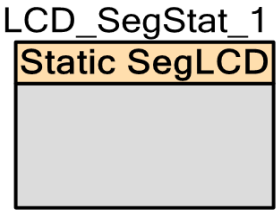
# スタティックセグメント LCD (LCD_SegStat)

2.0




## 特長

- 1 から 61 のピクセルおよびシンボル
- 10- to 150-Hz リフレッシュレート
- ユーザー定義のピクセルおよびシンボルマップ、オプションとして 7 セグメント、14 セグメント、16 セグメント、およびバーグラフ計算ルーチンが付属
- 直接駆動スタティック (1 コモン) LCD

## 概要説明

スタティックセグメント LCD (LCD_SegStat)コンポーネントは、3.3 V および 5.0 V の LCD を直接駆動できます。このコンポーネントは、カスタムもしくは標準的な LCD セグメントディスプレー向けに、PSoC デバイスを設定する、簡単な方法を提供します。

各 LCD ピクセル/ シンボルはオンまたはオフにできます。このスタティック セグメント LCD コンポーネントはまた、ガラス内表示構造の以下のタイプのディスプレーに対しても、簡単に設定できるようになっています。

- 7 セグメント (数字)
- 14 セグメント (英数字)
- 16 セグメント (英数字)
- 1～255 エレメントのバーグラフ

### スタティックセグメント LCD を使用する場合

スタティックモード（コモンラインを一つだけ持つパネルコンフィギュレーション用）にある 3.3Vや 5VのLCDパネルを直接駆動する必要があるときは、スタティックセグメントドライブLCDコンポーネントを使用してください。スタティックセグメントLCDコンポーネントは、ターゲットとなるPSoCデバイスに対してLCDドライブ用ハードウェアリソースを提供することを必要としません。コンポーネントは、コモンおよびセグメントラインに必要な本数に対して十分なピン数を持っているどのターゲットチップでも使用できます。

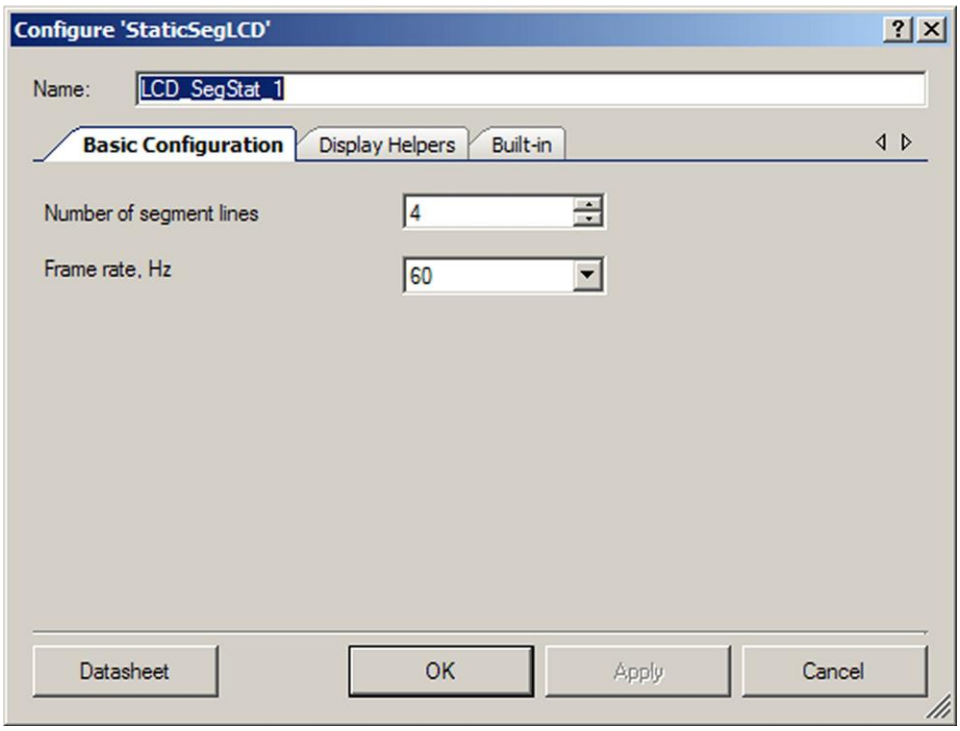
# 入出力接続

回路図上には、コンポーネントの接続は表示されませんが、デザインワイド リソース ピン エディタを使ってさまざまな信号を接続できます。

# コンポーネント・パラメータ

スタティックセグメント LCD コンポーネントをデザイン画面上にドラッグし、ダブルクリックして **Configure** (設定) ダイアログを開きます。

## Basic Configuration (基本設定) タブ



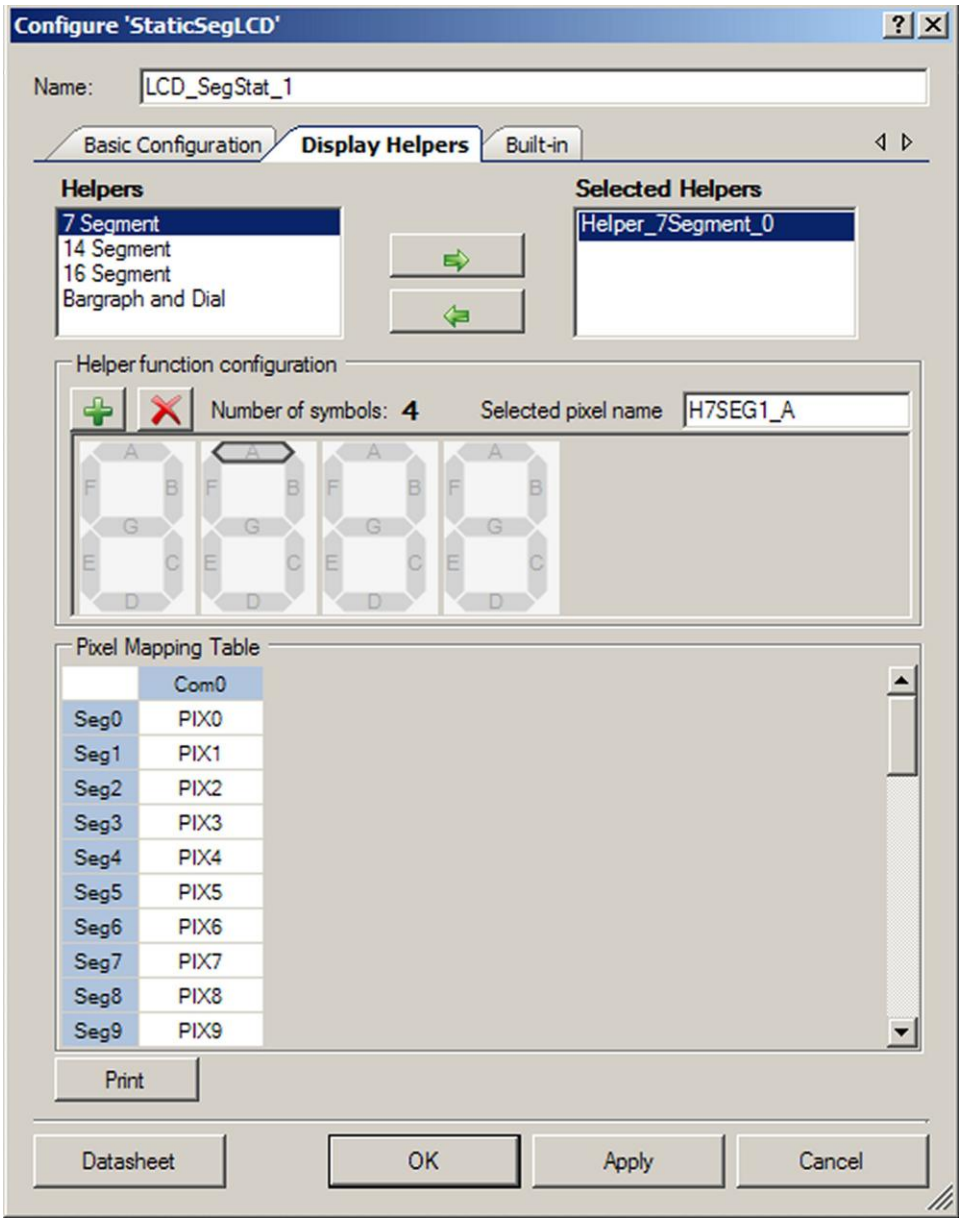

### Number of Segment Lines (セグメントライン数)

ユーザー定義ディスプレイにおけるセグメントライン数を定義します。デフォルトは 4 です。

### フレーム レート、Hz

ディスプレイのリフレッシュレートを決定します。可能な値は 10 Hz から 150 Hz です。デフォルトは 30Hz です。

## Display Helpers (ディスプレイ ヘルパー) タブ




Display Helpers (ディスプレイ ヘルパー) では、いくつかある、あらかじめ定義されたディスプレイ エレメント タイプの 1 つとして一緒に使用するディスプレイ セグメントのグループを設定できます。

- 7、14、または 16 セグメント ディスプレイ
- 直線または円形バーグラフディスプレイ

キャラクタベースのディスプレイ ヘルパーは、複数のディスプレイ シンボルを組み合わせて複数キャラクタ構成のディスプレイ エレメントを作成できます。

### Helpers (ヘルパー) / Selected Helpers (選択したヘルパー)

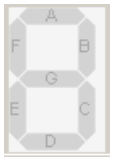
**Helpers** (ヘルパー) リストから使用するヘルパーのタイプを選択し、右向き矢印ボタンをクリックすることで、1 つまたは複数のヘルパーを **Selected Helpers** (選択したヘルパー) リストに追加できます。新しいヘルパーをサポートするのに十分なピンがない場合は、そのヘルパーは追加されません。ヘルパーを削除するには、**Selected Helpers** (選択したヘルパー) リストからそのヘルパーを選択して左向き矢印ボタンをクリックします。

**注**: ディスプレイ ヘルパーをコンポーネントに追加した後は、コモンライン数やセグメントライン数を変更することはできません。ディスプレイ ヘルパーを指定する前にコンポーネントのコモンライン数とセグメントライン数を設定することが重要です。コモンライン数やセグメントライン数を変更する場合は、その前に指定されているディスプレイ ヘルパーを削除する必要があります。

選択したヘルパーがリストに表示される順序は重要です。デフォルトで、**Selected Helper** (選択したヘルパー) リストに加えられた各ヘルパータイプの最初のヘルパーには 0 が後付けされ、そのタイプの次のヘルパーには 1 が後付けされるというように名前が付けられます。選択したヘルパーのどれかが Selected Helper リストから削除されても、残りのヘルパーの名前は変更されません。ヘルパーが追加されると、名称には使用可能な最小のサフィックス(番号)値が使われます。

各ヘルパーに API が提供されます。詳細については、アプリケーション プログラミング インタフェース (アプリケーション プログラミング インターフェイス) のセクションを参照してください。

- **7 Segment Helper (7 セグメント ヘルパー)** – このヘルパーは長さが 1 ～ 5 桁で、16 進数の 0 ～ F または 16 ビットの符号なし 10 進数整数 (uint16) で値を表示します。ヘルパー関数では小数点はサポートされていません。



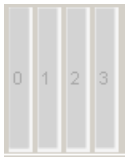

- **14 Segment Helper (14 セグメント ヘルパー)** – このヘルパーの長さは最大 20 文字 です。単一の ASCII 文字を表示することもヌル終端文字列を表示することもできます。指定できる値は標準 ASCII の印刷可能な文字です (コード 0 ～ 127)。




- **16 Segment Helper (16 セグメント ヘルパー)** – このヘルパーの長さは最大 20 文字です。単一の ASCII 文字を表示することも完全なヌル終端文字列を表示することもできます。指定できる値は標準 ASCII 文字と拡張コード表 (コード 0～255) です。拡張コード表は提供されていません。




- **Bargraph and Dial Helper (バーグラフとダイヤル ヘルパー)** – これらのヘルパーは 1 ～ 255 セグメントのバーグラフやダイヤル表示器に使用します。バーグラフは選択した単一ピクセルとしても、選択したピクセルとその右または左側のすべてのピクセルとしても指定できます。




### ヘルパー関数の設定

ダイアログのこのセクションでは、ユーザはシンボルの追加と削除、ピクセルの命名など、ヘルパーの設定を行うことができます。

**Selected Helpers** (選択したヘルパー) リストから設定するヘルパーを選択します。

選択したヘルパーのシンボルを追加するには **[+]** ボタンをクリックし、削除するには **[x]** ボタンをクリックします。追加できるシンボルの最大数はヘルパーのタイプとコンポーネントでサポートされているピクセルの数によって異なります。使用可能なピン数が新しいシンボルをサポートするのに十分でない場合、そのシンボルは追加されません。

ヘルパー関数の一部であるピクセルの名前を変更するには、**ヘルパー関数設定**画面上のそのシンボル画像でそのピクセルを選択します。選択されたピクセルの名前フィールドに現在の名前が表示されます。これを好きな名前に変更します。

### ピクセルの命名

デフォルトのピクセル名は「PIX#」の形式で指定されます。ここで、「#」は**ピクセル マッピング テーブル**の右上端から始まり、1 ずつ増分されるピクセル番号です。

ヘルパー シンボルに関連付けられているピクセルのデフォルト名は異なる形式です。デフォルト名は、1 つのシンボル内のすべてのピクセルに共通のプリフィックス部分と各ピクセルで一意なセグメント識別子から成ります。デフォルト名のプリフィックス部分はヘルパー タイプとシンボル インスタンスを示します。たとえば、7 セグメント ディスプレイ ヘルパー内の 1 つのシンボルの 1 つのピクセルのデフォルト名が「H7SEG4_A」であるとすると、

| | |
|---|---|
| H7 | そのピクセルが 7 セグメント ヘルパーの一部であることを示します |
| SEG4 | そのピクセルがそのプロジェクト内の 4 番目の 7 セグメント シンボルとして指定されているシンボルの一部であることを示します |
| A | 7 セグメント シンボル内の一意なセグメントであることを示します |

デフォルトのピクセル名では、ピクセル名の一意部分のみがシンボル画像に表示されます。ピクセル名を変更した場合、共通の接頭辞があったとしても、ピクセル名全てがシンボル画像に表示されます。

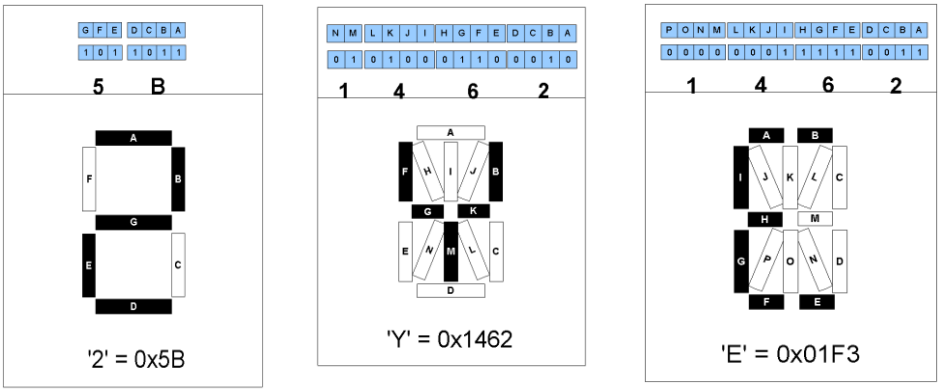
**注** すべてのピクセル名は（重複の無い）一意でなければなりません。

ヘルパー関数シンボル エレメントが**ピクセルマッピング テーブル** (次項で説明) に割り当てられているとき、そのピクセルはそのヘルパー シンボル エレメントの名前を受け継ぎます。ヘルパー シンボル エレメント名はデフォルトピクセル名に優先しますが、それを置き換えるものではありません。ヘルパ関数と関連付けられたピクセルのデフォルト名は、他のピクセルで使用できません。

### 文字の符号化

高レベルヘルパー API はすべて独自のルックアップ テーブルを持ちます。このテーブルには、指定した文字を形成する一連の符号化ピクセルの状態が表示されています。次の例は、ある文字がどのように符号化されるのかを示しています (セグメント名はカスタマイザに示されるものとは異なる場合があります)。

**7 セグメント符号化** **14 セグメント符号化** **16 セグメント符号化**




### ピクセルマッピング テーブル

**ピクセルマッピング テーブル**はフレーム バッファを表示したものです。API 関数が正しく働くためには、**Helper Function Configuration** (ヘルパー関数設定) からの各ピクセルが**ピクセルマッピング テーブル**のピクセル位置に割り当てられている必要があります。正しい割り当てを行うために必要な情報は、LCD グラスのデータシートを参照してください。

ピクセルを割り当てるには、**Helper Function Configuration** (ヘルパー関数設定) パネルでそのピクセルを選択し、**ピクセルマッピング テーブル**内の正しい位置にドラッグします。

**ピクセル マッピング テーブル**にてピクセル名を変更するには、テーブル ディスプレイにてピクセルをダブルクリックし、新しい名前を入力してください。この方法では、使用可能なヘルパー タイプに関連付けられていないピクセルに名を付けることができます。

**Print** (印刷) ボタンをクリックすると、ピクセルマッピング テーブルが印刷されます。

# Clock Select（クロック選択）

スタティックセグメント LCD コンポーネントは、内部クロックを 1 つ使用し、外部クロックを必要としません。LCD コンポーネントが配置されると、クロックは自動的にそのコンポーネント専用になります。クロックは、フレームレート周波数を生成します。

# 配置

デフォルト ピン割り当てはビルド プロセス中に行われ、PSoC Creator Design Wide Resources ツール内のピン エディタを使って変更できます。

# リソース

下の表に、スタティック セグメント LCD コンポーネントに対して指定できるすべての設定を示します。設定名の意味は次の通りです。

**基本**：高レベルのヘルパーAPI のない低レベルの API 関数

**基本、7 セグメント ヘルパー**：低レベルの API 関数群 + 7 セグメント ヘルパー API

**基本、14 セグメント ヘルパー**：低レベルの API 関数群 + 14 セグメント ヘルパー API

**基本、16 セグメント ヘルパー**：低レベルの API 関数群 + 16 セグメント ヘルパー API

**基本、バーグラフ ヘルパー**：低レベルの API 関数群 + バーグラフ ヘルパーの高レベル API

| リソース | リソースのタイプ | | | | | | API メモリ（バイト） | | ピン（外部入出力ごと） |
|---|---|---|---|---|---|---|---|---|---|
| | データパス セル | PLD | ステータス セル | Control/ Count7 セル | 同期 セル | 割り込み | フラッシュ | RAM | |
| 基本 | 0 | 0 | 0 | 0 | 0 | 1 | 1372 | 40 | 2 ~ 62 |
| 基本、7 セグメント ヘルパー | 0 | 0 | 0 | 0 | 0 | 1 | 2373 | 42 | 2 ~ 62 |
| 基本、14 セグメント ヘルパー | 0 | 0 | 0 | 0 | 0 | 1 | 2496 | 42 | 2 ~ 62 |
| 基本、16 セグメント ヘルパー | 0 | 0 | 0 | 0 | 0 | 1 | 2480 | 42 | 2 ~ 62 |
| 基本、バーグラフ ヘルパー | 0 | 0 | 0 | 0 | 0 | 1 | 3268 | 42 | 2 ~ 62 |

# アプリケーション プログラミング インタフェース

アプリケーション・プログラミング・インターフェース (API) ルーチンにより、ソフトウェアを使用してコンポーネントを設定できます。次の表は、各関数へのインターフェースとその説明を示しています。続くセクションでは、各関数について詳しく説明します。

デフォルトで、PSoC Creator は、インスタンス名「LCD_SegStat_1」を、特定のデザインにおける最初のコンポーネント インスタンスに割り当てます。インスタンス名は、識別子の構文ルールに従った独自の値に変更できます。インスタンス名は、すべてのグローバル関数名、変数名、定数名のプリフィックスになります。分かりやすいよう、次の表では、インスタンス名「LCD_SegStat」を使用しています。

| 関数 | 説明 |
|---|---|
| LCD_SegStat_Start() | LCDコンポーネント、および DMAチャネルを起動します。フレームバッファを初期化します。前に定義されていない場合は、フレームバッファ RAM のクリアを行いません。 |
| LCD_SegStat_Stop() | LCD コンポーネント、それに関連した割り込み、DMA チャネルを無効にします。フレーム バッファはクリアされません。 |
| LCD_SegStat_EnableInt() | LCD の割り込みを有効にします。LCD_SegStat_Start が呼び出された場合、不要です。 |
| LCD_SegStat_DisableInt() | LCD の割り込みを無効にします。LCD_SegStat_Stop が呼び出された場合、不要です。 |
| LCD_SegStat_ClearDisplay() | フレームバッファのディスプレイ RAM をクリアします。 |
| LCD_SegStat_WritePixel() | PixelState の値に基づいてピクセルを設定またはクリアします。ピクセルは、パック数値で指定されます。 |
| LCD_SegStat_ReadPixel() | フレーム バッファ内のピクセルの状態を読み取ります。ピクセルは、パック数値で指定されます。 |
| LCD_SegStat_Sleep() | LCD を停止し、ユーザ設定を保存します。 |
| LCD_SegStat_Wakeup() | ユーザコンフィギュレーションを復元し、イネーブルにします。 |
| LCD_SegStat_Init() | 全フレーム割り込みを設定し、フレームバッファを初期化します。 |
| LCD_SegStat_Enable() | コンポーネントのクロック生成を有効化します。 |
| LCD_SegStat_SaveConfig() | LCDコンフィギュレーションを保存します。 |
| LCD_SegStat_RestoreConfig() | LCDコンフィギュレーションを復元します。 |

**注:** 接尾辞「n」を含む関数名は、コンポーネントカスタマイザ中に、同じシンボルタイプで複数の表示ヘルパが生成されたことを示します。各表示ヘルパ要素は、それぞれ「n」インデックス値を関数名中に持つ API 関数によってコントロールされます。

## グローバル変数

| 変数 | 説明 |
|---|---|
| LCD_SegStat_initVar | LCD_SegStatが初期化されたか否かを示します 変数は、0 に初期化され、最初に LCD_SegStat_Start() が呼び出されると 1 に設定されます。これで、LCD_SegStat_Start() ルーチンを最初に呼び出した後で、再初期化を行うことなく、コンポーネントを再起動できます。<br>コンポーネントの初期化が必要な場合、LCD_SegStat_Init()関数をLCD_SegStat_Start()関数またはLCD_SegStat_Enable() 関数の前に呼び出します。 |

## uint8 LCD_SegStat_Start(void)

**説明:** LCDコンポーネント、DMAチャネル、フレームバッファ、およびハードウェアを起動します。フレーム バッファ RAM はクリアされません。

**パラメータ:** なし

**返り値:** uint8 cystatus: 標準 API 戻り値。

| 戻り値: | 説明 |
|---|---|
| CYRET_BAD_PARAM | 関数のパラメータが一つ以上無効です |
| CYRET_SUCCESS | 関数は正しく完了しました |

**副作用:** なし

## void LCD_SegStat_Stop(void)

**説明:** LCD コンポーネント、それに関連した割り込み、DMA チャネルを無効にします。自動的に表示をブランクにし、DC オフセットからの損傷を避けます。フレーム バッファはクリアされません。

**パラメータ:** なし

**返り値:** なし

**副作用:**

## void LCD_SegStat_EnableInt(void)

**説明:** LCD の割り込みを有効にします。LCD_SegStat_Start が呼び出された場合、不要です。LCD の更新 (TD の完了) ごとに割り込みが発生します。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_DisableInt(void)

**説明:** LCD の割り込みを無効にします。LCD_SegStat_Stop が呼び出された場合、不要です。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_ClearDisplay(void)

**説明:** この関数は、ページバッファのディスプレイRAMをクリアします。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## uint8 LCD_SegStat_WritePixel(uint16 PixelNumber, uint8 PixelState)

**説明:** この関数は、PixelStateパラメータに基づいて、フレームバッファ内のピクセルを設定もしくはクリアします。ピクセルは、パック数値で指定されます。

**パラメータ:** uint16 PixelNumber: フレームバッファ内のピクセル位置をポイントするパック番号。LSB の下位ニブル内の下位 3 ビットがそのバイト内のビット位置で、LSB の上位ニブル (4 ビット) はマルチプレクス行内のバイトアドレス、MSB の下位ニブル (4 ビット) はマルチプレクス行番号です。

uint8 PixelState: 指定されたPixelNumberがこのピクセル状態に設定されます。ピクセルステートのシンボリック名が提供されており、対応する値は以下のようになります:

| 値 | 説明 |
|---|---|
| LCD_SegStat_PIXEL_STATE_OFF | ピクセルをオフに設定します。 |
| LCD_SegStat_PIXEL_STATE_ON | ピクセルをオンに設定します。 |
| LCD_SegStat_PIXEL_STATE_INVERT | ピクセルの現在の状態を反転します。 |

**返り値:** uint8 Status: バイトアドレスおよび多重化アドレスの列番号に対する、範囲チェックの戻り値 (pass または fail) です。ビット位置のチェックは行われません。

| 戻り値: | 説明 |
|---|---|
| CYRET_BAD_PARAM | パックされたバイトアドレスまたは列の値が無効です |
| CYRET_SUCCESS | 関数は正しく完了しました |

**副作用:** なし

## uint8 LCD_SegStat_ReadPixel(uint16 PixelNumber)

**説明:** この関数は、フレームバッファにおけるピクセルの状態を読み込みます。ピクセルは、パック数値で指定されます。

**パラメータ:** uint16: PixelNumber: フレームバッファ内のピクセル位置をポイントするパック番号。LSB の下位ニブル内の最下位 3 ビットがそのバイト内のビット位置で、LSB の上位ニブル (4 ビット) はマルチプレクス行内のバイトアドレス、MSB の下位ニブル (4 ビット) はマルチプレクス行番号です。

**返り値:** uint8 PixelState: 指定された PixelNumber の現在の状態を返します。

| 値 | 説明 |
|---|---|
| 0 | ピクセルはオンです。 |
| 1 | ピクセルはオフです。 |

**副作用:** なし

## void LCD_SegStat_Init(void)

**説明:** カスタマイザの [Configure] (設定) ダイアログの設定に従って、コンポーネントを初期化または復元します。LCD_SegStat_Start() ルーチンが LCD_SegStat_Init() 関数を呼び出すので、この関数を呼び出す必要はありません。これはコンポーネントの動作を開始する際に推奨される方法です。すべてのフレーム割り込みを設定し、フレームバッファを初期化します。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_Enable(void)

**説明:** コンポーネントのクロック生成を有効化します。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

**注** 次の 4 つの API は PSoC 5 ではサポートされていません。

## void LCD_SegStat_Sleep(void)

**説明:** これは、コンポーネントのスリープを準備するのに推奨されるルーチンです。LCD_SegStat_Sleep() ルーチンは、現在のコンポーネントの状態を保存します。次に、LCD_SegStat_Stop() 関数を呼び出し、さらに LCD_SegStat_SaveConfig() 関数を呼び出して、ハードウェアの設定を保存します。

CyPmSleep() または CyPmHibernate() 関数を呼び出す前に、LCD_SegStat_Sleep()関数を呼び出してください。電源管理関数については、PSoC Creator *System Reference Guide* を参照してください。

**パラメータ:** なし

**返り値:** なし

**副作用:** コンポーネント ピンのドライブ モードは変更しません。

## void LCD_SegStat_Wakeup(void)

**説明:** これは、コンポーネントをLCD_SegStat_Sleep()が呼び出された時の状態に復元するための、推奨されるルーチンです。LCD_SegStat_Wakeup()関数は、設定を復元するために LCD_SegStat_RestoreConfig()関数を呼び出します。LCD_SegStat_Sleep() 関数が呼び出される前にコンポーネントが有効であった場合は、LCD_SegStat_Wakeup() 関数も、コンポーネントを再度有効にします。

**パラメータ:** なし

**返り値:** なし

**副作用:** 最初に LCD_SegStat_Sleep()または LCD_SegStat_SaveConfig() 関数を呼び出さずに LCD_SegStat_Wakeup() 関数を呼び出すと、予期せぬ動作を引き起こす場合があります。

## void LCD_SegStat_SaveConfig(void)

**説明:** この関数は、コンポーネントの設定と保持されないレジスタを保存します。この関数は、[Configure] ダイアログで定義されている、または該当する API で変更される、現在のコンポーネント・パラメータ値も保存します。この関数は、LCD_SegStat_Sleep()関数によって呼び出されます。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_RestoreConfig(void)

**説明:** この関数は、コンポーネントの設定と非保持レジスタを復元します。この関数はまた、コンポーネントのパラメータ値を LCD_SegStat_Sleep() 関数を呼び出す前の状態に復旧します。

**パラメータ:** なし

**返り値:** なし

**副作用:** 最初に LCD_SegStat_Sleep() または LCD_SegStat_SaveConfig() 関数を呼び出さずに LCD_SegStat_RestoreConfig() 関数を呼び出すと、予期せぬ動作を引き起こす場合があります。

## オプションのヘルパー API

以下の API は、Configure (設定) ダイアログで対応ヘルパーが選択されている場合にのみ表示されます。

| 関数 | 説明 |
|---|---|
| LCD_SegStat_Write7SegDigit_n | 7 セグメント ディスプレイ エレメントのアレイ上に 16 進数を表示します。 |
| LCD_SegStat_Write7SegNumber_n | 7 セグメント ディスプレイ エレメントの 1～5 桁のアレイ上に整数値を表示します。 |
| LCD_SegStat_WriteBargraph_n | 直線、または円形のバーグラフにおける、整数位置を表示します |
| LCD_SegStat_PutChar14Seg_n | 14 セグメント英数字ディスプレイ エレメントのアレイ上に文字を表示します。 |
| LCD_SegStat_WriteString14Seg_n | 14 セグメント英数字ディスプレイ エレメントのアレイ上にヌル終端文字列を表示します。 |
| LCD_SegStat_PutChar16Seg_n | 16 セグメント英数字ディスプレイ エレメントのアレイ上に文字を表示します。 |
| LCD_SegStat_WriteString16Seg_n | 16 セグメント英数字ディスプレイ エレメントのアレイ上にヌル終端文字列を表示します。 |

## void LCD_SegStat_Write7SegDigit_n(uint8 Digit, uint8 Position)

**説明:** この関数は、7 セグメント ディスプレイ エレメントのアレイ上に 16 進数を表示します。桁は、0～9 と A～F の範囲の 16 進値です。7 セグメントのディスプレイ エレメントに関連付けられているピクセル群を定義するには、カスタマイザ ディスプレイ ヘルパー機能を使用する必要があります。フレーム バッファ内で複数の 7 セグメント ディスプレイ エレメントを定義し、関数名内のサフィックス (n) を通してアクセスできます。この関数は、コンポーネント カスタマイザ内で 7 セグメント ディスプレイ エレメントが定義されている場合にのみ含まれます。

**パラメータ:** uint8 Digit: 16進数字として表示される、0 ～ 15 の間の符号なし整数値。

uint8 Position: 右端桁を 0 として右から左に数えた桁位置。位置が定義されている表示領域の外側であれば、表示されません。

**返り値:** なし

**副作用:** なし

## void LCD_SegStat Write7SegNumber_n(uint8 Value, uint8 Position, uint8 Mode)

**説明:** この関数は、7 セグメント ディスプレイ エレメントの 1～5 桁のアレイ上に 16 ビット整数値を表示します。7 セグメント ディスプレイ エレメントに関連付けられているピクセル群を定義するには、カスタマイザ ディスプレイ ヘルパー機能が使用されなくてはなりません。複数の 7 セグメント ディスプレイ エレメント グループをフレーム バッファ内に定義し、関数名内のサフィックスを使ってアクセスすることができます。符号変換、符号表示、小数点、その他のカスタム機能は、アプリケーション固有のユーザ コードで取り扱う必要があります。この関数は、コンポーネント カスタマイザ内で 7 セグメント ディスプレイ エレメントが定義されている場合にのみ含まれます。

**パラメータ:** uint16 Value: 表示する符号なし整数値。

uint8 Position: 右端桁を 0 として右から左に数えた最下位桁位置。定義された表示領域内の桁数が Value で必要な桁数より少ない場合、最上位桁が何桁か表示されないことがあります。

uint8 Mode: ディスプレイ モードを設定します。0または1になります。

| 値 | 説明 |
|---|---|
| 0 | 頭の0を表示しません。 |
| 1 | 頭の0を表示します。 |

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_WriteBargraph_n(uint8 Location, uint8 Mode)

**説明:** この関数は、1 ～ 255 セグメントのバーグラフ(左から右)上の 8 ビット整数位置を表示します。バーグラフは 1 ～255 セグメントの範囲でユーザが定義したサイズとなります。バーグラフは、回転位置を表示するために円形にすることもできます。バーグラフ ディスプレイ エレメントに関連付けられているピクセル群を定義するには、カスタマイザ ディスプレイ ヘルパー機能を使用する必要があります。フレーム バッファ内で複数のバーグラフ ディスプレイ エレメントを定義し、関数名内のサフィックス (n) を通してアクセスできます。この関数は、コンポーネント カスタマイザ内でバーグラフ ディスプレイ エレメントが定義されている場合にのみ含まれます。

**パラメータ:** uint8 Location: 表示する符号なし整数位置。有効な値は、ゼロからバーグラフ内のセグメント数までです。ゼロの値は、すべてのバーグラフ エレメントをオフにします。バーグラフ内のセグメント数を超える値を指定すると、全エレメントがオンになります。

uint8 Mode: バーグラフ ディスプレイ モードを設定します。

| 値 | 説明 |
|---|---|
| 0 | 指定した位置のセグメントがオンになります。 |
| 1 | 指定した位置のセグメントとその左側のすべてのセグメントがオンになります。 |
| –1 | 指定した位置のセグメントとその右側のすべてのセグメントがオンになります。 |
| 2～10 | 指定した位置のセグメントとその右側 2～10 セグメントが表示されます。このモードは幅広なインジケータの作成に使用できます。 |

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_PutChar14Seg_n(uint8 Digit, uint8 Position)

**説明:** この関数は、14 セグメント英数字ディスプレイ エレメントのアレイ上に 8 ビット文字を表示します。14 セグメント ディスプレイ エレメントに関連付けられているピクセル群を定義するには、カスタマイザ ディスプレイ ヘルパー機能が必要です。複数の 14 セグメント ディスプレイ エレメント グループをフレーム バッファ内に定義し、関数名内のサフィックス (n) を使ってアクセスすることができます。この関数は、コンポーネント カスタマイザ内で 14 セグメント ディスプレイ エレメントが定義されている場合にのみ含まれます。

**パラメータ:** uint8 Character: 表示する文字の ASCII 値 (ASCII 値 0 ～ 127 の印刷可能文字)。

uint8 Position: 文字の位置は左端を 0 として左から右に数えます。位置が定義されている表示領域の外側であれば、表示されません。

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_WriteString14Seg_n(*uint8 Character, uint8 Position)

**説明:** この関数は、14 セグメント英数字ディスプレイ エレメントのアレイ上にヌル終端文字列を表示します。14 セグメント表示要素に関連付けられるピクセルセットを定義するためには、カスタマイザの表示ヘルパファシリティを使う必要があります。フレームバッファには複数の 14 セグメント英数字表示要素群を定義することができ、これらの要素はそれぞれ関数名内の接尾辞 (n) を用いて識別することができます。この関数は、コンポーネント カスタマイザ内で 14 セグメント ディスプレイ エレメントが定義されている場合にのみ含まれます。

**パラメータ:** *uint8 Character: ヌル終端文字列を示します。

uint8 Position: 最初の文字の位置は左端を 0 として左から右に数えます。文字列の長さが定義した表示領域のサイズを超える場合は、表示領域に入りきらない文字は表示されません。

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_PutChar16Seg_n(uint8 Digit, uint8 Position)

**説明:** この関数は、16 セグメント英数字ディスプレイ エレメントのアレイ上に 8 ビット文字を表示します。16 セグメント ディスプレイ エレメントに関連付けられているピクセル群の定義のため、カスタマイザ ディスプレイ ヘルパー機能が使用されなくてはなりません。複数の 16 セグメント ディスプレイ エレメント グループをフレーム バッファ内に定義し、関数名内のサフィックス (n) を使ってアドレス設定することができます。この関数は、コンポーネント カスタマイザ内で 16 セグメント ディスプレイ エレメントが定義されている場合にのみ含まれます。

**パラメータ:** uint8 Character: 表示する文字の ASCII 値 (値 0 ～ 255 の印刷可能な ASCII およびテーブル拡張文字) 。

uint8 Position: 文字の位置は左端を 0 として左から右に数えます。位置が定義されている表示領域の外側であれば、表示されません。

**返り値:** なし

**副作用:** なし

## void LCD_SegStat_WriteString16Seg_n(*uint8 Character, uint8 Position)

**説明:** この関数は、16 セグメント英数字ディスプレイ エレメントのアレイ上にヌル終端文字列を表示します。16 セグメント ディスプレイ エレメントに関連付けられているピクセル群を定義するには、カスタマイザ ディスプレイ ヘルパー機能が使用されなくてはなりません。複数の 16 セグメント ディスプレイ エレメント グループをフレーム バッファ内に定義し、関数名内のサフィックス (n) を使ってアドレス設定することができます。この関数は、コンポーネント カスタマイザ内で 16 セグメント ディスプレイ エレメントが定義されている場合にのみ含まれます。

**パラメータ:** *uint8 Character: ヌル終端文字列を示します。

uint8 Position: 最初の文字の位置は左端を 0 として左から右に数えます。文字列の長さが定義した表示領域のサイズを超える場合は、表示領域に入りきらない文字は表示されません。

**返り値:** なし

**副作用:** なし

# ピン API

これらの API 関数は、スタティックセグメント LCD コンポーネントが使用するピンの駆動モードを変更するために使用されます。これらの API は主に、デバイスがローパワーモードのときに、リーク電流を最小化するために、スタティックセグメント LCD コンポーネントのピンを HI-Z モードにするために使用されます。

| 関数 | 説明 |
|---|---|
| LCD_SegStat_ComPort_SetDriveMode | スタティックセグメントLCDコンポーネントのコモン配線が使用するピンの駆動モードを設定します。 |
| LCD_SegStat_SegPort_SetDriveMode | スタティックセグメントLCDコンポーネントのセグメント配線が使用する全てのピンの駆動モードを設定します。 |

## void LCD_SegStat_ComPort_SetDriveMode(uint8 mode)

**説明:** スタティックセグメントLCDコンポーネントのコモン配線が使用するピンの駆動モードを設定します。

**パラメータ:** uint8 mode: 使用するドライブモード。ドライブモードの詳細は、ピンコンポーネントデータシートを参照してください。

**返り値:** なし

**副作用:** なし

### LCD_SegStat_SegPort_SetDriveMode(uint8 mode)

**説明:** スタティックセグメントLCDコンポーネントのセグメント配線が使用する全てのピンの駆動モードを設定します。

**パラメータ:** uint8 mode: 使用するドライブモード。ドライブモードの詳細は、ピンコンポーネントデータシートを参照してください。

**返り値:** なし

**副作用:** なし

## 定義

### LCD_SegStat_SEG_NUM

このマクロは、コンポーネントの現在の設定でのユーザ定義表示内のセグメント ラインの数を定義します。

### LCD_SegStat_FRAME_RATE

このマクロは、コンポーネントの現在の設定でのユーザ定義表示のリフレッシュ レートを定義します。

### LCD_SegStat_WRITE_PIXEL

これは、void を返す LCD_SegStat_WritePixel() 関数のマクロ定義です。

### LCD_SegStat_READ_PIXEL

これは、LCD_SegStat_ReadPixel()関数のマクロ定義です。

### LCD_SegStat_FIND_PIXEL

このマクロはフレーム バッファ内のピクセル位置を計算します。カスタマイザのピクセルテーブルからの情報および LCD 専用の物理ピンに関する情報を用います。ピクセルマッピング機構は、このマクロを基にしています。ピクセルテーブルの各ピクセル名は、フレーム バッファ内の計算されたピクセル位置で定義されます。API はピクセル名を使用して、それぞれのピクセルにアクセスします。

# ファームウェア・ソースコードのサンプル

PSoC Creator は、Find Example Project ダイアログに数多くのサンプルプロジェクトを提供しており、そこには回路図およびコード例が含まれています。コンポーネント固有の例を見るには、[Component Catalog ] または回路図に置いたコンポーネント インスタンスからダイアログを開きます。一般例については、[Start Page] または **[File（ファイル）]** メニューからダイアログを開きます。必要に応じてダイアログにある **Filter Options** を使用し、選択できるプロジェクトのリストを絞り込みます。

詳しくは、PSoC Creator ヘルプの「Find Example Project（サンプルプロジェクトを検索）」を参照してください。

# 機能の説明

スタティックセグメント LCD コンポーネントは、様々なタイプの LCD パネルを駆動するための、強力で柔軟性の高い機構を提供します。設定ダイアログを用いて、コンポーネントをカスタマイズするためのパラメータにアクセスできます。標準 API ルーチン群により、ディスプレイと指定ピクセルを制御します。定義されたディスプレイ ヘルパーのタイプと数に基づいて追加のディスプレイ API が生成されます。

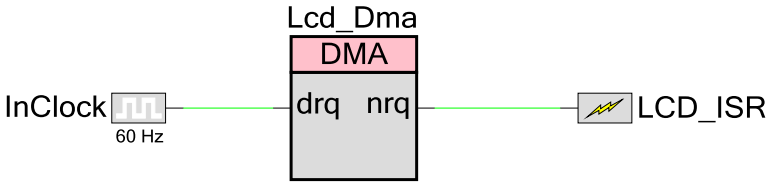
## デフォルト設定

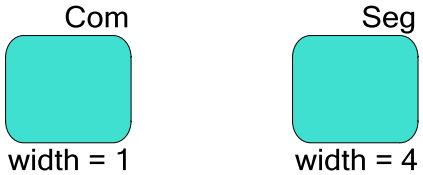
LCD_SegStat コンポーネントのデフォルトコンフィギュレーションは、汎用的なダイレクト LCD セグメントドライブコントローラを提供します。デフォルトの LCD_SegStat の設定は:

- 4 セグメント ライン
- 30 Hz リフレッシュレート
- ディスプレイ ヘルパーは定義されていません。デフォルト API 生成には、サポートされているディスプレイ エレメントの関数は含まれません。

# ブロックダイアグラムと設定

以下の図は、スタティックセグメント LCD コンポーネントの回路図です。コンポーネントは、LCD を駆動するための特別なハードウェアを必要としません。コンポーネントは、DMA および標準デジタルポートの I/O を使用します。




以下の図は、スタティックセグメント LCD のコンポーネントの内部の回路図です。DMA コンポーネント、ISR コンポーネント、クロックコンポーネント、および 2 つの LCD ポートにより構成されます。

- DMA コンポーネントは、フレーム バッファからエイリアス メモリ領域経由でデータを LCD データ レジスタに転送するために使用します。
- 論理信号を物理的なピンにマッピングする際には、LCD ポート (Com および Seg) が使用されます。LCD ポート には 2 つのインスタンスがあります。その 1 つはコモン ラインでもう 1 つはセグメント ラインです。
- ISR コンポーネントが、ユーザー割り込みにおいて使用可能です。

### トップ レベル アーキテクチャ

コンポーネントのアーキテクチャは、とても単純です。いくつかのブロックを使うだけです。コンポーネントのベースとなるのは、LCD データを含むフレームバッファです。フレームバッファの修正は、LCD_SegStat_WritePixel() 関数またはこの関数を参照するハイレベル関数を用いて行うことができます。DMA は、エイリアス メモリを使用して LCD データをポートレジスタに転送します。DMA 操作は、LCD に正しいリフレッシュレートを提供する、あらかじめ計算された数値が設定してある内部クロックによりトリガされます。

## コンポーネント変更履歴

ここでは、過去のバージョンからコンポーネントに加えられた主な変更を示します。

| バージョン | 変更の説明 | 変更の理由 / 影響 |
|---|---|---|
| 2.0 | PSoC 5 シリコンに対するローパワー API サポートは、削除されました。 | 次の API が条件付きコンパイルに追加されました。<br>LCD_SegStat_SaveConfig();<br>LCD_SegStat_RestoreConfig();<br>LCD_SegStat_Sleep();<br>LCD_SegStat_Wakeup()。<br>これらの API は、PSoC 5 デバイスで実行される場合は、プロジェクトに含まれません。 |
| | LCD_SegStat_Stop() API の問題が修正されました。 | LCD_SegStat_Stop() 関数は DMA 設定をリリースしていませんでした。それにより、コンポーネントが、何回かの LCD_SegStat_Start() -> LCD_SegStat_Stop() シーケンスの後で関数の停止を引き起こしました。これは、コンポーネントが LCD_SegStat_Start() 関数に割り当てられた DMA リソースを使い果し、LCD_SegStat_Stop() API にリリースされないためです。正しい DMA リリース手順が実装され、LCD_SegStat_Stop() API で使用されるようになりました。 |

| バージョン | 変更の説明 | 変更の理由 / 影響 |
|---|---|---|
| | WRITE_PIXEL() マクロ定義を、<br>#define LCD_SegStat_WRITE_PIXEL(pixelNumber, pixelState) LCD_SegStat_WritePixel(pixelNumber, pixelState)<br>から<br>#define LCD_SegStat_WRITE_PIXEL(pixelNumber, pixelState) (void) LCD_SegStat_WritePixel(pixelNumber, pixelState) | マクロがそれを処理するように更新されたので、コンポーネント内で LCD_Seg_WritePixel() API の戻り値を解析する必要はありません。 |
| | 次の古い名前がヘッダー ファイルから削除されました:<br>LCD_SegStat_Buffer<br>LCD_SegStat_Channel<br>LCD_SegStat_TermOut<br>LCD_SegStat_TD<br>LCD_SegStat_GCommons<br>LCD_SegStat_Commons | これらの名前は、前のコンポーネント バージョンで廃止の印が付けられました。 |
| | 関数 LCD_SegStat_WriteBargraph() が最適化されました。 | リファクタリングにより、コンポーネントはこの API のフラッシュ メモリのスペースを節約できるようになりました。 |

| バージョン | 変更の説明 | 変更の理由 / 影響 |
|---|---|---|
| | 関数 LCD_SegStat_WriteString14Seg() および LCD_SegStat_WriteString16Seg() が最適化されました。 | 配列のインデクシングがポインタ インデクシングに変更されたことにより、コンポーネントは RAM スペースを節約できるようになり、これらの関数の実行時間が向上します。 |
| | 1 ラインでのソース コード全体に対する複数の変数宣言が修正されました。 | これは、内部コード標準仕様に違反していました。 |
| | LCD_SegStat_Write7SegNumber() API 実装が最適化されました。 | LCD_SegStat_Write7SegNumber() API の一部に無駄なコードがあり、この API の実行時間が長くなっていました。 |
| | LCD_SegStat_Wakeup() API が修正されました。 | LCD_SegStat_Wakeup() API は関数が正しく実行された場合にも、常に CYRET_LOCKED を返しました。正しいリターン ステータス処理の手順が実装されました。 |
| 1.50.a | データシートの LCD_SegStat_Stop() の説明に、手動呼び出しの回避策を追加しました | LCD_SegStat_Stop()ルーチンにて、LCD_SegStat_DmaDispose() への呼び出しが抜けていました。 |
| | データシートに ピン API の節を追加しました。 | |
| | データシートのマイナーな編集と更新を行いました。 | |
| 1.50 | スリープおよび復帰APIを追加しました。 | 低電力モードをサポートするためです。 |
| | コンポーネントの状態を保存する、バックアップ構造を追加しました。 | バックアップ構造は、スリープモードに入る前のコンポーネントの状態(有効もしくは無効)を保存します。コンポーネントをスリープから復帰するためにLCD_SegStat_Wakeup() APIが呼ばれると、この構造を用いて、コンポーネントがスリープモードに入る前の状態に復元します。 |
| | [ヘルパ]タブにツールチップを追加しました。 | [設定(Configure)]ダイアログのデフォルトのサイズでは、ピクセル マッピング テーブルが表示されないため、追加されました。 |
| | 数値のアップ/ダウンコントロール、および有効なセグメントの列数を編集するフィールドを、ユーザーがヘルパ関数を追加しても有効にしました (以前は無効でした) 。 | 以前は、ヘルパを追加した後にセグメントの列数を変更するには、全てのヘルパを削除する必要があったため、全ての設定に関する情報が失われました。ヘルパをセグメントLCDコンポーネントに追加した後で、戻ってセグメント配線の数を変更することができます。セグメント数が増える場合、特に問題ありません。セグメント数が減る場合、「設定の情報を失うかもしれません」という警告が出ます。<br>削除したいセグメント配線にピクセルを割り当ててあった場合、割り当ては解除されます。 |
| | 選択されたピクセルの名前テキストボックスに、エラープロバイダを追加しました。テキストボックスに不正な値を入力した場合、エラーアイコンが表示され、値が正しいものになるまでテキストボックスから抜けることができなくなります。エラーアイコンには、問題の説明が書いてあるツールチップがあります。 | エラープロバイダは、不正な値を扱うための、よりよい方法です。メッセージボックスを表示する代わりに使用されます。 |

| バージョン | 変更の説明 | 変更の理由 / 影響 |
|---|---|---|
| | 以下のグローバル変数が、ローワーキャメルケース名で再定義されました。<br>#define `$LCD_SegStat`_Buffer<br>#define `$LCD_SegStat`_Channel<br>#define `$LCD_SegStat`_TermOut<br>#define `$LCD_SegStat`_TD<br>#define `$$LCD_SegStat`_GCommons<br>#define `$$LCD_SegStat`_Commons | コーディング規約は、変数名がローワーキャメルケース（複合語の先頭文字が小文字で始まる）であることを要求します。これらの変数は今後使用されず、将来的に削除されます。 |
| 1.30.b | シリコン リビジョンとの互換性について知らせる情報をコンポーネントに追加しました。 | このツールは、コンポーネントが不整合のシリコン上で使用された場合にエラー/警告を発します。エラーが表示されたら、対象デバイスをサポートするリビジョンにアップデートしてください。 |
| | v1.30 データシートの変更ログを累積的にしました | |
| 1.30.a | ローカルパラメータを、フォーマルパラメータリストに移動しました。 | PSoC Creator v1.0 Beta 4.1およびそれ以前のバージョンに存在した欠陥を修正するために、コンポーネントを更新し、ツールの新しいバージョンで使用し続けることができるようにしました。このコンポーネントは、ユーザー入力に対してバックグラウンドで行う計算に、ユーザーに不可視のローカルパラメータを使用していました。これらのパラメータは、可視であるが編集できない、フォーマルパラメータに変更されました。コンポーネントの機能に関する変更はありませんが、関連するパラメータは、[カスタマイザ] ダイアログの「式を見る」から見ることができるようになりました。 |
| 1.30 | 生成されたソースコードを更新しました。 | コンポーネントのライブラリに、*StaticSegLCD_INT.c* ファイルを追加しました。<br>コンポーネントのソースコードは、内部割り込みの設定に、*CyLib.h* 関数を使用するように変更されました。<br>DMAが32ビットアドレス空間で正しく動作するように修正しました。<br>コンポーネントのソースファイルに、DMAから正しい終了信号を選択するための式を追加しました。 |
| 1.20.b | シリコン リビジョンとの互換性について知らせる情報をコンポーネントに追加しました。 | このツールは、コンポーネントが不整合のシリコン上で使用された場合にエラー/警告を発します。エラーが表示されたら、対象デバイスをサポートするリビジョンにアップデートしてください。 |
| | データシートの変更ログを累積的にしました | |
| 1.10.b | シリコン リビジョンとの互換性について知らせる情報をコンポーネントに追加しました。 | このツールは、コンポーネントが不整合のシリコン上で使用された場合にエラー/警告を発します。エラーが表示されたら、対象デバイスをサポートするリビジョンにアップデートしてください。 |

| バージョン | 変更の説明 | 変更の理由 / 影響 |
|---|---|---|
| 1.10.a | ローカルパラメータを、フォーマルパラメータリストに移動しました。 | PSoC Creator v1.0 Beta 4.1およびそれ以前のバージョンに存在した欠陥を修正するために、コンポーネントを更新し、ツールの新しいバージョンで使用し続けることができるようにしました。このコンポーネントは、ユーザー入力に対してバックグラウンドで行う計算に、ユーザーに不可視のローカルパラメータを使用していました。これらのパラメータは、可視であるが編集できない、フォーマルパラメータに変更されました。コンポーネントの機能に関する変更はありませんが、関連するパラメータは、[カスタマイザ] ダイアログの「式を見る」から見ることができるようになりました。 |

Copyright © 2005-2012 Cypress Semiconductor Corporation 本文書に記載される情報は、予告なく変更される場合があります。Cypress Semiconductor Corporation は、サイプレス製品に組み込まれた回路以外のいかなる回路を使用することに対しても一切の責任を負いません。特許又はその他の権限下で、ライセンスを譲渡又は暗示することもありません。サイプレス製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、又は安全の用途のために仕様することを保証するものではなく、また使用することを意図したものでもありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことを合理的に予想される、生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

PSoC Designer™及び Programmable System-on-Chip™は、Cypress Semiconductor Corp.の商標、PSoC®は同社の登録商標です。本文書で言及するその他全ての商標又は登録商標は各社の所有物です。

全てのソースコード(ソフトウェア及び/又はファームウェア)は Cypress Semiconductor Corporation (以下「サイプレス」)が所有し、全世界(米国及びその他の国)の特許権保護、米国の著作権法並びに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によるライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであって、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタムソフトウェア及び/又はカスタムファームウェアを作成する目的に限って、サイプレスのソースコードの派生著作物を複製、使用、変更、そして作成するためのライセンス、並びにサイプレスのソースコード及び派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソースコードを複製、変更、変換、コンパイル、又は表示することは全て禁止されます。

免責条項:サイプレスは、明示的又は黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性又は特定目的への適合性の黙示的な保証が含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品又は回路を適用又は使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレスソフトウェアライセンス契約によって制限され、かつ制約される場合があります。