RX21A Group
Gain Calibration and Compensation with the Temperature Sensor for the ΔΣ A/D Converter

Abstract
The RX21A Group has the function to satisfy the requirements of class 0.2S and 0.5S meters standardized in IEC62052-11 and IEC 62053-22.

The RX21A Group can amplify analog input using the on-chip PGA (programmable gain amplifier) providing the mechanism to reduce errors during amplifying. In the range of current $0.01I_{n} \leq I \leq I_{\text{max}}$, the measurement accuracy after calibration at the reference temperature satisfies the class 0.2S meter requirements standardized in IEC 62053-22.

The measurement values with the 24-bit ΔΣ A/D converter (DSAD) are influenced by temperature. However the temperature characteristics of the RX21A have been clarified. Thus the measured values can be compensated using the powerful calculation ability of the RX21A. Thus even if not using an external reference power supply with high precision, the measurement accuracy after compensation by the RX21A DSAD can satisfy the requirement for the class 0.5S meter standardized by IEC62052-11 and IEC 62053-22 in the temperature range from -25°C to +75°C.

This document describes calibration for DSAD gains in the RX21A Group and the method for compensating the temperature characteristics of the DSAD gain using the built-in temperature sensor.

Products
- RX21A Group 100-pin package with a ROM size between 256 KB and 512 KB
- RX21A Group 80-pin package with a ROM size between 256 KB and 512 KB
- RX21A Group 64-pin package with a ROM size between 256 KB and 512 KB

Note: Only the G version (operating temperature: -40°C to +105°C) of RX21A is the target device in this application note.
Contents

1. Specifications .................................................................................................................. 4

2. Operation Confirmation Conditions .................................................................................. 5

3. Reference Application Note ............................................................................................. 5

4. Hardware ........................................................................................................................ 6
   4.1 Example of the Hardware Configuration ....................................................................... 6
   4.2 Pins Used ..................................................................................................................... 7

5. Calibration for Gain and Offset Errors ............................................................................ 8
   5.1 Errors of the DSAD ................................................................................................. 8
   5.2 Calculating the Calibration Values for Gains and the Offsets ..................................... 9
      5.2.1 Calibration with a DC Power Supply ................................................................. 9
      5.2.2 Calibration with Sine Wave AC Power Supply .................................................. 10

6. System Gain Calibration .................................................................................................. 11
   6.1 Calibrating the System Gain ..................................................................................... 11
   6.2 Device Gain .............................................................................................................. 11
   6.3 Influences of External Input Resistor and Internal Input Resistor ............................. 12

7. Temperature Characteristics and Compensation ............................................................ 14
   7.1 Compensating Temperature Characteristics ............................................................. 14
   7.2 Coefficients of the Temperature Characteristics ....................................................... 15
   7.3 Device Gain ............................................................................................................ 15
   7.4 VBGR ....................................................................................................................... 16
   7.5 Input Impedance ...................................................................................................... 17
   7.6 Influences of External Load Resistor and Input Impedance ....................................... 18
   7.7 Compensation for the Temperature Characteristics of the System Gain ................ 19

8. Calibration and Compensation Procedures ..................................................................... 20
   8.1 System Gain Calibration and Compensation (Differential Input) .............................. 20
   8.2 System Gain Calibration and Compensation (Single-Ended Input) ......................... 22

9. Software .......................................................................................................................... 24
   9.1 Operation Overview .................................................................................................. 24
   9.2 Required Memory Size .............................................................................................. 26
   9.3 File Composition ...................................................................................................... 27
   9.4 Option-Setting Memory ............................................................................................ 28
   9.5 Constants ................................................................................................................. 29
   9.6 Variables ................................................................................................................... 32
   9.7 Functions ................................................................................................................. 36
   9.8 Function Specifications ............................................................................................. 37
   9.9 Flowcharts ............................................................................................................... 43
      9.9.1 Main Processing .................................................................................................. 43
      9.9.2 Peripheral Function Initialization ................................................................. 45
      9.9.3 CMT1 Initialization .......................................................................................... 45
      9.9.4 IRQ2 Initialization .......................................................................................... 46
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.9.5</td>
<td>Compare Match 1 Interrupt Handler</td>
<td>47</td>
</tr>
<tr>
<td>9.9.6</td>
<td>DSAD Initialization</td>
<td>48</td>
</tr>
<tr>
<td>9.9.7</td>
<td>DSAD Conversion Start Processing</td>
<td>49</td>
</tr>
<tr>
<td>9.9.8</td>
<td>Coefficient Initialization for Gain Calibration</td>
<td>50</td>
</tr>
<tr>
<td>9.9.9</td>
<td>Coefficient Initialization for Gain Temperature Compensation</td>
<td>51</td>
</tr>
<tr>
<td>9.9.10</td>
<td>System Gain Calibration</td>
<td>52</td>
</tr>
<tr>
<td>9.9.11</td>
<td>Temperature Compensation for the System Gain</td>
<td>53</td>
</tr>
<tr>
<td>9.9.12</td>
<td>Obtaining DSAD Conversion Result at Calibration</td>
<td>54</td>
</tr>
<tr>
<td>9.9.13</td>
<td>Obtaining DSAD Conversion Result</td>
<td>55</td>
</tr>
<tr>
<td>9.9.14</td>
<td>DSAD Conversion Interrupt Handler</td>
<td>56</td>
</tr>
<tr>
<td>9.9.15</td>
<td>A/D Converter and Temperature Sensor Initializations</td>
<td>57</td>
</tr>
<tr>
<td>9.9.16</td>
<td>A/D Converter and Temperature Sensor Stop Processing</td>
<td>58</td>
</tr>
<tr>
<td>9.9.17</td>
<td>Obtaining A/D Conversion Status</td>
<td>58</td>
</tr>
<tr>
<td>9.9.18</td>
<td>Obtaining Temperature Sensor Measurement Result</td>
<td>59</td>
</tr>
<tr>
<td>9.9.19</td>
<td>Obtaining Current Temperature</td>
<td>59</td>
</tr>
<tr>
<td>9.9.20</td>
<td>Temperature Sensor Calibration Processing</td>
<td>59</td>
</tr>
<tr>
<td>9.9.21</td>
<td>Temperature Sensor Measurement Processing</td>
<td>60</td>
</tr>
<tr>
<td>9.9.22</td>
<td>Current Temperature Calculation</td>
<td>60</td>
</tr>
<tr>
<td>9.9.23</td>
<td>A/D Conversion End Interrupt Handler</td>
<td>61</td>
</tr>
</tbody>
</table>

10. Appendices (Calibration and Compensation Results) | 62   |
| 10.1    | Result of the System Gain Calibration | 62   |
| 10.2    | Result of Temperature Compensations | 63   |
| 10.2.1  | Temperature Characteristics of the VBGR | 63   |
| 10.2.2  | System Gain of the Differential Input Pins | 64   |
| 10.2.3  | System Gain of Single-Ended Input Pin | 66   |

11. Sample Code | 68   |

12. Reference Documents | 68   |
1. Specifications

With 7 channels of independent DSAD, different input voltages can concurrently be measured switching PGA gains. Then some errors will occur due to chip variations or temperature characteristics. When high accuracy is required on a DSAD conversion, the system gain including the external circuit such as sensor may have to be calibrated among channels and the temperature characteristics may have to be compensated. If the application cannot eliminate an offset with the bypass filter such as direct current measurement, an offset will also need to be calibrated.

The G version of the RX21A Group has the I/O registers (ΔΣ A/D input impedance calibration data register and ΔΣ A/D gain calibration data registers) which store the calibration data for input impedance and gains measured on each chip at factory shipping. With these calibration data, the user can calibrate gains for all channels by calibrating only one given gain.

The ΔΣ A/D input impedance calibration data register and the ΔΣ A/D gain calibration data registers are not included in the RX21A Group products other than the G version. In those products, the sample code handles the read value as 1 (no effect on a calculation) when it performs calculations. The sample code cannot be used for calibrating gains for all channels by calibrating one given gain. However, it can be used as a reference when calibrating among channels and compensating the temperature characteristics.

Table 1.1 lists the Peripheral Functions and Their Applications.

### Table 1.1 Peripheral Functions and Their Applications

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-bit ΔΣ A/D converter (DSAD)</td>
<td>Measures analog input voltage.</td>
</tr>
<tr>
<td>Temperature sensor (TEMPSa)</td>
<td>Measures an ambient temperature for the MCU.</td>
</tr>
<tr>
<td>10-bit A/D converter (AD)</td>
<td>Measures the temperature sensor output.</td>
</tr>
<tr>
<td>Compare match timer (CMT1)</td>
<td>Used as the start trigger source of DSAD conversion and used for start processing of the temperature sensor.</td>
</tr>
<tr>
<td>Event link controller (ELC)</td>
<td>Used as the start trigger of DSAD conversion.</td>
</tr>
</tbody>
</table>
2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>R5F521A8BGFP (RX21A Group G version)</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td>• Main clock: 20 MHz</td>
</tr>
<tr>
<td></td>
<td>• System clock (ICLK): 50 MHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral module clock B (PCLKB): 25 MHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral module clock C (PCLKC): 25 MHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral module clock D (PCLKD): 12.5 MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>3.3 V</td>
</tr>
<tr>
<td>Integrated development</td>
<td>Renesas Electronics Corporation</td>
</tr>
<tr>
<td>environment</td>
<td>High-performance Embedded Workshop Version 4.09.01</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01</td>
</tr>
<tr>
<td></td>
<td>Compile options</td>
</tr>
<tr>
<td></td>
<td>-cpu=rx200 -output=obj=&quot;$(CONFIGDIR)/$(FILELEAF).obj&quot; -debug -nologo</td>
</tr>
<tr>
<td></td>
<td>(The default setting in the integrated development environment is used.)</td>
</tr>
<tr>
<td>iodifine.h version</td>
<td>Version 1.1B</td>
</tr>
<tr>
<td>Endian</td>
<td>Little endian</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Single-chip mode</td>
</tr>
<tr>
<td>Processor mode</td>
<td>Supervisor mode</td>
</tr>
<tr>
<td>Sample code version</td>
<td>Version 1.10</td>
</tr>
</tbody>
</table>

3. Reference Application Note

For additional information associated with this document, refer to the following application notes.

- RX21A Group Initial Setting (R01AN1486EJ)
- RX21A Group Using the Temperature Sensor to Calculate the Ambient Temperature (R01AN1923EJ)
- RX21A Group ΔΣ A/D Converter User's Guide (R01AN1437EJ)
- RX Family Coding Example of Wait Processing by Software (R01AN1852EJ)

The sample code in this application note uses the initial setting functions and wait processing by the software in the reference application notes. The revision number of the reference application note is current as of when this application note was created. However, the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.
4. Hardware

4.1 Example of the Hardware Configuration

Figure 4.1 shows the block diagram of functions used.
4.2 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

The number of pins in the sample code is set for the 100-pin package. When using products with less than 100 pins, select pins appropriate to the package used.

Table 4.1 Pins Used and Their Functions

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANDS0P, ANDS0N</td>
<td>Input</td>
<td>Analog differential input pin, channel 0</td>
</tr>
<tr>
<td>ANDS1P, ANDS1N</td>
<td>Input</td>
<td>Analog differential input pin, channel 1</td>
</tr>
<tr>
<td>ANDS2P, ANDS2N</td>
<td>Input</td>
<td>Analog differential input pin, channel 2</td>
</tr>
<tr>
<td>ANDS3P, ANDS3N</td>
<td>Input</td>
<td>Analog differential input pin, channel 3</td>
</tr>
<tr>
<td>ANDS4</td>
<td>Input</td>
<td>Analog single-ended input pin, channel 4</td>
</tr>
<tr>
<td>ANDS5</td>
<td>Input</td>
<td>Analog single-ended input pin, channel 5</td>
</tr>
<tr>
<td>ANDS6</td>
<td>Input</td>
<td>Analog single-ended input pin, channel 6</td>
</tr>
<tr>
<td>ANDSSG</td>
<td>Input</td>
<td>Analog single-ended input pin, connected to the common signal ground</td>
</tr>
<tr>
<td>BGR_BO</td>
<td>Input</td>
<td>Reference external application terminal, the input is used as the internal reference voltage.</td>
</tr>
</tbody>
</table>
5. Calibration for Gain and Offset Errors

5.1 Errors of the DSAD

Figure 5.1 shows an Example of the DSAD I/O Characteristics.

The values of the ΔΣ A/D data registers are expressed as 32 bits of 2’s complement. When the DSAD has the ideal characteristics, the formula becomes as follows:

\[
\text{Formula 5.1} \quad \frac{\text{A/D conversion value}}{\text{gain}} = \frac{\text{analog input voltage}}{\text{VREFDSH pin voltage}} \times 2^{23} \times \frac{t_{\text{TRIG}}}{t_{\text{os}} \times 256}
\]

However, the DSAD actually has gain and offset errors and the DSAD conversion value will be slightly different from the theoretical value. Furthermore, the sensor externally connected to the DSAD normally has gain and offset errors as well. To calculate an analog input voltage value with high accuracy based on the measured digital value, errors of gains and offsets need to be calibrated.

![Figure 5.1 Example of the DSAD I/O Characteristics](image-url)
5.2 Calculating the Calibration Values for Gains and the Offsets

Gain and offset can be calibrated by inputting voltages on two or more reference points to the DSAD input pins and measuring digital output values from each point beforehand.

When the voltage applied to the DSAD input pin is used as the reference voltage, gain and offset of RX21A itself (device gain and device offset) can be calculated. When the voltage or current applied to the sensor input section of the system is used as the reference value, gain and offset of a whole system including the sensor (system gain and system offset) can be calculated.

Calibration value error must be reduced as much as possible by averaging multiple measurement values when measuring the calibration value or by using the least squares method when calculating the calibration value.

DC power supply or sine wave AC power supply can be used for calibration.

5.2.1 Calibration with a DC Power Supply

Figure 5.2 shows the Method to Calculate the Calibration Values for Gain and Offset with DC Power Supply.

Calibration values for gain and offset can be calculated based on the measured values of two different DC voltages applied to the DSAD input pins.

When input voltages are yB and yC, and digital output values at them are xB and xC, the formula to calculate calibration values for gain and offset are as follows:

Formula 5.2
Gain = \( \frac{xC - xB}{yC - yB} \)

Formula 5.3
Offset = \( xB - \frac{xC - xB}{yC - yB} \times yB \)

![Diagram showing the method to calculate the calibration values for gain and offset with DC power supply.](image)
5.2.2 Calibration with Sine Wave AC Power Supply

Figure 5.3 shows the Method to Calculate the Calibration Values for Gain and Offset with Sine Wave AC Power Supply.

When calculating the calibration values for gain and offset with sine wave AC power supply, formulas 5.2 and 5.3 are also used. In formula 5.2, values are assigned to xB and xC assuming that absolute values of positive and negative peak values of sine waves are same, and the minimum and maximum values of sine waves based on the digital values which are oversampled in the DSAD are assigned to yB and yC. And at this time, offset corresponds to the average value of the sine waves based on the digital values.

Calibration values for gain and offset can also be calculated using formula 5.4 instead of formula 5.2.

**Formula 5.4**

\[
\text{Gain} = \frac{x_{\text{RMS}}}{y_{\text{RMS}}}
\]

- \(x_{\text{RMS}}\): RMS value of a sine wave based on the digital value
- \(y_{\text{RMS}}\): RMS value of an input sine wave

---

**Figure 5.3** Method to Calculate the Calibration Values for Gain and Offset with Sine Wave AC Power Supply
6. System Gain Calibration

6.1 Calibrating the System Gain

System gain error is caused by the DSAD internal circuit for each selectable gain and external sensor input circuits. If related voltages are measured in multiple channels, the system gain error needs to be calibrated to reduce measurement error among channels.

With the finalized product, the system gain must be calibrated at least once for all channels while the external circuit is connected.

Figure 6.1 shows the concept of the gain calibration. The left chart shows raw gains, the center shows gains after compensating linearity mismatches among gains using the calibration data stored in the device, and the right shows gains after calibrating offset errors. Actual gains for channels appear in a variety of positions relative to the PGA gain settings (left chart). The gains are calibrated starting in order from gain x1 to make each gain be closer with fewer mismatches (center chart). Then offset errors are removed shown in the right chart.

![Gain Calibration Diagram](image)

Note: This figure shows concept only. Errors in these charts are exaggerated.

Figure 6.1  Gain Calibration

6.2 Device Gain

The calibration data for gains on each device (device gain) is measured and stored in the GCD[15:0] bits in the ΔΣ A/D gain calibration data registers (DSADGmXn) (m = 0 to 6, n = 1, 2, 4, 8, 16, and 32) before shipping.

The device gain for each channel with each gain setting can be obtained using the formula 6.1.

Note that the calibration data for gain x64 is not stored. Twice the gain x32 for the device gain of gain x64.

\[
\text{DeviceGain} (m, n) = n \times \text{DSADGmXn.GCD}[15:0] / 47971 \\
\text{DeviceGain} (m, 64) = \text{DeviceGain} (m, 32) \times 2
\]

\(m\): Input channel (0 to 6)  
\(n\): Gain (1, 2, 4, 8, 16, and 32) selected with ΔΣ A/D gain select registers 0 to 6 (DSADGRSm)
6.3 Influences of External Input Resistor and Internal Input Resistor

Figure 6.2 shows the Connection Diagram of the Differential Input Channel.

![Connection Diagram of the Differential Input Channel](image)

Rp: External input resistor, Ri: Internal input resistor, Rf: Feedback resistor

A low-pass filter (anti-aliasing filter) composed of the external input resistor Rp and the capacitor must be connected to the input pins of the DSAD for preventing an aliasing error.

In this connection example, the input resistor is the sum of the internal input resistor Ri within the DSAD and the external input resistor Rp. Then the device gain is proportional to the ratio between the input resistor and the feedback resistor.

**Formula 6.2**

\[
\text{DeviceGain} (m, n) \propto \frac{Rf(n)}{\{ Ri(n) + Rp(m) \}}
\]

- m: Input channel (0 to 3)
- n: Gain (1, 2, 4, 8, 16, 32 and 64) selected with ΔΣ A/D gain select registers 0 to 3 (DSADGSRm)
- Rf(n): Feedback resistor at gain n
- Ri(n): Internal input resistor at gain n
- Rp(m): External input resistor of channel m

Table 6.1 shows the internal resistor (Ri and Rf) for each gain setting.

**Table 6.1  Internal Resistor Values when Setting Each Gain**

<table>
<thead>
<tr>
<th>DSADGSRm. GAIN[2:0]</th>
<th>Gain</th>
<th>Internal Input Resistor Ri(n)</th>
<th>Feedback Resistor Rf(n)</th>
<th>Gain of the ΔΣ Modulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>x1</td>
<td>R0</td>
<td>Rf0</td>
<td>1</td>
</tr>
<tr>
<td>001b</td>
<td>x2</td>
<td>R0</td>
<td>2Rf0</td>
<td>1</td>
</tr>
<tr>
<td>010b</td>
<td>x4</td>
<td>R0</td>
<td>4Rf0</td>
<td>1</td>
</tr>
<tr>
<td>011b</td>
<td>x8</td>
<td>8R0</td>
<td>8Rf0</td>
<td>1</td>
</tr>
<tr>
<td>100b</td>
<td>x16</td>
<td>8R0/2</td>
<td>8Rf0</td>
<td>1</td>
</tr>
<tr>
<td>101b</td>
<td>x32</td>
<td>8R0/2</td>
<td>8Rf0</td>
<td>2</td>
</tr>
<tr>
<td>110b</td>
<td>x64</td>
<td>8R0/2</td>
<td>8Rf0</td>
<td>4</td>
</tr>
</tbody>
</table>
Ri₀ and Rf₀ values in Table 6.1 are designed to 100 kΩ. In practice, these values vary depending on devices. This variation in Ri₀ is proportional to variation in impedance. Ri₀ can be expressed with formula 6.3 using the value of the IICD[15:0] bits in the ΔΣ A/D input impedance calibration data register (DSADIIC), which is measured and stored before shipping.

\[ R_{i0} = 100.0 \times \text{DSADIIC.IICD}_{[15:0]} / 32768 \text{ [kΩ]} \]

The system gain is a product of the sensor gain and the device gain. The sensor gain is the gain on the circuit that inputs to the DSAD.

\[ \text{SystemGain} (m, n) = \text{SensorGain} (m) \times \text{DeviceGain} (m, n) \]

m: Input channel (0 to 6)

n: Gain selected with ΔΣ A/D gain select registers 0 to 6 (DSADGSRm) (1, 2, 4, 8, 16, and 32)

SystemGain (m, n): Total of the sensor gain and the device gain on channel m with gain setting n

SensorGain (m): Sensor gain on channel m

When the gain setting is for x16, x32, and x64, the input resistor becomes half the value of the input resistor with gain setting for x1, x2, x4 and x8. Therefore the influence of the external input resistor Rp on the system gain varies depending on the gain setting. Formula 6.5 shows the influence ratio.

\[ \frac{\text{SystemGain} (n_H = 16, 32, 64)}{\text{SystemGain} (n_L = 1, 2, 4, 8)} \propto \left( \frac{R_{i0}}{2 + Rp} \right) / \left( R_{i0} + Rp \right) \]

\[ \approx 1 + \frac{Rp}{R_{i0}} \]
7. Temperature Characteristics and Compensation

7.1 Compensating Temperature Characteristics

If the device temperature varies, the device gain, the VBGR, and the temperature characteristics of input impedance cause DSAD measurement errors. The DSAD measurement errors can be reduced by compensating the system gain, which is calibrated as described in Section 6, using the temperature information for the device.

The device temperature is obtained using the built-in temperature sensor. The accuracy of the temperature measured with the temperature sensor affects the accuracy of gain compensation. Therefore the temperature sensor must be calibrated beforehand.

Figure 7.1 shows the Elements that Have Temperature Characteristics.
7.2 Coefficients of the Temperature Characteristics

Table 7.1 lists the Coefficients of the Temperature Characteristics in the RX21A Group.

<table>
<thead>
<tr>
<th>Element</th>
<th>Coefficient</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGR</td>
<td>Quadratic slope</td>
<td>C_{BA}</td>
<td>-0.26 \times 10^{-6}</td>
<td>K^2</td>
</tr>
<tr>
<td></td>
<td>Linear slope</td>
<td>C_{BB}</td>
<td>5.5 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td>Device Gain</td>
<td>x1</td>
<td>C_{X1}</td>
<td>-5 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td></td>
<td>x2</td>
<td>C_{X2}</td>
<td>-4 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td></td>
<td>x4</td>
<td>C_{X4}</td>
<td>-7 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td></td>
<td>x8</td>
<td>C_{X8}</td>
<td>-2 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td></td>
<td>x16</td>
<td>C_{X16}</td>
<td>-14 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td></td>
<td>x32</td>
<td>C_{X32}</td>
<td>-14 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td></td>
<td>x64</td>
<td>C_{X64}</td>
<td>-14 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
<tr>
<td>Input impedance</td>
<td>Linear slope</td>
<td>C_{Z}</td>
<td>-1200 \times 10^{-6}</td>
<td>K^{-1}</td>
</tr>
</tbody>
</table>

7.3 Device Gain

Figure 7.2 shows the theoretical temperature characteristics of the device gain when the reference voltage (VBG) assumes to have no temperature characteristics.

![Figure 7.2](image-url)

The temperature characteristics of the device gain can be calculated with formula 7.1.

**Formula 7.1**

\[
\text{Device gain (Tj)} = \text{Device gain(Tj = 25)} \times \{1 + C_{Xn}(Tj - 25)\}
\]

- \(Tj\): Junction temperature on the chip [°C]
- \(C_{Xn}\): (n = 1, 2, 4, 8, 16, 32, and 64): Coefficient of the temperature characteristics.
- Refer to Table 7.1 for coefficient values.
7.4 VBGR

The on-chip BGR voltage (VBGR) or the external reference voltage (BGR_BO) can be used as the reference voltage (VBG). When BGR_BO is used, the influence of the VBGR can be excluded. However, the temperature characteristics of BGR_BO need to be taken into account.

Figure 7.3 shows the Temperature Characteristics of the VBGR. The VBGR is adjusted to output 1.22 V at 25°C before shipping, however, it may drop up to 1.2189 V depending on the temperature.

\[
\text{Formula 7.2} \\
\text{VBGR}(T_j) = \text{VBGR}(T_j = 25) \times \{1 + C_{BA}(T_j - 25)^2 + C_{BB}(T_j - 25)\}
\]

- \(T_j\): Junction temperature on the chip [°C]
- \(C_{BA}\): Coefficient of quadratic slope. Refer to Table 7.1 for the coefficient values.
- \(C_{BB}\): Coefficient of linear slope. Refer to Table 7.1 for coefficient values.
- \(\text{VBGR}(T_j = 25)\): Typical voltage of BGR: 1.220 [V]
7.5 Input Impedance

Figure 7.4 shows the temperature characteristics of input impedance $Z_i$ for the single-ended input at gain x1.

The temperature characteristic of input impedance $Z_i$ at each gain setting can be calculated with formula 7.3.

**Formula 7.3**

\[ Z_i (T_j) = Z_i (T_j = 25) \times \frac{DSADIIC.IICD[15:0]}{32768} \times \left(1 + C_Z \times (T_j - 25)\right) \]

- $T_j$: Junction temperature on the chip [°C]
- $Z_i (T_j = 25)$: Typical value of the input impedance. The value differs depending on the gain setting.
- $C_Z$: Coefficient. See Table 7.1 for coefficient values.
7.6 Influences of External Load Resistor and Input Impedance

Figure 7.4 shows the Connection Example of Single-Ended Input.

![Connection Example of Single-Ended Input](image)

R_S: External input resistor, R_L: External load resistor, R_P: Resistor of the anti-aliasing filter, Z_I: Input impedance, E_S: Measurement voltage, E_I: Input voltage

Figure 7.5 Connection Example of Single-Ended Input

In this connection example, when the input impedance becomes lower due to rise in temperature, electrical current flows through the external load resistor is reduced. Then the input voltage (E_I) to the DSAD is also reduced and the appearance of the system gain changes. Formula 7.4 shows the relations among the load resistor, the input impedance, and the system gain.

**Formula 7.4**

\[
\text{SystemGain} \propto \frac{E_S}{E_I} = \left[ \left( R_S \times (R_P + Z_I) + R_L \times (R_S + R_P + Z_I) \right) / \left( R_L \times Z_I \right) \right] \\
\approx \left( 1 + \frac{R_L}{Z_I} \right) \times \left( \frac{R_S}{R_L} \right) \propto 1 + \frac{R_L}{Z_I}
\]

- R_P: Resistor of the anti-aliasing filter
- R_L: External load resistor
- Z_I: Input impedance

* The approximation formula is when R_P is 100Ω, R_L is 1.8kΩ, Z_I is 80kΩ, and R_S is 1.32 MΩ.

In Figure 6.2 Connection Diagram of the Differential Input Channel, when the output impedance from the sensor is low enough, even if the input impedance to the DSAD changes, the input voltage to the DSAD does not change. Thus influences on the system gain by the temperature characteristics of the input impedance can be ignored.
7.7 Compensation for the Temperature Characteristics of the System Gain

The system gain of the differential input is proportional to the device gain and is inversely proportional to the reference voltage VBG composed of the on-chip BGR or the external reference voltage.

The system gain of the single-ended impedance is influenced by the input impedance shown in formula 7.4.

System gain of the differential input

**Formula 7.5**

\[
\text{SystemGain (differential input)} \propto \text{Device gain} / \text{VBG}
\]

- Device gain \( (T_j) = \text{Davice gain} (T_j = 25) \times \{1 + C_{Xn}(T_j - 25)\} \) (from formula 7.1)
- \( \text{VBGR}(T_j) = \text{VBGR}(T_j = 25) \times \{1 + C_{BA}(T_j - 25)^2 + C_{BB}(T_j - 25)\} \) (from formula 7.2)

**Formula 7.6**

\[
\text{SystemGain (differential input)} (T_j) \\
\approx \text{SystemGain (differential input)} \times \{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB})(T_j - 25)\}
\]

System gain of the single-ended input

**Formula 7.7**

\[
\text{SystemGain (single-ended input)} \propto \text{Device gain} / \text{VBG} \times (1 + R_L/Z_I)
\]

- \( R_L: \) External load resistor [Ω]
- \( Z_I: \) Input impedance [Ω]

- DeviceGain \( (T_j) = \text{Davice gain} (T_j = 25) \times \{1 + C_{Xn}(T_j - 25)\} \) (from formula 7.1)
- \( \text{VBGR (T_j)} = \text{VBGR (T_j = 25)} \times \{1 + C_{BA}(T_j - 25)^2 + C_{BB}(T_j - 25)\} \) (from formula 7.2)
- \( Z_I (T_j) = Z_I (T_j = 25) \times \text{DSADIIC.IICD}[15:0] / 32768 \times \{1 + C_Z \times (T_j - 25)\} \) (from formula 7.3)

**Formula 7.8**

\[
\text{SystemGain (single-ended input)} (T_j) \\
\approx \text{SystemGain (single-ended input)} (T_j = 25) \\
\times \{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB} + R_L / Z_I (T_j = 25) / \text{DSADIIC.IICD}[15:0] \times 32768 \times C_Z) (T_j - 25)\}
\]
8. Calibration and Compensation Procedures

This chapter describes calibration and compensation for the system gains of differential input channels and single-ended input channels, and calibration for offsets using formulas described in sections 6 and 7.

Conditions for calibration and compensation for input channels, and calibration for offsets are as follows:
- Reference temperature = 25°C (when inputting a voltage for a test)
- Voltage yB = 450 mV, yC = -450 mV

8.1 System Gain Calibration and Compensation (Differential Input)

This section describes the procedure for system gain calibration.

1. Calculate the linear coefficient (C_Xn - C_BB) used for temperature compensation and store the result in the appropriate variable. Then assign the quadratic coefficient to the appropriate variable.

   Refer to Table 7.1 for details on C_BB, C_Xn, and C_BA.

   \[
   \text{SystemGain} (m, 1) \text{ (differential input)} (T_j) \approx \text{SystemGain} (m, 1) \text{ (differential input)} (T_j = 25) \times \left\{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB})(T_j - 25)\right\}
   \]

2. Calculate the sensor gain using the system. First initialize the DSAD and specify gains for all channels (e.g. gain x1).

3. Input the voltage (yB in Figure 5.2) for testing to the input pins for each channel in the system used.

4. Repeat DSAD conversion at a given cycle for an appropriate number of times and obtain the average of the conversion results.

5. Input the voltage (yC in Figure 5.2) for testing to the input pins for each channel.

6. Repeat DSAD conversion at a given cycle for an appropriate number of times and obtain the average of the conversion results.

7. Input 0 V voltage for testing, repeat DSAD conversion at a given cycle for an appropriate number of times and obtain the average of the conversion results, and measure the offset for each gain.

8. Calculate the sensor gain using the DSAD conversion data obtained from the voltages at two points.

   \[
   \text{SystemGain} (m, 1) (T_j) = \left(\frac{V_{REFDSH \text{ voltage}}}{yC - yB}\right) \times \frac{TRIG}{(T_{OS} \times 256)} / 2^{23} \quad \text{(from formula 5.1)}
   \]

9. To perform temperature compensation for the sensor gain at the temperature when the DSAD conversion data has been obtained, perform temperature compensation for the result of formula 5.1.

   \[
   \text{SensorGain} (m) (T_j) = \text{SystemGain} (m, 1) (T_j = 25) \times \left\{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB})(T_j - 25)\right\}
   \]

10. Calculate the system gains for gains x1 to x8 based on the sensor gain obtained with formulas 6.1 and 8.1.

    \[
    \text{DeviceGain} (m, n) (T_j) = n \times \frac{\text{DSADGmXn.GCD[15:0]}}{47971} \quad \text{(from formula 6.1)}
    \]

    \[
    \text{SystemGain} (m, n) (T_j) = \text{SensorGain} (m) \times \text{DeviceGain} (m, n) \quad \text{(from formula 6.4)}
    \]

Formula 8.1

\[
\text{SystemGain} (m, 1) (T_j = 25) = \text{SystemGain} (m, 1) (T_j) / \left\{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB})(T_j - 25)\right\}
\]

\[
\text{SensorGain} (m, n) (T_j) = \text{SystemGain} (m, 1) (T_j = 25) / \text{DeviceGain} (m, 1) \quad \text{(from formula 7.6)}
\]

11. Compensate the system gains (formula 8.2) for gain x1 to gain x8 with a temperature.

    \[
    \text{SystemGain} (m, n) (T_j) = \left(\frac{\text{SensorGain} (m) (T_j) \times n \times \text{DSADGmXn.GCD[15:0]}}{47971}\right) \quad \text{(from formula 6.4)}
    \]

Formula 8.2

\[
\text{SystemGain} (m, n) (T_j) = \text{SensorGain} (m) (T_j) \times n \times \frac{\text{DSADGmXn.GCD[15:0]}}{47971}
\]

Formula 8.3

\[
\text{SystemGain} (m, n) \text{ (differential input)} (T_j) \approx \text{SystemGain} (m, n) (T_j - 25) \times \left\{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB})(T_j - 25)\right\}
\]

Formula 8.4
12. Calculate the system gains for gains x16 to x32 based on formulas 6.1, and 6.3 to 6.5.

\[
\text{DeviceGain} (m, n) (Tj) = n \times \text{DSADGmXn.GCD}[15:0] / 47971 \quad \text{(from formula 6.1)}
\]
\[
\text{Ri0} = 100.0 \times \text{DSADIIC.IICD}[15:0] / 32768 \text{ [kΩ]} \quad \text{(from formula 6.3)}
\]
\[
\text{SystemGain} (m, n) (Tj) = \text{SensorGain} (m) \times \text{DeviceGain} (m, n) \quad \text{(from formula 6.4)}
\]
\[
\text{SystemGain} (n = 16, 32, 64) (Tj) / \text{SystemGain} (n = 1, 2, 4, 8) (Tj)
\]
\[
\approx 1 + \frac{R_p}{R_i}
\]

**Formula 8.4**

\[
\text{SystemGain} (m, n) (Tj) = \text{SensorGain} (m, 1) (Tj) / 1^* / \\
\text{DSADGmX1.GCD}[15:0] \times 47971 \times \text{DSADGmXn.GCD}[15:0] \times n / 47971 \\
\times (1 + \frac{R_p}{100k} \times 32768 / \text{DSADIIC.IICD}[15:0])
\]

* This formula is when the reference gain is set to gain x1.

13. Calculate the system gain for gain x64 based on formula 6.1.

\[
\text{DeviceGain} (m, 64) = \text{DeviceGain} (m, 32) \times 2 \quad \text{(from formula 6.1)}
\]

**Formula 8.5**

\[
\text{SystemGain} (m, 64) = \text{SystemGain} (m, 32) \times 2
\]

14. Compensate the system gain for gains x16 to x64 with a temperature.

\[
\text{SystemGain} (m, n) (\text{differential input}) (Tj) \approx \text{SystemGain} (m, n) (\text{differential input}) (Tj = 25) \times \{1 - C_{BA}(Tj - 25)^2 + (C_{xn} - C_{BB})(Tj - 25)\}
\]

**Formula 8.6**

\[
\text{SystemGain} (m, n) = (\text{Differential input}) (Tj) \approx \text{SystemGain} (m, n) (\text{differential input}) (Tj = 25) \\
\times \{1 - C_{BA}(Tj - 25)^2 + (C_{xn} - C_{BB})(Tj - 25)\}
\]

Up to here, the calibration and compensation are complete. Step 15 is used when the temperature result is obtained in the main loop.

15. Compensate the measured result using the information of the system gain compensation after the temperature compensation.

**Formula 8.7**

\[
(\text{DSAD value after compensation}) = ((\text{value from any of DSADDR0 to DSADDR 3}) - \\
(\text{gain offset value (result in step 7)})) / \text{SystemGain} (m, n) (Tj = 25) \\
\times \text{SystemGain} (m, n) (\text{differential input}) (Tj)
\]
8.2 System Gain Calibration and Compensation (Single-Ended Input)

This section describes the procedure for system gain calibration.

1. Calculate the linear coefficient \((C_{Xn} - C_{BB} + R_{L} / Z_{I}(T_j = 25)) / DSADIIC.IICD[15:0] \times 32768 \times C_{Z})\) used for temperature compensation and store the result in an appropriate variable. Then assign the quadratic coefficient \((C_{BA})\) to an appropriate variable.

Refer to Table 7.1 for details on \(C_{BA}, C_{Xn}, C_{BB},\) and \(C_{Z}\).

\[
\text{SystemGain (m, 1) (single-ended input) (Tj)} = \text{SystemGain (m, 1) (single-ended input) (Tj = 25)} \times \{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB} + R_{L} / Z_{I}(T_j = 25)) / DSADIIC.IICD[15:0] \times 32768 \times C_{Z}(T_j - 25)\}
\]

2. Calculate the sensor gain using the system. First initialize the DSAD and specify gains for all channels (e.g. gain \(x1\)).

3. Input the voltage \((yB\) in Figure 5.2) for testing to the input pins for each channel in the system used.

4. Repeat DSAD conversion in a given cycle for an appropriate number of times and obtain the average of the conversion results.

5. Input the voltage \((yC\) in Figure 5.2) for testing to the input pins for each channel.

6. Repeat DSAD conversion in a given cycle for an appropriate number of times and obtain the average of the conversion results.

7. Input 0 V voltage for testing, repeat DSAD conversion at a given cycle for an appropriate number of times and obtain the average of the conversion results, and measure the offset for each gain.

8. Calculate the sensor gain using the DSAD conversion data obtained from the voltages at two points.

\[
\text{SystemGain (m, 1) (Tj) = } (xC - xB) / \{T_{TRIG} / (T_{OS} \times 256)\} / 2^{23}
\]

\[
\times \text{VREFDSH voltage / (yC - yB)}
\]

9. To perform temperature compensation for the sensor gain at the temperature when the DSAD conversion data has been obtained, perform temperature compensation for the result of formula 5.1.

\[
\text{SystemGain (m, n) (Tj) = SensorGain (m) (Tj) \times DeviceGain (m, n)}
\]

\[
\approx \text{SystemGain (m, n) (single-ended input) (Tj)} \times \{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB} + R_{L} / Z_{I}(T_j = 25)) / DSADIIC.IICD[15:0] \times 32768 \times C_{Z}(T_j - 25)\}
\]

**Formula 8.8**

\[
\text{SystemGain (m, n) (Tj) = } \text{SystemGain (m, 1) (Tj = 25)} / \{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB} + R_{L} / Z_{I}(T_j = 25)) / DSADIIC.IICD[15:0] \times 32768 \times C_{Z}(T_j - 25)\}
\]

\[
\times \text{DeviceGain (m, 1) / SensorGain (m) (Tj)}
\]

10. Calculate the system gains for gains \(x1\) to \(x4\) based on the sensor gain obtained with formulas 6.1 and 6.4.

\[
\text{DeviceGain (m, n) (Tj) = } n \times \text{DSADGmXn.GCD[15:0] / 47971}
\]

\[
\text{SystemGain (m, n) (Tj) = } \text{SensorGain (m) (Tj)} \times \text{DeviceGain (m, n)}
\]

**Formula 8.9**

\[
\text{SystemGain (m, n) (Tj) = SensorGain (m) \times n \times DSADGmXn.GCD[15:0] / 47971}
\]

11. Compensate the system gains (formula 8.9) for gains \(x1\) to \(x4\) with a temperature.

\[
\text{SystemGain (m, n) (single-ended input) (Tj)} \approx \text{SystemGain (m, n) (single-ended input) (Tj = 25)} \times \{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB} + R_{L} / Z_{I}(T_j = 25)) / DSADIIC.IICD[15:0] \times 32768 \times C_{Z}(T_j - 25)\}
\]

**Formula 8.10**

\[
\text{SystemGain (m, n) (differential input) (Tj)} \approx \text{SensorGain (m) (Tj) \times DeviceGain (m, n) \times} \{1 - C_{BA}(T_j - 25)^2 + (C_{Xn} - C_{BB} + R_{L} / Z_{I}(T_j = 25)) / DSADIIC.IICD[15:0] \times 32768 \times C_{Z}(T_j - 25)\}
12. Compensate the measured result using the information of the system gain compensation after the temperature compensation.

Formula 8.11

\[(\text{DSAD value after compensation}) = (\text{value from any of DSADDR0 to DSADDR3}) - \left(\frac{\text{(gain offset value (result in step 7))}}{\text{SystemGain (m, n) (Tj = 25)}}\right) / \text{SystemGain (m, n) (single-ended Input) (Tj)}\]
9. Software

9.1 Operation Overview

After a reset, the DSAD values for calibration and compensation are obtained, the system gain is calibrated and compensated, the temperature sensor is calibrated, and then DSAD conversion is performed while the system gain is compensated for the DSAD using the temperature information obtained from the temperature sensor.

The DSAD values for calibration and compensation are measured for gains x2 to x64 at 0 V and at voltages of two points. The voltages at two points are used to calculate the sensor gain. The measurement results at 0 V for gains x2 to x64 are used as offset values. The offset value varies for each gain, thus the offset value needs to be obtained for all gains from x2 to x64.

The temperature sensor is calibrated using the A/D conversion results after a reset and the calibration values at 125°C stored internally. After the temperature sensor has been calibrated, the calibration results are used for calculating the temperature data.

DSAD conversion is triggered by CMT1 every 163.84 µs and ten conversions are treated as one sampling. First two conversion results are discarded and the rest of eight results are used to calculate the sum. Then the sum is divided by 8 to obtain the average.

For A/D conversion used by the temperature sensor, 163.84 ms is counted by CMT1 610 times, A/D conversion is performed every 100 ms, six A/D conversion results are stored in the RAM, the maximum and minimum values are subtracted from the sum of the conversion results, and then the subtraction result is divided by 4 to obtain the average.
Figure 9.1 shows the Operation Timing Diagram.

(1) Reset is released with the voltage of $y_C$ [V] and temperature of ORDINARY_REF_TEMP, the initialization is performed, DSAD conversion values are obtained at $y_C$ [V] for each channel, and then a wait for the IRQ2 interrupt request is performed.

(2) At an appropriate timing that the conversions can be completed, the voltage is changed to $y_B$ [V] and SW is pressed. Then the IRQ2 interrupt request is generated and DSAD conversion values at $y_B$ [V] are obtained for each channel. At the same time, the temperature sensor is started and the A/D conversion values at the temperature of ORDINARY_REF_TEMP are obtained.

(3) A/D conversion results are obtained six times and the temperature sensor is calibrated.

(4) At an appropriate timing that the calibration for the temperature sensor can be completed, the voltage is changed to 0 [V] and SW is pressed. Then the IRQ2 interrupt request is generated and DSAD conversion values at 0 [V] are obtained for each channel at all gains.

(5) The DSAD conversion results are calibrated and compensated. At this point, the voltage and temperature held can be released.

(6) The DSAD conversion results are compensated using the ambient temperature information obtained with the temperature sensor. The DSAD conversion result is compensated using the calculated compensation value.
The settings for the DSAD, CMT1, temperature sensor, and A/D converter are as follows:

**DSAD**
- Reference voltage generation: On-chip BGR circuit
- Gain setting: All gains for offset measurement. Gain x1 for measurements other than offset.
- A/D conversion end interrupt: Used
- Overwrite interrupt (interrupt by data register overwrite): Not used
- Input select: Input from analog input pins used.

**CMT1**
- Count clock: PCLK divided by 32
- Compare match interrupt period: 163.84 µs

Note: The period must be set with an integral multiple of t_{OS}, so the trigger period will be between t_{OS} × 256 and t_{OS} × 768.

**Temperature sensor**
- PGA gain \(^{(1)}\): 2.7 V ≤ AVCC0 ≤ 3.6 V \(^{(2)}\)

Notes:
1. PGA: Programmable Gain Amplifier
2. Change the parameter setting according to the system used.

**A/D converter**
- Operating mode: Single scan mode
- A/D conversion start trigger: Synchronous trigger (trigger from the temperature sensor)
- Sampling state count: 180 states (sampling time: 72 µs)
- Analog input disconnection detection assist: Not used.
- A/D-converted value addition mode: Not used.
- Self-diagnosis of 10-bit A/D converter: Not used.

### 9.2 Required Memory Size

Table 9.1 lists the Required Memory Size.

<table>
<thead>
<tr>
<th>Memory Used</th>
<th>Size</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>7311 bytes</td>
<td></td>
</tr>
<tr>
<td>RAM</td>
<td>2046 bytes</td>
<td></td>
</tr>
<tr>
<td>Maximum user stack usage</td>
<td>80 bytes</td>
<td></td>
</tr>
<tr>
<td>Maximum interrupt stack usage</td>
<td>0 bytes</td>
<td></td>
</tr>
</tbody>
</table>

Note: The required memory sizes vary depending on the C compiler version and compile options.
9.3 File Composition

Table 9.2 lists the Files Used in the Sample Code, Table 9.3 lists the Standard Include Files, and Table 9.4 and Table 9.5 list the Functions and Setting Values in the Reference Application Notes. Files generated by the integrated development environment are not included in this table.

### Table 9.2  Files Used in the Sample Code

<table>
<thead>
<tr>
<th>File Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>main.c</td>
<td>Main processing</td>
</tr>
<tr>
<td>dsad.c</td>
<td>Functions for DSAD gain calibration and temperature compensation</td>
</tr>
<tr>
<td>dsad.h</td>
<td>Header file for dsad.c</td>
</tr>
<tr>
<td>temps.c</td>
<td>Temperature sensor processing</td>
</tr>
<tr>
<td>temps.h</td>
<td>Header file for temps.c</td>
</tr>
</tbody>
</table>

### Table 9.3  Standard Include Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>stdbool.h</td>
<td>Defines macros associated with Boolean and its value.</td>
</tr>
<tr>
<td>stdint.h</td>
<td>Defines macros declaring the integer type with the specified width.</td>
</tr>
<tr>
<td>float.h</td>
<td>Defines various limit values relating to the limits of floating-point numbers.</td>
</tr>
<tr>
<td>machine.h</td>
<td>Defines types of intrinsic functions for the RX Family.</td>
</tr>
</tbody>
</table>

### Table 9.4  Functions and Setting Values in the Reference Application Note (RX21A Group Initial Setting)

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
<th>Setting Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>r_init_stop_module.c</td>
<td>R_INIT_StopModule()</td>
<td></td>
</tr>
<tr>
<td>r_init_stop_module.h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>r_init_non_existent_port.c</td>
<td>R_INIT_NonExistentPort()</td>
<td></td>
</tr>
<tr>
<td>r_init_non_existent_port.h</td>
<td></td>
<td>100-pin version is specified.</td>
</tr>
<tr>
<td>r_init_clock.c</td>
<td>R_INIT_Clock()</td>
<td></td>
</tr>
<tr>
<td>r_init_clock.h</td>
<td></td>
<td>Clock selection: No. 1 is specified. The PCLKD division ratio is changed to divide-by-16.</td>
</tr>
</tbody>
</table>

### Table 9.5  Functions and Setting Values in the Reference Application Note (RX Family Coding Example of Wait Processing by Software)

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
<th>Setting Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>r_delay.c</td>
<td>R_DELAY_Us(unsigned long us, unsigned long khz)</td>
<td>Wait time is specified.</td>
</tr>
<tr>
<td>r_delay.h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9.4 Option-Setting Memory

Table 9.6 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFS0</td>
<td>FFFF FF8Fh to FFFF FF8Ch</td>
<td>FFFF FFFFh</td>
<td>The IWDT is stopped after a reset.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The WDT is stopped after a reset.</td>
</tr>
<tr>
<td>OFS1</td>
<td>FFFF FF8Bh to FFFF FF88h</td>
<td>FFFF FFFFh</td>
<td>The voltage monitor 0 reset is disabled after a reset.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HOCO oscillation is disabled after a reset.</td>
</tr>
<tr>
<td>MDES</td>
<td>FFFF FF83h to FFFF FF80h</td>
<td>FFFF FFFFh</td>
<td>Little endian</td>
</tr>
</tbody>
</table>
9.5 Constants

Table 9.7 to Table 9.13 list the Constants Used in the Sample Code.

Table 9.7 Constants Used in the Sample Code (dsad.c)
* The constants listed in the table can be changed by the user.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>R10TYP</td>
<td>100e3</td>
<td>Designed value of the internal input resistor [Ω]</td>
</tr>
<tr>
<td>CHIP_VER</td>
<td>1</td>
<td>Selection of the device version used.</td>
</tr>
<tr>
<td>BGR_CIRCUIT</td>
<td>1</td>
<td>Enable/disable setting of the BGR circuit</td>
</tr>
<tr>
<td>VALID_CHANNEL</td>
<td>0x7F</td>
<td>Selection of channels used (1: Used, 0: Not used) * With this, channels are associated with the corresponding bits (ch0 to ch6 correspond to bit 0 to bit 6, respectively).</td>
</tr>
<tr>
<td>VREFDSH_VOLT</td>
<td>600</td>
<td>Reference voltage (mV)</td>
</tr>
<tr>
<td>TRIG_MS</td>
<td>1 / 25.0f * 32 * 128</td>
<td>Setting value of $\tau_{TRIG}$ (CMT1 cycle)</td>
</tr>
<tr>
<td>TOS_MS</td>
<td>1.0f / (25.0f / 8)</td>
<td>Setting value of $\tau_{OS}$ (3.125 MHz: DSADCLK divided by 8)</td>
</tr>
<tr>
<td>SENSOR_CALC</td>
<td>TRIG_MS / (TOS_MS * 256)</td>
<td>Constant used for calculating the analog input voltage based on the A/D conversion value.</td>
</tr>
</tbody>
</table>

Table 9.8 Constants Used in the Sample Code (dsad.h)
* The constants listed in the table can be changed by the user.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSAD_CH_NUM</td>
<td>7</td>
<td>Number of channels of the DSAD</td>
</tr>
<tr>
<td>DSAD_DIFFER_CH_NUM</td>
<td>4</td>
<td>Number of channels for the differential input</td>
</tr>
<tr>
<td>DSAD_SINGLE_CH_NUM</td>
<td>3</td>
<td>Number of channels for the single-ended input</td>
</tr>
<tr>
<td>DSAD_GAIN_NUM</td>
<td>7</td>
<td>Number of gains available in the DSAD</td>
</tr>
<tr>
<td>DSAD_DIFFER_GAIN_NUM</td>
<td>7</td>
<td>Number of gains available with the differential input channel</td>
</tr>
<tr>
<td>DSAD_SINGLE_GAIN_NUM</td>
<td>3</td>
<td>Number of gains available with the single-ended input channel</td>
</tr>
<tr>
<td>DSAD_DISCARD_CNT</td>
<td>2</td>
<td>Number of DSAD conversions to be discarded</td>
</tr>
<tr>
<td>DSAD_CNT_MAX</td>
<td>(DSAD_DISCARD_CNT+8)</td>
<td>Number of DSAD conversions (‘2’ for discarding + ‘8’ for calculating the average)</td>
</tr>
</tbody>
</table>

Table 9.9 Constants Used in the Sample Code (dsad.h)
* The constants listed in the table cannot be changed by the user.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>STA_DSAD_IDLE</td>
<td>0</td>
<td>DSAD in preparation</td>
</tr>
<tr>
<td>STA_DSAD_PATERN_0</td>
<td>1</td>
<td>DSAD conversion with $y_C$ [V]</td>
</tr>
<tr>
<td>STA_DSAD_PATERN_1</td>
<td>2</td>
<td>DSAD conversion with $y_B$ [V]</td>
</tr>
<tr>
<td>STA_DSAD_COMPESETE</td>
<td>3</td>
<td>DSAD conversion with 0 V</td>
</tr>
<tr>
<td>DSAD_GAIN_X1</td>
<td>0</td>
<td>Gain number to be selected</td>
</tr>
<tr>
<td>DSAD_GAIN_X2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DSAD_GAIN_X4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>DSAD_GAIN_X8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>DSAD_GAIN_X16</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>DSAD_GAIN_X32</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>DSAD_GAIN_X64</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
Table 9.10 Constant Used in the Sample Code (main.c)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMT_CYCLE_MS</td>
<td>610</td>
<td>A/D conversion cycle (163.84 \mu s \times 610 = \text{approx. 100 ms})</td>
</tr>
</tbody>
</table>

Table 9.11 Constants Used in the Sample Code (temps.c)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH_REF_TEMP</td>
<td>125</td>
<td>High reference temperature (\degree C)</td>
</tr>
<tr>
<td>ADCONV_IN_OPERATION</td>
<td>0xFFFF</td>
<td>A/D conversion value during A/D conversion being performed (invalid value)</td>
</tr>
<tr>
<td>SLOPE_COEFFICIENT_TEMP</td>
<td>((\text{HIGH_REF_TEMP} - \text{ORDINARY_REF_TEMP}) \times \text{TEMP_ACCURACY})</td>
<td>Temperature slope</td>
</tr>
<tr>
<td>ORDINARY_REF_TEMP_IN_ACC</td>
<td>ORDINARY_REF_TEMP \times \text{TEMP_ACCURACY}</td>
<td>Product of normal reference temperature (25\degree C) and temperature calculation accuracy</td>
</tr>
</tbody>
</table>

Table 9.12 Constants Used in the Sample Code (temps.h)

* The constants listed in the table can be changed by the user.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEL_PGAGAIN</td>
<td>GAIN_RANGE1</td>
<td>Selection of the PGA gain (^{(1)})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GAIN_RANGE0: (1.8 V \leq AVCC0 &lt; 2.7 V)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GAIN_RANGE1: (2.7 V \leq AVCC0 \leq 3.6 V)</td>
</tr>
<tr>
<td>AVCC_VOLTAGE</td>
<td>3.3</td>
<td>Voltage [V] applied to the AVCC0 pin (^{(1)})</td>
</tr>
<tr>
<td>VREF_VOLTAGE</td>
<td>3.3</td>
<td>Voltage [V] applied to the VREFH0 pin</td>
</tr>
<tr>
<td>ORDINARY_REF_TEMP</td>
<td>25</td>
<td>Normal reference temperature (\degree C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* When the setting value is 25, the normal reference temperature is recognized as 25\degree C.</td>
</tr>
<tr>
<td>TEMP_ACCURACY</td>
<td>10</td>
<td>Temperature calculation accuracy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Multiplying factor is specified. When the setting value is 10, the tenth place is included in calculations and when the setting value is 100, the hundredth place is included in calculations. Do not set a value other than multiplier of 10 or a negative value.</td>
</tr>
<tr>
<td>CNV_CNT_MAX</td>
<td>6</td>
<td>Number of samplings for calculating the average</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* When the setting value is 6, A/D conversion results are obtained six times. The minimum and maximum values are discarded from the sum of the conversion results. The average of the remaining four values is used as the A/D conversion value.</td>
</tr>
</tbody>
</table>

Note:

1. The conditions of an applied voltage to the AVCC0 pin and the PGA gain must be consistent. Otherwise the calculation result will not be correct.
Table 9.13 Constants Used in the Sample Code (temps.h)
* The constants listed in the table cannot be changed by the user.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAIN_RANGE0</td>
<td>00h</td>
<td>PGA gain: 1.8 V ≤ AVCC0 &lt; 2.7 V</td>
</tr>
<tr>
<td>GAIN_RANGE1</td>
<td>01h</td>
<td>PGA gain: 2.7 V ≤ AVCC0 ≤ 3.6 V</td>
</tr>
<tr>
<td>STA_AD_IDLE</td>
<td>0</td>
<td>A/D conversion status: Not operating</td>
</tr>
<tr>
<td>STA_AD_WAIT</td>
<td>1</td>
<td>A/D conversion status: Wait for completion of A/D conversion</td>
</tr>
<tr>
<td>STA_AD_FINISH</td>
<td>2</td>
<td>A/D conversion status: A/D conversion completed</td>
</tr>
<tr>
<td>TSCDR0_VALUE</td>
<td>(TEPSCONST.TSCDR0.BIT.TSCD)</td>
<td>TSCDR0 register value</td>
</tr>
<tr>
<td>TSCDR1_VALUE</td>
<td>(TEPSCONST.TSCDR1.BIT.TSCD)</td>
<td>TSCDR1 register value</td>
</tr>
<tr>
<td>TSCDR3_VALUE</td>
<td>(TEPSCONST.TSCDR3.BIT.TSCD)</td>
<td>TSCDR3 register value</td>
</tr>
<tr>
<td>HIGH_REF_POTENTIAL_VAL</td>
<td>Note 1</td>
<td>A/D conversion value of the high reference temperature (125°C)</td>
</tr>
</tbody>
</table>

Note:
1. The setting value differs depending on the PGA gain selected. The setting value for each PGA gain is shown below.

   When ‘GAIN_RANGE0’ is selected:
   \[(\text{uint16}_t) \times (1.8 / \text{VREF_VOLTAGE} \times \text{TSCDR0\_VALUE})\]

   When ‘GAIN_RANGE1’ is selected:
   \[(\text{uint16}_t) \times ((2.7 / \text{VREF_VOLTAGE} \times \text{TSCDR1\_VALUE}) + ((3.3 / \text{VREF_VOLTAGE} \times \text{TSCDR3\_VALUE}) - (2.7 / \text{VREF_VOLTAGE} \times \text{TSCDR1\_VALUE})) \times (\text{AVCC\_VOLTAGE} - 2.7) / 0.6)\]
### 9.6 Variables

Table 9.14 lists the Global Variables (dsad.c), Table 9.15 to Table 9.17 list the static Variables, and Table 9.18 and Table 9.19 list the const Variables.

#### Table 9.14 Global Variables (dsad.c)

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>int32_t</td>
<td>g_dsad_data[DSAD_</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CH_NUM]</td>
<td>Areas for ch0 to ch6 to store the DSAD conversion value after calibration and compensation</td>
<td>measure_dsad_calib</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>measure_dsad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_DSAD_DSADI0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_DSAD_DSADI1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_DSAD_DSADI2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_DSAD_DSADI3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_DSAD_DSADI4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_DSAD_DSADI5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_DSAD_DSADI6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dsad_init</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R_DSAD_Calibration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>measure_dsad_calib</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>measure_dsad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excep_CMT1_CMI1</td>
</tr>
<tr>
<td></td>
<td>sel_ch_gain[DSAD_</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CH_NUM]</td>
<td>Area to store the specified gain. The value stored is used as the gain value. When a change is required, rewrite the value.</td>
<td>dsad_init</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R_DSAD_Calibration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>measure_dsad_calib</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>measure_dsad</td>
</tr>
<tr>
<td>volatile float</td>
<td>g_compensated_gain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[DSAD_CH_NUM]</td>
<td>System gain of the whole system after temperature compensation for each DSAD channel and gain setting</td>
<td>R_DSAD_CompensatedGain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>measure_dsad</td>
</tr>
</tbody>
</table>

#### Table 9.15 static Variables (main.c)

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>static const</td>
<td>valid_dsad_channel</td>
<td>Indicates whether each channel is available or not. Specify an appropriate value according to the user system.</td>
<td>main</td>
</tr>
<tr>
<td>bool</td>
<td>[DSAD_CH_NUM]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>static volatile</td>
<td>cnt_cycle</td>
<td>Counter for the A/D conversion cycle</td>
<td>Excep_CMT1_CMI1</td>
</tr>
<tr>
<td>uint16_t</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 9.16 static Variables (temps.c)

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int16_t</td>
<td>high_ref_potential</td>
<td>A/D conversion value (= CAL125) of the high reference temperature (125°C)</td>
<td>temps_init temps_calibration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static volatile uint16_t</td>
<td>slope_potential</td>
<td>A/D conversion slope</td>
<td>temps_calibration temps_calc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static volatile int16_t</td>
<td>ordinary_potential</td>
<td>A/D conversion value (= CAL25) of the normal reference temperature (25°C)</td>
<td>temps_calibration temps_calc</td>
</tr>
<tr>
<td>static volatile uint8_t</td>
<td>ad_status</td>
<td>A/D conversion status</td>
<td>main temps_get_ad_status temps_calibration temps_measurement temps_measurement Excep_AD_ADI</td>
</tr>
<tr>
<td>static volatile int16_t</td>
<td>now_temp</td>
<td>Current temperature calculated</td>
<td>temps_get_now_temp temps_get_now_temp Excep_AD_ADI</td>
</tr>
<tr>
<td>static volatile uint16_t</td>
<td>now_potential</td>
<td>Current A/D conversion value</td>
<td>temps_calibration Excep_AD_ADI</td>
</tr>
<tr>
<td>static volatile uint16_t</td>
<td>buf_ad_value[CNT_CNT_MAX]</td>
<td>Buffer for the A/D conversion value</td>
<td>Excep_AD_ADI</td>
</tr>
<tr>
<td>static volatile uint8_t</td>
<td>ad_smp_cnt</td>
<td>Pointer for writing the A/D conversion value</td>
<td>Excep_AD_ADI</td>
</tr>
<tr>
<td>static volatile int16_t</td>
<td>ad_max_value</td>
<td>A/D conversion maximum value</td>
<td>Excep_AD_ADI</td>
</tr>
<tr>
<td>static volatile int16_t</td>
<td>ad_min_value</td>
<td>A/D conversion minimum value</td>
<td>Excep_AD_ADI</td>
</tr>
</tbody>
</table>
### Table 9.17  static Variables (dsad.c)

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>static volatile</td>
<td>uint16_t dsad_smp_cnt[DSAD_CH_NUM]</td>
<td>Number of times for reading the DSAD conversion result</td>
<td>Excep_DSAD_DSADI0, Excep_DSAD_DSADI1, Excep_DSAD_DSADI2</td>
</tr>
<tr>
<td>static int32_t</td>
<td>dsad_data_sum [DSAD_CH_NUM]</td>
<td>Area used to sum the DSAD conversion results</td>
<td>Excep_DSAD_DSADI3, Excep_DSAD_DSADI4, Excep_DSAD_DSADI5, Excep_DSAD_DSADI6</td>
</tr>
<tr>
<td>static uint16_t</td>
<td>dsad_comp_fin</td>
<td>Flag to check whether the DSAD value in each register has been read and the average of the values has been obtained.</td>
<td>measure_dsad_calib, measure_dsad, Excep_DSAD_DSADI0, Excep_DSAD_DSADI1, Excep_DSAD_DSADI2, Excep_DSAD_DSADI3, Excep_DSAD_DSADI4, Excep_DSAD_DSADI5, Excep_DSAD_DSADI6</td>
</tr>
<tr>
<td>static uint16_t</td>
<td>dsad_comp_status</td>
<td>Information of the status to check the progress of the calibration and compensation</td>
<td>measure_dsad_calib, R_DSAD_Calibration, R_DSAD_InternalCompensatedGain</td>
</tr>
<tr>
<td>static int32_t</td>
<td>dsad_comp_data [DSAD_CH_NUM] [DSAD_GAIN_NUM+2]</td>
<td>Area to store the averaged DSAD conversion result before calibration and compensation [ch][0]: Measurement result at yC [ch][1]: Measurement result at yB [ch][2] to [9]: Measurement result at 0V for each gain</td>
<td>R_DSAD_Calibration, measure_dsad_calib, measure_dsad, R_DSAD_InternalCalibration, R_DSAD_Calibration, R_DSAD_InternalCompensatedGain</td>
</tr>
<tr>
<td>static volatile float</td>
<td>coef_temp_quad</td>
<td>Quadratic coefficient of the temperature characteristics for temperature compensation</td>
<td>R_DSAD_InternalCompensatedGain, R_DSAD_Calibration, R_DSAD_InternalCompensated</td>
</tr>
<tr>
<td>static volatile float</td>
<td>coef_temp_linear [DSAD_CH_NUM] [DSAD_GAIN_NUM]</td>
<td>Linear coefficient of the temperature characteristics for temperature compensation</td>
<td>R_DSAD_InternalCompensatedGain, R_DSAD_Calibration, R_DSAD_InternalCompensated</td>
</tr>
<tr>
<td>static volatile float</td>
<td>device_gain [DSAD_CH_NUM] [DSAD_GAIN_NUM]</td>
<td>Device gain for DSAD channels with each gain setting at 25°C</td>
<td>R_DSAD_InternalCalibration, R_DSAD_Calibration, R_DSAD_InternalCompensatedGain</td>
</tr>
<tr>
<td>static volatile float</td>
<td>sensor_gain [DSAD_CH_NUM]</td>
<td>Sensor gain for DSAD channels with each gain setting at 25°C in an external circuit such as a sensor</td>
<td>R_DSAD_Calibration, R_DSAD_InternalCompensatedGain</td>
</tr>
<tr>
<td>static volatile float</td>
<td>system_gain [DSAD_CH_NUM] [DSAD_GAIN_NUM]</td>
<td>System gain for DSAD channels with each gain setting at 25°C in the whole system gain including the sensor</td>
<td>R_DSAD_Calibration, R_DSAD_InternalCompensatedGain, measure_dsad</td>
</tr>
</tbody>
</table>
### Table 9.18  const Variables (main.c)

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>const float</td>
<td>g_dsad_ext_load_res [DSAD_SINGLE_CH_NUM]</td>
<td>Value [Ω] of the external load resistor for single-ended input channels (channels 4 to 6). Specify an appropriate value according to the user system.</td>
<td>main R_DSAD_Internal Compensated</td>
</tr>
</tbody>
</table>

### Table 9.19  const Variables (dsad.c)

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>static const float</td>
<td>dsad_data_volt[3] [DSAD_CH_NUM]</td>
<td>Voltage [mV] when calibrating, compensating, or measuring an offset</td>
<td>R_DSAD_Calibration</td>
</tr>
<tr>
<td>static const float</td>
<td>typ_zi[DSAD_SINGLE_GAIN_NUM]</td>
<td>Typical value [Ω] of the input impedance (x1, x2, and x4) of the single-ended input. Refer to the ΔΣ A/D Conversion Characteristics section in the User’s Manual: Hardware for details.</td>
<td>R_DSAD_Internal Compensated</td>
</tr>
<tr>
<td>static const float</td>
<td>coef_temp_cba</td>
<td>Quadratic coefficient of the temperature characteristics for the on-chip BGR. The coefficient value is listed in Table 7.1.</td>
<td>R_DSAD_Internal Compensated</td>
</tr>
<tr>
<td>static const float</td>
<td>coef_temp_cbb</td>
<td>Linear coefficient of the temperature characteristics for the on-chip BGR. The coefficient value is listed in Table 7.1.</td>
<td>R_DSAD_Internal Compensated</td>
</tr>
<tr>
<td>static const float</td>
<td>coef_temp_cxn[DSAD_GAIN_NUM]</td>
<td>Coefficient of the temperature characteristics for the device gain. The coefficient value is listed in Table 7.1.</td>
<td>R_DSAD_Internal Compensated</td>
</tr>
<tr>
<td>static const float</td>
<td>coef_temp_cz</td>
<td>Coefficient of the temperature characteristics for the input impedance. The coefficient value is listed in Table 7.1.</td>
<td>R_DSAD_Internal Calibration</td>
</tr>
<tr>
<td>static const float</td>
<td>gain_val[DSAD_GAIN_NUM]</td>
<td>Gain amplification</td>
<td>R_DSAD_Internal Calibration</td>
</tr>
</tbody>
</table>
## 9.7 Functions

Table 9.20 lists the Functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>Main processing</td>
<td>main.c</td>
</tr>
<tr>
<td>peripheral_init</td>
<td>Peripheral function initialization</td>
<td>main.c</td>
</tr>
<tr>
<td>cmt_init</td>
<td>CMT1 initialization</td>
<td>main.c</td>
</tr>
<tr>
<td>irq_init</td>
<td>IRQ2 initialization</td>
<td>main.c</td>
</tr>
<tr>
<td>Excep_CMT1_CMI1</td>
<td>Compare match 1 interrupt handler</td>
<td>main.c</td>
</tr>
<tr>
<td>dsad_init</td>
<td>DSAD initialization</td>
<td>dsad.c</td>
</tr>
<tr>
<td>dsad_start</td>
<td>DSAD conversion start processing</td>
<td>dsad.c</td>
</tr>
<tr>
<td>R_DSAD_InternalCalibration</td>
<td>Coefficient initialization for gain calibration</td>
<td>dsad.c</td>
</tr>
<tr>
<td>R_DSAD_InternalCompensated</td>
<td>Coefficient initialization for gain temperature</td>
<td>dsad.c</td>
</tr>
<tr>
<td>R_DSAD_Calibration</td>
<td>System gain calibration</td>
<td>dsad.c</td>
</tr>
<tr>
<td>R_DSAD_CompensatedGain</td>
<td>Temperature compensation for the system gain</td>
<td>dsad.c</td>
</tr>
<tr>
<td>measure_dsad_calib</td>
<td>Obtaining DSAD conversion result at calibration</td>
<td>dsad.c</td>
</tr>
<tr>
<td>measure_dsad</td>
<td>Obtaining DSAD conversion result</td>
<td>dsad.c</td>
</tr>
<tr>
<td>Excep_DSAD_DSADI</td>
<td>DSAD conversion interrupt handler</td>
<td>dsad.c</td>
</tr>
<tr>
<td>temps_init</td>
<td>A/D converter and temperature sensor initializations</td>
<td>temps.c</td>
</tr>
<tr>
<td>temps_close</td>
<td>A/D converter and temperature sensor stop processing</td>
<td>temps.c</td>
</tr>
<tr>
<td>temps_get_ad_status</td>
<td>Obtaining A/D conversion status</td>
<td>temps.c</td>
</tr>
<tr>
<td>temps_get_potential</td>
<td>Obtaining temperature sensor measurement result</td>
<td>temps.c</td>
</tr>
<tr>
<td>temps_get_now_temp</td>
<td>Obtaining current temperature</td>
<td>temps.c</td>
</tr>
<tr>
<td>temps_calibration</td>
<td>Temperature sensor calibration processing</td>
<td>temps.c</td>
</tr>
<tr>
<td>temps_measurement</td>
<td>Temperature sensor measurement processing</td>
<td>temps.c</td>
</tr>
<tr>
<td>temps_calc</td>
<td>Current temperature calculation</td>
<td>temps.c</td>
</tr>
<tr>
<td>Excep_AD_ADI</td>
<td>A/D conversion end interrupt handler</td>
<td>temps.c</td>
</tr>
</tbody>
</table>
9.8 Function Specifications

The following tables list the sample code function specifications.

<table>
<thead>
<tr>
<th>main</th>
<th>Outline</th>
<th>Main processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Declaration</td>
<td>void main(void)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>After the clock initialization, performs calibration of DSAD conversion, compensation by temperature, and calibration for the temperature sensor. Then performs DSAD conversion every 1.6384 ms and A/D conversion of the temperature sensor output every 100 ms. Compensates the DSAD conversion result by temperature using the temperature sensor output as needed.</td>
<td></td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>peripheral_init</th>
<th>Outline</th>
<th>Peripheral initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Declaration</td>
<td>static void peripheral_init(void)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Initializes the peripheral functions used.</td>
<td></td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cmt_init</th>
<th>Outline</th>
<th>CMT1 initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Declaration</td>
<td>static void cmt_init(void)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Initializes CMT1.</td>
<td></td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>irq_init</th>
<th>Outline</th>
<th>IRQ initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Declaration</td>
<td>static void irq_init(void)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Initializes IRQ2.</td>
<td></td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
Excep_CMT1_CMI1

Outline
Compare match 1 interrupt handler

Header
None

Declaration
static void Excep_CMT1_CMI1(void)

Description
Executes the interrupt handler every 163.84 µs. The counter is incremented every time an interrupt request is generated. When the counter reaches 610 times (approx. 100 ms), performs a temperature measurement. The compare match interrupt is used as the start trigger for the DSAD channels via the ELC.

Arguments
None

Return Value
None

dsad_init

Outline
DSAD initialization

Header
dsad.h

Declaration
void dsad_init(void)

Description
Initializes the DSAD converter.

Arguments
None

Return Value
None

dsad_start

Outline
DSAD conversion start processing

Header
dsad.h

Declaration
void dsad_start(void)

Description
Starts operating the DSAD converter.

Arguments
None

Return Value
None

R_DSAD_InternalCalibration

Outline
Coefficient initialization for gain calibration

Header
dsad.h

Declaration
void R_DSAD_InternalCalibration(uint16_t channel)

Description
Prepares intermediate calculation results necessary for gain calibration.

Arguments
unit16_t channel: Input channel (0 to 6)

Return Value
None

Remarks
Execute this function before executing the R_DSAD_Calibration and R_DSAD_CompensatedGain functions. Otherwise calibration and compensation cannot be performed correctly.

If the constant for the device version is specified as a version other than G, 32768 is used instead of the DSADIIC register value and 47974 is used instead of the DSADGmXn register without reading these registers.
### R_DSAD_InternalCompensated

**Outline**  
Coefficient initialization for gain temperature compensation

**Header**  
dsad.h

**Declaration**  
void R_DSAD_InternalCompensated(uint16_t channel,  
const float dsad_ext_load_res[DSAD_SINGLE_CH_NUM])

**Description**  
Prepares intermediate calculation results necessary for temperature compensation for gain.

**Arguments**  
unit16_t channel:  
Input channel (0 to 6)

const float dsad_ext_load_res[DSAD_SINGLE_CH_NUM]:  
External load resistor [Ω] for the single-ended input channels (channels 4 to 6).

**Return Value**  
None

**Remarks**  
Execute this function before executing the R_DSAD_Calibration and R_DSAD_CompensatedGain functions. Otherwise calibration and compensation cannot be performed correctly.

If the constant for the device version is specified as a version other than G version, 32768 is used without reading the DSADIIC register.

### R_DSAD_Calibration

**Outline**  
System gain calibration

**Header**  
dsad.h

**Declaration**  
void R_DSAD_Calibration(uint16_t channel)

**Description**  
Calculates the gain for the specified channel before temperature compensation.

**Arguments**  
unit16_t channel:  
Input channel (0 to 6)

**Return Value**  
None

**Remarks**  
Execute the R_DSAD_InternalCalibration and R_DSAD_InternalCompensated functions before executing this function. Otherwise calibration and compensation cannot be performed correctly.

### R_DSAD_CompensatedGain

**Outline**  
Temperature compensation for the system gain

**Header**  
r_dsad_compensate.h

**Declaration**  
void R_DSAD_CompensatedGain(uint16_t channel, int16_t junction_temp)

**Description**  
Calculates the system gain for the specified channel after temperature compensation.

**Arguments**  
uint16_t channel:  
Input channel (0 to 6)

int16_t junction_temp:  
The device temperature measured by the temperature sensor.  
The value should be from -40°C to +105°C.

**Return Value**  
None

**Remarks**  
Execute the R_DSAD_InternalCalibration, R_DSAD_InternalCompensated, and R_DSAD_Calibration functions before executing this function. Otherwise calibration and compensation cannot be performed correctly.
### measure_dsad_calib

<table>
<thead>
<tr>
<th>Outline</th>
<th>Obtaining DSAD conversion result at calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>dsad.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void measure_dsad_calib(void)</td>
</tr>
<tr>
<td>Description</td>
<td>This function is used for calibration. Obtains measurement results of voltages at 2 points, which are used for calculating the sensor gain, obtains measurement results of voltages at 0 V for gains x2 to x64, and transfers the results to the RAM.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### measure_dsad

<table>
<thead>
<tr>
<th>Outline</th>
<th>Obtaining DSAD conversion result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>dsad.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void measure_dsad(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Performs calibration and compensation for the DSAD conversion results and transfer the processed result to the RAM.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### Excep_DSAD_DSADI\textsubscript{m} (m = 0 to 6)

<table>
<thead>
<tr>
<th>Outline</th>
<th>DSAD conversion interrupt handler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>dsad.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void Excep_DSAD_DSADI\textsubscript{0}(void)</td>
</tr>
<tr>
<td></td>
<td>static void Excep_DSAD_DSADI\textsubscript{1}(void)</td>
</tr>
<tr>
<td></td>
<td>static void Excep_DSAD_DSADI\textsubscript{2}(void)</td>
</tr>
<tr>
<td></td>
<td>static void Excep_DSAD_DSADI\textsubscript{3}(void)</td>
</tr>
<tr>
<td></td>
<td>static void Excep_DSAD_DSADI\textsubscript{4}(void)</td>
</tr>
<tr>
<td></td>
<td>static void Excep_DSAD_DSADI\textsubscript{5}(void)</td>
</tr>
<tr>
<td></td>
<td>static void Excep_DSAD_DSADI\textsubscript{6}(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Updates the number of times for obtaining conversion results, stores the DSAD conversion result in the RAM (as described below), and clears the interrupt request. Ten DSAD conversions are treated as one sampling. First two conversion results are discarded and the subsequent results are stored in the RAM. When the DSAD result has been obtained 10 times, the average of the eight results is calculated. The average is treated as the DSAD conversion result.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### temps_init

<table>
<thead>
<tr>
<th>Outline</th>
<th>A/D converter and temperature sensor initializations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>temps.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void temps_init(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Initializes the A/D converter and the temperature sensor.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>
temps_close

Outline | A/D converter and temperature sensor stop processing
---|---
Header | temps.h
Declaration | static void temps_close(void)
Description | Stops the A/D converter and the temperature sensor.
Arguments | None
Return Value | None

temps_get_ad_status

Outline | Obtaining A/D conversion status
---|---
Header | temps.h
Declaration | uint8_t temps_get_ad_status(void)
Description | Obtains the current A/D conversion status.
Arguments | None
Return Value | uint8_t: A/D conversion status
- STA_AD_IDLE: Not operating
- STA_AD_WAIT: Waiting for completion of A/D conversion
- STA_AD_FINISH: A/D conversion completed

temps_get_potential

Outline | Obtaining temperature sensor measurement result
---|---
Header | None
Declaration | static uint16_t temps_get_potential (void)
Description | Obtains the measured A/D conversion value.
Arguments | None
Return Value | uint16_t: A/D conversion value of the temperature sensor
- ADCONV_IN_OPERATION: A/D conversion in process
- Other than above: A/D conversion value

temps_get_now_temp

Outline | Obtaining current temperature
---|---
Header | temps.h
Declaration | int16_t temps_get_now_temp (void)
Description | Obtains the current temperature.
Arguments | None
Return Value | int16_t: Current temperature

temps_calibration

Outline | Temperature sensor calibration processing
---|---
Header | temps.h
Declaration | void temps_calibration(void)
Description | Obtains the A/D conversion value at the normal reference temperature and stores it in the RAM.
Arguments | None
Return Value | None
### temps_measurement

<table>
<thead>
<tr>
<th>Outline</th>
<th>Temperature sensor measurement processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>temps.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void temps_measurement(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Starts measuring the current temperature.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### temps_calc

<table>
<thead>
<tr>
<th>Outline</th>
<th>Current temperature calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static int16_t temps_calc(uint16_t w_now_potential)</td>
</tr>
<tr>
<td>Description</td>
<td>Calculates temperature from the A/D conversion value passed with the argument.</td>
</tr>
<tr>
<td>Arguments</td>
<td>uint16_t w_now_potential: A/D conversion value</td>
</tr>
<tr>
<td>Return Value</td>
<td>int16_t: Current temperature [°C]</td>
</tr>
</tbody>
</table>

### Excep_AD_ADI

<table>
<thead>
<tr>
<th>Outline</th>
<th>A/D conversion end interrupt handler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void Excep_AD_ADI(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Stores the A/D conversion value in the RAM when an A/D conversion has been completed. At the completion of the sixth A/D conversion, subtracts the maximum and minimum values from the sum of six conversion results, divides the subtraction result by 4 to get the average, and calculates the temperature based on the average.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>
9.9 Flowcharts

9.9.1 Main Processing

Figure 9.2 and Figure 9.3 show the Main Processing.

```plaintext
main

I flag ← 0

Disable maskable interrupts

Stop processing for active peripheral functions after a reset
R_INIT_StopModule()

Nonexistent port initialization
R_INIT_NonExistentPort()

Clock initialization
R_INIT_Clock()

Peripheral function initialization
peripheral_init()

Enable maskable interrupts

I flag ← 1

Starts CMT1 count

DSAD conversion start processing
dsad_start()

CMSTR0 register
STR1 bit ← 1: CMT1.CMCNT count is started.

DSAD measurement for calibration
measure_dsad_calib()

Measures the DSAD for calibration (first time: yC)

Wait for calibration and compensation preparation

Reads the IR066.IR flag (wait until it becomes 1).

Clear the IRQ2 interrupt request

IR066 register
IR flag ← 0: No IRQ2 interrupt request is generated.

DSAD measurement for calibration
measure_dsad_calib()

Measures the DSAD for calibration (second time: yB)

Temperature sensor calibration processing
temps_calibration()

Figure 9.2 Main Processing (1/2)
```
Is the state of the channel ‘used’?

Yes

Coefficient initialization for gain calibration
R_DSAD_InternalCalibration()

Coefficient initialization for temperature compensation
R_DSAD_InternalCompensated()

System gain calibration
R_DSAD_Calibration

Has calibration for all channels been completed?

No

Obtaining the A/D conversion status
temps_get_ad_status()

Has A/D conversion been completed?

No

Obtaining the current temperature
temps_get_ad_status()

Is the state of the channel ‘used’?

Yes

Temperature compensation for the system gain
R_DSAD_CompensatedGain()

Has calibration for all channels been completed?

No

DSAD measurement
measure_dsad()
9.9.2 Peripheral Function Initialization
Figure 9.4 shows the Peripheral Function Initialization.

```
peripheral_init

DSAD initialization
dsad_init()

CMT initialization
cmt_init()

IRQ initialization
irq_init()

return
```

Figure 9.4 Peripheral Function Initialization

9.9.3 CMT1 Initialization
Figure 9.5 shows the CMT1 Initialization.

```
cmt_init

Disable write protection

Cancel the module-stop state for CMT1

Enable write protection

Stop CMT1 count

Select the count source

Clear the CMT1 counter

Specify the CMT1 compare match constant register

Clear the interrupt request

Specify the interrupt settings

return

PRCR register ← A502h
PRC1 bit = 1: Enable writing to the registers related to the low power consumption function.

MSTPCRA register
MSTPA15 bit ← 0: The module-stop state is canceled for CMT1.

PRCR register ← A500h
PRC1 bit = 0: Disable writing to the registers related to the low power consumption function.

CMSTR0 register
STR1 bit ← 0: CMT1.CMCNT count is stopped.

CMT1.CMCR register ← 00C1h
CMIE bit = 1: Compare match interrupt (CMIn) enabled
CKS[1:0] bits = 01b: PCLK/32

CMT1.CMCOR register ← 128 - 1: CMT cycle = 163.84 μs[(128 - 1) × (PCLKB/32)]

IR029 register
IR flag ← 0: No CMI1 interrupt request is generated.

IPR005 register
IPR[3:0] bits ← 0001b: CMI1 interrupt priority level 1
IER03 register
IEN5 bit ← 1: CMI1 interrupt request is enabled.
```

Figure 9.5 CMT1 Initialization
### 9.9.4 IRQ2 Initialization

Figure 9.6 shows the IRQ2 Initialization.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. irq_init</td>
<td>Disable an interrupt</td>
</tr>
<tr>
<td>2. Disable the digital filter</td>
<td></td>
</tr>
</tbody>
</table>
| 3. Specify the digital filter sampling clock | IER08 register
|  | IEN2 bit ← 0: IRQ2 interrupt is disabled.
| 4. Enable writing to the PWPR register | IRQFLTE0 register
|  | FLTEN2 bit ← 0: Digital filter is disabled.
| 5. Set the P32PFS register | IRQFLTC0 register
|  | ISEL bit ← 1: Used as IRQ2 input pin.
| 6. Enable writing to the PWPR register | PORT3.PMR register
|  | B2 bit ← 0: Use the pin as a general I/O pin.
| 7. Enable writing to the PFS register | PWPR register
|  | B0WI bit ← 0: Writing to the PFSWE bit is enabled.
| 8. Set the P32PFS register | PWPR register
|  | PFSWE bit ← 1: Writing to the PFS register is enabled.
| 9. Disable writing to the PWPR register | P32PFS register
|  | B2 bit ← 0: Use the pin as a general I/O pin.
| 10. Enable writing to the PFS register | PWPR register
|  | PFSWE bit ← 0: Writing to the PFS register is disabled.
| 11. Disable writing to the PWPR register | PWPR register
|  | B0WI bit ← 1: Writing to the PFSWE bit is disabled.
| 12. Specify the IRQ detection setting | IRQCR2 register
|  | IRQMD[1:0] bits ← 01b: Falling edge
| 13. Clear the interrupt request | IR066 register
|  | IR flag ← 0: No IRQ2 interrupt request is generated.
| 14. Enable the digital filter | IRQFLTE0 register
|  | FLTEN2 bit ← 1: Digital filter is enabled.

**Figure 9.6** IRQ2 Initialization
### 9.9.5 Compare Match 1 Interrupt Handler

Figure 9.7 shows the Compare Match 1 Interrupt Handler.

```
Excep_CMT1_CMI1

Increment the A/D conversion cycle counter
cnt_cycle++

Has the time set in CMT_CYCLE_MS elapsed?
Yes

Clear the A/D conversion cycle counter
cnt_cycle ← 0

Temperature sensor measurement processing temps_measurement()

No

return
```

Figure 9.7  Compare Match 1 Interrupt Handler
9.9.6 DSAD Initialization

Figure 9.8 shows the DSAD Initialization for system gain calibration.

```
<table>
<thead>
<tr>
<th>dsad_init</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable write protection</td>
</tr>
<tr>
<td>Cancel the module-stop state</td>
</tr>
<tr>
<td>for DSAD and ELC</td>
</tr>
<tr>
<td>Enable write protection</td>
</tr>
<tr>
<td>Specify BGO</td>
</tr>
<tr>
<td>(internal/external) setting</td>
</tr>
<tr>
<td>Start operating the PGA</td>
</tr>
<tr>
<td>and the ΔΣ modulator</td>
</tr>
<tr>
<td>Specify a gain</td>
</tr>
<tr>
<td>Wait 5 ms R_DELAY_Us()</td>
</tr>
<tr>
<td>Reset the PGA and the ΔΣ</td>
</tr>
<tr>
<td>modulator</td>
</tr>
<tr>
<td>Wait for the PGA and the ΔΣ</td>
</tr>
<tr>
<td>modulator to be reset</td>
</tr>
<tr>
<td>Clear the interrupt request</td>
</tr>
<tr>
<td>Specify the interrupt settings</td>
</tr>
<tr>
<td>Enable DSAD conversion</td>
</tr>
<tr>
<td>and interrupt</td>
</tr>
<tr>
<td>return</td>
</tr>
</tbody>
</table>
```

PRCR register ← A502h
PRC1 bit = 1: Enable writing to the registers related to the low power consumption function.

MSTPCA25 register
MSTP25 bit ← 0: The module-stop state is canceled for DSAD.
MSTPCRB register
MSTPB9 bit ← 0: The module-stop state is canceled for ELC.

PRCR register ← A500h
PRC1 bit = 0: Disable writing to the registers related to the low power consumption function.

DSADCR register
EXREF bit ← 0: Generates reference voltage from on-chip BGR circuit.
BGRE bit ← 1: Activates BGR circuit.

DSADRm register (m = 0 to 6)
ADSE bit = 1: Activates PGA and ΔΣ modulator.

DSADGSRm register (m = 0 to 6)
GAIN bit: A gain is selected.

Waits for the reference voltage to be ready, and the PGA and the ΔΣ modulator to be activated.

DSADRSTR register
DSRST bit ← 1: PGA and ΔΣ modulator are under initialization.

Reads the DSADRSTR register
DSRST bit: 0: Initialization completed 1: Under initialization

IR206 to IR213 registers
IR flag ← 0: No DSADI interrupt is generated.

IPR206 to IPR213 registers
IPR[3:0] bits ← 0002b: DSADI interrupt priority level 2
IER19 register
IEN6 bit ← 0: DSADORI interrupt request is disabled.
IER19 register
IEN7 bit ← 1: DSADI interrupt request is enabled.
IER1A register
Bits IEN0 to IEN5 ← 1: DSADI interrupt request is enabled.

DSADCSTRm register (m = 0 to 6)
ADIE bit = 1: Enables DSADI interrupt at A/D conversion end.
9.9.7 DSAD Conversion Start Processing

Figure 9.9 shows the DSAD Conversion Start Processing.

- `dsad_start`
- Enable start trigger for conversion
- Specify the event link
- Enable the event link
- Return

**DSADCSRm register (m = 0 to 6)**
- TGRE bit = 1: Enables A/D conversion start by trigger.

**ELSRm register (m = 30 to 36)**
- 1Fh: CMT1, compare match 1 signal

**ELCR register**
- ELCN bit = 1: Linkage of all the event is enabled.
9.9.8 Coefficient Initialization for Gain Calibration

Figure 9.10 shows the Coefficient Initialization for Gain Calibration.

**Arguments:**
- uint16_t channel: Input channel
- g_dsad_ext_load_res: Value of the external load resistor

**Calculating the calibration coefficient for calibration with resistor**

1. **Calculate** the calibration coefficient for calibration with resistor.

   - **When CHIP_VER = 1 (G version is selected)**
     - Reads the DSADIIIC register and calculates the coefficient.
     - Calibration coefficient = \( \text{DSADIIIC} \times \frac{1}{32768} \)

2. **Make the address of the \( \Delta \Sigma \) A/D gain calibration data register.

   - Calculates the start address based on the DSADG0X1 register for each channel.

3. **Calculate device gains for gains x1 to x4 using the calibration data**

   - Reads the DSADGmXn register and calculates the device gain.
   - Calibration coefficient = \( \text{Gain value} \times \frac{\text{DSADGmXn.GCD}[15:0]}{47971} \)

4. **Update the address of the calibration data register for the gain to be calculated**

5. **Have gains x1 to x4 all been calculated?**

   - **Yes**
     - **Have all gains for channels 0 to 3 been calculated?**
       - **Yes**
       - Compensate device gains for gains x16 with input resistor value
       - Compensate device gains for gains x32 with input resistor value
       - Calculate the device gain for gain x64
         - Calculates the gain using the value of gain x32.

   - **No**
     - Specify the gain value

6. **When CHIP_VER = 0 (chip version other than G version is selected)**

7. **Have gain settings for x1 to x64 been completed?**

   - **Yes**
     - Return
   - **No**
     - Specify the gain value

**Note:**
1. Change the parameter according to the system used.
9.9.9 Coefficient Initialization for Gain Temperature Compensation

Figure 9.11 shows the Coefficient Initialization for Gain Temperature Compensation.

Figure 9.11 Coefficient Initialization for Gain Temperature Compensation

Note:
1. Change the parameter according to the system used.
9.9.10 System Gain Calibration

Figure 9.12 shows the System Gain Calibration.

- Calculate the sensor gain at 25°C
- Calculate the temperature compensation coefficient at the reference temperature.
- Calculate the system gain after temperature compensation based on the device gain and sensor gain for gains x1 to x4.
  - System gain = sensor gain \times device gain \times temperature compensation coefficient

Have gains x1 to x4 all been calculated?
  - No
  - Calculate the system gain after temperature compensation based on the device gain and sensor gain for gains x8 to x32.
  - System gain = sensor gain \times device gain \times temperature compensation coefficient

Have all gains for channels 0 to 3 been calculated?
  - No
  - Yes

Calculate the system gain after temperature compensation based on the device gain and sensor gain for gains x8 to x32.
  - System gain = sensor gain \times device gain \times temperature compensation coefficient

Have gains x8 to x32 all been calculated?
  - No
  - Yes

return
9.9.11 Temperature Compensation for the System Gain

Figure 9.13 shows the Temperature Compensation for the System Gain.

Figure 9.13  Temperature Compensation for the System Gain
9.9.12 Obtaining DSAD Conversion Result at Calibration

Figure 9.14 shows the Obtaining DSAD Conversion Result at Calibration.

```
measure_dsad_calib

Do the calibration status indicate 'before 2nd acquisition'?  No

Yes

DSAD conversion been completed for the channel used?  No

Clear the DSAD conversion end flag

Store the DSAD conversion value

Have the DSAD conversion values been obtained for all channels?  No

return

Specify the gain for which an offset is obtained

DSADGSRm register (m = 0 to 6)  GAIN[2:0] bits: Gain select

Is the gain set from ×8 to ×32?  Yes

No

Specify the gain for which an offset is obtained

DSADGSRm register (m = 0 to 3)  GAIN[2:0] bits: Gain select

Has DSAD conversion been completed for the channel used?  No

Clear the DSAD conversion end flag

Store the DSAD conversion value

Have DSAD conversion values been compensated for all channels?  No

Yes

DSADGSRm register (m = 0 to 6)  GAIN[2:0] bits: Gain select

Have DSAD conversion values been compensated for all gains?  No

Yes

Set the calibration status to 'completed'.

Reset the gain

return
```

Figure 9.14 Obtaining DSAD Conversion Result at Calibration
9.9.13 Obtaining DSAD Conversion Result

Figure 9.15 shows the Obtaining DSAD Conversion Result.

```
measure_dsad

Has DSAD conversion been completed for the channel used?

No

Clear the DSAD conversion end flag

Calibrate and compensate the DSAD conversion value

DSAD conversion value = (DSAD conversion value - offset value) / system gain × system gain after compensation

Have compensations for DSAD conversions been completed for all channels?

No

Yes

return
```

Figure 9.15 Obtaining DSAD Conversion Result
9.9.14 DSAD Conversion Interrupt Handler

Figure 9.16 shows the DSAD Conversion Interrupt Handler.

```
Excep_DSAD_DSADii (i = 0 to 6)

Count up the number of times for A/D conversion value acquisition

Has the A/D conversion value been obtained more than 3 times?
  No
  Yes
    Add the DSAD conversion value to the sum area buffer

Number of times for A/D conversion value acquisition = DSAD_CNT_MAX?
  No
  Yes
    Average the sum of the conversion values

Clear the sum area buffer and the number of A/D conversion value acquisition

Set the acquisition flag

Clear the interrupt request
  IR206 to IR213 registers
  IR flag ← 0: No DSADI interrupt is requested.

return
```

Figure 9.16 DSAD Conversion Interrupt Handler
9.9.15 A/D Converter and Temperature Sensor Initializations

Figure 9.17 shows the A/D Converter and Temperature Sensor Initializations.

```
<table>
<thead>
<tr>
<th>temps_init</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the A/D conversion end interrupt</td>
</tr>
<tr>
<td>Disable write protection</td>
</tr>
<tr>
<td>Cancel the module-stop state for A/D converter and temperature sensor</td>
</tr>
<tr>
<td>Enable write protection</td>
</tr>
<tr>
<td>Wait 2 μs R_DELAY_Us()</td>
</tr>
<tr>
<td>Set scan mode</td>
</tr>
<tr>
<td>Specify the A/D conversion pins</td>
</tr>
<tr>
<td>Specify temperature sensor output to be A/D converted</td>
</tr>
<tr>
<td>Specify the sampling time</td>
</tr>
<tr>
<td>Specify the A/D conversion start trigger setting with the temperature sensor</td>
</tr>
<tr>
<td>Select the PGA gain</td>
</tr>
<tr>
<td>Enable the temperature sensor</td>
</tr>
<tr>
<td>Wait the temperature sensor start time (80 μs) R_DELAY_Us()</td>
</tr>
<tr>
<td>Specify the scan end interrupt setting</td>
</tr>
<tr>
<td>Clear the interrupt request</td>
</tr>
<tr>
<td>Enable interrupts</td>
</tr>
<tr>
<td>Obtain the high reference temperature high_ref_potential ← HIGH_REF_POTENTIAL_VAL</td>
</tr>
</tbody>
</table>

Notes:
1. Change the parameter according to the system used.
2. A value set here varies depending on the PGA gain selected with the constant.
```

Figure 9.17 A/D Converter and Temperature Sensor Initializations
9.9.16 A/D Converter and Temperature Sensor Stop Processing

Figure 9.18 shows the A/D Converter and Temperature Sensor Stop Processing.

```
| temps_close |
| IER0C register |
| IEN2 bit ← 0: ADI interrupt request is disabled. |
| IPR08 register |
| IPR[3:0] bits ← 0000b: ADI interrupt priority level is 0. |
| IR098 register |
| IR flag ← 0: No ADI interrupt request is generated. |
| Disable the interrupt |
| Clear the interrupt request |
| Specify the scan end interrupt setting |
| Stop the temperature sensor |
| Cancel the A/D converter start trigger |
| Specify temperature sensor output not to be A/D converted |
| Disable write protection |
| Transition to the module-stop state is made for A/D converter and temperature sensor |
| Enable write protection |
| return |
```

Figure 9.18 A/D Converter and Temperature Sensor Stop Processing

9.9.17 Obtaining A/D Conversion Status

Figure 9.19 shows the Obtaining A/D Conversion Status.

```
| temps_get_ad_status |
| return (ad_status) |
```

Figure 9.19 Obtaining A/D Conversion Status
9.9.18 Obtaining Temperature Sensor Measurement Result
Figure 9.20 shows the Obtaining Temperature Sensor Measurement Result.

```plaintext
temps_get_potential

Does the PGA stop?

No

Obtain the A/D conversion value

w_now_potential ← ADTSDR register value

return(w_now_potential)

Yes

Reads the TSCR register
PGAEN bit: 0: Disables the PGA.
1: Enables the PGA.

Figure 9.20 Obtaining Temperature Sensor Measurement Result
```

9.9.19 Obtaining Current Temperature
Figure 9.21 shows the Obtaining Current Temperature.

```plaintext
temps_get_now_temp

return(now_temp)

Figure 9.21 Obtaining Current Temperature
```

9.9.20 Temperature Sensor Calibration Processing
Figure 9.22 shows the Temperature Sensor Calibration Processing.

```plaintext
temps_calibration

Wait for completion of A/D conversion

Set the A/D conversion status to 'not operating'.

ad_status ← STA_AD_IDLE

ordinary_potential ← now_potential

Stores the A/D conversion value at the normal reference temperature (25°C)

Slope_potential ← high_ref_potential - ordinary_potential

Stores the slope of the A/D conversion value in the RAM

return

Figure 9.22 Temperature Sensor Calibration Processing
```
9.9.21 Temperature Sensor Measurement Processing

Figure 9.23 shows the Temperature Sensor Measurement Processing.

![Temperature Sensor Measurement Processing Diagram]

9.9.22 Current Temperature Calculation

Figure 9.24 shows the Current Temperature Calculation.

![Current Temperature Calculation Diagram]

Note:
1. When AVCC0 is 1.8 V ≤ AVCC0 < 2.7 V:
   \[
   \text{CAL125} = \frac{1.8}{V_{\text{REFH0}}} \times TSCDR(0)
   \]

2. When AVCC0 is 2.7 V ≤ AVCC0 ≤ 3.6 V:
   \[
   \text{CAL125} = \frac{2.7}{V_{\text{REFH0}}} \times TSCDR(1) + \left\{ \frac{3.3}{V_{\text{REFH0}}} \times TSCDR(3) - \frac{2.7}{V_{\text{REFH0}}} \times TSCDR(1) \right\} \times \frac{(AVCC0 - 2.7)}{0.6}
   \]
9.9.23 A/D Conversion End Interrupt Handler

Figure 9.25 shows the A/D Conversion End Interrupt Handler.

```
Excep_AD_ADI

Is the A/D conversion status wait for completion of A/D conversion?
Yes (ad_status == STA_AD_WAIT)

Obtaining temperature sensor measurement result temps_get_potential()

Is the obtained A/D conversion value the minimum value?
Yes ad_min_value ← A/D conversion value

Update the minimum value

Is the obtained A/D conversion value the maximum value?
Yes ad_max_value ← A/D conversion value

Update the maximum value

Store the A/D conversion value in the buffer

Number of times for A/D conversion value acquisition > CNV_CNT_MAX?
Yes

Count up the number of times for A/D conversion value acquisition

Disable the temperature sensor
TSCR register TSEN bit ← 0: Disables the temperature sensor.

Calculate the sum of A/D conversion values for the number of times indicated by CNV_CNT_MAX

Subtract the maximum and minimum values from the sum

Calculate the average with the subtraction result

Store the average in the RAM now_potential ← average

Current temperature calculation temps_calc()

Set the A/D conversion status to ‘A/D conversion completed’
ad_status ← STA_AD_FINISH

Initialize the maximum and minimum values

Initialize the number of A/D conversions

Temperature sensor stop processing temps_close()

return
```

Figure 9.25 A/D Conversion End Interrupt Handler
10. Appendices (Calibration and Compensation Results)

This chapter analyzes the results of the system gain calibration and the temperature characteristic compensation.

10.1 Result of the System Gain Calibration

Figure 10.1 shows an example of the result for the system gain calibration. In the example, the gain is calibrated for each channel with each gain setting based on the gain with channel 0 and gain x4 using formulas 8.3 and 8.5 (for differential input, formula 8.11 for single-ended input). In the result, the gain errors have been reduced from 6 ppm to 2 ppm.

To make the gain measurement conditions consistent, in this example, 14.06 mV of voltage is input taking into account the limit of gain x32 (14.4 mV). To raise the precision of the calibration, use the test voltage and current appropriate to the reference gain selected.

![Figure 10.1 Result of the System Gain Calibration](image)

Measurement conditions:
- Number of samples: 5
- Channels: 0 to 3
- Gains: x1 to x32,
- Input: AC 50.664 Hz ± 14.06 mV (peak-to-peak: 28.12 mV) for all gains
- External input resistor: 0 Ω
10.2 Result of Temperature Compensations

10.2.1 Temperature Characteristics of the VBGR

The Figure 10.2 shows the Temperature Characteristics of the VBGR (Difference Between the Measured Values and Typical Values).

The typical VBGR voltage can be calculated by assigning the coefficients shown in Table 7.1 and the temperature measured by the temperature sensor to formula 7.2. If errors exist in temperatures, calculations for the typical VBGR also have errors.

<table>
<thead>
<tr>
<th>VBGR / VBGR at 25 degC</th>
<th>Raw data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9985</td>
<td>0.9990</td>
</tr>
<tr>
<td>0.9995</td>
<td>1.0000</td>
</tr>
<tr>
<td>1.0005</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Temperature (degC)</th>
<th>Compensation error caused by the measurement error of temperature sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>-50</td>
<td>0.9995</td>
</tr>
<tr>
<td>-25</td>
<td>0.9990</td>
</tr>
<tr>
<td>0</td>
<td>0.9995</td>
</tr>
<tr>
<td>25</td>
<td>1.0000</td>
</tr>
<tr>
<td>50</td>
<td>0.9995</td>
</tr>
<tr>
<td>75</td>
<td>1.0000</td>
</tr>
<tr>
<td>100</td>
<td>0.9995</td>
</tr>
</tbody>
</table>

Degree Celsius: Degrees Celsius

Figure 10.2 Temperature Characteristics of the VBGR (Difference Between the Measured Values and Typical Values)

The temperature characteristics of the VBGR can be decreased from 30 ppm/°C to 10 ppm/°C by compensating with formula 7.2.

Table 10.1 Results of the VBGR Compensation

<table>
<thead>
<tr>
<th>Reference Voltage Temperature Coefficient</th>
<th>-40 to +105 °C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical characteristics in the User’s Manual: Hardware</td>
<td>±30 ppm/°C</td>
</tr>
<tr>
<td>Maximum value of the raw data</td>
<td>+30 ppm/°C (-40°C to +25°C)</td>
</tr>
<tr>
<td></td>
<td>-24 ppm/°C (+25°C to +105°C)</td>
</tr>
<tr>
<td>Residual error after compensation</td>
<td>±10 ppm/°C</td>
</tr>
</tbody>
</table>
10.2.2 System Gain of the Differential Input Pins

Figure 10.3 shows the System Gain of the Differential Input Pins.

The temperature characteristics are compensated to appear around 1.000 whereas they appear as parabola before compensation.

<table>
<thead>
<tr>
<th>Gain Test voltage</th>
<th>Raw data</th>
<th>The ratio of Raw data / Compensated</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1 900 mV</td>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
</tr>
<tr>
<td>x2 450 mV</td>
<td><img src="image3" alt="Graph" /></td>
<td><img src="image4" alt="Graph" /></td>
</tr>
<tr>
<td>x4 225 mV</td>
<td><img src="image5" alt="Graph" /></td>
<td><img src="image6" alt="Graph" /></td>
</tr>
<tr>
<td>x8 112.5 mV</td>
<td><img src="image7" alt="Graph" /></td>
<td><img src="image8" alt="Graph" /></td>
</tr>
<tr>
<td>x16 56.25 mV</td>
<td><img src="image9" alt="Graph" /></td>
<td><img src="image10" alt="Graph" /></td>
</tr>
<tr>
<td>x32 28.12 mV</td>
<td><img src="image11" alt="Graph" /></td>
<td><img src="image12" alt="Graph" /></td>
</tr>
</tbody>
</table>

degC: Degrees Celsius

Measurement condition:
- Test input: AC 50.66 Hz

Figure 10.3 System Gain of the Differential Input Pins
Compensating the temperature characteristics of the system gains on the differential input pins can reduce differences in the temperature characteristics among devices and then the temperature characteristics appear as flat.

Table 10.2 lists the Results of the Compensation for Temperature Characteristics of the System Gain on the Differential Input Pins.

Table 10.2  Results of the Compensation for Temperature Characteristics of the System Gain on the Differential Input Pins

<table>
<thead>
<tr>
<th>Gain Setting</th>
<th>Temperature Compensation Coefficient [ppm/K]</th>
<th>Raw data</th>
<th>Data after compensation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Every 25 K (1)</td>
<td>-25°C to +75°C (2)</td>
</tr>
<tr>
<td>x1</td>
<td>-38</td>
<td>16</td>
<td>-24</td>
</tr>
<tr>
<td></td>
<td>+21</td>
<td></td>
<td>+25</td>
</tr>
<tr>
<td>x2</td>
<td>-39</td>
<td>14</td>
<td>-17</td>
</tr>
<tr>
<td></td>
<td>+17</td>
<td></td>
<td>+23</td>
</tr>
<tr>
<td>x4</td>
<td>-31</td>
<td>15</td>
<td>-13</td>
</tr>
<tr>
<td></td>
<td>+21</td>
<td></td>
<td>+24</td>
</tr>
<tr>
<td>x8</td>
<td>-48</td>
<td>18</td>
<td>-21</td>
</tr>
<tr>
<td></td>
<td>+29</td>
<td></td>
<td>+30</td>
</tr>
<tr>
<td>x16</td>
<td>-96</td>
<td>33</td>
<td>-57</td>
</tr>
<tr>
<td></td>
<td>+45</td>
<td></td>
<td>+64</td>
</tr>
<tr>
<td>x32</td>
<td>-136</td>
<td>41</td>
<td>-97</td>
</tr>
<tr>
<td></td>
<td>+94</td>
<td></td>
<td>+111</td>
</tr>
</tbody>
</table>

Notes:
1. The range between -25°C and +75°C is divided every 25 K, the temperature characteristic coefficients are calculated for all divided ranges, and the minimum and maximum values are picked up and shown in the table.
2. Value calculated with the box method.
   Temperature compensation coefficient = Gain range (maximum value - minimum value) / Temperature range (75 - (-25))
10.2.3 System Gain of Single-Ended Input Pin

Figure 10.4 shows the System Gain of the Single-Ended Input Pins. System gains are inversely proportional to temperatures in the temperature characteristics before compensation. After compensation, system gains are compensated to appear around 1.000.

<table>
<thead>
<tr>
<th>External load resistor</th>
<th>Raw Data Gain x1</th>
<th>Raw Data / Compensated gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8 kΩ</td>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
</tr>
<tr>
<td>5.4 kΩ</td>
<td><img src="image3" alt="Graph" /></td>
<td><img src="image4" alt="Graph" /></td>
</tr>
</tbody>
</table>

degC: Degrees Celsius

Figures 10.4 System Gain of the Single-Ended Input Pins
Compensating the temperature characteristics of the system gains on the single-ended input pins can reduce differences in temperature characteristics among devices and then the temperature characteristics appear as flat. Table 10.3 lists the Results of the Compensation for Temperature Characteristics of the System Gain on the Single-Ended Input Pins.

Table 10.3 Results of the Compensation for Temperature Characteristics of the System Gain on the Single-Ended Input Pins

<table>
<thead>
<tr>
<th>External Load Resistor [kΩ]</th>
<th>Temperature Characteristic Coefficient [ppm/K]</th>
<th>Raw data</th>
<th>Data after compensation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Every 25 K (1)</td>
<td>-25°C to +75°C (2)</td>
<td>Every 25 K (1)</td>
</tr>
<tr>
<td>1.8</td>
<td>-186</td>
<td>54</td>
<td>-145</td>
</tr>
<tr>
<td></td>
<td>+114</td>
<td></td>
<td>+167</td>
</tr>
<tr>
<td>5.4</td>
<td>-249</td>
<td>90</td>
<td>-136</td>
</tr>
<tr>
<td></td>
<td>+104</td>
<td></td>
<td>+176</td>
</tr>
</tbody>
</table>

Notes:
1. The range between -25°C and +75°C is divided every 25 K, the temperature characteristic coefficients are calculated for all divided ranges, and the minimum and maximum values are picked up and shown in the table.
2. Value calculated with the box method.
   
   Temperature compensation coefficient = Gain range (maximum value - minimum value) / Temperature range (75 - (-25))
11. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

12. Reference Documents

User’s Manual: Hardware
RX21A Group User’s Manual: Hardware Rev.1.10 (R01UH0251EJ)
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

User’s Manual: Development Tools
RX Family C/C++ Compiler Package V.1.01 User’s Manual Rev.1.00 (R20UT0570EJ)
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

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

Inquiries
http://www.renesas.com/contact/
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Oct. 1, 2014</td>
<td>—</td>
<td>First edition issued</td>
</tr>
<tr>
<td>1.10</td>
<td>Mar. 2, 2015</td>
<td>—</td>
<td>Revised the structure and contents of the document.</td>
</tr>
</tbody>
</table>

All trademarks and registered trademarks are the property of their respective owners.
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 accordance 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 an 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 a product with a different part number, confirm that the change will not lead to problems.
   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.
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 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": Consumer electronics; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronics; appliance, machine tools; personal electronic equipment; and industrial robots etc.

   "High Quality": Transportation equipment; 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 implementations 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. You are responsible for ensuring that products which incorporate any Renesas Electronics products achieve the quality, durability, reliability, safety and other characteristics necessary for the use to which they are put.

8. 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 and follow the procedures required by such laws and regulations. You should evaluate the safety of the final products or systems manufactured by you.

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. If you are 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.