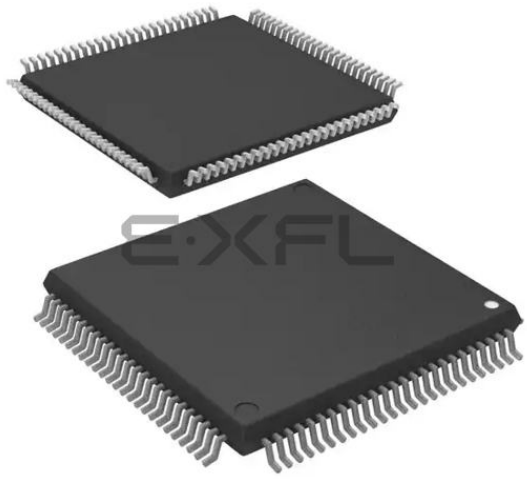

Welcome to E-XFL.COM

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	H8/300H
Core Size	16-Bit
Speed	10MHz
Connectivity	I²C, IrDA, SCI
Peripherals	POR, PWM, WDT
Number of I/O	75
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LFQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df38799fp10wv

Table 2.4 Logic Operations Instructions

Instruction	Size*	Function
AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement (logical complement) of general register contents.

Note: * Refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.5 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents.
SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents.
ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents.
ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag.

Note: * Refers to the operand size.
B: Byte
W: Word
L: Longword

4.7.2 Instructions that Disable Interrupts

The instructions that disable interrupts are LDC, ANDC, ORC, and XORC.

When an interrupt request is generated, an interrupt is requested to the CPU after the interrupt controller has determined the mask level. At that time, if the CPU is executing an instruction that disables interrupts, the CPU always executes the next instruction after the instruction execution is completed.

4.7.3 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during transfer is not accepted until the transfer is completed.

With the EEPMOV.W instruction, even if an interrupt request other than the NMI is issued during transfer, the interrupt is not accepted until the transfer is completed. If the NMI interrupt request is issued, NMI exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an NMI interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1:  EEPMOV.W
     MOV.W      R4,R4
     BNE        L1
```

4.7.4 IENR Clearing

When an interrupt request is disabled by clearing the interrupt enable register or when the interrupt request register is cleared, the interrupt request should be masked (I bit = 1). If the above operation is executed while the I bit is 0 and contention between the instruction execution and the interrupt request generation occurs, exception handling, which corresponds to the interrupt request generated after instruction execution of the above operation is completed, is executed.

Table 7.1 Setting Programming Modes

TEST	$\overline{\text{NMI}}$	P36	PB0	PB1	PB2	LSI State after Reset End
0	1	x	x	x	x	User mode
0	0	1	x	x	x	Boot mode
1	x	x	0	0	0	Programmer mode

[Legend]
x: Don't care

7.3.1 Boot Mode

Table 7.2 shows the boot mode operations between reset end and branching 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 7.5, Flash Memory Programming/Erasing.
2. SCI3 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity. The inversion function of TXD and RXD pins by SPCR is set to “Not to be inverted,” so do not put the circuit for inverting a value between the host and this LSI.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI3 bit rate to match that of the host. The reset should end with the RXD pin high. The RXD and TXD pins should be pulled up on the board if necessary. After the reset is complete, it takes approximately 100 states before the chip is ready to measure the low-level period.
4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the completion of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 7.3.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'FFF380 to H'FFFE7F is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.

- PB3/AN3 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting Value	Other than B'0111	PB3 input pin
	B'0111	AN3 input pin

- PB2/AN2/$\overline{\text{IRQ3}}$ pin

Register Name	PMRB	AMR	Pin Function
Bit Name	IRQ3	CH3 to CH0	
Setting Value	0	Other than B'0110	PB2 input pin
		B'0110	AN2 input pin
	1	Other than B'0110	$\overline{\text{IRQ3}}$ input pin
		B'0110	Setting prohibited

- PB1/AN1/$\overline{\text{IRQ1}}$ pin

Register Name	PMRB	AMR	Pin Function
Bit Name	IRQ1	CH3 to CH0	
Setting Value	0	Other than B'0101	PB1 input pin
		B'0101	AN1 input pin
	1	Other than B'0101	$\overline{\text{IRQ1}}$ input pin
		B'0101	Setting prohibited

- PB0/AN0/$\overline{\text{IRQ0}}$ pin

Register Name	PMRB	AMR	Pin Function
Bit Name	IRQ0	CH3 to CH0	
Setting Value	0	Other than B'0100	PB0 input pin
		B'0100	AN0 input pin
	1	Other than B'0100	$\overline{\text{IRQ0}}$ input pin
		B'0100	Setting prohibited

9.13.4 Pin Functions

The relationship between the register settings and the port functions is shown below.

- PF3 (/TXD31/IrTXD) pin

Register Name	PFCR	SPCR	IrCR	PCRF	Pin Function
Bit Name	SC31S	SPC31	IrE	PCRF3	
Setting Value	0	x	x	0	PF3 input pin
				1	PF3 output pin
	1	0		0	PF3 input pin
				1	PF3 output pin
		1	0	x	TXD31 output pin
			1		IrTXD output pin

[Legend]
x: Don't care

- PF2 (/RXD31/IrRXD) pin

Register Name	PFCR	SCR3_1	IrCR	PCRF	Pin Function
Bit Name	SC31S	RE	IrE	PCRF2	
Setting Value	0	x	x	0	PF2 input pin
				1	PF2 output pin
	1	0		0	PF2 input pin
				1	PF2 output pin
		1	0	x	RXD31 input pin
			1		IrRXD input pin

[Legend]
x: Don't care

10.2 Input/Output Pin

Table 10.1 shows the RTC input/output pin.

Table 10.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Clock output	TMOW	Output	RTC divided clock output

10.3 Register Descriptions

The RTC has the following registers.

- Second data register/free running counter data register (RSECDR)
- Minute data register (RMINDR)
- Hour data register (RHRDR)
- Day-of-week data register (RWKDR)
- RTC control register 1 (RTCCR1)
- RTC control register 2 (RTCCR2)
- Clock source select register (RTCCSR)
- RTC Interrupt flag register (RTCFLG)

13.5 Operation

Timer G is an 8-bit timer with built-in input capture and interval functions.

13.5.1 Timer G Functions

Timer G is an 8-bit up-counter with two functions, an input capture timer function and an interval timer function.

The operation of these two functions is described below.

(1) Input Capture Timer Operation

When the TMIG bit in the port mode register F (PMRF) is set to 1, timer G functions as an input capture timer*.

Upon reset, the timer mode register G (TMG), timer counter G (TCG), input capture register GF (ICRGF), and input capture register GR (ICRGR) are all initialized to H'00.

Following a reset, TCG starts counting on the ϕ/64 internal clock.

The input clock can be selected from four internal clock sources by bits CKS1 and CKS0 in TMG.

When a rising edge/falling edge is detected in the input capture signal input from the TMIG pin, the TCG value at that time is transferred to ICRGR/ICRGF. When the edge selected by IIEGS in TMG is input, IRRTG in IRR2 is set to 1, and if the IENTG bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU. For details on interrupts, see section 4, Interrupt Controller.

TCG can be cleared by a rising edge, falling edge, or both edges of the input capture signal, according to the setting of bits CCLR1 and CCLR0 in TMG. If TCG overflows when the input capture signal is high, the OVFH bit in TMG is set; if TCG overflows when the input capture signal is low, the OVFL bit in TMG is set. If the OVIE bit in TMG is 1 when these bits are set, IRRTG in IRR2 is set to 1, and if the IENTG bit in IENR2 is 1, timer G sends an interrupt request to the CPU. For details on interrupts, see section 4, Interrupt Controller.

Timer G has a built-in noise canceller that enables high-frequency component noise to be eliminated from pulses input from the TMIG pin. For details, see section 13.4, Noise Canceller.

Note: * An input capture signal may be generated when TMIG is modified.

14.8.10 Contention between TCNT Write and Overflow

If there is an up-count in the T2 state of a TCNT write cycle and overflow occurs, the TCNT write takes priority and the TCFV flag in TSR is not set.

Figure 14.37 shows the operation timing when there is contention between TCNT write and overflow.

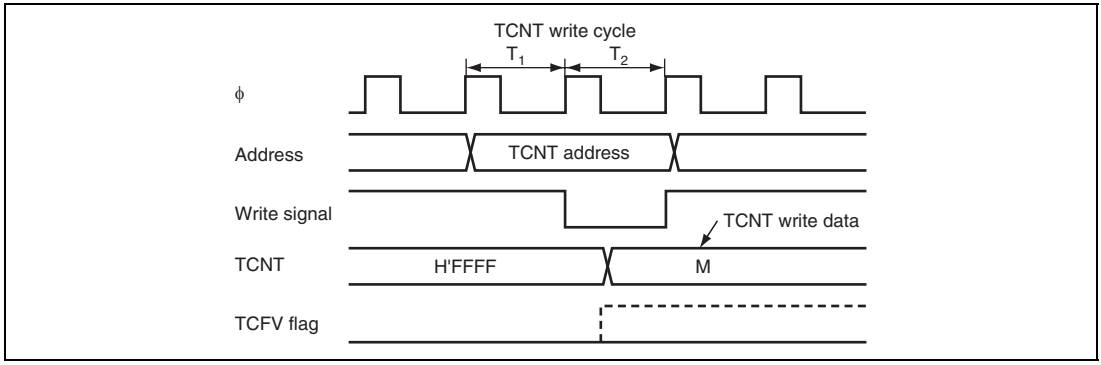

Figure 14.37 Contention between TCNT Write and Overflow

14.8.11 Multiplexing of I/O Pins

The TIOCA1 I/O pin is multiplexed with the TCLKA input pin, the TIOCB1 I/O pin with the TCLKB input pin, and the TIOCA2 I/O pin with the TCLKC input pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

14.8.12 Interrupts when Module Standby Function is Used

If the module standby function is used when an interrupt has been requested, it will not be possible to clear the CPU interrupt source with the interrupt request enabled. Interrupts should therefore be disabled before using the module standby function.

14.8.13 Output Conditions for 0% Duty and 100% Duty

When TGR is rewritten to change the duty in PWM mode, 0% duty or 100% duty is specified depending on the TCT value when rewritten, the TGR value before rewritten, and the TGR value after rewritten. For derails, refer to figure 14.20.

15.3.5 Event Counter Control/Status Register (ECCSR)

ECCSR controls counter overflow detection, counter resetting, and count-up function.

Bit	Bit Name	Initial Value	R/W	Description
7	OVH	0	R/(W)*	Counter Overflow H This is a status flag indicating that ECH has overflowed. [Setting condition] ECH overflows from H'FF to H'00 [Clearing condition] Writing of 0 to bit OVH after reading OVH = 1
6	OVL	0	R/(W)*	Counter Overflow L This is a status flag indicating that ECL has overflowed. [Setting condition] ECL overflows from H'FF to H'00 while CH2 is set to 1 [Clearing condition] Writing of 0 to bit OVL after reading OVL = 1
5	—	0	R/W	Reserved Although this bit is readable/writable, it should not be set to 1.
4	CH2	0	R/W	Channel Select Selects how ECH and ECL event counters are used 0: ECH and ECL are used together as a single-channel 16-bit event counter 1: ECH and ECL are used as two-channel 8-bit event counter
3	CUEH	0	R/W	Count-Up Enable H Enables event clock input to ECH. 0: ECH event clock input is disabled (ECH value is retained) 1: ECH event clock input is enabled

17.3.12 Serial Extended Mode Register (SEMR)

SEMR controls extended functions of the SCI3_1, i.e. specifies the basic clock in asynchronous mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved The write value should always be 0.
3	ABCS	0	R/W	Asynchronous Mode Basic Clock Select Selects the basic clock for one-bit interval in asynchronous mode. The ABCS setting is enabled in asynchronous mode (COM = 0 in SMR3) 0: Basic clock with a frequency 16 times the transfer rate 1: Basic clock with a frequency 8 times the transfer rate Clear this bit to 0 when the IrDA function is enabled.
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

Table 17.9 Data Transfer Formats (Asynchronous Mode)

SMR				Serial Data Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	START 8-bit data STOP
0	0	0	1	START 8-bit data STOP STOP
0	0	1	0	START 8-bit data MPB STOP
0	0	1	1	START 8-bit data MPB STOP STOP
0	1	0	0	START 8-bit data P STOP
0	1	0	1	START 8-bit data P STOP STOP
0	1	1	0	START 5-bit data STOP
0	1	1	1	START 5-bit data STOP STOP
1	0	0	0	START 7-bit data STOP
1	0	0	1	START 7-bit data STOP STOP
1	0	1	0	START 7-bit data MPB STOP
1	0	1	1	START 7-bit data MPB STOP STOP
1	1	0	0	START 7-bit data P STOP
1	1	0	1	START 7-bit data P STOP STOP
1	1	1	0	START 5-bit data P STOP
1	1	1	1	START 5-bit data P STOP STOP

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

17.9.7 Relation between RDR Reading and bit RDRF

In a receive operation, the SCI3 continually checks the RDRF flag. If bit RDRF is cleared to 0 when reception of one frame ends, normal data reception is completed. If bit RDRF is set to 1, this indicates that an overrun error has occurred.

When the contents of RDR are read, bit RDRF is cleared to 0 automatically. Therefore, if RDR is read more than once, the second and subsequent read operations will be performed while bit RDRF is cleared to 0. Note that, when an RDR read is performed while bit RDRF is cleared to 0, if the read operation coincides with completion of reception of a frame, the next frame of data may be read. This is shown in figure 17.23.

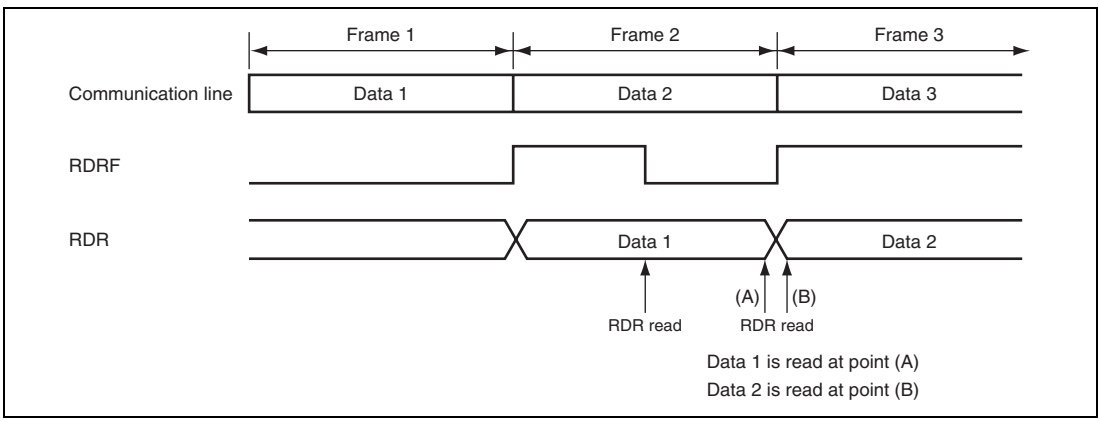

Figure 17.23 Relation between RDR Read Timing and Data

In this case, only a single RDR read operation (not two or more) should be performed after first checking that bit RDRF is set to 1. If two or more reads are performed, the data read the first time should be transferred to RAM, etc., and the RAM contents used. Also, ensure that there is sufficient margin in an RDR read operation before reception of the next frame is completed. To be precise in terms of timing, the RDR read should be completed before bit 7 is transferred in clock synchronous mode, or before the STOP bit is transferred in asynchronous mode.

20.2 Input/Output Pins

Table 20.1 shows the input pins used by the A/D converter.

Table 20.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVcc	Input	Power supply and reference voltage of analog part
Analog ground pin	AVss	Input	Ground and reference voltage of analog part
Analog input pin 0	AN0	Input	Analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
External trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input that controls the A/D conversion start.

20.3 Register Descriptions

The A/D converter has the following registers.

- A/D result register (ADRR)
- A/D mode register (AMR)
- A/D start register (ADSR)

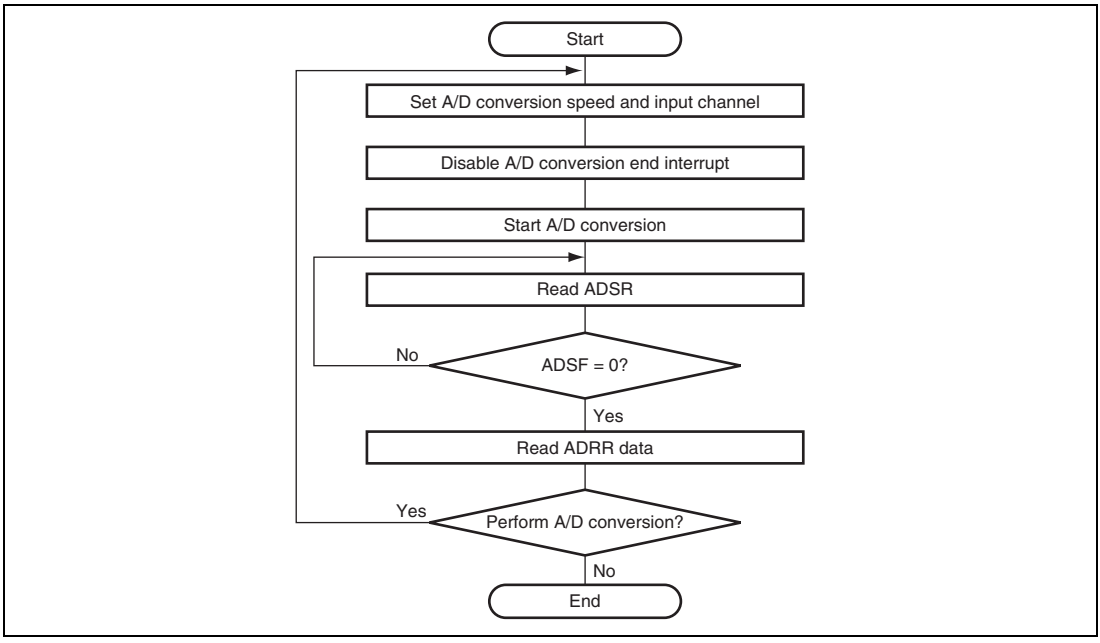

Figure 20.4 Flowchart of Procedure for Using A/D Converter (Polling by Software)

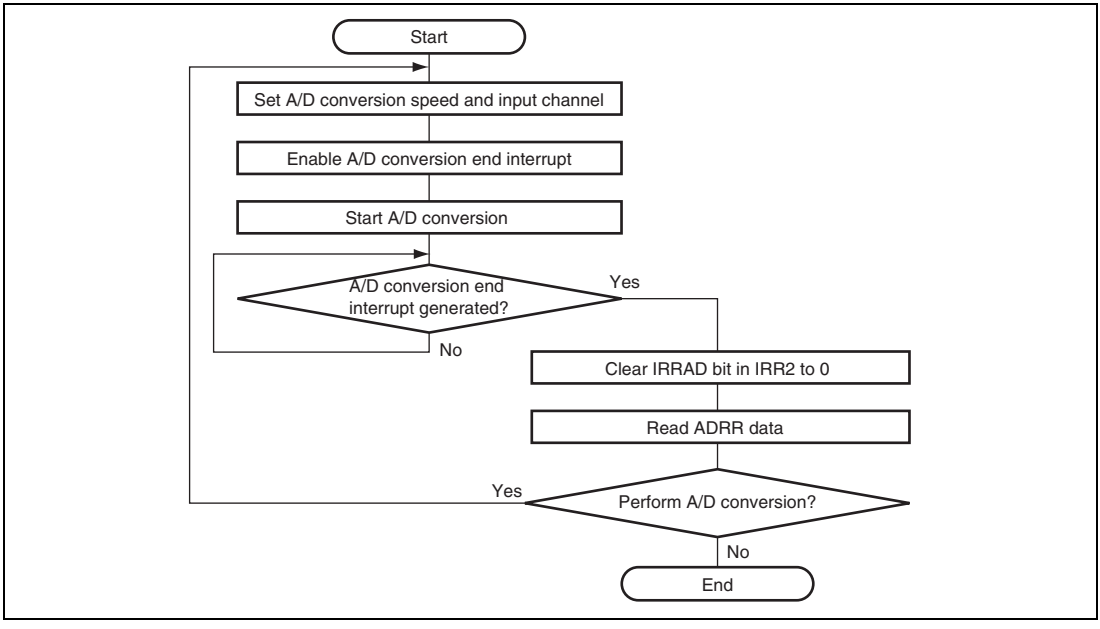

Figure 20.5 Flowchart of Procedure for Using A/D Converter (Interrupts Used)

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Input low voltage	V_{IL}	$\overline{RES}$, TEST, $\overline{NMI}$*[3], $\overline{WKP0}$ to $\overline{WKP7}$, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{IRQ3}$, $\overline{IRQ4}$, IRQAEC, AEVL, AEVH, TMIC, TMIF, TMIG, $\overline{ADTRG}$, SCK33, SCK32, SCK31, SCK4		–0.3	—	$0.1V_{cc}$	V	
		RXD33, RXD32, RXD31, IrRXD, UD		–0.3	—	$0.2V_{cc}$	V	
		OSC1		–0.3	—	$0.1V_{cc}$	V	
		X1		–0.3	—	$0.1V_{cc}$	V	
		P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P93, PA0 to PA3, PB0 to PB7, PC0 to PC7, PE0 to PE7, PF0 to PF3, TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2, SCL, SDA		–0.3	—	$0.2V_{cc}$	V	

Appendix

A. Instruction Set

A.1 Instruction List

Condition Code

Symbol	Description
Rd	General destination register
Rs	General source register
Rn	General register
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
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
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Logical exclusive OR of the operands on both sides

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
DEC	DEC.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	↕	↕	↕	—	2	
	DEC.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	↕	↕	↕	—	2	
DAS	DAS.Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	—	2	
MULXU	MULXU. B Rs, Rd	B		2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	—	—	—	—	—	—	14	
	MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	—	22	
MULXS	MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	—	16	
	MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	—	24	
DIVXU	DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	—	14	
	DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—	22	
DIVXS	DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—	16	
	DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—	24	
CMP	CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	↕	2	
	CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	↕	2	
	CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	↕	4	
	CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	↕	2	
	CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	↕	4	
	CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	↕	2	

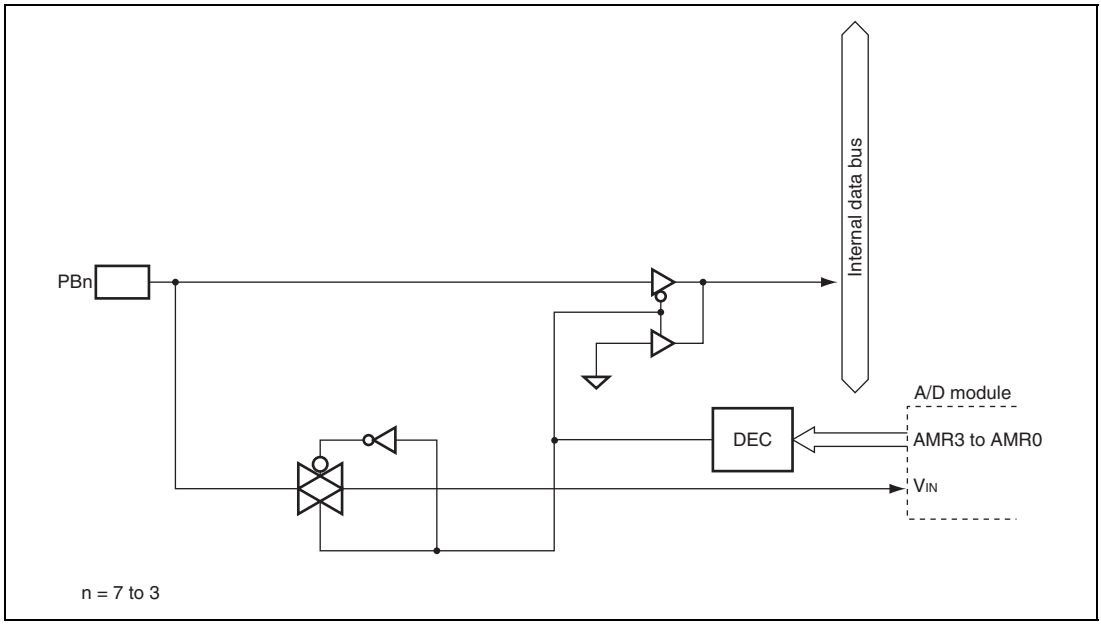

Figure B.10 (a) Port B Block Diagram (PB7 to PB3)