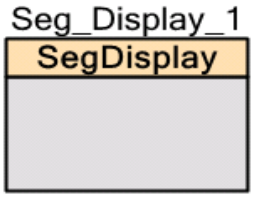
段显示器（Seg_Display）

1.20



特性

- 仅适用于 PSoC 5 器件（对于 PSoC 3 和 PSoC 5LP 器件，请使用 LCD 版本 3.10 组件。）
- 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（帧缓冲器）
- 用户定义的像素或符号映射，可选 7、14 或 16 段字符；5x7 或 5x8 点阵；条形图计算子程序

概述

段显示器（Seg_Display）组件可以在复用率达到 16x 时直接驱动 3.3 V 和 5.0 V LCD 显示屏。此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。

生成内部偏压，这去除了对任何外部硬件的需求并支持基于软件的对比度调整。使用升压转换器，显示屏偏置电压可能高于 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/符号都可以打开或者关闭。此外，段显示器组件还提供高级支持，从而简化下列显示屏显示结构的类型：

- 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。）
- 1-255 个元件条形图

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*。

何时使用段显示器

当需要在复用率为 2x-16x下直接驱动 3.3 V或 5.0 V LCD显示屏时，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。

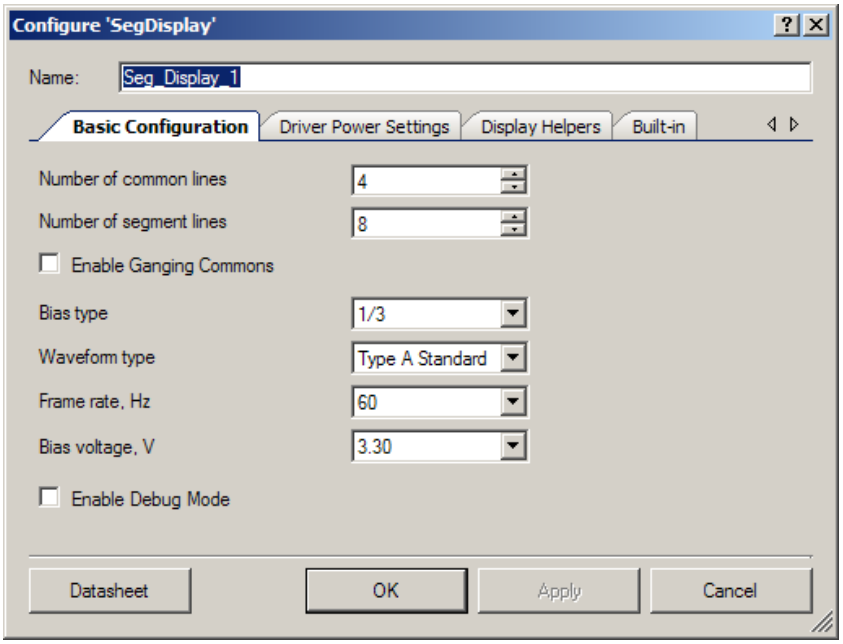
输入/输出接口

在原理图画布上没有可见的组件连接；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。

组件参数

将段显示器组件拖入设计中，双击该组件，打开 **Configure**（配置）对话框。**Configure** 对话框包含多个不同参数类型的选项卡，用于设置段显示器组件。

Basic Configuration（基本配置）选项卡



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。可能值的范围为（2 至 62）— **通用线路数**。默认值为 **8**。

Enable Ganging Commons（使能共用信号组连接）

选中此复选框时，可以将 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）

此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。

帧率

该参数确定了显示器的刷新率。可能值的范围为 10 Hz 至 150 Hz。默认值为 **60 Hz**。

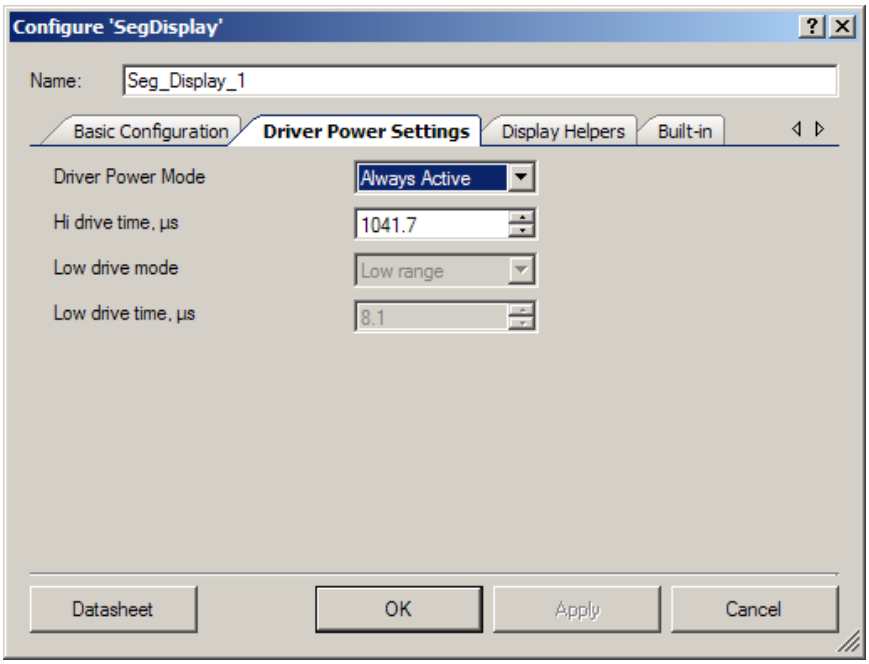
偏压

此参数确定了 LCD DAC 的偏置电压电平。可能值的范围为 2 V 至 5.2 V。默认值为 **3.3 V**。

使能调试模式

如果使能，则将调试端口添加到组件，由此查看驱动 LCD 驱动程序的信号。例如，可以使用它来查看高电平和低电平驱动时间。

Driver Power Settings（驱动程序功耗设置）选项卡



Driver Power Mode（驱动器功耗模式）

Driver Power Mode 参数定义了组件的功耗模式。有两种可用的功耗模式：

- **Always Active**（始终运行）：LCD DAC 始终为打开状态
- **Low Power**（低功耗）：LCD DAC 在电压转变之间为关闭状态

请参见此数据手册后面章节功能说明中的驱动程序功耗模式。

高电平驱动时间

此参数用来定义高电平驱动模式在一个电压转变过程内的有效时间。

High drive time 最小值、最大值和默认值的计算在此数据手册后一部分中被定义。

低电平驱动模式

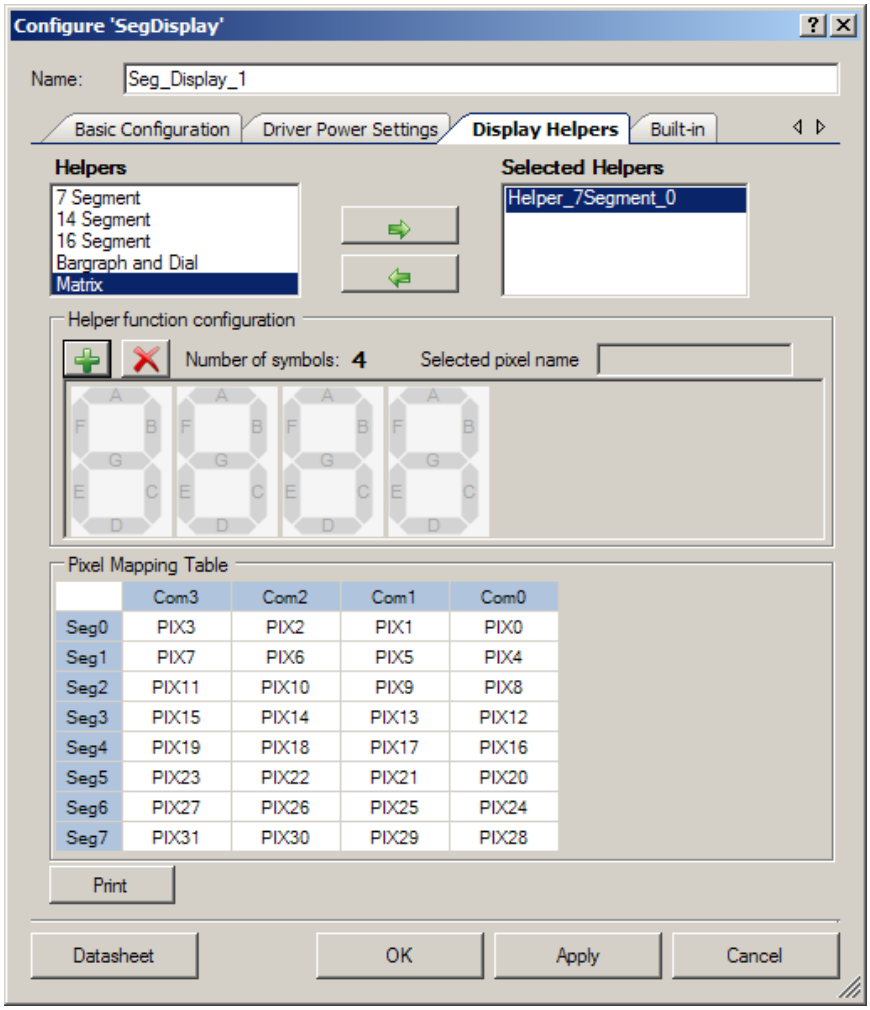
将 **Driver Power Mode** 设置为 **Low Power** 时，此参数可用。**Low drive mode** 有两个可用值：

- **Low range**（低电平范围）— 激活低电平驱动模式
- **High range**（高电平范围）— 激活第二个低电平驱动高电流模式（Lo2）。

低电平驱动时间

该 **Low drive time** 参数定义低电平驱动模式在电压转变过程中有效的时间。

Display Helpers（显示助手）选项卡



通过显示助手，您可以配置一组显示段，将之作为其中一种预定义的显示器元件类型：

- 7、14 或 16 段显示
- 点阵显示器（5x7 或 5x8）
- 线或圆条形图显示器

基于字符的显示助手可用于组合多个显示符号，从而创建多字符显示器元件。

助手/选定助手

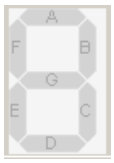
可以将一个或多个助手添加到 **Selected Helpers**（选定助手）列表中，其方法是在 **Helpers**（助手）列表中选择所需助手类型，然后单击右箭头按钮。如果没有足够的引脚来支持新助手，则不会添加该助手。要删除某个助手，在 **Selected Helpers** 列表中选中该助手，然后单击左箭头按钮。

注意：将显示助手添加到组件后，你不可以再更改通用线路或段线路数。重要的一点是，定义任何显示助手前，一定要为组件设置通用和段线路数。更改通用或段线路数之前，必须删除已定义的所有显示助手。

Selected Helpers 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。如果从列表中移除 **Selected Helper** ，则要重新命名剩余助手。 添加助手时，名称将使用最低有效的后缀。

为每个助手提供了 API。有关详细信息，请参见应用编程接口一节中的内容。

- **7 Segment Helper**(7 段助手) — 此助手长度可能为 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 Segment Helper**(14 段助手) — 此助手长度可高达 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) — 此助手长度可高达 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。

- **Bargraph and Dial Helper**(条形图和拨号助手) — 这些助手用于 1 到 255 段条形图和拨号指示符。条形图是单个选择的像素或选定像素及指定像素左边或右边的所有像素

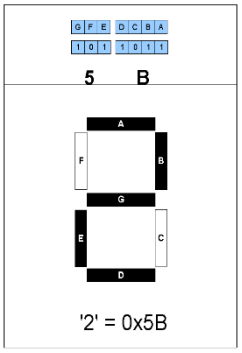
- **Matrix Helper**（矩阵助手） — 此助手支持最多八个字符元素。该组件支持 5x7 或 5x8 行/列字符。通过配置 2 个或多个点阵助手，可以创建较长字符串，用来定义显示屏相邻点阵部分。该助手显示一个单一 ASCII 字符或空结尾的字符串。

点阵助手具有管脚限制。它必须对矩阵行使用 7 或 8 个连续通用驱动，对矩阵列使用 5-40 个连续段驱动。该组件支持标准的 Hitachi HD44780 字符集。

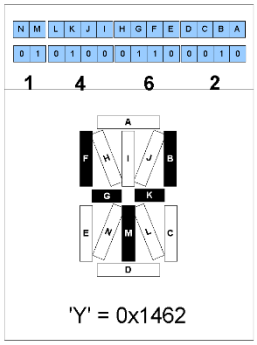
字符编码

每个高级助手 API 有其自己的查找表。该表包含一组编码像素状态，这构成特定的字符映像。下列示例说明了如何对特定字符进行编码（段名称与自定义程序中显示的名称不同）。

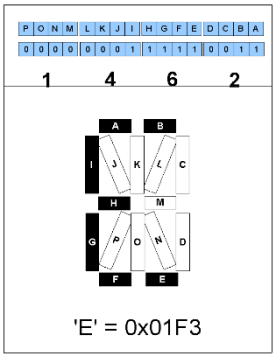
7段编码



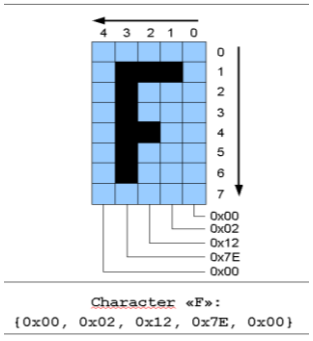
14段编码



16段编码



点阵编码



助手功能配置

通过本节对话框，可以配置助手；这包括为助手添加符号或从助手中移除符号及命名像素。

1. 从 **Selected Helpers** 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。如果没有足够的可用引脚数支持新符号，则不会再添加符号。
3. 要重新命名作为助手功能组成部分的像素，请在**助手功能配置**显示屏中，选择符号图像上的像素。当前名称显示在选定像素名称字段中，可以根据需要进行修改。

像素命名

默认像素名称为“PIX#”，其中“#”表示像素从 **Pixel Mapping Table**（像素映射表）右上角开始的增量顺序号。

像素（与助手符号相关）的默认命名方式具有不同格式。默认名称由前缀部分、所有像素的通用的符号部分和唯一段标识符组成。默认前缀表示助手类型和符号实例。例如，在 7 段助手中，其中一个符号的默认像素名称可能是“H7SEG4_A”，其中：

H7	表示像素是 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
建议	指示 7 段符号中的唯一段

仅有默认像素名称的唯一部分显示在符号图像中。如果修改像素名称，则整个名称显示在符号图像中，即使与其他像素共享通用前缀也如此。

注意：所有像素名称必须是唯一的。

当助手功能符号元素分配到**像素映射表**（如下所述）中的某个像素时，该像素采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**像素映射表**是帧缓冲器的表示法。为使 API 函数正确工作，**助手函数配置**中的每一个像素必须分配到**像素映射表**中指定的像素位置。有关正确分配位置的信息，请参见 LCD 显示屏数据手册。

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。

通过在表显示屏上双击像素，然后输入所需名称，可以重新命名**像素映射表**中的像素。可以使用此方法命名与其中某个可用助手类型关联的像素。

Print（打印）按键用于打印**像素映射表**。

时钟选择

Seg_Display 组件使用两个内部时钟，而无需使用外部时钟。放置组件之后，LCD 自动专门使用这两个时钟。第一个时钟生成刷新频率，第二个时钟为低驱动缓冲器生成 100-kHz 频率的时钟。

应用编程接口

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

默认情况下，PSoC Creator 将实例名称“Seg_Display_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、变量和符号常量的前缀。出于可读性考虑，下表中使用的实例名称为“Seg_Display”。

函数	说明
Seg_Display_Start()	启动LCD组件，使能所需中断。
Seg_Display_Stop()	禁用LCD组件和关联的中断和DMA通道。
Seg_Display_EnableInt()	使能LCD中断。如果调用Seg_Display_Start()，则无需使能
Seg_Display_DisableInt()	禁用LCD中断。如果调用Seg_Display_Stop()，则无需禁用
Seg_Display_SetBias()	将LCD显示屏的偏压电平设置为128个值的其中一个。
Seg_Display_WriteInvertState()	根据输入参数反转显示屏。
Seg_Display_ReadInvertState()	返回显示屏反转状态的当前值：正常或反转
Seg_Display_ClearDisplay()	清除显示屏和关联帧缓冲器RAM。
Seg_Display_WritePixel()	根据**PixelState**（像素状态）设置或清除像素
Seg_Display_ReadPixel()	读取帧缓冲器中像素的状态。
Seg_Display_SetAwakeMode()	当处于低功耗模式时，将LCD驱动程序缓冲输出设置为高阻抗。
Seg_Display_Sleep()	停止LCD，然后保存用户配置。
Seg_Display_Wakeup()	恢复并使能用户配置。
Seg_Display_Init()	根据Configure（配置）对话框设置初始化或恢复LCD。
Seg_Display_Enable()	使能组件。
Seg_Display_SaveConfig()	保存LCD配置。
Seg_Display_RestoreConfig()	恢复LCD配置。

全局变量

变量	说明
Seg_Display_initVar	表示Seg_Display是否完成初始化。变量将初始化为0，并在第一次调用Seg_Display_Start()时设置为1。这样，第一次调用Seg_Display_Start()子程序后，组件不用重新初始化即可重启。

	如果需要重新初始化组件，则在调用Seg_Display_Init()函数后，可以调用Seg_Display_Start()或Seg_Display_Enable()函数。

uint8 Seg_Display_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： 无

void Seg_Display_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： 无

void Seg_Display_EnableInt(void)

说明： 使能LCD中断。如果调用Seg_Display_Start()，则无需调用该参数。每次更新LCD后（TD完成）都会发生中断。

参数： 无

返回值： 无

其他影响： 无

void Seg_Display_DisableInt(void)

说明： 禁用LCD中断。如果调用Seg_Display_Stop()，则无需调用该参数。

参数： 无

返回值： 无

其他影响： 无

uint8 Seg_Display_SetBias(uint8 biasLevel)

说明： 此函数用于将LCD显示屏的偏压电平设置为128个值的其中一个值。实际值数由模拟供电电压V_{DDA}来限制。偏置电压不能超过V_{DDA}。更改偏压电平会影响LCD对比度。

参数： uint8 biasLevel：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 Seg_Display_WriteInvertState(uint8 invertState)

说明： 此函数用来根据输入参数反转显示屏。在硬件中反转显示屏，无需在帧缓冲器中对显示RAM进行更改

参数： uint8 invertState：设置显示的反转状态。

数值	说明
Seg_Display_NORMAL_STATE	设置显示屏的正常状态
Seg_Display_INVERTED_STATE	设置显示屏的反转状态

返回值： uint8 cystatus：标准API返回值

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

其他影响： 无

uint8 Seg_Display_ReadInvertState(void)

说明： 此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：设置显示的反转状态

返回值	说明
Seg_Display_NORMAL_STATE	显示器的正常状态
Seg_Display_INVERTED_STATE	显示器的反转状态

其他影响： 无

void Seg_Display_ClearDisplay(void)

说明： 此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 Seg_Display_WritePixel(uint16 pixelNumber, uint8 pixelState)

说明： 此函数用来根据输入参数**PixelState**设置或清除某个像素。通过一个封装好的数据包寻址像素。

参数： uint16 pixelNumber：帧缓冲器中指向像素位置的封装数组。LSB低效半字节中3个最低有效位是字节中的数位位置，LSB高效半字节（4位）是复用行中的字节地址，而MSB低效半字节（4位）则是复用行编号。生成组件“.h”文件，其中包括各个像素使用“#define”的格式。

uint8 pixelState：指定的**pixelNumber**被设为此像素状态。

数值	说明
Seg_Display _PIXEL_STATE_OFF	将像素设置为关闭状态。
Seg_Display_PIXEL_STATE_ON	将像素设置为打开状态。
Seg_Display_PIXEL_STATE_INVERT	反转像素的当前值。

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。未对数位位置执行检查。

返回值	说明
CYRET_BAD_PARAM	封装字节地址或行值无效
CYRET_SUCCESS	函数已成功完成

其他影响： 无

uint8 Seg_Display_ReadPixel(uint16 pixelNumber)

说明： 此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：帧缓冲器中指向像素位置的封装数组。LSB低效半字节中3个最低有效位是字节中的数位位置，LSB高效半字节（4位）是复用行中的字节地址，而MSB低效半字节（4位）则是复用行编号。生成组件“.h”文件，其中包括各个像素使用“#define”的格式。

返回值： uint8 pixelState：返回指定**pixelNumber**的当前状态。

数值	说明
0x00	像素关闭。
0x01	像素打开。
0xFF	像素未连接

其他影响： 无

void Seg_Display_SetAwakeMode(void)

说明： 当处于低功耗模式时，将LCD驱动程序缓冲输出设置为高阻抗状态。

参数： 无

返回值： 无

其他影响： 无

void Seg_Display_SetSleepMode(void)

说明： 当处于低功耗模式时，将LCD驱动程序缓冲输出设置为接地状态。

参数： 无

返回值： 无

其他影响： 无

void Seg_Display_Sleep(void)

说明： Seg_Display_Sleep()函数检查组件是否为使能状态，并保存该状态。然后调用Seg_Display_Stop()函数和Seg_Display_SaveConfig()函数以保存用户配置。

调用Seg_Display_Sleep()函数后，调用CyPmSleep()或CyPmHibernate()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： 请勿更改组件引脚驱动模式。

uint8 Seg_Display_Wakeup(void)

说明： Seg_Display_Wakeup()函数调用Seg_Display_RestoreConfig()函数以用来恢复用户配置。如果在调用Seg_Display_Sleep()函数之前已使能组件，则Seg_Display_Wakeup()函数将重新启用该组件。

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： 如果在调用Seg_Display_Wakeup()函数之前未调用Seg_Display_Sleep()或Seg_Display_SaveConfig()函数，可能产生意外行为。

void Seg_Display_Init(void)

说明： 根据Configure（配置）对话框设置初始化或恢复组件参数。配置并使能所有必要的硬件模块，清除帧缓冲器。

参数： 无

返回值： 无

其他影响： 无

uint8 Seg_Display_Enable(void)

说明： 使能组件。使能所有必要的时钟，设置寄存器初始值，并执行组件复位。

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： 无

void Seg_Display_SaveConfig(void)

说明： 此函数会保存组件配置以及非保留寄存器。它还会保存“Configure”对话框中定义的或通过相应API修改的当前组件参数值。通过Seg_Display_Sleep()函数调用此函数。

参数： 无

返回值： 无

其他影响： 无

void Seg_Display_RestoreConfig(void)

说明： 此函数会恢复组件配置以及非保留寄存器。此外，该函数还用于将组件参数值恢复至调用Seg_Display_Sleep()函数之前的状态。

参数： 无

返回值： 无

其他影响： 如果在调用此函数之前未调用Seg_Display_Sleep()或Seg_Display_SaveConfig()函数，可能产生意外行为。

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
Seg_Display_Write7SegDigit_n	显示7段显示元素阵列中的十六进制数字。
Seg_Display_Write7SegNumber_n	显示7段显示元素1-5位数字的整型值。
Seg_Display_WriteBargraph_n	将显示线性或圆形条形图上的整数位置。
Seg_Display_PutChar14Seg_n	显示14段字母数字字符显示元素阵列中的字符。
Seg_Display_WriteString14Seg_n	显示14段字母数字字符显示元素阵列中的空字符结束的字符串。
Seg_Display_PutChar16Seg_n	显示16段字母数字字符显示元素阵列中的字符。
Seg_Display_WriteString16Seg_n	显示16段字母数字字符显示元素阵列中的空字符结束的字符串。
Seg_Display_PutCharDotMatrix_n	显示点阵字母数字字符显示元素阵列中的字符。
Seg_Display_WriteStringDotMatrix_n	显示点阵字母数字字符显示元素阵列中的空字符结束的字符串。

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 Seg_Display_Write7SegDigit_n(uint8 digit, uint8 position)

说明： 此函数用来显示7段显示元素阵列中的十六进制数字。这些数字可以是十六进制值，取值范围为0-9和A-F。自定义程序显示助手工厂必须用于定义与7段显示元素关联的像素集。在帧缓冲器中，可以定义多个7段显示元素，并通过函数名称中的后缀(n)来寻址。只有在器件自定义程序中定义了7段显示元素，才包含此函数。

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16）。此外，十六进制字符ASCII数字也有效。如果**数字**无效，则在指定的位置显示0**值**。将此参数设置为16，可以清除指定**位置**上的数字。

uint8 position：从右侧的0开始从右到左计数的数字位置。如果数字位置位于定义的显示区以外，则字符将不显示。

返回值： 无

其他影响： 无

void Seg_Display Write7SegNumber_n(uint16 value, uint8 position, uint8 mode)

说明： 此函数用来显示7段显示元素1-5位数字阵列的16位整型值。自定义程序显示助手工具必须用于定义与7段显示元素相关联的像素集。在帧缓冲器中，可以定义多个7段显示元素组，并通过函数名称中的后缀(n)来寻址。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。只有在器件自定义程序中定义了7段显示元素，才包含此函数。

参数： uint16值：显示无符号的整型值。

uint8 position：从右侧的0开始自右向左计数中最低有效位的位置。如果定义的显示区域包含**值**所需的较少数字，则不会显示最高有效数字或部分数字

uint8 mode：设置显示模式。可能为0或1。

返回值： 无

其他影响： 无

void Seg_Display_WriteBargraph_n(uint8 location, uint8 mode)

说明： 此函数用来显示1-255段条形图中的8位整数位置（编号顺序从左到右）。条形图可能是用户定义的1-255段之间的任意大小。此外，条形图还可以创建为圆形，用来显示旋转位置。自定义程序显示助手工厂必须用于定义与条形图显示元素相关联的像素集。在帧缓冲器中，可以创建多个条形图显示，并通过函数名称中的后缀(n)来寻址。只有在器件自定义程序中定义了条形图显示元素，才可以包含此函数

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	特定位置的段处于打开状态。
1	位置段和左侧所有段均为打开状态。
–1	位置段和右侧所有段均为打开状态。
2至10	显示位置段和右侧2-10段。此模式可用于创建较宽的指示符。

返回值： 无

其他影响： 无

void Seg_Display_PutChar14Seg_n(uint8 character, uint8 position)

说明： 此函数用来显示14段字母数字字符显示元素阵列中的一个8位字符。必须使用定制器显示助手工具定义与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，并通过函数名称中的后缀(n)来寻址。只有在器件自定义程序中定义了14段显示元素，才包含此函数。

参数： uint8 character：要显示字符的ASCII值（ASCII值为0-127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 Seg_Display_WriteString14Seg_n(*uint8 character, uint8 position)

说明： 此函数可显示14段字母数字字符显示元素阵列上的空结尾字符串。自定义程序显示助手工厂必须用于定义与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，并通过函数名称中的后缀(n)来寻址它们。仅在组件定制器中定义了14段显示元素时，此函数才可用。

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。不受影响的所有位置仍处于它们先前的像素状态。

void Seg_Display_PutChar16Seg_n(uint8 character, uint8 position)

说明： 此函数用来显示16段字母数字字符显示元素阵列中的一个8位字符。自定义显示助手工具必须用于定义与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，并通过函数名称中的后缀(n)来寻址。只有在组件自定义程序中定义16段显示元素，才包含此函数。

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) Seg_Display_WriteString16Seg_n(*uint8 character, uint8 position)

说明： 此函数用来显示16段字母数字字符显示元素阵列中的空字符结束的字符串。自定义显示助手工具必须用于定义与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，并通过函数名称中的后缀(n)来寻址。只有在组件自定义程序中定义了16段显示元素时，才包含此函数。

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。未受影响的所有位置将仍处于其先前的像素状态。

void Seg_Display_PutCharDotMatrix_n(uint8 character, uint8 position)

说明： 此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。在帧缓冲器中，可以定义多个点阵字母数字显示元素组，并通过函数名称中的后缀(n)寻址它们。仅在组件定制器中定义了点阵显示元素时，此函数才可用。

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 Seg_Display_WriteStringDotMatrix_n(*uint8 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。在帧缓冲器中，可以定义多个点阵字母数字显示元素组，并通过函数名称中的后缀（n）寻址它们。仅在组件定制器中定义了点阵显示元素时，此函数才可用。

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 函数用于更改由段显示器组件使用的引脚驱动模式。

函数	说明
Seg_Display_ComPort_SetDriveMode	设置由段显示器组件通用线路占用的所有引脚的驱动模式。
Seg_Display_SegPort_SetDriveMode	设置由段显示器组件段线路占用的所有引脚的驱动模式。

void Seg_Display_ComPort_SetDriveMode(uint8 mode)

说明： 设置由段显示器组件通用线路占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

Seg_Display_SegPort_SetDriveMode(uint8 mode)

说明： 设置由段显示器组件段线路占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

Seg_Display_COMM_NUM

定义当前组件配置用户定义显示的通用线路数。

Seg_Display_SEG_NUM

定义当前组件配置用户定义显示的段线路数。

Seg_Display_BIAS_TYPE

定义当前组件配置用户定义显示的偏压类型。

Seg_Display_BIAS_VOLTAGE

定义用户定义显示的偏置电压电平。初始化过程中，将在 LCDDAC 控制寄存器中设置此值。

Seg_Display_FRAME_RATE

定义当前组件配置用户定义显示的刷新率。

Seg_Display_WRITE_PIXEL

这是包含无效类型的 Seg_Display_WritePixel()函数的宏定义。

Seg_Display_READ_PIXEL

这是 Seg_Display_ReadPixel()函数的宏定义。

Seg_Display_FIND_PIXEL

此宏用来计算帧缓冲器中的像素位置。它采用自定义程序像素表和专用于 LCD 的物理引脚的信息。此宏是像素映射机制的基础。在帧缓冲器中，使用计算后的像素位置来定义像素表中的每一个像素名称。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合规性和本组件的偏差情况。定义了两种类型的偏差：

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

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

尚未根据 MISRA-C:2004 编码准则合规性验证段显示器组件源代码。

固件源代码示例

PSoC Creator 在“Find Example Project”（查找示例项目）对话框中提供了很多包括原理图和代码示例的示例项目。要查看特定组件实例，请打开“Component Catalog”中的对话框或原理图中的组件样例。要查看通用示例，请打开 Start Page 或 **File** 菜单中的对话框。根据要求，可以通过使用对话框中的 **Filter Options** 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”的内容。

功能说明

默认配置

Seg_Display 组件的默认配置提供通用的 LCD 直接段式驱动控制器。默认 Seg_Display 配置为：

- 4 个共用线路
- 8 个段线路
- 60 Hz 刷新率

- 始终有效的功耗模式
- 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

段显示器组件的主要特性为具有不同特征和布局的 LCD 提供灵活支持。

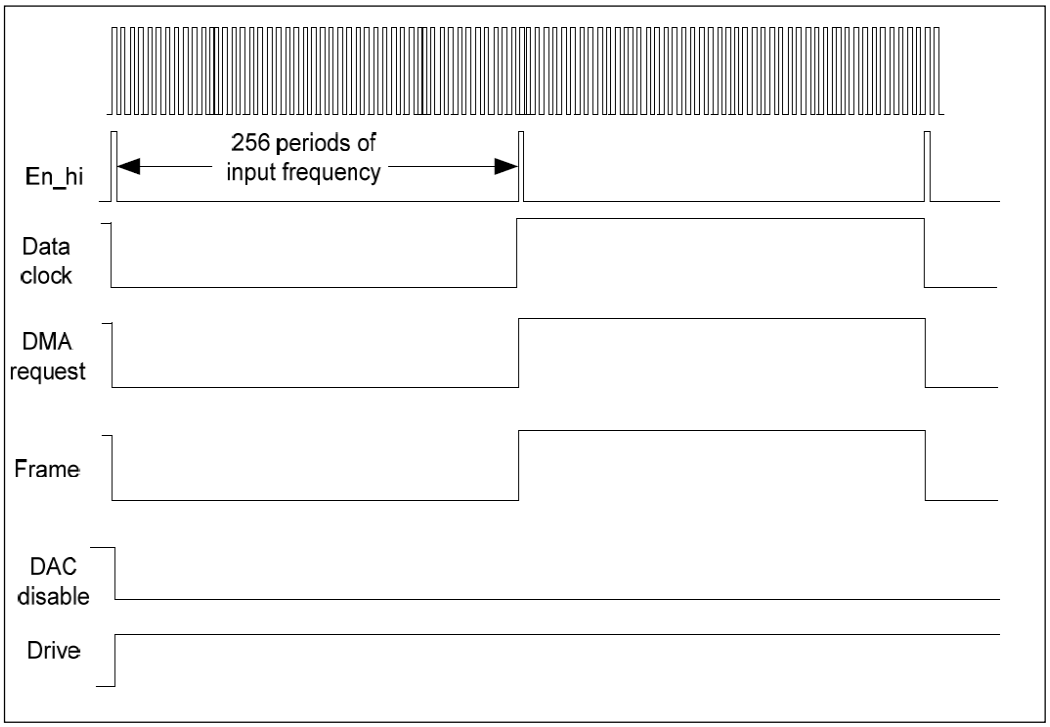
驱动程序功耗模式

始终有效的模式

在此使用模式下，LCD 可以在整个帧内获得驱动。这意味着该模式可以为 LCD DAC 供电，无论何时使能组件，均可以将内部信号驱动置为高电平。

图 1 是 Seg_Display 组件（处于始终运行模式）UDB 生成的（内部）信号的波形（类型 A）：

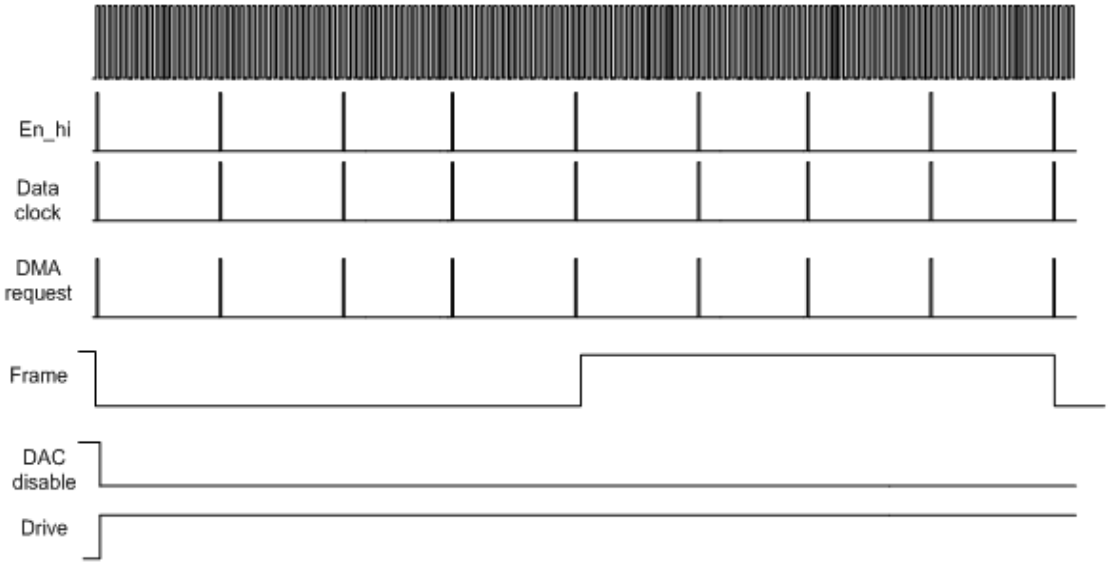
图 1. 段显示器控制信号类型 A 波形（处于始终运行模式）



注意：有关详细信息，请参考时序计算。

图 2 是 Seg_Display 组件（处于始终运行模式）UDB 生成的（内部）信号的波形（类型 B）：

图 2. 段显示器控制信号类型 B 波形（始终运行模式）



显示的信号针对 1/4 复用率情况。

低功耗模式

在此使用模式下，仅在电压转变时才有效驱动 LCD，并在每次电压转变间隔内为 LCD 系统模拟组件断电。

图 3 是 Seg_Display 组件（低功耗模式）UDB 生成的信号的波形（类型 A）：

图 3. 段显示器控制信号类型 A 波形（低功耗模式）

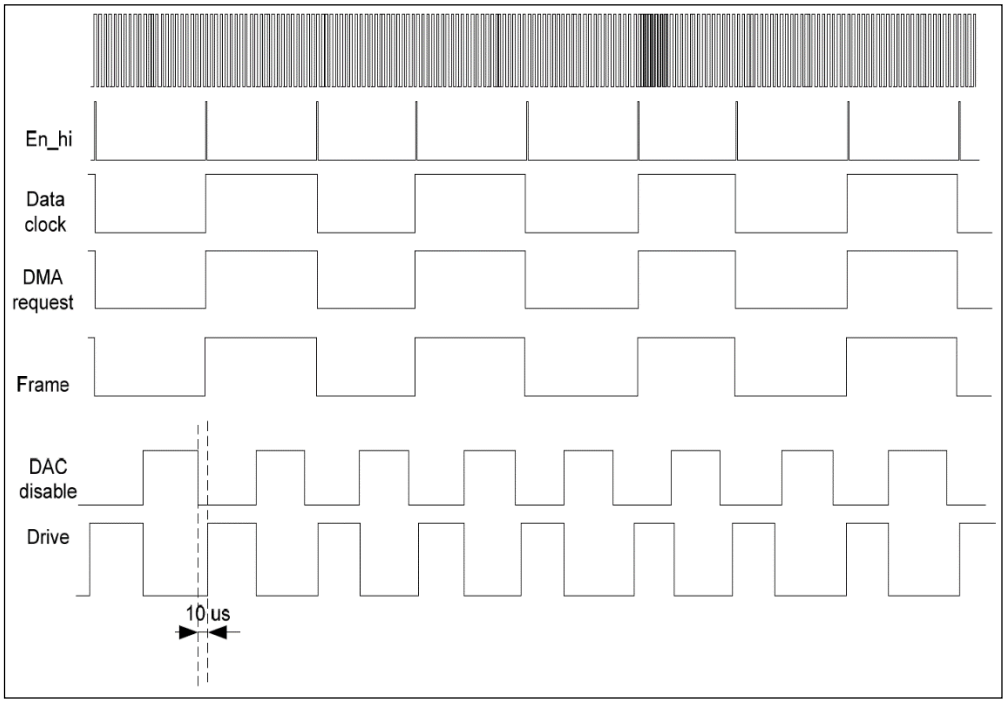
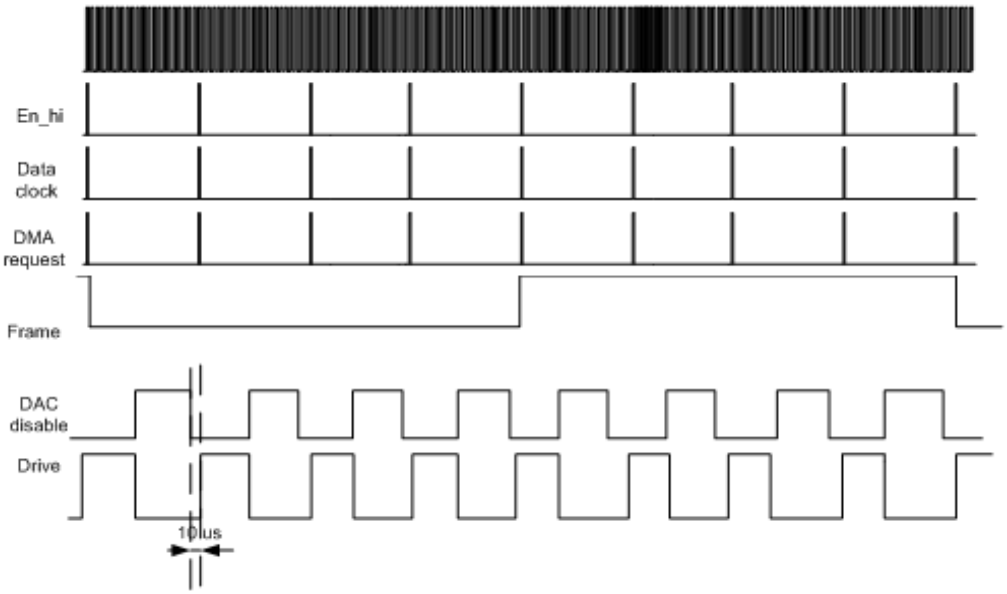


图 4 是 Seg_Display 组件（低功耗模式）UDB 生成的信号的波形（类型 B）：

图 4. 段显示器控制信号类型 B 波形（低功耗模式）



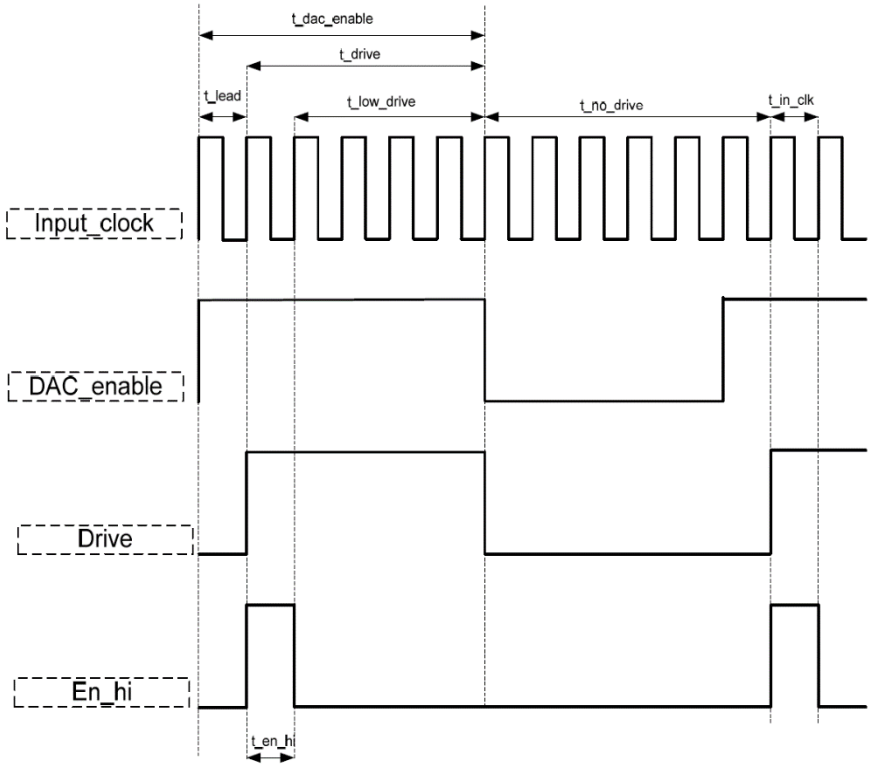
显示的信号针对“j”复用率情况。

时序计算

图 5 和下表说明了 UDB 生成信号的时序信息。在图 5 中，仅表示三种内部信号。其他信号可以由前几个图表衍生。图 5 基于低功耗模式和类型 B 波形。

DAC_disable 信号是通过反转 DAC_enable 内部信号所生成的。

图 5. 控制信号时序图示例



参数	时间	说明	计算
输入时钟频率	-	类型A波形的输入时钟频率值（由定制器自动设置）	f_in = f_frame_rate × N_common_lines × 256 × 2
		类型A波形的输入时钟频率值（由定制器自动设置）	f_in = f_frame_rate × N_common_lines × 256
输入时钟周期	t_in_clk	指定输入时钟周期	t_in_clk = 1/(f_in)
高电平驱动时间	t_en_hi	表示HiDrive有效的周期时间	t_en_hi = t_in_clk * N，其中N是1到253范围内的常量阶数值（始终运行模式），或者是1到247范围内的常量阶数值（低功耗模式）

参数	时间	说明	计算
	t_lead	指定LCD DAC设置时间。	~10 µs
	-	HiDrive无效时间	t_low_drive（始终有效） t_low_drive + t_no_drive（低功耗）
	t_drive	指定驱动时间	t_drive = t_en_hi + t_low_drive
	t_drive	指定驱动时间	t_drive = t_en_hi + t_low_duty_cycle
	t_low_drive	指定低电平驱动时间	t_low_drive = t_drive - t_en_hi
	t_no_drive	指定驱动信号置为低电平时的时间	t_no_drive = t_in_clk * 256 - t_drive
	t_dac_enable	指定LCD DAC打开的时间（仅针对低功耗模式）	t_dac_enable = t_drive + t_lead

用户特定配置

通过在组件自定义程序中更改时序参数，可以调整信号时序。

高电平驱动时间

默认情况下，根据操作模式，确定高电平驱动时间：

对于时钟运行模式，高电平驱动时间 = 128 × t_in_clk；

对于低功耗模式，高电平驱动时间 = 64 × t_in_clk。

此值由定制器自动计算。可以将此时间增加到最大值：

HiDriveTimemax = HiDriveTimemin × 253（始终运行模式）

HiDriveTimemax = HiDriveTimemin × 247（低功耗模式）

HiDriveTime 以输入时钟的一个周期值增量递增。这导致 en_hi signal 的有效时间被延长。

框图和配置

图 6. 段显示器组件原理图

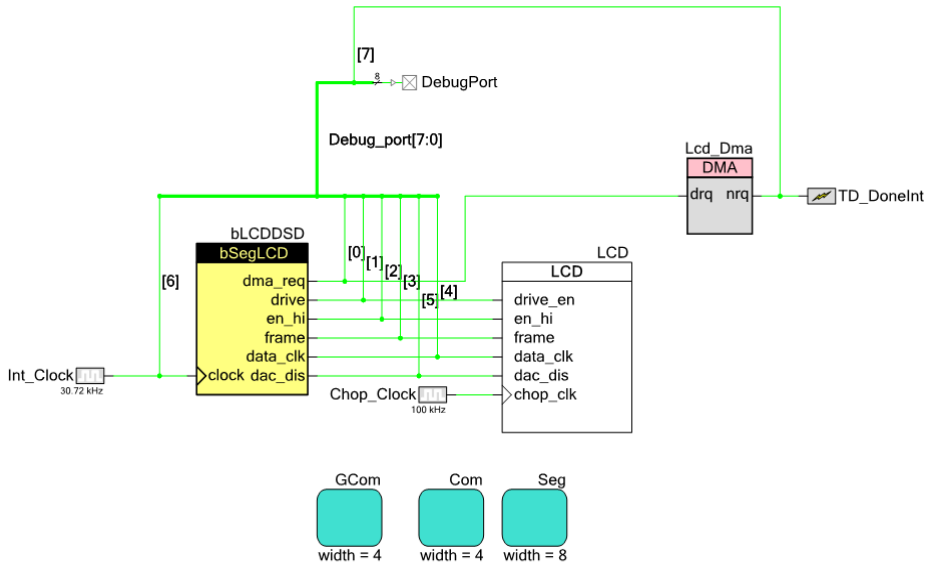
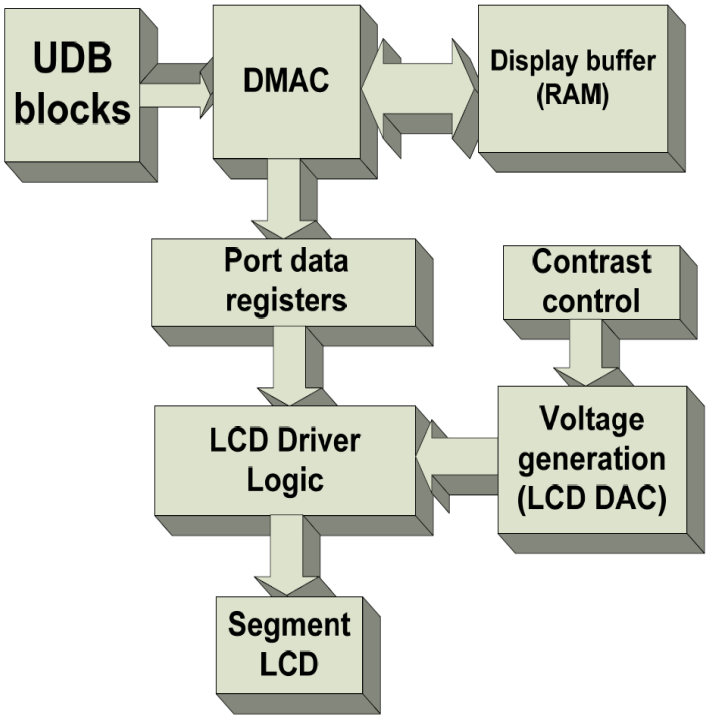


图 6 说明段显示器组件的内部原理图。该组件由以下器件组成：基本的段显示器组件、LCD 控制模块(LCD)组件、DMA 组件、2 个 LCD 端口，1 个数字端口、ISR 组件和 2 个时钟。

- 基本段显示器负责为 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器传输到 LCD 数据寄存器。
- LCD 组件将处理所需的 DSI 布线。此外，此模块还提供了如 *cyfitter.h* 文件中所定义的必要的寄存器名称。
- LCD 端口（GCom、Com 和 Seg）用于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。通用信号的 LCD 端口宽度限为 16 个引脚；段信号的 LCD 端口宽度限为 48 个引脚。
- DebugPort 端口组件仅用于调试。默认情况下，此组件被移除。

顶层架构

图 7. 顶层段显示器



寄存器

Seg_Display_CONTRAST_CONTROL

保持 LCD DAC 使用的偏置电压电平，用以生成正确的偏置电压。提供用来更改偏置电压电平的 API。

位	7	6	5	4	3	2	1	0
值	保留	对比度						

- contrast level（对比度）：偏置电压如上所述。

Seg_Display_LCDDAC_CONTROL

位	7	6	5	4	3	2	1	0
值	保留					DAC禁用	偏压选择	

- bias select（偏压选择）：此位将选择偏压。
- DAC disable（DAC 禁用）：如果不需要对比度控制，则禁用 LCD DAC。

Seg_Display_DRIVER_CONTROL

位	7	6	5	4	3	2	1	0
值	保留					反转	lo2	睡眠模式

- sleep mode（睡眠模式）：设置段显示器的睡眠模式。
- lo2：使能或禁用 LCD 驱动程序模块 loDrive 模式中的高电流模式
- invert （反转）：如果设置了此位，将反转段显示器上的所有数据。

Seg_Display_CONTROL

位	7	6	5	4	3	2	1	0
值	保留						控制复位	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。
- control reset（时钟复位）：执行组件数字部分的初始复位。

Seg_Display_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]）：此位字段将为 LCD 驱动器选择电压源。

资源

在 PSoC 5 中，段显示器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。此组件采用以下资源：

配置[1]	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
始终有效的模式	1	5	–	1	1	1
低功耗模式	2	9	–	1	1	1

1. 该配置正在使用类型 A 波形；如果使用类型 B 波形，将要多使用控制单元 1 和 28 字节的闪存。

API 存储器大小

根据编译器、器件、所使用的 API 数量以及组件的配置情况的不同，组件所用的存储器大小也不一样。下表提供了在某一器件配置中的所有 API 使用的存储器大小。

通过使用“释放”模式中的相应编译器，可以进行测量操作。在该模式下，存储器的大小得到优化。对于特定设计，可以分析编译器生成的映射文件，从而确定存储器的使用大小。

基本：低级 API 函数集，无任何高级助手 API

基本，7 段助手：低级 API 函数集 + 7 段助手 API

基本，14 段助手：低级 API 函数集 + 14 段助手 API

基本，16 段助手：低级 API 函数集 + 16 段助手 API

基本，点阵助手：低级 API 函数集 + 点阵助手高级 API

基本，条形图助手：低级 API 函数集 + 条形图助手高级 API

配置[2]	PSoC 3 （Keil_PK51）		PSoC 5（GCC）		PSoC 5LP（GCC）	
	闪存 字节	SRAM 字节	闪存 字节	SRAM 字节	闪存 字节	SRAM 字节
基本	N/A	N/A	1162	81	N/A	N/A
基本，7段助手	N/A	N/A	322	81	N/A	N/A
基本，14段助手	N/A	N/A	1528	81	N/A	N/A
基本，16段助手	N/A	N/A	1532	81	N/A	N/A
基本，点阵助手	N/A	N/A	2620	97	N/A	N/A
基本的条形图助手	N/A	N/A	1266	81	N/A	N/A

2. 该配置正在使用类型 A 波形；如果使用类型 B 波形，将要多使用控制单元 1 和 28 字节的闪存。

直流和交流电气特性

除非另有说明，否则这些规范的适用条件是：-40°C ≤ T_A ≤ 85 °C 和 T_J ≤ 100 °C。除非另有说明，否则这些规范的适用范围为 2.7 V 到 5.5 V。

直流电特性

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	总线时钟 = 3 MHz，VDDIO = VDDA = 3 V，4个共模信号，16个段信号，占空比为¼，频率为50 Hz，未连接玻璃屏	–	63	–	µA
I_{CC_SEG}	每个段驱动器的电流		–	148	–	µA
V_{BIAS}	LCD偏压范围（V_{BIAS}是指LCD DAC的主要输出电压(V0)）	对于驱动引脚，3 V ≤ V_{BIAS} ≤ V_{DDIO}	2.09	–	5.2	V
	LCD偏压步长大小	对于驱动引脚，3 V ≤ V_{BIAS} ≤ V_{DDIO}	–	25.8	–	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	V_{BIAS} ≤ V_{DDA} - 0.5 V	–	–	20	mV
I_{OUT}	每个段驱动器的输出驱动电流	V_{DDIO} = 5.5 V	90	–	165	µA

交流特性

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

组件更改

版本	更改说明	更改原因/影响
1.20	已添加了MISRA合规性章节。	此组件未进行MISRA合规性验证。
	更新了段显示器时钟的最新版本、中断及DMA组件。	
	解决了有关Seg_Display_WriteStringDotMatrix_n()和Seg_Display_WriteString16Seg_n() API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函数。
	解决了Seg_Display_WriteBargraph_n() API的相关问题。	此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Seg_Display_Write7SegNumber_n() API的相关问题。	在“no leading zeros”（无前导零）模式下使用此函数时，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。
1.10	对数据手册进行了少量纠正。	
	更新了组件资源的使用情况和API存储器占用数据。	
1.0.a	更正了数据手册	
1.0	第一版。	

注意：段显示器组件仅适用于 PSoC 5 器件。对于 PSoC 3 产品和 PSoC 5LP 器件，请使用段式 LCD 的版本为 3.10 或版本更高的组件。