RL78/I1E

Strain Gauge Solution

R01AN2821EJ0101
Rev.1.01
Mar. 01, 2016

Introduction

This application note explains the Renesas solution for measuring weight using a load cell, a type of strain gauge, with the RL78/I1E microcontroller.

Target Device

RL78/I1E

Contents

1. Outline

The essential elements to high-precision sensing of sensor output signals are an amplifier circuit, offset adjustment, a precision A/D converter. RL78/I1E embeds all three features, offering high-precision sensing with a strain gauge, thermoelectric coupling, and a resistance temperature detector.

This application note explains how to measure weight using a load cell as the strain gauge and RL78/I1E.

A load cell combines multiple strain gauges and a strain body to detect applied force. The strain gauge, a key structural element, is a sensor that generates resistance change when its shape is distorted. This reaction is based on the principle that, when a metal object is expanded, it experiences electrical resistance changes due to changes in area and length. The term “strain body” refers to the metal component that is processed to change shape (distort) when applied with force. The load cell is attached to a position on the strain gauge that generates the greatest strain within the strain body for the most efficient strain detection. Note that the strain gauge resistance change is directly relational to the strain, which means changes in strain can also be detected when measuring for changes in resistance.

Load cell types vary according to the structure of the strain body. The device used in the solution described here is a beam-type single point load cell (1004-00.6-JW00-RS) manufactured by Tedea Huntleigh. This load cell consists of 4 strain gauges in Wheatstone bridge configuration. When voltage is applied, the strain gauges convert the change in the resistance value due to the applied force (weight) to a change in voltage, which is output as an electric signal. The strain gauge itself, rather than the load cell, can also be used. As the change in the strain gauge resistance value is also quite minute even, it is used in a bridge circuit that provides an expanded dynamic range. Bridge circuits are available in half and full configurations. A half bridge circuit uses two fixed resistors and two strain gauges, while the full bridge circuit uses four strain gauges. The bridge configuration enables the user to mitigate strain gauge temperature characteristics (zero point, sensitivity, etc).

Tedea Huntleigh’s load cell boasts a capacity (maximum measurable weight) of 600g and an output of 0.8787mV/V. Therefore, with applied voltage of 2V, you need to detect a voltage differential of 2.9uV to take measurements in 1g units. The solution described here uses the RL78/I1E’s built-in 24-bit ΔΣA/D converter with programmable gain instrumentation amplifier to amplify the minute voltage difference and then convert the difference to a digital value. A digital noise filter is used to remove noise from the retrieved digital value, which is then converted to a physical quantity. The measured value is finally output to a host device such as a personal computer. Figure 1-1 shows the system block configuration used in this application note.

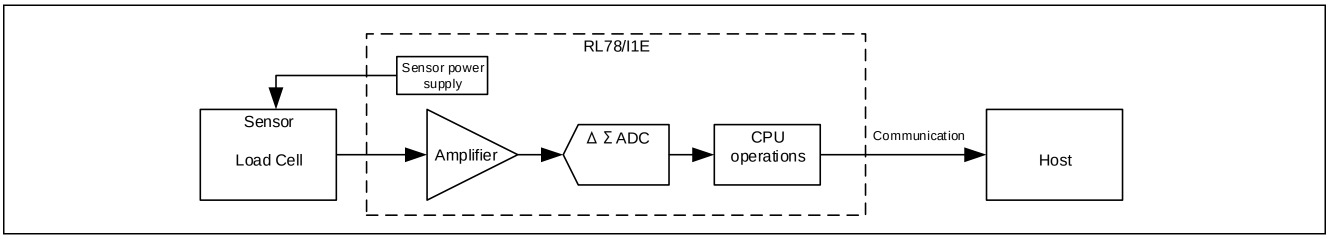


Figure 1-1 System Block Configuration

2. Related Documents

Documents related to this application note are shown below. Please refer to these as needed.

- RL78/I1E Strain Gauge PC Application Software Manual (R01AN2822) Application Note
- RL78/I1E Analog Characteristics Evaluation PC Application Software Manual (R01AN2820) Application Note
- RL78/I1E Analog Characteristics Evaluation Sample Code Specification (R01AN2819) Application Note

3. Operating Conditions

This section describes the operating conditions required for using the strain gauge solution as described in this application note.

Table 3-1 Operating Conditions

Item	Specification
Main components	MCU: RL78/I1E (R5F11CCC) Load cell: 1004-00.6-JW00-RS (manufactured by Tedea Huntleigh)
Operating voltage	2.7 to 5.5V
Integrated development environment	CS+ for CA, CX V3.01.00 [19 Aug 2015]
C compiler (build tool)	CA78K0R V1.71

4. Hardware Configuration

This section explains the strain gauge specifications and circuit diagrams use in the hardware configuration.

4.1 Strain Gauge Specifications

Table 4-1 shows the specifications of the load cells used in the strain gauge solution described in this application note.

Table 4-1 Load Cell (1004-00.6-JW00-RS, manuf. by Tedea Huntleigh) Specifications

Item	Description
Capacity	0.6 kg
Output @R.C.	0.8787 mV/V
Input Impedance	415 ± 20 ohm
Output Impedance	350 ± 3 ohm
Insulation Resistance	2 G ohm
Zero Balance	0.0028 mV/V
Test Excitation	10 V DC

Note Table 4-1 is a partial extraction from the data sheet. For more details, please refer to the latest corresponding data sheet.

4.2 Circuit Diagram

Figure 4-1 shows the connection circuit for the load cell and RL78/I1E.

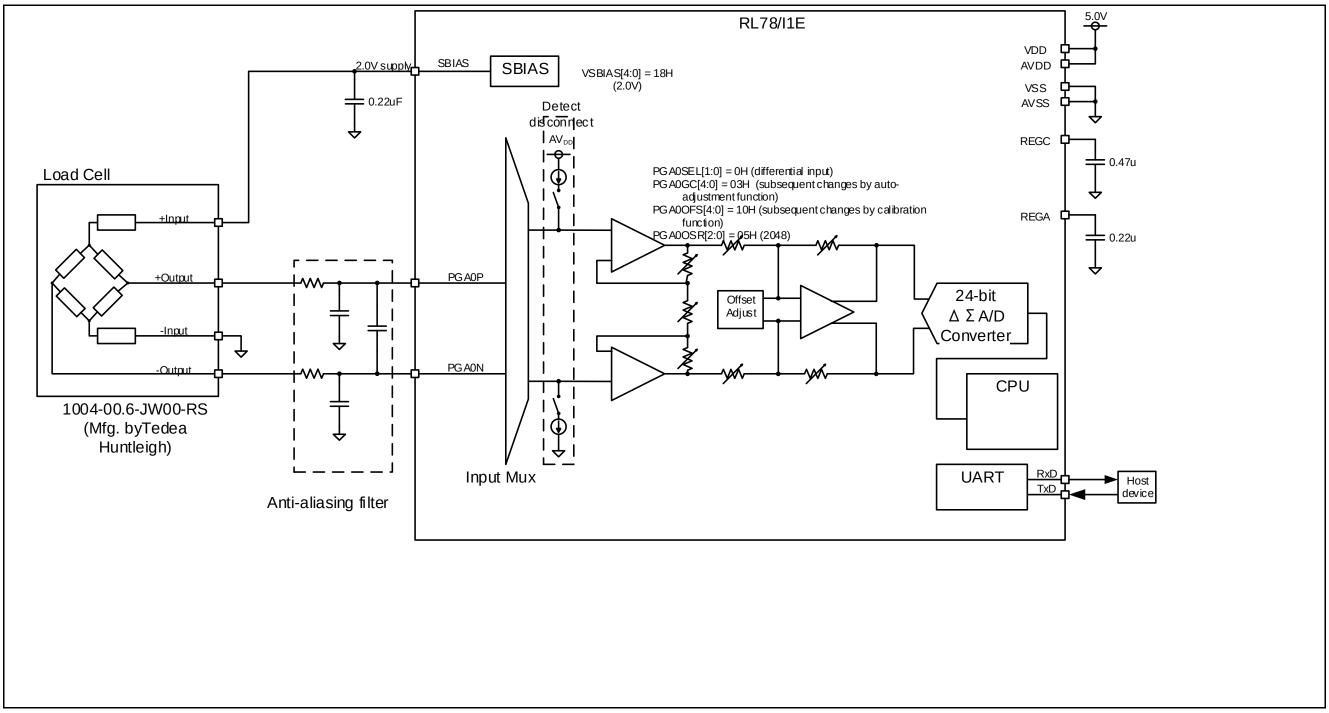


Figure 4-1 Connection Circuit for Load Cell and RL78/I1E

Note: This circuit diagram is an abbreviated version of the actual diagram to provide an overall view of the connection. When designing the actual circuit, make sure all pins are properly treated and electrical characteristics meet requirements of the corresponding specification by evaluating thoroughly before using.

As shown in Figure 4-1, the driving power for the load cell is supplied by power supply circuit SBIAS, which is used for the RL78/I1E sensor. SBIAS is also used as reference supply for the ΔΣA/D converter. Supplying power from SBIAS to the sensor cancels out power supply fluctuations, enabling a more accurate measurement. SBIAS can output a maximum current of 5mA. The load cell input impedance is 415 ohm (as per data sheet), which means the supply voltage will be 5mA*415 ohm =2.075V or less. Combined with the input voltage range, described later in this document, SBIAS applies 2.0V (output current = SBIAS output voltage (2.0V)/load cell input impedance (415 ohm) = 4.8mA). When the load applied to the load cell from the load cell driving voltage is a fixed quantity, the voltage output from the load cell is 1.757mV (output voltage = fixed output (0.8787mV/V) x applied voltage (2.0V) = 1.757mV). Therefore, the output voltage is 0.293uV when the 0.1g load is applied to the load cell (output voltage = output voltage @ capacity (1.757mV)/capacity (0.6 kg)/measured minimum capacity (0.1g) = 0.2929uV).

An anti-aliasing filter is connected between the load cell output pin and the programmable gain instrumentation amplifier input pin. An anti-aliasing filter is a low-pass filter application that removes aliasing (repeating) errors that occur during sampling. This low-pass filter attenuates more than half of the sampling frequency signals, preventing the occurrence alias errors.

Load cell output pins +Output and -Output are connected to the programmable gain instrumentation amplifier input pins PGA0P and PGA0N on RL78/I1E through the anti-aliasing filter, as shown in Figure 4-1. The programmable gain instrumentation amplifier amplifies the output signal from the load cell and the ΔΣA/D converter converts the signal to a digital value. The PGA operates in differential input mode, single-ended input mode, or internal temperature sensor input mode, according to the setting of the input multiplexer used. In addition, the instrumentation amplifier can be set to 1- to 64-fold gain by combining preamplifier and post amplifier gains.
The programmable gain instrumentation amplifier uses the differential input mode because differential output signal is output from the load cell. At this time, the input voltage range of programmable gain instrumentation amplifier pins PGA0P and PGA0N is 0.2V to 1.8V and the middle voltage 1.0V; input must fit in this voltage range. If the load cell driving voltage is 2.0V, the voltage output from the load cell will hover around 1.0V, providing the widest input voltage range for measurement. The D/A converter for offset voltage adjustment is connected in the instrumentation amplifier's preamplifier. The offset voltage can be adjusted (from -164 mV to +164 mV, in 31 steps (5 bits)) by using this D/A converter. For more details, refer to the explanation in section 5.1(1)Calibration function, subsection a)Input range adjustment function.

The RL78/I1E's ΔΣA/D converter must be set in the oversampling ratio, operating, or similar mode. Oversampling is a method of sampling input at a higher frequency than that of the required signal band width. In normal sampling, half of the sampling frequency is the signal bandwidth. However, by sampling at K times the normal sampling frequency, the noise density of the quantization noise (noise per 1Hz) is reduced by $1/\sqrt{K}$. Although the total amount of quantization noise is the same as in normal sampling, oversampling disperses the quantization noise up to K times the frequency. When noise is removed from the non-signal bandwidth area using a digital filter, better noise reduction can be expected with oversampling frequency multiplier K, resulting in higher measurement accuracy. In other words, the higher the oversampling ratio, the better the accuracy.

Weight measurement, the target of this application note, does not require high-speed conversion, because the signal from the load cell is output as a direct current signal. The impact of noise must be removed as much as possible in order to achieve 0.1g measurement resolution. Therefore, the ΔΣA/D converter oversampling ratio is set to 2048, the largest ratio. The operating mode is set to normal, allowing the output data rate to be set to 488.28sps. The low-power mode setting is also available. However, while reducing current consumption output data rate, it also reduces the output data rate compared to that of normal mode, in which the AFE operation clock frequency is divided by 8. When setting your system to optimal values, keep in mind that operating modes generate a direct trade-off between current consumption and measurement accuracy.

The next step in the process is the averaging of the retrieved A/D converted value. Although oversampling is used to reduce the quantization error, any remaining noise is removed in the averaging process. Among the various noise factors, noise related to power supply, 50Hz and 60Hz, must be removed. As described above, the output data rate of A/D conversion is set to 488.28sps. The measurement results of the weight are generally displayed as a sample rate in the LCD, preferably for about 1 second, easily viewed by the human eye. However, the sample rate display for this application note is 0.5 seconds. Therefore, the averaging processing is executed 244 times on data retrieved within 0.5 seconds, in an integer multiple close to the 50Hz and 60Hz power supply noise. The averaging processing can be executed as in internal function of the RL78/I1E ΔΣA/D converter. However, the maximum number of averaging is 64, which means the averaging processing cannot be achieved in the solution described in this application note. The programmable gain instrumentation amplifier's internal averaging processing can be used for systems requiring 64 or fewer times.

When selecting a load cell and measurement accuracy, note that some programmable gain instrumentation amplifiers require a preamplifier in the previous stage. This programmable gain instrumentation amplifier has a maximum gain of x64, and therefore may not provide enough gain for your system specifications. If you need to select a preamplifier, please keep the following factors in consideration.

- **Low input offset voltage**
  The D/A converter is connected to the programmable gain instrumentation amplifier (PGA) for post-amplifier offset voltage adjustment. The D/A converter can be used in the differential input mode to execute offset voltage adjustment (-164mV to +164mV, 31 stages: 5 bits). The preamplifier is connected in the latter stage of the instrumentation amplifier. Therefore, its input offset voltage is adjusted after amplification in the preamplifier or in the instrumentation amplifier’s early stage. However, when both gains and the input offset voltage are large, adjustment is difficult. When both gains are large and the offset voltage is large, adjustment may be impossible. For example, if preamplifier gain is x10 and PGA preamplifier gain is x8, the cancellable preamplifier input conversion offset voltage is ±164mV/(10*8)=2mV
- **Low 1/f noise, high slew-rate**
  When using the direct current amplification to output from the load cell, as described in this application note, you will need to reduce the 1/f noise of the preamplifier. If 1/f noise is too large, amplification of the preamplifier and the PGA will cause 1/f to be amplified as well, drowning the signal in noise. The lower the frequency, the greater the 1/f noise, which makes this noise source difficult to remove with averaging or other software processing.
  An amplifier with a high slew rate must be selected if switching output from the load cell with an analog switch or using an alternating current (AC) signal with an alternating current drive. A low slew rate may prevent accurate data measurement (due to the drive signal frequency or other factor).

RL78/I1E comes with a range of other functions required for weight measurement and other industrial-use sensor systems. For example, the disconnection detection function prevents line downtime by detecting electrical wiring failure between the sensor and RL78/I1E. The configurable amplifier and 12-bit D/A converter can be combined with an external transistor to control 4-20mA communications.

In addition, RL78 series MCUs are equipped with a data flash function, enabling easy recording of trimming and other data at shipment. The low-voltage detection circuit helps suppress malfunctions due to sudden power failure.

5. Software Specifications

This section describes the software specifications for the strain gauge solution.

5.1 Required Weight Measurement Functions

This section explains the functions necessary for the weight measurement: calibration function (input range adjustment function, multi-point calibration), auto-gain adjustment, physical quantity conversion, zero adjustment function, and error judgement processing.

(1) Calibration function

The load cell gain varies depending on the type and place of sensor attachment. In addition, the input ranges of the RL78/I1E programmable gain instrumentation amplifier and ΔΣA/D converter differ from the output range of the load cell. Therefore, RL78/I1E must perform calibrations to ensure accurate measurements. This application note explains the calibration functions available in the strain gauge solution: input range adjustment function and multi-point calibration function. Calibration requires an externally applied calibration signal. Therefore, rather than calibrating each time a measurement is taken, calibration is executed the first time the MCU is booted up, such as at shipment, allowing calibration data to be used after shipment for measurements. The input range adjustment function calculates the RL78/I1E programmable gain instrumentation amplifier offset voltage and the multi-point calibration function calculates the proportional relationship between the weight and A/D converted value. Following calibration, a process is executed to store the retrieved data in the RL78/I1E data flash.

a) Input range adjustment function

The input range adjustment function is executed to prevent the ΔΣA/D converter input voltage from exceeding the input range using the offset error of the load cell or programmable gain instrumentation amplifier. The function uses the D/A converter for offset voltage adjustment which is integrated in the post-amplifier of the RL78/I1E programmable gain instrumentation amplifier. Figure 5-1 shows the function execution flow.

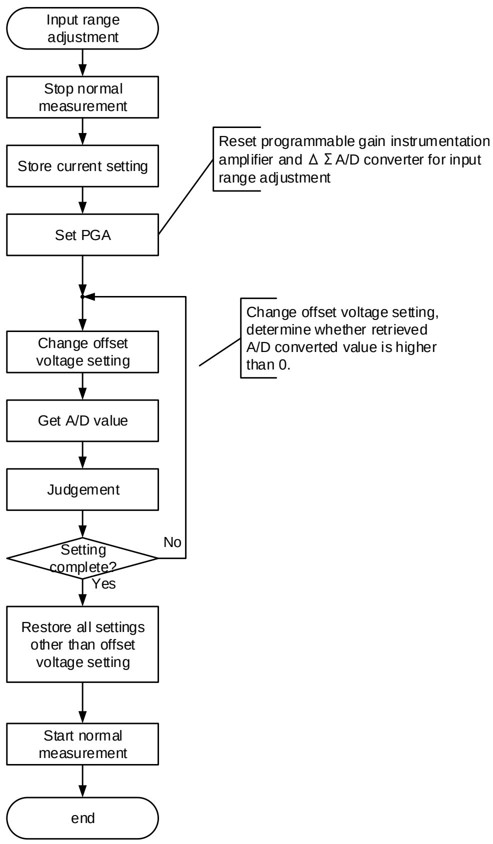


Figure 5-1 Input Range Adjustment Flow

b) Multi-point calibration function

Datasheet values can be used when converting the measured A/D converted value to a physical quantity (grams). However, the actual measured A/D conversion value includes individual differences of the load cell and gain and offset errors of the programmable gain instrumentation amplifier. While these errors must be taken into consideration when calculating the physical quantity, measuring each of these errors is extremely difficult. Therefore, in this application note, employs linear interpolation to generate the physical quantity of the A/D converted value. Linear interpolation is a common approximation method which measures 2 points of data and determines the linear function (slope and intercept) or point on a straight line where the two points meet. When measuring the two points of data, the accuracy of the weight, which serves as the external calibration data, is very important. The execution flow of two-step calibration is shown in Figure 5-2.

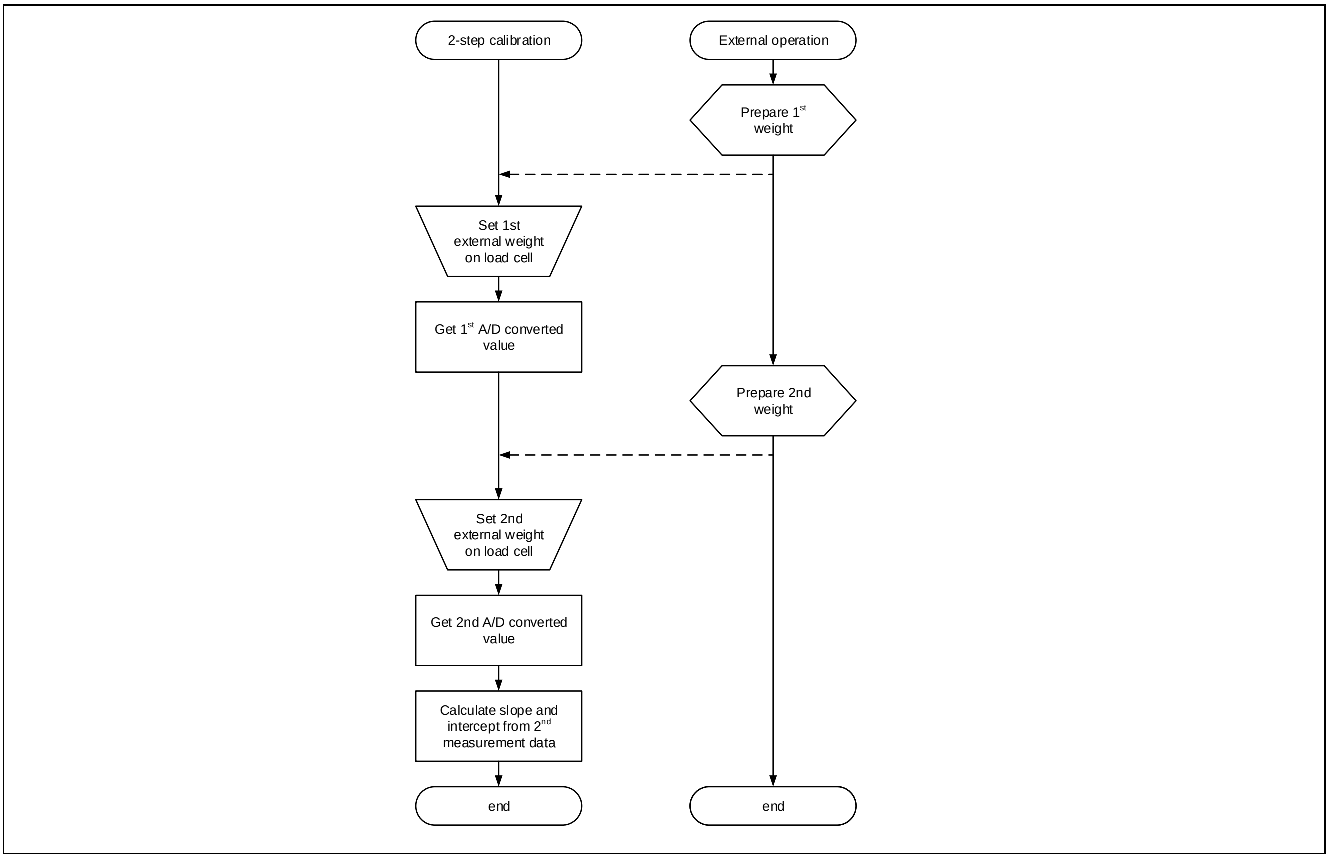


Figure 5-2 2-step Calibration Execution Flow

In 2-step calibration, as shown in Figure 5-2, two points of data are measured to calculate the linear function. Each coefficient in the calculated linear function is a value that includes variations due to zero balance and weight/resistance. Therefore, use of the linear function to convert the A/D converted value in to a physical quantity will not be affected by such differentials.

On the other hand, the place and method in which the load cell is attached may generate errors in the linear function calculated in the 2-stage calibration. In particular, the load cell deflection differs in relation to a lighter and heavier load applied to the cell. This disturbs the linearity, making it difficult to approximate a value with just on linear function. Figure 5-3 shows an example using 3-stage calibration. In this example, calibration is performed at 0g, 200, and 500g. The method is identical to that of 2-stage calibration, linear function y_1 is calibrated with slope a_1 from 0g-200g, intercept b_1; linear function y_2 is calibrated with slope a_2 from 200g-500g, intercept at b_2. If the measured A/D converted value is less than AD_x, linear function y_1 is used, if the converted value is AD_x or higher, linear function y_2 is used to calculate the physical quantity. This method of linear interpolation performed in sections reduces non-linearity errors. Naturally, the greater the number of calibration points, the lower the occurrence of error due to non-linearity.

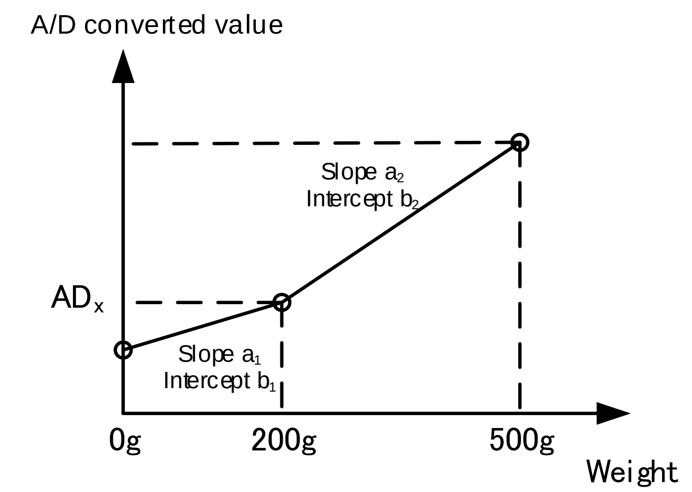


Figure 5-3 3-step Calibration Example

(2) Auto-gain adjustment

The PGA's preamplifier gain has 5 settings (x1, x2, x3, x4, x8) and the post amplifier has 4 settings (x1, x2, x4, x8). Combination of the pre- and post-amplifier settings enables gain settings from x1 to x64. The solution described in this application note uses the full input range of the ΔΣA/D converter, and the optimum gain is automatically set in accordance with the input voltage.

The auto-gain adjustment is a process that adjusts the pre-amplifier gain in order to reach the optimum gain. With a D/A converter connected to the post-amplifier for offset voltage adjustment, when the post-stage gain changes, the gain of the D/A converter output voltage is also multiplied. Therefore, the programmable gain instrumentation amplifier is used with post-stage gain fixed at x8. Auto-gain adjustment is executed every 5 seconds. However, if the retrieved A/D converted value overflows, the gain setting is judged as incorrect, and a process to lower the gain is immediately implemented. Figure 5-4 shows the auto-gain adjustment flow.

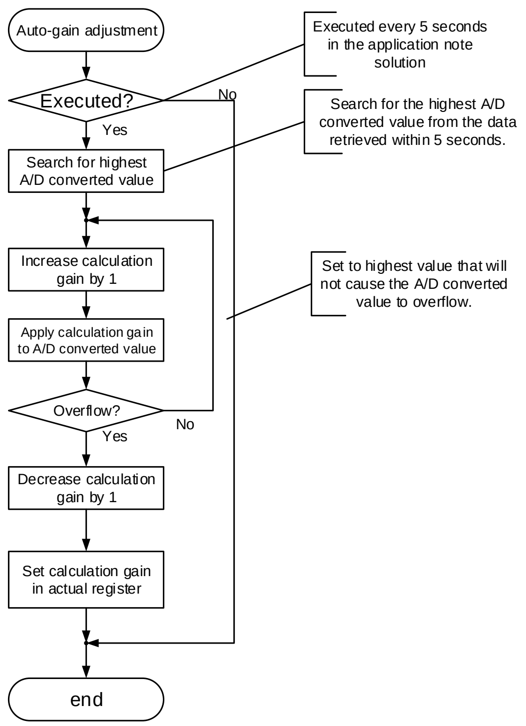


Figure 5-4 Auto-gain Adjustment Execution Flow

(3) Physical quantity conversion

The following is an explanation of how to convert the retrieved A/D converted value into a physical quantity (grams). Physical quantity conversion employs calibration data retrieved through multiple calibrations. The relationship between an A/D converted value and the physical quantity is shown below. In this case, data measured while confirming operations of this application note are used as the pre-converted values.

$$\text{Physical quantity (weight)} = \frac{\text{Retrieved A/D converted value} - \text{intercept}}{\text{slope}}$$

(4) Zero adjustment function

Common scales have as a zero adjustment function that sets the current gram display to 0g. In this application note, the tare weight* on the load cell is subtracted by the program so that the display is set to 0g.

**Tare* weight, or unladen weight, indicates the weight of the container, bag or box that holds the object to be weighed. In this case, tare weight indicates everything on top of the load cell.

(5) Error judgement processing

Error judgement processing judges the following errors: range error when the measurement range of the load cell is exceeded, overflow error when the retrieved A/D converted value overflows, and disconnection error when the load cell and programmable gain instrumentation amplifier are disconnected. The flow for error judgement processing is shown below.

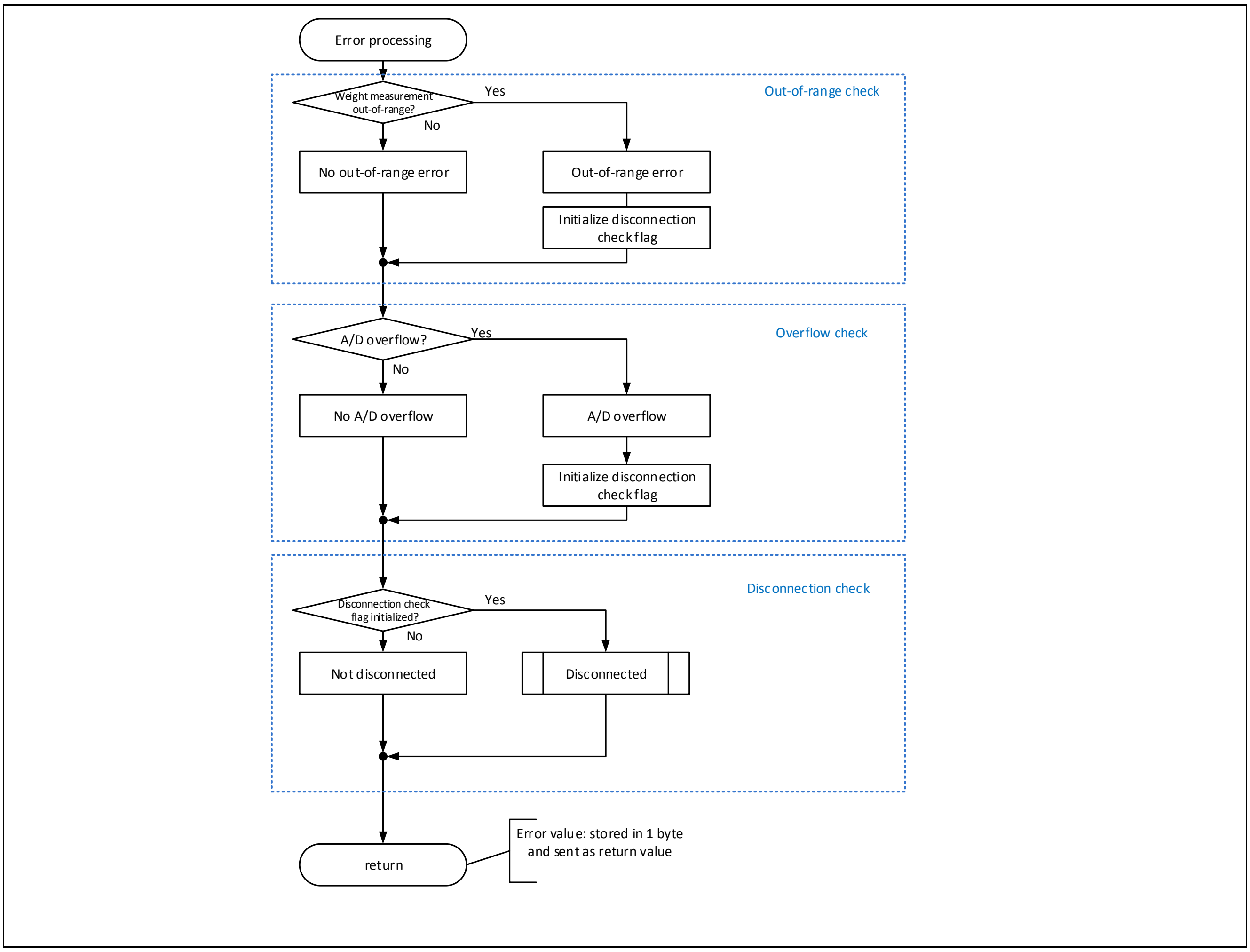


Figure 5-5 Error Judgement Processing

Details concerning disconnection detection are shown in the following figure. This function prevents line downtime by detecting electrical wiring failure between the gauge and RL78/I1E. The RL78/I1E disconnection detection function can detect whether the A/D converted value will overflow or not by connecting a minute current source to the programmable gain instrumentation amplifier input in the IC. In this application note, a disconnection is detected when the results during normal measurement show either a range error or A/D overflow error. Figure 5-6 shows the detection flow.

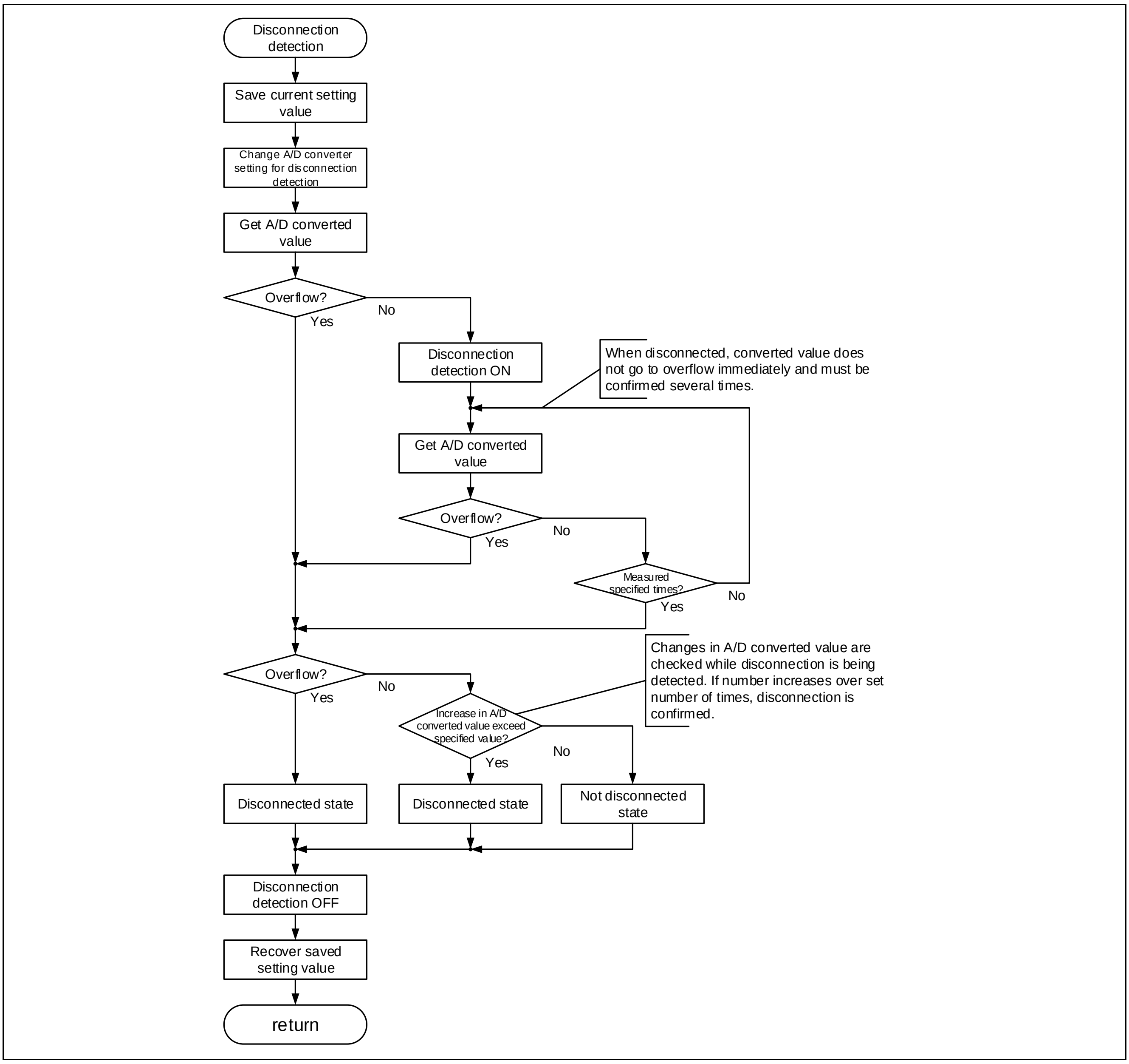


Figure 5-6 Disconnection Detection Flow

5.2 Changes to Analog Characteristics Evaluation Sample Code

The solution described in this application note uses certain parts of the RL78/I1E analog characteristics evaluation sample code. Processes unique to this application note are described in this section. For all other details, please see the RL78/I1E Analog Characteristics Evaluation Sample Code Specification (R01AN2818J) Application Note.

5.2.1 Changes to strain gauge solution

(1) r_scale.c/h file additions

- Weight calculation
  - Weight calculation function (R_SCALE_WeightCalculation)
  - Global variables in file used for weight calculation
    - Slope coefficient storage variable (gs_slope_buf[][])
    - Intercept coefficient storage variable (gs_intercept_buf [][])
    - Coefficient change threshold storage variable (gs_coefficient_buf[][])
- Coefficient calibration
  - Calibration data storage processing function (R_SCALE_InputCalibrationValue)
  - Internal coefficient calculation function (r_scale_coefficient_calculation)
  - Calibration data storage variable (g_calibration_value_buf[][])
  - Calibration value physical quantity storage variable (gs_calibration_point_buf [])
- Zero adjustment
  - Zero adjustment execution function (R_SCALE_ZeroAdjustment)
  - Zero adjustment data storage function (R_SCALE_SetZeroAdjustmentValue)
  - Zero adjustment data storage variable (gs_zero_adjustment_value)
- Weight stability detection processing
  - Weight stability detection processing function (R_SCALE_StabilityCheck)
- Calculation-related global variable initialization processing function (R_SCALE_Init)

(2) main function modifications

- Additional scale control commands

  Add processing to add and support commands required for the scale (zero adjustment, offset calibration, 1st stage calibration, 2nd stage calibration, 3rd stage calibration)
- Add support for scale control commands in receive data analysis function (r_uart_receive_check)
- Add command check function (r_command_check)
- Modifications to communication data generation function
  - Change transmit data as follows:
    - Post-averaging A/D converted value
    - Calculated weight
    - Measurement state
    - Error flag
    - Weight stability state
    - A/D converted value (Bulk transfer)
- Add A/D converted value averaging processing function (r_get_dsad_average)
- Add calibration data read and write to flash storage variable (r_flash_data_decode、r_flash_data_update)
- Add measurement range error processing to error check function (r_error_check)
- Add offset adjustment function
  - Add gain offset adjustment and storage function (r_offset_adjustment)
  - Add gain offset storage variable (g_gain_offset_reg_table[])
- Get offset setting update function (r_offset_change)

(3) Differences in code generation settings

Table 5-1 Code Generation Setting Differences

Peripheral function		RL78/I1E Analog characteristics evaluation sample code setting	RL78/I1E Strain gauge solution sample code setting
PGA+ ΔΣA/D Converter	Auto-scan mode setting	Single scan	Continuous scan
PGA+ ΔΣA/D Converter └ Multiplexer 0	Over-sampling ratio	256	2048

5.2.2 Additional and modified functions from the analog characteristics evaluation sample

(1) r_cg_main.c

Table 5-2 r_cg_main.c File Functions

Function Name	Description
main	Main function
R_MAIN_UserInit	User interface initialization function
r_error_check	Error check function
r_uart_receive_check	Receive data analysis function
r_communication_data_generation	PC communication data generation function
r_get_dsad_average	DSAD value averaging processing function
r_measurement_start	Measurement start processing function
r_measurement_stop	Measurement stop processing function
r_offset_change	Offset setting update function
r_offset_adjustment	PGA offset adjustment function
r_command_check	Command check generation function
r_flash_data_decode	Flash data decode function
r_flash_data_update	Flash data update function

(2) r_scale.c

Table 5-3 r_scale.c File Functions

Function Name	Description
R_SCALE_Init	Initialization of scale-related variable
R_SCALE_SetZeroAdjustmentValue	Zero adjustment data storage function
R_SCALE_ZeroAdjustment	Zero adjustment execution function
R_SCALE_InputCalibrationValue	Calibration data storage processing function
R_SCALE_WeightCalculation	DSAD value to gram calculation, return value in weight storage pointer function
R_SCALE_StabilityCheck	Weight stability check processing function

5.2.3 Macro Declarations for User Environment-dependent Settings

This sample code defines settings dependent on the user environment and usage conditions in macro declarations. Please modify definitions according to the user environment.

(1) r_cg_main.c

Table 5.2.3-1 Macro Declarations for User Environment-dependent Settings

Macro Declaration	Default Setting Value	Input Range	Description
D_INTERMITTENT_DRIVE	0U	0U,1U	Intermittent drive 0: In continuous operations 1: In intermittent operations
D_BULK_NUM	488U: in continuous operations 100U: in intermittent operations	1 or higher	Specified total number of transfer data from BULKSTART command To BULKEND command
D_AVERAGE_BUF_SIZE	244U: in continuous operations 50U: in intermittent operations	1 or higher	Specified number of transfer data for 1 BULK command
D_GRAM_UPDATE_DSAD_NUM	D_AVERAGE_BUF_SIZE	1 or higher	Gram update timing
D_MPX_NUMBER	E_PGA_DSAD_MPX0		Measurement MPX pin
D_STREAMHEADER	"STREAMHEADER:Count,Averaged_dsad_Value,Weight,Status,Error,Stability¥r¥n"		STREAMHEADER string definition
D_BULK_COMMAND_NUM	20U		DSAD division transfer number

(2) r_rl78_i1e_common.h

Table 5.2.3-2 Macro Declarations for User Environment-dependent Settings

Macro Declaration	Default Setting Value	Input Range	Description
D_MCU_CLOCK_MHZ	32U	uint8_t [Note 1]	Defines frequency for CPU and peripheral clocks.[Note1, Note2] This is used to roughly calculate the time of the software wait used in the API. (MHz)
D_FLASH_MEMORY_DATA_USE	1U	0U,1U	Flash memory data use setting 0: Not used 1: Used
D_FLASH_FORCE_WRITING	0U	0U,1U	Flash data forced write setting 0: Do not forcibly overwrite (overwrite only when flash value is invalid) 1: Forcibly overwrite
D_DSAD_CORRECT_USE	1U	0U,1U	PGA differential measurement enable setting 0: PGA differential measurement disabled 1: PGA differential measurement enabled
D_DSAD_CORRECT_MPXn	D_PGA_DSAD_MPX0	MPX0-MPX4[Note3]	Input multiplexer number used for PGA differential measurement[Note 3]
D_DSAD_VALUE_BUFFER_SIZE	256U	uint16_t	Buffer size for DSAD converted value storage
D_DSAD_AUTO_GAIN_USE	1U	0U,1U	PGA auto-gain adjustment enable/disable setting 0: PGA auto-gain adjustment disabled 1: PGA auto-gain adjustment enabled
D_DSAD_AUTO_GAIN_TRIGGER_SEC	5U[Note 4]	uint8_t	PGA Auto-gain adjustment timing (sec)
D_GAIN_ERROR_REFERENCE_mV	10.0F	float	PGA gain error measurement reference voltage (mV)
D_UART_SEND_USE	1U	0U,1U	UART transmission enable setting 0: UART transmission disabled 1: UART transmission enabled
D_UART_SEND_BUFFER_SIZE	256U	Max.256	Transmit buffer size for PC transmission

Note 1: Set a value higher than 0.
Note 2: Specify the setting value for the CPU clock of the MCU you are using.
Note 3: Specify the define declaration value of the input multiplexer number
D_PGA_DSAD_MPX0 = input multiplexer 0
D_PGA_DSAD_MPX1 = input multiplexer 1
D_PGA_DSAD_MPX2 = input multiplexer 2
D_PGA_DSAD_MPX3 = input multiplexer 3

(3) r_keyscan.h

Table 5.2.3-3 Macro Declarations for User Environment-dependent Settings

Macro Declaration	Default Setting Value	Input Range	Description
DEF_KEY_ACTIVE	DEF_KEY_ACTIVE_LOW	DEF_KEY_ACTIVE_LOW, DEF_KEY_ACTIVE_HIGH	SW active level
KEY_SCAN_NORM	10U	uint8_t	Single push determination time (10ms)
KEY_SCAN_LONG	100U	uint8_t	Long push determination time (10ms)
KEY_SCAN_DEAD	5U	uint8_t	Dead zone time after change in key state (10ms)
KEY_SCAN_NOT	5U	uint8_t	Non-active time (10ms)
KEY_SCAN_DOUBLE	50U	uint8_t	Double click determination enable time (10ms)

(4) r_cg_userdefine.h

Table 5-4 Macro Declarations for User Environment-dependent Settings

Macro Declaration	Default Setting Value	Input Range	Description
D_DEBUG_LED_USE	0U	0U,1U	Debug LED usage setting 0U : Not used 1U : Used
D_DEBUG_LED_PORT	P1.5	(output port)[Note1]	Debug LED port setting

Note 1: Set the digital output port connected to the pulled-up LED.

(5) r_scale.h

Table 5.2.3-5 Macro Declarations for User Environment-dependent Settings

Macro Declaration	Default Setting Value	Description
D_WEIGHT_MIN	-600	Minimum measurement (gram)
D_WEIGHT_MAX	600	Maximum measurement (gram)
D_COEFFICIENT_NUMBER	5U	Number of coefficients
D_CALIBRATION_POINTS	3U	Number of calibration points (2 or more)
D_COEFFICIENT_POINTS	D_CALIBRATION_POINTS -1U	Number of coefficient points
D_COEFFICIENT_CHANGE_POINTS	D_COEFFICIENT_POINTS -1U	Number of coefficient change points
D_CALIBRATION_1_GRAM	0	Weight of 1st calibration point (g)
D_CALIBRATION_2_GRAM	200	Weight of 2nd calibration point (g)
D_CALIBRATION_3_GRAM	500	Weight of 3rd calibration point (g)
D_STABILITY_CHECK_NUM	3U	Number of stability checks (no. of times)

D_SLOPE_1_DEFAULT_VALUE	1976.6	Slope 1 default value (LSB/g)
D_SLOPE_2_DEFAULT_VALUE	D_SLOPE_1_DEFAULT_VALUE	Slope 2 default value (LSB/g)
D_INTERCEPT_1_DEFAULT_VALUE	2677181.7	Intercept 1 default value (LSB)
D_INTERCEPT_2_DEFAULT_VALUE	D_INTERCEPT_1_DEFAULT_VALUE	Intercept 2 default value (LSB)
D_COEFFICIENT_CHANGE_1_VALUE	683223	Coefficient change value 1 default value (LSB)

5.2.4 Macro Declarations

This section describes the macro declarations defined in r_cg_main.c.

Table 5.2.4-1 Macro Declarations

Macro Declaration	Default Setting Value	Description
D_BULK_SPLIT_NUM	D_BULK_NUM / D_DSAD_VALUE_BUF_SIZE	BULK division number
D_BULK_SPLIT_MOD	D_BULK_NUM % D_DSAD_VALUE_BUF_SIZE	BULK division remainder
D_NO_ERROR	0x00U	No error
D_OVER_RANGE_ERROR	0x01U	Range error
D_OVER_FLOW_ERROR	0x02U	Overflow error
D_DIVIDE_BY_ZERO_ERROR	0x04U	Divide-by-zero error
D_DISCONNECTION_ERROR	0x80U	Disconnect error
D_ON_OFF_COMMAND	@0¥r¥n	ON/OFF command
D_ZERO_ADJUSTMENT_COMMAND	@1¥r¥n	Zero adjustment command
D_OFFSET_ADJUSTMENT_COMMAND	@2¥r¥n	Offset adjustment command
D_CALIBRATION_1_COMMAND	@3¥r¥n	Calibration 1 command
D_CALIBRATION_2_COMMAND	@4¥r¥n	Calibration 2 command
D_CALIBRATION_3_COMMAND	@5¥r¥n	Calibration 3 command
D_COMMAND_SIZE	4U	Receive command size for PC communication

5.2.5 Enumerations

This section describes the enumerations defined in r_cg_main.c.

Table 5-5 BULK Format Data Generation Control Enumeration

Type Name	Macro Name	Description
e_bulk_control_t	E_BULKDATA_BULKSTART	BULKSTART format data generation
	E_BULKDATA_BULK	BULK format data generation
	E_BULKDATA_BULKEND	BULKEND format data generation

Table 5-6 Commnication Command Generation Control Enumeration

Type Name	Macro Name	Description
e_communication_data_t	E_STREAMHEADER	STREAMHEADER format data generation
	E_STREAM	STREAM format data generation
	E_BULK	BULK format data generation
	E_BINARY	BINARY format data generation

Table 5-7 Commnication Command Judgement Enumeration

Type Name	Macro Name	Description
e_communication_command_t	E_ON_OFF_COMMAND	ON/OFF command
	E_ZERO_ADJUSTMENT_COMMAND	Zero adjustment command
	E_OFFSET_ADJUSTMENT_COMMAND	Offset calibration command
	E_CALIBRATION_1_COMMAND	Calibration 1 command
	E_CALIBRATION_2_COMMAND	Calibration 2 command
	E_CALIBRATION_3_COMMAND	Calibration 3 command
	E_COMMAND_NONE	Determines no command is present

Table 5-8 Measurement State Communication Enumeration

Type Name	Macro Name	Description
e_measurement_state_t	E_DEFAULT	Uncalibrated, PGA next-stage offset adjustment enabled state
	E_OFFSET_ADJ_1	PGA next-stage offset adjustment executed, calibration 1 enabled state
	E_CALIBRATION_1	calibration 1 executed, calibration 2 enabled state
	E_CALIBRATION_2	calibration 2 executed, calibration 3 enabled state
	E_CALIBRATION_3	calibration 3 executed, normal measurement state

Table 5-9 MCU State Determination Enumeration

Type Name	Macro Name	Description
e_mcu_state_t	E_MCU_HALT	HALT mode
	E_MCU_RUN	Now running

Table 5-10 Measurement Control Enumeration

Type Name	Macro Name	Description
e_measurement_control_t	E_TRANSMISSION_START	Serial transmission start
	E_DATA_GENERATION	Weight calculation
	E_ERROR_CHECK	Error check
	E_WAIT	No operation

Table 5-11 Divide-by-zero State Judgement Enumeration

Type Name	Macro Name	Description
divide_by_zero_error_t	E_DIVIDE_OK	Divide-by-zero normal completion
	E_DIVIDE_BY_ZERO_ERROR	Divide-by-zero generated

5.2.6 Structures

This section describes the structure declarations defined in r_cg_main.c.

Table 5-12 Structure for Variables Storing Measured Data

Structure Name	measurement_data_t		
Outline	Definition of variables storing measured data		
Member Variable	Type	Name	Description
	uint32_t	count	Transmit counter value storage variable
	str_pga_dsad_value_t	dsad_value	ΔΣA/D converted value structure variable
	int32_t	averaged_dsad_value	Averaged DSAD value storage variable
	float	weight	Gram converted value storage variable
	e_mcu_state_t	mcu_state	MCU operating state storage variable
	e_measurement_state_t	measurement_state	Measurement operating state storage variable
	uint8_t	error_state	Error state storage variable
	stability_check_t	stability_check_flag	Stability state flag storage variable

Table 5-13 Structure for Variables Storing BULK Transfer Data

Structure Name	bulk_data_t		
Outline	Definition of variables storing bulk transfer data		
Member Variable	Type	Name	Description
	uint16_t	send_count	Bulk transfer send counter
	e_bulk_control_t	control	Bulk transfer control
	int32_t	buf[D_BULK_BUFFER_SIZE]	BULK transfer data buffer
	uint16_t	str_count	BULK transfer data buffer counter
	uint16_t	split_count	Bulk transfer split send counter

Table 5-14 Structure for Variables Storing Gain Setting Value and Corresponding Offset Setting Value

Structure Name	gain_offset_reg_t		
Outline	Structure for variables storing gain setting value and corresponding offset setting value		
Member Variable	Type	Name	Description
	e_pga_dsad_gain_t	dsad_gain	PGA gain setting specification enumeration variable
	e_pga_dsad_offset_t	dsad_offset	DC offset voltage setting specification enumeration variable

5.2.7 Global Variables

(a) r_cg_main.c

Table 5-15 Global Variables

Type Name	Global Constant Name	Description
int32_t	g_dsad_average_buf[]	DSAD averaged value calculation data storage buffer
uint16_t	g_dsad_get_count	Number of times DSAD value is retrieved
gain_offset_reg_t	g_gain_offset_reg_table[]	Gain change register value storage variable
uint8_t	g_gain_offset_reg_table_size	Gain change register value storage variable size
uint8_t	g_gain_offset_reg_table_index	Gain change register value storage variable index
bulk_data_t	g_bulk_data	Bulk transfer data storage variable

(b) r_scale.c

Table 5-16 Global Variables

Type Name	Global Constant Name	Description
int32_t	g_calibration_value_buf[][]	Calibration measurement value storage variable

Table 5-17 Internal Global Variables

Type Name	Global Constant Name	Description
float	gs_slope_buf[][]	Slope storage variable
float	gs_intercept_buf[][]	Slope storage variable
float	gs_calibration_point_buf[]	Calibration value physical quantity storage variable
float	gs_coefficient_buf[]	Coefficient change threshold storage variable
uint8_t	gs_calibration_value_buf_index[]	Calibration measurement value storage variable index
uint8_t	gs_flash_data_set_flag	Set flash data initialization flag
float	gs_zero_adjustment_value	Zero adjustment data storage variable

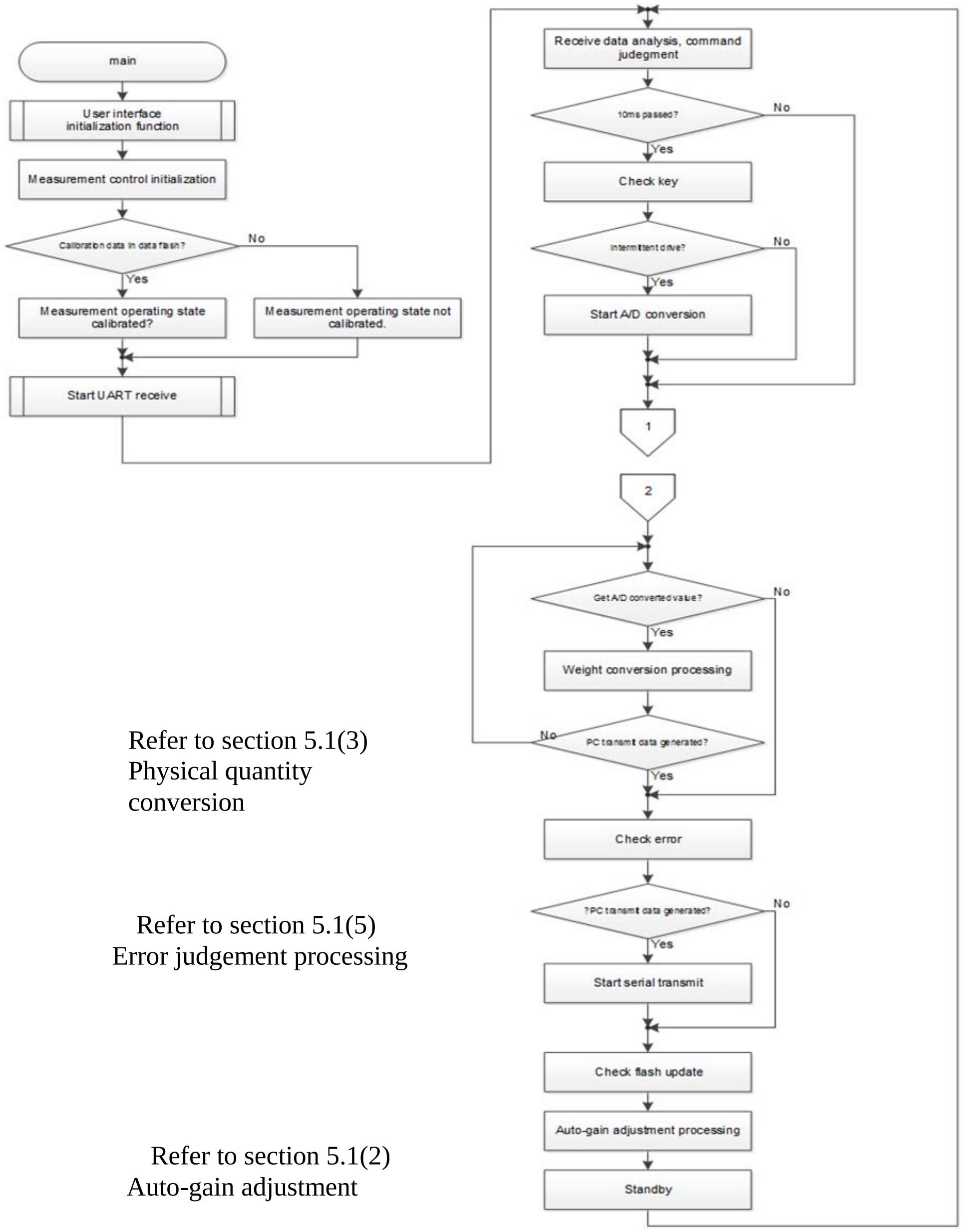
5.3 Function Specifications

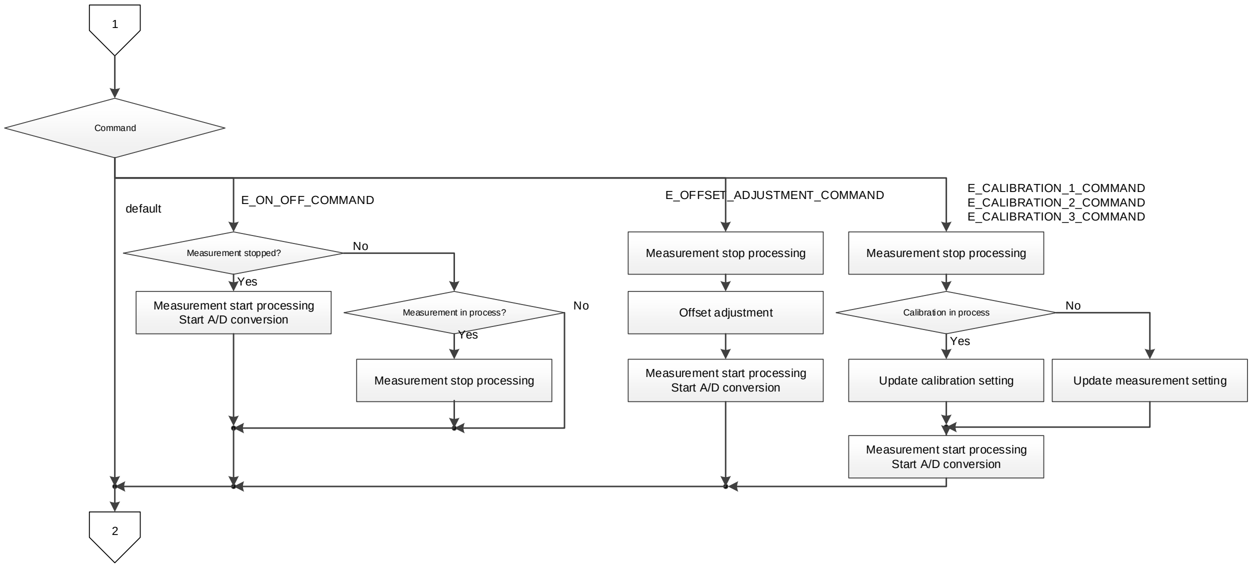
5.3.1 r_cg_main.c

(1) main function

void R_MAIN_UserInit(void)	
Description	main function
Argument	None
Return Value	None
Processing	See flowchart for details.

Flowchart





(2) User interface initialization function

void main(void)

Description	User interface initialization function
Argument	None
Return Value	None
Processing	• Call R_I1E_Variable_Initialize function to execute initialization processing of variables used in sample code. • If flash memory data is used in the sample code, read data (default is “use data”). • If forced write of flash memory data is enabled, write data (default is “no forced write”). • Call R_KEY_Initialize function to initialize key information. • Call R_INTC0_Start function to enable INTP0 interrupt. • Set g_tx_end_flag to 0, call R_UART1_Start function and start UART1. • Call R_SCALE_Init function to initialize scale variable.

(3) Error check function

static uint8_t r_error_check(measurement_data_t * const p_measurement_data)

Description	ΔΣA/D conversion result error check function
Argument	p_measurement_data: Measured data storage variable pointer
Global Variables	None
SFR	None
Return Value	uint8_t: Error information
Processing	• If storage variable of gram value converted from measured data storage variable is out of weight measurement range or ΔΣA/D converted value structure variable overflow flag is 1, initialize the error information corresponding to the return value storage variable, and execute the disconnect detection processing. • If in disconnect state, initialize the corresponding error information and send the return value.

(4) Receive data analysis function

```
static e_communication_command_t r_uart_receive_check(int8_t * p_receive_buf)
```

Description	Receive data analysis function
Argument	* p_receve_buf: Receive data buffer pointer
Global Variables	None
SFR	None
Return Value	e_communication_command_t: Receive command ENUM definition E_ON_OFF_COMMAND ON/OFF command: E_ZERO_ADJUSTMENT_COMMAND Zero adjustment command E_OFFSET_ADJUSTMENT_COMMAND PGA offset adjustment command: E_CALIBRATION_1_COMMAND Calibration 1 command E_CALIBRATION_2_COMMAND Calibration 2 command: E_CALIBRATION_3_COMMAND Calibration 3 command E_COMMAND_NONE No command
Processing	Compare the receive data buffer data and the defined receive command string. If they match, return the corresponding ENUM; if not, return E_COMMAND_NONE.

(5) PC software communication data generation function

```
static uint8_t r_communication_data_generation
  (e_communication_data_t communication_data, measurement_data_t * const p_measurement_data)
```

Description	PC software communication data generation function
Argument	communication_data: Communication data *p_measurement_data: Measured data storage variable pointer
Global Variables	g_i1e_uart_send_buffer[][]: PC communication transmit buffer g_buffer_number: Buffer number of PC communication transmit buffer
SFR	None
Return Value	uint8_t: 0 No data generated 1 Data generation completed
Processing	• Convert data from measured data storage pointer and store in PC buffer based on the format specified in the communication data • Send “no data generated” or “data generation completed” in return value.

(6) DSAD value averaging processing function

static int32_t r_get_dsad_average(int32_t * p_dsad_buf, uint8_t dsad_num)	
Description	DSAD value averaging processing function
Argument	*p_dsad_buf: DSAD value storage buffer dsad_num: Number of data to be averaged
Global Variables	None
SFR	None
Return Value	int32_t: DSAD averaged value
Processing	Average the number of data specified in the DSAD value storage buffer and send in return value.

(7) Measurement start processing function

static void r_measurement_start(void)	
Description	Measurement start processing function
Argument	None
Global Variables	gs_dsad_autoscan_mode: Auto-scan mode storage variable g_dsad_get_count: Number of times DSAD value is retrieved g_bulk_data: Bulk transfer data storage variable
SFR	TMIF10: TAU channel 10 count complete/capture complete interrupt request flag TMMK10: TAU channel 10 count complete/capture complete interrupt mask flag TS1: Timer channel start register 1 DSADIF: 24-bit ΔΣA/D conversion complete interrupt request flag DSADMK: 24-bit ΔΣA/D conversion complete interrupt mask flag DSADST: A/D conversion (AUTOSCAN) control
Return Value	None
Processing	• Call R_I1E_RingBuffer_Initialize function to initialize DSAD ring buffer. • Set g_dsad_get_count to 0, initialize DSAD averaging buffer • Assign E_BULKDATA_BULKSTART to g_bulk_data.control and set g_bulk_data. str_count to 0, then initialize bulk transfer control. • Call R_TAU1_Channel0_Start function and start SW timer. • In intermittent operations, call R_PGA_DSAD_Start function and start DSAD conversion.

(8) Measurement stop processing function

static void r_measurement_stop(void)

Description	Measurement stop processing function
Argument	None
Global Variables	None
SFR	TMIF10: TAU channel 10 count complete/capture complete interrupt request flag TMMK10: TAU channel 10 count complete/capture complete interrupt mask flag TS1: Timer channel start register 1 DSADIF: 24-bit ΔΣA/D conversion complete interrupt request flag DSADMK: 24-bit ΔΣA/D conversion complete interrupt mask flag DSADST: A/D conversion (AUTOSCAN) control
Return Value	None
Processing	• Call R_PGA_DSAD_Stop function and stop A/D conversion (AUTOSCAN). • Call R_TAU1_Channel0_Stop function and stop SW timer.

(9) Gain setting-related offset setting update function

static void r_offset_change(e_pga_dsad_mpx_t dsad_mpx)

Description	Gain setting-related offset setting update function
Argument	dsad_mpx: 24-bit ΔΣA/D converter input multiplexer number
Global Variables	g_dsad_setting[]: input multiplexer settings g_gain_offset_reg_table[]: Gain change register value storage variable g_gain_offset_reg_table_index: Gain change register value storage variable index
SFR	PGAxCTL1: Input multiplexer x (x = 0 to 4) setting register 1
Return Value	None
Processing	• Obtain the corresponding offset value of the input multiplexer specified in dsad_mpx from the current gain value stored in g_dsad_setting[].dsad_gain • Assign g_dsad_setting[].dsad_offset as the value of g_gain_offset_reg_table[].dsad_offset. • At this time, assign g_gain_offset_reg_table_index as the index for g_gain_offset_reg_table[]. • Call R_I1E_PGA_DSAD_OffsetRegSet function, and write the offset setting value to the PGAxCTL1 register.

(10) Flash data decode function

static void r_flash_data_decode(void)

Description	Flash data decode function
Argument	None
Global Variables	g_flash_value: Data flash storage data structure g_gain_offset_reg_table[]: Gain change register value storage variable g_gain_offset_reg_table_size: Gain change register value storage variable size g_calibration_value_buf[][]: Calibration measurement value storage variable
SFR	None
Return Value	None
Processing	• Copy g_flash_value. offset_correct[] to g_gain_offset_reg_table[].dsad_offset • Copy g_flash_value.calibration_value[][] to g_calibration_value_buf[][]

(11) Command check generation function

static e_communication_command_t r_command_check
(measurement_data_t measurement_data, e_communication_command_t command)

Description	Command check generation function
Argument	measurement_data: Measurement data storage variable command: Receive command
Global Variables	None
SFR	None
Return Value	e_communication_command_t: Receive command ENUM definition E_ON_OFF_COMMAND ON/OFF command E_ZERO_ADJUSTMENT_COMMAND Zero adjustment command E_OFFSET_ADJUSTMENT_COMMAND PGA offset adjustment command E_CALIBRATION_1_COMMAND Calibration 1 command E_CALIBRATION_2_COMMAND Calibration 2 command E_CALIBRATION_3_COMMAND Calibration 3 command E_COMMAND_NONE No command
Processing	• Determine whether to enable or disable a receive operation based on the MCU operating state storage variable in the measured data storage variable and the receive command. • If the receive operation is enabled, return the receive command as is; if disabled, return E_COMMAND_NONE.

(12) Flash data decode function

static void r_flash_data_decode(void)	
Description	Flash data decode function
Argument	None
Global Variables	g_flash_value: Data flash storage data structure g_gain_offset_reg_table[]: Gain change register value storage variable g_gain_offset_reg_table_size: Gain change register value storage variable size g_calibration_value_buf[][]: Calibration measurement value storage variable
SFR	None
Return Value	None
Processing	• Copy g_flash_value. offset_correct[] to g_gain_offset_reg_table[].dsad_offset • Copy g_flash_value.calibration_value[][] to g_calibration_value_buf[][]

(13) Flash data update function

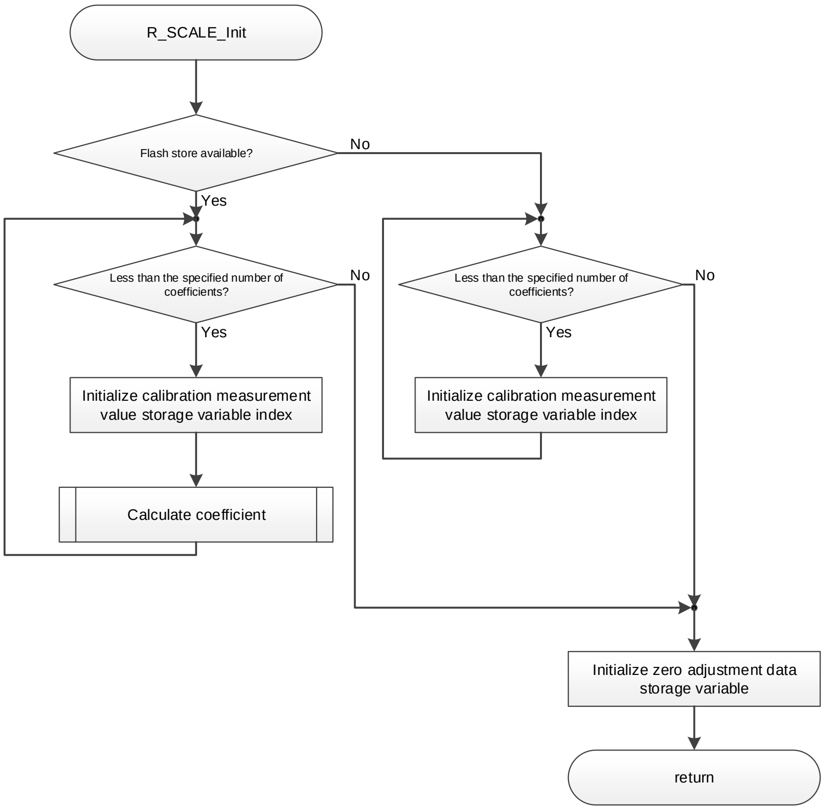
static void r_flash_data_update(void)	
Description	Flash data update function
Argument	None
Global Variables	g_flash_value: Data flash storage data structure g_gain_offset_reg_table: Data flash storage data structure g_gain_offset_reg_table_size: Gain change register value storage variable size g_calibration_value_buf: Calibration measurement value storage variable
SFR	None
Return Value	None
Processing	• Copy g_gain_offset_reg_table[].dsad_offset to g_flash_value. offset_correct[] • Copy g_calibration_value_buf[][] to g_flash_value.calibration_value[][]

5.3.2 r_scale.c

(1) Scale usage variable initialization

void R_SCALE_Init (uint8_t calibration_data_set_flag)

Description	Scale usage variable initialization
Argument	calibration_data_set_flag: Calibration data flag 0: No calibration data 1: Calibration data available
Global Variables	gs_calibration_value_buf_index: Calibration measurement value storage variable index gs_zero_adjustment_value: Zero adjustment data storage variable
SFR	None
Return Value	None
Processing	



(2) Zero adjustment data storage function

void R_SCALE_SetZeroAdjustmentValue(float weight)	
Description	Zero adjustment data storage function
Argument	weight: Weight
Global Variables	gs_zero_adjustment_value: Zero adjustment data storage variable
SFR	None
Return Value	None
Processing	Set Weight value in gs_zero_adjustment_value

(3) Zero adjustment execution function

void R_SCALE_ZeroAdjustment(float * p_weight)	
Description	Zero adjustment execution function
Argument	p_weight: Weight storage pointer
Global Variables	gs_zero_adjustment_value: Zero adjustment data storage variable
SFR	None
Return Value	None
Processing	Retrieve gs_zero_adjustment_value from p_weight pointer storage data.

(4) Calibration data storage processing function

MD_STATUS R_SCALE_InputCalibrationValue
(uint8_t coefficient_num, int32_t dsad_value, calibration_state_t state)

Description: Calibration data storage processing function

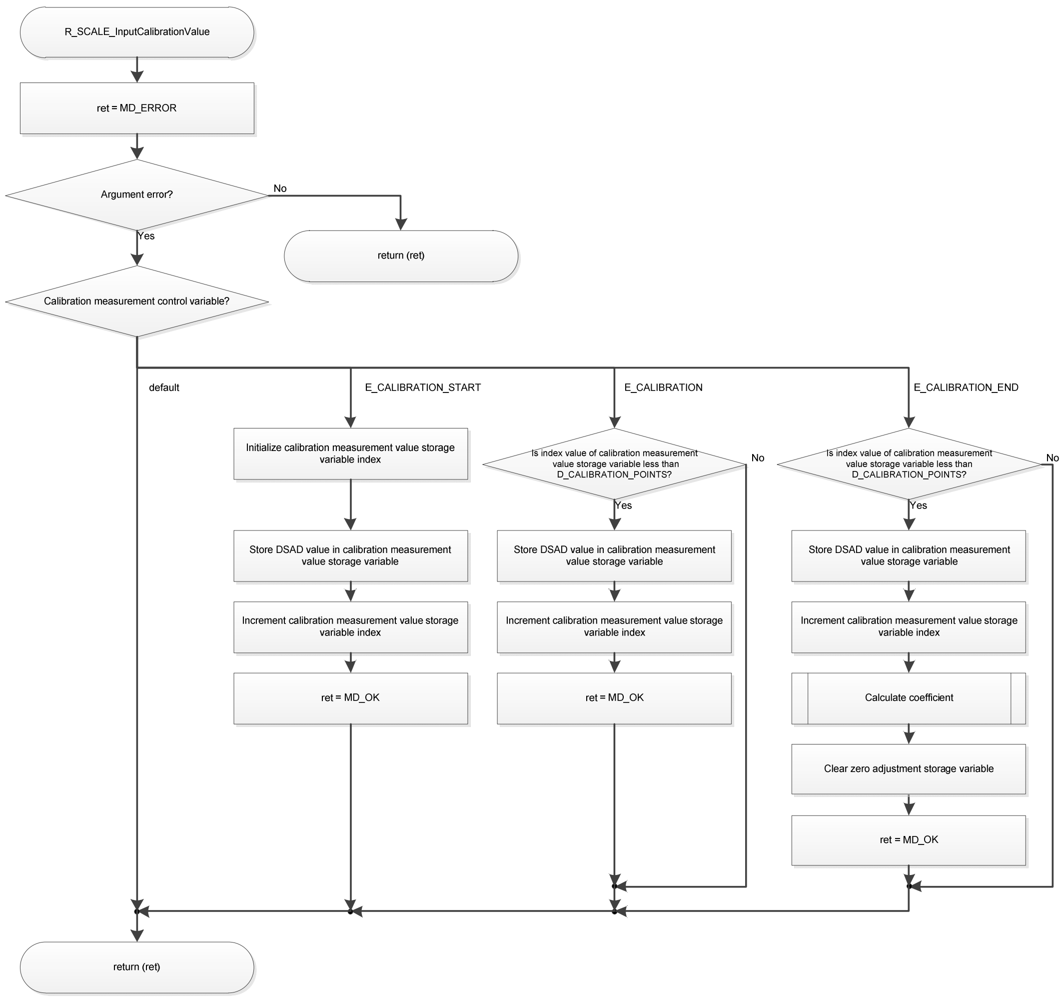
Argument:
coefficient_num:
Coefficient number value
dsad_value:
DSAD value
state:
Calibration measurement control variable

Global Variables:
g_calibration_value_buf[][]:
Calibration measurement value storage variable
gs_calibration_value_buf_index[]:
Calibration measurement value storage variable index
gs_zero_adjustment_value:
Zero adjustment data storage variable

SFR: None

Return Value:
MD_STATUS:
MD_OK
MD_ERROR

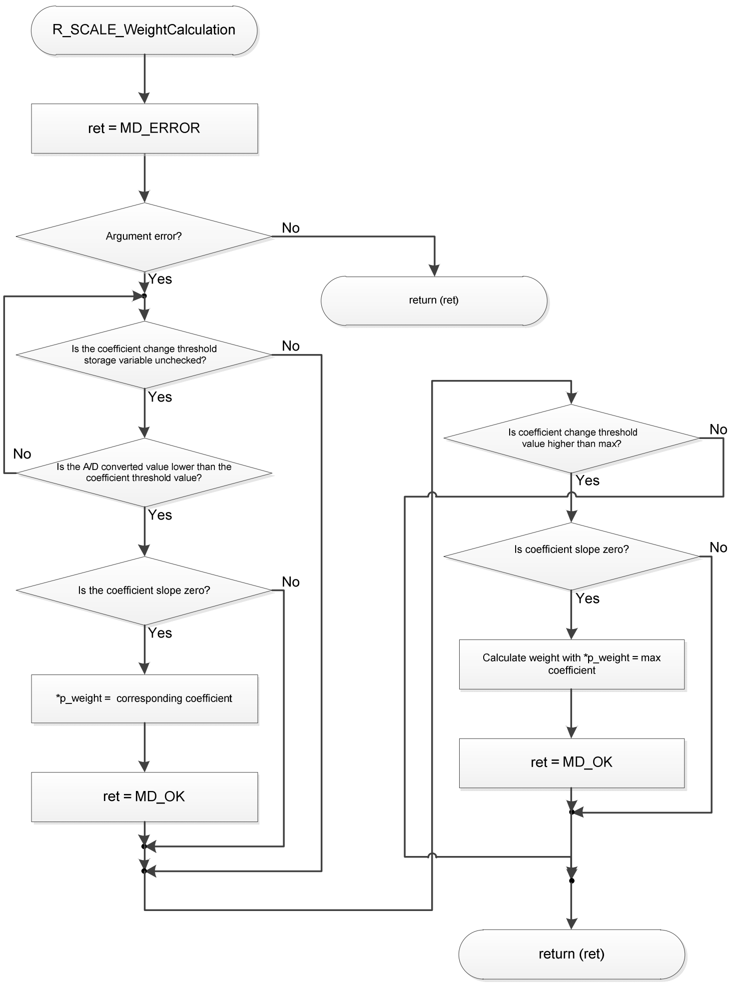
Processing



(5) DSAD value gram calculation, return value in weight storage pointer function

MD_STATUS R_SCALE_WeightCalculation
(uint8_t coefficient_num, int32_t dsad_value, float * p_weight)

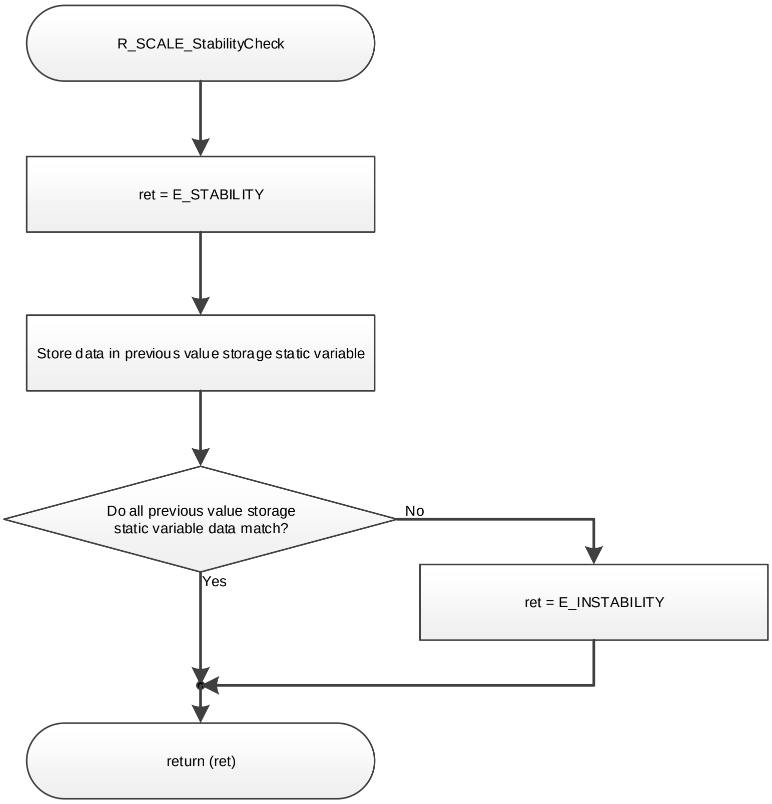
Description	DSAD value gram calculation, value return in weight storage pointer function
Argument	coefficient_num: Number of coefficients dsad_value: DSAD value p_weight: Weight storage pointer
Global Variables	gs_slope_buf[][]: Slope coefficient storage variable gs_intercept_buf[][]: Intercept coefficient storage variable gs_coefficient_buf[][]: Coefficient change threshold storage variable
SFR	None
Return Value	MD_STATUS: MD_OK MD_ERROR
Processing	



(6) Weight stability detection processing function

stability_check_t R_SCALE_StabilityCheck (float weight)

Description	Weight stability detection processing function
Argument	weight: Weight
Global Variables	None
SFR	None
Return Value	stability_check_t: E_INSTABILITY unstable state E_STABILITY stable state
Processing	



6. Measurement Example

Figure 6-1 shows the measurement example as described in this application note, using screen displays from the Strain Gauge PC Application Software. For details on how to use this software, please refer to **RL78/I1E Strain Gauge PC Application Software Manual (R01AN2822J) Application Note**.

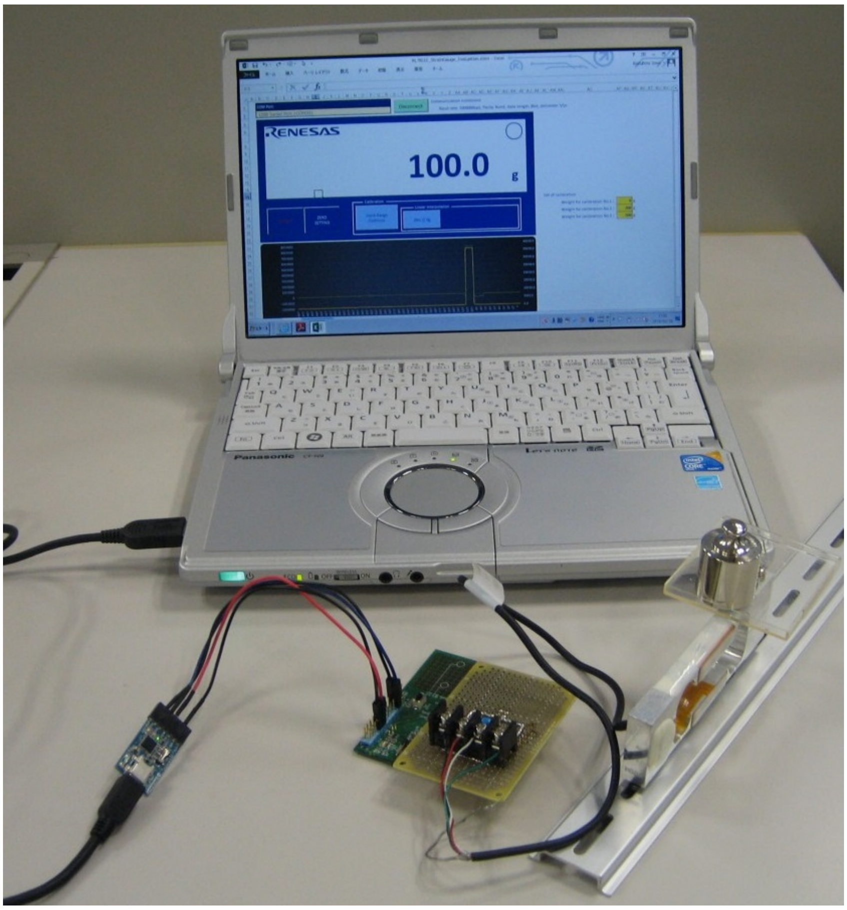


Figure 6-1 Measurement Example

The Strain Gauge PC Application Software displays the current physical quantity (grams), the filtered A/D converted value, and the pre-filtered A/D converted value, as shown in Figure 6-2 from top to bottom.

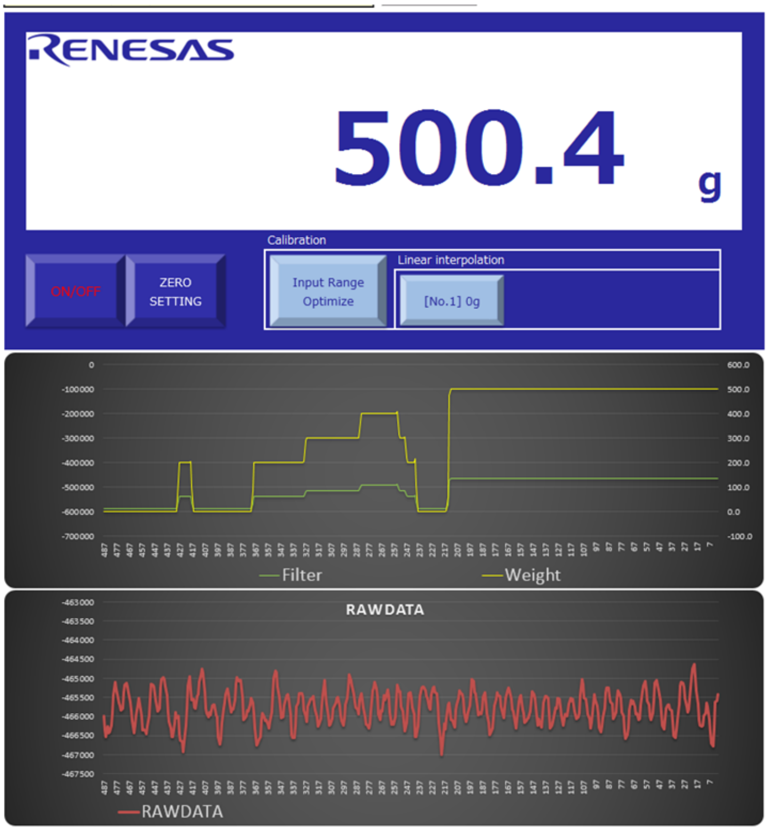


Figure 6-2 Measurement Data

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov. 09, 2015	---	First edition issued
1.01	Mar. 01, 2016	3	R01AN2818J -> R01AN2819J

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. 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 Microprocessing unit or Microcontroller unit products 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.

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

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

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

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