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	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f9488gk-9eu-a

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

LIST OF FIGURES (6/6)

3.4.4 Register addressing

[Function]

In the register addressing mode, general-purpose registers are accessed as operands. The general-purpose register to be accessed is specified by a register specification code or functional name in the instruction code. Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the instruction code.

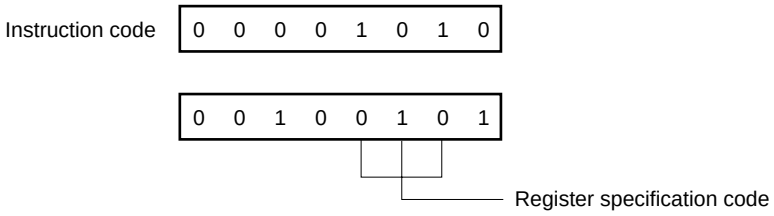
[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

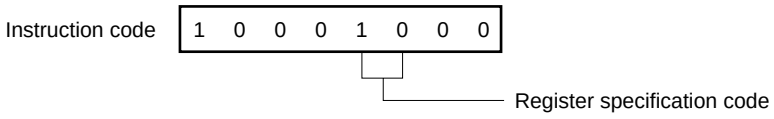
r and rp can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; When selecting the C register for r



INCW DE; When selecting the DE register pair for rp



(4) Subclock selection register (SSCK) (μPD78F9488, 78F9489 only)

This register is used to control the operation of the ×4 subsystem clock multiplication circuit.

SSCK is set via a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 00H.

Caution This register is valid only in the μPD78F9488 and 78F9489; however, writing to it in the μPD789488 and 789489 will simply make it invalid, causing no operational effect.

Figure 5-6. Subclock Selection Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SSCK	0	0	0	0	0	0	0	SCT	FF46H	Retained Note	R/W

SCT	Control of ×4 subsystem clock multiplication circuit
0	Operation stopped (subsystem clock source (32.768 kHz) supplied to the CPU)
1	Operation enabled (clock that is the subsystem clock multiplied by 8 (262 kHz) supplied to the CPU)

Note The register is set to 00H only by $\overline{\text{RESET}}$ input.

Cautions 1. Always set bits 1 to 7 to 0.

2. Write to the SCT flag prior to setting the CSS0 flag to 1 following the release of reset. Write operations following the first operation are invalid (input the $\overline{\text{RESET}}$ signal to rewrite).

5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as the standby mode.

- Main system clock f_X
- Subsystem clock f_{XT}
- CPU clock f_{CPU}
- Clock to peripheral hardware

The operation and function of the clock generator is determined by the processor clock control register (PCC), subclock oscillation mode register (SCKM), and subclock control register (CSS), as follows.

(a) The low-speed mode (1.6 μs: at 5.0 MHz operation) of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is being to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Three types of minimum instruction execution time (0.4 μs and 1.6 μs: main system clock (at 5.0 MHz operation), 122 μs: subsystem clock (at 32.768 kHz operation)) can be selected by the PCC, SCKM, and CSS settings. Also, the subsystem clock can be changed to a clock that uses a circuit to multiply the subclock by 4 via a mask option in the μPD789488 and 789489 or the subclock selection register (SSCK) in the μPD78F9488 and 78F9489 (15.26 μs: a circuit to multiply the subsystem clock by 4 is used).

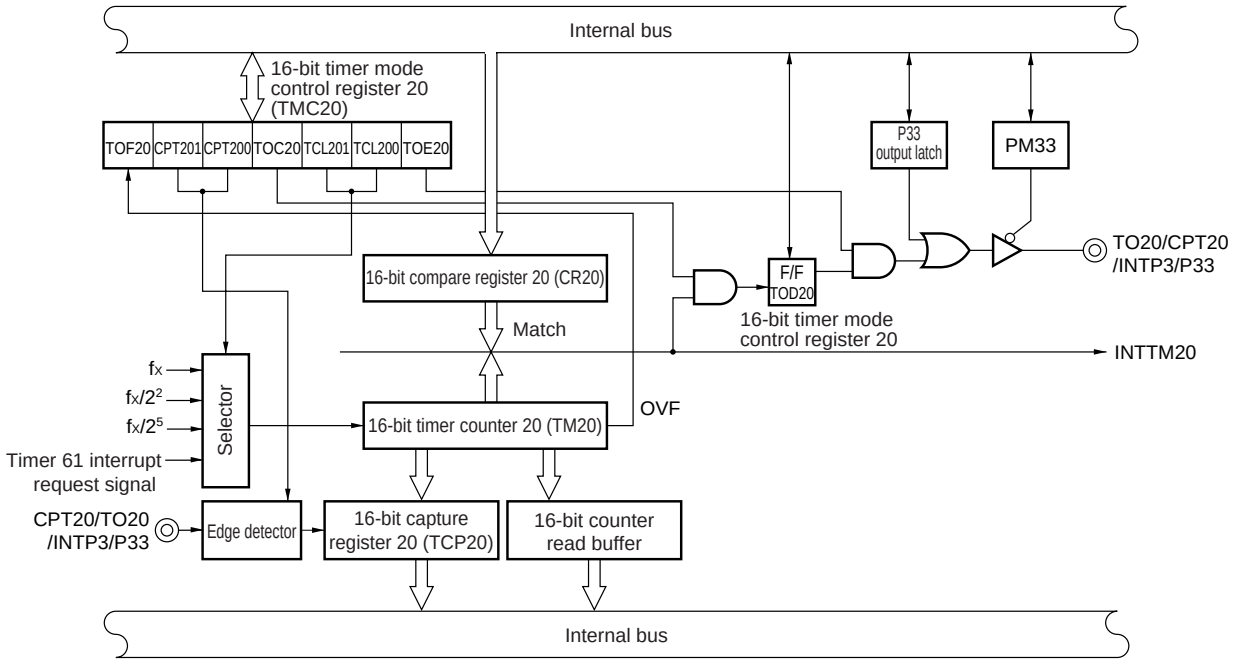
(c) Two standby modes, STOP and HALT, can be used with the main system clock selected. In a system where the subsystem clock is not used, setting bit 1 (FRC) of SCKM so that the on-chip feedback resistor cannot be used reduces current consumption in STOP mode. In a system where the subsystem clock is used, setting SCKM bit 0 to 1 can cause the subsystem clock to stop oscillation.

(d) CSS bit 4 (CSS0) can be used to select the subsystem clock so that low current consumption operation is used (122 μs: at 32.768 kHz operation).

(e) With the subsystem clock selected, it is possible to cause the main system clock to stop oscillating using bit 7 (MCC) of PCC. The HALT mode can be used, but the STOP mode cannot.

(f) The clock pulse for the peripheral hardware is generated by dividing the frequency of the main system clock, but the subsystem clock pulse is only supplied to 8-bit timer 50, the watch timer, and the LCD controller/driver. 8-bit timer 50, the watch timer, and the LCD controller/driver can therefore keep running even during standby. The other hardware stops when the main system clock stops because it runs based on the main system clock (except for external input clock operations).

Figure 6-1. Block Diagram of 16-Bit Timer 20



(1) 16-bit compare register 20 (CR20)

This 16-bit register is used to continually compare the value set to CR20 with the count value in 16-bit timer counter 20 (TM20) and to issue an interrupt request (INTTM20) when a match occurs.

CR20 is set via a 16-bit memory manipulation instruction. Values from 0000H to FFFFH can be set.

$\overline{\text{RESET}}$ input sets this register to FFFFH.

Caution To rewrite CR20 during a count operation, first set interrupt mask flag register 0 (MK0) to disable interrupts. Also, set inversion inhibited for the timer output data in 16-bit timer mode control register 20 (TMC20). If CR20 is rewritten while interrupts are enabled, an interrupt request may be issued at the point of rewrite.

(2) 16-bit timer counter 20 (TM20)

This is a 16-bit register that is used to count the count pulses.

TM20 can be read with a 16-bit memory manipulation instruction.

The counter is in free-running mode when the count clock is being input.

$\overline{\text{RESET}}$ input sets this counter to 0000H and restarts free-running mode.

Caution The count value after releasing STOP mode is undefined because the count operation occurred during the oscillation stabilization time.

(3) 16-bit capture register 20 (TCP20)

This is a 16-bit register used to capture the contents of 16-bit timer counter 20 (TM20).

TCP20 is set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undefined.

(1) Watch timer

An interrupt request (INTWT) occurs at an interval of 0.5 second when using either the 4.19 MHz main system clock or the 32.768 kHz subsystem clock.

Also, an interrupt request (INTWT) occurs at an interval of 1.0 seconds when using the 32.768 kHz subsystem clock via a setting in the watch timer interrupt time selection register (WTIM).

Caution An interval of 0.5 second cannot be created when using the 5.0 MHz main system clock. Instead, switch to the 32.768 kHz subsystem clock, and then create the 0.5-second interval.

(2) Interval timer

An interrupt request (INTWTI) occurs at preset intervals.

Table 8-1. Interval Time of Interval Timer

Interval Time	At f_X = 5.0 MHz	At f_X = 4.19 MHz	At f_{XT} = 32.768 kHz	At $f_{XT}/2$ = 16.384 kHz
$2^4 \times 1/f_W$	409.6 μs	488 μs	488 μs	976 μs
$2^5 \times 1/f_W$	819.2 μs	977 μs	977 μs	1.95 ms
$2^6 \times 1/f_W$	1.64 ms	1.95 ms	1.95 ms	3.90 ms
$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms	7.82 ms
$2^8 \times 1/f_W$	6.55 ms	7.81 ms	7.81 ms	15.6 ms
$2^9 \times 1/f_W$	13.1 ms	15.6 ms	15.6 ms	31.2 ms

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$, f_{XT}, or $f_{XT}/2$)

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

8.2 Configuration of Watch Timer

The watch timer includes the following hardware.

Table 8-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits × 1
Prescaler	9 bits × 1
Control registers	Watch timer mode control register (WTM) Watch timer interrupt time selection register (WTIM)

8.3 Control Registers for Watch Timer

The watch timer is controlled by the following registers.

- Watch timer mode control register (WTM)
- Watch timer interrupt time selection register (WTIM)

(1) Watch timer mode control register (WTM)

This register is used to control the watch timer count clock, operation enable/disable status, prescaler interval time, and the 5-bit counter operation.

WTM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 00H.

Figure 8-2. Format of Watch Timer Mode Control Register

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
WTM	WTM7	WTM6	WTM5	WTM4	0	0	WTM1	WTM0	FF4AH	00H	R/W

WTM7	Selection of watch timer count clock (f_W)
0	$f_X/2^7$ (39.1 kHz)
1	f_{XT} (32.768 kHz) or $f_{XT}/2$ (16.384 kHz)**Note**

WTM6	WTM5	WTM4	Selection of prescaler interval time
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
Other than above			Setting prohibited

WTM1	Control of 5-bit counter operation
0	Cleared after stopping operation
1	Start

WTM0	Watch timer operation enable/disable
0	Operation stopped (prescaler and timer are both cleared)
1	Operation enabled

Note This is the frequency (f_{XT} or $f_{XT}/2$) set via the watch timer interrupt time selection register (WTIM).

Remarks **1.** f_W: Watch timer clock frequency ($f_X/2^7$, f_{XT}, or $f_{XT}/2$)

2. f_X: Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

4. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

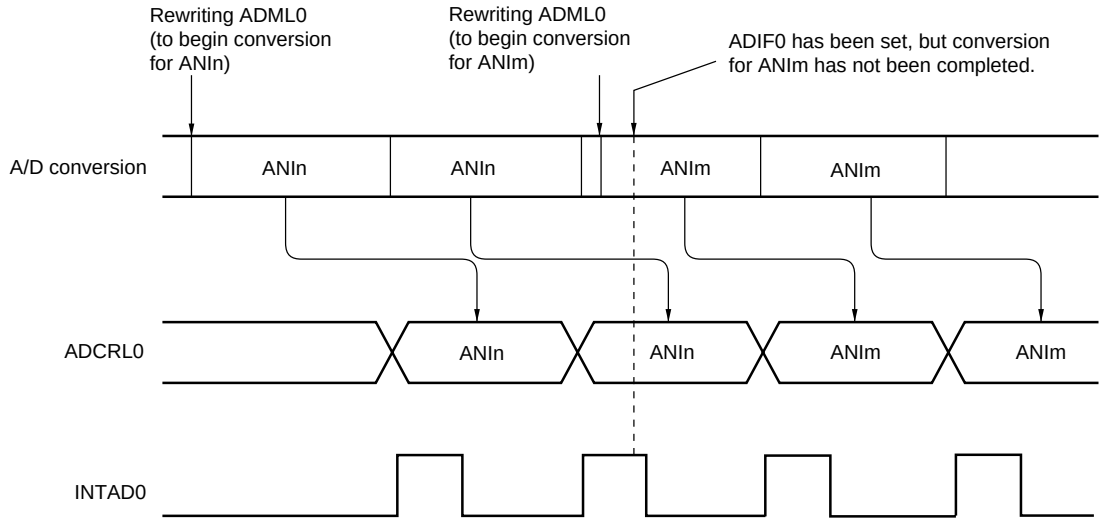
(9) Interrupt request flag (ADIF0)

Changing the contents of A/D converter mode register 0 (ADML0) does not clear the interrupt request flag (ADIF0).

If the analog input pins are changed during A/D conversion, therefore, the A/D conversion result and the conversion end interrupt request flag may reflect the previous analog input immediately before rewriting ADML0. In this case, ADIF0 may already be set if it is read-accessed immediately after ADML0 is rewritten, even when A/D conversion has not been completed for the new analog input.

In addition, when A/D conversion is restarted, ADIF0 must be cleared beforehand.

Figure 10-11. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0 to 7

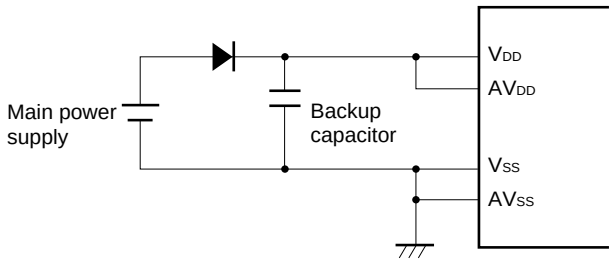
2. m = 0 to 7

(10) AV_{DD} pin

The AV_{DD} pin is used to supply power to the analog circuit. It is also used to supply power to the ANI0 to ANI7 input circuit.

If your application is designed to be changed to backup power, the AV_{DD} pin must be supplied with the same voltage level as the V_{DD} pin, as shown in Figure 10-12.

Figure 10-12. AV_{DD} Pin Handling

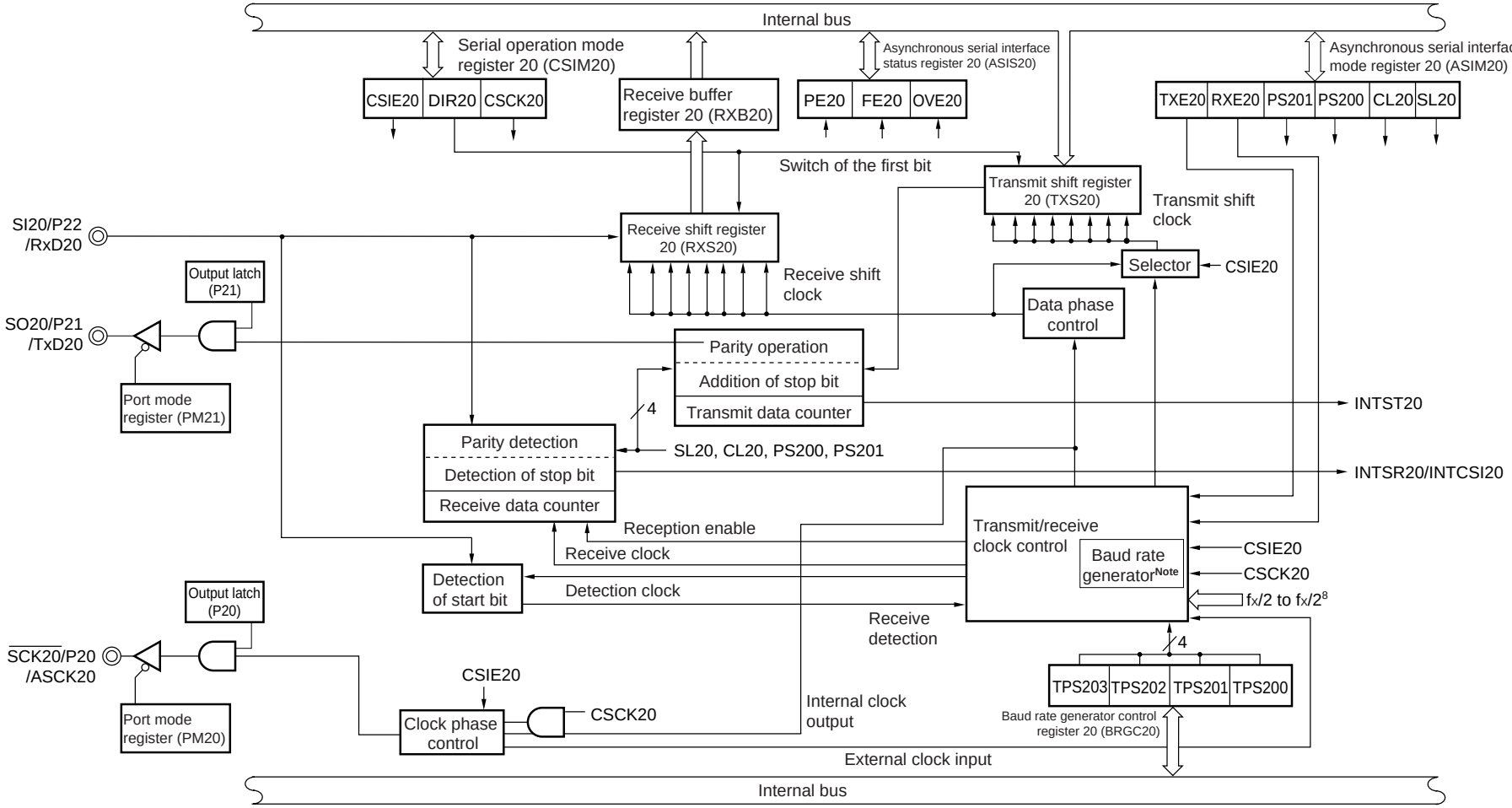


(11) AV_{DD} pin input impedance

A series resistor string of several ten of kΩ is connected between the AV_{DD} and AV_{SS} pins. Consequently, if the output impedance of the reference voltage supply is high, the reference voltage supply will form a series connection with the series resistor string, creating a large reference voltage differential.

★

Figure 11-1. Block Diagram of Serial Interface 20



Note See **Figure11-2** for the configuration of the baud rate generator.

12.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional clocked serial interface, such as the 75XL Series, 78K Series, and 17K Series.

Communication is performed using three lines: a serial clock ($\overline{SCK10}$), serial output (SO10), and serial input (SI10).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 1A0 (CSIM1A0), port mode register 2 (PM2), and port 2 (P2).

(a) Serial operation mode register 1A0 (CSIM1A0)

CSIM1A0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM1A0 to 00H.

Caution Set the port mode register 2 (PM2) in the 3-wire serial I/O mode as follows.

- **In the case of serial clock output (master transmission or master reception)**
  Set the $\overline{SCK10}$/P23 pin to output mode (PM23 = 0) and clear the output latch of P23 to 0.
- **In the case of serial clock input (slave transmission or slave reception)**
  Set the $\overline{SCK10}$/P23 pin to input mode (PM23 = 1).
- **In transmission or transmission/reception mode**
  Set the SO10/P24 pin to output mode (PM24 = 0) and clear the output latch of P24 to 0.
  Set the SI10/P25 pin to input mode (PM25 = 1).
- **In reception mode**
  Set the SI10/P25 pin to input mode (PM25 = 1).

13.7.2 Four-time-slice display example

Figure 13-13 shows how a 14-digit LCD panel having the display pattern shown in Figure 13-12 is connected to the segment signals (S0 to S27) and the common signals (COM0 to COM3) of the μPD789489 Subseries chip. This example displays the data "123456.78901234" in the LCD panel. The contents of the display data memory (addresses FA00H to FA1BH) correspond to this display.

The following description focuses on numeral "6." (5.) displayed as the ninth digit from the right. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the S16 and S17 pins according to Table 13-6 at the timing of the common signals COM0 to COM3; see Figure 13-12 for the relationship between the segment signals and LCD segments.

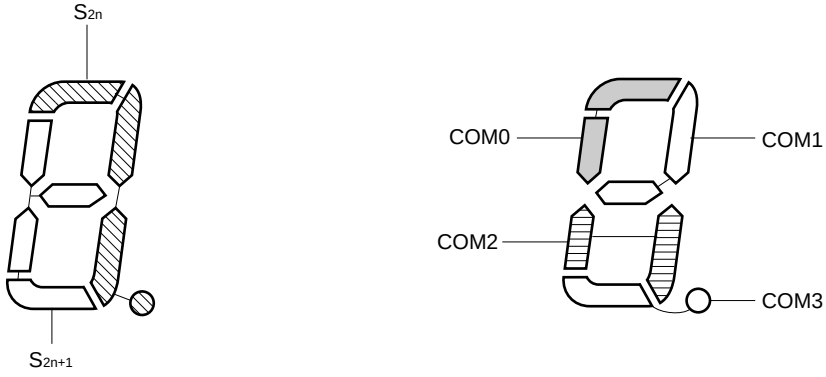
Table 13-6. Select and Deselect Voltages (COM0 to COM3)

Segment / Common	S16	S17
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 13-7, it is determined that the display data memory location (FA10H) that corresponds to S16 must contain 1101.

Figure 13-14 shows examples of LCD drive waveforms between the S16 signal and the common signals. When the select voltage is applied to S16 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 13-12. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 13

(2) Remote controller receive data register (RMDR)

This register holds the remote controller reception data. When the remote controller receive shift register (RMSR) overflows, the data in RMSR is transferred to RMDR. Bit 7 stores the last data, and bit 0 stores the first data. INTDFULL is generated at the same time as data is transferred from RMSR to RMDR.

RMDR is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RMDR to 00H.

When the remote controller operation is disabled (RMEN = 0), RMDR is cleared to 00H.

Caution When INTDFULL has been generated, read RMDR before the next 8-bit data is received. If the next INTDFULL is generated before the read operation is complete, RMDR is overwritten.

(3) Remote controller shift register receive counter register (RMSCR)

This is an 8-bit counter register used to indicate the number of valid bits remaining in the remote controller receive shift register (RMSR) when remote controller reception is complete (INTREND is generated). Reading the values of this register allows confirmation of the number of bits, even if the received data is in a format other than an integral multiple of 8 bits.

RMSCR is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RMSCR to 00H.

It is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- RMSR is read after INTREND has been generated.

Caution When INTREND has been generated, immediately read RMSCR before reading RMSR. If reading occurs at another timing, the value is not guaranteed.

Figure 15-2. Operation Examples of RMSR, RMSCR, and RMDR Registers When Receiving 1010101011111111B (16 Bits)

	RMSR								RMSCR	RMDR
	7	6	5	4	3	2	1	0		
After reset	0	0	0	0	0	0	0	0	00H	00000000B
Receiving 1 bit	1	0	0	0	0	0	0	0	01H	00000000B
Receiving 2 bits	0	1	0	0	0	0	0	0	02H	00000000B
Receiving 3 bits	1	0	1	0	0	0	0	0	03H	00000000B
…	…	…	…	…	…	…	…	…	…	…
Receiving 7 bits	1	0	1	0	1	0	1	0	07H	00000000B
Receiving 8 bits	0	1	0	1	0	1	0	1	00H	00000000B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDR transfer	0	0	0	0	0	0	0	0	00H	01010101B
Receiving 9 bits	1	0	0	0	0	0	0	0	01H	01010101B
Receiving 10 bits	1	1	0	0	0	0	0	0	02H	01010101B
…	…	…	…	…	…	…	…	…	…	…
Receiving 16 bits	1	1	1	1	1	1	1	1	00H	01010101B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDR transfer	0	0	0	0	0	0	0	0	00H	11111111B

(6) Key return mode register 01 (KRM01) (μPD789489, 78F9489 only)

This register is used to set the pin that is to detect the key return signal (falling edge of port 6).

KRM01 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 00H.

Figure 16-8. Format of Key Return Mode Register 01

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM01	KRM017	KRM016	KRM015	KRM014	0	0	0	KRM010	FFF4H	00H	R/W

KRM010	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P60 to P63 falling edge detection)

KRM01n	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P6n falling edge detection)

Remark n = 4 to 7

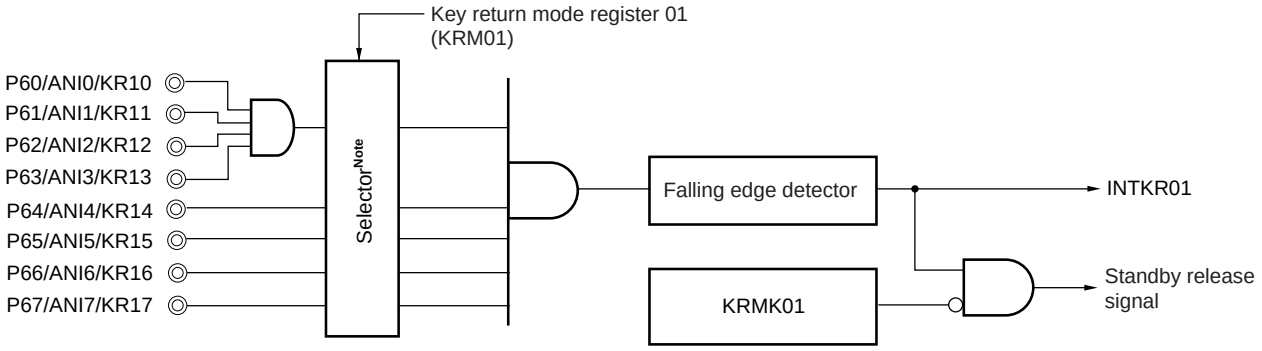
Cautions 1. Always set bits 1 to 3 to 0.

2. Before setting KRM01, set bit 5 of MK2 (KRMK01 = 1) to disable interrupts. To enable interrupts, clear KRMK01 after clearing bit 5 of IF2 (KRIF01 = 0)

3. If any of the pins specified for key return signal detection is low level, the key return signal cannot be detected even if a falling edge is generated at other key return pins.

4. When even one of the P60/ANI0/KR10 to P67/ANI7/KR17 pins is used as an A/D input, set KRM010 and KRM014 to KRM017 to 0.

Figure 16-9. Block Diagram of Falling Edge Detector



Note For selecting the pin to be used as falling edge input

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

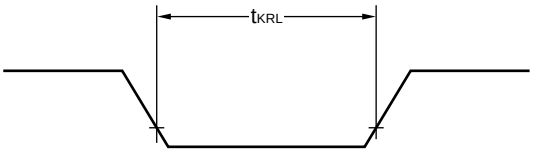
1st Operand \ 2nd Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

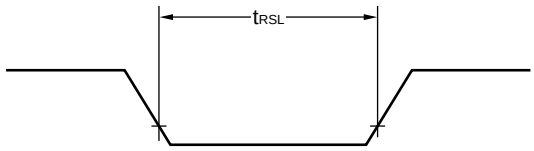
Key Return Input Timing

KR0 to KR7
(μPD789488, 78F9488) ,
KR00 to KR07, KR10 to KR17
(μPD789489, 78F9489)



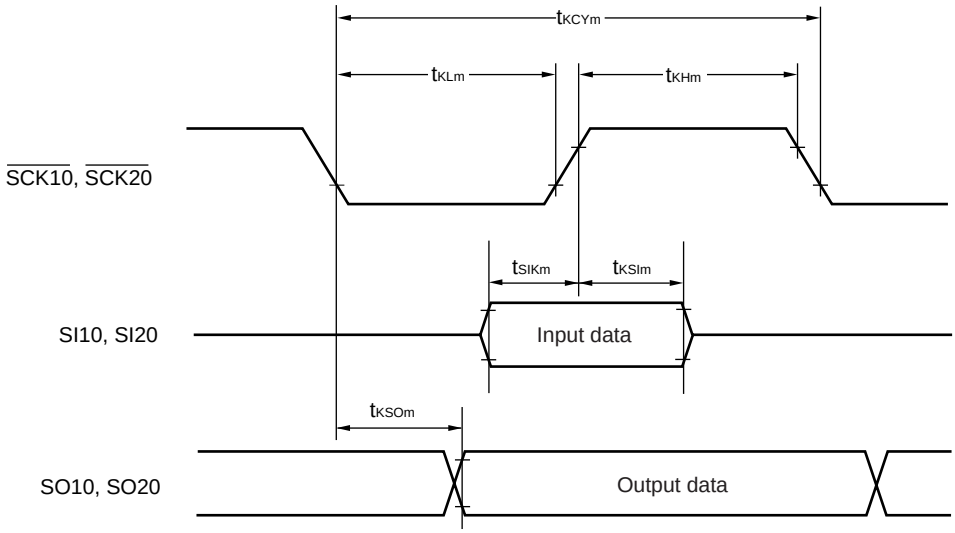
$\overline{\text{RESET}}$ Input Timing

$\overline{\text{RESET}}$



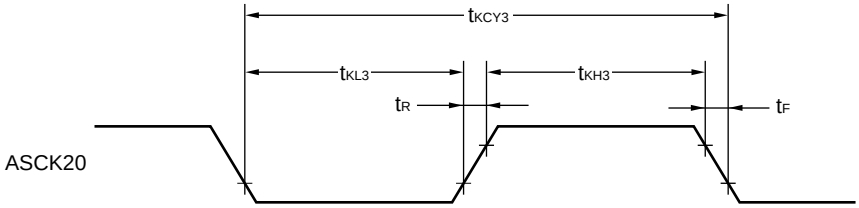
Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2, 4, 5

UART mode (external clock input):



10-Bit A/D Converter Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note]		4.5 V ≤ AV_{DD} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{DD} < 4.5 V		±0.4	±0.6	%FSR
		1.8 V ≤ AV_{DD} < 2.7 V		±0.8	±1.2	%FSR
Conversion time	t_{CONV}	4.5 V ≤ AV_{DD} ≤ 5.5 V	14		100	μs
		2.7 V ≤ AV_{DD} < 4.5 V	14		100	μs
		1.8 V ≤ AV_{DD} < 2.7 V	28		100	μs
Zero-scale error[Note]	AINL	4.5 V ≤ AV_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{DD} < 4.5 V			±0.6	%FSR
		1.8 V ≤ AV_{DD} < 2.7 V			±1.2	%FSR
Full-scale error[Note]	AINL	4.5 V ≤ AV_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{DD} < 4.5 V			±0.6	%FSR
		1.8 V ≤ AV_{DD} < 2.7 V			±1.2	%FSR
Non-integral linearity[Note]	INL	4.5 V ≤ AV_{DD} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{DD} < 4.5 V			±4.5	LSB
		1.8 V ≤ AV_{DD} < 2.7 V			±8.5	LSB
Non-differential linearity[Note]	DNL	4.5 V ≤ AV_{DD} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{DD} < 4.5 V			±2.0	LSB
		1.8 V ≤ AV_{DD} < 2.7 V			±3.5	LSB
Analog input voltage	V_{IAN}		0		AV_{DD}	V

Note Excludes quantization error (±0.05%)

Remark FSR: Full scale range

CHAPTER 25 RECOMMENDED SOLDERING CONDITIONS

The μPD789489 subseries should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 25-1. Surface Mounting Type Soldering Conditions (1/3)

(1) μPD789488GC-×××-8BT: 80-pin plastic QFP (14x14)
μPD78F9488GC-8BT: 80-pin plastic QFP (14x14)
★ μPD789489GC-×××-8BT: 80-pin plastic QFP (14x14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less	VP15-00-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Caution Do not use different soldering methods together (except for partial heating).

(2) μPD789488GK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12x12)
μPD78F9488GK-9EU: 80-pin plastic TQFP (fine pitch) (12x12)
★ μPD789489GK-×××-9EU: 80-pin plastic TQFP (fine pitch) (12x12)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Interface reflow	Package peak temperature: 235°C, Time:30 seconds max. (at 210°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time:40 seconds max. (at 200°C or higher), Count: Twice or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-107-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry peak, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

A.1 Software Package

SP78K0S Software package	Software tools for development of the 78K/0S Series are combined in this package. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S, and device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the OS used

μS××××SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

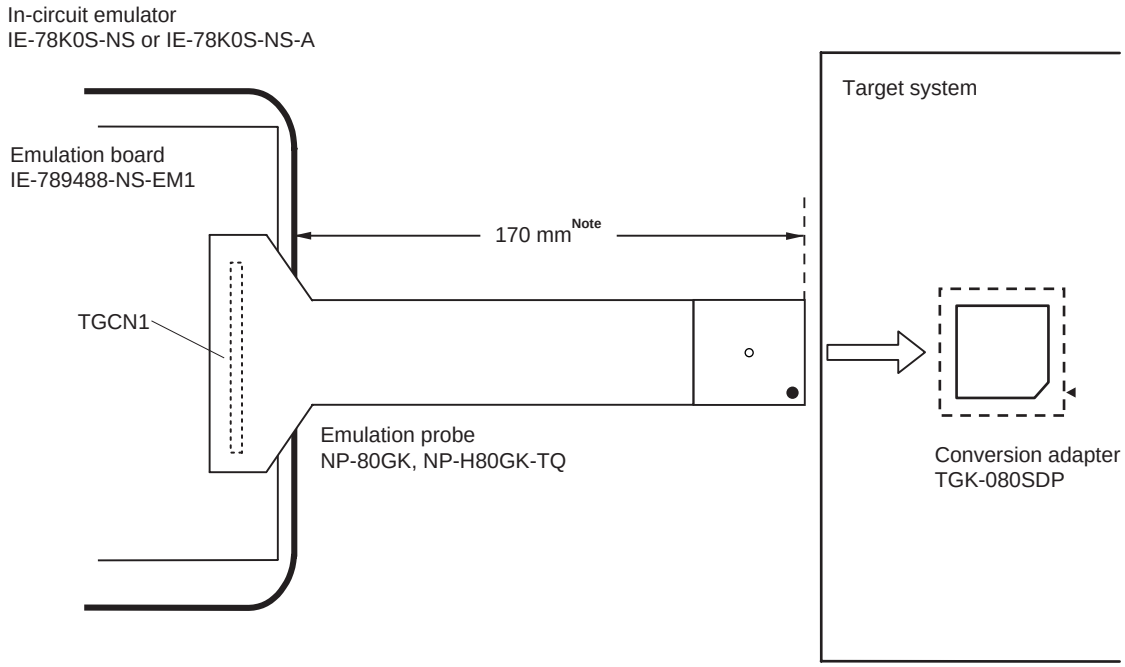
RA78K0S Assembler package	Program that converts program written in mnemonic into object codes that can be executed by a microcontroller. In addition, automatic functions to generate symbol tables and optimize branch instructions are also provided. Used in combination with a device file (DF789488) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used in the Windows environment by using the project manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C compiler package	Program that converts program written in C language into object codes that can be executed by a microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789488) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using the project manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789488[Note 1] Device file	File containing information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789488
CC78K0S-L[Note 2] C library source file	Source file of functions for generating object library included in the C compiler package. Necessary for changing the object library included in the C compiler package according to the customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789488 is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

(2) NP-80GK, NP-H80GK-TQ

Figure B-4. Distance Between In-Circuit Emulator and Conversion Adapter (80GK)



Note Distance when NP-80GK is used. When NP-H80GK-TQ is used, the distance is 370 mm.