

# Smart Analog IC 300

R02AN0020JJ0100

Rev.1.00

## 自己測定方法と補正方法 MCU ソフトウェア説明

2014.01.31

### 要旨

本アプリケーションノートでは、Smart Analog IC 300 のコンフィギュラブル・アンプが持つ誤差（ゲイン誤差、オフセット電圧）を、内蔵の D/A コンバータと制御用マイクロコントローラの A/D コンバータで測定し、補正するサンプルプログラムについて説明します。

### 動作確認デバイス

Smart Analog IC 300 （品名：RAA730300）、RL78/G1A （品名：R5F10ELE）

本アプリケーションノートを、他のマイクロコントローラへ適用する場合、そのマイクロコントローラの仕様にあわせて変更し、十分評価してください。

## 目次

1. 仕様 .....	3
2. 動作確認条件 .....	4
3. 関連アプリケーションノート .....	4
4. ハードウェア説明 .....	5
4.1 ハードウェア構成例 .....	5
4.2 使用機能一覧 .....	6
4.3 使用端子一覧 .....	6
5. 自己補正動作 .....	7
5.1 自己補正のタイミングチャート .....	7
5.2 ゲイン誤差測定 .....	8
5.2.1 ゲイン誤差測定手順 .....	8
5.2.2 ゲイン誤差測定時の Smart Analog IC 300 レジスタ設定値 .....	11
5.3 入力換算オフセット電圧測定 .....	12
5.3.1 入力換算オフセット電圧測定手順 .....	12
5.3.2 入力換算オフセット電圧測定時の Smart Analog IC 300 レジスタ設定値 .....	15
6. ソフトウェア説明 .....	16
6.1 関数一覧 .....	16
6.2 関数仕様 .....	18
6.3 グローバル変数一覧 .....	32
6.4 Smart Analog IC 300 のレジスタ値 .....	33
6.5 コード生成ツールの設定一覧 .....	35
6.6 フローチャート .....	38
7. サンプルコードの組み込み手順 .....	52

## 1. 仕様

関連アプリケーションノート（Smart Analog IC 300 ゲイン誤差・オフセット電圧の自己測定方法と補正方法（R02AN0019J）：以下、原理編と省略）にて、Smart Analog IC 300 のコンフィギュラブル・アンプが持つゲイン誤差およびオフセット電圧を測定する方法について示しました。

測定で得られたゲイン誤差とオフセット電圧は、制御用マイクロコントローラの RAM 内に格納され、コンフィギュラブル・アンプの出力電圧から入力電圧を逆算するのに用いられます。この算出された入力電圧は、ゲイン誤差とオフセット電圧がデジタル的に補正された値である、と言えます。

本アプリケーションノートでは、ゲイン誤差およびオフセット電圧を測定し、デジタル的に補正する一連の手法を自己補正と呼び、原理編で説明した、この自己補正手法を実現する MCU ソフトウェアの構成とサンプルコードについて示します。

本アプリケーションノートは、システム上で定期的に自己補正を行う方法について示し、実施例として、10 分ごとに自己補正を行います。図 1-1 に、自己補正のタイミングについて、状態遷移図を示します。

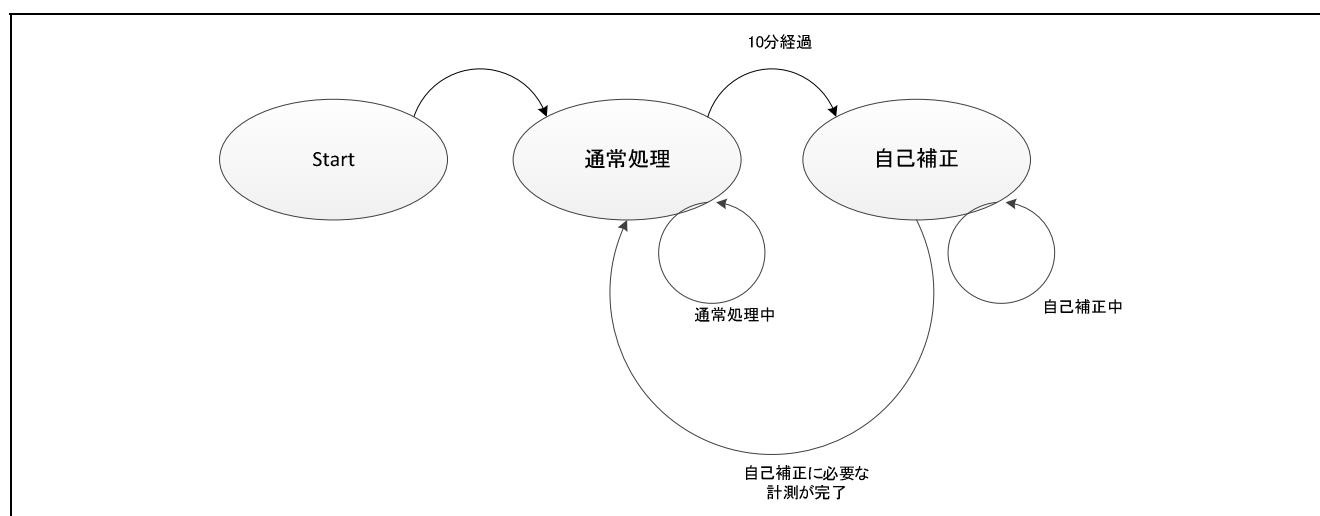


図 1-1 自己補正の状態遷移図

本アプリケーションノートでは、コンフィギュラブル・アンプ Ch1 を、40 dB ゲインの差動アンプ構成として用いることを仕様とします。差動アンプ構成時のゲイン誤差および入力換算オフセット電圧の測定に関する詳細は、原理編 3.2 節をご参照ください。（なお、他のコンフィギュラブル・アンプを、他のアンプ構成でご使用になる場合は、原理編をご参照の上、提供しますサンプルコードを基に、ソフトウェア作成を実施して下さい。）

自己補正の開始タイミングは、リアルタイム・クロックで生成・割り込みを 1 分ごとに発生させます。10 回発生すると自己補正を開始し、ゲイン誤差測定、入力換算オフセット電圧測定、の順で行います。また、自己補正が終了すると、通常の処理へ遷移します。本アプリケーションノートでは、この一連の流れを繰り返しています。

## 2. 動作確認条件

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。

表 2-1 動作確認条件

項目	内容
使用デバイス	RL78/G1A (R5F10ELE) Smart Analog IC 300 (RAA730300)
使用ボード	TSA-IC300 (テセラ・テクノロジー社製)
動作周波数	• 高速オンチップ・オシレータ・クロック : 32 MHz • CPU/周辺ハードウェア・クロック : 32 MHz
動作電圧	3.3 V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.00.00 [15 Mar 2013]
C コンパイラ	ルネサス エレクトロニクス製 CA78K0R V1.60
RL78/G1A コードライブラリ	ルネサス エレクトロニクス製 CodeGenerator for RL78/G1A V2.00.03

## 3. 関連アプリケーションノート

関連するアプリケーションノートを以下に示します。併せてご参照ください。

- RL78/G1E マルチセンサ接続時のアンプ構成切り替え (R01AN1129J) アプリケーションノート
- RL78/G1E アナログ部との SPI 通信サンプルコード (R01AN1130J) アプリケーションノート
- Smart Analog IC 300 センサタイプ別によるアンプ構成選択方法 (R02AN0016J) アプリケーションノート
- Smart Analog IC 300/301 機能特長と使用例 (Smart Analog IC 500 シリーズとの比較) (R02AN0018J) アプリケーションノート
- Smart Analog IC 300 ゲイン誤差・オフセット電圧の自己測定方法と補正方法 (R02AN0019J) アプリケーションノート
- RL78/G13 リアルタイム・クロック (R01AN0454J) アプリケーションノート

## 4. ハードウェア説明

### 4.1 ハードウェア構成例

図 4-1に本アプリケーションノートで使用するハードウェア構成例を示します。

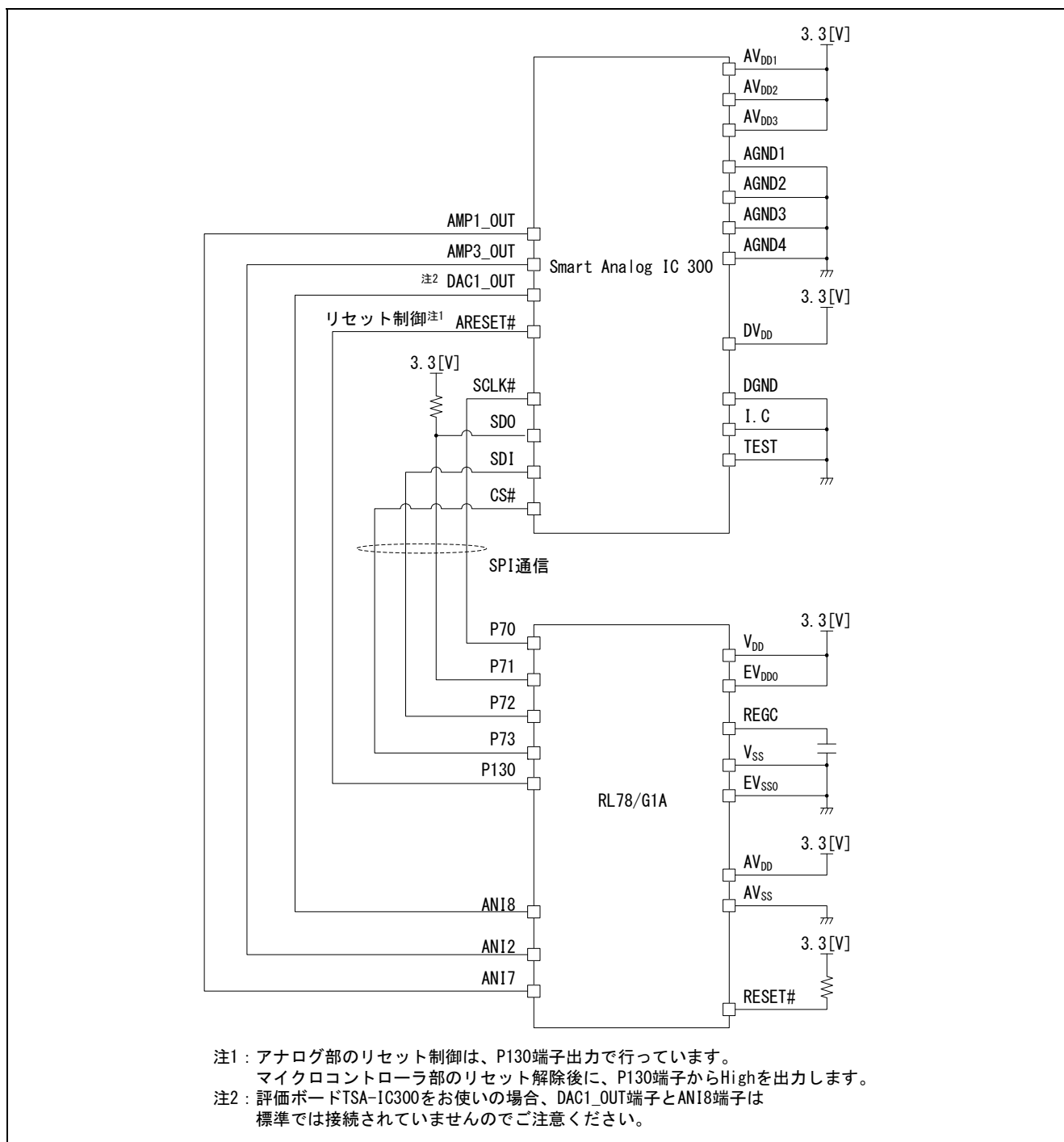


図 4-1 ハードウェア構成例

**注意** この回路イメージは接続の概要を示すため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<sub>DD</sub> または V<sub>SS</sub> に接続してください）。

## 4.2 使用機能一覧

本アプリケーションノートで使用する Smart Analog IC 300 (RAA730300) と RL78/G1A (R5F10ELE) の機能と用途を表 4-1に示します。

表 4-1 Smart Analog IC 300 (RAA730300) と RL78/G1A (R5F10ELE) の使用する機能と用途

デバイス	機能	用途
Smart Analog IC 300	コンフィギュラブル・アンプ Ch1	自己補正の対象 アンプ構成は差動アンプ構成とし、ゲインは 40dB で使用
	D/A コンバータ Ch1	自己補正のゲイン誤差測定および入力換算オフセット電圧測定で使用 (差動アンプ構成のバイアス電圧生成)
	D/A コンバータ Ch5	自己補正のゲイン誤差測定および入力換算オフセット電圧測定で使用 (差動アンプ構成の印加信号)
	D/A コンバータ Ch6	自己補正のゲイン誤差測定で使用 (差動アンプ構成の印加信号)
	SPI インタフェース	RL78/G1A との SPI 通信制御
RL78/G1A	A/D コンバータ	コンフィギュラブル・アンプおよび D/A コンバータの出力電圧をデジタル値に変換
	高速オンチップ・オシレータ・クロック	メイン・システム・クロックとして 32 MHz のクロックを生成
	シリアル・アレイ・ユニット 1 チャンネル 1	3 線シリアル I/O 機能 (CSI21) を使用して、アナログ部との SPI 通信制御
	I/O ポート	Smart Analog IC 300 のリセット制御および SPI 通信制御におけるチップ・セレクト信号 (CS#) の制御
	タイマ・アレイ・ユニット 0 チャンネル 5	Smart Analog IC 300 のセットリング・タイムを計測するために使用
	リアルタイム・クロック	自己補正のタイミング生成 (1 分ごとに割り込み発生)
	DMA コントローラ	取得した A/D 変換値 (ADCR) を内蔵 RAM に転送するために使用

## 4.3 使用端子一覧

本アプリケーションノートにおける Smart Analog IC 300 の使用端子と機能を表 4-2に示します。

表 4-2 Smart Analog IC 300 の使用端子と機能

端子名	入出力	機能
AMP1_OUT	出力	誤差測定 (ゲイン誤差、入力換算オフセット電圧) のため、A/D コンバータアナログ入力端子 7 (ANI7) に接続
DAC1_OUT	出力	誤差測定 (入力換算オフセット電圧測定時のみ) のため、A/D コンバータアナログ入力端子 8 (ANI8) に接続
ARESET#	入力	Smart Analog IC 300 のリセット機能を制御するため、P130 に接続
SCLK#	入力	SPI 通信制御のため、3 線シリアル I/O 機能の SCLK#と接続
SDO	入出力	SPI 通信制御のため、3 線シリアル I/O 機能の SDO と接続
SDI	入出力	SPI 通信制御のため、3 線シリアル I/O 機能の SDI と接続
CS#	入力	SPI 通信制御のため、P73 と接続

## 5. 自己補正動作

本章では、ゲイン誤差および入力換算オフセット電圧測定時の手順やタイミング、状態遷移、Smart Analog IC 300 の設定値について説明します。

### 5.1 自己補正のタイミングチャート

本アプリケーションノートでは、リアルタイム・クロックにより自己補正を開始するタイミングを生成し、ゲイン誤差測定、入力換算オフセット電圧測定の順で自己補正を行います。図 5-1 に本アプリケーションノートの自己補正開始タイミングについて説明します。

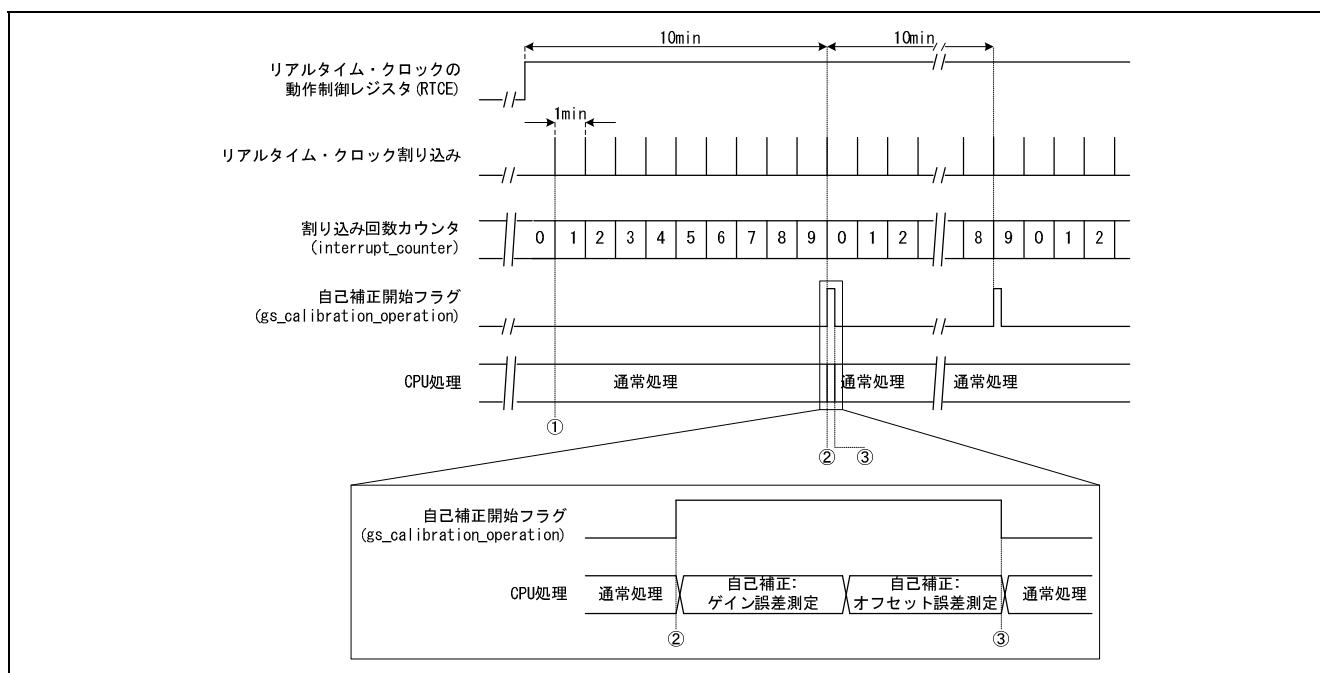


図 5-1 自己補正開始のタイミングチャート

- ① リアルタイム・クロックの定周期割り込み (INTRTC) により、1 分ごとに割り込みが発生し、RTC 割り込み回数カウンタの値に 1 を足します。
- ② RTC 割り込み回数カウンタの値が 0 (10 分経過) になると、自己補正開始フラグがセットされます。
- ③ 自己補正で必要な計測 (ゲイン誤差測定、入力換算オフセット電圧測定) が完了すると、自己補正開始フラグがクリアされます。

## 5.2 ゲイン誤差測定

ゲイン誤差測定では、コンフィギュラブル・アンプに入力する電圧と出力された電圧からゲインを求め、設定値との誤差を算出します。なお、ゲイン誤差測定では、D/A コンバータの誤差やコンフィギュラブル・アンプのオフセット電圧の影響を限りなく小さくするため、2 点間の電圧の差から算出しています。以下に、測定手順やタイミングチャート、Smart Analog IC 300 のレジスタ設定値を示します。

### 5.2.1 ゲイン誤差測定手順

ゲイン誤差測定では、原理編の図 3-2 の手順で計測を行います。本アプリケーションノートでは、コンフィギュラブル・アンプ Ch1 を 40 dB の差動アンプ構成に設定し、D/A コンバータ Ch1、Ch5、Ch6 を用いて測定します。以下に、ゲイン誤差測定手順を示します。

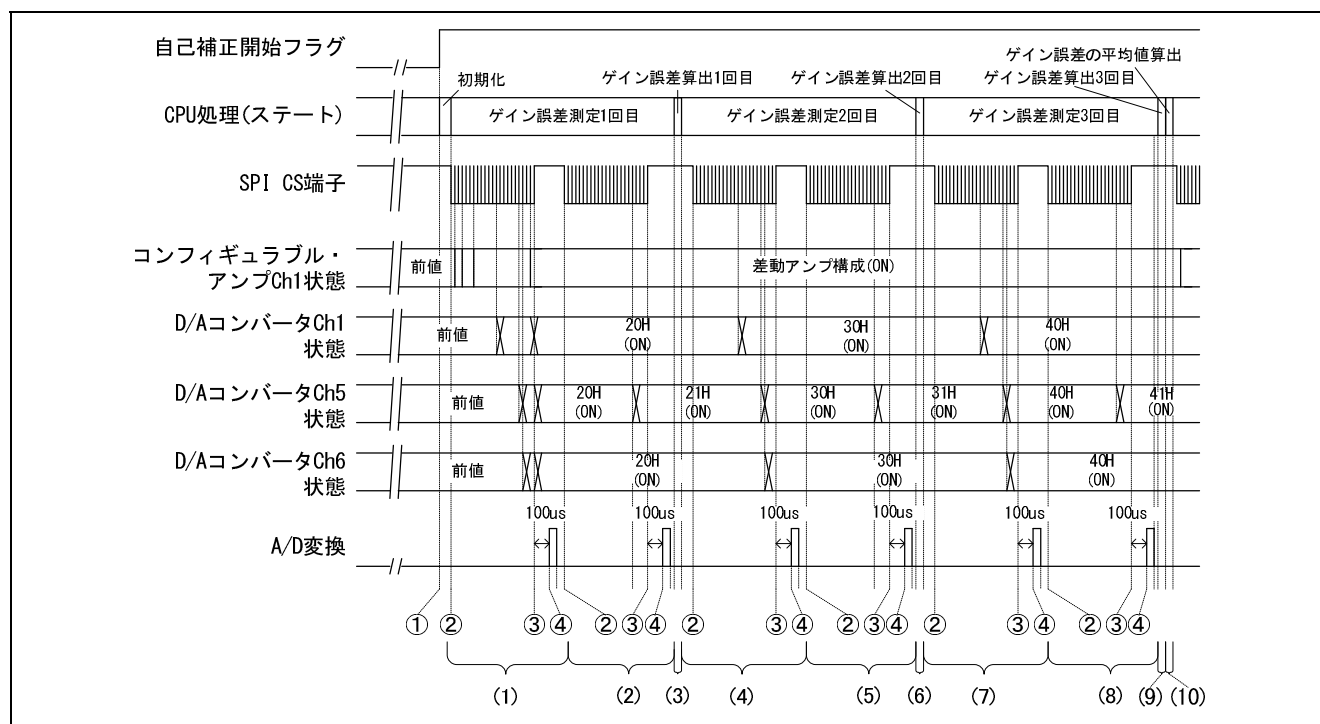


図 5-2 ゲイン誤差測定のタイミングチャート

#### ① A/D 変換初期化

自己補正開始フラグがセットされると、自己補正処理を開始するために必要な初期化処理を行います。初期化には、A/D コンバータの A/D 変換トリガ・モードをソフトウェア・トリガ・モード、連続変換モードに設定します。



### (1) 20H 設定時の最小値電圧の測定

#### ② Smart Analog のレジスタ設定変更

Smart Analog IC 300 のレジスタ設定を下記に変更します。(図 5-3参照)

##### (ア) コンフィギュラブル・アンプのアンプ構成変更

コンフィギュラブル・アンプ Ch1 をゲイン 40 dB の差動アンプ構成に設定するため、コンフィギュレーション・レジスタ 1 (CONFIG1) を 90H に、ゲイン制御レジスタ 1 (GC1) を 11H 設定します。

##### (イ) MPX 設定変更

コンフィギュラブル・アンプ Ch1 の反転入力端子に D/A コンバータ Ch6 を、非反転入力端子には D/A コンバータ Ch5 を接続するため、MPX 設定レジスタ 1 (MPX1) を E8H に、コンフィギュレーション・レジスタ 2 (CONFIG2) を 83H に設定します。

##### (ウ) D/A コンバータ設定変更

D/A コンバータ Ch1、Ch5、Ch6 の DAC 制御レジスタ (DAC1C、DAC5C、DAC6C) をそれぞれ 20H に設定します。

##### (エ) コンフィギュラブル・アンプおよび D/A コンバータ動作許可

パワー制御レジスタ 1 (PC1) を 11H に、パワー制御レジスタ 2 (PC2) を 60H に設定し、コンフィギュラブル・アンプ Ch1、D/A コンバータ Ch1、Ch5、Ch6 の動作を許可します。

#### ③ Smart Analog のレジスタ設定が完了後、セットリング・タイム (100 $\mu$ 秒) を待ちます。

#### ④ セットリング・タイム経過後、コンフィギュラブル・アンプ Ch1 の出力電圧 (AMP1\_OUT 端子) を A/D 変換します。この時、A/D 変換を連続で 10 回実施し、10 回のデータから最大値、最小値を除いた 8 回で平均化処理を実施します。

### (2) 20H 設定時の最大値電圧の測定

20H 設定時の最大値電圧を測定するため、上記②～④の動作を繰り返します。ただし、②の(ウ)で設定する D/A コンバータ Ch5 は 21H に設定します。(図 5-3参照)

### (3) 20H 設定のゲイン誤差の算出

(1)、(2)で取得した A/D 変換値と D/A コンバータ設定値からゲイン誤差を算出します。

### (4) 30H 設定時の最小値電圧の測定

(1)の手順と同様に測定を行います。ただし、各 D/A コンバータに設定する値は 30H に変えます。

### (5) 30H 設定時の最大値電圧の測定

(2)の手順と同様に測定を行います。ただし、D/A コンバータ Ch5 に設定する値は 31H に変えます。

### (6) 30H 設定のゲイン誤差の算出

(4)、(5)で取得した A/D 変換値と D/A コンバータ設定値からゲイン誤差を算出します。

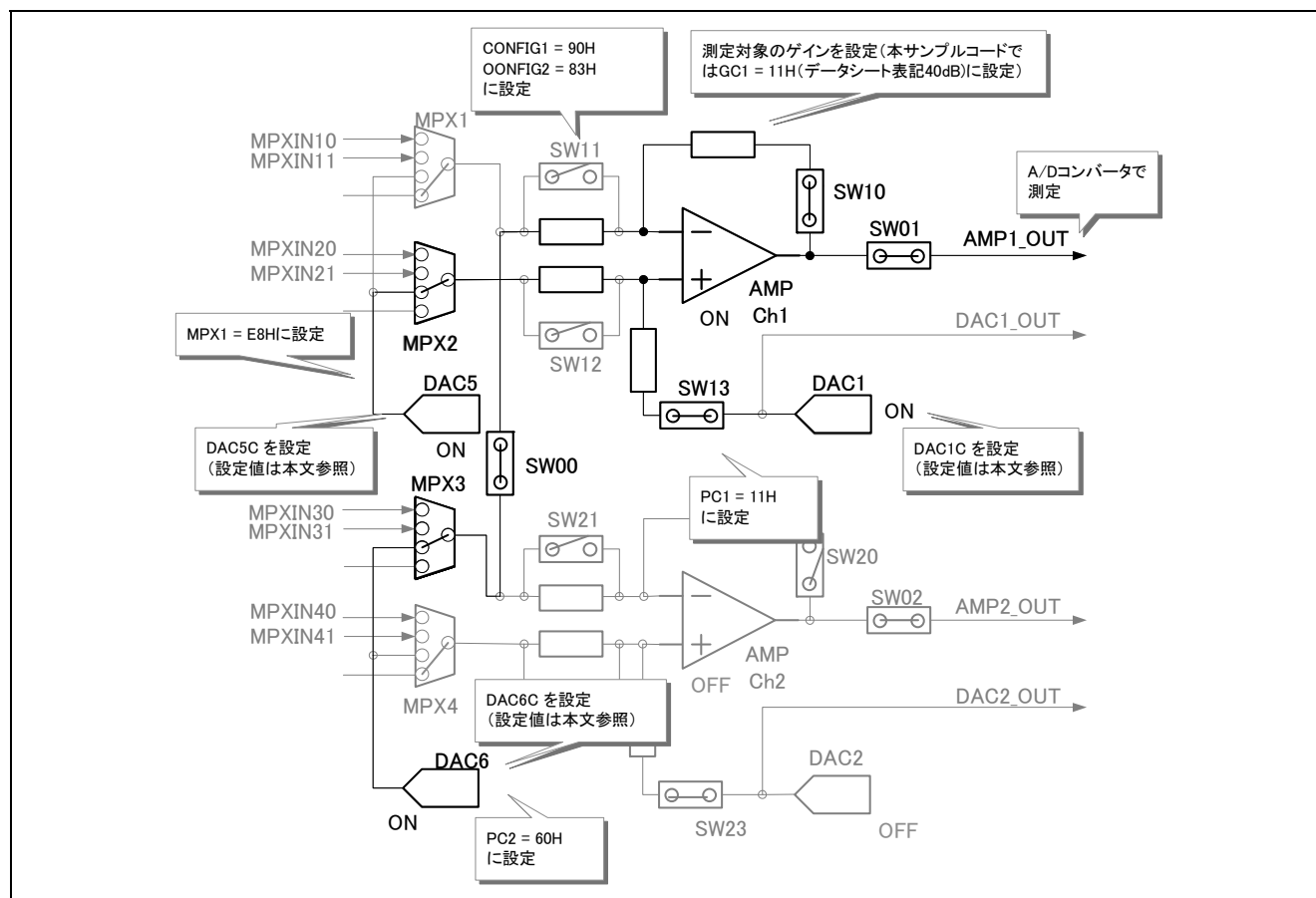


図 5-3 ゲイン誤差測定

## (7) 40H 設定時の最小値電圧の測定

(1)の手順と同様に測定を行います。ただし、各 D/A コンバータに設定する値は 40H に変えます。

## (8) 40H 設定時の最大値電圧の測定

(7)の手順と同様に測定を行います。ただし、D/A コンバータ Ch5 に設定する値は 41H に変えます。

## (9) 40H 設定のゲイン誤差の算出

(7)、(8)で取得した A/D 変換値と D/A コンバータ設定値からゲイン誤差を算出します。

## (10) ゲイン誤差の平均値算出

(3)、(6)、(9)で算出したゲイン誤差から平均値を算出します。

以上の手順で、ゲイン誤差を算出することができます。

## 5.2.2 ゲイン誤差測定時の Smart Analog IC 300 レジスタ設定値

ゲイン誤差測定時の Smart Analog IC 300 のレジスタ設定値を表 5-2に示します。レジスタ設定値の詳細につきましては、「6.4 Smart Analog IC 300 のレジスタ値」および「RAA730300 モノリシック・プログラマブル・アナログ IC のデータシート」をご参照ください。

表 5-1 ゲイン誤差測定時の Smart Analog IC 300 レジスタ設定値

アドレス	SPI 制御レジスタ名称		設定値
00H	コンフィギュレーション・レジスタ 1	CONFIG1	90H
01H	コンフィギュレーション・レジスタ 2	CONFIG2	83H
03H	MPX 設定レジスタ 1	MPX1	E8H
04H	MPX 設定レジスタ 2	MPX2	00H
05H	MPX 設定レジスタ 3	MPX3	00H
06H	ゲイン制御レジスタ 1	GC1	11H
07H	ゲイン制御レジスタ 2	GC2	00H
08H	ゲイン制御レジスタ 3	GC3	00H
09H	アンプ動作モード制御レジスタ	AOMC	00H
0BH	LDO 出力電圧制御レジスタ	LDOC	00H
0CH	DAC 基準制御レジスタ	DACRC	00H
0DH	DAC 制御レジスタ 1	DAC1C	**H
0EH	DAC 制御レジスタ 2	DAC2C	80H
0FH	DAC 制御レジスタ 3	DAC3C	80H
10H	DAC 制御レジスタ 4	DAC4C	80H
11H	パワー制御レジスタ 1	PC1	11H
12H	パワー制御レジスタ 2	PC2	60H
13H	リセット制御レジスタ	RC	00H
14H	入力モード制御レジスタ	IMS	00H
15H	DAC 制御レジスタ 5	DAC5C	**H
16H	DAC 制御レジスタ 6	DAC6C	**H
17H	DAC 制御レジスタ 7	DAC7C	80H

備考 \* : DAC1C、DAC5C、DAC6C の設定値は各測定条件により異なります。

### 5.3 入力換算オフセット電圧測定

入力換算オフセット電圧は、コンフィギュラブル・アンプに入力した電圧と出力された電圧の差から算出することができます。本アプリケーションノートでは、コンフィギュラブル・アンプ Ch1 の入力換算オフセット電圧を測定します。入力換算オフセット電圧の測定原理は原理編の 3.2 節をご参照ください。以下に、測定手順やタイミングチャート、Smart Analog IC 300 のレジスタ設定値を示します。

#### 5.3.1 入力換算オフセット電圧測定手順

入力換算オフセット電圧の測定は、ゲイン誤差測定が完了後開始されます。

以下に、入力換算オフセット電圧の測定手順を示します。

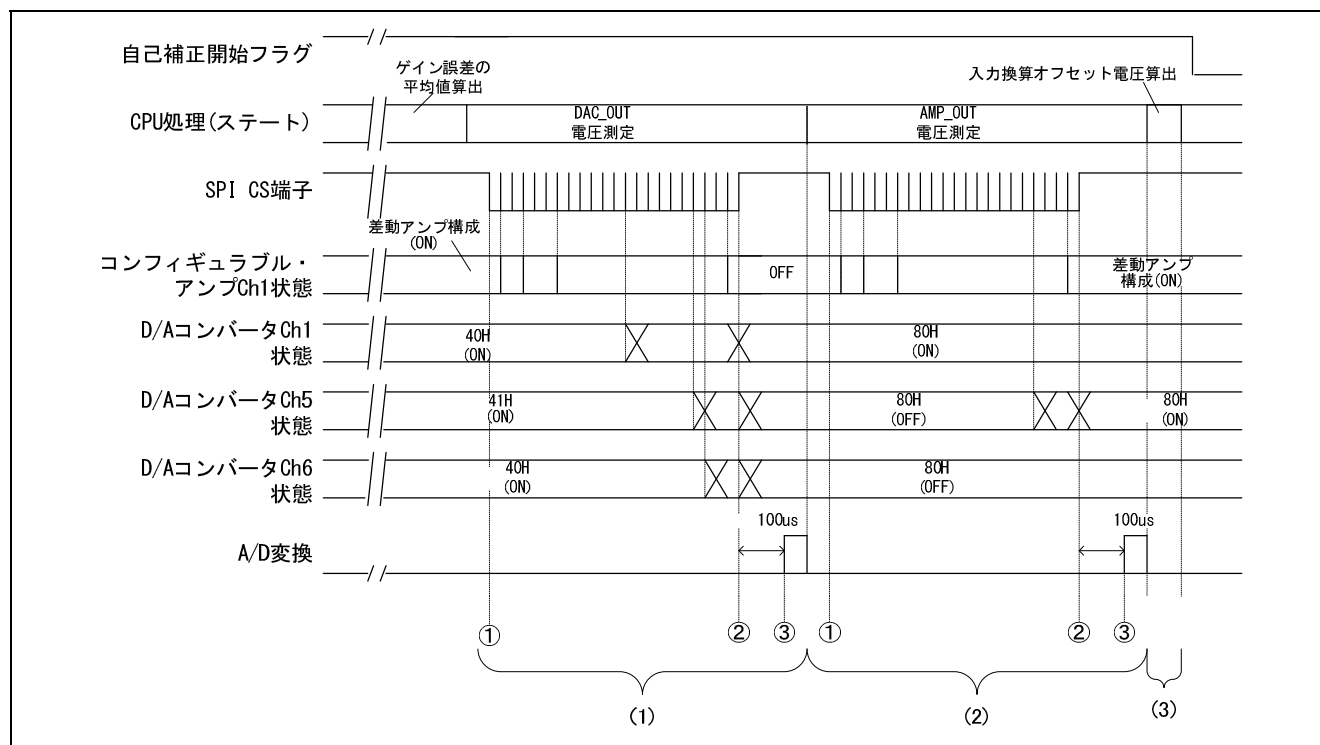


図 5-4 入力換算オフセット電圧測定のタイミングチャート

#### (1) D/A コンバータ Ch1 の出力電圧（DAC\_OUT 電圧）測定

##### ① Smart Analog のレジスタ設定変更

Smart Analog IC 300 のレジスタ設定を下記に変更します。(図 5-5参照)

##### (ア) コンフィギュラブル・アンプの SW 切り替え

D/A コンバータ出力電圧測定に影響を受けないようコンフィギュラブル・アンプ Ch1 の SW を切り替えます。コンフィギュレーション・レジスタ 1 (CONFIG1) を 80H に設定します。

##### (イ) MPX 設定変更

コンフィギュラブル・アンプ Ch1 の入力端子をリセット後の値に変更します。本設定は測定には影響がないため、変更しなくても問題はありません。

##### (ウ) D/A コンバータ設定変更

D/A コンバータ Ch1、Ch5、Ch6 の DAC 制御レジスタ (DAC1C、DAC5C、DAC6C) をそれぞれ 80H に設定します。

- (エ) コンフィギュラブル・アンプおよび D/A コンバータ動作許可/禁止  
 パワー制御レジスタ 1 (PC1) を 10H に、パワー制御レジスタ 2 (PC2) を 00H に設定し、コンフィギュラブル・アンプ Ch1、D/A コンバータ Ch5、Ch6 の動作を禁止、D/A コンバータ Ch1 の動作を許可します。
- ② Smart Analog のレジスタ設定が完了後、セッティング・タイム (100 $\mu$  秒) を待ちます。
- ③ セッティング・タイム経過後、D/A コンバータ Ch1 の出力電圧 (DAC1\_OUT 端子) を A/D 変換します。この時、A/D 変換を連続で 10 回実施し、10 回のデータから最大値、最小値を除いた 8 回で平均化処理を実施します。

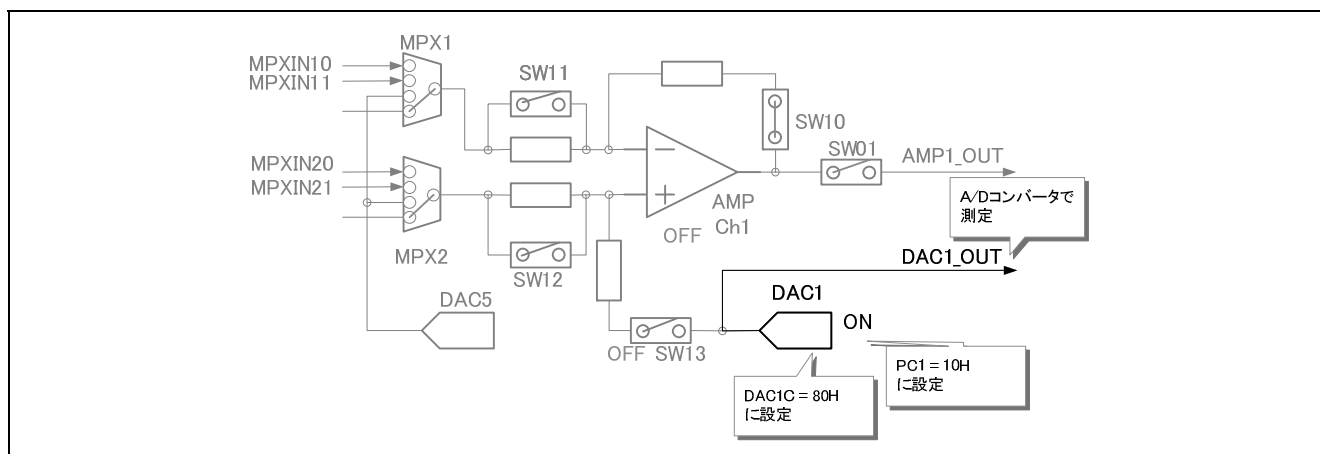


図 5-5 D/A コンバータ Ch1 の出力電圧測定

(2) コンフィギュラブル・アンプ Ch1 の出力電圧 (AMP1\_OUT 電圧) 測定

- ① **Smart Analog** のレジスタ設定変更  
Smart Analog IC 300 のレジスタ設定を下記に変更します。（図 5-6参照）
  - (ア) **コンフィギュラブル・アンプのアンプ構成変更**  
コンフィギュラブル・アンプ Ch1 をゲイン 40 dB の差動アンプ構成に設定するため、コンフィギュレーション・レジスタ 1 (CONFIG1) を 90H に、ゲイン制御レジスタ 1 (GC1) を 11H 設定します。
  - (イ) **MPX 設定変更**  
コンフィギュラブル・アンプ Ch1 の反転入力端子および非反転入力端子に D/A コンバータ Ch5 を接続するため、MPX 設定レジスタ 1 (MPX1) を A0H に設定します。
  - (ウ) **D/A コンバータ設定変更**  
D/A コンバータ Ch1、Ch5、Ch6 の DAC 制御レジスタ (DAC1C、DAC5C、DAC6C) をそれぞれ 80H に設定します。
  - (エ) **コンフィギュラブル・アンプおよび D/A コンバータ動作許可**  
パワー制御レジスタ 1 (PC1) を 11H に、パワー制御レジスタ 2 (PC2) を 20H に設定し、コンフィギュラブル・アンプ Ch1、D/A コンバータ Ch1、Ch5 の動作を許可します。
- ② **Smart Analog** のレジスタ設定が完了後、セットリング・タイム(100 $\mu$  秒)を待ちます。
- ③ セットリング・タイム経過後、コンフィギュラブル・アンプ Ch1 の出力電圧 (AMP1\_OUT 端子) を A/D 変換します。この時、A/D 変換を連続で 10 回実施し、10 回のデータから最大値、最小値を除いた 8 回で平均化処理を実施します。

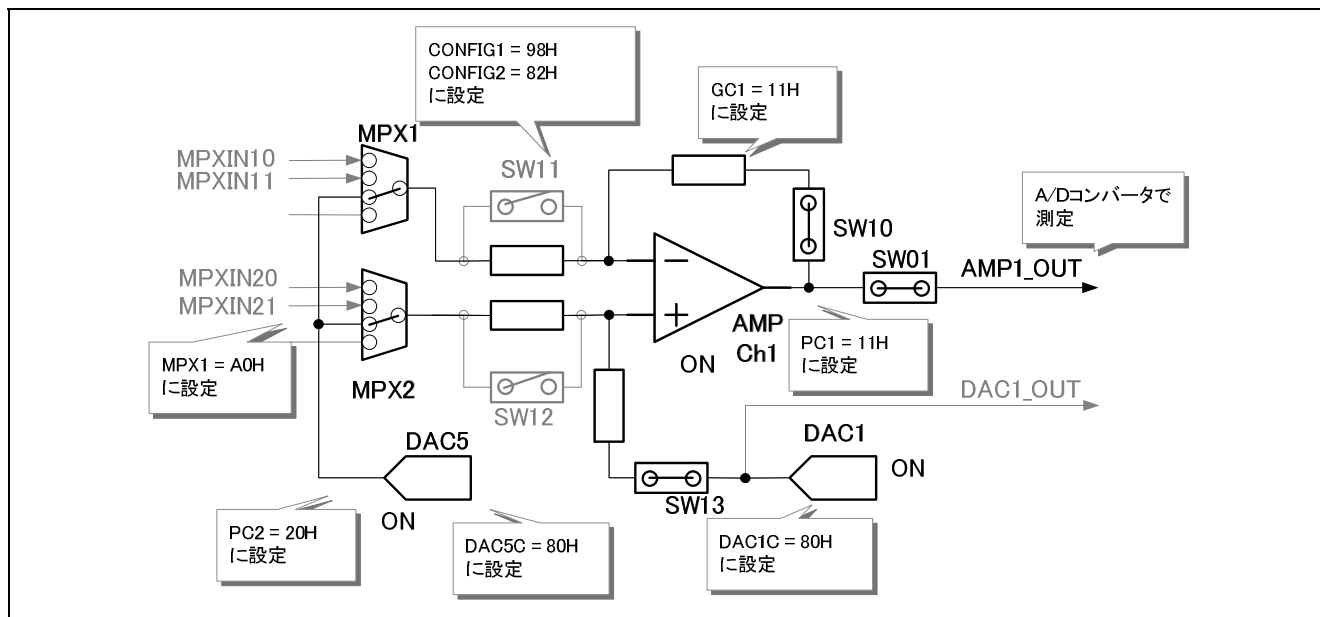


図 5-6 コンフィギュラブル・アンプ Ch1 の出力電圧測定

### (3) 入力換算オフセット電圧の算出

入力換算オフセット電圧を算出します。入力換算オフセット電圧は、(2)で取得した AMP1\_OUT 端子の A/D 変換値から(1)で取得した DAC1\_OUT 端子の A/D 変換値を減算し、ゲイン 40 dB で除算することで算出することができます。

以上の手順で、入力換算オフセット電圧を測定することができます。

## 5.3.2 入力換算オフセット電圧測定時の Smart Analog IC 300 レジスタ設定値

入力換算オフセット電圧測定時の Smart Analog IC 300 のレジスタ設定値を表 5-2に示します。レジスタ設定値の詳細につきましては、「6.4 Smart Analog IC 300 のレジスタ値」および「RAA730300 モノリシック・プログラマブル・アナログ IC のデータシート」をご参照ください。

表 5-2 入力換算オフセット電圧測定時の Smart Analog IC 300 レジスタ設定値

アドレス	SPI 制御レジスタ名称		設定値	
			D/A コンバータ Ch1 出力電圧測定	コンフィギュラブル・アンプ Ch1 出力電圧測定
00H	コンフィギュレーション・レジスタ 1	CONFIG1	80H	90H
01H	コンフィギュレーション・レジスタ 2	CONFIG2	80H	82H
03H	MPX 設定レジスタ 1	MPX1	00H	A0H
04H	MPX 設定レジスタ 2	MPX2	00H	00H
05H	MPX 設定レジスタ 3	MPX3	00H	00H
06H	ゲイン制御レジスタ 1	GC1	00H	11H
07H	ゲイン制御レジスタ 2	GC2	00H	00H
08H	ゲイン制御レジスタ 3	GC3	00H	00H
09H	アンプ動作モード制御レジスタ	AOMC	00H	00H
0BH	LDO 出力電圧制御レジスタ	LDOC	00H	00H
0CH	DAC 基準制御レジスタ	DACRC	00H	00H
0DH	DAC 制御レジスタ 1	DAC1C	80H	80H
0EH	DAC 制御レジスタ 2	DAC2C	80H	80H
0FH	DAC 制御レジスタ 3	DAC3C	80H	80H
10H	DAC 制御レジスタ 4	DAC4C	80H	80H
11H	パワー制御レジスタ 1	PC1	10H	11H
12H	パワー制御レジスタ 2	PC2	00H	20H
13H	リセット制御レジスタ	RC	00H	00H
14H	入力モード制御レジスタ	IMS	00H	00H
15H	DAC 制御レジスタ 5	DAC5C	80H	80H
16H	DAC 制御レジスタ 6	DAC6C	80H	80H
17H	DAC 制御レジスタ 7	DAC7C	80H	80H

## 6. ソフトウェア説明

## 6.1 関数一覧

表 6-1 関数一覧(1)

ファイル名	関数名	備考
r_systeminit.c	hdwinit	コード生成ツールで生成
	R_Systeminit	
r_main.c	main	
	R_MAIN_UserInit	
calibration.c	R_Calibration	
	R_CalibrationIsDoing	
	calibration_init	
	calibration_end	
	calibration_trigger_check	
	calibration_get_state_rtc_tbl	
self_calibration.c	R_SelfCalibrationInit	
	setup_self_cal_AFE_measure_ch	
	R_SelfCalibAmp1DiffOffsetDac	
	R_SelfCalibAmp1DiffOffsetAmp	
	R_SelfCalibAmp1DiffOffsetCalc	
	R_SelfCalibAmp1DiffGainMeasure	
	R_SelfCalibAmp1DiffGainCalc	
	R_SelfCalibAmp1DiffGainAverage	
	R_SelfCalibrationMeasure	
	amp1_diff_calibration_setting	
smart_analog_sample_code_common.c	R_SmartAnalogHardReset	
	R_SmartAnalogMeasureControllInit	
	R_CalcAverageExceptMinMax	
	R_SmartAnalogRegBufSearch	
	R_SmartAnalogRegBufUpdate	
r_sa_spi_control_register.c	R_SPI_SmartAnalogRead	RL78/G1E アナログ部との SPI 通信サンプルコード (R01AN1130J)アプリケーション ノートのサンプル コード流用
	R_SPI_SmartAnalogWrite	
	R_SPI_SmartAnalogWriteVerify	
	R_SPI_SmartAnalogReadBit	
	R_SPI_SmartAnalogWriteBit	
	R_SPI_SmartAnalogWriteVerifyBit	
r_cg_timer.c	R_TAU0_Create	コード生成ツールで生成
	R_TAU0_Channel0_Start	
	R_TAU0_Channel0_Stop	
	R_TAU0_Channel4_Start	
	R_TAU0_Channel4_Stop	
	R_TAU0_Channel5_Start	
	R_TAU0_Channel5_Stop	
r_cg_timer_user.c	R_GetTickCount	
	R_CmpTickCount	
	R_TAU0_TDR05_set	
	r_tau0_channel5_interrupt	



表 6-2 関数一覧(2)

ファイル名	関数名	備考
r_cg_serial_user.c	r_csi21_interrupt	コード生成ツールで生成
	r_csi21_callback_error	コード生成ツールで生成し、処理を追加。
	r_csi21_callback_receiveend	RL78/G1E アナログ部との SPI 通信サンプルコード (R01AN1130J) アプリケーションノートのサンプルコード流用
r_cg_serial.c	R_SAU1_Create	コード生成ツールで生成し、処理を追加。
	R_CSI21_Create	RL78/G1E アナログ部との SPI 通信サンプルコード (R01AN1130J) アプリケーションノートのサンプルコード流用
	R_CSI21_Start	
	R_CSI21_Stop	
	R_CSI21_Send_Receive	
r_cg_rtc.c	R_RTC_Create	コード生成ツールで生成
	R_RTC_Start	
	R_RTC_Stop	
	R_RTC_Get_CounterValue	
	R_RTC_Set_CounterValue	
	R_RTC_Set_ConstPeriodInterruptOn	
r_cg_rtc_user.c	R_SelfCalibrationFlagCheck	
	R_SelfCalibrationFlagClear	
	r_rtc_interrupt	コード生成ツールで生成
	r_rtc_callback_constperiod	コード生成ツールで生成し、処理を追加。
	r_rtc_interrupt_counter	
r_cg_port.c	R_PORT_Create	コード生成ツールで生成
r_cg_dmac.c	R_DMAM1_Create	コード生成ツールで生成
	R_DMAM1_Start	
	R_DMAM1_Stop	
r_cg_dmac_user.c	R_DMAM1_ADC_Create	
	r_dmac1_interrupt	コード生成ツールで生成し、処理を追加。
r_cg_cgc.c	R_CGC_Create	コード生成ツールで生成
r_cg_adc.c	R_ADC_Create	コード生成ツールで生成
	R_ADC_Start	
	R_ADC_Stop	
	R_ADC_Set_OperationOn	
	R_ADC_Set_OperationOff	
	R_ADC_Get_Result	
r_cg_adc_user.c	R_ADC_EndFlagCheck	
	R_ADC_EndFlagSet	
	R_ADC_EndFlagClear	
	R_ADC_MaskStart	
	R_ADC_ChangeSoftTrigger	
	R_ADC_StartSetting	
	R_ADC_EndSetting	
	R_ADC_ChannelChange	
	r_adc_interrupt	コード生成ツールで生成

## 6.2 関数仕様

本アプリケーションノートにおける主要な関数の関数仕様を以下に示します。

### (1) main 関数

宣言	void main(void)
概要	メイン関数 ・ R_MAIN_UserInit 関数の呼び出し ・ R_Calibration 関数の呼び出し
引数	なし
グローバル変数	なし
戻り値	なし

### (2) R\_MAIN\_UserInit 関数

宣言	void R_MAIN_UserInit(void)
概要	ユーザーインターフェース初期化関数 ・ 自己補正に必要な機能の初期化を行う
引数	なし
グローバル変数	なし
戻り値	なし

### (3) R\_Calibration 関数

宣言	void R_Calibration( void )
概要	キャリブレーション制御関数 ・ キャリブレーション処理を実施する ・ キャリブレーション動作が完了するまでループを繰り返す
引数	なし
グローバル変数	なし
戻り値	なし

### (4) R\_CalibrationIsDoing 関数

宣言	uint8_t R_CalibrationIsDoing (void)
概要	キャリブレーショントリガ判定関数 ・ キャリブレーショントリガの有無を判定する ・ キャリブレーション動作のトリガを追加する場合は本関数に追加する
引数	なし
グローバル変数	なし
戻り値	ret 0U : トリガなし 1U : トリガ有り

## (5) calibration\_init 関数

宣言	static void calibration_init (void)
概要	キャリブレーション初期化関数 ・ A/D 変換のトリガをソフトウェアトリガに変更する
引数	なし
グローバル変数	なし
戻り値	なし

## (6) calibration\_end 関数

宣言	static void calibration_end (void)
概要	キャリブレーション終了処理関数 ・ 自己補正完了時に停止/起動させる機能がある場合は本関数で記述する
引数	なし
グローバル変数	なし
戻り値	なし

## (7) calibration\_trigger\_check 関数

宣言	static calibration_state_t calibration_trigger_check ( calibration_state_t pre_state ,uint8_t * p_ret_state_execution_flag)
概要	キャリブレーショントリガチェック関数 ・ キャリブレーショントリガのチェックを行う ・ キャリブレーションステートの更新を行う
引数	pre_state 現在のステート p_ret_state_execution_flag ステート実行フラグのポインタ
グローバル変数	なし
戻り値	calibration_state_ret 更新したステート

## (8) calibration\_get\_state\_rtc\_tbl 関数

宣言	static calibration_state_t calibration_get_state_rtc_tbl ( calibration_state_t pre_state )
概要	自己補正ステート制御関数 ・ 自己補正ステートをチェック、更新する
引数	pre_state 現在のステート
グローバル変数	なし
戻り値	ret_state 更新したステート

## (9) R\_SelfCalibrationInit 関数

宣言	void R_SelfCalibrationInit (void)
概要	自己補正グローバル変数初期化関数 ・ 自己補正用のグローバル変数の初期化
引数	なし
グローバル変数	gs_self_cal_sa_reg_buf 自己補正用設定格納変数 gp_self_cal_setting_data 自己補正用設定値の格納変数のポインタ配列
戻り値	なし

## (10) setup\_self\_cal\_AFE\_measure\_ch 関数

宣言	void setup_self_cal_AFE_measure_ch ( self_calibration_state_t self_cal_cnt )
概要	SmartAnalog レジスタ、A/D コンバータ設定変更関数 ・ SmartAnalog のレジスタ設定値を自己補正用設定格納変数の設定に書き換える ・ A/D コンバータのアナログ入力チャネルを自己補正用の設定に書き換える
引数	set_cnt 自己補正中の SA 設定格納変数(gs_self_cal_sa_reg_buf)用の添え字
グローバル変数	gs_self_cal_sa_reg_buf 自己補正用設定格納変数
戻り値	なし

## (11) R\_SelfCalibAmp1DiffOffsetDac 関数

宣言	uint8_t R_SelfCalibAmp1DiffOffsetDac ( void )
概要	DAC1_OUT 測定関数 ・ コンフィギュラブル・アンプ Ch1 の差動アンプ構成のオフセット電圧測定用関数の一つ ・ D/A コンバータ Ch1 の出力端子(DAC1_OUT 端子)を A/D コンバータで測定する
引数	なし
グローバル変数	gs_offset_cal_data オフセット電圧算出用測定値格納変数
戻り値	ret_state D_OFF:未完了 D_ON:完了

## (12) R\_SelfCalibAmp1DiffOffsetAmp 関数

宣言	uint8_t R_SelfCalibAmp1DiffOffsetAmp ( void )
概要	AMP1_OUT 測定関数 ・ コンフィギュラブル・アンプ Ch1 の差動アンプ構成のオフセット電圧測定用関数の一つ ・ コンフィギュラブル・アンプ Ch1 を 40dB の差動アンプ構成に設定する ・ コンフィギュラブル・アンプ Ch1 の反転入力端子、非反転入力端子に D/A コンバータ Ch5 を接続する ・ コンフィギュラブル・アンプ Ch1 の出力端子(AMP1_OUT 端子)を A/D コンバータで測定する
引数	なし
グローバル変数	gs_offset_cal_data オフセット電圧算出用測定値格納変数
戻り値	ret_state D_OFF : 未完了 D_ON : 完了

## (13) R\_SelfCalibAmp1DiffOffsetCalc 関数

宣言	uint8_t R_SelfCalibAmp1DiffOffsetCalc ( amp_num_t amp_num, amp_config_t amp_config, uint8_t gain_reg_val )
概要	入力換算オフセット電圧算出関数 ・ コンフィギュラブル・アンプ Ch1 の差動アンプ構成でのオフセット電圧算出関数 ・ R_SelfCalibAmp1DiffOffsetDac 関数と R_SelfCalibAmp1DiffOffsetAmp 関数で測定した値から入力換算オフセット電圧を算出する
引数	amp_num アンプの CH 番号 amp_config アンプの構成番号 gain_reg_val ゲインのレジスタ値
グローバル変数	gs_offset_cal_data オフセット電圧算出用測定値格納変数 g_amp_table Smart Analog IC 300 ゲイン・オフセット電圧格納テーブル
戻り値	D_ON : 完了

## (14) R\_SelfCalibAmp1DiffGainMeasure 関数

宣言	uint8_t R_SelfCalibAmp1DiffGainMeasure (uint8_t dac_reg_val, ic300_diff_amp_gain_t amp1_gain, self_calibration_state_t self_cal_cnt)
概要	ゲイン測定関数 ・ コンフィギュラブル・アンプ Ch1 を 40dB の差動アンプ構成に設定する ・ D/A コンバータ Ch5 を最小値設定にし、最小値電圧を取得する ・ D/A コンバータ Ch5 を最大値設定にし、最大値電圧を取得する
引数	dac_reg_val DAC レジスタ設定値 amp1_gain AMP1 のゲインレジスタ設定値 self_cal_cnt 自己補正設定番号
グローバル変数	なし
戻り値	ret_state D_OFF : 処理中 D_ON : 完了 D_ERROR : 引数が不正

## (15) R\_SelfCalibAmp1DiffGainCalc 関数

宣言	uint8_t R_SelfCalibAmp1DiffGainCalc ( self_calibration_dac_cnt_t buf_num )
概要	ゲイン算出関数 ・ 測定したデータからゲインを算出する
引数	buf_num 校正用測定値格納バッファ番号(0:0x20U、1:0x30U、2:0x40U)
グローバル変数	gs_self_cal_gain_data ゲイン格納バッファ(算出したゲインを一時的に保管する)
戻り値	ret_state D_ON : 完了

## (16) R\_SelfCalibAmp1DiffGainAverage 関数

宣言	uint8_t R_SelfCalibAmp1DiffGainAverage ( amp_num_t amp_num, amp_config_t amp_config, uint8_t gain_reg_val)
概要	ゲイン平均化関数 ・ 算出した 3 回分のゲインの平均化 ・ Smart Analog IC 300 ゲイン・オフセット電圧格納テーブルに算出したゲインを保存する
引数	amp_num アンプの CH 番号 amp_config アンプの構成番号 gain_reg_val ゲインのレジスタ値
グローバル変数	gs_self_cal_gain_data ゲイン格納バッファ(0x20、0x30、0x40 設定で測定したゲインが格納されている) g_amp_table Smart Analog IC 300 ゲイン・オフセット電圧格納テーブル (本関数内で自己補正で得られた結果を格納する)
戻り値	ret_state D_ON : 完了

## (17) R\_SelfCalibrationMeasure 関数

宣言	uint8_t R_SelfCalibrationMeasure ( uint16_t * p_adcr, self_calibration_state_t self_cal_cnt)
概要	自己補正用 A/D 変換制御関数 ・ setup_self_cal_AFE_measure_ch 関数を呼び出し、Smart Analog レジスタ設定を書き換える ・ Smart Analog のセットリング・タイムを待つ ・ A/D 変換値を取得する
引数	p_adcr A/D 変換値(本関数内で書き換える) self_cal_cnt 自己補正設定番号
グローバル変数	なし
戻り値	ret 0U:測定中 1U:完了

## (18) amp1\_diff\_calibration\_setting 関数

宣言	static void amp1_diff_calibration_setting ( uint8_t set_dac_reg, ic300_diff_amp_gain_t set_gain_reg, self_calibration_state_t self_cal_cnt )
概要	ゲイン測定時の DAC、ゲイン設定関数 ・ gs_self_cal_sa_reg_buf 変数の D/A コンバータ Ch1、Ch5、Ch6 の設定をゲイン測定時設定に書き換える ・ gs_self_cal_sa_reg_buf 変数のコンフィギュラブル・アンプ Ch1 のゲインを 40dB 設定に書き換える
引数	set_dac_reg DAC1,DAC5,DAC6 の設定値を書き換えるデータ set_gain_reg AMP1 の GAIN の設定値を書き換えるデータ self_cal_cnt 自己補正設定番号
グローバル変数	gs_self_cal_sa_reg_buf 自己補正用設定格納変数
戻り値	なし

## (19) R\_SmartAnalogHardReset 関数

宣言	uint8_t R_SmartAnalogHardReset (void)
概要	SmartAnalog リセット関数 ・ Smart Analog のハードウェアリセットを行う
引数	なし
グローバル変数	なし
戻り値	0U : 成功 1U : 失敗

## (20) R\_SmartAnalogMeasureControllnit 関数

宣言	void R_SmartAnalogMeasureControllnit (void)
概要	Smart Analog 初期化関数 ・ Smart Analog 制御用グローバル変数の初期化
引数	なし
グローバル変数	g_amp_table Smart Analog IC 300 ゲイン・オフセット電圧格納テーブル
戻り値	なし



## (21) R\_CalcAverageExceptMinMax 関数

宣言	uint16_t R_CalcAverageExceptMinMax (const uint16_t * p_buf, uint8_t buf_size)
概要	A/D 変換値の平均化関数 ・ 取得した A/D 変換値から最大値、最小値を除いた値で平均化を行う
引数	p_buf 取得した A/D 変換値が格納されたバッファの先頭アドレス buf_size p_buf のサイズ(平均化回数)
グローバル変数	なし
戻り値	avg_tmp p_buf[0]から p_buf[buf_size-1]までの A/D 変換値から最大値、最小値を除いた平均値

## (22) R\_SmartAnalogRegBufSearch 関数

宣言	uint8_t R_SmartAnalogRegBufSearch (sa_ctl_set_data_t * p_sa_buf, uint8_t target_addr)
概要	Smart Analog レジスタ探索関数 ・ sa_ctl_set_data_t 型のバッファから Smart Analog アドレスを探索し、格納されているインデックスを返す
引数	p_sa_buf sa_ctl_set_data_t 型のバッファの先頭アドレス target_addr 探索する Smart Analog のレジスタのアドレス
グローバル変数	なし
戻り値	reg_cnt 指定のアドレスの配列番号 (アドレスが見つからない場合は 0xFF を返す)

## (23) R\_SmartAnalogRegBufUpdate 関数

宣言	uint8_t R_SmartAnalogRegBufUpdate (sa_ctl_set_data_t * p_sa_buf, uint8_t target_addr, uint8_t set_data)
概要	Smart Analog 設定値変更関数 ・ SmartAnalog レジスタ探索関数で設定を書き換えるアドレスを探索し、設定値を書き換える
引数	p_sa_buf sa_ctl_set_data_t 型のバッファの先頭アドレス target_addr 設定を書き換える Smart Analog のレジスタのアドレス set_data 設定を書き換える Smart Analog レジスタの設定値
グローバル変数	なし
戻り値	ret 0U:正常 1U:エラー

## (24) R\_GetTickCount 関数

宣言	uint32_t R_GetTickCount ( void )
概要	タイマカウント取得関数 ・フリーランタイムの現在のカウンタ値を取得する
引数	なし
グローバル変数	gs_freerun_timer フリーランタイムカウンタ変数
戻り値	gs_freerun_timer.tdata.timer32bit

## (25) R\_CmpTickCount 関数

宣言	int32_t R_CmpTickCount ( uint32_t src )
概要	タイマカウント比較関数 ・フリーランタイムの現在のカウンタ値と比較する
引数	src 比較カウンタ値
グローバル変数	なし
戻り値	int32_t 0 未満:未経過 0 以上:経過

## (26) R\_TAU0\_TDR05\_set 関数

宣言	void R_TAU0_TDR05_set (uint16_t set_data)
概要	タイマ・データ・レジスタ 05 変更関数
引数	set_data 設定カウンタ値
グローバル変数	なし
戻り値	なし

## (27) r\_tau0\_channel5\_interrupt 関数

宣言	__interrupt static void r_tau0_channel5_interrupt (void)
概要	INTTM05 割り込み関数
引数	なし
グローバル変数	gs_freerun_timer フリーランタイムカウンタ変数
戻り値	なし

## (28) r\_csi21\_interrupt 関数

宣言	__interrupt static void r_csi21_interrupt (void)
概要	INTCSI21 割り込み関数
引数	なし
グローバル変数	なし
戻り値	なし

## (29) r\_csi21\_callback\_error 関数

宣言	static void r_csi21_callback_error (uint8_t err_type)
概要	CSI21 コールバックエラー関数
引数	err_type エラータイプ
グローバル変数	g_csi21_overrun_flag csi21 オーバーランフラグ
戻り値	なし

## (30) r\_csi21\_callback\_receiveend 関数

宣言	static void r_csi21_callback_receiveend (void)
概要	CSI21 受信完了コールバック関数
引数	なし
グローバル変数	なし
戻り値	なし

## (31) R\_SelfCalibrationFlagCheck 関数

宣言	uint8_t R_SelfCalibrationFlagCheck (void)
概要	自己補正開始トリガフラグチェック関数 ・ 自己補正開始トリガフラグを返す
引数	なし
グローバル変数	gs_calibration_operation 自己補正開始トリガフラグ
戻り値	ret 0U:自己補正開始トリガなし 1U:自己補正開始トリガあり

## (32) R\_SelfCalibrationFlagClear 関数

宣言	void R_SelfCalibrationFlagClear (void)
概要	自己補正開始トリガフラグクリア関数
引数	なし
グローバル変数	gs_calibration_operation 自己補正開始トリガフラグ
戻り値	なし

## (33) r\_rtc\_interrupt 関数

宣言	__interrupt static void r_rtc_interrupt (void)
概要	INTRTC 割り込み関数
引数	なし
グローバル変数	なし
戻り値	なし

## (34) r\_rtc\_callback\_constperiod 関数

宣言	static void r_rtc_callback_constperiod (void)
概要	リアルタイム・クロックコールバック関数
引数	なし
グローバル変数	なし
戻り値	なし

## (35) r\_rtc\_interrupt\_counter 関数

宣言	static void r_rtc_interrupt_counter (void)
概要	自己補正タイミング生成関数 ・ 本関数が呼び出されるごとに呼び出し回数をカウントする ・ 指定の回数に達すると自己補正開始トリガフラグをセットする
引数	なし
グローバル変数	gs_calibration_operation 自己補正開始トリガフラグ
戻り値	なし

## (36) R\_DMAL1\_ADC\_Create 関数

宣言	void R_DMAL1_ADC_Create (uint16_t txnum, uint16_t * p_sendbuf)
概要	DMA1 設定関数 ・ DMA1 を A/D 変換結果転送に設定する
引数	txnum 転送 Byte 数 p_sendbuf 転送バッファのポインタ
グローバル変数	なし
戻り値	なし

## (37) r\_dmac1\_interrupt 関数

宣言	__interrupt static void r_dmac1_interrupt (void)
概要	INTDMA1 割り込み関数
引数	なし
グローバル変数	なし
戻り値	なし

## (38) R\_ADC\_EndFlagCheck 関数

宣言	uint8_t R_ADC_EndFlagCheck (void)
概要	A/D 変換終了フラグチェック関数
引数	なし
グローバル変数	gs_adc_end_flag A/D 変換状態フラグ
戻り値	ret 0U : 未完了 1U : 完了

## (39) R\_ADC\_EndFlagSet 関数

宣言	void R_ADC_EndFlagSet(void)
概要	A/D 変換終了フラグセット関数
引数	なし
グローバル変数	gs_adc_end_flag A/D 変換状態フラグ
戻り値	なし

## (40) R\_ADC\_EndFlagClear 関数

宣言	void R_ADC_EndFlagClear(void)
概要	AD 変換終了フラグクリア関数
引数	なし
グローバル変数	gs_adc_end_flag A/D 変換状態フラグ
戻り値	なし

## (41) R\_ADC\_MaskStart 関数

宣言	void R_ADC_MaskStart(void)
概要	A/D 変換開始(割り込み禁止)関数
引数	なし
グローバル変数	なし
戻り値	なし

## (42) R\_ADC\_ChangeSoftTrigger 関数

宣言	void R_ADC_ChangeSoftTrigger (void)
概要	A/D 変換トリガソフトウェア変更関数 ・ A/D 変換トリガをソフトウェア・トリガ・モード、連続変換モードに変更する
引数	なし
グローバル変数	なし
戻り値	なし

## (43) R\_ADC\_StartSetting 関数

宣言	void R_ADC_StartSetting (uint16_t * p_dma_buf, uint8_t dma_buf_size)
概要	A/D 変換動作開始設定関数 ・ A/D 変換を開始するための設定を行う
引数	p_dma_buf dma 転送先アドレス p_dma_buf_size dma 転送 Byte 数
グローバル変数	なし
戻り値	なし

## (44) R\_ADC\_EndSetting 関数

宣言	void R_ADC_EndSetting(void)
概要	A/D 変換動作終了設定関数 ・ A/D 変換の停止処理を行う
引数	なし
グローバル変数	なし
戻り値	なし

## (45) R\_ADC\_ChannelChange 関数

宣言	void R_ADC_ChannelChange (ad_channel_t adc_ch)
概要	A/D 変換チャネル変更関数 ・ A/D 変換を行うアナログ入力チャネル(ANLxx 端子)の変更を行う
引数	adc_ch ADS に設定する ch (ANLxx)
グローバル変数	なし
戻り値	なし

## 6.3 グローバル変数一覧

表 6-3 グローバル変数一覧

型	変数名	内容	使用関数
unsigned char *	gp_csi21_rx_address	csi21 receive buffer address	R_CSI21_Send_Receive r_csi21_interrupt
unsigned short	g_csi21_rx_length	csi21 receive data length	未使用
unsigned short	g_csi21_rx_count	csi21 receive data count	未使用
unsigned char *	gp_csi21_tx_address	csi21 send buffer address	R_CSI21_Send_Receive r_csi21_interrupt
unsigned short	g_csi21_send_length	csi21 send data length	R_CSI21_Send_Receive r_csi21_interrupt
unsigned short	g_csi21_tx_count	csi21 send data count	R_CSI21_Send_Receive r_csi21_interrupt
unsigned char	g_csi21_overrun_flag	csi21 overrun flag	r_csi21_callback_error R_SPI_SmartAnalogRead R_SPI_SmartAnalogWrite
struct	g_self_cal_11	D/A コンバータ出力 電圧測定時設定格納	R_SelfCalibrationInit
struct	g_self_cal_12	コンフィギュラブル・アンプ Ch1 出力 電圧測定時設定格納	R_SelfCalibrationInit
struct	g_self_cal_21	ゲイン誤差測定時設定格納	R_SelfCalibrationInit
struct *([3])	gp_self_cal_setting_data	自己補正用設定値の 格納変数のポインタ 配列	R_SelfCalibrationInit
struct [4]	g_ic300_conf_amp_default	Smart Analog IC 300 の単独チャネル使用 時のゲイン設定初期 値格納	R_SmartAnalogMeasureControllInit
struct	g_ic300_inst_amp_default	Smart Analog IC 300 の複数チャネル使用 時のゲイン設定初期 値格納	R_SmartAnalogMeasureControllInit
struct	g_amp_table	Smart Analog IC 300 ゲイン・オフセット 電圧格納テーブル	R_SmartAnalogMeasureControllInit R_SelfCalibAmp1DiffOffsetCalc R_SelfCalibAmp1DiffGainAverage



## 6.4 Smart Analog IC 300 のレジスタ値

本アプリケーションノートにおける Smart Analog IC 300 の SPI 制御レジスタ設定について説明します。なお、本アプリケーションノートで使用していない SPI 制御レジスタについては説明を省略します（初期値の設定での使用となります）。

**注意** SPI 制御レジスタ設定方法の詳細につきましては、RAA730300 モノリシック・プログラマブル・アナログ IC のデータシートをご参照ください。

### (1) コンフィギュレーション・レジスタ 1 (CONFIG1)

コンフィギュラブル・アンプ Ch1、Ch2 の各スイッチの ON/OFF を設定します。

アドレス : 00H	リセット時 : 88H				R/W		設定値 : *0H	
(略号)	7	6	5	4	3	2	1	0
CONFIG1	SW10	SW11	SW12	SW13	SW20	SW21	SW22	SW23
(設定値)	1	0	0	*	0	0	0	0

**備考** \*: 測定状態（ゲイン誤差測定、入力換算オフセット電圧測定）により値が異なります。詳細は、表 5-1 または表 5-2 でご確認ください。

### (2) コンフィギュレーション・レジスタ 2 (CONFIG2)

コンフィギュラブル・アンプ Ch1～Ch3 の各スイッチの ON/OFF を設定します。

アドレス : 01H	リセット時 : 80H				R/W		設定値 : 8*H	
(略号)	7	6	5	4	3	2	1	0
CONFIG2	SW30	SW31	SW32	SW33	0	SW02	SW01	SW00
(設定値)	1	0	0	0	0	0	*	*

**備考** \*: 測定状態（ゲイン誤差測定、入力換算オフセット電圧測定）により値が異なります。詳細は、表 5-1 または表 5-2 でご確認ください。

### (3) MPX 設定レジスタ 1 (MPX1)

コンフィギュラブル・アンプ Ch1、Ch2 の入力信号を設定します。

アドレス : 03H	リセット時 : 00H				R/W		設定値 : **H	
(略号)	7	6	5	4	3	2	1	0
MPX1	MPX11	MPX10	MPX21	MPX20	MPX31	MPX30	MPX41	MPX40
(設定値)	*	*	*	0	*	0	0	0

**備考** \*: 測定状態（ゲイン誤差測定、入力換算オフセット電圧測定）により値が異なります。詳細は、表 5-1 または表 5-2 でご確認ください。

## (4) ゲイン制御レジスタ 1 (GC1)

コンフィギュラブル・アンプ Ch1 のゲインを設定します。

アドレス : 06H	リセット時 : 00H			R/W		設定値 : **H	
(略号)	7	6	5	4	3	2	1 0
GC1	0	0	0	AMPG14	AMPG13	AMPG12	AMPG11 AMPG10
(設定値)	0	0	0	*	0	0	0 *

備考 \*: 測定状態 (ゲイン誤差測定、入力換算オフセット電圧測定) により値が異なります。詳細は、表 5-1 または表 5-2 でご確認ください。

## (5) パワー制御レジスタ 1 (PC1)

コンフィギュラブル・アンプ、D/A コンバータの動作許可/停止を設定します。

アドレス : 11H	リセット時 : 00H			R/W		設定値 : 1*H	
(略号)	7	6	5	4	3	2	1 0
PC1	DAC4OF	DAC3OF	DAC2OF	DAC1OF	AMP4OF	AMP3OF	AMP2OF AMP1OF
(設定値)	0	0	0	1	0	0	0 *

備考 \*: 測定状態 (ゲイン誤差測定、入力換算オフセット電圧測定) により値が異なります。詳細は、表 5-1 または表 5-2 でご確認ください。

## (6) パワー制御レジスタ 2 (PC2)

D/A コンバータの動作許可/停止を設定します。

アドレス : 12H	リセット時 : 00H			R/W		設定値 : *0H	
(略号)	7	6	5	4	3	2	1 0
PC2	DAC7OF	DAC6OF	DAC5OF	AMP5OF	LPFOF	HPFOF	LDOOF TEMPOF
(設定値)	0	*	*	0	0	0	0 0

備考 \*: 測定状態 (ゲイン誤差測定、入力換算オフセット電圧測定) により値が異なります。詳細は、表 5-1 または表 5-2 でご確認ください。

## 6.5 コード生成ツールの設定一覧

コード生成ツールで設定すべき箇所について示します。なお、使用していない機能については、省略しています。

表 6-4 コード生成ツール設定一覧 (1)

モジュール	マクロ	サブ	設定	状態
クロック発生回路				使用する
	CGC			使用する
			端子割り当て設定-PIOR0 ビット=1	使用しない
			端子割り当て設定-PIOR1 ビット=1	使用しない
			動作モード設定	高速メイン・モード $2.7(V) \leq VDD \leq 3.6(V)$
			EVDD 設定	$2.7(V) \leq EVDD \leq 3.6(V)$
			メイン・システム・クロック (fMAIN)設定	高速オンチップ オシレータクロック (fIH)
			fIH 動作	使用する
			fIH 周波数	32(MHz)
			fMX 動作	使用しない
			fSUB 動作	使用する
			サブシステム・クロック (fSUB)設定	XT1 発振(fXT)
			fSUB 周波数	32.768(kHz)
			XT1 発振回路の発振モード選択	低消費発振
			STOP,HALT モード時のクロック供給設定	供給許可
			fIL 周波数	15(kHz)
			RTC,インターバル・タイマ動作クロック	32.768 (fSUB)(kHz)
			CPU と周辺クロック (fCLK)	32000 (fIH)(kHz)
			オンチップ・デバッグ動作設定	使用しない
			セキュリティ ID 設定	使用する
			セキュリティ ID	0x000000000000000000000000
			リセット要因を確認する関数を出力する	使用しない
			フラッシュ・メモリ CRC 演算機能設定	使用しない
			不正メモリ・アクセス検出機能設定	使用しない
			RAM ガード機能設定	使用しない
			ポート・レジスタのガード機能設定	使用しない
			割り込みレジスタのガード機能設定	使用しない
			チップ・ステート・コントロール・レジスタのガード機能設定	使用しない

表 6-5 コード生成ツール設定一覧 (2)

モジュール	マクロ	サブ	設定	状態
シリアル				使用する
	SAU1			使用する
		Channel1		
			チャンネル 1	CSI21(送信/受信機能)
			転送モード設定	連続転送モード
			データ長設定	8 ビット
			データ転送方向設定	MSB
			データ送受信タイミング設定	タイプ 1
			クロック・モード	内部クロック(マスタ)
			ボー・レート	500000(bps)(実際の値 : 500000)
			通信完了割り込み優先順位 (INTCSI21)	低
			送信完了 (コールバック機能設定)	使用しない
			受信完了 (コールバック機能設定)	使用する
			オーバラン・エラー (コールバック機能設定)	使用する
A/D コンバータ				使用する
	ADC			使用する
			A/D コンバータ動作設定	使用する
			コンパレータ動作設定	許可
			分解能設定	12 ビット
			VREF(+)設定	AVDD
			VREF(-)設定	AVSS
			トリガ・モード設定	ハードウェア・トリガ・ノーウエイト・モード
			ハードウェア・トリガ・ノーウエイト・モード	INTTM01(INTTM01 を設定してください)
			動作モード設定	連続セレクト・モード
			ANI0 - ANI12 アナログ入力端子設定	ANI0 - ANI12
			ANI16 - ANI30 アナログ入力端子	ANI16, ANI17, ANI19
			変換開始チャンネル設定	ANI2
			基準電圧	$2.7 \leq AVDD \leq 3.6$ (V)
			変換時間モード	標準 1
			変換時間	3.375 (108/fCLK)( $\mu$ s)
			変換結果上限/下限値設定	ADLL $\leq$ ADCRH $\leq$ ADUL で割り込み要求信号(INTAD)を発生
			上限値(ADUL)	255
			下限値(ADLL)	0
			A/D の割り込み許可(INTAD)	使用する
			優先順位	低

表 6-6 コード生成ツール設定一覧 (3)

モジュール	マクロ	サブ	設定	状態
タイマ				使用する
	TAU0			使用する
		Channel5		
			チャンネル 5	インターバル・タイマ
			インターバル時間(16 ビット)	100μs, (実際の値 : 100)
			カウント開始時に割り込みを発生する	使用しない
			タイマ・チャンネル 5 のカウント完了で割り込み発生(INTTM05)	使用する
			優先順位 (INTTM05)	低
リアルタイム・クロック				使用する
	RTC			使用する
			リアルタイムクロック動作設定	使用する
			時間制の選択	24 時間制
			リアルタイムクロック初期値設定	使用しない
			RTC1HZ 端子の出力(1 Hz)許可	使用しない
			アラーム検出機能	使用しない
			補正機能	使用しない
			定周期割り込み機能(INTRTC)	使用する, 1 秒に 1 度 (秒カウントアップと同時)
			優先順位(INTRTC)	低
DMA コントローラ				使用する
	DMA0			使用しない
	DMA1			使用する
			DMA 動作設定	使用する
			転送方向設定	SFR → 内蔵 RAM
			転送データ・サイズ設定	16 ビット
			SFR アドレス	ADCR - 0x000fff1e
			RAM アドレス	0xfef00
			転送回数	64
			トリガ信号	INTAD
			DMA1 送信終了割り込み (INTDMA1)	使用する
			優先順位	低

## 6.6 フローチャート

図 6-1に本アプリケーションノートにおける全体フローチャートを示します。

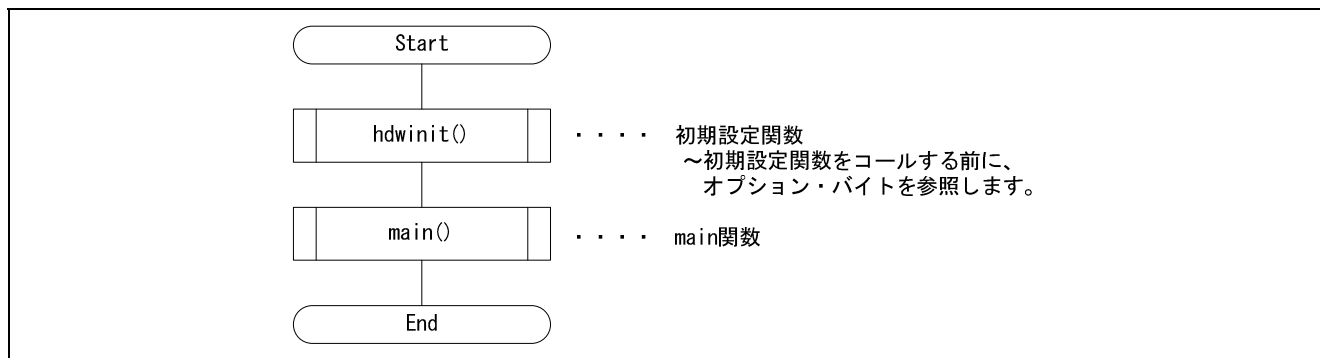


図 6-1 全体フローチャート

### (1) システム初期化 (hwinit) 関数

図 6-2にシステム初期化 (hwinit) 関数を示します。

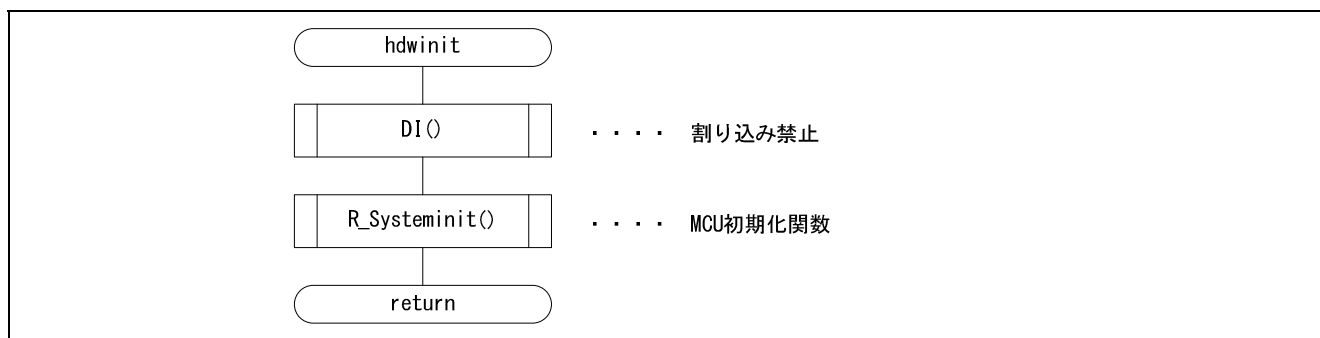


図 6-2 hwinit 関数

## (2) MCU 初期化 (R\_Systeminit) 関数

図 6-3に MCU 初期化 (R\_Systeminit) 関数を示します。

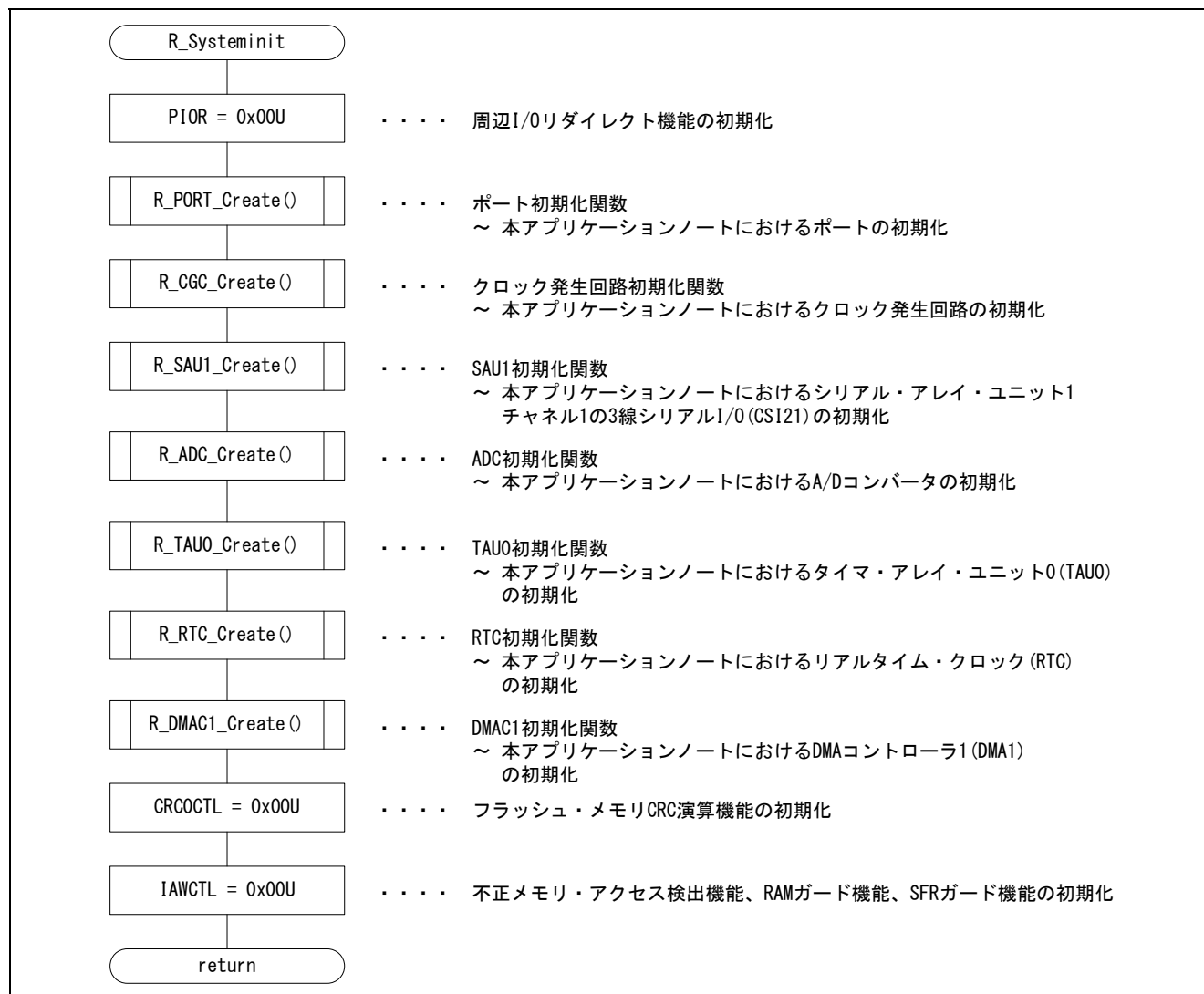


図 6-3 R\_Systeminit 関数

## (3) メイン (main) 関数

図 6-4にメイン (main) 関数を示します。

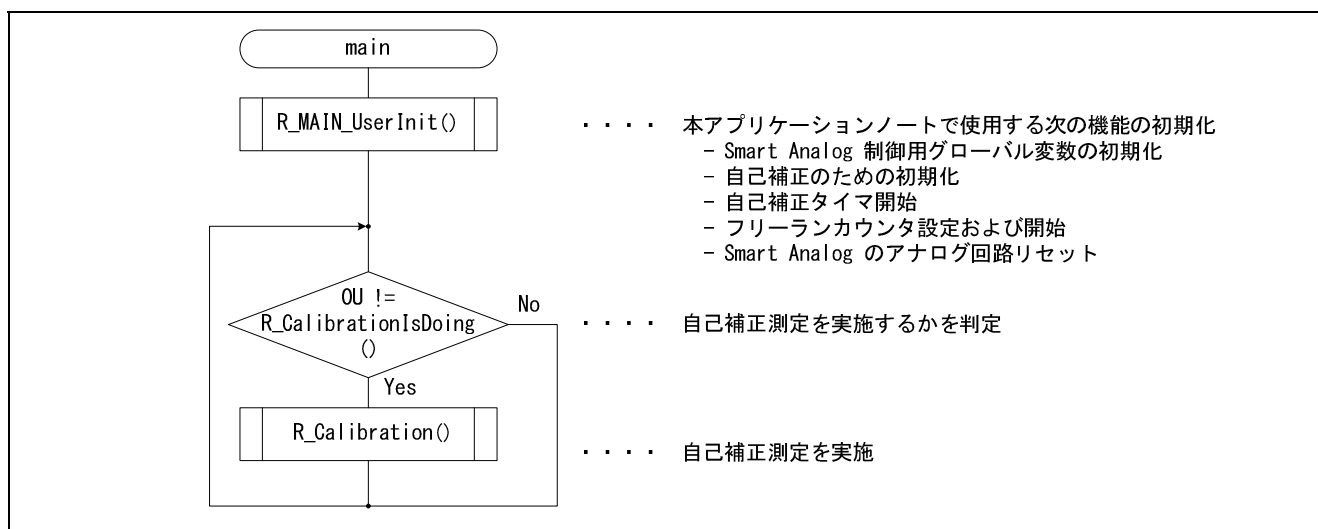


図 6-4 main 関数

## (4) ユーザーインターフェース初期化関数

図 6-5にユーザーインターフェース初期化関数を示します。

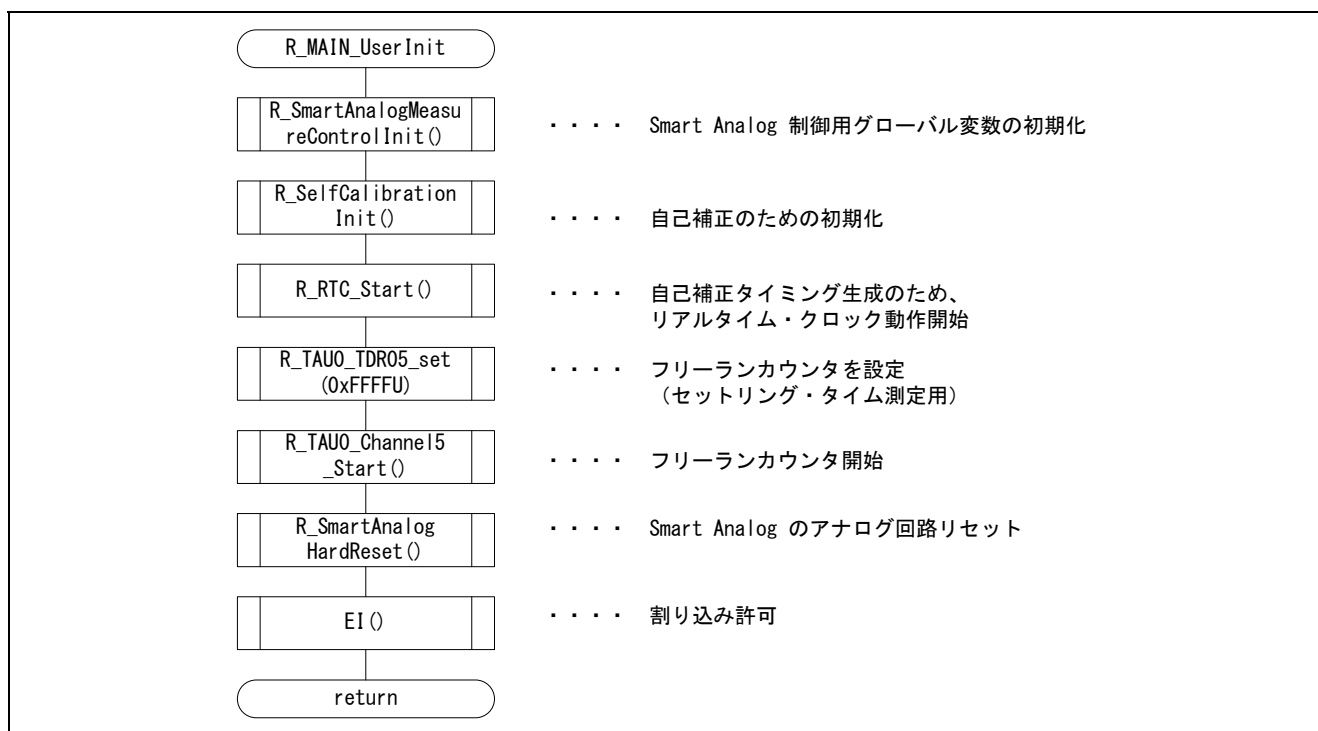


図 6-5 ユーザーインターフェース初期化関数



## (5) キャリブレーショントリガ判定関数

図 6-6にキャリブレーショントリガ判定関数を示します。

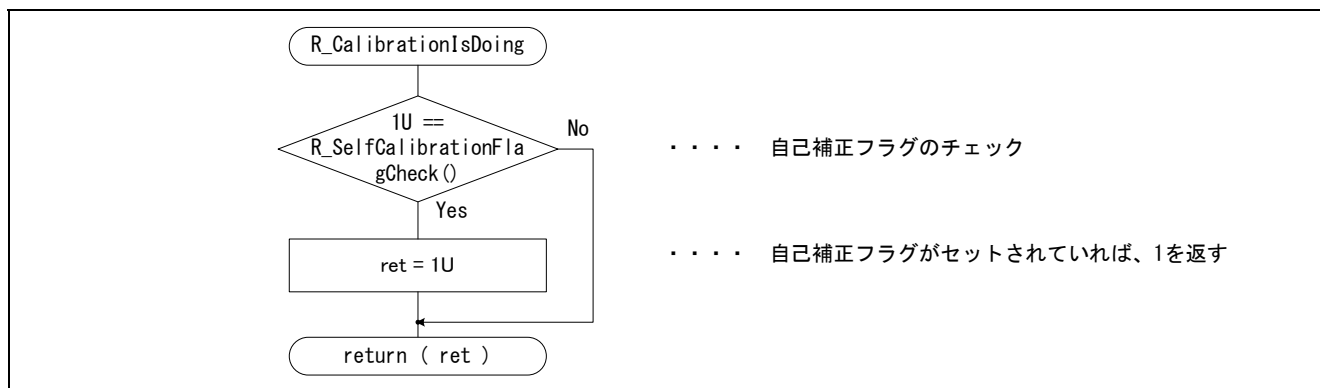


図 6-6 キャリブレーショントリガ判定関数

## (6) 自己補正開始トリガフラグチェック関数

図 6-7に、自己補正開始トリガフラグチェック関数を示します。

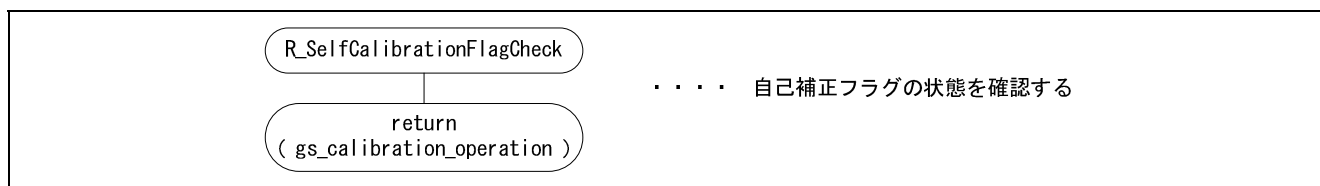


図 6-7 自己補正開始トリガフラグチェック関数

## (7) 自己補正タイミング生成関数

図 6-8に、自己補正タイミング生成関数を示します。

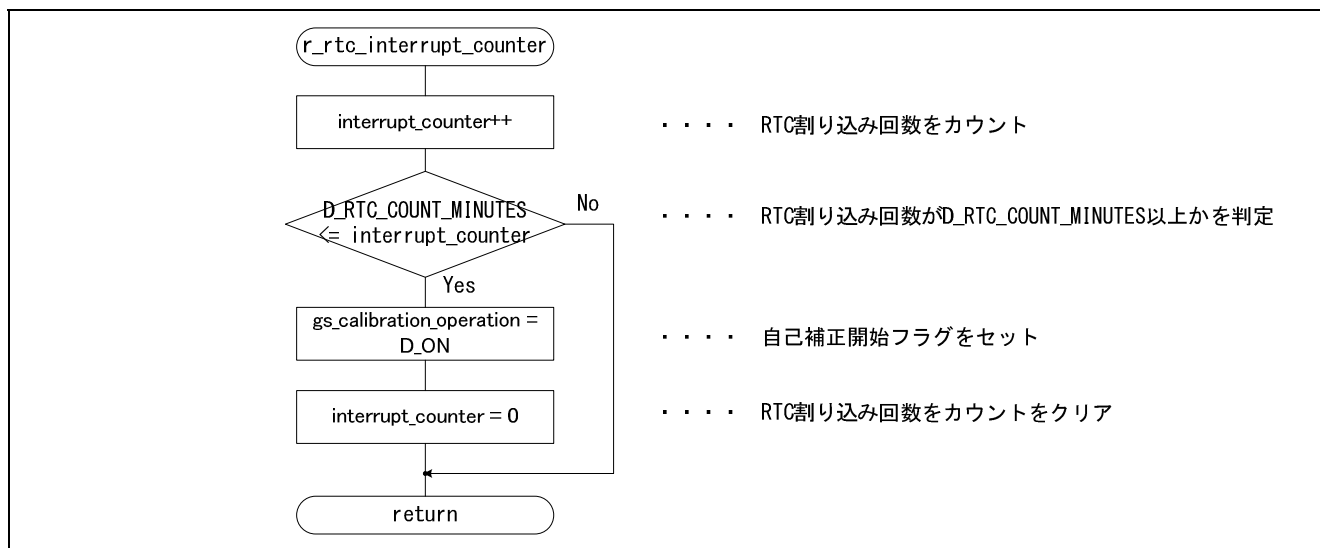


図6-8 自己補正タイミング生成関数

## (8) RTC 割り込み関数

図 6-9に、RTC 割り込み関数を示します。

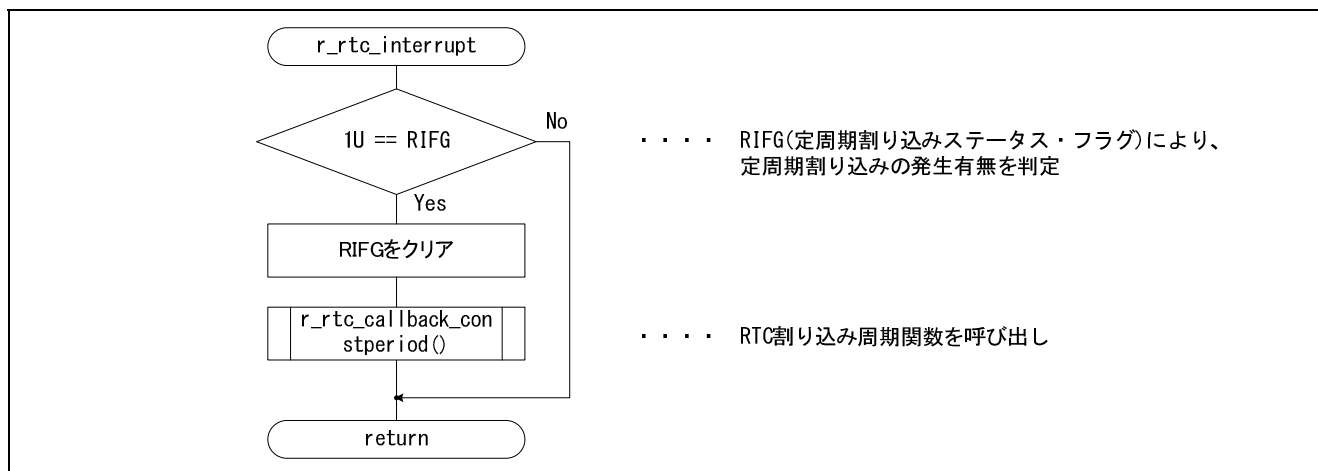


図6-9 RTC 割り込み関数

## (9) RTC 割り込みハンドラー関数

図 6-10に、RTC 割り込みハンドラー関数を示します。

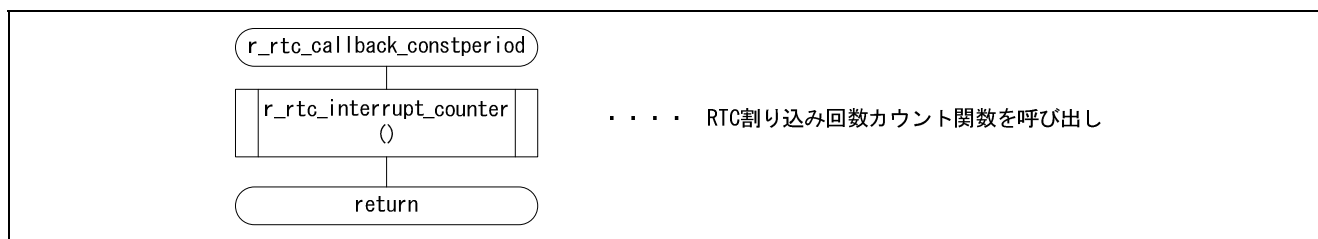


図 6-10 RTC 割り込みハンドラー関数

## (10) 自己補正開始トリガフラグクリア関数

図 6-11に、自己補正開始トリガフラグクリア関数を示します。

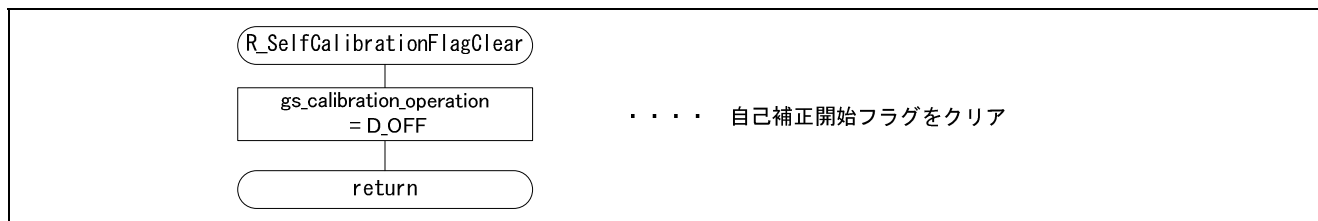


図 6-11 自己補正開始トリガフラグクリア関数

## (11) キャリブレーション制御関数

図 6-12に、キャリブレーション制御関数を示します。なお、calibration\_state 判定はステートマシンになっています。各ステートの詳細については、図 6-13の状態遷移図をご参照ください。

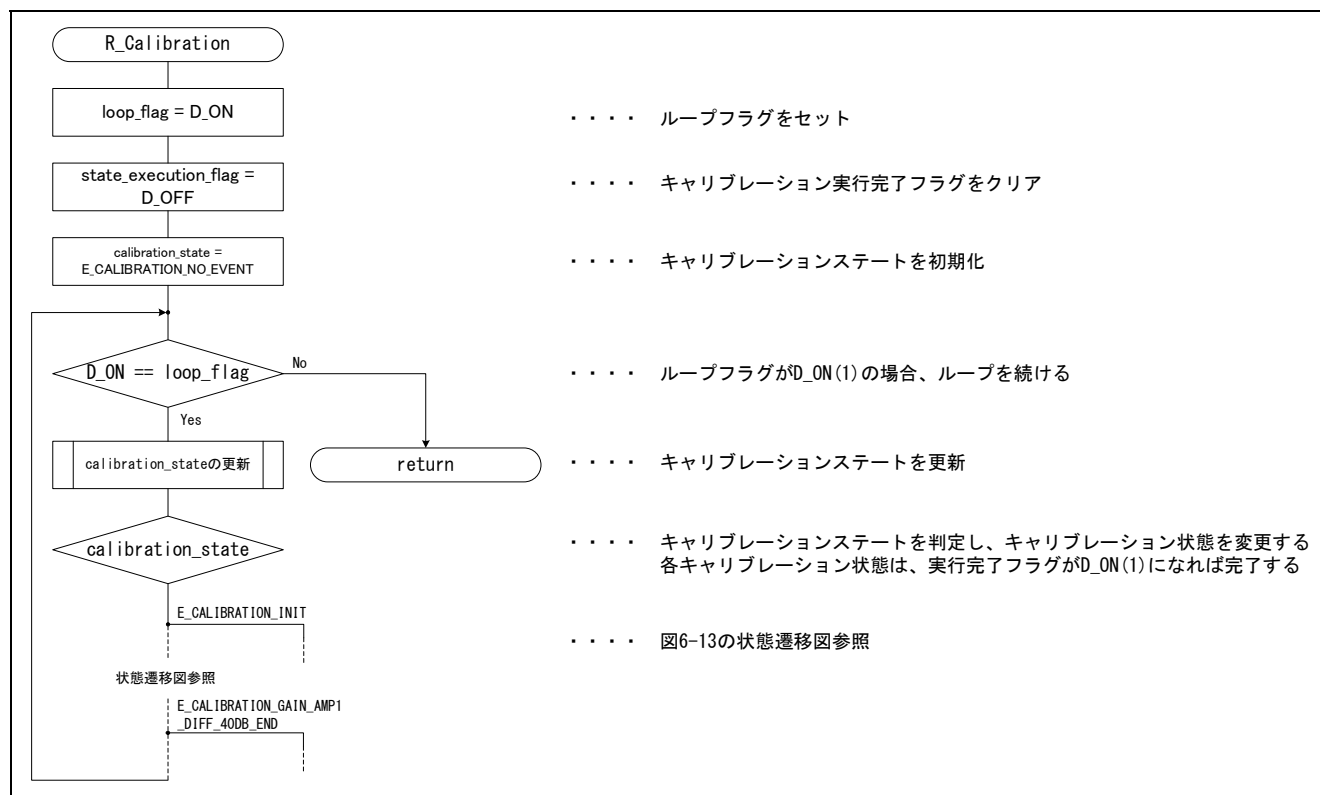


図 6-12 キャリブレーション制御関数

本アプリケーションノートで示す、自己補正の状態遷移について、以下に図示します。自己補正では、合わせて 10 個のプロセスを実施しています。そのため、本アプリケーションノートでは、10 個のプロセスに初期処理および終了処理を加え、12 個のステートに分けて実施しています。

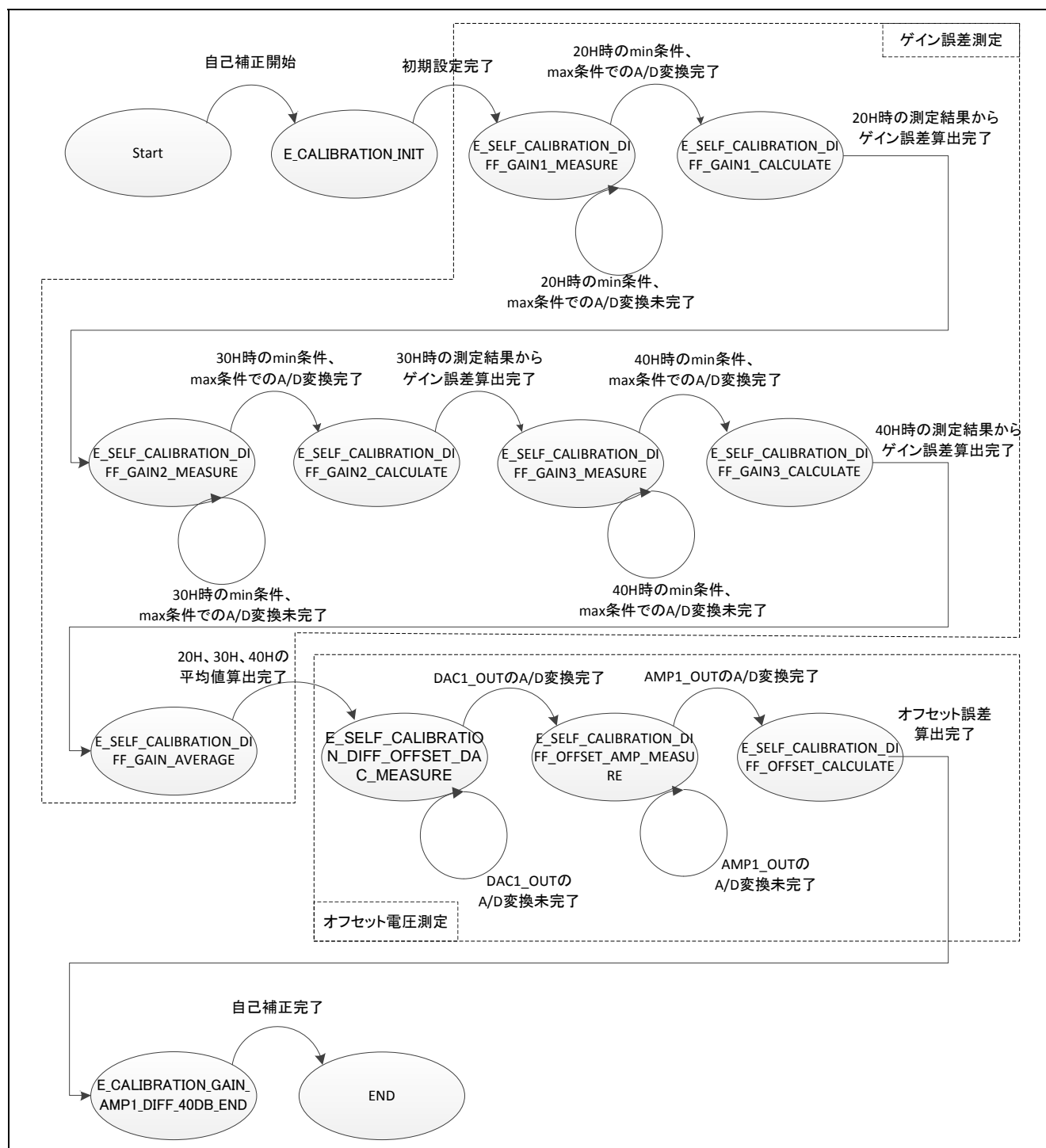


図 6-13 自己補正の状態遷移図

表 6-7に、alibration\_state 判定ステートと呼び出す関数一覧について示します。

表 6-7 ステートと関数対応表

ステート名	概要	呼び出し関数
E_CALIBRATION_INIT	自己補正の初期化	calibration_init 関数
E_SELF_CALIBRATION_DIFF_GAIN1_MEASURE	ゲイン誤差測定 1 回目	R_SelfCalibAmp1DiffGainMeasure 関数
E_SELF_CALIBRATION_DIFF_GAIN1_CALCULATE	ゲイン誤差算出 1 回目	R_SelfCalibAmp1DiffGainCalc 関数
E_SELF_CALIBRATION_DIFF_GAIN2_MEASURE	ゲイン誤差測定 2 回目	R_SelfCalibAmp1DiffGainMeasure 関数
E_SELF_CALIBRATION_DIFF_GAIN2_CALCULATE	ゲイン誤差算出 2 回目	R_SelfCalibAmp1DiffGainCalc 関数
E_SELF_CALIBRATION_DIFF_GAIN3_MEASURE	ゲイン誤差測定 3 回目	R_SelfCalibAmp1DiffGainMeasure 関数
E_SELF_CALIBRATION_DIFF_GAIN3_CALCULATE	ゲイン誤差算出 3 回目	R_SelfCalibAmp1DiffGainCalc 関数
E_SELF_CALIBRATION_DIFF_GAIN_AVERAGE	ゲイン誤差の平均値算出	R_SelfCalibAmp1DiffGainAverage 関数
E_SELF_CALIBRATION_DIFF_OFFSET_DAC_MEASURE	オフセット電圧測定 DAC_OUT 電圧	R_SelfCalibAmp1DiffOffsetDac 関数
E_SELF_CALIBRATION_DIFF_OFFSET_AMP_MEASURE	オフセット電圧測定 AMP_OUT 電圧	R_SelfCalibAmp1DiffOffsetAmp 関数
E_SELF_CALIBRATION_DIFF_OFFSET_CALCULATE	オフセット電圧算出	R_SelfCalibAmp1DiffOffsetCalc 関数
E_CALIBRATION_GAIN_AMP1_DIFF_40DB_END	自己補正の終了処理	R_SelfCalibrationFlagClear 関数

## (12) キャリブレーション初期化関数

図 6-14に、キャリブレーション初期化関数を示します。

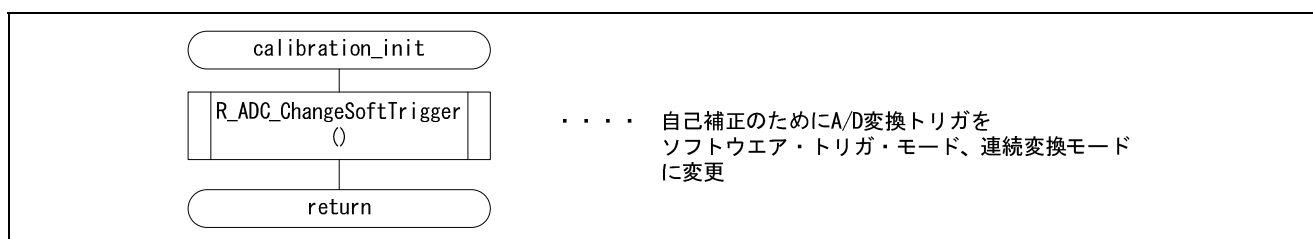


図 6-14 キャリブレーション初期化関数

## (13) ゲイン測定関数

図 6-15、図 6-16に、ゲイン測定関数を示します。

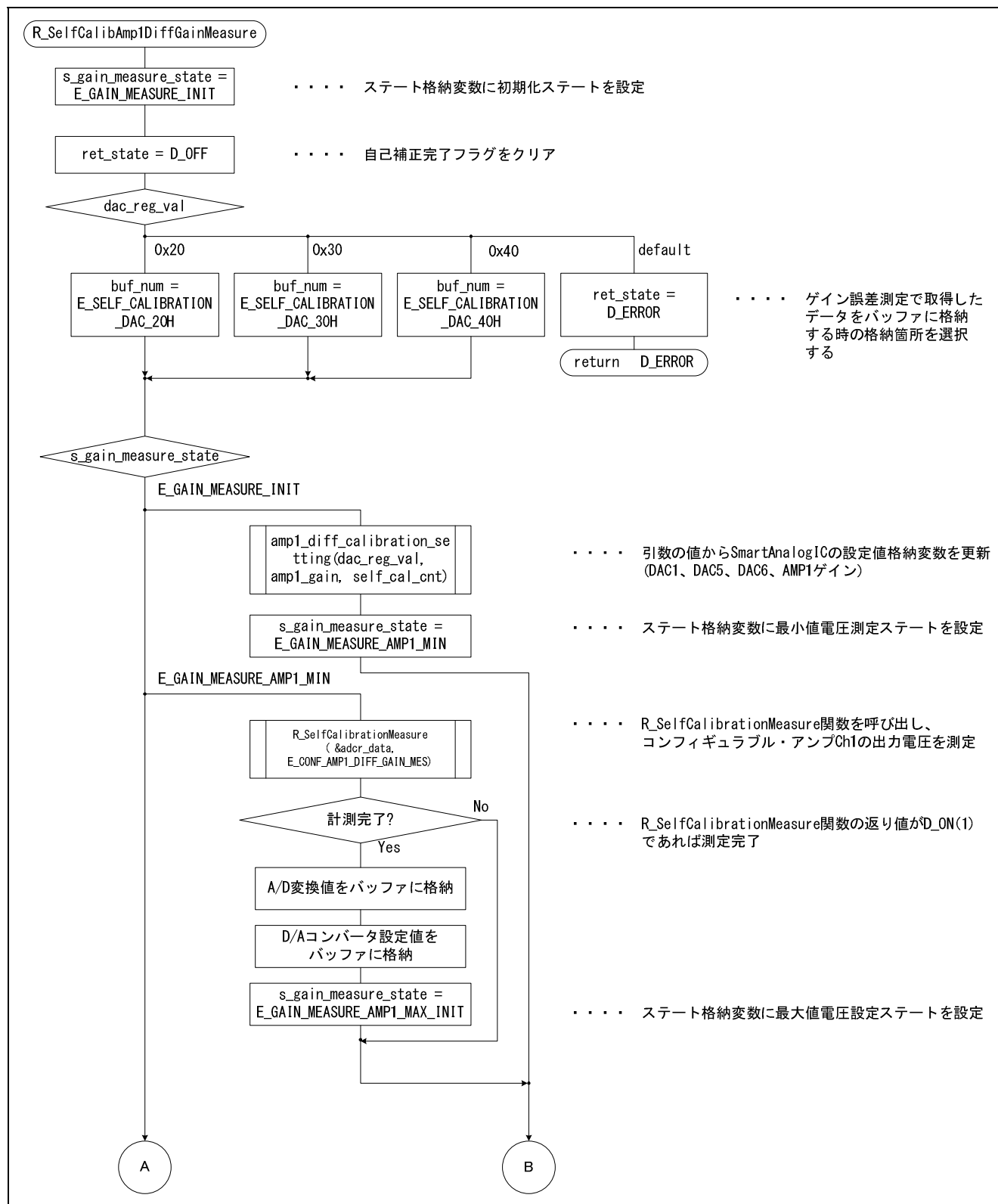


図 6-15 ゲイン測定関数(1)

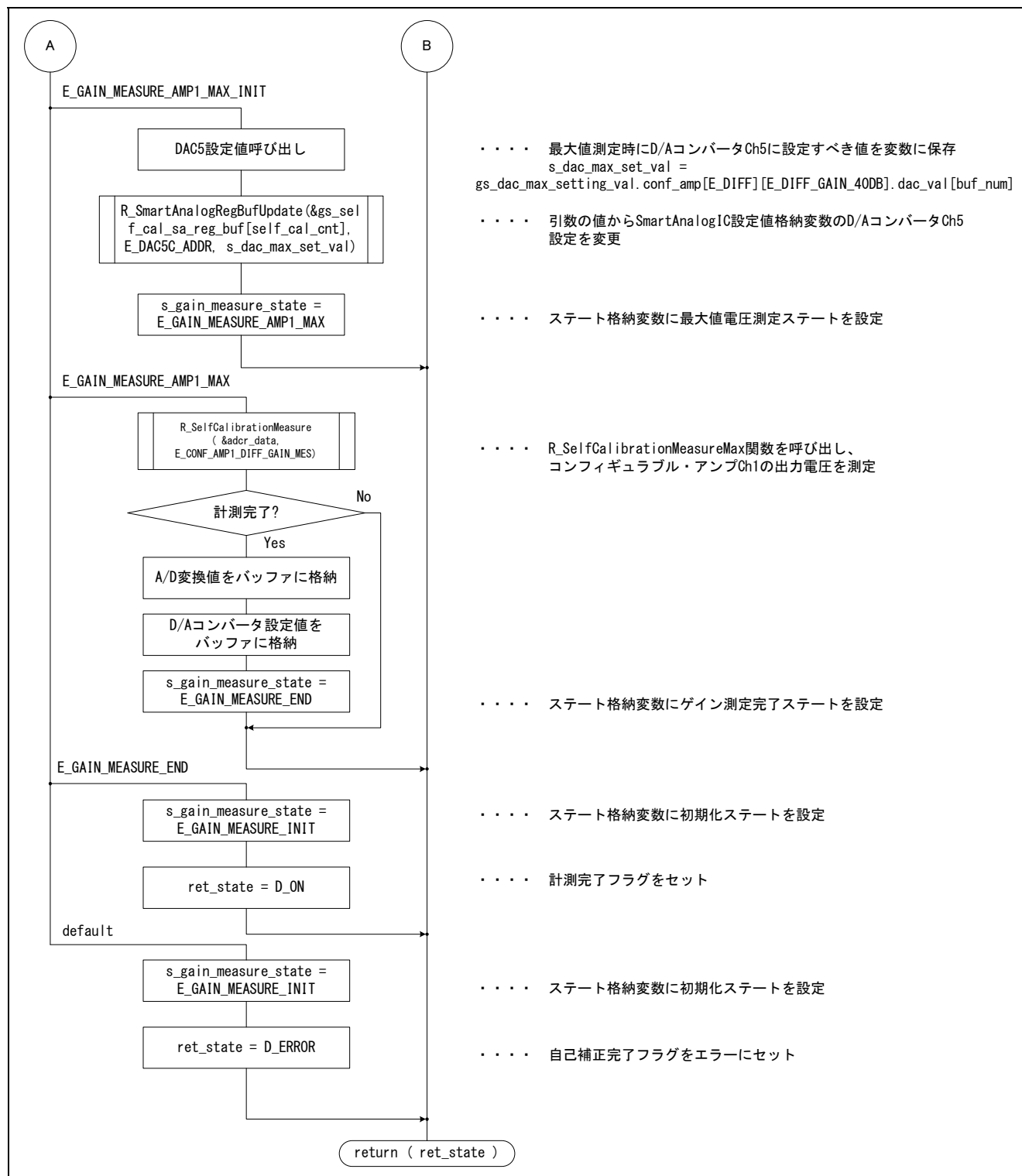


図 6-16 ゲイン測定関数(2)

## (14) 自己補正用 A/D 変換制御関数

図 6-17に、自己補正用 A/D 変換制御関数を示します。

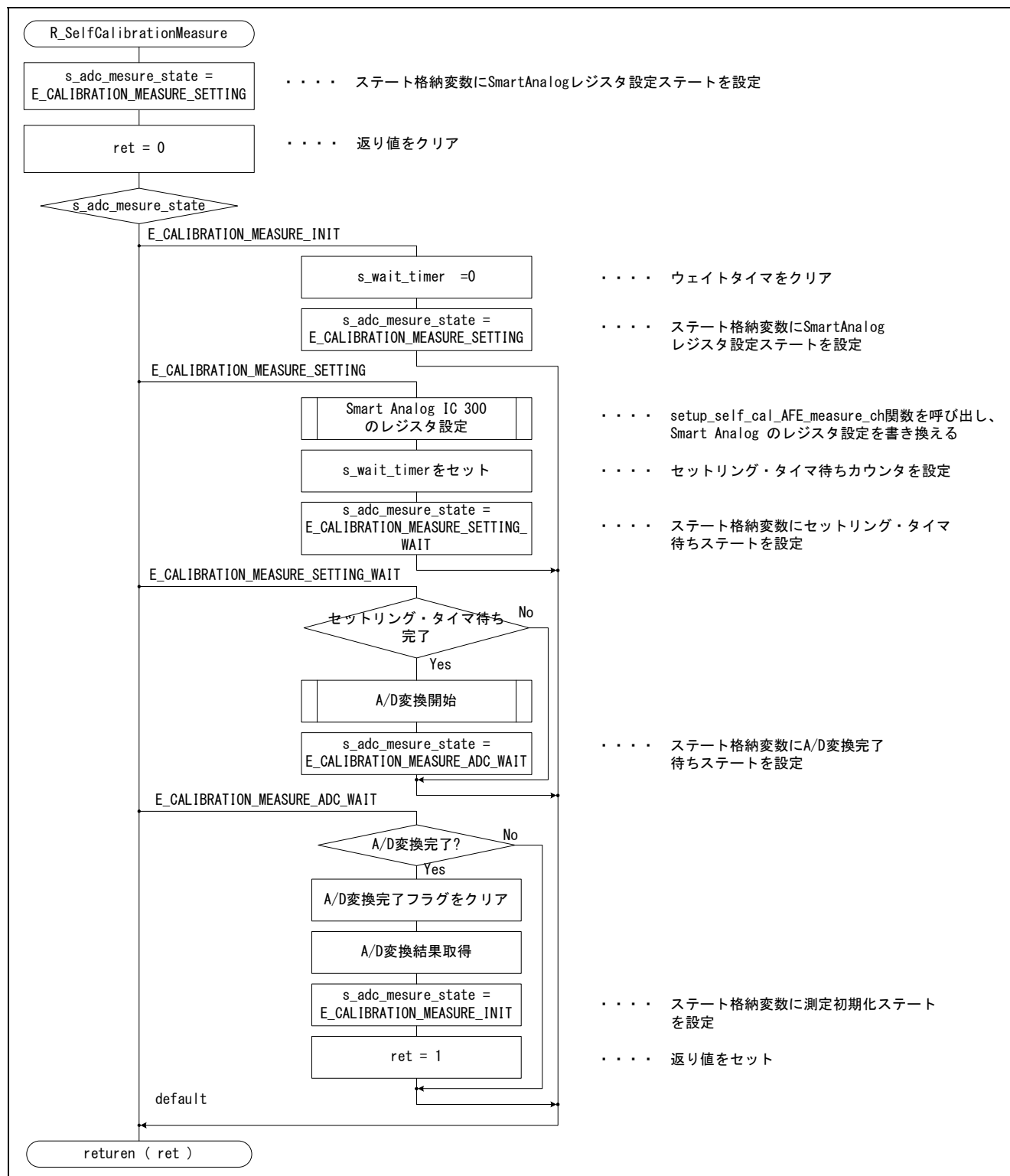


図 6-17 自己補正用 A/D 変換制御関数



## (15) ゲイン算出関数

図 6-18に、ゲイン算出関数を示します。

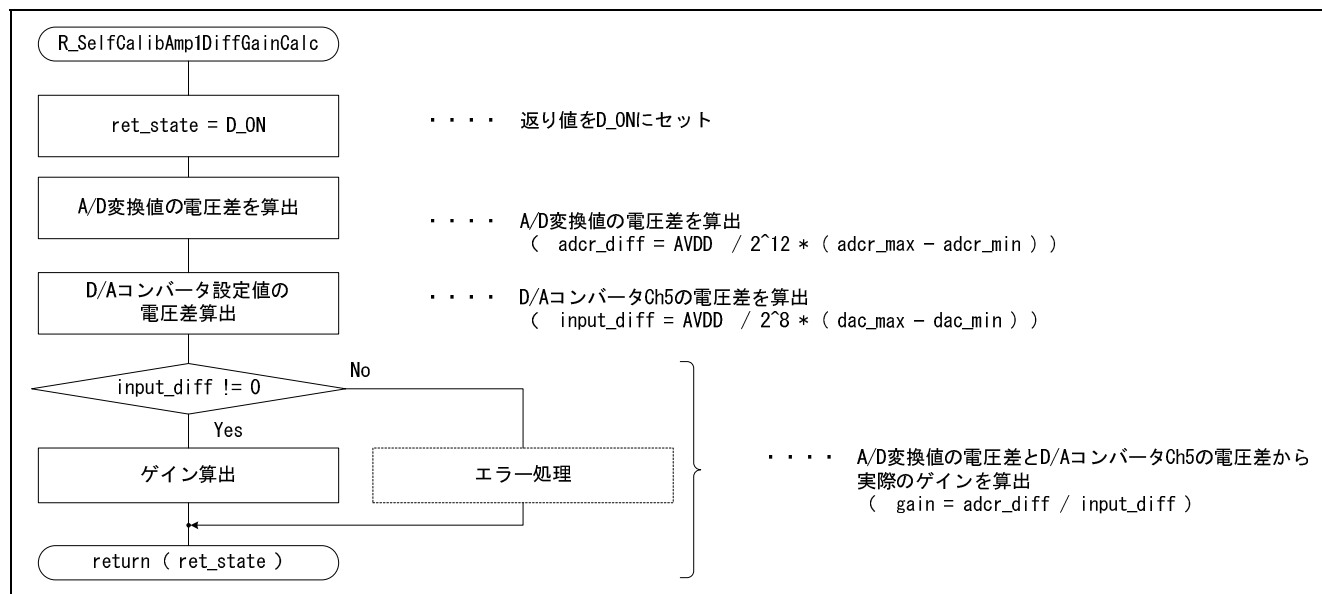


図 6-18 ゲイン算出関数

## (16) ゲイン平均化関数

図 6-19に、ゲイン平均化関数を示します。

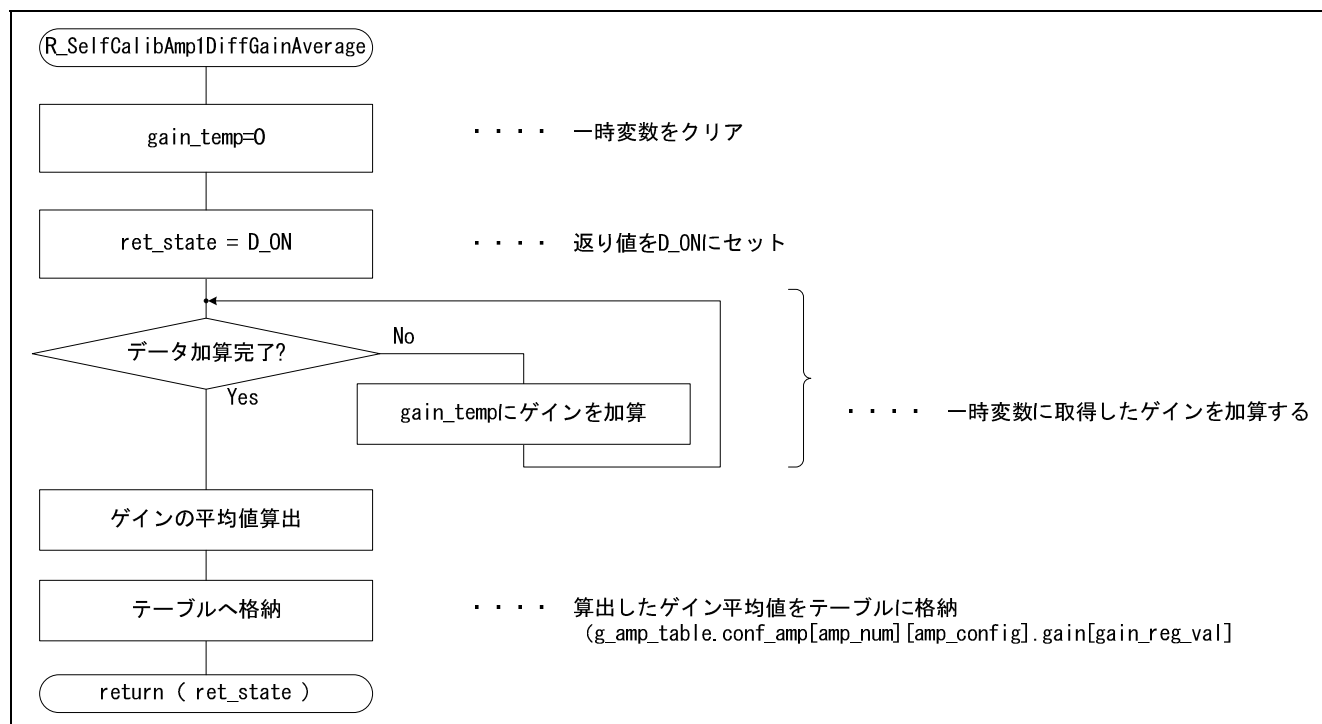


図 6-19 ゲイン平均化関数

## (17) DAC1\_OUT 測定関数

図 6-20に、DAC1\_OUT 測定関数を示します。

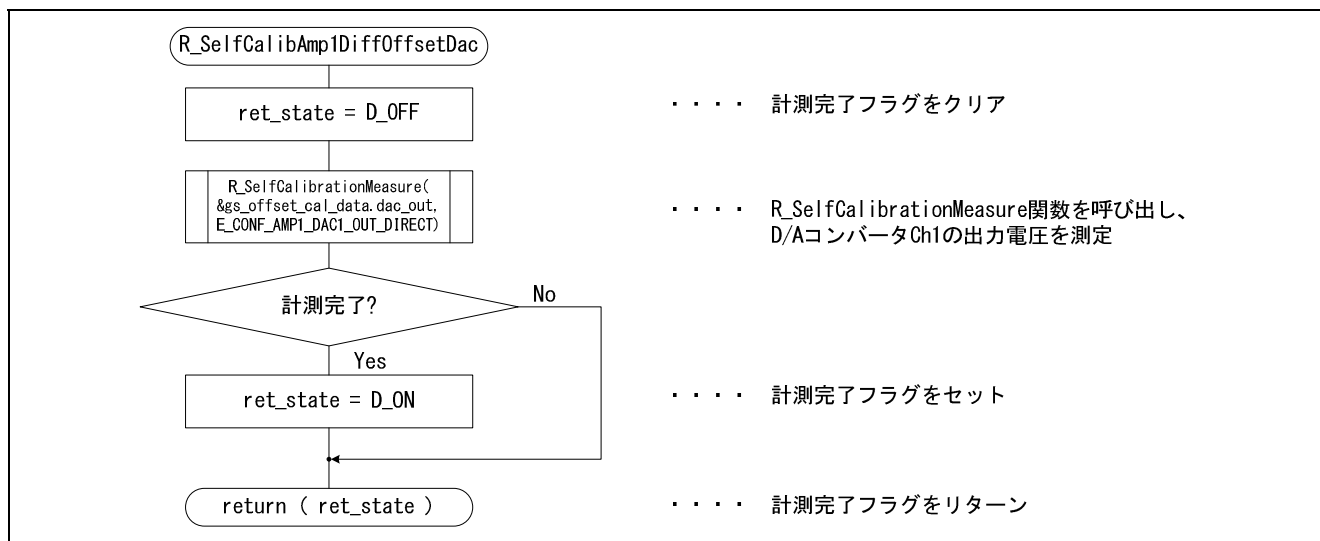


図 6-20 DAC1\_OUT 測定関数

## (18) AMP1\_OUT 測定関数

図 6-21に、AMP1\_OUT 測定関数を示します。

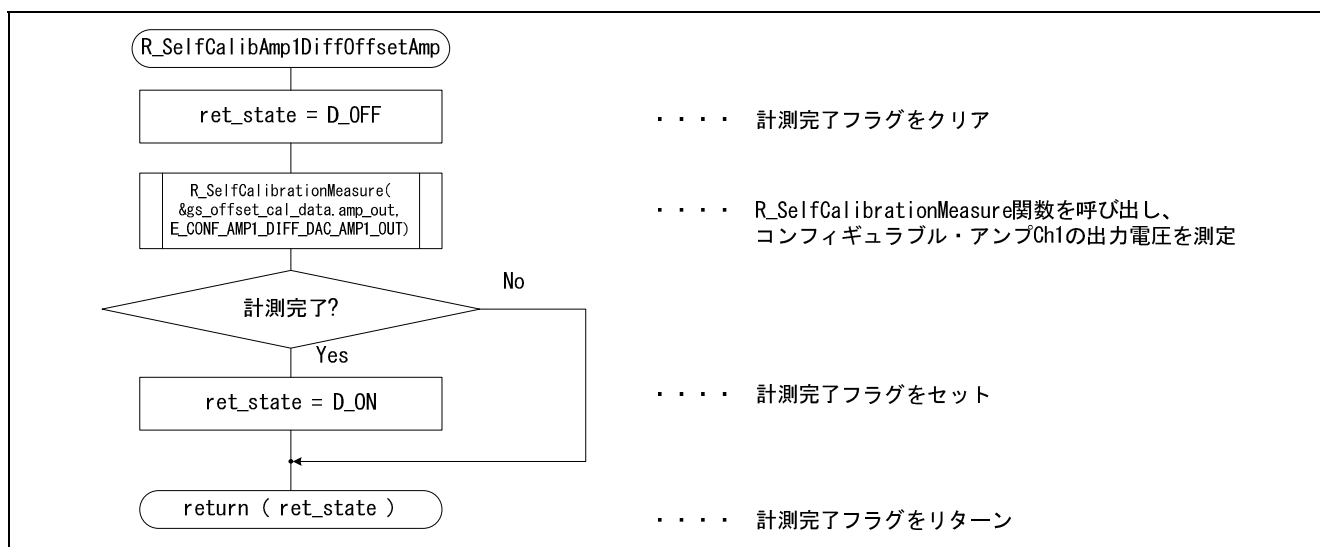


図 6-21 AMP1\_OUT 測定関数

## (19) 入力換算オフセット電圧算出関数

図 6-22に、入力換算オフセット電圧算出関数を示します。

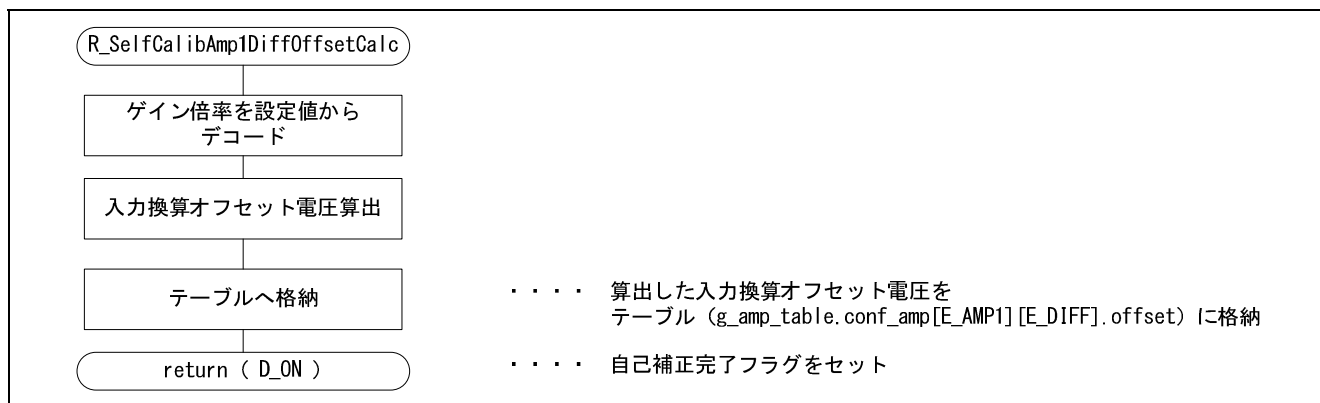


図 6-22 入力換算オフセット電圧算出関数

## 7. サンプルコードの組み込み手順

本アプリケーションノートのサンプルコードとして、2つのサンプルプロジェクトを用意しています。

「error\_measurement」プロジェクトは、誤差測定のみを実施する基本的なサンプルコードです。

「example\_self-calibration」プロジェクトは、誤差測定で得られたゲインとオフセット電圧を用いて、実際に補正を実施するサンプルコードです。「example\_self-calibration」プロジェクトでは、コンフィギュラブル・アンプの出力電圧から、0.1秒ごとの周期にて、入力差電圧を算出しています。なお、どちらのプロジェクトでも、コンフィギュラブル・アンプ Ch1 を、40 dB の差動アンプ構成にて用いております。

本章では、サンプルコードにおける自己補正データの取り出し方や、コンフィギュラブル・アンプのチャネルおよびアンプ構成を変更する方法について説明します。

### (1) 自己補正データの取り出し方

本サンプルコードで測定した自己補正データの取り出し方について説明します。

自己補正により得られた測定データは、Smart Analog IC 300 ゲイン・オフセット電圧格納テーブル（g\_amp\_table 変数）に格納されています。なお、g\_amp\_table 変数は ic300\_amp\_table\_t 型構造体を使用しており、ic300\_amp\_table\_t 型構造体は、下記の構成となっています。

```
typedef struct
{
    ic300_single_amp_t conf_amp[E_AMP_NUM_MAX][D_IC300_SINGLE_AMP_CONFIG_MAX];
    ic300_multi_amp_t inst_amp;
}ic300_amp_table_t ;
```

表 7-1 ic300\_amp\_table\_t 構造体のメンバ

メンバ名	説明
conf_amp	ic300_single_amp_t 型構造体変数 コンフィギュラブル・アンプを単独チャネルで動作させた場合のゲイン、入力換算オフセット電圧の情報を保存します。 2次元配列の第一配列 コンフィギュラブル・アンプのチャネル番号 0：コンフィギュラブル・アンプ Ch1 1：コンフィギュラブル・アンプ Ch2 2：コンフィギュラブル・アンプ Ch3 2次元配列の第二配列 コンフィギュラブル・アンプのアンプ構成 0：非反転アンプ構成 1：反転アンプ構成 2：差動アンプ構成
inst_amp	ic300_multi_amp_t 型構造体変数 コンフィギュラブル・アンプを計装アンプ構成で使用させた場合のゲイン、入力換算オフセット電圧の情報を保存します。

また、ic300\_single\_amp\_t 型および ic300\_multi\_amp\_t 型は下記構造体となっています。

```
typedef struct
{
    float gain[D_IC300_SINGLE_AMP_REG_MAX];
    float offset;
}ic300_single_amp_t ;
```

表 7-2 ic300\_single\_amp\_t 構造体メンバ

メンバ名	説明
gain	コンフィギュラブル・アンプのゲインの情報（単位は倍）を保存します。 配列の要素はレジスタ値を指しています。 なお、初期値はデータシートの値を倍数表現に変換した値が格納されます。
offset	コンフィギュラブル・アンプのオフセット電圧（単位は V）の情報を保存します。 なお、初期値は 0V が格納されます。

```
typedef struct
{
    float gain[D_IC300_INST_AMP_REG_MAX];
    float offset;
}ic300_multi_amp_t ;
```

表 7-3 ic300\_multi\_amp\_t 構造体メンバ

メンバ名	説明
gain	計装アンプ構成のゲインの情報（単位は倍）を保存します。 配列の要素はレジスタ値を指しています。 なお、初期値はデータシートの値を倍数表現に変換した値が格納されます。
offset	計装アンプ構成のオフセット電圧（単位は V）の情報を保存します。 なお、初期値は 0V が格納されます。

上記より、本サンプルコードで計測した自己補正の測定データ（コンフィギュラブル・アンプ Ch1 の 40dB ゲインの差動アンプ構成）はそれぞれ下記に格納されています。

- ・ ゲイン：g\_amp\_table.conf\_amp[0][2].gain[17]
- ・ オフセット電圧：g\_amp\_table.conf\_amp[0][2].offset

## (2) 自己補正データの使用例

測定した自己補正データの使用例について説明します。自己補正で測定したデータはユーザーが実際に測定するデータの補正データとして使用します。ここでは、「example」プロジェクトで使用している方法について説明します。

「example」プロジェクトでは、自己補正処理以外に 0.1 秒ごとに差動信号の周期測定を実施しています。ここでは、差動信号とは差動出力型のセンサを想定し、Smart Analog IC 300 の MPXIN10 端子と MPXIN20 端子間に印加される信号のことを指します。この差動入力信号をコンフィギュラブル・アンプ Ch1 で増幅し、A/D コンバータでデジタル値に変換、その後演算により差動電圧を算出します。自己補正データは差動電圧の演算処理に使用します。

次に、本サンプルコードで演算処理を実施している箇所と呼び出し方法について説明します。

周期測定用の A/D 変換が完了すると、メイン関数内から `R_Calc_PhysicalQuantity` 関数が呼び出されます。`R_Calc_PhysicalQuantity` 関数内では関数ポインタにより、差動電圧を演算するための関数として、`R_Voltage_Measurement_Calc` 関数が呼び出されます。本サンプルコードでは、`R_Voltage_Measurement_Calc` 関数内で A/D 変換値から差動電圧を演算しています。

## (3) 自己補正の周期を変更する方法

本サンプルコードでは、10 分間の測定にリアルタイム・クロックの定周期割り込み機能を用いて実現しています。実現方法としては、リアルタイム・クロックの定周期割り込み (INTRTC) の選択で 1 分に 1 度 (毎分 00 秒) に設定し、割り込み回数を 10 回カウントして 10 分を生成しています。そのため、下記方法で自己補正の周期を変更します。

### (a) 測定単位(秒、時、日など)を変更する場合

リアルタイム・クロックの定周期割り込みの周期を変更します。リアルタイム・クロック・コントロール・レジスタ 0 (RTCC0) の CT2-CT0 ビットの値を変更してください。なお、リアルタイム・クロックの詳細な使い方につきましては、RL78/G1A ユーザーズマニュアル ハードウェア編および RL78/G13 リアルタイム・クロック (R01AN0454J) アプリケーションノートをご参照ください。

### (b) 分単位で変更する場合

RTC 割り込み回数カウンタの判定条件を変更します。判定条件は `r_cg_rtc_user.c` ファイルの `r_rtc_interrupt_counter` 関数内で判定しています。

```
if ( D_RTC_COUNT_MINUTES <= s_interrupt_counter )
```

したがって、`smart_analog_sample_code_common.h` ファイルで定義されている `D_RTC_COUNT_MINUTES` の値を指定の時間に変更してください。

#### (4) 差動アンプ構成でチャンネルやゲインを変更する場合の変更手順

アンプ構成は差動アンプ構成のままで、コンフィギュラブル・アンプのチャンネルやゲインを変更する方法について説明します。

##### ① Smart Analog IC 300 のレジスタ設定値を変更

使用するチャンネルに応じて self\_calibration.c ファイル内の g\_self\_cal\_11、g\_self\_cal\_12、g\_self\_cal\_21 変数を変更します。g\_self\_cal\_11 変数を例に挙げ、変更する箇所をコメントで示します。

self\_calibration.c ファイル

```
static const sa_ctl_set_data_t g_self_cal_11 =
{
    { D_DAC1_OUT_CH },      //A/D 変換するアナログ入力チャンネルを指定してください
    { D_AMP1_GAIN_ADDR },  //ご使用になるアンプのゲイン制御レジスタのアドレスを指定してください
    { D_AMP1_BIAS_ADDR },  //ご使用になるアンプのバイアス電圧供給用に接続された DAC の
                          //DAC 制御レジスタのアドレスを指定してください
    { D_WAIT100US_32MHZ }, //Smart Analog のセトリング・タイムを指定してください
    { D_IC300_REG_NUM },   //Smart Analog のレジスタ数
    {
        { 0x00, 0x80 } ,      //Smart Analog IC 300 レジスタのアドレス、設定値を指定してください
        { 0x01, 0x80 } ,
        { 0x03, 0x00 } ,
        { 0x04, 0x00 } ,
        { 0x05, 0x00 } ,
        { 0x06, 0x00 } ,
        { 0x07, 0x00 } ,
        { 0x08, 0x00 } ,
        { 0x09, 0x00 } ,
        { 0x0b, 0x00 } ,
        { 0x0c, 0x00 } ,
        { 0x0d, 0x80 } ,
        { 0x0e, 0x80 } ,
        { 0x0f, 0x80 } ,
        { 0x10, 0x80 } ,
        { 0x13, 0x00 } ,
        { 0x14, 0x00 } ,
        { 0x15, 0x80 } ,
        { 0x16, 0x80 } ,
        { 0x17, 0x80 } ,
        { 0x11, 0x10 } ,
        { 0x12, 0x00 } ,
    },
};
```

##### ② ゲイン格納変数を変更する

測定するゲインに応じてゲイン格納変数 amp\_gain\_val の値を変更します。

calibration.c ファイル内の R\_Calibration 関数

```
ic300_diff_amp_gain_t amp_gain_val = E_DIFF_GAIN_40DB;    //ご使用になるゲインのレ
ジスタ値を指定してください
```

## ③ 呼び出す関数の引数を変更する

コンフィギュラブル・アンプのチャンネルを変更する場合は、R\_SelfCalibAmp1DiffGainAverage 関数や R\_SelfCalibAmp1DiffOffsetCalc 関数の引数にあるアンプチャンネルを変更します。

calibration.c ファイル内の R\_Calibration 関数

```
state_execution_flag = R_SelfCalibAmp1DiffGainAverage(E_AMP1 , E_DIFF,
amp_gain_val ); //ご使用になるコンフィギュラブル・アンプのチャンネルを指定してください
```

```
state_execution_flag = R_SelfCalibAmp1DiffOffsetCalc( E_AMP1 , E_DIFF,
amp_gain_val ); //ご使用になるコンフィギュラブル・アンプのチャンネルを指定してください
```

## (5) コンフィギュラブル・アンプのアンプ構成を変更する方法

下記手順にしたがって自己補正用の設定格納変数、ステートや関数を変更してください。なお、詳細な測定手順につきましては原理編をご参照ください。

## ① Smart Analog IC 300 のレジスタ設定追加

self\_calibration.c ファイルに測定する設定の条件を追加します。

```
static const sa_ctl_set_data_t g_self_cal_11 =
{
    { D_DAC1_OUT_CH },      //A/D 変換するアナログ入力チャンネルを指定してください
    { D_AMP1_GAIN_ADDR },  //ご使用になるアンプのゲイン制御レジスタのアドレスを指定してください
    { D_AMP1_BIAS_ADDR },  //ご使用になるアンプのバイアス電圧供給用に接続された DAC の
                          //DAC 制御レジスタのアドレスを指定してください
    { D_WAIT100US_32MHZ }, //Smart Analog のセトリング・タイムを指定してください
    { D_IC300_REG_NUM },   //Smart Analog のレジスタ数
    {
        { 0x00, 0x80 } ,      //Smart Analog IC 300 レジスタのアドレス、設定値を指定してください
        { 0x01, 0x80 } ,
        { 0x03, 0x00 } ,
        { 0x04, 0x00 } ,
        { 0x05, 0x00 } ,
        { 0x06, 0x00 } ,
        { 0x07, 0x00 } ,
        { 0x08, 0x00 } ,
        { 0x09, 0x00 } ,
        { 0x0b, 0x00 } ,
        { 0x0c, 0x00 } ,
        { 0x0d, 0x80 } ,
        { 0x0e, 0x80 } ,
        { 0x0f, 0x80 } ,
        { 0x10, 0x80 } ,
        { 0x13, 0x00 } ,
        { 0x14, 0x00 } ,
        { 0x15, 0x80 } ,
        { 0x16, 0x80 } ,
        { 0x17, 0x80 } ,
        { 0x11, 0x10 } ,
        { 0x12, 0x00 } ,
    },
};
```



## ② self\_calibration\_state\_t および calibration\_state\_t の列挙型にステート名を追加

self\_calibration\_state\_t 型に自己補正時の Smart Analog IC の状態を、calibration\_state\_t 型には自己補正時の測定ステートの定義を追加します。

smart\_analog\_sample\_code\_common.h ファイル

```
typedef enum
{
    E_CONF_AMPL_DAC1_OUT_DIRECT = 0U, //AMP1 差動アンプ用の DAC の直接測定する設定
    E_CONF_AMPL_DIFF_DAC_AMPL_OUT,    //AMP1 差動アンプ用の DAC のアンプを介して測定する設定
    E_CONF_AMPL_DIFF_GAIN_MES,        //AMP1 差動アンプ用の AMP のゲイン誤差を測定する設定
    //ここに①で変更した変数の定義を追加してください
    E_CONF_MAX,                        //MAX
} self_calibration_state_t;

typedef enum
{
    E_CALIBRATION_NO_EVENT=0U,          //処理なし
    E_CALIBRATION_INIT,                 //初期化ステート
    E_CALIBRATION_END,                  //終了ステート
    E_SELF_CALIBRATION_DIFF_GAIN1_MEASURE,
    E_SELF_CALIBRATION_DIFF_GAIN1_CALCULATE,
    E_SELF_CALIBRATION_DIFF_GAIN2_MEASURE,
    E_SELF_CALIBRATION_DIFF_GAIN2_CALCULATE,
    E_SELF_CALIBRATION_DIFF_GAIN3_MEASURE,
    E_SELF_CALIBRATION_DIFF_GAIN3_CALCULATE,
    E_SELF_CALIBRATION_DIFF_GAIN_AVERAGE,
    E_SELF_CALIBRATION_DIFF_OFFSET_DAC_MEASURE,
    E_SELF_CALIBRATION_DIFF_OFFSET_AMP_MEASURE,
    E_SELF_CALIBRATION_DIFF_OFFSET_CALCULATE,
    E_CALIBRATION_GAIN_AMPL_DIFF_40DB_END,
    //ここに①で変更した変数を測定するステート名の定義を追加してください
    E_CALIBRATION_ENUM_MAX,             //テーブルの最大値
} calibration_state_t;
```

## ③ 関数を作成する

下記関数の名前を複製し、任意の名前に変更します。

self\_calibration.c ファイル

```
R_SelfCalibAmplDiffGainMeasure
R_SelfCalibAmplDiffGainCalc
R_SelfCalibAmplDiffGainMeasure
R_SelfCalibAmplDiffGainAverage
R_SelfCalibAmplDiffOffsetDac
R_SelfCalibAmplDiffOffsetAmp
R_SelfCalibAmplDiffOffsetCalc
```

## ④ キャリブレーション制御関数にステートを追加する

switch ( calibration\_state)文の中に calibration\_state\_t で追加した新しいステートを追加します。下記、コード間をコピーし、ステート名を新しいステートに変更します。

```

case E_SELF_CALIBRATION_DIFF_GAIN1_MEASURE:
    if ( D_OFF == state_execution_flag )
    {
        state_execution_flag = R_SelfCalibAmp1DiffGainMeasure(0x20,
amp_gain_val, E_CONF_AMP1_DIFF_GAIN_MES );
    }
    break;

:
:
:

case E_SELF_CALIBRATION_DIFF_OFFSET_CALCULATE:
    if ( D_OFF == state_execution_flag )
    {
        state_execution_flag = R_SelfCalibAmp1DiffOffsetCalc( E_AMP1 , E_DIFF,
amp_gain_val );
    }
    break;

```

⑤ ④で追加したステート内で呼び出している関数を③で作成した関数に変更します。

⑥ ⑤で呼び出す関数の引数を変更します。

## ⑦ 自己補正ステート制御関数に②で追加したステート定義を追加する

calibration\_get\_state\_rtc\_tbl 関数内のステート更新処理を任意の順番に変更します。

```

case E_CALIBRATION_INIT: //自己補正の初期化後、最初に実行すべきステートに変更してください
    ret_state = E_SELF_CALIBRATION_DIFF_GAIN1_MEASURE;
break;

```

```

case E_SELF_CALIBRATION_DIFF_GAIN1_MEASURE:
.
.
.

case E_SELF_CALIBRATION_DIFF_OFFSET_CALCULATE:
    ret_state = pre_state + 1; //次のステートに移行
break;

```

## 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
Rev.1.00	2014.01.31	－	初版発行

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。

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようにご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社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。



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 〒100-0004 千代田区大手町2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/contact/>