Introduction

This application note explains how to use UART communication through the serial array unit (SAU). ASCII characters transmitted from the device on the opposite side are analyzed to make responses.

Target Device

RL78/I1A

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.
Contents

1. Specifications .................................................................................................................................... 3
2. Operation Check Conditions ............................................................................................................. 5
3. Related Application Note ................................................................................................................... 5
4. Description of the Hardware ........................................................................................................... 6
   4.1 Hardware Configuration Example ............................................................................................. 6
   4.2 List of Pins to be Used ............................................................................................................... 6
5. Description of the Software ............................................................................................................... 7
   5.1 Operation Outline ...................................................................................................................... 7
   5.2 List of Option Byte Settings ..................................................................................................... 8
   5.3 List of Constants ....................................................................................................................... 8
   5.4 List of Variables ........................................................................................................................ 8
   5.5 List of Functions ......................................................................................................................... 9
   5.6 Function Specifications ............................................................................................................. 9
   5.7 Flowcharts ................................................................................................................................ 12
      5.7.1 Initialization Function ......................................................................................................... 12
      5.7.2 System Function .................................................................................................................. 13
      5.7.3 I/O Port Setup ..................................................................................................................... 14
      5.7.4 CPU Clock Setup ............................................................................................................... 15
      5.7.5 Serial Array Unit Setup .................................................................................................... 16
      5.7.6 UART0 Setup ..................................................................................................................... 18
      5.7.7 Main Function ..................................................................................................................... 29
      5.7.8 Main initializes settings ....................................................................................................... 32
      5.7.9 UART0 Reception Status Initialization Function ................................................................ 33
      5.7.10 UART0 Operation Start Function ..................................................................................... 34
      5.7.11 INTSR0 Interrupt Service Routine .................................................................................... 38
      5.7.12 UART0 Receive Data Classification Function ................................................................. 39
      5.7.13 UART0 Data Transmission Function ............................................................................... 40
      5.7.14 UART0 Reception Error Interrupt Function .................................................................... 41
      5.7.15 UART0 Reception Error Classification Function ............................................................ 42
      5.7.16 INTST0 Interrupt Service Routine ................................................................................... 43
      5.7.17 UART0 Transmission End Processing Function ............................................................... 44
6. Sample Code ................................................................................................................................... 45
7. Documents for Reference ............................................................................................................... 45
1. Specifications

In this application note, UART communication is performed through the serial array unit (SAU). ASCII characters transmitted from the device on the opposite side are analyzed to make responses.

Table 1.1 shows the peripheral function to be used and its use. Figures 1.1 and 1.2 illustrate UART communication operation.

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial array unit 0</td>
<td>Perform UART communication using the TxD0 pin (transmission) and the RxD0 pin (reception).</td>
</tr>
</tbody>
</table>

![Figure 1.1 UART Reception Timing Chart](image-url)

- SAUEN: Permit operation of UART
- SS01: Permit start of UART communication
- SE01: 
- SDR01: Receive data
- RxD0 pin: Receive data
- Shift register 01: Reception & shift operation
- INTSR0: 
- TSF01: 

ST: Start bit
P: Parity bit
SP: Stop bit
Figure 1.2   UART Transmission Timing Chart

- Permit operation of UART
- Transmit data
- Shift operation
- Permit start of UART communication
- Transfer completed

SAUEN
SS00
SE00
SDR00
TxD0 pin
Shift register 00
INTST0
TSF00

ST: Start bit
P: Parity bit
SP: Stop bit
2. **Operation Check Conditions**

The sample code contained in this application note has been checked under the conditions listed in the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Microcontroller used</strong></td>
<td>RL78/I1A (R5F107DE)</td>
</tr>
</tbody>
</table>
| **Operating frequency**           | • High-speed on-chip oscillator (HOCO) clock: 24 MHz  
• CPU/peripheral hardware clock: 24 MHz |
| **Operating voltage**             | 5.0 V (can run on a voltage range of 2.9 V to 5.5 V.)  
LVD operation (V_LVD): Reset mode 2.81 V (2.76V to 2.87V) |
| **Integrated development environment (CS+)** | Renesas Electronics Corporation  
CS+ for CC V3.03.00 |
| **C compiler (CS+)**              | Renesas Electronics Corporation  
CC-RL V1.02.00 |
| **Integrated development environment (e² studio)** | Renesas Electronics Corporation  
e² studio V4.0.0.26 |
| **C compiler (e² studio)**        | Renesas Electronics Corporation  
CC-RL V1.02.00 |
| **Board**                         | RL78/I1A CPU board (QB-R5F107DE-TB)              |

3. **Related Application Note**

The application note that is related to this application note is listed below for reference.

RL78/G13 Serial Array Unit (UART Communication) CC-RL (R01AN2517E) Application Note
4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used for this application note.

Caution: 1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to \( V_{DD} \) or \( V_{SS} \) via a resistor).

2. \( V_{DD} \) must be held at not lower than the reset release voltage (\( V_{LVD} \)) that is specified as LVD.

4.2 List of Pins to be Used

Table 4.1 lists the pins to be used and their function.

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P10/SO00/TxD0/TKC00/INTP20/SCLA0</td>
<td>Output</td>
<td>Data transmission pin</td>
</tr>
<tr>
<td>P11/SI00/RxD0/TKCO01/INTP21 /SDAA0</td>
<td>Input</td>
<td>Data reception pin</td>
</tr>
</tbody>
</table>
5. Description of the Software

5.1 Operation Outline

This sample code transmits, to the device on the opposite side, the data corresponding to that received from the device. If an error occurs, it transmits to the device the data corresponding to the error. Tables 5.1 and 5.2 show the correspondence between transmit data and receive data.

### Table 5.1 Correspondence between Receive Data and Transmit Data

<table>
<thead>
<tr>
<th>Receive Data</th>
<th>Response (Transmit) Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>T (54H)</td>
<td>O (4FH), K (4BH), “CR” (0DH), “LF” (0AH)</td>
</tr>
<tr>
<td>t (74H)</td>
<td>o (6FH), k (6BH), “CR” (0DH), “LF” (0AH)</td>
</tr>
<tr>
<td>Other than above</td>
<td>U (55H), C (43H), “CR” (0DH), “LF” (0AH)</td>
</tr>
</tbody>
</table>

### Table 5.2 Correspondence between Error and Transmit Data

<table>
<thead>
<tr>
<th>Error</th>
<th>Response (Transmit) Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parity error</td>
<td>P (50H), E (45H), “CR” (0DH), “LF” (0AH)</td>
</tr>
<tr>
<td>Framing error</td>
<td>F (46H), E (45H), “CR” (0DH), “LF” (0AH)</td>
</tr>
<tr>
<td>Overrun error</td>
<td>O (4FH), E (45H), “CR” (0DH), “LF” (0AH)</td>
</tr>
</tbody>
</table>

1. Perform initial setting of UART.

   <UART Setting Conditions>
   - Use SAU0 channels 0 and 1 as UART.
   - Use the P10/TxD0 pin and the P11/RxD0 pin for data output and data input, respectively.
   - The data length is 8 bits.
   - Set the data transfer direction to LSB first.
   - Use even parity as the parity setting.
   - Set the receive data level to standard.
   - Set the transfer rate to 9600 bps.
   - Use reception end interrupt (INTSR0), transmission end interrupt (INTST0), and error interrupt (INTSRE0).
   - Select interrupt priority level 2 or 1 for INTSR0 and for INTSRE0. Select the low interrupt priority level (level 3) for INTST0.

2. After the system is made to enter a UART communication wait state by using the serial channel start register, a HALT instruction is executed. Processing is performed in response to reception end interrupt (INTSR0) and error interrupt (INTSRE0).
   - When an INTSR0 occurs, the received data is taken in and the data corresponding to the received data is transmitted. When an INTSRE0 occurs, error handling is performed to transmit the data corresponding to the error.
   - After data transmission, a HALT instruction is executed again to wait for reception end interrupt (INTSR0) and error interrupt (INTSRE0).
5.2 List of Option Byte Settings

Table 5.1 summarizes the settings of the option bytes.

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000C0H/010C0H</td>
<td>11101111B</td>
<td>Disables the watchdog timer. (Stops counting after the release from the reset state.)</td>
</tr>
<tr>
<td>000C1H/010C1H</td>
<td>01111111B</td>
<td>LVD reset mode, 2.81 V (2.76V to 2.87V)</td>
</tr>
<tr>
<td>000C2H/010C2H</td>
<td>11101000B</td>
<td>HS mode; HOCO: 24MHz</td>
</tr>
<tr>
<td>000C3H/010C3H</td>
<td>10001000B</td>
<td>Enables the on-chip debugger.</td>
</tr>
</tbody>
</table>

5.3 List of Constants

Table 5.2 lists the constants that are used in this sample program.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_messageOK[4]</td>
<td>&quot;OK\r\n&quot;</td>
<td>Response message to reception of &quot;T&quot;.</td>
</tr>
<tr>
<td>g_messageok[4]</td>
<td>&quot;ok\r\n&quot;</td>
<td>Response message to reception of &quot;t&quot;.</td>
</tr>
<tr>
<td>g_messageUC[4]</td>
<td>&quot;UC\r\n&quot;</td>
<td>Response message to reception of characters other than &quot;T&quot; or &quot;t&quot;.</td>
</tr>
<tr>
<td>g_messageFE[4]</td>
<td>&quot;FE\r\n&quot;</td>
<td>Response message to a framing error.</td>
</tr>
<tr>
<td>g_messagePE[4]</td>
<td>&quot;PE\r\n&quot;</td>
<td>Response message to a parity error.</td>
</tr>
<tr>
<td>g_messageOE[4]</td>
<td>&quot;OE\r\n&quot;</td>
<td>Response message to an overrun error.</td>
</tr>
</tbody>
</table>

5.4 List of Variables

Table 5.3 lists the global variable that is used by this sample program.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint8_t</td>
<td>g_uart0_rx_buffer</td>
<td>Receive data buffer</td>
<td>main()</td>
</tr>
<tr>
<td>uint8_t</td>
<td>gp_uart0_tx_address</td>
<td>Transmit data pointer</td>
<td>R_UART0_Send(), R_UART0_Interrupt_Send()</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_uart0_tx_count</td>
<td>Transmit data number counter</td>
<td>R_UART0_Send(), R_UART0_Interrupt_Send()</td>
</tr>
<tr>
<td>uint8_t</td>
<td>gp_uart0_rx_address</td>
<td>Receive data pointer</td>
<td>R_UART0_Receive(), R_UART0_Interrupt_Receive(), R_UART0_Interrupt_Error()</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_uart0_rx_count</td>
<td>Receive data number counter</td>
<td>R_UART0_Receive(), R_UART0_Interrupt_Receive()</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_uart0_rx_length</td>
<td>Receive data number</td>
<td>R_UART0_Receive(), R_UART0_Interrupt_Receive()</td>
</tr>
<tr>
<td>MD_STATUS</td>
<td>g_uart0_tx_end</td>
<td>Transmit status</td>
<td>main(), r_uart0_callback_sendend()</td>
</tr>
<tr>
<td>uint8_t</td>
<td>g_uart0_rx_error</td>
<td>Receive error status</td>
<td>main(), r_uart0_callback_receiveend(), r_uart0_callback_error()</td>
</tr>
</tbody>
</table>
## 5.5 List of Functions

Table 5.4 lists the functions that are used in this sample program.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_UART0_Start</td>
<td>UART0 operation start</td>
</tr>
<tr>
<td>R_UART0_Receive</td>
<td>UART0 reception status initialization function</td>
</tr>
<tr>
<td>R_UART0_Send</td>
<td>UART0 data transmission function</td>
</tr>
<tr>
<td>r_uart0_interrupt_receive</td>
<td>UART0 reception end interrupt handling</td>
</tr>
<tr>
<td>r_uart0_callback_receiveend</td>
<td>UART0 receive data classification function</td>
</tr>
<tr>
<td>r_uart0_interrupt_error</td>
<td>UART0 error interrupt handling</td>
</tr>
<tr>
<td>r_uart0_callback_error</td>
<td>UART0 reception error classification function</td>
</tr>
<tr>
<td>r_uart0_interrupt_send</td>
<td>UART0 transmission end interrupt handling</td>
</tr>
<tr>
<td>r_uart0_callback_sendend</td>
<td>UART0 transmission end processing function</td>
</tr>
<tr>
<td>r_uart0_callback_softwareoverrun</td>
<td>UART0 overflow data receive function</td>
</tr>
</tbody>
</table>

## 5.6 Function Specifications

This section describes the specifications for the functions that are used in this sample program.

### [Function Name] R_UART0_Start

- **Synopsis**: UART0 operation start
- **Header**: r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
- **Declaration**: void R_UART0_Start(void)
- **Explanation**: Starts operation of channel 0 of serial array units 0 and 1 to make the system enter a communication wait state.
- **Arguments**: None
- **Return value**: None
- **Remarks**: None

### [Function Name] R_UART0_Receive

- **Synopsis**: UART0 reception status initialization function
- **Header**: r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h
- **Declaration**: MD_STATUS R_UART0_Receive(uint8_t *rx_buf, uint16_t rx_num)
- **Explanation**: Makes initial setting for UART0 reception.
- **Arguments**: uint8_t *rx_buf : [Receive data buffer address]  
  uint16_t rx_num : [Receive data buffer size]
- **Return value**: [MD_OK]: Reception setting is completed  
  [MD_ARGERROR]: Reception setting failed
- **Remarks**: None
[Function Name] R_UART0_Send

Synopsis UART0 data transmission function

Header r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h

Declaration MD_STATUS R_UART0_Send(uint8_t* tx_buf, uint16_t tx_num)

Explanation Makes initial setting for UART0 transmission, and starts data transmission.

Arguments uint8_t *tx_buf : [Transmit data buffer address]
uint16_t tx_num : [Transmit data buffer size]

Return value [MD_OK]: Transmission setting is completed
[MD_ARGERROR]: Transmission setting failed

Remarks None

[Function Name] r_uart0_interrupt_receive

Synopsis UART0 reception end interrupt handling

Header r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h

Declaration static void __near r_uart0_interrupt_receive(void)

Explanation Makes a response (data transmission) corresponding to received data.

Arguments None

Return value None

Remarks None

[Function Name] r_uart0_interrupt_error

Synopsis UART error interrupt function

Header r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h

Declaration static void __near r_uart0_interrupt_error(void)

Explanation Transmits the data corresponding to a detected error.

Arguments None

Return value None

Remarks None

[Function Name] r_uart0_callback_receiveend

Synopsis UART0 receive data classification function

Header r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h

Declaration static void r_uart0_callback_receiveend(void)

Explanation Clears the reception error flag.

Arguments None

Return value None

Remarks None

[Function Name] r_uart0_callback_error

Synopsis UART0 reception error classification function

Header r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h

Declaration static void r_uart0_callback_error(uint8_t err_type)

Explanation Makes flag setting for transmission of the data corresponding to an error.

Arguments err_type : Error type

Return value None

Remarks None
### r_uart0_interrupt_send

**Synopsis:** UART0 transmission end interrupt handling

**Header:** r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h

**Declaration:**
```
static void __near r_uart0_interrupt_send(void)
```

**Explanation:** Transmits a specified number of pieces of data.

**Arguments:** None

**Return value:** None

**Remarks:** None

### r_uart0_callback_sendend

**Synopsis:** UART0 transmission end processing function

**Header:** r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h

**Declaration:**
```
static void r_uart0_callback_sendend(void)
```

**Explanation:** Makes transmission end flag setting.

**Arguments:** None

**Return value:** None

**Remarks:** None

### r_uart0_callback_softwareoverrun

**Synopsis:** UART0 overflow data receive function

**Header:** r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h

**Declaration:**
```
static void r_uart0_callback_softwareoverrun(void)
```

**Explanation:** Executes when detected overflow of data by software.

**Arguments:** None

**Return value:** None

**Remarks:** Unused function
5.7 Flowcharts

Figure 5.1 shows the overall flow of the sample program described in this application note.

Note: Startup routine is executed before and after the initialization function.

5.7.1 Initialization Function

Figure 5.2 shows the flowchart for the initialization function.
5.7.2 System Function

Figure 5.3 shows the flowchart for the system function.

```
R_Systeminit()

Disuse peripheral I/O redirection function

PIOR register ← 00000000B

Set up CPU clock
R_CGC_Create()

Set up I/O ports
R_PORT_Create()

Set up Serial Array Unit
R_SAU0_Create()

Disabled the invalid memory access detection

IAWCTL register ← 00000000B

return
```

Figure 5.3 System Function
### 5.7.3 I/O Port Setup

Figure 5.4 shows the flowchart for setting up the I/O ports.

![Flowchart for I/O Port Setup](image)

**Figure 5.4  I/O Port Setup**

**Note:** Refer to the RL78/I1A User’s Manual: Hardware for the setting of the unused ports.

**Caution:** Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to $V_{DD}$ or $V_{SS}$ via a separate resistor.
5.7.4 CPU Clock Setup

Figure 5.5 shows the flowchart for setting up the CPU clock.

Figure 5.5  CPU Clock Setup

- **R_CGC_Create()**
  - Set up high-speed system clock/subsystem clock
  - Select CPU/peripheral hardware clock ($f_{CLK}$)

- CMC register ← 00H: Does not use high-speed system clock and subsystem clock.
- MSTOP bit ← 1
- MCM0 bit ← 0: Selects high-speed OCO clock ($f_{OCO}$) as main system clock ($f_{MAIN}$).
- CSS bit ← 0: Selects main system clock ($f_{MAIN}$) as CPU/peripheral hardware clock ($f_{CLK}$).

return
5.7.5 Serial Array Unit Setup
Figure 5.6 shows the flowchart for setting up the serial array unit.

![Flowchart for Serial Array Unit Setup]

- R_SAU0_Create()
  - Supply input clock to serial array unit 0
    - SAU0EN bit ← 1
  - Set CK00 and CK01 to 1.5MHz
    - SPS0 register ← 44H
  - Set UART0
    - R_UART0_Create()
  - return

Figure 5.6 Serial Array Unit Setup
Start supplying clock to the SAU

- Peripheral enable register 0 (PER0)
  Clock supply

Symbol: PER0

<table>
<thead>
<tr>
<th>Bit</th>
<th>RTCEN</th>
<th>ADCEN</th>
<th>IICA0EN</th>
<th>SAU0EN</th>
<th>TAU0EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 2

<table>
<thead>
<tr>
<th>Input clock control for serial array unit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Select serial clock

- Serial clock select register 0 (SPS0)
  Operation clock setting

Symbol: SPS0

<table>
<thead>
<tr>
<th>Bit 7 to 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

Bits 7 to 0

<table>
<thead>
<tr>
<th>PRS 0n3</th>
<th>PRS 0n2</th>
<th>PRS 0n1</th>
<th>PRS 0n0</th>
<th>Operation clock (CK00) selection (n = 0, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0</td>
<td>fCLK = 2 MHz</td>
<td>fCLK = 5 MHz</td>
<td>fCLK = 10 MHz</td>
<td>fCLK = 20 MHz</td>
</tr>
<tr>
<td>0 0 0 0 1 0 0</td>
<td>fCLK/2 = 1 MHz</td>
<td>fCLK/2 = 2.5 MHz</td>
<td>fCLK/2 = 5 MHz</td>
<td>fCLK/2 = 10 MHz</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0</td>
<td>fCLK/2^5 = 1.25 MHz</td>
<td>fCLK/2^5 = 2.5 MHz</td>
<td>fCLK/2^5 = 5 MHz</td>
<td>fCLK/2^5 = 10 MHz</td>
</tr>
<tr>
<td>0 1 1 0 0 0 0</td>
<td>fCLK/2^7 = 250 kHz</td>
<td>fCLK/2^7 = 625 kHz</td>
<td>fCLK/2^7 = 1.25 MHz</td>
<td>fCLK/2^7 = 2.5 MHz</td>
</tr>
<tr>
<td>0 1 1 0 1 0 0</td>
<td>fCLK/2^7 = 250 kHz</td>
<td>fCLK/2^7 = 625 kHz</td>
<td>fCLK/2^7 = 1.25 MHz</td>
<td>fCLK/2^7 = 3 MHz</td>
</tr>
<tr>
<td>0 0 0 1 0 0 0</td>
<td>fCLK/2^9 = 3.91 kHz</td>
<td>fCLK/2^9 = 7.81 kHz</td>
<td>fCLK/2^9 = 15.62 kHz</td>
<td>fCLK/2^9 = 31.25 kHz</td>
</tr>
<tr>
<td>0 0 0 1 1 0 0</td>
<td>fCLK/2^9 = 3.91 kHz</td>
<td>fCLK/2^9 = 7.81 kHz</td>
<td>fCLK/2^9 = 15.62 kHz</td>
<td>fCLK/2^9 = 31.25 kHz</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0</td>
<td>fCLK/2^9 = 3.91 kHz</td>
<td>fCLK/2^9 = 7.81 kHz</td>
<td>fCLK/2^9 = 15.62 kHz</td>
<td>fCLK/2^9 = 31.25 kHz</td>
</tr>
<tr>
<td>1 0 0 0 0 0 0</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
<tr>
<td>1 0 0 0 1 0 0</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
<tr>
<td>1 0 0 1 0 0 0</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
<tr>
<td>1 0 0 1 1 0 0</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
<tr>
<td>1 0 0 1 1 1 0</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
<tr>
<td>1 0 0 1 1 1 0</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
<tr>
<td>1 0 0 1 1 1 1</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
<tr>
<td>1 0 0 1 1 1 1</td>
<td>fCLK/2^10 = 7.81 kHz</td>
<td>fCLK/2^10 = 15.62 kHz</td>
<td>fCLK/2^10 = 31.25 kHz</td>
<td>fCLK/2^10 = 62.5 kHz</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User’s Manual: Hardware.
5.7.6 UART0 Setup

Figures 5.7, 5.8, and 5.9 show the flowcharts for setting up UART0.

![Flowchart of UART0 Setup](image)

**R_UART0_Create()**
- Stop operation of channels 0 and 1.
  - ST0 register $\leftarrow 03H$

**Disable INTST0 interrupt & clear the interrupt request flag**
- STMK0 bit $\leftarrow 1$
- STIF0 bit $\leftarrow 0$

**Disable INTSR0 interrupt & clear the interrupt request flag**
- SRMK0 bit $\leftarrow 1$
- SRIF0 bit $\leftarrow 0$

**Disable INTSRE0 interrupt & clear the interrupt request flag**
- SREMK0 bit $\leftarrow 1$
- SREIF0 bit $\leftarrow 0$

**INTST0 interrupt priority level:**
- Select level 3 (lowest)
  - STPR10 bit $\leftarrow 1$
  - STPR00 bit $\leftarrow 1$

**INTSR0 interrupt priority level:**
- Select level 2
  - SRPR10 bit $\leftarrow 1$
  - SRPR00 bit $\leftarrow 0$

**INTSRE0 interrupt priority level:**
- Select level 1
  - SREPR10 bit $\leftarrow 0$
  - SREPR00 bit $\leftarrow 1$

**Set SAU channel 0 operation mode**
- Channel 0 operation clock:
  - CK00
- Channel 0 transfer clock:
  - Division of CK00
- Start trigger source:
  - Software trigger
- Detect a falling edge as a start bit
- Channel 0 operation mode:
  - UART mode
- Channel 0 interrupt source:
  - Transfer end interrupt

**SMR00 register $\leftarrow 0022H$**

**1**

*Figure 5.7 UART0 Setup (1/3)*
Figure 5.8  UART0 Setup (2/3)

SAU channel 0 communication operation setting
- Only perform transmission
- Phase with clock: type 1
- Error interrupt INTSREx : Masked
- Output even parities
- Input/output in LSB first
- Stop bit length: 1 bit
- Data length: 8 bits

Channel 0 transfer clock : Operation clock divided by 156

RxD3 pin noise filter
RxD2 pin noise filter
RxD1 pin noise filter : OFF
RxD0 pin noise filter : ON

Clear the error flag

SAU channel 1 operation mode setting
- Channel 1 operation clock: CK00
- Channel 1 transfer clock : Division of CK00
- Start trigger source : Valid edge of the RxD pin
- Detect a falling edge as a start bit
- Channel 1 operation mode : UART mode
- Channel 1 interrupt source : Transfer end interrupt

SCR00 register ← 8297H
SDR00 register ← 9A00H
NFEN0 register ← 0001H
SIR01 register ← 0007H
SMR01 register ← 0122H
Figure 5.9 UART0 Setup (3/3)

2

SAU channel 1 communication operation setting
- Only perform reception
- Phase with clock: type 1
- Error interrupt INTSREx: Enabled
- Check for even parity
- Input/output in LSB first
- Stop bit length: 1 bit
- Data length: 8 bits

Channel 1 transfer clock: Operation clock divided by 156

SCR01 register ← 4697H

SDR01 register ← 9A00H

SO00 bit ← 1
S0L0 bit ← 0
SOE00 bit ← 1
PM11 bit ← 1

P10 bit ← 1
PM10 bit ← 0

return

Prepare for the use of channel 0

Set the PxDO pin (input mode)

Set the TxDO pin (output mode)
Transmission channel operation mode setting

- **Serial mode register 00 (SMR00)**
  - Inteerrupt source
  - Operation mode
  - Transfer clock selection
  - \( f_{MCK} \) selection

**Symbol: SMR00**

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>CKS00</th>
<th>Channel 0 operation clock (( f_{MCK} )) selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Prescaler output clock CK00 configured by the SPS0 register</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Prescaler output clock CK01 configured by the SPS0 register</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 14</th>
<th>CCS00</th>
<th>Channel 0 transfer clock (TCLK) selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Clock obtained by dividing the operation clock ( f_{MCK} ) specified by the CKS00 bit.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Clock input from the SCK pin.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 2 and 1</th>
<th>MD002</th>
<th>MD001</th>
<th>Channel 0 operation mode setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>CSI mode</td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>UART mode</td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>Simplified I 2C mode</td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>Setting prohibited</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 0</th>
<th>MD000</th>
<th>Channel 0 interrupt source selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Transfer end interrupt</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Buffer empty interrupt</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Transmission channel communication operation setting

- Serial communication operation setting register 00 (SCR00)
  
  Data length setting, data transfer order, error interrupt signal mask availability, and operation mode

Symbol: SCR00

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXE00</td>
<td>15, 14</td>
<td>TXE00 RXE00 Channel 0 operation mode setting</td>
</tr>
<tr>
<td>RXE00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
<th>Operation Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Communication prohibited</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Reception Only</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Transmission only</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Both transmission and reception</td>
</tr>
</tbody>
</table>

Bit 10

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOC00</td>
<td>10</td>
<td>EOC00 Error interrupt signal (INTSREx (x = 0, 1)) mask availability selection</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Error interrupt INTSREx is masked</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Generation of error interrupt INTSREx is enabled</td>
</tr>
</tbody>
</table>

Bits 9 and 8

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTC001</td>
<td>9, 8</td>
<td>PTC001 PTC000 Parity bit setting in UART mode</td>
</tr>
<tr>
<td>PTC000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Transmission</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>No parity bit is output</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0 parity is output</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Even parity is output</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Odd parity is output</td>
</tr>
</tbody>
</table>

Bit 7

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR00</td>
<td>7</td>
<td>DIR00 Selection of data transfer order in CSI and UART modes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Input and output in MSB first</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Input and output in LSB first</td>
</tr>
</tbody>
</table>

Bits 5 and 4

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLC001</td>
<td>5, 4</td>
<td>SLC001 SLC000 Stop bit setting in UART mode</td>
</tr>
<tr>
<td>SLC000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>No stop bit</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Stop bit length = 1 bit</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Stop bit length = 2 bits</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Setting prohibited</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Symbol: SDR00

<table>
<thead>
<tr>
<th>TXE 00</th>
<th>RXE 00</th>
<th>DAP 00</th>
<th>CKP 00</th>
<th>EOC 00</th>
<th>PTC 000</th>
<th>PTC 001</th>
<th>DIR 00</th>
<th>0</th>
<th>SLC 0001</th>
<th>SLC 000</th>
<th>0</th>
<th>1</th>
<th>DLS 001</th>
<th>DLS 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Bits 1 and 0

<table>
<thead>
<tr>
<th>DLS001</th>
<th>DLS000</th>
<th>Data length setting in CSI mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>9-bit data length</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>7-bit data length</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>8-bit data length</td>
</tr>
</tbody>
</table>

Transmission channel transfer clock setting

- Serial data register 00 (SDR00)
  Transfer clock frequency: \( f_{MCK}/156 \approx 9600 \text{ Hz} \)

Symbol: SDR00

<table>
<thead>
<tr>
<th>SDR00[15:9]</th>
<th>Transfer clock setting by dividing operation clock ( f_{MCK} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>( f_{MCK}/2 )</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>( f_{MCK}/4 )</td>
</tr>
<tr>
<td>0 0 0 0 0 1 0 0</td>
<td>( f_{MCK}/8 )</td>
</tr>
<tr>
<td>0 0 0 0 0 1 1 1</td>
<td>( f_{MCK}/156 )</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Reception channel operation mode setting
- Serial mode register 01 (SMR01)
  - Interrupt source
  - Operation mode
  - Transfer clock selection
  - fMCK selection

Symbol: SMR01

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>CKS01 Channel 1 operation clock (fMCK) selection</td>
</tr>
<tr>
<td>14</td>
<td>CCS01 Channel 1 transfer clock (TCLK) selection</td>
</tr>
<tr>
<td>8</td>
<td>STS01 Start trigger source selection</td>
</tr>
<tr>
<td>6</td>
<td>SIS010 Control of receive data level inversion on channel 1 in UART mode</td>
</tr>
<tr>
<td>2 and 1</td>
<td>MD012 MD011 Channel 1 operation mode setting</td>
</tr>
<tr>
<td>0</td>
<td>MD010 Channel 1 interrupt source selection</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Reception channel communication operation setting

- Serial communication operation setting register 01 (SCR01)
  
  Data length setting, data transfer order, error interrupt signal mask availability, and operation mode

Symbol: SCR01

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXE01</td>
<td>RXE01</td>
<td>DAP01</td>
<td>CKP01</td>
<td>0</td>
<td>EOC01</td>
<td>PTC01</td>
<td>PTC010</td>
<td>DIR01</td>
<td>0</td>
<td>0</td>
<td>SLC010</td>
<td>0</td>
<td>1</td>
<td>DLS011</td>
<td>DLS010</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Bits 15 and 14

<table>
<thead>
<tr>
<th>TXE01</th>
<th>RXE01</th>
<th>Channel 1 operation mode setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>Communication prohibited</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>Reception only</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>Transmission only</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>Both transmission and reception</td>
</tr>
</tbody>
</table>

For UART reception, wait for 4 fCLK clock cycles or more before setting SS01 to 1, after setting the RXE01 bit of the SCR01 register to 1.

Bit 10

<table>
<thead>
<tr>
<th>EOC01</th>
<th>Error interrupt signal (INTSRE1) mask availability selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Error interrupt INTSRE1 is masked</td>
</tr>
<tr>
<td>1</td>
<td>Generation of error interrupt INTSRE1 is enabled</td>
</tr>
</tbody>
</table>

Bits 9 and 8

<table>
<thead>
<tr>
<th>PTC011</th>
<th>PTC010</th>
<th>Parity bit setting in UART mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Transmission</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>No parity bit is output</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>0 parity is output</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>Even parity is output</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>Odd parity is output</td>
</tr>
</tbody>
</table>

Bit 7

<table>
<thead>
<tr>
<th>DIR01</th>
<th>Selection of data transfer order in CSI and UART modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Input and output in MSB first</td>
</tr>
<tr>
<td>1</td>
<td>Input and output in LSB first</td>
</tr>
</tbody>
</table>

Bits 5 and 4

<table>
<thead>
<tr>
<th>SLC011</th>
<th>SLC010</th>
<th>Stop bit setting in UART mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>No stop bit</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>Stop bit length = 1 bit</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>Stop bit length = 2 bits</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>Setting prohibited</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Serial data register 01 (SDR01)

Transfer clock frequency: \( f_{\text{MCK}} / 156 \) (\( \approx 9600 \) Hz)

Reception transfer clock setting
- Serial data register 01 (SDR01)
  - Transfer clock frequency: \( f_{\text{MCK}} / 156 \) (\( \approx 9600 \) Hz)

Symbol: SDR01

<table>
<thead>
<tr>
<th>TXE</th>
<th>RXE</th>
<th>DAP</th>
<th>CKP</th>
<th>EOC</th>
<th>PTC</th>
<th>PTC</th>
<th>DIR</th>
<th>SLC</th>
<th>SLC</th>
<th>DLS</th>
<th>DLS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Bits 1 and 0

<table>
<thead>
<tr>
<th>DLS011</th>
<th>DLS010</th>
<th>Data length setting in CSI mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>9-bit data length</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>7-bit data length</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>8-bit data length</td>
</tr>
</tbody>
</table>

Others: Setting prohibited

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Initial output level setting

- Serial output register 0 (SO0)
  Initial output: 1

Symbol: SO0

```
  15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 1 CKO 02 1 CKO 00 0 0 0 1 SO 02 1 SO 00
0 0 0 0 1 x 1 x 0 0 0 1 x 1 1
```

Bit 0

<table>
<thead>
<tr>
<th>SO00</th>
<th>Channel 0 serial data output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Serial data output value is &quot;0&quot;</td>
</tr>
<tr>
<td>1</td>
<td>Serial data output value is &quot;1&quot;</td>
</tr>
</tbody>
</table>

Enabling of data output on target channel

- Serial output enable register 0 (SOE0)
  Output enable

Symbol: SOE0

```
  15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 SOE 02 0 SOE 00
0 0 0 0 0 0 0 0 0 0 0 0 0 x 0 1
```

Bit 0

<table>
<thead>
<tr>
<th>SOE00</th>
<th>Channel 0 serial output enable/stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Serial communication output is stopped</td>
</tr>
<tr>
<td>1</td>
<td>Serial communication output is enabled</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Port setting

- Port register 1 (P0)
- Port mode register 1 (PM1)

Port setting for each of transmit data and receive data.

Symbol: P1

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>P12</td>
<td>P11</td>
<td>P10</td>
</tr>
</tbody>
</table>

Bit 1

- **P11 Output data control (in output mode)**
  - 0: 0 is output
  - 1: 1 is output

Bit 0

- **P10 Output data control (in output mode)**
  - 0: 0 is output
  - 1: 1 is output

Symbol: PM1

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>PM12</td>
<td>PM11</td>
<td>PM10</td>
</tr>
</tbody>
</table>

Bit 1

- **PM11 P11 I/O mode selection**
  - 0: Output mode (output buffer is on)
  - 1: Input mode (output buffer is off)

Bit 0

- **PM10 P10 I/O mode selection**
  - 0: Output mode (output buffer is on)
  - 1: Input mode (output buffer is off)

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
5.7.7 Main Function

Figures 5.10, 5.11 and 5.12 show the flowchart for the main function.

![Flowchart for Main Function](image)
Figure 5.11  Main Function (2/3)
Figure 5.12  Main Function (3/3)
5.7.8 Main initializes settings

Figure 5.13 shows the flowchart for the main initializes settings.

![Main initializes settings flowchart]

**Figure 5.13 Main initializes settings**
5.7.9 UART0 Reception Status Initialization Function

Figure 5.14 shows the flowchart for the UART0 reception status initialization function.

![Flowchart for UART0 Reception Status Initialization Function]

- **R_UART0_Receive()**
- Set the initial status value to OK
- **Number of pieces of data = 0?**
  - Yes: Set the status to ERROR
  - No: Initialize the receive data number counter, Set the number of pieces of receive data, Initialize the receive data pointer
- Set the return value to the status
- Return

Figure 5.14  UART0 Reception Status Initialization Function
5.7.10 UART0 Operation Start Function

Figure 5.15 shows the flowchart for the UART0 operation start function.

```
R_UART0_Start()

Set the TxD0 output level

Enable UART0 output

Enable UART0 operation

Clear the transmission interrupt request flag

Clear the reception interrupt request flag

Enable transmission interrupt

Enable reception interrupt

Return
```

- SO0.0 bit ← 1
- SOE0.0 bit ← 1
- SS0.0 bit ← 1
- SS0.1 bit ← 1
- STIF0 bit ← 0
- SRIF0 bit ← 0
- SREIF0 bit ← 0
- STMK0 bit ← 0
- SRMK0 bit ← 0
- SREMK0 bit ← 0

Figure 5.15 UART0 Operation Start Function
Interrupt setting
- Interrupt request flag register (IF0H)
  Clear the interrupt request flag
- Interrupt mask flag register (MK0H)
  Cancel interrupt mask

Symbol: IF0H

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SREIF0</td>
<td>SRIF0</td>
<td>STIF0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</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>

Bit 7
- **SREIF0**: Interrupt request flag
  - 0: No interrupt request signal is generated
  - 1: Interrupt request is generated, interrupt request status

Bit 6
- **SRIF0**: Interrupt request flag
  - 0: No interrupt request signal is generated
  - 1: Interrupt request is generated, interrupt request status

Bit 5
- **STIF0**: Interrupt request flag
  - 0: No interrupt request signal is generated
  - 1: Interrupt request is generated, interrupt request status

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Symbol: MK0H

<table>
<thead>
<tr>
<th>SREMK0</th>
<th>SREMK0</th>
<th>STMK0</th>
<th>DMAMK1</th>
<th>DMAMK0</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Bit 7

<table>
<thead>
<tr>
<th>SREMK0</th>
<th>Interrupt processing control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Enables interrupt processing.</td>
</tr>
<tr>
<td>1</td>
<td>Disables interrupt processing.</td>
</tr>
</tbody>
</table>

Bit 6

<table>
<thead>
<tr>
<th>SRMK0</th>
<th>Interrupt processing control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Enables interrupt processing.</td>
</tr>
<tr>
<td>1</td>
<td>Disables interrupt processing.</td>
</tr>
</tbody>
</table>

Bit 5

<table>
<thead>
<tr>
<th>STMK0</th>
<th>Interrupt processing control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Enables interrupt processing.</td>
</tr>
<tr>
<td>1</td>
<td>Disables interrupt processing.</td>
</tr>
</tbody>
</table>

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
Transition to communication wait state

- Serial channel start register 0 (SS0)
  Operation start

Symbol: SS0

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</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>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SS03</td>
<td>SS02</td>
</tr>
</tbody>
</table>
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | Note 1

Bits 3 to 0

<table>
<thead>
<tr>
<th>SS0n</th>
<th>Channel n operation start trigger</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Trigger operation is not performed</td>
</tr>
<tr>
<td>1</td>
<td><strong>SE0n is set to 1, and a communication wait state is entered.</strong></td>
</tr>
</tbody>
</table>

Note For UART reception, wait for 4 \( f_{CLK} \) clock cycles or more before setting SS0n to 1, after setting the RXE0n bit of the SCR0n register to 1.

Caution: For details on the register setup procedures, refer to RL78/I1A User's Manual: Hardware.
5.7.11 INTSR0 Interrupt Service Routine

Figure 5.16 shows the flowchart for the INTSR0 interrupt service routine.

![Flowchart for INTSR0 Interrupt Service Routine](image-url)

Figure 5.16 INTSR0 Interrupt Service Routine
5.7.12 UART0 Receive Data Classification Function

Figure 5.17 shows the flowchart for the UART0 receive data classification function.

![Flowchart for UART0 Receive Data Classification Function](image-url)
5.7.13 UART0 Data Transmission Function

Figure 5.18 shows the flowchart for the UART0 data transmission function.

![UART0 Data Transmission Function Flowchart](image-url)
5.7.14 UART0 Reception Error Interrupt Function

Figure 5.19 shows the flowchart for the UART0 reception error interrupt function.

Figure 5.19 UART0 Reception Error Interrupt Function
5.7.15 UART0 Reception Error Classification Function

Figure 5.20 shows the flowchart for the UART0 reception error classification function.

![Flowchart for UART0 Reception Error Classification Function](image-url)

Figure 5.20 UART0 Reception Error Classification Function
5.7.16 INTST0 Interrupt Service Routine

Figure 5.21 shows the flowchart for the INTST0 interrupt service routine.

![Flowchart for INTST0 Interrupt Service Routine](image-url)

**Figure 5.21** INTST0 Interrupt Service Routine
5.7.17 UART0 Transmission End Processing Function

Figure 5.22 shows the flowchart for the UART0 transmission end processing function.

```
r_uart0_callback_sendend()

Set transmission end flag

return
```

Figure 5.22 UART0 Transmission End Processing Function
6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

RL78/I1A User's Manual: Hardware (R01UH0210E)
RL78 Family User's Manual: Software (R01US0015E)

(The latest versions of the documents are available on the Renesas Electronics Website.)

Technical Updates/Technical Brochures

(The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

Renesas Electronics Website
- http://www.renesas.com/index.jsp

Inquiries
- http://www.renesas.com/contact/
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>July 25, 2016</td>
<td>—</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 document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

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

2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   ⎯ The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   ⎯ The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   ⎯ When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.
   ⎯ The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.
Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.

5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tool, personal electronic equipment, and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, and safety equipment etc.

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

7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Please evaluate the safety of the final products or systems manufactured by you.

8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under applicable domestic or foreign laws or regulations. You should use Renesas Electronics products or technology described in this document for any purposes relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document; Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.

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

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

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

(Note 2) "Renesas Electronics products" means any product developed or manufactured by or for Renesas Electronics.

SALES OFFICES

Renesas Electronics America Inc.
2501 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2204

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadistrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 Zhonglu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Tower, 595 Kanggu Road, Pudong District, Shanghai, P. R. China 200333
Tel: +86-21-2266-9991, Fax: +86-21-2266-9999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mong kok, Kowloon, Hong Kong
Tel: +852-2266-6668, Fax: +852-2888-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shih North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886-2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bundermee Road, Unit 80-03, Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-2030, Fax: +65-6213-3030

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-2056-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No:727C, 100 Fast Road, Hall II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F, 234 Tohehan-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3377, Fax: +82-2-558-5141

© 2016 Renesas Electronics Corporation. All rights reserved.
Colophon 5.0