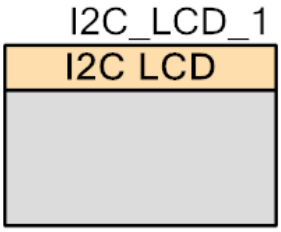
带有I2C接口的字符LCD（I2C LCD）

1.10



特性

- 基于 2 线 I²C 总线进行通信
- API 兼容当前的字符 LCD 组件
- 单个组件可在同一条 I²C 总线上驱动一个或多个 LCD
- 如果 PSoC 是 I²C 主控，则 I²C LCD 可以工作于此 I²C 总线上
- 支持 NXP PCF2119x 命令格式

概述

I²C LCD 组件可驱动带有 I²C 接口的 2 线式 16 字符 LCD。I²C LCD 组件是 I²C 主控组件的封装，并利用已有的 I²C 主控组件。如果一个工程没有 I²C 主控组件，则需添加一个以便操作。当调用某个 API 函数时，此函数将调用一个或多个 I²C 主控函数，以便与 LCD 进行通信。

何时使用 I²C LCD

I²C LCD 组件用于需要画面或文本显示的应用。此组件亦可用于需要显示字符的应用（在此类应用中，不能使用一个引脚端口上的七个连续 GPIO）。如果工程中已含有一个 I²C 主控，则不需要其他的 GPIO 引脚。

输入/输出连接

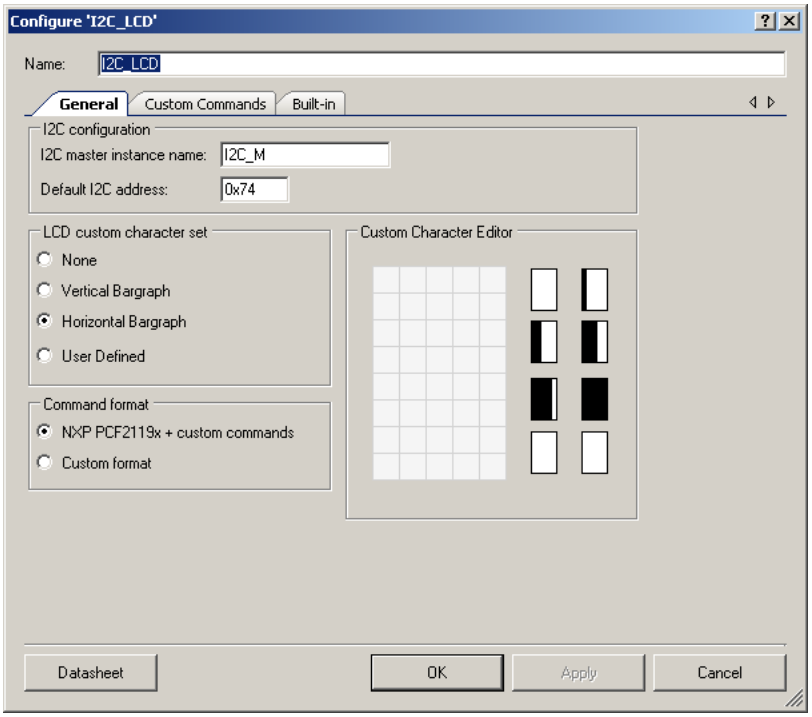
此组件属于软件组件，因此没有任何连接。

口件参数

将一个 I²C LCD 拖放到您的设计上，双击打开“Configure”（配置）对话框，如图 1 所示。

General（常规）选项卡

General 选项卡提供用于配置 I²C LCD 常规设置的选项。这些参数均是可用的。



I²C Master Instance Name（I²C 主控实例名称）

此参数设置为 I²C 主控组件的实例名称。I²C 主控必须拖放到原理图中，此组件才可运行。I²C LCD 组件用此名称访问 I²C 主控 API。LCD 也必须连接至由 I²C 主控组件控制的 I²C 总线上。可用的字符包括 A-Z、0-9、_。实例名称不能以下划线开始或结束。

Default I²C Address（默认 I²C 地址）

此参数用于设置 I²C LCD 的默认地址。在连接到 I²C 主控的 API 时，此地址用于正确访问 LCD。如果在总线上放置了多个 LCD，则 LCD 的 I²C 地址可在运行时改变。此参数的默认值是 0x40。

注意：对于随 PSoC 4 套件附上的 NXP LCD 模块，I²C 地址应设为 0x74。

LCD Custom Character Set （LCD 自定义字符集）

此参数可以选择以下选项：

- **None**（无）（默认）— 不对自定义字符执行任何操作。
- **Vertical Bar Graph**（垂直条形图）— 生成自定义字符和 API 以操控垂直条形图。
- **Horizontal Bar Graph**（水平条形图） — 生成自定义字符和 API 以操控水平条形图。
- **User Defined**（用户定义的）— 创建自定义字符和 API 以操控它们。

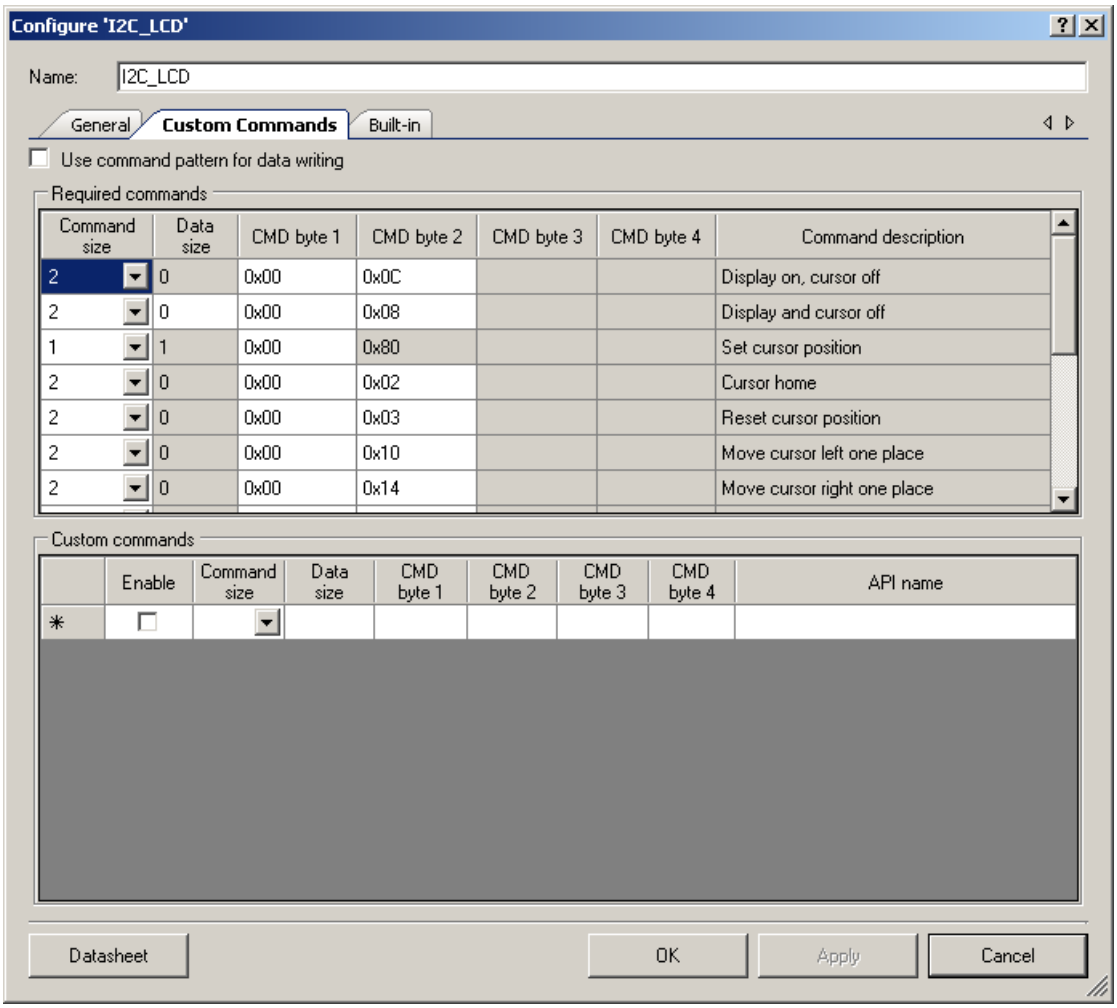
在组件载入了字符之后，可以使用 I2C_LCD_PutChar()函数和自定义字符常量（来自头文件）显示它们。

Command format（命令格式）

此参数有两个选项。如果选择了 **NXP PCF2119x + custom commands**（NXP PCF2119x + 自定义指令），则代表组件使用对 NXP PCF2119x 指令协议的嵌入支持，同时亦允许用户输入兼容 NXP 的 LCD 模块所支持的任意扩展（自定义）指令。可以在 **Custom Commands** 选项卡中输入自定义指令。如果您打算使用的 LCD 所支持的命令集并非 NXP PCF2119x，则应选择 **Custom format**（自定义格式）。在此种情况下，请在 **Custom Commands** 选项卡中输入需要的指令。您也可以在 **Custom commands** 表中输入所用 LCD 模块所支持的任意其他指令。

Custom Commands（自定义指令）选项卡

此选项卡包含一组选项，可以为组件添加自定义指令支持。



Use command pattern for data writing （使用指令模式进行数据写入）

此选项用于控制要通过 I2C_LCD_PrintString() 等函数在 LCD 上显示文本时，是否需要随数据发送命令模式。如果不勾选此复选框，待显示的数据将被直接发送。如果勾选此复选框，则会将指令模式与文本一起发送。NXP PCF2119x 兼容模块需要指令模式，因此，若选择此操作模式，则始终勾选此选项。对于自定义 LCD 模式，请查看相应模块的数据手册，以确定此选项的正确设置。

Required commands（必需命令）

此组合框包括一组您必须为操作组件定义的强制性指令。此表包含一组命令集，对兼容 NXP PCF2119x 的 LCD 模块是必须的。

#	指令描述
1	写入一个字节或多个字节数据
2	启用显示，关闭光标
3	关闭显示和光标
4	设置光标位置
5	光标归位
6	复位光标位置
7	将光标左移一位
8	将光标右移一位
9	开启下划线光标
10	启用显示，关闭光标，设置光标闪烁
11	启用显示，关闭光标，设置光标持续闪烁
12	清空屏幕
13	加载自定义字符
14	将显示设置为2行16个字符
15	右移光标/移位显示（光标位置自动增加）
16	左移光标/移位显示（光标位置自动减少）

如果选中 **NXP PCF2119x + custom commands**（NXP PCF2119x + 自定义指令）选项，**Required commands**（必需指令）表将变为灰色，因为必需指令的模式已嵌入组件中。如果选择 **Custom format**（自定义格式），则您需要为每个必需指令输入合适的指令模式。如果指定 LCD 模块不支持部分指令，则您必须输入一个伪指令模式。

Set cursor position（设置光标位置）

对于兼容 NXP PCF2119x 的 LCD 模块，0x80 以上的指令代码用于设置显示数据 RAM(DDRAM) 的地址。例如，地址 0x80 映射至 LCD 模块的第 0 行 0 列，地址 0x83 则映射至第 0 行 3 列。更多有关 DDRAM 映射至 LCD 位置的信息，请参阅 LCD 模块数据手册。

Command Size（指令大小）

此参数用于指定指令序列的大小（字节），不涉及参数或实际数据。

Data size（数据大小）

此参数用于定义根据具体指令的指令序列发送的数据大小。

CMD byte 1（指令字节 1）— CMD byte 4（指令字节 4）

此参数用于指定指令序列中字节 1-4 的值。如果“Command Size”（指令大小）小于四（最大值），则未使用的字节将在定制器中显示为灰色。**CMD byte 2**（指令字节 2）用于定义指令序列的唯一指令代码。此指令代码可用作 I2C_LCD_WriteControl()或 I2C_LCD_HandleCustom()函数的参数。

Command description（指令描述）

这是对指令功能的简要介绍，有助于将定制器的指令模式与 LCD 模块数据手册中的指令进行匹配。

默认情况下，在选择 **Custom commands** 参数时，required commands（必需指令）将会以 NXP PCF2119x 指令格式的指令模式填充。

Custom commands（自定义命令）

此部分可以为特定的指令设定自定义指令处理程序，亦可提供添加新指令的机制。在此组合框中的各列与 **Required Commands（必需命令）**的各列含义相同，但多了两列。

Enable（使能）

勾选此选项后，将在设计中使能相应的指令模式，即为指令模式提供 API 函数。

API name（API 名称）

此参数用于设置为指令模式生成（如果已使能）的 API 函数的名称。在此列中指定的 API 名称将以一个组件实例名称开头。此外，API 函数是否带参数取决于数据的大小。例如，如果 API 名称为“**Func1**”，带有组件实例名称“**I2C_LCD**”，且数据大小等于零，则 API 函数的原型如下：

```
void I2C_LCD_Func1(void);
```

如果**数据大小**为 1，则原型如下：

```
void I2C_LCD_Func1(uint8 cmdByte);
```

如果**数据大小**大于 1，则原型如下：

```
void I2C_LCD_Func1(uint8 cmdData[]);
```

cmdData[]的大小由**数据大小**确定。

请注意，您输入的 API 名称不能全部为大写字母。如果全是大写字母，则组件将会显示重新定义宏的警告。这是因为，对于每个自定义指令，组件会为指令代码生成#define 常量，其名称与 **API name** 中指定的全部大写的名称相同。

应用程序接口（API）

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

默认情况下，PSoC Creator 会将实例名称 I2C_LCD_1 分配给指定设计组件的第一个实例。您可以将其重新命名为遵循标识符语法规则的任何唯一值。实例名称会成为与该组件实例相关的每个全局函数名称、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 I2C_LCD。

函数

函数	说明
I2C_LCD_Start()	启动模块并将自定义字符集加载到LCD（如果它已定义）。
I2C_LCD_Stop()	关闭LCD
I2C_LCD_DisplayOn()	打开LCD模块的显示
I2C_LCD_DisplayOff()	关闭LCD模块的显示
I2C_LCD_PrintString()	将以空字符结尾的字符串逐个字符地打印到屏幕
I2C_LCD_PutChar()	将单个字符发送到当前位置的LCD模块数据寄存器。
I2C_LCD_Position()	设置光标的位置与提供的行和列匹配
I2C_LCD_WriteData()	将单个字节的数据写入LCD模块数据寄存器
I2C_LCD_WriteControl()	将单字节指令写入LCD模块控制寄存器
I2C_LCD_ClearDisplay()	从LCD模块的屏幕清除数据
I2C_LCD_Init()	执行组件正常工作所需的初始化
I2C_LCD_Enable()	打开显示
I2C_LCD_SetAddr()	此函数允许用户改变默认的LCD的I²C地址。
I2C_LCD_PrintInt8()	将 8 位值的双 ASCII 字符以十六进制表示形式打印到字符 LCD 模块。
I2C_LCD_PrintInt16()	将16位值的四ASCII字符以十六进制表示形式打印到字符LCD模块。
I2C_LCD_PrintNumber()	以左对齐ASCII字符的形式打印16位值的十进制值
I2C_LCD_HandleOneByteCommand()	此指令增加了对发送带有1个字节参数的自定义指令的支持
I2C_LCD_HandleCustomCommand()	发送有变量参数的指令。

如果您选择了自定义字体，则将在需要时提供以下这些可选函数。每个自定义字符集（无论是用户定义还是 PSoC Creator 生成）均提供 I2C_LCD_LoadCustomFonts()函数。I2C_LCD_LoadCustomFonts()函数可以用于将用户定义的字符或条形图字符加载到 LCD 硬件中。要加载由工具创建的自定义字体，您需要在使用此函数前，先将指向自定义字体的指针导入项

目（请参阅 I2C_LCD_LoadCustomFonts()说明）。默认情况下，I2C_LCD_Init()子程序将加载用户选择的自定义字体。在选择了条形图并启用了条形图的方便动态调整时，会生成绘制条形图指令。

可选自定义字体函数	说明
I2C_LCD_LoadCustomFonts()	将自定义字符加载到LCD模块中。
I2C_LCD_DrawHorizontalBG()	绘制水平条形图。仅当选择了条形图字符集时可用。
I2C_LCD_DrawVerticalBG()	绘制垂直条形图。仅当选择了条形图字符集时可用。

全局变量

函数	说明
I2C_LCD_initVar	I2C_LCD_initVar变量用于指示此组件是否进行了初始化配置。此变量前面加有组件名称。此变量被初始化为0，并在第一次调用I2C_LCD_Start()时设置为1。这可以实现组件初始化，并且当后续子程序调用I2C_LCD_Start()子程序时，无需重新初始化该组件。 此组件经过睡眠周期后，需要重新初始化此组件。因此，在执行睡眠函数I2C_LCD_Sleep()时，此变量设置为0；并在I2C_LCD_Wakeup()中执行的唤醒期间进行重新初始化设置。
I2C_LCD_customFonts[]	存储在定制器GUI中定义的自定义字符集。

void I2C_LCD_Start(void)

说明： 首次调用此函数时，此函数对LCD硬件模块进行如下初始化：

- 打开显示；
- 使能光标自动增加；
- 将光标复位到起始位置；
- 清空显示；
- 将自定义字符集加载到 LCD（如果在自定义程序的 GUI 中进行了定义）。将光标复位到起始位置。

以下所有对此函数的调用将开启LCD模块。

注意：此函数使用I²C主控将指令发送至显示。I²C主控必须进行初始化，且在调用此函数前，必须使能全局中断。

此外，如果您正在使用的兼容NXP的LCD模块与PSoC 4套件中的模块相同，那么在调用I2C_LCD_Start()前需要1ms的复位脉冲。请参阅I2C LCD示例项目，了解LCD模块的复位方法。

参数： 无

返回值： 无

其他影响： 无

void I2C_LCD_Stop(void)

说明： 关闭LCD屏幕的显示，但不停止I²C主控组件。

参数： 无

返回值： 无

其他影响： 无

void I2C_LCD_PrintString(char8 const string[])

说明： 从当前光标位置开始，将以空字符结尾的字符串写入屏幕。

参数： char8 const string[]:要在LCD模块屏幕上显示的以空字符结尾的ASCII字符数组。

返回值： 无

其他影响： 无

注意：用于 PSoC4 处理器模块的字符集是硬编码在 NXP PCF2119x LCD 模块中的，因此，有些字符将不会显示。如需详细信息，请参见功能说明一节。

void I2C_LCD_PutChar(char8 character)

说明： 将单个字符写入屏幕上的当前光标位置。用于通过字符的命名值显示自定义字符。（I2C_LCD_CUSTOM_0到I2C_LCD_CUSTOM_7）。

参数： char8 character：将在LCD模块屏幕上显示的ASCII字符。

返回值： 无

其他影响： 无

注意用于 PSoC4 处理器模块的字符集是硬编码在 NXP PCF2119x LCD 模块中的，因此，有些字符将不会显示。如需详细信息，请参见**功能**说明一节。

void I2C_LCD_Position(uint8 row, uint8 column)

说明： 将光标移动到参数**row**（行）和**column**（列）指定的位置。

参数： uint8 row：放置光标的行号。最小值为零。

uint8 column：放置光标的列号。最小值为零。

返回值： 无

其他影响： 无

void I2C_LCD_WriteData(uint8 dByte)

说明： 将数据写入LCD RAM的当前位置。当完成写入操作时，位置会根据指定的输入模式递增或递减。

参数： dByte：要写入LCD模块的字节值。

返回值： 无

其他影响： 无

void I2C_LCD_WriteControl(uint8 cByte)

说明： 将指令字节写入LCD模块。不同LCD模式可能具有自己的指令。请查看特定LCD数据手册以了解对该模式有效的命令。

参数： cByte：8位值，表示要加载到LCD模块指令寄存器中的指令值。下表中指定了有效指令参数：

值	说明
I2C_LCD_CLEAR_DISPLAY	清除显示
I2C_LCD_RESET_CURSOR_POSITION I2C_LCD_CURSOR_HOME	将光标和LCD返回到起始位置
I2C_LCD_CURSOR_LEFT	设置光标移动方向为左
I2C_LCD_CURSOR_RIGHT	设置光标移动方向为右
I2C_LCD_DISPLAY_CURSOR_ON	使能显示和光标
I2C_LCD_DISPLAY_ON_CURSOR_OFF	使能显示，光标关闭
I2C_LCD_CURSOR_WINK	使能显示，光标关闭，设置光标闪烁
I2C_LCD_CURSOR_BLINK	使能显示和光标，设置光标闪烁
I2C_LCD_CURSOR_SH_LEFT	左移光标/移位显示
I2C_LCD_CURSOR_SH_RIGHT	右移光标/移位显示
I2C_LCD_DISPLAY_2_LINES_5x10	将显示设置为2行10个字符
I2C_LCD_CURSOR_RIGHT I2C_LCD_CURSOR_AUTO_INCR_ON	右移光标/移位显示（光标位置自动增加）
I2C_LCD_CURSOR_LEFT	左移光标/移位显示（光标位置自动减少）

返回值： 无

其他影响： 无

void I2C_LCD_ClearDisplay(void)

说明： 清除屏幕内容并将光标位置复位为第零行零列。它使用合适的参数调用I2C_LCD_WriteControl()以激活显示。

参数： 无

返回值： 无

其他影响： 光标位置复位至0行0列。

void I2C_LCD_DisplayOff(void)

说明： 关闭显示，但是不复位LCD模块。它调用带有合适参数的函数I2C_LCD_WriteControl()来停止显示。

参数： 无

返回值： 无

其他影响： 无

void I2C_LCD_DisplayOn(void)

说明： 打开显示，但不进行初始化。它调用带有合适参数的函数I2C_LCD_WriteControl()来激活显示。

参数： 无

返回值： 无

其他影响： 如果之前使能，则将禁用光标下划线功能。

void I2C_LCD_Init(void)

说明： 对LCD模块进行如下初始化：

- 打开显示；
- 使能光标自动增加；
- 将光标复位到起始位置；
- 清空显示；
- 它还将自定义字符集加载到 LCD（如果在自定义程序的 GUI 中进行了定义）。将光标复位到起始位置

参数： 无

返回值： 无

其他影响： 无

void I2C_LCD_Enable(void)

说明： 打开显示。

参数： 无

返回值： 无

其他影响： 无

void I2C_LCD_PrintInt8(uint8 value)

说明： 将8位值以双ASCII字符表示形式打印到字符I²C LCD模块。

参数： uint8 value：要以十六进制ASCII字符打印的8位值。

返回值： 无

其他影响： 无

void I2C_LCD_PrintInt16(uint16 value)

说明： 将16位值以四ASCII字符表示形式打印到字符I2C LCD模块。

参数： uint16 value：要以十六进制ASCII字符打印的16位值。

返回值： 无

其他影响： 无

void I2C_LCD_PrintNumber(uint16 value)

说明： 以左对齐ASCII字符的形式打印16位值的十进制值。

参数： uint16 value：要以十进制ASCII字符打印的16位值。

返回值： 无

其他影响： 无

void I2C_LCD_HandleOneByteCommand(uint8 cmdId, uint8 cmdByte)

说明： 此指令增加了对发送带有1个字节参数的自定义指令的支持。

参数： uint8 cmdId：指令代码。

uint8 cmdByte：发送至I2C LCD的1个字节数据。

返回值： 无

其他影响： 无

void I2C_LCD_HandleCustomCommand(uint8 cmdId, uint8 dataLength, uint8 const cmdData[])

说明： 发送有变量参数的指令。

参数： uint8 cmdId：指令代码。

uint8 dataLength：发送至此指令的数据长度。

uint8 const cmdData[]：发送至I²C LCD的数据。

返回值： 无

其他影响： 无

void I2C_LCD_LoadCustomFonts(uint8 constcustomData[])

说明： 将八个自定义字符（条形图或用户定义的字体）加载到LCD模块中以在运行时使用自定义字体。此函数仅当在自定义程序中选择了自定义字符集时才可用。

参数： uint8 const customData[]：指向字节数组头的地址。数组长度应为64个字节，因为每个5x8字符需要8字节，共8个自定义字符集。

返回值： 无

其他影响： 覆盖以前已存储在LCD模块中的任何自定义字符。

void I2C_LCD_DrawHorizontalBG(uint8 row, uint8 column, uint8 maxCharacters, uint8 value)

说明： 绘制水平条形图。仅当选择了水平或垂直条形图时才可用。

参数： uint8 row：条形图中第一个字符的行。

uint8 column：条形图中第一个字符的列。

uint8 maxCharacters：条形图使用的所有字符数量。根据条形图选择高度或宽度。每个字符为5像素宽和8像素高。

uint8 value：要绘制的阴影像素数。不能超过条形图的总像素长度（高度）。

返回值： 无

其他影响： 无

void I2C_LCD_DrawVerticalBG(uint8 row, uint8 column, uint8 maxCharacters, uint8 value)

说明： 绘制垂直条形图。仅当选择了水平或垂直条形图时才可用。

参数： uint8 row：条形图中第一个字符的行。

uint8 column：条形图中第一个字符的列。

uint8 maxCharacters：条形图使用的所有字符数量。根据条形图选择高度或宽度。每个字符为5像素宽和8像素高。

uint8 value：要绘制的阴影像素数。不能超过条形图的总像素长度（高度）。

返回值： 无

其他影响： 无

宏

为简化在一条总线上使用多个 LCD 的系统的应用，为每个 API 添加了一个宏，包括 I²C LCD 地址。例如，指令“I2C_LCD_Position(row, col)”的地址宏是“I2C_LCD_AddrPosition(addr, row, col)”。参数“addr”是 LCD 的 I2C 地址。调用宏函数相当于在调用本地函数前先调用 I2C_LCD_SetAddr()函数。例如，调用宏函数：

```
I2C_LCD_AddrPosition(addr, row, col);
```

此函数和先后调用以下两个函数完全一样：

```
I2C_LCD_SetAddr(addr);
I2C_LCD_Position(row, col);
```

成功调用后，I²C LCD 地址将予以保留，直到另一个宏或 I2C_LCD_SetAddr()函数被调用。下表总结了所支持的宏。

函数	地址宏
I2C_LCD_Start()	I2C_LCD_AddrStart()
I2C_LCD_Stop()	I2C_LCD_AddrStop()
I2C_LCD_DisplayOn()	I2C_LCD_AddrDisplayOn()
I2C_LCD_DisplayOff()	I2C_LCD_AddrDisplayOff()
I2C_LCD_PrintString()	I2C_LCD_AddrPrintString()
I2C_LCD_PutChar()	I2C_LCD_AddrPutChar()
I2C_LCD_Position()	I2C_LCD_AddrPosition()
I2C_LCD_WriteData()	I2C_LCD_AddrWriteData()
I2C_LCD_WriteControl()	I2C_LCD_AddrWriteControl()
I2C_LCD_ClearDisplay()	I2C_LCD_AddrClearDisplay()
I2C_LCD_Init()	I2C_LCD_Init()
I2C_LCD_Enable()	I2C_LCD_AddrEnable()
I2C_LCD_LoadCustomFonts()	I2C_LCD_AddrLoadCustomFonts()
I2C_LCD_DrawHorizontalBG()	I2C_LCD_AddrDrawHorizontalBG()
I2C_LCD_DrawVerticalBG()	I2C_LCD_AddrDrawVerticalBG()
I2C_LCD_SetAddr()	NA
I2C_LCD_PrintInt8()	I2C_LCD_AddrPrintInt8()
I2C_LCD_PrintInt16()	I2C_LCD_AddrPrintInt16()
I2C_LCD_PrintNumber()	I2C_LCD_AddrPrintNumber()
I2C_LCD_HandleCustomCommand()	I2C_LCD_AddrHandleCustomCommand()

生成这些宏的目的是为了简化对组件 GUI 中定义的自定义指令的访问。以下显示的名称对应的是在 **Custom command** 表 API 名称中输入的“Func1”。

- **`I2C_LCD_Func1()`** — 类似函数宏，用于自定义指令的**数据大小（Data size）**参数设定为零的情况。
- **`I2C_LCD_Func1(cmdByte)`** — 类似函数宏。用于自定义指令的**数据大小（Data size）**参数等于 1 的情况。cmdByte 是此指令的 1 个字节参数。
- **`I2C_LCD_Func1(cmdData)`** — 类似函数宏。用于自定义指令的**数据大小（Data size）**参数大于 1 的情况。cmdData 是**数据大小（Data size）**参数规定的数组。

- **I2C_LCD_FUNC1** — 定义指令代码（**CMD byte 2** 中输入的值），可用作 I2C_LCD_WriteControl()、I2C_LCD_HandleOneByteCommand() 或 I2C_LCD_HandleCustomCommand() 的参数。

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 — 仅适用于该组件的偏差。本节提供了有关组件特定偏差的信息。*《系统参考指南》* 的 MISRA 合规性章节中介绍了项目偏差以及有关 MISRA 合规性验证环境的信息。

I²C LCD 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	偏差说明
19.7	必须	函数应优先于类函数宏使用。	组件中使用的类函数宏用于处理UDB中实现的I²C主控和SCB中实现的I²C主控的API函数名称差异。 此外，类函数宏亦用于简化自定义指令的使用。

固件源代码示例

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，请右击组件并打开“Find Example Project”对话框来查看组件实例。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** 选项来限定可选的项目列表。

如需更多信息，请参阅 PSoC Creator 帮助中的“Find Example Project”（查找示例项目）主题。

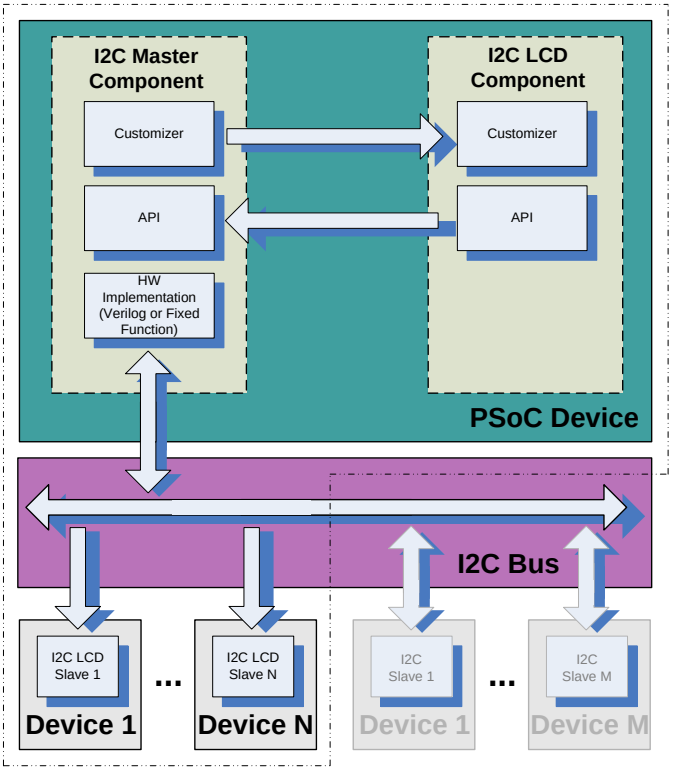
功能说明

框图和配置

I²C LCD 组件是 I²C 主控组件的封装，并利用已有的 I2C 主控组件。此组件为生成 NXP PCF2119x 芯片定义的数据操作提供库，亦支持生成自定义 I²C LCD 模块，此类模块类似于 NXP PCF2119x 指令格式。使用 I²C LCD 组件，可以通过地址宏与多个同一类型的 LCD 进行通信。此

外，您还可以使用多个组件为 LCD 添加不同指令格式的支持。所有 I²C 组件可共享同一个 I²C 主控组件。

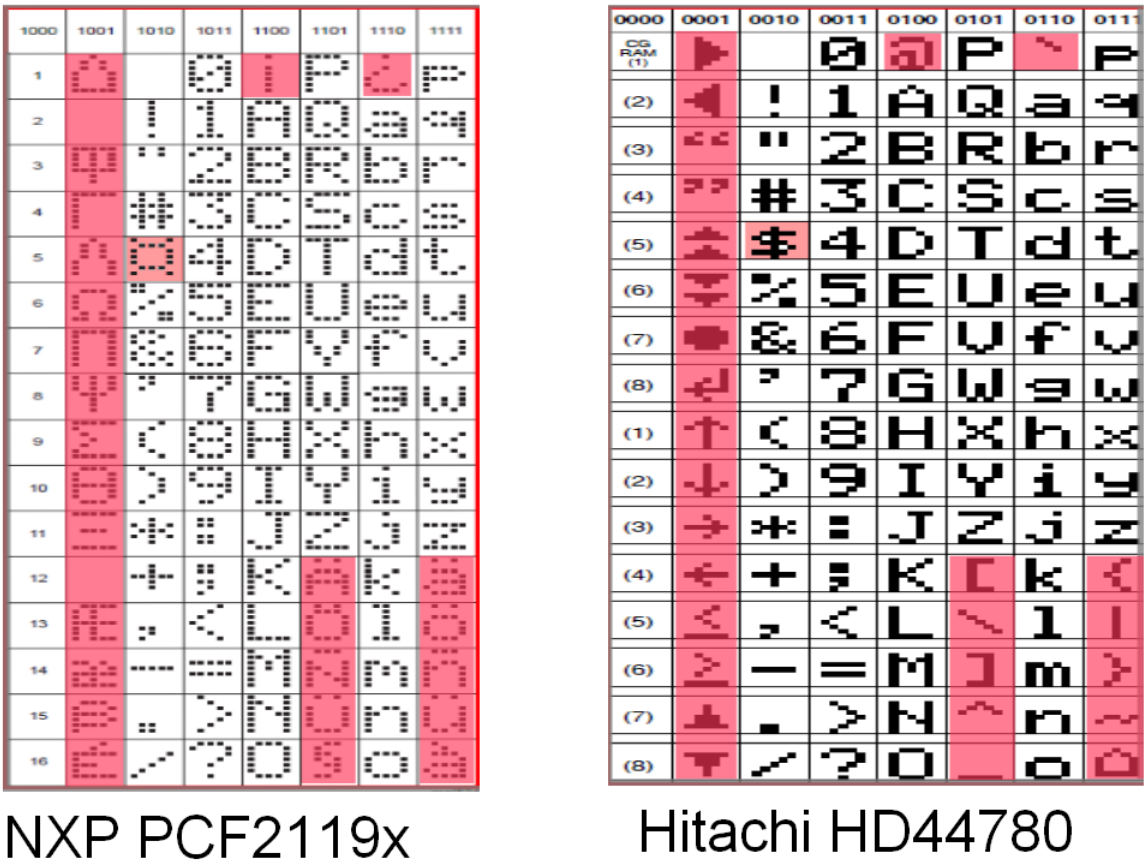
图 3. I²C LCD 和 I²C 主控组件的交互



I²C LCD 和字符 LCD

I²C LCD 组件的设计旨在使用现有 PSoC Creator 字符 LCD 组件的用户界面。尽管 I²C LCD 和字符 LCD 有相同之处，但在输出时则有所不同。由于用于 PSoC4 处理器模块的字符集是硬编码至 NXP PCF2119x LCD 模块的，因此，LCD 上将不能显示某些字符，比如括号（“[”“]”）。图 4 展示了两类 LCD 的字符列表，不同的字符用红色标注。

图 4. NXP PCF2119x 和日立 HD44780 LCD 模块的字符集



口源

I²C LCD 组件是软件组件，不使用硬件资源。

API 存储器的使用情况

根据不同的编译器、设备、所用 API 的数量以及组件配置，组件存储器的使用有着显著的不同。下表列出了在指定组件配置中可用的所有 API 的存储器使用情况。

使用 release 模式中配置的关联编译器进行了测量，测量中采用了大小的优化设置。有关特定的设计，可分析编译器生成的映射文件以确定存储器使用情况。

配置	PSoC 3 （Keil_PK51）		PSoC 4（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
无	1059	259	748	268	772	265
水平或垂直条形图	1654	259	1136	268	1148	265
用户定义的	1200	259	870	268	894	265

寄存器

N/A

直流和交流电气特性

下面各值表示其期望性能，它们是基于初始特性数据得到的。

参数	描述	条件	最小值	典型值	最大值	单位
NumLCD	可放置于I²C总线上的LCD模块数量	由I²C总线电容和上拉电阻确定。	1	-	>10	
t_{fnc_ex}[1]						
	I2C_LCD_Start()	数据速率 = 100kbps	-	14.8	-	ms
	I2C_LCD_Stop()		-	1.94	-	ms
	I2C_LCD_DisplayOn()		-	0.44	-	ms
	I2C_LCD_DisplayOff()		-	1.94	-	ms

[1] 在BUS_CLK设置为24 MHz时，PSoC3设备测得的数值

参数	描述	条件	最小值	典型值	最大值	单位
	I2C_LCD_PrintString()		-	1.56[2]	-	ms
	I2C_LCD_PutChar()		-	0.43	-	ms
	I2C_LCD_Position()		-	0.43	-	ms
	I2C_LCD_WriteData()		-	0.43	-	ms
	I2C_LCD_WriteControl()		-	0.49	-	ms
	I2C_LCD_ClearDisplay()		-	1.98	-	ms
	I2C_LCD_Init()		-	14.4	-	ms
	I2C_LCD_Enable()		-	0.44	-	ms
	I2C_LCD_SetAddr()		-	2	-	us
	I2C_LCD_HandleCustomCommand()		-	0.62[3]	-	ms
	I2C_LCD_HandleOneByteCommand()		-	0.62	-	ms
	I2C_LCD_LoadCustomFonts()		-	9.0	-	ms
	I2C_LCD_DrawHorizontalBG()		-	1.94[4]	-	ms
	I2C_LCD_DrawVerticalBG()		-	3.46[5]	-	ms
	I2C_LCD_PrintInt8()		-	0.55	-	ms
	I2C_LCD_PrintInt16()		-	1.1	-	ms
	I2C_LCD_PrintNumber()		-	1.03	-	ms

[2] 此值为打印包含10个字符的字符串的数值

[3] 此值在2个字节指令模式（数据大小= 1字节）下计算得出

[4] 此值为打印10条形图所得的数值

[5] 此值为打印4条形图所得的数值

口件更改

本节列出了各版本的主要组件更改内容。

版本	更改内容	更改/影响的原因
1.10	修复了有关静态函数的警报 — 修复了（已声明但未定义的）I2C_LCD_SendCmd()。	当将指令格式选项设置为“Custom format”时，会显示该警报。
1.0a	将兼容NXP的LCD模块的复位注释添加到I2C_LCD_Start()说明内。	解决有关兼容NXP的LCD模块的问题。
1.0	版本1.0是I²C LCD组件的首次发行版。	

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。