Abstract

This document describes EEPROM control using UART2 special mode 1 (I2C mode). Special mode 1 in the R32C/118 Group has seven channels (UART0 to UART6). When using a channel other than UART2, refer to the UARTi-related register in the User’s Manual: Hardware for setting details.

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 ........................................................................................................... 3
2. Operation Confirmation Conditions ............................................................................ 6
3. Reference Application Notes .................................................................................... 6
4. Hardware .................................................................................................................. 6
   4.1 Pins Used ............................................................................................................. 6
5. Software .................................................................................................................... 7
   5.1 Operation Overview ............................................................................................... 8
   5.1.1 Operation During Write Mode ........................................................................... 8
   5.1.2 Operation During Read Mode ............................................................................ 10
5.2 Constants ............................................................................................................... 12
5.3 Structure/Union List .............................................................................................. 13
5.4 Variables ............................................................................................................... 14
5.5 Functions .............................................................................................................. 14
5.6 Function Specifications ......................................................................................... 15
5.7 Flowcharts ......................................................................................................... 20
   5.7.1 Main Processing ............................................................................................... 20
   5.7.2 UART2 Initial Setting ....................................................................................... 22
   5.7.3 Master Control Start Processing ..................................................................... 23
   5.7.4 Start Condition/Stop Condition Detection Interrupt Handling ......................... 24
   5.7.5 Start Condition Detection Processing .............................................................. 25
   5.7.6 Stop Condition Detection Processing .............................................................. 26
   5.7.7 Restart Condition Detection Processing .......................................................... 28
   5.7.8 UART2 Transmit Interrupt Handling ............................................................... 29
   5.7.9 Memory Address Transmission ...................................................................... 30
   5.7.10 Master Transmission .................................................................................... 31
   5.7.11 Master Reception ........................................................................................... 32
   5.7.12 INT0 and INT1 Initial Setting ........................................................................ 33
   5.7.13 INT0 Interrupt Handling ............................................................................... 33
   5.7.14 INT1 Interrupt Handling ............................................................................... 33
6. Sample Code ............................................................................................................ 34
7. Reference Documents .............................................................................................. 34
1. Specifications

This chapter describes EEPROM control using UART2 special mode 1 (I2C mode). Writing data to the EEPROM (write mode) and reading data from the EEPROM (read mode) are performed. The Renesas Electronics R1EX24xxx Series EEPROM is used in the explanation below.

The settings below comply with I2C-bus communication protocols.

Settings
- Transfer rate: Approx. 350 kbps
- Transfer data length: 1 to 256 bytes (not including the device address word and memory address)
- Single-master communication (multi-master is not supported)
- Restart condition generation applicable when in read mode

Notes:
1. The setting value is 378 kbps. (When clock synchronization is enabled, a sampling delay of 1 to 1.5 cycles of the U2BRG count source + the noise filter length occurs, high recognition of the SCL clock is delayed, and the high width of the SCL clock widens. Therefore, the actual SCL clock transfer rate will be slower than what is set. As clock synchronization is enabled in this application example, the actual transfer rate will be approx. 350 kbps (reference values: pull-up voltage is 5 V, pull-up resistor value is 1 kΩ).) This is the transfer rate in Standard-Mode and Fast-Mode.
2. The device address word is comprised of a 4-bit device code, a 3-bit device address code, and a 1-bit read/write code.
Table 1.1 lists the Peripheral Functions and Their Applications, Figure 1.1 shows a Connection Example, and Figure 1.2 shows the Communication Format.

### Table 1.1 Peripheral Functions and Their Applications

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial interface (UART2)</td>
<td>Communication with the EEPROM</td>
</tr>
<tr>
<td>INT0 interrupt</td>
<td>Change mode to write mode</td>
</tr>
<tr>
<td>INT1 interrupt</td>
<td>Change mode to read mode</td>
</tr>
</tbody>
</table>

**Figure 1.1 Connection Example**

**Notes:**
1. In this application note, the WP pin of the EEPROM is not controlled by the MCU.
2. The SCL2 and SDA2 pins in the R32C/118 Group are set to N-channel open drain. Enable the pull-up resistors according to the user system.
3. Set the device address to pins A0 to A2. Connect VCC or VSS depending on the set device address.
### Figure 1.2 Communication Format

#### Write mode

<table>
<thead>
<tr>
<th>Device address word</th>
<th>1st memory address</th>
<th>2nd memory address</th>
<th>Write data (1st byte)</th>
<th>Write data (32nd byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>1 0 1 0</td>
<td>0 0 0 1</td>
<td>A 0 0 0 1 0 0 1</td>
<td>0 0 0 1 0 0 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1</td>
<td>0 1 1 0</td>
<td>A 0 0 0 1 0 0 1</td>
<td>0 0 0 1 0 0 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1</td>
<td>1 0 0 0</td>
<td>A 0 0 0 1 0 0 1</td>
<td>0 0 0 1 0 0 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1</td>
<td>1 0 0 0</td>
<td>A 0 0 0 1 0 0 1</td>
<td>0 0 0 1 0 0 1 0</td>
</tr>
</tbody>
</table>

#### Read mode

<table>
<thead>
<tr>
<th>Device address word</th>
<th>1st memory address</th>
<th>2nd memory address</th>
<th>Device address word</th>
<th>Read data (1st byte)</th>
<th>Read data (256th byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>1 0 1 0</td>
<td>0 0 0 1</td>
<td>ST</td>
<td>1 0 1 0</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1</td>
<td>0 1 1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 0 0 1</td>
<td>1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 0 0 1</td>
<td>1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ST: Start condition</th>
<th>SP: Stop condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST: Restart condition</td>
<td>@: EEPROM device address</td>
</tr>
<tr>
<td>*: Don't care bit</td>
<td>D7 to D0: Data</td>
</tr>
<tr>
<td>W: Write 0</td>
<td>R: Read 1</td>
</tr>
<tr>
<td>ACK: Acknowledge 0</td>
<td>NACK: Not acknowledge 1</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>R5F64189DFD (R32C/118 Group)</td>
</tr>
<tr>
<td>Device used</td>
<td>R1EX24032ASAS0A</td>
</tr>
</tbody>
</table>
| 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 | 5 V                                                                     |
| Integrated development environment | Renesas Electronics Corporation  
|                  | High-performance Embedded Workshop Version 4.08                        |
| C compiler      | Renesas Electronics Corporation                                      |
|                  | R32C/100 Series C Compiler V.1.02 Release 01                            |
| Compile options | -D__STACKSIZE__=0X300  
|                  | -D__ISTACKSIZE__=0X300  
|                  | -DVVECTORADR=0x0FFFFFFBDC  
|                  | -c-finfo -dir "$(CONFIGDIR)"                                             |
| (Default setting is used in the integrated development environment.) | |
| Operating mode  | Single-chip mode                                                        |
| Sample code version | 1.00                                                                   |

3. Reference Application Notes

Application notes associated with this application note are listed below. Refer to these application notes for additional information.

- R32C/100 Series Configuring PLL Mode (REJ05B1221)
- M16C Family, R8C Family I2C-bus Interface Using UARTi Special Mode 1 (REJ05B1349)
- R32C/100 Series I2C-bus Interface Using UARTi Special Mode 1 (Master Transmit/Receive) (REJ05B1395)

4. Hardware

4.1 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

Table 4.1 Pins Used and Their Functions

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P7_0/SDA2</td>
<td>I/O</td>
<td>Data input and output in I2C mode</td>
</tr>
<tr>
<td>P7_1/SCL2</td>
<td>Output</td>
<td>Clock output in I2C mode</td>
</tr>
<tr>
<td>P8_2/INT0</td>
<td>Input</td>
<td>Input for INT0 interrupt</td>
</tr>
<tr>
<td>P8_3/INT1</td>
<td>Input</td>
<td>Input for INT1 interrupt</td>
</tr>
</tbody>
</table>
5. Software

This chapter describes how to write 256 bytes to the EEPROM (write mode) or read 256 bytes from the EEPROM (read mode) using the R32C/118 Group MCU. In write mode, 32 bytes of data (1 page) are written eight times (= 256 bytes). The mode changes to write mode when an INT0 interrupt is generated. The mode changes to read mode when an INT1 interrupt is generated.

The settings below are used with the serial interface (UART2) in special mode 1 (I^2C mode).

Settings
- Use I^2C mode.
- Use the internal clock for the transfer clock.
- Use f1 for the U2BRG count source.
- Set the SDA2 and SCL2 pins as N-channel open drain output.
- Use MSB first as the transfer format.
- Use transmission completed (TXEPT is 1) as the UART2 transmit interrupt source.
- Select clock delayed for the clock-phase setting.
- Use seven to eight cycles of the U2BRG count source for the SDA2 digital delay time.
- Use clock synchronization.
- Do not use the function that performs SCL2 wait output.
- Do not use the function that performs SDA2 output stop.
- Use the start condition/stop condition detection interrupt.
- Use the UART2 transmit interrupt.
- Do not use the UART2 receive interrupt.
- Set the transfer rate to approx. 378 kbps.

Transfer rate formula:
Transfer rate = U2BRG count source ÷ (2 × (U2BRG register setting value + 1))
= 25 MHz (f1) ÷ (2 × (32 + 1))
≈ 378.788 kbps
5.1 Operation Overview

5.1.1 Operation During Write Mode

(1) Initial setting
This operation initializes the system clock, UART2-related SFRs, and INT interrupt-related SFRs. When an INT0 interrupt is generated, write mode is set to the variable (mode) on the RAM, and the start condition is generated.

(2) Start condition/stop condition detection interrupt
When the start condition generation is completed, a start condition/stop condition detection interrupt is generated. Set the device address word to the U2TB register, and start transmission.

(3) UART2 transmit interrupt
A UART2 transmit interrupt is generated at the falling edge of the 9th bit of the SCL clock. In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the first memory address to the U2TB register.

(4) UART2 transmit interrupt
In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the second memory address to the U2TB register.

(5) UART2 transmit interrupt
In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the write data to the U2TB register. Repeat step (5) until 32 bytes of data have been transmitted.

(6) UART2 transmit interrupt
A stop condition is generated in the UART2 transmit interrupt handling for every 32 bytes of data transmitted.

(7) Start condition/stop condition detection interrupt
When a stop condition is detected, a start condition/stop condition detection interrupt is generated, and stop condition detection processing is performed.

(8) Wait until data is written to EEPROM
In the stop condition detection processing, set a software loop to 5 ms in order for data to be written to the EEPROM. If write data remains after the stop condition detection processing, a start condition is generated.

Repeat steps (2) through (8) until 256 bytes have been transmitted.
After 256 bytes have been transmitted, the CPU waits for an INT0 interrupt or INT1 interrupt to be generated.
Figure 5.1 shows the Timing Diagram in Write Mode.

![Timing Diagram in Write Mode](image)

@: EEPROM device address
*: Don't care bit
a11 to a0: EEPROM memory address
D7 to D0: Data
### 5.1.2 Operation During Read Mode

1. **Initial setting**
   - This operation initializes the system clock, UART2-related SFRs, and INT interrupt-related SFRs. When an INT1 interrupt is generated, read mode is set to the variable (mode) on the RAM, and the start condition is generated.

2. **Start condition/stop condition detection interrupt**
   - When the start condition generation is completed, a start condition/stop condition detection interrupt is generated. Set the device address word to the U2TB register, and start transmission. (At this point, the 8th bit of the device address word is 0 (write code).)

3. **UART2 transmit interrupt**
   - A UART2 transmit interrupt is generated at the falling edge of the 9th bit of the SCL clock. In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the first memory address to the U2TB register.

4. **UART2 transmit interrupt**
   - In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the second memory address to the U2TB register.

5. **UART2 transmit interrupt**
   - Generate a restart condition in the UART2 transmit interrupt handling.

6. **Start condition/stop condition detection interrupt**
   - After a restart condition generation is completed, a start condition/stop condition detection interrupt is generated. Set the device address word to the U2TB register, and start transmission. (At this point, the 8th bit of the device address word is 1 (read code).)

7. **UART2 transmit interrupt**
   - In order to receive the next data, set dummy data and ACK data to the U2TB register.

8. **UART2 transmit interrupt**
   - Read the read data from the U2RB register, and write it to the receive data buffer. In order to receive the next data, set dummy data and ACK data to the U2TB register. Repeat step (8) until 254 bytes of data have been received.

9. **UART2 transmit interrupt**
   - Read the read data from the U2RB register, and write it to the receive data buffer. In order to receive the 256th byte of data (the last data), set dummy data and NACK data to the U2TB register.

10. **UART2 transmit interrupt**
    - Read the read data from the U2RB register, and write it to the receive data buffer. Generate a stop condition in the UART2 transmit interrupt handling.

11. **Start condition/stop condition detection interrupt**
    - When a stop condition is detected, a start/stop condition detection interrupt is generated, and stop condition detection processing is performed. Wait for an INT0 or INT1 interrupt to be generated.
Figure 5.2 shows the Timing Diagram in Read Mode.

Figure 5.2 Timing Diagram in Read Mode
5.2 Constants

Table 5.1 lists the Constants Used in the Sample Code.

Table 5.1 Constants Used in the Sample Code

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
</table>
| DEVICE_ADDR_WORD    | 1010 0000b    | Device address word  
b7 to b4: Device code  
b3 to b1: Slave address  
b0: R/W code |
| EEPROM_MEM_ADDR     | 0000h         | EEPROM memory address |
| LENGTH              | 32            | Transfer size per page |
| W_SIZE              | 256           | Transmit data size (W_SIZE \(\leq\) BUFSIZE) |
| R_SIZE              | 256           | Receive data size (R_SIZE \(\leq\) BUFSIZE) |
| BUFSIZE             | 256           | Transmit/receive data buffer size |
| PAGE                | (Note 1)      | Number of pages transmitted |
| IIC_BRG             | (33 - 1)      | 25 MHz ÷ (2 \(×\) (32 + 1)) \(\approx\) 378 kbps |
| WAIT_LOOP           | 50000         | Wait time to write to EEPROM |
| DISABLE             | 0             | Disable |
| ENABLE              | 1             | Enable |
| BUSY                | 0             | Midcommunication |
| RDY                 | 1             | Communication completed |
| PAR_ERR             | FFh           | Parameter error |
| IDLE                | FFh           | Idle |
| WRITE               | 0             | R/W (write mode) |
| READ                | 1             | R/W (read mode) |
| R_CODE              | 1             | EEPROM read code |
| W_CODE              | 0             | EEPROM write code |
| IIC_SP_ON           | 1             | Generate stop condition |
| IIC_SP_OFF          | 0             | Do not generate stop condition |
| PD_IIC              | pd7           | PD7_0: SDA2, PD7_1: SCL2 |
| PD_IIC_INIT         | 0000 0011b    | PD7_0: Output, PD7_1: Output |
| PD_IIC_INIT_INPUT   | 1111 1100b    | PD7_0: Input, PD7_1: Input |

Note:

1. \(((W\_SIZE + LENGTH - 1) \div LENGTH)\)
5.3 Structure/Union List

Figure 5.3 shows the Structure/Union Used in the Sample Code.

```
typedef union{
    struct{
        unsigned char b0:1;
        unsigned char b1:1;
        unsigned char b2:1;
        unsigned char b3:1;
        unsigned char b4:1;
        unsigned char b5:1;
        unsigned char b6:1;
        unsigned char b7:1;
    }bit;
    unsigned char all;
}byte_dt;

typedef union{
    struct{
        unsigned char b0:1;
        unsigned char b1:1;
        unsigned char b2:1;
        unsigned char b3:1;
        unsigned char b4:1;
        unsigned char b5:1;
        unsigned char b6:1;
        unsigned char b7:1;
        unsigned char b8:1;
        unsigned char b9:1;
        unsigned char b10:1;
        unsigned char b11:1;
        unsigned char b12:1;
        unsigned char b13:1;
        unsigned char b14:1;
        unsigned char b15:1;
    }bit;
    struct{
        unsigned char byte0;
        unsigned char byte1;
    }byte;
    unsigned short all;
}word_dt;

byte_dt iic_str1; /* Device address word */
#define device_addr_word iic_str1.all /* b7 to b4: Device code (fixed) b3 to b1: Device address code b0: R/W code */
#define iic_rw iic_str1.bit.b0 /* 0: Write (master transmit) 1: Read (master receive) */

byte_dt iic_str2; /* Status */
#define iic_status iic_str2.all /* All statuses */
#define iic_start iic_str2.bit.b0 /* 1: Midcommunication 0: Communication completed */
#define iic_err_par iic_str2.bit.b1 /* 1: Parameter error 0: No error detected */
#define iic_err_nack iic_str2.bit.b2 /* 1: NACK detection error 0: No error detected */
```

Figure 5.3 Structure/Union Used in the Sample Code
5.4 Variables

Table 5.2 lists the Global Variables.

### Table 5.2 Global Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsigned short</td>
<td>iic_length</td>
<td>Transfer data length</td>
<td>iic_master_start, master_trn_int, master_rcv_int</td>
</tr>
<tr>
<td>unsigned char</td>
<td>trn_data[BUFSIZE]</td>
<td>Transmit data buffer</td>
<td>main, stp_int</td>
</tr>
<tr>
<td>unsigned char</td>
<td>rcv_data[BUFSIZE]</td>
<td>Receive data buffer</td>
<td>main</td>
</tr>
<tr>
<td>unsigned short</td>
<td>mem_addr</td>
<td>Memory address buffer</td>
<td>main, stp_int, mem_addr_trn_int</td>
</tr>
<tr>
<td>unsigned char</td>
<td>mode</td>
<td>Save mode</td>
<td>main, _int0, _int1</td>
</tr>
<tr>
<td>unsigned char far</td>
<td>*ptr_buf</td>
<td>Buffer pointer</td>
<td>iic_master_start, master_trn_int, master_rcv_int</td>
</tr>
<tr>
<td>unsigned short</td>
<td>cnt_tr_index</td>
<td>Number of transmitted/received bytes</td>
<td>sta_int, stp_int, _uart2_trans, master_trn_int, master_rcv_int</td>
</tr>
<tr>
<td>unsigned short</td>
<td>cnt_remain_t_index</td>
<td>Remaining transmit data counter</td>
<td>iic_master_start, stp_int</td>
</tr>
<tr>
<td>unsigned char</td>
<td>cnt_mem_addr</td>
<td>Memory address transmit counter</td>
<td>_start_stop_condition_detection, sta_int, stp_int, _uart2_trans</td>
</tr>
<tr>
<td>unsigned char</td>
<td>cnt_page</td>
<td>Page counter</td>
<td>iic_master_start, stp_int</td>
</tr>
</tbody>
</table>

5.5 Functions

Table 5.3 lists the Functions.

### Table 5.3 Functions

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>Main processing</td>
</tr>
<tr>
<td>uart2_init</td>
<td>UART2 initial setting</td>
</tr>
<tr>
<td>iic_master_start</td>
<td>Master control start processing</td>
</tr>
<tr>
<td>_start_stop_condition_detection</td>
<td>Start condition/stop condition detection interrupt handling</td>
</tr>
<tr>
<td>sta_int</td>
<td>Start condition detection processing</td>
</tr>
<tr>
<td>stp_int</td>
<td>Stop condition detection processing</td>
</tr>
<tr>
<td>re_sta_int</td>
<td>Restart condition detection processing</td>
</tr>
<tr>
<td>_uart2_trans</td>
<td>UART2 transmit interrupt handling</td>
</tr>
<tr>
<td>mem_addr_trn_int</td>
<td>Memory address transmission</td>
</tr>
<tr>
<td>master_trn_int</td>
<td>Master transmission</td>
</tr>
<tr>
<td>master_rcv_int</td>
<td>Master reception</td>
</tr>
<tr>
<td>int_init</td>
<td>INT0 and INT1 initial setting</td>
</tr>
<tr>
<td>_int0</td>
<td>INT0 interrupt handling</td>
</tr>
<tr>
<td>_int1</td>
<td>INT1 interrupt handling</td>
</tr>
</tbody>
</table>
### 5.6 Function Specifications

The following tables list the sample code function specifications.

#### main

<table>
<thead>
<tr>
<th>Outline</th>
<th>Main processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void main(void)</td>
</tr>
<tr>
<td>Description</td>
<td>The initial setting of the system clock, UART2, INT0, and INT1 are performed. After the initial setting of INT0 and INT1, wait for the INT0 and INT1 interrupts to be generated. In write mode, the value of the variable (mode) on the RAM is the write data for the EEPROM; in read mode the value is the read data for the EEPROM.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Remark</td>
<td></td>
</tr>
</tbody>
</table>

#### uart2_init

<table>
<thead>
<tr>
<th>Outline</th>
<th>UART2 initial setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void uart2_init(unsigned char init)</td>
</tr>
<tr>
<td>Description</td>
<td>Set UART2 special mode 1 (I²C mode).</td>
</tr>
<tr>
<td>Argument</td>
<td>First argument: init: I²C mode enabled/disabled</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Remark</td>
<td></td>
</tr>
</tbody>
</table>
### iic_master_start

**Outline**
Master control start processing

**Header**
None

**Declaration**
```c
unsigned char iic_master_start(unsigned char addr, unsigned char rw, unsigned char *buf, unsigned short len)
```

**Description**
Processing to start transmission/reception. All statuses are initialized at the start of this function, and a parameter check of the argument is performed. If the parameters are outside the range of the argument, the parameter error flag becomes 1, and the returned value PAR_ERR is returned. When there is a parameter error, master control start processing is not performed. Confirm the following bus states:

- When the bus is busy, the returned value BUSY is returned, and master control start processing is not performed.
- When the bus is free, the returned value RDY is returned, and master control start processing is performed. The midcommunication flag becomes 1, and a start condition is generated.

**Arguments**
- First argument: `addr` : Device address word
- Second argument: `rw` : Write mode/read mode
- Third argument: `*buf` : Pointer for transmission buffer or reception buffer
- Fourth argument: `len` : Transfer data length

**Returned value**
- When bus is busy: BUSY
- When bus is free: RDY
- When there is a parameter error: PAR_ERR

**Remark**

### _start_stop_condition_detection

**Outline**
Start condition/stop condition detection interrupt handling

**Header**
None

**Declaration**
```c
void _start_stop_condition_detection(void)
```

**Description**
An interrupt is generated when a start condition/stop condition is detected. The sta_int function is called when a start condition generation is completed, the re_sta_int function is called when a restart condition generation is completed, and a stp_int function is called when a stop condition is detected.

**Argument**
None

**Returned value**
None

**Remark**

### sta_int

**Outline**
Start condition detection processing

**Header**
None

**Declaration**
```c
void sta_int(void)
```

**Description**
This function is called from the start condition/stop condition detection interrupt handling. After transmission/reception is enabled, the device address word is transmitted.

**Argument**
None

**Returned value**
None

**Remark**
### stp_int

**Outline**: Stop condition detection processing  
**Header**: None  
**Declaration**: void stp_int(void)  
**Description**: This function is called from the start condition/stop condition detection interrupt handling. If there is no data remaining to transmit/receive, UART2-related SFRs that were rewritten during communication return to their initial setting, and the midcommunication flag becomes 0. Enable the INT0 and INT1 interrupts.  
**Argument**: None  
**Returned value**: None  
**Remark**:  

### re_sta_int

**Outline**: Restart condition detection processing  
**Header**: None  
**Declaration**: void re_sta_int(void)  
**Description**: This function is called from the start condition/stop condition detection interrupt handling. When in read mode, the device address word is transmitted.  
**Argument**: None  
**Returned value**: None  
**Remark**:  

### _uart2_trans

**Outline**: UART2 transmit interrupt handling  
**Header**: None  
**Declaration**: void _uart2_trans(void)  
**Description**: An interrupt is generated at the falling edge of the 9th bit of the SCL clock. The U2RB register is read at the start of this function.  
- When a device address word is transmitted, a stop condition is generated when a NACK is detected.  
- Before the memory address is transmitted  
  Read the mem_addr_trn_int function.  
- After the memory address is transmitted  
  a) While in write mode, the master_trn_int function is called.  
  b) While in read mode before a restart condition is generated, a restart condition is generated.  
  c) While in read mode after a restart condition is generated, the master_rcv_int function is called.  
A stop condition is generated when communication is completed.  
**Argument**: None  
**Returned value**: None  
**Remark**:  

### mem_addr_trn_int

**Outline**
Memory address transmission

**Header**
None

**Declaration**
unsigned char mem_addr_trn_int(unsigned short rb_data, unsigned char cnt, unsigned char len)

**Description**
This function is called in the UART2 transmit handling. The memory address is transmitted.
- When the first byte of the memory address is transmitted, only the first byte of the memory address is transmitted.
- When the second byte of the memory address is transmitted, after the second byte of the memory address is transmitted, the memory address transmitted next is updated.

**Argument**
- First argument: rb_data : Data read from the U2RB data
- Second argument: cnt : Number of data counted
- Third argument: len : Transfer data length

**Returned value**
- When a stop condition is generated: IIC_SP_ON
- When a stop condition is not generated: IIC_SP_OFF

**Remark**

### master_trn_int

**Outline**
Master transmission

**Header**
None

**Declaration**
unsigned char master_trn_int(unsigned short rb_data)

**Description**
This function is read in the UART2 transmit interrupt handling. When an ACK is detected, and the data is not the last byte (the next transmission starts), the returned value is IIC_SP_OFF. When a NACK is detected (the NACK detection error flag becomes 1), and the last byte has been transmitted, the returned value is IIC_SP_ON.

**Argument**
- First argument: rb_data : Data read from the U2RB register

**Returned value**
- When a stop condition is generated: IIC_SP_ON
- When a stop condition is not generated: IIC_SP_OFF

**Remark**

### master_rcv_int

**Outline**
Master reception

**Header**
None

**Declaration**
unsigned char master_rcv_int(unsigned short rb_data)

**Description**
This function is read in the UART2 transmit interrupt handling. Store the argument value in the data buffer (except when the value is the device address word). If the next data is the last byte, set a NACK; if the next data is not the last byte, set an ACK. Then start the next reception.

When the data is not the last byte, the returned value IIC_SP_OFF is returned. When the last byte reception is completed, the returned value IIC_SP_ON is returned.

**Argument**
- First argument: rb_data : Data read from the U2RB register

**Returned value**
- When a stop condition is generated: IIC_SP_ON
- When a stop condition is not generated: IIC_SP_OFF

**Remark**
### int_init

**Outline**
INT0 and INT1 initial setting

**Header**
None

**Declaration**
void int_init(void)

**Description**
Set the INT0 and INT1 interrupt priority levels to 1.

**Argument**
None

**Returned value**
None

**Remark**

### _int0

**Outline**
INT0 interrupt handling

**Header**
None

**Declaration**
void _int0(void)

**Description**
Disable the INT0 and INT1 interrupts, and enable write mode.

**Argument**
None

**Returned value**
None

**Remark**

### _int1

**Outline**
INT1 interrupt handling

**Header**
None

**Declaration**
void _int1(void)

**Description**
Disable the INT0 and INT1 interrupts, and enable read mode.

**Argument**
None

**Returned value**
None

**Remark**
5.7 Flowcharts

5.7.1 Main Processing

Figure 5.4 and Figure 5.5 show the Main Processing.

![Flowchart of Main Processing](image)

- Disable maskable interrupts
  - I flag ← 0
  - PLL clock setting
    - setPLLClock()
  - UART2 initial setting
    - uart2_init()
  - INT0 and INT1 initial setting
    - int_init()
  - Set each clock frequency in PLL mode.
    - I^2C mode enabled
  - • Set transmit data to the transmission data buffer
    - • Initialize reception data buffer
  - Store EEPROM memory address
  - Set mode to an idle state
  - Set RAM data for transmission/reception
  - Enable maskable interrupts
    - I flag ← 1

Figure 5.4 Main Processing (1/2)
Figure 5.5  Main Processing (2/2)

```
A

Mode

Write mode
- Master control start processing
  - iic_master_start()
- Set mode to idle state

Read mode
- Master control start processing
  - iic_master_start()
- Set mode to idle state

default

Midcommunication?
  No
  Error occurred?
    No
    Parameter error occurred?
      No
      NACK detection error occurred?
        Yes
        Set mode to idle state
    Yes
      Parameter error occurred?
        Yes
        NACK detection error occurred?
          Yes
          Set mode to idle state
        No
        Error occurred?
          Yes
          Parameter error occurred?
            Yes
            NACK detection error occurred?
              Yes
              Set mode to idle state
            No
            Error occurred?
              Yes
              Parameter error occurred?
                Yes
                NACK detection error occurred?
                  Yes
                  Set mode to idle state
                No
                Error occurred?
                  Yes
                  Parameter error occurred?
                    Yes
                    NACK detection error occurred?
                      Yes
                      Set mode to idle state
                    No
                    Error occurred?
                      Yes
                      Parameter error occurred?
                        Yes
                        NACK detection error occurred?
                          Yes
                          Set mode to idle state
                        No
                        Error occurred?
                          Yes
                          Parameter error occurred?
                            Yes
                            NACK detection error occurred?
                              Yes
                              Set mode to idle state
                            No
                            Error occurred?
                              Yes
                              Parameter error occurred?
                                Yes
                                NACK detection error occurred?
                                  Yes
                                  Set mode to idle state
                                No
                                Error occurred?
                                  Yes
                                  Parameter error occurred?
                                    Yes
                                    NACK detection error occurred?
                                      Yes
                                      Set mode to idle state
                                    No
                                    Error occurred?
                                      Yes
                                      Parameter error occurred?
                                        Yes
                                        NACK detection error occurred?
                                          Yes
                                          Set mode to idle state
                                        No
                                        Error occurred?
                                          Yes
                                          Parameter error occurred?
                                            Yes
                                            NACK detection error occurred?
                                              Yes
                                              Set mode to idle state
                                            No
                                            Error occurred?
                                              Yes
                                              Parameter error occurred?
                                                Yes
                                                NACK detection error occurred?
                                                  Yes
                                                  Set mode to idle state
                                                No
                                                Error occurred?
                                                  Yes
                                                  Parameter error occurred?
                                                    Yes
                                                    NACK detection error occurred?
                                                      Yes
                                                      Set mode to idle state
                                                    No
                                                    Error occurred?
                                                      Yes
                                                      Parameter error occurred?
                                                        Yes
                                                        NACK detection error occurred?
                                                          Yes
                                                          Set mode to idle state
                                                        No
                                                        Error occurred?
                                                          Yes
                                                          Parameter error occurred?
                                                            Yes
                                                            NACK detection error occurred?
                                                              Yes
                                                              Set mode to idle state
                                                            No
                                                            Error occurred?
                                                              Yes
                                                              Parameter error occurred?
                                                                Yes
                                                                NACK detection error occurred?
                                                                  Yes
                                                                  Set mode to idle state
                                                                No
                                                                Error occurred?
                                                                  Yes
                                                                  Parameter error occurred?
                                                                    Yes
                                                                    NACK detection error occurred?
                                                                      Yes
                                                                      Set mode to idle state
                                                                    No
                                                                    Error occurred?
                                                                      Yes
                                                                      Parameter error occurred?
                                                                        Yes
                                                                        NACK detection error occurred?
                                                                          Yes
                                                                          Set mode to idle state
                                                                        No
                                                                        Error occurred?
                                                                          Yes
                                                                          Parameter error occurred?
                                                                            Yes
                                                                            NACK detection error occurred?
                                                                              Yes
                                                                              Set mode to idle state
                                                                            No
                                                                            Error occurred?
                                                                              Yes
                                                                              Parameter error occurred?
                                                                                Yes
                                                                                NACK detection error occurred?
                                                                                  Yes
                                                                                  Set mode to idle state
                                                                                No
                                                                                Error occurred?
                                                                                  Yes
                                                                                  Parameter error occurred?
                                                                                    Yes
                                                                                    NACK detection error occurred?
                                                                                      Yes
                                                                                      Set mode to idle state
                                                                                  No
                                                                                  Error occurred?
                                                                                    Yes
                                                                                    Parameter error occurred?
                                                                                      Yes
                                                                                      NACK detection error occurred?
                                                                                        Yes
                                                                                        Set mode to idle state
                                                                                      No
                                                                                      Error occurred?
                                                                                        Yes
                                                                                        Parameter error occurred?
                                                                                          Yes
                                                                                          NACK detection error occurred?
                                                                                            Yes
                                                                                            Set mode to idle state
                                                                                          No
                                                                                          Error occurred?
                                                                                              Yes
                                                                                              Parameter error occurred?
                                                                                                    Yes
                                                                                                    NACK detection error occurred?
                                                                                                      Yes
                                                                                                      Set mode to idle state
                                                                                                    No
                                                                                                    Error occurred?
                                                                                                        Yes
                                                                                                        Parameter error occurred?
                                                                                                          Yes
                                                                                                          NACK detection error occurred?
                                                                                                            Yes
                                                                                                            Set mode to idle state
                                                                                                          No
                                                                                                          Error occurred?
                                                                                                              Yes
                                                                                                              Parameter error occurred?
                                                                                                                Yes
                                                                                                                NACK detection error occurred?
                                                                                                                  Yes
                                                                                                                  Set mode to idle state
                                                                                                                No
                                                                                                                Error occurred?
                                                                                                                    Yes
                                                                                                                    Parameter error occurred?
                                                                                                                      Yes
                                                                                                                      NACK detection error occurred?
                                                                                                                        Yes
                                                                                                                        Set mode to idle state
                                                                                                                      No
                                                                                                                      Error occurred?
                                                                                                                        Yes
                                                                                                                        Parameter error occurred?
                                                                                                                          Yes
                                                                                                                          NACK detection error occurred?
                                                                                                                            Yes
                                                                                                                            Set mode to idle state
                                                                                                                          No
                                                                                                                          Error occurred?
                                                                                                                              Yes
                                                                                                                              Parameter error occurred?
                                                                                                                                Yes
                                                                                                                                NACK detection error occurred?
                                                                                                                                  Yes
                                                                                                                                  Set mode to idle state
                                                                                                                                No
                                                                                                                                Error occurred?
                                                                                                                                    Yes
                                                                                                                                    Parameter error occurred?
                                                                                                                                      Yes
                                                                                                                                      NACK detection error occurred?
                                                                                                                                        Yes
                                                                                                                                        Set mode to idle state
                                                                                                                                      No
                                                                                                                                      Error occurred?
                                                                                                                                          Yes
                                                                                                                                          Parameter error occurred?
                                                                                                                                            Yes
                                                                                                                                            NACK detection error occurred?
                                                                                                                                                Yes
                                                                                                                                                Set mode to idle state
                                                                                                                                            No
                                                                                                                                            Error occurred?
                                                                                                                                                Yes
                                                                                                                                                Parameter error occurred?
                                                                                                                                                    Yes
                                                                                                                                                    NACK detection error occurred?
                                                                                                                                                      Yes
                                                                                                                                                      Set mode to idle state
                                                                                                                                                    No
                                                                                                                                                    Error occurred?
                                                                                                                                                        Yes
                                                                                                                                                        Parameter error occurred?
                                                                                                                                                              Yes
                                                                                                                                                              NACK detection error occurred?
                                                                                                                                                                 Yes
                                                                                                                                                                 Set mode to idle state
                                                                                                                                                              No
                                                                                                                                                              Error occurred?
                                                                                                                                                                Yes
                                                                                                                                                                Parameter error occurred?
                                                                                                                                                                      Yes
                                                                                                                                                                      NACK detection error occurred?
                                                                                                                                                                           Yes
                                                                                                                                                                           Set mode to idle state
                                                                                                                                                                      No
                                                                                                                                                                      Error occurred?
                                                                                                                                                                           Yes
                                                                                                                                                                           Parameter error occurred?
                                                                                                                                                                               Yes
                                                                                                                                                                               NACK detection error occurred?
                                                                                                                                                                                Yes
                                                                                                                                                                                Set mode to idle state
                                                                                                                                                                               No
                                                                                                                                                                               Error occurred?
                                                                                                                                                                                   Yes
                                                                                                                                                                                   Parameter error occurred?
                                                                                                                                                                                       Yes
                                                                                                                                                                                       NACK detection error occurred?
                                                                                                                                                                                             Yes
                                                                                                                             Figure 5.5  Main Processing (2/2)
```
5.7.2 UART2 Initial Setting

Figure 5.6 shows the UART2 Initial Setting.

```
uart2_init

I2C mode enabled? Yes

Select I2C mode
U2SMR register ← 01h
IICM bit = 1

Disable start and stop condition detection interrupts
BCN2IC register ← 00h
Bits ILVL2 to ILVL0 = 000b

Disable UART2 transmit interrupt
S2TIC register ← 00h
Bits ILVL2 to ILVL0 = 000b

Disable transmission/reception
U2C1 register ← 00h
TE bit = 0
RE bit = 0

Set UART2 transmit/receive mode register
U2MR register ← 02h
Bits SMD2 to SMD0 = 010b
CKDIR bit = 0: Internal clock

Set UART2 special mode register 2
U2SMR2 register ← 03h
IICM2 bit = 1: Use transmit/receive interrupt
CSC bit = 1: Clock synchronization enabled

Set UART2 special mode register 3
U2SMR3 register ← E0h
Bits DL2 to DL0 = 111b: 7 to 8 cycles

Set UART2 special mode register 4
U2SMR4 register ← 70h
ACKD bit = 1: NACK
SCLH1 bit = 1: Stop SCL2 output

Set UART2 transmit/receive control register 0
U2C0 register ← 90h
Bits CLK1 and CLK0 = 00b: f1
UFORM bit = 1: MSB first

Set bit rate (378 kbps)
U2RG ← IIC_BRG

Select transmission is completed for UART2 transmit interrupt source (TXEPT = 1 )
U2C1 register ← 10h
U2IRS bit = 1

Clear the IR bit
BCN2IC register ← 00h
S2TIC register ← 00h

Set SDA2 pin as N-channel open drain output
P7_0S register ← 43h
Bits PSEL2 to PSEL0 = 011b
NOD bit = 1

Set SCL2 pin as N-channel open drain output
P7_1S register ← 43h
Bits PSEL2 to PSEL0 = 011b
NOD bit = 1

Set P7_0 (SDA2) and P7_1 (SCL2) to output mode

return
```

Figure 5.6 UART2 Initial Setting
5.7.3 Master Control Start Processing

Figure 5.7 shows Master Control Start Processing.

Arguments
- unsigned char addr: Specified device address word
- unsigned char rw: Write mode, read mode
- unsigned char far *buf: Pointer to transmit/receive data buffer
- unsigned char len: Transfer data length

Clear all statuses

Parameter error occurred?
- Yes
- No

Bus is free?
- Yes
- No

Set parameter error flag

return(PAR_ERR) Communication does not start.

return(BUSY) Bus is busy

Set device address

Set device address word
Set R/W

Write mode enabled, and remaining data length is less than 1 page?
- Yes
- No

Set remaining transmit data length

Set transfer data length

Set buffer address

Set midcommunication flag

Enable start condition/stop condition detection interrupt

BCN2IC register ← 01h
Bits ILVL2 to ILVL0 = 001b: Level 1

Set UART2 special mode register

U2SMR4 register ← 70h
STSPSEL bit = 0: Select serial I/O circuit

Set UART2 transmit/receive mode register

U2MR register ← 02h
Bits SMD2 to SMD0 = 010b: I2C mode
CKDIR bit = 0: Internal clock

The shortest wait time is derived by setting the U2BRG register value to its fastest.

Set U2BRG bit rate register

U2SMR2 register ← 03h
IICM2 bit = 1: Use transmit/receive interrupt
CSC bit = 1: Clock synchronization enabled

Set U2SMR2 special mode register 2

Return U2BRG setting to desired transfer rate

Generate a start condition

U2SMR4 register ← 71h
U2SMR4 register ← 09h
STAREQ bit = 1: Start
STSPSEL bit = 1: Select start condition/stop condition generate circuit

Communication started

Figure 5.7 Master Control Start Processing
### 5.7.4 Start Condition/Stop Condition Detection Interrupt Handling

Figure 5.8 shows Start Condition/Stop Condition Detection Interrupt Handling.

![Diagram of Start Condition/Stop Condition Detection Interrupt Handling]

**Figure 5.8** Start Condition/Stop Condition Detection Interrupt Handling
5.7.5 Start Condition Detection Processing

Figure 5.9 shows Start Condition Detection Processing.

Figure 5.9 Start Condition Detection Processing

sta_int

Set clock phase to clock delayed
U2SMR3 register ← E2h
CKPH bit = 1: Clock delayed

Enable transmit/receive
U2C1 register ← 15h
TE bit = 1: Transmission enabled
RE bit = 1: Reception enabled

Do not generate start or stop conditions
U2SMR4 register ← 00h

Clear ABT bit
U2RB register ← 0000h
ABT bit = 0: Not detected (win)

Set device address word from 1st to 8th bits

Set data to the 9th bit to leave SDA2 pin open

Start transmitting device address word

Clear the start condition/stop condition interrupt request flag (because the CKPH bit was changed)
BCN2IC register ← 01h
IR bit = 0: No interrupt requested

Enable UART2 transmission interrupt
S2TIC register
Bits ILVL2 to ILVL0 = 001b: Level 1

Initialize number of transmit/receive bytes

Initialize memory address transmit counter

return
5.7.6 Stop Condition Detection Processing

Figure 5.10 and Figure 5.11 show Stop Condition Detection Processing.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Disable transmit/receive U2C1 register ← 10h</td>
</tr>
<tr>
<td></td>
<td>TE bit = 0: Transmission disabled</td>
</tr>
<tr>
<td></td>
<td>RE bit = 0: Reception disabled</td>
</tr>
<tr>
<td>2.</td>
<td>Disable serial interface U2MR register ← 00h</td>
</tr>
<tr>
<td></td>
<td>Bits SMD2 to SMD0 = 000b: Serial interface disabled</td>
</tr>
<tr>
<td>3.</td>
<td>Set clock phase to no clock delay</td>
</tr>
<tr>
<td></td>
<td>Disable serial interface U2MR register ← 00h</td>
</tr>
<tr>
<td></td>
<td>Bits SMD2 to SMD0 = 000b: Serial interface disabled</td>
</tr>
<tr>
<td>4.</td>
<td>Set UART2 special mode register 4 U2SMR4 register ← 70h</td>
</tr>
<tr>
<td></td>
<td>ACKD bit = 1: NACK (leave SDA2 pin open)</td>
</tr>
<tr>
<td></td>
<td>SCLHI bit = 1: Stop SCL2 output</td>
</tr>
<tr>
<td>5.</td>
<td>Set UART2 transmit/receive mode register U2MR register ← 02h</td>
</tr>
<tr>
<td></td>
<td>Bits SMD2 to SMD0 = 010b: I2C mode</td>
</tr>
<tr>
<td></td>
<td>CKDIR bit = 0: Internal clock</td>
</tr>
<tr>
<td>6.</td>
<td>Disable UART2 transmit interrupt S2TIC register ← 00h</td>
</tr>
<tr>
<td></td>
<td>Bits ILVL2 to ILVL0 = 000b: Level 0 (interrupt disabled)</td>
</tr>
<tr>
<td>7.</td>
<td>Disable start and stop condition detection interrupts BCN2IC register ← 00h</td>
</tr>
<tr>
<td></td>
<td>Bits ILVL2 to ILVL0 = 000b: Level 0 (interrupt disabled)</td>
</tr>
<tr>
<td>8.</td>
<td>Write mode enabled? Yes</td>
</tr>
<tr>
<td></td>
<td>Wait time to write to EEPROM (at least 5 ms)</td>
</tr>
<tr>
<td></td>
<td>Calculate remaining transmit data</td>
</tr>
<tr>
<td></td>
<td>Update page counter</td>
</tr>
<tr>
<td></td>
<td>Initialize number of transmit/receive bytes</td>
</tr>
<tr>
<td></td>
<td>Initialize number of memory address transmissions</td>
</tr>
</tbody>
</table>

Figure 5.10 Stop Condition Detection Processing (1/2)
Figure 5.11 Stop Condition Detection Processing (2/2)
5.7.7 Restart Condition Detection Processing

Figure 5.12 shows Restart Condition Detection Processing.

```
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>re_sta_int</td>
<td></td>
</tr>
<tr>
<td>Set clock phase to clock delayed</td>
<td>U2SMR3 register ← E2h</td>
</tr>
<tr>
<td>Enable transmission/reception</td>
<td>CKPH bit = 1: Clock delayed</td>
</tr>
<tr>
<td></td>
<td>U2C1 register ← 15h</td>
</tr>
<tr>
<td></td>
<td>TE bit = 1: Transmission enabled</td>
</tr>
<tr>
<td></td>
<td>RE bit = 1: Reception enabled</td>
</tr>
<tr>
<td></td>
<td>U2SMR4 register ← 00h</td>
</tr>
<tr>
<td>Do not generate a start condition/stop condition</td>
<td></td>
</tr>
<tr>
<td>Clear the ABT bit</td>
<td>U2RB register ← 0000h</td>
</tr>
<tr>
<td></td>
<td>ABT bit = 0: Not detected (win)</td>
</tr>
<tr>
<td>Set the device address word to the 1st to 8th bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Start transmitting device address word</td>
</tr>
<tr>
<td></td>
<td>Clear the start condition/stop condition</td>
</tr>
<tr>
<td></td>
<td>interrupt request flag</td>
</tr>
<tr>
<td></td>
<td>(because the CKPH bit was changed)</td>
</tr>
<tr>
<td></td>
<td>BCN2IC register ← 01h</td>
</tr>
<tr>
<td></td>
<td>IR bit = 0: No interrupt requested</td>
</tr>
<tr>
<td></td>
<td>Enable UART2 transmission interrupt</td>
</tr>
<tr>
<td></td>
<td>S2TIC register</td>
</tr>
<tr>
<td></td>
<td>Bits ILVL2 to ILVL0 = 001b: Level 1</td>
</tr>
<tr>
<td></td>
<td>return</td>
</tr>
</tbody>
</table>
```

Figure 5.12 Restart Condition Detection Processing
### 5.7.8 UART2 Transmit Interrupt Handling

Figure 5.13 shows UART2 Transmit Interrupt Handling.

- **_uart2_trans**
  - Generate stop condition
  - Initialize variable for determination (stop_req)
  - Read receive buffer register
  - NACK detected by device address word?
    - Yes
      - Set IIC_SP_ON to stop_req
      - Set NACK detection error flag
    - No
  - Memory address transmitted in 2 bytes?
    - Yes
      - Immediately after the 2nd byte of the memory address is transmitted, a transmission interrupt is generated, and communication is in read mode?
        - Yes
          - U2SMR4 register ← 02h
          - U2SMR4 register ← 3Ah
          - RSTAREQ bit = 1: Start
          - STSPSEL bit = 1: Select start condition/stop condition
          - ACKC bit = 1: ACK data output
          - Update memory address counter
          - Generate restart condition
          - U2SMR4 register ← 04h
          - U2SMR4 register ← 3Ch
          - STPREQ bit = 1: Start
          - STSPSEL bit = 1: Select start condition/stop condition generate circuit
          - ACKC bit = 1: ACK data output
          - Update memory address counter
          - Read mode?
            - No
              - Master reception processing
                - master_rcv_int()
            - Yes
              - Master transmission processing
                - master_tm_int()
        - No
          - Generate stop condition
          - return
  - No
    - Is IIC_SP_ON set to stop_req?
      - Yes
        - Generate stop condition
        - return
      - No
        - Initialize variable for determination (stop_req)
        - Generate stop condition
        - return

---

**Figure 5.13** UART2 Transmit Interrupt Handling
5.7.9 Memory Address Transmission

Figure 5.14 shows Memory Address Transmission.

```
mem_addr_trn_int

Set data read from the receive buffer register

ACK detected?
  Yes
  No

1st memory address transmitted?
  Yes
  No

Set 1st memory address

Set 2nd memory address

Update memory address

Set data to the 9th bit to leave SDA2 pin open

Set write data to U2TB register (transmission start)

return(IIC_SP_OFF)

Arguments
unsigned short rb_data: Read data from U2RB register
unsigned char cnt: Number of memory address transmissions
unsigned char len: Transfer data length

Do not generate a stop condition

return(IIC_SP_ON)

Generate a stop condition
```

Figure 5.14 Memory Address Transmission
5.7.10 Master Transmission

Figure 5.15 shows Master Transmission.

```
master_trn_int

Set data read from the receive buffer register

ACK detected?
  Yes
  No

Other than last byte?
  Yes
  No

Set next byte of write data

Set data to the 9th bit to leave SDA2 pin open

Set write data to U2TB register (transmission start)

Pointer for transmit buffer + 1

Number of transmitted/received bytes + 1

return(IIC_SP_OFF)  Do not generate a stop condition

return(IIC_SP_ON)  Generate a stop condition

Argument
unsigned short rb_data: Data read from the U2RB register
```

Figure 5.15 Master Transmission
5.7.11 Master Reception

Figure 5.16 shows Master Reception.

```
master_rcv_int
```

Argument
unsigned short rb_data: Data read from the U2RB register

- Received data from 2nd byte on?
  - No
  - Store read data in buffer
  - Pointer to receive buffer + 1
- Last byte?
  - No
  - Number of transmit/receive bytes + 1
  - Next data received is last byte?
    - No
      - Set NACK data of next byte (prepare for next reception)
      - Set ACK data of next byte (prepare for next reception)
      - return(IIC_SP_OFF)
    - Yes
      - U2TB register ← 0FFh
      - U2TB register ← 0FFh
      - return(IIC_SP_OFF)
  - Yes
    - Generate a stop condition
    - return(IIC_SP_OFF)
```

Do not generate a stop condition

Figure 5.16 Master Reception
5.7.12 INT0 and INT1 Initial Setting

Figure 5.17 shows the INT0 and INT1 Initial Setting.

![Diagram of INT0 and INT1 Initial Setting]

5.7.13 INT0 Interrupt Handling

Figure 5.18 shows INT0 Interrupt Handling.

![Diagram of INT0 Interrupt Handling]

5.7.14 INT1 Interrupt Handling

Figure 5.19 shows INT1 Interrupt Handling.

![Diagram of INT1 Interrupt Handling]
6. **Sample Code**
   Sample code can be downloaded from the Renesas Electronics website.

7. **Reference Documents**
   R32C/116 Group User’s Manual: Hardware Rev.1.10
   R32C/117 Group User’s Manual: Hardware Rev.1.10
   R32C/118 Group User’s Manual: Hardware Rev.1.10

   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/inquiry
Revision History

R32C/100 Series
EEPROM Control Using UARTI Special Mode 1 (I²C Mode)

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Oct. 28, 2011</td>
<td>— First edition issued</td>
</tr>
</tbody>
</table>

All trademarks and registered trademarks are the property of their respective owners.
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. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.

2. 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.

3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.

4. 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.

5. When exporting the 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. You should not use Renesas Electronics products or the 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. Renesas Electronics products and technology may not be used or incorporated into any products or systems whose manufacturers, use, or sale is prohibited under any applicable domestic or foreign laws or regulation.

6. 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.

7. Renesas Electronics products are classified according to the following three quality grades: “Standard,” “High Quality,” and “Specific.” The recommended applications for each Renesas Electronics product depend on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.

8. You should use the Renesas Electronics products described in the 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.

9. 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 against them. 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, the control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of reconfigurable computer alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

10. 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.

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. You are not entitled to use Renesas Electronics products listed herein for any non-technical purposes such as that required for the development and enhancement of Renesas Electronics products or that required for the manufacture of Renesas Electronics products or that required for the manufacture of other Renesas Electronics products.

(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 and manufactured by or for Renesas Electronics.