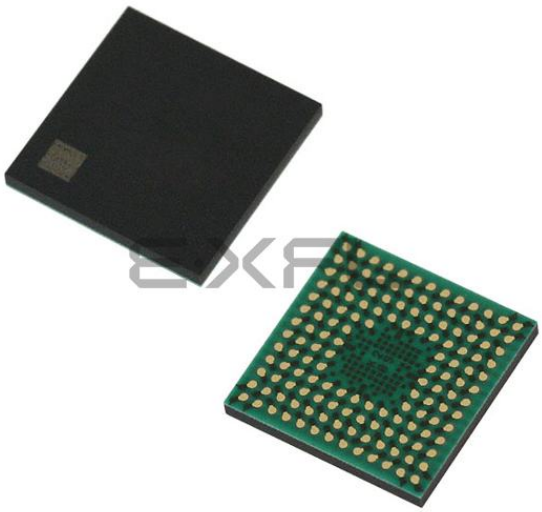


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	H8S/2600
Core Size	16-Bit
Speed	20MHz
Connectivity	FIFO, I²C, LPC, SCI, SmartCard
Peripherals	POR, PWM, WDT
Number of I/O	112
Program Memory Size	160KB (160K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	145-TFLGA
Supplier Device Package	145-TFLGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2117rvlp20dhv

2. Description of Numbers and Symbols

Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(1) Overall notation

In descriptions involving the names of bits and bit fields within this manual, the modules and registers to which the bits belong may be clarified by giving the names in the forms "module name"."register name"."bit name" or "register name"."bit name".

(2) Register notation

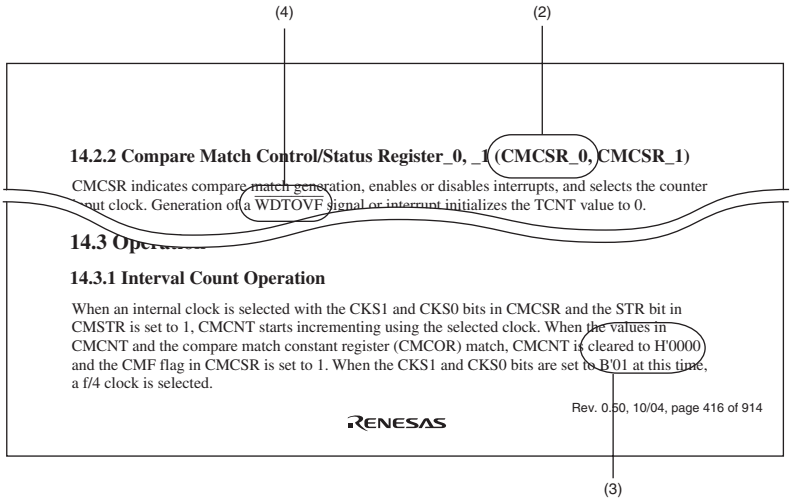
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.
[Examples] Binary: B'11 or 11
Hexadecimal: H'EFA0 or 0xEFA0
Decimal: 1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.
[Example] $\overline{\text{WDTOVF}}$



Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode Mode 2 (EXPE = 0)
—	A3	—	NC
141	D4	B3	PH4
142	B3	C4	PH5
143	A2	A3	XTAL
144	B2	A2	EXTAL

Notes: (N) in Pin No. indicates the pin is driven by NMOS push-pull/open drain and has 5 V input tolerance.

(T) in Pin No. indicates the pin has 5 V input tolerance.

* This pin is not supported by the system development tool (emulator).

5.3.5 IRQ Enable Registers (IER16, IER)

The IER registers enable and disable interrupt requests IRQ15 to IRQ0.

- IER16

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15E	0	R/W	IRQn Enable
6	IRQ14E	0	R/W	The IRQn interrupt request is enabled when this bit is 1.
5	IRQ13E	0	R/W	
4	IRQ12E	0	R/W	(n = 15 to 8)
3	IRQ11E	0	R/W	
2	IRQ10E	0	R/W	
1	IRQ9E	0	R/W	
0	IRQ8E	0	R/W	

- IER

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQn Enable
6	IRQ6E	0	R/W	The IRQn interrupt request is enabled when this bit is 1.
5	IRQ5E	0	R/W	
4	IRQ4E	0	R/W	(n = 7 to 0)
3	IRQ3E	0	R/W	
2	IRQ2E	0	R/W	
1	IRQ1E	0	R/W	
0	IRQ0E	0	R/W	

5.6.4 Interrupt Response Times

Table 5.10 shows interrupt response times – the intervals between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

Table 5.10 Interrupt Response Times

No.	Execution Status	Advanced Mode
1	Interrupt priority determination*[1]	3
2	Number of wait states until executing instruction ends*[2]	1 to 21
3	Saving of PC and CCR in stack	2
4	Vector fetch	2
5	Instruction fetch*[3]	2
6	Internal processing*[4]	2
	Total (using on-chip memory)	12 to 32

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and prefetch of interrupt handling routine.
4. Internal processing after interrupt acceptance and internal processing after vector fetch.

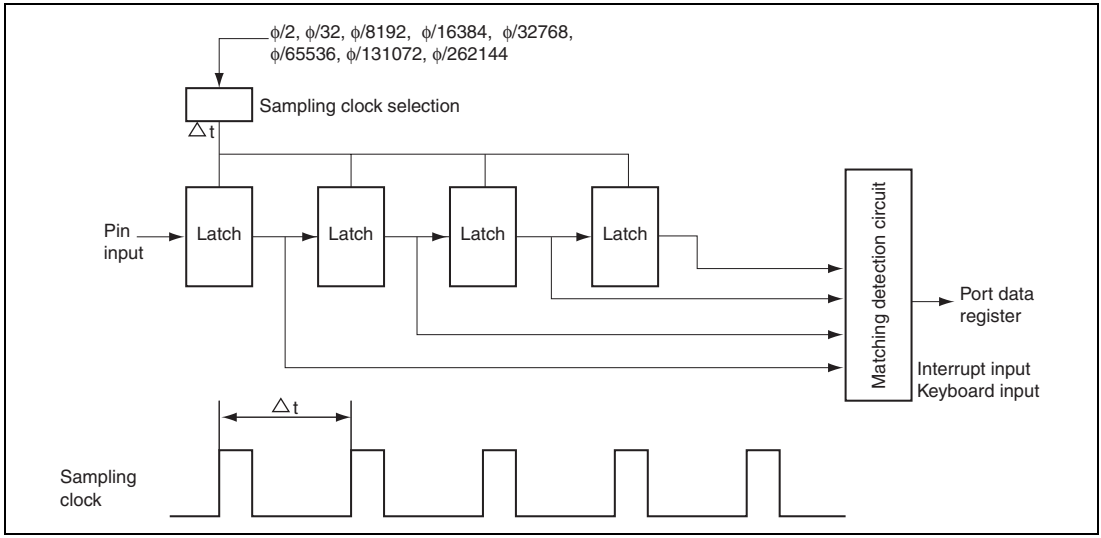


Figure 7.1 Noise Cancel Circuit

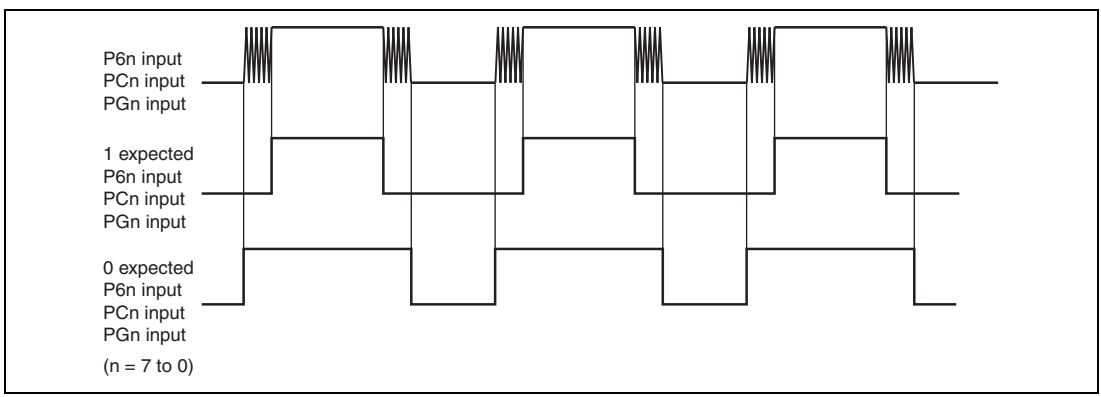


Figure 7.2 Schematic View of Noise Cancel Operation

9.2 Input/Output Pins

Table 9.1 lists the PWMX (D/A) module input and output pins.

Table 9.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
PWMX output pin 0	PWX0	Output	PWMX output of channel A
PWMX output pin 1	PWX1	Output	PWMX output of channel B

9.3 Register Descriptions

The PWMX (D/A) module has the following registers. The PWMX (D/A) registers are assigned to the same addresses with other registers. The registers are selected by the IICE bit in the serial timer control register (STCR). For details on the module stop control register, see section 26.1.3, Module Stop Control Registers H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB).

Table 9.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
PWMX (D/A) counter H	DACNTH	R/W	H'00	H'FFA6 H'FEA6*	8
PWMX (D/A) counter L	DACNTL	R/W	H'03	H'FFA7 H'FEA7*	8
PWMX (D/A) data register AH	DADRAH	R/W	H'FF	H'FFA0 H'FEA0*	8
PWMX (D/A) data register AL	DADRAL	R/W	H'FF	H'FFA1 H'FEA1*	8
PWMX (D/A) data register BH	DADRBH	R/W	H'FF	H'FFA6 H'FEA6*	8
PWMX (D/A) data register BL	DADRBL	R/W	H'FF	H'FFA7 H'FEA7*	8
PWMX (D/A) control register	DACR	R/W	H'30	H'FFA0 H'FEA0*	8
Peripheral clock select register	PCSR	R/W	H'00	H'FF82	8

Notes: The same addresses are shared by DADRA and DACR, and by DADRB and DACNT. Switching is performed by the REGS bit in DACNT or DADRB.

* Upper address: when RELOCATE = 0
Lower address: when RELOCATE = 1

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge. Rising edge, falling edge, or both edges can be selected as the detected edge.

(a) Example of input capture operation setting procedure

Figure 10.12 shows an example of the input capture operation setting procedure.

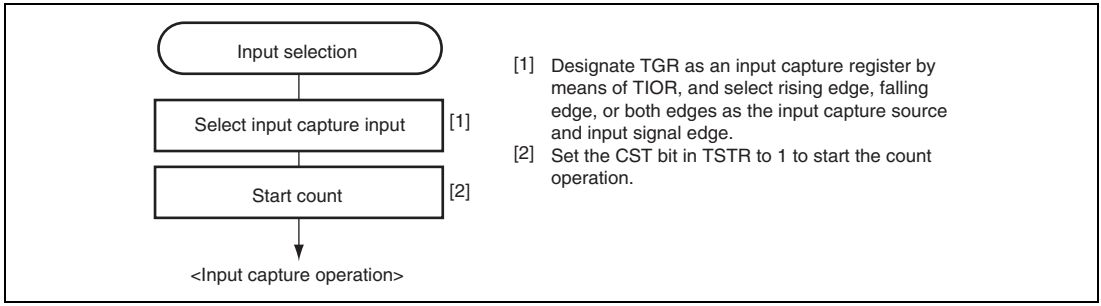


Figure 10.12 Example of Input Capture Operation Setting Procedure

(2) Measuring a Cycle

In cycle measurement mode, one cycle of the TDP input waveform forms one measurement cycle. Start by setting TDPMDS = 0 and CST = 0, which clears TDPCNT to H'0000. Next, set the upper limit and lower limit values of the measurement pulse width in TDPWDMX and TDPWDMN, and set the upper limit and lower limit values of the measurement cycle in the TDPPDMX and TDPPDMN. Finally, place the TDP in cycle measurement mode by setting the TDPMDS bit in TDPCR1 to 1. TDPCNT will count up cycles of the selected clock. When the first edge (either rising or falling, as selected by the POCTL bit in TDPCR1) of the measurement cycle is detected, TDPCNT is automatically cleared to H'0000. When the second edge is detected, the value in TDPCNT is transferred to TDPICR. At this time, the value in TDPICR is compared with the values in TDPWDMX and TDPWDMN. If TDPIR is greater than TDPWDMX or less than TDPWDMN, the TWDMXOVF or TWDMNUDF flag, respectively, in TDPCSR is set to 1. When the third edge is detected, the value in TDPCNT is transferred to TDPICR. At this time, the value in TDPICR is compared with the values in TDPPDMX and TDPPDMN. If TDPICR is greater than TDPPDMX or less than TDPPDMN, the TPDMXOVF or TPDMNUDF flag, respectively, in TDPCSR is set to 1. Generation of the corresponding interrupt request is enabled by the setting in TDPIER. Also, when the third edge is detected, TDPCNT is cleared to H'0000, and the next round of measurement starts.

When the CPSPE bit in TDPCR1 is cleared to 0, the next round of cycle measurement will start regardless of whether any of these flags is set to 1.

If any of these flags is set to 1 while the CPSPE bit in TDPCR1 is set to 1, counting up by TDPCNT stops and cycle measurement also stops. Subsequently clearing the corresponding flag to 0 automatically clears TDPCNT to H'0000, and counting up for cycle measurement is restarted.

Clocked Synchronous Mode:

- Data length: 8 bits
- Receive error detection: Overrun errors

Smart Card Interface:

- An error signal can be automatically transmitted on detection of a parity error during reception.
- Data can be automatically re-transmitted on detection of an error signal during transmission.
- Both direct convention and inverse convention are supported.

Figure 15.1 shows a block diagram of SCI.

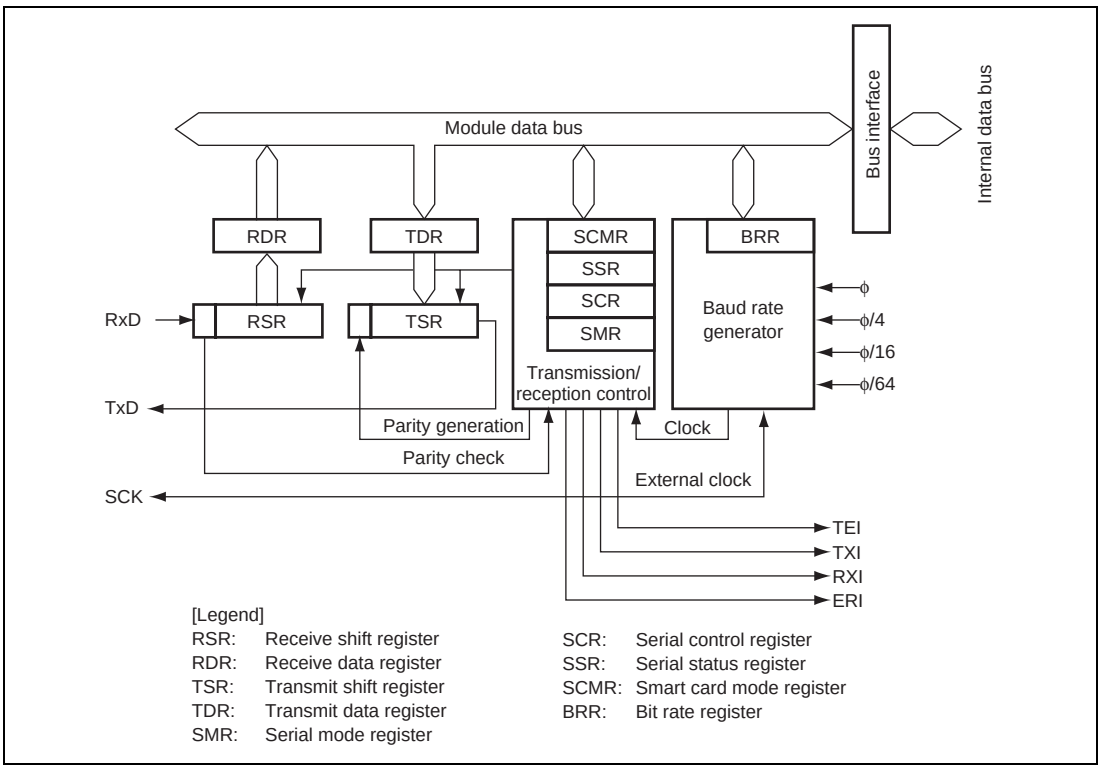


Figure 15.1 Block Diagram of SCI

15.5.2 Multiprocessor Serial Data Reception

Figure 15.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 15.12 shows an example of SCI operation for multiprocessor format reception.

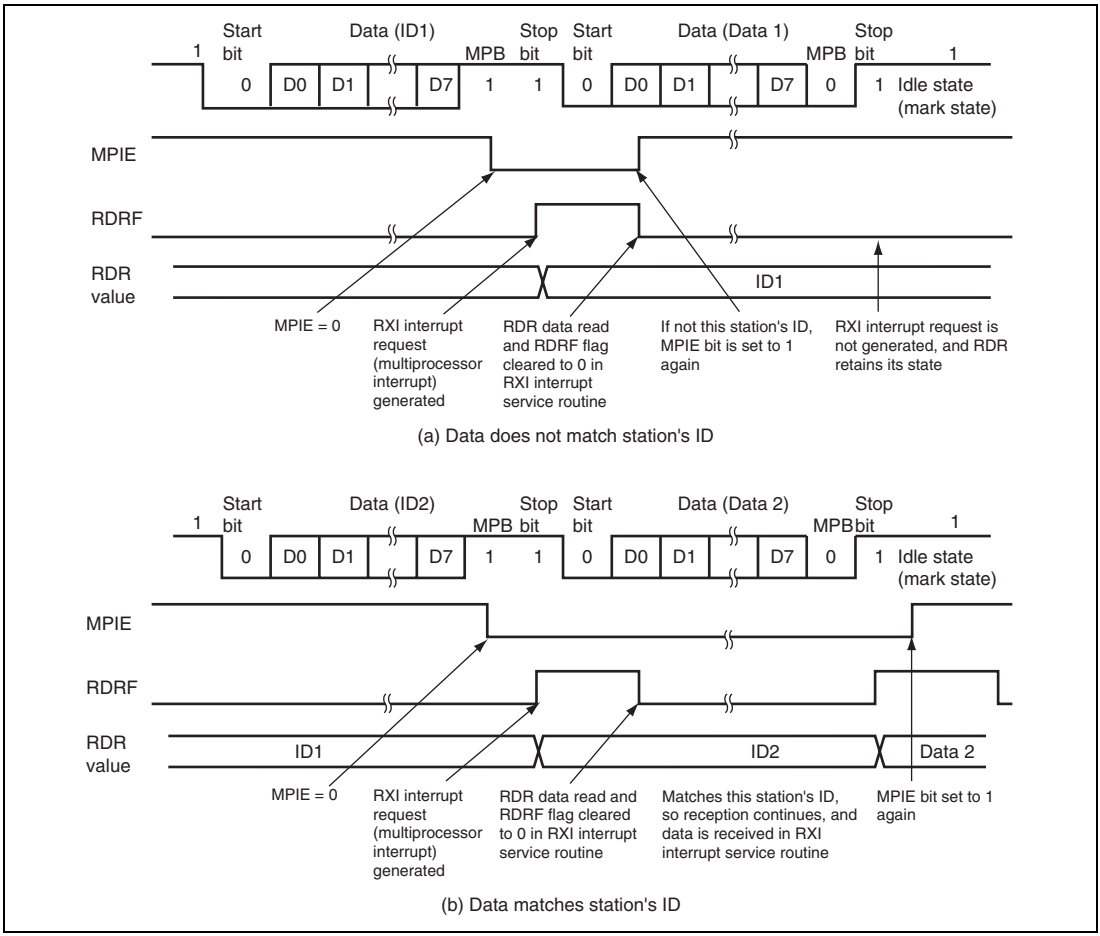


Figure 15.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

15.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 15.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 15.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB-first as the start character, as shown in figure 15.23. Therefore, data in the start character in the figure is H'3B. When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the O/$\overline{E}$ bit in SMR in order to use even parity, which is prescribed by the smart card standard.

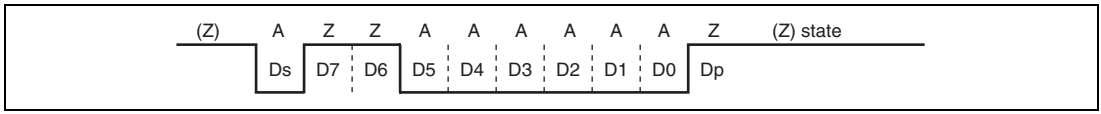


Figure 15.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB-first as the start character, as shown in figure 15.24. Therefore, data in the start character in the figure is H'3F. When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of this LSI only inverts data bits D7 to D0, write 1 to the O/$\overline{E}$ bit in SMR to invert the parity bit in both transmission and reception.

15.7.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- If a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time after the end of the parity bit before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in normal smart card interface mode, the flag is always read as 0 because no error signal is transferred.

17.4 Operation

17.4.1 Baud Rate

The SCIF includes a baud rate generator and can set the desired baud rate using registers FDLH, FDLL, and the CKSEL bit in SCIFCR. Table 17.6 shows an example of baud rate settings.

Table 17.6 Example of Baud Rate Settings

	00		01		01	
CKSEL1, CKSEL0	LCLK (33 MHz) divided by 18		System Clock (20 MHz) divided by 11		System Clock (10 MHz) divided by 11	
Baud rate	FDLH, FDLL (Hex)	Error (%)	FDLH, FDLL (Hex)	Error (%)	FDLH, FDLL (Hex)	Error (%)
50	0900	0.54 %	0900	1.36 %	0480	1.36 %
75	0600	0.54 %	0600	1.36 %	0300	1.36 %
110	0417	0.54 %	0417	1.36 %	—	—
300	0180	0.54 %	0180	1.36 %	00C0	1.36 %
600	00C0	0.54 %	00C0	1.36 %	0060	1.36 %
1200	0060	0.54 %	0060	1.36 %	0030	1.36 %
1800	0040	0.54 %	0040	1.36 %	0020	1.36 %
2400	0030	0.54 %	0030	1.36 %	0018	1.36 %
4800	0018	0.54 %	0018	1.36 %	000C	1.36 %
9600	000C	0.54 %	000C	1.36 %	0006	1.36 %
14400	0008	0.54 %	0008	1.36 %	0004	1.36 %
19200	0006	0.54 %	0006	1.36 %	0003	1.36 %
38400	0003	0.54 %	0003	1.36 %	—	—
57600	0002	0.54 %	0002	1.36 %	0001	1.36 %
115200	0001	0.54 %	0001	1.36 %	—	—

(b) Programming

FPFR indicates the return value of the programming result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Programming Mode Related Setting Error Detect Detects the error protection state and returns the result. When the error protection state is entered, this bit is set to 1. Whether the error protection state is entered or not can be confirmed with the FLER bit in FCCS. For conditions to enter the error protection state see section 24.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming cannot be performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data could not be written because the user MAT was not erased. If this bit is set to 1, there is a high possibility that the user MAT has been written to partially. In this case, after removing the error factor, erase the user MAT. Also an attempt to write the user MAT when the FMATS value is H'AA and the user boot MAT is selected leads to a programming execution error. In that case, both the user MAT and user boot MAT are not rewritten. Writing to the user boot MAT must be performed in boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (programming result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'5A) before programming starts, and returns the result. 0: FKEY setting is normal (H'5A) 1: FKEY setting is abnormal (value other than H'5A)
3	—	—	—	Unused Returns 0.

(2) State Transition Diagram

The state transition after boot mode is initiated is shown in figure 24.8.

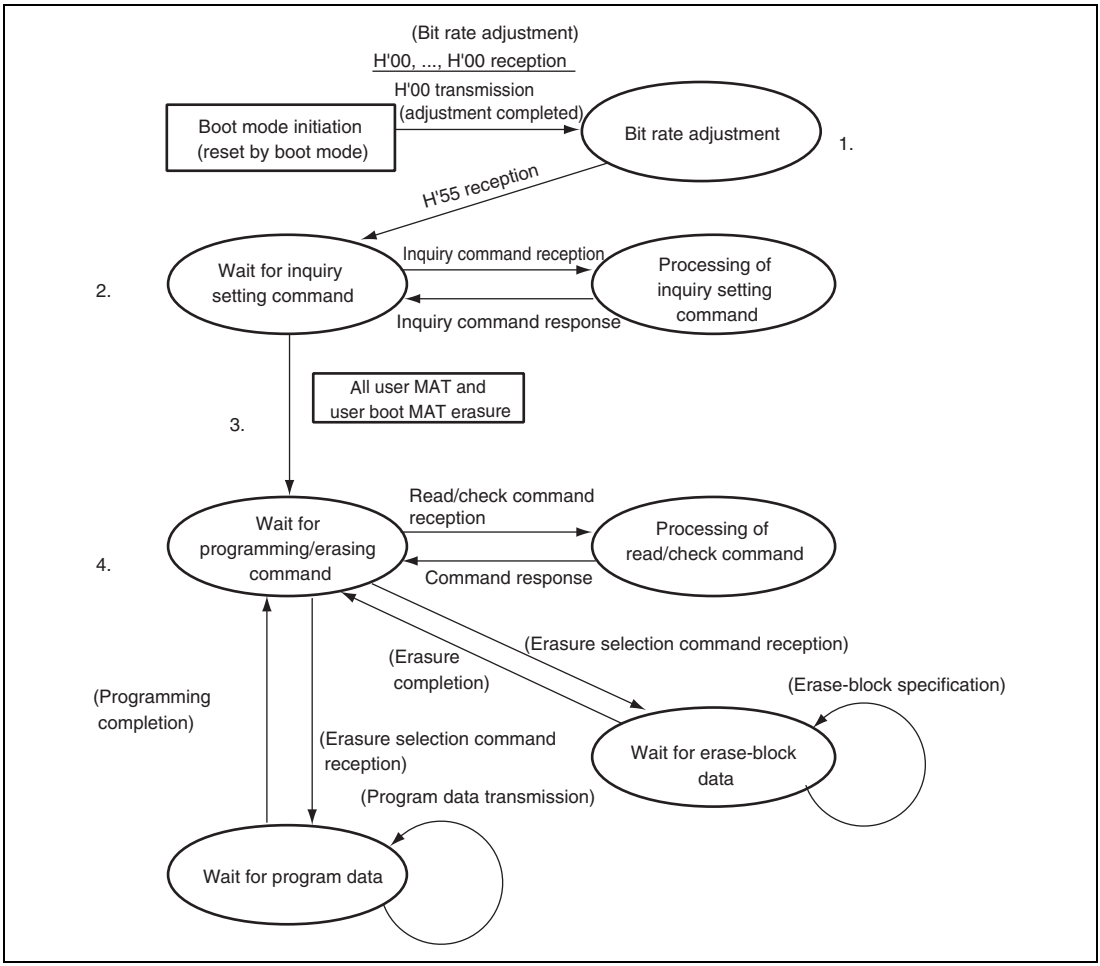


Figure 24.8 Boot Mode State Transition Diagram

(2) Programming Procedure in User Program Mode

The procedures for download of the on-chip program, initialization, and programming are shown in figure 24.12.

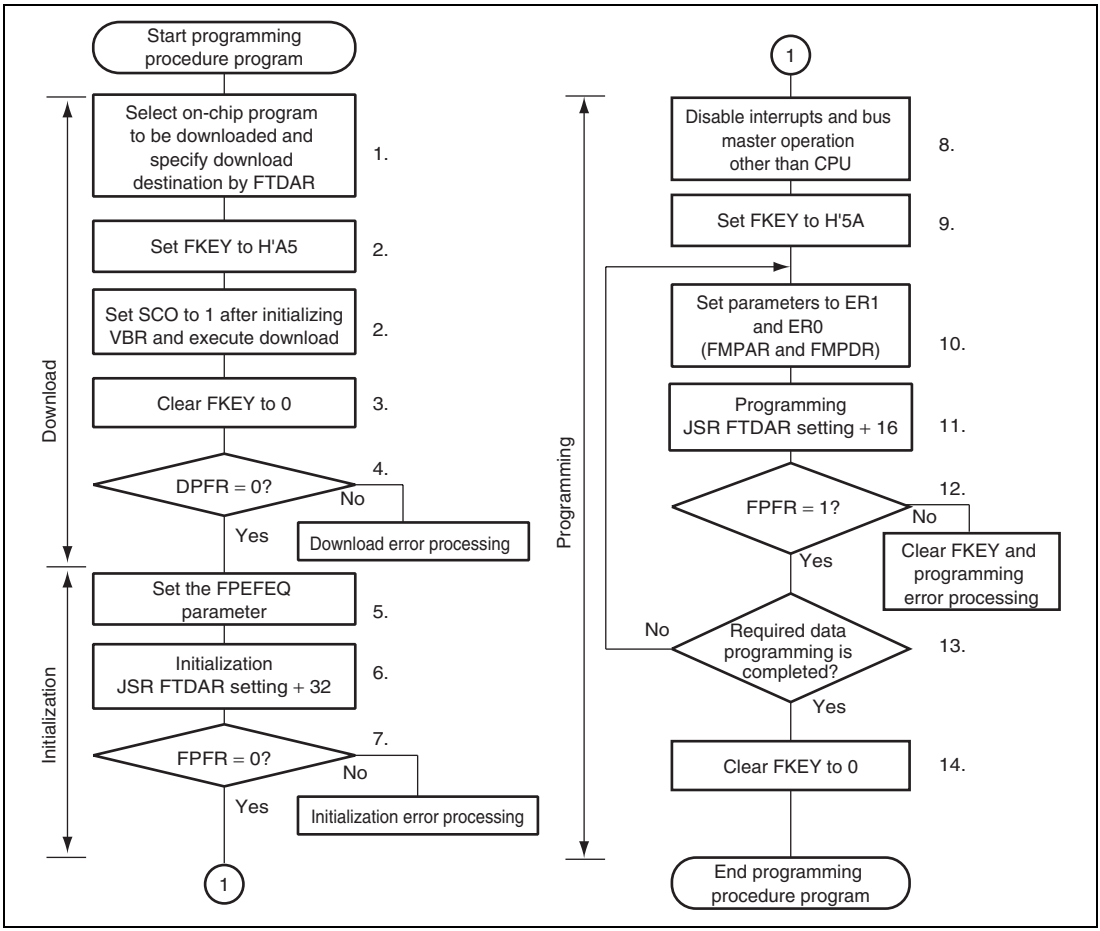


Figure 24.12 Programming Procedure in User Program Mode

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCMCNT_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TCM_3
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMMLCM_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICR_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICRF_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMCSR_3	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—	
TCMCR_3	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0	
TCMIER_3	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—	
TCMMINCM_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRA	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	A/D converter
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRB	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRC	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRD	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRE	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRF	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRG	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRH	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PWMREG0_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	PWMU_A
PWMPRE0_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG1_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE1_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG2_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE2_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG3_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE3_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG4_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE4_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG5_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE5_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMCONA_A	CLK1	CLK0	—	—	—	—	—	—	
PWMCONB_A	—	—	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E	
PWMCONC_A	—	CNTMD01	PWMSL5	PWMSL4	PWMSL3	PWMSL2	PWMSL1	PWMSL0	
PWMCOND_A	PH5S	PH4S	PH3S	PH2S	PH1S	PH0S	CNTMD45	CNTMD23	
PWMREG0_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	PWMU_B
PWMPRE0_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG1_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE1_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG2_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE2_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG3_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE3_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG4_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE4_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG5_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE5_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMCONA_B	CLK1	CLK0	—	—	—	—	—	—	
PWMCONB_B	—	—	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E	
PWMCONC_B	—	CNTMD01	PWMSL5	PWMSL4	PWMSL3	PWMSL2	PWMSL1	PWMSL0	
PWMCOND_B	PH5S	PH4S	PH3S	PH2S	PH1S	PH0S	CNTMD45	CNTMD23	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
DACR	Initialized	—	Initialized	—	Initialized	Initialized	PWMX
DADRA	Initialized	—	Initialized	—	Initialized	Initialized	
DADRB	Initialized	—	Initialized	—	Initialized	Initialized	
DACNT	Initialized	—	Initialized	—	Initialized	Initialized	
FCCS	Initialized	—	—	—	—	—	ROM
FPCS	Initialized	—	—	—	—	—	
FECS	Initialized	—	—	—	—	—	
FKEY	Initialized	—	—	—	—	—	
FMATS	Initialized	—	—	—	—	—	
FTDAR	Initialized	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	TPU common
TSYR	Initialized	—	—	—	—	—	
KBCR1_0	Initialized	—	—	—	—	—	PS2
KBTR_0	Initialized	—	—	—	—	—	
KBCR1_1	Initialized	—	—	—	—	—	
KBTR_1	Initialized	—	—	—	—	—	
KBCR1_2	Initialized	—	—	—	—	—	
KBTR_2	Initialized	—	—	—	—	—	
TCRXY	Initialized	—	—	—	—	—	TMR_XY
TCR_Y	Initialized	—	—	—	—	—	TMR_Y
TCSR_Y	Initialized	—	—	—	—	—	
TCORA_Y	Initialized	—	—	—	—	—	
TCORB_Y	Initialized	—	—	—	—	—	
TCNT_Y	Initialized	—	—	—	—	—	
ICDR_1	—	—	—	—	—	—	IIC_1
SARX_1	Initialized	—	—	—	—	—	
ICMR_1	Initialized	—	—	—	—	—	
SAR_1	Initialized	—	—	—	—	—	
ICCR_1	Initialized	—	—	—	—	—	
ICSR_1	Initialized	—	—	—	—	—	

D. Treatment of Unused Pins

The treatments of unused pins are listed in table D.1.

Table D.1 Treatment of Unused Pins

Pin Name	Example of Pin Treatment
$\overline{\text{RES}}$	(Always used as a reset pin)
$\overline{\text{ETRST}}$	(Always used as a reset pin)
MD2, MD1	(Always used as mode pins)
NMI	• Connect to V_{CC} via a pull-up resistor
EXTAL	(Always used as a clock pin)
XTAL	(Always used as a clock pin)
Port 1 Port 2 Port 3 Port 4 Port 5 Port 6 Port 8 Port 9 Port A Port B Port C Port D Port F Port G Port H Port I Port J	• Connect each pin to V_{CC} via a pull-up resistor or to V_{SS} via a pull-down resistor
Port 7	• Connect each pin to AV_{CC} via a pull-up resistor or to AV_{SS} via a pull-down resistor
Port E	• Connect each pin to V_{CC} via a pull-up resistor