

V850E2/ML4

R01AN1226EJ0100

Rev.1.00

UARTJ Control

Jun. 22, 2012

Abstract

This document describes how to set up the UARTJ and also gives an outline of the operation and describes the procedures for using a sample program.

The features of the operation are described below:

- Executes serial communication between the UARTJ0 and the UARTJ1. The UARTJ1 transmits data and the UARTJ0 receives the data.
- The baud rate is 19200 bps
- The internal RAM has a 16-byte user transmit array and a 16-byte user receive array.

Products

V850E2/ML4

Integrated development environments

CubeSuite+, GHS MULTI V5.1.7D, and IAR for V850 Kickstart V3.80.

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications	3
2. Operation Confirmation Conditions	4
3. Hardware	5
3.1 Hardware Configuration	5
3.2 Pin(s) Used	6
4. Software	7
4.1 Operation Overview	7
4.2 Required Memory Size	8
4.3 File Composition	9
4.4 Option-Setting Memory	10
4.5 Variables	11
4.6 Function(s)	12
4.7 Function Specification(s)	13
4.8 Flowchart(s)	17
4.8.1 Main Processing	17
4.8.2 Initialize UARTJ0 as receiver	18
4.8.3 Initialize UARTJ1 as transmitter	19
4.8.4 Transmit data by UARTJ1	20
4.8.5 Interrupt: UARTJ0 error	21
4.8.6 Interrupt: UARTJ0 reception	22
4.8.7 Interrupt: UARTJ1 error	23
4.8.8 Interrupt: UARTJ1 transmission end	24
5. Sample Code	25
6. Reference Documents	25

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	V850E2/ML4
Operating frequency	200MHz (PLL multiplies the oscillator input frequency (fx: 10MHz) by 20.)
Operating voltage	3.3V
Integrated development environment	CubeSuite+ V1.00
	GHS MULTI V5.1.7D
	IAR for V850 Kickstart V3.80.1
C compiler	CX V1.20(CubeSuite+), optimization: default
	C-V850E 5.1.7 RELEASE(GHS MULTI) , optimization: default
	IAR C/C++ Compiler for V850 3.80.1 [Kickstart] (3.80.1.30078), optimization: default
Operating mode	Normal operation mode
Sample code version	V1.00
Board used	RSK board
Device used	E1 emulator or MINICUBE
Tool used	none

3. Hardware

3.1 Hardware Configuration

Figure 3.1 shows a Connection Example.

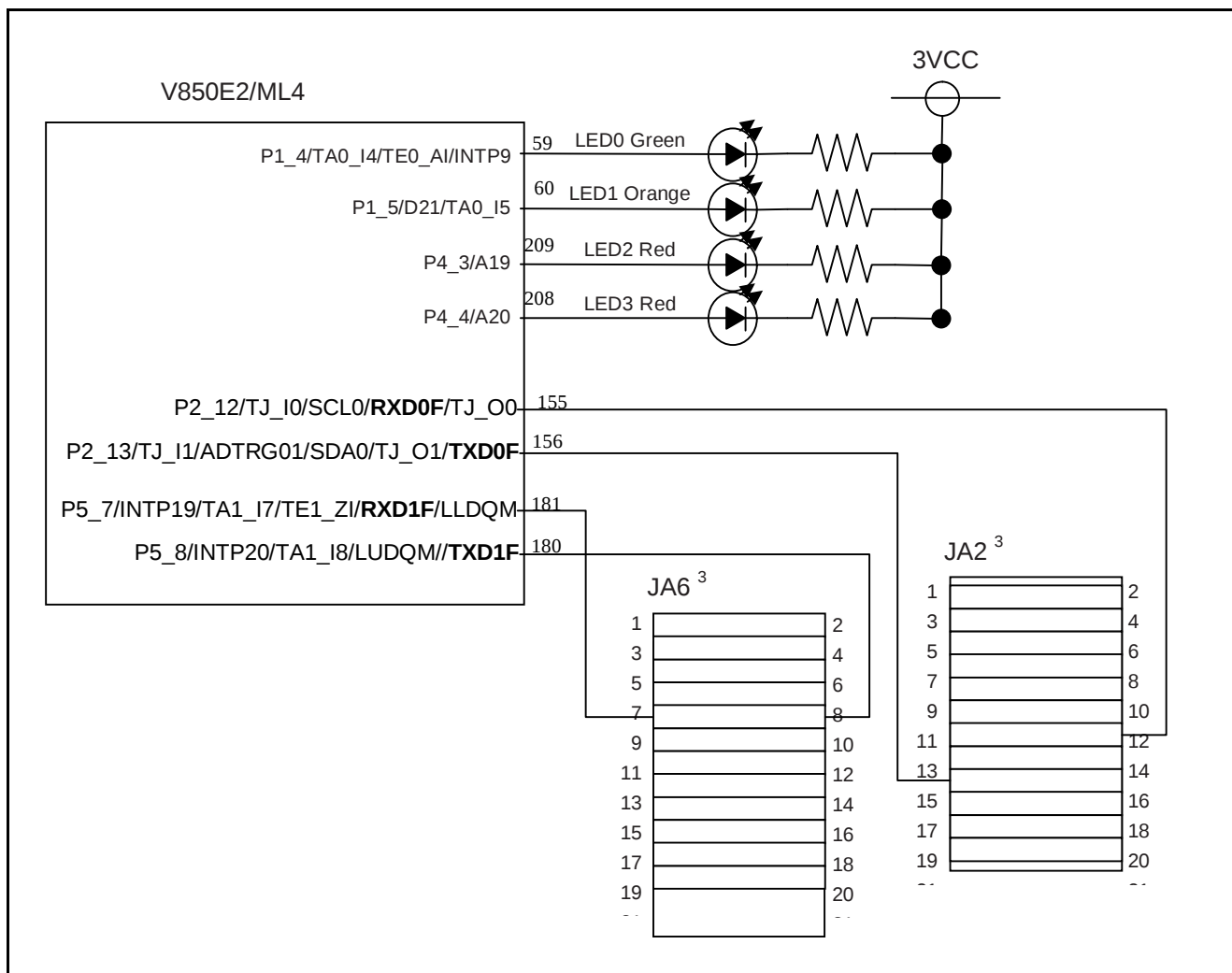


Figure 3.1 Connection Example

³ JA2-6, JA2-8, JA6-7, JA6-8 is not connected to MCU by default. Connect them with 0-ohm resistor, or connect the pins of MCU directly.

3.2 Pin(s) Used

Table 3.1 lists the Pins Used and Its Function.

Table 3.1 Pins Used and Its Functions

Pin Name	I/O	Function
PORT P1_4	output	Port mode, output, LED0
PORT P1_5	output	Port mode, output, LED1
PORT P4_3	output	Port mode, output, LED2
PORT P4_4	output	Port mode, output, LED3
TXD0F	output	UARTJ0 output (not used)
RXD0F	input	UARTJ0 input
TXD1F	output	UARTJ1 output
RXD1F	input	UARTJ1 input (not used)

4. Software

4.1 Operation Overview

Operation Overview is described in the following figure. The main() function initializes each functions, transmit the data, and waits interrupt. When INTUARJ0IR occurred, save the received data and set the reception flag to 1. If the reception flag is detected in the main loop, restart the transmission.

Figure4.1 shows the Sequence diagram.

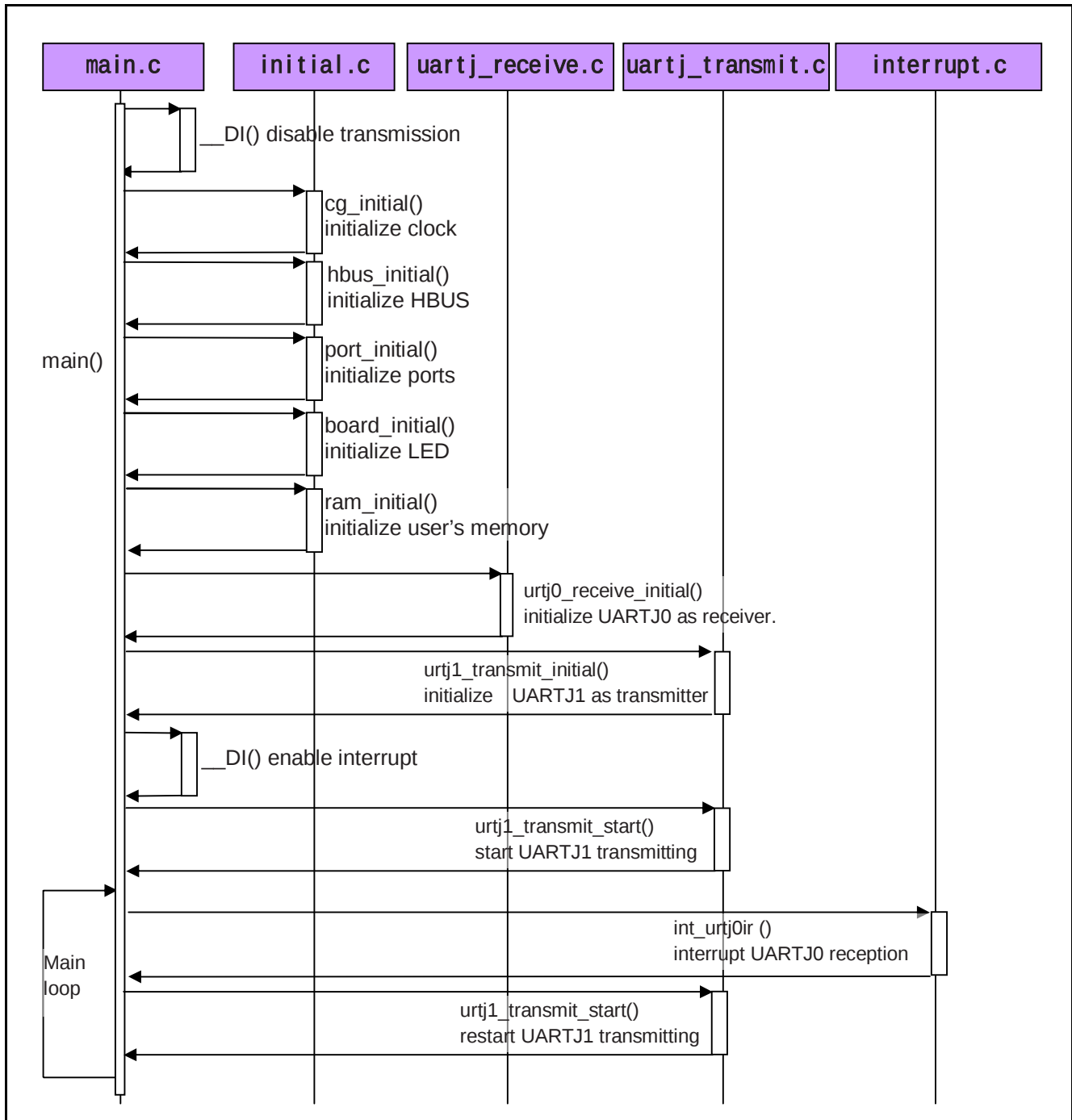


Figure4.1 Sequence diagram

4.2 Required Memory Size

Table 4.1 lists the Required Memory Size. (CubeSuite+, optimization=default)

Table 4.1 Required Memory Size

Memory Used	Size	Remarks
ROM	5900	Shown as ROM area size in map file
RAM	4144	Shown as RAM area size in map file
Maximum user stack usage	4	CubeSuite+ stack estimation tool calculated.
Maximum interrupt stack usage	352	The same as above.

Note: The required memory size varies depending on the C compiler version and its options.

4.3 File Composition

Table 4.2 lists the Files Used in the Sample Code. Files not generated by the integrated development environment should not be listed in this table.

Table 4.2 Files Used in the Sample Code

File Name	Outline	Remarks
crtE.s	Initialize hardware	Only in the project for CubeSuite+
startup.s		Only in the project for GHS MULTI
V850E2ML4.dir	Linker directive file	Only in the project for CubeSuite+
V850E2_ML4.ld		Only in the project for GHS MULTI
vector.s	Vector table	Only in the project for GHS MULTI
uartj.h	Declare variables and functions.	
df4022_800.h	Declare register macros for V850E2/ML4	Only in the project for GHS MULTI
main.c	Main routine	
initial.c	Initialize software	
uartj_receive.c	Initialize UARTJ0 as receiver.	
uartj_transmit.c	Initialize UARTJ1 as transmitter	
interrupt.c	Interrupt routines	

4.4 Option-Setting Memory

This sample does not specify any option-bytes. Specify them if necessary.

4.5 Variables

Table 4.3 lists the Global Variables.

Table 4.3 Global Variables

Type	Variable Name	Contents	Function Used
unsigned char	rxdata_urtj0[BUFFER_SIZE]	Buffer for the received data.	void ram_initial(void) __interrupt void int_urtj1ire(void) __interrupt void int_urtj0ire(void) __interrupt void int_urtj0ir(void) void urtj1_transmit_start(void)
unsigned char	flag_transmit_over	Transmission end flag	void ram_initial(void) void display(void) __interrupt void int_urtj1ire(void) __interrupt void int_urtj1ic(void) __interrupt void int_urtj0ire(void)
unsigned char	flag_receive_over	Reception end flag	void ram_initial(void) void display(void) __interrupt void int_urtj1ire(void) __interrupt void int_urtj0ire(void) __interrupt void int_urtj0ir(void)
unsigned char	flag_fifo_error	FIFO error flag	void ram_initial(void) void display(void) __interrupt void int_urtj1ire(void) __interrupt void int_urtj0ire(void)
unsigned char	flag_uart_error	UART error flag	void ram_initial(void) void display(void) __interrupt void int_urtj1ire(void) __interrupt void int_urtj0ire(void)
unsigned char	txdata_urtj1[BUFFER_SIZE]	Buffer for the data to be transmitted	void ram_initial(void) void urtj1_transmit_start(void)

4.6 Function(s)

Table 4.4 lists the Function(s).

Table 4.4 Function(s)

Function Name	Outline
void main(void)	Calls necessary initialization functions and waits interrupt in infinite loop.
void cg_initial(void)	Initializes the special clock frequency control register.
void hbus_initial(void)	Initializes the AHB bus
void port_initial(void)	Sets up ports and their mode.
void board_initial(void)	Initializes the LEDs
void ram_initial(void)	Sets up the initial state of the user RAM
void urtj0_receive_initial(void)	Initialize UARTJ0 as receiver
void urtj1_transmit_initial(void)	Initialize UARTJ1 as transmitter
void urtj1_transmit_start(void)	Start transmission
void wait(int number)	Wait some time
void display(void)	Display errors and transmission status on LED.
__interrupt void int_urtj1ire(void)	Error interrupt on UART1
__interrupt void int_urtj1ic(void)	Transmission end interrupt on UART1
__interrupt void int_urtj0ire(void)	Error interrupt on UART0
__interrupt void int_urtj0ir(void)	Reception end interrupt on UART0

4.7 Function Specification(s)

The following tables list the sample code function specification(s).

main()	
Outline	Main routine
Header	-
Declaration	void main(void)
Description	Calls necessary initialization functions, starts transmission, and waits interrupt in infinite loop.
Arguments	-
Return Value	-
cg_initial()	
Outline	Initialize clock
Header	uartj.h
Declaration	void cg_initial(void)
Description	Initializes the special clock frequency control register.
Arguments	none
Return Value	none
hbus_initial()	
Outline	Initialize H-bus
Header	uartj.h
Declaration	void hbus_initial(void)
Description	Initializes AHB-bus
Arguments	none
Return Value	none
port_initial()	
Outline	Sets up ports and their mode.
Header	uartj.h
Declaration	void port_initial (void)
Description	Sets up ports in port-mode, output, for controlling UARTJ0, UARTJ1, and LEDs.
Arguments	none
Return Value	none
board_initial()	
Outline	Initialize board
Header	uartj.h
Declaration	void board_initial(void)
Description	Initialize LED on the board.
Arguments	-
Return Value	-

ram_initial()

Outline	Initialize users' memory.
Header	uartj.h
Declaration	void ram_initial(void)
Description	Set the reception buffers, reception/transmission end flags, and FIFO/UARTJ error flags to 0. Set initial data to the transmission buffer.
Arguments	-
Return Value	-

wait()

Outline	Wait some time
Header	uartj.h
Declaration	void dma0_initial(void)
Description	Initialize DMA0 as software-triggered DMAC.
Arguments	-
Return Value	-

display()

Outline	Display status on LED
Header	uartj.h
Declaration	void dma1_initial(void)
Description	Displays error and transmission status on LED
Arguments	-
Return Value	-

urtj0_receive_initial()

Outline	Initialize UARTJ0
Header	uartj.h
Declaration	void urtj0_receive_initial(void)
Description	Initialize UARTJ0 as receiver.
Arguments	-
Return Value	-

urtj1_transmit_initial()

Outline	Initialize UARTJ1
Header	uartj.h
Declaration	void urtj0_transmit_initial(void)
Description	Initialize UARTJ1 as transmitter.
Arguments	-
Return Value	-

urtj1_transmit_start()

Outline	Start transmission on UARTJ1
Header	-
Declaration	void urtj1_transmit_start(void)
Description	Let UARTJ1 transmit the data in the buffer
Arguments	-
Return Value	-

int_urtj1ire()

Outline	Reception Interrupt on UARTJ1
Header	-
Declaration	__interrupt void int_urtj1ire(void)
Description	1. Set error flags by error register. 2. Stop UARTJ0, UARTJ1 3. Save the received data in FIFO buffer 4. Clear error status registers. 5. Displays on LED. 6. Restart UARTJ0, UARTJ1. 7. Wait some time.
Arguments	-
Return Value	-

int_urtj1ic()

Outline	Transmission end interrupt on UARTJ1
Header	-
Declaration	__interrupt void int_urtj1ic(void)
Description	Set transmission end flag to 1, and display on LED.
Arguments	-
Return Value	-

int_urtj0ire()

Outline	Transmission error interrupt on UARTJ0
Header	-
Declaration	__interrupt void int_urtj0ire(void)
Description	1. Set error flags by error register. 2. Stop UARTJ0, UARTJ1 3. Save the received data in FIFO buffer 4. Clear error status registers. 5. Displays on LED. 6. Restart UARTJ0, UARTJ1. 7. Wait some time.
Arguments	-
Return Value	-

`int_urtj0ir()`

Outline	Transmission end interrupt on UARTJ0
Header	-
Declaration	__interrupt void int_urtj0ir(void)
Description	1. Get number of received data. 2. Save the received data in FIFO buffer to the reception buffer. 3. Set the reception flag to 1. 4. Displays on LED.
Arguments	-
Return Value	-

4.8 Flowchart(s)

4.8.1 Main Processing

Figure4.2 shows the Main Processing.

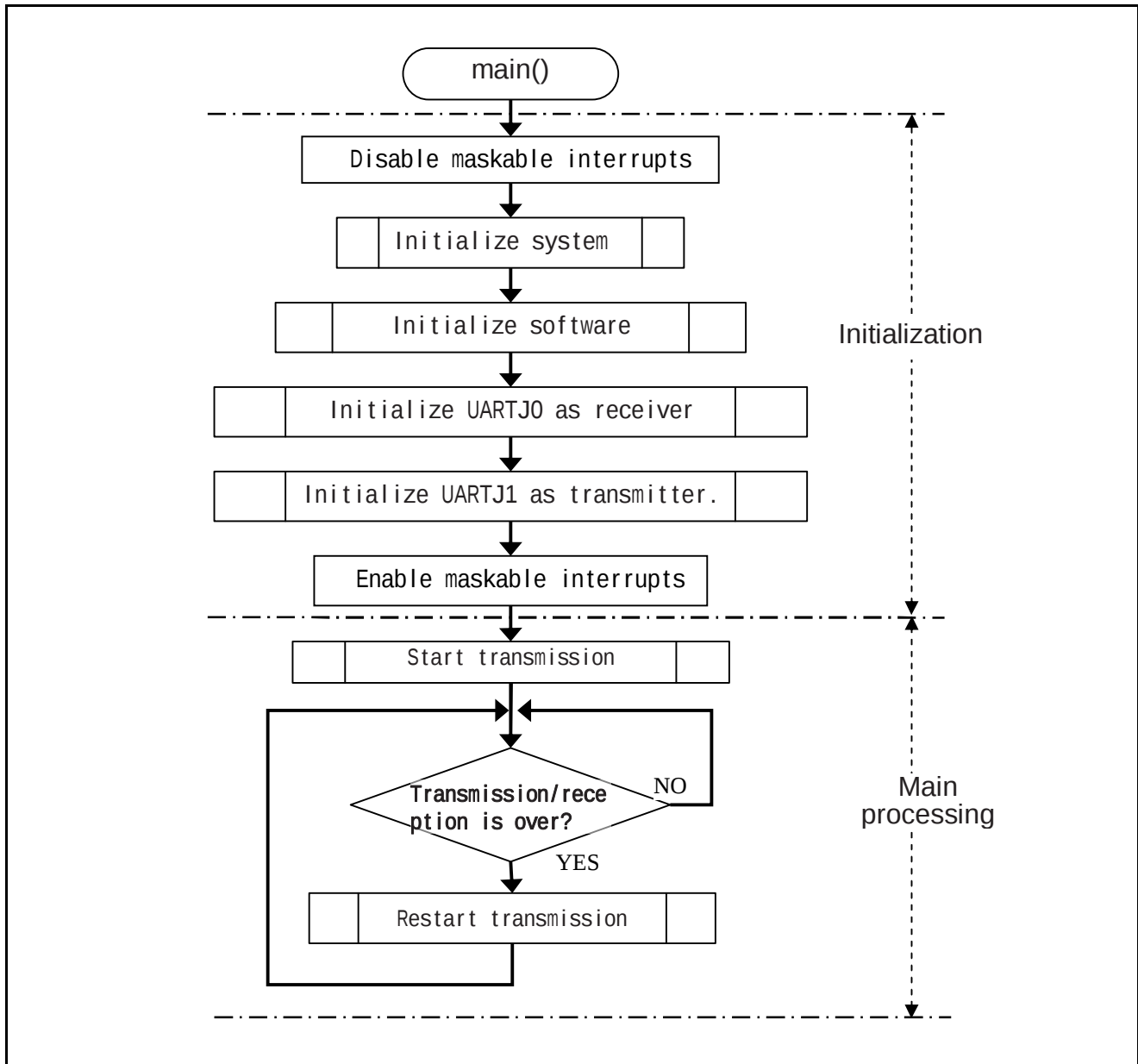


Figure4.2 Main Processing

4.8.2 Initialize UARTJ0 as receiver

Figure4.3 shows flowchart of Initialize UARTJ0 as receiver.

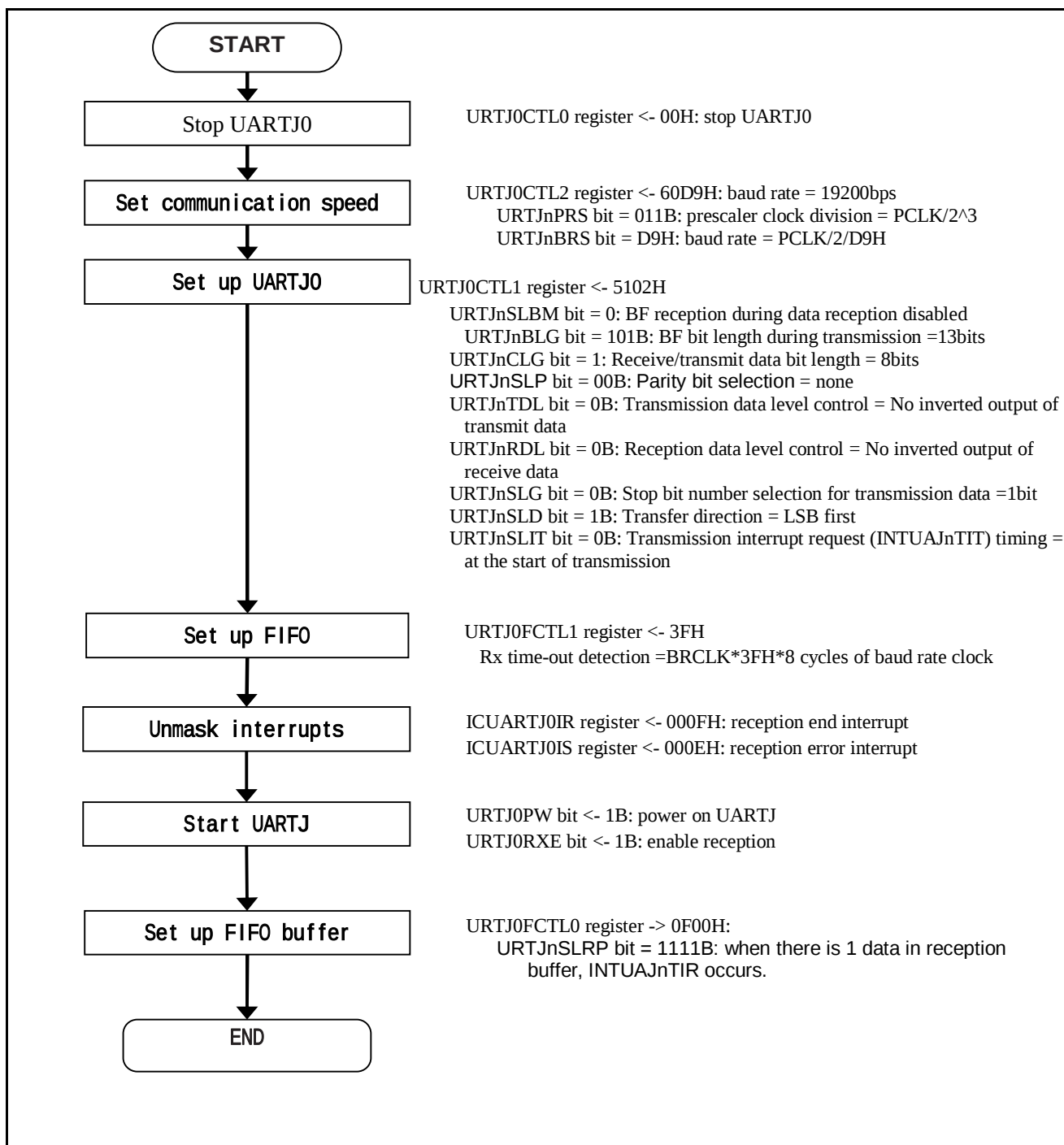


Figure4.3 Initialize UARTJ0 as receiver

4.8.3 Initialize UARTJ1 as transmitter

Figure4.3 shows flowchart of initializing UARTJ1 as transmitter.

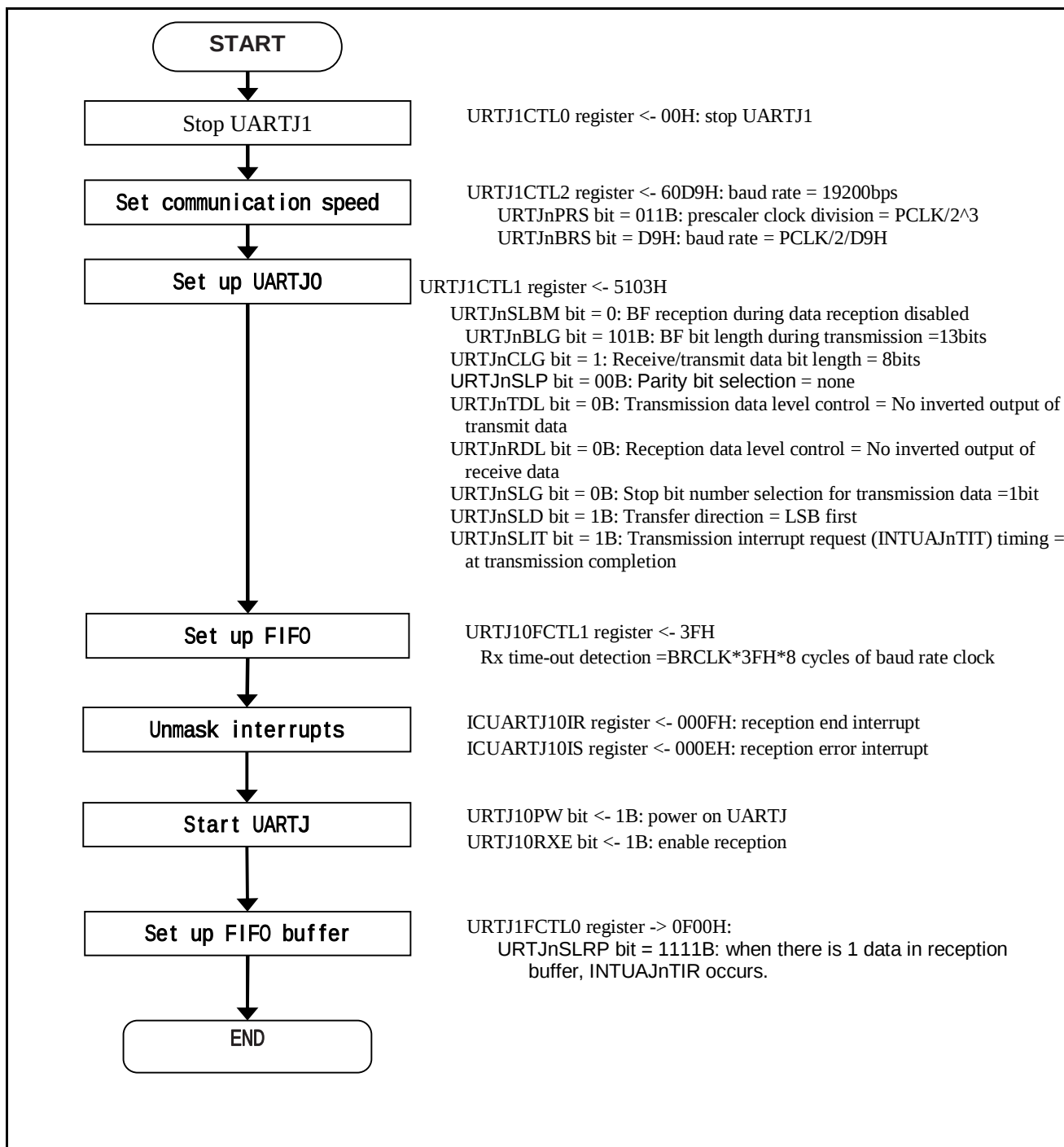


Figure4.4 Initialize UARTJ1 as transmitter

4.8.4 Transmit data by UARTJ1

Figure4.5 shows flowchart of Transmit data by UARTJ1.

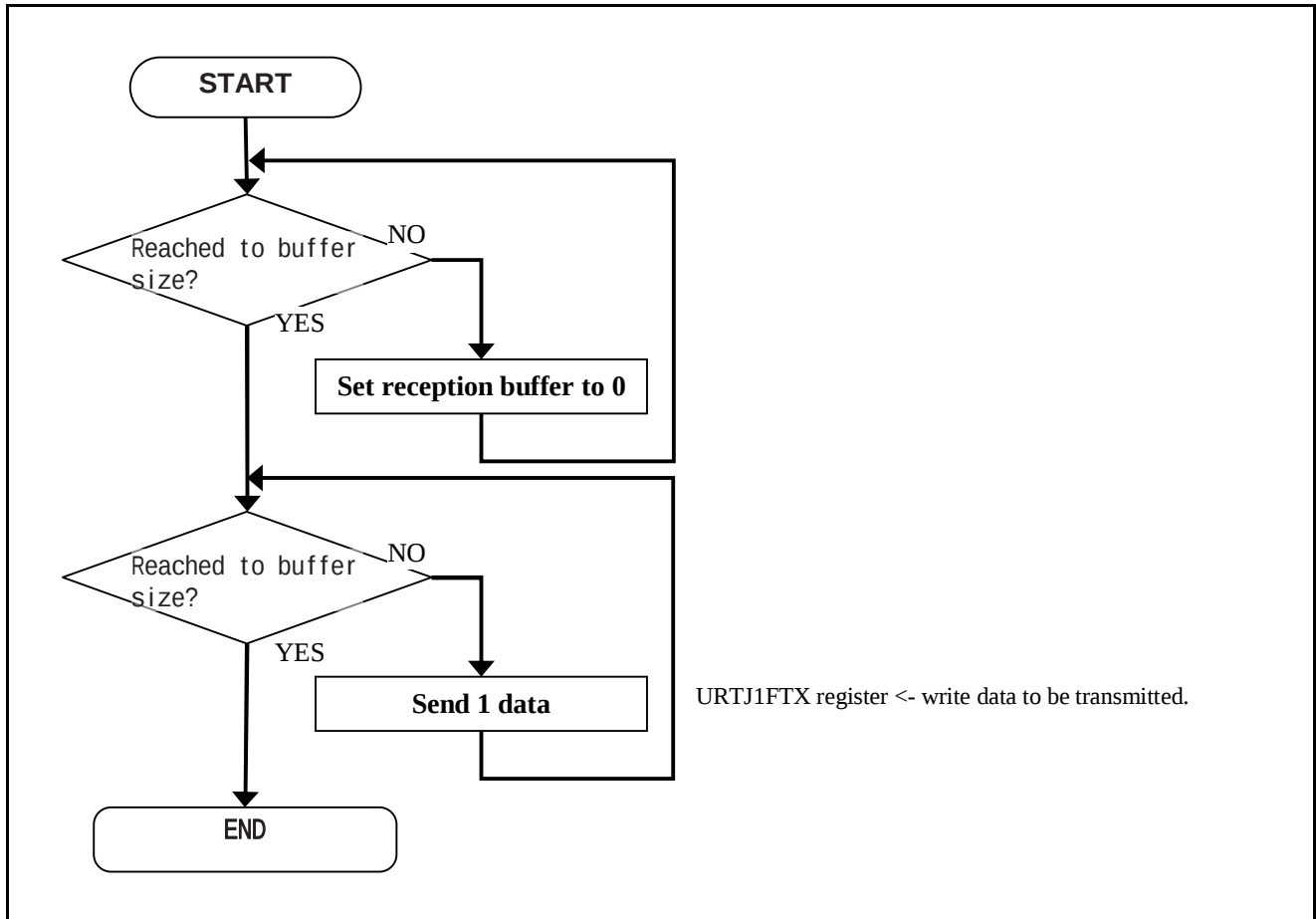


Figure4.5 Transmit data by UARTJ1

4.8.5 Interrupt: UARTJ0 error

Figure4.6 shows flowchart of UARTJ0 error interrupt.

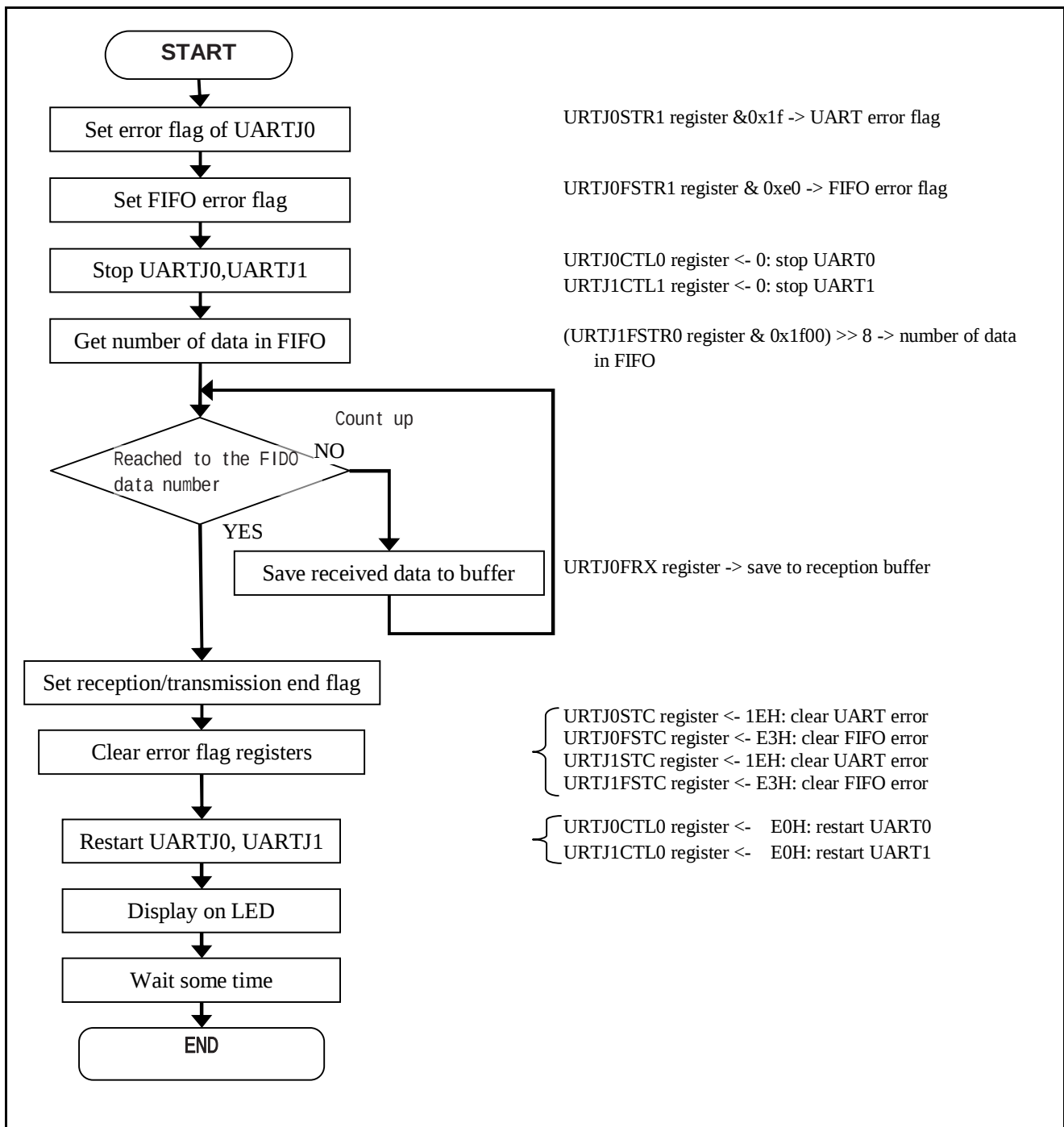


Figure4.6 UARTJ0 error interrupt

4.8.6 Interrupt: UARTJ0 reception

Figure4.7 shows flowchart of UARTJ0 reception interrupt.

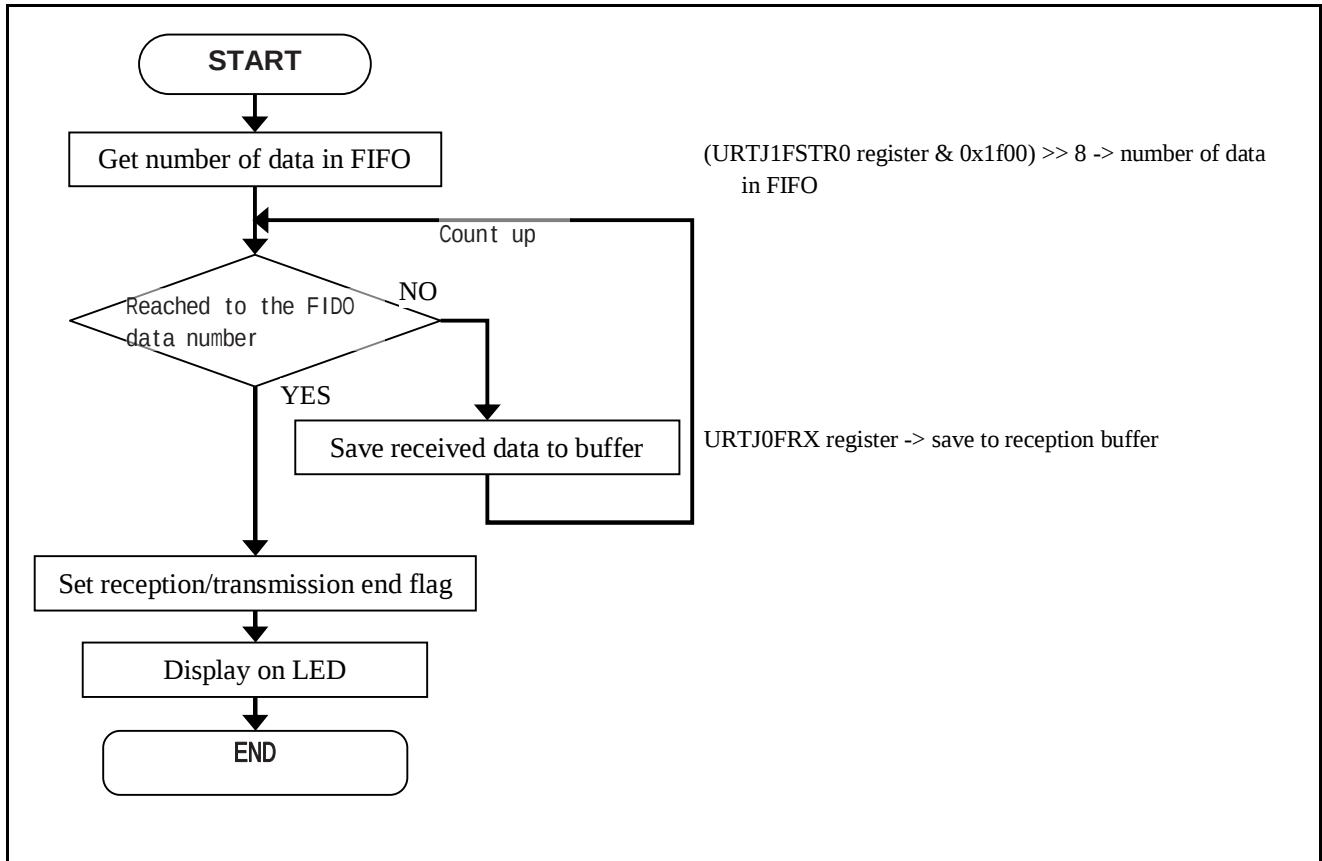


Figure4.7 UARTJ0 reception interrupt

4.8.7 Interrupt: UARTJ1 error

Figure4.8 shows flowchart of UARTJ1 error interrupt.

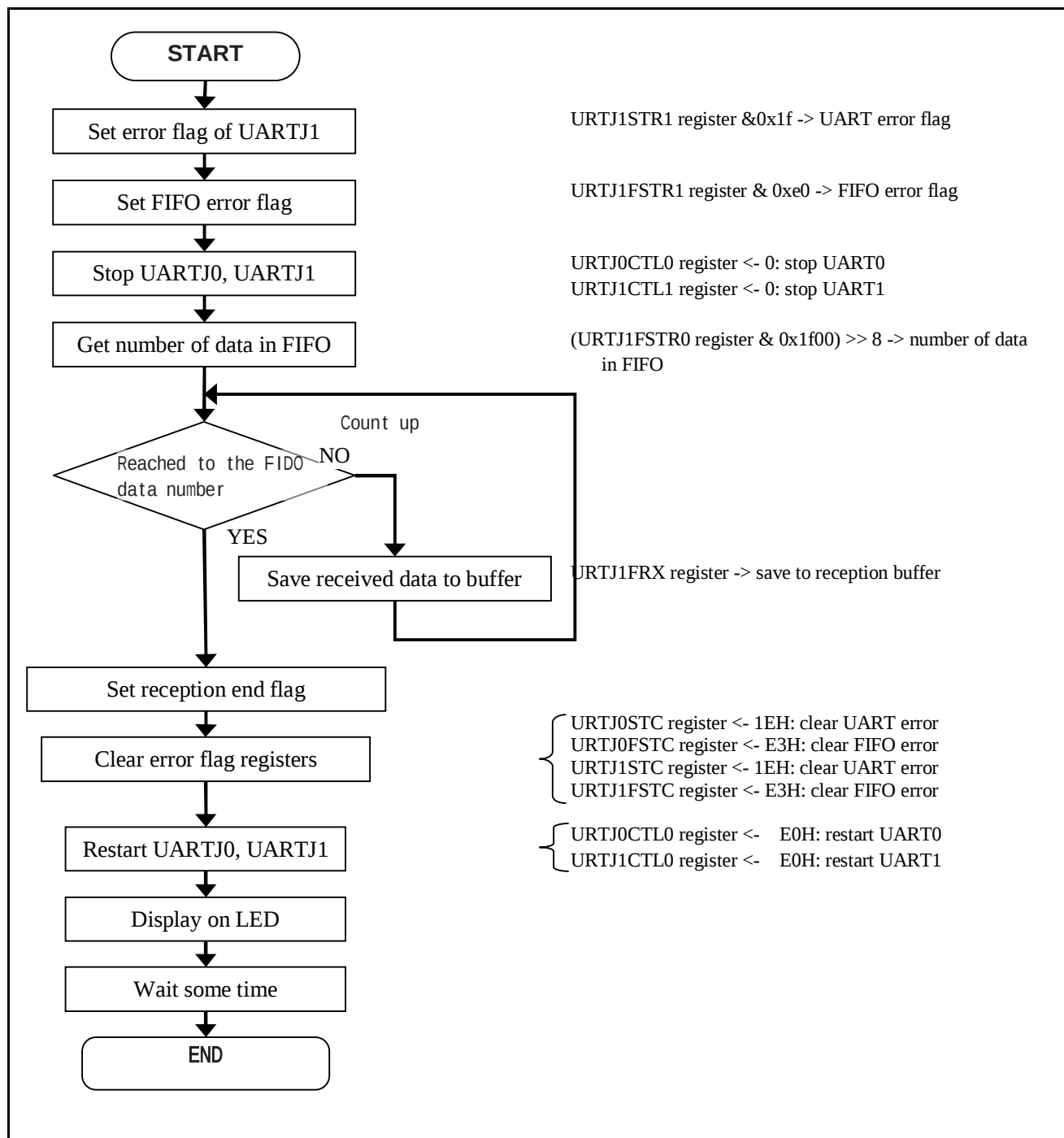


Figure4.8 UARTJ1 error interrupt

4.8.8 Interrupt: UARTJ1 transmission end

Figure4.9 shows flowchart of UARTJ1 transmission end interrupt.

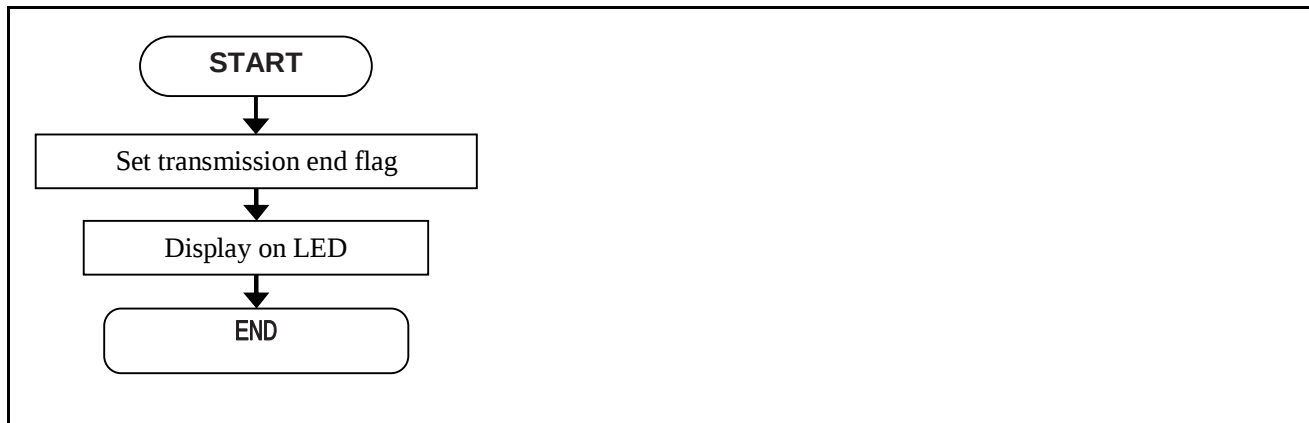


Figure4.9 UARTJ1 transmission end interrupt

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware

V850E2/ML4 User's Manual: Hardware (R01UH0262EJ)

The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website

<http://www.renesas.com>

Inquiries

<http://www.renesas.com/contact/>

REVISION HISTORY	V850E2/ML4 Application Note UARTJ Controller
------------------	--

Rev.	Date	Description	
		Page	Summary
1.00	Jun. 22, 2012	—	First edition issued

All trademarks and registered trademarks are the property of their respective owners.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computer, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, general electronic equipment and industrial robots etc.
"High Quality": Transportation equipment (automobiles, taxis, ships, etc.), traffic control systems, air disaster systems, anti-aircraft systems, and safety equipment etc.
 Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (critical life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with regard to the maximum rating, operating supply voltage range, maximum power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of a failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, the control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of semiconductor reliability alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the installation or use of semiconductor components, including without limitation, the RoHS and ELV Directives. Renesas Electronics assumes no liability for damages or losses resulting as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or disclosed in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

Notes 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

Notes 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

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

Renesas Electronics America Inc.
 2880 Scott Boulevard Santa Clara, CA 95050-2884, U.S.A.
 Tel: +1-408-386-6000, Fax: +1-408-386-6128

Renesas Electronics Canada Limited
 1181 Midland Road, Markham, Ontario L3Y 9C3, Canada
 Tel: +1-416-496-8441, Fax: +1-416-496-5220

Renesas Electronics Europe Limited
 Dulux Meadow, Millbrook Road, Bourne End, Buckinghamshire, SL8 5PH, U.K.
 Tel: +44-1825-436-100, Fax: +44-1825-436-260

Renesas Electronics Europe GmbH
 Amalienstrasse 10, 40472 Düsseldorf, Germany
 Tel: +49-211-45030, Fax: +49-211-4503-1367

Renesas Electronics (China) Co., Ltd.
 7th Floor, Greenland Plaza, No.67 Zhongren Lu, Haidian District, Beijing 100080, P.R.China
 Tel: +86-10-8526-1155, Fax: +86-10-8526-7678

Renesas Electronics (Shanghai) Co., Ltd.
 Unit 204, 205, AZIA Center, No.1225 Lujiacui Road, Pudong District, Shanghai 200120, China
 Tel: +86-21-6877-1915, Fax: +86-21-6877-7668 / -7966

Renesas Electronics Hong Kong Limited
 Unit 1801-1813, 18/F., Tower 2, Grand Century Place, 180 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
 Tel: +852-2399-9314, Fax: +852-2399-9333/9344

Renesas Electronics Taiwan Co., Ltd.
 12F, No. 205, Fu Shing North Road, Taipei, Taiwan
 Tel: +886-2-6175-9600, Fax: +886-2-6175-9670

Renesas Electronics Singapore Pte. Ltd.
 1 HarbourFront Avenue, #09-10, Harbour Bay Tower, Singapore 099202
 Tel: +65-6813-0800, Fax: +65-6813-0001

Renesas Electronics Malaysia Sdn. Bhd.
 Unit 805, Block B, Menara Axiata, Axiata Trade Centre, No. 13, Jln Persiaran Surut, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
 Tel: +603-7555-9980, Fax: +603-7555-9610

Renesas Electronics Korea Co., Ltd.
 11F, Samik Lariel or Bldg., 700-2 Yongsan-Dong, Kangnam-Gu, Seoul 136-080, Korea
 Tel: +82-2-559-3737, Fax: +82-2-559-6141

