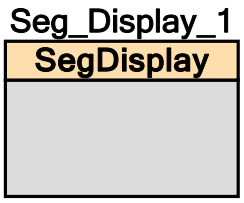
セグメント ディスプレイ (Seg_Display)
1.0



特長

- 2 ～ 768 ピクセルまたはシンボル
- 1/3、1/4、1/5 バイアス対応
- 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 5 シリコン バージョンをサポート
- PSoC 3 未サポート。PSoC 3 Production シリコン リビジョンまたはそれ以上には Segment LCD バージョン 3.0 を使用

一般的な説明

セグメント ディスプレイ (Seg_Display) コンポーネントは、最大 16x のマルチプレックス レシオでで 3.3-V および 5.0-V LCD パネル直接駆動できます。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、パネルバイアスを 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。またセグメントディスプレイ コンポーネントは、次のタイプのパネル内表示構造の扱い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 ディスプレイ 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 - PSoC® 3: Segment LCD Direct Drive* を参照してください。

セグメント ディスプレイを使用する場合

ダイレクト セグメント ドライブ LCD コンポーネントは、2x ～16x のマルチプレックス レシオで 3.3 V または 5.0 V LCD パネルを直接駆動する必要がある場合に使用します。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。

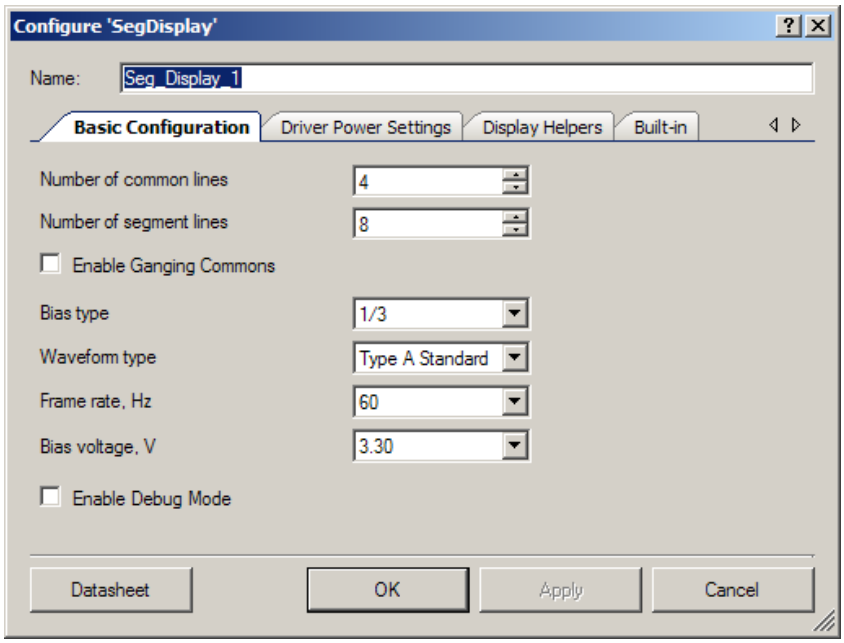
入出力接続

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。

コンポーネント パラメータ

セグメント ディスプレイ 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 ディスプレイ 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。指定できる値は 2 ～ 62 です - **コモンラインの数**。デフォルトは **8** です。

Enable Ganging Commons (コモンの連結)

このチェックボックスを選択して、コモン信号を駆動する PSoC ピンを連結します。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)

波形タイプを定義します。タイプ A - 単一フレームの平均 0 VDC (デフォルト) またはタイプ 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。指定できる値の範囲は 10 Hz ～ 150 Hz です。デフォルトは **60 Hz**。

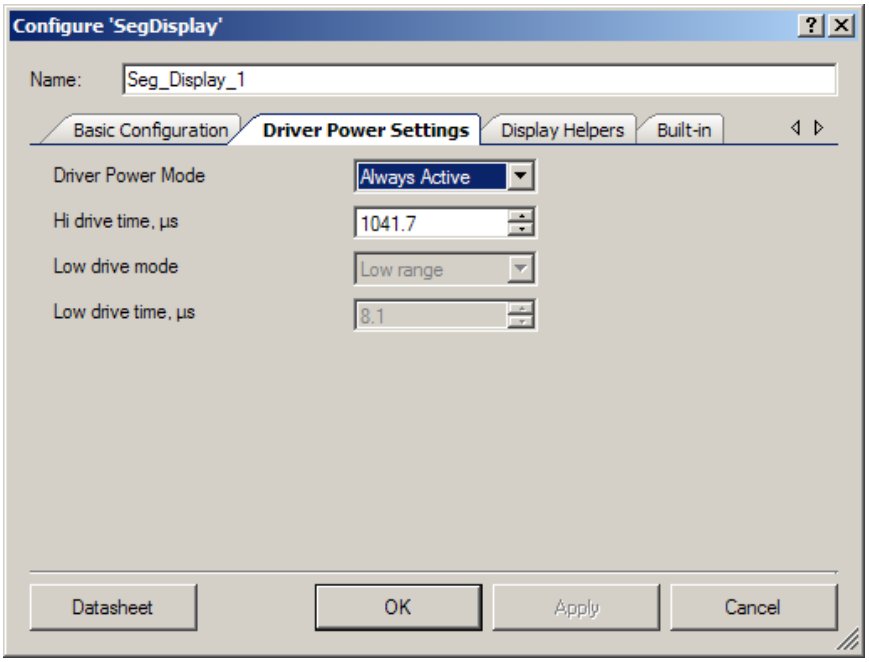
Bias Voltage (バイアス電圧)

LCD DAC のバイアス電圧レベルを設定します。指定できる値の範囲は 2 V ～ 5.2 V です。デフォルト値は **3.3 V** です。

Enable Debug Mode (デバッグ モードを有効にする)

選択すると、デバッグ ポート がコンポーネントに追加され、LCD ドライバを駆動する信号を見る事ができます。たとえば、ハイとローのドライブ時間を確認するのに便利です。

Driver Power Settings (ドライバ パワー設定) タブ



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 2 つのパワー モードがあります。

- **AlwaysActive (常時アクティブ)**:LCD DAC は常にオンになります
- **Low Power(ローパワー)**: LCD DAC は電圧遷移の間オフになります

このデータシートのドライバ パワー モードセクションの機能の説明を参照してください。

High Drive Time (High ドライブ時間)

1 回の電圧遷移内で High ドライブ モードにする時間を指定します。

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** (High ドライブ時間) パラメータは最小値に設定されます。**High 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 **High ドライブ時間**の最大値は以下の式で計算されます。

HiDriveTimemax= 1/(FrameRate × (2 × NumCommonLines) × 256) × 253

Low Drive Mode (Low ドライブ モード)

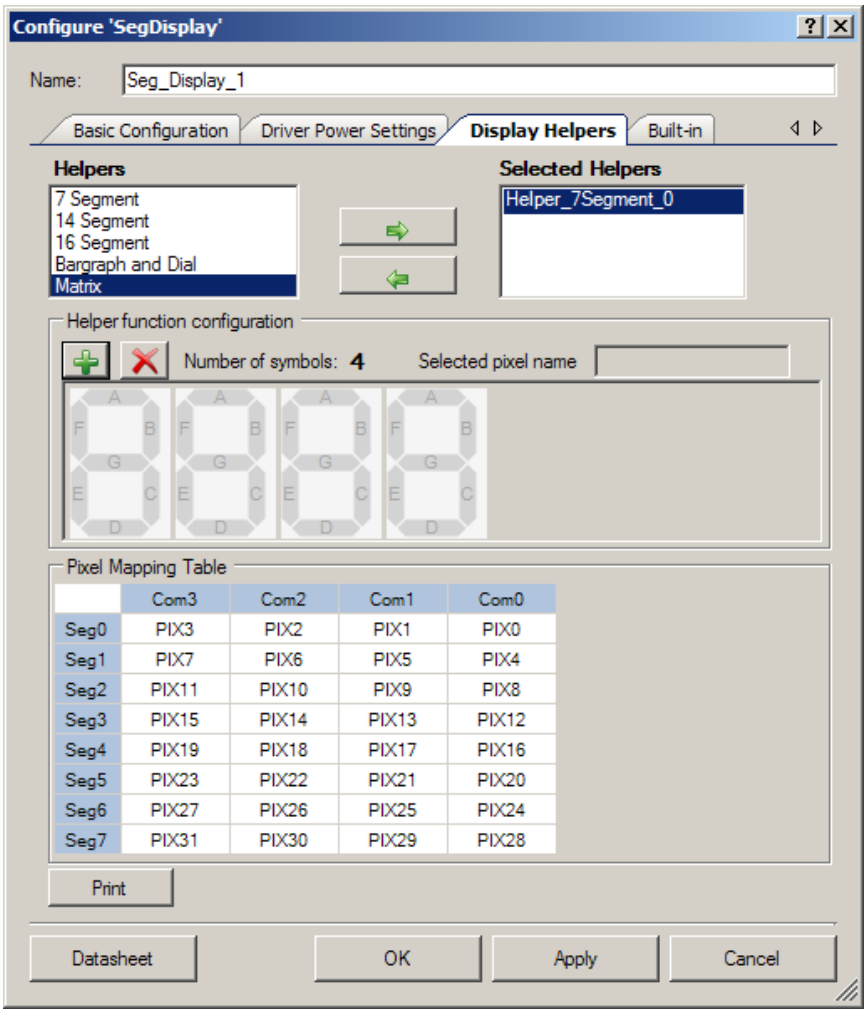
このパラメータは、**Driver Power Mode** (ドライバ パワー モード) を **Low Power** (ローパワー) に設定した場合に使用できます。次の 2 つの **Low ドライブ モード**から選択できます。

- **Low range**(ロー レンジ) – Low ドライブ モードにします
- **High range**(ハイ レンジ) – セカンド Low ドライブハイ カレント モード (Lo2) にします

Low DriveTime (Low ドライブ時間)

1 回の電圧遷移内で Low ドライブ モードをアクティブにする時間を指定します。

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 (選択済みヘルパー)

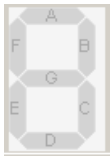
Helpers (ヘルパー) リストから使用するヘルパーのタイプを選択し、右向き矢印ボタンをクリックすることで、1 つまたは複数のヘルパーを **Selected Helpers** (選択済みヘルパー) リストに追加できます。新しいヘルパーをサポートするのに十分なピンがない場合は、そのヘルパーは追加されません。ヘルパーを削除するには、**Selected Helpers** (選択済みヘルパー) リストからそのヘルパーを選択して左向き矢印ボタンをクリックします。

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。

選択済みヘルパーが **Selected Helpers** (選択済みヘルパー)リストに表示される順序は重要です。デフォルトでは、**Selected Helper**(選択済みヘルパー) 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。**Selected Helpers** (選択済みヘルパー)を リストから削除すると、残りのヘルパーの名前は変更されます。ヘルパーが追加されると、使用できるサフィックス(訳注:後付けされる数値)のうちもっとも小さいものが使用されます。

各ヘルパーに API が提供されます。詳しくは、「アプリケーションプログラミングインタフェース」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 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) – このヘルパーで扱えるのは最長 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全体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のみ、または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のどちらもできます。

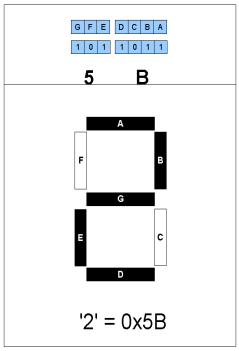
- **Matrix Helper** (ドットマトリクス ヘルパー) – このヘルパーは最大 8 つのキャラクタ エレメントをサポートします。このコンポーネントは 5x7 または 5x8 行/列の文字をサポートします。より長い文字列は、複数のドットマトリクス ヘルパーを設定し、ディスプレイの隣接ドットマトリクス セクションを定義することで作成できます。このヘルパーは単一の ASCII 文字を表示することもヌル終端文字列を表示することもできます。

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。マトリクスの行には 7 または 8 個のシーケンシャル コモン ドライバを、マトリクスの列には 5 ～ 40 個のシーケンシャル セグメント ドライバを使用する必要があります。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。

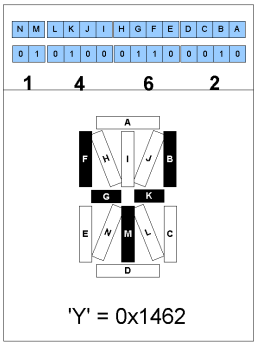
文字の符号化

高レベルのヘルパー API はには、自身のルックアップ テーブルがあります。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)。

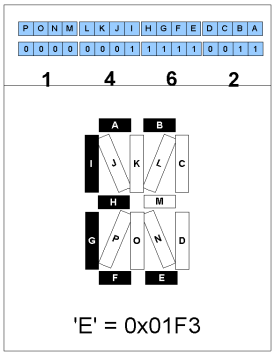
7 セグメント符号化



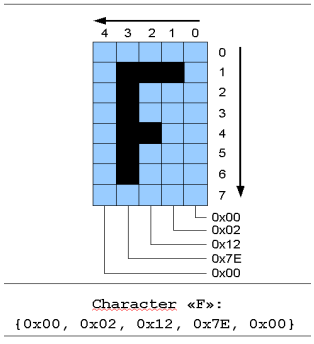
14 セグメント符号化



16 セグメント符号化



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、ヘルパーへのシンボルの追加や削除、ピクセルの命名などを含む、ヘルパーの設定を行うことができます。

1. **Selected Helpers** (選択済みヘルパー) リストからヘルパーを選択します。
2. **[+]** または **[x]** ボタンをクリックして、選択済みヘルパーのシンボルを追加または削除します。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、必要に応じて変更することができます。

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、それぞれ異なるフォーマットになっています。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 ヘルパー内のいずれか 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	は、7 セグメント ヘルパーの一部であることを示します
SEG4	は、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	は、7 セグメント シンボル内の一意なセグメントを示します

デフォルトのピクセル名のみの場合、ピクセル名の一意部分がシンボル画像に表示されます。(訳注:共通部分は省略されます) ピクセル名を変更する場合、同じプリフィックス名を他のピクセルと共有していても、ピクセル名全体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、そのピクセルの名称はヘルパー 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、置き換わるものではありません。ヘルパー関数に関連付けられたピクセルのデフォルトピクセル名は再使用(他のピクセルの名称として使用)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。正しい割り当てについては、LCD パネル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。この方法は、使用可能ないずれかのヘルパー タイプに関連付けられていないピクセルに名前を付けるのに使用できます。

Print (印刷) ボタンは、**ピクセルマッピング テーブル**を印刷します。

Clock Select（クロック選択）

Seg_Display コンポーネントは 2 つの内部クロックを使用し、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、2 つの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。最初のクロックはリフレッシュ レートの周波数を生成し、2 番目のクロックは Low ドライブバッファ用に 100 KHz のクロックを生成します。

配置

Seg_Display 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。プロジェクト内でコンポーネントのインスタンスが2つ以上使われていると、ビルド プロセス中に配置エラーとなります。

デフォルトの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

リソース	リソースのタイプ						API メモリ(バイト)		ピン(外部入出力当たり)
	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	フラッシュ	RAM	
基本	1～2	3～5	1	2	0	1	2068	77	3 ~ 62
基本、7 セグメントヘルパー	1～2	3～5	1	2	0	1	2441	95	3 ~ 62
基本、14 セグメントヘルパー	1～2	3～5	1	2	0	1	3319	293	3 ~ 62
基本、16 セグメントヘルパー	1～2	3～5	1	2	0	1	3399	293	3 ~ 62
基本、ドットマトリックスヘルパー	1～2	3～5	1	2	0	1	3182	293	3 ~ 62
基本、バーグラフヘルパー	1～2	3～5	1	2	0	1	4462	293	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。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「Seg_Display_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「Seg_Display」を使っています。

関数	説明
Seg_Display_Start()	LCD コンポーネントを起動し、必要な割り込みを有効にします。
Seg_Display_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。
Seg_Display_EnableInt()	LCD の割り込みを有効にします。Seg_Display_Start() が呼び出される場合は必要あり

関数	説明
	ません。
Seg_Display_DisableInt()	LCD の割り込みを無効にします。Seg_Display_Stop() が呼び出される場合は必要ありません。
Seg_Display_SetBias()	LCD パネルのバイアス レベルを 128 ある値のどれかに設定します。
Seg_Display_WriteInvertState()	入力パラメータに基づいて表示を反転します。
Seg_Display_ReadInvertState()	標準か反転かを示す、表示反転状態の現在値を返します。
Seg_Display_ClearDisplay()	表示とその関連フレーム バッファ RAM をクリアします。
Seg_Display_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。
Seg_Display_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。
Seg_Display_SetAwakeMode()	ローパワー モードの場合、LCD ドライバ バッファの出力をハイインピーダンスに設定します。
Seg_Display_Sleep()	LCD を停止し、ユーザ設定を保存します。
Seg_Display_Wakeup()	ユーザ構成を復元し、イネーブルにします。
Seg_Display_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。
Seg_Display_Enable()	コンポーネントを有効にします。
Seg_Display_SaveConfig()	LCD 設定を保存します。
Seg_Display_RestoreConfig()	LCD 設定を復元します。

グローバル変数

変数	説明
Seg_Display_initVar	Seg_Display が初期化されているかどうかを示します。変数は 0 に初期化され、Seg_Display_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、Seg_Display_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は、Seg_Display_Init() 関数を Seg_Display_Start() または Seg_Display_Enable() 関数の前に呼び出します。

uint8 Seg_Display_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 はクリアしません。

パラメータ: なし

返り値: uint8 cystatus: 標準 API 返り値

返り値	説明
CYRET_LOCKED	一部の DMA TD（トランザクションディスクリプタ）、またはチャンネルが既に使用されています。
CYRET_SUCCESS	関数は正しく完了しました

副作用: なし

void Seg_Display_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 オフセットによるダメージを避けます。フレーム バッファはクリアしません。

パラメータ: なし

返り値: なし

副作用: なし

void Seg_Display_EnableInt(void)

説明: LCD の割り込みを有効にします。Seg_Display_Start() が呼び出される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

返り値: なし

副作用: なし

void Seg_Display_DisableInt(void)

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

パラメータ: なし

返り値: なし

副作用: なし

uint8 Seg_Display_SetBias(uint8 biasLevel)

説明: この関数は、LCD パネルのバイアス レベルを 128 ある値のどれかに設定します。実際に設定できる値の数はアナログ供給電圧 V_{DDA} によって制限されます。バイアス電圧は V_{DDA} を超えることはできません。バイアス レベルの変更は LCD のコントラストに影響を与えます。

パラメータ: uint8 biasLevel: ディスプレイのバイアス レベル

返り値: uint8 cystatus: 標準 API 返り値。

返り値	説明
CYRET_BAD_PARAM	**バイアス レベル**のパラメータの評価に失敗しました
CYRET_SUCCESS	関数は正しく完了しました

副作用: なし

uint8 Seg_Display_WriteInvertState(uint8 invertState)

説明: この関数は、入力パラメータに基づいてディスプレイを反転します。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: ディスプレイの反転状態を設定します

値	説明
Seg_Display_NORMAL_STATE	ディスプレイの通常状態に設定
Seg_Display_INVERTED_STATE	ディスプレイの反転状態に設定

返り値: uint8 cystatus: 標準 API 返り値

返り値	説明
CYRET_BAD_PARAM	**反転状態**パラメータの評価に失敗しました
CYRET_SUCCESS	関数は正しく完了しました

副作用: なし

uint8 Seg_Display_ReadInvertState(void)

説明: この関数は、ディスプレイの状態の現在値を返します。

パラメータ: なし

返り値: uint8 invertState: ディスプレイの反転状態

返り値	説明
Seg_Display_NORMAL_STATE	ディスプレイの通常状態
Seg_Display_INVERTED_STATE	ディスプレイの反転状態

副作用: なし

void Seg_Display_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

返り値: なし

副作用: なし

uint8 Seg_Display_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、パック化数値（packed number）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示すパック化数値（packed number）です。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。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。

値	説明
Seg_Display _PIXEL_STATE_OFF	ピクセルをオフに設定します
Seg_Display_PIXEL_STATE_ON	ピクセルをオンに設定します
Seg_Display_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マルチプレックス行番号の範囲チェックの合否。ビット位置のチェックは行われません。

返り値	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 Seg_Display_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、パック化数値（packed number）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示すパック化数値（packed number）です。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。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。

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
0x00	ピクセルはオンです
0x01	ピクセルはオフです
0xFF	ピクセルが接続されていません

副作用: なし

void Seg_Display_SetAwakeMode(void)

説明: ローパワー モードの場合、LCD ドライバ バッファの出力をハイインピーダンスに設定します。

パラメータ: なし

返り値: なし

副作用: なし

void Seg_Display_SetSleepMode(void)

説明: ローパワー モードの場合、LCD ドライバ バッファの出力をグラウンド(0V)に設定します。

パラメータ: なし

返り値: なし

副作用: なし

void Seg_Display_Sleep(void)

説明: Seg_Display_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、Seg_Display_Stop() 関数を呼び出し、更にSeg_Display_SaveConfig() 関数を呼び出して、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 Seg_Display_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

返り値: なし

副作用: コンポーネン ピンのドライブ モードは変更しません。

uint8 Seg_Display_Wakeup(void)

説明: Seg_Display_Wakeup() 関数は Seg_Display_RestoreConfig() 関数を呼び出してユーザ設定を復元します。Seg_Display_Sleep() 関数を呼び出す前にコンポーネントがイネーブルされると、Seg_Display_Wakeup() 関数がコンポーネントを再度イネーブルにします。

パラメータ: なし

返り値: uint8 cystatus: 標準 API 返り値

返り値	説明
CYRET_LOCKED	一部の DMA TD 、またはチャンネルが既に使用されています
CYRET_SUCCESS	関数は正しく完了しました

副作用: 最初に Seg_Display_Sleep() または Seg_Display_SaveConfig() 関数を呼び出すことなく Seg_Display_Wakeup() 関数を呼び出すと、予期しない動作が発生する可能性があります。

void Seg_Display_Init(void)

説明: 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。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

返り値: なし

副作用: なし

uint8 Seg_Display_Enable(void)

説明: コンポーネントをイネーブルします。必要なクロックをすべて有効にし、レジスタの初期値を設定して、コンポーネントのリセットを実行します。

パラメータ: なし

返り値: uint8 cystatus: 標準 API 返り値。

返り値	説明
CYRET_LOCKED	一部の DMA TD 、またはチャンネルが既に使用されています
CYRET_SUCCESS	関数は正しく完了しました

副作用: なし

void Seg_Display_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 Seg_Display_Sleep() 関数で呼び出します。

パラメータ: なし

返り値: なし

副作用: なし

void Seg_Display_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。また、コンポーネントのパラメータ値を Seg_Display_Sleep() 関数を呼び出す前の状態に復元します。

パラメータ: なし

返り値: なし

副作用: 最初に Seg_Display_Sleep() または Seg_Display_SaveConfig() 関数を呼び出すことなくこの関数を呼び出すと、予期しない動作が発生する可能性があります。

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
Seg_Display_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 1 桁の16 進数を表示します。
Seg_Display_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。
Seg_Display_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。
Seg_Display_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1つ表示します。
Seg_Display_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。
Seg_Display_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1つ表示します。
Seg_Display_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。
Seg_Display_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1つ表示します。
Seg_Display_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 Seg_Display_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1桁の 16 進数を表示します。数字は、0～9 と A～F の範囲の 16 進数値です。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します。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(n) を使って選択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 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 から16の間の符号なし整数値（訳注：16は表示クリア用）。16進数文字のASCIIコードも使用できます。**digit** が無効な場合、指定した位置に表示される**値**は0になります。このパラメータを 16 に設定すると、指定した**位置**の数字をクリアします。

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 Seg_Display 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 ヘルパー機能を使用します。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、アプリケーション毎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、最上位桁から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。

返り値: なし

副作用: なし

void Seg_Display_WriteBargraph_n(uint8 location, uint8 mode)

説明: この関数は、1から255セグメントのバーグラフに 8 ビット整数値を表示します（左から右に数えます）。バーグラフは、1 から255 セグメントの間の、任意のユーザ定義サイズにすることができます。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します。複数のバーグラフ ディスプレイをフレーム バッファ内に作成し、関数名内のサフィックス(n) を使って選択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。値がゼロの時は、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。このモードは幅広の表示作成に使用できます。

返り値: なし

副作用: なし

void Seg_Display_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 ヘルパー機能を使用します。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) を使って選択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: 左端を 0 として左から右に数えた文字の位置。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 Seg_Display_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します。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) を使って選択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: ヌル終端文字列のポインタ。

uint8 position: 左端を 0 として左から右に数えた最初の文字の位置。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 Seg_Display_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 ヘルパー機能を使用します。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) を使って選択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 および拡張文字コード表の文字) 。

uint8 position: 左端を 0 として左から右に数えた文字の位置。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) Seg_Display_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 ヘルパー機能を使用します。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) を使って選択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: ヌル終端文字列のポインタ。

uint8 position: 左端を 0 として左から右に数えた最初の文字の位置。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 Seg_Display_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 ヘルパー機能を使用します。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) を使って選択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: 左端を 0 として左から右に数えた文字の位置。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 Seg_Display_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 ヘルパー機能を使用します。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) を使って選択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: ヌル終端文字列のポインタ。

uint8 position: 左端を 0 として左から右に数えた最初の文字の位置。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、セグメント ディスプレイ 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	説明
Seg_Display_ComPort_SetDriveMode	セグメント ディスプレイ 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。
Seg_Display_SegPort_SetDriveMode	セグメント ディスプレイ 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 Seg_Display_ComPort_SetDriveMode(uint8 mode)

説明: セグメント ディスプレイ 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

Seg_Display_SegPort_SetDriveMode(uint8 mode)

説明: セグメント ディスプレイ 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

Macros

Seg_Display_COMM_NUM

コンポーネントが現在コンフィグレーションされているユーザ定義ディスプレイのコモン ライン数を定義します。

Seg_Display_SEG_NUM

コンポーネントが現在コンフィグレーションされているユーザ定義ディスプレイのセグメント ライン数を定義します。

Seg_Display_BIAS_TYPE

コンポーネントが現在コンフィグレーションされているユーザ定義ディスプレイのバイアス タイプを定義します。

Seg_Display_BIAS_VOLTAGE

ユーザ定義ディスプレイのデフォルト 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 制御レジスタ内に設定されます。

Seg_Display_FRAME_RATE

コンポーネントが現在コンフィグレーションされているユーザ定義ディスプレイのリフレッシュ レートを定義します。

Seg_Display_WRITE_PIXEL

これは、void 型 Seg_Display_WritePixel() 関数のマクロ定義です。

Seg_Display_READ_PIXEL

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

Seg_Display_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 ピクセル テーブル、および LCD 専用の物理ピンに関する情報を参照してください。このマクロはピクセル マッピング メカニズムのベースになるものです。ピクセルテーブルの各ピクセル名は、フレーム バッファ内で計算されたピクセル位置で定義されます。API は個々のピクセルをアクセスするのにピクセル名を使用します。

ファームウェア ソースコードのサンプル

PSoC Creator は、[Find Example Project（サンプルプロジェクトを検索）] ダイアログに多数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サンプル コードが含まれています。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

デフォルト設定

Seg_Display 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。デフォルトの Seg_Display 設定は:

- 4 つのコモンライン
- 8 つのセグメントライン
- 60 Hz リフレッシュ レート
- Always Active (常に有効) パワー モード
- 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 エレメントのすべての関数が含まれているとは限りません。

カスタム設定

セグメント ディスプレイ コンポーネントの主な特長は、さまざまな文字やレイアウトを持つ LCD の柔軟なサポートです。

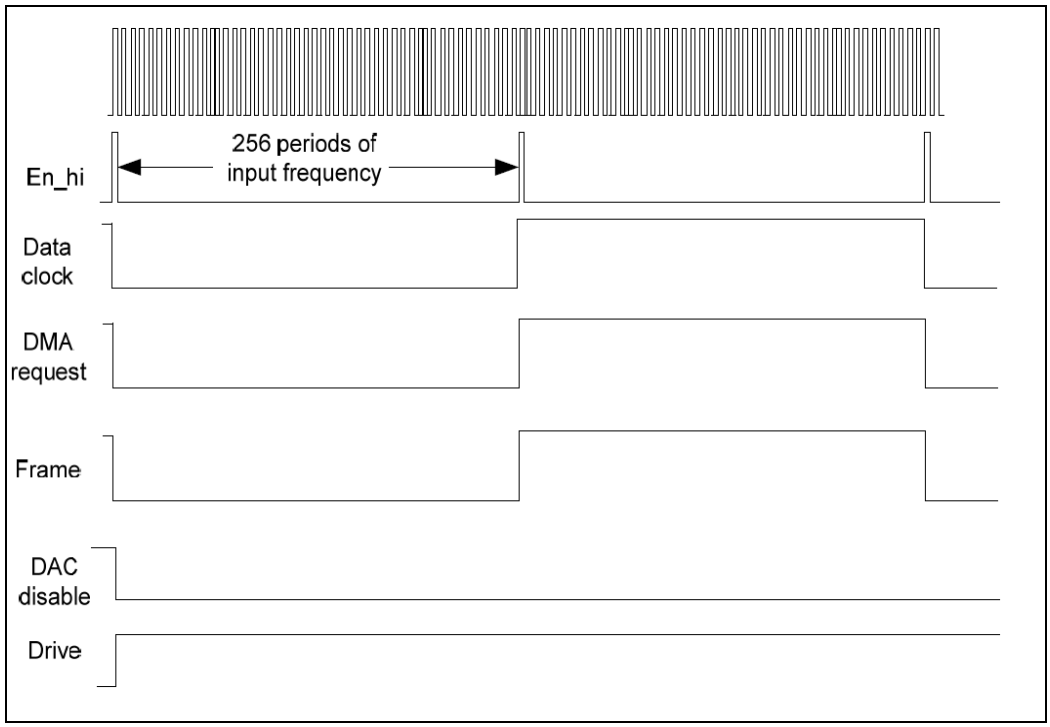
ドライバ パワー モード

Always Active (常に有効) モード

この使用モデルでは、LCD はフレーム全体を通して駆動されます。これは、コンポーネントがイネーブルされるたびに LCD DAC に電源が入り、内部信号ドライブがハイにアサートされることを意味します。

図 1 は、Always Active (常に有効) モード (Type A) での Seg_Display コンポーネントの UDB 生成 (内部) 信号の波形を示しています。

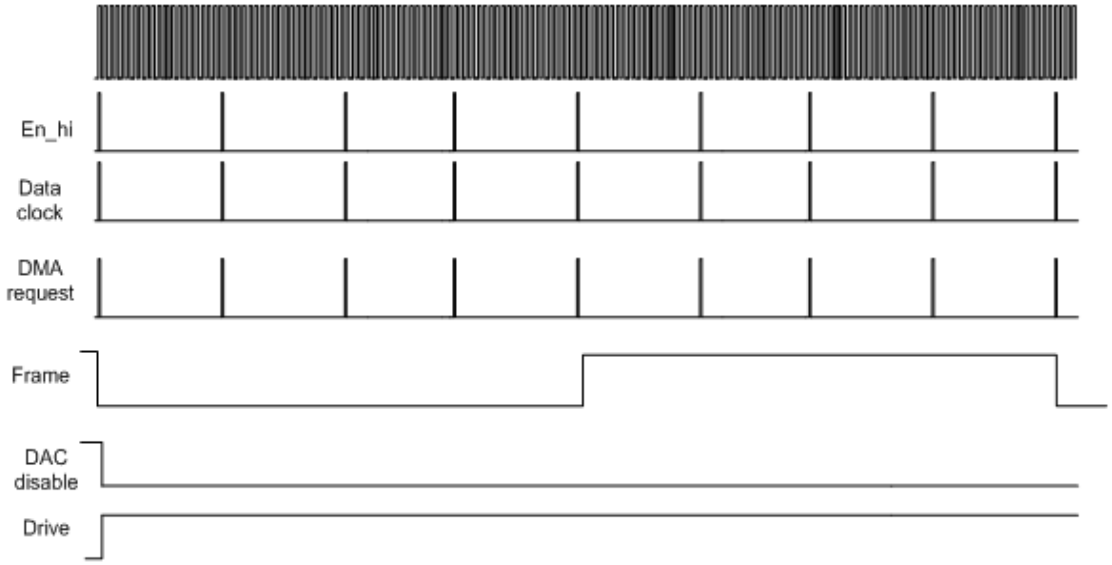
図 1. セグメント LCD コントロール信号 Type A 波形 (Always Active モード)



注 詳細については、「タイミングの計算」を参照してください。

図 2 は、Always Active (常に有効) モード (Type A) での Seg_Display コンポーネントの UDB 生成信号の波形を示しています。

図 2. セグメント LCD コントロール信号 Type A 波形 (Always Active モード)



1/4 マルチプレクス レシオ ケースの信号を表示。

ロー パワー モード

この使用モデルでは、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時以外は電源が切られます。

図 3 は、Low Power モード (Type A) での Seg_Display コンポーネントの UDB 生成信号の波形を示しています。

図 3. セグメント ディスプレイ コントロール信号 Type A 波形 (ロー パワー モード)

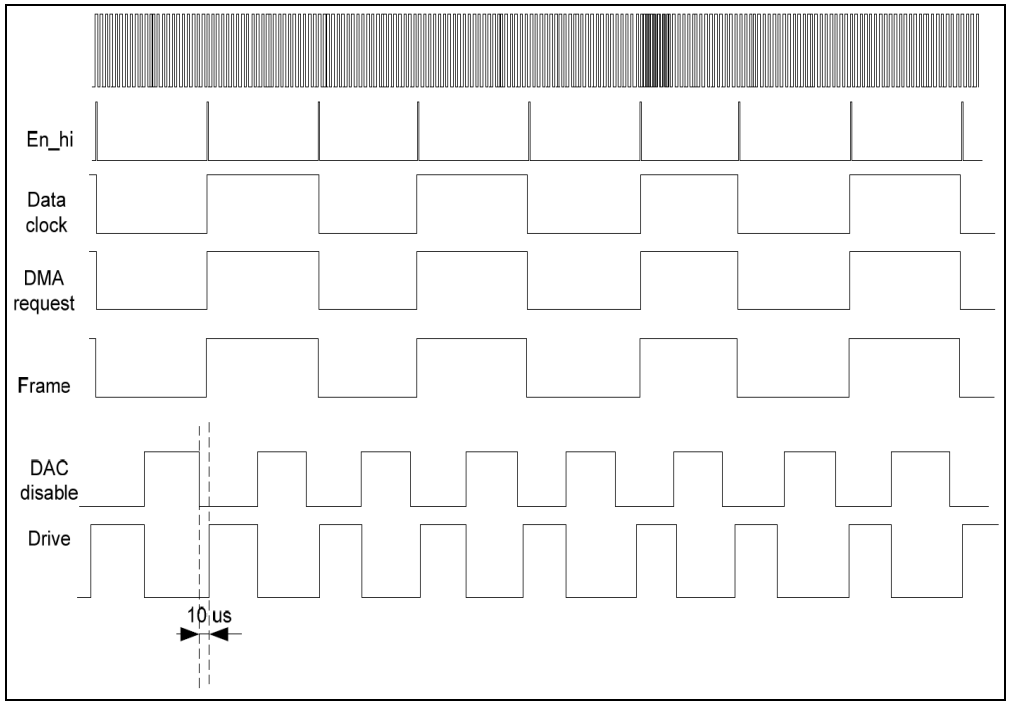
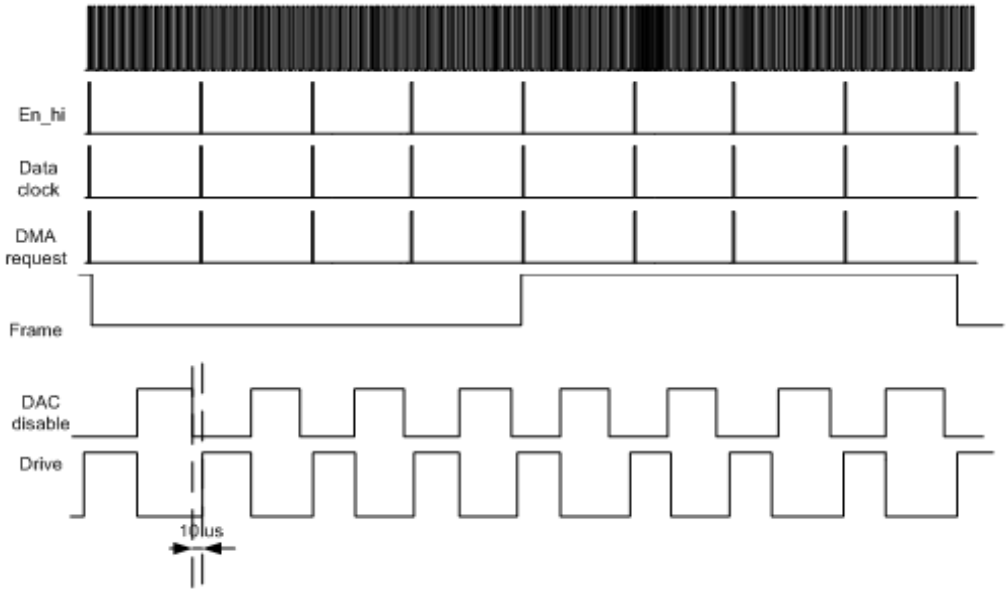


図 4 は、Low Power モード (Type B) での Seg_Display コンポーネントの UDB 生成信号の波形を示しています。

図 4. セグメント ディスプレイ コントロール信号 Type B 波形 (ロー パワー モード)



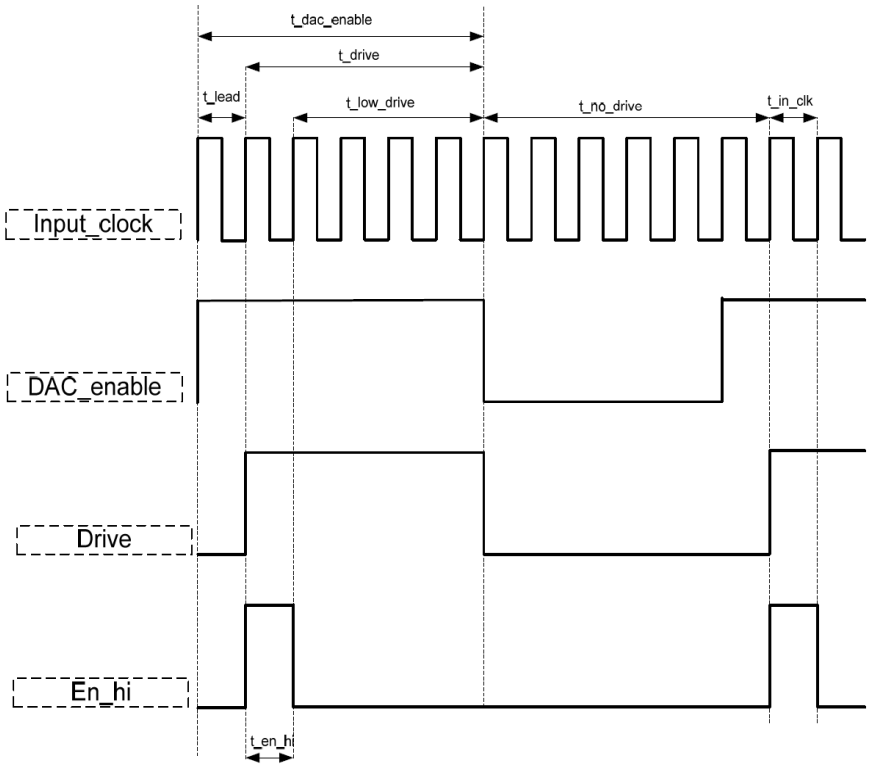
j マルチプレクス レシオ ケースの信号を表示。

タイミングの計算

図 5 と以下の表は、UDB 生成信号のタイミング情報を示しています。図 5 では、3 つの内部信号のみが表示されています。他の信号は前の図をご覧ください。図 5 は、ローパワー モードと Type B 波形に基づいています。

DAC_disable 信号は、内部 DAC_enable 信号を反転することで生成します。

図 5. コントロール信号タイミング図の例



パラメータ	時間	説明	計算
入力クロック周波数	-	入力クロック周波数値 (カスタマイザで自動設定)	f_in = f_frame_rate × N_common_lines × 256 × 2
入力クロック周期	t_in_clk	入力クロック周期を指定します	t_in_clk = 1/(f_in)
High Drive Time (Highドライブ時間)	t_en_hi	HiDrive が有効な時間を指定します	t_en_hi = t_in_clk
	t_lead	LCD DAC 設定時間を指定します	~10 µs
	-	HiDrive 無効時間	t_low_drive (Always Active) t_low_drive + t_no_drive (Low Power)
	t_drive	ドライブ時間を指定します	t_drive = t_en_hi + t_low_drive
	t_drive	ドライブ時間を指定します	t_drive = t_en_hi + t_low_duty_cycle

パラメータ	時間	説明	計算
	t_low_drive	Lowドライブ時間を指定します	t_low_drive = t_drive - t_en_hi
	t_no_drive	ドライブ信号がローにアサートされる時間を指定します	t_no_drive = t_in_clk * 256 - t_drive
	t_dac_enable	LCD DAC がオンである期間を指定します (ローパワー モードのみ)	t_dac_enable = t_drive + t_lead

ユーザ指定の設定

信号タイミングは、コンポーネント カスタマイザでタイミング パラメータを変更することで調整できます。

High Drive Time (High ドライブ時間)

デフォルトでは、Always Active モードの場合は t_en_hi = 128 × t_in_clk、Low Power モードの場合は t_en_hi = 64 × t_in_clk。この値は、カスタマイザで自動的に計算されます。この時間は、最大値まで増加できます。

Always Active モードの場合、HiDriveTimemax = HiDriveTimemin × 253

Low Power モードの場合、HiDriveTimemax = HiDriveTimemin × 247

HiDriveTime 値は、入力クロックの 1 サイクルに従ってインクリメントで増加します。この結果、en_hi 信号の有効時間が延長されます。

ブロックダイアグラムと設定

図 6. セグメント ディスプレイ コンポーネント回路図

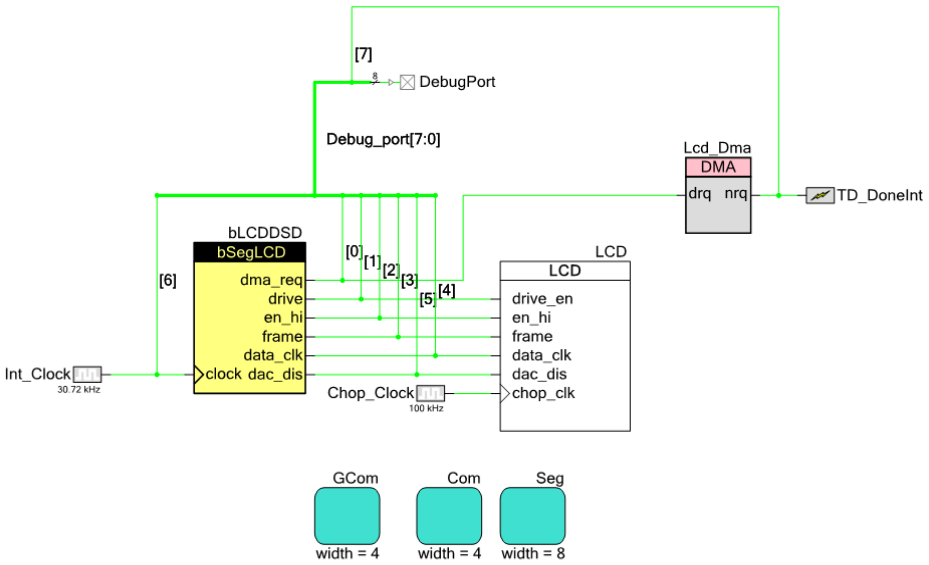
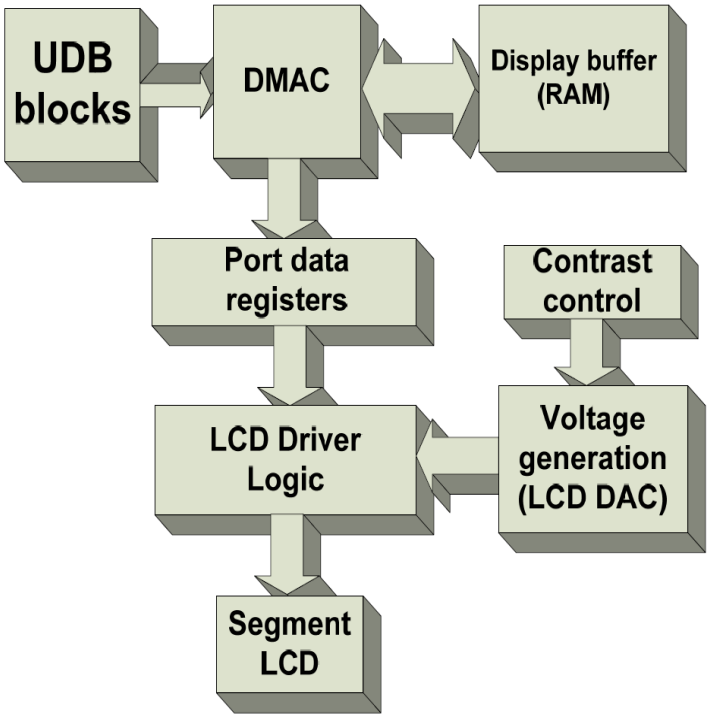


図 6 は、セグメント ディスプレイ コンポーネントの内部回路図です。これは、基本セグメント ディスプレイ コンポーネント、LCD コントロール ブロック (LCD) コンポーネント、DMA コンポーネント、2 つの LCD ポート、1 つのデジタル ポート、1 つの ISR コンポーネント、2 つのクロックで構成されています。

- 基本セグメント ディスプレイ 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 ルーティングに対応します。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。
- DebugPort はデバッグ目的でのみ使用します。このコンポーネントはデフォルトで除去されています。

トップ レベル アーキテクチャ

図 7. セグメント ディスプレイ トップ レベル



レジスタ

Seg_Display_CONTRAST_CONTROL

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	コントラスト レベル						

- コントラスト レベル: 上記のバイアス電圧レベル。

Seg_Display_LCDDAC_CONTROL

ビット	7	6	5	4	3	2	1	0
値	予約済み					DAC 無効	バイアス選択	

- バイアス選択:バイアスを選択。
- DAC 無効:コントラスト コントロールが不要な場合は LCD DAC を無効にします。

Seg_Display_DRIVER_CONTROL

ビット	7	6	5	4	3	2	1	0
値	予約済み					反転	Io2	スリープ モード

- スリープ モード:セグメント ディスプレイにスリープ モードを設定します。
- Io2: LCD ドライバ ブロックの IoDrive モードのハイ電流モードを有効 にします。
- 反転:設定すると、セグメント ディスプレイのすべてのデータを反転します。

Seg_Display_CONTROL

ビット	7	6	5	4	3	2	1	0
値	予約済み						コントロール リセット	クロック イネーブル

- クロックイネーブル: 上記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。
- コントロール リセット: コンポーネントのデジタル部分の初期リセットを行います。

Seg_Display_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。

DC 電気的特性と AC 電気的特性

5.0-V/3.3-V DC と AC 電気的特性

パラメータ	説明	条件	Min	標準値	Max	単位
入力電圧範囲	LCD動作電流		–	–	–	µA
	LCD バイアス範囲		2	–	5.2	V
	LCD バイアス ステップ サイズ		–	25.2	–	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
	セグメント ドライバあたりの Iout					
	ストロング(Strong)駆動		120	160	200	µA

パラメータ	説明	条件	Min	標準値	Max	単位
	弱い(Weak)駆動		–	0.5	–	µA
	弱い(Weak)駆動2		–	1	–	µA
	駆動なし		–	2	–	µA
	コモン ドライバあたりの Iout					
	ストロング(Strong)駆動		160	220	300	µA
	弱い(Weak)駆動		–	11	–	µA
	弱い(Weak)駆動2		–	22	–	µA
	駆動なし		–	<25	–	µA

コンポーネントの変更

バージョン	変更の説明	変更の理由 / 影響
1.0.a	データ シートの訂正	

バージョン 1.0 はセグメント ディスプレイ コンポーネントの最初のリリースです。

注 セグメント ディスプレイは PSoC 5 シリコン リビジョンしかサポートしません。PSoC 3 Production シリコン リビジョンまたはそれ以上には Segment LCD バージョン 3.0 を使用。

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提供することは、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提供することは、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。