1. **Abstract**

   This document describes a program for clock synchronous serial data communication using clock synchronous serial I/O with chip select (SSU).

2. **Introduction**

   The application example described in this document applies to the following MCU:

   - **MCU**: R8C/2D Group

   This program can be used with other R8C/Tiny Series MCUs which have the same special function registers (SFRs) as the R8C/2D Group. Careful evaluation is recommended before using this application note.

3. **Application Example**

   SSU has three modes: clock synchronous communication mode, 4-wire bus communication mode, and bidirectional communication mode. It can use MCUs as a master device and a slave device. This document provides an explanation about each bus communication mode between two MCUs (R8C/2D Group).

   In this application example, the master device starts master transmission/reception after an INT0 interrupt request is accepted. The slave device waits for data in a receive state until it receives data from the master device.

   The following four pins are used during communication:

   - SSCK: Clock I/O pin
   - SSI: Data I/O pin
   - SSO: Data I/O pin
   - SCS: Chip-select I/O pin

   When transmitting, the master device outputs a clock from the SSCK pin, data from the SSO pin, and a low signal from the SCS pin. When receiving, it outputs a clock from the SSCK pin, a low signal from the SCS pin, and data from the SSI pin.

   Specifications of transmission/reception are as follows:

   - Transfer clocks: Internal clock
   - Data transfer direction: MSB first
   - SSCK clock phase: Change data at odd edge

   Figure 3.1 to Figure 3.3 show the connection diagram in each communication mode.
Figure 3.1 Connections in 4-Wire Bus Communication Mode

Figure 3.2 Connections in 4-Wire Bus Communication (Bidirectional) Mode

Figure 3.3 Connections in Clock Synchronous Communication Mode
3.1 Pins

Table 3.1 Pin and Function

<table>
<thead>
<tr>
<th>Pin</th>
<th>I/O pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3_5/SSCK</td>
<td>I/O pin</td>
<td>Clock I/O pin</td>
</tr>
<tr>
<td>P3_7/SSO</td>
<td>I/O pin</td>
<td>Data I/O pin</td>
</tr>
<tr>
<td>P3_3/SSI</td>
<td>I/O pin</td>
<td>Data I/O pin</td>
</tr>
<tr>
<td>P3_4/SCS</td>
<td>I/O pin</td>
<td>Chip select I/O pin</td>
</tr>
</tbody>
</table>

3.2 Memory

Table 3.2 Memory

<table>
<thead>
<tr>
<th>Memory</th>
<th>Size</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM (Program only)</td>
<td>309 bytes</td>
<td>266 bytes Only in the rej05b1153_src_master.c module or rej05b1153_src_slave.c module</td>
</tr>
<tr>
<td>RAM</td>
<td>6 bytes</td>
<td>6 bytes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum user stack</th>
<th>Master</th>
<th>Slave</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13 bytes</td>
<td>13 bytes</td>
<td>main function: 7 bytes (master) 7 bytes (slave) sfr_init function: 3 bytes (master) 6 bytes (slave) cs_communication function: 6 bytes</td>
</tr>
<tr>
<td>Maximum interrupt stack</td>
<td>0 bytes</td>
<td>Not used</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3 RAM and Definition (Master Transmit/Receive Mode)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Type</th>
<th>Size</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>_data</td>
<td>unsigned char</td>
<td>3 bytes</td>
<td>Transmit data</td>
</tr>
<tr>
<td>_data_store</td>
<td>unsigned char</td>
<td>3 bytes</td>
<td>Receive data</td>
</tr>
</tbody>
</table>

Table 3.4 RAM and Definition (Slave Transmit/Receive Mode)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Type</th>
<th>Size</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>_data</td>
<td>unsigned char</td>
<td>3 bytes</td>
<td>Transmit data</td>
</tr>
<tr>
<td>_data_store</td>
<td>unsigned char</td>
<td>3 bytes</td>
<td>Receive data</td>
</tr>
</tbody>
</table>
4. Setup

This section shows the initial setting procedures and values to set the example described in 3. Application Example. Refer to the R8C/2D Group Hardware Manual for details on individual registers.

4.1 Setting the System Clock

(1) Enable writing to registers CM0, CM1, OCD, FRA0, FRA1, and FRA2.

```plaintext
Protect Register

PRC0 Protect bit 0
Writing to registers CM0, CM1, OCD, FRA0, FRA1, and FRA2 is enabled.

PRCR Address 000Ah
```

(2) Start the low-speed on-chip oscillator.

```plaintext
System Clock Control Register 1

CM1 Address 0007h
CM14 Low-speed on-chip oscillation stop bit
Low-speed on-chip oscillator on
```

(3) Set the division ratio of the high-speed on-chip oscillator.

```plaintext
High-Speed On-Chip Oscillator Control Register 2

FRA2 Address 0025h
FRA22 - FRA20 High-speed on-chip oscillator frequency switching bits
Divide-by-2 mode
(b7-b3) Reserved bits
Set to 0.
```

(4) Start the high-speed on-chip oscillator.

```plaintext
High-Speed On-Chip Oscillator Control Register 0

FRA0 Address 0023h
FRA00 High-speed on-chip oscillator enable bit
High-speed on-chip oscillator on
```

1
(5) Wait until oscillation stabilizes.

(6) Select the high-speed on-chip oscillator.

![High-Speed On-Chip Oscillator Control Register 0](image)

(7) Set the system clock division select bits 1.

![System Clock Control Register 1](image)

(8) Set the system clock division select bit 0.

![System Clock Control Register 0](image)

(9) Disable writing to registers CM0, CM1, OCD, FRA0, FRA1, and FRA2.

![Protect Register](image)
### 4.2 Setting INT0 Interrupt Request

1. Set the external input enable register.

#### External Input Enable Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
<th>INTEN</th>
<th>INT0EN</th>
<th>INT0PL</th>
<th>INT1EN</th>
<th>INT1PL</th>
<th>INT2EN</th>
<th>INT2PL</th>
<th>INT3EN</th>
<th>INT3PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **INTEN** Address 00F9h
- **INT0EN** INT0 input enable bit
- **INT0PL** INT0 input polarity select bit
- **INT1EN** INT1 input enable bit
- **INT1PL** INT1 input polarity select bit
- **INT2EN** INT2 input enable bit
- **INT2PL** INT2 input polarity select bit
- **INT3EN** INT3 input enable bit
- **INT3PL** INT3 input polarity select bit

2. Set the INT input filter select register.

#### INT Input Filter Select Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
<th>INTF</th>
<th>INTO1F1 - INTO0F0</th>
<th>INTO1F0 - INTO0F0</th>
<th>INTO2F1 - INTO2F0</th>
<th>INTO2F0 - INTO2F0</th>
<th>INTO3F1 - INTO3F0</th>
<th>INTO3F0 - INTO3F0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **INTF** Address 00FAh
- **INT0F1 - INTO0F0** INT0 input filter select bits
- **INT1F1 - INTO1F0** INT1 input filter select bits
- **INT2F1 - INTO2F0** INT2 input filter select bits
- **INT3F1 - INTO3F0** INT3 input filter select bits

**INTF0**: No filter
**INTF1 - INTF0**: Unused bit. Set to 0.
(3) Set the INT0 interrupt control register.

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b0</th>
<th>INT0IC</th>
<th>Address 005Dh</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ILVL2 - ILVL0</td>
<td>Interrupt priority level select bits 2 - 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Level 0 (interrupt disable)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>IR</td>
<td></td>
<td>IR Interrupt request bit</td>
<td>Requests no interrupt</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>POL</td>
<td></td>
<td>POL Polarity switch bit</td>
<td>Selects falling edge</td>
</tr>
<tr>
<td>0</td>
<td>(b5)</td>
<td>(b5)</td>
<td></td>
<td>Reserved bit</td>
<td>Set to 0.</td>
</tr>
<tr>
<td>0</td>
<td>(b7-b6)</td>
<td>(b7-b6)</td>
<td></td>
<td>Nothing is assigned.</td>
<td>Set to 0.</td>
</tr>
</tbody>
</table>
4.3 Setting Master Transmit/Receive Mode

4.3.1 Initial Setting

Enable transferring and set the transfer clock and transfer format.

(1) Set the port P3_7, P3_5, P3_4, and P3_3 direction bits as input ports.

<table>
<thead>
<tr>
<th>Port P3 Direction Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7</td>
</tr>
<tr>
<td>PD3</td>
</tr>
<tr>
<td>PD3_3 Port P3_3 direction bit</td>
</tr>
<tr>
<td>PD3_3 Input mode</td>
</tr>
<tr>
<td>PD3_4 Port P3_4 direction bit</td>
</tr>
<tr>
<td>PD3_4 Input mode</td>
</tr>
<tr>
<td>PD3_5 Port P3_5 direction bit</td>
</tr>
<tr>
<td>PD3_5 Input mode</td>
</tr>
<tr>
<td>PD3_7 Port P3_7 direction bit</td>
</tr>
<tr>
<td>PD3_7 Input mode</td>
</tr>
</tbody>
</table>

(2) Set the SSU bus operation enable bit.

<table>
<thead>
<tr>
<th>Module Operation Enable Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7</td>
</tr>
<tr>
<td>MSTCR Address 0008h</td>
</tr>
<tr>
<td>MSTIIC SSU, (^2)C bus operation enable bit</td>
</tr>
<tr>
<td>Enable</td>
</tr>
</tbody>
</table>

(3) Set the SSU/\(^2\)C bus switch bit.

<table>
<thead>
<tr>
<th>Port Mode Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7</td>
</tr>
<tr>
<td>PMR Address 00F8h</td>
</tr>
<tr>
<td>IICSEL SSU, (^2)C bus switch bit</td>
</tr>
<tr>
<td>Selects SSU function</td>
</tr>
</tbody>
</table>
(4) Disable reception and transmission.

SS Enable Register

- **CEIE**: Conflict error interrupt enable bit
  - Disables conflict error interrupt request
- **(b2-b1)**: Nothing is assigned.
  - Set to 0.
- **RE**: Receive enable bit
  - Disables receive
- **TE**: Transmit enable bit
  - Disables transmit
- **RIE**: Receive interrupt enable bit
  - Disables receive data full and overrun error interrupt request
- **TEIE**: Transmit end interrupt enable bit
  - Disables transmit end interrupt request
- **TIE**: Transmit interrupt enable bit
  - Disables transmit data empty interrupt request

(5) Select the communication mode.

SS Mode Register 2

- **SSMR2**: Address 00BDh
  - Clock synchronous serial I/O with chip select mode select bit
  - Set to 1 to use 4-wire bus communication mode or bidirectional communication mode.
  - Set to 0 to use clock synchronous communication mode.
(6) Select MSB first.

SS Mode Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SSMR Address 00BAh</td>
</tr>
<tr>
<td>BC2 - BC0</td>
<td>Bits counter 2 to 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved bit Set to 0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nothing is assigned. Set to 1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPHS SSCK clock phase select bit Change data at odd edge (download data at even edge)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPOS SSCK clock polarity select bit High when clock stops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLS MSB first/LSB first select bit Transfers data MSB first</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(7) Select master device.

SS Control Register H

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MSS Master/slave device select bit Operates as master device</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSCRH Address 00B8h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(8) Set the bidirectional mode enable bit, SSCK pin select bit, and SCS pin select bit.

![SS Mode Register 2 Diagram]

- **SSMR2** (Address 00BDh) - Clock synchronous serial I/O with chip select mode select bit
  - Set to 1 to use 4-wire bus communication mode or bidirectional communication mode.
  - Set to 0 to use clock synchronous communication mode.

- **SSUMS** - Clock synchronous serial I/O with chip select mode select bit
  - Set to 1 to use 4-wire bus communication mode or bidirectional communication mode.
  - Set to 0 to use clock synchronous communication mode.

- **CSOS** - SCS pin open drain output select bit
  - Unused bit. Set to 0.

- **SOOS** - Serial data pin open output drain select bit
  - Unused bit. Set to 0.

- **SCKOS** - SSCK pin open drain output select bit
  - Unused bit. Set to 0.

- **CSS1 - CSS0** - SCS pin select bits
  - Functions as SCS output pin
  - (Functions as a port, regardless of the values of the bits when using clock synchronous communication mode).

- **SCKS** - SSCK pin select bit
  - Functions as serial clock pin

- **BIDE** - Bidirectional mode enable bit
  - Set to 1 to use bidirectional communication mode.
  - Set to 0 to use 4-wire bus communication mode or clock synchronous communication mode.

(9) Set the RSSTP bit and transfer clock rate select bits.

![SS Control Register H Diagram]

- **SSCRH** (Address 00B8h) - Transfer clock rate select bits
  - **CKS2 - CKS0** - f1/256
  - **RSSTP** - Receive single stop bit
    - Maintains receive operation after receiving 1 byte of data

(10) Set the ORER bit in the SSSR register to 0.

![SS Status Register Diagram]

- **SSSR** (Address 00BCh) - Overrun error flag
  - No overrun errors generated
(11) Enable transmission and disable reception.

<table>
<thead>
<tr>
<th>SS Enable Register</th>
<th>SSER Address 00BBh</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7 b6 b5 b4 b3 b2 b1 b0</td>
<td></td>
</tr>
<tr>
<td>CEIE</td>
<td>Conflict error interrupt enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables conflict error interrupt request</td>
</tr>
<tr>
<td>(b2-b1)</td>
<td>Nothing is assigned.</td>
</tr>
<tr>
<td></td>
<td>Set to 0.</td>
</tr>
<tr>
<td>RE</td>
<td>Receive enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables receive</td>
</tr>
<tr>
<td>TE</td>
<td>Transmit enable bit</td>
</tr>
<tr>
<td></td>
<td>Enables transmit</td>
</tr>
<tr>
<td>RIE</td>
<td>Receive interrupt enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables receive data full and overrun error interrupt request</td>
</tr>
<tr>
<td>TEIE</td>
<td>Transmit end interrupt enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables transmit end interrupt request</td>
</tr>
<tr>
<td>TIE</td>
<td>Transmit interrupt enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables transmit data empty interrupt request</td>
</tr>
</tbody>
</table>
4.3.2 Master Transmission

(1) Read the TDRE bit in the SSSR register to confirm that the TDRE bit is 1 (data is transferred from the SSTDR register to the SSTRSR register).

(2) After confirming that the TDRE bit is 1, write transmit data to the SSTDR register. When data is written to the SSTDR register, the TDRE bit becomes 0 (data is not transferred from the SSTDR register to the SSTRSR register), and the data is transferred from the SSTDR register to the SSTRSR register. Then, the TDRE bit becomes 1 and data transmission starts.

(3) After the second byte of the transmit data, write to the SSTDR register each time the TDRE bit becomes 1.

(4) The TEND bit in the SSSR register becomes 1 when data transmission is completed.

(5) Set the TEND bit in the SSSR register to 0.

(6) Disable transmission.
### 4.3.3 Master Reception

1. Enable reception.

   **SS Enable Register**
   
<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

   SSER   Address 00BBh
   RE     Receive enable bit
   Enables receive

2. Perform a dummy read on the SSRDR register.

   **SS Receive Data Register**
   
<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   SSRDR   Address 00BFh
   Perform dummy read

3. Determine whether the receive data is the last data. If so, jump to the procedure in 4.3.4 Master Reception (When Receiving the Last Byte).

4. Confirm that the ORER bit in the SSSR register is 0 (no overrun errors generated) to determine if an overrun error occurred. If an overrun error occurred, jump to the procedure in 4.3.5 Master Reception (Overrun Error).

5. Read the RDRF bit in the SSSR register and confirm that the RDRF bit is 1 (data present in the SSRDR register).

6. After confirming that the RDRF bit is 1, read the receive data from the SSRDR register. After reading the receive data from the SSRDR register, the RDRF bit becomes 0 (no data present in the SSRDR register).
### 4.3.4 Master Reception (When Receiving the Last Byte)

1. Set the RSSTP bit in the SS control register H to 1.

   ![SS Control Register H](image)

   - **b7**: 0
   - **b6**: 1
   - **b5**: 1
   - **b4**: 0
   - **b3**: 0
   - **b2**: 0
   - **RSSTP**: 0
   - **Address**: 00B8h
   - **SSCRH**: Receive single stop bit
   - **Completes receive operation after receiving 1 byte of data**

2. Confirm that the ORER bit in the SSSR register is 0 (no overrun errors generated) to determine if an overrun error occurred. If an overrun error occurred, jump to the procedure in 4.3.5 Master Reception (Overrun Error).

3. Read the RDRF bit in the SSSR register and confirm that the RDRF bit is 1 (data present in the SSRDR register).

4. Set the RSSTP bit in the SS control register H to 0.

   ![SS Control Register H](image)

   - **b7**: 0
   - **b6**: 1
   - **b5**: 1
   - **b4**: 0
   - **b3**: 0
   - **b2**: 0
   - **RSSTP**: 0
   - **Address**: 00B8h
   - **SSCRH**: Receive single stop bit
   - **Maintains receive operation after receiving 1 byte of data**

5. Disable reception.

   ![SS Enable Register](image)

   - **b7**: 0
   - **b6**: 0
   - **b5**: 0
   - **b4**: 0
   - **b3**: 0
   - **b2**: 0
   - **RE**: 1
   - **Address**: 00BBh
   - **SSER**: Receive enable bit
   - **Disables receive**

6. Read the receive data from the SSRDR register. The RDRF bit becomes 0.

   ![SS Receive Data Register](image)

   - **b7**: 0
   - **b6**: 0
   - **b5**: 0
   - **b4**: 0
   - **b3**: 0
   - **b2**: 0
   - **SSRDR**: Address 00BFh
   - **Read the receive data**
(7) Next, enable the master transmission.

![SS Enable Register Diagram]

- **SS Enable Register**
- Address 00BBh
- TE (Transmit enable bit)
- Enables transmit
## 4.3.5 Master Reception (Overrun Error)

1. Disable reception and transmission.

   **SS Enable Register**

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

   SSER Address 00BBh

   RE: Receive enable bit
   Disables receive

   TE: Transmit enable bit
   Disables transmit

2. Set the ORER bit in the SSSR register to 0.

   **SS Status Register**

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

   SSSR Address 00BCh

   ORER Overrun error flag
   No overrun errors generated

3. Next, enable the master transmission.

   **SS Enable Register**

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

   SSER Address 00BBh

   TE: Transmit enable bit
   Enables transmit
4.4 Setting Slave Transmit/Receive Mode

4.4.1 Initial Setting
Enable transferring and set the transfer clock and transfer format.

(1) Set the port P3_7, P3_5, P3_4, and P3_3 direction bits as input ports.

(2) Set the SSU bus operation enable bit.

(3) Set SSU/I²C bus switch bit.
(4) Disable reception and transmission.

SS Enable Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **SSER** Address 00BBh
  - **CEIE** Conflict error interrupt enable bit
  - Disables conflict error interrupt request
  - **RIE** Receive interrupt enable bit
  - Disables receive data full and overrun error interrupt request
  - **TEIE** Transmit end interrupt enable bit
  - Disables transmit end interrupt request
  - **TIE** Transmit interrupt enable bit
  - Disables transmit data empty interrupt request

(5) Select the communication mode.

SS Mode Register 2

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **SSMR2** Address 00BDh
  - **SSUMS** Clock synchronous serial I/O with chip select mode select bit
    - Set to 1 to use 4-wire bus communication mode or bidirectional communication mode.
    - Set to 0 to use clock synchronous communication mode.
(6) Select MSB first.

SS Mode Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SSMR Address 00B0h</td>
</tr>
</tbody>
</table>

- **BC2 - BC0**: Bits counter 2 to 0
  - Unused bit. Set to 0.
- **(b3)**: Reserved bit
  - Set to 1.
- **(b4)**: Nothing is assigned.
  - Set to 0.
- **CPHS**: SSCK clock phase select bit
  - Change data at odd edge
  - (download data at even edge)
- **CPOS**: SSCK clock polarity select bit
  - High when clock stops
- **MLS**: MSB first/LSB first select bit
  - Transfers data MSB first

(7) Select the slave device.

SS Control Register H

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SSCRH Address 00B8h</td>
</tr>
</tbody>
</table>

- **MSS**: Master/slave device select bit
  - Operates as master device
(8) Set the bidirectional mode enable bit, SSCK pin select bit, and SCS pin select bit.

SS Mode Register 2

- **SSMR2** Address 00BDh
  - Clock synchronous serial I/O with chip select mode select bit
    - Set to 1 to use 4-wire bus communication mode or bidirectional communication mode.
    - Set to 0 to use clock synchronous communication mode.

- **SSUMS**
  - Clock synchronous serial I/O with chip select mode select bit
    - Set to 1 to use 4-wire bus communication mode or bidirectional communication mode.
    - Set to 0 to use clock synchronous communication mode.

- **CSOS**
  - SCS pin open drain output select bit
    - Unused bit. Set to 0.

- **SOOS**
  - Serial data pin open drain output select bit
    - Unused bit. Set to 0.

- **SCKOS**
  - SSCK pin open drain output select bit
    - Unused bit. Set to 0.

- **CSS1 - CSS0**
  - SCS pin select bits
    - Functions as SCS input pin (Functions as a port, regardless of the values of the bits when using clock synchronous communication mode).

- **SCKS**
  - SSCK pin select bit
    - Functions as serial clock pin

- **BIDE**
  - Bidirectional mode enable bit
    - Set to 1 to use bidirectional communication mode.
    - Set to 0 to use 4-wire bus communication mode or clock synchronous communication mode

(9) Set the transfer clock rate select bits.

SS Control Register H

- **SSCRH** Address 00B8h
  - Transfer clock rate select bits
    - f1/256

(10) Set the ORER bit in the SSSR register to 0.
(11) Disable transmission and enable reception.

**SS Enable Register**

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- **CEIE**: Conflict error interrupt enable bit
  - Disables conflict error interrupt request
- **(E2-E5)**: Nothing is assigned.
  - Set to 0.
- **RE**: Receive enable bit
  - Enables receive
- **TE**: Transmit enable bit
  - Disables transmit
- **RIE**: Receive interrupt enable bit
  - Disables receive data full and overrun error interrupt request
- **TEIE**: Transmit end interrupt enable bit
  - Disables transmit end interrupt request
- **TIE**: Transmit interrupt enable bit
  - Disables transmit data empty interrupt request

(12) Perform a dummy read on the SSRDR register.

**SS Receive Data Register**

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **SSRDR**: Address 00BFh
  - Perform dummy read
4.4.2 Slave Reception

(1) Confirm that the ORER bit in the SSSR register is 0 (no overrun errors generated) to determine if an overrun error occurred. If an overrun error occurred, jump to the procedure in 4.4.3 Slave Reception (Overrun Error).

(2) Read the RDRF bit in the SSSR register and confirm that the RDRF bit is 1 (data present in the SSRDR register).

(3) After confirming that the RDRF bit is 1, read the receive data from the SSRDR register. After reading the receive data from the SSRDR register, the RDRF bit becomes 0 (no data present in the SSRDR register).

(4) After reading the last byte of the data to be received, disable reception.

![SS Receive Data Register](image)

![SS Enable Register](image)
4.4.3 Slave Reception (Overrun Error)

(1) Disable reception and transmission.

SS Enable Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SSER Address 00Bh</td>
<td></td>
</tr>
<tr>
<td>RE</td>
<td>Receive enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables receive</td>
</tr>
<tr>
<td>TE</td>
<td>Transmit enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables transmit</td>
</tr>
</tbody>
</table>

(2) Set the ORER bit in the SSSR register to 0.

SS Status Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>b0</td>
<td>0</td>
</tr>
<tr>
<td>SSSR Address 00BCh</td>
<td></td>
</tr>
<tr>
<td>ORER</td>
<td>Overrun error flag</td>
</tr>
<tr>
<td></td>
<td>No overrun errors generated</td>
</tr>
</tbody>
</table>

(3) Enable reception.

SS Enable Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SSER Address 00Bh</td>
<td></td>
</tr>
<tr>
<td>RE</td>
<td>Receive enable bit</td>
</tr>
<tr>
<td></td>
<td>Enables receive</td>
</tr>
</tbody>
</table>

(4) Perform a dummy read on the SSRDR register.

SS Receive Data Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SSRDR Address 00BFh</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perform dummy read</td>
</tr>
</tbody>
</table>
4.4.4 Slave Transmission

(1) Enable transmission.

<table>
<thead>
<tr>
<th>SS Enable Register</th>
<th>SSER Address 00BBh</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7 0 0 0 1 0 0 0 0</td>
<td>TE Transmit enable bit</td>
</tr>
<tr>
<td></td>
<td>Enables transmit</td>
</tr>
</tbody>
</table>

(2) Write the transmit data to the SSTDR register. When data is written to the SSTDR register, the TDRE bit in the SSSR register becomes 0 (data is not transferred from the SSTDR register to the SSTRSR register), and the data is transferred from the SSTDR register to the SSTRSR register. Then, the TDRE bit becomes 1 (data is transferred from the SSTDR register to the SSTRSR register), and data transmission starts.

<table>
<thead>
<tr>
<th>SS Transmit Data Register</th>
<th>SSTDR Address 00BEh</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7 0 0 0 0 0 0 0 0</td>
<td>Write transmit data</td>
</tr>
</tbody>
</table>

(3) The TEND bit in the SSSR register becomes 1 (the TDRE bit is 1 when transmitting the last bit of transmit data) when data transmission is completed. After the second byte of the transmit data, write to the SSTDR register after confirming that the TDRE bit is 1.

(4) After transmitting a specified number of bytes, check that the TEND bit in the SSSR register is 1 to confirm completion of data transmission. Set the TEND bit in the SSSR register to 0.

<table>
<thead>
<tr>
<th>SS Status Register</th>
<th>SSSR Address 00BCh</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7 0 0 0 0 0 0 0 0</td>
<td>TEND Transmit end</td>
</tr>
<tr>
<td></td>
<td>The TDRE bit is set to 0 when transmitting the last bit of transmit data</td>
</tr>
</tbody>
</table>

(5) Disable transmission.

<table>
<thead>
<tr>
<th>SS Enable Register</th>
<th>SSER Address 00BBh</th>
</tr>
</thead>
<tbody>
<tr>
<td>b7 0 0 0 0 0 0 0 0</td>
<td>TE Transmit enable bit</td>
</tr>
<tr>
<td></td>
<td>Disables transmit</td>
</tr>
</tbody>
</table>
(6) Next, enable the slave reception.

SS Enable Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>RE</td>
</tr>
</tbody>
</table>

SSER Address 00B8h

RE Receive enable bit

Enables receive

(7) Perform a dummy read on the SSRDR register.

SS Receive Data Register

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
</table>

SSRDR Address 00BFh

Perform dummy read
5. Flowcharts

5.1 Master Transmit/Receive Mode

5.1.1 Initial Setting and Main Loop

```
main()
asm("FCLR I")
prc0 ← 1

cm14 ← 0

fra2 ← 0x00
fra00 ← 1

Repeat
(i <= 30)
i++;

fra01 ← 1

SFR Initial Setting
sfr_init()

asm("FSET I")
```

- Disable interrupt
- Disable system control register protect
- Start low-speed OCO
- High-speed OCO clock divide-by-2 mode
- Start high-speed OCO
- Wait until oscillation stabilizes
- Select high-speed OCO
- No main clock division
- Enable CM16, CM17
- System control register protect
- SFR initial setting
- Enable interrupt
INT0 interrupt requested?

Yes

int0ic ← 0x00

Clear INT0 interrupt request flag

No

INT0 interrupt request judgment

CS communication

cs_communication()

Confirm that CS communication is completed successfully

return = PASS

Yes

No

sser ← sser & 0xe7

Disable reception and transmission

Clear overrun error flag

asm("nop");

Enable transmit
5.1.2 SFR Initial Setting

```
sfr_init()

pd3 ← pd3 & 0x47

mstiic ← 1

iicsel ← 0

sser ← 0x00

ssums_ssmr2 ← SSUMS_INIT

ssmr ← 0x08

mss_sscrh ← 1

ssmr2 ← SSMR2_INIT

sscrh ← sscrh & 0x20

orer_sssr ← 0

sser ← 0x10

inten ← 0x01

intf ← 0x00

int0ic ← 0x00

return
```

P3_7(SSO), P3_5(SSCK), P3_4(SCS), P3_3(SII) input mode setting

Enable SSU operation

Select SSU function

Disable transmission and reception

Mode setting (1)

MSB first setting

Select master device

Set to bidirectional mode enable, SSCK pin select, and SCS pin select (2)

Set to maintains receive operation after receiving 1 byte of data and f1/256

Clear overrun error flag

Enable transmission

Disable reception and interrupt request

Select INT0 input enable and INT0 input polarity one edge

No INT0 input filter

Disable INT0 interrupt, clear INT0 interrupt request flag, and select INT0 input polarity falling edge

Note 1. SSUMS_INIT value in each mode is deified as below.

4-wire bus communication mode: 1

Bidirectional communication mode: 1

Clock synchronous communication mode: 0

Note 2. SSMR2_INIT value in each mode is deified as below.

4-wire bus communication mode: 0x71

Bidirectional communication mode: 0xf1

Clock synchronous communication mode: 0x40
5.1.3 CS Communication Main Routine

```
cs_communication()
```

Repeat the specified number of times

```
trdre_sssr = 1
```

Yes

SSTDR register blank check

No

```
sstdr <- data[trans_cnt]
```

Set transmit data

```
asm("nop")
```

```
asm("nop")
```

```
asm("nop")
```

```
asm("nop")
```

```
tend_sssr = 1
```

Yes

Transmit completion check

No

```
tend_sssr <- 0
```

Clear transmit end bit

```
te_sser <- 0
```

Disable transmission

```
re_sser <- 1
```

Enable reception

```
dummy_read <- ssdr
```

SSRDR register dummy read
Clock Synchronous Serial I/O with Chip Select (SSU)

Receive last 1 byte

- **rsstp_sscrh** = 1
  - **receive_cnt < 3**
    - Yes: **return(PASS)**
    - **receive_cnt < 3**
      - No: **rsstp_sscrh** = 1
        - **orer_sssr** = 1
          - Yes: **return(Overrun_ERROR)**
          - No: **rdrf_sssr** = 1
            - Yes: Read received data
              - **data_store[receive_cnt] ← ssrdr**
            - No: **rsstp_sscrh** = 0
              - **re_sser** = 0
                - **data_store[receive_cnt] ← ssrdr**
      - **receive_cnt < 3**
        - No: **rsstp_sscrh** = 1
          - **orer_sssr** = 1
            - Yes: **return(Overrun_ERROR)**
            - No: **rdrf_sssr** = 1
              - Yes: **re_sser** = 0
                - **data_store[receive_cnt] ← ssrdr**
              - No: **receive_cnt ++**

Set the RSSTP bit to 1 (completes receive operation after receiving 1 byte of data) before receiving last data

- **receive_cnt < 3**
  - Yes: **return(PASS)**
  - No: **receive_cnt < 3**
    - Yes: **orer_sssr** = 1
      - Yes: **return(Overrun_ERROR)**
      - No: **rdrf_sssr** = 1
        - Yes: Read received data
          - **data_store[receive_cnt] ← ssrdr**
        - No: **rsstp_sscrh** = 0
          - **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
    - No: **rsstp_sscrh** = 1
      - **orer_sssr** = 1
        - Yes: **return(Overrun_ERROR)**
        - No: **rdrf_sssr** = 1
          - Yes: **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
          - No: **receive_cnt ++**

Initialize receive counter

- **receive_cnt < 3**
  - Yes: **return(PASS)**
  - No: **receive_cnt < 3**
    - Yes: **orer_sssr** = 1
      - Yes: **return(Overrun_ERROR)**
      - No: **rdrf_sssr** = 1
        - Yes: Read received data
          - **data_store[receive_cnt] ← ssrdr**
        - No: **rsstp_sscrh** = 0
          - **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
    - No: **rsstp_sscrh** = 1
      - **orer_sssr** = 1
        - Yes: **return(Overrun_ERROR)**
        - No: **rdrf_sssr** = 1
          - Yes: **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
          - No: **receive_cnt ++**

Maintains receive operation after receiving 1 byte of data

- **receive_cnt < 3**
  - Yes: **return(PASS)**
  - No: **receive_cnt < 3**
    - Yes: **orer_sssr** = 1
      - Yes: **return(Overrun_ERROR)**
      - No: **rdrf_sssr** = 1
        - Yes: Read received data
          - **data_store[receive_cnt] ← ssrdr**
        - No: **rsstp_sscrh** = 0
          - **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
    - No: **rsstp_sscrh** = 1
      - **orer_sssr** = 1
        - Yes: **return(Overrun_ERROR)**
        - No: **rdrf_sssr** = 1
          - Yes: **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
          - No: **receive_cnt ++**

Disable reception

- **receive_cnt < 3**
  - Yes: **return(PASS)**
  - No: **receive_cnt < 3**
    - Yes: **orer_sssr** = 1
      - Yes: **return(Overrun_ERROR)**
      - No: **rdrf_sssr** = 1
        - Yes: Read received data
          - **data_store[receive_cnt] ← ssrdr**
        - No: **rsstp_sscrh** = 0
          - **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
    - No: **rsstp_sscrh** = 1
      - **orer_sssr** = 1
        - Yes: **return(Overrun_ERROR)**
        - No: **rdrf_sssr** = 1
          - Yes: **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
          - No: **receive_cnt ++**

Enable transmission

- **receive_cnt < 3**
  - Yes: **return(PASS)**
  - No: **receive_cnt < 3**
    - Yes: **orer_sssr** = 1
      - Yes: **return(Overrun_ERROR)**
      - No: **rdrf_sssr** = 1
        - Yes: Read received data
          - **data_store[receive_cnt] ← ssrdr**
        - No: **rsstp_sscrh** = 0
          - **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
    - No: **rsstp_sscrh** = 1
      - **orer_sssr** = 1
        - Yes: **return(Overrun_ERROR)**
        - No: **rdrf_sssr** = 1
          - Yes: **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
          - No: **receive_cnt ++**

Update receive counter

- **receive_cnt < 3**
  - Yes: **return(PASS)**
  - No: **receive_cnt < 3**
    - Yes: **orer_sssr** = 1
      - Yes: **return(Overrun_ERROR)**
      - No: **rdrf_sssr** = 1
        - Yes: Read received data
          - **data_store[receive_cnt] ← ssrdr**
        - No: **rsstp_sscrh** = 0
          - **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
    - No: **rsstp_sscrh** = 1
      - **orer_sssr** = 1
        - Yes: **return(Overrun_ERROR)**
        - No: **rdrf_sssr** = 1
          - Yes: **re_sser** = 0
            - **data_store[receive_cnt] ← ssrdr**
          - No: **receive_cnt ++**
5.2 Slave Transmit/Receive Mode

5.2.1 Initial Setting and Main Loop

```
main()
asm("FCLR I") // Disable interrupt
prc0 ← 1 // Disable system control register protect
cm14 ← 0 // Start low-speed OCO
fra2 ← 0x00 // High-speed OCO clock divide-by-2 mode
fra00 ← 1 // Start high-speed OCO
Repeat
   (i <= 30)
   i++:
   Wait until oscillation stabilizes
fra01 ← 1 // Select high-speed OCO
cm16 ← 0 // No main clock division
cm17 ← 0
cm06 ← 0
prc0 ← 0 // System control register protect
SFR Initial Setting
sfr_init() // SFR initial setting
asm("FSET I") // Enable interrupt
```

Clock Synchronous Serial I/O with Chip Select (SSU)

CS communication

```
cs_communication()
return = PASS
```

<table>
<thead>
<tr>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>CS communication</td>
</tr>
<tr>
<td>cs_communication()</td>
</tr>
<tr>
<td>return = PASS</td>
</tr>
</tbody>
</table>

No

| sser ← sser & 0xe7 |
| cser_sssr ← 0 |
| re_sser ← 1 |
| dummy_read ← ssrdr |

Confirm that CS communication is completed successfully

Disable reception and transmission

Clear overrun error flag

Enable transmit

SSRDR register dummy read
5.2.2 SFR Initial Setting

- sfr_init()
- pd3 ← pd3 & 0x47
  - P3_7(SSO), P3_5(SSCK), P3_4(SCS), P3_3(SSI) input mode setting
- mstic ← 1
  - Enable SSU operation
- iicsel ← 0
  - Select SSU function
- sser ← 0x00
  - Disable transmission and reception
- ssums_ssmr2 ← SSUMS_INIT
  - Mode setting (1)
- ssmr ← 0x08
  - MSB first setting
- mss_sscrh ← 0
  - Select slave device
- ssmr2 ← SSMR2_INIT
  - Set bidirectional mode enable, SSCK pin select, and SCS pin select (2)
- sscrh ← sscrh & 0x60
  - Set f1/256
- orer_sssr ← 0
  - Clear overrun error flag
- sser ← 0x08
  - Disable transmission and interrupt request
  - Enable reception
- dummy_read ← ssrdr
  - SSRDR register dummy read
- return

Note 1. SSUMS_INIT value in each mode is defined as below.
   - 4-wire bus communication mode: 1
   - Bidirectional communication mode: 1
   - Clock synchronous communication mode: 0

Note 2. SSMR2_INIT value in each mode is defined as below.
   - 4-wire bus communication mode: 0x51
   - Bidirectional communication mode: 0xd1
   - Clock synchronous communication mode: 0x40
5.2.3 CS Communication Main Routine

```
cs_communication()

Repeat the specified number of times

orer_sssr = 1

No

rdrf_sssr = 1

Yes

data_store[receive_cnt] ← ssrdr

transmit the specified number of bytes

re_sser ← 0

te_sser ← 1

Repeat the specified number of times

sstdr ← data_set[trans_cnt]

Set transmit data

asm("nop")

asm("nop")

asm("nop")

tdre_sssr = 1

No

Yes

transmit complete check

tend_sssr = 1

No

Yes

tend_sssr ← 0

Clear transmit end flag

ted ← 0

Enable transmission

re_sser ← 1

Enable reception

dummy_read ← ssrdr

SSRDR register dummy read

return(PASS)

When reception is completed successfully, PASS is returned

return(OVERRUN_ERROR)

If overrun error occurs, OVERRUN_ERROR is returned

Overrun occur check

Data receive check

Read received data

Disable reception

Enable transmission

Transmit the specified number of bytes

SSTDR register blank check

Transmit complete check

Clear transmit end flag

Disable transmission

Enable reception

SSRDR register dummy read
```
6. Sample Program
A sample program can be downloaded from the Renesas Electronics website.
To download, click “Application Notes” in the left-hand side menu of the R8C Family page.

7. Reference Documents
R8C/2D Group User’s Manual: Hardware
The latest version can be downloaded from the Renesas Electronics website.

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

Website and Support
Renesas Electronics website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Apr. 29, 2008</td>
<td>—</td>
<td>First edition issued</td>
</tr>
<tr>
<td>1.01</td>
<td>Dec. 20, 2010</td>
<td>3</td>
<td>Table 3.2 changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>5.1.3 Procedure of CS Communication Main Routine changed (TN-R8C/A016A/E supported)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35</td>
<td>5.2.3 Procedure of CS Communication Main Routine changed (TN-R8C/A016A/E supported)</td>
</tr>
</tbody>
</table>
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 hereafter, 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 damages 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 for or incorporated into any products or systems whose manufacturers, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.

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 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 depends 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. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

9. “Standard” - Computer; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliance; machine tools; personal electronic equipment; and industrial robots.

10. “High Quality” - Transportation equipment (automobiles, train, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

11. “Specific” - Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. aspiration, etc.); and any other applications or purposes that pose a direct threat to human life.

12. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

13. 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 the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunctions prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

14. 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 compliance with applicable laws and regulations.

15. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.

16. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Notice 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Notice 2) "Renesas Electronics products" mean any product developed or manufactured by or for Renesas Electronics.

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

SALES OFFICES

Renesas Electronics America Inc.
3980 Scott Boulevard, Santa Clara, CA 95054-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Mississauga, Ontario L4Y 9C3, Canada
Tel: +1-905-899-3441, Fax: +1-905-988-3320

Renesas Electronics Europe Limited
Dulles Meadow, Millbrow Road, Brimstone End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-358-100, Fax: +44-1628-358-900

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

Renesas Electronics (China) Co., Ltd., 7th Floor, Quantum Plaza, No.57 Zhongguancun Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd., Unit 202, 209, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858/-7898

Renesas Electronics Singapore Pte. Ltd.
1 North Quay Avenue, #06-10, Keppel Bay Tower, Singapore 089932
Tel: +65-6215-1220, Fax: +65-6215-9801

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9950, Fax: +60-3-7955-9910

Renesas Electronics Korea Co., Ltd.
11F., Bank Of Korea Bldg., 7-25 Yeoksam-dong, Gangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3757, Fax: +82-2-558-5141

© 2010 Renesas Electronics Corporation. All rights reserved.