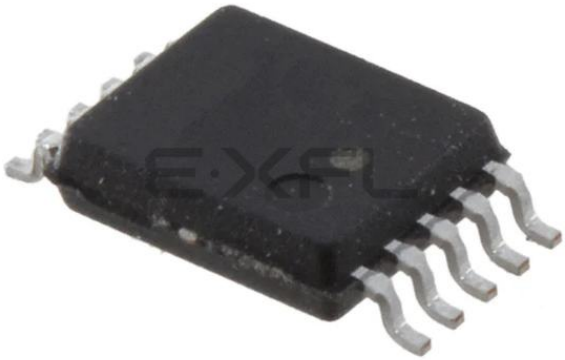


Welcome to E-XFL.COM

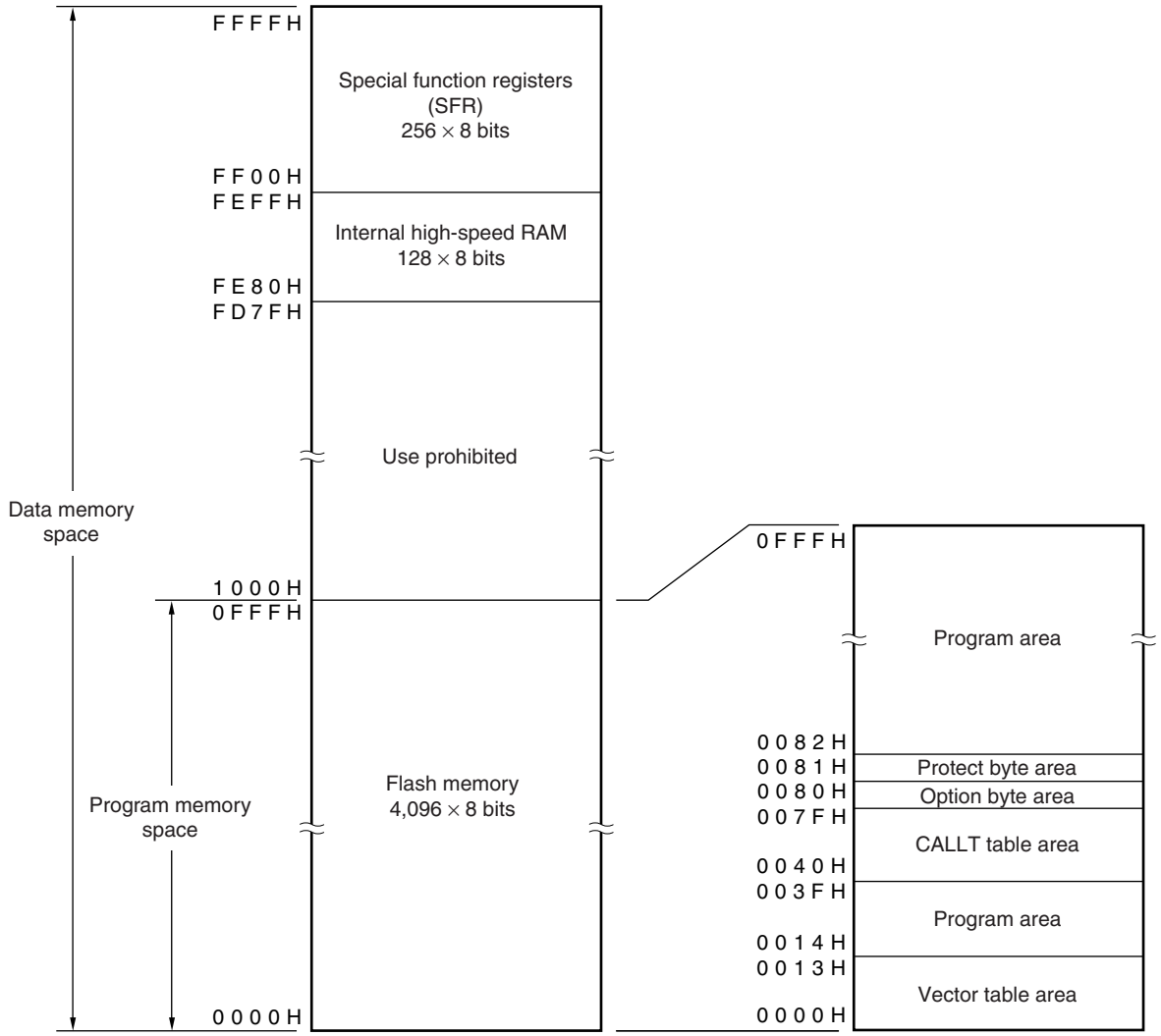
What is "Embedded - Microcontrollers"?

"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	78K0S
Core Size	8-Bit
Speed	10MHz
Connectivity	-
Peripherals	LVD, POR, PWM, WDT
Number of I/O	7
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	10-LSSOP (0.225", 5.72mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f9502ma-cac-a

<R> **Figure 3-3. Memory Map (μPD78F9202, 78F9502)**



Remark The option byte and protect byte are 1 byte each.

3.4.2 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with the 8-bit data in an instruction word.

The fixed space where this addressing is applied is the 160-byte space FE80H to FF1FH (FE80H to FEFFH (internal high-speed RAM) + FF00H to FF1FH (special function registers)).

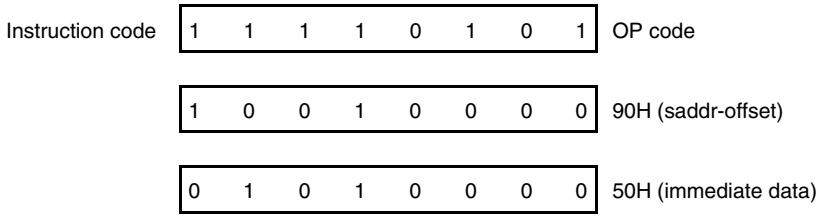
The SFR area where short direct addressing is applied (FF00H to FF1FH) is a part of the total SFR area. In this area, ports which are frequently accessed in a program and a compare register of the timer counter are mapped, and these SFRs can be manipulated with a small number of bytes and clocks.

When 8-bit immediate data is at 80H to FFH, bit 8 of an effective address is cleared to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

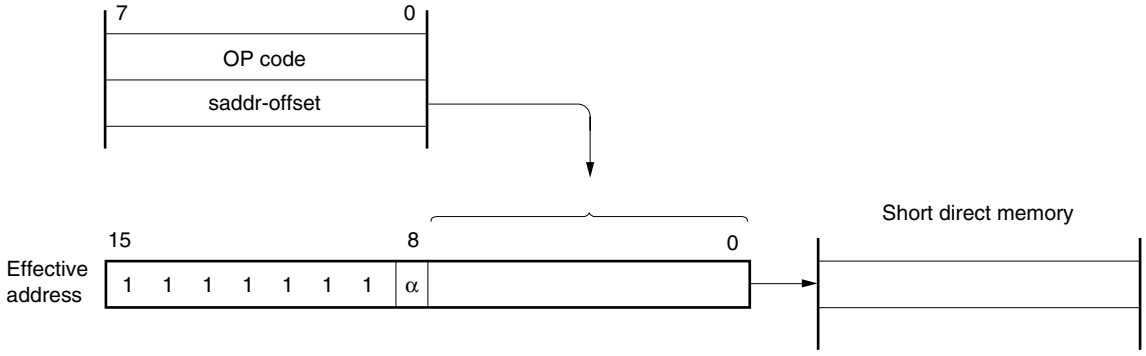
Identifier	Description
saddr	Label or FE80H to FF1FH immediate data
saddrp	Label or FE80H to FF1FH immediate data (even address only)

[Description example]

```
EQU DATA1 0FE90H  ; DATA1 shows FE90H of a saddr area,
MOV DATA1, #50H   ; When setting the immediate data to 50H
```



[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$.
When 8-bit immediate data is 00H to 1FH, $\alpha = 1$.

(2) Port registers (P2 to P4)

These registers are used to write data to be output from the corresponding port pin to an external device connected to the chip.

When a port register is read, the pin level is read in the input mode, and the value of the output latch of the port is read in the output mode.

P20 to P23, P32, P40 and P43 are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to 00H.

Figure 4-13. Format of Port Register

Address: FF02H, After reset: 00H (Output latch) R/W

Symbol	7	6	5	4	3	2	1	0
P2	0	0	0	0	P23	P22	P21	P20

Address: FF03H, After reset: 00H[Note] (Output latch) R/W[Note]

Symbol	7	6	5	4	3	2	1	0
P3	0	0	0	P34	0	P32	0	0

Address: FF04H, After reset: 00H (Output latch) R/W

Symbol	7	6	5	4	3	2	1	0
P4	0	0	0	0	P43	0	0	P40

Pmn	m = 2 to 4; n = 0 to 4	
	Controls of output data (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note Because P34 is read-only, its reset value is undefined.

(3) Port mode control register 2 (PMC2) (μPD78F920x only)

This register specifies the port/alternate function (except the A/D converter function) mode or the A/D converter mode.

Each bit of the PMC2 register corresponds to each pin of port 2 and can be specified in 1-bit units.

PMC2 is set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PMC2 to 00H.

Figure 6-26. Control Register Settings for Pulse Width Measurement by Means of Restart (with Rising Edge Specified) (2/2)

(b) Prescaler mode register 00 (PRM00)

	ES110	ES100	ES010	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).
Specifies rising edge for pulse width detection.
Setting invalid (setting “10” is prohibited.)

(c) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	0	0/1	0

Clears and starts at valid edge of TI000 pin.

Figure 6-27. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)

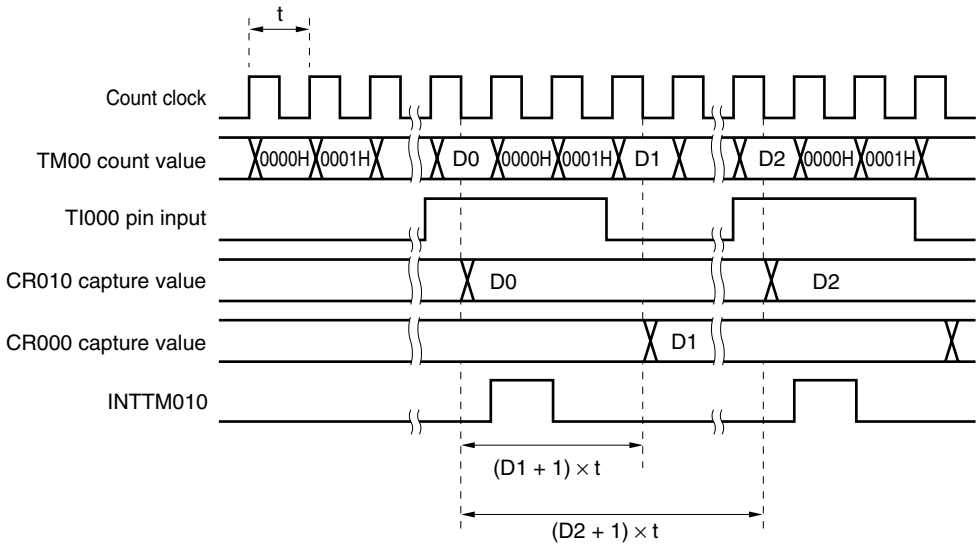


Figure 7-4. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF70H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stop timer count operation (counter is cleared to 0)
1	Enable timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock (f_{CNT}) selection
0	0	0	f_{XP} (10 MHz)
0	0	1	$f_{XP}/2^2$ (2.5 MHz)
0	1	0	$f_{XP}/2^4$ (625 kHz)
0	1	1	$f_{XP}/2^6$ (156.25 kHz)
1	0	0	$f_{XP}/2^{12}$ (2.44 kHz)
1	0	1	$f_{RL}/2^7$ (1.88 kHz (TYP.))
Other than above			Setting prohibited

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disable output
1	Enable output

Cautions 1. When TMHE1 = 1, setting the other bits of the TMHMD1 register is prohibited.

2. In the PWM output mode, be sure to set 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

Remarks 1. f_{XP}: Oscillation frequency of clock to peripheral hardware

2. f_{RL}: Low-speed internal oscillation clock oscillation frequency

3. Figures in parentheses apply to operation at f_{XP} = 10 MHz, f_{RL} = 240 kHz (TYP.).

<R> **(2) Port mode register 2 (PM2) and port mode control register 2 (PMC2)[Note]**

When using the P20/TOH1/TI000/ANI0 pin for timer output, clear PM20, the output latch of P20, and PMC20 to 0.

PM2 and PMC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM2 to FFH, and clears PMC2 to 00H.

Note μPD78F920x only

Figure 7-5. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

PM2n	P2n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 7-6. Format of Port Mode Control Register 2 (PMC2) (μPD78F920x Only)

Address: FF84H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	Specification of operation mode (n = 0 to 3)
0	Port/Alternate-function (except A/D converter) mode
1	A/D converter mode

7.4 Operation of 8-Bit Timer H1

7.4.1 Operation as interval timer/square-wave output

When 8-bit timer counter H1 and compare register 01 (CMP01) match, an interrupt request signal (INTTMH1) is generated and 8-bit timer counter H1 is cleared to 00H.

Compare register 11 (CMP11) is not used in interval timer mode. Since a match of 8-bit timer counter H1 and the CMP11 register is not detected even if the CMP11 register is set, timer output is not affected.

By setting bit 0 (TOEN1) of timer H mode register 1 (TMHMD1) to 1, a square wave of any frequency (duty = 50%) is output from TOH1.

<4> When 8-bit timer counter H1 and the CMP11 register match, TOH1 output becomes inactive and the compare register to be compared with 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register. At this time, 8-bit timer counter H1 is not cleared and the INTTMH1 signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHE1 = 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

PWM pulse output cycle = $(N+1)/f_{CNT}$
Duty = Active width : Total width of PWM = (M + 1) : (N + 1)

Cautions 1. In PWM output mode, the setting value for the CMP11 register can be changed during timer count operation. However, three operation clocks (signal selected using the CKS12 to CKS10 bits of the TMHMD1 register) or more are required to transfer the register value after rewriting the CMP11 register value.

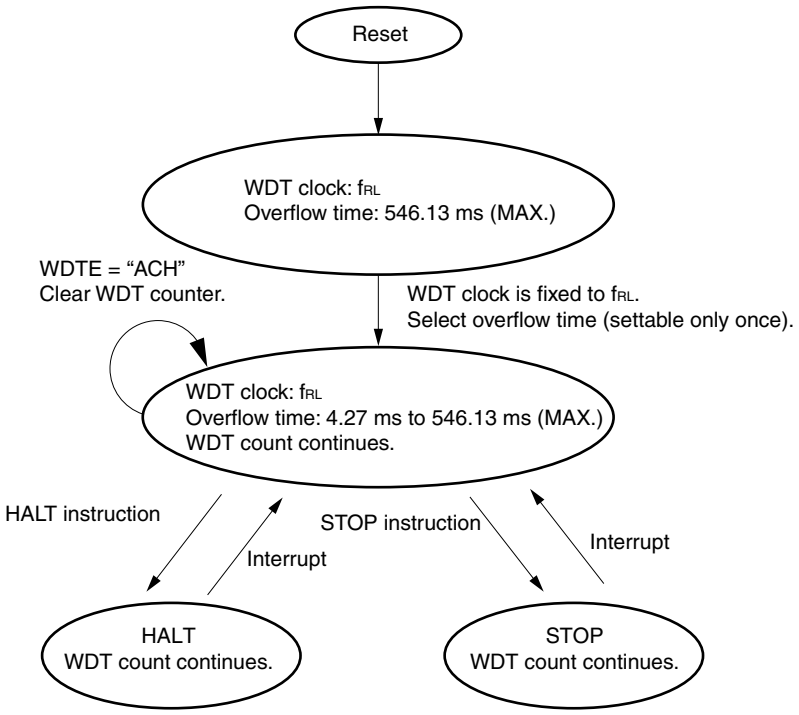
2. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

(2) Timing chart

The operation timing in PWM output mode is shown below.

Caution Make sure that the CMP11 register setting value (M) and CMP01 register setting value (N) are within the following range.
00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH

Figure 8-4. Status Transition Diagram When "Low-Speed Internal Oscillator Cannot Be Stopped" Is Selected by Option Byte



9.4 A/D Converter Operations

9.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion using the analog input channel specification register (ADS).
<2> Set ADCE to 1 and wait for 1 μs or longer.
<3> Execute two NOP instructions or an instruction equivalent to two machine cycles.
<4> Set ADCS to 1 and start the conversion operation.
(<5> to <11> are operations performed by hardware.)
<5> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.
<6> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation has ended.
<7> Bit 9 of the successive approximation register (SAR) is set. The D/A converter voltage tap is set to (1/2) V_{DD} by the tap selector.
<8> The voltage difference between the D/A converter voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{DD}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) V_{DD}, the MSB is reset to 0.
<9> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The D/A converter voltage tap is selected according to the preset value of bit 9, as described below.
- Bit 9 = 1: (3/4) V_{DD}
- Bit 9 = 0: (1/4) V_{DD}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<10> Comparison is continued in this way up to bit 0 of SAR.
<11> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.
<12> Repeat steps <5> to <11>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <3>. To restart A/D conversion from the status of ADCE = 0, start from <2>.

Cautions 1. Make sure the period of <1> to <4> is 1 μs or more.
2. It is no problem if the order of <1> and <2> is reversed.

Remark The following two types of A/D conversion result registers can be used.
- ADCR (16 bits): Stores a 10-bit A/D conversion value.
- ADCRH (8 bits): Stores an 8-bit A/D conversion value.

9.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$ADCR = INT\left(\frac{V_{AIN}}{V_{DD}} \times 1024 + 0.5\right)$$

or

$$(ADCR - 0.5) \times \frac{V_{DD}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{V_{DD}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

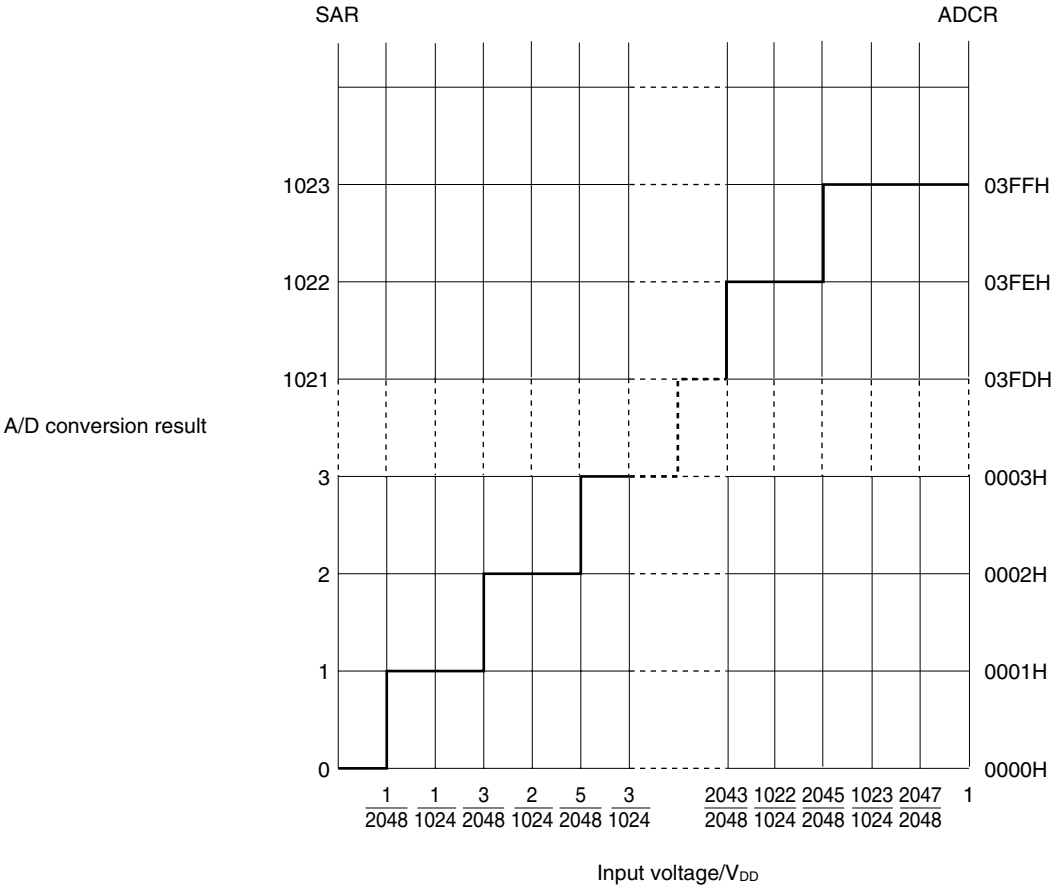
V_{AIN}: Analog input voltage

V_{DD}: V_{DD} pin voltage

ADCR: 10-bit A/D conversion result register (ADCR) value

Figure 9-11 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 9-11. Relationship Between Analog Input Voltage and A/D Conversion Result



11.2 Standby Function Operation

11.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The operating statuses in the HALT mode are shown below.

Caution Because an interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag clear, the standby mode is immediately cleared if set.

Table 11-2. Operating Statuses in HALT Mode

Setting of HALT Mode / Item		Low-Speed Internal Oscillator cannot be stopped[Note 1].	Low-Speed Internal Oscillator can be stopped[Note 1].	
			When Low-Speed Internal Oscillation Continues	When Low-Speed Internal Oscillation Stops
System clock		Clock supply to CPU is stopped.		
CPU		Operation stops.		
Port (latch)		Holds status before HALT mode was set.		
16-bit timer/event counter 00[Note 2]		Operable		
8-bit timer H1	Sets count clock to f_{XP} to $f_{XP}/2^{12}$	Operable		
	Sets count clock to $f_{RL}/2^{7}$	Operable	Operable	Operation stops.
Watchdog timer	“System clock” selected as operating clock	Setting disabled.	Operation stops.	
	“Low-speed internal oscillation clock” selected as operating clock	Operable (Operation continues)	Operation stops.	
A/D converter[Note 2]		Operable		
Power-on-clear circuit		Always operates.		
Low-voltage detector		Operable		
External interrupt		Operable		

Notes 1. “Cannot be stopped” or “Stopped by software” is selected for low-speed internal oscillator by the option byte (for the option byte, see **CHAPTER 15 OPTION BYTE**).

<R> **2.** μPD78F920x only

Table 16-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash memory programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash memory programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of on-board/off-board programming.	Self programming mode

Remarks 1. The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

2. Refer to the following sections for details on the flash memory writing control function.

- **16.7 On-Board and Off-Board Flash Memory Programming**
- **16.8 Flash Memory Programming by Self Programming**

16.4 Writing with Flash Memory Programmer

The following two types of dedicated flash memory programmers can be used for writing data to the internal flash memory of the 78K0S/KU1+.

- FlashPro5 (PG-FP5, FL-PR5)
<R> - QB-MINI2

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the 78K0S/KU1+ have been mounted on the target system. The connectors that connect the dedicated flash memory programmer and the test pad must be mounted on the target system. The test pad is required only when writing data with the crystal/ceramic resonator mounted (refer to Figure 16-4 for mounting of the test pad).

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0S/KU1+ is mounted on the target system.

<R> **Remark** The FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

(7) Flash write buffer register (FLW)

This register is used to store the data to be written to the flash memory.

This register is set with an 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 16-16. Format of Flash Write Buffer Register (FLW)

Address: FFA8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

(8) Protect byte

This protect byte is used to specify the area that is to be protected from writing or erasing. The specified area is valid only in the self-programming mode. Because self-programming of the protected area is invalid, the data written to the protected area is guaranteed.

Figure 16-17. Format of Protect Byte (1/2)

Address: 0081H

7	6	5	4	3	2	1	0
1	PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	1	1

• μPD78F9200, 78F9500

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	Status
0	1	1	1	0	Blocks 3 to 0 are protected.
0	1	1	1	1	Blocks 1 and 0 are protected. Blocks 2 and 3 can be written or erased.
1	1	1	1	1	All blocks can be written or erased.
Other than above					Setting prohibited

• μPD78F9201, 78F9501

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	Status
0	1	1	0	0	Blocks 7 to 0 are protected.
0	1	1	0	1	Blocks 5 to 0 are protected. Blocks 6 and 7 can be written or erased.
0	1	1	1	0	Blocks 3 to 0 are protected. Blocks 4 to 7 can be written or erased.
0	1	1	1	1	Blocks 1 and 0 are protected. Blocks 2 to 7 can be written or erased.
1	1	1	1	1	All blocks can be written or erased.
Other than above					Setting prohibited

18.1.2 Description of "Operation" column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
IE:	Interrupt request enable flag
( ):	Memory contents indicated by address or register contents in parentheses
×H, ×L:	Higher 8 bits and lower 8 bits of 16-bit register
∧:	Logical product (AND)
∨:	Logical sum (OR)
∀:	Exclusive logical sum (exclusive OR)
¯:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

18.1.3 Description of "Flag" column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to the result
R:	Previously saved value is stored

DC Characteristics (1/4)

(1) μPD78F920x (T_A = –40 to +85°C, V_{DD} = 2.0 to 5.5 V[Note], V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin	2.0 V ≤ V_{DD} ≤ 5.5 V			–5	mA
		Total of all pins	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
			2.0 V ≤ V_{DD} < 4.0 V			–15	mA
Output current, low	I_{OL}	Per pin	2.0 V ≤ V_{DD} ≤ 5.5 V			10	mA
		Total of all pins	4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
			2.0 V ≤ V_{DD} < 4.0 V			15	mA
Input voltage, high	V_{IH1}	P23 in external clock mode and pins other than P20 and P21		0.8V_{DD}		V_{DD}	V
	V_{IH2}	P23 in other than external clock mode, P20 and P21		0.7V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P23 in external clock mode and pins other than P20 and P21		0		0.2V_{DD}	V
	V_{IL2}	P23 in other than external clock mode, P20 and P21		0		0.3V_{DD}	V
Output voltage, high	V_{OH}	Total of output pins I_{OH} = –15 mA	4.0 V ≤ V_{DD} ≤ 5.5 V I_{OH} = –5 mA	V_{DD} – 1.0			V
		I_{OH} = –100 μA	2.0 V ≤ V_{DD} < 4.0 V	V_{DD} – 0.5			V
Output voltage, low	V_{OL}	Total of output pins I_{OL} = 30 mA	4.0 V ≤ V_{DD} ≤ 5.5 V I_{OL} = 10 mA			1.3	V
		2.0 V ≤ V_{DD} < 4.0 V I_{OL} = 400 μA				0.4	V
Input leakage current, high	I_{LIH}	V_I = V_{DD}	Pins other than X1			1	μA
Input leakage current, low	I_{LIL}	V_I = 0 V	Pins other than X1			–1	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD}	Pins other than X2			1	μA
Output leakage current, low	I_{LOL}	V_O = 0 V	Pins other than X2			–1	μA
Pull-up resistance value	R_{PU}	V_I = 0 V		10	30	100	kΩ
Pull-down resistance value	R_{PD}	P22, P23, reset status		10	30	100	kΩ

Note Use this product in a voltage range of 2.2 to 5.5 V because the detection voltage (V_{POC}) of the power-on clear (POC) circuit is 2.1 V ±0.1 V.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

A.1 Software Package

SP78K0S 78K0S microcontroller software package	Development tools (software) common to the 78K0S microcontrollers are combined in this package.

A.2 Language Processing Software

RA78K0S[Note 1] Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF789234). **<Precaution when using RA78K0S in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (PM+) on Windows. PM+ is included in assembler package.
CC78K0S[Note 1] C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file. **<Precaution when using CC78K0S in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (PM+) on Windows. PM+ is included in assembler package.
DF789234[Note 2] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0S, CC78K0S, ID78K0S-QB, and the system simulator). The corresponding OS and host machine differ depending on the tool to be used.

Notes 1. If the versions of RA78K0S and CC78K0S are Ver.2.00 or later, different versions of RA78K0S and CC78K0S can be installed on the same machine.

2. The DF789234 can be used in common with the RA78K0S, CC78K0S, ID78K0S-QB, and the system simulator. Download the DF789234 from the download site for development tools (http://www.necel.com/micro/en/ods/).

Remarks 1. The QB-78K0SKX1 is supplied with the integrated debugger ID78K0S-QB, a USB interface cable, the on-chip debug emulator QB-MINI2, and a connection cable.
Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/en/ods/) when using the QB-MINI2.

2. The packed contents of QB-78K0SKX1 differ depending on the part number, as follows.

Packed Contents / Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	Target Connector
QB-78K0SKX1-ZZZ	QB-78K0SKX1	None		
QB-78K0SKX1-T10MA		QB-50-EP-01T	QB-10MA-EA-01T	QB-10MA-NQ-01T

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0S/Kx1+ microcontrollers. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. When using this as on-chip debug emulator, it should be used in combination with a connection cable and a USB interface cable that is used to connect the host machine.
Target connector specifications	16-pin general-purpose connector (2.54 mm pitch)

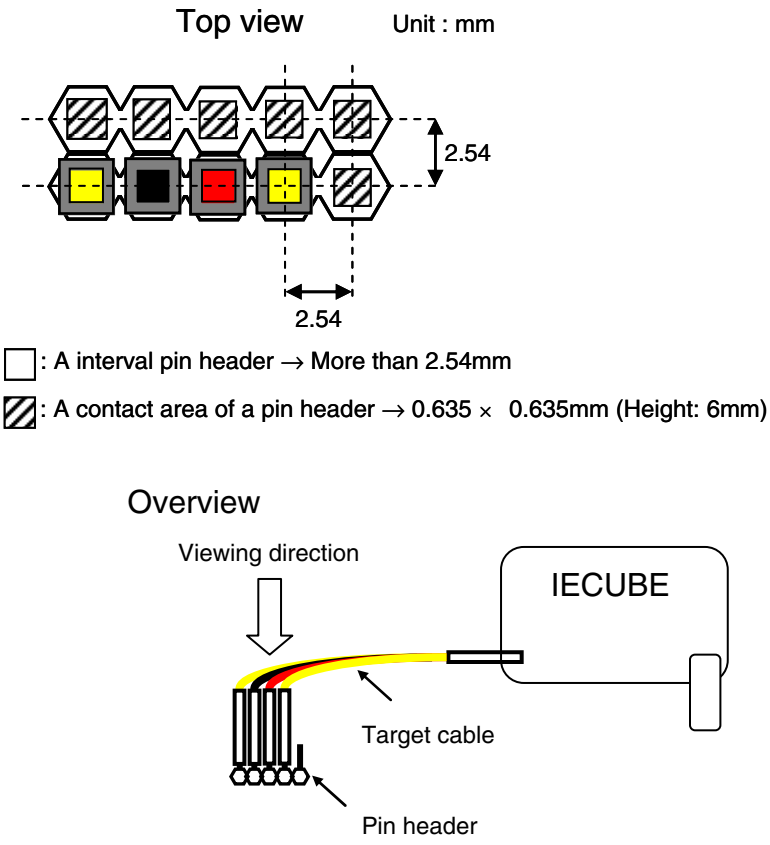
Remark Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/en/ods/).

A.5 Debugging Tools (Software)

ID78K0S-QB[Note] (supporting QB-78K0SKX1, QB-MINI2) Integrated debugger	This debugger supports the in-circuit emulators for the 78K0S/Kx1+ microcontrollers. The ID78K0S-QB is Windows-based software. Provided with the debug function supporting C language, source programming, disassemble display, and memory display are possible. It should be used in combination with the device file (DF789234).
SM+ for 78K0S SM+ for 78K0S/Kx1+[Note] System simulator	System simulator is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of system simulator allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. System simulator should be used in combination with the device file (DF789234). The following two types of system simulators supporting the 78K0S/Kx1+ microcontrollers are available. • SM+ for 78K0S (instruction simulation version) This can only simulate a CPU. It is included in the software package. • SM+ for 78K0S/Kx1+ (instruction + peripheral simulation version) This can simulate a CPU and peripheral hardware (ports, timers, serial interfaces, etc.).

Note Download the ID78K0S-QB from the download site for development tools (http://www.necel.com/micro/en/ods/).

Figure B-2. When using the 78K0S/Kx1+ target cable (single track)



(6/15)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 6	Hard	16-bit timer/ event counters 00 (μPD78F920x only)	Timer start errors	An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 00 (TM00) is started asynchronously to the count clock.	p. 124 ☐
	Soft		One-shot pulse output	One-shot pulse output normally operates only in the free-running mode or in the clear & start mode at the valid edge of the TI000 pin. Because an overflow does not occur in the clear & start mode on a match between TM00 and CR000, one-shot pulse output is not possible.	p. 125 ☐
			Capture operation	When the CRC001 bit value is 1, capture is not performed in the CR000 register if both the rising and falling edges have been selected as the valid edges of the TI000 pin.	p. 127 ☐
				When the CRC001 bit value is 1, the TM00 count value is not captured in the CR000 register when a valid edge of the TI010 pin is detected, but the input from the TI010 pin can be used as an external interrupt source because INTTM000 is generated at that timing.	p. 127 ☐
			Changing compare register during timer operation	With the 16-bit timer capture/compare register 0n0 (CR0n0) used as a compare register, when changing CR0n0 around the timing of a match between 16-bit timer counter 00 (TM00) and 16-bit timer capture/compare register 0n0 (CR0n0) during timer counting, the change timing may conflict with the timing of the match, so the operation is not guaranteed in such cases. To change CR0n0 during timer counting, INTTM000 interrupt servicing performs the following operation.	p. 128 ☐
				If CR010 is changed during timer counting without performing processing <1> above, the value in CR010 may be rewritten twice or more, causing an inversion of the output level of the TO00 pin at each rewrite.	p. 128 ☐
			External event counter	The timing of the count start is after two valid edge detections.	p. 129 ☐
	Hard		External clock limitation	When using an input pulse of the TI000 pin as a count clock (external trigger), be sure to input the pulse width which satisfies the AC characteristics. For the AC characteristics, refer to CHAPTER 19 ELECTRICAL SPECIFICATIONS.	p. 130 ☐
				When an external waveform is input to 16-bit timer/event counter 00, it is sampled by the noise limiter circuit and thus an error occurs on the timing to become valid inside the device.	p. 130 ☐
Chapter 7	Soft	8-bit timer H1	CMP01: 8-bit timer H compare register 01	CMP01 cannot be rewritten during timer count operation.	p. 133 ☐
			CMP11: 8-bit timer H compare register 11	In the PWM output mode, be sure to set CMP11 when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).	p. 133 ☐
			TMHMD1: 8-bit timer H mode register 1	When TMHE1 = 1, setting the other bits of the TMHMD1 register is prohibited.	p. 135 ☐
				In the PWM output mode, be sure to set 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).	p. 135 ☐
	Hard		PWM output	In PWM output mode, the setting value for the CMP11 register can be changed during timer count operation. However, three operation clocks (signal selected using the CKS12 to CKS10 bits of the TMHMD1 register) or more are required to transfer the register value after rewriting the CMP11 register value.	p. 141 ☐
	Soft			Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).	p. 141 ☐

(7/15)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 7	Soft	8-bit timer H1	PWM output	Make sure that the CMP11 register setting value (M) and CMP01 register setting value (N) are within the following range. 00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH	p. 141 ☐
Chapter 8	Soft	Watchdog timer	WDTM: Watchdog timer mode register	Set bits 7, 6, and 5 to 0, 1, and 1, respectively. Do not set the other values.	p. 149 ☐
				After reset is released, WDTM can be written only once by an 8-bit memory manipulation instruction. If writing is attempted a second time, an internal reset signal is generated. However, at the first write, if "1" and "x" are set for WDCS4 and WDCS3 respectively and the watchdog timer is stopped, then the internal reset signal does not occur even if the following are executed. • Second write to WDTM • 1-bit memory manipulation instruction to WDTE • Writing of a value other than "ACH" to WDTE	p. 150 ☐
				WDTM cannot be set by a 1-bit memory manipulation instruction.	p. 150 ☐
				When using the flash memory programming by self programming, set the overflow time for the watchdog timer so that enough overflow time is secured (Example 1-byte writing: 200 μs MIN., 1-block deletion: 10 ms MIN.).	p. 150 ☐
			WDTE: Watchdog timer enable register	If a value other than ACH is written to WDTE, an internal reset signal is generated.	p. 150 ☐
				If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.	p. 150 ☐
				The value read from WDTE is 9AH (this differs from the written value (ACH)).	p. 150 ☐
	Hard		When "low-speed internal oscillator cannot be stopped" is selected by option byte	In this mode, operation of the watchdog timer cannot be stopped even during STOP instruction execution. For 8-bit timer H1 (TMH1), a division of the low-speed internal oscillation clock can be selected as the count source, so clear the watchdog timer using the interrupt request of TMH1 before the watchdog timer overflows after STOP instruction execution. If this processing is not performed, an internal reset signal is generated when the watchdog timer overflows after STOP instruction execution.	p. 151 ☐
			When "low-speed internal oscillator can be stopped by software" is selected by option byte	In this mode, watchdog timer operation is stopped during HALT/STOP instruction execution. After HALT/STOP mode is released, counting is started again using the operation clock of the watchdog timer set before HALT/STOP instruction execution by WDTM. At this time, the counter is not cleared to 0 but holds its value.	p. 153 ☐
Chapter 9	Soft	A/D converter (μPD78F920x only)	Sampling time and A/D conversion time	The above sampling time and conversion time do not include the clock frequency error. Select the sampling time and conversion time such that Notes 2 and 3 above are satisfied, while taking the clock frequency error into consideration (an error margin maximum of ±5% when using the high-speed internal oscillator).	p. 158 ☐
	Hard		Block Diagram	In μPD78F920x, V_{SS} functions alternately as the ground potential of the A/D converter. Be sure to connect V_{SS} to a stabilized GND (= 0 V).	p. 159 ☐
				In μPD78F920x, V_{DD} functions alternately as the A/D converter reference voltage input. When using the A/D converter, stabilize V_{DD} at the supply voltage used (2.7 to 5.5 V).	p. 159 ☐
	Soft		ADM: A/D converter mode register	The above sampling time and conversion time do not include the clock frequency error. Select the sampling time and conversion time such that Notes 3 and 4 above are satisfied, while taking the clock frequency error into consideration (an error margin maximum of ±5% when using the high-speed internal oscillator).	p. 163 ☐