

## RX ファミリ

### DSMIF モジュール Firmware Integration Technology

#### 要旨

本アプリケーションノートでは、Firmware Integration Technology (FIT)を使用した DSMIF モジュールについて説明します。本モジュールは DSMIF を使用して、 $\Delta$ - $\Sigma$  変調の高サンプリングレートの 1 ビットデジタルデータストリームをフィルタリングして低サンプリングレートの 16 ビットデジタルデータに変換します。以降、本モジュールを DSMIF FIT モジュールと称します。

#### 対象デバイス

- RX72M 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

## 内容

1. 概要 .....	3
1.1 DSMIF FIT モジュールとは .....	3
2. API 情報 .....	4
2.1 ハードウェアの要求 .....	4
2.2 ハードウェアリソースの要求 .....	4
2.2.1 周辺機能 .....	4
2.2.2 その他の周辺機能 .....	4
2.3 ソフトウェアの要求 .....	4
2.4 サポートされているツールチェーン .....	4
2.5 使用する割り込みベクタ .....	4
2.6 ヘッダファイル .....	5
2.7 整数型 .....	5
2.8 コンパイル時の設定 .....	5
2.9 コードサイズ .....	5
2.10 引数 .....	6
2.11 FIT モジュールの追加方法 .....	8
3. API 関数 .....	9
3.1 R_DSMIF_Create .....	9
3.2 R_DSMIF_Start .....	10
3.3 R_DSMIF_Stop .....	11
3.4 R_DSMIF_SetFilter .....	12
3.5 R_DSMIF_ReadData .....	14
3.6 R_DSMIF_GetErrorStatus .....	15
4. 端子設定 .....	16
5. デモプロジェクト .....	17
5.1 dsmif_demo_rskrx72m .....	17
5.1.1 ボードの設定と接続 .....	17
5.1.2 デモプロジェクト動作手順 .....	20
6. 付録 .....	28
6.1 動作確認環境 .....	28
7. 参考ドキュメント .....	30
改訂記録 .....	31

## 1. 概要

DSMIF FIT モジュールは、DSMIF を使用し、 $\Delta$ - $\Sigma$  変調の 1 ビットデジタルストリームを受信するための手段を提供します。

以下に本モジュールがサポートしている機能を列挙します。

- 1 ビットデジタル入力データをフィルタリングして、16 ビットデジタルデータに変換
- $\Delta$ - $\Sigma$  クロック信号の、マスタ動作およびスレーブ動作に対応
- 過電流、短絡、合計電流エラーの 3 種類のエラー検出に対応

### 1.1 DSMIF FIT モジュールとは

本モジュールは API として、プロジェクトに組み込んで使用します。本モジュールの組み込み方法については「2.11 FIT モジュールの追加方法」を参照してください。

## 2. API 情報

本 FIT モジュールは、下記の条件で動作を確認しています。

### 2.1 ハードウェアの要求

ご使用になる MCU が以下の機能をサポートしている必要があります。

- $\Delta$ - $\Sigma$  モジュレータインターフェース(DSMIF)

### 2.2 ハードウェアリソースの要求

本 FIT モジュールに必要なハードウェアリソースについて説明します。明示的に記載していない限り、これらのリソースは他の周辺機能では使用できません。

#### 2.2.1 周辺機能

$\Delta$ - $\Sigma$  モジュレータインターフェース(DSMIF)

#### 2.2.2 その他の周辺機能

DSMIF のクロック送受信とデータ受信に 2 本の I/O ポートを割り当てる必要があります。割り当たされたポートは汎用入出力ポートとして使用できません。

### 2.3 ソフトウェアの要求

FIT モジュールは以下の FIT モジュールに依存しています。

- ボードサポートパッケージモジュール(r\_bsp) v5.20 以上
- CMT モジュール(r\_cmt) v3.40 以上

### 2.4 サポートされているツールチェーン

本 FIT モジュールは「6.1 動作確認環境」に示すツールチェーンで動作確認を行っています。

### 2.5 使用する割り込みベクタ

DSMIF の割り込みは、R\_DSMIF\_Start 関数を実行することで有効化されます。

表 2.1 使用する割り込みベクター一覧

デバイス	割り込みベクタ
RX72M	GROUPBL2(ベクタ番号 : 107) OCDI0 (過電流検出割り込み) : ビット 1 SUMEI0 (合計電流エラー割り込み) : ビット 2 SCDI0 (短絡検出割り込み) : ビット 3 OCDI1 (過電流検出割り込み) : ビット 4 SUMEI1 (合計電流エラー割り込み) : ビット 5 SCDI1 (短絡検出割り込み) : ビット 6

## 2.6 ヘッダファイル

すべての API 呼び出しとそれをサポートするインターフェース定義は `r_dsmif_rx_if.h` に記載しています。

ビルド時に設定可能なコンフィギュレーションオプションは `r_dsmif_rx_config.h` ファイルで選択または定義されています。

本 FIT モジュールの API をユーザプログラムから参照するには、`r_dsmif_rx_if.h` をインクルードしてください。

## 2.7 整数型

このドライバは ANSI C99 を使用しています。これらの型は `stdint.h` で定義されています。

## 2.8 コンパイル時の設定

本モジュールのコンフィギュレーションオプションの設定は、`r_dsmif_rx_config.h` で行います。

オプション名および設定値に関する説明を、下表に示します。

コンフィギュレーションオプション( <code>r_dsmif_rx_config.h</code> )	
<code>DSMIF_ERR_INTR_PRI_LVL</code> ※デフォルト値は"15"	GROUPBL2 割り込みで使用される割り込み優先レベル
<code>DSMIF_SETTING_WAIT_CNT</code> ※デフォルト値は"250"	フィルタ動作開始後の出力安定待ち時間(単位:usec)
<code>DSMIF_OCTLR_VALUE</code> ※デフォルト値は"0"	過電流検出条件の設定に使用する有効電流値の下限値
<code>DSMIF_SHORT_CNT_1</code> ※デフォルト値は"0x1FFE"	短絡検出条件の設定に使用する連続して入力される"1"の回数の上限値
<code>DSMIF_SHORT_CNT_0</code> ※デフォルト値は"0x1FFE"	短絡検出条件の設定に使用する連続して入力される"0"の回数の上限値

## 2.9 コードサイズ

本モジュールのコードサイズを下表に示します。

掲載した値は、「2.4 サポートされているツールチェーン」の C コンパイラでのコンパイルオプションがデフォルト時の参考値です。コンパイルオプションのデフォルトは最適化レベル : 2、最適化のタイプ : サイズ優先、データ・エンディアン : リトルエンディアンです。コードサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。

ROM および RAM のサイズ		
デバイス	領域	サイズ
RX72M	ROM	2040 バイト
	RAM	50 バイト

## 2.10 引数

API 関数の引数である定数、列挙体、共用体、構造体を示します。これらは API 関数のプロトタイプ宣言とともに r\_dsmif\_rx\_if.h で記載されています。

```

typedef struct
{
    uint32_t unit_no;                      /* DSMIF ユニット番号の指定(0/1) */
    uint32_t mode;                         /* DSMCLK の入出力方向の指定(マスタ/スレーブ) */
    uint32_t ckdiv;                        /* マスタ動作時のクロック分周比の指定 */
    uint32_t channel;                      /* 端子設定対象のチャネルを論理和で指定 */
    uint32_t dis_error;                    /* エラー検出を無効化する対象を論理和で指定 */
    void (*pcallback)(uint32_t);           /* コールバック関数ポインタ */
} st_dsmif_config_t;

/* DSMIF Unit No */
#define DSMIF_UNIT_DSMIF0                 (0U)   /* DSMIF ユニット 0 */
#define DSMIF_UNIT_DSMIF1                 (1U)   /* DSMIF ユニット 1 */

/* DSMIF Mode Setting */
#define DSMIF_MODE_SLAVE                  (0U)   /* スレーブモード */
#define DSMIF_MODE_MASTER                 (1U)   /* マスタモード */

/* DSMIF Channel&ErrorDisable No */
#define DSMIF_CH0                         (1U)   /* チャネル 0 */
#define DSMIF_CH1                         (2U)   /* チャネル 1 */
#define DSMIF_CH2                         (4U)   /* チャネル 2 */
#define DSMIF_SUM                          (8U)   /* 合計電流エラー検出 */

/* DSMIF Filter Setting Code ----- */
#define DSMIF_FILTER_SET_0                (0U)   /* フィルタ番号 0 */
#define DSMIF_FILTER_SET_1                (1U)   /* フィルタ番号 1 */
#define DSMIF_FILTER_SET_2                (2U)   /* フィルタ番号 2 */
#define DSMIF_FILTER_SET_3                (3U)   /* フィルタ番号 3 */
#define DSMIF_FILTER_SET_4                (4U)   /* フィルタ番号 4 */
#define DSMIF_FILTER_SET_5                (5U)   /* フィルタ番号 5 */
#define DSMIF_FILTER_SET_6                (6U)   /* フィルタ番号 6 */
#define DSMIF_FILTER_SET_7                (7U)   /* フィルタ番号 7 */
#define DSMIF_FILTER_SET_8                (8U)   /* フィルタ番号 8 */

```

#define DSMIF_FILTER_SET_9	(9U)	/* フィルタ番号 9 */
#define DSMIF_FILTER_SET_10	(10U)	/* フィルタ番号 10 */
#define DSMIF_FILTER_SET_11	(11U)	/* フィルタ番号 11 */
#define DSMIF_FILTER_SET_12	(12U)	/* フィルタ番号 12 */
#define DSMIF_FILTER_SET_13	(13U)	/* フィルタ番号 13 */
#define DSMIF_FILTER_SET_14	(14U)	/* フィルタ番号 14 */
 /* DSMIF Target Register to Read -----*/		
#define DSMIF_READ_DSMIF0_CDR0	(1U)	/* ユニット 0、CDR0 レジスタ */
#define DSMIF_READ_DSMIF0_CDR1	(2U)	/* ユニット 0、CDR1 レジスタ */
#define DSMIF_READ_DSMIF0_CDR2	(3U)	/* ユニット 0、CDR2 レジスタ */
#define DSMIF_READ_DSMIF1_CDR0	(4U)	/* ユニット 1、CDR0 レジスタ */
#define DSMIF_READ_DSMIF1_CDR1	(5U)	/* ユニット 1、CDR1 レジスタ */
#define DSMIF_READ_DSMIF1_CDR2	(6U)	/* ユニット 1、CDR2 レジスタ */
#define DSMIF_READ_DSMIF0_CCCDR0	(7U)	/* ユニット 0、CCCDR0 レジスタ */
#define DSMIF_READ_DSMIF0_CCCDR1	(8U)	/* ユニット 0、CCCDR1 レジスタ */
#define DSMIF_READ_DSMIF0_CCCDR2	(9U)	/* ユニット 0、CCCDR2 レジスタ */
#define DSMIF_READ_DSMIF1_CCCDR0	(10U)	/* ユニット 1、CCCDR0 レジスタ */
#define DSMIF_READ_DSMIF1_CCCDR1	(11U)	/* ユニット 1、CCCDR1 レジスタ */
#define DSMIF_READ_DSMIF1_CCCDR2	(12U)	/* ユニット 1、CCCDR2 レジスタ */
#define DSMIF_READ_DSMIF0_TCDDR0	(13U)	/* ユニット 0、TCDDR0 レジスタ */
#define DSMIF_READ_DSMIF0_TCDDR1	(14U)	/* ユニット 0、TCDDR1 レジスタ */
#define DSMIF_READ_DSMIF0_TCDDR2	(15U)	/* ユニット 0、TCDDR2 レジスタ */
#define DSMIF_READ_DSMIF1_TCDDR0	(16U)	/* ユニット 1、TCDDR0 レジスタ */
#define DSMIF_READ_DSMIF1_TCDDR1	(17U)	/* ユニット 1、TCDDR1 レジスタ */
#define DSMIF_READ_DSMIF1_TCDDR2	(18U)	/* ユニット 1、TCDDR2 レジスタ */
#define DSMIF_READ_DSMIF0_OCDR0	(19U)	/* ユニット 0、OCDR0 レジスタ */
#define DSMIF_READ_DSMIF0_OCDR1	(20U)	/* ユニット 0、OCDR1 レジスタ */
#define DSMIF_READ_DSMIF0_OCDR2	(21U)	/* ユニット 0、OCDR2 レジスタ */
#define DSMIF_READ_DSMIF1_OCDR0	(22U)	/* ユニット 1、OCDR0 レジスタ */
#define DSMIF_READ_DSMIF1_OCDR1	(23U)	/* ユニット 1、OCDR1 レジスタ */
#define DSMIF_READ_DSMIF1_OCDR2	(24U)	/* ユニット 1、OCDR2 レジスタ */

## 2.11 FIT モジュールの追加方法

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、Smart Configurator を使用した(1)、(3)の追加方法を推奨しています。ただし、Smart Configurator は、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(2)、(4)の方法を使用してください。

### (1) e<sup>2</sup> studio 上で Smart Configurator を使用して FIT モジュールを追加する場合

e<sup>2</sup> studio の Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e<sup>2</sup> studio スマート・コンフィグレータユーザガイド(R20AN0451)」を参照してください。

### (2) e<sup>2</sup> studio 上で FIT Configurator を使用して FIT モジュールを追加する場合

e<sup>2</sup> studio の FIT Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e<sup>2</sup> studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。

### (3) CS+上で Smart Configurator を使用して FIT モジュールを追加する場合

CS+上で、スタンダード版 Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e<sup>2</sup> studio スマート・コンフィグレータユーザガイド(R20AN0451)」を参照してください。

### (4) CS+上で FIT モジュールを追加する場合

CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」を参照してください。

### 3. API 関数

#### 3.1 R\_DSMIF\_Create

DSMIF の初期設定を行う関数です。

##### Format

```
void R_DSMIF_Create(st_dsmif_config_t* config);
```

##### Parameters

config

初期化設定構造体

構造体メンバの設定内容は、「2.10 引数」を参照してください。

##### Return Values

なし

##### Properties

r\_dsmif\_rx\_if.h にプロトタイプ宣言されています。

##### Description

DSMIF を使用するにあたって必要な下記の初期設定を行います。

- DSMIF のモジュールストップ状態の解除
- DSMIF レジスタの初期化
- エラー割り込みハンドラの登録
- ポートの設定

##### Example

```
st_dsmif_config_t dsmif_config;  
  
dsmif_config.unit_no = DSMIF_UNIT_DSMIF1;  
dsmif_config.mode    = DSMIF_MODE_SLAVE;  
dsmif_config.channel = DSMIF_CH0;  
dsmif_config.dis_error = (DSMIF_CH1| DSMIF_CH2 | DSMIF_SUM);  
dsmif_config.pcallback = &dsmif_error_callback;  
R_DSMIF_Create(&dsmif_config);
```

### 3.2 R\_DSMIF\_Start

DSMIF のフィルタを動作開始します。

#### Format

```
void R_DSMIF_Start(uint32_t unit_no);
```

#### Parameters

unit\_no

DSMIF ユニット番号(0~1)

#### Return Values

なし

#### Properties

r\_dsmif\_rx\_if.h にプロトタイプ宣言されています。

#### Description

DSMIF のフィルタ動作を開始（電流値モニタの開始）する際に必要な設定を行います。

- エラー検出条件の設定
- フィルタ動作を開始
- フィルタ出力安定時間のウェイト
- エラー検出状態のクリア
- エラー割り込みの許可

#### Example

```
/* Start DSMIF filtering */
R_DSMIF_Start(DSMIF_UNIT_DSMIF1);
```

---

### 3.3 R\_DSMIF\_Stop

---

DSMIF のフィルタを動作停止します。

#### Format

```
void R_DSMIF_Stop(uint32_t unit_no);
```

#### Parameters

unit\_no

DSMIF ユニット番号(0~1)

#### Return Values

なし

#### Properties

r\_dsmif\_rx\_if.h にプロトタイプ宣言されています。

#### Description

DSMIF のフィルタ動作を停止（電流値モニタの停止）する際に必要な設定を行います。

- フィルタ動作を停止
- エラー割り込みの禁止

#### Example

```
/* Stop DSMIF filtering */  
R_DSMIF_Stop(DSMIF_UNIT_DSMIF1);
```

### 3.4 R\_DSMIF\_SetFilter

DSMIF のフィルタ設定を変更する際に使用する関数です。

#### Format

```
void R_DSMIF_SetFilter(uint32_t unit_no, uint8_t filter_setting);
```

#### Parameters

unit\_no

DSMIF ユニット番号(0~1)

filter\_setting

フィルタ番号(0~14)

フィルタ番号により設定される内容については「表 3.1 フィルタ番号による設定内容」を参照してください。

#### Return Values

なし

#### Properties

r\_dsmif\_rx\_if.h にプロトタイプ宣言されています。

#### Description

DSMIF のフィルタ設定の変更を行います。

指定したフィルタ番号によって以下の項目を変更します。

- SINC フィルタ次数
- デシメーション比
- 分解能

#### Example

```
R_DSMIF_SetFilter(DSMIF_UNIT_DSMIF1, DSMIF_FILTER_SET_14);
```

表 3.1 フィルタ番号による設定内容

フィルタ番号	SINC フィルタ次数	デシメーション比	分解能
0	3 (sinc3)	8	9 ビット
1		16	12 ビット
2		32	15 ビット
3		64	16 ビット
4		128	16 ビット
5		256	16 ビット
6	2 (sinc2)	16	8 ビット
7		32	10 ビット
8		64	12 ビット
9		128	14 ビット
10		256	16 ビット
11	1 (sinc1)	32	5 ビット
12		64	6 ビット
13		128	7 ビット
14		256	8 ビット

### 3.5 R\_DSMIF\_ReadData

DSMIF のフィルタリング動作により変換されたデータを読み出します。

#### Format

```
uint16_t R_DSMIF_ReadData(uint8_t data_reg);
```

#### Parameters

data\_reg

リード対象のレジスタ名

レジスタ名として使用する定数については、「2.10 引数」を参照してください。

#### Return Values

データレジスタ値

#### Properties

r\_dsmif\_rx\_if.h にプロトタイプ宣言されています。

#### Description

DSMIF のフィルタリング動作により変換された 16 ビット・デジタルデータをレジスタより読み出します。

引数に指定されたレジスタに格納されている電流値を読み出します。

#### Example

```
/* Read DSMIF Unit1 CDR0 register */  
uint32_t read_data;  
read_data = R_DSMIF_ReadData(DSMIF_READ_DSMIF1_CDR0);
```

---

### 3.6 R\_DSMIF\_GetErrorStatus

---

DSMIF のエラーステータスレジスタを読み出します。

#### Format

```
uint32_t R_DSMIF_GetErrorStatus(uint32_t unit_no);
```

#### Parameters

unit\_no

DSMIF ユニット番号(0~1)

#### Return Values

DSMIF ステータスレジスタ(DSSR)の値

#### Properties

r\_dsmif\_rx\_if.h にプロトタイプ宣言されています。

#### Description

エラー検出状態を格納した DSMIF ステータスレジスタ(DSSR)の値を読み出します

#### Example

```
/* Read DSMIF Unit1 status register */  
  
uint32_t err_sts;  
err_sts = R_DSMIF_GetErrorStatus(DSMIF_UNIT_DSMIF1);
```

#### 4. 端子設定

DSMIFI FIT モジュールを使用するためには、マルチファンクションピンコントローラ (MPC) で周辺機能の入出力信号を端子に割り付ける（以下、端子設定と称す）必要があります。端子設定は、R\_DSMIF\_Open 関数を呼び出した後、R\_DSMIF\_Start 関数を呼び出す前に行ってください。

e<sup>2</sup> studio の場合は「FIT Configurator」または「Smart Configurator」の端子設定機能を使用することができます。FIT Configurator、Smart Configurator の端子設定機能を使用すると、端子設定画面で選択したオプションに応じて、ソースファイルが出力されます。そのソースファイルで定義された関数を呼び出すことにより端子を設定できます。詳細は表 4.1 を参照してください。

表 4.1 FIT Configurator が出力する関数一覧

使用マイコン	出力される関数名	備考
RX72M	R_DSMIF_PinSet_DSMIFx	x : ユニット番号

## 5. デモプロジェクト

デモプロジェクトには、FIT モジュールとそのモジュールが依存するモジュール（例：r\_bsp）を使用する main() 関数が含まれます。本 FIT モジュールには以下のデモプロジェクトが含まれます。

### 5.1 dsmif\_demo\_rskrx72m

dsmif\_demo\_rskrx72m プログラムは、Renesas Starter Kit+ for RX72M ボード（以下 RSKRX72M ボード）に接続した Photo coupler (SDIP8) Evaluation Sub Board（以下 DSMIF サブボード）から受信した電流測定値を、e<sup>2</sup> studio の仮想コンソールに出力します。

デモプログラムのセットアップ方法と動作手順について、以下に説明します。

#### 5.1.1 ボードの設定と接続

ボードの詳細情報に関しては、「Renesas Starter Kit+ for RX72M ユーザーズマニュアル」（以下 RSKRX72M ボードマニュアル）をご参照ください。

##### 5.1.1.1 RSKRX72M ボードの設定

本アプリケーションノートのデモプログラムを動作させるためには、RSKRX72M ボードの DSMDAT3 を有効にします。具体的には R345 チップ抵抗取り外し、R344 に実装します。

詳細は RSKRX72M ボードマニュアルの「6.9 DSMIF 設定」を参照してください。

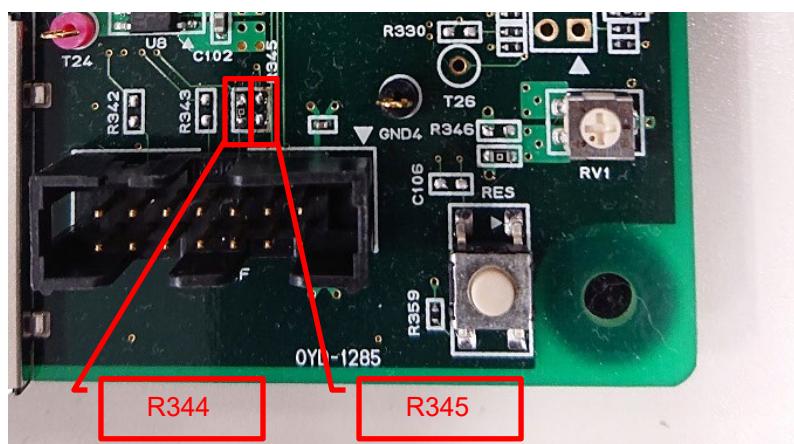


図 5.1 DSMDAT3 抵抗設定

### 5.1.1.2 RSKRX72M ボードと DSMIF サブボードの接続

RSKRX72M ボードの J9 コネクタに DSMIF サブボードを接続します。

図 5.2 の向きで接続してください。

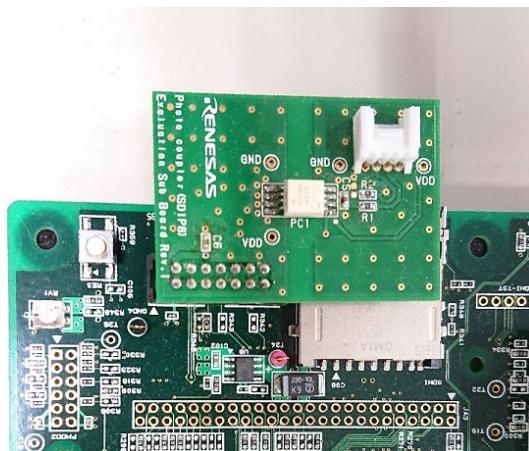


図 5.2 RSKRX72M ボードと DSMIF サブボードの接続

### 5.1.1.3 DSMIF サブボードへの電源接続

DSMIF サブボードの VDD, GND 端子に DC5V を供給してください。

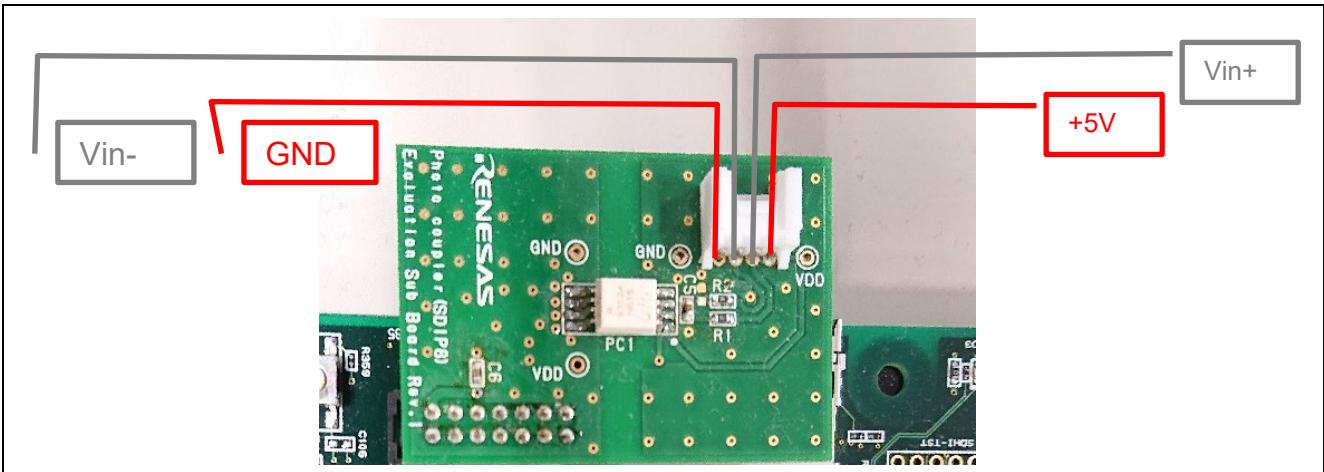


図 5.3 DSMIF サブボードへの電源接続

#### 5.1.1.4 DSMIF サブボードへの測定信号接続

DSMIF サブボードの Vin+, Vin-端子に測定対象のアナログ電圧を供給してください。

入力値は、DSMIF サブボードに搭載された PS9352A の Vin+, Vin-定格内にしてください。

入力電圧の推奨値は±200mV 以内です。

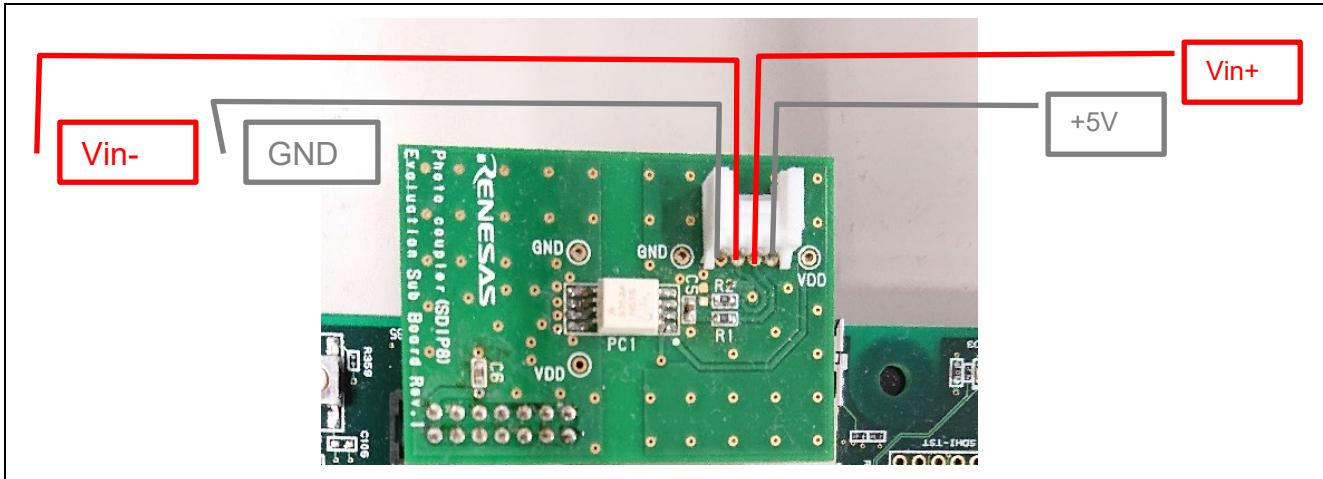


図 5.4 DSMIF サブボードへの測定信号接続

### 5.1.2 デモプロジェクト動作手順

本デモプロジェクトの動作手順を以下に示します。

#### 5.1.2.1 デモプロジェクトのメニュー

デモプロジェクトを実行し通信可能状態になると、デモプロジェクトのメニューが仮想コンソールに表示されます。

##### (1) トップメニュー

プログラム開始直後は、下記のトップメニューが表示されます。

```
Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [High]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

図 5.5 トップメニュー

ここでは、下記の動作を選択することができます。

- 1を選択した場合：電流値計測開始へ
- 2を選択した場合：スキャン頻度設定メニューへ
- 3を選択した場合：フィルタ設定メニューへ
- 8を選択した場合：ログ出力へ
- 9を選択した場合：デモプロジェクト終了へ

## (2) スキャン頻度設定メニュー

トップメニューにて[2]を選択すると、スキャン頻度の設定変更ができます。

```
>2[Enter]
```

Input the number to select the measurement mode.

[1] High Frequency

[2] Low Frequency

```
>2[Enter]
```

図 5.6 スキャン頻度設定メニュー

設定したいスキャン頻度の番号（1~2）を選択します。

スキャン頻度を変更せずにメニューを抜ける場合は、[Enter]のみを入力してください。

ここでは2番の「低頻度」を選択しています。

スキャン頻度を設定すると、設定変更した旨のメッセージが表示され、メニュー画面へ戻ります。

[2]の項目末尾の現在の設定値[ ]が変更されていることが確認できます。

```
> 2
```

Set to Low frequency mode.

Delta Sigma Interface test program

[1] Start Measurement

[2] Measurement Frequency [Low]

[3] Filter Settings [0]

[8] Log Output

[9] Exit

```
>
```

図 5.7 スキャン頻度設定後のトップメニューの表示更新

## (3) フィルタ設定メニュー

メニューにて[3]を選択すると、フィルタ設定の変更ができます。

```
>3[Enter]

Input the number to set the filter (0-14)

  SINC filter  Decimation  Data
-----
[0]  3          1/8        b8-b0, 0000000
[1]  3          1/16       b11-b0, 0000
[2]  3          1/32       b14-b0, 0
[3]  3          1/64       b17-b2
[4]  3          1/128      b20-b5
[5]  3          1/256      b23-b8
[6]  2          1/16       b7-b0, 00000000
[7]  2          1/32       b9-b0, 000000
[8]  2          1/64       b11-b0, 0000
[9]  2          1/128      b13-b0, 00
[10] 2          1/256      b15-b0
[11] 1          1/32       b4-b0, 000000000000
[12] 1          1/64       b5-b0, 000000000000
[13] 1          1/128      b6-b0, 0000000000
[14] 1          1/256      b7-b0, 00000000

>5
```

図 5.8 フィルタ設定メニュー

設定したいフィルタ設定の組み合わせを 0~14 の番号にて選択します。フィルタ設定の組み合わせの詳細は、表 3.1 「フィルタ番号による設定内容」を参照してください。

フィルタ設定を変更しないでメニューを抜ける場合は、[Enter]のみ入力してください。

ここでは 5 番のフィルタ設定（フィルタ 3 段、デシメーションクロック 1/256、最終データ出力 b23-b8）を選択しています。

フィルタを選択すると、設定変更した旨のメッセージが表示され、メニュー画面へ戻ります。

[3]の項目末尾の現在の設定値[ ]が変更されていることが確認できます。

```
> 5
```

Set to 5.

Delta Sigma Interface test program

[1] Start Measurement

[2] Measurement Frequency [Low]

[3] Filter Settings [5]

[8] Log Output

[9] Exit

```
>
```

図 5.9 フィルタ設定後のトップメニューの表示更新

#### (4) 電流値計測開始

トップメニューにて[1]を選択すると、現在設定されているスキャン頻度・フィルタ設定にて電流値のモニタが開始されます。

Delta Sigma Interface test program

[1] Start Measurement

[2] Measurement Frequency [High]

[3] Filter Settings [0]

[8] Log Output

[9] Exit

```
>1[Enter]
```

```
<start>
```

33792

図 5.10 電流値計測開始後の表示

## (a) 高頻度スキャン

高頻度スキャンモードでは、1秒間に 1000 回の電流値スキャンを行い、スキャン終了後にまとめて全ログを表示します。

1万回分のログを表示し終わると、メニュー画面へ戻ります。

```
<start>
33792
33280
.
.
32128
<end>

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [High]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

図 5.11 高頻度スキャンの結果表示

## (b) 低頻度スキャン

低頻度スキャンモードでは、1秒ごとに電流値のスキャンを行い、ログを逐一表示します。

スキャンを停止させる場合は、CPU ボードの SW1 を押してください。ボタン割り込みによりスキャンは停止し、メニュー画面へ戻ります。

```
<start>
34048
32512
33408
34304
33792
[SW1 押下]
<end>

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [Low]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

図 5.12 低頻度スキャンの結果表示

## (c) エラー検出時のスキャン停止

電流値計測動作中にエラー（過電流異常検出エラー/短絡異常検出エラー/合計電流エラー）が発生した場合、電流値計測動作は中断され、エラー内容を表示した後メニュー画面へ戻ります。

```
<start>
58496
63232
59392
62336
OVER CURRENT ERROR
<end>

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [High]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

図 5.13 エラー発生時のスキャン停止

## (5) ログ出力

トップメニューにて[8]を選択すると、直前のスキャン結果を出力します。

ログは 10 データずつ改行して表示されます。

```
> 8[Enter]

Log Output
-----
[1] 35686 35686 32791 32779 32779 32752 32752 32661 32721 32721
[11] 32727 32727 32738 32733 32733 32736 32736 32729 32731 32731
[21] 32726 32726 32716 32723

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [Low]
[3] Filter Settings [5]

[8] Log Output
[9] Exit

>
```

図 5.14 ログ出力

## (6) デモプロジェクト終了

トップメニューにて[9]を選択すると、デモプロジェクトの動作が停止し、メニューの表示が止まります。

## 6. 付録

### 6.1 動作確認環境

このセクションでは、DSMIF モジュールの動作確認用の環境について説明します。

表 6.1 動作確認環境(Rev1.02)

項目	内容
統合開発環境 (IDE)	ルネサスエレクトロニクス製 e2 studio 2025-01 IAR Embedded Workbench for Renesas RX 5.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V3.07.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99 GCC for Renesas RX 8.3.0.202411 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 IAR C/C++ Compiler for Renesas RX version 5.10.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン／リトルエンディアン
モジュールのリビジョン	Rev.1.02
使用ボード	-

表 6.2 動作確認環境(Rev1.01)

項目	内容
統合開発環境 (IDE)	ルネサスエレクトロニクス製 e2 studio V.7.5.0 IAR Embedded Workbench for Renesas RX 4.12.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V3.01.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99 GCC for Renesas RX 4.8.4.201803 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 IAR C/C++ Compiler for Renesas RX version 4.12.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン／リトルエンディアン
モジュールのリビジョン	Rev.1.01
使用ボード	Renesas Starter Kit+ for RX72M (型名：RTK5572MNDCxxxxBJ) + Photo coupler (SDIP8) Evaluation Sub Board

表 6.3 動作確認環境(Rev1.00)

項目	内容
統合開発環境 (IDE)	ルネサスエレクトロニクス製 e2 studio V.7.5.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V3.01.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 4.8.4.201803 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
エンディアン	ビッグエンディアン／リトルエンディアン
モジュールのリビジョン	Rev.1.00
使用ボード	Renesas Starter Kit+ for RX72M (型名 : RTK5572MNDCxxxxBJ) + Photo coupler (SDIP8) Evaluation Sub Board

## 7. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX72M グループ ユーザーズマニュアル ハードウェア編（ドキュメント No. R01UH0804）

PS9352AL2 光結合型デルタシグマ変調器 データシート（ドキュメント No. R08DS0129）

（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

RX ファミリ C/C++コンパイラ、アセンブラー、最適化リンクエディタコンパイラパッケージ

（R20UT0570）

（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.02.2019	—	初版発行
1.01	Sep.17.2019	—	以下のコンパイラをサポート。 - IAR C/C++ Compiler for Renesas RX
1.02	Mar.20.2025	28	表 6.1 動作確認環境(Rev1.02) を追加
		プログラム	プログラムの免責事項を変更

## 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誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。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、何らの保証を行ふものではなく、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いたします。
  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 (豊洲フォレシア)

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

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

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。