

RZ/T1 グループ

シリアルフラッシュサンプルプログラム(SPIBSC)

R01AN3010JJ0140

Rev.1.40

2018.06.07

要旨

本アプリケーションノートは、RZ/T1 グループマイコンに搭載されている SPI マルチ I/O バスコントローラの設定方法について説明します。

制限事項

本サンプルプログラムには以下の制限事項があります。

- (1) シリアルフラッシュのエリアのみに対応しています。他のエリアにリード/ライトすることはできません。

対象デバイス

RZ/T1 グループ

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。

目次

1.	仕様	4
2.	動作環境	5
3.	関連アプリケーションノート	6
4.	ハードウェア説明	7
4.1	使用端子一覧	7
4.2	参考回路	8
5.	ソフトウェア説明	9
5.1	動作概要	9
5.1.1	プロジェクト設定	10
5.1.2	使用準備	11
5.2	固定幅整数一覧	12
5.3	構造体 / 共用体一覧	13
5.4	定数一覧	20
5.5	変数一覧	23
5.6	関数一覧	23
5.7	関数詳細	25
5.8	サンプルプログラムの関数フロー	32
5.8.1	SPIBSC 初期設定関数	32
5.8.2	メイン関数	33
5.9	サンプルプログラム動作説明	34
5.9.1	デモプログラム	34
5.9.2	シリアルフラッシュ読み出し	35
5.9.3	シリアルフラッシュ書き込み	36
5.9.4	シリアルフラッシュ消去	36
6.	SPIBSC モード切り替え手順	37
6.1	分岐予測 / 投機実行により発生する可能性のある問題について	37
6.1.1	Cortex-R4 における ROM/RAM アクセス	37
6.1.2	SPI マルチ I/O バスコントローラのモード	37
6.2	モード切り替え方法 概略	37
6.2.1	モードの切り替えについて	37
6.2.2	変更手順概略	37
7.	応用例	38
7.1	サンプルプログラムの条件	38
7.2	シリアルフラッシュメモリを変更しない場合のサンプルプログラム変更方法	38
7.3	シリアルフラッシュメモリを変更する場合のサンプルプログラム変更方法	39
7.3.1	リードコマンド波形の変更	40
7.3.2	シリアルフラッシュメモリ内レジスタ設定	42
7.3.3	シリアルフラッシュメモリライト許可	47
7.3.4	シリアルフラッシュメモリレディー待ち	48

7.3.5	参考：シリアルフラッシュメモリプロテクト解除	49
8.	サンプルプログラム.....	50
9.	参考ドキュメント	51

1. 仕様

SPI マルチ I/O バスコントローラ（以下、SPIBSC とします。）は、SPI マルチ I/O バス空間（以下、SPIBSC 空間とします。）に接続されたシリアルフラッシュメモリに対し、制御信号を出力するバスコントローラです。SPIBSC を設定することにより、SPIBSC 空間に接続されたシリアルフラッシュメモリに対し、直接リードおよび SPI モードによるデータ送受信を行うことができます。

本アプリケーションノートは、SPIBSC 空間から直接リードするための SPIBSC の設定方法および、シリアルフラッシュメモリを 4 ビットでアクセスするため必要な、SPI モードによるシリアルフラッシュメモリ内レジスタの設定方法を説明します。

表 1.1 に使用する周辺機能と用途を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
SPI マルチ I/O バスコントローラ (SPIBSC)	外部アドレス空間 (SPI マルチ I/O バス空間) に接続されたシリアルフラッシュメモリにアクセスするための信号を生成するために使用します。

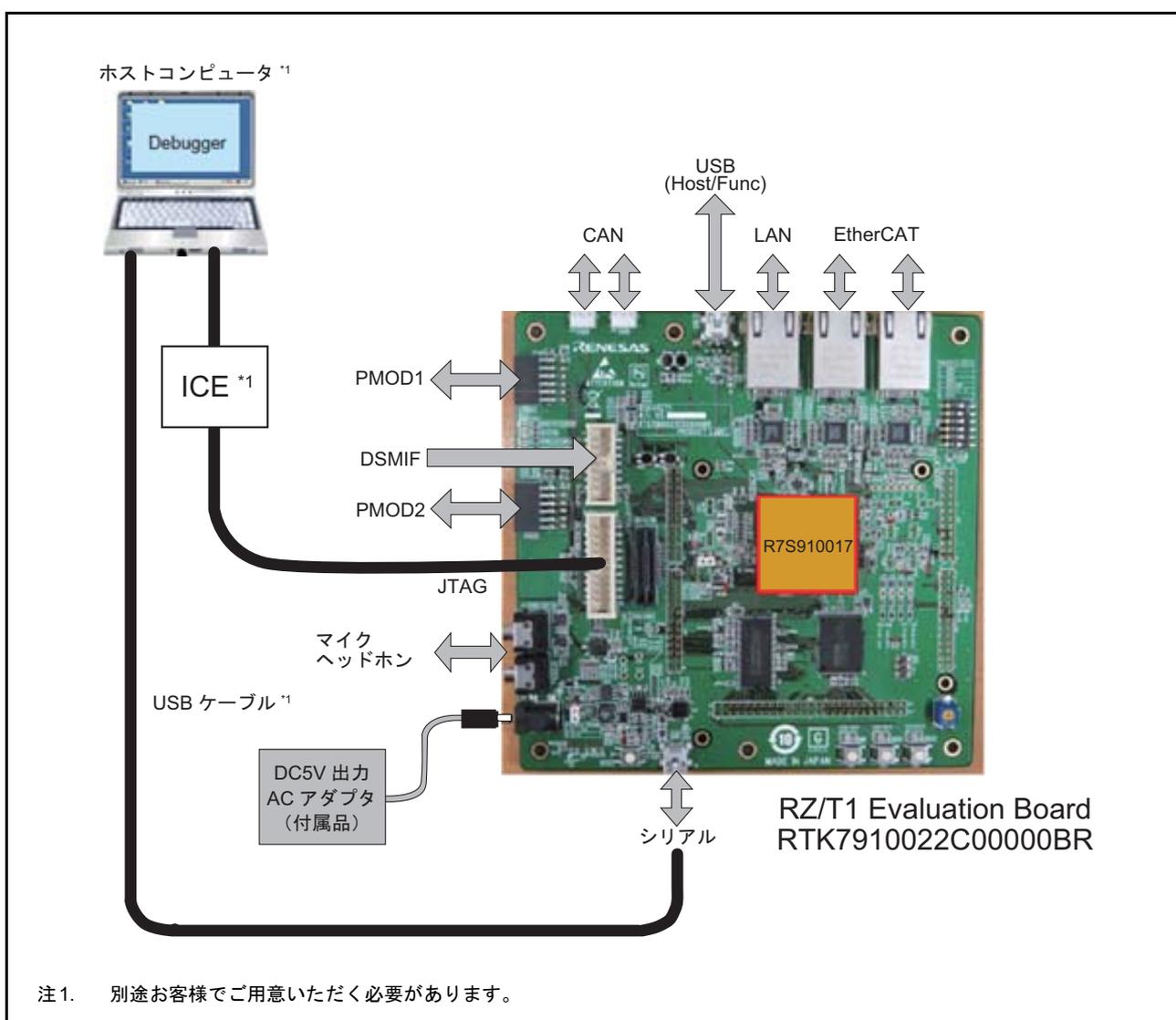


図 1.1 動作環境

2. 動作環境

本アプリケーションノートのサンプルコードは、下記の環境を想定しています。

表2.1 動作環境

項目	内容
使用マイコン	RZ/T1グループ
動作周波数	CPUクロック (CPUCLK) : 450MHz
動作電圧	電源電圧 (I/O) : 3.3V
統合開発環境	IARシステムズ製 Embedded Workbench® for Arm Version 8.20.2 Arm製 DS-5™ 5.26.2 RENESAS製 e2studio 6.1.0
動作モード	SPIブートモード (シリアルフラッシュ) 16ビットバスブートモード (NORフラッシュ)
使用ボード	RZ/T1 Evaluation Board (RTK7910022C00000BR)
使用デバイス (ボード上で使用する機能)	SPIマルチI/Oバス空間 (1または4ビットバス幅) に配置されたシリアルフラッシュメモリ • メーカー名 : Macronix International Co. • 型名 : MX25L51245G

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/T1 グループ初期設定アプリケーションノート (R01AN2554JJ)

注. 本アプリケーションノートで記載しないレジスタに関しては、RZ/T1 グループ 初期設定アプリケーションノートで設定した値のまま使用します。

4. ハードウェア説明

4.1 使用端子一覧

表 4.1 に使用端子と機能を示します。

表 4.1 使用端子と機能

端子名	入出力	内容
SPBCLK	出力	クロック出力
SPBSSL	出力	スレーブセレクト
SPBMO/SPBIO0	入出力	マスタ送出データ/データ0
SPBMI/SPBIO1	入出力	マスタ入力データ/データ1
SPBIO2	入出力	データ2
SPBIO3	入出力	データ3
MD2、 MD1、 MD0	入力	ブートモードの選択 (SPIブートモードに設定) MD_BOOT2 : "L" MD_BOOT1 : "L" MD_BOOT0 : "L"
RES#	入力	システムリセット信号

注. #は負論理 (またはアクティブロー) を示す記号です。

4.2 参考回路

図 4.1 に接続例を示します。

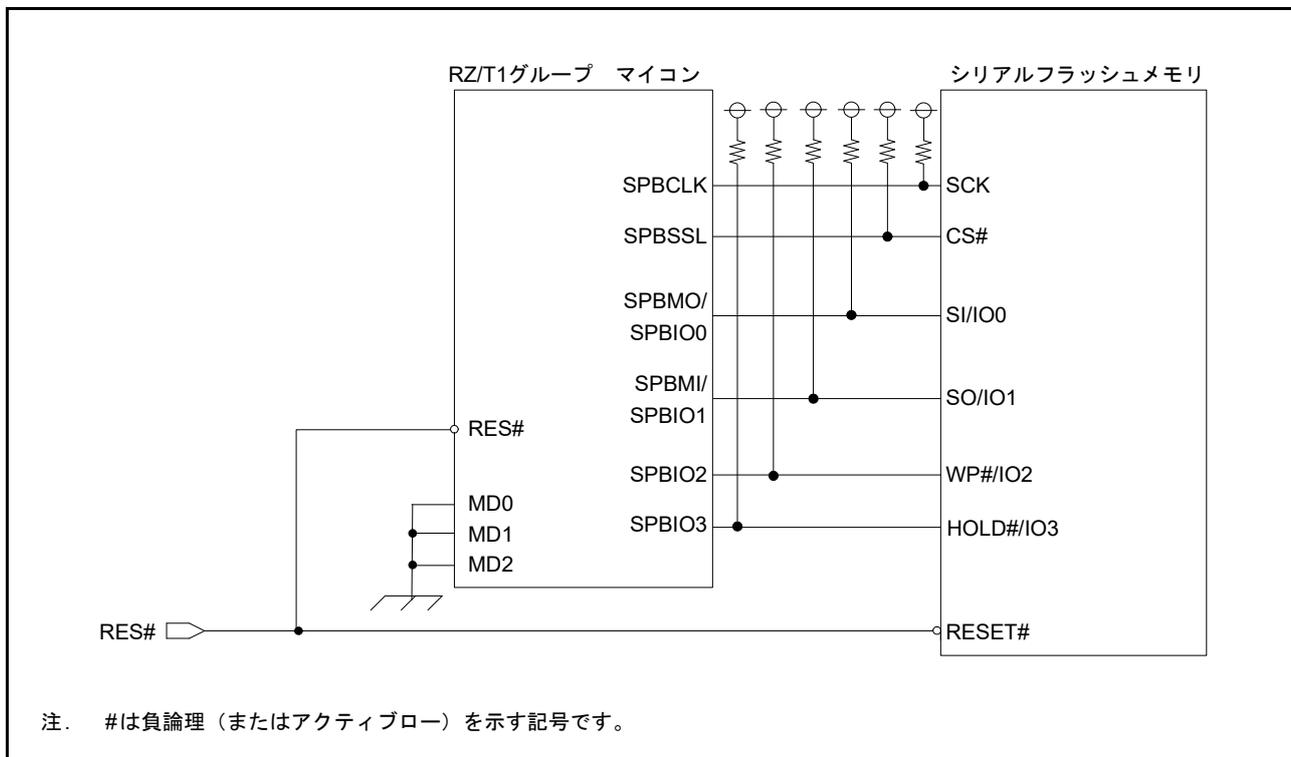


図 4.1 接続例

5. ソフトウェア説明

5.1 動作概要

ここでは、本アプリケーションノートのサンプルプログラムの動作概要を説明します。

RZ/T1 グループマイコンは、SPIブートモードにてブートした場合、SPIBSC空間に接続されたシリアルフラッシュメモリに格納されているローダプログラムをRZ/T1グループマイコンに搭載されている密結合メモリ(BTCM)に転送し、転送されたローダプログラムに分岐します。ローダプログラムに分岐した時点で、SPIBSC空間から直接リードできる状態となっていますが、SPIBSCの設定は、一般的なシリアルフラッシュメモリに共通する設定を行っているため、個々のシリアルフラッシュメモリに合わせてSPIBSCを最適に設定する必要があります。例えば、高速にリードするために、4ビットデータ端子を使用したリードコマンドを設定するなどがが必要です。本サンプルプログラムでは、Macronix社製シリアルフラッシュメモリ(型名:MX25L51245G)に最適なSPIBSC設定を行っています。

5.1.1 プロジェクト設定

開発環境となる EWARM / DS-5 / e2studio 上で使用されるプロジェクト設定については、RZ/T1 グループ 初期設定アプリケーションノートに記載しています。

- 注． 本サンプルプログラムの、EWARM 環境の SPI ブート版 (RZ_T1_serialflash_serial_boot.eww) については、シリアルフラッシュの内容を書き換えた際に、rodata セクションの内容を書き換えないために、rodata セクションを ATCM (密結合メモリ) に配置しています。配置の際に、RZ_T1_init_serial_boot.icf を変更しており、その変更内容を以下に示します (太字は追加文字、二重取り消し線は削除文字)。

```
define block USER_PRG_RBLOCK { ro code, section .rodata_init };
define block USER_PRG_WBLOCK { rw code, section .rodata };
:
initialize manually { ro code object loader_init.o,
                    ro code object loader_init2.o,
                    ro code object r_atcm_init.o,
                    ro code object r_cpg.o,
                    ro code object r_ram_init.o,
                    ro code object r_mpc.o,
                    ro code object bus_init_serial_boot.o,
                    ro code object r_reset.o,
                    ro code object vector.o,
                    ro code object spibsc_flash_api.o,
                    ro code object spibsc_flash_userdef.o,
                    ro code object spibsc_ioaset_api.o,
                    ro code object spibsc_ioaset_drv.o,
                    ro code object spibsc_ioaset_userdef.o,
                    ro code,
                    section .rodata
                    };
:
place in ROM_region { block USER_PRG_RBLOCK,section .rodata };
```

5.1.2 使用準備

本サンプルプログラムでは、PC との通信動作を行いますので、その実行準備を説明します。

- (1) ホスト PC にてターミナルソフトを起動し、シリアルポートの設定を次のように設定します。(Tera Term で COM5 使用の場合)



図 5.1 シリアルポートの設定

- (2) サンプルプログラムを実行し通信可能状態になると、下図に示すようにサンプルプログラムからの受信データがターミナルソフトに表示されます。

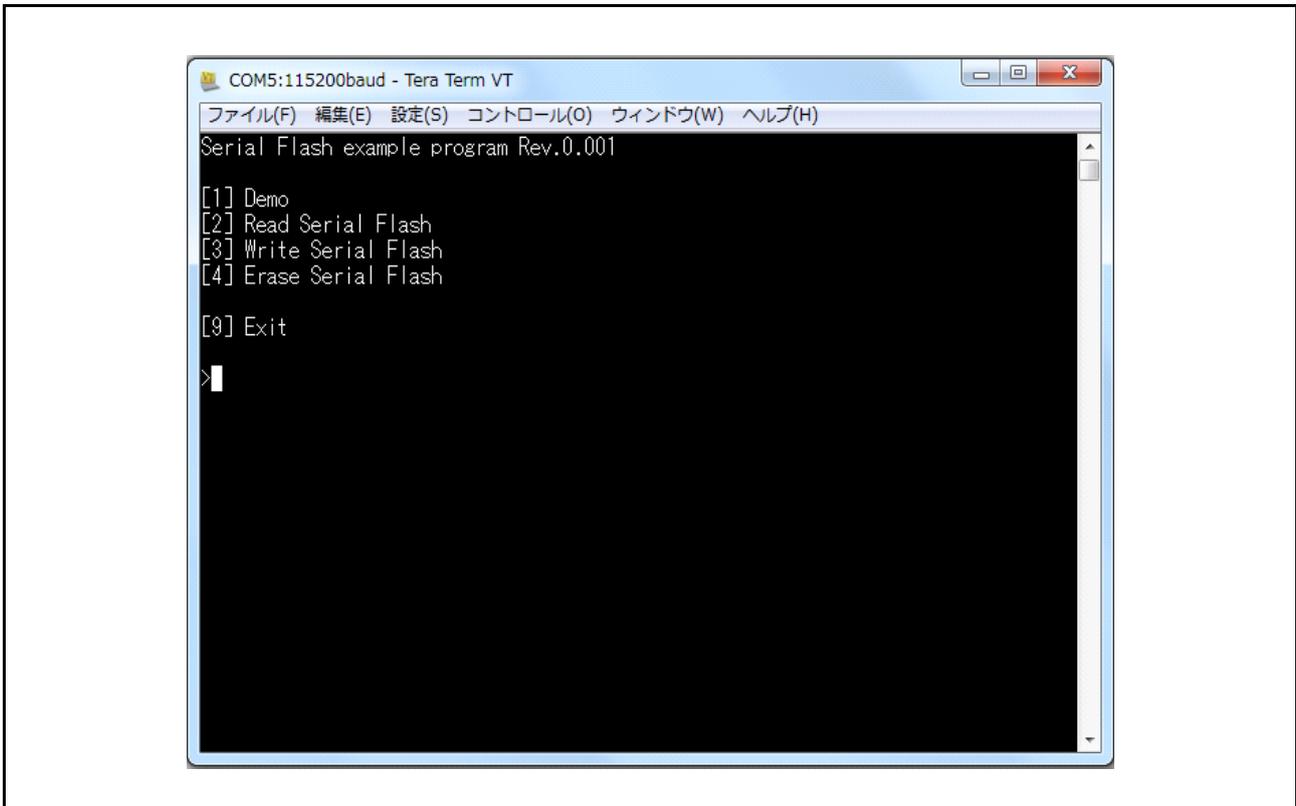


図 5.2 サンプルプログラム実行後のターミナルソフトの表示

5.2 固定幅整数一覧

表 5.1 にサンプルプログラムで使用する固定幅整数を示します。

表 5.1 サンプルプログラムで使用する固定幅整数

シンボル	内容
char_t	8ビット整数、符号あり
int_t	整数、符号あり
int32_t	32ビット整数、符号あり
uint8_t	8ビット整数、符号なし
uint16_t	16ビット整数、符号なし
uint32_t	32ビット整数、符号なし

5.3 構造体 / 共用体一覧

表 5.2 ~ 表 5.8 にサンプルプログラムで使用する構造体を示します。

表 5.2 SPIBSC 外部アドレスリード設定構造体 (st_spibsc_cfg_t) (1)

メンバ名	内容
uint8_t udef_cmd	リードコマンド <ul style="list-style-type: none"> • SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するリードコマンドを設定します。 • 本メンバに設定した値をデータリードコマンド設定レジスタ (DRCMR) のCMD[7:0]に設定します。
uint8_t udef_cmd_width	リードコマンドビット幅 <ul style="list-style-type: none"> • リードコマンド発行時のビット幅を設定します。 • 設定可能な値 : SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 • 本メンバに設定した値をデータリードイネーブル設定レジスタ (DRENr) のCDB[1:0]に設定します。
uint8_t udef_opd3 uint8_t udef_opd2 uint8_t udef_opd1 uint8_t udef_opd0	オプションルデータ <ul style="list-style-type: none"> • SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するオプションルデータを設定します。 • 本メンバに設定した値をデータリードオプション設定レジスタ (DROPR) のOPD3[7:0]、OPD2[7:0]、OPD1[7:0]、OPD0[7:0]に設定します。
uint8_t udef_opd_enable	オプションルデータイネーブル <ul style="list-style-type: none"> • オプションルデータを発行するかどうかを選択します。 • 設定可能な値 : SPIBSC_OUTPUT_DISABLE : 出力しない SPIBSC_OUTPUT_OPD_3 : OPD3を出力 SPIBSC_OUTPUT_OPD_32 : OPD3,OPD2を出力 SPIBSC_OUTPUT_OPD_321 : OPD3,OPD2,OPD1を出力 SPIBSC_OUTPUT_OPD_3210 : OPD3,OPD2,OPD1,OPD0を出力 • 本メンバに設定した値をデータリードイネーブル設定レジスタ (DRENr) のOPDE[3:0]に設定します。
uint8_t udef_opd_width	オプションルデータビット幅 <ul style="list-style-type: none"> • オプションルデータ発行時のビット幅を設定します。 • 設定可能な値 : SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 • 本メンバに設定した値をデータリードイネーブル設定レジスタ (DRENr) のOPDB[1:0]に設定します。

表5.3 SPIBSC 外部アドレスリード設定構造体 (st_spibsc_cfg_t) (2)

メンバ名	内容
uint8_t udef_dmycyc_num	<p>ダミーサイクル数</p> <ul style="list-style-type: none"> SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_DUMMY_1CYC : 1サイクル SPIBSC_DUMMY_2CYC : 2サイクル SPIBSC_DUMMY_3CYC : 3サイクル SPIBSC_DUMMY_4CYC : 4サイクル SPIBSC_DUMMY_5CYC : 5サイクル SPIBSC_DUMMY_6CYC : 6サイクル SPIBSC_DUMMY_7CYC : 7サイクル SPIBSC_DUMMY_8CYC : 8サイクル 本メンバに設定した値をデータリードダミーサイクル設定レジスタ (DRDMCR) のDMCYC[2:0]に設定します。
uint8_t udef_dmycyc_enable	<p>ダミーサイクルイネーブル</p> <ul style="list-style-type: none"> ダミーサイクルを挿入するかを選択します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_DUMMY_CYC_DISABLE : 挿入しない SPIBSC_DUMMY_CYC_ENABLE : 挿入する 本メンバに設定した値をデータリードイネーブル設定レジスタ (DRENr) のDMEに設定します。
uint8_t udef_dmycyc_width	<p>ダミーサイクルビット幅</p> <ul style="list-style-type: none"> ダミーサイクル発行時のビット幅を設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をデータリードダミーサイクル設定レジスタ (DRDMCR) のDMDB[1:0]に設定します。
uint8_t udef_data_width	<p>データリードビット幅</p> <ul style="list-style-type: none"> SPIマルチI/Oバス空間へのリードをSPI通信に変換する時のシリアルフラッシュメモリのデータリードビット幅を設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をデータリードイネーブル設定レジスタ (DRENr) のDRDB[1:0]に設定します。

表5.4 SPIBSC 外部アドレスリード設定構造体 (st_spibsc_cfg_t) (3)

メンバ名	内容
uint8_t udef_spbr	ビットレート <ul style="list-style-type: none"> SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するシリアルクロック (SPBCLK) のビットレートを設定します。 設定可能な値: ビットレート分周設定 (udef_brdv) と合わせて設定を行ってください。 本メンバに設定した値をビットレート設定レジスタ (SPBCR) のSPBR[7:0]に設定します。
uint8_t udef_brdv	ビットレート分周設定 <ul style="list-style-type: none"> SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するシリアルクロック (SPBCLK) のビットレートを設定します。 設定可能な値: ビットレート (udef_spbr) と合わせて設定を行ってください。 本メンバに設定した値をビットレート設定レジスタ (SPBCR) のBRDV[1:0]に設定します。
uint8_t udef_addr_width	アドレスビット幅 <ul style="list-style-type: none"> SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するアドレスのビット幅を設定します。 設定可能な値: SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をデータリードイネーブル設定レジスタ (DRENr) のADB[1:0]に設定します。
uint8_t udef_addr_mode	アドレスイネーブル <ul style="list-style-type: none"> SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 設定可能な値: SPIBSC_OUTPUT_ADDR_24 : 24ビットのアドレスを出力 SPIBSC_OUTPUT_ADDR_32 : 32ビットのアドレスを出力 本メンバに設定した値をデータリードイネーブル設定レジスタ (DRENr) のADE[3:0]に設定します。

表 5.5 SPIBSC SPIモード設定構造体 (st_spibsc_spimd_reg_t) (1)

メンバ名	内容
uint32_t cdb	<p>コマンドビット幅</p> <ul style="list-style-type: none"> SPI動作モード時のコマンドビット幅を指定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のCDB[1:0]に設定します。
uint32_t ocdb	<p>オプションコマンドビット幅</p> <ul style="list-style-type: none"> SPI動作モード時のオプションコマンドビット幅を指定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のOCDB[1:0]に設定します。
uint32_t adb	<p>アドレスビット幅</p> <ul style="list-style-type: none"> SPI動作モード時のアドレスビット幅を指定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のADB[1:0]に設定します。
uint32_t opdb	<p>オプションデータビット幅</p> <ul style="list-style-type: none"> SPI動作モード時のオプションデータビット幅を指定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のOPDB[1:0]に設定します。
uint32_t spidb	<p>転送データビット幅</p> <ul style="list-style-type: none"> SPI動作モード時の転送データビット幅を指定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のSPIDB[1:0]に設定します。
uint32_t cde	<p>SPI動作モード時にコマンドを出力するかを設定します。</p> <ul style="list-style-type: none"> 設定可能な値： <ul style="list-style-type: none"> SPIBSC_OUTPUT_DISABLE : 出力しない SPIBSC_OUTPUT_ENABLE : 出力する 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のCDEに設定します。

表 5.6 SPIBSC SPIモード設定構造体 (st_spibsc_spimd_reg_t) (2)

メンバ名	内容
uint32_t ocde	<p>オプションコマンドイネーブル</p> <ul style="list-style-type: none"> SPI動作モード時にオプションコマンドを出力するかを設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_ENABLE：出力する 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のOCDEに設定します。
uint32_t ade	<p>アドレスイネーブル</p> <ul style="list-style-type: none"> SPI動作モード時にアドレスを出力するかを設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_ADDR_24：ADR[23:0]を出力 SPIBSC_OUTPUT_ADDR_32：ADR[31:0]を出力 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のADE[3:0]に設定します。
uint32_t opde	<p>オプションデータイネーブル</p> <ul style="list-style-type: none"> SPI動作モード時にオプションデータを出力するかを設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_OPD_3：OPD3を出力 SPIBSC_OUTPUT_OPD_32：OPD3,OPD2を出力 SPIBSC_OUTPUT_OPD_321：OPD3,OPD2,OPD1を出力 SPIBSC_OUTPUT_OPD_3210：OPD3,OPD2,OPD1,OPD0を出力 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のOPDE[3:0]に設定します。
uint32_t spide	<p>転送データイネーブル</p> <ul style="list-style-type: none"> SPI動作モード時にデータ転送を行うかを設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_OUTPUT_DISABLE：出力しない SPIBSC_OUTPUT_SPID_8：8 (または16) ビット転送 SPIBSC_OUTPUT_SPID_16：16 (または32) ビット転送 SPIBSC_OUTPUT_SPID_32：32 (または64) ビット転送 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のSPIDE[3:0]に設定します。
uint32_t sslkp	<p>SPBSSL信号レベル保持</p> <ul style="list-style-type: none"> SPI動作モード時に転送終了後のSPBSSL信号状態を設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_SPISSL_NEGATE：転送終了時にネゲート SPIBSC_SPISSL_KEEP：転送終了後から次アクセス開始までSPBSSL信号レベルを保持 本メンバに設定した値をSPIモードコントロールレジスタ (SMCR) のSSLKPに設定します。

表 5.7 Structure of SPIBSC SPI Mode Settings (st_spijsc_spimnd_reg_t) (3)

メンバ名	内容
uint32_t spire	データリードイネーブル <ul style="list-style-type: none"> SPI 動作モード時にデータリードするかを設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_SPIDATA_DISABLE : データリードしない SPIBSC_SPIDATA_ENABLE : データリードする 本メンバに設定した値をSPIモードコントロールレジスタ (SMCR) のSPIREに設定します。
uint32_t spiwe	データライトイネーブル <ul style="list-style-type: none"> SPI 動作モード時にデータライトするかを設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_SPIDATA_DISABLE : データライトしない SPIBSC_SPIDATA_ENABLE : データライトする 本メンバに設定した値をSPIモードコントロールレジスタ (SMCR) のSPIWEに設定します。
uint32_t dme	ダミーサイクルイネーブル <ul style="list-style-type: none"> SPI 動作モード時にダミーサイクル挿入するかどうかを設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_DUMMY_CYC_DISABLE : 挿入しない SPIBSC_DUMMY_CYC_ENABLE : 挿入する 本メンバに設定した値をSPIモードイネーブル設定レジスタ (SMENR) のDMEに設定します。
uint8_t dmdb	ダミーサイクルビット幅 <ul style="list-style-type: none"> SPI 動作モード時のダミーサイクルのビット幅を設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_1BIT : 1ビット幅 SPIBSC_4BIT : 4ビット幅 本メンバに設定した値をSPIモードダミーサイクル設定レジスタ (SMDMCR) のDMDB[1:0]に設定します。
uint8_t dmcyc	ダミーサイクル数 <ul style="list-style-type: none"> SPI 動作モード時のダミーサイクル数を設定します。 設定可能な値： <ul style="list-style-type: none"> SPIBSC_DUMMY_1CYC : 1サイクル SPIBSC_DUMMY_2CYC : 2サイクル SPIBSC_DUMMY_3CYC : 3サイクル SPIBSC_DUMMY_4CYC : 4サイクル SPIBSC_DUMMY_5CYC : 5サイクル SPIBSC_DUMMY_6CYC : 6サイクル SPIBSC_DUMMY_7CYC : 7サイクル SPIBSC_DUMMY_8CYC : 8サイクル 本メンバに設定した値をSPIモードダミーサイクル設定レジスタ (SMDMCR) のDMCYC[2:0]に設定します。

表 5.8 Structure of SPIBSC SPI Mode Settings (st_spibsc_spimd_reg_t) (4)

メンバ名	内容
uint8_t cmd	<p>コマンド</p> <ul style="list-style-type: none"> SPI動作モード時に出力するコマンドを設定します。 本メンバに設定した値をSPIモードコマンド設定レジスタ (SMCMR) のCMD[7:0]に設定します。
uint8_t ocmd	<p>オプションコマンド</p> <ul style="list-style-type: none"> SPI動作モード時に出力するオプションコマンドを設定します。 本メンバに設定した値をSPIモードコマンド設定レジスタ (SMCMR) のOCMD[7:0]に設定します。
uint32_t addr	<p>アドレス</p> <ul style="list-style-type: none"> SPI動作モード時に出力するアドレスを設定します。 本メンバに設定した値をSPIモードアドレス設定レジスタ (SMADR) のADR[23:0]に設定します。 <p>32ビットで出力する場合は、ADRE[7:0]を設定します。 この設定は (SMENR) のADE[3]ビット = 1のときに有効になります。</p>
uint8_t opd[4]	<p>オプションデータ</p> <ul style="list-style-type: none"> SPI動作モード時に出力するオプションデータを設定します。 本メンバに設定した値をSPIモードオプション設定レジスタ (SMOPR) のOPDn[7:0]に以下のように設定します。 <p>OPD3[7:0] ← opd[0] OPD2[7:0] ← opd[1] OPD1[7:0] ← opd[2] OPD0[7:0] ← opd[3]</p>
uint32_t smrdr	<p>リードデータ格納バッファ</p> <ul style="list-style-type: none"> SPI動作モード時にリードしたデータ (SPIモードリードデータレジスタ0 (SMRDR0)) を以下のように格納します。 <p>SMRDR0 → smrdr</p>
uint32_t smwdr	<p>ライトデータ格納バッファ</p> <ul style="list-style-type: none"> SPI動作モード時にライトするデータ (SPIモードライトデータレジスタ0 (SMWDR0)) を以下のように格納します。 <p>SMWDR0 ← swrdr</p>

5.4 定数一覧

表 5.9 ~ 表 5.12 にサンプルプログラムで使用する定数を示します。

表5.9 サンプルプログラムで使用する定数(1)

定数名	設定値	内容
SFLASHCMD_SECTOR_ERASE	(0xD8)	Sector Erase (3-byte Address) コマンド注1
SFLASHCMD_BYTE_PROGRAM	(0x02)	Page Program (3-byte Address) コマンド
SFLASHCMD_FAST_READ	(0x0B)	Read Fast (3-byte Address) コマンド
SFLASHCMD_QUAD_FAST_READ	(0x6B)	Quad Read Fast (3-byte Address) コマンド
SFLASHCMD_QUAD_IO_READ	(0xEB)	Quad I/O Read (3-byte Address) コマンド
SFLASHCMD_WRITE_ENABLE	(0x06)	Write Enable コマンド
SFLASHCMD_READ_STATUS	(0x05)	Read Status Register-1 コマンド
SFLASHCMD_READ_CONFIG	(0x15)	Read Configuration Register-1 コマンド
SFLASHCMD_WRITE_STATUS	(0x01)	Write Register (Status-1, Configuration-1) コマンド
SFLASHCMD_SECTOR_ERASE_4B	(0xDC)	Sector Erase (4-byte Address) コマンド注1
SFLASHCMD_BYTE_PROGRAM_4B	(0x12)	Page Program (4-byte Address) コマンド
SFLASHCMD_FAST_READ_4B	(0x0C)	Read Fast (4-byte Address) コマンド
SFLASHCMD_QUAD_FAST_READ_4B	(0x6C)	Quad Read Fast (4-byte Address) コマンド
SFLASHCMD_QUAD_IO_READ_4B	(0xEC)	Quad I/O Read (4-byte Address) コマンド

注1. 本サンプルプログラムが対象としているMacronix社製シリアルフラッシュメモリ（型名：MX25L51245G）では64kBの削除を行います。

表5.10 サンプルプログラムで使用する定数(2)

定数名	設定値	内容
STREG_SRWD_BIT	(0x80)	Status Register/SRWD ビットマスク値
STREG_QUAD_BIT	(0x40)	Status Register/QUAD ビットマスク値
STREG_BPROTECT_BIT	(0x3C)	Status Register/Block Protection ビットマスク値
STREG_WEL_BIT	(0x02)	Status Register/Write Enable Latch ビットマスク値
STREG_WIP_BIT	(0x01)	Status Register/Write in Progress ビットマスク値
CFREG_LC_BIT	(0xC0)	Configuration Register/Latency Code ビットマスク値
CFREG_4BYTE_BIT	(0x20)	Configuration Register/4BYTE ビットマスク値

表5.11 サンプルプログラムで使用する定数(3)

定数名	設定値	内容
SPIBSC_CMNCR_BSZ_SINGLE	(0)	SPIBSCデータバスに接続するシリアルフラッシュメモリは1つ
SPIBSC_CMNCR_MD_EXTRD	(0)	SPIBSCのモードを外部アドレス空間リードモードに設定
SPIBSC_CMNCR_MD_SPI	(1)	SPIBSCのモードをSPI動作モードに設定
SPIBSC_OUTPUT_LOW	(0)	端子の出力値を0に設定
SPIBSC_OUTPUT_HIGH	(1)	端子の出力値を1に設定
SPIBSC_OUTPUT_LAST	(2)	端子の出力値を前回転送の最終ビットに設定
SPIBSC_OUTPUT_HI_Z	(3)	端子の出力値をHI-Zに設定
SPIBSC_CMNCR_CPHAT_EVEN	(0)	受信データに対するSPBCLK信号のエッジを奇数エッジに設定
SPIBSC_CMNCR_CPHAT_ODD	(1)	受信データに対するSPBCLK信号のエッジを偶数エッジに設定
SPIBSC_CMNCR_CPHAR_ODD	(0)	受信データに対するSPBCLK信号のエッジを偶数エッジに設定
SPIBSC_CMNCR_CPHAR_EVEN	(1)	受信データに対するSPBCLK信号のエッジを奇数エッジに設定
SPIBSC_CMNCR_SSLP_LOW	(0)	SPBSSL信号の極性をLowアクティブに設定
SPIBSC_CMNCR_SSLP_HIGH	(1)	SPBSSL信号の極性をHighアクティブに設定
SPIBSC_CMNCR_CPOL_LOW	(0)	SPBSSL信号が印アクティブ期間のSPBCLK端子の出力レベルを0に設定
SPIBSC_CMNCR_CPOL_HIGH	(1)	SPBSSL信号が印アクティブ期間のSPBCLK端子の出力レベルを1に設定
SPIBSC_DELAY_1SPBCLK	(0)	SSL遅延レジスタの各シンボルにB'000を設定
SPIBSC_DELAY_2SPBCLK	(1)	SSL遅延レジスタの各シンボルにB'001を設定
SPIBSC_DELAY_3SPBCLK	(2)	SSL遅延レジスタの各シンボルにB'010を設定
SPIBSC_DELAY_4SPBCLK	(3)	SSL遅延レジスタの各シンボルにB'011を設定
SPIBSC_DELAY_5SPBCLK	(4)	SSL遅延レジスタの各シンボルにB'100を設定
SPIBSC_DELAY_6SPBCLK	(5)	SSL遅延レジスタの各シンボルにB'101を設定
SPIBSC_DELAY_7SPBCLK	(6)	SSL遅延レジスタの各シンボルにB'110を設定
SPIBSC_DELAY_8SPBCLK	(7)	SSL遅延レジスタの各シンボルにB'111を設定
SPIBSC_BURST_1 ~ SPIBSC_BURST_16	(0x00) ~ (0x0f)	リード時のバースト長： 1~16データ長ビット
SPIBSC_BURST_DISABLE	(0)	リードキャッシュを無効に設定
SPIBSC_BURST_ENABLE	(1)	リードキャッシュを有効に設定
SPIBSC_DRCCR_RCF_EXE	(1)	リードキャッシュの全エントリをクリア
SPIBSC_SSL_NEGATE	(0)	SPBSSL端子はインアクティブ状態
SPIBSC_TRANS_END	(1)	データ転送が終了したことを示す
SPIBSC_1BIT	(0)	リードコマンド発行時のビット幅を1ビットに設定
SPIBSC_4BIT	(2)	リードコマンド発行時のビット幅を4ビットに設定
SPIBSC_OUTPUT_DISABLE	(0)	リードコマンド発行時のコマンドを出力しない設定
SPIBSC_OUTPUT_ENABLE	(1)	リードコマンド発行時のコマンドを出力する設定
SPIBSC_OUTPUT_ADDR_24	(0x07)	24ビットのアドレスを出力
SPIBSC_OUTPUT_ADDR_32	(0x0f)	32ビットのアドレスを出力
SPIBSC_OUTPUT_OPD_3	(0x08)	リードコマンド発行時のオプションルデータイネーブルOPD3を出力
SPIBSC_OUTPUT_OPD_32	(0x0c)	リードコマンド発行時のオプションルデータイネーブルOPD3,OPD2を出力
SPIBSC_OUTPUT_OPD_321	(0x0e)	リードコマンド発行時のオプションルデータイネーブルOPD3,OPD2,OPD1を出力
SPIBSC_OUTPUT_OPD_3210	(0x0f)	リードコマンド発行時のオプションルデータイネーブルOPD3,OPD2,OPD1,OPD0を出力
SPIBSC_OUTPUT_SPID_8	(0x08)	SPI動作モード時に転送データイネーブルを8(または16)ビット転送に設定
SPIBSC_OUTPUT_SPID_16	(0x0c)	SPI動作モード時に転送データイネーブルを16(または32)ビット転送に設定

表5.11 サンプルプログラムで使用する定数(3)

定数名	設定値	内容
SPIBSC_OUTPUT_SPID_32	(0x0f)	SPI動作モード時に転送データタイナブルを32(または64)ビット転送に設定
SPIBSC_SPISSL_NEGATE	(0)	SPI動作モード時に転送終了後のSPBSSL信号状態をネゲートに設定
SPIBSC_SPISSL_KEEP	(1)	SPI動作モード時に転送終了後から次アクセス開始までSPBSSL信号レベルを保持する設定
SPIBSC_SPIDATA_DISABLE	(0)	SPI動作モード時にデータリードしない設定
SPIBSC_SPIDATA_ENABLE	(1)	SPI動作モード時にデータリードする設定
SPIBSC_SPI_ENABLE	(1)	SPIデータ転送を開始する
SPIBSC_DUMMY_CYC_DISABLE	(0)	ダミーサイクルを挿入しない設定
SPIBSC_DUMMY_CYC_ENABLE	(1)	ダミーサイクルを挿入する設定
SPIBSC_DUMMY_1CYC	(0)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を1に設定
SPIBSC_DUMMY_2CYC	(1)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を2に設定
SPIBSC_DUMMY_3CYC	(2)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を3に設定
SPIBSC_DUMMY_4CYC	(3)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を4に設定
SPIBSC_DUMMY_5CYC	(4)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を5に設定
SPIBSC_DUMMY_6CYC	(5)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を6に設定
SPIBSC_DUMMY_7CYC	(6)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を7に設定
SPIBSC_DUMMY_8CYC	(7)	SPIマルチI/Oバス空間へのリードをSPI通信に変換する時にシリアルフラッシュメモリに出力するダミーサイクル数を8に設定

表5.12 サンプルプログラムで使用する定数(4)

定数名	設定値	内容
R_SERIAL_FLASH_TOP	(0x10000000U)	実装されているシリアルフラッシュのトップアドレス
R_SERIAL_FLASH_END	(0x13FFFFFFCU)	実装されているシリアルフラッシュのエンドアドレス
R_SERIAL_FLASH_ADDR_LIMIT	0x14000000U	実装されているシリアルフラッシュのアドレスの境界値
R_SERIAL_FLASH_READ_SIZE	(64U)	読み出しサイズの最大値
R_SERIAL_FLASH_CNT_ADDR	(0x13FF0000U)	カウンタ変数領域
R_SERIAL_FLASH_DATA_WIDTH	SPIBSC_4BIT (2)	データリードビット幅
R_SERIAL_FLASH_ADDR_MODE	SPIBSC_OUTPUT_ADDR_32 (0x0f)	アドレスバイト数
R_SERIAL_FLASH_WRITE_BYTE	(4)	書き込みバイトサイズ
R_SERIAL_FLASH_READ_BYTE	(4)	読み出しサイズ

5.5 変数一覧

表 5.13 に static 型変数を、表 5.14 にグローバル変数を示します。

表 5.13 static型変数

型	変数名	内容
static char	gbuffer[16]	ターミナルソフトからの受信データ
st_spibsc_cfg_t	spibsc_cfg	SPIBSC 外部アドレスリード設定内容格納変数 SPIBSC 外部アドレスリード設定内容を格納します。

表 5.14 グローバル変数

型	変数名	内容
st_spibsc_spimd_reg_t	g_spibsc_spimd_reg	SPIBSC SPIモード動作設定内容格納変数 <ul style="list-style-type: none"> SPIBSC SPIモードで使用する場合に、SPIBSC設定内容を格納します。 サンプルプログラムでは、API関数およびユーザ定義関数内でシリアルフラッシュメモリ制御関数を実行する際の引数として共用で使用しています。

5.6 関数一覧

サンプルコードは、周辺機能を使用するためのインターフェース関数（API 関数）、ユーザシステムの用途に合わせてユーザで準備が必要なユーザ定義関数（API 関数からコールされる関数）、サンプルコードを動作させるために必要なサンプル関数から構成されています。

表 5.15 ～表 5.17 に関数一覧を示します。

表 5.15 List of Functions (1)

関数名	説明
spibsc_init	SPIBSC初期設定関数 本関数内で、Macronix社製シリアルフラッシュメモリ（型名：MX25L51245G）に最適なSPIBSC設定を行います。
serial_flash_demo	デモプログラムを実行する関数
serial_flash_read	シリアルフラッシュ読み出しプログラムを実行する関数
serial_flash_write	シリアルフラッシュ書き込みプログラムを実行する関数
serial_flash_erase	シリアルフラッシュ消去プログラムを実行する関数
main	サンプルプログラムのメイン関数

表 5.16 関数一覧(2)

関数名	説明
R_SPIBSC_ExmodeSetting	SPIBSC初期設定関数 SPIBSCにてシリアルフラッシュメモリを制御するために必要な初期設定およびSPIBSCを外部アドレスリードモードで使用するために必要な初期設定を行います。また初期設定に合わせて、シリアルフラッシュメモリ内のレジスタ設定を行います。初期設定後、外部アドレスリードモードに設定します。
R_SPIBSC_WaitTend	SPIBSCデータ転送終了待ち関数 SPIBSCより、データ転送が終了するのを待ちます。
R_SPIBSC_Exmode	SPIBSC外部アドレスモード設定関数 SPIBSCを外部アドレスリードモードに設定します。
R_SPIBSC_SetConfig	SPIBSC外部アドレスリード設定関数 SPIBSCを外部アドレスリードモードで使用するために必要な初期設定を行います。
R_SPIBSC_Spimode	SPIBSC SPIモード設定関数 SPIBSCをSPIモードに設定します。
R_SPIBSC_Exmodelnit	SPIBSC外部アドレスモード初期設定関数 SPIBSCを外部アドレスリードモードで使用するために必要な初期設定を行います。初期設定後、外部アドレスリードモードに設定します。
R_SPIBSC_EraseSector	シリアルフラッシュメモリ消去関数 SPIBSCのSPIモードを使用して、シリアルフラッシュメモリの消去を行います。
R_SPIBSC_ByteProgram	シリアルフラッシュメモリ書き込み関数 SPIBSCのSPIモードを使用して、シリアルフラッシュメモリにデータを書き込みます。
R_SPIBSC_ByteRead	シリアルフラッシュメモリ読み出し関数 SPIBSCのSPIモードを使用して、シリアルフラッシュメモリのデータを読み出します。
R_SPIBSC_SpibscTransfer	シリアルフラッシュメモリ制御関数 引数にしたがい、シリアルフラッシュメモリにコマンドを発行します。

表 5.17 関数一覧(3)

関数名	説明
userdef_spibsc_set_config	SPIBSC外部アドレスリード設定関数 使用するシリアルフラッシュメモリに応じて、SPIBSC外部アドレスリードモードの設定する内容を決定します。サンプルプログラムは、本関数で設定された内容を基にSPIBSCを外部アドレスリードモードで使用するために必要な初期設定を行います。 サンプルプログラムでは、Macronix社製シリアルフラッシュメモリ（型名：MX25L51245G）を使用する場合のSPIBSC初期設定となっています。
userdef_sflash_set_mode	シリアルフラッシュメモリ内レジスタ設定関数 使用するシリアルフラッシュメモリに応じて、SPIBSCを外部アドレスリードモードで使用する場合に必要なシリアルフラッシュメモリ内レジスタの設定を行います。 サンプルプログラムでは、Macronix社製シリアルフラッシュメモリ（型名：MX25L51245G）へのレジスタ設定を行っています。
userdef_sflash_write_enable	シリアルフラッシュメモリライト許可関数 使用するシリアルフラッシュメモリに応じて、シリアルフラッシュメモリ内レジスタを設定し、ライト許可に設定します。 サンプルプログラムでは、Macronix社製シリアルフラッシュメモリ（型名：MX25L51245G）へのレジスタ設定を行っています。
userdef_sflash_busy_wait	シリアルフラッシュメモリレディー待ち関数 使用するシリアルフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを読み出し、シリアルフラッシュメモリがレディー状態に遷移するのを待ちます。 サンプルプログラムでは、Macronix社製シリアルフラッシュメモリ（型名：MX25L51245G）のレディー待ちを行っています。

5.7 関数詳細

以下に関数詳細を示します。

spibsc_init	
概要	SPIBSC 初期設定関数
宣言	static void spibsc_init(void);
説明	本関数内で、Macronix 社製シリアルフラッシュメモリ（型名：MX25L51245G）に最適な SPIBSC 設定を行います。
引数	なし
リターン値	なし
注意事項	本関数は、16 ビットバスブートモードでブートした際に使用されます。 SPI ブートモードでブートした際は、ブート時にバス接続設定内で SPIBSC の初期設定がされるため、本関数は使用していません。詳細は RZ/T1 グループ初期設定アプリケーションノート（R01AN2554JJ）をご確認ください。

serial_flash_demo	
概要	デモプログラムを実行する関数
宣言	static void serial_flash_demo (void)
説明	入力されたコマンドにより、デモプログラムを実行します。
引数	なし
リターン値	なし

serial_flash_read	
概要	シリアルフラッシュ読み出しプログラムを実行する関数
宣言	static void serial_flash_read (void)
説明	入力されたコマンドにより、シリアルフラッシュ読み出しプログラムを実行します。
引数	なし
リターン値	なし

serial_flash_write	
概要	シリアルフラッシュ書き込みプログラムを実行する関数
宣言	static void serial_flash_write (void)
説明	入力されたコマンドにより、シリアルフラッシュ書き込みプログラムを実行します。
引数	なし
リターン値	なし

serial_flash_erase

概要	シリアルフラッシュ消去プログラムを実行する関数
宣言	static void serial_flash_erase (void)
説明	入力されたコマンドにより、シリアルフラッシュ消去プログラムを実行します。
引数	なし
リターン値	なし

main

概要	サンプルプログラムのメイン関数
宣言	void main (void)
説明	サンプルプログラムのメイン処理です。
引数	なし
リターン値	なし

R_SPIBSC_ExmodeSetting

概要	SPIBSC 初期設定関数
宣言	int32_t R_SPIBSC_ExmodeSetting (st_spibsc_cfg_t *spibsccfg);
説明	SPIBSC にてシリアルフラッシュメモリを制御するために必要な初期設定および SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。また初期設定に合わせて、シリアルフラッシュメモリ内のレジスタ設定を行います。初期設定後、外部アドレスリードモードに設定します。本関数内で SPIBSC 外部アドレスモード初期設定関数 (R_SPIBSC_ExmodeInit) を実行します。
引数	st_spibsc_cfg_t *spibsccfg SPIBSC 外部アドレスリード設定 設定内容は表 5.2 ~ 表 5.4 を参照してください。
リターン値	0 : 正常終了 -1: エラー

R_SPIBSC_SetConfig

概要	SPIBSC 外部アドレスリード設定関数
宣言	int32_t R_SPIBSC_SetConfig (st_spibsc_cfg_t *spibsccfg);
説明	使用するシリアルフラッシュメモリに応じて、SPIBSC を外部アドレスリードモードで使用するための設定内容を決定します。本関数内で、ユーザ定義関数 (SPIBSC 外部アドレスリード設定関数 : userdef_spibsc_set_config) を実行します。
引数	st_spibsc_cfg_t *spibsccfg SPIBSC 外部アドレスリード設定 設定内容は表 5.2 ~ 表 5.4 を参照してください。
リターン値	0 : 正常終了 -1: エラー

R_SPIBSC_WaitTend

概要	SPIBSC データ転送終了待ち関数
宣言	void R_SPIBSC_WaitTend(void);
説明	SPIBSC より、データ転送が終了するのを待ちます。
引数	なし
リターン値	なし

R_SPIBSC_Exmode

概要	SPIBSC 外部アドレスモード設定関数
宣言	int32_t R_SPIBSC_Exmode(void);
説明	SPIBSC を外部アドレスリードモードに設定します。
引数	なし
リターン値	0 : 設定成功

R_SPIBSC_Spimode

概要	SPIBSC SPI モード設定関数
宣言	int32_t R_SPIBSC_Spimode(void);
説明	SPIBSC を SPI モードに設定します。
引数	なし
リターン値	0 : 設定成功

R_SPIBSC_Exmodelnit

概要	SPIBSC 外部アドレスモード初期設定関数
宣言	int32_t R_SPIBSC_Exmodelnit(st_spibsc_cfg_t *spibsccfg)
説明	SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。 初期設定後、外部アドレスリードモードに設定します。
引数	st_spibsc_cfg_t *spibsccfg SPIBSC 外部アドレスリード設定 設定内容は表 5.2 ~ 表 5.4 を参照してください。
リターン値	0 : 正常終了 -1: エラー

R_SPIBSC_EraseSector

概要	シリアルフラッシュメモリ消去関数	
宣言	int32_t R_SPIBSC_EraseSector(uint32_t addr, uint32_t data_width, uint32_t addr_mode);	
説明	SPIBSC の SPI モードを使用して、シリアルフラッシュメモリの消去を行います。	
引数	uint32_t addr	消去するシリアルフラッシュメモリのアドレス
	uint32_t data_width	データリードビット幅 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT : 1 ビット幅 SPIBSC_4BIT : 4 ビット幅
	uint32_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24 : 24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32 : 32 ビットアドレス出力
	リターン値	0 : 設定成功 -1 : 設定失敗

R_SPIBSC_ByteProgram

概要	シリアルフラッシュメモリ書き込み関数	
宣言	int32_t R_SPIBSC_ByteProgram(uint32_t addr, uint8_t *buf, int32_t size, uint32_t data_width, uint32_t addr_mode);	
説明	SPIBSC の SPI モードを使用して、シリアルフラッシュメモリにデータを書き込みます。	
引数	uint32_t addr	書き込むシリアルフラッシュメモリのアドレス 4 バイト境界に整列されたアドレスを指定する必要がある。
	uint8_t *buf	書き込みデータ格納バッファ
	int32_t size	書き込みサイズ (バイト単位) サイズは 4 の倍数を指定する必要がある。
	uint32_t data_width	データリードビット幅 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT : 1 ビット幅 SPIBSC_4BIT : 4 ビット幅
uint32_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24 : 24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32 : 32 ビットアドレス出力	
リターン値	0 : 設定成功 -1 : 設定失敗	

R_SPIBSC_ByteRead

概要	シリアルフラッシュメモリ読み出し関数	
宣言	int32_t R_SPIBSC_ByteRead (uint32_t addr, uint8_t *buf, int32_t size, uint32_t data_width, uint32_t addr_mode);	
説明	SPIBSC の SPI モードを使用して、シリアルフラッシュメモリにデータを読み出します。	
引数	uint32_t addr	書き込むシリアルフラッシュメモリのアドレス データビット幅が4ビット幅の場合は4バイトアドレスに整列されたアドレスを指定する必要がある。
	uint8_t *buf	書き込みデータ格納バッファ
	int32_t size	書き込みサイズ (バイト単位)
	uint32_t data_width	データリードビット幅 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT : 1 ビット幅 SPIBSC_4BIT : 4 ビット幅
	uint32_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24 : 24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32 : 32 ビットアドレス出力
リターン値	0 : 設定成功 -1 : 設定失敗	

R_SPIBSC_SpibscTransfer

概要	シリアルフラッシュメモリ制御関数	
宣言	int32_t R_SPIBSC_SpibscTransfer(st_spibsc_spimd_reg_t *regset);	
説明	SPIBSC の SPI モードを使用して、シリアルフラッシュメモリにアクセスします。	
引数	st_spibsc_spimd_reg_t *	SPIBSC SPI モード設定 設定内容は表 5.5 ~ 表 5.8 を参照してください。
	regset	
リターン値	0 : 設定成功 -1 : 設定失敗	

userdef_spibsc_set_config

概要	SPIBSC 外部アドレスリード設定関数
宣言	void userdef_spibsc_set_config (st_spibsc_cfg_t *spibsccfg);
説明	使用するシリアルフラッシュメモリに応じて、SPIBSC 外部アドレスリードモードの設定する内容を決定します。本関数にて、引数 spibsccfg にて指定された領域に SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行ってください。 引数 spibsccfg に設定する内容については、表 5.2 ~ 表 5.4 を参照してください。
引数	st_spibsc_cfg_t *spibsccfg SPIBSC 外部アドレスリード設定 設定内容は表 5.2 ~ 表 5.4 を参照してください。
リターン値	なし
注意	サンプルプログラムでは、Macronix 社製シリアルフラッシュメモリ（型名：MX25L51245G）を使用する場合の SPIBSC 初期設定となっています。

userdef_sflash_set_mode

概要	シリアルフラッシュメモリ内レジスタ設定関数
宣言	int32_t userdef_sflash_set_mode (uint32_t data_width, uint32_t addr_mode);
説明	本関数内で使用するシリアルフラッシュメモリに応じて、SPIBSC を外部アドレスリードモードで使用する場合に必要なシリアルフラッシュメモリ内レジスタを設定する処理を実装してください。
引数	uint32_t data_width データリードビット幅 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT : 1 ビット幅 SPIBSC_4BIT : 4 ビット幅 uint32_t addr_mode アドレスモード設定 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24 : 24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32 : 32 ビットアドレス出力
リターン値	0 : 設定成功 -1 : 設定失敗
注意	サンプルプログラムでは、Macronix 社製シリアルフラッシュメモリ（型名：MX25L51245G）を使用する場合の SPIBSC 初期設定となっています。

userdef_sflash_write_enable

概要	シリアルフラッシュメモリライト許可関数
宣言	int32_t userdef_spibsc_write_enable (void);
説明	本関数内で使用するシリアルフラッシュメモリに応じて、シリアルフラッシュメモリ内レジスタを設定し、ライト許可に設定する処理を実装してください。
引数	なし
リターン値	0 : 設定成功 -1 : 設定失敗
注意	サンプルプログラムでは、Macronix 社製シリアルフラッシュメモリ（型名 : MX25L51245G）を使用する場合の SPIBSC 初期設定となっています。

userdef_sflash_busy_wait

概要	シリアルフラッシュメモリレディー待ち関数
宣言	int32_t userdef_spibsc_busy_wait (uint32_t data_width);
説明	本関数内で使用するシリアルフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを読み出し、シリアルフラッシュメモリがレディー状態に遷移する処理を実装してください。
引数	uint32_t data_width データリードビット幅 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時の シリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT : 1 ビット幅 SPIBSC_4BIT : 4 ビット幅
リターン値	なし
注意	サンプルプログラムでは、Macronix 社製シリアルフラッシュメモリ（型名 : MX25L51245G）を使用する場合の SPIBSC 初期設定となっています。

5.8 サンプルプログラムの関数フロー

5.8.1 SPIBSC 初期設定関数

図 5.3 に SPIBSC 初期設定関数のフローを示します。

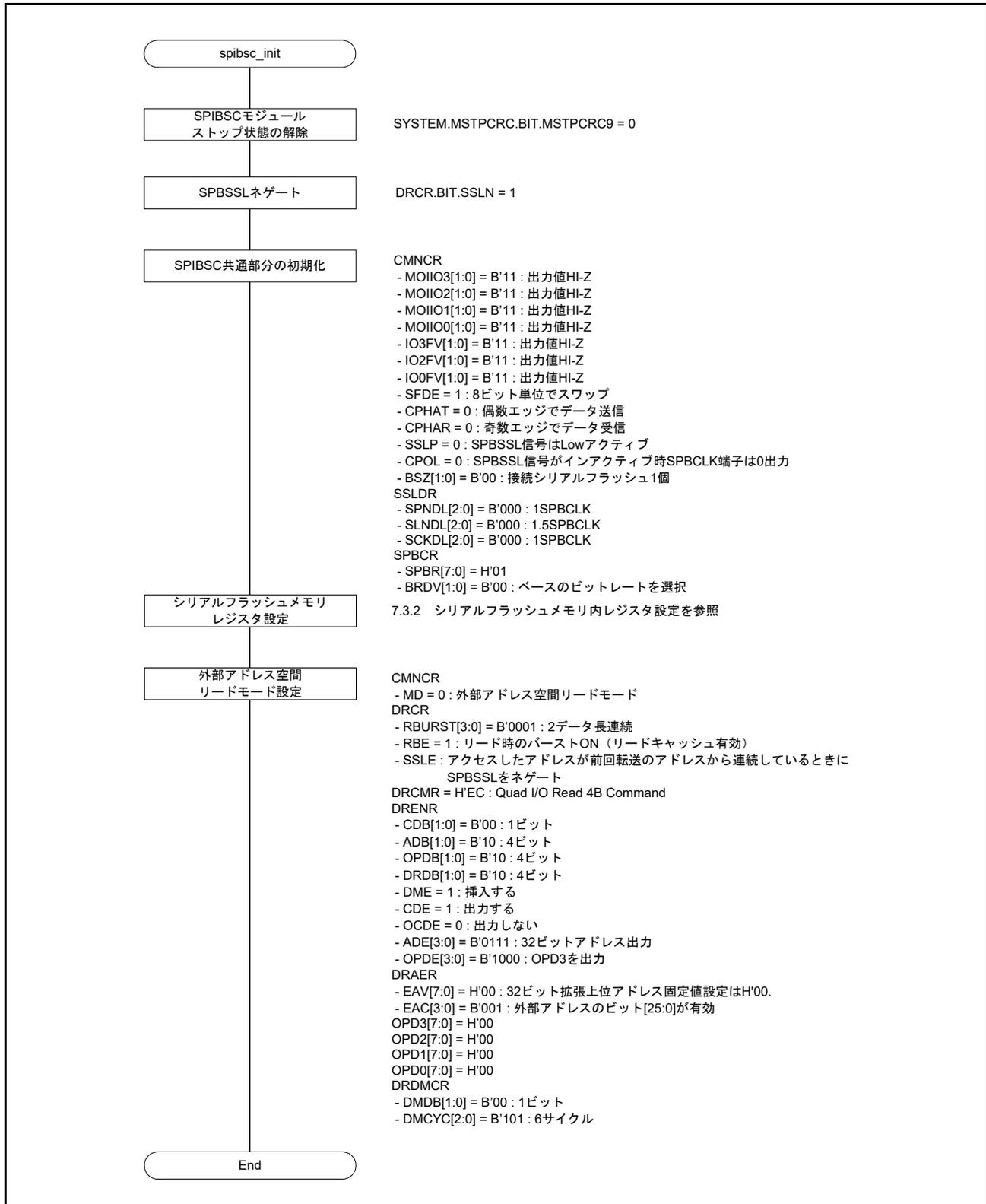


図 5.3 SPIBSC 初期設定関数のフロー

5.8.2 メイン関数

図 5.4 にメイン関数のフローを示します。

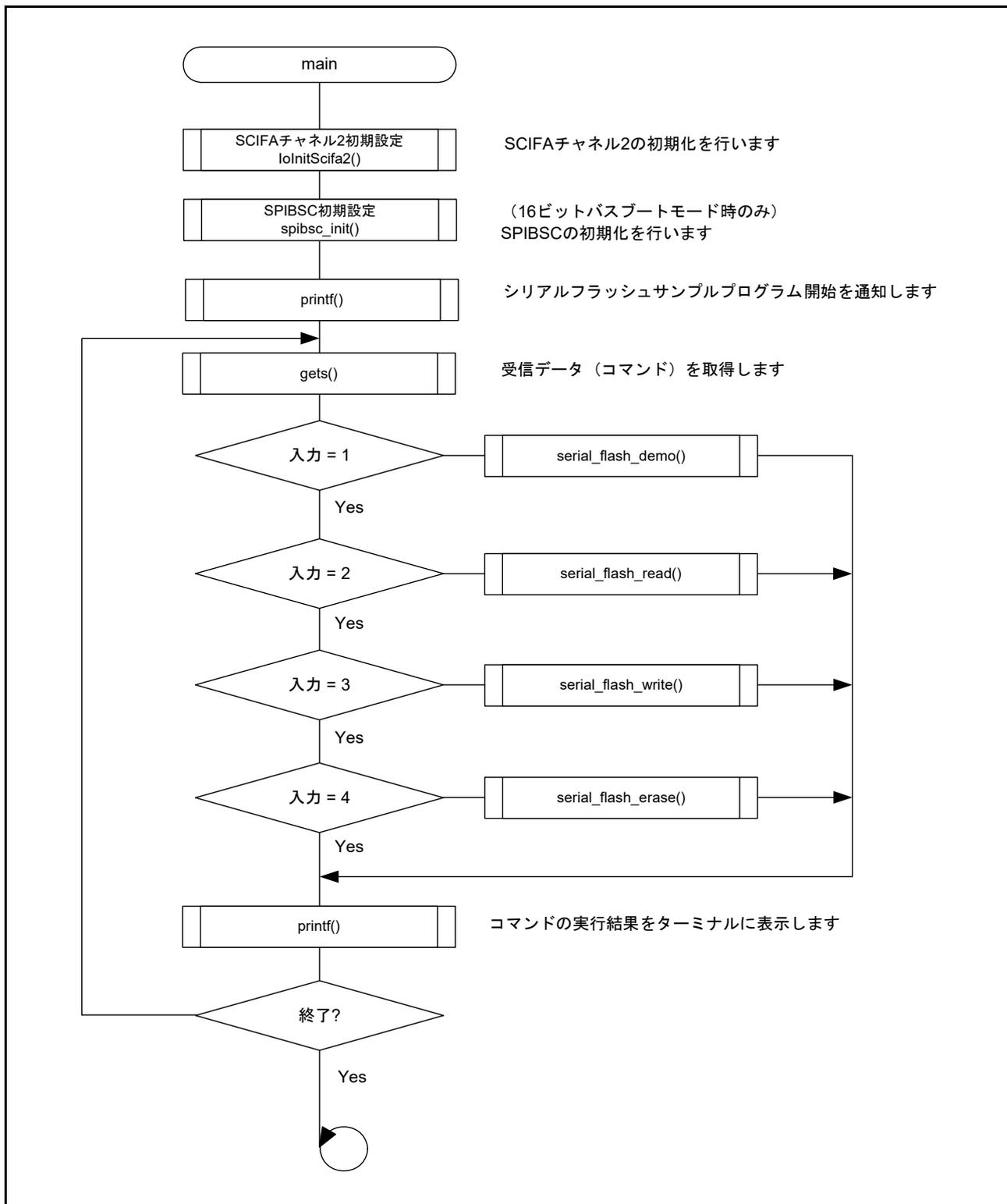


図 5.4 メイン関数のフロー

5.9 サンプルプログラム動作説明

PCのターミナルソフトウェア上でのサンプルプログラムの動作を示します。

サンプルプログラムにおけるユーザ入力値は、数字0～9または文字A～F以外を入力した場合は0になりますのでご注意ください。

RZ/T1 Evaluation Boardの電源ONまたはリセットで以下のメニューが表示されます。

```
Serial Flash example program
```

```
[1] Demo  
[2] Read Serial Flash  
[3] Write Serial Flash  
[4] Erase Serial Flash
```

```
[9] Exit
```

```
>
```

5.9.1 デモプログラム

SPIBSCのモードを切り替え、シリアルフラッシュの読み出し/書き込みを行います。

メニューにて[1]を選択すると、まず外部アドレス空間リードモードでシリアルフラッシュ末尾に設定したカウンタ変数を読み出して表示します。

```
>1[Enter]
```

```
OutAddrMode 13FF0000 FFFFFFFF
```

本サンプルプログラムでは、カウンタ変数のアドレスを0x13FF0000に設定しています。

SPI動作モードに切り替え、カウンタ変数の値を再度読み出して表示します。

このとき、両モードで読み出したカウンタ変数の値が同じであることを確認します。

```
>1[Enter]
```

```
OutAddrMode 13FF0000 FFFFFFFF  
SPIMode      13FF0000 FFFFFFFF
```

SPI動作モードでの読み出し後、カウンタ変数に加算(+1)した値が書き込まれます。

再度ブートして[1]を選択すると、両モードともにカウンタ変数の値が加算されていることが確認できます。

```
>1[Enter]
```

```
OutAddrMode 13FF0000 00000000  
SPIMode      13FF0000 00000000
```

5.9.2 シリアルフラッシュ読み出し

指定したアドレスのデータを SPI 動作モードで読み出します。読み出しサイズが選択可能です。

メニューにて [2] を選択し、アドレスとサイズをそれぞれ入力します。

ここではアドレス : 11000000、サイズ : 20 を入力しています。

```
>2[Enter]
Input the top address (10000000 - 13FFFFFFC)
>11000000[Enter]
>Input the number of blocks (1 - 64 : 4byte blocks)
>20[Enter]
>
```

以下のように 20 ロングワードのデータが表示されます。

```
>20[Enter]
      00      04      08      0C
11000000 00000000 00070707 00000003 00000000
11000010 00000000 3000004C 00006000 00802000
11000020 00000000 00000000 00000000 00000000
11000030 00000000 00000000 00000000 00000000
11000040 00000000 00000000 0000B7DD F1020011

[1] Demo
[2] Read Serial Flash
[3] Write Serial Flash
[4] Erase Serial Flash

[9] Exit
```

5.9.3 シリアルフラッシュ書き込み

指定したアドレスに対し、SPI動作モードでデータの書き込みを行います。

メニューにて [3] を選択し、データを書き込む先頭アドレスを入力します。

```
>3[Enter]
Input the top address (10000000 - 13FFFFFFC)
>13000000[Enter]
Input the data (00000000 - FFFFFFFF)
Input just [Enter] without data to exit
13000000 FFFFFFFF >
```

先頭アドレスとそのデータの現在値が表示されるので、データを入力します。

現在値が未設定 (FFFFFFF) でない場合は、先に「5.9.4 シリアルフラッシュ消去」を行った後に書き込みを行ってください。

データ書き込み処理が終わると、次のロングワードのアドレスと現在値が表示されます。

```
13000000 FFFFFFFF > 1234[Enter]
13000004 FFFFFFFF >
```

書き込みを終えたい場合は、[Enter] のみを入力します。

```
13000000 FFFFFFFF > 00001234[Enter]
13000004 FFFFFFFF > 00000000[Enter]
13000008 FFFFFFFF > 00001234[Enter]
1300000C FFFFFFFF > 00000000[Enter]
13000010 FFFFFFFF > [Enter]
```

5.9.4 シリアルフラッシュ消去

指定したアドレスを含むセクタを、SPI動作モードで消去します。

メニューにて [4] を選択し、消去したいセクタ範囲内のアドレスを入力します。

```
>4[Enter]
Input the top address (10000000 - 13FFFFFFC)
>13001000[Enter]
```

入力したアドレスの含まれるセクタのデータが消去されます (例の場合は 13000000 - 1300FFFF の範囲)。

6. SPIBSC モード切り替え手順

6.1 分岐予測 / 投機実行により発生する可能性のある問題について

6.1.1 Cortex-R4 における ROM/RAM アクセス

MPUにて、あるアドレス領域のメモリタイプを Normal にした場合、分岐予測 / 投機実行により意図しないアクセスが発生する可能性があります。そのため、メモリタイプを Normal に設定したアドレス領域については、必ずアクセス可能状態にしておく必要があります。

6.1.2 SPI マルチ I/O バスコントローラのモード

SPI マルチ I/O バスコントローラには、外部アドレス空間リードモードと SPI 動作モードの 2 種類のモードがあります。

外部アドレス空間リードモードでは、SPI マルチ I/O バス空間に対し、CPU から直接アクセス可能です。そのため、このモードの時は SPI マルチ I/O バス空間のメモリタイプを Normal に設定することが出来ます。一方、SPI 動作モードは、SPI マルチ I/O バス空間に対し、CPU から直接アクセス不可です。そのため、このモードの時は SPI マルチ I/O バス空間のメモリタイプを Normal 以外またはアクセス禁止に設定する必要があります。

6.2 モード切り替え方法 概略

6.2.1 モードの切り替えについて

外部アドレス空間リードモードと SPI 動作モードを切り替える際は、適切に MPU の設定を変更してください。

6.2.2 変更手順概略

外部アドレス空間リードモードの状態では、MPU の変換テーブルを書き換えます。

SPI 動作モード切り替え後に、MPU の変換テーブルを書き換えると、変更が反映される前に SPI マルチ I/O バス空間へ意図しないアクセスが発生する可能性がありますので、必ず上記順序で行ってください。

SPI マルチ I/O バスコントローラのレジスタ設定変更処理を SPI マルチ I/O バス空間（シリアルフラッシュメモリ）上に配置して実行した場合、以降の動作が保証できません。当該処理については、内蔵 RAM で実行して頂きますようお願いいたします。

7. 応用例

本章では、サンプルプログラムの応用例として、お客様が使用するシリアルフラッシュメモリに応じたサンプルプログラムの変更方法について説明します。

7.1 サンプルプログラムの条件

サンプルプログラムは、Macronix 社製シリアルフラッシュメモリ（型名：MX25L51245G）を表 7.1 に示す条件で使用する場合に最適な設定を行っています。

これらの条件を変更する場合のサンプルプログラムの変更方法について説明します。

表7.1 サンプルプログラムの条件

条件	設定内容	備考
シリアルフラッシュメモリ	Macronix 社製シリアルフラッシュメモリ (型名：MX25L51245G)	—
データバス幅	4ビット	データリード時のビット幅
アドレスバイト数	4バイト	アドレス指定時に発行するバイト数

7.2 シリアルフラッシュメモリを変更しない場合のサンプルプログラム変更方法

表 7.2 に使用するシリアルフラッシュメモリを変更しない場合のサンプルプログラムの変更方法を示します。

表7.2 シリアルフラッシュメモリを変更しない場合のサンプルプログラム変更方法

条件	変更内容	変更方法
データリード時のデータバス幅	1ビット	マクロ定義 (SPIBSC_BUS_WITDH) 注1に (1) を定義
	4ビット	マクロ定義 (SPIBSC_BUS_WITDH) に (4) を定義
アドレスバイト数	3バイト	マクロ定義 (SPIBSC_OUTPUT_ADDR) 注2に (SPIBSC_OUTPUT_ADDR_24) を定義
	4バイト	マクロ定義 (SPIBSC_OUTPUT_ADDR) に (SPIBSC_OUTPUT_ADDR_32) を定義

注1. マクロ定義 (SPIBSC_BUS_WITDH) はspibsc_ioreset_userdef.c ファイル内に定義されています。注3

注2. マクロ定義 (SPIBSC_OUTPUT_ADDR) はspibsc_ioreset_userdef.c ファイル内に定義されています。注3

注3. 16ビットバスブートモード時は、serial_flash_ioreset_userdef.c ファイル内の上記マクロ定義を参照してください。

7.3 シリアルフラッシュメモリを変更する場合のサンプルプログラム変更方法

シリアルフラッシュメモリを変更する場合、使用するシリアルフラッシュメモリの仕様に合わせてサンプルプログラムを変更する必要があります。

表 7.3 にサンプルプログラムの変更のポイントを示します。

表7.3 サンプルプログラムの変更のポイント

変更のポイント	内容
リードコマンド波形	外部アドレス空間リードモードにより、SPIマルチI/Oバス空間へのリードをSPI通信に変換する際にシリアルフラッシュメモリに出力する信号を使用するシリアルフラッシュメモリのリードコマンドに合わせて変更します。
シリアルフラッシュメモリ内レジスタ設定	使用するシリアルフラッシュメモリに応じて、SPIBSCを外部アドレスリードモードで使用する場合に必要なシリアルフラッシュメモリ内レジスタの設定を行います。
シリアルフラッシュメモリライト許可	使用するシリアルフラッシュメモリに応じて、シリアルフラッシュメモリ内レジスタを設定し、ライト許可に設定します。注1
シリアルフラッシュメモリレディー待ち	使用するフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを読み出し、シリアルフラッシュメモリがレディー状態に遷移するのを待ちます。
シリアルフラッシュメモリプロテクト解除	使用するフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを設定し、プロテクトを解除します。注2

注1. シリアルフラッシュメモリにより、シリアルフラッシュメモリ内のレジスタを設定するために、ライト許可が必要な場合があります。

注2. シリアルフラッシュメモリにより、シリアルフラッシュメモリ内のレジスタを設定するために、プロテクト解除が必要な場合があります。

7.3.1 リードコマンド波形の変更

外部アドレス空間リードモードにより、SPI マルチ I/O バス空間へのリードを SPI 通信に変換する際にシリアルフラッシュメモリに出力する信号を使用するシリアルフラッシュメモリのリードコマンドに合わせて変更します。

SPIBSC は、SPIBSC 制御レジスタ設定より、外部アドレス空間リードモードにてシリアルフラッシュメモリに出力する信号を変更することが可能です。

サンプルプログラムでは、SPIBSC 制御レジスタに設定する値をグローバル変数 (SPIBSC 外部アドレスリード設定内容格納変数 : spibsc_cfg) にて変更することが可能です。spibsc_cfg の設定はユーザ定義関数 (SPIBSC 外部アドレスリード設定関数 : userdef_spibsc_set_config) にて行っています。

図 7.1 に SPIBSC 制御レジスタと SPIBSC 外部アドレスリード時にシリアルフラッシュメモリに出力される波形の関係を、表 7.4 にサンプルプログラムの SPIBSC 制御レジスタ設定値を示します。

本設定例を参考に、使用するシリアルフラッシュメモリのリードコマンドに合わせて spibsc_cfg の設定を行ってください。

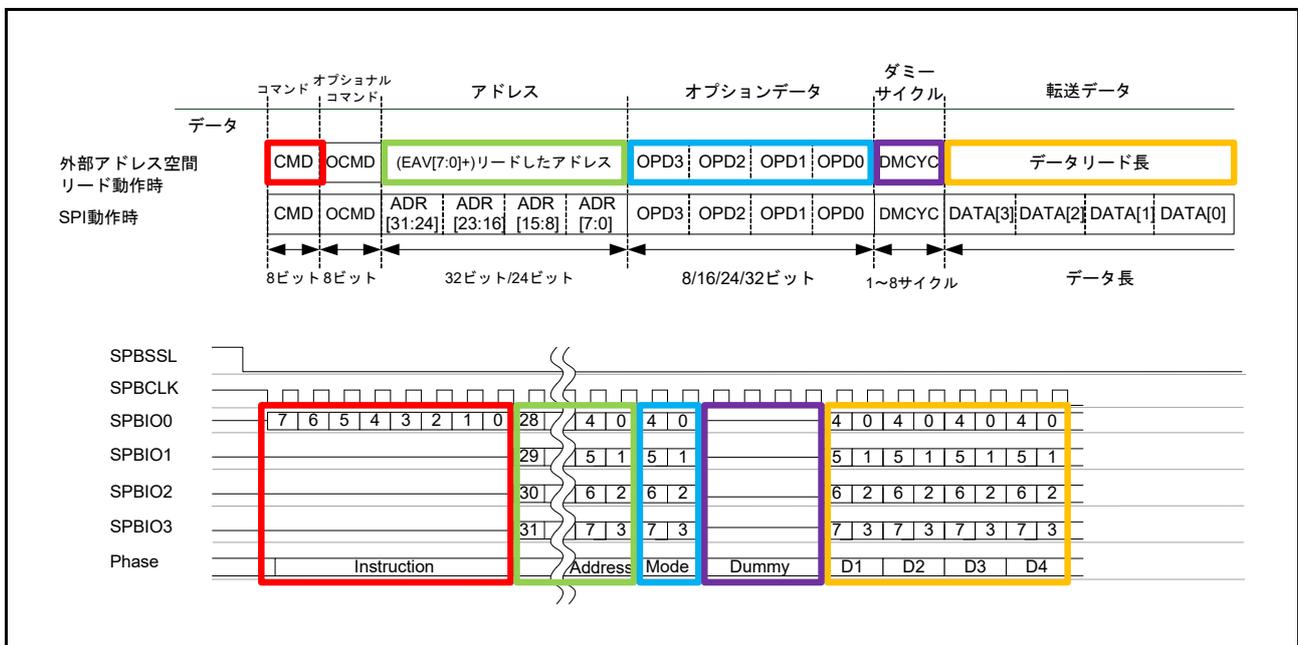


図 7.1 SPIBSC 制御レジスタと SPIBSC 外部アドレスリード時にシリアルフラッシュメモリに出力される波形の関係

なお、SPI ブート時は、ブート処理のユーザ定義関数 Userdef_SPIBSC_Set_Config も、使用するシリアルフラッシュメモリのリードコマンドに合わせて変更してください。

表7.4 サンプルプログラムのSPIBSC制御レジスタ設定内容

SPIBSC Registers		Setting	Remark
DRCMR	CMD[7:0]	H'EB	Quad I/O read command
	OCMD[7:0]	H'00	—
DROPR	OPD3[7:0]	H'00	—
	OPD2[7:0]	H'00	—
	OPD1[7:0]	H'00	—
	OPD0[7:0]	H'00	—
DRENr	CDB[1:0]	B'00	コマンドビット幅：1ビット幅
	OCDB[3:0]	B'0000	—
	ADB[1:0]	B'10	アドレスビット幅：4ビット幅
	OPDB[1:0]	B'10	オプションデータビット幅：4ビット幅
	DRDB[1:0]	B'10	データリードビット幅：4ビット幅
	DME	B'1	ダミーサイクル：挿入する
	CDE	B'1	コマンド：発行する
	OCDE	B'0	オプションコマンド：発行しない
	ADE[3:0]	B'0111	アドレスイネーブル：24ビットアドレス出力
	OPDE[3:0]	B'1000	オプションナルデータ：OPD3を出力
DRDMCR	DMDB[1:0]	B'00	ダミーサイクルビット幅：1ビット幅
	DMCYC[2:0]	B'101	ダミーサイクル数：6サイクル
SPBCR	SPBR[7:0]	H'01	ビットレート：PCLKA/2
	BRDV[1:0]	B'00	

7.3.2 シリアルフラッシュメモリ内レジスタ設定

7.3.1 リードコマンド波形の変更にて、シリアルフラッシュメモリからリードする場合、シリアルフラッシュメモリ内のレジスタを設定する必要があります。

サンプルプログラムでは、ユーザ定義関数（シリアルフラッシュメモリ内レジスタ設定関数：`userdef_sflash_set_mode`）に処理を実装しており、Macronix 社製シリアルフラッシュメモリ（型名：MX25L51245G）内の Status Register 内の QUAD ビットを 1 (=Quad) に、Configuration Register 内の DC1 (Dummy Cycle1) ビットを 1 に、DC0 (Dummy Cycle0) ビットを 0 に設定しています。

なおシリアルフラッシュメモリ内のレジスタの設定には SPIBSC の SPI モードを使用します。また Macronix 社製シリアルフラッシュメモリ（型名：MX25L51245G）内のレジスタを設定する場合、Write Enable Command (WREN) より、Write Enable Latch (WEL) bit を 1 に設定する必要があるため、Status Register および Configuration Register 設定前に Write Enable Command (WREN) を発行しています。サンプルプログラムではユーザ定義関数（シリアルフラッシュメモリライト許可関数：`userdef_sflash_write_enable`）にて、Write Enable Command (WREN) 発行処理を実装しています。

図 7.2 にサンプルプログラムのシリアルフラッシュメモリ内レジスタ設定フローを示します。

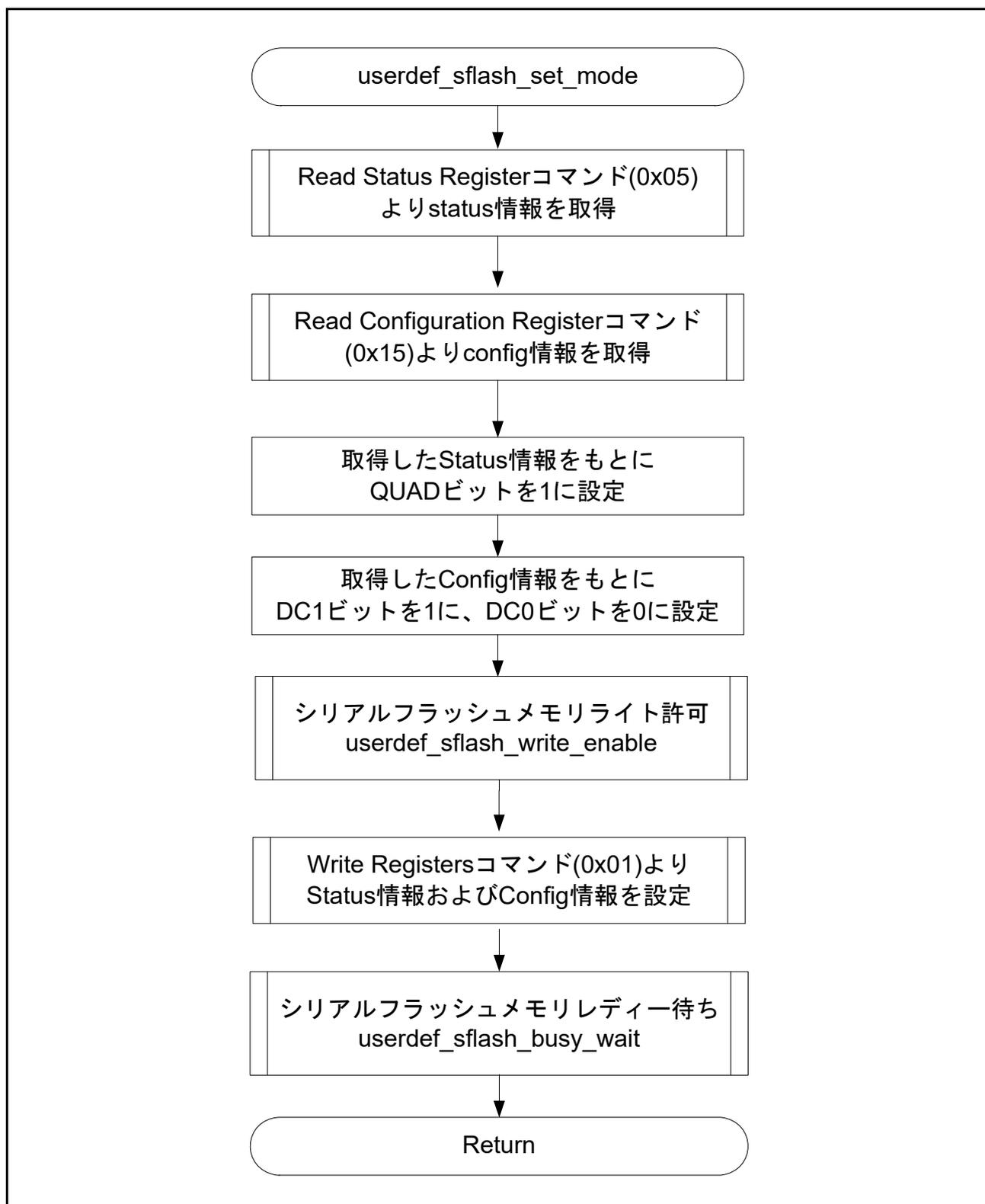


図 7.2 シリアルフラッシュメモリ内レジスタ設定フロー

なお、SPI ブート時は、ブート処理のユーザ定義関数 `Userdef_SPIBSC_Set_Mode` も、使用するシリアルフラッシュメモリのレジスタ設定に合わせて変更してください。

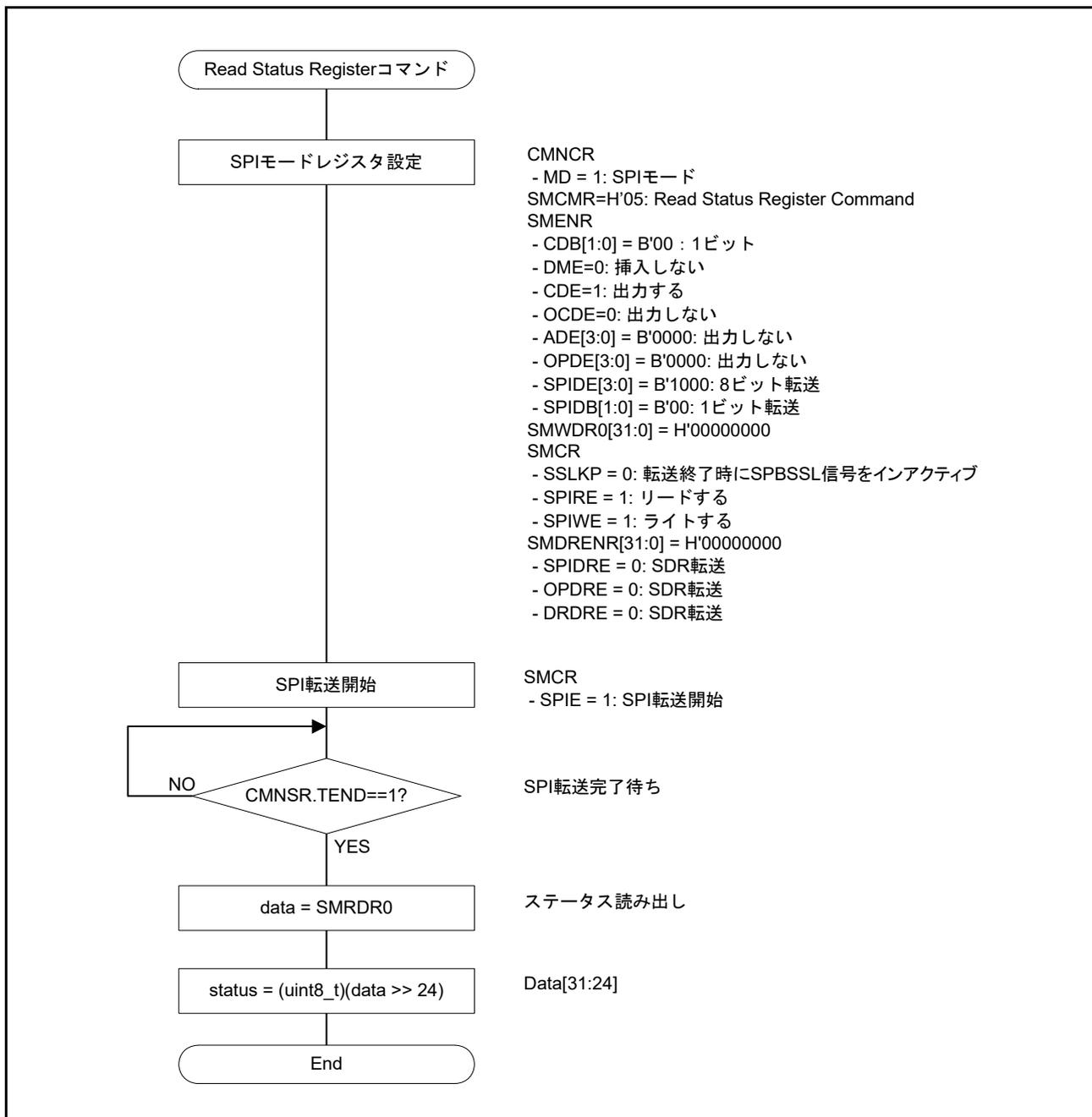


図 7.3 Read Status Register コマンドフロー

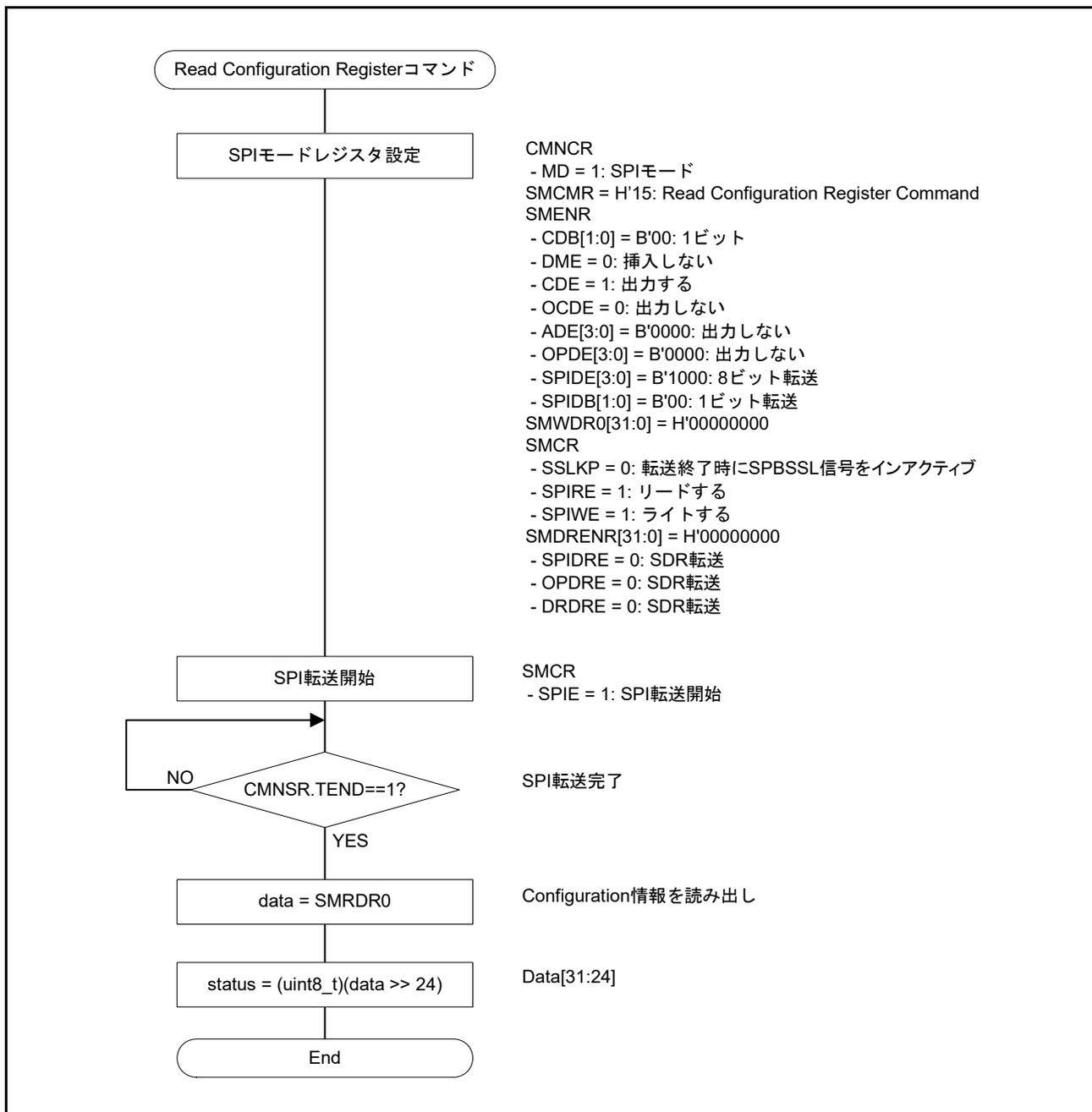


図 7.4 Read Configuration Register コマンドフロー

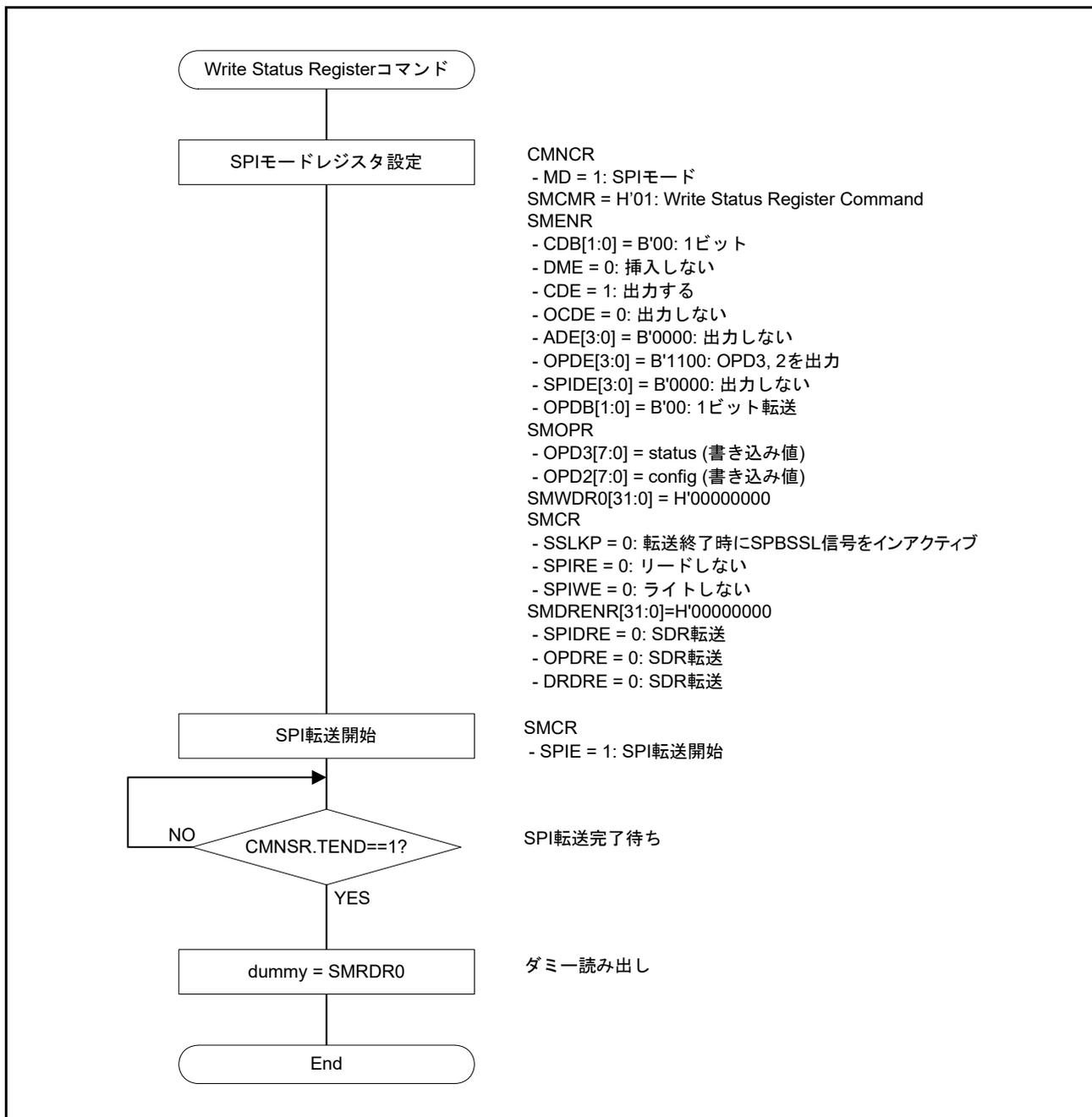


図 7.5 Write Status Register コマンドフロー

7.3.3 シリアルフラッシュメモリライト許可

7.3.2 シリアルフラッシュメモリ内レジスタ設定より、シリアルフラッシュメモリ内レジスタを設定する際に必要なライト許可処理をユーザ定義関数（シリアルフラッシュメモリライト許可関数：`userdef_sflash_write_enable`）に実装してください。

図 7.6 にサンプルプログラムのシリアルフラッシュメモリライト許可フローを示します。

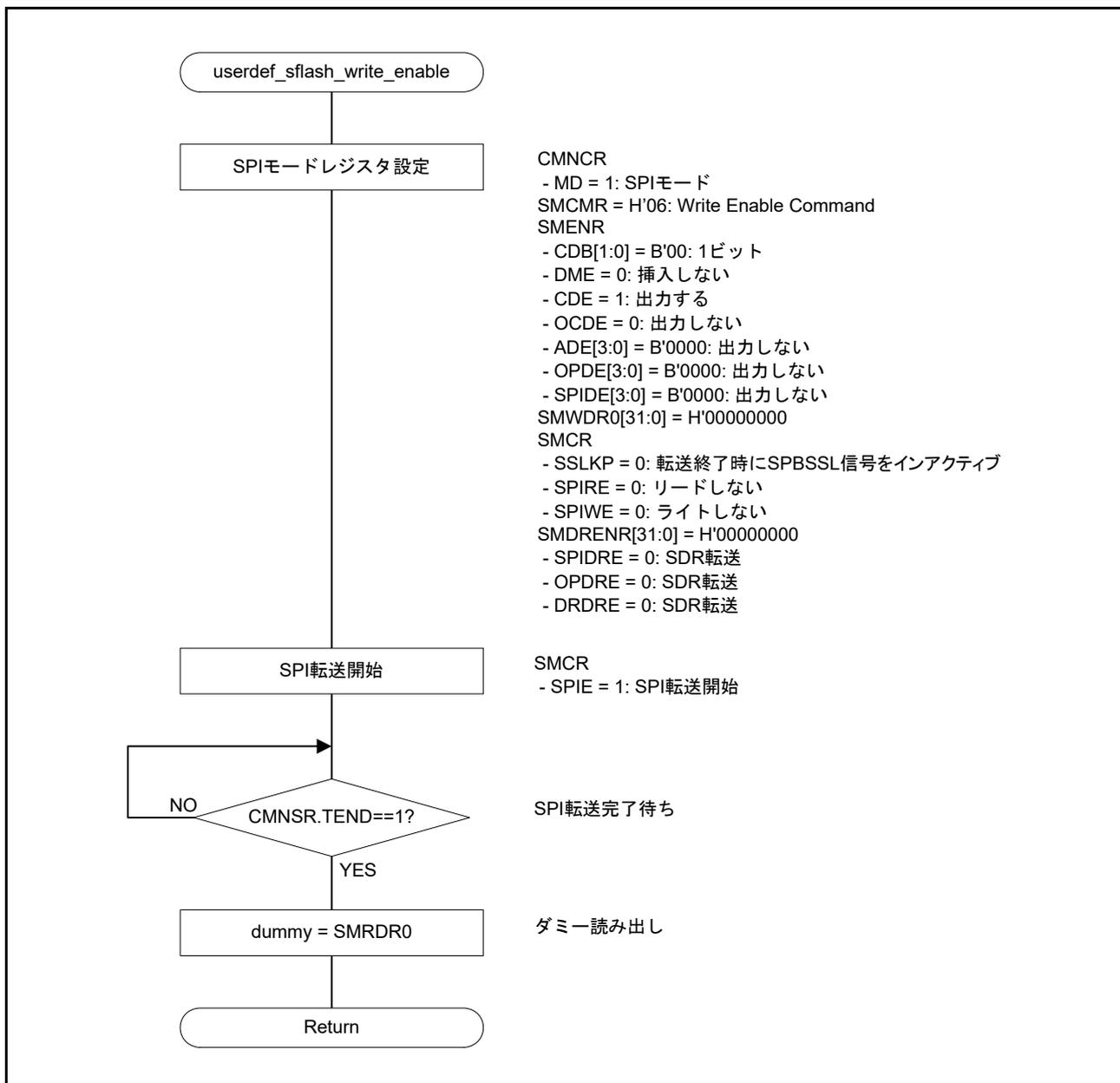


図 7.6 シリアルフラッシュメモリライト許可フロー

なお、SPI ブート時は、ブート処理のユーザ定義関数 `Userdef_SPIBSC_Write_Enable` も、使用するシリアルフラッシュメモリに必要なライト許可処理に合わせて変更してください。

7.3.4 シリアルフラッシュメモリレディー待ち

シリアルフラッシュメモリに対して、書き込みコマンド (Page Program) または消去コマンド (Sector Erase) を発行した場合、シリアルフラッシュメモリはビジー状態に遷移します。ビジー状態からレディー状態への遷移を待つ処理をユーザ定義関数 (シリアルフラッシュメモリレディー待ち関数 : `userdef_sflash_busy_wait`) に実装してください。

Macronix 社製シリアルフラッシュメモリ (型名 : MX25L51245G) では、レディー状態への遷移をシリアルフラッシュメモリ内のレジスタ (Status Register) にて確認することができます。

図 7.7 にサンプルプログラムのシリアルフラッシュメモリレディー待ちフローを示します。

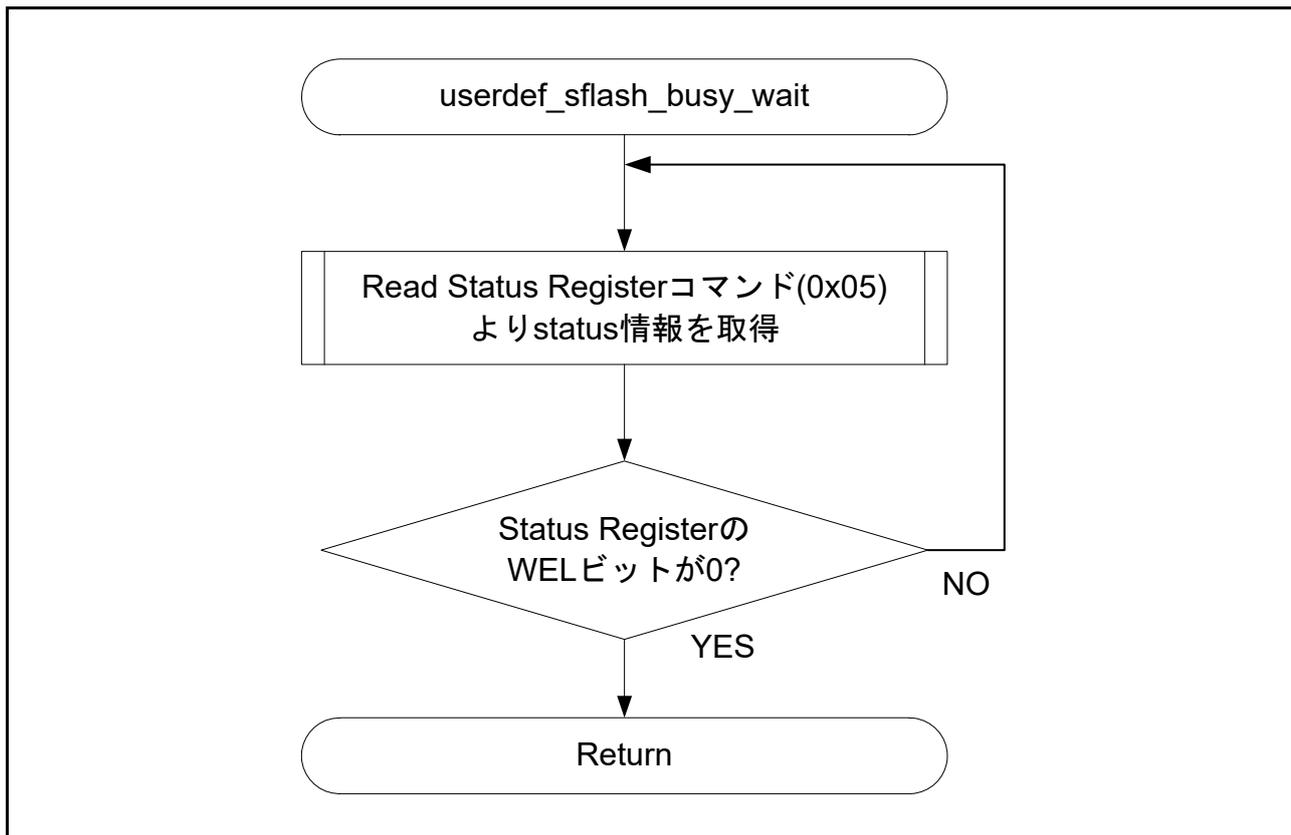


図 7.7 シリアルフラッシュメモリレディー待ちフロー

なお、SPI ブート時は、ブート処理のユーザ定義関数 `Userdef_SFLASH_Busy_Wait` も、使用するシリアルフラッシュメモリのレディー待ち処理に合わせて変更してください。

7.3.5 参考：シリアルフラッシュメモリプロテクト解除

シリアルフラッシュメモリ仕様より、シリアルフラッシュメモリ内のデータを変更する場合、シリアルフラッシュメモリ内のレジスタを操作し、プロテクトを解除する必要があります。プロテクト解除処理が必要な場合は、SPIブート処理のユーザ定義関数 `Userdef_SFLASH_Ctrl_Protect` 関数を参考に実装してください。

Macronix社製シリアルフラッシュメモリ（型名：MX25L51245G）では、シリアルフラッシュメモリがプロテクト状態にある場合、シリアルフラッシュメモリへの書き込みおよび消去を実行することができません。プロテクトを解除するためには、Status Register内のBlock Protection（BP3, BP2, BP1, BP0）ビットに0を設定する必要があります。

図7.8にシリアルフラッシュメモリプロテクト解除フローを示します。

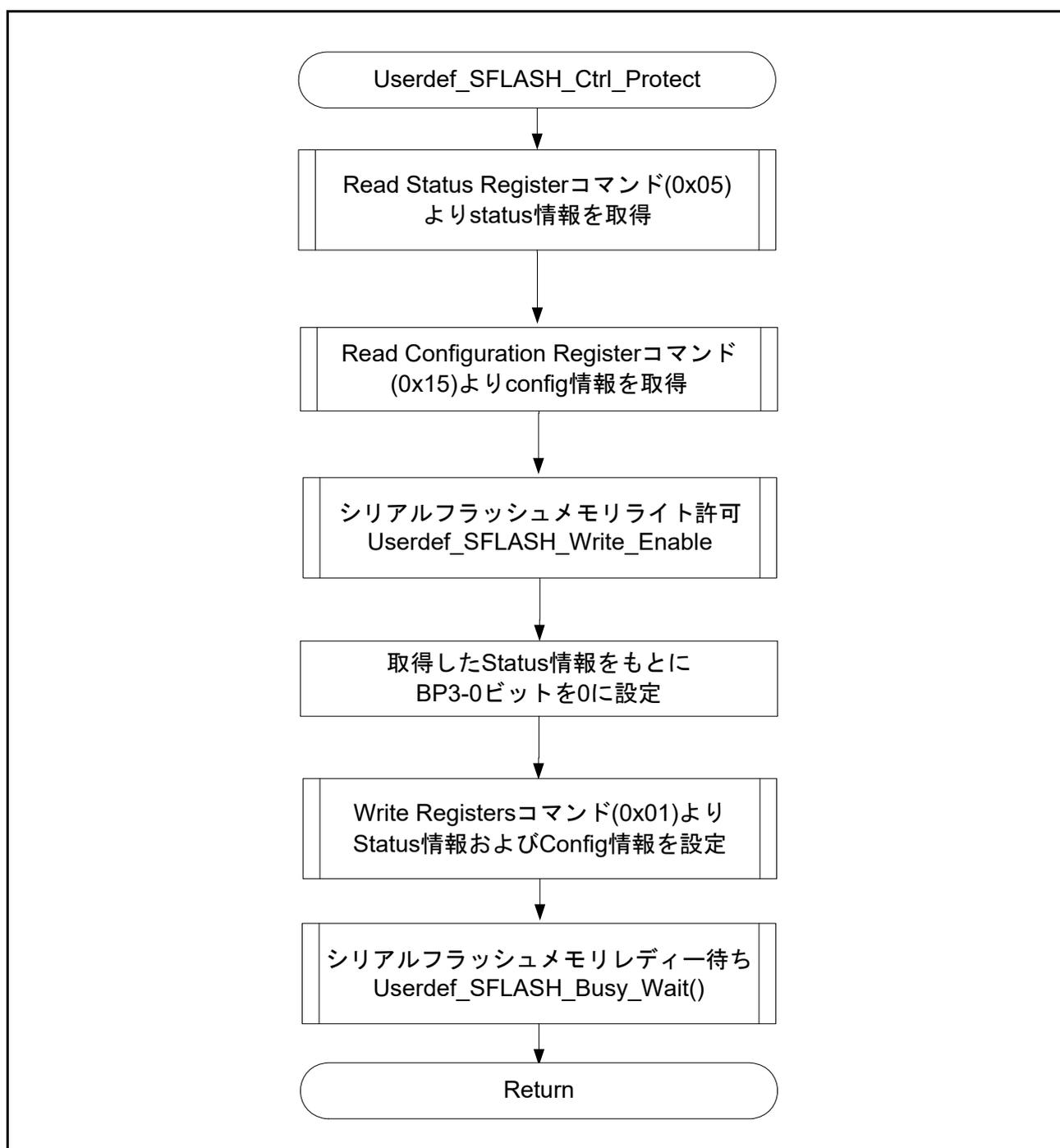


図 7.8 シリアルフラッシュメモリプロテクト解除フロー

8. サンプルプログラム

サンプルプログラムは、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
RZ/T1 グループ ユーザーズマニュアルハードウェア編
(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。)
 - RZ/T1 Evaluation Board RTK7910022C00000BR ユーザーズマニュアル
(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
IAR 統合開発環境 (IAR Embedded Workbench® for Arm) に関しては、IAR ホームページから入手してください。
(最新版を IAR ホームページから入手してください。)
- Arm ソフトウェア開発ツール (Arm Compiler toolchain、Arm DS-5 等) に関しては、Arm ホームページから入手してください。
(最新版を Arm ホームページから入手してください。)
- ルネサス エレクトロニクスソフトウェア開発ツール (e2studio 等) に関しては、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。
(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。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/contact/>

改訂記録	シリアルフラッシュサンプルプログラム(SPIBSC) アプリケーションノート
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.30	2017.09.14	—	初版発行
1.40	2018.06.07	2. 動作環境	
		5	表2.1 動作環境 統合開発環境の内容変更
		9. 参考ドキュメント	
		51	ARM→Armに変更

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。

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. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】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. リザーブアドレス（予約領域）のアクセス禁止

【注意】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、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）がありません。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。

4. クロックについて

【注意】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。

5. 製品間の相違について

【注意】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っていると、内部 ROM、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。型名が異なる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)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>