RL78/G15

Serial Array Unit (UART Communication)

Introduction

This application note explains how to use UART communication through the serial array unit (SAU). ASCII characters transmitted from the device on the opposite side are analyzed to make responses.

Target Device

RL78/G15

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. Specification

1.1 Overview of Specification

In this application note, UART communication is performed through the serial array unit (SAU). ASCII characters transmitted from the device on the opposite side are analyzed to make responses.

Table 1-1 shows the peripheral function to be used and its use. Figures 1-1 and 1-2 illustrate UART communication operation.

Table 1-1 Peripheral Function and Use

Peripheral Function	Use
Serial array unit UART0	Perform UART communication using the TxD0 pin (transmission) and the RxD0 pin (reception).

Figure 1-1 UART Reception Timing Chart

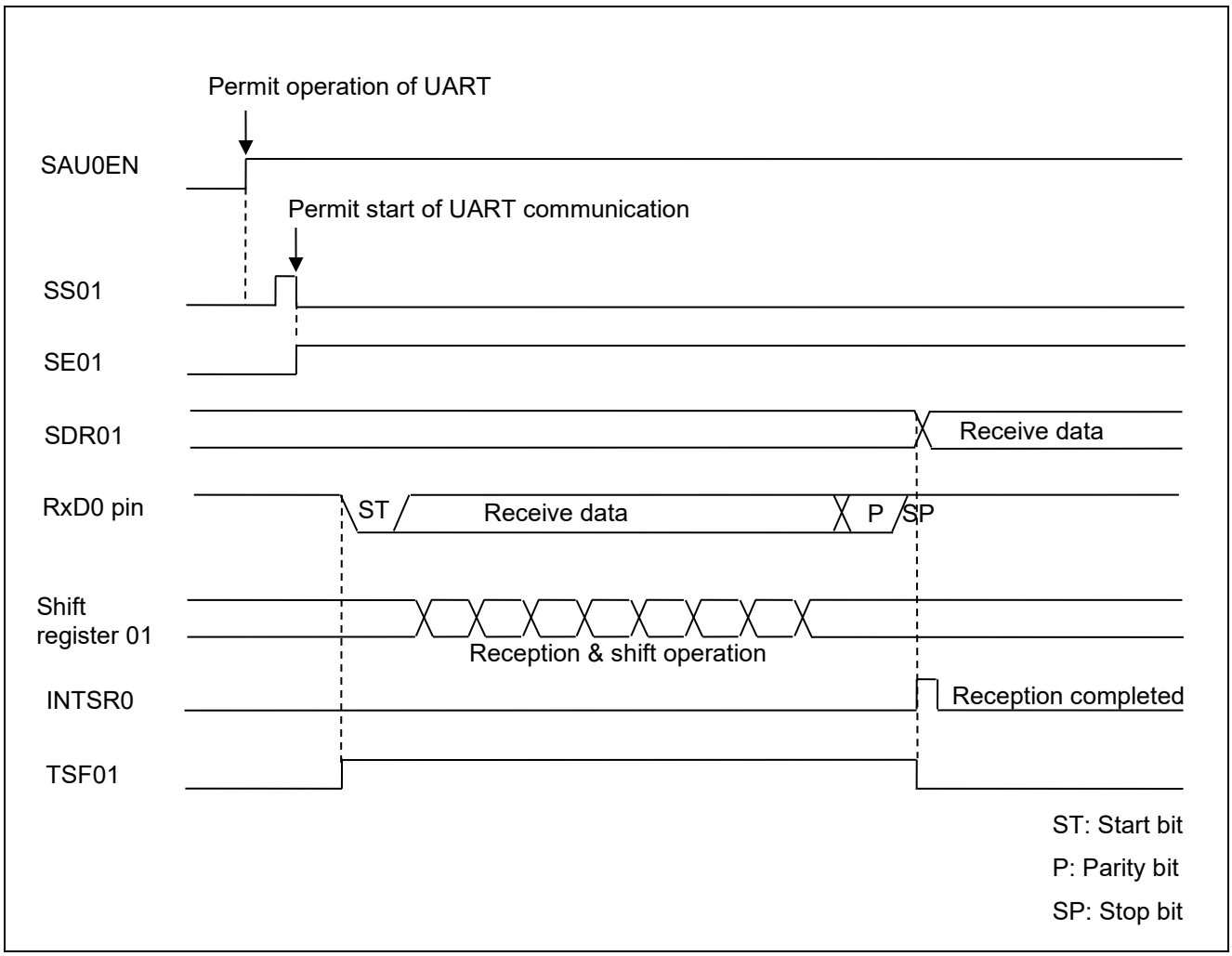
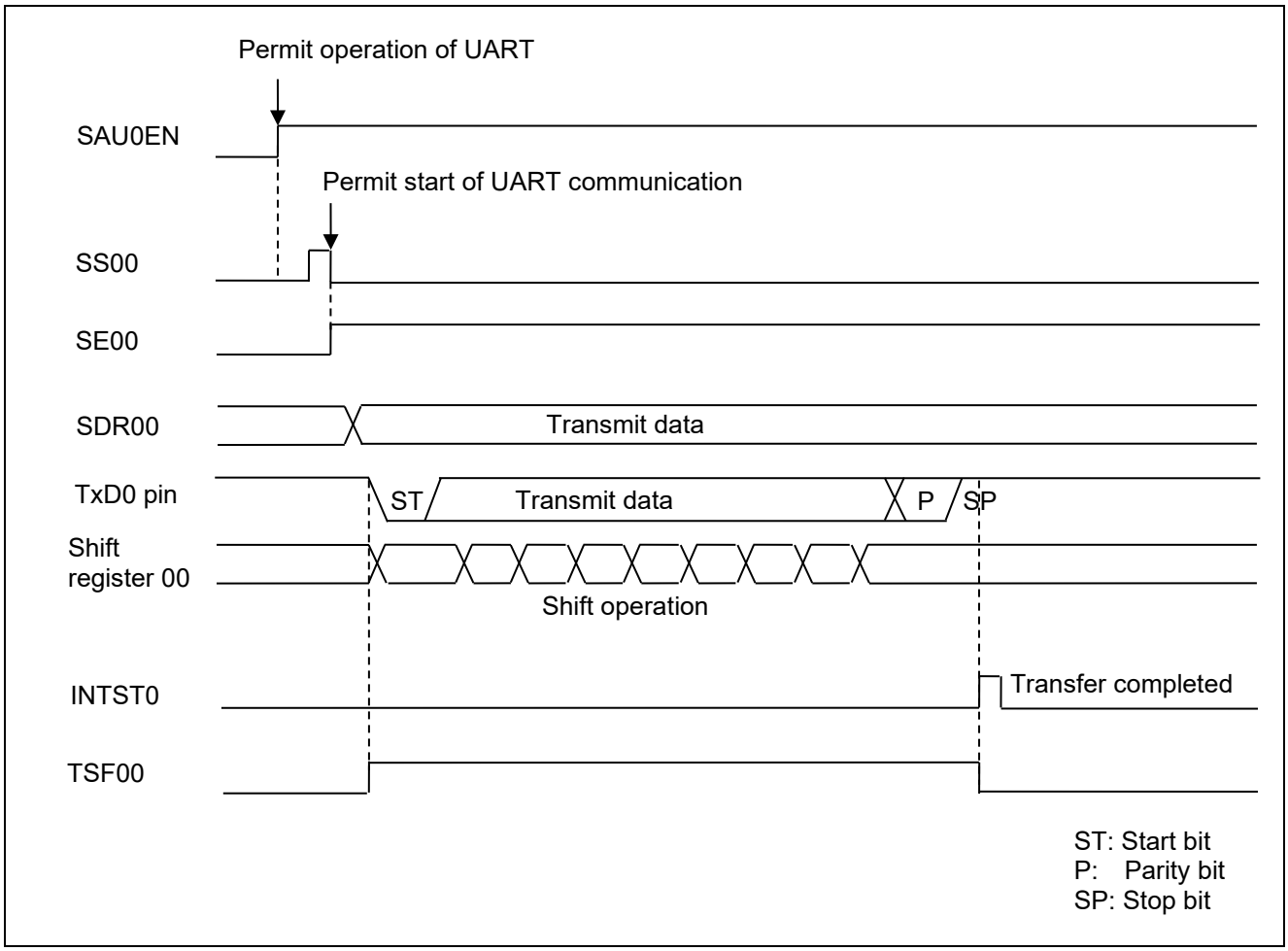


Figure 1-2 UART Transmission Timing Chart



1.2 Outline of Operation

This sample code transmits, to the device on the opposite side, the data corresponding to that received from the device. If an error occurs, it transmits to the device the data corresponding to the error. Tables 1-2 and 1-3 show the correspondence between transmit data and receive data.

Table 1-2 Correspondence between Receive Data and Transmit Data

Receive Data	Response (Transmit) Data
T (54H)	O (4FH), K (4BH), ”CR” (0DH), ”LF” (0AH)
t (74H)	o (6FH), k (6BH), ”CR” (0DH), ”LF” (0AH)
Other than above	U (55H), C (43H), ”CR” (0DH), ”LF” (0AH)

Table 1-3 Correspondence between Error and Transmit Data

Error	Response (Transmit) Data
Parity error	P (50H), E (45H), ”CR” (0DH), ”LF” (0AH)
Framing error	F (46H), E (45H), ”CR” (0DH), ”LF” (0AH)
Overrun error	O (4FH), E (45H), ”CR” (0DH), ”LF” (0AH)

The following describes the major settings of the peripheral functions.

(1) Initial settings of the UART

- Use SAU0 channels 0 and 1 as UART.
- Use the P12/TxD0 pin and the P11/RxD0 pin for data output and data input, respectively.
- The data length is 8 bits.
- Set the data transfer direction to LSB first.
- Use even parity as the parity setting.
- Set the receive data level to standard.
- Set the transfer rate to 9600 bps.
- Use reception end interrupt (INTSR0), transmission end interrupt (INTST0), and error interrupt (INTSRE0).
- Set the interrupt priority orders of INTSR0, INTST0, and INTSRE0 to low priority.

2. Operation Confirmation Conditions

The operation of the sample code provided with this application note has been tested under the following conditions.

Table 2-1 Operation Confirmation Conditions

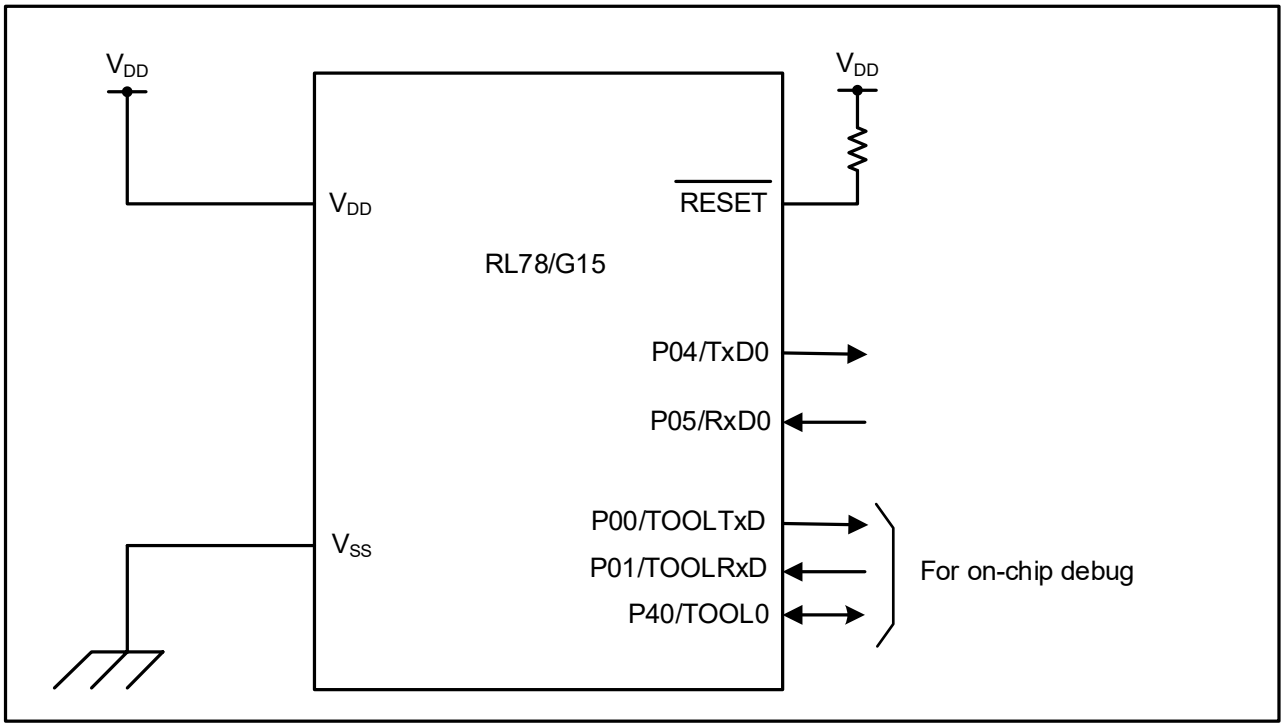
Item	Description
MCU used	RL78/G15 (R5F12068)
Board used	RL78/G15-20p Fast Prototyping Board (RTK5RLG150CLG000BJ)
Operating frequency	High-speed on-chip oscillator clock (f_{IH}): 16 MHz
Operating voltage	5.0 V (can be operated at 2.4 V to 5.5 V)) SPOR operations (V_{SPOR}) At rising edge TYP. 2.57 V (2.44 V to 2.68 V) At falling edge TYP. 2.52 V (2.40 V to 2.62 V)
Integrated development environment (CS+)	CS+ V8.09.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.12.00 from Renesas Electronics Corp.
Integrated development environment (e2studio)	e2 studio V2023-01 (23.1.0) from Renesas Electronics Corp.
C compiler (e2studio)	CC-RL V1.12.00 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 V5.10.1 from IAR Systems Corp.
C compiler (IAR)	IAR C/C++ Compiler for Renesas RL78 V5.10.1.2667 from IAR Systems Corp.
Smart Configurator	V.1.5.0 from Renesas Electronics Corp.
Board support package (r_bsp)	V.1.40 from Renesas Electronics Corp.

3. Hardware Descriptions

3.1 Example of Hardware Configuration

Figure 3-1 shows an example of the hardware configuration used in the application note.

Figure 3-1 Hardware Configuration



Note 1. This schematic circuit diagram is simplified to show the outline of connections. When creating circuits, design them so that they meet electrical characteristics by properly performing pin processing. (Connect input-only ports to V_{DD} or V_{SS} individually through a resistor.)

3.2 List of Pins to be Used

Table 3-1 lists the pins to be used and their functions.

Table 3-1 Pins to be Used and Their Functions

Pin name	I/O	Function
P04 / TxD0	Output	Data transmission pin
P05 / RxD0	Input	Data reception pin

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 Explanation

4.1 Setting of Option Byte

Table 4-1 shows the option byte settings.

Table 4-1 Option Byte Settings

Address	Setting Value	Contents
000C0H	11101111B	Disables the watchdog timer. (Counting stopped after reset)
000C1H	11111011B	SPOR detection voltage: Reset mode At rising edge TYP. 2.57 V (2.44 V ～ 2.68 V) At falling edge TYP. 2.52 V (2.40 V ～ 2.62 V)
000C2H	11111001.B	High-speed on-chip oscillator clock (f_{IH}): 16 MHz
000C3H	10000101B	Enables on-chip debugging

4.2 List of Constants

Table 4-2 Constants lists the constants that are used in the sample code.

Table 4-2 Constants

Constant Name	Setting Value	Description
s_messageok0[4]	"OK¥r¥n"	Response message to reception of “T”.
s_messageok[4]	"ok¥r¥n"	Response message to reception of “t”.
s_messageuc[4]	"UC¥r¥n"	Response message to reception of characters other than “T” or “t”.
s_messagefe[4]	"FE¥r¥n"	Response message to a framing error.
s_messagepe[4]	"PE¥r¥n"	Response message to a parity error.
s_messageoe[4]	"OE¥r¥n"	Response message to an overrun error.

4.3 List of Variables

Table 4-3 lists global variables.

Table 4-3 Global Variables

Type	Variable Name	Contents	Function Used
uint8_t	g_uart0rxbuf	Receive data buffer	main()
uint8_t	g_uart0rxerr	Receive error factor	main(), r_Config_UART0_callback_receiveend(), r_Config_UART0_callback_error()
MD_STATUS	g_uart0txend	transmission completion flag	main(), r_Config_UART0_callback_sendend()
uint8_t	gp_uart0_tx_address	Transmit data pointer	R_Config_UART0_Send(), r_Config_UART0_interrupt_send()
uint16_t	g_uart0_tx_count	Transmit data number counter	R_Config_UART0_Send(), r_Config_UART0_interrupt_send()
uint8_t	gp_uart0_rx_address	Receive data pointer	R_Config_UART0_Receive(), r_Config_UART0_interrupt_receive(), r_Config_UART0_interrupt_error()
uint16_t	g_uart0_rx_count	Receive data number counter	R_Config_UART0_Receive(), r_Config_UART0_interrupt_receive()
uint16_t	g_uart0_rx_length	Receive data number	R_Config_UART0_Receive(), r_Config_UART0_interrupt_receive()

4.4 List of Functions

Table 4-4 shows a list of functions.

Table 4-4 Functions

Function Name	Outline
main()	Main processing
r_Config_UART0_interrupt_send()	UART0 transmission end interrupt handling
r_Config_UART0_interrupt_receive()	UART0 reception end interrupt handling
r_Config_UART0_interrupt_error()	UART0 error interrupt handling

4.5 Specification of Functions

The function specifications of the sample code are shown below.

main()	
Outline	Main processing
Header	r_cg_macrodriver.h、r_cg_userdefine.h、Config_UART0.h
Declaration	void main(void);
Description	This function sends the data corresponding to the data received from the counterpart device to the counterpart device.
Argument	None
Return Value	None

r_Config_UART0_interrupt_send()	
Outline	UART0 transmission end interrupt handling
Header	r_cg_macrodriver.h、r_cg_userdefine.h、Config_UART0.h
Declaration	#pragma interrupt r_Config_UART0_interrupt_send(vect=INTST0)
Description	This function starts sending data. It then updates the send data pointer and the number of data to be sent counter.
Argument	None
Return Value	None

r_Config_UART0_interrupt_receive()	
Outline	UART0 reception end interrupt handling
Header	r_cg_macrodriver.h、r_cg_userdefine.h、Config_UART0.h
Declaration	#pragma interrupt r_Config_UART0_interrupt_receive(vect=INTSR0)
Description	This function stores the received data in RAM. It then updates the receive data pointer and the receive data count counter.
Argument	None
Return Value	None

r_Config_UART0_interrupt_error()	
Outline	UART0 error interrupt handling
Header	r_cg_macrodriver.h、r_cg_userdefine.h、Config_UART0.h
Declaration	#pragma interrupt r_Config_UART0_interrupt_error(vect=INTSRE0)
Description	This function stores the error factor in RAM and clears the receive error flag.
Argument	None
Return Value	None

4.6 Flowcharts

4.6.1 Main Processing

Figure 4-1 and 4-2 show flowcharts of the main processing.

Figure 4-1 Main Processing (1/2)

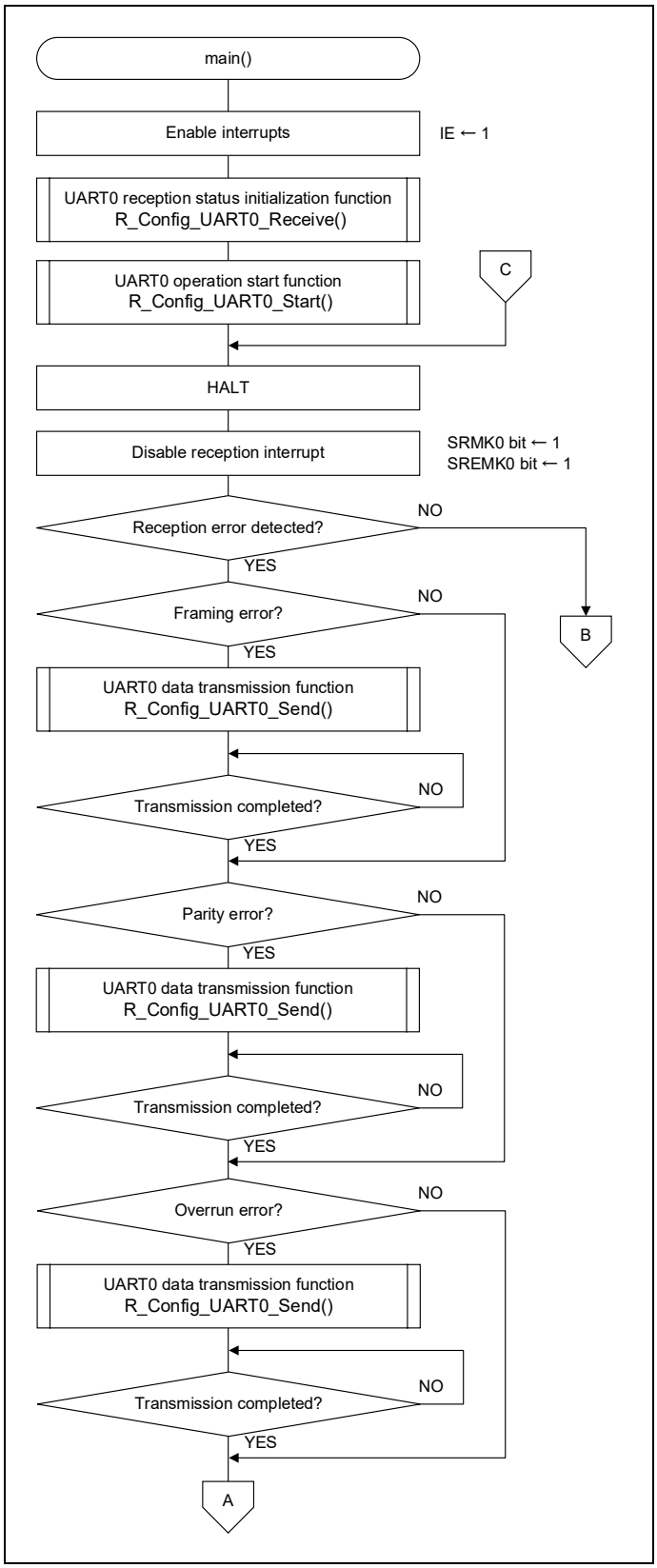
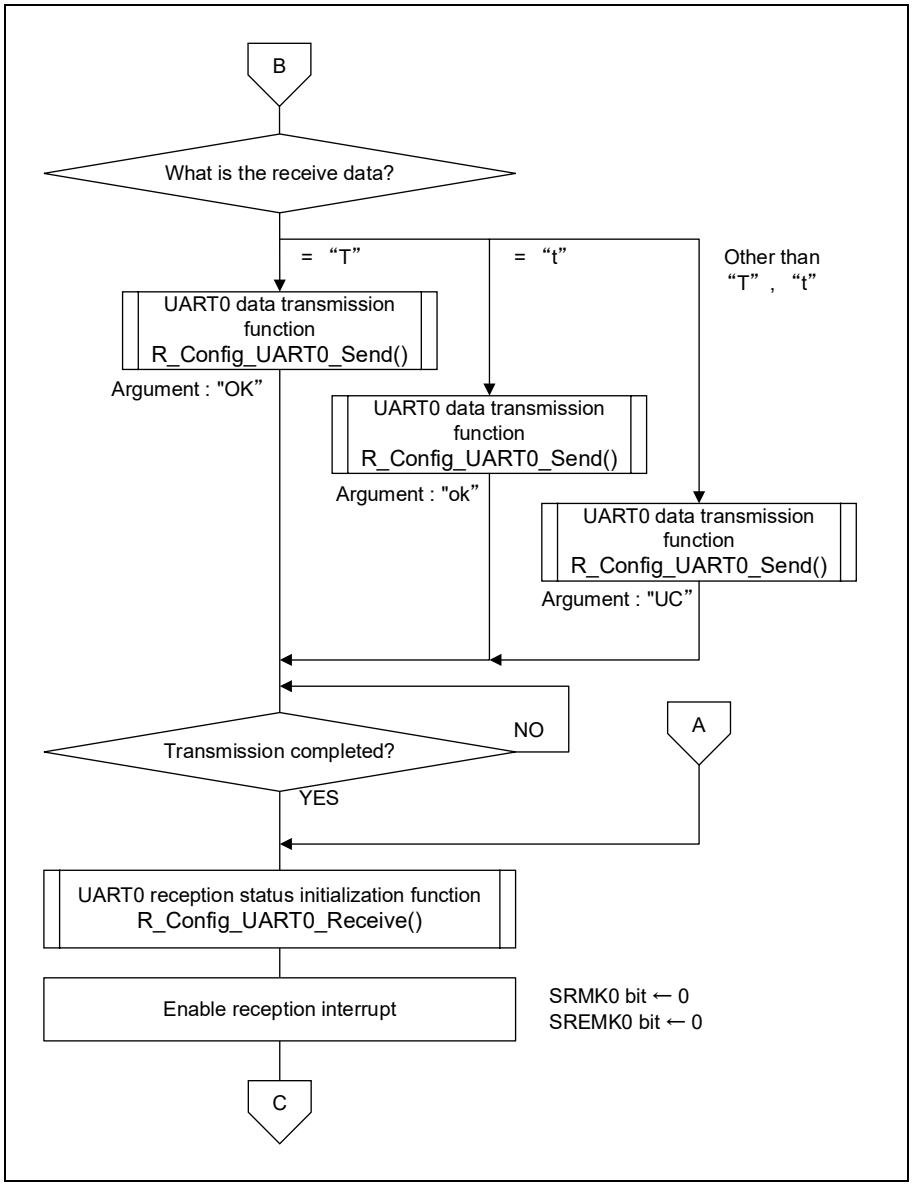


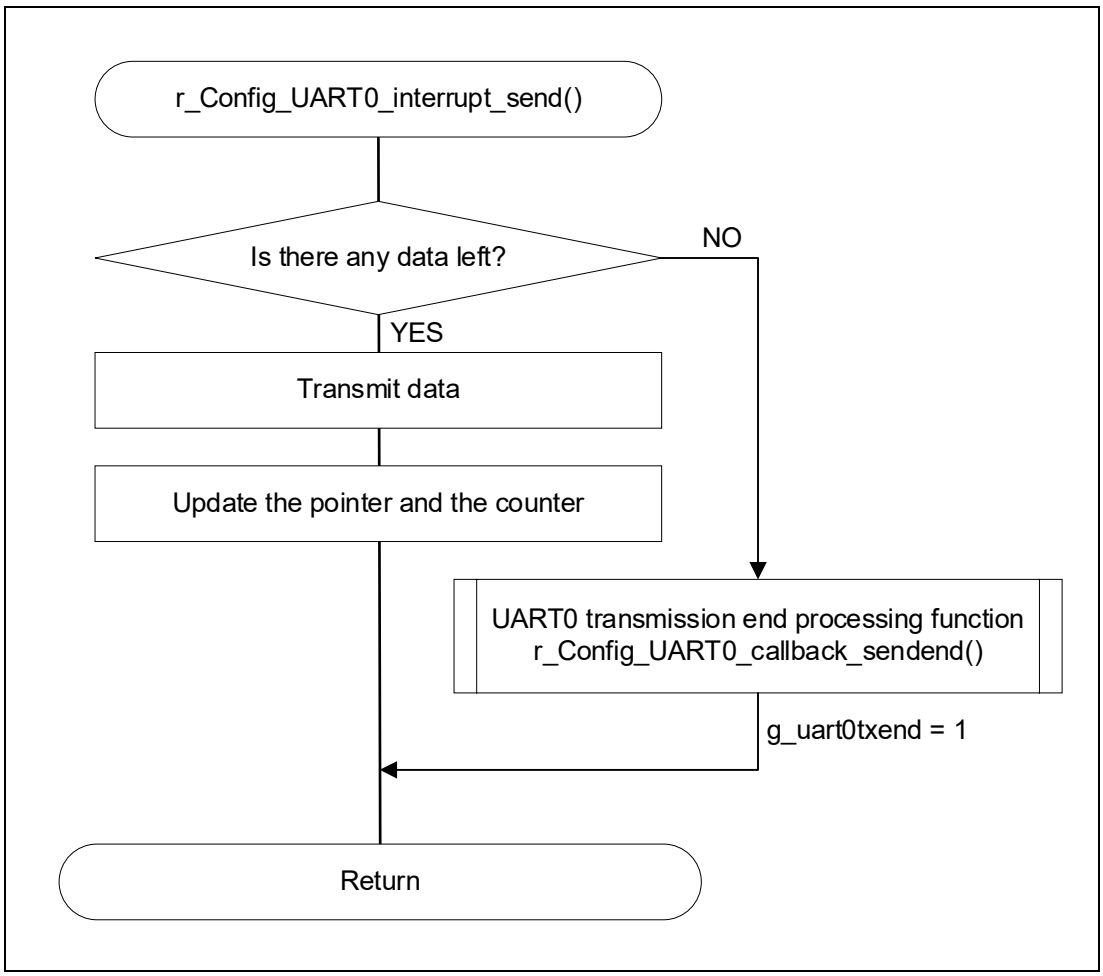
Figure 4-2 Main Processing (2/2)



4.6.2 UART0 Transmission End Interrupt Handling

Figure 4-3 shows the flowchart of the UART0 transmission end interrupt handling.

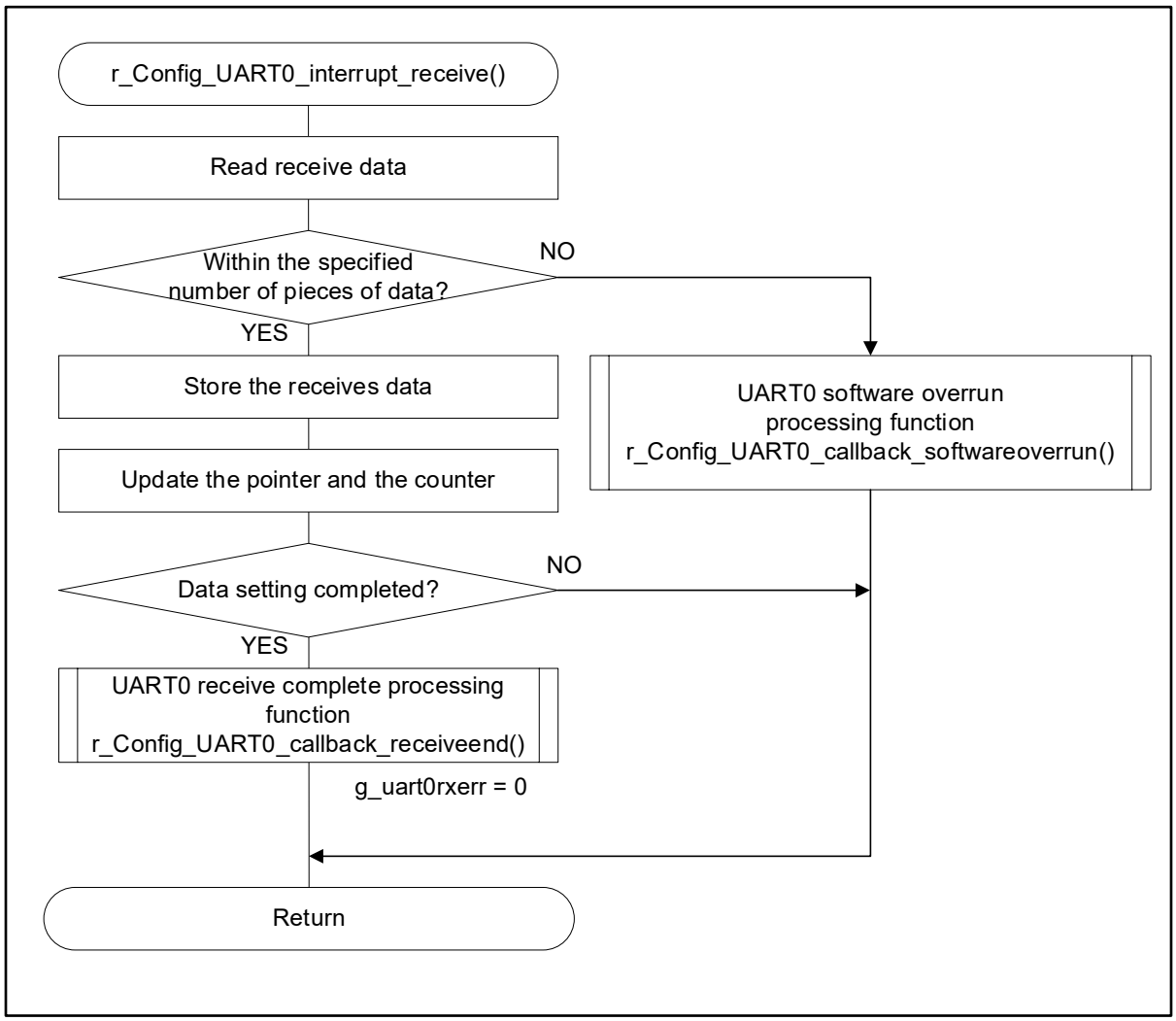
Figure 4-3 UART0 Transmission End Interrupt Handling



4.6.3 UART0 Reception End Interrupt Handling

Figure 4-4 shows the flowchart of the UART0 reception end interrupt handling.

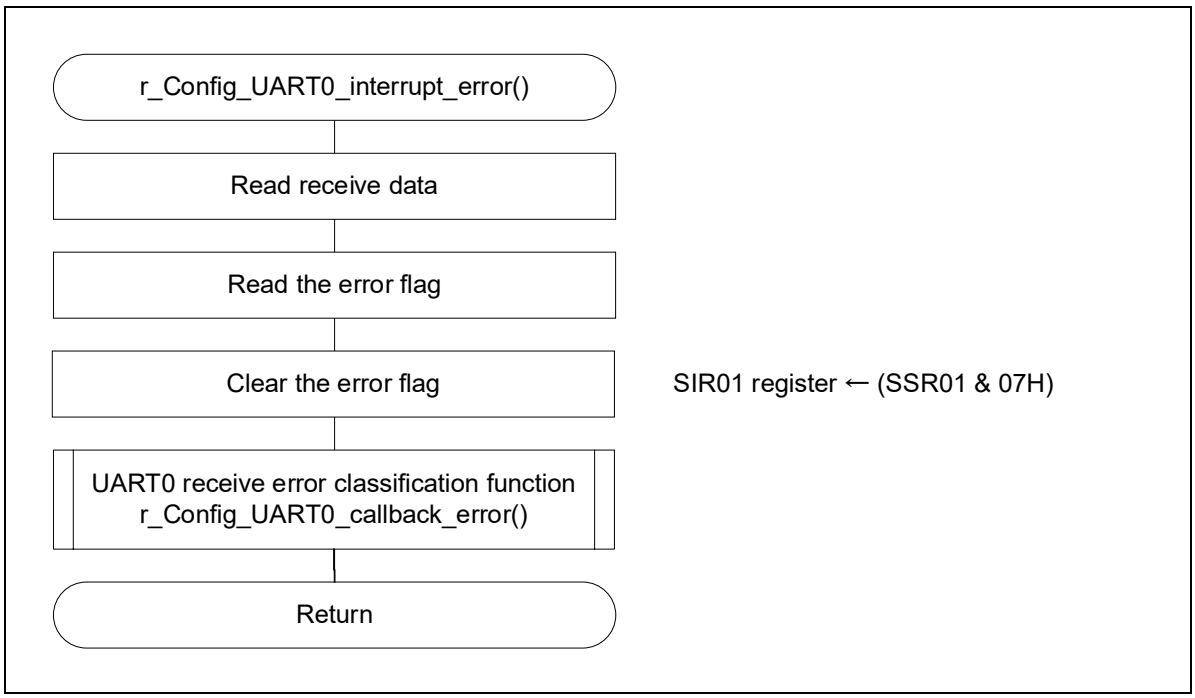
Figure 4-4 UART0 Reception End Interrupt Handling



4.6.4 UART0 Error Interrupt Handling

Figure 4-5 shows the flowchart of the UART0 error interrupt handling.

Figure 4-5 UART0 Error Interrupt Handling



5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

RL78/G15 User’s Manual: Hardware (R01UH0959)
RL78 family user's manual software (R01US0015)
The latest versions can be downloaded from the Renesas Electronics website.

Technical update
The latest versions can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan.20.2023	-	First Edition
1.01	Apr.5.2023	Page.6	Update the Operation Confirmation Conditions IAR added