Renesas Synergy™ プラットフォーム

ADC 周期フレームワークモジュールガイド

R11AN0115JU0102
Rev.1.02
2019.06.21

(注1)本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

(注 2)本資料の第 6 章まで(要旨除く)の日本語訳は、「Synergy™ Software Package (SSP) v1.5.0 ユーザーズマニュアル モジュール概要編(参考資料)」の第 4 章「モジュールの概要」に掲載されていますのでそちらを参照ください 。

要旨(Introduction)

本モジュールガイドは、ユーザがモジュールを効果的に使用してシステムが開発できるようになることを目的としています。このモジュールガイドを習得することで、開発システムへのモジュールの追加とターゲットアプリケーション向けの正確な設定(configuration)ができ、さらに付属のアプリケーションプロジェクトコードを参照して、効率的なコード記述が行えるようになります。

より詳細な API や、より高度なモジュール使用法を記述した他のアプリケーションプロジェクト例もルネサス WEB サイト(本書末尾の「参考文献」の項を参照)から入手でき、より複雑な設計に役立ちます。

ADC 周期フレームワーク(ADC Periodic Framework)は、信号処理アプリケーション向けのハイレベル API(high-level API)です。 このモジュールは、指定したレートでチャネルをサンプリングするように ADC/SDADC を設定し(シングルスキャンモードを使用)、アプリケーションへの通知を行う前に設定したサンプリング回数のデータをバッファします。ADC 周期フレームワークは、Renesas Synergy™マイクロコントローラの ADC/SDADC、GPT、AGT および DTC の各周辺回路を使用します。新しいサンプリング(new sample)が完了するたびにデータを処理する目的で、ユーザ定義のコールバック関数を作成することもできます。

目次 (Contents)

1. ADC 周期フレームワークモジュールの特徴(ADC Periodic Framework Module Features) 3

2. ADC 周期フレームワークモジュール API の概要(ADC Periodic Framework Module APIs Overview) 3

3. ADC 周期フレームワークモジュール動作の概要(ADC Periodic Framework Module Operational Overview) 3

4. アプリケーションへの ADC 周期フレームワークモジュールの組み込み(Including the ADC Periodic Framework Module in an Application) 3

5. ADC 周期フレームワークモジュールの構成(Configuring the ADC Periodic Framework Module)3

6. アプリケーションでの ADC 周期フレームワークモジュールの使用(Using the ADC Periodic Framework Module in an Application) 3

7. ADC フレームワークモジュールのアプリケーションプロジェクト(The ADC Periodic Framework Module Application Project) 3

8. ターゲットアプリケーションに対応する ADC 周期フレームワークモジュールのカスタマイズ

（Customizing the ADC Periodic Framework Module for a Target Application）.....7

9. ADC 周期フレームワークモジュールのアプリケーションプロジェクトの実行（Running the ADC Periodic Framework Module Application Project）.....7

10. ADC 周期フレームワークモジュールのまとめ（ADC Periodic Framework Module Conclusion）...9

11. ADC 周期フレームワークモジュールの次の手順（ADC Periodic Framework Module Next Steps）.....10

12. ADC 周期フレームワークモジュールの参考情報（ADC Periodic Framework Module Reference Information）.....10

1. **ADC 周期フレームワークモジュールの特徴（ADC Periodic Framework Module Features）**
2. **ADC 周期フレームワークモジュール API の概要（ADC Periodic Framework Module APIs Overview）**
3. **ADC 周期フレームワークモジュール動作の概要（ADC Periodic Framework Module Operational Overview）**
4. **アプリケーションへの ADC 周期フレームワークモジュールの組み込み（Including the ADC Periodic Framework Module in an Application）**
5. **ADC 周期フレームワークモジュールの構成（Configuring the ADC Periodic Framework Module）**
6. **アプリケーションでの ADC 周期フレームワークモジュールの使用（Using the ADC Periodic Framework Module in an Application）**
7. **ADC フレームワークモジュールのアプリケーションプロジェクト（The ADC Periodic Framework Module Application Project）**

このモジュールガイドに関連するアプリケーションプロジェクトは、設計全体の手順を示しています。このプロジェクトは、このドキュメントの末尾にある「参考情報」の章のリンク先から参照できます。ISDE でアプリケーションプロジェクトをインポートして開き、ADC 周期フレームワークモジュールに対応する設定項目を表示することができます。

アプリケーションプロジェクト全体は、このドキュメントの末尾にある「参考情報」の章のリンク先から参照できます。adc_example_entry.c ファイルは、このプロジェクトを ISDE にインポートした後、プロジェクト内に配置されます。ISDE でこのファイルを開き、API の主な使い方を確認するのに役立つ説明を参照することができます。

このアプリケーションプロジェクトは、ADC 周期フレームワークモジュール API の一般的な使用方法を示しています。このアプリケーションプロジェクトのメインスレッドのエントリは、ADC 周期フレームワークを初期化し、周期的に温度センサをスキャンします。スキャン結果は、ユーザ指定のバッファ内に格納されます。スキャン結果が存在した時点で、ユーザコールバック関数を呼び出します。ユーザ指定のコールバック関数は、共通のセミホスト機能（semi-hosting function）を使用して、結果をデバッグコンソール（debug console）に出力します。

表 11　このアプリケーションプロジェクトが使用するソフトウェアとハードウェアのリソース

リソース	リビジョン	説明
e² studio	5.3.1 またはそれ以降	統合ソリューション開発環境 (ISDE)
SSP	1.2.0 またはそれ以降	Synergy ソフトウェアプラットフォーム
IAR EW for Synergy	7.71.2 またはそれ以降	IAR Embedded Workbench® for Renesas Synergy™
SSC	5.3.1 またはそれ以降	Synergy Standalone Configurator
SK-S7G2	v3.0, v3.1 またはそれ以降	スタータキット

次の図に、このアプリケーションプロジェクトの簡単なフローを示します。

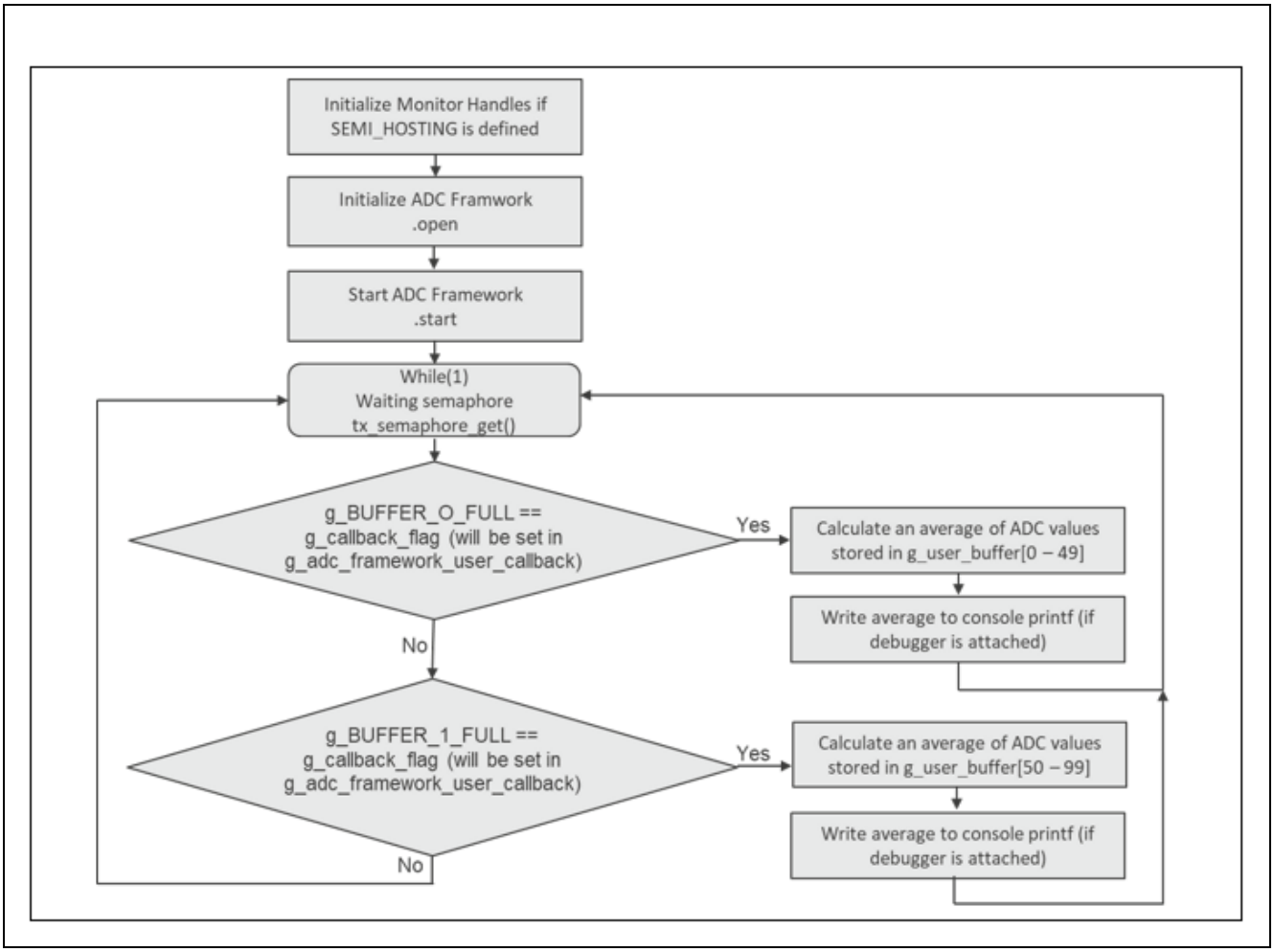

図 1　　ADC 周期フレームワークモジュールのアプリケーションプロジェクトのフロー図

ファイル adc_framework_mg_entry.c は、ISDE にインポートされるとプロジェクト内に配置されます。 ISDE 内でこのファイルを開き、記載されている説明に従って、ADC PERIODIC FRAMEWORK API の使用方法を確認します。

ユーザ定義のアプリケーションコードは、ADC インスタンス構造と外部関数宣言を参照するヘッダーファイルを含めることから始まります。これに続いて、アプリケーションに必要なグローバル変数が定義されます。これらは bsp_leds_t g_LED です。S7G2-SK ボードの LED および g_tsn_temperature の決定に使用され、デバイス温度の計算に使用されます。

次に、基本エラー処理関数（error_trap）が定義されています。このエラー処理関数は、ADC PERIODIC FRAMEWORK API のいずれかが正常に返されなかった場合に呼び出されます。アプリケーションの実行は、ソフトウェアブレーク命令__BKPT(1)を介して、このエラー処理関数内で停止します。 SEMI_HOSTING の使用が定義されている場合、エラーメッセージが仮想デバッグウィンドウ（e2 studio）/ターミナル I / O ウィンドウ（IAR-EW）に表示されます。

アプリケーション関数 void adc_framework_mg_entry（void）が定義されました。 SEMI_HOSTING と GCC ツールチェーンが定義されている場合、およびオンチップデバッガがアタッチされている場合は、関数 initialise_monitor_handles（）;と呼ばれます。

次に、ADC PERIODIC FRAMEWORK を初期化するための API が呼び出され、対応する制御ポインタと設定ポインタが渡されます。g_sf_adc_periodic0.p_api-> open（g_sf_adc_periodic0.p_ctrl、g_sf_adc_periodic0.p_cfg）; .open が正常に開かない場合は、error_trap 関数が呼び出されます。

次に、アプリケーション固有の 3 つの API が呼び出されます。これらの API は ADC PERIODIC FRAMEWORK の動作には必要ありませんが、ユーザがアプリケーションと対話できるようにします。 外部 IRQ を開くための 2 つの API と開発ボードの LED を識別するための API が呼び出されます。これらの API が正常に返されない場合は、error_trap 関数が呼び出されます。

次に、ADC PERIODIC FRAMEWORK を起動するための API が呼び出され、対応する制御ポインタが渡されます。g_sf_adc_periodic0.p_api-> start（g_sf_adc_periodic0.p_ctrl）; .open が正常に開かない場合は、error_trap 関数が呼び出されます。

アプリケーションコードは while（1）処理ループに入ります。 アプリケーションはイベントフラグを無期限に待機します 。tx_event_flags_get （ & g_buffer_button_event_flags 、ALL_EVENT_FLAGS 、TX_OR_CLEAR 、 & actual_flags、TX_WAIT_FOREVER）;

イベントフラグ（ファイル#include "adc_framework_mg_entry.h"で定義されている）は、ADC バッファフレームワークコールバック関数（データバッファに有効なデータがあることを示す）または外部 IRQ コールバック関数（プッシュボタンが押されたことを示す）で設定されます。

どのステートメントフラグが設定されているかを判断するには、switch ステートメントを使用します。

ユーザープッシュボタン S4 が押された場合、イベントフラグ EVENT_FLAG_0x1_S4_BUTTON が設定され、ADC 周期枠を停止するための API が呼び出されます。g_sf_adc_periodic0.p_api-> stop（g_sf_adc_periodic0.p_ctrl）;

ユーザープッシュボタン S5 が押された場合は、イベントフラグ EVENT_FLAG_0x2_S5_BUTTON が設定され、ADC PERIODIC FRAMEWORK を開始するための API が呼び出されます。g_sf_adc_periodic0.p_api-> start（g_sf_adc_periodic0.p_ctrl）;

ADC PERIODIC FRAMEWORK バッファに有効なデータがある場合、コールバック関数は処理するバッファを示すイベントフラグを設定します。 コールバックは EVENT_FLAG_0x4_BUFFER_0_49_FULL または EVENT_FLAG_0x8_BUFFER_50_99_FULL を設定します。このデータからデバイスの温度が計算され、

SEMI_HOSTING が有効になっていると、温度の結果が仮想デバッグウィンドウ（e2 Studio）/ターミナルウィンドウ（IAR-EW）に表示されます。各組のバッファデータに基づく温度値が示されている。プッシュボタンを押すと、状態を示すメッセージが表示されます。F（華氏）の C（摂氏）の温度表示フォーマットは adc_framework_mg_entry.h の定義によって設定できます。

注:　この説明は、Synergy ソフトウェアパッケージ内のデバッグコンソールで printf() を使用する方法をユーザが理解していることを想定しています。このような経験がない場合は、下記 WEB サイトの FAQ 2000008「Synergy ソフトウェアパッケージのデバッグコンソールで Printf_使用方法」という記事を参照してください。デバッグモードで変数ウォッチ機能を使用して結果を表示することもできます。
https://ja-support.renesas.com/knowledgeBase/17792531

ターゲットボードや MCU の必須の操作と物理プロパティ(physical properties)をサポートするために、このアプリケーションプロジェクトではいくつかの重要なプロパティを設定しています。以下の表に、それらのプロパティと、このプロジェクトにおける設定値を示します。実践的な演習(hands-on exercise)として、このアプリケーションプロジェクトを開き、[Properties] ウィンドウでこれらの設定を表示することもできます。

表 12　アプリケーションプロジェクトに対応する ADC 周期フレームワークモジュールの設定項目

ISDE のプロパティ	設定値
Name	g_sf_adc_periodic0
Name of the data buffer to store samples	g_user_buffer
Length of the buffer	100
Number of Sampling iterations	50
Callback	g_adc_framework_user_callback
Name of generated initialization function	sf_adc_periodic_init0
Auto initialization	Disable

表 13　アプリケーションプロジェクトに対応する g_adc0 ADC on r_adc の設定項目

ISDE のプロパティ	設定値
Name	g_adc0
Unit	0
Resolution	12-bit
Alignment	Right
Clear after read	On
Internal Calibration During Open()	Disabled
Channel Scan Mask >Temperature Sensor	Used in Normal/ Group A
Scan End Interrupt Priority	Priority 8

表 14.　g_timer0 Timer Driver on r_gpt

ISDE のプロパティ	設定値
Name	g_timer0
Channel	0
Period Value	20
Period Unit	Milliseconds

表 15.　g_transfer0 Transfer Driver on r_dtc ADC COMPARE MATCH

ISDE のプロパティ	設定値
Name	g_timer0

表 16.　g_external_irq10 External IRQ Driver on r_icu

ISDE のプロパティ	設定値
Name	g_external_irq10
Channel	10
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock	PCLK / 64
Interrupt enabled after initialization	True
Callback	cb_external_irq10
Pin Interrupt Priority	Priority 10

表 17.　Pin Configuration Settings for g_external_irq10 External IRQ Driver on r_icu

ISDE のプロパティ	設定値
Ports ⟶ P0 ⟶ P005	
Mode	Input Mode
IRQ	IRQ10-DS

表 18.　g_external_irq11 External IRQ Driver on r_icu

ISDE のプロパティ	設定値
Name	g_external_irq11
Channel	11
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock	PCLK / 64
Interrupt enabled after initialization	True
Callback	cb_external_irq10
Pin Interrupt Priority	Priority 12

表 19.　Pin Configuration Settings for g_external_irq11 External IRQ Driver on r_icu

ISDE のプロパティ	設定値
Ports ⟶ P0 ⟶ P006	
Mode	Input Mode
IRQ	IRQ11-DS

8.　ターゲットアプリケーションに対応する ADC 周期フレームワークモジュールのカスタマイズ（Customizing the ADC Periodic Framework Module for a Target Application）

いくつかの設定項目は通常、アプリケーションプロジェクトが示している値に対し、ユーザが変更を加えます。たとえば、ユーザは **[Clocks]** タブで [PCLKC] を更新する方法により、ADC クロックに関する設定項目を変更することができます。また、開発者は希望のアナログ入力を選択するために、ADC ポートの端子を変更することもできます。この変更を行うには、コンフィギュレータの **[Pins]** タブを使用します。この ADC アプリケーションプロジェクトは、オンチップの温度センサ（on-chip temperature sensor）を使用し、ダイ（チップ）温度（die temperature）を読み取ります。

9.　ADC 周期フレームワークモジュールのアプリケーションプロジェクトの実行（Running the ADC Periodic Framework Module Application Project）

ADC 周期フレームワークのアプリケーションプロジェクトをターゲットキットでその動作を確認するために、ISDE にこのプロジェクトをインポートし、コンパイルしてデバッグを実行することができます。

新しいプロジェクト内で ADC 周期フレームワークアプリケーションを実装するには、ターゲットキットで定義、設定、ファイルの自動生成、コードの追加、コンパイル、デバッグを行うための手順に従います。実際にこれらの手順に従うことで SSP での開発プロセスをより実践的に習得するのに役立ちます。

注:　Synergy 開発プロセスの基本的な流れを経験したことのあるユーザにとって、以下の手順は十分詳細なものです。これらの手順をまだ理解していない場合、これらの手順を実行する説明について、このドキュメントの末尾にある「参考情報」の章に掲載されているリンク先から『SSP ユーザーズマニュアル』の最初にあるいくつかの章を参照してください。

ADC 周期フレームワークのアプリケーションプロジェクトを作成し、実行するために、以下の手順に従ってください。

1. e² studio または IAR Embedded Workbench® for Renesas Synergy™ にプロジェクトをインポートし、アプリケーションをビルドして実行する手順については、『Synergy プロジェクトインポートガイド』（下記 WEB）を参照してください。
   英語版：
   https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0023eu0121-synergy-ssp-import-guide.pdf
   日本語版（参考資料）：
   https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0023ju0121-synergy-ssp-import-guide.pdf
   注：GCC コンパイラを使用する場合、このアプリケーション例は printf を介して仮想デバッグウィンドウに浮動小数点を送信するため、図 7 に示すように、リンカ設定「Use float with nano printf（-u _printf_float）」が選択されていることを確認してください。
2. micro USB ケーブルで、SK-S7G2 キットの J19 とホスト PC を接続します。
3. アプリケーションのデバッグを開始します。

出力は、デバッグコンソール（debug console）に表示されます。チップを親指で軽く押さえるか、チップ全体に息を吹きかける方法で、温度の読み取り値の変動を観察することができます。

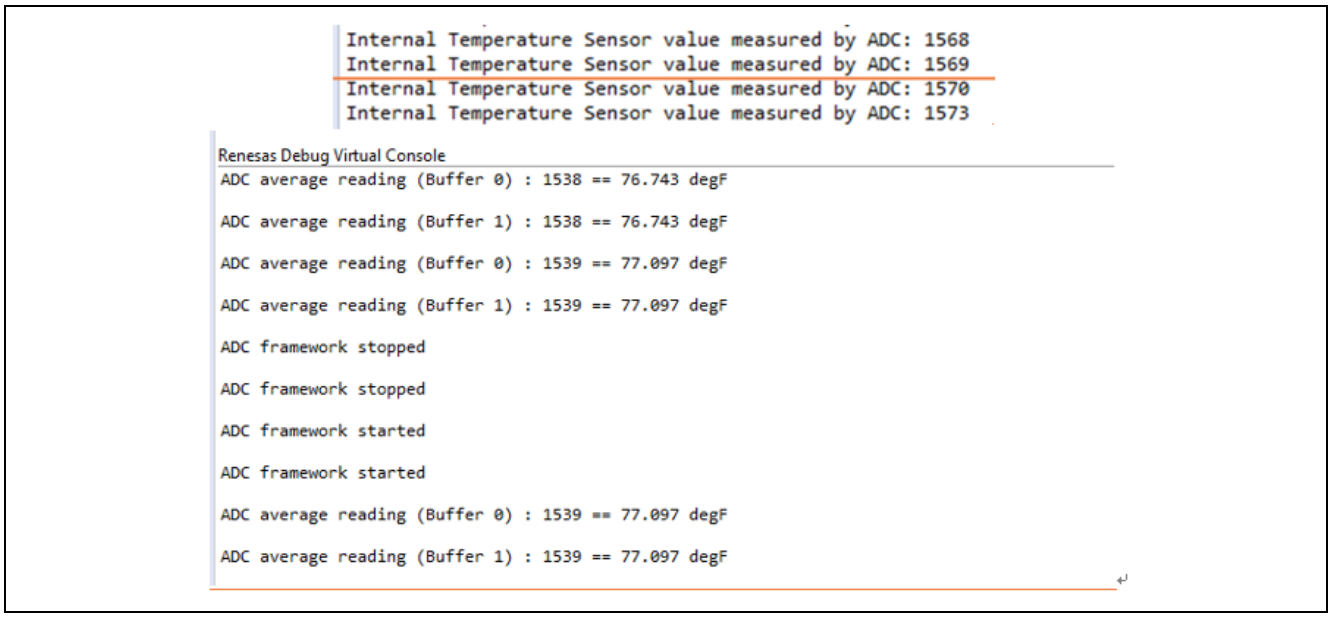

図 6　ADC 周期フレームワークのアプリケーションプロジェクトのサンプル出力（E2Studio, 華氏温度表示選択）

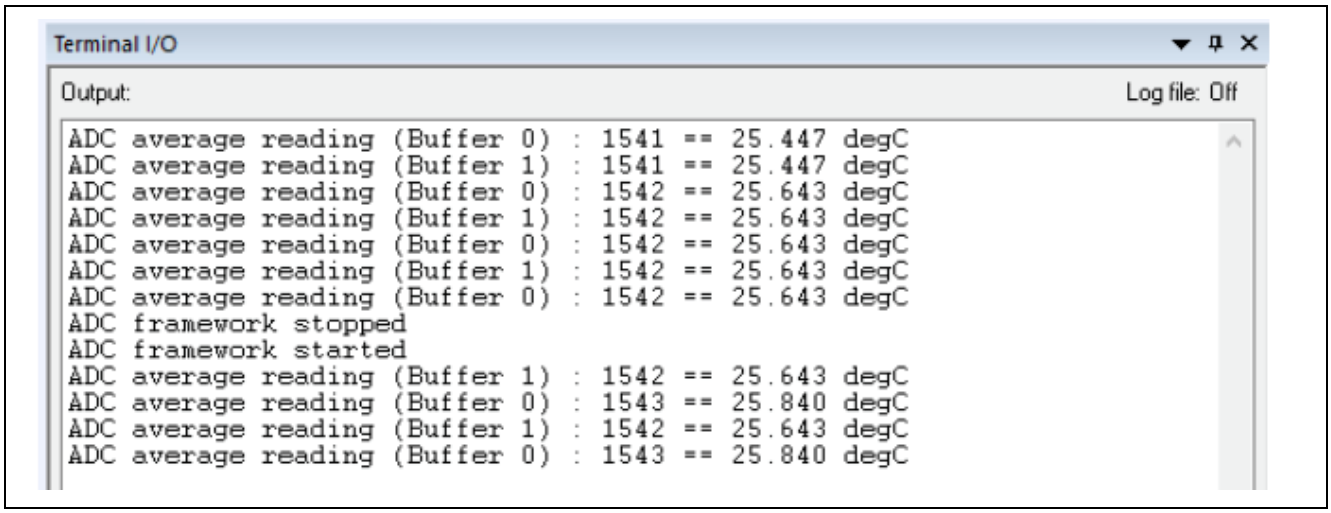

図 7　ADC 周期フレームワークのアプリケーションプロジェクトのサンプル出力（IAR-EW, 摂氏温度表示選択）

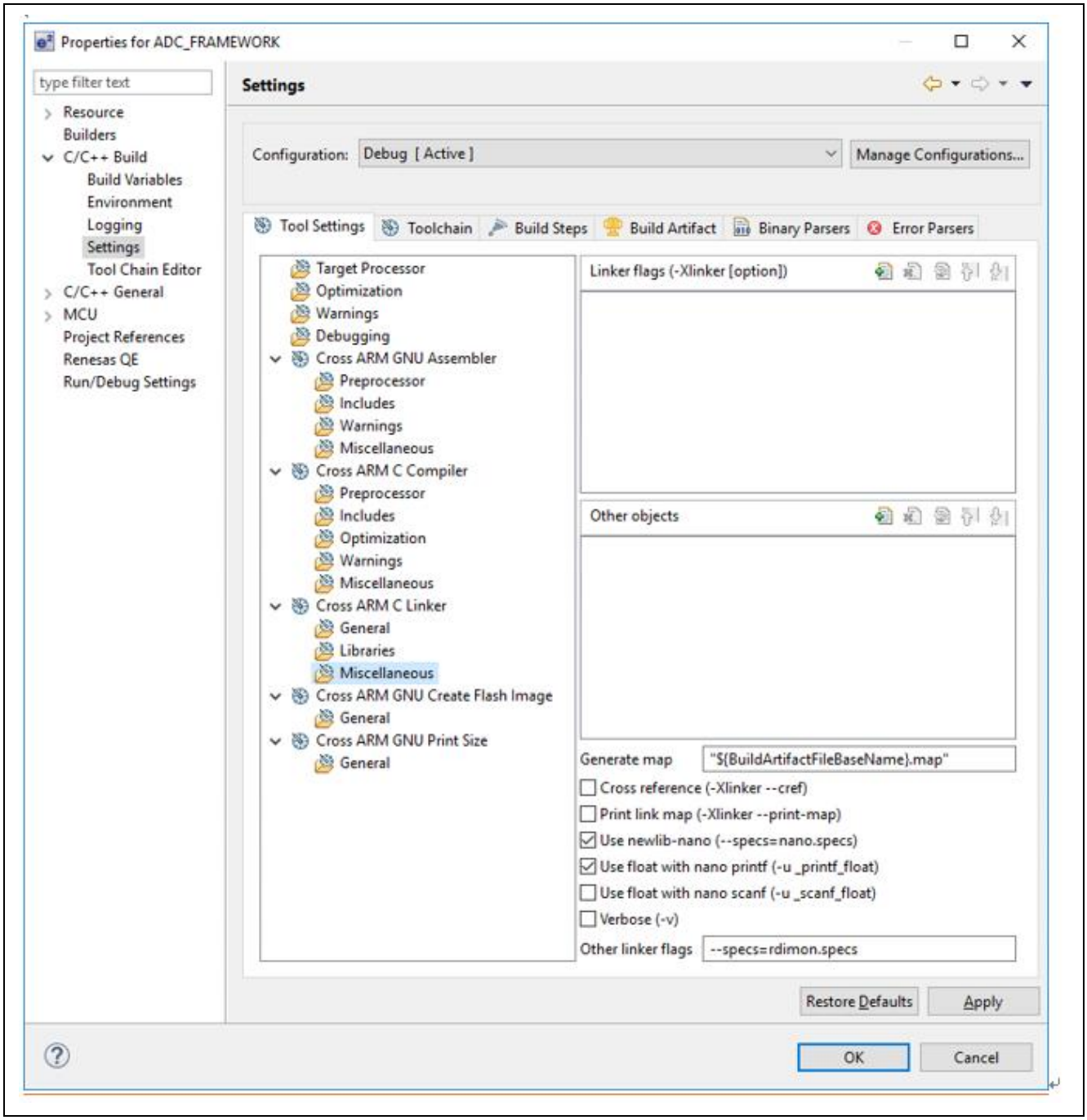

図 8　　float with nano printf の有効化

10. ADC 周期フレームワークモジュールのまとめ（ADC Periodic Framework Module Conclusion）

このモジュールガイドは、サンプルプロジェクトでモジュールの選択、追加、設定、使用を行うために必要な背景となる情報全般を説明しました。従来の組み込みシステムでは、これらの手順を理解することに多くに時間を必要とし、また間違いが起こりやすい操作でした。Renesas Synergy プラットフォームにより、これら手順の所要時間が短くなり、設定項目の競合や、ローレベルドライバの誤った選択など、誤りが防止できるようになりました。アプリケーションプロジェクトで示したように、ハイレベル API を使用することで高いレベルの開発からスタートし、ローレベルドライバを作成するような従来の開発環境で必要とされる時間が不要になり、開発時間を短縮できます。

11. ADC 周期フレームワークモジュールの次の手順（ADC Periodic Framework Module Next Steps）

ADC 周期フレームワークモジュールのプロジェクトを 1 つマスターした後、他のサンプルを確認することができます。ターゲットアプリケーションによっては、ADC HAL モジュール（ADC HAL module）の方が適していることがあります。『ADC HAL モジュールガイド』は、他の実装で ADC を使用する方法を示します。ADC の使用方法を示す他のアプリケーションプロジェクトとアプリケーションノートは、このドキュメントの末尾にある「参考情報」の章から参照できます。

12. ADC 周期フレームワークモジュールの参考情報（ADC Periodic Framework Module Reference Information）

『SSP ユーザーズマニュアル』：SSP ディストリビューションパッケージの一部として HTML 形式が入手できるほか、Renesas Synergy™ WEBサイトのSSPページ
https://www.renesas.com/jp/ja/products/synergy/software/ssp.htmlから pdf を入手することもできます。

最新版のsf_adc_periodic モジュールの参考資料やリソースへのリンクは、以下の Synergy WEBサイトから入手できます。

https://www.renesas.com/jp/ja/products/synergy.html

ホームページとサポート窓口

テクニカルサポート:　https://synergygallery.renesas.com/support

技術的なお問い合わせ先:

- アメリカ:　https://renesas.zendesk.com/anonymous_requests/new
- ヨーロッパ:　https://www.renesas.com/en-eu/support/contact.html
- 日本:　https://www.renesas.com/ja-jp/support/contact.html

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.02	2019.06.21		・初版 ・英文版（R11AN0115EU0102、Rev.1.02、2019.Mar.03）の巻頭と第 7 章以降を翻訳

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、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。
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、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。
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、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。
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。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。
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.4.0-1 2017.11)

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/

© 2018 Renesas Electronics Corporation. All rights reserved.

Colophon 6.0