RL78/G24

FAA オーバーサンプリング A/D 変換ライブラリ 導入ガイド

要旨

本サンプルアプリケーションでは、Flexible Application Accelerator（FAA）を用いたオーバーサンプリング A/D 変換ライブラリ（以下、オーバーサンプリング A/D 変換ライブラリ）について記載しています。FAA は RL78/G24 マイクロコントローラに内蔵されており、CPU とは独立したプロセッサとして動作します。

動作確認デバイス

RL78/G24

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

・RL78/G24 ユーザーズマニュアル　ハードウェア編（R01UH0961）

・RL78 ファミリ　ユーザーズマニュアル　ソフトウェア編（R01US0015）

・RL78/G24 Fast Prototyping Board ユーザーズマニュアル（R20UT5091）

・RL78 スマート・コンフィグレータ　ユーザーガイド：e^2 studio 編（R20AN0579）

・RL78 スマート・コンフィグレータ　ユーザーガイド：CS+編（R20AN0580）

・フレキシブル・アプリケーション・アクセラレータ(FAA)ツールガイド e^2 studio 編（R01AN7095）

・フレキシブル・アプリケーション・アクセラレータ(FAA)ツールガイド CS＋編（R01AN7094）

目次

1. オーバーサンプリング A/D 変換ライブラリの特徴

本章では、オーバーサンプリング A/D 変換ライブラリの特徴について説明します。

1.1 オーバーサンプリング A/D 変換制御

本ライブラリでは、指定した 1 つのアナログ入力チャネルのオーバーサンプリングを行います。

オーバーサンプリングとは、A/D コンバータの分解能を疑似的に向上させる手法です。A/D コンバータの識別可能な最小アナログ入力電圧、つまり、デジタル出力 1 ビットあたりのアナログ入力電圧の比率を 1LSB といい、この LSB が A/D 変換の分解能を表しています。本ライブラリでは 12 ビットの A/D 変換を行い、変換結果をバッファに格納し、そのバッファを使ってオーバーサンプリング処理を行うことでノイズを低減し、より正確で分解能の高い A/D 変換結果となります。また、パラメータとして、A/D 変換時間とアナログ入力チャネルの切り換えが可能です。

1.2 Flexible Application Accelerator（FAA）

オーバーサンプリング A/D 変換制御を実現するための処理は FAA にて行われます。FAA は RL78/G24 マイクロコントローラに内蔵されており、CPU とは独立したプロセッサとして動作します。これにより、CPU を占有することなくオーバーサンプリング A/D 変換制御処理を行うことができます。

FAA の詳細情報については「RL78/G24 ユーザーズマニュアル ハードウェア編（R01UH0961）」をご覧ください。

2. オーバーサンプリング A/D 変換ライブラリを使用可能なツール

本ライブラリは、スマート・コンフィグレータによるコード生成機能によりソースコードとして提供されます。生成されたソースコードは統合開発環境上でユーザープロジェクトに追加、ビルドする形で組み込みを行います。表 2-1 に本ライブラリを使用するためのツール一覧を示します。利用手順については 5 オーバーサンプリング A/D 変換ライブラリの利用手順を参照してください。

表 2-1 オーバーサンプリング A/D 変換ライブラリを使用可能なツール

項目	内容
統合開発環境（IDE）	e^2 studio CS+ for CC
スマート・コンフィグレータ（SC）	Renesas Smart Configurator for RL78
コンパイラ	CC-RL

3. オーバーサンプリング A/D 変換ライブラリのフォルダ構成

以下に本ライブラリのフォルダ構成を示します。各ファイルはスマート・コンフィグレータによるコード生成機能により提供されます。

表 3-1 フォルダ構成

フォルダ、ファイル名	説明
smc_gen	SC 生成フォルダ
\{ConfigName}	FAA コンポーネントフォルダ
{ConfigName}_common.c	FAA 共通機能のソースファイル
{ConfigName}_common.h	FAA 共通機能のヘッダファイル
{ConfigName}_common.inc	FAA 共通機能のインクルードファイル
{ConfigName}_ADC_OverSampling.c	オーバーサンプリング A/D 変換機能のソースファイル
{ConfigName}_ADC_OverSampling.h	オーバーサンプリング A/D 変換機能のヘッダファイル
{ConfigName}_src.dsp	FAA DSP アセンブラソースファイル

4. 使用リソース

4.1 周辺機能

本ライブラリは以下の周辺機能を使用します。

- フレキシブル・アプリケーション・アクセラレータ
- A/D 変換コンバータ

4.2 ROM/RAM サイズ

以下のオプションを使用してビルドした際の ROM/RAM サイズを参考として記します。

コンパイラオプション

-cpu=S3 -character_set=utf8 –Odefault

リンク・オプション

-NOOPtimize

表 4-1 ROM/RAM サイズ [byte]

ROM	RAM	FAACODE	FAADATA
1843	7	92	1060

5. オーバーサンプリング A/D 変換ライブラリの利用手順

本ライブラリの利用手順を以下に示します。

5.1 CC-RL 環境

5.1.1 統合開発環境

e² studio、CS+は Renesas ホームページからダウンロードしてください。

[Renesas ホームページ]

https://www.renesas.com/ja/products/software-tools/tools.html

e² studio の基本操作については以下ユーザーズマニュアルを参照してください。

- e² studio 統合開発環境 ユーザーズマニュアル 入門ガイド（R20UT2858）

CS+の基本操作については以下ユーザーズマニュアルを参照してください。

- CS+ 統合開発環境 ユーザーズマニュアル プロジェクト操作編（R20UT4691)

5.1.2 スマート・コンフィグレータ

CS+を使用する場合、下記 URL から「RL78 スマート・コンフィグレータ」および「CS+ RL78 スマート・コンフィグレータ通信プラグイン」をダウンロードしてください。CS+ RL78 スマート・コンフィグレータ通信プラグインは、スマート・コンフィグレータで生成したソースを CS+に登録するために必要です。

https://www.renesas.com/rl78-smart-configurator

インストーラ起動後、インストーラの手順に従ってインストールしてください。インストールは管理者権限で行ってください。

5.1.3 オーバーサンプリング A/D 変換ライブラリ

スマート・コンフィグレータ上で、オーバーサンプリング A/D 変換ライブラリを利用する手順について説明します。スマート・コンフィグレータの基本的な操作方法については下記ユーザーガイドを参照してください。

- RL78 スマート・コンフィグレータ ユーザーガイド：e² studio 編（R20AN0579）
- RL78 スマート・コンフィグレータ ユーザーガイド：CS+編（R20AN0580）

5.1.3.1 スマート・コンフィグレータの起動

新規プロジェクトの作成または既存プロジェクトの読み込みを行い、スマート・コンフィグレータを起動してください。

5.1.3.2 クロック設定

「スマート・コンフィグレータビュー」の「クロック」ページを選択し、「タイマークロック」および「fCLK」出力を下記のように設定してください。

- タイマークロック：96000[kHz]
- fCLK：48000[kHz]

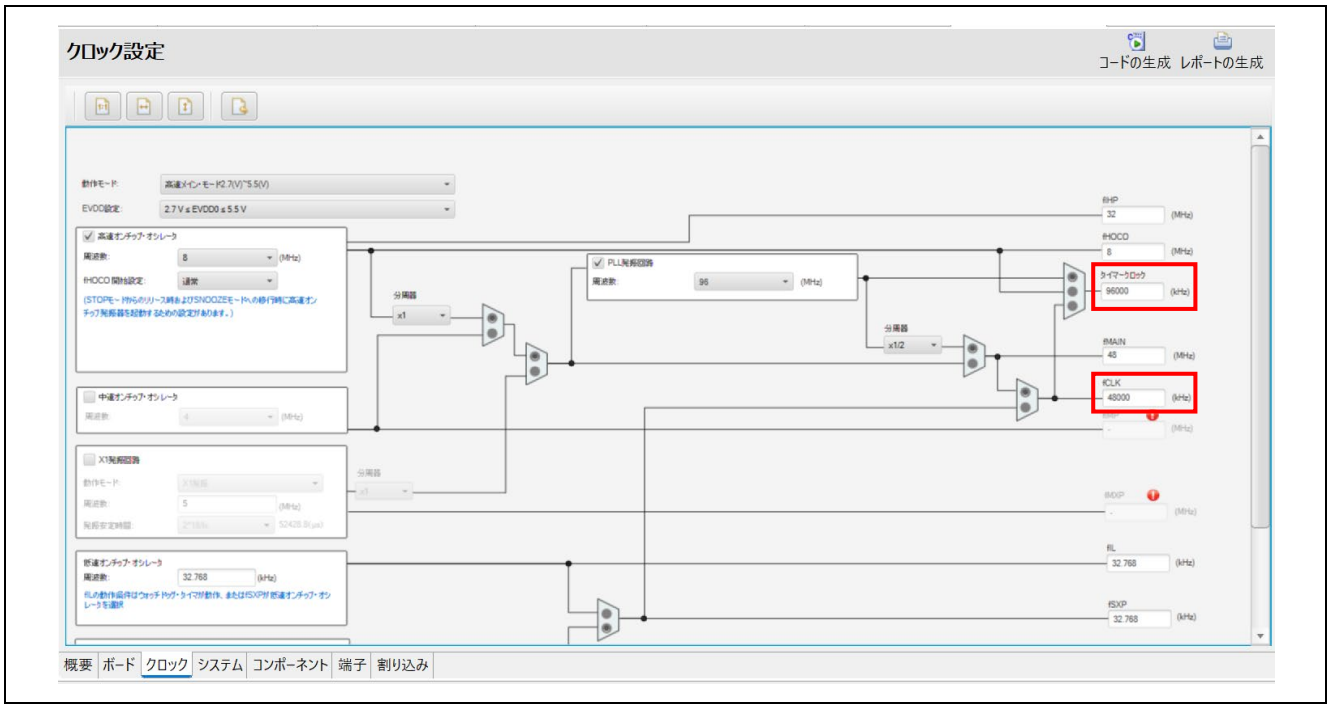

図 5-1 クロック設定

5.1.3.3 FAA コンポーネントの追加

「スマート・コンフィグレータビュー」の「コンポーネント」ページを選択し、「コンポーネントの追加」ボタンを押下してください。

次に「ソフトウェアコンポーネントの選択」画面より「フレキシブル・アプリケーション・アクセラレータ」コンポーネントを追加してください。

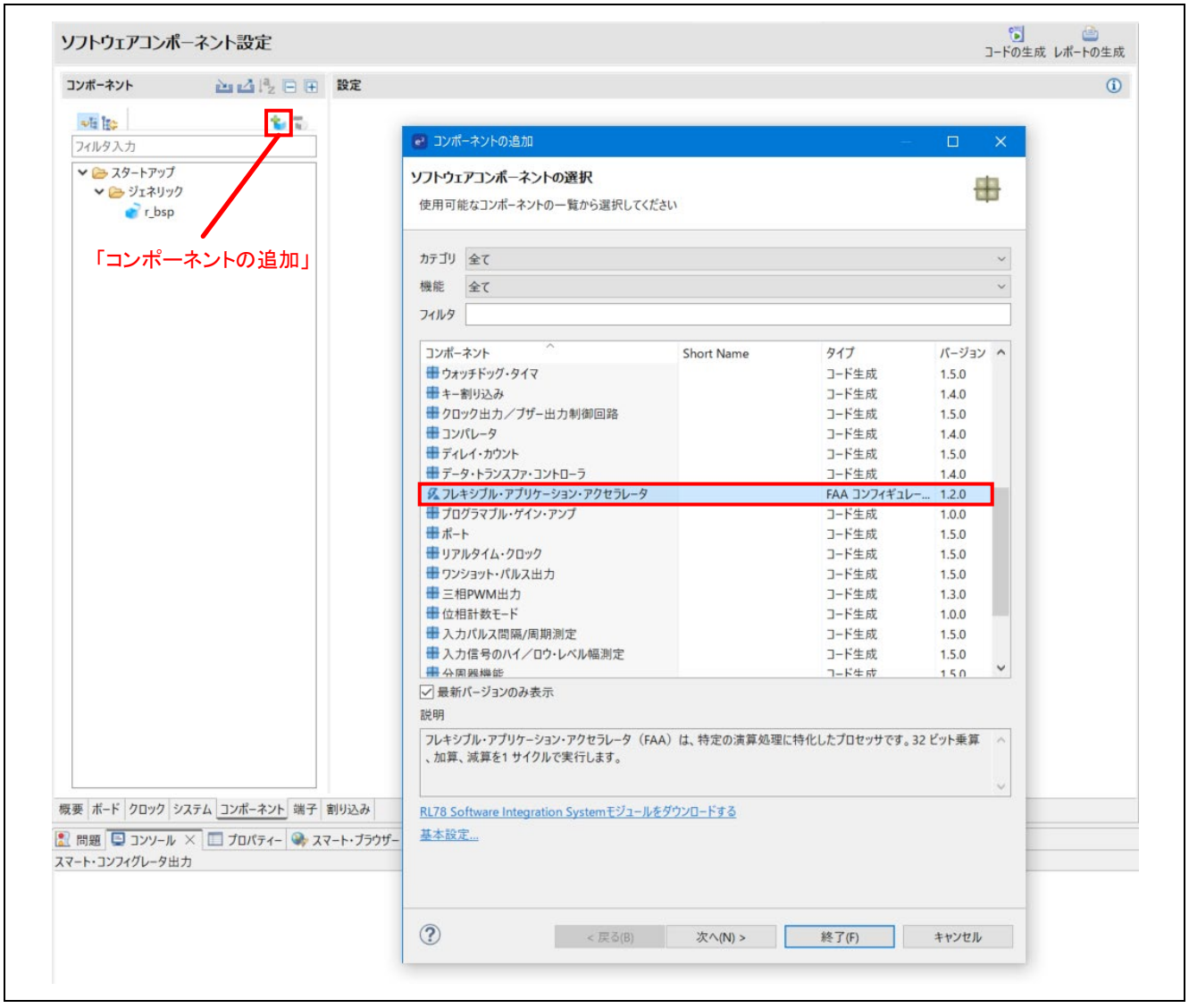

図 5-2 FAA コンポーネントの追加

5.1.3.4 FAA モジュールのダウンロード

画面上に表示されている「Please download FAA data」をクリックするとダウンロード可能な FAA モジュールが表示されます。「OverSampling Library」を選択してダウンロードしてください。

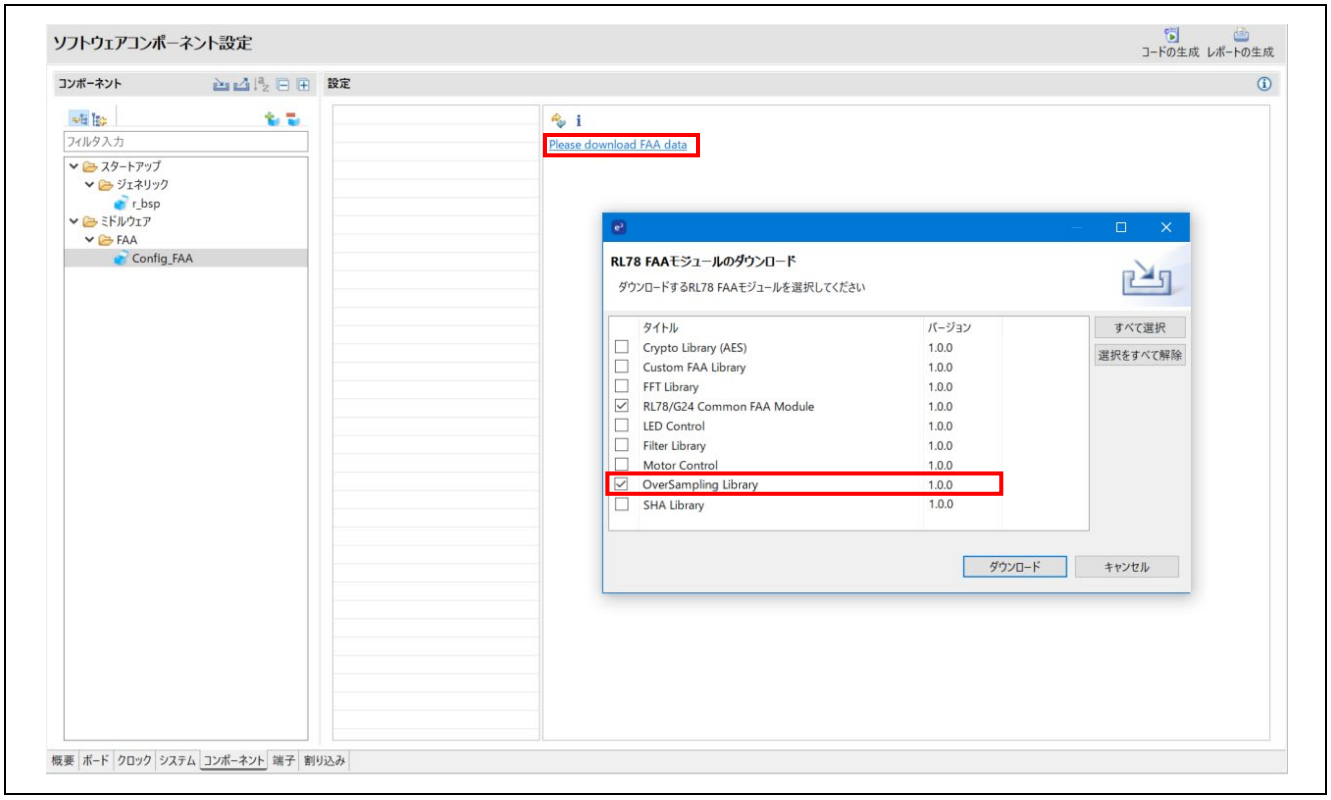

図 5-3 FAA モジュールのダウンロード

5.1.3.5 FAA モジュールのコンフィグレーション

ダウンロードされた FAA モジュールの一覧より「ADC OverSampling」モジュールを選択すると、コンフィグレーション画面が表示されます。

ユーザー環境に応じてコンフィグレーション設定を行ってください。表 5-1 にコンフィグレーション項目の詳細を示します。

表 5-1 スマート・コンフィグレータ設定項目一覧

項目	取りうる値	説明
ADC VREF(+)	VDD/AVREFP	A/D コンバータのプラス側の基準電圧を選択します。
ADC VREF(-)	VSS/AVREFM	A/D コンバータのマイナス側の基準電圧を選択します。

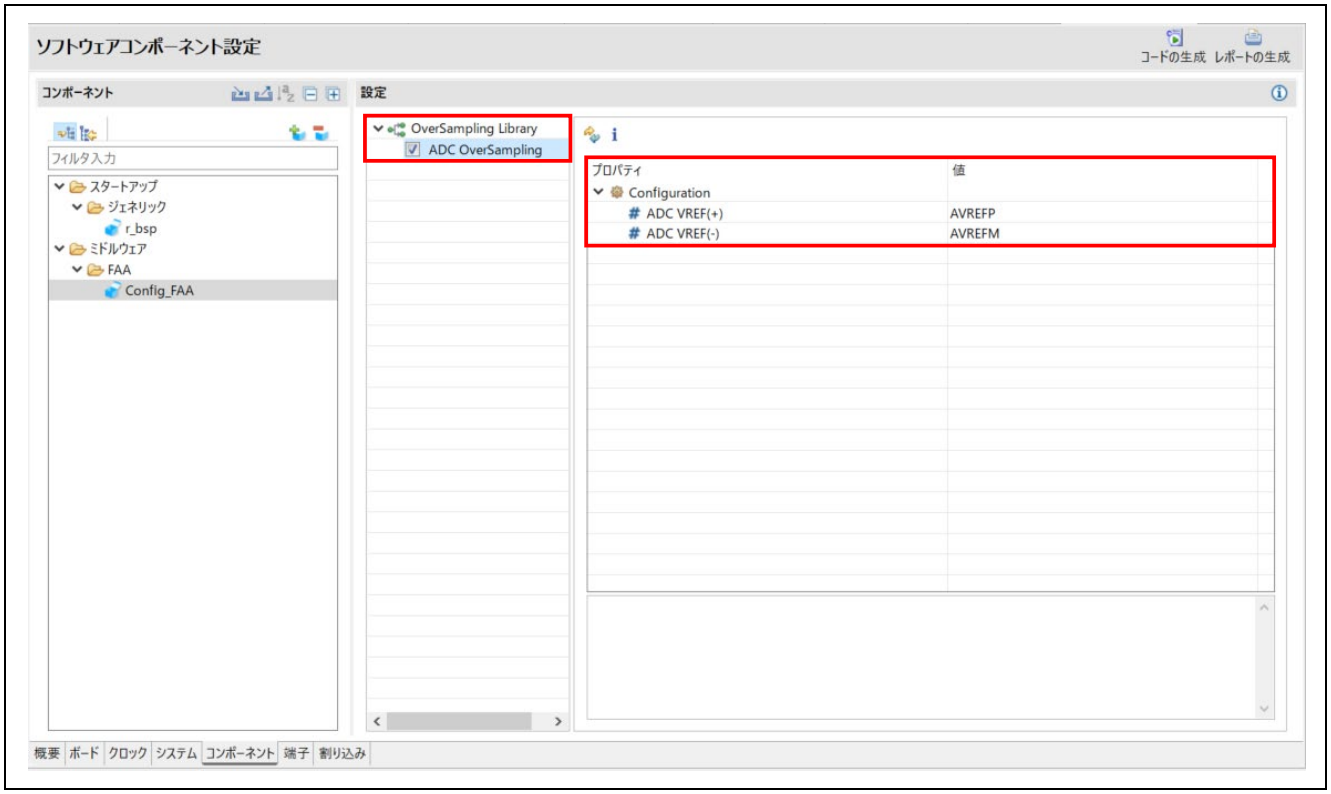

図 5-4 FAA モジュールのコンフィグレーション

5.1.3.6 A/D コンバータコンポーネントの追加

本ライブラリはスマート・コンフィグレータの A/D コンバータのコンポーネントを使用するため、A/D コンバータコンポーネントを追加してください。動作モードは「アドバンスド・モード」を選択してください。

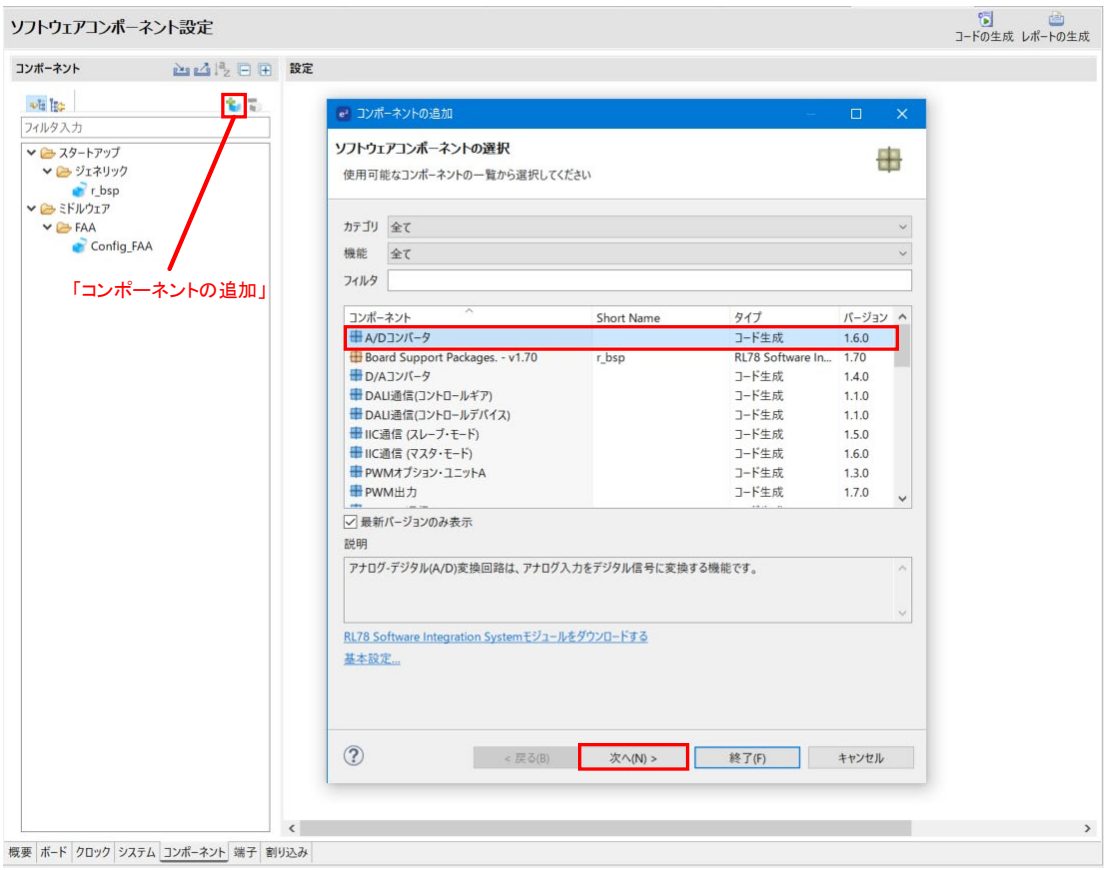

図 5-5 A/D コンバータコンポーネントの追加

5.1.3.7 コード生成

コンフィグレーション設定後、「コード生成」ボタンを押下することで、オーバーサンプリング A/D 変換ライブラリのソースコードが生成されます。生成されたソースコードは e^2 studio、または CS+上のユーザープロジェクトに自動登録されます。

ユーザーアプリケーションにて、本ライブラリより提供される API 関数のコール処理を実装してください。各種 API 関数の詳細については 7.4 API 関数仕様をご覧ください。

6. ライブラリ仕様（FAA プログラム）

本ライブラリのオーバーサンプリング A/D 変換制御について説明します。

制御処理は”{ConfigName}_src.dsp”ファイルに実装されており、FAA によって実行されます。

6.1 定数一覧

表 6-1 に本ライブラリが提供する定数一覧を示します。

表 6-1 ライブラリで使用している定数

定数名	設定値	内容
_C_ADControl_FAAAP_ADM3	#ADM3_PTR（014H）	ADM3 レジスタに FAA・アドレス・ポインタでアクセス時の設定値
_C_ADControl_ADTRSWT	80H	ソフトウェア・トリガを発生させるための設定値（ADM3 設定値）
_C_ADControl_FAAAP_ADINST	#ADINTST_PTR（029H）	ADINTST レジスタに FAA・アドレス・ポインタでアクセス時の設定値
_C_ADControl_ADINTST_ST0S	02H	A/D 変換成功を判定する値
_C_ADControl_FAAAP_ADCR0	#ADCR0_PTR（020H, 021H）	ADCR0 レジスタに FAA・アドレス・ポインタでアクセス時の設定値
_C_ADControl_ADINTST_ST0S_CLEAR	01H	A/D 変換結果のステータスをクリアするための設定値

6.2 変数一覧

表 6-2 に本ライブラリが提供する変数一覧を示します。

表 6-2 ライブラリで使用している変数

サイズ	変数名	内容
4 バイト	_V_OverSampling_Result	256 回分の A/D 変換結果をオーバーサンプリングした値の格納変数
4 バイト	_V_OverSampling_Start	256 回分の A/D 変換値の格納変数
4 バイト	_V_OverSampling_End	256 回分の A/D 変換値の格納用バッファ最後尾を示す変数
4 バイト	_V_ADControl_ADINTST_ST0S	A/D 変換完了確認
4 バイト	_V_Buffer_Full	256 回分の A/D 変換値の格納用バッファ最後尾のアドレス

6.3 関数一覧

表 6-3 に本ライブラリが提供する関数、処理を示します。

表 6-3 処理一覧

ラベル名	概要	ソースファイル
_P_Pointer_Init	アドレス・ポインタの初期化	Config_FAA_src.dsp
_P_ADControl	A/D 変換処理	Config_FAA_src.dsp
_P_OverSampling	オーバーサンプリング処理	Config_FAA_src.dsp

6.4 関数仕様

本ライブラリの関数仕様を示します。

FAA プログラム

[ラベル名] _P_Pointer_Init

概要	アドレス・ポインタの初期化
ヘッダ	Config_FAA_common.inc
宣言	ー
説明	アドレス・ポインタの初期化を行います。
引数	なし
リターン値	なし
備考	なし

[ラベル名] _P_ADControl

概要	A/D 変換処理
ヘッダ	Config_FAA_common.inc
宣言	ー
説明	指定したアナログ入力チャネルの A/D 変換を行います。
引数	なし
リターン値	なし
備考	なし

[ラベル名] _P_OverSampling

概要	オーバーサンプリング処理
ヘッダ	Config_FAA_common.inc
宣言	ー
説明	256 回分の A/D 変換結果を用いてオーバーサンプリング処理後、SHDMEM に格納し、FAA 終了割り込みを発生させます。
引数	なし
リターン値	なし
備考	なし

6.5 制御フロー

オーバーサンプリング A/D 変換制御のフローを以下に示します。

6.5.1 ポインタ初期化処理

図 6-1 にポインタ初期化処理のフローチャートを示します。

API 関数「R_{ConfigName}_ADC_OverSampling_Start」コールにより、ポインタ初期化処理が実行されます。

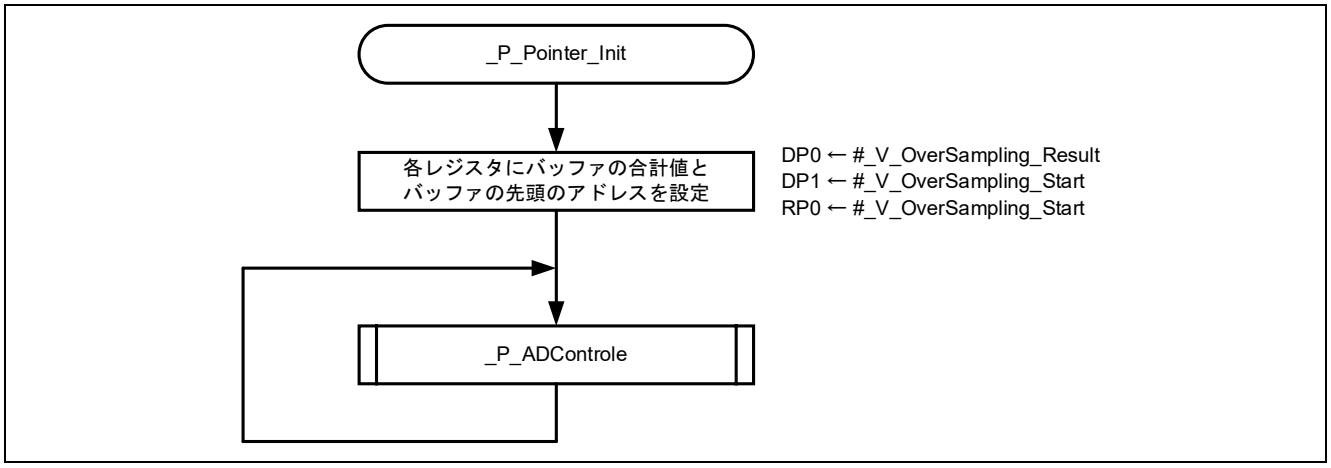

図 6-1 ポインタ初期化処理

6.5.2 A/D 変換処理

図 6-2 に A/D 変換処理のフローチャートを示します。

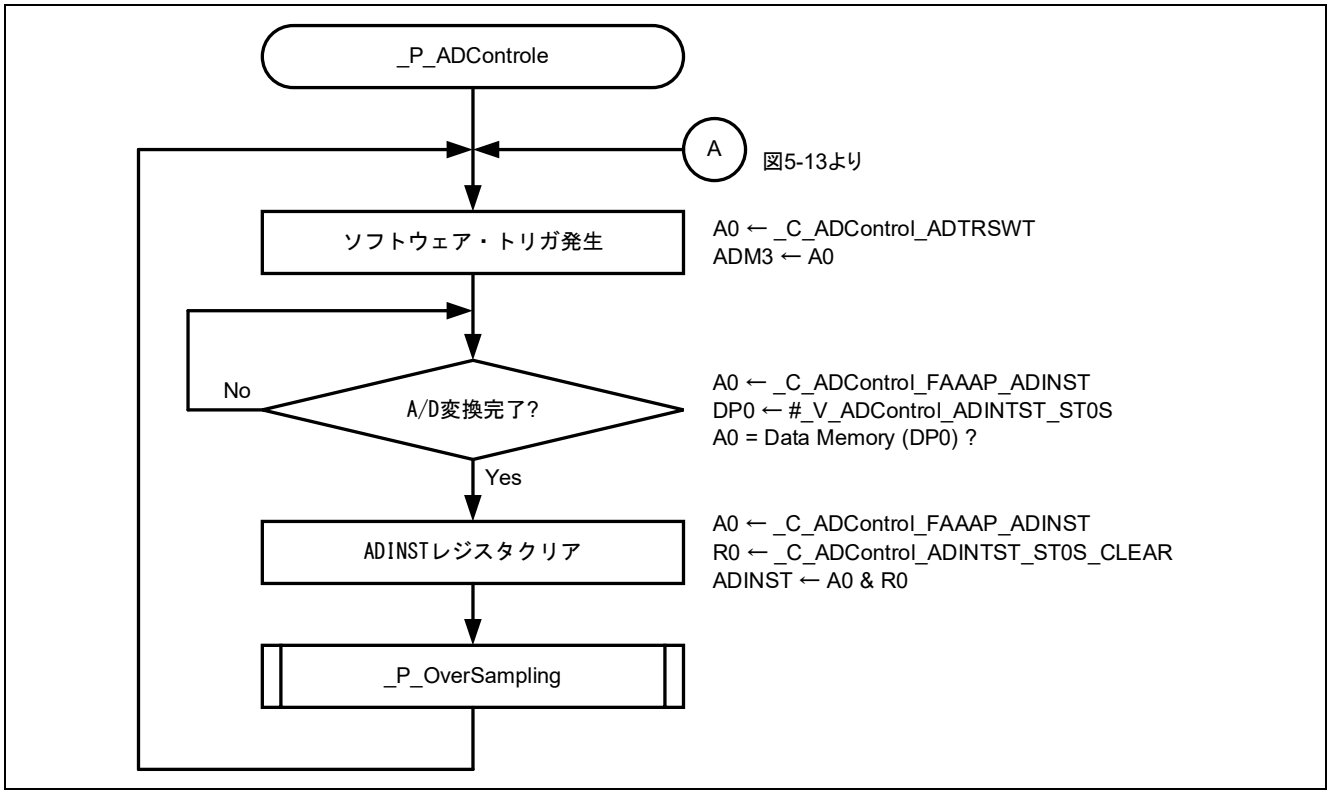

図 6-2 A/D 変換処理

6.5.3 オーバーサンプリング処理

図 6-3 にオーバーサンプリング処理のフローチャートを示します。

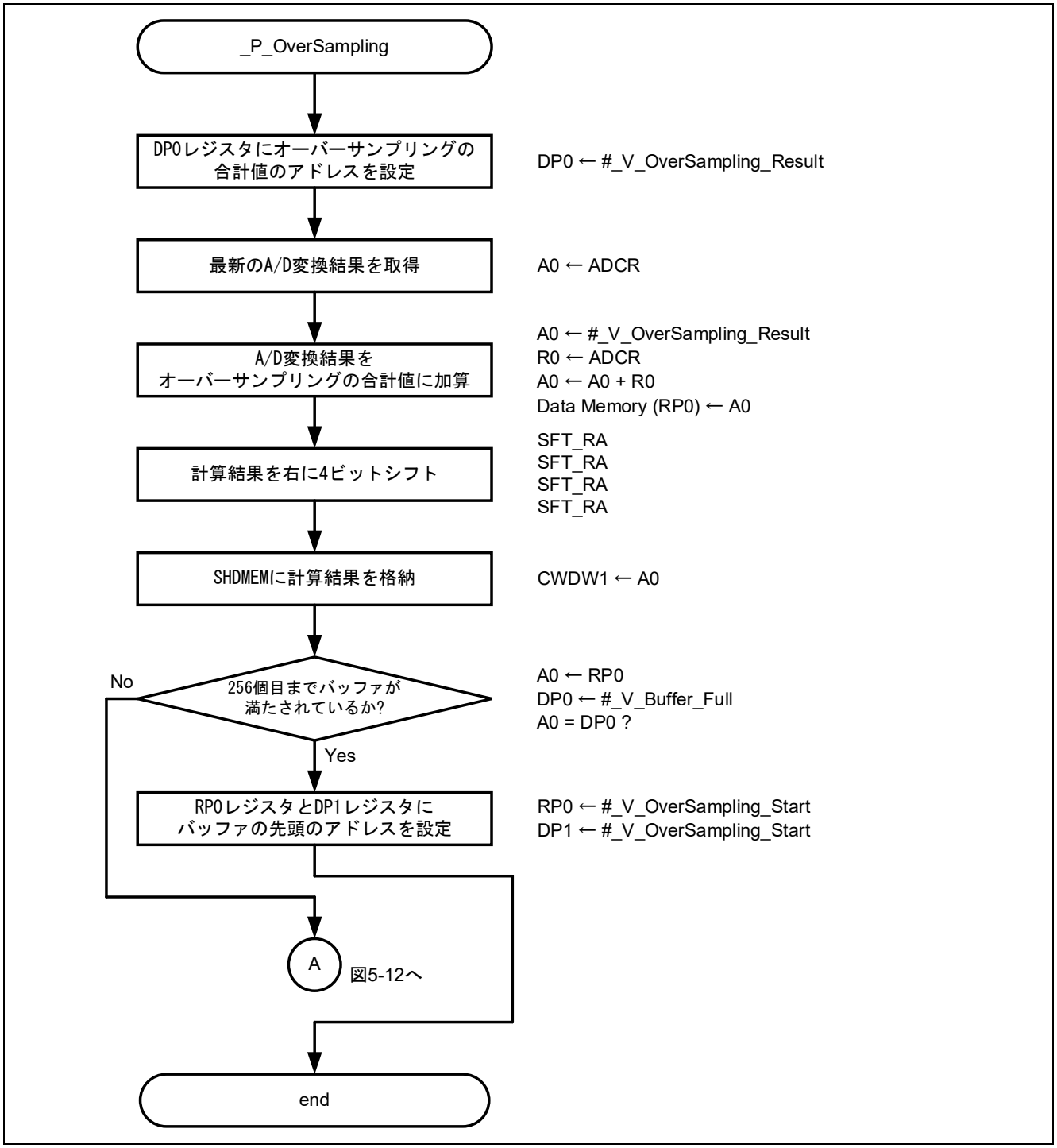

図 6-3 オーバーサンプリング処理

7. API 情報

7.1 定数一覧

表 7-1 に本ライブラリが提供する定数一覧を示します。

表 7-1 ライブラリで使用している定数

定数名	設定値	内容
ENABLE_CALLBACK_FUNCTION	1	コールバック関数呼び出し許可 （0：コールバック関数呼び出し禁止）

7.2 変数一覧

表 7-2 に本ライブラリが提供する変数一覧を示します。

表 7-2 ライブラリで使用している変数

変数名	型	内容	使用関数
g_oversampling_result	uint16_t	オーバーサンプリング結果を格納	Config_FAA_ADC_OverSampling.c
g_faa_stop_flg	bool	オーバーサンプリング終了フラグ	Config_FAA_ADC_OverSampling.c

7.3 API Typedef 定義

本ライブラリが提供する Typedef 定義について説明します。

7.3.1 e_ad_convert_time_t

この Typedef は A/D 変換時間を定義します。A/D コンバータの A/D 変換時間を指定するために使用されます。

```
typedef enum
{
    AD_CONVERT_TIME_1600 = 0x00,    // 1600/fCLK
    AD_CONVERT_TIME_800  = 0x08,    // 800/fCLK
    AD_CONVERT_TIME_400  = 0x10,    // 400/fCLK
    AD_CONVERT_TIME_200  = 0x18,    // 200/fCLK
    AD_CONVERT_TIME_100  = 0x20,    // 100/fCLK
    AD_CONVERT_TIME_50   = 0x28,    // 50/fCLK
} e_ad_convert_time_t;
```

7.3.2 e_oversampling_channel_t

この Typedef はアナログ入力チャネルを定義します。A/D コンバータのオーバーサンプリング対象のチャネルを指定するために使用されます。

```
typedef enum
{
    OVERSAMPLING_CHANNEL_0,
    OVERSAMPLING_CHANNEL_1,
    OVERSAMPLING_CHANNEL_2,
・・・
    OVERSAMPLING_CHANNEL_6,
    OVERSAMPLING_CHANNEL_7,
    OVERSAMPLING_CHANNEL_16 = 16U,
    OVERSAMPLING_CHANNEL_17,
・・・
    OVERSAMPLING_CHANNEL_30
} e_oversampling_channel_t;
```

7.4 API 関数仕様

本ライブラリが提供する API 関数について説明します。

API 関数名に含まれる”{ConfigName}”はスマート・コンフィグレータにて設定した FAA コンポーネントのコンフィグレーション名を示します。

7.4.1 R_{ConfigName}_ADC_OverSampling_Create

Format

void R_{ConfigName}_ADC_OverSampling_Create (CallbackFunction callback)

Parameters

callback コールバック関数へのポインタ

（備考）コールバック関数を使用しない場合は、NULL を設定してください。

Return Values

なし

Properties

{ConfigName}_ADC_OverSampling_Create.h にプロトタイプ宣言されています。

Description

この関数は、FAA で使用する SHDMEM の初期化処理、コールバック関数の関数ポインタへの設定、A/D コンバータの基準電圧の設定を実施します。

7.4.2 R_{ConfigName}_ADC_OverSampling_SetParameter

Format

void R_{ConfigName}_ADC_OverSampling_SetParameter (st_sampling_parameter_t *config)

Parameters

config.time	A/D 変換時間
config.channel	オーバーサンプリング対象のアナログ入力チャネル

Return Values

なし

Properties

{ConfigName}_ADC_OverSampling.h にプロトタイプ宣言されています。

Description

この関数は、A/D 変換時間とオーバーサンプリング対象のチャネルを設定します。

Example

```
config.time    = AD_CONVERT_TIME_50;
config.channel = OVERSAMPLING_CHANNEL_2;
R_Config_FAA_ADC_OverSampling_SetParameter(&config);
```

Notes:

変換時間は A/D コンバータの仕様の範囲内で設定してください。本ライブラリで設定可能な A/D 変換時間は下記の通りです。

システムクロック：48MHz

1. 「AD_CONVERT_TIME_1600」の場合
   変換時間 = $\frac{1600}{48\times10^6}$ ≒ 33.333μs
2. 「AD_CONVERT_TIME_800」の場合
   変換時間 = $\frac{800}{48\times10^6}$ ≒ 16.667μs
3. 「AD_CONVERT_TIME_400」の場合
   変換時間 = $\frac{400}{48\times10^6}$ ≒ 8.3333μs
4. 「AD_CONVERT_TIME_200」の場合
   変換時間 = $\frac{200}{48\times10^6}$ ≒ 4.1667μs
5. 「AD_CONVERT_TIME_100」の場合
   変換時間 = $\frac{100}{48\times10^6}$ ≒ 2.0833μs
6. 「AD_CONVERT_TIME_50」の場合
   変換時間 = $\frac{50}{48\times10^6}$ ≒ 1.0417μs

詳細は「RL78/G24 ユーザーズマニュアル　ハードウェア編（R01UH0961）」の「表 20-6 A/D 変換時間の選択（10/11）」を参照してください。

AVREFP を A/D コンバータの＋側の基準電圧として使用している場合、ANI0 を A/D 変換チャネルとして選択しないでください。また、AVREFM を A/D コンバータの－側の基準電圧として使用している場合は、ANI1 を A/D 変換チャネルとして選択しないでください。

7.4.3 R_{ConfigName}_ADC_OverSampling_Start

Format

FAA_Status_t R_{ConfigName}_ADC_OverSampling_Start (void)

Parameters

なし

Return Values

FAA_SUCCESS	FAA 処理成功
FAA_ALREADY_RUNNING	FAA 実行中

Properties

{ConfigName}_ADC_OverSampling.h にプロトタイプ宣言されています。

Description

この関数は、A/D コンバータへのバスアクセスを FAA に切り替え、FAA 処理を開始します。

7.4.4 R_{ConfigName}_ADC_OverSampling_Stop

Format

void R_{ConfigName}_ADC_OverSampling_Stop (void)

Parameters

なし

Return Values

なし

Properties

{ConfigName}_ADC_OverSampling.h にプロトタイプ宣言されています。

Description

この関数は、A/D コンバータへのバスアクセスを CPU に切り替え、FAA 処理を停止します。

7.4.5 R_{ConfigName}_ADC_OverSampling_Get

Format

void R_{ConfigName}_ ADC_OverSampling_Get (void)

Parameters

なし

Return Values

なし

Properties

{ConfigName}_ADC_OverSampling.h にプロトタイプ宣言されています。

Description

この関数は、オーバーサンプリング結果を取得します。オーバーサンプリング結果の格納先は SHDMEM の CWDW1L です。

7.4.6 R_{ConfigName}_ADC_OverSampling_INTFAAE_interrupt

Format

static void __near R_Config_FAA_ADC_OverSampling_INTFAAE_interrupt(void)

Parameters

なし

Return Values

なし

Properties

{ConfigName}_ADC_OverSampling.h にプロトタイプ宣言されています。

Description

この関数は、オーバーサンプリング処理終了割り込みであり、関数内でフラグをセットします。

オーバーサンプリング A/D 変換処理が終了したことを判定するには、以下の 2 つの方法があります。

① FAA 終了フラグをポーリングする
オーバーサンプリング終了割り込み関数内でセットされる FAA 終了フラグ（g_faa_stop_flg）をユーザープログラムでポーリングしてください。フラグの状態を確認することで処理終了を判定します。

② コールバック関数を登録する
「ENABLE_CALLBACK_FUNCTION」を有効にすることで、コールバック関数を使用できます。
FAA 終了割り込み発生時にライブラリからコールバック関数を呼び出します。このコールバック関数にユーザープログラムを追加することで処理終了時に必要な処理を実行可能です。

7.5 ライブラリ API の呼び出し手順

オーバーサンプリング A/D 変換ライブラリを使用して、1 つのアナログ入力チャネルをオーバーサンプリング処理する場合の API 関数の呼び出し手順例を示します。

ライブラリ API は太字で示しています。

表 7-3 API 関数の呼び出し手順例（1/2）

No.	処理	説明
開始処理		
(1)	EI();	CPU の割り込み要求受け付けを許可。
(2)	**R_Config_FAA_ADC_OverSampling_Create(oversampling_callback);**	SHDMEM の初期化を行う。コールバック関数を使用する場合は、コールバック関数を設定。
(3)	st_sampling_parameter_t config = {AD_CONVERT_TIME_50, OVERSAMPLING_CHANNEL_2};	A/D 変換時間とアナログ入力チャネルの設定用変数を宣言。
(4)	**R_Config_FAA_ADC_OverSampling_SetParameter(&config);**	A/D 変換時間とアナログ入力チャネルを設定。
(5)	R_Config_ADC_Set_OperationOn();	A/D コンバータを A/D 変換待機状態に設定。
(6)	R_Config_ADC_Start();	A/D コンバータを A/D トリガ待機状態に設定。
(7)	**R_Config_FAA_ADC_OverSampling_Start();**	A/D コンバータへのバスアクセスを FAA に切り替え、FAA 処理を開始。
(8)	FAA によるオーバーサンプリング A/D 変換の完了待ち	(a) コールバック関数を使用する場合 FAA によるオーバーサンプリング A/D 変換が完了後、FAA 終了割り込み(INTFAAE)が発生し、(2)で設定したコールバック関数が呼び出される。オーバーサンプリング A/D 変換が完了するまで、CPU は他のタスクを実行可能。 (b) コールバック関数を使用しない場合 オーバーサンプリング A/D 変換完了後、g_faa_stop_flg フラグがセットされる。g_faa_stop_flg フラグのセットをソフトウェアで検出し、検出後、g_faa_stop_flg フラグをクリアすること。
(9)	**R_Config_FAA_ADC_OverSampling_Get();**	オーバーサンプリング A/D 変換結果を SHDMEM から取得し、g_oversampling_result 変数に格納する。再度、オーバーサンプリング A/D 変換を行う場合は、(7)に戻る。

表 7-4 API 関数の呼び出し手順例（2/2）

No.	処理	説明
終了処理		
(10)	**R_Config_FAA_ADC_OverSampling_Stop();**	A/D コンバータへのバスアクセスを CPU に切り替え、FAA 処理を停止。
(11)	R_Config_ADC_Stop();	A/D コンバータを A/D 変換待機状態に設定。
(12)	R_Config_ADC_Set_OperationOff();	A/D コンバータを A/D 変換停止状態に設定。

8. サンプルアプリケーション

本章以降では、オーバーサンプリング A/D 変換ライブラリを使用したサンプルコードについて説明します。

8.1 仕様概要

本サンプルアプリケーションでは、CPU と FAA の 2 つのプロセッサを使用し、FAA によるオーバーサンプリング制御を行い、12 ビット以上の分解能を実現します。プログラム上で A/D 変換時間とアナログ入力チャネルの設定/切り替えが可能です。

CPU プログラムで、A/D コンバータの初期設定を行い、A/D 変換時間とアナログ入力チャネルの設定後、FAA の動作を開始します。FAA 終了割り込みで SHDMEM に格納されているオーバーサンプリングの結果を取得します。FAA の終了待機中はポートトグルを行います。

FAA プログラムでは、A/D コンバータの制御を行います。動作開始後、A/D 変換時間：50/fCLK、アナログ入力チャネル：P22 / ANI2 端子の A/D 変換を実行します。256 回分の A/D 変換結果に対してオーバーサンプリング処理を行い、SHDMEM に格納します。オーバーサンプリングが終了すると、FAA 終了割り込みが発生し、FAA は停止します。

CPU 側で A/D 変換時間：1600/fCLK、とアナログ入力チャネル：P120 / ANI19 端子に設定を切り換え、再度オーバーサンプリング処理を実行します。

表 8-1 に使用する周辺機能と用途を示します。

表 8-1 使用する周辺機能と用途

周辺機能	用途
フレキシブル・アプリケーション・アクセラレータ（FAA）	A/D コンバータの制御を行い、オーバーサンプリング処理を行う
A/D コンバータ （アドバンスド・モード ON）	アナログ入力端子のアナログ入力電圧を A/D 変換する
ポート（PORT）	CPU と FAA の同時動作を示すため、常時、ポートのトグルを行う。

図 8-1 にアドレス・バス選択レジスタ（ADBSEL）の設定を示します。

	15	14	13	12	11	10	9	8
ADBSEL	FAADIVSEL	0	TRNGSEL	PORTSEL	TKB32SEL	TKB31SEL	TKB30SEL	TRGSEL
設定値	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TRD0SEL	PWMOP SEL	TRXSEL	DACSEL	PGACMP SEL	ADCSEL	SAU0SEL	TAU0SEL
0	0	0	0	0	1	0	0

0	CPUからのバスアクセスを許可
1	FAAからのバスアクセスを許可

ADCSEL：A/D コンバータ

図 8-1 アドレス・バス選択レジスタ（ADBSEL）の設定

8.2 動作説明

本サンプルコードでは、A/D コンバータによる変換結果を 256 回分集め、オーバーサンプリングを行うことで、12 ビット以上の分解能を実現します。

FAA 終了割り込みでフラグをセットし、CPU が SHDMEM に格納されているオーバーサンプリング結果を取得します。

1. ［CPU プログラム］SHDMEM の初期化を行います。
2. ［CPU プログラム］A/D 変換時間とオーバーサンプリング対象のチャネルを設定します。
3. ［CPU プログラム］A/D コンバータへのバスアクセスを FAA に設定します。
4. ［CPU プログラム］FAA のスタックポインタ、FAA プログラムの開始アドレスを設定し、FAA の動作を開始させます。
5. ［FAA プログラム］P22 / ANI2 端子の A/D 変換を行い、256 回分の変換結果をオーバーサンプリングし、結果を SHDMEM に格納します。その後、FAA へのクロック供給を停止し、FAA の動作を終了します。
6. ［CPU プログラム］FAA 終了割り込みが発生し、フラグセット後、SHDMEM に格納された値を取得します。
7. ［CPU プログラム］A/D 変換時間とオーバーサンプリング対象のチャネルを切り換えます。
8. ［CPU プログラム］3～6 を繰り返します。

9. 動作確認条件

本サンプルアプリケーション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。

表 9-1 動作確認条件

項目	内容
使用マイコン	RL78/G24（R7F101GLG）
動作周波数	高速オンチップ・オシレータ・クロック（f_{HOCO}）：8MHz PLL 発振回路出力（f_{PLL}）：96MHz CPU／周辺ハードウェア・クロック（f_{CLK}）：48MHz
動作電圧	3.3V（2.7V～5.5V で動作可能） LVD0 動作（V_{LVD0}）：リセット・モード 立ち上がり時 TYP. 2.97V 立ち下がり時 TYP. 2.91V
統合開発環境（CS+）	ルネサスエレクトロニクス製 CS+ for CC V8.12.00
C コンパイラ（CS+）	ルネサスエレクトロニクス製 CC-RL V1.12.00
統合開発環境（e² studio）	ルネサスエレクトロニクス製 e² studio 2024-10 (24.10.0)
C コンパイラ（e² studio）	ルネサスエレクトロニクス製 CC-RL V1.14.00
スマート・コンフィグレータ	V.1.11.0
ボードサポートパッケージ（r_bsp）	V.1.70
エミュレータ	CS+、e² studio：COM ポート
使用ボード	RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ)

10. ハードウェア説明

10.1 ハードウェア構成例

図 10-1 に本サンプルアプリケーションのサンプルコードで使用するハードウェア構成例を示します。

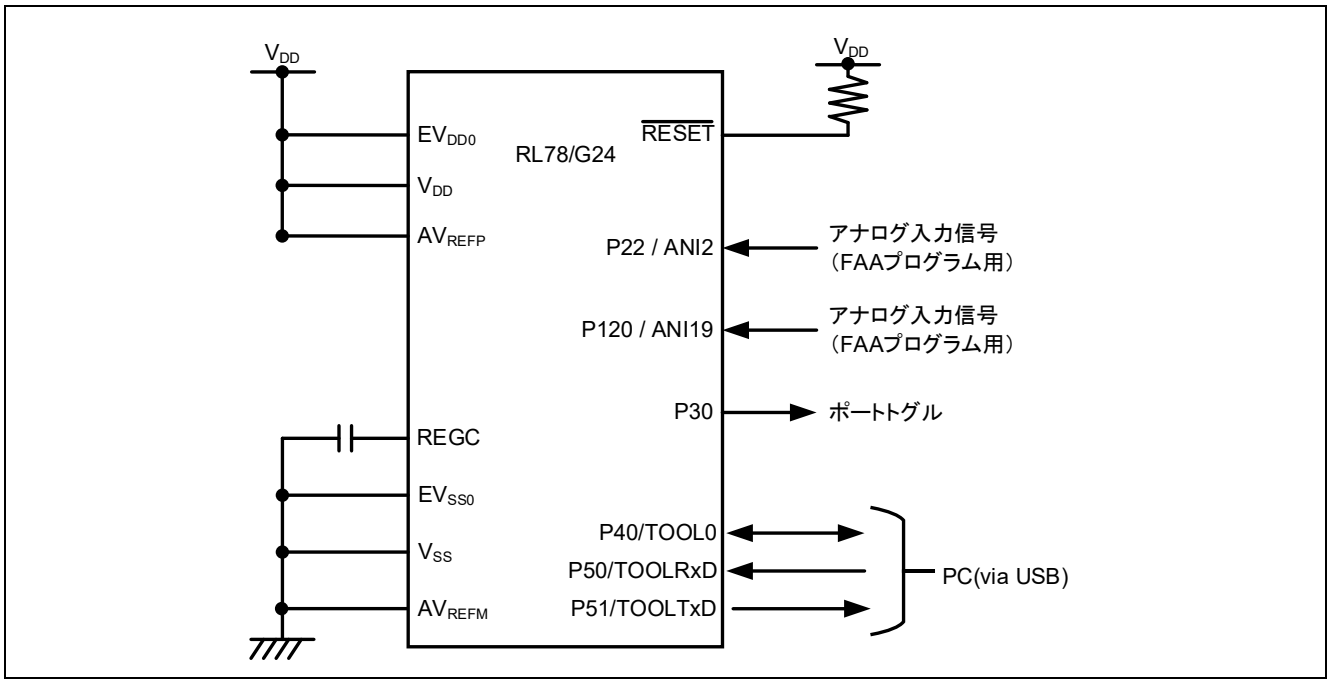

図 10-1 ハードウェア構成例

注意 1. この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい）。

注意 2. EV_{SS} で始まる名前の端子がある場合には V_{SS} に、EV_{DD} で始まる名前の端子がある場合には V_{DD} にそれぞれ接続してください。

注意 3. V_{DD} は LVD0 にて設定したリセット解除電圧（V_{LVD0}）以上にしてください。

10.2 使用端子一覧

表 10-1 に使用端子と機能を示します。

表 10-1 使用端子と機能

端子名	入出力	機能
P22 / ANI2	入力	A/D コンバータ アナログ入力（FAA プログラム用）
P120 / ANI19	入力	A/D コンバータ アナログ入力（FAA プログラム用）
P30	出力	ポートトグル

注意 本サンプルアプリケーションは、ノイズ対策のために未使用端子処理は”出力”に設定しています。入力専用端子の場合は該当ビットの設定、または、個別に抵抗を介して V_{DD} または V_{SS} に接続する必要があります。詳細は「RL78/G24 ユーザーズマニュアル　ハードウェア編（R01UH0961）」の「2.3 未使用端子の処理」を参照し、実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

11. ソフトウェア説明

11.1 スマート・コンフィグレータの設定

本サンプルコードにおけるスマート・コンフィグレータの設定を示します。スマート・コンフィグレータの設定における各表の項目、設定内容は設定画面の表記で記載しています。

11.1.1 システム設定

図 11-1 に本サンプルコード（e² studio、CS+）で使用しているシステム設定を示します。

RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ) で COM port デバッグを行う場合、統合開発環境（e² studio、CS+）内の設定を適切に行う必要があります。詳細は、「RL78/G24 Fast Prototyping Board ユーザーズマニュアル（R20UT5091）」の「7.1 e² studio で COM port デバッグを使用する場合」と「7.2 CS+を COM port デバッグ使用する場合」を参照してください。

図 11-1 システム設定（e² studio、CS+）

11.1.2 コンポーネントの設定

本サンプルコードで使用しているコンポーネントの設定を以下に示します。

表 11-1 コンポーネントの設定（FAA）

項目	内容
コンポーネント	フレキシブル・アプリケーション・アクセラレータ
コンフィグレーション名	Config_FAA
リソース	FAA

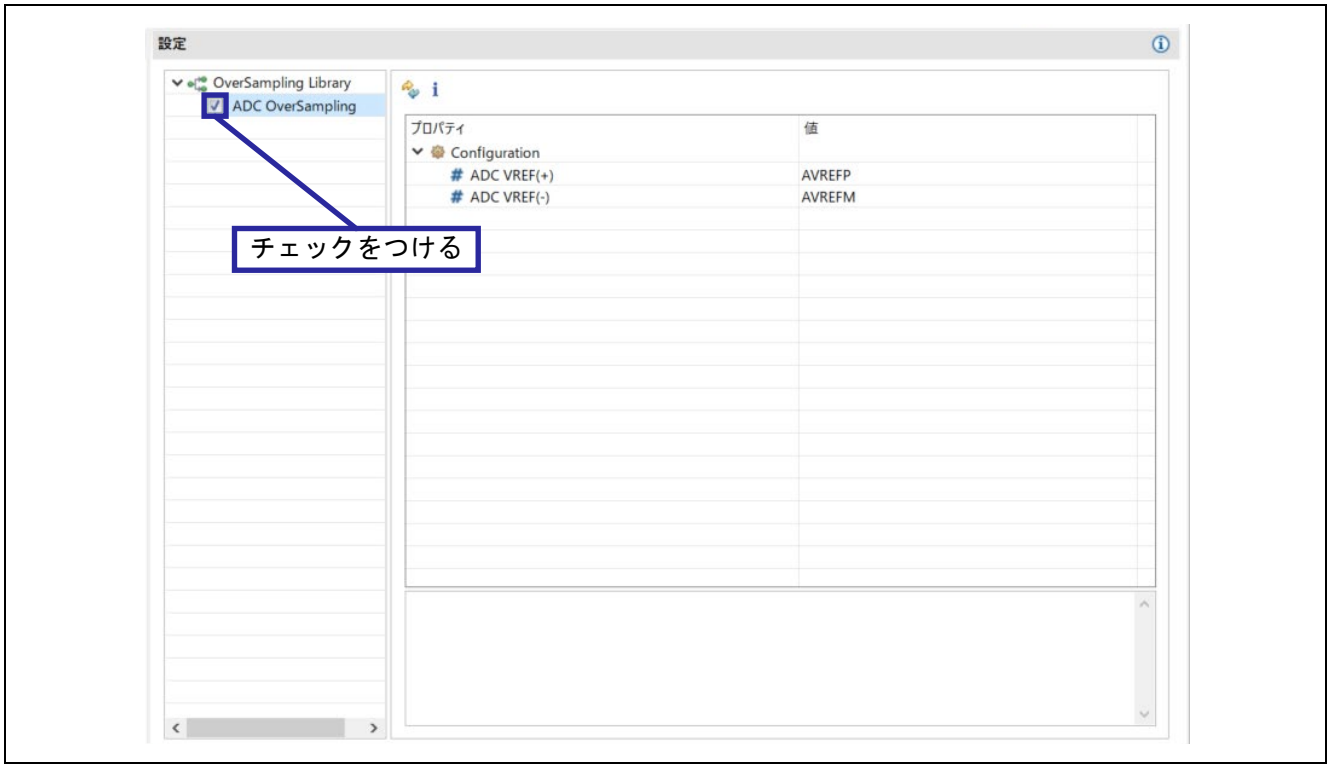

図 11-2 FAA の設定

注意　サンプルコードを読み込み後に FAA ライブラリが表示されない場合、「フレキシブル・アプリケーション・アクセラレータ(FAA)ツールガイド CS＋編（e² studio 編）」の「2.3.1 FAA コンポーネントの追加」の No.11 を参照して、FAA ライブラリをダウンロードしてください。

表 11-2 コンポーネントの設定（A/D コンバータ）

項目	内容
コンポーネント	A/D コンバータ
コンフィグレーション名	Config_ADC
リソース	ADC
動作モード	アドバンスド・モード

図 11-3 A/D コンバータの設定（1/2）

図 11-4 A/D コンバータの設定（2/2）

表 11-3 コンポーネントの設定（ポート）

項目	内容
コンポーネント	ポート
コンフィグレーション名	Config_PORT
リソース	PORT

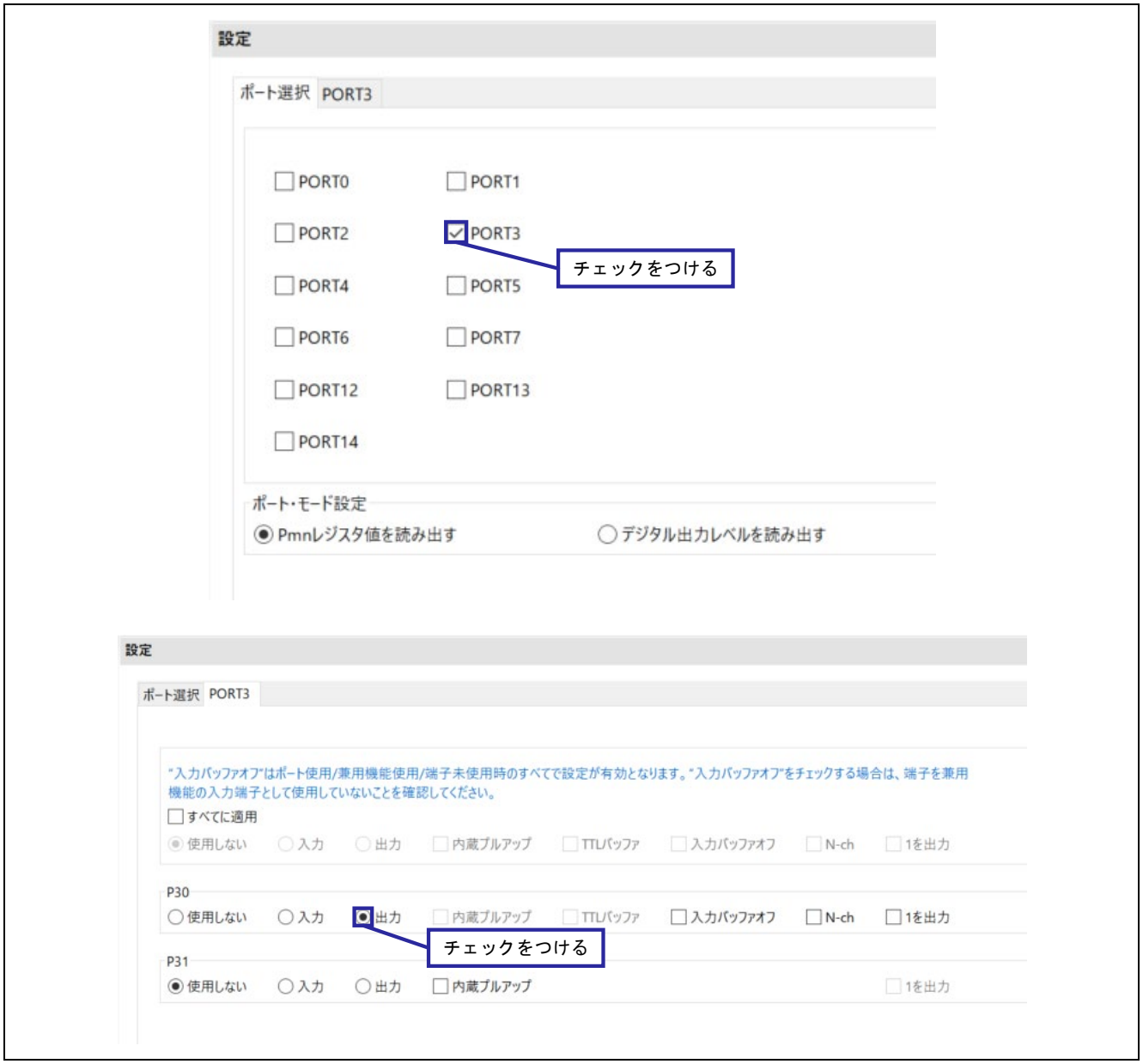

図 11-5 ポートの設定

注意　本サンプルプログラムでは、ノイズによる A/D 変換への影響を低減することを目的として、未使用端子を出力に固定しています。なお、入力専用端子については、内蔵プルアップ抵抗を有効にする、または抵抗を介して個別に VDD もしくは VSS に接続するなど、適切な未使用端子処理を行う必要があります。詳細は「RL78/G24 ユーザーズマニュアル　ハードウェア編（R01UH0961）」の「2.3 未使用端子の処理」を参照してください。本サンプルプログラムでは、P137 のみ入力専用端子の未使用端子処理を行う必要があります。

11.2 フォルダ構成

表 11-4 にサンプルコードの使用しているソースファイル／ヘッダファイルの構成を示します。なお、統合開発環境で自動生成されるファイル、bsp 環境のファイルは除きます。

表 11-4 フォルダ構成

フォルダ、ファイル名	説明	スマート・コンフィグレータを使用
\r01an7576_faa_adc_oversampling<DIR>	サンプルコードのフォルダ	
\src<DIR>	プログラム格納用フォルダ	
main.c	サンプルコードソースファイル	
main.h	サンプルコードヘッダファイル	
faa_oversampling.c	FAA 用ソースファイル	
faa_oversampling.h	FAA 用ヘッダファイル	
\smc_gen<DIR>	スマート・コンフィグレータ生成フォルダ	√
\Config_ADC<DIR>	ADC 用プログラム格納フォルダ	√
Config_ADC.c	ADC 用ソースファイル	√
Config_ADC.h	ADC 用ヘッダファイル	√
Config_ADC_user.c	ADC 用割り込みソースファイル	√注 1
\Config_FAA<DIR>	FAA 用プログラム格納フォルダ	√
Config_FAA_ADC_OverSampling.c	オーバーサンプリング機能のソースファイル	√注 2
Config_FAA_ADC_OverSampling.h	オーバーサンプリング機能のヘッダファイル	√
Config_FAA_common.c	Common FAA module のソースファイル	√
Config_FAA_common.h	Common FAA module のヘッダファイル	√
Config_FAA_common.inc	FAA 用インクルードファイル	√
Config_FAA_src.dsp	FAA 用アセンブラ・ソースファイル	√
\Config_PORT<DIR>	PORT 用プログラム格納フォルダ	√
Config_PORT.c	PORT 用ソースファイル	√
Config_PORT.h	PORT 用ヘッダファイル	√
Config_PORT_user.c	PORT 用割り込みソースファイル	√注 1
¥general<DIR>	初期化、共通プログラム格納フォルダ	
¥r_bsp<DIR>	BSP 用プログラム格納フォルダ	
¥r_config<DIR>	プログラム格納フォルダ	

備考 ”<DIR>”は、ディレクトリを意味します。
注1. 本サンプルコードでは使用しません。
注2. スマート・コンフィグレータのユーザコードエリアに、コードを追加しています。

11.3 オプション・バイトの設定一覧

表 11-5 にオプション・バイト設定を示します。

表 11-5 オプション・バイト設定

アドレス	設定値	内容
000C0H/040C0H	1110 1111B (EFH)	ウォッチドッグ・タイマ動作停止 （リセット解除後、カウント停止）
000C1H/040C1H	1111 1011B (FBH)	LVD0 リセット・モード 検出電圧：立ち上がり 2.97V／立下り 2.91V
000C2H/040C2H	1110 1010B (EAH)	フラッシュ動作モード：高速メインモード 高速オンチップ・オシレータの周波数：8MHz
000C3H/040C3H	1000 0101B (85H)	オンチップ・デバッグ動作許可

11.4 定数一覧

本サンプルコードでは定数は使用しません。

11.5 変数一覧

本サンプルコードでは変数は使用しません。

11.6 関数一覧

表 11-6 にサンプルコードで使用する関数を示します。ただし、スマート・コンフィグレータで生成された関数の内、変更を行っていないものは除きます。

表 11-6 関数一覧

関数名	概要	ソースファイル
main	メイン処理	main.c
oversampling_callback	コールバック処理	main.c

11.7 関数仕様

サンプルコードの関数仕様を示します。

[関数名] main

概要	メイン処理
ヘッダ	r_smc_entry.h、main.h、Config_FAA_ADC_OverSampling.h
宣言	void main (void);
説明	A/D コンバータの初期設定を行い、A/D 変換時間とオーバーサンプリング対象チャネルを設定後、FAA の動作を開始させます。オーバーサンプリング終了をポーリングし、SHDMEM から結果を取得します。A/D 変換時間とオーバーサンプリング対象チャネルの切り換えと FAA 動作開始を繰り返します。
引数	なし
リターン値	なし
備考	なし

[関数名] oversampling_callback

概要	コールバック処理
ヘッダ	Config_FAA_common.h、main.h、Config_FAA_ADC_OverSampling.h
宣言	void oversampling_callback(void)
説明	ユーザーコードを書き込んでください。サンプルコードでは NOP 命令を実行しています。
引数	なし
リターン値	なし
備考	なし

11.8 フローチャート

11.8.1 メイン処理

図 11-6 にメイン処理のフローチャートを示します。

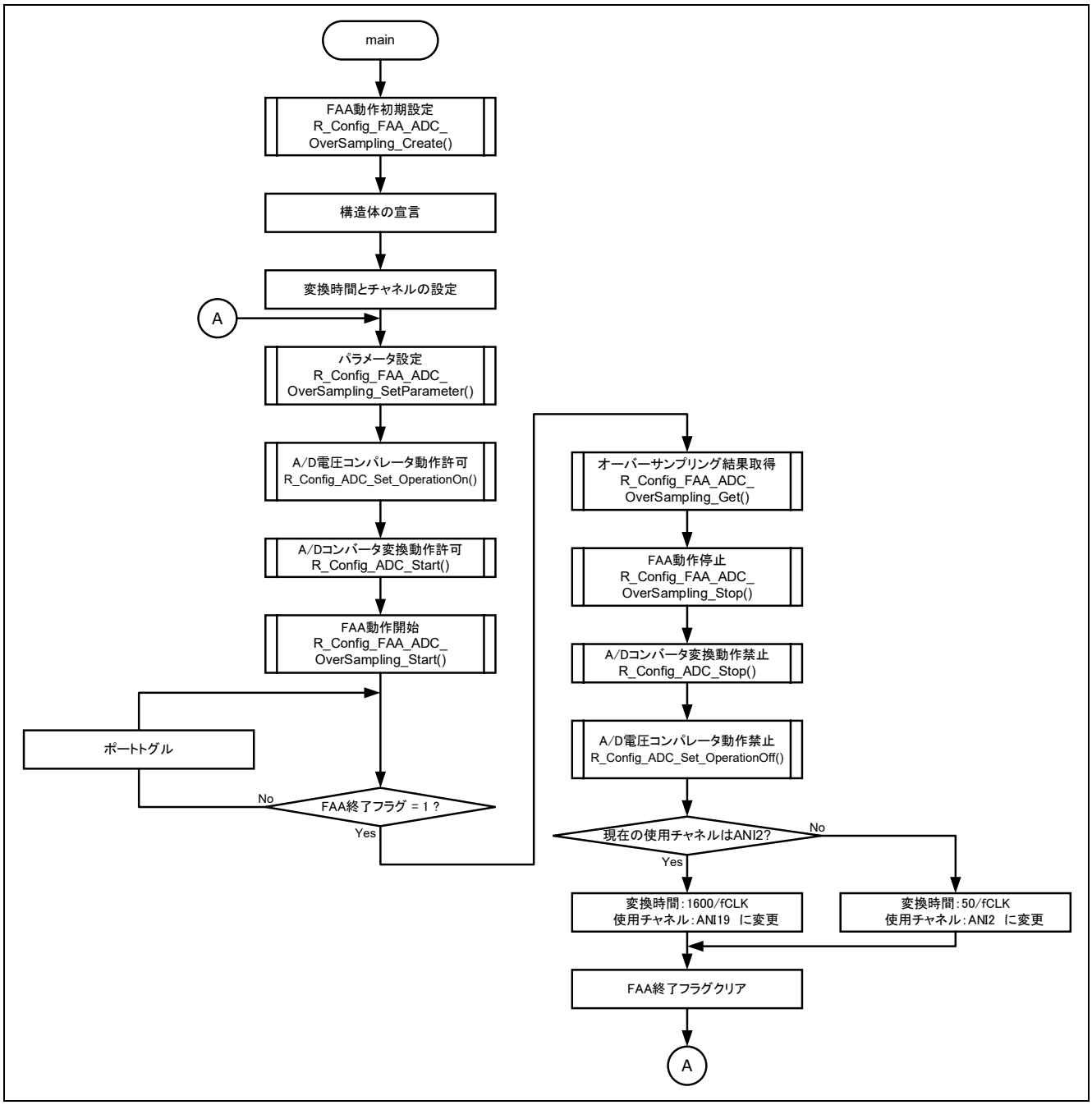

図 11-6 メイン処理

11.8.2 oversampling_callback 関数

図 11-7 に oversampling_callback のフローチャートを示します。

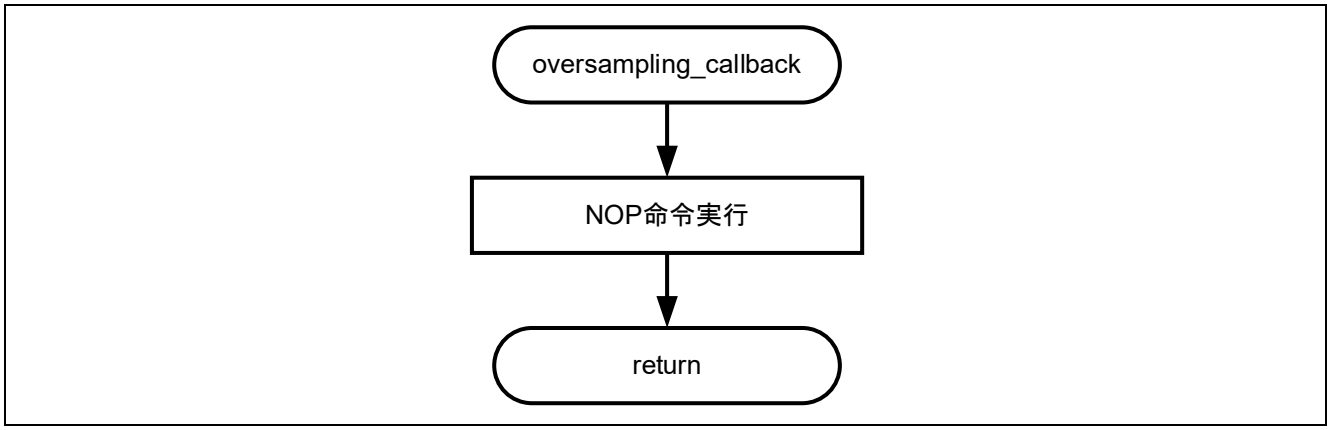

図 11-7 oversampling_callback 関数

12. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

13. 参考ドキュメント

RL78/G24 ユーザーズマニュアル　ハードウェア編（R01UH0961）

RL78 ファミリ　ユーザーズマニュアル　ソフトウェア編（R01US0015）

RL78/G24 Fast Prototyping Board ユーザーズマニュアル（R20UT5091）

RL78 スマート・コンフィグレータ　ユーザーガイド：e2 studio 編（R20AN0579）

RL78 スマート・コンフィグレータ　ユーザーガイド：CS+編（R20AN0580）

フレキシブル・アプリケーション・アクセラレータ(FAA)ツールガイド e2 studio 編（R01AN7095）

フレキシブル・アプリケーション・アクセラレータ(FAA)ツールガイド CS＋編（R01AN7094）

（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

（最新版の情報をルネサス エレクトロニクスホームページから入手してください。）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jan. 27th, 2026	ー	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. 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て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

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/

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。