SUD-FT-04-0105 (1/11)
April 19, 2004
Seiji Iigusa, Senior System Integrator Car Electronics Group 1st Solutions Division Solutions Operations Unit NEC Electronics Corporation

CUSTOMER NOTIFICATION

CP(K), O

IE-703239-G1-EM1
(Control Code: A, B, C)

Operating Precautions

Be sure to read this document before using the product.

Notes on Using IE-703239-G1-EM1

This document describes the restrictions applicable only to the emulator and restrictions that have been corrected in the emulator.

Refer to the user's manual of the emulator for cautions on using the emulator (document number: SUD-FT-03-0102).

1. Product Version

Control Code	Bord Version	Peripheral EVA Chip
A	V1.00	μPD70F3239GM ES1.0
B	V1.10	μPD70F3239GM ES1.1
C	V1.11	μPD70F3239GM ES2.0

2. Product History

No.	Bugs and Changes/Additions to Specifications		IE-703239-G1-EM1		
		Board Version	1.00	1.10	1.11
		Control Code[Note]	A	B	C
1	Operation at EV$_{DD}$ ≥ 4.0 V		×	√	√
2	Operation at BV$_{DD}$ ≤ 5.5 V		×	√	√
3	RC oscillation of the subclock not supported		×	√	√
4	Bug in port mode control registers		×	√	√
5	Change of TMP0 pin function specification		×	√	√
6	Bug in subclock code execution by flash memory macro		×	×	√
7	Improvement of conversion accuracy of AD converter		×	×	√
8	Bug in CAN transmission/reception		×	×	√
9	Restriction when reset by watchdog timer or external reset occurs		×	×	√
10	Bug in watchdog timer during break		×	×	√
11	Bug in 16-bit timer M during break		×	×	√
12	Bug in 16-bit timer M compare register		×	×	√
13	Error in voltage level detected by low-voltage detector (LVI)		×	×	√
14	Change of specification for programmable clock mode register (PCLM)		×	×	√
15	Bug in accessing UAnRX register during break		Permanent restriction		
16	Bug in accessing CBnRX register during break				
17	Bug in accessing CnRGPT register during break				
18	Bug in accessing CnTGPT register during break				
19	Bug in accessing CnGNCTRL register during break				
20	Restriction on TMQn/TMPn and external event counter				

×: Applicable, √: Not applicable or already corrected

Note The "control code" is the second digit from the left in the 10-digit serial number in the warranty supplied with the product you purchased (if it has not been upgraded). If the product has been upgraded, a label indicating the new version is attached to the product and the x in V-UP LEVEL x on this label indicates the control code.

3. Details of Bugs and Added Specifications

No.1 Operation at $EV_{DD} \geq 4.0$ V

[Description]

When the debugger is connected to the emulator, the regulation of EV_{DD} on the IE-703239-G1-EM1 is restricted to $EV_{DD} \geq 4.0$ V for operating frequency lower than 20 MHz. Therefore operating frequencies lower than 20 MHz should not be used when EV_{DD} on the target circuit is < 4.0 V.

[Workaround]

When the debugger is connected to the emulator, use $EV_{DD} \geq 4.0$ V or use an operating frequency of 20 MHz.

This bug has been corrected in control code B.

No.2 Operation at $BV_{DD} \leq 5.5$ V

[Description]

The internal level shifters may be damaged if BV_{DD} becomes higher than 5.5 V.

[Workaround]

Operate the IE-703239-G1-EM1 at $BV_{DD} \leq 5.5$ V.

This bug has been corrected in control code B.

No.3 Use of RC resonator as subclock not supported

[Description]

Use of an RC resonator as the subclock is not supported in the IE-703239-G1-EM1.

[Workaround]

There is no workaround.

The use of an RC resonator is now supported in control code B.

No.4 Bug in port mode control registers

[Description]

All the peripheral I/O register values become undefined if any of the following bits of the port mode control registers is set to 1.

Bits 2 and 3 of the PMC0 register

Bits 0 to 5 of the PMC3 register

[Workaround]

Do not set the above bits of the port mode control registers to 1.

This bug has been corrected in control code B.

No.5 Change of TMP0 pin function specification

[Description]

TOP01 has now been assigned to the port pin P32.

Products	Pin Number	Before Change	After Change
V850ES/FJ2	27	P32/ASCKA0/TIP00/TOP00	P32/ASCKA0/TIP00/TOP00/TOP01
V850ES/FG2	27		
V850ES/FF2	24		
V850ES/FE2	24		

Pin assignment of port mode control register (PMC3) after change:

	15	14	13	12	11	10	9	8
PMC3	0	0	0	0	0	0	PMC39	PMC38
	7	6	5	4	3	2	1	0
	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC32	P32 Pin Operation Mode Selection
0	I/O port
1	ASCKA0/TIP00/TOP00/TOP01

Pin assignment port function control register (PFC3) and port function control expansion register (PFCEL3) after change:

	7	6	5	4	3	2	1	0
PFC3L	0	0	PFC35	PFC34	PFC33	PFC32	0	0

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	PFCE34	PFCE33	PFCE32	0	0

PFCE32	PFC32	P32 Pin Control Mode Specification
0	0	ASCKA0 input
0	1	TOP01 output
1	0	TIP00 input
1	1	TOP00 output

This item is implemented in control code B.

No.6 Bug in subclock code execution by flash memory macro

[Description]

When the main clock is stopped and the CPU is operating on the subclock, the CPU may illegally recognize the first 4 bytes as nop instructions when it shifts execution to another flash macro.

However, this bug does not apply in the following cases.

(a) When execution returns from another macro to the macro in which the main clock was stopped.

(b) When execution shifts to a macro in which this bug has already been generated.

[Workaround]

This bug has been corrected in control code C.

No.7 Improvement of conversion accuracy of A/D converter

[Description]

The A/D converter has an offset of approx. 30 mV.

This item is implemented in control code C.

No.8 Bug in CAN transmission/reception

[Description]

(a) Illegal message reception

The receive message may not be stored normally when a transmit request is set (by setting the TRQx flag) by setting an extension ID during reception (data field or later).

As a result, extension IDs (ID15 to ID0) of the receive message buffer are erroneously recognized upon comparison between the receive message buffer ID setting and the receive message ID. If they match, the receive data is illegally written to the next buffer to the matched receive message buffer (MSG#n) (MSG#(n+1)). However, MSG#0 will never be written illegally.

(b) Illegal message transmission

The operations shown below may occur when a conflict between an internal operation during transmission/reception or upon completion of transmission/reception and a write to the transmit request flag (TRQx) occurs.

- If the transmit request flag for message buffer 0 (MSG#0) is written, transmission is not performed or data in MSG#1 is transmitted illegally.
- If multiple transmit requests have been set by an extension ID, the priority judgment of the subsequent messages becomes illegal.

[Workaround]

Implement the following software settings.

(a) Use the standard ID setting (not the extension ID).

(b) Set message buffer 0 (MSG#0) as the receive buffer.

* In automatic block transmission mode, however, MSG#0 can be used as the transmit buffer. This bug has been corrected in control code B or later.

No.9 Restriction when reset by watchdog timer or external reset occurs

[Description]

Access to the I/O register becomes illegal when a reset by the watchdog timer or an external reset occurs.

A correct value cannot be read during read, and cannot be written during write.

[Workaround]

The normal status is restored by pressing the CPU reset button on the debugger.

This restriction has been corrected in control code C.

No.10 Bug in watchdog timer during break

[Description]

When both the following conditions (a) and (b) are satisfied and a break occurs, the watchdog timer does not stop and a reset or a non-maskable interrupt occurs. If a reset occurs, the debugger hangs up.

(a) The main clock or subclock is selected as the source clock for the watchdog timer

(b) The Ring clock is stopped (RSTOP flag =1)

[Workaround]

Implement (a) or (b) below.

(a) Use the Ring clock as the source clock.

(b) Do not stop Ring clock oscillation.

This bug has been corrected in control code C.

No.11 Bug in 16-bit timer M during break

[Description]

When a break occurs while the following both (a) and (b) are satisfied, timer M does not stop even if the Peripheral Break function has been set to “Break”.

(a) INTWT, Ring clock (f_R/8), or subclock is selected as the source clock of timer M.

(b) The main clock is stopped by setting the MCK flag.

* The Peripheral Break function is not supported by the debugger ID850 V2.51.

[Workaround]

Implement (a) or (b) below to stop timer M during a break using the Peripheral Break function.

(a) Use the main clock (fxx, fxx/2, fxx/4, fxx/64, or fxx/512) as the source clock.

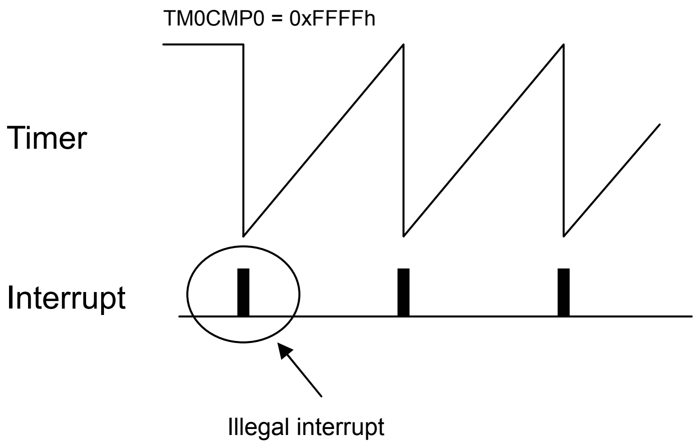
(b) Do not stop main clock oscillation.

This bug has been corrected in control code C.

No.12 Bug in 16-bit timer M compare register

[Description]

An illegal interrupt occurs after timer M activation when the compare register TM0CMP0 is set to FFFFH. See the figure below for details.



[Workaround]

Do not set TM0CMP0 to FFFFH.

This bug has been corrected in control code C.

No.13 Error in voltage level detected by low-voltage detector (LVI)

[Description]

The voltage level that can be detected by the low-voltage detector (LVI) is higher than that prescribed in the specification. The specification defines 4.2 ±0.2 V or 4.4 V ±0.2 V, whereas the level actually detected is 5.1 V.

[Workaround]

There is no workaround.

This error has been corrected in control code C.

No.14 Change of specification for programmable clock mode register (PCLM)

[Description]

The divider of the processor clock output is not connected as described in the specification.

<Specification of clock divider>

PCK1	PCK0	PCL Clock Output Selection
0	0	fxx/2
0	1	fxx/4
1	0	fxx/8
1	1	fxx/16

<Current clock divider>

PCK1	PCK0	PCL Clock Output Selection
0	0	fxx
0	1	fxx/2
1	0	fxx/4
1	1	fxx/8

[Workaround]

There is no workaround.

This bug has been corrected in control code C.

No.15 Bug in accessing UAnRX register during break

[Description]

An overrun error occurs under the following conditions (a) to (c).

(a) If a break occurs after reading the UART receive buffer register (UAnRX) and the UAnRX register is displayed in the I/O register window of the debugger, an overrun error occurs when UART reception is performed next time.

(b) If a software break occurs immediately after reading the UART receive buffer register (UAnRX), an overrun error occurs when UART reception is performed next time regardless of whether or not the UAnRX register is displayed in the I/O register window.

(c) If a DMA transfer from the UART receive buffer register (UAnRX) is performed during a break[Note], an overrun error occurs when UART reception is performed next time.

Note Including breaks by the RAM monitor function or DMM function. However, the real-time RAM monitor function does not cause this bug because it does not set breaks.

Remark An overrun error also occurs when UART receives data multiple times during a break. This is an emulator specification.

[Workaround]

(a) Do not display the UAnRX register in the I/O register window.

(b) Set a hardware break when setting a break immediately after reading the UAnRX register.

(c) There is no workaround.

Please regard items (a), (b), and (c) as permanent restrictions.

No.16 Bug in accessing CBnRX register during break

[Description]

When the CSIBn receive data register (CBnRX) is read, it usually starts the next reception operation. Under the following conditions (a) and (b), however, the next reception operation is not started even if CBnRX is read.

(a) If a software break occurs immediately after reading the CSIBn receive data register (CBnRX).

(b) If a DMA transfer from the CSIBn receive data register (CBnRX) is performed during a break[Note].

As a result, communication stops or the DMA controller stops.

Note Including breaks by the RAM monitor function or DMM function. However, the real-time RAM monitor function does not cause this bug because it does not set breaks.

[Workaround]

(a) Set a hardware break when setting a break immediately after reading the CBnRX register.

(b) There is no workaround.

Please regard items (a) and (b) as permanent restrictions.

No.17 Bug in accessing CnRGPT register during break

[Description]

Under the following conditions (a) and (b), the read pointer (RGPT) that should be incremented is not incremented, and the same data as previously read is read.

(a) If a software break occurs immediately after reading the CANn module receive history list register (CnRGPT).

(b) If a DMA transfer from the CANn module receive history list register (CnRGPT) is performed during a break$^{\text{Note}}$.

Note Including breaks by the RAM monitor function or DMM function. However, the real-time RAM monitor function does not cause this bug because it does not set breaks.

[Workaround]

(a) Set a hardware break when setting a break immediately after reading the CnRGPT register.

(b) There is no workaround.

Please regard items (a) and (b) as permanent restrictions.

No.18 Bug in accessing CnTGPT register during break

[Description]

Under the following conditions (a) and (b), the read pointer (TGPT) that should be incremented is not incremented, and the same data as previously transmitted is transmitted.

(a) If a software break occurs immediately after reading the CANn module transmit history list register (CnTGPT).

(b) If a DMA transfer from the CAN0 module transmit history list register (CnTGPT) is performed during a break$^{\text{Note}}$.

Note Including breaks by the RAM monitor function or DMM function. However, the real-time RAM monitor function does not cause this bug because it does not set breaks.

[Workaround]

(a) Set a hardware break when setting a break immediately after reading the CnTGPT register.

(b) There is no workaround.

Please regard items (a) and (b) as permanent restrictions.

No.19 Bug in accessing CnGNCTRL register during break

[Description]

When a register access is performed in the following sequence, a forcible shutdown that should not take place normally may occur after the sequence is complete.

[Sequence for bug occurrence]

(1) The EFSD bit of the CANn module control register (CnGMCTRL) is set.

(2) The I/O register[Note] is accessed.

(3) The GOM bit of the CANn module control register (CnGMCTRL) is cleared.

Note I/O register access except for clearing the GOM bit of the CnGMCTRL register

Conditions under which a forcible shutdown takes place are shown below.

(a) If a break occurs immediately after the I/O register access in (2) occurs

(b) If a break by the RAM monitor function or DMM function occurs immediately after the I/O register access in (2) occurs

(c) Stepwise execution is performed for the I/O register access in (2)

[Workaround]

Be sure to set the EFSD bit and clear the GOM bit successively when executing a forcible shutdown.

Do not perform register access in the above sequence when not performing a forcible shutdown.

Please regard this item as a permanent restriction.

No.20 Restriction on TMQn/TMPn and external event counter

[Description]

An interrupt occurs when the compare register of TMQ or TMP overflows (FFFFh is set to 0000h) in the external event counter mode.

[Workaround]

Take measures by software so that the timer compare register of TMQ or TMP is set to 0000h.

Please regard this item as a permanent restriction.

4. Cautions

(1) Emulation of POC, RAM retention flag by action voltage fluctuation is impossible

Emulation can be performed by setting the RAM retention flag of the RAMS register, or by generating a POC reset signal using the PEMU1 register. (Quasi emulation on the LVI interrupt or reset signal can be performed by changing the voltage of the V_{DD} pin (target V_{DD}) or using the PEMU1 register.)

(2) Restrictions on emulation of RAM retention flag

- Restriction on RAM retention flag initialization by forcible emulator reset

  The RAMS register is not cleared to the initial value (01H) after a power is applied by a forcible reset. To set the RAMS register to 01H after a forcible reset, clear bit 2 of the PEMU1 register to 0 using the debugger.

- Restriction on RAM retention flag initialization due to reset during RAM access

  The RAM retention flag is set if a reset occurs during RAM access in the real device, whereas it is not set in this emulator.

(3) Self-programming cannot be emulated

The user program cannot be changed by self-programming.

(4) Oscillation stabilization period after reset release cannot be emulated

The CPU operates without waiting for the oscillation stabilization period to elapse.

The CPU clock cannot be changed from the main clock to Ring-OSC during the oscillation stabilization period. (The CPU clock changes from the main clock to Ring-OSC in the real device if the main clock is stopped during the oscillation stabilization period.)

(5) On-chip debug mode release cannot be emulated when an external reset is input

The control operation (clearing OCDM0 to 0) of the DRSTZ pin that is synchronous to an external reset cannot be emulated because on-chip debug mode is not set during emulation.

(6) POC circuit and clock monitor cannot be emulated

The POC circuit and clock monitor cannot be emulated.

This restriction will be corrected in the next version of the debugger and device file.