RX Family, M16C Family

Migrating from the M16C to the RX Asynchronous Serial Communications (UART)

R01AN1859EJ0100
Rev. 1.00
Apr. 1, 2014

Abstract

This document describes migrating from the serial I/O UART mode in the M16C Family to the SCI asynchronous mode in the RX Family.

Products

RX Family, M16C Family

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Differences in Terminology

Item	RX Family	M16C Family
Serial communication interface (SCI)	SCI	Serial I/O
Asynchronous serial communications mode	Asynchronous mode	UART mode (clock asynchronous serial I/O mode)
SCI operating clock (clock source)	Clock source	Count source
Peripheral function operating clock	Peripheral module clocks: PCLKA, PCLKB, PCLKC, PCLKD	Peripheral function clocks: f1, fOCO40M, fOCO-F, fOCO-S, fC32
Transmit buffer	TDR register	UiTB register (transmit buffer)
Transmit shift register	TSR register	UART transmit shift register
Receive buffer	RDR register	UiRB register
Transmit interrupt	TXI interrupt	UARTi transmit interrupt (transmit buffer empty)
Transmit complete interrupt (M16C) Transmit end interrupt (RX)	TEI interrupt	UARTi transmit interrupt (transmission completed)
Receive interrupt	RXI interrupt	UARTi receive interrupt
Function to select I/O of peripheral functions for pins	MPC *1	Function select register, input function select register *2

Note 1. Only available in the M32C Group and R32C Group.
Note 2. The MPC is not available in some groups.

Contents

1. Differences in Asynchronous Serial Communications 3
2. Peripheral Functions Used 4
3. Differences in Asynchronous Serial Communications 4
3.1 Transmit/Receive Timing 5
3.1.1 Differences in Transmitting 5
3.1.2 Differences in Receiving 7
3.2 Differences in Setting Procedures 9
3.3 Calculating the Bit Rate 12
4. Appendix 13
4.1 Points on Migrating from the M16C to the RX 13
4.1.1 Interrupts 13
4.1.2 I/O ports 13
4.1.3 Module Stop Function 13
4.2 I/O Register Macros 14
4.3 Intrinsic Functions 14
5. Reference Documents 15

1. Differences in Asynchronous Serial Communications

Table 1.1 lists the Differences in Asynchronous Serial Communications.

Table 1.1 Differences in Asynchronous Serial Communications

Item	RX (RX210)	M16C (M16C/65C)
Operation clock source	PCLKB	f1, fOCO40M, fOCO-F, fOCO-S, or fC32
Data length	7 bits or 8 bits	7 bits, 8 bits or 9 bits
Parity bit	Selectable from even, odd, or no parity	Selectable from even, odd, or no parity
Stop bits	1 bit or 2 bits	1 bit or 2 bits
Data format	LSB first or MSB first	LSB first or MSB first
Hardware flow control	Available (selectable)	Available (selectable)
Separate CTS/RTS pins	Not available	Available (UART0)
Interrupt sources	Transmit data empty (TXI) interrupt Transmit end (TEI) interrupt Receive data full (RXI) interrupt Receive error (ERI) interrupt	Transmit interrupt Receive interrupt
Error detection	Overrun error Framing error Parity error	Overrun error Framing error Parity error
Multi-processor function	Available	Not available
Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters	Not available
Data logic switch	Available	Available
TXD, RXD I/O polarity switch	Not available	Available

2. Peripheral Functions Used

Table 2.1 lists Peripheral Functions and Modes Used When Performing Asynchronous Communications.

Table 2.1 Peripheral Functions and Modes Used When Performing Asynchronous Communications

No.	Operating Example	RX		M16C		Reference
		Peripheral Function	Mode	Peripheral Function	Mode	
1	Asynchronous serial communications (transmit/receive operations)	SCI	Asynchronous mode	Serial I/O	UART mode	Chapter 3

3. Differences in Asynchronous Serial Communications

Table 3.1 lists the Conditions for Asynchronous Serial Communications.

Table 3.1 Conditions for Asynchronous Serial Communications

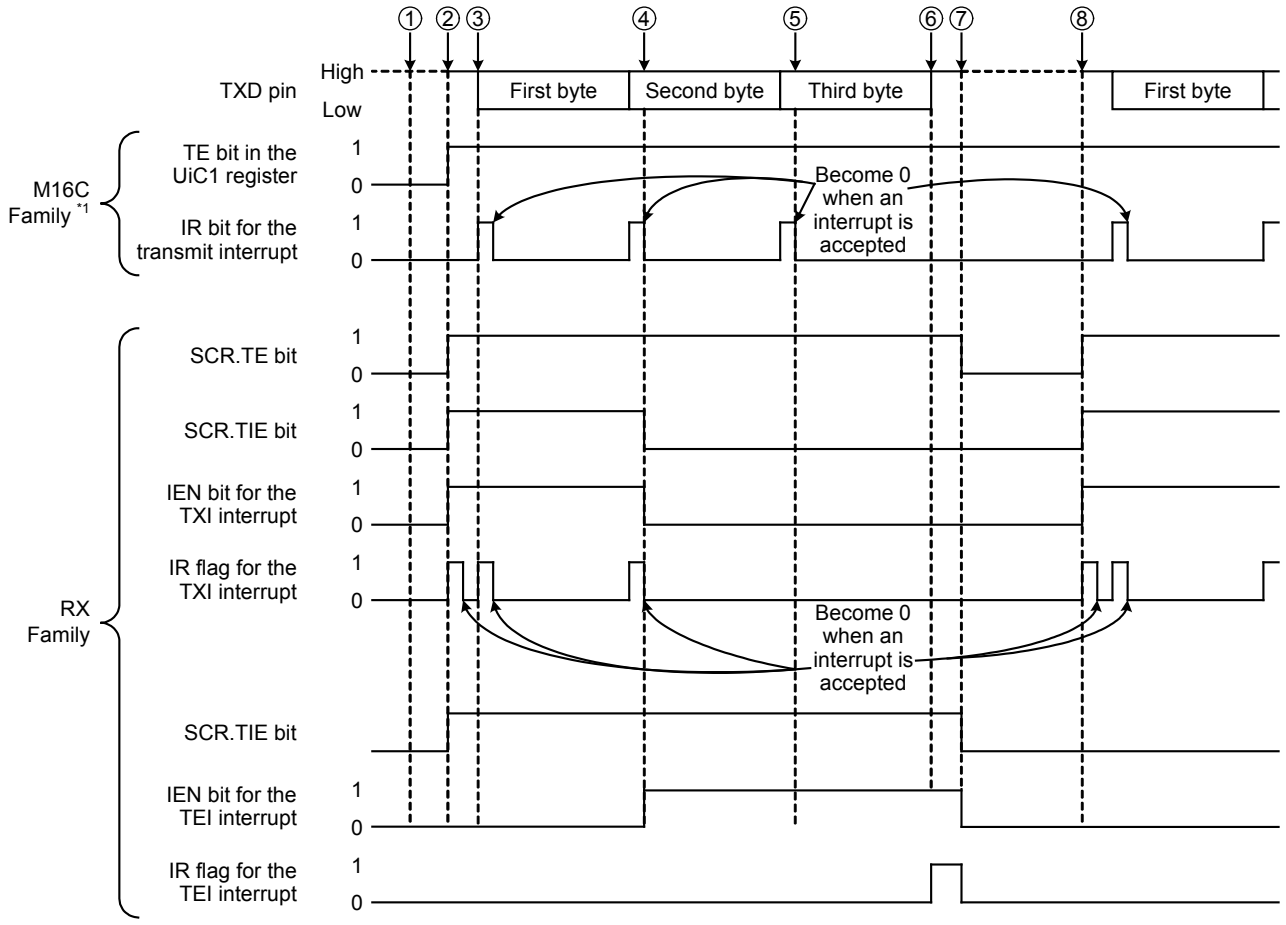
Item	Conditions for Transmission and Reception
Peripheral function operating clock	16 MHz
Transfer rate	9600 bps
Data length	8 bits
Stop bits	1 stop bit
Parity	None
Data format	LSB first
Hardware flow control	None
Channels used	M16C Family: UART0 RX Family: SCI0
Pin processing	Connect pin TXD and RXD to a pull-up resistor *1

Note 1. In the RX Family, when the SCR.TE bit is 0 (serial transmission is disabled), the TXD pin is in the Hi-Z state. When a pull-up resistor is not connected, while serial transmission is disabled, switch the pin function to the output state of general I/O ports.

3.1 Transmit/Receive Timing

3.1.1 Differences in Transmitting

Figure 3.1 shows the Differences in Timing When Transmitting 3 Bytes at a Time.



Note 1. In the M16C Family, the transmit interrupt source can only be selected from "UiTB register empty" or "transmission completed". This figure shows the timing when "UiTB register empty" is selected.

Figure 3.1 Differences in Timing When Transmitting 3 Bytes at a Time

Table 3.2 lists Differences in Operation at Various Timing When Transmitting 3 Bytes at a Time.

Table 3.2 Differences in Operation at Various Timing When Transmitting 3 Bytes at a Time

Timing	RX (RX210)	M16C (M16C/65C)
① Before transmission starts	The TXD pin is in the Hi-Z until the SCR.TE bit is set to 1 (serial transmission is enabled).	Check when the pin status selects the serial I/O mode.
② When transmission starts	Set the SCR.TE bit to 1, set the TIE bit to 1 (a TXI interrupt request is enabled), set the TEIE bit to 1 (a TEI interrupt request is enabled), and set the IEN bit for the TXI interrupt to 1 (TXI interrupt request enabled). When the SCR.TE bit is set to 1, the IR flag for the transmit interrupt (TXI interrupt) becomes 1, and the transmit interrupt is generated. Write the first byte of transmit data in the transmit interrupt handling.	The TE bit is set to 1 (transmission enabled). The transmit interrupt is not generated even if the TE bit is 1. The first byte of data is written in the main processing, etc.
③ When transmit data is transferred to the transmit shift register	The IR flag (IR bit) for the transmit interrupt becomes 1, and the transmit interrupt is generated. The second byte of data is written in the transmit interrupt handling.	
④ Transmit interrupt when writing the last data	Set the IEN bit for the TEI interrupt to 1 (TEI interrupt enabled), set the SCR.TIE bit to 0 (a TXI interrupt request is disabled), and set the IEN bit for the TXI interrupt to 0 (TXI interrupt disabled).	N/A
⑤ Transmit interrupt after writing the last data	N/A (The transmit interrupt is not generated.)	Interrupt handling is completed without transmit data being written.
⑥ After outputting the last data	The transmit complete interrupt is generated.	N/A
⑦ When transmission is complete	In the transmit end interrupt processing, set the SCR.TE bit to 0 (serial transmission is disabled), set the TEIE bit to 0 (a TEI interrupt request is disabled), and set the IEN bit for the TEI interrupt to 0 (TEI interrupt disabled) to disable transmission. When transmission is disabled, the IR flag for the transmit end interrupt becomes 0, and the TXD pin becomes Hi-Z.	
⑧ When transmission restarts	The same processing as in ② occurs.	The next data is written in the main processing, etc.

3.1.2 Differences in Receiving

Figure 3.2 shows the Differences in Timing When Receiving.

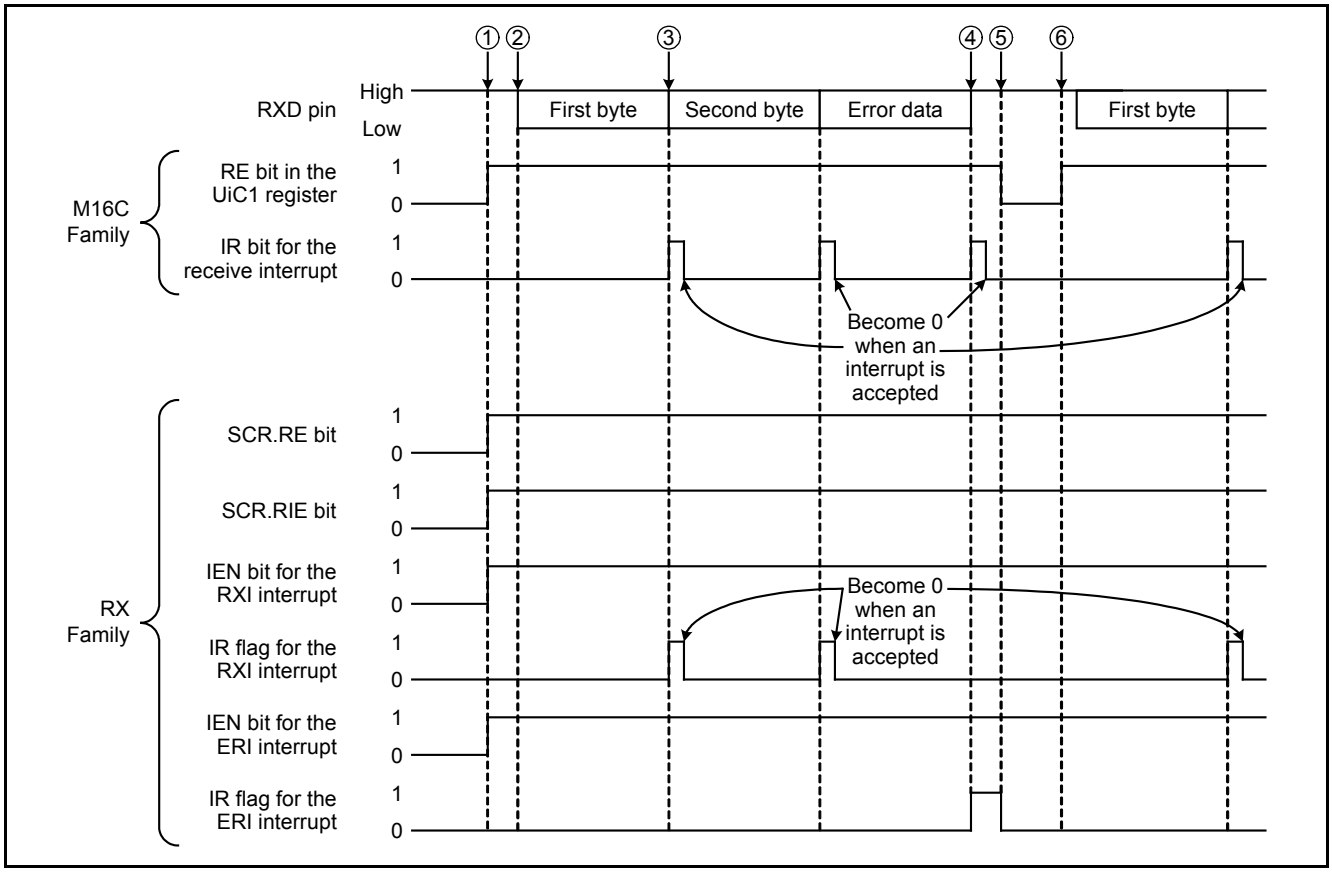


Figure 3.2 Differences in Timing When Receiving

Table 3.3 lists Differences in Operation at Various Timing When Receiving.

Table 3.3 Differences in Operation at Various Timing When Receiving

Timing	RX (RX210)	M16C (M16C/65C)
① When reception is enabled	Set the SCR.RE bit to 1 (serial reception is enabled), the RIE bit to 1 (RXI interrupt request is enabled), the IEN bit for the RXI interrupt to 1 (RXI interrupt request enabled), and the IEN bit for the ERI interrupt to 1 (ERI interrupt request enabled) to enable reception.	Set the RE bit to 1 (reception enabled) to enable reception.
② When reception starts	Receive operation starts when the start bit is input to the RXD pin.	
③ When reception is completed	When 1 byte of data is received, the received data is stored in the receive buffer, the IR flag (IR bit) for the receive interrupt (RXI interrupt) becomes 1, and the receive interrupt is generated. The value from the receive buffer is read in the receive interrupt handling.	
④ When a receive error occurs	The ERI interrupt is generated. Receive error processing is performed in the ERI interrupt handling.	A receive interrupt occurs. In the receive interrupt handling, read the error flag for the receive buffer register, and check to see if a receive error occurred.
⑤ Clear the receive error flag	After reading the error flags in the SSR register, set the flags to 0 to clear them. After all error flags have been cleared, the IR flag for the ERI interrupt becomes 0, and reception is enabled.	Set the RE bit to 0 (reception disabled), and set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
⑥ When reception is restarted		When bits SMD2 to SMD0 in the UiMR register are set to 101b (UART mode character length is 8 bits), and the RE bit is set to 1, reception is enabled.

3.2 Differences in Setting Procedures

The tables in this section list the steps for transmission and reception in asynchronous mode.

Table 3.4 Differences in the Initial Setting Procedures When Transmitting and Receiving Data

Step		RX (RX210)	M16C (M16C/65C)
1	Exiting the module-stop state *1	SYSTEM.PRCR.WORD = 0xA502; MSTP(SCI0) = 0; SYSTEM.PRCR.WORD = 0xA500;	N/A (no module-stop function)
2	Setting the I/O port functions *2	PORT2.PMR.BIT.B0 = 0; PORT2.PMR.BIT.B1 = 0; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.P20PFS.BYTE = 0x0A; MPC.P21PFS.BYTE = 0x0A; MPC.PWPR.BYTE = 0x80; PORT2.PMR.BIT.B0 = 1; PORT2.PMR.BIT.B1 = 1;	No processing *3

3	Setting the transmit/receive mode, etc.	SCI0.SCR.BIT.CKE = 0;
		SCI0.SMR.BYTE = 0x00;
		SCI0.SCMR.BYTE = 0xF2;
		SCI0.SEMR.BYTE = 0x00;
			uclksel0 = 0x00;
			pclk1 = 1;
			u0mr = 0x05;
			u0c0 = 0x10;
			ucon = 0x00
			u0c1 = 0x00;

4	Setting the bit rate *4	SCI0.BRR = 51;	u0brg = 103;
5	Setting the interrupt priority level	IPR(SCI0,) =0x01	s0tic = 0x01; s0ric = 0x01;
6	Clear the interrupt requests	IR(SCI0,RXI0) = 0; IR(SCI0,TXI0) = 0;	
7	Enable peripheral function interrupt requests *5	SCI0.SCR.BYTE |= 0xF4;	
8	Enable transmission/reception		u0c1 = 0x05;
9	Enable interrupt requests *5	IEN(SCI0,ERI0) = 1; /* Note 6 */ IEN(SCI0,RXI0) = 1; IEN(SCI0,TXI0) = 1;	N/A (no processing)
10	Enable maskable interrupts	setpsw_i();	asm("fset i");
11	Write the first byte of transmit data	N/A (no processing)	/* Write the first byte of data to the U0TB register */

Note 1. For details on the module stop function, refer to section 4.1.3 Module Stop Function.
Note 2. In the RX Family, pin settings for peripheral functions are configured in the MPC. Refer to section 4.1.2 I/O ports for details.
Note 3. In the M32C/80 Series and R32C/100 Series, select the pin function in the function select register.
Note 4. The bit rate calculation method differs between the RX Family and M16C Family. Refer to section 3.3 Calculating the Bit Rate for details.
Note 5. The method of enabling interrupt requests differs. Refer to 4.1.1 Interrupts for details.
Note 6. Specifications for the receive error interrupt differ according to the MCU used. Refer to the User's Manual: Hardware (UHM) for details.

Table 3.5 Differences in Transmit Interrupt Handling

Step		RX (RX210)	M16C (M16C/65C)

1	Write transmit data	/* Write transmit data to the SCR.TDR register */	if (Transmit data present?)
			{
			/* Write transmit data to U0TB register */
			}

2	Confirm last data was written	if (Last data written?) {	

3	Disable TXI interrupt (only after last data is written)	IEN(SCI0,TXI0) = 0;
		SCI0.SCR.BIT.TIE = 0;
		while (0 != SCI0.SCR.BIT.TIE)
		{
		}
			N/A (no processing) *1

4	Clear the interrupt request (only after the last data is written)	IR(SCI0,TXI0) = 0;	
5	Enable the transmit end interrupt (only after the last data is written)	IEN(SCI0,TEI0) = 1; }	

Note 1. There is no need to disable the transmit interrupt in the M16C Family.

Table 3.6 Differences in Transmit End Interrupt Handling

Step		RX (RX210)	M16C (M16C/65C)

1	Check the source of the generated interrupt request	do
		{
		while ((0 != SCI0.SCR.BIT.TEIE) &&
		(0 != SCI0.SSR.BIT.TEND))
		{
			N/A *1

2	Disable transmission	SCI0.SCR.BIT.TE = 0;
		while (0 != SCI0.SCR.BIT.TE)
		{
		}	

3	Disable the TEI interrupt	SCI0.SCR.BIT.TEIE = 0;
		while (0 != SCI0.SCR.BIT.TEIE)
		{
		}	

4	Read the IR flag	} }while(0 != IR(SCI0, TEI0));	

Note 1. When "UiTB register empty" is selected as the transmit interrupt source, an interrupt is not generated when transmission is complete.

Table 3.7 Differences in Receive Interrupt Handling

Step		RX (RX210)	M16C (M16C/65C)
1	Read the receive data	/* Read receive data from the SCI0.RDR register */	/* Read receive data from the U0RB register */
2	Determine the receive error	N/A (no processing) *1	if(Did a receive error occur?) {
3	Processing when a receive error occurs		/* Describe processing when an error occurs */

4	Clear the receive error (disable reception)		re_u0c1 = 0;
			u0mr = u0mr & 0xf8;
			}

Note 1. When an error occurs in the RX Family, error processing for a receive interrupt is not necessary since an ERI interrupt is generated.

Table 3.8 Differences in ERI Interrupt Handling

Step		RX (RX210)	M16C (M16C/65C)

1	Check the source of the generated interrupt request	do
		{
		while ((0 != SCI0.SCR.BIT.RIE) &&
		(0x00 != SCI0.SSR.BYTE & 0x38))
		{
			N/A *1

2	Processing when a receive error occurs	/* Describe processing when an error occurs */	
3	Dummy read the receive buffer	dummy = SCI0.RDR;	

4	Clear the receive error	dummy = SCI0.SSR.BYTE; /* Note 2 */
		SCI0.SSR.BYTE = 0xC0;
		while (0x00 != SCI0.SSR.BYTE & 0x38))
		{
		}	

5	Verify the IR flag	} }while(0 != IR(SCI0, ERI0));	

Note 1. In the M16C Family, an individual interrupt is not generated when a receive error occurs. Error determination and an appropriate processing are performed in the receive interrupt handling. Refer to Table 3.7 for details.

Note 2. Clear the parity error flag (PER flag), framing error flag (FER flag), or overrun error flag (ORER flag) after confirming the flag is read as 1.

3.3 Calculating the Bit Rate

There are differences in calculating the bit rate between the RX Family and M16C Family. Table 3.9 lists the Differences in Calculating the Bit Rate.

Table 3.9 Differences in Calculating the Bit Rate

Item	RX (RX210)	M16C (M16C/65C)
Calculating the bit rate using the internal clock	When the SEMR.ABCS bit is 0: Clock source ÷ 32(*N* + 1) *1 When the SEMR.ABCS bit is 1 Clock source ÷ 16(*N* + 1) *1 where the clock source is PCLK, PCLK/4, PCLK/16, or PCLK/64, and *N* is the BRR register setting value	Clock source ÷ 16(*m* + 1) where the clock source is 1SIO, f2SIO, f8SIO, or f32SIO, and *m* is the UiBRG setting value
Calculating the bit rate using the external clock	fEXT ÷ 16 where fEXT is input from the SCKi pin	fEXT ÷ 16(*m* + 1) where fEXT is input from the CLKi pin and *m* is the UiBRG setting value
Calculating the bit rate when the reference clock is generated by the TMR	Only when using SCI5, SCI6, and SCI12: The clock can be input from the TMR (refer to the User's Manual: Hardware for details).	N/A

Note 1. Based on the "Relationships between N Setting in BRR and Bit Rate B" in the UMH (SEMR.ABCS bit is 0)

$B = PCLK \div (64 \times 2^{2n-1} \times (N + 1))$

$= PCLK \div (32 \times 2^{2n} \times (N + 1))$

$= (PCLK \div 2^{2n}) \div (32 \times (N + 1))$

$=$ Clock source $\div (32 \times (N + 1))$

4. Appendix

4.1 Points on Migrating from the M16C to the RX

This chapter explains points on migrating from the M16C to the RX.

4.1.1 Interrupts

In the RX, if an interrupt request is received when all of the following conditions are met, the interrupt can be accepted.

- The I flag (PSW.I flag) is 1.
- Registers IER and IPR in the ICU are set to "interrupt enabled".
- The interrupt request is enabled by the interrupt request enable bit for the peripheral function.

Table 4.1 lists the Comparison of Conditions for Generating an Interrupt.

Table 4.1 Comparison of Conditions for Generating an Interrupt

Item	RX	M16C
I flag	When the I flag is set to 1 (enabled), the maskable interrupts are accepted.	
Interrupt request flag	When there is an interrupt request from a peripheral function, the interrupt request flag becomes 1 (interrupt requested).	
Interrupt priority level	Selected by setting the IPR[3:0] bits.	Selected by setting bits ILVL2 to ILVL0.
Interrupt request enable	Specified by setting the IER register.	N/A
Interrupt enable for peripheral functions	Interrupt enable or disable can be specified in each peripheral function.	N/A

For more information, refer to sections Interrupt Controller (ICU), CPU, and sections for other peripheral functions used in the UMH.

4.1.2 I/O ports

In the RX Family, the MPC must be configured in order to assign I/O signals of peripheral functions to pins. Before controlling the input and output pins in the RX Family, the following two items must be set.

- In the MPC.PFS register, select the peripheral functions that are assigned to the appropriate pins.
- In the PORTn.PMR register, select the function for the pin to be used as a general I/O port or I/O port for a peripheral function.

Table 4.2 lists Comparison of I/O Settings for Peripheral Function Pins.

Table 4.2 Comparison of I/O Settings for Peripheral Function Pins

Function	RX (RX210)	M16C (M16C/65C)
Select the pin function	A pin to which a peripheral I/O is assigned can be selected from multiple pins. The PFS register is used to select a peripheral function I/O that is assigned to a pin used.	These are not available in the M16C Family. *1 When a mode of a peripheral function is selected, an appropriate pin is assigned as an I/O pin for the peripheral function.
Switch between general I/O port and peripheral function	With the PMR register, the pin function can be selected as a general I/O port or a peripheral function.	

Note 1. The M32C Group and R32C Group have registers to assign a peripheral function to a pin.

For more information, refer to the Multi-Function Pin Controller (MPC) and I/O port sections in the UMH.

4.1.3 Module Stop Function

RX has the ability to stop each peripheral module individually. By placing unused peripheral modules in the module stop state, power consumption can be reduced. After a reset is released, all modules (with a few exceptions) are in the module stop state. Registers for modules in the module stop state cannot be read or write accessed.

For more information, refer to the Low Power Consumption section in the UMH.

4.2 I/O Register Macros

Macro definitions listed in Table 4.3 can be found in the RX I/O register definitions (iodefine.h).

The readability of programs can be achieved with these macro definitions.

Table 4.3 lists examples of Macros.

Table 4.3 Using Macros

Macro	Usage Example
IR("module name", "bit name")	**IR(MTU0,TGIA0)** = 0 ; The IR bit corresponding to MTU0.TGIA0 is cleared to 0 (interrupt request cleared).
IEN("module name", "bit name")	**IEN(MTU0, TGIA0)** = 1 ; The IEN bit corresponding to MTU0.TGIA0 is set to 1 (interrupt enabled).
IPR("module name", "bit name")	**IPR(MTU0, TGIA0)** = 0x02 ; The IPR bit corresponding to MTU0.TGIA0 is set to 2 (interrupt priority level 2).
MSTP("module name")	**MSTP(MTU)** = 0 ; The MTU0 module stop setting bit is cleared to 0 (module stop state is canceled).
VECT("module name", "bit name")	#pragma interrupt(Excep_MTU0_TGIA0 (vect=**VECT(MTU0, TGIA0)**) The interrupt function is declared for the corresponding MTU0.TGIA0 register.

4.3 Intrinsic Functions

The RX has intrinsic functions for setting control registers and special instructions. When using intrinsic functions, include machine.h.

Table 4.4 lists examples of Descriptions of Special Instructions and Control Register Settings.

Table 4.4 Descriptions of Special Instructions and Control Register Settings

Item	Description	
	RX	M16C
Set the I flag to 1	setpsw_i (); *1	asm("fset i");
Set the I flag to 0	clrpsw_i (); *1	asm("fclr i");
Expanded into the WAIT instruction	wait(); *1	asm("wait");
Expanded into the NOP instruction	nop(); *1	asm("nop");

Note 1. "machine.h" must be included.

5. Reference Documents

User's Manual: Hardware
RX210 Group User's Manual: Hardware Rev.1.50 (R01UH0037EJ)
M16C/65C Group User's Manual: Hardware Rev.1.10 (R01UH0093)
Refer to the corresponding UMH when using products other than the RX210 Group and M16C/65C Group.
The latest versions can be downloaded from the Renesas Electronics website.

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

User's Manual: Development Tools
RX Family C/C++ Compiler Package V.1.01 User's Manual Rev.1.00 (R20UT0570EJ)
M16C Series, R8C Family C Compiler Package V5.45
C Compiler User's Manual Rev.3.00
The latest versions can be downloaded from the Renesas Electronics website.

Website and Support

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

Inquiries
http://www.renesas.com/contact/

REVISION HISTORY	RX Family, M16C Family Application Note Migrating From the M16C to the RX Asynchronous Serial Communications (UART)

Rev.	Date	Description	
		Page	Summary
1.00	Apr. 1, 2014	—	First edition issued

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 should 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 tools; 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.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
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. Because the evaluation of microcomputer software alone is very difficult, 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 all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
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 any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or 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. 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 product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

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

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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
Arcadiastrasse 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 ZhiChunLu 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 Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

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

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

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: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2014 Renesas Electronics Corporation. All rights reserved.
Colophon 4.0