

Running Bluetooth® low energy on CC2640 Without 32 kHz Crystal

*Christin Lee and Fredrik Kervel**Bluetooth low energy Solutions*

ABSTRACT

This application report explains how you can configure the SimpleLink™ Bluetooth low energy CC2640 wireless MCU, CC2640R2F wireless MCU and multi-standard CC2650 wireless MCU to run the Bluetooth low energy (BLE) software stack without the need for a 32 kHz crystal.

This application report explains how to configure this mode of operation, what considerations have to be taken to use the internal RC low frequency oscillator (RCOSC_LF) for the Bluetooth low energy peripheral and broadcast (beacon) role devices instead of the 32 kHz crystal, and what impact it has on current consumption. This document assumes the reader is familiar with the concepts described in [CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack Developer's Guide](#) and the [CC13xx, CC26xx SimpleLink™ Wireless MCU Technical Reference Manual](#).

Removing the 32 kHz crystal from a design lowers the bill of material (BOM) cost, reduces the required board space and simplifies procurement.

Contents

1	Introduction	2
2	Requirements	2
3	Configuration	3
4	Performance	5
5	Recommendations	6
6	References	7

List of Figures

1	Selected Target	2
2	Current Consumption vs Connection Interval	6

List of Tables

1	Current Consumption for Connection Event From SimpleBLEPeripheral	6
---	-------------------------------------------------------------------------	---

Trademarks

SimpleLink is a trademark of Texas Instruments.
iBeacon is a registered trademark of Apple Inc.
Bluetooth is a registered trademark of Bluetooth SIG, Inc.
Eddystone is a trademark of Google Inc.
All other trademarks are the property of their respective owners.

1 Introduction

The Bluetooth specification puts a strict requirement on the accuracy of the sleep clocks for Bluetooth low energy devices that are intended to enter (and stay) in a connection. The specification requires that devices have a sleep clock accuracy (SCA) that meets ± 500 ppm. For more details on the SCA requirement, see the *Sleep Clock Accuracy* section in the Bluetooth core specification 4.0, Volume 6. that can be downloaded from the following URL: <https://www.bluetooth.com/specifications/adopted-specifications>.

This sleep clock accuracy (SCA) requirement is valid for both the master and slave side of the Bluetooth low energy connection; however, devices such as Bluetooth low energy beacons do not require such accuracy as the advertising interval is intentionally varied to prevent collisions. The intention of the SCA requirement is to ensure low power consumption while maintaining flexibility in the component selection. The amount of time a slave device must stay in active RX mode, referred to as the receive window, is dependent upon the sleep clock tolerance; a less accurate sleep clock requires the receive window time to be increased, thus, increasing the average current consumption.

2 Requirements

2.1 CC2640F128 and CC2650F128 Hardware Revision

NOTE: This section is not applicable to the CC2640R2F. All revisions of CC2640R2F supports using the 32 kHz RC oscillator.

In order to use the 32kHz crystal less feature on CC2640F128 or CC2650F128, a device with silicon revision 2.3 or later is required. For more details, see the [CC2640 SimpleLink™ Wireless MCU Errata](#).

There are three ways to determine the chip revision:

- Use `ChipInfo_GetMinorHwRev()`, which can be found in `chipinfo.h` in our driverlib. This function returns 0 for the older revision and 1 or higher for later revisions (2.3 or newer). Only the latter supports usage of the 32 kHz RC oscillator.
- Check the revision label on the box the chips comes shipped in. The revision should be “D” or higher.
- Use [Flash Programmer 2](#). You can find the revision number at the lower left corner as shown below:

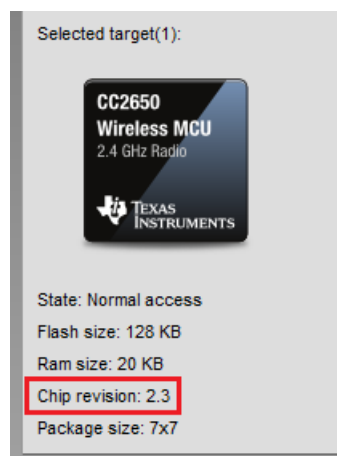


Figure 1. Selected Target

CAUTION

Relying on the RCOSC_LF oscillator as a sleep clock on a CC2640F128 or a CC2650F128 device with revision 2.2 or earlier can lead to the application hanging, resulting in an unresponsive state, and leads to unexpected and unintentional behavior.

2.2 BLE-Stack Software Versions

Using the internal 32 kHz RC oscillator (RCOSC_LF) as Bluetooth low energy sleep clock requires performing a periodic, software-based calibration of the RCOSC_LF oscillator. This functionality is included in the royalty-free TI Bluetooth low energy software protocol stack (BLE Stack) versions 2.2.x and TI SimpleLink BLE-Stack 3.x.x. The BLE-stack SDK can be downloaded from www.ti.com/ble-stack.

2.3 Constraints

The RCOSC_LF calibration is supported for the whole CC2640 temperature range, but care must be taken regarding temperature gradients. To stay within the sleep accuracy requirement of ± 500 ppm, the maximum temperature change per calibration interval cannot be higher than 1°C, with the default calibration interval being 1 second.

With a supported Bluetooth low energy SDK version, the calibration routine will be enabled when selecting RCOSC_LF build configuration as the sleep clock source or modify the project according to the software configuration section (for applications not requiring sleep clock accuracy, the calibration can be manually disabled. For example, non-connectable advertisement used by beacon applications). The calibration will then run automatically every time when the more precise 24 MHz starts.

To maintain ± 500 ppm accuracy, the calibration must run at least every second assuming the CC2640 temperature variation does not exceed 1°C per second. In applications with effective Bluetooth low energy connection intervals higher than 1 s (time in Standby mode > 1 s), wake-ups must be scheduled at least every second to perform the RCOSC_LF calibration. Similarly, if the application is active for longer time periods than 1 s, the calibration must be triggered by the application. Using the supported build configurations in [Section 3.2](#), the BLE-Stack software will automatically handle these calibration requirements.

The 32 kHz crystal-less feature is supported on CC2640/CC2650 wireless MCUs implementing the Bluetooth low energy peripheral, observer and broadcast (beacon) roles only. Thus, all central role or master devices must use the 32 kHz crystal oscillator.

3 Configuration

3.1 Hardware Configuration

No specific hardware configuration is required to run on the internal 32 kHz RC oscillator (RCOSC_LF). The 32 kHz crystal pins will be in a Hi-Z state when not used and can safely be tied to any logic level, or left unconnected.

3.2 Software Configuration

To configure the device to run off the calibrated RCOSC_LF (low frequency oscillator), the following project modifications must be done. The following examples detail the configuration steps using the BLEStack v2.2 SDK and TI SimpleLink BLE-Stack 3.00.00 Software Development Kit for the IAR, Code Composer Studio™ (CCS) Integrated Development Environments (IDEs).

1. Include rcosc_calibration.c, rcosc_calibration.h and ccfg_app_ble_rcosc.c files that are located at <SDK_INSTALL_DIR>\examples\rtos\CC2640R2_LAUNCHXL\ble\common\cc26xx\rcosc or <SDK_INSTALL_DIR>\src\common\cc26xx\rcosc.
2. Exclude ccfg_app_ble.c from build.
3. Add USE_RCOSC to Defined symbols.
4. Add the following code to your peripheralproject.c

```
#ifndef USE_RCOSC
#include "rcosc_calibration.h"
#endif //USE_RCOSC
```

5. Add the following code to your peripheralproject_init function in peripheralproject.c

```
#ifndef USE_RCOSC
RCOSC_enableCalibration();
#endif // USE_RCOSC
```

6. If using a custom board file, enable the RCOSC in the power policy. The board files included with the BLE-Stack:

```
PowerCC26XX_Config PowerCC26XX_config = {
    .policyInitFxn      = NULL,
    .policyFxn          = &PowerCC26XX_standbyPolicy,
    .calibrateFxn       = &PowerCC26XX_calibrate,
    .enablePolicy       = TRUE,
    .calibrateRCOSC_LF  = TRUE,
    .calibrateRCOSC_HF  = TRUE,
}
```

7. Constrain the temperature variation to be less than 1°C/sec. If the temperature is to change faster than 1°C/sec, then a short calibration interval must be used. Calibration interval can be tuned in rcosc_calibration.h

```
// 1000 ms
#define RCOSC_CALIBRATION_PERIOD 1000
```

3.2.1 Software Overview

RCOSC_LF calibration is done by the TI-RTOS power driver. Each time the device wakes up from standby, a calibration is performed on the RCOSC_LF using the more precise, high frequency crystal oscillator (XOSC_HF) as the reference. This calibration ensures the sleep clock source is within the ± 500 ppm accuracy requirement.

The aforementioned software configuration section takes care of the calibration of RCOSC_LF. The implementation can be found in rcosc_calibration.c and rcosc_calibration.h source files in the BLE-Stack SDK.

In our BLE Stack, the RCOSC_LF calibration is performed under the following situations:

- Standby time is longer than 1 second (for example, longer effective connection intervals)
 - Schedule a wakeup with 1s period from the last active event by using Power_registerNotify function as shown below and a one shot clock

```
// Receive callback when device wakes up from Standby Mode.
Power_registerNotify(&injectCalibrationPowerNotifyObj,
PowerCC26XX_AWAKE_STANDBY,
(Power_NotifyFxn)rcosc_injectCalibrationPostNotify, NULL);

// Create RCOSC clock - one-shot clock for calibration injections.
Util_constructClock(&injectCalibrationClock,
    rcosc_injectCalibrationClockHandler,
    RCOSC_CALIBRATION_PERIOD, 0, false, 0);
```

- Active time is longer than 1 second
 - The clock object expires and calls the following function:

```
// Inject calibration.
PowerCC26XX_injectCalibration();
```

The differences between ccfg_app_ble.c and ccfg_app_ble_rcosc.c are listed below

- SET_CC_CFG_MODE_CONF_VDDR_TRIM_SLEEP_TC: This parameter determines whether or not to change the lowest VDDR voltage level in standby mode according to temperature. RCOSC_LF drifts more when the device is in cold temp and a higher VDDR sleep trim value decreases the temperature sensitivity of RCOSC_LF. Enabling this functionality can ensure that the 1s calibration interval will result in SCA stays within ± 500 ppm across all temperature assuming temperature variation is smaller than 1°C/sec. This results device wakes up more under cold temperature than room temp and high temperature.
- SET_CC_CFG_MODE_CONF_SCLK_LF_OPTION: This parameter determines the clock source for low frequency system clock (SCLK_LF).

3.2.2 Changes for Beacon/Broadcaster Application

1. Select the FlashROM build configuration in the simple_broadcaster application project (\examples\cc2650em\simple_broadcaster).
2. Exclude ccfg_app_ble.c under Startup folder from build.
3. Include ccfg_app_ble_rcosc.c under Startup folder from build.
4. Disable temperature compensation of VDDR level as for beacon application, SCA does not require to be within ± 500 ppm.

```
#define SET_CCFG_MODE_CONF_VDDR_TRIM_SLEEP_TC      0x1          // Temperature compensation
on VDDR sleep trim disabled (default)
```

5. Disable RCOSC_LF calibration in the board files.

```
const PowerCC26XX_Config PowerCC26XX_config = {
    .policyInitFxn      = NULL,
    .policyFxn          = &PowerCC26XX_standbyPolicy,
    .calibrateFxn       = &PowerCC26XX_calibrate,
    .enablePolicy       = TRUE,
    .calibrateRCOSC_LF  = FALSE,
    .calibrateRCOSC_HF  = TRUE,
};
```

4 Performance

4.1 Current Consumption

Using the internal RCOSC_LF as the sleep clock has a net effect on the device current consumption as compared to board designs that utilize an external 32 kHz crystal oscillator. The difference in current consumption varies depending on the configured role of the device. For peripheral (slave) devices in a Bluetooth low energy connection, the current consumption will be higher when using the RCOSC_LF as compared to using an external 32 kHz crystal; however, the increase in current consumption is dependent on a number of factors. This increase is due to:

- Performing the calibration at a certain interval
- The extended receive window due to the maximum allowed sleep clock accuracy (± 500 ppm, vs. typically ± 40 ppm with a 32 kHz crystal).

During periods where the CC2640 is advertising (for example, as a beacon or waiting for a connection request) or in standby (while idle), current consumption using the internal RCOSC_LF will be less (better) than using a 32 kHz crystal oscillator.

The calibration process itself takes approximately 1 ms, and for a typical Bluetooth low energy connection event the calibration will happen in the background while the radio operates. In most cases, the added current consumption from performing the calibration will thus be negligible. In configurations with longer effective Bluetooth low energy connection intervals, that is the connection interval with the maximum slave latency applied is greater than 1 second, there will be additional power consumption because the device has to wake up from standby (sleep) between the connection events to perform the RCOSC_LF calibration. In a board design that uses a 32 kHz crystal, these calibration wakeups would not be required.

The average current consumption using the 32 kHz crystal as compared to using the internal RCOSC_LF for some Bluetooth low energy effective connection intervals can be seen in [Table 1](#) and [Figure 2](#). To get actual current consumption for any given configuration, follow the measurement procedure in [Measuring Bluetooth Smart Power Consumption](#).

Table 1. Current Consumption for Connection Event From SimpleBLEPeripheral

Connection Interval [ms]	Average Current w. RC OSC [μ A]	Average Current w. XOSC [μ A]
10	963.3	897
50	194.5	180
100	98.3	90.8
500	23.9	19.7
1000	15.4	10.8
4000	11.7	4.2

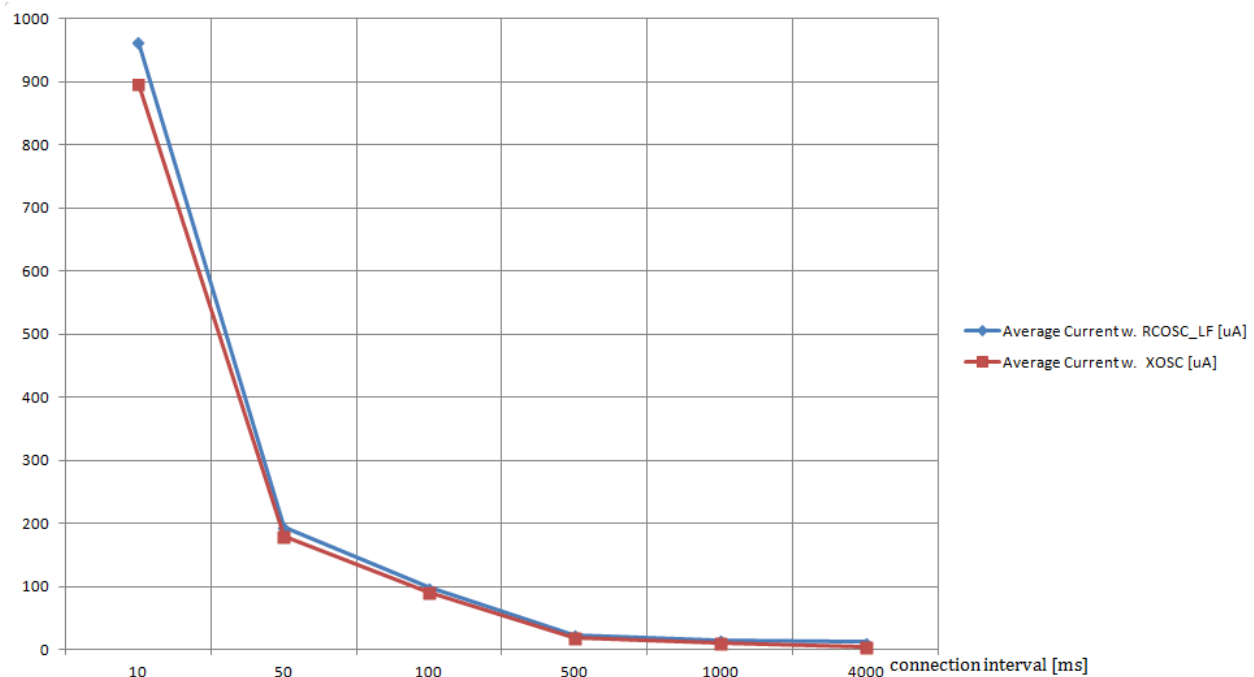


Figure 2. Current Consumption vs Connection Interval

5 Recommendations

Use of the 32 kHz crystal-less feature should be considered for the following BLE operating conditions:

- Where the lowest possible BOM cost is desired or when board layout space is limited.
- Peripheral role devices (slave) that maintain short (fast) connection intervals or enter BLE connections infrequently and remain idle or advertising most of the time. Example devices include door locks, light bulbs, blood glucose meters (BGMs) and fitness/activity trackers.
- Beacon or broadcast role devices, such as Apple iBeacon® location and proximity detection technology and Eddystone™ an open beacon format by Google. These devices do not typically form connections and spend most of the time performing BLE advertising. These devices will achieve better (lower) current consumption with the RCOSC_LF than using an external 32 kHz crystal.

6 References

- [CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack 2.2.1 Developer's Guide](#)
- [CC13xx, CC26xx SimpleLink™ Wireless MCU Technical Reference Manual](#)
- [CC2640 SimpleLink™ Wireless MCU Errata](#)
- [Measuring Bluetooth Smart Power Consumption](#)

Revision History

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

Changes from A Revision (January 2017) to B Revision	Page
------------------------------------------------------	------

- | | |
|------------------------------------------------------|---|
| • Updates were made in Section 2.1 . | 2 |
|------------------------------------------------------|---|

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2017, Texas Instruments Incorporated