Date: Sep. 04, 2013

RENESAS TECHNICAL UPDATE

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan
Renesas Electronics Corporation

Product Category	MPU/MCU		Document No.	TN-RX*-A073A/E	Rev.	1.00
Title	Corrections of 'RX21A Group User's Manual: Hardware Rev.1.00'		Information Category	Technical Notification		
Applicable Product	RX21A Group	Lot No. All	Reference Document	RX21A Group User's Manual: Hardware Rev.1.00 (R01UH0251EJ0100)		

This document describes the corrections in RX21A Group User's Manual: Hardware Rev.1.00. Changes are underlined in the list below.

- Page 41

Add a note to 'Table 1.1 Outline of Specifications (3 / 4)'.

Operating temperature D version: –40 to +85°C, G version: –40 to +105°C ***1**

Note 1. Contact Renesas Electronics when G version is to be used.

- Page 148

Correct a description in '8.2.6 Voltage Detection Level Select Register (LVDLVLR)'.

Original text

The contents of the LVDLVLR register can only be changed if the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits (voltage detection n circuit disable; n = 1, 2) are both 0.

It should be:

When changing the LVDLVLR register, first set the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits to 0 (voltage detection n circuit disabled) (n = 1, 2).

- Page 170

Correct the description in '9.2.5 Main Clock Oscillator Control Register (MOSCCR)'.

Original text

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited while the main clock oscillator or PLL is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

It should be:

Do not set the MOSTP bit to 1 when one of the following conditions is met.

- **When the main clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 010b)**
- **When the PLL clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 100b)**
- **When the PLL is operating (PLLCR2.PLLEN = 0)**

- Page 176

Insert a notice of the detection time into the description in '9.2.11 Oscillation Stop Detection Control Register (OSTDCR)'.

This bit enables or disables the oscillation stop detection function.

After this function is enabled, tdr (see Table 42.73, Oscillation Stop Detection Circuit Characteristics) is required before stable operation starts.

When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is cleared to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 to the LOCOCR.LCSTP bit (LOCO stopped) is invalid.

- Page 179

Correct the description of HOCOPCNT Bit in '9.2.14 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR)'.

It should be:

Do not change the value of the HOCOPCNT bit in the following cases:

- When the HOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- When the setting of the operating-power control mode selection bits in the operating-power control register (OPCCR.OPCM[2:0]) is for **high-speed operating mode,** middle-speed operating mode 2A, middle-speed operating mode 2B, or low-speed operating mode 1 or 2.

- Page 190

Correct the description in 'Notes on Sub-Clock'.

Original text

(1) Set the RCR3.RTCEN bit to 0.

(2) Read the RCR3.RTCEN bit and confirm that it is 0.

(3) Set the RCR3.RTCDV[2:0] bits.

If the RCR3.RTCDV[2:0] bits are set in this step, they do not need to be reset in section 26.3.2, Clock and count mode Setting Procedure.

(4) Set the SOSCCR.SOSTP bit to 1.

(5) Read the SOSCCR.SOSTP bit and confirm that it is 1.

(6) Set the SOSCWTCR.SSTS[4:0] bits to specify the wait time necessary for sub-clock oscillation.

(7) Set the SOSCCR.SOSTP bit to 0 (sub-clock oscillator is operating).

(8) Wait for the oscillation stabilization wait time of the sub-clock*1 to elapse.

(9) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten to 0, then set the RCR3.RTCEN bit to 1 (sub-clock oscillator is operating).

(10) Read the RCR3.RTCEN bit and confirm that it has been rewritten to 1, then set the SOSCCR.SOSTP bit to 1.

(11) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten to 1, **then set the SOSCWTCR.SSTS[4:0] bits to 00000b.**

(12) Wait for at least five cycles of the sub-clock to elapse.

(13) Set the SOSCCR.SOSTP bit to 0.

(14) Wait for at least two cycles of the sub-clock to elapse.

(15) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten, **then set the RCR3.RTCEN bit to 0.**

It should be:

(1) Wait for the oscillation stabilization wait time of the sub-clock*1 to elapse.

(2) Set the SOSCCR.SOSTP bit to 1.

(3) Read the SOSCCR.SOSTP bit and confirm that it is 1.

(4) Set the RCR3.RTCEN bit to 0.

(5) Read the RCR3.RTCEN bit and confirm that it is 0.

(6) Wait for at least five cycles of the sub-clock to elapse.

(7) Set the RCR3.RTCDV[2:0] bits.

If the RCR3.RTCDV[2:0] bits are set in this step, they do not need to be reset in section 26.3.2, Clock Setting Procedure.

(8) Set the SOSCWTCR.SSTS[4:0] bits to specify the wait time necessary for sub-clock oscillation.

(9) Set the SOSCCR.SOSTP bit to 0 (sub-clock oscillator is operating).

(10) Wait for the oscillation stabilization wait time of the sub-clock*1 to elapse.

(11) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten to 0, then set the RCR3.RTCEN bit to 1 (sub-clock oscillator is operating).

(12) Read the RCR3.RTCEN bit and confirm that it has been rewritten to 1, then set the SOSCCR.SOSTP bit to 1.

(13) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten to 1, **then wait for at least five cycles of the sub-clock to elapse.**

(14) Set the SOSCWTCR.SSTS[4:0] bits to 00000b.

(15) Set the SOSCCR.SOSTP bit to 0.

(16) Wait for at least two cycles of the sub-clock to elapse.

(17) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten.

Page 203

Change wording in ‘Table 11.1 Specifications of Low Power Consumption Functions’.

Original text

Item	Description
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the system clock (ICLK), DEU clock (PCLKA), peripheral module clock (PCLKB), DSAD clock (PCLKC), AD clock (PCLKD), and **flash interface** clock (FCLK).*1

It should be:

Item	Description
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the system clock (ICLK), DEU clock (PCLKA), peripheral module clock (PCLKB), DSAD clock (PCLKC), AD clock (PCLKD), and **FlashIF** clock (FCLK).*1

- Page 165 and 167 to 180

Add a note which refers the write protection bit to the following registers’ description: SCKCR, SCKCR3, PLLCR, PLLCR2, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOCR, HOCOCR2, OSTDCR and OSTDSR.

Note. Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

- Page 89, 126, 178 to 180, 206 to 212, 219 to 228, and 230 to 232

Add a note which refers the write protection bit to the following registers’ description: SYSCR1, SWRR, MOFCR, HOCOPCR, PLLPCR, SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, OPCCR, RSTCKCR, MOSCWTCR, SOSCWTCR, PLLWTCR, HOCOWTCR2, DPSBYCR, DPSIER0, DPSIER2, DPSIFR0, DPSIFR2, DPSIEGR0, DPSIEGR2, and FHSSBYCR.

Note. Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

- Page 143 to 151

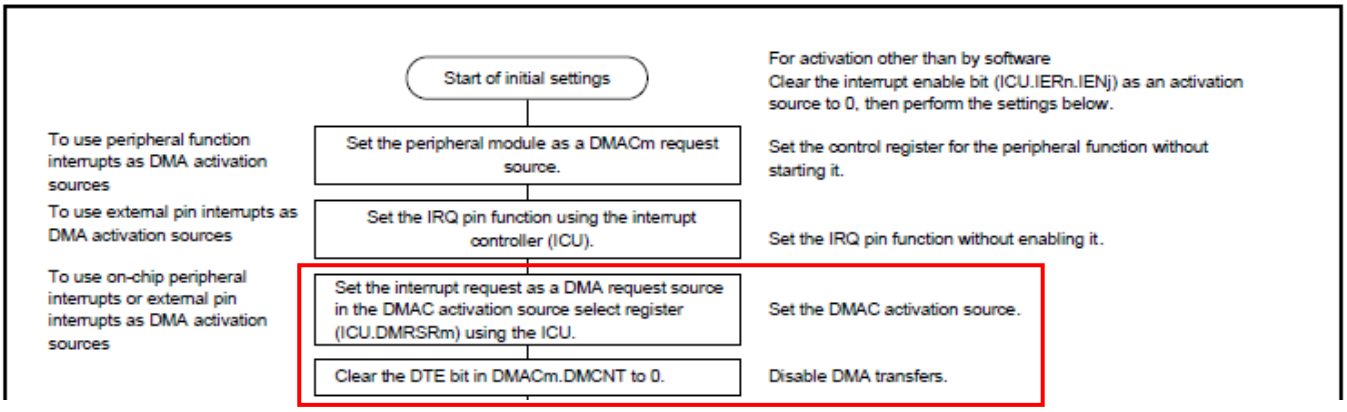
Add a note which refers the write protection bit to the following registers’ description: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVCMPCR, LVDLVLR, LVD1CR0, LVD2CR0.

Note. Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

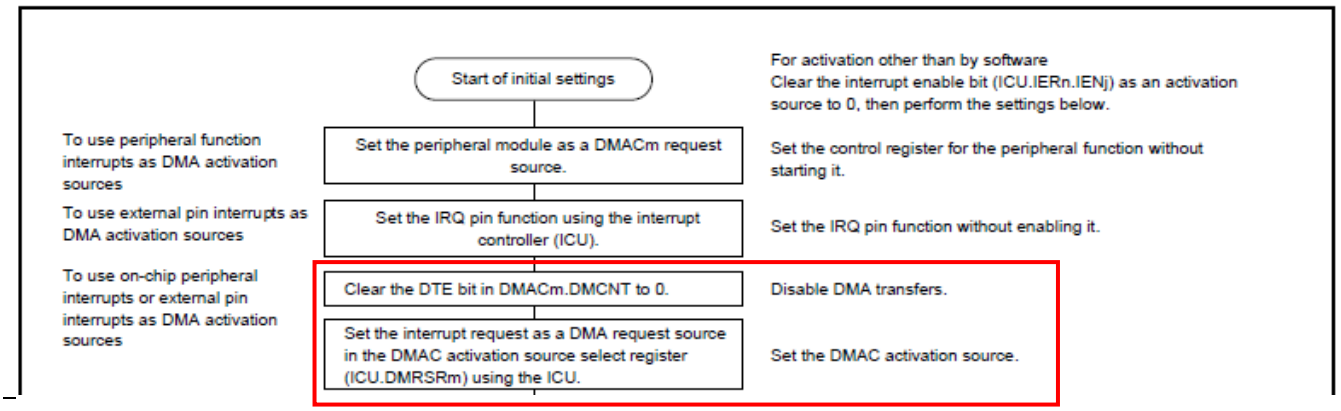
- Page 363

Change the order of steps in ‘Figure 17.12 Register Setting Procedure’.

Original flow



It should be:



- Page 401

Remove unnecessary restriction from ‘18.8 Event Link Function’.

The DTC outputs the event link request after transfer for that single activating source is completed. ~~However, if the transfer destination is the internal peripheral bus, the event link request is issued at the end of writing to the write buffer.~~

- Page 419

Add a note to ‘Table 19.5 Operations of Modules when Event is Input’.

Table 19.5 Operations of Modules when Event is Input

Module	Operations when Event is Input
Clock oscillator	Switches the clock source to the low-speed on-chip oscillator operation.***1**

Note 1. The SCKCR3.CKSEL[2:0] bits are modified to 000b (LOCO) regardless of the value of the protect register (PRCR.PRC0).

- Page 433

Add ‘Figure 20.6 I/O Port Configuration (6)’ next to Figure 20.5.

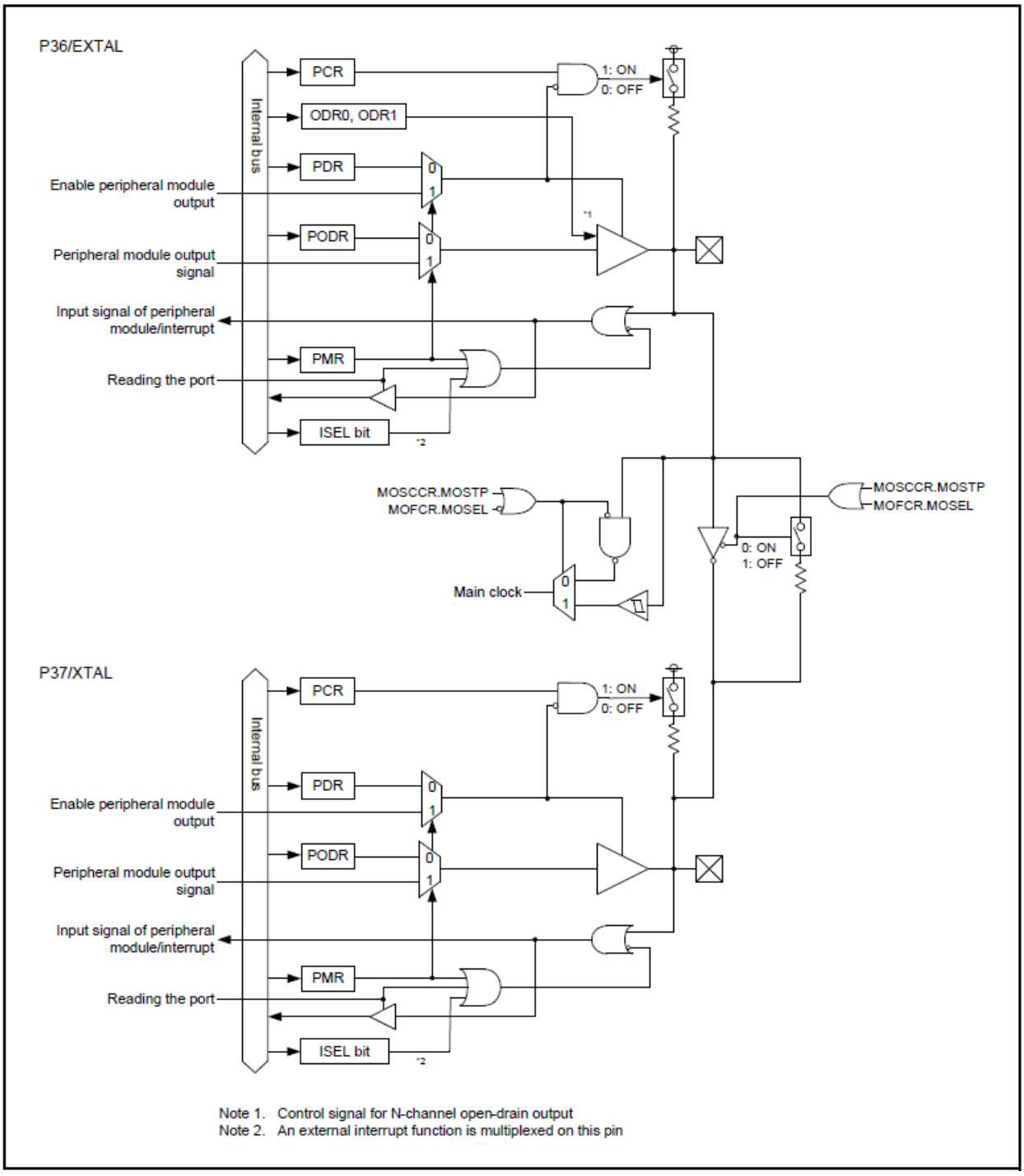


Figure 20.6 I/O Port Configuration (6)

- Page 441

Add a notice of the high-drive output.

When a transition to deep software standby mode is made, all bits are set to normal drive output. They are also set to normal drive output after canceling the mode.

When high-drive output is selected, switching noise increases compared to when normal output is selected. Carefully evaluate the effect of noise on the MCU caused by adjacent pins before selecting high-drive output.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

- Page 443

Modify 'Table 20.3 Treatment of Unused Pins'.

Original text

Pin Name	Description
EXTAL	**Connect this pin to VSS via a pull-down resistor.**
XTAL	**Leave this pin open.**

Modified

Pin Name	Description
P36/EXTAL	**When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P36). When this pin is not used as port P36 either, it is configured in the same way as port 0 to 5.**
P37/XTAL	**When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P37). When this pin is not used as port P37 either, it is configured in the same way as port 0 to 5. When the external clock is input to the EXTAL pin, leave this pin open.**

- Page 792

Correct a description in '28.3.4 Status Flags'.

Original text

In addition, several cycles of the IWDTCLK (at least three) and of the PCLK (at least two) are required to read the reflected value after clearing the flag by writing 0 to the bit.

It should be:

After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

- Page 837

Add a note to ‘Figure 29.7 Sample SCI Initialization Flowchart (Asynchronous Mode)’.

Note: Also refer to ‘29.12.8 Points to Note on Starting Transfer’.

- Page 851 and 875

Add a note to ‘Figure 29.20 Example of SCI Initialization Flowchart (Clock Synchronous Mode)’ and ‘Figure 29.44 Example of the Flow of SCI Initialization (for Simple I2C Mode)’.

Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.

Also refer to ‘29.12.8 Points to Note on Starting Transfer’.

- Page 890

Add the description in '29.12.8 Points to Note on Starting Transfer'.

Original text

At the point where transfer starts when the interrupt status flag in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has actually become 0.
- Set the interrupt status flag in the interrupt controller to 0.

It should be:

At the point where transfer starts when the interrupt status flag in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). **Refer to '14. Interrupt Controller (ICUb) ' for more details of the interrupt status flag.**

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has actually become 0.
- Set the interrupt status flag **(IRn.IR)** in the interrupt controller to 0.

- Page 905

Delete the incorrect description of BBSY flag in '31.2.2 I2C Bus Control Register 2 (ICCR2)'.

Original text

Bit	Symbol	Bit Name	Description	R/W
b7	BBSY	Bus Busy Detection Flag	0: The I2C bus is released (bus free state). 1: The I2C bus is occupied (bus busy state **or in the bus free state**).	R

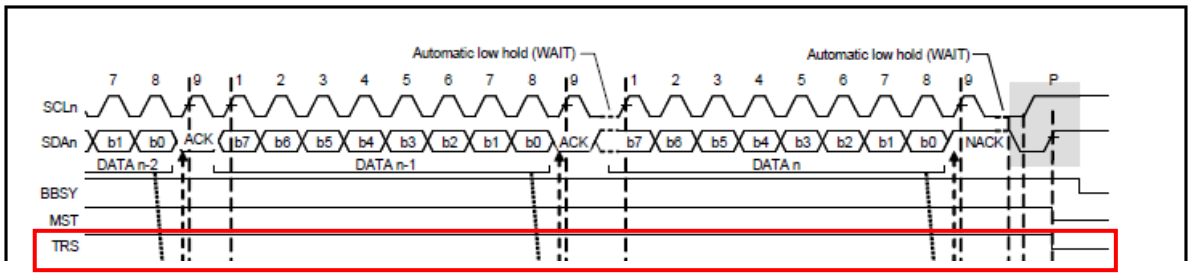
It should be:

Bit	Symbol	Bit Name	Description	R/W
b7	BBSY	Bus Busy Detection Flag	0: The I2C bus is released (bus free state). 1: The I2C bus is occupied (bus busy state).	R

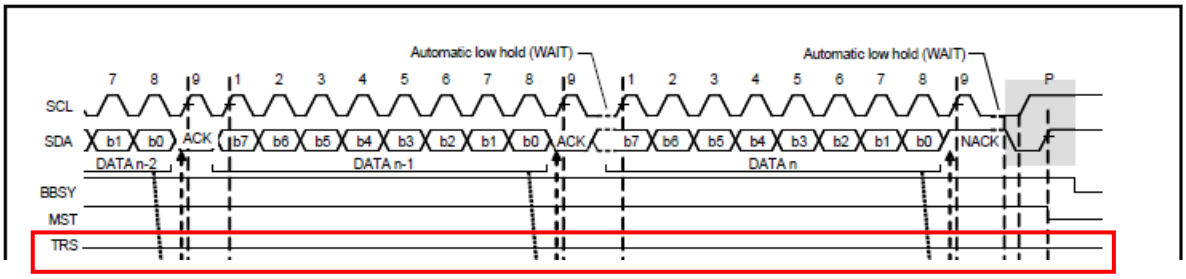
- Page 943

Correct the waveform of TRS in ‘Figure 31.13 Master Receive Operation Timing (3) (when RDRFS = 0)’.

Original



Correction



- Page 987

Correct the description of OVRF Flag (Overrun Error Flag) in ‘32.2.4 RSPI Status Register (SPSR)’.

Original text

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer **holds data that has not yet been read**

It should be:

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer **is full.**

- Page 990

Add a description in '32.2.5 RSPI Data Register (SPDR)'.

Original text

(a) Writing

The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.

It should be:

(a) Writing

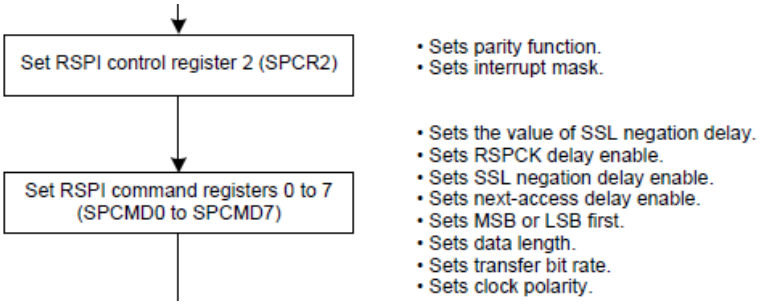
SPDR can be written to write a value to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading the SPDR register.

The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data is written to SPDR.

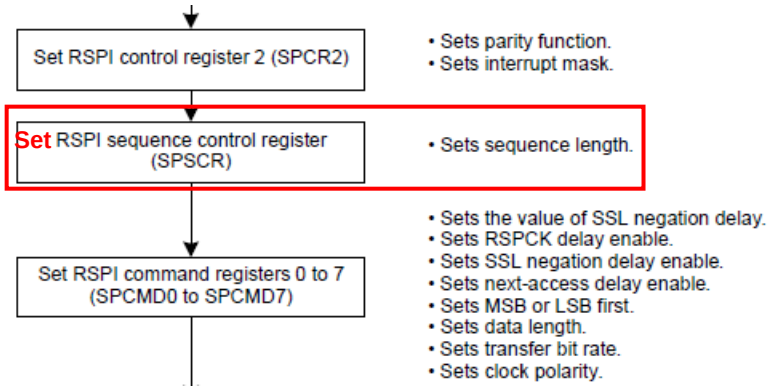
- Page 1039 and 1050

Add a step in 'Figure 32.34 Example of Initialization Flowchart in Master Mode (SPI Operation)' and 'Figure 32.45 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)'.

Original flow



Corrected flow



- **Page 1332**

Correct the inconsistent term in '44.6 A/D Conversion Characteristics'

Original text

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal AD conversion characteristics

and the width of the actually output code.

It should be:

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics

and the width of the actually output code.

- **Page 1111**

Correct the description in '35.3.2.2 Channel Selection and Self-Diagnosis'.

Original text

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0 (×0,

×1/2, or ×1) supplied to the A/D converter, and then A/D conversion is performed once on the analog input of the selected channels as below.

It should be:

When channels and self-diagnosis are selected, A/D conversion is first performed for **the self-diagnosis voltage (VREFH0 × 0, VREFH0 × 1/2, or VREFH0 × 1) generated from the reference power supply voltage (VREFH0)** supplied to the **10-bit** A/D converter, and then A/D conversion is performed once on the analog input of the selected channels as below.

- Page 1115

Correct the description in '35.3.3.2 Channel Selection and Self-Diagnosis'.

Original text

When channels and self-diagnosis are selected, A/D conversion is first performed for **the reference voltage VREFH0 (×0, ×1/2, or ×1)** supplied to the 10-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below.

It should be:

When channels and self-diagnosis are selected, A/D conversion is first performed for **the self-diagnosis voltage (VREFH0 × 0, VREFH0 × 1/2, or VREFH0 × 1) generated from the reference power supply voltage (VREFH0)** supplied

to the 10-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, for which the sequence is repeated as below.

- Page 1170

Add a note to ‘40.3.1 Data Comparison Mode’.

Original text

5. If a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1. On setting of the DOCR.DOPCIE bit to 1, a data operation circuit interrupt is also generated.

It should be:

5. If a value written to DODIR does not match that in DODSR***1**, the DOCR.DOPCF flag is set to 1. On setting of the DOCR.DOPCIE bit to 1, a data operation circuit interrupt is also generated.

Note 1. When DOCR.DCSEL = 1

- Page 1208

Correct the description in '42.6.4.2 Programming and Erasure Procedures'.

Original text

(4) Programming

The programming command is used to write data to the ROM.

In the first cycle of the programming command, the value E8h is written as a byte to the address range for programming and erasure of the ROM. In the second cycle, the values, 01h (for 2-byte programming), 04h (for 8-byte programming), or 40h (for 128-byte programming) are written to the same address range. In the third cycle, write the actual data to be programmed, as a word unit, to the start address of the target area for programming. **For this start address, always use an address that is aligned on a 128-byte boundary. In the fourth to the 66th (in the case of 128-byte programming) cycles, write the data for programming in 63 word-unit rounds to the address range for programming and erasure of the ROM.**

It should be:

(4) Programming

The programming command is used to write data to the ROM.

In the first cycle of the programming command, the value E8h is written as a byte to the address range for programming and erasure of the ROM. In the second cycle, the values, 01h (for 2-byte programming), 04h (for 8-byte programming), or 40h (for 128-byte programming) are written to the same address range. In the third cycle, write the actual data to be programmed, as a word unit, to the start address of the target area for programming.

In the case of 128-byte programming, 128 bytes (64 words) of data are written to the ROM in the third to the 66th cycles, divided into 64 rounds. The start address of 128 bytes for programming is specified in the third cycle. The address range specified at this time must be an integral multiple of 128. The address range specified in the fourth to the 66th cycles does not need to be the address range for actual programming.

In the case of 8-byte programming, 8 bytes (4 words) of data are written to the ROM in the third to the sixth cycles, divided into four rounds. The start address of 8 bytes for programming is specified in the third cycle. The address range specified at this time must be an integral multiple of 8. The address range specified in the fourth to the sixth cycles does not need to be the address range for actual programming.

In the case of 2-byte programming, the address range and data for programming are specified in the third cycle. The address range must be an even number.

- Page 1219

Correct inconsistent terms in '42.7.2 Suspension during Programming/Erasure (Programming/Erasure Priority Mode)'

Original text

Figure 42.20 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (FCPSR.ESUSPMD bit is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.

It should be:

Figure 42.20 shows the suspend operation of erasure when the erasure suspend mode is set to the **programming/**erasure priority mode (FCPSR.ESUSPMD bit is 1). The control method of erasure pulses in **programming/**erasure priority mode is the same as that of programming pulses for the programming suspend processing.

- Page 1219

Correct an inconsistent term in the caption of Figure 42.20.

Original text

Figure 42.20 Suspension during Erasure (Erasure Priority Mode)

It should be:

Figure 42.20 Suspension during Erasure (**Programming/**ErasurePriority Mode)

- Page 1265

Correct the description below the 'Table 43.4 FCU Command Formats (Commands Dedicated to the E2 DataFlash Memory)'.

Original text

BA: **Block address where reading, programming, or erasure of the E2 DataFlash is enabled or prohibited.**
Any address in a block where reading, programming, or erasure is enabled or prohibited.

It should be:

BA: **Block start address within the E2 DataFlash area (2-Kbyte block)**

- Page 1267

Correct the description in '43.6.4 FCU Command Usage'.

Original text

(2) Programming

To program the E2 DataFlash, use one of the programming commands.

Use byte access to write E8h to an address within the E2 DataFlash area in the first cycle of the programming command, and the number of words (N)*1 to be programmed in the second cycle. Access the peripheral bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the first word of data for programming to the address where the target area for programming starts. **This address must be on a 2-byte boundary for 2-byte programming. After writing words to addresses in the E2 DataFlash area N times, write byte D0h to an address within the E2 DataFlash area in cycle N + 3; the FCU will then start actual programming of the E2 DataFlash. Read the FRDY bit in FSTATR0 to confirm the completion of E2 DataFlash programming.**

It should be:

(2) Programming

To program the E2 DataFlash, use one of the programming commands.

Use byte access to write E8h to an address within the E2 DataFlash area in the first cycle of the programming command, and the number of words (N)*1 to be programmed in the second cycle. Access the peripheral bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the first word of data for programming to the address where the target area for programming starts.

In the case of 8-byte programming, 8 bytes (4 words) of data are written to the E2 DataFlash in the third to the sixth cycles, divided into four rounds. The start address of 8 bytes for programming is specified in the third cycle. The address range specified at this time must be an integral multiple of 8. The address range specified in the fourth to the sixth cycles does not need to be the address range for actual programming.

In the case of 2-byte programming, the address range and data for programming are specified in the third cycle. The address range must be an even number.

After writing words to addresses in the E2 DataFlash area N times, write byte D0h to an address within the E2 DataFlash area in cycle N + 3; the FCU will then start actual programming of the E2 DataFlash. Read the FRDY bit in FSTATR0 to confirm the completion of E2 DataFlash programming.

- Page 1336

Add missing brackets in the notes of 'Table 44.42 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (1)'

Correction

Note 1. When the CPU is in a mode other than software standby and deep software standby modes, when the CPU transits to software standby mode with the FHSSBYCR.SOFTCUT[2] bit set to 0, or when the CPU transits to deep software standby mode with the DPSBYCR.DEEPCUT1 bit set to 0.

Note 2. When the CPU transits to software standby mode with the FHSSBYCR.SOFTCUT[2] bit set to 1 or when the CPU transits to deep software standby mode with the DPSBYCR.DEEPCUT1 bit set to 1.

- Page 1346

Add a description to the footnotes of 'Table 1.1 Port States in Each Processing State' in 'Appendix 1. Port States in Each Processing Mode'.

Correction

Keep: Pin states are retained during periods in software standby **mode (pulling up and open-drain settings are also retained).**