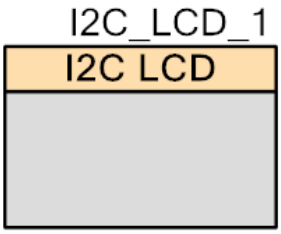
带有 I²C 接口的字符 LCD (I2C LCD)

1.0



特性

- 在 2 线式 I²C 总线上进行通信
- API 兼容当前的字符 LCD 器件
- 单个器件可在同一条 I²C 总线上驱动一个或多个 LCD
- 如果 PSoC 是 I²C 主控，则 I²C LCD 可以共存于已有的 I²C 总线上
- 支持 NXP PCF2119x 命令格式

概述

I²C LCD 器件通过驱动带有 I²C 接口的 2 线式 16 字符 LCD。I²C LCD 器件是 I²C 主控器件的封装，并利用已有的 I²C 主控器件。如果一个工程没有 I²C 主控器件，则需安装一个以便运行。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 器件用于需要画面或文本显示的应用。此器件亦用于需要显示字符的应用，但在此类应用中，不能使用一个通用 I/O 端口上的七个连续通用 I/O。如果工程中已含有一个 I²C 主控，则不需要其他的通用 I/O 引脚。

输入/输出连接

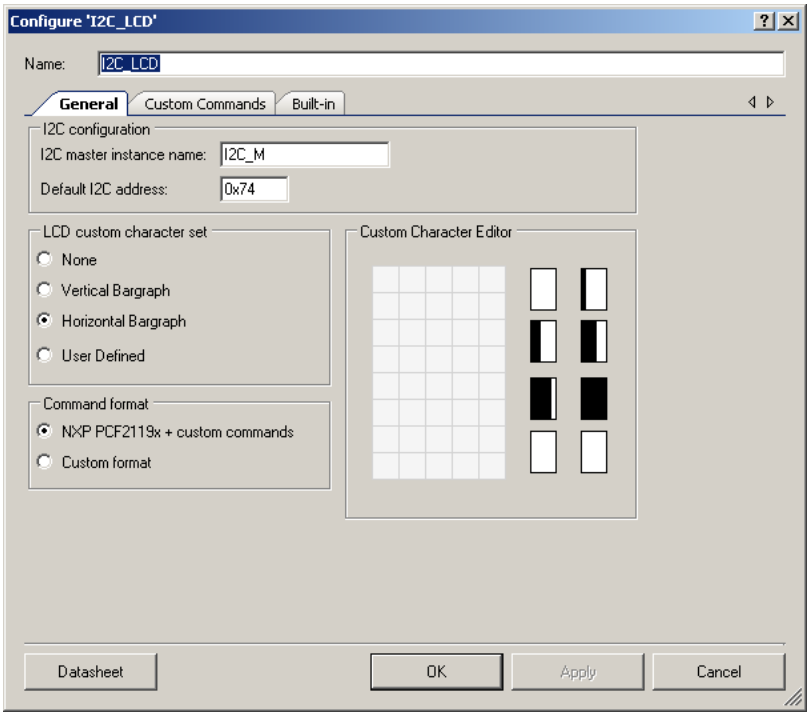
此器件属于软件器件，因此没有任何连接。

组件参数

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（一般）选项卡

General（一般）选项卡提供用于配置 I²C LCD 一般设置的选项。这些参数均已提供。



I²C Master Instance Name（I2C 主控实例名称）

此参数设置为 I²C 主控器件的实例名称。I²C 主控必须常驻于原理图中，此器件才可运行。用此名称访问 I²C 主控 API。LCD 也必须连接至由 I²C 主控器件控制的 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 地址可能在运行时有所改变。此参数的默认值是 0x40。

LCD Custom Character Set（LCD 自定义字符集）

通过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。

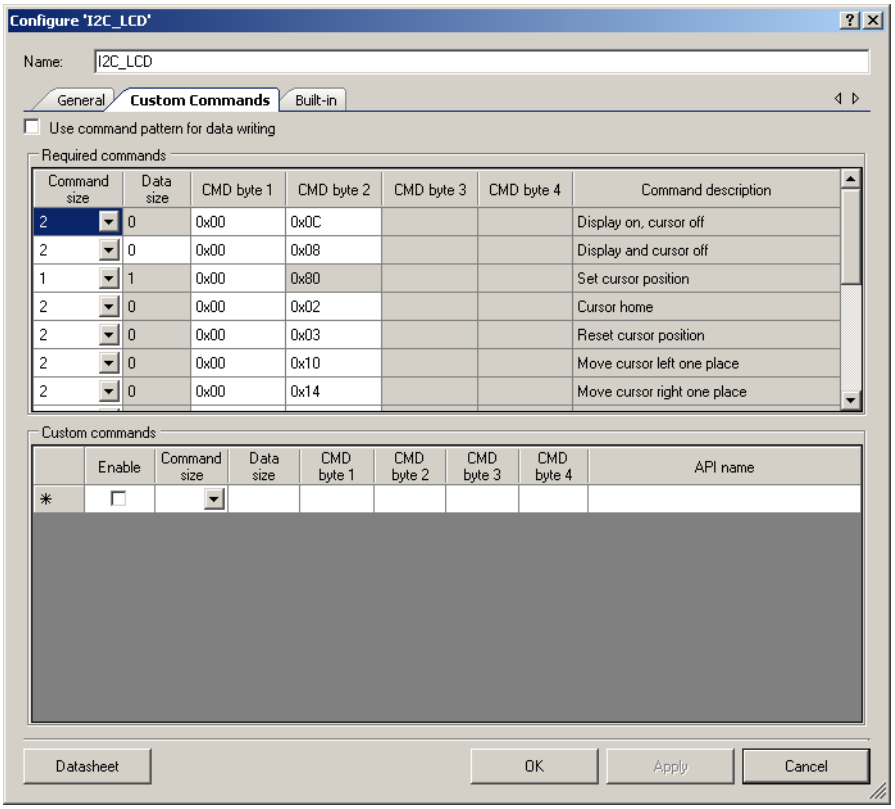
在器件载入了字符之后，可以使用 I2C_LCD_PutChar() 函数和自定义字符常量（来自头文件）显示它们。

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 + 自定义命令），则代表器件使用对 NXP PCF2119x 命令协议的嵌入支持，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）命令。可以在 **Custom Commands** （自定义命令）选项卡中输入自定义命令。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** （自定义命令）选项卡中输入需要的命令。您也可以在 **Custom commands** （自定义命令）表中输入指定 LCD 模块所支持的任意其他命令。

Custom Commands（自定义命令）选项卡

此选项卡包含一组选项，可以为器件添加自定义命令支持。



Use command pattern for data writing（使用命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，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，则会将命令模式与文本一起发送。NXP PCF2119x 兼容模块需要命令模式，因此，要启动运动模式，应始终勾选此选项。如需获取自定义 LCD 模式支持，请查看相应模块的数据表，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器件定义的强制性命令。此表包含一组命令集，对兼容 NXP PCF2119x 的 LCD 模块是必须的。

#	命令描述
1	写入一字节或多字节数据
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 + 自定义命令）选项，**Required commands**（必需命令）表将变为灰色，因为必需命令的模式已嵌入器件中。如果选择 **Custom format**（自定义格式），则您需要为每个必需命令输入合适的命令模式。如果指定 LCD 模块不支持部分命令，则您必须输入一个伪命令模式。

设置光标位置

对于兼容 NXP PCF2119x 的 LCD 模块，大于等于 0x80 的命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 0 行及 0 列，地址 0x83 则映射至 0 行及 3 列。如需获取 DDRAM 与 LCD 位置映射的详细信息，请查看 LCD 模块数据表。

Command Size（命令大小）

此参数用于指定命令序列的大小（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命令的命令序列发送的数据大小。

CMD byte 1（CMD 字节 1）–CMD byte 4（CMD 字节 4）

此参数用于指定命令序列中字节 1 - 4 的值。如果“Command Size”（命令大小）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**（CMD 字节 2）用于定义命令序列的唯一命令代码。此命令代码可用作 I2C_LCD_WriteControl() 或 I2C_LCD_HandleCustom() 函数的参数。

Command description（命令描述）

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

默认情况下，在选择 **Custom commands**（自定义命令）参数时，必需命令将会以 NXP PCF2119x 命令格式的命令模式填充。

Custom commands（自定义命令）

此部分可以为特定的命令设定自定义命令处理程序，亦可提供添加新命令的机制。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（启用）

勾选此选项后，将在设计中启用相应的命令模式。这代表为命令模式提供 API 函数。

API name（API 名称）

此参数用于指定为命令模式生成（如果已启用）的 API 函数的名称。在此列中指定的 API 名称将以一个器件实例名称开头。此外，API 函数是否将带参数亦取决于数据的大小。例如，如果 API 名称为“**Func1**”，带有器件实例名称“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，则器件将会显示重新定义宏的警告。这是因为，对于每个自定义命令，器件会为命令代码生成 #define 常量，其名称与 **API name**（API 名称）中指定的全部大写的名称相同。

应用程序编程接口

应用程序编程接口 (API) 子程序允许您使用软件配置器件。此表列出了每个函数的接口，并进行了说明。以下各节将更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 分配给指定设计器件的第一个实例。您可以将其重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

函数	说明
I2C_LCD_Enable()	打开显示
I2C_LCD_SetAddr()	此函数允许用户改变默认的 LCD 的I^2C 地址。
I2C_LCD_PrintInt8()	将 8 位值的双 ASCII 字符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()	此命令增加了对发送带有 1 字节参数的自定义命令的支持
I2C_LCD_HandleCustomCommand()	发送有变量参数的命令。

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，会生成绘制条形图命令。

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 变量用于设置此器件的初始配置。此变量前面加有器件名称。此变量被初始化为 0，并在第一次调用 I2C_LCD_Start() 时设置为 1。这可以实现器件初始化，同时无需重新初始化当后续子程序调用 I2C_LCD_Start() 例程时。 当此器件经过睡眠周期时，需要重新初始化此器件。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；
- 启用自动光标增加；
- 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。

以下所有对此函数的调用将开启 LCD 模块。

注意：此函数使用 I^2C 主控将命令发送至显示。I^2C 主控必须进行初始化，且在调用此函数前，必须启用全局中断。

参数： 无

返回值： 无

副作用： 无

void I2C_LCD_Stop(void)

说明： 关闭 LCD 屏幕的显示，但不停止 I^2C 主控器件。

参数： 无

返回值： 无

副作用： 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： char8const 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

副作用： 无

注意：用于 PSoC4 处理器模块的字符集是在 NXP PCF2119x LCD 模块上的硬编码，因此，有些字符将不会显示。如需详细信息，请参见**功能描述**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 字符：在 LCD 模块屏幕上将显示的 ASCII 字符。

返回值： 无

副作用： 无

注意：用于 PSoC4 处理器模块的字符集是硬编码是在 NXP PCF2119x LCD 模块上的硬编码，因此，有些字符将不会显示。如需详细信息，请参见**功能描述**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 行：放置光标的行号。最小值为零。

uint8 列：放置光标的列号。最小值为零。

返回值： 无

副作用： 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

副作用： 无

void I2C_LCD_WriteControl(uint8 cByte)

说明： 将命令字节写入 LCD 模块。不同 LCD 模式可能具有自己的命令。请查看具体 LCD 数据表，了解对该模式有效的命令。

参数： cByte：8 位值，表示要加载到 LCD 模块命令寄存器中的命令值。下表中指定了有效命令参数：

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 返回到主位置
I2C_LCD_CURSOR_LEFT	设置左光标移动方向
I2C_LCD_CURSOR_RIGHT	设置右光标移动方向
I2C_LCD_DISPLAY_CURSOR_ON	启用显示和光标
I2C_LCD_DISPLAY_ON_CURSOR_OFF	启用显示，光标关闭
I2C_LCD_CURSOR_WINK	启用显示，光标关闭，设置光标闪烁
I2C_LCD_CURSOR_BLINK	启用显示和光标，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

副作用： 无

void I2C_LCD_ClearDisplay(void)

说明： 清除屏幕内容并将光标位置复位为零行和零列。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

副作用： 光标位置复位至 0 行 0 列。

void I2C_LCD_DisplayOff(void)

说明： 关闭显示，但是不以任何方式复位 LCD 模块。它调用带有合适的参数的函数 I2C_LCD_WriteControl() 来停用显示。

参数： 无

返回值： 无

副作用： 无

void I2C_LCD_DisplayOn(void)

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

参数： 无

返回值： 无

副作用： 如果之前启用，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；
- 启用自动光标增加；
- 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

副作用： 无

void I2C_LCD_Enable(void)

说明： 打开显示。

参数： 无

返回值： 无

副作用： 无

void I2C_LCD_PrintInt8(uint8 value)

说明： 将 8 位值的双 ASCII 字符表示形式打印到字符 I^2C LCD 模块。

参数： uint8 值：要以十六进制 ASCII 字符打印的 8 位值。

返回值： 无

副作用： 无

void I2C_LCD_PrintInt16(uint16 value)

说明： 将 16 位值的四 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

副作用： 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：要以十进制数字的 ASCII 字符打印的 16 位值。

返回值： 无

副作用： 无

void I2C_LCD_HandleOneByteCommand(uint8 cmdId, uint8 cmdByte)

说明： 此命令增加了对发送带有 1 字节参数的自定义命令的支持。

参数： uint8 cmdId：命令代码。
uint8 cmdByte：发送至 I2C LCD 的一字节数据。

返回值： 无

副作用： 无

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

说明： 发送有变量参数的命令。

参数： uint8 cmdId：命令代码。
uint8 dataLength：发送至此命令的数据长度。
const cmdData[]：发送至 I^2C LCD 的数据。

返回值： 无

副作用： 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。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 constcustomData[]: 指向字节数组头的指针。数组长度应为 64 个字节，因为 5x8 字符需要每个字符 8 字节。

返回值： 无

副作用： 覆盖可能已存储在 LCD 模块中的任何以前的自定义字符。

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 行：条形图中第一个字符的行。

uint8 列：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。根据条形图选择表示高度或宽度。每个字符为 5 像素宽和 8 像素高。

uint8 值：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

副作用： 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 行：条形图中第一个字符的行。

uint8 列：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。根据条形图选择表示高度或宽度。每个字符为 5 像素宽和 8 像素高。

uint8 值：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

副作用： 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 I^2C LCD 地址。例如，命令“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);
```

如预期结果一样，I²C LCD 地址将予以保留，直到另一个宏或 I2C_LCD_SetAddr() 函数本身被调用。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器件 GUI 中定义的自定义命令的访问。以下显示的名称对应的是在 **Custom command**（自定义命令）表 API 名称中输入的 “Func1”。

- **`I2C_LCD_Func1()`** – 类函数宏，用于自定义命令的**数据大小**参数设定为零的情况。
- **`I2C_LCD_Func1(cmdByte)`** – 类函数宏。用于自定义命令的**数据大小**参数等于一的情况。cmdByte 是此命令的一字节参数。
- **`I2C_LCD_Func1(cmdData)`** – 类函数宏。用于自定义命令的**数据大小**参数大于一的情况。cmdData 是**数据大小**参数规定的数组。

- **I2C_LCD_FUNC1** – 定义命令代码（**CMD 字节 2** 中输入的值），可用作 I2C_LCD_WriteControl()、I2C_LCD_HandleOneByteCommand() 或 I2C_LCD_HandleCustomCommand() 的参数。

MISRA 兼容性

本节介绍了本组件与 MISRA-C:2004 的 兼容和偏差情况。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。本节提供了有关组件特定偏差的信息。*系统参考指南*的 MISRA 兼容性章节中介绍项目偏差以及有关 MISRA 兼容性验证环境的信息。

I^2C LCD 具有以下特定偏差：

MISRA-C:2004 规则	规则类（必需/参考）	规则描述	偏差描述
19.7	必需	函数应优先于类函数宏使用。	器件中使用的类函数宏用于处理 UDB 中实现的 I^2C 主控和 SCB 中实现的 I^2C 主控的 API 函数名称差异。 此外，类函数宏亦用于简化自定义命令的用法。

固件源代码示例

PSoC Creator 在“查找示例项目”对话框中提供了大量包括原理图和代码的例子项目。要获取器件特定的示例，请打开器件目录中的对话框或原理图中的器件实例。要获取通用的示例，请打开 Start Page（开始页）或 **File**（文件）菜单中的对话框。根据需要，使用对话框中的 **Filter Options**（筛选选项）可缩小可选项目的列表。

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

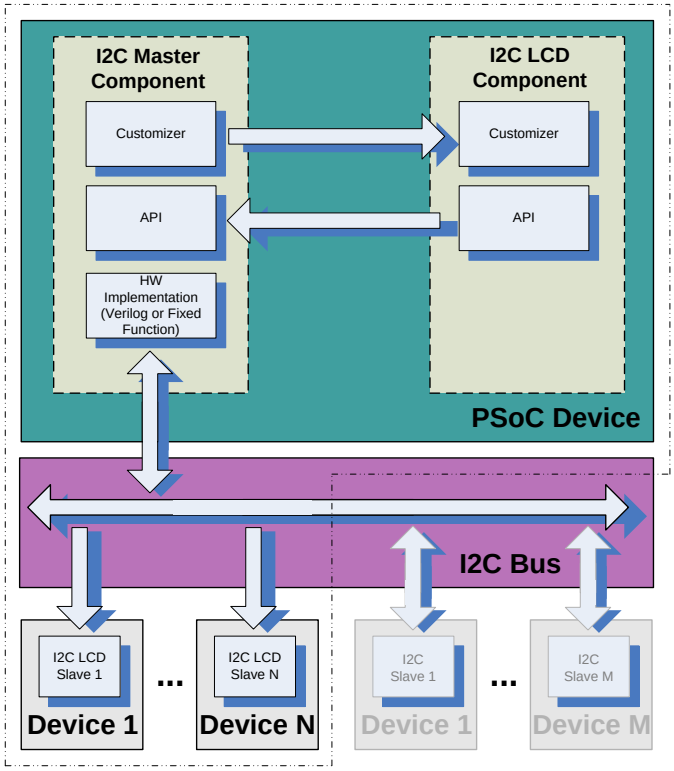
功能描述

框图和配置

I^2C LCD 器件是 I^2C 主控器件的包装器。此器件为生成 NXP PCF2119x 芯片定义的数据操作提供库，亦支持生成自定义 I^2C LCD 模块，此类模块类似于 NXP PCF2119x 命令格式。使用 I^2C

LCD 器件，可以通过地址宏与多个同一类型的 LCD 进行通信。此外，您还可以使用多个器件为 LCD 添加不同命令格式的支持。所有 I²C 器件可共享同一个 I²C 主控器件。

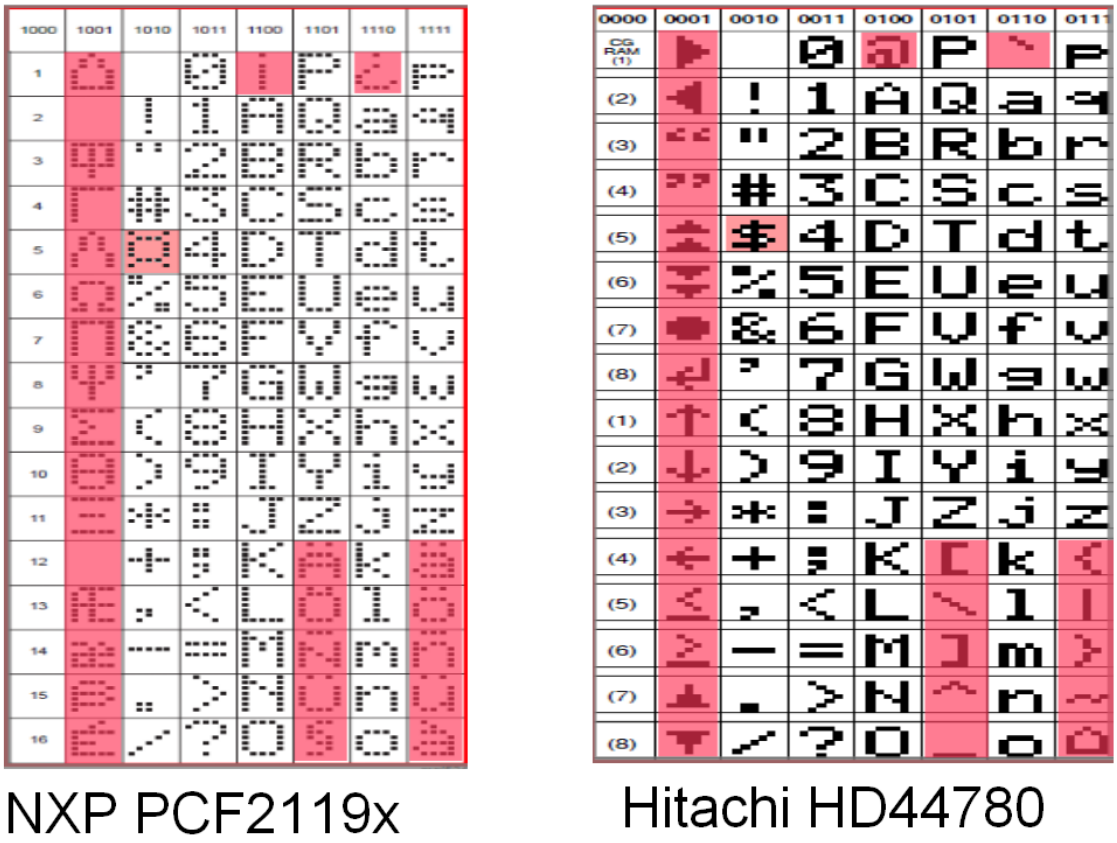
图 1. I²C LCD 和 I²C 主控器件的交互



I²C LCD 和字符 LCD

I²C LCD 器件的设计旨在使用现有 PSoC Creator 字符 LCD 器件的用户界面。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在 NXP PCF2119x LCD 模块上的硬编码，因此，LCD 上将不会显示某些字符，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 2. NXP PCF2119x 和日立 HD44780 LCD 模块的字符集



资源

I²C LCD 器件是软件器件，不使用硬件资源。

API 存储器使用

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

使用发布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最佳设置。有关特定的设计，可分析编译器生成的映射文件以确定内存使用情况。

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

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
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

无

直流和交流电气特性

下面的值表示了预计性能，它们基于初始特性数据。

参数	说明	条件	最小值	典型值	最大值	单位
NumLCD	可放置于 I^2C 总线的 LCD 上的 LCD 模块数量	由 I^2C 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
	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

[1] 在 BUS_CLK 设置为 24 MHz 时，PSoC3 设备接收的数值

[2] 此值为打印包含 10 个字符的字符串的数值

参数	说明	条件	最小值	典型值	最大值	单位
	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

器件更改

版本 1.0 是 I^2C LCD 器件的首个发布版本。

© 赛普拉斯半导体公司，2013。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品的内嵌电路之外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯产品不保证能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键器件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

PSoC® 是赛普拉斯半导体公司的注册商标，PSoC Creator™ 和 Programmable System-on-Chip™ 是赛普拉斯半导体公司的商标。此处引用的所有其他商标或注册商标归其各自所有者所有。

所有源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途之外，未经赛普拉斯的明确书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不仅限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不另行通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键器件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用的赛普拉斯软件许可协议限制。

[3] 此值在 2 字节命令模式（数据大小 = 1 字节）下计算得出

[4] 此值为打印 10 条所得的数值

[5] 此值为打印 4 条所得的数值