Date: Jul. 28, 2025

RENESAS TECHNICAL UPDATE

TOYOSU FORESIA, 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan
Renesas Electronics Corporation

Product Category	MPU/MCU		Document No.	TN-RL*-A0147A/E	Rev.	1.00
Title	Correction for Incorrect Description Notice Descriptions of errors in the RL78/F23, F24 User's Manual: Hardware Rev.1.10		Information Category	Technical Notification		
Applicable Product	RL78/F23, F24 Group	Lot No. All lots	Reference Document	RL78/F23, F24 User's Manual: Hardware Rev.1.10 (R01UH0944EJ0110)		

This document describes misstatements found in RL78/F23, F24 User's Manual: Hardware Rev.1.10 (R01UH0944EJ0110).

Corrections:

No	Corrections	R01UH0944EJ0110	Pages in this document
1	Addition of precautions when using PWM Option Unit A (PWMOPA)	P.553, P.654	P.2
2	Add description about IICWL, IICWH register settings	P.1037, P.1041	P.3
3	Correct CAN interface (RS-CANFD lite) typos	P.1393, P.1474, P.1499	P.4
4	Correction of typo in Example of Settings for SNOOZE Mode Status Output	P.1611	P.6
5	Add explanation of bit 2 in the CFERRCTLR register	P.1728	P.8
6	Add explanation of FLWE register	P.1735	P.8
7	Correction of typos in flash memory	P.1756, P.1764, P.1778, P.1807, P.1813, P.1814, P.1816, P.1817	P.8
8	Add Flash Memory description	P.1764, P.1777, P.1813	P.12
9	Correction of typos in electrical characteristics	P.1873, P.1874, P.1884, P.1890, P.1926, P.1927, P.1937, P.1943, P.1978, P.1979, P.1989, P.1995	P.13
10	Add description of package drawing (32-pin product)	P.2006	P.14

Incorrect: Bold with underline; Correct: Gray hatched

No.1: Addition of precautions when using PWM Option Unit A (PWMOPA)

- **8.2.32, Add Caution 7 to the PWMDLY0 register.**

(Incorrect)

Cautions 1. Set the PWMDLY0 register before outputting a PWM signal.

:

6. Even if this register is set for a delay, this doesn't affect the operation of other pin functions multiplexed on the same pin as the TRDIOji pin function (j = A, B, C, D, i = 0, 1).

(Correct)

Cautions 1. Set the PWMDLY0 register before outputting a PWM signal.

:

6. Even if this register is set for a delay, this doesn't affect the operation of other pin functions multiplexed on the same pin as the TRDIOji pin function (j = A, B, C, D, i = 0, 1).

7. When using reset synchronous PWM mode, complementary PWM mode, PWM3 mode, or extended complementary PWM mode, and when using PWM option unit A (PWMOPA), the TRDji[1,0] bits for the target pin should be set to 00B (no delay) (i = 0, 1, j = A, B C, D).

- **8.6.3.8, Add (8) and (9) to the Cautions.**

(Incorrect)

(1) The following table lists the priorities when pulse output forced cutoff of timer RDe operates simultaneously with output cutoff of PWMOPA.

:

(7) Set the input enabled level period of comparator 0 and INTP0 to one cycle of PWMOPA operating clock or longer.

(Correct)

(1) The following table lists the priorities when pulse output forced cutoff of timer RDe operates simultaneously with output cutoff of PWMOPA.

:

(7) Set the input enabled level period of comparator 0 and INTP0 to one cycle of PWMOPA operating clock or longer.

(8) When using reset synchronous PWM mode, complementary PWM mode, PWM3 mode, or extended complementary PWM mode, and when using PWM option unit A (PWMOPA), the TRDji[1,0] bits in the PWMDLY0 register for the target pin should be set to 00B (no delay) (i = 0, 1, j = A, B, C, D).

(9) When using the PWM function and extended PWM mode, if the PWM output is delayed by the PWMDLY0 register and PWM option unit A (PWMOPA) is used, the level held in the output delay circuit just before the output is cutoff is output from the target pin (output delay enabled) for the number of pulses equal to the delay setting value —1 after the forced cutoff by PWMOPA is released.

No.2: Add description about IICWL, IICWH register settings

- **16.3.7, Moved the remarks for the IICW0 register to the description section and add a remark.**

(Incorrect)

16.3.7 IICA High-level Width Setting Register 0 (IICWH0)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA.
The IICWH0 register can be set by an 8-bit memory manipulation instruction.
Set the IICWH0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

Remark For how to set the transfer clock by using the IICWL0 and IICWH0 registers, see **16.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

(Correct)

16.3.7 IICA High-level Width Setting Register 0 (IICWH0)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA.
The IICWH0 register can be set by an 8-bit memory manipulation instruction.
Set the IICWH0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).
For details about setting the IICWH0 register, see 17.4.2 Setting Transfer Clock by Using IICWL0 and IICWH0 registers.
Reset signal generation sets this register to FFH.

Remark The minimum serial clock cycle is (IICWL0 + 1) + (IICWH0 + 1).

- **16.3.7, Add caution 3 to the setting clock by using IICWL0 and IICWH0 registers.**

(Incorrect)

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (min.)
Fast mode plus: f_{CLK} = 10 MHz (min.)
Normal mode: f_{CLK} = 1 MHz (min.)

(Correct)

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (min.)
Fast mode plus: f_{CLK} = 10 MHz (min.)
Normal mode: f_{CLK} = 1 MHz (min.)
3. The minimum serial clock cycle is (IICWL0 + 1) + (IICWH0 + 1). Determine the values to be set in the IICWL0 and IICWH0 registers by considering the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals.

No.3: Correct CAN interface (RS-CANFD lite) typos

- **18.5.9, Correction of typo in CFIDH.CFRTR bit.**

(Incorrect)

- CFRTR

This bit selects whether a Data Frame or a Remote Frame will be transmitted from or was received in the FIFO Buffer.

Note There are no remote frames in CAN-FD format. In case a CAN-FD frame was received (RX mode) the register reflects the state of the received value (RRS bit in FD frame format). In case of CAN-FD transmission (TX mode CFFDCSTS.CFFDF = 1) the bit is always transmitted dominant (Data Frame).

In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
In RX mode, this bit can only be read data from FIFO buffers.

(Correct)

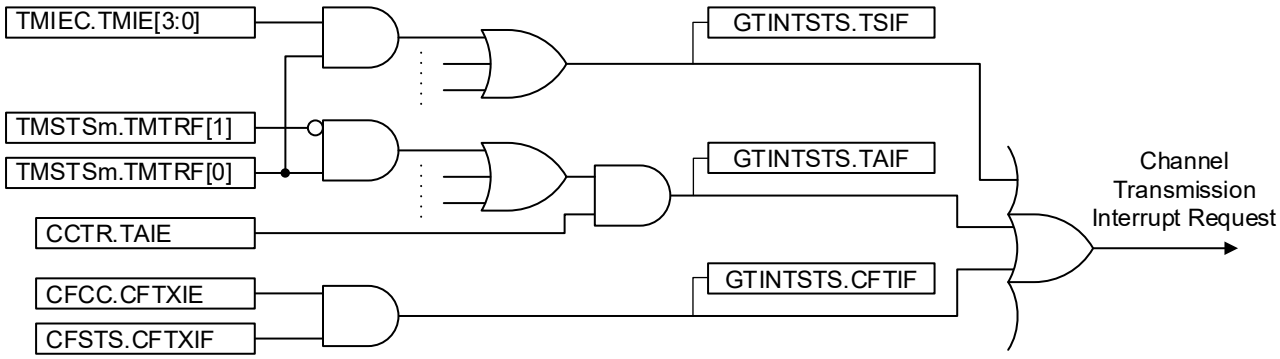
- CFRTR

This bit selects whether a Data Frame or a Remote Frame will be transmitted from or was received in the FIFO Buffer.
In case of CFCC.CFM is 1 (TX mode), sets the data format of message to be transmission from the common FIFO buffer.
Cautions1. There are no Remote frames in CAN-FD format. In case a CAN-FD frame was received (RX mode) the register reflects the state of the received value (RRS bit in FD frame format).
2. If CFFDCSTS.CFFDF is 1 (CAN-FD frame), a dominant (Data frame) is transmitted.

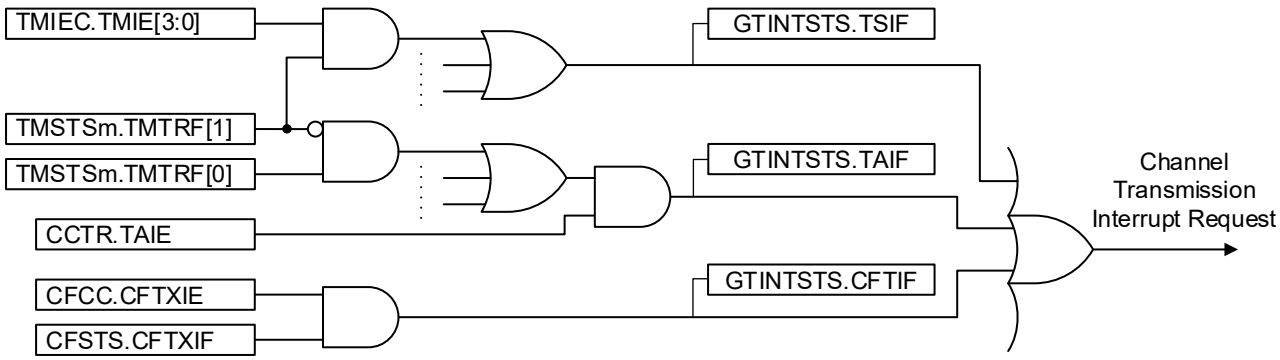
In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
In RX mode, this bit can only be read data from FIFO buffers.

- **18.12.1, Figure 18–45: Correction of typo in TMSTSm.TMTRF[1:0] bit in figure.**

(Incorrect) Generate from TMSTSm.TMTRF[0] to GTINTSTS.TSIF.



(Correct) Generate from TMSTSm.TMTRF[1] to GTINTSTS.TSIF.



● **18.17, Add the explanation shown in "18.12.2" to the Notes.**

(Incorrect)

- For the clock source setting for CAN Module, refer to the following:
  - f_{MP} and f_{CLK} for CAN module are should be supplied with same clock source. Also, select f_{MP} / 2 in the f_{MP} Clock Division Register (MDIV) so that $f_{CLK} = f_{MP} / 2$.
  - The high-speed on-chip oscillator can be used for f_{MP} and f_{CLK}, but it's prohibited for f_{CAN} (communication clock).
  - When using X1 clock for f_{CAN} (communication clock) with setting of GCFG.DCS bit = 1, the frequency must be equal or less than f_{CLK}.

(Correct)

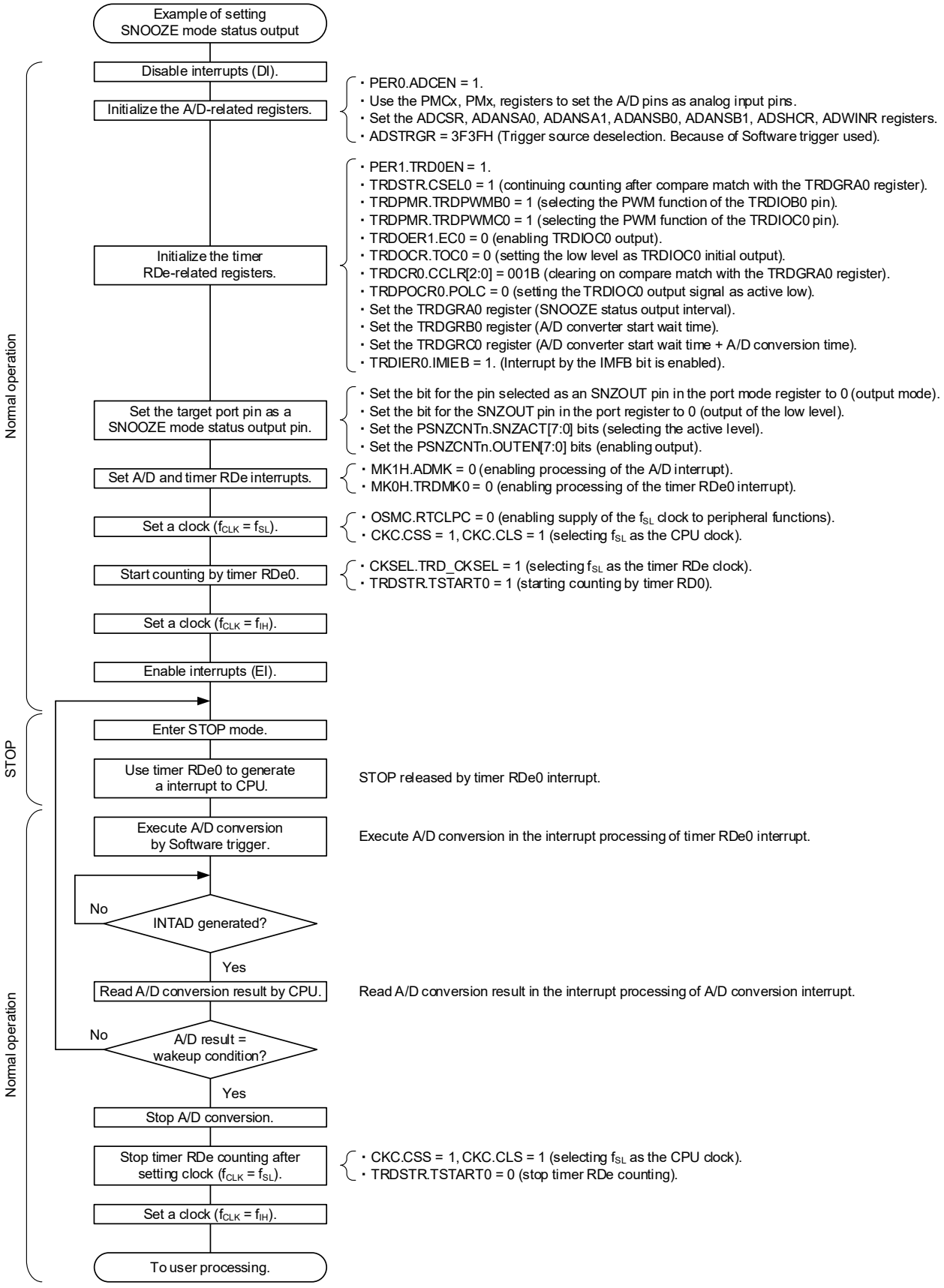
- For the clock source setting for CAN Module, refer to the following:
  - f_{MP} and f_{CLK} for CAN module are should be supplied with same clock source. Also, select f_{MP} / 2 in the f_{MP} Clock Division Register (MDIV) so that $f_{CLK} = f_{MP} / 2$.
  - The high-speed on-chip oscillator can be used for f_{MP} and f_{CLK}, but it's prohibited for f_{CAN} (communication clock).
  - When using X1 clock for f_{CAN} (communication clock) with setting of GCFG.DCS bit = 1, the frequency must be equal or less than f_{CLK}.
- Interrupt processing of RS-CANFD lite

  The interrupt flags in RS-CANFD lite (such as the RFIF bit in the RFSTSk register) are not cleared automatically when an interrupt occurs, so all these interrupt flags must be cleared within the interrupt processing. See **18.12.2 Interrupt Processing Flow** for details.

No.4: Correction of typo in Example of Settings for SNOOZE Mode Status Output

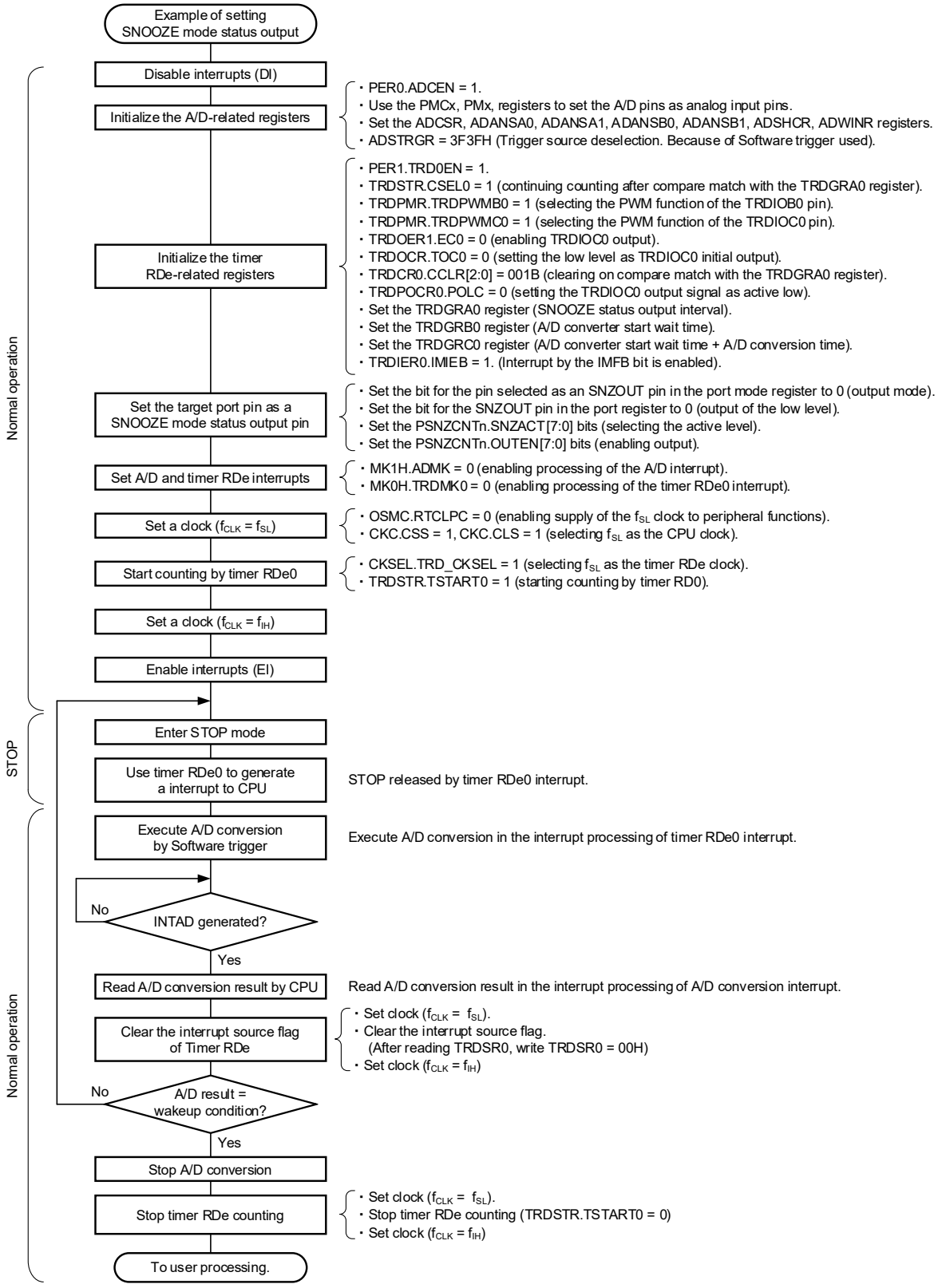
(Incorrect)

Figure 23-10. Example of Settings for SNOOZE Mode Status Output



(Correct) Add clearing process for interrupt source flag.

Figure 23-10. Example of Settings for SNOOZE Mode Status Output



No.5: Add explanation of bit 2 in the CFERRCTLR register

(Incorrect)

Bit 2	Reserved
–	The write value must be 0. The read value is 0.

(Correct)

Bit 2	Reserved
–	The write value must be 0. When read, the value read depends on the operation mode. In normal operation mode: 0 is read. In on-chip debug mode: An undefined value is read.

No.6: Add explanation of FLWE register

(Incorrect)

Cautions 1. This register is cleared when a reset signal is generated.
2. When the DCLR bit of the (FSSQ) is set, FLWE is effective as ECC data for Flash memory.

(Correct)

Cautions 1. This register is cleared when a reset signal is generated.
2. When the DCLR bit (in the FSSQ register) is 0, the FLWE register stores ECC data from the ECC area sequencer in the flash memory. When the DCLR bit is 1 (ECC area sequencer stopped), the FLWE register can be used as a diagnostic self-test.

No.7: Correction of typos in flash memory

- **31.2, Correction of typo in the caution for user option byte (000C0H/040C0H).**

(Incorrect)

Caution The watchdog timer continues its operation during EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window open period taking this delay into consideration.

(Correct)

Caution The watchdog timer continues to operate even when the flash memory is being rewritten. Set the overflow time and window open period taking into consideration the delay in clearing the watchdog timer counter.

- **32, Correction of typo in the data flash memory description.**

(Incorrect)

The data flash memory can be rewritten to by using the **data flash code** during user program execution (background operation). For details about accessing or writing to the data flash memory, see **32.9 Data Flash Memory**.

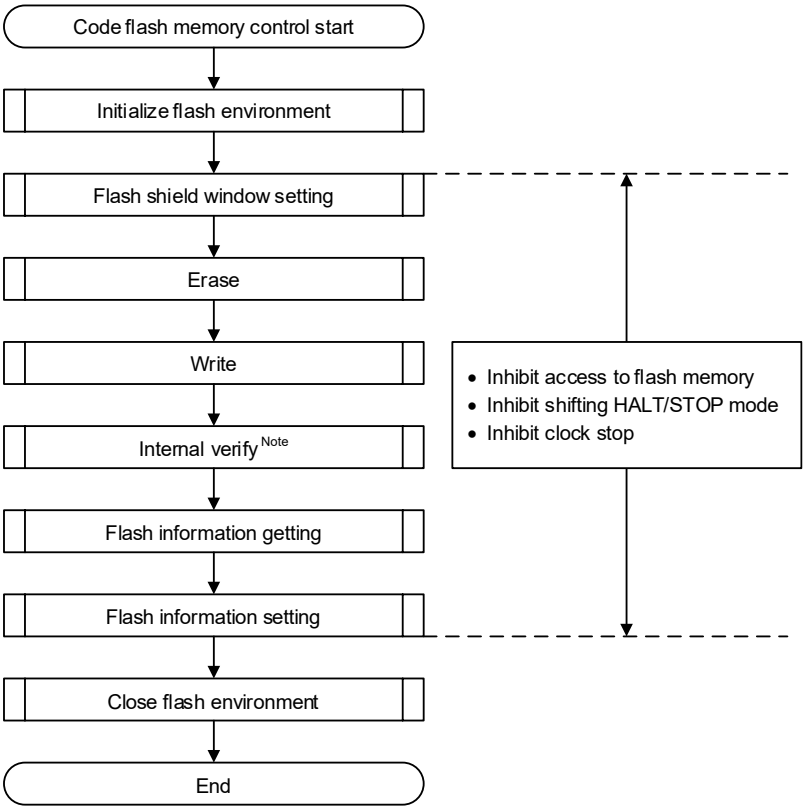
(Correct)

The data flash memory can be written using a dedicated flash memory programmer or using serial programming by an external device. It can also be rewritten during user program execution (background operation). For details about accessing or writing to the data flash memory, see "**32.7 Self-Programming**" and "**32.9 Data Flash Memory**".

- **32.7.1, Correction of typo in Figure 32-8.**

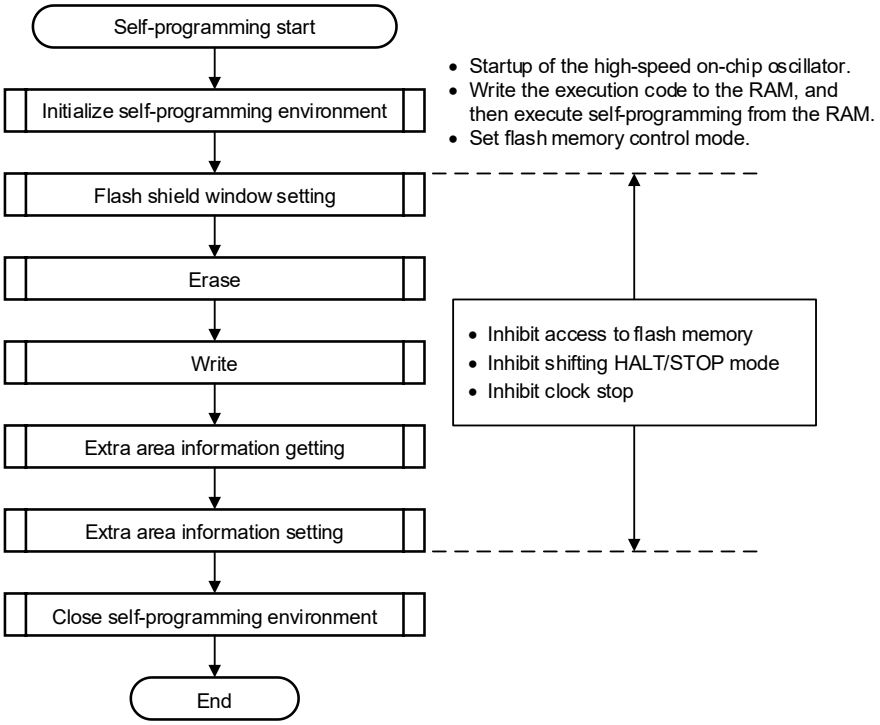
(Incorrect)

Figure 32-8. Flow of Self-Programming (Rewriting Flash Memory)



(Correct)

Figure 32-8. Flow of Self-Programming (Rewriting Flash Memory)



- **32.7.7, Correct typo in boot swap function.**

(Incorrect)

When the program has been correctly written to boot cluster 1, **swap this boot cluster 1 and boot cluster 0 by using the boot firmware of the RL78/F23 or RL78/F24, so that boot cluster 1 is used as a boot area**. After that, erase or write the original area, boot cluster 0.

(Correct)

When the program has been correctly written to boot cluster 1, the boot area changes from boot cluster 0 to boot cluster 1 through self-programming. After that, erase or write the original area, boot cluster 0.

- **32.7.10, Correction of errors in the notes on self-programming.**

(Incorrect)

(1) Rewriting the code flash memory or extra area

To rewrite the code flash memory or extra area, **place the code or values in the RAM**.

(6) Notes when using on-chip debugger

When using on-chip debugger, do not set breakpoints on the on-chip debugger during the process from the start of the flash memory sequencer operation to the completion of the operation.

(Correct)

(1) Rewriting the code flash memory or extra area

To rewrite the code flash memory or extra area, place the re-programming code in the RAM.

(6) Notes when using on-chip debugger

When using on-chip debugger to break during flash self-programming. Set “Using the flash self programming” to [Yes] in the debug settings of the integrated development environment such as CS+ or e2studio. For details, refer to the User’s Manual of the development environment.

- **32.8, Correct typo in security settings.**

(Incorrect)

32.8 Security Settings

The RL78/F23 and RL78/F24 support security functions that prohibit rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the **Security Set command**.

(Correct)

32.8 Security Settings

The RL78/F23 and RL78/F24 support security functions that prohibit rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed by serial programming or self-programming.

● **32.9.1, Correction to typo in Overview of the Data Flash Memory.**

(Incorrect)

- The data flash memory can be rewritten by a user program using the data flash code.

Cautions 1. The data flash memory is stopped after a reset is released. To use the data flash memory, the data flash control register (DFLCTL) must be set up.

2. The high-speed on-chip oscillator must be running while rewriting the data flash memory. If this oscillator is stopped, start the high-speed on-chip oscillator clock by setting HIOSTOP to 0.
Then, after 30 µs has elapsed, execute the data flash code.

(Correct)

- The data flash memory can be rewritten by using self-programming with a user program.

Cautions 1. The data flash memory is stopped after a reset is released. To use the data flash memory, the data flash control register (DFLCTL) must be set up.

2. The high-speed on-chip oscillator must be running while rewriting the data flash memory. If this oscillator is stopped, start the high-speed on-chip oscillator clock by setting HIOSTOP to 0.
Then, wait at least 30 µs in software for oscillation accuracy stabilization time before executing the data flash self-programming.

● **32.9.2, Correction of typos in the data flash memory access procedure.**

(Incorrect)

32.9.2 Procedure for Accessing Data Flash Memory

The data flash memory is stopped after a reset is released. To access the data flash memory, initial settings must be specified as described below.

After the initial settings are specified, the data flash memory can be read by CPU instructions and can be read or rewritten by using a data flash programming code.

Caution 3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash programming should be executed after 30 µs have elapsed.

(Correct)

32.9.2 Procedure for Accessing Data Flash Memory

The data flash memory is stopped after a reset is released. To access the data flash memory, initial settings must be specified as described below.

After the initial settings are specified, the data flash memory can be read by CPU instructions and it can be read/written by a user program using self-programming.

Caution 3. The high-speed on-chip oscillator must be running while rewriting the data flash memory. If this oscillator is stopped, start the high-speed on-chip oscillator clock by setting HIOSTOP to 0. Then, wait at least 30 µs in software for oscillation accuracy stabilization time before executing the data flash self-programming.

No.8: Add Flash Memory description

- **32, Add cautions when erasing Code Flash Memory.**

(Incorrect)

The methods for programming the flash memory are shown below.

The code flash memory can be rewritten to through serial programming using a flash memory programmer or external device (UART communication) or through self-programming.

(Correct)

The methods for programming the flash memory are shown below.

The code flash memory can be rewritten to through serial programming using a flash memory programmer or external device (UART communication) or through self-programming.

When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs. Write data to all unused areas in the code flash memory. Also, it is recommended to writing data “FFH” to unused areas. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.

- **32.7, Correction to Caution 5 in the Self-Programming explanation.**

(Incorrect)

Caution 5. When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs. It is recommended to write data ("FFH") in the area where code flash memory area is not written. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.

(Correct)

Caution 5. Interrupts should be disabled (DI state) while self-programming is in progress. When suspending self-programming and use interrupts, before enabling interrupts (EI state), move to non-programable mode and mask the code flash memory ECC interrupt. When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs. Write data to all unused areas in the code flash memory. Also, it is recommended to writing data “FFH” to unused areas. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.

- **32.7.10, Correction of errors in the notes on self-programming.**

(Incorrect)

(2) Erasing the code flash memory area

When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs.

It is recommended to write data (“FFH”) in the area where code flash memory area is not written. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.

(Correct)

(2) Erasing the code flash memory area

Interrupts should be disabled (DI state) while self-programming is in progress. When suspending self-programming and use interrupts, before enabling interrupts (EI state), move to non-programable mode and mask the code flash memory ECC interrupt.

When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs. Write data to all unused areas in the code flash memory. Also, it is recommended to writing data “FFH” to unused areas. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.

No.9: Correction of typos in electrical characteristics

- **36.5.1, 37.5.1, 38.5.1, Correction of typos in the Serial Array Unit (2), (3), and (9).**

(Incorrect)

(2), (3): Note 5. $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.
(9): Note 3. $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

(Correct)

(2), (3): Note 5. $t_{KCY1} \geq 4/f_{MCK}$ must also be satisfied.
(9): Note 3. $t_{KCY1} \geq 4/f_{MCK}$ must also be satisfied.

- **36.5.2, 37.5.2, 38.5.2, Corrected typos in the description of conditions in the table.**

(Incorrect)

Parameter	Symbol	Conditions	Normal Mode		Fast Mode		Fast Mode Plus		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: 10 MHz ≤ f_{CLK}					0	1000	kHz
		Fast mode: 3.5 MHz ≤ f_{CLK}			0	400			kHz
		Normal mode: 1 MHz ≤ f_{CLK}	0	100					kHz

(Correct)

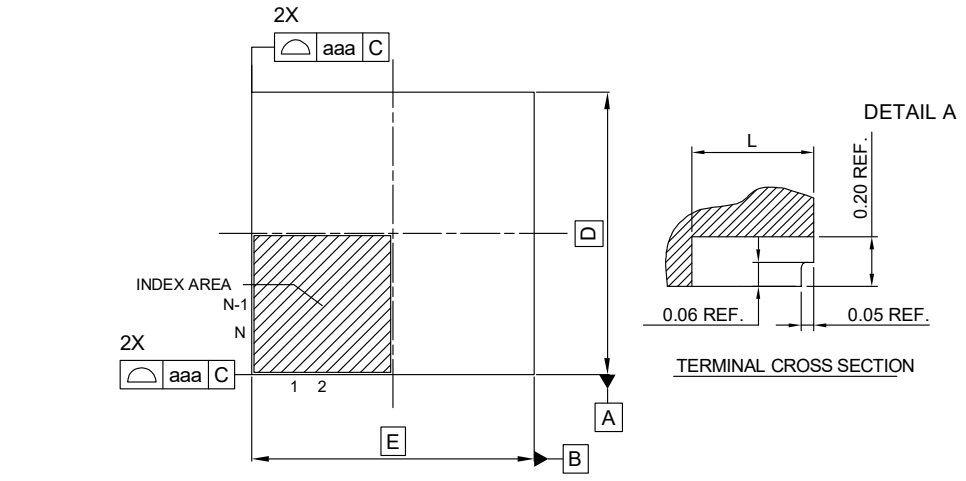
Parameter	Symbol	Conditions	Normal Mode		Fast Mode		Fast Mode Plus		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: 10 MHz ≤ f_{MCK}					0	1000	kHz
		Fast mode: 3.5 MHz ≤ f_{MCK}			0	400			kHz
		Normal mode: 1 MHz ≤ f_{MCK}	0	100					kHz

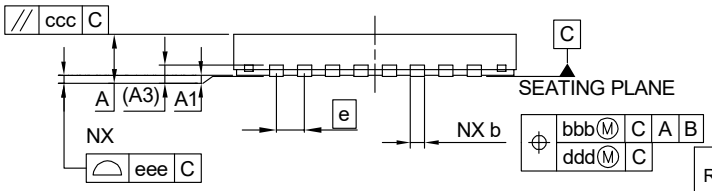
No.10: Add description of package drawing (32-pin product)

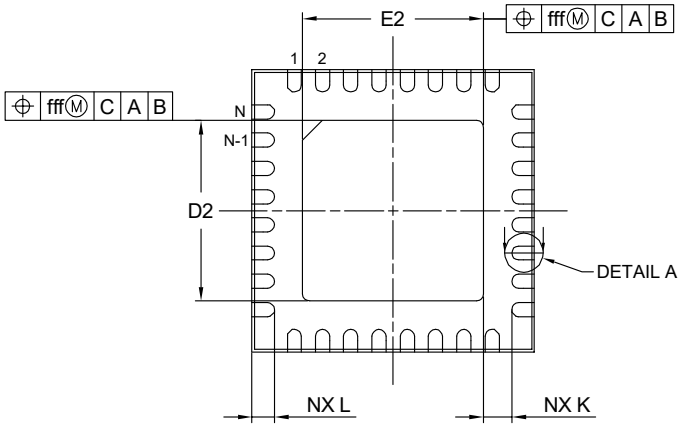
(Incorrect)

39.1 32-pin Products

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KF-B	0.06







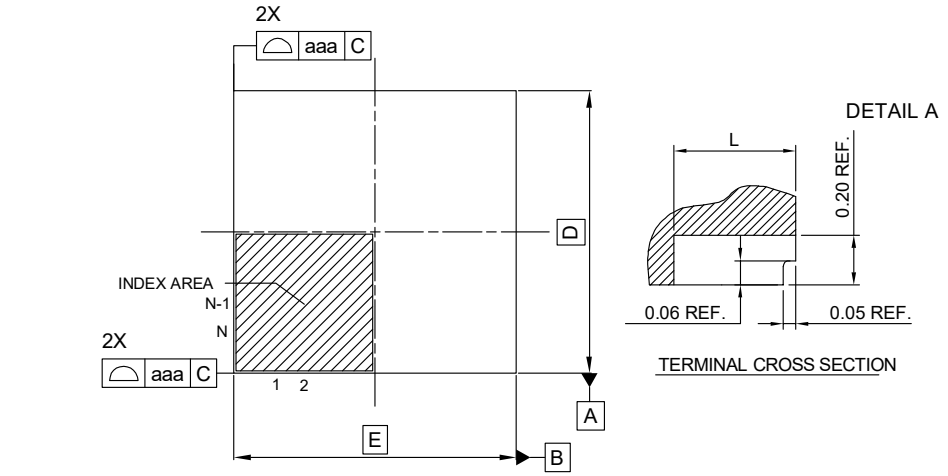
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	0.80
A1	0.00	–	0.05
A3		0.20 REF.	
b	0.18	0.25	0.30
D	–	5.00	–
E	–	5.00	–
e	–	0.50	–
N		32	
L	0.35	0.40	0.45
K	0.20	–	–
D2	3.15	3.20	3.25
E2	3.15	3.20	3.25
aaa	–	–	0.15
bbb	–	–	0.10
ccc	–	–	0.10
ddd	–	–	0.05
eee	–	–	0.08
fff	–	–	0.10

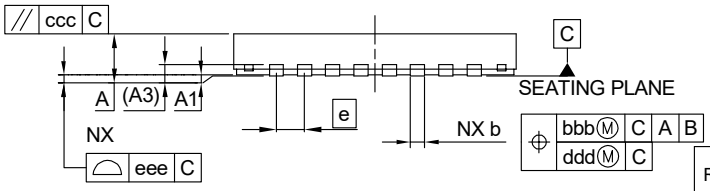
(Correct)

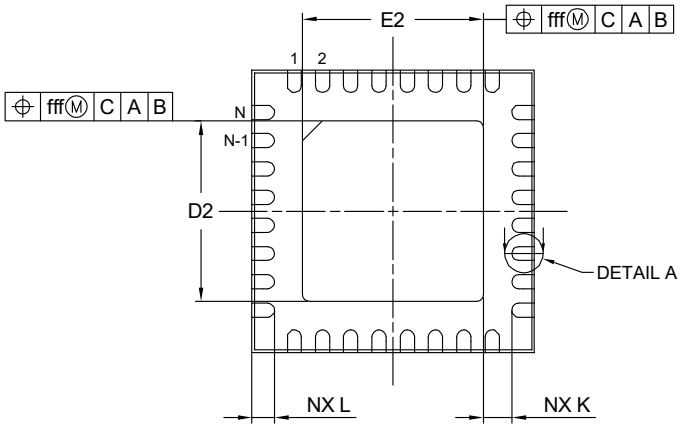
39.1 32-pin Products

There are two types of package drawings. You can check which package drawing used by product marking.
Please refer to the product packaging information on Renesas website for more information.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KF-B	0.06

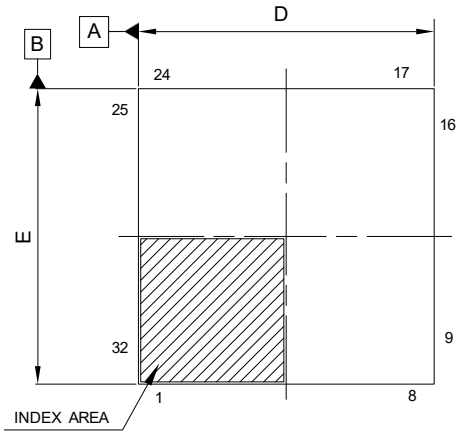


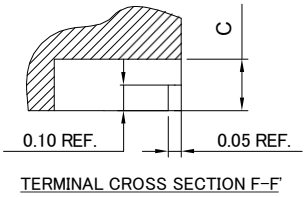


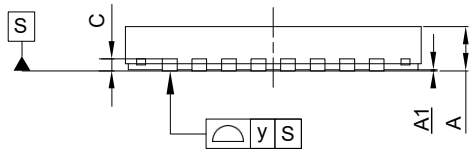


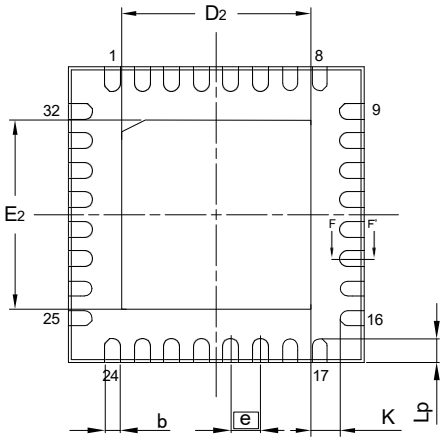
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A1	0.00	—	0.05
A3		0.20 REF.	
b	0.18	0.25	0.30
D	—	5.00	—
E	—	5.00	—
e	—	0.50	—
N		32	
L	0.35	0.40	0.45
K	0.20	—	—
D2	3.15	3.20	3.25
E2	3.15	3.20	3.25
aaa	—	—	0.15
bbb	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08
fff	—	—	0.10

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN32-5x5-0.50	PWQN0032KH-A	0.06









Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	4.85	5.00	5.15
E	4.85	5.00	5.15
A	—	—	0.80
A_1	0.00	—	0.05
b	0.18	0.25	0.30
e		0.50 BSC	
Lp	0.35	0.40	0.45
y	—	—	0.08
c	—	0.20	—
K	0.20	—	—
D_2	—	3.20	—
E_2	—	3.20	—