RA Family

RA2A1 Weight Measurement Example Using a Load Cell

Summary

Microcontrollers of the RA2A1 group include a 24-bit sigma-delta A/D converter (SDADC24) that has a programmable gain instrumentation amplifier (PGA). This document describes an example of weight measurement by using the SDADC24 to A/D convert the output from a load cell with strain gauges.

Contents

1. Overview

This document describes an example of weight measurement by the RA2A1 with the use of a load cell. This example uses a 4-wire load cell implemented as a Wheatstone bridge. A sample program runs on the board of the evaluation kit for RA2A1 microcontrollers (EK-RA2A1), and the results of measurement can be displayed with terminal software such as Teraterm.

Figure 1-1 shows the weight measurement system in this example.

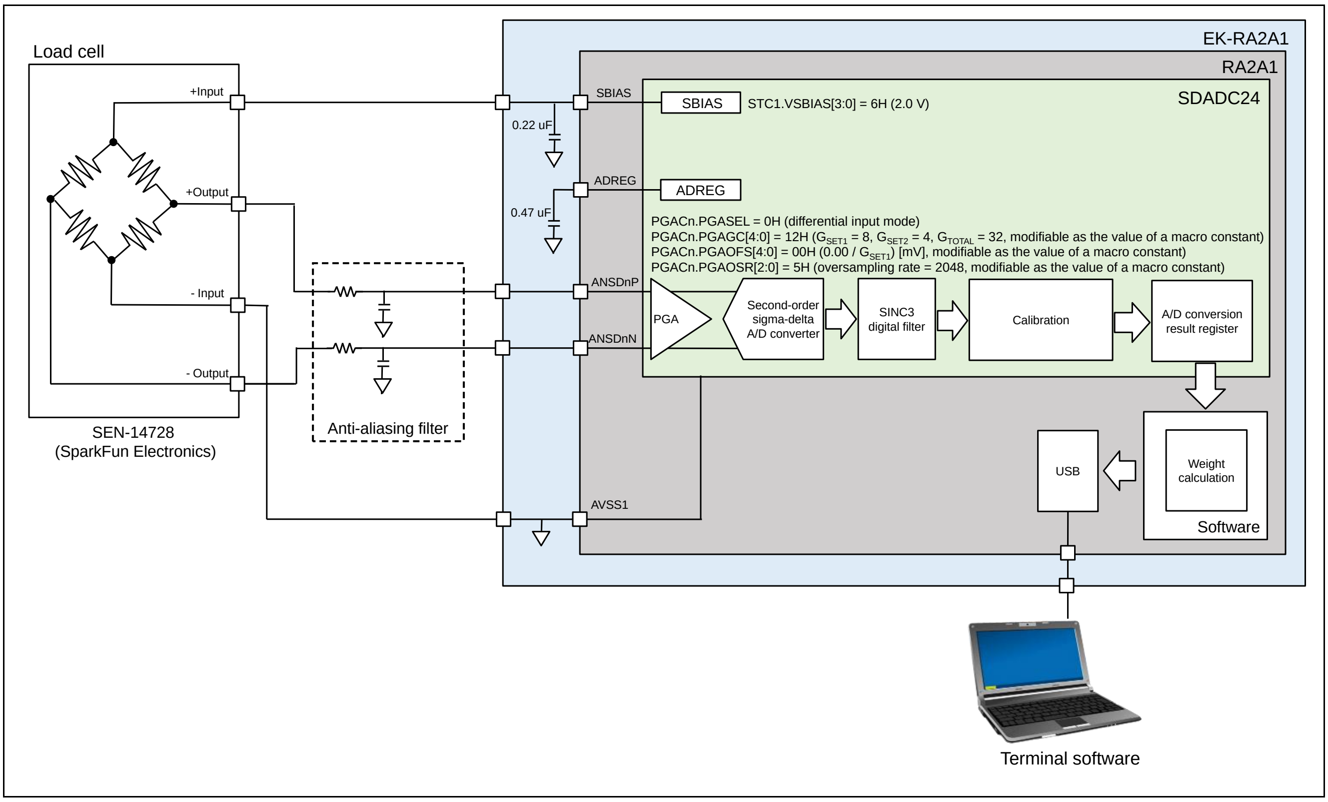

Figure 1-1 Example of Weight Measurement System Using a Load Cell

2. Environment for Confirming Operation

Details of the environment for confirming operation are given in Table 2-1.

Table 2-1 Environment for Confirming operation

Item	**Description**
Board	EK-RA2A1 board
MCU	RA2A1 (R7FA2A1AB3CFM) Power-supply voltage (VCC, AVCC0, AVCC1): 3.3 V System clock (ICLK): 48 MHz Peripheral module clock B (PCLKB): 24 MHz Peripheral module clock D (PCLKD): 24 MHz USB clock (UCLK): 48 MHz SDADC24 operating clock (SDADCCLK): 4 MHz SDADC24 oversampling frequency (Fos): 1 MHz
Load cell	SEN-14728 from SparkFun Electronics
IDE	e^2 studio V7.8.0 from Renesas Electronics Flexible Software Package (FSP) V1.1.0
Tool Chain	GNU Tools for ARM Embedded Processors 9-2-19-q4-major (9.2.1.20191025)
Emulator	J-Link

3. Related Application Note

Also refer to the following document, which is related to this application note.

Application Note for 24-Bit Sigma-Delta A/D Converter Performance for RA2A1 (R01AN5286)

4. Weight Measurement Method

An example of connections with the EK-RA2A1 board is shown in Figure 4-1. In this example, channel 0 (ANSD0P and ANSD0N) of the SDADC24 is used for weight measurement.

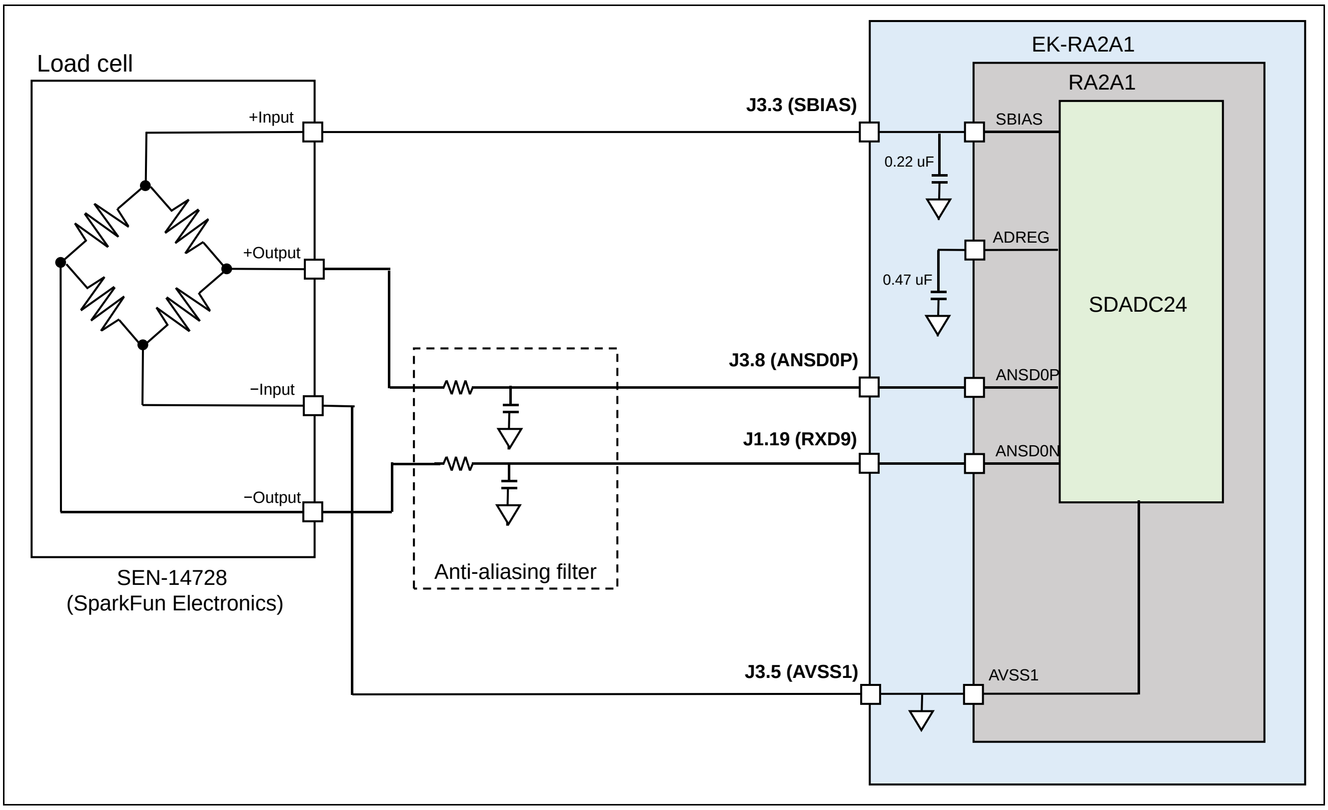

Figure 4-1 Example of Connections with the EK-RA2A1 Board

Table 4-1 is a list of the pins to be used. Channel 0 is used for weight measurement in this example, but another channel can be selected for use by modifying the macro constant for specifying the channel, which is shown in Table 5-9. In accordance with the selected channel, connect the load cell pins to the correct pin headers on the EK-RA2A1 board as listed in the table below.

Table 4-1 Pins to be Used

<table>
<tr><th>Load Cell SEN-14728
Pin Name</th><th>EK-RA2A1
Connector Number, Indication on Board</th><th>RA2A1
Pin Name</th></tr>
<tr><td>+Input: Excitation (+)</td><td>J3.3, SBIAS</td><td>SBIAS/VREF</td></tr>
<tr><td>-Input: Excitation (-)</td><td>J3.5, AVSS1</td><td>AVSS1</td></tr>
<tr><td rowspan="8">+Output: Signal (+)
-Output: Signal (-)</td><td colspan="2">Channel 0</td></tr>
<tr><td>J3.8, ANSD0P
J1.19, RXD9</td><td>ANSD0P (P100)
ANSD0N (P101)</td></tr>
<tr><td colspan="2">Channel 1</td></tr>
<tr><td>J3.6, ANSD1P
J1.5, RSPCKB</td><td>ANSD1P (P102)
ANSD1N (P103)</td></tr>
<tr><td colspan="2">Channel 2</td></tr>
<tr><td>J1.9, MISOB
J1.7, MOSIB</td><td>ANSD2P (P104)
ANSD2N (P105)</td></tr>
<tr><td colspan="2">Channel 3</td></tr>
<tr><td>J2.30, AN022
J2.28, AN023</td><td>ANSD3P (P106)
ANSD3N (P107)</td></tr>
</table>

4.1 Load Cell

The load cell used in this example has a Wheatstone bridge circuit and outputs a voltage that indicates applied weight. Table 4-2 consists of excerpts from the specifications of the load cell.

Table 4-2 Excerpts from the SEN-14728 Specifications

Item	Symbol	Value
Excitation voltage (applied voltage)	VE	≤ 6 V
Rated capacity	L	500 g
Rated output	RO	0.7 ± 0.15 mV/V
Zero balance	ZB	0.1 %
Input impedance (impedance between the input pins)	ZI	1090 ± 10 Ω
Output impedance (impedance between the output pins)	ZO	1000 ± 10 Ω
Combined error	εC	0.05 %FS
Temperature coefficient of ZERO (temperature effect on zero point)	εZ	0.01 %FS/°C
Temperature coefficient of SPAN (temperature effect on output)	εS	0.01 %FS/°C

Table 4-2 consists of excerpts from the data sheet of the load cell. For details and latest values of the specifications, refer to the data sheet.

4.2 Accuracy of Measurement by the Load Cell

The accuracy ε of measurement by the load cell can be obtained from the following equation, where N is the number of load cells to be used, W is the maximum load (weight) to be measured, and t is the range of temperature change across the load cell.

$$\varepsilon = \sqrt{\varepsilon C^2 + \left(\frac{\varepsilon Z * L * N}{W} * t\right)^2 + (\varepsilon S * t)^2}$$

By assigning the values from the SEN-14728 specifications used in this example to the above equation, we obtain the following equation when the number N of load cells is 1, the maximum load to be measured is 500 g, and the temperature change is ±10°C (the range of change t is 20°C).

$$\varepsilon = \sqrt{0.05^2 + \left(\frac{0.01 * 500 * 1}{500} * 20\right)^2 + (0.01 * 20)^2} \fallingdotseq 0.287(\%)$$

This means that the load cell used in this example can measure weight with an accuracy of approx. 0.287% of the full-scale weight (500 g) — that is, an accuracy of ± 1.435 g, under the temperature range of ±10°C. In estimating the total accuracy of a system, the accuracy of the measuring instruments must also be considered.

4.3 Connecting the Load Cell

Use the sensor power-supply pin SBIAS of the RA2A1 to supply the driving power for the load cell. As SBIAS is also used as the reference voltage for the SDADC24, supplying the SBIAS power to the load cell cancels out fluctuations in the power-supply voltage, improving the accuracy of measurement. The SBIAS pin can output up to 10 mA and the impedance between the input pins of the load cell is 1090 Ω, so the voltage to be supplied should be no greater than 10 mA * 1090 Ω = 10.9 V. In this example, 2.0 V is output from SBIAS with taking the input voltage range (described later) into account. The output current is 2.0 V / 1090 Ω ≈ 1.835 mA.

The voltage output from the load cell can be calculated from the voltage VE applied to the load cell and the rated output RO. When a load of 500 g (rated capacity) is applied to the load cell, the output voltage is VE * RO = 2.0 V * 0.7 mV/V = 1.4 mV. This means that applying a 0.1-g load to the load cell produces an output voltage of 1.4 mV/(500 g/0.1 g) = 0.28 uV.

As the load cell outputs a differential signal, the PGA should be set to the differential input mode. The range of voltage that can be applied to the PGA input pins is 0.2 V to 1.8 V and the range of differential input voltage is ±0.8 V/PGA multiplication factor; the input signals should be within these ranges. When a voltage of 2.0 V is being applied to the load cell, the midpoint of the voltage range of the differential signal to be output is 1.0 V and measurement is possible over a widest input voltage range of 1.0 V ± 0.8 V.

The weight vs. output voltage characteristic and the error range determined from the specifications when the applied voltage is 2 V are shown in Figure 4-2.

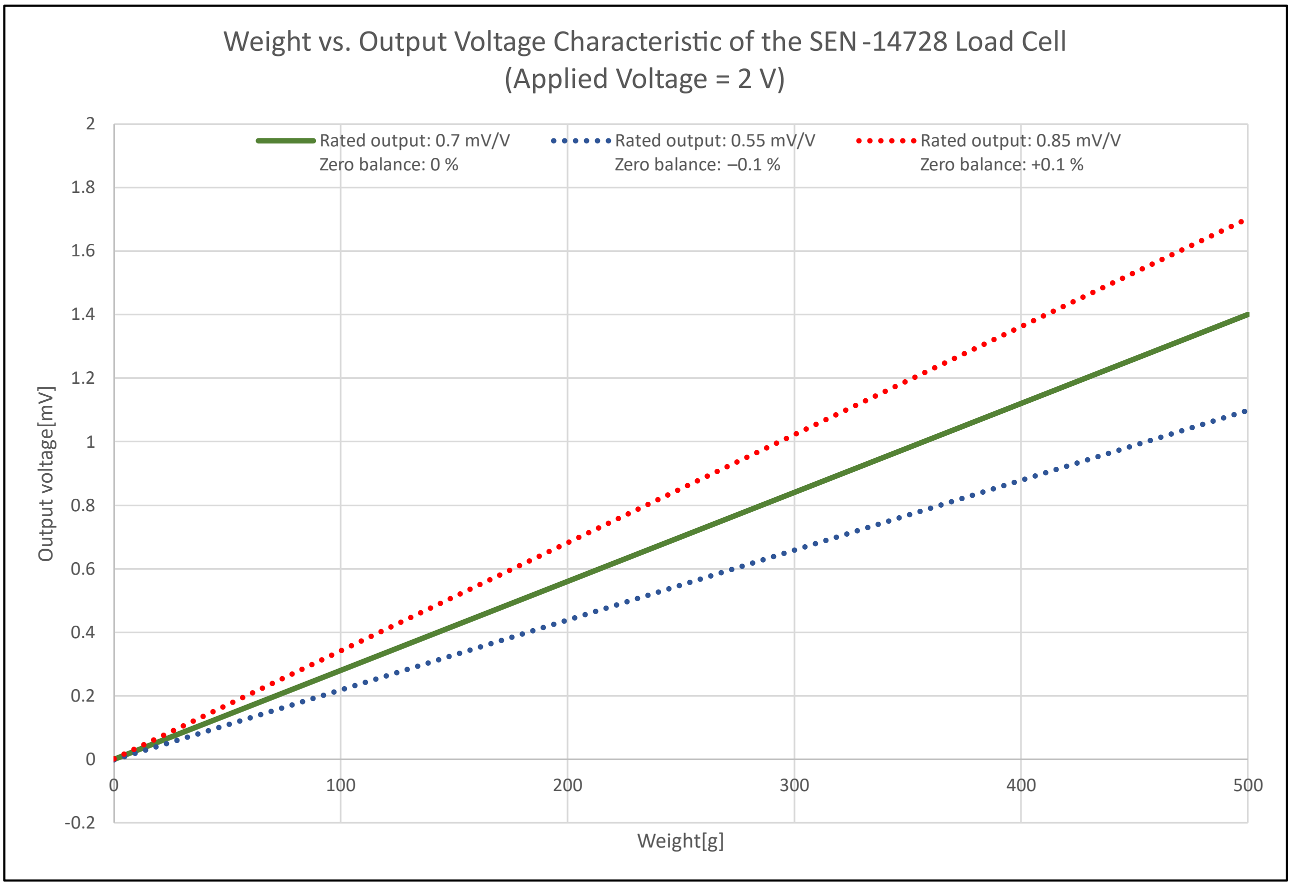

Figure 4-2 Weight vs. Output Voltage Characteristic of the SEN-14728 Load Cell (Applied Voltage = 2 V)

4.4 Calculating Voltage from the A/D Converted Value

The A/D converted value can be converted to a voltage by the following equation, where Vin is the voltage input to the SDADC24, G_{TOTAL} is the multiplication factor of the PGA, and ADCDATA is the result of A/D conversion. Here, the voltage input to the SDADC24 is the voltage difference obtained by subtracting the "negative-channel" voltage from the "positive-channel" voltage. As the result of A/D conversion in differential input mode is a two's complement, ADCDATA is between +0.8 V/G_{TOTAL} and –0.8 V/G_{TOTAL} — that is, the result is between $2^{23} - 1$ and -2^{23}.

$$Vin = \left(\frac{1.6}{Gtotal * 2^{24}}\right) * ADCDATA$$

When a 500-g load is applied, the output of the load cell is a differential signal having a voltage difference of 1.4 mV. In this case, when the multiplication factor of the PGA is 1 (G_{TOTAL} = 1), the A/D converted value is 14680.

To measure the rated capacity (500 g) of the load cell used in this example over the widest allowable range of A/D conversion, obtain the highest multiplication factor of the PGA that satisfies the above A/D conversion result range ($2^{23} - 1$ to -2^{23}) of the SDADC24. Even when the PGA is set to its maximum factor of ×32 (G_{TOTAL} = 32), the result of A/D conversion is 469762, which is within the range of $2^{23} - 1$ to -2^{23}. Therefore, the PGA is set to ×32 in this example.

4.5 Settings of SDADC24

The following describes the settings of the SDADC24 in this example.

4.5.1 Gain of PGA

The result of A/D conversion in the SDADC24 is between $2^{23} - 1$ and -2^{23}. To measure the rated capacity (500 g) of the load cell used in this example over the widest allowable range of A/D conversion, obtain the highest multiplication factor of the PGA that satisfies the above range.

Obtain the A/D converted value for the rated capacity (500 g) of the load cell from the equation shown in section 4.4, Calculating Voltage from the A/D Converted Value. When a 500-g load is applied, the output of the load cell is a differential signal having a voltage difference of 1.4 mV. In this case, when the multiplication factor of the PGA is 1, the A/D converted value is 14680. Even when the PGA is set to its maximum factor of ×32, the A/D conversion result is 469762, which is within the range of $2^{23} - 1$ to -2^{23}. Therefore, the PGA is set to ×32 in this example.

4.5.2 Offset Voltage

The voltage input to the SDADC24 may exceed the allowable input range due to the offset errors of the load cell and PGA. To prevent this, the offset voltage may require adjustment in some cases.

As the PGA is set to ×32 in this example, the voltage range of the differential input signal becomes ±0.8 V/32 = ±25 mV. As shown in Figure 4-2, Weight vs. Output Voltage Characteristic of the SEN-14728 Load Cell (Applied Voltage = 2 V), the input voltage never exceeds ±25 mV even if it is affected by the rated output or the zero balance error of the load cell. In addition, the offset error of the PGA in the SDADC24 is ±1 mV. Even if this error is added, the input voltage never exceeds ±25 mV. Therefore, the offset voltage is set to 0 mV in this example.

4.5.3 Oversampling Ratio

Oversampling is a technique for sampling the input at a higher frequency than the required signal bandwidth. In ordinary sampling, the sampling frequency is twice the signal bandwidth. Sampling the input at a frequency multiplied by K in comparison with that for ordinary sampling reduces the quantization noise density (noise per 1 Hz) to $1/\sqrt{K}$. The total amount of quantization noise is equivalent to that in ordinary sampling but the quantization noise is dispersed up to the frequency multiplied by K. Therefore, by using a digital filter to eliminate the noise outside the signal bandwidth, a reduction in noise can be expected in accord with the multiplication factor K of the oversampling frequency, improving the accuracy of measurement. This means that a larger improvement in the accuracy can be expected with a higher oversampling ratio.

In weight measurement in this example, however, the load cell outputs a DC signal, so high-speed conversion is not required. Therefore, the oversampling ratio for the SDADC24 is set to its maximum value (2048).

4.5.4 A/D Conversion Mode

There are two A/D conversion modes: normal A/D conversion mode and low-power A/D conversion mode.

In normal A/D conversion mode, the SDADC24 reference clock runs at 4 MHz and the oversampling clock runs at 1 MHz. In low-power A/D conversion mode, the SDADC24 reference clock runs at 500 kHz and the oversampling clock runs at 125 kHz — that is, the frequencies in low-power mode are one-eighth of those in normal mode. While power consumption is reduced in low-power mode, the output data rate also becomes lower than in normal mode. As there is also a trade-off between the power consumption and the accuracy of measurement, the selected A/D conversion mode (and frequencies) should be appropriate for the target system.

In this example, normal A/D conversion mode is selected to obtain the maximum available accuracy. As the oversampling ratio is 2048 and the oversampling clock runs at 1 MHz, the output data rate is 1 MHz/2048 ≈ 488.28 sps.

4.6 Averaging A/D Converted Values

Averaging the results of consecutive A/D conversions reduces the effect of transient noise and generally provides a more accurate result. In general, modern weighing scales visually display the results of measurement on LCDs, etc., so the sampling rate used for a weighing scale must be such that the human eye can observe the changes in the displayed values. In this example, the weight is calculated from the average of the values from 244 rounds of A/D conversion. The output data rate of A/D conversion is 488.28 sps, so the displayed weight will be updated roughly every 0.5 seconds.

Although the SDADC24 has a function for averaging results of A/D conversion, it is not used in this example because the limit on the maximum number of results of conversion to be averaged is 64. If a system only requires the averaging of 64 or fewer results of conversion, the averaging function can be used.

4.7 Calculating Weight from the A/D Converted Value

The result of A/D conversion by the SDADC24 includes a value obtained by multiplying the voltage from the offset adjustment DAC by the gain of the second stage of the PGA. Therefore, before calculating a weight from the A/D converted value, subtract the offset adjustment value from the result of conversion. The A/D converted value OFFSET corresponding to the offset adjustment value can be obtained from the following equation, where VOFR is the voltage output from the offset adjustment DAC and G_{SET2} is the gain of the second stage of the PGA.

$$OFFSET = \frac{VOFR * 2^{24} * Gset2}{1.6}$$

The output voltage V for a weight M can be expressed by the following equation, where Vcc is the voltage applied to the load cell, RO is the rated output, and Mmax is the rated capacity.

$$V = RO * Vcc * \frac{M}{Mmax}$$

From the equation above, the weight M for an output voltage V can be calculated with the linear equation below.

$$M = \alpha V + \beta, \qquad \{\alpha = \frac{Mmax}{RO * Vcc}, \beta = 0$$

By replacing the output voltage V in the above equation with the A/D converted value, we obtain the following equation, where G_{TOTAL} is the multiplication factor of the PGA and ADCDATA is the result of A/D conversion.

$$\begin{aligned} M &= \alpha V + \beta \\ &= \alpha * \left(\frac{1.6}{Gtotal * 2^{24}}\right) * (ADCDATA - OFFSET) + \beta \end{aligned}$$

From the above equation, we define the equation for calculating a weight from the A/D converted value as follows.

$$M = \alpha V + \beta = a * (ADCDATA - OFFSET) + b, \qquad \{a = \alpha * \left(\frac{1.6}{Gtotal * 2^{24}}\right), b = \beta = 0$$

Note that the output voltage of the load cell also includes error components due to the rated output, zero balance, and so on. Thus, use calibration to correct the coefficients a and b in the equation. Use A/D converted values that include the gain and offset in calibration (calculation of coefficients a and b) so that the processing to eliminate the gain and offset components from the A/D converted value can be omitted from calculating the weight.

If the output voltage of the load cell is non-linear relative to the weight, the characteristic curve is divided into multiple regions so that linear approximation, for example, can be used in each of the regions to improve the accuracy of measurement, thereby closely approximating the characteristic curve. In this example, however, the output voltage is regarded as a single linear characteristic which does not require such division, and the voltage is converted to weight through linear interpolation.

The following shows a sample procedure for obtaining the coefficient values by calibration from two reference weights such as test weights and their A/D converted values within the range of weight that satisfies the conversion equation.

(1) Obtain the A/D converted value ADCDATA1 of the weight M1 used as reference 1.
(2) Obtain the A/D converted value ADCDATA2 of the weight M2 used as reference 2.
(3) Calculate coefficients a and b for the line that passes through (ADCDATA1, M1) and (ADCDATA2, M2) from the following equations.

$$a = \frac{M2 - M1}{ADCDATA2 - ADCDATA1}$$

$$b = M1 - a * ADCDATA1 = M2 - a * ADCDATA2$$

By applying the obtained coefficients a and b, the weight M can be calculated from the A/D converted value ADCDATA by the following conversion equation.

$$M = a * ADCDATA + b$$

Although this example is based on an assumption of linearity, to further improve the accuracy of measurement, the non-linearity error can be taken into account by dividing the characteristic curve into multiple regions and applying multiple-point calibration for linear approximation in each of the regions. Use the same procedure as that used in the above two-point calibration. For calibration at three points (0 g, 200 g, and 500 g), for example, use the following procedure.

Calculate coefficients a1 and b1 of the line between 0 g and 200 g. Here, the A/D converted value for 200 g is called AD1. Then, calculate coefficients a2 and b2 of the line between 200 g and 500 g. If the A/D converted value for a measured voltage is smaller than AD1, use coefficients a1 and b1 to calculate the weight. If the A/D converted value is greater than or equal to AD1, use coefficients a2 and b2. Applying linear interpolation to each of the divided regions in this way can reduce the non-linearity error. In addition, the use of a greater number of calibration points can further reduce the non-linearity error.

4.8 Zero Reset

The measured weight is corrected by subtracting the reference result of measurement determined as the state of zero weight from the result of weight calculation. The reference value is the result of measurement and calculation of weight in the state where no weight is being applied.

4.9 Accuracy of Weight Measurement

In section 4.2, Accuracy of Measurement by the Load Cell, only the accuracy of the load cell was described. In this section, the accuracy of the SDADC24 is also considered.

As described in the Application Note for 24-Bit Sigma-Delta A/D Converter Performance for RA2A1 (R01AN5286), the effective resolution of the SDADC24 is 16.4 bits in the case of the settings for this sample program (G_{SET1} = 8, G_{SET2} = 4, and OSR = 2048). The effective resolution is greater than the noise-free resolution by $\log_2(6.6) \approx 2.7$ bits — that is, the noise-free resolution is 13.7 bits. These resolutions correspond to voltages (noise converted to the input) and values for weight as follows.

The effective resolution of 16.4 bits corresponds to 0.58 uVRMS and 0.21 g, and the noise-free resolution of 13.7 bits corresponds to 3.76 uV and 1.34 g. This means that measurement can be performed with an accuracy of 0.268 % of the full-scale weight (500 g). This accuracy is for a single A/D conversion. For further improvement in accuracy, the results of A/D conversions performed multiple times should be averaged as described in section 4.6, Averaging A/D Converted Values. As the effect of averaging on the accuracy depends on the individual environment, an appropriate number of values for averaging should be selected in accordance with the required accuracy and response speed in the actual target environment.

Assuming that A/D conversion is only performed once, the load cell accuracy εl is 0.287 %, and the SDADC24 accuracy εad is 0.268 %, the total accuracy ε is obtained as follows.

$$\boldsymbol{\varepsilon} = \sqrt{\boldsymbol{\varepsilon l^2} + \boldsymbol{\varepsilon ad^2}} \risingdotseq \mathbf{0.393}(\%)$$

The accuracy of the entire system is 0.393 % of the full-scale weight (500 g) — that is, the weight can be measured with an accuracy of ±1.965 g.

5. Sample Program

The following describes the specifications of the sample program.

5.1 Overview of Operation

This sample program starts weight measurement when the switch is pressed and sends the result of weight measurement through the USB interface. If the switch is pressed again while the weight is being measured, the program stops measurement and waits for the switch to be pressed again. If an overflow occurs in A/D conversion during weight measurement or if the result of weight measurement exceeds a specified threshold value, the program executes processing for detecting disconnection.

If the result of the above processing by the sample program is the detection of disconnection, the program stops measurement and waits for the switch to be pressed again. If it determines that there is no disconnection, it continues weight measurement.

If an error occurs and the operation cannot be continued, the program stops operation.

Figure 5-1 is a state transition diagram of the sample program.

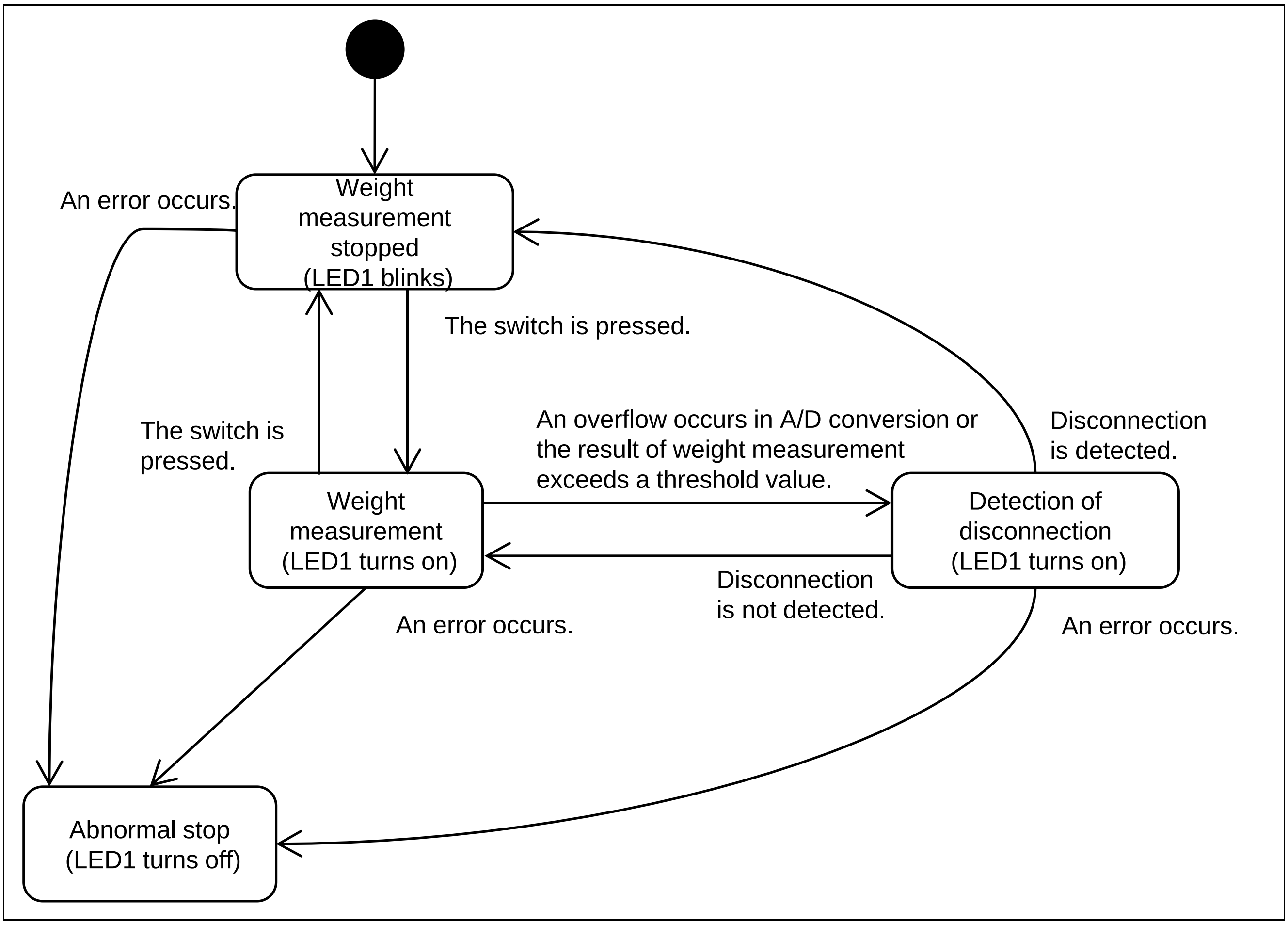

Figure 5-1 State Transition Diagram of the Sample Program

5.1.1 Processing Flowcharts

Figure 5-2 and Figure 5-3 show flowcharts of sample program processing.

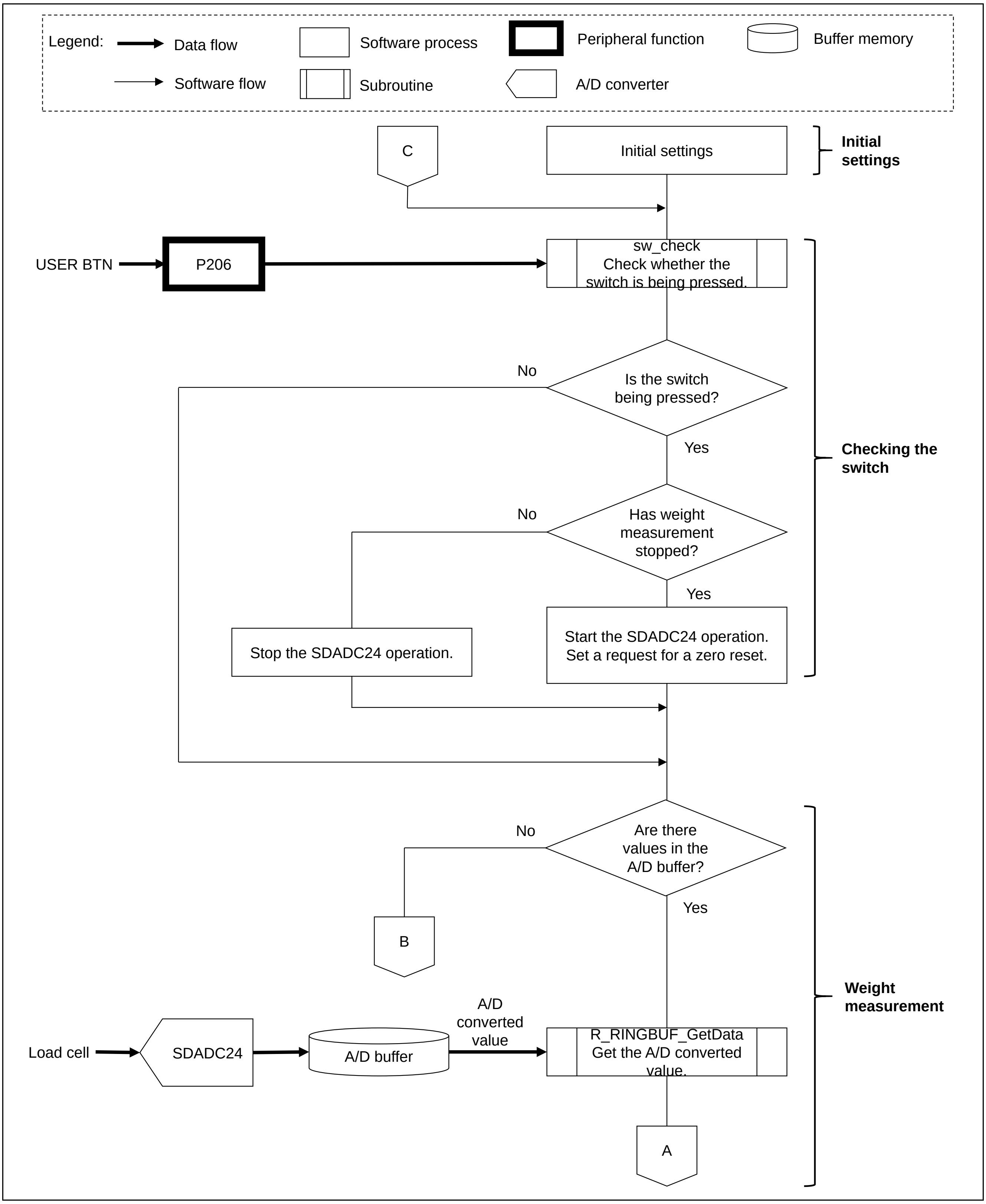

Figure 5-2 Flowchart of Sample Program Processing (1/2)

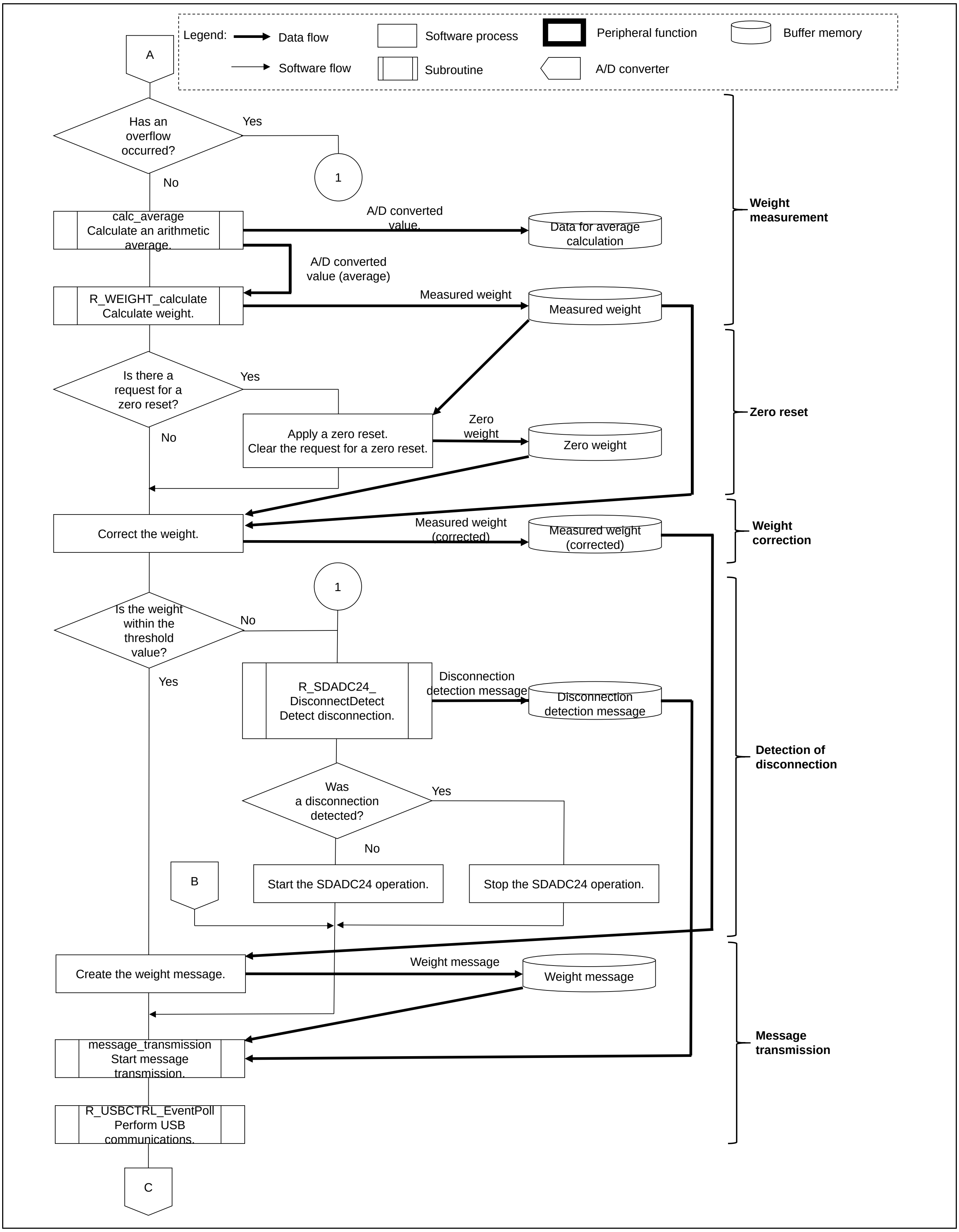

Figure 5-3 Flowchart of Sample Program Processing (2/2)

The following provides an overview of each step in the processing.

- Initial Settings
  The timer to control the blinking of LED1 is started and the initial settings of the USB are made.

- Checking the Switch
  The input from USER BTN (P206) is read. Two consecutive input values that match are considered to represent the state of the switch. Either of the following operations proceeds according to the program state when the pressing of USER BTN is detected.

  — Weight measurement stopped state
    The variables are cleared, the zero adjustment request flag is set, and the SDADC24 operation is started.
    The timer to control the blinking of LED1 is stopped and LED1 is turned on.

  — Weight measurement state
    The SDADC24 operation is stopped and the timer to control the blinking of LED1 is started.

- Weight Measurement
  A/D converted values are read from the A/D converted value ring buffer and the average is calculated from them. After that, the weight is calculated from the average.
  If values cannot be read from the ring buffer or the necessary number of A/D converted values cannot be obtained, the weight is not calculated. As the number of values used in calculating the average is set to 244 in this sample program, the weight is calculated about every 0.5 seconds.
  If an overflow occurs in A/D conversion or the calculated weight exceeds the specified upper or lower limit, the disconnection detection request flag is set.

- Zero Reset
  Before the zero reset processing, the measured weight values are checked for stability to prevent a value measured in an unstable state from being set as the zero weight. In this example, when the difference between the previous and current measured weights is no greater than 30 mg in three consecutive rounds of measurement, the program determines that the measured values are stable.
  When the zero adjustment request flag is set and the measured values are stable, the current measured weight is set as the zero weight.

- Weight Correction
  The weight value set as the zero weight is subtracted from the measured weight value to correct the weight.

- Detection of Disconnection
  If an overflow occurs in A/D conversion or the measured weight exceeds a specified threshold value (overweight threshold), disconnection detection processing proceeds. In this example, the overweight threshold is set to 125 % of the rated capacity of the load cell.
  In this processing, SDADC24 operation is temporarily stopped, settings are switched those for detecting disconnection, and disconnection detection then proceeds with the use of the disconnection detection assistance function. If disconnection is not detected, the SDADC24 is restored to the settings for weight measurement, input scanning by the SDADC24 is restarted, and weight measurement is resumed. If disconnection is detected, weight measurement does not proceed and the program enters into the weight measurement stopped state (the state of waiting for the switch to be pressed).

- Message Transmission
  Messages are sent through the USB (CDC) interface. For the specifications of messages, refer to section 0,

Specifications of Messages.

5.1.2 Specifications of Messages

This section describes the messages to be sent by this sample program. Four types of messages are sent: ZeroAdjust messages (results of zero adjustment processing), Weight messages (results of weight measurement), Overflow messages (occurrence of overflows), and DisconnectDetect messages (results of disconnection detection processing). Each type of message is described below and on the following pages.

5.1.2.1 ZeroAdjust Messages

ZeroAdjust messages provide notification of the results of zero adjustment processing. There are two types of message: one is sent when the measured weight values are not stable and the current value is not used as the zero adjustment value; the other is sent when the measured weight values have become stable and the current value is used as the zero adjustment value.

- Message when the measured weight values are not stable and the current value is not used as the zero adjustment value

  [ZeroAdjust], The measured value is not stable.

- Message when the measured weight values have become stable and the current value is used as the zero adjustment value

  [ZeroAdjust], Zero adjust value is 0.2[g].

5.1.2.2 Weight Messages

Weight messages provide notification of the results of weight measurement. A Weight message consists of the header "[Weight]," followed by multiple parameters separated by commas. The following shows the format of a Weight message.

[Weight], param1, param2, param3, param4, param5, param6, param7, param8

Each parameter is described in Table 5-1.

Table 5-1 Parameters of the Weight Message

Parameter	Description
param1	Measured weight value (g)
param2	Zero adjustment weight (g)
param3	Averaged A/D converted value (including the gain and offset components)
param4	Averaged A/D converted value (with the gain and offset components eliminated)
param5	Offset value (converted to A/D value)*
param6	Gain of the first stage of the PGA (G_{SET1})
param7	Gain of the second stage of the PGA (G_{SET2})
param8	Total gain of the PGA ($G_{TOTAL} = G_{SET1} * G_{SET2}$)

Note: * The offset voltage (mV) specified in the User's Manual is converted to an A/D value without taking the offset error into account. For example, when G_{SET1} = 8, G_{SET2} = 4, and offset voltage = 10.9375/G_{SET1} mV, the offset value included in the A/D value is 458752.
Voltage per 1 LSB = (1.6V / 32) * (1 / 2^24) ≈ 2.98 nV
(10.9375 mV / 8) / 2.98 nV = 458752

The following shows an example of a Weight message.

- Example of a Weight message
  [Weight], 99.9, 0.2, 152165.078125, 4755.158691, 0, 8, 4, 32

  This message means the following.
  Measured weight: 99.9 g
  Zero adjustment weight: 0.2 g
  Averaged A/D converted value (including the gain and offset): 152165.078125
  Averaged A/D converted value (with the gain and offset eliminated): 4755.158691
  Offset value (converted to A/D value): 0
  Gain of the first stage in the PGA (G_{SET1}): 8
  Gain of the second stage in the PGA (G_{SET2}): 4
  Total gain of PGA: 32

5.1.2.3 Overflow Messages

Overflow messages provide notification of an overflow having occurred in A/D conversion or the measured weight value having exceeded the specified threshold value.

- Message when an overflow has occurred in A/D conversion
  [Overflow], A/D conversion value overflow.
- Message when the measured weight value has exceeded a specified threshold value
  [Overflow], Weight overload.

5.1.2.4 DisconnectDetect Messages

DisconnectDetect messages provide notification of the results of disconnection detection processing. There are two types of message: one is sent when a disconnection was not detected and the other is sent when a disconnection was detected.

- Message when a disconnection was not detected
  [DisconnectDetect], Connected – continue weighing process.
- Message when a disconnection was detected
  [DisconnectDetect], Disconnected – stop weighing process.

5.2 Peripheral Functions and Pins Used

The peripheral functions used in this example are listed in Table 5-2, and the pins used are listed in Table 5-3. The settings of the SDADC24 are also described in this section. The SDADC24 settings are generated by using the facilities of the RA Smart Configurator.

Table 5-2 Peripheral Functions Used

Peripheral Function	Purpose of Use
SDADC24	A/D conversion of the output from the load cell
GPT	Generation of the timing for blinking LED1 (500 ms)
USBFS	Transmission of messages (CDC)
P205	Control of LED1 (turn on and off)
P206	Input from USER BTN

Table 5-3 Pins Used

Pin Name	Input/Output	Purpose of Use
SBIAS	Output	Power supply to the load cell
P100/ANSD0N	Input	Input of – side differential signal from the load cell
P101/ANSD0P	Input	Input of + side differential signal from the load cell
P205	Output	Control of LED1 (turn on and off)
P206	Input	Input from USER BTN
P407/USB_VBUS	Input	Input of VBUS
P914/USB_DP	Input/output	USB data (+ side differential signal)
P915/USB_DM	Input/output	USB data (– side differential signal)

5.2.1 Settings of SDADC24

The SDADC24 settings for weight measurement are shown in Table 5-4.

Table 5-4 Settings of SDADC24

Item	Setting
Reference clock frequency division	SDADCCLK/12 SDSDCCLK = HOCO = 48 MHz and reference clock = 4 MHz
A/D conversion mode	Normal A/D conversion mode SDADC24 reference clock: 4 MHz Oversampling clock: 1 MHz
Reference voltage	2.0 V
VREF mode	Internal VREF mode
Gain of the programmable gain instrumentation amplifier	Total gain of the PGA: ×32 Gain of the first-stage amplifier: ×8 Gain of the second-stage amplifier: ×4
Oversampling ratio	2048
Offset voltage	0.00 / G_{SET1} (mV)
Analog channel input mode	Differential input mode
Number of A/D conversions in auto scan	One time
Averaging processing	Results of A/D conversion are not averaged.
Auto scan mode	Continuous scan mode
A/D conversion trigger signal	Software trigger

5.3 Configuration of Program

The following describes the configuration of this sample program.

5.3.1 Configuration of Files

The configuration of files is shown in Table 5-5.

Table 5-5 Configuration of Files

Folder Name and File Name	Description
ek_ra2a1_loadcell	—
├ ra	Generated by RA Smart Configurator
├ ra_cfg	Generated by RA Smart Configurator
├ ra_gen	Generated by RA Smart Configurator
└ src	—
├ board_cfg.h	Definition of the information regarding LED1 on EK-RA2A1
├ hal_entry.c	Main processing
├ hal_entry.h	Definitions of the constants used in the main processing function
├ r_interrupt_callback.c	Interrupt callback processing
├ r_interrupt_callback.h	Definitions related to the interrupt callback functions
├ r_ring_buffer_control_api.c	Ring buffer control processing
├ r_ring_buffer_control_api.h	Definitions of ring buffer control API
├ r_sdadc24_control_api.c	SDADC24 control processing
├ r_sdadc24_control_api.h	Definitions of SDADC24 control API
├ r_usb_control_api.c	USB control processing
├ r_usb_control_api.h	Definitions of USB control API
├ r_usb_pcdc_descriptor.c	Definitions of USB descriptors
├ r_weight_calculate_api.c	Weight calculation processing
└ r_weight_calculate_api.h	Definitions of weight calculation API

5.3.2 Enumeration Types

The following describes the enumeration types defined in this sample program.

Table 5-6 Enumeration Types in r_sdadc24_control_api.h (1/2)

Type Name	**e_pga_sdadc24_gain_t**		
Member	Name	Value	Description (Gain Setting)
	E_PGA_SDADC24_GAIN_1_1_1	0x00U	Total gain of PGA: ×1 (G_{SET1} = 1, G_{SET2} = 1)
	E_PGA_SDADC24_GAIN_2_1_2	0x04U	Total gain of PGA: ×2 (G_{SET1} = 2, G_{SET2} = 1)
	E_PGA_SDADC24_GAIN_3_1_3	0x08U	Total gain of PGA: ×3 (G_{SET1} = 3, G_{SET2} = 1)
	E_PGA_SDADC24_GAIN_4_1_4	0x0CU	Total gain of PGA: ×4 (G_{SET1} = 4, G_{SET2} = 1)
	E_PGA_SDADC24_GAIN_8_1_8	0x10U	Total gain of PGA: ×8 (G_{SET1} = 8, G_{SET2} = 1)
	E_PGA_SDADC24_GAIN_1_2_2	0x01U	Total gain of PGA: ×2 (G_{SET1} = 1, G_{SET2} = 2)
	E_PGA_SDADC24_GAIN_2_2_4	0x05U	Total gain of PGA: ×4 (G_{SET1} = 2, G_{SET2} = 2)
	E_PGA_SDADC24_GAIN_3_2_6	0x09U	Total gain of PGA: ×6 (G_{SET1} = 3, G_{SET2} = 2)
	E_PGA_SDADC24_GAIN_4_2_8	0x0DU	Total gain of PGA: ×8 (G_{SET1} = 4, G_{SET2} = 2)
	E_PGA_SDADC24_GAIN_8_2_16	0x11U	Total gain of PGA: ×16 (G_{SET1} = 8, G_{SET2} = 2)
	E_PGA_SDADC24_GAIN_1_4_4	0x02U	Total gain of PGA: ×4 (G_{SET1} = 1, G_{SET2} = 4)
	E_PGA_SDADC24_GAIN_2_4_8	0x06U	Total gain of PGA: ×8 (G_{SET1} = 2, G_{SET2} = 4)
	E_PGA_SDADC24_GAIN_3_4_12	0x0AU	Total gain of PGA: ×12 (G_{SET1} = 3, G_{SET2} = 4)
	E_PGA_SDADC24_GAIN_4_4_16	0x0EU	Total gain of PGA: ×16 (G_{SET1} = 4, G_{SET2} = 4)
	E_PGA_SDADC24_GAIN_8_4_32	0x12U	Total gain of PGA: ×32 (G_{SET1} = 8, G_{SET2} = 4)
	E_PGA_SDADC24_GAIN_1_8_8	0x03U	Total gain of PGA: ×8 (G_{SET1} = 1, G_{SET2} = 8)
	E_PGA_SDADC24_GAIN_2_8_16	0x07U	Total gain of PGA: ×16 (G_{SET1} = 2, G_{SET2} = 8)
	E_PGA_SDADC24_GAIN_3_8_24	0x0BU	Total gain of PGA: ×24 (G_{SET1} = 3, G_{SET2} = 8)
	E_PGA_SDADC24_GAIN_4_8_32	0x0FU	Total gain of PGA: ×32 (G_{SET1} = 4, G_{SET2} = 8)

Table 5-7 Enumeration Types in r_sdadc24_control_api.h (2/2)

Type Name	**e_pga_sdadc24_offset_t**		
Member	**Name**	**Value**	**Description (Offset Voltage Setting (mV))**
	E_PGA_SDADC24_OFFSET_M164P06	0x11U	–164.06 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M153P13	0x12U	–153.13 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M142P19	0x13U	–142.19 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M131P25	0x14U	–131.25 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M120P31	0x15U	–120.31 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M109P38	0x16U	–109.38 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M98P44	0x17U	–98.44 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M87P50	0x18U	–87.50 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M76P56	0x19U	–76.56 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M65P63	0x1AU	–65.63 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M54P69	0x1BU	–54.69 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M43P75	0x1CU	–43.75 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M32P81	0x1DU	–32.81 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M21P88	0x1EU	–21.88 / G_{SET1}
	E_PGA_SDADC24_OFFSET_M10P94	0x1FU	–10.94 / G_{SET1}
	E_PGA_SDADC24_OFFSET_0P00	0x00U	0.00 / G_{SET1}
	E_PGA_SDADC24_OFFSET_10P94	0x01U	10.94 / G_{SET1}
	E_PGA_SDADC24_OFFSET_21P88	0x02U	21.88 / G_{SET1}
	E_PGA_SDADC24_OFFSET_32P81	0x03U	32.81 / G_{SET1}
	E_PGA_SDADC24_OFFSET_43P75	0x04U	43.75 / G_{SET1}
	E_PGA_SDADC24_OFFSET_54P69	0x05U	54.69 / G_{SET1}
	E_PGA_SDADC24_OFFSET_65P63	0x06U	65.63 / G_{SET1}
	E_PGA_SDADC24_OFFSET_76P56	0x07U	76.56 / G_{SET1}
	E_PGA_SDADC24_OFFSET_87P50	0x08U	87.50 / G_{SET1}
	E_PGA_SDADC24_OFFSET_98P44	0x09U	98.44 / G_{SET1}
	E_PGA_SDADC24_OFFSET_109P38	0x0AU	109.38 / G_{SET1}
	E_PGA_SDADC24_OFFSET_120P31	0x0BU	120.31 / G_{SET1}
	E_PGA_SDADC24_OFFSET_131P25	0x0CU	131.25 / G_{SET1}
	E_PGA_SDADC24_OFFSET_142P19	0x0DU	142.19 / G_{SET1}
	E_PGA_SDADC24_OFFSET_153P13	0x0EU	153.13 / G_{SET1}
	E_PGA_SDADC24_OFFSET_164P06	0x0FU	164.06 / G_{SET1}

5.3.3 Macro Definitions

The following describes the macros defined in this sample program.

5.3.3.1 User-Modifiable Macros

The following macros can be modified by the user as required and the new settings are applied in the processing by rebuilding the code.

Table 5-8 Definitions in hal_entry.h (Modifiable by the User)

Definition Name	Type	Initial Value	Description
D_WEIGHT_SAMPLE_NUM	size_t	244U	Number of samples used to obtain the averaged A/D converted value
D_ARRAY_LEN	uint8_t	32U	Size of the ring buffers for storing A/D converted values and results of weight measurement

Table 5-9 Definitions in r_sdadc24_control_api.h (Modifiable by the User)

Definition Name	Type	Initial Value	Description
D_SDADC24_WEIGHT_CHANNEL	adc_channel_t	ADC_CHANNEL_0	Channel number to be used
D_SDADC24_GAIN	e_pga_sdadc24_gain_t	E_PGA_SDADC24_GAIN_8_4_32	Total gain of PGA: ×32 (G_{SET1} = 8, G_{SET2} = 4)
D_SDADC24_OFFSET	e_pga_sdadc24_offset_t	E_PGA_SDADC24_OFFSET_0P00	Offset voltage 0.00 / G_{SET1} (mV)
D_SDADC24_DISCONN_NUM	uint8_t	10U	Maximum number of A/D conversions in disconnection detection processing

Table 5-10 Definitions in r_weight_calculate_api.h (Modifiable by the User)

Definition Name	Type	Initial Value	Description
D_LOADCELL_MAX_WEIGHT	float	500.0F	Rated capacity of the load cell (upper limit)
D_LOADCELL_MIN_WEIGHT	float	–500.0F	Rated capacity of the load cell (lower limit)
D_LOADCELL_OVERLOAD_PERCENT	float	125.0F	Threshold for checking the overweight (%)*1
D_LOADCELL_COEFFICIENT_A	float	0.000987142F	Coefficient a (slope) of the weight conversion equation*2
D_LOADCELL_COEFFICIENT_B	float	–50.10420444F	Coefficient b (intercept) of the weight conversion equation*2
D_WEIGHT_STABILITY_CHECK_VALUE	uint32_t	30U	Allowable range in checking for the stability of measured weight values (mg)*3
D_WEIGHT_STABILITY_CHECK_NUM	uint8_t	3U	Number of times to check for the stability of measured weight values*3

Note 1. When the result of weight measurement before zero adjustment has exceeded D_LOADCELL_MAX_WEIGHT * D_LOADCELL_OVERLOAD_PERCENT or D_LOADCELL_MIN_WEIGHT * D_LOADCELL_OVERLOAD_PERCENT, the disconnection detection processing is executed.

Note 2. The initial values are the coefficients calculated when the operation of this sample program was confirmed.

Note 3. When the difference between the previous and current measured weights is within the range ±D_WEIGHT_STABILITY_CHECK_VALUE the consecutive D_WEIGHT_STABILITY_CHECK_NUM number of times, the measured weight values are judged to be stable.

5.3.3.2 Macros

The following macro values in this sample program are fixed.

Table 5-11 Definitions in board_cfg.h

Definition Name	Type	Setting	Description
OFF	bsp_io_level_t	0U	LED off
ON	bsp_io_level_t	1U	LED on
RED	bsp_led_t	BSP_LED_LED1	LED1 control port (P205)
TURN_RED_ON	fsp_err_t	R_IOPORT_PinWrite (&g_ioport_ctrl, g_bsp_leds.p_leds[RED], ON);	LED1 turns on.
TURN_RED_OFF	fsp_err_t	R_IOPORT_PinWrite (&g_ioport_ctrl, g_bsp_leds.p_leds[RED], OFF);	LED1 turns off.

Table 5-12 Definitions in hal_entry.h

Definition Name	Type	Setting	Description
D_MESSAGE_STEP_ZEROADJUST	uint8_t	2U	Number of steps (divisions) of each message transmission
D_MESSAGE_STEP_WEIGHT	uint8_t	3U	
D_MESSAGE_STEP_OVERFLOW	uint8_t	2U	
D_MESSAGE_STEP_DISCONNECT	uint8_t	3U	
APP_ERR_TRAP(a)	void	if (a) {__asm("BKPT #0¥n");}	Error trap processing (break)

Table 5-13 Definitions in r_ring_buffer_control_api.h

Definition Name	Type	Setting	Description
D_NO_UPDATE	bool	false	Index update flag to be passed as an argument of the R_RINGBUF_GetData function (the index is not updated after data are read).
D_UPDATE	bool	true	Index update flag to be passed as an argument of the R_RINGBUF_GetData function (the index is updated after data are read).
D_W_INDEX	uint8_t	1U	Index type specification to be passed as an argument of the R_RINGBUF_SetDataIndex function (write pointer).
D_R_INDEX	uint8_t	0U	Index type specification to be passed as an argument of the R_RINGBUF_SetDataIndex function (read pointer).

Table 5-14 Definition in r_sdadc24_control_api.h

Definition Name	Type	Setting	Description
D_SDADC24_OVERFLOW_THRESH	long	0x01000000L	SDADC24 overflow criterion A value of or greater than 0x01000000L is considered to represent an overflow.

Table 5-15 Definitions in r_weight_calculate_api.h

Definition Name	Type	Setting	Description
D_LOADCELL_MAX_OVERLOAD	float	D_LOADCELL_MAX_WEIGHT * (D_LOADCELL_OVERLOAD_PERCENT / 100.0F)	Overweight criterion for the load cell (upper limit)
D_LOADCELL_MIN_OVERLOAD	float	D_LOADCELL_MIN_WEIGHT * (D_LOADCELL_OVERLOAD_PERCENT / 100.0F)	Overweight criterion for the load cell (lower limit)

5.3.4 Structures

The following describes the structures used in this sample program.

Table 5-16 Structures in hal_entry.c

Type Name	st_weight_dataset_t		
Member	**Type**	**Name**	**Description**
	float	weight_uncorrect	Weight (g) before zero adjustment
	float	weight	Weight (g) after zero adjustment
	float	average_uncorrect	Averaged A/D converted value (including the gain and offset)
	float	average_correct	Averaged A/D converted value (with the gain and offset eliminated)
Type Name	**st_message_request_t**		
Member	**Type**	**Name**	**Description**
	bool	flag	Message transmission request flag
	uint8_t	step	Message transmission step
Type Name	**st_message_control_t**		
Member	**Type**	**Name**	**Description**
	st_message_request_t	zeroadjust	ZeroAdjust message transmission request
	st_message_request_t	weight	Weight message transmission request
	st_message_request_t	overflow	Overflow message transmission request
	st_message_request_t	disconnect	DisconnectDetect message transmission request

Table 5-17 Structure in r_sdadc24_control_api.h

Type Name	st_gain_setting_t		
Member	**Type**	**Name**	**Description**
	uint8_t	gset1	Gain of the first-stage amplifier of the PGA
	uint8_t	gset2	Gain of the second-stage amplifier of the PGA
	uint8_t	gtotal	Total gain of the PGA

Table 5-18 Structure in r_usb_control_api.h

Type Name	st_usbctrl_ctrl_t		
Member	**Type**	**Name**	**Description**
	usb_instance_ctrl_t *	p_ctrl	Pointer to the USB instance
	usb_status_t *	p_status	Pointer to the USB driver status structure
	usb_setup_t *	p_setup	Pointer to the setup packet structure
	usb_pcdc_linecoding_t *	p_linecoding	Pointer to the LineCoding structure
	uint8_t *	p_readbuf	Pointer to the USB read buffer
	uint8_t	readbuf_size	Size of the USB read buffer

5.3.5 Functions

The following shows a list of functions.

5.3.5.1 hal_entry.c (Main Processing)

Table 5-19 Functions in hal_entry.c (1/2)

Function Name	hal_entry				
Overview	main function (This is called from the main function in main.c.)				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	—	void	—	—

Function Name	R_BSP_WarmStart				
Overview	Opens I/O ports. (This is called from the SystemInit function in system.c.)				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	—	void	—	—

Function Name	sw_check				
Overview	Detects whether or not USER_BTN is being pressed.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
fsp_err_t	Return value from the FSP	O	bool *	p_sw_status	Pointer to the location for storing the indicator of whether or not USER_BTN is being pressed. true: Pressed false: Not pressed

Function Name	message_transmission				
Overview	Sends a message through the USB interface in response to the state of the transmission request flags for individual types of messages.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
fsp_err_t	Return value from the FSP	—	void	—	—

Function Name	calc_average				
Overview	Calculates an arithmetic average. When the value of w_index reaches that of size, this function calculates the average and stores it in the output location.				
Return Value		Argument			
Type	Value	I/O	Type	Variable Name	Description
bool	Average calculation state true: Completed false: Not completed	I	float	input	A/D value to be added as a target of calculation
		I	size_t	size	Number of samples used for average calculation
		I	float *	p_array	Pointer to the buffer used for average calculation
		I/O	uint32_t *	p_w_index	Pointer to the position for writing in the buffer
		O	float *	p_output	Pointer to the location for storing the result of average calculation

Table 5-20 Functions in hal_entry.c (2/2)

Function Name	init_variables				
Overview	Initializes the following variables. • Position for writing in the buffer for average calculation • Positions for reading and writing in the buffer for storing measured weight values • Zero adjustment value				
Return Value		Argument			
Type	Value	I/O	Type	Variable Name	Description
void	—	—	void	—	—

Function Name	app_error_terminate				
Overview	Abnormal end processing. This function stops the USB, SDADC24, and GPT and turns LED1 off.				
Return Value		Argument			
Type	Value	I/O	Type	Variable Name	Description
void	—	—	void	—	—

5.3.5.2 r_interrupt_callback.c (Interrupt Callback Processing)

Table 5-21 Functions in r_interrupt_callback.c

Function Name	R_SDADC24_conversion_callback				
Overview	Callback function for the A/D conversion end interrupt in the SDADC24 (in continuous scan mode). This function reads an A/D converted value from the SDADC24 and stores it in the ring buffer for A/D converted values. If an overflow has occurred, the converted value is stored and the bit to indicate an overflow is set.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	—	adc_callback_args_t *	p_args	Pointer to the ADC callback argument structure

Function Name	R_SDADC24_conversion_oneshot_callback				
Overview	Callback function for the A/D conversion end interrupt in the SDADC24 (in one-shot operation). This function reads an A/D converted value from the SDADC24 and stores it in the global variable. If an overflow has occurred, the converted value is stored and the bit to indicate an overflow is set.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	—	adc_callback_args_t *	p_args	Pointer to the ADC callback argument structure

Function Name	R_GPT166_blinker_callback				
Overview	Callback function for the counter overflow interrupt in channel 6 (GPT166) of the GPT. This function toggles the signal on P205 (LED1).				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	—	timer_callback_args_t *	p_args	Pointer to the timer callback argument structure

5.3.5.3 r_ring_buffer_control_api.c (Ring Buffer Control Processing)

Table 5-22 Functions in r_ring_buffer_control_api.c

Function Name	R_RINGBUF_GetData				
Overview	Reads data from a ring buffer				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
size_t	Number of bytes read	I	st_ring_buf_t *	p_ary	Pointer to the target ring buffer
		O	uint8_t *	data	Pointer to the location for storing the read data
		I	size_t	len	Size of data to be read (bytes)
		I	bool	index_ update	Read position update flag true: To be updated false: Not to be updated

Function Name	R_RINGBUF_SetData				
Overview	Writes data to a ring buffer				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
size_t	Number of bytes written. If there is no unused space in the buffer, 0 is returned.	I	st_ring_buf_t *	p_ary	Pointer to the target ring buffer
		I	uint8_t *	data	Pointer to the location that holds the write data
		I	size_t	len	Size of data to be written (bytes)

Function Name	R_RINGBUF_GetDataLength				
Overview	Obtains the amount of data (in bytes) that has been written to a ring buffer but not read.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
size_t	Amount of data (in bytes) that has not been read	I	st_ring_buf_t *	p_ary	Pointer to the target ring buffer

Function Name	R_RINGBUF_SetDataIndex				
Overview	Specifies the position for reading or writing in a ring buffer.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
uint32_t	Value that has been set	I	st_ring_buf_t *	p_ary	Pointer to the target ring buffer
		I	uint32_t	value	Value to be set
		I	uint8_t	select	Selection of read or write 0: Read 1: Write

5.3.5.4 r_sdadc24_control_api.c (SDADC24 Control Processing)

Table 5-23 Functions in r_sdadc24_control_api.c (1/3)

Function Name	R_SDADC24_Init				
Overview	Sets up the SDADC24 setting structures so that the converter can be used for weight measurement. Continuous scan in differential input mode is specified, and the gain and channel are set to those specified for D_SDADC24_GAIN and D_SDADC24_WEIGHT_CHANNEL in r_sdadc24_control_api.h.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	I	adc_cfg_t *	p_cfg	Pointer to the ADC setting structure
		I	sdadc_scan_cfg_t *	p_channel_cfg	Pointer to the ADC scan setting structure
		I	sdadc_channel_cfg_t *	p_channel_cfgs	Pointer to the ADC channel setting structure
		I	sdadc_extended_cfg_t *	p_extended_cfg	Pointer to the ADC extended setting structure

Function Name	R_SDADC24_ConfigDisconnectDetection				
Overview	Sets up the SDASDC24 setting structures so that the converter can be used for disconnection detection. Single scan in single-ended input mode is specified, the gain is set to ×1, and the channel is set to that specified for D_SDADC24_WEIGHT_CHANNEL in r_sdadc24_control_api.h.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	I	adc_cfg_t *	p_cfg	Pointer to the ADC setting structure
		I	sdadc_scan_cfg_t *	p_channel_cfg	Pointer to the ADC scan setting structure
		I	sdadc_channel_cfg_t *	p_channel_cfgs	Pointer to the ADC channel setting structure
		I	sdadc_extended_cfg_t *	p_extended_cfg	Pointer to the ADC extended setting structure
		I	uint8_t	polarity	Selection of the polarity (state of charging) for use in the disconnection detection assistance function 0: Positive (precharge) 1: Negative (discharge)

<table>
<tr><td>Function Name</td><td colspan="5">R_SDADC24_ScanStart</td></tr>
<tr><td>Overview</td><td colspan="5">Specifies the offset voltage and starts scanning.</td></tr>
<tr><td colspan="2">Return Value</td><td colspan="4">Argument</td></tr>
<tr><td>Type</td><td>Value</td><td>I/O</td><td>Type</td><td>Variable Name</td><td>Description</td></tr>
<tr><td rowspan="2">fsp_err_t</td><td rowspan="2">Return value from the FSP</td><td>I</td><td>adc_ctrl_t *</td><td>p_instance</td><td>Pointer to the ADC instance</td></tr>
<tr><td>I</td><td>int32_t</td><td>offset_reg_val</td><td>Value to be set in the PGAOFS[4:0] bits of the PGACn register</td></tr>
</table>

Table 5-24 Functions in r_sdadc24_control_api.c (2/3)

Function Name	R_SDADC24_DisconnectDetectSinglePolarity				
Overview	Disconnection detection processing (with either the positive or negative polarity selected) This function performs A/D conversion with disconnection detection assistance mode enabled and tests for disconnection.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
fsp_err_t	Return value from the FSP	I	adc_ctrl_t *	p_instance	Pointer to the ADC instance
		O	bool *	p_result	Pointer to the location for storing the result of disconnection detection

Function Name	R_SDADC24_DisconnectDetect				
Overview	Disconnection detection processing (for both the positive polarity and negative polarity) This function calls the R_SDADC24_ConfigDisconnectDetection function and then the R_SDADC24_DisconnectDetectSinglePolarity function. It handles processing with the positive polarity selected and then repeats it with the negative polarity selected.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
fsp_err_t	Return value from the FSP	I	adc_cfg_t *	p_cfg	Pointer to the ADC setting structure
		I	sdadc_scan_cfg_t *	p_channel_cfg	Pointer to the ADC scan setting structure
		I	sdadc_channel_cfg_t *	p_channel_cfgs	Pointer to the ADC channel setting structure
		I	sdadc_extended_cfg_t *	p_extended_cfg	Pointer to the ADC extended setting structure
		O	bool *	p_result	Pointer to the location for storing the result of disconnection detection

Function Name	R_SDADC24_GetGain				
Overview	Obtains the setting of the SDADC24 gain				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	O	st_gain_setting_t *	p_gain	Pointer to the gain setting structure
		I	adc_channel_t	adc_channel	Target channel number

Function Name	R_SDADC24_GetOffset				
Overview	Obtains the setting of the SDADC24 offset voltage (voltage converted to the A/D value at the specified total gain).				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
int32_t	Offset voltage setting (voltage converted to the A/D value at the specified total gain)	I	adc_channel_t	adc_channel	Target channel number

Table 5-25 Functions in r_sdadc24_control_api.c (3/3)

Function Name	r_sdadc24_oneshot				
Overview	Executes A/D conversion (one shot). This function is used in disconnection detection.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
fsp_err_t	Return value from the FSP	I	adc_ctrl_t *	p_instance	Pointer to the ADC instance
		O	int32_t *	p_ad_value	Pointer to the location for storing the result of A/D conversion.

5.3.5.5 r_usb_control_api.c (USB Control Processing)

Table 5-26 Functions in r_usb_control_api.c

Function Name	R_USBCTRL_EventPoll				
Overview	Checks USB events and performs the processing corresponding to the detected event.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
fsp_err_t	Return value from the FSP	I	st_usbctrl_ctrl_t *	p_usbctrl_ctrl	Pointer to the USB control data structure

Function Name	R_USBCTRL_Write				
Overview	Starts data transmission.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
fsp_err_t	Return value from the FSP	I	st_usbctrl_ctrl_t *	p_usbctrl_ctrl	Pointer to the USB control data structure
		I	uint8_t *	p_buf	Pointer to the location that holds transmit data
		I	uint32_t	size	Number of data to be sent (bytes)
		I	uint8_t	destination	USB class type

Function Name	R_USBCTRL_GetWriteStatus				
Overview	Obtains the state of USB data transmission.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
bool	State of data transmission true: Completed false: In progress	—	—	—	—

5.3.5.6 r_weight_calculate_api.c (Weight Calculation Processing)

Table 5-27 Functions in r_weight_calculate_api.c (1/2)

Function Name	R_WEIGHT_SetZeroAdjustWeight				
Overview	Specifies the zero adjustment weight (g).				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	I	float	weight	Zero adjustment weight (g)

Function Name	R_WEIGHT_GetZeroAdjustWeight				
Overview	Obtains the zero adjustment weight (g).				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
float	Zero adjustment weight (g)	—	—	—	—

Function Name	R_WEIGHT_ADValueCorrect				
Overview	Eliminates the gain and offset components from an A/D converted value				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
float	Value obtained by eliminating the gain and offset components from the argument ad_value	I	float	ad_value	A/D converted value
		I	st_gain_setting_t *	p_gain	Pointer to the gain setting structure
		I	int32_t	offset	Offset voltage setting (voltage converted to the A/D value at the specified total gain)

Function Name	R_WEIGHT_Calculate				
Overview	Converts an A/D value to weight (g). This value returns the result of (ad_value * coef_a) + coef_b.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
float	Weight (g)	I	float	ad_value	A/D converted value
		I	float	coef_a	Coefficient a (slope) for the weight conversion equation
		I	float	coef_b	Coefficient b (intercept) for the weight conversion equation

Function Name	r_weight_stability_check_init				
Overview	Initializes the buffer used in checking the stability of measured weight values. This function is called from the R_WEIGHT_StabilityCheck function.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
void	—	—	—	—	—

Table 5-28 Functions in r_weight_calculate_api.c (2/2)

Function Name	R_WEIGHT_StabilityCheck				
Overview	Checks the stability of measured weight values. When the differences between the previous and current measured values are within the range of ±m mg for the consecutive number n of rounds of measurement, the measured weight values are judged to be stable. For n and m, D_WEIGHT_STABILITY_CHECK_NUM and D_WEIGHT_STABILITY_CHECK_VALUE defined in r_weight_calculate_api.h are used respectively. When *p_clear_request is set to true, the buffer for checking the stability is cleared, the value of the argument weight is placed in the buffer, after which return proceeds with *p_clear_request having been set to false.				
Return Value		**Argument**			
Type	**Value**	**I/O**	**Type**	**Variable Name**	**Description**
bool	Result of measured weight stability check true: Stable false: Not stable	I	float	weight	Weight (g)
		I	bool *	p_clear_request	Pointer to the variable clear request flag

6. Results of Measurement with the Sample Program

The following describes the results of weight measurement by using the SEN-14728 load cell with some specifications listed in Table 4-2, the EK-RA2A1 board, and the sample program.

6.1 Measurement Conditions

Figure 6-1 shows the configuration of the weight measurement system. The constants for the anti-aliasing circuit are set as follows.

R = 3.3 kΩ, C = 47 uF (fc ≈ 1.026 Hz)

As described in section 4.7, Calculating Weight from the A/D Converted Value, calibration is carried out for two weight points: 0 g (no load) and 500 g. The average of ten A/D values output to Teraterm was used in calibration for each weight.

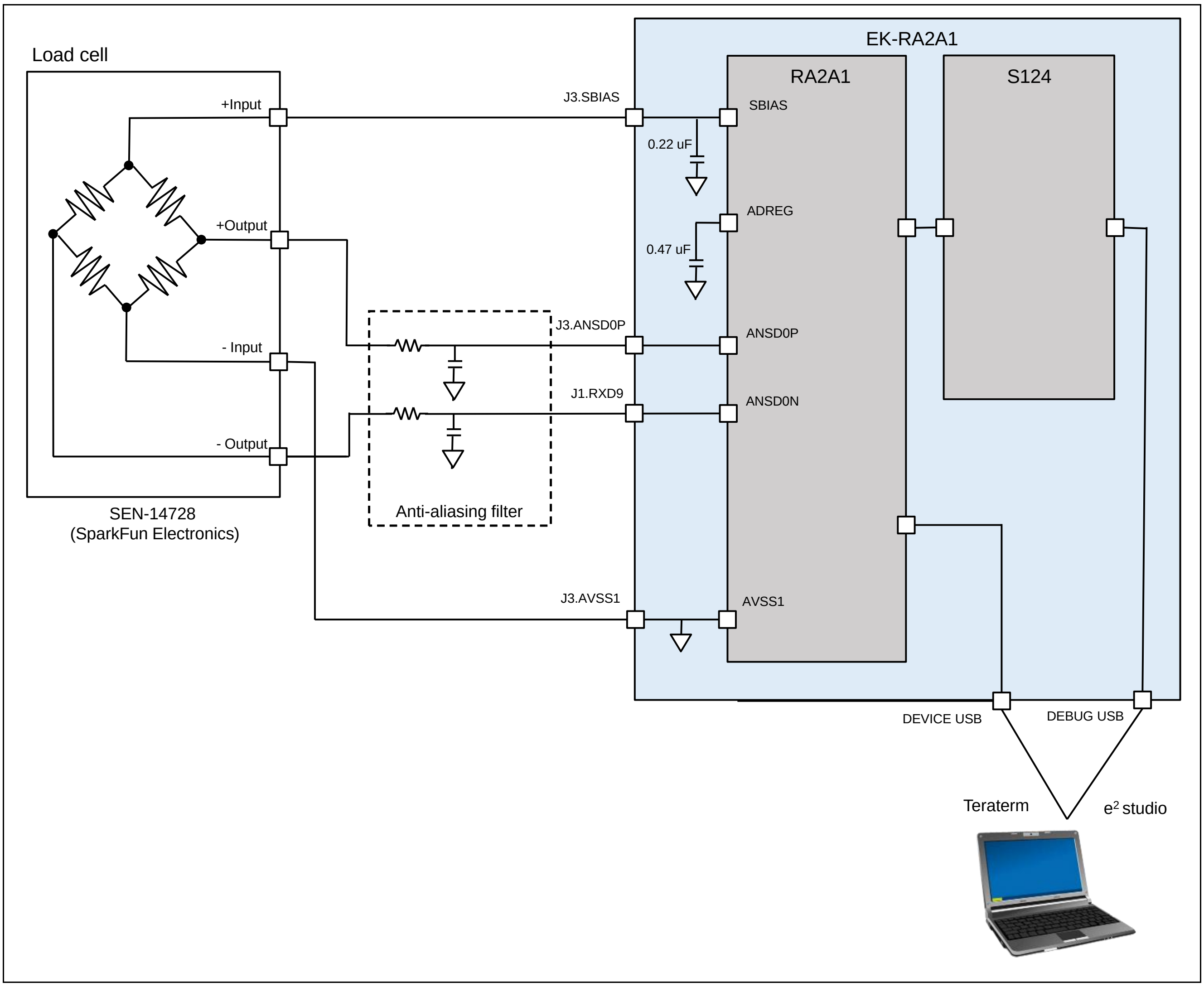

Figure 6-1 Configuration of Weight Measurement by Load Cell

6.2 Measurement Results

The results of measuring a test weight of 500 g (error: 100 mg) are shown in Figure 6-2.

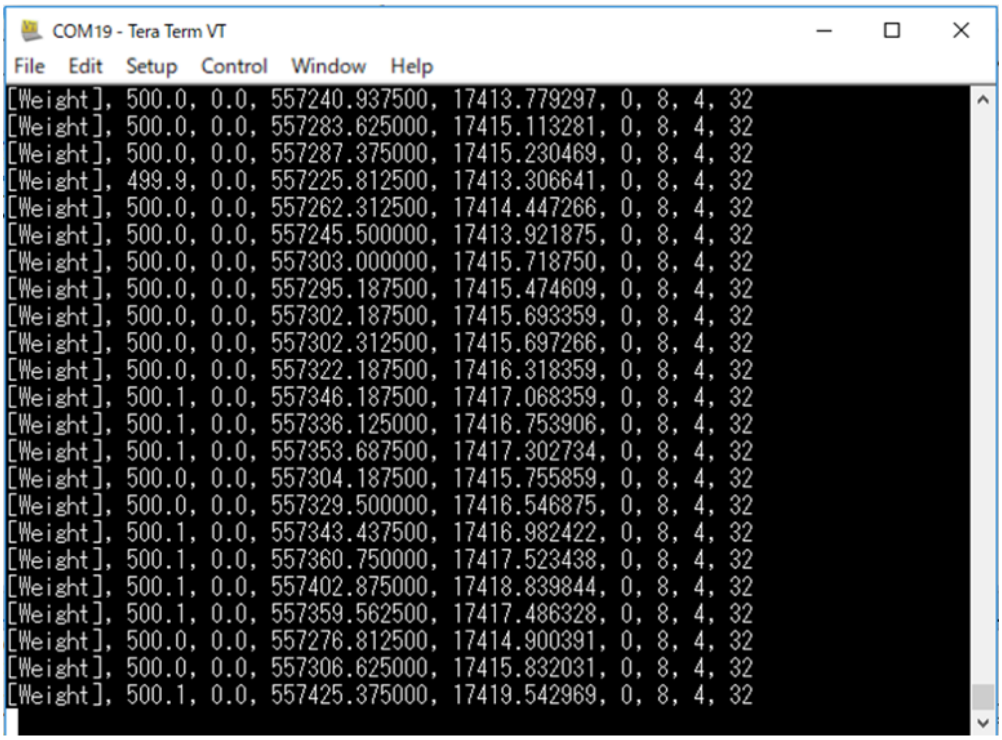

Figure 6-2 Results of Measuring a 500-g Test Weight

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2020.12.01	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit 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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.