Errata

CC1314R10 SimpleLink™ Wireless MCU Device Revision B

TEXAS INSTRUMENTS

ABSTRACT

This document describes the known exceptions to functional specifications (advisories) to the CC1314R10 SimpleLink™ device.

ADVANCE INFORMATION

Table of Contents

Trademarks

SimpleLink™ and Texas Instruments™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

ADVANCE INFORMATION

1 Advisories Matrix

Table 1-1 lists all advisories, modules affected, and the applicable silicon revisions.

Table 1-1. Advisories Matrix

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED
		B
Radio	Advisory Radio_01 — Proprietary radio modes: spurious emissions can affect regulatory compliance	Yes
Power	Advisory Power_03 — Increased voltage ripple at low supply voltages when DC/DC converter is enabled	Yes
PKA	Advisory PKA_01 — Public key accelerator (PKA) interrupt line is always high when module is enabled and PKA is idle	Yes
PKA	Advisory PKA_02 — Public key accelerator (PKA) RAM is not byte accessible	Yes
I2C	Advisory I2C_01 — I^2C module master status bit is set late	Yes
I2S	Advisory I2S_01 — I^2S bus faults are not reported	Yes
CPU, System	Advisory CPU_Sys_01 — The SysTick calibration value (register field CPU_SCS.STCR.TENMS) used to set up 10-ms periodic ticks is incorrect when the system CPU is running off divided down 48 MHz clock	Yes
System	Advisory Sys_01 — Device might boot into ROM serial bootloader when waking up from shutdown	Yes
System	Advisory Sys_06 — Device may fail to boot at temperatures below -5°C or under certain voltage conditions	Yes
System Controller	Advisory SYSCTRL_01 — Resets occurring in a specific 2 MHz period during initial power up are incorrectly reported	Yes
IO Controller	Advisory IOC_01 — Limited number of DIOs available for the bootloader backdoor	Yes
ADC	Advisory ADC_02 — ADC samples can be delayed by 2 or 14 clock cycles (24 MHz) when XOSC_HF is turned on or off, resulting in sample jitter	Yes

2 Nomenclature, Package Symbolization, and Revision Identification

2.1 Device and Development Support-Tool Nomenclature

To designate the stages in the product development cycle, Texas Instruments™ assigns prefixes to the part numbers of all devices and support tools. Each device has one of three prefixes: X, P, or null (for example, XCC1314R10). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (X/TMDX) through fully qualified production devices/tools (null/TMDS).

Device development evolutionary flow:

X Experimental device that is not necessarily representative of the final device's electrical specifications and may not use production assembly flow.

P Prototype device that is not necessarily the final silicon die and may not necessarily meet final electrical specifications.

null Production version of the silicon die that is fully qualified.

Support tool development evolutionary flow:

TMDX Development-support product that has not yet completed Texas Instruments internal qualification testing.

TMDS Fully-qualified development-support product.

X and P devices and TMDX development-support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

Production devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (X or P) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

2.2 Devices Supported

This document supports the following device:

- CC1314R10

2.3 Package Symbolization and Revision Identification

Figure 2-1 and Table 2-1 describe package symbolization and the device revision code.

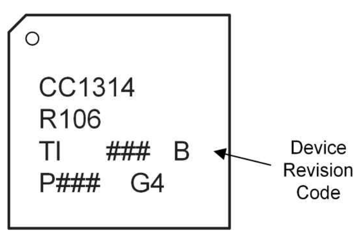

Figure 2-1. Package Symbolization

Table 2-1. Revision Identification

Device Revision Code	Silicon Revision
B	PG2.0

ADVANCE INFORMATION

3 Advisories

Radio_01 — *Proprietary radio modes: spurious emissions can affect regulatory compliance*

Revisions Affected: Revision B

Details: When device internal load capacitors are used with the external 48 MHz crystal, energy couples from the crystal oscillator circuit to the RF output. This coupling causes spurious emissions at N × 48 MHz from carrier frequency. This includes, but is not limited to, the frequency bands supported by the device covered by the following regulations:

When using the +14-dBm RF power amplifier

- ARIB T-108 (Japan)

Workaround: For compliance with affected standards, external load capacitors might be needed for the 48 MHz crystal to reduce spurious emissions. Internal capacitors (default 7 pF connected capacitance) must then be disconnected internally.

This workaround is implemented by defining the following symbols in the included customer configuration file (ccfg.c) available in the examples:

```
#define SET_CCFG_MODE_CONF_XOSC_CAPARRAY_DELTA -128
#define SET_CCFG_MODE_CONF_XOSC_CAP_MOD 0
```

Power_03 — *Increased Voltage Ripple at Low Supply Voltages When DC/DC Converter is Enabled*

Revisions Affected: Revision B

Details: At supply voltages <2.0 V, a hardware control module disables the DC/DC converter to maximize system efficiency. This module does not have enough hysteresis, causing approx 10 mV of ripple on the VDDR regulated power supply. Based on internal testing of the device, it is not anticipated that this erratum affects RF performance. However, these test results cannot ensure that a customer's application or end equipment will not be affected.

Workaround: Use the TI-provided Power driver (PowerCC26X2.c) which automatically disables the DC/DC converter when supply voltage is <2.2V.

The workaround is available in all SDK versions.

PKA_01 *Public Key Accelerator (PKA) Interrupt Line is Always High When Module is Enabled and PKA is Idle*

Revisions Affected: Revision B

Details: When the PKA module is enabled and idle, the interrupt line is always high and the interrupt can thus not be used as is.

Workaround: The workaround is to disable the PKA interrupt in the interrupt service routine while the PKA module is idle and re-enable the interrupt right after starting an operation.

The workaround is implemented in the TI-provided cryptography drivers (ECDHCC26X2.c, ECDSACC26X2.c, ECJPAKECC26X2.c_list.c) available in all versions of the SimpleLink Software Development Kit (SDK) that support this device.

PKA_02 *Public Key Accelerator (PKA) RAM is Not Byte Accessible*

Revisions Affected: Revision B

Details: When accessing the PKA RAM, the RAM is not byte accessible. If a single byte is accessed (read or written), 4 bytes will be accessed instead.

Workaround: The workaround is to use word access (4 bytes) when accessing the PKA RAM.

The workaround is implemented in the TI-provided cryptography drivers (ECDHCC26X2.c, ECDSACC26X2.c, ECJPAKECC26X2.c_list.c) available in all versions of the SimpleLink Software Development Kit (SDK) that support this device.

ADVANCE INFORMATION

I2C_01 — *I^2C Module Controller Status Bit is Set Late*

Revisions Affected: Revision B

Details: The I2C.MSTAT[0] bit is not set immediately after writing to the I2C.MCTRL register. This can lead an I^2C controller to believe it is no longer busy and continuing to write data.

Workaround: Add four NOPs between writing to the MCTRL register and polling the MSTAT register.

The workaround is implemented in the TI-provided I2C Controller driver (I2CCC26XX.c) and in the I2C driver Library APIs (driverlib/i2c.c).

The workaround is available in all Software Development Kit (SDK) versions.

I2S_01 — *I^2S Bus Faults are Not Reported*

Revisions Affected: Revision B

Details: The I^2S module will not set the bus error interrupt flag (I2S0.IRQFLAGS.BUS_ERR) if an I^2S read or write causes a system bus fault that results from access to illegal addresses (usage error).

Workaround: Software must ensure that memory area used by the I^2S DMA is accessible, meaning that the memory is powered on and the system bus is connected..

As an example; The TI-provided SPI driver SPICC26X2DMA.c will ensure that the flash memory is kept accessible also in Idle power mode if the transmit buffer address starts with 0x0 to ensure no bus faults occur. A similar approach needs to be taken if writing a peripheral driver utilizing I2S.

CPU_Sys_01 *The SysTick Calibration Value (Register Field CPU_SCS.STCR.TENMS) Used to Set Up 10 ms Periodic Ticks is Incorrect When the System CPU is Running Off Divided Down 48 MHz Clock*

Revisions Affected: Revision B

Details: When using the Arm® Cortex® SysTick timer, the TENMS register field (CPU_SCS.STCR.TENMS) will always shows the value corresponding to a 48 MHz CPU clock, regardless of the CPU division factor.

Workarounds: One of the following two workarounds must be implemented:

Workaround 1: Do not use a divided down system CPU clock. In general, power savings are maximized by completing a task at full clock speed and then stopping the system CPU entirely after the task is complete.

Workaround 2: Read the system CPU division factor from the PRCM.CPUCLKDIV.RATIO register and compensate the TENMS field in software based on this value.

TI-provided drivers do not offer any functionality to divide the system CPU clock.

Sys_01 *Device Might Boot Into ROM Serial Bootloader When Waking Up From Shutdown*

Revisions Affected: Revision B

Details: For the conditions given below, the device will boot into and execute the ROM serial bootloader when waking up from Shutdown power mode. Intended behavior is to execute the application image. The prerequisites for this erratum to happen are:

- The wake up from Shutdown must be caused by toggling or noise on the JTAG TCK pin and not by a GPIO event.
- The Customer Configuration Section (CCFG) must have configured the bootloader with the following field values:
  - BOOTLOADER_ENABLE = 0xC5 (Bootloader enabled)
  - BL_ENABLE = 0xC5 (Bootloader pin backdoor enabled)
  - BL_PIN_NUMBER = n (any valid DIO number)

With the above prerequisites, the bootloader will be entered in the following cases:

- The CCFG bootloader pin level (BL_LEVEL) is set to 0x0 (active low) AND the input buffer enable for the DIO defined in BL_PIN_NUMBER is disabled in register IOC.IOCFGn.IE. If the input buffer is not enabled, the DIO level will always read 0 and bootloader will be entered.
- The input buffer controlled by IOC.IOCFGn.IE is enabled and the DIO input value is the same level as the CCFG bootloader pin level (BL_LEVEL) when entering Shutdown (GPIO input values are latched when entering Shutdown)

Please refer to the ICEMelter chapter in CC13x4x10, CC26x4x10 SimpleLink™ Wireless MCU TRM for details on how noise entering the JTAG TCK pin can wake up the device

Workarounds: One of the following workarounds must be implemented:

- If input buffer is not enabled, use only active high bootloader pin level (BL_LEVEL)
- If input buffer is enabled, ensure DIO input pin level is not the same as bootloader pin level (BL_LEVEL) when entering Shutdown.

ADVANCE INFORMATION

Sys_06 — *Device can fail to boot at temperatures below -5°C or under certain voltage conditions.*

Revisions Affected Revision B

Details If the device is transitioning from Standby, waking up from Shutdown or any reset srouces, some devices have a probability of resetting and staying in reset. This can occur at temperatures below -5°C or under certain voltage conditions.

Recommended operating conditions for Revision B devices is therefore:

	Min	Max	Unit
Operating ambient temperature	-5	85	°C
Operating junction temperature	-5	85	°C
Operating supply voltage (VDDS)	2.70	3.64	V

Workaround Use recommended operating conditions for Revision B devices as indicated above.

A silicon fix on device Revision C is intended to fix this erratum.

SYSCTRL_01 — *Resets Occurring in a Specific 2 MHz Period During Initial Power Up are Incorrectly Reported*

Revisions Affected: Revision B

Details: If a reset occurs in a specific 2 MHz period during initial power-up (boot), the reset source in AON_PMCTL.RESETCTL.RESET_SRC is reported as PWR_ON regardless of the reset source. This means that there is a window of 0.5 μs during boot where a reset can be incorrectly reported.

Workaround: None

ADVANCE INFORMATION

IOC_01 *Limited number of DIOs available for the bootloader backdoor*

Revisions Affected: Revision B

Details: The highest possible DIO number that can be used for the bootloader backdoor is limited to the number of available GPIOs minus 1. The bootloader backdoor pin is configured through SET_CCFG_BL_CONFIG_BL_PIN_NUMBER in ccfg.c. That means that if the device has x GPIOs, the highest DIO number that can be selected for the bootloader backdoor is DIO_{x-1}, even if higher DIO numbers are available for the device.

Workarounds: There are no workaround for this issue.

ADC_02 *ADC samples can be delayed by 2 or 14 clock cycles (24 MHz) when XOSC_HF is turned on or off, resulting in sample jitter*

Revisions Affected: Revision B

Details: There is no dedicated clock source selection for the ADC clock. The clock is derived from either XOSC_HF or RCOSC_HF, but defaults to XOSC_HF-derived clock whenever this is turned on.

When the ADC clock source is switched from RCOSC_HF to XOSC_HF-derived clock, the clock will stop for 2 cycles (24 MHz).

When the ADC clock source is switched from XOSC_HF-derived clock to RCOSC_HF-derived clock, the clock will stop for additionally 12 clock cycles, as the RCOSC_HF-derived clock is not ready when switch is done.

SCLK_HF switches from RCOSC_HF to XOSC_HF at different times compared to ADC clock. This leads to sample jitter.

Workaround 1: Use asynchronous sampling

- This will reduce the delay of 14 clock cycles down to 2 clock cycles.
- Using asynchronous sampling and an external trigger source (GPIO input pin) will eliminate the delay completely

To use the ADC in asynchronous mode from the Sensor Controller:

Call `adcEnableAsync()`to enable the ADC, instead of `adcEnableSync()`

Example:

```
adcEnableAsync(ADC_REF_FIXED, ADC_TRIGGER_AUX_TIMER0);
```

To use the ADC in asynchronous mode from the System CPU, by using the ADCBuf driver:

```
ADCBuf_Params params;
ADCBufCC26X2_ParamsExtension paramsExtension;

ADCBuf_Params_init(&params);
ADCBufCC26X2_ParamsExtension_init(&paramsExtension);

paramsExtension.samplingMode = ADCBufCC26X2_SAMPING_MODE_ASYNCHRONOUS;
params.custom = &paramsExtension;
```

To use the ADC in asynchronous mode from the System CPU, by using DriverLib API:

Call `AUXADCEnableAsync()` to enable the ADC, instead of `AUXADCEnableSync()`

Example:

```
AUXADCEnableAsync(AUXADC_REF_FIXED, AUXADC_TRIGGER_GPT0A);
```

Please note the difference between the asynchronous and synchronous ADC modes:

ADC_02 (continued) ***ADC samples can be delayed by 2 or 14 clock cycles (24 MHz) when XOSC_HF is turned on or off, resulting in sample jitter***

- In asynchronous mode, the ADC trigger ends the sampling period (which started immediately after the previous conversion), and starts conversion.
- In synchronous mode, the ADC trigger starts the sampling period (with configurable duration), followed by conversion

Workaround 2: Ensure that XOSC_HF is not turned on or off while the ADC is used.

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from September 29, 2022 to June 21, 2023 (from Revision * (September 2022) to Revision A (June 2023))	Page
• Replaced Sys_05 with the encompassing Sys_06 advisory	8

ADVANCE INFORMATION