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 (hereinafter referred to as IEBB). To transfer the transmission data and the reception data, the RL78/F15 data transfer controller (hereinafter referred to as DTC) is employed. For the setup procedures without employing the DTC, refer to "Setup Procedures for IEBB Master Transmission/Slave Reception (R01AN4096)".

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

Contents

1. Specifications of IEBB Master Transmission/Slave Reception Employing DTC........ 2
   1.1 Description of Memories Used .................................................................................. 4
   1.2 RAM Area Used for DTC (in IEBB Master Transmission/Slave Reception) ............. 6

2. Setup Procedures for IEBB Master Transmission/Slave Reception Employing DTC .. 7
   2.1 Initial Setting for IEBB ............................................................................................... 9
      2.1.1 Initial Setting for IEBB Ports ............................................................................... 10
      2.1.2 Initial Setting for IEBB Interrupts ....................................................................... 11
      2.1.3 Initial Setting for IEBB RAM .............................................................................. 12
      2.1.4 Initial Setting for DTC ......................................................................................... 13
   2.2 Re-initialization Setting for IEBB .............................................................................. 14
   2.3 IEBB Stop ................................................................................................................. 15
   2.4 IEBB Transmission Start ............................................................................................ 16
   2.5 IEBB Vector Interrupt ............................................................................................... 17
      2.5.1 Communication Error Handling Routine ........................................................... 18
      2.5.2 Start Request Handling Routine ........................................................................... 19
      2.5.3 Communication End Handling Routine ............................................................... 20
   2.6 DTC Setting .............................................................................................................. 21
      2.6.1 DTC Setting for Master Transmission ................................................................ 21
      2.6.2 DTC Setting for Slave Reception ....................................................................... 22
   2.7 Notification Routines ................................................................................................ 23
   2.8 Error Handling Examples ......................................................................................... 24
1. Specifications of IEBB Master Transmission/Slave Reception Employing DTC

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 provides examples in which the DTC is employed to perform the master transmission/slave reception with the control bit limited to "0FH (Write data)". Activated by an IEBus data interrupt signal, the DTC during the master transmission transfers the transmission data stored in RAM to the IEBB0DR register, whereas during the slave reception transfers the reception data stored in the IEBB0DR register to RAM for storing the reception data. The conditions for use of the DTC in the IEBB master transmission/slave reception are shown in Table 1-2.

<table>
<thead>
<tr>
<th>Items</th>
<th>Conditions for Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU/peripheral hardware clock (fCLK)</td>
<td>32 MHz</td>
</tr>
<tr>
<td>IEBB operation clock (fMCK)</td>
<td>8 MHz (fCLK/4)</td>
</tr>
<tr>
<td>IEBB communication mode</td>
<td>Mode 1 (32 bytes/frame), Individual communication</td>
</tr>
<tr>
<td>Unit address</td>
<td>100H</td>
</tr>
<tr>
<td>Slave unit address</td>
<td>180H</td>
</tr>
<tr>
<td>Master transmission data</td>
<td>Employing the DTC, transmission of 1 to 32 bytes of data is performed Note 1,2 (Control bit = &quot;0FH&quot;)</td>
</tr>
<tr>
<td>Slave reception data</td>
<td>Employing the DTC, reception of 1 to 32 bytes of data is performed Note 2 (Control bit = &quot;0FH&quot;)</td>
</tr>
</tbody>
</table>

Notes: 1. The master transmission is performed without employing the DTC if the transfer size of the transmission data is 1 byte because no IEBus data interrupt (Transmission data write request) signal is generated. Employing the DTC, the master transmission is performed as long as the transfer size is 2 to 32 bytes.
2. The conditions for use of the DTC are shown in Table 1-2.

<table>
<thead>
<tr>
<th>Item</th>
<th>Master transmission</th>
<th>Slave reception</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation source</td>
<td>IEBus data interrupt (Transmission data write request)</td>
<td>IEBus data interrupt (Reception data read request)</td>
</tr>
<tr>
<td>Transfer mode</td>
<td>Normal mode</td>
<td>Normal mode</td>
</tr>
<tr>
<td>Unit of transfers</td>
<td>8 bits</td>
<td>8 bits</td>
</tr>
<tr>
<td>Transfer source address</td>
<td>RAM (Memory where the transmission data are stored)</td>
<td>IEBB0DR register</td>
</tr>
<tr>
<td>Transfer destination address</td>
<td>IEBB0DR register</td>
<td>RAM (Memory where the reception data are stored)</td>
</tr>
<tr>
<td>Address control</td>
<td>Transfer source address: Incremented Transfer destination address: Fixed</td>
<td>Transfer source address: Fixed Transfer destination address: Incremented</td>
</tr>
<tr>
<td>Number of transfers</td>
<td>Length of message data to be transmitted minus 1 (1 to 31) Note 1</td>
<td>Length of message data to be received (1 to 32) Note 2</td>
</tr>
<tr>
<td>Interrupt that activates the DTC</td>
<td>The interrupt servicing on completion of DTC transfer is disabled</td>
<td>The interrupt servicing on completion of DTC transfer is disabled</td>
</tr>
</tbody>
</table>

Notes: 1. The transmission is performed without employing the DTC if the transfer size of the data is 1 byte because no IEBus data interrupt (Transmission data write request) signal is generated.
2. This document provides examples in which the slave reception is initiated with the number of transfers set to "32", which is the maximum number of transfers in mode 1.
Caution: When designing hardware, examine whether pull-up/pull-down resistors are necessary for the communication lines to IERXD and 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 employing the DTC are shown below.

Table 1-3 List of SFRs Used for IEBB Master Transmission/Slave Reception Employing DTC

<table>
<thead>
<tr>
<th>Register Name</th>
<th>In Use (✓)/ Not in Use (-)</th>
<th>Read/Write</th>
<th>Setting Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peripheral enable register 1 (PER1)</td>
<td>✓</td>
<td>Write</td>
<td>DTCEN = 1</td>
</tr>
<tr>
<td>Peripheral enable register 2 (PER2)</td>
<td>✓</td>
<td>Write</td>
<td>IEBUSEN = 1</td>
</tr>
<tr>
<td>IEBB0 bus control register (IEBB0BCR)</td>
<td>✓</td>
<td>Write</td>
<td>88H (at initializing)</td>
</tr>
<tr>
<td>IEBB0 power save register (IEBB0PSR)</td>
<td>✓</td>
<td>Write</td>
<td>80H</td>
</tr>
<tr>
<td>IEBB0 unit address register (IEBB0UAR)</td>
<td>✓</td>
<td>Write</td>
<td>100H</td>
</tr>
<tr>
<td>IEBB0 slave address register (IEBB0SAR)</td>
<td>✓</td>
<td>Write</td>
<td>180H</td>
</tr>
<tr>
<td>IEBB0 partner address register (IEBB0PAR)</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEBB0 reception slave address register (IEBB0RSA)</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEBB0 control data register (IEBB0CDR)</td>
<td>✓</td>
<td>Read/Write</td>
<td>0FH</td>
</tr>
<tr>
<td>IEBB0 message length register (IEBB0DLR)</td>
<td>✓</td>
<td>Read/Write</td>
<td>01H to 20H</td>
</tr>
<tr>
<td>IEBB0 transmission control data register (IEBB0TCD)</td>
<td>✓</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>IEBB0 reception control data register (IEBB0RCD)</td>
<td>✓</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>IEBB0 transmission message length register (IEBB0TDL)</td>
<td>✓</td>
<td>Read</td>
<td>01H to 20H</td>
</tr>
<tr>
<td>IEBB0 reception message length register (IEBB0RDL)</td>
<td>✓</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>IEBB0 clock selection register (IEBB0CKS)</td>
<td>✓</td>
<td>Write</td>
<td>04H</td>
</tr>
<tr>
<td>IEBB0 slave status register (IEBB0SSR)</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEBB0 unit status register (IEBB0USR)</td>
<td>✓</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>IEBB0 interrupt status register (IEBB0ISR)</td>
<td>✓</td>
<td>Read/Write</td>
<td>IEBB0IEBE = 0</td>
</tr>
<tr>
<td>IEBB0 error status register (IEBB0ESR)</td>
<td>✓</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>IEBB0 field status register (IEBB0FSR)</td>
<td>✓</td>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>IEBB0 success count register (IEBB0SCCR)</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEBB0 communication count register (IEBB0CCCR)</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEBB0 status clear register 0 (IEBB0STC0)</td>
<td>✓</td>
<td>Write</td>
<td>F9H</td>
</tr>
<tr>
<td>IEBB0 data register (IEBB0DR)</td>
<td>✓</td>
<td>Read/Write</td>
<td>Transmission/Reception data</td>
</tr>
<tr>
<td>IEBB0 data polarity select register (IEBB0DPS)</td>
<td>✓</td>
<td>Write</td>
<td>00H</td>
</tr>
<tr>
<td>DTC activation enable register 0 (DTCEN0)</td>
<td>✓</td>
<td>Write</td>
<td>00H</td>
</tr>
<tr>
<td>DTC activation enable register 1 (DTCEN1)</td>
<td>✓</td>
<td>Write</td>
<td>00H</td>
</tr>
<tr>
<td>DTC activation enable register 2 (DTCEN2)</td>
<td>✓</td>
<td>Write</td>
<td>00H</td>
</tr>
<tr>
<td>DTC activation enable register 3 (DTCEN3)</td>
<td>✓</td>
<td>Write</td>
<td>00H</td>
</tr>
<tr>
<td>DTC activation enable register 4 (DTCEN4)</td>
<td>✓</td>
<td>Write</td>
<td>00H</td>
</tr>
<tr>
<td>DTC activation enable register 5 (DTCEN5)</td>
<td>✓</td>
<td>Write</td>
<td>00H</td>
</tr>
<tr>
<td>DTC activation enable register 6 (DTCEN6)</td>
<td>✓</td>
<td>Write</td>
<td>20H</td>
</tr>
<tr>
<td>DTC base address register (DTCBAR)</td>
<td>✓</td>
<td>Write</td>
<td>FDH</td>
</tr>
</tbody>
</table>

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 1-4 List of RAM/Variables for Use in IEBB Master Transmission/Slave Reception Employing DTC

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>u8_iebb_state</td>
<td>The variable for storing the master transmission/slave reception status</td>
</tr>
<tr>
<td>00H</td>
<td>Not initialized</td>
</tr>
<tr>
<td>01H</td>
<td>Initialized</td>
</tr>
<tr>
<td>02H</td>
<td>Arbitration situation during the arbitration period</td>
</tr>
<tr>
<td>03H</td>
<td>Data-transmitting</td>
</tr>
<tr>
<td>04H</td>
<td>Data-receiving</td>
</tr>
<tr>
<td>u8_iebb_comerr</td>
<td>The variable for storing the communication error status which indicates the cause of communication error</td>
</tr>
<tr>
<td>b0</td>
<td>An inter-third-party communication error has occurred</td>
</tr>
<tr>
<td>b1, b2</td>
<td>-</td>
</tr>
<tr>
<td>b3</td>
<td>An overrun error has occurred</td>
</tr>
<tr>
<td>b4</td>
<td>An underrun error has occurred</td>
</tr>
<tr>
<td>b5</td>
<td>A NACK reception error has occurred</td>
</tr>
<tr>
<td>b6</td>
<td>A parity error has occurred</td>
</tr>
<tr>
<td>b7</td>
<td>A timing error has occurred</td>
</tr>
<tr>
<td>u8_iebb_comerr2</td>
<td>The variable for storing the communication error status 2 which indicates the cause of communication error</td>
</tr>
<tr>
<td>b0</td>
<td>Transmission has been requested in any state except &quot;Initialized&quot;</td>
</tr>
<tr>
<td>b1</td>
<td>A command error has occurred</td>
</tr>
<tr>
<td>b2</td>
<td>An arbitration loss has occurred during the arbitration period (with a slave request)</td>
</tr>
<tr>
<td>b3</td>
<td>An arbitration loss has occurred during the arbitration period (without a slave request)</td>
</tr>
<tr>
<td>b4-b6</td>
<td>-</td>
</tr>
<tr>
<td>b7</td>
<td>An illegal interrupt has occurred</td>
</tr>
<tr>
<td>u16_iebb_txslaveaddr</td>
<td>The variable for storing the communication-partner slave unit address used in the master transmission (<em>180H</em> in this document)</td>
</tr>
<tr>
<td>u16_iebb_rxmasteraddr</td>
<td>The variable for storing the master unit address used in the slave reception</td>
</tr>
<tr>
<td>u8_iebb_txcontrolbit</td>
<td>The variable for storing the control bit used in the master transmission (<em>0FH</em> in this document)</td>
</tr>
<tr>
<td>u8_iebb_rxcontrolbit</td>
<td>The variable for storing the control bit used in the slave reception (<em>0FH</em> in this document)</td>
</tr>
<tr>
<td>u8_iebb_txdatalength</td>
<td>The variable for storing the message data length used in the master transmission (<em>01H</em> to <em>20H</em> in this document)</td>
</tr>
<tr>
<td>u8_iebb_rxdatalength</td>
<td>The variable for storing the message data length used in the slave reception (<em>01H</em> to <em>20H</em> in this document)</td>
</tr>
<tr>
<td>u8_iebb_txreserve</td>
<td>The variable for storing the master transmission suspension status</td>
</tr>
<tr>
<td>00H</td>
<td>The master transmission has not been suspended</td>
</tr>
<tr>
<td>81H</td>
<td>The master transmission has been suspended</td>
</tr>
<tr>
<td>u8_iebb_txbuff[32]</td>
<td>The array for storing the message data bits used in the master transmission</td>
</tr>
<tr>
<td>u8_iebb_rxbuff[32]</td>
<td>The array for storing the message data bits used in the slave reception</td>
</tr>
<tr>
<td>dtc_ram</td>
<td>DTC RAM area[^Note]</td>
</tr>
</tbody>
</table>

Note: The 256-byte RAM area where the DTC vector table area and the DTC control data area are allocated.
1.2 RAM Area Used for DTC (in IEBB Master Transmission/Slave Reception)

The DTC uses a part of the RAM area, where the DTC vector table area and the DTC control data area are allocated. The area that the DTC uses is 256-byte area (xxx00H to xxxFFH) starting at the address indicated by the DTCBAR register. Hereinafter, this 256-byte area is referred to as "DTC RAM area".

Table 1-2 shows the DTC RAM area (an example in which the DTCBAR register is set to "FDH").

Note: Store the lower 8 bits of the start address of the DTC control data to be used. For use of the DTC control data 0, for example, store "40H" (FFD40H) to the address "FFD32H".

Figure 1-2 DTC RAM Area (Example in Which DTCBAR Register Is Set to "FDH")
2. Setup Procedures for IEBB Master Transmission/Slave Reception Employing DTC

This chapter describes the IEBB master transmission/slave reception processing routines (setup procedures) employing the DTC. Figure 2-1 shows the IEBB master transmission/slave reception processing routines. Figure 2-2 and Figure 2-3 show the timing.

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

Figure 2-1 IEBB Master Transmission/Slave Reception Processing Employing DTC
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 and afterwards, execute the DTC setting for the master transmission.
(4) Activated by the generation of an IEBus data interrupt (Transmission data write request) signal, the DTC transfers the next byte of the message data to be transmitted to the IEBB0DR register.
(5) Activated by the generation of an IEBus data interrupt (Transmission data write request) signal, the DTC transfers 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.

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 and afterwards, execute the DTC setting for the slave reception.
(3) Activated by the generation of an IEBus data interrupt (Reception data read request) signal, the DTC transfers the received message data read from the IEBB0DR to the array for storing the reception data.
(4) Activated by the generation of an IEBus data interrupt (Reception data read request) signal, the DTC transfers the last received message data (DATA(n)) read from the IEBB0DR register to the array for storing the reception data.
(5) An IEBus vector interrupt (Communication end) occurs and the slave reception finishes. Read the received master address, control bit, and message length bit.


Figure 2-2 Processing of Master Transmission Employing DTC (Timing Chart)

Figure 2-3 Processing of Slave Reception Employing DTC (Timing Chart)
### 2.1 Initial Setting for IEBB

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

---

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

1. **Initial setting routine for the IEBB**
   - `PER2.IEBUSEN = 1`
     - Enable the IEBB input clock supply
   - `IEBB0CR.IEBBB0PW = 0`
     - Stop the IEBB unit operation
   - `IEBB0CKS = 04H`
     - Frequency of the IEBB operation clock (fMCK) = fCLK/4 (8 MHz = 32 MHz/4)
   - `IEBB0PSR = 80H`
     - Enable the operation clock and select communication mode 1
   - `IEBB0UAR = 0100H`
     - Set the unit address (to "100H")
   - `IEBB0DPS = 00H`
     - Select the negative logic to the polarity of input/output to/from the IEBB
   - **Initial setting routine for the IEBB ports**
     - Initial setting routine for the ports for use in the IEBB communication
   - **Initial setting routine for the IEBB interrupts**
     - Initial setting routine for the interrupts for use in the IEBB communication
   - **Initial setting routine for the IEBB RAM**
     - Initial setting routine for the memory (RAM) for use in the IEBB communication
   - **Initial setting routine for the DTC**
     - Initial setting routine for the DTC for use in the IEBB communication
   - `IEBB0STC0 = F9H`
     - Clear each communication error status flag
   - `u8_iebb_state = 01H`
     - Update the master transmission/slave reception status to "01H (Initialized)"
   - `IEBB0BCR = 88H`
     - Enable the IEBB operation (IEBB0PW = 1)
     - Enable the slave reception (IEBB0SRXE = 1)
   - **End**

Notes:
1. See Figure 2-5.
2. See Figure 2-6.
3. See Figure 2-7.
4. See Figure 2-8.
2.1.1 Initial Setting for IEBB Ports

In this document, IETXD and 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.

```
<table>
<thead>
<tr>
<th>IEBB pin setting</th>
<th>PIOR9</th>
<th>PMn</th>
<th>Pn</th>
<th>POMn</th>
<th>PITHLn</th>
</tr>
</thead>
<tbody>
<tr>
<td>P60/IERXD</td>
<td>PIOR9 = 0</td>
<td>PM60 = 1</td>
<td>-</td>
<td>-</td>
<td>PITHL60 = x</td>
</tr>
<tr>
<td>P61/IETXD</td>
<td>PIOR9 = 0</td>
<td>PM61 = 0</td>
<td>P61 = 0</td>
<td>POM61 = x</td>
<td>-</td>
</tr>
<tr>
<td>P50/IERXD Note</td>
<td>PIOR9 = 1</td>
<td>PM50 = 1</td>
<td>-</td>
<td>-</td>
<td>PITHL50 = x</td>
</tr>
<tr>
<td>P51/IETXD Note</td>
<td>PIOR9 = 1</td>
<td>PM51 = 0</td>
<td>P51 = 0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
```

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 level of the IEBus vector interrupt is specified as level 1 and the IEBus data interrupt servicing is disabled.

```
Initial setting routine for the IEBB Interrupts

PR01H.IEBBTVP0 = 1
PR11H.IEBBTVP1 = 0

Set the priority level of the IEBus vector interrupt to level 1

MK1H.IEBBTVMK = 0

Enable the IEBus vector interrupt servicing

MK1H.IEBBTD Mk = 1

Disable the IEBus data interrupt servicing

IF1H.IEBBTVIF = 0
IF1H.IEBBTDIF = 0

Clear the IEBus vector interrupt flag and the IEBus data interrupt flag

End
```

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 employing the DTC is shown in Figure 2-7.

Figure 2-7 Initial Setting Routine for IEBB RAM
2.1.4 Initial Setting for DTC

The flow of the initial setting routine for the DTC is shown in Figure 2-8. This document provides an example in which the start address of the DTC RAM area is specified as "FFD00H".

![Figure 2-8 Initial Setting Routine for DTC]
### 2.2 Re-initialization Setting for IEBB

The flow of the re-initialization setting routine for the IEBB is shown in Figure 2-9.

---

**Figure 2-9 Re-initialization Setting Routine for IEBB**

```
Re-Initialization setting routine for the IEBB

PER2.IEBUSEN = 1
  Enable the IEBB input clock supply

IEBB0CR.IEBBB0PW = 0
  Stop the IEBB unit operation

IEBB0CKS = 04H
  Frequency of the IEBB operation clock \( f_{MCK} = f_{CLK}/4 \) (8 MHz = 32 MHz/4)

IEBB0PSR = 80H
  Enable the operation clock and select communication mode 1

IEBB0UAR = 0100H
  Set the unit address (to "100H")

IEBB0DPS = 00H
  Select the negative logic to the polarity of input/output to/from the IEBB

Initial setting routine for the IEBB ports Note 1
  Initial setting routine for the ports for use in the IEBB communication

Initial setting routine for the IEBB interrupts Note 2
  Initial setting routine for the interrupts for use in the IEBB communication

u8_iebb_state = 01H
  Update the master transmission/slave reception status to "01H (Initialized)"

IEBB0BCR = 88H
  Enable the IEBB operation (IEBB0PW = 1)
  Enable the slave reception (IEBB0SRXE = 1)

IEBB0STC0 = F9H
  Clear each communication error status flag

u8_iebb_comerr = 00H
  Initialize each variable for storing the communication error status which indicates the cause of the communication error

u8_iebb_comerr2 = 00H
  Initialize the variable for storing the master transmission suspension status

IEBB0STC0 = F9H

PER.DTCEN = 1
  Enable the DTC input clock supply

DTCEN6.DTCEN65 = 0
  Disable the DTC activation by the IEBus data interrupt signal

u8_iebb_state = 01H
  Update the master transmission/slave reception status to "01H (Initialized)"

IEBB0BCR = 88H
  Enable the IEBB operation (IEBB0PW = 1)
  Enable the slave reception (IEBB0SRXE = 1)

End
```

---

**Notes:**
1. See Figure 2-5.
2. See Figure 2-6.
### 2.3 IEBB Stop

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

![Figure 2-10 IEBB Stop Routine](image)

- **IEBB stop routine**
  - MK1H.IEBBTVMK = 1
  - MK1H.IEBBTDMK = 1
    - Disable the IEBus vector interrupt servicing and the IEBus data interrupt servicing
  - IEBB0BCR = 00H
    - Disable the slave reception (IEBB0SRXE = 0)
  - IEBB0PSR.IEBB0CLKE = 0
    - Stop the IEBB operation clock
  - PER2.IEBUSEN = 0
    - Stop the IEBB input clock supply
  - DTCE0 = 00H
    - Disable the DTC activation by all the DTC activation sources
  - DTCE1 = 00H
  - DTCE2 = 00H
  - DTCE3 = 00H
  - DTCE4 = 00H
  - DTCE5 = 00H
  - DTCE6 = 00H
  - PER1.DTCEN = 0
    - Stop the DTC input clock supply
  - u8_iebb_state = 00H
    - Update the master transmission/slave reception status to "00H (Not initialized)"
  - End
### 2.4 IEBB Transmission Start

The flow of the IEBB transmission start routine is shown in Figure 2-11.

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: The master transmission is performed without employing the DTC if the transfer size of the data is 1 byte, because no IEBus data interrupt (Transmission data write request) is generated.

---

#### Figure 2-11 IEBB Transmission Start Routine
2.5 IEBB Vector Interrupt

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

![Figure 2-12 IEBus Vector Interrupt Routine](image)

Notes:
1. See Figure 2-13.
2. See Figure 2-14.
3. See Figure 2-15.
4. See Figure 2-18.
2.5.1 Communication Error Handling Routine
The flow of the IEBus vector interrupt (Communication error) routine is shown in Figure 2-13.

![Diagram of Communication Error Handling Routine]

Note: See Figure 2-18.

Figure 2-13 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 2-14.

Notes: 1. See Figure 2-17.
2. See Figure 2-16.
3. See Figure 2-18.
### 2.5.3 Communication End Handling Routine

The flow of the IEBus vector interrupt (Communication end) routine is shown in Figure 2-15.

![Figure 2-15 Communication End Handling Routine](image-url)

Notes: 1. See Figure 2-16.
2. See Figure 2-18.
2.6 DTC Setting

The flows of the DTC setting routines for the IEBB master transmission/slave reception employing the DTC are shown in Figure 2-16 and Figure 2-17.

2.6.1 DTC Setting for Master Transmission

![DTC Setting Routine for Master Transmission Diagram]

Remark: j = 0 to 23

Figure 2-16 DTC Setting Routine for Master Transmission
2.6.2 DTC Setting for Slave Reception

- **DTC setting routine for slave reception**
  - **DTCEN6.DTCEN65 = 0**
    - Disable the DTC activation by the IEBus data interrupt signal
  - **IF1H.IEBBDTIF = 0**
    - Clear the IEBus data interrupt request flag
  - **Set up the DTC vector table [50]**
    - Store the start address (lower 8 bits) of the DTC control data j to be used in the DTC vector table [50], which is corresponding to the IEBus data interrupt.
  - **DTCCTj = 08H**
    - Normal mode (MODE = 0)
    - 8-bit transfer (SZ = 0)
    - Chain transfer disabled (CHNE = 0)
    - Transfer destination address: Incremented after the data transfer (DAMOD = 1)
    - Transfer source address: Fixed (SAMOD = 0)
  - **DTBLsj = 01H**
    - The size of the data block to be transferred by one DTC activation: 1 byte
  - **DTCTlj = 20H**
    - Set to the number of the DTC transfers (32 times)
  - **DTLdj = any value**
    - Initialize the register with any desired value
  - **DTSARj = 07D9H**
    - Transfer source address:
      - Set to the IEBB0DR address (lower 16 bits)
  - **DTDARj = (unsigned short)&u8_iebb_rxbuff[0]**
    - Transfer destination address:
      - Set to the start address (lower 16 bits) of the array for storing the message data bits used in the slave reception
  - **DTCEN6.DTCEN65 = 1**
    - Enable the DTC activation by the IEBus data interrupt signal

Remark: 
- **j = 0 to 23**

Figure 2-17 DTC Setting Routine for Slave Reception
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.

- **Error notification routine**
  - Your error handling routine corresponding to the error that has occurred
  - Read the error status information from the variables, “u8_iebb_comerr” and “u8_iebb_comerr2”, and execute your error handling routine corresponding to the error that has occurred.

- **Communication end notification routine**
  - Argument = Data transmission?
    - Yes (End of data transmission)
    - No (Not end of data transmission)
      - Argument = Data reception?
        - Yes (End of data reception)
          - User processing routine for end of data reception
        - No (Not end of data reception)
          - User processing routine for end of data transmission

- **When the message data needs re-transmitting, request re-initialization and then transmission**

- **End**

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)

<table>
<thead>
<tr>
<th>u8_iebb_comerr2</th>
<th>u8_iebb_comerr</th>
<th>Error status</th>
<th>Error handling example</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXH</td>
<td>* * * * * * 0 0 1</td>
<td>An inter-third-party communication error has occurred</td>
<td>Execute the re-initialization setting routine for IEBB</td>
</tr>
<tr>
<td></td>
<td>* * * 1 0 0 *</td>
<td>An overrun error has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* * 1 * 0 0 *</td>
<td>An underrun error has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* 1 * * 0 0 *</td>
<td>A NACK reception error has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 * * * 0 0 *</td>
<td>A timing error has occurred</td>
<td></td>
</tr>
</tbody>
</table>

Remark: 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)

<table>
<thead>
<tr>
<th>u8_iebb_comerr</th>
<th>u8_iebb_comerr2</th>
<th>Error status</th>
<th>Error handling example</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H</td>
<td>* 0 0 * * * 1</td>
<td>Transmission has been requested in any state except &quot;Initialized&quot;</td>
<td>Execute the re-initialization setting routine for IEBB</td>
</tr>
<tr>
<td></td>
<td>* 0 0 * * * 1</td>
<td>A command error has occurred (Transmission has been requested in unexpected condition)</td>
<td>Execute the re-initialization setting routine for IEBB</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0 1 0 0</td>
<td>An arbitration loss has occurred (with a slave request) as a result of transmission request</td>
<td>The transmission request has been cancelled and slave reception has been initiated already. Clear b2 of &quot;u8_iebb_comerr2&quot; to &quot;0&quot; in the error notification routine. If re-transmission is necessary, execute the IEBB transmission start routine after the reception finishes.</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 1 0 0 0</td>
<td>An arbitration loss has occurred (without a slave request) as a result of transmission request</td>
<td>Execute the re-initialization setting routine for IEBB. If re-transmission is necessary, execute the IEBB transmission start routine.</td>
</tr>
<tr>
<td></td>
<td>1 0 0 * * * *</td>
<td>An illegal interrupt has occurred</td>
<td>Execute the re-initialization setting routine for IEBB</td>
</tr>
</tbody>
</table>

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/

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

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rev. 1.00</td>
<td>Mar. 30, 2018</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>
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.
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 or any of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depend 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, industrial robots, etc.

- **High Quality**: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; underwater repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.) and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 other any appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

11. This document shall not be reproduced, reprinted 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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.