



## 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                  | 5  |
| 2.3   | エラーコード一覧                      | 5  |
| 3.    | 関数                            | 6  |
|       | R_SGPIO_SetOpr                | 7  |
|       | R_SGPIO_SetClockSpeed         | 8  |
|       | R_SGPIO_SetMaskPtn            | 9  |
|       | R_SGPIO_SetBitPtn             | 10 |
|       | R_SGPIO_ClrBitmatch           | 11 |
|       | R_SGPIO_GetStatus             | 12 |
|       | R_SGPIO_SingleTransferStart   | 13 |
|       | R_SGPIO_ContTransferStart     | 14 |
|       | R_SGPIO_ContTransferStop      | 15 |
|       | R_SGPIO_SetTrsData            | 16 |
|       | R_SGPIO_GetRcvData            | 17 |
|       | R_SGPIOOn_Interrupt(n=0 to 2) | 18 |
| 4.    | 動作条件                          | 19 |
| 4.1   | ポートについて                       | 19 |
| 4.2   | 割り込みについて                      | 19 |
| 4.3   | 使用メモリ領域について                   | 20 |
| 5.    | 付録                            | 21 |
| 5.1   | ドライバ関数の使用例                    | 21 |
| 5.1.1 | SGPIO単発転送時の使用例                | 21 |
| 5.1.2 | SGPIOビット一致機能の処理例              | 22 |

## 1. 機能概要

本ドライバは SH7752 に搭載されている SGPIO を使用し、汎用シフトレジスタ IC（74LV595A、74LV165A）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

SGPIO ドライバの主な機能を以下に示します。

- ・転送有効/クロック/ビット一致機能設定
- ・ステータス情報取得
- ・単発転送開始
- ・連続転送開始/停止
- ・送信データ設定
- ・受信データ取得

### 1.2 関連ドキュメント

SH7752 グループ ユーザーズマニュアル：ハードウェア編 Rev.1.00

## 2. ドライバ仕様

## 2.1 関数一覧

表 1に、SGPIOドライバの関数一覧を示します。

表 1. SGPIOドライバ関数一覧

| 分類             | 関数名                         | 機能概要               | スタック<br>サイズ |
|----------------|-----------------------------|--------------------|-------------|
| 設定関数           | R_SGPIO_SetOpr              | 転送有効設定処理           | 4 バイト       |
|                | R_SGPIO_SetClockSpeed       | SGPIO クロック設定処理     | 4 バイト       |
|                | R_SGPIO_SetMaskPtn          | ビット一致機能マスクパターン設定処理 | 4 バイト       |
|                | R_SGPIO_SetBitPtn           | ビット一致機能ビットパターン設定処理 | 4 バイト       |
|                | R_SGPIO_ClrBitmatch         | ビット一致レジスタクリア処理     | 12 バイト      |
| 情報取得関数         | R_SGPIO_GetStatus           | ステータス情報取得処理        | 12 バイト      |
| データ通信関数        | R_SGPIO_SingleTransferStart | 単発転送開始処理           | 8 バイト       |
|                | R_SGPIO_ContTransferStart   | 連続転送開始処理           | 20 バイト      |
|                | R_SGPIO_ContTransferStop    | 連続転送停止処理           | 4 バイト       |
|                | R_SGPIO_SetTrsData          | 送信データ設定処理          | 4 バイト       |
|                | R_SGPIO_GetRcvData          | 受信データ取得処理          | 4 バイト       |
| 割り込みハンドラ<br>関数 | R_SGPIOOn_Interrupt         | SGPIO 割り込みハンドラ     | 8 バイト       |

## 2.2 コールバック関数仕様一覧

表 2にSGPIOドライバのコールバック関数仕様一覧を示します。SGPIO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. SPI1 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_SGPIO_SingleTransferStart | void (*pv_xfer_end_callback)( void ) | 転送完了時のコールバック。<br>(NULL の場合、コールバックなし)    |
| R_SGPIO_ContTransferStart   | void (*pv_bitmatch_callback)( void ) | ビット一致検出時のコールバック。<br>(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SGPIOドライバのエラーコード一覧を示します。

表 3. SGPIO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. 関数

本章では、SGPIO 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てください。 |    |
| 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。  |    |
| 注意事項 | 注意事項があればここに示します。   |    |

## R\_SGPIO\_SetOpr

SGPIO 設定関数

SGPIO 転送有効設定処理

|                     |  |  |      |  |                     |                |                               |
|---------------------|--|--|------|--|---------------------|----------------|-------------------------------|
| 書式                  | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_SetOpr( uchar_t uc_ch_num, uchar_t uc_trans_en );</pre>  |  |      |  |                     |                |                               |
| 引数                  | <table><tr><td>uchar_t uc_ch_num</td><td> </td><td>SGPIO のチャンネル番号<br/>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)</td></tr><tr><td>uchar_t uc_trans_en</td><td> </td><td>転送有効設定<br/>(0 : 転送無効、1 : 転送有効)</td></tr></table> | uchar_t uc_ch_num                                    |      | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) | uchar_t uc_trans_en |                | 転送有効設定<br>(0 : 転送無効、1 : 転送有効) |
| uchar_t uc_ch_num   |  | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) |      |  |                     |                |                               |
| uchar_t uc_trans_en |  | 転送有効設定<br>(0 : 転送無効、1 : 転送有効)                        |      |  |                     |                |                               |
| 戻り値                 | <table><tr><td>RET_NORMAL</td><td>正常終了</td></tr><tr><td>RET_ERR_PARAM1</td><td>第 1 引数不正</td></tr><tr><td>RET_ERR_PARAM2</td><td>第 2 引数不正</td></tr></table>   | RET_NORMAL   | 正常終了 | RET_ERR_PARAM1                                       | 第 1 引数不正            | RET_ERR_PARAM2 | 第 2 引数不正                      |
| RET_NORMAL          | 正常終了   |  |      |  |                     |                |                               |
| RET_ERR_PARAM1      | 第 1 引数不正   |  |      |  |                     |                |                               |
| RET_ERR_PARAM2      | 第 2 引数不正   |  |      |  |                     |                |                               |
| 解説                  | <p>本関数は、SGPIO の転送有効設定を行います。</p> <p>SGPIO が汎用シフトレジスタ IC とデータ通信中に、本関数で転送有効設定を無効にした場合、データ通信を強制終了することが出来ます。</p>  |  |      |  |                     |                |                               |
| 注意                  | なし   |  |      |  |                     |                |                               |

# R\_SGPIO\_SetClockSpeed

SGPIO 設定関数

SGPIO クロック設定処理

|                        |   |  |      |  |                        |                |  |
|------------------------|---|--|------|--|------------------------|----------------|--|
| 書式                     | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_SetClockSpeed( uchar_t uc_ch_num, uchar_t uc_sclock_freq );</pre>   |  |      |  |                        |                |  |
| 引数                     | <table><tr><td>uchar_t uc_ch_num</td><td> </td><td>SGPIO のチャンネル番号<br/>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)</td></tr><tr><td>uchar_t uc_sclock_freq</td><td> </td><td>SGPIO クロック周波数<br/>H'0 : 24MHz<br/>H'1 : 12MHz<br/>H'2 : 6MHz<br/>H'3 : 3MHz<br/>H'4 : 1.5MHz<br/>H'5 : 750kHz<br/>H'6 : 375kHz<br/>H'7 : 187.5kHz<br/>H'8 : 93.75kHz<br/>H'F : 48MHz</td></tr></table> | uchar_t uc_ch_num  |      | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) | uchar_t uc_sclock_freq |                | SGPIO クロック周波数<br>H'0 : 24MHz<br>H'1 : 12MHz<br>H'2 : 6MHz<br>H'3 : 3MHz<br>H'4 : 1.5MHz<br>H'5 : 750kHz<br>H'6 : 375kHz<br>H'7 : 187.5kHz<br>H'8 : 93.75kHz<br>H'F : 48MHz |
| uchar_t uc_ch_num      |   | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)   |      |  |                        |                |  |
| uchar_t uc_sclock_freq |   | SGPIO クロック周波数<br>H'0 : 24MHz<br>H'1 : 12MHz<br>H'2 : 6MHz<br>H'3 : 3MHz<br>H'4 : 1.5MHz<br>H'5 : 750kHz<br>H'6 : 375kHz<br>H'7 : 187.5kHz<br>H'8 : 93.75kHz<br>H'F : 48MHz |      |  |                        |                |  |
| 戻り値                    | <table><tr><td>RET_NORMAL</td><td>正常終了</td></tr><tr><td>RET_ERR_PARAM1</td><td>第 1 引数不正</td></tr><tr><td>RET_ERR_PARAM2</td><td>第 2 引数不正</td></tr></table>  | RET_NORMAL   | 正常終了 | RET_ERR_PARAM1                                       | 第 1 引数不正               | RET_ERR_PARAM2 | 第 2 引数不正   |
| RET_NORMAL             | 正常終了  |  |      |  |                        |                |  |
| RET_ERR_PARAM1         | 第 1 引数不正  |  |      |  |                        |                |  |
| RET_ERR_PARAM2         | 第 2 引数不正  |  |      |  |                        |                |  |
| 解説                     | 本関数は、SGPIO のクロック周波数設定を行います。   |  |      |  |                        |                |  |
| 注意                     | <p>SGPIO の転送有効設定が有効であるときに、本関数をコールしても SGPIO クロック周波数は更新されません。</p> <p>本関数をコールする際は、R_SGPIO_GetStatus関数で転送有効設定が無効であることを確認してから実行してください。</p>   |  |      |  |                        |                |  |

# R\_SGPIO\_SetMaskPtn

SGPIO 設定関数

SGPIO ビット一致機能マスクパターン設定処理

|     |  |  |
|-----|--|--|
| 書式  | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_SetMaskPtn( uchar_t uc_ch_num, uchar_t uc_sel_mask_reg,     ulong_t ul_mask_ptn );</pre>   |  |
| 引数  | <pre>uchar_t uc_ch_num uchar_t uc_sel_mask_reg ulong_t ul_mask_ptn</pre>   | <pre>  SGPIO のチャンネル番号     (0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)   ビット一致機能のマスクパターンを設定するレジスタを選択     0 : GPILATCHEN0     1 : GPILATCHEN1     2 : GPILATCHEN2     3 : GPILATCHEN3   ビット一致機能のマスクパターン</pre> |
| 戻り値 | <pre>RET_NORMAL RET_ERR_PARAM1 RET_ERR_PARAM2</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、ビット一致機能のマスクパターン設定を行います。その後、R_SGPIO_ContTransferStart関数を実行することで、SGPIOの受信データに対してビット一致機能を実行します。</p> <p>マスクパターンの設定にあたっては、任意の GPI ラッチ有効レジスタのみを設定できるように、GPIラッチ有効レジスタを選択するようになっています。4バイト以上のマスクパターンを設定したい場合は、本関数を複数回コールして繰り返し設定を行ってください。</p> |  |
| 注意  | <p>SGPIO が汎用シフトレジスタ IC とデータ通信中に、本関数をコールしないでください。</p> <p>本関数をコールする際は、R_SGPIO_GetStatus関数で転送アイドル状態であることを確認してから実行してください。また、必ずR_SGPIO_ContTransStart関数を実行する前にコールしてください。</p> <p>本関数を一度もコールせずにR_SGPIO_ContTransferStartを実行した場合、全ビットがマスクされます。</p>   |  |

## R\_SGPIO\_SetBitPtn

SGPIO 設定関数

SGPIO ビット一致機能ビットパターン設定処理

|     |  |  |
|-----|--|--|
| 書式  | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_SetBitPtn( uchar_t uc_ch_num, uchar_t uc_sel_bit_reg, ulong_t ul_bit_ptn );</pre>  |  |
| 引数  | <pre>uchar_t uc_ch_num uchar_t uc_sel_bit_reg ulong_t ul_bit_ptn</pre>   | <pre>  SGPIO のチャンネル番号       (0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)   ビット一致機能のビットパターンを設定するレジスタを選択       0 : GPIPOLARITY0       1 : GPIPOLARITY1       2 : GPIPOLARITY2       3 : GPIPOLARITY3   ビット一致機能のビットパターン</pre> |
| 戻り値 | <pre>RET_NORMAL RET_ERR_PARAM1 RET_ERR_PARAM2</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、ビット一致機能のビットパターン設定を行います。その後、R_SGPIO_ContTransferStart関数を実行することで、SGPIOの受信データに対してビット一致機能を実行します。</p> <p>ビットパターンの設定にあたっては、任意の GPI 極性レジスタのみを設定できるように、GPI 極性レジスタを選択するようになっています。4 バイト以上のビットパターンを設定したい場合は、本関数を複数回コールして繰り返し設定を行ってください。</p>       |  |
| 注意  | <p>SGPIO が汎用シフトレジスタ IC とデータ通信中に、本関数をコールしないでください。</p> <p>本関数をコールする際は、R_SGPIO_GetStatus関数で転送アイドル状態であることを確認してから実行してください。また、必ずR_SGPIO_ContTransStart関数を実行する前にコールしてください。</p> <p>本関数を一度もコールせずにR_SGPIO_ContTransferStartを実行した場合、全ビットの極性が 0 となります。</p> |  |

## R\_SGPIO\_ClrBitmatch

SGPIO 設定関数

SGPIO ビット一致レジスタクリア処理

|                   |  |  |      |  |          |
|-------------------|--|--|------|--|----------|
| 書式                | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_(uchar_t uc_ch_num);</pre>         |  |      |  |          |
| 引数                | <table><tr><td>uchar_t uc_ch_num</td><td> </td><td>SGPIO のチャンネル番号<br/>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)</td></tr></table> | uchar_t uc_ch_num                                    |      | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) |          |
| uchar_t uc_ch_num |  | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) |      |  |          |
| 戻り値               | <table><tr><td>RET_NORMAL</td><td>正常終了</td></tr><tr><td>RET_ERR_PARAM1</td><td>第 1 引数不正</td></tr></table>                  | RET_NORMAL   | 正常終了 | RET_ERR_PARAM1                                       | 第 1 引数不正 |
| RET_NORMAL        | 正常終了   |  |      |  |          |
| RET_ERR_PARAM1    | 第 1 引数不正   |  |      |  |          |
| 解説                | 本関数はビット一致機能を無効にした上で、SGPIO のビット一致レジスタをクリアします。また、ビット一致検出の割り込み要因もクリアされます。   |  |      |  |          |
| 注意                | SGPIO が汎用シフトレジスタ IC とデータ通信中に、本関数をコールしないでください。<br>本関数をコールする際は、R_SGPIO_GetStatus関数で転送アイドル状態であることを確認してから実行してください。             |  |      |  |          |

# R\_SGPIO\_GetStatus

SGPIO 情報取得関数

SGPIO ステータス情報取得処理

|                   |  |  |         |  |          |
|-------------------|--|--|---------|--|----------|
| 書式                | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_GetStatus( uchar_t uc_ch_num );</pre>  |  |         |  |          |
| 引数                | <table><tr><td>uchar_t uc_ch_num</td><td>I</td><td>SGPIO のチャンネル番号<br/>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)</td></tr></table>   | uchar_t uc_ch_num                                    | I       | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) |          |
| uchar_t uc_ch_num | I  | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) |         |  |          |
| 戻り値               | <table><tr><td>0~H'7F</td><td>ステータス情報</td></tr><tr><td>RET_ERR_PARAM1</td><td>第 1 引数不正</td></tr></table>   | 0~H'7F   | ステータス情報 | RET_ERR_PARAM1                                       | 第 1 引数不正 |
| 0~H'7F            | ステータス情報  |  |         |  |          |
| RET_ERR_PARAM1    | 第 1 引数不正   |  |         |  |          |
| 解説                | <p>本関数は、SGPIO のステータス情報の取得を行います。<br/>本関数のコールで取得可能なステータスは、以下の通りです。</p> <p><b>【SGPIO ステータス情報詳細】</b></p> <p>SGPIO クロック周波数 : bit[3:0]</p> <ul style="list-style-type: none"><li>H'0 : 24MHz</li><li>H'1 : 12MHz</li><li>H'2 : 6MHz</li><li>H'3 : 3MHz</li><li>H'4 : 1.5MHz</li><li>H'5 : 750kHz</li><li>H'6 : 375kHz</li><li>H'7 : 187.5kHz</li><li>H'8 : 93.75kHz</li><li>H'F : 48MHz</li></ul> <p>転送有効設定 : bit[4]</p> <ul style="list-style-type: none"><li>0 : 無効</li><li>1 : 有効</li></ul> <p>転送状態 : bit[5]</p> <ul style="list-style-type: none"><li>0 : アイドル状態</li><li>1 : 転送中</li></ul> <p>転送モード設定 : bit[6]</p> <ul style="list-style-type: none"><li>0 : 単発転送モード</li><li>1 : 連続転送モード</li></ul> |  |         |  |          |
| 注意                | なし   |  |         |  |          |

# R\_SGPIO\_SingleTransferStart

SGPIO データ通信関数

SGPIO 単発転送開始処理

```
書式      #include "r_common.h"
          #include "r_sh7752.h"
          #include "r_sgpio_if.h"
          char_t R_SGPIO_SingleTransferStart( uchar_t uc_ch_num, uchar_t uc_xfer_cnt,
          void (*pv_xfer_end_callback)(void) );
```

|    |                                    |  |  |
|----|------------------------------------|--|--|
| 引数 | uchar_t uc_ch_num                  |  | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) |
|    | uchar_t uc_xfer_cnt                |  | 送受信データのビット数(2~128)                                   |
|    | void (*pv_xfer_end_callback)(void) |  | 転送完了時のコールバックアドレス<br>(NULL の場合、コールバックなし)              |

|     |                |          |
|-----|----------------|----------|
| 戻り値 | RET_NORMAL     | 正常終了     |
|     | RET_ERR_PARAM1 | 第 1 引数不正 |
|     | RET_ERR_PARAM2 | 第 2 引数不正 |

**解説**            本関数は受信データレジスタのクリアを行なった上で、単発転送モードで汎用シフトレジスタ IC とデータ送受信を開始します。

                  本関数はコールバックする場合のみ転送完了割り込みを有効にし、その上で引数"uc\_xfer\_cnt"に指定したビット数を 1 ビットストリームとして単発転送の開始を行います。

**注意**            SGPIO が汎用シフトレジスタ IC とデータ通信中、または転送有効設定が無効であるときに、本関数をコールしないでください。

                  本関数をコールする際は、R\_SGPIO\_GetStatus関数で転送有効設定が有効、かつ転送アイドル状態であることを確認してから実行してください。

# R\_SGPIO\_ContTransferStart

SGPIO データ通信関数

SGPIO 連続転送開始処理

|     |   |   |
|-----|---|---|
| 書式  | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_ContTransferStart( uchar_t uc_ch_num, uchar_t uc_xfer_cnt, void(*pv_bitmatch_callback)(void) );</pre> |   |
| 引数  | <pre>uchar_t uc_ch_num uchar_t uc_xfer_cnt void (*pv_bitmatch_callback)(void)</pre>   | <pre>  SGPIO のチャンネル番号   (0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)   送受信データのビット数(2~128)   ビット一致検出時のコールバックアドレス   (NULL の場合、コールバックなし)</pre> |
| 戻り値 | <pre>RET_NORMAL RET_ERR_PARAM1 RET_ERR_PARAM2</pre>   | <pre>正常終了 第 1 引数不正 第 2 引数不正</pre>   |

**解説**

本関数はビット一致機能を使用した連続転送を行なうのに使用します。  
 本関数は受信データレジスタのクリアを行なった上で、ビット一致機能を有効にし、連続転送モードで汎用シフトレジスタ IC とデータ送受信を開始します。  
 またコールバックする場合のみビット一致割込みを有効にし、その上で引数"uc\_xfer\_cnt"に指定したビット数を 1 ビットストリームとした連続転送の開始を行います。連続転送を停止させるには、R\_SGPIO\_ContTransferStop関数をコールする必要があります。

**注意**

SGPIO が汎用シフトレジスタ IC とデータ通信中、または転送有効設定が無効であるときに、本関数をコールしないでください。  
 本関数をコールする際は、R\_SGPIO\_GetStatus関数で転送有効設定が有効、かつ転送アイドル状態であることを確認してから実行してください。

## R\_SGPIO\_ContTransferStop

SGPIO データ通信関数

SGPIO 連続転送停止処理

書式

```
#include "r_common.h"
#include "r_sh7752.h"
#include "r_sgpio_if.h"
char_t R_SGPIO_ContTransferStop( uchar_t uc_ch_num );
```

引数

|                   |   |  |
|-------------------|---|--|
| uchar_t uc_ch_num | I | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2) |
|-------------------|---|--|

戻り値

|                |          |
|----------------|----------|
| RET_NORMAL     | 正常終了     |
| RET_ERR_PARAM1 | 第 1 引数不正 |

解説

本関数は、汎用シフトレジスタ IC との連続転送の停止処理を実行します。

本関数をコールしても、転送中のビットストリームは最終ビットが転送されるまで止まりません。このため、本関数コール後に R\_SGPIO\_GetStatus 関数で転送アイドル状態になったことを確認してください。

注意

なし

# R\_SGPIO\_SetTrsData

SGPIO データ通信関数

SGPIO 送信データ設定処理

|    |  |  |  |
|----|--|--|--|
| 書式 | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h" char_t R_SGPIO_SetTrsData( uchar_t uc_ch_num, uchar_t uc_sel_trsdata_reg,     ulong_t ul_tx_data );</pre> |  |  |
| 引数 | uchar_t uc_ch_num  |  | SGPIO のチャンネル番号<br>(0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)                             |
|    | uchar_t uc_sel_trsdata_reg   |  | 設定する送信データレジスタの選択<br>0 : TRSDATA0<br>1 : TRSDATA1<br>2 : TRSDATA2<br>3 : TRSDATA3 |
|    | ulong_t ul_tx_data   |  | 送信データパターン  |

|     |                |          |
|-----|----------------|----------|
| 戻り値 | RET_NORMAL     | 正常終了     |
|     | RET_ERR_PARAM1 | 第 1 引数不正 |
|     | RET_ERR_PARAM2 | 第 2 引数不正 |

**解説** 本関数は、SGPIOが汎用シフトレジスタICへ送信するデータの設定を行います。その後、R\_SGPIO\_SingleTransferStart、もしくはR\_SGPIO\_ContTransferStartを実行することで、本関数で設定した送信データが送信されます。

送信データの設定にあたっては、任意の送信データレジスタを設定できるように、送信データレジスタを選択するようになっています。4 バイト以上の送信データを設定したい場合は、本関数を複数回コールして繰り返し設定を行なってください。

**注意** SGPIO が汎用シフトレジスタ IC とデータ通信中に、本関数をコールしないでください。  
本関数をコールする際は、R\_SGPIO\_GetStatus関数で転送アイドル状態であることを確認してから実行してください。  
本関数を一度もコールせずにR\_SGPIO\_SingleTransferStartやR\_SGPIO\_ContTransferStartを実行した場合、送信する全ビットが0となります。

# R\_SGPIO\_GetRcvData

SGPIO データ通信関数

SGPIO 受信データ取得処理

|     |  |  |
|-----|--|--|
| 書式  | <pre>#include "r_common.h" #include "r_sh7752.h" #include "r_sgpio_if.h"  char_t R_SGPIO_GetRcvData( uchar_t uc_ch_num, uchar_t uc_sel_rcvdata_reg,     ulong_t *pul_rx_data_addr );</pre> |  |
| 引数  | <pre>uchar_t uc_ch_num uchar_t uc_sel_rcvdata_reg ulong_t *pul_rx_data_addr</pre>  | <pre>  SGPIO のチャンネル番号     (0 : SGPIO0、1 : SGPIO1、2 : SGPIO2)   取得する受信データレジスタの選択     0 : RCVDATA0     1 : RCVDATA1     2 : RCVDATA2     3 : RCVDATA3   取得した受信データの格納領域先頭アドレス</pre> |
| 戻り値 | <pre>RET_NORMAL RET_ERR_PARAM1 RET_ERR_PARAM2</pre>  | <pre>正常終了 第 1 引数不正 第 2 引数不正</pre>  |

**解説** 本関数は、SGPIO が汎用シフトレジスタ IC から受信したデータの取得を行います。受信データの取得にあたっては、任意の受信データレジスタを読み出せるように、受信データレジスタを選択するようになっています。4 バイト以上の受信データを取得したい場合は、本関数を複数回コールして繰り返し取得を行なってください。

**注意** SGPIO が汎用シフトレジスタ IC とデータ通信中に、本関数をコールしないでください。本関数をコールする際は、R\_SGPIO\_GetStatus関数で転送アイドル状態であることを確認してから実行してください。受信したビット数を超える受信データレジスタを読み出した場合、その値は不定値になります。

---

## R\_SGPIOn\_Interrupt(n=0 to 2)

割り込みハンドラ関数

SGPIO 割り込みハンドラ

---

書式

```
#include "r_common.h"
#include "r_sh7752.h"
#include "r_sgpio_if.h"
void R_SGPIOn_Interrupt( void );
```

引数

なし

戻り値

なし

解説

本関数は、SGPIO 割り込み内での処理を行います。  
本割り込みハンドラ関数で、転送完了・ビット一致検出を確認した場合、該当コールバック関数をコールします。

注意

本関数は、SGPIO012 割り込みのハンドラ内からコールしてください。

## 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してください。表 4に有効の設定が必要なポートを示します。

表 4. ポート設定一覧

| GPIO レジスタ | ビット    | 設定                        |
|-----------|--------|---------------------------|
| PNCR      | PN4MD  | 0 : SGPIO1-CLK 機能を設定する    |
|           | PN3MD  | 0 : SGPIO1-LOAD 機能を設定する   |
|           | PN2MD  | 0 : SGPIO1-DO 機能を設定する     |
|           | PN1MD  | 0 : SGPIO1-DI 機能を設定する     |
| POCR      | PO7MD  | 0 : SGPIO0-CLK 機能を設定する    |
|           | PO6MD  | 0 : SGPIO0-LOAD 機能を設定する   |
|           | PO5MD  | 0 : SGPIO0-DI 機能を設定する     |
|           | PO4MD  | 0 : SGPIO0-DO 機能を設定する     |
|           | PO3MD  | 0 : SGPIO2-CLK 機能を設定する    |
|           | PO2MD  | 0 : SGPIO2-LOAD 機能を設定する   |
|           | PO1MD  | 0 : SGPIO2-DI 機能を設定する     |
|           | PO0MD  | 0 : SGPIO2-DO 機能を設定する     |
| PSEL4     | PN4SEL | 1 : SGPIO1-CLK 機能選択を設定する  |
|           | PN3SEL | 1 : SGPIO1-LOAD 機能選択を設定する |
|           | PN2SEL | 1 : SGPIO1-DO 機能選択を設定する   |
|           | PN1SEL | 1 : SGPIO1-DI 機能選択を設定する   |
| PSEL5     | PO3SEL | 0 : SGPIO2-CLK 機能選択を設定する  |
|           | PO2SEL | 0 : SGPIO2-LOAD 機能選択を設定する |
|           | PO1SEL | 0 : SGPIO2-DI 機能選択を設定する   |
|           | PO0SEL | 0 : SGPIO2-DO 機能選択を設定する   |

### 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してください。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してください。表 5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         |
|----------|--------|--------------------|
| 0x1F80   | SGPIO0 | R_SGPIO0_Interrupt |
| 0x1FA0   | SGPIO1 | R_SGPIO1_Interrupt |
| 0x1FC0   | SGPIO2 | R_SGPIO2_Interrupt |

### 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 | 2148 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 | 8 バイト    |

## 5. 付録

## 5.1 ドライバ関数の使用例

汎用シフトレジスタICとデータ通信時のSGPIOドライバの使用例を図 1～図 2に示します。

## 5.1.1 SGPIO単発転送時の使用例

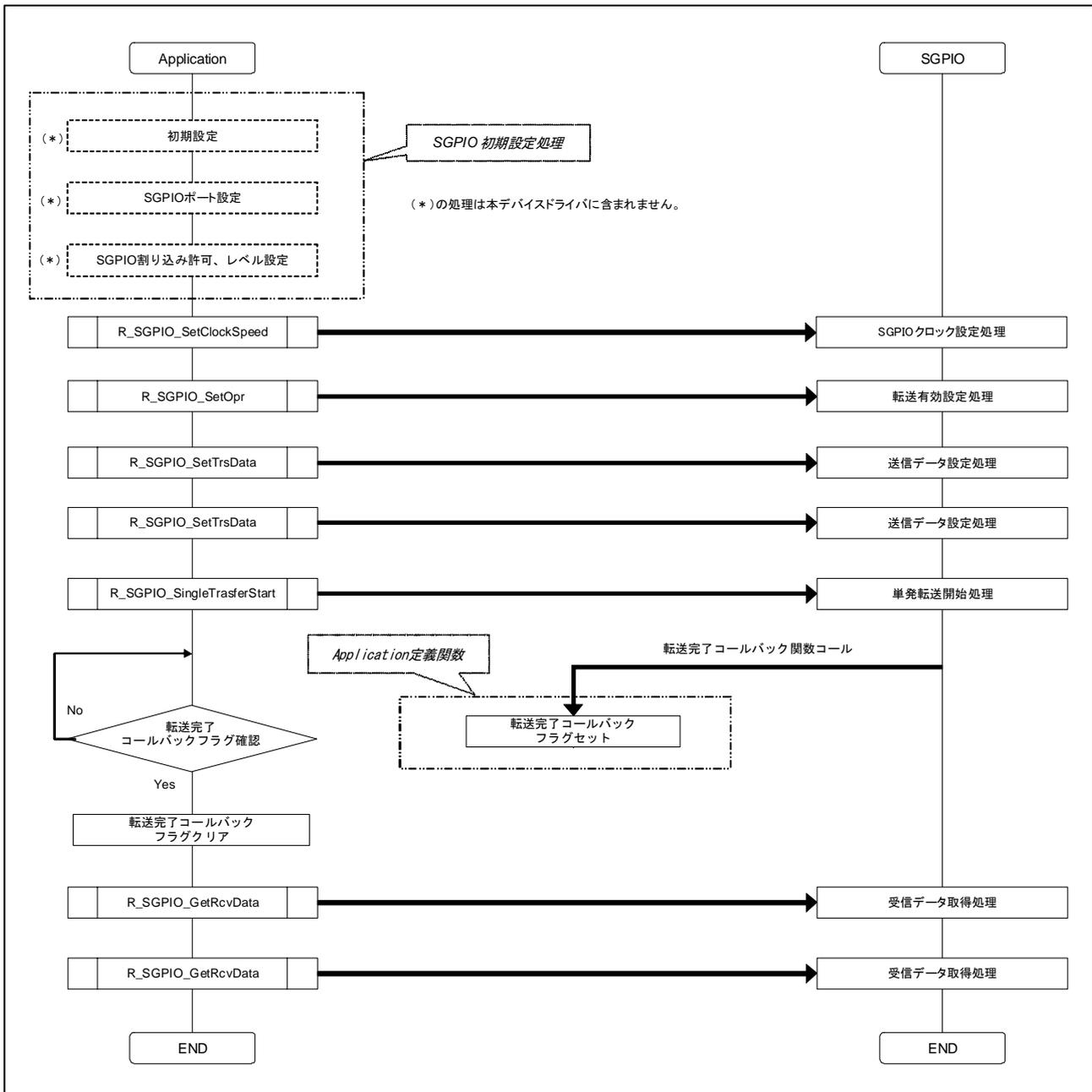


図 1. SGPIO単発転送時(64ビット転送)の使用例

5.1.2 SGPIOビット一致機能の処理例

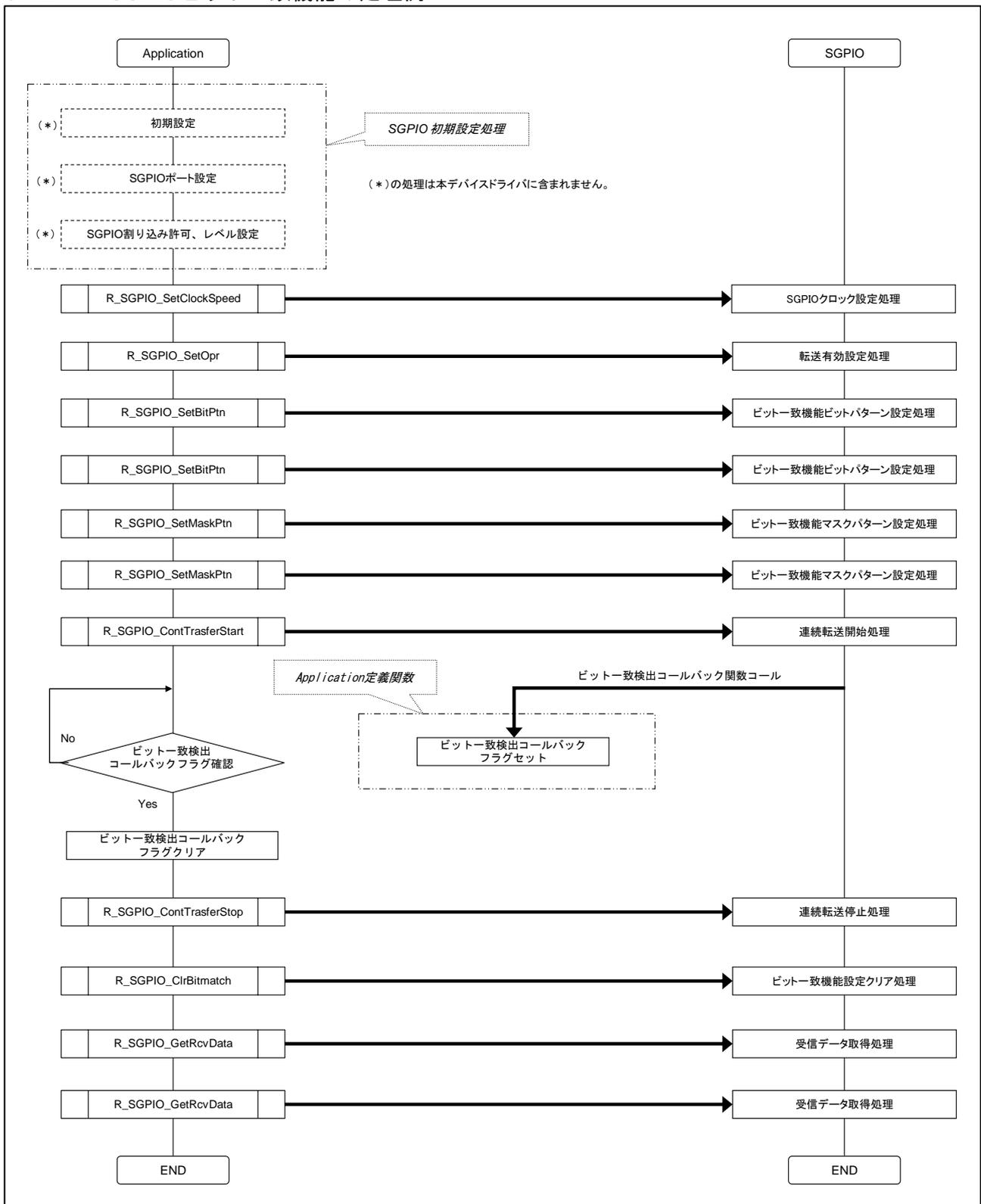


図 2. SGPIO ビット一致機能(64 ビット転送)の処理例

## 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/>

お問い合わせ先

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



## 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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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。

### 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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

## 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、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。
2. 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。
3. 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。
4. 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。
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も、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。
6. 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。
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、当社製品は耐放射線設計については行っておりません。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。
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。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、当社は、一切その責任を負いません。
9. 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行ってください。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。
10. 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、お客様にてご負担して頂きますのでご了承ください。
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。

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と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。



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・電話番号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）

(03)5201-5307

■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/>