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	Motor Control PWM, POR, PWM, WDT
Number of I/O	72
Program Memory Size	384KB (384K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	128-BFQFP
Supplier Device Package	128-QFP (14x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2630uf16lv

2.6.3 Table of Instructions Classified by Function

Table 2-3 summarizes the instructions in each functional category. The notation used in table 2-3 is defined below.

Operation Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
¬	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

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)*3 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.

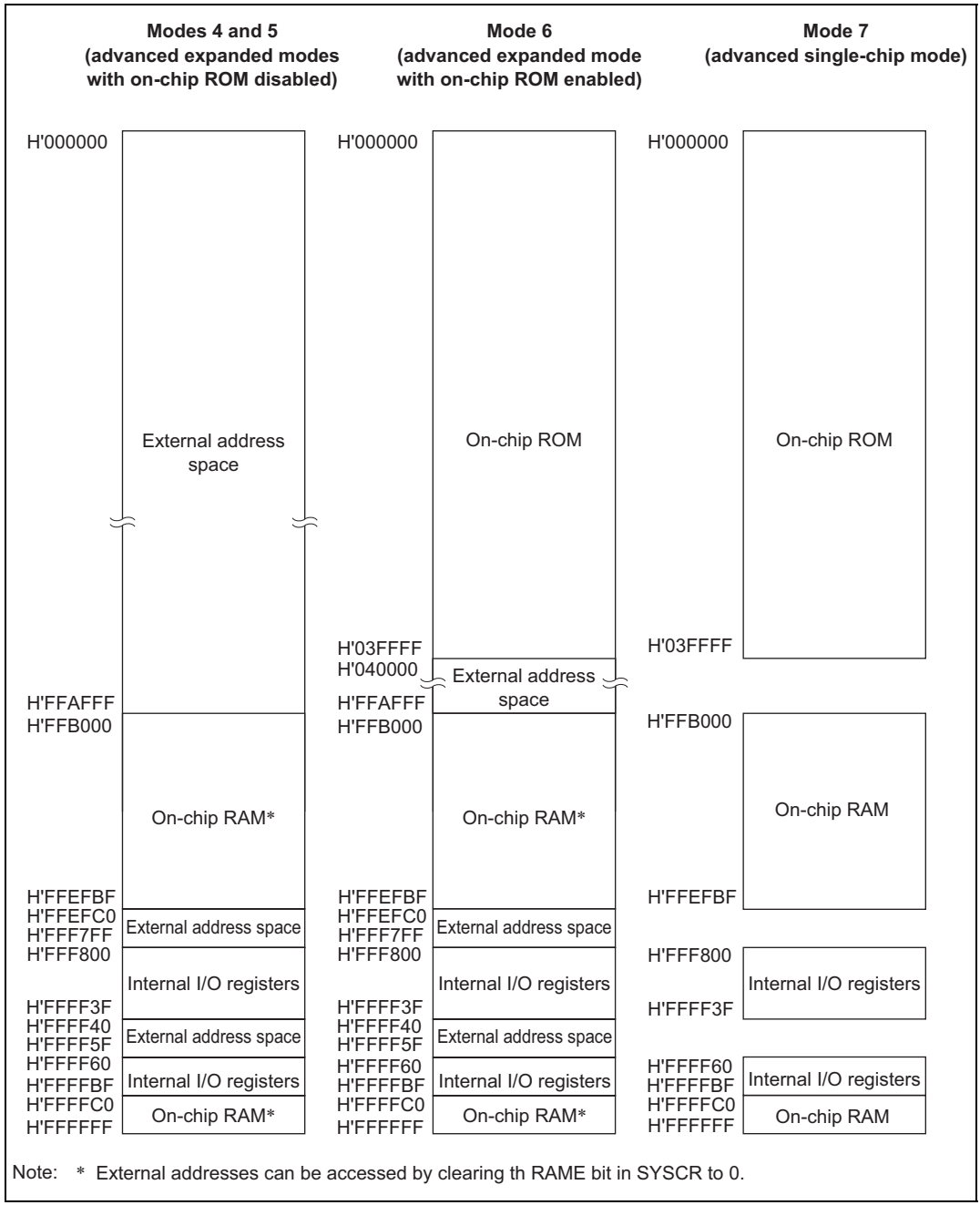


Figure 3-2 Memory Map in Each Operating Mode in the H8S/2638 and H8S/2639

9.7.2 Register Configuration

Table 9-11 shows the port B register configuration.

Table 9-11 Port B Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port B data direction register	PBDDR	W	H'00	H'FE3A
Port B data register	PBDR	R/W	H'00	H'FF0A
Port B register	PORTB	R	Undefined	H'FFBA
Port B MOS pull-up control register	PBPCR	R/W	H'00	H'FF41
Port B open-drain control register	PBODR	R/W	H'00	H'FE48

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

PBDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 to 6

  The corresponding port B pins become address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, irrespective of the value of the PBDDR bits. When pins are not used as address outputs, setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

9.10 Port E

9.10.1 Overview

Port E is an 8-bit I/O port. Port E has a data bus I/O function, and the pin functions change according to the operating mode and whether 8-bit or 16-bit bus mode is selected.

Port E has an on-chip MOS input pull-up function that can be controlled by software.

Figure 9-18 shows the port E pin configuration.

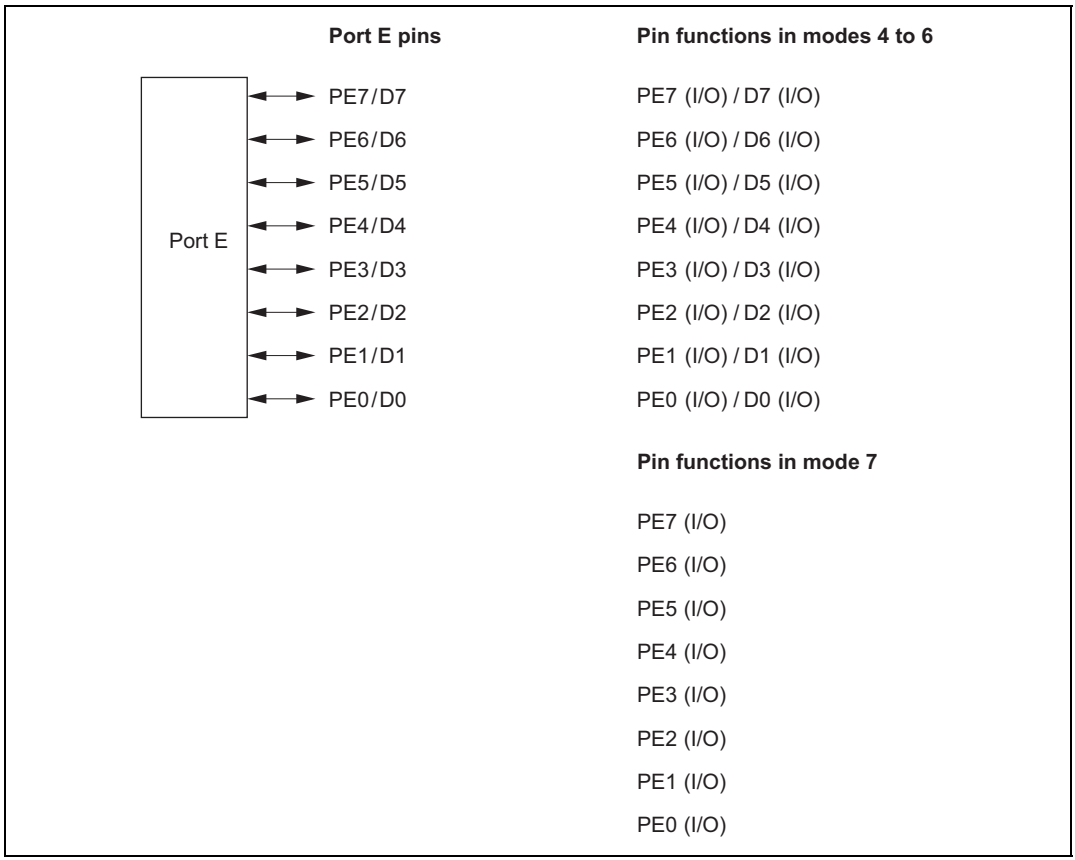


Figure 9-18 Port E Pin Functions

Port E Data Register (PEDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
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

PEDR is an 8-bit readable/writable register that stores output data for the port E pins (PE7 to PE0).

PEDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1	0
		PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PE7 to PE0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port E pins (PE7 to PE0) must always be performed on PEDR.

If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read. If a port E read is performed while PEDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTE contents are determined by the pin states, as PEDDR and PEDR are initialized. PORTE retains its prior state in software standby mode.

11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the PPG.

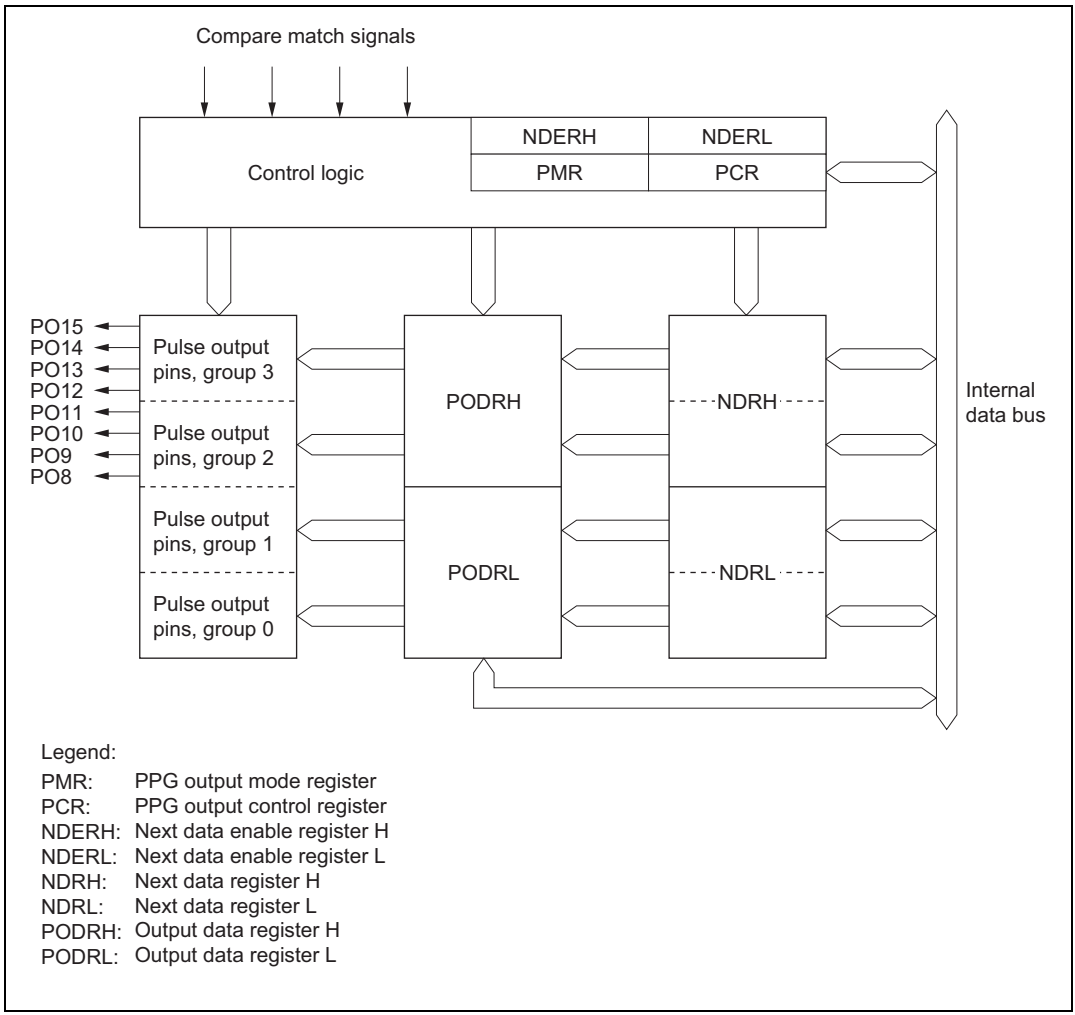


Figure 11-1 Block Diagram of PPG

Switching from SCK Pin Function to Port Pin Function:

- Problem in Operation: When switching the SCK pin function to the output port function (high-level output) by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1 (synchronous mode), low-level output occurs for one half-cycle.

1. End of serial data transmission
2. TE bit = 0
3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 13-27)

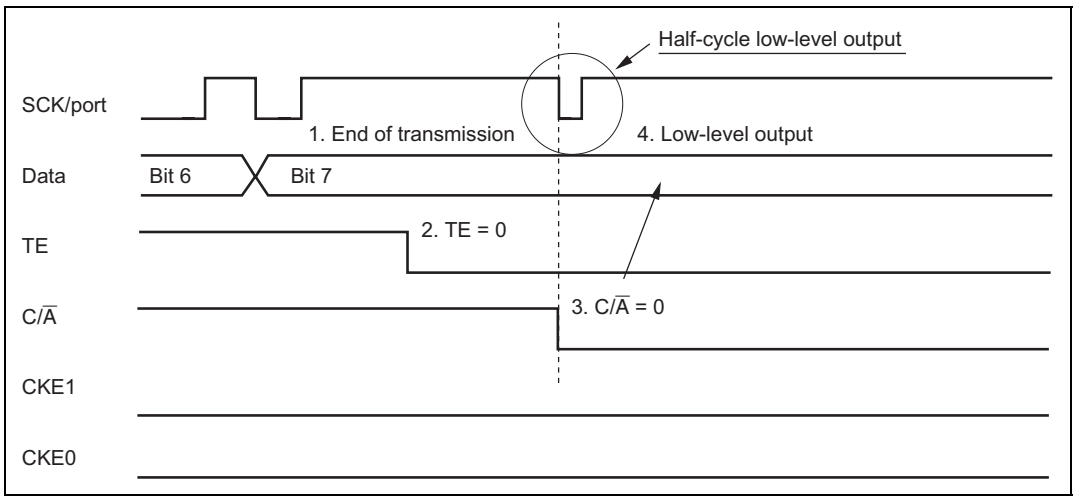


Figure 13-27 Operation when Switching from SCK Pin Function to Port Pin Function

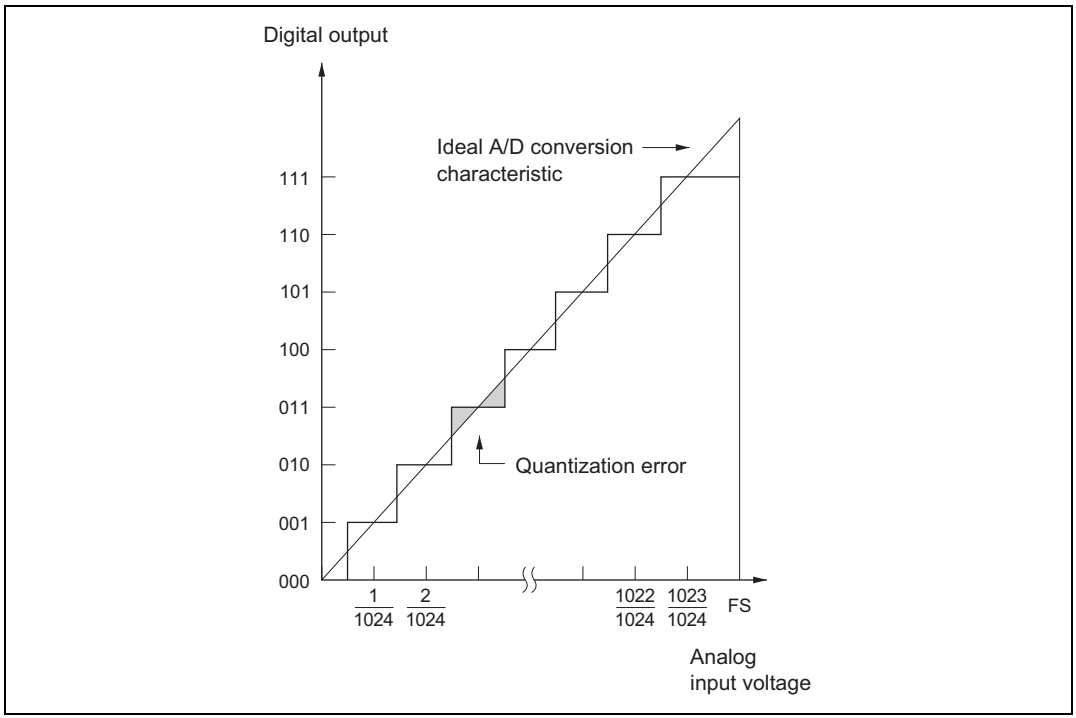


Figure 17-9 A/D Conversion Precision Definitions (1)

- Before branching to the programming control program (RAM area H'FFE800), the chip terminates transmit and receive operations by the on-chip SCI (channel 1) (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD1, goes to the high-level output state (P33DDR = 1, P33DR = 1).

  The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

  The initial values of other on-chip registers are not changed.

- Boot mode can be entered by making the pin settings shown in table 21A-9 and executing a reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release*1. Boot mode can also be cleared by a WDT overflow reset.

  Do not change the mode pin input levels in boot mode, and do not drive the FWE pin low*3 while the boot program is being executed or while flash memory is being programmed or erased.

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) will change according to the change in the microcomputer's operating mode*2.

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: 1. Mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 4 states) with respect to the reset release timing.

2. See appendix D, Pin States.

3. For precautions on applying and disconnecting FWE, see section 21A.15, Flash Memory Programming and Erasing Precautions.

21A.8.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means of FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary.

3. The H8S/2638 and H8S/2639 have 256 kbytes of on-chip ROM. The H8S/2630 has 384 kbytes of on-chip ROM.

Table 21B-3 Operating Modes and ROM (Mask ROM Version)

		Mode Pins			
	Operating Mode	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	—
Mode 1				1	
Mode 2			1	0	
Mode 3				1	
Mode 4	Advanced expanded mode with on-chip ROM disabled	1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled			1	
Mode 6	Advanced expanded mode with on-chip ROM enabled		1	0	Enabled (256 kbytes/ 384 kbytes)*
Mode 7	Advanced single-chip mode			1	Enabled (256 kbytes/ 384 kbytes)*

Note: * The H8S/2638 and H8S/2639 have 256 kbytes of on-chip ROM. The H8S/2630 has 384 kbytes of on-chip ROM.

FWE application/disconnection (see figures 21B-18 to 21B-20): FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

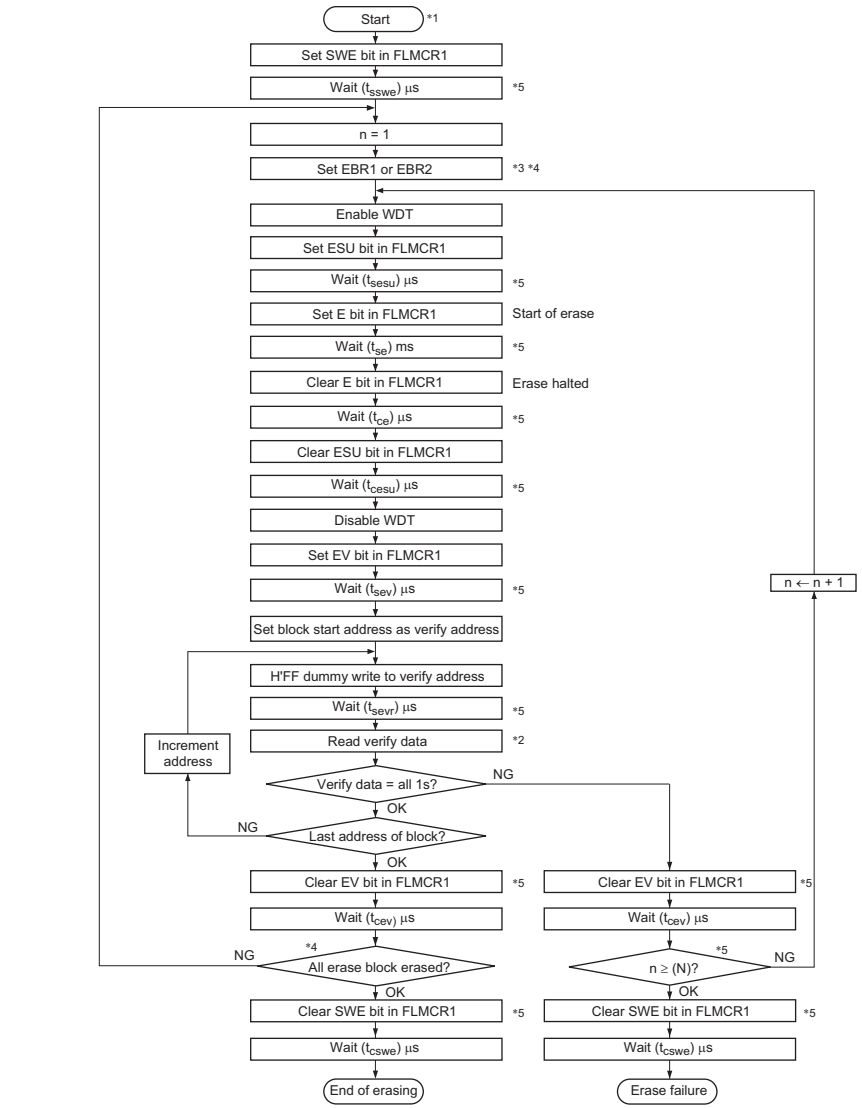
- Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.
  Apply FWE when oscillation has stabilized (after the elapse of the oscillation stabilization time).
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regardless of the reset state. FWE input can also be switched during execution of a program in flash memory.
- Do not apply FWE if program runaway has occurred.
- Disconnect FWE only when the SWE, ESU, PSU, EV, PV, P, and E bits in FLMCR1 are cleared.
  Make sure that the SWE, ESU, PSU, EV, PV, P, and E bits are not set by mistake when applying or disconnecting FWE.

Do not apply a constant high level to the FWE pin: Apply a high level to the FWE pin only when programming or erasing flash memory. A system configuration in which a high level is constantly applied to the FWE pin should be avoided. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE bit during execution of a program in flash memory: Wait for at least 100 µs after clearing the SWE bit before executing a program or reading data in flash memory. When the SWE bit is set, data in flash memory can be rewritten, but when SWE = 1, flash memory can only be read in program-verify or erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying.

Similarly, when using the RAM emulation function while a high level is being input to the FWE pin, the SWE bit must be cleared before executing a program or reading data in flash memory.



Notes: 1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Verify data is read in 16-bit (word) units.
3. Make only a single-bit specification in the erase block registers (EBR1 and EBR2). Two or more bits must not be set simultaneously.
4. Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.
5. The wait times and the value of N are shown in section 24.2.7, 24.3.7, and 24.4.7, Flash Memory Characteristics.

Figure 21C-13 Erase/Erase-Verify Flowchart

23B.5 Module Stop Mode

23B.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 23B-5 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI, Motor control PWM, A/D converter and HCAN are retained.

After reset clearance, all modules other than DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

Instruction	H	N	Z	V	C	Definition
SUB	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4 N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
SUBS	—	—	—	—	—	
SUBX	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4 N = Rm Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$ V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
TAS	—	↕	↕	0	—	N = Dm Z = $\overline{Dm}$ · $\overline{Dm–1}$ · ····· · $\overline{D0}$
TRAPA	—	—	—	—	—	
XOR	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
XORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F8C8	MD31									HCAN0	8/16
H'F8C9	MD32										
H'F8CA	MD33										
H'F8CB	MD34										
H'F8CC	MD35										
H'F8CD	MD36										
H'F8CE	MD37										
H'F8CF	MD38										
H'F8D0	MD41									HCAN0	8/16
H'F8D1	MD42										
H'F8D2	MD43										
H'F8D3	MD44										
H'F8D4	MD45										
H'F8D5	MD46										
H'F8D6	MD47										
H'F8D7	MD48										
H'F8D8	MD51									HCAN0	8/16
H'F8D9	MD52										
H'F8DA	MD53										
H'F8DB	MD54										
H'F8DC	MD55										
H'F8DD	MD56										
H'F8DE	MD57										
H'F8DF	MD58										
H'F8E0	MD61									HCAN0	8/16
H'F8E1	MD62										
H'F8E2	MD63										
H'F8E3	MD64										
H'F8E4	MD65										
H'F8E5	MD66										
H'F8E6	MD67										
H'F8E7	MD68										
H'F8E8	MD71									HCAN0	8/16
H'F8E9	MD72										
H'F8EA	MD73										
H'F8EB	MD74										
H'F8EC	MD75										
H'F8ED	MD76										
H'F8EE	MD77										
H'F8EF	MD78										

BCR0—Bit Configuration Register **H'F802** **HCAN0**
BCR1—Bit Configuration Register **H'FA02** **HCAN1***

Bit	15	14	13	12	11	10	9	8
	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
Initial value	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

Resynchronization Jump Width

0	0	Bit synchronization width = 1 time quantum
	1	Bit synchronization width = 2 time quanta
1	0	Bit synchronization width = 3 time quanta
	1	Bit synchronization width = 4 time quanta

Baud Rate Prescaler

0	0	0	0	0	0	2 × system clock
0	0	0	0	0	1	4 × system clock
0	0	0	0	1	0	6 × system clock
:	:	:	:	:	:	:
1	1	1	1	1	1	128 × system clock

Bit	7	6	5	4	3	2	1	0
	BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8
Initial value	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

Time Segment 2

0	0	0	Setting prohibited
		1	TSEG2 = 2 time quanta
	1	0	TSEG2 = 3 time quanta
		1	TSEG2 = 4 time quanta
1	0	0	TSEG2 = 5 time quanta
		1	TSEG2 = 6 time quanta
	1	0	TSEG2 = 7 time quanta
		1	TSEG2 = 8 time quanta

Time Segment 1

0	0	0	0	Setting prohibited
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	TSEG1 = 4 time quanta
0	1	0	0	TSEG1 = 5 time quanta
:	:	:	:	:
1	1	1	1	TSEG1 = 16 time quanta

Bit Sample Point

0	Bit sampling at one point (end of time segment 1 (TSEG1))
1	Bit sampling at three points (end of time segment 1 (TSEG1) and preceding and following time quanta)

Note: * This register is not available in the H8S/2635 Group.

MC11[1]—Message Control 11[1]	**H'FA78**	**HCAN1**
MC11[2]—Message Control 11[2]	**H'FA79**	**HCAN1**
MC11[3]—Message Control 11[3]	**H'FA7A**	**HCAN1**
MC11[4]—Message Control 11[4]	**H'FA7B**	**HCAN1**
MC11[5]—Message Control 11[5]	**H'FA7C**	**HCAN1**
MC11[6]—Message Control 11[6]	**H'FA7D**	**HCAN1**
MC11[7]—Message Control 11[7]	**H'FA7E**	**HCAN1**
MC11[8]—Message Control 11[8]	**H'FA7F**	**HCAN1**

Note: These registers are not available in the H8S/2635 Group.

MC11[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
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

Data Length Code

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

MC11[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
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

MC11[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
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

MD7[1]—Message Data 7[1]	**H'FAE8**	**HCAN1**
MD7[2]—Message Data 7[2]	**H'FAE9**	**HCAN1**
MD7[3]—Message Data 7[3]	**H'FAEA**	**HCAN1**
MD7[4]—Message Data 7[4]	**H'FAEB**	**HCAN1**
MD7[5]—Message Data 7[5]	**H'FAEC**	**HCAN1**
MD7[6]—Message Data 7[6]	**H'FAED**	**HCAN1**
MD7[7]—Message Data 7[7]	**H'FAEE**	**HCAN1**
MD7[8]—Message Data 7[8]	**H'FAEF**	**HCAN1**

Note: These registers are not available in the H8S/2635 Group.

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

BRR0—Bit Rate Register 0 **H'FF79** **SCI0, Smart Card Interface 0**

Bit	7	6	5	4	3	2	1	0
Initial value	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

Set the serial transmit/receive bit rate

Note: For details see section 13.2.8, Bit Rate Register (BRR).