

Application Note

外部 32kHz 水晶振動子を使用しない場合の CC27xx の動作と構成



Cashmere Ramos

概要

CC27xx デバイス ファミリは、内部の低周波発振器 (LFOSC) とハードウェア支援 LFOSC キャリブレーション (LFCAL) を組み合わせた、外付けの 32.768kHz 水晶振動子なしの動作をサポートしています。この機能により、デバイスは Bluetooth Low Energy® (BLE) に完全に準拠しながら、外部コンポーネントに対するシステムレベルの依存度を低減できます。

低消費電力のスタンバイ動作時の LFOSC 周波数ドリフトを補償するため、CC27xx には専用の LFCAL ハードウェア ブロックが内蔵されています。このブロックによって、高周波水晶発振器 (HFXT) は内部発振器 (LFOSC) の自動キャリブレーションを定期的に行うことができようになり、温度や電源電圧が変動しても、デバイスが Bluetooth コア仕様で許容される最大許容ドリフトである $\pm 500\text{ppm}$ のスリープ クロック精度 (SCA) を、CPU の介入を必要とせずに維持できることを検証します。

外部 32.768kHz 水晶振動子が不要なため、部品表 (BOM) コストの削減、PCB レイアウトの簡素化、全体の基板フットプリントの小型化などのシステムレベルでのメリットがあり、Bluetooth Low Energy 仕様のタイミング要件にも準拠できます。

目次

1 概要	3
1.1 該当デバイス	3
1.2 必要なソフトウェア	3
2 クイック選択ガイド: 水晶振動子なしモードを使用する状況	4
2.1 LFOSC の用途	4
2.2 トレードオフ	4
2.3 外部水晶振動子 (LFXT) を使用する場面	4
3 内部発振器動作	5
3.1 動作の仕組み	5
3.2 CC27xx の設計: LFCAL ハードウェア	6
4 クイック スタート ガイド	7
4.1 SysConfig を開く	7
4.2 LFOSC クロック ソースの選択	7
4.3 LFOSC 補償の有効化	7
4.4 補償プロファイルの追加	8
4.5 デフォルト プロファイルとして設定	8
4.6 ビルドとテスト	9
5 構成の詳細と消費電力	10
5.1 補償プロファイルについて	10
5.2 プロファイルの例	11
5.3 電力ポリシー	11
6 ウィンドウの拡大	12
6.1 課題	12
6.2 設計	12
7 ハードウェア設計	13
7.1 取り外せるもの	13

7.2 LFXT ピン.....	13
7.3 HFXT の要件.....	13
7.4 ボードの立ち上げチェックリスト.....	13
8 まとめ.....	14
9 参考資料.....	14

商標

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

すべての商標は、それぞれの所有者に帰属します。

1 概要

このアプリケーション ノートでは、内蔵の低周波発振器 (LFOSC) を使用して、必要なスリープ クロック精度 (SCA) を達成するためにハードウェア支援キャリブレーションを備えて、32kHz の水晶振動子なしモードでテキサス インストルメンツの CC27xx ワイヤレス MCU を動作させる方法について説明します。CC27xx ファミリには専用の LFCAL ハードウェアが含まれており、スタンバイ モード時に効率的な LFOSC キャリブレーションを可能にします。

1.1 該当デバイス

このアプリケーション ノートは、以下を含むすべての CC27xx デバイ스에適用されます。

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

1.2 必要なソフトウェア

- SimpleLink 低消費電力 F3 SDK
- SysConfig

2 クイック選択ガイド: 水晶振動子なしモードを使用する状況

水晶振動子なしの動作を採用するかどうかを評価する際、システム設計ではアプリケーションの電力バジェット、接続間隔の要件、動作温度範囲、PCB 面積の制約を考慮する必要があります。水晶振動子なしモードは、LFOSC キャリブレーションや受信ウィンドウ拡大による追加電力オーバーヘッドがバジェット全体の範囲内で許容できる、コストと省スペース重視の設計に適しています。低消費電力要件が厳格な設計、またはコストとスペースの制約が厳しい設計で 32.768kHz の外部水晶振動子を構成から外す際は、事前にそのトレードオフを注意深く評価する必要があります。

2.1 LFOSC の用途

表 2-1. LFOSC を使用することの利点

優位	詳細
コスト削減	水晶振動子と負荷コンデンサ
PCB 面積の制約	水晶振動子のフットプリントと配線の制約を排除

2.2 トレードオフ

- **平均電流の増加:** キャリブレーションのためのウェークアップが追加されたため
- **RX ウィンドウの拡大:** Bluetooth Low Energy では、水晶発振器使用時の 50ppm (室温時) に対して 500ppm のクロック精度となるため、受信 (RX) ウィンドウが拡大
- **温度感度:** 温度変動率に応じて、キャリブレーションの周波数が上昇

2.3 外部水晶振動子 (LFXT) を使用する場面

表 2-2. LFXT を使用することの利点

使用事例	説明
最小電力にすることが重要	水晶振動子により、キャリブレーションのオーバーヘッドとウィンドウの幅の拡大が不要
最高のタイミング精度が必要	水晶振動子は 50ppm 未満の精度があり、LFOSC のベース クロックよりも高い精度を達成しています

3 内部発振器動作

3.1 動作の仕組み

Bluetooth Low Energy の接続イベントは、デバイス間で信頼性の高いパケット交換を確実に実現するために、高精度のタイミングが必要です。Bluetooth コア仕様では、最大 $\pm 500\text{ppm} + 16\mu\text{s}$ のスリープ クロック精度 (SCA) が許容されます。この SCA では、スタンバイ間隔中の低周波数クロック ソースの最大許容ドリフトが定義されています。

内部 LFOSC はフリーランニング発振器で、デバイスの動作条件によって周波数が増減します。LFOSC 周波数ドリフトを補償するため、システムは周期的なハードウェア キャリブレーション メカニズムを採用しています。LFOSC 周期は、アクティブ モードでの正確なリファレンスとして機能する超低リーク (ULL) クロックに対して測定されます。同じモードで、ULL クロック (CLKULL) が 24MHz で動作します。このモードは、高周波水晶発振器 (HFXT) を基準とする高周波発振器 (HFOSC) から生成されます。デバイスがスタンバイに移行すると、CLKULL は構成された低周波クロック (LFCLK) を使用するように切り替わります。

LFINC と呼ばれる変換係数は、この測定値から得られ、リアルタイムクロック (RTC) によって適用されることで、正確な時間管理を維持します。これにより、Bluetooth Low Energy (LE) 接続イベント スケジューリングが、デバイスの動作寿命を通じて、Bluetooth コア仕様に定義されているスリープ クロックのタイミング精度要件内に維持されていることを確認します。

このキャリブレーション メカニズムでは、高周波水晶発振器 (HFXT) がアクティブになっている必要があります。スタンバイ中、HFXT は電源オフになり、消費電流を最小限に抑え、LFINC の更新を一時停止し、この期間中に発生した LFOSC 周波数ドリフトは補償されないようにします。LFINC 再キャリブレーションは、次の周期的なウェークアップ イベントまたは HFXT を再アクティブにする BLE 動作時に再開されます。

LFCAL なしで LFOSC を使用するシンプルな例

手順 1 — アクティブ モード

- HFXT (48MHz) はアクティブな状態にあり、HFOSC の正確なリファレンスとして機能し、CC27xx デバイスでは 96MHz に、CC23xx デバイスでは 48MHz にトラッキングされた HFOSC を生成します。
- CLKULL は、24MHz を生成するトラッキングされた HFOSC から導出されます。
- ハードウェアは、選択した LFCLK (この場合は LFOSC) 周期を CLKULL に対して測定します。たとえば、最大 -300ppm の偏差については、LFOSC 周波数 = 32.758kHz です。

$$\text{CLKULL} \div \text{LFOSC actual frequency} = \text{CLKULL ticks per LFTICK}$$

$$24 \text{ MHz} \div 32.758 \text{ kHz} = 732.64 \text{ CLKULL ticks per LFTICK}$$
- LFINC は 30.527 μs に更新されます。LFTICK ごとに、ハードウェアが RTC TIME カウンタに LFINC を追加します。1 LFTICK = $1/32758 = 30.527 \mu\text{s}$
- 次の接続イベントをスケジュールするには、COMPARE 値が現在の TIME + 目的の間隔 (1000ms) に設定されます。TIME が COMPARE 値に達すると、RTC はウェークアップを起動します。

手順 2 — スタンバイ モード

- HFXT は無効になっており、HFOSC のトラッキングは行われなくなります。
- CLKULL は選択された LFCLK (この例では LFOSC) で動作するように切り替わりますが、LFOSC を測定するための基準クロックが存在しないため、LFINC は更新できません。
- 前回の測定から LFINC を使用して RTC と WDT はアクティブのままです。
- LFOSC は公称 (32.7844kHz) から +500ppm までドリフトし、約 800ppm の周波数の変化に対応します。LFINC が古い場合、LFOSC ドリフトは補償されません。

$$\text{New actual LFTICK period} = 24 \text{ MHz} \div 32.7844 \text{ kHz} = 30.5023 \mu\text{s per tick}$$
- RTC は LFTICK エッジをカウントし続けますが、各ティックは LFINC が想定する値よりもわずかに短い実時間を表すようになります。ドリフトが補償されない限り、タイミング誤差は累積します。スタンバイ期間内のドリフトが早いほど、ウェークアップ時の誤差は大きくなります。ワーストケースでは、HFXT がディセーブされた直後にドリフトが発生し、スタンバイ時間全体にわたって累積します。

手順 3 — RX ウィンドウのタイミング

1. RTC は 32,758 LFTICK エッジをカウントし、1000.00ms が経過すると予想します。実際の経過時間: $32,758 \text{ ティック} \times 30.5023\mu\text{s} = 999.194\text{ms}$
2. ペリフェラルの推定 RX ウィンドウは $806\mu\text{s}$ 早く開きます
3. 単純化のために中央 SCA を無視する (通常は約 50ppm) と、中央デバイスは $T=1000.00\text{ms}$ で接続パケットを送信します。
4. ペリフェラル SCA が 500ppm で、Bluetooth 仕様の $16\mu\text{s}$ 瞬時ジッタ許容値により、RX ウィンドウは、推定アンカーポイントの前後に $516\mu\text{s}$ 拡張します。
 $RX \text{ window opens: } 999.194 - 0.516 = 998.678 \text{ ms}$
 $RX \text{ window closes: } 999.194 + 0.516 = 999.710 \text{ ms}$

実際のパケットは推定アンカーポイントから $806\mu\text{s}$ 後に到着します。その時点ではウィンドウがすでに閉じてから $290\mu\text{s}$ 経過しています。接続パケットは受信されません。

3.2 CC27xx の設計:LFCAL ハードウェア

CC27xx には、CPU の介入を必要とせずに、Bluetooth コア仕様で許容される $\pm 500\text{ppm}$ 範囲内でスリープクロックの精度を維持するように設計された専用の低周波キャリブレーション (LFCAL) ハードウェアブロックが内蔵されています。スタンバイ中、LFCAL ハードウェアは、以下の動作で構成される定期的なキャリブレーションシーケンスを自律的に実行します。

1. 高周波水晶発振器 (HFXT) が、測定用の CLKULL 基準クロックを供給できるようにします。
2. CLKULL 基準クロックに対する現在の LFOSC 周期を測定します。
3. 更新された LFINC 変換係数を求めて、リアルタイムクロック (RTC) に適用します。
4. HFXT を無効化し、デバイスを低消費電力スタンバイに戻します。

キャリブレーションシーケンス全体は、ハードウェア内で自律的に実行されます。CPU ウェークアップイベントは必要ないため、関連ソフトウェアが不要になります。このメカニズムにより、Bluetooth Low Energy 接続イベントのタイミングが、スタンバイ間隔全体を通して仕様準拠の範囲内に維持されることを検証できます。

上記の簡略化した例をベースとして、LFCAL を使用した同じシナリオを考えてみましょう。1000ms のウェークアップ間隔に補償プロファイルを構成した場合、LFCAL ハードウェアはスタンバイ中に定期的なキャリブレーションを実行し手順 2 と手順 3 の間に追加手順のを挿入します。その結果、手順 3 RX ウィンドウ タイミングは異なる結果をもたらすようになりました。

手順 2a — スタンバイモード時のキャリブレーション

1. スタンバイに入る前に、SW は 1000ms の接続間隔の補償プロファイルを使用して LFCAL ハードウェアを構成しました。この特定の例のみにおいて、結果として得られるキャリブレーション間隔は 150ms と想定されます。実際の値は、すべてのプロファイルパラメータによって異なります (「補償プロファイルについて」を参照)。
2. スタンバイ中、LFCAL は $T = 150, 300, 450, 600, 750\text{ms}$ でウェークアップします。基準に対して LFOSC を測定し、それに応じて LFINC を更新します
3. $T = 750\text{ms}$ で最終的な追加キャリブレーションを実行した後、LFINC は実際の LFOSC 周期と一致します。LFOSC が補償されない残り 250ms のスタンバイ期間中に直ちに 500ppm ドリフトした場合、残留タイミング誤差が生じます

$$500 \text{ ppm} \times 250 \text{ ms} = 125 \mu\text{s}$$

$$\text{Estimated anchor point: } 1000.00 - 0.125 = 999.875 \text{ ms}$$

手順 3 — RX ウィンドウのタイミング

1. RX ウィンドウは、LFCAL なしの 806 μ s と比べて、125 μ s だけ早く開くと見込まれます。
2. 単純化のために中央 SCA を無視する (通常は約 50ppm) と、中央デバイスは T=1000.00ms で接続パケットを送信します。
3. 同じペリフェラル SCA を 500ppm、瞬時ジッタ許容値を 16 μ s と想定すると、RX ウィンドウは約 999.359ms に開き、約 1000.391ms で閉じ、パケットが正常に受信されたことを確認します。

この例で使用するキャリブレーション間隔は、ウェークアップ間隔として 1000ms を使用するすべての補償プロファイルを表すわけではないことに注意してください。詳細については、「[補償プロファイルについて](#)」を参照してください。

4 クイック スタート ガイド

以下の手順では、SysConfig と TI SimpleLink 低消費電力 F3 SDK を使用して、CC27xx プロジェクトで水晶振動子なしの動作を有効にする手順について説明します。すべての構成は SysConfig を使用して行われ、必要なドライバ初期化コードが自動的に生成されます。

4.1 SysConfig を開く

Code Composer Studio (CCS) Theia でターゲット プロジェクトを開き、Project Explorer でプロジェクトの .syscfg ファイルを見つけます。このファイルをダブルクリックして、SysConfig グラフィカル構成ツールを起動します。SysConfig は、レジスタレベルの手動構成なしで、デバイス ペリフェラル、クロック、ドライバ設定を構成するためのハードウェア抽象化インターフェイスを提供します。

4.2 LFOSC クロックソースの選択

アクセス先: 「TI Devices」(TI デバイス) → 「Device Configuration」(デバイス構成) → 「Low Frequency Clock Source」(低周波数クロック ソース)

表示されているオプションから「LF RCOSC」を選択して、内部低周波発振器 (LFOSC) を LFCLK ソースとして指定します。デフォルトで選択されている「LF XOSC」には、外付けの 32.768kHz 水晶振動子が必要です。「LF RCOSC」を選択すると、起動時に PowerLPF3_selectLFOSC() を呼び出すコードが生成され、これによって内部 LFOSC がアクティブになり、これが LFCLK ドメインにルーティングされます。

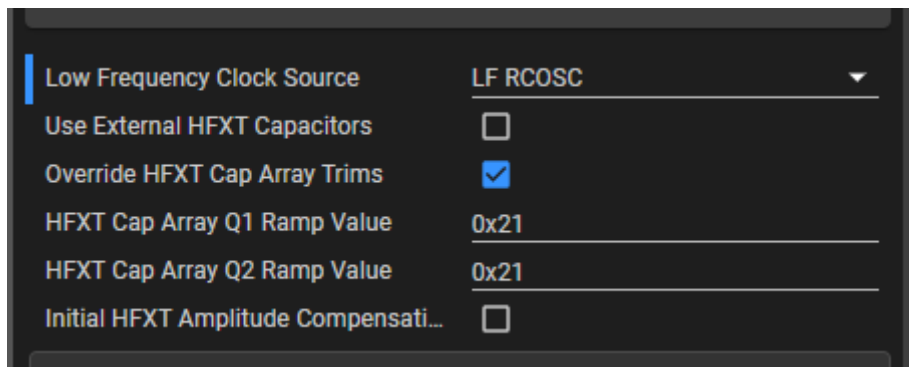


図 4-1. LF クロックソースの選択

4.3 LFOSC 補償の有効化

「Enable LFOSC Compensation」(LFOSC 補償を有効にする) チェックボックスをオンにします。LFOSC が LFCLK のソースである場合、HFXT が利用可能になると常に LFINC 計算が自動的に行われます。このオプションを有効にすると、定期的な補正を実行する追加の LFCAL ウェイクアップがアクティブになり、起動時の初期化中に PowerLPF3_initLfoscCompensation() および PowerLPF3_enableLfoscCompensation() を呼び出すように生成されたコードに指示します。

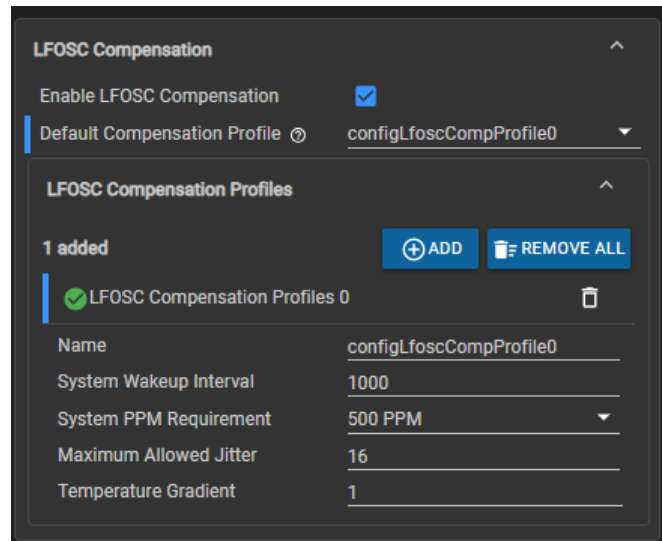


図 4-2. LFOSC 補償チェックボックス

4.4 補償プロファイルの追加

「LFOSC Compensation Profiles」(LFOSC 補償プロファイル) の下の「Add」(追加) をクリックして、新しいプロファイルを作成します。補償プロファイルでは、LFCAL ハードウェアが対象アプリケーションのキャリブレーション間隔を計算するために使用する動作パラメータを定義します。次の表の説明に従って、プロファイルの各フィールドを設定します。

表 4-1. 補償プロファイルの値

パラメータ	標準値	説明
System Wakeup Interval (システム ウェークアップ間隔) (μs)	1000000 (1s)	定義された接続間隔
System PPM Requirement (システム PPM 要件)	500ppm	Bluetooth Low Energy の準拠に必要な
Temperature Gradient (温度勾配)	1°C/s	予想されるワーストケースの温度変化率

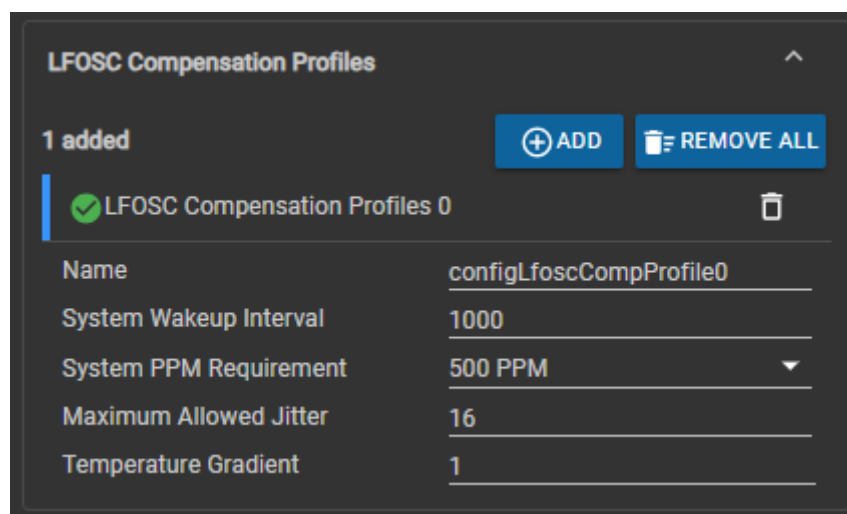


図 4-3. 1 ミリ秒のウェークアップ間隔を持つプロファイルの設定

4.5 デフォルト プロファイルとして設定

構成したプロファイルを「Default Compensation Profile」(デフォルトの補償プロファイル) として選択します。デフォルトのプロファイルは、生成された初期化コードによって起動時に自動的に適用され、BLE スタックが最初の接続イベントを開

始する前に、LFOSC 補償がアクティブであることを確認します。1 つのプロジェクト内で複数の補償プロファイルを定義することができるため、異なる接続間隔やデューティ サイクルなど、さまざまな動作モードを持つアプリケーションに対応できます。ただし、デフォルトとして指定できるプロファイルは 1 つだけです。プロファイルは、`PowerLpf3_setLfoscCompensationProfile()` を使用して実行時に切り替えることができ、アプリケーションの条件の変化に合わせてキャリブレーション間隔を調整できます。

4.6 ビルドとテスト

SysConfig が水晶振動子なしで動作するように構成されると、生成された初期化コードは、以下のパワー ドライバ API 呼び出しのシーケンスを起動して、LFOSC 補償をアクティブにします。

プロジェクトをビルドすると、**SysConfig** で生成された `Board_init()` 関数に以下のコードが自動的に追加されます。

```

/* 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();
  
```

プロジェクトをビルドし、延長スタンバイ間隔中にタイミング エラーがなく BLE 接続イベントが維持されていることを確認して、正しい動作を検証します。

5 構成の詳細と消費電力

5.1 補償プロファイルについて

PowerLPF3_LfoscCompensationProfile 構造は、LFCAL ハードウェアがターゲット アプリケーションのキャリブレーション間隔を計算するために使用する動作パラメータを定義します。

アプリケーションには多くの場合、動作中に異なる状態が発生し、それぞれ異なるウェークアップ間隔とタイミング特性が必要です。たとえば、アプリケーションはウェークアップ間隔が長いアドバタイズ状態で時間を費やし、その後、接続間隔がかなり短いアクティブ接続状態に移行する場合があります。すべてのアプリケーション状態に対して単一のプロファイルを適用することも可能ですが、最も短いウェークアップ間隔、最も厳格な ppm 要件、最も高い温度勾配、許容可能な最小ジッタなど、ワーストケースの条件に合わせて構成する必要があります。キャリブレーション間隔はワーストケース (最短) のウェークアップ間隔によって駆動されるため、LFCAL ハードウェアは、より長いウェークアップ間隔の状態にある間、必要以上に頻繁にキャリブレーションされます。消費電力を最適化するために、複数の補償プロファイルを定義すれば、現在の動作条件に最適なキャリブレーション間隔を LFCAL ハードウェアに適用できます。ユーザーが明示的に割り当てない場合は、デフォルト プロファイルが適用されます。次のフィールドでは、補償プロファイルを設定します。

表 5-1. 補償プロファイルの説明

フィールド	説明	デフォルト値	単位
systemWakeupIntervalUsec	システム ウェークアップ イベントが連続して発生するまでの予想間隔 (マイクロ秒単位)。通常は設定された接続間隔に対応します。	1000000	μs
ppmRequirement	必要なスリープ クロック精度 (SCA) (単位:ppm)。	500	ppm
temperatureGradientMilliCelsiusPerSec	ワーストケースの温度変化率。勾配が大きいほど平均電流は大きくなります	1	C/s
maxAllowedJitterUsec	ppm 要件に加えて許容されるウェークアップごとのタイミング ジッタ最大許容値 (単位:マイクロ秒)。値が小さいほど、平均電流は大きくなります	16	μs

デフォルト値の 16μs は、Bluetooth 仕様で許容される最大瞬時 / ウェークアップ ジッタを反映しています。この仕様では、すべての瞬時タイミングが、スリープ クロック精度 ($\pm 500\text{ppm}$ 以内) から導出された平均タイミングから 16μs を超えてはならないとされています。ドライバはこのバジェットを使用して、キャリブレーション間隔を延長し、平均消費電流を低減します。

LFCAL 補償モデルは、200ppm のベースライン LFOSC 誤差 (LFOSC_BASELINE_PPM) を考慮しています。この値は、プロセス、電圧、温度 (PVT) 変動に対する LFINC フィルタの精度を表しています。このベースラインはスリープ期間とは無関係で、周波数摂動の寄与は含まれていません。キャリブレーション間隔は、温度ドリフト、周波数摂動、このベースライン オフセットを含む合計累積誤差が、Bluetooth コア仕様で許容されるスリープ クロック精度要件の $\pm 500\text{ppm}$ 以内に収まるように計算されます。

LFOSC の温度係数 (ppm/°C) で表され、工場出荷時構成 (FCFG) に保存されます。

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

パワー ドライバは、LFCAL キャリブレーション間隔を計算する際に、CKMDGetLfoscMidTempCoefficientPpmPerC() および CKMDGetLfoscExtTempCoefficientPpmPerC() を介して実行時にこれらの値を自動的に読み取ります。FCFG へのアプリケーション レベルの明示的なアクセスは必要ありません。パワー ドライバは、FCFG で定義された温度係数と、ユーザーが定義する温度勾配を考慮し、要件を維持するために LFCAL ハードウェアがキャリブレーションを行う必頻度を決定します。

5.2 プロファイルの例

以下のプロファイル例は、一般的な Bluetooth Low Energy アプリケーション シナリオの代表的な LFCAL 構成を示しています。各構成について、LFOSC ベースのプロファイルと水晶振動子 (LFXT) リファレンスとの消費電流の比較を示します。以下の例は、LP_EM_CC2745R10-Q1 から取得したものです。表 5-2 は、水晶発振器を使用したときと内部 RC 発振器を使用したときの間隔あたりの平均消費電流を比較したものです。表 5-3 に、設定された各ウェイクアップ間隔における平均消費電流の変化を示します。

なお、間隔が長くなるほどウェイクアップ回数は増えるものの、各キャリブレーション間の時間も長くなります。

固定値:

- 許容ジッタ: 16 μ s
- 温度勾配: 1C/s、
- 温度: 25°C

表 5-2. LFXT と LFOSC の電流測定値の比較

クロック ソース	接続間隔	TX 出力 (dBm)	1 回の間隔の平均電流 (μ A)	追加されたウェイクアップの数
LFXT	30ms	0	222	-
LFOSC	30ms	0	266	1
LFXT	30ms	10	265	-
LFOSC	30ms	10	308.5	1

表 5-3. LFOSC の電流計算値 (各ユーザー構成の比較)

接続間隔 (ms)	予測されるスタンバイ平均電流 (μ A)	1 回の間隔の予測平均電流 (μ A、0dBm、3.3V)	追加されたウェイクアップの数
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 電力ポリシー

スタンバイ モードをイネーブルにするには、電力ドライバ ポリシーを PowerCC27XX_standbyPolicy に構成する必要があります。

SysConfig 内で、以下の場所に移動することで、電力ポリシーを設定できます。「TI Drivers」(TI ドライバ) → 「Power」(電源管理) で、アクティブなポリシー機能として PowerCC27XX_standbyPolicy を選択します。

6 ウィンドウの拡大

Bluetooth Low Energy (BLE) 接続の場合、中央デバイスとペリフェラル デバイスは、接続イベントとして定義された正確な間隔でデータを交換します。各接続イベントのタイミングは、Bluetooth コア仕様で定義されているように、接続セットアップ手順中の接続間隔パラメータによって制御されます。連続する接続イベントの間は、消費電流を最小限に抑えるため、両方のデバイスが低消費電力のスタンバイ状態に移行します。

6.1 課題

デバイスがスタンバイ モードにある間、各デバイスのスリープ クロックはそれぞれのレートで独立してドリフトします。その結果、次の接続イベントを予測して各デバイスがウェークアップすると、中央部と周辺部のどちらも、他方のデバイスの正確なウェークアップ タイミングを確実に判断することができません。各接続イベントでの信頼性の高いパケット受信を検証する際、このクロック不確実性を考慮する必要があります。

6.2 設計

ウィンドウ幅の拡大は、接続されたデバイス間のスリープ クロックの不確実性を補償するために Bluetooth コア仕様で定義されているメカニズムです。ペリフェラルは、累積したクロックドリフトがあっても中央デバイスからのパケットを確実に受信できるよう、想定アンカー ポイントより早めに受信 (RX) ウィンドウを開き、長めの時間維持します。このメカニズムを有効にするために必要な追加構成はありません。Bluetooth Low Energy 接続手順の一部として自動的に適用されます。

受信ウィンドウの期間は、両方のデバイスの合計スリープ クロック精度 (SCA) と、最後の同期アンカー ポイントからの経過時間によって決まります。関係式は次のとおりです。

$$\text{ウィンドウ幅} = 2 \times (\text{SCA_peripheral} + \text{SCA_central}) \times \text{time_since_last_anchor}$$

ここで、SCA 値は 100 万分の 1 (ppm) で表され、time_since_last_anchor は最後に正常に受信されたアンカー ポイントからの経過時間を表します。

LFOSC を $\pm 500\text{ppm}$ で動作させ、 $\pm 20 \sim 40\text{ppm}$ の水晶ベースの発振器と比較し、かつ典型的な中央 SCA を 50ppm と仮定すると、RX ウィンドウの長さは水晶ベース実装の約 6 ~ 8 倍になります。

詳細については、Bluetooth コア仕様の「[ウィンドウの拡大](#)」セクションを参照してください。

7 ハードウェア設計

7.1 取り外せるもの

LFOSC モードを使用する場合、BOM から次の部品を除くことができます。

- 32.768kHz の水晶振動子
- 2 つの負荷コンデンサ (通常は 6.8 ~ 15pF)

7.2 LFXT ピン

LFXT ピンの状態:

- 未接続 (外付け部品不要)
- 他の機能のための GPIO/SPI/UART として使用
- 詳細については、こちらの「[製品ページ](#)」をご覧ください。

7.3 HFXT の要件

HFXT は、キャリブレーション リファレンスおよび RF キャリア周波数基準クロックとして機能するため、48MHz HFXT が常に必要です。

7.4 ボードの立ち上げチェックリスト

1. basic_ble の例から始めます
2. 「[クイック スタート](#)」の説明に従って、SysConfig で LFOSC を構成します
3. ビルドとフラッシュを行います
4. スマートフォンとの Bluetooth Low Energy 接続を確認します
5. 接続の安定性を監視します
6. アプリケーション機能を徐々に追加します

8 まとめ

CC27xx デバイス ファミリーは、外付けの 32.768kHz 水晶振動子を専用のハードウェア キャリブレーション ブロック (LFCAL) と組み合わせて内蔵の低周波発振器 (LFOSC) に置き換えることで、水晶振動子なしの BLE 動作をサポートします。この機能により、水晶振動子や関連する負荷コンデンサが BOM から不要になるため、システムコストの削減、PCB レイアウトの簡素化、基板フットプリントの低減を実現すると同時に、Bluetooth コア仕様の $\pm 500\text{ppm}$ のスリープ クロック 精度 (SCA) 要件への完全な準拠が可能です。

LFCAL ハードウェアは、CPU の介在なしでスタンバイ モード中に自律的に動作し、高周波水晶発振器 (HFXT) を定期的にウェークアップして LFOSC の再キャリブレーションを行い、リアルタイム クロックで使用される LFINC 変換係数を更新します。このように再キャリブレーションを定期的に行うことにより、累積タイミング誤差が抑制され、温度や電源電圧の条件が変化しても、BLE 接続イベントのスケジューリングが仕様範囲内に維持されます。48MHz HFXT は、キャリブレーションのリファレンスと RF キャリア周波数の基準クロックを兼ねているため、基板上で常に利用可能な状態に維持する必要があります。

水晶振動子なしモードには、外部水晶発振子実装と比較した際の、よく知られたトレードオフがあります。追加の LFCAL ウェークアップ イベントにより、平均消費電流が大きくなり、BLE 受信ウィンドウも水晶振動子ベースの発振器使用時より広がります。温度変化率が速いと、キャリブレーションの周波数も高くなるため、補償プロファイルを構成するときは温度勾配が重要なパラメータになります。

構成はすべて SysConfig と SimpleLink 低消費電力 F3 SDK を使用して実行されるため、手動でレジスタにアクセスする必要はありません。PowerLPF3_LfoscCompensationProfile 構造を使用すると、設計者は 1 つまたは複数のプロファイルを定義できます。各プロファイルでウェークアップ間隔、PPM 要件、温度勾配、ジッタ許容値を指定し、アドバタイズメントやアクティブ接続など、さまざまなアプリケーションの状態でのキャリブレーション間隔を実現できます。実行時に PowerLPF3_setLfoscCompensationProfile() を使用して複数のプロファイルを切り替えることができ、動作条件の変化に動的に適応できます。

9 参考資料

1. テキサス・インスツルメンツ、[CC2745R10-Q1](#)、製品ページ
2. Bluetooth、[『Core Specifications 6.2』](#)仕様書。

重要なお知らせと免責事項

TI は、技術データと信頼性データ(データシートを含みます)、設計リソース(リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日 : 2025 年 10 月