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	H8S/2600
Core Size	16-Bit
Speed	20MHz
Connectivity	CANbus, SCI, SmartCard
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	92
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-BFQFP
Supplier Device Package	144-QFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2648rfc20jv

Type	Symbol	I/O	Name and Function
Data bus	D15 to D0	I/O	Data bus: These pins constitute a bidirectional data bus.
Bus control	$\overline{AS}$	Output	Address strobe: When this pin is low, it indicates that address output on the address bus is enabled.
	$\overline{RD}$	Output	Read: When this pin is low, it indicates that the external address space can be read.
	$\overline{HWR}$	Output	High write: A strobe signal that writes to external space and indicates that the upper half (D15 to D8) of the data bus is enabled.
	$\overline{LWR}$	Output	Low write: A strobe signal that writes to external space and indicates that the lower half (D7 to D0) of the data bus is enabled.
	$\overline{WAIT}$	Input	Wait: It is necessary to insert a wait state into the bus cycle when accessing the external three-state address space.
16-bit timer pulse unit (TPU)	TCLKD to TCLKA	Input	Clock input D to A: These pins input an external clock.
	TIOCA0, TIOCB0, TIOCC0, TIOCD0	I/O	Input capture/ output compare match A0 to D0: The TGR0A to TGR0D input capture input or output compare output, or PWM output pins.
	TIOCA1, TIOCB1	I/O	Input capture/ output compare match A1 and B1: The TGR1A and TGR1B input capture input or output compare output, or PWM output pins.
	TIOCA2, TIOCB2	I/O	Input capture/ output compare match A2 and B2: The TGR2A and TGR2B input capture input or output compare output, or PWM output pins.
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	I/O	Input capture/ output compare match A3 to D3: The TGR3A to TGR3D input capture input or output compare output, or PWM output pins.
	TIOCA4, TIOCB4	I/O	Input capture/output compare match A4 and B4: The TGR4A and TGR4B input capture input or output compare output, or PWM output pins.
	TIOCA5, TIOCB5	I/O	Input capture/output compare match A5 and B5: The TGR5A and TGR5B input capture input or output compare output, or PWM output pins.

Type	Instruction	Size*1	Function
Arithmetic operations	DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
	CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
	NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
	EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
	EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
	TAS	B	@ERd – 0, 1 → (<bit 7> of @ERd)*2 Tests memory contents, and sets the most significant bit (bit 7) to 1.
	MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating
	CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
	LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accumulate register.

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8S/2646 Group has four operating modes (modes 4 to 7). These modes enable selection of the CPU operating mode, enabling/disabling of on-chip ROM, and the initial bus width setting, by setting the mode pins (MD2 to MD0).

Table 3.1 lists the MCU operating modes.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
							Initial Width	Max. Width
0*	0	0	0	—	—	—	—	—
1*			1	—				
2*		1	0					
3*			1					
4	1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bits	16 bits
5			1				8 bits	16 bits
6		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits	16 bits
7			1		Single-chip mode		—	—

Note: * Not available in the H8S/2646 Group.

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2646 Group actually accesses a maximum of 16 Mbytes.

Modes 4 to 6 are externally expanded modes that allow access to external memory and peripheral devices.

The external expansion modes allow switching between 8-bit and 16-bit bus modes. After program execution starts, an 8-bit or 16-bit address space can be set for each area, depending on the bus controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; if 8-bit access is selected for all areas, 8-bit bus mode is set.

Note that the functions of each pin depend on the operating mode.

The H8S/2646 Group can be used only in modes 4 to 7. This means that the mode pins must be set to select one of these modes. Do not change the inputs at the mode pins during operation.

3.1.2 Register Configuration

The H8S/2646 Group has a mode control register (MDCR) that indicates the inputs at the mode pins (MD2 to MD0), and a system control register (SYSCR) that controls the operation of the H8S/2646 Group. Table 3.2 summarizes these registers.

Table 3.2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R	Undetermined	H'FDE7
System control register	SYSCR	R/W	H'01	H'FDE5
Pin function control register	PFCR	R/W	H'0D/H'00	H'FDEB

Note: * Lower 16 bits of the address.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
R/W	:	—	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8S/2646 Group.

Bit 7—Reserved: Cannot be written to.

Bits 6 to 3—Reserved: These bits are always read as 0 and cannot be written to.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits, and they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are cancelled by a reset.

5.4.3 Interrupt Control Mode 2

Eight-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Figure 5.6 shows a flowchart of the interrupt acceptance operation in this case.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.

[2] When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.4 is selected.

[3] Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.

[4] When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.

[5] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.

If the accepted interrupt is NMI, the interrupt mask level is set to H'7.

[7] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

7.4 Basic Bus Interface

7.4.1 Overview

The basic bus interface enables direct connection of ROM, SRAM, and so on.

The bus specifications can be selected with ABWCR, ASTCR, WCRH, and WCRL (see table 7.3).

7.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space: Figure 7.3 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word transfer instruction is performed as two byte accesses, and a longword transfer instruction, as four byte accesses.

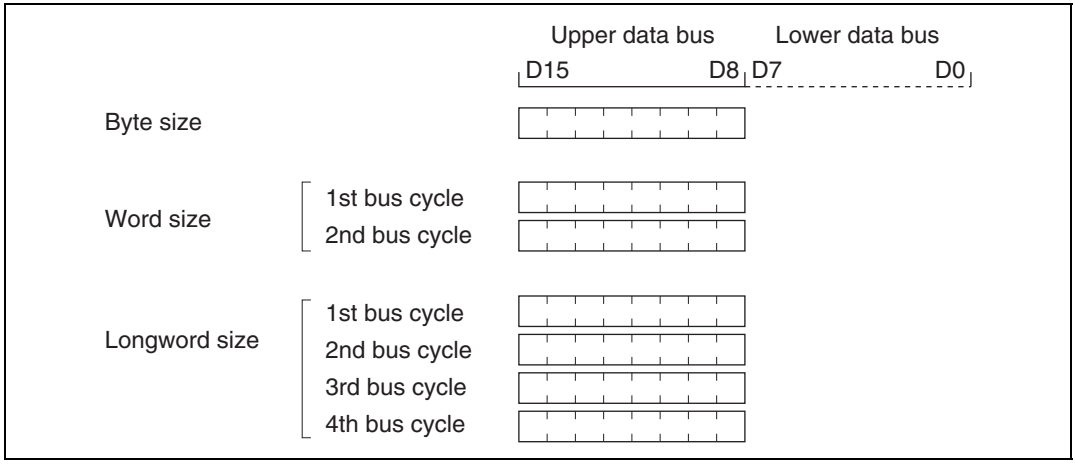


Figure 7.3 Access Sizes and Data Alignment Control (8-Bit Access Space)

8.4 Interrupts

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

8.5 Usage Notes

Module Stop: When the MSTPA6 bit in MSTPCRA is set to 1, the DTC clock stops, and the DTC enters the module stop state. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating.

On-Chip RAM: The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

DTCE Bit Setting: For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are masked, multiple activation sources can be set at one time by writing data after executing a dummy read on the relevant register.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
0	0	0	0	Internal clock: counts on ϕ/1	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	External clock: counts on TCLKD pin input	

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
1	0	0	0	Internal clock: counts on ϕ/1	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	Internal clock: counts on ϕ/256	
			1	Counts on TCNT2 overflow/underflow	

Note: This setting is ignored when channel 1 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
2	0	0	0	Internal clock: counts on ϕ/1	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	Internal clock: counts on ϕ/1024	

Note: This setting is ignored when channel 2 is in phase counting mode.

Example of Synchronous Operation: Figure 10.15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGR0B compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGR0B compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGR0B is used as the PWM cycle.

For details of PWM modes, see section 10.4.6, PWM Modes.

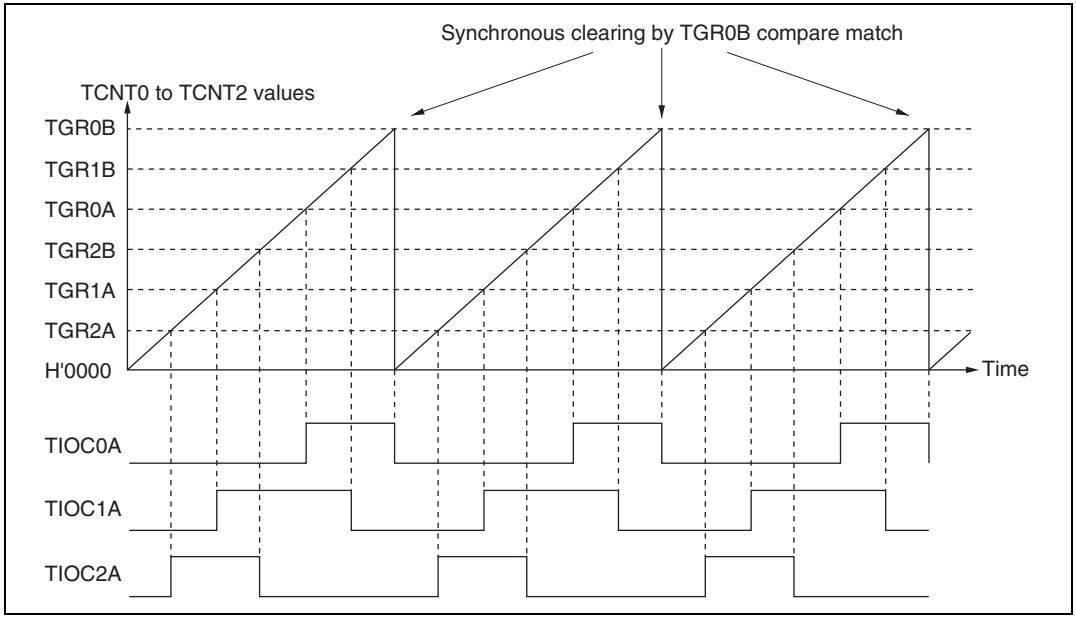


Figure 10.15 Example of Synchronous Operation

11.1.3 Pin Configuration

Table 11.1 summarizes the PPG pins.

Table 11.1 PPG Pins

Name	Symbol	I/O	Function
Pulse output 8	PO8	Output	Group 2 pulse output
Pulse output 9	PO9	Output	
Pulse output 10	PO10	Output	
Pulse output 11	PO11	Output	
Pulse output 12	PO12	Output	Group 3 pulse output
Pulse output 13	PO13	Output	
Pulse output 14	PO14	Output	
Pulse output 15	PO15	Output	

15.2.10 Remote Request Register (RFPR)

The remote request register (RFPR) is a 16-bit readable/writable register containing status flags that indicate normal reception of remote frames in mailboxes (buffers). When this bit is set, the corresponding receive-completed bit is set the same time.

RFPR

Bit:	15	14	13	12	11	10	9	8
	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only a write of 1 is permitted, to clear the flag.

Bits 15 to 0—Remote Request Register: These bits indicate that a remote frame has been received normally in the corresponding mailbox.

Bit x RFPRx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of remote frame reception in corresponding mailbox

(x = 15 to 0)

MCx[1] Bits 3 to 0—Data Length Code (DLC): These bits indicate the required length of data frames and remote frames.

Bit 3 DLC3	Bit 2 DLC2	Bit 1 DLC1	Bit 0 DLC0	Description
0	0	0	0	Data length = 0 bytes
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

MCx[3] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

MCx[4] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

MCx[6] Bits 7 to 0—Standard Identifier (STD_ID10 to STD_ID3):
MCx[5] Bits 7 to 5—Standard Identifier (STD_ID2 to STD_ID0):
These bits set the identifier (standard identifier) of data frames and remote frames.

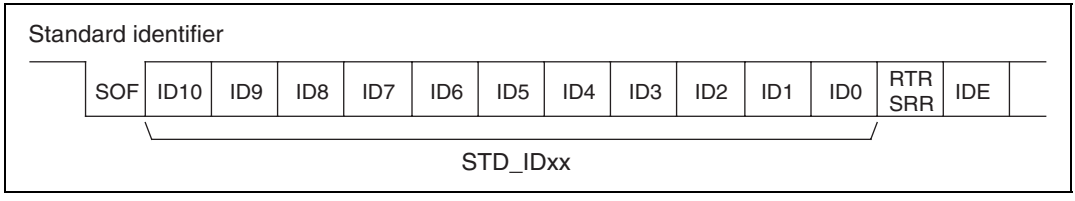


Figure 15.2 Standard Identifier

- Ongoing message transmission from a mailbox is canceled by TXCR.

If this occurs, the transmission is canceled but TXPR and TXCR continue to indicate a wrong status telling that a message is being cancelled. As a result, transmission cannot be restarted even after the HRxD pin is released from the tied state and the CAN bus has recovered. If there are two or more messages for transmission, a message which is not being transmitted is canceled and a message being transmitted retains its state.

To avoid this, take either of the following countermeasures.

[Countermeasures]

- Do not cancel transmission by TXCR. Transmission will be completed after the CAN bus has recovered, then TXPR is cleared and the HCAN operates normally.
- To cancel transmission, write 1 to the corresponding bit in TXCR repeatedly until the bit becomes 0. TXPR and TXCR are cleared, and the HCAN operates normally.

B. When the bus-off state is entered while any mailbox is waiting for transmission with TXPR set, transmission cannot be canceled even if TXCR is set because the internal state machine does not operate during the bus-off state. Because of this, on recovery from the bus-off state, one message will be transmitted or the message will be canceled with a transmission error. For message clearing on recovery from the bus-off state, take the following countermeasure.

[Countermeasure]

- Reset the HCAN during the bus-off period to clear the messages in the mailboxes waiting for transmission. To reset the HCAN, set the module stop bit (MSTPC3 in MSTPCRC) to 1 and then clear it. In this case, the HCAN is entirely reset. Therefore the initial settings must be made again.

14. HCAN Transmit Procedure

When transmission is set while the bus is in the idle state, if the next transmission is set or the set transmission is canceled under the following conditions within 50 µs, the transmit message ID of being set may be damaged.

— When the second transmission has the message whose priority is higher than the first one

— When the massage of the highest priority is canceled in the first transmission

Make whichever setting shown below to avoid the message IDs from being damaged.

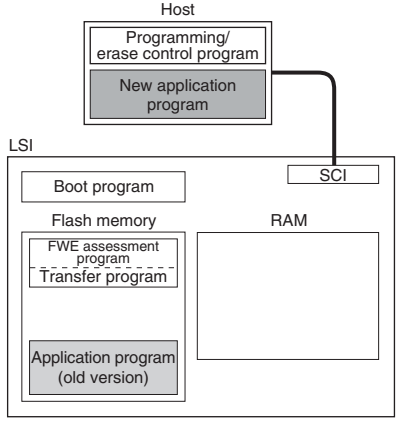
— Set transmission in one TXPR. After transmission of all transmit messages is completed, set transmission again (mass transmission setting). The interval between transmission settings should be 50 µs or longer.

— Make the transmission setting according to the priority of transmit messages.

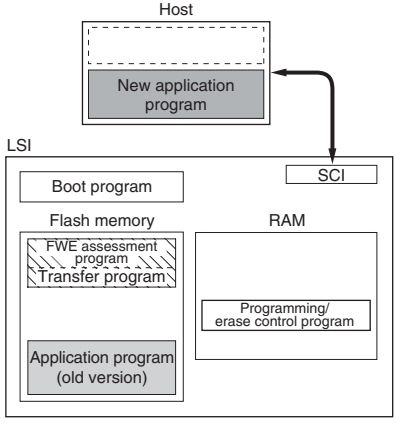
— Set the interval to be 50 µs or longer between TXPR and another TXPR or between TXPR and TXCR.

User Program Mode

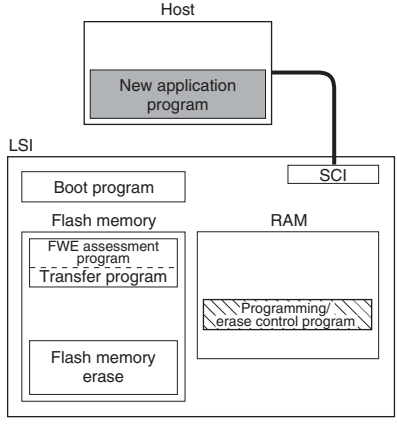
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



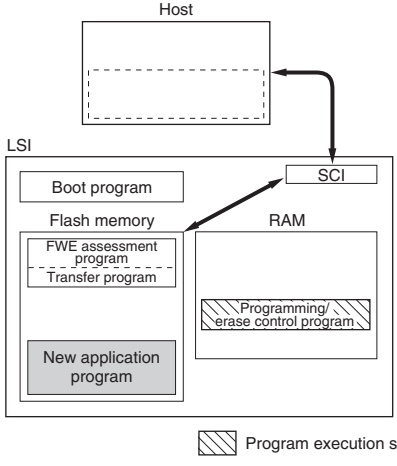
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



MC67

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC68

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

TMDR5—Timer Mode Register 5 H'FEA1 TPU5

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Mode

0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*: Don't care

Note: MD3 is a reserved bit. In a write, it should always be written with 0.

TCNT1—Timer Counter 1 **H'FF26** **TPU1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up/down-counter*

Note: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

TGR1A—Timer General Register 1A **H'FF28** **TPU1**
TGR1B—Timer General Register 1B **H'FF2A** **TPU1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PORT4—Port 4 Register **H'FFB3** **Port**

Bit	7	6	5	4	3	2	1	0
	P47	P46	P45	P44	P43	P42	P41	P40
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port 4 pins

Note: * Determined by state of pins P47 to P40.

PORT5—Port 5 Register **H'FFB4** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	P52	P51	P50
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	—*	—*	—*
Read/Write	—	—	—	—	—	R	R	R

State of the port 5 pins

Note: * Determined by state of pins P52 to P50.

PORT9—Port 9 Register **H'FFB8** **Port**

Bit	7	6	5	4	3	2	1	0
	P97	P96	P95	P94	P93	P92	P91	P90
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port 9 pins

Note: * Determined by state of pins P97 to P90.

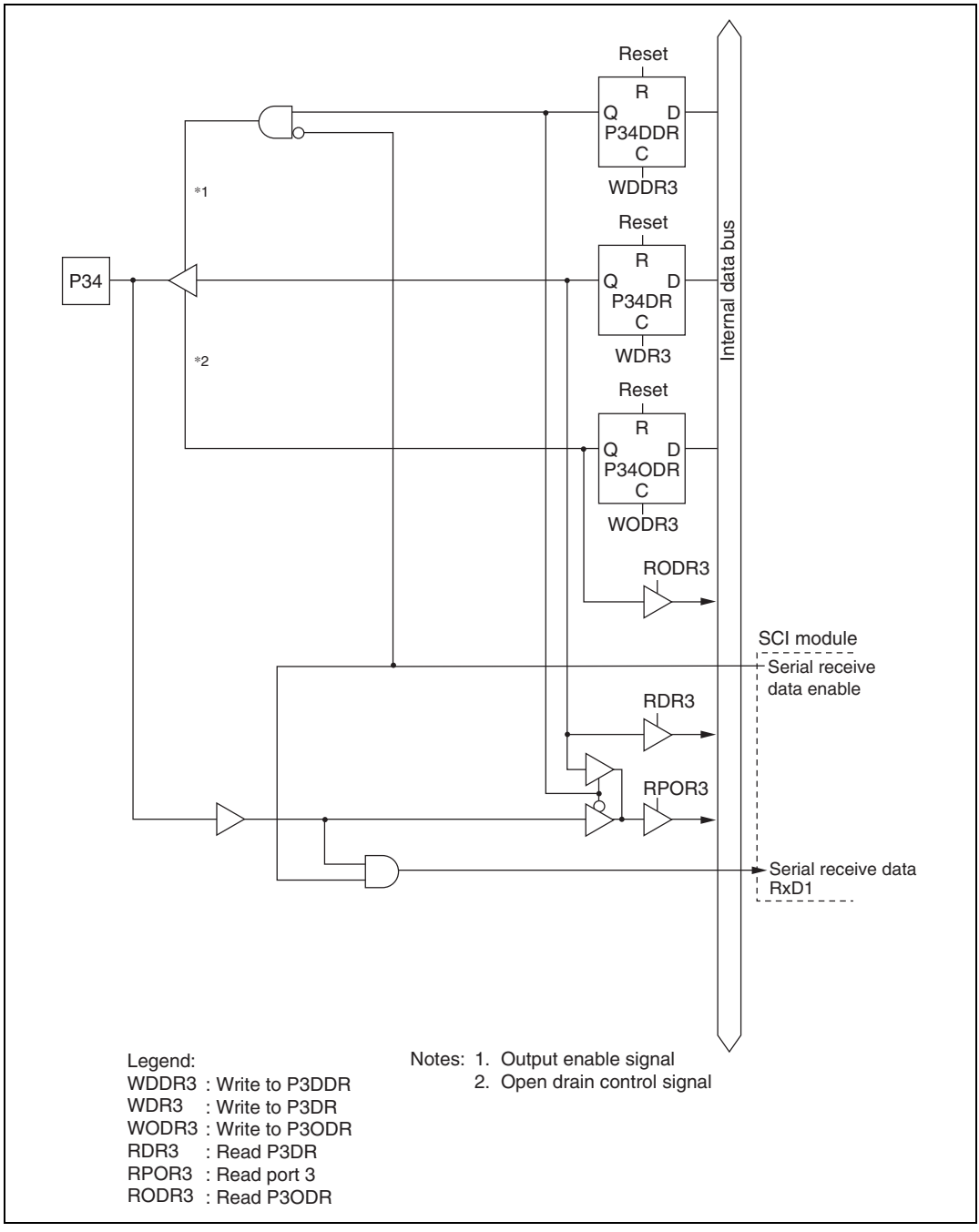


Figure C.3 (e) Port 3 Block Diagram (Pin P34)

Table D.2 I/O Port States in Each Processing State (H8S/2648, H8S/2648R, H8S/2647)

Port Name / Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State / Sleep Mode
Port 1	4 to 7	T	T	kept	I/O port
Port 2	4 to 7	T	T	kept	I/O port
Port 3	4 to 7	T	T	kept	I/O port
Port 4	4 to 7	T	T	T	Input port
Port 5	4 to 7	T	T	kept	I/O port
Port 9	4 to 7	T	T	T	Input port
Port A	4, 5	L	T	[Address output, OPE = 0] T [Address output, OPE = 1] kept [Segment, common output] port [Otherwise] kept	[Address output] A23 to A16 [Segment, common output] SEG40 to SEG37 COM4 to COM1 [Otherwise] I/O port
	6	T	T		
	7	T	T	[Segment, common output] port [Otherwise] kept	[Segment, common output] SEG40 to SEG37 COM4 to COM1 [Otherwise] I/O port
Port B	4, 5	L	T	[Address output, OPE = 0] T [Address output, OPE = 1] kept [Segment output] port [Otherwise] kept	[Address output] A15 to A8 [Segment output] SEG32 to SEG25 [Otherwise] I/O port
	6	T	T		
	7	T	T	[Segment output] port [Otherwise] kept	[Segment output] SEG32 to SEG25 [Otherwise] I/O port