Smart Analog IC 300

MCU software used for auto calibration

R02AN0020EJ0100
Rev.1.00
Jan 31, 2014

Introduction

This application note describes a sample program that measures and corrects errors (gain errors and offset voltage) that occur in the configurable amplifier of Smart Analog IC 300 by using a D/A converter incorporated in Smart Analog IC 300 and an A/D converter incorporated in the microcontroller for control.

Target Device

Smart Analog IC 300 (RAA730300), RL78/G1A (R5F10ELE)

When applying this application note to other microcontrollers, make the necessary changes according to the specifications of the microcontroller and verify them thoroughly.

Contents

1. Specifications

The basis of method for measuring and correcting the gain error and input-referred offset voltage generated in Smart Analog IC 300 is described in *Smart Analog IC 300 How to measure and correct the gain error and offset voltage automatically(R02AN0019E) Application Note*. (Hereafter Smart Analog IC 300 How to measure and correct the gain error and offset voltage automatically (R02AN0019E) Application Note is referred to as *Basis AN*.) This application note describes how to execute auto calibration described in Basis AN by using MCU software and how to use sample code.

The gain error and input-referred offset voltage vary depending on the ambient factors such as temperature. Therefore, auto measurement and calibration need to be performed periodically after shipment. This application note uses an example that measure the errors for auto calibration every 10 minutes. Figure 1-1 shows the state transition timing of auto calibration and measurement.

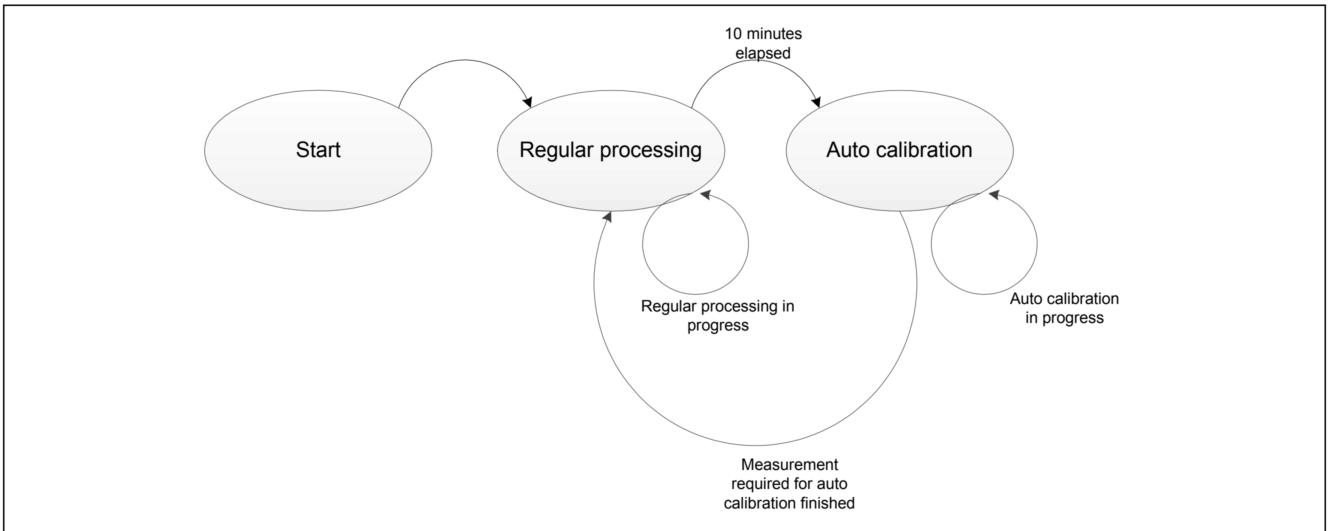


Figure 1-1 State transition during auto calibration measurement

In this application note, configurable amplifier Ch1 is used for auto calibration and measurement of gain errors and input-referred offset voltage and is used as a differential amplifier with a gain of 40 dB. For details about auto calibration including measurement of gain errors and input-referred offset voltage when using a configurable amplifier as a differential amplifier, see section 3.2 in Basis AN. When using another configurable amplifier channel or amplifier configuration, see the sample code or Basis AN.

The auto calibration start timing is generated based on the real-time clock, as an interrupt generated every one minute. When the start trigger interrupt occurs 10 times, auto calibration and measurement starts, and the gain error and input-referred offset voltage are measured in that order. When auto calibration ends, the process transits to regular processing. This operation is performed repeatedly in this application note.

2. Conditions Under Which Operation Has Been Verified

The operation of the sample code shown in this application note has been verified under the conditions shown below.

Table 2-1 Conditions Under Which Operation Has Been Verified

Parameter	Description
Devices used	RL78/G1A (R5F10ELE) Smart Analog IC 300 (RAA730300)
Board used	TSA-IC300 (from Tessera Technology, Inc.)
Operating frequency	High-speed on-chip oscillator clock: 32 MHz CPU/peripheral hardware clock: 32 MHz
Operating voltage	3.3 V
Integrated development environment	made by Renesas Electronics CubeSuite+ V2.00.00 [15 Mar 2013]
C compiler	made by Renesas Electronics CA78K0R V1.60
RL78/G1A code library	made by Renesas Electronics CodeGenerator for RL78/G1A V2.00.03

3. Related Application Notes

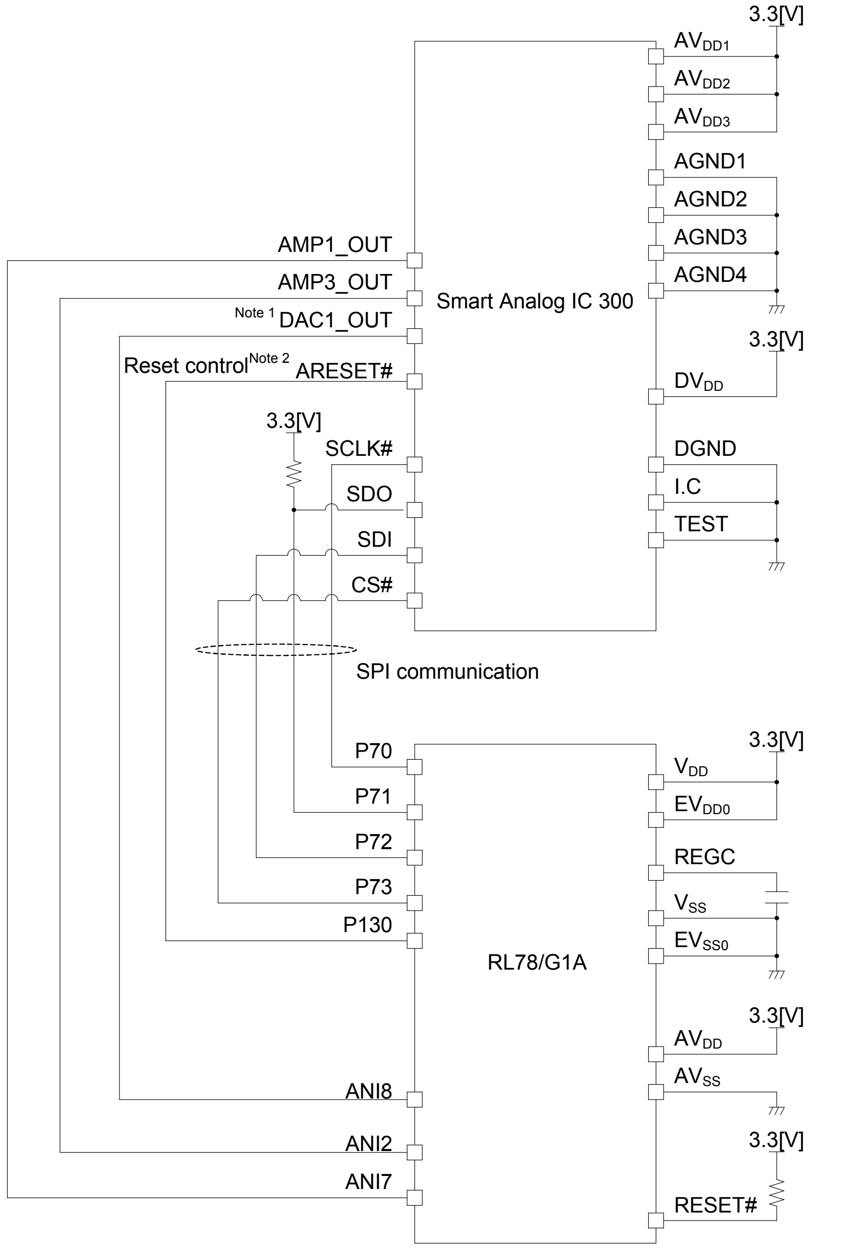
Also refer to these documents when using this application note.

- RL78/G1E Switching Amplifiers When Connecting Multiple Sensors Application Note (R01AN1129E)
- RL78/G1E Sample Code for Performing SPI Communication with Analog Block Application Note (R01AN1130E)
- Smart Analog IC 300 Selecting Amplifiers Based on Sensor Type (R02AN0016E) Application Note
- Smart Analog IC 300/301 Features and Usage Examples (Comparison with Smart Analog IC 500 Series) (R02AN0018E) Application Note
- Smart Analog IC 300 How to measure and correct the gain error and offset voltage automatically (R02AN0019E) Application Note
- RL78/G13 Real-time Clock (R01AN0454E) Application Note

4. Hardware

4.1 Hardware Configuration Example

Figure 4-1 shows an example of the hardware configuration described in this application note.



Note 1: The analog block is reset via the P130 pin output.
After the reset of the microcontroller block ends, P130 outputs a high level.
Note 2: When using the evaluation board TSA-IC300, note that the DAC1_OUT and ANI8 pins are not connected by default.

Figure 4-1 Hardware Configuration Example

Caution This circuit diagram is simplified to show an overview of the circuit connection. When designing an actual circuit, connect pins appropriately so as to satisfy the electrical specifications. (Connect input-only ports to V_{DD} or V_{SS} individually via a resistor.)

4.2 Functions Used

Table 4-1 shows the the functions of Smart Analog IC 300 (RAA73030) and the RL78/G1A (R5F10ELE) used in this application note and their applications.

Table 4-1 Features used in Smart Analog IC 300 (RAA730300) and RL78/G1A (R5F10ELE) and their applications

Device	Feature	Applications
Smart Analog IC 300	Configurable amplifier Ch1	Subject to auto calibration and measurement Used as a differential amplifier with a gain of 40 dB
	D/A converter Ch1	Used to measure the gain error and input-referred offset voltage for calibration (used to generate a bias voltage for differential amplifier)
	D/A converter Ch1	Used to measure the gain error and input-referred offset voltage for auto calibration (signal applied to differential amplifier)
	D/A converter Ch1	Used to measure the gain error for auto calibration (signal applied to differential amplifier)
	SPI interface	Controls SPI communication with the RL78/G1A
RL78/G1A	A/D converter	Converts the voltage output from the configurable amplifier and D/A converter to a digital value.
	High-speed on-chip oscillator clock: 32 MHz	Generates the 32 MHz clock used as the main system clock.
	Serial array unit 1 Channel 1	Controls SPI communication with the analog block by using the 3-wire serial I/O function (CSI21).
	I/O ports	Controls the reset of Smart Analog IC 300, and the chip select signal (CS) used to control SPI communication.
	Timer array unit 0 Channel 5	Used to measure the settling time of Smart Analog IC 300.
	Real-time clock	Used to generate timing for auto calibration and measurement (generates an interrupt per minute)
	DMA controller	Used to transfer the obtained A/D-converted value (ADCR) to the internal RAM

4.3 Pins Used

Table 4-2 shows the Smart Analog IC 300 pins used in this application note and their features.

Table 4-2 Smart Analog IC 300 pins used and their roles

Pin Name	I/O	Description
AMP1_OUT	Output	Connected to A/D converter analog input pin 7 (ANI7) for auto calibration and measurement of gain errors and input-referred offset voltage
DAC1_OUT	Output	Connected to A/D converter analog input pin 8 (ANI8) for auto calibration and measurement of input-referred offset voltage
ARESET#	Input	Connected to P130 to control the Smart Analog IC 300 reset feature
SCLK#	Input	Connected to SCLK# (3-wire serial I/O) to control SPI communication
SDO	I/O	Connected to SDO (3-wire serial I/O) to control SPI communication
SDI	I/O	Connected to SDI (3-wire serial I/O) to control SPI communication
CS#	Input	Connected to P73 to control SPI communication

5. Auto calibration

This chapter describes the procedure and timing for measuring the gain error and input-referred offset voltage, state transition, and values set to Smart Analog IC 300.

5.1 Timing chart for auto calibration

In this application note, timing to start auto calibration is generated by using the real-time clock, the gain error and input-referred offset voltage are measured and corrected in that order. Figure 5-1 shows the auto calibration start timing used in this application note.

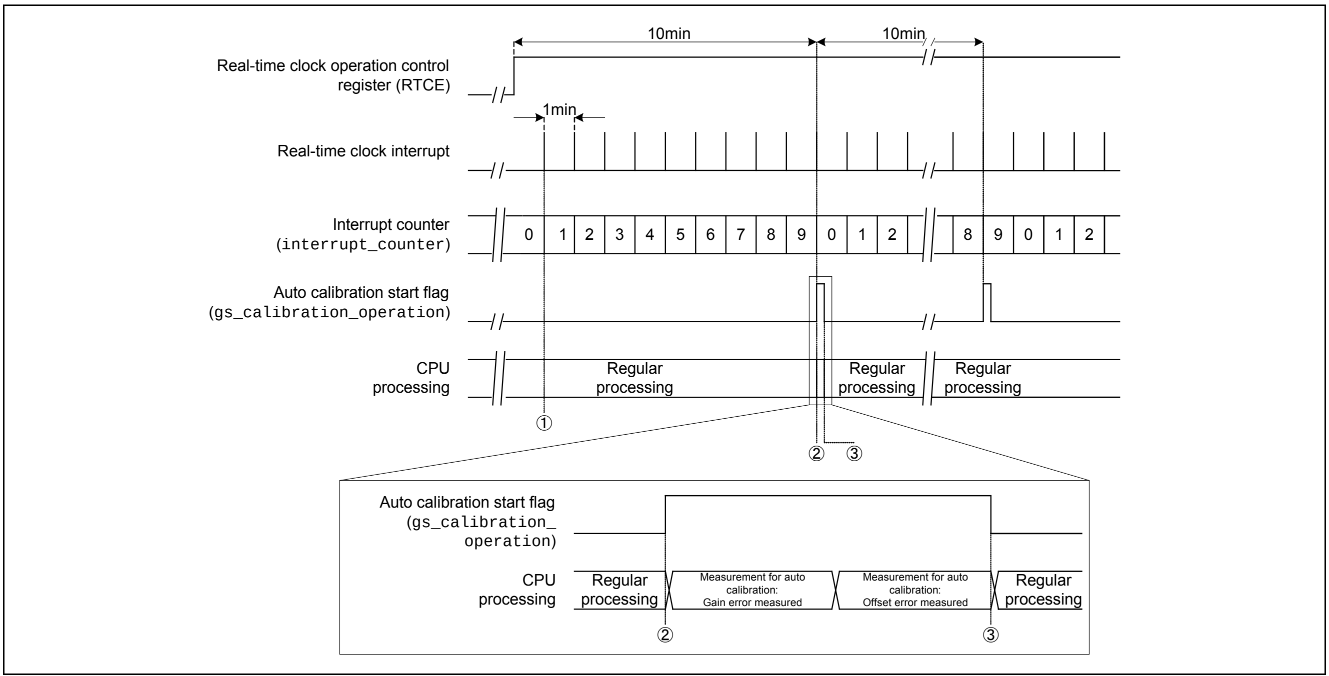


Figure 5-1 Timing chart for auto calibration

① An interrupt is generated by using the constant-period interrupt (INTRTC) of the real-time clock every minute, and the RTC interrupt counter is incremented by 1.

② When the RTC interrupt counter reaches 0 (10 minutes after the count start), the auto calibration start trigger flag is set.

③ When measurement of gain errors and input-referred offset voltage required for auto calibration ends, the auto calibration start trigger flag is cleared.

5.2 Gain error measurement

During gain error measurement, the gain is calculated based on the voltage input to and output from the configurable amplifier, and the difference from the specified value is calculated. During gain error measurement, to minimize the error in D/A converters and the effect of the offset voltage in the configurable amplifier, a difference between to voltage values are used for calculation. The following shows the measurement procedure, timing charts, and the Smart Analog IC 300 register settings.

5.2.1 Procedure for measuring the gain error

To measure the gain error, perform the measurement by using the procedure shown in Figure 3-2 in Basis AN. In this application note, configurable amplifier Ch1 is used as a differential amplifier with a gain of 40 dB and D/A converter channels Ch1, Ch5, and Ch6 are used. The following shows the procedure for measuring the gain error.

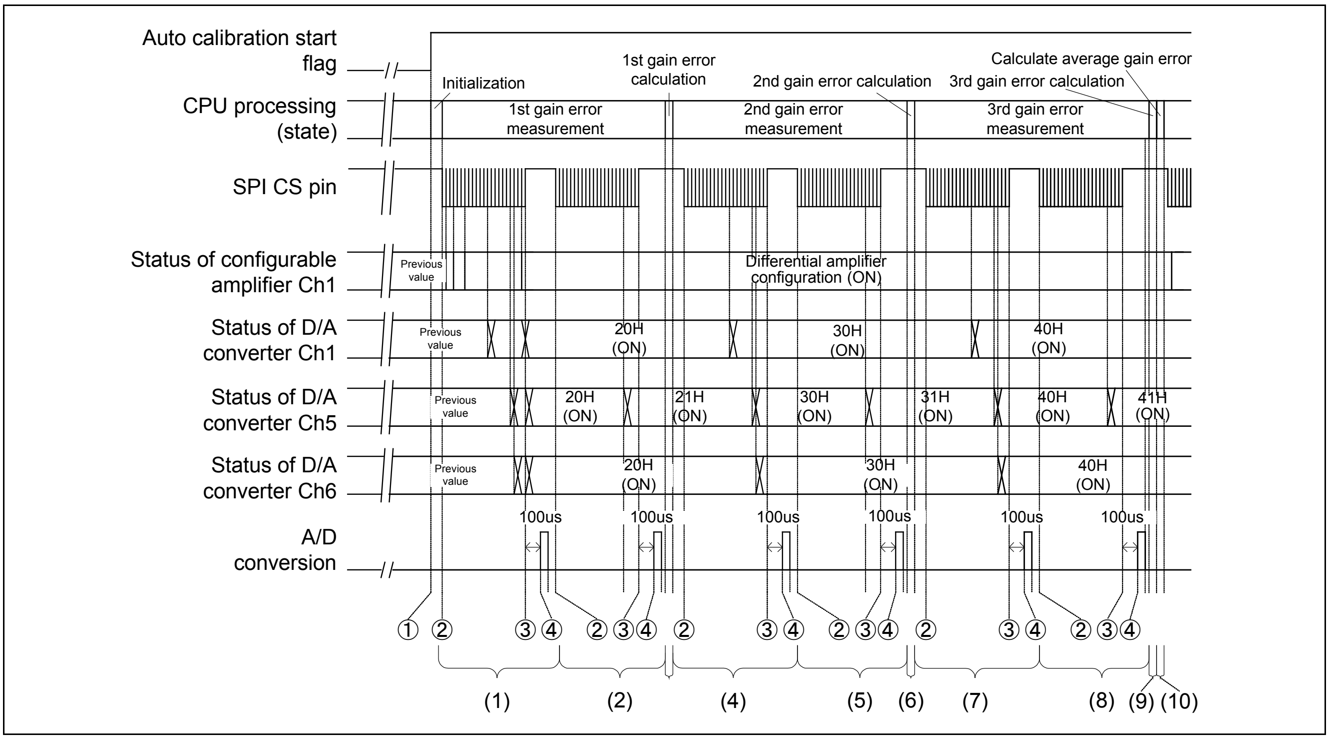


Figure 5-2 Timing chart for measuring the gain error

1 Initializing A/D converter
When the auto calibration start trigger flag is set, initialization required to start auto calibration starts. During initialization, the A/D conversion trigger is changed to the software trigger mode or sequential conversion mode

(1) **Measuring the minimum value voltage when 20H is set**

2 Changing the Smart Analog register settings
Change the Smart Analog IC 300 register settings as follows: (see Figure 5-3)

a Configurable amplifier configuration
To use configurable amplifier Ch1 as a differential amplifier with a gain of 40 dB, set configuration register 1 (CONFIG1) to 90H, and gain control register 1 (GC1) to 11H.

b MPX settings
To connect D/A converter Ch6 to the inverted input pin and D/A converter Ch5 to the non-inverted input pin of configurable amplifier Ch1, set MPX setting register 1(MPX1) to E8H, and configuration register 2 (CONFIG2) to 83H.

c D/A converter settings

Set DAC control registers (DAC1C, DAC5C, and DAC6C) corresponding to D/A converter channels Ch1, Ch5, and Ch6 to 20H.

d Enabling configurable amplifier and D/A converters

Set power control register 1 (PC1) to 11H and power control register 2 (PC2) to 60H to enable configurable amplifier Ch1 and converter channels Ch1, Ch5, and Ch6.

3 After the Smart Analog register settings are specified, the system waits for the settling time (100 μs).

4 After the settling time has elapsed, A/D-convert the voltage output from configurable amplifier Ch1 (AMP1_OUT pin). At this time, A/D conversion is executed 10 times in succession, and the values excluding the maximum and minimum value (that is, data of eight conversions) are averaged.

(2) Measuring the maximum value voltage when 20H is set

To measure the maximum value voltage when 20H is set, repeat steps 2 to 4 above. Here, set D/A converter Ch5 specified in 2-(c) to 21H. (see Figure 5-3)

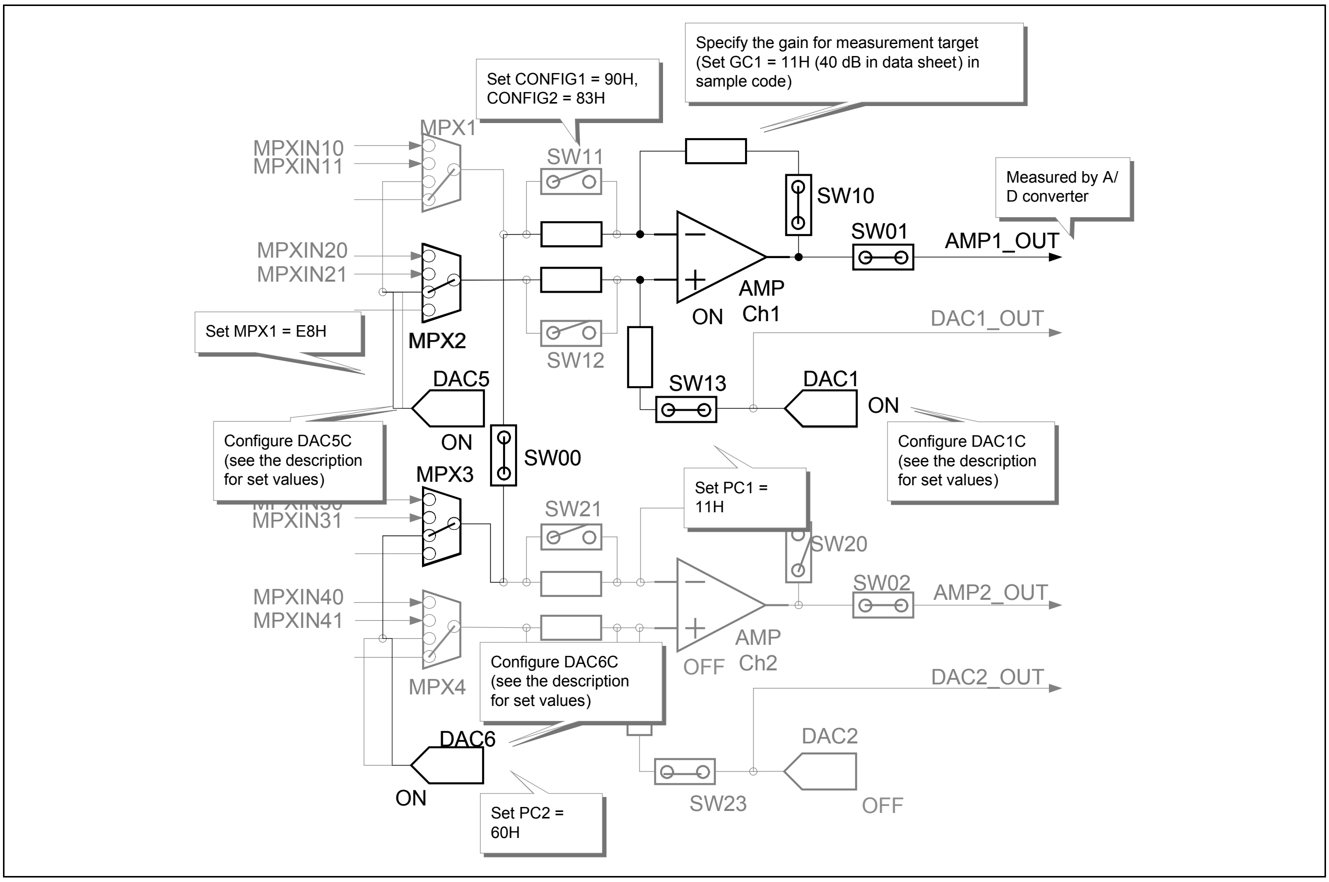


Figure 5-3 Measuring the gain error

(3) Calculating the gain error when 20H is set

Calculate the gain error based on the A/D-converted values and D/A converter set values obtained in (1) and (2)

(4) Measuring the minimum value voltage when 30H is set

Perform the same procedure as shown in (1). Note, however, set each D/A converter to 30H.

(5) Measuring the maximum value voltage when 30H is set

Perform the same procedure as shown in (2). Note, however, set each D/A converter to 30H.

(6) Calculating the gain error when 30H is set

Calculate the gain error based on the A/D-converted values and D/A converter set values obtained in (4) and (5)

(7) Measuring the minimum value voltage when 40H is set

Perform the same procedure as shown in (1). Note, however, set each D/A converter to 40H.

(8) Measuring the maximum value voltage when 40H is set

Perform the same procedure as shown in (7). Note, however, set each D/A converter to 40H.

(9) Calculating the gain error when 40H is set

Calculate the gain error based on the A/D-converted values and D/A converter set values obtained in (7) and (8)

(10) Calculate average gain error

Average the gain errors o\calculated in (3), (6), and (9).

The gain error can now be calculated.

5.2.2 Smart Analog IC 300 register settings for measuring the gain error

Smart Analog IC 300 register settings for measuring the gain error are shown in Table 5-2. For details about the register setting values, see the 6.4 Smart Analog IC 300 register values and *RAA730300 Monolithic Programmable Analog IC Datasheet*.

Table 5-1 Smart Analog IC 300 register settings for measuring the gain error

Address	SPI control register		Set value
00H	Configuration register 1	CONFIG1	90H
01H	Configuration register 2	CONFIG2	83H
03H	MPX setting register 1	MPX1	E8H
04H	MPX setting register 2	MPX2	00H
05H	MPX setting register 3	MPX3	00H
06H	Gain control register 1	GC1	11H
07H	Gain control register 2	GC2	00H
08H	Gain control register 3	GC3	00H
09H	AMP operation mode control register	AOMC	00H
0BH	LDO control register	LDOC	00H
0CH	DAC reference voltage control register	DACRC	00H
0DH	DAC control register 1	DAC1C	**H
0EH	DAC control register 2	DAC2C	80H
0FH	DAC control register 3	DAC3C	80H
10H	DAC control register 4	DAC4C	80H
11H	Power control register 1	PC1	11H
12H	Power control register 2	PC2	60H
13H	Reset control register	RC	00H
14H	Input mode control register	IMS	00H
15H	DAC control register 5	DAC5C	**H
16H	DAC control register 6	DAC6C	**H
17H	DAC control register 7	DAC7C	80H

Remark *: The DAC1C, DAC5C, and DAC6C settings vary depending on the measurement conditions.

5.3 Calculating the input-referred offset voltage

The input-referred offset voltage can be calculated by using the difference between the voltage input to and output from the configurable amplifier. In this application note, the input-referred offset voltage of configurable amplifier Ch1 is measured For the basis of the input-referred offset voltage measurement, see section 3.2 in Basis AN. The following shows the measurement procedure, timing charts, and the Smart Analog IC 300 register settings.

5.3.1 Calculating the input-referred offset voltage

Measuring the input-referred offset voltage starts when measurement of gain errors ends

The following shows the procedure for measuring the input-referred offset voltage.

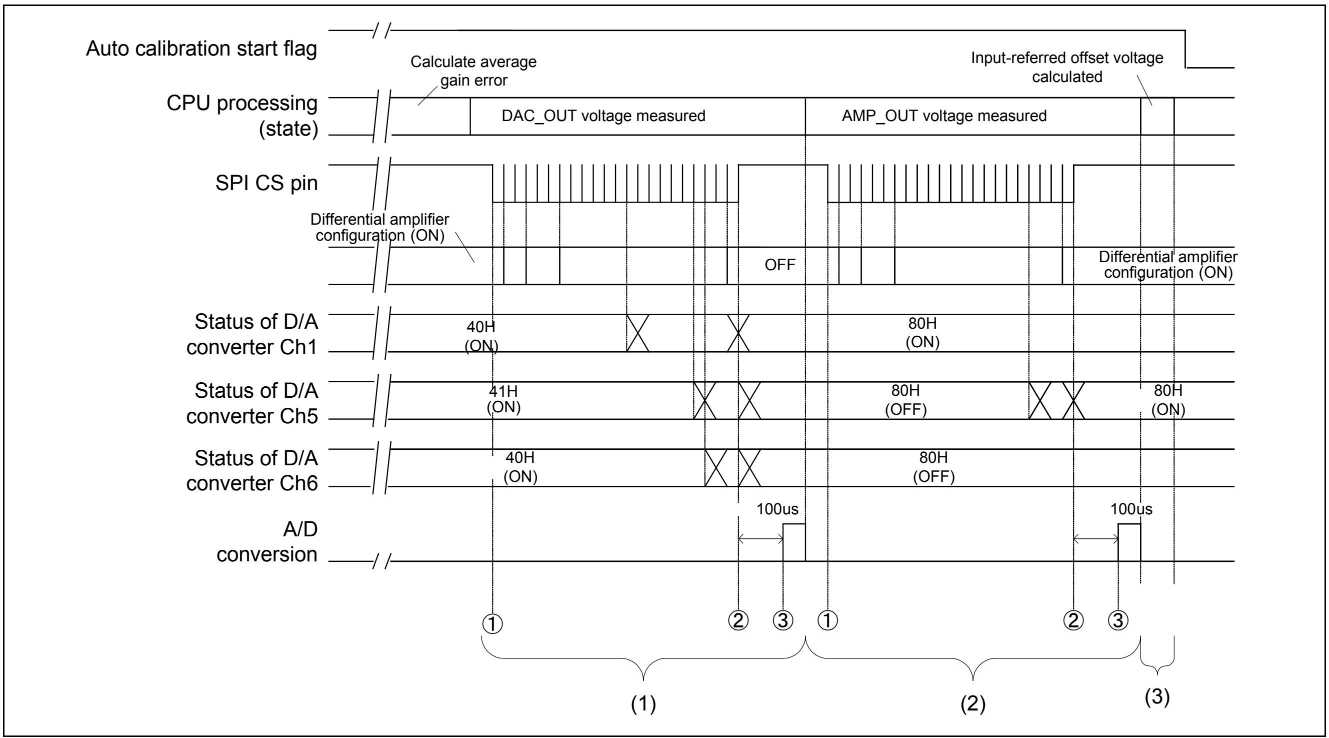


Figure 5-4 timing chart for measuring the input-referred offset voltage

(1) **Measuring D/A converter Ch1 for output voltage**

1 Changing the Smart Analog register settings
Change the Smart Analog IC 300 register settings as follows: (see Figure 5-5)

a Changing the configurable amplifier switches
Select the configurable amplifier Ch1 switches so as not to be affected by the D/A converter output voltage. Set configuration register 1 (CONFIG1) to 80H.

b MPX settings
Reset the configurable amplifier Ch1 input pin to the default value. This setting does not affect measurement, so you can skip this setting.

c D/A converter settings
Set DAC control registers (DAC1C, DAC5C, and DAC6C) corresponding to D/A converter channels Ch1, Ch5, and Ch6 to 80H.

d Enabling configurable amplifier and D/A converters
Set power control register 1 (PC1) to 10H and power control register 2 (PC2) to 00H to disable configurable amplifier Ch1 and converter channels Ch5 and Ch6, and enable D/A converter channel Ch1.

② After the Smart Analog register settings are specified, the system waits for the settling time (100 µs).

③ After the settling time has elapsed, A/D-convert the voltage output from D/A converter Ch1 (DAC1_OUT pin). At this time, A/D conversion is executed 10 times in succession, and the values excluding the maximum and minimum value (that is, data of eight conversions) are averaged.

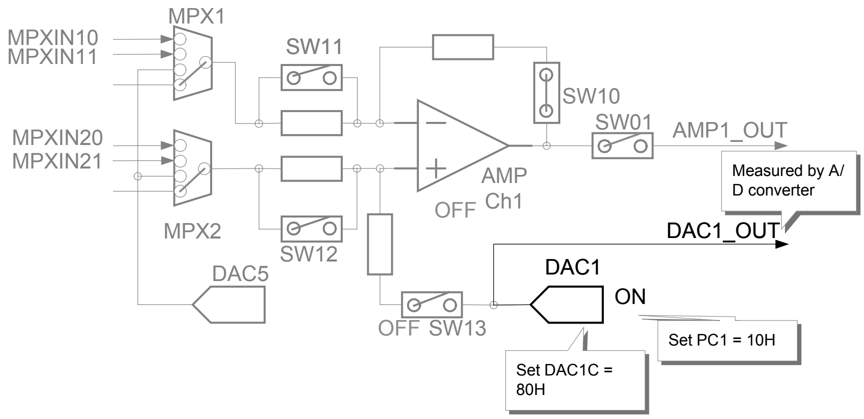


Figure 5-5 Measuring the D/A converter Ch1 output voltage

(2) Measuring the onfigurable amplifier Ch1 (AMP1_OUT) output voltage

1 Changing the Smart Analog register settings
Change the Smart Analog IC 300 register settings as follows: (see Figure 5-6)

a Configurable amplifier configuration
To use configurable amplifier Ch1 as a differential amplifier with a gain of 40 dB, set configuration register 1 (CONFIG1) to 90H, and gain control register 1 (GC1) to 11H.

b MPX settings
To connect D/A converter Ch5 to the inverted and non-inverted input pins of configurable amplifier Ch1, set MPX setting register 1(MPX1) to A0H.

c D/A converter settings
Set DAC control registers (DAC1C, DAC5C, and DAC6C) corresponding to D/A converter channels Ch1, Ch5, and Ch6 to 80H.

d Enabling configurable amplifier and D/A converters
Set power control register 1 (PC1) to 11H and power control register 2 (PC2) to 60H to enable configurable amplifier Ch1 and converter channels Ch1, Ch5, and Ch6.

2 After the Smart Analog register settings are specified, the system waits for the settling time (100 µs).

3 After the settling time has elapsed, A/D-convert the voltage output from configurable amplifier Ch1 (AMP1_OUT pin). At this time, A/D conversion is executed 10 times in succession, and the values excluding the maximum and minimum value (that is, data of eight conversions) are averaged.

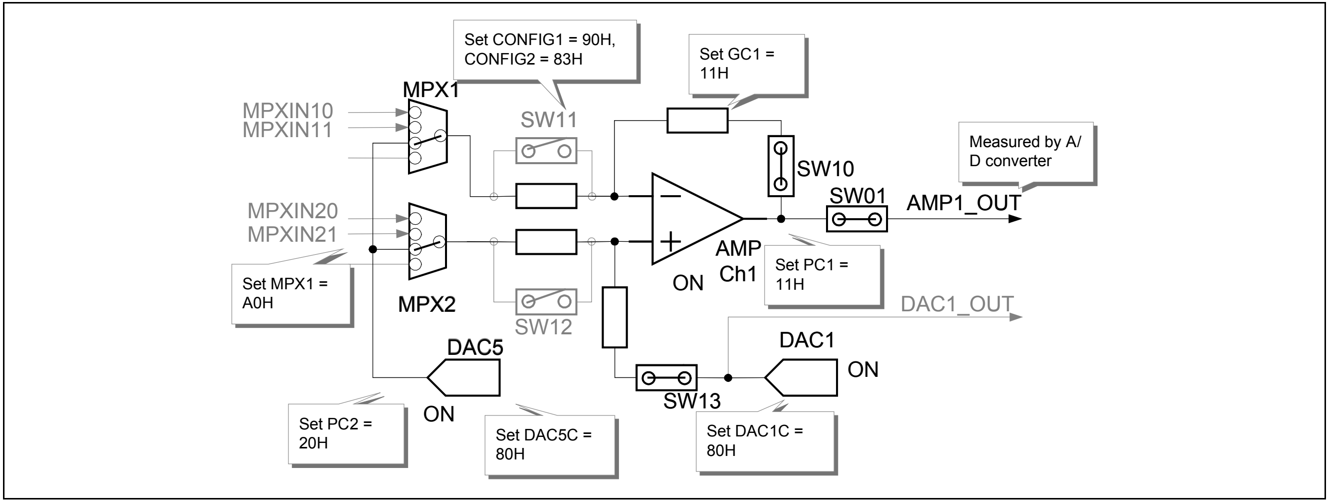


Figure 5-6 Measuring the configurable amplifier Ch1 (AMP1_OUT) output voltage

(3) Calculating the input-referred offset voltage

Calculate the input-referred offset voltage. The input-referred offset voltage can be calculated by subtracting the A/D-converted DAC1_OUT output value obtained in (1) from the A/D-converted AMP1_OUT output obtained in (2) and dividing the result by gain (40 dB).

The input-referred offset voltage can now be measured.

5.3.2 Smart Analog IC 300 register settings for measuring the input-referred offset voltage

Table 5-2 shows the Smart Analog IC 300 register settings for measuring the input-referred offset voltage For details about the register setting values, see the 6.4 Smart Analog IC 300 register values and *RAA730300 Monolithic Programmable Analog IC Datasheet*.

Table 5-2 Smart Analog IC 300 register settings for measuring the input-referred offset voltage

Address	SPI control register		Set value	
			D/A converter Ch1 output voltage measurement	Configurable amplifier Ch1 output voltage measurement
00H	Configuration register 1	CONFIG1	80H	90H
01H	Configuration register 2	CONFIG2	80H	82H
03H	MPX setting register 1	MPX1	00H	A0H
04H	MPX setting register 2	MPX2	00H	00H
05H	MPX setting register 3	MPX3	00H	00H
06H	Gain control register 1	GC1	00H	11H
07H	Gain control register 2	GC2	00H	00H
08H	Gain control register 3	GC3	00H	00H
09H	AMP operation mode control register	AOMC	00H	00H
0BH	LDO control register	LDOC	00H	00H
0CH	DAC reference voltage control register	DACRC	00H	00H
0DH	DAC control register 1	DAC1C	80H	80H
0EH	DAC control register 2	DAC2C	80H	80H
0FH	DAC control register 3	DAC3C	80H	80H
10H	DAC control register 4	DAC4C	80H	80H
11H	Power control register 1	PC1	10H	11H
12H	Power control register 2	PC2	00H	20H
13H	Reset control register	RC	00H	00H
14H	Input mode control register	IMS	00H	00H
15H	DAC control register 5	DAC5C	80H	80H
16H	DAC control register 6	DAC6C	80H	80H
17H	DAC control register 7	DAC7C	80H	80H

6. Software

6.1 Functions

Table 6-1 Functions (1)

File Name	Function	Remark
r_systeminit.c	hdwinit	Output by the code generator
	R_Systeminit	
r_main.c	main	
	R_MAIN_UserInit	
calibration.c	R_Calibration	
	R_CalibrationIsDoing	
	calibration_init	
	calibration_end	
	calibration_trigger_check	
	calibaration_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_SmartAnalogMeasureControlInit	
	R_CalcAverageExceptMinMax	
	R_SmartAnalogRegBufSearch	
	R_SmartAnalogRegBufUpdate	
r_sa_spi_control_register.c	R_SPI_SmartAnalogRead	Use sample code in *RL78/G1E Sample Code for Performing SPI Communication with Analog Block Application Note (R01AN1130E)*
	R_SPI_SmartAnalogWrite	
	R_SPI_SmartAnalogWriteVerify	
	R_SPI_SmartAnalogReadBit	
	R_SPI_SmartAnalogWriteBit	
	R_SPI_SmartAnalogWriteVerifyBit	
r_cg_timer.c	R_TAU0_Create	Output by the code generator
	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	

Table 6-2 Functions (2)

File Name	Function	Remark
r_cg_serial_user.c	r_csi21_interrupt	Output by the code generator
	r_csi21_callback_error	Output by the code generator and processing added. Use sample code in *RL78/G1E Sample Code for Performing SPI Communication with Analog Block Application Note (R01AN1130E)*
	r_csi21_callback_receiveend	
r_cg_serial.c	R_SAU1_Create	Output by the code generator and processing added. Use sample code in *RL78/G1E Sample Code for Performing SPI Communication with Analog Block Application Note (R01AN1130E)*
	R_CSI21_Create	
	R_CSI21_Start	
	R_CSI21_Stop	
	R_CSI21_Send_Receive	
r_cg_rtc.c	R_RTC_Create	Output by the code generator
	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	Output by the code generator
	r_rtc_callback_constperiod	Output by the code generator and processing added.
	r_rtc_interrupt_counter	
r_cg_port.c	R_PORT_Create	Output by the code generator
r_cg_dmac.c	R_DMAC1_Create	Output by the code generator
	R_DMAC1_Start	
	R_DMAC1_Stop	
r_cg_dmac_user.c	R_DMAC1_ADC_Create	
	r_dmac1_interrupt	Output by the code generator and processing added.
r_cg_cgc.c	R_CGC_Create	Output by the code generator
r_cg_adc.c	R_ADC_Create	Output by the code generator
	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	Output by the code generator

6.2 Function specifications

The specifications of the major functions used in this application note are described below.

(1) **main function**

Declaration	void main(void)
Overview	main function - Calls the R_MAIN_UserIni function - Calls the R_Calibration function
Parameters	None
Global variables	None
Return value	None

(2) **R_MAIN_UserInit**

Declaration	void R_MAIN_UserInit(void)
Overview	User interface initialization function - Initializes the modules required for auto calibration
Parameters	None
Global variables	None
Return value	None

(3) **R_Calibration**

Declaration	void R_Calibration(void)
Overview	Calibration control function - Perform calibration. - Loops calibration until the processing ends.
Parameters	None
Global variables	None
Return value	None

(4) **R_CalibrationIsDoing**

Declaration	uint8_t R_CalibrationIsDoing (void)
Overview	Calibration trigger judge function - Determines whether a calibration trigger has occurred. - Add new calibration triggers to this function, if any.
Parameters	None
Global variables	None
Return value	ret 0U: No trigger 1U: A trigger has occurred

(5) **calibration_init**

Declaration	static void calibration_init (void)
Overview	Calibration initialization function - Changes the A/D conversion trigger type to the software trigger.
Parameters	None
Global variables	None
Return value	None

(6) **calibration_end**

Declaration	static void calibration_end (void)
Overview	Calibration completion function - If you want to stop or start a module when calibration is completed, describe it in this function.
Parameters	None
Global variables	None
Return value	None

(7) **calibration_trigger_check**

Declaration	static calibration_state_t calibration_trigger_check (calibration_state_t pre_state ,uint8_t * p_ret_state_execution_flag)
Overview	Calibration trigger check function - Checks the calibration trigger that has occurred. - Updates the calibration state.
Parameters	pre_state Current state p_ret_state_execution_flag Pointer to state execution flag
Global variables	None
Return value	calibration_state_ret Updated state

(8) **calibration_get_state_rtc_tbl**

Declaration	static calibration_state_t calibaration_get_state_rtc_tbl (calibration_state_t pre_state)
Overview	Auto calibration state control function - Checks and update the auto calibration state.
Parameters	pre_state Current state
Global variables	None
Return value	ret_state Updated state

(9) **R_SelfCalibrationInit**

Declaration	void R_SelfCalibrationInit (void)
Overview	Auto calibration global variable initialization function Initializations the global variables for auto calibration
Parameters	None
Global variables	gs_self_cal_sa_reg_buf Auto calibration setting storage variable : gp_self_cal_setting_data Pointer to the array that stores values set for auto calibration
Return value	None

(10) **setup_self_cal_AFE_measure_ch**

Declaration	void setup_self_cal_AFE_measure_ch (self_calibration_state_t self_cal_cnt)
Overview	Smart Analog register/AD converter setting change function - Overwrites the values of auto calibration setting storage variable to the Smart Analog register values. - Overwrites the auto calibration setting to the A/D converter analog input channel setting.
Parameters	set_cnt Index of SA setting storage variables (gs_self_cal_sa_reg_buf) being calibrated automatically
Global variables	gs_self_cal_sa_reg_buf Auto calibration setting storage variable
Return value	None

(11) **R_SelfCalibAmp1DiffOffsetDac**

Declaration	uint8_t R_SelfCalibAmp1DiffOffsetDac (void)
Overview	DAC1_OUT measurement function - A function for measuring the offset voltage when configurable amplifier Ch1 is used as a differential amplifier - Measures the voltage output from D/A converter Ch1 pin (DAC1_OUT) by using the A/D converter.
Parameters	None
Global variables	gs_offset_cal_data Variable for storing the measured value used to calculate the offset voltage
Return value	ret_state D_OFF: Not completed D_ON: Completed

(12) **R_SelfCalibAmp1DiffOffsetAmp**

Declaration	uint8_t R_SelfCalibAmp1DiffOffsetAmp (void)
Overview	AMP1_OUT measurement function - A function for measuring the offset voltage when configurable amplifier Ch1 is used as a differential amplifier - Specifies configurable amplifier Ch1 to be used as a differential amplifier with a gain of 40 dB. - Connects D/A converter Ch5 to the inverted and non-inverted input pins of configurable amplifier Ch1. - Measures the voltage output from configurable amplifier Ch1 pin (AMP1_OUT) by using the A/D converter.
Parameters	None
Global variables	Variable for storing the measured value used to calculate the offset voltage
Return value	ret_state D_OFF: Not completed D_ON: Completed

(13) **R_SelfCalibAmp1DiffOffsetCalc**

Declaration	uint8_t R_SelfCalibAmp1DiffOffsetCalc (amp_num_t amp_num, amp_config_t amp_config, uint8_t gain_reg_val)
Overview	Input-referred offset voltage calculation function - - A function for calculating the offset voltage when configurable amplifier Ch1 is used as a differential amplifier - Calculates the input-referred offset voltage based on the value measured by using functions R_SelfCalibAmp1DiffOffsetDac and R_SelfCalibAmp1DiffOffsetAmp .
Parameters	amp_num Number of amplifier channel amp_config Number of amplifier configuration gain_reg_val Gain register set value
Global variables	gs_offset_cal_data Variable for storing the measured value used to calculate the offset voltage g_amp_table Table for storing the gain and offset voltage in Smart Analog IC 300
Return value	D_ON Completed

(14) **R_SelfCalibAmp1DiffGainMeasure**

Declaration	uint8_t R_SelfCalibAmp1DiffGainMeasure (uint8_t dac_reg_val, ic300_diff_amp_gain_t amp1_gain, self_calibration_state_t self_cal_cnt)
Overview	Gain measurement function - Specifies configurable amplifier Ch1 to be used as a differential amplifier with a gain of 40 dB. - Specifies the minimum value voltage for D/A converter Ch5 to obtain the minimum value voltage. - Specifies the maximum value voltage for D/A converter Ch5 to obtain the maximum value voltage.
Parameters	dac_reg_val DAC register set values amp1_gain AMP1 gain register set values self_cal_cnt Auto calibration setting number
Global Variables	None
Return value	ret_state D_OFF: Processing in progress D_ON: Completed D_ERROR: Invalid parameter

(15) **R_SelfCalibAmp1DiffGainCalc**

Declaration	uint8_t R_SelfCalibAmp1DiffGainCalc (self_calibration_dac_cnt_t buf_num)
Overview	Gain calculation function - Calculates the gain based on the measured data.
Parameters	buf_num Number of buffer for storing the measured value used for calibration (0: 0x20U, 1: 0x30U, 2: 0x40U)
Global variables	gs_self_cal_gain_data Gain storage buffer that temporarily stores the calculated gain
Return value	ret_state D_ON Completed

(16) **R_SelfCalibAmp1DiffGainAverage**

Declaration	uint8_t R_SelfCalibAmp1DiffGainAverage (amp_num_t amp_num, amp_config_t amp_config, uint8_t gain_reg_val)
Overview	Gain averaging function - Averages the gain values measured three times. - Stores the calculated gain in the table for storing the gain and offset voltage in Smart Analog IC 300.
Parameters	amp_num Number of amplifier channel amp_config Number of amplifier configuration gain_reg_val Gain register set value
Global Variables	gs_self_cal_gain_data Gain storage buffer that stores the gain with the settings 0x20, 0x30, and 0x40 g_amp_table Table for storing the gain and offset voltage in Smart Analog IC 300 that stores the result of auto calibration obtained by using this function
Return value	ret_state D_ON: Completed

(17) **R_SelfCalibrationMeasure**

Declaration	uint8_t R_SelfCalibrationMeasure (uint16_t * p_adcr, self_calibration_state_t self_cal_cnt)
Overview	A/D conversion control function for auto calibration - Calls the setup_self_cal_AFE_measure_ch function to overwrite the Smart Analog register values. - Waits for the Smart Analog IC to be settled. - Obtains the A/D-converted values.
Parameters	p_adcr A/D-converted values (o be used to overwrite the Smart Analog registers in the process of this function self_cal_cnt Auto calibration setting number
Global Variables	None
Return value	ret 0U: Measurement in progress 1U: Completed

(18) **amp1_diff_calibration_setting**

Declaration	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)
Overview	DAC/gain setting function for gain measurement - Overwrites the setting for gain measurement to the settings of D/A converter channels Ch1, Ch5, and Ch6 stored in the gs_self_cal_sa_reg_buf variable. - Overwrites 40 dB to the configurable amplifier Ch1 gain stored in the gs_self_cal_sa_reg_buf variable.
Parameters	set_dac_reg Data to be overwritten to the DAC1, DAC5, and DAC6 settings set_gain_reg Data to be overwritten to the AMP1 gain. self_cal_cnt Auto calibration setting number
Global Variables	gs_self_cal_sa_reg_buf Auto calibration setting storage variable
Return value	None

(19) **R_SmartAnalogHardReset**

Declaration	uint8_t R_SmartAnalogHardReset (void)
Overview	Smart Analog reset function - Resets the Smart Analog hardware.
Parameters	None
Global variables	None
Return value	0U: Successful 1U: Failed

(20) **R_SmartAnalogMeasureControlInit**

Declaration	void R_SmartAnalogMeasureControlInit (void)
Overview	Smart Analog initialization function - Initializes global variables for controlling Smart Analog
Parameters	None
Global variables	g_amp_table Table for storing the gain and offset voltage in Smart Analog IC 300
Return value	None

(21) **R_CalcAverageExceptMinMax**

Declaration	uint16_t R_CalcAverageExceptMinMax (const uint16_t * p_buf, uint8_t buf_size)
Overview	A/D-converted value averaging function - Averages the obtained A/D-converted values excluding the minimum and maximum values
Parameters	p_buf Start address of the buffer in which the obtained A/D-converted values are stored buf_size p_buf size (Number of averaging times)
Global Variables	None
Return value	avg_tmp Average of A/D-converted values stored in p_buf[0] to p_buf[buf_size 1] excluding the minimum and maximum values

(22) **R_SmartAnalogRegBufSearch**

Declaration	uint8_t R_SmartAnalogRegBufSearch (sa_ctl_set_data_t * p_sa_buf, uint8_t target_addr)
Overview	Smart Analog register search function - Searches for the Smart Analog address in sa_ctl_set_data_t buffers and returns the index stored.
Parameters	p_sa_buf Start address of the sa_ctl_set_data_t buffer subject to search target_addr Smart Analog register address to be searched for
Global variables	None
Return value	reg_cnt Array number of the specified address (0xFF is returned if the address is not fount)

(23) **R_SmartAnalogRegBufUpdate**

Declaration	uint8_t R_SmartAnalogRegBufUpdate (sa_ctl_set_data_t * p_sa_buf, uint8_t target_addr ,uint8_t set_data)
Overview	Smart Analog setting change function - Searches for the address of the register by using the Smart Analog register search function and rewrites the settings of the found register.
Parameters	p_sa_buf Start address of the sa_ctl_set_data_t buffer subject to search target_addr Address of the Smart Analog register to be rewritten set_data Value to be overwritten to the Smart Analog register
Global variables	None
Return value	ret 0U: Successful 1U: Error

(24) **R_GetTickCount**

Declaration	uint32_t R_GetTickCount (void)
Overview	Timer count acquisition function - Acquires the current value of the free-running timer.
Parameters	None
Global variables	gs_freerun_timer Free-running timer counter variable
Return value	gs_freerun_timer.tdata.timer32bit

(25) **R_CmpTickCount**

Declaration	int32_t R_CmpTickCount (uint32_t src)
Overview	Returns the result of the comparison with the current time.
Parameters	src compare source time
Global Variables	None
Return value	int32_t 0> :Does not exceed. or 0<= :Is over

(26) **R_TAU0_TDR05_set**

Declaration	void R_TAU0_TDR05_set (uint16_t set_data)
Overview	Timer data register 05 change function
Parameters	set_data Counter value to be set
Global Variables	None
Return value	None

(27) **r_tau0_channel5_interrupt**

Declaration	__interrupt static void r_tau0_channel5_interrupt (void)
Overview	INTTM05 interrupt request
Parameters	None
Global variables	gs_freerun_timer Free-running timer counter variable
Return value	None

(28) **r_csi21_interrupt**

Declaration	__interrupt static void r_csi21_interrupt (void)
Overview	An INTCSI21 interrupt service routine
Parameters	None
Global variables	None
Return value	None

(29) **r_csi21_callback_error**

Declaration	static void r_csi21_callback_error (uint8_t err_type)
Overview	A callback function used when a CSI21 reception error occurs
Parameters	err_type error type value
Global Variables	g_csi21_overrun_flag csi21 overrun flag
Return value	None

(30) **r_csi21_callback_receiveend**

Declaration	static void r_csi21_callback_receiveend (void)
Overview	A callback function used when CSI21 finishes reception.
Parameters	None
Global variables	None
Return value	None

(31) **R_SelfCalibrationFlagCheck**

Declaration	uint8_t R_SelfCalibrationFlagCheck (void)
Overview	Auto calibration start trigger flag check function - Returns the auto calibration start trigger flag.
Parameters	None
Global variables	gs_calibration_operation Auto calibration start trigger flag
Return value	ret 0U: Auto calibration start trigger not issued 1U: Auto calibration start trigger issued

(32) **R_SelfCalibrationFlagClear**

Declaration	void R_SelfCalibrationFlagClear (void)
Overview	Auto calibration start trigger flag clear function
Parameters	None
Global variables	gs_calibration_operation Auto calibration start trigger flag
Return value	None

(33) **r_rtc_interrupt**

Declaration	__interrupt static void r_rtc_interrupt (void)
Overview	INTRTC interrupt function
Parameters	None
Global variables	None
Return value	None

(34) **r_rtc_callback_constperiod**

Declaration	static void r_rtc_callback_constperiod (void)
Overview	Real-time clock call back
Parameters	None
Global variables	None
Return value	None

(35) **r_rtc_interrupt_counter**

Declaration	static void r_rtc_interrupt_counter (void)
Overview	Auto calibration timing generation function - Counts the number of calls each time this function is called. - Sets the auto calibration start trigger flag when this function is called the specified number of times.
Parameters	None
Global variables	gs_calibration_operation Auto calibration start trigger flag
Return value	None

(36) **R_DMAC1_ADC_Create**

Declaration	void R_DMAC1_ADC_Create (uint16_t txnum, uint16_t * p_sendbuf)
Overview	DMA1 setting function - Sets DMA1 to be used for A/D conversion result transfer.
Parameters	txnum Number of transferred bytes p_sendbuf Pointer to the transfer buffer
Global Variables	None
Return value	None

(37) **r_dmac1_interrupt**

Declaration	__interrupt static void r_dmac1_interrupt (void)
Overview	INTDMA1 interrupt function
Parameters	None
Global variables	None
Return value	None

(38) **R_ADC_EndFlagCheck**

Declaration	uint8_t R_ADC_EndFlagCheck (void)
Overview	A/D conversion end flag check function
Parameters	None
Global variables	gs_adc_end_flag A/D conversion status flag
Return value	ret 0U: Not completed 1U: Completed

(39) **R_ADC_EndFlagSet**

Declaration	void R_ADC_EndFlagSet(void)
Overview	A/D conversion end flag set function
Parameters	None
Global variables	gs_adc_end_flag A/D conversion status flag
Return value	None

(40) **R_ADC_EndFlagClear**

Declaration	void R_ADC_EndFlagClear(void)
Overview	AD conversion end flag clear function
Parameters	None
Global variables	gs_adc_end_flag A/D conversion status flag
Return value	None

(41) **R_ADC_MaskStart**

Declaration	void R_ADC_MaskStart(void)
Overview	Starts A/D conversion (disables interrupts).
Parameters	None
Global variables	None
Return value	None

(42) **R_ADC_ChangeSoftTrigger**

Declaration	void R_ADC_ChangeSoftTrigger (void)
Overview	A/D conversion trigger software change function - Changes the A/D conversion trigger to the software trigger mode or sequential conversion mode
Parameters	None
Global variables	None
Return value	None

(43) **R_ADC_StartSetting**

Declaration	void R_ADC_StartSetting (uint16_t * p_dma_buf, uint8_t dma_buf_size)
Overview	A/D conversion start setting function - Specifies the settings used to start A/D conversion.
Parameters	p_dma_buf DMA transfer destination address p_dma_buf_size Number of transferred bytes by using DMA
Global Variables	None
Return value	None

(44) **R_ADC_EndSetting**

Declaration	void R_ADC_EndSetting(void)
Overview	A/D conversion end setting function - Performs A/D conversion end processing.
Parameters	None
Global variables	None
Return value	None

(45) **R_ADC_ChannelChange**

Declaration	void R_ADC_ChannelChange (ad_channel_t adc_ch)
Overview	A/D conversion channel change function - Changes the analog input channel (ANIxx pin) on which to perform A/D conversion
Parameters	adc_ch Channel (ANIxx) to be set to ADS
Global Variables	None
Return value	None

6.3 Global Variables

Table 6-3 Global Variables

Data type	Variable Name	Description	Function used
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	Not used
unsigned short	g_csi21_rx_count	csi21 receive data count	Not used
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	Used to store the settings for D/A converter output voltage measurement	R_SelfCalibrationInit
struct	g_self_cal_12	Used to store the settings for configurable amplifier Ch1 output voltage measurement	R_SelfCalibrationInit
struct	g_self_cal_21	Used to store the settings for gain measurement	R_SelfCalibrationInit
struct *([3])	gp_self_cal_setting_data	Pointer to the array that stores values set for auto calibration	R_SelfCalibrationInit
struct [4]	g_ic300_conf_amp_default	Used to store the initial gain setting when a single Smart Analog IC 300 channel is used	R_SmartAnalogMeasureControlInit
struct	g_ic300_inst_amp_default	Used to store the initial gain setting when multiple Smart Analog IC 300 channels are used	R_SmartAnalogMeasureControlInit
struct	g_amp_table	Table for storing the gain and offset voltage in Smart Analog IC 300	R_SmartAnalogMeasureControlInit R_SelfCalibAmp1DiffOffsetCalc R_SelfCalibAmp1DiffGainAverage

6.4 Smart Analog IC 300 register values

This section describes the settings of the Smart Analog IC 300 SPI control registers used in this application note. This section omits descriptions of the SPI control registers not used in this application note. (They are used with their default values.)

Caution For how to configure the SPI registers, see *RAA730300 Monolithic Programmable Analog IC Datasheet*.

(1) Configuration register 1 (CONFIG1)

This register is used to turn on or off the SW11, SW12, and SW13 switches of configurable amplifier channels Ch1 and Ch2.

Address: 00H After reset: 88H R/W Set value: **H

Symbol	7	6	5	4	3	2	1	0
CONFIG1	SW10	SW11	SW12	SW13	SW20	SW21	SW22	SW23
Setting	1	0	0	*	0	0	0	0

Remark *: The value depends on what is measured (gain error or input-referred offset voltage). For details, see Table 5-1 or Table 5-2.

(2) Configuration register 2 (CONFIG2)

This register is used to turn on or off the switches of configurable amplifier channels Ch1 to Ch3.

Address: 01H After reset: 80H R/W Set value: 8*H

Symbol	7	6	5	4	3	2	1	0
CONFIG2	SW30	SW31	SW32	SW33	0	SW02	SW01	SW00
Setting	1	0	0	0	0	0	*	*

Remark *: The value depends on what is measured (gain error or input-referred offset voltage). For details, see Table 5-1 or Table 5-2.

(3) MPX setting register 1 (MPX1)

This register is used to select the signal input to configurable amplifier channels Ch1 and Ch2.

Address: 03H After reset: 00H R/W Set value: **H

Symbol	7	6	5	4	3	2	1	0
MPX1	MPX11	MPX10	MPX21	MPX20	MPX31	MPX30	MPX41	MPX40
Setting	*	*	*	0	*	0	0	0

Remark *: The value depends on what is measured (gain error or input-referred offset voltage). For details, see Table 5-1 or Table 5-2.

(4) Gain control register 1 (GC1)

This register is used to select the gain for configurable amplifier Ch1.

Address: 06H After reset: 00H R/W Set value: **H

Symbol	7	6	5	4	3	2	1	0
GC1	0	0	0	AMPG14	AMPG13	AMPG12	AMPG11	AMPG10
Setting	**0**	**0**	**0**	*	**0**	**0**	**0**	*

Remark *: The value depends on what is measured (gain error or input-referred offset voltage). For details, see Table 5-1 or Table 5-2.

(5) Power control register 1 (PC1)

This register is used to enable or disable operation of the configurable amplifier and D/A converter.

Address: 11H After reset: 00H R/W Set value: 1*H

Symbol	7	6	5	4	3	2	1	0
PC1	DAC4OF	DAC3OF	DAC2OF	DAC1OF	AMP4OF	AMP3OF	AMP2OF	AMP1OF
Setting	**0**	**0**	**0**	**1**	**0**	**0**	**0**	*

Remark *: The value depends on what is measured (gain error or input-referred offset voltage). For details, see Table 5-1 or Table 5-2.

(6) Power control register 2 (PC2)

This register is used to enable or disable operation of D/A converters.

Address: 12H After reset: 00H R/W Set value: **H

Symbol	7	6	5	4	3	2	1	0
PC2	DAC7OF	DAC6OF	DAC5OF	AMP5OF	LPFOF	HPFOF	LDOOF	TEMPOF
Setting	**0**	*	*	**0**	**0**	**0**	**0**	**0**

Remark *: The value depends on what is measured (gain error or input-referred offset voltage). For details, see Table 5-1 or Table 5-2.

6.5 Code generator settings

The following table only shows the code generator settings that must be specified in this application note. (The modules not used are omitted.)

Table 6-4 Code generator settings (1)

Module	Macro	Sub	Setting	Status
Clock generator				Used
	CGC			Used
			Pin assignment: PIOR0 = 1	Not used
			Pin assignment: PIOR0 = 1	Not used
			Operating mode	High-speed main mode 2.7 V ≤ VDD ≤ 3.6 V
			EVDD	2.7 V ≤ EVDD ≤ 3.6 V
			Main system clock (fMAIN)	High-speed on-chip oscillator clock (fIH)
			fIH operation	Used
			fIH frequency	32 MHz
			fMX operation	Not used
			fSUB operation	Used
			Subsystem clock (fSUB)	XT1 oscillation (fXT)
			fSUB frequency	32.768 kHz
			XT1 oscillator mode	Low-power oscillation
			Clock supply in STOP/HALT mode	Clock supplied
			fIL frequency	1 kHz
			RTC/interval timer operating clock	32.768 (fSUB) (kHz)
			CPU and peripheral clock (fCLK)	32,000 (fIH) (kHz)
			On-chip debug operation	Not used
			Security ID	Used
			Security ID	0x00000000000000000000
			Outputting a reset source check function	Not used
			Flash memory CRC calculation unit	Not used
			Illegal memory access detection	Not used
			RAM guard	Not used
			Port register guard	Not used
			Interrupt register guard	Not used
			Chip state control register guard	Not used

Table 6-5 Code generator settings (2)

Module	Macro	Sub	Setting	Status
Serial interface				Used
	SAU1			Used
		Channel 1		
			Channel 1	CSI21 (transmission/reception)
			Transfer mode	Successive transfer
			Data length	8 bits
			Transfer direction	MSB
			Data transmission/reception timing	Type 1
			Clock mode	Internal clock (master)
			Baud rate	500,000 bps (actual value: 500,000)
			Priority level of communication completion interrupt (INTCSI21)	Low
			Transmission completion (callback setting)	Not used
			Reception completion (callback setting)	Used
			Overrun error (callback setting)	Used
A/D converter				Used
	ADC			Used
			A/D converter operation	Used
			Comparator operation	Enabled
			Resolution	12 bits
			VREF(+)	AVDD
			VREF(-)	AVSS
			Trigger mode	Hardware trigger no-wait mode
			Hardware trigger no-wait mode	INTTM01 (specify INTTM01.)
			Operating mode setting	Continuous select mode
			Analog input pins ANI0 to ANI12	ANI0 to ANI12
			Analog input pins ANI16 to ANI30	ANI16, ANI17, ANI19
			Conversion start channel	ANI2
			Reference voltage	2.7 V ≤ AVDD ≤ 3.6 V
			Conversion time mode	Normal 1
			Conversion time	3.375 (108/fCLK) (μs)
			Upper/lower limit for conversion result values	Generates an interrupt request signal (INTAD) at ADLL ≤ ADCRH ≤ ADUL
			Upper limit (ADUL)	255
			Upper limit (ADUL)	0
			A/D converter interrupt (INTAD)	Used
			Priority	Low

Table 6-6 Code generator settings (3)

Module	Macro	Sub	Setting	Status
Timer				Used
	TAU0			Used
		Channel 5		
			Channel 5	Interval timer
			Interval (16 bits)	100 μs (actual value: 100)
			Generating an interrupt when counting is started	Not used
			Generating an interrupt (INTTM05) when counting by timer channel 5 ends	Used
			Priority (INTTM05)	Low
Real-time clock				Used
	RTC			Used
			Real-time clock operation	Used
			Time notation	24-hour clock
			Real-time clock initial value	Not used
			RTC1HZ pin output (1 Hz)	Not used
			Alarm detection	Not used
			Correction	Not used
			Constant-period interrupt (INTRTC)	Used, once per second (in synchronization with counting up the number of seconds)
			Priority (INTRTC)	Low
DMA controller				Used
	DMA0			Not used
	DMA1			Used
			DMA operation	Used
			Transfer direction	SFR → internal RAM
			Transfer data size	16 bits
			SFR address	ADCR - 0x000fff1e
			RAM address	0xfef00
			Number of transfers	64
			Trigger signal	INTAD
			DMA1 transmission completion interrupt (INTDMA1)	Used
			Priority	Low

6.6 Flowcharts

Figure 6-1 shows an overview of the processing flow used in this application note.

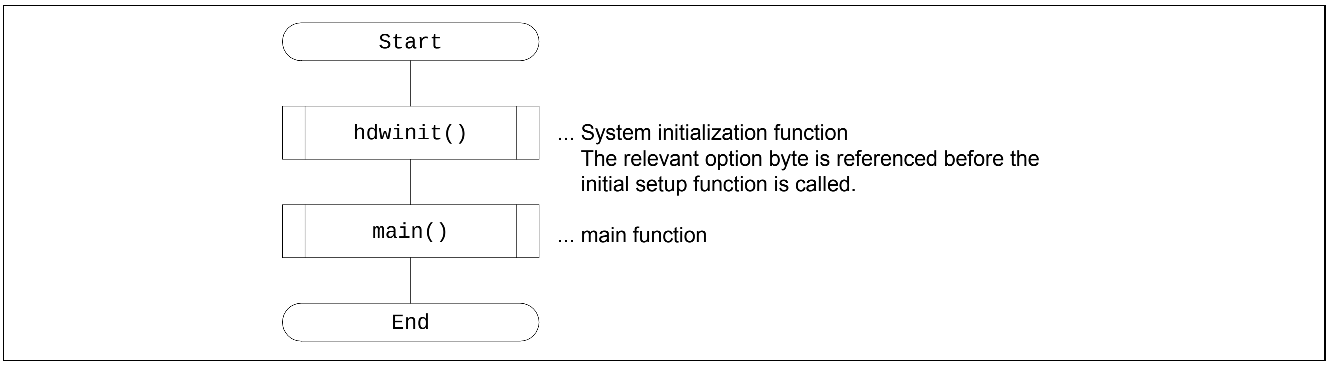


Figure 6-1 Flowcharts

(1) System initialization function (hdwinit)

Figure 6-2 shows the processing flow of the system initialization function (hdwinit).

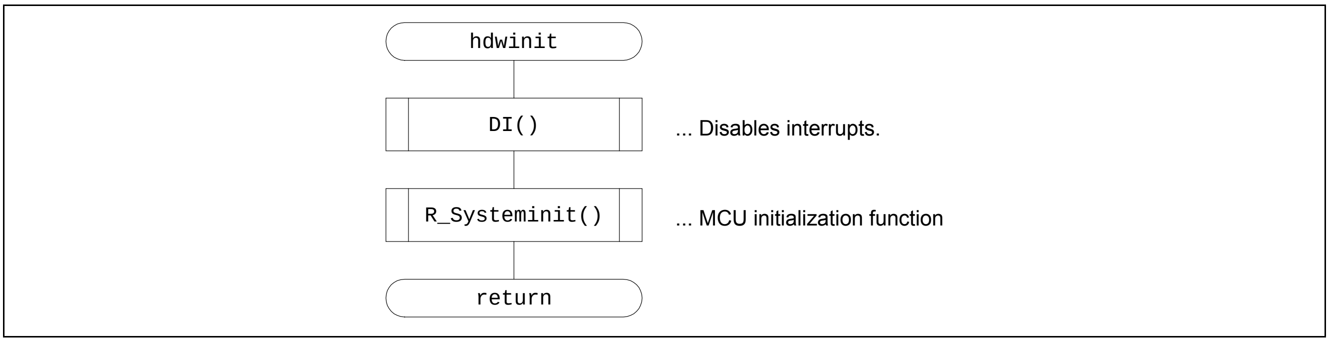


Figure 6-2 hdwinit function

(2) **MCU initialization function (R_Systeminit)**

Figure 6-3 shows the processing flow of the MCU initialization function (R_Systeminit).

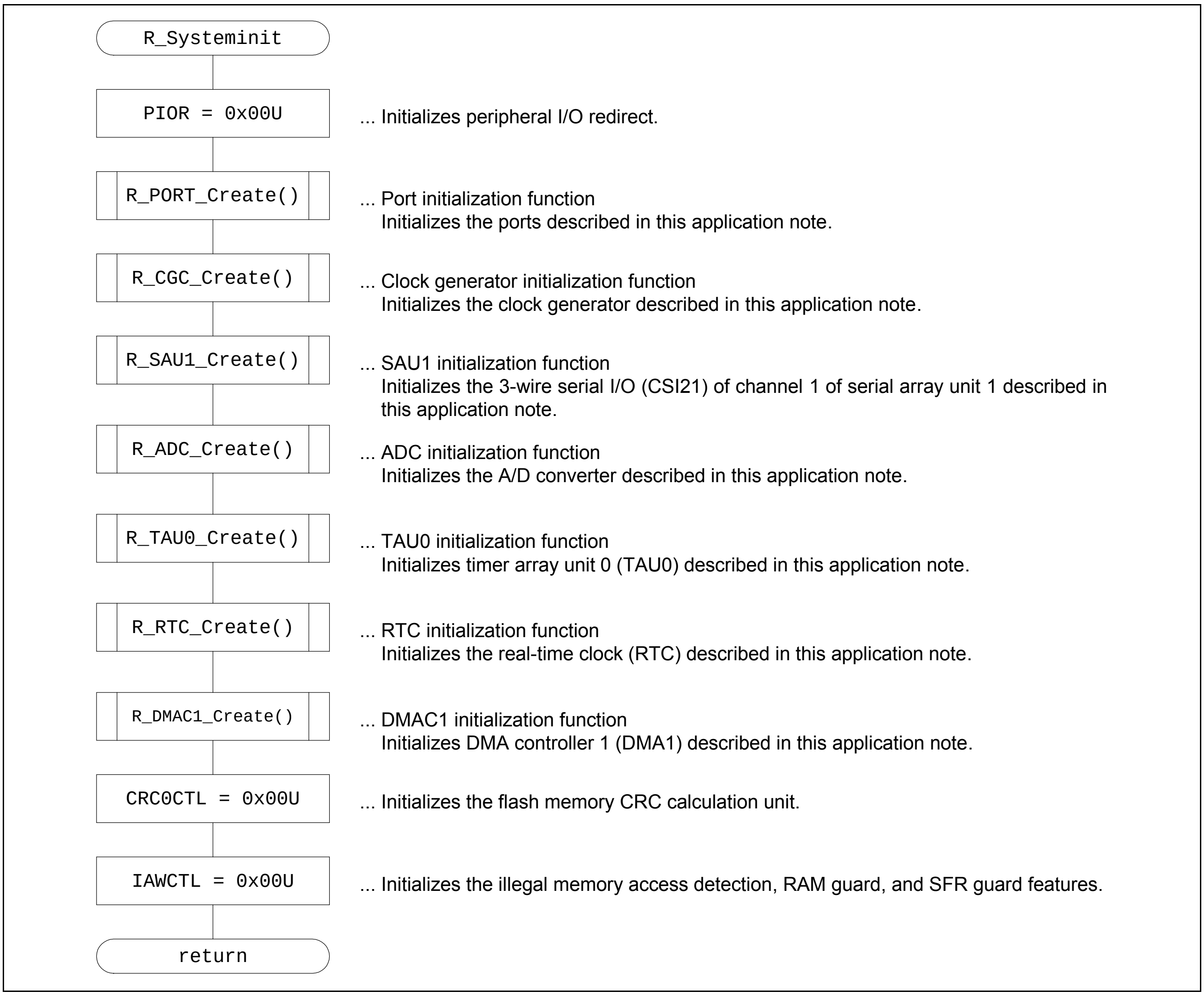


Figure 6-3 R_systeminit function

(3) main function

Figure 6-4 shows the processing flow of the main function.

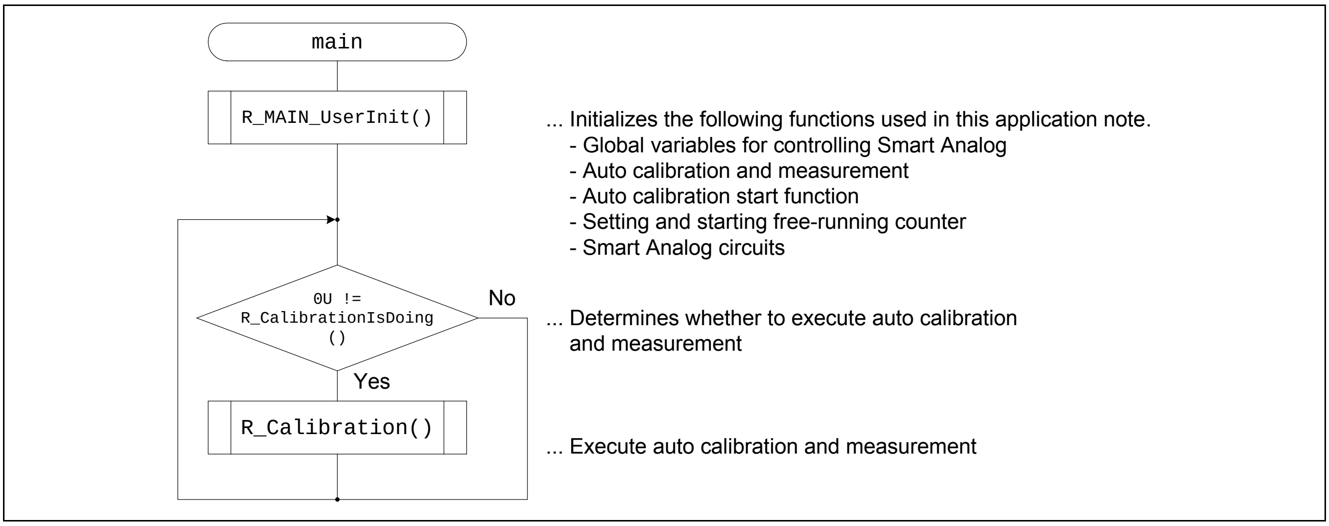


Figure 6-4 main function

(4) User interface initialization function

Figure 6-5 shows User interface initialization function.

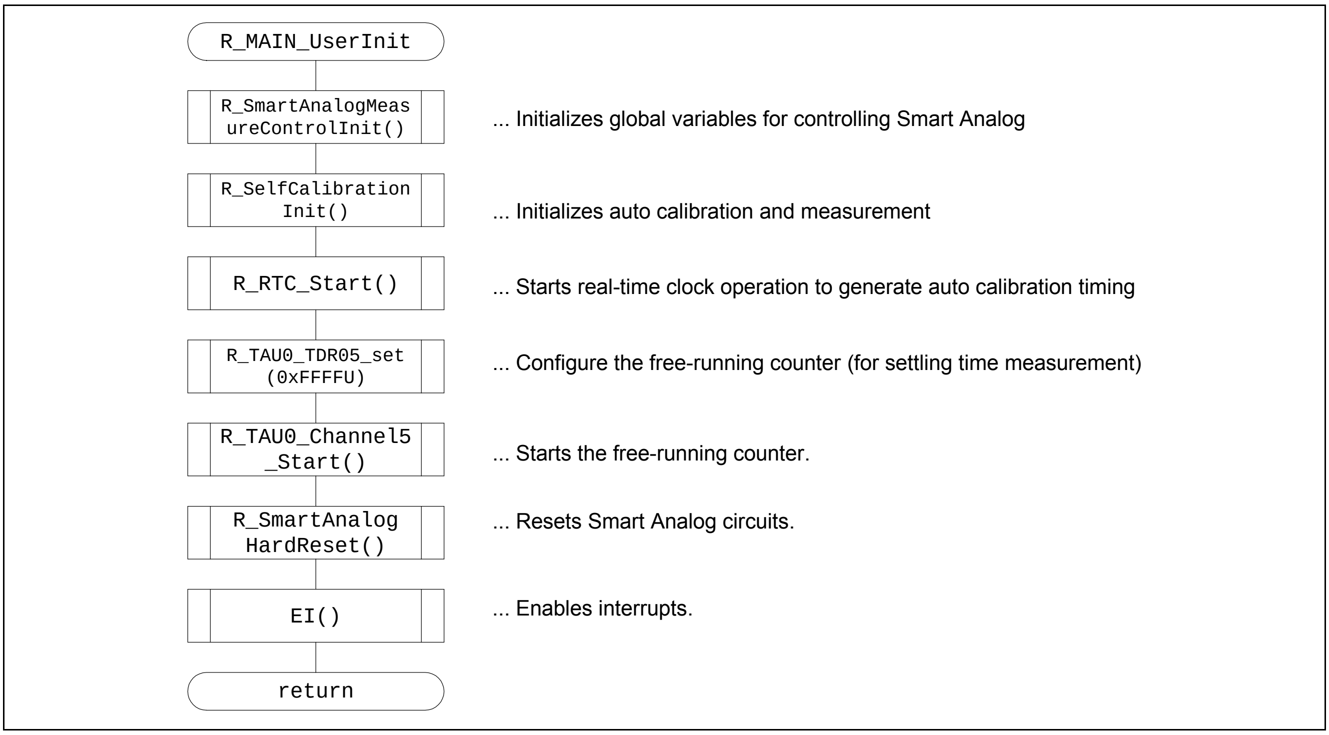


Figure 6-5 User interface initialization function

(5) **Calibration trigger judge function**

Figure 6-6 shows the calibration trigger judge function.

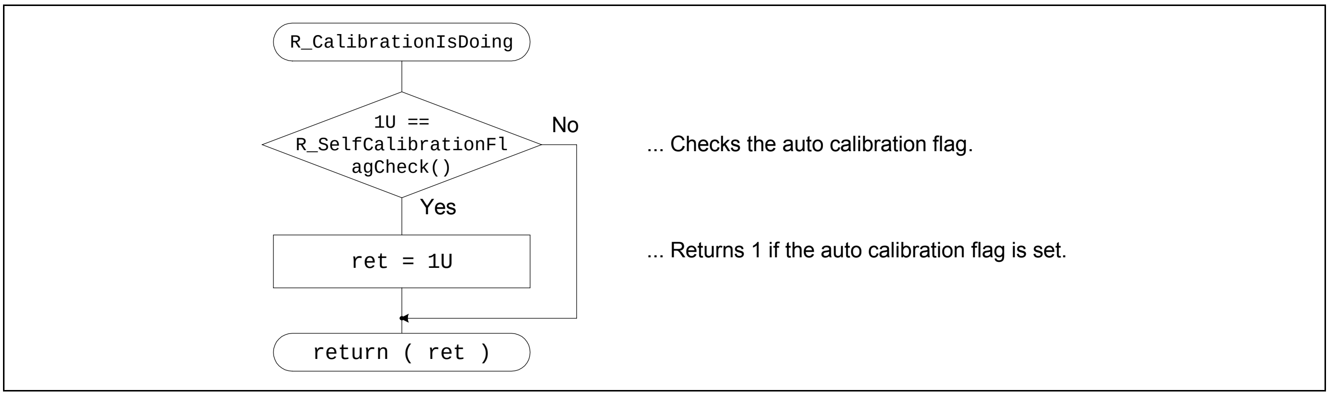


Figure 6-6 Calibration trigger judge function

(6) **Auto calibration start trigger flag check function**

Figure 6-7 shows Auto calibration start trigger flag check function.

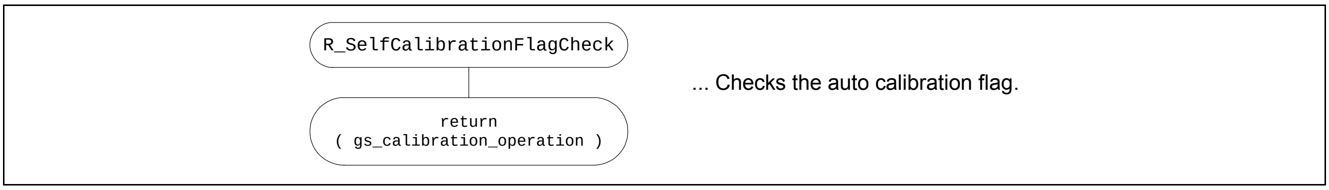


Figure 6-7 Auto calibration start trigger flag check function

(7) **Auto calibration timing generation function**

Figure 6-8 shows Auto calibration timing generation function.

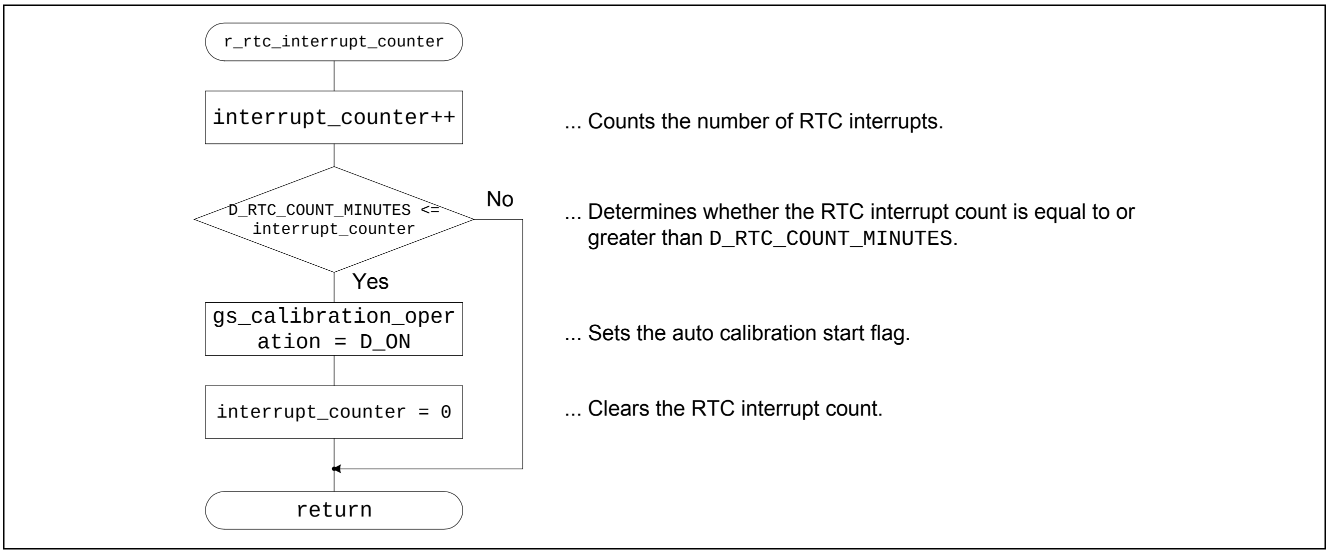


Figure 6-8 Auto calibration timing generation function

(8) RTC interrupt function

Figure 6-9 shows RTC interrupt function.

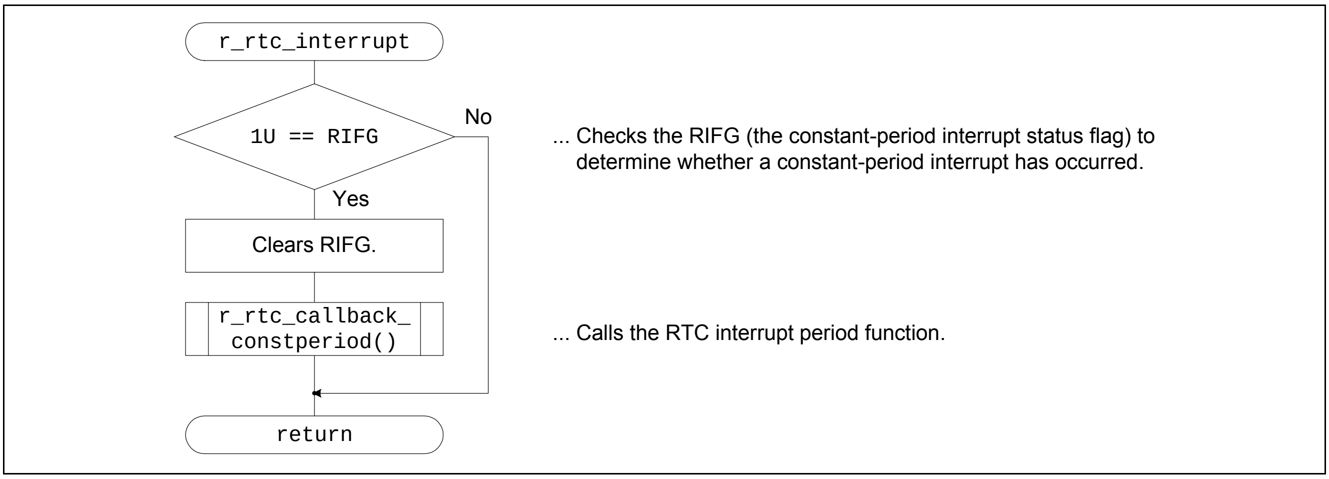


Figure 6-9 RTC interrupt function

(9) RTC interrupt handler function

Figure 6-10 shows RTC interrupt handler function.

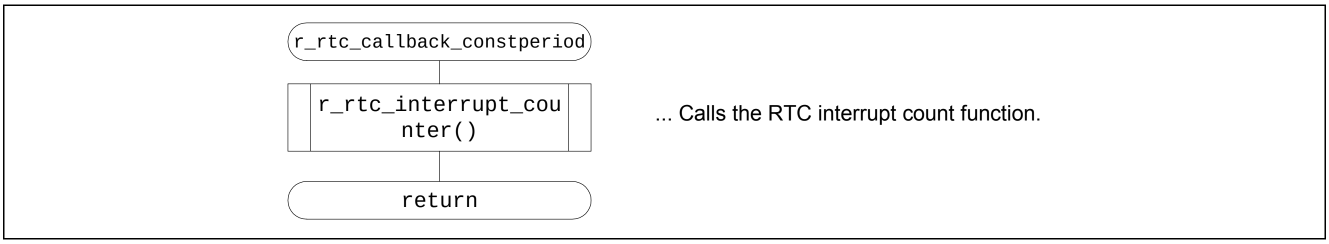


Figure 6-10 RTC interrupt handler function

(10) Auto calibration start trigger flag clear function

Figure 6-11 shows Auto calibration start trigger flag clear function.

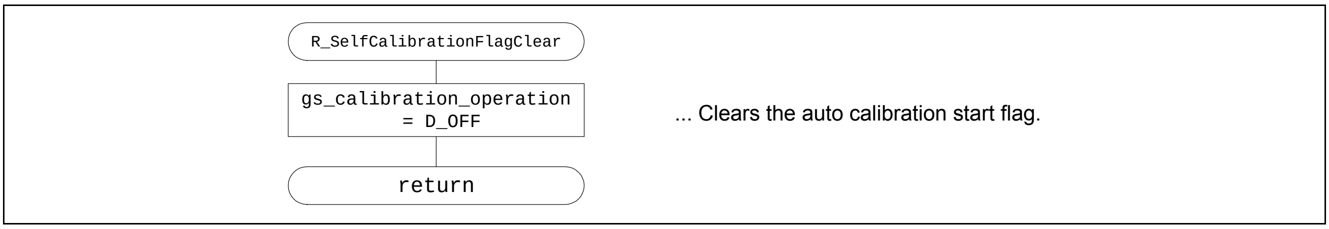


Figure 6-11 Auto calibration start trigger flag clear function

(11) Calibration control function

Figure 6-12 shows Calibration control function. Judgment of calibration_state is performed by a state machine. For details about each state, see the state transition diagram in Figure 6-13.

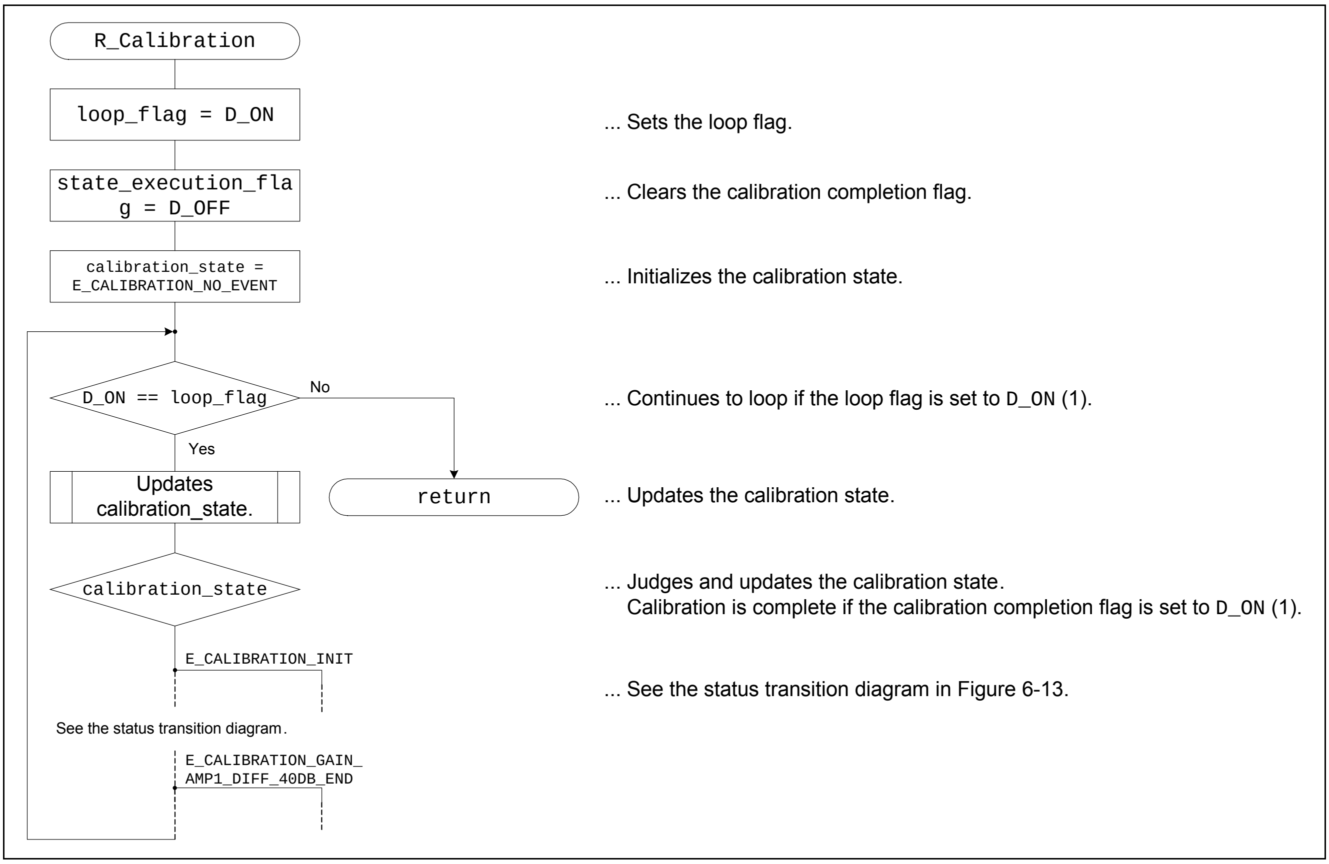


Figure 6-12 Calibration control function

How to use the sample code described in this application note is shown below. Ten processes are used to perform auto calibration measurement. In this application note, initialization and end processing are added to these ten processes and divide the processes into 12 states to perform measurement.

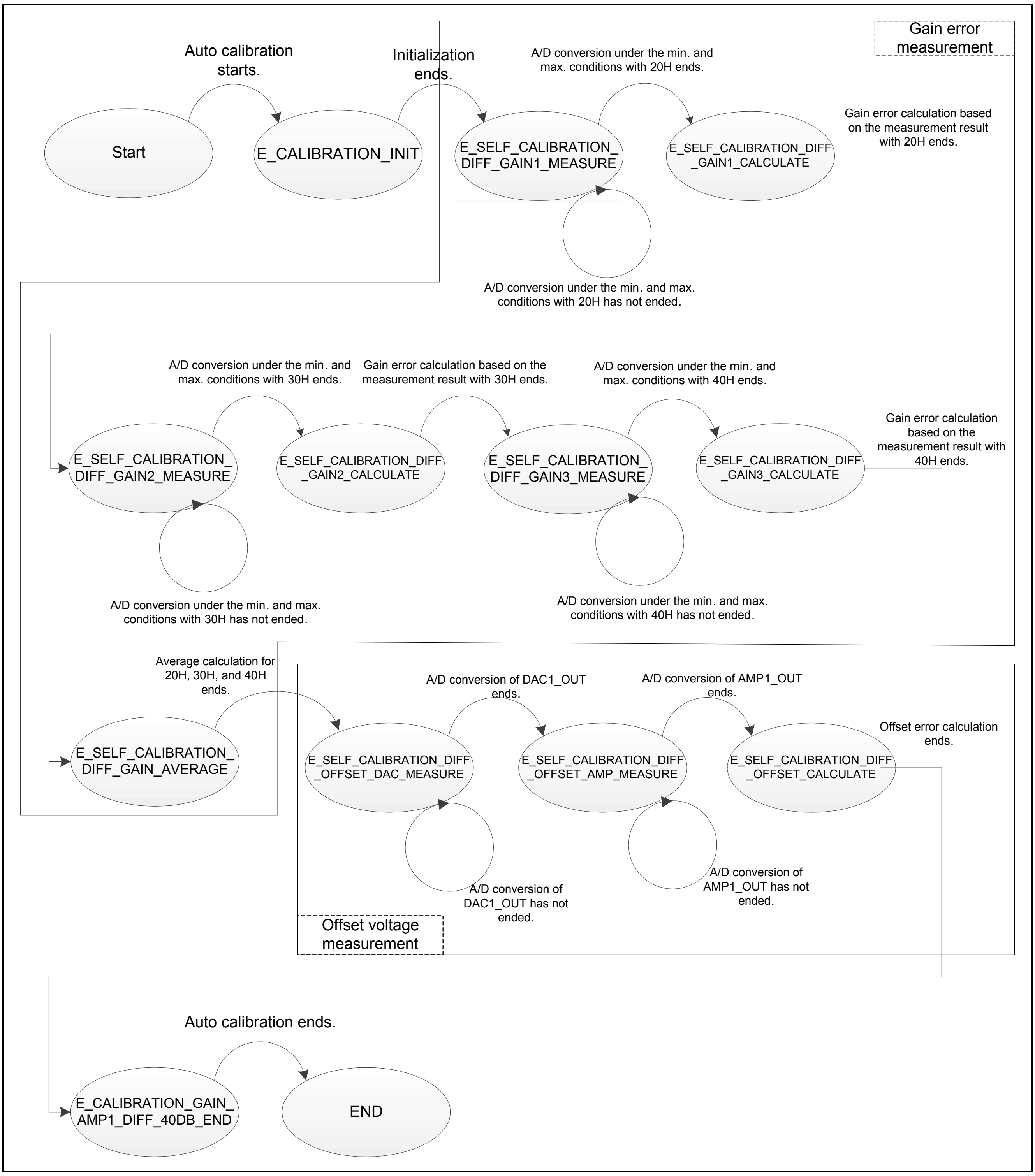


Figure 6-13 State transition during auto calibration measurement

Table 6-7 shows the calibration_state judgement states and the functions that call each state.

Table 6-7 Correspondence between states and functions

State	Overview	Function that calls the process
E_CALIBRATION_INIT	Initialization of auto calibration and measurement	calibration_init
E_SELF_CALIBRATION_DIFF_GAIN1_MEASURE	Gain error measurement First time	R_SelfCalibAmp1DiffGainMeasure
E_SELF_CALIBRATION_DIFF_GAIN1_CALCULATE	Gain error calculation First time	R_SelfCalibAmp1DiffGainCalc
E_SELF_CALIBRATION_DIFF_GAIN2_MEASURE	Gain error measurement Second time	R_SelfCalibAmp1DiffGainMeasure
E_SELF_CALIBRATION_DIFF_GAIN2_CALCULATE	Gain error calculation Second time	R_SelfCalibAmp1DiffGainCalc
E_SELF_CALIBRATION_DIFF_GAIN3_MEASURE	Gain error measurement Third time	R_SelfCalibAmp1DiffGainMeasure
E_SELF_CALIBRATION_DIFF_GAIN3_CALCULATE	Gain error calculation Third time	R_SelfCalibAmp1DiffGainCalc
E_SELF_CALIBRATION_DIFF_GAIN_AVERAGE	Gain error average calculation	R_SelfCalibAmp1DiffGainAverage
E_SELF_CALIBRATION_DIFF_OFFSET_DAC_MEASURE	Offset voltage measurement DAC_OUT voltage	R_SelfCalibAmp1DiffOffsetDac
E_SELF_CALIBRATION_DIFF_OFFSET_AMP_MEASURE	Offset voltage measurement AMP_OUT voltage	R_SelfCalibAmp1DiffOffsetAmp
E_SELF_CALIBRATION_DIFF_OFFSET_CALCULATE	Offset voltage calculation	R_SelfCalibAmp1DiffOffsetCalc
E_CALIBRATION_GAIN_AMP1_DIFF_40DB_END	End processing of auto calibration and measurement	R_SelfCalibrationFlagClear

(12) **Calibration initialization function**

Figure 6-14 shows Calibration initialization function.

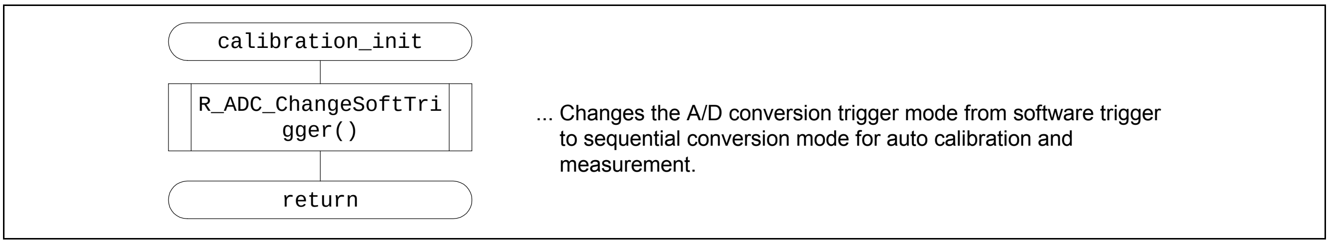


Figure 6-14 Calibration initialization function

(13) Gain measurement function

Figure 6-15 and Figure 6-16 show the gain measurement function.

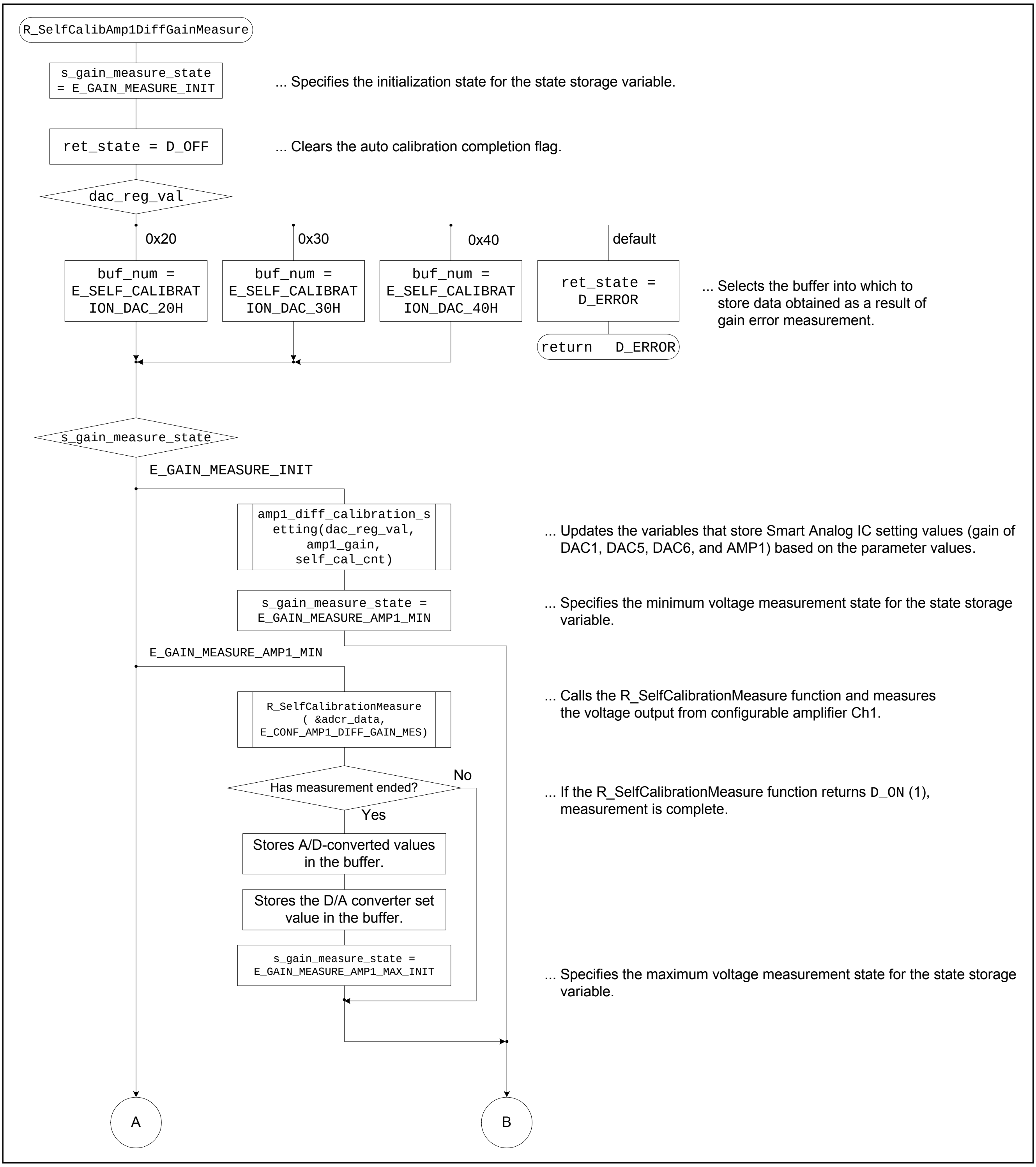


Figure 6-15 Gain measurement function (1)

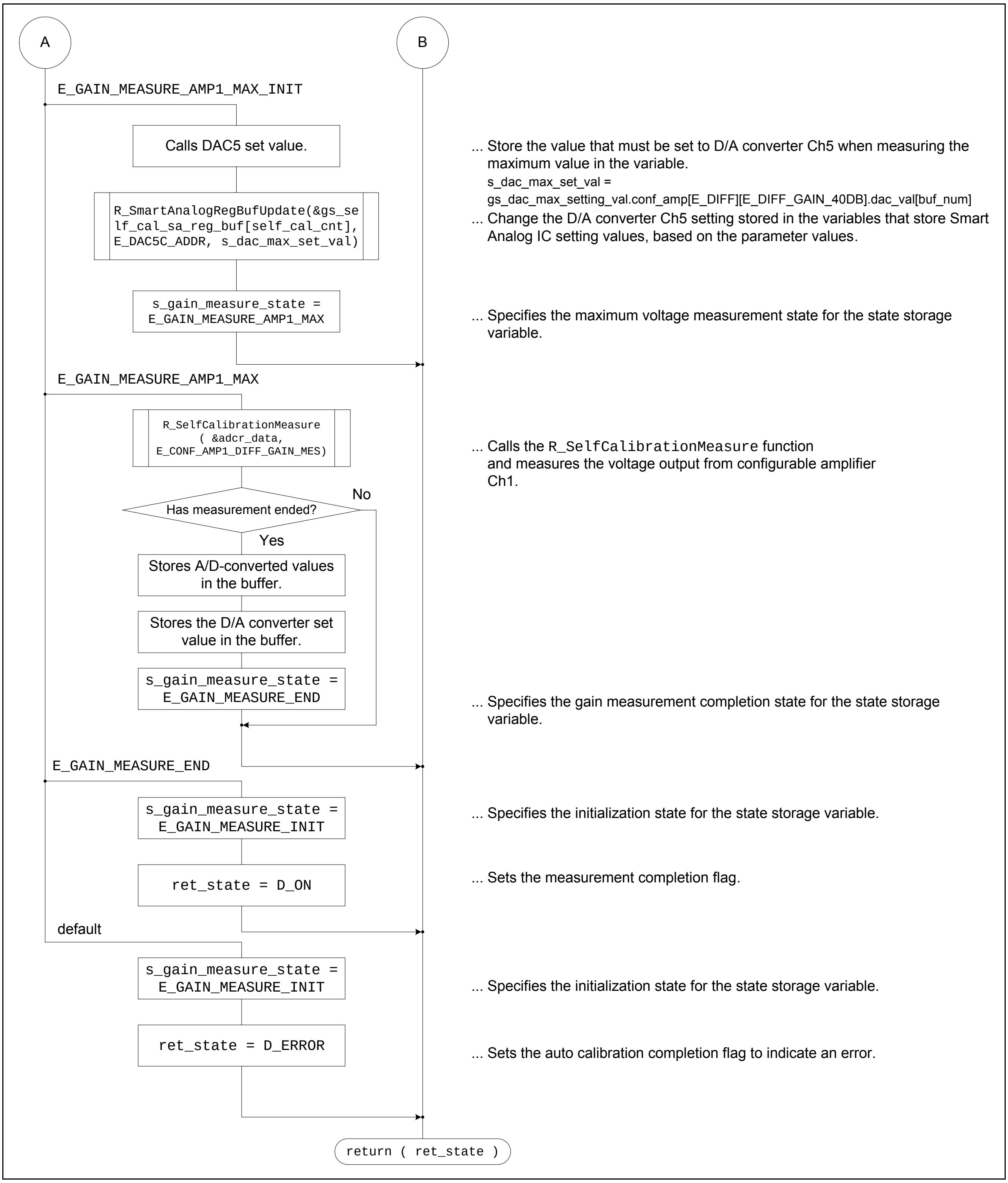


Figure 6-16 Gain measurement function (2)

(14) **A/D conversion control function for auto calibration**

Figure 6-17 shows A/D conversion control function for auto calibration.

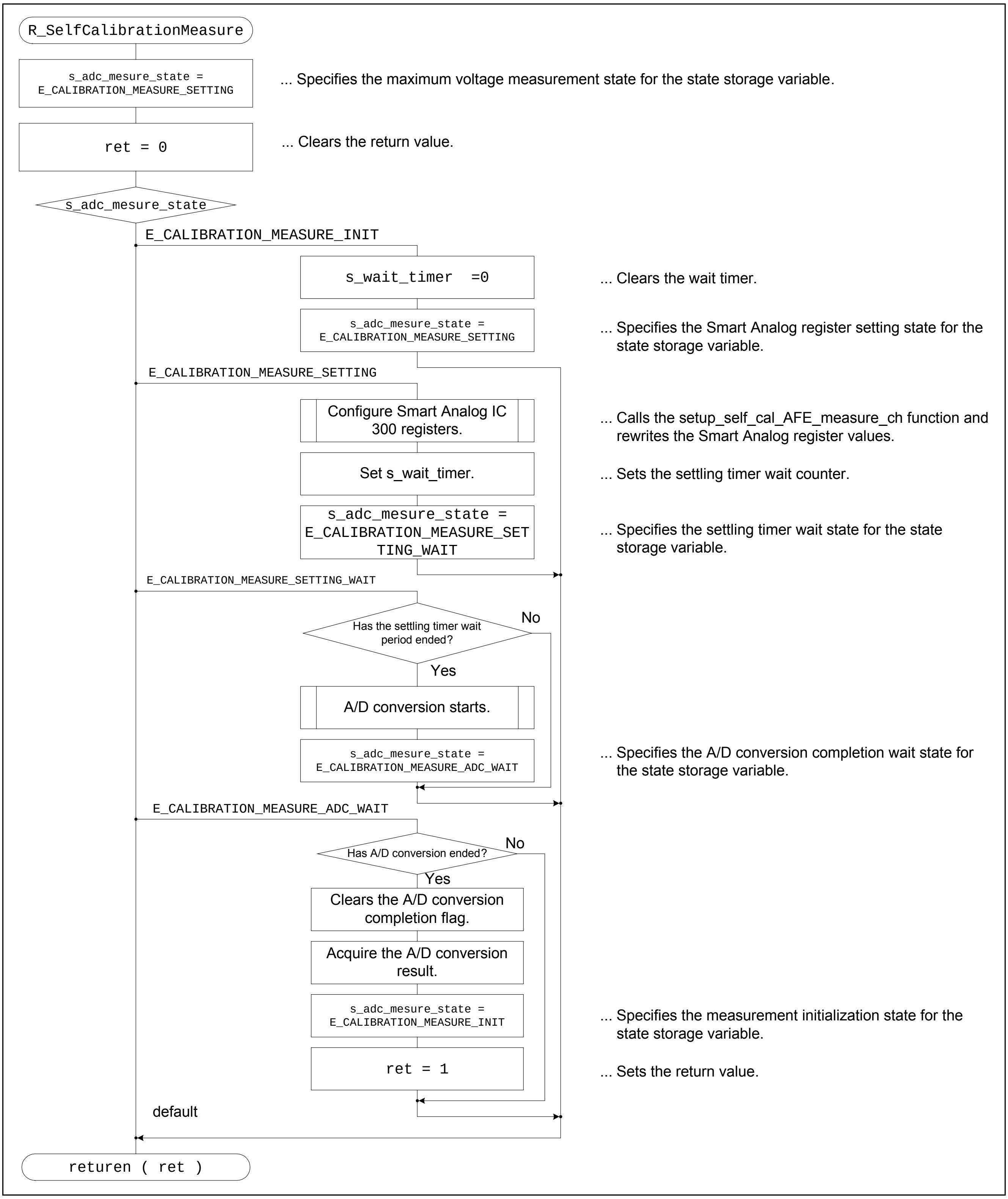


Figure 6-17 A/D conversion control function for auto calibration

(15) **Gain calculation function**

Figure 6-18 shows the gain calculation function.

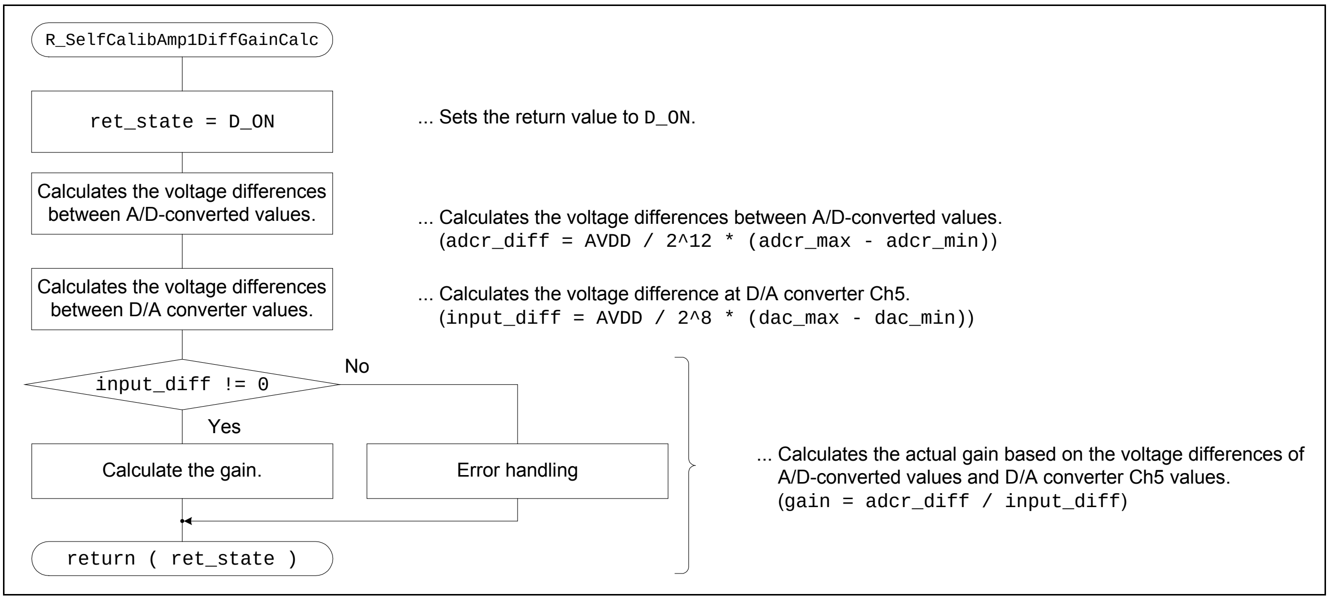


Figure 6-18 Gain calculation function

(16) **Gain averaging function**

Figure 6-19 shows Gain averaging function.

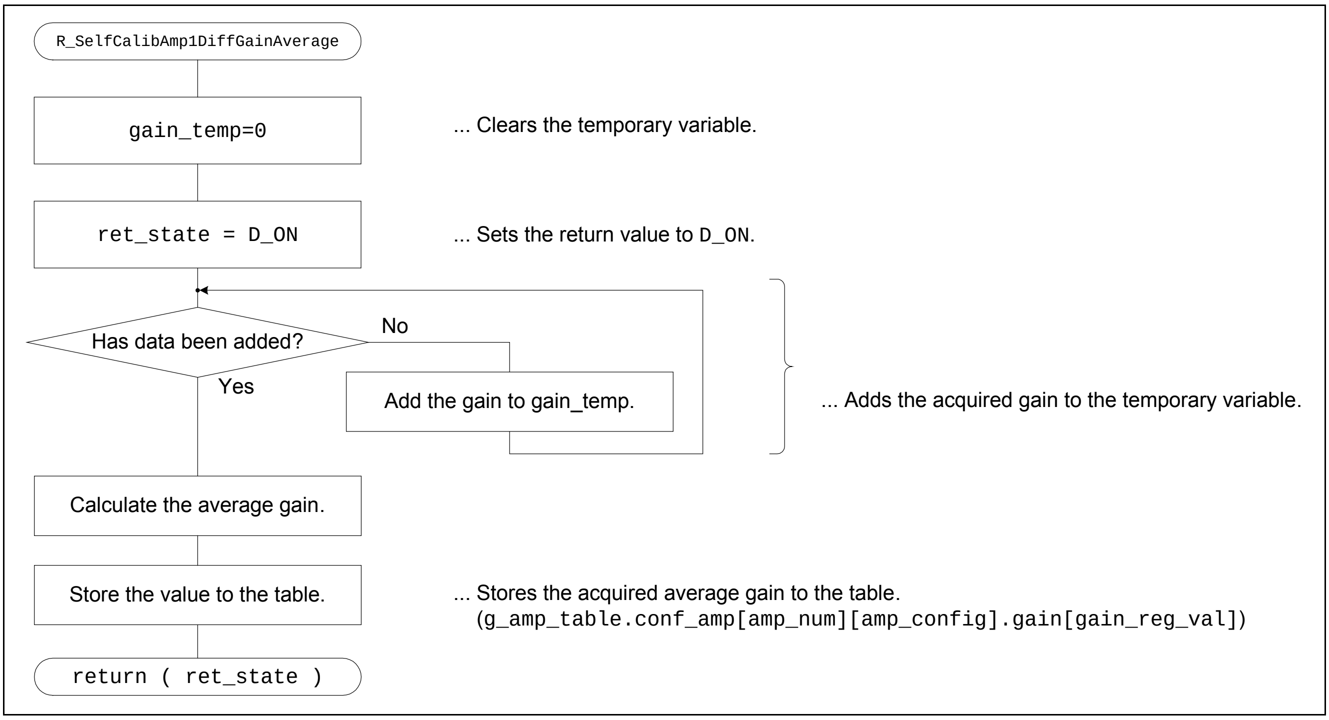


Figure 6-19 Gain averaging function

(17) DAC1_OUT measurement function

Figure 6-20 shows DAC1_OUT measurement function.

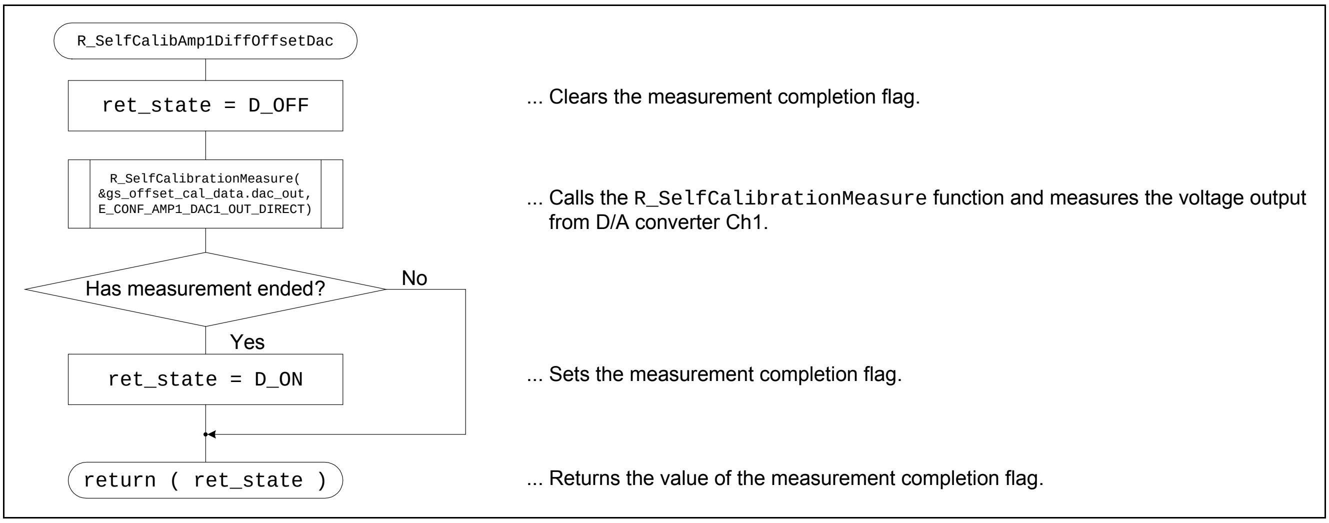


Figure 6-20 DAC1_OUT measurement function

(18) AMP1_OUT measurement function

Figure 6-21 shows AMP1_OUT measurement function.

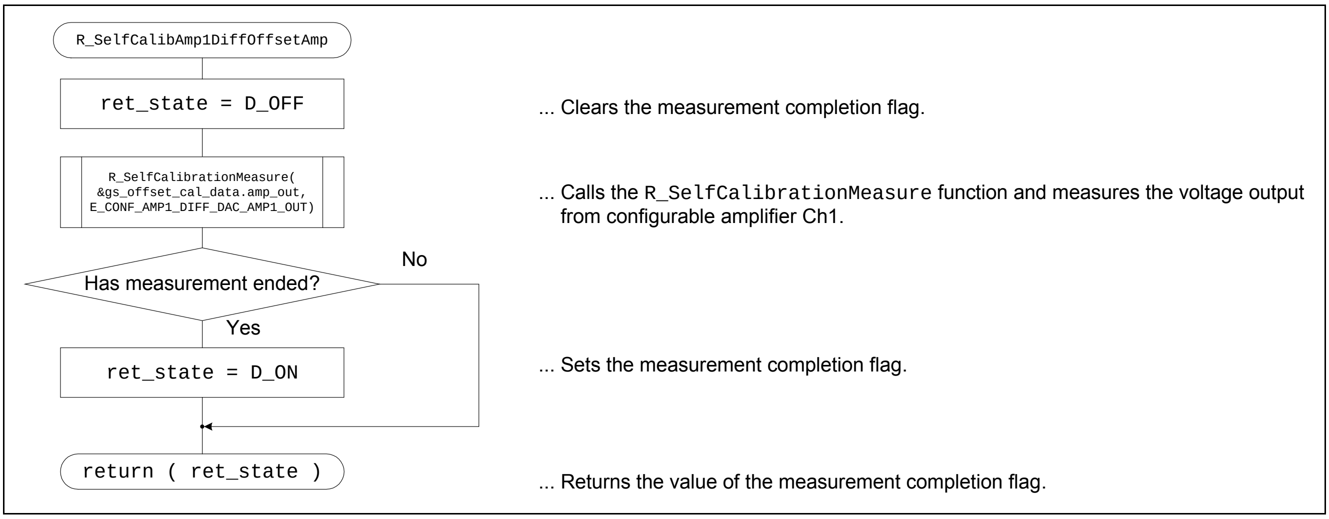


Figure 6-21 AMP1_OUT measurement function

(19) Input conversion offset voltage calculation function

Figure 6-22 shows the input conversion offset voltage calculation function

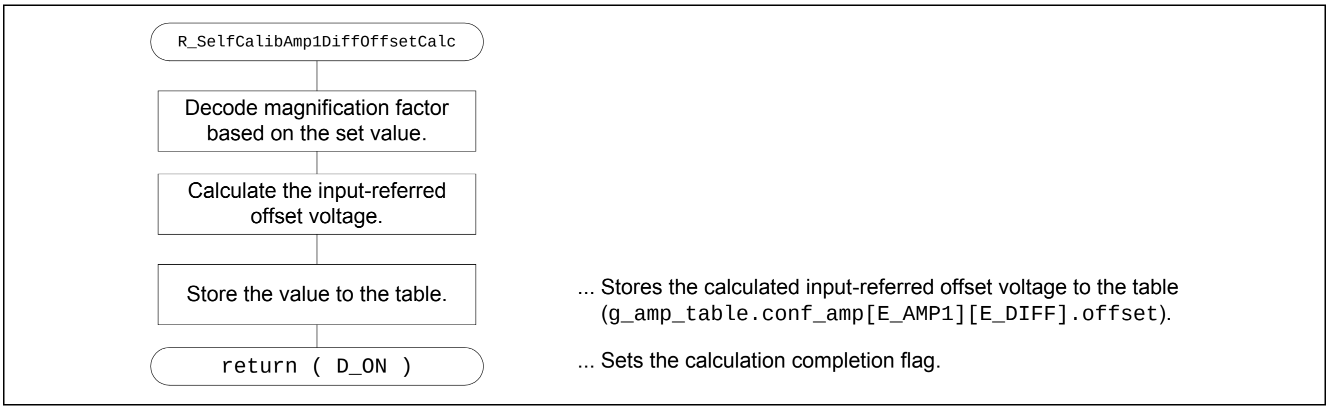


Figure 6-22 Input conversion offset voltage calculation function

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では、2 つのサンプルプロジェクトを用意しています。「selfcalibration」プロジェクトは、自己補正計測のみを実施する基本的なサンプルコードです。 「example」プロジェクトは、自己補正計測を用いた応用例として、自己補正で得られたゲイン・オフセット電圧を用いて 0.1 秒ごとに差動信号の周期測定を測定するサンプルコードです。 なお、どちらのプロジェクトでも自己補正計測では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した時の実ゲインおよび入力換算オフセット電圧の測定を実施しています。

本章では、サンプルコードの自己補正データの取り出し方やコンフィギュラブル・アンプのチャネルや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_single_amp_t inst_amp;
}ic300_amp_table_t ;
```

Table 7-1　　ic300_amp_table_t 構造体のメンバ

Member name	Description
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 　Configurable amplifier channel number 　　0: Configurable amplifier Ch1 　　1: Configurable amplifier Ch2 　　2: Configurable amplifier Ch2 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 ;
```

Table 7-2 ic300_single_amp_t 構造体メンバ

Member name	Description
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 ;
```

Table 7-3 ic300_multi_amp_t 構造体メンバ

Member name	Description
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。

- Gain:g_amp_table.conf_amp[0][2].gain[17]
- Offset voltage: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 file

```
static const sa_ctl_set_data_t g_self_cal_11 =
{
    { D_DAC1_OUT_CH },    //Specify the analog input channel for which to execute
A/D conversion.
    { D_AMP1_GAIN_ADDR },//Specify the address of the gain control register for the
amplifier used.
    { D_AMP1_BIAS_ADDR },//Specify the address of the DAC control register
corresponding to the DAC connected for supplying the bias voltage for the amplifier
used.
    { D_WAIT100US_32MHZ },//Specify the Smart Analog settling time.
    { D_IC300_REG_NUM },    //Number of Smart Analog registers
    {
  { 0x00, 0x80 } ,        //Specify the address and values of Smart Analog IC 30
registers.
  { 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。

R_Calibration function in calibration.c file

```
  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ネルを変更します。

R_Calibration function in calibration.c file

```
state_execution_flag = R_SelfCalibAmp1DiffGainAverage(E_AMP1 , E_DIFF,
amp_gain_val );//Specify the channel of configurable amplifier used.

state_execution_flag = R_SelfCalibAmp1DiffOffsetCalc( E_AMP1 , E_DIFF,
amp_gain_val ); //Specify the channel of configurable amplifier used.
```

(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 },    Specify the analog input channel for which to execute A/D
conversion.
    { D_AMP1_GAIN_ADDR },Specify the address of the gain control register for the
amplifier used.
    { D_AMP1_BIAS_ADDR },Specify the address of the DAC control register corresponding
to the DAC connected for supplying the bias voltage for the amplifier used.
    { D_WAIT100US_32MHZ },Specify the Smart Analog settling time.
    { D_IC300_REG_NUM },    //Number of Smart Analog registers
    {
  { 0x00, 0x80 } ,        Specify the address and values of Smart Analog IC 30
registers.
  { 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 file

```
typedef enum
{
    E_CONF_AMP1_DAC1_OUT_DIRECT = 0U,   //AMP1 差動アンプ用の DAC の直接測定する設定
    E_CONF_AMP1_DIFF_DAC_AMP1_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_AMP1_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_AMP1_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 file

```
R_SelfCalibAmp1DiffGainMeasure
R_SelfCalibAmp1DiffGainCalc
R_SelfCalibAmp1DiffGainMeasure
R_SelfCalibAmp1DiffGainAverage
R_SelfCalibAmp1DiffOffsetDac
R_SelfCalibAmp1DiffOffsetAmp
R_SelfCalibAmp1DiffOffsetCalc
```

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

calibaration_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;
```

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00		—	First edition issued.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 LanGao Rd., Putuo District, Shanghai, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 3.0