R32C/100 Series

Rewriting the Flash Memory Using the Serial Interface (UART)

R01AN1473EJ0100
Rev. 1.00
Mar. 8, 2013

Abstract

This document describes using the serial interface (UART) in the R32C/100 Series to rewrite the flash memory.
The sample code uses EW1 mode of the CPU rewrite mode to rewrite the flash memory.
In the R32C/118 Group, UART0 to UART8 can be used in asynchronous serial interface mode. The sample code uses UART2. When using a channel other than UART2, refer to the User's Manual: Hardware and rewrite registers associated with UART0 to UART8.

Products

R32C/116 Group
R32C/117 Group
R32C/118 Group

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

Contents

1. Specifications

In this document, the serial interface (UART2) is in asynchronous serial interface mode. The MCU receives control commands and write data from the master device. Depending on the control command received, the flash memory is either erased or written.
Control commands transmitted by the master device are either erase commands or program commands. The program command includes up to 256 bytes of data to write to the flash memory.
In the sample code, after erase or write processing is performed on the flash memory, a processing result is transmitted to the master device.

Table 1.1 lists the Peripheral Functions and Their Applications. Figure 1.1 shows a Connection Example.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
Flash memory	Executes programming or block erasing
Serial interface (UART2)	Communicates with the master device
Timer A0	Timer for receive processing timeout detection

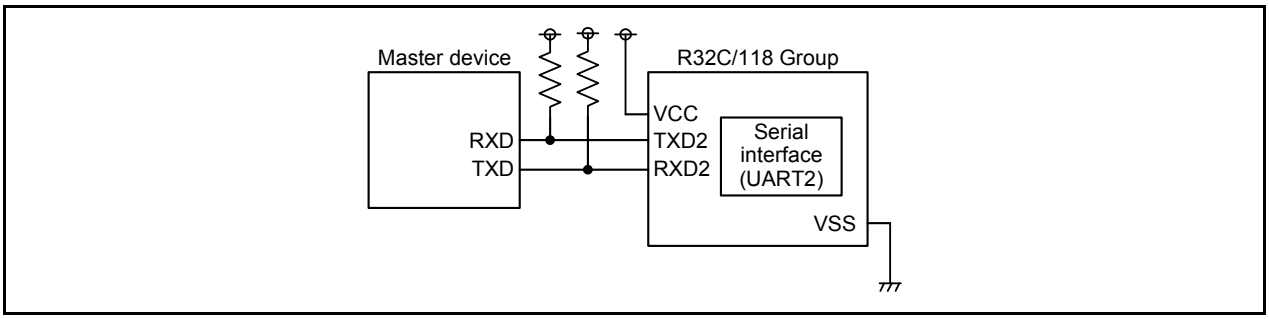


Figure 1.1 Connection Example

2. Operation Confirmation Conditions

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

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	R5F64189DFD (R32C/118 Group)
Operating frequencies	• XIN clock: 16 MHz • PLL clock: 100 MHz • Base clock: 50 MHz • CPU clock: 50 MHz • Peripheral bus clock: 25 MHz • Peripheral clock: 25 MHz
Operating voltage	5V
Integrated development environment	Renesas Electronics High-performance Embedded Workshop Version 4.09
C compiler	Renesas Electronics R32C/100 Series C Compiler V.1.02 Release 01
	Compile options -D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300 -DVECTOR_ADR=0x0FFFFFBDC -c -finfo -dir “$(CONFIGDIR)” Default setting is used in the integrated development environment.
Operating mode	Single-chip mode
Sample code version	1.00
Board used	Renesas Starter Kit for R32C/118 (device part no.: R0K564189S000BE)

3. Reference Application Note

An application note associated with this application note is listed below. Refer to this application note for additional information.

- R32C/100 Series Rewriting ROM Area Using EW1 Mode of CPU Rewrite Mode (REJ05B1394)

4. Hardware

4.1 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P7_0/TXD2	Output	Responds to the master device
P7_1/RXD2	Input	Receives control commands and write data from the master device

5. Software

5.1 Operation Overview

In the sample code, after the MCU starts up, it waits to receive a 3-byte control command from the master device.
If the command received is the erase command, block 7 in the flash memory is block erased.
If the command received is the program command, the MCU waits to receive the size, data, and SUM value from the master device. If the sum of the received data matches the received SUM value, the received data is written to block 7 in the flash memory.
If the control command processing is successful, 6Fh ('o') is transmitted to the master device. If the control command processing ends in error, 65h ('e') is transmitted to the master device.
The sample code does not include recovery processing when an error occurs. Add recovery processing to the user program when necessary. In particular, when an overrun error occurs, subsequent reception is not possible.
Table 5.1 lists the Control Commands, and Table 5.2 lists the Conditions for Configuring Communication.

Table 5.1 Control Commands

Control Command Name	Explanation	First to Third Bytes	Fourth to Fifth Bytes	From Sixth Byte		
Program command	Flash memory is programmed	"prg"	Size (2 bytes)	Data (256 bytes max.)	SUM value (2 bytes)	Result (1)
Erase command	Flash memory is erased	"ers"	Result (1)			

Note:
1. The result is transferred from the sample code to the master device. If the program or erase processing is successful, 6Fh ('o') is returned; if the program or erase processing ends in error, 65h ('e') is returned.

Table 5.2 Conditions for Configuring Communication

Item	Setting
Bit rate	38400 bps
Character length	8 bits
Parity	No parity
Stop bit length	1 stop bit
Transmit/receive clock	Internal clock
CTS	Disabled
Bit order	LSB first

5.1.1 Operation Example

Figure 5.1 shows an Example of Operation.

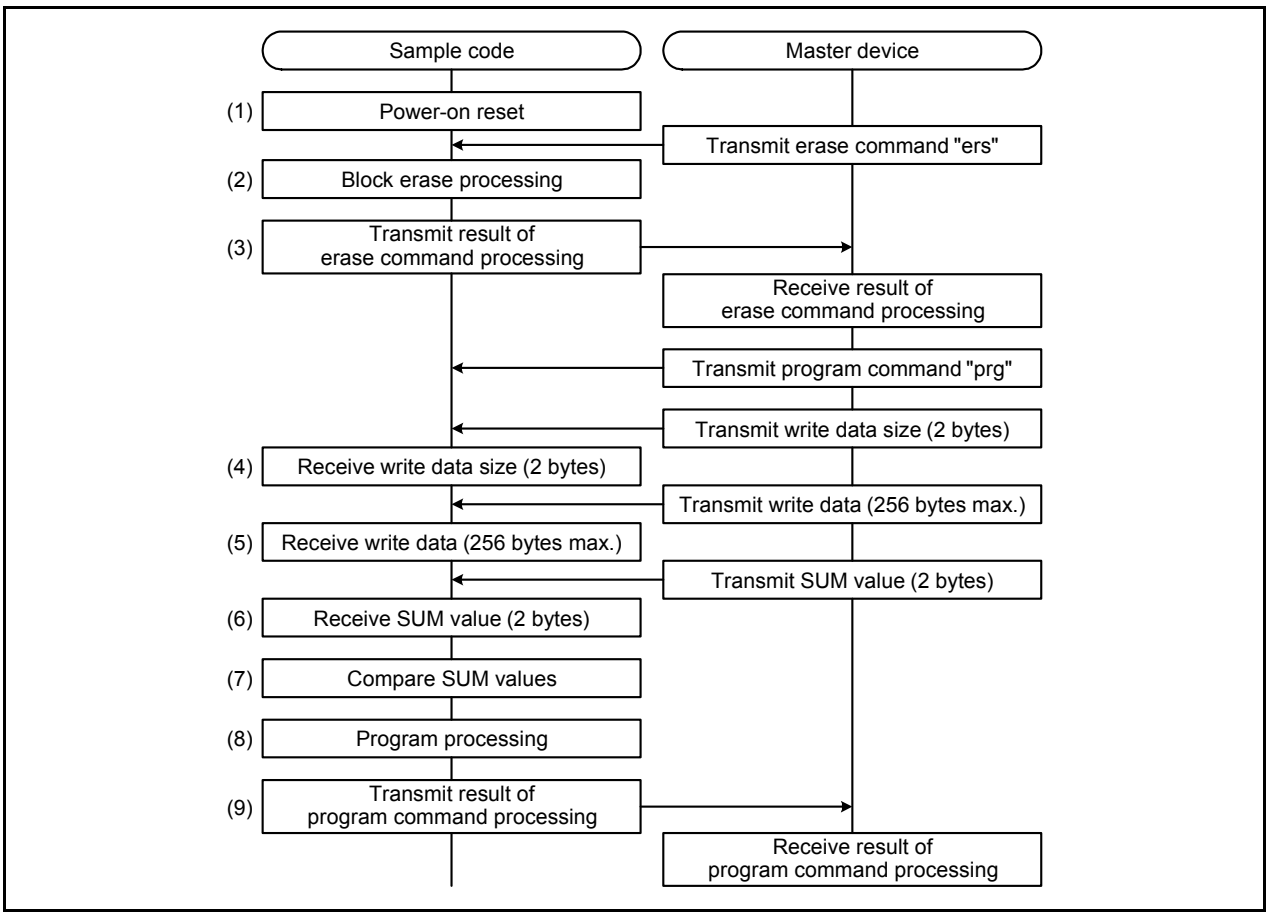


Figure 5.1 Example of Operation

Sample code operation is as follows:

(1) After a power-on reset, the sample code waits to receive a 3-byte control command from the master device.
(2) If the command received is the erase command (“ers”), block 7 in the flash memory is block erased.
(3) If the erase command processing is successful, 6Fh (‘o’) is transmitted to the master device. If an error occurs, 65h (‘e’) is transmitted. After the processing result is transmitted to the master device, return to step (1).
(4) If the command received is the program command (“prg”), the MCU receives the write data size (2-byte data).
(5) The MCU receives one packet of data (up to 256 bytes).
(6) The MCU receives the SUM value (2-byte data).
(7) The SUM value of the write data received is calculated and compared to the received SUM value.
(8) Data received is written to block 7 in the flash memory.
(9) If the program command processing is successful, 6Fh (‘o’) is transmitted to the master device. If an error occurs, 65h (‘e’) is transmitted. After the processing result is transmitted to the master device, return to step (1).

In the sample code, after the first byte of data is received, if the next data is not received within 5 ms, a timeout error occurs. If a timeout error occurs, 65h (‘e’) is transmitted to the master device and the sample code returns to step (1).

5.2 Constants

Table 5.3 lists the Constants Used in the Sample Code.

Table 5.3 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
ADR_BLOCK_7	((uint16_t *)0xFFFA0000)	Address for block 7 in the flash memory
ADR_CMD_1ST	((uint16_t *)0xFFFFF800)	Write address for the first command
CMD_BLOCK_ERASE_1ST	((uint16_t)0x0020)	Software command: Block erase (first command)
CMD_BLOCK_ERASE_2ND	((uint16_t)0x00D0)	Software command: Block erase (second command)
CMD_PROGRAM	((uint16_t)0x0043)	Software command: Program
CMD_CLEAR_STATUS	((uint16_t)0x0050)	Software command: Clear the status register
PROGRAM	(0x00707267)	Control command: Program command (‘p’’r’’g’)
ERASE	(0x00657273)	Control command: Erase command (‘e’’r’’s’)
CMD_SIZE	(3)	Receive data size (control command) [bytes]
LENGTH_SIZE	(2)	Receive data size (size) [bytes]
RECORD_SIZE	(256)	Receive data size (data) [bytes]
RECORD_SIZE_WORD	(RECORD_SIZE/2)	Receive data size (data) [words]
CHECKSUM_SIZE	(2)	Receive data size (SUM value) [bytes]
PROGRAM_SIZE_UNIT	(4)	Write size [words]
OK	(0)	Completed successfully
NG	(-1)	Error

5.3 Receive Data Storage Structure

Figure 5.2 shows the Receive Data Storage Structure.

```
/* **** Receive data storage structure **** */
typedef struct
{
    uint_32 command;                          /* Control command */
    uint_16 size;                             /* Size */
    uint_16 program_data[RECORD_SIZE_WORD];   /* Data */
    uint_16 check_sum;                        /* SUM value */
} rx_data_t;
```

Figure 5.2 Receive Data Storage Structure

5.4 Variables

Table 5.4 lists the Global Variables.

Table 5.4 Global Variables

Type	Variable Name	Contents	Function Used
rx_data_t	receive_data	Receive data storage structure variable	receive_command, receive_program_data, program
uint16_t	ebc0_tmp	EBC0 register save variable	FLASH_ew1_start, FLASH_ew1_end

5.5 Functions

Table 5.5 lists the Functions.

Table 5.5 Functions

Function Name	Outline
main	Main processing
receive_command	Control command reception
receive_program_data	Receive program command data
receive_message	Receive data from the master device
send_message	Send a result to the master device
UART2_init	UART2 initialization
TIMER_A0_init	Initialize timer for receive processing timeout
erase	Erase command processing
program	Program command processing
FLASH_ew1_start	Processing to enter EW1 mode
FLASH_ew1_end	Processing to exit EW1 mode

5.6 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Disable maskable interrupts, initialize the system clock, initialize timer A0, initialize UART2, and enter the main loop. The following processing occurs in the main loop: (1) Wait for data from UART2. (2) Receive the control command. (3) Perform processing for each control command. When processing the erase command: Perform the erase command processing and erase block 7. When processing the program command: Receive the size, data, and SUM values, perform the program command processing, and write data to block 7. (4) Transmit the result to the master device. The sample code does not perform recovery processing from a block erase error, program error, or UART2 error. Add recovery processing to the user program as needed.
Argument	None
Returned value	None

receive_command	
Outline	Control command reception
Header	None
Declaration	static void receive_command(void)
Description	Receive 3 bytes of data from the master device through the MCU's RXD2 pin, and set the received data as the control command of the receive data storage structure variable.
Argument	None
Returned value	None

receive_program_data	
Outline	Receive program command data
Header	None
Declaration	static int8_t receive_program_data(void)
Description	Write FFh to clear data of the receive data storage structure variable, receive data from the master device through the MCU's RXD2 pin, and set the data to the size, data, and SUM value for the receive data storage structure variable. Then, calculate the SUM value of the received data. If the calculated SUM value is equal to the received SUM value, return "completed successfully (OK)". If the size is less than 1 byte or more than 256 bytes, or if the calculated SUM value is not equal to the received SUM value, then return "error (NG)".
Argument	None
Returned value	Processing result: OK = Completed successfully NG = Error

receive_message	
Outline	Receive data from the master device
Header	None
Declaration	static int8_t receive_message(uint8_t *prx_data, uint16_t size)
Description	After setting the receive data storage area to 0, receive the specified amount of bytes using UART2. (1) Start the timer for the receive processing timeout. (2) Wait for UART2 to complete data reception or wait for a timer A0 interrupt request. (3) Confirm that either the data has been received or that a timer A0 interrupt request was generated, and then perform the following processes: Data has been received through UART2 - If the error flag in the UART receive register is set, an error is assumed to have occurred, and UART2 reception processing is interrupted. If the error flag is not set, receive data is saved to the receive data storage area specified by the argument. Timer A0 interrupt request was generated - A receive timeout error is assumed to have occurred, and UART2 receive processing is interrupted. (4) Repeat steps (1) to (3) until data of the receive data size specified by the argument has been received. When all data has been received, “completed successfully (OK)” is returned; when an error occurs during reception, “error (NG)” is returned.
Argument	uint8_t *prx_data: Pointer for the receive data storage area uint16_t size: Size of the data received
Returned value	Processing result: OK = Completed successfully NG = Error

send_message	
Outline	Send a result to the master device
Header	None
Declaration	static void send_message(char *message)
Description	Wait for the U2TB register to become empty, and then write transmit data to the U2TB register.
Argument	char *message: Pointer for the transmit message
Returned value	None

UART2_init	
Outline	UART2 initialization
Header	None
Declaration	static void UART2_init(void)
Description	Set registers associated with UART2 to asynchronous serial interface mode.
Argument	None
Returned value	None

TIMER_A0_init	
Outline	Initialize timer for receive processing timeout
Header	None
Declaration	static void TIMER_A0_init(void)
Description	Set timer A0 to timer mode.
Argument	None
Returned value	None

erase	
Outline	Erase command processing
Header	None
Declaration	static int8_t erase(void)
Description	Erase block 7 in the flash memory. (1) Enter EW1 mode. (2) Issue the flash memory block erase command (0020h,00D0h) and erase block 7. (3) Perform the status check. If a command sequence error or erase error occurs, issue the clear status register command (0050h) and set the return value to “error (NG)”. When no errors occur, set the return value to “completed successfully (OK)”. (4) Exit EW1 mode.
Argument	None
Returned value	Processing result: OK = Completed successfully NG = Error

program	
Outline	Program command processing
Header	None
Declaration	static int8_t program(void)
Description	Write data of the receive data storage structure variable to block 7 in the flash memory. (1) Enter EW1 mode. (2) Set the loop counter to 0. Set the start address of the flash memory write destination to the start address of block 7. (3) Issue the flash memory program command (0043h) and write the receive data storage structure variable 8-byte (4 word) data to the flash memory write destination address. (4) Perform a status check. If a command sequence error or program error occurs, issue the clear status register command (0050h) and the flash memory write processing is interrupted. (5) Increment the loop counter by 4 words. (6) Repeat steps (3) to (5) until all data has been written. (7) When all data has been written, set the return value to “completed successfully (OK)”. If an error occurs during the write procedure, set the return value to “error (NG)”. (8) Exit EW1 mode.
Argument	None
Returned value	Processing result: OK = Completed successfully NG = Error

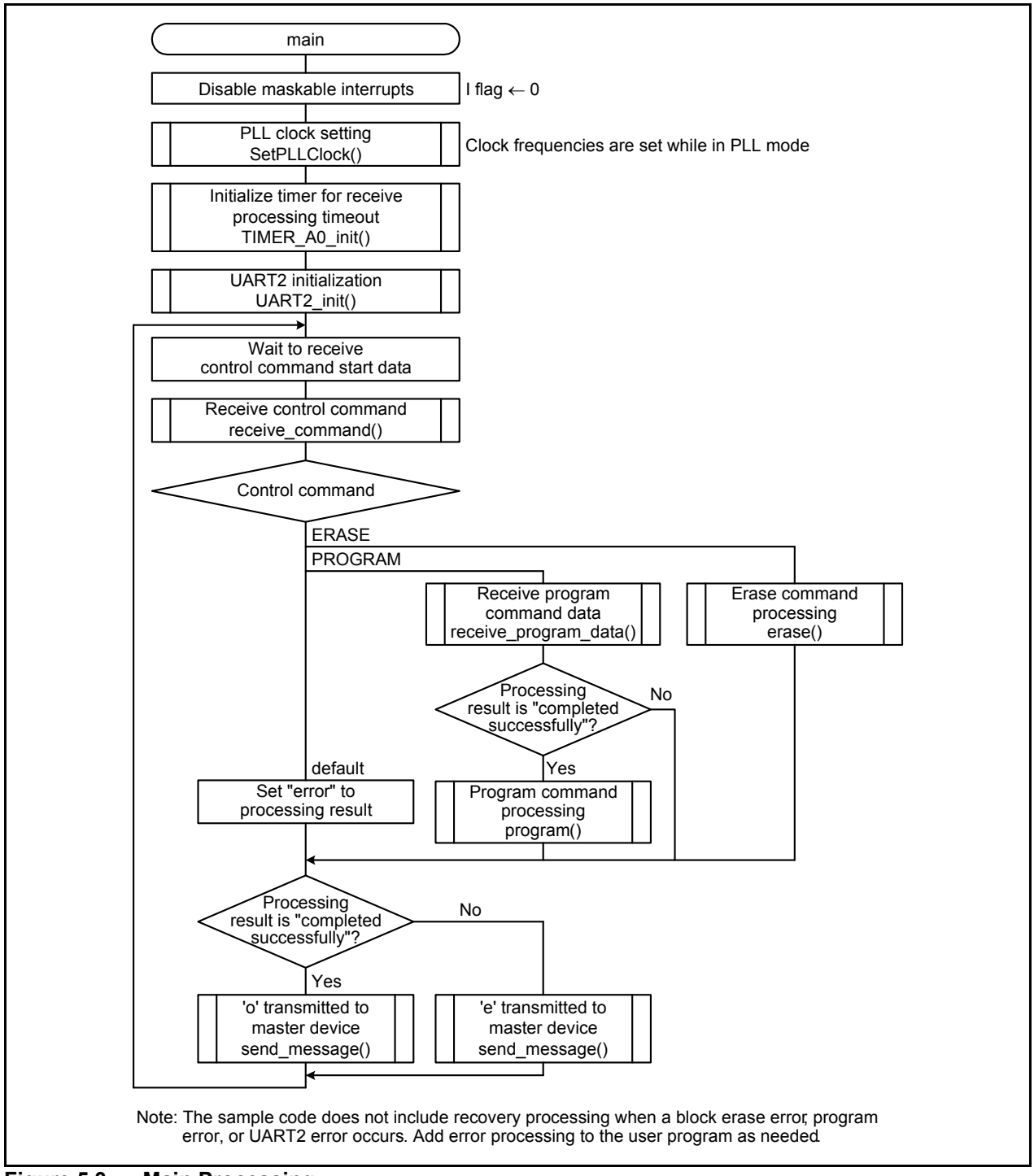
FLASH_ew1_start	
Outline	Processing to enter EW1 mode
Header	None
Declaration	static void FLASH_ew1_start(void)
Description	Disable maskable interrupts and save the external bus control register value. Then set the flash memory rewrite bus control register, and enter EW1 mode of the CPU rewrite mode.
Argument	None
Returned value	None

FLASH_ew1_end	
Outline	Processing to exit EW1 mode
Header	None
Declaration	static void FLASH_ew1_end(void)
Description	Enter the EW0 mode of the CPU rewrite mode, disable the CPU rewrite mode, and then set the external bus control register value back to the original setting value.
Argument	None
Returned value	None

5.7 Flowcharts

5.7.1 Main Processing

Figure 5.3 shows the Main Processing.



Note: The sample code does not include recovery processing when a block erase error, program error, or UART2 error occurs. Add error processing to the user program as needed.

Figure 5.3 Main Processing

5.7.2 Control Command Reception

Figure 5.4 shows the Control Command Reception.

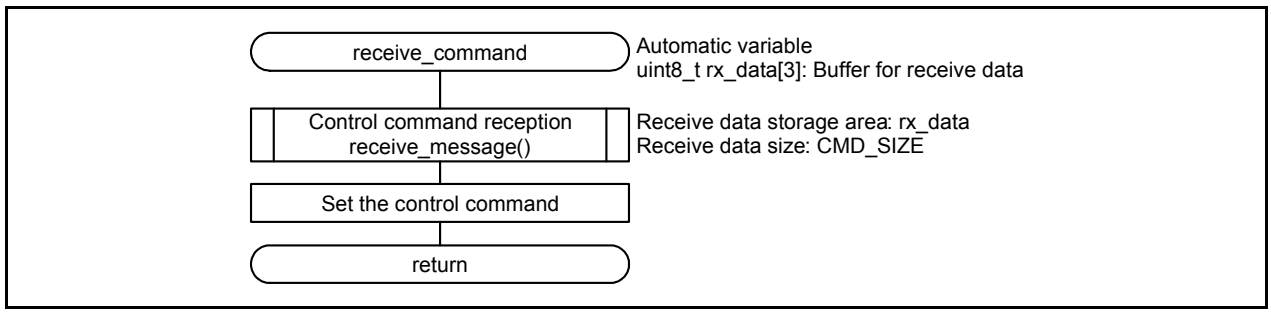


Figure 5.4 Control Command Reception

5.7.3 Receive Program Command Data

Figure 5.5 shows Receive Program Command Data.

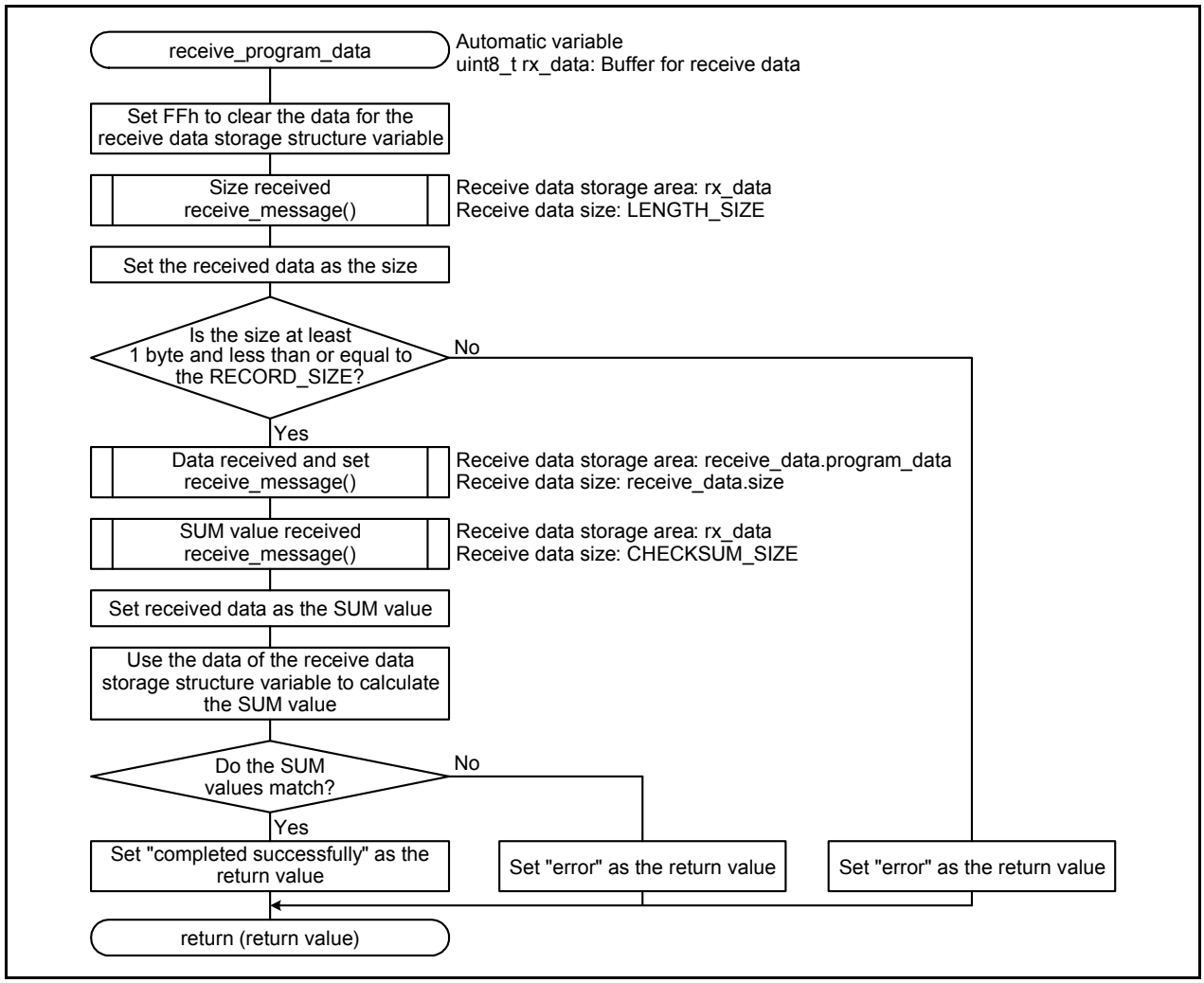


Figure 5.5 Receive Program Command Data

5.7.4 Receive Data From the Master Device

Figure 5.6 shows how to Receive Data From the Master Device.

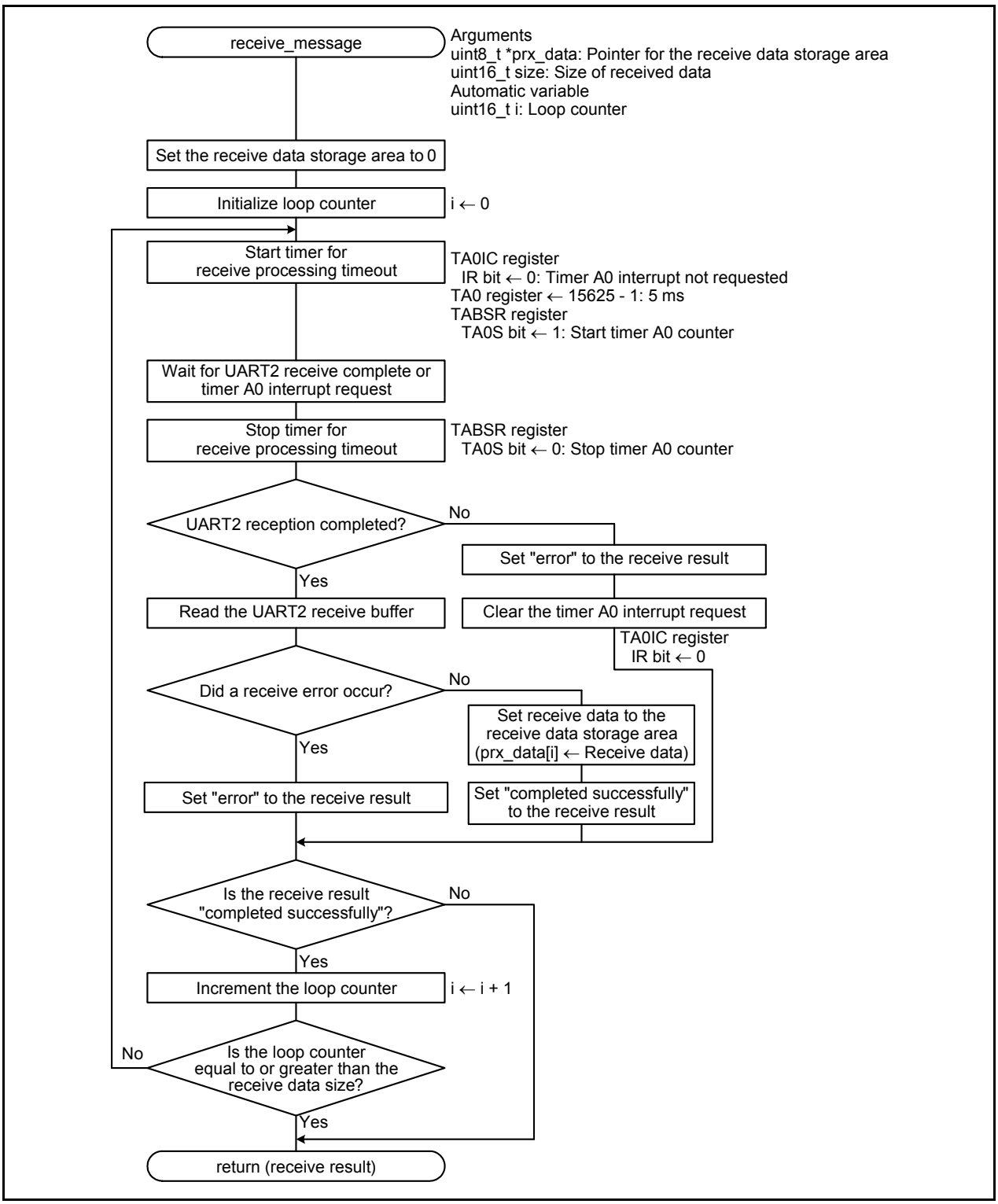


Figure 5.6 Receive Data From the Master Device

5.7.5 Send a Result to the Master Device

Figure 5.7 shows how to Send a Result to the Master Device.

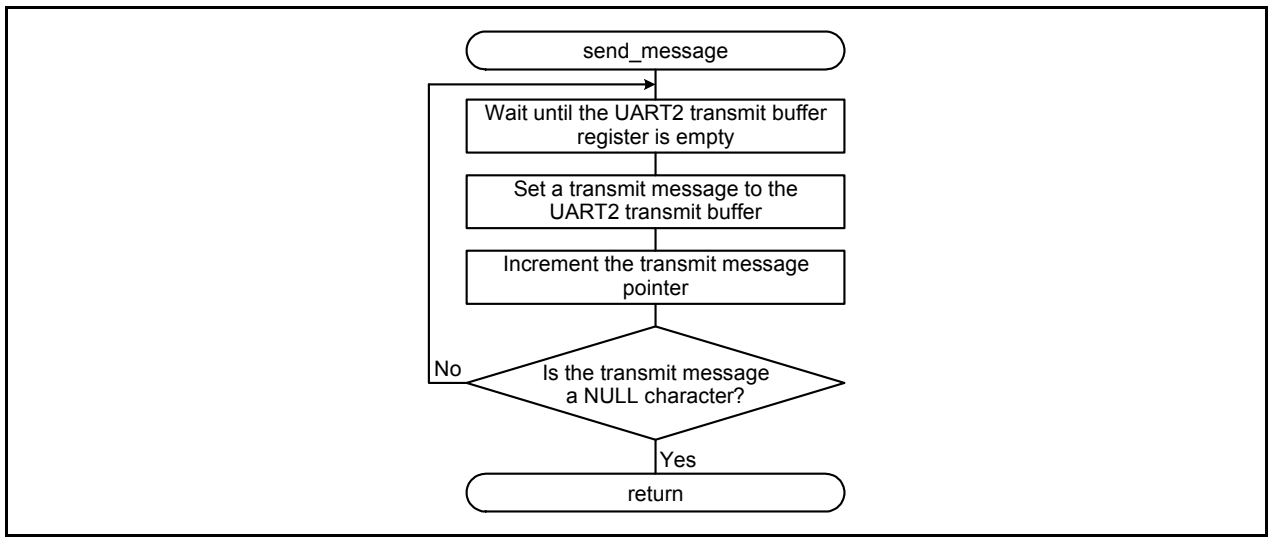


Figure 5.7 Send a Result to the Master Device

5.7.6 UART2 Initialization

Figure 5.8 shows UART2 Initialization.

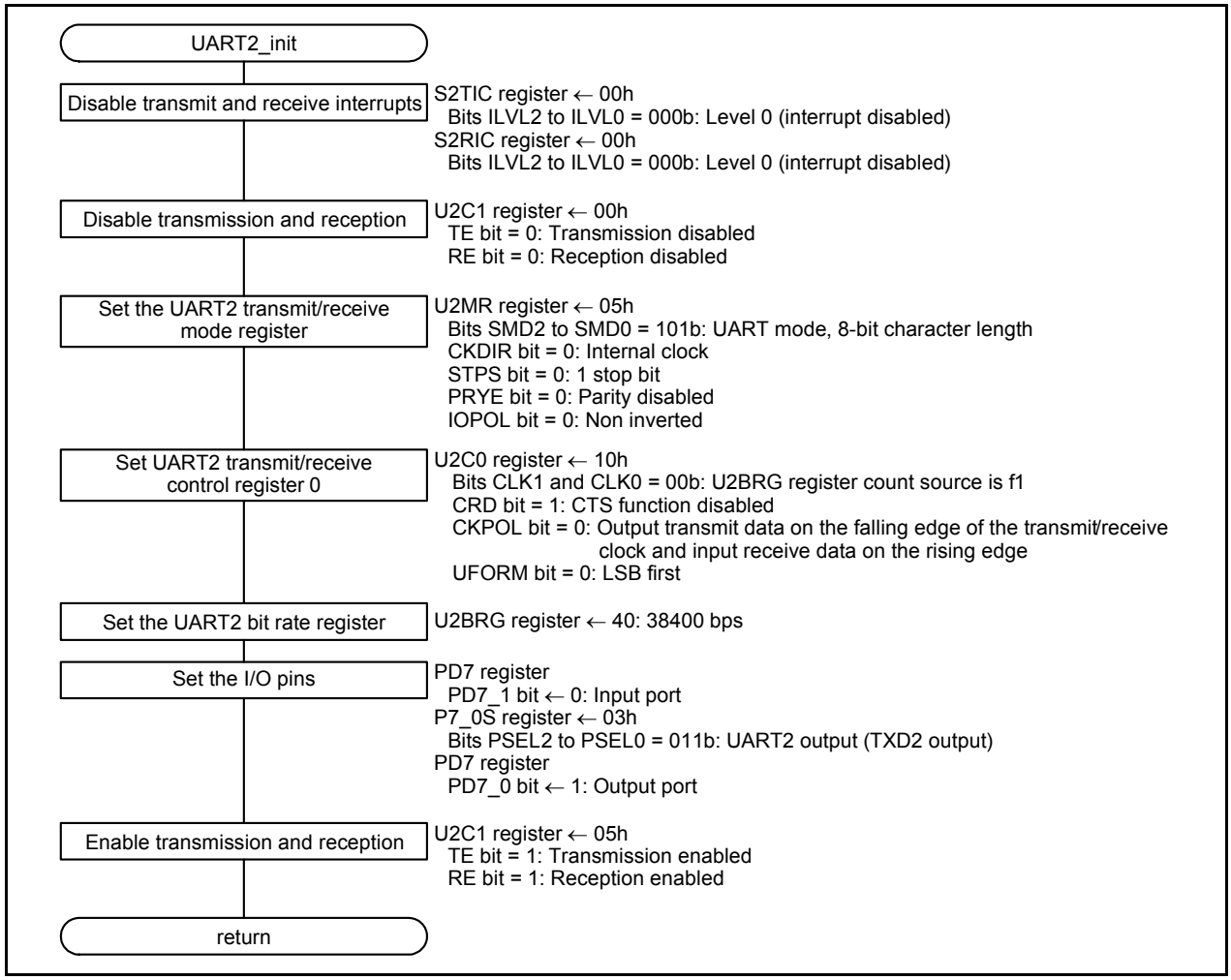


Figure 5.8 UART2 Initialization

5.7.7 Initialize Timer for Receive Processing Timeout

Figure 5.9 shows Initialize Timer for Receive Processing Timeout.

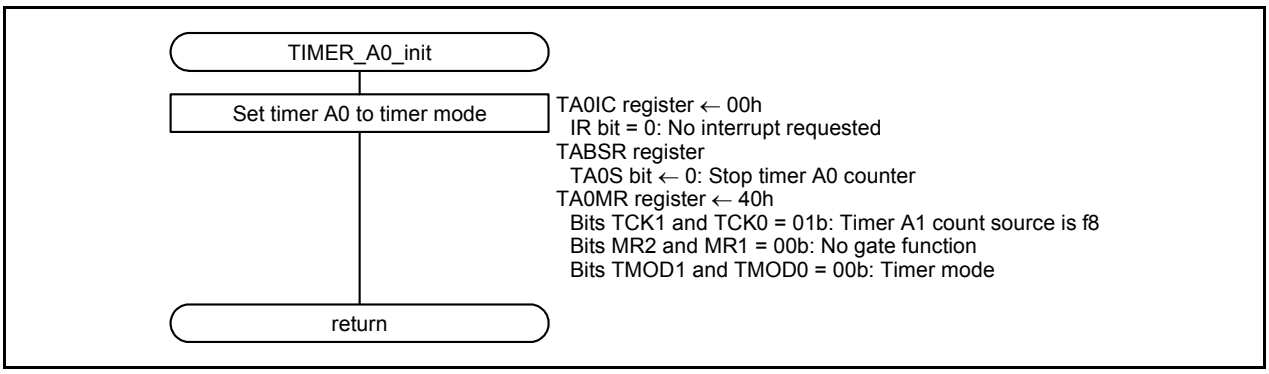


Figure 5.9 Initialize Timer for Receive Processing Timeout

5.7.8 Erase Command Processing

Figure 5.10 shows the Erase Command Processing.

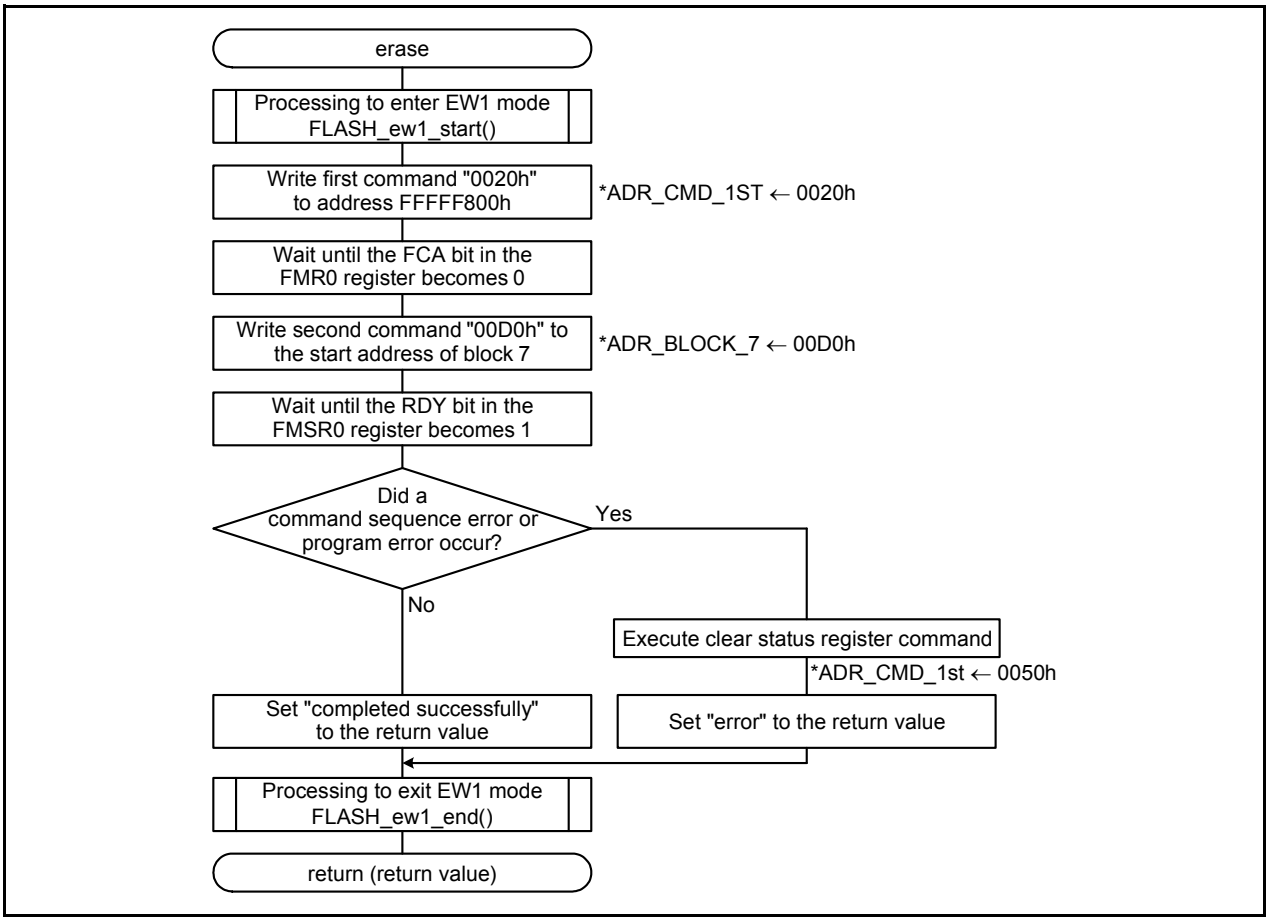


Figure 5.10 Erase Command Processing

5.7.9 Program Command Processing

Figure 5.11 shows the Program Command Processing.

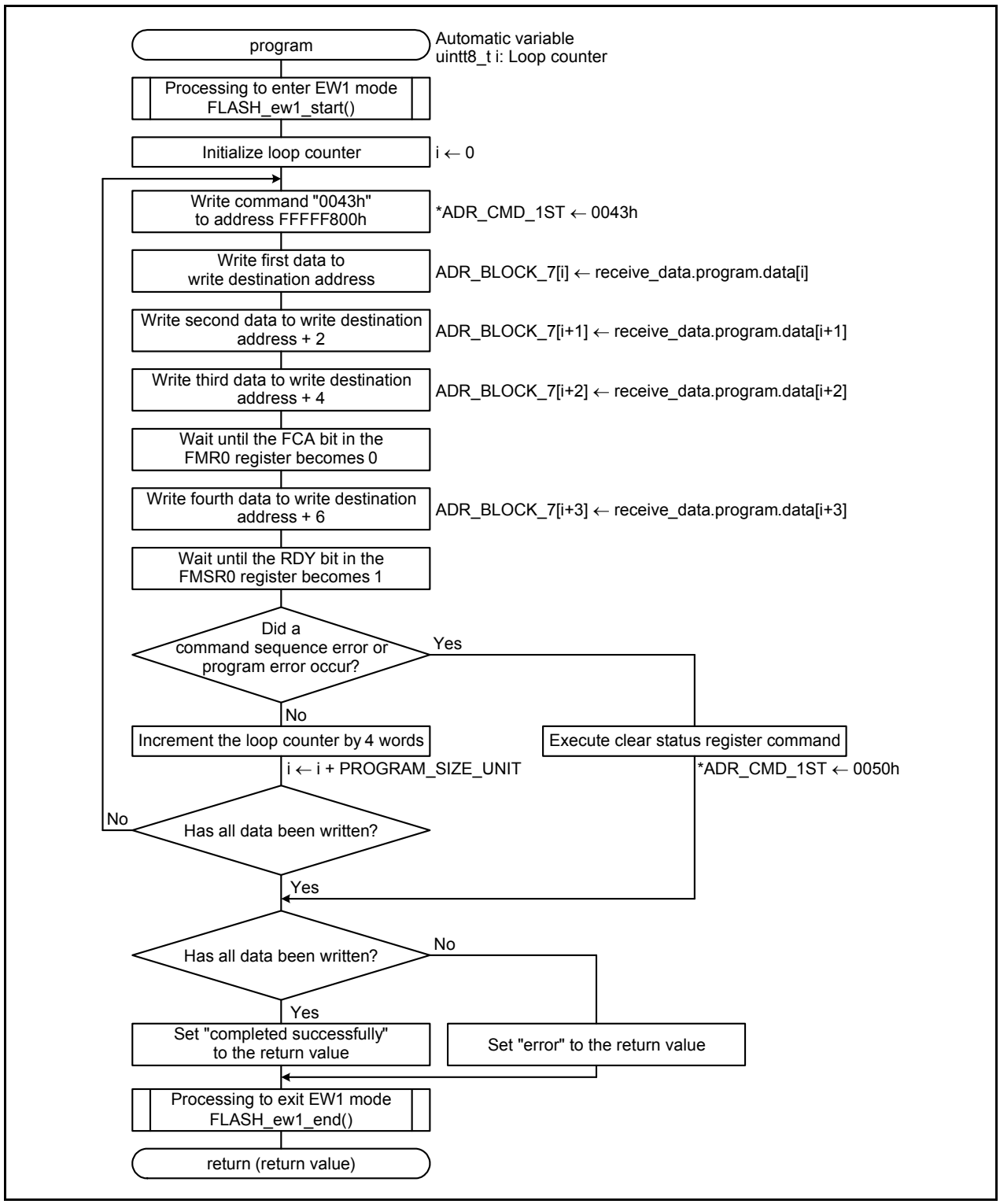


Figure 5.11 Program Command Processing

5.7.10 Processing to Enter EW1 Mode

Figure 5.12 shows the Processing to Enter EW1 Mode.

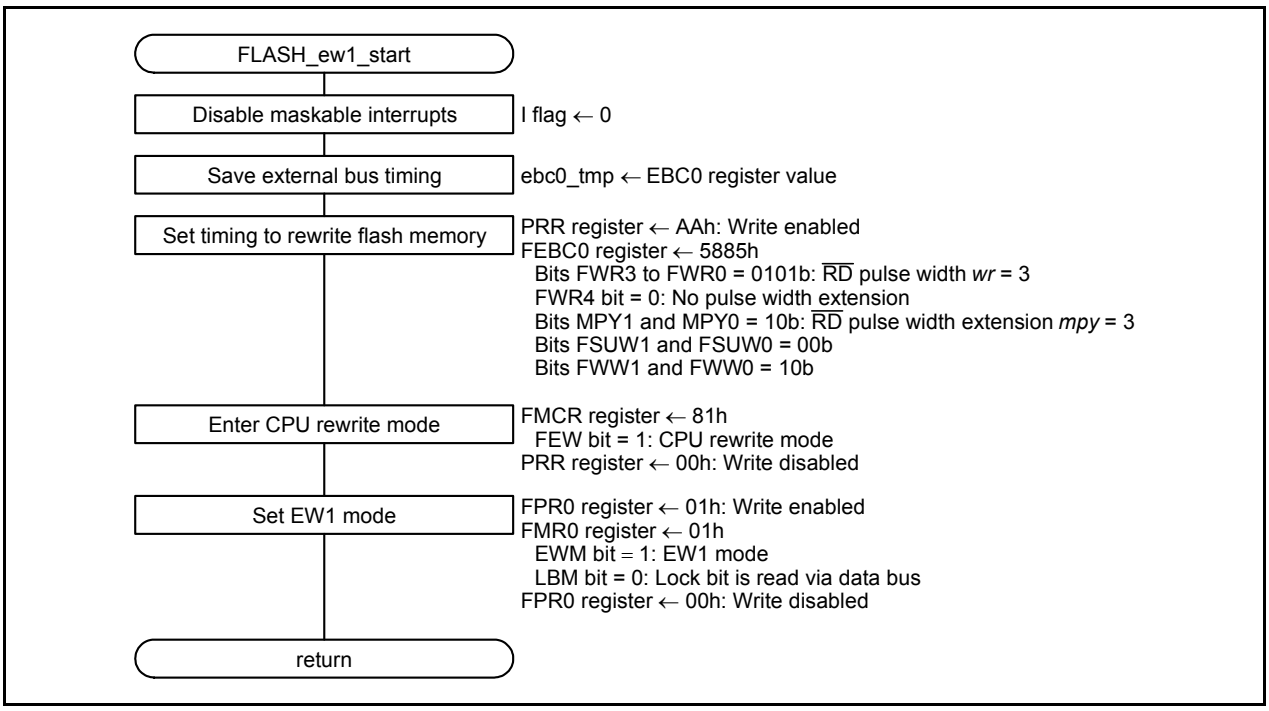


Figure 5.12 Processing to Enter EW1 Mode

5.7.11 Processing to Exit EW1 Mode

Figure 5.13 shows the Processing to Exit EW1 Mode.

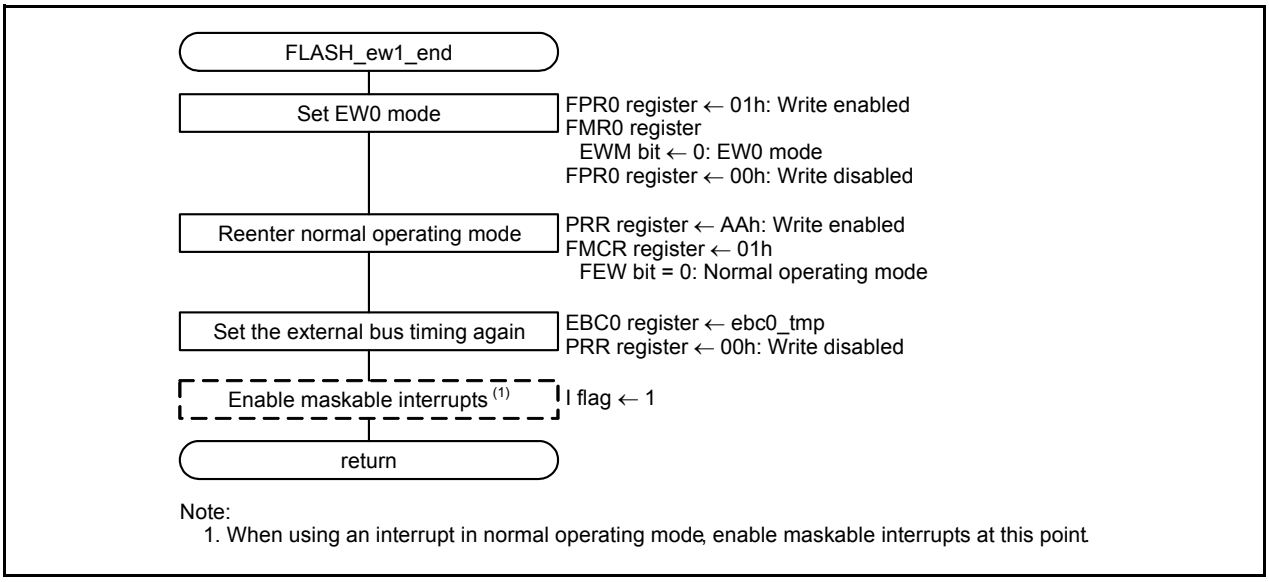


Figure 5.13 Processing to Exit EW1 Mode

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

R32C/116 Group User's Manual: Hardware Rev.1.20
R32C/117 Group User's Manual: Hardware Rev.1.20
R32C/118 Group User's Manual: Hardware Rev.1.20
The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual
R32C/100 Series C Compiler Package V.1.02
C Compiler User's Manual Rev.2.00
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision History	R32C/100 Series Rewriting the Flash Memory Using the Serial Interface (UART)

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 8, 2013	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": 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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

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

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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 906, 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 Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2