Renesas Electronics America Inc - UPD78F1156AGK-GAK-AX Datasheet

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	Active
Core Processor	78K/0R
Core Size	16-Bit
Speed	20MHz
Connectivity	3-Wire SIO, I²C, LINbus, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	65
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f1156agk-gak-ax

2.2.15 AV_{REF1}

This is the D/A converter reference voltage input pin and the positive power supply pin of P110, P111, and D/A converter.

The voltage that can be supplied to AV_{REF1} varies as follows, depending on whether P110/ANI0, P111/ANO1 are used as digital I/Os or analog inputs.

Table 2-3. AV_{REF1} Voltage Applied to P110/ANO0, P111/ANO1 Pins

Analog/Digital	V_{DD} Condition	AV_{REF1} Voltage
Using at least one pin as an analog output and using all pins not as digital I/Os	$1.8\ V \le V_{DD} \le 5.5\ V$	$1.8\ V \le AV_{REF1} \le V_{DD} = EV_{DD}$
Pins used as analog outputs and digital I/Os are mixed[Note]	$2.7\ V \le V_{DD} \le 5.5\ V$	$2.7\ V \le AV_{REF1} \le V_{DD} = EV_{DD}$
	$1.8\ V \le V_{DD} < 2.7\ V$	AV_{REF1} has same potential as EV_{DD}, and V_{DD}
Using at least one pin as a digital I/O and using all pins not as analog outputs[Note]	$2.7\ V \le V_{DD} \le 5.5\ V$	$2.7\ V \le AV_{REF1} \le V_{DD} = EV_{DD}$
	$1.8\ V \le V_{DD} < 2.7\ V$	AV_{REF1} has same potential as EV_{DD}, and V_{DD}

Note AV_{REF1} is the reference for the I/O voltage of a port to be used as a digital port.
- High-/low-level input voltage (V_{IH5}/V_{IL5})
- High-/low-level output voltage (V_{OH2}/V_{OL2})

2.2.16 AV_{SS}

This is the ground potential pin of A/D converter, D/A converter, P20 to P27, and P110, P111. Even when the A/D converter and D/A converter are not used, always use this pin with the same potential as EV_{SS} and V_{SS}.

2.2.17 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

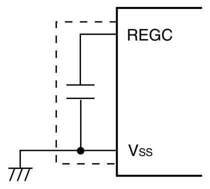
When the external reset pin is not used, connect this pin directly to EV_{DD} or via a resistor.

When the external reset pin is used, design the circuit based on V_{DD}.

2.2.18 REGC

This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47 μF is recommended.

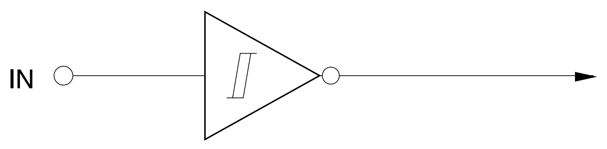
Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

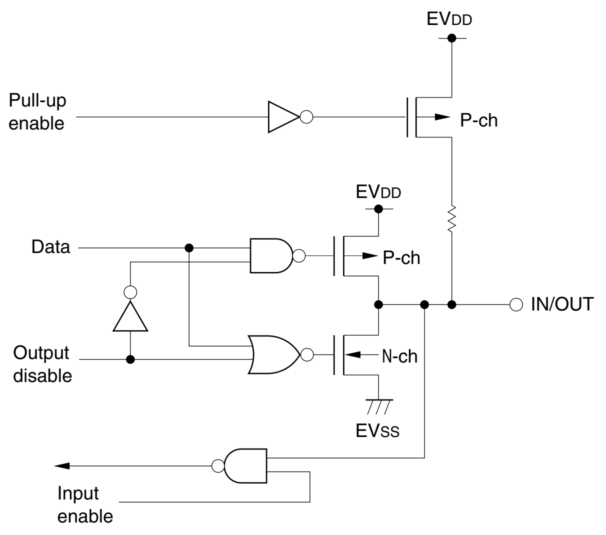
Figure 2-1. Pin I/O Circuit List (1/2)

Type 2

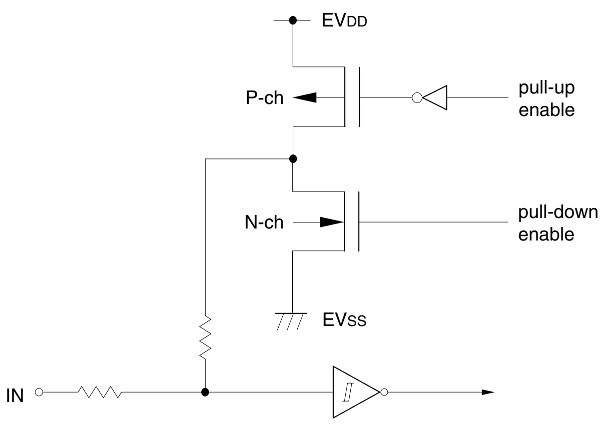


Schmitt-triggered input with hysteresis characteristics

Type 5-AG

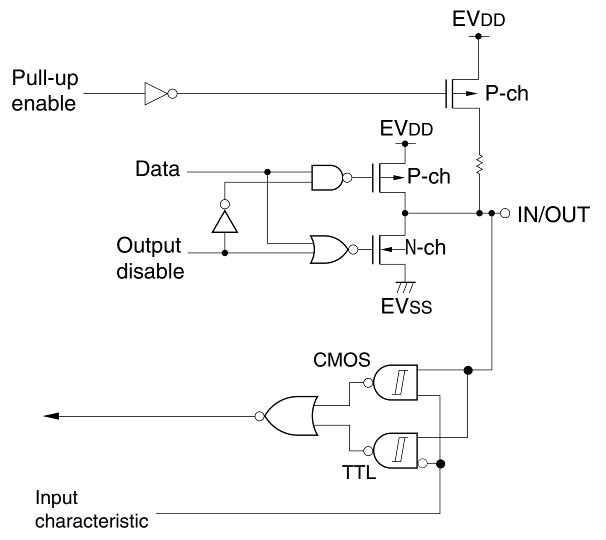


Type 2-W

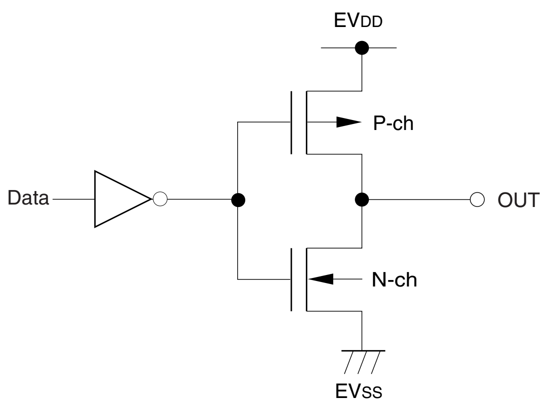


Schmitt-triggered input with hysteresis characteristics

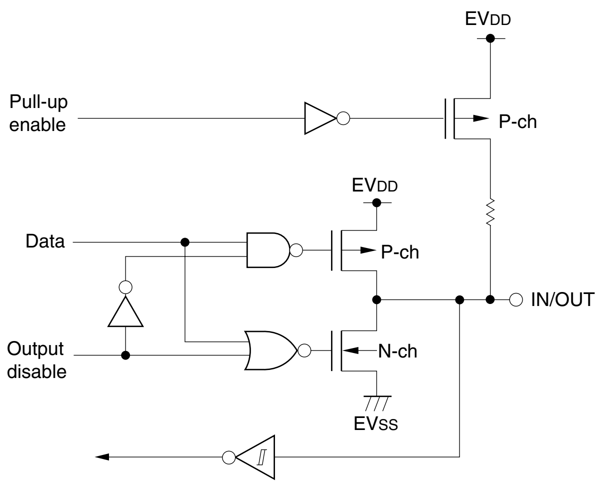
Type 5-AN



Type 3-C



Type 8-R



Remark The flash memory is divided into blocks (one block = 2 KB). For the address values and block numbers, see **Table 3-1 Correspondence Between Address Values and Block Numbers in Flash Memory**.

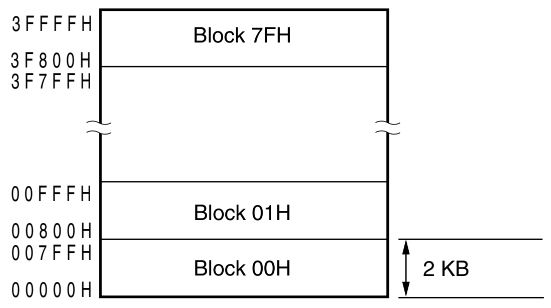
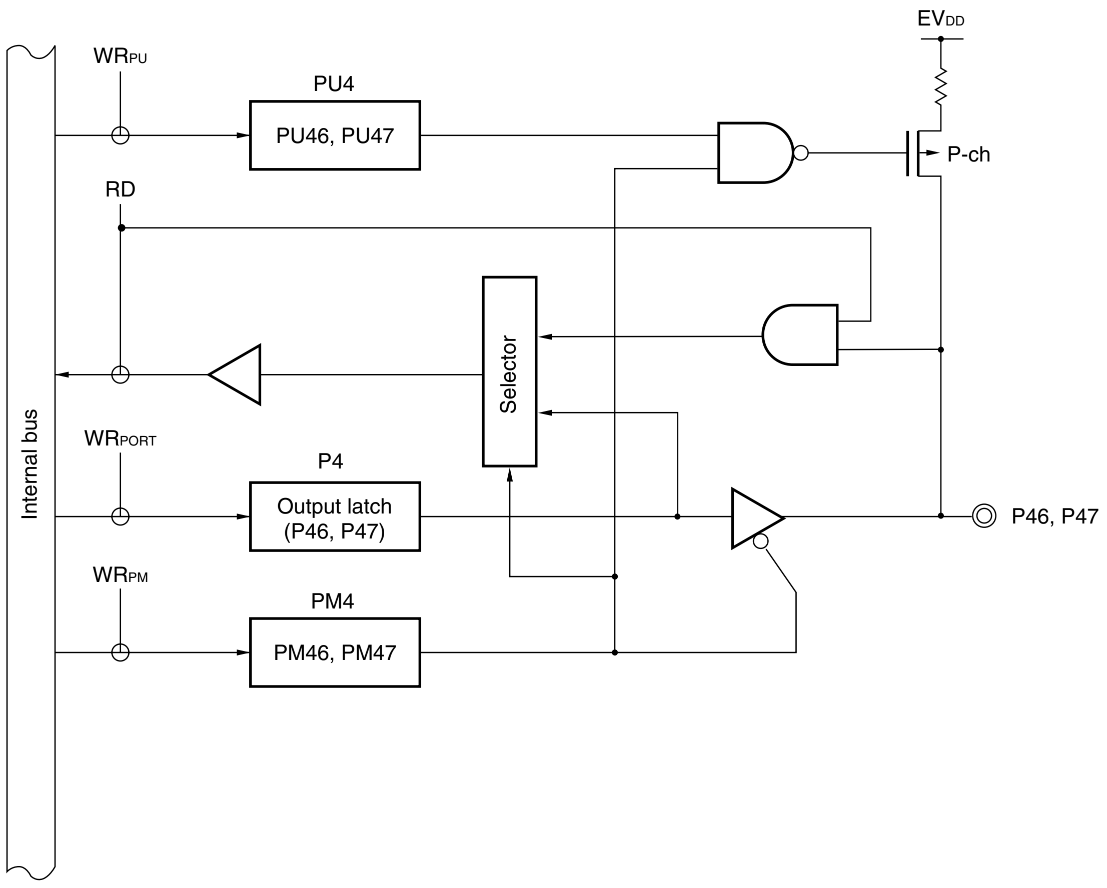


Figure 4-20. Block Diagram of P46 and P47



P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
RD:	Read signal
WR××:	Write signal

(3) Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU9, PU12, PU14)

These registers specify whether the on-chip pull-up resistors of P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P55, P64 to P67, P70 to P77, P90, P120, or P140 to P145 are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified in PU0, PU1, PU3 to PU7, PU9, PU12, and PU14. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU0, PU1, PU3 to PU7, PU9, PU12, and PU14.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 4-38. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU9	0	0	0	0	0	0	0	PU90	F0039H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 7, 9, 12,14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Figure 5-4. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

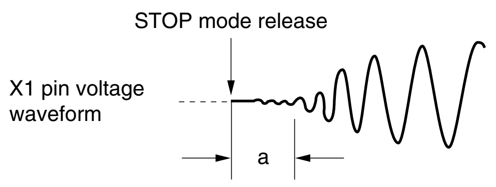
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.21 ms min.	13.11 ms min.

Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.
In the following cases, set the oscillation stabilization time of OSTS to the value greater than or equal to the count value which is to be checked by the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock with the X1 clock oscillating.**
  (Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after the STOP mode is released.)

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

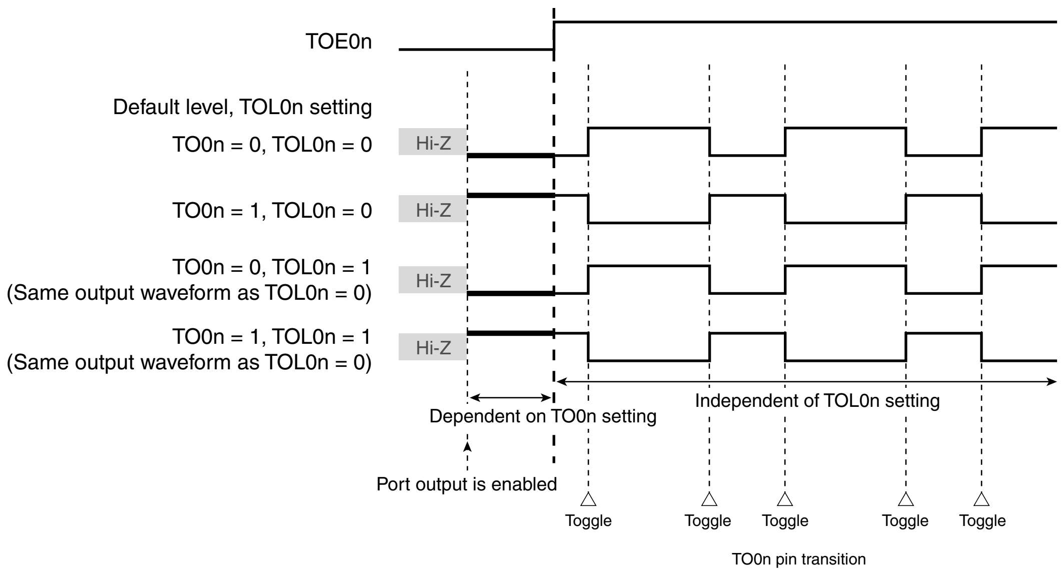
(2) Default level of TO0n pin and output level after timer operation start

The following figure shows the TO0n pin output level transition when writing has been done in the state of TOE0n = 0 before port output is enabled and TOE0n = 1 is set after changing the default level.

(a) When operation starts with TOM0n = 0 setting (toggle output)

The setting of TOL0n is invalid when TOM0n = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TO0n pin is reversed.

Figure 6-26. TO0n Pin Output Status at Toggle Output (TOM0n = 0)



Remarks 1. Toggle: Reverse TO0n pin output status

2. n = 0 to 7

(5) Sub-count register (RSUBC)

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter. It takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

RSUBC can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Cautions 1. When a correction is made by using the SUBCUD register, the value may become 8000H or more.

2. This register is also cleared by reset effected by writing the second count register.

3. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

Figure 7-6. Format of Sub-Count Register (RSUBC)

Address: FFF90H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

Address: FFF91H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(6) Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

SEC can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-7. Format of Second Count Register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

12.1.3 Simplified I^{2}C (IIC10, IIC20)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I^{2}C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master and does not have a function to detect wait states.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

* [Functions not supported by simplified I^{2}C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

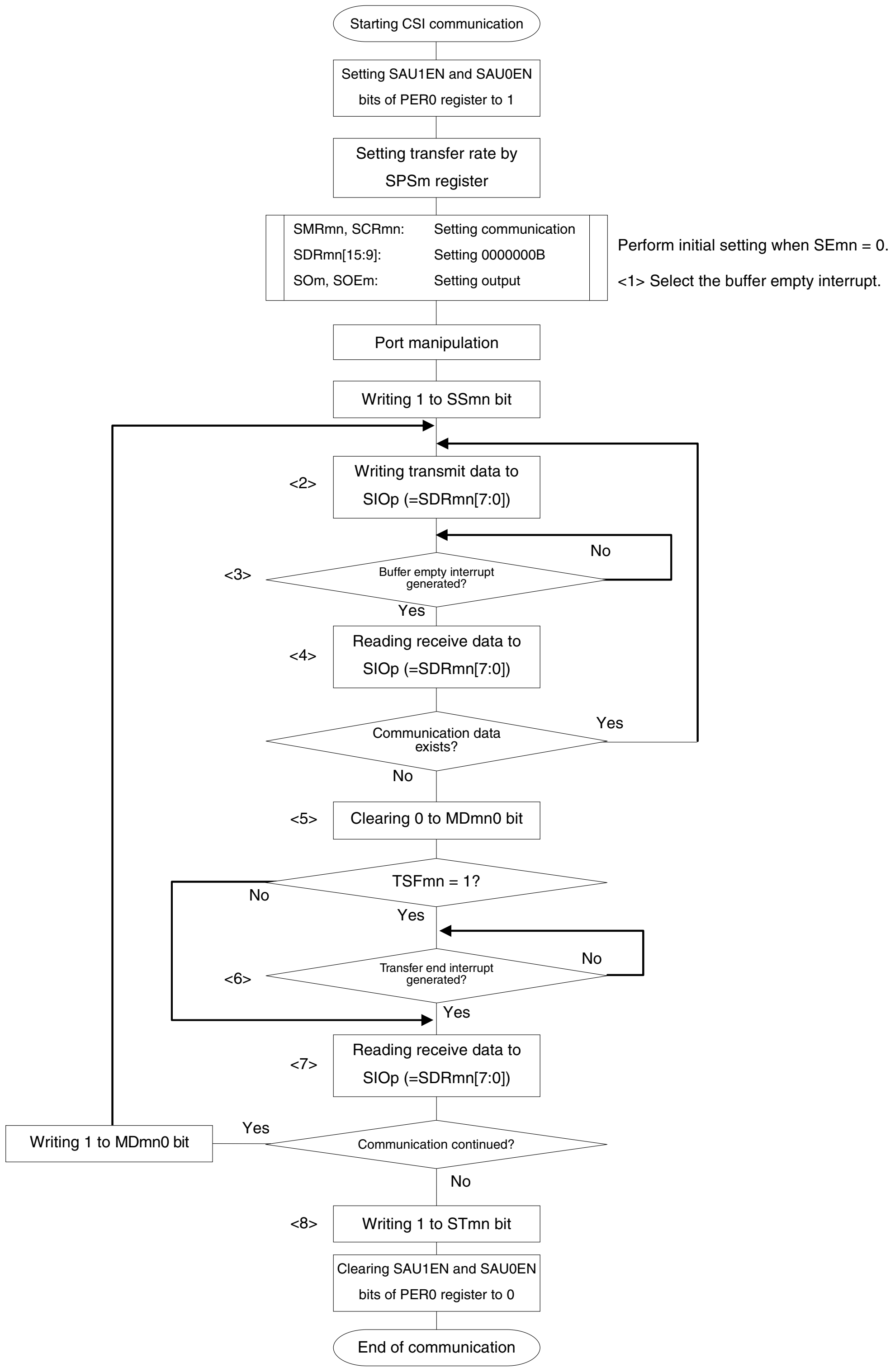
Note An ACK is not output when the last data is being received by writing 0 to the SOEmn (SOEm register) bit and stopping the output of serial communication data. See **12.7.3 (2) Processing flow** for details.

Remarks 1. To use the full-function I^{2}C bus, see **CHAPTER 13 SERIAL INTERFACE IIC0**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

Figure 12-69. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)

<R>


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting transfer rate by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEmn = 0.
<1> Select the buffer empty interrupt.
Port manipulation
Writing 1 to SSmn bit
<2> Writing transmit data to SIOp (=SDRmn[7:0])
<3> Buffer empty interrupt generated?
No
Yes
<4> Reading receive data to SIOp (=SDRmn[7:0])
Communication data exists?
Yes
No
<5> Clearing 0 to MDmn0 bit
TSFmn = 1?
No
Yes
<6> Transfer end interrupt generated?
No
Yes
<7> Reading receive data to SIOp (=SDRmn[7:0])
Communication continued?
Yes
Writing 1 to MDmn0 bit
No
<8> Writing 1 to STmn bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

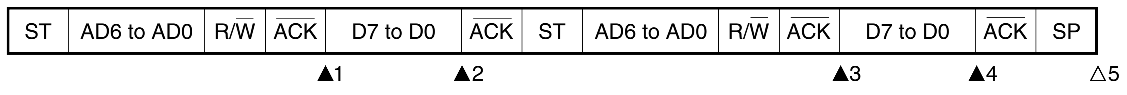
Cautions 1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

<R> **2. Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-68 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

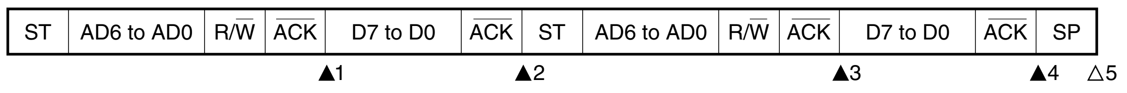
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

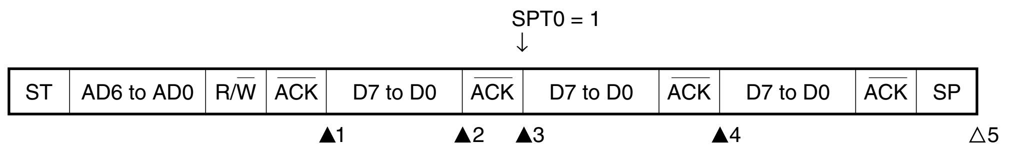


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

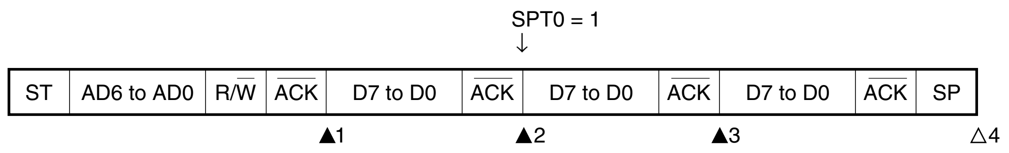
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICS0 = 1000×100B (Clears WTIM0 to 0)
▲4: IICS0 = 01000100B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets SPT0 to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF2H	0	0	0	0	0	0	0	PIF11

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Be sure to clear bits 1 to 7 of IF2H to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

3. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.
MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H can be set by a 1-bit or 8-bit memory manipulation instruction.
When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

16.4 Interrupt Servicing Operations

16.4.1 Maskable interrupt acknowledgment

A maskable interrupt becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 16-4 below.

For the interrupt request acknowledgment timing, see **Figures 16-8** and **16-9**.

Table 16-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	14 clocks

Note If an interrupt request is generated just before the RET instruction, the wait time becomes longer.

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 16-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

Operation example 2: When used as interrupt

Interrupt requests may be generated frequently.

Take the following action.

<Action>

Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

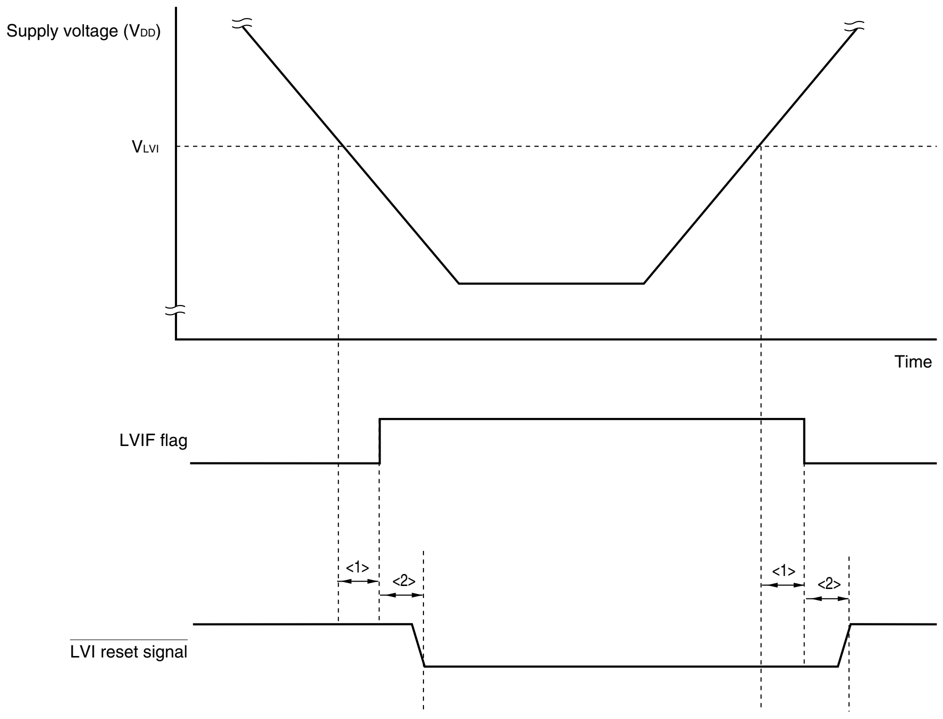
(2) Delay from the time LVI reset source is generated until the time LVI reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVI detection voltage (V_{LVI}) until the time LVI reset has been generated.

In the same way, there is also some delay from the time LVI detection voltage (V_{LVI}) ≤ supply voltage (V_{DD}) until the time LVI reset has been released (see **Figure 21-12**).

See the timing in **Figure 20-2 (2) When LVI is ON upon power application (option byte: LVIOFF = 0)** for the reset processing time until the normal operation is entered after the LVI reset is released.

Figure 21-12. Delay from the time LVI reset source is generated until the time LVI reset has been generated or released



<1> : Minimum pulse width (200 μs (MIN.))

<2> : Detection delay time (200 μs (MAX.))

Table 24-3. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

24.6.3 Selecting communication mode

Communication mode of the 78K0R/KF3 as follows.

Table 24-4. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (dedicated single-line UART)	UART-ch0	115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps	–	–	TOOL0

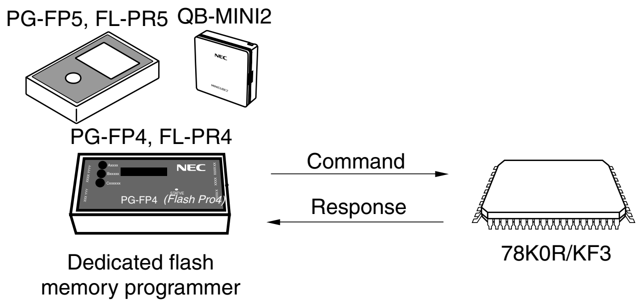
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

24.6.4 Communication commands

The 78K0R/KF3 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/KF3 are called commands, and the signals sent from the 78K0R/KF3 to the dedicated flash memory programmer are called response.

Figure 24-9. Communication Commands



The flash memory control commands of the 78K0R/KF3 are listed in the table below. All these commands are issued from the programmer and the 78K0R/KF3 perform processing corresponding to the respective commands.

26.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCDADJ register.

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 1: 99 + 89 = 188

Instruction	A Register	CY flag	AC Flag	BCDADJ Register
`MOV A, #99H ; <1>`	99H	–	–	–
`ADD A, #89H ; <2>`	22H	1	1	66H
`ADD A, !BCDADJ ; <3>`	88H	1	0	–

Examples 2: 85 + 15 = 100

<R>

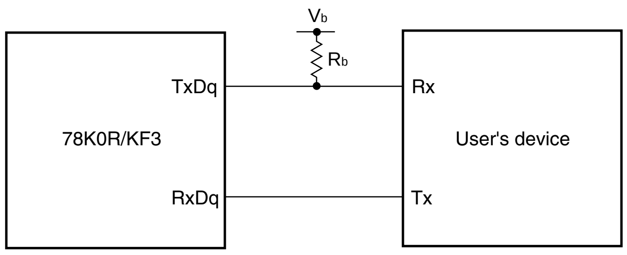
Instruction	A Register	CY flag	AC Flag	BCDADJ Register
`MOV A, #85H ; <1>`	85H	–	–	–
`ADD A, #15H ; <2>`	9AH	0	0	66H
`ADD A, !BCDADJ ; <3>`	00H	1	1	–

Examples 3: 80 + 80 = 160

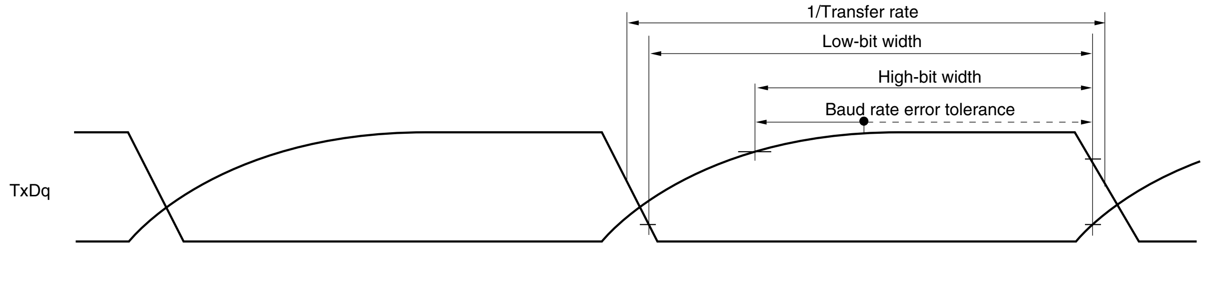
Instruction	A Register	CY flag	AC Flag	BCDADJ Register
`MOV A, #80H ; <1>`	80H	–	–	–
`ADD A, #80H ; <2>`	00H	1	0	60H
`ADD A, !BCDADJ ; <3>`	60H	1	0	–

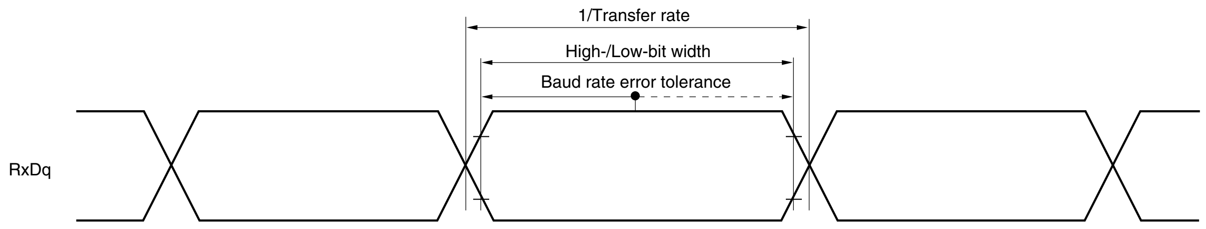
(2) Serial interface: Serial array unit (10/18)

UART mode connection diagram (During communication at different potential)



UART mode bit width (During communication at different potential)





Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMg registers.

Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, V_b[V]: Communication line voltage

2. q: UART number (q = 1, 2), g: PIM and POM number (g = 0, 14)

3. UART0 and UART3 cannot communicate at different potential. Use UART1 and UART2 for communication at different potential.

(31/34)

Chapter	Classification	Function	Details of Function	Cautions	Page	
Chapter 28	Hard	Electrical specifications (standard products)	Recommended oscillator constants	The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KF3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.	p.735	☐
			DC characteristics	P02 to P04, P43, P45, P142 to P144 do not output high level in N-ch open-drain mode.	p.736	☐
				The maximum value of V_{IH} of pins P02 to P04, P43, P45, and P142 to P144 is V_{DD}, even in the N-ch open-drain mode.	pp.738, 739	☐
				For P122/EXCLK, the value of V_{IH} and V_{IL} differs according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.	pp.738, 739	☐
	Soft		During communication at same potential (UART mode) (dedicated baud rate generator output)	Select the normal input buffer for RxDi and the normal output mode for TxDi by using the PIMg and POMg registers.	p.754	☐
			During communication at same potential (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output)	Select the normal input buffer for SIj and the normal output mode for SOj and $\overline{\text{SCKj}}$ by using the PIMg and POMg registers.	p.755	☐
			During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input)	Select the normal input buffer for SIj and $\overline{\text{SCKj}}$ and the normal output mode for SOj by using the PIMg and POMg registers.	p.756	☐
			During communication at same potential (simplified I^2C mode)	Select the normal input buffer and the N-ch open-drain output (V_{DD} tolerance) mode for SDAr and the normal output mode for SCLr by using the PIMg and POMg registers.	p.759	☐
			During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output)	Select the TTL input buffer for RxDq and the N-ch open-drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMg registers.	pp.760, 761, 763	☐

(7/20)

Edition	Description	Chapter
3rd edition	Addition of **Table 6-4 Operations from Count Operation Enabled State to TCR0n Count Start**, and (a) through (e)	CHAPTER 6 TIMER ARRAY UNIT
	Addition of description to **6.3 (11) Timer output level register 0 (TOL0)**	
	Change of description of **6.3 (12) Timer output mode register 0 (TOM0)**	
	Change of **Figure 6-20 Format of Timer Output Mode Register 0 (TOM0)** and **Remark**	
	Change of description of bit 7 and addition of **Note** in **Figure 6-22 Format of Noise Filter Enable Register 1 (NFEN1)**	
	Addition of **6.4 Channel Output (TO0n pin) Control**	
	Addition of **6.5 Channel Input (TI0n Pin) Control**	
	Addition of MD0n0 bit condition to titles in the following figures • **Figure 6-37 Example of Basic Timing of Operation as Interval Timer/Square Wave Output** **(MD0n0 = 1)** • **Figure 6-45 Example of Basic Timing of Operation as Frequency Divider (MD0n0 = 1)** • **Figure 6-49 Example of Block Diagram of Operation as Input Pulse Interval Measurement** **(MD0n0 = 0)**	
	Change of description of **6.7.3 Operation as frequency divider**	
	Change of description of **6.8.3 Operation as multiple PWM output function**	
	Change of clear conditions of real-time counter	CHAPTER 7 REAL-TIME COUNTER
	Change of description and **Caution 1** in **Figure 7-2 Format of Peripheral Enable Register 0 (PER0)**	
	Addition of **Caution 2** to **Figure 7-2 Format of Peripheral Enable Register 0 (PER0)**	
	Addition of **Caution** to **Figure 7-4 Format of Real-Time Counter Control Register 1 (RTCC1)**	
	Addition of **Caution** to **Figure 7-5 Format of Real-Time Counter Control Register 2 (RTCC2)**	
	Change of **Note 2** in **7.3 (5) Sub-count register (RSUBC)**	
	Change of bit name in **Figure 7-17 Format of Alarm Week Register (ALARMWW)**	
	Addition of **Caution 2** to **10.3 (1) Peripheral enable register 0 (PER0)**	CHAPTER 10 A/D CONVERTER
	Change of **Table 10-2 A/D Conversion Time Selection**	
	Addition of **Caution 2** to **11.3 (1) Peripheral enable register 0 (PER0)**	CHAPTER 11 D/A CONVERTER
	Addition of **Caution 3** to **12.3 (1) Peripheral enable register 0 (PER0)**	CHAPTER 12 SERIAL ARRAY UNIT
	Changes of **Figure 12-7 Format of Serial Communication Operation Setting Register mn (SCRmn)**	
	Addition of description to **12.3 (13) Serial output level register m (SOLm)**	
	Changes of bits 1 and 3 in **Figure 12-16 Format of Serial Output Level Register m (SOLm)**	
	Changes of setting of **(a) Serial output register m (SOm)**, **(d) Serial output level register m (SOLm)**, and **Note** in **Figure 12-66 Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2, UART3)**	