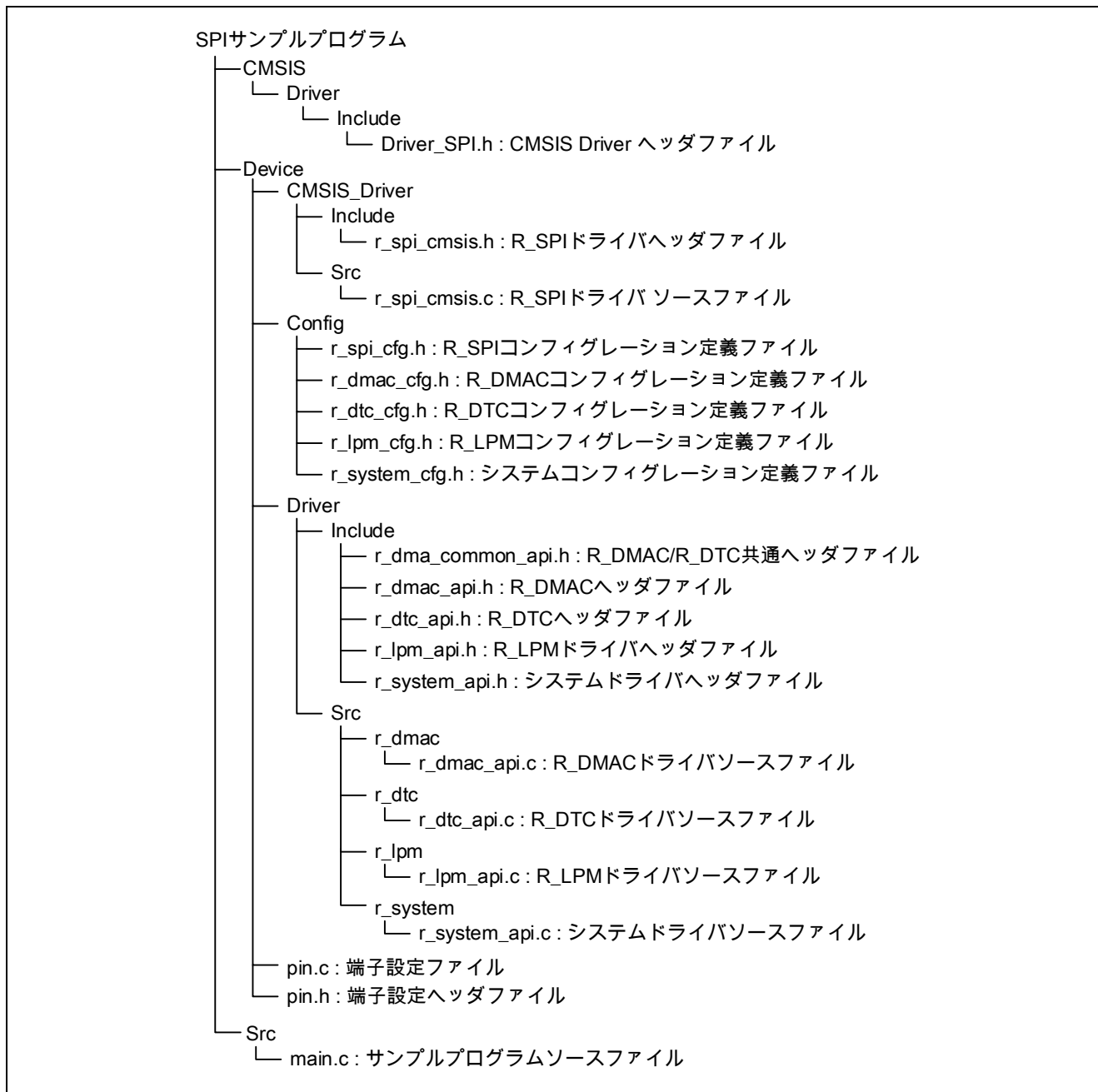


図 1.1 フォルダ構成

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	
r_spi_cfg.h	SPI 設定	SPI 割り込み発生時の DMAC、DTC 転送要因設定
pin.c	端子設定	SPI0、SPI1 端子設定
r_system_cfg.h	システム設定	SPI 転送完了割り込みの 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 動作確認条件 (RE01 1500KB グループ)

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	<ul style="list-style-type: none"> メインクロック: 32MHz PLL: 64MHz (メインクロック 4 分周 8 通倍) 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		<ul style="list-style-type: none"> 3.3V
統合開発環境	IAR	IAR Embedded Workbench for ARM Version 8.50.6 C コンパイラ : IAR C/C++ Compiler for ARM Version 8.50.6
	e ² studio	Renesas e ² studio Version 2021-01 C コンパイラ : GCC ARM Embedded Version 6.3.1.20170620 GNU 6-2017-q2-update
デバッグ		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		Rev1.10
サンプルコードのバージョン		Rev1.05

表 2-2 動作確認条件 (RE01 256KB グループ)

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	<ul style="list-style-type: none"> 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		<ul style="list-style-type: none"> 3.3V
統合開発環境	IAR	IAR Embedded Workbench for ARM Version 8.50.6 C コンパイラ : IAR C/C++ Compiler for ARM Version 8.50.6
	e ² studio	Renesas e ² studio 2021-01 C コンパイラ : GCC ARM Embedded Version 6.3.1.20170620 GNU 6-2017-q2-update
デバッグ		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		Rev1.05

3. ソフトウェア説明

本サンプルコードは、R_SPI ドライバを使用して、SPI0 をマスタ送信モードで送信したデータを、SPI1 のスレーブ受信モードで受信します。また、SPI0 の送信処理は DMAC を使用、SPI1 の受信は DTC を使用します。

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、SPI0、SPI1 の初期設定を行います。
- SPI1 のスレーブ受信、SPI0 のマスタ送信を開始します。
- SPI0 の送信完了割り込み、SPI1 の受信完了割り込みが完了後、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同じであれば、SPI1 のスレーブ受信、SPI0 のマスタ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(SPI0)

項目	設定値
マスタ/スレーブモード	マスタモード
転送速度	100kbps
RSPCK 位相	立ち上がりエッジでデータサンプリング、立ち下がりエッジでデータ変化を選択
RSPCK 極性	アイドル時の RSPCK が Low
SPI LSB ファースト	MSB ファースト
データ転送サイズ	5 バイト(40 ビット)

表 3-2 サンプルプログラムの動作情報(SPI1)

項目	設定値
マスタ/スレーブモード	スレーブモード
RSPCK 位相	立ち上がりエッジでデータサンプリング、立ち下がりエッジでデータ変化を選択
RSPCK 極性	アイドル時の RSPCK が Low
SPI LSB ファースト	MSB ファースト
データ転送サイズ	5 バイト(40 ビット)

3.1 システム構成図

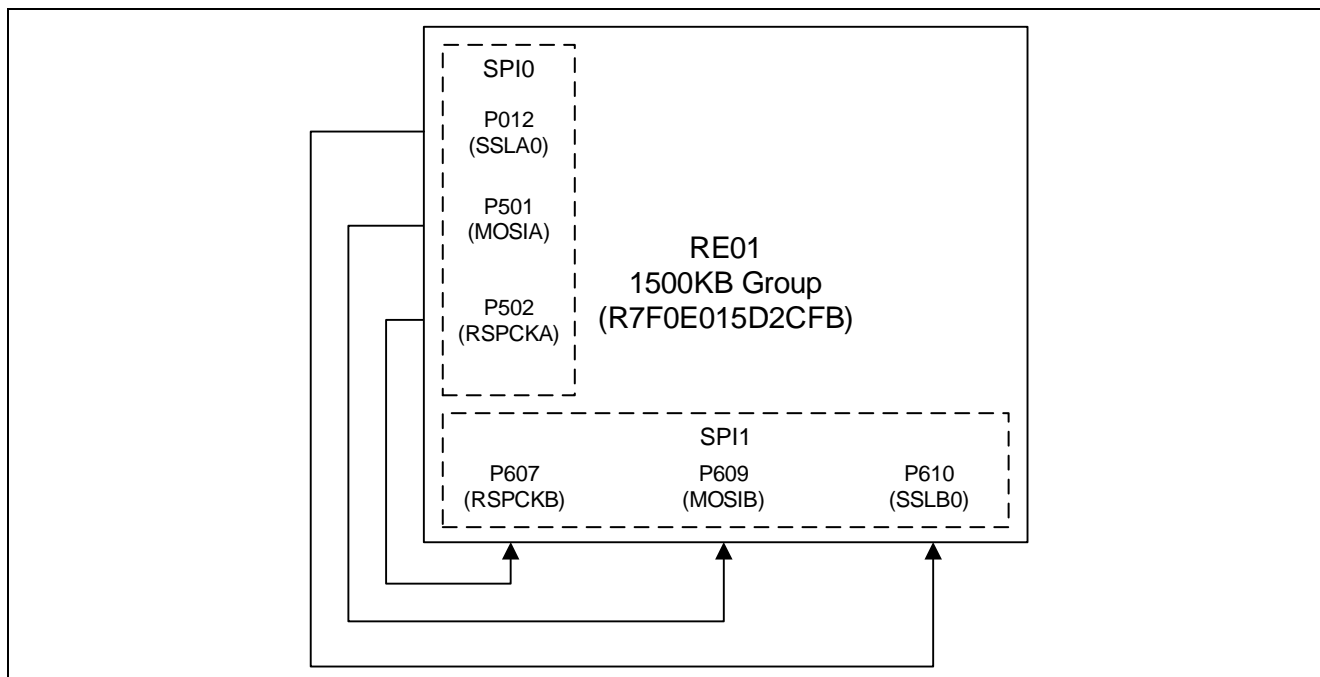


図 3.1 RE01 1500KB グループ用サンプルコード システム構成図

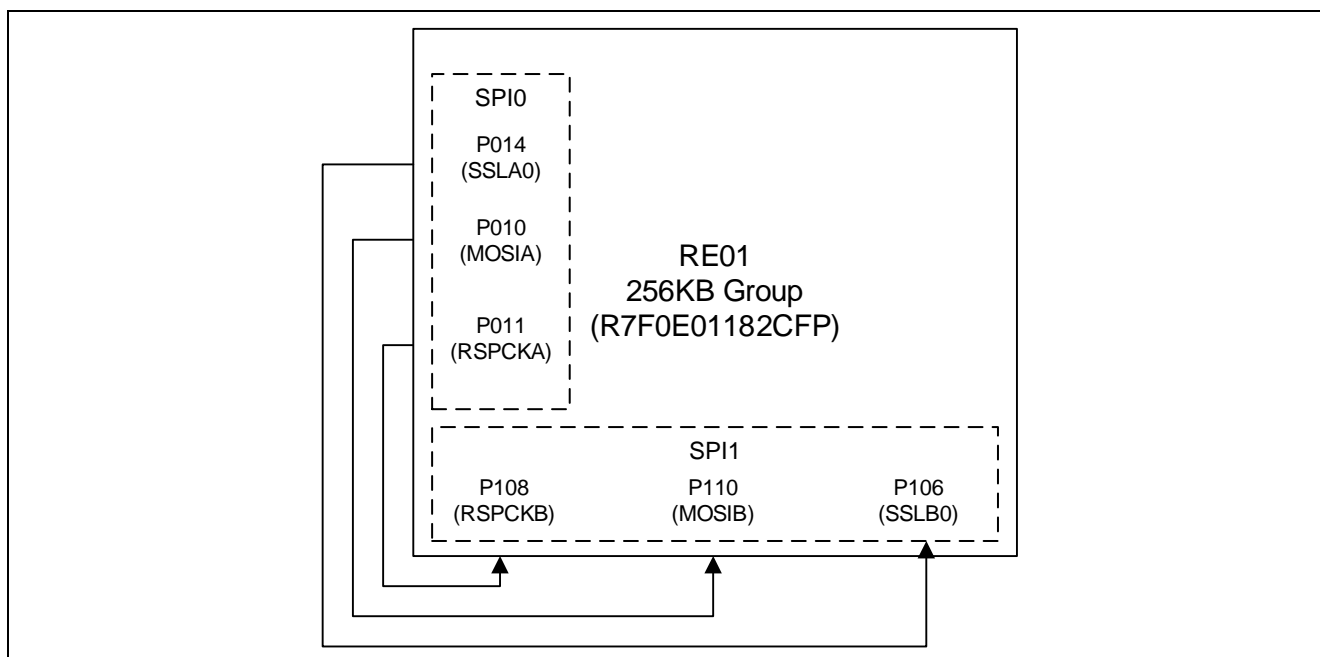


図 3.2 RE01 256KB グループ用サンプルコード システム構成図

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(1/2)

項目	変更箇所	変更内容
SPI 送信完了時、DMAC で送信データを転送するように変更	[r_spi_cfg.h] SPI0_TRANSMIT_CONTROL	● 設定値変更 SPI_USED_DMACH
SPI 受信完了時、DTC で受信データを転送するように変更	[r_spi_cfg.h] SPI1_RECEIVE_CONTROL	● 設定値変更 SPI_USED_DTC
NVIC への SPI0 アイドル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI0_SPII	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER18)
NVIC への SPI1 送信バッファエンプティ割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI1_SPTI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER5)
NVIC への SPI1 受信バッファフル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI1_SPRI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER28)
NVIC への SPI1 エラー割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI1_SPEI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER7)
NVIC への DMACH 転送完了割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_DMACH_INT	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER0)

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

項目	変更箇所	変更内容
MOSIA を P501 に設定	[pin.c] R_RSPI_Pinset_CH0 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P501PFS_b.PMR = 0U; PFS->P501PFS_b.ASEL = 0U; PFS->P501PFS_b.ISEL = 0U; PFS->P501PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P501PFS_b.PMR = 1U;
RSPCKA を P502 に設定	[pin.c] R_RSPI_Pinset_CH0 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P502PFS_b.PMR = 0U; PFS->P502PFS_b.ASEL = 0U; PFS->P502PFS_b.ISEL = 0U; PFS->P502PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P502PFS_b.PMR = 1U;
SSLA0 を P012 に設定	[pin.c] R_RSPI_Pinset_CH0 関数	<ul style="list-style-type: none"> 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_PRIV_RSPI_PSEL; PFS->P012PFS_b.PMR = 1U;
MOSIB を P609 に設定	[pin.c] R_RSPI_Pinset_CH1 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P609PFS_b.PMR = 0U; PFS->P609PFS_b.ASEL = 0U; PFS-> P609PFS_b.ISEL = 0U; PFS-> P609PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS-> P609PFS_b.PMR = 1U;
RSPCLKB を P607 に設定	[pin.c] R_RSPI_Pinset_CH1 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P607PFS_b.PMR = 0U; PFS->P607PFS_b.ASEL = 0U; PFS->P607PFS_b.ISEL = 0U; PFS->P607PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P607PFS_b.PMR = 1U;
SSLB0 を P610 に設定	[pin.c] R_RSPI_Pinset_CH1 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P610PFS_b.PMR = 0U; PFS->P610PFS_b.ASEL = 0U; PFS->P610PFS_b.ISEL = 0U; PFS->P610PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P610PFS_b.PMR = 1U;

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 3-5 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
SPI 送信完了時、DMAC で送信データを転送するように変更	[r_spi_cfg.h] SPI0_TRANSMIT_CONTROL	● 設定値変更 SPI_USED_DMACH0
SPI 受信完了時、DTC で受信データを転送するように変更	[r_spi_cfg.h] SPI1_RECEIVE_CONTROL	● 設定値変更 SPI_USED_DTC
NVIC への SPI0 アイドル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI0_SPII	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER18)
NVIC への SPI1 送信バッファエンプティ割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI1_SPTI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER5)
NVIC への SPI1 受信バッファフル割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI1_SPRI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER28)
NVIC への SPI1 エラー割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_SPI1_SPEI	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER7)
NVIC への DMACH0 転送完了割り込み登録	[r_system_cfg.h] SYSTEM_CFG_EVENT_NUMBER_DMACH0_INT	● 設定値変更 (SYSTEM_IRQ_EVENT_NUMBER0)

表 3-6 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
MOSIA を P010 に設定	[pin.c] R_RSPI_Pinset_CH0 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P010PFS_b.PMR = 0U; PFS->P010PFS_b.ASEL = 0U; PFS->P010PFS_b.ISEL = 0U; PFS->P010PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P010PFS_b.PMR = 1U;
RSPCKA を P011 に設定	[pin.c] R_RSPI_Pinset_CH0 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P011PFS_b.PMR = 0U; PFS->P011PFS_b.ASEL = 0U; PFS->P011PFS_b.ISEL = 0U; PFS->P011PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P011PFS_b.PMR = 1U;
SSLA0 を P014 に設定	[pin.c] R_RSPI_Pinset_CH0 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P014PFS_b.PMR = 0U; PFS->P014PFS_b.ASEL = 0U; PFS->P014PFS_b.ISEL = 0U; PFS->P014PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P014PFS_b.PMR = 1U;
MOSIB を P110 に設定	[pin.c] R_RSPI_Pinset_CH1 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P110PFS_b.ASEL = 0U; PFS->P110PFS_b.ISEL = 0U; PFS->P110PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P110PFS_b.PMR = 1U;
RSPCKB を P108 に設定	[pin.c] R_RSPI_Pinset_CH1 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P108PFS_b.PMR = 0U; PFS->P108PFS_b.ASEL = 0U; PFS->P108PFS_b.ISEL = 0U; PFS->P108PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P108PFS_b.PMR = 1U;
SSLB0 を P106 に設定	[pin.c] R_RSPI_Pinset_CH1 関数	<ul style="list-style-type: none"> 以下の部分をコメントアウト解除 PFS->P106PFS_b.PMR = 0U; PFS->P106PFS_b.ASEL = 0U; PFS->P106PFS_b.ISEL = 0U; PFS->P106PFS_b.PSEL = R_PIN_PRIV_RSPI_PSEL; PFS->P106PFS_b.PMR = 1U;

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	システム初期設定関数を呼び出します。その後、SPI 通信設定を行い、SPI 送受信を行います。送受信が完了すると、送信データと受信データを比較します。
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

spi0_callback	
概要	SPI0 転送完了コールバック処理
ヘッダ	なし
宣言	static void spi0_callback(uint32_t event)
説明	SPI0 送信データ転送完了時、送信完了フラグをセットします。
引数	uint32_t event コールバック発生要因 ARM_SPI_EVENT_TRANSFER_COMPLETE 転送完了
リターン値	なし

spi1_callback	
概要	SPI1 転送完了コールバック処理
ヘッダ	なし
宣言	static void spi1_callback(uint32_t event)
説明	SPI1 受信データ転送完了時、受信完了フラグをセットします。
引数	uint32_t event コールバック発生要因 ARM_SPI_EVENT_TRANSFER_COMPLETE 転送完了
リターン値	なし

3.4 定数一覧

表 3-7 に定数一覧を示します。

表 3-7 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
SPI_TRANSFER_SPEED	100000U	SPI 転送速度
SPI_DATA_SIZE	5	SPI 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。

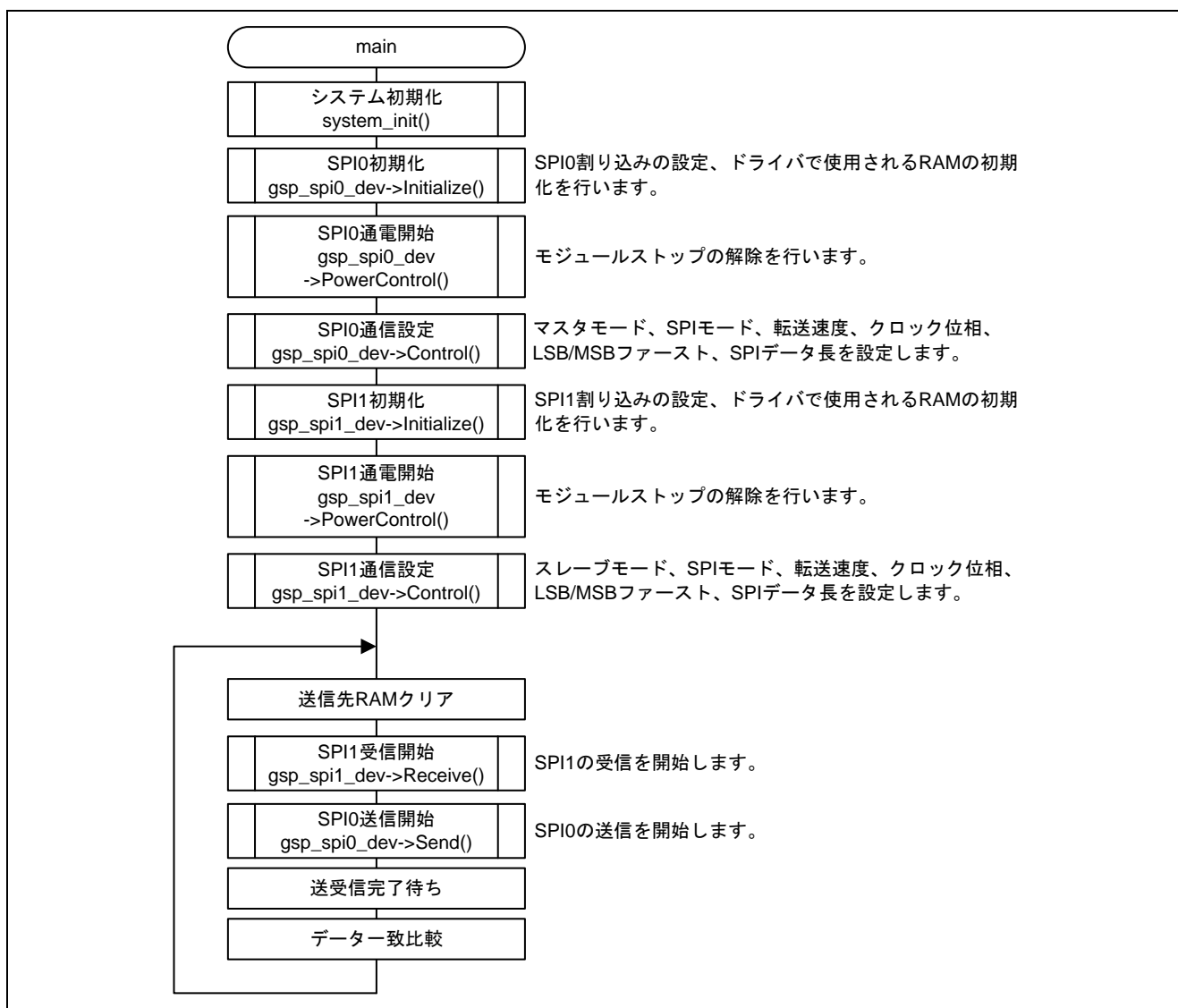


図 3.3 メイン処理

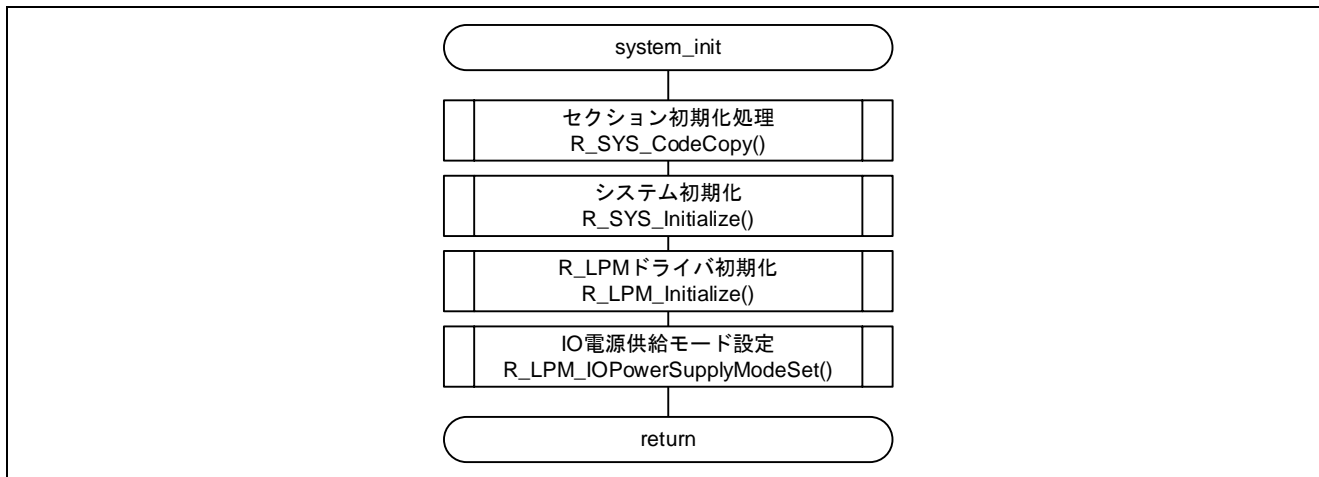


図 3.4 システム初期処理

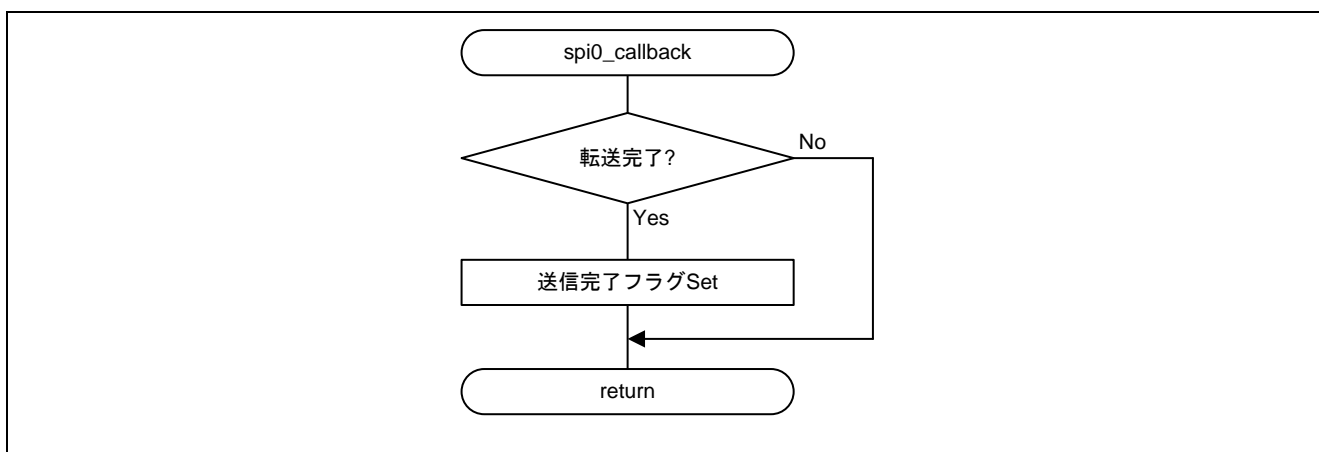


図 3.5 SPI0 転送完了コールバック処理

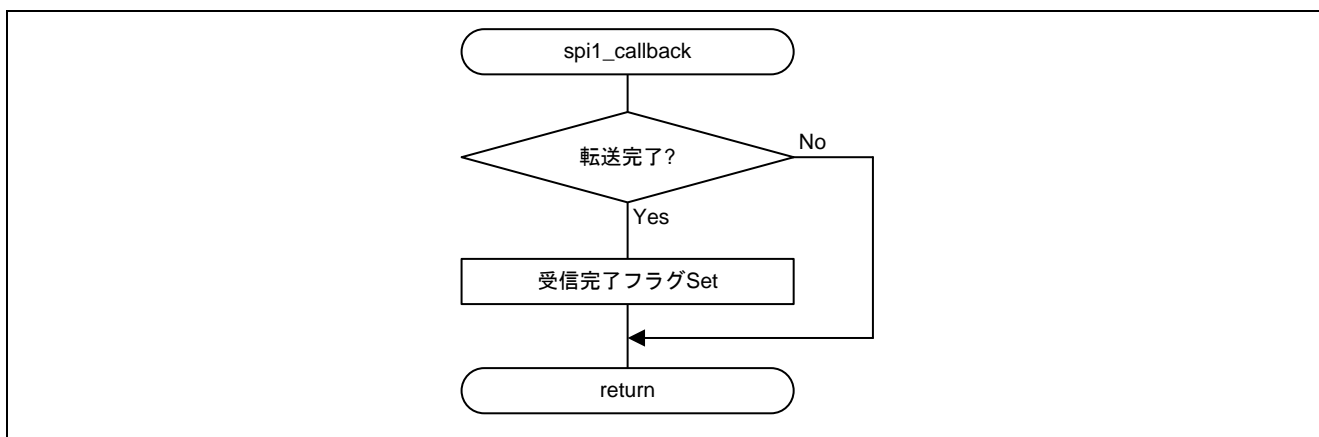


図 3.5 SPI1 転送完了コールバック処理

4. DTC、DMAC を使用した SPI 通信について

本サンプルコードでは、SPI0 の送信処理を DMAC0 で、SPI1 の受信処理を DTC で行っています。r_spi_cfg.h および r_system_cfg.h の設定を図に示します。

4.1 SPI0 の DMAC 設定

SPI0(マスタ送信)での設定を、表 4-1、表 4-2 に示します。

表 4-1 r_spi_cfg.h 設定変更箇所

項目	変更内容	
送信データを DMAC0 で転送する場合	SPI0_TRANSMIT_CONTROL	SPI_USED_INTERRUPT →SPI_USED_DMAC0

表 4-2 r_system_cfg.h 設定変更箇所

項目	変更内容	
NVIC への DMAC0 転送割り込み登録	SYSTEM_CFG_EVENT_NUMBER_DMCA0_INT	SYSTEM_IRQ_EVENT_NUMBER0

4.2 SPI1 の DTC 設定

SPI1(スレーブ受信)での設定を、表 4-3 に示します。

表 4-3 r_spi_cfg.h 設定変更箇所

項目	変更内容	
受信データを DTC で転送する場合	SPI1_RECEIVE_CONTROL	SPI_USED_INTERRUPT →SPI_USED_DTC

5. ドライバの API 仕様

5.1 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。

6. R_SPI ドライバを使用する上での注意事項

本章では R_SPI 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ていません。
注意点について『5.1 外部仕様書』をご参照ください。

6.1 DMAC/DTC を使用した SPI 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_spi_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 6-1 に、送信/受信制御方法の定義を表 6-2 に示します。

表 6-1 送信/受信制御方法の設定定義 (n=0、1)

定義	初期値	内容
SPI _n _TRANSMIT_CONTROL	SPI_USED_INTERRUPT	SPI _n の送信制御 (初期値: 割り込み)
SPI _n _RECEIVE_CONTROL	SPI_USED_INTERRUPT	SPI _n の受信制御 (初期値: 割り込み)

表 6-2 送信/受信制御方法の定義

定義	値	内容
SPI_USED_INTERRUPT	(0)	送信/受信制御に割り込みを使用
SPI_USED_DMACH0	(1<<0)	送信/受信制御に DMACH0 を使用
SPI_USED_DMACH1	(1<<1)	送信/受信制御に DMACH1 を使用
SPI_USED_DMACH2	(1<<2)	送信/受信制御に DMACH2 を使用
SPI_USED_DMACH3	(1<<3)	送信/受信制御に DMACH3 を使用
SPI_USED_DTC	(1<<15)	送信/受信制御に DTC を使用

6.2 NVIC への割り込み登録について

SPI 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 6-3 に、NVIC への割り込み登録例を図 6.1 に示します。

表 6-3 使用用途に対する NVIC の登録定義

使用用途	NVIC 登録定義	備考
マスタ送信で使用	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPTI	(注 1)
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPII	
マスタ送受信で使用(注 4)	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPTI	(注 1)
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPRI	(注 3)
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPII	
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPEI	
スレーブ送信で使用	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPTI	(注 1)
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPEI	
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPTEND	
スレーブ送受信で使用 (注 4)	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPTI	(注 1)
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPRI	(注 2)
	SYSTEM_CFG_EVENT_NUMBER_SPIn_SPEI	

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

注3. 受信制御に DMACm(m=0~3)を使用する場合は、NVIC への登録は不要です。

注4. 受信のみ実施する場合でも、ダミーデータの送信が必要なため、送信の設定も必要です。

```

. . .
#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_SPI0_SPRI
    (SYSTEM_IRQ_EVENT_NUMBER0) /*!< Numbers 0/4/8/12/16/20/24/28 only */
#define SYSTEM_CFG_EVENT_NUMBER_SOL_DH
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 0/8/16/24 only */
. . .
#define SYSTEM_CFG_EVENT_NUMBER_SCI0_TXI
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 1/5/9/13/17/21/25/29 only */
#define SYSTEM_CFG_EVENT_NUMBER_SPI0_SPTI
    (SYSTEM_IRQ_EVENT_NUMBER1) /*!< Numbers 1/5/9/13/17/21/25/29 only */
#define SYSTEM_CFG_EVENT_NUMBER_SOL_DL
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 1/9/17/25 only */
. . .
#define SYSTEM_CFG_EVENT_NUMBER_SCI0_TEI
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 2/6/10/14/18/22/26/30 only */
#define SYSTEM_CFG_EVENT_NUMBER_SPI0_SPTI
    (SYSTEM_IRQ_EVENT_NUMBER2) /*!< Numbers 2/6/10/14/18/22/26/30 only */
#define SYSTEM_CFG_EVENT_NUMBER_SPI0_SPTEND
    (SYSTEM_IRQ_EVENT_NUMBER6) /*!< Numbers 2/6/10/14/18/22/26/30 only */
#define SYSTEM_CFG_EVENT_NUMBER_USBFS_USBI
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 2/10/18/26 only */
. . .

```

図 6.1 NVIC への割り込み登録例(SPI0 使用時)

6.3 SS 端子をソフトウェア制御で出力する場合

SS 端子をソフトウェア (ARM_SPI_CONTROL_SS コマンドを使用した Control 関数) で制御する場合、SS 端子として使用する端子を r_spi_cfg.h で指定してください。

SS 端子をソフトウェア制御で使用する場合の端子指定例を図 6.2 に示します。

```

. . .
/* When using the ARM_SPI_CONTROL_SS command, cancel the following comment and
   set the terminal to use */
#define SPI0_SS_PORT PORT1->PODR /* コメントアウトを削除、ポート 1 選択 */
#define SPI0_SS_PIN 7 /* SS 端子を P107 に選択 */
. . .

```

図 6.2 SS 端子をソフトウェア制御で使用する場合の端子指定例

6.4 端子設定について

本ドライバで使用する端子は、Control 関数で動作モードを選択したときに設定されます。使用する端子は、pin.c の R_RSPI_Pinset_CHn、R_SPI_RPinclr_CHn (n=0、1)関数内を編集して設定します。SPI 機能で使用する各端子名には_A、_B、_C および_D という接尾語が付加されています。SPI 機能を割り当てる場合、同じ接尾語の機能端子を選択してください。(注)

また、SPI0 の MISOA 端子を P500 に、MOSIA 端子を P010 に、RSPCKA 端子を P011 に、SSLA0 端子を P012 に設定する場合の端子設定変更例を図 6.3、図 6.4、図 6.5 に示します。

注 接尾語が同じ信号は、タイミング調整されているグループを表しています。違うグループの信号を同時に使用することはできません。例外として、SPI の「RSPCKA_C」と「MOSIA_C」は「_B」のグループと同時に使用します。また、「SSLB0_D」は「_B」のグループと同時に使用します。

```

/*****
* @brief This function sets Pin of RSPI0.
* @note Several pin names have added _A, _B, and _C suffixes.@n
*       When assigning the SPI functions, select the functional pins with the same suffix.@n
*       Comment out the terminal of unused suffix.@n
*       When using "RSPCKA_C, MOSIA_C" added by the SPI, select the pair of RSPCKA_B and RSPCKA_C
*       and the pair of MOSIA_B and MOSIA_C. When using "SSLB0_D" added by SPI,
*       select the pair of SSLB0_D and SSLB0_B.
*****/
/* Function Name : R_RSPI_Pinset_CH0 */
void R_RSPI_Pinset_CH0(void) // @suppress("API function naming") @suppress("Function length")
{
    /* Disable protection for PFS function (Set to PWRP register) */
    R_SYS_RegisterProtectDisable(SYSTEM_REG_PROTECT_MPC);

    // /* MISOA_A : P105 */
    // PFS->P105PFS_b.PMR = 0U;
    // PFS->P105PFS_b.ASEL = 0U;
    // PFS->P105PFS_b.ISEL = 0U;
    // PFS->P105PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
    // PFS->P105PFS_b.PMR = 1U;

    /* P500 を MISOA 端子に設定*/
    /* MISOA_B : P500 */
    PFS->P500PFS_b.PMR = 0U;
    PFS->P500PFS_b.ASEL = 0U;
    PFS->P500PFS_b.ISEL = 0U;
    PFS->P500PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
    PFS->P500PFS_b.PMR = 1U;

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

    /* P010 を MOSIA 端子に設定 */
    /* MOSIA_B : P010 */
    PFS->P010PFS_b.PMR = 0U;
    PFS->P010PFS_b.ASEL = 0U;
    PFS->P010PFS_b.ISEL = 0U;
    PFS->P010PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
    PFS->P010PFS_b.PMR = 1U;

    /* MOSIA_C : P501 */
    // PFS->P501PFS_b.PMR = 0U;
    // PFS->P501PFS_b.ASEL = 0U;
    // PFS->P501PFS_b.ISEL = 0U;
    // PFS->P501PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
    // PFS->P501PFS_b.PMR = 1U;

```

図 6.3 端子設定変更例(1/3)

```

// /* RSPCKA_A : P107 */
// PFS->P107PFS_b.PMR = 0U;
// PFS->P107PFS_b.ASEL = 0U;
// PFS->P107PFS_b.ISEL = 0U;
// PFS->P107PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
// PFS->P107PFS_b.PMR = 1U;

/* P011 を RSPCKA 端子に設定 */
/* RSPCKA_B : P011 */
PFS->P011PFS_b.PMR = 0U;
PFS->P011PFS_b.ASEL = 0U;
PFS->P011PFS_b.ISEL = 0U;
PFS->P011PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
PFS->P011PFS_b.PMR = 1U;

/* RSPCKA_C : P502 */
// PFS->P502PFS_b.PMR = 0U;
// PFS->P502PFS_b.ASEL = 0U;
// PFS->P502PFS_b.ISEL = 0U;
// PFS->P502PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
// PFS->P502PFS_b.PMR = 1U;

// /* SSLA0_A : P103 */
// PFS->P103PFS_b.PMR = 0U;
// PFS->P103PFS_b.ASEL = 0U;
// PFS->P103PFS_b.ISEL = 0U;
// PFS->P103PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
// PFS->P103PFS_b.PMR = 1U;

/* P012 を SSLA0 端子に設定 */
/* SSLA0_B : P012 */
PFS->P012PFS_b.PMR = 0U;
PFS->P012PFS_b.ASEL = 0U;
PFS->P012PFS_b.ISEL = 0U;
PFS->P012PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
PFS->P012PFS_b.PMR = 1U;

// /* SSLA1_A : P102 */
// PFS->P102PFS_b.PMR = 0U;
// PFS->P102PFS_b.ASEL = 0U;
// PFS->P102PFS_b.ISEL = 0U;
// PFS->P102PFS_b.PSEL = R_PIN_PRV_RSPI_PSEL;
// PFS->P102PFS_b.PMR = 1U;

...

/* Enable protection for PFS function (Set to PWPR register) */
R_SYS_RegisterProtectEnable(SYSTEM_REG_PROTECT_MPC);
}/* End of function R_RSPI_Pinset_CH0() */

```

図 6.4 端子設定変更例(2/3)

```

/*****//**
 * @brief This function clears the pin setting of RSPI0.
 *****/
/* Function Name : R_RSPI_Pinclr_CH0 */
void R_RSPI_Pinclr_CH0(void) // @suppress("API function naming")
{
    /* Disable protection for PFS function (Set to PWPR register) */
    R_SYS_RegisterProtectDisable(SYSTEM_REG_PROTECT_MPC);

    // /* MISOA_A : P105 */
    // PFS->P105PFS &= R_PIN_PRIV_CLR_MASK;

    /* P500 を MISOA 端子に設定*/
    /* MISOA_B : P500 */
    PFS->P500PFS &= R_PIN_PRIV_CLR_MASK;

    // /* MOSIA_A : P104 */
    // PFS->P104PFS &= R_PIN_PRIV_CLR_MASK;

    /* P100 を MOSIA 端子に設定 */
    /* MOSIA_B : P010 */
    PFS->P010PFS &= R_PIN_PRIV_CLR_MASK;

    /* MOSIA_C : P501 */
    // PFS->P501PFS &= R_PIN_PRIV_CLR_MASK;

    // /* RSPCKA_A : P107 */
    // PFS->P107PFS &= R_PIN_PRIV_CLR_MASK;

    /* P011 を RSPCKA 端子に設定 */
    /* RSPCKA_B : P011 */
    PFS->P011PFS &= R_PIN_PRIV_CLR_MASK;

    /* RSPCKA : P502 */
    // PFS->P502PFS &= R_PIN_PRIV_CLR_MASK;

    // /* SSLA0_A : P103 */
    // PFS->P103PFS &= R_PIN_PRIV_CLR_MASK;

    /* P012 を SSLA0 端子に設定 */
    /* SSLA0_B : P012 */
    PFS->P012PFS &= R_PIN_PRIV_CLR_MASK;

    // /* SSLA1_A : P102 */
    // PFS->P102PFS &= R_PIN_PRIV_CLR_MASK;

    /* SSLA1_B : P013 */
    // PFS->P013PFS &= R_PIN_PRIV_CLR_MASK;

    // /* SSLA2_A : P101 */
    // PFS->P101PFS &= R_PIN_PRIV_CLR_MASK;

    /* SSLA2_B : P014 */
    // PFS->P014PFS &= R_PIN_PRIV_CLR_MASK;

    // /* SSLA3_A : P100 */
    // PFS->P100PFS &= R_PIN_PRIV_CLR_MASK;

    /* SSLA3_B : P015 */
    // PFS->P015PFS &= R_PIN_PRIV_CLR_MASK;

    /* Enable protection for PFS function (Set to PWPR register) */
    R_SYS_RegisterProtectEnable(SYSTEM_REG_PROTECT_MPC);
}/* End of function R_RSPI_Pinclr_CH0() */

```

図 6.5 端子設定変更例(3/3)

6.5 スレーブモードかつ CPHA0 での通信再開について

スレーブモードで CPHA (クロック位相) を 0 (立ち上がりエッジでデータサンプリング、立ち下がりエッジでデータ変化) に設定した場合、クロック信号 RSPCK の最後の半サイクル期間に通信を再開するとアンダランエラーやビットずれなどの不正な動作を行う可能性があります。コールバック関数の呼び出しもしくは GetStatus 関数によるレディー判定後、再度通信を開始する場合は RSPCK の半サイクル待つてから実行してください。スレーブモードかつ CPHA0 設定時の通信再開例を図 6-6 に示します。

```
static uint8_t tx_data[3] = {0x01, 0x02, 0x03};

/*****
 * callback function
 *****/
static void spi_callback(uint32_t event)
{
    switch( event )
    {
        case ARM_SPI_EVENT_TRANSFER_COMPLETE:
        {
            /* RSPCK の半サイクル待ち(通信速度 100kbps の場合 5us) */
            R_SYS_SoftwareDelay(5, SYSTEM_DELAY_UNITS_MICROSECONDS);
            /* 再開 */
            spi0Drv->Send(&tx_data[0], 3);
        }
        break;

        case ARM_SPI_EVENT_DATA_LOST:
        default:
        {
            /* 通信異常が発生した場合の処理を記述 */
        }
        break;
    }
}

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

図 6-6 スレーブモードかつ CPHA0 での通信再開例

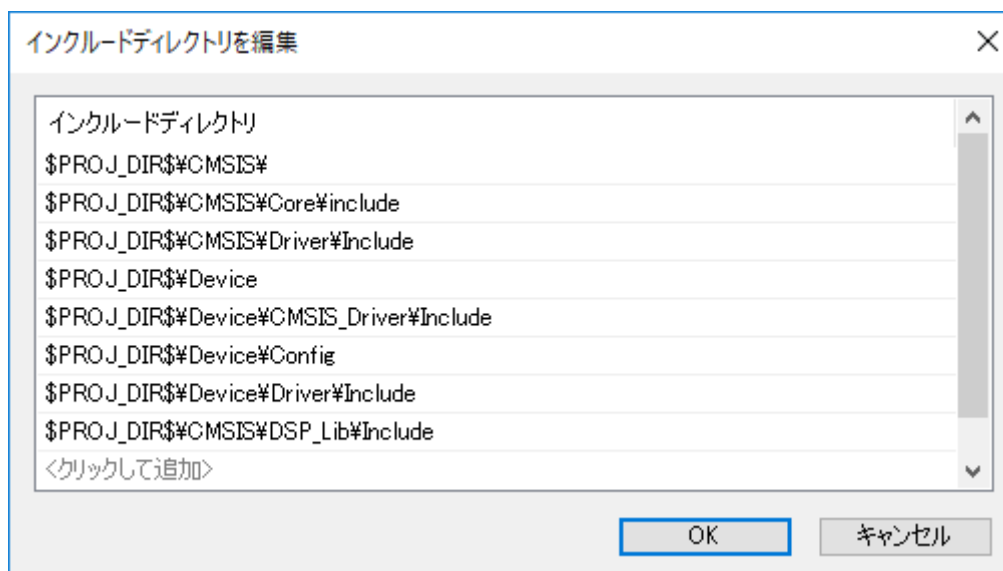
7. トラブルシューティング

7.1 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。



7.2 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。

7.3 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。

7.4 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。

7.5 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 VOCCR 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。

8. 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。

9. 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.20.2020	-	RE01 256KB グループを追加
1.02	Mar.19.2020	3,6 10,20,21 - プログラム	RE01 256KB グループのターゲットボードを Evaluation Kit RE01 256KB に変更 RE01 1500KB グループ pin.c(ver.1.10)の仕様にあわせて端子設定説明を変更 誤記修正 CMSIS Driver Package を差し替え - RE01 1500KB: CMSIS Driver Package Rev.1.10 - 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
1.04	Nov.05.2020	- 6 プログラム	誤記修正 統合開発環境を更新 不要な割り込み設定 (SPI0 の SPTI、SPI1 の SPTEND) を修正
1.05	Jan.11.2021	3 8 10, 12 プログラム	使用端子を更新 サンプルコードプロジェクト名を変更 システム構成図を更新 ドライバ変更箇所に SSLA0 端子設定を記載 SPI0(マスタ)の SSLA0 を有効に変更 サンプルコードプロジェクト名を変更

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えば、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、ノイズ輻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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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したします。
13. 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。
14. 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.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレストシア）

www.renesas.com

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。

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/