Application Report
SWRA499–June 2016

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 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

List of Figures

List of Tables

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 Bluetoooth 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 CC2640 Hardware Revision

In order to use the 32 kHz crystal-less feature, a CC2640/CC2650 device with silicon revision PG2.3 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 (PG2.3 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:

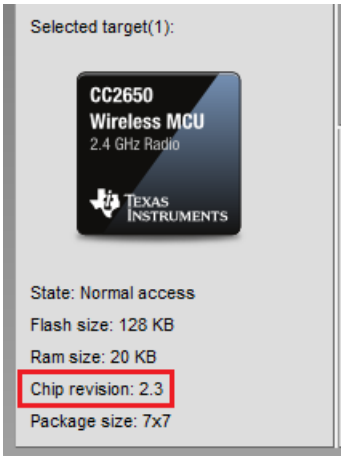

Figure 1. Selected Target

WARNING

Relying on the RCOSC_LF oscillator as a sleep clock on chip 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) SDK versions 2.2 and later. 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 (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 BLE-Stack v2.2 SDK for the IAR and Code Composer Studio™ (CCS) Integrated Development Environments (IDEs).

3.2.1 IAR

1. Select the FlashROM_RCOSC build configuration in the simple_peripheral application project (\examples\cc2650em\simple_peripheral).
2. Exclude ccfg_app_ble.c under Startup folder from build.
3. Make sure RCOSC_LF calibration is enabled in board files.

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

3.2.2 CCS

Select the FlashROM_RCOSC build configuration in the simple_peripheral application project (\examples\cc2650em\simple_peripheral).

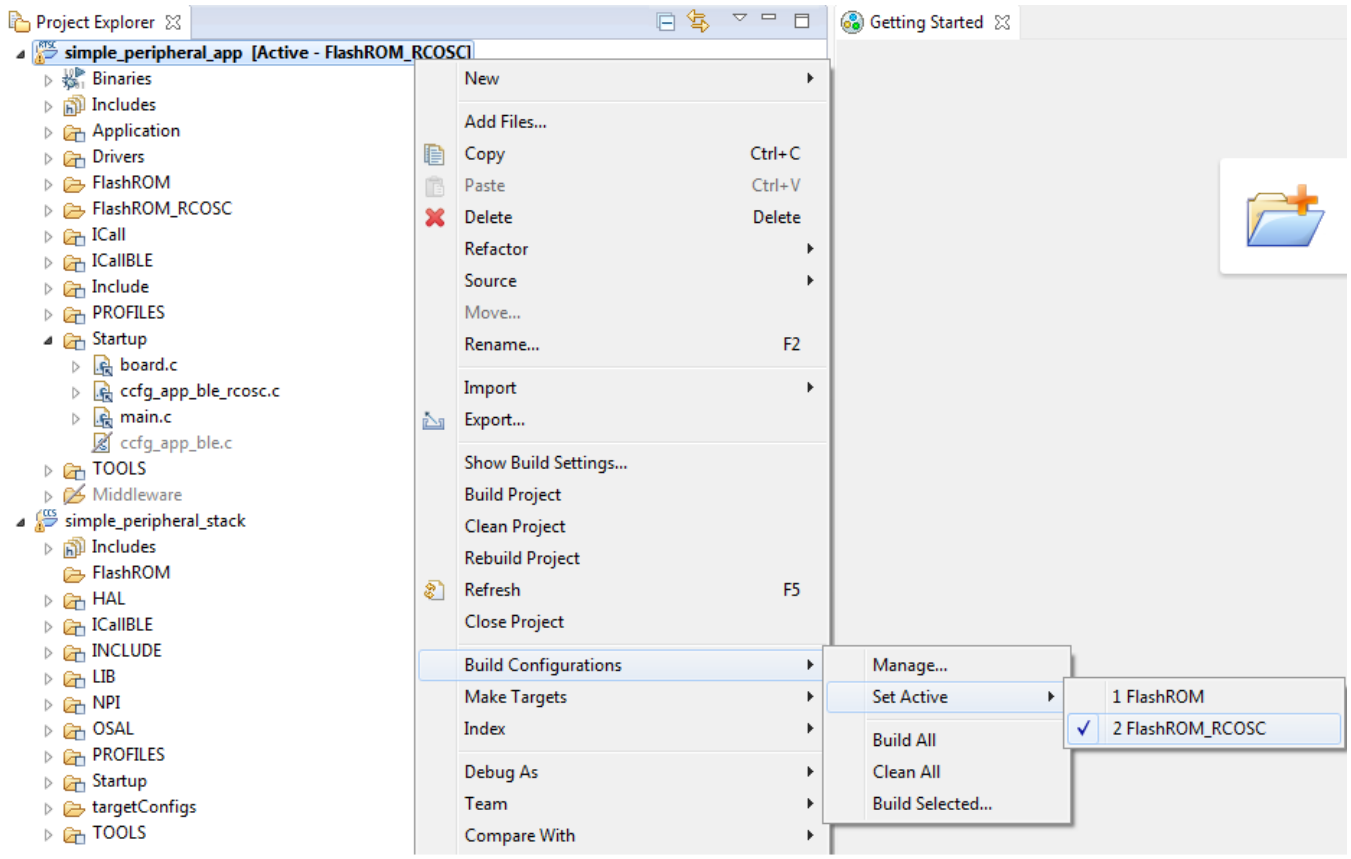

Figure 2. simple_peripheral Application Project

Make sure RCOSC_LF calibration is enable in board files.

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

NOTE: For specific guidance in the actual BLE-Stack software release, see CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack Developer's Guide.

3.2.3 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 FlashROM_RCOSC build configuration 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. If the known temperature variation is larger than 1 deg C/sec, then you can shorten the calibration period by changing RCOSC_CALIBRATION_PERIOD, which can be found in rcosc_calibration.*h*.

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

For example, if the known temperature variation is 2°C/sec, then the calibration interval has to be shorter than 500 ms. Using a shorter calibration period will result in more frequent wakeups from standby (sleep).

In BLE-Stack v2.2, 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_CCFG_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_CCFG_MODE_CONF_SCLK_LF_OPTION: This parameter determines the clock source for low frequency system clock (SCLK_LF).

3.2.4 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 3. 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 [µA]	Average Current w. XOSC [µA]
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

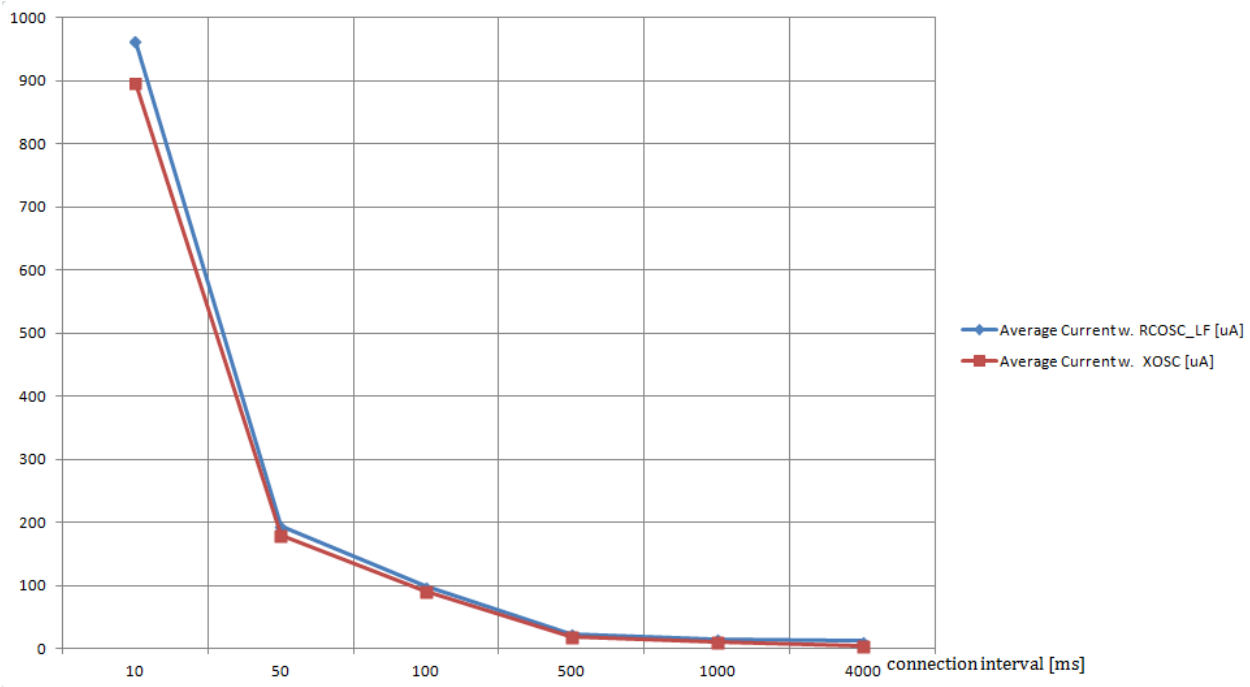

Figure 3. Current Consumption vs Connection Interval

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 2.1.0/2.1.1 Developer's Guide
- CC13xx, CC26xx SimpleLink™ Wireless MCU Technical Reference Manual
- CC2640 SimpleLink™ Wireless MCU Errata
- Measuring Bluetooth Smart Power Consumption

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265