# RL78/F15 IEBus

## Setup Procedures for IEBB Master Transmission/Slave Reception

R01AN4096EJ0100
Rev. 1.00
Feb. 23, 2018

## Abstract

This application note describes the setup procedures for the transmission and reception (master transmission/slave reception) employing a multi-master bus system using the RL78/F15 IEBus controller (referred to as IEBB from here).

Under certain use conditions, the operations of the microcontroller might be different from the examples shown in this document. For details of IEBB function, refer to the user’s manuals. In addition, customers are required to sufficiently evaluate the use of IEBus in their environment.

## Contents

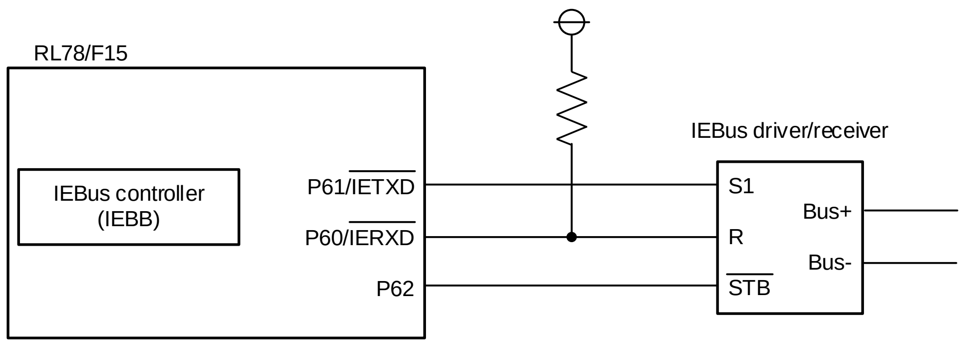
# 1. IEBB Master Transmission/Slave Reception Specifications

The conditions for use of the IEBB master transmission/slave reception that this application note describes are shown in Table 1-1. Also, the pin connections are shown in Figure 1-1.

This application note adopts the specification that the master transmission/slave reception are performed with the control bit limited to "0FH (Write data)".

**Table 1-1 Conditions for Use of IEBB Master Transmission/Slave Reception**

| Items | Conditions for Use |
|---|---|
| CPU/peripheral hardware clock ($f_{CLK}$) | 32 MHz |
| IEBB operation clock ($f_{MCK}$) | 8 MHz ($f_{CLK}$/4) |
| IEBB communication mode | Mode 1 (32 bytes/frame), Individual communication |
| Unit address | 100H |
| Slave unit address | 180H |
| Master transmission data | 1 byte to 32 bytes (Control bit = "0FH") |
| Slave reception data | 1 byte to 32 bytes (Control bit = "0FH") |




Caution: When designing hardware, examine whether pull-up/pull-down resistors are necessary for the communication lines to $\overline{IERXD}$ and $\overline{IETXD}$ in your environment. If such resistors are necessary, examine the resistor value and take into consideration the delay of the bus data and the current in your environment.

**Figure 1-1 Pin Connections**

## 1.1 Description of Memories Used

Memories (SFRs, RAM) used for the IEBB master transmission/slave reception are shown below.

**Table 1-2 List of SFRs Used for IEBB Master Transmission/Slave Reception**

| Register Name | In Use (√) / Not in Use (-) | Read/Write | Setting Example |
|---|---|---|---|
| Peripheral enable register 2 (PER2) | √ | Write | IEBUSEN = 1 |
| IEBB0 bus control register (IEBB0BCR) | √ | Write | 88H (at initializing) |
| | | | C8H (at master transmission) |
| IEBB0 power save register (IEBB0PSR) | √ | Write | 80H |
| IEBB0 unit address register (IEBB0UAR) | √ | Write | 100H |
| IEBB0 slave address register (IEBB0SAR) | √ | Write | 180H |
| IEBB0 partner address register (IEBB0PAR) | - | - | - |
| IEBB0 reception slave address register (IEBB0RSA) | - | - | - |
| IEBB0 control data register (IEBB0CDR) Note 1, 3 | √ | Read/Write | 0FH |
| IEBB0 message length register (IEBB0DLR) Note 2, 4 | √ | Read/Write | 01H to 20H |
| IEBB0 transmission control data register (IEBB0TCD) Note 1 | √ | Write | 0FH |
| IEBB0 reception control data register (IEBB0RCD) Note 3 | √ | Read | - |
| IEBB0 transmission message length register (IEBB0TDL) Note 2 | √ | Write | 01H to 20H |
| IEBB0 reception message length register (IEBB0RDL) Note 4 | √ | Read | - |
| IEBB0 clock selection register (IEBB0CKS) | √ | Write | 04H |
| IEBB0 slave status register (IEBB0SSR) | - | - | - |
| IEBB0 unit status register (IEBB0USR) | √ | Read | - |
| IEBB0 interrupt status register (IEBB0ISR) | √ | Read/Write | IEBB0IEBE = 0 Note 5 |
| IEBB0 error status register (IEBB0ESR) | √ | Read | - |
| IEBB0 field status register (IEBB0FSR) | √ | Read | - |
| IEBB0 success count register (IEBB0SCR) | - | - | - |
| IEBB0 communication count register (IEBB0CCR) | - | - | - |
| IEBB0 status clear register 0 (IEBB0STC0) | √ | Write | F9H |
| IEBB0 data register (IEBB0DR) | √ | Read/Write | Transmission/ Reception data |
| IEBB0 data polarity select register (IEBB0DPS) | √ | Write | 00H |

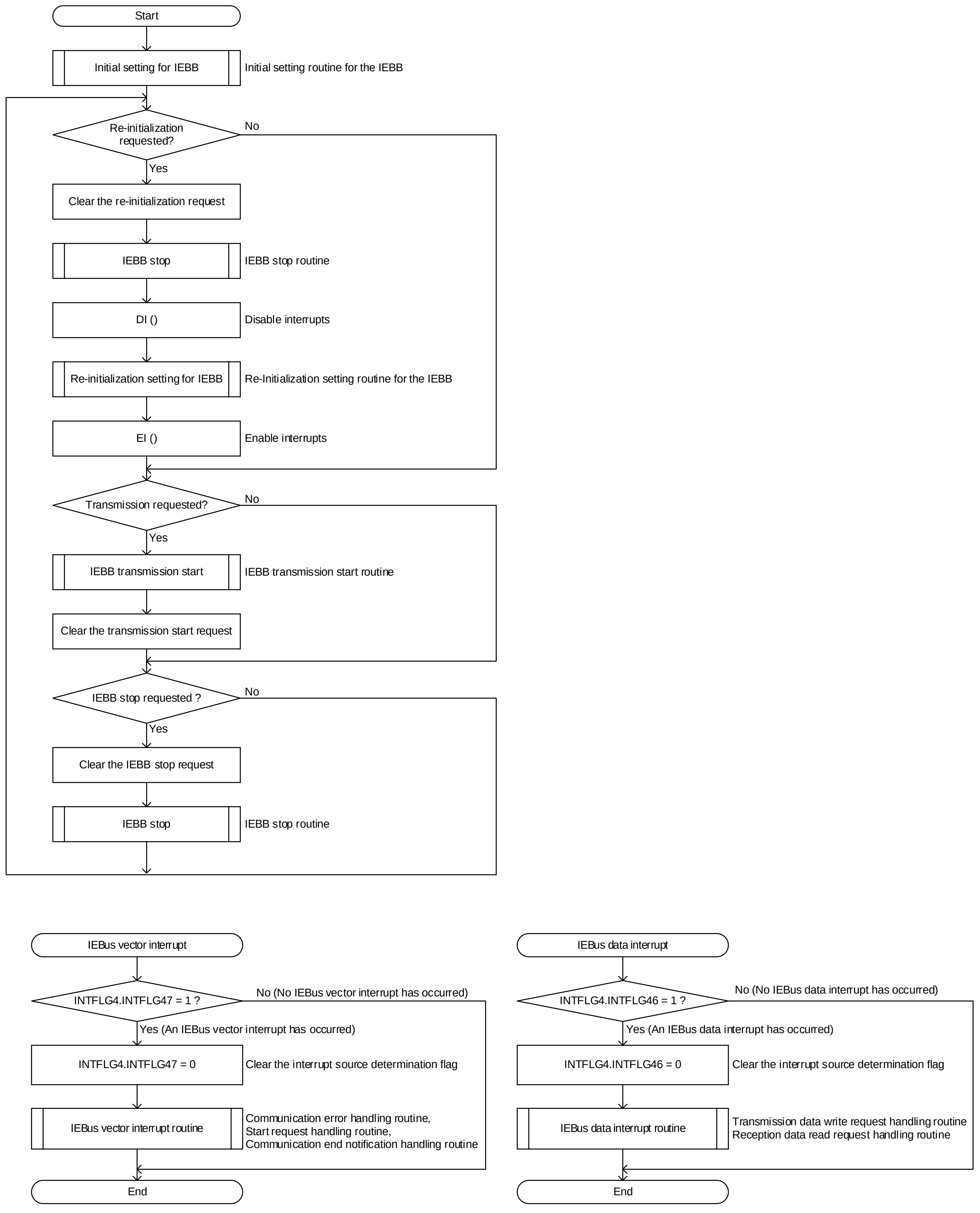
Notes: 1. For the master transmission, set up either the IEBB0CDR register or the IEBB0TCD register.
2. For the master transmission, set up either the IEBB0DLR register or the IEBB0TDL register.
3. For the slave reception, read the value from either the IEBB0CDR register or the IEBB0RCD register.
4. For the slave reception, read the value from either the IEBB0DLR register or the IEBB0RDL register.
5. The value can be written only to the IEBB0IEBE bit.

**Table 1-3 List of RAM/Variables for Use in IEBB Master Transmission/Slave Reception Procedure**

| Variable Name | Specification |
|---|---|
| u8_iebb_state | The variable for storing the master transmission/slave reception status<br>00H : Not initialized<br>01H : Initialized<br>02H : Arbitration situation during the arbitration period<br>03H : Data-transmitting<br>04H : Data-receiving |
| u8_iebb_txcount | The data field transmission times counter used in the master transmission |
| u8_iebb_rxcount | The data field reception times counter used in the slave reception |
| u8_iebb_comerr | The variable for storing the communication error status which indicates the cause of communication error<br>b0 : An inter-third-party communication error has occurred<br>b1, b2 : -<br>b3 : An overrun error has occurred<br>b4 : An underrun error has occurred<br>b5 : A NACK reception error has occurred<br>b6 : A parity error has occurred<br>b7 : A timing error has occurred |
| u8_iebb_comerr2 | The variable for storing the communication error status 2 which indicates the cause of communication error<br>b0 : Transmission has been requested in any state except "Initialized"<br>b1 : A command error has occurred<br>b2 : An arbitration loss has occurred during the arbitration period (with a slave request)<br>b3 : An arbitration loss has occurred during the arbitration period (without a slave request)<br>b4 : An excessive reception has occurred (Data read requests have been generated over the expected number of times)<br>b5-b6 : -<br>b7 : An illegal interrupt has occurred |
| u16_iebb_txslaveaddr | The variable for storing the communication-partner slave unit address used in the master transmission ("180H" in this document) |
| u16_iebb_rxmasteraddr | The variable for storing the master unit address used in the slave reception |
| u8_iebb_txcontrolbit | The variable for storing the control bit used in the master transmission ("0FH" in this document) |
| u8_iebb_rxcontrolbit | The variable for storing the control bit use in the slave reception ("0FH" in this document) |
| u8_iebb_txdatalength | The variable for storing the message data length used in the master transmission ("01H" to "20H" in this document) |
| u8_iebb_rxdatalength | The variable for storing the message data length used in the slave reception ("01H" to "20H" in this document) |
| u8_iebb_txreserve | The variable for storing the master transmission suspension status<br>00H : The master transmission has not been suspended<br>81H : The master transmission has been suspended |
| u8_iebb_txbuff[32] | The array for storing the message data bits used in the master transmission |
| u8_iebb_rxbuff[32] | The array for storing the message data bits used in the slave reception |

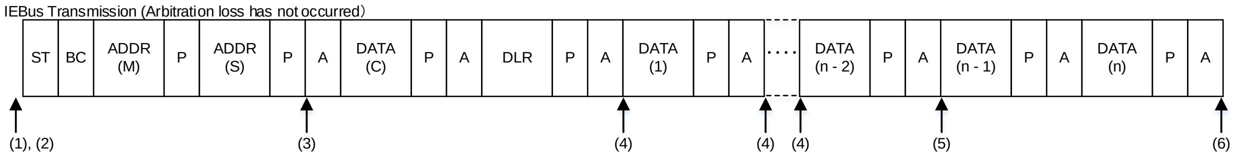
## 2. Setup Procedures for IEBB Master Transmission/Slave Reception

This chapter describes the IEBB master transmission/slave reception processing routines (setup procedures). Figure 2-1 to Figure 2-3 show the IEBB master transmission/slave reception processing routines.




Caution: The IEBus vector interrupt source is allocated to the same vector table address as serial array unit 1 interrupt sources (INTSR1/INTCSI11/INTIIC11). Also, the IEBus data interrupt source is allocated to the same vector table address as serial array unit 1 interrupt sources (INTST1/INTCSI10/INTIIC10).

**Figure 2-1 IEBB Master Transmission/Slave Reception Processing**




IEBB Master transmission processing

(1) Initial setting for the IEBB.

(2) Set up the IEBB0SAR, IEBB0TCD (to "0FH"), IEBB0DLR, IEBB0DR (the first byte of the message data to be transmitted), and IEBB0BCR (to "C8H") registers to initiate the transmission.

(3) An IEBus vector interrupt (Start request) occurs. Make sure that neither a slave request nor an arbitration loss has occurred.

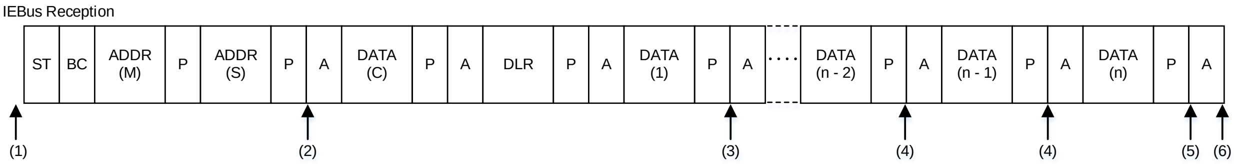
(4) An IEBus data interrupt (Transmission data write request) occurs. Write the next byte of the message data to be transmitted to the IEBB0DR register.

(5) An IEBus data interrupt (Transmission data write request) occurs. Write the last byte of the message data to be transmitted (DATA(n)) to the IEBB0DR register.

(6) An IEBus vector interrupt (Communication end) occurs and the master transmission finishes.

Remark: ST: Start bit, BC: Broadcast bit, Address (M): Master address bit, Address (S): Slave address bit, Data (C): Control bit, DLR: Message length bit, Data (n): Data bit, P: Parity bit, A: Acknowledge bit

**Figure 2-2 Master Transmission Processing (Timing Chart)**




IEBB Slave reception processing

(1) Initial setting for the IEBB (Write "88H" to the IEBB0BCR register to enable slave reception).

(2) An IEBus vector interrupt (Start request) occurs. Make sure that a slave request has been generated.

(3) An IEBus data interrupt (Reception data read request) occurs. Read the received master address, control bit, and message length bit. Then read the received message data from the IEBB0DR register.

(4) An IEBus data interrupt (Reception data read request) occurs. Read the received message data from the IEBB0DR register.

(5) An IEBus data interrupt (Reception data read request) occurs. Read the last received message data (Data(n)) from the IEBB0DR register.

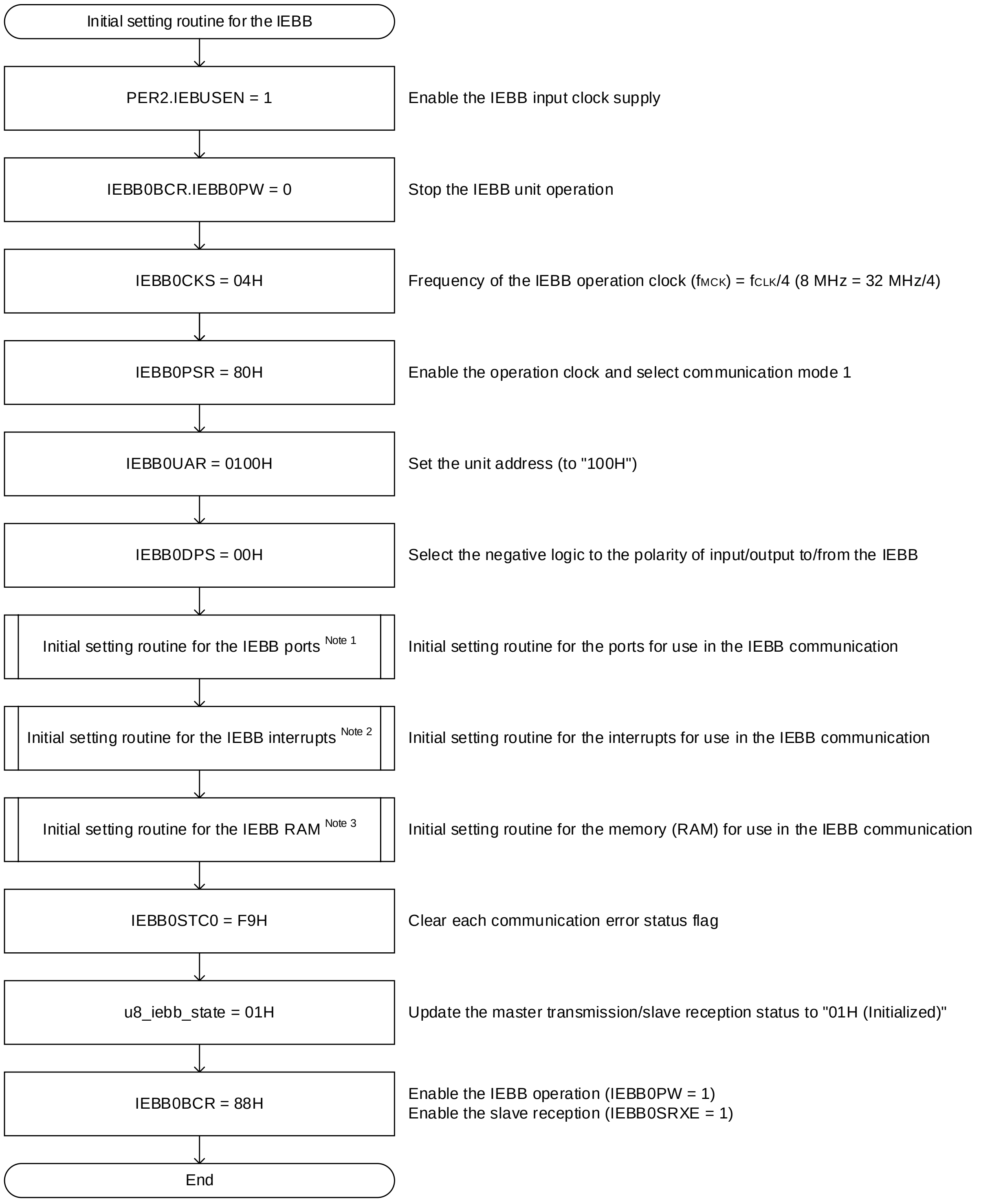
(6) An IEBus vector interrupt (Communication end) occurs and the slave reception finishes.

Remark: ST: Start bit, BC: Broadcast bit, Address (M): Master address bit, Address (S): Slave address bit, Data (C): Control bit, DLR: Message length bit, Data (n): Data bit, P: Parity bit, A: Acknowledge bit

**Figure 2-3 Slave Reception Processing (Timing Chart)**

## 2.1 Initial Setting for IEBB

The flow of the initial setting routine for the IEBB is shown in Figure 2-4.

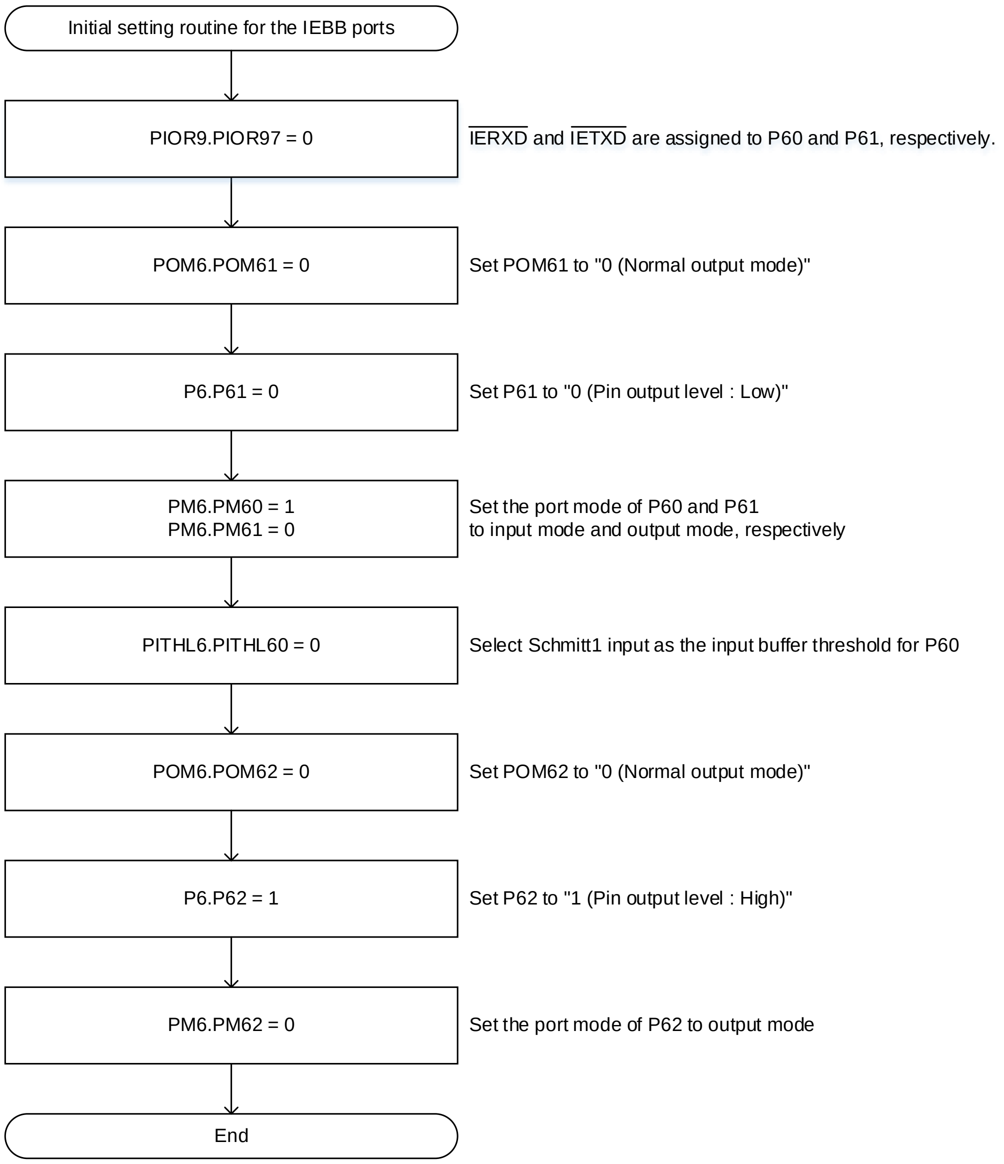



Notes: 1. See Figure 2-5.
2. See Figure 2-6.
3. See Figure 2-7.

**Figure 2-4 Initial Setting Routine for IEBB**

### 2.1.1 Initial Setting for IEBB Ports

In this document, $\overline{\text{IETXD}}$ and $\overline{\text{IERXD}}$ are assigned to P61 and P60, respectively. Also, P62 is used as the standby control pin for the IEBus transceiver. The flow of the initial setting routine for the IEBB ports is shown in Figure 2-5.




| IEBB pin setting | PIOR9 | PMn | Pn | POM6 | PITHLn |
|---|---|---|---|---|---|
| P60/$\overline{\text{IERXD}}$ | PIOR97 = 0 | PM60 = 1 | - | - | PITHL60 = x |
| P61/$\overline{\text{IETXD}}$ | PIOR97 = 0 | PM61 = 0 | P61 = 0 | POM61 = x | - |
| P50/($\overline{\text{IERXD}}$) Note | PIOR97 = 1 | PM50 = 1 | - | - | PITHL50 = x |
| P51/($\overline{\text{IETXD}}$) Note | PIOR97 = 1 | PM51 = 0 | P51 = 0 | - | - |

n = 5, 6

Remark: - : Setting is not required, or neither the corresponding register nor the bit exists (POM5 register does not exist).

x : Select the value that satisfies the specification of the IEBus transceiver for your product.

Note: The 48-pin products are not equipped with the pins.

**Figure 2-5 Initial Setting Routine for IEBB Ports**

### 2.1.2 Initial Setting for IEBB Interrupts

The flow of the initial setting routine for the IEBB interrupts (the IEBus vector interrupt and the IEBus data interrupt) is shown in Figure 2-6. This document provides an example in which the priority levels of the IEBus vector interrupt and the IEBus data interrupt are specified as level 1 and level 0, respectively.

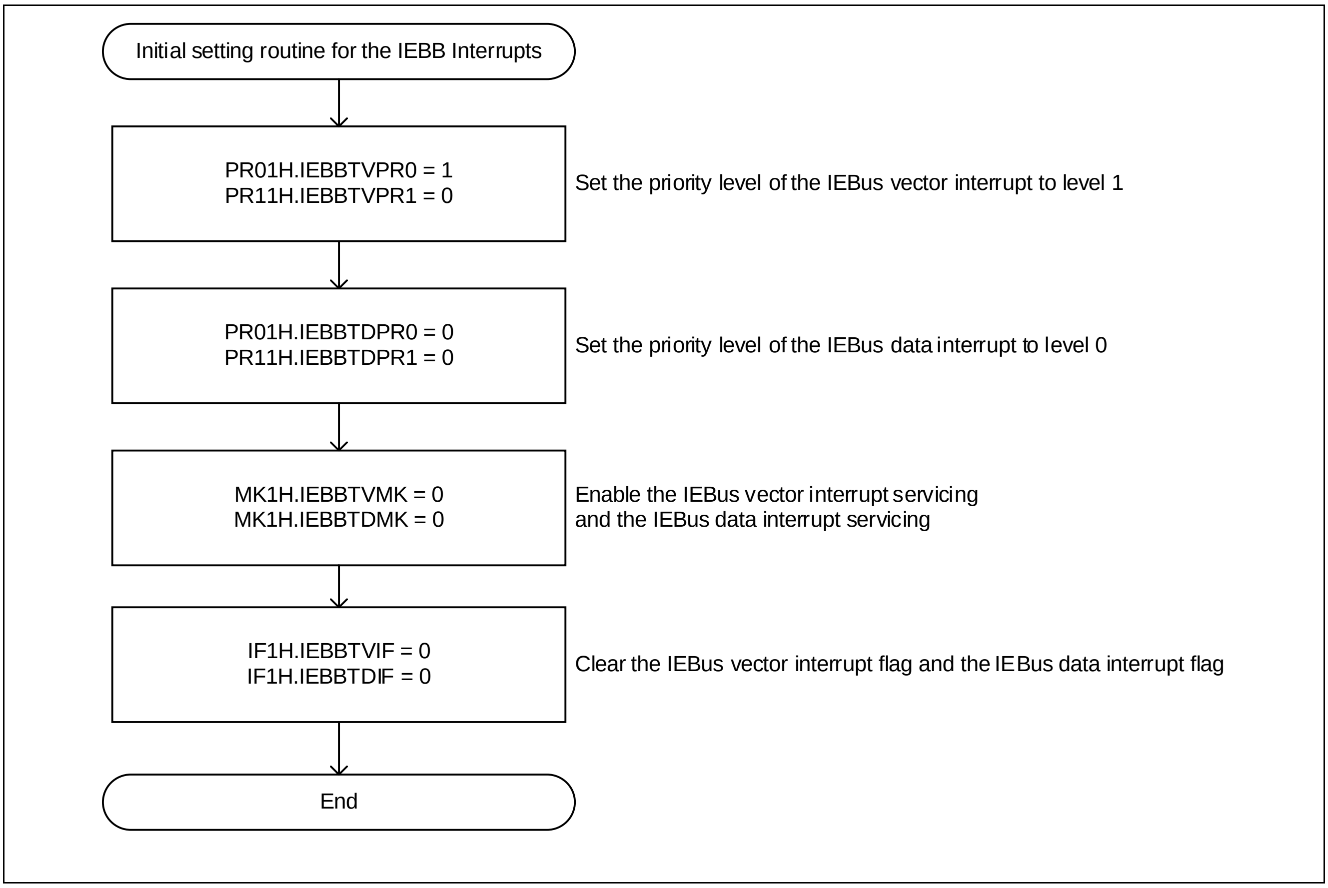



**Figure 2-6 Initial Setting Routine for IEBB interrupts**

### 2.1.3 Initial Setting for IEBB RAM

The specification of the initial setting routine for memory (RAM) used in the IEBB master transmission/slave reception is shown in Figure 2-7.

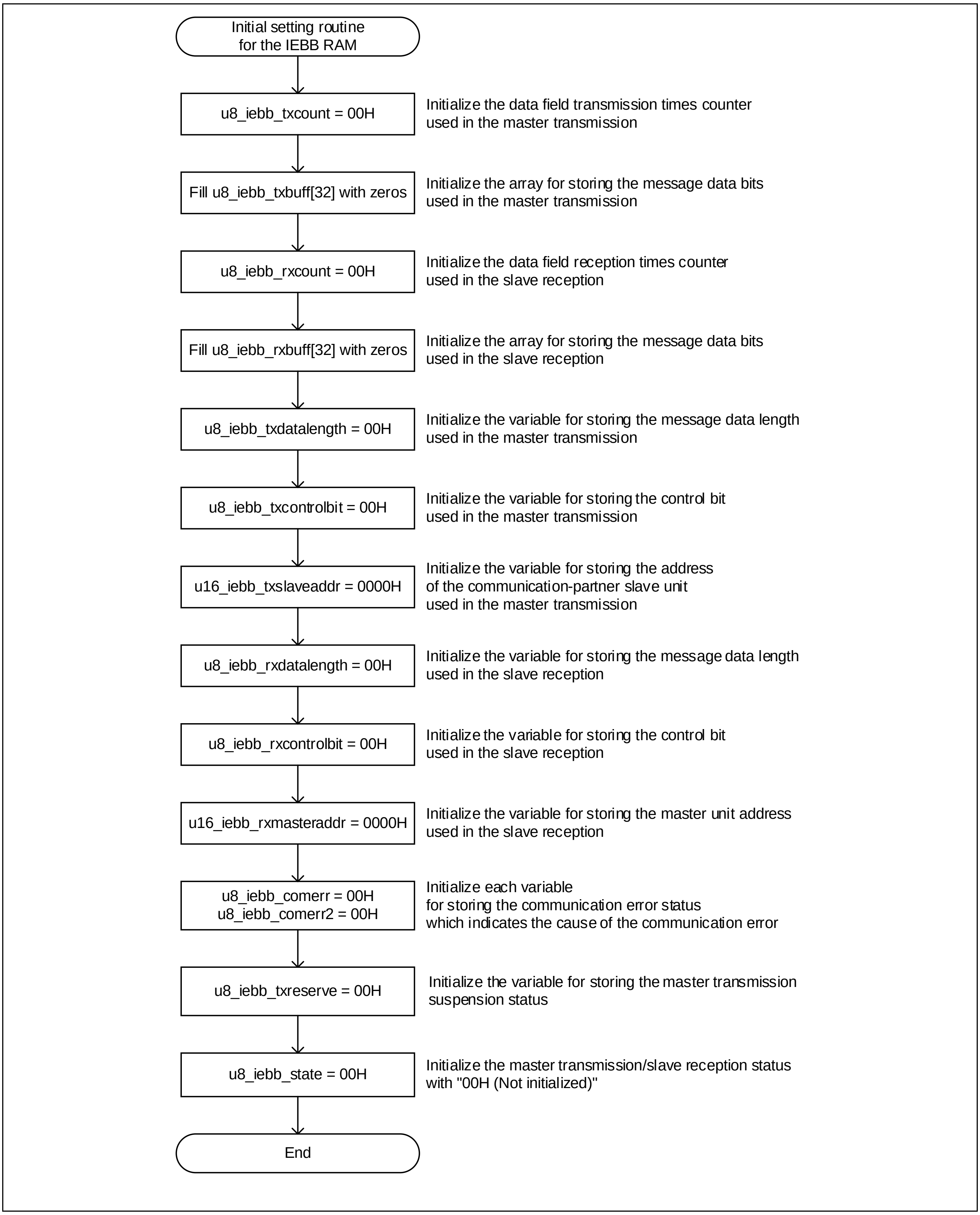
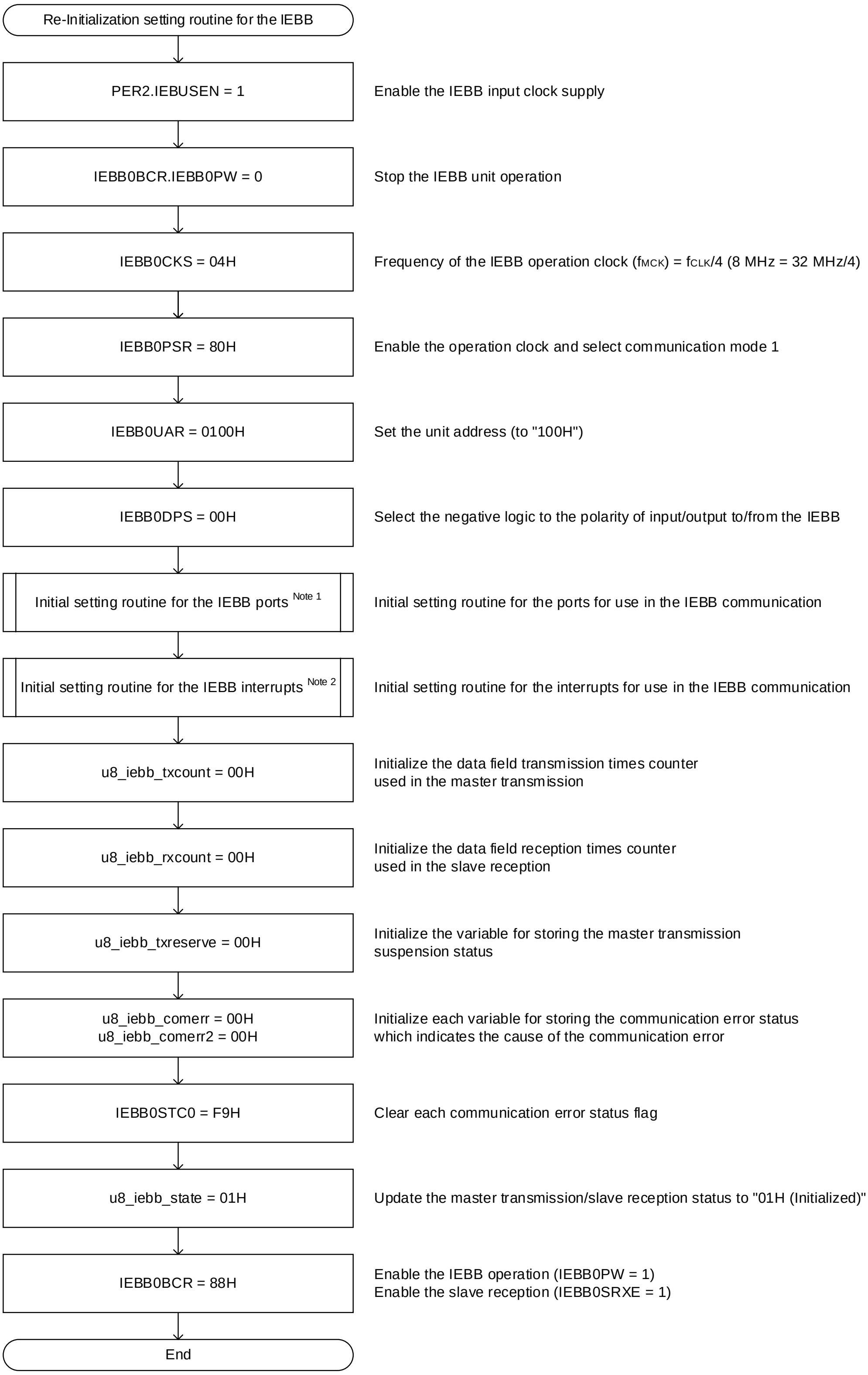



**Figure 2-7 Initial Setting Routine for IEBB RAM**

## 2.2 Re-initialization Setting for IEBB

The flow of the re-initialization setting routine for the IEBB is shown in Figure 2-8.




Notes: 1. See Figure 2-5.
2. See Figure 2-6.

**Figure 2-8 Re-initialization Setting Routine for IEBB**

## 2.3 IEBB Stop

The flow of the IEBB stop routine is shown in Figure 2-9.

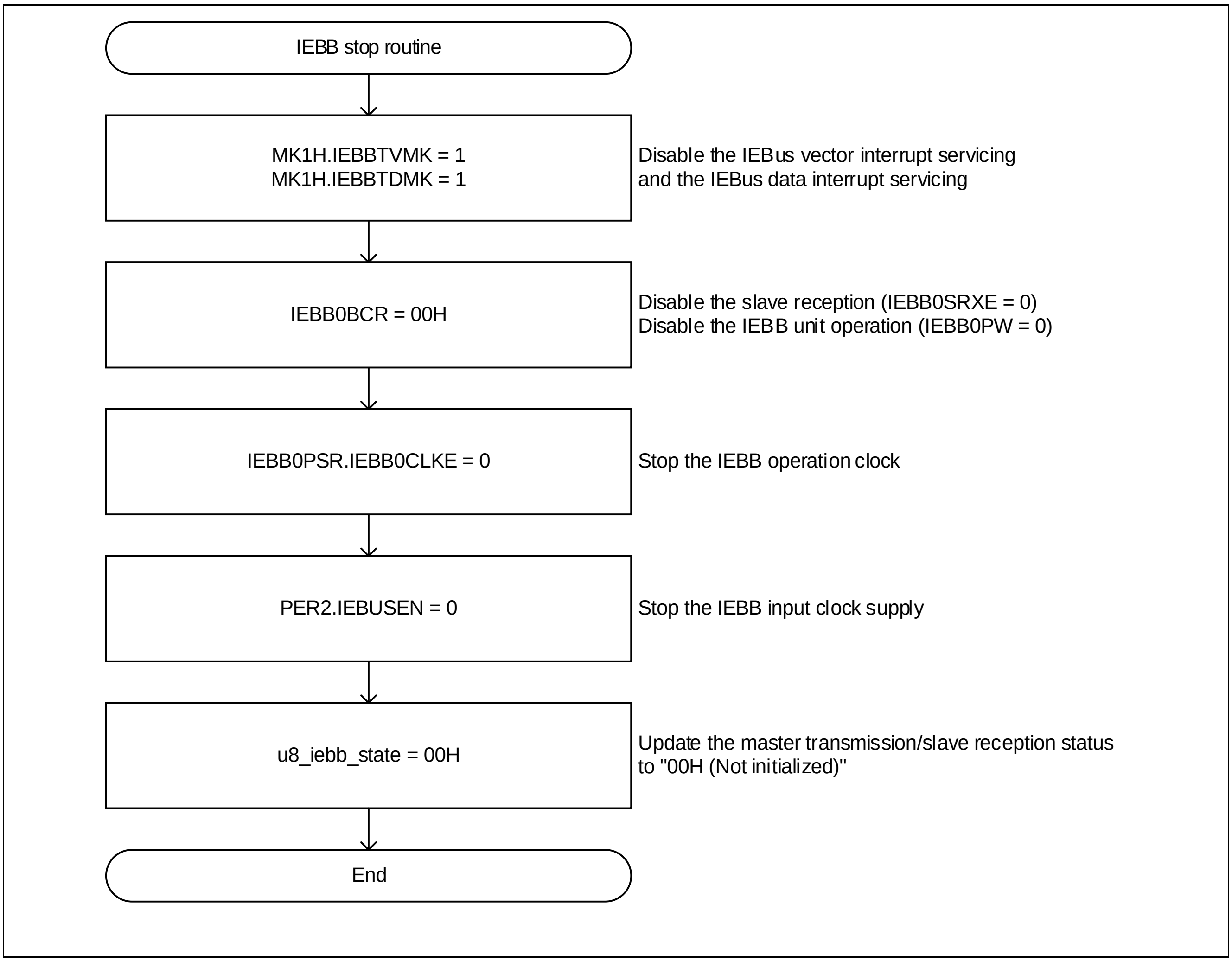


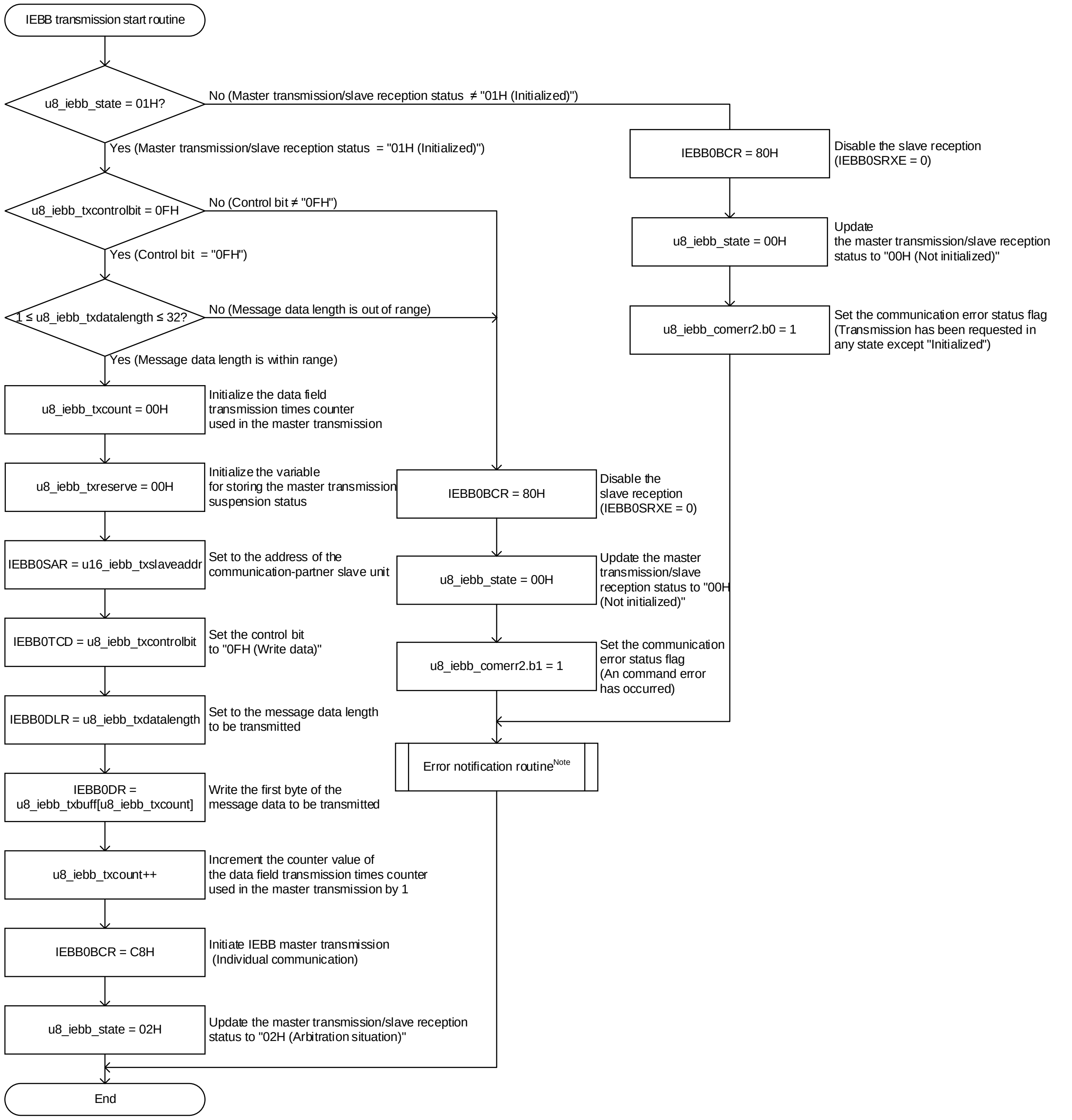

**Figure 2-9 IEBB Stop Routine**

## 2.4 IEBB Transmission Start

The flow of the IEBB transmission start routine is shown in Figure 2-10.

Before executing the IEBB transmission start routine, set the following variables to the appropriate values.

- u16_iebb_txslaveaddr: The variable for storing the address of the communication-partner slave unit
- u8_iebb_txcontrolbit: The variable for storing the control bit (set to "0FH")
- u8_iebb_txdatalength: The variable for storing the length of the message data to be transmitted (set to any value ranging from "01H" to "20H")
- u8_iebb_txbuff[32]: The array for storing the message data bits to be transmitted

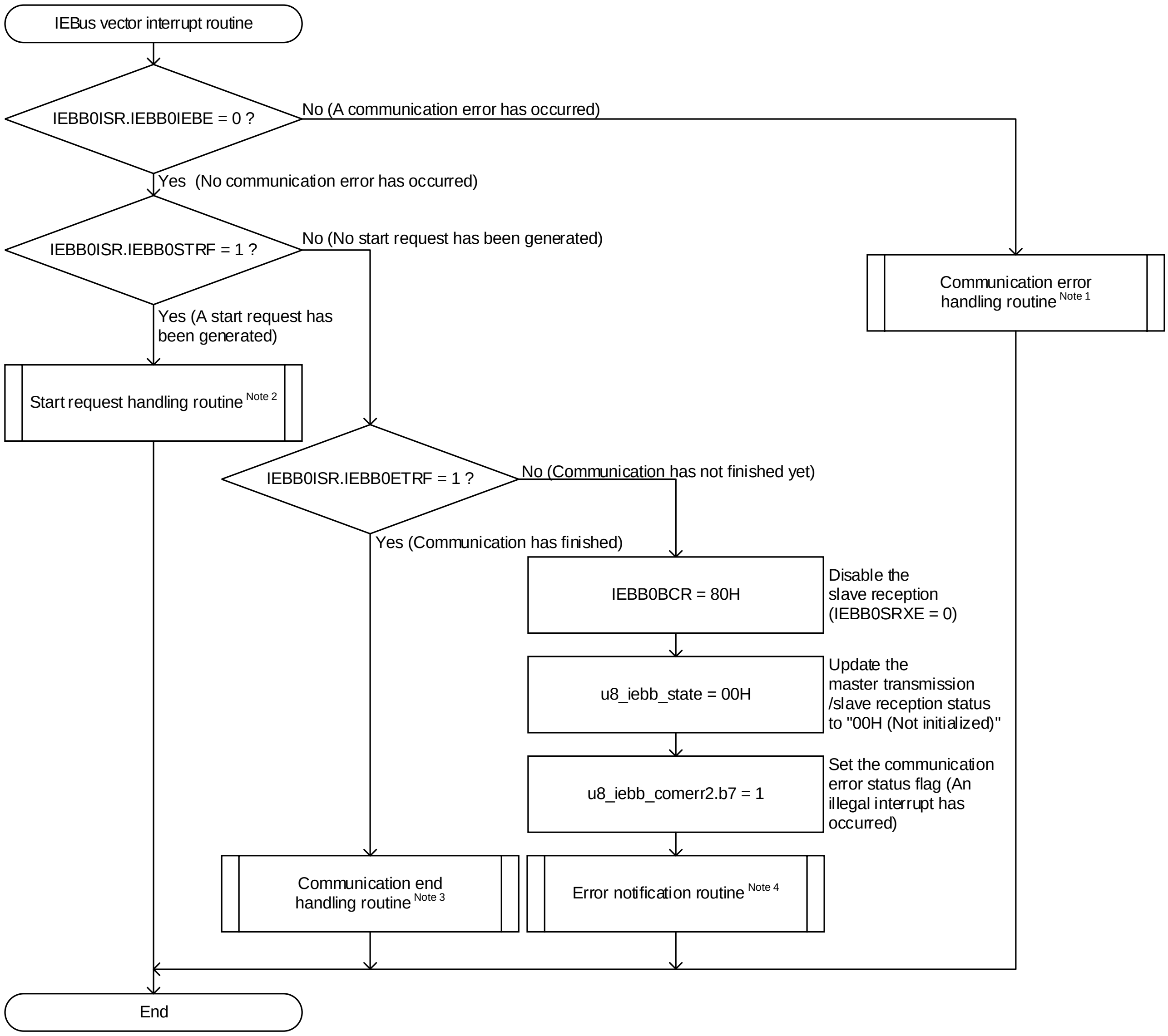



Note: See Figure 2-18.

**Figure 2-10 IEBB Transmission Start Routine**

## 2.5 IEBB Vector Interrupt

The flow of the IEBus vector interrupt routine is shown in Figure 2-11.

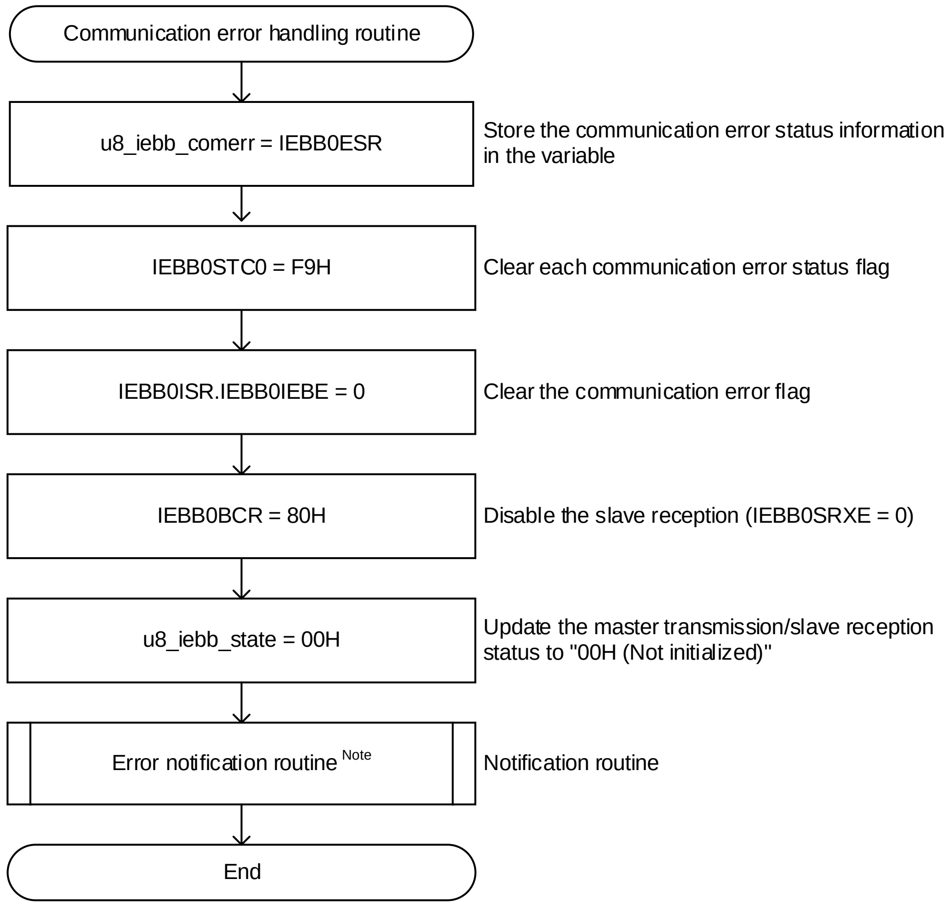



Notes: 1. See Figure 2-12.
2. See Figure 2-13.
3. See Figure 2-14.
4. See Figure 2-18.

**Figure 2-11 IEBus Vector Interrupt Routine**

### 2.5.1 Communication Error Handling Routine

The flow of the IEBus vector interrupt (Communication error) routine is shown in Figure 2-12.

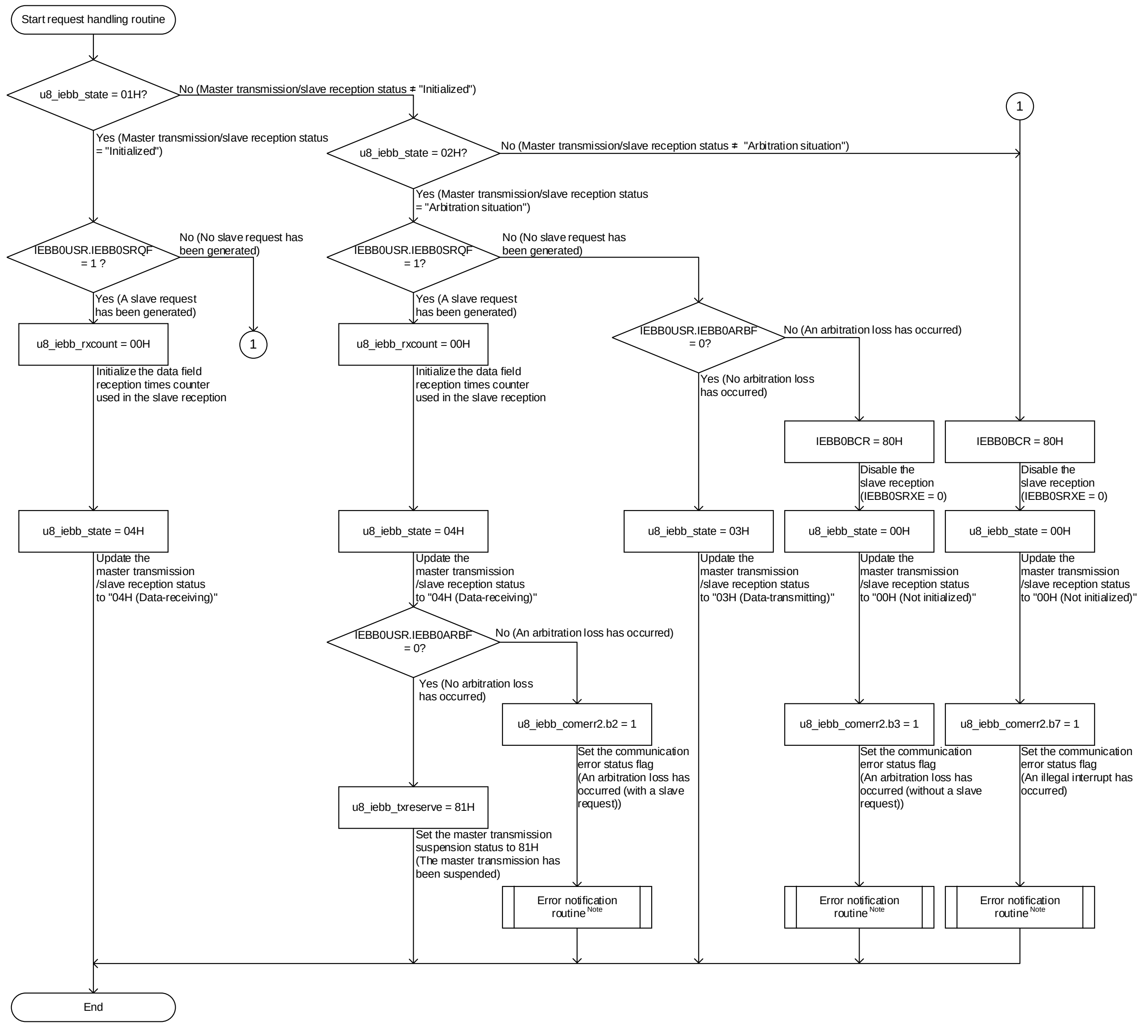



Note: See Figure 2-18.

**Figure 2-12 Communication Error Handling Routine**

### 2.5.2 Start Request Handling Routine

The flow of the IEBus vector interrupt (Start request) routine is shown in Figure 2-13.

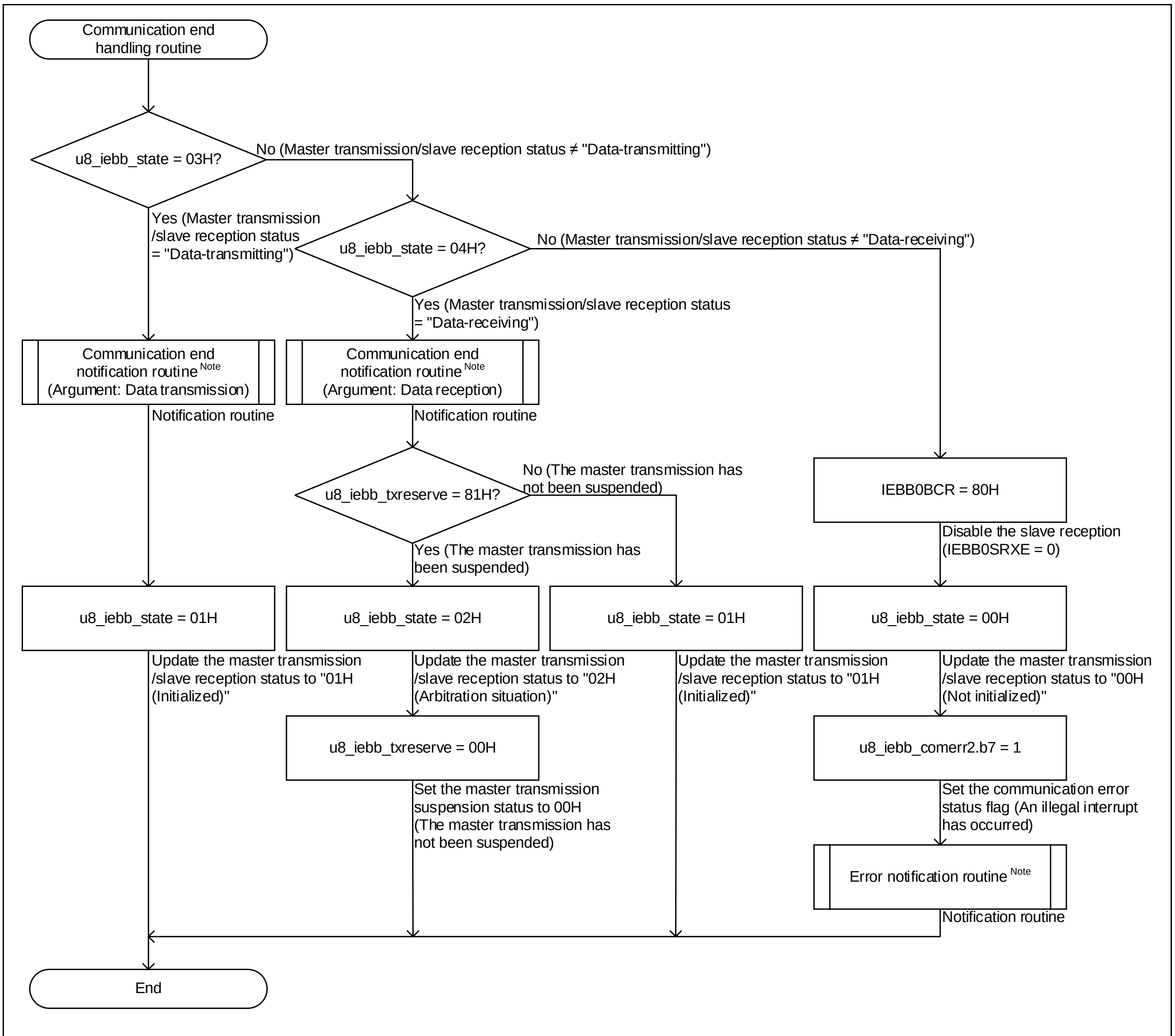



Note: See Figure 2-18.

**Figure 2-13 Start Request Handling Routine**

### 2.5.3 Communication End Handling Routine

The flow of the IEBus vector interrupt (Communication end) routine is shown in Figure 2-14.

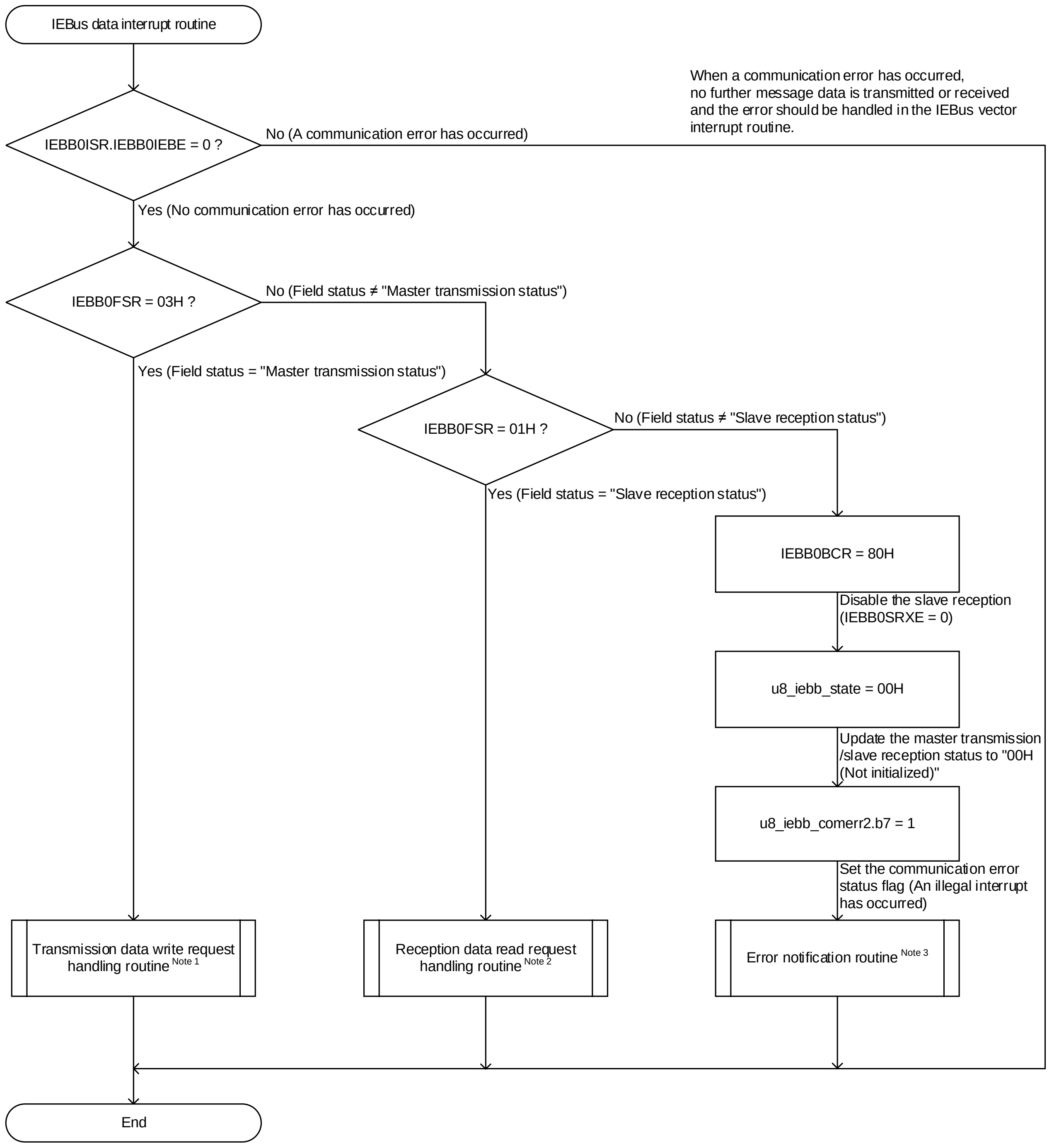



Note: See Figure 2-18.

**Figure 2-14 Communication End Handling Routine**

## 2.6 IEBus Data Interrupt

The flow of the IEBus data interrupt routine is shown in Figure 2-15. For your information, an IEBus data interrupt (Transmission data write request) does not occur if the transfer size is 1 byte.

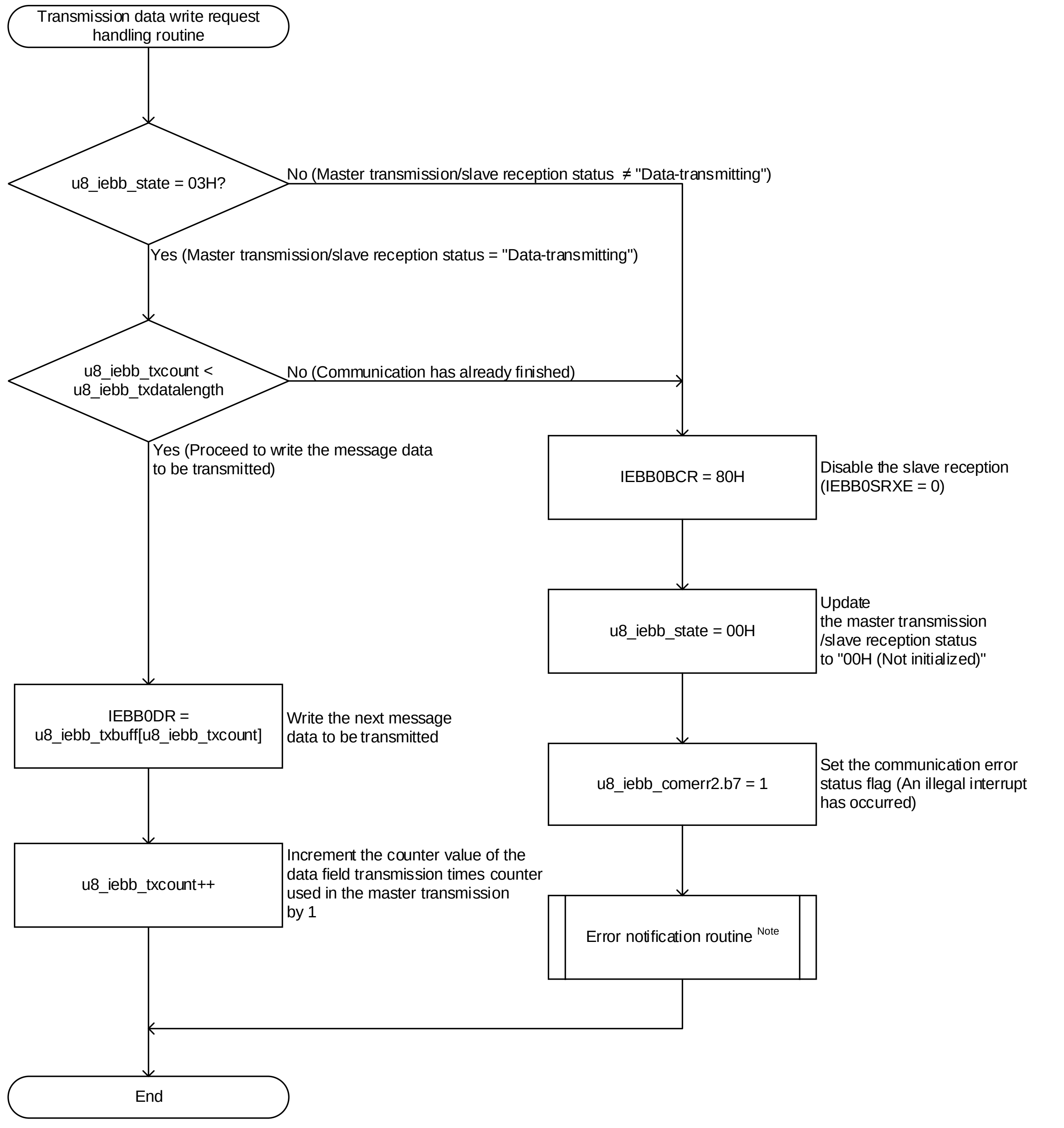



Notes: 1. See Figure 2-16.
2. See Figure 2-17.
3. See Figure 2-18.

**Figure 2-15 IEBus Data Interrupt Routine**

### 2.6.1 Transmission Data Write Request Handling Routine

The flow of the transmission data write request handling routine is shown in Figure 2-16.

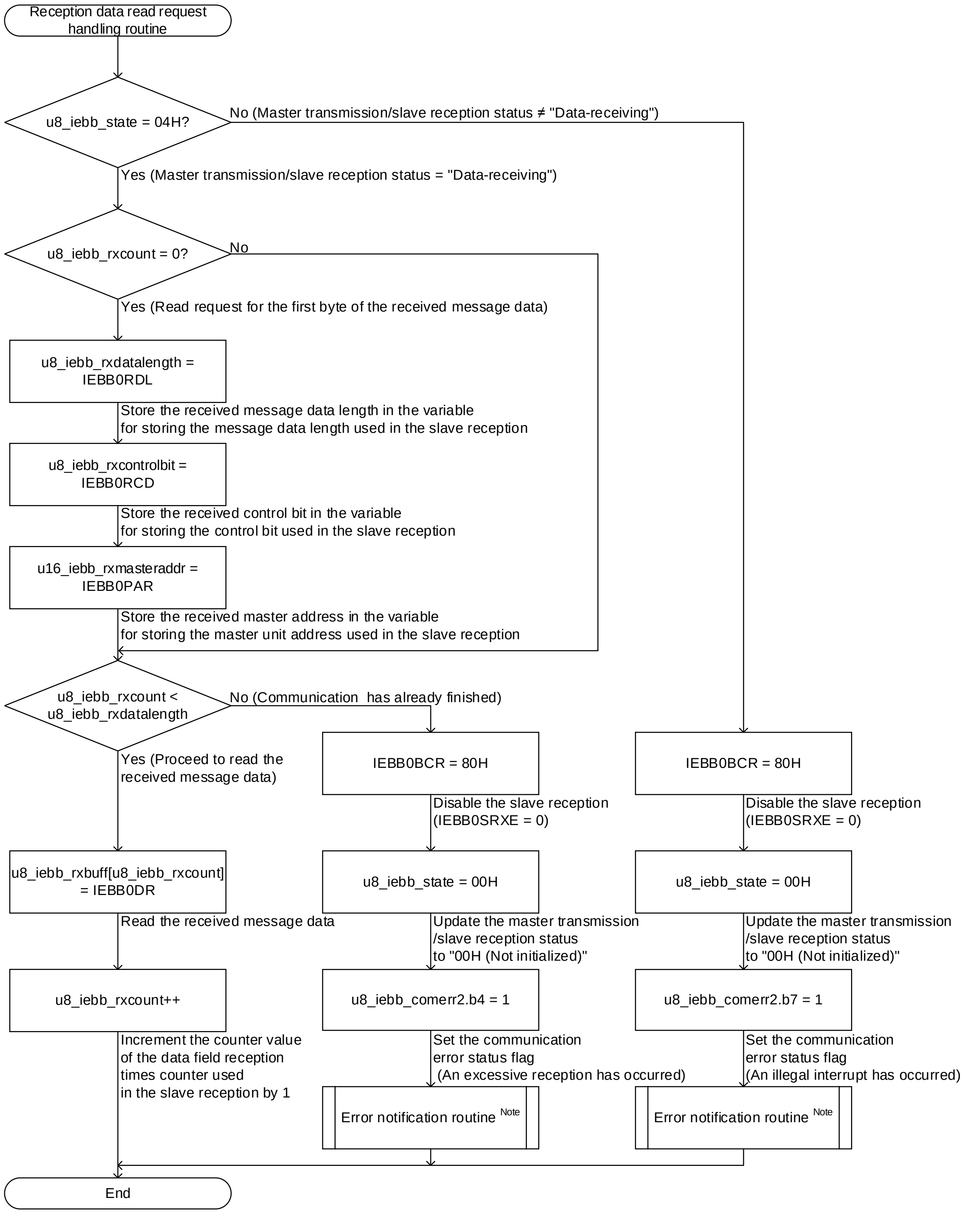



Note: See Figure 2-18.

**Figure 2-16 IEBus Transmission Data Write Request Handling Routine**

### 2.6.2 Reception Data Read Request Handling Routine

The flow of the reception data read request handling routine is shown in Figure 2-17.




Note: See Figure 2-18.

**Figure** 2-17 **IEBus Reception Data Read Request Handling Routine**

## 2.7 Notification Routines

The flows of the notification routines are shown in Figure 2-18. However, customers are required to modify each notification function for their system.

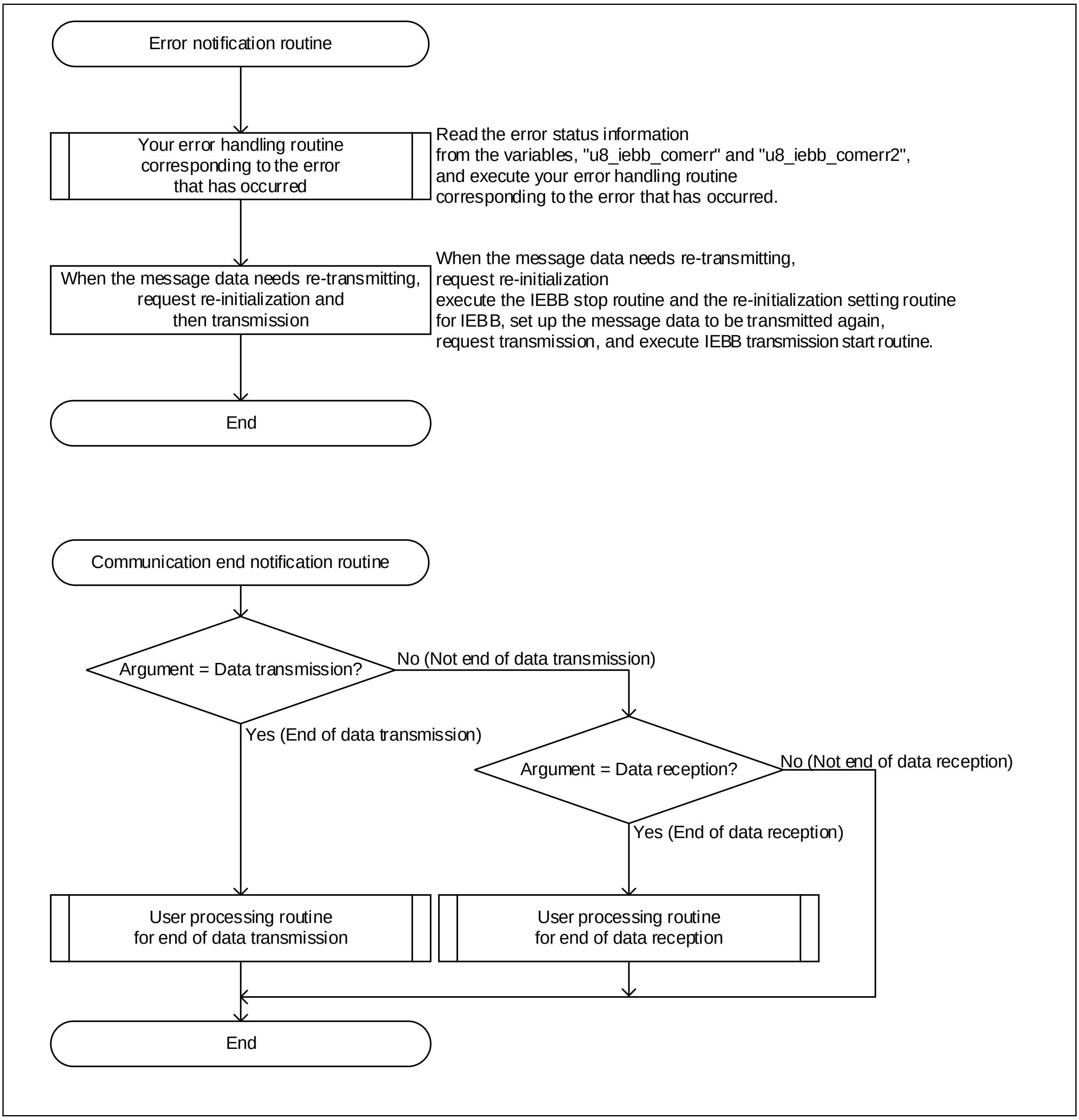



**Figure 2-18 Notification Routines**

## 2.8 Error Handling Examples

If an error such as a communication error has occurred, read the variables for storing the error status ("u8_iebb_comerr" and "u8_iebb_comerr2") and execute your error handling routine corresponding to the error that has occurred in the error notification routine. Table 2-1 and Table 2-2 show the values of error status and the error handling examples when each error has occurred.

**Table 2-1 Values of Error Status and Error Handling Examples When Each Error Has Occurred (Variable for Storing Error Status: u8_iebb_comerr)**

| u8_iebb_comerr2 | u8_iebb_comerr | | | | | | | | Error status | Error handling example |
|---|---|---|---|---|---|---|---|---|---|---|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | |
| XXH | * | * | * | * | * | 0 | 0 | 1 | An inter-third-party communication error has occurred | Execute the re-initialization setting routine for IEBB |
| | * | * | * | * | 1 | 0 | 0 | * | An overrun error has occurred | |
| | * | * | * | 1 | * | 0 | 0 | * | An underrun error has occurred | |
| | * | * | 1 | * | * | 0 | 0 | * | A NACK reception error has occurred | |
| | * | 1 | * | * | * | 0 | 0 | * | A parity error has occurred | |
| | 1 | * | * | * | * | 0 | 0 | * | A timing error has occurred | |

Remark: XXH: Any value
0: The corresponding bit is "0", 1: The corresponding bit is "1", *: The corresponding bit is either "0" or "1".

**Table 2-2 Values of Error Status and Error Handling Examples When Each Error Has Occurred (Variable for Storing Error Status: u8_iebb_comerr2)**

| u8_iebb_comerr | u8_iebb_comerr2 | | | | | | | | Error status | Error handling example |
|---|---|---|---|---|---|---|---|---|---|---|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | |
| 00H | * | 0 | 0 | * | * | * | * | 1 | Transmission has been requested in any state except "Initialized" | Execute the re-initialization setting routine for IEBB |
| | * | 0 | 0 | * | * | * | 1 | * | A command error has occurred (Transmission has been requested in unexpected condition) | Execute the re-initialization setting routine for IEBB |
| | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | An arbitration loss has occurred (with a slave request) as a result of transmission request | The transmission request has been cancelled and slave reception has been initiated already. Clear b2 of "u8_iebb_comerr2" to "0" in the error notification routine. If re-transmission is necessary, execute the IEBB transmission start routine after the reception finishes |
| | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | An arbitration loss has occurred (without a slave request) as a result of transmission request | Execute the re-initialization setting routine for IEBB. If re-transmission is necessary, execute the IEBB transmission start routine |
| | 1 | 0 | 0 | * | * | * | * | * | An illegal interrupt has occurred | Execute the re-initialization setting routine for IEBB |

Remark: 0: The corresponding bit is "0", 1: The corresponding bit is "1", *: The corresponding bit is either "0" or "1".

## Website and Support <website and support,ws>

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

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

## Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| Rev. 1.00 | Feb. 23, 2018 | | 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.**
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

**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: +44-1628-651-700, Fax: +44-1628-651-804

**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, 200333 P. R. China
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, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

**Renesas Electronics Korea Co., Ltd.**
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338





Colophon 7.0