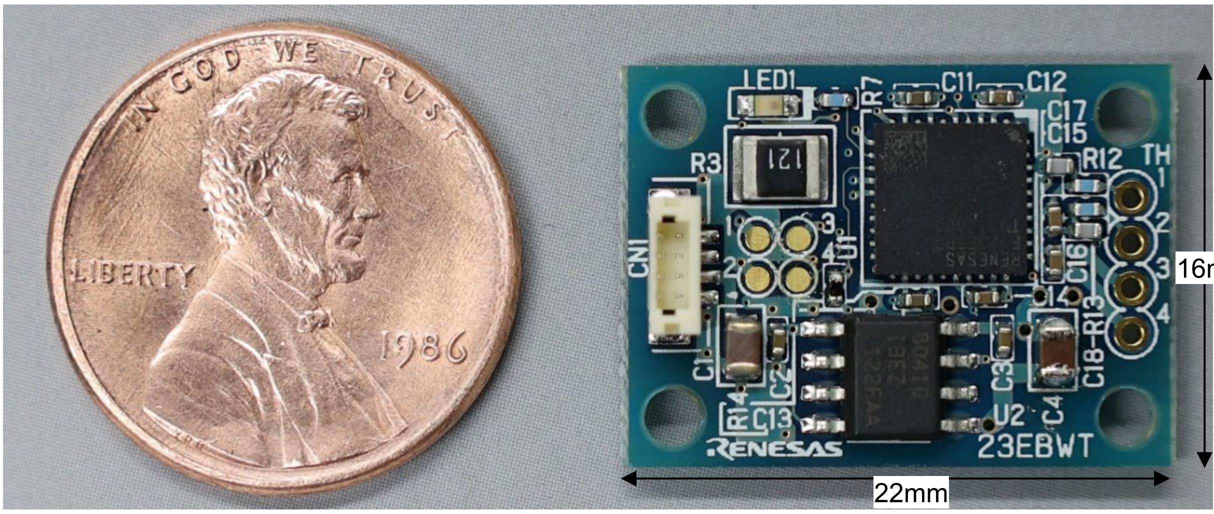
RX23E-B Group

Design and weight measurement of tiny board for digital load cell

Introduction

This document describes a weight measurement program example using the RX23E-B-QFN40-WT, which is a board for a digital load cell using Renesas MCU, RX23E-B, and a load cell.

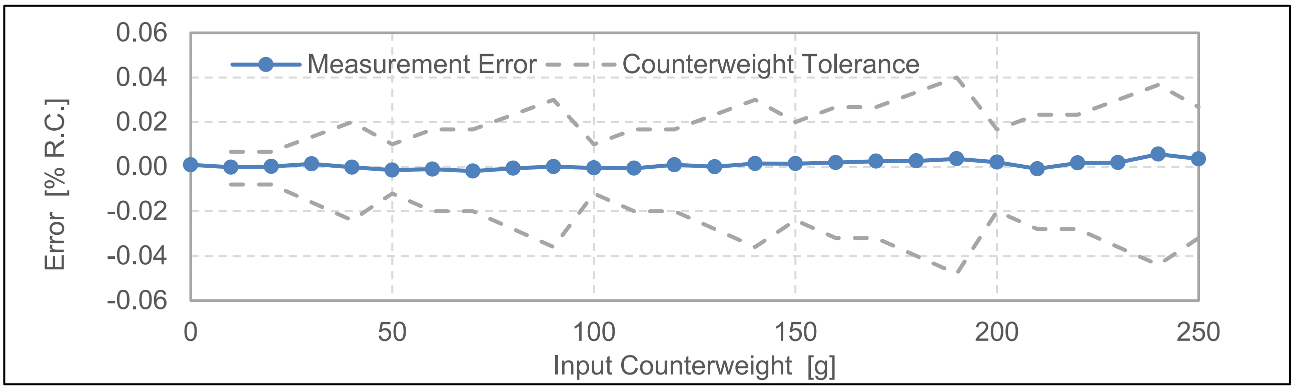
The RX23E-B-QFN40-WT is miniaturized to be incorporated into a digital load cell, using RX23E-B, MCU with AFE in 40 pin HWQFN package, LDO ISL80410 for power supply, and RAA788155 as RS-485 driver. Measurement results can be checked with QE for AFE or the Modbus RTU host program via RS-485, using the sample program running on this board.



Weight of counterweight was measured with a load cell, using this board and the sample program. The error of measured value was divided by load cell's rated capacity of 300g. The results are shown in figure below.

Rated Capacity (R. C.):	300g
Rated Output (R. O.):	0.9 mV/V ±0.1 mV/V
Non-linearity[Note]:	0.015%R. O. or less
Equivalent input noise:	23.8nVrms (21.6 Bits): equivalent to1.58mgrms
Peak-to-peak noise:	140nV (19.1 Bits): equivalent to 9.3mg

Note: Including counterweight error, non-linearity characteristic of load cell etc.



Device

RX23E-B (R5F523E6MDNF)

Contents

1. Overview

This document describes a weight measurement example using the RX23E-B-QFN40-WT, which is a load cell tiny board containing RX23E-B, and a load cell. The sample program conducts weight measurement with a load cell, communicates with QE for AFE or Modbus host via an RS-485 half-duplex communication channel, and transmits measurement results.

A diagram of the weight measurement system in this example is shown in Figure 1-1.

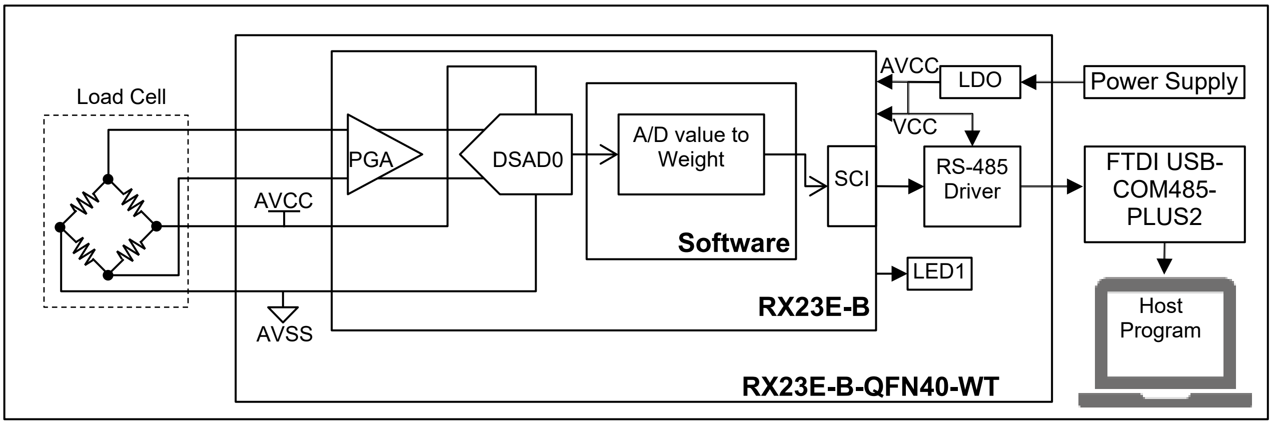


Figure 1-1 Weight Measurement System Example Using a Load Cell

The QE for AFE version sample program in this example uses the Application tab screen of QE for AFE to make various settings, conduct measurements, and display measurement results. Some registers can be set on the AFE connection screen. Operable items are listed in Figure 1-2, Table 1-1, and Table 1-2.

The Modbus version sample program performs the same operations by making settings in Modbus Coil or Holding register listed in Table 7-6. For details, refer to "7.2.3 Operation".

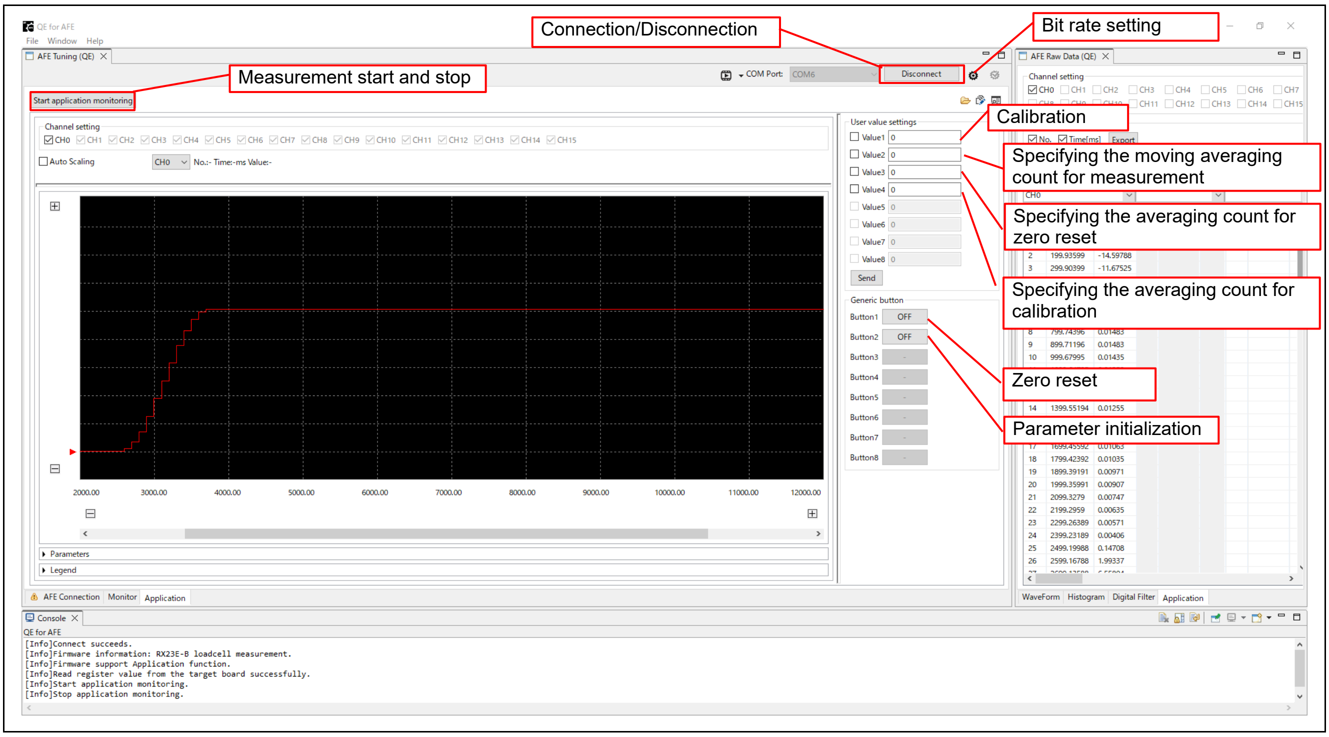


Figure 1-2 QE for AFE Application Tab Screen

Table 1-1 QE for AFE Operation Settings

Item	Operation	Supplement
Connection	Connection/disconnection button	
Measurement start and stop	Measurement start and stop button	LED1 OFF during measurement
Zero reset	Button1	Enabled only during measurement
Calibration	Specify measured weight 1 in Value1 Specify measured weight 2 in Value1 while LED1 is blinking	Enabled while measurement is stopped LED1 OFF during A/D conversion LED1 blinks five times in case of abnormal termination. refer to "6.3Calibration".
Specifying the moving averaging count for measurement	Value2: 1 to 128, default: 8	Enabled only during standby (LED1 ON)
Specifying the averaging count for zero reset	Value3: 1 to 512, default: 256	
Specifying the averaging count for calibration	Value4: 64 to 512, default: 256	
Parameter initialization	Button2	

Note: Set the communication rate for QE for AFE to the communication rate in "Table 7-1 Communication Conditions".
Since QE for AFE is based on full-duplex communication, transmission and reception may conflict and stop on stop measurement. If it stops, turn on the power to the board again.

Table 1-2 QE for AFE Register Setting Changeable Items

Item	Operation	Supplement
PGA gain	Change the PGA gain setting for CH0	
Oversampling ratio	Change the OSR1 and OSR2 for CH0	The lower limit of oversampling ratio setting is OSR1=256 and OSR2=2. If it is lower than this, communication error will occur.
Digital filter selection	Change the Sinc Filter for CH0	
Digital filter gain correction	Change the SGCR for CH0	QE for AFE calculates according to the values of OSR0 and OSR1.

Parameters listed in Table 1-3 maintain their changes using the E2 data flash. For details, refer to structure st_prm_t in Table 8-28.

Table 1-3 Retention Parameters

Item	Number of items/sets that can be stored
Weight conversion coefficients	1 set
Moving averaging count for measurement	1
Averaging count for zero reset	1
CR0.GAIN register value	1
MR0.FSEL register value	1
OSR0 register value	1
SGCR0 register value	1
Averaging count for calibration	1

2. Package Contents

Table 2-1Package Contents

File/folder name	Description
r01an6512jj0100-rx23e-b.pdf	This document (Japanese)
r01an6512ej0100-rx23e-b.pdf	This document (English)
BoardData	Board data of RX23E-B-QFN40-WT
rx23eb_loadcell_qe_rssk.mot	QE for AFE version program binary file for RSSKRX23E-B
rx23eb_loadcell_qe	QE for AFE version sample project set
rx23eb_loadcell_modbus	Modbus version sample project set
readme_j.txt	Package explanation (Japanese)
readme_e.txt	Package explanation (English)

3. Environment for Operation Confirmation

The environment for operation confirmation is shown in Table 3-1.

Table 3-1 Environment for Operation Confirmation

Item		Description
Board		RX23E-B-QFN40-WT
MCU		RX23E-B (R5F523E6MDNF) Power voltage (VCC, AVCC0): 5V Operating frequency (ICLK): 32MHz Peripheral operating frequency (PCKKB, PCLKC): 32MHz DSAD0 operating frequency (f_{OP}): 16MHz DSAD0 modulator clock frequency (f_{MOD}): 4MHz
Load cell		Minebea Mitsumi Inc. BCL-300GM-C3
RS485-USB I/F		FTDI USB-COM485-PLUS2
Host	QE for AFE version	Renesas QE for AFE V2.1.1
	Modbus version	QModMaster 0.5.3-beta
IDE		Renesas e^2 Studio 2023-04 Renesas Smart Configurator V23.4.0
Tool Chain		Renesas CC-RX V3.5.0
Emulator		E2 Emulator Lite

4. Related Documents

- R01UH0972 RX23E-B Group User's Manual: Hardware
- R01AN6364 RX23E-B Group RSSKRX23E-B Board Control Program
- R12UZ0108 RSSKRX23E-B User's Manual

5. RX23E-B-QFN40-WT

5.1 Board Specifications

Table 5-1 RX23E-B-QFN40-WT Specifications

Item	Description
MCU	R5F523E6MDNF, or R5F523E6MGNF Note
External dimensions	22mm x 16mm
Layer structure	4 layers, Laminating order: Signal - GND - Power supply - Signal
Operating voltage	Recommended operating voltage: 6 to 12.6V Maximum operating voltage: 14V
Current consumption	Typ. 44mA (When connecting to 350Ω load cell, during measurement)
Communication I/F	RS-485, Half-duplex communication Maximum communication speed: 1Mbps Terminating resistor: 120Ω
Compatible emulator	Renesas E2 Emulator, E2 Emulator Lite

Note: Either one is mounted. The mounted MCU cannot be specified.

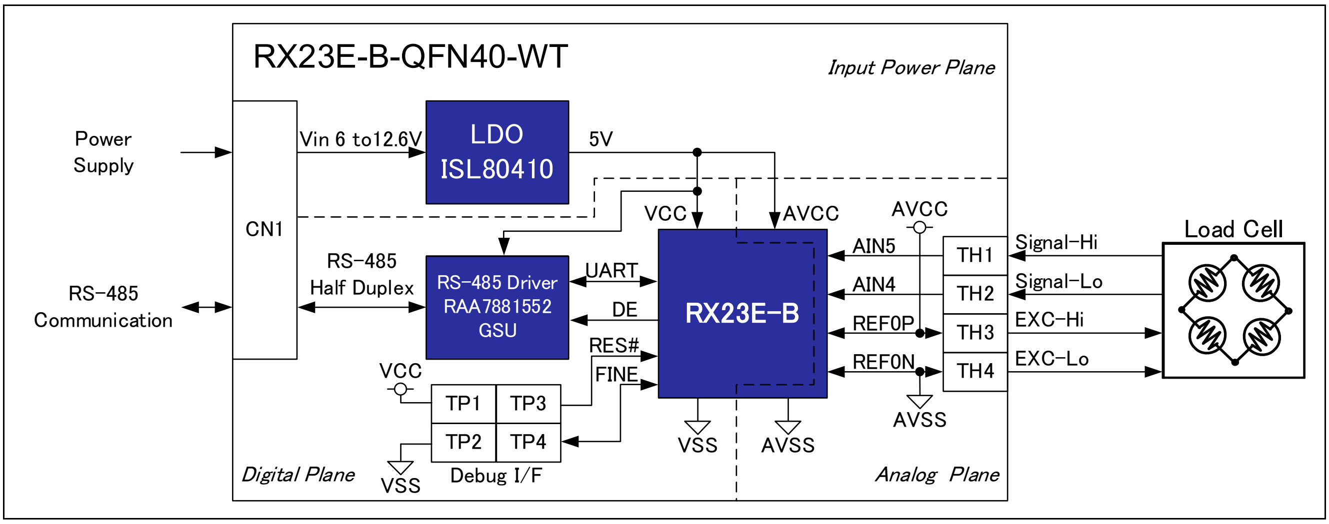


Figure 5-1 System Configuration Diagram

5.2 Circuit Diagram

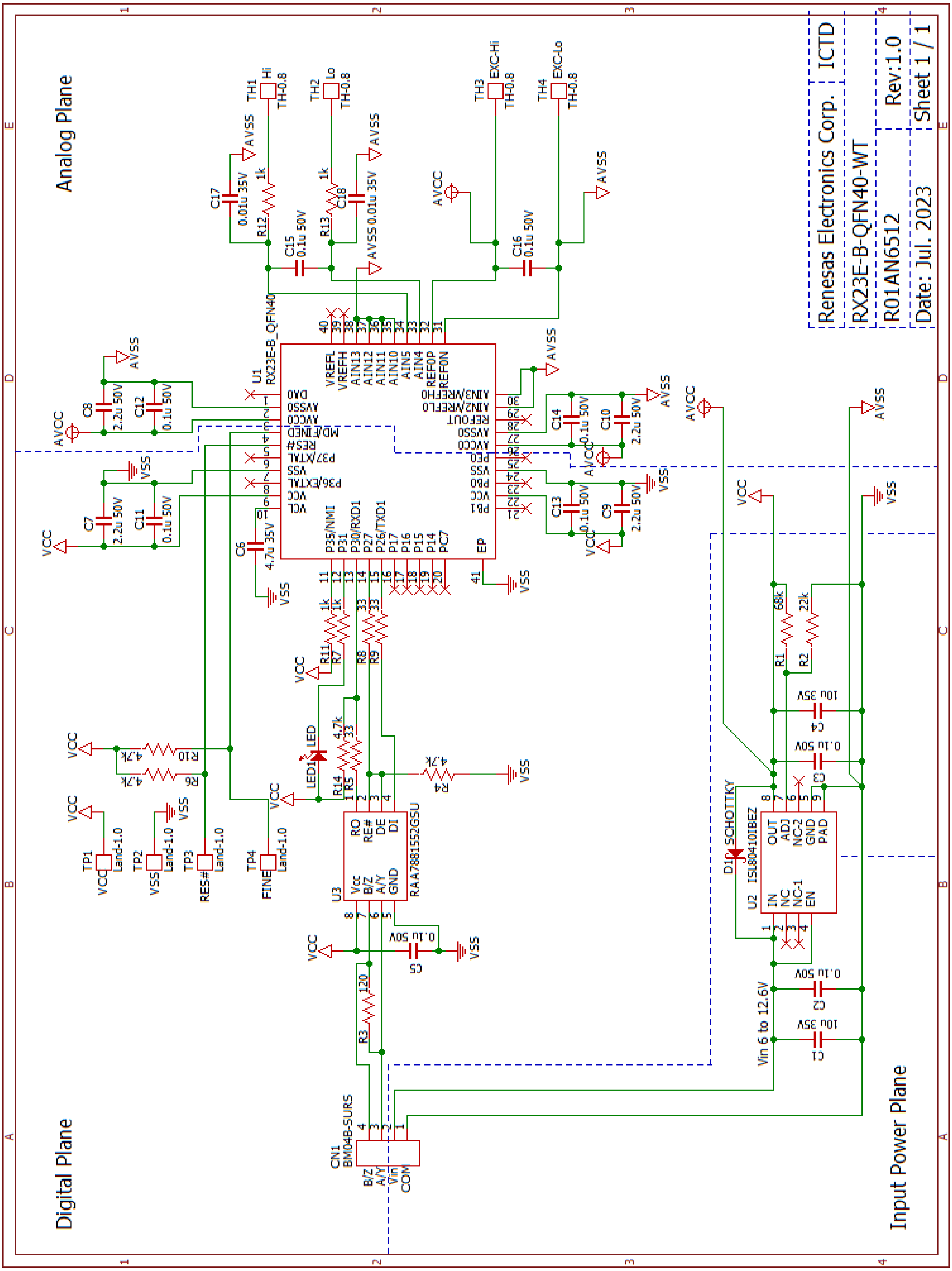


Figure 5-2 RX23E-B-QFN40-WT Circuit Diagram

5.3 Bill of Materials

No.	Q'ty	Reference Designator	Description	Part Name	Manufacturer Part Name	Maker Name
1	1	U1	RX23E-B	IC	R5F523E6MDNF	Renesas
2	1	U2	LDO	IC	ISL80410IBEZ	Renesas
3	1	U3	RS-485 Driver	IC	RAA788155GSU	Renesas
4	1	CN1	4pin	Connector	BM04B-SURS-TF(LF)(SN)	JST
5	2	C1,C4	10u 35V	Capacitor	GRM21BR6YA106KE43	Murata
6	9	C2,C3,C5,C11 C12,C13,C14, C15,C16	0.1u 50V	Capacitor	GRM155R71H104KE14	Murata
7	1	C6	4.7u 35V	Capacitor	GRM219R6YA475KA73	Murata
8	4	C7,C8,C9,C10	2.2u 50V	Capacitor	GRM188R61H225KE11	Murata
9	2	C17,C18	0.01u 35V	Capacitor	GRM1555CYA103GE01	Murata
10	1	D1	SCHOTTKY	Diode	RB551VM-30TE-17	Rohm
11	1	LED1	Green	LED	SML-D13FWT86	Rohm
12	1	R1	68k 1%	Resistor	RK73H1ETTP6802F	KOA
13	1	R2	22k 1%	Resistor	RK73H1ETTP2202F	KOA
14	1	R3	120 5%	Resistor	RK73B2ETTD121J	KOA
15	4	R4,R6,R10,R14	4.7k 5%	Resistor	RK73B1ETTP472J	KOA
16	3	R5,R8,R9	33 5%	Resistor	RK73B1ETTP330J	KOA
17	4	R7,R11,R12, R13	1k 1%	Resistor	RK73H1ETTP1001F	KOA

Note: This list may be changed without notice.

5.4 Pattern Diagram (Viewed from the Part Side)

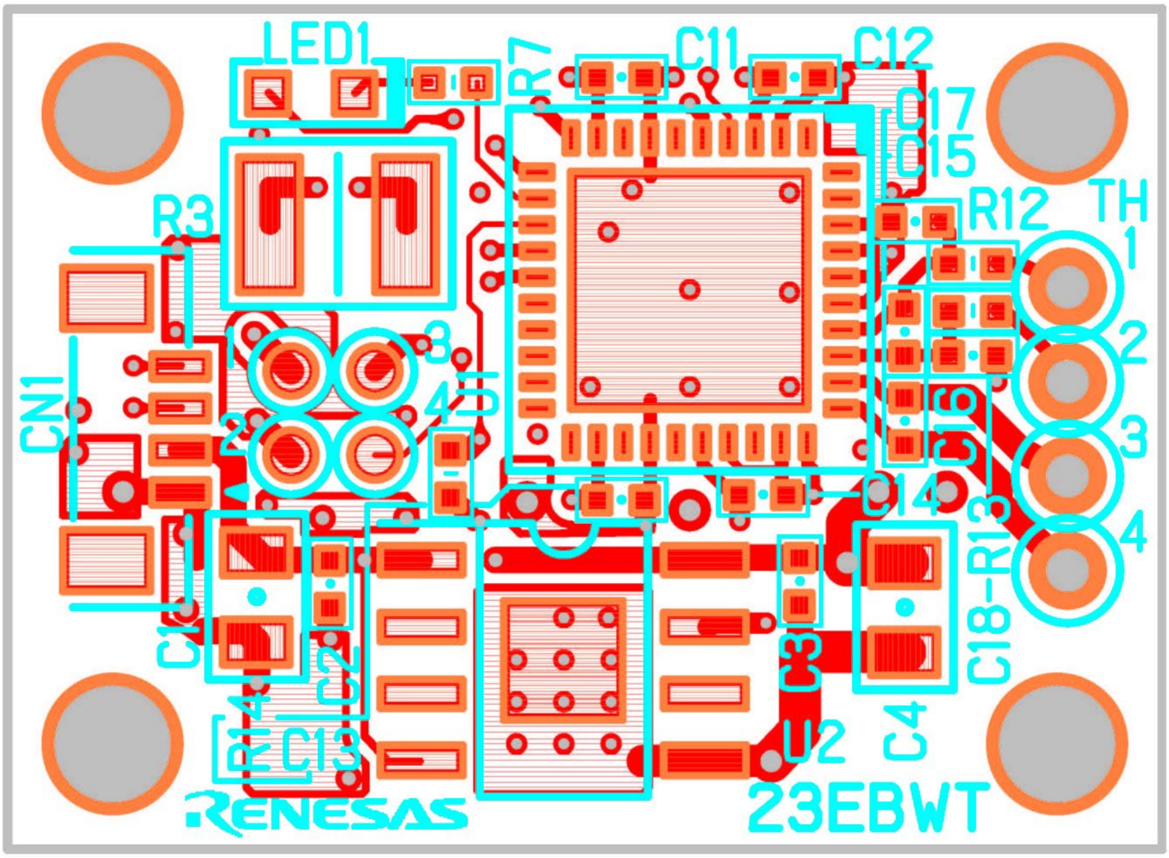


Figure 5-3 Layer 1

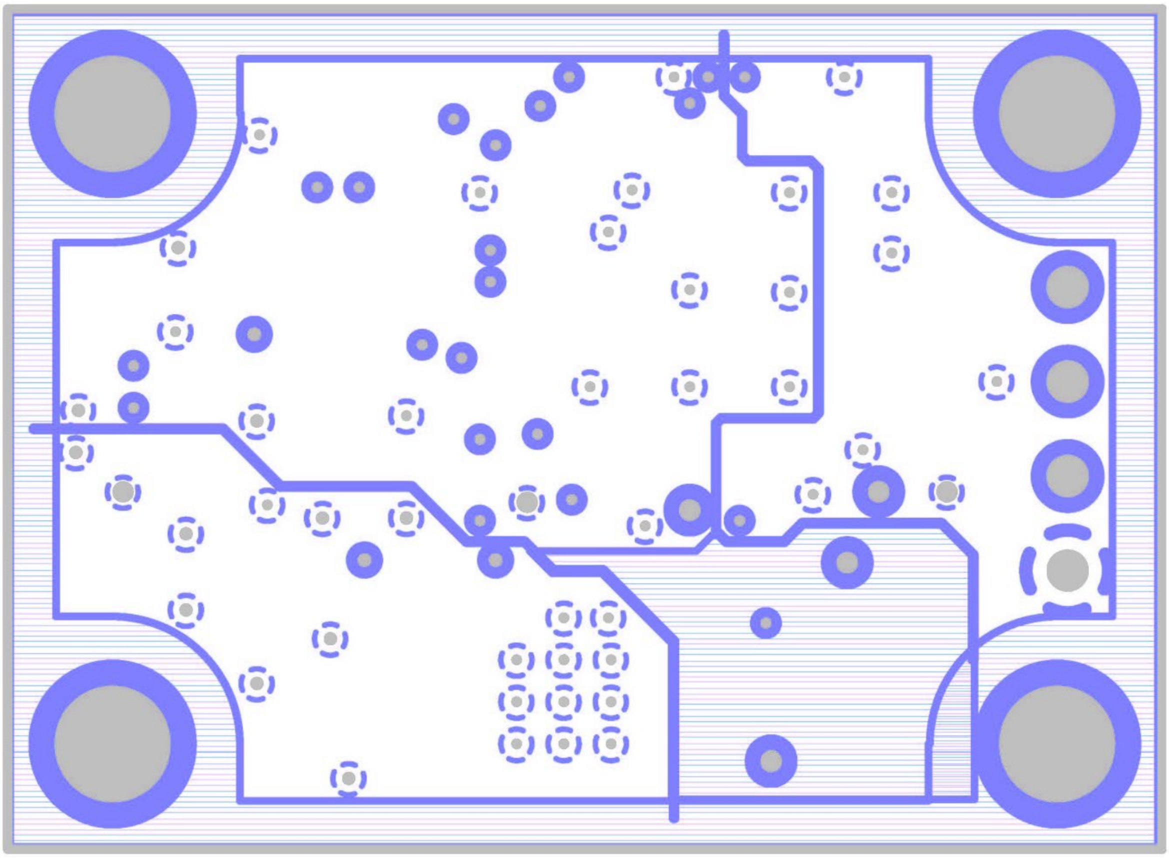


Figure 5-4 Layer 2

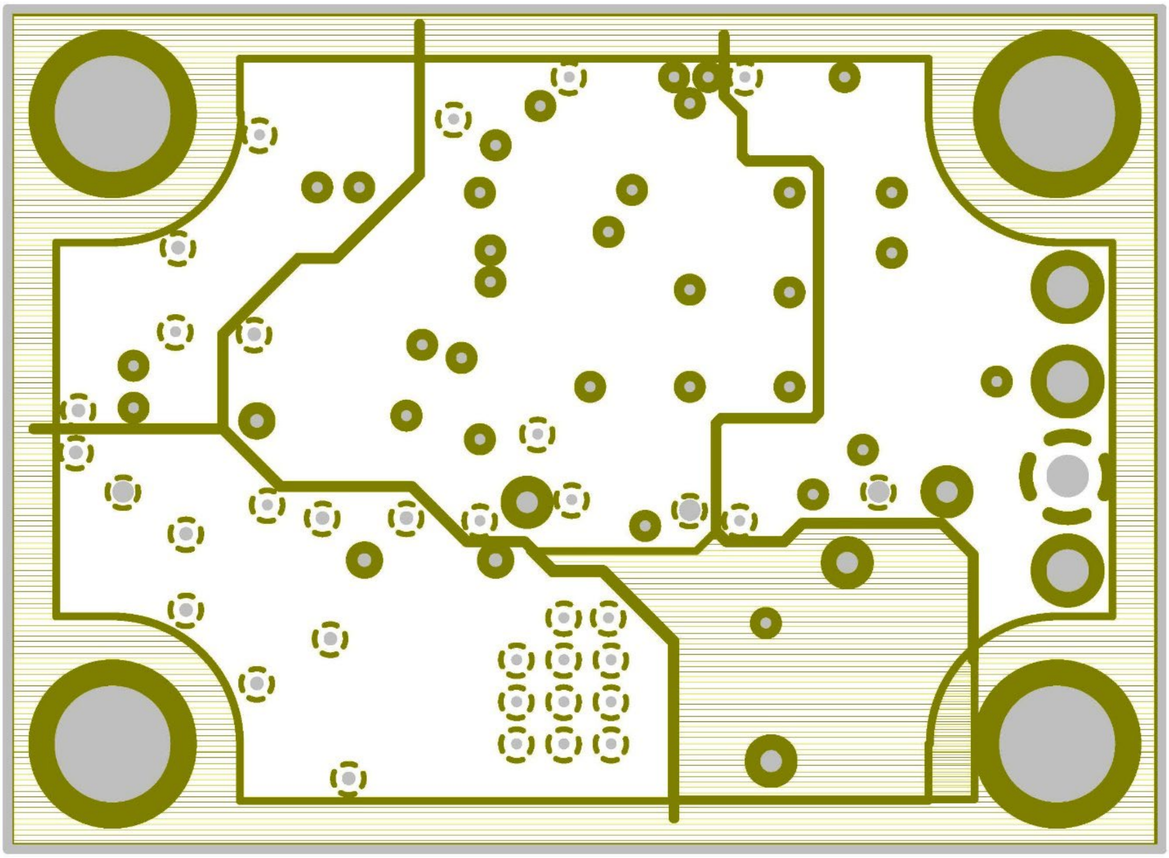


Figure 5-5 Layer 3

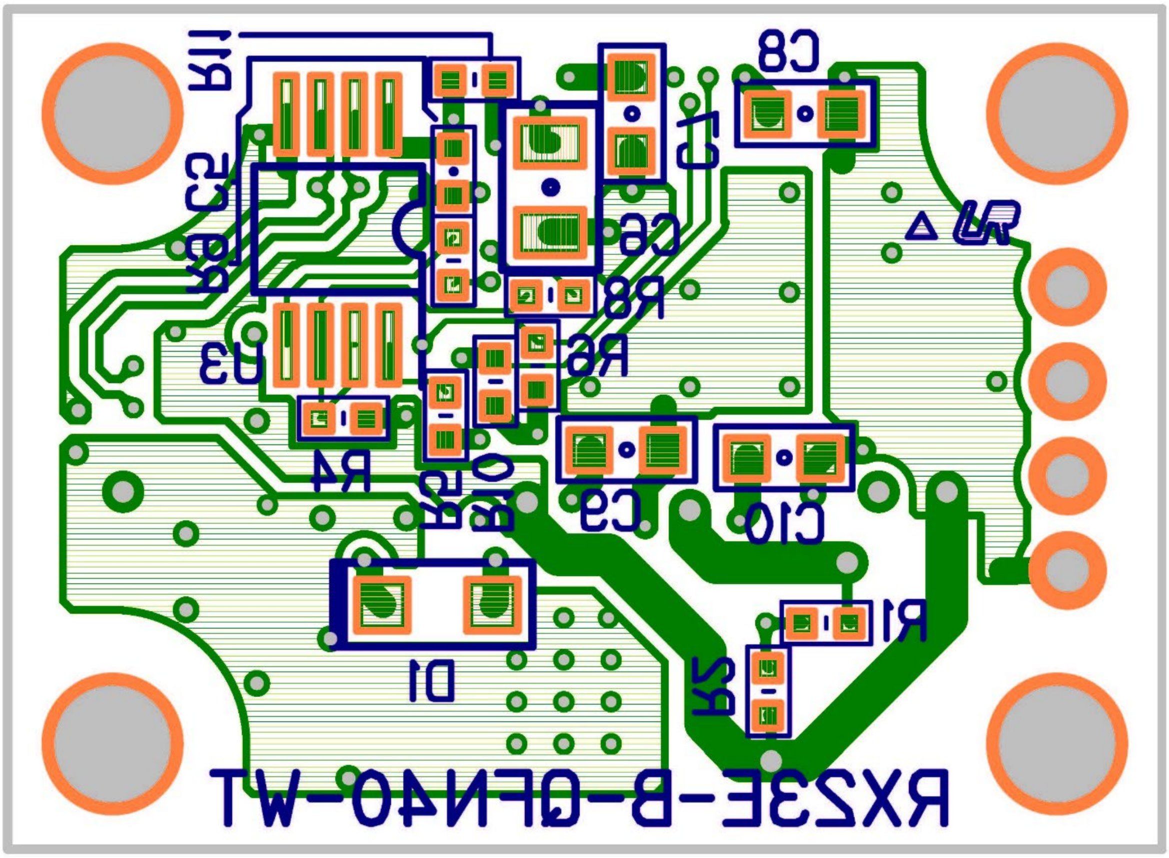


Figure 5-6 Layer 4

6. Weight Measurement Method

Connection between RX23E-B-QFN40-WT and load cell is shown in Figure 6-1.

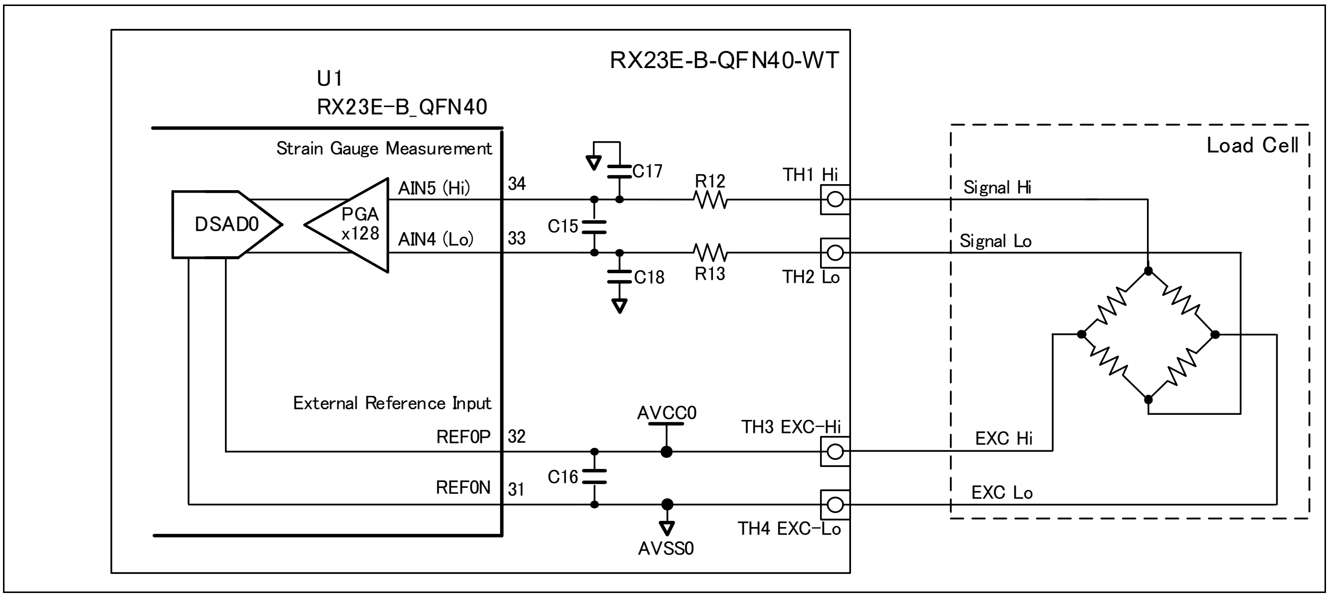


Figure 6-1 Connection between RX23E-B-QFN40-WT and Load Cell

6.1 Load Cell

The load cell used in this example outputs a weight as a voltage by using a Wheatstone bridge circuit. Table 6-1 shows an excerpt of the specifications of the load cell used, and Figure 6-2 shows the weight vs. output voltage characteristics obtained from the specifications for an applied voltage of 5V and the weight vs. output voltage characteristics including the error range of the rated output.

Table 6-1 Excerpt of Specifications of Load Cell BCL-300GM-C3 (Minebea Mitsumi Inc.)

Item	Value
Recommended Excitation	10V or less
Maximum Excitation	15V
Rated Capacity	300g
Rated Output	0.9 ±0.1 mV/V
Zero Balance	±0.04 mV/V

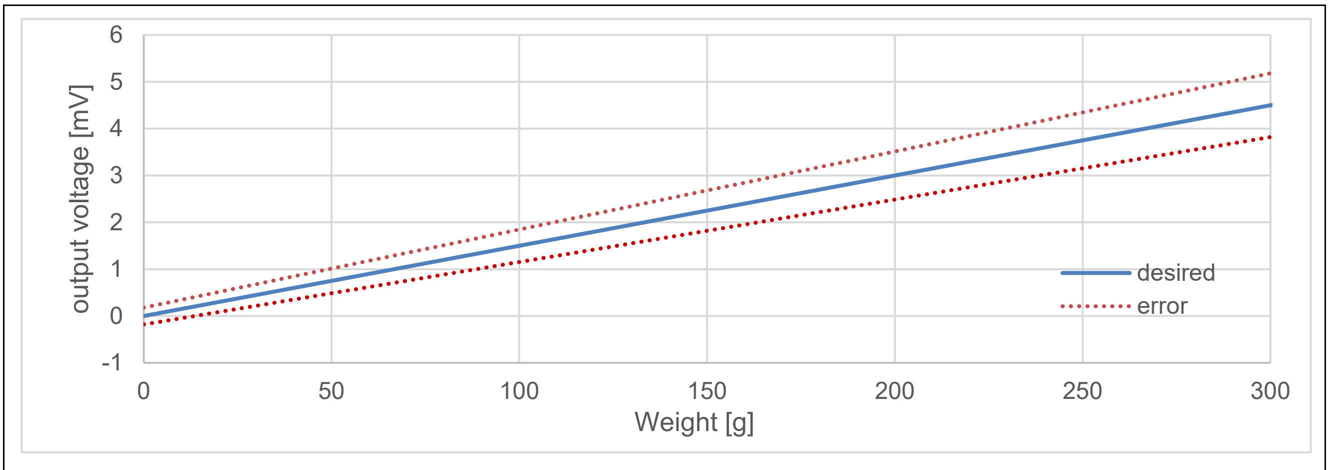


Figure 6-2 Weight vs. Output Voltage Characteristics of BCL-300GM-C3 (Applied Voltage 5V)

In this example, the voltage between the output pins of the load cell is A/D converted with the voltage supplied to the load cell as the reference voltage as shown in Figure 6-1.

If the output voltage of the load cell is non-linear in relation to the weight, the characteristic curve is divided into multiple regions and linear approximation, for example, is performed in each of the regions to increase the measurement accuracy, thereby matching the characteristic curve.

In this example, the region is regarded as a single linear characteristic without being divided, and the voltage is converted to weight with linear interpolation.

Assuming that the voltage applied to the load cell is V_{cc}, the rated output is RO, and the rated load is M_{max}, the output voltage V for the weight M can be expressed with the formula below.

$$V = \mathrm{RO} \cdot V_{cc} \cdot \frac{M}{M_{max}}$$

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

$$M = \alpha V + \beta, \qquad \begin{cases} \alpha = \dfrac{M_{max}}{\mathrm{RO} \cdot V_{cc}} \\ \beta = 0 \end{cases}$$

For the output voltage of the load cell, an error occurs due to the rated output, zero balance, and so on. Thus, the coefficients α and β in the equation above are corrected with calibration.

Table 6-2 shows the measurement conditions. If the oversampling ratio is not a power of two, the digital filter of the DSAD0 generates a gain of x1/2 to x1. The Sinc filter gain correction value is set so that the gain of the digital filter will become 1.

Table 6-2 Load Cell Measurement Conditions

Item	Condition	Remarks
PGA gain G_{PGA}	x128	
DSAD0 reference voltage V_{REF}	5V	Voltage applied to the load cell. (REF0P=AVCC0, REF0N=ACSS0)
Oversampling ratio OSR	399872	A/D conversion value output rate: Approx. 10SPS
Sinc filter gain correction value	1.477629984615	1/G_{DF}
DSAD0 output format	2's Complement	

6.2 Weight Calculation Procedure

Conversion from A/D conversion value to weight is performed with the procedure below.

(1) Smoothing of the A/D conversion value
Variations in A/D conversion results due to noise, vibration, and other factors are removed with a filter, etc. In this example, a smoothing process is performed with an 8-sample moving average.

(2) Weight conversion
In the above-mentioned weight conversion formula, the voltage is replaced by the A/D conversion value for calculation. Assuming that the PGA gain is G_{PGA}, the reference voltage of the DSAD0 is V_{REF}, and the A/D conversion value is DATA, the weight can be determined from the 24-bit resolution of the DSAD0 with the formula below.

$$\begin{aligned} M &= \alpha V + \beta \\ &= \alpha \cdot \frac{2V_{REF}}{2^{24} \cdot G_{PGA}} \cdot \mathrm{DATA} + \beta \\ &= \alpha \cdot \frac{V_{REF}}{2^{23} \cdot G_{PGA}} \cdot \mathrm{DATA} + \beta, \qquad V_{REF} = AVCC0 - AVSS0 \end{aligned}$$

From the formula above, the formula for calculating the weight from the A/D conversion value is defined as below.

$$M = \alpha V + \beta = a \cdot \mathrm{DATA} + b, \qquad \begin{cases} a = \alpha \cdot \dfrac{V_{REF}}{2^{23} \cdot G_{PGA}} \\ b = \beta = 0 \end{cases}$$

6.3 Calibration

To improve the measurement accuracy for the error in the load cell, coefficients a and b in the formula to convert A/D conversion values to weight can be calibrated.

This process is carried out using two reference weights, such as counterweights and their corresponding A/D conversion values, as follows:

(1) Obtain the A/D conversion value, $DATA_1$, for weight M_1 using the first reference weight

(2) Obtain the A/D conversion value, $DATA_2$, for weight M_2 using the second reference weight

(3) Calculate the coefficients a and b of the line passing through ($DATA_1$, M_1) and ($DATA_2$, M_2) with the formula below and apply them

$$\begin{cases} a = \dfrac{M_2 - M_1}{DATA_2 - DATA_1} \\ b = M_1 - a \cdot DATA_1 = M_2 - a \cdot DATA_2 \end{cases}$$

6.4 Zero Reset

The measured weight is corrected by subtracting the reference measurement result assumed to be the zero weight from the weight conversion result.

The reference value is the value resulting from conducting measurement in the zero-weight state and converting the value to weight.

7. Communication

Communication is conducted as RS-485, half-duplex communication. This program uses QE for AFE or Modbus RTU as the communication protocol. Table 7-1 lists the communication conditions for each communication protocol.

Table 7-1 Communication Conditions

Item	QE for AFE	Modbus RTU
Communication speed	1,000,000 bps	115,200 bps
Data length	8 bits	
Start bit	1 bit	
Parity	None	Even parity
Stop bit	1 bit	

7.1 QE for AFE

For details about the QE for AFE communication specifications, refer to the Application Note "RX23E-B Group RSSKRX23E-B Board Control Program".

Note: Since QE for AFE is based on full-duplex communication, transmission and reception may conflict and stop on stop measurement.

7.2 Modbus RTU

Operation setting and measurement result acquisition are performed with Modbus RTU communication. For details of Modbus RTU, refer to the Modbus official site (https://modbus.org/specs.php).

In this example, communication is conducted under the conditions listed in Table 7-2.

Table 7-2 Modbus RTU Communication Conditions

Item	Condition
Slave address	H’01
Silent interval	(3.5 bytes or more)
Maximum receive byte interval	(3 byte)
Maximum transmit byte interval	(2 byte)
Response time	1ms or less
Maximum frame length	256 byte
Supported Query functions	H’01: Read Coil H’02: Read Status H’03: Read Holding Register H’04: Read Input Register H’05: Write Single Coil H’06: Write Single Holding Register H’10: Write Multiple Holding Register
Supported Exception codes	H’01: Illegal function H’02: Illegal data address H’03: Illegal data H’04: Device Failure H’05: Acknowledge H’06: Device Busy

7.2.1 Supported Frame Format

The message frame format is shown in Table 7-3. The function codes used in this example and the data format for each function are listed in Table 7-4, and the storage order for data of single precision floating point type is shown in Table 7-5.

Table 7-3 Message Frame for Modbus RTU

Address	Function	Data	CRC
1 byte	1 byte	N byte	2 bytes

Table 7-4 Supported Function Code and Description of Data

Supported Function Code	Type	Bytes of Data	Data +0	+1	+2	+3	+4	+5	+6	+2m-1 +2k+3	+2m +2k+4
Read Coil (H'01) Read Input Status (H'02)	query	4	Start Address		Num of read (M)						
			Upper	Lower	Upper	Lower					
	response	1+ Round up of (M/8)	Data bytes	Data1	Data2					Data (roundup of M/8)	
Read Holding Register (H'03) Read Input Register (H'04)	query	4	Start Address		Num of read (m)						
			Upper	Lower	Upper	Lower					
	response	1+2m	Data bytes	Data1						Data m	
				Upper	Lower					Upper	Lower
Write Single Coil (H'05) Write Single Holding Register (H'06)	query	4	Address		Data						
			Upper	Lower	Upper	Lower					
	response	4	Address		Data						
			Upper	Upper	Upper	Lower					
Write Multiple Holding Registers (H'10)	query	5+2k	Start Address		Num of Register (k)		Data bytes	data1		data k	
			Upper	Lower	Upper	Lower		Upper	Lower	Upper	Lower
	response	4	Start Address		Num of Register (m)						
			Upper	Lower	Upper	Lower					
exception	response	1	Exception Code H'01: Illegal function H'02: Illegal data address H'03: Illegal data H'04: Device Failure H'05: Acknowledge H'06: Device Busy								

Table 7-5 Single Precision Floating Data Format

bit	31	30 24	23	22 16	15 8	7 0
Allocation	sign	exponent		faction		
	Upper byte		Lower byte		Upper byte	Lower byte
	Upper 16bit				Lower 16bit	

7.2.2 Data

The data used in this example and their arrangement are shown in Table 7-6.

Table 7-6 Data List

Function	Address	Size	Format	Name	Description
Coil	0	2byte	uint32	Measurement	Weight measurement H'0000: Stop (default) H'FF00: Measurement
	1			Calibration Step1	Calibration 1 H'0000: End/stop (default) H'FF00: Start
	2			Calibration Step2	Calibration 2 H'0000: End/stop (default) H'FF00: Start
	3			Zero Reset	Zero reset during weight measurement H'0000: End/stop (default) H'FF00: Processing start
	4			Parameter Reset	Set the Holding Register value to the initial value. H'0000: End/stop (default) H'FF00: Reset start
Input Status	0	2byte	uint32	DSAD0 OVF	Error/Overflow flag during DSAD0 operation
	1			DSAD0 ERR	
	2			Calibration 1 end	Calibration 1 end notification
Input Register	0	4byte	float	Weight	Measured weight
	2	4byte	int32	A/D Value	A/D conversion value
Holding Register	0	4byte	float	Zero Weight	Zero reset correction weight
	2	4byte	float	Calibration Weight 1	Specified weight for calibration 1
	4	4byte	float	Calibration Weight 2	Specified weight for calibration 2
	6	4byte	float	Calibration1 Delay	Wait time [s] before starting Calibration 1
	8	4byte	float	Coefficient a	Weight conversion coefficient a
	10	4byte	float	Coefficient b	Weight conversion coefficient b
	12	2byte	uint16	Num of Moving Average	Moving averaging count for weight measurement
	13	2byte	uint16	Num of Zero Reset Average	Averaging count for zero reset processing
	14	2byte	uint16	PGA Gain	DSAD0 CR0.GAIN register setting value
	15	2byte	uint16	FSEL	DSAD0 MR0.FSEL register setting value
	16	4byte	uint32	OSR	DSAD0 OSR0 register setting value
	18	4byte	uint32	SGCR	DSAD0 SGCR0 register setting value
	20	2byte	uint16	Num of Calibration Average	A/D conversion value averaging count for calibration

7.2.3 Operation

Operation via Modbus equivalent to "Table 1-1 QE for AFE " is shown in Table 7-7.

Table 7-7 Items Operable via Modbus

Item	Operation	Remarks
Measurement start and stop	Operate Coil:0	LED1 is OFF during measurement
Zero reset	Set Coil:3	Enabled only during measurement
Calibration	STEP1 Set measured weight 1 in HoldingReg:2-3 and set Coil:1 STEP2 While LED1 is blinking, set measured weight 2 in HoldingReg:4-5 and set Coil:2	Enabled while measurement is stopped LED1 is OFF during A/D conversion LED1 blinks five times in case of abnormal termination. Refer to "6.3 Calibration".
Specifying the moving averaging count for measurement	Set 1 to 128 in HoldingReg:10, default: 8	Enabled only during standby (LED1 is ON)
Specifying the averaging count for zero reset	Set 1 to 512 in HoldingReg:11, default: 256	
Specifying the averaging count for calibration	Set 64 to 512 in HoldingReg:18, default: 256	
Parameter initialization	Set Coil:4	

Table 7-8 DSAD0/AFE Register Setting Changeable Items

Item	Operation	Remarks
PGA gain	Specify CR0.GAIN setting value in HoldingReg:14	
Oversampling ratio	Specify OSR0 setting value in HoldingReg:16-17	The lower limit of oversampling ratio setting is OSR1=256 and OSR2=2. If it is lower than this, communication error will occur.
Digital filter selection	Specify MR0.FSEL setting value in HoldingReg:15	
Digital filter gain correction	Specify SGCR0 setting value in HoldingReg:18-19	

8. Sample Program

8.1 Overview of Operation

Figure 8-1 shows the process flow of this sample program.

a

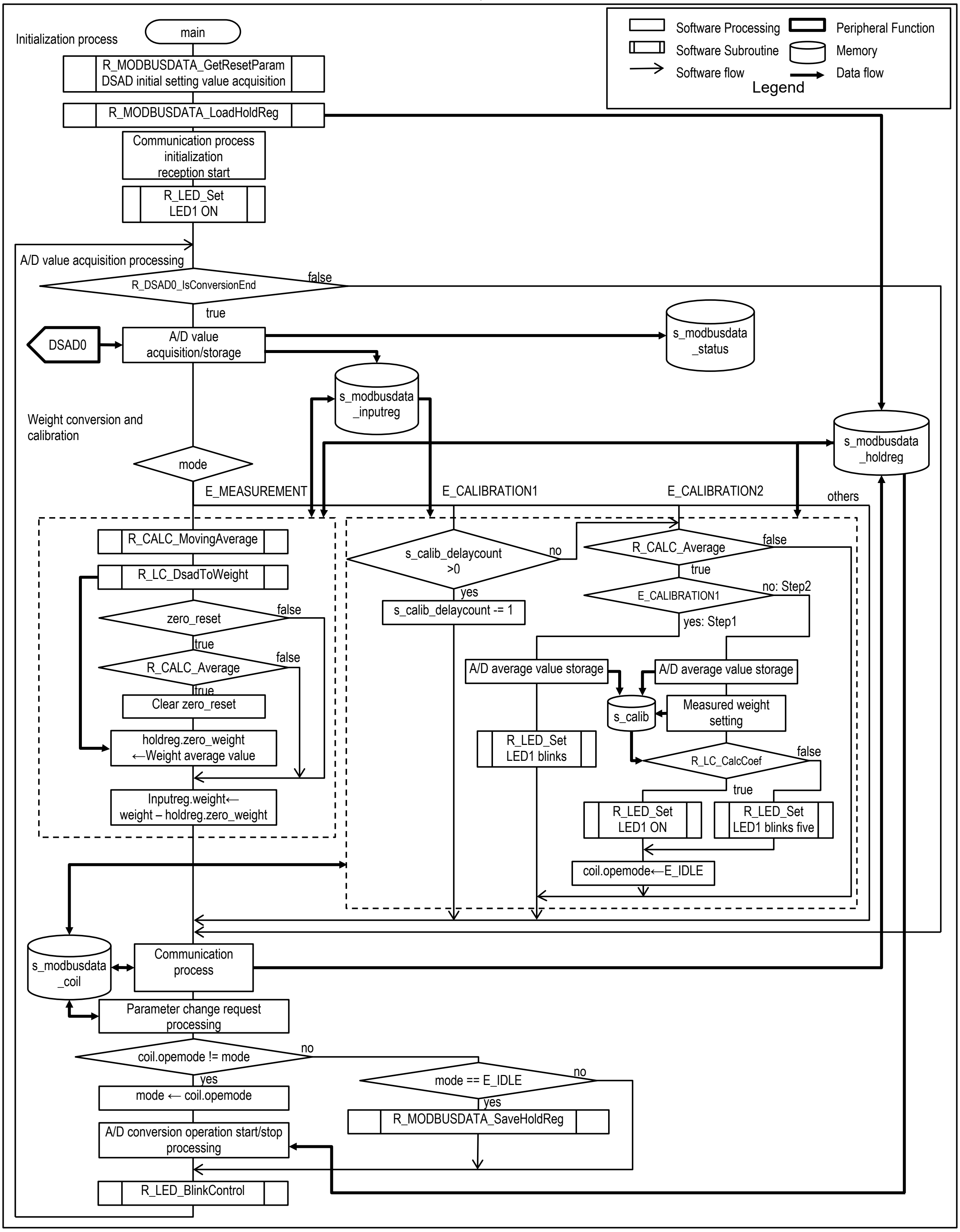


Figure 8-1 Weight Measurement Process Flow

This program works based on ModbusData. The operating mode is specified with the Coil member ope_mode. Operating modes are listed in Table 8-1.

Table 8-1 Operating Modes

Name	ope_mode	Description
E_IDLE	0	Standby
E_MEASUREMENT	1	Measurement
E_CALIBRATION1	2	Calibration STEP1
E_CALIBRATION2	4	Calibration STEP2

The following provides an overview of each of the processes in Figure 8-1.

- Initialization process
  - DSAD0 initial setting value acquisition
  - Initialization of holding register in ModbusData and loading of parameters stored in E2 data flash
  - Initialization of communication process and start receiving
  - Turning LED1 on

- A/D value acquisition processing
  If the end of A/D conversion (ADI0) is detected, A/D conversion results are acquired, and the A/D value is stored in the Input register in ModbusData and A/D conversion error information is stored in the Status in ModbusData.

- Weight conversion and calibration
  The acquired A/D value is processed according to the operating mode "mode" in A/D conversion start/stop processing.
  - mode: E_MEASUREMENT
    The A/D value subject to moving averaging the number of times specified in the Holding register member moving_average in ModbusData is converted to weight, using the Holding register members coef_a and coef_b, and the result of subtracting the Holding register member zero_weight is stored in the Input register member weight in ModbusData as the measured weight.
    When a zero reset has been requested, the converted weight is averaged the number of times specified in the Holding register member zero_reset_average. The result is stored in the Holding register member zero_weight as the zero reset weight.
  - mode: E_CALIBRATION1
    A/D value average processing is performed from the A/D value after the number of delay samples calculated from the calibration delay time and the averaging result is retained. The number of averages is determined based on the Holding register member calib_average.
  - mode: E_CALIBRATION2
    A/D value average processing is performed, and after its end, the weight conversion coefficients are calculated from this A/D average value, the A/D average value acquired with E_CALIBRATION1, and the Holding register member calib_weight1 and calib_weight2 in ModbusData, according to "6.3 Calibration". The calculation results are stored in the Holding register members coef_a and coef_b in ModbusData.

- Communication process
  A request from the Host is processed and the transmission of a response is set. For the QE for AFE version, the measured weight is transmitted. For details, refer to "8.3 Communication Control".

- Parameter change request processing
  The following processing, requested by the Coil in ModbusData, is performed.
  — reset_param: During standby (E_IDLE), the Holding register in ModbusData is returned to the initial value.
  — zero_reset: During measurement (E_MEASUREMENT), zero reset processing is initialized.

- A/D conversion operation start/stop processing
  If the Coil member ope_mode in ModbusData is changed, the following processing is performed according to the new ope_mode.
  — ope_mode: E_MEASUREMENT
- Applying the DSAD0 setting value of the Holding register in ModbusData
- Starting A/D conversion
- Initializing the A/D value moving average processing and setting the moving_average of the Holding register as the moving averaging count
- Turning LED1 off

  — ope_mode: E_CALIBRATION1
- Applying the DSAD0 setting value of the Holding register in ModbusData
- Starting A/D conversion
- Converting the calibration delay time to the number of delay samples
- Initializing the A/D value average processing and setting the calib_average of the Holding register as the averaging count
- Turning LED1 off

  — ope_mode: E_CALIBRATION2
- Initializing the A/D value average processing and setting the calib_average of the Holding register as the averaging count
- Turning LED1 off

  — ope_mode: E_IDLE
- Stopping A/D conversion
- Turning LED1 on

- E2 data flash storage processing
  If the Coil member opemode in ModbusData does not change from E_IDLE, and there is a change in the retention parameter in the Holding register in ModbusData, it will be stored in E2 data flash.

8.2 Peripheral Functions and Pins Used

Table 8-2 lists the peripheral functions used in this example, and Table 8-3 lists the pins used. Also, Table 8-4 shows the clock settings. Unused pins are set to output Low.

The settings for the peripheral functions are generated by using the code generation function of Smart Configurator. The following shows the peripheral function settings.

Table 8-2 Peripheral Functions

Peripheral function	Use	
	QE for AFE version	**Modbus version**
DSAD0	A/D conversion of load cell output	
SCI1	Communication with QE for AFE	Communication with Modbus host
DMAC0	Reception of packets from QE for AFE	-
DMAC1	Transmission of packets to QE for AFE	-
TMR0	-	Communication with Modbus host
DTC	-	Communication with Modbus host
CRC	-	Modbus frame error check
CMT1	LED1 blinking cycle	
P31	LED1 ON/OFF control	
P27	RS-485 driver transmission/reception switching	
E2DataFlash	Saving of retention parameters	
LVD	Setting of reset voltage	

Table 8-3 Pins

Pin name	I/O	Use
AIN5	I	Input pin on the positive side of the load cell
AIN4	I	Input pin on the negative side of the load cell
REF0P	I	Positive side of the DSAD0 reference voltage input pin
REF0N	I	Negative side of the DSAD0 reference voltage input pin
P26/TXD1	O	UART1 transmit pin
P30/RXD1	I	UART1 receive pin
P27	O	RS-485 driver transmission/reception switching control pin
P31	O	LED1 ON/OFF control pin

Table 8-4 Clock Settings

Item	Setting
Clock used	HOCO clock (32MHz) Enable HOCO oscillation after reset
SCKCR (FCLK)	x1 (32MHz)
SCKCR (ICLK)	x1 (32MHz)
SCKCR (PCLKA)	x1 (32MHz)
SCKCR (PCLKB)	x1 (32MHz)
SCKCR (PCLKC)	x1 (32MHz)
SCKCR (PCLKD)	x1 (32MHz)

8.2.1 Load Cell Measurement

In load cell measurement, DSAD0 is used in continuous scan mode. Table 8-5 shows the setting conditions based on the measurement conditions in Table 6-2.

Table 8-5 DSAD0 Settings

Continuous scan mode

Item		Setting
Operation clock setting		PCLK/2(16MHz)
Conversion start trigger source		Software trigger
Interrupt setting	Enable ΔΣA/D conversion completion interrupt (ADI0)	Enable, Priority: Level 0(disabled)
	Enable ΔΣA/D conversion scan completion interrupt (SCANEND0)	Disable
	Enable ΔΣA/D channel change interrupt (CHCHG0)	Disable
Voltage fault and disconnection setting		Not used
Analog input channel setting		0
Analog input setting	Positive input signal	AIN5
	Negative input signal	AIN4
	Reference input	REF0P/REF0N
	Positive reference voltage buffer	Disable
	Negative reference voltage buffer	Disable
Amplifier setting	Amplifier selection	PGA
	PGA gain setting	x128
ΔΣA/D conversion setting	A/D conversion mode	Normal operation
	Data format	Two's complement
	A/D conversion number	1
	First stage oversampling ratio	256
	Second stage oversampling ratio	1562
	Set offset calibration value	Not used
	Set gain calibration value	Not used
Disconnect detection assist setting		Disable
Digital filter setting	Sinc filter select	Sinc4 + Sinc4
	Set sinc filter gain calibration	Used
	Sinc filter gain calibration value	1.477629984615

8.2.2 Communication

In communication with QE for AFE or the Modbus host, SCI1 is used for transmission/reception in asynchronous mode. The TXD1/P26 pin is used in built-in pull-up setting, and P27 is used in RS-485 driver transmit/receive switching.

For the QE for AFE version, DMAC0 is used to acquire receive data, and DMAC3 is used to transmit data.

For the Modbus version, DTC is used to acquire receive data and transmit data, and TMR0 is used to detect frame reception and to detect the end of frame transmission.

The following shows the setting conditions for each peripheral function.

Table 8-6 SCI1 Settings

Asynchronous mode
Operation mode: Transmission/reception

Item		Setting	
		For QE for AFE	For Modbus
Start bit edge detection setting		Falling edge on RXD1 pin	
Data length setting		8 bits	
Parity setting		None	Even
Stop bit length setting		1 bit	
Transfer direction setting		LSB-first	
Transfer rate setting	Transfer clock	Internal clock	
	Bit rate	1,000,000bps	115,200bps
	Enable modulation duty correction	Not used	Used
	SCK1 pin function	SCK1 is not used	
Noise filter setting		Not used	
Hardware flow control setting		None	
Data handling setting	Transfer data handling	Data handled by DMAC	Data handled by DTC
	Receive data handling	Data handled by DMAC	Data handled by DTC
Interrupt setting	Enable reception error interrupt (ERI1)	Not used	
	TXI1, RXI1, TEI1, ERI1 priority	Level 1	
Callback function setting		Not used	

Table 8-7 PORT2 Settings

Item	Setting	
Port selection	PORT2	
Used port	P26	P27
Setting	Unused GPIO Pull-up CMOS output	Out CMOS output

Table 8-8 DMAC Settings (for QE for AFE)

Item		Setting	
		DMAC0	**DMAC1**
Transfer setting	Activation source	SCI1 (RXI1)	SCI1 (TXI1)
	Activation source flag control	Clear interrupt flag of the activation source	
	Transfer mode	Free running mode	Normal mode
	Transfer data size	8 bits	
	Transfer count / Repeat size / Block size	-	(Setting on execution)
Source address setting	Source address	0x0008A025(SCI1.RDR) Fixed	(Setting on execution) Incremented
	Specify the transfer source as extended repeat area	-	Enable
	Extended repeat area		Lower 9 bits of the address (512 bytes)
Destination address setting	Destination address	(Set by the program) Incremented	0x0008A023(SCI1.TDR) Fixed
	Specify the transfer destination as extended repeat area	Enable	-
	Extended repeat area	Lower 9 bits of the address (512 bytes)	
Interrupt setting		Not used	

Table 8-9 TMR0 Settings (for Modbus)

Item		Setting
Count setting	Clock source	PCLK/64 (500kHz)
	Counter clear	Disabled
	Compare match A value (TCORA)	334μs
	Compare match B value (TCORB)	238μs
TMO0 output setting		Not used
Interrupt setting	Enable TCORA compare match interrupt (CMIA0)	Enabled
	Enable TCORB compare match interrupt (CMIB0)	Enabled
	Enable TCNToverflow interrupt (OVI0)	Disabled
	Priority	Level 1

Table 8-10 DTC Settings: Config_DTC_RXI1 (for Modbus)

Item		Setting		
		DTC0	DTC1	DTC2
Basic setting	Transfer data read skip	Enable		
	Address mode	Short-address mode (24 bits)		
	DTC vector base address	0x00007C00 (default value)		
Activation source setting	Activation source	SCI1 (RXI1)	-	-
	Chain transfer	Used		Not used
Chain transfer setting		Continuous		-
Transfer mode setting		Repeat mode		
Transfer data size setting		8 bits	8 bits	8 bits
Interrupt setting		An interrupt request to the CPU is disabled when specified data transfer is completed		
Block / Repeat area setting		Transfer destination		
Transfer address and count setting	Source address	0x0008A025 (SCI1.RDR) Address fixed	(Set by the program) Address fixed	
	Destination address	(Set by the program) Address incremented	0x00088208 (TMR0.TCNT) Address fixed	0x0008820A (TMR0.TCCR) Address fixed
	Count	256	1	1

Table 8-11 DTC Settings: Config_DTC_TXI1 (for Modbus)

Item		Setting
Basic setting	Transfer data read skip	Enable
	Address mode	Short-address mode (24 bits)
	DTC vector base address	0x00007C00 (default value)
Activation source setting	Activation source	SCI1 (TXI1)
	Chain transfer	Not used
Transfer mode setting		Normal mode
Transfer data size setting		8 bits
Interrupt setting		An interrupt request to the CPU is generated when specified data transfer is completed
Transfer address and count setting	Source address	(Set by the program) Address incremented
	Destination address	0x0008A023(SCI1.TDR) Address fixed
	Count	(Setting on execution)

Table 8-12 DTC Settings: Config_DTC_CMIA0 (for Modbus)

Item		Setting			
		DTC0	DTC1	DTC2	DTC3
Basic setting	Transfer data read skip	Enable			
	Address mode	Short-address mode (24 bits)			
	DTC vector base address	0x00007C00 (default value)			
Activation source setting	Activation source	TMR0(CMIA0)	-	-	-
	Chain transfer	Used			Not used
Chain transfer setting		Continuous			-
Transfer mode setting		Repeat mode			
Transfer data size setting		8 bits	8 bits	16 bits	16 bits
Interrupt setting		An interrupt request to the CPU is disabled when specified data transfer is completed			An interrupt request to the CPU is generated each time DTC data transfer is performed
Block / Repeat area setting					
Transfer address and count setting	Source address	(Set by the program) Address fixed			
	Destination address	0x0008820A (TMR0.TCCR) Address fixed	(Set by the program) Address fixed		
	Count	1	1	1	1

Table 8-13 CRC Settings (for Modbus)

Item		Setting
Calculation setting	Generating polynomial	CRC_16
	Bit order	LSB
	Initial value	0xFFFF
	Invert result of calculated value	Not used

8.2.3 LED

P31 is used to turn on and off LED1. For the blinking interval, CMT1 is used.

LED1 is on during standby, off during measurement, and blinks to indicate the calibration status.

Table 8-14 shows the settings for P31, and Table 8-15 shows the settings for CMT1.

Table 8-14 P31 Settings

Item	Setting
Port selection	PORT3
Used port	P31
Setting	Out CMOS output Output 1

Table 8-15 CMT1 Settings

Item		Setting
Count clock setting		PCLK/128
Compare match setting	Interval value	250ms
	Compare match interrupt (CMI1)	Enable Priority: Level 0 (disabled)

8.2.4 E2 Data Flash

E2 Data Flash is used to retain the set parameters. To access E2 Data Flash, the FIT flash module is used.

Table 8-16 FIT Flash Module Settings

Item	Setting
Parameter check	Enable parameter checks
Enable code flash programming	Only data flash
Enable BGO/Non-blocking data flash operation	Forces data flash API function to block until completed.
Enable BGO/Non-blocking code flash operation	Forces ROM API function to block until completed.
Enable code flash self-programming	Programming code flash while executing in RAM.

8.2.5 Voltage Detection Circuit (LVD)

The reset voltage can be set.

Table 8-17 LVD settings

Item	Setting
Voltage detection level	3.84V
Voltage detection 0 circuit start setting	Voltage monitor 0 reset enabled after reset

8.3 Communication Control

8.3.1 QE for AFE Communication

QE for AFE communication uses the communication module included in the "RX23E-B Group RSSKRX23E-B Board Control Program". For details, refer to the Application Note.

The QE for AFE communication process flow in this example is shown in Figure 8-2.

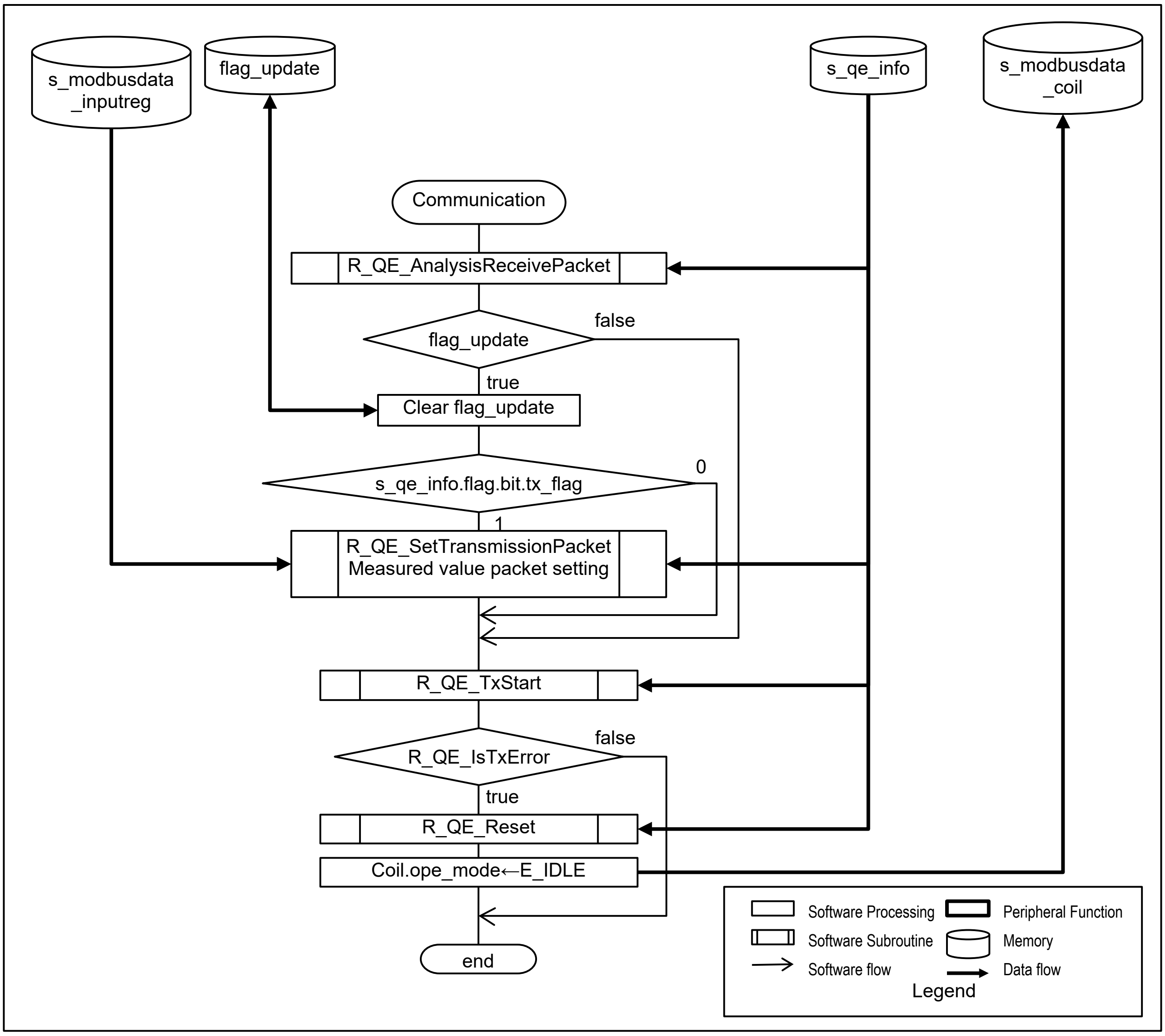


Figure 8-2 QE for AFE Communication Process Flow

8.3.2 Modbus RTU Communication

This sample program conducts data transmission/reception with DTC transfer, and the designated wait time indicating the end of communication is set with the timer TMR0.

Reception processing is handled only by DTC and TMR0, and the CPU is not involved.

Transmission processing sets transmit data in SCI1 with DTC, detects completion of data transmission with the TEI of SCI1, and waits for the transmit end with TMR0. Transmit end processing is performed with the compare match interrupt CMIB0 of TMR0.

The program detects reception with the compare match interrupt request CMIA0 of TMR0, creates the response frame for the received Modbus frame, and makes transmission settings.

8.3.2.1 Transmit/Receive Processing

A communication timing chart is shown in Figure 8-3, and the communication process flow is shown in Figure 8-4.

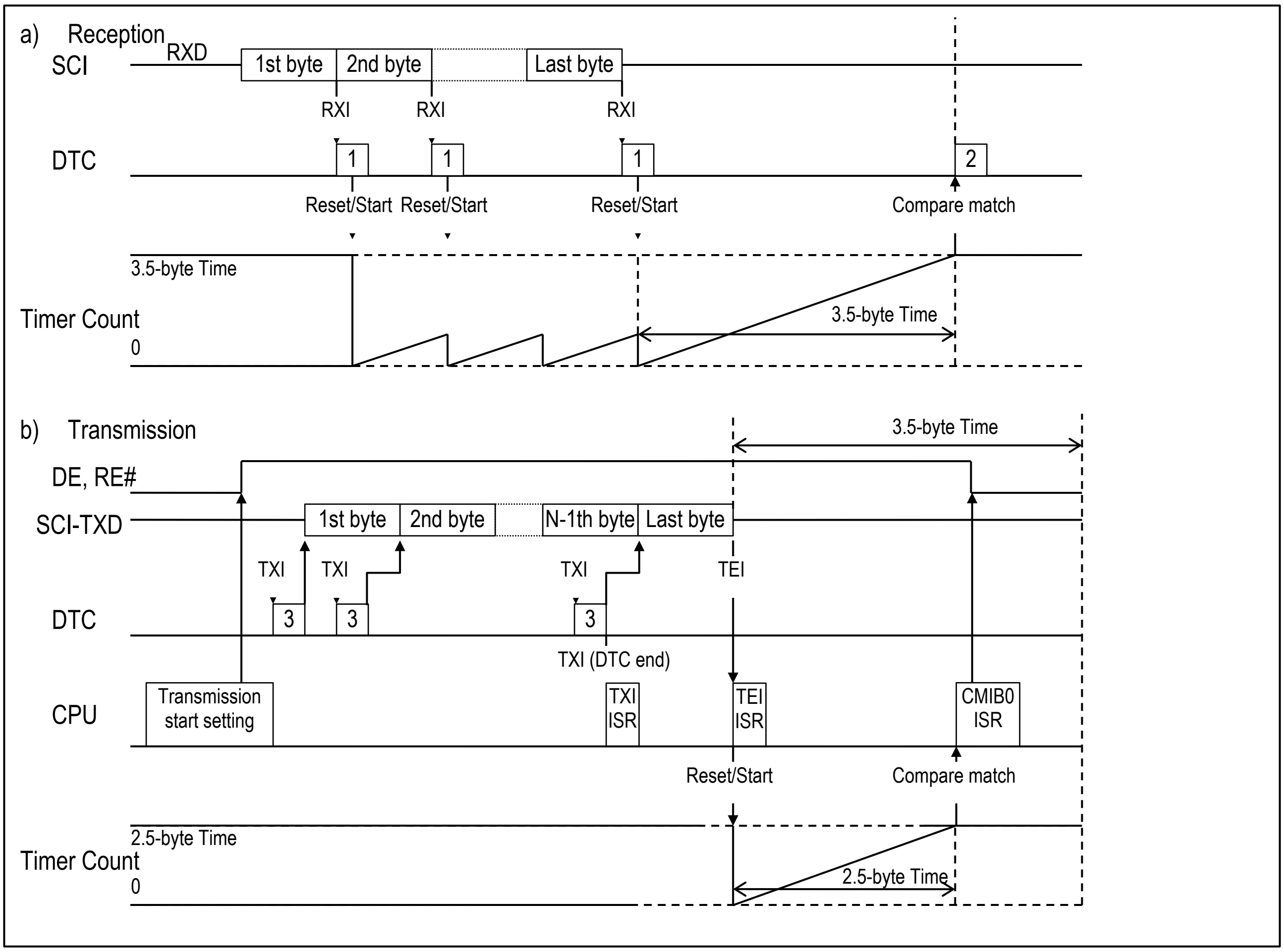


Figure 8-3 Modbus Communication Timing Chart

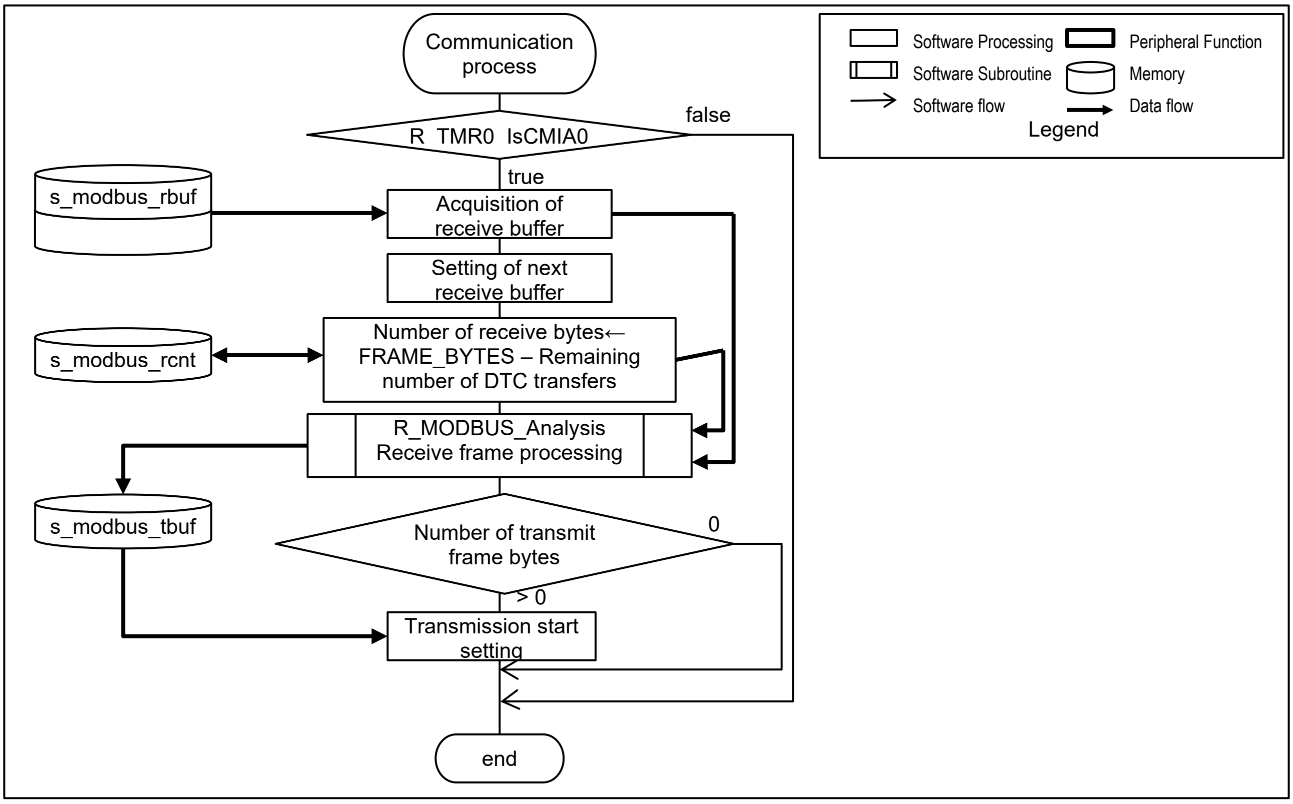


Figure 8-4 Modbus Communication Process Flow

Receive processing and transmit processing are performed as described below.

- Receive processing
  (1) At the RXI1 interrupt request for every 1-byte reception, DTC transfer 1 (DTC_RXI1) performs the following:
    - Transfer the receive data to the receive buffer on memory
    - Reset and restart TMR0

  (2) On TMR0 compare match A interrupt request (CMIA0), DTC transfer 2 (DTC_CMIA0) performs the following:
    - Stop the TMR0 count
    - Transfer the DTC transfer count to memory
    - Switch receive buffers
    - Reset the DTC transfer count

  (3) As shown in Figure 8-4, when CMIA0 is detected, the program acquires and clears the DTC transfer count and processes the Modbus receive frame in the receive buffer.

- Transmit processing
  (1) To prepare for transmission, the program performs the following in "Transmission start setting" in Figure 8-4.
    - Set DE (= RE#) to H for transmission
    - To transmit a transmission frame with a TXI interrupt of SCI, set a transmit buffer and the number of transmit bytes in DTC transfer 3 (DTC_TXI1) and permit transfer
    - Make SCI transmit start setting

(2) On TXI1 interrupt request, DTC transfer 3 (DTC_TXI1) transfers 1 byte of the transmission frame to the transmission register.

(3) On TXI1 interrupt request due to the completion of DTC transfer 3 (DTC_TXI1), the interrupt handler performs the following actions:
- Enable the transmit end interrupt (TEI1)
- Disable TXI1 interrupt

(4) On TEI1 interrupt, the interrupt handler performs the following actions:
- Reset TMR0
- Disable CMIA0 and enable CMIB0
- Start the TMR0 count
- Disable TEI1 interrupt

(5) On a TMR0 compare match interrupt request (CMIB0), the interrupt handler performs the following actions:
- Stop the TMR0 count
- Set DE (= RE#) to L
- Disable CMIB0 and enable CMIA0

8.3.2.2 Receive Frame Processing

The received self-addressed frame is processed, a response frame is generated, and set transmission.

The processing on receive frames and whether response frames are transmitted are shown in Table 8-18, and the receive frame processing flowchart is shown in Figure 8-5 and Figure 8-6.

Table 8-18 Processing on Receive Frames and Responses

Receive frame		Processing	Response
No frame		None	None
Frame addressed to others		Discard	None
Broadcast query		Supported processing	None
Self-addressed frame	CRC error	Discard	None
	Unsupported query	Discard	Exception response
	Normal	Supported processing	Response

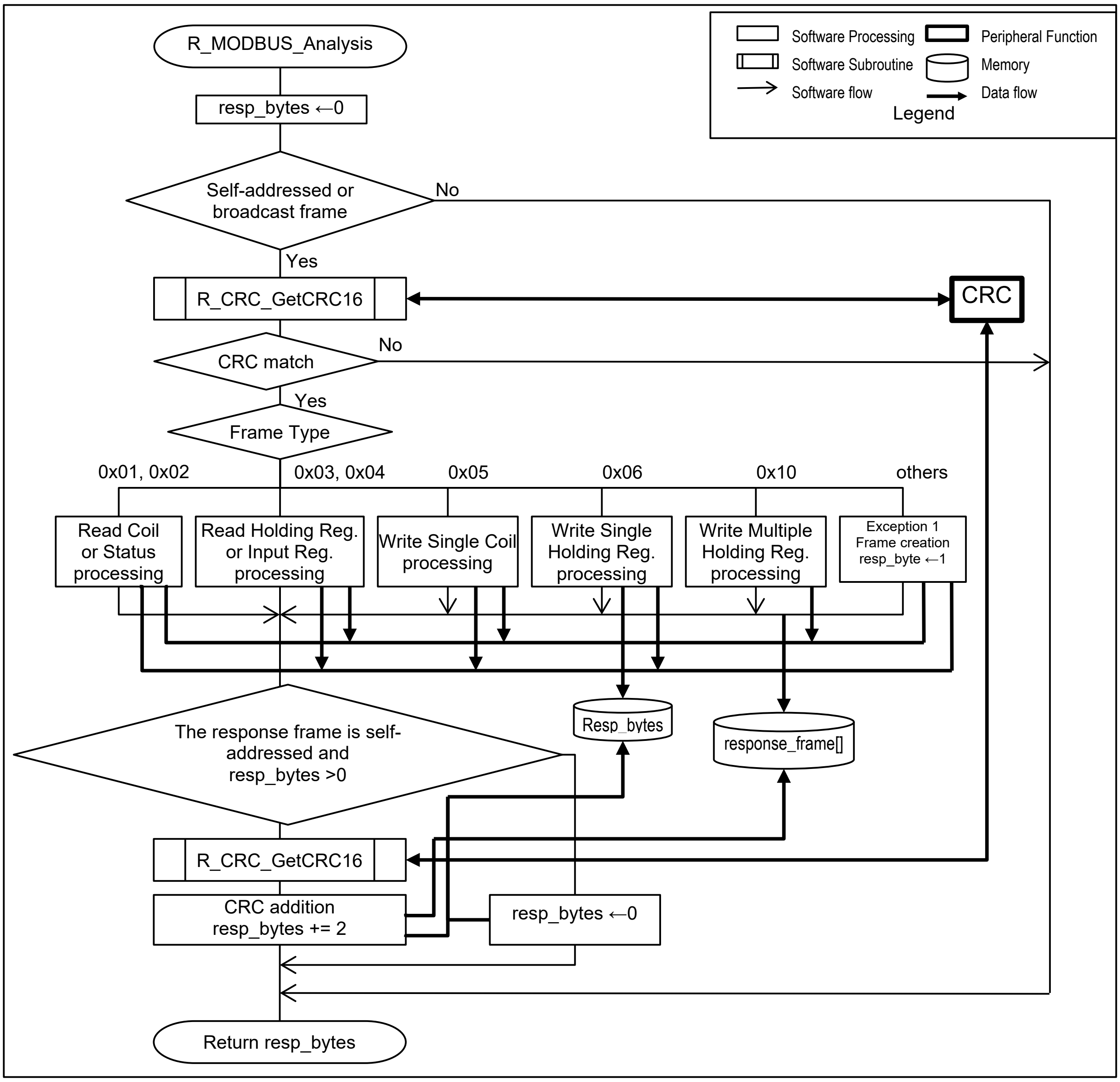


Figure 8-5 Modbus Receive Frame Processing Flow (1)

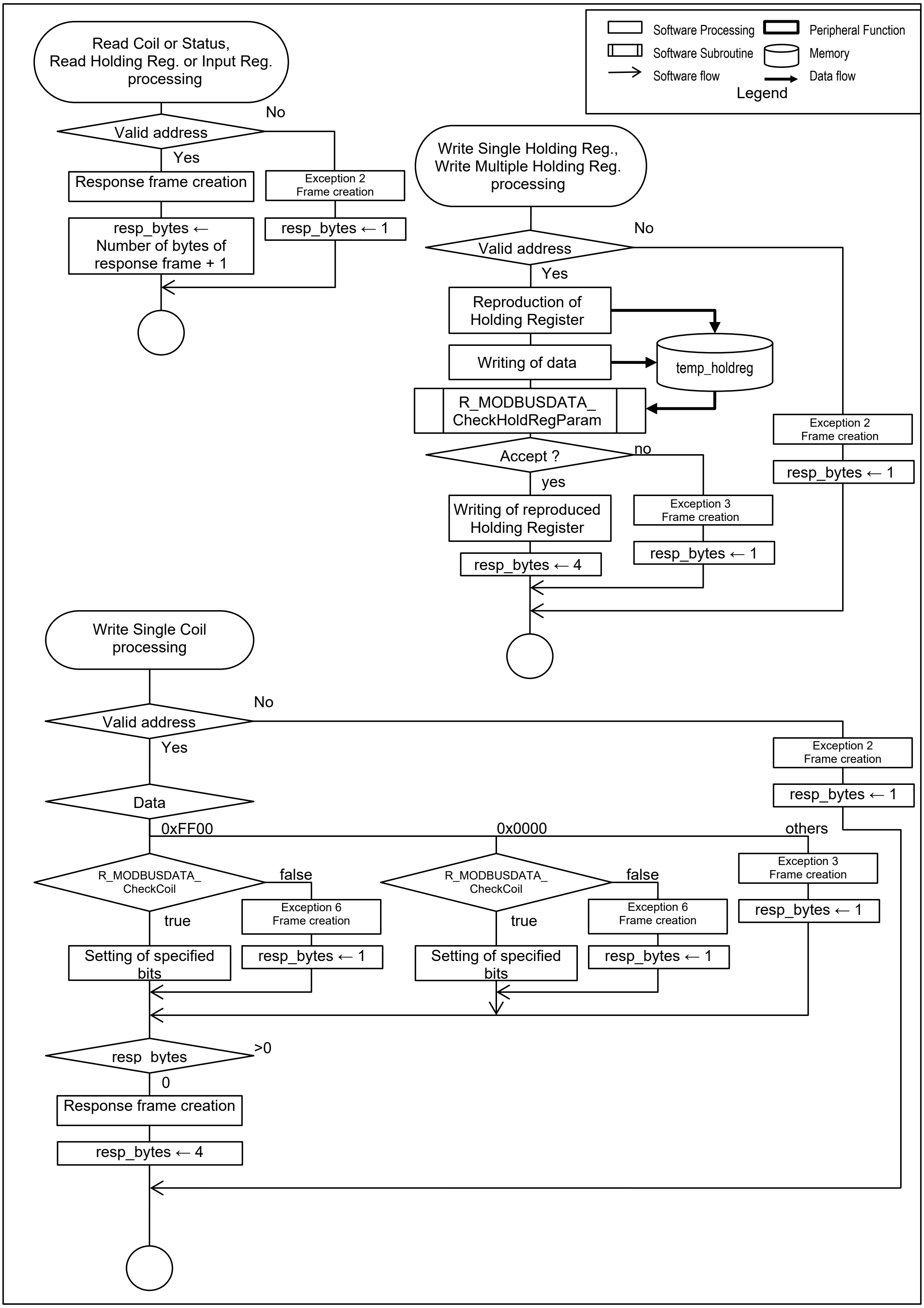


Figure 8-6 Modbus Receive Frame Processing Flow (2)

8.4 Program Configuration

8.4.1 Source File Configuration

Table 8-19 File Configuration

Folder name, file name	Description	
	QE for AFE version	Modbus version
src		
├ smc_gen	Generated by Smart Configurator	
│ ├ general	Generated by Smart Configurator	Generated by Smart Configurator
│ ├ r_bsp		
│ ├ r_config		
│ └ r_pincfg		
│ ├ Config_DSAD0	Load cell measurement setting	
│ ├ Config_SCI1	QE for AFE communication	Modbus communication
│ ├ Config_DMAC0		-
│ ├ Config_DMAC1		
│ ├ Config_DTC_RXI	-	Modbus communication
│ ├ Config_DTC_TXI		
│ ├ Config_DTC_CMIA0		
│ ├ Config_TMR0		
│ ├ Config_PORT	LED1, RS-485 transmit/receive switching setting	
│ ├ r_flash_rx	Flash API	
├ main.c	Main function	
├ r_calc_api.c	Calculations such as average processing and moving average processing	
├ r_calc_api.h		
├ r_loadcell_cfg.h	Load cell weight conversion processing	
├ r_loadcell_api.c		
├ r_loadcell_api.h		
├ r_modbusdata_api.c	Modbus data processing	
├ r_modbusdata_api.h		
├ r_modbusdata_cfg.h		
├ r_qe_cfg.h	QE for AFE communication module	-
├ r_qe_cfg_typedef.h		
├ r_qe_packet.h		
├ r_qe_sc_if.h		
├ r_qe_api.c		
├ r_qe_api.h		
├ r_qe_api_user.c		
├ r_ring_buffer_control_api.c		
├ r_ring_buffer_control_api.h		
├ r_modbus_cfg.c	-	Modbus communication process
├ r_modbus_api.c		
└ r_modbus_api.h		

8.4.2 Build Settings

There are two types of sample projects, QE for AFE version and Modbus version, as shown in Table 2-1. Additional settings for each sample project are listed in Table 8-20.

Table 8-20 Build Settings for Sample Projects

<table>
<tr><th>Project name</th><th colspan="2">rx23eb_loadcell_qe</th><th colspan="2">rx23eb_loadcell_modbus</th></tr>
<tr><td>Additional definition</td><td colspan="2">-define D_CFG_QE_TOOL_USE</td><td colspan="2">None</td></tr>
<tr><td rowspan="3">Additional section definition</td><td>Address</td><td>Section name</td><td>Address</td><td>Section name</td></tr>
<tr><td>0x00100000</td><td>C_DATAFLASH_1</td><td colspan="2">Same as the left</td></tr>
<tr><td>0x00003000</td><td>B_DMAC_REPEAT_AREA_1</td><td colspan="2">-</td></tr>
</table>

8.4.3 Macro Definitions

Table 8-21 r_modbusdata_cfg.h Definitions

Definition name	value	Description
D_MODBUSDATA_CFG _ZERO_WEIGHT_DEFAULT	0.0F	Initial value of zero reset weight
D_MODBUSDATA_CFG _MOVING_AVERAGE_NUM_DEFAULT	8	Initial value of moving averaging count for weight measurement processing
D_MODBUSDATA_CFG _ZERORESET_AVERAGE_DEFAULT	256	Initial value of averaging count for zero reset processing
D_MODBUSDATA_CFG _CALIB_AVERAGE_DEFAULT	256	Initial value of A/D value averaging count for calibration processing
D_MODBUSDATA_CFG _MOVING_AVERAGE_MAX	128	Maximum moving averaging count
D_MODBUSDATA_CFG _MOVING_AVERAGE_MIN	1	Minimum moving averaging count
D_MODBUSDATA_CFG _ZERO_RESET_AVERAGE_MAX	512	Maximum averaging count for zero reset processing
D_MODBUSDATA_CFG _ZERO_RESET_AVERAGE_MIN	1	Minimum averaging count for zero reset processing
D_MODBUSDATA_CFG _CALIBRATION_AVERAGE_MAX	512	Maximum A/D value averaging count for calibration processing
D_MODBUSDATA_CFG _CALIBRATION_AVERAGE_MIN	64	Minimum A/D value averaging count for calibration processing
D_MODBUSDATA_CFG _CALIBRATION_DELAY	5.0F	Delay time [s] before starting Calibration 1

Table 8-22 r_roadcell_cfg.h Definitions

Definition name	value	Description
D_LC_CFG_PGA_GAIN	128.0F	PGA gain G_{PGA} in DSAD0
D_LC_CFG_DSADRES	24	Number of bits of A/D converter
D_LC_CFG_VREF	5.0F	DSAD0 reference voltage V_{REF}
D_LC_CFG_VCC	5.0F	Applied voltage V_{CC} to load cell
D_LC_CFG_RO	0.0009F	Rated output RO [V/V] of load cell
D_LC_CFG_MMAX	300.0F	Rated load MMAX [g] of load cell

Note: The initial values of the weight conversion coefficients are defined from the definitions above.

Table 8-23 r_modbus_cfg.h Definition (Modbus Version)

Definition name	value	Description
D_MODBUS_CFG_ADDRESS	0x01	Modbus slave address

Table 8-24 r_qe_cfg.h Settings (QE for AFE Version)

Definition name	Value	Description
D_QE_CFG_TX_RINGBUF_SIZE	512U	Transmission ring buffer size [bytes]
D_QE_CFG_RX_RINGBUF_SIZE	512U	Reception ring buffer size [bytes]
D_QE_CFG_FORMAT_REV	3	Communication specifications revision
D_QE_CFG_READ	1	Register read permission
D_QE_CFG_WRITE	1	Register write permission
D_QE_CFG_USER_VAL0	1	User Value setting 0: Not used 1: Used
D_QE_CFG_USER_VAL1	1	
D_QE_CFG_USER_VAL2	1	
D_QE_CFG_USER_VAL3	1	
D_QE_CFG_USER_VAL4	0	
D_QE_CFG_USER_VAL5	0	
D_QE_CFG_USER_VAL6	0	
D_QE_CFG_USER_VAL7	0	
D_QE_CFG_EX_SPS	1	SPS information support 0: Not used 1: Used
D_QE_CFG_EX_USER_BTN0	1	User Button use setting 0: Not used 1: Used
D_QE_CFG_EX_USER_BTN1	1	
D_QE_CFG_EX_USER_BTN2	0	
D_QE_CFG_EX_USER_BTN3	0	
D_QE_CFG_EX_USER_BTN4	0	
D_QE_CFG_EX_USER_BTN5	0	
D_QE_CFG_EX_USER_BTN6	0	
D_QE_CFG_EX_USER_BTN7	0	
D_QE_CFG_CH0	0x3	Data transmission CH use setting 0x3: Measurement value transmission 0x0: Not used
D_QE_CFG_CH1	0x0	
D_QE_CFG_CH2	0x0	
D_QE_CFG_CH3	0x0	
D_QE_CFG_CH4	0x0	
D_QE_CFG_CH5	0x0	
D_QE_CFG_CH6	0x0	
D_QE_CFG_CH7	0x0	
D_QE_CFG_CH8	0x0	
D_QE_CFG_CH9	0x0	
D_QE_CFG_CH10	0x0	
D_QE_CFG_CH11	0x0	
D_QE_CFG_CH12	0x0	
D_QE_CFG_CH13	0x0	
D_QE_CFG_CH14	0x0	
D_QE_CFG_CH15	0x0	
D_QE_CFG_TXT_INFO	"RX23E-B loadcell measurement.”	Program information
D_QE_CFG_TXERRCHK_EN	0	Transmission error detection enabled
D_QE_CFG_TIMEOUT	0	Error is detected when timeout is reached
D_QE_CFG_SCI	0	SCI channel number used for communication
D_QE_CFG_DMAC_RX	0	DMAC channel number for reception
D_QE_CFG_DMAC_TX	1	DMAC channel number for transmission
D_QE_CFG_CMT	0	CMT channel number for timeout detection

8.4.4 Structures, Unions, and Enumeration Types

Table 8-25 r_loadcell_api.h List

Structure type name	st_lc_coef_t		
Description	Weight conversion coefficients		
Member	**Type**	**Name**	**Description**
	float	a	Coefficient a (slope)
	float	b	Coefficient b (intercept)
Structure type name	st_lc_calibration_data_t		
Description	Calibration data		
Member	**Type**	**Name**	**Description**
	float	weight[2]	Weights for measurement (2 different weights)
	float	adval[2]	Acquired A/D values (2 different values)

Table 8-26 r_calc_api.h List

Structure type name	st_calc_moveavg_data_t		
Description	Moving averaging parameters		
Member	**Type**	**Name**	**Description**
	int32_t	count	Number of acquired data
	float	sumdata	Total value of acquired data
	float *	p_deldata	Pointer to acquired data storage array
	int32_t	avgnum	Moving averaging count
Structure type name	st_calc_average_data_t		
Description	Averaging parameters		
Member	**Type**	**Name**	**Description**
	uint32_t	num	Averaging count
	uint32_t	count	Number of acquired data
	float	sum	Total value of acquired data

Table 8-27 r_modbusdata_api.h List (1/3)

Union type name	u_modbus_float_t		
Description	float type Modbus data		
Member	**Type**	**Name**	**Description**
	float	float32	float type
	uint32_t	uint32	uint32 type
	uint16_t	word[2]	uint16 type
	uint8_t	byte[4]	uint8 type
Union type name	u_modbus_long_t		
Description	int32_t type Modbus data		
Member	**Type**	**Name**	**Description**
	int32_t	int32	int32 type
	uint16_t	word[2]	uint16 type
	uint8_t	byte[4]	uint8 type
Union type name	u_modbus_ulong_t		
Description	uint32_t type Modbus data		
Member	**Type**	**Name**	**Description**
	uint32_t	int32	uint32 type
	uint16_t	word[2]	uint16 type
	uint8_t	byte[4]	uint8 type
Union type name	u_modbus_ushort_t		
Description	uint16_t type Modbus data		
Member	**Type**	**Name**	**Description**
	uint16_t	word	uint16 type
	uint8_t	byte[2]	uint8 type

Table 8-28 r_modbusdata_api.h List (2/3)

Enumeration type name	e_opemode_t		
Description	Processing mode		
Member	**Name**	**Value**	**Description**
	E_IDLE	0	Standby
	E_MEASUREMENT	1	Measurement
	E_CALIBRATION1	2	Calibration STEP1
	E_CALIBRATION2	4	Calibration STEP2
Union type name	u_modbusdata_coil_t		
Description	Modbus Coil		
Member	**Type**	**Name**	**Description**
	uint32_t	uint32	Entire data
	Union	bit	Access in bit units
	uint32_t:3	opemode	Operating mode bit group
	struct	flag	Each bit
	uint32_t:1	measure	Measurement mode bit
	uint32_t:1	calib1	Calibration STEP1 bit
	uint32_t:1	calib2	Calibration STEP2 bit
	uint32_t:1	zero_reset	Zero reset bit
	uint32_t:1	reset_param	Parameter initialization bit
	uint32_t:1	register_write	Register rewriting bit (for QE for AFE)
Union type name	u_modbusdata_status_t		
Description	Modbus Status		
Member	**Type**	**Name**	**Description**
	uint32_t	uint32	Entire data
	union	status	Access in bit units
	struct	bit	Each bit
	uint32_t:1	dsad_ovf	DSAD0 Overflow bit
	uint32_t:1	dsad_err	DSAD0 Error bit
	uint32_t:1	cal1_end	Notification of the end of Calibration STEP1
Union type name	u_modbusdata_inputreg_t		
Description	Modbus Input register		
Member	**Type**	**Name**	**Description**
	uint16_t	reg[4]	Access in register units
	struct	member	Each register definition
	u_modbus_float_t	weight	Measured weight
	u_modbus_long_t	dsad	A/D value
	struct	params	Internal access definition
	float	weight	Measured weight
	int32_t	dsad	A/D value

Table 8-29 r_modbusdata_api.h List (3/3)

Structure type name	st_prm_t		
Description	Retained parameter		
Member	**Type**	**Name**	**Description**
	st_lc_coef_t	coef	Weight conversion coefficients
	uint16_t	moving_average	Moving averaging count for measurement
	uint16_t	zero_reset_average	Averaging count for zero reset
	uint16_t	reg_cr0_gain	CR0.GAIN register value
	uint16_t	reg_mr0_fsel	MR0.FSEL register value
	uint32_t	reg_osr0	OSR0 register value
	uint32_t	reg_sgcr0	SGCR0 register value
	uint16_t	calib_average	Averaging count for calibration
Union type name	u_modbusdata_holdreg_t		
Description	Modbus Holding register		
Member	**Type**	**Name**	**Description**
	uint16_t	reg[19]	Access in register units
	struct	member	Each register definition
	u_modbus_float_t	zero_weight	Zero reset weight
	u_modbus_float_t	calib_weight1	Measured weight in calibration STEP1
	u_modbus_float_t	calib_weight2	Measured weight in calibration STEP2
	u_modbus_float_t	calib_delay	Delay time before starting Calibration1 [s]
	u_modbus_float_t	coef_a	Weight conversion coefficient a
	u_modbus_float_t	coef_b	Weight conversion coefficient b
	u_modbus_ushort_t	moving_average	Moving averaging count for measurement
	u_modbus_ushort_t	zero_reset_average	Averaging count for zero reset
	u_modbus_ushort_t	reg_cr0_gain	CR0.GAIN register value
	u_modbus_ushort_t	reg_mr0_fsel	MR0.FSEL register value
	u_modbus_ulong_t	reg_osr0	OSR0 register value
	u_modbus_ulong_t	reg_sgcr0	SGCR0 register value
	u_modbus_ushort_t	calib_average	Averaging count for calibration
	struct	params	Internal access definition
	float	zero_weight	Zero reset weight
	float	calib_weight1	Measured weight in calibration STEP1
	float	calib_weight2	Measured weight in calibration STEP2
	float	calib_delay	Delay time before starting Calibration1 [s]
	st_prm_t	prm	Retained parameter

8.4.5 Functions

8.4.5.1 Common Functions

Table 8-30 main.c

Function name	**main**			
Description	main function			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	

Table 8-31 r_calc_api

Function name	**R_CALC_MovingAverage**			
Description	Calculates the average value for the specified moving averaging count			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	const float	data	Input value
	I/O	st_calc_moveavg_data_t *	p_cal_moveavg	Pointer to the moving averaging parameter
Return value	O	float	Moving average value	
Function name	**R_CALC_MovingAverageReset**			
Description	Resets the moving averaging parameters			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_calc_moveavg_data_t *	p_cal_moveavg	Pointer to the moving averaging parameter
	I	int32_t	average_num	Moving averaging count
Return value	-	void	-	
Function name	**R_CALC_Average**			
Description	Calculates the average value for the specified averaging count			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	float	input	Input value
	I/O	st_calc_average_data_t *	average	Pointer to the averaging parameter
	O	float *	result	Pointer to the averaging result storage destination
Return value	O	bool	true: Averaging completed false: Not completed	
Function name	**R_CALC_AverageInit**			
Description	Initializes the averaging parameters			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_calc_average_data_t *	average	Pointer to the averaging parameter
	I	uint32_t	num	Averaging count
Return value	-	void	-	

Table 8-32 r_loadcell_api

Function name	**R_LC_DsadToWeight**			
Description	Converts an A/D value to weight using the weight conversion coefficient			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	float	dsad	A/D value
	I	st_lc_coef_t *	coef	Pointer to the weight conversion coefficient
Return value	O	float	Weight	
Function name	**R_LC_CalcCoef**			
Description	Calculates the weight conversion coefficient			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	st_lc_calibration_data_t *	prm	Pointer to the weight conversion calibration parameter
	O	st_lc_coef_t *	coef	Pointer to the weight conversion coefficients
Return value	O	bool	true: Successful false: Failed	

Table 8-33 r_modbusdata_api (1/2)

Function name	**R_MODBUSDATA_GetCoilPtr**			
Description	Acquires a pointer to Modbus Coil			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	u_modbusdata_coil_t *	Pointer to Modbus Coil	
Function name	**R_MODBUSDATA_GetStatusPtr**			
Description	Acquires a pointer to Modbus Status			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	u_modbusdata_status_t *	Pointer to Modbus Status	
Function name	**R_MODBUSDATA_GetInputRegPtr**			
Description	Acquires a pointer to the Modbus Input register			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	u_modbusdata_inputreg_t *	Pointer to the Modbus Input register	
Function name	**R_MODBUSDATA_GetHoldRegPtr**			
Description	Acquires a pointer to the Modbus Holding register			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	u_modbusdata_holdreg_t *	Pointer to the Modbus Holding register	
Function name	**R_MODBUSDATA_GetResetParam**			
Description	Acquires and retains the initial value of each of the CR0.GAIN, MR0.FSEL, OSR0, and SGCR0 of DSAD0			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	

Table 8-34 r_modbusdata_api (2/2)

Function name	**R_MODBUSDATA_LoadHoldReg**			
Description	Initializes the Modbus Holding Register and loads the value stored in E2 data flash			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_MODBUSDATA_SaveHoldReg**			
Description	If the value retained in the Modbus Holding Register does not match the value stored in E2 data flash, stores that value in E2 data flash			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_MODBUSDATA_ResetHoldReg**			
Description	Sets the Modbus Holding Register to the initial value			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_MODBUSDATA_CheckCoil**			
Description	Judges whether it is possible to set and clear the specified address of Coil			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint16_t	addr	Coil address
	I	bool	flag	true: Set false: Clear
Return value	O	bool	true: Possible false: Not possible	
Function name	**R_MODBUSDATA_CheckHoldRegParam**			
Description	Judges the acceptability of the Holding register value			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	u_modbusdata_holdreg_t *	p_holdreg	Pointer to the Holding Register union variable to be judged
Return value	O	bool	true: Acceptable false: Unacceptable	

Table 8-35 r_led_api

Function name	**R_LED_Set**			
Description	Specifies whether to turn on, turn off or blink LED1			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	flag	true: ON false: OFF
	I	int32_t	count	Blinking count 0: Does not blink >0: Blinking count -1: Blinks without specified count
Return value	-	void	-	
Function name	**R_LED_BlinkControl**			
Description	Controls blinking of LED1			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_LED_IsBlink**			
Description	Acquires whether LED1 is blinking			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Blinking false: Not blinking	

Table 8-36 Config_DSAD0 User Defined Functions (1/2)

Function name	**R_DSAD0_IsConversionEnd**			
Description	Detects A/D conversion end (ADI0)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Detected false: Not detected	
Function name	**R_DSAD0_ConvSignedValue**			
Description	Acquires a signed A/D value (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	val	Acquired DR register value
Return value	O	int32_t	Signed A/D value	
Function name	**R_DSAD0_GetErrorFlag**			
Description	Extracts the ERR flag from the acquired DR register value (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	val	Acquired DR register value
Return value	O	uint32_t	DR.ERR flag	
Function name	**R_DSAD0_GetOverflowFlag**			
Description	Extracts the OVF flag from the acquired DR register value (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	val	Acquired DR register value
Return value	O	uint32_t	DR.OVF flag	

Table 8-37 Config_DSAD0 User Defined Functions (2/2)

Function name	**R_Config_DSAD0_GetScanRate**			
Description	Calculates the A/D conversion rate from the DSAD0 CH0 setting			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	float	A/D conversion rate [SPS]	
Function name	**R_Config_DSAD0_GetParam**			
Description	Acquires the setting value of each register of the specified CH of DSAD0			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	ch	Channel m to be read
	O	uint16_t *	gain	Pointer to CRm.GAIN setting value storage destination
	O	uint16_t *	fsel	Pointer to MRm.FSEL setting value storage destination
	O	uint32_t *	osr	Pointer to OSRm setting value storage destination
	O	uint32_t *	sgcr	Pointer to SGCRm setting value storage destination
Return value	-	void	-	
Function name	**R_Config_DSAD0_SetParam**			
Description	Sets a value to each register of the specified CH of DSAD0			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	ch	Channel m to which to set a value
	I	uint16_t	gain	CRm.GAIN configuration value
	I	uint16_t	fsel	MRm.FSEL configuration value
	I	uint32_t	osr	OSRm configuration value
	I	uint32_t	sgcr	SGCRm configuration value
Return value	-	void	-	
Function name	**R_DSAD0_GetData**			
Description	Acquires the DR register value			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	uint32_t	DR register value	

Table 8-38 Config_PORT User Defined Functions

Function name	**R_Config_PORT_LEDON**			
Description	Turns on and off LED1			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	flag	true: ON false: OFF
Return value	-	void	-	
Function name	**R_Config_PORT_LED_Blink**			
Description	Reverses the ON/OFF of LED1 (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_CONFIG_PORT_SET_DE**			
Description	Sets the transmission or reception of the RS-485 driver (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	uint8_t	value	0: Reception 1: Transmission
Return value	-	void	-	

Table 8-39 Config_CMT1 User Defined Function

Function name	**R_CMT1_IsCompareMatch**			
Description	Detects CMT1 compare match			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Compare match detected false: Compare match not detected	

8.4.5.2 QE for AFE Version

Table 8-40 r_qe_api_user.c User-Defined Processes

User processes only

Function name	**r_QE_WriteUser**
Description	If opemode is E_IDLE, accepts and sets coil.flag.register_write
Function name	**r_QE_RunUser**
Description	If opemode is E_IDLE, accepts and sets coil.flag.measure
Function name	**r_QE_StopUser**
Description	Clear coil.flag.measure
Function name	**r_QE_UserValueUser** Note
Description	Judged to be accepted or not for each User Value No. and if accepted, updates the value of the corresponding Coil or Holding register as appropriate
Function name	**r_QE_ExSpsInfoUser**
Description	Calculates the output rate from the DSAD0 setting and updates SPS information
Function name	**r_QE_ExUseButtonStatusUser** Note
Description	Judged to be accepted or not for each Button No. and if accepted, sets the flag for the corresponding Coil
Function name	**r_QE_ResetUser**
Description	Sets the RS-485 driver to the receive status (DE = L)

Note: For details about each of the corresponding QE for AFE functions, refer to Table 1-1.

Table 8-41 r_qe_api.c Processing Change

Change only

Function name	**R_QE_TxStart**
Description	Sets the RS-485 driver to the transmit status (DE = H) at the start of transmission

8.4.5.3 Modbus Version

Table 8-42 r_modbus_api

Function name	**R_MODBUS_Analysis**			
Description	Inspects and analyzes the receive frame, processes the corresponding Modbus data, and creates a transmission frame			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	const uint8_t *	QueryFrame	Pointer to the receive frame
	I	uint32_t	QueryBytes	Number of bytes of the receive frame
	O	uint8_t *	ResponseFrame	Pointer to the destination to stores transmission frame
Return value	O	uint32_t	Number of transmit frame bytes	

Table 8-43 Config_CRC User Defined Function

Function name	**R_CRC_GetCRC16**			
Description	Calculates CRC-16			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	array	Pointer to the target array
	I	uint32_t	num	Number of target bytes
Return value	O	uint16_t	CRC-16 value	

Table 8-44 Config_TMR0 User Defined Functions

Function name	**R_Config_TMR0_SetCMIA0** **R_Config_TMR0_SetCMIB0**			
Description	Sets CMIx0 interrupt enable/disable			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	enable	true: Enable false: Disable
Return value	-	void	-	
Function name	**R_Config_TMR0_ClearCount**			
Description	Clears the timer count value			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_Config_TMR0_StartCount**			
Description	Starts the timer count			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_Config_TMR0_StopCount**			
Description	Stops the timer count			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	-	
Function name	**R_TMR0_IsCMIA0**			
Description	Detects a CMIA0 interrupt request			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Detected false: Not detected	

Table 8-45 Config_DTC_RXI1 User Defined Functions

Function name	**R_Config_DTC_RXI1_SetDstAddr**			
Description	Sets the destination address of DTC transfer			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	number	Chain transfer number
	I	void *	addr	Destination address
Return value	-	void	-	
Function name	**R_Config_DTC_RXI1_SetSrcAddr**			
Description	Sets the source address of DTC transfer			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	number	Chain transfer number
	I	void *	addr	Source address
Return value	-	void	-	
Function name	**R_Config_DTC_RXI1_GetCraAddr**			
Description	Acquires the CRA address in DTC transfer information			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	number	Chain transfer number
Return value	O	void *	CRA address	
Function name	**R_Config_DTC_RXI1_GetDarAddr**			
Description	Acquires the DAR address in DTC transfer information			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	number	Chain transfer number
Return value	O	void *	DAR address	

Table 8-46 Config_DTC_TXI1 User Defined Functions

Function name	**R_Config_DTC_TXI1_SetCount**			
Description	Sets the destination address of DTC transfer			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	count	Number of transfer bytes
Return value	-	void	-	
Function name	**R_Config_DTC_TXI1_SetSrcAddr**			
Description	Sets the source address of DTC transfer			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	void *	addr	Source address
Return value	-	void	-	

Table 8-47 Config_DTC_CMIA0 User Defined Functions

Function name	**R_Config_DTC_CMIA0_SetDstAddr**			
Description	Sets the destination address of DTC transfer			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	number	Chain transfer number
	I	void *	addr	Destination address
Return value	-	void	-	
Function name	**R_Config_DTC_CMIA0_SetSrcAddr**			
Description	Sets the source address of DTC transfer			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	number	Chain transfer number
	I	void *	addr	Source address
Return value	-	void	-	

9. Importing a Project

After importing the sample project, make sure to confirm build and debugger setting.

9.1 Importing a Project into e2 studio

Follow the steps below to import your project into e^2 studio. Pictures may be different depending on the version of e^2 studio to be used.

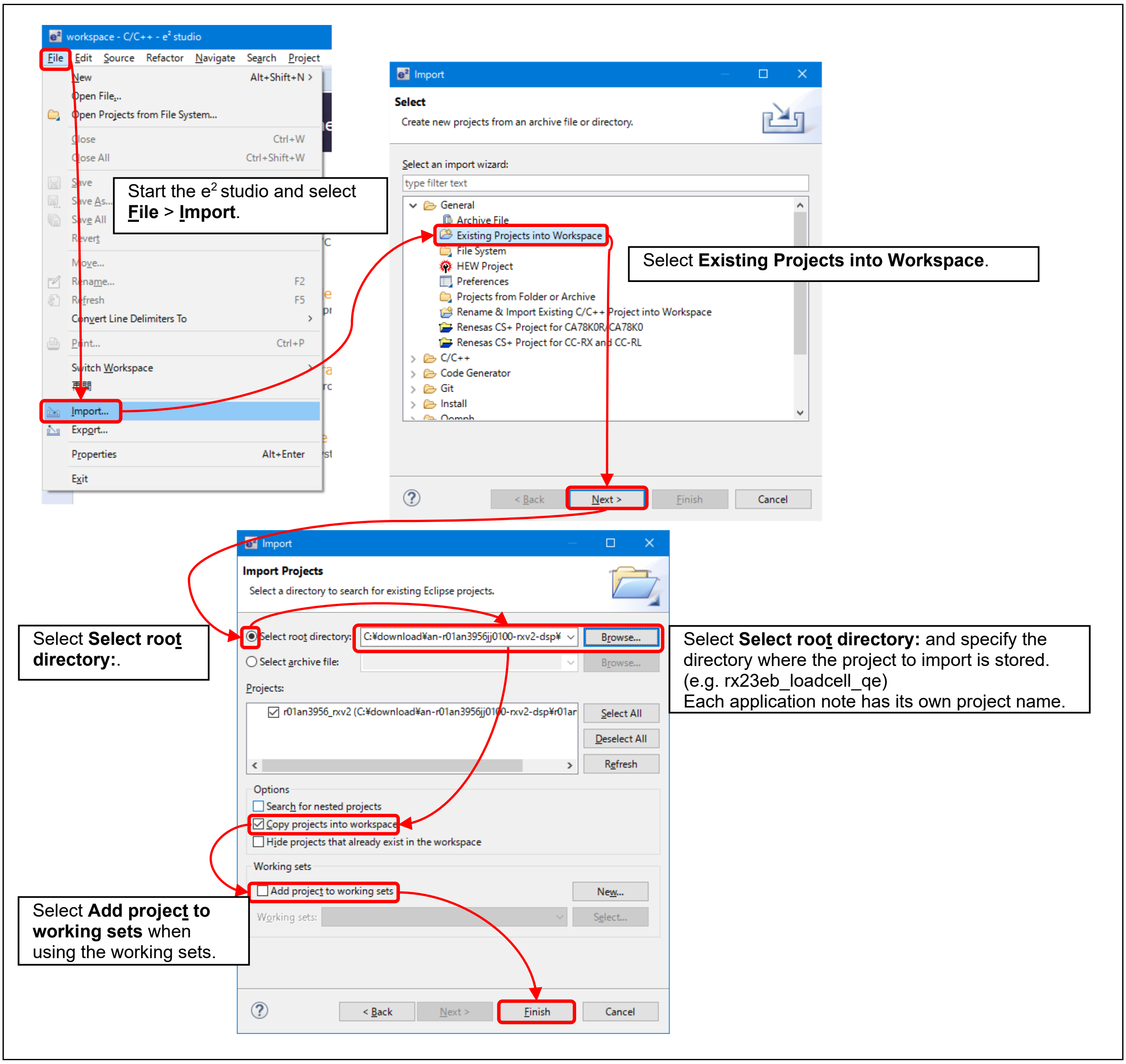


Figure 9-1 Importing a project into e^2 studio

9.2 Importing a Project into CS+

Follow the steps below to import your project into CS+. Pictures may be different depending on the version of CS+ to be used.

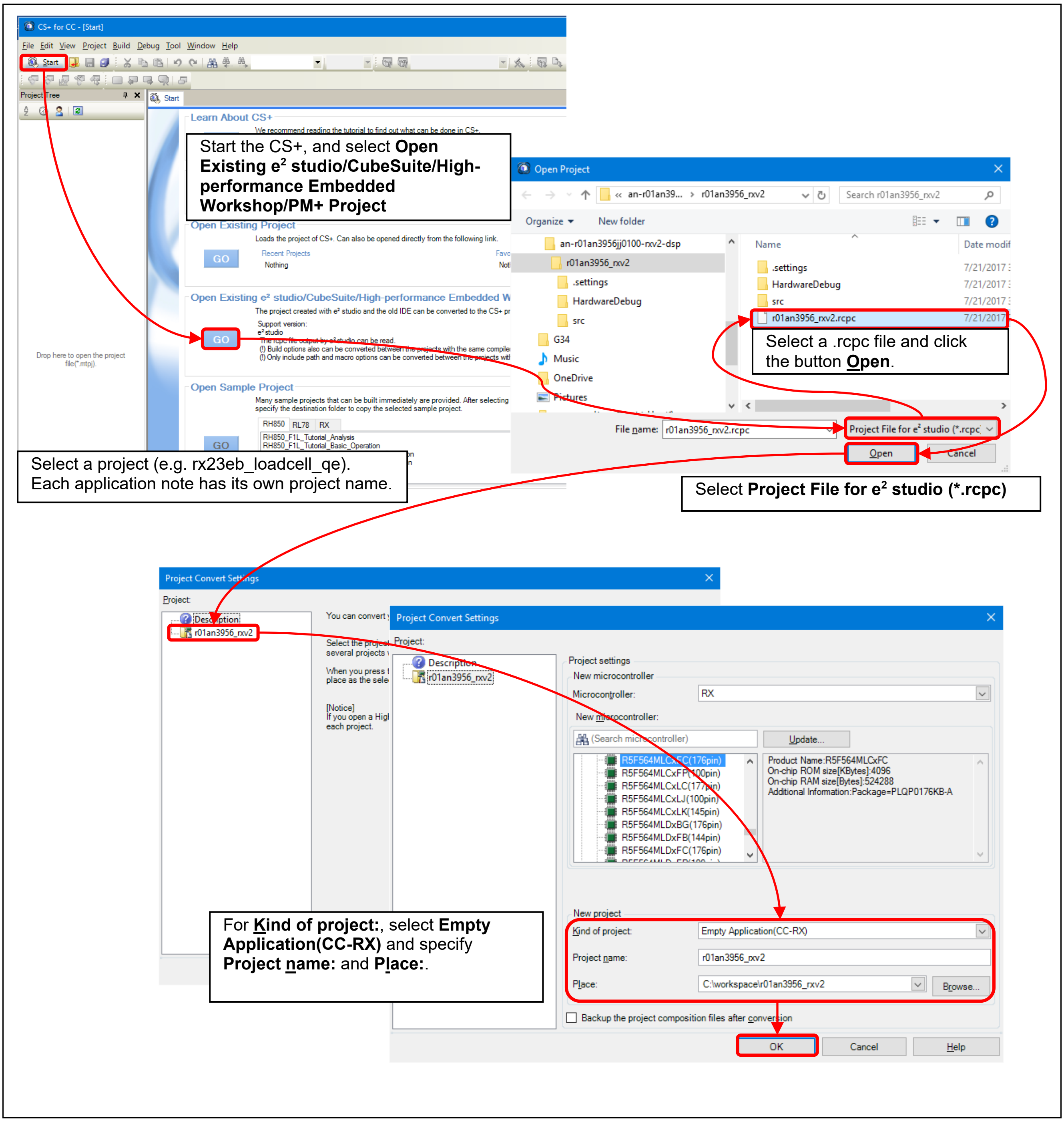


Figure 9-2 Importing a project into CS+

10. Operation on the Renesas Solution Starter Kit for RX23E-B Board

The Renesas Solution Starter Kit for RX23E-B (referred to as RSSKRX23E-B) is offered for the evaluation of RX23E-B. The RSSKRX23E-B board contains the R5F523E6LxFP in the QFP100 package and supports various sensor measurements including load cell measurement.

This sample program can be run on the RSSKRX23E-B board by changing settings with the Smart Configuration according to "2.4.7 Load Cell Measurement Circuit in RSSKRX23E-B User's Manual". Communication can be conducted via the USB-UART conversion IC.

The following explains the procedure.

(1) Change the device type name
Click the button next to "Board" on the "Board" tab, and then change "Target device" to R5F523E6LxFP then uncheck the "Sections (-start)" in HardwareDebug_RSSK on "Changes to be performed" in the "Change Device" dialog.

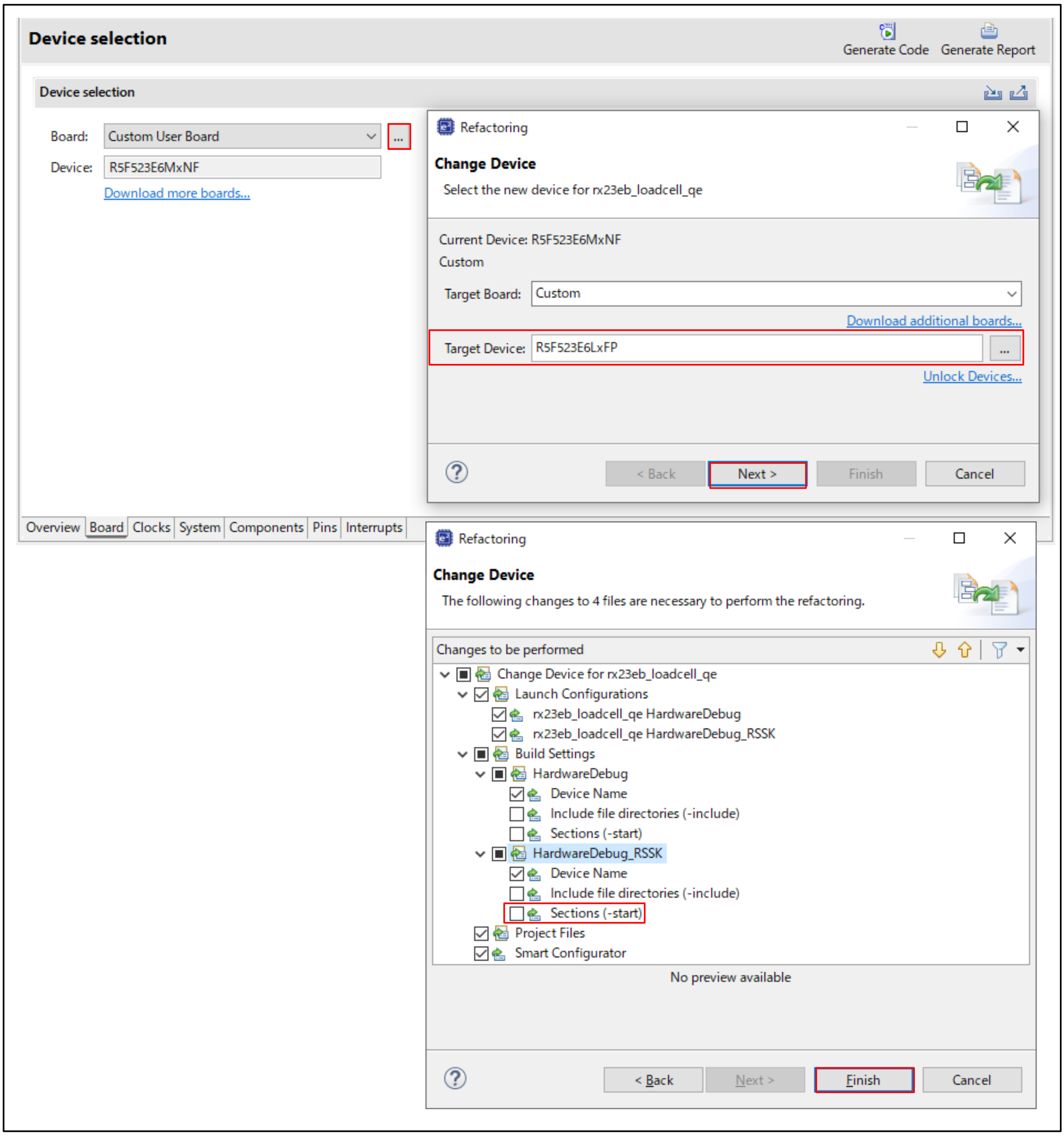


Figure 10-1 Changing the Target Device

(2) Change the DSAD0 setting

Change the analog input setting of channel 0 to the setting listed in Table 10-1 for "Config_DSAD0" on the "Component" tab, and then generate code. Bold text in the table indicates changes.

Table 10-1 Changing the DSAD0 Setting

Continuous scan mode

Item		**Setting**	
		RX23E-B-QFN40-WT	**RSSKRX23E-B**
Channel setting		0	
Analog input setting	Positive input signal	AIN5	**AIN15**
	Negative input signal	AIN4	**AIN14**
	Reference voltage	REF0P/REF0N	**REF1P/REF1N**
	Positive reference voltage buffer	Disable	
	Negative reference voltage buffer	Disable	

(3) Build

Change the active build configuration to “HardwareDebug_RSSK”, then build.
To run the product on the RSSKRX23E-B board, pin settings have been changed using the R_Config_PORT_Create_UserInit function in Config_PORT_user.c, according to Table 10-2. In addition, assignment of LED port is changed using Config_PORT.h.

Table 10-2 RSSKRX23E-B Changes to Pin Settings

RX23E-B-QFN40-WT	**RSSKRX23E-B board**			
Pin: Function	**Assignment**	**Pin**	**Initial setting**	**Supplement**
P27: Switches the setting of RS-485 driver between transmit/receive	DE	PC6	Output: L	RAA788158/DE
P31: LED1	LED0	P70	Output: H	The P31 setting is retained.
-	LED1	P71	Output: H	
-	LED2	P72	Output: H	
-	LED3	P73	Output: H	
-	SW1	PE1	Input	
-	SW2	PE2	Input	
-	SW3-1	PE3	Input	
-	SW3-2	PE4	Input	
-	XTAL	P36,P37	Peripheral function	XTAL is unused.
-	-	P15	Input	CTS1# input
-	-	PC1	Input	MAX13053/RXD input
TXD1/P26: RS-485 data output	TXD1	P26	Peripheral function	Pull-up is canceled.

Note: Set I/O ports not listed above to output: L.

11. Measurement Results with Sample Program

11.1 Memory Usage and Number of Execution Cycles

11.1.1 Build Conditions

The build conditions for the sample program are listed in Table 11-1.

Table 11-1 Build Conditions

Item		Setting	
		QF for AFE	Modbus
Compiler	Common	-isa=rxv2 -fpu -utf8 -nomessage -output=obj -obj_path=${workspace_loc:/${ProjName}/${ConfigName}} -debug -outcode=utf8 -nologo	
	Difference	-define=D_CFG_QE_TOOL_USE	
Linker	Common	-noprelink -form=absolute -nomessage -vect=_undefined_interrupt_source_isr -nooptimize -rom=D=R,D_1=R_1,D_2=R_2 -cpu=RAM=00000000-00007fff, FIX=00080000-00083fff,FIX=00086000-00087fff,FIX=00088000-0008dfff, FIX=00090000-0009ffff,FIX=000a0000-000bffff,FIX=000c0000-000fffff, ROM=00100000-00101fff,FIX=007fc000-007fc4ff,FIX=007ffc00-007fffff, ROM=fffc0000-ffffffff -nologo	
	Difference	-output="rx23eb_loadcell_qe.abs" -list=rx23eb_loadcell_qe.map	-output="rx23eb_loadcell_modbus.abs" -list=rx23eb_loadcell_modbus.map
	Section	SU,SI,B_1,R_1,B_2,R_2,B,R/04, B_DMAC_REPEAT_AREA_1/03000, C_DATAFLASH_1/0100000, PResetPRG,C_1,C_2,C,C$*,D*,W*,L, P/0FFFC0000,EXCEPTVECT/0FFFFFF80, RESETVECT/0FFFFFFFC	SU,SI,B_1,R_1,B_2,R_2,B,R/04, C_DATAFLASH_1/0100000, PResetPRG,C_1,C_2,C,C$*,D*,W*,L, P/0FFFC0000,EXCEPTVECT/0FFFFFF80, RESETVECT/0FFFFFFFC

11.1.2 Memory Usage

The amount of memory usage of the sample program is shown in Table 11-2.

Table 11-2 Amount of Memory Usage

Item		Size (byte)		Remarks
		QF for AFE	Modbus	
ROM		11795	11574	
	Code	9860	9704	
	Data	1935	1870	
E2 DataFlashROM		26	26	
RAM		13875 (9455)	13625 (8697)	Note
	Data	8755	8505	
	Stack	5120 (700)	5120 (192)	Note

Note: RAM usage shown in "()" is calculated from stack usage.

11.1.3 Number of Execution Cycles

The number of execution cycles and processing load during measurement for each block in "Figure 8-1 Weight Measurement Process Flow" is shown in Table 11-3.

Table 11-3 Number of Execution Cycles, Execution Time, and Processing Load

ICLK=32MHz
Measurement rate: 10SPS

Item	QF for AFE version		Modbus version		Condition
	Number of cycles (Execution time)	Processing load [%]	Number of cycles (Execution time)	Processing load [%]	
A/D value acquisition	32cycle (1.00μs)	0.0010	32cycle (1.00μs)	0.0010	
Weight conversion	186cycle (5.81μs)	0.0058	186cycle (5.81μs)	0.0058	Includes Zero reset processing
Communication control	328cycle (10.97μs)	0.0103	410cycle (12.81μs)	0.0128	QE: On sending weights data Modbus: On weight request processing
Others	80cycle (2.50μs)	0.0025	22cycle (0.69μs)	0.0007	
Total	626cycle (19.56μs)	0.0335	650cycle (20.31μs)	0.0203	

Note: The processing load is calculated based on the execution time in the measurement rate.

11.2 Weight Measurement Accuracy Evaluation

Results of weight measurement by load cell BCL-300GM-C3 shown in Table 6-1 are described in this section by using the RX23E-B-QFN40-WT board and the sample program.

11.2.1 Weight Measurement Accuracy Evaluation Conditions

System configuration of weight measurement is shown in Figure 11-1. Equipment used in the measurement is shown in Table 11-4. Also, combination of counterweights for measured weight and weight tolerance is shown in Table 11-5 and Table 11-6. Calibration was carried out for two kinds of weight, which are 0g (no load) and 250g, following "6.3 Calibration".

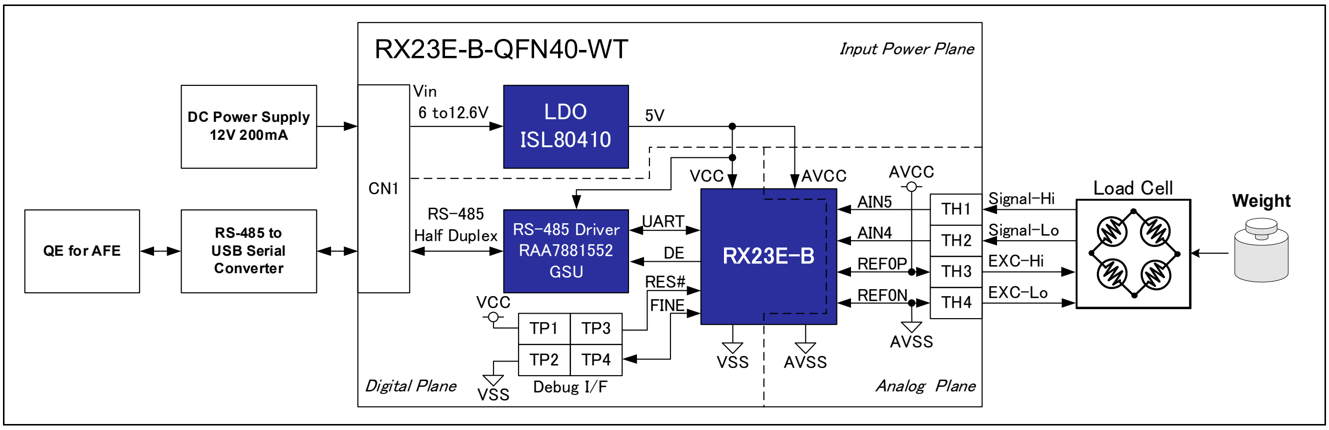


Figure 11-1 Configuration of Weight Measurement Accuracy Evaluation

Table 11-4 Equipment Used in Weight Measurement Accuracy Evaluation

Item	Model	Manufacturer
Load Cell	BCL-300GM-C3	Minebea Mitsumi Inc.
DC Power Supply	PCR1000MS	KIKUSUI ELECTRONICS CORPORATION
Counterweight	738-65-53-04	Tokyo Garasu Kikai Co., Ltd.

Table 11-5 Combination of Counterweights for Weight Settings

Weight setting	Combination of counterweights			
10g	10g x1			
20g	20g x1			
30g	10g x1	20g x1		
40g	10g x2	20g x1		
50g	50g x1			
60g	10g x1	50g x1		
70g	20g x1	50g x1		
80g	10g x1	20g x1	50g x1	
90g	10g x2	20g x1	50g x1	
100g	100g x1			
110g	10g x1	100g x1		
120g	20g x1	100g x1		
130g	10g x1	20g x1	100g x1	
140g	10g x2	20g x1	100g x1	
150g	50g x1	100g x1		
160g	10g x1	50g x1	100g x1	
170g	20g x1	50g x1	100g x1	
180g	10g x1	20g x1	50g x1	100g x1
190g	10g x2	20g x1	50g x1	100g x1
200g	200g x1			
210g	10g x1	200g x1		
220g	20g x1	200g x1		
230g	10g x1	20g x1	200g x1	
240g	10g x2	20g x1	200g x1	
250g	50g x1	200g x1		

Table 11-6 Counterweight Tolerance

Weight of counterweight	Counterweight tolerance
10g	±20mg
20g	±20mg
50g	±30mg
100g	±30mg
200g	±50mg

11.2.2 Weight Measurement Accuracy Evaluation Results

The weight measurement error is calculated from the weight measurement results. The result calculated by dividing the error by full by the load cell's rated capacity of 300g is shown in Figure 11-2. Since the error fits within the counterweight tolerance, which shows RX23E-B is sufficient measurement accuracy.

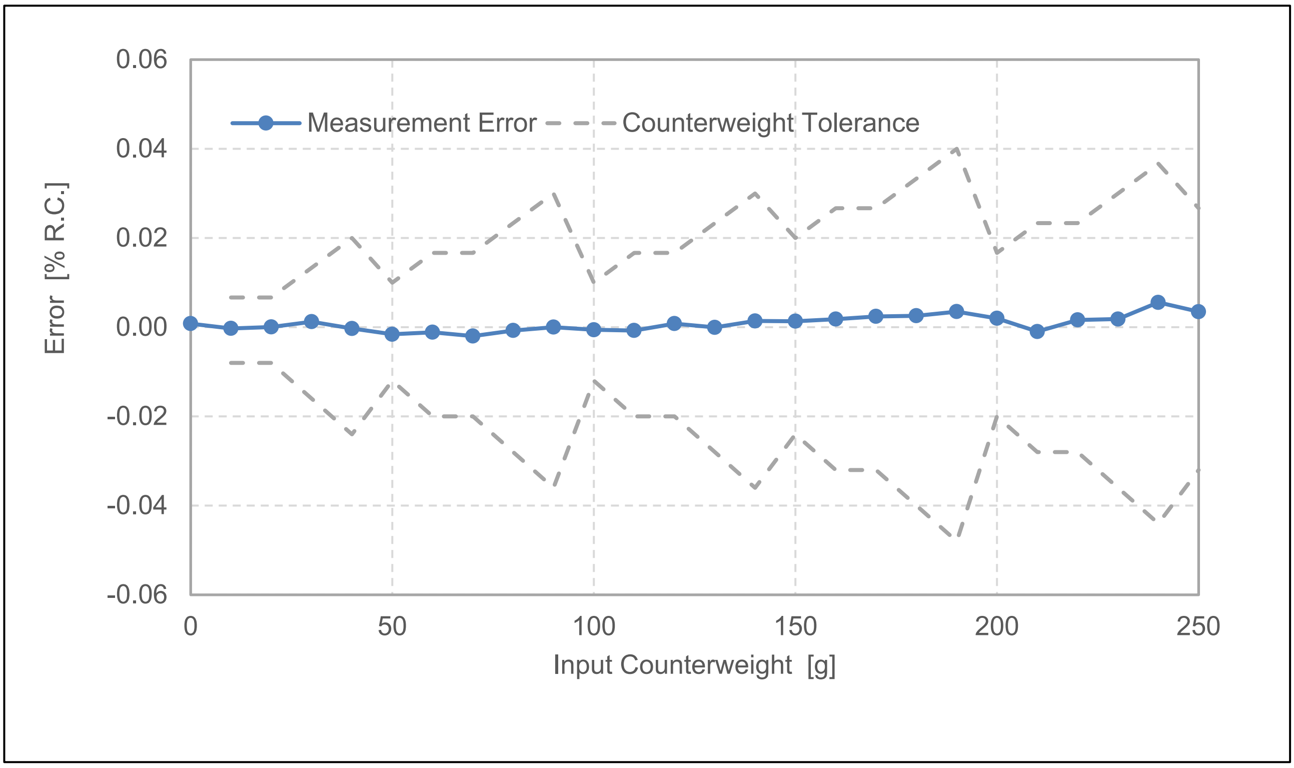


Figure 11-2 Weight Measurement Error (Ambient Temperature: 25°C)

11.3 Resolution Evaluation

Results of resolution evaluation by the precision calibrator PSC-350 are described in this section by using the RX23E-B-QFN40-WT board and the sample program.

11.3.1 Resolution Evaluation Conditions

The system configuration for resolution evaluation is shown in Figure 11-3 and the equipment used for the measurements is shown in Table 11-7. Calibration was performed at two voltages, 0mV and 10mV, according to "6.3 Calibration".

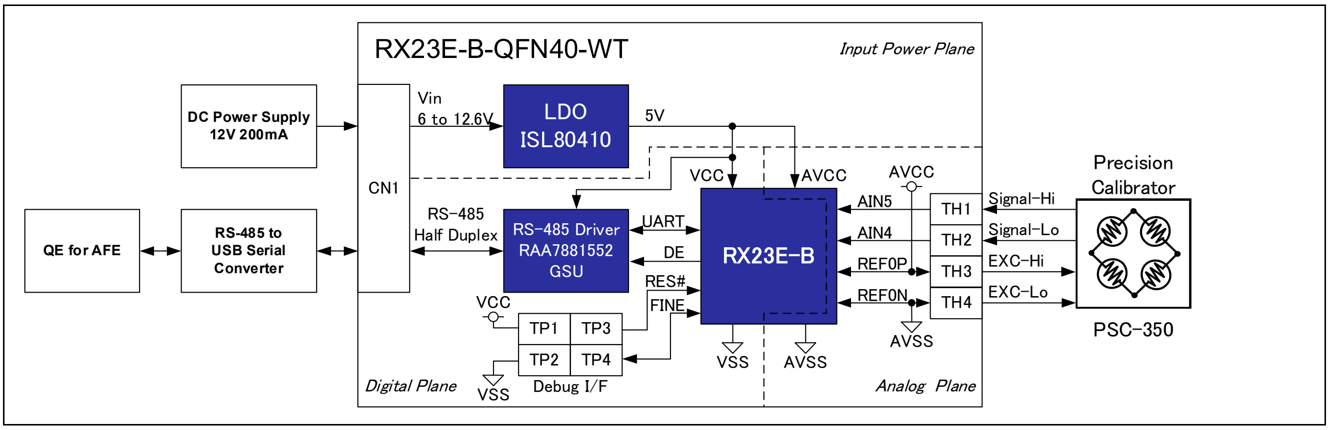


Figure 11-3 Configuration of Resolution Evaluation

Table 11-7 Equipment Used in Resolution Evaluation

Item	Model	Manufacturer
Precision Calibrator	PSC-350	Minebea Mitsumi Inc.
DC Power Supply	PCR1000MS	KIKUSUI ELECTRONICS CORPORATION

11.3.2 Resolution Evaluation Results

The output voltage of the precision calibrator was 10mV, RX23E-B-QFN40-WT was 10SPS, PGA gain was x128, and the voltage measurement of 1000 samples was performed under the condition of 0 times moving average. The histogram of the average of 1000 samples and the deviation from each data is shown in Figure 11-4. The measurement results are shown in Table 11-8. The input converted noise voltage was 23.8nVrms and the peak-to-peak noise voltage was 140nV.

Based on the noise measurement results, assuming a case of weight measurement using the load cell BCL-300GM-C3 instead of the precision calibrator, the noise weight caused by the measurement system is obtained by removing noise factors caused by the sensor itself and vibration. If the rated capacity of the load cell is 300g, the rated output is 0.9 mV/V, and the excitation voltage is 5V, the weight voltage sensitivity is 15uV/g, so the input equivalent noise weight is 1.58mgrms and the peak-to-peak noise weight is 9.3mg. 10SPS and PGA gain of RX23E-B Although the noise is slightly higher than the input equivalent noise of 18nVrms at x128, it shows that the RX23E-B can perform weight measurement with high accuracy.

Table 11-8 Resolution Evaluation Result

Equivalent input noise:	23.8nVrms (21.6 Bits): equivalent to1.58mgrms
Peak-to-peak noise:	140nV (19.1 Bits): equivalent to 9.3mg

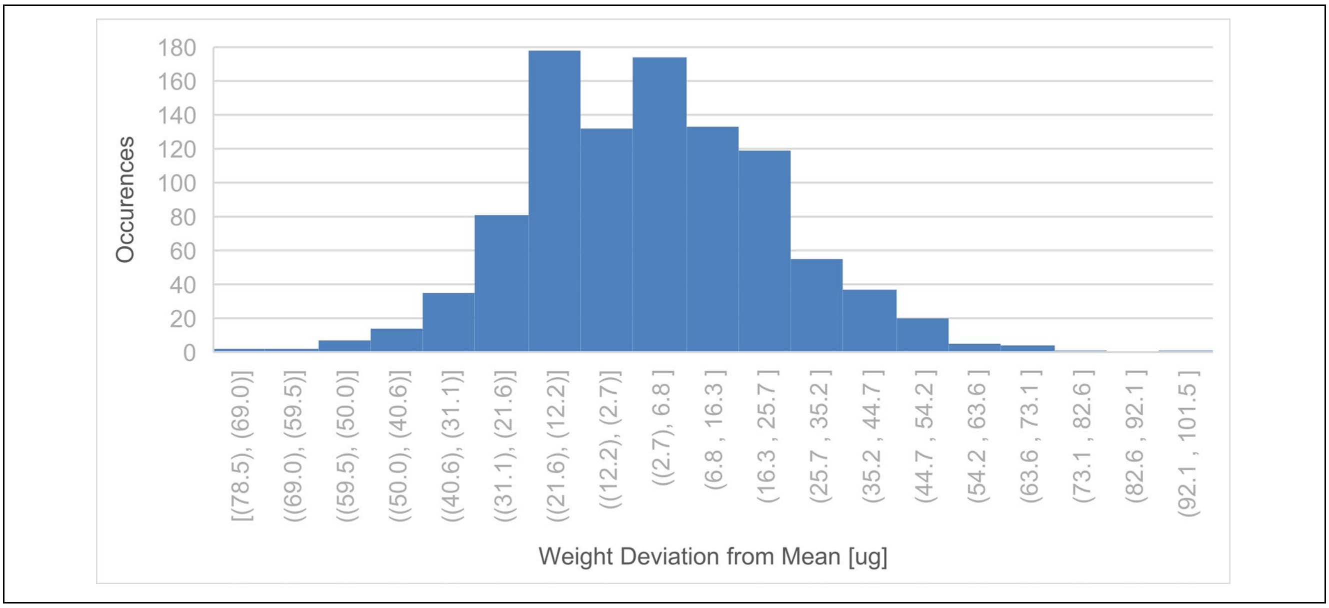


Figure 11-4 Histogram of Errors from the Average of 1000 Sample Measurements at 10mV Input

Revision History

Rev.	Date	Description	
		Page	Summary
Rev.1.00	Oct.31.23	-	First issue

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.

Corporate Headquarters

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

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/.