[Notes]

R20TS0239EJ0101
Rev.1.01
Dec. 1, 2017

RX600 and RX200 Series
I^{2}C Bus Single Master Control Software Using RIIC Serial Interface

Outline

When using I^{2}C bus single master control software using RIIC serial interface of the RX600 and RX200 series MCUs, note the following points.

1. Using the SCL clock at low speed
2. Using a channel other than channel 0 of the RIIC

1. Using the SCL Clock at Low Speed

1.1 Applicable Products

➢ I^{2}C bus single master control software using RIIC serial interface of the RX600 and RX200 series MCUs Rev.1.02 and Rev.1.03

1.2 Applicable MCUs

RX62N, RX63N, RX63T, RX210, and RX21A groups

1.3 Details

In successful reception of the last data, the MCU receives 8-bit data, and then transmits 1-bit NACK. However, in the error case, NACK is not transmitted even after reception of the last data.

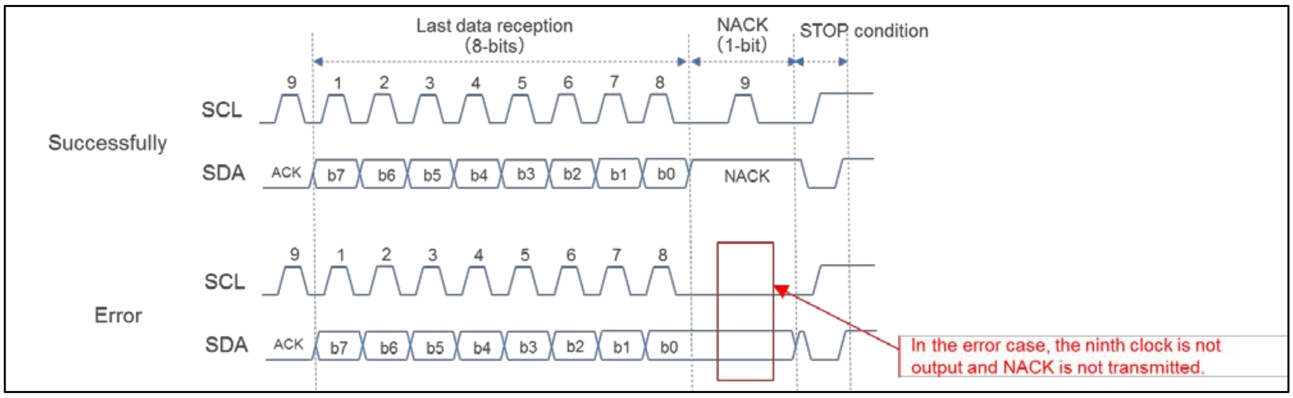
Last data reception (8-bits)
NACK (1-bit)
STOP condition
Successfully
SCL
SDA
ACK
b7
b6
b5
b4
b3
b2
b1
b0
NACK
Error
In the error case, the ninth clock is not output and NACK is not transmitted.

Figure 1 Last Data Reception Waveform in Successful and Error Cases

1.4 Conditions

The problem arises when all of the conditions listed below are met.

(1) The software control protocol is set to master reception or master composite.

(2) System clock (ICLK) > peripheral module clock (PCLK) > SCL clock is established(Note).

The problem is more likely to occur with larger differences between SCL clock and ICLK or PCLK.

Note: (Reference values) The problem occurs when ICLK = 100 MHz, PCLK = 50 MHz, SCL clock =< 100 KHz.

1.5 Workaround

Change the source code for the following function as shown in red.

r_iic_drv_receive_end_setting() function of r_iic_drv_sfr_rxXXX.c(Note)

Note: XXX indicates the name of the RX MCU you use. For example, r_iic_drv_sfr_rx63t.c.

[Workaround]

```
#define R_IIC_ICMR3_ACKWP_CLR   (uint8_t)(0xEF)
#define R_IIC_ICMR3_ACKBT_SET   (uint8_t)(0x08)
#define R_IIC_ICMR3_ACKBT_CLR   (uint8_t)(0xF7)

void r_iic_drv_receive_end_setting(r_iic_drv_info_t * pRIic_Info)
{
    /* Creates the register pointer for the specified RIIC channel. */
    volatile uint8_t * const    pcICMR3    = ICMR3_ADR(pRIic_Info->ChNo);

    /* Sets ICMR3.ACKBT bit. */
    *pcICMR3 |= R_IIC_ICMR3_ACKWP_SET;
    *pcICMR3 |= R_IIC_ICMR3_ACKBT_SET;
    *pcICMR3 &= R_IIC_ICMR3_ACKWP_CLR;

    /* Clears ICMR3.WAIT bit. */
    *pcICMR3 &= R_IIC_ICMR3_WAIT_CLR;
    if (R_IIC_ICMR3_WAIT_CLR != *pcICMR3)
    {
        nop();
    }
}
```

1.6 Schedule for Fixing the Problem

This problem will be fixed in the next version. The release date has not yet been decided.

2. Using a Channel Other Than Channel 0 of the RIIC

2.1 Applicable Products

- I^{2}C bus single master control software using RIIC serial interface of the RX600 and RX200 series MCUs Rev.1.02 and Rev.1.03

2.2 Applicable MCUs

RX62N, RX63N, RX63T, RX210, and RX21A groups

2.3 Details

In successful reception of data, the MCU receives 8-bit data, and then transmits 1-bit ACK (for the last data, NACK). In the error case, NACK is always transmitted after reception of data. Therefore, an undefined value is received from the slave device during the period from reception of the second byte of data to reception of the last data(Note).

Note: For data transmitted from the slave device after reception of NACK, see the specification of the slave device.

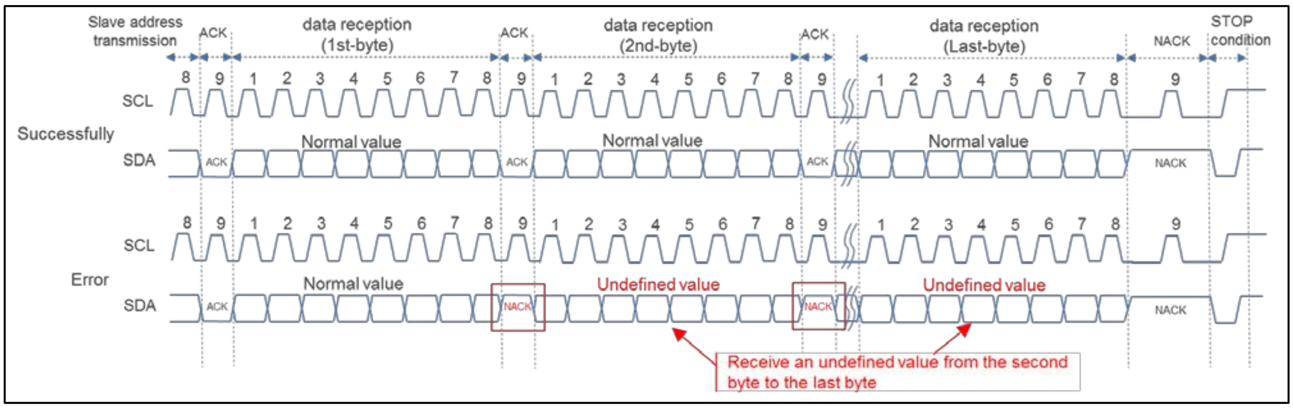
Slave address transmission
ACK
data reception (1st-byte)
ACK
data reception (2nd-byte)
ACK
data reception (Last-byte)
NACK
STOP condition
Successfully
SCL
SDA
Normal value
Error
Undefined value
NACK
Receive an undefined value from the second byte to the last byte

Figure 2 Communication Waveform When Using a Channel Other Than Channel 0 of the RIIC in Successful and Error Cases

2.4 Conditions

The problem arises when all of the conditions listed below are met.

(1) The software control protocol is set to master reception or master composite.

(2) A channel other than channel 0 of the RIIC is used.

(3) An interrupt other than the RIIC occurs in the period from when the ICDRR register is read to when the ICMR3.ACKBT bit is set to "1" in the last data reception processing.

(4) A stop condition has been issued during the interrupt processing of (3).

(5) Data reception is re-executed after the issue of a stop condition of (4).

2.5 Workaround

Change the source code for the following function as shown in red.

r_iic_drv_next_comm_setting() function of r_iic_drv_sfr_rxXXX.c(Note)

Note: XXX indicates the name of the RX MCU you use. For example, r_iic_drv_sfr_rx63t.c.

[Workaround]

```
void r_iic_drv_next_comm_setting(r_iic_drv_info_t * pRIic_Info)
{
   /* Creates the register pointer for the specified RIIC channel. */
   volatile uint8_t * const   pcICSR2  = ICSR2_ADR(pRIic_Info->ChNo);
   volatile uint8_t * const   pcICMR3  = ICMR3_ADR(pRIic_Info->ChNo);

   /* Checks the internal mode. */
   if ((R_IIC_MODE_READ == g_iic_InternalInfo[pRIic_Info->ChNo].Mode) ||
       (R_IIC_MODE_COMBINED == g_iic_InternalInfo[pRIic_Info->ChNo].Mode))
   {
        /* Clears ICMR3.RDRFS bit.*/
        /* Clears ICMR3.ACKBT bit. */
        *pcICMR3 |= R_IIC_ICMR3_ACKWP_SET;
        *pcICMR3 = R_IIC_ICMR3_INIT;
    }

(Omitted)
}
```

2.6 Schedule for Fixing the Problem

This problem will be fixed in the next version. The release date has not yet been decided.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 1, 2017	-	First edition issued
1.01	May 16, 2018	2	1.5 Workaround updated

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan
Renesas Electronics Corporation

■Inquiry
https://www.renesas.com/contact/