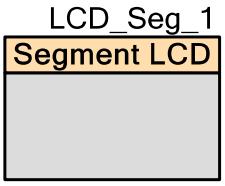
セグメント LCD (LCD_Seg)
2.10



特長

- 2 ～ 768 ピクセルまたはシンボル
- 1/3、1/4、1/5 バイアスをサポート
- 10 ～ 150 Hz のリフレッシュレート
- 動的コントラスト制御用に最大 128 までのデジタル制御バイアス レベルを持つ 2.0 V ～ 5.2 V の統合バイアス生成
- タイプ A (標準) およびタイプ B (ローパワー) の波形をサポート
- 表示のピクセル状態をネガティブ画像用に反転
- 256 バイトの表示メモリ (フレーム バッファ)
- ユーザ定義のピクセルまたはシンボルマップとオプションの 7、14、または 16 セグメント文字、5x7 または 5x8 ドットマトリクス、およびバーグラフ計算ルーチン
- PSoC 3 ES3 シリコン リビジョン用バージョン 2.0 以降を使用してください。

概要説明

セグメント LCD (LCD_Seg) コンポーネントは、最高 16 倍 までの異なる電圧レベルの様々な LCD を直接駆動できます。このコンポーネントは、カスタムまたは標準の LCD を駆動するように PSoC デバイスを設定する容易な方法を提供します。

内部でバイアスを生成するので、外部ハードウェアが必要なく、ソフトウェアベースのコントラストの調整が可能になります。ブースト コンバータを使うと、グラス バイアスを PSoC 供給電圧より高い電圧にすることができます。これにより、ポータブル アプリケーションの表示の柔軟性を向上できます。

各 LCD ピクセル/シンボルはオンまたはオフにできます。セグメント LCD コンポーネントはまた、次のタイプのグラス内表示構造を簡素化するための高度なサポートを提供します。

- 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)

- 5x7 および 5x8 ドットマトリクス 英数字（5x7 と 5x8 で同じルックアップ テーブルを使用します。ルックアップ テーブル内のすべてのシンボルは 5x7 ピクセルサイズです。）
- 1～255 エレメントのバーグラフ

セグメント LCD コンポーネント使用の詳細については、アプリケーション ノート *AN52927: 『LCD ダイレクト ドライブの基本 PSoC 3 および PSoC 5』* を参照してください。

セグメント LCD はどのような場合に使うか

ダイレクト セグメント ドライブ LCD コンポーネントは、最高 16x までのマルチプレクスを、異なる電圧レベルで様々な LCD グラスを直接駆動する必要があるときに使用できます。ダイレクト セグメント ドライブ LCD コンポーネントを使うには、対象となる PSoC デバイスが LCD ダイレクト ドライブをサポートする必要があります。

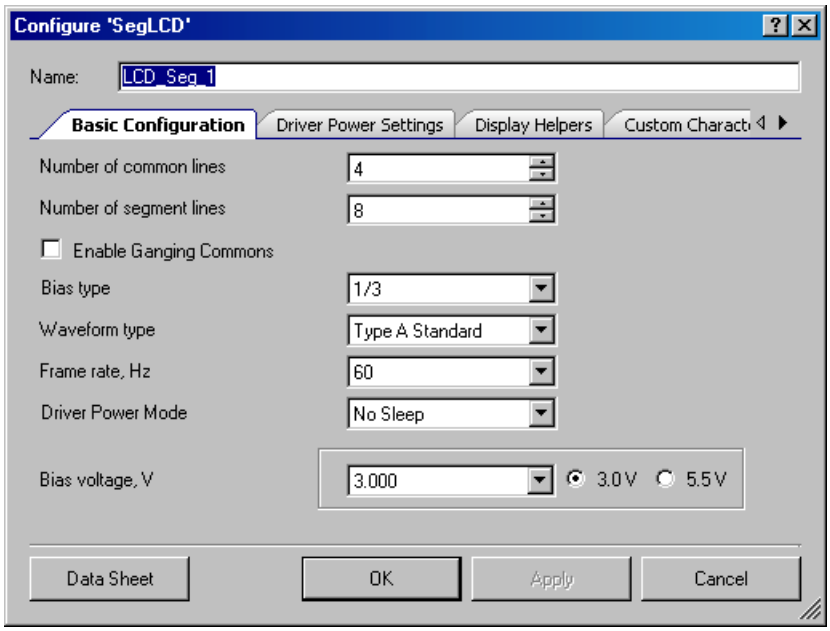
入出力接続

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。

パラメータおよびセットアップ

セグメント LCD コンポーネントを設計上にドラッグし、ダブルクリックして Configure (設定) ダイアログを開きます。Configure (設定) ダイアログには、セグメント LCD コンポーネントの設定に使うパラメータ用のタブがいくつかあります。

Basic Configuration (基本設定) タブ



Number of Common Lines (コモンライン数)

そのディスプレイに必要なコモン信号数を指定します (デフォルトは 4)。

[Number of Segment Lines] (セグメントライン数)

そのディスプレイに必要なコモン信号数を指定します。指定できる値の範囲は 2 ～ 62 です。デフォルトは 8 です。

Enable Ganging Commons (コモンの連結)

PSoC ピンを連結してコモン信号を駆動するには、このチェックボックスを選択します。各コモン信号に 2 つの PSoC ピンが割り当てられます。これは大きいディスプレイを駆動する場合に使用します。

Bias Type (バイアス モード)

コモンとセグメント ラインのセットに適切なバイアス モードを設定します。

Waveform Type (波形タイプ)

波形タイプとしては、Type A - 単一フレームの平均 0 VDC (デフォルト) または Type B - 2 フレームの平均 0 VDC を指定します。

Frame Rate (フレーム レート)

ディスプレイのリフレッシュ レートを設定します。No Sleep (スリープなし) モードでは 10 Hz ～ 150 Hz の範囲で 10 の増分で選択可能です。デフォルトは 60 Hz です。

Low Power (ローパワー) モードでは、Frame Rate (フレーム レート) の選択は制限付きで、各設定ごとに異なります。. 詳細については、このデータシートの「関数の説明」セクションの「ドライバ パワー モード」を参照してください。

Driver Power Mode (ドライバ パワー モード)

Driver Power Mode (ドライバ パワー モード) パラメータは、コンポーネントのパワー モードを指定します。次のパワー モード設定が使用可能です。

- No Sleep (スリープなし) : LCD DAC は常にオンで、チップはスリープ モードになりません
- Low Power using ILO (ILO を 使用したローパワー) : 電圧遷移の間、LCD DAC はオンですが、チップはスリープ モードになります。コンポーネントは 1 KHz の内部 ILO をウェイクアップ ソースとして使用します。
- Low Power using Ext 32 KHz crystal (Ext 32 KHz クリスタル使用のローパワー) : 電圧遷移の間、LCD DAC はオンですが、チップはスリープ モードになります。コンポーネントは OPPS タイマからの 8 K タップをウェイクアップ ソースとして使用します。

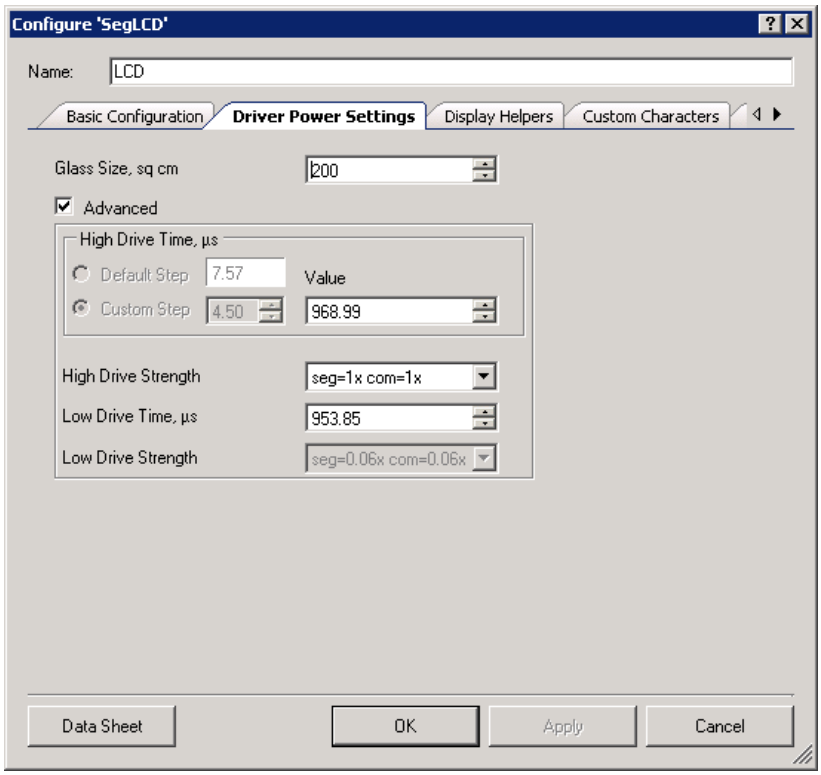
注 使用するローパワー モードに応じて、ILO を 1 KHz に設定して使うか、外部 32 KHz クリスタルを接続し有効にして使用してください。32 KHz を有効にするか、ILO の周波数を Design-Wide Resources Clock Editor で設定します。

このデータシートの「関数の説明」セクションの「ドライバ パワー モード」も参照してください。

Bias Voltage (バイアス電圧)

LCD DAC のバイアス電圧レベルを設定します。指定できる値は供給源によって異なり、 2.35 V ～ 5.5 V または 2.017 V ～ 3 V です。

Driver Power Settings (ドライバ パワー設定) タブ



Glass Size (グラス サイズ)

グラス有効領域のおよそのサイズを平方センチメートルで入力します。この値は、**Number of Common Lines** (コモンライン数) パラメータと共におよその容量性負荷の計算に使用されます。

Advanced (詳細設定)

このチェックボックスを選択すると、LCD の **Driver Power Settings** (ドライバ パワー設定) を手動で変更できます。このチェックボックスを選択しないと、ドライバ パワー設定は **Basic Configuration** (基本設定) と **Glass Size** (グラス サイズ) の選択に基づいて自動的に設定されます。

High Drive Time (ハイ ドライブ時間)

1 回の電圧遷移内でハイ ドライブ モードにする時間を指定します。

Default step (デフォルト増分値)

選択した設定の自動的に計算された High Drive (ハイドライブ) 増分値を指定します。これは、選択すると High Drive time (ハイドライブ時間) の値の増分値となります。

No Sleep (スリープなし) モードで選択できるのは、Default High Drive step (デフォルト ハイドライブ増分値) だけです。

Custom Step (カスタム増分値)

Low Power (ローパワー) モードでのみ使用可能です。ユーザ選択可能な High Drive Step (ハイドライブ増分値) を指定します。Custom High Drive step (カスタム ハイドライブ増分値) として指定可能な最小値はマスタクロックの周波数によって異なり、最大値は **Default step** (デフォルト増分値) によって制限されます。

注 Frame Rate (フレームレート)、 **Number of Common Lines** (コモンライン数)、または **Waveform Type** (波形タイプ) パラメータを変更した場合、**High Drive Time Value** (ハイドライブ時間値) パラメータは自動的にデフォルト値の半分に設定されます (**Advanced** (詳細設定) チェックボックスが選択されている場合のみ)。

*HighDriveTimeValdef= 1.075/DefaultStep * 128 (Type A)*

*HighDriveTimeValdef= 1.075/DefaultStep * 128 (Type B)*

ハイ ドライブ時間の最小値の計算については、このデータシートで後述します。現在の設定でのハイ ドライブ時間の最大値は以下の式で計算されます。

*HighDriveTimeValmax= 1.075/DefaultStep * 253 (Type A)*

*HighDriveTimeValmax= 1.075/DefaultStep * 253 (Type B)*

上記の計算式は、Custom Step (カスタム増分値) 指定の High Drive Time Value (ハイドライブ時間値) にも適用されます。

ローパワー モードのどれかを選択した場合、Default High Drive Step (デフォルト ハイドライブ増分値) を使って **High Drive Time** (ハイドライブ時間) の最大値を設定することは推奨されません。デバイスは短時間スリープモードになるだけで、最悪の場合は全くスリープにならないので、ローパワー モードの使用は効果的ではありません。Custom Step (カスタム増分) を使うと、多くのローパワー アプリケーションの要件であるハイドライブ時間値をより正確に調整できます。

High Drive Strength (高ドライブ強度)

このパラメータを選択すると、High Drive (ハイドライブ) の駆動強度を選択できます。

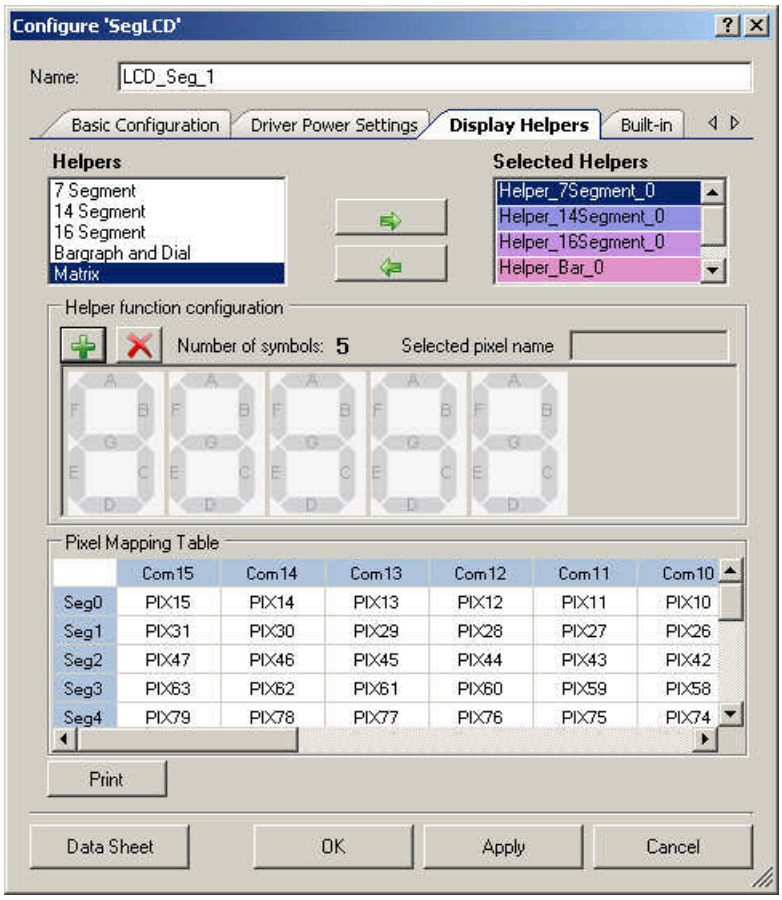
Low DriveTime (ロー ドライブ時間モード)

Low Drive Time (ロードライブ時間) パラメータは、1 回の電圧遷移内でロー ドライブ モードをアクティブにする時間を指定します。

Low Drive Strength（ロードライブ強度）

このパラメータを選択すると、Low Drive (ロードライブ) の駆動強度を選択できます。Low Drive Strength (ロードライブ強度) は High Drive Strength (ハイ ドライブ強度) に相対的で、High Drive (ハイ ドライブ強度) に応じて自動的に設定されます。

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
- ドット マトリクス ディスプレイ (5x7 または 5x8)
- 直線または円形棒グラフ 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)

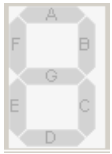
Helpers ( ) 1 つまたは複数のヘルパーを **Selected Helpers** (選択したヘルパー) リストに追加できます。新しいヘルパーをサポートするのに十分なピンがない場合は、そのヘルパーは追加されません。ヘルパーを削除するには、**Selected Helpers** (選択したヘルパー) リストからそのヘルパーを選択して左向き矢印ボタンをクリックします。

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ヘルパー設定情報が失われる可能性があるので、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。コモン ライン数またはセグメント ライン数を変更しようとすると、ヘルパーのピクセルマッピング設定が失われる可能性があるという警告が表示されます。

選択したヘルパーが **Selected Helpers** リストに表示される順序は重要です。デフォルトで、**Selected Helper** 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、最後の番号の次の番号が後付けされます。

各ヘルパーに API が提供されます。詳細については、API 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 進数の桁 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 印刷可能文字です (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。



- **Bar Graph 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と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。



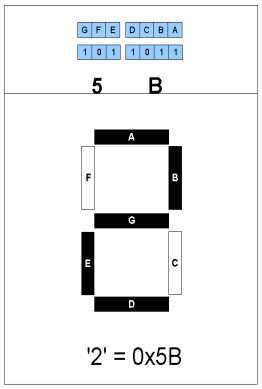
- **Dot Matrix Helper (ドットマトリクス ヘルパー) –** このヘルパーは 8 つのキャラクタ エレメントをサポートします。このコンポーネントは x5x7 または x5x8 行/列文字をサポートします。より長い文字列は、複数のドットマトリクス ヘルパーを設定し、ディスプレイの隣接ドットマトリクス セクションを定義することで作成できます。このヘルパーは単一の ASCII 文字を表示することもヌル終端文字列を表示することもできます。

ドットマトリクス ヘルパーにはピンアウト制約があります。ドットマトリクス ヘルパーは、マトリクスの行用に 7 または 8 個のシーケンシャル コモン ドライバを持ち、マトリクスの列用に 5 ～ 40 個のシーケンシャル セグメント ドライバを持つ必要があります。このコンポーネントは標準 Hitachi HD44780 文字セットをサポートします。

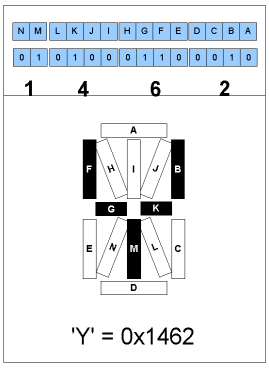
文字の符号化

高レベルのヘルパー API はすべてそれ自身のルックアップ 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、特定の文字が符号化される方法を示しています (セグメント名は [Confiture (設定)] ダイアログに示されるものとは異なる場合があります)。

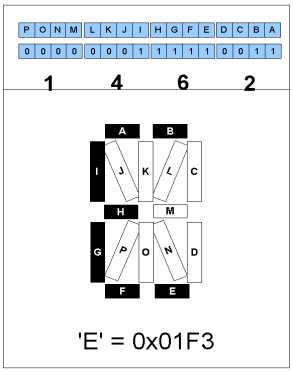
7 セグメント符号化



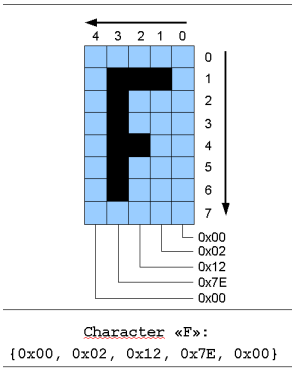
14 セグメント符号化



16 セグメント符号化



ドットマトリクス符号化



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。

1. 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。
2. 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。
3. ヘルパー機能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。現在の名前が選択したピクセル名フィールドに表示され、自由に変更できます。

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。ピクセル名を変更すると、共通プリフィックスを持つ場合にもシンボル画像上に名前全体が表示されます。

注 すべてのピクセル名は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、上書きはしません。ヘルパー関数に関連付けられているピクセルのデフォルトピクセル名は再使用できません。

ピクセルマッピング テーブル

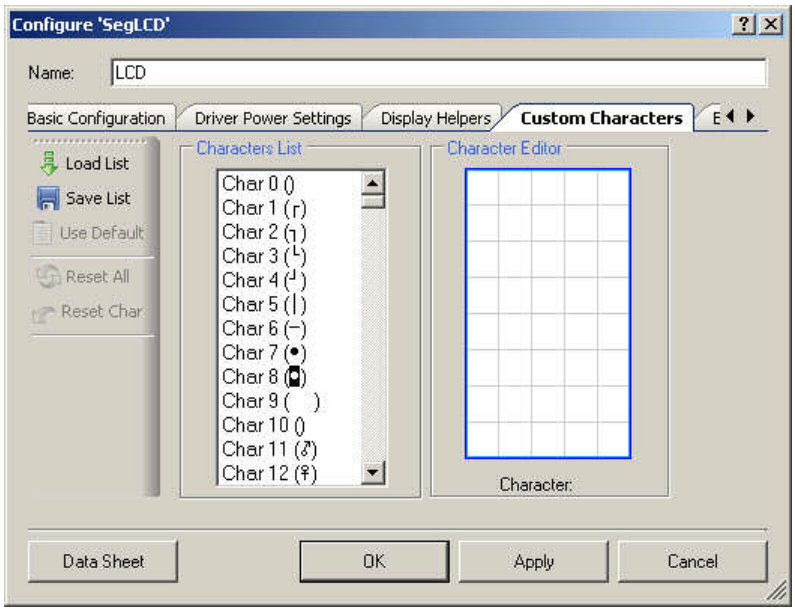
ピクセルマッピング テーブルはフレーム バッファを表示したものです。API 関数が正しく働くためには、Helper Function Configuration (ヘルパー関数設定) からの各ピクセルがピクセルマッピング テーブルのピクセル位置に割り当てられている必要があります。正しい割り当てを行うために必要な情報は、LCD グラスのデータシートを参照してください。

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 テーブル内のピクセルの名前を変更するには、そのピクセルをダブルクリックしてから新しい名前を入力します。この方法は、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名に使用できます。

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。

Custom Characters (カスタム文字) タブ



このタブでは、5x8 ドットマトリクス表示用のカスタム文字を作成できます。カスタム文字のルックアップ テーブルを XML 文字列として保存するために使用することもできます。

Character List (文字リスト) フィールドにはデフォルトで、標準の Hitachi HD44780 文字セットに入っている 255 個の ASCII 文字があります。これらの文字はどれも **Character Editor** (文字エディタ) を使ってアクセスし、変更できます。

Reset Char (文字のリセット) オプションを使うと、未保存に文字をデフォルト設定に戻すことができます。**Reset All** (すべてリセット) オプションを使うと、未保存の文字をすべて標準設定に戻すことができます。

保存した自分の文字セットは、 **Save List** (リストの保存) コマンドを用いて XML 文字列として保存できます。**Load List** (リストのロード) コマンドを使うと、XML 文字列から自分の文字リストをロードできます。**Use Default** (デフォルトの使用) オプションを使って、標準文字セットに戻すこともできます。

Clock Select (クロック選択)

LCD_Seg コンポーネントは 1 つの内部クロックを使用するので、外部クロックは必要ありません。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。周波数は、コモンライン数、リフレッシュレート、および波形タイプに基づいて自動的に計算されます。

配置

LCD_Seg コンポーネントの実装は 2 つの部分から成ります。LCDDAC は、このコンポーネントで使用される PSoC 内の固定機能ハードウェア ブロックです。UDB には、ドライブ信号用の追加のタイミング ロジックが実装されます。UDB リソースは、プロジェクト生成プロセス中に自動的に UCB アレイ内に配置されます。

注: 1 つのプロジェクトで使用できるコンポーネントのインスタンスは 1 つだけです。プロジェクト内で複数のコンポーネントのインスタンスが使われると、ビルド プロセス中に配置エラーとなります。

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：高レベルのﾍﾙﾊﾟｰ 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 ﾍﾙﾊﾟｰ 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 ﾍﾙﾊﾟｰ 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 ﾍﾙﾊﾟｰ API

基本、ドットマトリックス ヘルパー：低レベルの API 関数群 + ドットマトリクス ﾍﾙﾊﾟｰの高レベル 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 ﾍﾙﾊﾟｰの高レベル 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	LCD 固定ブロック	Control/ Count7 セル	同期セル	割り込み	Flash	RAM	
基本	1	2	1	1	0	1	4625	409	3～62
基本、 7 セグメント ヘルパー	1	2	1	1	0	1	5182	419	3～62
基本、 14 セグメント ヘルパー	1	2	1	1	0	1	5728	419	3～62

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	LCD 固定ブロック	Control/ Count7 セル	同期セル	割り込み	Flash	RAM	
基本、 16 セグメント ヘルパー	1	2	1	1	0	1	5860	419	3〜62
基本、ドットマトリックス ヘルパー	1	2	1	1	0	1	6924	420	3〜62
基本、バーグラフ ヘルパー	1	2	1	1	0	1	6772	419	3〜62

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。その次のセクションでは、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_Seg_1」を、特定の設計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。そのインスタンス名は、 PSoC Creator の識別子の構文ルールに従う任意の一意名に変更できます。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。便宜上、次の表ではインスタンス名として「LCD_Seg」を使っています。

関数	説明
LCD_Seg_Start	initVar 変数を設定し、Init 関数を呼び出して、Enable 関数を呼び出します。
LCD_Seg_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。
LCD_Seg_EnableInt	LCD の割り込みを有効にします。LCD_Seg_Start が呼び出された場合は必要ありません。
LCD_Seg_DisableInt	LCD の割り込みを無効にします。LCD_Seg_Stop が呼び出された場合は必要ありません。
LCD_Seg_SetBias	LCD グラスのバイアス レベルを 128 ある値のどれかに設定します。
LCD_Seg_WriteInvertState	入力パラメータに基づいて表示を反転します。
LCD_Seg_ReadInvertState	標準か反転かという表示反転状態の現在値を返します。

関数	説明
LCD_Seg_ClearDisplay	表示とその関連フレーム バッファ RAM をクリアします。
LCD_Seg_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。
LCD_Seg_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。
LCD_Seg_Sleep	LCD を停止し、ユーザ設定を保存します。
LCD_Seg_Wakeup	ユーザ設定を復元し、LCDを有効にします。
LCD_Seg_SaveConfig	LCD 設定を保存します。
LCD_Seg_RestoreConfig	LCD 設定を復元します。
LCD_Seg_Init	Configure (設定) ダイアログでの設定に従って LCD を初期化または復元します。
LCD_Seg_Enable	LCD を有効にします。

グローバル変数

変数	説明
LCD_Seg_initVar	LCD_Seg が初期化されているかを示します。変数は 0 に初期化され、LCD_Seg_Start() が最初に呼び出されたときに 1 に設定されます。これにより、LCD_Seg_Start() ルーチンを最初に呼び出した後、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。 コンポーネントの再初期化が必要な場合には、LCD_Seg_Start() または LCD_Seg_Enable() 関数の前に LCD_Seg_Init() 関数を呼び出すことができます。

uint8 LCD_Seg_Start (void)

説明： LCD コンポーネントを起動し、必要な割り込み、DMA チャネル、フレーム バッファ、ハードウェアを有効にします。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

戻り値： (uint8) cystatus: 標準 API 戻り値。

副作用： この API は、Timewheel Configuration Register 2 からの 1 pulse-per-second ビットを有効にします。これは Low Power 32KHz External Xtal コンポーネント モードでのみ有効です。

void LCD_Seg_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

戻り値： なし

副作用： この API は、LCD_Seg_Init() API で前に有効にされた Timewheel Configuration Register 2 からの 1 pulse-per-second ビットをクリアしません。これは Low Power 32KHz External Xtal コンポーネント モードでのみ有効です。

void LCD_Seg_EnableInt(void)

説明： LCD の割り込みを有効にします。LCD_Seg_Start() が呼び出された場合は必要ありません。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

戻り値： なし

副作用： なし

void LCD_Seg_DisableInt(void)

説明： LCD の割り込みを無効にします。LCD_Seg_Stop() が呼び出された場合は必要ありません。

パラメータ： なし

戻り値： なし

副作用： なし

void LCD_Seg_SetBias(uint8 biasLevel)

説明： この関数は、LCD グラスのバイアス レベルを 128 ある値のどれかに設定します。実際の値の数はアナログ供給電圧 Vdda によって制限されます。バイアス電圧は Vdda を超えることはできません。バイアス レベルを変更すると LCD のコントラストに影響します。

パラメータ： (uint8) biasLevel: 表示のバイアス レベル

戻り値： なし

副作用： なし

uint8 LCD_Seg_WriteInvertState(uint8 invertState)

説明： この関数は、入力パラメータに基づいて表示を反転します。反転はハードウェアで行われ、フレーム バッファ内のディスプレイ RAM の変更は必要ありません。

パラメータ： (uint8) invertState: 表示の反転状態を設定します。

戻り値： (uint8) cystatus: 標準 API 戻り値。

副作用： なし

uint8 LCD_Seg_ReadInvertState(void)

説明： この関数は、標準か反転かという表示反転状態の現在値を返します。

パラメータ： なし

戻り値： (uint8) invertState: 表示の反転状態。

副作用： なし

void LCD_Seg_ClearDisplay(void)

説明： この関数は、表示とその関連フレーム バッファ RAM をクリアします。

パラメータ： なし

戻り値： なし

副作用： なし

uint8 LCD_Seg_WritePixel(uint16 pixelNumber, uint8 pixelState)

説明： 入力パラメータ PixelState の値に基づいてピクセルを設定またはクリアします。ピクセルはパック番号でアドレス設定されます。

パラメータ： (uint16) pixelNumber: はフレーム バッファ内のピクセル位置を示すパック番号です。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。生成された component .h ファイルには、各ピクセルの #defines がこの形式で格納されています。

(uint8) pixelState: 指定された pixelNumber がこのピクセル状態に設定されます。

戻り値： (uint8) status: バイト アドレスとマルチプレクス行番号の範囲チェックに基づく合否状態。ビット位置のチェックは行われません。

副作用： なし

uint8 LCD_Seg_ReadPixel(uint16 pixelNumber)

説明： この機能は、フレーム バッファ内のピクセルの状態を読み取ります。ピクセルはパック番号でアドレス設定されます。

パラメータ： uint16: pixelNumber: はフレーム バッファ内のピクセル位置を示すパック番号です。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。生成された component .h ファイルには、各ピクセルの #defines がこの形式で格納されています。

戻り値： (uint8) pixelState: 指定された PixelNumber の現在の状態を返します。

副作用： なし

void LCD_Seg_Sleep(void)

説明： LCD_Seg_Sleep() 関数は、コンポーネントが有効であるかどうかをチェックし、その状態を保存します。その後、 LCD_Seg_Stop() 関数と LCD_Seg_SaveConfig() 関数を呼び出し、ユーザ設定を保存します。
CyPmSleep() または CyPmHibernate() 関数を呼び出す前に LCD_Seg_Sleep() 関数を呼び出します。電源管理関数については、『PSoC Creator システム リファレンス ガイド』を参照してください。

パラメータ： なし

戻り値： なし

副作用： コンポーネント ピンのドライブ モードを変更しません。

void LCD_Seg_Wakeup(void)

説明： LCD_Seg _Wakeup() 関数は LCD_Seg _RestoreConfig() 関数を呼び出してユーザ設定を復元します。LCD_Seg _Sleep() 関数を呼び出す前にコンポーネントを有効にすると、LCD_Seg_Wakeup() 関数がコンポーネントを再度有効化します。

パラメータ： なし

戻り値： なし

副作用： 最初に LCD_Seg_Sleep() または LCD_Seg_SaveConfig() 関数を呼び出すことなく LCD_Seg _Wakeup() 関数を呼び出すと、予期されない振る舞いにつながる可能性があります。

void LCD_Seg_SaveConfig(void)

説明： この関数は、コンポーネントの設定を保存します。保持されないレジスタも保存します。この関数は、[Configure] (設定) 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_Seg_Sleep() 関数で呼び出します。

パラメータ： なし

戻り値： なし

副作用： なし

void LCD_Seg_RestoreConfig(void)

説明： この関数は、コンポーネントの設定を復元します。維持されないレジスタも復元します。この関数はまた、コンポーネント パラメータ値を LCD_Seg _Sleep() 関数を呼び出した前の状態に復元します。

パラメータ： なし

戻り値： なし

副作用： 最初に LCD_Seg_Sleep() または LCD_Seg_SaveConfig() 関数を呼び出すことなくこの関数を呼び出すと、予期されない振る舞いにつながる可能性があります。

void LCD_Seg_Init(void)

説明： Configure (設定) ダイアログでの設定に従ってコンポーネントのパラメータを初期化または復元します。必要なハードウェア ブロックをすべて設定し、有効にして、フレーム バッファをクリアします。

パラメータ： なし

戻り値： なし

副作用： 全レジスタは、Configure (設定) ダイアログの設定に従って、値が設定されます。この API は、Timewheel Configuration Register 2 からの 1 pulse-per-second ビットを有効にします。これは Low Power 32KHz External Xtal コンポーネント モードでのみ有効です。

void LCD_Seg_Enable(void)

説明： LCD 固定ハードウェアへの出力を有効にし、UDB 信号の生成を有効にします。

パラメータ： なし

戻り値： なし

副作用： なし

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_Seg_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 進桁を表示します。
LCD_Seg_Write7SegNumber_n	7 セグメント ディスプレイ エレメントの 1～5 桁アレイ上に整数値を表示します。
LCD_Seg_WriteBargraph_n	リニアまたはサーキュラ バーグラフ上の整数位置を表示します。
LCD_Seg_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_Seg_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_Seg_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_Seg_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。

関数	説明
LCD_Seg_PutCharDotMatrix_n	ドットマトリクス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_Seg_WriteStringDotMatrix_n	ドットマトリクス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。

注 サフィックス「n」を含む関数名は、コンポーネント マスタマイザ内で同じシンボル タイプのディスプレイ ヘルパーが複数作成されたことを示します。個々のディスプレイ ヘルパー エレメントは、関数名に「n」サフィックスが含まれる API 関数で制御されます。

void LCD_Seg_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 進桁を表示します。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) を通してアドレス設定できます。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 進桁として表示する 0 ～ 15 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。

戻り値： なし

副作用： なし

void LCD_Seg Write7SegNumber_n(uint16 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 ヘルパー機能が必要です。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ドレス設定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、アプリケーション別ユーザ 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 です。

戻り値： なし

副作用： なし

void LCD_Seg_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) を通してアドレス設定できます。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。このモードは幅広い表示器の作成に使用します。

戻り値： なし

副作用： なし

void LCD_Seg_PutChar14Seg_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 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 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。

戻り値： なし

副作用： なし

void LCD_Seg_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 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 14 セグメント英数字ディスプレイ エレメント グループを定義し、関数名内のサフィックス (n) を通してアドレス設定できます。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。

戻り値： なし

副作用： なし

void LCD_Seg_PutChar16Seg_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 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 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 (値が 0 ～ 255 の印刷可能 ASCII および表拡張文字)

(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。

戻り値： なし

副作用： なし

(void) LCD_Seg_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 ヘルパー機能が必要です。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。

戻り値： なし

副作用： なし

void LCD_Seg_PutCharDotMatrix_n(uint8 character, uint8 position)

説明： この関数は、ドットマトリクス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。ドットマトリクス 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 ドットマトリクス 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 カスタマイザ内でドットマトリクス 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。

(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。

戻り値： なし

副作用： なし

void LCD_Seg_WriteStringDotMatrix_n(*uint8 character, uint8 position)

説明： この関数は、ドットマトリクス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。ドットマトリクス 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 ドットマトリクス 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 カスタマイザ内でドットマトリクス 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。

戻り値： なし

副作用： なし

ピン API

これらの API 関数は、セグメント LCD コンポーネントで使用されるピンのドライブ モードを変更するために使用します。

関数名	説明
LCD_Seg_ComPort_SetDriveMode	セグメント LCD コンポーネントのコモン ラインで使用されるすべてのピンのドライブ モードを設定します。

LCD_Seg_SegPort_SetDriveMode	セグメント LCD コンポーネントのセグメント ラインで使用されるすべてのピンのドライブ モードを設定します。

void LCD_Seg_ComPort_SetDriveMode (uint8 mode)

説明： セグメント LCD コンポーネントのコモン ラインで使用されるすべてのピンのドライブ 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 コンポーネントのデータシートを参照してください。

戻り値： なし

副作用： なし

LCD_Seg_SegPort_SetDriveMode(uint8 mode)

説明： セグメント LCD コンポーネントのセグメント ラインで使用されるすべてのピンのドライブ 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 コンポーネントのデータシートを参照してください。

戻り値： なし

副作用： なし

定義

- LCD_Seg_COMM_NUM – コンポーネントの現在の設定でのユーザ定義表示内のコモン ラインの数を定義します。
- LCD_Seg_SEG_NUM – 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_Seg_BIAS_TYPE – コンポーネントの現在の設定でのユーザ定義表示のバイアス タイプを定義します。
- LCD_Seg_BIAS_VOLTAGE – ユーザ定義表示のデフォルト バイアス電圧レベルを定義します。この値は、初期化プロセス中に LCDDAC コントロール レジスタで設定されます。
- LCD_Seg_FRAME_RATE – 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_Seg_EXTRACT_ROW – フレーム バッファ内の指定ピクセルの行を計算します。
- LCD_Seg_EXTRACT_PORT – フレーム バッファ内の指定ピクセルのバイト オフセットを計算します。

- LCD_Seg_EXTRACT_PIN – フレーム バッファ内の指定ピクセルのビット位置を計算します。
- LCD_Seg_WRITE_PIXEL – これは WritePixel 関数のマクロ定義です。
- LCD_Seg_READ_PIXEL – これは ReadPixel 関数のマクロ定義です。
- LCD_Seg_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。これは、カスタマイザピクセルテーブルからの情報と LCD 専用物理ピンの情報を使用します。このマクロはピクセルマッピング機構の基礎を成します。ピクセルテーブルからの各ピクセル名は、フレーム バッファ内で計算されたピクセル位置により定義され、API はピクセル名を使って各ピクセルにアクセスします。

ファームウェア ソースコードの例

PSoC Creator は、[Find Example Project (プロジェクト例を検索)] ダイアログに数多くのプロジェクト例を提供しており、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 (コンポーネント カタログ)] 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 (スタート ページ)] 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 (フィルタのオプション)]** 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 (プロジェクト例を検索)」を参照してください。

機能説明

セグメント LCD コンポーネントは、タイプの異なる LCD グラスを駆動する強力で柔軟性に優れた機構を提供します。Configuration (設定) ダイアログから、コンポーネントの機能をカスタマイズするのに使用できる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_Seg コンポーネントのデフォルト設定は、汎用 LCD ダイレクト 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 LCD_Seg 設定：

- 4 コモン ライン
- 8 セグメント ライン
- 1/3 Bias Type (1/3 バイアス モード)

- 60 Hz リフレッシュ レート
- No Sleep power mode (スリープなしパワー モード)
- 3 V バイアス電圧
- グラス サイズ – 10 sq cm
- 968.99 us ハイドライブ時間
- seg - 1x、 com - 2x ハイドライブ 強度
- 7.57 us ロードライブ時間
- seg – 0.06x、com 0.12x ロードライブ 強度
- 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。

カスタム設定

セグメント LCD コンポーネントの主な特長は異なる文字やレイアウトを持つ LCD の柔軟なサポートです。

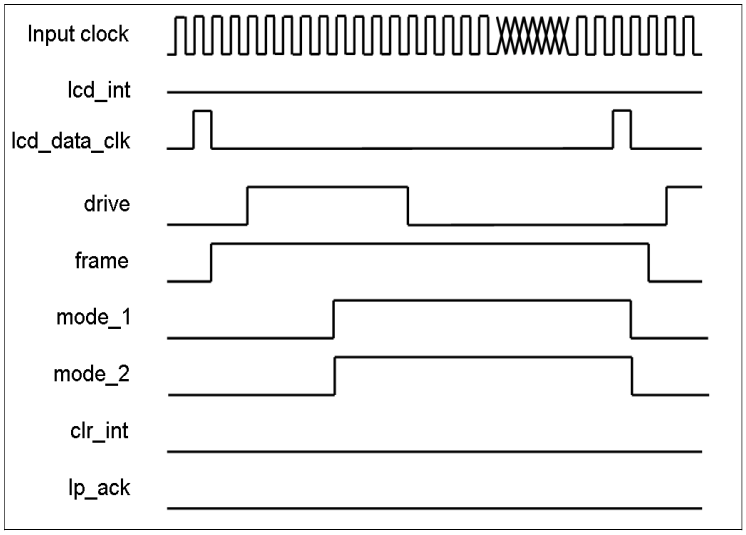
ドライバ パワー モード

セグメント LCD は No Sleep (スリープなし) と Low Power (ローパワー) の 2 つのパワーモードで作動できます。No Sleep (スリープなし) はデフォルト動作モードです。

No Sleep (スリープなし) モード

このモードでは、セグメント LCD は決してスリープ モードにならず、LCD はフレーム全体を通して駆動されます。以下に、No Sleep (スリープなし) モードでの LCD_Seg コンポーネントの UDB 生成 (内部) 信号の波形を示します。

図 1. セグメント LCD コントロール信号 (No Sleep (スリープなし)モード)



ロー パワー モード

このモードでは、LCD は電圧遷移時にのみアクティブに駆動され、LCD システムのアナログ コンポーネントは電圧遷移と次の電圧遷移との間にはスリープ モードになります。内部 LCD タイマは、LCD 画面を更新するために短時間デバイスをスリープ モードから起動します。その後、内部ロジックは次のリフレッシュ シーケンスまでデバイス全体をスリープ モードに戻します。

このモードでは、Frame Rate (フレームレート) パラメータは制限され、LCD タイマ用のクロック ソース [1 KHz (ILO) または 8 KHz (32 K XTAL)]、コモンライン数、波形タイプなど、その他のパラメータに依存します。フレームレートの制限は、LCD タイマの入力周波数の制限によって説明できます。以下に、8 個のコモン、ローパワー ILO モード、タイプ A 波形の設定用の最大フレームレートの計算例を示します。

*max FR = 1000 / (2 * Num Commons) = 62.5 Hz*

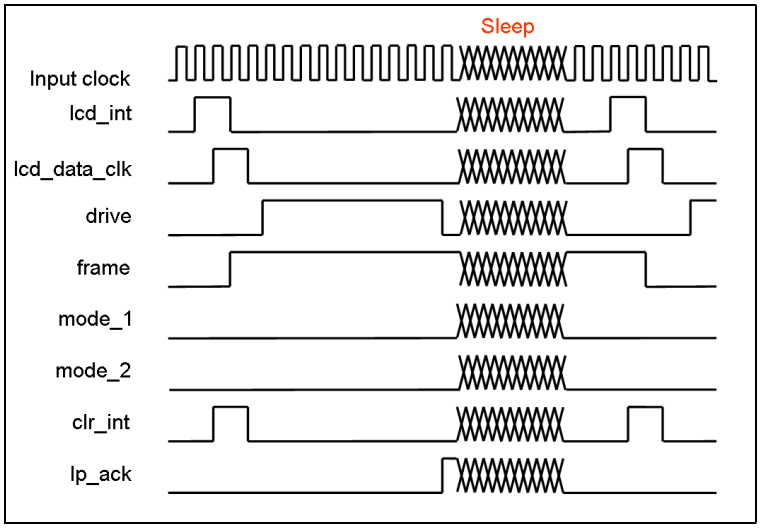
62.5 Hz は、LCD タイマが入力周波数の各クロックサイクルでウェイクアップ信号を生成するときに提供できる最大フレームレートです。

実際には、LCD タイマはウェイクアップ イベントの設定に 1 期間とそれをクリアするために 1 期間必要なだけです。このため、最大フレームレートを 2 で割ることにより 31.25 Hz が得られます。この値の整数部分をとり、実際の最大フレームレートとして 31 Hz が得られます。

タイプ B 波形には 1/2 のウェイクアップ イベントしか必要ないので、タイプ B 波形の実際のフレームレートはタイプ A 波形の 2 倍となります。

以下に、ローパワー モードでの LCD_Seg コンポーネントの UDB 生成 (内部) 信号の波形を示します。

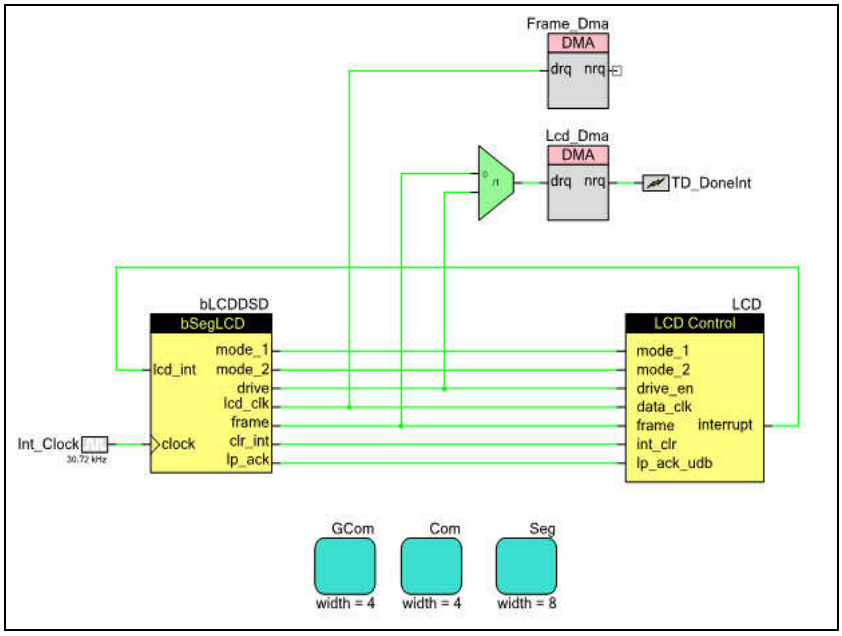
図 2. セグメント LCD コントロール信号 (ローパワー モード)



ブロック図と設定

下の図は、セグメント LCD コンポーネントの内部回路図を示します。これは、基本セグメント LCD コンポーネント、LCD コントロール ブロック (LCD) コンポーネント、DMA コンポーネント、3 つの LCD ポート、 1 つのデジタル ポート、1 つの ISR コンポーネント、 1 つのクロックで構成されています。

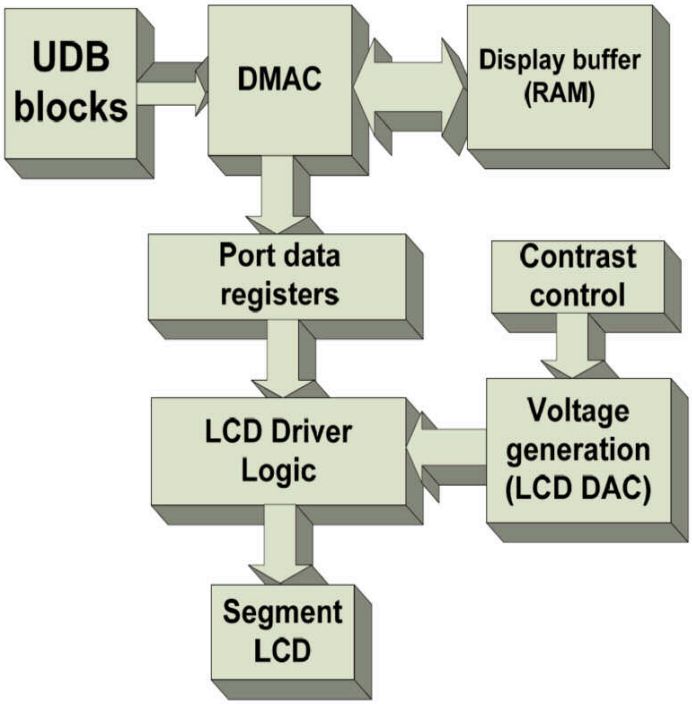
図 3. セグメント LCD コンポーネントの回路図



- **基本セグメント LCD** コンポーネントは、LCD ポート と DMA コンポーネントの適正なタイミング信号の生成を行います。
- **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。
- **LCD** コンポーネントは、必要な DSI ルーティングを取り扱います。このブロックはまた、必要なレジスタ名を cyfitter.h で定義されているとおりに提供します。
- **LCD ポート (GCom、Com、Seg)** は、論理信号を物理ピンにマップするために使用します。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。コモン信号用の LCD ポートは 16 ピン幅に制限されており、セグメント信号用の LCD ポートは 48 ピン幅に制限されています。また、追加のコモン ポート GCom もありますが、これは Ganging (連結) オプションを有効にした場合にのみ含まれます。

トップ レベル アーキテクチャ

図 4. セグメント LCD トップ レベル



レジスタ

LCD_Seg_CONTROL_REG

ビット	7	6	5	4	3	2	1	0
値	フレーム	予約済み			フレーム終了	モード 2	モード 1	クロック有効

- クロック有効：このビットは、これまでのセクションで説明したすべての内部信号の生成を有効にします。
- mode 1：モード[2:0] ビット フィールドの中位ビット。ハイとロー ドライブ強度を定義します。
- mode 2：モード[2:0] ビット フィールドの上位ビット。ハイとロー ドライブ強度を定義します。
- フレーム終了：DMA フレーム トランザクション完了後、同期パルスを生成します。
- フレーム：LCD ドライバ用のフレーム信号を生成します。

LCD_Seg_CONTRAST_CONTROL_REG

適正なバイアス電圧を生成するために LCD DAC で使用するバイアス電圧レベルを保持します。バイアス電圧レベルを変更するための API が提供されています。

ビット	7	6	5	4	3	2	1	0
値	予約済み	予約済み	コントラスト レベル					

- コントラスト レベル：上記の電圧レベル。

LCD_Seg_LCDDAC_CONTROL_REG

ビット	7	6	5	4	3	2	1	0
値	lp イネーブル	予約済み			連続ドライブ	予約済み	バイアス選択	

- バイアス選択：バイアスを選択。
- 連続ドライブ：チップがスリープ状態になっても LCDDAC がアクティブ状態を維持できるようにします。
- lp イネーブル：UDB が LCD サブシステム用の Low Power Ack をゲートできるようにします。

LCD_Seg_TIMER_CONTROL_REG

ビット	7	6	5	4	3	2	1	0
値	期間						clk select (クロック選択)	enable timer (タイマのイネーブル)

- enable timer (タイマのイネーブル) LCD タイマを有効にします。
- clk select (クロック選択)：LCD タイマソース選択クロック。
- 期間：LCD タイマ期間。

LCD_Seg_DRIVER_CONTROL_REG

ビット	7	6	5	4	3	2	1	0
値	予約済み			バイパス イネーブル	pts	反転	モード 0	スリープ モード

- スリープ モード：ローパワー モードでは、「1」に設定されている場合は LCD ドライバの出力バッファはグラウンドに設定され、それ以外の場合は Hi-Z に設定します。

- mode 0: モード[2:0] ビット フィールドの低位ビット。ハイとロー ドライブ強度を定義します。
- 反転：設定すると、セグメント ピン上のデータを反転します。
- pts: "0" - 標準動作、VOUT = VIO-0.5V、 "1" - VOUT = VIO。
- バイパス イネーブル： "1" - LCD ドライバ内のドライブ バッファをバイパスし、代わりに選択された電圧入力に直接接続します。"0" - 標準動作、電圧はドライバ バッファを通して通過します。

LCD_Seg_LCDDAC_SWITCH_REG[0..4]

ビット	7	6	5	4	3	2	1	0
値	予約済み						スイッチ コントロール[0..4]	

- スイッチ コントロール[0..4] このビットフィールド群は LCD ドライバの電圧源を選択します。

リファレンス

該当なし

DC 電気的特性と AC 電気的特性

LCD 直接駆動 DC 仕様

パラメータ	説明	条件	Min	Typ	Max	単位
I_{CC}	LCDシステム動作電流	デバイス スリープ モード、400 Hz でウェイクアップして LCD をリフレッシュ、バス クロック = 3 MHz、Vddio = Vdda = 3 V、4 コモン、16 セグメント、1/4 デューティサイクル、50 Hz フレームレート、グラス接続なし	–	38	–	µA
I_{CC_SEG}	セグメント ドライバあたりの電流	ストロングドライブモード	–	260	–	µA
$V_{バイアス}$	LCD バイアス範囲（V_{BIAS}は LCD DACのメイン出力電圧（V0））	$V_{DDA} \geq 3$ V および $V_{DDA} \geq V_{バイアス}$	2	–	5	V
	LCD バイアス ステップ サイズ	$V_{DDA} \geq 3$ V および $V_{DDA} \geq V_{バイアス}$	–	9.1 × V_{DDA}	–	mV

	セグメント/コモン ドライバあたりのLCD静電容量	ドライバは連結可能	–	500	5000	pF
	長時間セグメント オフセット		–	–	20	mV
I_{OUT}	セグメント ドライバあたりの出力駆動電流	Vddio = 5.5V、ストロング出力	355	–	710	µA

LCD ダイレクト ドライブ AC 仕様

パラメータ	説明	条件	Min	Typ	Max	単位
f_{LCD}	LCD フレームレート		10	50	150	Hz

コンポーネントの変更

ここでは、前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
2.10	固定 LCD_Seg_Start() ルーチン	実装にエラーがあったため、グローバル割り込みが前にイネーブルされていた場合、これをサイレントにディスエーブルしていました。
	カスタマイザの Driver Power Settings (ドライバ パワー設定) タブを変更しました。2 つの新しいフィールド Custom Step (カスタム ステップ) と Default Step (デフォルト ステップ) を追加しました。	ローパワー モードでの High Drive Time (ハイドライブ時間) パラメータの新しいオプションが使用可能です。現状選択できるデフォルト ステップに加え、より精密なカスタム ステップを選択できるようになりました。これはマスタ クロックの周波数に基づきます。また
	LCD リフレッシュ シーケンスの完了前に SegLCD がスリープ状態になるという問題を修正。	リフレッシュ シーケンスの開始時に内部コンポーネント signal lp_ack がハイになっていましたが、正しくはリフレッシュ シーケンスの終了時にハイになる必要があります。
	ハイ ドライブ強度が "seg=2x com=2x" または "seg=1x com=4x" に設定されたときに LCD グラス画像が反転するという問題を修正。	この問題は、LCD ドライバ コントロール レジスタに不正な値が書き込まれることに起因していました。この値はマスクとの AND 演算で生成されていましたが、このマスクが正しくありませんでした。このマスクを正しいものに変えることでこの問題が修正できました。
	LCD_Seg_WriteBargraph() API fix.	モード パラメータが 10 に設定されたときに LCD のバーグラフピクセルの不正な出力とし

バージョン	変更の説明	変更の理由 / 影響
		て反映されていた LCD_Seg_WriteBargraph() API の問題を修正。これは、ポジション パラメータが "Position + 10 > Bar Graph pixels" (位置 + 10 > バーグラフピクセル数) という条件を満たすときに発生していました。この場合、バーグラフの最後のピクセルが設定されていませんでしたが、これは設定する必要がありました。
	データシートに特性データを追加	
	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。	このツールは互換性のないシリコン上でこのコンポーネントを使うとエラーを表示します。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。
	データシートのマイナーな編集と更新	
2.0.a	データシートの LCD_Seg_Init()、 LCD_Seg_Start()、LCD_Set_Stop() の API に注を追加。	
	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。	このツールは互換性のないシリコン上でこのコンポーネントを使うとエラーを表示します。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。
2.0	Sleep/Wakeup (スリープ/ウェイクアップ) と Init/Enable (初期化/有効化) API を追加。	ローパワー モードをサポートし、ほとんどのコンポーネントの初期化と有効化の制御を分離する共通インターフェースを提供するため。
	スリープ モード API の宣言を含む新しい API ファイル - *SegLCD_PM.c* を追加。	ローパワー モードをサポートするための新しい要件。

バージョン	変更の説明	変更の理由 / 影響
	PSoC 3 ES3 以降をサポートするようにコンポーネントがアップデートされました。[Configure] (設定) ダイアログが更新されました。 • Enable Debug Mode (デバッグ有効モード) と Low Drive Mode (ロードライブ モード) のコントロールは使用されなくなったので削除しました。 • 新しいコントロール Glass Size (グラスサイズ)、High Drive Strength (ハイドライブ強度)、Low Drive Strength (ロードライブ強度) を追加。 • 古いモード Always Active (常に有効) と Low Power (ローパワー) を新しいモード No Sleep (スリープなし)、Low Power ILO (ローパワー ILO)、Low Power 32XTAL (ローパワー 32XTAL) に変更することで、ドライバ パワー モードの選択を変更。 • Custom Characters (カスタム文字) タブを追加し、ユーザがドットマトリクス ヘルパー用のカスタム文字を作成できるようにしました。 • バイアス タイプ コントロールを編集可能とし、バイアス電圧の選択値を変更しました。	PSoC 3 ES3 デバイスと LCD HW アーキテクチャをサポートするという新しい要件を満たすために新しい 2.0 バージョンが作成されました。 バージョン 1.xx は PSoC 3 ES2 および PSoC 5 ES1 シリコン リビジョンをサポートします。
	Added `=ReentrantKeil(LCD_Seg_ . "_...")` を以下の関数に追加。 LCD_Seg_Stop() LCD_Seg_EnableInt() LCD_Seg_DisableInt() LCD_Seg_SetBias() LCD_Seg_WriteInvertState() LCD_Seg_ReadInvertState() LCD_Seg_RedPixel() LCD_Seg_SaveConfig() LCD_Seg_RestoreConfig()	ユーザが必要ならこれらの API を再入可能できるようにするため。

© Cypress Semiconductor Corporation, 2011. 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 (サイプレス セミコンダクタ社) 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。特許またはその他の権限下で、ライセンスを譲渡また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、または安全の用途のために使用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ト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® は、サイプレス セミコンダクタ社の登録商標であり、PSoC Creator™ およびプログラマブル System-on-Chip™ は、サイプレス セミコンダクタ社の商標です。本書で言及するその他すべての商標または登録商標は、各社の所有物です。

全てのソース コード (ソフトウェアおよび / またはファームウェア) はサイプレス セミコンダクタ社 (以下「サイプレス」) が所有し、全世界の特許権保護 (米国およびその他の国)、米国の著作権法なら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り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 ソフトウェアおよび / また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コピー、使用、変更そして作成するためのライセンス、ならびにサイプレスのソース コードおよ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、また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、明示的または黙示的を問わず、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、商品性また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または回路を適用また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ト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。