RL78/G11

UART communication with middle speed on chip oscillator IAR

R01AN3621EJ0100
Rev.1.00
Feb. 15, 2017

Summary

The application note explains the UART communication method with middle speed on chip oscillator. The middle speed on chip oscillator is periodically measured by high speed on chip oscillator that has the frequency accurate enough requested by the UART communication. UART with the middle speed on chip oscillator communicates by correcting the baud rate of the UART communication based on the result of the measurement.

Object device

RL78/G11

When applying the application note to other microcomputers please change it according to the specification and evaluate it enough.

Contents

1. Specification

[Explanation of specification]

Table 1.1 shows the used peripheral function and the usage. Figure 1.1 shows the outline of the application.

Table 1.1 Used peripheral function and usage

Function in surrounding	Usage
Serial array unit	UART communication
Timer array unit	The division of middle speed on chip oscillator
8-bit interval timer	The frequency measurement of middle speed on chip oscillator

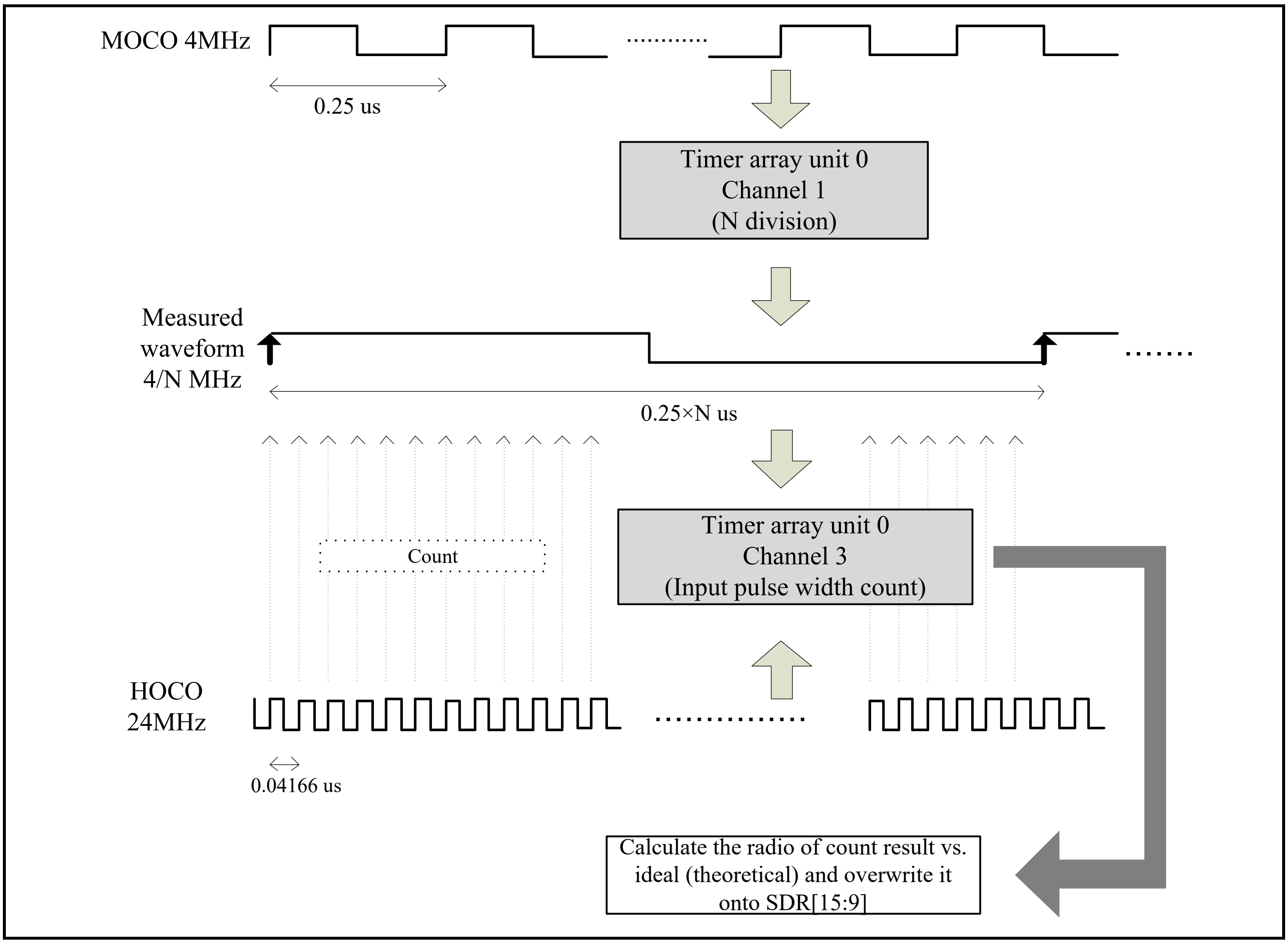


Figure 1.1 Outline of the application

2. Operation condition confirmation

The sample code of this application note confirms operation at the conditions stated below.

Table 2.1 Operation condition confirmation

Item	Content
MCU	RL78/G11 (R5F1056A)
Operating frequency	・Middle speed on chip oscillator clock (f_{IM}): 4MHz (at UART communication). ・High speed on chip oscillator clock (f_{IH}): 24MHz (at MOCO frequency measurement).
Operation voltage	3.0 V (Able to operate at 1.6V-3.6V). LVD operation (V_{LVD}): Reset mode 2.81V (2.76V to 2.87V)
Integrated development environment	IAR Embedded Workbench for Renesas RL78 V2.21.2
C compiler	IAR C/C++ Compiler for Renesas RL78 V2.21.1.1833

3. Related application note

The application note that relates to this application note is shown below.

RL78/G13 initialization (R01AN2575EJ)

RL78/G11 serial array unit (UART communication) IAR (R01AN3646EJ)

4. Hardware explanation

4.1 Example of hardware structure

Figure 4.1 shows the connection example.

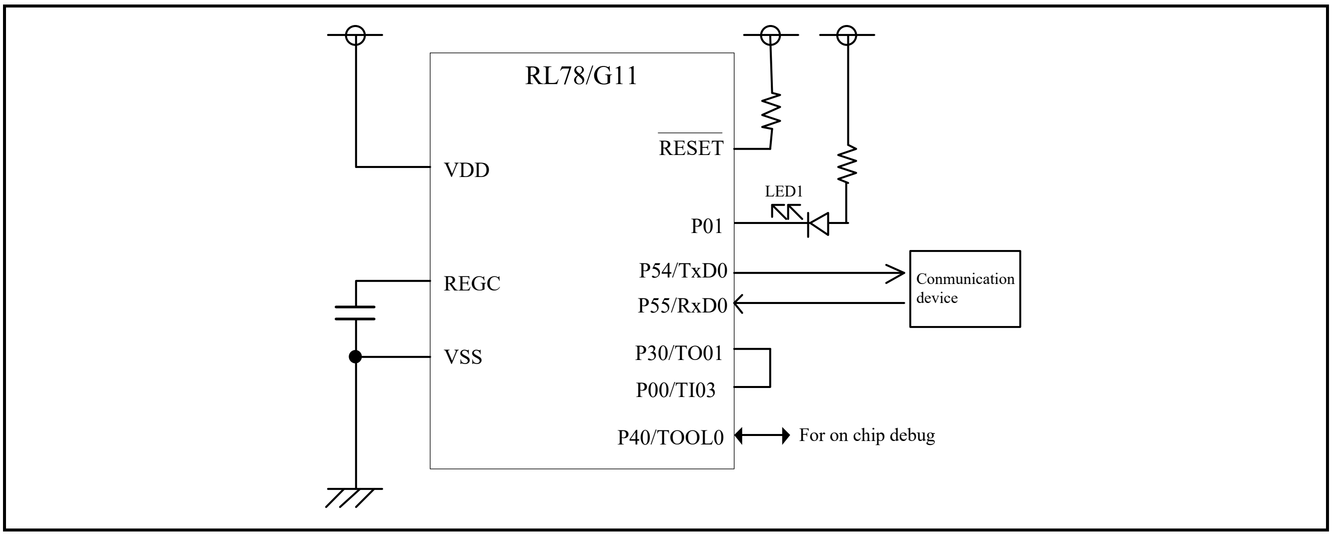


Figure 4.1 Connection example

Note: 1. It is the circuit image and shows the simplified outline of the connection. Please design to do the pin processing etc. appropriately, and to meet with the electrical characteristics when the circuit is actually designed (Please connect the input only port to VDD or VSS through a resistor individually).

2 Please connect it with VDD if there the pin name starts by EVDD, and to VSS if the pin name starts by EVSS respectively.

3 Please make the VDD more than reset release voltage (V_{LVD}) set in LVD.

4.2 Used pin list

Table 4.1 shows the used pin and function.

Table 4.1 Used pin and function

Pin name	I/O	Content
P01	Output	LED turning on
P30/TO01	Output	MOCO division waveform output
P00/TI03	Input	Frequency measuring waveform input

5. Software explanation

5.1 Outline of operation

In this sample code, data corresponding to the data received from another device is transmitted back to the device. When the error occurs, data corresponding to the error is transmitted to the device. Table 5.1 and Table 5.2 show correspondence tables of receive data and the transmission data, error detection respectively.

Table 5.1 Correspondence of receive data and transmission data

Receive data	Response (transmission) data
T(54H)	O(4FH), K(4BH), "CR"(0DH), "LF"(0AH)
t(74H)	o(6FH), k(6BH), "CR"(0DH), "LF"(0AH)
Others	U(55H), C(43H), "CR"(0DH), "LF"(0AH)

Table 5.2 Correspondence of transmission data when error is detected

Occurred error	Response (transmission) data
Parity error	P(50H), E(45H), "CR"(0DH), "LF"(0AH)
Flaming error	F(46H), E(45H), "CR"(0DH), "LF"(0AH)
Overrunning error	O(4FH), E(45H), "CR"(0DH), "LF"(0AH)

(1) UART is initialized.

< UART setting condition >

- SAU0 channels 0 and 1 are used as UART.
- P54/TxD0 is used for data output pin P55/RxD0 is used for data input pin.
- The data length uses 8-bit.
- Data transfer.direction.setting.is LSB first.
- The parity setting uses the even parity.
- Receive data and the level setting use the standard.
- Transfer rate uses 9600bps.
- Reception complete interrupt (INTSR0), transmission complete interrupt (INTST0), and error interrupt (INTSRE0) are used.
- The interrupt priority level of INTSR0, INTST0, and INTSRE0 uses the low priority (level 3).

(2) 8-bit interval timer is initialized.

< 8-bit interval timer setting condition >

- The interrupt priority level of 8-bit interval timer interrupt (INTIT00) uses the low priority (level 3).
- It uses it in 8-bit count mode.
- One division frequency is set to fail/64.
- The compare value is set to 0xE9 so that INTIT00 is generated around every second.

(3)The timer array unit is initialized.

< Channel 1 setting condition >

- The operation clock is middle speed on chip oscillator (MOCO) 4MHz.
- It operates as 16-bit timers.
- It sets software trigger start
- MOCO effective edge is falling edge
- It operates as interval timer mode/square waveform output.
- The timer interrupt is not generated at the start of counting.
- Positive logic output

< Channel 3 setting condition >

- The operation clock is high speed on chip oscillator (HOCO) 24MHz.
- It sets single channel operation
- An effective edge of the pin TI03 is set as start trigger and a capture trigger.
- Effective edge of pin TI03: Rising edge
- It operates as input pulse width measurement mode.
- Positive logic output

(4) After setting communication standby state in the serial channel start register, the HALT instruction is executed. The process is triggered by the reception complete interrupt (INTSR0) and error interrupt (INTSRE0).

- When INTSR0 is generated, data is received, and data corresponding to the receive data is transmitted. When INTSRE0 is generated, it processes the error and transmits the data corresponding to the error. 8-bit interval timer stops in the period of transmission response processing. After data reception, the HALT instruction is executed again, and it waits for reception complete interrupt (INTSR0), error interrupt (INTSRE0).
- In the case of any factors, that doesn't relates to 8-bit interval interrupt and timer interrupt but wake up MCU from HALT to normal operation, it goes back to HALT without any processing.

(5) The baud rate is corrected by 8-bit interval timer interrupt.

- LED that shows the baud rate correction processing is turned on.
- UART0 stops operation.
- Operation HOCO and wait for oscillation stabilization time.
- CPU clock source is switched from MOCO to HOCO.
- TAU0 channel 1 and channel 3 start to operate. It stands by in HALT state until the 2nd time channel 3 complete interrupt generation.
- After the 2nd time channel 3 complete interrupt generation, TAU0, channel 1, and channel 3 stop operation.
- It counts backward from the result of a measurement, and the baud rate correction value is written into the SDR register.
- CPU clock is switched back from HOCO (24MHz) to MOCO (4MHz) in preparation for the UART communication.
- HOCO oscillation is stopped.
- UART0 starts operation.
- LED that shows the baud rate correction processing is turned off.

5.2 Consideration of correction processing

The following explains the detail of the correction processing method.

(1) The generation of object waveform to be measured

As shown in Figure 1.1, the middle speed on chip oscillator (MOCO) is divided in N and generates the object waveform to be measured. Enlarging the value of N as much as possible makes the measurement more accurate. In other hand, it should be within the 16-bit range for timer array unit 0 channels 3 to count. Based on HOCO to be 24MHz±1% and MOCO to be 4MHz±12% , Table 5.3 shows the measurement (count) result in the combination of each Max., Typ. and Min. value.

Table 5.3 The result of measurement for the waveform N divided from MOCO.

		MOCO (4 MHz±12%)		
		Min.: 3.52 MHz	Typ.: 4.00 MHz	Max.: 4.42 MHz
HOCO (24 MHz±1%)	Max: 24.24 MHz	(24.24 / 3.52) N = **6.886N**	(24.24 / 4.00) N = **6.060N**	(24.24 / 4.42) N = **5.484N**
	Typ.: 24.00 MHz	(24.00 / 3.52) N = **6.818N**	(24.00 / 4.00) N = **6.000N**	(24.00 / 4.42) N = **5.430N**
	Min.: 23.76 MHz	(23.76 / 3.52) N = **6.750N**	(23.76 / 4.00) N = **5.940N**	(23.76 / 4.42) N = **5.376N**

As shown in Table 5.3, in the case of MOCO is 3.52MHz and HOCO is 24.24MHz, the count value of the measurement result goes to the most. Therefore, N can be calculated by the following expressions.

$$(24.24 / 3.52)\ N < 2^{16} - 1$$

$$N < 65535 * 3.52 / 24.24$$

$$N < 9516.6$$

$$\therefore\ N\ (Max) = 9516$$

Here, the TDR01 register sets the half cycle of the object waveform to be measured. Therefore, the set value is as follows. Please confirm the r_it8bit0_channel0_interrupt function.

$$\text{Value set to TDR01} = N/2-1$$

$$= \mathbf{4757\ (0x1295)}$$

(2) Baud rate correction

UART of this application sets MOCO driven as fCLK = 4MHz and fMCK = CK00= fCLK/ 2= 2MHz. Because the target baud rate is 9600bps, the SDR [15:9] value for upper 7 bits of SDR register is calculated by the following expression.

$$SDR[15:9] + 1 = f_{MCK} / (2 * 9600)$$
$$= 2\ MHz / (2 * 9600)$$

In addition, considering the frequency accuracy of MOCO, the expression is as follows.

$$SDR\ 15:9 + 1 = 2\ MHz * (ideal\ count\ value/\ capture\ value\ of\ CH3) / (2 * 9600)$$
$$= 2\ MHz * (6.000 * N/\ capture\ value\ of\ CH3) / (2 * 9600)$$
$$= 2\ MHz * (6 * \quad 9516/\ capture\ value\ of\ CH3) / (2 * 9600)$$
$$= 5947500/\ (capture\ value\ of\ CH3)$$

To simplify the calculation, the right denominator and the molecule are divided respectively by 256(= 2^8), the final expression is as follows. In this sample program, it is described by the r it8bit0_channel0_interrupt function. Please confirm that the left side value SDR[15:9].is described as local variable k.

$$= 23232/\ (eight\ high\ rank\ bits\ of\ capture\ value\ of\ CH3)$$

5.3 Set list of option byte

Table 5.4 shows the optional byte setting.

Table 5.4 Optional byte setting

Address	Set value	Content
000C0H/010C0H	1110 1111B	Watch dog timer operation prohibition (count stop after reset)
000C1H/010C1H	0111 1111B	LVD reset mode, 2.81V(2.76 to 2.87V)
000C2H/010C2H	1110 0000B	HS mode, HOCO:24MHz
000C3H/010C3H	1000 0100B	Enables the on-chip debugging

5.4 Constant list

Table 5.5 shows the constant used by the sample code.

Table 5.5 Constant used by sample code

Constant name	Set value	Content
MessageOK[4]	"OK¥r¥n"	Return message when “T" is received
Messageok[4]	"ok¥r¥n"	Return message when “t" is received
MessageUC[4]	"UC¥r¥n"	Return message when receiving other than “T" or "t"
MessageFE[4]	"FE¥r¥n"	Return message at flaming error
MessagePE[4]	"PE¥r¥n"	Return message at parity error
MessageOE[4]	"OE¥r¥n"	Return message at overrun error

5.5 Variable list

Table 5.6 shows the global variable.

Table 5.6 Global variable

Type	Variable identifier	Content	Used function
uint8_t	g_uart0_rx_buf	Receive data buffer	main
uint8_t*	gp_uart0_tx_address	Transmission data pointer	R_UART0_Send、 R_uart0_interrupt_send
uint16_t	g_uart0_tx_count	Transmission data number counter	R_UART0_Send、 r_uart0_interrupt_send
uint8_t*	gp_uart0_rx_address	Receive data pointer	R_UART0_Receive、 r_uart0_interrupt_receive、 r_uart0_interrupt_error
uint16_t	g_uart0_rx_count	Receive data number counter	R_UART0_Receive、 r_uart0_interrupt_receive
uint16_t	g_uart0_rx_length	Receive data number	R_UART0_Receive、 r_uart0_interrupt_receive
uint8_t	g_valid_measure	Interrupt generation times of TAU0 channel 3	r_it8bit0_channel0_interrupt r_tau0_channel3_interrupt
MDSTATUS	g_uart0_tx_end	UART transmission processing complete flag	main
uint8_t	g_uart0_after_adjustment	Flag immediately after passage of 8-bit interval timer interrupt function	main r_it8bit0_channel0_interrupt
uint32_t	g_tau0_ch3_width	Result of a measurement at one square wave cycle	r_tau0_channel3_interrupt r_it8bit0_channel0_interrupt
uint8_t	g_uart0_rx_err	UART communication status	main r_uart0_callback_receiveend r_uart0_callback_error

5.6 Function list

Table 5.7 shows the function list.

Table 5.7 Function list

Function name	Outline
main	The main function
R_MAIN_UserInit	The main user initialization function
R_UART0_Receive	UART0 reception status initialization function
R_UART0_Start	UART0 operation start function
R_UART0_Send	UART0 data transmission function
R_UART0_Stop	UART0 operation stop function
r_uart0_interrupt_receive	UART0 reception complete interrupt function
r_uart0_interrupt_error	UART0 reception error interrupt function
r_uart0_interrupt_send	UART0 transmission complete interrupt function
r_it8bit0_channel0_interrupt	8-bit interval timer interrupt function
r_tau0_channel3_interrupt	TAU Channel 3 count complete interrupt function
R_TAU0_Channel1_Start	TAU Channel 1 operation start function
R_TAU0_Channel3_Start	TAU Channel 3 operation start function
R_TAU0_Channel1_Stop	TAU Channel 1 operation stop function
R_TAU0_Channel3_Stop	TAU Channel 3 operation stop function
R_IT8Bit0_Channel0_Start	8-bit interval timer operation start function
R_IT8Bit0_Channel0_Stop	8-bit interval timer operation stop function

5.7 Function specification

The following shows the sample code function specification.

Function name Main	
Outline	The main function
Header	r_cg_macrodriver.h、r_cg_cgc.h、r_cg_port.h、r_cg_tau.h、r_cg_it8bit.h、r_cg_sau.h、intrinsics.h
Declaration	-
Explanation	After the main user initialization function is executed, MCU enters HALT state and waits the UART reception interrupt. When the UART reception complete interrupt is generated, UART transmission is processed for response. When the factor for the HALT release is not UART reception, UART transmission will not be processed and MCU returns to HALT state again.
Parameter	None
Return value	None

Function name R_MAIN_UserInit	
Outline	The main user initialization function
Header	r_cg_macrodriver.h、r_cg_cgc.h、r_cg_port.h、r_cg_tau.h、r_cg_it8bit.h、r_cg_sau.h、intrinsics.h
Declaration	void R_MAIN_UserInit(void);
Explanation	After UART0 and LED are initialized, enable the UART0 and 8-bit interval timer operation. at last, enable interrupt by the EI instruction.
Parameter	None
Return value	None

Function name R_UART0_Receive		
Outline	UART0 reception status initialization function	
Header	r_cg_macrodriver.h、r_cg_sau.h	
Declaration	MD_STATUS R_UART0_Receive(uint8_t * const rx_buf, uint16_t rx_num);	
Explanation	Set the UART0 reception status.	
Parameter	uint8_t* const rx_buf	:Address in receive data buffer
	uint16_t rx_num	:Size of receive data buffer
Return value	For MD_OK: The reception setting completion.	
	For MD_ARGERROR: The reception setting failure.	

Function name R_UART0_Start	
Outline	UART0 operation start function
Header	r_cg_macrodriver.h、r_cg_sau.h
Declaration	void R_UART0_Start(void);
Explanation	Enable serial array unit channel 0, 1 operation. Let UART in communication standby state..
Parameter	None
Return value	None

Function name R_UART0_Send	
Outline	UART0 data transmission function
Header	r_cg_macrodriver.h、r_cg_sau.h
Declaration	MD_STATUS R_UART0_Send(uint8_t * const tx_buf, uint16_t tx_num);
Explanation	Initialize the UART0 transmission starts the data transmission.
Parameter	uint8_t* const tx_buf :Address in transmission data buffer
	uint16_t tx_num :Size of transmission data buffer
Return value	For MD_OK: The transmission setting completion.
	For MD_ARGERROR: The transmission setting failure.

Function name R_UART0_Stop	
Outline	UART0 operation stop function
Header	r_cg_macrodriver.h、r_cg_sau.h
Declaration	void R_UART0_Stop(void);
Explanation	Stop serial array 0 unit 0 channel 0, 1 operation. Stop UART communication.
Parameter	None
Return value	None

Function name r_uart0_interrupt_receive	
Outline	UART0 reception complete interrupt function
Header	r_cg_macrodriver.h、r_cg_sau.h
Declaration	__interrupt void r_uart0_interrupt_receive(void)
Explanation	The received data is stored in RAM, and the address and the reception times values are updated.
Parameter	None
Return value	None

Function name r_uart0_interrupt_error	
Outline	UART0 reception error interrupt function
Header	r_cg_macrodriver.h、r_cg_sau.h
Declaration	__interrupt void r_uart0_interrupt_error(void)
Explanation	Store the reception data in RAM Response the corresponding data by the detected error from the r_uart0_callback_error function.
Parameter	None
Return value	None

Function name r_uart0_interrupt_send	
Outline	UART0 transmission complete interrupt function
Header	r_cg_macrodriver.h、r_cg_sau.h
Declaration	__interrupt void r_uart0_interrupt_send(void)
Explanation	Transmit the data update the pointer and the counter. In the case of transmission data doesn't remain, process the transmission completion.
Parameter	None
Return value	None

Function name r_it8bit0_interrupt	
Outline	8-bit interval timer interrupt function
Header	r_cg_macrodriver.h、r_cg_it8bit.h、r_cg_sau.h、r_cg_tau.h、intrinsics.h
Declaration	__interrupt void r_it8bit0_channel0_interrupt(void)
Explanation	The function that executes the baud rate correction periodically (It is every 1 second in this application). Disable the UART communication first. Start timer array unit 0 channel 1 and channel 3. Channel 3 measures the pulse width of object square waveform generated from channel 1. Calculate the result backward and modify the clock division to make it as close to the ideal value as possible. After that, enable the UART communication and return to the UART communication standby state again.
Parameter	None
Return value	None

Function name R_tau0_channel3_interrupt	
Outline	TAU channel 3 count complete interrupt function
Header	r_cg_macrodriver.h、r_cg_tau.h
Declaration	__interrupt void r_tau0_channel3_interrupt(void)
Explanation	Store the result of the pulse width measured by the timer array unit channel 3 in the global variable.
Parameter	None
Return value	None

Function name R_TAU0_Channel1_Start	
Outline	TAU Channel 1 operation start function
Header	r_cg_macrodriver.h、r_cg_tau.h
Declaration	void R_TAU0_Channel1_Start(void);
Explanation	Enable timer array unit channel 1 operation. Start a square wave output.
Parameter	None
Return value	None

Function name R_TAU0_Channel3_Start	
Outline	TAU Channel 3 operation start function
Header	r_cg_macrodriver.h、r_cg_tau.h
Declaration	void R_TAU0_Channel3_Start(void);
Explanation	Enable timer array unit channel 1 operation. Start the pulse width measurement.
Parameter	None
Return value	None

Function name R_TAU0_Channel1_Stop	
Outline	TAU Channel 1 operation stop function
Header	r_cg_macrodriver.h、r_cg_tau.h
Declaration	void R_TAU0_Channel1_Stop(void);
Explanation	Stop timer array unit channel 1.
Parameter	None
Return value	None

Function name R_TAU0_Channel3_Stop	
Outline	TAU Channel 3 operation stop function
Header	r_cg_macrodriver.h、r_cg_tau.h
Declaration	void R_TAU0_Channel3_Stop(void);
Explanation	Stop timer array unit channel 3.
Parameter	None
Return value	None

Function name R_IT8Bit0_Channel0_Start	
Outline	8-bit interval timer operation start function
Header	r_cg_macrodriver.h、r_cg_it8bit.h
Declaration	void R_IT8Bit0_Channel0_Start(void);
Explanation	Enable 8-bit interval timer operation.
Parameter	None
Return value	None

Function name R_IT8Bit0_Channel0_Stop	
Outline	8-bit interval timer operation stop function
Header	r_cg_macrodriver.h、r_cg_it8bit.h
Declaration	void R_IT8Bit0_Channel0_Stop(void);
Explanation	Stop 8-bit interval timer.
Parameter	None
Return value	None

5.8 Flow chart

Figure 5.1 shows the application note whole flow.

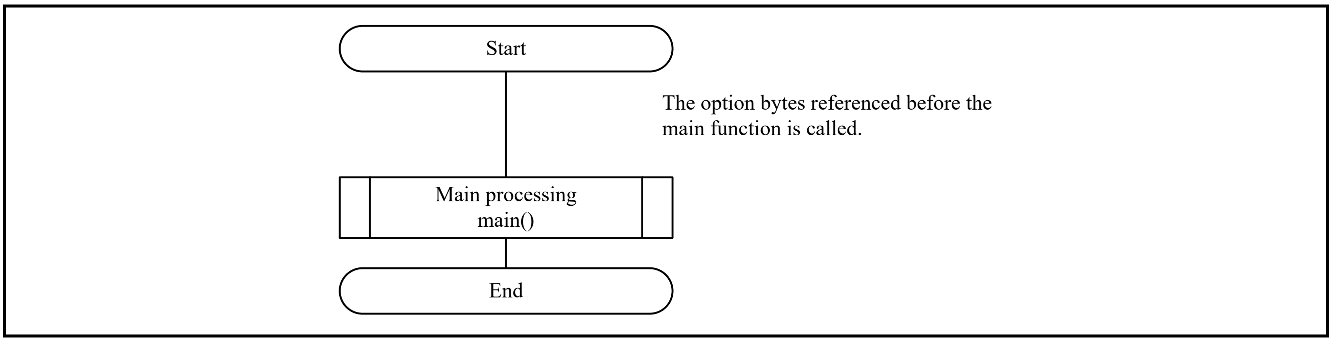


Figure 5.1 Whole flow

5.8.1 The main function

Figure 5.2 and Figure 5.3 show main function flowcharts.

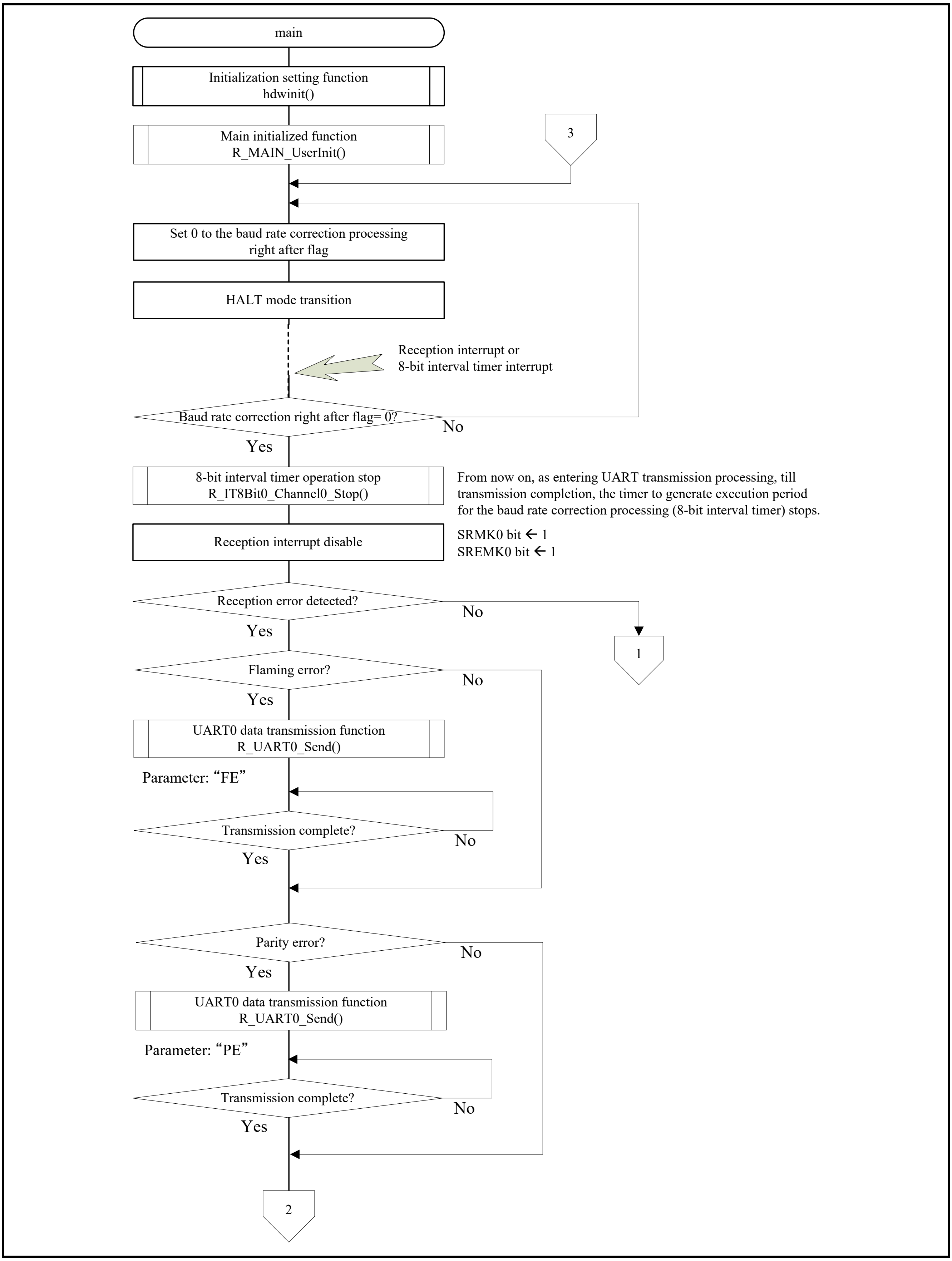


Figure 5.2 Main function (1/2)

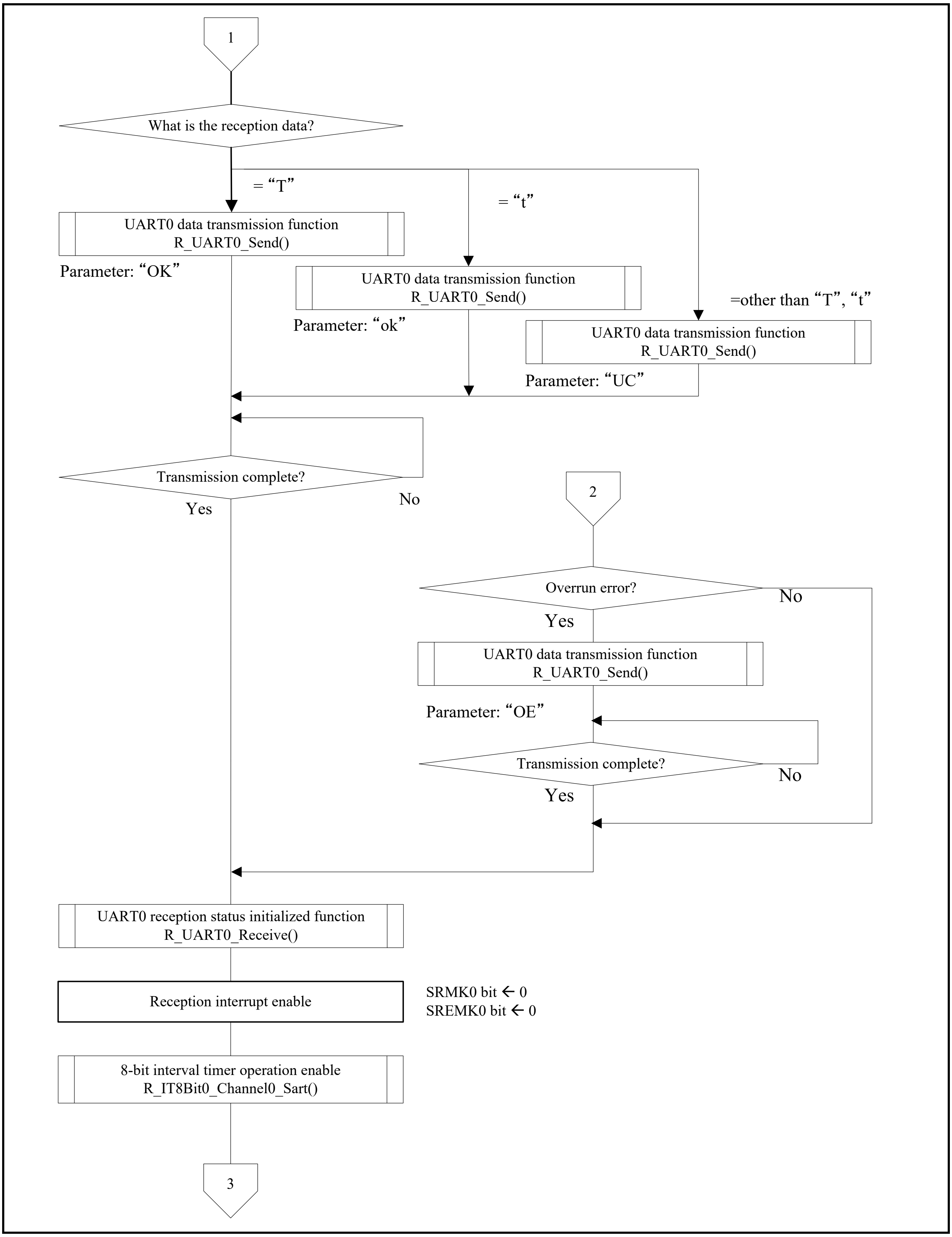


Figure 5.3 Main processing (2/2)

5.8.2 Initialization

Figure 5.4 shows the initialization function flowchart.

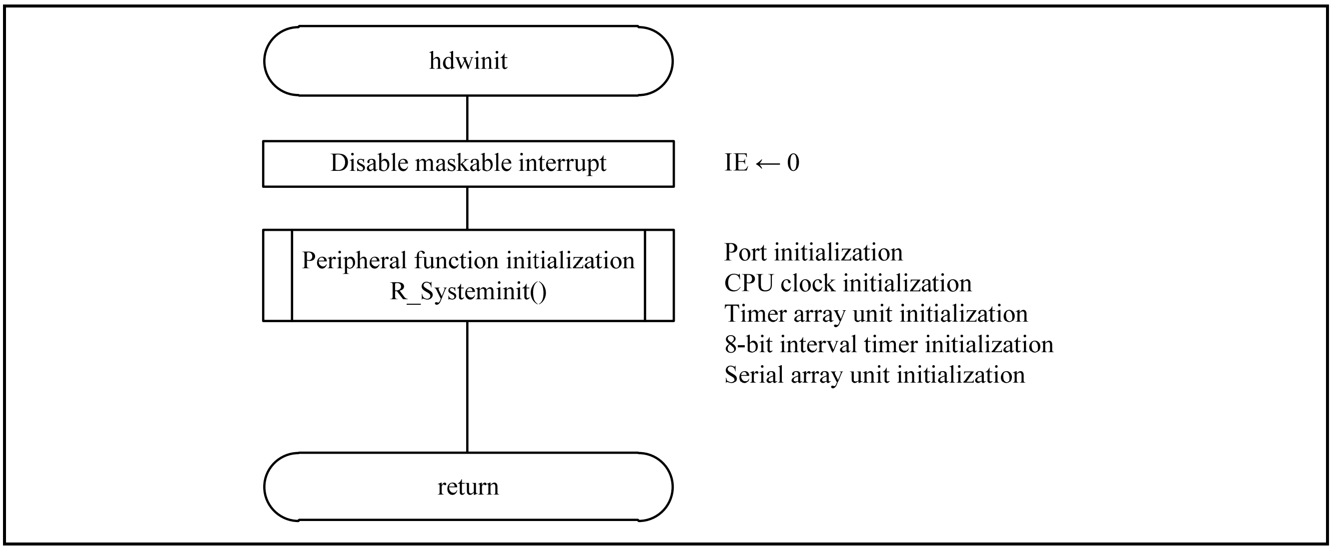


Figure 5.4 Initialization function

5.8.3 Peripheral function initialization

Figure 5.5 shows the peripheral function initialization flowchart.

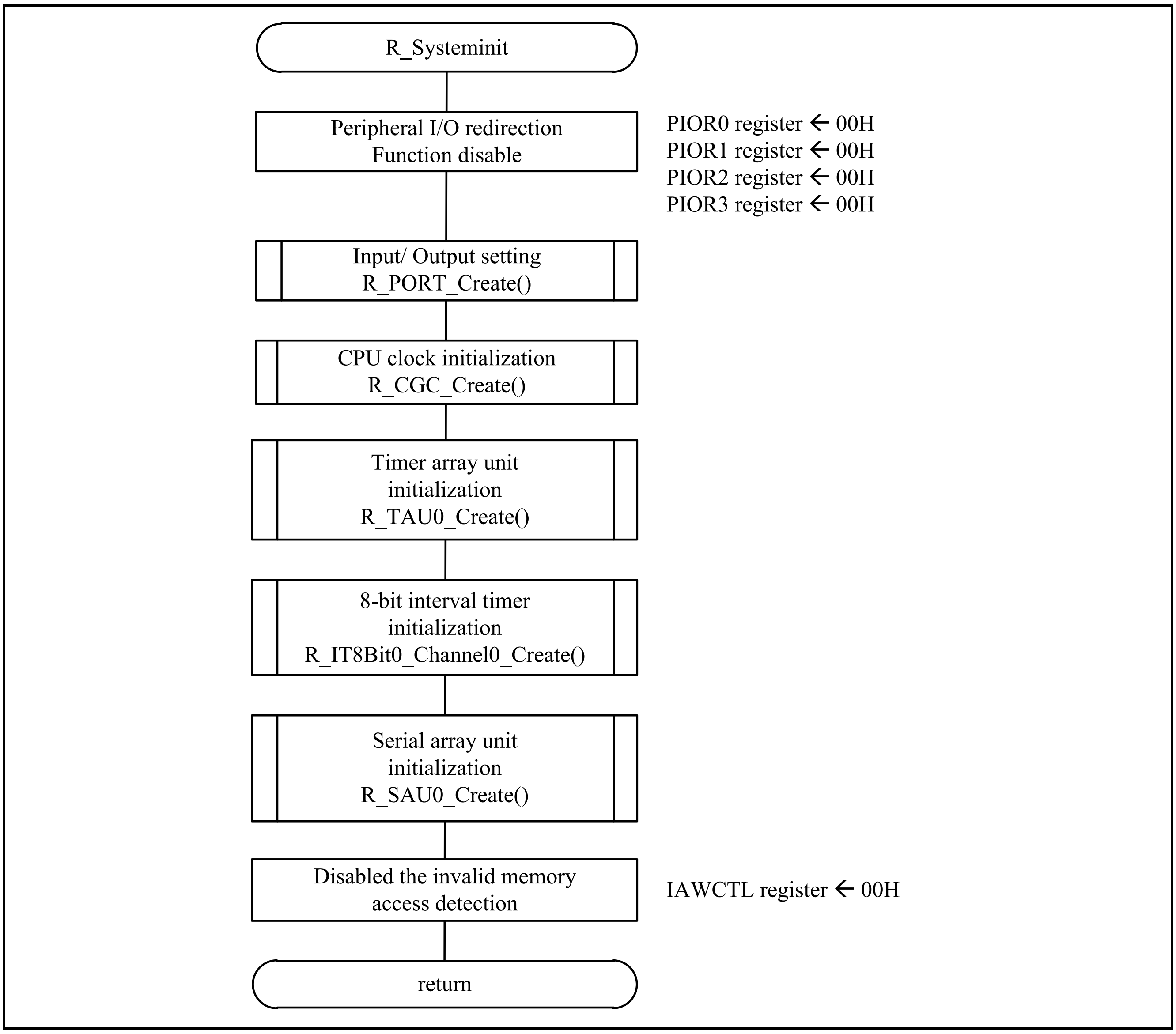


Figure 5.5 Peripheral function initialization function

5.8.4 I/O Port Setup

Figure 5.6 shows the I/O Port Setup flowchart.

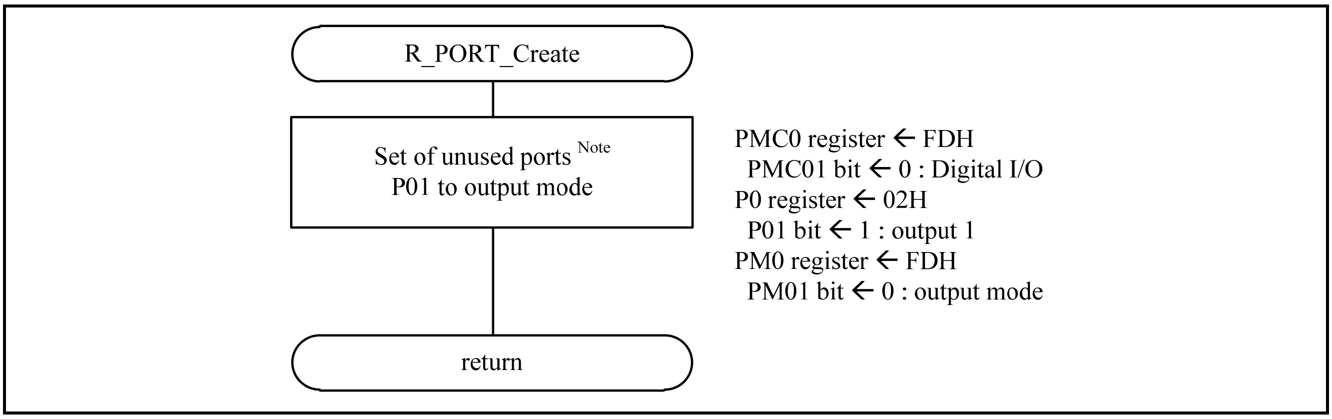


Figure 5.6 I/O Port Setup

Note: Refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575EJ0100) for the configuration of the unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to VDD or VSS via a separate resistor.

5.8.5 CPU clock initialization

Figure 5.7 shows the CPU clock initialization function flowchart.

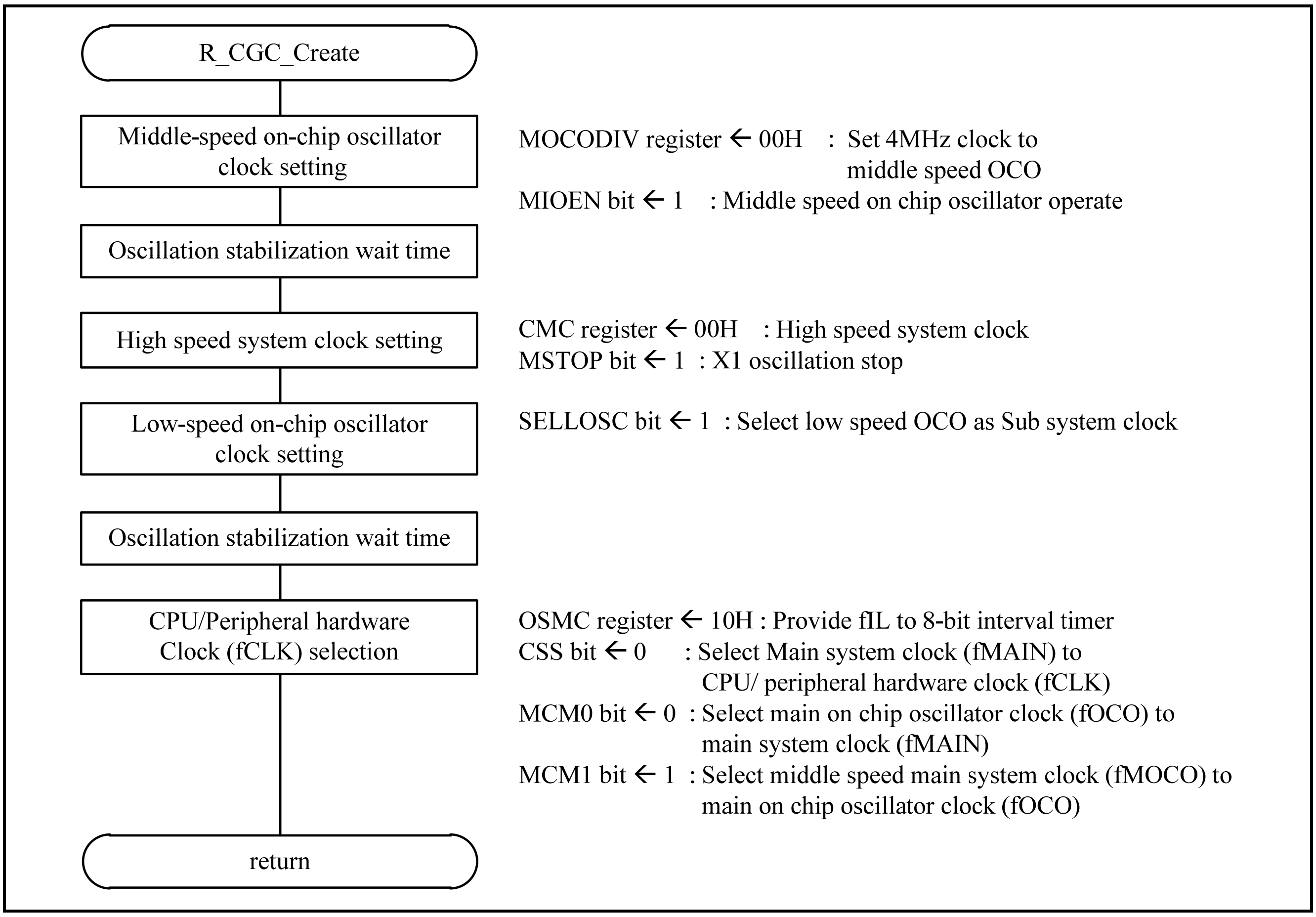


Figure 5.7 CPU clock initialization function

5.8.6 Timer array unit initialization

Figure 5.8, Figure 5.9 shows the timer array unit initialization flowchart.

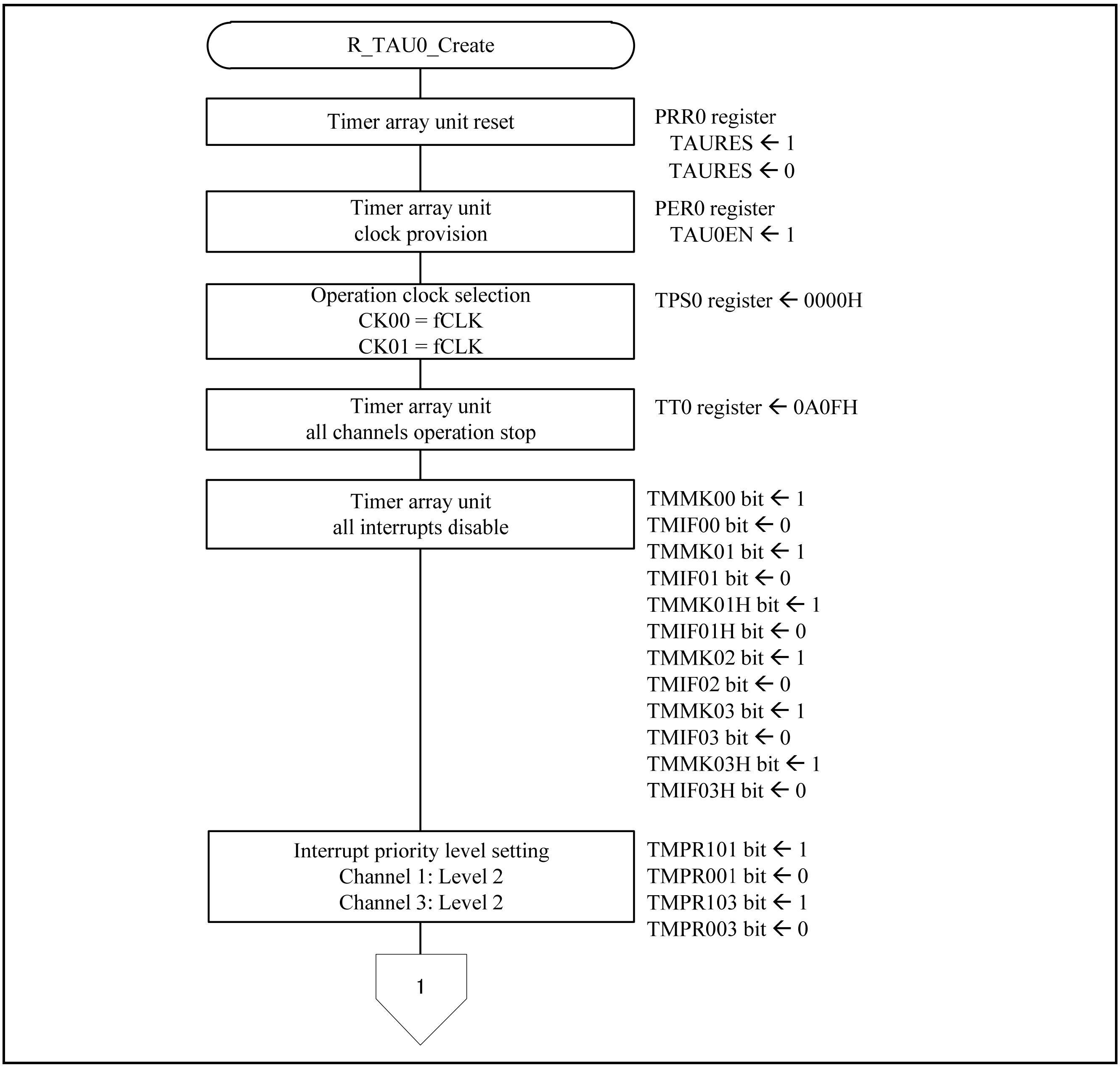


Figure 5.8 Timer array unit initialization(1/2)

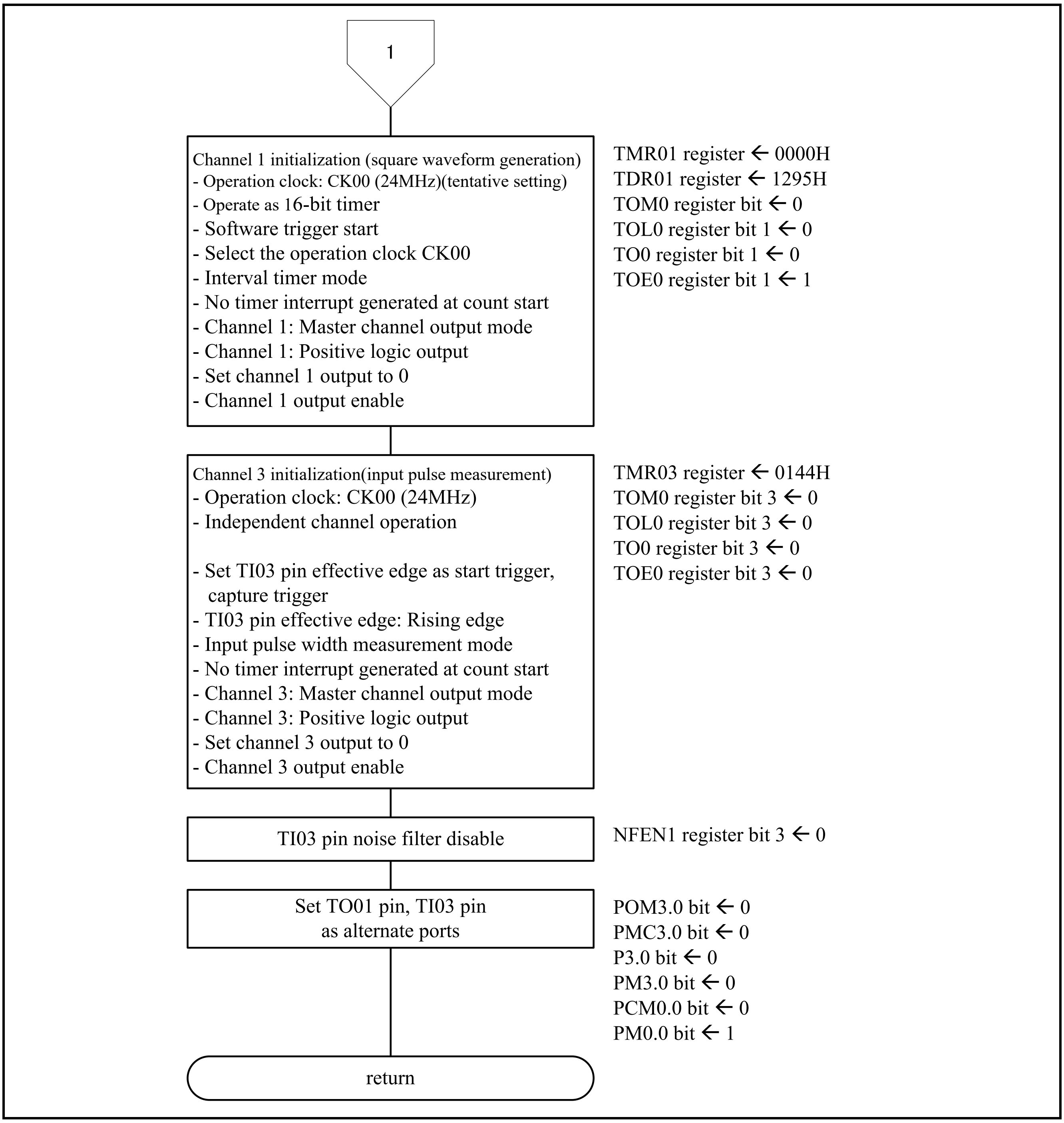


Figure 5.9 Timer array unit initialization(2/2)

5.8.7 8-bit interval timer initialization function

Figure 5.10 shows the 8-bit interval timer initialization function flowchart.

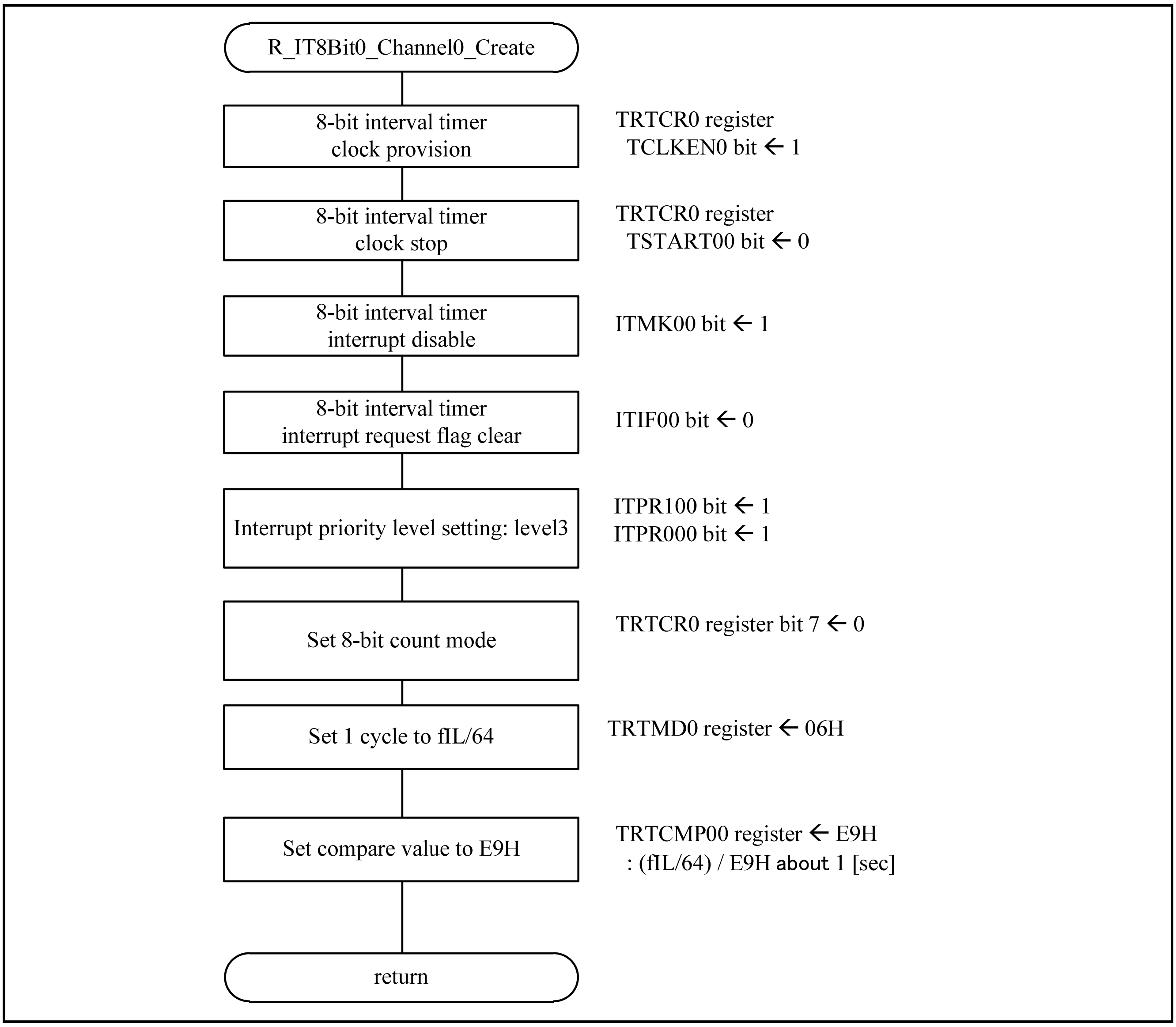


Figure 5.10 8-bit interval timer initialization function

5.8.8 Serial array unit initialization

Figure 5.11 shows the serial array unit initialization function flowchart.

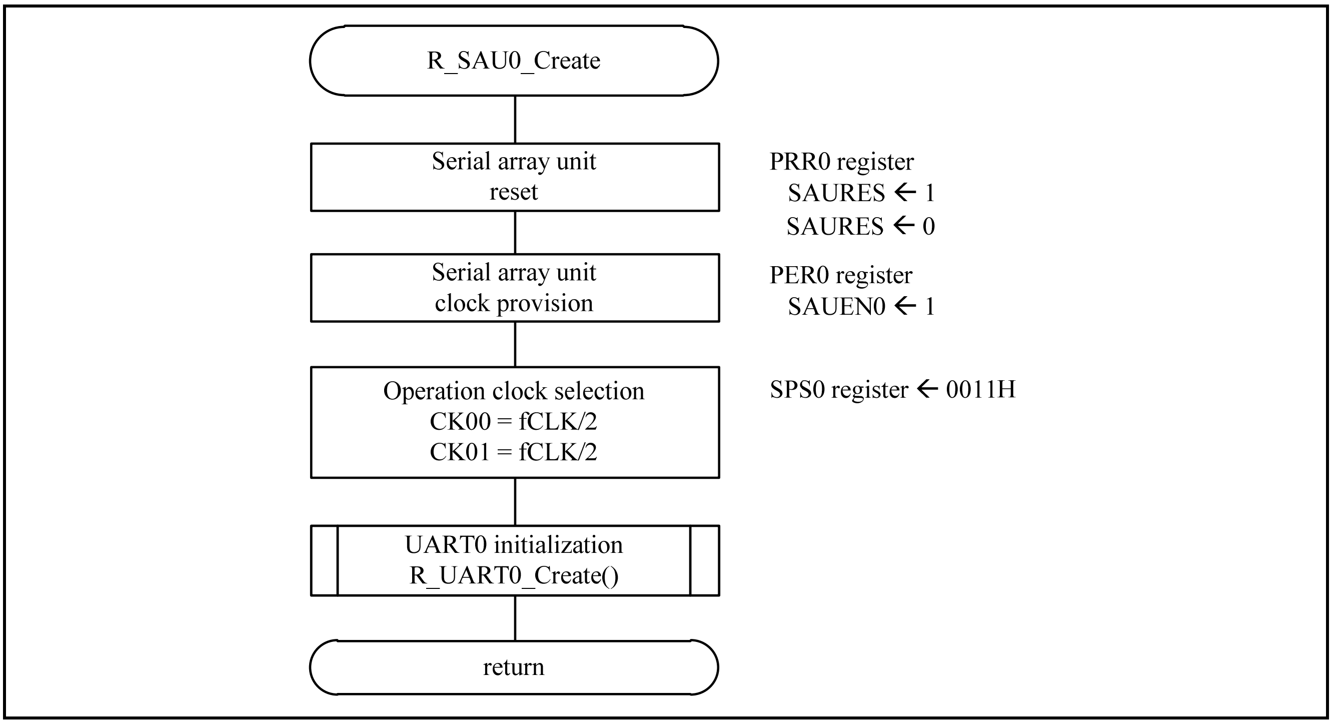


Figure 5.11 Serial array unit initialization function

5.8.9 UART0 initialization

Figure 5.12, Figure 5.13 shows UART0 initialization flowchart.

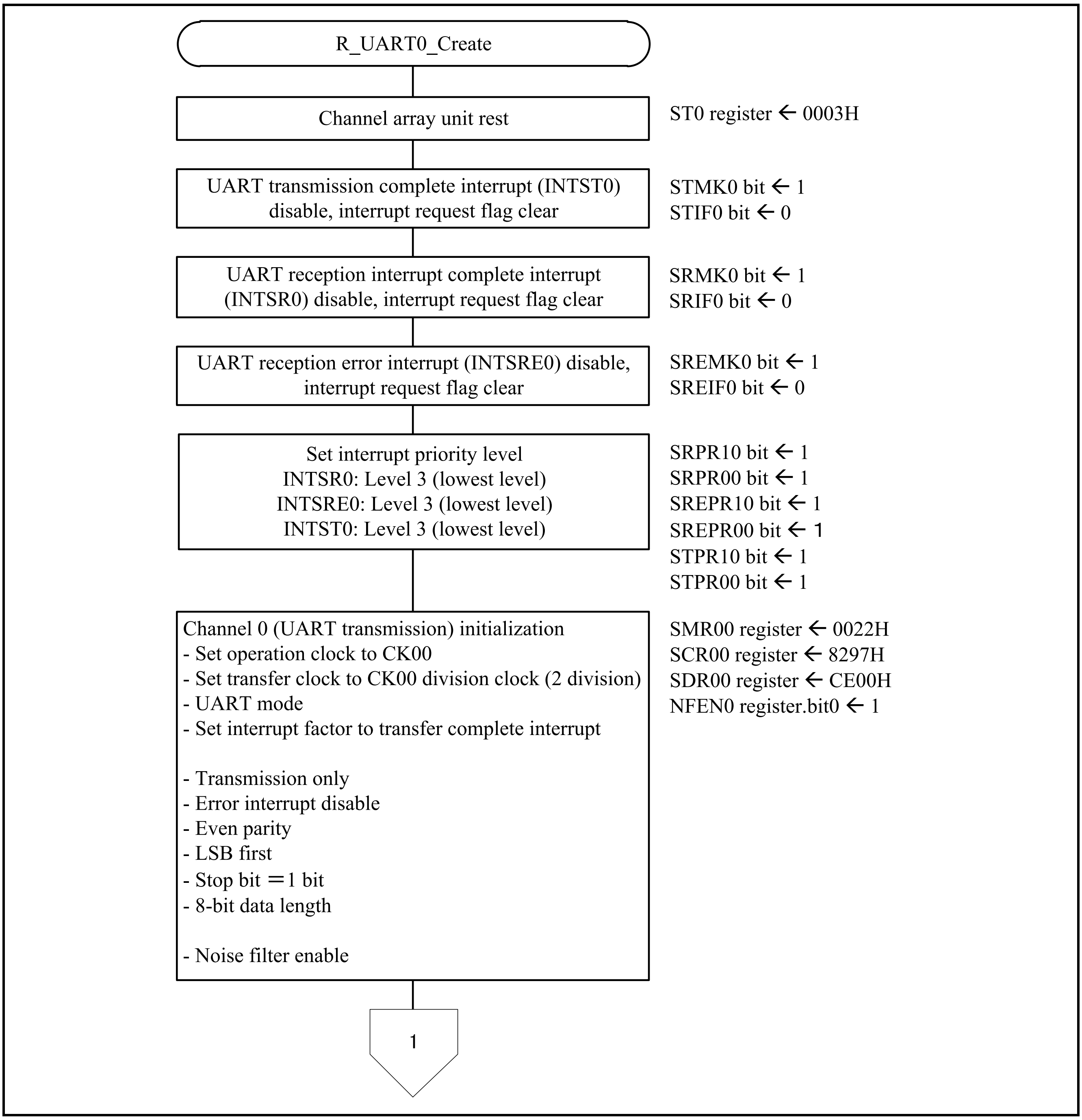


Figure 5.12 UART0 initialization(1/2)

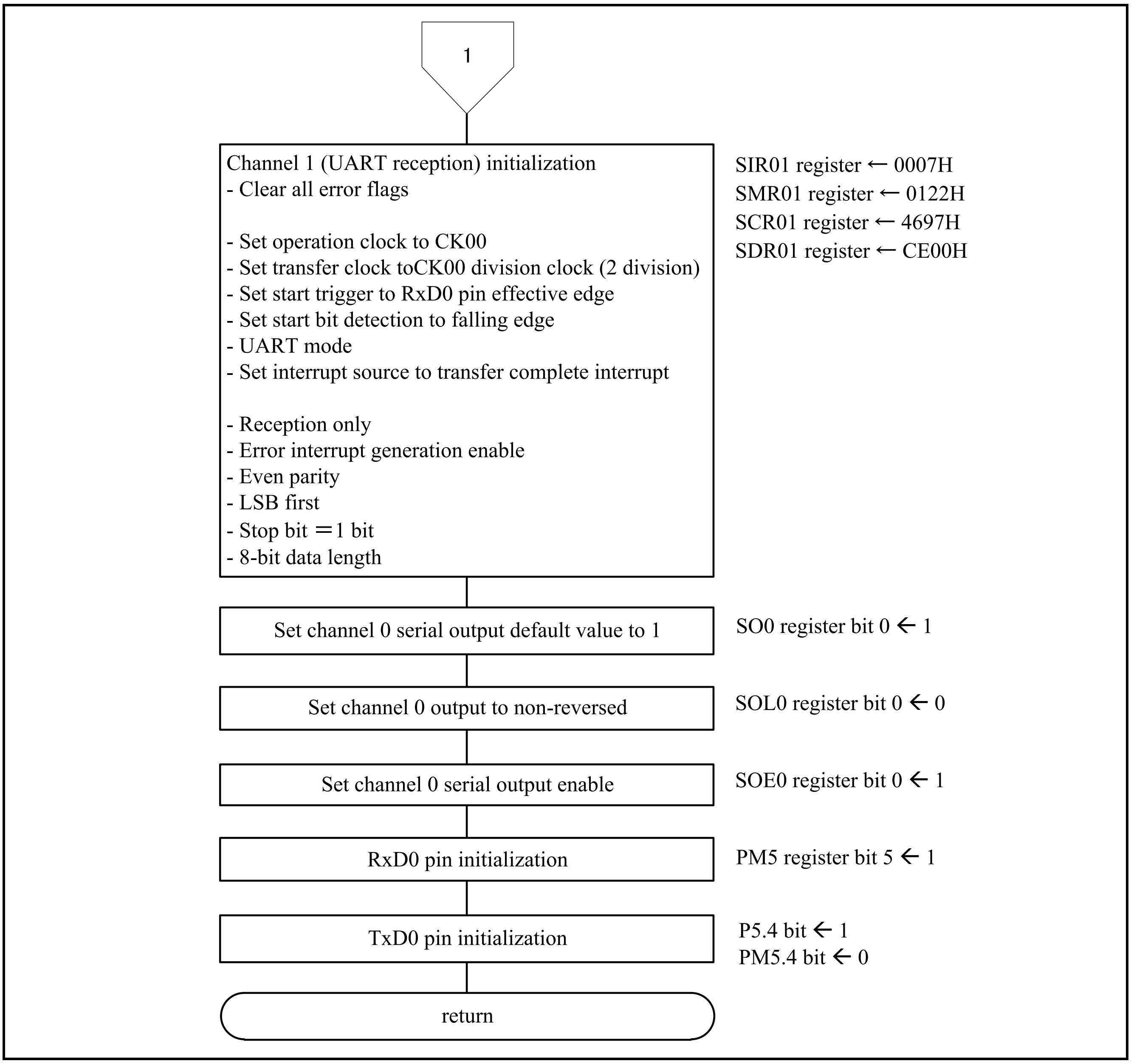


Figure 5.13 UART0 initialization(2/2)

5.8.10 The main user initialization

Figure 5.14 shows the main user initialization function flowchart.

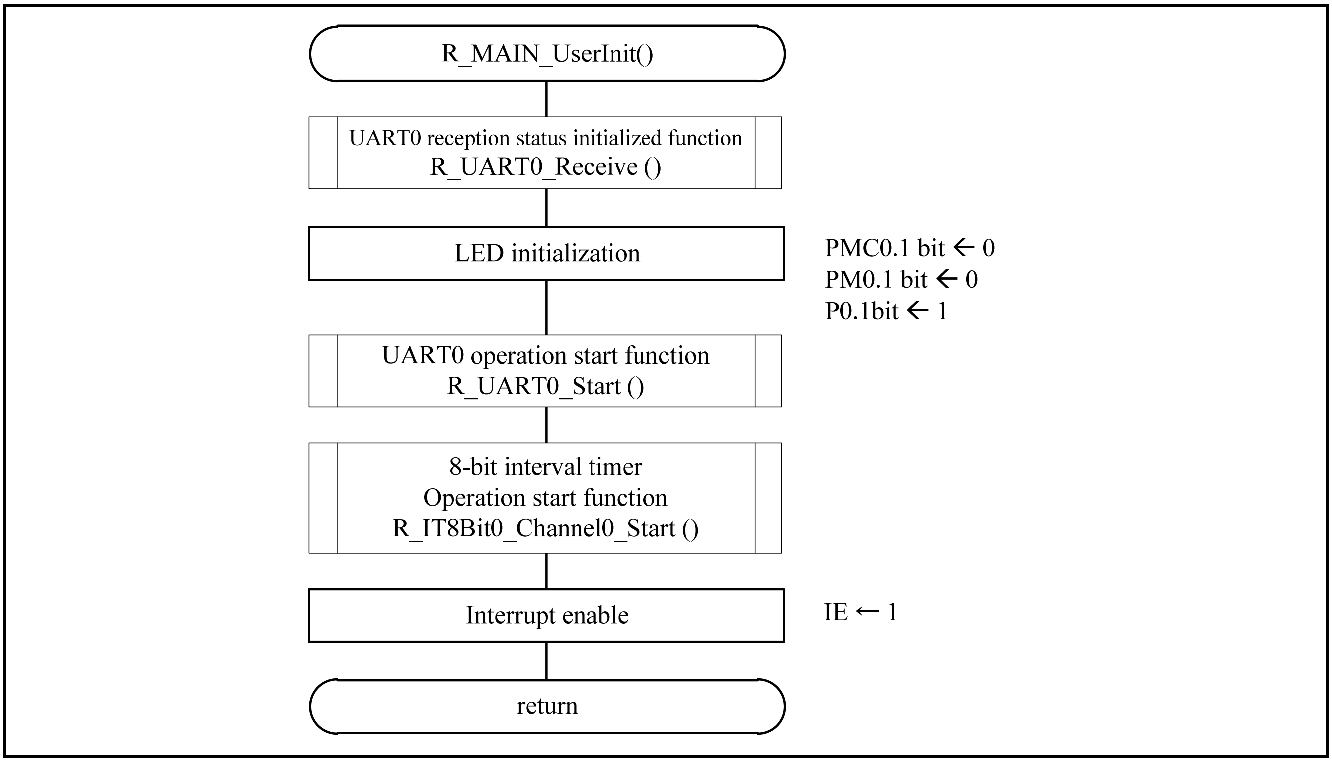


Figure 5.14 Main user initialization function

5.8.11 UART0 reception status initialization

Figure 5.15 shows the UART0 reception status initialization function flowchart.

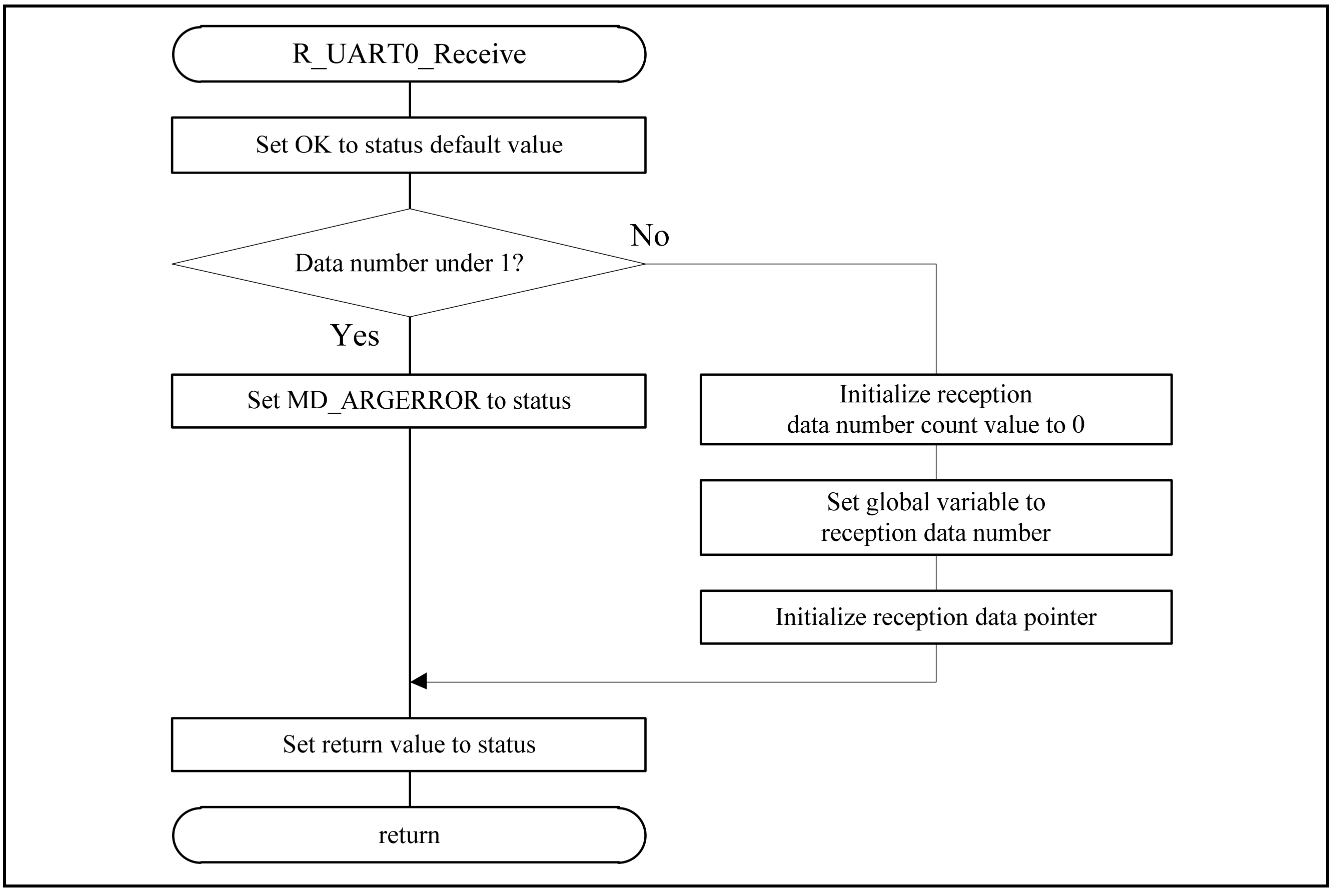


Figure 5.15 UART0 reception status initialization function

5.8.12 UART0 operation start function

Figure 5.16 shows UART0 operation start function flowchart.

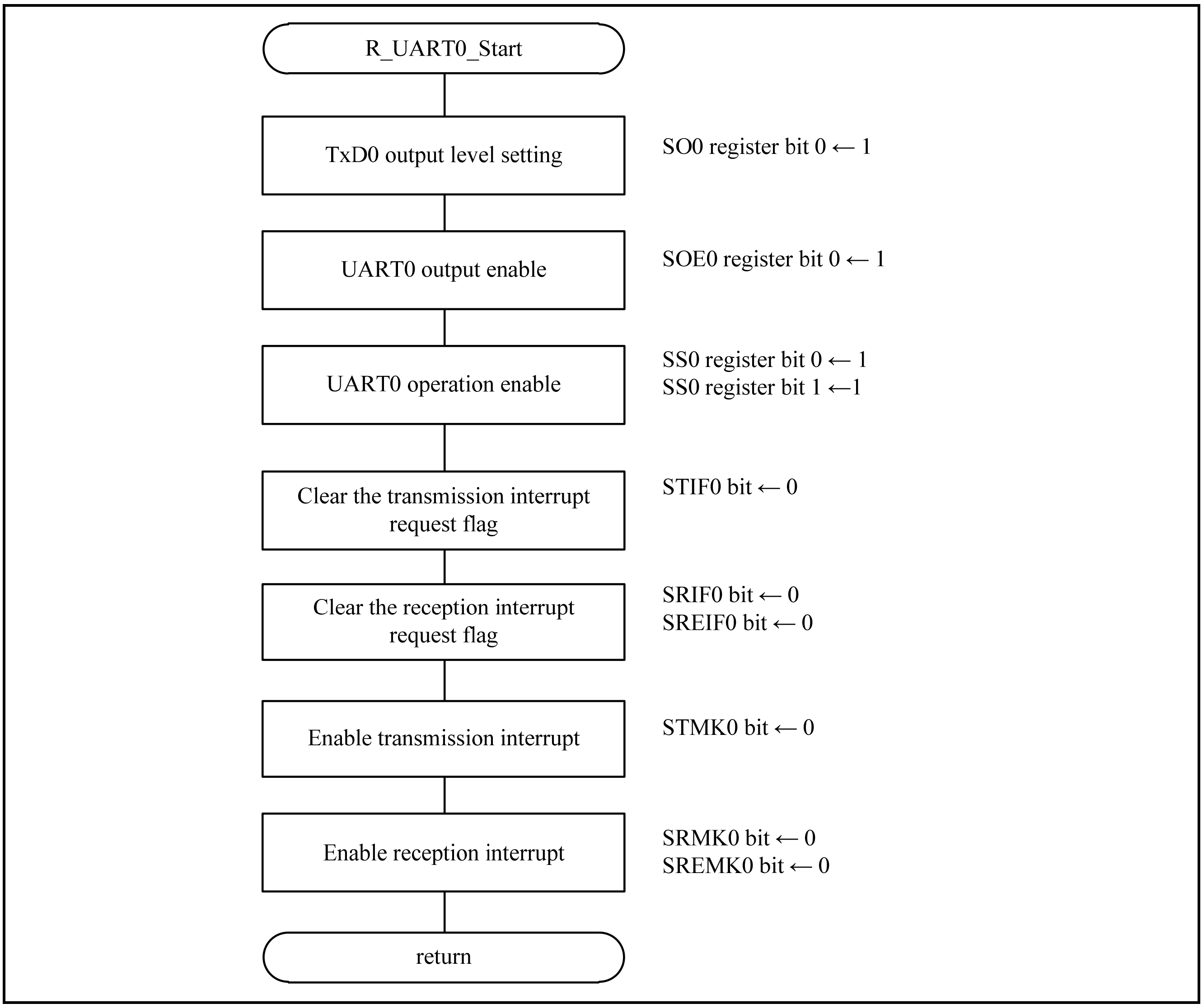


Figure 5.16 UART0 operation start function

5.8.13 UART0 data transmission

Figure 5.17 shows the UART0 data transmission function flowchart.

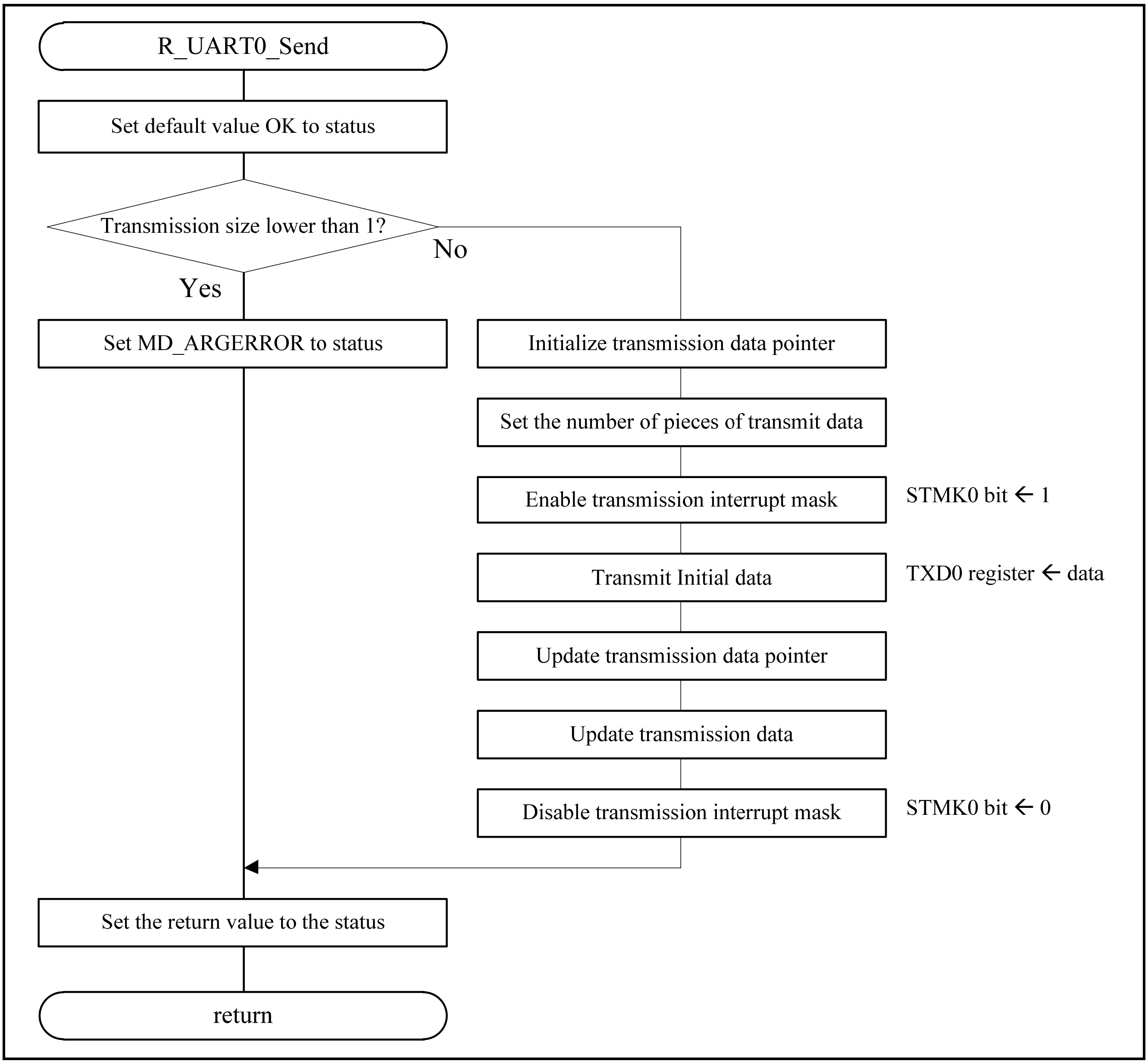


Figure 5.17 UART0 data transmission function

5.8.14 UART0 operation stop

Figure 5.18 shows the UART0 operation stop function flowchart.

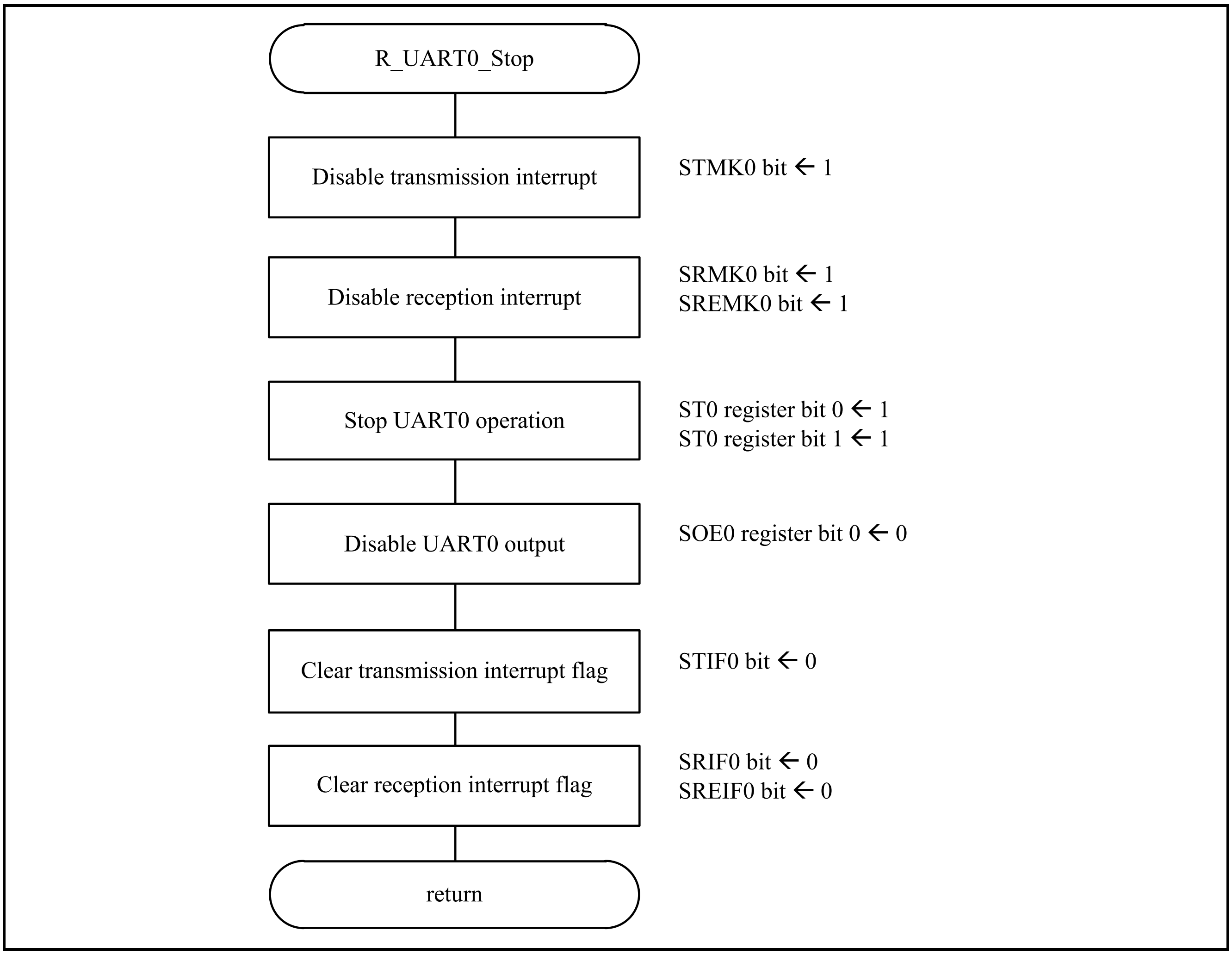


Figure 5.18 UART0 operation stop function

5.8.15 UART0 reception completion interrupt

Figure 5.19 shows the UART0 reception complete interrupt function flowchart.

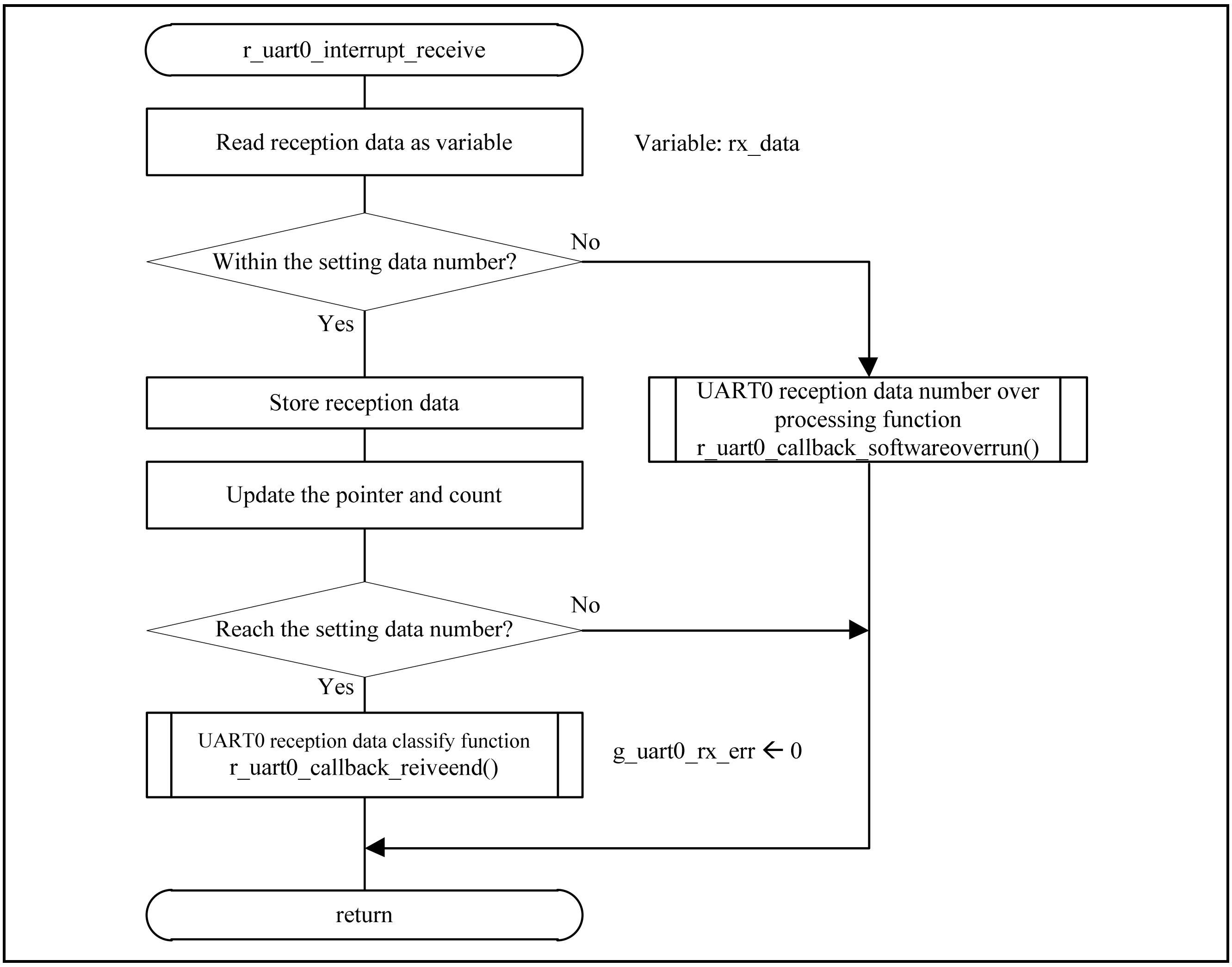


Figure 5.19 UART0 reception complete interrupt function

5.8.16 UART0 reception error interrupt

Figure 5.20 shows the UART0 reception error interrupt function flowchart.

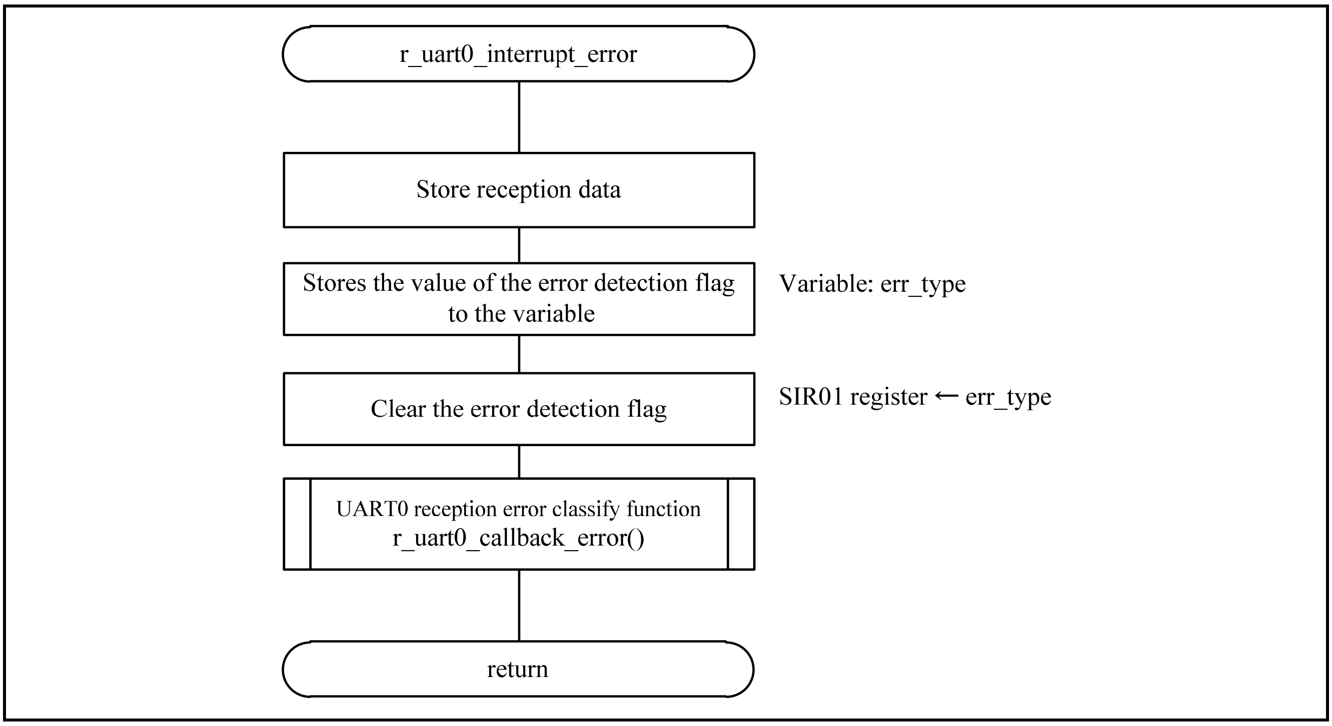


Figure 5.20 UART0 reception error interrupt function

5.8.17 UART0 transmission complete interrupt

Figure 5.21 shows the UART0 transmission completion interrupt function flowchart.

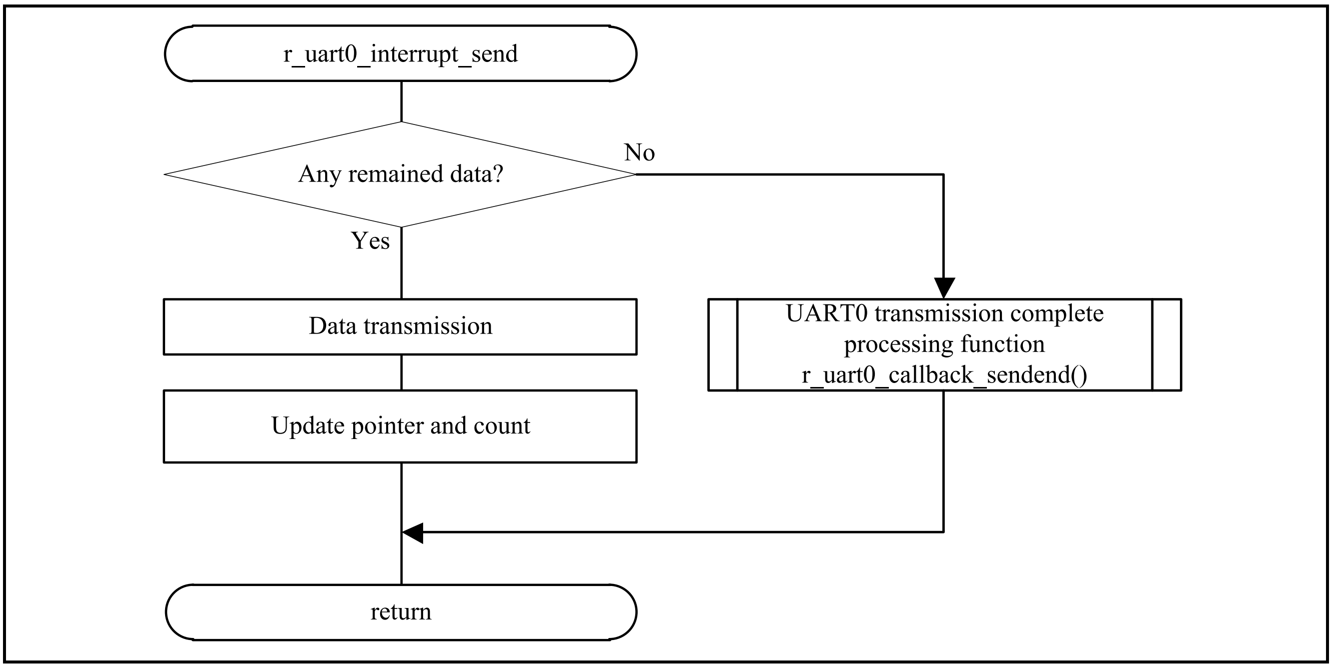


Figure 5.21 UART0 transmission complete interrupt function

5.8.18 8-bit interval timer interrupt

Figure 5.22 and Figure 5.23 show the 8-bit interval timer interrupt function flowcharts.

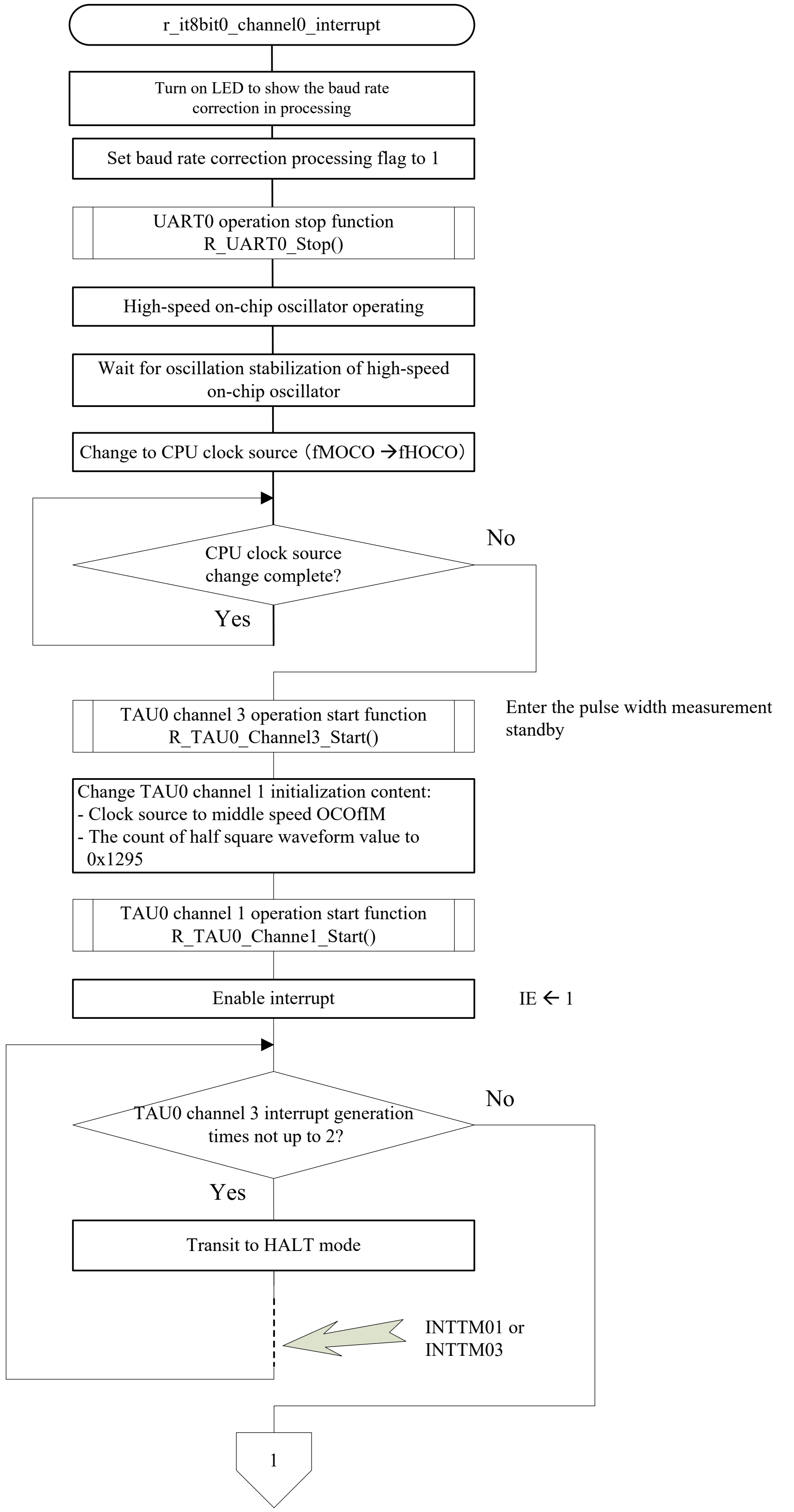


Figure 5.22 8-bit interval timer interrupt function (1/2)

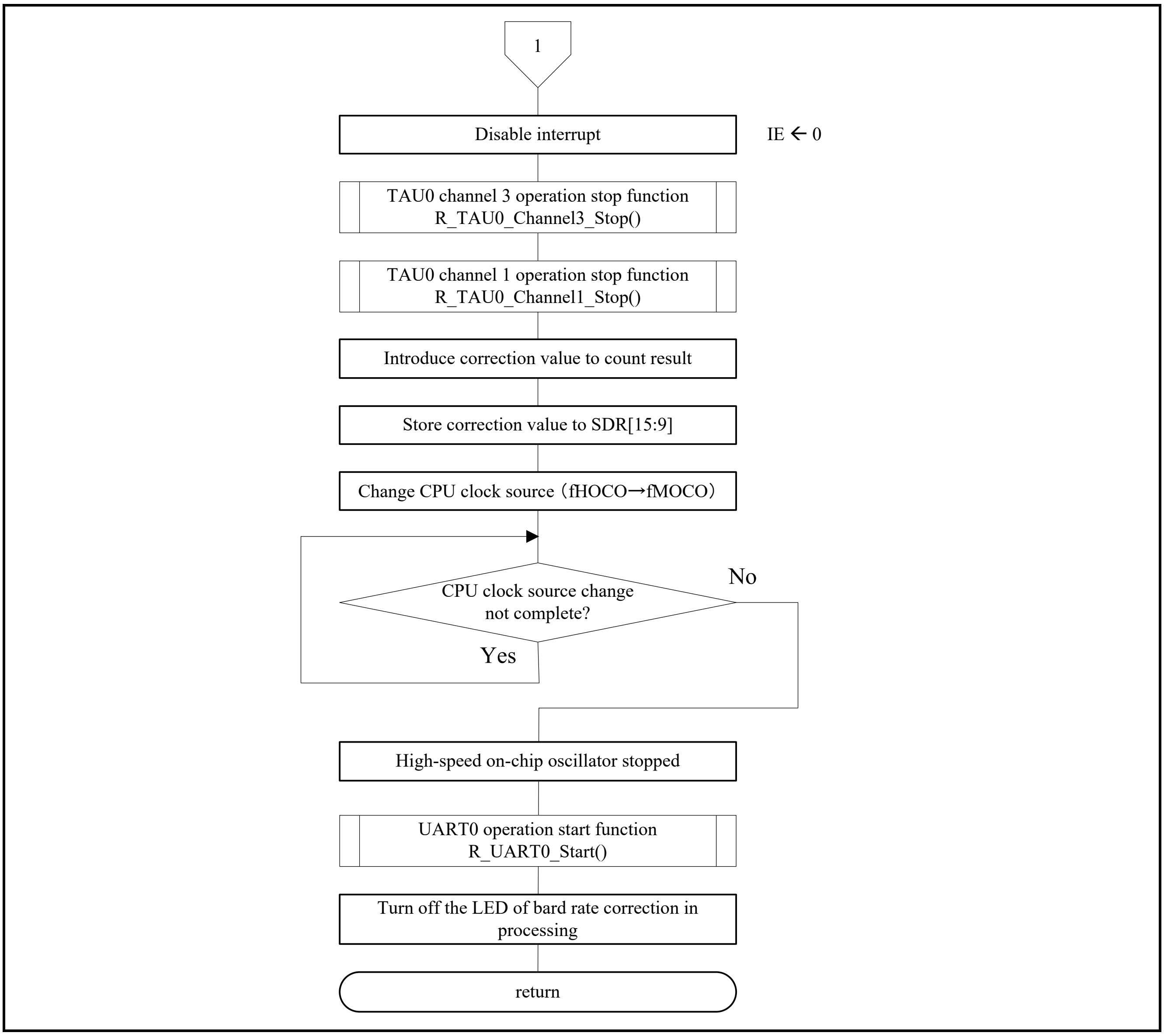
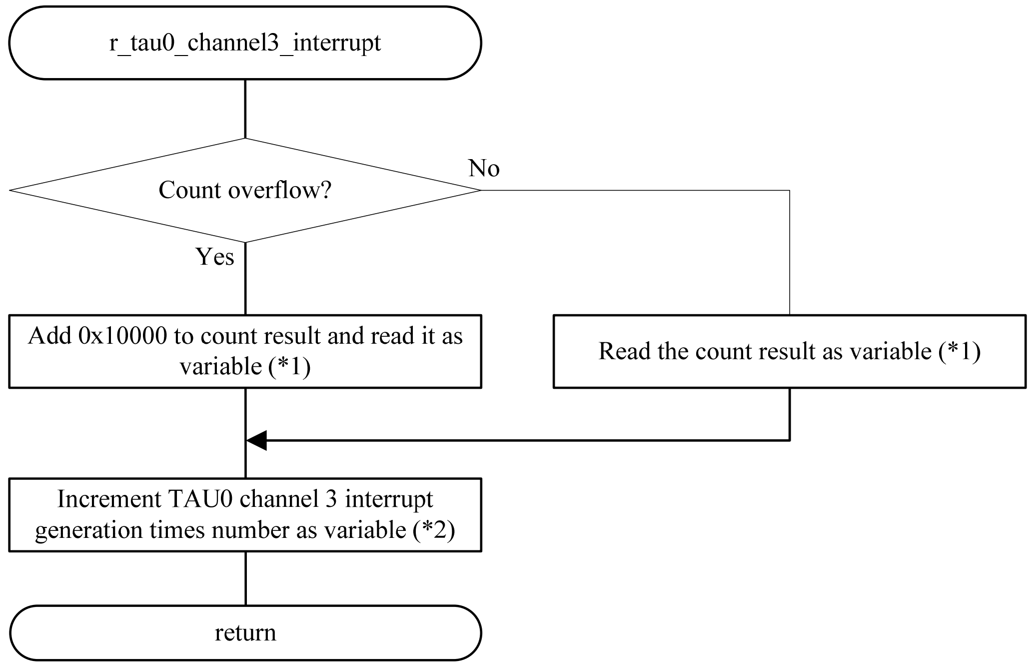


Figure 5.23 8-bit interval timer interrupt function (2/2)

5.8.19 TAU channel 3 count complete interrupt

Figure 5.24 shows the TAU channel 3 count complete interrupt function flowchart.



(*1) variable name: g_tau0_ch3_width

(*2) variable name: g_valid_measure

Figure 5.24 TAU channel 3 count complete interrupt function

6. Sample code

Please get the sample code from Renesas Electronics homepage.

7. Reference document

RL78/G11 User's Manual: Hardware (R01UH0637E)
RL78 family user's manual software (R01US0015E)
(Please get the latest version from Renesas Electronics homepage.)

Technical update
(Please acquire latest information from Renesas Electronics homepage.)

wibesite and support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb. 15, 2017	-	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.

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