

Errata

# CC1310 SimpleLink™ Ultra-Low-Power Sub-1 GHz Wireless MCU Silicon Revisions B, A



## ABSTRACT

This document describes the known exceptions to functional specifications (advisories) for the CC1310 SimpleLink™ device.

---

## Table of Contents

1 Advisory Matrix.....	2
2 Nomenclature, Package Symbolization, and Revision Identification.....	3
3 Silicon Revision B Advisories.....	5
4 Silicon Revision A Advisories.....	16
5 Revision History.....	17

## Trademarks

SimpleLink™ is a trademark of TI.

Texas Instruments™ and SmartRF™ are trademarks of Texas Instruments.

Arm® and Cortex® are registered trademarks of Arm Limited.

Motorola® is a registered trademark of Motorola Trademark Holdings, LLC.

All trademarks are the property of their respective owners.

## 1 Advisory 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	A
OSC	<a href="#">Advisory 01</a> , 32-kHz RC Oscillator (RCOSC_LF) Not Working		✓
Digital PLL	<a href="#">Advisory 02</a> , Synthesizer Calibration Might Fail on Rare Occasions	✓	✓
System level	<a href="#">Advisory 03</a> , Reading From Flash While Performing Clock Switching Between the High-Speed Oscillators (XOSC_HF and RCOSC_HF) Will Cause the System to Hang	✓	✓
SSI	<a href="#">Advisory 04</a> , Slave Mode Can Sample New TX Data From SYSBUS Clock Domain Using SSPCLK With No Synchronization	✓	✓
SSI	<a href="#">Advisory 05</a> , Motorola® SPI Format Slave Mode Writes to Transmit FIFO Can Lose Data	✓	✓
Sensor Controller	<a href="#">Advisory 06</a> , Insufficient Power Supply Recharging When Using the Sensor Controller Might Cause the System to Hang or Force a Pin Reset	✓	✓
PRCM	<a href="#">Advisory 07</a> , Wrong Reset Source Indication	✓	✓
Receiver	<a href="#">Advisory 08</a> , Temporary Loss of Receive Function During Continuous Receive Operation Over Long Time	✓	✓
Receiver	<a href="#">Advisory 09</a> , Receive Sensitivity Degradation in Continuous Receive Operation Over Long Time	✓	✓
System level	<a href="#">Advisory 10</a> , Late Time-out of Receiver Operation in Sniff Mode	✓	✓
System level	<a href="#">Advisory 11</a> , Radio Frequency Error Glitch When Device Switches From IDLE to ACTIVE Mode While Radio is in TX or RX	✓	✓
System level	<a href="#">Advisory 12</a> , Slow Transition Across Brown-Out Detect (BOD) Threshold Might Cause the Device to Hang	✓	✓
Digital IOs	<a href="#">Advisory 13</a> , Limited Number of DIOs Available for the Bootloader Backdoor	✓	✓
General-Purpose Timer	<a href="#">Advisory 14</a> , An Incorrect Value Might Be Written to the General-Purpose (GP) Timers MMRs (Memory Mapped Registers) When Simultaneously Accessing the AES (Advanced Encryption Standard) Engine from a Different Master	✓	✓
Radio	<a href="#">Advisory 15</a> , CMD_TX_TEST May End Before Transmitting	✓	✓
ADC	<a href="#">Advisory 16</a> , Periodic ADC Trigger at 200 kHz Rate Can Be Ignored When XOSC_HF Is Turned On or Off	✓	✓
ADC	<a href="#">Advisory 17</a> , ADC Samples Can Be Delayed by 2 Clock Cycles (24 MHz) When XOSC_HF Is Turned On or Off, Resulting in Sample Jitter	✓	✓
ADC	<a href="#">Advisory 18</a> , Writing Any Value to AUX_ANAIF:ADCTRIG.START Will Create an ADC Trigger	✓	✓

## 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, *CC1310F128RGZR*). 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.

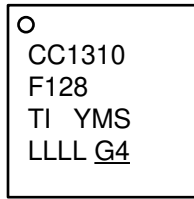
### 2.2 Devices Supported

This document supports the [CC1310](#) device.

There are two revisions of the CC1310 device: revision A and revision B. Revision B is the latest silicon revision. Only CC1310 revision B has been manufactured since April 2017.

## 2.3 Package Symbolization and Revision Identification

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



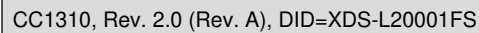
**Figure 2-1. Package Symbolization**

**Table 2-1. Revision Identification**

DEVICE REVISION CODE	SILICON REVISION
A	2.0
B	2.1

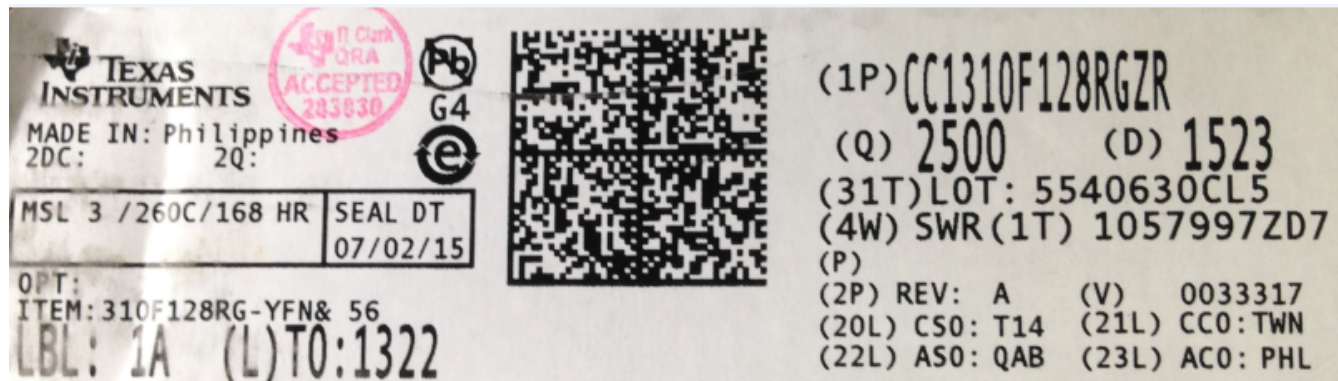
The die markings do not indicate the silicon revision. However, customers can determine the silicon revision by one of the following methods:

- TI provided software functions in chipinfo.c:
  - HwRevision\_t ChipInfo\_GetHwRevision( void )
  - Returns:** chip HW revision
  - HWREV\_2\_0** is returned for CC1310 revision A.
  - HWREV\_2\_1** is returned for CC1310 revision B.
- SmartRF™ Studio:
  - When connecting to a CC1310 device, the version number is shown in the lower-left corner (see [Figure 2-2](#)).
  - Customers can also read out the chip revision using [SmartRF™ Studio version 2.4.3 or later](#).



**Figure 2-2. Device Identification with SmartRF™ Studio**

- Package label:
  - The die revision name is shown on the tape and reel label as shown in the example label in [Figure 2-3](#).
  - Entry (2P) lists the revision (either A or B).



**Figure 2-3. Sample Product Shipping Label for CC1310 Device**

### 3 Silicon Revision B Advisories

This section lists the advisories for this silicon revision.

#### **Advisory 02**      ***Synthesizer Calibration Might Fail on Rare Occasions***

---

**Revision Affected:** A and B

**Details:** The synthesizer calibration has been observed to fail on rare occasions. This will cause any radio command that requires a calibrated synthesizer to fail.

**Workaround:** SIMPLELINK-CC13X0-SDK\_2.20.00.38 and later includes an improved synthesizer start-up sequence that reduces the probability of synthesizer calibration failure.

The workaround for a failed synthesizer calibration is to calibrate again.

#### **TI provided software stacks**

For applications that use the software stacks delivered by TI, no action is required because error handling is implemented by the stacks.

#### **Other (customer developed) software stacks**

For applications that use TI-RTOS or CC13XXWARE together with customer-developed software stacks, TI recommends that proper run-time error handling is performed when executing a radio command. See the software documentation included in [TI-RTOS for Microcontrollers \(MCU\)](#) documentation for error codes.

**Advisory 03**      ***Reading From Flash While Performing Clock Switching Between the High-Speed Oscillators (XOSC\_HF and RCOSC\_HF) Will Cause the System to Hang*****Revision Affected:** A and B**Details:**      The CC1310 device contains five modules that can read from flash independently of each other. These five modules are:

- Arm® Cortex®-M3
- RF Core
- I2S
- $\mu$ DMA
- Crypto (AES module)

Clock switching between XOSC\_HF and RCOSC\_HF can only be initiated by the Cortex-M3. While the Cortex-M3 performs clock switching, no other modules are allowed to read from the flash. The system will hang if any of the other four modules (RF core, I2S,  $\mu$ DMA, or Crypto) are reading from flash during this period.

**Workaround:**      The user must make sure that the Cortex-M3 does not perform clock switching while any of the other four modules (RF core, I2S,  $\mu$ DMA, or Crypto) are reading from flash.**SimpleLink SDK (TI-RTOS)**

Clock switching from RCOSC\_HF to XOSC\_HF is done by calling `Power_setDependency(XOSC_HF)`. The user must register a notification to be notified when the clock switching is completed. When the notification function is called by the power driver, the clock switching is completed and it is safe for all modules to read from flash again.

Clock switching from XOSC\_HF to RCOSC\_HF is done by calling `Power_releaseDependency(XOSC_HF)`. When this function returns, the clock switching is completed and it is safe for all modules to read from flash again. The TI-RTOS radio driver never reads from flash, so it is safe to perform clock switching at all times when using this driver. However, it is up to the user to make sure that no clock switching is performed if the RF Core, I2S,  $\mu$ DMA, or Crypto modules are reading from flash.

**No RTOS**

Clock switching is performed by calling the driverLib API `OSCHfSourceSwitch()`. When this function returns, it is safe for all modules to read from flash again. It is up to the user to make sure that no clock switching is performed if the RF core, I2S,  $\mu$ DMA, or Crypto modules are read from flash.

---

**Advisory 04**      ***Slave Mode Can Sample New TX Data From SYSBUS Clock Domain Using SSPCLK With No Synchronization***

---

**Revision Affected:** A and B

**Details:** When the SSI is programmed to operate in slave mode, the data written to the SSI data register (SSIn:DR) in the SYSBUS clock domain can be sampled in the SSPCLK domain or without any synchronization. This sampling condition occurs when all of the following conditions are met:

- The SSI Transmit FIFO is empty.
- The SSI Data register (SSIn:DR) write access occurs as a new SPI master transfer starts.
- The SSI slave-state machine samples data to transmit.

This issue causes written data to be lost.

**Workaround:** Use TI's [Unified Network Processor Interface \(NPI\)](#).

---

**Advisory 05**      ***Motorola® SPI Format Slave Mode Writes to Transmit FIFO Can Lose Data***

---

**Revision Affected:** A and B

**Details:** If the SSI is configured to operate in Motorola® SPI format slave mode, loss of write data can occur when the following two conditions are met:

1. A write to the SSI data register (SSIn:DR) occurs between a new SPI master transfer starting and the end of the first bit of incoming data.
2. A write to the SSI data register (SSIn:DR) occurs during the first bit of new incoming data in a back-to-back transfer sequence.

For more details, see the Synchronous Serial Interface (SSI) chapter in the [CC13x0, CC26x0 SimpleLink™ Wireless MCU Technical Reference Manual](#).

**Workaround:** Use TI's [Unified Network Processor Interface \(NPI\)](#).

**Advisory 06**      ***Insufficient Power Supply Recharging When Using the Sensor Controller Might Cause the System to Hang or Force a Pin Reset*****Revision Affected:** A and B

**Details:** When the CC1310 device enters standby, a time interval must be set for the initial period between VDDR recharges based on temperature and certain device-specific trims. However, if the Sensor Controller wakes and returns the device to standby without waking the CM3, the initial recharge period programmed by the CM3 is used again by the hardware. If the temperature has increased sufficiently between the time the CM3 put the device into standby and the time the Sensor Controller put the device into standby, the recharge period programmed may be too long. VDDR may drop below the permitted minimum threshold due to the temperature-induced increased leakage. This may cause an inconsistent internal state in the device or cause the device to hang, forcing a pin reset.

**Workaround:** If the device is put in standby while using the Sensor Controller, the user must ensure that the initial recharge period programmed by the CM3 is more conservative than what is strictly necessary at the current temperature. The longest duration that the CM3 sleeps at a specific temperature gradient, defines a maximum limit for temperature increase between recalculations of the initial recharge period. The sleep duration can be altered such that a decrease in the converged recharge period matches the margin subtracted from the initial recharge period.

The following parameters are needed for the workaround:

- The user's maximum expected temperature gradient in °C/s
- The upper bound of the user's expected operating temperature range in °C
- The initial recharge period set at the upper bound of the expected operating temperature range in SCLK\_LF clock periods
- A table of converged recharge periods in SCLK\_LF clock periods at varying temperatures decreasing from the upper bound of the expected operating temperature
- The margin in SCLK\_LF clock periods that the user wants to subtract from the regular initial recharge period. The higher this value, the longer the device can sleep before requiring a wakeup to recalculate the initial recharge period. However, higher values will lead to slower convergence of the recharge algorithm towards the ideal recharge period at a given temperature.

Based on the parameters above, make the following calculations to determine the maximum duration the CM3 can sleep before waking up to recalculate the initial recharge period:

- Add the margin to the converged recharge period at the upper bound of your operating temperature.
- Find the value in the table of converged recharge periods that comes closest to this value and determine the temperature this value occurs at.
- Find the temperature difference between that temperature and the upper bound of your expected operating temperature range.
- Divide this temperature delta by your maximum expected temperature gradient.

The result of the calculations above is a time value in seconds. This value is the longest time the device can safely stay in standby before a CM3 wakeup must be forced if the Sensor Controller is active.

To apply the workaround, convert this duration from seconds to system ticks. When using SIMPLELINK-CC13X0-SDK\_1.60.00.21 or later, apply this value into PowerCC26XX\_Config.maxStandbyDuration, set PowerCC26XX\_Config.enableMaxStandbyDuration to true, and set PowerCC26XX\_Config.vddrRechargeMargin to the value chosen earlier.



**Advisory 07** *Wrong Reset Source Indication*

---

**Revision Affected:** A and B

**Details:** The field RESET\_SRC in the register AON\_SYSCTL:RESETCTL shows the source of the last system reset. Occurrence of one of the reset sources may trigger several other reset sources as essential parts of the system are undergoing a reset. This field will report the source of the reset (not the other resets that are consequence of the system reset). To support this feature, the actual register is not captured before the reset source is released. If a new reset source is triggered within a window of four 32-kHz periods after the previous reset source was released, the register field RESET\_SRC may indicate power-on reset as source regardless of the actual reset source.

**Workaround:** None**Advisory 08** *Temporary Loss of Receive Function During Continuous Receive Operation Over Long Time*

---

**Revision Affected:** A and B

**Details:** The CC1310 modem has a mismatch in the data rates between two modules. If a CC1310 device operates in continuous receive mode for a long period of time without receiving any packets, the mismatch might cause a temporary loss of proper RF reception. This temporary loss occurs at a rate which depends on operating frequency and data rate settings, and lasts for a short period of time before the receive functionality self-recovers.

**Workaround:** Use SIMPLELINK-CC13X0-SDK\_2.20.00.38 or later. For customers using an SDK that pre-dates 2.20, the operational mismatch between the two modules in the modem and the resulting temporary loss in the receiver can be avoided by updating the fractFreq value in the radio commands to generate correct settings for the radio.

SmartRF™ Studio v2.8.0 or later can be used to generate the correct fractFreq value for the radio at each operating frequency and data rate. This (new) fracFreq value can be directly applied to settings generated by older SmartRF™ Studio versions. Except for avoiding the temporary loss in the receiver, RF functionalities and performance are not affected by updating the fractFreq value.

**Advisory 09** *Receive Sensitivity Degradation in Continuous Receive Operation Over Long Time*

---

**Revision Affected:** A and B

**Details:** The CC1310 receiver uses an advanced and dynamic synchronization word search algorithm to optimize the reception of wanted packets and minimize false packet detection. If a CC1310 device operates in continuous receive mode for a long period of time, the dynamic synchronization word search algorithm might in some rare cases lead to unusually high internal threshold values and thereby degraded receive sensitivity. The sensitivity degradation probability depends on the operating noise and interfering conditions and the auto-correlation properties of the synchronization word.

**Workaround:** Use SIMPLELINK-CC13X0-SDK\_2.20.00.38 or later. For customers using an SDK that pre-dates 2.20, a restart of the receiver resets all internal threshold values and ensures optimal performance of the synchronization word search operation and best receive sensitivity.

**Advisory 10** *Late Time-out of Receiver Operation in Sniff Mode*

---

**Revision Affected:** A and B**Details:** The receive sniff mode commands might run the receiver longer than specified for certain settings, leading to increased average current consumption.

When running CMD\_PROP\_RX\_SNIFF or CMD\_PROP\_RX\_ADV\_SNIFF, and the following conditions are met:

- csEndTrigger is set to something other than TRIG\_NEVER and
- the channel is busy when the trigger defined by csEndTrigger and csEndTime is observed,

then the carrier sense operation will always end, regardless of csConf.busyOp.

After this, the receiver will run until a packet is received or the command times out due to endTrigger.

**Workaround:** Use SIMPLELINK-CC13X0-SDK\_2.20.00.38 or later.**Advisory 11** *Radio Frequency Error Glitch When Device Switches From IDLE to ACTIVE Mode While Radio is in TX or RX*

---

**Revision Affected:** A and B**Details:** If the device switches from IDLE to ACTIVE mode (for example, if the CPU starts executing code) while the radio is transmitting or receiving, it can result in a frequency deviation error in the modulated signal or erroneous signal reception.**Workaround:** Keep the flash ON in IDLE by using the TI Power driver API:  
Power\_setConstraint(PowerCC26XX\_NEED\_FLASH\_IN\_IDLE)

**Advisory 12**      ***Slow Transition Across Brown-Out Detect (BOD) Threshold Might Cause the Device to Hang***

---

**Revision Affected:** A and B**Details:** For applications using non-rechargeable (primary) battery, the issue described in this advisory would potentially occur only at end-of-life of the battery, and therefore a workaround is not necessary as the battery would anyway need to be replaced, triggering a power-on reset.

If the VDDS supply voltage is held in the BOD threshold region (approximately 1.78 V), the device might on rare occasions end up in a lock-up state. The current draw is approximately 2.25 mA in this state. The device will not exit this state by increasing the VDDS supply voltage above the BOD threshold. To get out of this state, a pin reset must be performed or the VDDS supply voltage must be decreased below the power-on reset (POR) threshold (1.0 V), triggering a POR reset.

The lock-up state is triggered if a brown-out-detect (BOD) event occurs during specific stages of the boot code execution. There are two critical, narrow time windows, each of approximately 10 ns duration, and both of these time windows occur within 100  $\mu$ s to 1 ms after the reset event that started the boot code execution. Typically, this can happen when the supply voltage is ramped slowly across the BOD threshold. Supply resistance, in combination with device startup current will then pull the VDDS supply voltage below the BOD threshold multiple times as the device turns on and off due to resets.

For Li-Ion and NiMH rechargeable batteries, a first level protection disconnecting the chip VDDS supply would typically prevent the device from entering this state during battery discharge as the device power supply would fall below the POR threshold.

**Workaround:** The following workarounds must be implemented:

The specified operating supply voltage range for the device is 1.8 V to 3.8 V. When using rechargeable batteries, the battery protection system must ensure that either:

- The device supply voltage remains at or above the minimum operating supply voltage (1.8 V) once powered on, or
- If the device supply is discharged below the minimum operating supply voltage (1.8 V), the device must be reset (pin or power-on reset) when the supply is charged above the minimum operating supply voltage (1.8 V) again.

**Advisory 13** *Limited Number of DIOs Available for the Bootloader Backdoor*

---

**Revisions Affected:** A and 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<sub>x-1</sub>, even if higher DIO numbers are available for the device.**Workarounds:** There are no workarounds for this issue.**Advisory 14** *An Incorrect Value Might Be Written to the General-Purpose (GP) Timers MMRs (Memory Mapped Registers) When Simultaneously Accessing the AES (Advanced Encryption Standard) Engine from a Different Master*

---

**Revisions Affected:** A and B**Details:** When writing data to the GP Timer MMRs from one master while simultaneously accessing the AES modules from another master (read/write), an incorrect value might be captured in the GP Timer MMRs. In some cases, the incorrect value is replaced by the correct one after two clock cycles, but not always. No issue is seen when accessing the modules from the same master.**Workaround 1:** Avoid accesses to AES by other masters while writing to the GP Timer MMRs. This can be accomplished by acquiring the relevant semaphores (depending on drivers/stacks) before writing to the GP Timer.**Workaround 2:** Verify the value written to the GP Timer MMR by reading it back in software. Correct the value if necessary.**Advisory 15** *CMD\_TX\_TEST May End Before Transmitting*

---

**Revisions Affected:** A and B**Details:** The CMD\_TX\_TEST may end before transmitting anything if it follows another command in a chain, unless the command directly in front of CMD\_TX\_TEST is CMD\_FS.**Workaround 1:** Insert a CMD\_FS directly before any CMD\_TX\_TEST in the chain.**Workaround 2:** Send the CMD\_TX\_TEST as a standalone command instead of using a chain.

**Advisory 16**      ***Periodic ADC Trigger at 200 kHz Rate Can Be Ignored When XOSC\_HF Is Turned On or Off***

---

**Revisions Affected:** A and 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).

SCLK\_HF switches from RCOSC\_HF to XOSC\_HF at different times compared to the ADC clock. The fact that the clock is stopped, together with the difference in frequency between XOSC\_HF and RCOSC\_HF, may cause the ADC sampling and conversion to finish too late to catch the next trigger.

**Workaround 1:**      Use asynchronous sampling

The sampling period after the issue occurs can be reduced by 1.5% (1 clock cycle at 24 MHz)

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:

- 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.

**Workaround 3:**      Increase the sampling period by 1/24 μs or more.

## Advisory 17 **ADC Samples Can Be Delayed by 2 clock cycles (24 MHz) When XOSC\_HF Is Turned On or Off, Resulting in Sample Jitter**

**Revisions Affected:** A and B

**Details:** There is no dedicated clock source selection for the ADC clock. 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).

SCLK\_HF switches from RCOSC\_HF to XOSC\_HF at different times compared to the ADC clock. This leads to sample jitter.

**Workaround 1:** Use asynchronous sampling

- 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_ASYNC;
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:

- 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.

---

**Advisory 18**      ***Writing Any Value to AUX\_ANAIF:ADCTRIG.START Will Create an ADC Trigger***

---

**Revisions Affected:** A and B

**Details:**            The original register documentation states that you need to write a '1' to AUX\_ANAIF:ADCTRIG.START to manually trigger the ADC. This is not correct, as any write will do this.

**Workaround:**      Avoid writing to AUX\_ANAIF:ADCTRIG.START unless an ADC trigger should be generated.

## 4 Silicon Revision A Advisories

This section lists the advisories for this silicon revision.

Silicon revision-applicable advisories have been found on a later silicon revision. For more details, see [Silicon Revision B Advisories](#).

### Advisory 01 **32-kHz RC Oscillator (RCOSC\_LF) Not Working**

---

**Revision Affected:** A only.

**Details:** RCOSC\_LF (32.768 kHz) may occasionally produce glitches. The most common symptom when using RCOSC\_LF as the low-frequency clock source is that the RTC is advanced or delayed by one SCLK\_LF period. However, there may be other consequences, such as system lock-up, potentially leading to an erroneous digital state and unpredicted behavior.

**Workaround:** The user *must* use XOSC\_LF as the low-frequency clock source (32.768-kHz crystal is required).

For more details, see the [CC13x0, CC26x0 SimpleLink™ Wireless MCU Technical Reference Manual](#).



## 5 Revision History

<b>Changes from , to , (from Revision D (October 2018) to Revision E (December 2020))</b>	<b>Page</b>
• Updated the numbering format for tables, figures, and cross-references throughout the document.....	<a href="#">2</a>
• Added Advisories 13 through 18.....	<a href="#">2</a>

---

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2022, Texas Instruments Incorporated