RX ファミリ
DAC モジュール
Firmware Integration Technology

R01AN1818JJ0300
Rev. 3.00
2017.02.28

要旨

本ドキュメントは、Firmware Integration Technology (FIT)を使用した D/A コンバータ（DAC）モジュールについて説明します。本モジュールは、RX111、RX113、RX130、RX210、RX230、RX231、RX23T、RX24T、RX24U、RX63N、RX631、RX64M、RX651、RX65N、RX71M グループの D/A コンバータ周辺機能をサポートします。チャネルは独立して動作が可能です。本モジュールの API は、8/10/12 ビットコンバータで共通でご使用いただけます。

以降、本モジュールを DAC FIT モジュールと称します。

対象デバイス

本モジュールは以下のデバイスで使用できます。

- RX111 グループ
- RX113 グループ
- RX130 グループ
- RX210 グループ
- RX230 グループ、RX231 グループ
- RX23T グループ
- RX24T グループ
- RX24U グループ
- RX631 グループ、RX63N グループ
- RX64M グループ
- RX651 グループ、RX65N グループ
- RX71M 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。

関連ドキュメント

- Firmware Integration Technology ユーザーズマニュアル (R01AN1833)
- ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)
- e^2 studio に組み込む方法 Firmware Integration Technology (R01AN1723)
- CS+に組み込む方法 Firmware Integration Technology (R01AN1826)

目次

1. 概要

DAC FIT モジュールは、RX111、RX113、RX130、RX210、RX230、RX231、RX23T、RX24T、RX24U、RX63N、RX631、RX64M、RX651、RX65N、RX71M グループの D/A コンバータ周辺機能をサポートします。D/A コンバータ周辺機能の詳細は、ご使用の MCU のユーザーズマニュアル: ハードウェアの「D/A コンバータ（DA）」の章をご覧ください。

以降、D/A コンバータ周辺機能を D/A コンバータと称します。

アナログに変換するデータは左揃え、または右揃えで配置でき、チャネルは個別に出力できます。各 MCU でサポートしているハードウェア機能にも対応しています。以下に機能の例を示します。

- 基準電圧の選択
- A/D コンバータ周辺機能との同期変換
- 出力禁止の場合は、変換を無効にする
- 負荷が大きい場合は、内蔵アンプを有効にする

2. API 情報

本アプリケーションノートのサンプルコードは、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

本モジュールを使用するには、ご使用の MCU が以下の機能をサポートしていることが要求されます。

- D/A コンバータ（DA、DAa、R12DA または R12DAA）

2.2 ハードウェアリソースの要求

ここでは、本モジュールが要求するハードウェアの周辺機能について説明します。特に記載がない場合、ここで説明するリソースは本モジュールが使用できるように、ユーザは使用しないでください。

2.2.1 DA、DAa、R12DA、R12DAA

本モジュールでは、DA、DAa、R12DA、R12DAA のすべての機能を使用できます。

2.2.2 GPIO

本モジュールでは、各チャネルに対応するポート端子を使用できます。これらの端子は汎用入出力端子としては使用できません。

2.3 ソフトウェアの要求

本モジュールは以下のソフトウェアに依存します。

- ルネサスボードサポートパッケージ（r_bsp）

2.4 制限事項

ソフトウェアに関する制限事項はありません。

2.5 対応ツールチェーン

本モジュールは下記ツールチェーンで動作確認を行っています。

- Renesas RX Toolchain v.2.02.00 (RX111、RX113、RX210、RX231、RX631、RX63N、RX64M、RX71M)
- Renesas RX Toolchain v.2.03.00 (RX130、RX23T、RX230、RX24T)
- Renesas RX Toolchain v.2.05.00 (RX24U、RX651、RX65N)
- Renesas RX Toolchain v.2.06.00 (RX24U)

2.6 ヘッダファイル

すべての API 呼び出しと対応するインタフェース定義は、r_dac_rx_if.h ファイルに記述されています。r_dac_rx_config.h ファイルで、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を選択あるいは定義できます。

上記 2 ファイルは、ユーザアプリケーションによってインクルードする必要があります。

2.7 整数型

コードをわかりやすく、また移植が容易に行えるように、本プロジェクトでは ANSI C99 (Exact width integer types (固定幅の整数型)) 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

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_dac_rx_config.h ファイルに含まれます。下表に設定の概要を示します。

コンフィギュレーションオプション（r_dac_rx_config.h）

定義	説明
#define DAC_CFG_PARAM_CHECKING_ENABLE ※デフォルト値は“1”	• 1：ビルド時にパラメータチェック処理をコードに含めます。 • 0：ビルド時にパラメータチェック処理をコードから省略します。 • BSP_CFG_PARAM_CHECKING_ENABLE（デフォルト）：システムのデフォルト設定を使用します。 注: ビルド時にパラメータチェックのコードを省略することで、コードサイズを小さくすることができます。

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。RX100 シリーズ、RX200 シリーズ、RX600 シリーズから代表して1 デバイスずつ掲載しています。
ROM（コードおよび定数）と RAM（グローバルデータ）のサイズは、ビルド時の「2.8 コンパイル時の設定」のコンフィギュレーションオプションによって決まります。掲載した値は、「2.5 サポートされているツールチェーン」の C コンパイラで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、RAM およびスタックのコードサイズ

デバイス	分類	使用メモリ		備考
		パラメータチェック処理あり	パラメータチェック処理なし	
RX130	ROM	319 バイト	285 バイト	
	RAM	0 バイト	0 バイト	
	最大使用スタックサイズ	20 バイト		R_DAC_Open 関数使用時
RX231	ROM	338 バイト	289 バイト	
	RAM	0 バイト	0 バイト	
	最大使用スタックサイズ	20 バイト		R_DAC_Open 関数使用時
RX65N	ROM	403 バイト	359 バイト	
	RAM	0 バイト	0 バイト	
	最大使用スタックサイズ	20 バイト		R_DAC_Open 関数使用時

2.10 API データ構造体

本モジュールの API 関数で使用される構造体は、“r_dac_rx_if.h”に記載されます。詳細はセクション 3 をご覧ください。

2.11 FIT モジュールの追加方法

本モジュールは、e² studio で、使用するプロジェクトごとに追加する必要があります。

プロジェクトへの追加方法は、FIT プラグインを使用する方法と、手動で追加する方法があります。

FIT プラグインを使用すると、簡単にプロジェクトに FIT モジュールを追加でき、またインクルードファイルパスも自動的に更新できます。このため、プロジェクトへ FIT モジュールを追加する際は、FIT プラグインの使用を推奨します。

FIT プラグインを使用して FIT モジュールを追加する方法は、アプリケーションノート「e² studio に組み込む方法(R01AN1723)」の「3. FIT プラグインを使用して FIT モジュールをプロジェクトに追加する方法」を参照してください。

FIT プラグインを使用せず手動で FIT モジュールを追加する方法は、「4. 手作業で FIT モジュールをプロジェクトに追加する方法」を参照してください。

FIT モジュールを使用する場合、ボードサポートパッケージ FIT モジュール(BSP モジュール)もプロジェクトに追加する必要があります。BSP モジュールの追加方法は、アプリケーションノート「ボードサポートパッケージモジュール(R01AN1685)」を参照してください。

3. API 関数

3.1 概要

本モジュールには以下の関数が含まれます。

関数	説明
R_DAC_Open()	D/A コンバータを起動し、関連するレジスタを初期化して、MCU ごとで選択可能なオプションを設定します。
R_DAC_Close()	D/A コンバータを停止します。
R_DAC_Write()	変換動作のために、チャネルに対応するレジスタにデータを書き込みます。
R_DAC_Control()	チャネルの出力を有効または無効します。 内蔵アンプを有効または無効にします（RX64M、RX71M）。
R_DAC_GetVersion()	本モジュールのバージョン番号を返します。

3.2 戻り値

以下に本モジュールの API 関数で使用する戻り値（エラーコード）を示します。戻り値の列挙型は、API 関数の宣言と共に r_dac_rx_if.h に記載されています。

```
/* DAC で使用される API エラーコード */
typedef enum e_dac_err
{
     DAC_SUCCESS=0,
     DAC_ERR_BAD_CHAN,          // 存在しないチャネル番号
     DAC_ERR_INVALID_CMD,       // 無効な動作コマンド
     DAC_ERR_INVALID_ARG,       // パラメータに対して無効な引数です。
     DAC_ERR_NULL_PTR,          // null ptr を受信; 要求された引数がありません。
     DAC_ERR_LOCK_FAILED,       // D/A コンバータのロックに失敗しました
                                // (モジュールは既に起動しています)。
     DAC_ERR_UNLOCK_FAILED      // D/A コンバータのロック解除に失敗しました。
     DAC_ERR_ADC_NOT_POWERED,// A/D コンバータが起動していないので、同期変換できません。
     DAC_ERR_ADC_CONVERTING     // A/D コンバータが変換動作中なので、同期変換できません。
} dac_err_t;
```

3.3 R_DAC_Open()

D/A コンバータを起動し、関連するレジスタを初期化して、MCU ごとで選択可能なオプションを設定します。

Format

```
dac_err_t R_DAC_Open(dac_cfg_t * p_cfg);
```

Parameters

p_cfg

設定構造体へのポインタ

“p_cfg”で使用される構造体サンプル：

```
typedef struct st_dac_cfg
{
    bool        fmt_flush_right;              // 全 MCU
    bool        sync_with_adc;                // X113/RX130/RX230/RX231/
                                              // RX24U/RX63N/RX631/RX64M/
                                              // RX65N/RX651/RX71M
    uint8_t     sync_unit;              // 0 or 1; RX64M/RX65N/RX651/RX71M
    bool        ch_conv_off_when_output_off;  // RX210/RX63N/RX631/RX64M/
                                              // RX65N/RX651/RX71M
    dac_refv_t  ref_voltage;                  // RX113/RX230/RX231
} dac_cfg_t;

typedef enum e_dac_refv                       // D/A コンバータ基準電圧
{
    DAC_REFV_AVVC0_AVSS0 = 1,
    DAC_REFV_INTERNAL_AVSS0 = 3,
    DAC_REFV_VREFH_VREFL = 6
} dac_refv_t;
```

Return Values

```
DAC_SUCCESS             /* 成功; D/A コンバータが初期化されました。*/
DAC_ERR_NULL_PTR        /* “p_cfg”ポインタが NULL です。*/
DAC_ERR_LOCK_FAILED     /* DAC モジュールのロックに失敗; DAC は既に動作中です。*/
DAC_ERR_INVALID_ARG     /* 同期するユニットの番号が無効です。*/
DAC_ERR_ADC_NOT_POWERED /* A/D コンバータが起動していないので、同期変換できません。*/
DAC_ERR_ADC_CONVERTING  /* A/D コンバータが変換動作中なので、同期変換できません。*/
```

Properties

ファイル r_dac_rx_if.h にプロトタイプ宣言されています。

Description

本関数は、D/A コンバータを起動し、関連するレジスタを初期化して、MCU ごとで選択可能なオプションを設定します。

Reentrant

不可

Example

```
dac_err_t err;
dac_cfg_t config;

/* RX63N DAC を初期化する */
config.fmt_flush_right = true;
config.sync_with_adc = false;
config.ch_conv_off_when_output_off = true;
err = R_DAC_Open(&config);
```

Special Notes:

データはアプリケーションによって、左揃え、または右揃えで配置できます。“fmt_flush_right”パラメータで、DAC にデータの配置方法を示します。

“DAC_ERR_ADC_CONVERTING”エラーを回避するためには、A/D コンバータが起動してスキャンを開始する前に、D/A コンバータを起動します。

D/A コンバータの I/O 端子は、本関数を呼び出す前に設定しておく必要があります。

以下に初期化の例を示します。

```
R_BSP_RegisterProtectDisable(BSP_REG_PROTECT_MPC); // ロック解除

#ifdef BSP_MCU_RX113
 /*
 * RX113 グループユーザーズマニュアル:ハードウェアの注記に従って設定。
 * 表 19.1「 マルチプル端子の割り当て端子一覧 (10/10)」 下の注 1 :
 *「この端子機能を使用する場合は、該当端子の設定を汎用入力にしてください
 *（PORT.PDR.Bm ビットお* よび PORT.PMR.Bm ビットを“0”にする）。」
 */
 PORTJ.PDR.BIT.B0 = 0;
 PORTJ.PMR.BIT.B0 = 0;
 PORTJ.PDR.BIT.B2 = 0;
 PORTJ.PMR.BIT.B2 = 0;

 /* PJ0 と PJ2 を D/A コンバータのアナログ出力端子として設定 */
 MPC.PJ0PFS.BIT.ASEL = 1;
 MPC.PJ2PFS.BIT.ASEL = 1;
 /*
 * D/A コンバータの基準電圧に VREFH/VREFL を使用する場合は、以下の 2 行のコメントを
 * 解除します。
 */
 //MPC.P41PFS.BIT.ASEL = 1; // P41 を VREFH アナログ端子として設定
 //MPC.P42PFS.BIT.ASEL = 1; // P42 を VREFL アナログ端子として設定

 #else /* RX111, RX210, RX63N */

 /* アナログ出力用の I/O ポート端子を汎用入力端子として設定
 PORT0.PDR.BIT.B3 = 0;
 PORT0.PMR.BIT.B3 = 0;
 PORT0.PDR.BIT.B5 = 0;
 PORT0.PMR.BIT.B5 = 0;
```

```
/* P03 と P05 の端子機能を D/A コンバータのアナログ出力端子として設定*/
MPC.P03PFS.BIT.ASEL = 1;
MPC.P05PFS.BIT.ASEL = 1;
#endif
R_BSP_RegisterProtectEnable(BSP_REG_PROTECT_MPC); // ロック
```

アンプを使用する時の注意事項

アンプを使用するときは“ch_conv_off_when_output_off”を“true”に設定してください。

A/D コンバータ使用時の注意事項

D/A A/D 同期変換有効 (sync_with_adc = true) に設定していた場合、A/D コンバータ（注 1）をモジュールストップ状態にするのであれば、その前に R_DAC_Close 関数を実行してください。

注1. RX64M/RX651/RX65N/RX71M の場合はユニット 1、RX24U の場合はユニット 2 が対象になります。

3.4 R_DAC_Close()

この関数は D/A コンバータを停止します。

Format

```
dac_err_t R_DAC_Close(void);
```

Parameters

なし

Return Values

```
DAC_SUCCESS                 /* 成功; チャネルの出力を無効にしました。*/
DAC_ERR_UNLOCK_FAILED       /* D/A コンバータのロック解除に失敗しました。*/
```

Properties

ファイル r_dac_rx_if.h にプロトタイプ宣言されています。

Description

DAC の出力を禁止して、モジュールストップに移行します。

Reentrant

不可

Example

```
:
/* D/A コンバータを初期化 */
err = R_DAC_Open(&config);
:
:
/* D/A コンバータを終了 */
err = R_DAC_Close();
```

Special Notes:

D/A A/D 同期変換有効 (sync_with_adc = true) に設定していた場合、A/D コンバータ（注 1）をモジュールストップ状態にするのであれば、その前に R_DAC_Close 関数を実行してください。

注1. RX64M/RX651/RX65N/RX71M の場合はユニット 1、RX24U の場合はユニット 2 が対象になります。

3.5 R_DAC_Write()

この関数は、チャネルに対応するデータレジスタにデータを書き込みます。

Format

```
dac_err_t R_DAC_Write(uint8_t const chan, uint16_t data);
```

Parameters

chan

書き込みするチャネル

data

書き込むデータ

Return Values

```
DAC_SUCCESS          /* 成功; チャネルに対応するレジスタにデータが書き込まれました。*/
DAC_ERR_BAD_CHAN     /* 存在しないチャネル番号 */
```

Properties

ファイル r_dac_rx_if.h にプロトタイプ宣言されています。

Description

変換に際して、チャネルに対応するレジスタにデータを書き込みます。ご使用の MCU によって、データ長は 8/10/12 ビットのいずれかになります。Write()関数にてデータを格納する場合は、データは選択された右詰または左詰のフォーマットで格納されなければなりません。

Reentrant

この関数は異なるチャネルに対して再入可能（リエントラント）です。

Example

```
dac_err_t     err;
uint16_t      g_short;
       :
       :
/* チャネル 1 に 0V へ変換するためのデータを書き込む。 */
g_short = 0x0000;
err = R_DAC_Write(DAC_CH1, g_short);
```

Special Notes:

なし

3.6 R_DAC_Control()

この関数はチャネルを有効または無効にします。

Format

```
dac_err_t R_DAC_Control(uint8_t const chan, dac_cmd_t const cmd);
```

Parameters

chan

使用するチャネル

cmd

実行するコマンド（以下の列挙型参照）

以下に“cmd”の値を示します。

```
typedef enum e_dac_cmd
{
     DAC_CMD_OUTPUT_ON,        // チャネルのアナログ出力が有効にされました。
     DAC_CMD_OUTPUT_OFF,       // チャネルのアナログ出力が無効にされました。

     DAC_CMD_AMP_ON,           // RX64M、RX71M: アンプ、ゲイン 1。ユーザーズマニュアル:
     DAC_CMD_AMP_OFF,          // ハードウェアで「電気的特性」の章をご覧ください。
     DAC_CMD_END_ENUM
} dac_cmd_t;
```

Return Values

```
DAC_SUCCESS                /* 成功; チャネルを無効にしました。*/
DAC_ERR_BAD_CHAN           /* 存在しないチャネル番号 */
DAC_ERR_INVALID_CMD        /* 無効なコマンド */
```

Properties

ファイル r_dac_rx_if.h にプロトタイプ宣言されています。

Description

OUTPUT コマンドを使って、Write()関数にてデータレジスタに書き込まれた変換データ値を出力します。AMP コマンドを使って、アンプを有効または無効にします。アンプを有効にした後は、出力を許可しておかなければなりません。

Reentrant

この関数は異なるチャネルに対して再入可能（リエントラント）です。

Example

```
dac_cfg_t config;
dac_err_t err;

/* RX64M, RX71M D/A コンバータを初期化する */
config.fmt_flush_right = true;
config.sync_with_adc = true;
config.sync_unit = 1;
config.ch_conv_off_when_output_off = true;


err = R_DAC_Open(&config);

/* チャネル 0 に 0V に変換するためのデータを書き込む */
err = R_DAC_Write(DAC_CH0, 0x0);

/* 大き目の負荷に対応 */
err = R_DAC_Control(DAC_CH0, DAC_CMD_AMP_ON);

/* 変換データの出力 */
err = R_DAC_Control(DAC_CH0, DAC_CMD_OUTPUT_ON);
/* 3us 以上の wait を入れてください  */
/* チャネル 0 に 3.3V に変換するためのデータを書き込む */
err = R_DAC_Write(DAC_CH0, 0x0FFF);
```

Special Notes:

・R_DAC_Write(DAC_CHx, 0x0)を実施して出力アンプを作動させること。

・出力アンプ使用時（コマンド DAC_CMD_AMP_ON 実行時）はオープン関数にて
“ch_conv_off_when_output_off”を“true”に設定してください。
設定は無効となります。

・アンプを使うときは、下記の手順で行ってください。

1. R_DAC_Control()関数で、DAC_CMD_AMP_ON コマンド実行
2. R_DAC_Control()関数で、DAC_CMD_OUTPUT_ON コマンドを実行
3. 3us 以上の時間待つ
4. R_DAC_Write 関数にて D/A 出力値を書き込む

DAC_CMD_OUTPUT_ON、および DAC_CMD_OUTPUT_OFF コマンドは、D/A A/D 同期変換を有効（R_DAC_Open 関数の p_cfg.sync_with_adc に true を設定して実行）にしている場合、同期する A/D コンバータ（注 1）が停止している状態で実行してください

注1. RX64M/RX651/RX65N/RX71M の場合、ユニット 1 を、RX24U の場合はユニット 2 を停止してください。その他の MCU グループではユニットが 1 つのため、指定はありません。

3.7 R_DAC_GetVersion()

この関数は実行時に本モジュールのバージョンを返します。

Format

```
uint32_t R_DAC_GetVersion(void)
```

Parameters

なし

Return Values

本モジュールのバージョン

Properties

ファイル r_dac_rx_if.h にプロトタイプ宣言されています。

Description

この関数は本モジュールのバージョンを返します。バージョン番号は符号化され、最上位の 2 バイトがメジャーバージョン番号を、最下位の 2 バイトがマイナーバージョン番号を示しています。

Reentrant

この関数は再入可能（リエントラント）です。

Example

```
uint32_t version;
:
version = R_DAC_GetVersion();
```

Special Notes:

本関数は#pragma ディレクティブを使ったインライン関数となります。

4. デモプロジェクト

デモプロジェクトはスタンドアロンプログラムです。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。デモプロジェクトの標準命名規則は “<module>_demo_<board>” で、<module>部には周辺機能の頭字語（例 : s12ad、cmt、sci）が、<board>部には標準的な RSK（例 : rskrx113）が入ります。例えば、RSKRX113 対応の s12ad FIT モジュールのデモプロジェクトの場合、デモプロジェクト名は “s12ad_demo_rskrx113” となります。同様に ZIP ファイルをエクスポートする場合の命名規則も “<module>_demo_<board>.zip” となります。デモプロジェクト “s12ad_demo_rskrx113” をインポート／エクスポートする場合、“s12ad_demo_rskrx113.zip” となります。

4.1 dac_demo_rskrx113

dac_demo_rskrx113 は、RSKRX113、RX113 D/A コンバータ（R12DAA）対応の DAC FIT モジュール（r_dac_rx）のシンプルなデモプロジェクトです。デモでは r_dac_rx API を使用して、DAC の起動、設定、書き込みを行います。LOW、MEDIUM、または HIGH レベルの値が DAC チャネル 1 に書き込まれると、連続的なループに 1 秒間遷移します。LOW レベルの値が書き込まれると LED0（緑）が点灯し、MEDIUM レベルの値が書き込まれると LED1（オレンジ）が、HIGH レベルの値が書き込まれると LED2（赤）が点灯します。

RSKRX113 ボード使用時の DAC 出力チャネル信号のアクセスおよび設定、また基準電圧に関する詳細については、以下の「DAC チャネル 0、チャネル 1 出力信号測定時の注意事項」をご覧ください。

設定と実行

1. サンプルコードをコンパイルし、ダウンロードします。
2. 必要な場合、マルチメータ用リードかオシロスコープ用プローブを DAC チャネル出力端子に取り付けます。
3. 「Reset Go」をクリックしてソフトウェアを起動します。PC が Main 関数で停止した場合、F8 を押してレジュームしてください。
4. ブレークポイントを設定して、グローバル変数を確認します。

対応ボード

RSKRX113

DAC チャネル 0、チャネル 1 出力信号測定時の注意事項

- DAC チャネル 0（DA0）出力では、MCU の端子 2 にマップする I/O ポート PJ0 を使用します。

  RSKRX113 では、DA0 は SW1 と端子 2 を共用しています。DA0 のアナログ出力信号用に端子 2 を使用するには、0 オーム抵抗を R241 から R239 に移動する必要があります。これによって、JA1_13 または J1_2 を介して DA0 にアクセスが可能になります。ただし、リンクの設定をこのように変更した場合、SW1 は使用不可となりますのでご注意ください。
- DAC チャネル 1（DA1）出力では、MCU の端子 100 にマップする I/O ポート PJ2 を使用します。

  RSKRX113 では、JA1_14 を介して DA1 にアクセスできます。

  DA1 には J4_25 を介してもアクセスが可能です。
- GROUND には JA1_2 を介してアクセスできます（その横の端子 4 もグランド端子です）。

- RX113 は、DAVREFCR レジスタを使用し、3 種類の DAC 基準電圧に対応しています。
  1. AVCC0/AVSS0

     RSKRX113 ボードでは、AVCC0 および AVSS0 は、UC_VCC (typ. 3.3V)および GROUND にそれぞれ接続されます。
  2. 内部基準電圧/AVSS0

     内部基準電圧は typ. 1.4 V、AVSS0 は GROUND です。
  3. VREFH/VREFL

     RSKRX113 ボードでは、VREFH/VREFL は接続されていません。VREFH/VREFL には J4_18 (CON_VREFH)および J4_17 (CON_VREFL)が使用されます。VREFH/VREFL を DAC の基準電圧として使用するには、以下の条件が必要となります。

     - I/O 端子 P41 と P42 は、MPC を介してアナログ端子として設定してください。
     - VREFH/VREEFL は HIGH/LOW 供給電圧に接続してください。

       これらの信号の設定に使用するオプションリンクの詳細は、「RX113 グループ Renesas Starter Kit ユーザーズマニュアル（R20UT2756JG0101）」の DAC 設定のセクションをご覧ください。
     - 代替オプション
       - J4_17 (CON_VREFL)には J3_12 (GROUND)を使用
       - J4_18 (CON_VREFH)には J3_10 (UC_VCC, 3.3V)を使用

4.2 dac_demo_rskrx231

dac_demo_rskrx231 は、RSKRX231、RX231 D/A コンバータ（R12DAA）対応の DAC FIT モジュール（r_dac_rx）のシンプルなデモプロジェクトです。デモでは r_dac_rx API を使用して、DAC の起動、設定、書き込みを行います。LOW、MEDIUM、または HIGH レベルの値が DAC チャネル 1 に書き込まれると、連続的なループに 1 秒間遷移します。LOW レベルの値が書き込まれると LED0（緑）が点灯し、MEDIUM レベルの値が書き込まれると LED1（オレンジ）が、HIGH レベルの値が書き込まれると LED2（赤）が点灯します。

RSKRX231 ボード使用時の DAC 出力チャネル信号のアクセスおよび設定、また基準電圧に関する詳細については、以下の「DAC チャネル 0、チャネル 1 出力信号測定時の注意事項」をご覧ください。

設定と実行

1. サンプルコードをコンパイルし、ダウンロードします。
2. 必要な場合、マルチメータ用リードかオシロスコープ用プローブを DAC チャネル出力端子に取り付けます。
3. 「Reset Go」をクリックしてソフトウェアを起動します。PC が Main 関数で停止した場合、F8 を押してレジュームしてください。
4. ブレークポイントを設定して、グローバル変数を確認します。

対応ボード

RSKRX231

DAC チャネル 0、チャネル 1 出力信号測定時の注意事項

- DAC チャネル 0（DA0）出力では、MCU の端子 2 にマップする I/O ポート P03 を使用します。DA0 には J1_2 を介してアクセスが可能です。
- DAC チャネル 1（DA1）出力では、MCU の端子 100 にマップする I/O ポート P05 を使用します。DA1 には JA1_14（または J4_25）を介してアクセスが可能です。
- GROUND には JA1_2 を介してアクセスできます（その横の端子 4 もグランド端子です）。
- RX231 は、DAVREFCR レジスタを使用し、3 種類の DAC 基準電圧に対応しています。

1. AVCC0/AVSS0

   RSKRX231 ボードでは、AVCC0 および AVSS0 は、UC_VCC (typ. 3.3V)および GROUND にそれぞれ接続されます。

2. 内部基準電圧/AVSS0

   内部基準電圧は typ. 1.4 V、AVSS0 は GROUND です。

3. VREFH/VREFL

   RSKRX231 ボードでは、VREFH/VREFL は UC_VCC (typ. 3.3V)および GROUND に接続されています。下記の 0 オーム抵抗を移動した後で、CON_VREFH (J1_1)および CON_VREFL (J1_3)を使用して、外部基準電圧に接続できます。

   - CON_VREFH: R68 を R67 に移動
   - CON_VREFL: R65 を R66 に移動

4.3 dac_demo_rskrx64m

dac_demo_rskrx64m は、RSKRX64M、RX64M D/A コンバータ（R12DA）対応の DAC FIT モジュール（r_dac_rx）のシンプルなデモプロジェクトです。デモでは r_dac_rx API を使用して、DAC の起動、設定、書き込みを行います。LOW、MEDIUM、または HIGH レベルの値が DAC チャネル 0 に書き込まれると、連続的なループに 1 秒間遷移します。LOW レベルの値が書き込まれると LED1 が点灯し、MEDIUM レベルの値が書き込まれると LED2 が、HIGH レベルの値が書き込まれると LED3 が点灯します。

RSKRX64M ボード使用時の DAC 出力チャネル信号のアクセスおよび設定、また基準電圧に関する詳細については、以下の「DAC チャネル 0、チャネル 1 出力信号測定時の注意事項」をご覧ください。

設定と実行

1. サンプルコードをコンパイルし、ダウンロードします。
2. 必要な場合、マルチメータ用リードかオシロスコープ用プローブを DAC チャネル出力端子に取り付けます。
3. 「Reset Go」をクリックしてソフトウェアを起動します。PC が Main 関数で停止した場合、F8 を押してレジュームしてください。
4. ブレークポイントを設定して、グローバル変数を確認します。

対応ボード

RSKRX64M

DAC チャネル 0、チャネル 1 出力信号測定時の注意事項

- DAC チャネル 0（DA0）出力では、MCU の端子 4 にマップする I/O ポート P03 を使用します。

  RSKRX64M では、DA0 は LED0 と端子 4 を共用しています。DA0 のアナログ出力信号用に端子 4 を使用するには、0 オーム抵抗を R277 から R189 に移動する必要があります。これによって、JA1_13 を介して DA0 にアクセスが可能になります。ただし、リンクの設定をこのように変更した場合、LED0 は使用不可となりますのでご注意ください。

- DAC チャネル 1（DA1）出力では、MCU の端子 2 にマップする I/O ポート P05 を使用します。

  RSKRX64M では、DA1 は LED1 と端子 2 を共用しています。DA1 のアナログ出力信号用に端子 2 を使用するには、0 オーム抵抗を R280 から R188 に移動する必要があります。これによって、JA1_14 を介して DA1 にアクセスが可能になります。ただし、リンクの設定をこのように変更した場合、LED1 は使用不可となりますのでご注意ください。

- RSKRX64M ボードでは、DA 基準電圧 AVCC1 (VREFH)および AVSS1 (VREFL)は、UC_VCC (typ. 3.3V)および GROUND にそれぞれ接続されます。

4.4 dac_demo_rskrx71m

dac_demo_rskrx71m は、RSKRX71M、RX71M D/A コンバータ（R12DA）対応の DAC FIT モジュール（r_dac_rx）のシンプルなデモプロジェクトです。デモでは r_dac_rx API を使用して、DAC の起動、設定、書き込みを行います。LOW、MEDIUM、または HIGH レベルの値が DAC チャネル 0 に書き込まれると、連続的なループに 1 秒間遷移します。LOW レベルの値が書き込まれると LED1 が点灯し、MEDIUM レベルの値が書き込まれると LED2 が、HIGH レベルの値が書き込まれると LED3 が点灯します。

RSKRX71M ボード使用時の DAC 出力チャネル信号のアクセスおよび設定、また基準電圧に関する詳細については、以下の「DAC チャネル 0、チャネル 1 出力信号測定時の注意事項」をご覧ください。

設定と実行

1. サンプルコードをコンパイルし、ダウンロードします。
2. 必要な場合、マルチメータ用リードかオシロスコープ用プローブを DAC チャネル出力端子に取り付けます。
3. 「Reset Go」をクリックしてソフトウェアを起動します。PC が Main 関数で停止した場合、F8 を押してレジュームしてください。
4. ブレークポイントを設定して、グローバル変数を確認します。

対応ボード

RSKRX71M

DAC チャネル 0、チャネル 1 出力信号測定時の注意事項

- DAC チャネル 0（DA0）出力では、MCU の端子 4 にマップする I/O ポート P03 を使用します。

  RSKRX71M では、DA0 は LED0 と端子 4 を共用しています。DA0 のアナログ出力信号用に端子 4 を使用するには、0 オーム抵抗を R277 から R189 に移動する必要があります。これによって、JA1_13 を介して DA0 にアクセスが可能になります。ただし、リンクの設定をこのように変更した場合、LED0 は使用不可となりますのでご注意ください。

- DAC チャネル 1（DA1）出力では、MCU の端子 2 にマップする I/O ポート P05 を使用します。

  RSKRX71M では、DA1 は LED1 と端子 2 を共用しています。DA1 のアナログ出力信号用に端子 2 を使用するには、0 オーム抵抗を R280 から R188 に移動する必要があります。これによって、JA1_14 を介して DA1 にアクセスが可能になります。ただし、リンクの設定をこのように変更した場合、LED1 は使用不可となりますのでご注意ください。

- RSKRX71M ボードでは、DA 基準電圧 AVCC1 (VREFH)および AVSS1 (VREFL)は、UC_VCC (typ. 3.3V)および GROUND にそれぞれ接続されます。

4.5 ワークスペースにデモを追加する

デモプロジェクトは、本アプリケーションノートで提供されるファイルの FITDemos サブディレクトリにあります。ワークスペースにデモプロジェクトを追加するには、「ファイル」→「インポート」を選択し、「インポート」ダイアログから「一般」の「既存プロジェクトをワークスペースへ」を選択して「次へ」ボタンをクリックします。「インポート」ダイアログで「アーカイブ・ファイルの選択」ラジオボタンを選択し、「参照」ボタンをクリックして FITDemos サブディレクトリを開き、使用するデモの zip ファイルを選択して「終了」をクリックします。

テクニカルアップデートの対応について

本モジュールは以下のテクニカルアップデートの内容を反映しています。

- 対応しているテクニカルアップデートはありません。

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
2.50	2015.09.30	—	初版発行
2.60	2015.10.01	—	FIT モジュールの RX130 グループ対応
2.70	2015.12.01	— 1, 6 3 8, 9, 12, 13 15	FIT モジュールの RX230 グループ対応 アプリケーションノート「ボードサポートパッケージモジュール Firmware Integration Technology」のドキュメント番号変更 「2.1 ハードウェアの要求」、「2.2 ハードウェアリソースの要求」：“DAA”の記載を削除 「3.3 R_DAC_Open()」、「3.6 R_DAC_Control()」：“Parameters”および “Example”に記載のコードを修正 「4. デモプロジェクト」追加
2.80	2016.02.01	— 20	FIT モジュールの RX24T グループ対応 「テクニカルアップデートの対応について」追加
2.91	2016.10.01	— 5 10 12,13	FIT モジュールの RX65N グループ対応 ROM,RAM およびスタックのコードサイズ変更 アンプを使用する時の注意事項を追加 Description、Example 記述変更 Special Notes 記述追加
3.00	2017.02.28	— 3 10, 11, 14 ﾌﾟﾛｸﾞﾗﾑ	FIT モジュールの RX24T グループ（ROM 512KB 版を含む）、RX24U グループ対応 「2.5 対応ツールチェーン」に RXC v2.06.00 を追加 3.3 R_DAC_Open()、3.4 R_DAC_Close()、 3.6 R_DAC_Control(): Special Notes に、D/A A/D 同期変換有効設定時の注意事項を追加 RX64M、RX71M、RX65N において、R_DAC_Open 関数で同期対象ユニットにユニット 0 を設定、かつ D/A A/D 同期変換を有効に設定した場合はエラーとなるように修正

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、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、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。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、その他の不適切に使用しないでください。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、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、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。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
   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. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
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. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。

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。

(Rev.3.0-1 2016.11)

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/

Colophon 5.0