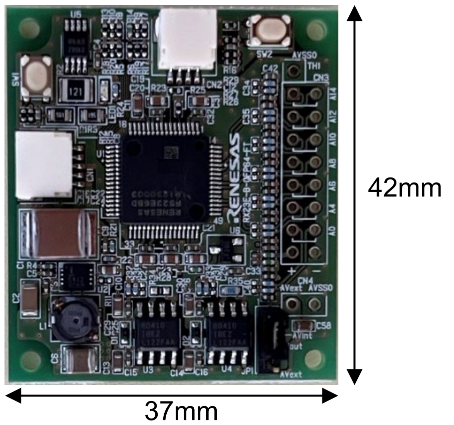
RX23E-B Group

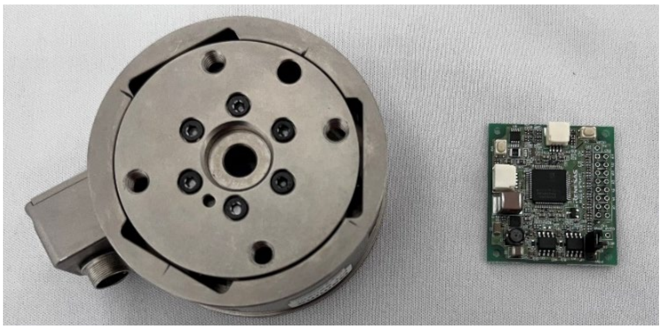
Design and measurement of small board for 6-axis force sensor

Introduction

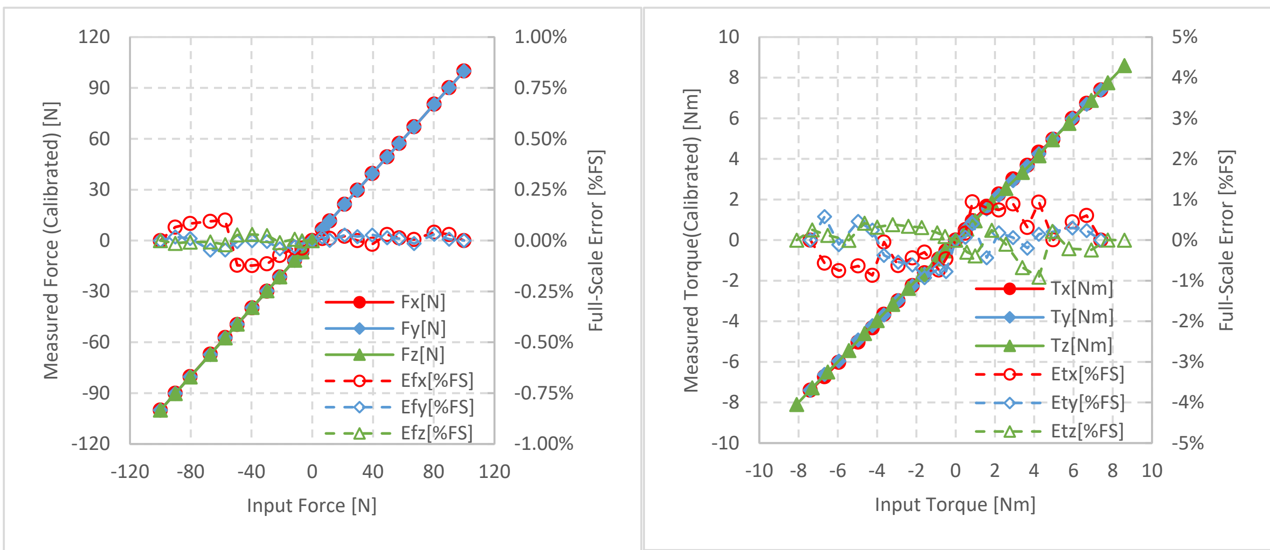
This document describes the RX23E-B-QFP64-FT, which is a board for 6-axis force sensor using Renesas MCU, RX23E-B, and the example of the program which obtains three-dimensional force and torque by the strain gauge type 6-axis force sensor.

RX23E-B-QFP64-FT is miniaturized to be incorporated into a 6-axis force sensor, using RX23E-B, MCU with AFE in 64 pin LFQFP package, a DC/DC converter ISL85412 and LDO ISL80410 as power supply, and RAA7881582GSU as RS-485 driver.





This sample program uses the DSAD built in the RX23E-B to obtain output from six channels of the force sensor by scanning them. We have measured the 6-axis force sensor with RX23E-B-QFP64-FT and this program. The evaluation results are shown below. The force measurement error is within ±0.25% FS, and the torque measurement error is within ±1% FS, indicating that these errors are within the measurement uncertainty (max ±1.25%[FS]) of the force sensor used in this measurement.



Result of Force Measurement (Left) and Torque Measurement (Right)

Target Device

RX23E-B (R5F523E6BDFM)

Contents

1. Overview

This document describes the example of measuring three-dimensional force and torque by using the small board RX23E-B-QFP64-FT containing RX23E-B and the 6-axis force sensor. The sample program conducts measurement with a force sensor, communicates with QE for AFE or Modbus host via an RS-485 half-duplex communication channel, and transmits measurement results.

Figure 1-1 shows the system of this example.

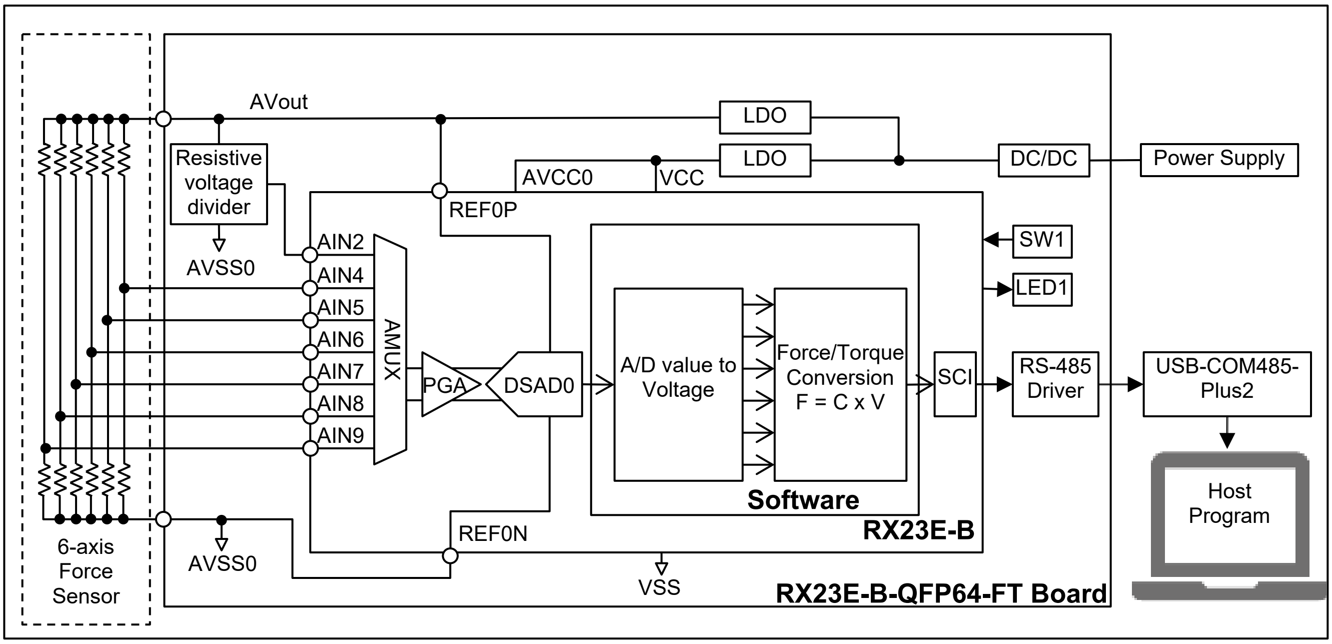


Figure 1-1 Example of Force Sensor Measurement System

In this example, the QE for AFE version sample program uses the application tab of QE for AFE to make various settings, conduct measurements, and display measurement results. The operable items are shown in Figure 1-2 and Table 1-1.

The Modbus version sample program operates similarly by making settings in Modbus Coil or Holding register listed in Table 7-6.

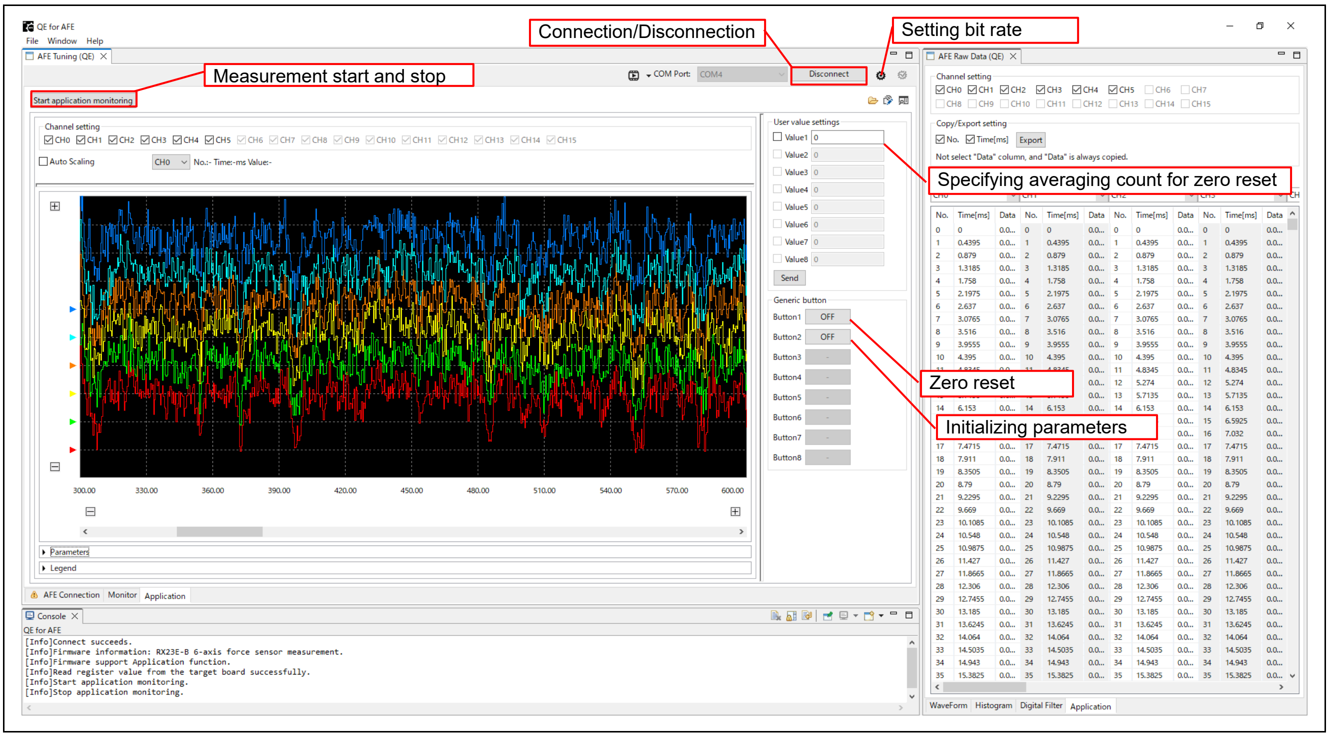


Figure 1-2 QE for AFE Application Tab Screen

Table 1-1 Operable Items

Item	Operation	Remarks
Connection	QE for AFE: Connect/Disconnect button	
Start/stop of measurement	QE for AFE: Start/Stop button	LED1 OFF during measurement
Zero reset	QE for AFE: Button1 Board: SW1	Enabled only during standby (LED1 ON)
Specifying the averaging count for zero reset	QE for AFE: Value1 64 to 512, default: 256	
Parameter initialization	QE for AFE: Button2	
	Board: SW1	Press SW1 and SW2 at the same time (reset), release SW2, and keep pressing SW1 until LED1 turns on.

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 the full-duplex communication, transmission and reception may conflict and stop on instruction of measurement stop. If it stops, restart the MCU.

Also, changes to the parameters listed in Table 1-2 are retained in E2 data flash. For details, refer to structure st_prm_t in Table 8-27.

Table 1-2 Retained Parameters

Item	**Number of data stored**
Voltage-load conversion matrix	1 set
DSAD0 offset correction value	1 set
Averaging count for zero reset	1

2. Package Contents

Table 2-1 Package Contents

File/folder name	Description
r01an6513jj0100-rx23e-b.pdf	This document (Japanese)
r01an6513ej0100-rx23e-b.pdf	This document (English)
BoardData	Board data of RX23E-B-QFP64-FT
rx23eb_force_qe	QE for AFE version sample project set
rx23eb_force_modbus	Modbus version sample project set
readme_j.txt	Package explanation (Japanese)
readme_e.txt	Package explanation (English)

3. Environment for Operation Confirmation

Table 3-1 Environment for Operation Confirmation

<table>
<tr><th colspan="2">Item</th><th colspan="2">Description</th></tr>
<tr><td colspan="2">Board</td><td colspan="2">RX23E-B-QFP64-FT</td></tr>
<tr><td colspan="2">MCU</td><td colspan="2">RX23E-B (R5F523E6BDFM)
Power voltage (VCC, AVCC0): 5V
Operating frequency (ICLK): 32MHz
Peripheral operating frequency (PCLKB, PCLKC): 32MHz
DSAD0 operating frequency (f_{OP}): 16MHz
DSAD0 modulator clock frequency (f_{MOD}): 4MHz</td></tr>
<tr><td colspan="2" rowspan="4">Force sensor</td><td>Manufacturer</td><td>ATI Industrial Automation</td></tr>
<tr><td>Model</td><td>9105-TWE-Gamma</td></tr>
<tr><td>Calibration</td><td>SI-130-10</td></tr>
<tr><td>Measurement uncertainty</td><td>Fx: 1.00%, Fy: 1.25%, Fz: 0.75%
Tx: 1.00%, Ty: 1.00%, Tz: 1.50%</td></tr>
<tr><td colspan="2">RS-485/USB conversion BOX</td><td colspan="2">Renesas RS-485-USB-POWER-BOX</td></tr>
<tr><td rowspan="2">Host</td><td>QE for AFE version</td><td colspan="2">Renesas QE for AFE V2.1.1</td></tr>
<tr><td>Modbus version</td><td colspan="2">QModMaster 0.5.3-beta</td></tr>
<tr><td colspan="2">IDE</td><td colspan="2">Renesas e2 Studio Version 2023-04
Renesas RX Smart Configurator V23.4.0</td></tr>
<tr><td colspan="2">Tool Chain</td><td colspan="2">Renesas CC-RX V3.05.00</td></tr>
<tr><td colspan="2">Emulator</td><td colspan="2">Renesas E2 Emulator Lite</td></tr>
</table>

4. Related Documents

- R01UH0972 RX23E-B Group User’s Manual: Hardware
- R01AN4359 RX family RX DSP Library Version 5.0
- R01AN6364 RX23E-B Group RSSKRX23E-B Board Control Program

5. RX23E-B-QFP64-FT

5.1 Board Specification

Table 5-1 RX23E-B-QFP64-FT Specifications

Item		Specification
External dimensions		37mm×42mm
Layer structure		4 layers, Laminating order: Signal - GND - Power supply - Signal
Board model name		RX23E-B-QFP64-FT
Operating voltage		Recommended operating voltage: 12 to 24V Maximum operating voltage: 26.4V
Current consumption		29.4mA typ. (Board alone)
Analog input specification		Up to 7 differential inputs, 14 single ended inputs
Communication I/F		RS-485, Half-duplex communication Maximum communication speed: 1Mbps Terminating resistor: 120Ω
Compatible emulator		Renesas E2 Emulator, E2 Emulator Lite
User I/F	LED	LED1: Green
	Switch	SW1: For Zero-Reset SW2: For MCU reset

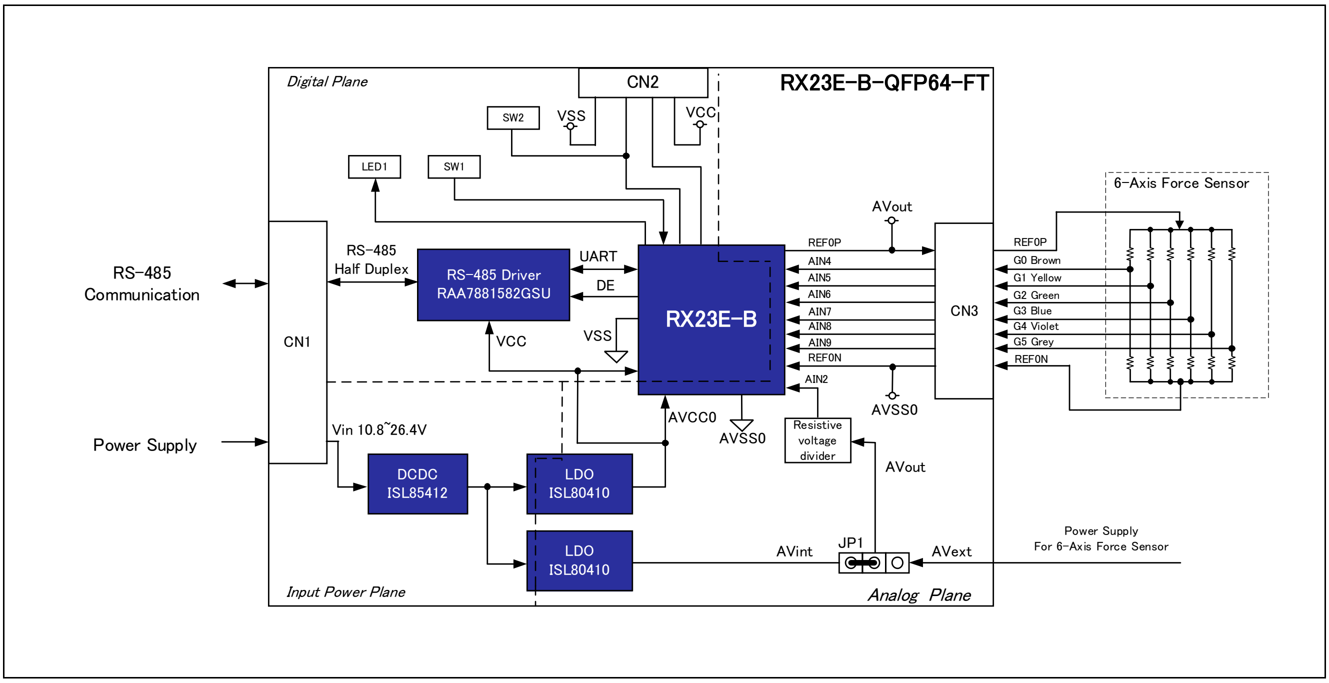


Figure 5-1 System Configuration

5.2 Circuit Diagram

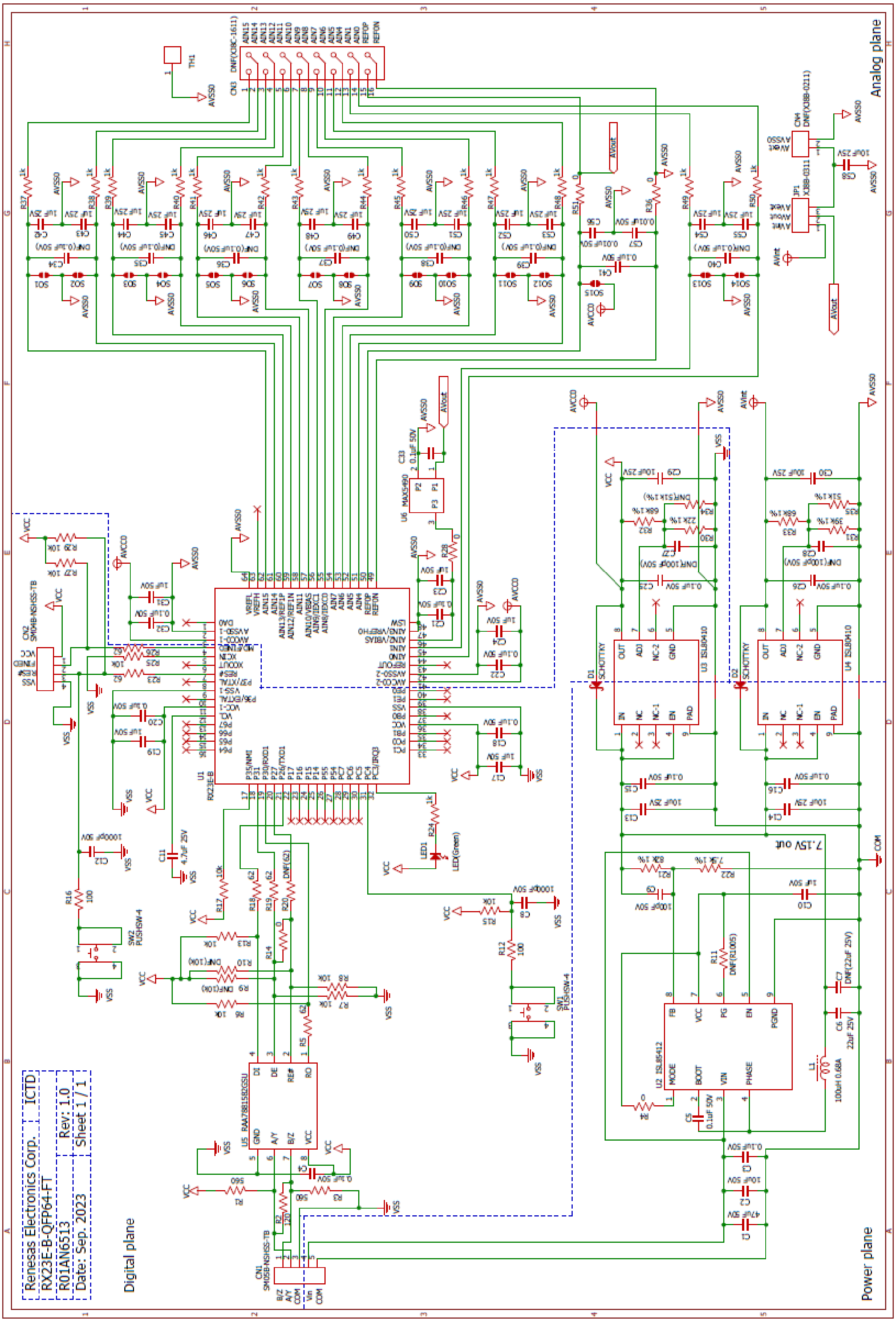


Figure 5-2 RX23E-B-QFP64-FT Circuit Diagram

5.3 Bill of Materials

Table 5-2 RX23E-B-QFP64-FT Bill of Materials (1/2)

No.	Q'ty	Reference Designator	Description	Part Name	Manufacturer Part Name	Maker Name
1	1	U1	RX23E-B	IC	R5F523E6BDFM	Renesas
2	1	U2	DCDC	IC	ISL85412FRTZ	Renesas
3	2	U3,U4	LDO	IC	ISL80410IBEZ	Renesas
4	1	U5	RS-485 Driver	IC	RAA7881582GSU	Renesas
5	1	U6	Resistive voltage divider	IC	MAX5490GA01000+T	Maxim
6	1	CN1	5pin	Connector	SM05B-NSHSS-TB	JST
7	1	CN2	4pin	Connector	SM04B-NSHSS-TB	JST
8	1	JP1	3pin	Pin header	XJ8B-0311	Omron
9	1	C9	100pF 50V	Ceramic Capacitor	GCM1552C1H101JA01	Murata
10	2	C8,C12	1000pF 50V	Ceramic Capacitor	GRM1552C1H102JA01	Murata
11	2	C56,C57	0.01uF 50V	Ceramic Capacitor	GCM155R71H103KA55	Murata
12	14	C3,C4,C5,C15, C16,C18,C20, C21,C22,C25, C26,C32,C33, C41	0.1uF 50V	Ceramic Capacitor	GRM155R71H104KE14	Murata
13	14	C42,C43,C44, C45,C46,C47, C48,C49,C50, C51,C52,C53, C54,C55	1uF 25V	Ceramic Capacitor	C1005X5R1E105K050BC	TDK
14	6	C10,C17,C19, C23,C24,C31	1uF 50V	Ceramic Capacitor	C1608X5R1H105K080AB	TDK
15	1	C11	4.7uF 25V	Ceramic Capacitor	GRM188C81E475KE11	Murata
16	5	C13,C14,C29, C30,C58	10uF 25V	Ceramic Capacitor	GRM188R61E106KA73	Murata
17	1	C2	10uF 50V	Ceramic Capacitor	GCM31CD71H106KE35	Murata
18	1	C6	22uF 25V	Ceramic Capacitor	GCM32EC71E226KE36	Murata
19	1	C1	47uF 50V	Ceramic Capacitor	CKG57NX7R1H476M500JJ	TDK
20	2	D1,D2	SCHOTTKY, VR=40V	Diode	RB521SM-40	Rohm
21	1	L1	100uH 0.68A	Coil	CLF5030NIT-101M-D	TDK
22	1	LED1	Green	LED	SML-E12P8W	ROHM
23	5	R4,R14,R28, R36,R51	0	Resistor	RK73Z1ETTP	KOA
24	4	R5,R18,R19, R26	62	Resistor	RK73B1ETTP620J	KOA
25	2	R12,R16	100	Resistor	RK73B1ETTP101J	KOA
26	1	R2	120	Resistor	RK73B2ETTD121J	KOA
27	2	R1,R3	560	Resistor	RK73B2ATTD561J	KOA

Table 5-3 RX23E-B-QFP64-FT Bill of Materials (2/2)

No.	Q'ty	Reference Designator	Description	Part Name	Manufacturer Part Name	Maker Name
28	15	R24,R37,R38,R39, R40,R41,R42,R43. R44,R45,R46,R47, R48,R49,R50	1k	Resistor	RK73H1ETTP1001F	KOA
29	1	R22	7.5k 1%	Resistor	RK73H1ETTP7501F	KOA
30	10	R6,R7,R8,R13,R15, R17,R23,R25,R27, R29	10k	Resistor	RK73B1ETTP103J	KOA
31	1	R30	22k 1%	Resistor	RK73H1ETTP2202F	KOA
32	1	R31	39k 1%	Resistor	RK73H1ETTP3902F	KOA
33	1	R35	51k 1%	Resistor	RK73H1JTTD5102F	KOA
34	2	R32,R33	68k 1%	Resistor	RK73H1ETTP6802F	KOA
35	1	R21	82k 1%	Resistor	RK73H1ETTP8202F	KOA
36	2	SW1,SW2	SW	switch	SKRPABE010	ALPS ALPINE CO., LTD.
M1	4	-		Screw	B-0206-S1	Hirosugi
M2	4	-		Spacer	ASB-2010E	Hirosugi
M3	1	JP1		Jumper socket	XJ8A-0241	Omron

Note: This list may be changed without notice.

5.4 Pattern Diagram

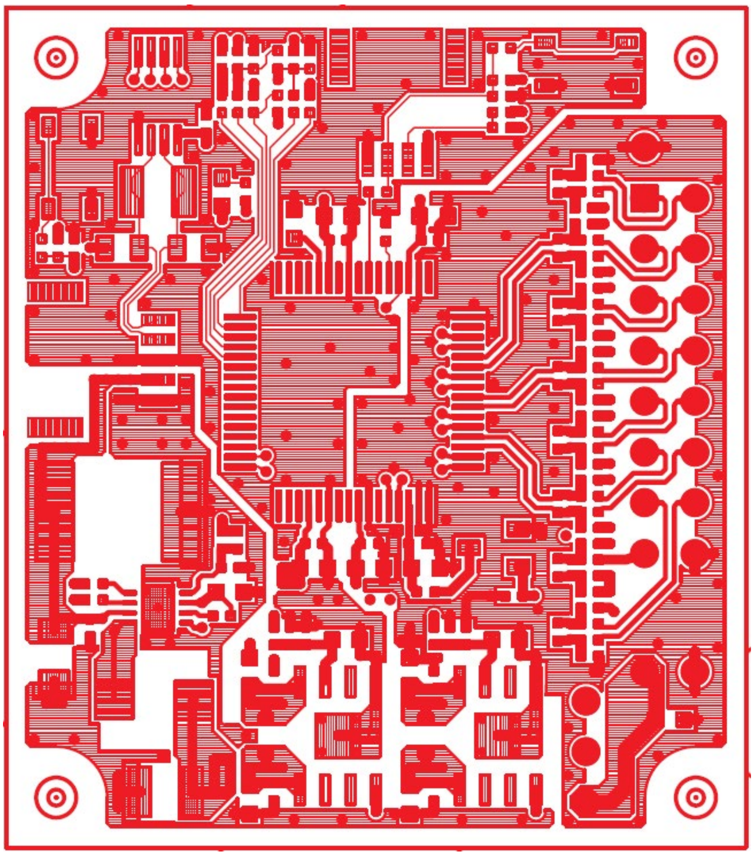


Figure 5-3 Layer 1

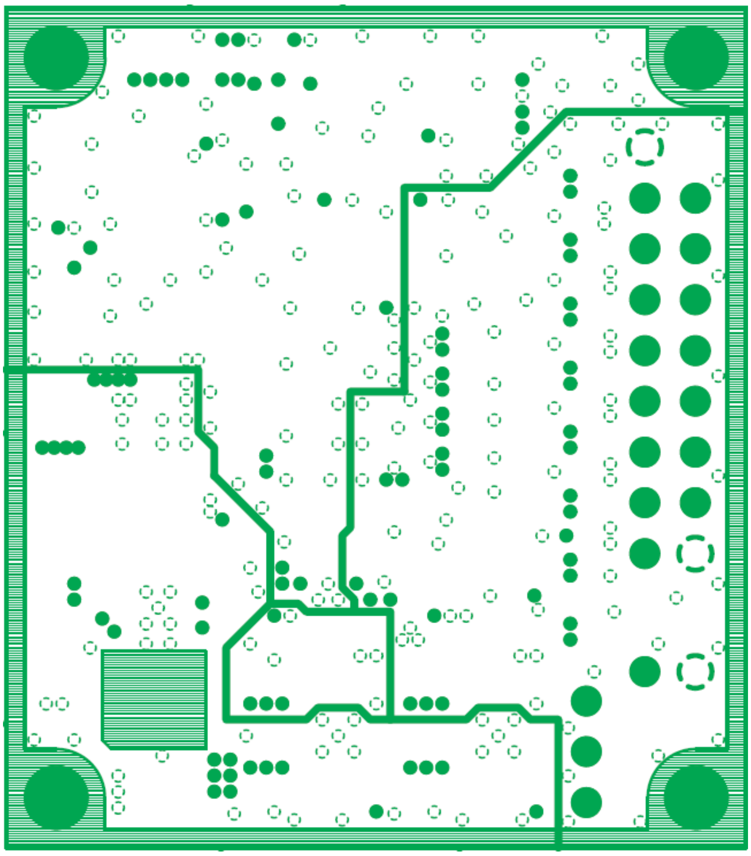


Figure 5-4 Layer 2

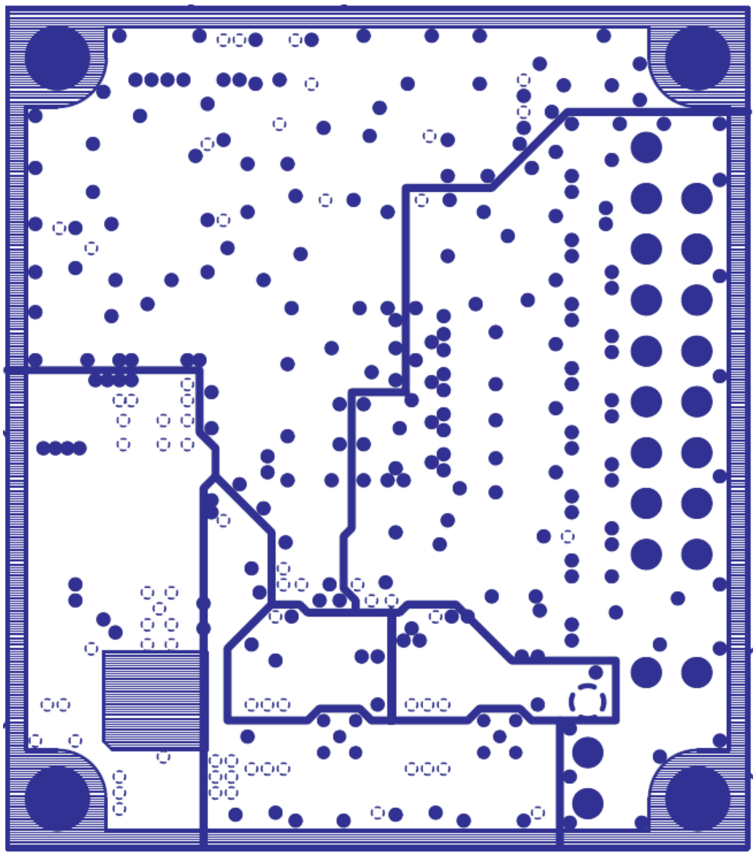


Figure 5-5 Layer 3

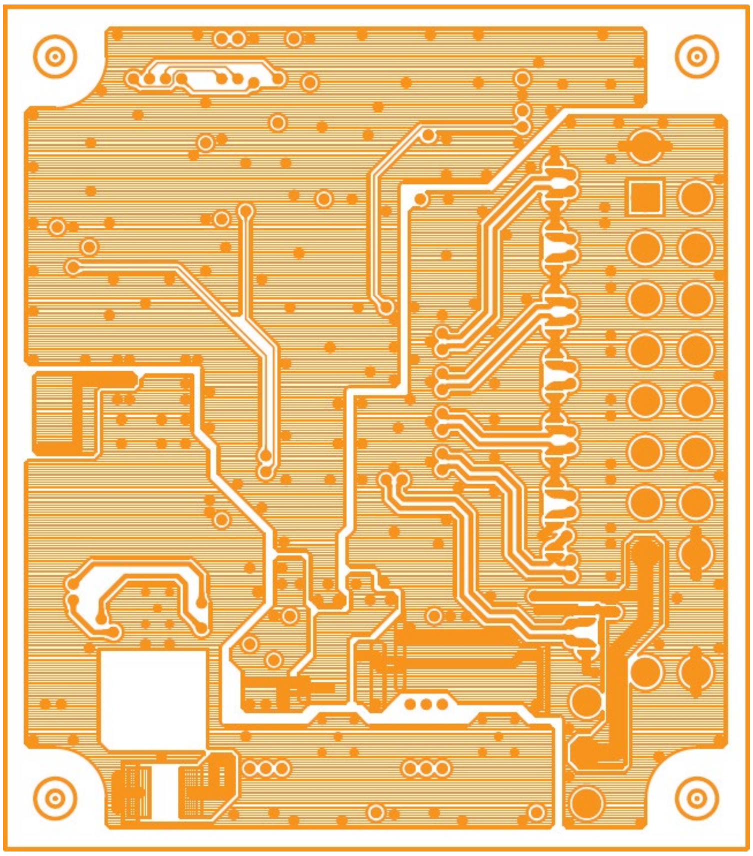


Figure 5-6 Layer 4

6. Force Sensor Measurement

Figure 6-1 shows the connection between the force sensor and the RX23E-B-QFP64-FT.

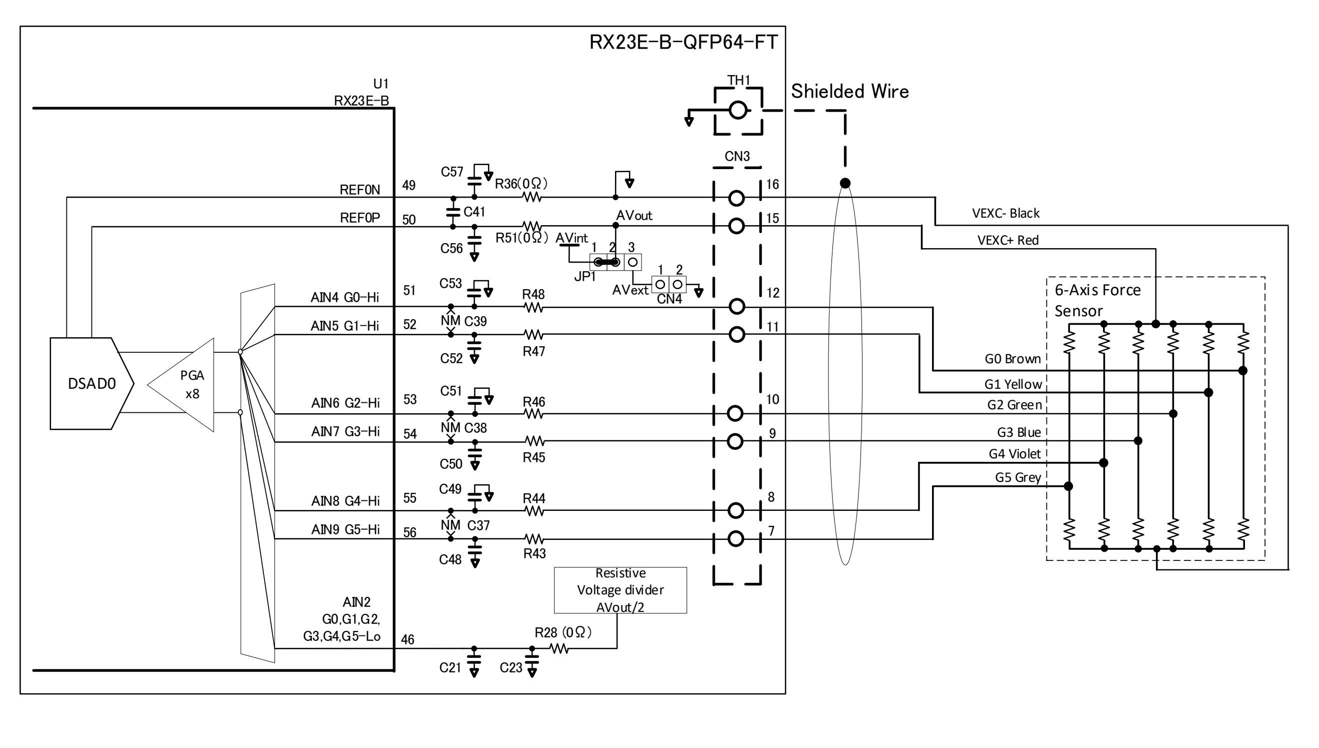


Figure 6-1 Connection of RX23E-B-QFP64-FT and Force Sensor

When a voltage is applied to the excitation voltage terminal of a force sensor, the force sensor outputs the voltage divided by the half-bridge resistors connecting the strain gauges in series in the force sensor. For the voltage AVout applied to the excitation terminal of the force sensor, you can select the power voltage AVint which is generated within the board when connecting the JP1_1-2 pins. When connecting the JP1_2-3 pins, you can select the external power supply voltage AVext which is input from CN4.

The output of the force sensor is connected to AIN4, AIN5, AIN6, AIN7, AIN8, and AIN9 as positive input for DSAD0. The half voltage of AVout is connected to AIN2 as negative input for DSAD0. In measurement, the voltages of AIN4, AIN5, AIN6, AIN7, AIN8, and AIN9 for AIN2 are sequentially A/D converted by using the channel function of the DSAD0, and the force and torque is converted from the A/D conversion results of 6 channels.

6.1 Force Sensor

The strain gauge type 6-axis force sensor is a sensor that utilizes the fact that the resistance value of each strain gauge mounted on the strain body changes due to stress. By applying a voltage to the 6-axis force sensor, the change in resistance value due to stress is measured as a voltage.

If the output voltage of the strain gauge is non-linear in relation to the stress, the characteristic curve is divided into multiple regions, and linear approximation, for example, is performed in each of the regions to increase the measurement precision, thereby matching the characteristic curve. In this example, the region is regarded as a single linear characteristic without being divided.

Supposing that the voltage applied to the strain gauge is V_{CC}, the rated output is RO, and the load rating is S_{max}, the output voltage V for the applied strain S is calculated as below.

$$V = \mathrm{RO} \cdot V_{cc} \cdot \frac{S}{S_{max}}$$

Multiply the acquired 6-axis voltage by the force sensor-specified voltage-load conversion matrix C to calculate the force and torque on x, y, and z axis.

$$F = C \times V$$

$$\begin{pmatrix} F_x \\ F_y \\ F_z \\ T_x \\ T_y \\ T_z \end{pmatrix} = \begin{pmatrix} C_{11} & \cdots & C_{16} \\ \vdots & \ddots & \vdots \\ C_{61} & \cdots & C_{66} \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{pmatrix}$$

In this example, ATI Industrial Automation 9105-TWE-Gamma is used as a force sensor for measurement. The appearance of the force sensor is shown in Figure 6-2.

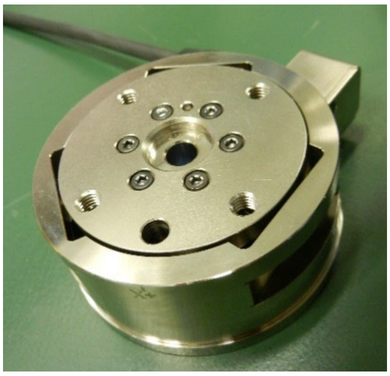


Figure 6-2 Appearance of ATI Industrial Automation 9105-TWE-Gamma

6.2 A/D Conversion of Strain Gauge Output

In this example, the output voltage of each strain gauge is A/D converted with the voltage supplied to each strain gauge as the reference voltage, as shown in Figure 6-1.

Table 6-1 shows the measurement conditions of the force sensor. The digital filter gain is corrected to be 1 by Sinc filter gain correction.

Table 6-1 Force Sensor Measurement Conditions

Item	Condition	Remarks
PGA gain G_{PGA}	x8	
DSAD0 reference voltage V_{REF}	5V	Voltage applied to the strain gauge (REF0P=AVOUT, REF0N=AVSS0)
Oversampling ratio OSR	32	
Digital filter gain correction value	1.0	$1/G_{DF}$
DSAD0 output format	2's Complement	

This example uses the DSAD0 on RX23E-B to scan the output from the 6-axis force sensor. Figure 6-3 shows the conversion sequence, and Table 6-2 shows the A/D conversion time.

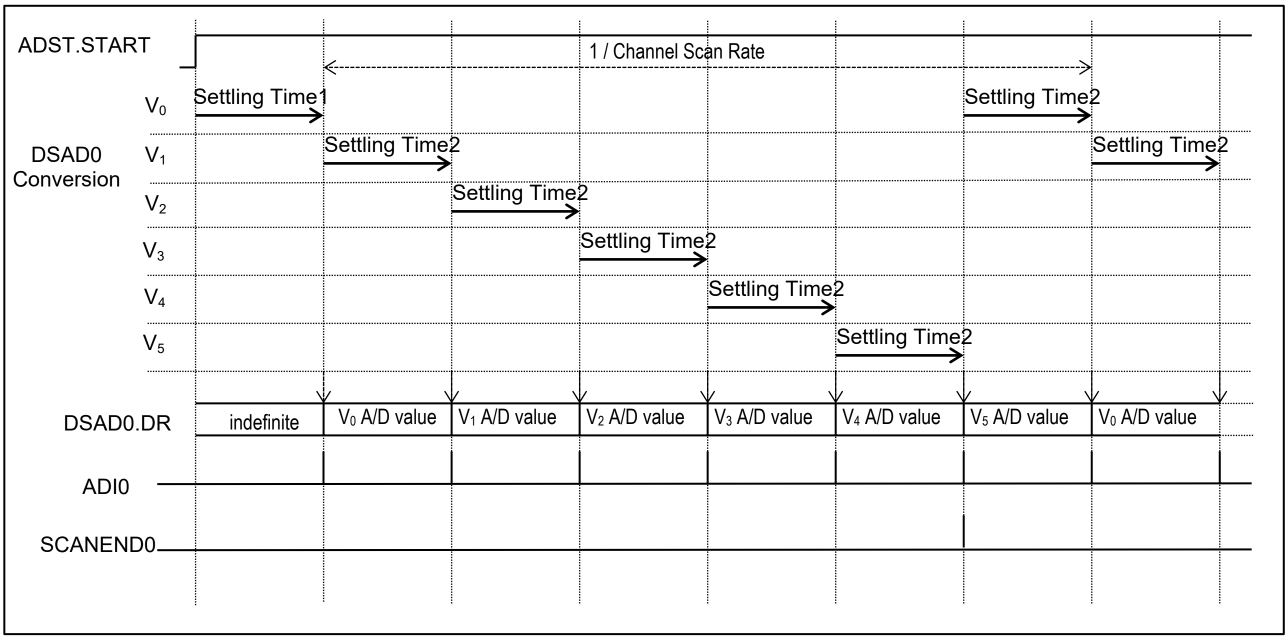


Figure 6-3 A/D Conversion Sequence and A/D Conversion Time

Table 6-2 A/D Conversion Time

f_{MOD} = 4MHz
PCLKB=32MHz
Over Sampling Ratio (OSR) = 32
Sinc Filter (FSEL): Sinc4 + Sinc4

Item	Value	Remarks
Settling Time1	73.65625μs	
Settling Time2	73.25μs	
Channel Scan Rate	2275.312856 scan/s	= 1 / (Settling Time2 x 6ch)

6.3 Calculation Procedure

Conversion from the A/D conversion value into the force and torque is performed with the procedure below.

(1) Voltage Calculation

Convert the A/D conversion values outputted from each strain gauge in the force sensor into voltages. Supposing that the PGA gain is G_{PGA}, the reference voltage of the DSAD0 is V_{REF}, and the A/D conversion value is $DATA_n$, output voltage V_n from each strain gauge is calculated from the DSAD0 resolution of 24bit by the equation below.

$$\begin{aligned} V_n &= \frac{2V_{REF}}{2^{24} \cdot G_{PGA}} \cdot \mathrm{DATA}_n \\ &= \frac{V_{REF}}{2^{23} \cdot G_{PGA}} \cdot \mathrm{DATA}_n, \quad n = 0 \sim 5 \end{aligned}$$

(2) Force and Torque Conversion

Multiply the acquired 6-axis voltage by the force sensor-specified voltage-load conversion matrix C to calculate the force and torque on x, y, and z axis.

$$F = C \times V$$

$$\begin{pmatrix} F_x \\ F_y \\ F_z \\ T_x \\ T_y \\ T_z \end{pmatrix} = \begin{pmatrix} C_{11} & \cdots & C_{16} \\ \vdots & \ddots & \vdots \\ C_{61} & \cdots & C_{66} \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{pmatrix}$$

6.4 Zero-Reset

To correct mechanical offset etc., the A/D conversion value at no load is adjusted to be zero.

In this example, supposing that the offset value is the average of A/D conversion values of individual strain gauge at no load, set the offset value in DSAD0 offset correction register OFCRm so that the offset is canceled.

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	4,000,000 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 module, 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 an instruction of measurement stop.

7.2 Modbus RTU

Operation setting and measurement result acquisition are performed with Modbus RTU communication. For details about 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 and Table 7-7.

Table 7-6 Data List (1/2)

Function	Address	size	Format	Name	description
Coil	0	2byte	uint16	Measurement	Force sensor measurement H'0000: Off (default) H'0001: On
	1			Zero reset	Zero reset H'0000: Stop (default) H'0001: Start
Input Status	0	2byte	uint16	CH0 OVF	CH0 Error/Overflow flag at DSAD0 operation
	1			CH0 ERR	
	2			CH1 OVF	CH1 Error/Overflow flag at DSAD0 operation
	3			CH1 ERR	
	4			CH2 OVF	CH2 Error/Overflow flag at DSAD0 operation
	5			CH2 ERR	
	6			CH3 OVF	CH3 Error/Overflow flag at DSAD0 operation
	7			CH3 ERR	
	8			CH4 OVF	CH4 Error/Overflow flag at DSAD0 operation
	9			CH4 ERR	
	10			CH5 OVF	CH5 Error/Overflow flag at DSAD0 operation
	11			CH5 ERR	
Input Register	0	4byte	float	Fx	Force on x-axis
	2	4byte	float	Fy	Force on y-axis
	4	4byte	float	Fx	Force on z-axis
	6	4byte	float	Tx	Torque on x-axis
	8	4byte	float	Ty	Torque on y-axis
	10	4byte	float	Tz	Torque on z-axis
	12	4byte	int32	CH0 A/D Value	CH0 A/D value at DSAD0 operation
	14	4byte	int32	CH1 A/D Value	CH1 A/D value at DSAD0 operation
	16	4byte	int32	CH2 A/D Value	CH2 A/D value at DSAD0 operation
	18	4byte	int32	CH3 A/D Value	CH3 A/D value at DSAD0 operation
	20	4byte	int32	CH4 A/D Value	CH4 A/D value at DSAD0 operation
	22	4byte	int32	CH5 A/D Value	CH5 A/D value at DSAD0 operation

Table 7-7 Data List (2/2)

Function	Address	size	Format	Name	description
Holding Register	0	4byte	float	C11	Voltage-load conversion matrix
	2	4byte	float	C12	
	4	4byte	float	C13	
	6	4byte	float	C14	
	8	4byte	float	C15	
	10	4byte	float	C16	
	12	4byte	float	C21	
	14	4byte	float	C22	
	16	4byte	float	C23	
	18	4byte	float	C24	
	20	4byte	float	C25	
	22	4byte	float	C26	
	24	4byte	float	C31	
	26	4byte	float	C32	
	28	4byte	float	C33	
	30	4byte	float	C34	
	32	4byte	float	C35	
	34	4byte	float	C36	
	36	4byte	float	C41	
	38	4byte	float	C42	
	40	4byte	float	C43	
	42	4byte	float	C44	
	44	4byte	float	C45	
	46	4byte	float	C46	
	48	4byte	float	C51	
	50	4byte	float	C52	
	52	4byte	float	C53	
	54	4byte	float	C54	
	56	4byte	float	C55	
	58	4byte	float	C56	
	60	4byte	float	C61	
	62	4byte	float	C62	
	64	4byte	float	C63	
	66	4byte	float	C64	
	68	4byte	float	C65	
	70	4byte	float	C66	
	72	4byte	int32	OFCR0	Offset correction value for each CH of DSAD0
	74	4byte	int32	OFCR1	
	76	4byte	int32	OFCR2	
	78	4byte	int32	OFCR3	
	80	4byte	int32	OFCR4	
	82	4byte	int32	OFCR5	
	84	2byte	uint16	Num of Average	Averaging count for zero reset

7.2.3 Operation

Operation via Modbus equivalent to operation on QE for AFE in "Table 1-1 Operable Items" is shown in Table 7-8.

Table 7-8 Items Operable via Modbus

<table>
<tr><th>Item</th><th>Operation</th><th>Remarks</th></tr>
<tr><td>Measurement start and stop</td><td>Operate Coil:0</td><td>LED1 is OFF during measurement</td></tr>
<tr><td>Zero reset</td><td>Set Coil:1</td><td rowspan="3">Enabled only during standby (LED1 is ON)</td></tr>
<tr><td>Specifying the averaging count for zero reset</td><td>Set 64 to 512 in HoldingReg:11, default: 128</td></tr>
<tr><td>Parameter initialization</td><td>Set Coil:2</td></tr>
</table>

8. Sample Program

8.1 Overview of Operation

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

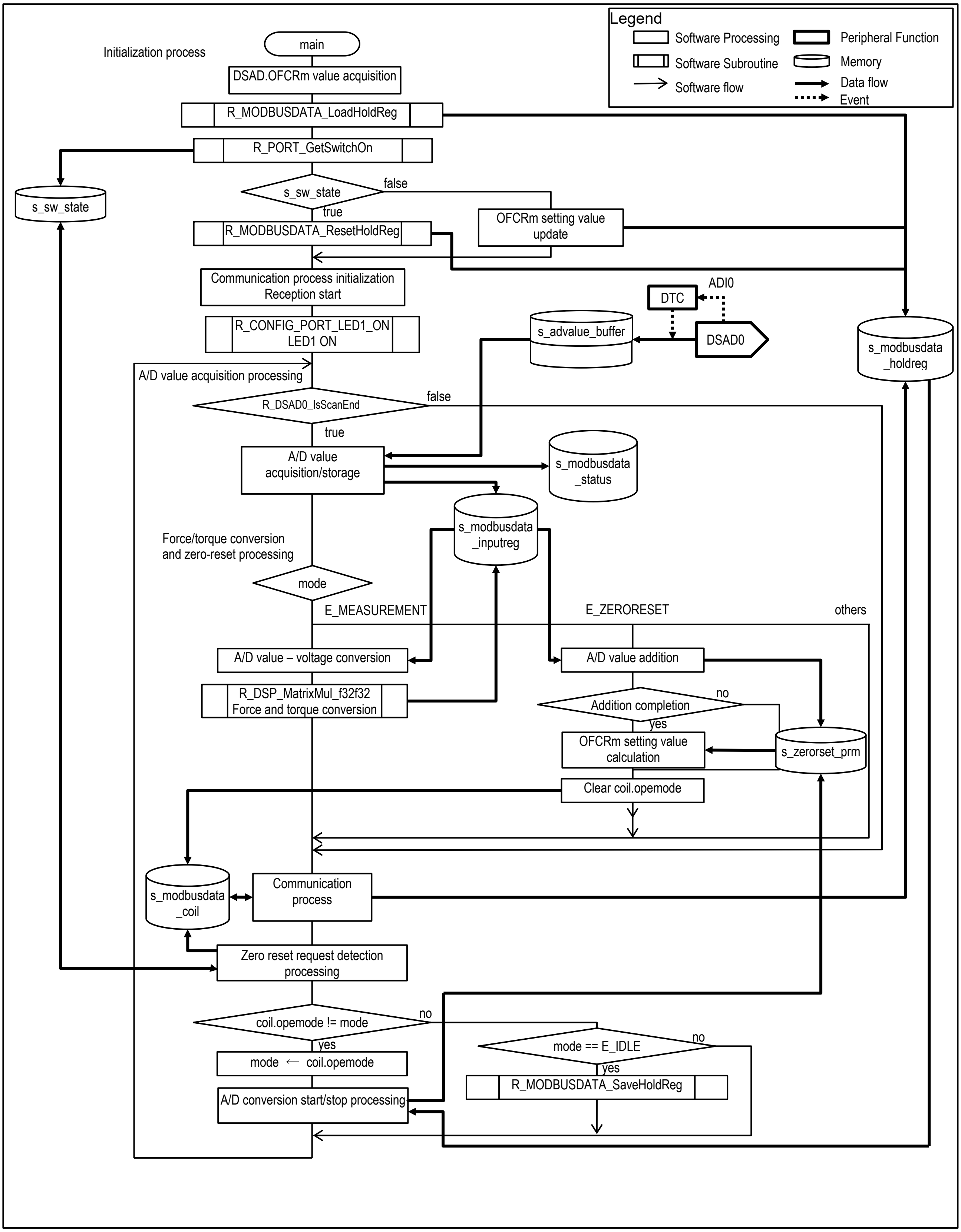


Figure 8-1 Force Sensor Measurement Process Flow

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_ZERORESET	2	Zero reset

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

- Initialization process
  - Loads the parameters stored in E2 data flash into the Holding register in ModbusData.
  - When SW1 is pressed, initializes the DSAD0 offset correction value and the averaging count for zero reset in the Holding register.
  - Initializes the communication process and starts reception.
  - Turns LED1 on.

- A/D value acquisition processing
  With the end of the A/D conversion (ADI0) as a trigger, transmits the A/D conversion result to the A/D conversion result storage array by DTC. The A/D conversion result storage array is 6 x 2, and the plane which is not the target of DTC transfer is to be processed.
  When the end of DSAD0 channel scan (SCANEND0) is detected, stores the A/D value in the Input register in ModbusData and the error information on A/D conversion in the Status in ModbusData from the acquired A/D conversion result of 6 channels.

- Force and torque conversion and zero-reset processing
  Processes the acquired A/D value according to the operating mode in A/D conversion start/stop processing.
  - opemode: E_MEASUREMENT
    According to “6.3 Calculation Procedure”, converts the A/D values into voltage values and multiplies them by voltage-load conversion matrix to calculate the force and torque. Stores the calculation results in the Input register in ModbusData.

  - opemode: E_ZERORESET
    According to “6.4 Zero Reset”, stores the average of the A/D values of each channel in the Holding register in ModbusData as offset correction values. The averaging count is based on the Holding register member average.

- Communication process
  Processes a request from the Host and sets transmission of a response. For the QE for AFE version, transmits the measurement result of the force and torque as well. For details, refer to "8.3 Communication Control".

- Zero reset request detection processing
  On detection of pressing of SW1, if the operating mode is E_IDLE, sets E_ZEROREST in the Coil member ope_mode in ModbusData.

- A/D conversion start/stop processing
  If the Coil member ope_mode in ModbusData is changed, the followings are processed based on the new ope_mode.
  — ope_mode: E_MEASUREMENT
  - Sets the DSAD0 offset correction value in the ModbusData Holding register in the register OFCRm.
  - Sets starting of DTC transfer of the A/D conversion result.
  - Starts A/D conversion.
  - Turns LED1 off.

  — ope_mode: E_ZERORESET
  - Sets 0 in the register OFCRm.
  - Initializes the zero-reset parameter.
  - Set starting of DTC transfer of the A/D conversion result.
  - Starts A/D conversion.
  - Turns LED1 off.

  — ope_mode: E_IDLE
  - Stops A/D conversion.
  - Sets stopping of DTC transfer of the A/D conversion result.
  - Turns LED1 on.

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

8.2 Functions and Settings of MCU 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 force sensor 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	Acquisition of A/D conversion result	Acquisition of A/D conversion result Communication with Modbus host
CRC	-	Communication with Modbus host
PC3	LED1 ON/OFF control	
PC4	Detection of SW1 state	
P31	Setting RS-485 driver transmission or reception	
E2DataFlash	Saving retention parameters	

Table 8-3 Pins

Pin name	I/O	Use
AIN2	I	Negative input signal for sensor output
AIN4	I	Positive input signal for sensor output 1
AIN5	I	Positive input signal for sensor output 2
AIN6	I	Positive input signal for sensor output 3
AIN7	I	Positive input signal for sensor output 4
AIN8	I	Positive input signal for sensor output 5
AIN9	I	Positive input signal for sensor output 6
REF0P	I	DSAD0 positive side reference voltage
REF0N	I	DSAD0 negative side reference voltage
P26/TXD1	O	UART1 transmission pin
P30/RXD1	I	UART1 reception pin
P31	O	RS-485 driver transmission/reception switching control pin
PC3	O	LED1 ON/OFF control pin
PC4	I	SW1 input 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 Force Sensor Measurement

DSAD0 is used for A/D conversion of the force sensor output, and DTC is used to acquire the A/D conversion result. Table 8-5 and Table 8-6 show the settings of AFE and DSAD0 based on the measurement conditions in Table 6-1, and Table 8-7 shows the settings of DTC.

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)	Enable, Priority: Level 0(disabled)					
	Enable ΔΣA/D channel change interrupt (CHCHG0)	Disable					
Voltage fault and disconnection setting		Not used					
Analog input channel setting		0	1	2	3	4	5
Analog input setting	Positive input signal	AIN4	AIN5	AIN6	AIN7	AIN8	AIN9
	Negative input signal	AIN2					
	Reference input	REF0P/REF0N					
	Positive reference voltage buffer	Disable					
	Negative reference voltage buffer	Disable					
Amplifier setting	Amplifier selection	PGA					
	PGA gain setting	x8					
ΔΣA/D conversion setting	A/D conversion mode	Normal operation					
	Data format	Two's complement					
	A/D conversion number	1					
	First stage oversampling ratio	32					
	Second stage oversampling ratio	Not used					
	Set offset 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					

Table 8-6 AFE Settings

Item		Setting
Bias output setting	Enable bias voltage output	Enable
	AIN2 output pin	Disable
	AIN10 output pin	Disable
Excitation current output setting		Disable
Low level voltage detection setting		Disable
Low-side switch control setting		Disable

Table 8-7 DTC Settings (Config_DTC_DSAD0)

Item		Setting
Base 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	DSAD0 (ADI0)
	Chain transfer	Not used
Transfer mode setting		Repeat mode
Transfer data size setting		32bit
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	0x000A1070 (DSAD0.DR) Address fixed
	Destination address	(Set by the program) Address incremented
	Count	12

8.2.2 Communication

To communicate with QE for AFE or the Modbus host, SCI1 is used for transmission/reception in the asynchronous mode. To switch RS-485 driver between transmission and reception, P31 is used.

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

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

The following shows the setting conditions for each peripheral function.

Table 8-8 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	4,000,000bps	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 8-9 P31 Setting

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

Table 8-10 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 8-11 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 TCNT overflow interrupt (OVI0)	Disabled
	Priority	Level 1

Table 8-12 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	0x00088208 (TMR0.TCNT) Address fixed
	Count	256	1	1

Table 8-13 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 8-14 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 destination			
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 8-15 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 LED1 and SW1

PC3 is used to turn LED1 on and off. PC4 is used to get the state of SW1.

Table 8-16 shows the settings for each port.

Table 8-16 PORTC Settings

Item	Setting	
Port selection	PORTC	
Used port	PC3	PC4
Setting	Out CMOS output Output 1	In

8.2.4 E2 Data Flash

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

Table 8-17 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.

8.3 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. Measured value packet setting is performed based on flag_update which is set in Force/torque conversion processing.

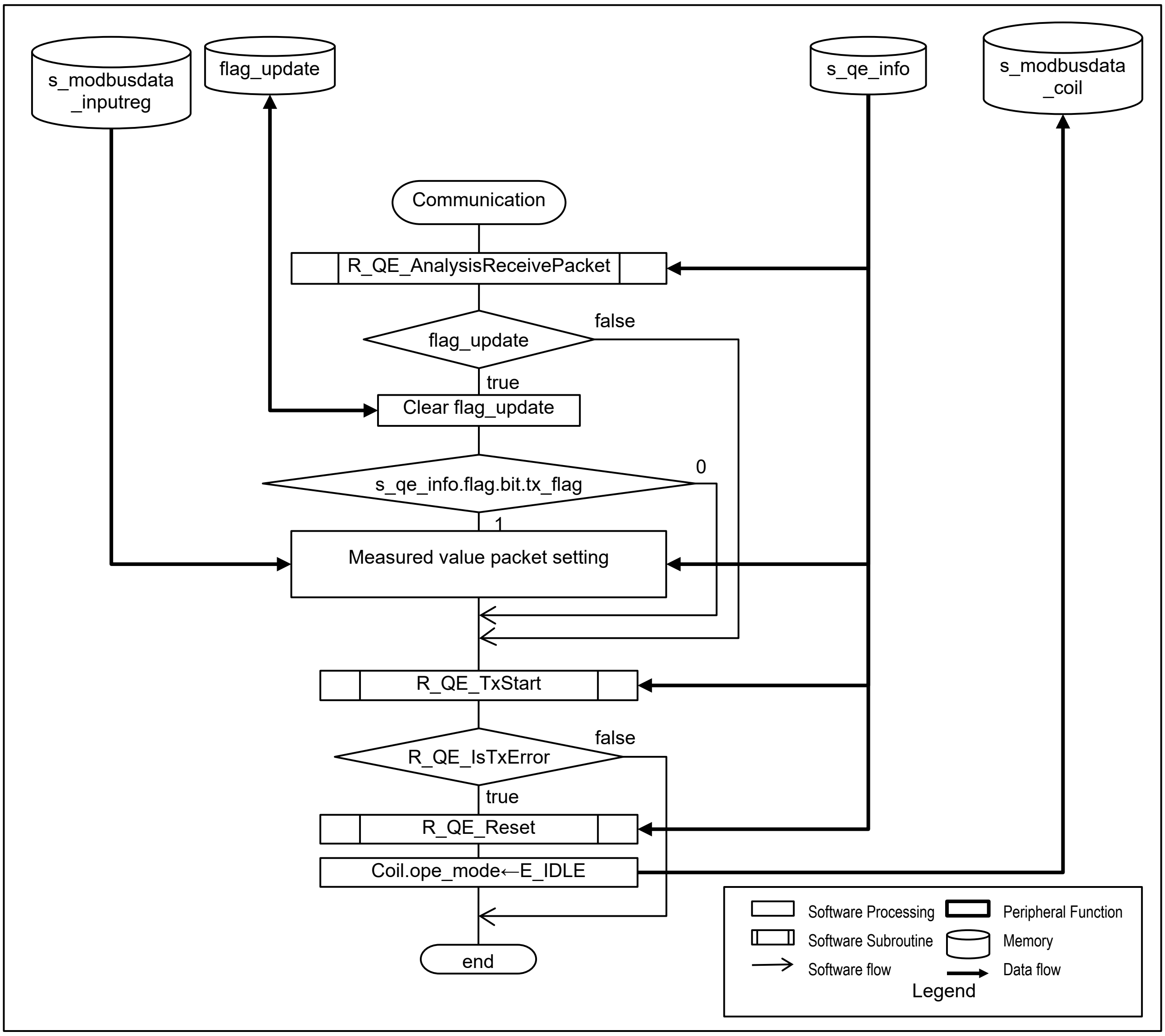


Figure 8-2 QE for AFE Communication Process Flow

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.

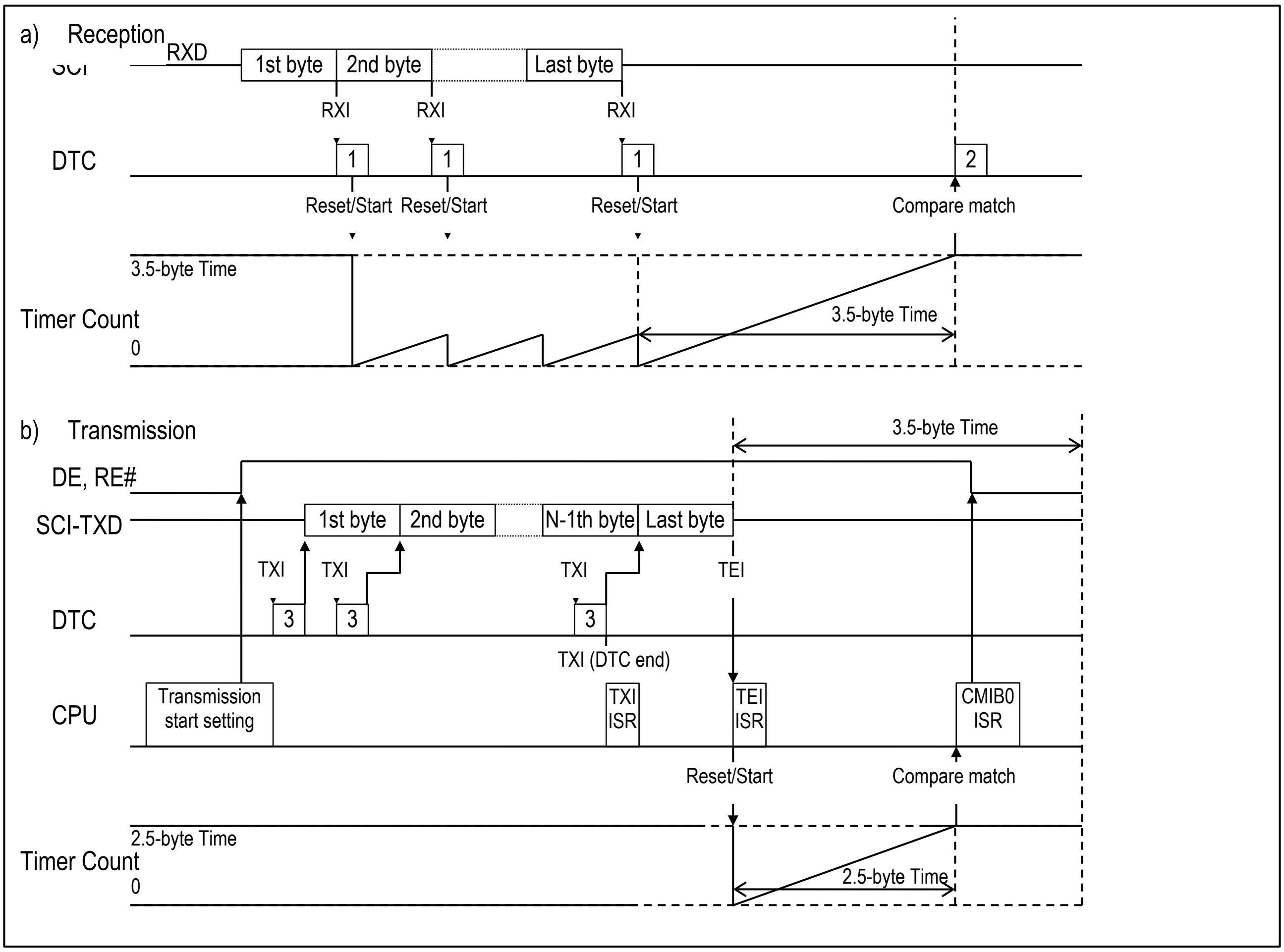


Figure 8-3 Modbus Communication Timing Chart

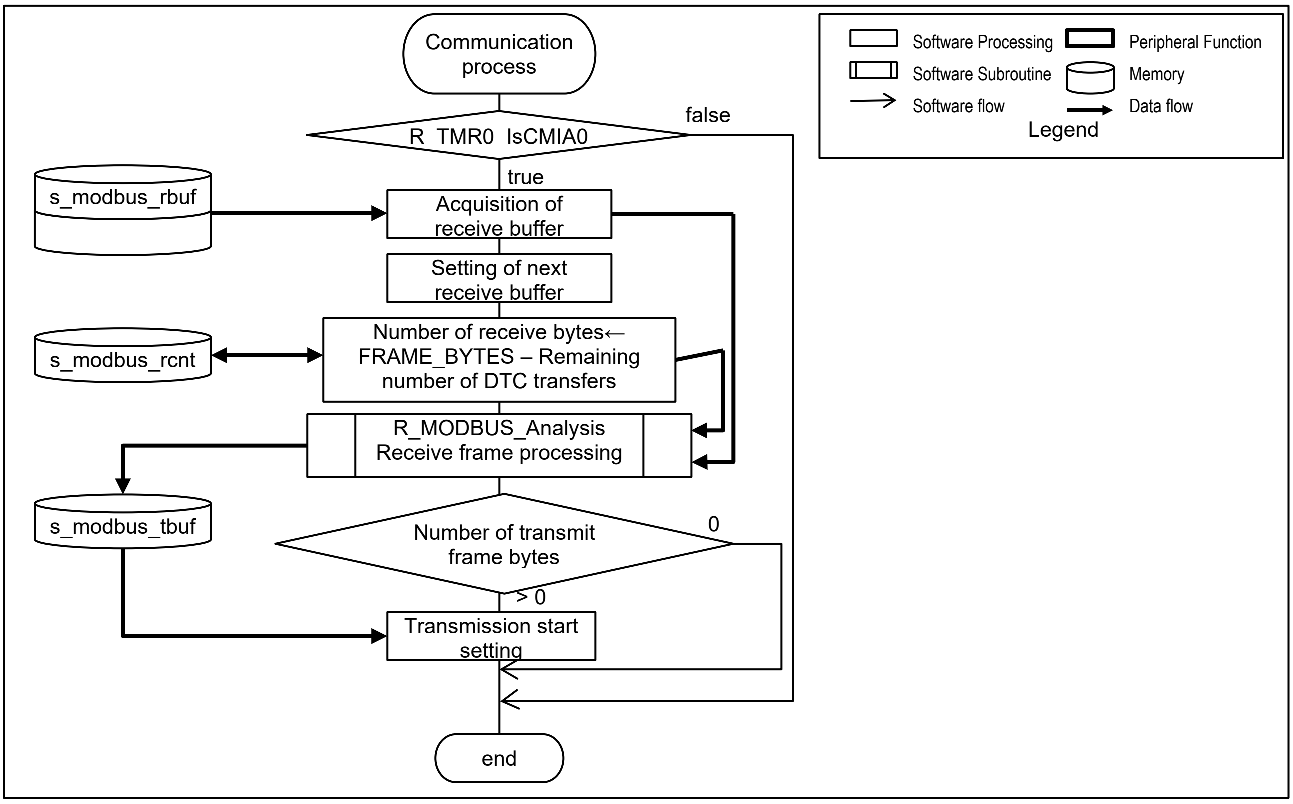


Figure 8-4 Modbus Communication Process Flow

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 TXI1, set a transmit buffer and the number of transmit bytes in DTC transfer 3 (DTC_TXI1) and permit transfer
    - Make SCI1 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

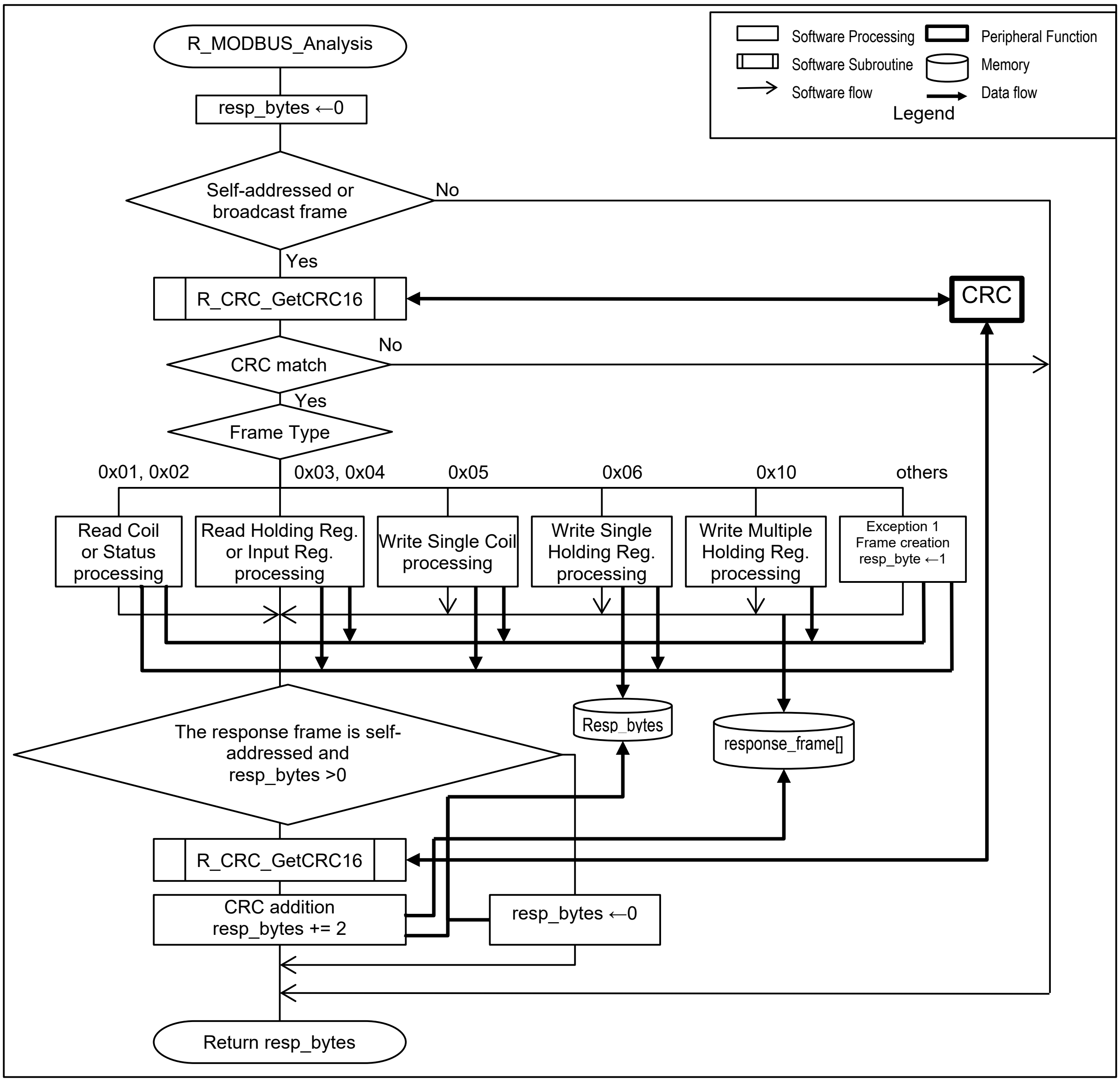


Figure 8-5 Modbus Receive Frame Processing Flow (1)

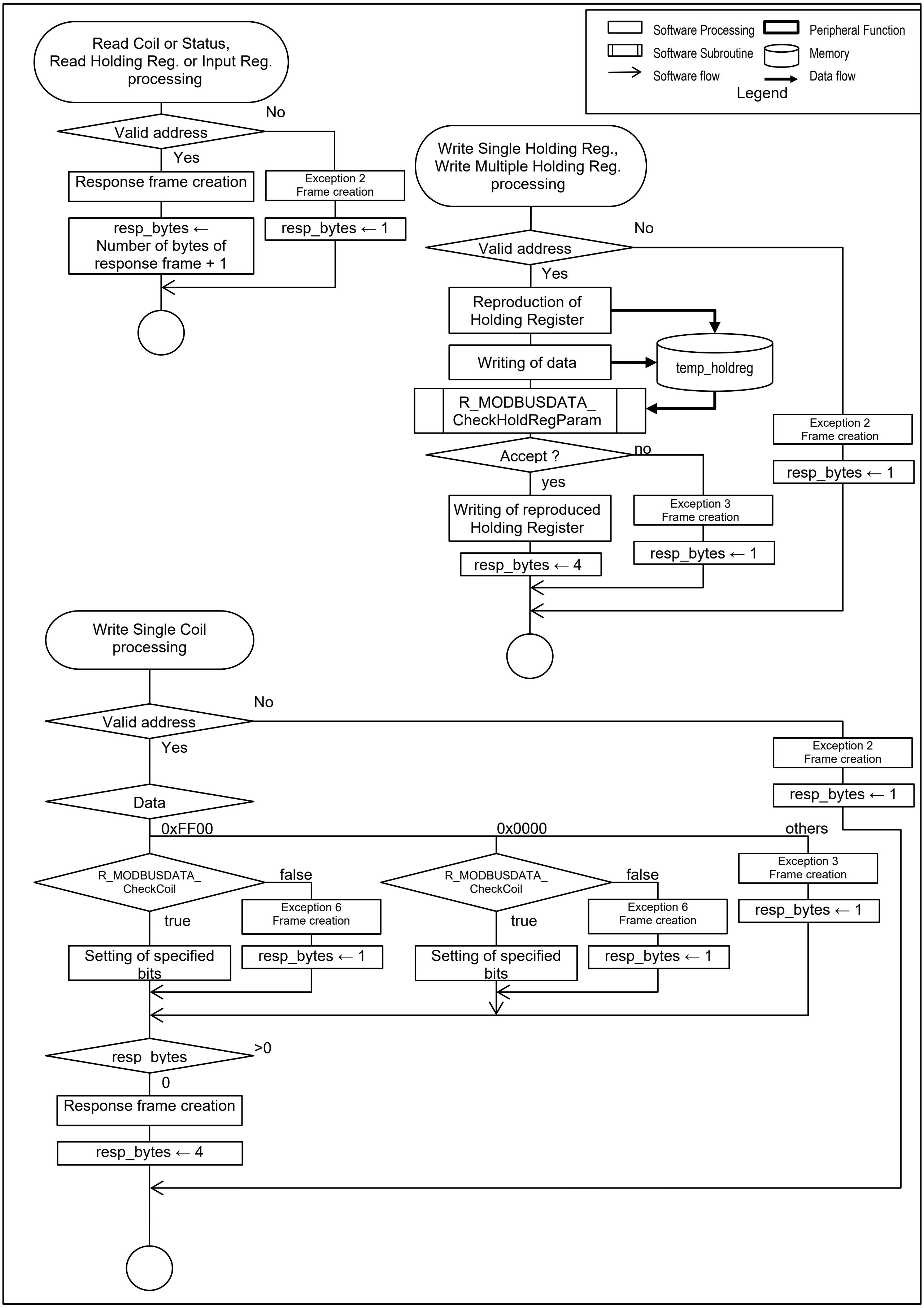


Figure 8-6 Modbus Receive Frame Processing Flow (2)

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
dsplib-rxv2	RX DSP library file	
src		
├ smc_gen	Generated by Smart Configurator	
│ ├ general	Generated by Smart Configurator	Generated by Smart Configurator
│ ├ r_bsp		
│ ├ r_config		
│ └ r_pincfg		
│ ├ Config_AFE	Force sensor measurement setting	
│ ├ Config_DSAD0		
│ ├ Config_DTC_DSAD0	A/D conversion result transfer	
│ ├ 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	Settings for LED, SW, and RS-485 transmission/reception switching	
│ ├ r_flash_rx	Flash API	
├ main.c	Main function	
├ r_fs_cfg.h	Force sensor measurement processing	
├ r_fs_api.c		
├ r_fs_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

Project name	rx23eb_loadcell_qe		rx23eb_loadcell_modbus	
Additional definition	-define D_CFG_QE_TOOL_USE		None	
Additional section definition	**Address**	**Section name**	**Address**	**Section name**
	-	B_MODBUS_HOLDREG_1	Same as the left	
	0x00100000	C_DATAFLASH_1	Same as the left	
	0x00003000	B_DMAC_REPEAT_AREA_1	-	-

8.4.3 Macro Definitions

Table 8-21 r_modbusdata_cfg.h Definitions

Definition name	value	Description
D_MODBUSDATA_CFG_ZERORESET _AVERAGE_DEFAULT	0	Initial value of averaging count for zero reset processing
D_MODBUSDATA_CFG_ZERORESET_MIN	64	Minimum averaging count for zero reset processing
D_MODBUSDATA_CFG_ZERORESET_MAX	512	Maximum averaging count for zero reset processing

Table 8-22 r_fs_cfg.h Definitions

Definition name	value	Description
D_FS_CFG_VREF	5.0F	DSAD0 reference voltage V_{REF} [V]
D_FS_CFG_DSADRES	24	A/D value resolution [bits]
D_FS_CFG_CHANNELS	6	Number of channels DSAD0 uses
D_FS_CFG_CONVMATRIX	Identity matrix	Force sensor voltage-load conversion matrix

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

Definition name	value	Description
D_MODBUS_CFG_ADDRESS	0x01	Modbus slave address

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

Definition name	value	Description
D_QE_CFG_TX_RINGBUF_SIZE	512U	Transmission ring buffer size [byte]
D_QE_CFG_RX_RINGBUF_SIZE	512U	Reception ring buffer size [byte]
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	0	
D_QE_CFG_USER_VAL2	0	
D_QE_CFG_USER_VAL3	0	
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 settings 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	0x3	
D_QE_CFG_CH2	0x3	
D_QE_CFG_CH3	0x3	
D_QE_CFG_CH4	0x3	
D_QE_CFG_CH5	0x3	
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 6-axis force sensor 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 for reception
D_QE_CFG_DMAC_TX	1	DMAC channel for transmission
D_QE_CFG_CMT	0	CMT number for timeout detection

8.4.4 Structures, Unions, and Enumeration Types

Table 8-25 main.c List

Structure type name	st_zeroreset_param_t		
Description	Zero reset parameters		
Member	**Type**	**Name**	**Description**
	uint16_t	num	Averaging count
	uint16_t	count	Input count
	float	sum[D_FS_CFG_CHANNELS]	Total value storage array

Table 8-26 r_modbusdata_api.h List (1/3)

Structure 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
Structure 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
Structure 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
Structure 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
Enumeration type name	e_opemode_t		
Description	Operation mode		
Member	**Name**	**Value**	**Description**
	E_IDLE	0	Standby
	E_MEASUREMENT	1	Measurement
	E_ZERORESET	2	Zero reset

Table 8-27 r_modbusdata_api.h List (2/3)

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:2	ope_mode	Operating mode bit group
	struct	flag	Each bit
	uint32_t:1	measure	Measurement mode 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	dsad0_ovf	DSAD0 CH0 Overflow bit
	uint32_t:1	dsad0_err	DSAD0 CH0 Error bit
	uint32_t:1	dsad1_ovf	DSAD0 CH1 Overflow bit
	uint32_t:1	dsad1_err	DSAD0 CH1 Error bit
	uint32_t:1	dsad2_ovf	DSAD0 CH2 Overflow bit
	uint32_t:1	dsad2_err	DSAD0 CH2 Error bit
	uint32_t:1	dsad3_ovf	DSAD0 CH3 Overflow bit
	uint32_t:1	dsad3_err	DSAD0 CH3 Error bit
	uint32_t:1	dsad4_ovf	DSAD0 CH4 Overflow bit
	uint32_t:1	dsad4_err	DSAD0 CH4 Error bit
	uint32_t:1	dsad5_ovf	DSAD0 CH5 Overflow bit
	uint32_t:1	dsad5_err	DSAD0 CH5 Error bit

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

Union type name	u_modbusdata_inputreg_t		
Description	Modbus Input register		
Member	**Type**	**Name**	**Description**
	uint16_t	reg[24]	Access in register units
	struct	member	Each register definition
	u_modbus_float_t	fx	X-axis force
	u_modbus_float_t	fy	Y-axis force
	u_modbus_float_t	fz	Z-axis force
	u_modbus_float_t	tx	X-axis torque
	u_modbus_float_t	ty	Y-axis torque
	u_modbus_float_t	tz	Z-axis torque
	u_modbus_long_t	dsad0_ad	CH0 A/D value
	u_modbus_long_t	dsad1_ad	CH1 A/D value
	u_modbus_long_t	dsad2_ad	CH2 A/D value
	u_modbus_long_t	dsad3_ad	CH3 A/D value
	u_modbus_long_t	dsad4_ad	CH4 A/D value
	u_modbus_long_t	dsad5_ad	CH5 A/D value
	struct	params	Internal access definition
	float	result[6]	Matrix calculation output
	int32_t	adval[6]	A/D value array
Union type name	u_modbusdata_holdreg_t		
Description	Modbus Holding register		
Member	**Type**	**Name**	**Description**
	uint16_t	reg [(D_FS_CFG_CHANNELS * D_FS_CFG_CHANNELS + D_FS_CFG_CHANNELS) * 2 + 1]	Access in register units
	struct	member	Each register definition
	u_modbus_float_t	matrix [D_FS_CFG_CHANNELS] [D_FS_CFG_CHANNELS]	Voltage-load conversion matrix
	u_modbus_long_t	ofcr0	DSAD0 CH0 offset correction value
	u_modbus_long_t	ofcr1	DSAD0 CH1 offset correction value
	u_modbus_long_t	ofcr2	DSAD0 CH2 offset correction value
	u_modbus_long_t	ofcr3	DSAD0 CH3 offset correction value
	u_modbus_long_t	ofcr4	DSAD0 CH4 offset correction value
	u_modbus_long_t	ofcr5	DSAD0 CH5 offset correction value
	u_modbus_ushort_t	average	Averaging count for zero reset
	struct	params	Internal access definition
	float	matrix [D_FS_CFG_CHANNELS] [D_FS_CFG_CHANNELS]	Voltage-load conversion matrix
	int32_t	ofcrs [D_FS_CFG_CHANNELS]	Offset correction value array for each channel of DSAD0
	uint16_t	average	Averaging count for zero reset

8.4.5 Functions

8.4.5.1 Common Functions

Table 8-29 main.c

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

Table 8-30 r_fs_api

Function name	R_FS_DsadToVoltage			
Description	Converts an A/D value to voltage			
Argument	I/O	Type	Name	Description
	I	float	dsad	A/D value
	I	float	gain	PGA gain
Return value	O	float	Voltage [V]	

Table 8-31 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 Modbus Input register			
Argument	I/O	Type	Name	Description
	-	void	-	-
Return value	O	u_modbusdata_inputreg_t *	Pointer to Modbus Input register	
Function name	R_MODBUSDATA_GetHoldRegPtr			
Description	Acquires a pointer to Modbus Holding register			
Argument	I/O	Type	Name	Description
	-	void	-	-
Return value	O	u_modbusdata_holdreg_t *	Pointer to Modbus Holding register	

Table 8-32 r_modbusdata_api (2/2)

Function name	**R_MODBUSDATA_LoadHoldReg**			
Description	Initializes the Modbus Holding Register and loads the values 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	Set 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 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 8-33 Config_DSAD0 User-Defined Functions

Function name	**R_DSAD0_IsScanEnd**			
Description	Detects DSAD0 channel scan end (SCANEND0)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Detected false: Not detected	
Function name	**R_DSAD0_CONV_SIGNED_VALUE**			
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_GET_ERROR_FLAGS**			
Description	Extracts the ERR flag and 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.ERR flag and DR.OVF flag	
Function name	**R_DSAD0_GetScanRate**			
Description	Calculates the channel scan rate of the enabled channel of DSAS0			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	float	Channel scan rate [scan/s]	
Function name	**R_Config_DSAD0_SetOFCR**			
Description	Set a value to OFCRm register of DSAD0			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	ch	Channel m to which to set a value
	I	int32_t	ofs	OFCRm register setting value
Return value	O	bool	true: Successful false: Failed	
Function name	**R_Config_DSAD0_GetOFCR**			
Description	Acquires the OFCRm register value of DSAD0			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	ch	Channel m to which to set a value
Return value	O	int32_t	OFCRm register value	

Table 8-34 Config_DTC_DSAD0 User-Defined Functions

Function name	**R_Config_DTC_DSAD0_SetSrcAddr**			
Description	Sets the source address of DTC transfer			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	void *	addr	Source address
Return value	-	void	-	
Function name	**R_Config_DTC_DSAD0_ResetCount**			
Description	Resets the remaining DTC transfer count			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	-	void	CRA address	
Function name	**R_Config_DTC_DSAD0_GetCount**			
Description	Acquires the remaining DTC transfer count			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	uint8_t	Remaining DTC transfer count	

Table 8-35 Config_PORT User-Defined Functions

Function name	**R_CONFIG_PORT_LED1_ON**			
Description	Turns LED ON/OFF (macro functions)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	flag	true: ON false: OFF
Return value	-	void	-	
Function name	**R_PORT_GetSwitchOn**			
Description	Acquires the state of SW1			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: On false: Off	
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	-	

8.4.5.2 QE for AFE Version

Table 8-36 r_qe_api_user.c User-Defined Processes

User processes only

Function name	**r_QE_WriteUser**
Description	If ope_mode is E_IDLE, accepts and sets coil.flag.register_write
Function name	**r_QE_RunUser**
Description	If ope_mode is E_IDLE, accepts and sets coil.flag.measure
Function name	**r_QE_StopUser**
Description	Resets coil.flag.measure
Function name	**r_QE_UserValueUser**[Note]
Description	Judges to be accepted or not for each User Value No., and if accepted, updates the value of the corresponding Holding register
Function name	**r_QE_ExSpsInfoUser**
Description	Calculates the output rate from the DSAD0 settings and updates SPS information
Function name	**r_QE_ExUseButtonStatusUser**[Note]
Description	Judges 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 8-37 r_qe_api.c Processing Modification

Modification 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 8-38 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 store the transmission frame
Return value	O	uint32_t	Number of bytes of the transmission frame	

Table 8-39 Config_CRC User-Defined Functions

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 8-40 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 8-41 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 8-42 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 8-43 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 e² studio. Pictures may be different depending on the version of e² studio to be used.

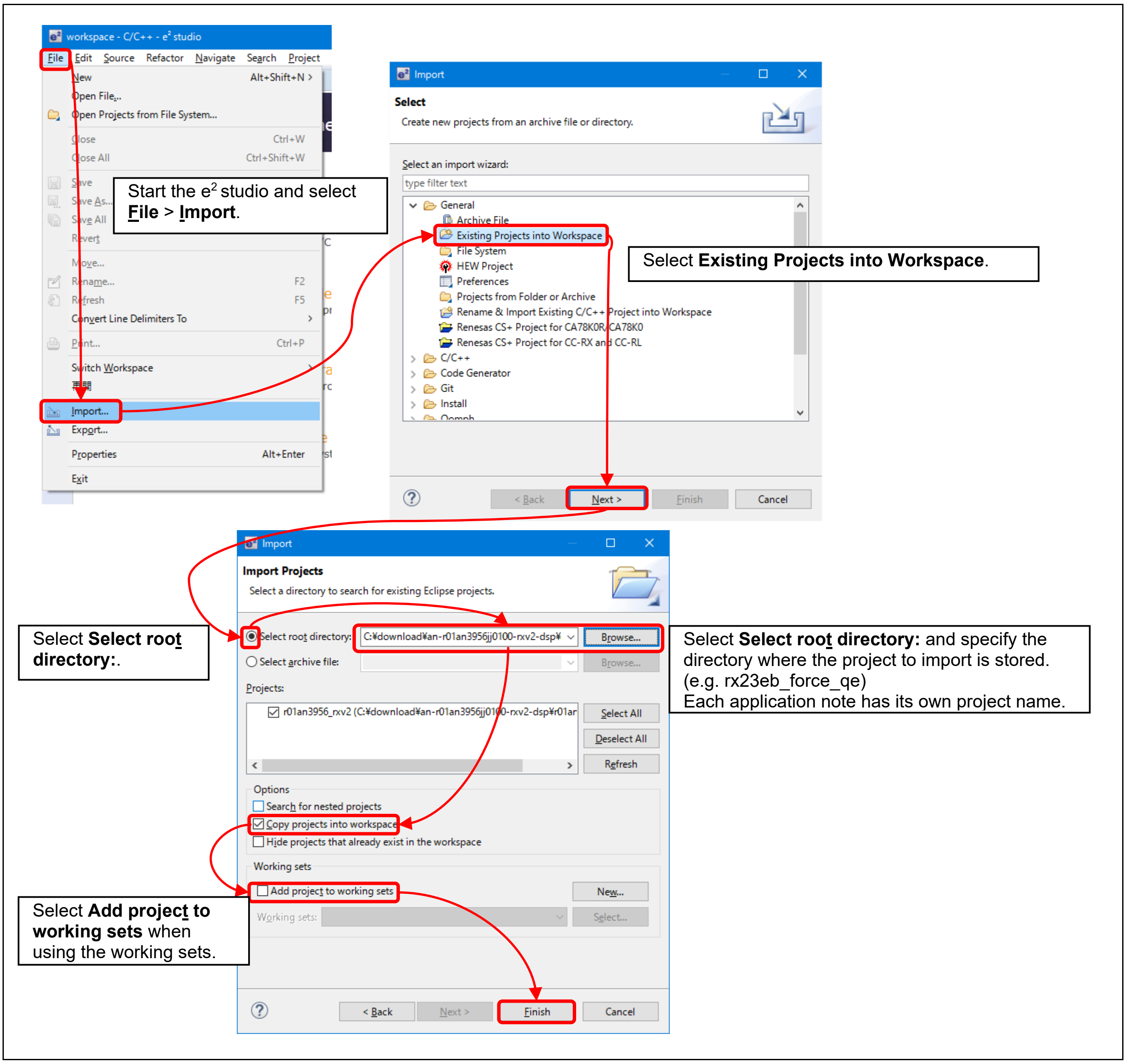


Figure 9-1 Importing a project into e² studio

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.

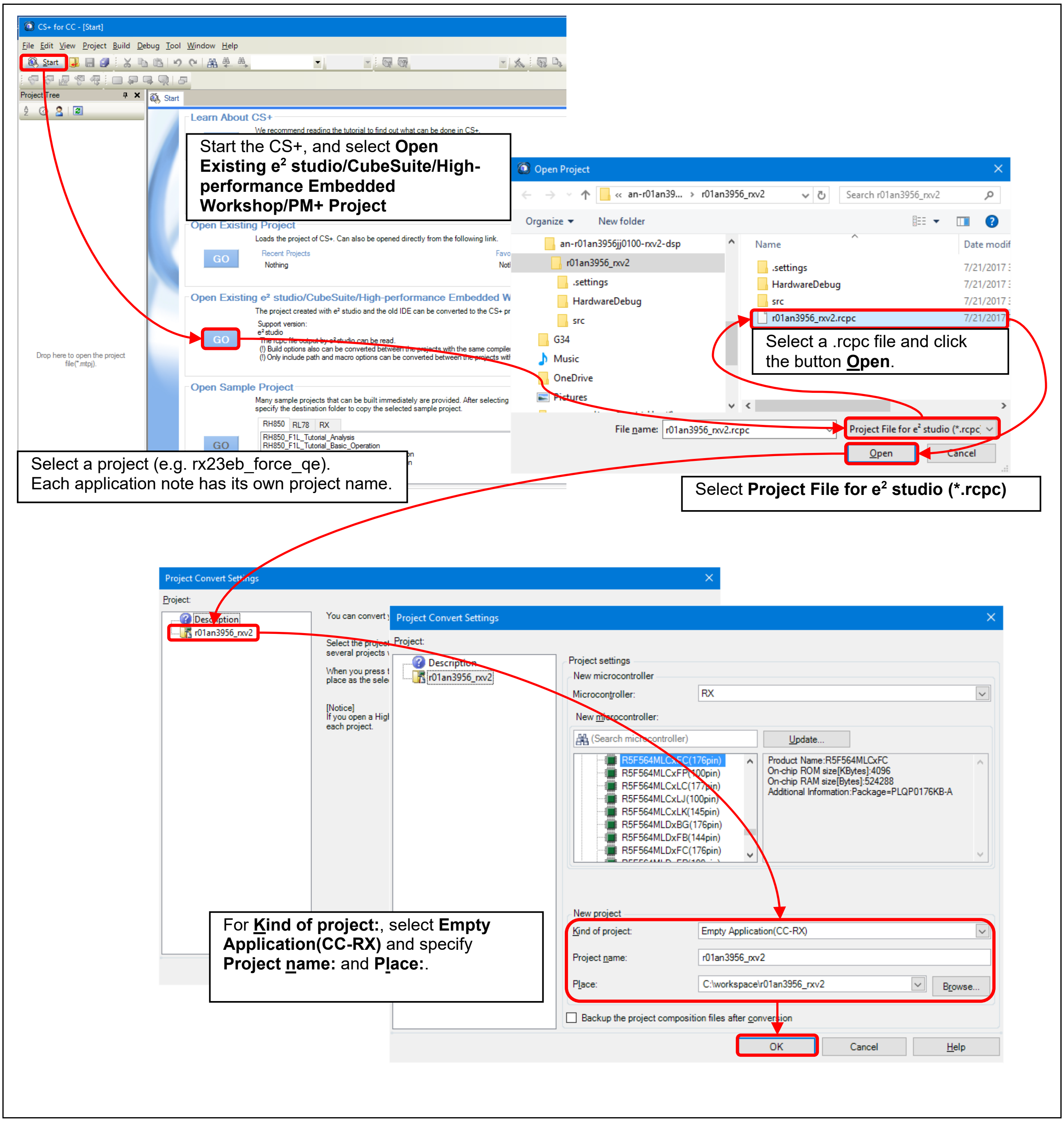


Figure 9-2 Importing a project into CS+

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

The Renesas Solution Starter Kit for RX23E-B (hereafter, 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.

This sample program can be run on the RSSKRX23E-B board by changing settings with the Smart Configurator. Communication can be conducted via the USB-UART conversion IC.

The following explains an example of connection between the RSSKRX23E-B board and the force sensor and the procedure for changing the sample project.

(1) Connection to RSSKRX23E-B board

Figure 10-1 shows the connection example, Table 10-1 shows the changes to component constants, and Table 10-2 shows the jumper settings. In this connection example, the force sensor output is connected to AIN9, AIN8, AIN7, AIN6, AIN5, and AIN4 as positive input signal of DSAD0. For negative input signal of DSAD0, the RX23E-B bias voltage generator VBIAS is used to apply the half voltage of AVCC0 to AIN10. To measure the sensor output, the voltages of AIN9, AIN8, AIN7, AIN6, AIN5, and AIN4 for AIN10 are sequentially A/D converted by using the channel function of DSAD0, and the A/D conversion results of 6 channels are converted into the force and torque.

If a voltage different from AVCC0 is applied to the force senser, or if a high-precision reference voltage is required, generate the voltage by an external circuit and input it to AIN10. If the external voltage is input to AIN10, it is necessary to disable the VBIAS output of the RX23E-B, mount 0Ω resistors on R175 and R196, and remove R197.

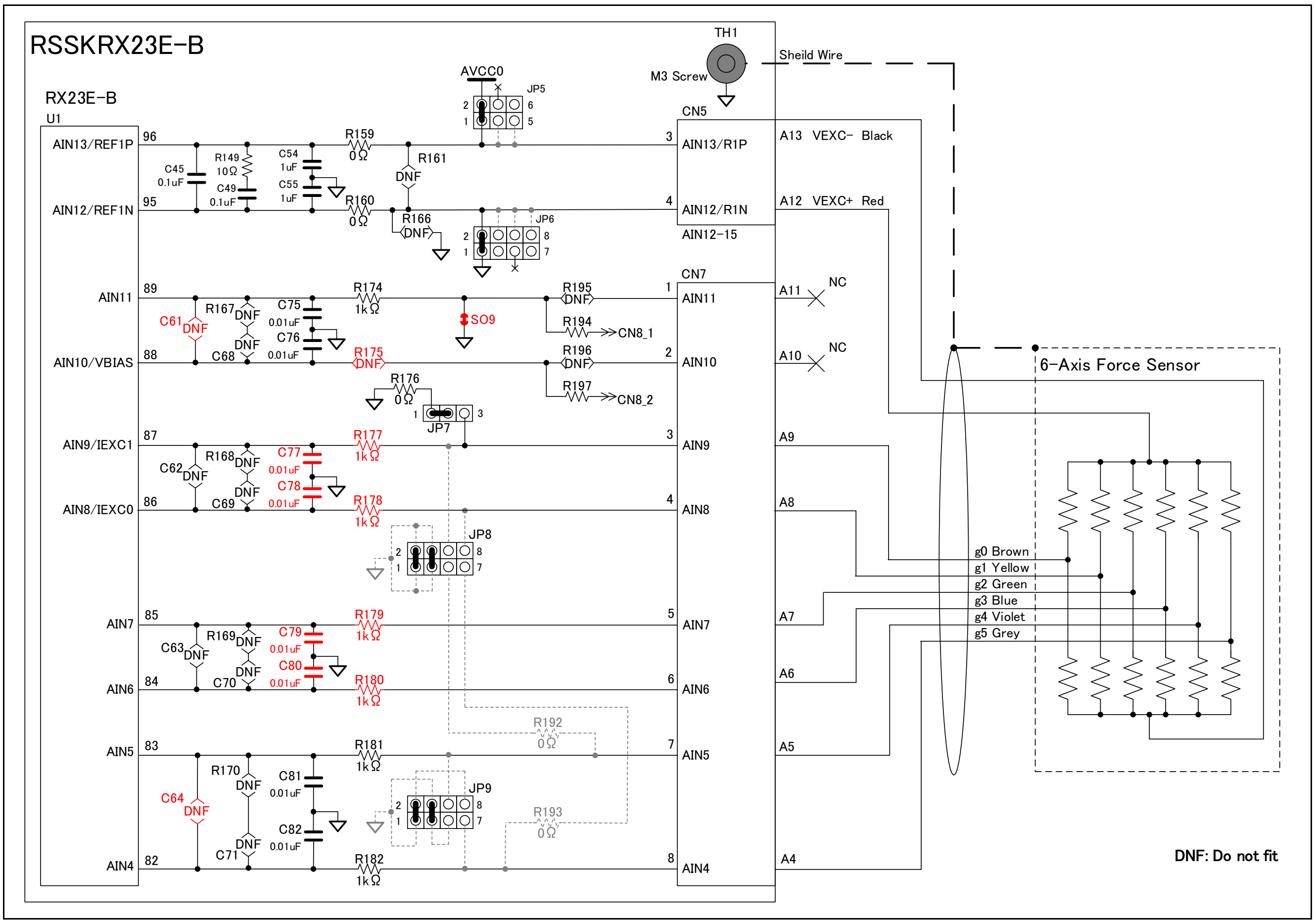


Figure 10-1 Example of connecting RSSKRX23E-B Board to Force Sensor

Table 10-1 Changes to RSSKRX23E-B Board for Force Sensor Connection

Circuit Code	Before change	After change
R175	1kΩ	DNF
R177, R178, R179, R180	0Ω	1kΩ[Note]
C61, C64	0.1μF	DNF
C77, C78, C79, C80	DNF	0.01μF[Note]
SO9	Open	Short

Note: The component constants are reference values. Change them based on operating conditions and target performance.

Table 10-2 Jumper Settings for RSSKRX23E-B Board for Force Sensor Connection

Function	Code	Connection	Setting
External reference REF1P selection	JP5	1-2	Input AVCC0 to REF1P
External reference REF1N selection	JP6	1-2	Input AVSS0 to REF1N
AIN9 onboard RTD connection selection	JP7	1-2	AIN9: No connection to RTD1
3-wire RTD connection selection	JP8	1-2 and 3-4	AIN8, AIN9: No connection to AIN4, AIN5
AIN4, AIN5 onboard RTD connection selection	JP9	1-2 and 3-4	AIN5, AIN4: No connection to RTD

(2) 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.

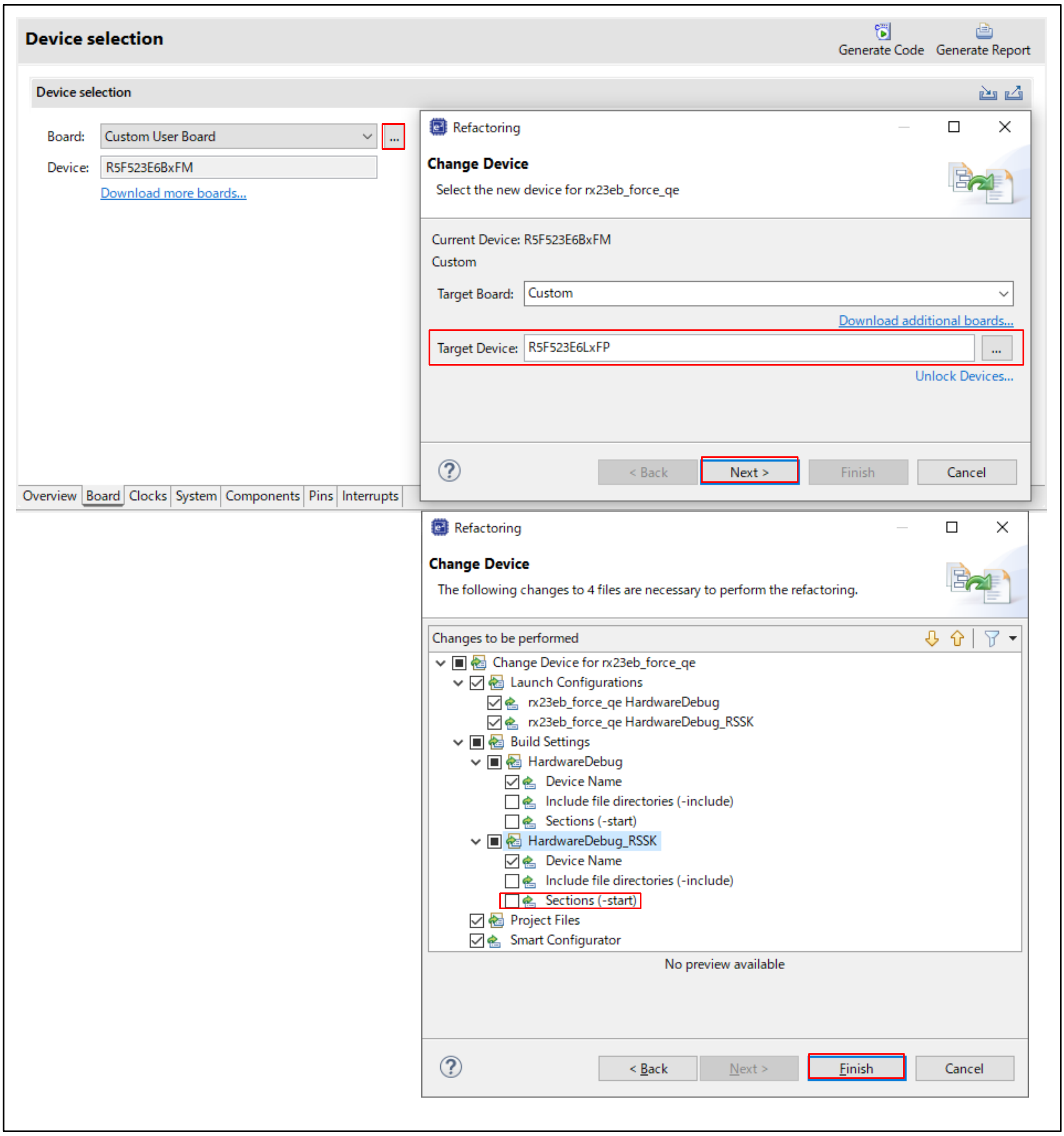


Figure 10-2 Changing the Target Device

(3) Change AFE and DSAD settings

On the "Component" tab, change the settings for DSAD0 and AFE, and then generate code.

— DSAD0 setting

Change the analog input settings of channel 0 to 5 with Config_DSAD0 according to Table 10-3. Bold text in the table indicates changes.

Table 10-3 DSAD0 Settings for RSSKRX23E-B

Continuous scan mode

Item		**Setting**					
Analog input channel setting		0	1	2	3	4	5
Analog input setting	Positive input signal	**AIN9**	**AIN8**	**AIN7**	**AIN6**	**AIN5**	**AIN4**
	Negative input signal	**AIN10**					
	Reference input	REF0P/REF0N					
	Positive reference voltage buffer	Disable					
	Negative reference voltage buffer	Disable					

— AFE setting

If using VBIAS for negative input signal of DSAD0, select the bias voltage output pin with Config_AFE as shown in Table 10-4. Bold text in the table indicates changes.

Table 10-4 AFE Settings for RSSKRX23E-B

Item		**Setting**
Bias output setting	Enable bias voltage output	Enable
	AIN2 output pin	Disable
	AIN10 output pin	**Enable**

(4) 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 10-5. In addition, assignment of LED1 port and SW1 port are changed using Config_PORT.h.

Table 10-5 RSSKRX23E-B Pin Used

RX23E-B-QFP64-FT	**RSSKRX23E-B board**			
Pin: Function	**Assignment**	**Pin**	**Initial setting**	**Supplement**
P31: Switches the setting of RS-485 driver between transmit/receive	DE	PC6	Output: L	RAA7881582/DE
PC3: LED1	LED0	P70	Output: H	
-	LED1	P71	Output: H	
-	LED2	P72	Output: H	
-	LED3	P73	Output: H	
PC4: SW1	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

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	
		QE for AFE	modbus
Compiler	Common	-isa=rxv2 -fpu -include="${workspace_loc:/${ProjName}/dsplib-rxv2}" -utf8 -nomessage -output=obj -obj_path=${workspace_loc:/${ProjName}/${ConfigName}} -debug -outcode=utf8 -nologo	
	Difference	-define= D_CFG_QE_TOOL_USE	
Linker	Common	-library="${workspace_loc:/${ProjName}/dsplib-rxv2/RX_DSP_FPU_LE.lib}" -noprelink -form=absolute -nomessage -vect=_undefined_interrupt_source_isr -nooptimize -rom=D=R,D_1=R_1,D_2=R_2 -cpu=RAM=00000000-00007fff, FIX=00080000-00083fff,FIX=00086000-00087fff,FIX=00088000-0008dfff, FIX=00090000-0009ffff,FIX=000a0000-000bffff,FIX=000c0000-000fffff, ROM=00100000-00101fff,FIX=007fc000-007fc4ff,FIX=007ffc00-007fffff, ROM=fffc0000-ffffffff -nologo	
	Difference	-output="rx23eb_force_qe.abs" -list=rx23eb_force_qe.map	-output="rx23eb_force_modbus.abs" -list=rx23eb_force_modbus.map
	Section	SU,SI,B_MODBUSDATA_HOLDREG_1, B_1,R_1,B_2,R_2,B,R/04, B_DMAC_REPEAT_AREA_1/03000, C_DATAFLASH_1/0100000, PResetPRG,C_1,C_2,C,C$*,D*,W*,L, P/0FFFC0000,EXCEPTVECT/0FFFFFF80, RESETVECT/0FFFFFFFC	SU,SI,B_MODBUSDATA_HOLDREG_1, 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

Note: Included paths other than user settings in compiler setting are omitted.

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 ver.	Modbus ver.	
ROM		12394	11888	
	Code	10430	10000	
	Data	1946	1888	
E2 DataFlashROM		170	170	
RAM		13618(9182)	13368(8688)	Note
	Data	8498	8248	
	Stack	5120(684)	5120(440)	Note

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

11.1.3 Number of Execution Cycle and Execution Time

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

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

ICLK=32MHz

Measurement rate: 2275.312856SPS

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	115 (3.53μs)	0.82	115 (3.53μs)	0.82	
Force and torque conversion	546 (17.06μs)	3.88	546 (17.06μs)	3.88	
Communication	729 (22.78μs)	5.18	693 (21.66μs)	5.30	QE: Measurement result transmission process Modbus: 6ch measurement value request process
Others	91 (2.84μs)	0.65	23 (0.72μs)	0.16	
Total	1457 (46.28μs)	10.53	1429 (44.66μs)	10.16	

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

11.2 Measurement Result

11.2.1 Measurement Appearance

Connecting a force sensor based on the configuration in “Figure 6-1 Connection of RX23E-B-QFP64-FT and Force Sensor”, we have performed measurement applying force and torque to the force sensor with evaluation jigs and weights. Figure 11-1 shows the appearance of this measurement.

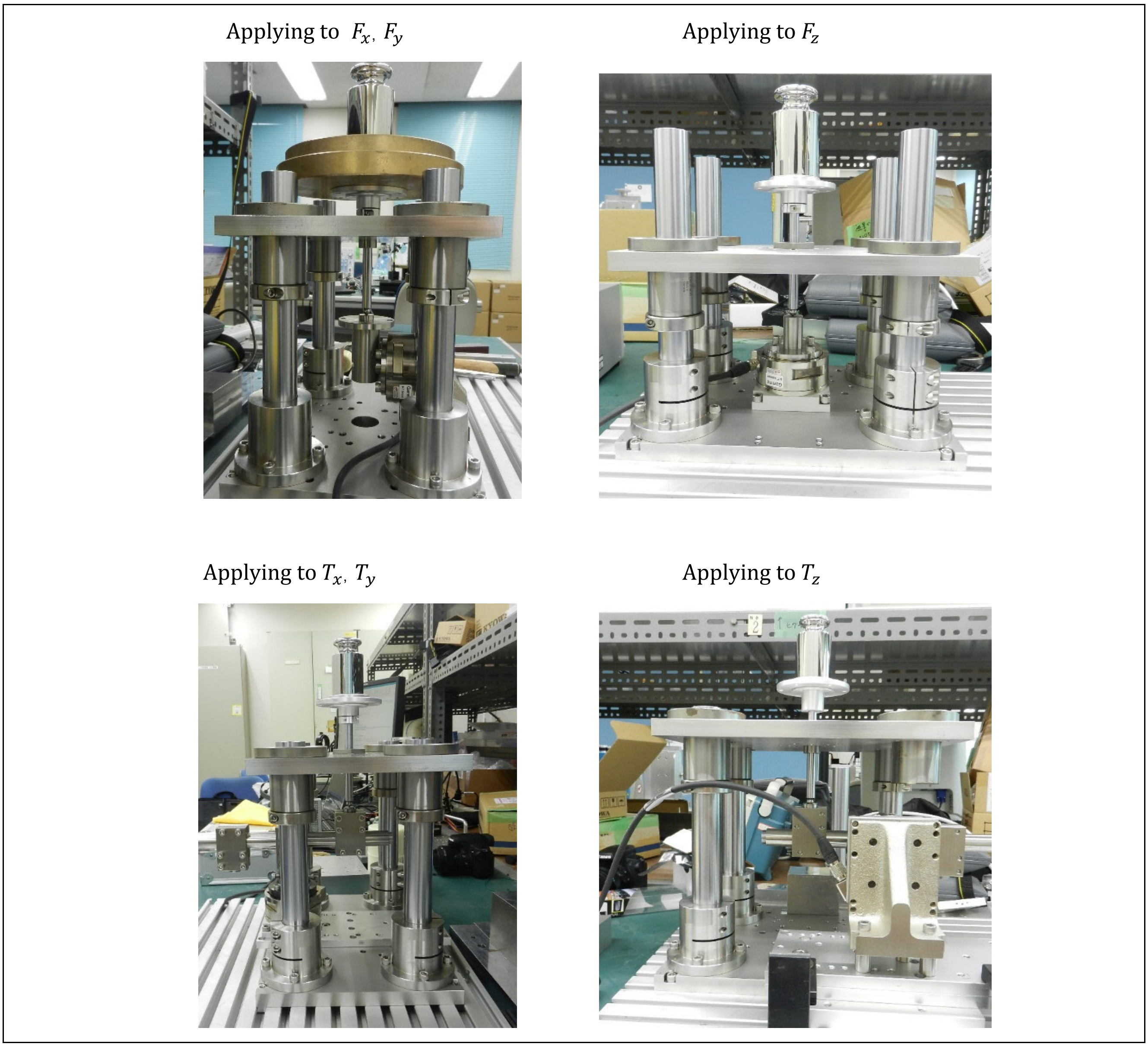


Figure 11-1 Evaluation Jigs

11.2.2 Measurement Conditions

Figure 11-2 and Figure 11-3 show how to apply force and torque, and Figure 11-4 shows the weights used in measurement.

For measurement, Zero-reset is processed in the posture shown in Figure 11-2 and Figure 11-3 at no load.

(1) Force Measurement

Force F [N] applied to a force sensor is calculated from weight m [kg] and gravitational acceleration g [m/s2] with the equation below.

$$F = m \times g$$

(2) Torque Measurement

Torque T [N·m] applied to a force sensor is calculated from weight m [kg], gravitational acceleration g [m/s2], and the distance between a fulcrum and a force point L [m] with the equation below.

$$T = m \times g \times L$$

Suppose that gravitational acceleration g is the standard gravitational acceleration 9.80665 [m/s2].

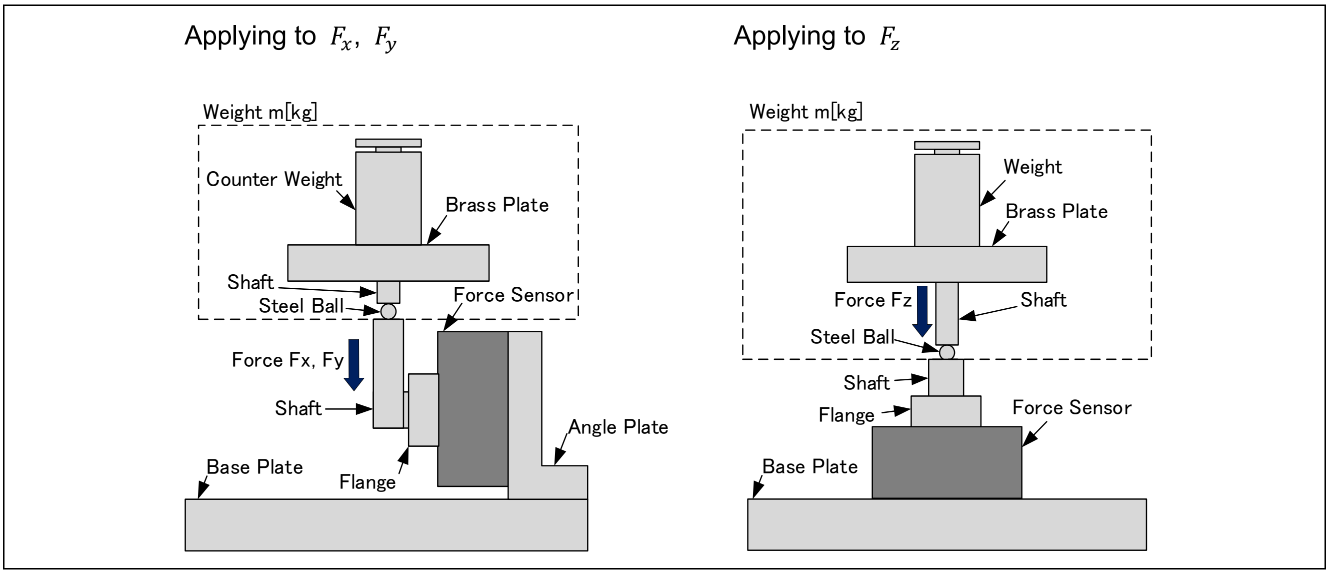


Figure 11-2 How to Apply Force

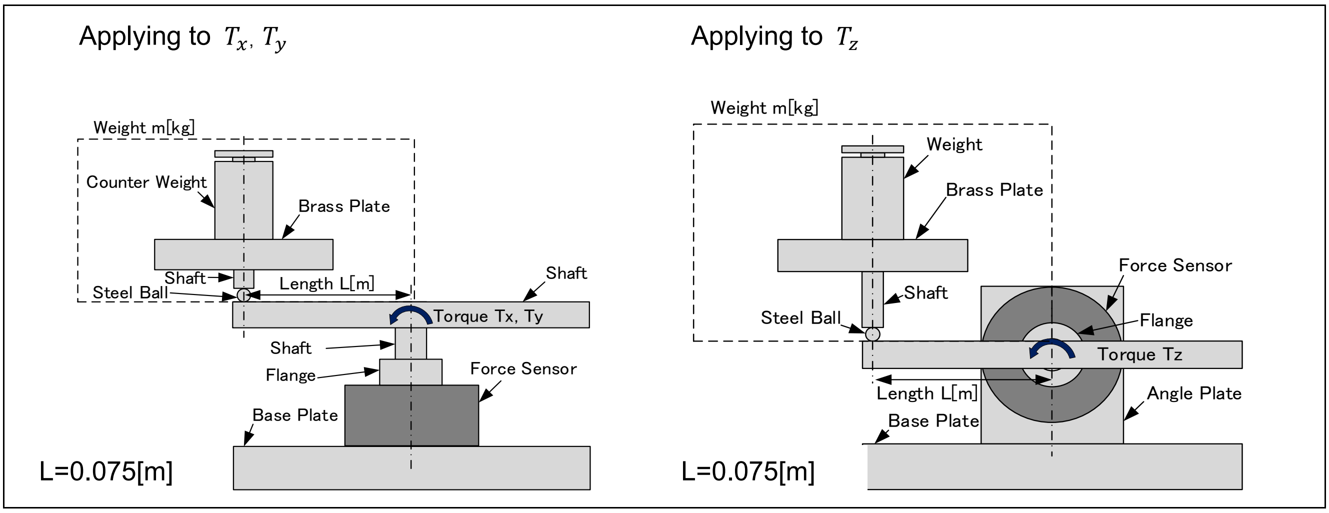


Figure 11-3 How to Apply Torque

Table 11-4 Weight Used in Measurement

No.	Name	Model	Weight	Grade	Manufacturer
1	Weight set	WS1M1K	1mg x1, 2mg x2, 5mg x1 10mg x1, 20mg x2, 50mg x1 100mg x1, 200mg x2, 500mg x1 1g x1, 2g x2, 5g x1 10g x1, 20g x2, 50g x1 100g x1, 200g x2, 500g x1 1kg x1	M1	AS ONE CORPORATION
2	Cylindrical Weight	SWM2000	2kg	M1	AS ONE CORPORATION
3	Brass Plate	INERTIAPLATE: C	2.853kg[Note]	-	Renesas
4	Brass Plate	INERTIAPLATE: D	4.6625kg[Note]	-	Renesas

Note: Confirmed with A&D FC-5000i (A&D Company, Limited).

11.2.3 Measurement Result

The result of force measurement is shown in Figure 11-4, and the result of torque measurement is shown in Figure 11-5. The measurement results are corrected by calculating scale factor error and bias error from the measurement values at no load and at maximum load.

From the measurement result, the force measurement error $E_{F:FS}$ for full-scale is calculated from the force input value F_{in}, the force measurement value F_{mea}, and the force measurement range of the force sensor F_{FS} (F_x, F_y :130N, F_z :400N) with the equation below.

$$E_{F:FS} = \frac{F_{mea} - F_{in}}{F_{FS}} \times 100[\%FS]$$

Similarly, the torque measurement error $E_{T:FS}$ is calculated from the torque input value T_{in}, the torque measurement value T_{mea}, the torque measurement range of the force sensor T_{FS} (T_x, T_y, T_z: 10N·m) with the equation below

$$E_{T:FS} = \frac{T_{mea} - T_{in}}{T_{FS}} \times 100[\%FS]$$

Table 11-5 shows the measurement uncertainty of the force sensor 9105-TWE-Gamma used in this measurement and the full-scale error of this measurement. These errors are indicators showing the linearity of the measurement.

The force measurement error is within ±0.25% FS, and the torque measurement error is within ±1% FS, indicating that these errors are within the measurement uncertainty of the force sensor used in this measurement. Though this result contains not only the error of the circuit and the nonlinearity of the force sensor itself, but also flexure or inclination of the evaluation jigs and the error caused by friction, it is confirmed that this system configuration allows the measurement of the force sensor.

Table 11-5 Measurement Uncertainty

Item	$E_{Fx:FS}$ [%FS]	$E_{Fy:FS}$ [%FS]	$E_{Fz:FS}$ [%FS]	$E_{Tx:FS}$ [%FS]	$E_{Ty:FS}$ [%FS]	$E_{Tz:FS}$ [%FS]
9105-TWE-Gamma SI-130-10 Measurement uncertainty (95% CI)	1.00%	1.25%	0.75%	1.00%	1.25%	1.50%
Result of full-scale error measurement (Worst case)	0.12%	0.05%	0.03%	0.94%	0.92%	0.91%

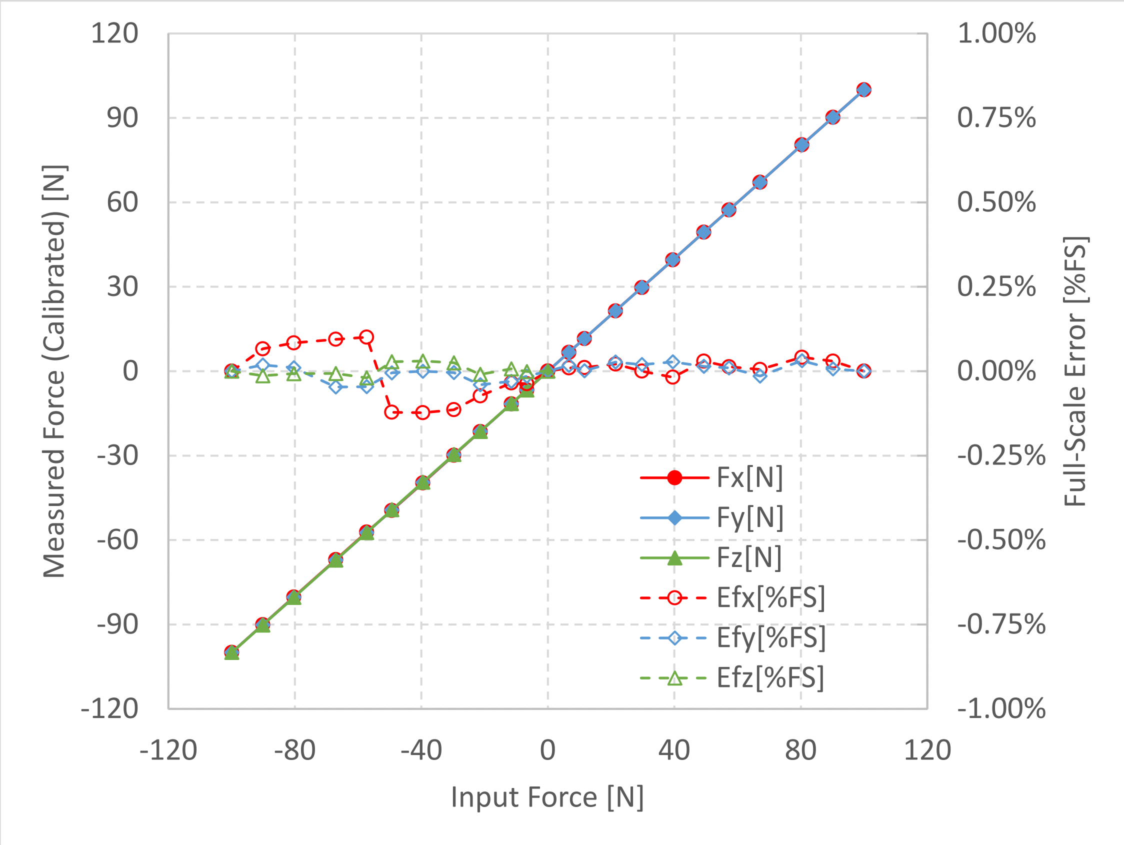


Figure 11-4 Force Measurement Result

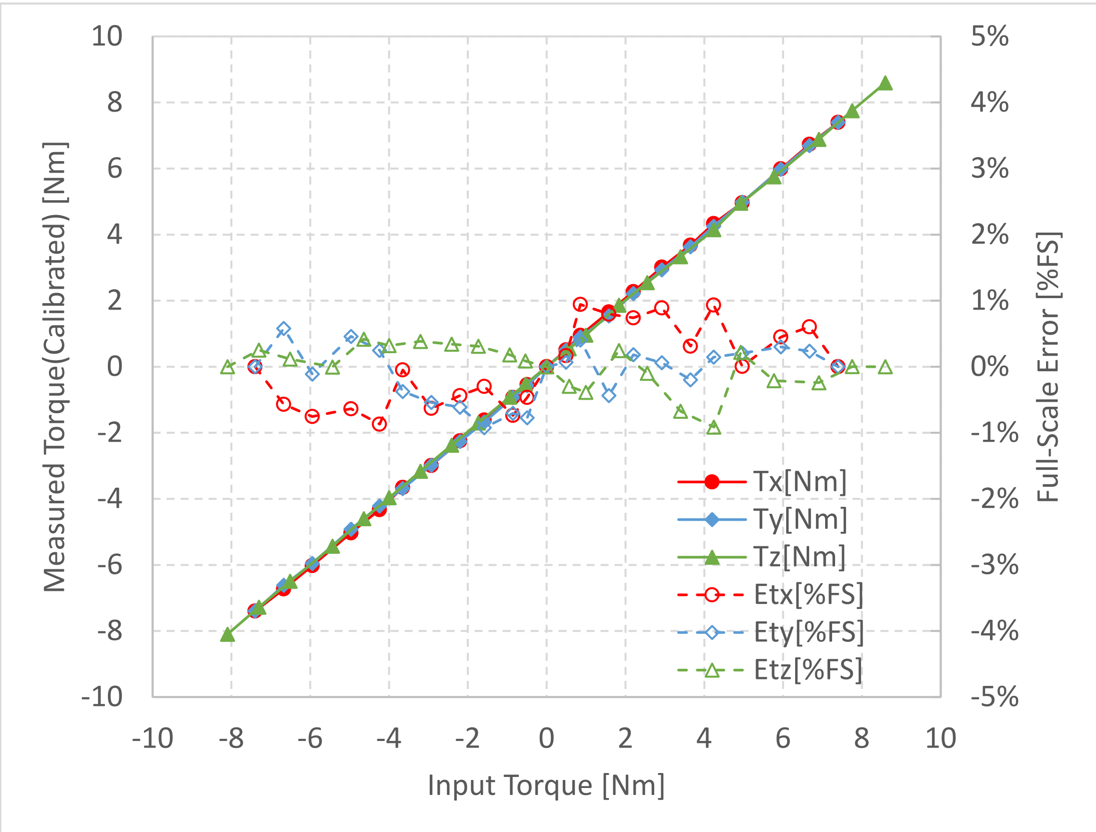


Figure 11-5 Torque Measurement Result

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.23.23	-	First release

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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