Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

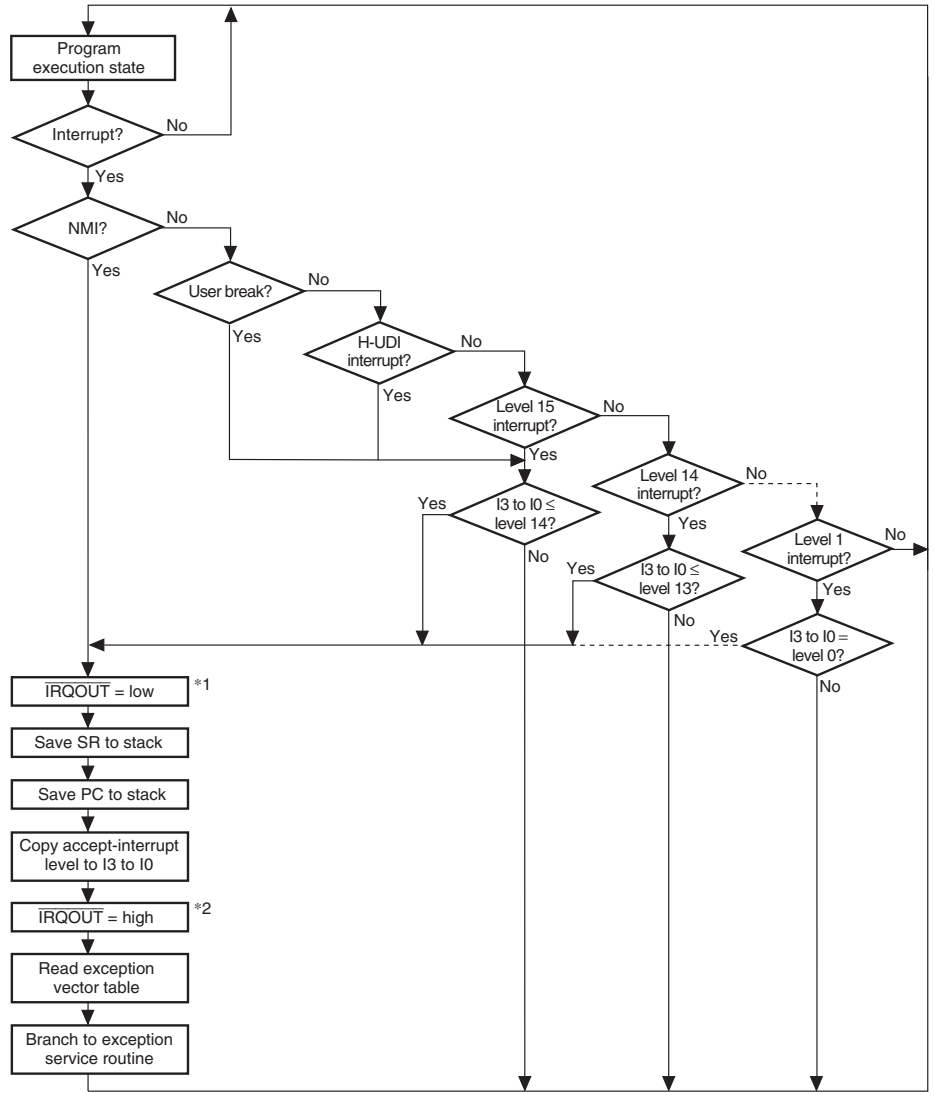
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	SH-2
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	EBI/EMI, I²C, SCI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	98
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/hd6417145fw50v

Section 9 Bus State Controller (BSC)

Section 10 Direct Memory Access Controller (DMAC)

the IRQ status register (ISR). Interrupts held pending due to edge detection are cleared by a power-on reset or a manual reset.



Notes: I3 to I0 are Interrupt mask bits of status register (SR) in the CPU

1. $\overline{\text{IRQOUT}}$ is the same signal as interrupt request signal to the CPU (see figure 6.1). Therefore, $\overline{\text{IRQOUT}}$ is output when the request priority level is higher than the level in bits I3–I0 of SR.
2. When the accepted interrupt is sensed by edge, a high level is output from the IRQOUT pin at the moment when the CPU starts interrupt exception processing instead of instruction execution (namely, before saving SR to stack). However, if the interrupt controller accepts an interrupt with a higher priority than the interrupt just to be accepted and has output an interrupt request to the CPU, the $\overline{\text{IRQOUT}}$ pin holds low level.

Figure 6.3 Interrupt Sequence Flowchart

8.3 Operation

8.3.1 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTCSR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source interrupt flag or corresponding DTER bit is cleared. The activation source flag, in the case of RXI_2, for example, is the RDRF flag of SCI2.

When a DTC is activated by an interrupt, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

Figure 8.2 shows a block diagram of activation source control. For details see section 6, Interrupt Controller (INTC).

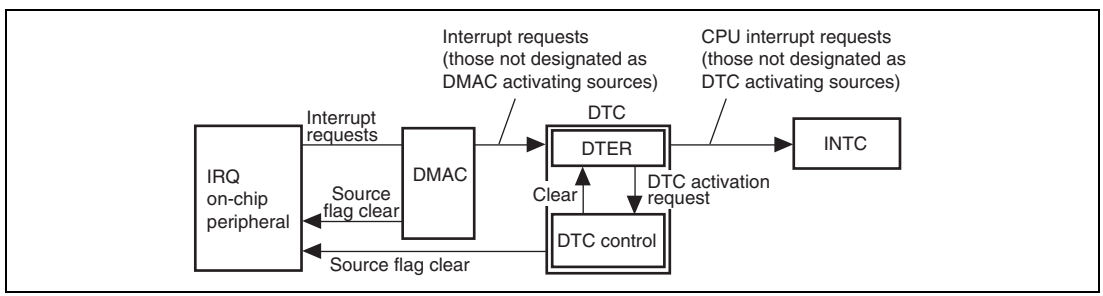


Figure 8.2 Activating Source Control Block Diagram

8.3.2 Location of Register Information and DTC Vector Table

Figure 8.3 shows the allocation of register information in memory space. The register information start addresses are designated by DTBR for the upper 16 bits, and the DTC vector table for the lower 16 bits.

The allocation in order from the register information start address in normal mode is DTMR, DTCRA, 4 bytes empty (no effect on DTC operation), DTSAR, then DTDAR. In repeat mode it is DTMR, DTCRA, DTIAR, DTSAR, and DTDAR. In block transfer mode, it is DTMR, DTCRA, 2 bytes empty (no effect on DTC operation), DTCRB, DTSAR, then DTDAR.

Fundamentally, certain RAM areas are designated for addresses storing register information.

Table 9.2 Address Map

- On-chip ROM enabled mode

Address	Space	Memory	Size	Bus Width
H'00000000 to H'0003FFFF	On-chip ROM	On-chip ROM	256 kbytes	32 bits
H'00040000 to H'001FFFFF	Reserved	Reserved		
H'00200000 to H'003FFFFF	CS0 space	External space	2 Mbytes	8/16/32 bits*1
H'00400000 to H'007FFFFF	CS1 space	External space	4 Mbytes	8/16/32 bits*1
H'00800000 to H'00BFFFFF	CS2 space	External space	4 Mbytes	8/16/32 bits*1
H'00C00000 to H'00FFFFFF	CS3 space	External space	4 Mbytes	8/16/32 bits*1
H'01000000 to H'011FFFFF	Reserved	Reserved		
H'01200000 to H'013FFFFF	CS4 space*3	External space	2 Mbytes	8/16/32 bits*1
H'01400000 to H'017FFFFF	CS5 space*3	External space	4 Mbytes	8/16/32 bits*1
H'01800000 to H'01BFFFFF	CS6 space*3	External space	4 Mbytes	8/16/32 bits*1
H'01C00000 to H'01FFFFFF	CS7 space*3	External space	4 Mbytes	8/16/32 bits*1
H'02000000 to H'FFFF7FFF	Reserved	Reserved		
H'FFFF8000 to H'FFFFBFFF	On-chip peripheral module	On-chip peripheral module	16 kbytes	8/16 bits
H'FFFFC000 to H'FFFFDFFF	Reserved	Reserved		
H'FFFFE000 to H'FFFFFFFF	On-chip RAM	On-chip RAM	8 kbytes	32 bits

Dual Address Mode:

Dual address mode is used for access of both the transfer source and destination by address. Transfer source and destination can be accessed either internally or externally. Dual address mode is subdivided into two other modes: direct address transfer mode and indirect address transfer mode.

- Direct Address Transfer Mode

  Data is read from the transfer source during the data read cycle, and written to the transfer destination during the write cycle, so transfer is conducted in two bus cycles. At this time, the transfer data is temporarily stored in the DMAC. With the kind of external memory transfer shown in figure 10.6, data is read from one of the memories by the DMAC during a read cycle, then written to the other external memory during the subsequent write cycle. Figure 10.7 shows the timing for this operation.

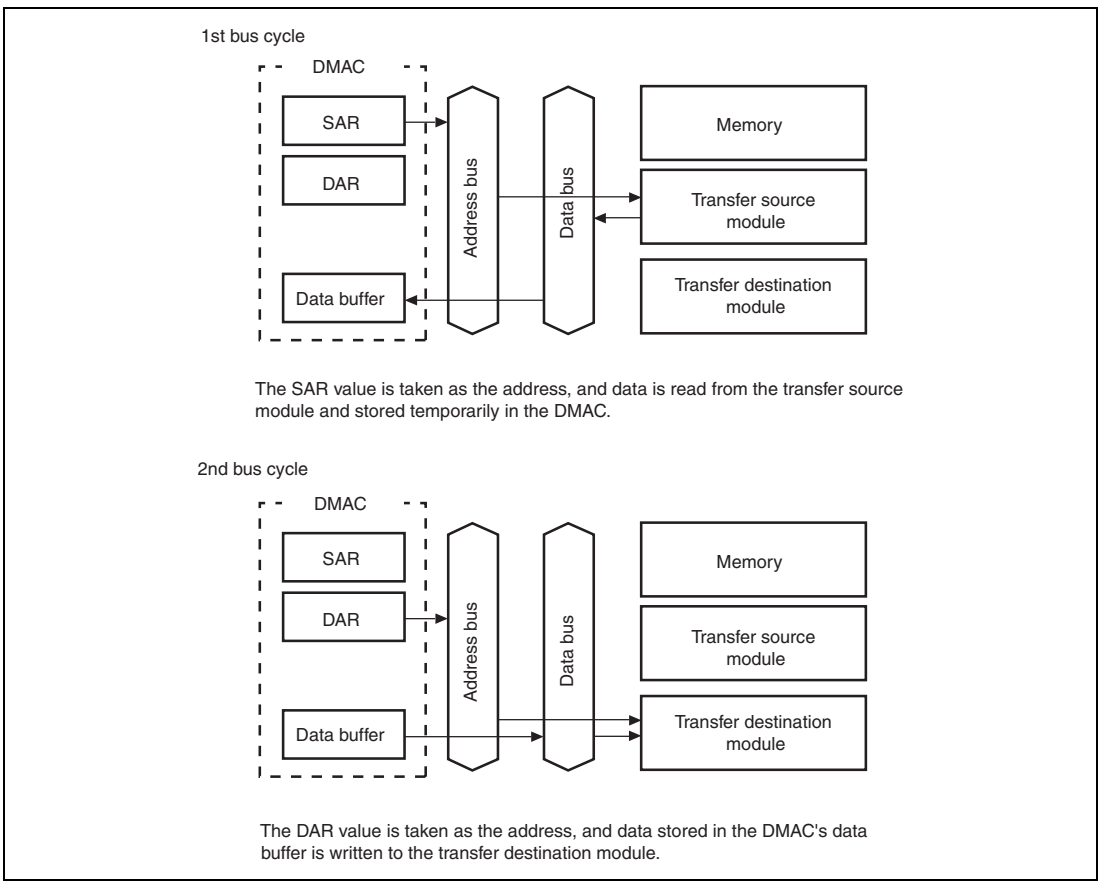


Figure 10.6 Direct Address Operation during Dual Address Mode

11.3.5 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU has five TSR registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1, 2, 3, and 4. In channel 0, bit 7 is reserved. It is always read as 1 and the write value should always be 1. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. [Setting condition] • When the TCNT value underflows (changes from H'0000 to H'FFFF) [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Setting condition] • When the TCNT value overflows (changes from H'FFFF to H'0000) In channel 4, when the TCNT_4 value underflows (changes from H'0001 to H'0000) in complementary PWM mode, this flag is also set. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1 In cannel 4, when DTC is activated by TCIV interrupt and the DISEL bit of DTMR in DTC is 0, this flag is also cleared.

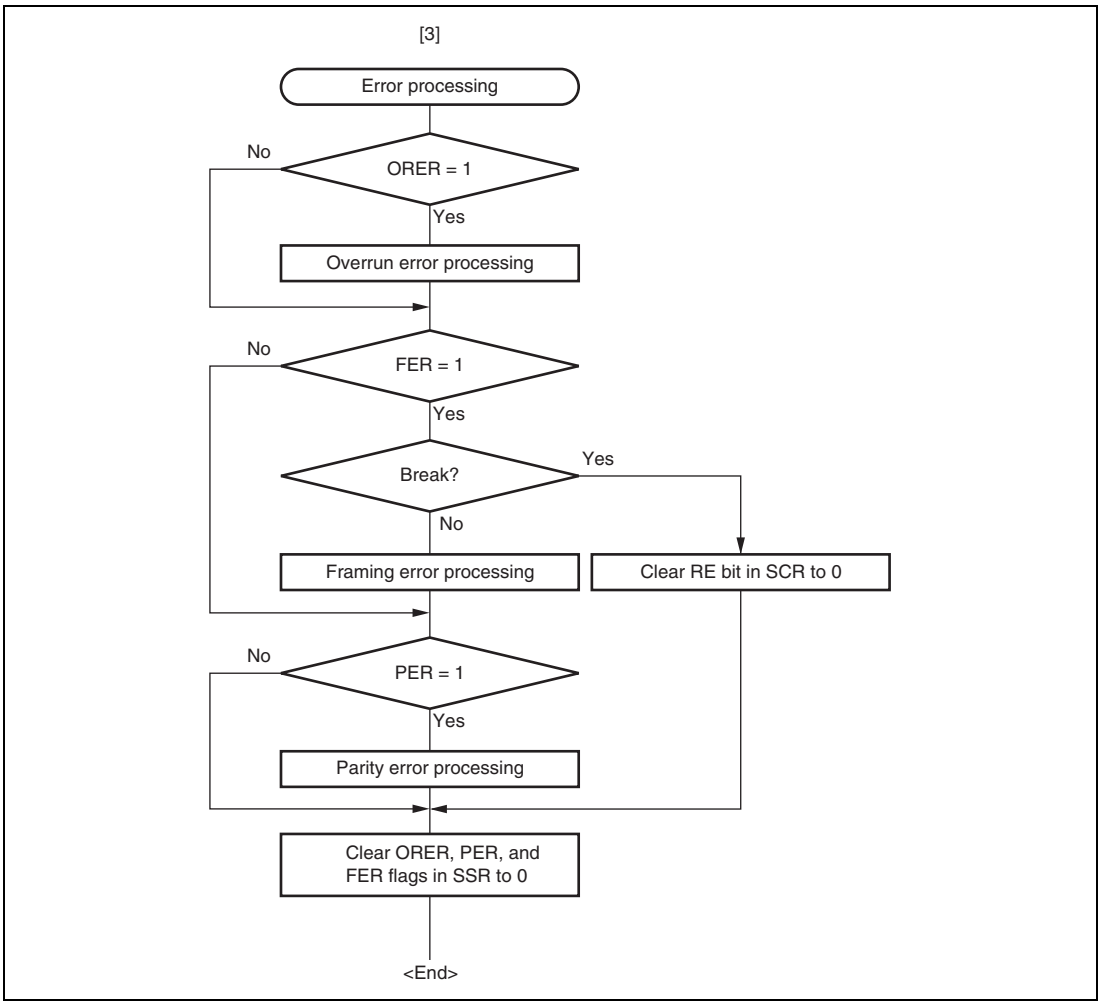


Figure 13.9 Sample Serial Reception Data Flowchart (2)

Table 13.12 Interrupt Sources in Serial Communication Interface Mode

Channel	Name	Interrupt Source	Interrupt Flag	DMAC or DTC Activation
0	ERI_0	Receive error	ORER, FER, PER	Not possible
	RXI_0	Receive data full	RDRF	Possible
	TXI_0	Transmit data empty	TDRE	Possible
	TEI_0	Transmission end	TEND	Not possible
1	ERI_1	Receive error	ORER, FER, PER	Not possible
	RXI_1	Receive data full	RDRF	Possible
	TXI_1	Transmit data empty	TDRE	Possible
	TEI_1	Transmission end	TEND	Not possible
2	ERI_2	Receive error	ORER, FER, PER	Not possible
	RXI_2	Receive data full	RDRF	Possible
	TXI_2	Transmit data empty	TDRE	Possible
	TEI_2	Transmission end	TEND	Not possible
3	ERI_3	Receive error	ORER, FER, PER	Not possible
	RXI_3	Receive data full	RDRF	Possible
	TXI_3	Transmit data empty	TDRE	Possible
	TEI_3	Transmission end	TEND	Not possible

The following description gives the procedures for and operations of receiving data in one byte units by fixing SCL low for every data reception using the HNDS bit function.

1. Clear the TRS bit in ICCR to 0 to change from the transmit mode to the receive mode.
   Clear the ACKB bit in ICSR to 0 (setting of the acknowledge data).
   Set the HNDS bit in SCRX to 1.
   Clear the IRIC flag to 0 to confirm that reception has been completed.
   When the first frame is the final receive data, perform end processing in step 6 and subsequent steps.
2. When ICDR is read (a dummy read operation), the receiving of data starts; the receive clock is output in synchronization with the internal clock, and the first datum is then received. (Data of the SDA pin is stored in ICDRS in synchronization with the rising edge of receive clock.)
3. The master device sets SDA to low on the 9th cycle of the receive clock and returns the acknowledge bit. The receive data is transferred from ICDRS to ICDRR at the rising edge of the 9th cycle of the receive clock, and the ICDRF, IRIC, and IRTR flags are set to 1. When the IEIC bit in ICCR has been set to 1, an interrupt request is generated for the CPU. The master devise fixes SCL low between at the falling edge of 9th cycle of the receive clock and read of ICDR data.
4. To identify the next interrupt, the IRIC flag is cleared to 0.
   When the next frame is the final receive data, perform end processing in step 6 and subsequent steps.
5. Read the receive data of ICDR. This clears the ICDRF flag to 0, and the master devise outputs the receive clock continuously for the reception of the next data.

   Data can be received by repeating the steps 3 to 5.

6. Set the ACKB bit to 1 (setting of acknowledge data for the final reception).
7. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs the receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are set to 1 at the rising edge of the 9th cycle of receive clock.
9. Clear the IRIC flag to 0.
10. Read ICDR receives data after setting the TRS bit to 1. This clears the ICDRF flag to 0.
11. Write 0 to BBSY and SCP in ICCR to generate the stop condition.
    This changes SDA from low to high when SCL is high, and generates the stop condition.

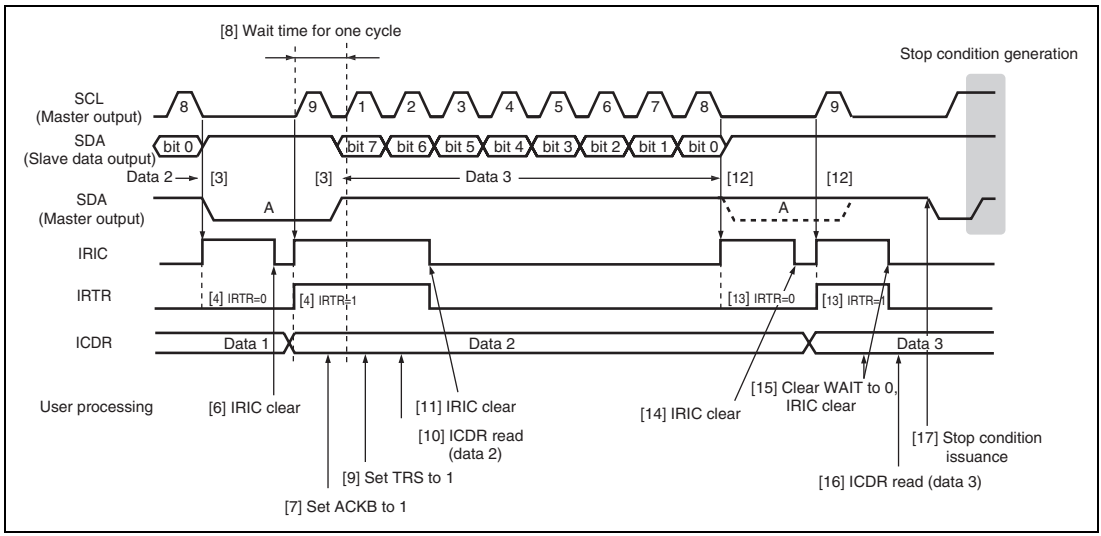


Figure 14.16 An Example of the Stop Condition Issuance Timing in Master Receive Mode (MLS = ACKB = 0, WAIT = 1)

14.4.5 Operations in Slave Reception

In the slave receive mode of the I²C bus format, the master device transmits the transmit clock and data, and the slave device returns acknowledgements of reception. The slave device compares the address of the slave address and the slave address of the first frame issued after the start condition issuance by the master device. If the addresses match, the slave device operates as a slave device specified by the master device.

Reception with HNDS Function (HNDS = 1):

Figure 14.17 is a flowchart that gives an example of operations in slave receive mode (HNDS = 1).

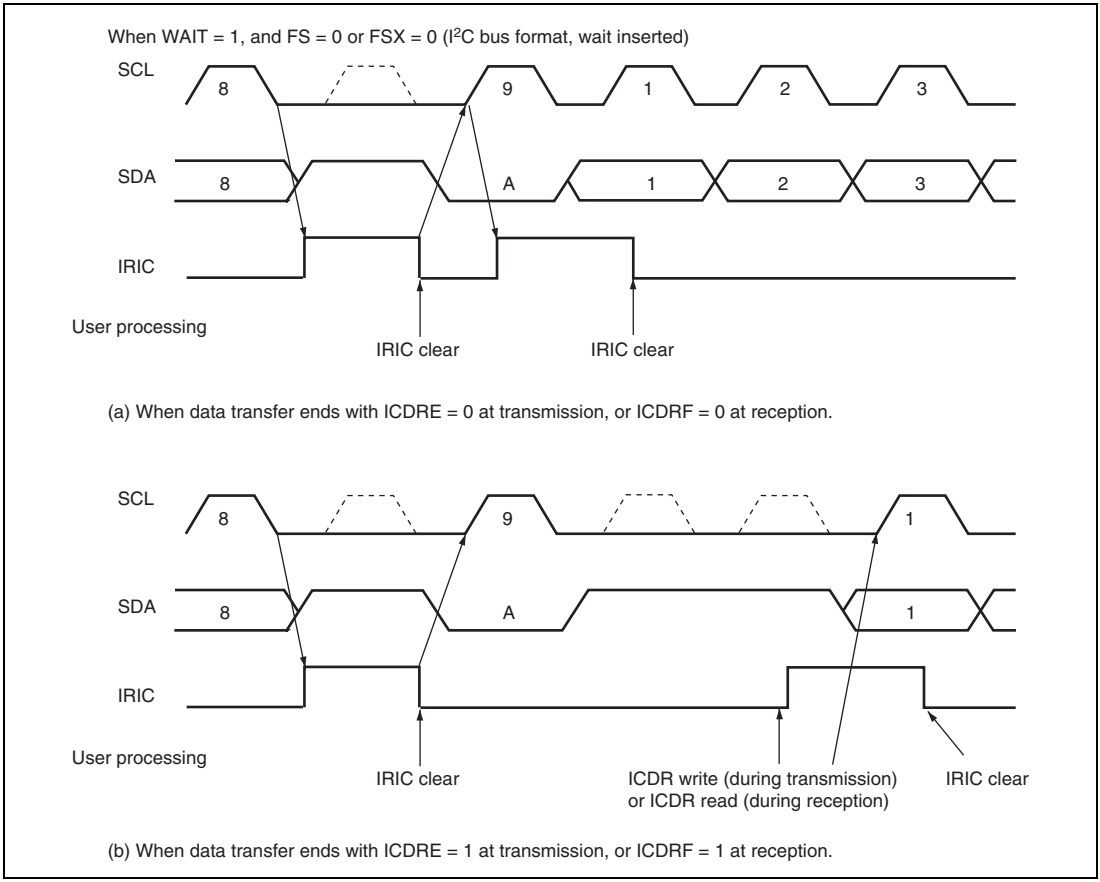


Figure 14.26 IRIC Flag Set Timing and the Control of SCL (2)

14. Points for cautions when reading ICDR in transmit mode and writing to ICDR in receive mode

When ICDR is read in transmit mode (TRS = 1) or ICDR is written to in receive mode (TRS = 0), the SCL pin may not be held low in some cases after transmit/receive operation has been completed, thus inconveniently allowing clock pulses to be output on the SCL bus line before ICDR is accessed correctly. To access ICDR correctly, read the ICDR after setting to receive mode or write to the ICDR after setting to transmit mode.

15. Points for cautions on ACKE and TRS bits in slave mode

In the I²C bus interface, if 1 is received as the acknowledge bit value (ACKB = 1) in transmit mode (TRS = 1) and then the address is received in slave mode without performing appropriate processing, interrupt handling may start at the rising edge of the 9th cycle of the clock even when the address does not match.

Similarly, in slave mode, if the start condition or address is transmitted from the master device in transmit mode (TRS = 1), the IRIC flag may be set as a result of the ICDRE flag set or receiving 1 as the acknowledge bit value (ACKB = 1), thus causing an interrupt source to occur even when the address does not match.

To use the I²C bus interface module in slave mode, be sure to follow the procedures below.

— When 1 is received as the acknowledge bit value for the final transmit data at the end of a series of transmit operations, clear the ACKE bit in ICCR once to initialize the ACKB bit to 0.

— Set to receive mode (TRS = 0) before the next start condition is input in slave mode. Complete transmit operation by the procedure shown in figure 14.23, in order to switch from slave transmit mode to slave receive mode.

14.5.1 Module Stop Mode Setting

IIC is enabled or disabled using the module stop control register. IIC is disabled with the initial value. Cancelling module stop mode allows the register to be accessed. For details, see section 24, Power-Down Modes.

Pin No.	Pin Name			
	On-Chip ROM Enabled (MCU Mode 2)		Single Chip Mode (MCU Mode 3)	
SH7144	Initial Function	PFC Selected Function Possibilities	Initial Function	PFC Selected Function Possibilities
99	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7
102	PE5	PE5/TIOC1B/TXD3	PE5	PE5/TIOC1B/TXD3
104	PE6	PE6/TIOC2A/SCK3	PE6	PE6/TIOC2A/SCK3
105	PE7	PE7/TIOC2B/RXD2	PE7	PE7/TIOC2B/RXD2
106	PE8	PE8/TIOC3A/ SCK2	PE8	PE8/TIOC3A/SCK2
107	PE9	PE9/TIOC3B/ SCK3	PE9	PE9/TIOC3B/SCK3
108	PE10	PE10/TIOC3C/TXD2	PE10	PE10/TIOC3C/TXD2
110	PE11	PE11/TIOC3D/RXD3	PE11	PE11/TIOC3D/RXD3
111	PE12	PE12/TIOC4A/TXD3	PE12	PE12/TIOC4A/TXD3
112	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$

Notes: 1. F-ZTAT version only
2. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, and TCK when using the E10A (in DBGMD = high).
3. Masked ROM version and ROM less version only

19.6.1 Boot Mode

Table 19.4 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.
2. The SCI1 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI1 bit rate to match that of the host. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary.
4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the completion of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 19.5.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'FFFFE800 to H'FFFFFFFF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
6. Before branching to the programming control program, the chip terminates transfer operations by SCI1 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of write data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, as the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, waiting at least 25 states, and then setting the mode (MD) pins. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the MD pin input levels in boot mode.
9. All interrupts are disabled during programming or erasing of the flash memory.

Figure 19.8 shows a sample procedure for flash memory block area overlapping.

1. The RAM area to be overlapped is fixed at a 4-kbyte area in the range H'FFFFE000 to H'FFFFEFFF.
2. The flash memory area to be overlapped is selected by RAMER from a 4-kbyte area of the EB0 to EB7 blocks.
3. The overlapped RAM area can be accessed from both the flash memory addresses and RAM addresses.
4. When the RAMS bit in RAMER is set to 1, program/erase protection is enabled for all flash memory blocks (emulation protection). In this state, setting the P or E bit in FLMCR1 to 1 does not cause a transition to program mode or erase mode.
5. A RAM area cannot be erased by execution of software in accordance with the erase algorithm.
6. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlapped RAM.

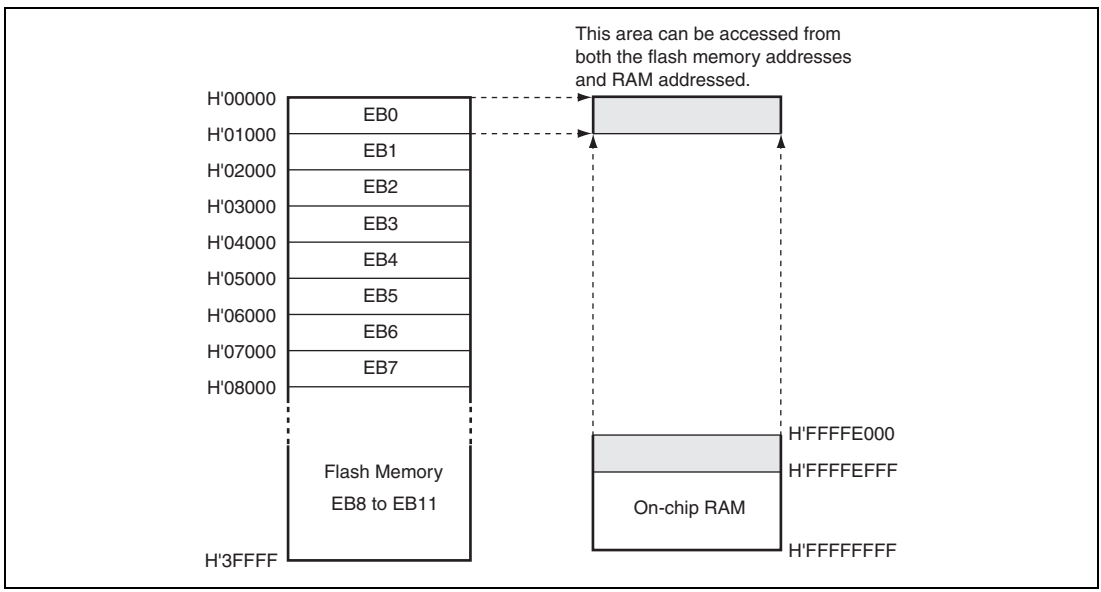


Figure 19.8 Example of RAM Overlap Operation (RAM[2:0] = B'000)

Register abbreviation	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Module
—	—	—	—	—	—	—	—	—	—
BCR1	—	—	MTURWE	—	—	—	—	—	BSC
	A3LG	A2LG	A1LG	A0LG	A3SZ	A2SZ	A1SZ	A0SZ	
BCR2	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00	
	CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0	
WCR1	W33	W32	W31	W30	W23	W22	W21	W20	
	W13	W12	W11	W10	W03	W02	W01	W00	
WCR2	—	—	—	—	—	—	—	—	
	—	—	—	—	DSW3	DSW2	DSW1	DSW0	
—	—	—	—	—	—	—	—	—	—
RAMER	—	—	—	—	—	—	—	—	FLASH
	—	—	—	—	RAMS	RAM2	RAM1	RAM0	(Only in F-ZTAT version)
—	—	—	—	—	—	—	—	—	—
DMAOR	—	—	—	—	—	—	PR1	PR0	DMAC
	—	—	—	—	—	AE	NMIF	DME	for all channels
SAR_0									DMAC
									(Channel 0)
DAR_0									
DMATCR_0	—	—	—	—	—	—	—	—	
CHCR_0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	

Pin Function		Pin State								
		Reset State					Power-Down State			
		Power-On								
Type	Pin Name	Expansion without ROM 8 bits	Expansion without ROM 16 bits	Expansion with ROM	Single-chip	Manual	Software Standby	Sleep	Bus Master-ship Release	Software Standby in Bus Master-ship Release
A/D converter	AN0 to AN7	Z	Z	Z	Z	I	Z	I	I	Z
	$\overline{\text{ADTRG}}$	Z	Z	Z	Z	I	Z	I	I	Z
I^{2}C	SCL0	Z	Z	Z	Z	I/O	Z	I/O	I/O	Z
	SDA0	Z	Z	Z	Z	I/O	Z	I/O	I/O	Z
I/O port	PA0 to PA15	Z	Z	Z	Z	I/O	K*[1]	I/O	I/O	K*[1]
	PB0 to PB9									
	PC0 to PC15									
	PD0 to PD15									
	PE0 to PE8, PE10									
	PE9, PE11 to PE15	Z	Z	Z	Z	I/O	Z (MZIZE in PPCR=0) K*[1] (MZIZE in PPCR=1)	I/O	I/O	Z (MZIZE in PPCR=0) K*[1] (MZIZE in PPCR=1)
	PF0 to PF7	Z	Z	Z	Z	I	Z	I	I	Z

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High-impedance

K: Input pins become high-impedance, and output pins retain their state.

Table A.2 Pin States (SH7145)

Pin Function		Pin State								
		Reset State					Power-Down State			
		Power-On								
Type	Pin Name	Expansion without ROM 8 bits	Expansion without ROM 16 bits	Expansion with ROM	Single-chip	Manual	Software Standby	Sleep	Bus Master-ship Release	Software Standby in Bus Master-ship Release
Clock	CK	O	O	O	Z	O	H*1	O	O	O
	XTAL	O	O	O	O	O	L	O	O	L
	EXTAL	I	I	I	I	I	I	I	I	I
	PLLCAP	I	I	I	I	I	I	I	I	I
System control	$\overline{\text{RES}}$	I	I	I	I	I	I	I	I	I
	$\overline{\text{MRES}}$	Z	Z	Z	Z	I	Z	I	I	Z
	$\overline{\text{WDTOVF}}$	O*2	O*2	O*2	O*2	O	O	O	O	O
	$\overline{\text{BREQ}}$	Z	Z	Z	Z	I	Z	I	I	I
	$\overline{\text{BACK}}$	Z	Z	Z	Z	O	Z	O	L	L
Operation mode control	MD0 to MD3	I	I	I	I	I	I	I	I	I
	DBGMD	I	I	I	I	I	I	I	I	I
	FWP	I	I	I	I	I	I	I	I	I
Interrupt	NMI	I	I	I	I	I	I	I	I	I
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$	Z	Z	Z	Z	I	Z*3	I	I	Z*3
	$\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$	Z	Z	Z	Z	I	Z*4	I	I	Z*4
	$\overline{\text{IRQOUT}}$ (PD30)	Z	Z	Z	Z	O	H*1	O	O	H*1
	$\overline{\text{IRQOUT}}$ (PE15)	Z	Z	Z	Z	O	Z*6 (MZIZE in PPCR=0) H*1 (MZIZE in PPCR=1)	O	O	Z*6 (MZIZE in PPCR=0) H*1 (MZIZE in PPCR=1)
Address bus	A0 to A17	O	O	Z	Z	O	Z	O	Z	Z
	A18 to A21	Z	Z	Z	Z	O	Z	O	Z	Z
Data bus	D0 to D31	Z	Z	Z	Z	I/O	Z	I/O	Z	Z

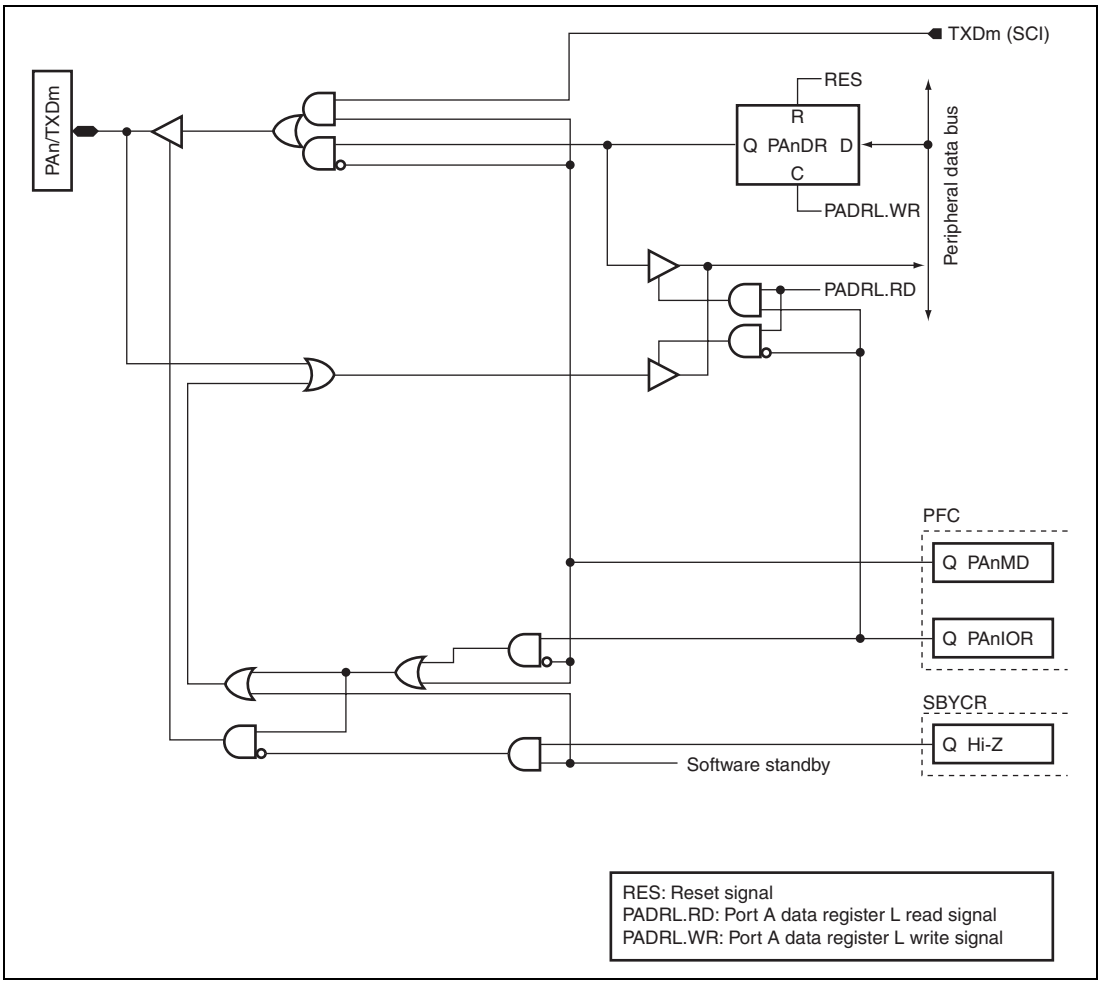


Figure D.2 PAn/TXDm

	Symbol in Figure D.2		Available Products			
			SH7144		SH7145	
Pins	PAn	TXDm	F-ZTAT version	Masked ROM version/ ROM less version	F-ZTAT version	Masked ROM version/ ROM less version
PA1/TXD0	PA1	TXD0 (SCI)	√	√	√	√
PA4/TXD1	PA4	TXD1 (SCI)	√	√	√	√

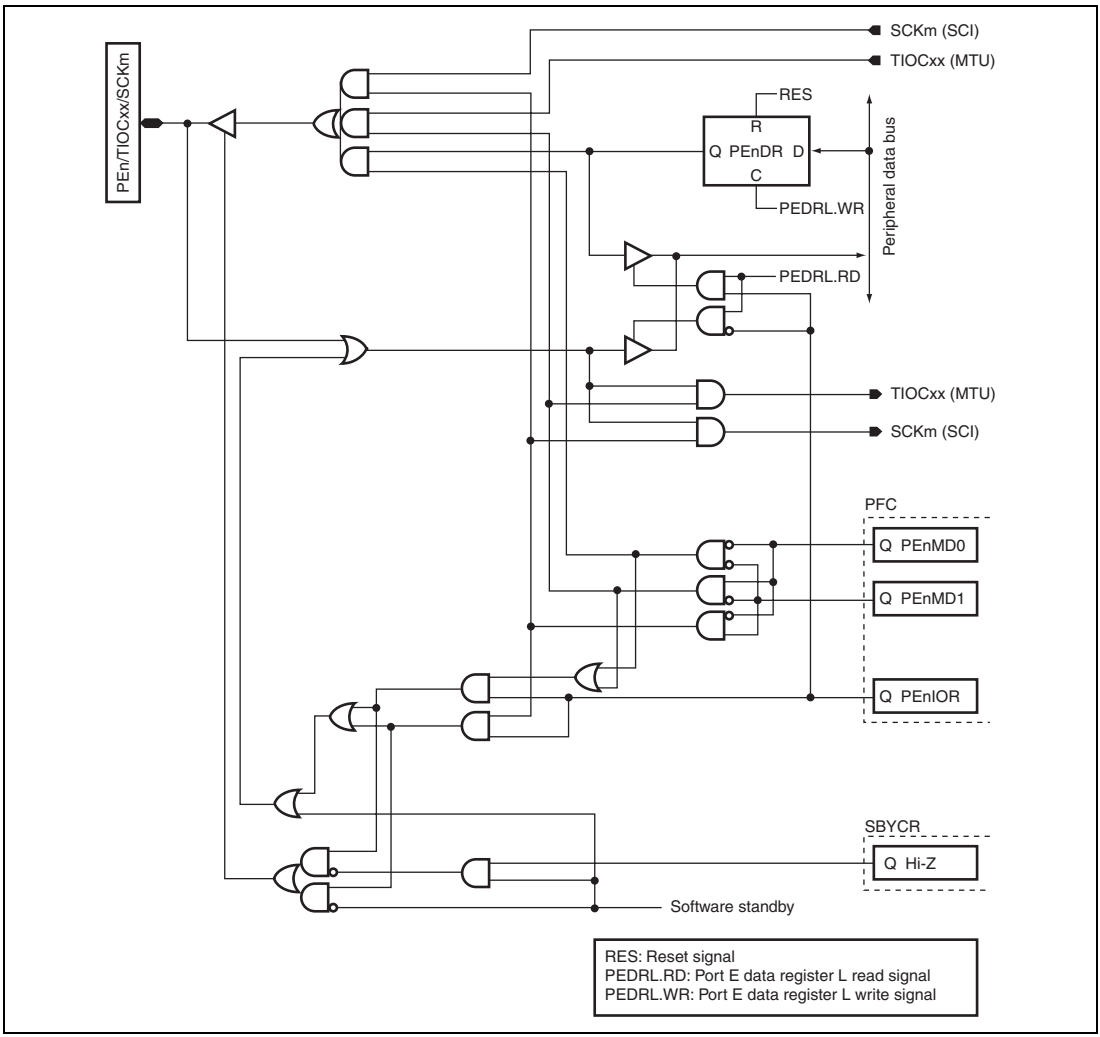


Figure D.36 PEn/TIOCxx/SCKm

Symbol in Figure D.36				Available Products			
				SH7144		SH7145	
Pins	PEn	TIOCxx	SCKm	F-ZTAT version	Masked ROM version/ ROM less version	F-ZTAT version	Masked ROM version/ ROM less version
PE9/TIOC3B/ SCK3	PE9	TIOC3B (MTU)	SCK3 (SCI)	√	√	—	√
PE8/TIOC3A/ SCK2	PE8	TIOC3A (MTU)	SCK2 (SCI)	√	√	—	√