

# SH7753 グループ

R01AN1606JJ0100

## RSPI ドライバソフトウェア

Rev.1.00

2013.7.1

### 要旨

本仕様書は SH7753 グループの RSPI (Renesas Serial Peripheral Interface) ドライバについて説明します。

### 対象デバイス

SH7753

### 動作環境

本仕様書に示す RSPI ドライバの動作環境を以下に示します。

- 評価ボード : SH7753 グループ EVB ボード
- SPI フラッシュ : モデル - MX25L64
  - : サイズ - 8MB
  - : モード - CPOL = 0, CPHA = 0 / CPOL = 1, CPHA = 1
  - : クロック - 最大使用周波数 24Mhz
  - : 動作 - リード ID、リードステータス、ライトイネーブル/ディセーブル、ページライト、リード、ファストリード、デュアルアウトプットリード、イレーズ (セクタ (64KB)、バルク)
- ソフトウェア : High-Performance Embedded Workshop- Ver 4.09.01.007
  - : Toolchain - Ver 9.4.1.0
  - : OptLinker - Ver 10.01.00
  - : SH アセンブラ - Ver 7.01.02
  - : SH C/C++コンパイラ- Ver 9.04.01
  - : SH C/C++ライブラリジェネレータ - Ver 3.00.03
- エミュレータ : E10A エミュレータ Ver 3.03.00

## 目次

1.	機能概要 .....	3
1.1	主な機能 .....	3
1.2	関連ドキュメント .....	3
2.	ドライバ仕様 .....	4
2.1	関数一覧 .....	4
2.2	コールバック関数仕様一覧 .....	4
2.3	エラーコード一覧 .....	5
3.	関数 .....	6
	R_RSPI_Init .....	7
	R_RSPI_SetCommand .....	8
	R_RSPI_ClearErrorStatus .....	9
	R_RSPI_GetCommandStatus .....	10
	R_RSPI_GetStatus .....	11
	R_RSPI_ComStart .....	12
	R_RSPI_ComDmacModeStart .....	13
	R_RSPI_TxFifoDataSet .....	14
	R_RSPI_RxFifoDataGet .....	15
	R_RSPI_ComStop .....	16
	R_RSPI_TxInterrupt .....	17
	R_RSPI_RxInterrupt .....	18
	R_RSPI_ErrInterrupt .....	19
	R_RSPI_IdleInterrupt .....	20
4.	動作条件 .....	21
4.1	ポートについて .....	21
4.2	割り込みについて .....	21
4.3	使用メモリ領域について .....	21
5.	付録 .....	22
5.1	ドライバ関数の使用例 .....	22
5.1.1	SPIフラッシュリードデータアクセス時の使用例 .....	22
5.1.2	SPIフラッシュページライトアクセス時の使用例 .....	23
5.1.3	SPIフラッシュリードデータアクセス時の使用例 (DMAC連動) .....	24
5.1.4	SPIフラッシュページライトアクセス時の使用例 (DMAC連動) .....	25

## 1. 機能概要

本ドライバは SH7753 に搭載されている RSPI を使用し、SPI インタフェースを持つデバイスと通信する為の関数群で構成されています。

### 1.1 主な機能

RSPI ドライバの主な機能を以下に示します。

- ・ 初期化設定
- ・ コマンドオペレーション設定／ステータス情報取得
- ・ マスタ／スレーブシリアルデータ通信
- ・ エラー通知
- ・ DMAC 連動シリアルデータ通信

### 1.2 関連ドキュメント

SH7753 グループ ユーザーズマニュアル：ハードウェア編

SH7753 グループ アプリケーションノート DMAC ドライバソフトウェア

## 2. ドライバ仕様

### 2.1 関数一覧

表 1に、RSPI ドライバの関数一覧を示します。

表 1. RSPI ドライバ関数一覧

分類	関数名	機能概要	スタック サイズ
設定関数	R_RSPI_Init	初期設定処理	8 バイト
	R_RSPI_SetCommand	コマンド設定処理	20 バイト
	R_RSPI_ClearErrorStatus	エラーステータスクリア処理	8 バイト
情報取得関数	R_RSPI_GetCommandStatus	カレントコマンドステータス情報取得処理	0 バイト
	R_RSPI_GetStatus	ステータス情報取得処理	8 バイト
データ通信関数	R_RSPI_ComStart	通信開始処理	0 バイト
	R_RSPI_ComDmacModeStart	DMAC 連動用通信開始処理	4 バイト
	R_RSPI_TxFifoDataSet	送信 FIFO データ設定処理	28 バイト
	R_RSPI_RxFifoDataGet	受信 FIFO データ取得処理	28 バイト
	R_RSPI_ComStop	通信停止処理	4 バイト
割り込みハンドラ 関数	R_RSPI_TxInterrupt	送信 FIFO エンプティ割り込みハンドラ	4 バイト
	R_RSPI_RxInterrupt	受信 FIFO フル割り込みハンドラ	4 バイト
	R_RSPI_ErrInterrupt	エラー割り込みハンドラ	8 バイト
	R_RSPI_IdleInterrupt	アイドル割り込みハンドラ (※)	0 バイト

※アイドル割り込みは、本ドライバでは使用しません。

### 2.2 コールバック関数仕様一覧

表 2に RSPI ドライバのコールバック関数仕様一覧を示します。RSPI ドライバのコールバック関数は各ドライバ関数の処理完了時に通知としてコールします。

表 2. RSPI ドライバコールバック関数仕様一覧

関数名	コールバック名	仕様内容
R_RSPI_ComStart	void ( *pv_error_ntf_callback )( char_t c_error_code )	エラー発生時のコールバック (NULL の場合、コールバックなし)
R_RSPI_TxFifoDataSet	void ( *pv_tx_fifo_empty_ntf_callback )( void )	送信 FIFO エンプティ通知コールバック (NULL の場合、コールバックなし)
	void ( *pv_rx_fifo_full_ntf_callback )( void )	受信 FIFO フル通知コールバック (NULL の場合、コールバックなし)

## 2.3 エラーコード一覧

表 3に RSPI ドライバのエラーコード一覧を示します。

表 3. RSPI ドライバエラーコード一覧

値	エラーコード (マクロ定義)	エラー内容
0	RET_NORMAL	正常終了
-1	RET_ERR_PARAM1	第 1 引数不正
-2	RET_ERR_PARAM2	第 2 引数不正
-3	RET_ERR_PARAM3	第 3 引数不正
-4	RET_ERR_PARAM4	第 4 引数不正
-20	RET_ERR_PARITY	パリティエラー
-21	RET_ERR_MODEFAULT	モードフォールトエラー
-22	RET_ERR_OVERRUN	オーバーランエラー

### 3. 関数

本章では、RSPI ドライバの各関数仕様詳細を示します。各関数詳細の読み方は以下のとおりです。

関数名		分類
機能概要		
書式	関数の呼び出し形式を示します。#include “ヘッダファイル”で示すヘッダファイルは、この関数の実行に必要な標準ヘッダファイルで、必ずインクルードします。	
引数	I,O は、引数がそれぞれ入力データ、出力データであることを意味します。	
戻り値	関数の戻り値を示します。	
解説	関数の仕様について説明します。	
注意	注意事項があればここに示します。	

## R\_RSPI\_Init

設定関数

## 初期設定処理

```
書式      #include "r_common.h"
          #include "iodefine.h"
          #include "r_rspi_if.h"
          char_t R_RSPI_Init( rspi_init_t *pst_init );
```

引数 rspi\_init\_t \*pst\_init I RSPI 初期設定管理構造体へのポインタ

戻り値 RET\_NORMAL 正常終了  
 RET\_ERR\_PARAM1 第 1 引数不正

解説 本関数は、RSPI 初期設定管理構造体の情報に基づき、RSPI の初期設定を行います。

## 【RSPI 初期設定管理構造体】 rspi\_init\_t

型	名称	I/O	機能説明
uchar_t	uc_mode	I	モード (0 : スレーブ、1 : マスタ)
uchar_t	uc_txmd	I	送信モード (0 : 送受信、1 : 送信のみ)
uchar_t	uc_spms	I	オペレーションモード選択 (0 : SPI、1 : クロック同期)
uchar_t	uc_splw	I	データレジスタアクセス幅 (0 : Word (16 ビット)、1 : Long (32 ビット))
uchar_t	uc_spfc	I	データレジスタフレーム数 (1~4)
uchar_t	uc_spsln	I	*シーケンスレングス (0~7)
uchar_t	uc_spbr	I	*ベースビットレート (0~255)
uchar_t	uc_sslp	I	スレーブセレクト信号極性 (0 : "L"アクティブ、1 : "H"アクティブ)
uchar_t	uc_slssel	I	*スレーブセレクトピン出力選択 0 : SLSEL1 = 0、SLSEL0 = 0 1 : SLSEL1 = 0、SLSEL0 = 1 2 : SLSEL1 = 1、SLSEL0 = 0
uchar_t	uc_moife	I	*MOSI アイドル値 (0 : 最終送信データ、1 : MOIFV)
uchar_t	uc_moifv	I	*MOSI アイドル固定値 (0 : 0 固定、1 : 1 固定)
uchar_t	uc_sckdl	I	*RSPCK 遅延 (0 : 1 RSPCK ~ 111 : 8 RSPCK)
uchar_t	uc_sslndl	I	*スレーブセレクト信号ネゲート遅延 0 : 1 RSPCK ~ 111 : 8 RSPCK
uchar_t	uc_spndl	I	*次アクセス遅延 0 : 1 RSPCK + 2 Pck ~ 111 : 8 RSPCK + 2 Pck
uchar_t	uc_modfen	I	*Mode Fault エラー検出 (0 : 無効、1 : 有効)
uchar_t	uc_sppe	I	パリティ有効/無効 (0 : 無効、1 : 有効)
uchar_t	uc_spoie	I	パリティモード (0 : 偶数パリティ、1 : 奇数パリティ)
uchar_t	uc_pte	I	パリティ Diagnosis (0 : 無効、1 : 有効)

(\*)スレーブモード設定時(uc\_mode = 0)に、"0"を設定してください。

注意 RSPI のポート設定後に、本関数をコールすること。

RSPI が通信状態で、本関数をコールしないこと。

# R\_RSPI\_SetCommand

設定関数

## コマンド設定処理

書式	<pre>#include "r_common.h" #include "iodefine.h" #include "r_rspi_if.h" char_t R_RSPI_SetCommand( uchar_t uc_spcmd_no, rspi_spcmd_t *pst_spcmd );</pre>		
引数	<pre>uchar_t uc_spcmd_no rspi_cmd_t *pst_spcmd</pre>	<pre>   </pre>	<pre>コマンド番号 (0~7) RSPI コマンド設定管理構造体へのポインタ</pre>
戻り値	<pre>RET_NORMAL RET_ERR_PARAM1 RET_ERR_PARAM2</pre>	<pre> </pre>	<pre>正常終了 第 1 引数不正 第 2 引数不正</pre>
解説	<p>本関数は、RSPI コマンド設定管理構造体の情報に基づき、該当コマンドレジスタの設定を行います。</p>		

### 【RSPI コマンド設定管理構造体】 rspi\_spcmd\_t

型	名称	I/O	機能説明
uchar_t	uc_lsbfb	I	LSB ファースト (0: 無効 (MSB ファースト)、1: 有効)
uchar_t	uc_spb	I	データ レンゲス 0: 20 ビット、1: 24 ビット、2~3: 32 ビット 4~7: 8 ビット、8~15: 9 ビット~16 ビット
uchar_t	uc_cpol	I	クロック Polarity (0、1)
uchar_t	uc_cpha	I	クロック Phase (0、1)
uchar_t	uc_brdv	I	*ビットレート分周設定 0: ベースビットレート 1: ベースビットレートの 2 分周 2: ベースビットレートの 4 分周 3: ベースビットレートの 8 分周
uchar_t	uc_sslkp	I	*スレーブセレクト信号のレベル保持 (0: ネゲート、1: 保持)
uchar_t	uc_ssla	I	*スレーブセレクト信号アサート設定 (0: SSL0、1: SSL1)
uchar_t	uc_sckden	I	*RSPCK 遅延制御 (0: 無効、1: 有効)
uchar_t	uc_slnden	I	*スレーブセレクト信号ネゲート遅延制御 (0: 無効、1: 有効)
uchar_t	uc_spnden	I	*次アクセス遅延制御 (0: 無効、1: 有効)

(\*)コマンド番号が”0”、かつスレーブモード状態で、”0”を設定してください。

**注意** マスタモードで RSPI が非アイドル状態 (IDNLF==1) 時、本関数をコールしないこと。スレーブモードの場合、RSPI が通信状態で、本関数を使用しないこと。



# R\_RSPI\_ClearErrorStatus

設定関数

エラーステータスクリア処理

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_rspi_if.h"
char_t R_RSPI_ClearErrorStatus( uchar_t uc_error_flag );
```

引数

uchar_t uc_error_flag		エラーフラグ
-----------------------	--	--------

戻り値

RET_NORMAL	正常終了
RET_ERR_PARAM1	第1引数不正

解説

本関数は、引数に指定したエラーフラグに該当する RSPI エラーステータスフラグ（オーバーラン／モードフォルト／パリティエラー）のクリア処理を行います。

## 【エラーフラグ詳細】

フラグ	ビット位置	詳細
パリティエラーフラグ	bit[0]	0: フラグクリアなし、1: フラグクリアあり
モードフォルトエラーフラグ	bit[1]	0: フラグクリアなし、1: フラグクリアあり
オーバーランエラーフラグ	bit[2]	0: フラグクリアなし、1: フラグクリアあり

注意

なし

---

## R\_RSPI\_GetCommandStatus

情報取得関数

カレントコマンドステータス情報取得処理

---

書式	<pre>#include "r_common.h" #include "iodefine.h" #include "r_rspi_if.h" char_t R_RSPI_GetCommandStatus( uchar_t *puc_current_spcmd, uchar_t *puc_error_spcmd );</pre>						
引数	<table><tr><td>uchar_t *puc_current_spcmd</td><td>○</td><td>カレントコマンド番号情報格納領域</td></tr><tr><td>uchar_t *puc_error_spcmd</td><td>○</td><td>エラーコマンド番号情報格納領域 (エラー発生時のみ有効)</td></tr></table>	uchar_t *puc_current_spcmd	○	カレントコマンド番号情報格納領域	uchar_t *puc_error_spcmd	○	エラーコマンド番号情報格納領域 (エラー発生時のみ有効)
uchar_t *puc_current_spcmd	○	カレントコマンド番号情報格納領域					
uchar_t *puc_error_spcmd	○	エラーコマンド番号情報格納領域 (エラー発生時のみ有効)					
戻り値	RET_NORMAL          正常終了						
解説	本関数は、RSPI のカレントコマンド及びエラーコマンドの番号情報取得処理を行います。						
注意	なし						

## R\_RSPI\_GetStatus

RSPI 情報取得関数

ステータス情報取得処理

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_rspi_if.h"
char_t R_RSPI_GetStatus( void );
```

引数           なし

戻り値        0~0x3F                               ステータス情報

解説           本関数は、RSPI のアイドルフラグ、エラーフラグ及び送受信 FIFO のステータス情報の取得処理を行います。

### 【ステータス情報詳細】

フラグ	ビット位置	詳細
アイドルフラグ	bit[0]	0 : アイドル、1 : 通信中
送信 FIFO エンプティフラグ	bit[1]	0 : 非エンプティ、1 : エンプティ
受信 FIFO フルフラグ	bit[2]	0 : 非フル、1 : フル
パリティエラーフラグ	bit[3]	0 : エラーなし、1 : エラーあり
モードフォルトエラーフラグ	bit[4]	0 : エラーなし、1 : エラーあり
オーバーランエラーフラグ	bit[5]	0 : エラーなし、1 : エラーあり

注意           なし

## R\_RSPI\_ComStart

データ通信関数

### 通信開始処理

書式	<pre>#include "r_common.h" #include "iodefine.h" #include "r_rspi_if.h" char_t R_RSPI_ComStart( void (*pv_error_callback)( char_t c_error_code ) );</pre>		
引数	<table><tr><td><code>void (*pv_error_ntf_callback)( char_t c_error_code )</code></td><td>エラー発生通知コールバックアドレス (NULL の場合、コールバックなし)</td></tr></table>	<code>void (*pv_error_ntf_callback)( char_t c_error_code )</code>	エラー発生通知コールバックアドレス (NULL の場合、コールバックなし)
<code>void (*pv_error_ntf_callback)( char_t c_error_code )</code>	エラー発生通知コールバックアドレス (NULL の場合、コールバックなし)		
戻り値	RET_NORMAL 正常終了		
解説	<p>本関数は、RSPI の通信開始処理を行います。</p> <p>本関数は、RSPI のエラー割り込み有効ビット及び、機能有効ビット (SPE) をセットし、RSPI を通信状態にします。</p> <p>本関数コール後、送信 FIFO 設定関数 (R_RSPI_TxFifoDataSet) をコールすることにより、接続デバイスへのデータ送受信ができます。また、エラー (オーバーラン、モードフォルト、パリティエラー) が発生した場合、エラーコードを引数として、pv_error_ntf_callback に指定した定義関数をコールすることで通知します。</p> <p>エラー発生通知コールバック関数の NULL 設定で本関数コール後、エラーステータスフラグをクリアまたは確認する場合、R_RSPI_ClearErrorStatus、R_RSPI_GetStatus関数を使用してください。</p>		
注意	RSPI が通信停止状態で、本関数をコールすること。		

## R\_RSPI\_ComDmacModeStart

データ通信関数

DMAC 連動用通信開始処理

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_rspi_if.h"
char_t R_RSPI_ComDmacModeStart( void );
```

引数           なし

戻り値         RET\_NORMAL                                 正常終了

解説

本関数は、DMAC 連動用 RSPI の通信開始処理を行います。

本関数は、RSPI の送信 FIFO エンプティ割り込み有効ビット、受信 FIFO フル割り込み有効ビット及び、機能有効ビット (SPE) をセットし、RSPI を通信状態にします。

本関数コール後、DMAC データ転送関数 (R\_DMCA\_SetOpr、R\_DMCA\_TransferStart) をコールすることにより、接続デバイスへの DMAC 連動データ送受信ができます。

注意

RSPI が通信停止状態で、本関数をコールすること。

DMAC ドライバについては、「SH7753 グループ アプリケーションノート DMAC ドライバソフトウェア」を参照のこと。

# R\_RSPI\_TxFifoDataSet

データ通信関数

## 送信 FIFO データ設定処理

書式	<pre>#include "r_common.h" #include "iodefine.h" #include "r_rspi_if.h" char_t R_RSPI_TxFifoDataSet( uchar_t *puc_data_src_addr, void (*pv_tx_fifo_empty_ntf_callback ) ( void ), void (*pv_rx_fifo_full_ntf_callback )( void ) );</pre>	
引数	<pre>uchar_t *puc_data_src_addr void (*pv_tx_fifo_empty_ntf_callback ) ( void ) void (*pv_rx_fifo_full_ntf_callback )( void )</pre>	<pre> I   データ格納領域先頭アドレス     (データアクセス (SPLW) 境界) I   送信 FIFO エンプティ通知コールバックアドレス     (NULL の場合、コールバックなし)     ※送受信モード時、“NULL” を設定してください I   受信 FIFO フル通知コールバックアドレス     (NULL の場合、コールバックなし)     ※送信モード時、“NULL” を設定してください</pre>
戻り値	RET_NORMAL	正常終了
解説	<p>本関数は、RSPI 送信 FIFO へのデータ設定処理を行います。データ格納領域の先頭アドレスから、データフレーム数 (SPFC ビット) 分のデータを読み出し、送信 FIFO に設定します。</p> <p>送信モードで本関数コール後、送信 FIFO がエンプティになった場合は、引数 “pv_tx_fifo_empty_ntf_callback” に指定した定義関数をコールすることで通知します。また、送受信モードで、本関数コール後、受信 FIFO がフルになった場合、引数 “pv_rx_fifo_full_ntf_callback” に指定した定義関数をコールすることで通知します。</p> <p>送受信モード時、受信 FIFO フル通知コールバック関数がコールされた後、R_RSPI_RxFifoDataSet関数を使用し、受信 FIFO のデータを取得してください。</p> <p>本関数は汎用 DMAC を使用しません。なお、DMAC 連動の送信 FIFO 設定を使用したい場合、DMAC 連動用通信開始関数 (R_RSPI_ComDmacModeStart) と DMAC データ転送関数 (R_DMCA_SetOpr、R_DMCA_TransferStart) 関数を使用してください。</p>	
注意	<p>本関数をコールする前に、R_RSPI_ComStart関数をコールし、RSPI をデータ通信可能状態にしておくこと。</p> <p>RSPI の送信 FIFO が非エンプティ状態、または送受信モードで受信 FIFO がフル状態で、本関数を使用しないこと。また、R_RSPI_ComDmacModeStart関数をコールした状態で、本関数を使用しないこと。</p> <p>送信データ格納元領域にデータアクセス幅 (SPLW ビット) の設定と合わせて、データを格納すること。</p>	

## R\_RSPI\_RxFifoDataGet

データ通信関数

受信 FIFO データ取得処理

書式	<pre>#include "r_common.h" #include "iodefine.h" #include "r_rspi_if.h" char_t R_RSPI_RxFifoDataGet( uchat_t *puc_data_dst_addr);</pre>			
引数	<table><tr><td>uchat_t *puc_data_dst_addr</td><td>○</td><td>データ格納先領域先頭アドレス (データアクセス (SPLW) 境界)</td></tr></table>	uchat_t *puc_data_dst_addr	○	データ格納先領域先頭アドレス (データアクセス (SPLW) 境界)
uchat_t *puc_data_dst_addr	○	データ格納先領域先頭アドレス (データアクセス (SPLW) 境界)		
戻り値	RET_NORMAL                      正常終了			
解説	<p>本関数は、RSPI 受信 FIFO よりデータ取得処理を行います。受信 FIFO より、データフレーム数 (SPFC ビット) でデータを取得し、データ格納先領域先頭アドレスから格納します。</p> <p>本関数は汎用 DMAC を使用しない。なお、DMAC 連動の受信 FIFO データ取得を使用したい場合、DMAC 連動用 RSPI 通信開処理関数 (R_RSPI_ComDmacModeStart) と DMAC データ転送関数 (R_DMxAC_SetOpr、R_DMxAC_TransferStart) 関数を使用してください。</p>			
注意	RSPI の受信 FIFO が非フル状態で、本関数を使用しないこと。また、R_RSPI_ComDmacModeStart 関数をコールした状態で、本関数を使用しないこと。			

---

## R\_RSPI\_ComStop

データ通信関数

データ通信停止処理

---

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_rspi_if.h"
char_t R_RSPI_ComStop( void );
```

引数

なし

戻り値

RET\_NORMAL                      正常終了

解説

本関数は、RSPI の通信停止処理を行います。

本関数は、RSPI の割り込み有効ビット（エラー割り込み、送信 FIFO エンプティ割り込み、受信 FIFO フル割り込み）及び、機能有効ビット（SPE）をクリアし、RSPI を通信停止状態にします。

データ通信の終了及び、中断する時、本関数を使用してください。

注意

RSPI が通信状態で、本関数をコールすること。



---

## R\_RSPI\_TxInterrupt

割り込みハンドラ関数

送信 FIFO エンプティ割り込みハンドラ

---

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_rspi_if.h"
void R_RSPI_TxInterrupt( void );
```

引数

なし

戻り値

なし

解説

本関数は、TCINT 割り込み内での処理を行います。

“TCINT” 割り込みハンドラから、本割り込みハンドラ関数をコールすることにより、送信 FIFO データ設定関数（R\_RSPI\_TxFifoDataSet）で指定した送信 FIFO エンプティ通知コールバック関数のコール処理を行います。

注意

本関数は、TCINT 割り込みのハンドラ内からコールすること。

---

## R\_RSPI\_RxInterrupt

割り込みハンドラ関数

受信 FIFO フル割り込みハンドラ

---

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_rspi_if.h"
void R_RSPI_RxInterrupt( void );
```

引数

なし

戻り値

なし

解説

本関数は、RXINT 割り込み内での処理を行います。

“RXINT” 割り込みハンドラから、本割り込みハンドラ関数をコールすることにより、送信 FIFO データ設定関数（R\_RSPI\_TxFifoDataSet）で指定した受信 FIFO フル通知コールバック関数のコール処理を行います。

注意

本関数は、RXINT 割り込みのハンドラ内からコールすること。

---

## R\_RSPI\_ErrInterrupt

割り込みハンドラ関数

エラー割り込みハンドラ

---

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_rspi_if.h"
void R_RSPI_ErrInterrupt( void );
```

引数           なし

戻り値       なし

解説           本関数は、ERRINT 割り込み内での処理を行います。  
“ERRINT” 割り込みハンドラから、本割り込みハンドラ関数をコールすることにより、エラーステータスを確認し、エラーコードを引数として、R\_RSPI\_ComStart関数で指定したエラー発生通知コールバック関数をコールします。

注意           本関数は、ERRINT 割り込みのハンドラ内からコールすること。

---

## R\_RSPI\_IdleInterrupt

割り込みハンドラ関数

エラー割り込みハンドラ

---

書式	<pre>#include "r_common.h" #include "iodefine.h" #include "r_rspi_if.h" void R_RSPI_IdleInterrupt( void );</pre>
引数	なし
戻り値	なし
解説	本関数は、IDINT 割り込み内での処理を行います。
注意	本ドライバでは、アイドル割り込みを使用しません。

## 4. 動作条件

### 4.1 ポートについて

本デバイスドライバ使用時は、各ポートの有効状態で使用すること。表 4に有効の設定が必要ポートを示します。

表 4. ポート設定一覧

GPIO レジスタ	ビット	設定
PVCR	PV6MD	0 : RSPI-MOSI 機能を設定します
	PV5MD	0 : RSPI-MISO 機能を設定します
	PV4MD	0 : RSPI-RSPCK 機能を設定します
	PH3MD	0 : RSPI-SSL0 機能を設定します
	PH2MD	0 : RSPI-SSL1 機能を設定します
PSEL6	PU6SEL	1 : RSPI-MOSI 機能を設定します
	PU5SEL	1 : RSPI-MISO 機能を設定します
	PU4SEL	1 : RSPI-RSPCK 機能を設定します
	PU3SEL	1 : RSPI-SSL0 機能を設定します
	PU2SEL	1 : RSPI-SSL1 機能を設定します

### 4.2 割り込みについて

本デバイスドライバ使用時は、各割り込み要因許可状態で使用すること。また、割り込みベクタテーブルに各割り込みハンドラ関数を登録すること。表 1に登録が必要な割り込みハンドラ関数を示します。

表 5. 割り込みベクタテーブル設定関数一覧

割り込み要因番号	割り込み要因	割り込みハンドラ関数
0x1DA0	TCINT	R_RSPI_TxInterrupt
0x1DE0	ERRINT	R_RSPI_ErrInterrupt
0x1D80	RXINT	R_RSPI_RxInterrupt
0x1DC0	IDINT	R_RSPI_IdleInterrupt (※)

※アイドル割り込みは、本ドライバでは使用しません。したがって、割り込みベクタテーブルの設定も必須ではありません。

### 4.3 使用メモリ領域について

本デバイスドライバで使用するメモリ領域を表 6に示します。

表 6. メモリ使用量一覧表

内容	セクション	属性	バイト数
プログラムコード	P	code, align=4	3340 バイト
定数データ	C	data, align=4	0 バイト
初期値ありデータ	D	data, align=4	0 バイト
初期値なしデータ	B	data, align=4	16 バイト

## 5. 付録

## 5.1 ドライバ関数の使用例

RSPI ドライバの使用例を図 1～図 4に示します。

## 5.1.1 SPI フラッシュリードアクセス時の使用例

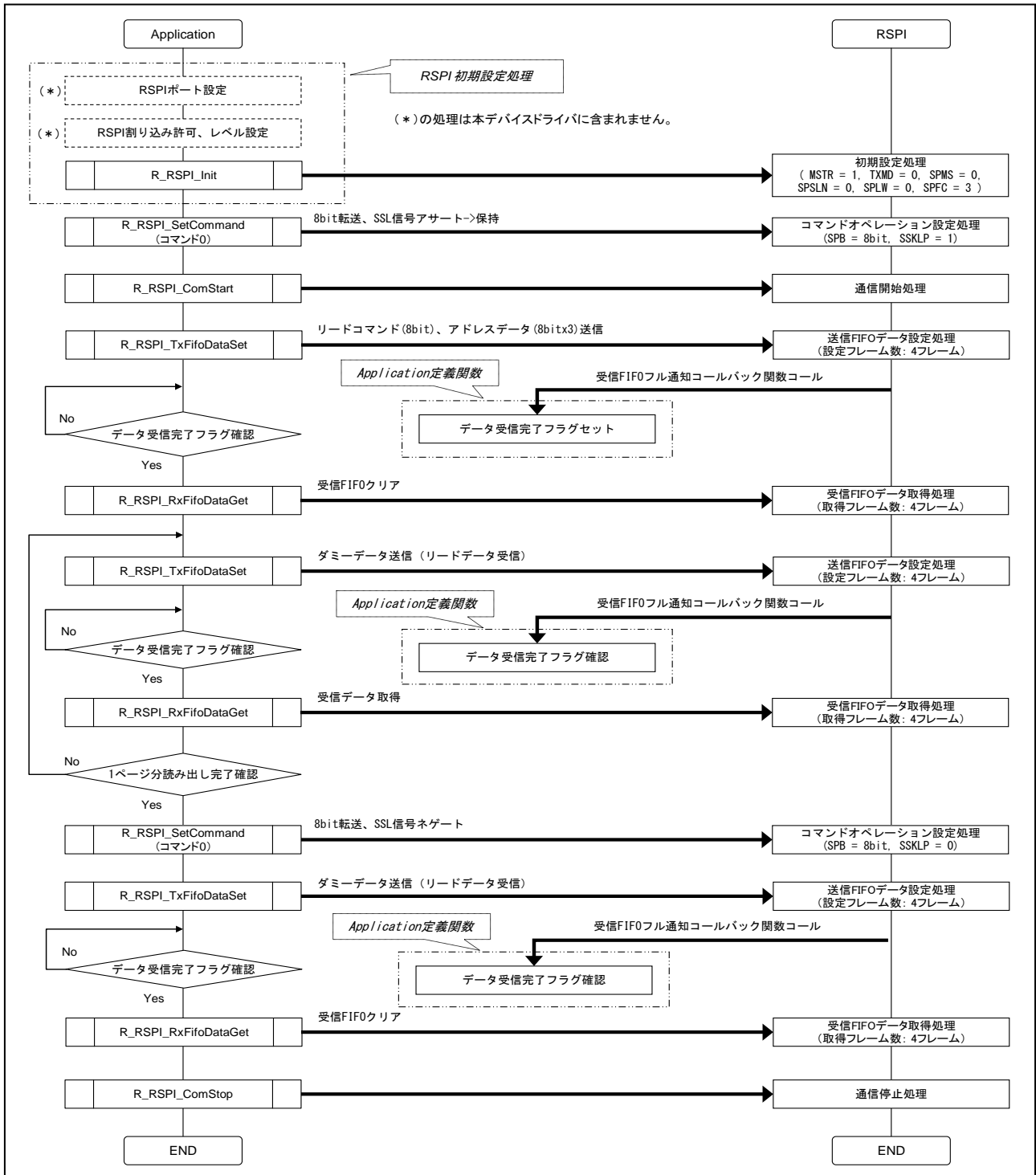


図 1. SPI フラッシュリードアクセス時の使用例

5.1.2 SPI フラッシュページライトアクセス時の使用例

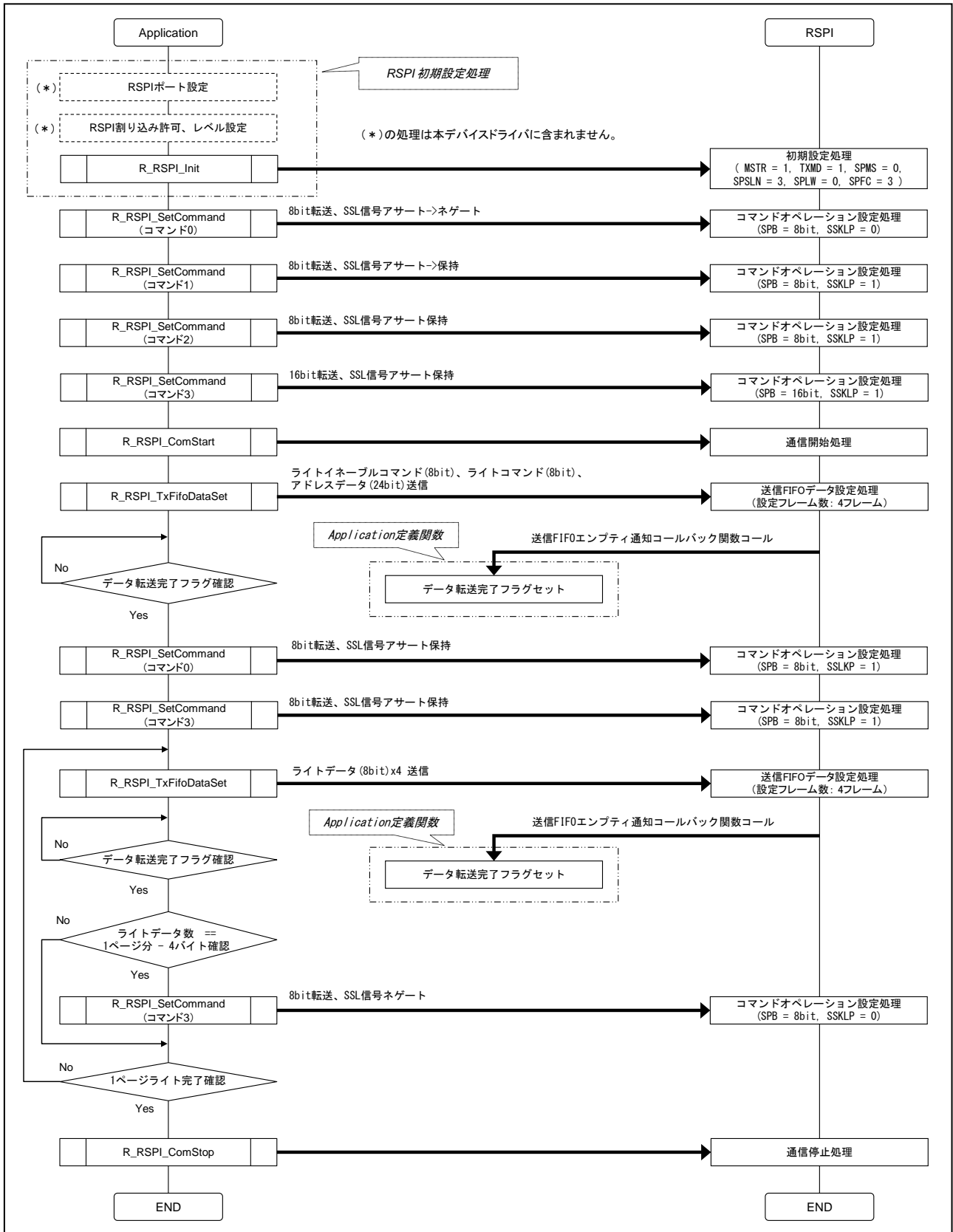


図 2. SPI フラッシュページライトアクセス時の使用例

5.1.3 SPI フラッシュリードアクセス時の使用例 (DMAC 連動)

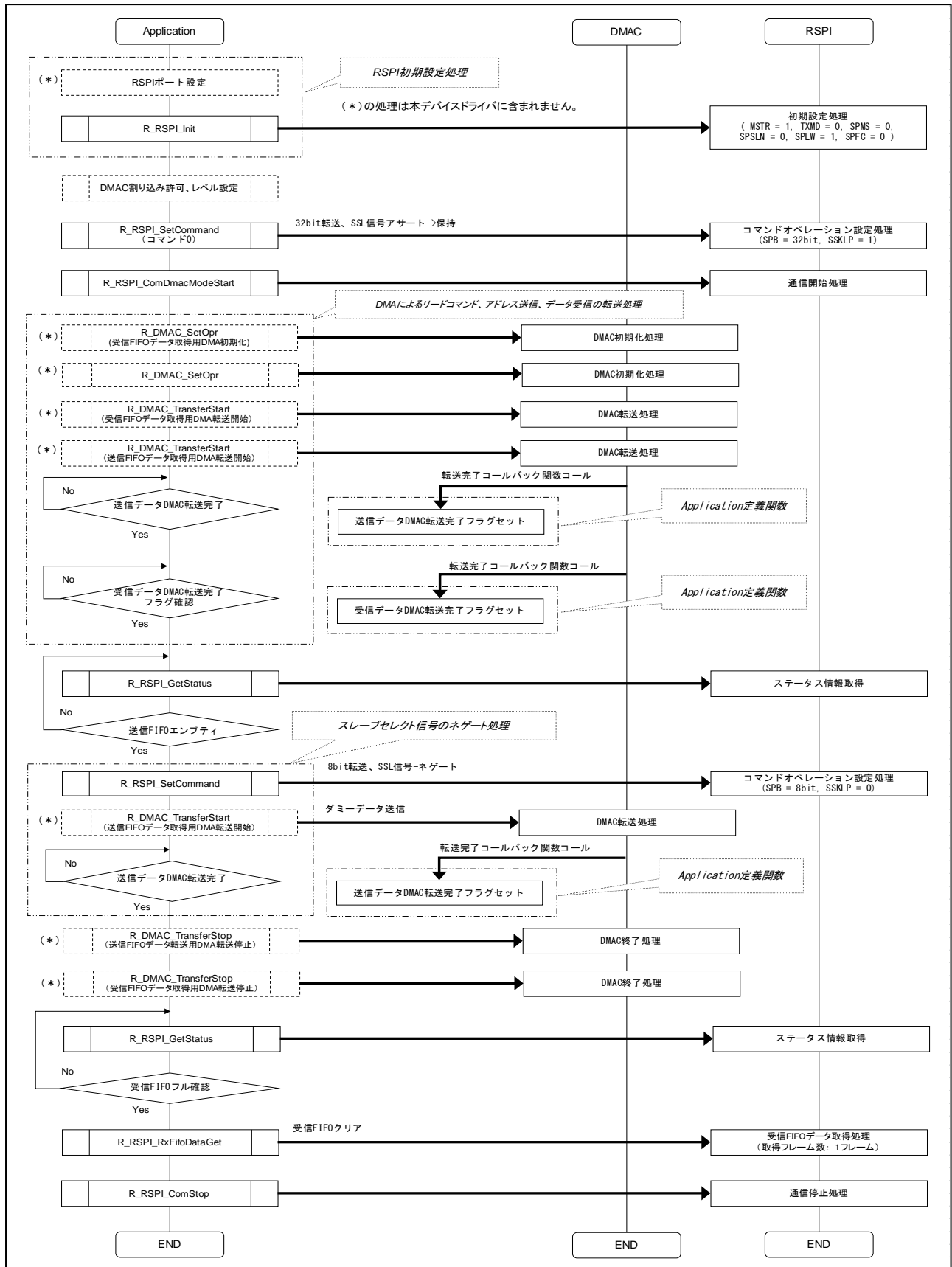


図 3. SPI フラッシュリードアクセス時の使用例 (DMAC 連動)



5.1.4 SPI フラッシュページライトアクセス時の使用例 (DMAC 運動)

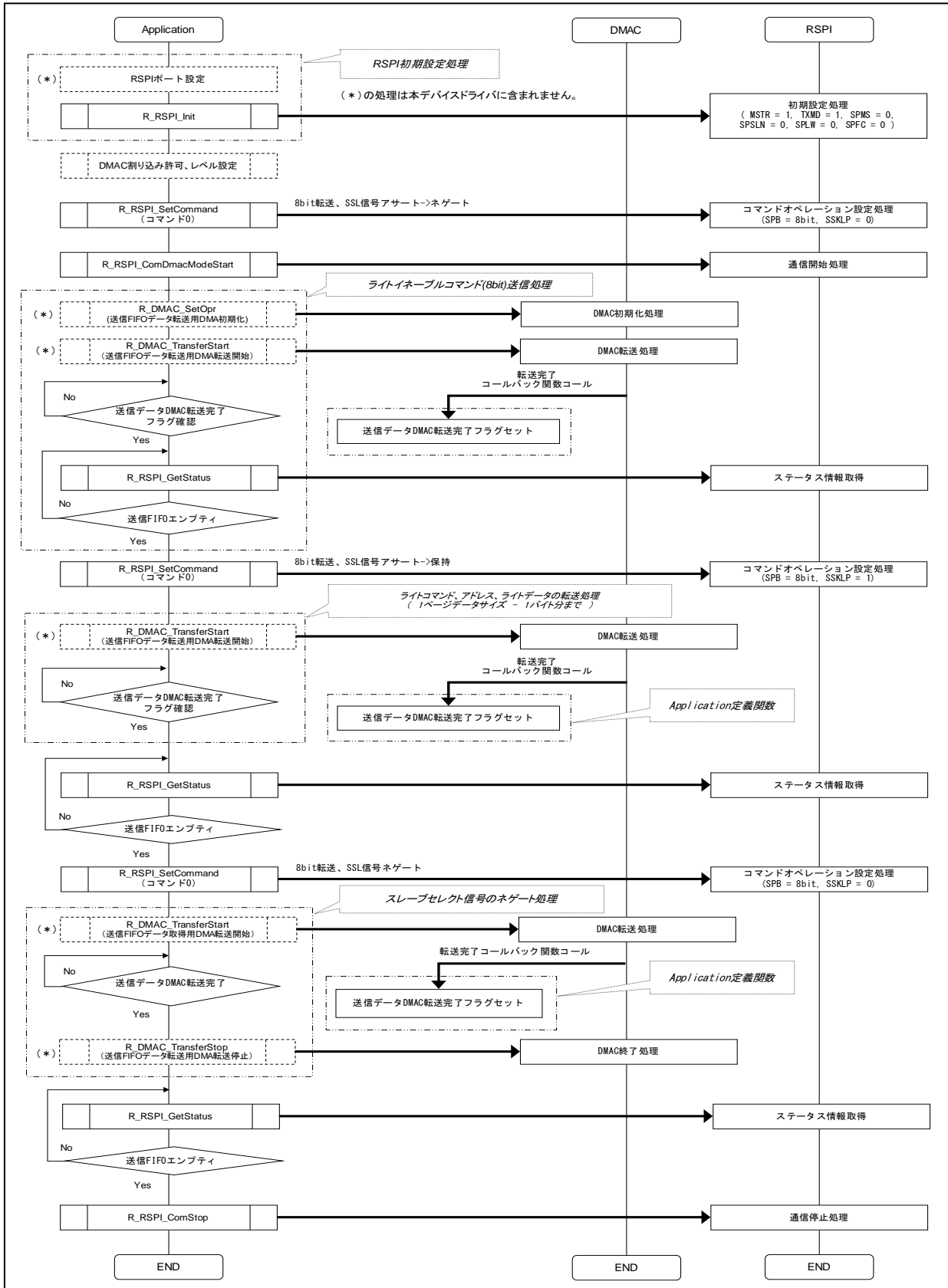


図 4. SPI フラッシュページライトアクセス時の使用例 (DMAC 運動)

## 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

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.7.1	—	初版発行

## 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

- 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、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
- 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
- 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に関し、当社は、何らの責任を負うものではありません。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。
- 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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。
- 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）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じて、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
- 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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。
- 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、当社製品は耐放射線設計については行っていません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、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。
- 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。お客様が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。
- 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情報に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
- お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
- 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。

- 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その総株主の議決権の過半数を直接または間接に保有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。



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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町 2-6-2 (日本ビル)

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： <http://japan.renesas.com/contact/>