

*Application Note***SimpleLink CC2340R5 マイコン上の GPIO を使用してコントラストを制御するセグメント LCD****概要**

7 セグメント液晶ディスプレイ (LCD) は、水道メーターや暖房メーターのような超低消費電力の計量アプリケーションで非常に一般的であり、キッチン家電などのホームオートメーション対応製品のいくつかにも使用されています。通常、これらの LED はマイコン内の専用のオンチップ LCD コントローラ モジュールによって制御されますが、このアプリケーションレポートでは、ソフトウェアのみを使用して LCD を独立に制御するため、抵抗回路を追加する可能性について検討します。このドキュメントで触れている資料は、[SimpleLink CC2340R5 マイコン](#)がこのタスクを達成できるとともに、オンチップのバッテリ モニタを内蔵してリアルタイムのコントラスト制御を実現可能であることを証明します。

このドキュメントで提示するソリューションでは、[TI.com](#) で購入可能な [LP-EM-CC2340R5 EVM](#) と、[SimpleLink 低消費電力 F3 Demos GitHub](#) で無償で提供されているファームウェアを使用します。この例を再現するには、接続ワイヤ、抵抗、および外部から電源を供給される LCD が必要です。動作条件に関する追加情報が記載されているため、読者は特定の LCD 要件に合わせてソースプロジェクトを変更し、アプリケーションをさらに開発できます。

**目次**

<b>1 はじめに</b> .....	<b>2</b>
1.1 CC2340R5.....	2
1.2 7 セグメント LCD.....	2
<b>2 ハードウェア設定</b> .....	<b>4</b>
2.1 LCD 回路図.....	4
2.2 CC2340R5 の接続図.....	4
<b>3 例を実行する</b> .....	<b>6</b>
3.1 依存関係.....	6
3.2 ファームウェアのロード.....	6
<b>4 ファームウェア設計</b> .....	<b>7</b>
4.1 コードの説明.....	7
4.2 タイマー ISR.....	7
4.3 ハードウェア コールバック.....	8
<b>5 テストと結果</b> .....	<b>9</b>
<b>6 まとめ</b> .....	<b>10</b>
<b>7 参考資料</b> .....	<b>11</b>

**商標**

すべての商標は、それぞれの所有者に帰属します。

## 1 はじめに

SimpleLink™ CC2340R5 は、512kB のフラッシュと 36kB または 64kB の SRAM を搭載した、強力で低コストのマイコン (MCU) で、Arm® Cortex®-M0+ と 2.4GHz の無線を搭載しています。これらの機能により、さまざまな無線プロトコルに対応する多数の最終アプリケーションを、單一チップ設計で実現することができます。このアプリケーションレポートでは、1 つのインスタンスを取り上げ、このデバイスによる、さらに広い可能性を証明します。

液晶ディスプレイ (LCD) は、液晶と、オプションのバックライトを使用するディスプレイの一種で、光を変調して個々のピクセルを制御し、画像を生み出すものです。多くの低コストの家庭および産業用アプリケーションでは、非常に消費電力が小さく、コストが低く、小型の 7 セグメント LCD 機能が必要です。これらの LCD ベースのアプリケーションはすべて、LCD を常時オンにすることも、電力を節約するために LCD を一時的にオフにすることもできます。

### 1.1 CC2340R5

CC2340R ファミリは SimpleLink™ MCU プラットフォームの一部であり、このプラットフォームには Wi-Fi®、Bluetooth Low Energy、Thread、Zigbee、Sub-1GHz MCU、ホスト MCU が含まれます。これらはすべて、共通の使いやすい開発環境、單一コアのソフトウェア開発キット (SDK)、および充実したツール セットを共有しています。これらのデバイスは、ビルディング オートメーション (ワイヤレス センサ、照明制御、ビーコン)、アセットトラッキング、医療、リテール EPOS (電子 POS)、ESL (電子棚札)、パーソナル エレクトロニクス (玩具、HID、スタイルス ペン) の市場における低消費電力のワイヤレス通信と OAD (Over the Air Download) サポートに最適化されています。

LP-EM-CC2340R5 開発キットは、Bluetooth 5 Low Energy および 2.4 GHz 専用プロトコルに対応した SimpleLink Bluetooth Low Energy MCU の開発を加速するために使用されます。ソフトウェア サポートは、SimpleLink 低消費電力 F3 ソフトウェア開発キット (SDK) で行われます。この SDK は、無償で提供されている [Code Composer Studio \(CCS\)](#) IDE を使用してビルドできます。BoosterPack™ プラグイン モジュール コネクタによる、すべての I/O 信号へのアクセスや、TI SimpleLink Connect を使用して LaunchPad 開発キットをスマートフォンに接続するなどの機能があります。プログラミング、デバッグ、および RF 評価を行うには、[LP-XDS110ET](#) または LP-XDS110 デバッガ (別売) が必要です。

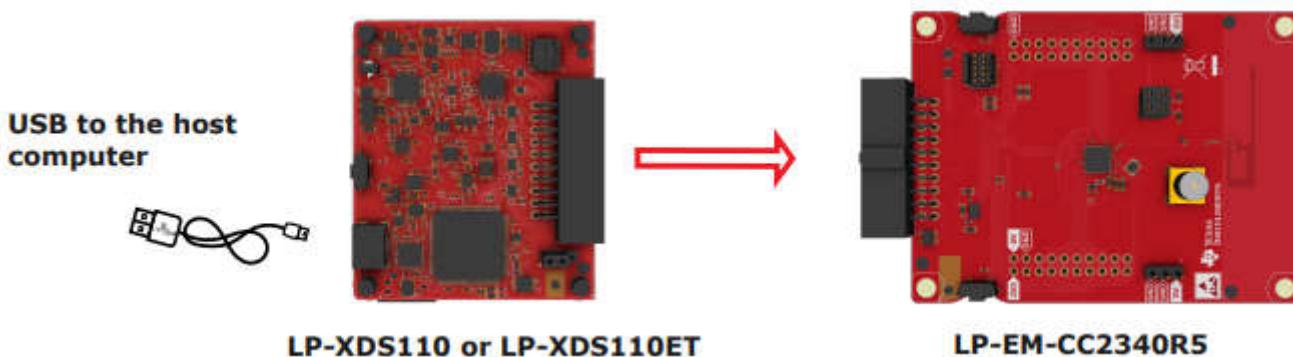


図 1-1. LP-XDS110ET および LP-EM-CC2340R5 の接続

### 1.2 7 セグメント LCD

このドキュメントのデモンストレーションとして、3 つの COM ラインと 7 つのセグメントラインを備えた、1/3 デューティ、1/3 バイアスの LCD が選択されました。そのため、このデバイスでは、2 つの 7 セグメント文字と 7 つの単一セグメント補助シンボルを含む、21 個の個別の LCD セグメントを制御できます。抵抗性ネットワークには 2 本の追加制御ピンが必要で、バックライトの駆動にはさらに 1 本の追加ピンが必要です。したがって、この LCD を制御するための総 GPIO 数は 13 で、さらに LCD セグメントを拡張するには、より多くの GPIO が必要になります。

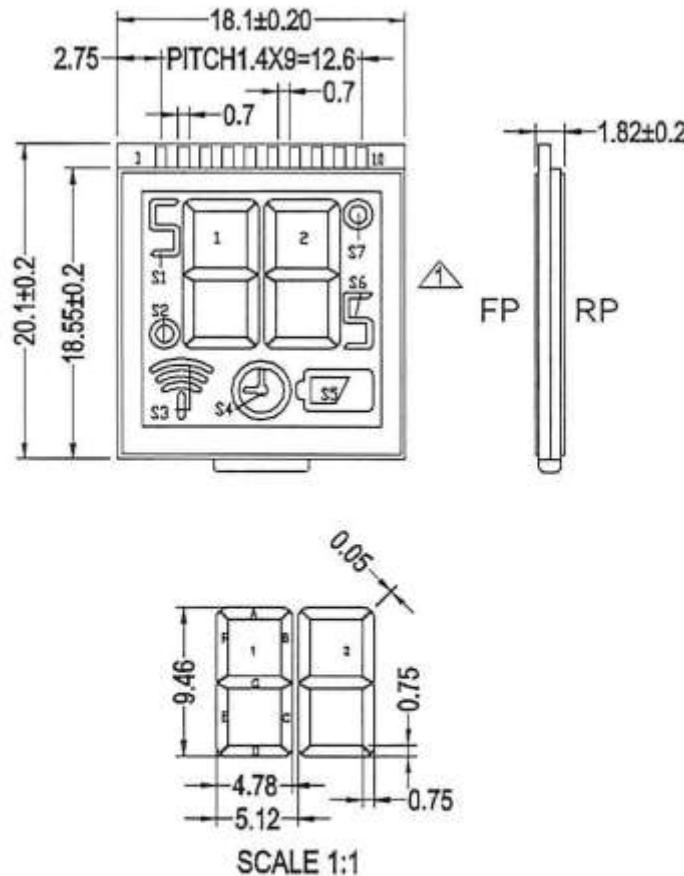


図 1-2. 7 セグメント LCD のピン配列

PIN	1	2	3	4	5	6	7	8	9	10
C <sub>0</sub> M1	1F	1A	1B	2F	2A	2B	S2	C <sub>0</sub> M1	/	/
C <sub>0</sub> M2	1E	1G	1C	2E	2G	2C	S4	/	C <sub>0</sub> M2	/
C <sub>0</sub> M3	S1	1D	S7	S6	2D	S5	S3	/	/	C <sub>0</sub> M3

## 2 ハードウェア設定

以下のセクションでは、抵抗性ネットワークのハードウェアと CC2340R5 GPIO の接続について説明します。これらは、LCD を正しく制御できるように形成する必要があります。

### 2.1 LCD 回路図

次の図は、LCD を駆動するために作成する必要のある抵抗性ネットワークを示しています。これは、ブレッドボード、プロトタイプ、または PCB 設計で作成できます。オプションのバックライトは示されていませんが、LCD 上のセグメントを見るとき非常に役立ちます。乗数（「x3」および「x7」）は、特定の「COM」または「SEG」ピンについて、その回数だけコンポーネントを繰り返す必要があることを表します。

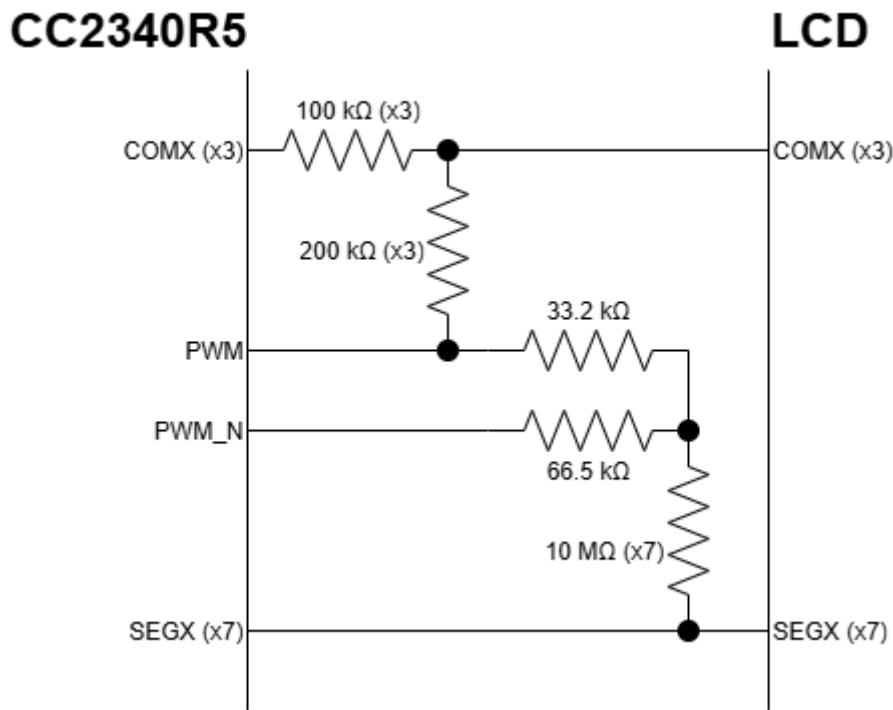


図 2-1. DRV8329A のハードウェア設定

### 2.2 CC2340R5 の接続図

次の表に、CC2340R5 と LCD 回路との接続を示します。各 CC2340R5 DIO はデジタル出力としてのみ動作し、ファームウェアプロジェクトの SysConfig ファイルの GPIO モジュール内から変更できます。

表 2-1. CC2340R5 と LCD との接続

CC2340R5 ピン	LCD 接続	SysConfig 名
DIO8	COM1	CONFIG_GPIO_COM1
DIO19	COM2	CONFIG_GPIO_COM2
DIO21	COM3	CONFIG_GPIO_COM3
DIO23	SEG1	CONFIG_GPIO_SEG1
DIO25	SEG2	CONFIG_GPIO_SEG2
DIO7	SEG3	CONFIG_GPIO_SEG3
DIO14	SEG4	CONFIG_GPIO_SEG4
DIO6	SEG5	CONFIG_GPIO_SEG5
DIO12	SEG6	CONFIG_GPIO_SEG6
DIO5	SEG7	CONFIG_GPIO_SEG7
DIO11	PWM	CONFIG_GPIO_PWM

**表 2-1. CC2340R5 と LCD との接続 (続き)**

CC2340R5 ピン	LCD 接続	SysConfig 名
DIO13	PWM_N	CONFIG_GPIO_PWM_N
DIO15	LED バックライト	CONFIG_GPIO_LED_BACKLIGHT

最終的な結果は次の図のようになります。これらの GPIO は LCD に使用されているため、LED ヘッダー ジャンパは LP-EM-CC2340R5 から取り外されていることに注意してください。XDS110 は、LCD と CC2340R5 の両方に十分な 3.3V の電力を供給できますが、電圧電源範囲全体でコントラストの調整をテストするには、ベンチ電源が必要です。LED と直列に接続された 2 つの抵抗を使用して、単一の CC2340R5 GPIO からバックライトを駆動します。


**図 2-2. 物理ハードウェアのセットアップ**

### 3 例を実行する

以下のセクションでは、LP-EM-CC2340R5 フームウェア プロジェクトの依存関係と、CCS でのプロジェクトのインポート、コードのビルド、CC2340R5 デバイスへのイメージのロードについて説明します。

#### 3.1 依存関係

SimpleLink 低消費電力 F3 デモ GitHub で提供されているコード プロジェクトは、SimpleLink F3 SDK v9.11.0.18、SysConfig v1.23.2、TI CLANG v4.0.3 コンパイラを使用しています。プロジェクトを Code Composer Studio (CCS) v20 以降にインポートする前に、これらの依存関係がすべてマシンにインストールされていることを確認してください。環境設定の詳細な例については、これらの [SimpleLink Academy Lab](#) を参照してください。上記以外の依存バージョンの移行およびサポートは、ユーザーが責任を負うことに注意してください。

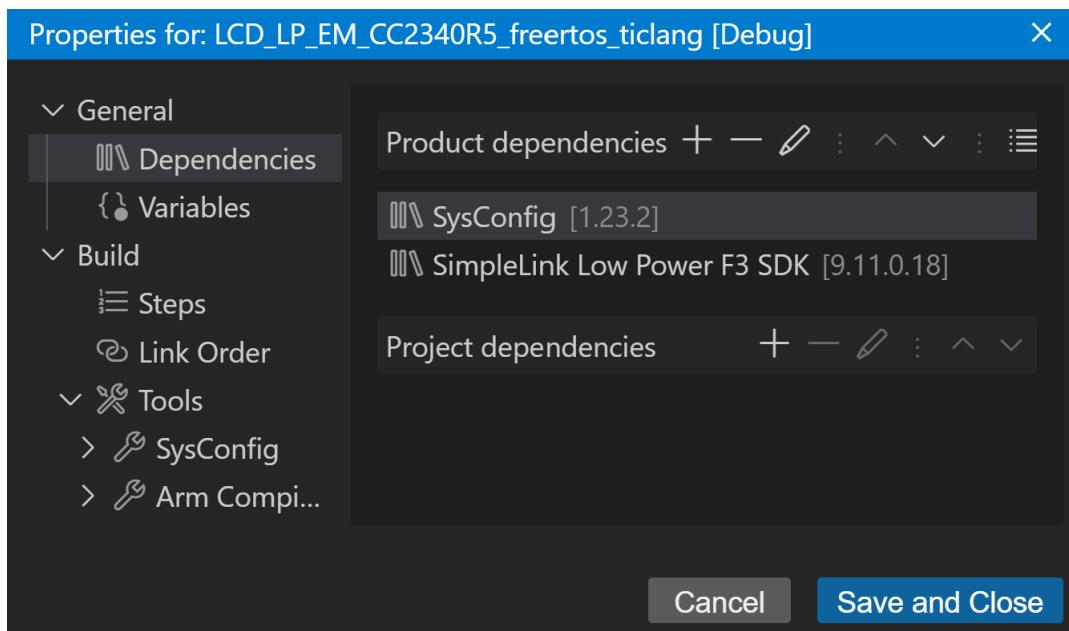


図 3-1. CCS のプロパティ

#### 3.2 フームウェアのロード

CCS 内でビルトされたプロジェクトは、IDE 上で「Run」(実行) → 「Flash Project」(プロジェクトのフラッシュ) (Ctrl + F5) または「Debug Project」(プロジェクトのデバッグ) (F5) を選択することで直接ロードできます。プロジェクトをアクティブにデバッグしていない場合は、デバッグ モードを終了し、フーランニングを許可することを推奨します。バイナリ イメージをロードするために、[Uniflash](#) ソフトウェア ツールの使用も検討してください。

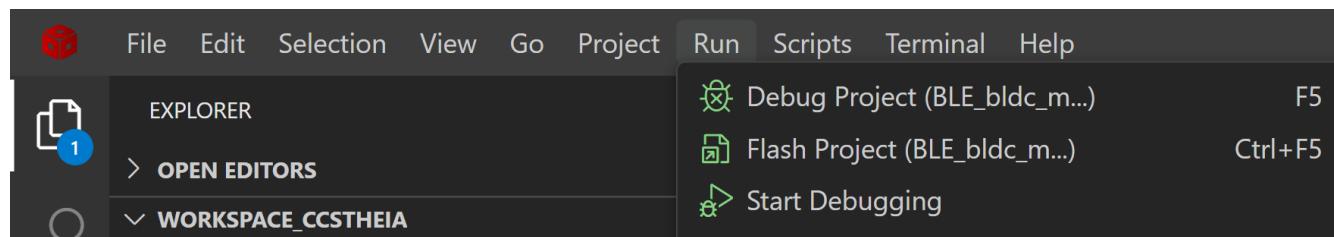


図 3-2. CCS ロード オプション

## 4 フームウェア設計

次にフームウェア自体を、CC2340R5 のオンチップ ペリフェラルと、LCD との相互動作を変更するために使用されるコードの定義に関して、さらに分析します。

### 4.1 コードの説明

サンプル プロジェクトのすべてのコード関数は `lcd.c` ファイルに実装されており、以下に構成可能な定義があります。  
`lrc.syscfg` ファイルは、デバイス構成、TI ドライバ、FREERTOS の設定を構成するためにも重要です。

**表 4-1. LCD アプリケーションの定義**

定義	デフォルト	単位	機能
<code>LIGHT_STRENGTH</code>	2	整数	バックライトの輝度は、0 (0%) から 4 (100%) までの範囲で 25% 刻み
<code>LIGHT_FREQUENCY</code>	1000	Hz	バックライトの周波数
<code>REFRESH_RATE</code>	120	Hz	LCD のリフレッシュ レート
<code>THRESHOLD_*</code>	200	mV	コントラスト制御で使用されるバッテリ監視の電圧レッショルド
<code>NUM_SEGMENTS</code>	7	整数	表示するセグメントの数
<code>NUM_DIGITS</code>	10	整数	表示する桁数
<code>TEMP_UPDATE</code>	10000000	μs	オンチップ温度センサから温度を測定する時間間隔
<code>USE_TEMP</code>	該当なし	ブール値	<code>TEMP_UPDATE</code> ごとに温度を読み取る必要がない場合は削除する

`main` 関数は、オンチップ温度センサ、バッテリ モニタ、GPIO などを含むすべての TI ドライバと、LCD サンプルの動作に必要な RTOS セマフォやタイマを初期化します。その後で、次に示すソフトウェア割り込みおよびハードウェア コールバック サービス ルーチンに追加の演算が付加されます。これらについては、以降のセクションで詳しく説明します。

- `lightUpdateHandler`: LCD のバックライト PWM 制御
- `lcdUpdateHandler`: LCD セグメントとコントラクトの制御
- `tempUpdateHandler`: 温度センサの読み取り値
- `deltaNotificationFxn`: バッテリ モニタのスレッショルドの更新
- `gpioButtonFxn[0/1]`: LaunchPad ボタン

### 4.2 タイマー ISR

これらは、定期的な `ClockP` タイマの満了時に呼び出される、ソフトウェア割り込みサービス ルーチン (ISR) です。

#### 4.2.1 `lightUpdateHandler`

この関数は、`LIGHT_FREQUENCY Hz` の周波数で `LIGHT_STRENGTH` を評価し、`CONFIG_GPIO_LED_BACKLIGHT` を `High` と `Low` のどちらで駆動するかを決定します。この方法は、汎用タイマ (LGPT) パルス幅変調 (PWM) (高周波クロックをオンのままにする必要があるため、スタンバイ低消費電力モードが無効になる) よりも消費電力を低減できます。

#### 4.2.2 `lcdUpdateHandler`

この関数は [TIDA-00848 のリファレンス デザイン](#) から着想を得たもので、8 つの `case` ステートメントを使用して、`REFRESH_RATE Hz` に近い周波数 (コントラストのデッドタイムがあるため、わずかに低くなる) で LCD を駆動します。各 `case` ステートメントにより、`PWM`、`PWM_N`、`COM1`、`COM2`、`COM3` の各ピンは、[『TIDA-00848 デザイン ガイド』](#) に記載されている LCD 波形に従って駆動されます。LCD の COM ラインの数に応じて `case` ステートメントのペアを追加または削除しない限り、これらのピンの `write` コマンドは変更しないことをお勧めします。

`case 0/1` を「デッドタイム」として使用し、CC2340R5 の電源電圧レベルに基づき、割り当てられた期間について各 LCD セグメントを `High` に駆動します。この目的は、2.2 ~ 3.8V の電源電圧範囲にわたって LCD のコントラストを制御することです。低電圧では、電圧が `Low` の間、保証されている LCD セグメントの駆動強度を増やすために、デッドタイムを長くす

る必要があります。しかし、高電圧では、不要な LCD セグメントをオーバードライブしないように、デッドタイムは最小限に抑えられます。

Case 2/3 は COM1 によって駆動されるセグメントを制御し、case 4/5 は COM2 を制御し、case 6/7 は COM3 を制御します。簡単に参照できるよう、表 3 にピン配置図を再度示します。SEGX ピンは、偶数 case (0, 2, 4, 6) で駆動され、奇数 case (1, 3, 5, 7) ではオフになります。

表 4-2. LCD ピン配置図

ピン	1	2	3	4	5	6	7	8	9	10
COM1	1F	1A	1B	2F	2A	2B	S2	COM1	該当なし	該当なし
COM2	1E	1G	1C	2E	2G	2C	S4	該当なし	COM2	該当なし
COM3	S1	1D	S7	S6	2D	S5	S3	該当なし	該当なし	COM3

「1X」セグメントとラベル付けされた LCD セグメントは左の 7 セグメントディスプレイを、「2X」セグメントは右の 7 セグメントディスプレイを制御します。「SX」セグメントは、補助シンボルを指します。コードのデモでは、参照テーブルを使用して、2 桁の 10 進数を、2 つの 7 セグメントディスプレイの駆動に使用される正しいセグメントに変換します。オンチップの温度センサのデモンストレーションとして、°シンボルもオンに駆動されます。参照テーブルは使用される LCD に固有のもので、ピンからセグメントへのマッピング構成が異なる LCD では、それに合わせて変更する必要があります。

```
uint8_t lookupTable[NUM_SEGMENTS][NUM_DIGITS] = {
    //  0  1  2  3  4  5  6  7  8  9 <- digit to write
    {1, 0, 1, 1, 0, 1, 1, 1, 1, 1}, // A  0
    {1, 1, 1, 1, 1, 0, 0, 1, 1, 1}, // B  1
    {1, 1, 0, 1, 1, 1, 1, 1, 1, 1}, // C  2
    {1, 0, 1, 1, 0, 1, 1, 0, 1, 1}, // D  3
    {1, 0, 1, 0, 0, 1, 0, 1, 0, 1}, // E  4
    {1, 0, 0, 1, 1, 1, 0, 1, 1, 1}, // F  5
    {0, 0, 1, 1, 1, 1, 0, 1, 1, 1}, // G  6
};
```

図 4-1. LCD の参照テーブル

#### 4.2.3 tempUpdateHandler

USE\_TEMP が定義されたままの場合、この関数が TEMP\_UPDATE  $\mu$ sごとに呼び出され、オンチップの温度センサを読み取り、その結果を °C 単位で返します。その後で、この値は 2 つの 7 セグメントディスプレイに割り当てられます。

### 4.3 ハードウェア コールバック

これらは、CC2340R5 のハードウェア ペリフェラルのいずれかに変化が起きたときに動作するコールバックです。

#### 4.3.1 deltaNotificationFxn

バッテリ モニタで、CC2340R5 の電圧供給が前回の読み取り値から THRESHOLD\_DELTA\_MILLIVOLT mV だけ増加または減少したことが通知されたとき、この関数を使用して電流電圧レベルを評価し、LCD のコントラスト デッドタイムを調整して、次の通知のために電圧範囲を再登録します。

#### 4.3.2 gpioButtonFxn

LaunchPad のプッシュ ボタンは、LCD に表示されている 2 つの 7 セグメントディスプレイで、2 桁の値をインクリメントするように構成されています。BTN\_LEFT は DIO10 を使用し、gpioButtonFxn0 に割り当てられて、「10 の桁の」7 セグメントをインクリメントし、BTN\_RIGHT は DIO9 を使用し、gpioButtonFxn1 に割り当てられて、「1 の桁の」7 セグメントをインクリメントします。2 桁の 10 進数に期待される方法でロールオーバが実装されています。USE\_TEMP が定義されたままだと、表示は TEMP\_UPDATE  $\mu$ sごとに現在のオンチップ温度 (°C) で更新されることに注意してください。

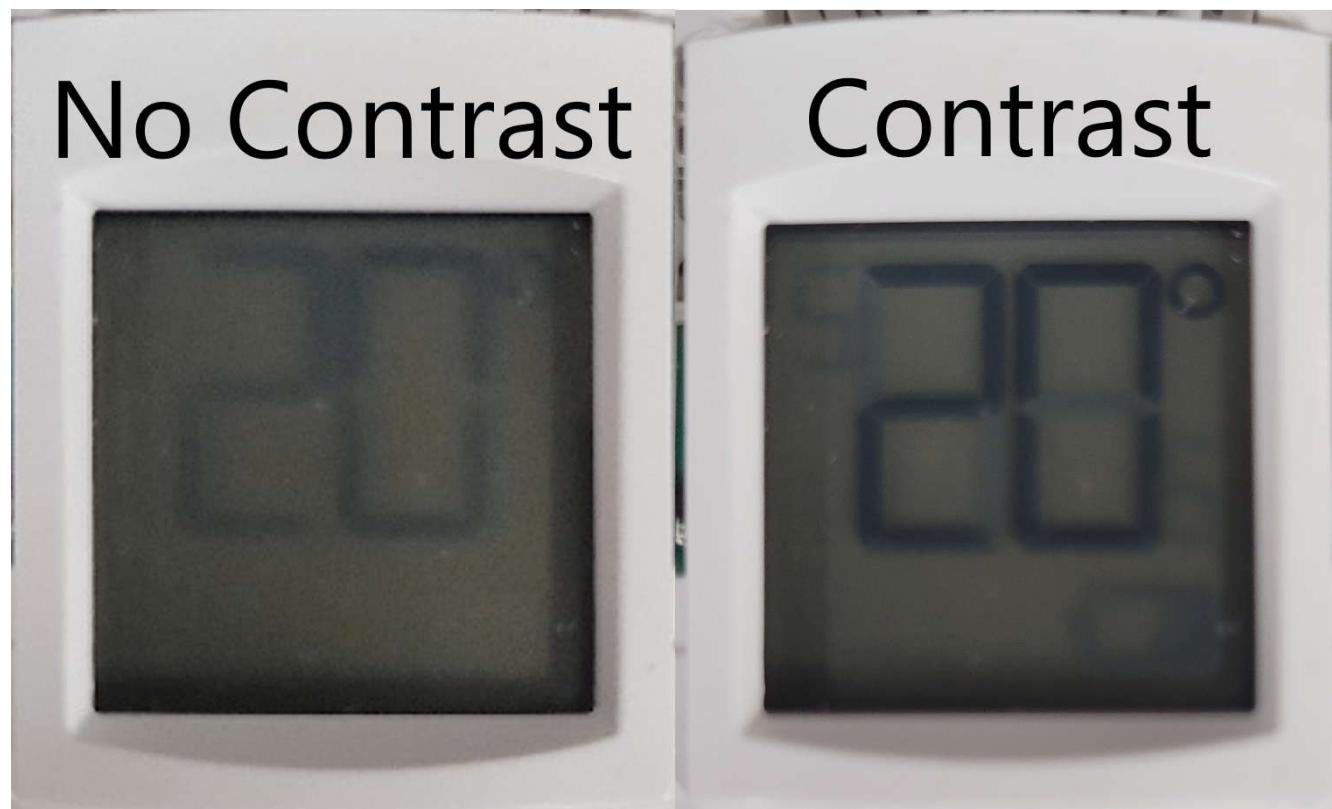
## 5 テストと結果

CC2340R5 LCD ソリューションの消費電力は、電力アナライザ ベンチ ツールを使用して 3V で測定されました。次の表に結果を示します。LCD の構成やバックライトの電源が異なると、電力の結果も異なるため、作業環境を確立した後は、独自の測定を行うことをお勧めします。

**表 5-1. LCD およびバックライトの消費電力**

LCD (60Hz)	バックライト (50%)	CC2340R5 電力 ( $\mu$ A)
オフ	オフ	<1
オン	オフ	3550
オフ	オン	5750
オン	オン	7300

電源ベンチ ツールを使用して、最低の 2.2V を含め、CC2340R5 の許容可能な電源電圧範囲全体でコントラストの調整を観測しました。



**図 5-1. CC2340R5 電圧電源を 2.2V で使用した LCD の可視性**

前と同様に、結果は LCD の選択によって異なり、開発者は `lcd.c` 内のコントラスト設定を調整して目的の結果を得ることができます。

## 6 まとめ

このアプリケーション レポートでは、SimpleLink CC2340R5 を使用するコントラクト調整機能を搭載した、GPIO 駆動の 7 セグメント LCD ソリューションを十分に定義しました。ユーザーがすぐにデモンストレーションを実行できるよう、必要な抵抗ネットワーク、ハードウェア接続、マイコンの動作モードについて説明しました。実装の消費電力とコントラスト出力を確認するためのテスト結果も示されています。ソースコードは GitHub から無償で入手でき、コードフローも詳細に説明されているため、開発者はプロジェクトの動作を理解し、自身の固有のアプリケーション要件に合わせてプロジェクトをさらに変更できます。これらのリソースに関連する追加の質問やサポートが必要な場合は、E2E フォーラムへの投稿を推奨します。

## 7 参考資料

1. テキサス インスツルメンツ、[CC2340R5 データシート](#)
2. テキサス インスツルメンツ、[『LP-EM-CC2340R5 クイック スタート ガイド』](#)
3. テキサス インスツルメンツ、[Code Composer Studio](#)
4. テキサス インスツルメンツ、[SimpleLink Academy Lab](#)
5. テキサス インスツルメンツ、[『TIDA-00848 リファレンス デザイン』](#)
6. テキサス・インスツルメンツ、[SimpleLink 低消費電力 F3 デモの GitHub](#)

## 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默示保証、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、明示的または默示的に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、(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、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ます。

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