*Application Note*

# *CC27xx Operation and Configuration Without an External 32kHz Crystal*

*Cashmere Ramos*

## ABSTRACT

The CC27xx device family supports operation without an external 32.768kHz crystal using an internal Low Frequency Oscillator (LFOSC) combined with hardware-assisted LFOSC calibration (LFCAL). This capability allows the device to maintain full Bluetooth Low Energy® (BLE) compliance while reducing system-level dependencies on external components.

To compensate for LFOSC frequency drift during low-power standby operation, the CC27xx integrates a dedicated LFCAL hardware block. This block periodically enables the High Frequency Crystal Oscillator (HFXT) to perform automatic calibration of the internal oscillator (LFOSC), verifying that the device maintains a Sleep Clock Accuracy (SCA) of ±500ppm, the maximum allowable drift permitted by the Bluetooth Core Specification, across varying temperature and supply voltage conditions, without requiring CPU intervention.

The elimination of the external 32.768kHz crystal yields measurable system-level benefits, including reduced bill of materials (BOM) cost, simplified PCB layout, and a smaller overall board footprint, while preserving compliance with Bluetooth Low Energy specification timing requirements.

---

## Table of Contents

## Trademarks

Bluetooth Low Energy® is a registered trademark of Bluetooth SIG.

All trademarks are the property of their respective owners.

# 1 Introduction

This application note describes how to operate Texas Instruments CC27xx wireless MCUs in 32kHz crystal-less mode using the internal Low Frequency Oscillator (LFOSC) with hardware-assisted calibrations to achieve required Sleep Clock Accuracy (SCA). The CC27xx family includes dedicated LFCAL hardware that enables efficient LFOSC calibration during standby mode.

## 1.1 Applicable Devices

This application note applies to all CC27xx devices, including:

- CC2745R10-Q1, CC2745P10-Q1
- CC2745R7-Q1, CC2744R7-Q1
- CC2755P20, CC2755R20

## 1.2 Required Software

- SimpleLink Low Power F3 SDK
- SysConfig

# 2 Quick Decision Guide: When To Use Crystal-less Mode?

When evaluating whether to adopt crystal-less operation, the system design must consider the power budget of the application, connection interval requirements, operating temperature range, and PCB area constraints. Crystal-less mode is preferred for cost and space-sensitive designs where the additional power overhead from LFOSC calibration and widened receive windows is acceptable within the overall system power budget. Designs with stringent low-power requirements or tight cost and space constraints must carefully assess the tradeoffs before eliminating the external 32.768kHz crystal.

## 2.1 Use LFOSC For

**Table 2-1. Advantages of using LFOSC**

| Advantage | Details |
|---|---|
| Cost Reduction | Crystal and load capacitors |
| PCB space limitation | Eliminate crystal footprint and routing constraints |

## 2.2 Tradeoffs

- **Increased Average Current:** This is due to added wake-ups for calibration
- **Wider RX Windows:** Bluetooth Low Energy Receive (RX) windows are wider due to 500 ppm clock accuracy versus 50 ppm with crystal (room temperature)
- **Temperature Sensitivity:** Calibration frequency increases with temperature variation rate

## 2.3 Use External Crystal (LFXT) When

**Table 2-2. Advantages of using LFXT**

| Use Case | Description |
|---|---|
| Lowest Power is Critical | Crystal eliminates calibration overhead and window widening |
| Highest Timing Precision Needed | Crystal provides <50 ppm accuracy, thereby achieving higher accuracy than base clock accuracy of the LFOSC |

# 3 Internal Oscillator Operation

## 3.1 How It Works

Bluetooth Low Energy connection events require precise timing to ensure reliable packet exchange between devices. The Bluetooth Core Specification permits a sleep clock accuracy (SCA) of up to ±500ppm + 16µs, which defines the maximum allowable drift of the low-frequency clock source during standby intervals.

The internal LFOSC is a free-running oscillator whose frequency varies across device operating conditions. To compensate for LFOSC frequency drift, the system employs a periodic hardware calibration mechanism. The LFOSC period is measured against the Ultra Low Leakage (ULL) clock, which serves as an accurate reference during active mode. In the same mode, the ULL clock (CLKULL) operates at 24MHz, derived from the High-Frequency Oscillator (HFOSC) which is derived from the High-Frequency Crystal Oscillator (HFXT). When the device enters standby, the CLKULL switches to using the configured Low Frequency Clock (LFCLK).

A conversion factor, referred to as LFINC, is derived from this measurement and applied by the Real-Time Clock (RTC) to maintain accurate timekeeping. This verifies that Bluetooth Low Energy (LE) connection event scheduling remains within the sleep clock timing accuracy requirements defined in the Bluetooth Core Specification, across the device's operating lifetime.

This calibration mechanism requires the High Frequency Crystal Oscillator (HFXT) to be active. During standby, the HFXT is powered down to minimize current consumption, suspending LFINC updates and leaving any LFOSC frequency drift occurring during this interval uncompensated. LFINC re-calibration resumes upon the next periodic wake-up event or BLE activity that reactivates the HFXT.

### Simplified Example of using LFOSC without LFCAL

### Step 1 - Active Mode

1. HFXT (48 MHz) is active and serves as the accurate reference for HFOSC, producing a tracked HFOSC at 96 MHz for CC27xx devices and at 48MHz for CC23xx devices.
2. CLKULL is derived from the Tracked HFOSC producing 24MHz.
3. The hardware measures the selected LFCLK (LFOSC in this case) period against the CLKULL. For example, for a ~-300ppm deviation, LFOSC frequency = 32.758kHz.
   $CLKULL \div LFOSC\ actual\ frequency = CLKULL\ ticks\ per\ LFTICK$
   $24\ MHz \div 32.758\ kHz = 732.64\ CLKULL\ ticks\ per\ LFTICK$
4. LFINC is updated to 30.527 µs. On every LFTICK, hardware adds LFINC to the RTC TIME counter.
   $1\ LFTICK = 1/32758 = 30.527\ \mu s$
5. To schedule the next connection event, a COMPARE value is set to current TIME + desired interval (1000ms). The RTC fires a wake-up when TIME reaches COMPARE.

### Step 2 - Standby Mode

1. HFXT is disabled and no longer being tracked for HFOSC.
2. CLKULL switches to run on selected LFCLK (LFOSC for this example) LFINC cannot be updated as there is no reference clock to measure LFOSC against.
3. The RTC and WDT remain active using LFINC from previous measurement.
4. LFOSC drifts to +500ppm from nominal (32.7844kHz), corresponding to a change of frequency of approximately 800ppm, with stale LFINC so any LFOSC drift is uncompensated.
   $New\ actual\ LFTICK\ period = 24\ MHz \div 32.7844\ kHz = 30.5023\ \mu s\ per\ tick$
5. The RTC continues to count LFTICK edges, but each tick now represents slightly less real time than LFINC assumes. The timing error accumulates for as long as the drift goes uncompensated. The earlier the drift occurs within the standby period, the larger the error at wakeup. In the worst case, the drift occurs immediately after HFXT is disabled, accumulating over the full standby duration.

### Step 3 - RX Window Timing

1. The RTC counts 32,758 LFTICK edges, expecting 1000.00 ms to have elapsed. Actual elapsed time: 32,758 ticks × 30.5023 μs = 999.194ms
2. The peripheral's estimated RX window opens 806μs too early
3. Ignoring central SCA for simplicity (typically approximately 50ppm), the central device transmits the connection packet at T = 1000.00ms.
4. With a peripheral SCA of 500ppm and the 16μs instantaneous jitter allowance from the Bluetooth Specification, the RX window extends 516μs before and after the estimated anchor point:
   $$RX\ window\ opens: 999.194 - 0.516 = 998.678\ ms$$
   $$RX\ window\ closes: 999.194 + 0.516 = 999.710\ ms$$

The real packet arrives 806μs after the estimated anchor point, 290μs after the window has already closed. The connection packet is missed.

## 3.2 The CC27xx Design: LFCAL Hardware

The CC27xx integrates a dedicated Low Frequency Calibration (LFCAL) hardware block designed to maintain sleep clock accuracy within the ±500ppm bounds permitted by the Bluetooth Core Specification, without requiring CPU intervention. During standby, the LFCAL hardware autonomously executes a periodic calibration sequence consisting of the following operations:

1. Enables the High Frequency Crystal Oscillator (HFXT) to provide the CLKULL reference clock for measurement.
2. Measures the current LFOSC period against the CLKULL reference clock.
3. Derives and applies an updated LFINC conversion factor to the Real-Time Clock (RTC).
4. Disables HFXT and returns the device to low-power standby.

The entire calibration sequence is executed autonomously in hardware. No CPU wake-up event is required, eliminating the associated software. This mechanism verifies that the Bluetooth Low Energy connection event timing remains within specification-compliant bounds throughout standby intervals.

Building on the simplified example above, consider the same scenario with LFCAL. With a compensation profile configured for a 1000ms wakeup interval, the LFCAL hardware performs periodic calibrations during standby, inserting an additional step between Step 2 and Step 3. As a result, Step 3 RX Window Timing now yields a different outcome.

### Step 2a - Calibrations during Standby Mode

1. Before entering standby, SW configured the LFCAL hardware with a compensation profile for the 1000ms connection interval. Only for this specific example, the resulting calibration interval is assumed to be 150ms. The actual value depends on all profile parameters (see Understanding the Compensation Profile).\
2. During standby, LFCAL wakes up at T = 150, 300, 450, 600, and 750ms, measuring LFOSC against a reference and updating LFINC accordingly
3. After the final extra calibration at T = 750ms, LFINC matches the actual LFOSC period. If LFOSC immediately drifts at 500ppm for the remaining 250ms of uncompensated standby, the residual timing error

$$500\ ppm \times 250\ ms = 125\ \mu s$$
$$Estimated\ anchor\ point: 1000.00 - 0.125 = 999.875\ ms$$

**Step 3 - RX Window Timing**

1. The RX Window is now expected to open only 125 µs early compared to 806 µs without LFCAL.
2. Ignoring central SCA for simplicity (typically approximately 50ppm), the central device transmits the connection packet at T = 1000.00ms.
3. With the same assumed peripheral SCA of 500ppm and 16µs instantaneous jitter allowance, the RX window opens around 999.359ms and closes around 1000.391ms which verifies that the packet is received successfully.

Note that the calibration interval used in this example does not represent all compensation profiles using 1000ms as the wake-up interval. See Understanding the Compensation Profile for more details.

## 4 Quick Start Guide

The following steps describe the procedure for enabling crystal-less operation in a CC27xx project using SysConfig and the TI SimpleLink Low Power F3 SDK. All configuration is performed through the SysConfig, which generates the required driver initialization code automatically.

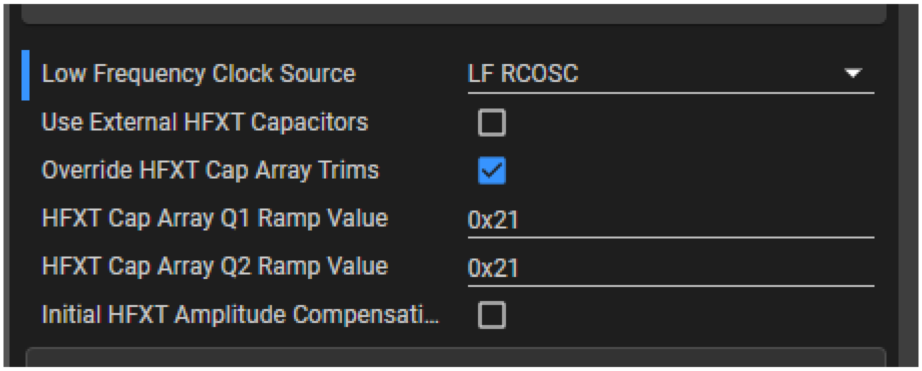
### 4.1 Open SysConfig

Open the target project in Code Composer Studio (CCS) Theia and locate the .syscfg file of the project in the Project Explorer. Double-click the file to launch the SysConfig graphical configuration tool. SysConfig provides a hardware-abstracted interface for configuring device peripherals, clocks, and driver settings without manual register-level configuration.

### 4.2 Select LFOSC Clock Source

Navigate to: TI Devices → Device Configuration → Low Frequency Clock Source

From the available options, select *LF RCOSC* to designate the internal Low Frequency Oscillator (LFOSC) as the LFCLK source. The default selection is *LF XOSC*, which requires an external 32.768kHz crystal. Selecting *LF RCOSC* generates code to call PowerLPF3_selectLFOSC() at startup, which activates the internal LFOSC and routes this to the LFCLK domain.

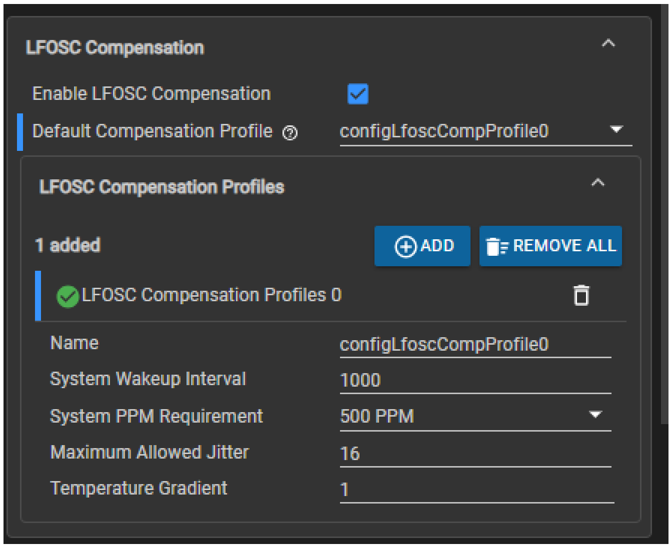


Low Frequency Clock Source: LF RCOSC
Use External HFXT Capacitors: ☐
Override HFXT Cap Array Trims: ☑
HFXT Cap Array Q1 Ramp Value: 0x21
HFXT Cap Array Q2 Ramp Value: 0x21
Initial HFXT Amplitude Compensati...: ☐

**Figure 4-1. LF Clock Source Selection**

### 4.3 Enable LFOSC Compensation

Check the *Enable LFOSC Compensation* checkbox. When LFOSC is the source for LFCLK, LFINC calculation occurs automatically whenever HFXT is available. Enabling this option activates the additional LFCAL wakeups that drive periodic compensation, and instructs the generated code to call PowerLPF3_initLfoscCompensation() and PowerLPF3_enableLfoscCompensation() during startup initialization.



Enable LFOSC Compensation ☑
Default Compensation Profile: configLfoscCompProfile0
Name: configLfoscCompProfile0
System Wakeup Interval: 1000
System PPM Requirement: 500 PPM
Maximum Allowed Jitter: 16
Temperature Gradient: 1

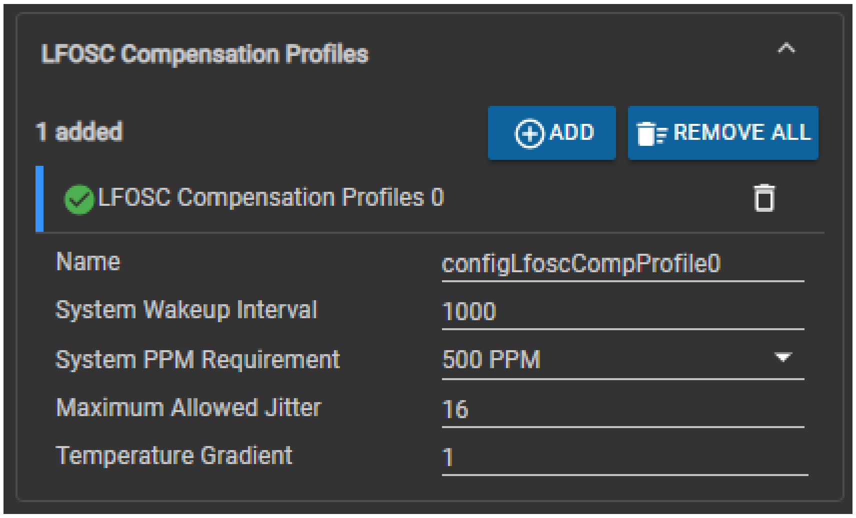
**Figure 4-2. LFOSC Compensation Checkbox**

## 4.4 Add Compensation Profile

Click *Add* under LFOSC Compensation Profiles to create a new profile. The compensation profile defines the operational parameters used by the LFCAL hardware to compute the calibration interval for the target application. Configure the profile fields as described in the following table:

**Table 4-1. Compensation Profile Values**

| Parameter | Typical Value | Description |
|---|---|---|
| System Wakeup Interval (us) | 1000000 (1s) | Defined connection interval |
| System PPM Requirement | 500ppm | Required for Bluetooth Low Energy compliance |
| Temperature Gradient | 1°C/s | Expected worst-case temp change rate |



Name: configLfoscCompProfile0
System Wakeup Interval: 1000
System PPM Requirement: 500 PPM
Maximum Allowed Jitter: 16
Temperature Gradient: 1

**Figure 4-3. Profile Configuration with 1ms Wakeup Interval**

## 4.5 Set as Default Profile

Select the configured profile as the *Default Compensation Profile*. The default profile is automatically applied at startup by the generated initialization code, verifying that LFOSC compensation is active before the BLE stack initiates the first connection event. Multiple compensation profiles can be defined within a single project to accommodate applications with varying operational modes, such as different connection intervals or duty cycles. However, only one profile can be designated as the default. Profiles can be switched at runtime using PowerLPF3_setLfoscCompensationProfile() to adapt the calibration interval to changing application conditions.

## 4.6 Build and Test

Once SysConfig has been configured for crystal-less operation, the generated initialization code invokes the following sequence of Power driver API calls to activate LFOSC compensation.

Building a project automatically adds the following code to the SysConfig generated Board_init() function:

```
/* activates the internal LFOSC and switches the low-frequency clock (LFCLK) source to the LFOSC.
Dynamic switching between low-frequency clock sources is currently not supported.*/
PowerLPF3_selectLFOSC();
/* initializes the internal state of the LFOSC compensation subsystem and configures the LFCAL
hardware measurement window. The default calibration measurement duration is 1500µs.. */
PowerLPF3_initLfoscCompensation();
/* supplies the compensation profile that governs how the LFCAL hardware computes the optimal
calibration interval. The profile is defined by the PowerLPF3_LfoscCompensationProfile structure.
*/
PowerLPF3_setLfoscCompensationProfile(&configLfoscCompProfile0);
/* activates the LFCAL hardware block; the driver reads the current device temperature, computes
the initial calibration interval based on the active profile, and begins the autonomous periodic
calibration cycle. Temperature change notifications are registered internally to dynamically adjust
the calibration interval ensuring that sleep clock accuracy remains within ±500ppm under varying
thermal conditions. */
PowerLPF3)enableLfoscCompensation();
```

Build the project and verify correct operation by confirming that BLE connection events are maintained without timing errors during extended standby intervals.

# 5 Configuration Details and Power Consumption

## 5.1 Understanding the Compensation Profile

The PowerLPF3_LfoscCompensationProfile structure defines the operational parameters used by the LFCAL hardware to compute the calibration interval for the target application.

Applications often involve different states throughout operation, each requiring different wakeup intervals and timing characteristics. For example, an application might spend time in an advertising state with a long wakeup interval, then transition to an active connection state with a much shorter connection interval. While using a single profile across all application states is possible, it must be configured for the worst-case conditions: the shortest wakeup interval, the most stringent ppm requirement, the highest temperature gradient, and the smallest allowed jitter. Because the calibration interval is driven by the worst-case (shortest) wakeup interval, the LFCAL hardware will calibrate more frequently than necessary during states with longer wakeup intervals. To optimize power consumption, multiple compensation profiles can be defined, allowing the LFCAL hardware to use the calibration interval best suited for the current operating conditions. A default profile is applied when the user does not explicitly assign one. The following fields configure a compensation profile:

**Table 5-1. Compensation Profile Description**

| Field | Description | Default Value | Unit |
|---|---|---|---|
| systemWakeupIntervalUsec | Expected interval between consecutive system wakeup events, in microseconds; usually corresponds to the configured connection interval; | 1000000 | µs |
| ppmRequirement | The required Sleep Clock Accuracy (SCA) expressed in parts per million. | 500 | ppm |
| temperatureGradientMilliCelsiusPerSec | The worst-case value of the temperature rate of change; the larger the gradient the higher the average current | 1 | C/s |
| maxAllowedJitterUsec | Maximum additional per-wakeup timing jitter tolerance, in microseconds, beyond the ppm requirement; the smaller the value, the higher the average current | 16 | µs |

The default value of 16µs reflects the maximum instantaneous per-wakeup jitter permitted by the Bluetooth Specification, which requires that all instantaneous timings shall not deviate more than 16 µs from the average timing derived from the sleep clock accuracy (≤ ±500ppm). The driver uses this budget to extend the calibration interval and reduce average current consumption.

The LFCAL compensation model accounts for the baseline LFOSC error of 200ppm (LFOSC_BASELINE_PPM), which represents the LFINC filter accuracy across process, voltage, and temperature (PVT) variation. This baseline is independent of sleep duration and does not include frequency perturbation contributions. The calibration interval is computed such that the total accumulated error, comprising temperature drift, frequency perturbations, and this baseline offset, remains within the ±500ppm Sleep Clock Accuracy requirement permitted by the Bluetooth Core Specification.

The LFOSC temperature coefficient, expressed in ppm/°C, is stored in the Factory Configuration (FCFG)

- fcfg->appTrims.cc27xx.misc0.lfoscPpmTempMid
- fcfg->appTrims.cc27xx.misc0.lfoscPpmTempExt

The Power driver reads these values automatically at runtime through CKMDGetLfoscMidTempCoefficientPpmPerC() and CKMDGetLfoscExtTempCoefficientPpmPerC() when computing the LFCAL calibration interval. No explicit application-level access to the FCFG is required. The power driver accounts for the temperature coefficient defined in FCFG and the temperature gradient defined by the user to determine how frequently the LFCAL hardware must calibrate to maintain the requirement.

## 5.2 Profile Examples

The following profile examples illustrate representative LFCAL configurations for common Bluetooth Low Energy application scenarios. Current consumption comparisons between LFOSC-based profiles and a crystal (LFXT) reference are provided for each configuration. The following examples are taken from LP_EM_CC2745R10-Q1. Table 5-2 compares the average current consumption per interval when using a crystal oscillator versus the internal RC oscillator. Table 5-3 lists how the average current consumption per interval varies across different configured wakeup intervals.

Note that although higher intervals require more wake ups, the interval between each calibration also increases.

Fixed values:

- Allowed jitter: 16µs
- Temperature Gradient: 1C/s,
- Temperature: 25°C

**Table 5-2. Measured Current Comparison for LFXT vs LFOSC**

| Clock Source | Connection Interval | TX power (dBm) | Average current for one interval (uA) | #of added wakeups |
|---|---|---|---|---|
| LFXT | 30 ms | 0 | 222 | - |
| LFOSC | 30 ms | 0 | 266 | 1 |
| LFXT | 30 ms | 10 | 265 | - |
| LFOSC | 30 ms | 10 | 308.5 | 1 |

**Table 5-3. Computed Current Comparison for Different User Configuration for LFOSC**

| Connection Interval (ms) | Expected Standby Average current (uA) | Expected Average Current for one interval (uA; 0dBm, 3.3V) | #of Added Wakeups |
|---|---|---|---|
| 30 | 46 | 266 | 1 |
| 100 | 14.3 | 80.5 | 2 |
| 250 | 7 | 33.7 | 3 |
| 500 | 4.4 | 17.8 | 4 |
| 750 | 3.5 | 12.4 | 4 |
| 1000 | 3.1 | 9.8 | 5 |

## 5.3 Power Policy

The Power driver policy must be configured to PowerCC27XX_standbyPolicy to enable standby mode.

In SysConfig, the power policy can be set by navigating to: TI Drivers → Power and selecting PowerCC27XX_standbyPolicy as the active policy function.

# 6 Window Widening

In a Bluetooth Low Energy (BLE) connection, the Central and Peripheral devices exchange data at precise intervals defined as connection events. The timing of each connection event is governed by the connection interval parameter during the connection setup procedure, as defined in the Bluetooth Core Specification. Between consecutive connection events, both devices enter a low-power standby state to minimize current consumption.

## 6.1 The Challenge

While the device is in standby mode, the sleep clock of each device drifts independently at its own rate. As a result, when each device wakes in anticipation of the next connection event, neither the Central nor the Peripheral can determine with certainty the precise wake-up timing of the other. This clock uncertainty must be accounted for to verify reliable packet reception at each connection event.

## 6.2 The Design

Window widening is the mechanism defined in the Bluetooth Core Specification to compensate for sleep clock uncertainty between connected devices. The Peripheral opens its receive (RX) window earlier than the anticipated anchor point and maintains it for an extended duration to ensure successful reception of the Central's packet, despite accumulated clock drift. No additional configuration is required to enable this mechanism; it is applied automatically as part of the Bluetooth Low Energy connection procedure.

The receive window duration is determined by the combined Sleep Clock Accuracy (SCA) of both devices and the elapsed time since the last synchronization anchor point, as expressed by the following relationship:

Window Width = 2 × (SCA_peripheral + SCA_central) × time_since_last_anchor

where SCA values are expressed in parts per million (ppm) and time_since_last_anchor represents the elapsed time since the last successfully received anchor point.

When operating with an LFOSC at ±500 ppm, compared to a crystal-based oscillator at ±20–40 ppm, and assuming a typical Central SCA of 50 ppm, the resulting RX window duration is approximately six to eight times wider than that of a crystal-based implementation.

See the *Window widening* section in the Bluetooth Core Specification for more information.

# 7 Hardware Design

## 7.1 What to Remove

When using LFOSC mode, following components can be removed from the BOM:

- 32.768kHz crystal
- Two load capacitors (typically 6.8-15pF)

## 7.2 LFXT Pins

The LFXT pins can be:

- Left unconnected (no external components needed)
- Used as GPIO/SPI/UART for other functions

- See this product page for more information.

## 7.3 HFXT Requirements

The 48MHz HFXT is always required as the HFXT serves as the calibration reference and RF carrier frequency reference clock.

## 7.4 Board Bring-Up Checklist

1. Start with basic_ble example
2. Configure LFOSC in SysConfig as described in Quick Start
3. Build and flash
4. Verify the Bluetooth Low Energy connection with smartphone
5. Monitor for connection stability
6. Gradually add application features

## 8 Summary

The CC27xx device family supports crystal-less BLE operation by replacing the external 32.768kHz crystal with an internal Low Frequency Oscillator (LFOSC) combined with a dedicated hardware calibration block (LFCAL). This capability eliminates the need for the crystal and the associated load capacitors from the BOM, reducing system cost, simplifying PCB layout, and shrinking board footprint, while maintaining full compliance with the Bluetooth Core Specification's ±500ppm Sleep Clock Accuracy (SCA) requirement.

The LFCAL hardware operates autonomously during standby mode without CPU intervention, periodically waking up the High Frequency Crystal Oscillator (HFXT) to recalibrate the LFOSC and update the LFINC conversion factor used by the Real-Time Clock. This periodic recalibration bounds accumulated timing error so that BLE connection event scheduling remains within specification, even under varying temperature and supply voltage conditions. The 48MHz HFXT must always remain available on the board, as it serves as both the calibration reference and the RF carrier frequency reference clock.

Crystal-less mode involves well-understood tradeoffs relative to an external crystal implementation. Average current consumption is higher due to additional LFCAL wake-up events, and BLE receive windows are wider than those achievable with a crystal-based oscillator. Calibration frequency also increases with faster temperature variation rates, making the temperature gradient a key parameter when configuring the compensation profile.

Configuration is performed entirely through SysConfig and the SimpleLink Low Power F3 SDK, requiring no manual register access. The PowerLPF3_LfoscCompensationProfile structure allows designers to define one or more profiles, each specifying the wakeup interval, PPM requirement, temperature gradient, and jitter tolerance, enabling calibration intervals across different application states such as advertising and active connection. Multiple profiles can be switched at runtime through PowerLPF3_setLfoscCompensationProfile() to adapt dynamically to changing operating conditions.

## 9 References

1. Texas Instruments, *CC2745R10-Q1*, product page.
2. Bluetooth, *Core Specifications 6.2*, specifications.

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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 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, TI’s General Quality Guidelines, or other applicable terms available either on 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

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

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025