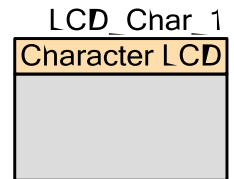


文字 LCD

1.50

特長

- 業界標準の Hitachi HD44780 LCD ディスプレイ ドライバ チップ制御を実装します。
- 1 つの I/O ポートで 7 つの I/O ピンのみを必要とします。
- ユーザ定義のカスタム文字を作成する、文字エディタを内蔵しています。
- 水平および垂直棒グラフを利用できます。



概要説明

文字 LCD コンポーネントには、Hitachi 44780 標準 4 ビット インターフェースに従う、1、2、または 4 つのライン LCD モジュールを簡単に使用できる、一連のライブラリ ルーチンが含まれています。コンポーネントは、水平および垂直棒グラフを実装する API を提供します。または、独自のカスタム文字を作成し、表示できます。

文字 LCD を使用する場合

製品のユーザにテキスト データを表示する、または設計やデバッグ中に開発者にテキスト データを表示する場合に、文字 LCD コンポーネントを使用します。

入出力接続

ここでは、文字 LCD のさまざまな入出力接続について説明します。

LCD_Port – ピン エディタ

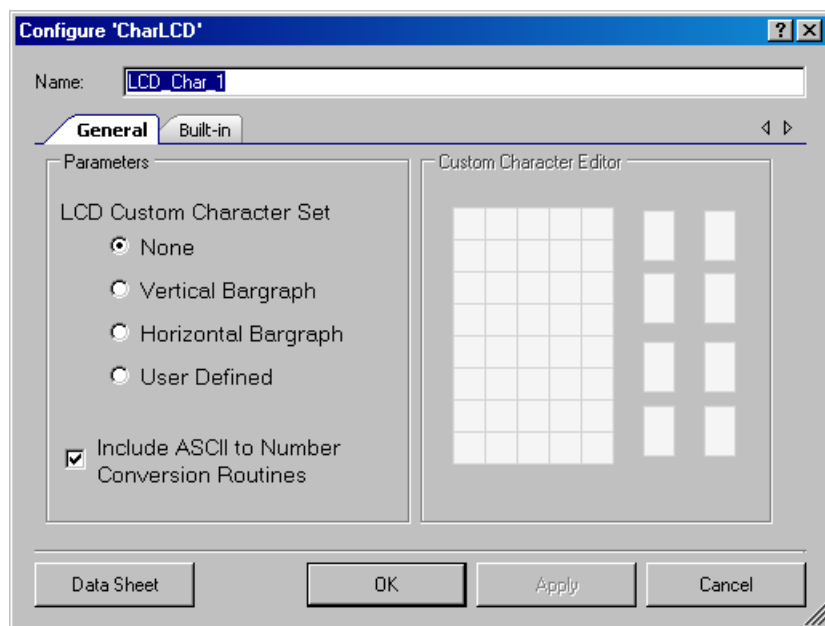
LCD は、物理ポートの 7 つの連続したピンを使用します。希望するポートに文字 LCD を配置するには、設計全体リソースピン エディタを使用します。ピン エディタを使うと、このコンポーネントのデジタルポートを、空いているどの出力ポートにでも配置できます。

注 選択したポートのピン 1 または 0 で開始するために、7 つのピンを配置できますが、ポートをスパンしない場合があります。これらのピンは、LCD ポートにのみ使用され、他の目的には利用できません。

ソフトウェア API がすべての読み取りと書き込みを処理するため、文字 LCD のポートへの直接アクセスは必要ありません。LCD モジュールと PSoC 論理ポート間のピン接続は、「機能の説明」で詳しく解説します。

パラメータおよびセットアップ

文字 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。



パラメータ

LCD Custom Character Set (LCD カスタム文字セット)

次のオプションを選択できるパラメータ:

- None (なし、デフォルト) - カスタム文字については何もしない。
- Vertical Bargraph (垂直棒グラフ) – カスタム文字と API を生成し、垂直棒グラフを操作します。
- Horizontal Bargraph (水平棒グラフ) – カスタム文字と API を生成し、水平棒グラフを操作します。
- User Defined (ユーザ定義) – カスタム文字と API を作成して、操作します。

コンポーネントが文字をロードしたら、LCD_Char_PutChar() 関数とカスタム文字定数 (ヘッダファイルから) を使用して、文字を表示できます。

変換ルーチン

[Include ASCII to Number Conversion Routines (数字変換ルーチンに ASCII を含める)] オプションを選択すると、生成されるコードに複数の API 関数が追加されます。(API テーブルと関数の説明については、これらのルーチンの説明を参照してください。)

Custom Character Editor (カスタム文字エディタ)

[Custom Character Editor (カスタム文字エディタ)] を使うと、GUI を使って、ユーザ定義の文字セットを簡単に作成できるようになります。8 文字は、それぞれ 5x8 ピクセルまでのサイズが可能です。一部のハードウェアでは、5x7 を超えるピクセルは表示できない場合があります。

[Custom Character Editor (カスタム文字エディタ)] を使用するには、**[LCD Custom Character Set (LCD カスタム文字セット)]** のオプションとして、「User Defined (ユーザ定義)」を選択します。次に、編集する文字のサムネイルをクリックします。

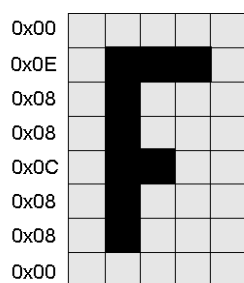
文字のピクセルを切り替えるには、拡大文字ビューで、選択されているピクセルをクリックします。複数のピクセルをクリックしてドラッグし、切り替えることもできます。



カスタム文字セットを作成したら、GUI が 8 つのカスタム文字のルックアップ アレイを生成します。これで、ルックアップ アレイを LCD モジュールにロードできます。デフォルトで、選択または作成された場合に、Start() ルーチンがカスタム文字をロードします。

次の図は、8 バイトのカスタム文字ルックアップ アレイ ローにエンコードされたカスタム文字を示しています。

カスタム文字エンコーディング



Custom character «F»:

```
{0x00, 0x0E, 0x08, 0x08, 0x0C, 0x08, 0x08, 0x08, 0x00}
```

図に示されているように、文字の各ローは、単一のバイトとしてエンコードされます。ここでは、5 つの最下位ビットのみが使用されます。最初の文字の上位ローは、カスタム フォント アレイの最初のバイトでエンコードされます。最初の文字の次のローは、アレイの 2 バイト目です。2 番目のローの最初のローは、アレイの 8 番目のバイト、というように続きます。カスタム フォント アレイ全体は、合計のアレイ サイズが 64 バイトとなる、8 つのカスタム文字から構成されます。

クロック選択

該当なし

配置

該当なし これは、ソフトウェア コンポーネントです。



リソース

構成	デジタル ブロック	API メモリ (バイト数)		ピン (外部入出力ごと)
		Flash	RAM	
なし	0	685	3	7
垂直	0	1276	3	7
水平	0	1276	3	7
ユーザ定義	0	823	3	7
なし + 変換ルーチン	0	932	3	7

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。次の表に、各機能に対するインタフェースを「include」ファイルによって提供される関連定数とともに示します。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_Char_1」を、特定のプロジェクト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固有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機能名、変数、定数記号の接頭辞になります。分かりやすいよう、次の表では、インスタンス名「LCD_Char」を使用しています。

機能	説明
LCD_Char_Start	モジュールを開始し、カスタム文字セットが定義されている場合は、それを LCD にロードします。
LCD_Char_Stop	LCD をオフにします。



機能	説明
LCD_Char_DisplayOn	LCD のモジュール ディスプレイをオンにします。
LCD_Char_DisplayOff	LCD のモジュール ディスプレイをオフにします。
LCD_Char_PrintString	文字ごとに、Null 終端された文字列を画面に表示します。
LCD_Char_PutChar	現在の位置で、単一の文字を LCD モジュール データ レジスタに送信します。
LCD_Char_Position	カーソルを提供されたローとコラムに一致する位置に設定します。
LCD_Char_WriteData	1 バイトのデータを LCD モジュール データ レジスタに書き込みます。
LCD_Char_WriteControl	LCD モジュール制御レジスタに単一のバイト命令を書き込みます。
LCD_Char_ClearDisplay	LCD モジュール画面からデータをクリアします。
LCD_Char_IsReady	レディ ビットが設定されるまで、LCD をポーリングします。
LCD_Char_Sleep	コンポーネントのスリープ モードを準備します。
LCD_Char_Wakeup	コンポーネント構成を復元し、LCD をオンにします。
LCD_Char_Init	コンポーネントの標準動作に必要な初期化を実行します。
LCD_Char_Enable	ディスプレイをオンにします。
LCD_Char_SaveConfig	提供されている API を空にして、スリープ モードに入る前に必要なデータを保存します。
LCD_Char_RestoreConfig	提供されている API を空にして、スリープ モードを終了した後で保存されたデータを復元します。

次のオプション関数は、カスタム フォント タイプを選択する必要がある場合のために含まれています。LCD_Char_LoadCustomFonts() 関数は、すべてのカスタム フォント セット、ユーザ定義または PSoC Creator が生成したフォントセットに含まれます。

LCD_Char_LoadCustomFonts() 関数は、ユーザ定義または棒グラフ文字を LCD ハードウェアにロードするために使用できます。ツールによって作成されたカスタム フォントをロードする場合は、この関数を使う前に、カスタム フォントへのポインタをプロジェクトにインポートする必要があります (LCD_Char_LoadCustomFonts() の説明を参照)。棒グラフの描画コマンドは、

棒グラフが選択され、棒グラフの簡易および動的な調整が有効になっている場合に生成されます。

オプションのカスタム フォント関数	説明
LCD_Char_LoadCustomFonts	LCD モジュールに、カスタム文字をロードします。
LCD_Char_DrawHorizontalBG	水平棒グラフを描画します。棒グラフ文字セットが選択されている場合にのみ利用できます。
LCD_Char_DrawVerticalBG	垂直棒グラフを描画します。棒グラフ文字セットが選択されている場合にのみ利用できます。

次のオプション関数は、必要な場合のために含まれています。

オプションの数の ASCII 変換ルーチン	説明
LCD_Char_PrintInt8	文字 LCD モジュールに、8 ビット値の 2 つの ASCII 文字 16 進数値を印刷します。
LCD_Char_PrintInt16	文字 LCD モジュールに、16 ビット値の 4 つの ASCII 文字 16 進数値を印刷します。
LCD_Char_PrintNumber	16 ビット値の小数值を、左揃えの ASCII 文字として印刷します。

void LCD_Char_Start(void)

説明	この関数は、次のように、LCD ハードウェア モジュールを初期化します。 <ul style="list-style-type: none">• 4ビット インターフェイスを有効にする• ディスプレイをクリアする• カーソルの自動増分を有効にする• カーソルを開始位置にリセットする また、カスタマイザの GUI で定義されている場合は、カスタム文字セットを LCD にロードします。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_Stop(void)

説明	他のコンポーネントとの統一性を保つために提供されています。この API は関数を持ちません。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_PrintString(char8 * string)

説明	現在のカーソル位置から、Null 終端された文字列を画面に書き込みます。
パラメータ	char8 * string: LCD モジュールの画面に、Null 終端された ASCII 文字のアレイが表示されます。
戻り値	なし
副作用	なし

void LCD_Char_PutChar(char8 character)

説明	現在のカーソル位置において、個々の文字を画面に書き込みます。名前付き値を通して、カスタム文字を表示するために使用されます。(LCD_Char_CUSTOM_0() through LCD_Char_CUSTOM_7()).
パラメータ	char8 character: LCD モジュールの画面に表示される ASCII 文字。
戻り値	なし
副作用	なし

void LCD_Char_Position(uint8 row, uint8 column)

説明	カーソルを、引数ローおよびコラムで指定された位置に移動します。
パラメータ	uint8 row: カーソルの位置を示すロー番号。最小値は 0 です。 uint8 column: カーソルの位置を示すコラム番号。最小値は 0 です。
戻り値	なし
副作用	なし

void LCD_Char_WriteData(uint8 dByte)

説明	現在の位置で、データを LCD RAM に書き込みます。書き込みの完了時に、指定した入力モードにより、位置が増分または減分されます。
パラメータ	dByte: LCD モジュールに書き込まれるバイト値。
戻り値	なし
副作用	なし

void LCD_Char_WriteControl(uint8 cByte)

説明 LCD モジュールにコマンド バイトを書き込みます。異なる LCD モデルは、それぞれ独自のコマンドを持つことができます。そのモデルに有効なコマンドについては、特定の LCD データシートをご覧ください。

パラメータ cByte: LCD モジュールのコマンド レジスタにロードされるコマンドを表す 8 ビット値。下の表で指定されている有効なコマンド パラメータ:

値	説明
LCD_Char_CLEAR_DISPLAY	ディスプレイをクリアする
LCD_Char_RESET_CURSOR_POSITION LCD_Char_CURSOR_HOME	カーソルと LCD をホーム位置に戻す
LCD_Char_CURSOR_LEFT	左カーソルの移動方向を設定する
LCD_Char_CURSOR_RIGHT	右カーソルの移動方向を設定する
LCD_Char_DISPLAY_CURSOR_ON	ディスプレイとカーソルを有効にする
LCD_Char_DISPLAY_ON_CURSOR_OFF	ディスプレイとカーソルをオフにする
LCD_Char_CURSOR_WINK	ディスプレイを有効にし、カーソルをオフにして、カーソル ウィンクを設定する
LCD_Char_CURSOR_BLINK	ディスプレイとカーソルを有効にし、カーソル ブリンクを設定する
LCD_Char_CURSOR_SH_LEFT	カーソルをディスプレイの左にシフトする
LCD_Char_CURSOR_SH_RIGHT	カーソルをディスプレイの右にシフトする
LCD_Char_DISPLAY_2_LINES_5x10	ディスプレイを 2 行、10 文字に設定する

戻り値 なし

副作用 なし

void LCD_Char_WriteControl(uint8 command)

説明	入力引数、コマンドを LCD の制御レジスタに書き込む 文字 LCD ヘッダ ファイルには、複数のコマンドが含まれます。そのモデルに有効なコマンドについては、特定の LCD データシートをご覧ください。
パラメータ	command: LCD モジュールのコマンド レジスタにロードされる有効なコマンドを示す、8 ビット値。
戻り値	なし
副作用	なし

void LCD_Char_ClearDisplay(void)

説明	画面の内容をクリアし、カーソル位置をローおよびコラム 0 にリセットします。適切な引数で LCD_Char_WriteControl() を呼び出し、ディスプレイを有効にします。
パラメータ	なし
戻り値	なし
副作用	カーソルの位置を 0,0 にリセットします。

void LCD_Char_IsReady(void)

説明	レディ ビットが設定されるまで、LCD をポーリングします。
パラメータ	なし
戻り値	なし
副作用	ピンを High-Z に変更します。

void LCD_Char_DisplayOff(void)

説明	このマクロは、ディスプレイをオフにしますが、LCD モジュールをリセットすることはありません。適切な引数で LCD_Char_WriteControl() を呼び出し、ディスプレイを無効にします。
パラメータ	なし
戻り値	なし
副作用	なし



void LCD_Char_DisplayOn(void)

説明	このマクロは、初期化なしでディスプレイをオンにします。適切な引数で LCD_Char_WriteControl() を呼び出し、ディスプレイを有効にします。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_Sleep(void)

説明	<p>これは、コンポーネントのスリープを準備するための、好ましいルーチンです。LCD_Char_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_Char_Stop() 関数と LCD_Char_SaveConfig() を呼び出して、ハードウェア構成を保存します。</p> <p>CyPmSleep() または CyPmHibernate() 関数を呼び出す前に、LCD_Char_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。</p>
パラメータ	なし
戻り値	なし
副作用	コンポーネントのピン ドライブ モードは変更しません。この目的には、ポート コンポーネントの API を使用してください。文字 LCD は、独自のプロトコルを持つインターフェース コンポーネントであるため、コンポーネントのピン状態を保存または復元した後で、コンポーネントの再初期化が必要になります。

void LCD_Char_Wakeup(void)

説明	コンポーネント構成を復元し、LCD をオンにします。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_Init(void)

説明	コンポーネントの標準動作に必要な初期化を実行します。また、[Configure (設定)] ダイアログで定義されている場合は、LCD_Char_Init() がカスタム文字セットをロードします。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_Enable(void)

説明	ディスプレイをオンにします。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_SaveConfig(void)

説明	提供されている API を空にして、スリープモードに入る前に必要なデータを保存します。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_RestoreConfig(void)

説明	提供されている API を空にして、スリープ モードを終了した後で保存されたデータを復元します。
パラメータ	なし
戻り値	なし
副作用	なし

void LCD_Char_LoadCustomFonts(const uint8 * customData)

説明	この関数は、8 つのカスタム文字 (棒グラフまたはユーザ定義のフォント) を LCD モジュールにロードし、ランタイム中にカスタム フォントを使用します。カスタム文字設定がカスタマイザで選択されている場合にのみ、利用可能です。
パラメータ	Const uint8 * customData: バイト アレイの先頭へのポインタ。5x8 文字には文字につき 8 バイトが必要となるため、アレイは 64 バイト長となる必要があります。
戻り値	なし
副作用	LCD モジュールに保存されている可能性のある、これまでのカスタム文字を上書きします。

void LCD_Char_DrawHorizontalBG(uint8 row, uint8 column, uint8 maxCharacters, uint8 value)

説明	水平棒グラフを描画します。水平または垂直棒グラフが選択されている場合のみ、利用可能です。棒グラフの文字は、次の関数を使って、まず LCD にロードする必要があります。 LCD_Char_LoadCustomFonts(customData);
パラメータ	uint8 row: 棒グラフの最初の文字のロー。 uint8 column: 棒グラフの最初の文字のコラム。 uint8 maxCharacters: 棒グラフが消費する文字の総数。ユーザの棒グラフ選択により、高さまたは幅を示します。各文字は、5 ピクセル幅、8 ピクセル高です。 uint8 value: 描画する影付きピクセルの数。棒グラフの総ピクセル長 (高さ) を超えることはできません。
戻り値	なし
副作用	なし



void LCD_Char_DrawVerticalBG(uint8 row, uint8 column, uint8 maxCharacters, uint8 value)

説明 垂直棒グラフを描画します。水平または垂直棒グラフが選択されている場合のみ、利用可能です。棒グラフの文字は、次の関数を使って、まず LCD にロードする必要があります。
LCD_Char_LoadCustomFonts(customData);

パラメータ uint8 row: 棒グラフの最初の文字のロー。
uint8 column: 棒グラフの最初の文字のコラム。
uint8 maxCharacters: 棒グラフが消費する文字の総数。ユーザの棒グラフ選択により、高さまたは幅を示します。各文字は、5 ピクセル幅、8 ピクセル高です。
uint8 value: 描画する影付きピクセルの数。棒グラフの総ピクセル長 (高さ) を超えることはできません。

戻り値 なし

副作用 なし

void LCD_Char_PrintInt8 (uint8 value)

説明 8 ビット値の 2 つの ASCII 文字表現を文字 LCD モジュールに印刷します。

パラメータ uint8 value: 16 進数 ASCII 文字で印刷される 8 ビット値。

戻り値 なし

副作用 なし

void LCD_Char_PrintInt16 (uint16 value)

説明 16 ビット値の 4 つの ASCII 文字表現を文字 LCD モジュールに印刷します。

パラメータ uint16 value: 16 進数 ASCII 文字で印刷される 16 ビット値。

戻り値 なし

副作用 なし



void LCD_Char_PrintNumber(uint16 value)

説明	16 ビット値の小数值を、左揃えの ASCII 文字として印刷します。
パラメータ	uint16 value: 小数として ASCII 文字で印刷される 16 ビット値。
戻り値	なし
副作用	なし

ファームウェア ソースコードの例

PSoC Creator は、[Find Example Project (例のプロジェクトを検索)] ダイアログに、スキーマおよびサンプルコードを含む、数多くの例プロジェクトを提供します。コンポーネント固有の例については、[Component Catalog (コンポーネント カタログ)] またはスキーマの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 (例プロジェクトを検索)」を参照してください。

割り込みサービス ルーチン

該当なし。

機能説明

LCD モジュールは、英数文字と制限通気のカスタム フォント用に、ビジュアル デisplayを提供します。API が必要に応じて PSoC を設定し、標準の Hitachi LCD ディスプレイドライバと PSoC 装置間で簡単なインターフェースを可能にします。

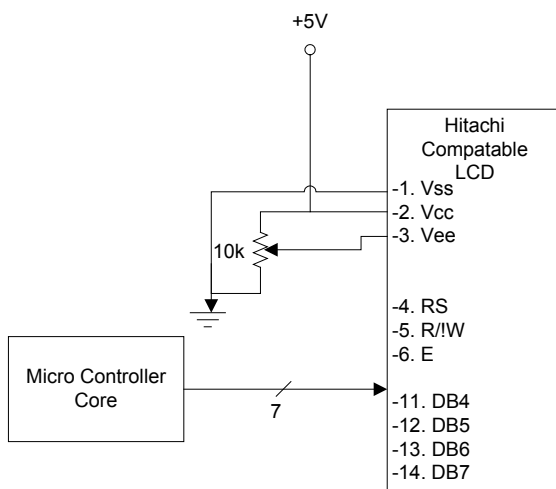
次の表は、物理 LCD モジュール ピンに対する LCD 論理ポート ピンのマッピングを示しています。LCD の論理ポートは、最初または 2 番目のポート物理ピンにマップできますが、ポートを



スパンすることはできないことに注意してください。つまり、論理的に LogicalPort_0 は、ポート 2、ピン 0、またはポート 2、ピン 1 のいずれかになります。書き込みのデータ整列に必要なシフト数を軽減することで、ピン 0 で開始するために LCD 論理ポートを強制するピン エディタの使用が効果的に改善されます。

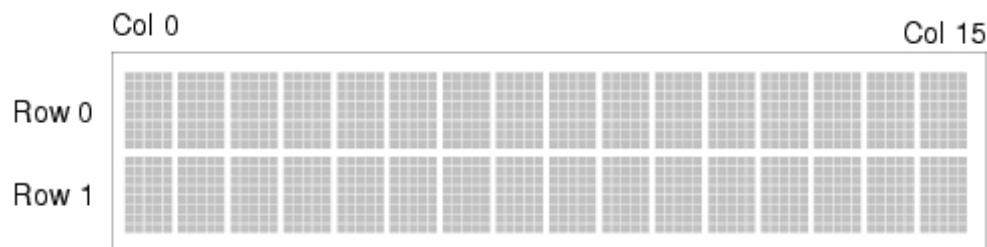
論理ポートピン	LCD モジュールピン	説明
LCDPort_0	DB4	データ ビット 0
LCDPort_1	DB5	データ ビット 1
LCDPort_2	DB6	データ ビット 2
LCDPort_3	DB7	データ ビット 3
LCDPort_4	E	LCD イネーブル (新しいデータが利用可能なことを確認するストローク)
LCDPort_5	RS	レジスタの選択 (データまたは制御入力ビットの選択)
LCDPort_6	R/IW	読み取り/書き込みなし (LCD のレディ ビットをポーリングするための切り替え)

ピン エディタ図



LCD_Char_Position 関数は、次のようにディスプレイのアドレス決定を処理します。ロー 0、コラム 0 は左上にあり、コラム番号は右に向かって増えます。4 行のディスプレイでは、ロー 0、コラム 20 のアドレスをロー 2、コラム 0 にマップするため、ロー 2 のコラム 19 を超える書き込みがロー 2 で破損します。これは、標準の 2x16 Hitachi モジュールでは問題ありません。

2x16 Hitachi LCD モジュール



ブロック図と設定

該当なし。

リファレンス

該当なし。

DC 電気的特性と AC 電気的特性

該当なし。

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
1.50.a	データシートで、LCD_Char_WriteData()、LCD_Char_WriteControl()、LCD_Char_Sleep() API の説明が更新されました。	必要な詳細情報
	データシートのマイナーな編集と更新	
1.50	LCD_Char_Sleep()、LCD_Char_Wakeup()、LCD_Char_Enable()、LCD_Char_Init()、LCD_Char_SaveConfig()、LCD_Char_RestoreConfig() API が追加されました。	これは、低電力モードをサポートし、ほとんどのコンポーネントの共通インターフェースを提供するためです。
	スリープモードの宣言を含む、新しい API ファイル、CharLCD_PM.c が追加されました。	これは、低電力モードをサポートするためです。
	LCD_Char_Init() API に LCD_Char_LoadCustomFonts() への呼び出しが追加されました。	[Configure (設定)] ダイアログで選択したカスタムフォントは、プロジェクトで最初に LCD_Char_Start() が呼び出されたときに自動的にロードされます。
1.40	ディスプレイのタイミング要件を満たすため、LCD_Char_Start() 関数の初期化シーケンスに、遅延が追加されました。	高周波数でシステムを実行する際に存在していた、初期化の失敗を防止します。
1.30	文字 LCD スキーマで、デジタルポートコンポーネントをピンコンポーネントに置換されました。	古いデジタルポートは、使用中止になり、新しいピンコンポーネントに置換されました。
	IsReady 関数の説明が追加されました。	データシートの旧バージョンには、LCD_Char_IsReady() の説明がありませんでした。
	関数 LCD_Char_DelayUS() が削除され、LCD_Char_CyDelay() または LCD_Char_IsReady() に置換されました。	これは、一部の失敗の原因となったコンポーネントのタイミング問題を解決しました。
1.20.a	シリコンリビジョンとの互換性を公表するコンポーネントへの情報が追加されました。	コンポーネントが非互換性のシリコンで使用された場合は、ツールがエラー/警告を報告します。これが発生した場合は、ターゲット装置をサポートするリビジョンに更新してください。

バージョン	変更の説明	変更の理由 / 影響
1.20	記号の更新	企業標準に準拠するよう更新されました。
1.10.a	シリコン リビジョンとの互換性を公表するコンポーネントへの情報が追加されました。	コンポーネントが非互換性のシリコンで使用された場合は、ツールがエラー/警告を報告します。これが発生した場合は、ターゲット装置をサポートするリビジョンに更新してください。
1.10	バージョン 0.2 からのさまざまな更新	0.2 バージョンは、アルファビルドに含まれていましたが、完全な機能性を持つコンポーネントではありませんでした。

© 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はありません。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はありません。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の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ス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ス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。

免責事項: 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、これに限られません。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。

