RL78/G23

UART Communication Using Hardware Flow Control

Introduction

This application note explains how to implement UART communication with hardware flow control.

Target Device

RL78/G23

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Specifications

This application note executes UART communication with hardware flow control. In hardware flow control, the RTS and CTS signals are managed via ports for each byte of UART transmission and reception.

1.1 Hardware Flow Control Specifications

The configuration of UART communication with hardware flow control is shown in Figure 1-1. Additionally, the pins used and their functions are listed in Table 1-1.

- Control of RTS

RTS control is performed for each byte of data received via UART.

When data reception begins, RTS is set to a high level, signaling to the opposing device that "data cannot be received at the moment." The start of data reception is detected using a key interrupt.

After completing the reception of one byte, the received data is read, and RTS is then returned to a low level, notifying the opposing device that "data can now be received."

- Check of CTS

When transmitting data via UART, the level of CTS is checked for each byte.

If CTS is at a low level, the opposing device is considered able to receive data, and the data is transmitted.

If CTS is at a high level, the opposing device is considered unable to receive data, and the data transmission is temporarily paused. Data transmission is resumed after detecting that CTS has returned to a low level using an external interrupt.

Figure 1-1 Flow Control Block Diagram

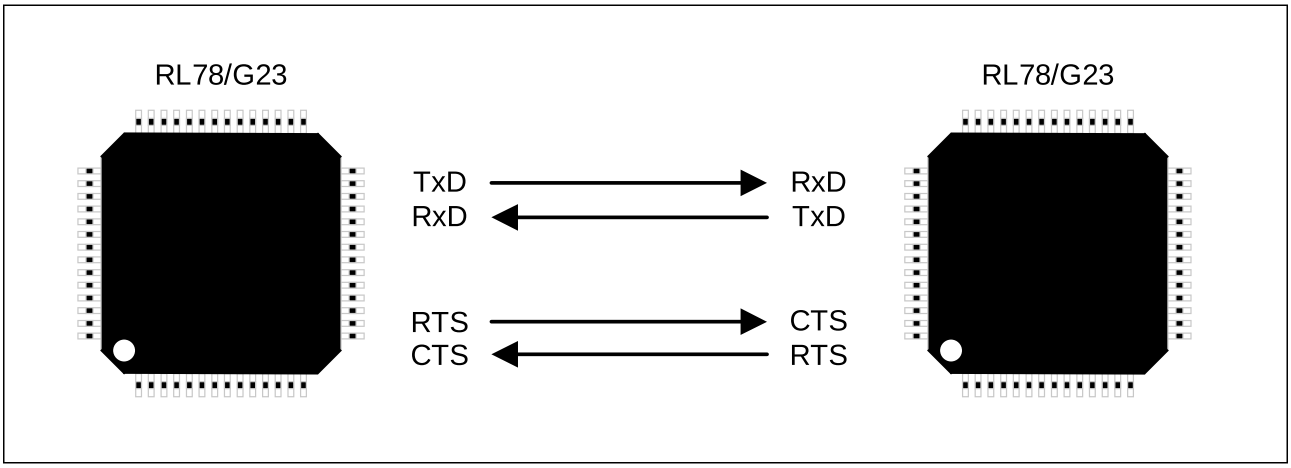


Table 1-1 Pins and Functions Used for Flow Control

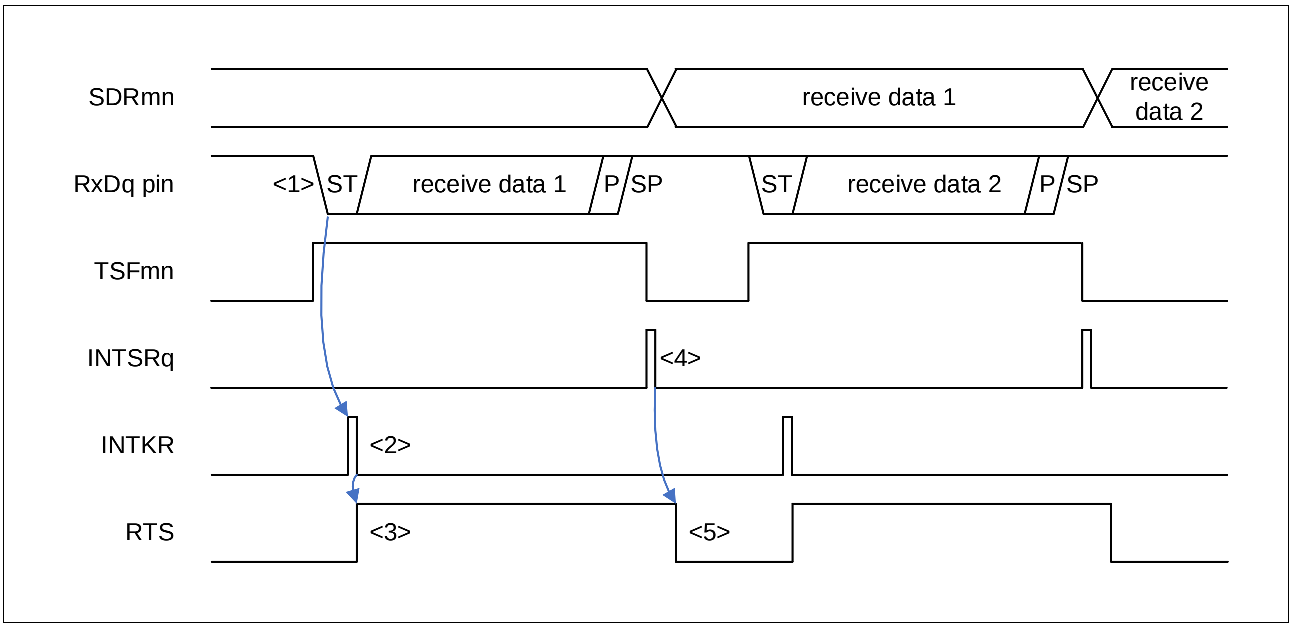
Item	Pin	Use
TxD	P77/TS09/KR7/INTP11 /(**TxD2**)	Used for UART data transmission.
RxD	P76/TS08/**KR6**/INTP10 /(**RxD2**)	Used for UART data reception. It detects the falling edge of the start bit during data reception using a key interrupt.
RTS	**P75**/TS07/KR5/INTP9 /SCK01/SCL01	Notifies the opposing device of the UART reception status. Low level: Receiver available High level: Receiver unavailable
CTS	**P31**/TS01/EI31/TI03/TO03 /**INTP4**/(PCLBUZ0)	Receives the UART reception status from the opposing device. It detects when CTS changes to a low level using an external interrupt.

- UART Reception

The timing chart for UART reception is shown in Figure 1-2.

In UART reception, the key interrupt function assigned to the RxDq pin is used to detect the start bit of the received data (<1>, <2>). Once the start bit is detected, RTS is set to a high level (<3>).
When data reception is complete, the INTSRq interrupt is triggered (<4>). During this interrupt handling, the received data is read from the SDRmn register. After that, RTS is set to a low level (<5>).

Figure 1-2 Timing Chart for Data Reception



- UART Transmission

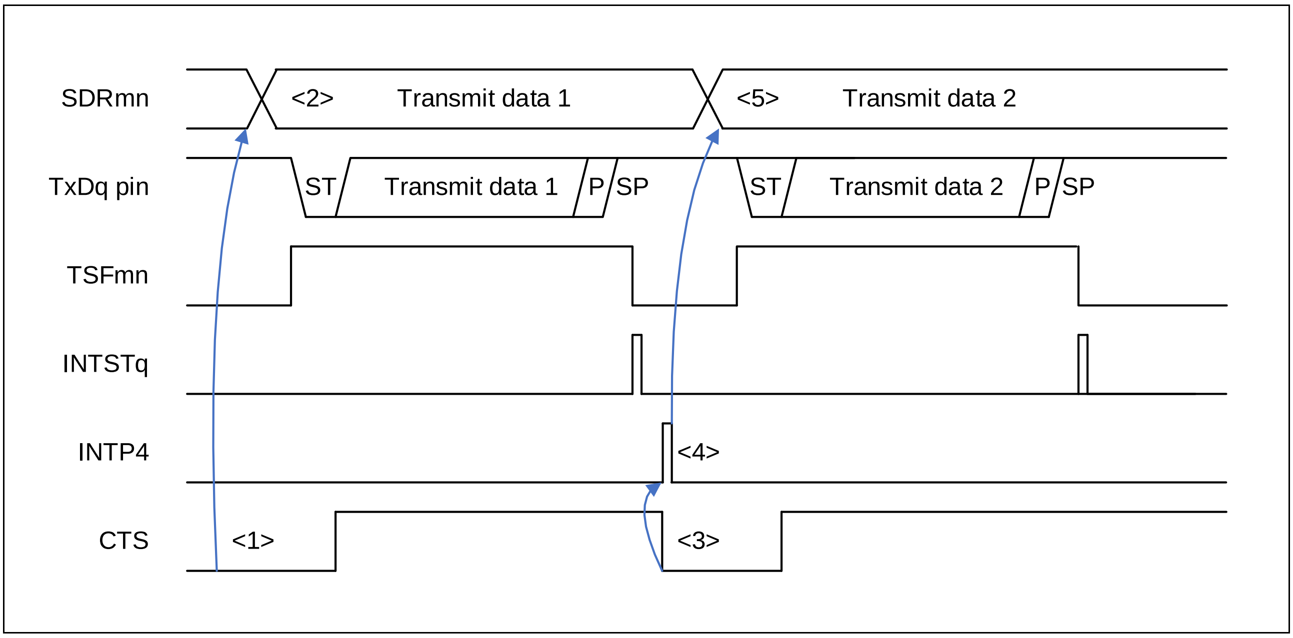
The timing chart for UART transmission is shown in Figure 1-3.

In UART transmission, the level of CTS is first checked (<1>).

If CTS is at a low level, the data to be transmitted is stored in the SDRmn register, and transmission begins (<2>).

If CTS is at a high level, the external interrupt assigned to CTS (INTP4) is enabled. When CTS goes low, the INTP4 interrupt is triggered (<3>, <4>). After the interrupt occurs, the transmission data is stored in the SDRmn register, and transmission begins (<5>).

Figure 1-3 Timing Chart for UART Transmission



1.2 Communication Specifications

This application note performs UART communication according to the data format shown in Figure 1-4. Additionally, the settings for UART communication are provided in the table below.

Table 1-2 UART Communication Settings

Item	Setting
Data bit length [bit]	8
Data transfer order	LSB first
Parity	Even parity
Transfer rate [bps]	1000000

Figure 1-4 Data Format

STX	LEN	Data (variable length)	SUM	ETX
(1 byte)	(1 byte)	(Up to 255 bytes)	(1 byte)	(1 byte)

Symbol	Value	Description
STX	02H	Frame header
LEN	-	Data length information
SUM	-	Checksum data within the frame. Calculated by subtracting each byte of the data (starting from the initial value of 00H) from the total data to be included (ignoring borrow).
ETX	03H	Frame footer

An example of calculating the checksum (SUM) within the frame is shown below.

For the following frame, the data to be included in the checksum calculation is from LEN to D4.

The value of SUM is calculated as 00H (initial value) – 04H – FFH – 80H – 40H – 22H = 1BH (ignoring borrow, using only the lower 8 bits).

STX	LEN	D1	D2	D3	D4	SUM	ETX
02H	04H	FFH	80H	40H	22H	Checksum	03H

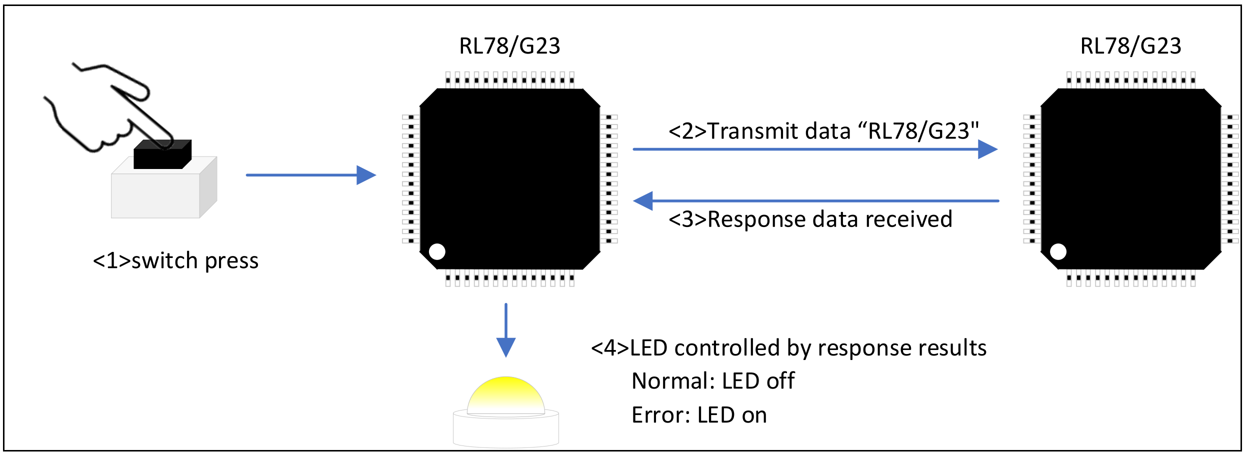
1.3 Detailed Specifications

The state transition diagram of this sample code is shown in Figure 1-5.

In this sample code, after the initial setup is completed, the system waits for the pressing of a switch. Once the switch is pressed, data transmission via UART begins. After all the data is transmitted, the microcontroller waits for a response from the counterpart device. If the response data indicates successful completion, it waits for the switch to be pressed again. If the response data indicates an abnormal termination or a checksum error, the LED will light up.

Additionally, the counterpart device can use this sample code. Upon receiving data, it will transmit a response based on the result of the reception.

Figure 1-5 State Transition Diagram



Initial Port Configuration

＜Port Configuration Conditions＞

- Set P52, which controls LED2, to high-level output.
- Set P53, which controls LED1, to high-level output.
- Set P75, which controls RTS, to high-level output.
- Set P31, which controls CTS, to input.

(2) Initial UART2 Configuration

＜UART2 Configuration Conditions＞

- Use SAU1 channels 0 and 1 as UART2.
- Use P76/RxD2 for data input and P77/TxD2 for data output.
- Set the transfer mode to single transfer mode.
- Set the data length to 8 bits.
- Set the parity to even parity.
- Set the data transfer order to LSB first.
- Set the transfer rate to 1 Mbps.

(3) Initial Configuration of External Interrupts

＜External Interrupt Configuration Conditions＞

- Use INTP0 for falling edge detection.
- Use INTP4 for falling edge detection.

(4) Initial Configuration of Key Interrupts

＜Key Interrupt Configuration Conditions＞

- Use KR6 for falling edge detection.

(5) Once the initial setup is complete, communication with the counterpart device is performed following the steps below:

<1> Set RTS to low level.

<2> Wait for the switch to be pressed or for data reception via UART in HALT mode.

Transmission Process（<3>~<9>）

<3> When the switch is pressed, an INTP0 interrupt occurs, and the INTP0 interrupt handler is executed.

In the INTP0 interrupt handler, the INTP0 interrupt is disabled to prevent the switch from being activated during data transmission. The data to be sent is stored in the transmission buffer according to the formatted structure.

<4> Check CTS.

If CTS is at a low level, proceed to transmit the data (move to <6>).

If CTS is at a high level, suspend data transmission and enable the INTP4 interrupt to detect when CTS changes to a low level.

<5> Detect changes in CTS.

When CTS changes to a low level, an INTP4 interrupt occurs. Within the INTP4 interrupt handler, disable the INTP4 interrupt and then transmit the data.

<6> Wait for the completion of data transmission in HALT mode.

<7> Repeat steps <4> to <6> until all data has been transmitted.

<8> Once data transmission is complete, enable the INTP0 interrupt to re-enable the switch.

<9> Wait for response data from the counterpart device in HALT mode.

Reception Process（<10>~<14>）

<10> When data reception from the counterpart device begins, a key interrupt is triggered by the falling edge of the start bit.

<11> In the key interrupt handler, set RTS to a high level. Next, disable the key interrupt until the reception of one byte is complete.

<12> When a reception completion interrupt occurs, read the data and execute processing based on the following statuses.

- Waiting for receiving：

  Verify whether the received data matches STX. If it matches, set the reception status to " STX received." If it does not match, set the reception status to "Error occurred."

- STX received：

  Store the received data as the value for data length in a variable. Begin the checksum calculation.

- Receiving data：

Store the received data in the reception buffer. Next, calculate the checksum. If the number of received data matches the data length, set the reception status to " Data reception completed."

- Data reception completed：

  Verify whether the received data matches the calculated checksum value. If it matches, set the reception status to " Checksum received." If it does not match, set the status to "Error occurred."

- Checksum received：

  Verify whether the received data matches ETX. If it matches, set the reception status to " ETX received " and set ACK as the response data. If it does not match, set the reception status to "Error occurred" and set NACK as the response data.

<13> Repeat steps <9> to <12> until the reception status indicates an error or all data has been received.

<14> Execute processing based on the occurrence of an error or the response data.

- If the reception status indicates an error, turn on LED1 and terminate the process.
- If the response data indicates normal completion, turn on LED2 and terminate the process.
- If the response data indicates an abnormal termination, perform initialization and proceed to step <2>.

Additionally, if data reception from the counterpart device is started at step <2>, execute the reception process from step <10> to <13>. Then, store the response data to be transmitted in the transmission buffer according to the formatted structure. After that, execute the transmission process from step <4> to <7> to send the response data.

2. Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed in the table below.

Table 2-1 Operation Confirmation Conditions

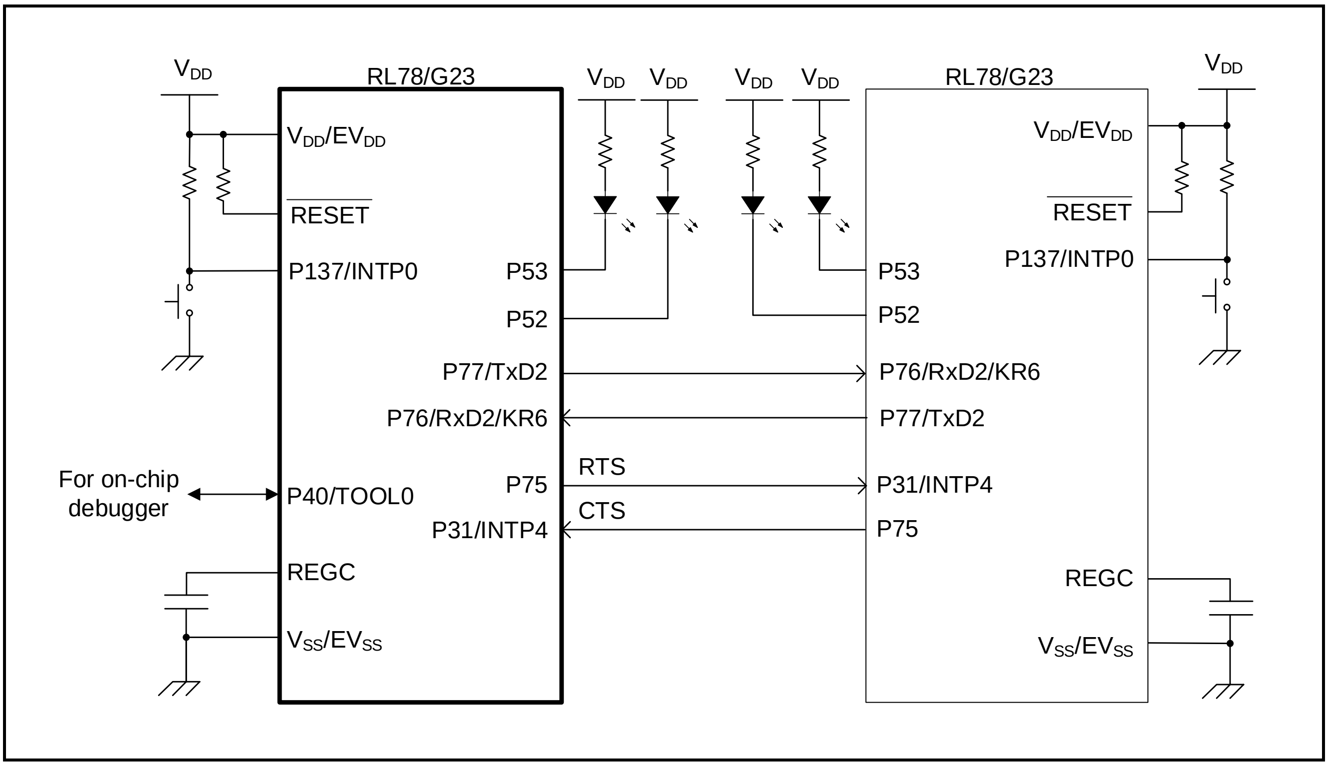
Item	Description
Microcontroller used	RL78/G23 (R7F100GLG)
Operating frequency	● High-speed on-chip oscillator (HOCO) clock: 32 MHz ● CPU/peripheral hardware clock: 32 MHz
Operating voltage	5.0 V (can be operated at 2.4 V Note to 5.5 V) LVD0 detection voltage: Reset mode At rising edge TYP. 1.90 V (1.84 V to 1.95 V) At falling edge TYP. 1.86 V (1.80 V to 1.91 V)
Integrated development environment (CS+)	CS+ for CC V8.11.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.13.00 from Renesas Electronics Corp.
Integrated development environment (e2 studio)	e^2 studio 2024-07 (24.7.0) from Renesas Electronics Corp.
C compiler (e2 studio)	CC-RL V1.13.00 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 V5.10.3 from IAR Systems
C compiler (IAR)	IAR C/C++ Compiler for Renesas RL78 V5.10.3.2716 from IAR Systems
Smart configurator (SC)	V.1.10.0
Board support package (BSP)	V.1.62
Board	RL78/G23-64p Fast Prototyping Board (RTK7RLG230CLG000BJ)

3. Description of the Hardware

3.1 Hardware Configuration Example

The example of configuration of the hardware that is used for this application note is shown below.

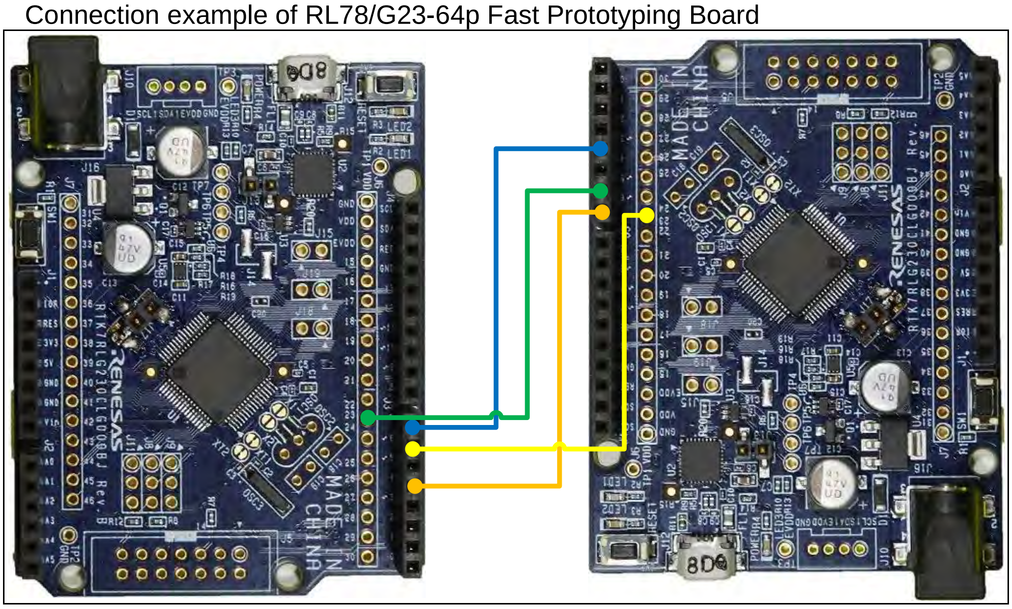
Figure 3-1 Hardware Configuration



Caution 1. This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements. (Connect each input-only port to VDD or VSS through a resistor.)

Caution 2. VDD must be held at not lower than the reset release voltage (VLVD0) that is specified as LVD.

Figure 3-2 Connection example of RL78/G23-64p Fast Prototyping Board



3.2 Used Pins

Table 3-1 shows list of used pins and assigned functions.

Table 3-1 List of Pins and Functions

Pin name	Input/Output	Description
P31/TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	Input	CTS signal input
P52/(INTP10)	Output	LED2 control signal output
P53/(INTP11)	Output	LED1 control signal output
P75/TS07/KR5/INTP9/SCK01/SCL01	Output	RTS signal output
P76/TS08/KR6/INTP10/(RxD2)	Input	Data reception
P77/TS09/KR7/INTP11/(TxD2)	Output	Data transmit
P137/EI137/INTP0	Input	Switch signal input

Caution. In this application note, only the used pins are processed. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

4. Software

4.1 Option Byte Settings

Table 4-1 shows the option byte settings.

Table 4-1 Option Byte Settings

Address	Setting Value	Description
000C0H	1110 1111B (EFH)	Operation of Watchdog timer is stopped (counting is stopped after reset)
000C1H	1111 1110B (FEH)	LVD0 operating mode: reset mode Detection voltage: Rising edge 1.90V Falling edge 1.86V
000C2H	11101000B (E8H)	Flash operating mode: HS mode High-speed on-chip oscillator clock: 32MHz
000C3H	10000100B (84H)	On-chip debugging is enabled

4.2 Constants

Table 4-2 shows the constants.

Table 4-2 Constants

Constant Name	Definition	Setting Value	Description
WAIT	r_cg_userdefine.h	3	Set value of waiting time from start bit detection to TSF flag judgment when receiving
CTS_HIGH	r_cg_userdefine.h	1	CTS signal level: High level
CTS_LOW	r_cg_userdefine.h	0	CTS signal level: Low level
RTS_HIGH	r_cg_userdefine.h	1	RTS signal level: High
RTS_LOW	r_cg_userdefine.h	0	RTS signal level: Low level
LED_ON	r_cg_userdefine.h	0	LED lighting value
SEND_START	r_cg_userdefine.h	1	Transmission status: Transmission started
SENDING	r_cg_userdefine.h	2	Transmission status: Transmitting
SEND_END	r_cg_userdefine.h	3	Transmission status: Transmission completed
RECEIVING_WAIT	r_cg_userdefine.h	0	Receive status: Waiting for receiving
STX_RECEIVED	r_cg_userdefine.h	1	Receive status: STX received
RECEIVING_DATA	r_cg_userdefine.h	2	Receive status: Receiving data
DATA_RECEIVED	r_cg_userdefine.h	3	Receive status: Data reception completed
SUM_RECEIVED	r_cg_userdefine.h	4	Receive status: Checksum received
ETX_RECEIVED	r_cg_userdefine.h	5	Receive status: ETX received
RECEIVING_ERROR	r_cg_userdefine.h	6	Receive status: Error occurred
ACK	r_cg_userdefine.h	0x06	Response after receiving: Normal termination
SUM_ERROR	r_cg_userdefine.h	0x07	Response after receiving: Checksum error occurred
NACK	r_cg_userdefine.h	0x15	Response after receiving: Abnormal termination

4.3 Variables

Table 4-3 shows the variables.

Table 4-3 Variables

Type	Variable Name	Contents	Function Used
uint8_t	g_send_data[]	Transmit Data	main.c, Config_INTC_user.c, Config_UART2_user.c
uint8_t	g_receive_buffer[]	Receive Data Buffer	main.c, Config_UART2_user.c
uint8_t	g_send_status	Transmission Status	main.c, Config_INTC_user.c, Config_UART2_user.c
uint8_t	g_receive_status	Reception Status	main.c, Config_UART2_user.c
uint8_t	g_receive_length	Data Length of Received Data	main.c, Config_UART2_user.c
uint8_t	g_receive_checksum	Checksum Value of Received Data	Config_UART2_user.c
uint8_t	g_receive_count	Number of Received Data	Config_UART2_user.c
uint8_t	g_receive_response	Response Data After Reception	main.c, Config_UART2_user.c
uint8_t	g_cmd_send_buffer[]	Transmit Data Buffer	Config_UART2_user.c

4.4 Functions

Table 4-4 shows the functions used in the sample code. However, the unchanged functions generated by the Smart Configurator are excluded.

Table 4-4 Functions

Function name	Outline	Source file
main	Main Processing	main.c
r_main_user_init	Initialization Function	main.c
r_Config_INTC_intp0_interrupt	INTP0 Interrupt Function	Config_INTC_user.c
r_Config_INTC_intp4_interrupt	INTP4 Interrupt Function	Config_INTC_user.c
r_Config_KR_interrupt	Key Interrupt Function	Config_KR_user.c
r_Config_UART2_interrupt_send	UART2 Transmission Completion Interrupt Function	Config_UART2_user.c
r_Config_UART2_interrupt_receive	UART2 Reception Completion Interrupt Function	Config_UART2_user.c
r_send_protocol	Transmit Data Storage Function	Config_UART2_user.c
r_uart_send_control	Transmission Control Function	Config_UART2_user.c
r_uart_receive_control	Reception Control Function	Config_UART2_user.c

4.5 Function Specifications

This part describes function specifications of the sample code.

[Function name] main

Outline	Main Processing
Header	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, Config_KR.h, r_cg_userdefine.h
Declaration	void main(void);
Description	After the initial setup, the operating mode transitions to HALT mode. HALT mode is exited by various interrupts. Based on the state of data reception, the following actions are performed: ・If a reception error occurs, LED1 is turned on. ・If the received response data indicates abnormal termination, LED2 is turned on. ・If the received data is not response data (data length is 2 or more), the microcontroller begins transmitting response data to the counterpart device.
Arguments	• None
Return value	• None
Remarks	None

[Function name] r_main_user_init

Outline	Initialization Function
Header	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, Config_KR.h, r_cg_userdefine.h
Declaration	static void r_main_user_init(void);
Description	Enable the switch input (INTP0). Allow UART reception to operate.
Arguments	• None
Return value	• None
Remarks	None

[Function name] r_Config_INTC_intp0_interrupt

Outline	INTP0 Interrupt Function
Header	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, r_cg_userdefine.h
Declaration	#pragma interrupt r_Config_INTC_intp0_interrupt(vect=INTP0)
Description	Wait to eliminate chattering. After that, data transmission will begin.
Arguments	• None
Return value	• None
Remarks	None

[Function name] r_Config_INTC_intp4_interrupt

Outline	INTP4 Interrupt Function
Header	r_cg_macrodriver.h, Config_INTC.h, Config_UART2.h, r_cg_userdefine.h
Declaration	#pragma interrupt r_Config_INTC_intp4_interrupt(vect=INTP4)
Description	Data transmission will be performed.
Arguments	• None
Return value	• None
Remarks	None

[Function name] r_Config_KR_interrupt

Outline	Key Interrupt Function
Header	r_cg_macrodriver.h, Config_KR.h, r_cg_userdefine.h
Declaration	#pragma interrupt r_Config_KR_interrupt(vect=INTKR)
Description	The start bit detection during UART reception will be performed. After waiting, the UART communication status will be checked. If communication is ongoing, the RTS signal will be set to high level.
Arguments	• None
Return value	• None
Remarks	None

[Function name] r_Config_UART2_interrupt_send

Outline	UART2 Transmission Completion Interrupt Function
Header	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
Declaration	#pragma interrupt r_Config_UART2_interrupt_send(vect=INTST2)
Description	The transmission control function will be executed.
Arguments	• None
Return value	• None
Remarks	The code generated by the Smart Configurator is being modified.

[Function name] r_Config_UART2_interrupt_receive

Outline	UART2 Reception Completion Interrupt Function
Header	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
Declaration	#pragma interrupt r_Config_UART2_interrupt_receive(vect=INTSR2)
Description	The reception control function will be executed.
Arguments	• None
Return value	• None
Remarks	The code generated by the Smart Configurator is being modified.

[Function name] r_send_protocol

Outline	Transmit Data Storage Function
Header	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
Declaration	MD_STATUS r_send_protocol(const uint8_t __far * data, const uint16_t data_len);
Description	The STX, data count, transmission data, checksum, and ETX are stored in the transmit buffer according to the transmission format.
Arguments	const uint8_t __far * data: Starting address of the data const uint16_t data_len: Length of the data
Return value	81H：Abnormal termination 00H：Normal termination
Remarks	None

[Function name] r_uart_send_control

Outline	Transmission Control Function
Header	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
Declaration	void r_uart_send_control(void);
Description	Check the level of CTS. If CTS is at a high level, enable the external interrupt (INTP4) to detect changes in CTS. If CTS is at a low level, perform data transmission.
Arguments	• None
Return value	• None
Remarks	None

[Function name] r_uart_receive_control

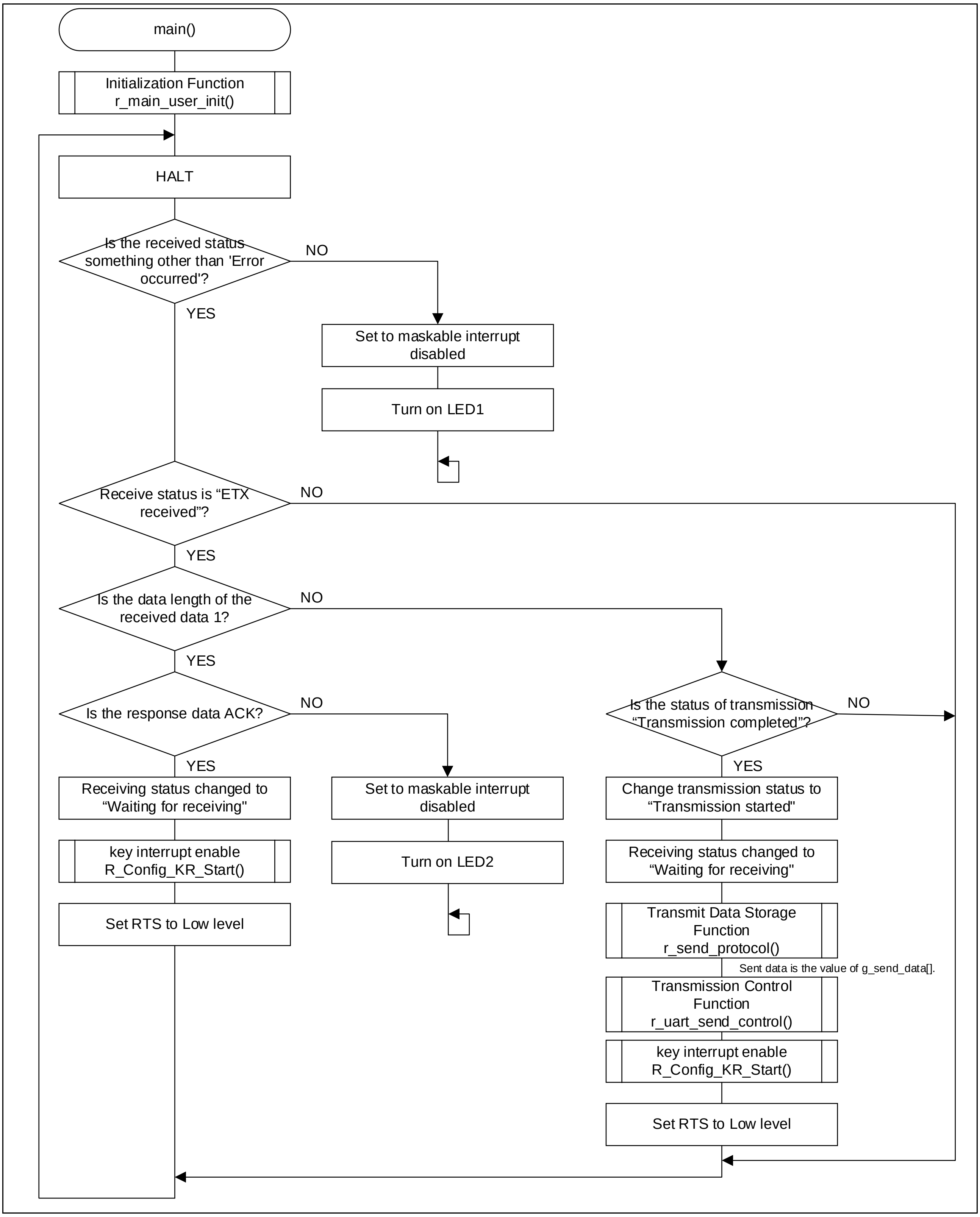
Outline	Reception Control Function
Header	r_cg_macrodriver.h, Config_UART2.h, Config_KR.h, Config_INTC.h,r_cg_userdefine.h
Declaration	void r_uart_receive_control (void);
Description	Read the received data. After that, execute the process according to the reception status as follows. ・RECEIVING_WAIT Determine if the received data matches the STX. ・STX_RECEIVED Start calculating the checksum. ・RECEIVING_DATA Store the received data and calculate the checksum. ・DATA_RECEIVED Verify if the received data matches the calculated checksum value. ・SUM_RECEIVED Determine if the received data matches the ETX.
Arguments	• None
Return value	• None
Remarks	None

4.6 Flow Charts

4.6.1 Main Processing

Figure 4-1 shows the flowchart of main processing.

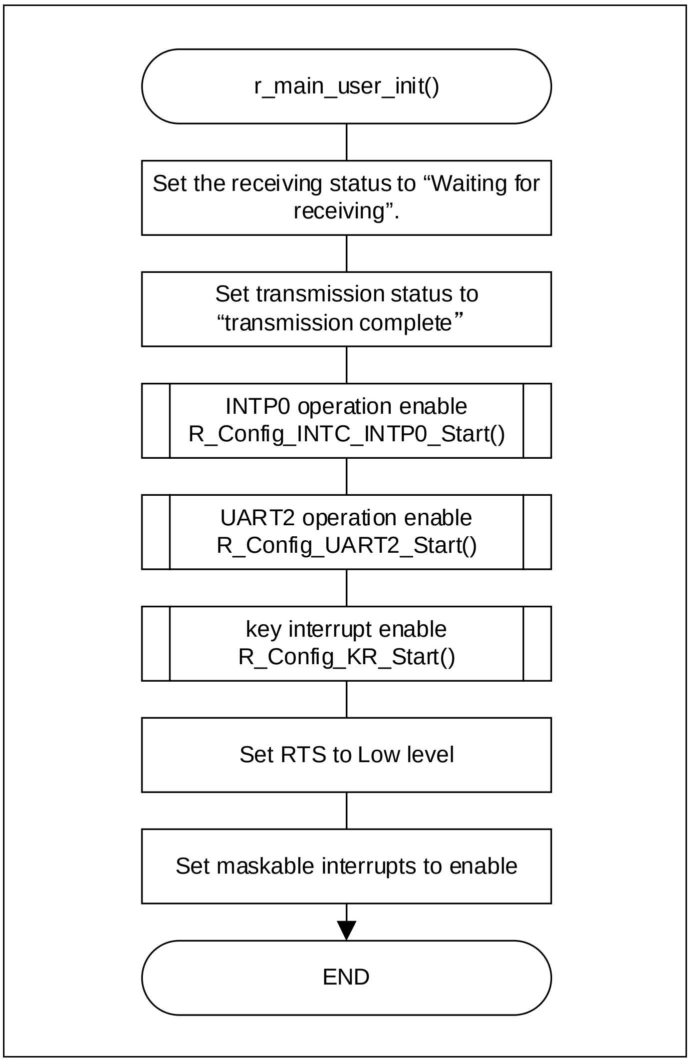
Figure 4-1 Main Processing



4.6.2 Initialization Function

Figure 4-2 shows the flowchart of initialization function.

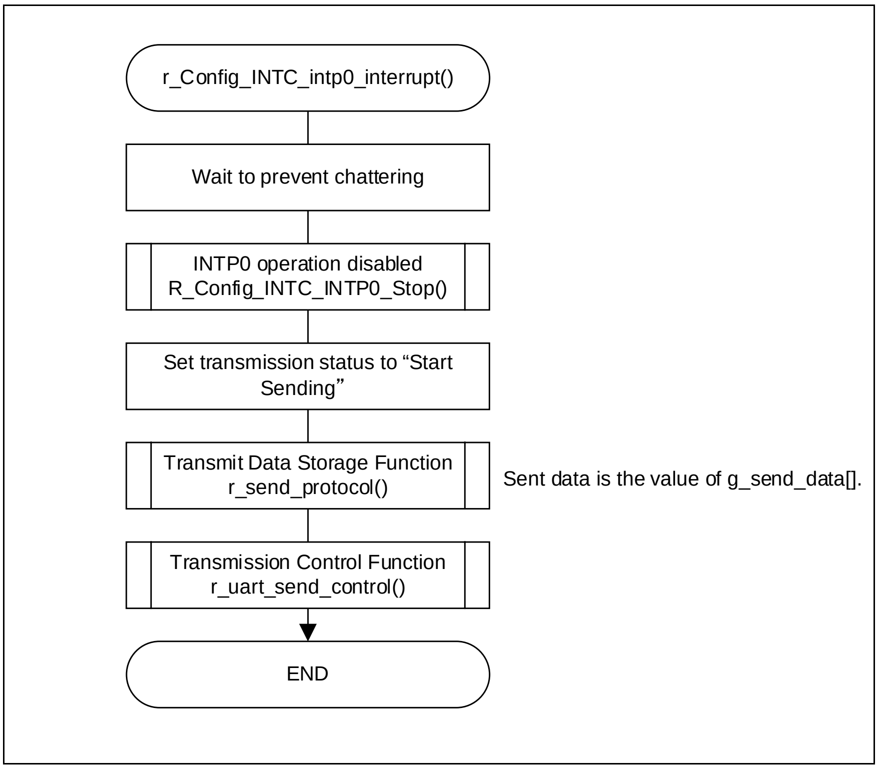
Figure 4-2 Initialization Function



4.6.3 INTP0 Interrupt Function

Figure 4-3 shows the flowchart of INTP0 interrupt function.

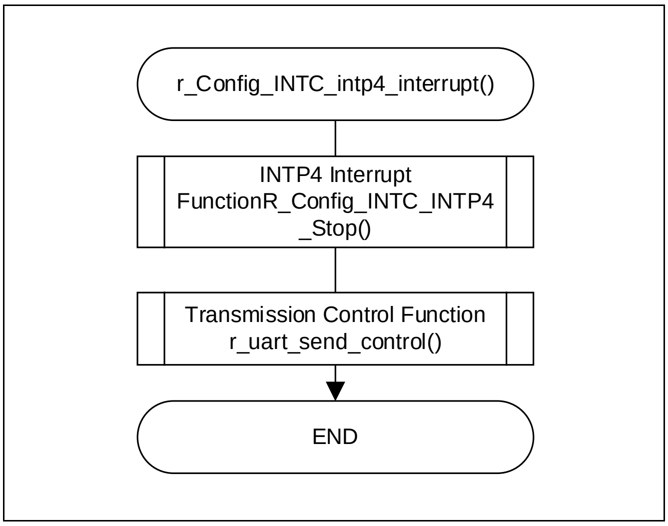
Figure 4-3 INTP0 Interrupt Function



4.6.4 INTP4 Interrupt Function

Figure 4-4 shows the flowchart of INTP4 interrupt function.

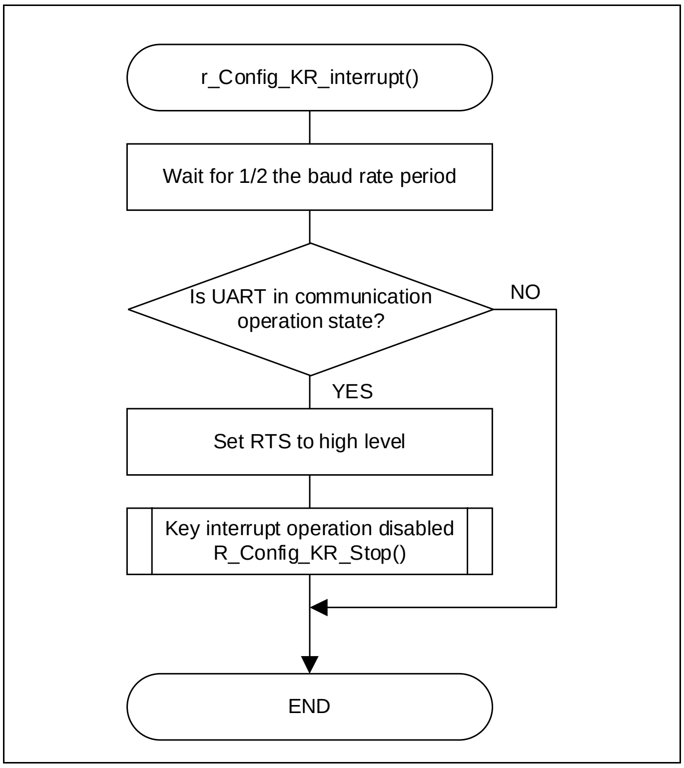
Figure 4-4 INTP4 Interrupt Function



4.6.5 Key Interrupt Function

Figure 4-5 shows the flowchart of key interrupt function.

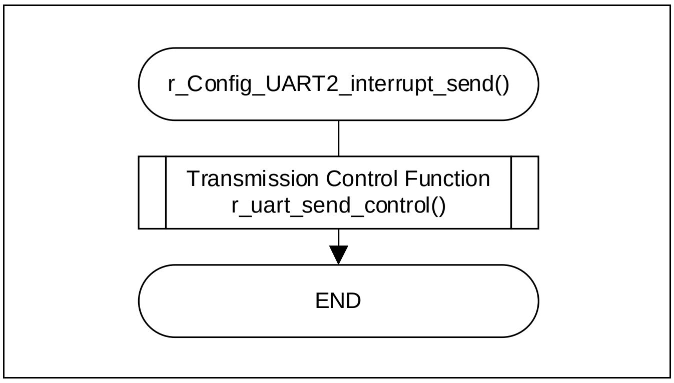
Figure 4-5 Key Interrupt Function



4.6.6 UART2 Transmission Completion Interrupt Function

Figure 4-6 shows the flowchart of UART2 transmission completion interrupt function.

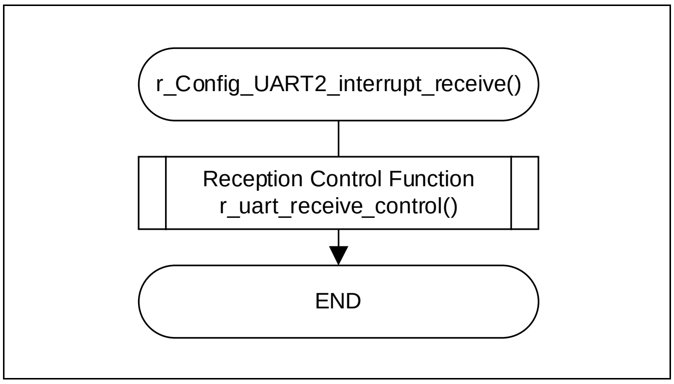
Figure 4-6 UART2 Transmission Completion Interrupt Function



4.6.7 UART2 Reception Completion Interrupt Function

Figure 4-7 shows the flowchart of UART2 reception completion interrupt function.

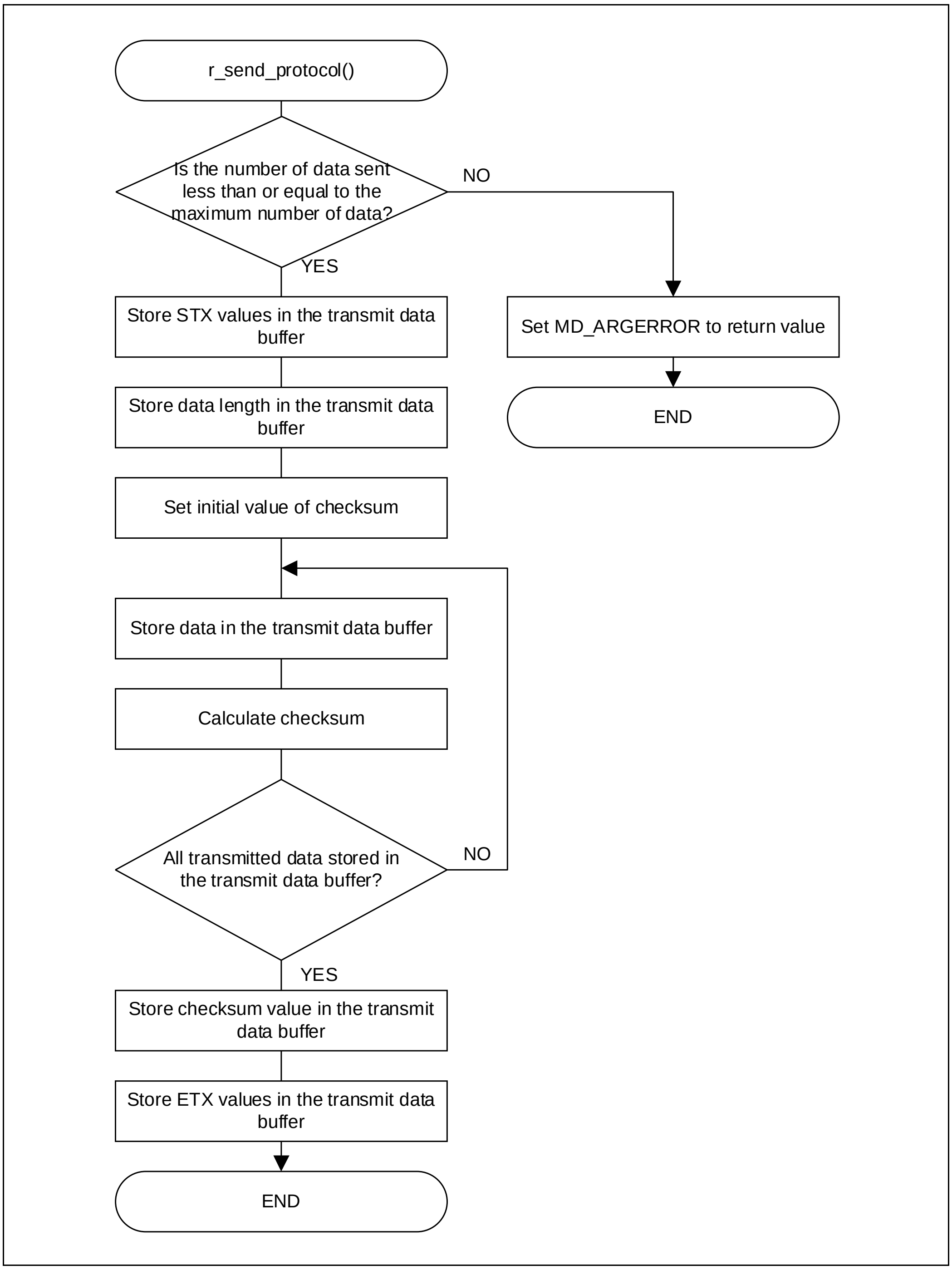
Figure 4-7 UART2 Reception Completion Interrupt Function



4.6.8 Transmit Data Storage Function

Figure 4-8 shows the flowchart of transmit data storage function.

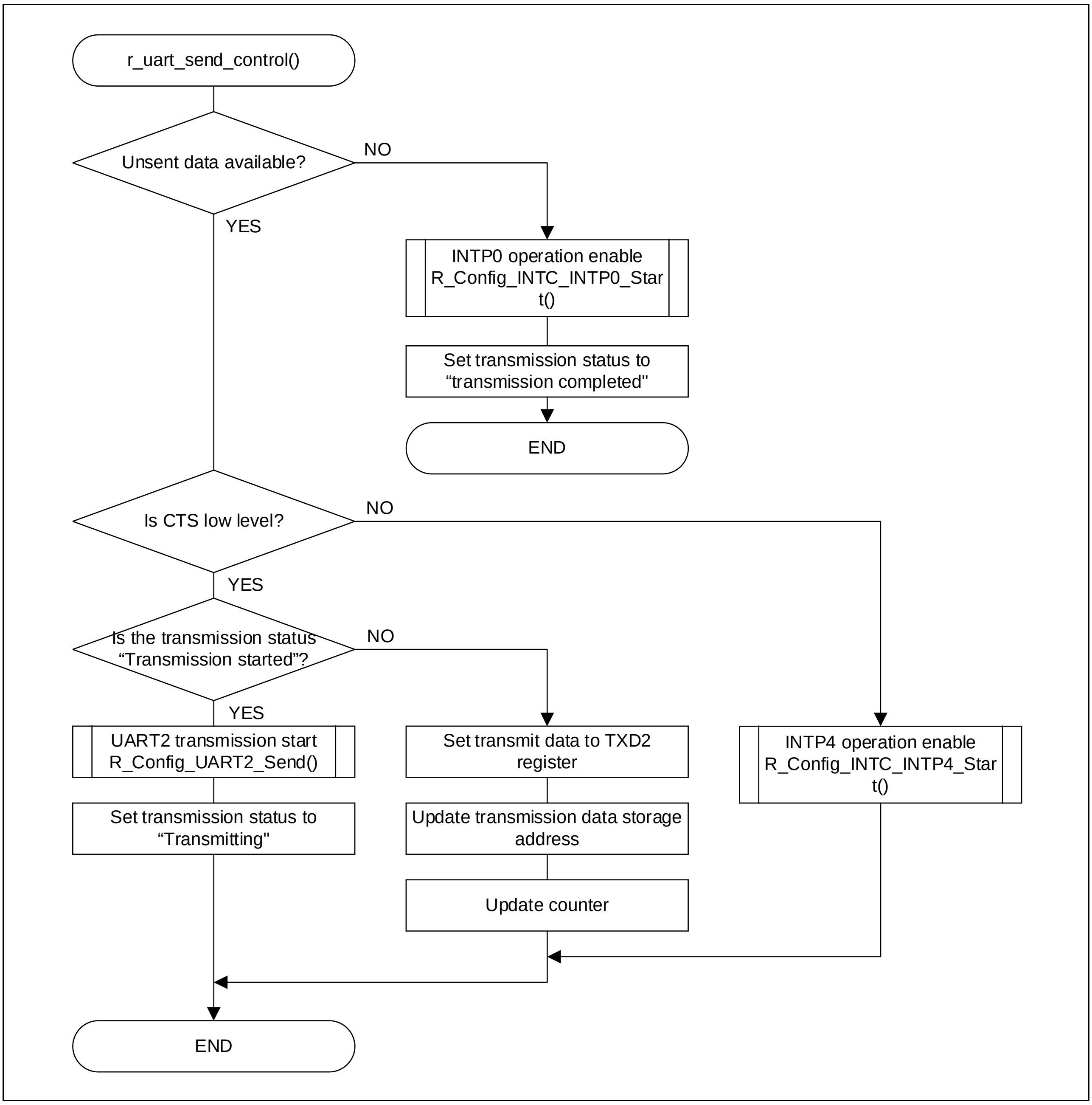
Figure 4-8 Transmit Data Storage Function



4.6.9 Transmission Control Function

Figure 4-9 shows the flowchart of transmission control function.

Figure 4-9 Transmission Control Function



4.6.10 Reception Control Function

Figure 4-10 to Figure 4-12 shows the flowchart of reception control function.

Figure 4-10 Reception Control Function (1/3)

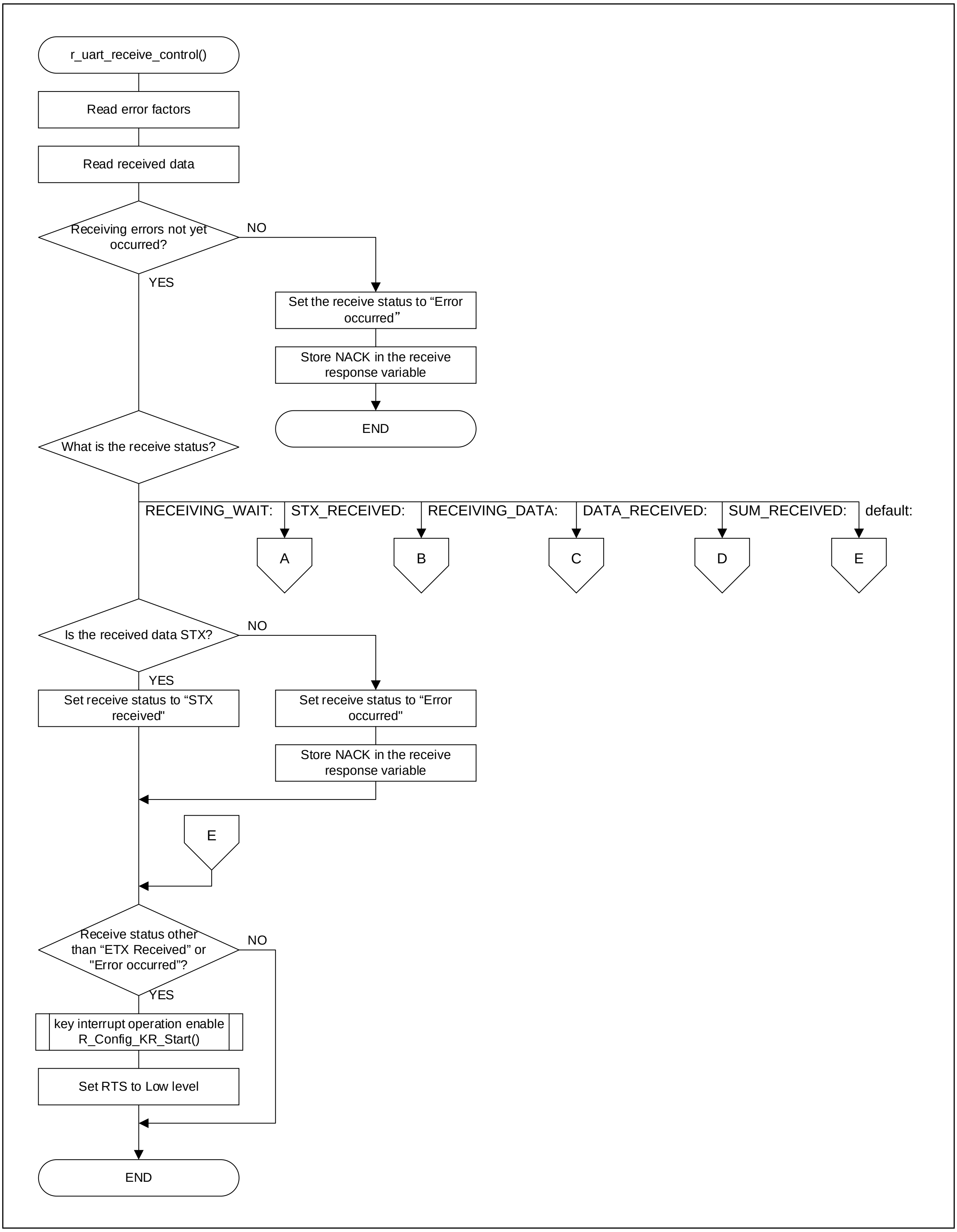


Figure 4-11 Reception Control Function (2/3)

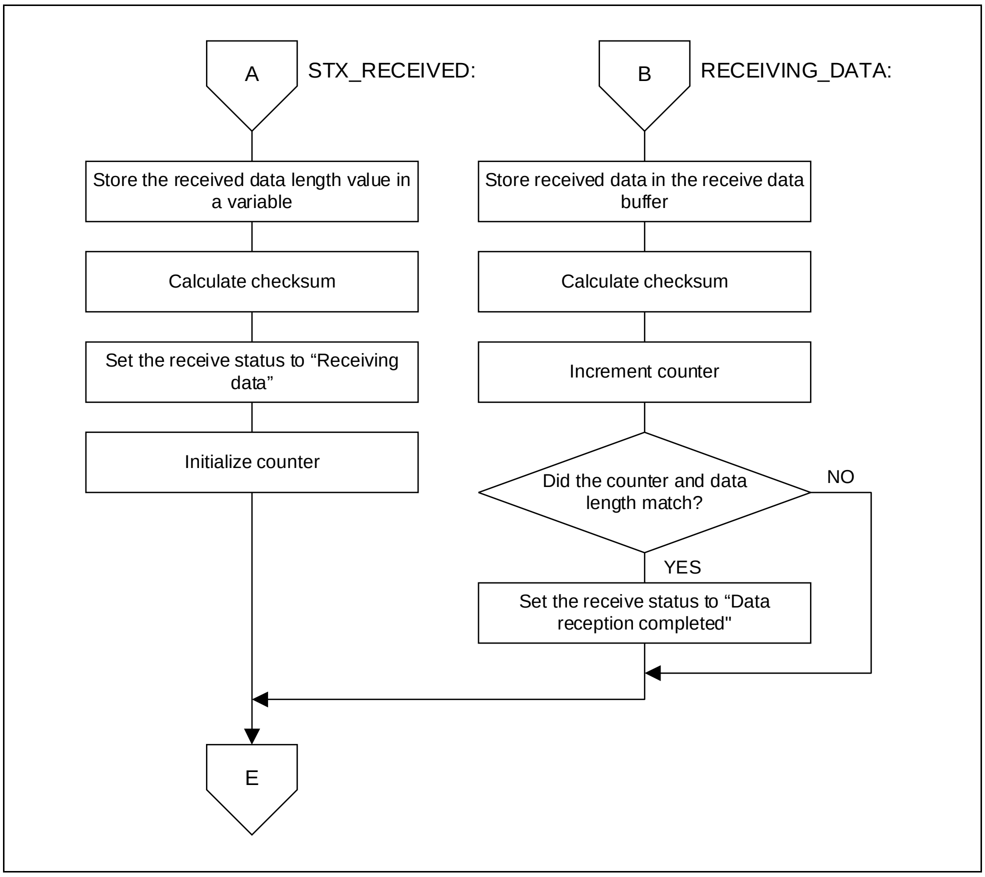
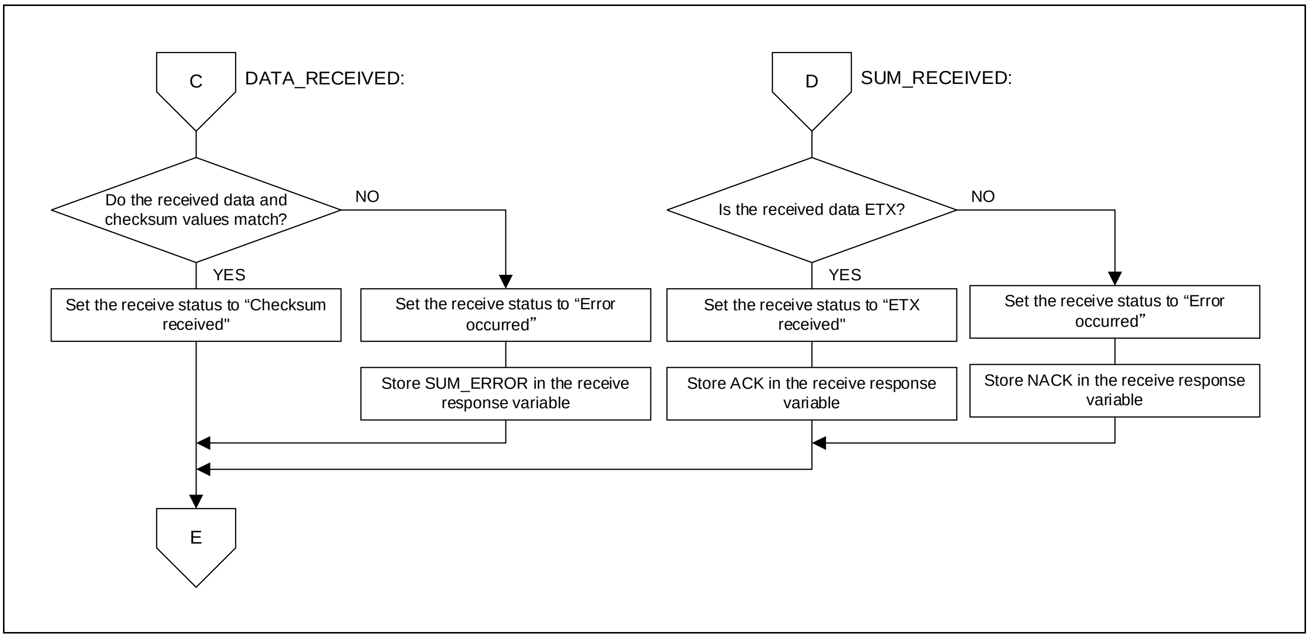


Figure 4-12 Reception Control Function (3/3)



5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference

RL78/G23 User's Manual: Hardware (R01UH0896E)
RL78 Family User's Manual: Software (R01US0015E)
(The latest version can be downloaded from the Renesas Electronics website.)

Technical Update / Technical News
(The latest version can be downloaded from the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2024.12.6	-	First edition

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/.