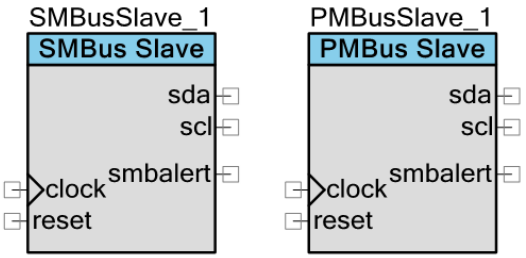
SMBus 和 PMBus 从器件

1.0



特性

- SMBus 从器件模式
- PMBus 从器件模式
- SMBALERT# 引脚支持
- 25 ms 超时
- 固定功能 (FF) 和 UDB 实现
- 可配置 SM/PM 总线命令

概述

系统管理总线 (SMBus) 和电源管理总线 (PMBus) 从器件组件提供了一种简单的方式来使用其上运行的 SMBus 或 PMBus 协议将 I^2C 物理层接口添加至 PSoC 3 或 PSoC 5 设计中。

SMBus 是一个双线接口，包括可与系统主机进行通信的各种系统管理芯片。它将 I^2C 用作物理层。SMBus 从器件组件实现了大多数的 SMBus 从器件规范，同时提供用于配置从器件参数的选项。此从器件可使用提供的 API 与 SMBus 主控进行通信。

PMBus 协议是更为通用的 SMBus 协议的特定实现。使用 PMBus，此器件显示所有可能的 PMBus 命令，并允许您选择哪些命令与您的应用程序相关联。

何时使用 SM 总线和 PM 总线从器件

此器件可用于需要 SMBus 或 PMBus 从器件通信接口的设计中。此组件处理硬件中的大量物理层要求。此固件处理协议和存储器缓冲区管理；它还管理 I^2C 中的数据传送。

输入/输出连接

本节介绍 SMBus 从器件的各种输入和输出连接。I/O 列表中的星号 (*) 表示该 I/O 是可隐藏 I/O，其隐藏条件在该 I/O 的说明中。

时钟 — 输入

时钟输入应用作为 I²C SCL/SDA 低电平停滞超时计时器的时钟源。当 Implementation（实现）参数设置为 UDB 时，它需要时钟提供 16 倍过采样。

当 I²C **Implementation（实现）**参数设置为 **UDB** 时，时钟输入可用。

Data rate（数据速率）	时钟
10 kbps	160 kHz
50 kbps	800 kHz
100 kbps	1.6 MHz
400 kbps	6.4 MHz
1000 kbps	16 MHz

复位 — 输入

UDB I²C 的硬件复位。如果高电平复位引脚保持在逻辑高电平状态，则 I²C 模块将处于复位状态，并且通过 I²C 进行的通信会停止。SDA 和 SCL 应强制为高电平。这仅适用于硬件复位。而软件必须使用 Stop() 和 Start() API 进行单独复位。

sda (SMBDAT) – 输入/输出

串行数据 (SDA) 是 I²C 数据信号。这种双向数据信号用于传输或接收所有总线数据。应该将连接至 sda 的引脚配置为开漏驱动低电平(Open-Drain-Drives-Low)。如果您在自定义程序中选择了 “External I/O Option”（外部 I/O 选项），单个 sda 双向信号将由单独的输入和输出（sda_o 和 sda_i）替换。在某些应用程序需要启用多个 I²C 总线的多路复用时，这是必要的。

scl (SMBCLK) – 输入/输出

串行时钟 (SCL) 是主控生成的 I²C 时钟。虽然从器件从不会生成时钟信号，但它可使时钟保持在低电平状态，并使总线停顿，直至它准备发送数据或确认/否认最新数据或地址为止。应该将连接至 scl 的引脚配置为开漏驱动低电平（Open-Drain-Drives-Low）。如果您在自定义程序中选择了 “External I/O Option”（外部 I/O 选项），单个 scl 双向信号将由单独的输入和输出（sdl_o 和 sdl_i）替换。在某些应用程序需要启用多个 I²C 总线的多路复用时，这是必要的。

smbalert (SMBALERT#) – 输出

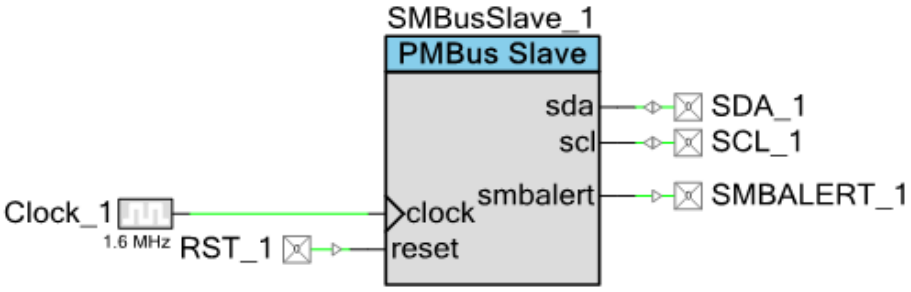
警报是可选 SMBus 定义的 SMBALERT# 引脚。此信号可选择性启用。警报引脚可通过后文介绍的 API 进行设置/解除。应该将连接至 smbalert 的引脚配置为开漏驱动低电平（ Open-Drain-Drives-Low）。

原理图宏信息

默认情况下，PSoC Creator 器件目录包括 SM/PM 总线从器件组件的原理图宏实现。这些宏包括已经连接和配置的引脚以及定义数据速率的内部配置的时钟值。原理图宏使用配置用于使用固定功能 I^2C 模块和硬件地址解址的 I^2C 从器件组件。

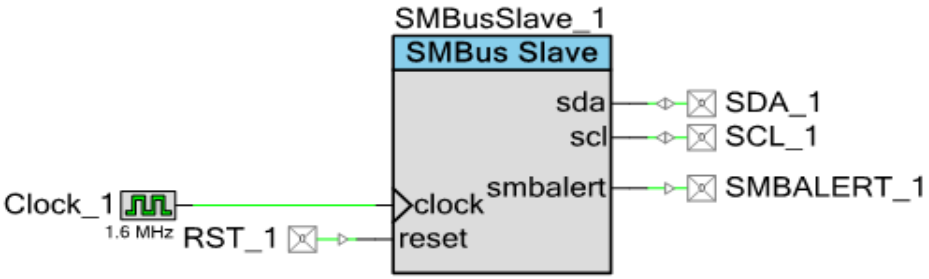
SM 总线从器件宏

此宏提供了 SM/PM 总线从器件组件在 SM 总线模式下的正确配置。连接至终端 SCL 和 SDA 的引脚被配置为双向，且 Drive（驱动）模式被设置为开漏驱动低电平（Open-Drain-Drives-Low）。此器件定义数据速率为 100 kHz。



PM 总线从器件宏

此宏提供了 SM/PM 总线从器件组件在 PM 总线模式下的正确配置。连接至终端 SCL 和 SDA 的引脚被配置为双向，且 Drive（驱动）模式被设置为开漏驱动低电平（Open-Drain-Drives-Low）。此器件定义数据速率为 400 kHz。



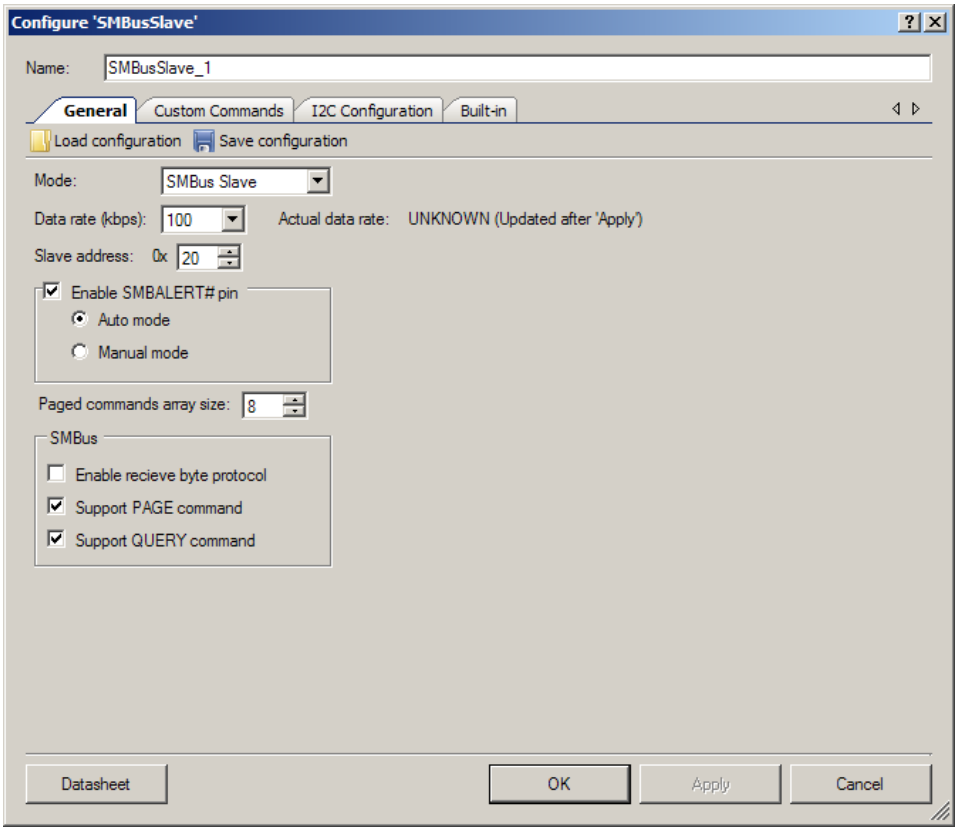
元件参数

将一个 SM/PM 总线从器件组件拖放到您的设计上，并双击以打开 **Configure（配置）**对话框。默认情况下，**Configure（配置）**对话框最初显示“General”（一般）选项卡，如图 1 所示。

一般选项卡

General（一般）选项卡提供用于配置 SM/PM 总线的一般设置的选项。有以下参数可用。

图 1. 一般 SM 总线和 PM 总线从器件选项卡



Export/Import Configuration（导出/导入配置）

Export/Import Configuration（导出/导入配置）允许您将自定义程序设置保存和恢复到外部文件中。它允许快速加载预设的配置文件和保留自定义设置。

模式

Mode（模式）参数将此器件的模式选择为 SMBus Slave（SMBus 从器件）模式或 PMBus Slave（PMBus 从器件）模式。如果选择了 SMBus Slave（SMBus 从器件）模式，则将禁用 PMBus Commands（PMBus 命令）选项卡。**Mode（模式）**还将确定可用的数据速率。在 PMBus Slave

（PMBus 从器件）模式中，只有两个选项：100 和 400 kHz。在 SMBus Slave（SMBus 从器件）模式中，还有额外的 10 和 50 kHz 选项。此参数的默认设置为 SMBus Slave（SMBus 从器件）模式。

Data Rate（数据速率）

Data Rate（数据速率）参数用于选择 I^2C 数据速率。可用选项取决于 SM/PMBus**Mode**（模式）选择。PMBus Slave（PMBus 从器件）模式允许 **Data Rate（数据速率）**值为 100 kHz 和 400 kHz。SMBus Slave（SMBus 从器件）模式提供 10 kHz、50 kHz、100 kHz 和 400 kHz 选项。**Data Rate（数据速率）**参数的默认值为 100 kHz。

Slave address（从器件地址）

Slave address（从器件地址）参数确定此器件的 I^2C 地址（7 位格式）。输入的值是十进制或十六进（如果最前面是“0x”）。自定义程序验证此地址以确保它不会与 SMBus 从器件保留列表上任何地址相冲突。**Slave address（从器件地址）**的默认值为默认值：0x20。

SMBALERT

SMBALERT 框允许您为主机通知配置可选 SMBALERT# 输出引脚。如果选择了 **Enable SMBALERT# pin（启用 SMBALERT# 引脚）**，此引脚将变为可见，作为器件符号上的输出。Auto/Manual（自动/手动）按钮决定在主机查询警报响应地址（ SetSmbAlertMode() API 的 GUI 表示）处的器件后， **SMBALERT# pin（SMBALERT# 引脚）**是否会自动解除。此参数的默认值设置为 EnabledAuto。

分页命令

此参数框配置 PMBus PAGE 命令。**Maximum page（最大页面）**参数确定分页命令的阵列大小。所有分页命令共享此阵列大小。默认值设置为 8 页。**有效范围为 1 到 32。**

SMBus 框

SMBus 框配置可选 SMBus 功能。它仅在 SMBus 从器件模式下显示。

- **Enable receives the byte protocol（启用接收字节协议）**复选框启用/禁用对 SMBus 接收字节协议的支持。如果未选中，任何接收字节数据操作都将被视为总线错误。如果已选中，此器件将调用 SMBus_GetReceiveByteResponse() API 确定接收字节请求的响应字节。
- **Support PAGE Command（支持 PAGE 命令）**复选框允许您在 SMBusSlave 模式下访问 PAGE 命令。
- **Support QUERY Command（支持 PAGE 命令）**复选框允许您在 SMBusSlave 模式下访问 QUERY 命令。

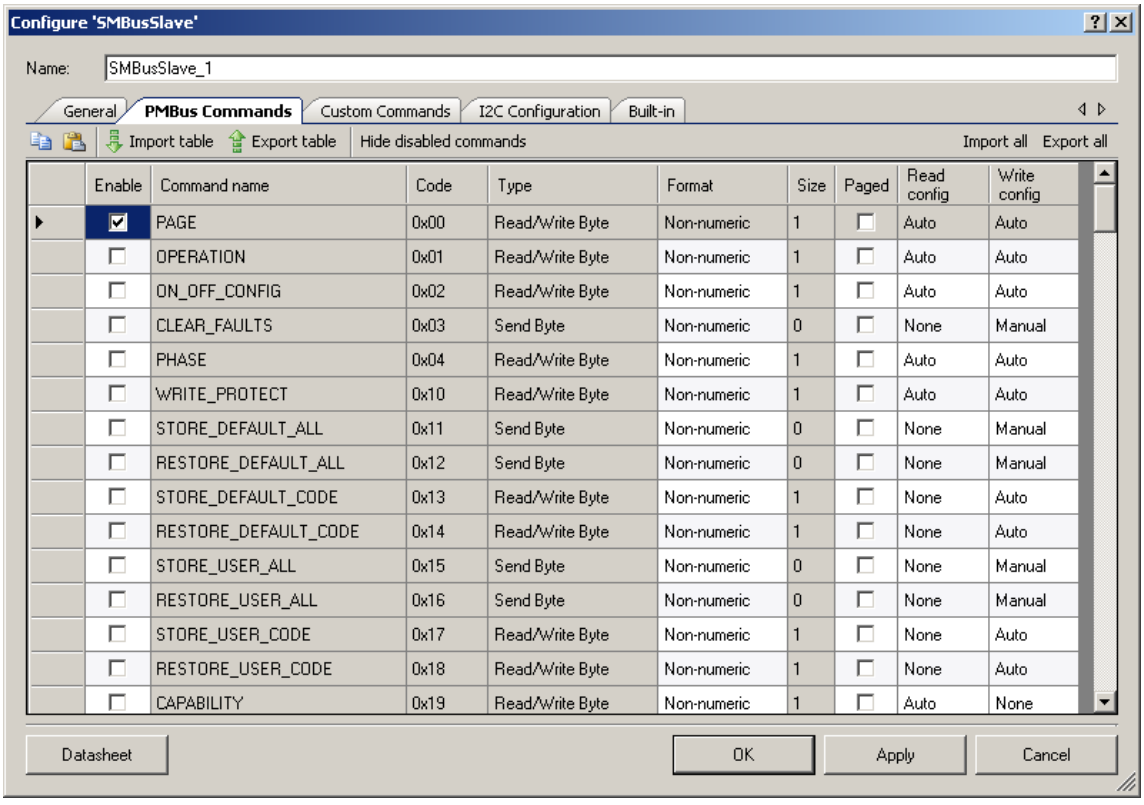
如果启用了 PAGE 或 QUERY 命令，则这些命令将被添加至 **Custom Commands（自定义命令）**选项卡上的命令列表中。这些命令的属性是基于 PMBus 规范的，但也可以对命令代码进行完全的自定义。

此框的默认值为：**Enable receives byte protocol（启动接收字节协议）**未选中，**Support PAGE Command（支持 PAGE 命令）**已启用，**Command Code（命令代码）**=0x00, **Support QUERY command（支持 QUERY 命令）**已启用，**Command Code（命令代码）**=0x1A。

PM Bus Commands（PM 总线命令）选项卡

PM Bus Commands（PM 总线命令）选项卡在此器件的 **Mode（模式）**设置为 PMBus Slave（PMBus 从器件）时可用。此选项卡显示 PMBus 规范中的定义命令的完整列表。预先填充的信息包括命令名称、其数字命令代码以及命令的类型（即 SMBus 协议）。您可以启用/禁用您希望此器件的实例处理的命令。Name（名称）、Code（代码）和 Type（类型）字段都是 Read-Only（只读）字段。此选项卡中的可用参数概述如下。

图 2. PM 总线命令 SM 总线和 PM 总线从器件选项卡



Enable	Command name	Code	Type	Format	Size	Paged	Read config	Write config
☑	PAGE	0x00	Read/Write Byte	Non-numeric	1	☐	Auto	Auto
☐	OPERATION	0x01	Read/Write Byte	Non-numeric	1	☐	Auto	Auto
☐	ON_OFF_CONFIG	0x02	Read/Write Byte	Non-numeric	1	☐	Auto	Auto
☐	CLEAR_FAULTS	0x03	Send Byte	Non-numeric	0	☐	None	Manual
☐	PHASE	0x04	Read/Write Byte	Non-numeric	1	☐	Auto	Auto
☐	WRITE_PROTECT	0x10	Read/Write Byte	Non-numeric	1	☐	Auto	Auto
☐	STORE_DEFAULT_ALL	0x11	Send Byte	Non-numeric	0	☐	None	Manual
☐	RESTORE_DEFAULT_ALL	0x12	Send Byte	Non-numeric	0	☐	None	Manual
☐	STORE_DEFAULT_CODE	0x13	Read/Write Byte	Non-numeric	1	☐	None	Auto
☐	RESTORE_DEFAULT_CODE	0x14	Read/Write Byte	Non-numeric	1	☐	None	Auto
☐	STORE_USER_ALL	0x15	Send Byte	Non-numeric	0	☐	None	Manual
☐	RESTORE_USER_ALL	0x16	Send Byte	Non-numeric	0	☐	None	Manual
☐	STORE_USER_CODE	0x17	Read/Write Byte	Non-numeric	1	☐	None	Auto
☐	RESTORE_USER_CODE	0x18	Read/Write Byte	Non-numeric	1	☐	None	Auto
☐	CAPABILITY	0x19	Read/Write Byte	Non-numeric	1	☐	Auto	None

Format（格式）

Format（格式）参数指定此命令的数字格式。此器件在指定对 QUERY 命令的响应时使用此格式。QUERY 命令中的可选格式值有 Linear（线性）、Signed（有符号）、Direct（直接号）、

Unsigned（无符号）和 VID Mode（VID 模式）。此字段仅用于 QUERY 命令目的，因为计算机并不执行任何实际的数字转换。

Size

对于模块和进程调用类型的命令，您可以编辑 **Size（大小）**字段以指定数据元素的大小。此大小不包括 SMBus 协议附加到模块传输的起始位置的大小/计数字节。仅可针对模块或进程调用命令编辑此字段。对于所有其他类型，**Size（大小）**字段来源于 PMBus 规范。默认值为 16。

Read/Write Config（读/写配置）

针对每个命令，您选择此命令是否通过 **Read Config（读配置）**和 **Write Config（写配置）**参数可读和/或可写。针对每个命令，您选择 None（无）、Auto（自动）或 Manual（手动）。None（无）说明此操作被禁用（即将 Read-Only（只读）命令的 Write Config（写配置）设置为 None（无））。“Auto”（自动）模式命令完全由此器件处理。它们在寄存器存储区与 I^2C 传输缓冲区之间传输，同时不会出现用户固件干扰或通知。“Manual”（手动）命令被添加至数据操作队列，必须由用户固件进行处理。这些参数的默认值为“Manual”（手动）。

注：由于写与读之间可能的不对称以及进程调用协议的复杂性质，不得针对使用此协议的命令选择自动模式。

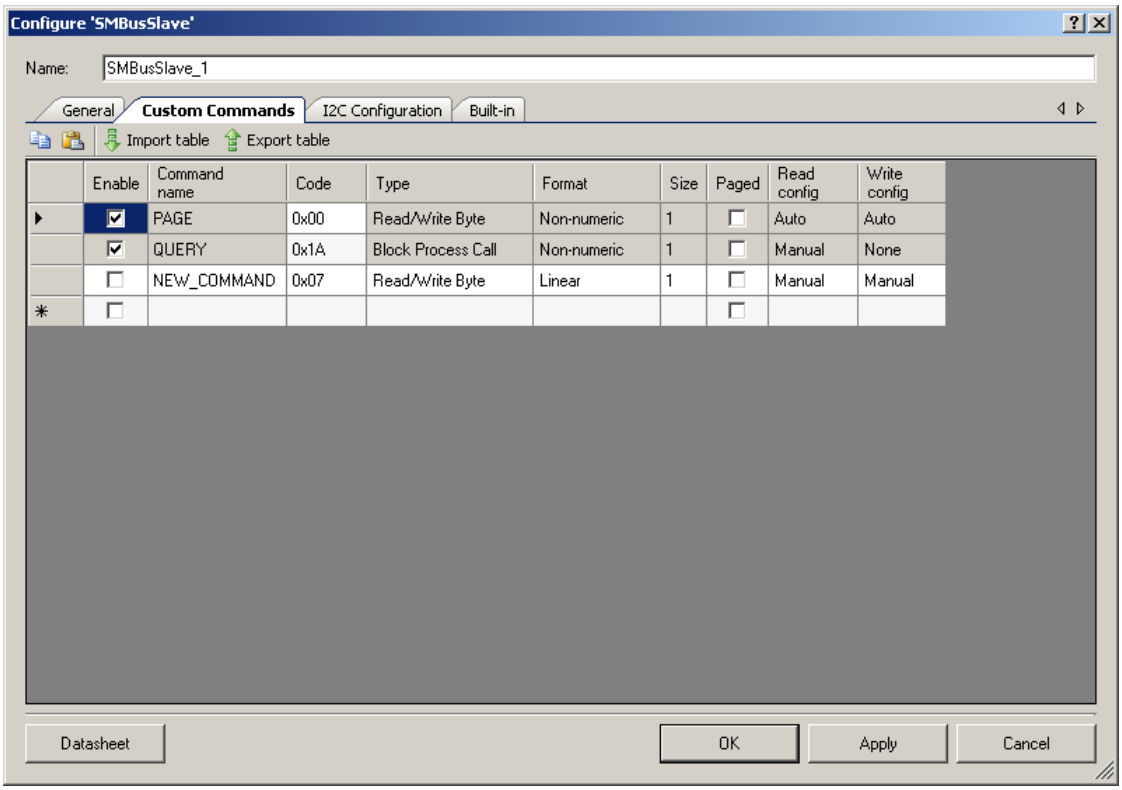
Paged（分页）

Paged（分页）复选框说明此命令是否分页（即索引）。针对被指定为分页的命令，此器件将自动在寄存器存储区中为此命令生成阵列。此阵列的大小取决于 **Maximum Page（最大页面参数）**。针对 Auto（自动）读写，此器件将基于当前的 PMBus 页面自动索引至分页参数的正确阵列编号（由上一个 PAGE 命令选定）。此参数的默认设置为未选中。

Custom Commands（自定义命令）选项卡

此选项卡允许您修改并自定义 Command Name（命令名称）、Code（代码）和 Type（类型）字段。

图 3. 自定义命令 SM 总线和 PM 总线从器件选项卡



Command Name（命令名称）

这是用户为此命令指定的名称。允许的字符为 A-Z（全部大写）、0-9 和下划线“_”。最大长度为 24 个字节。第一个字符不得为数字。不允许命令名称重复（包括重复标准的 PMBus 命令名称的自定义命令名称）。**Command Name（命令名称）**默认为空白。

Command Code（命令代码）

这是此命令的数字代码。它是十六进制值，限于两个字符（0-9、ssA-F）。不允许重复的命令代码。这包括与启用的 PMBus 命令相冲突的代码。**Command Code（命令代码）**默认为空白。

类型

Type（类型）指定针对此命令使用的 SMBus 传输协议/数据大小。可选值为 Send Byte（发送字节）、Read/Write Byte（读/写字节）、Read/Write Word（读/写字）、Read/Write Block（读/写

模块）、Process Call（进程调用）和 Block Process Call（模块进程调用）。它默认设置为 Read/Write Byte（读/写字节）。

Format（格式）

Format（格式）参数指定此命令的数字格式。此器件在指定对 QUERY 命令的响应时使用此格式。QUERY 命令中的可选格式值有 Linear（线性）、Signed（有符号）、Direct（直接号）、Unsigned（无符号）和 VID Mode（VID 模式）。此字段仅用于 QUERY 命令目的，因为计算机并不执行任何实际的数字转换。

Read/Write Config（读/写配置）

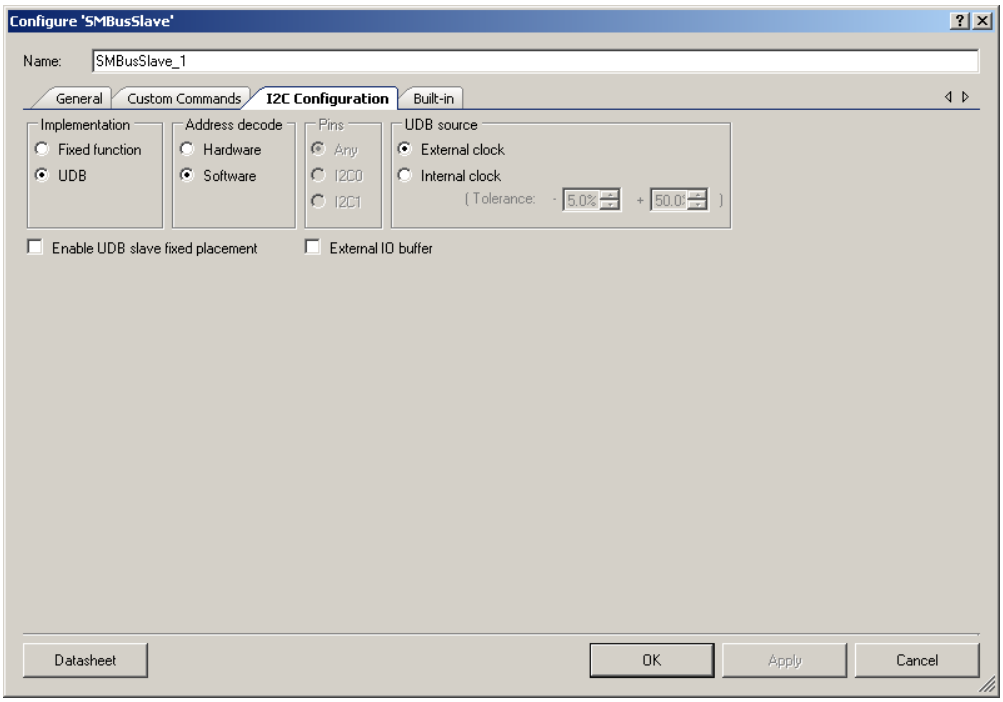
针对每个命令，您选择此命令是否通过 **Read Config（读配置）**和 **Write Config（写配置）**参数可读和/或可写。针对每个命令，您选择 None（无）、Auto（自动）或 Manual（手动）。None（无）说明此操作被禁用（即将 Read-Only（只读）命令的 Write Config（写配置）设置为 None（无））。“Auto”（自动）模式命令完全由此器件处理。它们在寄存器存储区与 I^2C 传输缓冲区之间传输，同时不会出现用户固件干扰或通知。“Manual”（手动）命令被添加至数据操作队列，必须由用户固件进行处理。这些参数的默认值为“Manual”（手动）。

注：由于写与读之间可能的不对称以及进程调用协议的复杂性质，不得针对使用此协议的命令选择自动模式。

I^2C Configuration（I2C 配置）选项卡

此选项卡允许配置 I^2C 硬件。

图 4. I²C 配置 SM 总线和 PM 总线从器件选项卡



实现

此参数确定 I²C 硬件是使用 **Fixed Function**[1]（**固定功能**）还是 **UDB** 实现。默认模式设置为 UDB。

Address decode（地址解码）

通过此参数，您可以选择软件地址解码或硬件地址解码。对于提供了足够的 API 并且只需一个从器件地址的大部分应用程序来说，将首选硬件地址解码。而对于希望修改源代码以检测多个从器件地址或 10 位地址的应用程序而言，则必须使用软件地址检测。Hardware（硬件）是此参数的默认设置。如果使能硬件地址解码，该模块会自动不响应不属于自己的地址，而无需 CPU 干预。它会在收到正确地址后自动中断 CPU 的运行，并将 SCL 线保持在低电平，直至 CPU 干预为止。

引脚

此参数可确定用于 SDA 和 SCL 信号连接的引脚类型。此参数包含三个可选值：Any（任意）、I2C0 和 I2C1。默认值为“Any”（任意）。“Any”（任意）表示通用 I/O（GPIO 或 SIO）。

[1]当在 PSoC5 上使用此器件时，不能选择 Fixed function（固定功能）。由于总线复位，PSoC5 I2C Fixed function（固定功能）无法从总线存储恢复。

UDB clock source（UDB 时钟源）

该参数允许您针对数据速率生成在内部配置的时钟和外部配置的时钟之间进行选择。考虑到 16 倍过采样，当设置为 Internal Clock（内部时钟）时，PSoC Creator 将基于 **Data Rate（数据速率）**参数计算和配置所需时钟频率。在 External Clock（外部时钟）模式下，器件不控制数据速率而是基于用户连接的时钟源显示实际速率。如果将此参数设置为 Internal Clock（内部时钟），则符号上将不显示时钟输入。您可以为内部时钟输入所需容差值。时钟容差可指定为百分比。默认范围为 -5% 到 +50%。

Enable UDB slave fixed placement（启用 UDB 从器件固定放置）

Enable UDB slave fixed placement（启用 UDB 从器件固定放置）参数允许您选择一个固定器件放置，以提供高于无限制放置下的器件性能。当设置此参数后，全部器件资源都将固定在器件的右上角。此参数可控制连接至器件的引脚分配。引脚分配的选择并非器件性能的决定因素。此选项仅在 **Implementation（实现）**设置为 UDB 时才有效。默认情况下，禁用此选项。此器件的固定放置状态消除了路由可变性。此外，还可以在一个完全空白的设计中按照与非固定放置设计相同的方式继续运行固定放置。

External IO Buffer（外部 IO 缓冲区）

此参数允许内部 I^2C 总线复用。内部 OE 缓冲区已被移除，双向 scl 和 sda 终端已被单独的输入（sda_i 和 scl_i）和输出（sda_o 和 scl_o）替换。应用程序编程接口 (API) 路由允许您使用软件配置此器件。下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 将实例名称“SMBusSlave_1”分配给提供的设计中的第一个器件实例。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。为增加可读性，下表中使用了实例名称“SMBusSlave”。

注意一些器件 API 功能被用于器件中断服务子程序中，因此，在使用 Keil 构建时，编译器可能会引起编译器警告提示。为了避免这种情况，js 将这些函数纳入到了“.cyre”文件中。

函数	说明
SMBusSlave_Start()	初始化并启用 SMBus 器件。启用 I^2C 中断后，该器件可响应 SMBus 通信。
SMBusSlave_Stop()	停止对 SMBus 通信的响应。同时禁用中断。
SMBusSlave_EnableInt()	启用 I^2C 中断。
SMBusSlave_DisableInt()	禁用 I^2C 中断。
SMBusSlave_Init()	使用自定义程序提供的初始值来初始化 I^2C 寄存器。
SMBusSlave_Enable()	激活 I^2C 硬件。不启用正常操作需要的 I^2C 中断。
SMBusSlave_SetAddress()	设置 I^2C 从器件地址。

函数	说明
SMBusSlave_SetAlertResponseAddress()	设置 I^2C 从器件地址，在此地址中当处于警报响应地址模式时，此器件将进行响应。
SMBusSlave_GetNextTransaction()	将一个指针返回至数据操作队列中的下一个数据操作记录。如果此队列为空，此函数将返回 NULL。
SMBusSlave_GetTransactionCount()	返回数据操作队列中的数据操作记录的数量。
SMBusSlave_CompleteTransaction()	促使此器件完成此队列最前面的当前待处理数据操作。
SMBusSlave_SetSmbAlert()	设置或解除 SMBALERT# smbalert 引脚。
SMBusSlave_SetSmbAlertMode()	确定此器件如何响应 SMBus 主控针对警报响应地址的读取。
SMBusSlave_HandleSmbAlertResponse()	当主机响应警报响应地址且 SMBALERT Mode（SMBALERT 模式）设置为 FIRMWARE_MODE 时，将被此器件调用。
SMBusSlave_GetReceiveByteResponse()	当 I^2C 中断服务子程序检测到“接收字节”协议请求时，将被其调用以确定响应字节。
SMBusSlave_HandleBusError()	当出现总线协议错误时，将被此器件调用。
SMBusSlave_StoreUserAll()	将 RAM 寄存器存储区保存至闪存中的用户寄存器存储区。
SMBusSlave_RestoreUserAll()	验证用户寄存器存储区的 CRC 字段，并将用户寄存器存储区的内容复制到 RAM 寄存器存储区中。
SMBusSlave_RestoreDefaultAll()	验证默认寄存器存储区的签名字段，并将默认寄存器存储区的内容复制至 RAM 寄存器存储区中。
SMBusSlave_StoreComponentAll()	调用此函数以使用当前 PMBus 设置更新系统中的其他器件的参数。
SMBusSlave_RestoreComponentAll()	调用此函数以使用系统中的其他器件的当前配置参数更新 PMBus 工作寄存器存储区。
SMBusSlave_Lin11ToFloat()	将参数“linear11”转换为浮点并将其返回。
SMBusSlave_FloatToLin11()	使用参数“floatvar”（浮点编号）并将其转换为 16 位 LINEAR11 值（11 位尾数 + 5 位指数），同时将其返回。
SMBusSlave_Lin16ToFloat()	将参数“linear16”转换为浮点并将其返回。
SMBusSlave_FloatToLin16()	使用参数“floatvar”（浮点编号）并将其转换为 16 位 LINEAR16 值（16 位尾数 ），同时将其返回。

全局变量

函数	说明
SMBusSlave_initVar (static)	initVar 变量用于说明此器件的初始配置。此变量前面加有器件名称，此处为 SMBusSlave。SMBusSlave_initVar 变量被初始化为 0，并在第一次调用 SMBusSlaveStart() 时被设置为 1。这可以实现器件初始化，同时无需重新初始化 SMBusSlave Start() 路由中的所有后续调用。 当此器件经历睡眠周期时，需要重新初始化此器件。因此，在执行睡眠 SMBusSlave Sleep() 时，此变量设置为 0；并在 SMBusSlave Wakeup() 中执行的重新初始化期间进行设置。

void SMBusSlave_Start(void)

说明： 这是开始执行器件操作的首选方法。SMBusSlave_Start() 调用 SMBusSlave_Init() 函数，然后调用 SMBusSlave_Enable() 函数。您必须在执行 I^2C 总线操作之前调用 SMBusSlave_Start()。此 API 可启用 I^2C 中断。

参数： None

返回值： None

副作用： None

void SMBusSlave_Stop(void)

说明： 此函数可禁用 I^2C 硬件和中断。如果 I^2C 总线被器件锁定且被设为空闲状态，则它会将其释放。

参数： None

返回值： None

副作用： None

void SMBusSlave_EnableInt(void)

说明： 此函数可使能 I^2C 中断。

参数： None

返回值： None

副作用： None

void SMBusSlave_DisableInt(void)

说明： 此函数可使能 I^2C 中断。通常情况下，在 I2C_Stop() 函数禁用中断后就不需要此函数了。

参数： None

返回值： None

副作用： 如果在运行 I^2C 的同时禁用 I^2C 中断，则可能造成 I^2C 总线锁定。

void SMBusSlave_Init(void)

说明： 此函数根据配置窗口 Configure（配置）对话框设置来初始化或恢复器件。您无需调用 SMBusSlave_Init()，因为 SMBusSlave_Start() API 会调用此函数，该函数是开始执行器件操作的首选方法。

参数： None

返回值： None

副作用： 所有寄存器将设置为自定义“配置”对话框中的值。

void SMBusSlave_Enable(void)

说明： 此函数激活硬件并开始执行器件操作。您无需调用 SMBusSlave_Enable()，因为 SMBusSlave_Start() API 会调用此函数，该函数是开始执行器件操作的首选方法。如果要调用此 API，则必须首先调用 SMBusSlave _Start() 或 SMBusSlave _Init()。

参数： None

返回值： None

副作用： None

void SMBusSlave_SetAddress(uint8 address)

说明： 此函数可设置 I^2C 从器件地址。

参数： uint8 address：主设备的 I^2C 从器件地址。此值可以为 0 至 127 (0x00 - 0x7F) 之间的任意地址。此地址为右对齐的 7 位从器件地址，它不包括读/写位。

返回值： None

副作用： None

voidSMBusSlave_SetAlertResponseAddress(uint8 address)

说明： 此函数设置 I^2C 从器件地址，在此地址中当处于警报响应地址模式时，此器件将进行响应。

参数： uint8 address：针对警报响应模式的 I^2C 从器件地址。此值可以为 0 至 127 (0x00 - 0x7F) 之间的任意地址。此地址为右对齐的 7 位从器件地址，它不包括读/写位。

返回值： None

副作用： None

TRANSACTION_STRUCT* SMBusSlave_GetNextTransaction(void)

说明： 此函数将一个指针返回至数据操作队列中的下一个数据操作记录中。如果此队列为空，此函数将返回 NULL。此函数将仅返回手动读取和写入，因为此器件将处理队列中的任何自动数据操作。如果是写入，服务于数据操作队列的用户固件负责将“有效负载”复制到寄存器存储区。如果是读取，用户固件负责更新寄存器存储区中此命令的变量的内容。如果是这两者，调用 SMBusSlave_CompleteTransaction() 以释放数据操作记录。

注意，对于读取数据操作，长度字段和有效负载字段不适于大多数的数据操作类型。例外情况是进程调用，其中写入相中的字将被存储在有效负载字段中。

参数： None

返回值： 指向下一个数据操作记录的指针

副作用： None

uint8 SMBusSlave_GetTransactionCount(void)

说明： 返回数据操作队列中的数据操作记录的数量。

参数： None

返回值： uint 8：数据操作队列中的记录数量

副作用： None

void SMBusSlave_CompleteTransaction(void)

说明： 促使此器件完成此队列最前面的当前待处理数据操作。用户固件数据操作处理程序在处理数据操作之后调用此函数。这将提醒器件代码将与寄存器存储区中的待处理读取数据操作相关联的寄存器变量复制到 I^2C 传输存储区中，以便于传输可以完成。它还可加快队列。必须针对读取和写入进行调用。

参数： None

返回值： None

副作用： None

void SMBusSlave_SetSmbAlert(uint8 assert)

说明： 设置或解除 SMBALERT# smbalert 引脚。只要设置了 SMBALERT#，此器件就将响应主控针对警报响应地址的读取。此响应将是此器件的主要从器件地址。根据模式设置，此器件将自动解除 SMBALERT#，调用 SMBusSlave_HandleSmbAlertResponse() API，或不执行任何操作。

参数： uint8: 激活

SPDIF 状态掩码	类型
SMBusSlave_SMBALERT_DEASSERT	解除 smbalert 引脚
SMBusSlave_SMBALERT_ASSERT	设置 smbalert 引脚

返回值： None

副作用： None

void SMBusSlave_SetSmbAlertMode(uint8 alertMode)

说明： 此函数确定此器件如何响应 SMBus 主控针对警报响应地址的读取。当设置了 SMBALERT# 时，SMBus 主控可将读取传播至全局警报响应地址以确定共享总线上的哪个 SMBus 设备已设置 SMBALERT#。

在自动模式下，一旦总线主控成功地 READ 警报响应地址，SMBALERT# 将自动解除。

在手动模式下，此器件将调用 API SMBusSlave_HandleSmbAlertResponse()，其中用户代码（在合并部分）负责解除 SMBALERT#。

在 DO_NOTHING 模式中，此器件将不执行任何操作。

参数： uint8: alertMode，定义 SMBALERT 引脚模式的字节

SPDIF 状态掩码	类型
SMBusSlave_DO_NOTHING	不对 SMBALERT# 引脚执行任何操作
SMBusSlave_AUTO_MODE	自动解除 SMBALERT# 引脚
SMBusSlave_FIRMWARE_MODE	用户负责解除 SMBALERT# 引脚

返回值： None

副作用： None

void SMBusSlave_HandleSmbAlertResponse(void)

说明： 当主机响应警报响应地址且 SMBALERT Mode（SMBALERT 模式）设置为 FIRMWARE_MODE 时，此 API 将被此器件调用。此函数包含合并代码部分，在此部分中当主控响应后，用户插入要运行的代码。例如，用户可能会更新状态寄存器并解除 SMBALERT# 引脚。

参数： None

返回值： None

副作用： None

uint8 SMBusSlave_GetReceiveByteResponse(void)

说明： 当 I^2C 中断服务子程序检测到“接收字节”协议请求时，此函数将被其调用以确定响应字节。此函数包括合并代码部分，在此部分中用户可插入其代码以覆盖此函数的返回值 - 即 0xFF。此函数将在中断服务子程序上下文中进行调用。因此，用户合并代码必须快速、无阻塞，且仅可调用重入函数。

参数： None

返回值： uint 8：用户指定的状态字节

SPDIF 状态掩码	类型
SMBusSlave_RET_UNDEFINED	默认返回状态

副作用： None

void SMBusSlave_HandleBusError(uint8 errorCode)

说明： 当出现总线协议错误时，此 API 将被此器件调用。总线错误的示例有：无效的命令、数据下溢和时钟延展冲突。此函数仅负责错误的后果，因为此器件将以确定的方式处理错误。此函数主要用于通知用户固件已出现错误。例如，在 PMBus 器件中，这会向用户固件提供在 STATUS_CML 寄存器中设置相关错误的机会。

参数： uint8 errorCode：

SPDIF 状态掩码	类型
SMBusSlave_ERR_READ_FLAG	读取标志设置错误
SMBusSlave_ERR_RD_TO_MANY_BYTES	主机尝试读取大量字节
SMBusSlave_ERR_WR_TO_MANY_BYTES	主机尝试写入大量字节
SMBusSlave_ERR_UNSUPORTED_CMD	不支持接收的命令
SMBusSlave_ERR_INVALID_DATA	收到的数据无效
SMBusSlave_ERR_TIMEOUT	出现了总线复位超时
SMBusSlave_ERR_WR_TO_FEW_BYTES	主机尝试写入少量字节

返回值： None

副作用： None

uint8 SMBusSlave_StoreUserAll(char * flashRegs)

说明： 此函数将 RAM 寄存器存储区保存至闪存中的用户寄存器存储区。寄存器存储区数据结构中的 CRC 字段将在保存之前进行重新计算和更新。默认情况下，此函数不会向闪存执行任何存储。相反，它包含合并区域，在此区域中用户可以执行将工作内存存储至闪存的算法。

参数： flashRegs：

指向闪存中应存储工作内存（RAM）的位置的指针。

返回值： uint 8：以下标准返回状态之一

SPDIF 状态掩码	类型
CYRET_SUCCESS	成功完成操作
CYRET_MEMORY	出现内存问题

副作用： None

uint8 SMBusSlave_RestoreUserAll(char * flashRegs)

说明： 此函数验证用户寄存器存储区的 CRC 字段，并将用户寄存器存储区的内容复制到 RAM 寄存器存储区中。默认情况下，此函数不会执行从闪存恢复寄存器存储区。相反，它包含合并区域，在此区域中用户可以执行将数据恢复至工作内存（RAM）的算法。

参数： flashRegs：

指向闪存中存储工作内存（RAM）的位置的指针 。

返回值： uint 8：以下标准返回状态之一。

SPDIF 状态掩码	类型
CYRET_SUCCESS	成功完成操作
CYRET_BAD_DATA	数据已损坏。CRC 不匹配

副作用： None

uint8 SMBusSlave_RestoreDefaultAll(void)

说明： 此函数验证默认寄存器存储区的签名字段，并将默认寄存器存储区的内容复制至 RAM 寄存器存储区中。

参数： None

返回值： uint 8：以下标准返回状态之一。

SPDIF 状态掩码	类型
CYRET_SUCCESS	成功完成操作
CYRET_BAD_DATA	数据已损坏。CRC 不匹配

副作用： None

uint8 SMBusSlave_StoreComponentAll(void)

说明： 用户调用此函数以使用当前 PMBus 设置更新系统中的其他器件的参数。因为此参数是非常特定于应用程序的，因此此函数几乎完全由合并部分组成。器件提供的唯一一个固件是返回值变量 (retval)，它将被初始化为 CYRET_SUCCESS 且在此函数结束时被恢复。此函数的其余部分必须由用户提供。

参数： None

返回值： uint 8：以下标准返回状态之一。

SPDIF 状态掩码	类型
CYRET_SUCCESS	成功完成操作

或用户定义的其他非成功状态。

副作用： None

uint8 SMBusSlave_RestoreComponentAll(void)

说明： 用户调用此函数以使用系统中的其他器件的当前配置参数更新 PMBus 工作寄存器存储区。因为此参数是非常特定于应用程序的，所以此函数几乎完全由合并部分组成。器件提供的唯一一个固件是返回值变量 (retval)，它将被初始化为 CYRET_SUCCESS 且在此函数结束时被恢复。此函数的其余部分必须由用户提供。

参数： None

返回值： uint 8：以下标准返回状态之一。

SPDIF 状态掩码	类型
CYRET_SUCCESS	成功完成操作

或用户定义的其他非成功状态。

副作用： None

float SMBusSlave_Lin11ToFloat (uint16 linear11)

说明： 此函数将参数“linear11”转换为浮点并将其返回。

参数： uint16 linear11：使用 LINEAR11 格式的数字。

返回值： float：被转换为浮点的 linear11 参数

副作用： None

uint16 SMBusSlave_FloatToLin11 (float floatvar)

说明： 此函数使用参数“floatvar”（浮点编号）并将其转换为 16 位 LINEAR11 值（11 位尾数 + 5 位指数），同时将其返回。

参数： float floatvar：浮点编号

返回值： uint16：被转换为 LINEAR11 的 floatvar

副作用： None

float SMBusSlave_Lin16ToFloat(uint16 linear16, int8 inExponent)

说明： 此函数将参数“linear16”转换为浮点并将其返回。参数 Linear16 包含尾数。参数 inExponent 是转换中要使用的 5 位二进制补码指数。

参数： uint16 linear16：LINEAR16 数字的 16 位尾数。

int8inExponent：LINEAR16 数字的 5 位指数。在低 5 位中。2 的补码。

返回值： float：被转换为浮点的参数

副作用： None

uint16 SMBus_FloatToLin16(float floatvar, int8 outExponent)

说明： 此函数使用参数“floatvar”（浮点编号）并将其转换为 16 位 LINEAR16 值（16 位尾数），同时将其返回。参数 outExponent 是转换中要使用的 5 位二进制补码指数。

参数： floatfloatvar：要转换为 LINEAR16 的浮点编号。

int8outExponent：转换中要使用的用户提供的 5 位指数。

返回值： uint16：被转换为 LINEAR16 的参数。

副作用： None

引导加载程序支持

SMBus 和 PMBus 从器件组件可用作引导加载程序的通信器件。有关引导加载程序的更多信息，请参考 *System Reference Guide*（《系统参考指南》）中的“Bootloader System”（引导加载程序系统）一节。

SMBus 和 PMBus 从器件组件为使用引导加载程序提供了一组 API 函数。

函数	说明
SMBusSlave_CyBtldrCommStart	启动 SMBus 和 PMBus 从器件组件，并使其中断。
SMBusSlave_CyBtldrCommStop	禁用 SMBus 和 PMBus 从器件组件并禁用其中断。
SMBusSlave_CyBtldrCommReset	将读取和写入 I^2C 缓冲区设置为初始状态，并使从器件状态复位。
SMBusSlave_CyBtldrCommWrite	允许调用程序将数据写入引导加载程序主机中。该函数将处理轮询以便让数据块完全发送至主机器件。
SMBusSlave_CyBtldrCommRead	允许调用程序读取引导加载程序主机中的数据。该函数将处理轮询以便从主机器件完全接收到数据块。

void SMBusSlave_CyBtldrCommStart(void)

说明： 启动通信器件并启用中断。读取缓冲区初始状态为已满，读取始终为 0xFFu。写入缓冲区已清理，可接收命令。

参数： None

返回值： None

副作用： 此函数启用器件中断。如果已启用 I^2C 而为启用中断，它将锁定总线。

void SMBusSlave_CyBtldrCommStop(void)

说明： 禁用通信器件并禁用中断。

参数： None

返回值： None

副作用： None

void SMBusSlave_CyBtldrCommReset(void)

说明： 将缓冲区设置为初始状态，并使从状态复位。读取缓冲区初始状态为已满，读取始终为 0xFFu。写入缓冲区已清理，可接收命令。

参数： None

返回值： None

副作用： None

cystatus SMBusSlave_CyBtldrCommRead(uint8 * Data, uint16 size, uint16 * count, uint8 timeOut)

说明： 接收来自主机的命令。所有字节由 I^2C 中断服务子程序接收并存储在内部 I^2C 缓冲区中。此函数使用超时检查状态以确定传输结束，然后将数据复制到引导加载程序缓冲区。退出此函数后，I^2C 中断服务子程序能够接收更多的数据。

参数： uint8 *Data: 指向要从引导加载程序主机读取的数据模块的存储的指针

uint16 size: 要读取的字节数

uint16 *count: 指向用于写实际读取字节数的变量的指针

uint8 timeOut: 等待的单位数（时间为 10 毫秒），之后会因超时而返回

返回值： cystatus：如果未遇到任何问题将返回 CYRET_SUCCESS，或返回可详尽描述该问题的值。有关更多信息，请参考 *System Reference Guide*（《系统参考指南》）中的“Return Codes”（返回代码）一节。

副作用： None

cystatus SMBusSlave_CyBtldrCommWrite(uint8 * Data, uint16 size, uint16 * count, uint8 timeOut)

说明： 将已执行命令的状态传送至主机。此函数使用响应更新 I^2C 读取缓冲区并将其释放给主机。所有读取都返回 0xFF，直至缓冲区被释放。所有字节由 I^2C 中断服务子程序传送。此函数超时等待，直至已读取所有字节。退出此参数后，所有读取返回 0xFF。

参数： uint8 *Data: 指向要写入引导加载程序主机的数据模块的指针

uint16 size: 要写入的字节数

uint16 *count: 用于写实际写入字节数的变量指针

uint8 timeOut: 等待的单位数（时间为 10 毫秒），之后会因超时而返回

返回值： cystatus：如果未遇到任何问题将返回 CYRET_SUCCESS，或返回可详尽描述该问题的值。有关更多信息，请参考 *System Reference Guide*（《系统参考指南》）中的“Return Codes”（返回代码）一节。

副作用： 在“Bootloader Write”（引导加载程序写入）数据操作期间调用启动时钟延展时，临时启用器件中断，但在返回之前通过此函数禁用此中断（以符合手动命令处理行为）。当不是使用“Bootloader Write”（引导加载程序写入）进行调用时 - 始终保留器件中断已启用。

宏

- SMBusSlave_FL_ADDR_TO_ROW(addr) - 从指定地址提取闪存行数。
- SMBusSlave_FL_ADDR_TO_ARRAYID(addr) - 从指定地址提取闪存阵列 ID。

- SMBusSlave_SIZE_TO_ROW(size) - 计算并返回存储 ***size（大小）***定义的数据量所需的闪存行数。
- SMBusSlave_MAX_PAGES - 指定分页命令所使用的最大页数。
- SMBusSlave_NUM_COMMANDS - 定义分页命令的页数。

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。

有关更多信息，请参考 PSoC Creator 帮助中的“查找示例项目”主题。

中断服务子程序

SMBus 和 PMBus 从器件组件使用 2 个中断服务子程序进行操作。一个中断服务子程序处理通过 SM/PM 总线的命令解码和数据操作。另一中断服务子程序是 25 ms 超时中断，设计用于在出现低电平停滞情况时复位总线。

功能描述

此器件的工作原理与 I^2C 从器件组件的工作原理非常类似。所有 SMBus 规范参考 SMBus 规范版本 2.0。本节将重点介绍此器件的关键功能。

I^2C 物理层

SM/PM 总线从器件组件的物理层是基于 I^2C 协议的。影响此器件的三个主要区别是：

SMBus 规范强制规定，如果检测到 SCL 信号低电平停滞 25 ms，此器件必须复位并释放 SCL 和 SDA 线。这在“AC & DC Electrical Performance Requirements”（交流和直流电气性能要求）一节中有更详细的说明。此器件还监控 SDA 线，并且如果其仍保持低电平停滞状态 25 ms, 作为额外防范措施，还将对其进行复位。

SMBus 规范强制规定此器件不得在任何给定传输中延展时钟超过 25 ms（累积时间）。如果任何数据操作中的累积延展时间未超过 25 ms，当此从器件忙碌于将 SCL 拉至低电平（时钟延展）时，允许此从器件延迟传输

增加 SMBALERT# 引脚以通知主机此器件需要引起关注

SMBus/PMBus 寻址

每个 SMBus/PMBus 从器件都有一个 I^2C 地址。以下地址留作特定的 SMBus 用途，且不得用作为 SMBus/PMBus 从器件的通用从器件地址。

从器件地址（位 7:1）	R/W# 位（位 0）	注释
0000 000	0	通用调用地址
0000 000	1	START 字节
0000 001	X	CBUS 地址
0000 010	X	留作用于其他总线格式
0000 011	X	留作日后使用
0000 1XX	X	留作日后使用
0101 000	X	留作用于 ACCESS 总线主机
0110 111	X	留作用于 ACCESS 总线默认地址
1111 0XX	X	留作用于 10 位从器件寻址
1111 1XX	X	留作日后使用
0001 000	X	留作用于 SMBus 主机
0001 100	X	SMBus 警报响应地址
1100 001	X	SMBus 器件默认地址

如果用户启用了 SMBALERT# 引脚选项，此器件将响应其主要地址以及警报响应地址。

SMBus/PMBus 协议

SMBus 规范中定义了 9 个不同的协议。这些协议的概要及其支持状态如下所示。

协议	支持状态	注释
快速命令	不受支持	仅 SMBus
发送字节	受支持	两者
接收字节	受支持	仅 SMBus
写入字节/字	受支持	两者
读取字节/字	受支持	两者
进程调用	受支持	两者
模块写入/读取	受支持	两者

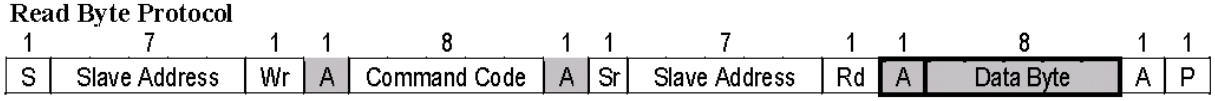
协议	支持状态	注释
模块写入/模块读取进程调用	受支持	两者
SMBus 主机通知协议	不受支持	两者

体现 SM/PM 总线器件的特性的关键概念如下。

SMBus 和 PMBus 都不使用子地址或 I^2C 寄存器映射的概念。所有传输都是基于命令的。I^2C 从器件地址后面紧跟的是命令代码。命令可以是只写、只读或读/写命令。在 SMBus 中，命令的定义以及读/写限制几乎完全取决于用户。在 PMBus 中，命令很大程度上是预先定义的，但提供了一系列的命令代码，这些命令代码未进行分配，可用于“特定于制造商的”实现。

对于读取数据操作，主机通过发出 REPEATED START 总线条件在编写读取命令代码与读取请求数据之间来回切换。一旦整个数据操作完成，将仅生成 STOP 位。如下例所示：

Read Byte Protocol



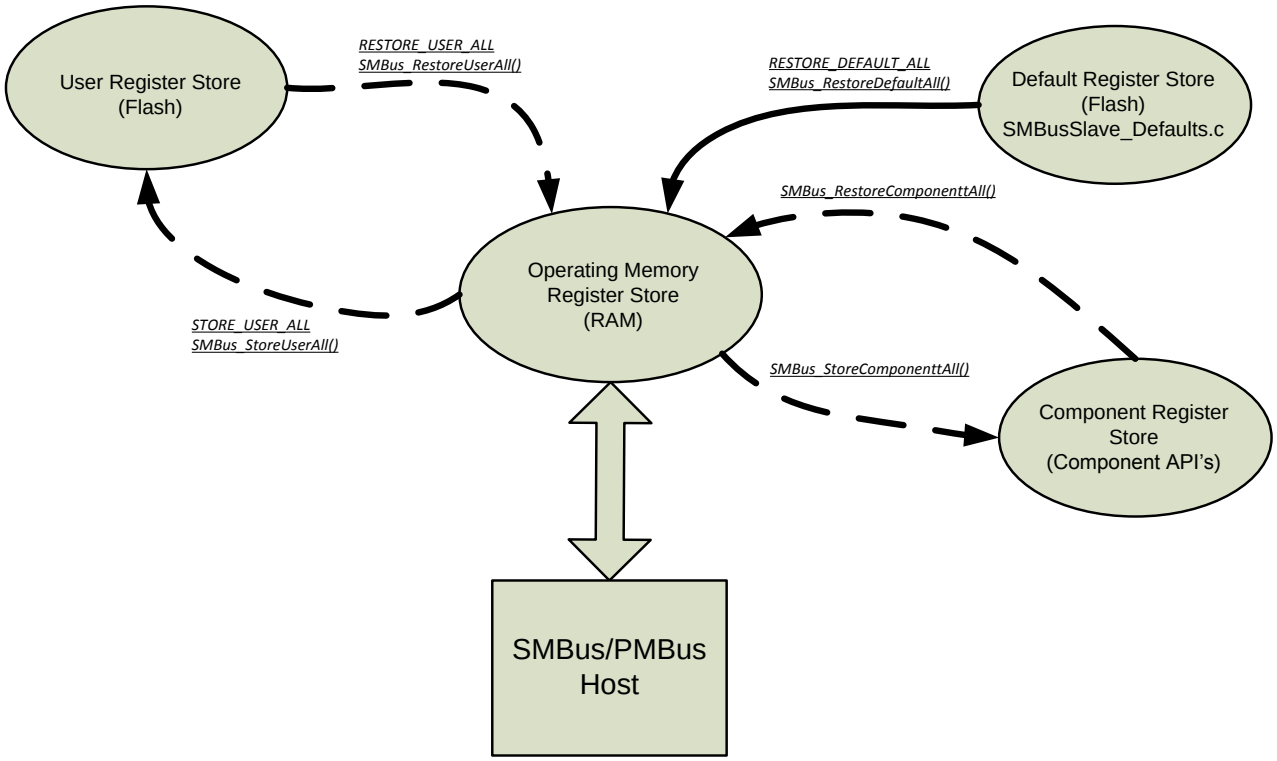
SMBus 和 PMBus 是低位优先协议（即在总线上将首先传送多字节数据类型的最低有效字节）。从 PSoC 角度需要所有寄存器以 PSoC 固有的字节顺序格式存储。此器件负责在 PSoC 与 I^2C 缓冲区之间正确地转换字节顺序。这仅适用于 16 位字传输。

SMBus 规范不定义任何命令代码；它们由用户进行定义。但是，PMBus 规范第 2 部分附录 1 定义了所有 255 个可选命令代码；其中 46 个由用户进行定义（被称为特定于制造商的命令）。

寄存器存储区概念

根据 PMBus 规范 II 5.4.2 – “Every Parameter That Can Be Written Must Be Readable”（可写入的每个参数必须可读）。通常，接受写入的值的任何命令在读取时也必须返回此值。为此，使用了寄存器存储区的概念。

图 5. 寄存器存储区概念



工作内存寄存器存储区

这是 RAM 版本的寄存器存储区。运行时 SMBus/PMBus 命令可修改此版本的寄存器存储区。由于此寄存器存储区位于 RAM 中，因此在复位时其内容被假定为无效。

默认寄存器存储区

闪存版本的寄存器存储区，包含所有 SMBus/PMBus 参数的默认值。启动时，默认寄存器存储区可被复制至工作内存寄存器存储区以初始化工作内寄存器存储区。默认寄存器存储区中的参数值在编译/链接期间是固定的。用户负责为寄存器存储区中的参数提供默认值。要打开 SMBusSlave_Defaults.c，从命令模块中复制参数并将其粘贴至下面的合并区域。如果您这样做，它们将存储在 SMBusSlave_regsDefault 中，且每次启动时，工作内存将使用这些参数进行初始

化。另一种使用默认值初始化工作内存的方式是调用 SMBus_RestoreDefaultAll()。此函数可在用户处理程序用于 RESTORE_DEFAULT_ALL (0x12) PM 总线命令。

用户寄存器存储区

这是另一个闪存版本的寄存器存储区。不同于与默认寄存器存储区，它基本上是只读的，而用户寄存器存储区则可以基于工作寄存器存储区的当前内容进行更新。由于将数据存储至闪存中的操作是特定于实现的，因此此器件不会执行任何存储至闪存中的操作；它提供了两个包括合并区域的 API 功能，此区域可用于此目的。SMBusSlave_StoreUserAll() 不会执行检查恢复时的数据有效性所需的 CRC 的计算。

要设计您自己的将数据存储至闪存的数据的方法，请参考随附 PSoC Creator 提供的 System Reference Guide（《系统参考指南》）。System Reference Guide（《系统参考指南》）的第 10 节提供了基本信息以及用于处理闪存的命令的说明。此器件中还提供了一组宏，在处理闪存时可使用这些宏。

器件寄存器存储区

器件寄存器存储的主要动机是允许 PMBus 从标准的 PSoC Creator 器件中提取参数并配置这些器件（从而，名称为器件寄存器存储区）。Creator 器件都有其自己的配置参数，这些参数通常是使用器件自定义程序设置的。在运行时可通过特定于器件的 SMBusSlave_StoreComponentAll()/SMBusSlave_RestoreComponentAll() API 访问这些参数。可通过 API 访问的器件参数包括器件寄存器存储区。启动时，用户 PMBus 固件可能需要：

- 基于用户或默认寄存器存储区中存储的 PMBus 参数更新其他器件设置，或
- 基于器件设置更新 PMBus 寄存器存储区。

特例命令

PAGE 命令

PMBus PAGE（代码 0x00）命令（PMBus 第 2 部分 – 第 11.10 节）允许访问同一 PMBus I^2C 地址下的多个逻辑 PMBus 器件。例如，控制多个电源轨的 PMBus 功耗监控器可提供对每个轨在其自己的页面上的命令/参数的访问。此页面可被视为到命令/寄存器的阵列的索引。一旦通过 PAGE 命令设置了此页面，页面设置将是持久性的，直至再次被另一 PAGE 命名设置。

在 PMBus 模式中，此器件具有对 PAGE 命令的内置支持。在 SMBus 模式中，用户可以选择启用 PAGE 命令并为 PAGE 指定要使用的命令代码。

如果启用了 PAGE 命令，PAGE 命令值的有效范围在 1 到 32 之间，且必须在用户确定的 **Max Page（最大页面）**设置范围内。例外情况是“All Pages”（所有页面）通配符设置，为 0xFF。“All

Pages”（所有页面）通配符仅针对写入数据操作有效，且必须在“Manual”（手动）模式中进行处理。如果 PAGE 设置为 0xFF，以下数据操作将被视为错误。

通过分页命令尝试读取

通过配置为 Manual（手动）的分页命令尝试写入

甚至在包括多个页面的 PMBus 器件中，一些命令会独立于当前页面，且始终按相同的方式进行处理，与 PAGE 设置无关。例如，用于返回此器件支持的 PMBus 版本的 PMBUS_REVISION 命令对于所有页面是通用的。因此，存在页面命令和常见命令的概念。对于每个 SMBus/PMBus 命令，自定义程序允许用户指定命令是 Paged（分页）命令还是 Common（常见）命令。当命令被标记为 Paged（分页）时，此器件将在寄存器存储区中为此命令定义一个阵列。

QUERY 命令

PMBus QUERY（代码 0x1A）命令用于询问 PMBus 器件它是否支持给出的命令，以及如果支持的话它针对此命令支持哪些数据格式。在 PMBus 模式中，此器件具有对 QUERY 命令的内置支持。在 SMBus 模式中，用户可以选择启用 QUERY 命令并为 QUERY 指定要使用的命令代码。

此命令使用 SMBus 规范中介绍的模块写入模块读取进程调用。针对进程调用的写入部分，一个数据字节是未分配的二进制整数，此整数等于正在被操作的命令的命令代码。针对进程调度的读取部分，一个数据字节是未分配的二进制整数，其包括以下值：

位	值	含义
7	1	支持命令
	0	不支持命令
6	1	支持命令进行写入
	0	不支持命令进行写入
5	1	支持命令进行读取
	0	不支持命令进行读取
4:2	000	使用的线性数据格式
	001	16 位有符号数字
	010	保留
	011	使用的直接模式格式
	100	8 位无符号数字
	101	使用的 VID 模式格式
	110	使用的特定于制造商的格式
	111	命令不返回数字数据。这还用于返回各个数据区块的命令。

位	值	含义
1:0	XX	留作日后使用

可以基于自定义程序中的用户选择生成上表中的所有信息。如果不支持使用 QUERY 研究的命令代码，在这种情况下，则将返回“0x00”。

引导加载程序命令

当此器件被安置在“引导加载程序”项目中，Configure（配置）对话中的 **Custom Commands（自定义命令）**选项卡上将有两个额外的命令可用。这两个命令是 BOOTLOAD_READ 和 BOOTLOAD_WRITE，它们的默认命令代码分别为 0xFD 和 0xFC。它们向此器件增加了一项用作为引导加载程序器件的通信器件的功能。

这两个命令可与设计原理图上安置的引导加载程序器件进行交互。在安置引导加载程序后，在引导加载程序器件的 Configure（配置）对话框中选择“Custom interface”（自定义界面）。

资源

下面列出了固定功能实现的 SM/PM 总线从器件组件的资源图表。数据是数据速率设置为 400 kbps 时收集的。

PSoC 3

配置	资源类型					
	数据路径单元	宏单元	状态单元	控制单元	I^2C 固定模块	中断
SM 总线/PM 总线 (UDB)	3	32	2	6	-	2
SM 总线/PM 总线（固定功能）	2	9	1	4	1	2

PSoC 5

配置	资源类型					
	数据路径单元	宏单元	状态单元	控制单元	I^2C 固定模块	中断
SM 总线/PM 总线 (UDB)	3	32	2	6	-	2
SM 总线/PM 总线（固定功能）	不可用	不可用	不可用	不可用	不可用	不可用

PSoC 5LP

配置	资源类型					
	数据路径单元	宏单元	状态单元	控制单元	I^2C 固定模块	中断
SM 总线/PM 总线 (UDB)	3	32	2	6	–	2
SM 总线/PM 总线（固定功能）	1	5	2	2	1	2

API 内存使用情况

器件使用情况显著不同，取决于编译器、设备、使用 API 的数量以及器件配置。下表提供了给定的器件配置中的所有 API 的内存使用情况。

已使用 Release（发布）模式中配置的关联编译器进行了测量，此编译器使用了 Size（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 5 (GCC)		PSoC 5LP (GCC)	
	Flash（闪存）字节	SRAM 字节	Flash（闪存）字节	SRAM 字节	Flash（闪存）字节	SRAM 字节
SM 总线从器件（固定功能）	5501	200	不可用	不可用	3540	168
SM 总线从器件 (UDB)	5587	196	3832	156	3824	156
PM 总线从器件（固定功能）	8166	1788	不可用	不可用	4632	4068
PM 总线从器件 (UDB)	8080	1792	4924	4056	4916	4056

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 且 T_J ≤ 100 °C。除非另有说明，否则这些规范的适用范围为 1.71 V 到 5.5 V。

SMBus 和 PMBus 从器件 DC 规范

参数	说明		最小值	典型值[2]	最大值	单位
I_{DD}	器件电流消耗					
	SMBus (UDB)	数据速率 = 50kbps	–	380	–	µA
	SMBus（固定功能）		–	670	–	µA
	PMBus (UDB)	数据速率 = 100kbps	–	440	–	µA
	PMBus（固定功能）		–	620	–	µA
	SMBus (UDB)	数据速率 = 400kbps	–	720	–	µA
	SMBus（固定功能）		–	860	–	µA
	PMBus (UDB)	数据速率 = 400kbps	–	750	–	µA
	PMBus（固定功能）		–	980	–	µA

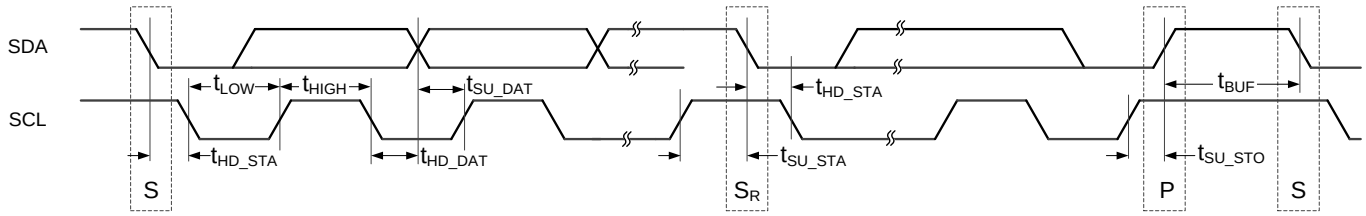
SMBus 和 PMBus 从器件 AC 规范

参数	说明	最小值	典型值	最大值	单位
f_{SCL}	SCL 时钟频率	– – –	– – –	100 400 1000	kHz
f_{CLOCK}	器件输入时钟频率	–	16 × f_{SCL}	–	kHz
t_{RESET}	复位脉冲宽度	–	2	–	t_{CY_clock}[3]
t_{LOW}	SCL 时钟的低电平周期	4.7 1.3 0.5	– – –	– – –	µs
t_{HIGH}	SCL 时钟的高电平周期	4.0 0.6 0.26	– – –	– – –	µs
t_{HD_STA}	保留时间（重复）启动条件	4.0 0.6 0.26	– – –	– – –	µs
t_{SU_STA}	重复 START 条件的建立时间	4.7 0.6 0.26	– – –	– – –	µs

[2] 未包括设备 IO 和时钟分配的电流。这些值是在 25 °C 时的值。数据是在 BUS_CLK 设置为 24 MHz 时测得的。
[3] $t_{CY_clock} = 1/f_{CLOCK}$。这是一个时钟周期的时间长度

参数	说明	最小值	典型值	最大值	单位
t_{HD_DAT}	数据保留时间	5.0 – –	– – –	– – –	µs
t_{SU_DAT}	数据建立时间	250 100 50	– – –	– – –	ns
t_{SU_STO}	STOP 条件的建立时间	4.0 0.6 0.26	– – –	– – –	µs
t_{BUF}	STOP 和 START 条件之间的总线空闲时间	4.7 1.3 0.5	– – –	– – –	µs
t_{FLASH_WRITE}		-	40.6[4]	-	ms

图 6. 数据转换时序图



器件更改

本节介绍器件与以前版本相比的主要更改。

版本	更改说明	更改/影响原因
1.0	首次发行版	

[4] 通过以下方式获取值：使用所有 PM 总线命令存储为 PM 总线模式配置的寄存器存储区，同时这些命令是使用 BUS_CLK 为 24 MHz 时的默认配置启用的。

© 赛普拉斯半导体公司，2012。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。

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。赛普拉斯保留在不做出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。