RENESAS Tool News

RENESAS TOOL NEWS on May 16, 2015: 150516/tn2

Note on Using the AP4 Coding Assistance Tool for RZ

When using the AP4 Coding Assistance Tool for RZ, take note of the problem on the following point that is described in this note.

- I2C Bus Interface (RIICa)
  Applicable products: RZ/T1 group

---

1. Product Concerned
   V1.00.00 of the AP4 Coding Assistance Tool for RZ

2. MPUs Involved
   RZ family: RZ/T1 group

3. Description
   When using the I2C bus interface (RIICa) for master reception, the interrupt following sending of the slave address cannot be accepted, since the transmission data empty interrupt (TXI) is in the interrupt-masked state.

4. Condition
   Using either channel 0 or channel 1 of the I2C bus interface for master reception.

5. Workaround
   Add processing to enable the transmission data empty interrupt (TXI) to either of the handlers for transfer errors and event generation interrupts listed below. These functions are in the r_cg_riic_user.c file.
   - void r_riic0_error_interrupt(void)
   - void r_riic1_error_interrupt(void)

   The following is an example of the required modification. The extra code should be added every time code is generated.

   Example of the required modification in the case of using channel 0 for

master reception

Before modification

```
------------------------------------------------------------
void r_riic0_error_interrupt(void)
{
.............
   else if (_IIC_MASTER_RECEIVE == g_riic0_mode_flag)
   {
      if ((_IIC_MASTER_SENDS_ADR_7_R == g_riic0_state) ||
         (_IIC_MASTER_SENDS_ADR_10A_W == g_riic0_state))
      {
         RIIC0.ICSR2.BIT.START = 0U;
         RIIC0.ICIER.BIT.STIE = 0U;
         RIIC0.ICIER.BIT.SPIE = 1U; /* Enable stop condition */
                                    /* detection to prepare for */
                                    /* the next receive */

         /* Enable the RXI0 interrupt */
         VIC.IEN3.LONG |= 0x04000000UL;
      }
      else if (_IIC_MASTER_RECEIVES_RESTART == g_riic0_state)
      {
         RIIC0.ICSR2.BIT.START = 0U;
         RIIC0.ICIER.BIT.STIE = 0U;
         g_riic0_state = _IIC_MASTER_SENDS_ADR_10A_R;
      }
      else if (_IIC_MASTER_RECEIVES_STOP == g_riic0_state)
      {
         RIIC0.ICMR3.BIT.RDRFS = 0U;
         RIIC0.ICMR3.BIT.ACKWP = 1U;
         RIIC0.ICMR3.BIT.ACKBT = 0U;
         RIIC0.ICSR2.BIT.NACKF = 0U;
         RIIC0.ICSR2.BIT.STOP = 0U;
         RIIC0.ICIER.BIT.SPIE = 0U;
         RIIC0.ICIER.BIT.STIE = 1U;  /* Enable start condition */
                                     /* detection to prepare for */
                                     /* the next receive */

         /* Clear RXI0 interrupt flag */
         VIC.PIC3.LONG = 0x04000000UL;
         /* Disable RXI0 interrupt */
         VIC.IEC3.LONG = 0x04000000UL;

         r_riic0_callback_receiveend();
```

```
      }
.............
}
```

After modification

```
void r_riic0_error_interrupt(void)
{
.............
   else if (_IIC_MASTER_RECEIVE == g_riic0_mode_flag)
   {
      if ((_IIC_MASTER_SENDS_ADR_7_R == g_riic0_state) ||
         (_IIC_MASTER_SENDS_ADR_10A_W == g_riic0_state))
      {
         RIIC0.ICSR2.BIT.START = 0U;
         RIIC0.ICIER.BIT.STIE = 0U;
         RIIC0.ICIER.BIT.SPIE = 1U; /* Enable stop condition */
                                    /* detection to prepare for */
                                    /* the next receive */

         /* Enable the TXI0 interrupt */
         VIC.IEN3.LONG |= 0x08000000UL;        <- added code

         /* Enable the RXI0 interrupt */
         VIC.IEN3.LONG |= 0x04000000UL;
      }
      else if (_IIC_MASTER_RECEIVES_RESTART == g_riic0_state)
      {
         RIIC0.ICSR2.BIT.START = 0U;
         RIIC0.ICIER.BIT.STIE = 0U;
         g_riic0_state = _IIC_MASTER_SENDS_ADR_10A_R;
      }
      else if (_IIC_MASTER_RECEIVES_STOP == g_riic0_state)
      {
         RIIC0.ICMR3.BIT.RDRFS = 0U;
         RIIC0.ICMR3.BIT.ACKWP = 1U;
         RIIC0.ICMR3.BIT.ACKBT = 0U;
         RIIC0.ICSR2.BIT.NACKF = 0U;
         RIIC0.ICSR2.BIT.STOP = 0U;
         RIIC0.ICIER.BIT.SPIE = 0U;
         RIIC0.ICIER.BIT.STIE = 1U;  /* Enable start condition */
                                     /* detection to prepare for */
                                     /* the next receive */
```

```
            /* Clear TXI0 interrupt flag */
            VIC.PIC3.LONG = 0x08000000UL;        <- added code
            /* Disable TXI0 interrupt */
            VIC.IEC3.LONG = 0x08000000UL;        <- added code

            /* Clear RXI0 interrupt flag */
            VIC.PIC3.LONG = 0x04000000UL;
            /* Disable RXI0 interrupt */
            VIC.IEC3.LONG = 0x04000000UL;

            r_riic0_callback_receiveend();
        }
    ..............
    }
    ----------------------------------------------------------------
```

6. Schedule for Fixing the Problem
   This problem will be fixed in a later version of the product.
