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	POR, PWM, WDT
Number of I/O	51
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-BFQFP
Supplier Device Package	100-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2626fa20jv

Preface

The H8S/2626 Group and H8S/2623 Group are series of high-performance microcontrollers with a 32-bit H8S/2600 CPU core, and a set of on-chip supporting modules required for system configuration.

The H8S/2600 CPU can execute basic instructions in one state, and is provided with sixteen 16-bit general registers with a 32-bit internal configuration, and a concise and optimized instruction set. The CPU can handle a 16 Mbyte linear address space (architecturally 4 Gbytes). Programs based on the high-level language C can also be run efficiently.

The address space is divided into eight areas. The data bus width and access states can be selected for each of these areas, and various kinds of memory can be connected fast and easily.

Single-power-supply flash memory (F-ZTAT™*), and mask ROM versions are available, providing a quick and flexible response to conditions from ramp-up through full-scale volume production, even for applications with frequently changing specifications.

On-chip supporting functions include a 16-bit timer pulse unit (TPU), programmable pulse generator (PPG), watchdog timer (WDT), serial communication interface (SCI), controller area network (HCAN), A/D converter, D/A converter (H8S/2626 Group only), and I/O ports.

In addition, data transfer controller (DTC) is provided, enabling high-speed data transfer without CPU intervention.

Use of the H8S/2626 Group or H8S/2623 Group enables easy implementation of compact, high-performance systems capable of processing large volumes of data.

This manual describes the hardware of the H8S/2626 Group and H8S/2623 Group. Refer to the H8S/2600 Series and H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Note: * F-ZTAT (Flexible-ZTAT) is a trademark of Renesas Technology Corp.

Table 1.3 Pin Functions in Each Operating Mode

Pin No.	Pin Name			
FP-100B	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
1	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB
2	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$
3	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC
4	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$
5	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD
6	VCC	VCC	VCC	VCC
7	HTxD	HTxD	HTxD	HTxD
8	VSS	VSS	VSS	VSS
9	HRxD	HRxD	HRxD	HRxD
10	PE0/D0	PE0/D0	PE0/D0	PE0
11	PE1/D1	PE1/D1	PE1/D1	PE1
12	PE2/D2	PE2/D2	PE2/D2	PE2
13	PE3/D3	PE3/D3	PE3/D3	PE3
14	PE4/D4	PE4/D4	PE4/D4	PE4
15	VSS	VSS	VSS	VSS
16	PE5/D5	PE5/D5	PE5/D5	PE5
17	PVCC1	PVCC1	PVCC1	PVCC1
18	PE6/D6	PE6/D6	PE6/D6	PE6
19	PE7/D7	PE7/D7	PE7/D7	PE7
20	D8	D8	D8	PD0
21	D9	D9	D9	PD1
22	D10	D10	D10	PD2
23	D11	D11	D11	PD3
24	D12	D12	D12	PD4
25	D13	D13	D13	PD5
26	D14	D14	D14	PD6

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Mode

The CPU supports the eight addressing modes listed in table 2.4. 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 2.4 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

(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) 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).

8.2 Register Descriptions

8.2.1 DTC Mode Register A (MRA)

Bit	:	7	6	5	4	3	2	1	0
		SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	—	—	—

MRA is an 8-bit register that controls the DTC operating mode.

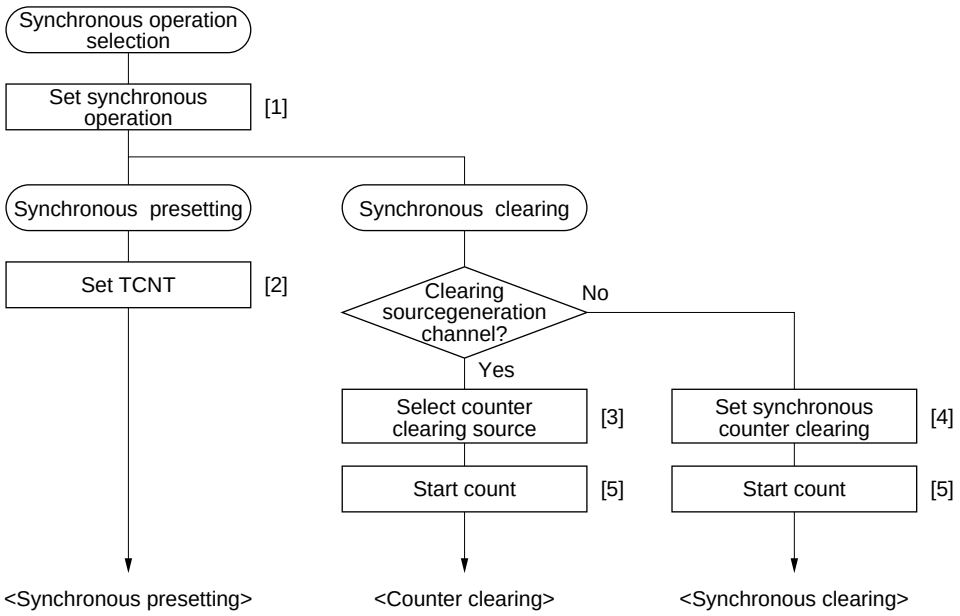
Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whether SAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 7	Bit 6	
SM1	**SM0**	**Description**
0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 5 and 4—Destination Address Mode 1 and 0 (DM1, DM0): These bits specify whether DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5	Bit 4	
DM1	**DM0**	**Description**
0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Example of Synchronous Operation Setting Procedure: Figure 10.14 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.14 Example of Synchronous Operation Setting Procedure

Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 13.9.

When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 13.3.

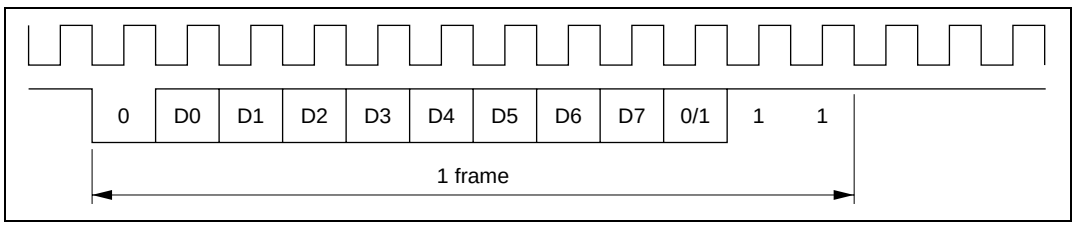


Figure 13.3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations

SCI initialization (asynchronous mode): Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

When an external clock is used the clock should not be stopped during operation, including initialization, since operation is uncertain.

Figure 13.4 shows a sample SCI initialization flowchart.

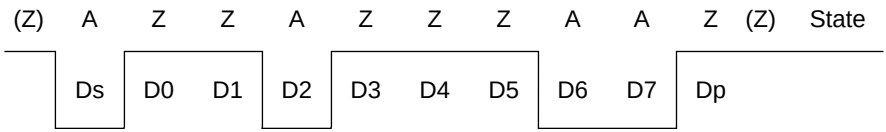
Smart Card Mode Register (SCMR) Setting: The SDIR bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SINV bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SMIF bit is set to 1 in the case of the Smart Card interface.

Examples of register settings and the waveform of the start character are shown below for the two types of IC card (direct convention and inverse convention).

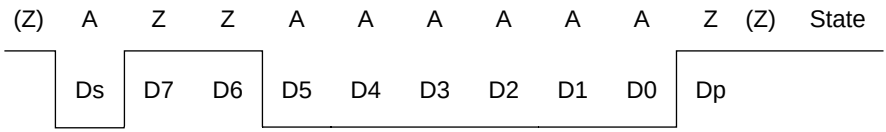
- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



  With the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B.

  The parity bit is 1 since even parity is stipulated for the Smart Card.

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



  With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F.

  The parity bit is 0, corresponding to state Z, since even parity is stipulated for the Smart Card.

  With the H8S/2626 Group and H8S/2623 Group, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to odd parity mode (the same applies to both transmission and reception).

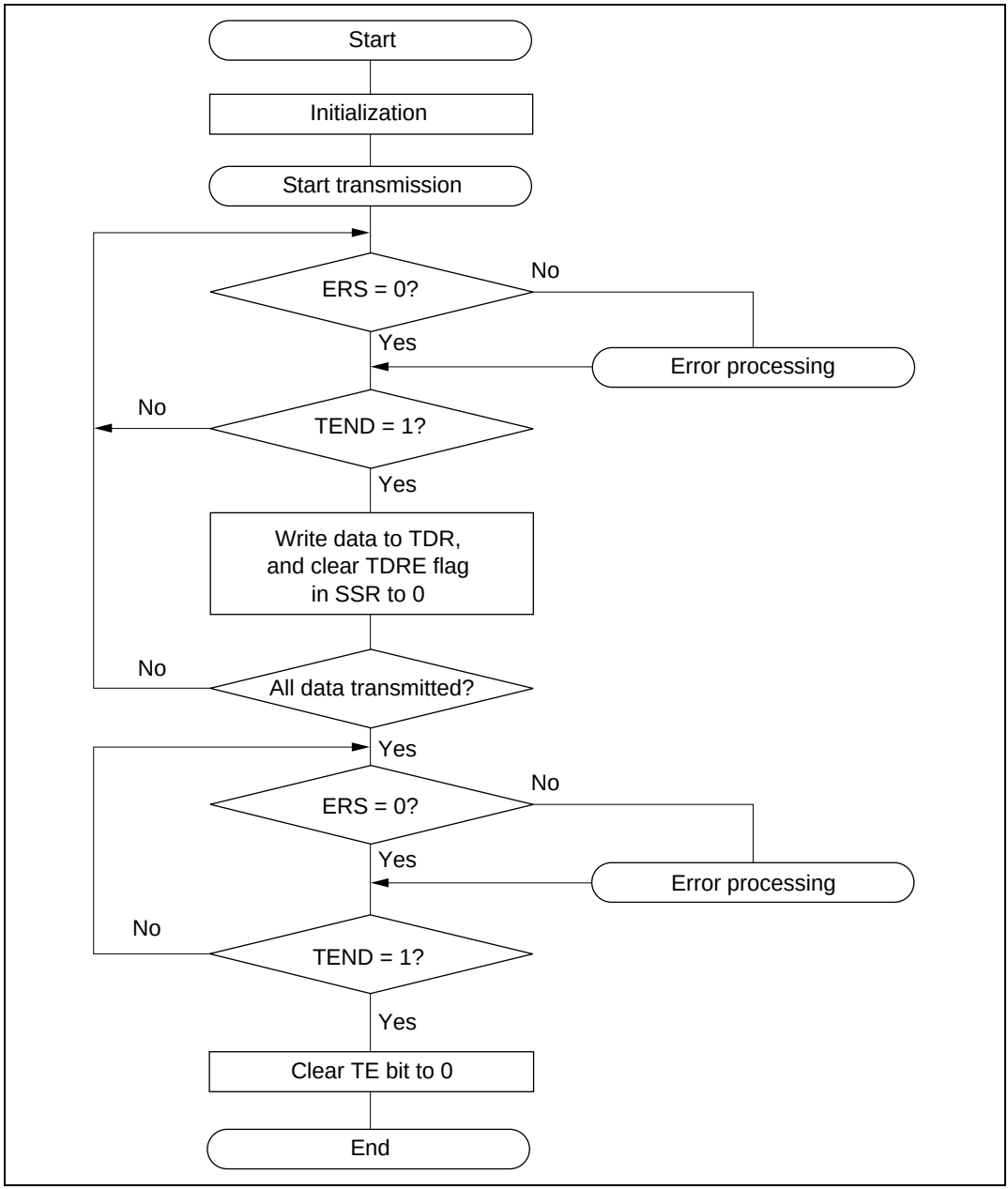


Figure 14.4 Example of Transmission Processing Flow

15.2.18 Message Control (MC0 to MC15)

The message control register sets (MC0 to MC15) consist of eight 8-bit readable/writable registers (MCx[1] to MCx[8]). The HCAN has 16 sets of these registers (MC0 to MC15).

The initial value of these registers is undefined, so they must be initialized (by writing 0 or 1).

MCx [1]

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [2]

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

MCx [3]

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

MCx [4]

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

Setting a bit to 1 in the mailbox configuration register (MBCR) designates the corresponding mailbox for reception use. When setting mailboxes for reception, to improve message transmission efficiency, high-priority messages should be set in low-to-high mailbox order (priority order: mailbox 1 (MCx[1]) > mailbox 15 (MCx[15]).

- Receive-only mailbox (mailbox 0)

  No setting is necessary, as this mailbox is always used for reception.

Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Settings: After power is supplied, all registers and RAM (message control/data, control registers, status registers, etc.) are initialized. Message control/data (MCx[x], MDx[x]) only are in RAM, and so their values are undefined. Initial values must therefore be set in all the mailboxes (by writing 0s or 1s).

Setting the Message Transmission Method: Either of the following message transmission methods can be selected with the message transmission method bit (MCR2) in the master control register (MCR):

a. Transmission order determined by message identifier priority
b. Transmission order determined by mailbox number priority

When a is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), the message with the highest priority set in the message identifier (MCx[5]–MCx[8]) is stored in the transmit buffer. CAN bus arbitration is then carried out for the message in the transmit buffer, and message transmission is performed when the transmission right is acquired. When the TXPR bit is set, internal arbitration is performed again, and the highest-priority message is found and stored in the transmit buffer.

When b is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), messages are stored in the transmit buffer in low-to-high mailbox order (priority order: mailbox 1 > mailbox 15). CAN bus arbitration is then carried out for the messages in the transmit buffer, and message transmission is performed when the bus is acquired.

15.3.6 HCAN Halt Mode

The HCAN halt mode is provided to enable mailbox settings to be changed without performing an HCAN hardware or software reset. Figure 15.12 shows a flowchart of the HCAN halt mode.

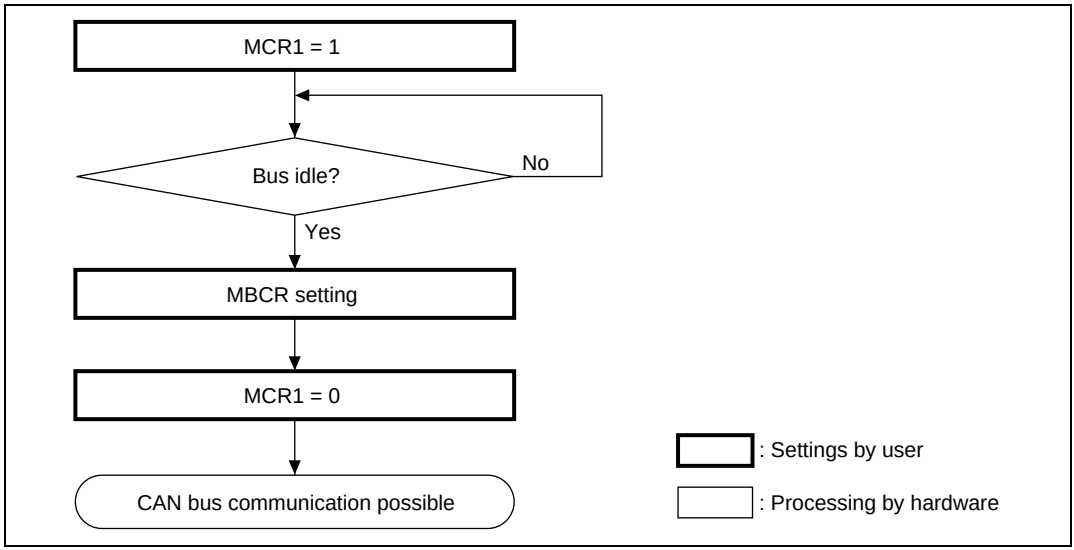


Figure 15.12 HCAN Halt Mode Flowchart

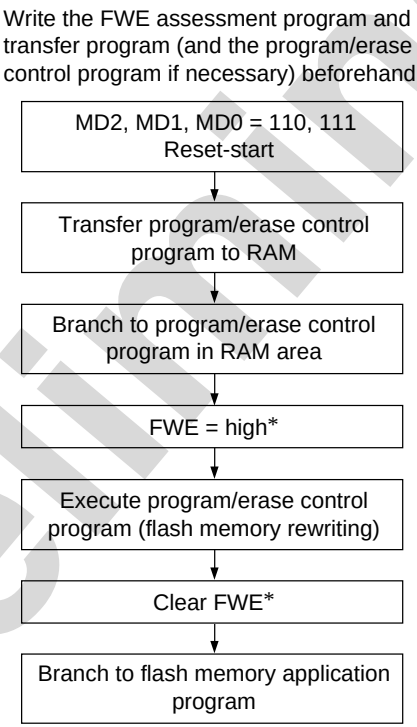
HCAN halt mode is entered by setting the halt request bit (MCR1) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN halt mode is delayed until the bus becomes idle.

HCAN halt mode is cleared by clearing MCR1 to 0.

To select user program mode, select a mode that enables the on-chip flash memory (mode 6 or 7), and apply a high level to the FWE pin. In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 6 and 7.

The flash memory itself cannot be read while the SWE1 bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip RAM or external memory. If the program is to be located in external memory, the instruction for writing to flash memory, and the following instruction, should be placed in on-chip RAM.

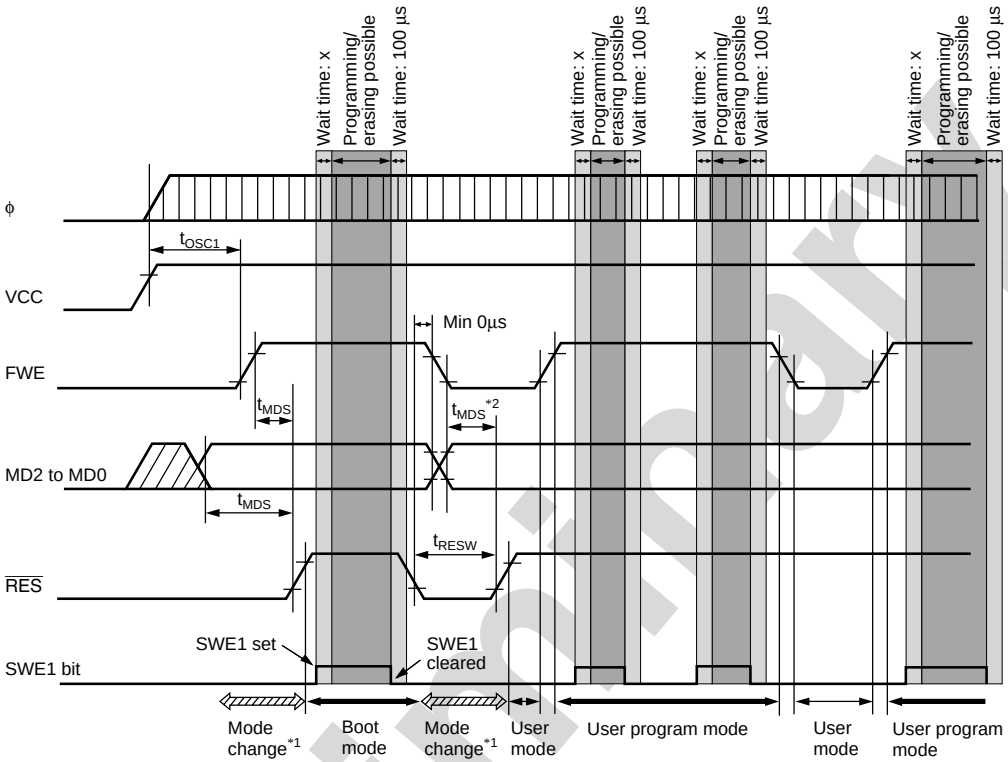
Figure 19.9 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.



Notes: Do not apply a constant high level to the FWE pin. Apply a high level to the FWE pin only when the flash memory is programmed or erased. 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.

* For further information on FWE application and disconnection, see section 19.13, Flash Memory Programming and Erasing Precautions.

Figure 19.9 User Program Mode Execution Procedure



Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of $\overline{RES}$ input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), and therefore these pins should not be used as output signals during this time.

2. When making a transition from boot mode to another mode, a mode programming setup time, t_{MDS} (min), of 200 ns is necessary with respect to the $\overline{RES}$ clearance timing.

3. See section 22.6, Flash Memory Characteristics.

Figure 19.28 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

Table 21A.1 LSI Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instructions Registers	Functioning	Medium-speed operation	Halted (retained)	High/medium-speed operation	Halted (retained)	Halted (undefined)
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0–IRQ5						
Peripheral functions	WDT0	Functioning	Functioning	Functioning	—	Halted (retained)	Halted (reset)
	DTC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	TPU	Functioning	Functioning (PBC medium-speed operation)	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	PBC						
	PPG						
	SCI0	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	SCI1						
	SCI2						
	PWM						
	A/D						
	RAM	Functioning	Functioning	Functioning (DTC)	Functioning	Retained	Retained
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	High impedance
	HCAN	Functioning	Functioning*	Functioning	Halted (reset)	Halted (reset)	Halted (reset)

Notes: “Halted (retained)” means that internal register values are retained. The internal state is “operation suspended.”

“Halted (reset)” means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

* Note, however, that registers cannot be read or written to.

21A.1.1 Register Configuration

Power-down modes are controlled by the SBYCR, SCKCR, LPWRCR, and MSTPCR registers. Table 21A.2 summarizes these registers.

Table 21A.2 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'08	H'FDE4
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low power control register	LPWRCR	R/W	H'00	H'FDEC
Module stop control register A, B, C	MSTPCRA	R/W	H'3F	H'FDE8
	MSTPCRB	R/W	H'FF	H'FDE9
	MSTPCRC	R/W	H'FF	H'FDEA

Note: * Lower 16 bits of the address.

21A.2 Register Descriptions

21A.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1	0
		SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	:	0	0	0	0	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—	—

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

SBYCR is initialized to H'08 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Software Standby (SSBY): When making a low power dissipation mode transition by executing the SLEEP instruction, the operating mode is determined in combination with other control bits.

Note that the value of the SSBY bit does not change even when shifting between modes using interrupts.

22.2 DC Characteristics

Table 22.2 lists the DC characteristics. Table 22.3 lists the permissible output currents.

Table 22.2 DC Characteristics — Preliminary —

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ0}$ to $\overline{IRQ5}$	V_T^-	$PV_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$PV_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$PV_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE	V_{IH}	$PV_{CC} \times 0.9$	—	PV_{CC} + 0.3	V	
	EXTAL, OSC1		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 1, A to F, HRxD		$PV_{CC} \times 0.7$	—	PV_{CC} + 0.3	V	
	Port 4 and 9		$AV_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE	V_{IL}	–0.3	—	$PV_{CC} \times 0.1$	V	
	EXTAL, OSC1		–0.3	—	$V_{CC} \times 0.2$	V	
	Ports 1, A to F, HRxD		–0.3	—	$PV_{CC} \times 0.2$	V	
	Ports 4 and 9		–0.3	—	$AV_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	PV_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			PV_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to PV_{CC} – 0.5 V
	$\overline{STBY}$, NMI, HRxD, MD2 to MD0, FWE		—	—	1.0	µA	
	Ports 4 and 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Instruction	H	N	Z	V	C	Definition
CLRMAC	—	—	—	—	—	
CMP	↕	↕	↕	↕	↕	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
DAA	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
INC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC	—	—	—	—	—	
MAC	—	—	—	—	—	

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'FE12	ISCRH	—	—	—	—	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	Interrupt controller	8
H'FE13	ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA		
H'FE14	IER	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E		
H'FE15	ISR	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F		
H'FE16	DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC	8
H'FE17	DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0		
H'FE18	DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0		
H'FE19	DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0		
H'FE1A	DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0		
H'FE1B	DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0		
H'FE1C	DTCERG	DTCEG7	DTCEG6	DTCEG5	DTCEG4	DTCEG3	DTCEG2	DTCEG1	DTCEG0		
H'FE1F	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0		
H'FE26	PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG	8
H'FE27	PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV		
H'FE28	NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8		
H'FE29	NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0		
H'FE2A	PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8		
H'FE2B	PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0		
H'FE2C	NDRH*2	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8		
H'FE2D	NDRL*2	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0		
H'FE2E	NDRH*2	—	—	—	—	NDR11	NDR10	NDR9	NDR8		
H'FE2F	NDRL*2	—	—	—	—	NDR3	NDR2	NDR1	NDR0		
H'FE30	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	I/O port	8
H'FE39	PADDR	—	—	PA5DDR*5	PA4DDR*5	PA3DDR	PA2DDR	PA1DDR	PA0DDR		
H'FE3A	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR		
H'FE3B	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR		
H'FE3C	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR		
H'FE3D	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR		
H'FE3E	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR		
H'FE40	PAPCR	—	—	PA5PCR*5	PA4PCR*5	PA3PCR	PA2PCR	PA1PCR	PA0PCR		
H'FE41	PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR		
H'FE42	PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR		
H'FE43	PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR		
H'FE44	PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR		
H'FE47	PAODR	—	—	PA5ODR*5	PA4ODR*5	PA3ODR	PA2ODR	PA1ODR	PA0ODR		
H'FE48	PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR		
H'FE49	PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR		

REC—Receive Error Counter **H'F818** **HCAN**

REC

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R	R

TEC—Transmit Error Counter **H'F819** **HCAN**

TEC

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R	R

UMSR—Unread Message Status Register **H'F81A** **HCAN**

UMSR

Bit	:	15	14	13	12	11	10	9	8
		UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0
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)*

Bit	:	7	6	5	4	3	2	1	0
		UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8
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)*

Unread message status flags

0	[Clearing condition] Writing 1
1	Unread receive message is overwritten by a new message [Setting condition] When a new message is received before RXPR is cleared

Note: * Can only be written with 1 for flag clearing.