RL78/G10

UART Software Handshake Communication CC-RL

R01AN3107EJ0100
Rev.1.00
Oct. 05, 2016

Introduction

This application note explains how to implement the software handshake communication in UART communication through the serial array unit (SAU).

Target Device

RL78/G10 microcomputer

10-pin: ROM = 2 KB, 4 KB (R5F10Y16, R5F10Y17)

16-pin: ROM = 2 KB, 4 KB (R5F10Y46, R5F10Y47)

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 describes how to perform software handshake communication in UART communication through the serial array unit (SAU). In this communication, XON (11H) is used as a transmission enable code and XOFF (13H) is used as a transmission disable code, and the received data is stored in a 16-byte ring buffer. When the free space in the buffer is three bytes or less, XOFF (13H) is issued to disable transmission. When the buffer is read and the number of bytes of data stored in the received data area in the buffer is three bytes or less, XON (11H) is issued to resume transmission.

This sample program performs UART communication between two evaluation boards. Each of the evaluation boards has a LED1 and a LED2 to show the status of the handshake communication. One of the LEDs operates as the data transmission side and the other as the data reception side, therefore the flow of data is unidirectional from the data transmission side to the data reception side. However, since it is necessary for the data reception side to send “XON (transmission enabled)/XOFF (transmission disabled)” to the data transmission side, these control codes are transmitted from the data reception side to the data transmission side. The two evaluations boards have the same specification and the same program, and LED1 is used to distinguish between the data transmission side and the data reception side. (Data transmission side: LED1 is lit. Data reception side: LED1 is not lit.)

エラー! ブックマークが自己参照を行っています。 lists the operating modes. The operating modes are distinguished by LED1. After reset is released, both two evaluation boards operate in initialization mode. In initialization mode LED1 flashes, and in this state the board whose switch is pressed becomes the data transmission side, and the other board becomes the data reception side.

Table 1.1 List of Operating Modes

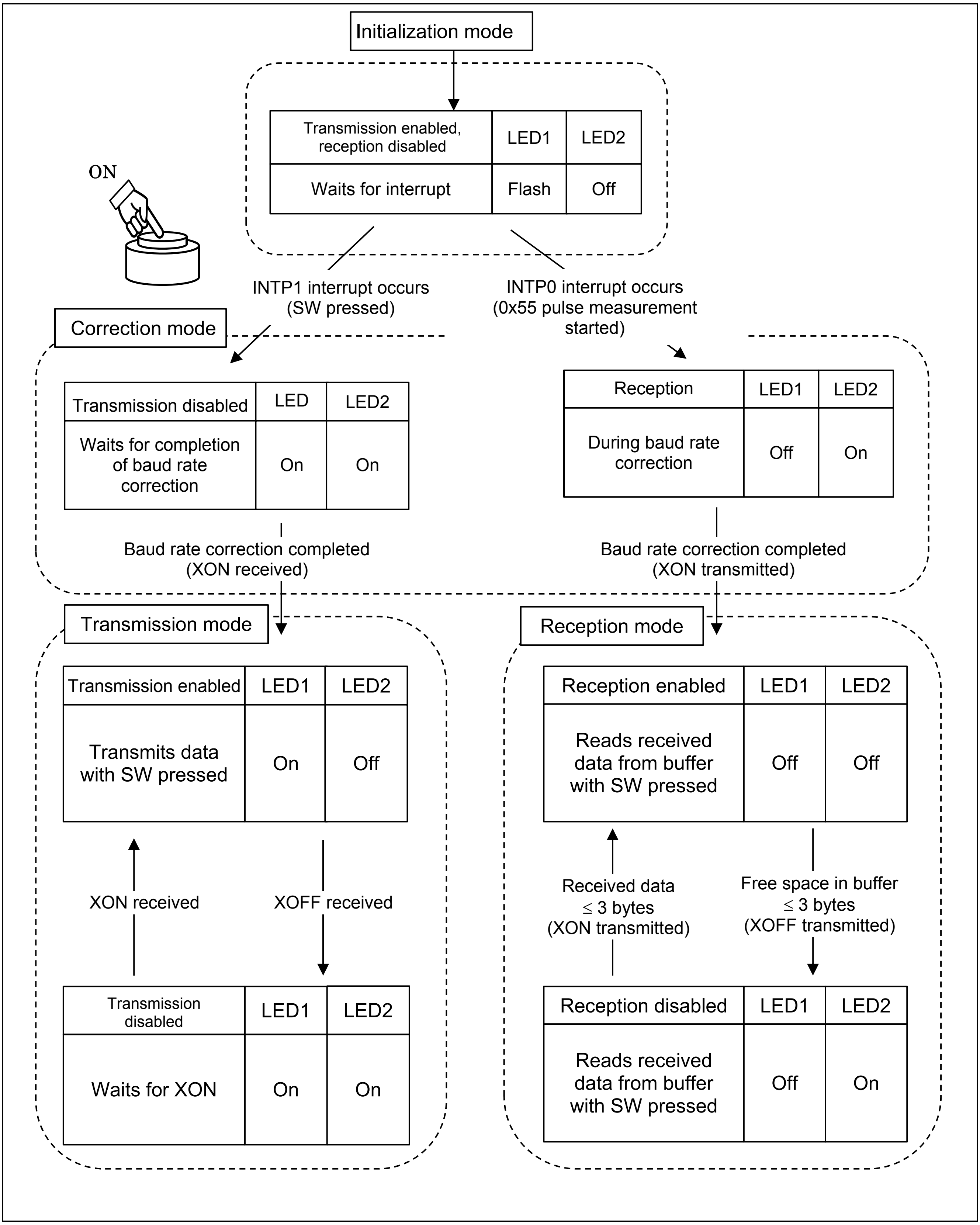
Operating Mode	Outline
Initialization mode	Operates in this mode after reset is released. Waits for an interrupt to be generated, and then switches the operating mode according to the interrupt source. During initialization mode, LED1 flashes with an interval of approximately 500 ms. • If baud rate correction is enabled: Enters correction mode if the switch is pressed or the start of UART communication is detected. • If baud rate correction is disabled: Enters transmission mode if the switch is pressed. Enters reception mode if data is received through the UART.
Correction mode	Operates in this mode if the switch is pressed or the start of UART communication is detected. Performs baud rate correction processing, and when the processing is complete, enters transmission mode or reception mode according to the interrupt source generated in initialization mode. Correction mode can be set to be enabled or disabled in SETTING.inc. For details, see Section エラー! 参照元が見つかりません。 “エラー! 参照元が見つかりません。.”
Transmission mode	This is the operating mode for the data transmission side. This mode is entered if the switch is pressed in initialization mode. If transmission is enabled, then LED2 goes out. If transmission is disabled, then LED2 is lit. When the switch is pressed, if transmission is enabled then data is transmitted through the UART. If transmission is disabled, then data is not transmitted. The data transmission side operates in this mode after initialization and baud rate correction have been completed.
Reception mode	This is the operating mode for the data reception side. If the start of UART communication is detected or data is received through the UART, then this mode is entered. If transmission is enabled, then LED2 goes out. If transmission is disabled, then LED2 is lit. Data transmitted from the data transmission side is stored in the reception buffer, and when the free space in the buffer is three bytes or less, XOFF (transmission disabled) is transmitted to the data transmission side. When the switch is pressed, one byte at a time of received data is read from the reception buffer. If the number of bytes of data stored in the received data area in the buffer is three bytes or less, XON (transmission enabled) is transmitted to the data transmission side. If an error occurs during data reception, then the error status (the lower three bits of the SSR01 register) is stored in the ring buffer. The data reception side operates in this mode after initialization and baud rate correction have been completed.

Table 1.2 shows the on-chip functions to be used and their usage.

Table 1.2 On-chip Functions to be Used and their Usage

On-chip Function	Usage
Serial array unit 0	Performs UART communication using the TXD0 pin (transmission) and the RXD0 pin (reception).
Timer array unit channel 1	Calculates baud rate for UART communication using the input pulse interval measurement function.
External interrupt	Uses INTP0 for detection of the start of UART communication. Uses INTP1 for switch input.
Port function	Uses the P02 pin to control LED1. Uses the P04 pin to control LED2.

shows the state transition diagram (baud rate correction enabled).



1.1 Handshake Communication Overview

As the communication speed increases, the processing on the reception side may not be able to keep up, and data may be lost. In handshake communication, when it does not seem that the processing on the reception side will be able to keep up, data loss is avoided by disabling transmission until the processing is completed. Handshake communication methods may be implemented in hardware or software, and an overview of each of them is given below.

(1) Handshake communication using hardware

Figure 1.1 shows an example of handshake communication that is used relatively often in microcomputers. The RTS and CTS signals are paired, where RTS is used as a transmission request signal (output), and CTS as a transmission enable signal (input). This method can communicate at high speed compared to a method implemented in software, but it is necessary to add these two signal lines.

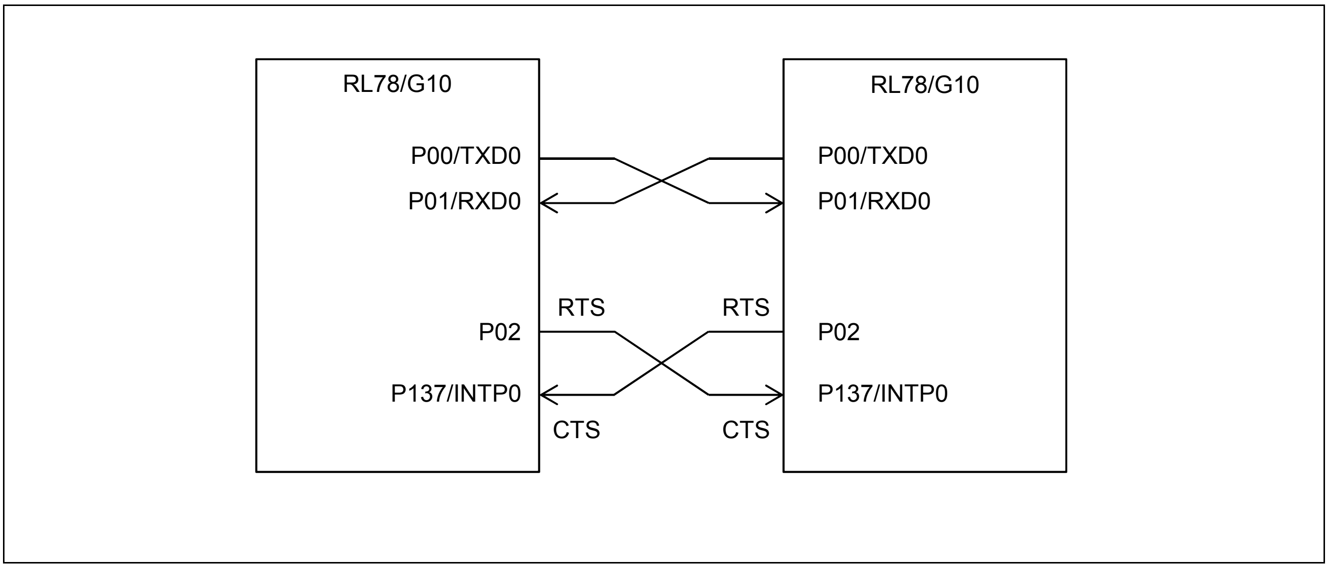


Figure 1.1 Operation of Handshake Communication using Hardware

(1) Handshake communication using software

This performs handshake communication by defining control codes for enabling transmission and disabling transmission, and transmitting them as shown in Figure 1.2. (This sample program uses XON and XOFF, which are defined as control signals in the ASCII code.) It is not necessary to add signal lines, but processing of the control codes takes time, so the communication speed is slower than with a method implemented in hardware. Also, when binary data is transmitted, it is necessary to implement processing to distinguish between the control codes and the data.

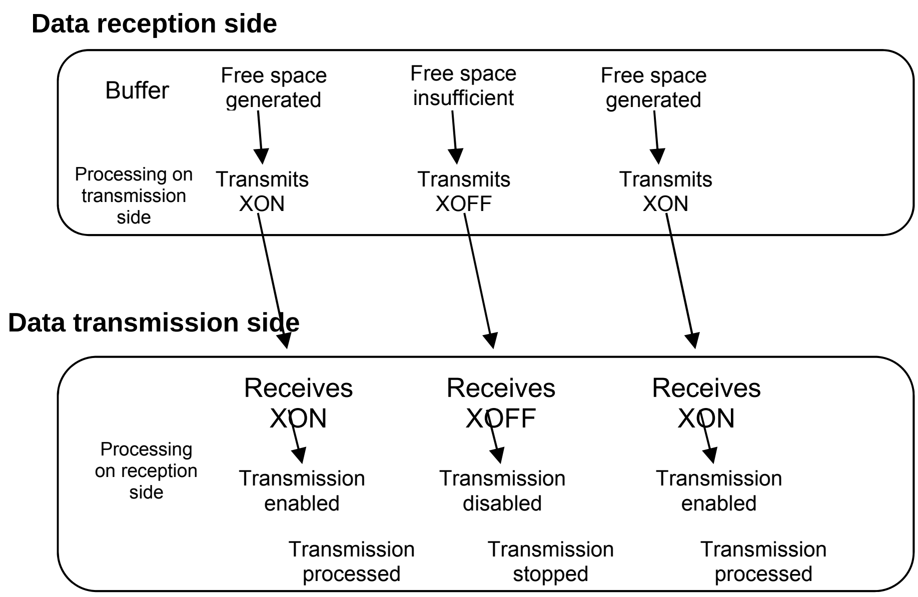


Figure 1.2 Operation of Handshake Communication Using Software

1.2 Ring Buffer Structure

Figure 1.3 shows the structure of the ring buffer. This sample program has a 16-byte ring buffer, and when the free space in the buffer is three bytes or less, the data reception side transmits XOFF (transmission disabled) to the data transmission side, thereby preventing the loss of received data. The data reception side reads the reception buffer, and when the number of bytes of data stored in the buffer is three bytes or less, the data reception side transmits XON (transmission enabled) to the data transmission side to resume communication. In order to perform stable handshake communication using software, in consideration of the time required to transmit the control codes, the data reception side transmits XOFF (transmission disabled) to the data transmission side at the point when the free space in the buffer is three bytes.

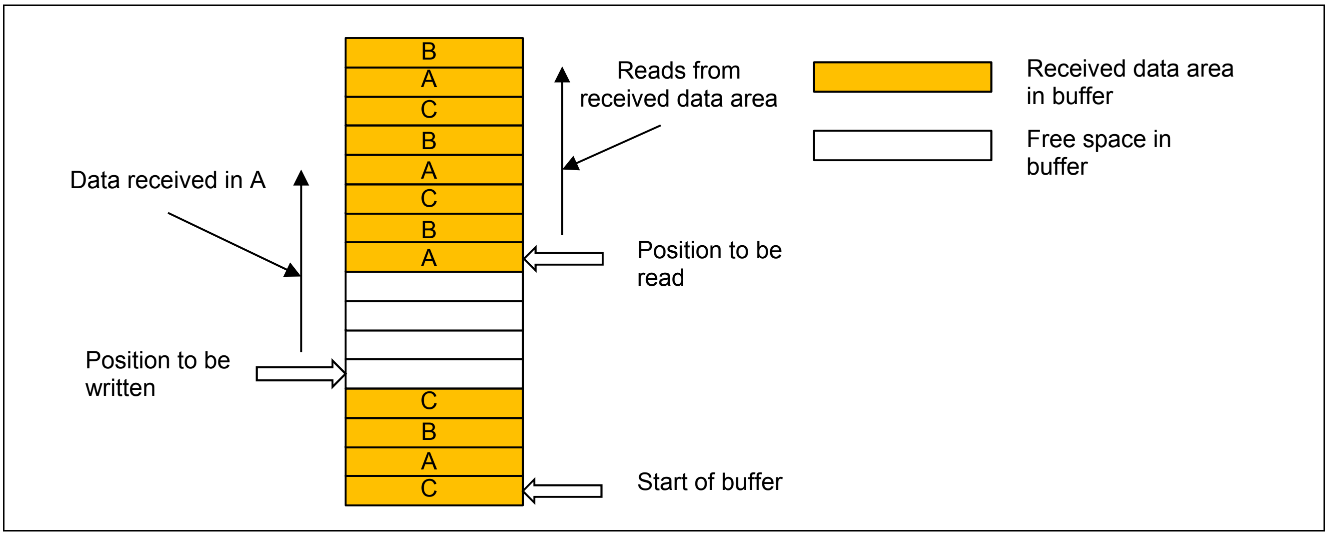


Figure 1.3 Ring Buffer Structure

1.3 Baud Rate Correction

In the sample program included in this application note, it is possible to change whether baud rate correction is enabled or disabled by using a constant definition. 55H is transmitted from the data transmission side, and the data reception side performs pulse width measurement. The baud rate is calculated and set from the measured value. The range of the baud rate value that can be corrected is from 1200 bps to 312500 bps. (If the baud rate exceeds 312500 bps, it is impossible to measure the baud rate because the INTTM01 interrupt processing will not be completed before the next INTTM01 interrupt is generated.) For details on the baud rate correction, refer to the related application note (Serial Array Unit (Baud Rate Correction) CC-RL (R01AN3083EJ)). Switching enable/disable of baud rate correction and the initial baud rate are set in an include file.

(1) Switching enable/disable of baud rate correction

Figure 1.4, if the constant CFLAGBRC in SETTING.inc is set to CBRCENABLE then baud rate correction is enabled, and if it is set to CBRCDISABLE then baud rate correction is disabled. By default, baud rate correction is enabled.

(2) Setting of Initial baud rate
As shown in Figure 1.4, the initial baud rate is set by setting the value of SET control instruction that corresponds to the baud rate to 1. The default baud rate is 9600 bps.

Figure 1.4 shows a setting example of initial baud rate and correction operation.

In this example, the baud rate correction is enabled and the initial baud rate is 9600 bps.

```
CBRCENABLE      .EQU    0x01
CBRCDISABLE     .EQU    0x00

;**********************************************************************
;       baud rate correction
;**********************************************************************
CFLAGBRC        .EQU    CBRCENABLE
;CFLAGBRC       .EQU    CBRCDISABLE

;**********************************************************************
;       baud rate select
;**********************************************************************
BAUDRATE1200    .SET 0  ; 1200bps
BAUDRATE9600    .SET 1  ; 9600bps
BAUDRATE38400   .SET 0  ; 38400bps
BAUDRATE115200  .SET 0  ; 115200bps
BAUDRATE312500  .SET 0  ; 312500bps
```

Figure 1.4 Setting Example of Initial Baud Rate and Correction Operation

Figure 1.5 shows the state transition diagram when baud rate correction is disabled.

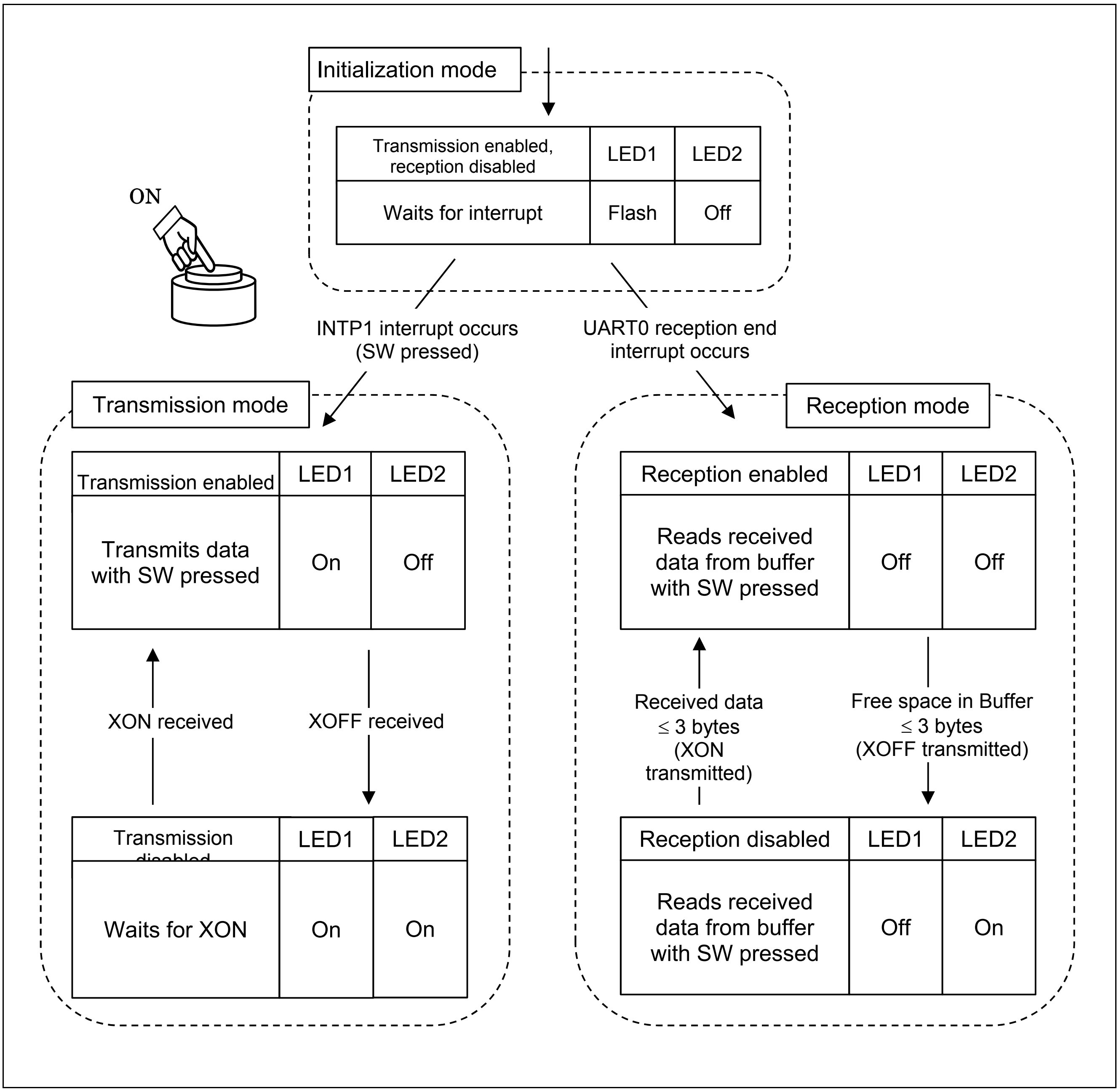


Figure 1.5 State Transition Diagram (Baud Rate Correction Disabled)

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 Check Conditions

Item	Description
Microcontroller used	RL78/G10 (R5F10Y47)
Operating frequency	High-speed on-chip oscillator (HOCO) clock: 10 MHz
Operation voltage	5.0V (Operation is possible over a voltage range of 2.9V to 5.5V.) SPOR setting: Rising edge: 2.90 V Falling edge: 2.84 V
Integrated development environment	CS+ for CC V3.02.00 from Renesas Electronics Corp.
Assembler	CC-RL V1.02.00 from Renesas Electronics Corp.
Board to be used	RL78/G10 target board (RTE510Y470TGB00000R)
Component for expanding circuit	Breadboard, LED, resistor, and tact switch

3. Related Application Notes

The application notes related to this application note are listed below for reference.

RL78/G10 Initialization CC-RL (R01AN2668EJ) Application Note
RL78/G10 Serial Array Unit (UART Communication) CC-RL (R01AN2918EJ) Application Note

RL78/G10 Serial Array Unit (Baud Rate Correction) CC-RL(R01AN3083EJ) Application Note

4. Description of Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used in this application note.

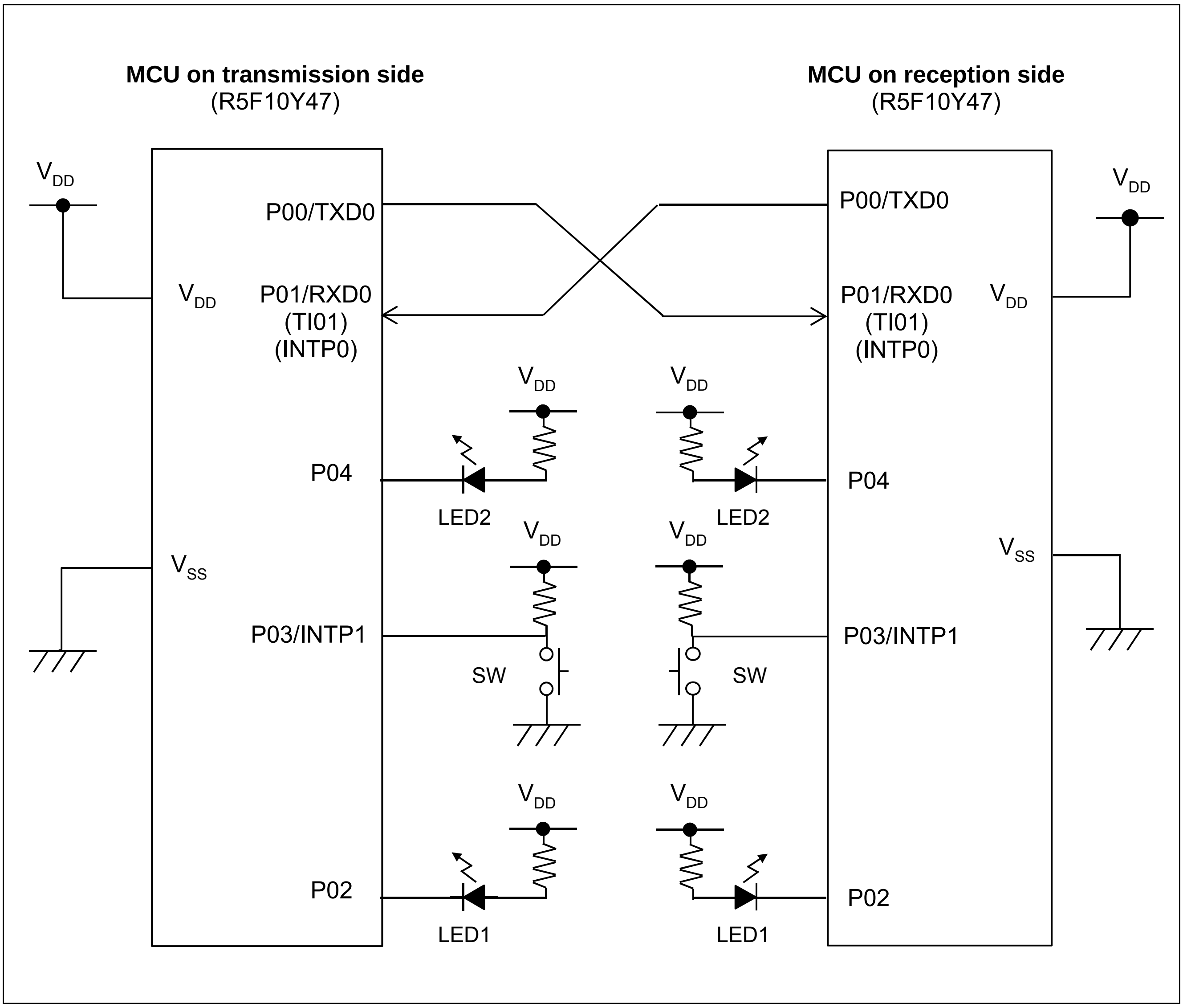


Figure 4.1 Hardware Configuration Example

Cautions:
1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide a proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to V_{DD} or V_{SS} via a resistor).
2. V_{DD} must be held at not lower than the reset release voltage (V_{SPOR}) that is specified as SPOR.

4.2 List of Pins to be Used

Table 4.1 lists the pins to be used and their functions.

Table 4.1 Pins to be Used and their Functions

Pin Name	I/O	Function
P00/TXD0	Output	UART transmission
P01/RXD0	Input	Can also be used for one of the following functions by setting the ISC register. - UART reception - TI01 input - INTP0 input
P04	Output	LED control
P03/(INTP1)	Input	Switch input
P02	Output	LED control

5. Description of Software

5.1 Operation Overview

The sample code included in this application note performs a handshake communication using the control signals shown in Table 5.1.

Table 5.1 Control Signals Used for Handshake Communication

Transmission Data	Control Function
XON (11H)	Enables transmission.
XOFF (13H)	Disables transmission.

In the sample program included in this application note, UART communication is performed between two evaluation boards. One of them operates as the data transmission side and the other as the data reception side, so data flow is unidirectional from the data transmission side to the data reception side. However, since it is necessary for the data reception side to send “XON (transmission enabled) / XOFF (transmission disabled)” to the data transmission side, these control codes are transmitted from the data reception side to the data transmission side.

For the software operation, four operating modes are defined as the internal state, and processing is performed based on the respective operating modes. エラー! ブックマークが自己参照を行っています。 lists the operating modes. The operating modes are distinguished by LED1. After reset is released, both two evaluation boards operate in initialization mode. In initialization mode LED1 flashes, and in this state the board whose switch is pressed becomes the data transmission side, and the other board becomes the data reception side.

Table 1.1 lists the operating modes. After reset is released, both boards operate in initialization mode. In initialization mode LED1 flashes, and in this state the board whose switch is pressed becomes the data transmission side and the other one becomes the data reception side.

The specific processing flow is as follows.

(1) Perform the initial setting of functions to be used.
<Setting conditions>

- Initial setting of SAU (Perform the UART0 setting.)
- Initial setting of TAU (Set pulse interval measurement mode for TAU01.)
- Set the falling edge detection for INTP0/INTP1 interrupts.

(2) While flashing LED1 with an interval of approximately 500 ms, wait for the generation of either the INTP0 (UART communication start) or INTP1 (switch pressed) interrupt.

(3) If the INTP0 interrupt occurs then this board will become the data reception side, so start the TAU01 operation to measure the baud rate.
If the INTP1 interrupt occurs then this board will become the data transmission side, so transmit the correction code 55H through the UART.

(4) The data reception side measures the pulse interval of the correction code 55H and calculates the baud rate. After applying the calculated baud rate, it transmits XON to inform the data transmission side that the transmission is currently enabled.
The data transmission side waits for XON to be transmitted from the data reception side.

(5) The data reception side stores received data in the ring buffer, and when the switch is pressed it reads one byte of the received data in the buffer. When the free space in the buffer is three bytes or less it transmits XOFF to set the transmission disable state, and when the number of bytes of data stored in the reception buffer is three bytes or less it transmits XON to set transmission enable state.
In transmission enable state the data transmission side transmits data when the switch is pressed. In transmission disable state it will not start a transmission operation even if the switch is pressed. If XOFF is received during a transmission operation, then the transmission end interrupt is prohibited so that the next transmission data is not set.

5.2 List of Option Byte Settings

Table 5.2 shows the settings of the option bytes.

Table 5.2 Option Byte Settings

Address	Setting Value	Description
000C0H	11101110B	Disables the watchdog timer. (Stops counting after release from the reset state.)
000C1H	11110111B	SPOR detection voltage Rising edge: 2.90 V Falling edge: 2.84 V
000C2H	11111010B	HOCO: 10 MHz
000C3H	10000101B	Enables the on-chip debugger.

5.3 List of Constants

Table 5.3 lists the constants that are used in the sample program.

Table 5.3 Constants Used in Sample Program

Constant Name	Setting Value	Description
CCODEXON	**11H**[Note1]	Transmission enable code: XON
CCODEXOFF	**13H**[Note1]	Transmission disable code: XOFF
CCODEBRC	**55H**	Value to be transmitted for baud rate correction: 55H
CMODETX	**'T'**	Operating mode: Transmission mode
CMODERX	**'R'**	Operating mode: Reception mode
CMODEBRC	**'B'**	Operating mode: Correction mode
CMODEINI	**'I'**	Operating mode: Initialization mode
CSTATEXON	**01H**	Communication state: Transmission enabled (XON)
CSTATEXOFF	**00H**	Communication state: Transmission disabled (XOFF)
CFLAGBRC	CBRCENABLE[Note2]	Whether baud rate correction is enabled/disabled
CBRCENABLE	**01H**	Baud rate correction enabled
CBRCDISABLE	**00H**	Baud rate correction disabled
CSENDDATASIZE	**03H**	Number of bytes of transmission data
CSENDDATA	**"ABC"** [Note3]	Transmission data
SW	**P0.3**	Switch
LED1	**P0.2**	LED1
LED2	**P0.4**	LED2

Notes: 1. The setting value can be arbitrarily changed, but CCODEXON cannot be the same value as CCODEXOFF.

2. Set CBRCENABLE or CBRCDISABLE in the include file to switch whether baud rate correction is enabled or disabled. The default value is CBRCENABLE (01H) and baud rate correction is enabled.

3. Data transmitted using the SUART0SENDDATA function. Although it can be set to an arbitrary value, always set the number of bytes to the same value as specified by the constant CDATASIZE.

5.4 List of Variables

Table 5.4 lists the global variables that are used in the sample program.

Table 5.4 Global Variables

Type	Variable Name	Content	Function Used
8 bits	COUNT	Used for 500 ms wait processing	main
8 bits	UASTT	Transmission status flag (counter)	SUART0SENDDATA, IINTST0
16 bits	SENDPNT	Pointer to UART transmission data	SUART0SENDDATA
8 bits × 16 arrays	DATABUF	Ring buffer for storing UART received data	SPUTDATA, SGETDATA
8 bits	SETPNT	Pointer for storing data in ring buffer	SINIMAIN, SPUTDATA
8 bits	GETPNT	Pointer for reading data from ring buffer	SINIMAIN, SGETDATA
8 bits	DATACNT	Number of bytes of data in ring buffer	SINIMAIN, SPUTDATA, SPUTDATASENDXOFF, SGETDATA, SGETDATASENDXON, IINTSR0, IINTSRE0
8 bits	MODE	Indicates operating mode (initialization, correction, transmission, or reception).	Main, SINIMAIN, IINTSR0, IINTTM01, IINTP0, IINTP1
8 bits	CMCSTATE	Indicates communication state (XON or XOFF).	Main, SINIMAIN, SUART0SENDDATA, SPUTDATASENDXOFF, SGETDATASENDXON, IINTP0, IINTP1
8 bits	FLAGSWON	Indicates that switch is ON.	Main, SINIMAIN, IINTP1
8 bits	FLAGSENDING	Indicates that data is being transmitted.	SUART0SENDDATA, IINTST0
8 bits	SPSDATA	Stores the setting value for SPS0 register.	Main, IINTTM01
8 bits	DIVDATA	Stores the setting value for SDR0nH register.	Main, IINTTM01
16 bits	CAPTUREL	Variable for accumulating capture values (lower 16 bits)	IINTTM01, IINTP0
8 bits	CAPTUREH	Variable for accumulating capture values (upper 8 bits)	IINTTM01, IINTP0
8 bits	LPCOUNT	Used for counting the number of captures	Main, SINIMAIN, IINTTM01

5.5 List of Functions (Subroutines)

Table 5.5 lists the functions (subroutines) that are used in the sample program.

Table 5.5 List of Functions (Subroutines)

Function Name	Outline
SINIMAIN	Initialization processing using the main function
SWAITMS	Wait processing (millisecond units)
SSTARTUART0TX	UART0 transmission enabling processing
SSTARTUART0RX	UART0 reception enabling processing
SSTOPUART0	UART0 operation stop processing
SUART0SENDDATA	UART0 data transmission processing
SUART0SENDCODE	UART0 control code transmission processing
SPUTDATA	Ring buffer store processing
SPUTDATASENDXOFF	Ring buffer store and XOFF transmission processing
SGETDATA	Ring buffer read processing
SGETDATASENDXON	Ring buffer read and XON transmission processing
IINTSR0	UART0 reception end interrupt processing
IINTSRE0	UART0 reception error interrupt processing
IINTST0	UART0 transmission end interrupt processing
IINTTM01	INTTM01 interrupt processing
IINTP0	INTP0 interrupt processing
IINTP1	INTP1 interrupt processing

5.6 Function (Subroutine) Specifications

This section describes the specifications for the functions (subroutines) that are used in the sample program.

[Function name] SINIMAIN

Outline	Initialization processing using the main function
Explanation	Initializes the variables and peripherals.
Argument	None
Return value	None
Remarks	None

[Function name] SWAITMS

Outline	Wait processing (millisecond units)
Explanation	Performs wait processing for the time specified by the AX register (millisecond units).
Argument	AX :[Time to be waited (ms)]
Return value	None
Remarks	Since wait is implemented by using loops, the wait time is not precise.

[Function name] SSTARTUART0TX

Outline	UART0 transmission enabling processing
Explanation	Starts operation of channel 0 of serial array unit 0 to make the system enter a transmission wait state. Disables transmission end interrupt.
Argument	None
Return value	None
Remarks	None

[Function name] SSTARTUART0RX

Outline	UART0 reception enabling processing
Explanation	Starts operation of channel 1 of serial array unit 0 to make the system enter a reception wait state. Enables reception end interrupt and reception error interrupt.
Argument	None
Return value	None
Remarks	None

[Function name] SSTOPUART0

Outline	UART0 operation stop processing
Explanation	Stops operation of UART0.
Argument	None
Return value	None
Remarks	This function is used to reconfigure the baud rate.

[Function name] SUART0SENDDATA

Outline	UART0 data transmission processing	
Explanation	Transmits data from UART0. Transmission processing is performed on an interrupt basis.	
Argument	HL	:[Address for storing transmission data]
Return value	None	
Remarks	None	

[Function name] SUART0SENDCODE

Outline	UART0 control code transmission processing	
Explanation	Transmits the control codes from UART0.	
Argument	A	: [Transmission data (control codes)]
Return value	None	
Remarks	None	

[Function name] SPUTDATA

Outline	Ring buffer store processing	
Explanation	Stores data if there is free space in the ring buffer.	
Argument	A	: [Data to be stored]
Return value	CY	: [1: Buffer full, 0: Store operation completed]
Remarks	None	

[Function name] SPUTDATASENDXOFF

Outline	Ring buffer store and XOFF transmission processing.	
Explanation	Stores data if there is free space in the ring buffer. As a result of storing, if free space in the buffer is three bytes or less then transmit XOFF.	
Argument	A	: [Data to be stored]
Return value	CY	: [1: Buffer full, 0: Store operation completed]
Remarks	None	

[Function name] SGETDATA

Outline	Ring buffer read processing	
Explanation	Reads the oldest data from the ring buffer if there is any data in the buffer.	
Argument	None	
Return value	CY	: [1: No data, 0: Reading completed]
	A	: [Read-out data]
Remarks	None	

[Function name] SGETDATASENDXON

Outline	Ring buffer read XON transmission processing	
Explanation	Reads the oldest data from the ring buffer if there is any data stored in the buffer. As a result of reading, if the number of bytes of data remaining in the buffer is three bytes or less then transmit XON.	
Argument	なし	
Return value	CY	: [1: No data, 0: Reading completed]
	A	: [Read-out data]
Remarks	None	

[Function name] IINTSR0

Outline	UART0 reception end interrupt processing
Explanation	Performs processing based on the operating modes. Initialization mode: Changes the mode setting to reception mode and performs the same processing as in reception mode. Reception mode: Stores the received data in the ring buffer. Other modes: Processes the received data as the control codes.
Argument	None
Return value	None
Remarks	None

[Function name] IINTSRE0

Outline	UART0 reception error interrupt processing
Explanation	Clears the reception end interrupt request. In reception mode, if there is free space in the ring buffer then store the UART0 reception error status (the lower three bits of the SSR01 register) in the ring buffer. Does nothing if the ring buffer is full.
Argument	None
Return value	None
Remarks	Bit 0 of the SSR01 register is the overrun error detection flag, bit 1 is the parity error detection flag, and bit 2 is the framing error detection flag. (In this sample program the UART operation is set to “no parity”, so parity error will not be generated.

[Function name] IINTST0

Outline	UART0 transmission end interrupt processing
Explanation	Decrements the number of bytes of transmission data (UASTT) by 1. Transmits the next data if there is any data remaining in the buffer. Disables transmission end interrupts if there is no data remaining.
Argument	None
Return value	None (UASTT indicates the number of remaining data bytes. 0 indicates that the transmission is completed.
Remarks	None

[Function name] IINTTM01

Outline	INTTM01 interrupt processing
Explanation	Performs processing according to the number of times an interrupt has occurred. First time: Counts down the number of captures (LPCOUNT). Second to fourth times: Adds the capture value to the variable for accumulation. Fifth time: Adds the capture value to the variable for accumulation and stops TM01. From the accumulated value, calculates the values to set in the SPS0 and SDR0nH registers, and stores them into variables.
Argument	None
Return value	None
Remarks	Stores the value to set in the SPS0 register into the SPSDATA variable, and the value to set in the SDR0nH register into the DIVDATA variable.

[Function name] IINTP0

Outline	INTP0 interrupt processing (Generates an interrupt when the UART start bit is detected)
Explanation	Prepares for baud rate measurement and enables TM01 operation. Disables an INTP0 interrupt.
Argument	None
Return value	None
Remarks	None

[Function name] IINTP1

Outline	INTP1 interrupt processing (Generates an interrupt when the switch is pressed)
Explanation	Waits for 10 ms to avoid chattering, then checks again that the switch is pressed, and sets the switch flag. Only when this function is invoked in initialization mode, changes the operating mode to correction mode or transmission mode.
Argument	None
Return value	None
Remarks	None

5.7 Flowcharts

"RET" and "RETI" are used as the termination symbols for functions and interrupt processing, respectively.

5.7.1 Overall Flow

Figure 5.1 shows the overall flow of the sample code described in this application note.

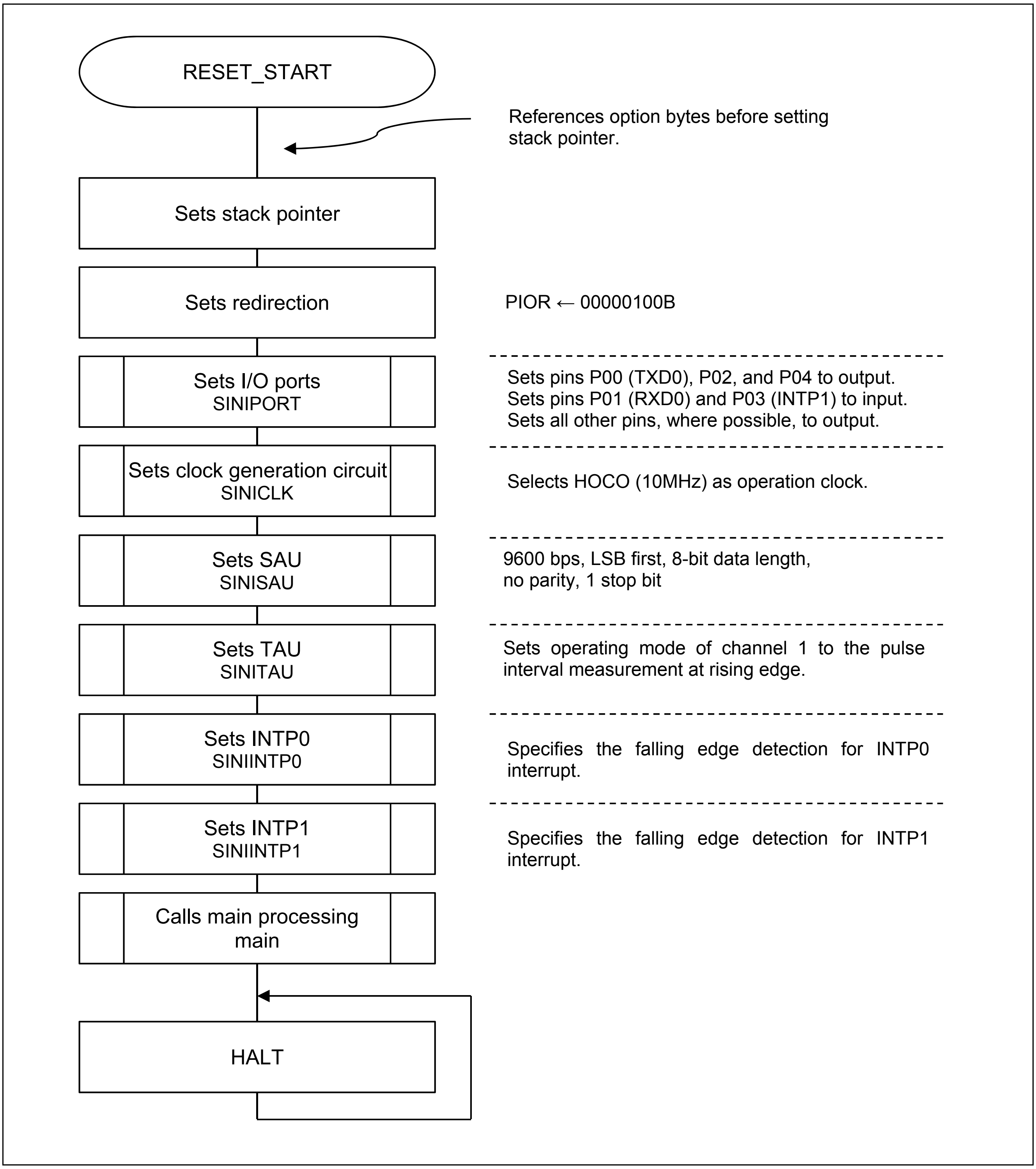


Figure 5.1 Overall Flow

5.7.2 I/O Port Settings

Figure 5.2 shows the flowchart for the I/O port settings.

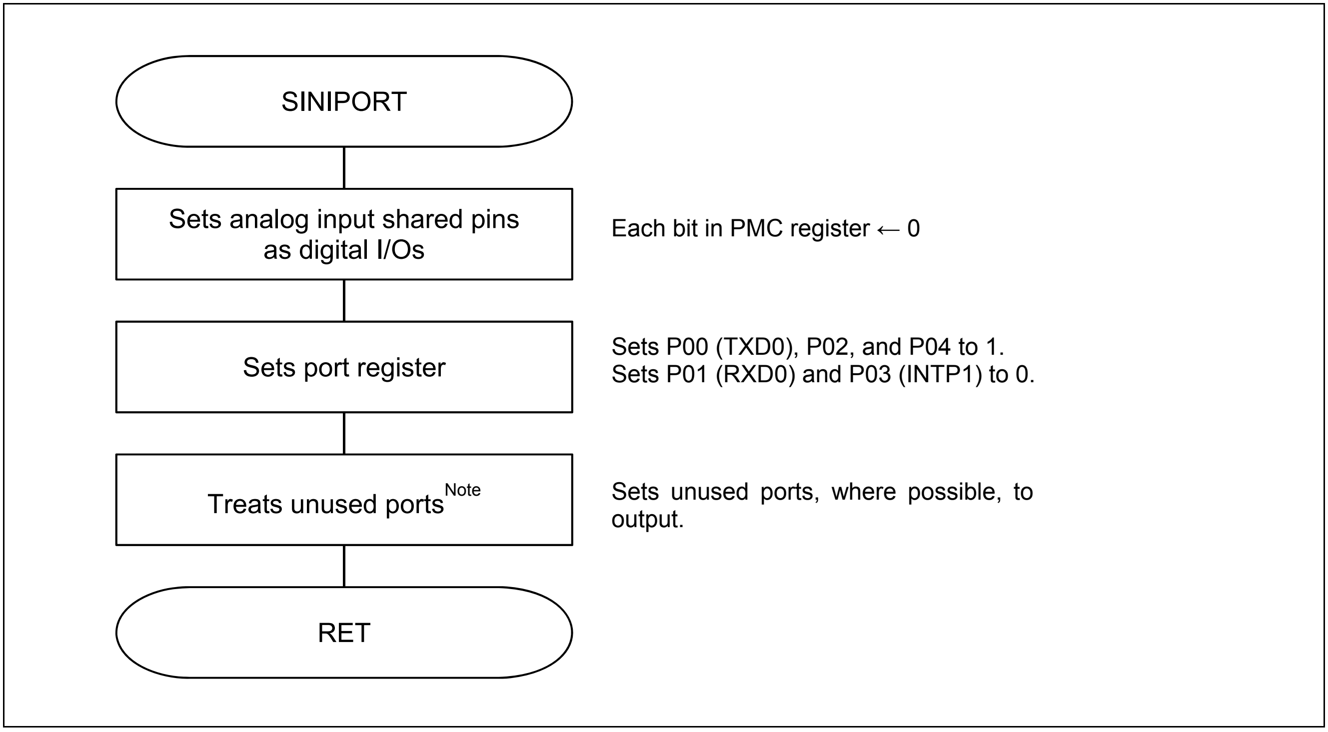


Figure 5.2 I/O Port Settings

Note: Refer to Section 2.3 “Connection of Unused Pins” in the “RL78/G10 User's Manual: Hardware (R01UG0384EJ)” for the setting of unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are met. Connect each of any unused input-only ports to V_{DD} or V_{SS} via a separate resistor.

5.7.3 Setting of Clock Generation Circuit

Figure 5.3 shows the flowchart for the clock generation circuit.

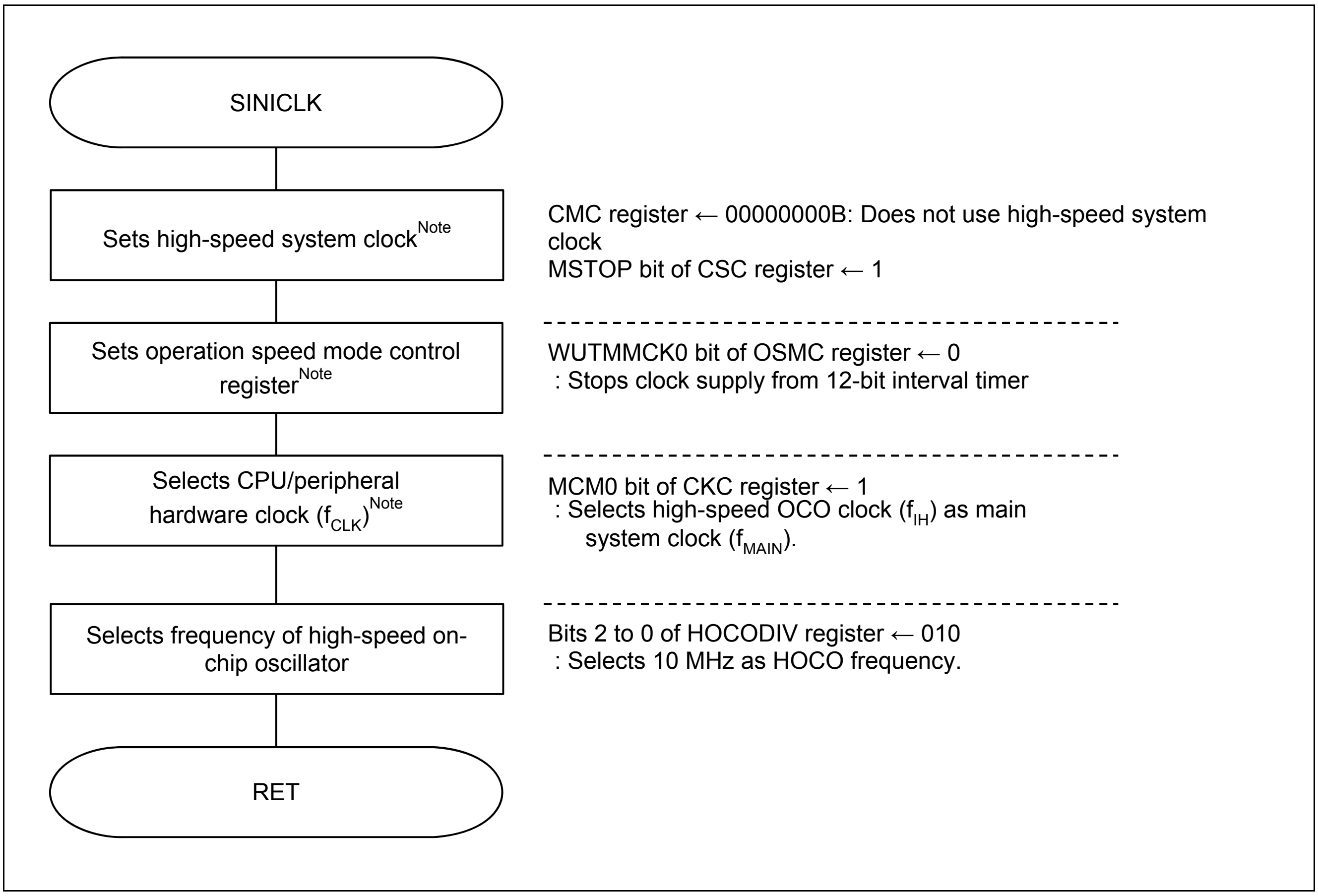


Figure 5.3 Setting of Clock Generation Circuit

Note: Set CMC, CKC, CSC, and OSMC registers only for the 16-pin version of the RL78/G10. (As for the 10-pin version, setting of these registers is unnecessary.)

Caution: Refer to Section 4.5.3 “Clock Generation Circuit Setup” in the “RL78/G10 Initialization CC-RLCPU Application Note (R01AN2668EJ)” for the setting of the CPU clock (SINICLK).

5.7.4 SAU Setting

Figure 5.4 andFigure 5.5 show the flowcharts for the SAU settings 1 of 2 and 2 of 2.

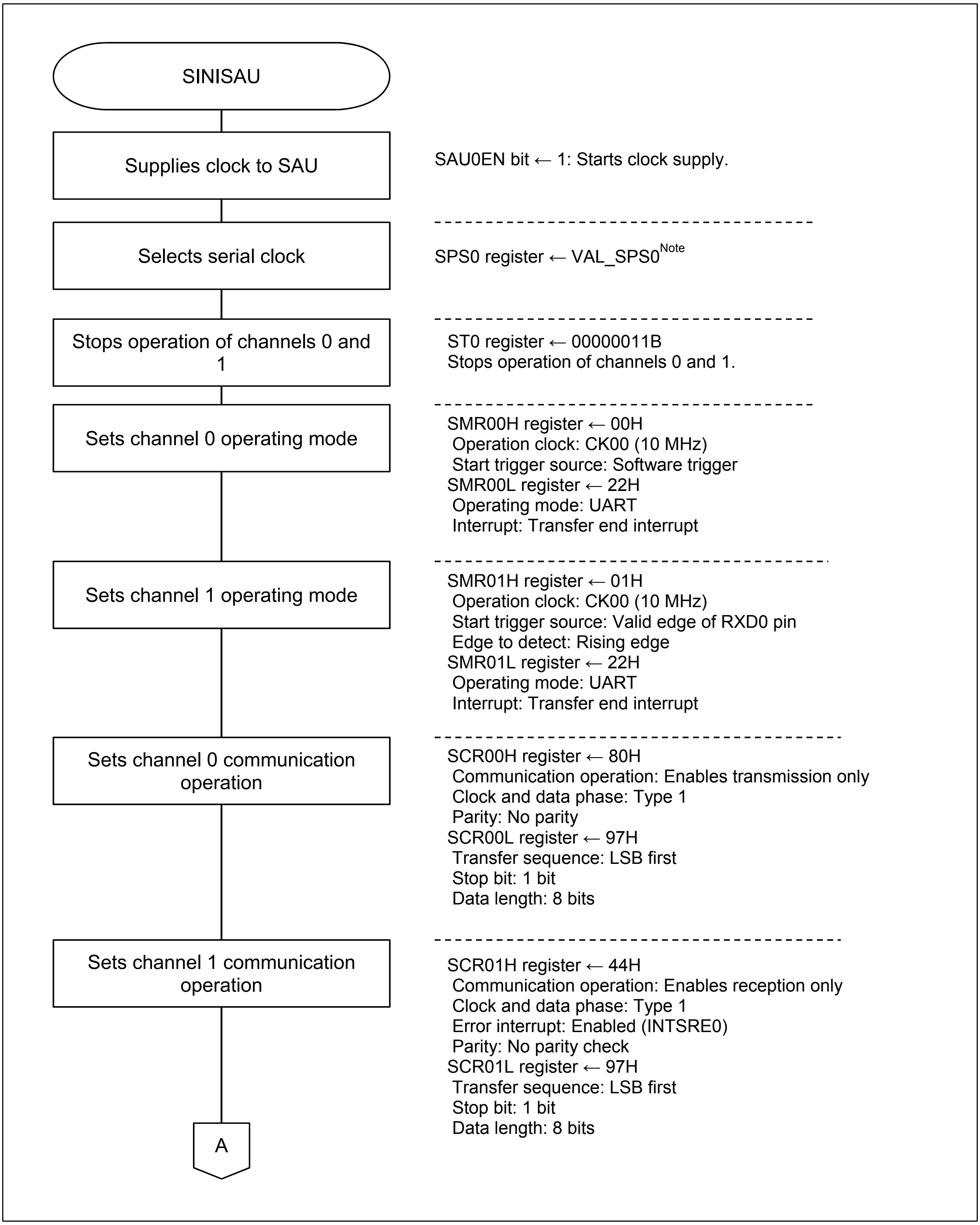


Figure 5.4 SAU Setting (1 of 2)

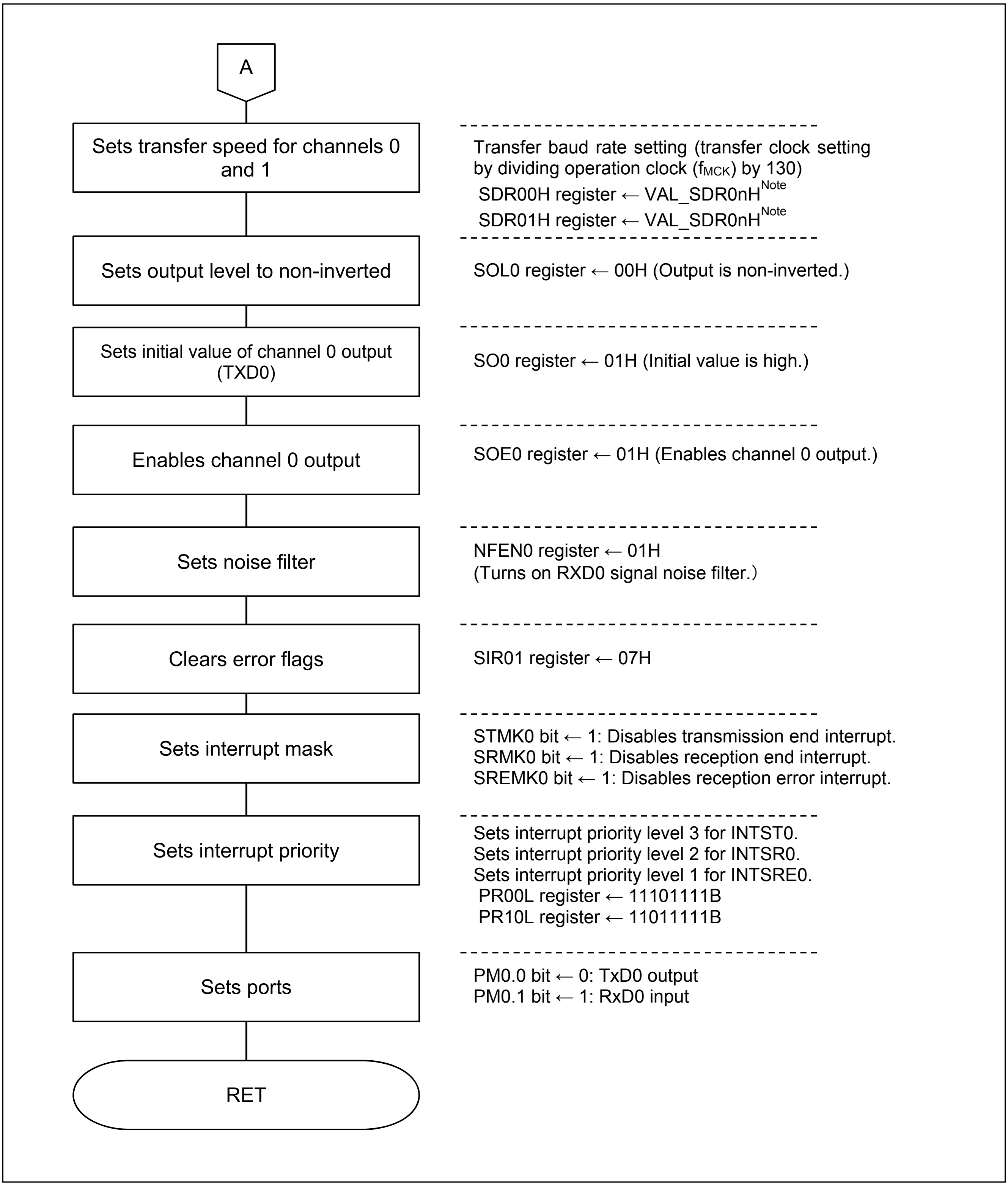


Figure 5.5 SAU Setting (2 of 2)

Note: Values of VAL_SPS0 in Figure 5.4 and VAL_SDR0nH in Figure 5.5 vary depending on the setting of the initial baud rate with SETTING.inc. For details, see Section エラー! 参照元が見つかりません。 “エラー! 参照元が見つかりません。”.

5.7.5 TAU Setting

Figure 5.6 shows the flowchart for the TAU setting.

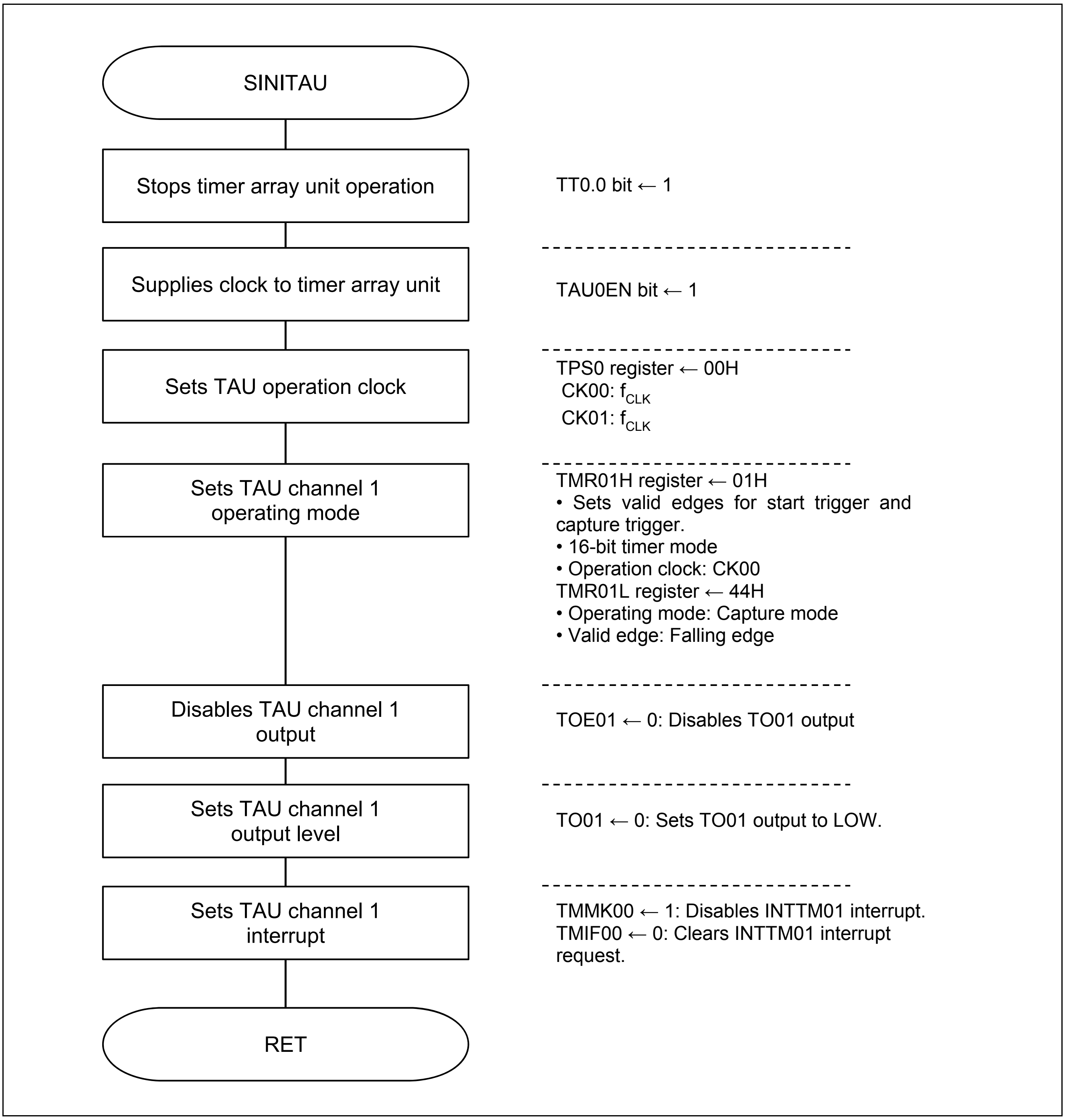


Figure 5.6 TAU Setting

5.7.6 Main Processing

Figure 5.7, Figure 5.8, andFigure 5.9 show the flowcharts for the main processing 1 of 3 to 3 of 3.

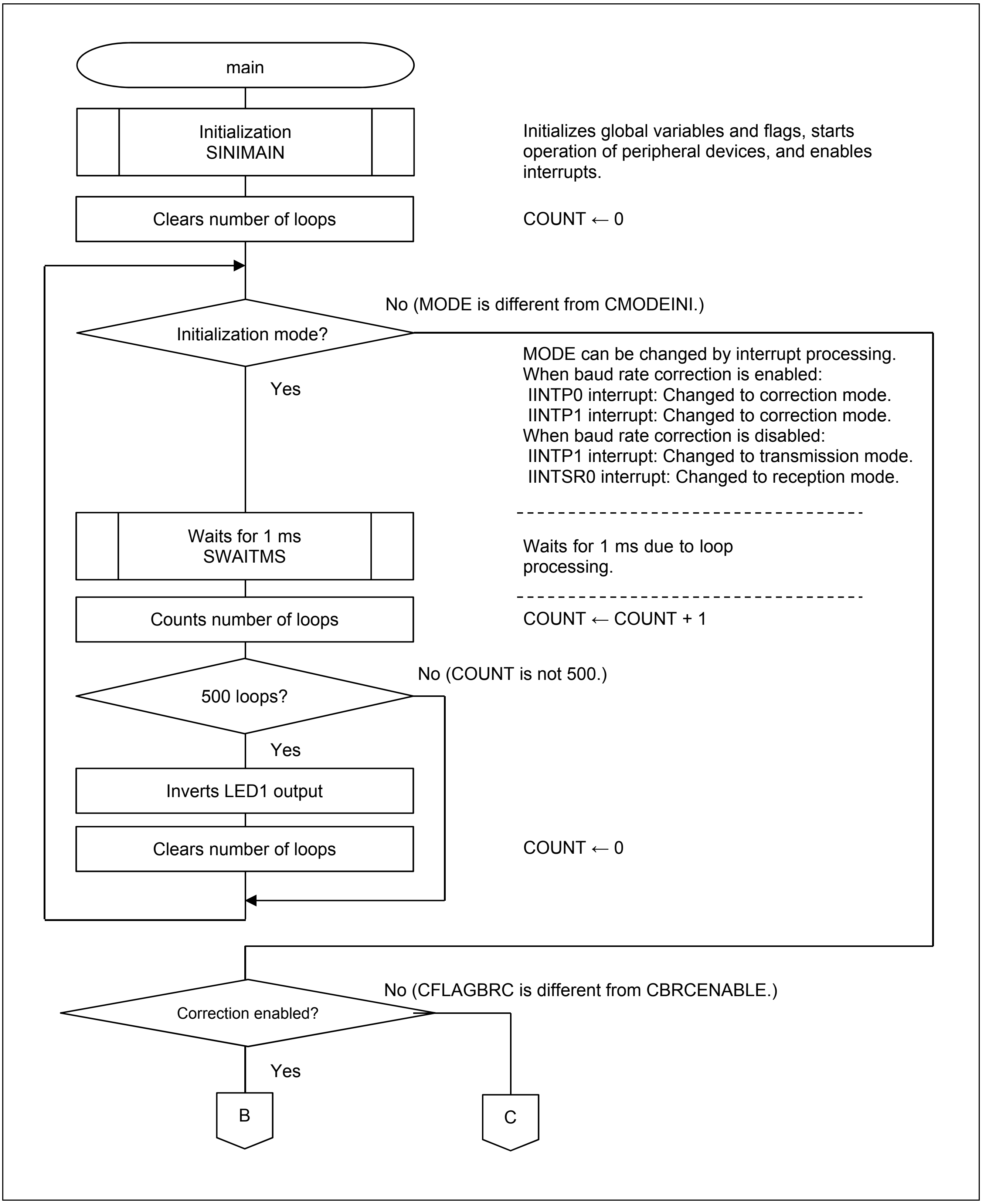


Figure 5.7 Main Processing (1 of 3)

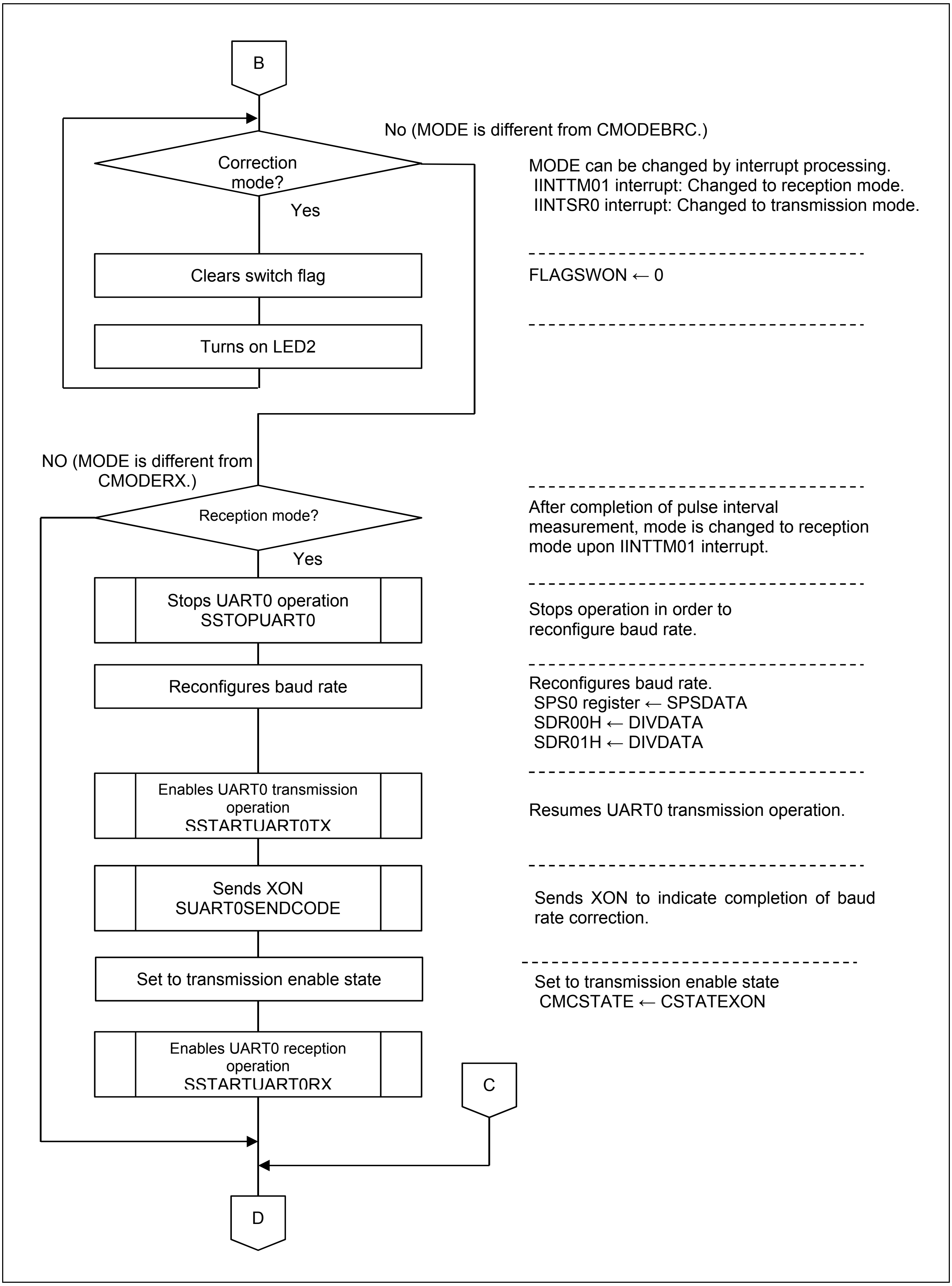


Figure 5.8 Main Processing (2 of 3)

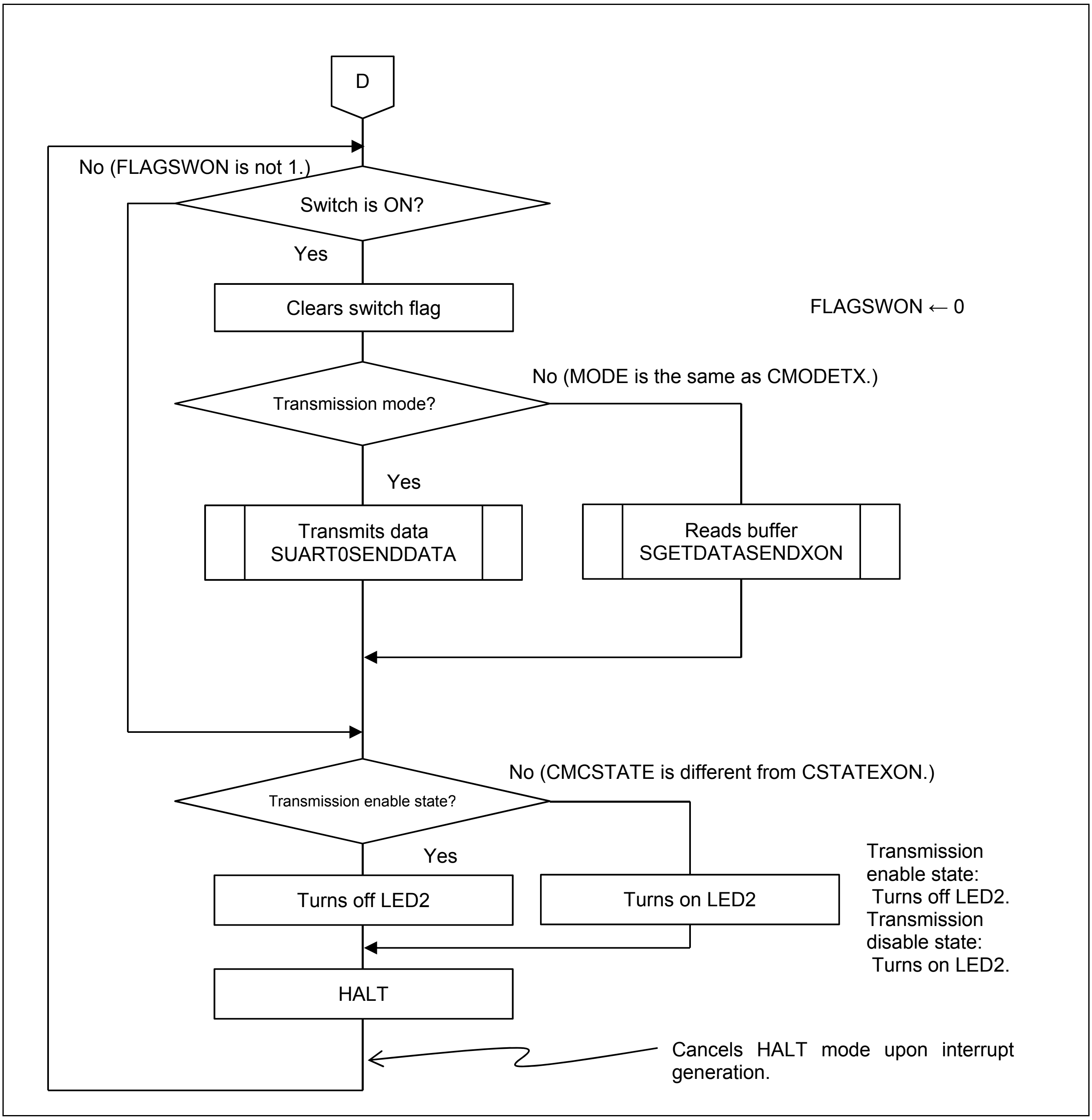


Figure 5.9 Main Processing (3 of 3)

5.7.7 Main Initialization Processing

Figure 5.10 shows the flowchart for the main initialization processing.

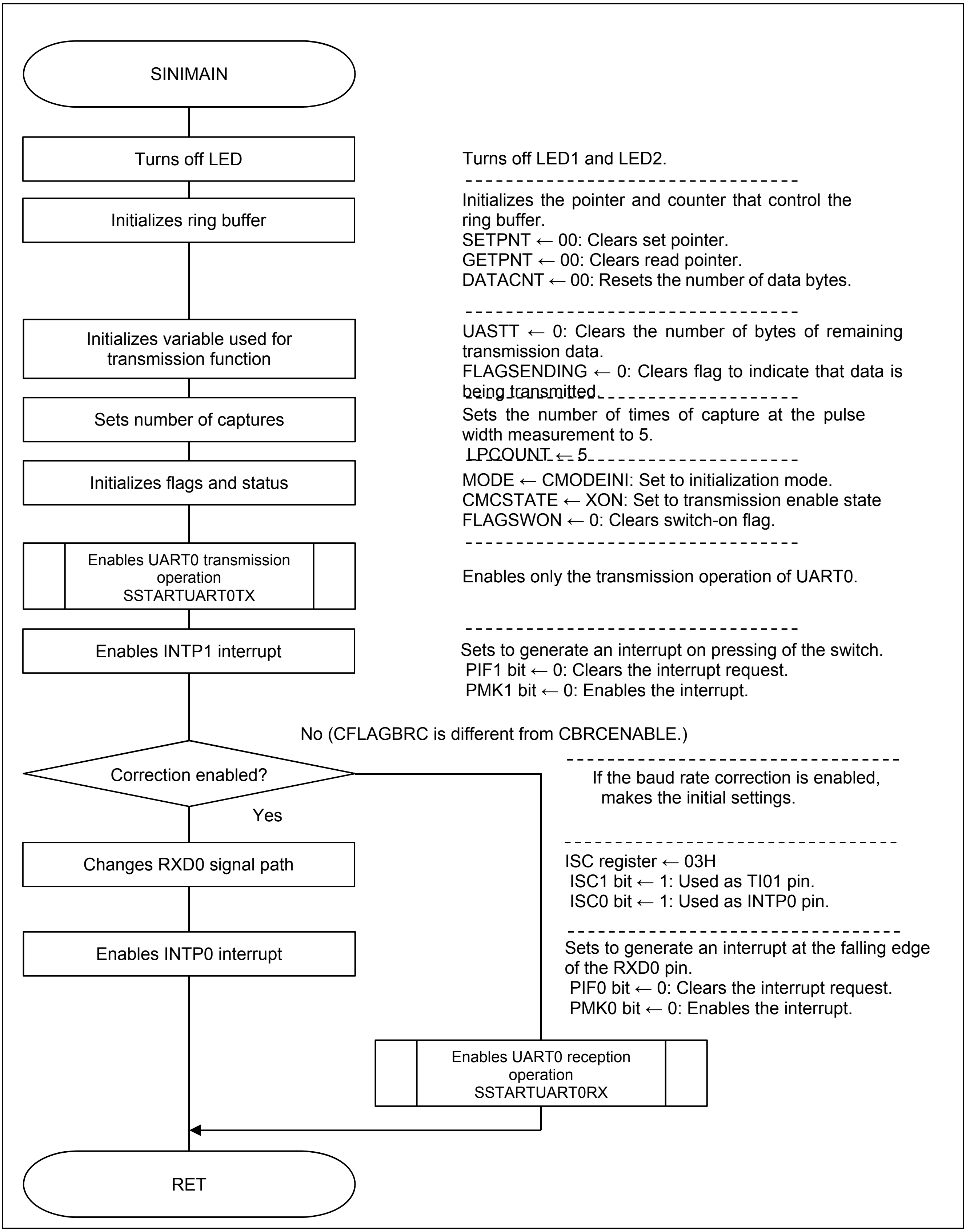


Figure 5.10 Main Initialization Processing

5.7.8 Wait Processing (millisecond units)

Figure 5.11 shows the flowchart for the wait processing (millisecond units).

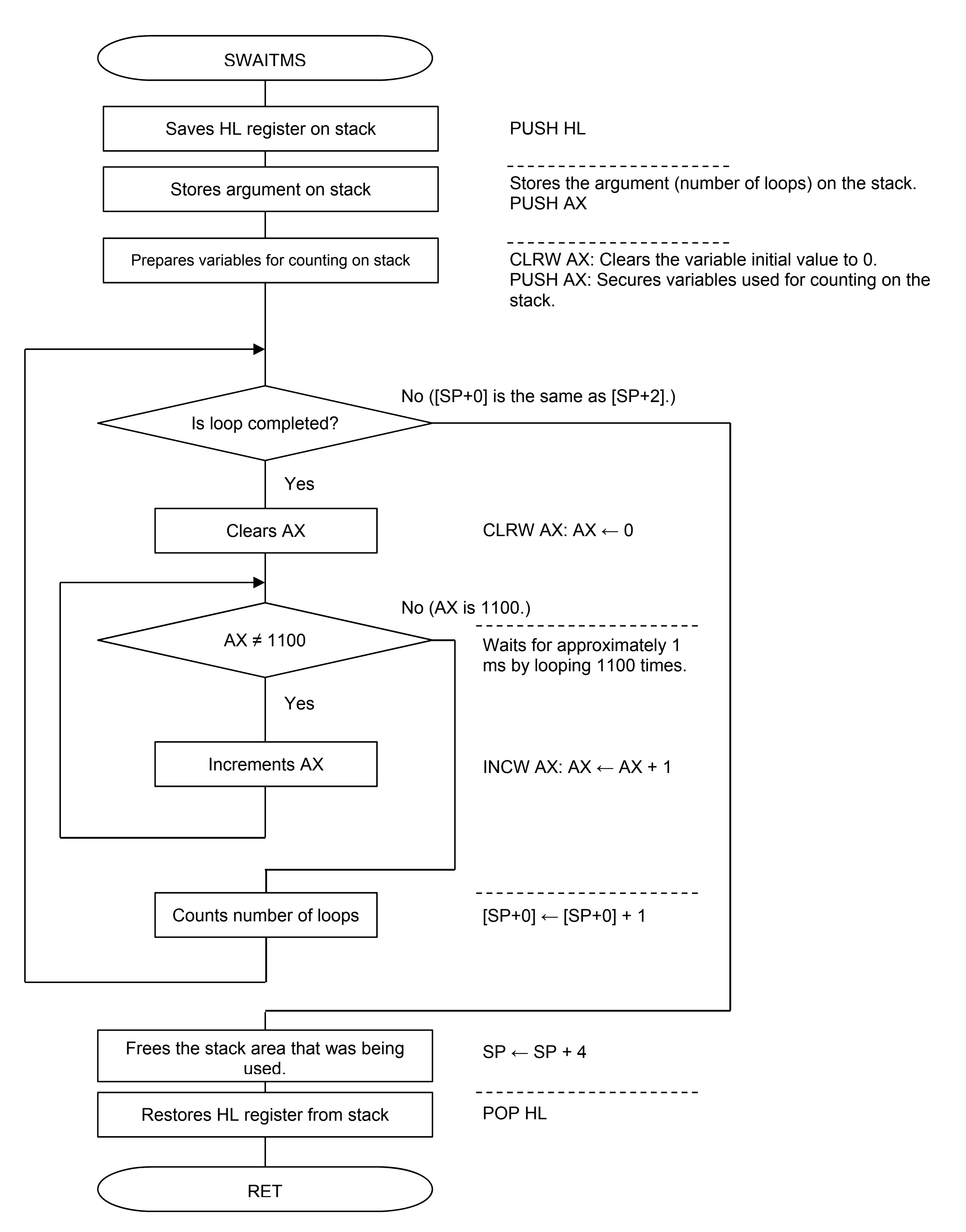


Figure 5.11 Wait Processing (millisecond units)

5.7.9 UART0 Transmission Enabling Processing

Figure 5.12 shows the flowchart for the UART0 transmission enabling processing.

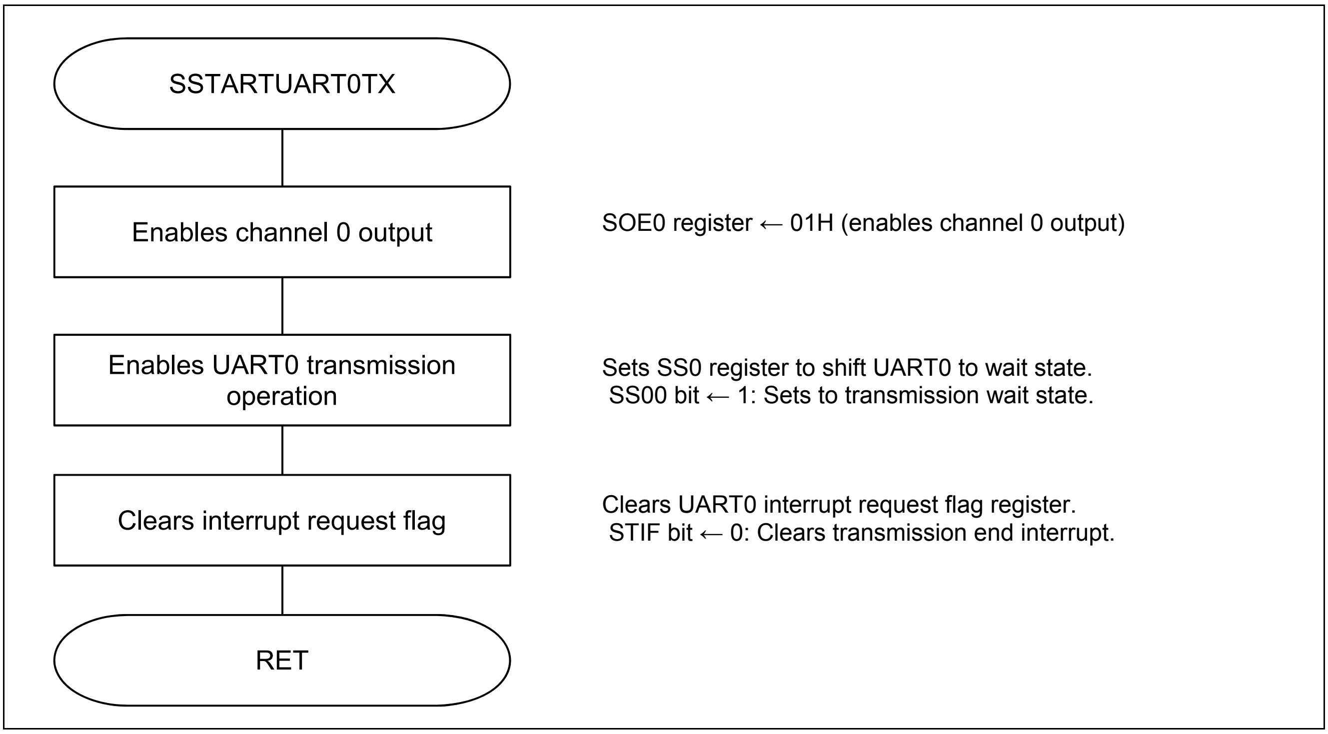


Figure 5.12 UART0 Transmission Enabling Processing

5.7.10 UART0 Reception Enabling Processing

Figure 5.13 shows the flowchart for the UART0 reception enabling processing.

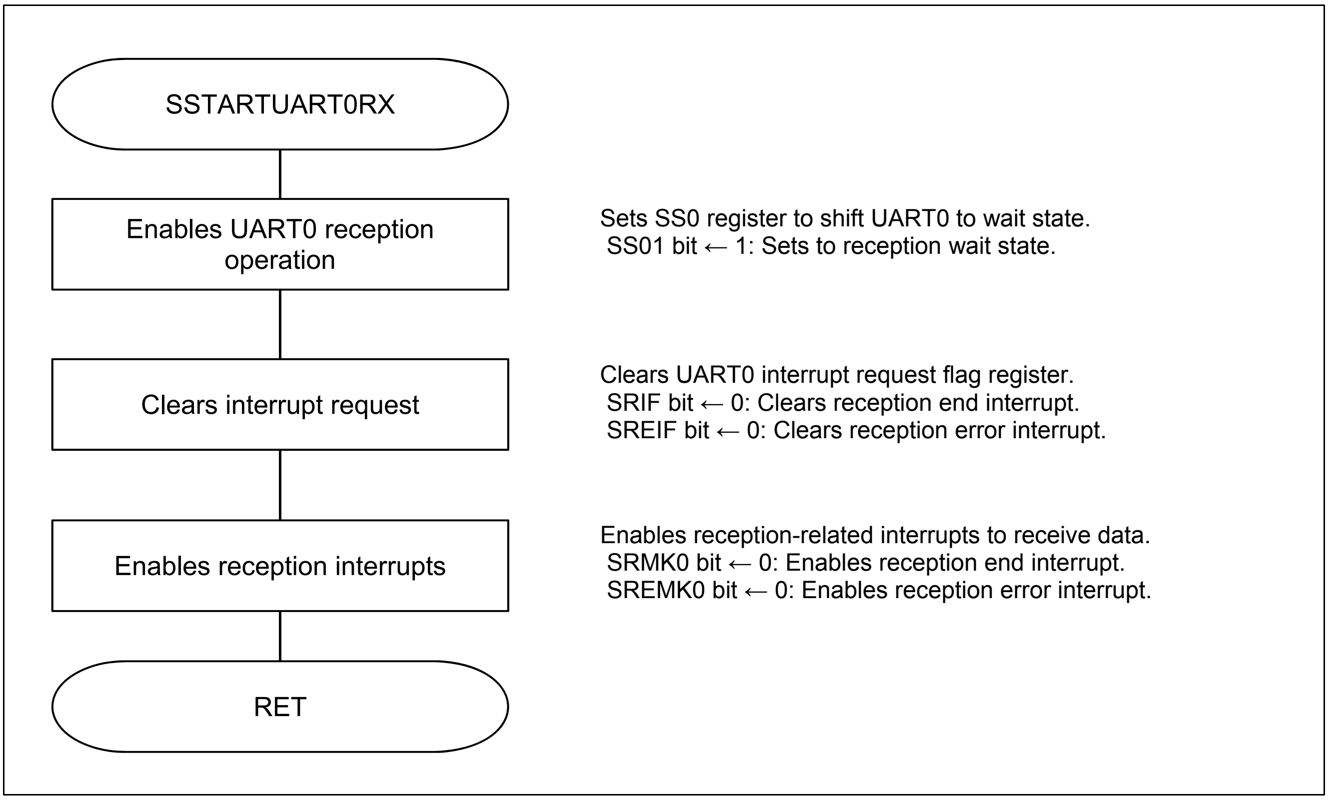


Figure 5.13 UART0 Reception Enabling Processing

5.7.11 UART0 Operation Stop Processing

Figure 5.14 shows the flowchart for the UART0 operation stop processing.

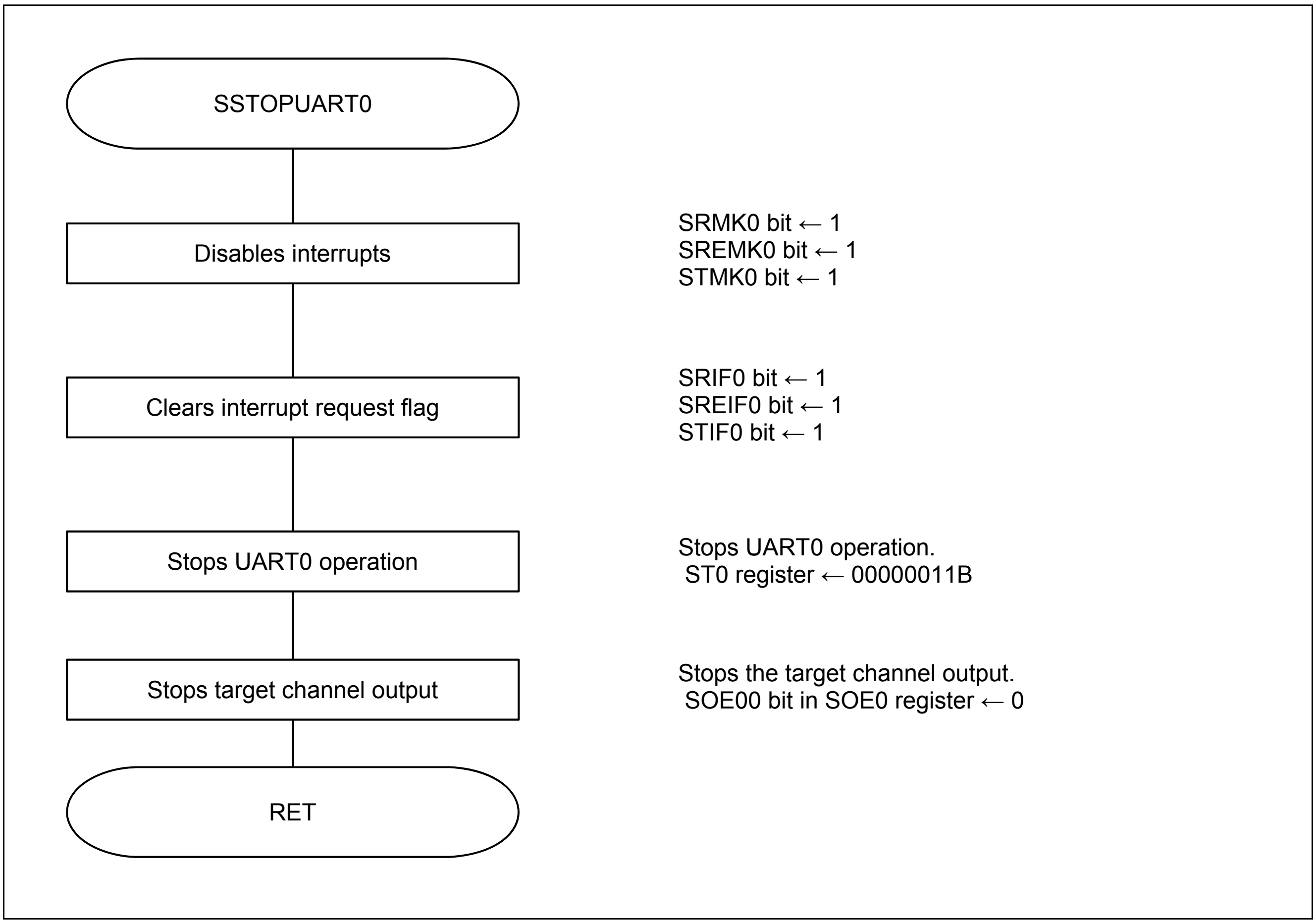


Figure 5.14 UART0 Operation Stop Processing

5.7.12 UART0 Data Transmission Processing

Figure 5.15 shows the flowchart for the UART0 data transmission processing.

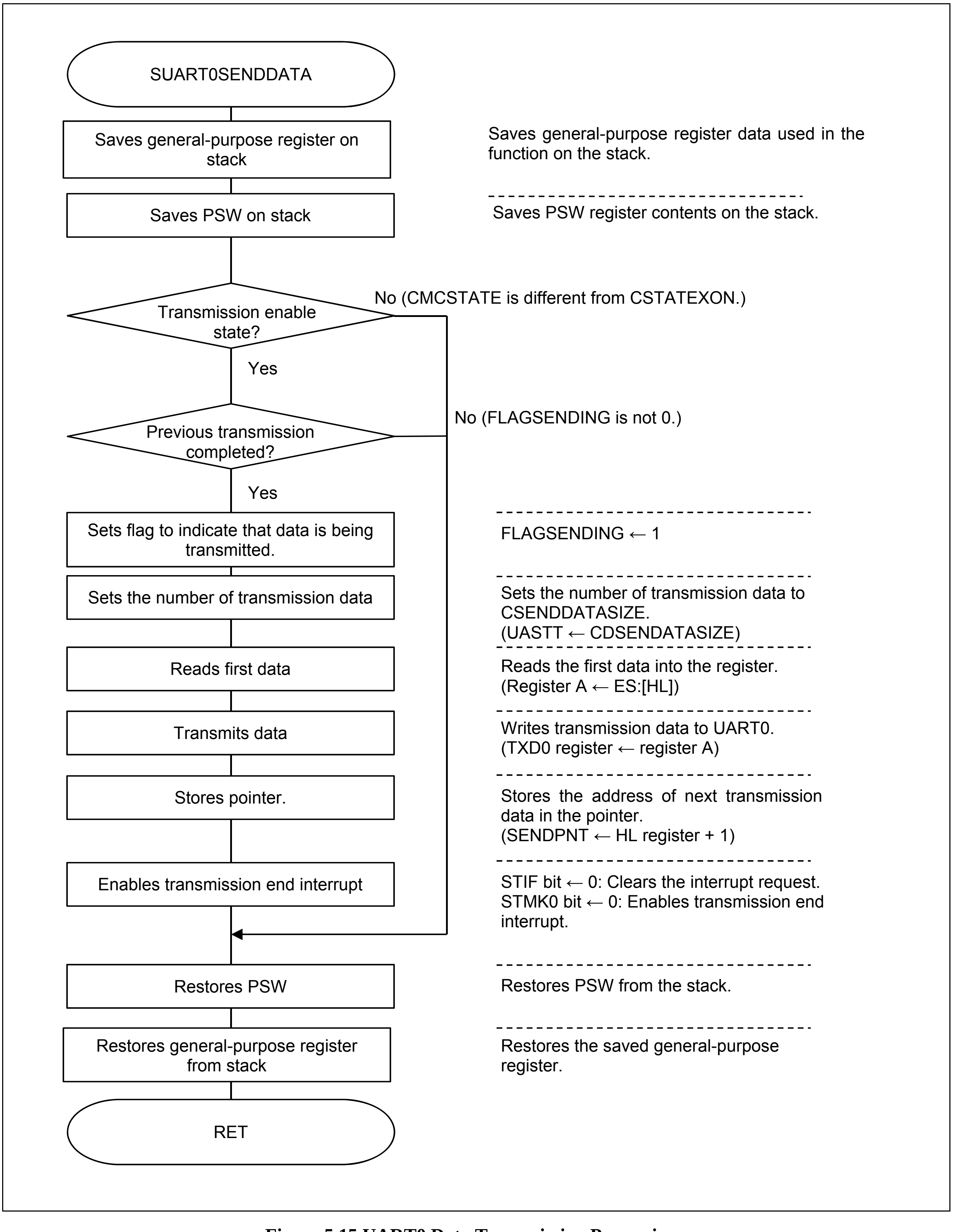


Figure 5.15 UART0 Data Transmission Processing

5.7.13 UART0 Control Code Transmission Processing

Figure 5.16 shows the flowchart for the UART0 control code transmission processing.

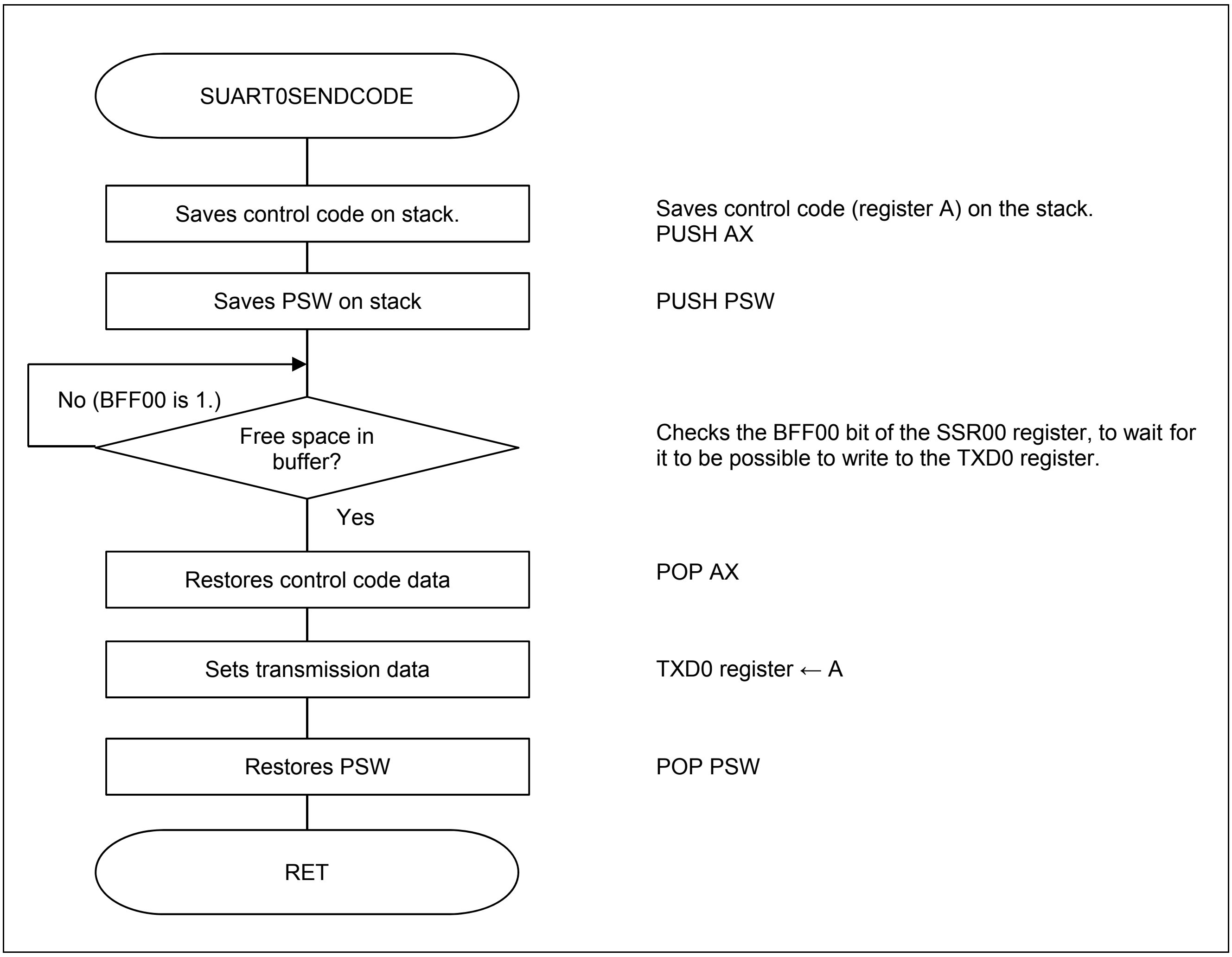


Figure 5.16 UART0 Control Code Transmission Processing

5.7.14 Ring Buffer Store Processing

Figure 5.17 shows the flowchart for the ring buffer store processing.

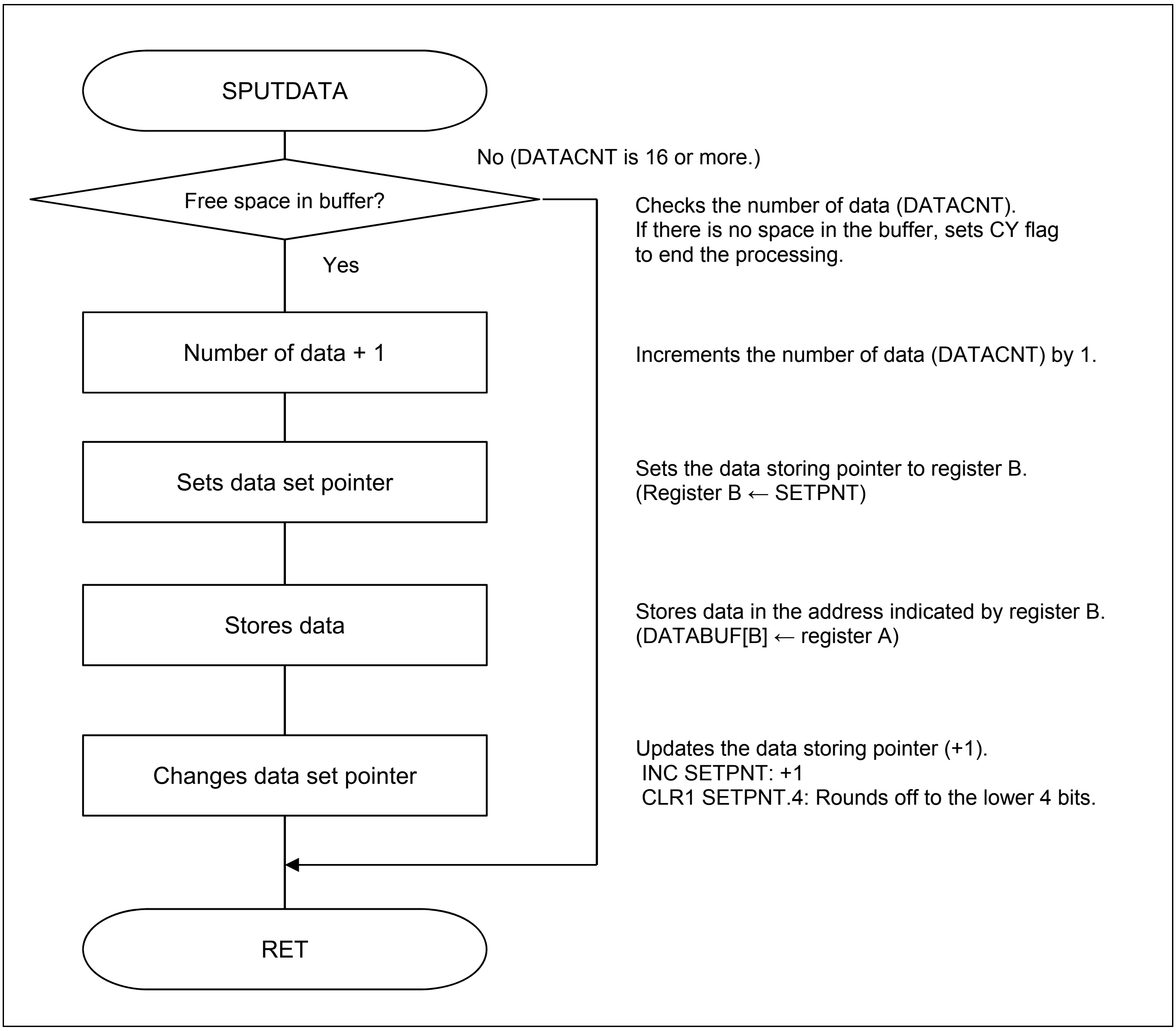


Figure 5.17 Ring Buffer Store Processing

5.7.15 Ring Buffer Store and XOFF Transmission Processing

Figure 5.18 shows the flowchart for the ring buffer store and XOFF transmission processing.

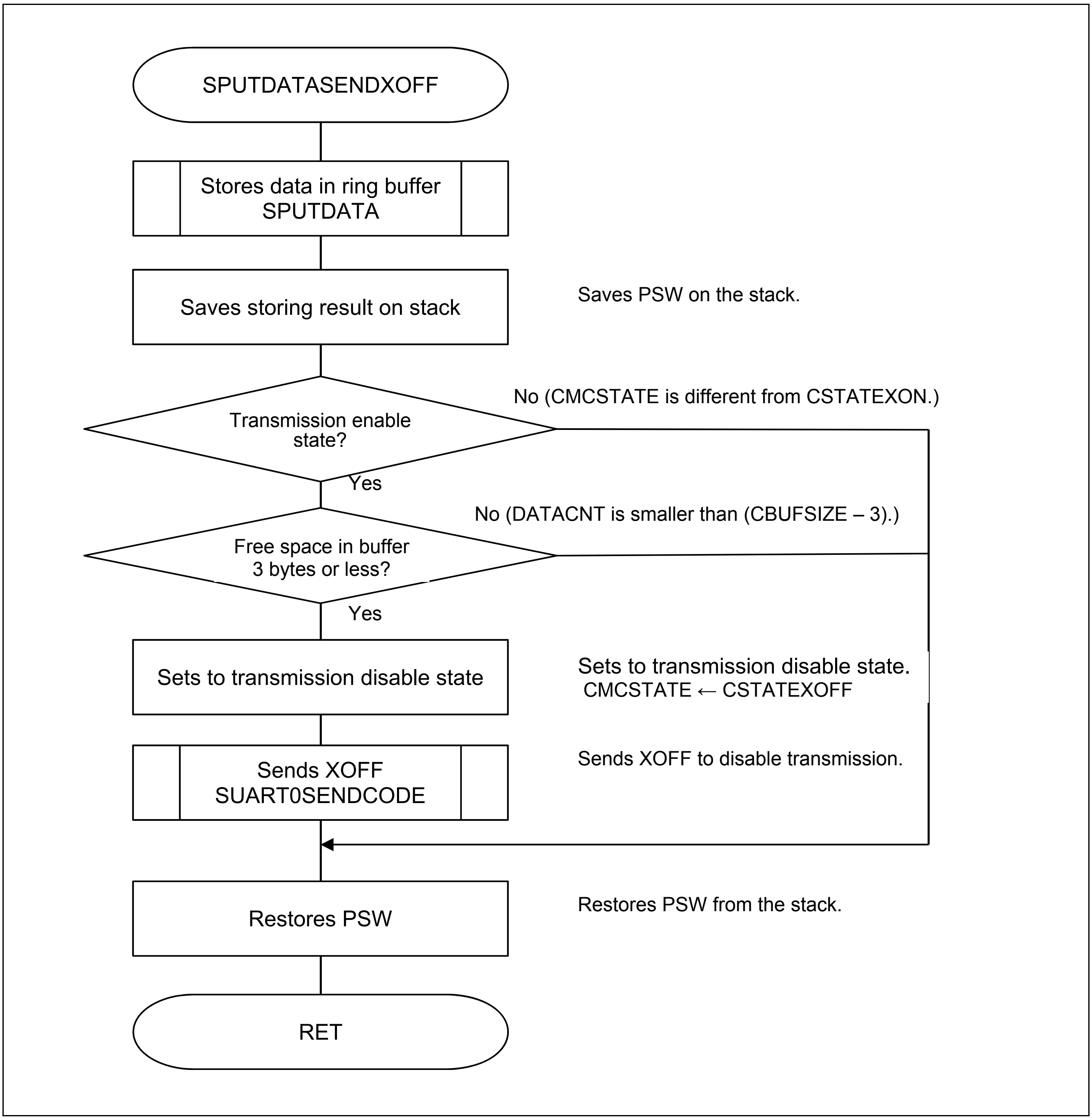


Figure 5.18 Ring Buffer Store and XOFF Transmission Processing

5.7.16 Ring Buffer Read Processing

Figure 5.19 shows the flowchart for the ring buffer read processing.

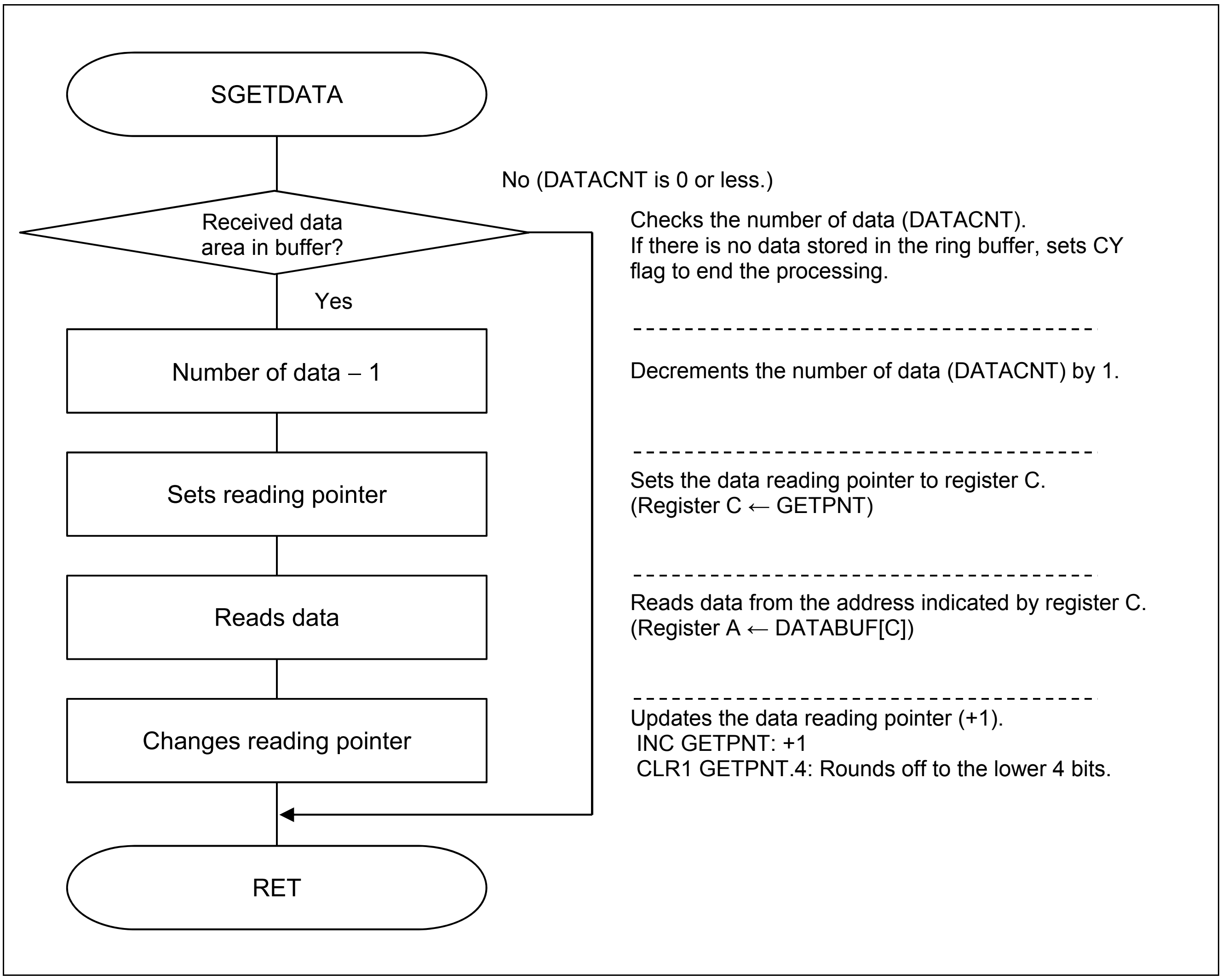


Figure 5.19 Ring Buffer Read Processing

5.7.17 Ring Buffer Read and XON Transmission Processing

Figure 5.20 shows the flowchart for the ring buffer read and XON transmission processing.

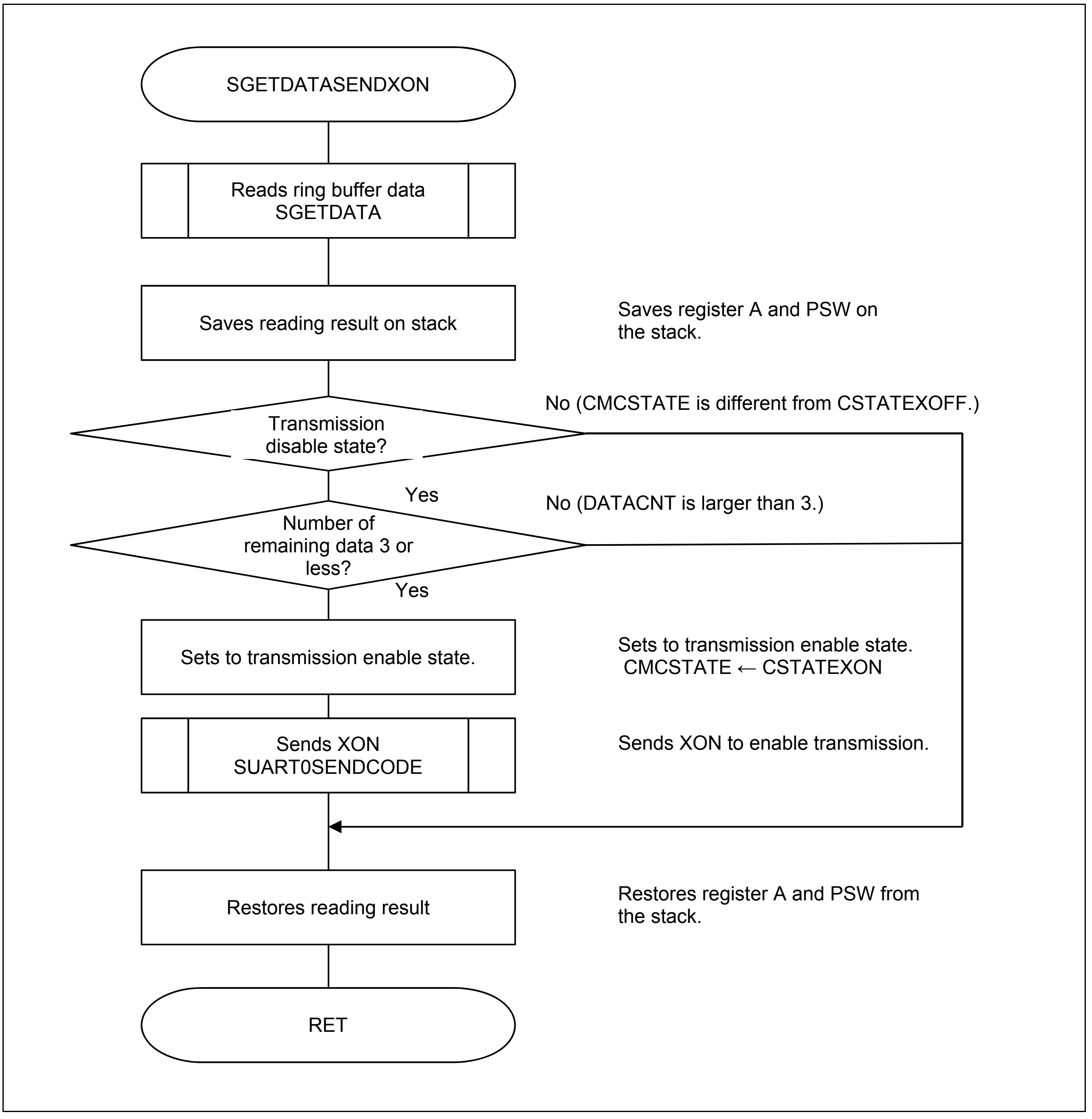


Figure 5.20 Ring Buffer Read and XON Transmission Processing

5.7.18 UART0 Reception End Interrupt Processing

Figure 5.21 and Figure 5.22 show the flowcharts for the UART0 reception end interrupt processing 1 of 2 and 2 of 2.

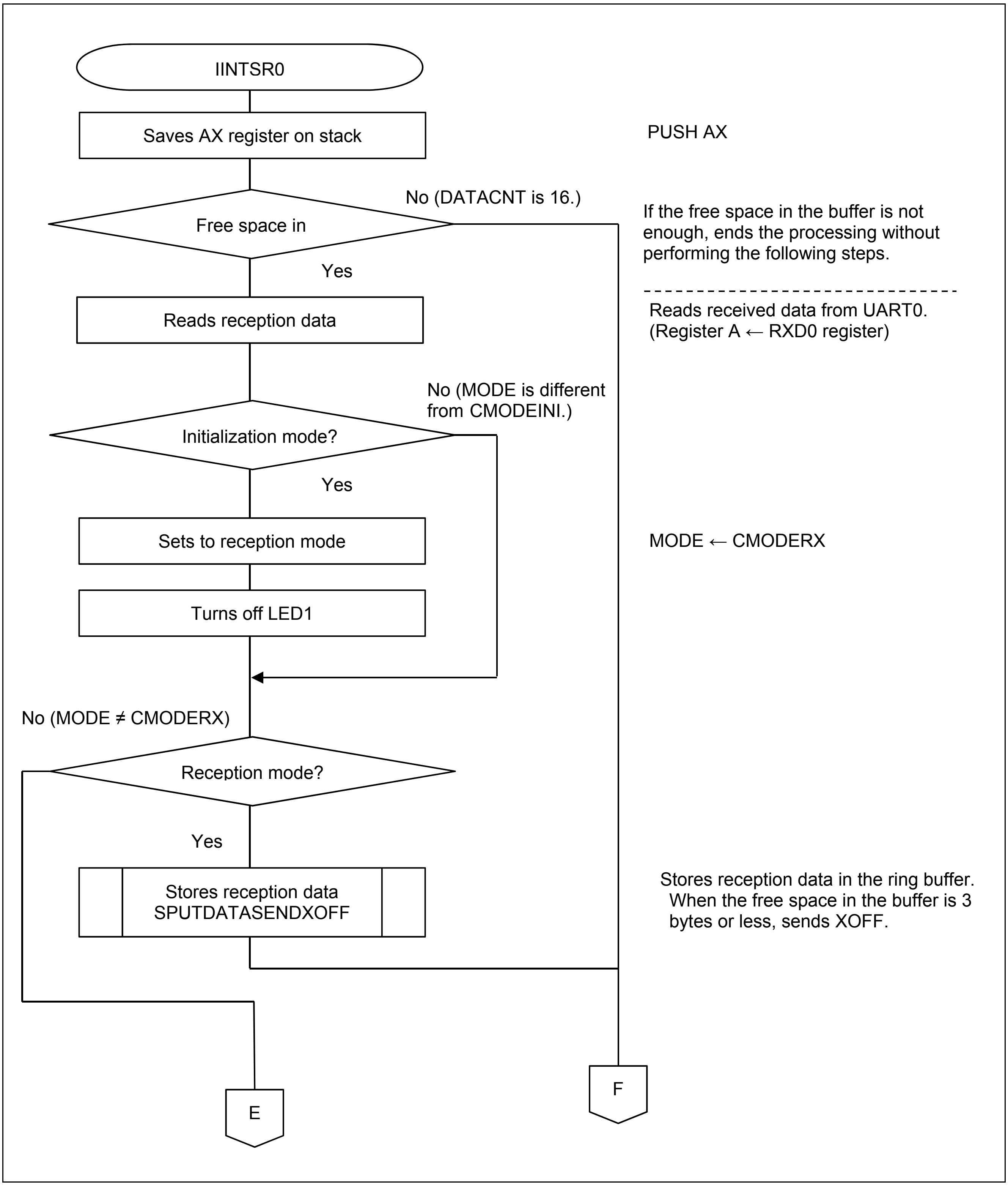


Figure 5.21 UART0 Reception End Interrupt Processing (1 of 2)

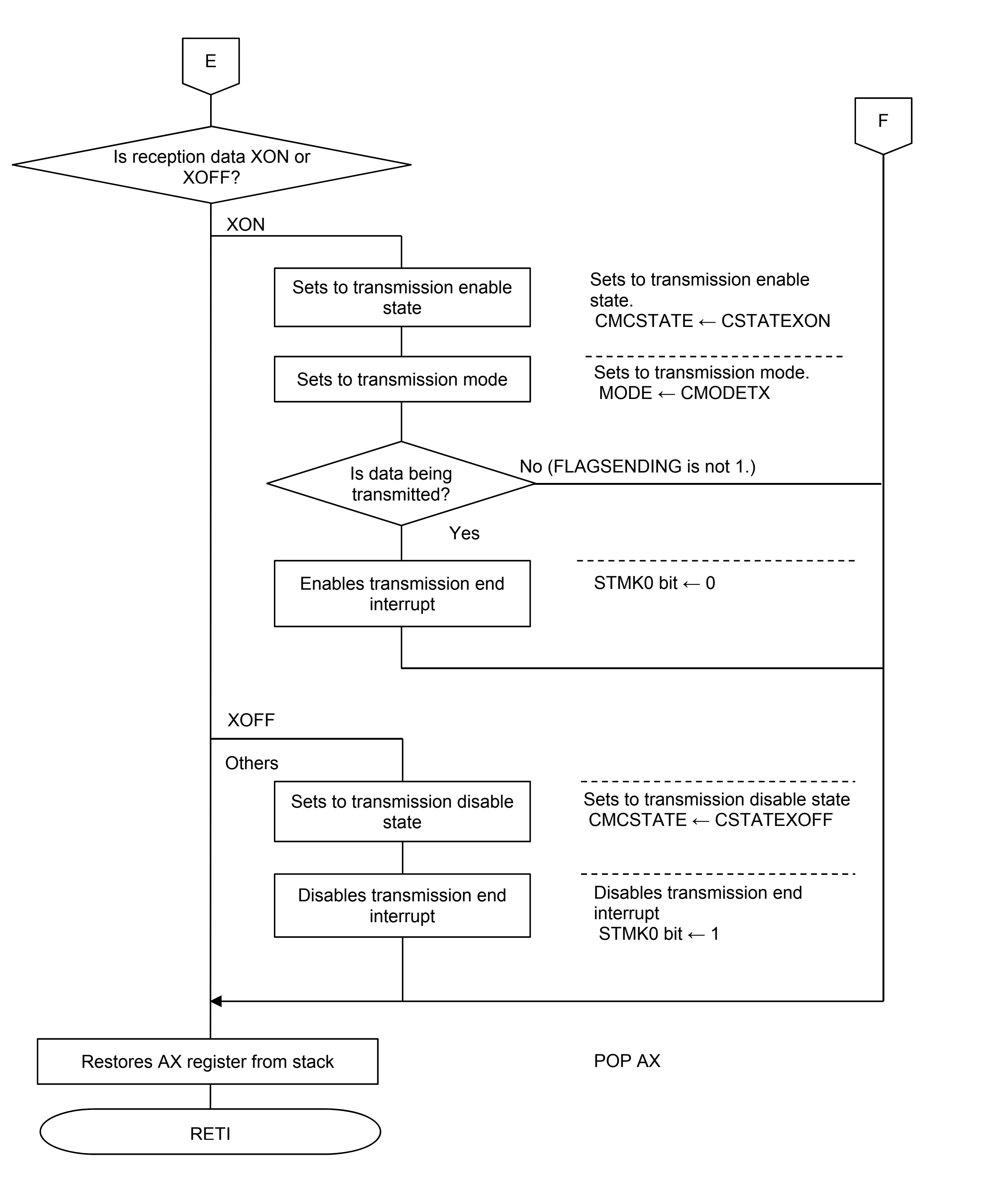


Figure 5.22 UART0 Reception End Interrupt Processing (2 of 2)

5.7.19 UART0 Reception Error Interrupt Processing

Figure 5.23 shows the flowchart for the UART0 reception error interrupt processing.

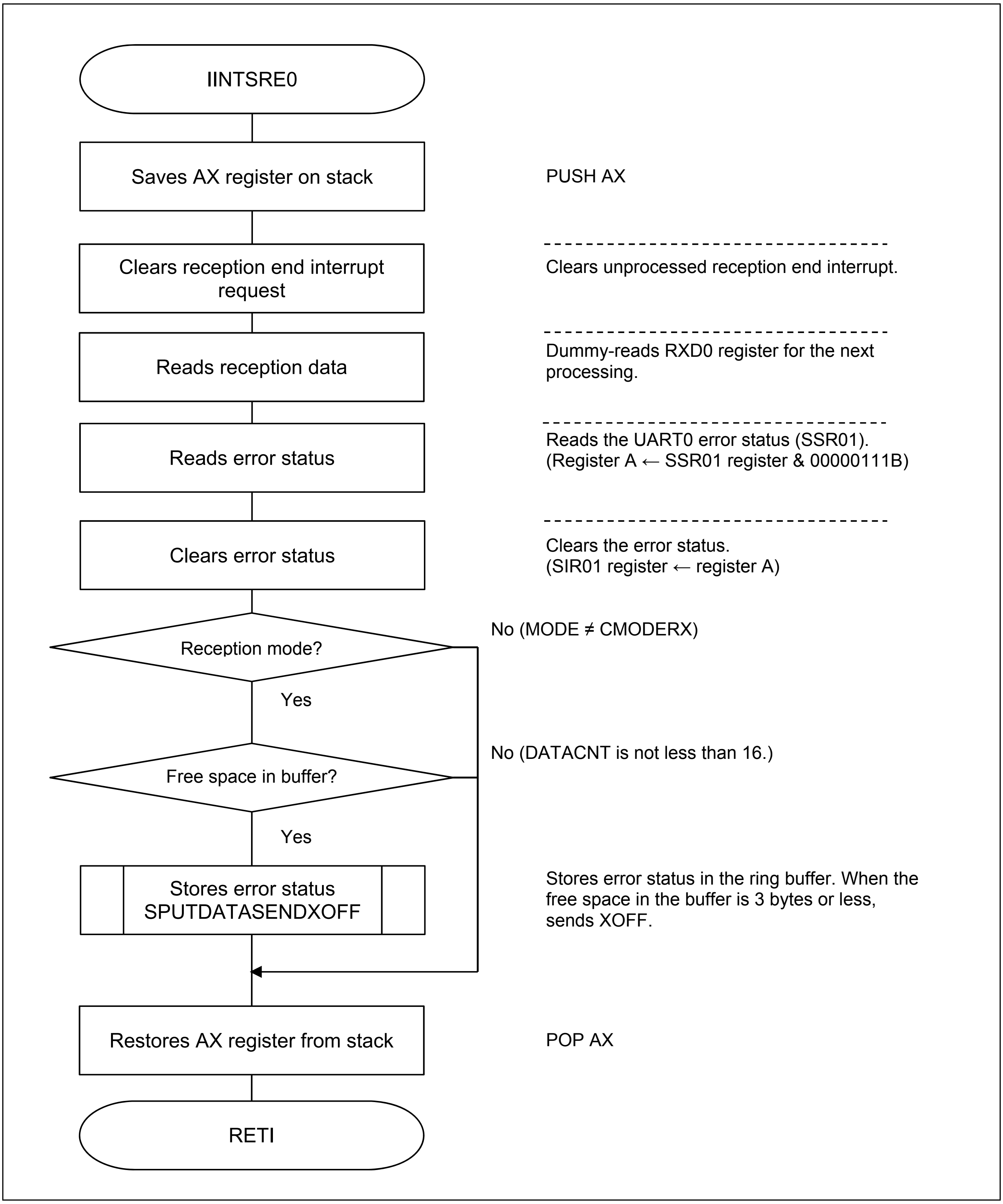


Figure 5.23 UART0 Reception Error Interrupt Processing

5.7.20 UART0 Transmission End Interrupt Processing

Figure 5.24 shows the flowchart for the UART0 transmission end interrupt processing.

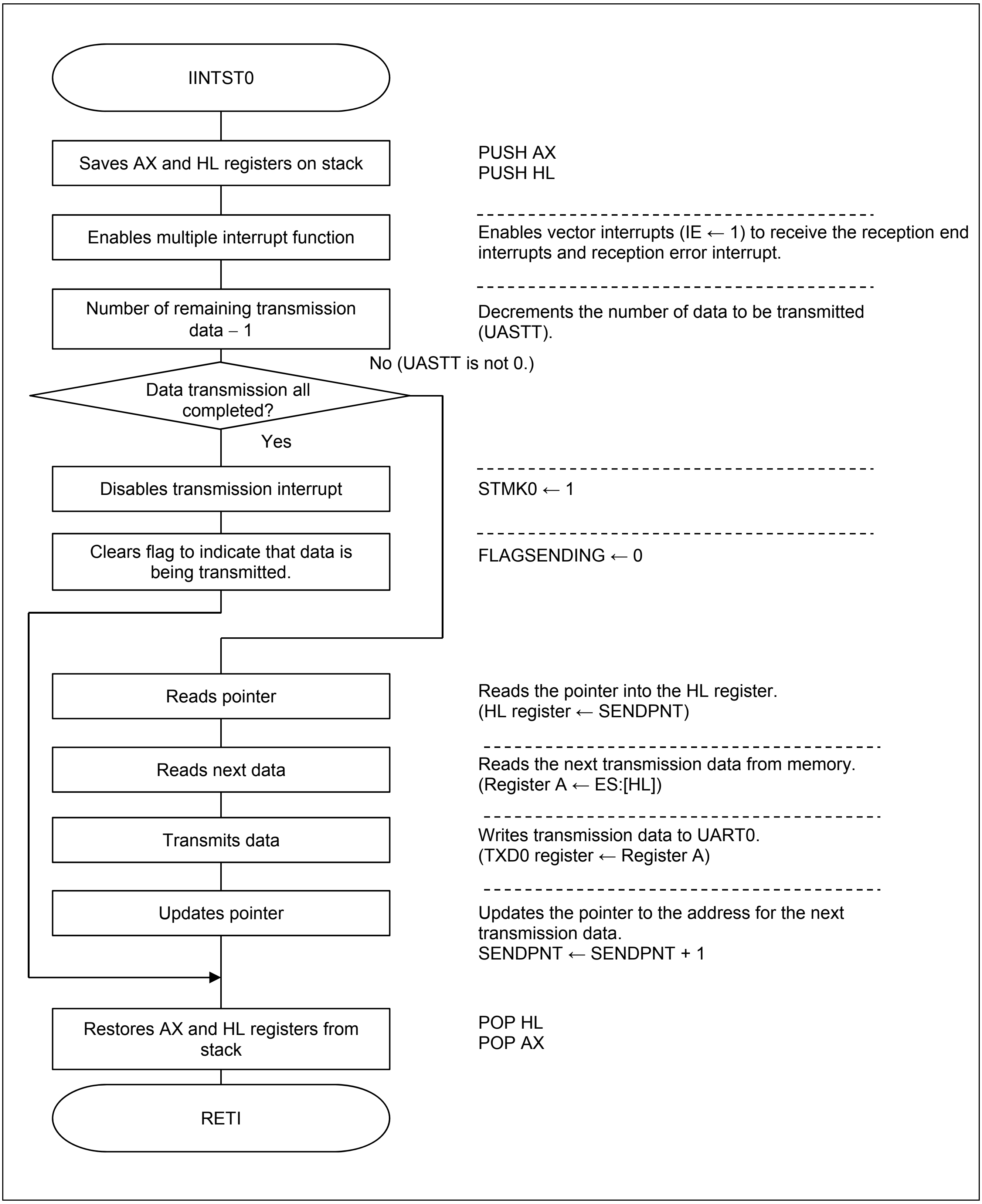


Figure 5.24 UART0 Transmission End Interrupt Processing

5.7.21 INTTM01 Interrupt Processing

Figure 5.25 and Figure 5.26 show the flowcharts for the INTTM01 interrupt processing 1 of 2 and 2 of 2.

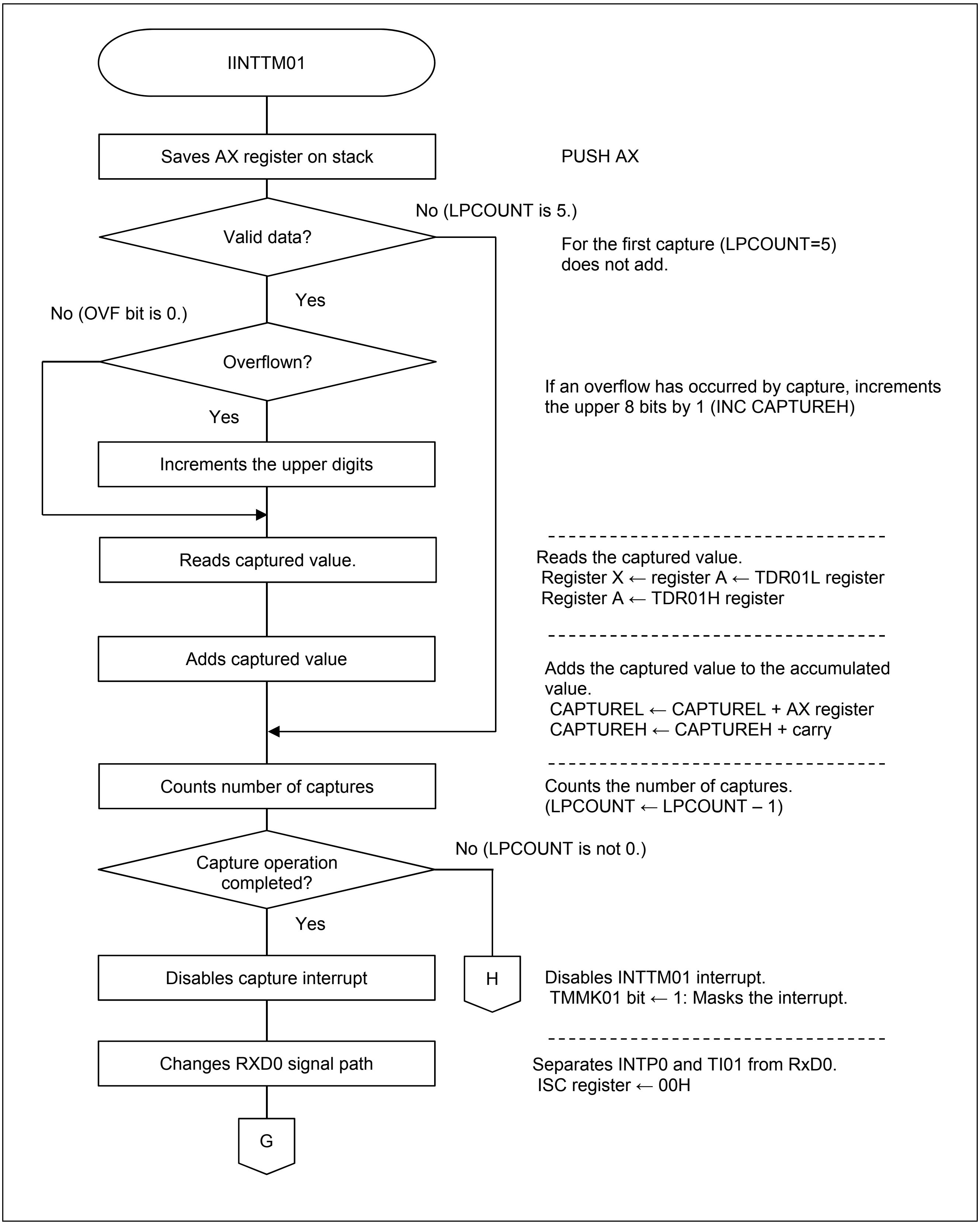


Figure 5.25 INTTM01 Interrupt Processing (1 of 2)

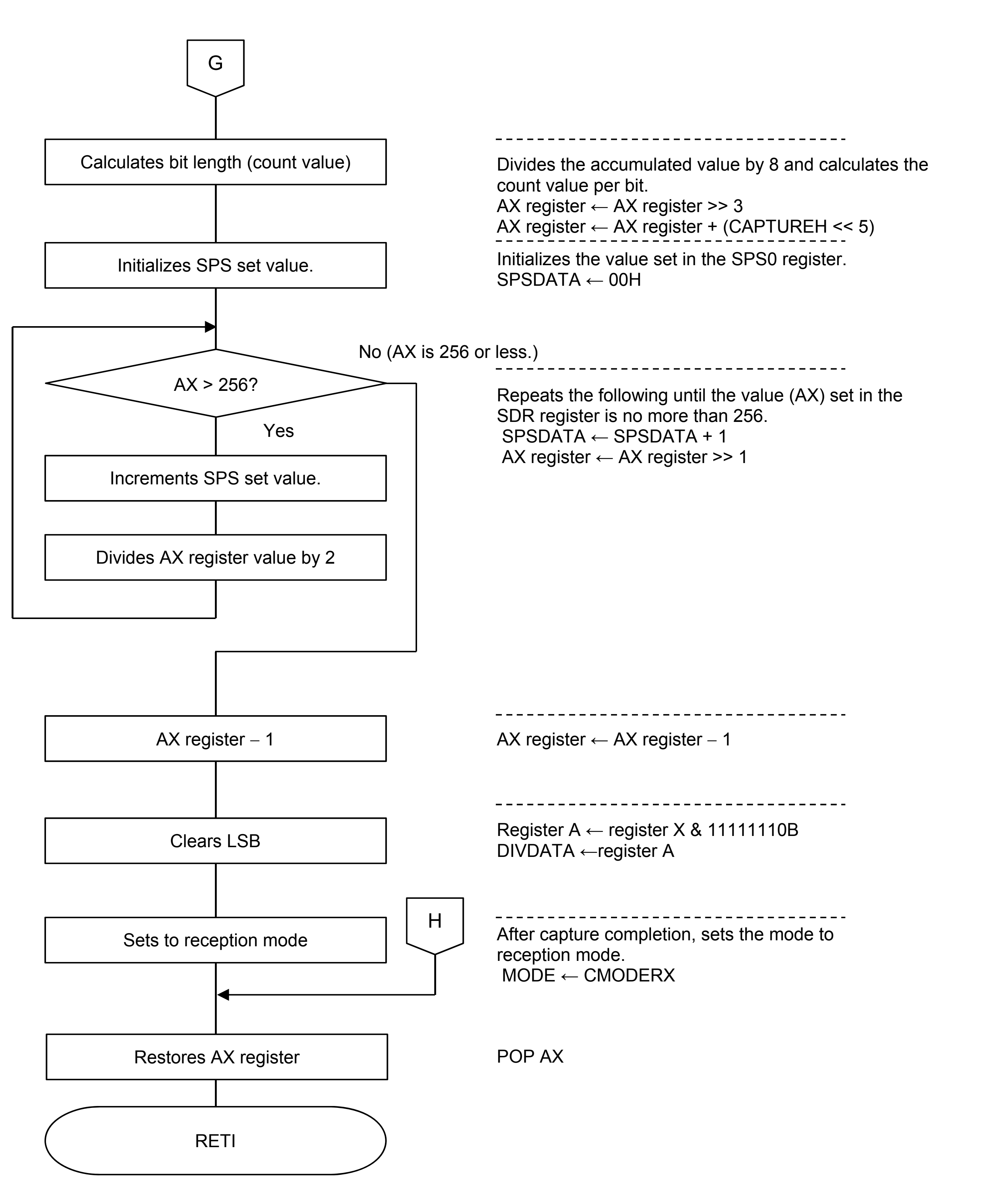


Figure 5.26 INTTM01 Interrupt Processing (2 of 2)

5.7.22 INTP0 Interrupt Processing

Figure 5.27 shows the flowchart for the INTP0 interrupt processing.

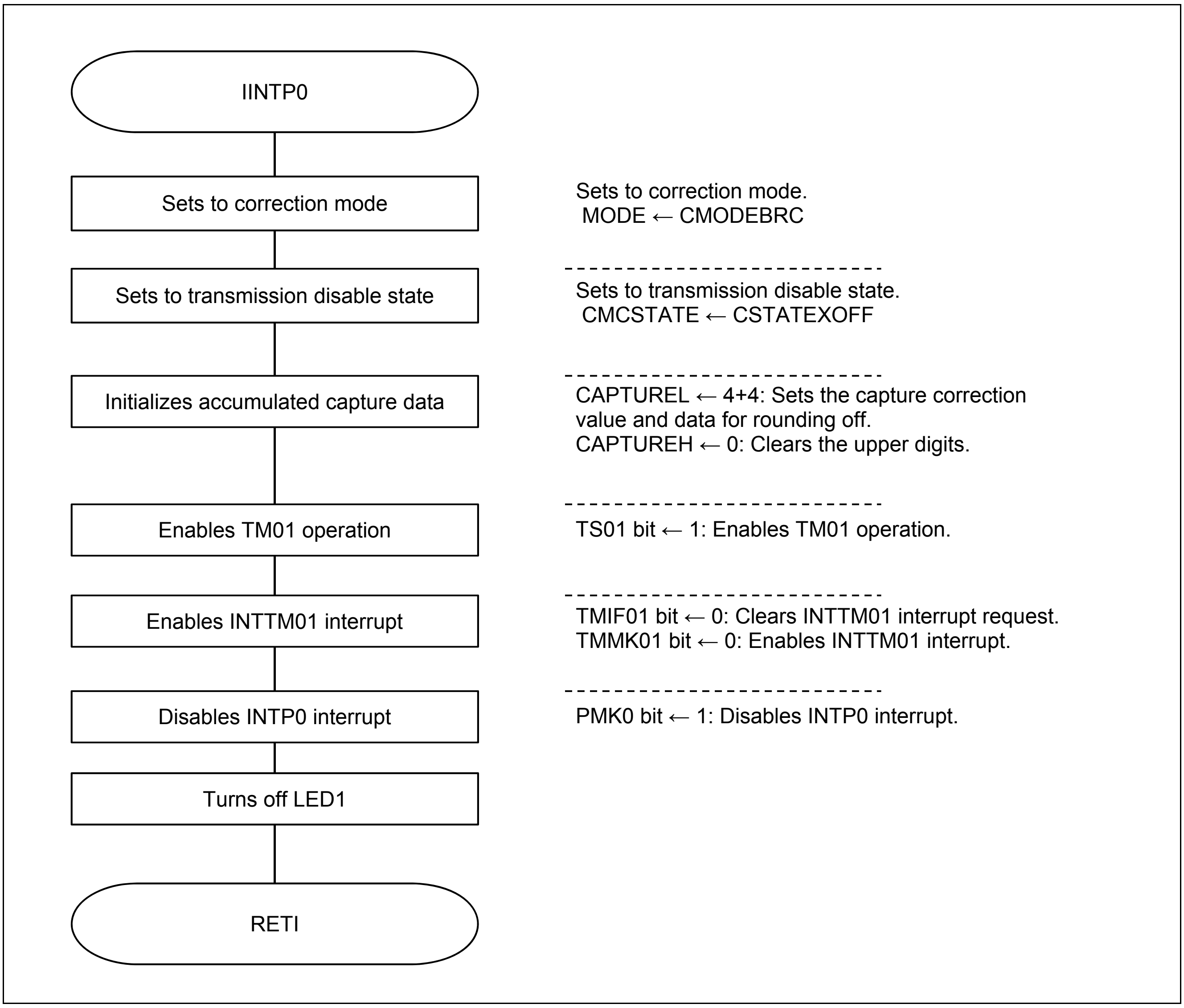


Figure 5.27 INTP0 Interrupt Processing

5.7.23 INTP1 Interrupt Processing

Figure 5.28 shows the flowchart for the INTP1 interrupt processing.

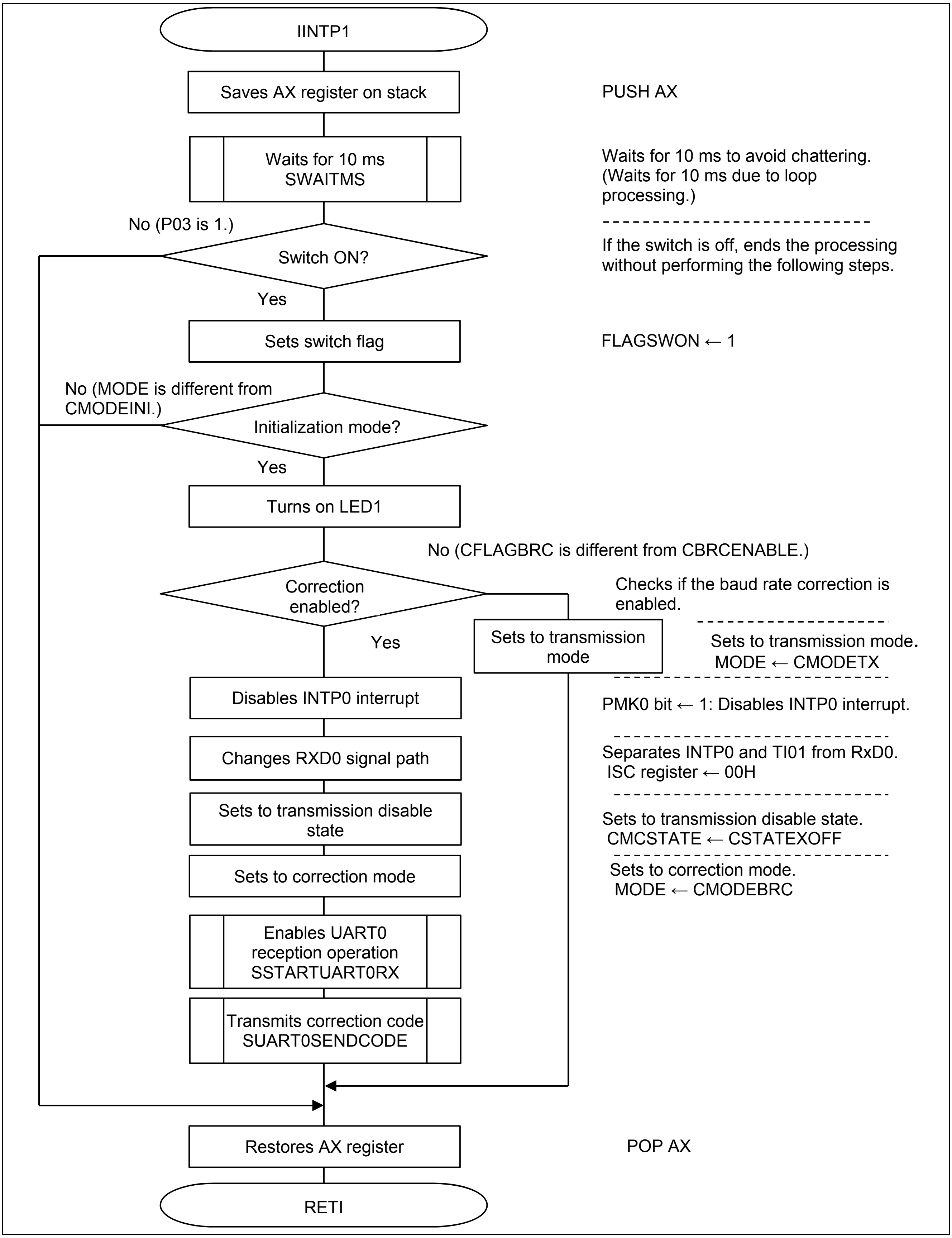


Figure 5.28 INTP1 Interrupt Processing

6. Operation Verification

6.1 Transmission Waveform Verification

Figure 6.1 shows the waveform when "ABC" is transmitted, and Figure 6.2 when XON and XOFF are transmitted.

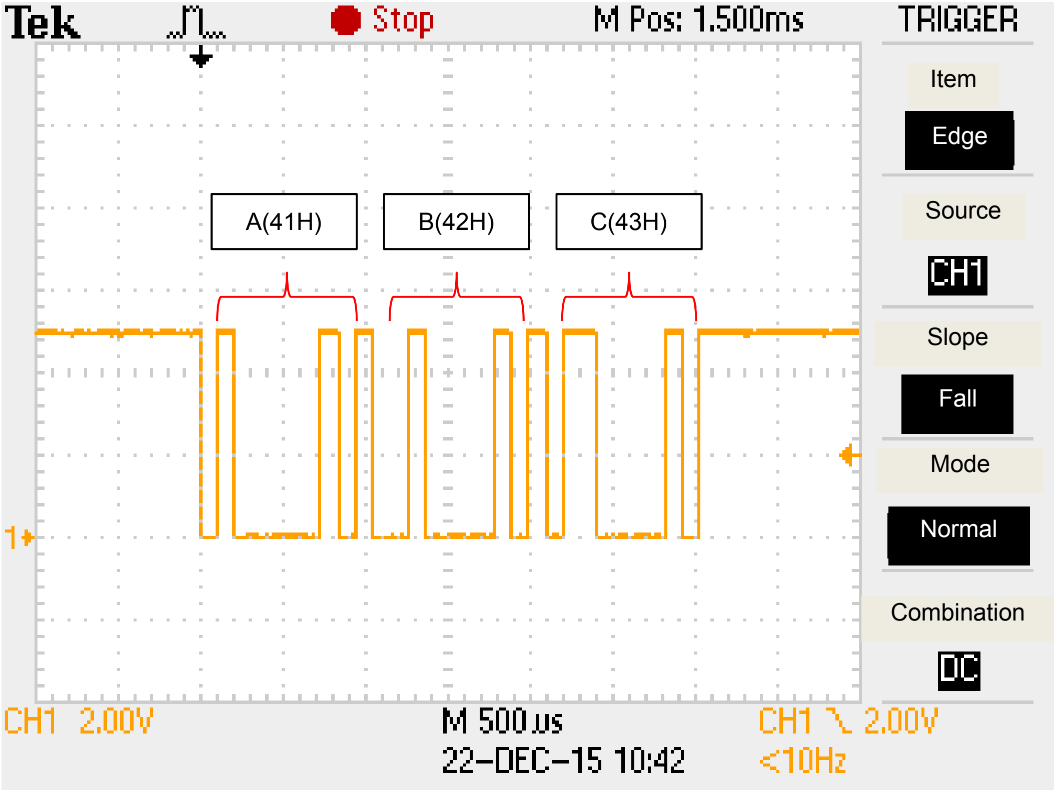


Figure 6.1 Waveform when "ABC" is Transmitted

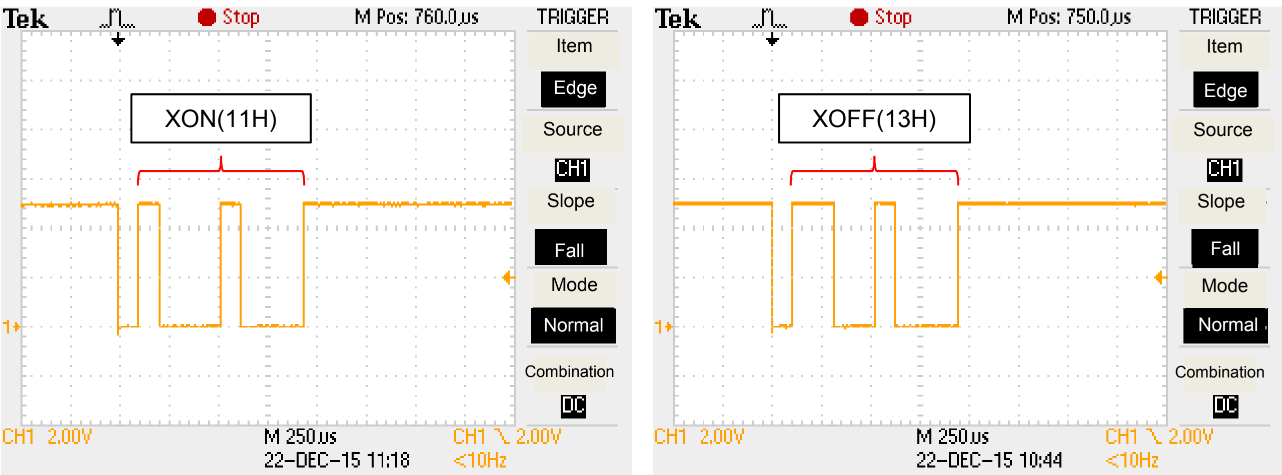


Figure 6.2 Waveforms when XON (Left) and XOFF (Right) are Transmitted

6.2 Baud Rate Correction Operation Verification

We used an oscilloscope to measure “55H” transmitted from the data transmission side and “11H (XON)” transmitted from the data reception side for each set baud rate value. The waveforms at those times are shown in Figure 6.3 through Figure 6.12. Also, Table 6.1 shows the actually measured baud rates calculated from the waveforms and the errors.

Table 6.1 Actually Measured Baud Rates and Errors

Set Value [bps]	Measurement Value on Data Transmission Side [bps]	Error from Set Value [%]	Measurement Value on Data Reception Side [bps]	Error from Data Transmission Side [%]
1200	1200	0	1200	0
9600	9574	-0.27	9574	0
38400	38460	0.16	38460	0
115200	116900	1.5	115400	-1.3
312500	312500	0	312500	0

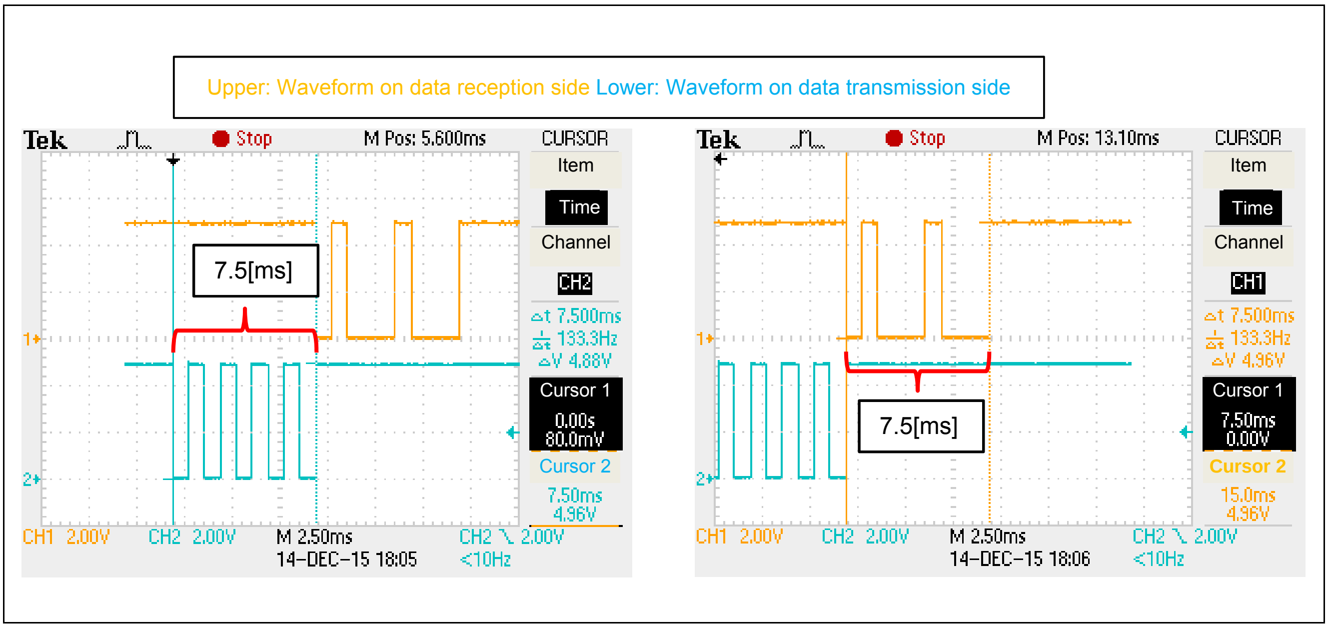


Figure 6.3 Waveforms on Data Transmission Side (Left) and Data Reception Side (Right) (1200bps)

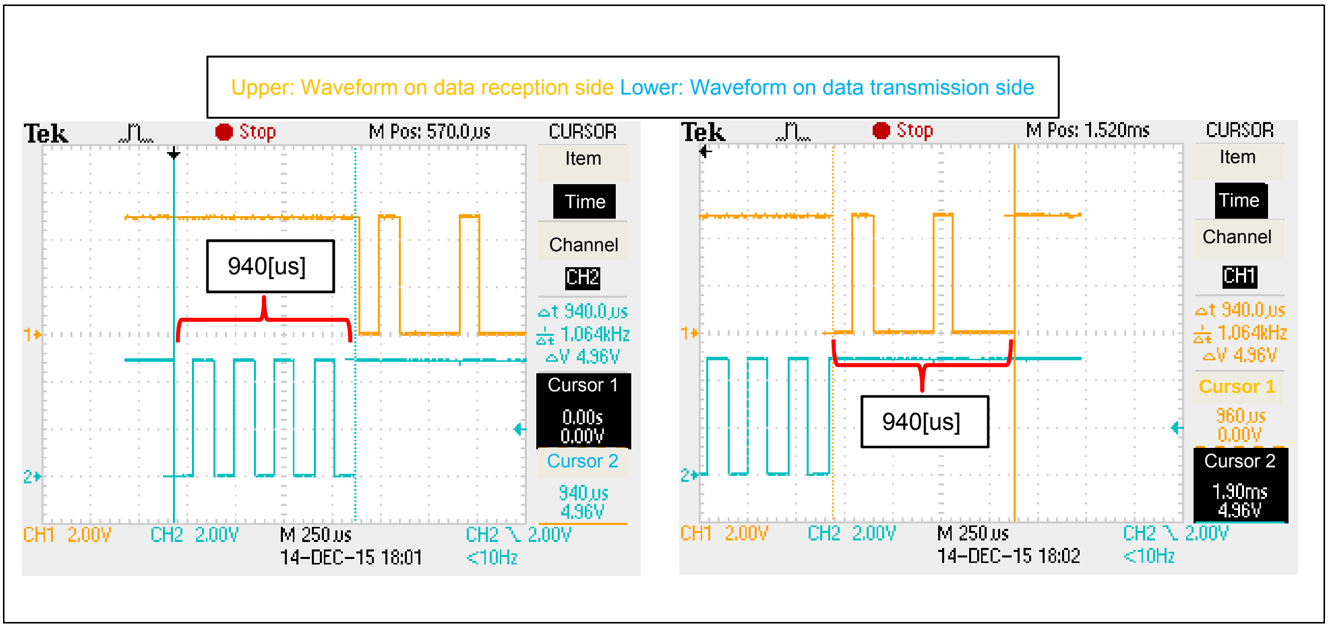


Figure 6.4 Waveforms on Data Transmission Side (Left) and Data Reception Side (Right) (9600bps)

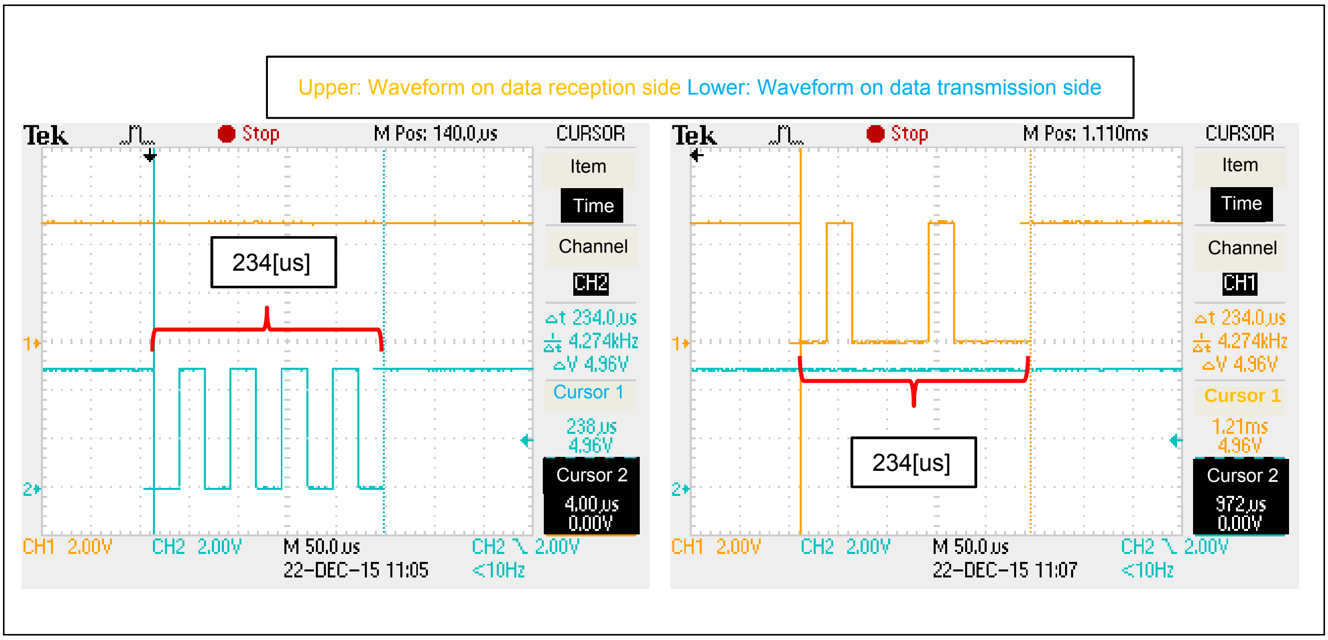


Figure 6.5 Waveforms on Data Transmission Side (Left) and Data Reception Side (Right) (38400bps)

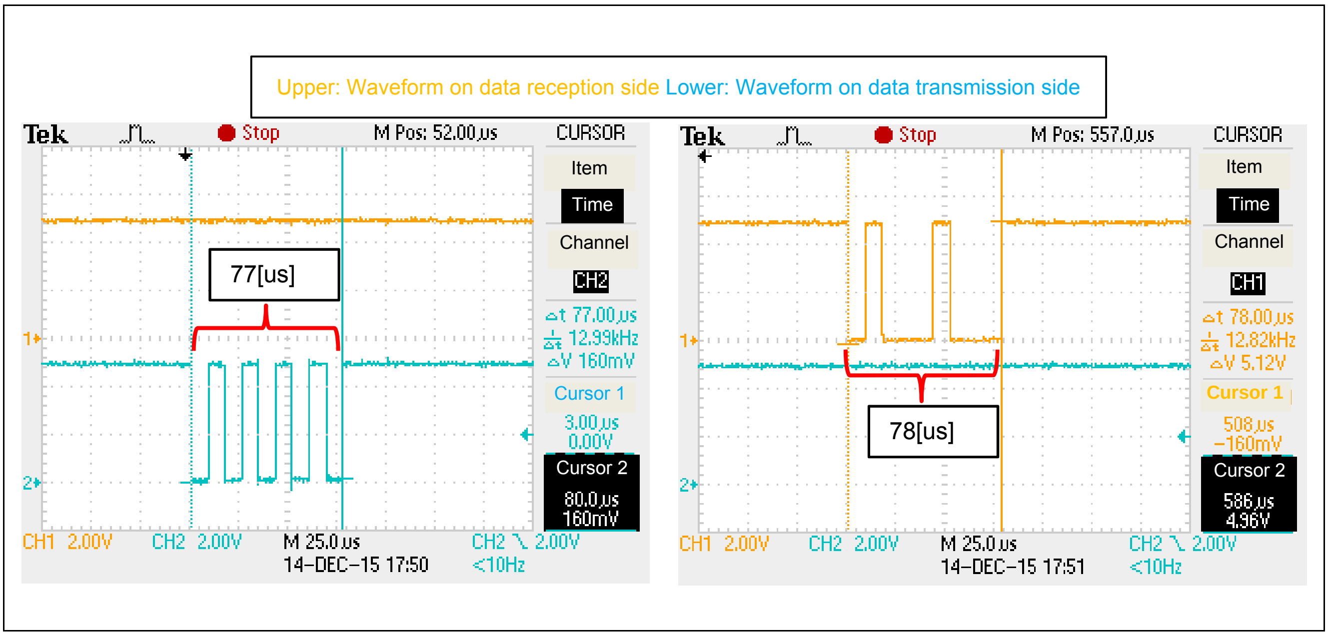


Figure 6.6 Waveforms on Data Transmission Side (Left) and Data Reception Side (Right) (115200bps)

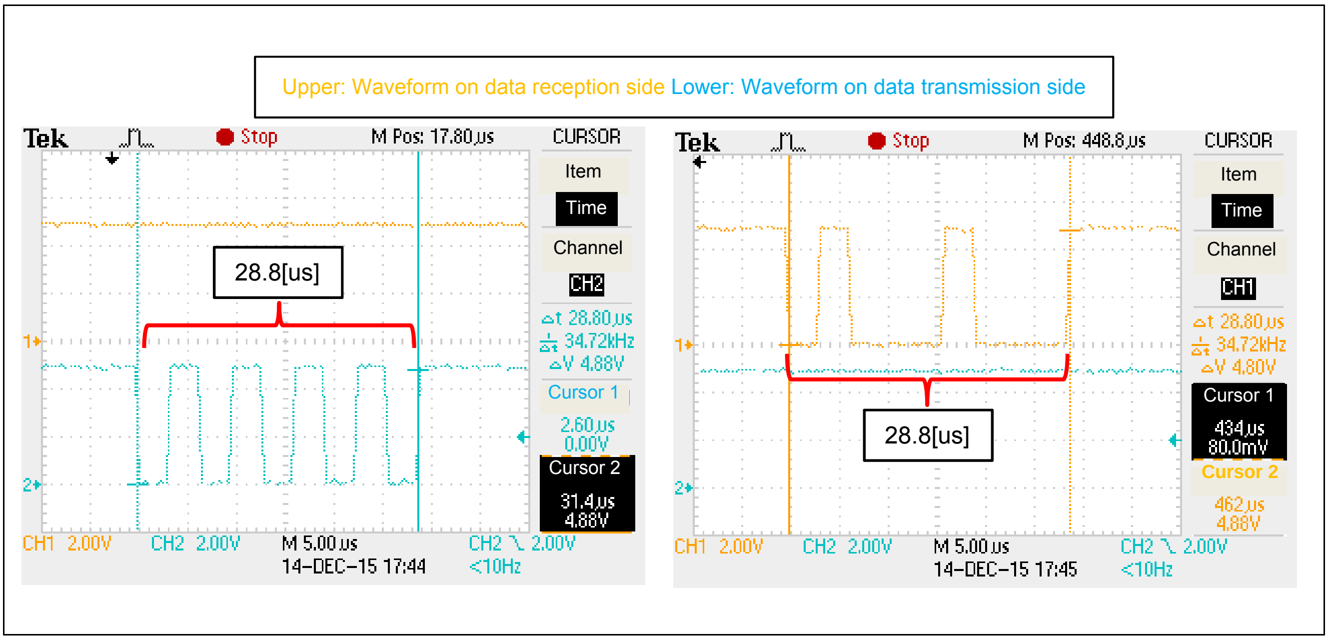


Figure 6.7 Waveforms on Data Transmission Side (Left) and Data Reception Side (Right) (312500bps)

6.3 Example of Baud Rate Correction Failure

If the baud rate of the data transmission side is too fast then baud rate correction will not be performed correctly. For example, Figure 6.8 shows the waveform when 55H is transmitted at 921,600 bps, which is outside the operation range of this sample program, and Figure 6.9 shows the contents of the variables at that time. Since the 11H (XON) that shows the completion of correction is not transmitted, and the value of MODE is ‘B’, we know that the operation of baud rate correction has not completed.

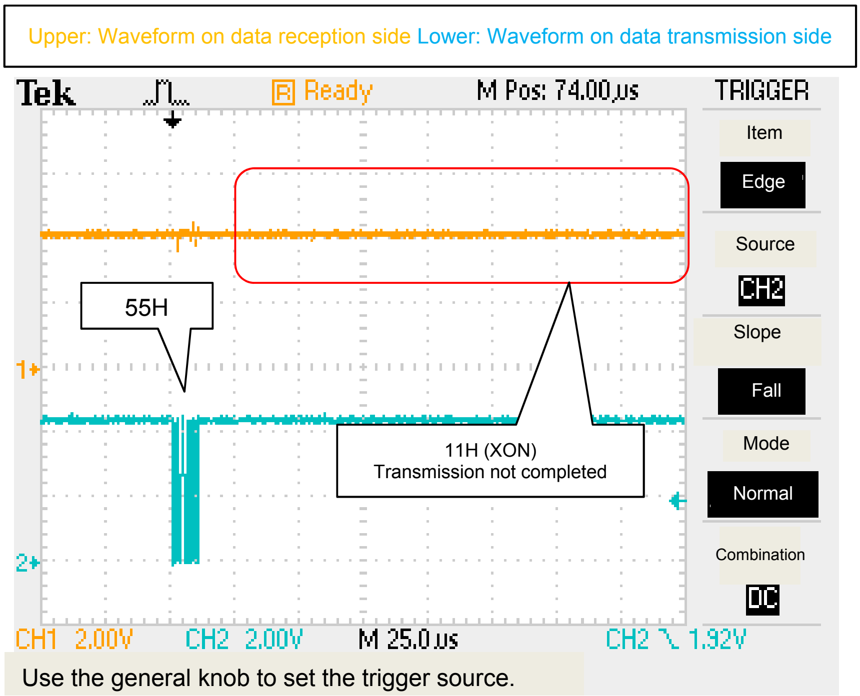


Figure 6.8 Waveform when Baud Rate Correction Fails

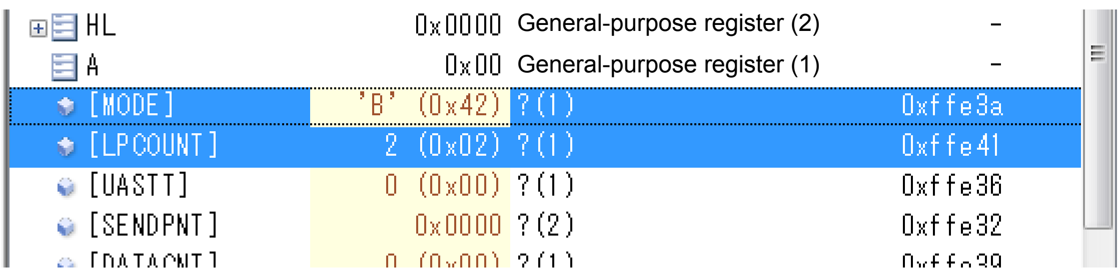


Figure 6.9 Contents of Variables when Baud Rate Correction Fails

6.4 Transitions between Operating Modes

Figure 6.10 shows the contents of the variables (on the transmission side) after the completion of baud rate correction. Since the “MODE” is ‘T’ we know that it has transitioned to transmission mode.

Figure 6.10 shows the contents of the variables (on the reception side) after the completion of baud rate correction. Since the “MODE” is ‘R’ we know that it has transitioned to reception mode.

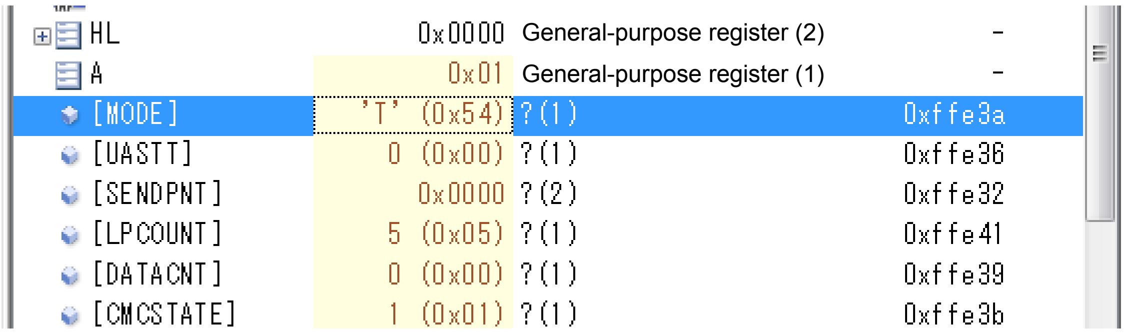


Figure 6.10 Contents of Variables (on Transmission Side) after Completion of Baud Rate Correction

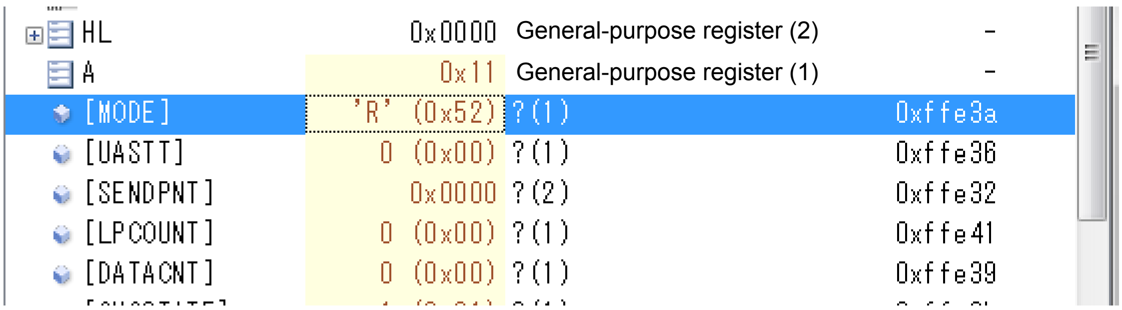


Figure 6.11 Contents of Variables (on Reception Side) after Completion of Baud Rate Correction

6.5 Software Handshake Communication

Figure 6.12 shows the contents of the variables on the data reception side when the three bytes of data (“ABC”) have been transmitted five times from the data transmission side. Since the value of CMCSTATE (which shows whether transmission is currently enabled or disabled) is 0, and the value of DATACNT (which shows the number of bytes in the received data area in the buffer) is 14, we know that it transitioned to transmission enabled state along the way, and the 15th byte of data was not transmitted.

Figure 6.13 shows the contents of the variables on the data reception side when, thereafter, the switch on the data reception side was pressed 11 times. Since the value of CMCSTATE has become 1, we know that the value of DATACNT was once 3, so that it transitioned to transmission enabled state. After that, the transmission of the 15th byte of data that had been interrupted was restarted, so the value of DATACNT is 4.

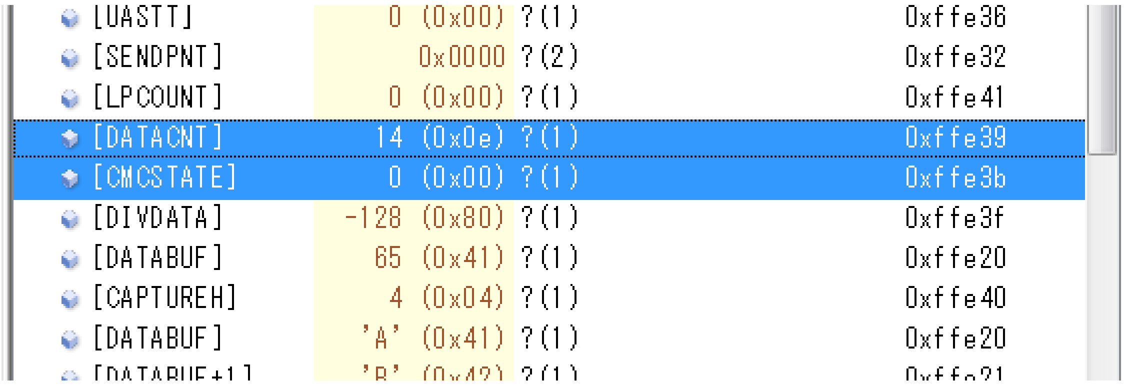


Figure 6.12 Contents of Variables on Data Reception Side after Transitioning to Transmission Disabled State

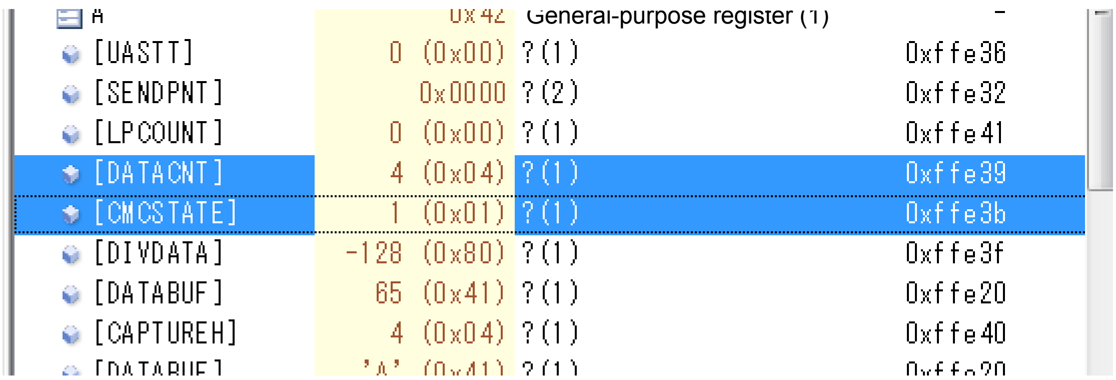


Figure 6.13 Contents of Variables on Data Reception Side after Transitioning to Transmission Enabled State

7. Sample Code

The user can get the sample code from the Renesas Electronics website.

8. Reference Documents

RL78/G10 User's Manual: Hardware (R01UH0384JJ)

RL78 Family User's Manual: Software (R01US0015JJ)

(Get the latest version from the Renesas Electronics website.)

Technical Updates/Technical News

(Get the latest information from the Renesas Electronics website.)

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History <revision history,rh>

Rev.	Date	Description	
		Page	Summary
1.00	2016/10/05	–	First edition issued

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. 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 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 a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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 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": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime 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 (artificial 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 respect to the maximum rating, operating supply voltage range, movement 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 the failure of a Renesas Electronics product, 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, 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 inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring 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 duplicated 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.

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

(Note 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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0