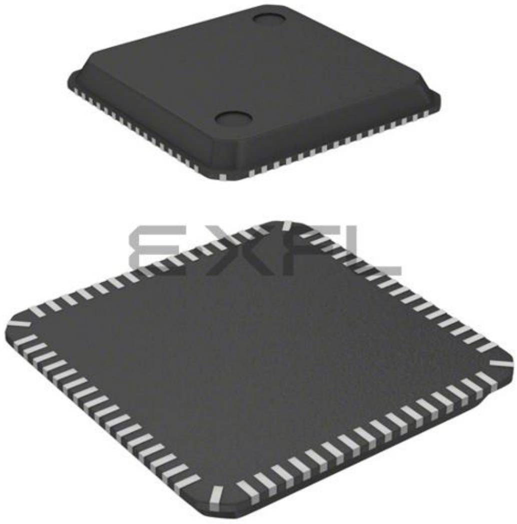
Renesas Electronics America Inc - DF2211CUNP24V Datasheet



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	Active
Core Processor	H8S/2000
Core Size	16-Bit
Speed	24MHz
Connectivity	SCI, SmartCard, USB
Peripherals	DMA, POR, PWM, WDT
Number of I/O	37
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 6x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN
Supplier Device Package	64-VQFN (8.2x8.2)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2211cunp24v

2.7 Addressing Modes and Effective Address Calculation

The H8S/2000 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.11 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

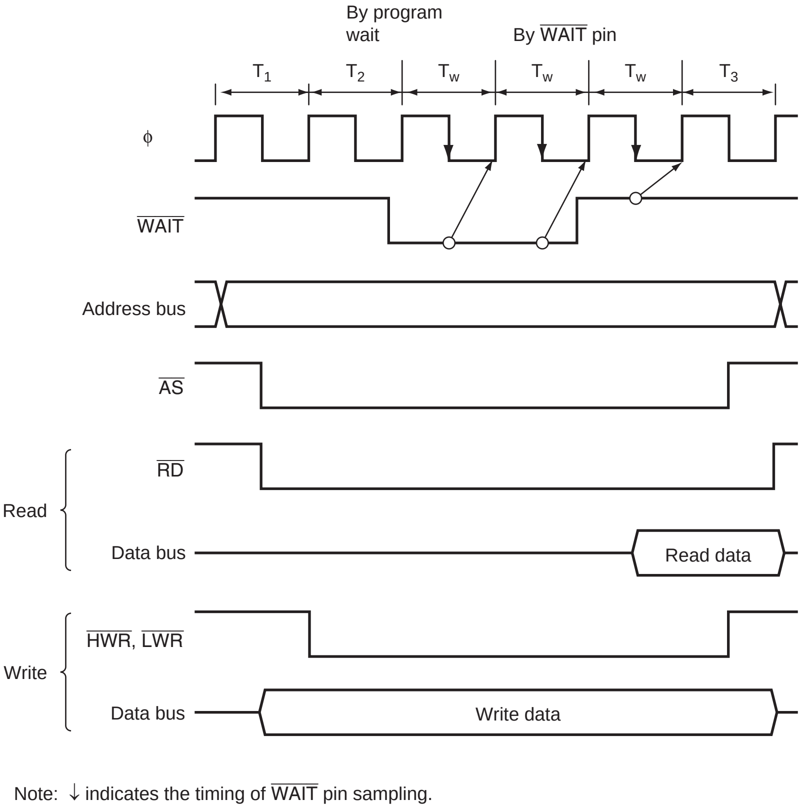
The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

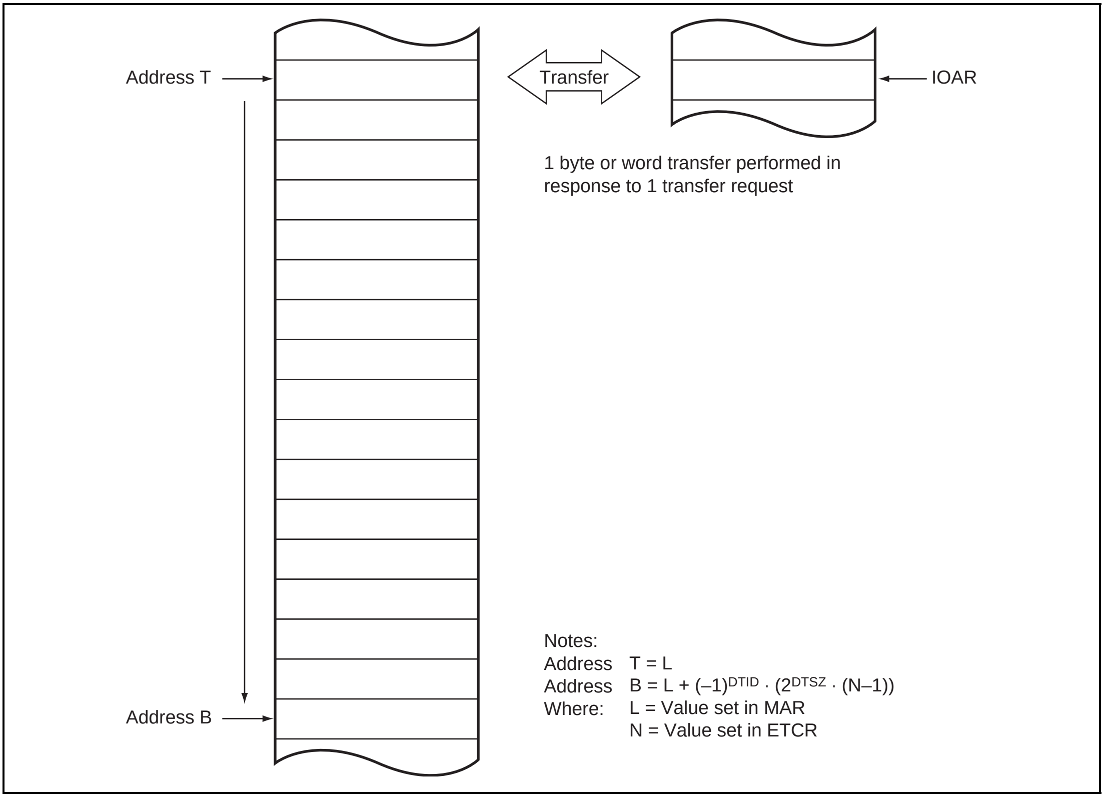
Figure 6.19 shows an example of wait state insertion timing.

In the H8S/2212 Group, the WAITE bit in BCRH should not be set to 1.



Note: ↓ indicates the timing of $\overline{\text{WAIT}}$ pin sampling.

Figure 6.19 Example of Wait State Insertion Timing


Address T
Transfer
IOAR
1 byte or word transfer performed in response to 1 transfer request
Address B

Notes:
Address T = L
Address B = L + (–1)^{DTID} · (2^{DTSZ} · (N–1))
Where: L = Value set in MAR
N = Value set in ETCR

Figure 7.2 Operation in Sequential Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU. The maximum number of transfers, when H'0000 is set in ETCR, is 65,536. Transfer requests (activation sources) consist of A/D conversion end interrupt, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare match/input capture A interrupts. Figure 7.3 shows an example of the setting procedure for sequential mode.

7.4.5 Normal Mode

In normal mode, transfer is performed with channels A and B used in combination. Normal mode can be specified by setting the FAE bit in DMABCR to 1 and clearing the BLKE bit in DMACRA to 0. In normal mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCRA. The transfer source is specified by MARA, and the transfer destination by MARB. Table 7.6 summarizes register functions in normal mode.

Table 7.6 Register Functions in Normal Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Transfer counter	Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed. Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB.

The number of transfers is specified by ETCRA as 16 bits. ETCRA is decremented each time a transfer is performed, and when its value reaches H'0000 the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU. The maximum number of transfers, when H'0000 is set in ETCRA, is 65,536.

8.10 Port E

The port E is an 8-bit I/O port also functioning as data bus (D7 to D0) I/O pins. The port E has the following registers.

- Port E data direction register (PEDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E pull-up MOS control register (PEPCR)

8.10.1 Port E Data Direction Register (PEDDR)

PEDDR specifies input or output for the pins of the port E.

Since PEDDR is a write-only register, the bit manipulation instructions must not be used to write PEDDR. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	(H8S/2218 Group) Modes 4 to 6: When 8-bit bus mode is selected, port E functions as an I/O port. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port. When 16-bit bus mode is selected, the input/output direction settings in PEDDR are ignored, and port E pins automatically function as data input/output pins. For details on 8-bit/16-bit bus mode, refer to section 6, Bus Controller. Mode 7: Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port. (H8S/2212 Group) Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

Table 9.3 CCLR2 to CCLR0 (channel 0)

	Bit 7	Bit 6	Bit 5	
Channel	**CCLR2**	**CCLR1**	**CCLR0**	**Description**
0	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

2. When TGRC or TGRD is used as a buffer register. TCNT is not cleared because the buffer register setting has priority, and compare match/input capture dose not occur.

Table 9.4 CCLR2 to CCLR0 (channels 1 and 2)

	Bit 7	Bit 6	Bit 5	
Channel	**Reserved*2**	**CCLR1**	**CCLR0**	**Description**
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

11.3.3 Hour Data Register (RHRDR)

RHRDR counts the BCD-coded hour value on the carry generated once per hour by RMINDR. This register is initialized to H'00 by a $\overline{\text{STBY}}$ input or the RST bit in RTCCR1, but not initialized by a $\overline{\text{RES}}$ input. The setting range is either decimal 0 to 11 or 0 to 23 by the selection of the 12/24 bit in RTCCR1.

Bit	Bit Name	Initial Value*	R/W	Description
7	BSY	—	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6	—	0	—	Reserved This bit is always read as 0.
5	HR11	—	R/W	Counting Ten's Position of Hours
4	HR10	—	R/W	Counts on 0 to 2 for ten's position of hours.
3	HR03	—	R/W	Counting One's Position of Hours
2	HR02	—	R/W	Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.
1	HR01	—	R/W	
0	HR00	—	R/W	

Note: * Initial value after $\overline{\text{RES}}$.

Table 11.3 Operating State in Each Mode

Function	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hard-ware Standby
Clock operation	Subclock operation	Subclock operation	Subclock operation	Halted (Retained)	Subclock operation	Subclock operation	Subclock operation	Subclock operation	Halted (Reset)
Free running timer operation	Operating	Operating	Operating	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)

11.7 Usage Notes

(1) Notes on Using the Emulator

In the E6000 emulator the RTC module is mounted on an external extended board. Since it must be accessed as an external module, the limitations listed below apply. These limitations do not apply to the E10A or to product chips.

- RTC operation is not supported in the H8S/2218 Group's mode 7 (single-chip mode).
- When using the RTC module in the H8S/2218 Group's mode 6 (on-chip ROM-enabled mode) or the H8S/2212 Group's mode 7 (single-chip mode), A7 to A0 are input pins in the initial status. Therefore, A7 to A0 must be set as output pins by setting PC7DDR to PC0DDR to H'FF before accessing the RTC module.
- The above setting is not necessary when using the RTC module in the H8S/2218 Group's modes 4 and 5 (on-chip ROM-disabled mode) because A7 to A0 are output pins.

(2) Bus Interface

The bus interface of the module conforms to the bus specifications for external area 7. Consequently, before accessing the RTC module, area 7 must be specified as having an 8-bit bus width and 3-state access using the bus controller register.

(3) Method for Reading Pin States Using the Port D Register (PORTD)

First access EXMDLSTP or the RTC register (address range: H'FFFF40 to H'FFFF5F). Then, you must perform a dummy read to the external address space (such as H'FFEF00 to H'FF7FF) outside the range H'FFFF40 to H'FFFF5F before reading PORTD.

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{\text{E}}$ bit settings are invalid in multiprocessor mode. For details, see section 12.5, Multi Processor Communication Function.
1	CKS1	0	R/W	Clock Select 0 and 1:
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: φ clock (n = 0) 01: φ/4 clock (n = 1) 10: φ/16 clock (n = 2) 11: φ/64 clock (n = 3) For the relationship between the bit rate register setting and the baud rate, see section 12.3.11, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 12.3.11, Bit Rate Register (BRR)).

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*[1]	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. [Setting condition] • When the next serial reception is completed while RDRF = 1 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after reading ORER = 1*[2] The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
4	ERS	0	R/(W)*[1]	Error Signal Status Indicates that the status of an error, signal 1 returned from the reception side at reception [Setting condition] • When the low level of the error signal is sampled [Clearing condition] • When 0 is written to ERS after reading ERS = 1*[2] The ERS flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

12.10.6 Operation in Case of Mode Transition

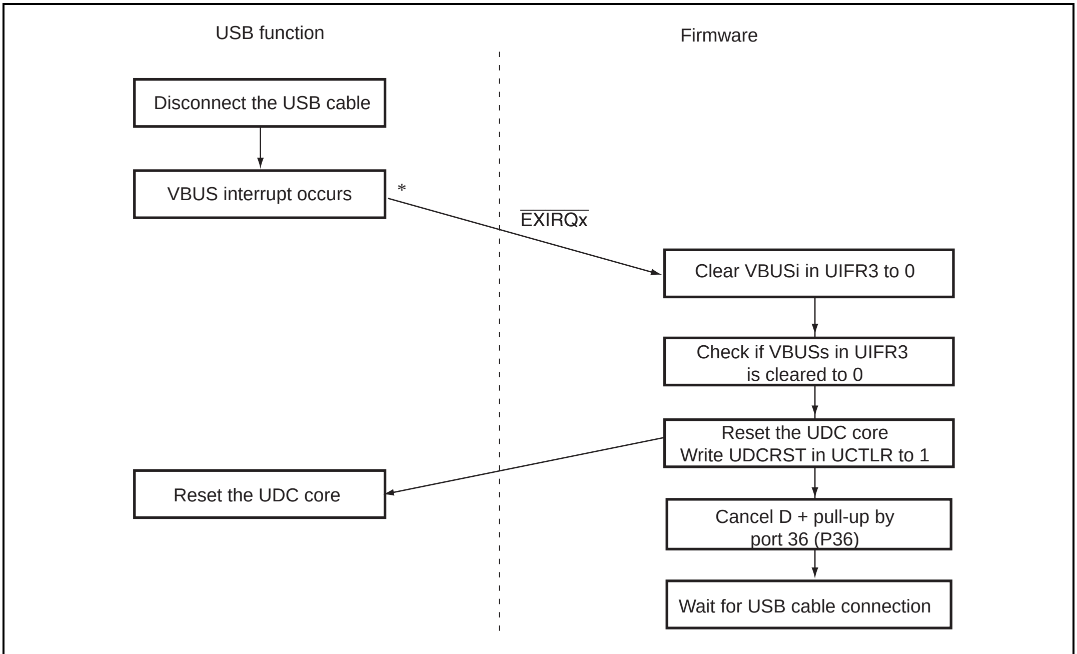
- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode depend on the port settings, and becomes high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined. When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence:

  SSR read -> TDR write -> TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization. Figure 12.39 shows a sample flowchart for mode transition during transmission. Port pin states are shown in figures 12.40 and 12.41.

(3) USB Cable Disconnection (When USB module stop or power-down mode is not used)

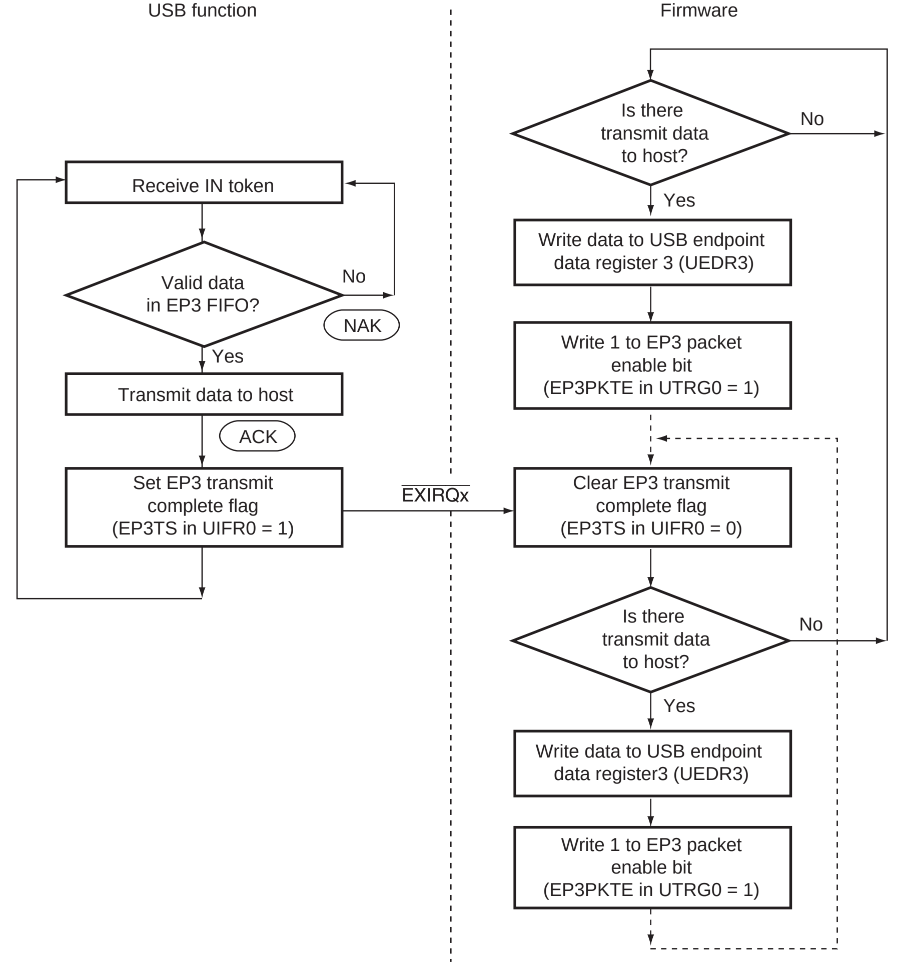
If the USB cable enters the disconnection state from the connection state in an application (self powered) where USB module stop or power-down mode is not used, perform the operation as shown in figure 14.5. In bus-powered mode, the power is automatically turned off when the USB cable is disconnected and the following processing is not required.


USB function
Firmware
Disconnect the USB cable
VBUS interrupt occurs
*
$\overline{\text{EXIRQx}}$
Clear VBUSi in UIFR3 to 0
Check if VBUSs in UIFR3 is cleared to 0
Reset the UDC core Write UDCRST in UCTLR to 1
Reset the UDC core
Cancel D + pull-up by port 36 (P36)
Wait for USB cable connection

Note: * VBUS interrupts in the USB module cannot be detected in power-down mode or in the USB module stop state.

Figure 14.5 USB Cable Disconnection (When USB Module Stop or Power-Down Mode Is not Used)

14.5.5 Interrupt-In Transfer (Endpoint 3)


USB function
Firmware
Receive IN token
Valid data in EP3 FIFO?
No
NAK
Yes
Transmit data to host
ACK
Set EP3 transmit complete flag (EP3TS in UIFR0 = 1)
EXIRQx
Is there transmit data to host?
No
Yes
Write data to USB endpoint data register 3 (UEDR3)
Write 1 to EP3 packet enable bit (EP3PKTE in UTRG0 = 1)
Clear EP3 transmit complete flag (EP3TS in UIFR0 = 0)
Is there transmit data to host?
No
Yes
Write data to USB endpoint data register3 (UEDR3)
Write 1 to EP3 packet enable bit (EP3PKTE in UTRG0 = 1)

Note: This flowchart shows just one example of interrupt-in transfer processing. Other possibilities include an operation flow in which, if there is data to be transmited, the EP3DE bit UDSR is referred to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 14.17 EP3 Interrupt-In Transfer Operation

15.4 Interface to Bus Master

ADDRA to ADDRD are 16-bit registers. As the data bus to the bus master is 8 bits wide, the bus master accesses to the upper byte of the registers directly while to the lower byte of the registers via the temporary register (TEMP).

Data in ADDR is read in the following way: When the upper-byte data is read, the upper-byte data will be transferred to the CPU and the lower-byte data will be transferred to TEMP. Then, when the lower-byte data is read, the lower-byte data will be transferred to the CPU.

When data in ADDR is read, the data should be read from the upper byte and lower byte in the order. When only the upper-byte data is read, the data is guaranteed. However, when only the lower-byte data is read, the data is not guaranteed.

Figure 15.2 shows data flow when accessing to ADDR.

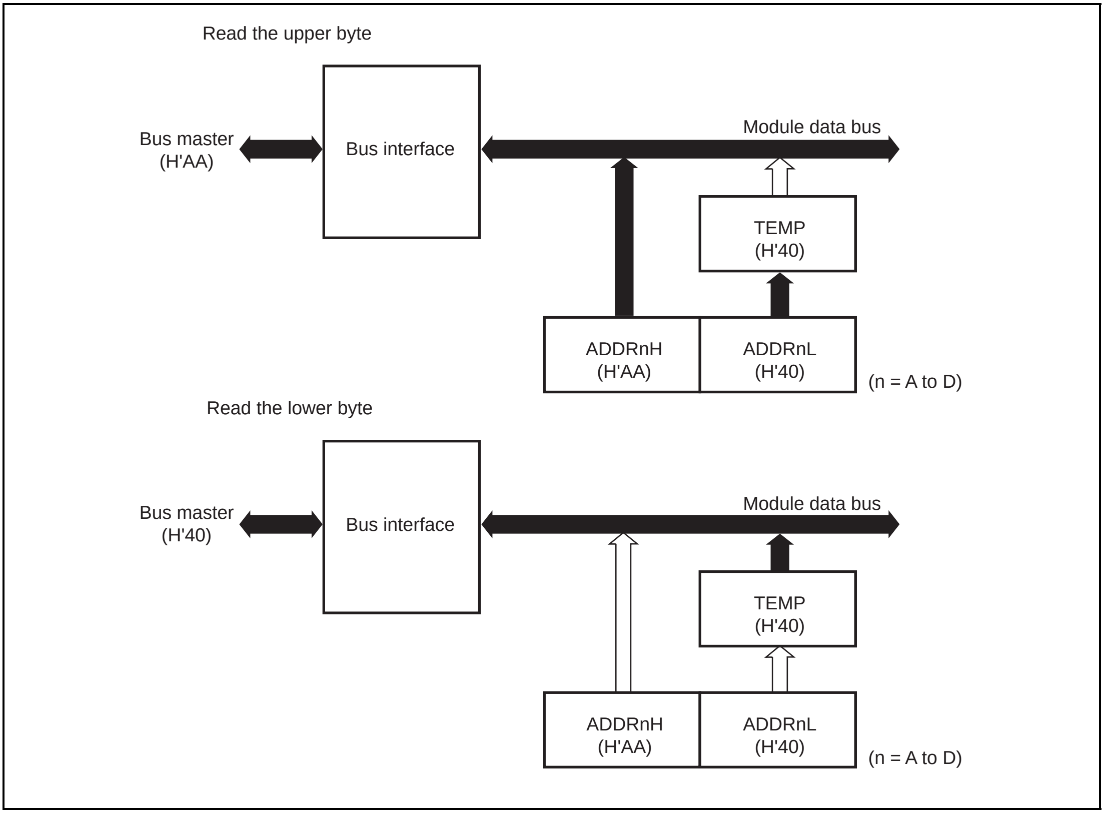


Figure 15.2 Access to ADDR (When Reading H'AA40)

17.4 Input/Output Pins

The flash memory is controlled by means of the pins shown in table 17.2.

Table 17.2 Pin Configuration

Pin Name	I/O	Function	
$\overline{\text{RES}}$	Input	Reset	All
FWE	Input	Flash program/erase protection by hardware	
MD2, MD1, MD0	Input	Sets this LSI's operating mode	
PF3, PF0, P16, P14	Input	Sets this LSI's operating mode in programmer mode	
EMLE	Input	Emulator enable	
TxD2	Output	Serial transmit data output	HD64F2218, HD64F2212, HD64F2211
RxD2	Input	Serial receive data input	
USD+, USD–	Input/output	USB data input/output	HD64F2218U, HD64F2218CU, HD64F2217CU, HD64F2212U, HD64F2212CU, HD64F2211U
VBUS	Input	USB cable connect/cut detect	
$\overline{\text{UBPM}}$	Input	USB bus power mode/self power mode select	
USPND	Output	USB suspend output	
P36 (PUPD+)	Output	D+ pull-up control	

17.5 Register Descriptions

The flash memory has the following registers. For details on register addresses and register states during each processing, refer to section 21, List of Registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)
- RAM emulation register (RAMER)
- Serial control register X (SCRX)

The masked ROM version is not equipped with the above registers. Attempting to read them with produce an undetermined value, and writing to them is invalid.

Bit	Bit Name	Initial Value	R/W	Description
2	PV1	0	R/W	Program-Verify When this bit is set to 1, the flash memory transits to program-verify mode. When it is cleared to 0, program-verify mode is cancelled. [Setting condition] When FWE = 1 and SWE1 = 1
1	E1	0	R/W	Erase When this bit is set to 1 while the SWE1 and ESU1 bits are 1, the flash memory transits to erase mode. When it is cleared to 0, erase mode is cancelled. [Setting condition] When FWE = 1, SWE1 = 1, and ESU1 = 1
0	P1	0	R/W	Program When this bit is set to 1 while the SWE1 and PSU1 bits are 1, the flash memory transits to program mode. When it is cleared to 0, program mode is cancelled. [Setting condition] When FWE = 1, SWE1 = 1, and PSU1 = 1

Note: * Set according to the FWE pin state.

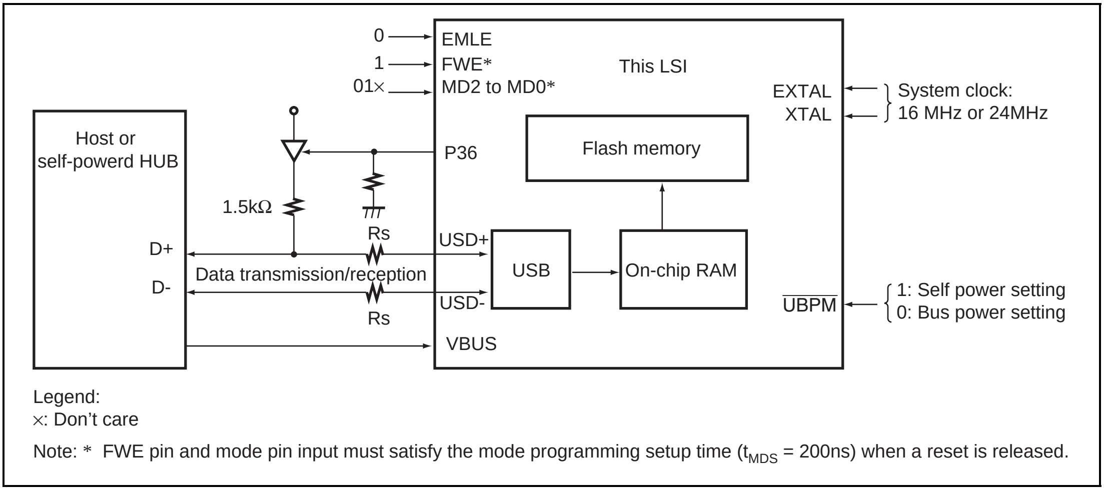
17.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLMCR2 is a read-only register, and should not be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See section 17.9.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

- Overview

  When a reset start preformed after the pins of this LSI have been set to boot mode, a boot program incorporated in the microcomputer beforehand is activated, and the prepared programming control program is transmitted sequentially to the host using the USB. With this LSI, the programming control program received by the USB is written to a programming control program area in on-chip RAM. After transfer is completed, control branches to the start address of the programming control program area, and the programming control program execution state is established (flash memory programming is performed). Figure 17.9 shows a system configuration diagram when using USB boot mode.



Legend:
×: Don’t care

Note: * FWE pin and mode pin input must satisfy the mode programming setup time (t_{MDS} = 200ns) when a reset is released.

Figure 17.9 System Configuration Diagram when Using USB Boot Mode

Table 17.7 shows operations from reset release in USB boot mode until processing branches 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 17.8, Flash Memory Programming/Erasing. In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use in enforced exit when user program mode is unavailable, such a the first time on-board programming control program, or performed, or if the program activated in user program mode is accidentally erased.
2. When the boot program is activated, enumeration with respect to the host is carried out. Enumeration information is shown in table 17.6. When enumeration is completed, transmit a single H'55 byte from the host. If reception has not been preformed normally, restart boot mode by means of a reset.

19.9 Usage Notes

19.9.1 Note on Crystal Resonator

Since various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

19.9.2 Note on Board Design

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL or OSC1 and EXTAL or OSC2 pins. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 19.10.

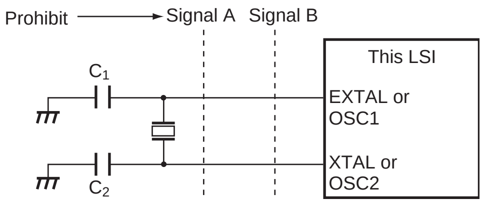


Figure 19.10 Note on Board Design of Oscillator Circuit

19.9.3 Note on Switchover of External Clock

When two or more external clocks (e.g. 16 MHz and 13 MHz) are used as the system clock, switchover of the input clock should be carried out in software standby mode.

An example of an external clock switching circuit is shown in figure 19.11, and an example of the external clock switchover timing in figure 19.12.

Bit	Bit Name	Initial Value	R/W	Description
6	STS2	0	R/W	Standby Timer Select 2 to 0 These bits select the MCU wait time for clock stabilization when cancel software standby mode, watch mode, or subactive mode by an external interrupt. With a crystal oscillator (tables 20.3 and 20.4), select a wait time of t_{OSC2} ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, there are no specific wait requirements. However, in the F-ZTAT version a standby time of 16 wait states cannot be used with an external clock. In this case, select a wait time of 100 µs or more. 000: Standby time = 8192 states 001: Standby time = 16384 states 010: Standby time = 32768 states 011: Standby time = 65536 states 100: Standby time = 131072 states 101: Standby time = 262144 states 110: Standby time = 2048 states 111: Standby time = 16 states
5	STS1	0	R/W	
4	STS0	0	R/W	
3	OPE	1	R/W	Output Port Enable This bit selects whether address bus and bus control signals ($\overline{\text{CS0}}$ to $\overline{\text{CS7}}$, $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, and $\overline{\text{LWR}}$) are brought to high impedance state or retained in software standby mode, watch mode, or direct transition. 0: High impedance state 1: Retained
2 to 0	—	All 0	—	Reserved These bits are always read as 0, and cannot be modified.