Date: May 12, 2016

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-SY*-A001A/E	Rev.	1.00
Title	Restriction on S3 Series MCU		Information Category	Technical Notification		
Applicable Product	Synergy/S3 Series S3A7 R7FS3A77C3A01CFB#AA0 R7FS3A77C3A01CFP#AA0 R7FS3A77C3A01CFM#AA0	Lot No. All lots	Reference Document	Synergy/S3 Series User's Manual: Microcontrollers		

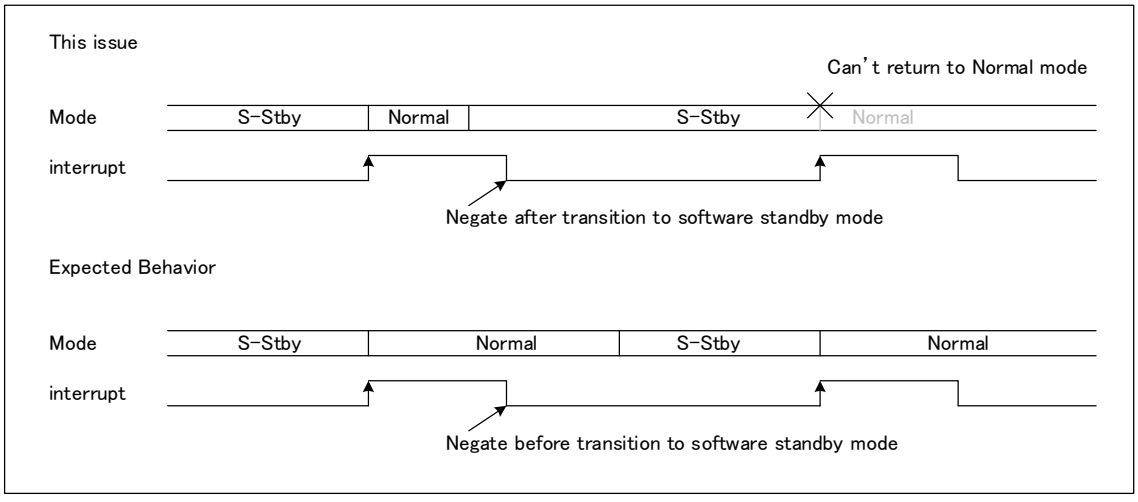
This document describes restrictions on the specified products.

- Restriction on Wakeup issued from Software Standby mode. See Software Standby mode in Chapter 11 Low Power Mode of the *S3 Series – User's Manual: Microcontrollers.*
- Restriction on Security MPU function. See Security MPU in Chapter 16 Memory Protection Unit of the *S3 Series – User's Manual: Microcontrollers.*
- Restriction on SRAM ECC function. See Chapter 47 SRAM of the *S3 Series – User's Manual: Microcontrollers.*

1. Restriction on Wakeup issued from Software Standby mode

1.1. Restriction

When a peripheral interrupt wakes the MCU up from Software Standby mode and the MCU enters Software Standby mode again before the interrupt signal is negated, the same subsequent peripheral interrupt will not be able to wake up the MCU from Software Standby mode.



Peripheral interrupts affected are those that use LOCO and sub-oscillator as clock sources:

- AGT1 interrupt (AGT1_AGTI: Event number 014h)
- AGT1 compare match A (AGT1_AGTCMAI: Event number 015h)
- AGT1 compare match B (AGT1_AGTCMBI: Event number 016h)
- Alarm interrupt (RTC_ALM: Event number 019h)

- Periodic interrupt (RTC_PRD: Event number 01Ah)

1.2. Workaround

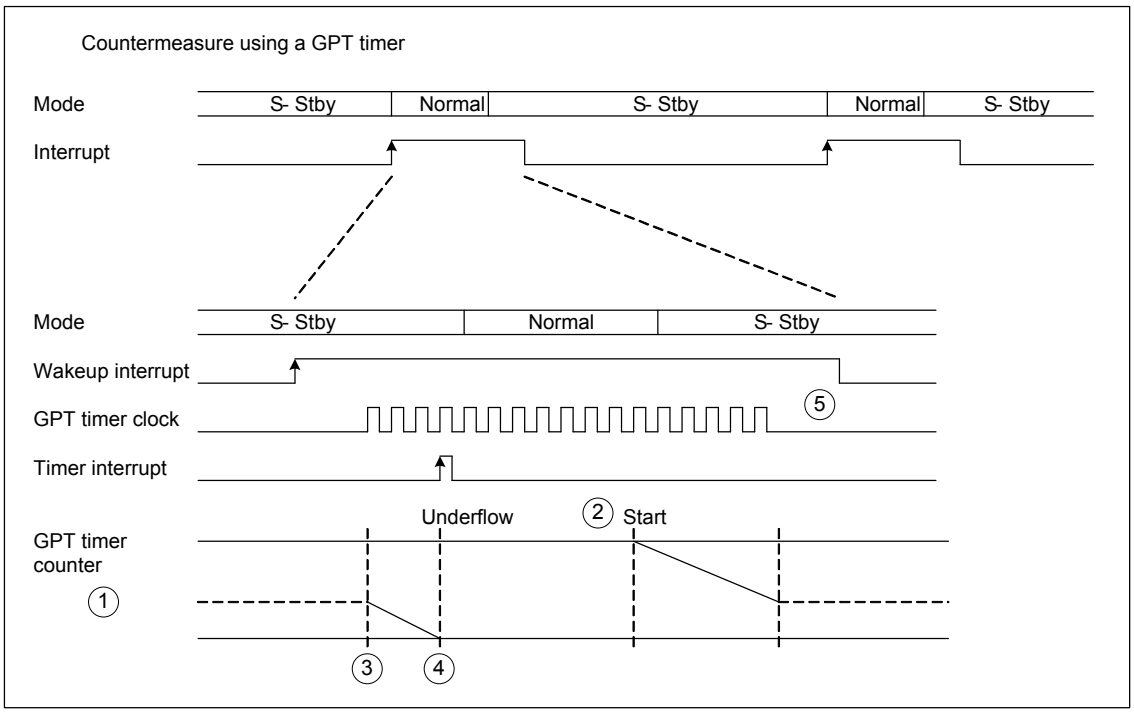
Applications executing the iteration of transition from Software Standby mode to Normal mode, ensure that the peripheral interrupt signal is negated before returning to Software Standby mode.

The following table indicates the assertion period of the peripheral interrupt signal and confirmation method if the peripheral interrupt signal is asserted.

Interrupt Signal	Assertion Period	Confirmation Method
AGT1_AGT1	- If the clock source is Sub-clock or LOCO, assertion period is 1 to 128 clock cycles of clock source -> 30 µs to 3.9 ms - If the clock source is the underflow signal of channel 0 of AGT, the assertion period is pulse width of underflow signal -> 30 µs to 25.6 s	AGT1.AGTCR.TUNDF
AGT1_AGTCMAI		AGT1.AGTCR.TCMAF
AGT1_AGTCMBI		AGT1.AGTCR.TCMBF
RTC_ALM	- During the alarm condition match (depends on alarm registers setting) -> more than 1 s	none
RTC_PRD	- 2 clock cycle of count source -> 60 µs	none

If the application cannot wait for the peripheral interrupt signal to be negated before returning to Software Standby mode, another timer can be used for the return to Normal mode. This requires additional resources: one additional GPT timer and one more interrupt vector.

The following figure indicates the countermeasure using a GPT timer.



1. Choose an available GPT timer and assign the interrupt vector.
2. Before execution of WFI instruction, start the GPT down counter. The GPT will start counting until the clock stops, at which time the MCU transfers to Software Standby mode.
3. When a wakeup interrupt occurs, the GPT clock source and GPT timer are activated. GPT restarts

counting from the previous value. Because wakeup interrupt takes longer to negate, it is not used to wake up the CPU.

4. When a GPT underflow occurs, the GPT interrupt wakes up the CPU to Normal mode. The interrupt handler of the GPT timer should then be stopped, and the application invokes the original wakeup interrupt handler.
5. Perform the same operation to transition to Software Standby mode again.

Note:

- If execution of the WFI instruction is faster than negation of the peripheral interrupt signal, the MCU can wake up from Software Standby mode normally.
- If you have multiple ISRs to wake up from Software Standby mode, stop the GPT timer in the first interrupt handler. Otherwise this GPT timer might generate an unnecessary interrupt.

2. Restriction on Security MPU function

2.1. Restriction

When the FPB function is enabled, the Security MPU function does not function properly.

- When the FPB function is enabled, a secure region cannot be protected from access by a non-secure program.

2.2. Workaround

No workaround exists for this issue since the FPB function is always enabled and the security MPU does not function properly.

3. Restriction on SRAM ECC function

3.1. Restriction

When the SRAM ECC function is enabled, the ECC function does not work properly on 8-bit and 16-bit accesses.

- When the SRAM ECC function is enabled and data with an error in the ECC area is accessed as 8 or 16 bits, the error is not detected, and the SRAM error interrupt signal is not asserted.
- When the SRAM ECC function is enabled and data with an error in the ECC area is accessed as 8 or 16 bits, the ECC code might be changed to the wrong code. Therefore, even when the data is accessed later, the error is not detected because of wrong ECC code.

3.2. Workaround

When the SRAM ECC function is enabled, access the data of the ECC area in 32-bit words.

4. Future plan

These products will be modified. Samples of the modified products will be shipped about March 2016.

[Remarks]

The modified product names are shown in the following table. The final characters are changed.

Package	Products with Restrictions	Modified products	
	MP	CS	MP
144-pin LQFP	R7FS3A77C3A01CFB#AA0	R7FS3A77C3A01CFB#YB1	R7FS3A77C3A01CFB#AA1
100-pin LQFP	R7FS3A77C3A01CFP#AA0	R7FS3A77C3A01CFP#YB1	R7FS3A77C3A01CFP#AA1
64-pin LQFP	R7FS3A77C3A01CFM#AA0	- (not planned)	R7FS3A77C3A01CFM#AA1
121-pin BGA	- (not planned)	- (not planned)	R7FS3A77C2A01CBJ#AA1
145-pin LGA	- (not planned)	- (not planned)	R7FS3A77C2A01CLK#AA1
100-pin LGA	- (not planned)	- (not planned)	R7FS3A77C2A01CLJ#AA1
48-pin QFN	- (not planned)	- (not planned)	R7FS3A77C3A01CNB#AA1