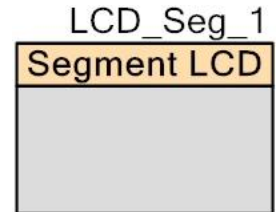


段式 LCD (LCD_Seg)

3.40

特性

- 2 至 768 像素或符号
- 支持 1/3、1/4 和 1/5 偏压
- 10 至 150 Hz 的刷新率
- 介于 2.0 V 至 5.2 V 之间的集成偏压生成，具有高达 128 个数控偏压电平，用于动态对比度控制
- 支持 A 型（标准）和 B 型（低功耗）波形
- 可反转显示像素状态，从而显示负像
- 256 个字节的显示存储器（帧缓冲区）
- 用户定义的像素或符号图，可使用 7 段、14 段或 16 段字符；5x7 或 5x8 点阵；条形图计算子程序。



概述

段式 LCD (LCD_Seg) 组件可在不同的电压电平下直接驱动各种 LCD 显示屏，复用率高达 16 倍。该组件提供一个简易方法来配置 PSoC 器件，以便驱动定制或标准显示屏。

内部偏压生成消除了任何外部硬件方面的需求，并允许进行基于软件的对亮度调整。使用升压转换器，显示屏偏压电压可能高于 PSoC 供电电压。这样会大大提升便携式应用的显示灵活性。

每个 LCD 像素/符号都可以打开或关闭。此外，段式 LCD 组件还提供高级支持，从而简化了显示屏中下列显示结构类型：

- 7 段数字
- 14 段字母数字
- 16 段字母数字
- 5x7 和 5x8 点阵字母数字（对 5x7 和 5x8 使用相同的查找表。查找表中的所有符号大小均为 5x7 像素。）

■ 1 到 255 个元素的条形图

更多有关使用段式 LCD 组件的信息，请参见应用手册 [AN52927: PSoC®3: 段式 LCD 直接驱动基础](#)。

何时使用段式 LCD

在复用率高达 16x 的情况下，如果需要使用不同的电压电平直接驱动各种 LCD 显示屏，应使用直接段式驱动 LCD 组件。直接段式驱动 LCD 组件要求目标 PSoC 器件支持 LCD 直接驱动。

输入/输出接口

在原理图画布上没有可见的组件连接接口；然而，使用“设计范围资源”中的引脚编辑器，可以将各种信号连接至引脚。

参数和设置

将段式 LCD 组件拖入您的设计中，双击该器件，打开 **Configure** 对话框。**Configure** 对话框包含多个不同参数类型的选项卡，用于设置段式 LCD 组件。

Basic Configuration（基本配置）选项卡

The screenshot shows the 'Configure 'SegLCD'' dialog box with the 'Basic Configuration' tab selected. The 'Name' field contains 'LCD_Seg_1'. The 'Number of common lines' is set to 4, and the 'Number of segment lines' is set to 8. A warning message states: 'Decreasing the number of common or segment lines will cause the loss of pixel mapping information for removed lines.' The 'Enable Ganging Commons' checkbox is unchecked. The 'Bias type' is set to '1/3', 'Waveform type' is 'Type A Standard', 'Frame rate, Hz' is '10', and 'Driver Power Mode' is 'No Sleep'. The 'Bias voltage, V' is set to '3.000', with radio buttons for '3.0 V' (selected) and '5.5 V'. At the bottom are buttons for 'Datasheet', 'OK', 'Apply', and 'Cancel'.

Parameter	Value
Name	LCD_Seg_1
Number of common lines	4
Number of segment lines	8
Enable Ganging Commons	<input type="checkbox"/>
Bias type	1/3
Waveform type	Type A Standard
Frame rate, Hz	10
Driver Power Mode	No Sleep
Bias voltage, V	3.000 (3.0 V selected)

Number of common lines (共用线路数量)

定义了显示屏所需的共用信号数（默认值为 4）。

Number of segment lines (段线路数)

定义了显示屏所需的段信号数。取值范围可从 2 到 62。默认值为 8。

Enable Ganging Commons (使能共用信号组连接)

选择该复选框，以将 PSoC 引脚组合在一起以驱动共用信号。为每个共用信号分配两个 PSoC 引脚。这用于驱动较大的显示器。

Bias type (偏压类型)

该值确定共用线路和段线路适当的偏置模式。

Waveform type (波形类型)

通过该参数可确定波形类型：A 型 — 单帧 0 VDC 平均值（默认）或 B 型 — 双帧 0 VDC 平均值。

Frame rate, Hz (帧率 (Hz))

该参数确定了显示器的刷新率。在“无睡眠”模式下，频率取值范围可从 10 Hz 以 10 为步长递增到 150 Hz。默认值为 60 Hz。

在低功耗模式下，帧率选择对于每个配置都有限制并且是唯一的。欲了解有关的详细信息，请参见该数据手册中[功能说明](#)一节的[驱动程序功耗模式](#)部分。

Driver Power Mode (驱动器功耗模式)

该参数定义了组件的功耗模式。以下的各功耗模式设置可用：

- **No Sleep (无睡眠)**：LCD DAC 始终处于打开状态，且芯片将不会进入睡眠模式
- **Low Power using ILO (使用 ILO 的低功耗)**：打开 LCD DAC，但芯片将在电压操作间进入睡眠状态。作为唤醒源时，组件将使用 1 kHz 的内部 ILO。
- **Low Power using Ext 32 kHz crystal (使用外部 32 kHz 晶振的低功耗)**：打开 LCD DAC，但芯片将在电压操作之间进入睡眠状态。作为唤醒源，组件将从 OPPS 定时器中使用 8 K 节拍。

注意：根据所使用的低功耗模式，可使用设置为 1 kHz 的 ILO 或已连接并使能的外部 32 kHz 晶振。可在“设计范围资源”的时钟编辑器中使能 32 kHz 的晶振或设置 ILO 的频率。

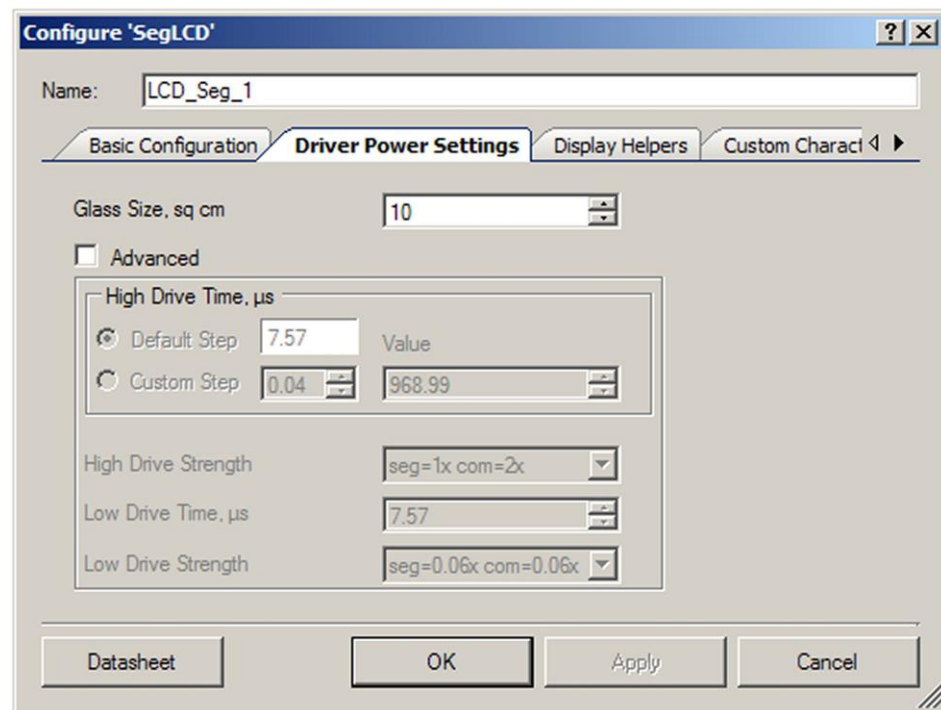


也可参见该数据手册中[功能说明](#)一节的[驱动程序功耗模式](#)部分了解详细信息。

Bias voltage, V (偏压电压 (V))

该参数确定了 LCD DAC 的偏压电压电平。取值范围可从 2.35 V 到 5.5 V 或 2.017 V 到 3 V，这取决于电源。

Driver Power Settings (驱动程序功耗设置) 选项卡



Glass Size, sq cm (显示屏大小 (平方厘米))

使用该字段输入显示屏活动区域的正确大小 (单位为平方厘米)。该值将与 **Number of common lines** (共用线路数) 参数一起使用，以计算正确的电容负载。

Advanced (高级)

如果选定了该复选框，您可手动更改 LCD 的 **Driver Power Settings** (驱动程序功耗设置)。如果未选定该复选框，将根据 **Basic Configuration** (基本配置) 选项卡和 **Glass Size** (显示屏大小) 参数中的选择自动设置驱动程序功耗设置。

High Drive Time, µs (高驱动时间 (µs))

该参数定义了每一次电压操作中高驱动模式的有效时间。

Default Step (默认步骤)

该参数定义针对选定配置的自动计算的高驱动步骤。如果选择了该参数，则高驱动时间值的递增/递减幅度将等于该步骤。

在“无睡眠”模式下，仅有默认高驱动步骤是可选的选项。

Custom Step (自定义步长)

仅在低功耗模式下才可用。它定义了用户可选择的高驱动步长。自定义高驱动步长的最小值取决于主时钟的频率大小。最大值受 **Default Step** (默认步长) 的值限制。

注意：如果更改了 **Frame rate** (帧率)、**Number of common lines** (共用线路数) 或 **Waveform type** (波形类型) 等参数，那么 **High Drive Time** (高驱动时间) 参数将自动被设为帧默认值的一半 (仅在选定了 **Advanced** (高级) 复选框时)：

$$\text{HighDriveTimeValdef} = \text{DefaultStep} * 128 \quad (\text{对于 A 型和 B 型的波形})$$

High Drive Time 最小值的计算在此数据手册后一部分中被定义。在当前的配置中，**High Drive Time** 的最大值为：

$$\text{HighDriveTimeValmax} = \text{DefaultStep} * 253 \quad (\text{对于 A 型和 B 型的波形})$$

通过以下各公式，可计算得出“Default Step” (默认步长) 的值：

$$\text{DefaultStep} = 1 / (\text{numCommons} * \text{FrameRate} * 512 * 1.075) \quad (\text{A 型})$$

$$\text{DefaultStep} = 1 / (\text{numCommons} * \text{FrameRate} * 256 * 1.075) \quad (\text{B 型})$$

上述公式也可将 **Custom Step** (自定义步长) 应用在 **High Drive Time** (高驱动时间) 值中。

如果选定了某个低功耗模式，使用默认高驱动步长时，不建议将 **High Drive Time** 设置为其最大值。使用低功耗模式的效率很低，因为器件将仅进入睡眠模式一小段时间，且在最差的情况下，可能根本不会进入睡眠模式。利用自定义步长时，可更精确地调节 **High Drive Time** 值，这是许多低功耗应用的要求。

High Drive Strength (高驱动强度)

该参数将选择高驱动模式的驱动强度。

Low Drive Time, μs (低驱动时间 (μs))

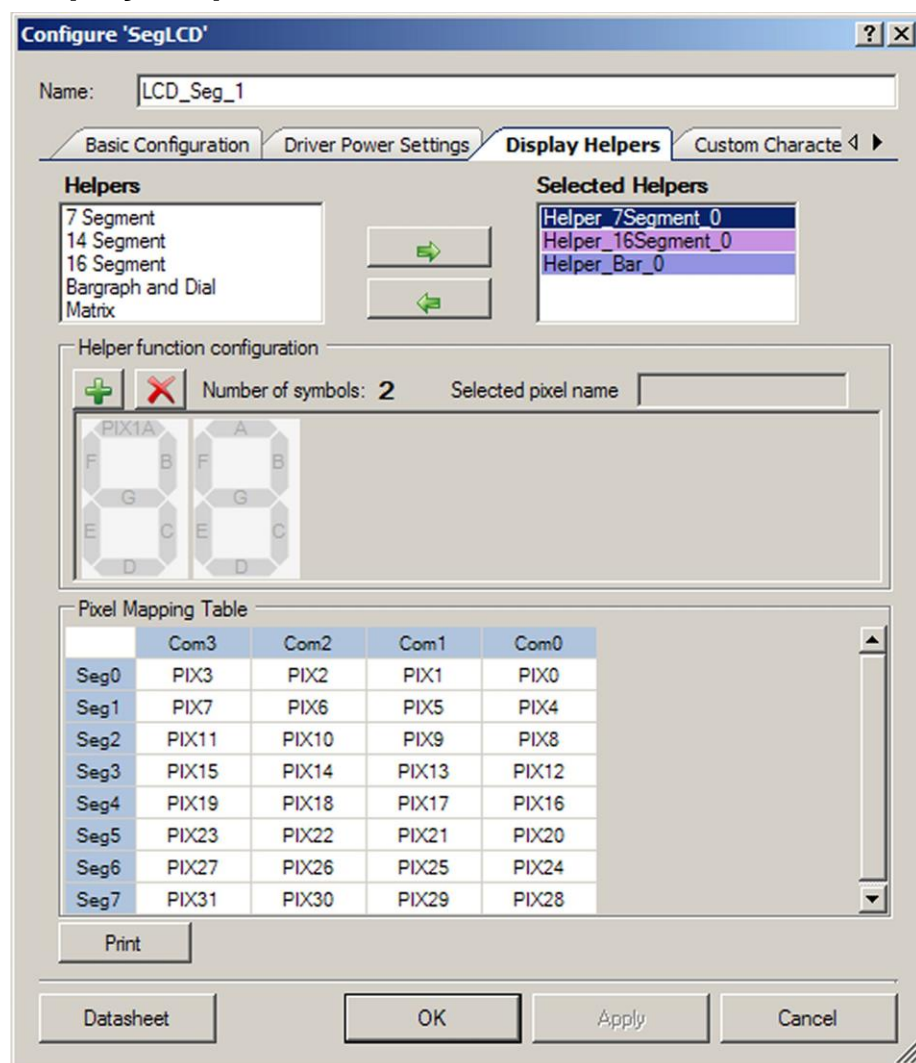
Low Drive Time (低驱动时间) 参数定义了在一电压操作中低驱动模式的有效时间。

Low Drive Strength (低驱动强度)

该参数选择 Low Drive (低驱动) 的驱动强度。**Low Drive Strength** (低驱动强度) 与 **High Drive Strength** (高驱动强度) 相对应，且基于高驱动模式自动设置它。



Display Helpers (显示助手) 选项卡



显示助手允许您配置一组共同使用显示屏段，作为多个预定义显示元素类型之一：

- 7、14 或 16 段显示
- 点阵显示（5x7 或 5x8）
- 线性或圆形条形图显示

基于字符的显示助手可用于将多个显示符号结合起来，以创建多字符的显示元素。

Helpers/Selected Helpers (助手/选定助手)

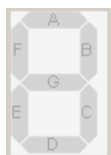
可通过在 **Helpers** (助手) 列表中选择所需的助手类型并单击右箭头按钮, 向 **Selected Helpers** (选定助手) 列表中添加一个或多个助手。如果没有足够的引脚来支持新助手, 则不会添加该助手。要删除助手, 请在 **Selected Helpers** (选定助手) 列表中选择该助手, 并单击左箭头按钮。

注意: 定义任何显示助手前, 一定要为组件设置共模线路和段线路的数量。更改共模线路和段线路的数量前, 必须移除任何已定义的显示助手, 因为您会丢失助手配置的信息。如果试图更改共模线路和段线路的数量, 将显示警告, 指示助手像素映射配置将丢失。

Selected Helpers (选定助手) 在列表中显示的顺序极为重要。默认情况下, 添加到 **Selected Helper** 列表的给定类型的第一个助手名称带后缀 0, 同类型的下一个助手名称带后缀 1, 以此类推。如果已从列表中移除了某个 **Selected Helper**, 将不会重命名剩下的助手。添加助手时, 名称将使用最小的有效后缀。

为每个助手提供了 API。更多有关信息, 请参见[应用编程接口](#)一节中的内容。

- **7 Segment Helper** (7 段助手) — 该助手长度可能为 1 到 5 位, 可显示十六进制位 0 到 F, 或十进制 16 位无符号整数 (uint16) 值。助手功能不支持小数点。



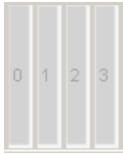
- **14 段助手** — 该助手长度可高达 20 个字符。它可以显示为单一的 ASCII 字符或空结尾的字符串。可能值是标准的 ASCII 可打印字符 (编码范围为 0 到 127)。



- **16 Segment Helper** (16 段助手) — 该助手长度可多达 20 个字符。它可以显示为单一的 ASCII 字符或空结尾的全字符串。可能值是标准的 ASCII 字符和扩展编码表 (编码范围为 0 到 255)。不提供扩展编码表。



- **Bar Graph and Dial Helper** (条形图和拨号助手) — 这些助手用于条形图和刻度盘 (段数为 1 到 255)。条形图可能是单个选定像素或指定像素左侧或右侧的所有像素以及选定像素。



- **Matrix Helper** (矩阵助手) — 此助手支持最多八个字符元素。组件支持 x5x7 或 x5x8 行/列字符。可通过配置两个或更多点阵助手以创建更长的字符串，以定义显示屏的相邻点阵部分。该助手显示一个单一 ASCII 字符或空结尾的字符串。

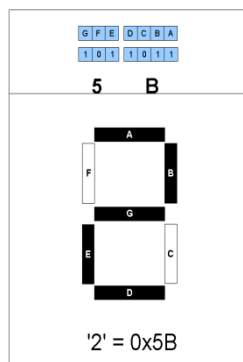


点阵助手具有引脚分布限制。针对矩阵行，点阵助手必须使用 7 到 8 个连续的通用驱动器；针对矩阵列，则必须使用 5 到 40 个连续的段驱动器。该组件支持标准的 Hitachi HD44780 字符集。

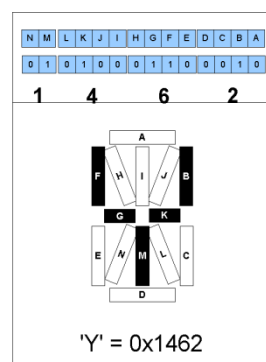
字符编码

所有高级助手 API 都有自己的查找表。该表包含一组编码像素状态，这构成特定的字符映像。以下示例显示的是特定字符的编码方式（段名称可能与 **Configure** 对话框中显示的不同）。

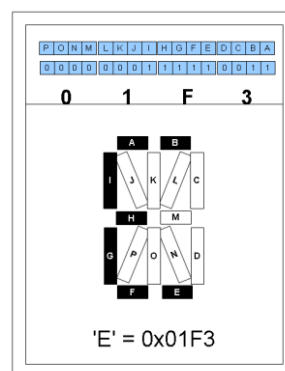
7段编码



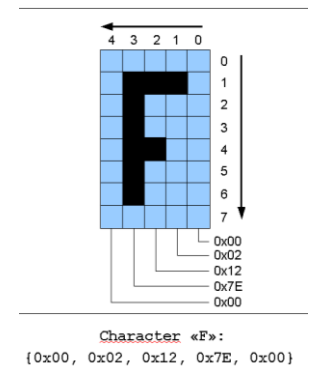
14段编码



16段编码



点阵编码



助手功能配置

对话框的该部分允许您配置助手，包括向/从助手中添加或移除符号，以及命名像素。

1. 从 **Selected Helpers**（选定助手）列表中选择助手。

2. 单击[+]或[x]按钮，为选定助手添加或移除符号。

可添加的最大符号数取决于助手类型和组件支持的总像素数。如果可用引脚数量不足以支持新符号，则不添加新符号。

3. 要想重命名作为助手功能的一部分的像素，请在 **Helper function configuration**（助手功能配置）显示中的符号图像上选择此像素。当前名称将显示在选定像素名称字段中，并可按需要进行修改。

像素命名

默认像素名称的格式为“PIX#”，其中“#”是 **Pixel Mapping Table**（像素映射表）右上角以递增顺序排列的像素数。

与助手符号相关联的像素的默认命名具有不同的格式。默认名称由前缀部分、所有像素的通用符号部分和唯一标识符组成。默认前缀表示助手类型和符号实例。例如，7 段显示助手中某符号中的像素默认名称可能是“H7SEG4_A”，其中：

H7 说明像素是 7 段助手的一部分

SEG4 说明像素是被指定为项目中第四个 7 段符号的符号的一部分

A 指示 7 段符号中的唯一一段

对于默认的像素名称，只有像素名称的唯一部分显示在符号图像中。如果修改了像素名称，则即使修改前和修改后的名称都有共同的前缀，符号图像上也将显示整个名称。

注意：所有像素名称必须是唯一的。

当将助手函数符号元素分配给 **Pixel Mapping Table**（像素映射表）（如下所述）中的一个像素时，该像素使用助手符号元素的名称。助手符号元素名称取代默认像素名称，但不会替换它。无法重新使用与助手功能相关的默认像素名称。

Pixel Mapping Table（像素映射表）

该 **Pixel Mapping Table** 是帧缓冲区的展示。为了使 API 函数正确地工作，**Helper function configuration**（助手函数配置）的每一个像素必须分配给 **Pixel Mapping Table**（像素映射表）中的像素位置。有关进行正确分配所需的信息，请参见你的 LCD 显示屏的数据手册。

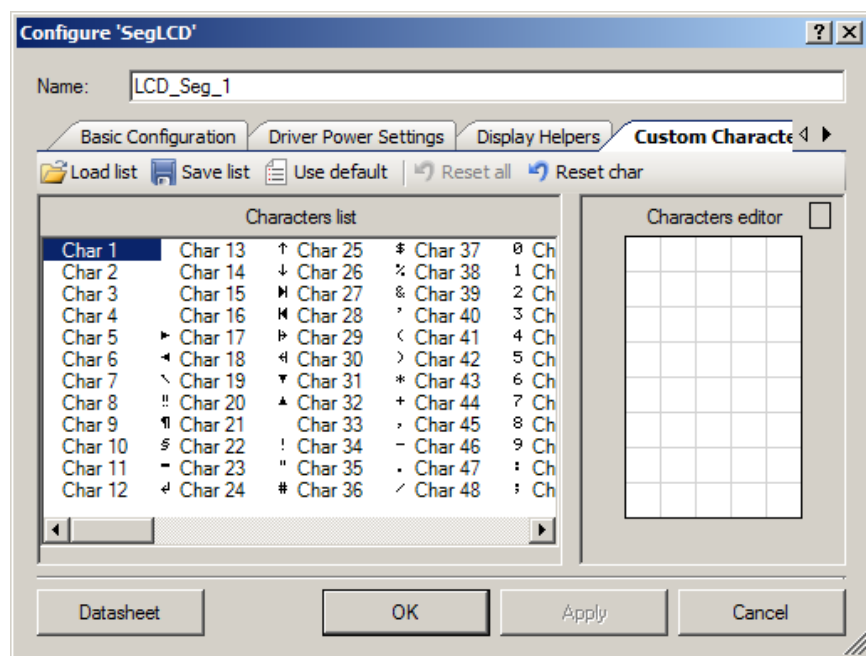
要分配像素，在**助手函数配置**面板中选择所需像素，然后将其拖入**像素映射表**中的正确位置。

可通过双击表中的像素并输入所需的名称，以在 **Pixel Mapping Table** 中重命名像素。该方法可用于对与某可用助手类型不关联的像素进行命名。



Print（打印）按钮用于打印 **Pixel Mapping Table**。

Custom Characters（自定义字符）选项卡



使用该选项卡，您可为 5x8 点阵显示屏创建自定义字符。也可用其将自定义字符查找表作为 XML 字符串进行存储。

默认情况下，**Characters List**（字符列表）字段包含 255 个 ASCII 字符，这些字符具有作为标准 Hitachi HD44780 字符集的映像。可使用 **Character editor**（字符编辑器）访问并修改这些字符。

使用 **Reset char**（复位字符）选项，可将未保存的字符复位到默认映像。使用 **Reset All**（全部复位）选项，可将所有未保存字符复位到标准映像。

保存您自己的字符集后，可使用 **Save List**（保存列表）指令将其另存为 XML 字符串。使用 **Load List**（加载列表）指令，可从 XML 字符串中加载您的字符列表。可使用 **Use default**（使用默认值）选项返回标准字符集。

时钟选择

LCD_Seg 组件使用内部时钟，而不需要外部时钟。一旦安装了组件，时钟将自动专用于 LCD 组件。将自动计算其频率，这取决于共用线路数、刷新率及波形类型。

应用编程接口

通过应用编程接口（API）子程序，您可以使用软件对组件进行配置。下表已列出并说明了每个函数的接口。以下各节将更详细地介绍每个函数。

默认情况下，PSoC Creator 将实例名称“LCD_Seg_1”分配给指定设计中组件的第一个实例。您可将该实例重命名为任意一个符合 PSoC Creator 标识符语法规则的唯一值。实例名称会成为每个全局函数名称、变量和常量符号的前缀。出于可读性考虑，下表中使用的实例名称为“LCD_Seg”。

函数	说明
LCD_Seg_Start()	设置initVar变量，依次调用LCD_Seg_Init()以及LCD_Seg_Enable()函数。
LCD_Seg_Stop()	禁用LCD组件和关联的中断及DMA通道。
LCD_Seg_EnableInt()	使能LCD 中断。如果调用了LCD_Seg_Start(), 则无需此函数
LCD_Seg_DisableInt()	禁用LCD中断。如果调用了LCD_Seg_Stop(), 则无此函数
LCD_Seg_SetBias()	将LCD显示屏的偏压电平设置为最多64个值的其中一个。
LCD_Seg_WriteInvertState()	根据输入参数反转显示屏。
LCD_Seg_ReadInvertState()	返回显示屏反转状态的当前值：正常或反转
LCD_Seg_ClearDisplay()	清除显示屏和关联帧缓冲区RAM。
LCD_Seg_WritePixel()	基于PixelState（像素状态）设置或清除像素
LCD_Seg_ReadPixel()	读取帧缓冲区中像素的状态。
LCD_Seg_Sleep()	停止LCD，并保存用户配置。
LCD_Seg_Wakeup()	恢复并使能用户配置。
LCD_Seg_SaveConfig()	保存LCD配置。
LCD_Seg_RestoreConfig()	恢复LCD配置。
LCD_Seg_Init()	根据Configure对话框设置，初始化或恢复LCD。
LCD_Seg_Enable()	使能LCD。

全局变量

变量	说明
LCD_Seg_initVar	LCD_Seg_InitVar表示段式LCD是否已被初始化。变量将被初始化为0，则在第一次调用LCD_Seg_Start()后它被置为1。这样，第一次调用LCD_Seg_Start()子程序后，器件不用重新初始化仍可重启。 如果需要重新初始化组件，可在LCD_Seg_Start()或LCD_Seg_Enable()函数前先调用LCD_Seg_Init()函数。

uint8 LCD_Seg_Start(void)

说明： 启动LCD组件，然后使能所需的中断、DMA通道、帧缓冲器及硬件。请勿清除帧缓冲器RAM。

参数： 无

返回值： uint8 cystatus: 标准API返回值。

返回值	说明
CYRET_LOCKED	一些DMA TD或某个通道已被使用。
CYRET_SUCCESS	函数已成功完成

其他影响： 该API从时轮配置寄存器2中使能一个‘每秒一个脉冲’位。仅在低功耗32 kHz外部Xtal组件模式下才会出现这种情况。

void LCD_Seg_Stop(void)

说明： 禁用LCD组件和关联的中断和DMA通道。自动清空显示屏以避免直流偏移。不清除帧缓冲器。

参数： 无

返回值： 无

其他影响： 该API不从之前由LCD_Seg_Init() API使能的时轮配置寄存器2中清除每秒一脉冲位。仅在低功耗32 kHz外部Xtal 组件模式下才会出现该情况。

void LCD_Seg_EnableInt(void)

- 说明:** 使能LCD中断。每当更新LCD（TD完成）后都会生成中断。
如果使用PSoC 5LP器件，该API也会使能LCD唤醒中断。当组件在睡眠模式下正常工作时，一般都会调用该函数。
- 参数:** 无
- 返回值:** 无
- 其他影响:** 无

void LCD_Seg_DisableInt(void)

- 说明:** 禁用“所有子帧”及LCD唤醒中断。
- 参数:** 无
- 返回值:** 无
- 其他影响:** 无

uint8 LCD_Seg_SetBias(uint8 biasLevel)

- 说明:** 通过该函数可将LCD显示屏的偏压电平设置为64个值中的一个。实际值数受模拟供电电压Vdda的限制。偏置电压不能超过Vdda。更改偏压电平会影响LCD对比度。
- 参数:** uint8 biasLeve: 显示屏的偏压电平
- 返回值:** uint8 cystatus: 标准API返回值。

返回值	说明
CYRET_BAD_PARAM	biasLevel参数赋值失败。
CYRET_SUCCESS	函数已成功完成。

- 其他影响:** 无

uint8 LCD_Seg_WriteInvertState(uint8 invertState)

- 说明:** 该函数基于输入参数反转显示屏。反转操作在硬件中发生，无需对帧缓冲区中的显示屏RAM进行任何更改。
- 参数:** uint8 invertState: 设置显示的反转状态。
- 返回值:** uint8 cystatus: 标准API返回值。
- 其他影响:** 无



uint8 LCD_Seg_ReadInvertState(void)

说明: 该函数用于返回显示器反转状态的当前值：正常或反转。

参数: 无

返回值: (uint8) invertState: 设置显示屏的反转状态。

其他影响: 无

void LCD_Seg_ClearDisplay(void)

说明: 该函数清除显示屏和关联帧缓冲器RAM。

参数: 无

返回值: 无

其他影响: 无

uint8 LCD_Seg_WritePixel(uint16 pixelNumber,uint8 pixelState)

说明: 该函数基于输入参数“PixelState”设置或清除像素。通过压缩数寻址像素。

参数: uint16 pixelNumber: 帧缓冲器中指向像素位置的封装数组。LSB低位半字节的最低三位是字节中的比特位置，LSB高位半字节低（四位）是复用器行中的字节地址，MSB低位半字节高（四位）是复用器行数。生成组件的“.h”文件包括各像素此格式的#定义。

uint8 pixelState: 指定的pixelNumber被设为该像素状态。

返回值: uint8 status: 根据字节地址和复用行编号范围检查通过或失败。未对该特位置执行检查。

其他影响: 无

uint8 LCD_Seg_ReadPixel (uint16 pixelNumber)

- 说明:

该函数可读取帧缓冲器中的像素状态。通过一个封装好的数据包寻址像素。
- 参数:

uint16: pixelNumber: 帧缓冲区中指向像素位置的压缩数。LSB低位半字节的最低三位是字节中的比特位置，LSB高位半字节低（四位）是复用器行中的字节地址，MSB低位半字节高（四位）是复用器行数。生成组件的“.h”文件包括各像素此格式的#定义。
- 返回值:

uint8 status: 返回指定PixelNumber的当前状态。

值	说明
0x00	像素关闭。
0x01	像素打开。
0xFF	像素未分配。

其他影响:

无

void LCD_Seg_Sleep(void)

- 说明:

这是使组件准备进入睡眠模式时的首选子程序。LCD_Seg_Sleep()子程序保存当前组件的状态。然后，该函数调用LCD_Seg_Stop()函数以及LCD_Seg_SaveConfig()来保存硬件配置。

调用CyPmSleep()或CyPmHibernate()函数前，先调用LCD_Seg_Sleep()函数。有关功耗管理函数的详细信息，请参考PSoC Creator 《系统参考指南》。
- 参数:

无
- 返回值:

无
- 其他影响:

不更改组件引脚的驱动模式。

void LCD_Seg_Wakeup(void)

- 说明:

该函数是将组件恢复到调用LCD_Seg_Sleep()时状态的首选子程序。LCD_Seg_Wakeup()函数将调用LCD_Seg_RestoreConfig()函数，以恢复配置。如果组件在调用LCD_Seg_Sleep()函数前已使能，则 LCD_Seg_Wakeup()函数也会重新使能组件。
- 参数:

无
- 返回值:

uint8 cystatus: 标准API返回值

返回值	说明
CYRET_LOCKED	一些DMA TD或某个通道已被使用
CYRET_SUCCESS	函数已成功完成

其他影响:

如果调用LCD_Seg_Wakeup()函数前未调用LCD_Seg_Sleep()或LCD_Seg_SaveConfig()函数，则可能会产生不可预计的行为。



void LCD_Seg_SaveConfig(void)

说明: 该函数保存组件配置，从而保存非保留寄存器。该函数还将保存当前“Configure”对话框中所定义的或通过相应API修改的组件参数值。通过LCD_Seg_Sleep()函数调用该函数。

参数: 无

返回值: 无

其他影响: 无

void LCD_Seg_RestoreConfig(void)

说明: 该函数恢复组件配置，它将恢复非保留寄存器。它还会将组件参数值恢复为调用LCD_Seg_Sleep()函数前的值。

参数: 无

返回值: 无

其他影响: 如果调用该函数前未调用LCD_Seg_Sleep()或LCD_Seg_SaveConfig()函数，那么可能会产生不可预计的行为。

void LCD_Seg_Init(void)

说明: 根据Configure对话框的设置，初始化或恢复组件参数。无需调用LCD_Seg_Init()，因为LCD_Seg_Start()子程序会调用该函数，且这是开始组件操作的首选方法。配置并使能所有必要的硬件模块，还会清除帧缓冲区。

参数: 无

返回值: 无

其他影响: 根据“Configure”对话框的设置，将所有寄存器设为相应的值。该API从时轮配置寄存器2中使能一个每秒一脉冲位。仅在低功耗32 kHz外部Xtal组件模式下才会出现该情况。

void LCD_Seg_Enable(void)

说明: 使能LCD固定硬件的电源，并使能UDB信号的生成。

参数: 无

返回值: uint8 cstatus: 标准API返回值

返回值	说明
CYRET_LOCKED	一些DMA TD或某个通道已被使用。
CYRET_SUCCESS	函数已成功完成

其他影响: 无

可选助手 API

只有在 Configure 对话框中选中个别助手时，才会提供下列 API。

功能	说明
LCD_Seg_Write7SegDigit_n	将显示7段显示元素阵列上的十六进制数字。
LCD_Seg_Write7SegNumber_n	将显示7段显示元素中1到5位阵列上的整数。
LCD_Seg_WriteBargraph_n	将显示线性或圆形条形图上的整数位置。
LCD_Seg_PutChar14Seg_n	将显示14段字母数字字符显示元素阵列上的字符。
LCD_Seg_WriteString14Seg_n	将显示14段字母数字字符显示元素阵列上空结尾的字符串。
LCD_Seg_PutChar16Seg_n	将显示16段字母数字字符显示元素阵列上的字符。
LCD_Seg_WriteString16Seg_n	将显示16段字母数字字符显示元素阵列上的空结尾字符串。
LCD_Seg_PutCharDotMatrix_n	将显示点阵字母数字字符显示元素阵列上的字符。
LCD_Seg_WriteStringDotMatrix_n	显示点阵字母数字字符显示元素阵列上的空结尾字符串。

注意： 带有后缀“n”的函数名称表示在组件定制器中创建了多个相同符号类型的显示助手。特定显示助手元素由 API 函数控制，函数名称各自带有“n”后缀。

void LCD_Seg_Write7SegDigit_n(uint8 digit,uint8 position)

- 说明：

该函数用来显示7段显示元素阵列中的十六进制数字。数字可以为0到9以及A到F范围内的十六进制值。必须使用定制器显示助手工具定义与7段显示元素相关联的像素集。可在帧缓冲器中定义多个7段显示元素，并可通过函数名称中的后缀（n）对这些元素进行寻址。只有在组件自定义程序中定义了7段显示元素，才包含该函数。
- 参数：

uint8 digit: 要显示为十六进制数字的未分配整数值（范围为0到15）。

uint8 position: 从右侧的0开始从右到左计数的数字位置。如果该位置位于所定义的显示区外，将不显示该字符。
- 返回值：

无
- 其他影响：

无



void LCD_Seg_Write7SegNumber_n(uint16 value,uint8 position,uint8 mode)

说明： 该函数将显示7段显示元素的1到5数字阵列上的16位整数。必须使用定制器显示助手工具定义与7段显示元素相关联的像素集。在帧缓冲区中，可以定义多个7段显示元素组，并可通过函数名称中的后缀（n）对这些元素进行寻址。符号转换、符号显示、小数点和其他自定义特性必须由应用特定的用户代码来进行处理。只有在组件自定义程序中定义了7段显示元素，才包含该函数。

参数： uint16值：显示无符号的整型值。

uint8位置：从右侧的0开始自右向左计算的最低有效位位置。如果所定义的显示区域包含的位数少于值需要的位数，则将不显示最高有效位或多个位。

uint8模式：设置显示模式。可能为0或1。

值	说明
0	不显示前导零。
1	显示前导零

返回值： 无

其他影响： 无

void LCD_Seg_WriteBargraph_n(uint8 location,uint8 mode)

说明： 该函数显示了一个1到255段条形图（从左到右编号）上的8位整数位置。该条形图可为任意用户定义的大小（1到255段）。此外，条形图还可创建为圆形，以显示旋转位置。必须使用定制器显示助手工具定义与条形图显示元素相关联的像素集。可在帧缓冲区中定义多个条形图显示，并可通过函数名称中的后缀（n）对这些显示进行寻址。只有在组件自定义程序中定义了条形图显示元素，才包含该函数。

参数： uint8 location：要显示的无符号整数位置。有效值为0到条形图中的段数。0值关闭所有条形图元素。大于条形图段数的值将会导致所有元素为打开状态。

uint8 mode：设置条形图显示模式。

值	说明
0	打开指定的“位置”段。
1	打开“位置”段及其左侧的所有段。
-1	打开“位置”段及其右侧的所有段。
2至10	显示“位置”段及其右侧的2到10个段。该模式可用于创建较宽的指示符。

返回值： 无

其他影响： 无

void LCD_Seg_PutChar14Seg_n(uint8 character,uint8 position)

说明: 该函数用来显示14段字母数字字符显示元素阵列中的一个8位字符。自定义程序显示助手工具必须用于定义与14段显示元素相关联的像素集。在帧缓冲器中，可以定义多个14段字母数字显示元素组，并可通过函数名称中的后缀（n）对这些元素进行寻址。只有在组件自定义程序中定义了14段显示元素，才包含该函数。

参数: uint8 character: 要显示的字符ASCII值（ASCII值为0至127的可打印字符）

uint8 position: 从左侧的0开始从左到右计数的字符位置。如果此位置不位于所定义的显示区内，将不显示该字符。

返回值: 无

其他影响: 无

void LCD_Seg_WriteString14Seg_n(uint8 const character[], uint8 position)

说明: 该函数可显示14段字母数字字符显示元素阵列上的空结尾字符串。定制器显示助手工具必须用于定义与14段显示元素关联的像素集。在帧缓冲器中，可以定义多个14段字母数字显示元素组，并可通过函数名称中的后缀（n）对这些元素进行寻址。只有在组件自定义程序中定义了14段显示元素，才包含该函数。

参数: *uint8 character: 指向空结尾字符串的指针。

uint8 position: 从左侧的0开始从左到右计数的第一个字符位置。如果字符串的长度超出所定义的显示区大小，将不显示额外字符。

返回值: 无

其他影响: 在数据输出之前不清除显示。不受影响的所有位置仍处于其之前的像素状态。

void LCD_Seg_PutChar16Seg_n(uint8 character, uint8 position)

说明: 该函数可显示16段字母数字字符显示元素阵列上的一个8位字符。必须使用定制器显示助手工具定义与16段显示元素相关联的像素集。在帧缓冲器中，可以定义多个16段字母数字显示元素组，并可通过函数名称中的后缀（n）对这些元素进行寻址。只有在组件自定义程序中定义16段显示元素，才包含该函数。

参数: uint8 character: 显示字符的ASCII值（取值范围为0到255的可打印ASCII和表扩展字符）

uint8 position: 从左侧的0开始从左到右计数的字符位置。如果该位置不位于所定义的显示区内，将不显示该字符。

返回值: 无

其他影响: 无



void LCD_Seg_WriteString16Seg_n(uint8 const character[],uint8 position)

说明: 该函数可显示16段字母数字字符显示元素阵列上空结尾的字符串。必须使用定制器显示助手工具定义与16段显示元素相关联的像素集。在帧缓冲器中，可以定义多个16段字母数字显示元素组，并可通过函数名称中的后缀（n）对这些元素进行寻址。只有在组件自定义程序中定义了16段显示元素时，才包含该函数。

参数: *uint8 character: 指向空结尾字符串的指针。

uint8 position: 从左侧的0开始从左到右计数的第一个字符位置。如果字符串的长度超出所定义的显示区大小，将不显示额外字符。

返回值: 无

其他影响: 在数据输出之前不清除显示。不受影响的所有位置仍处于它们先前的像素状态。

void LCD_Seg_PutCharDotMatrix_n(uint8 character、uint8 position)

说明: 该函数可显示点阵字母数字字符显示元素阵列中的一个8位字符。必须使用定制器显示助手工具定义与点阵显示元素相关联的像素集。在帧缓冲区中，可以定义多个点阵字母数字显示元素组，并可通过函数名称中的后缀（n）对这些元素进行寻址。只有在组件定制器中定义了点阵显示元素时，才包含该函数。

参数: uint8 character: 要显示的字符ASCII值。

uint8 position: 从左侧的0开始从左到右计数的字符位置。如果该位置不位于所定义的显示区内，将不显示该字符。

返回值: 无

其他影响: 无

void LCD_Seg_WriteStringDotMatrix_n(uint8 const character[],uint8 position)

- 说明:

通过此函数可显示点阵字母数字字符显示元素阵列上的空结尾字符串。必须使用定制器显示助手工具定义与点阵显示元素相关联的像素集。在帧缓冲区中，可以定义多个点阵字母数字显示元素组，并可通过函数名称中的后缀（n）对这些元素进行寻址。仅在组件定制器中定义了点阵显示元素时，该函数才可用
- 参数:

*uint8 character: 指向空结尾字符串的指针。

uint8 position: 从左侧的0开始从左到右计数的第一个字符位置。如果字符串的长度超出所定义的显示区大小，将不显示额外字符。
- 返回值:

无
- 其他影响:

在数据输出之前不清除显示。未受影响的所有位置将仍处于其先前的像素状态。

引脚 API

这些 API 函数用于更改段式 LCD 组件所使用引脚的驱动模式。

函数	说明
LCD_Seg_ComPort_SetDriveMode	设置段式LCD组件的共模线路所使用的所有引脚的驱动模式
LCD_Seg_SegPort_SetDriveMode	设置段式LCD组件的段线路使用的所有引脚的驱动模式。

void LCD_Seg_ComPort_SetDriveMode(uint8 mode)

- 说明:

设置段式LCD组件共模线路所使用的所有引脚的驱动模式。
- 参数:

uint8 mode: 所需的驱动模式有关驱动模式的信息，请参考引脚组件数据手册。
- 返回值:

无
- 其他影响:

无

LCD_Seg_SegPort_SetDriveMode(uint8 mode)

- 说明:

设置段式LCD组件的段线路使用的所有引脚的驱动模式。
- 参数:

uint8 mode: 所需的驱动模式有关驱动模式的信息，请参考引脚组件数据手册。
- 返回值:

无
- 其他影响:

无



宏

- LCD_Seg_COMM_NUM — 针对组件目前的配置定义了用户定义显示屏的共模线路的数量。
- LCD_Seg_SEG_NUM — 针对组件目前的配置定义了用户定义显示屏的段线路数量。
- LCD_Seg_BIAS_TYPE — 针对组件目前的配置定义了用户定义显示屏的偏压类型。
- LCD_Seg_BIAS_VOLTAGE — 为用户定义的显示屏定义了默认偏压电压。在初始化过程中，将在 LCDDAC 控制寄存器中设置该值。
- LCD_Seg_FRAME_RATE — 针对组件目前的配置定义了用户定义显示屏的刷新率。
- LCD_Seg_EXTRACT_ROW — 计算帧缓冲区中特定像素的行。
- LCD_Seg_EXTRACT_PORT — 计算帧缓冲区中特定像素的字节偏移。
- LCD_Seg_EXTRACT_PIN — 计算帧缓冲区中特定像素的比特位置。
- LCD_Seg_WRITE_PIXEL — 这是 viod 类型的 LCD_Seg_WritePixel() 函数的宏定义。
- LCD_Seg_READ_PIXEL — 这是 LCD_Seg_ReadPixel() 函数的宏定义。
- LCD_Seg_FIND_PIXEL — 该宏将计算帧缓冲区中的像素位置。它使用定制器像素表中的信息以及专用于 LCD 的物理引脚的相关信息。该宏是像素映射机制的基础。使用帧缓冲区中计算出的像素位置定义像素表中每个像素的名称。API 使用像素名称访问各自的像素。

MISRA 合规性

本节介绍了 MISRA-C:2004 规范以及本组件的偏差情况。有两种偏差类型，如下定义：

- 项目偏差 — 适用于所有 PSoC Creator 组件的偏差
- 特定偏差 — 仅适用于该组件的偏差

本节介绍了有关组件特定偏差的信息。《系统参考指南》的“MISRA 合规性”章节中介绍项目偏差以及有关 MISRA 合规性验证环境的信息。

段式 LCD 组件具有以下特定偏差:

MISRA-C: 2004规则	规则类别 (必须/建议)	规则说明	偏差描述
8.7	必须 (R)	如果仅可从单个函数中访问对象, 则应在模块范围内定义这些对象。	以下对象在C文件范围内被定义为静态常量: SegLCD_charDotMatrix[]、SegLCD_7SegDigits[]、SegLCD_16SegChars[]、SegLCD_14SegChars[]。 这些阵列拥有不同助手的查找表。每个助手使用其类型的查找表。根据配置, 助手可有将共享其访问查找表的多个实例。
10.1	必须 (R)	在下面情况下, 整型表达式的数值不应隐式转换为不同的底层类型。 a) 没有转换到相同符号的更宽的整数类型, b) 表达式是复合的, 或 c) 表达式不是一个常量, 而是函数参数, 或 d) 表达式不是一个常量, 而是返回表达式。	DMA组件提供了各个通用整数类型的定义。
13.2	建议 (A)	对一个非零值进行显性测试, 除非操作数是有效的布尔 (Boolean) 值。	DMA组件提供了对各个通用整数类型的定义。通过对这些定义进行“或”(OR) 运算, 可以得到各函数的正确参数。
13.7	必须 (R)	不允许结果不变的布尔运算。	当该条形图助手中所使用的像素数量小于2时, 将应用于LCD_Seg_WriteBargraph_n()。在这种情况下, 该函数的实现将使用一个输入参数, 以检查引起不变的布尔运算的机制。这样会导致不访问的代码。
14.1	必须 (R)	没有不可访问的代码。这些代码是指在任何情况下都不会被执行的代码。	当该条形图助手中所使用的像素数量小于2时, 将应用于LCD_Seg_WriteBargraph_n()。在这种情况下, 该函数的实现将使用一个输入参数, 以检查引起不变的布尔运算的机制。它会导致不访问的代码。
19.7	建议 (A)	使用时, 函数应优先于类函数宏。	以下各宏用于提高性能: SegLCD_FIND_PIXEL()、 SegLCD_EXTRACT_ROW()、

MISRA-C: 2004规则	规则类别 (必须/建议)	规则说明	偏差描述
			SegLCD_EXTRACT_PORT()、 SegLCD_EXTRACT_PIN()。 宏SegLCD_WRITE_PIXEL()映射到返回‘void’的SegLCD_WritePixel()。 该宏用于组件内部，不使用其返回值。 宏SegLCD_READ_PIXEL()直接映射到SegLCD_ReadPixel()函数。
21.1	必须 (R)	应至少通过使用以下各项中的一个，以确保最大程度地减少运行时发生的故障： a)静态分析工具/技术 b)动态分析工具/技术 c) 用于处理运行时发生故障的 检查的明确编码	当为单个数字显示生成函数 LCD_Seg_Write7SegDigit_n()和 LCD_Seg_Write7SegNumber_n() 时，这两个函数具有冗余编码。 当为3个以下像素生成函数 LCD_Seg_WriteBargraph_n()时，该 函数具有冗余编码。

该组件有以下嵌入式组件：DMA 和中断。MISRA 合规性与特定偏差的相关信息，请参见相应组件数据手册。

示例固件源代码

在“Find Example Project”对话框中，PSoC Creator 提供了大量的示例项目，包括原理图和代码。要获取组件特定的示例，请打开器件目录中的对话框或原理图中的器件实例。要查看通用示例，请打开‘Start Page’（起始页）或 **File**（文件）菜单中的对话框。根据要求，可以通过使用对话框中的 **Filter Options**（滤波器选项）项来限定可选的项目列表。

更多有关信息，请参考《PSoC Creator 帮助》部分中主题为“查找示例项目”中的内容。

注意：由于特定的组件实现，当组件处于低功耗模式下时，您不可使用带 CyPmSaveClocks()和 CyPmRestoreClocks() API 的组件。在其他情况下，则可使用 CyPmSaveClocks()和 CyPmRestoreClocks() API。

功能说明

段式 LCD 组件提供了既强大又灵活的机制，以驱动不同类型的 LCD 显示屏。使用配置对话框访问可用于定制组件功能的参数。可使用一组标准的 API 子程序控制显示屏和特定像素。根据所定义的显示助手的类型和数量，生成其他显示屏 API。

默认配置

LCD_Seg 组件的默认配置提供了通用的 LCD 直接段驱动控制器。默认 LCD_Seg 配置为：

- 4 个共用线路
- 8 个段线路
- 1/3 偏压类型
- 60 Hz 刷新率
- “无睡眠” 功耗模式
- 3 V 偏压电压
- 显示屏大小：10 平方厘米
- 968.99 μ s 高驱动时间
- seg - 1x、com - 2x 高驱动强度
- 7.57 μ s 低驱动时间
- seg - 0.06x、com - 0.12x 低驱动强度
- 未定义显示助手。默认的 API 生成不包含任何所支持的显示元素的函数。

自定义配置

段式 LCD 组件的一个关键功能就是为不同特性和布局的 LCD 提供灵活的支持。

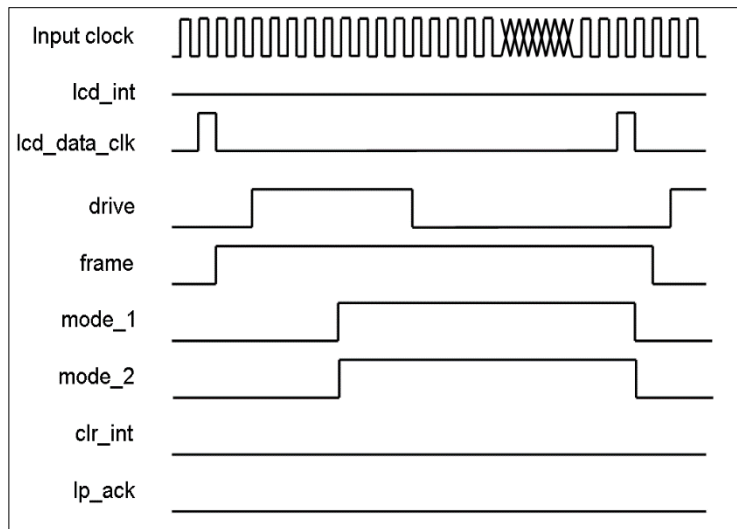
驱动程序功耗模式

段式 LCD 可在以下两种功耗模式下操作：无睡眠模式以及低功耗模式。无睡眠是默认操作模式。

无睡眠模式

在该模式下，段式 LCD 绝不会进入睡眠模式，并在整个帧流程中都将驱动 LCD。图 1 显示的是无睡眠模式下 LCD_Seg 组件的由 UDB 生成（内部）的信号波形。

图 1. 段式 LCD 控制信号（无睡眠模式）



低功耗模式

在该模式下，仅在电压转换时才会主动驱动 LCD，且 LCD 系统模拟组件在各电压转换之间进入睡眠模式。内部 LCD 定时器会将器件从睡眠模式下唤醒一小段时间，以更新 LCD 屏幕。之后，内部逻辑使整个器件返回睡眠模式，直至下一次刷新序列为止。

在低功耗模式下，“Frame Rate”（帧率）参数取决于其他组件参数（例如：LCD 定时器（1 kHz [ILO]或 8 kHz [32 kHz 晶振]）的时钟源、共用线路数和波形类型）。可用 LCD 定时器的有限输入频率说明帧率限制。下面的是针对八条共用线路、低功耗 ILO 模式和 A 型波形配置的最大帧率计算示例：

$$\text{最大帧率} = 1000 / (2 \times \text{通用线路数}) = 62.5 \text{ Hz}$$

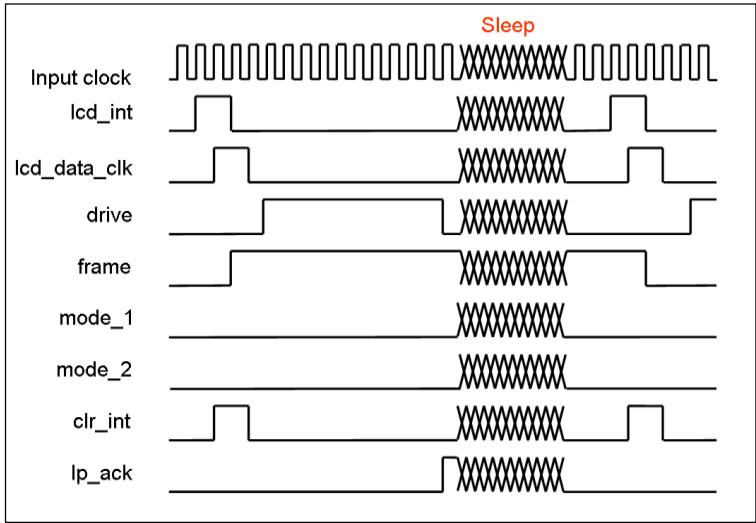
62.5 Hz 是 LCD 定时器可提供的最大帧率；在该帧率下，定时器在输入频率的每个时钟周期内生成一个唤醒信号。

在现实中，LCD 定时器需要一个周期以设置唤醒事件，另需要一个周期清除它。这意味着对最大帧率进行二分频，得到 31.25 Hz 的频率。根据该值，只取整数得到 31 Hz，作为实际的最大帧率。

对于 B 型波形，实际帧率比 A 型波形大两倍，因为 B 型波形需要的唤醒事件少两倍。

图 2 显示的是低功耗模式下 LCD_Seg 组件的由 UDB 生成的信号波形。

图 2. 段式 LCD 控制信号（低功耗模式）

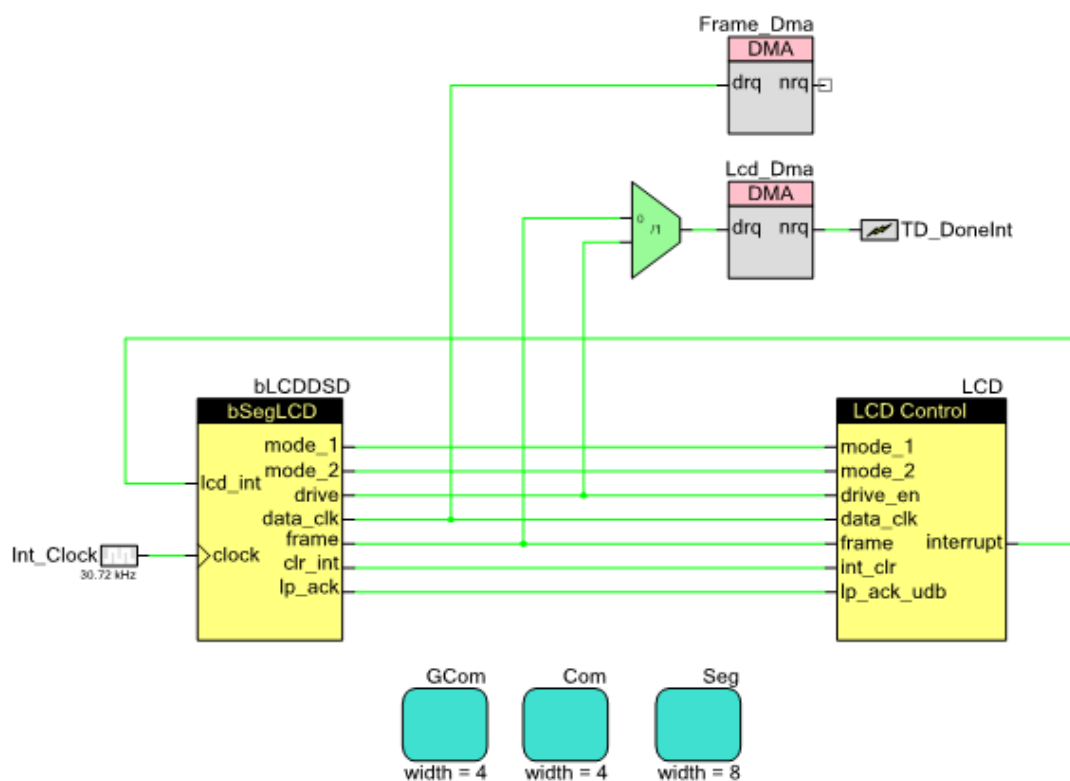


注意：对于 PSoC 5LP 器件，需要使用中断 ISR（即使它是空白的）从睡眠模式唤醒 CPU。因此，使器件进入睡眠模式前需要调用 LCD_Seg_EnableInt() 函数。

框图和配置

图 3 显示的是段式 LCD 组件的内部原理图。它包含基本段式 LCD 组件、LCD 控制模块 (LCD) 组件、DMA 组件、三个 LCD 端口、一个数字端口、一个 ISR 组件以及一个时钟组件。

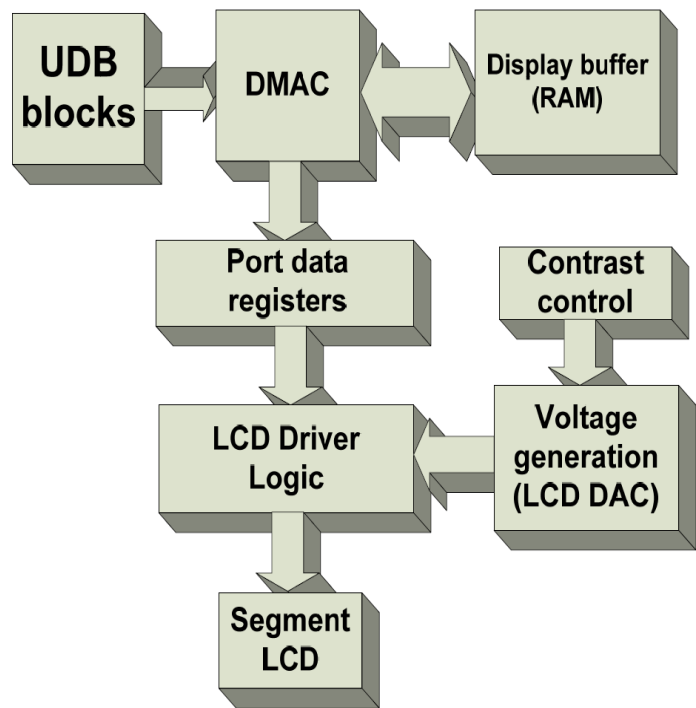
图 3. 段式 LCD 组件原理图



- 基本段式 LCD 组件负责为 LCD 端口和 DMA 组件生成适当的时序信号。
- DMA 组件用于将数据通过伪信号存储器区域从帧缓冲区传输到 LCD 数据寄存器内。
- LCD 组件将处理所需的 DSI 布线。此外，该模块还提供了如 *cyfitter.h* 文件中所定义的必要的寄存器名称。
- 通过 LCD 端口（GCom、Com 和 Seg），可将逻辑信号映射到物理引脚。有两个 LCD 端口实例：一个是共用线路实例，另一个是段线路实例。共用信号的 LCD 端口宽度限定为 16 个引脚，而段信号的 LCD 端口宽度限定为 48 个引脚。还有一个额外的共用端口（即 GCom），仅在使能了“Ganging”（组连接）选项后才包括该端口。

顶层架构

图 4. 段式 LCD 顶层



寄存器

LCD_Seg_CONTROL_REG

位	7	6	5	4	3	2	1	0
值	Frame (帧)	保留			frame done (帧 完成)	mode 2 (模式2)	mode 1 (模式1)	clock enable (时钟 使能)

- clock enable（时钟使能）：使能前面章节中所述的所有内部信号的生成。
- mode 1（模式 1）：mode[2:0]位字段的中间位，用于定义高驱动强度和低驱动强度。
- mode 2（模式 2）：mode[2:0]位字段的高位，用于定义高驱动强度和低驱动强度。
- frame done（帧完成）：DMA 帧数据操作完成后生成同步脉冲。



- frame（帧）：生成 LCD 驱动器的帧信号

LCD_Seg_CONTRAST_CONTROL_REG

保留偏压电压，由 LCD DAC 使用以生成正确的偏压电压。提供了 API，以更改偏置电压。

位	7	6	5	4	3	2	1	0
值	保留	保留	contrast level（对比度）					

- contrast level（对比度）：偏压电压如上所述。

LCD_Seg_LCDDAC_CONTROL_REG

位	7	6	5	4	3	2	1	0
值	lp enable (lp使能)	保留			continuous drive (持续 驱动)	保留	bias select (偏压选择)	

- bias select（偏压选择）：该位将选择偏压。
- continuous drive（持续驱动）：当芯片进入睡眠状态时，该位允许 LCDDAC 保持活动状态。
- lp enable（lp 使能）：允许 UDB 输入 LCD 子系统的低功耗 Ack。

LCD_Seg_TIMER_CONTROL_REG

位	7	6	5	4	3	2	1	0
值	period（周期）						clk select (clk 选择)	enable timer (使能 定时器)

- enable timer（使能定时器）：该位将使能 LCD 定时器。
- clk select（clk 选择）：LCD 定时器源选择时钟。
- period（周期）：LCD 定时器周期。

LCD_Seg_DRIVER_CONTROL_REG

位	7	6	5	4	3	2	1	0
值	保留			bypass enable (旁路使能)	pts	Invert (反转)	mode 0 (模式0)	睡眠模式

- Sleep mode（睡眠模式）：在低功耗模式下，如果该位被设为“1”，则将 LCD 驱动器中的输出缓冲区设为接地，否则将 LCD 驱动器设为 HI-Z。
- mode 0（模式 0）：mode[2:0]位字段的低位，用于定义高驱动强度和低驱动强度。
- invert（反转）：如果设置了该位，将反转段引脚上的数据。
- 如果将此位清除为‘0’ — 正常操作， $V_{OUT} = V_{IO} - 0.5\text{ V}$ 。如果设为‘1’，则 $V_{OUT} = V_{IO}$ 。
- bypass_enable（旁路使能）：如果将此位设为“1”，绕过 LCD 驱动器中的驱动缓冲区，并直接连接到所定的电压输入。如果将该位清除为“0”，即正常操作，电压经过驱动缓冲区传输。

LCD_Seg_LCDDAC_SWITCH_REG[0..4]

位	7	6	5	4	3	2	1	0
值	保留						开关控制[0..4]	

- switch control[0..4]（开关控制[0..4]）：该位字段将为 LCD 驱动器选择电压源。

资源

在 PSoC 3 和 PSoC 5LP 中，段式 LCD 组件是一个硬件专用模块（LCD 固定模块），它与各 UDB 一同使用以允许直接驱动外部 LCD 显示屏。该器件利用以下资源：

配置	资源类型					
	数据路径单元	宏单元	状态单元	控制单元	DMA通道	中断
无睡眠模式	1	5	—	1	2	1
低功耗模式	1	8	—	1	2	2

下表显示的是静态段式 LCD 组件的所有可能的配置。配置名称的含义如下：

Basic（基本）：低级 API 函数集，无任何高级助手 API

Basic, 7-Segment helper（基本，7 段助手）：低级 API 函数集 + 7 段助手 API

Basic, 14-Segment helper（基本，14 段助手）：低级 API 函数集 + 14 段助手 API

Basic, 16-Segment helper（基本，16 段助手）：低级 API 函数集 + 16 段助手 API

Basic, Dot Matrix helper：低级 API 函数集 + 点阵助手高级 API。

Basic, Bar Graph helper（基本，条形图助手）：低级 API 函数集 + 条形图助手高级 API。

配置	PSoC 3 (Keil_PK51)		PSoC 5LP (GCC)	
	闪存 字节	SRAM 字节	闪存 字节	SRAM 字节
基本	2254	86	1560	50
基本，7段助手	2574	87	1698	90
基本，14段助手	2748	87	1928	90
基本，16段助手	2752	87	1932	90
基本，点阵助手	3881	106	3032	106
基本，条形图助手	2794	87	1632	90

直流和交流电气特性

LCD 直接驱动直流规范 001-89755.docx

参数	说明	条件	最小值	典型值	最大值	单位
I _{CC}	LCD系统工作电流	将在400 Hz速率下从睡眠模式唤醒器件以刷新LCD，总线时钟 = 3 Mhz，V _{DDIO} = V _{DDA} = 3 V，4个共模线路，16个段，1/4占空比，50 Hz帧率，未连接显示屏	—	38	—	μA
I _{CC_SEG}	每个段的驱动电流	强驱动模式	—	260	—	μA
V _{BIAS}	LCD偏压范围（V _{BIAS} 是指LCD DAC的主要输出电压（V _O ））	V _{DDA} ≥ 3 V and V _{DDA} ≥ V _{BIAS}	2	—	5	V

	LCD偏压步长大小	$V_{DDA} \geq 3\text{ V}$, 且 $V_{DDA} \geq V_{BIAS}$	—	$9.1 \times V_{DDA}$	—	mV
	每个段/共模路线驱动器上的LCD电容	驱动器可以组合使用	—	500	5000	pF
	长期段偏移		—	—	20	mV
I_{OUT}	每个段驱动器的输出驱动电流	$V_{DDIO} = 5.5\text{ V}$, 强驱动模式	355	—	710	μA

LCD 直接驱动交流规范 001-89755.docx

参数	说明	条件	最小值	典型值	最大值	单位
f_{LCD}	LCD帧率		10	50	150	Hz

组件勘误表

本节列出了组件的已知问题。

赛普拉斯ID	组件版本	问题	解决方案
191257	v3.40	在没有修正PSoC Creator 3.0 SP1中的版本编号时进行更改这组件。更多信息, 请参见基础知识文章KBA94159 (网页地址: www.cypress.com/go/kba94159)。	解决方案是不必要的。设计不受任何影响。

组件更改

本节列出了组件与先前版本相比的主要更改的内容。

版本	更改说明	更改原因/影响
3.40.b	编辑了数据手册。	将组件勘误章节添加到被更改组件的文档, 但设计不受任何影响。
3.40.a	更新了LCD_Seg_SetBias()说明部分。	说明部分中描述了128个偏压电平值, 但它只有63个值。
3.40	更新了“Custom Characters”(自定义字符)选项卡以及默认字符集, 使它们与SegLCD_P4组件相匹配。	旨在保持SegLCD和PSoC 4 SegLCD组件之间的一致性。



版本	更改说明	更改原因/影响
	对“Basic Configuration”（基本配置）以及“Driver Power Settings”（驱动程序功耗设置）选项卡的GUI进行了少量改进。	可用性的改进。
3.30	添加了MISRA合规性章节。	该组件具有所描述的特定偏差。
	更新了段式LCD时钟的最新版本、中断及DMA组件。	
	解决了LCD_Seg_WriteStringDotMatrix_n()和LCD_Seg_WriteString16Seg_n() API中的问题。	这些函数有时可能会显示垃圾值。由于实现过程中的条件检查不正确。而不是NULL指针所指出的零终止字符串函数。
	解决了LCD_Seg_WriteBargraph_n() API的相关问题。	该API不会清除其先前所显示的输出，这样有时会导致LCD上显示意外的结果。
	解决了LCD_Seg_Write7SegNumber_n() API的相关问题。	在“no leading zeros”（无前导零）模式下使用该函数时，它不会清除左侧上剩余的数字。则在“leading zeros”（前导零）模式下，使用零填充这些数字。
	解决了LCD_Seg_SetBias()说明部分中的问题。	新增了LCD_Seg_SetBias()如以前所描述为这样返回‘void’的返回值描述。 此外，在API的说明部分中也描述了128个偏压电平值，但它只有63个值。
	已将LCD_Seg_WriteStringDotMatrix_n()、LCD_Seg_WriteString16Seg_n()和LCD_Seg_WriteString16Seg_n() API中的“character”（字符）参数声明从“uint8 * character”改为“uint8 const character[]”	这是与MISRA相关的更改。将变量声明为指针时，MISRA不允许使用变量的阵列索引。
3.20	已向组件添加了空白的唤醒ISR。该更改仅涉及到PSoC 5LP器件。	实现PSoC 5LP时，需要使用中断唤醒器件。
	已添加了用于计算DefaultStep（默认步长）的公式。	
	修复了用于计算HighDriveTimeValdef和HighDriveTimeValmax值的公式。	
3.10	解决了LCD_SegStat_Write7SegNumber() API中的相关问题。	API中有一个错误，它将LCD上所显示的最后一位数字覆盖为0。仅在采用“前导零”模式的输出时才发生这种问题。
	已向API中添加了Keil函数可重入支持。	已添加了该功能，以便客户能够指定哪个所生成的函数是可重入的。

版本	更改说明	更改原因/影响
3.0	已将WRITE_PIXEL()宏定义从 #define LCD_Seg_WRITE_PIXEL(pixelNumber, pixelState) LCD_Seg_WritePixel(pixelNumber, pixelState) 改为 #define LCD_Seg_WRITE_PIXEL(pixelNumber, pixelState) (void) LCD_Seg_WritePixel(pixelNumber, pixelState)	无需在组件实现过程中分析 LCD_Seg_WritePixel() API的返回值，因此更改了宏以处理该情况。
	已更改组件的verilog实现，以避免在低功耗模式下使用组件时出现STA组合周期警告。	异步SR锁存器已被替换为同步逻辑。
	已在LCD_Seg_ReadPixel() API中添加了针对未分配助手像素的验证。	将0xFF (LCD_Seg_Stat_PIXEL_UNKNOWN_STATE)新的返回状态添加到LCD_Seg_ReadPixel()。当像素存在于助手中但未分配时，返回此状态。
	优化了LCD_Seg_WriteBargraph()函数。	代码重构节省了一些闪存空间。
	优化了LCD_Seg_WriteString14Seg()和LCD_Seg_WriteString16Seg()函数。	已将索引更改为指针递增，从而节省RAM空间，并有更多时间执行这些函数。
	已优化了LCD_Seg_Write7SegNumber() API。	LCD_Seg_Write7SegNumber() API中有一些无效代码，该代码它在该API的较长执行时间内出现。
	完成了LCD_Seg_Wakeup() API补丁。	LCD_Seg_Wakeup() API应返回但未返回任何状态，因此修复了 API 以返回正确的状态。
2.10	已修复了LCD_Seg_Start()子程序	如果之前使能了全局中断，由于全局中断实施过程中出错，因此禁用全局中断而无任何通知。
	已更改定制器的“Driver Power Settings”（驱动程序功耗设置）选项卡。已添加了“Custom Step”（自定义步骤）和“Default Step”（默认步骤）等两个新字段。	在低功耗模式下，“High Drive Time”（高驱动时间）参数的新选项可用。除了递增的默认步长外，还可选择更精确的自定义步长。这取决于主时钟的频率。
	解决了在LCD刷新序列完成前使SegLCD进入睡眠状态的问题。	内部组件信号lp_ack在刷新序列开始时转变为高电平状态，但它本来应该在刷新序列结束时才转变为高电平状态。
	解决了在“High Drive Strength”（高驱动强度）被设为“seg=2x com=2x”或“seg=1x com=4x”时反转LCD显示屏图像时出现的问题。	该问题就是将错误的值写入到LCD驱动器控制寄存器上。通过掩码的：“和”（AND）运算生成该值，而掩码不正确。将它改成正确的掩码以解决问题。
	修复了LCD_Seg_WriteBargraph() API的相关问题。	解决了LCD_Seg_WriteBargraph() API问题。该问题在模式参数被设为10时由LCD的条形图像素的错误输出导致的。在“Position”（位置）参数满足了条件“位置 + 10 > 条形图像素”时，出

版本	更改说明	更改原因/影响
		现此情况。在这种情况下，条形图的最后一个像素未设置，但实际上是应该设置的。
	向数据手册中添加了特性数据。	
	向组件中添加了信息，以说明它与芯片修订版的兼容性。	如果在不兼容的芯片上使用组件，该工具将返回错误。如果发生该情况，请更新所支持您目标器件的修订版。
	对数据手册进行了少量编辑及更新。	
2.0.a	已向数据手册中的LCD_Seg_Init()、LCD_Seg_Start()和LCD_Set_Stop() API添加了注解。	
	向组件中添加了信息，以说明它与芯片修订版的兼容性。	如果组件在不兼容的芯片上使用，该工具将返回错误。如果发生该情况，请更新所支持您目标器件的修订版。
2.0	添加了睡眠/唤醒和初始化/使能API。	支持低功耗模式并提供常用接口，以单独控制大多数器件的初始化及使能。
	已添加了新的API文件，即SegLCD_PM.c，该文件包含睡眠模式API的声明。	支持低功耗模式的新要求。
	<p>更新了组件，以支持PSoC 3 ES3及更高版本。更新了“Configure”（配置）对话框：</p> <ul style="list-style-type: none"> ■ 移除了已过时的控制：使能调试模式、低驱动模式 ■ 已添加新控制：显示屏大小、高驱动强度、低驱动模式 ■ 已通过将“始终活动”和“低功耗”等旧模式更改为新的“无睡眠”、“低功耗 ILO”和“低功耗 32XTAL”更改了“驱动程序功耗模式”选项 ■ 已添加了自定义字符“Tab”，以允许用户为点阵助手创建自定义字符集 ■ 使偏压类型控制可编辑，并更改了偏压电压的选择值 	<p>新增了支持PSoC 3 ES3器件和LCD HW架构的新要求，因此创建了新的2.0版。</p> <p>版本1.xx支持PSoC 3 ES2和PSoC 5芯片的修订版</p>
	已向以下函数添加了`=ReentrantKeil(LCD_Seg_ . " _...")`： LCD_Seg_Stop() LCD_Seg_EnableInt() LCD_Seg_DisableInt() LCD_Seg_SetBias() LCD_Seg_WriteInvertState()	如果需要重入，请允许用户将这些API设置为可重入函数。

版本	更改说明	更改原因/影响
	LCD_Seg_ReadInvertState() LCD_Seg_RedPixel() LCD_Seg_SaveConfig() LCD_Seg_RestoreConfig()	

赛普拉斯半导体公司，2013-2016 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，WICED，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。

