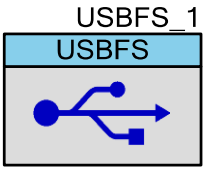
全速 USB (USBFS)

1.60

特点



- USB 全速器件接口驱动程序
- 支持中断传输、控制传输、批量传输和同步传输类型
- 对描述符集选择的运行时支持
- 可选 USB 字符串描述符
- 可选 USB HID 类支持
- 可选引导加载程序支持

一般说明

USBFS 组件提供与第 9 章兼容的 USB 全速器件框架。该组件为对来自 USB 主机的请求进行解码和调度的控制端点提供了低级别驱动程序。另外，此组件提供 USBFS 自定义程序以实现轻松描述符构建。

您可以选择构建基于 HID 的器件或泛型 USB 器件。通过设置配置/接口描述符，选择 HID（和 HID 与泛型之间的切换）。

有关描述符的其他信息，请参考 USB-IF 器件类文档 (http://www.usb.org/developers/devclass/)。

注意：赛普拉斯免费提供一组与赛普拉斯芯片一同使用的 USB 开发工具（称为 SuiteUSB）。可以从赛普拉斯网站获取 SuiteUSB：http://www.cypress.com。

何时使用 USBFS

当您要提供具有兼容 USB 2.0 器件接口的应用程序时，使用 USBFS 组件。

快速启动

1. 将 USBFS 组件从组件目录拖动到您的设计中。
2. 注意“通知列表”窗口中的时钟错误；双击错误可打开系统时钟编辑器。

3. 配置下列时钟：
    - **ILO**：选择 100 kHz。
    - **IMO**: 选择 Osc 24.000 MHz。
    - **USB**：使能并选择 IMOx2 – 48.000 MHz。

    注意：如果选择的器件为 PSoC 3 ES2 或 PSoC 5，还必须将 PLL 配置为“Desired 33 MHz”（要求的 33 MHz），将主控时钟配置为“PLL_OUT (33.000 MHz)”。
4. 选择**构建**以生成 API；有关演示 USBFS 组件的基本功能的示例以及基本设置说明，请参考“固件源代码示例”一节。

输入/输出连接

本节介绍 USBFS 的各种输入和输出连接。I/O 列表中的星号 (*) 表示 I/O 可能隐藏在该 I/O 说明中列出的条件下的符号中。

sof – 输出 *

帧起始 (sof) 输出允许端点识别帧起点并将内部端点时钟与主机同步。当高级配置选项卡中的“out_sof”参数设置为使能时，此输出可见。

组件参数

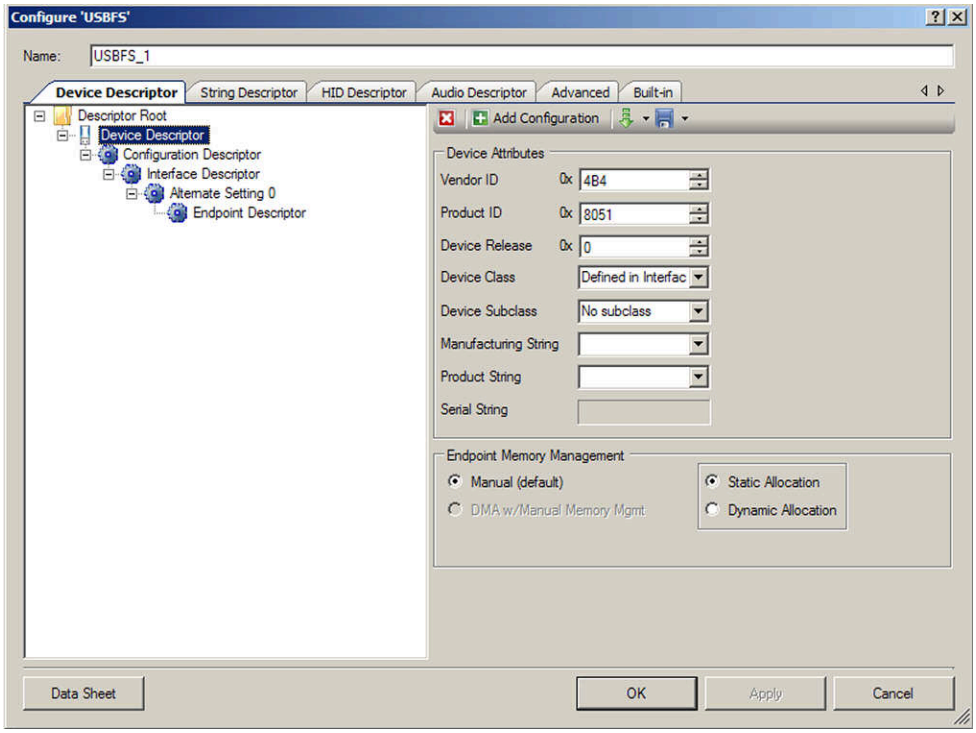
将 USBFS 组件拖动到您的设计中，双击它以打开“配置 USBFS”对话框。

组件由“配置 USBFS”对话框中生成的信息驱动。 此对话框或自定义程序简化了 USB 描述符的构建，并将生成的信息集成到用于器件仿真的驱动程序固件中。

如果不首先运行向导并选择相应的属性来描述您的器件，USBFS 组件将无法正常工作。代码发生器采用您的器件信息，并生成所有必需的 USB 描述符。

“配置 USBFS”对话框包含下列选项卡和设置：

“器件描述符”选项卡



器件属性

- 供货商 ID – 公司 USB 供货商 ID（从 USB-IF 获取）

  注意：供货商 ID 0x4B4 是仅供赛普拉斯使用的 VID，只能由于开发目的。不能使用此 VID 发布产品；您必须获取自己的 VID。
- 产品 ID – 您的特定产品 ID
- 器件版本 – 您的特定器件版本（器件 ID）
- 器件类 – 器件类是在接口描述符中定义的，或者它特定于供货商
- 器件子类 – 依赖于器件类
- 制造字符串 – 当连接器件时显示的特定于制造商的说明字符串。
- 产品字符串 – 当连接器件时显示的特定于产品的说明字符串。
- 序列字符串

端点存储器管理

某些应用程序可以利用直接存储器访问 (DMA) 将数据移进和移出端点存储器缓冲区。

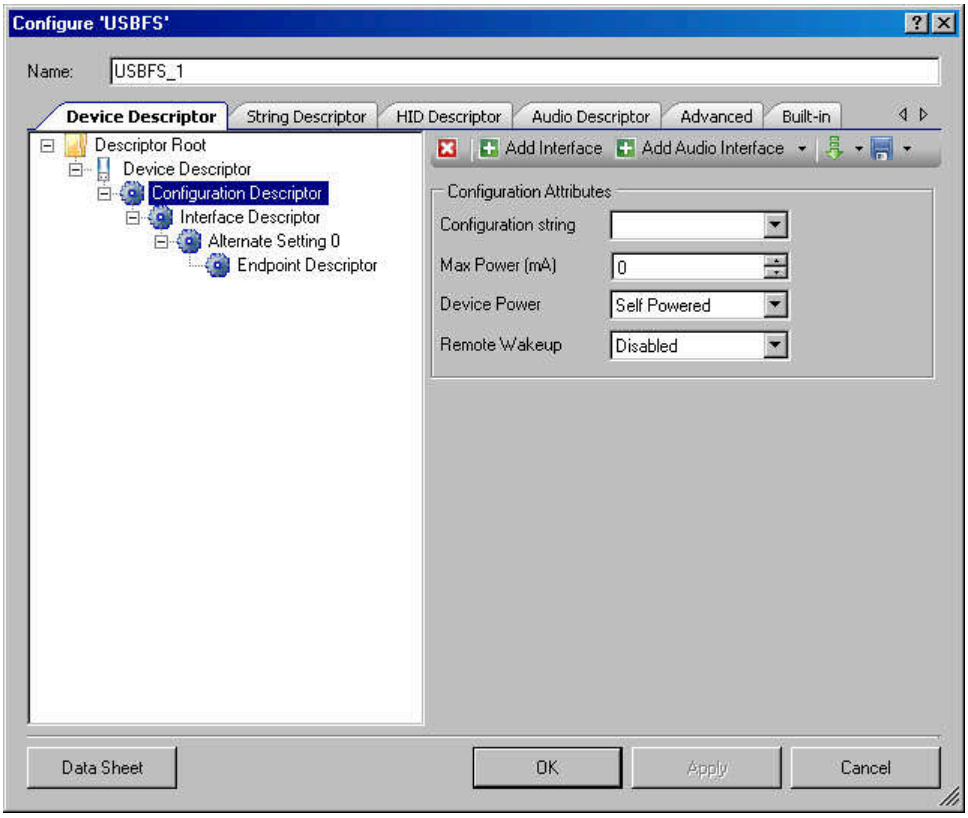
- 手动(默认设置) – 选择此选项可使用 LoadInEP/ReadOutEP 加载和卸载端点缓冲区。

  - 静态分配 - 在 SET_CONFIGURATION 请求后立即分配端点的存储器。当多个备用设置使用相同的 EP 编号时，采用最大长度。
  - 动态分配 - 每个 SET_CONFIGURATION 和 SET_INTERFACE 请求后动态分配端点的存储器。当多个备用设置用于互斥 EP 设置时，此选项非常有用。
- 带有手动存储器管理的 DMA – 选择此选项除了公开 LoadInEP/ReadOutEP 函数外，还可以公开 DMA 接口寄存器。

  PSoC 3 不支持 USB 端点与其他外设器件之间的直接 DMA 数据操作。所有涉及 USB 端点的 DMA 数据操作（输入和输出）必须随主系统存储器终止或启动。

  需要 USB 端点与其他外设之间的直接 DMA 的应用程序必须使用两个 DMA 数据操作。作为 USB 端点与其他外设之间的中间步骤，这两个数据操作将数据移动到主系统存储器中。

配置描述符



配置属性

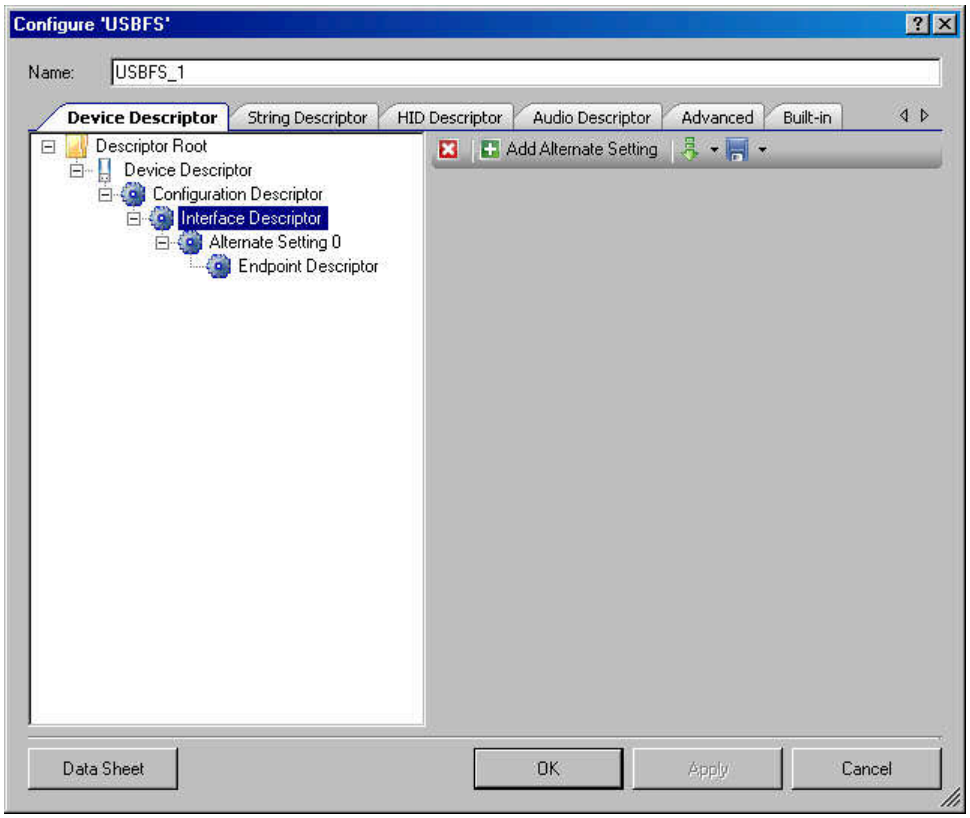
- 配置字符串
- 最大功耗 (mA) 在此特定配置中输入当器件完全运行时总线上 USB 器件的最大功耗。

注意：器件配置报告配置是采用总线供电还是自供电。器件状态报告器件当前是否采用自供电。如果器件断开与其外部电源的连接，它将更新器件状态以指示它不再采用自供电。当器件失去其外部电源时，不能将从总线得到的电源增加到超过其配置报告的数量。

- 器件供电 – 总线供电或自供电器件。USBFS 不支持同时使用这两种供电。
- 远程唤醒 – 使能或禁用

接口描述符

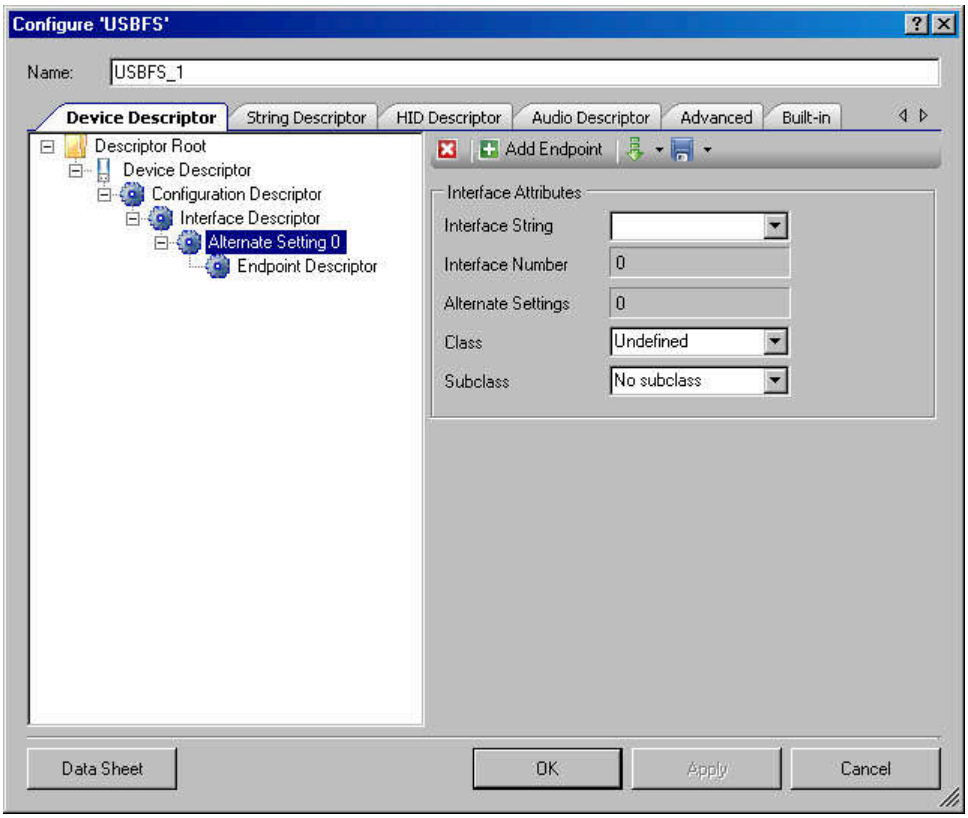
此级别用于添加和删除接口备用设置。接口在“备用设置”中进行配置。



自动提供**备用设置 0** 以配置您的器件。如果器件将使用同步端点，则 USB 2.0 规范要求所有器件默认接口设置不能包含任何具有非零数据有效负载大小的同步端点。这是通过端点描述符中的“最大数据包大小”指定的。

对于同步器件，应当使用备用接口设置（而不是默认备用设置 0）来为同步端点指定非零数据有效负载大小。另外，如果同步端点的数据有效负载大小非常大，建议使用其他备用配置或接口设置来指定数据有效负载大小范围。这可以增加该器件与其他 USB 器件成功组合使用的机会。

接口描述符 — 备用设置



接口属性

- 接口字符串
- 接口编号 — 接口编号由自定义程序计算
- 备用设置 — 备用设置由自定义程序计算
- 类 – HID、Vendor Specific 或 Undefined
- 子类 – 依赖于选定的类

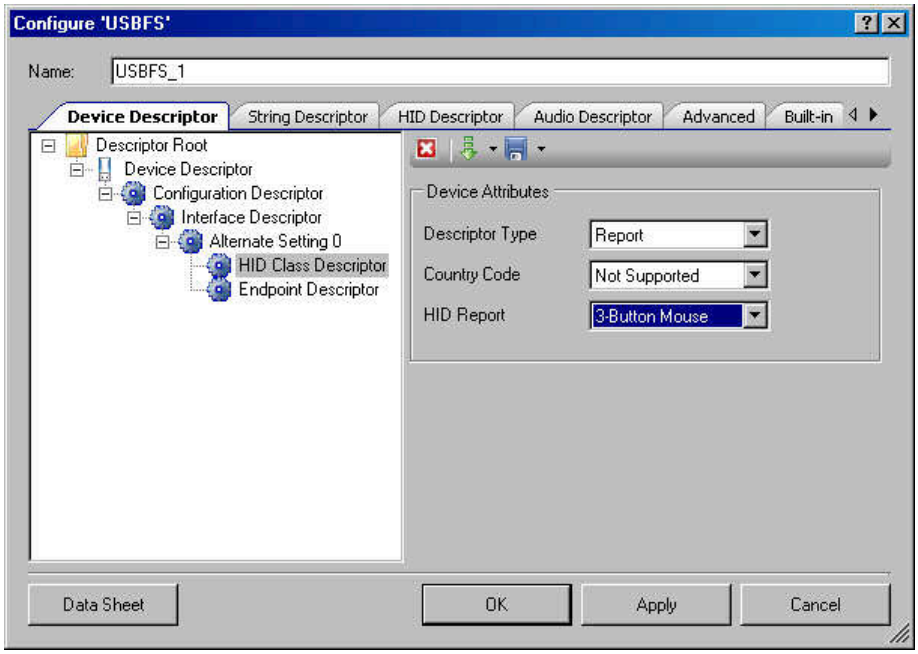
注意：字符串描述符是可选的。如果器件不支持字符串描述符，则器件、配置和接口描述符中对字符串描述符的所有引用必须设置为零。

HID 类描述符

默认情况下，不显示“HID 类描述符”项。它用于向备用设置中添加 HID 报告。

添加 HID 类描述符

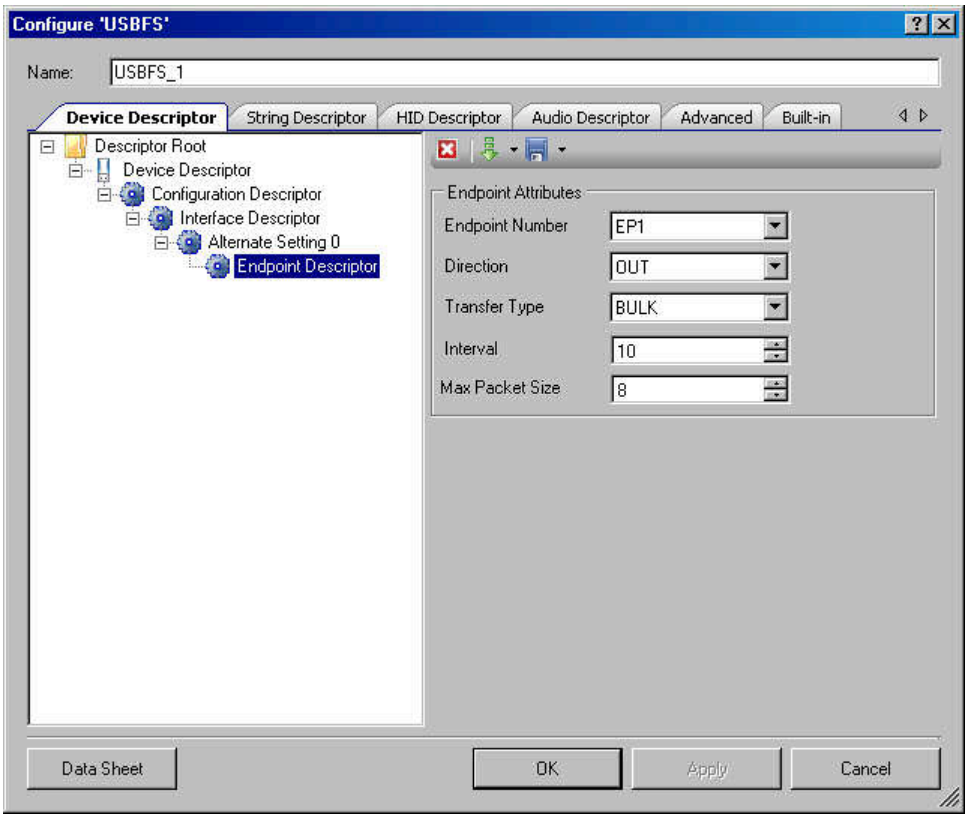
1. 在“描述符根目录”树中选择一个备用设置项。
2. 在右侧的**接口属性**下，为**类**字段选择“HID”。



器件属性

- 描述符类型 – 标识类描述符类型的常量名称。
- 国家/地区代码 – 标识本地化硬件的国家/地区代码的数字表达式。
- HID 报告 – 可用报告描述符列表。 报告描述符取自 **HID 描述符**选项卡。此字段为必填字段。

端点描述符

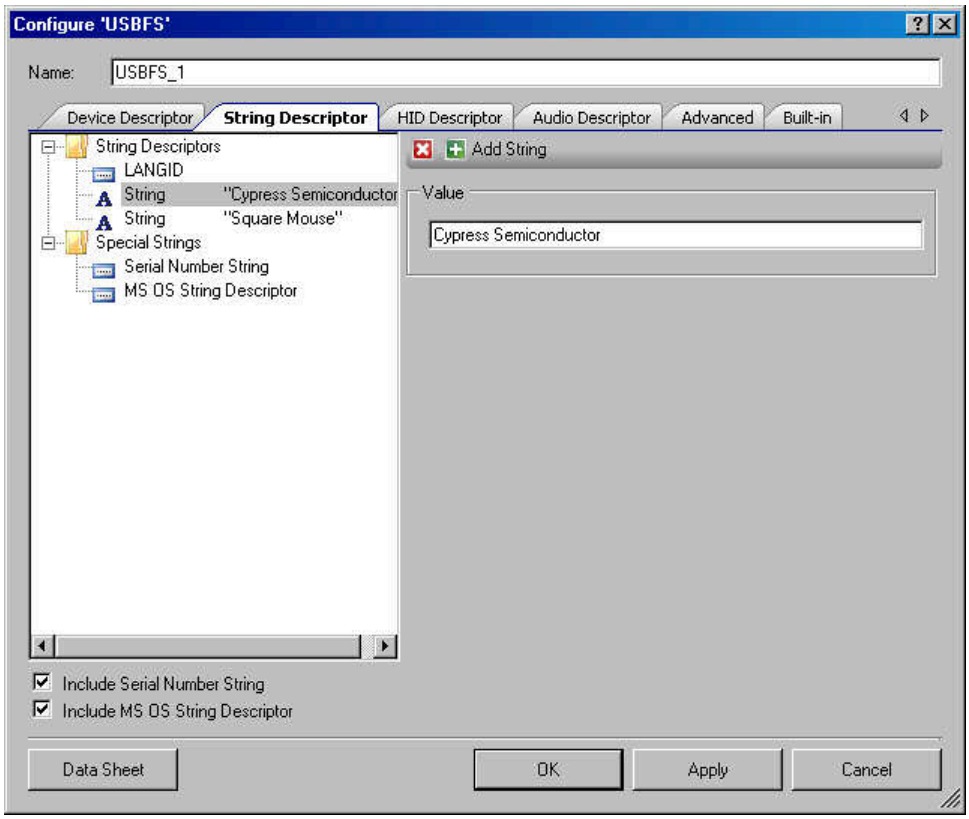


端点属性

- 端点编号
- 方向 –“输入”或“输出” USB 传输以主机为中心。因此输入指的是传输到主机；输出指的是从主机传输。
- 传输类型 – 控制传输、中断传输、批量传输或同步数据传输
- 间隔(ms) – 特定于此端点的轮询间隔。全速端点可以指定从 1 ms 到 255 ms 的所需周期。
- 最大数据包大小(字节) – 在全速器件中，对于批量或中断端点，最大数据包大小为 64 字节；对于同步端点，最大数据包大小为 1023 字节。

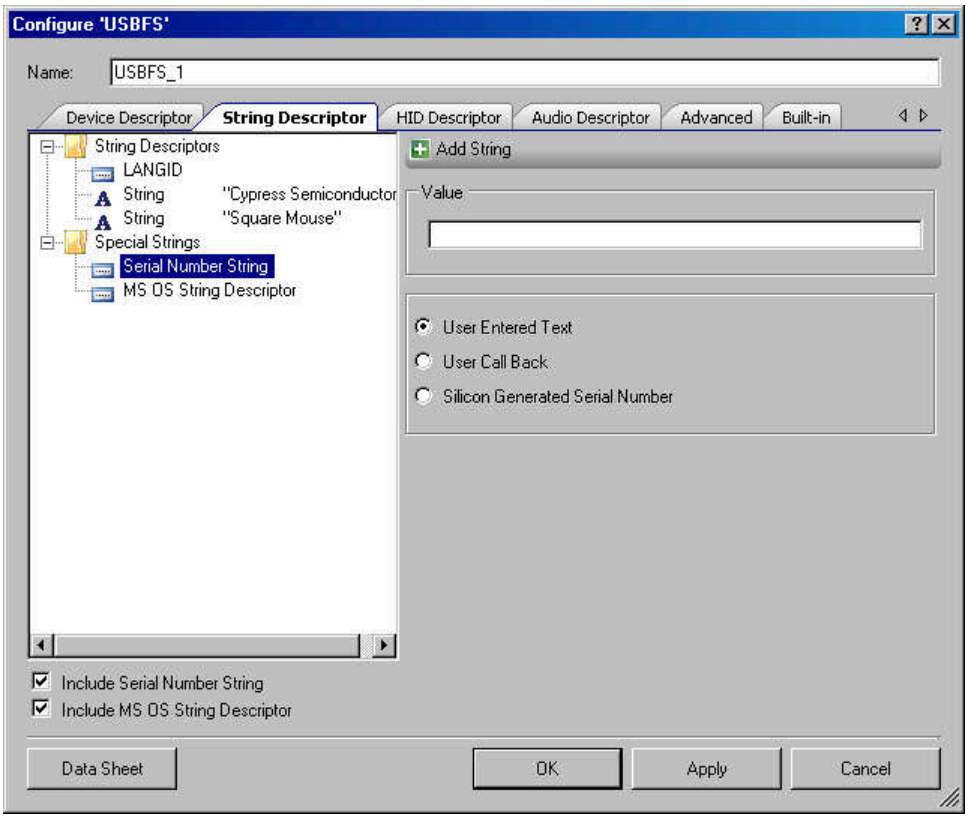
“字符串描述符”选项卡

字符串描述符



- LANGID – 语言 ID 选择。
- 字符串 – 字符串描述符的值。

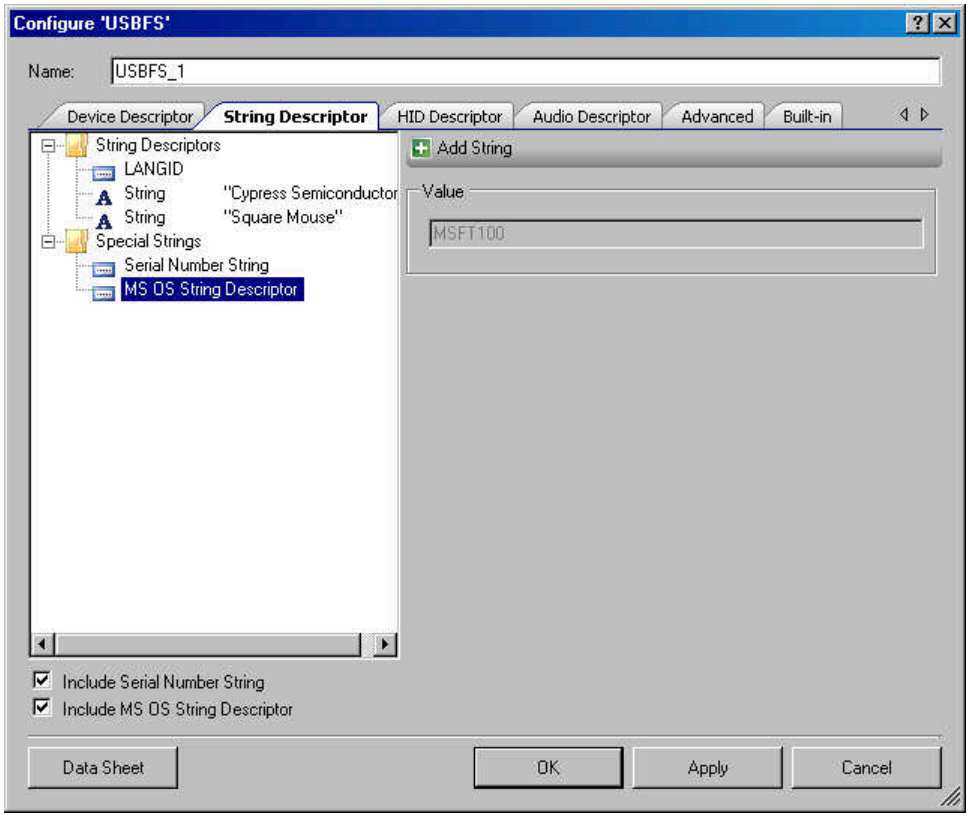
序列号字符串



- 值 – 默认字符串。
- 用户输入的文本 – 使能“值”文本框。
- 用户回调 – USBFS_SerialNumString 函数设置指针以使用用户生成的序列号字符串描述符。应用程序固件可以在运行时期间提供 USB 器件描述符的序列号字符串的源。
- 芯片生成的序列号

MS 操作系统字符串描述符

Microsoft 操作系统描述符为 USB 器件提供一种方法来向最新 Microsoft 操作系统提供附加配置信息

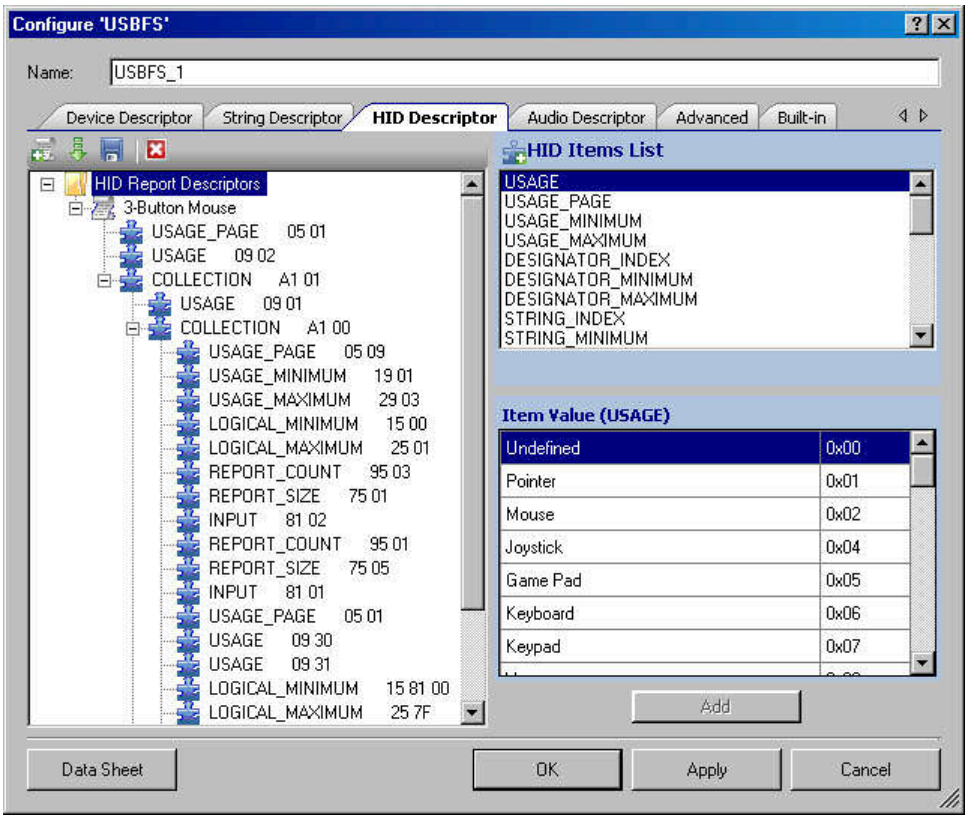


- 值 – 常量字符串“MSFT100”

HID 描述符选项卡

HID 描述符选项卡使您可以为器件快速构建 HID 描述符。使用**添加报告**按钮可添加和配置 HID 报告描述符。

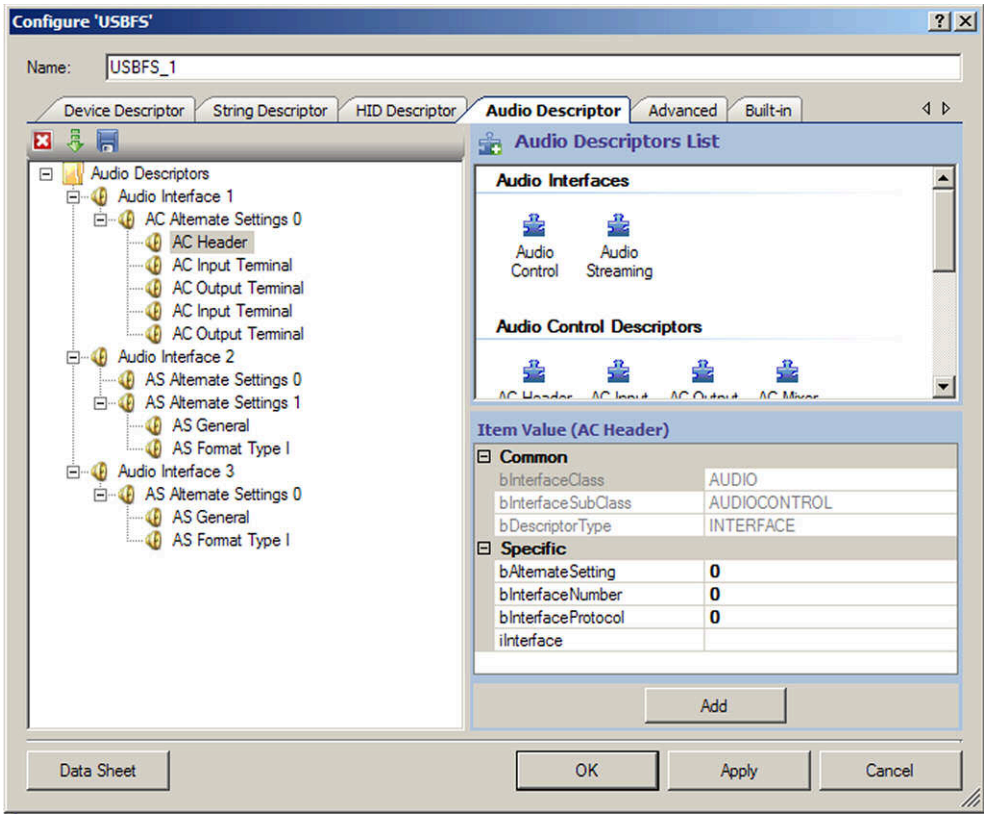
HID 描述符



- HID 项目列表 – 要在 HID 报告中添加的项目。
- 项目值 – 在 HID 项目列表或树中选择的项目的值。

“音频描述符”选项卡

音频描述符选项卡用于添加和配置音频接口描述符。



添加音频描述符

1. 在左侧树中选择“音频描述符”根项目。
2. 在右侧的**“音频描述符列表”**下，选择“音频控制”或“音频流”接口。
3. 在“项目值” 下，根据需要输入 bAlternateSetting 和 bInterfaceNumber 值。其他字段可选。

   注意：这些值是手动设置的。比较而言，对于一般接口描述符，这些值是自动设置的。
4. 单击**“添加”**，描述符添加到左侧的树中。

   您可以通过选择节点并单击它来重命名“音频接口 x”标题。

添加特定于类的音频控制或音频流接口描述符

1. 在左侧的树中选择相应的“AC 备用设置 x”或“AS 备用设置 x”项目。
2. 在右侧的**“音频描述符列表”**下，根据需要选择“音频控制描述符” 或 “音频流描述符”下的项目之一。
3. 在**“项目值”**下，在“特定”下输入相应的值。

4. 单击**“添加”**，描述符添加到左侧的树中。

添加音频端点描述符

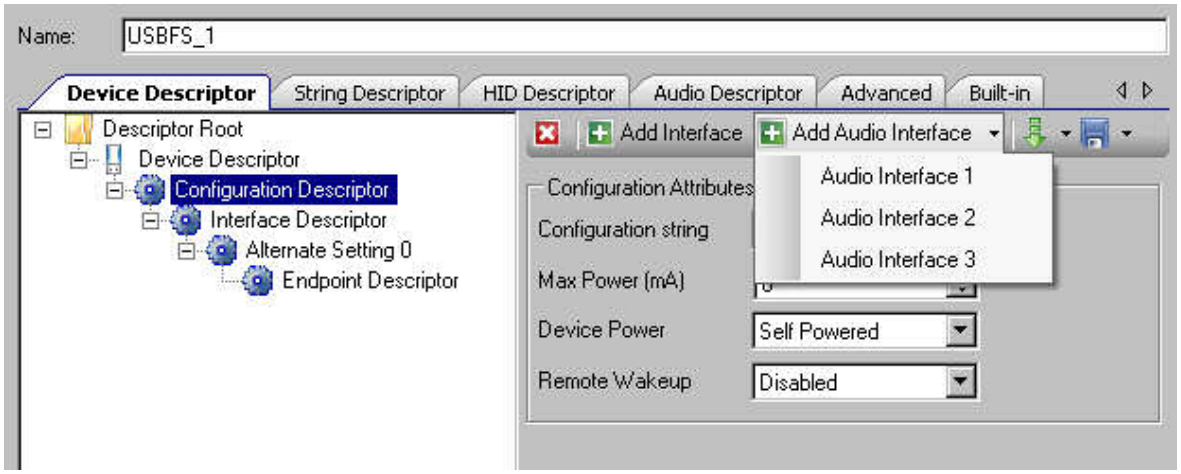
1. 在左侧的树中选择相应的“AC 备用设置 x”或“AS 备用设置 x”项目。
2. 在右侧的**“音频描述符列表”**下，选择“端点描述符”项目。
3. 在**“项目值”**下，在“特定”下输入相应的值。
4. 单击**“添加”**，描述符添加到左侧的树中。

添加标准 AS 同步端点描述符

1. 在左侧的树中选择相应的“端点描述符”。
2. 在右侧的**“音频描述符列表”**下，选择“AS 端点描述符”。
3. 在**“项目值”**下，在“特定”下输入相应的值。
4. 单击**“添加”**，描述符添加到左侧的树中。

将配置的音频接口描述符添加到器件描述符树中

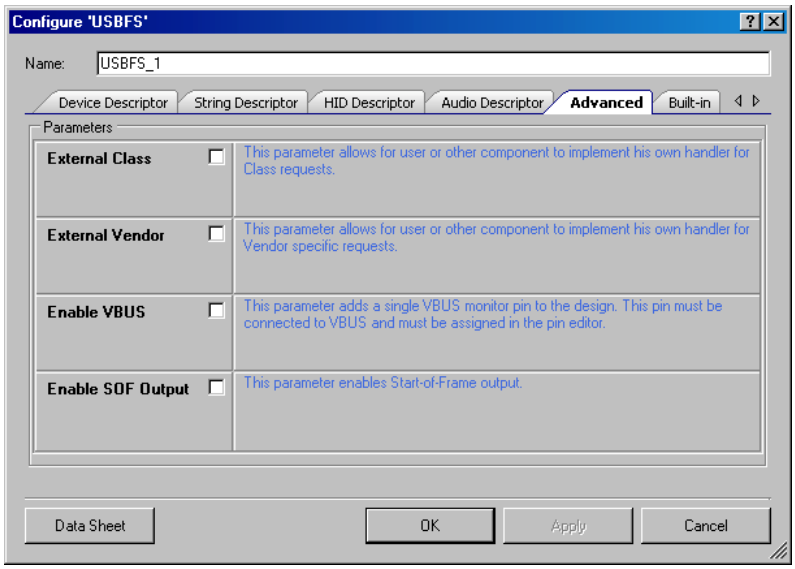
1. 转到**“器件描述符”**选项卡。
2. 选择新接口将所属的“配置描述符”。
3. 单击**“添加音频接口”**工具按钮，选择要添加的相应项目。



音频接口在**“器件描述符”**选项卡列表中将灰显，这是因为它们只能在**“音频描述符”**选项卡中编辑。

注意：单击**应用**或**确定**将保存各个选项卡上的更改。如果单击**取消**，将不保存添加的所有描述符。

“高级”选项卡



外部类

此参数允许用户固件（或其他位于解决方案级别的组件）提供对类请求的处理。如果使能此参数，应当实现 USBFS_DispatchClassRqst() 函数。

外部供货商

此参数允许用户固件（或其他位于解决方案级别的组件）提供对供货商特定请求的处理。如果使能此参数，应当实现 USBFS_HandleVendorRqst() 函数。

使能 VBUS 监控

USB 规范要求在任何时候都不应当有器件在面向上游的端口处为 VBUS 提供电流。为了满足此要求，器件必须监控 VBUS 是否存在，如果 VBUS 不存在，则去除来自 D+/D- 上拉电阻的供电。

对于总线供电设计，当从主机拔除 USB 电缆时，肯定会去除电源；但是对于自供电设计，器件必须符合此要求，才能进行正确的操作和通过 USB 认证。

此参数将单一 VBUS 监控引脚添加到设计中。此引脚必须连接到 VBUS 且必须在引脚编辑器中进行分配。有关其他信息，请参见“自供电器件的 USB 符合性”一节。

使能 SOF 输出

此参数使能“帧起始”输出。

放置

USB 作为固定功能模块实现。

资源

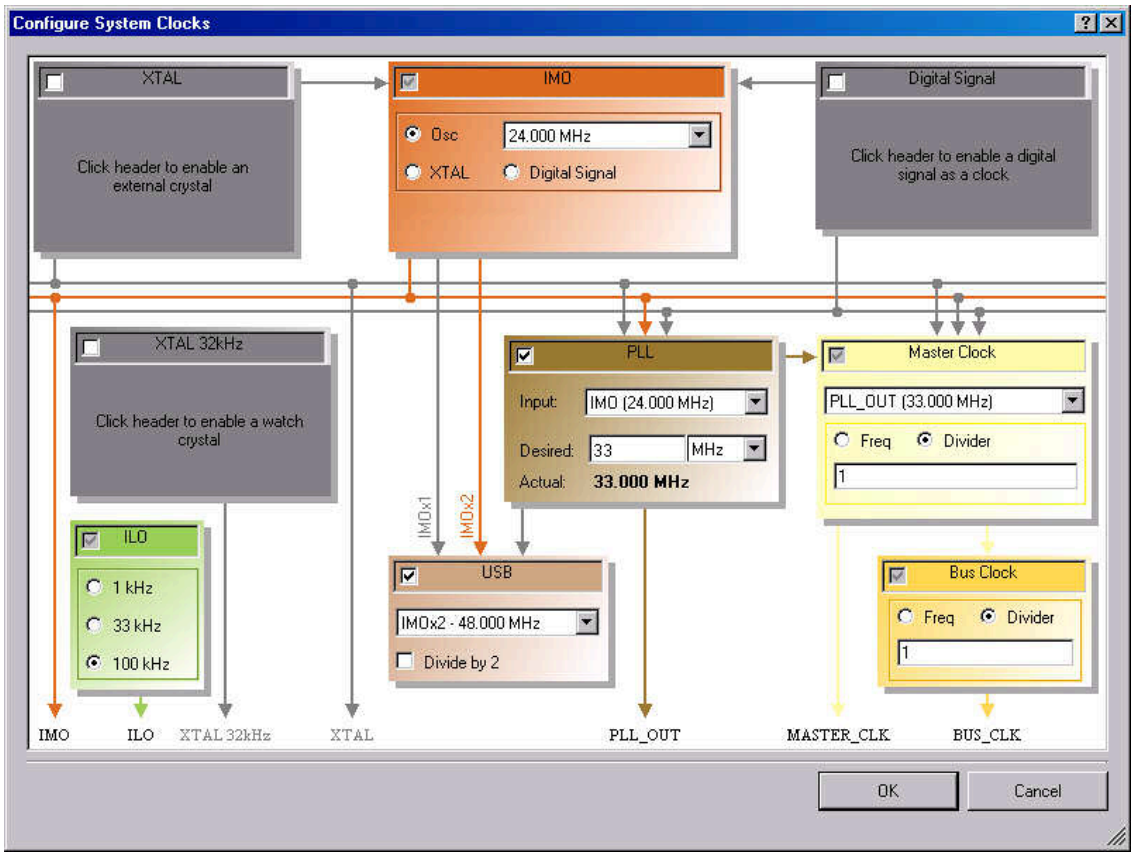
资源	资源类型				API 内存（字节）		引脚（每个外部 I/O）
	时钟分频器	宏单元	中断	USB 固定模块	闪存	RAM	
USBFS	0	0	7	1	3767	119	2

时钟设置

USB 硬件模块要求通过 PSoC Creator 设计范围资源时钟编辑器配置系统时钟。当使用 USBFS 组件时，时钟设置具有下列要求：

- 必须使能 USB 时钟。
- ILO 必须设置为 100 KHz。
- 如果选择的器件为 (PSoC 3 ES2) 或 (PSoC 5)，则总线时钟不能慢于 33 MHz [对于主控时钟，请选择 PLL_OUT (33.000 MHz)]。

有一些方法可将系统时钟配置为符合这些要求。下面显示了一组可使用的选项。您的设计可能需要不同的设置。



应用程序编程接口

应用程序编程接口 (API) 子程序使您可以使用软件来配置组件。下表列出并描述了每个函数的接口。后面的章节将更详细地介绍每个函数。

默认情况下，PSoC Creator 实例名称“USBFS_1”分配给提供的设计中的第一个组件实例。您可以将其重命名为遵循标识符语法规则的任何唯一值。该实例名称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“USBFS”。

基本 USBFS 器件 API

函数	说明
USBFS_Start	激活要用于器件和特定电压模式的组件。
USBFS_Init	初始化组件的硬件。
USBFS_InitComponent	初始化组件的全局变量，通过上拉 D+ 线路启动与主机的通信。
USBFS_Stop	禁用组件。
USBFS_GetConfiguration	返回当前分配的配置。如果未配置器件，则返回 0。

函数	说明
USBFS_GetInterfaceSetting	返回指定接口的当前备用设置。
USBFS_GetEPState	返回指定 USBFS 端点的当前状态。
USBFS_GetEPAckState	通过返回非零值来标识是否设置了 ACK。
USBFS_GetEPCount	从指定 USBFS 端点返回当前字节计数。
USBFS_LoadInEP	针对 IN 传输，加载并使能指定的 USBFS 端点。
USBFS_ReadOutEP	从端点 RAM 读取指定数量的字节，并将其放入 pSrc 指向的 RAM 阵列。该函数返回主机发送的字节数。
USBFS_EnableOutEP	使能指定的 USB 端点以接收 OUT 传输
USBFS_DisableOutEP	禁止指定的 USB 端点 NAK OUT 传输
USBFS_SetPowerStatus	将器件设置为自供电或总线供电
USBFS_Force	在 USB D+/D- 引脚上强制 J、K 或 SE0 状态 通常用于远程唤醒。
USBFS_SerialNumString	在运行时提供 USB 器件序列号字符串描述符。
USBFS_TerminateEP	终止端点传输。

全局变量

变量	说明
USBFS_initVar	指示是否已初始化 USBFS。该变量初始化为 0，在第一次调用 USBFS_Start() 时设置为 1。这使得组件无需在第一次调用 USBFS_Start() 子程式后重新初始化便可重新启动。 如果需要组件重新初始化，则该变量应当在调用 USBFS_Start() 子程式之前设置为 0。另外，可以通过调用 USBFS_Init() 和 USBFS_InitComponent() 来重新初始化 USBFS。
USBFS_device	包含启动器件编号。此变量由 USBFS_Start() 或 USBFS_InitComponent() API 设置。
USBFS_transferState	此变量由通信函数用来处理当前传输状态。 在 USBFS_InitComponent() API 中以及在状态阶段完成传输后，初始化为 TRANS_STATE_IDLE。 根据请求类型，在“设置”数据操作中更改为 TRANS_STATE_CONTROL_READ 或 TRANS_STATE_CONTROL_WRITE。
USBFS_configuration	包含主机使用 SET_CONFIGURATION 请求设置的当前配置编号。此变量在 USBFS_InitComponent() API 中初始化为零，然后返回到 USBFS_GetConfiguration() API 的应用程序级别。

变量	说明
USBFS_deviceAddress	包含的当前器件地址。此变量在 USBFS_InitComponent() API 中初始化为零。主机开始与地址为 0 的器件进行通信，然后使用 SET_ADDRESS 请求将其设置为任意值。
USBFS_deviceStatus	这是两位变量，第一位中指示电源状态（DEVICE_STATUS_BUS_POWERED 或 DEVICE_STATUS_SELF_POWERED），第二位指示远程唤醒状态 (DEVICE_STATUS_REMOTE_WAKEUP)。此变量在 USBFS_InitComponent() API 中初始化为零，由 USBFS_SetPowerStatus() API 配置。

void USBFS_Start(uint8 device, uint8 mode)

说明： 执行 USBFS 组件的所有必需初始化。

参数： (uint8) device：包含使用 USBFS 自定义程序输入的所需器件描述符集中的器件编号。

(uint8) mode：工作电压。它确定是否为 5V 操作使能了电压稳压器，或者是否为 3.3 V 操作使用了通过模式。下表中给出了象征名称及其相关值。

电源设置	注释
USBFS_3V_OPERATION	对于上拉，禁用电压稳压器和通过 Vcc
USBFS_5V_OPERATION	使能电压稳压器，将稳压器用于上拉
USBFS_DWR_VDDD_OPERATION	根据 DWR 中的 Vddd 电压配置，使能或禁用电压稳压器。

返回值： 无

副作用： 无

void USBFS_Init(void)

说明： 根据自定义程序的“配置”对话框设置，初始化或恢复组件。不需要调用 USBFS_Init()，因为 USBFS_Start() 子程式会调用此函数，这是开始组件操作的首选方法。

参数： 无

返回值： 无

副作用： 无

void USBFS_InitComponent (uint8 device, uint8 mode)

说明： 初始化组件的全局变量，通过上拉 D+ 线路启动与主机的通信。

参数： (uint8) device：包含使用 USBFS 自定义程序输入的所需器件描述符集中的器件编号。

(uint8) mode：工作电压。它确定是否为 5V 操作使能了电压稳压器，或者是否为 3.3 V 操作使用了通过模式。下表中给出了象征名称及其相关值。

电源设置	注释
USBFS_3V_OPERATION	对于上拉，禁用电压稳压器和通过 Vcc
USBFS_5V_OPERATION	使能电压稳压器，将稳压器用于上拉
USBFS_DWR_VDDD_OPERATION	根据 DWR 中的 Vddd 电压配置，使能或禁用电压稳压器。

返回值： 无

副作用： 此函数调用自用于初始化通信的复位 ISR。

void USBFS_Stop(void)

说明： 执行 USBFS 组件所需的所有必需关闭任务。

参数： 无

返回值： 无

副作用： 无

uint8 USBFS_GetConfiguration(void)

说明： 获取 USB 器件的当前配置。

参数： 无

返回值： uint8: 返回当前分配的配置。如果未配置器件，则返回 0。

副作用： 无

uint8 USBFS_GetInterfaceSetting(uint8 interfaceNumber)

说明： 获取指定接口的当前备用设置。

参数： uint8 interfaceNumber：接口编号

返回值： uint8: 返回指定接口的当前备用设置。

副作用： 无

uint8 USBFS_GetEPState(uint8 epNumber)

说明： 返回请求的端点的状态。

参数： (uint8) epNumber：数据端点编号。

返回值： (uint8) 返回指定 USBFS 端点的当前状态。下表中给出了提供的象征名称及其相关值。当您编写代码以更改端点状态（例如编写 ISR 代码以处理发送或接收的数据）时，使用这些常量。

返回值	说明
USBFS_NO_EVENT_PENDING	指示端点正在等待 SIE 操作
USBFS_EVENT_PENDING	指示端点正在等待 CPU 操作
USBFS_NO_EVENT_ALLOWED	指示端点已被锁定，无法访问
USBFS_IN_BUFFER_FULL	加载 IN 端点，模式设置为 ACK IN
USBFS_IN_BUFFER_EMPTY	发生 IN 数据操作，可以加载更多数据
USBFS_OUT_BUFFER_EMPTY	OUT 端点设置为 ACK OUT，正在等待数据
USBFS_OUT_BUFFER_FULL	发生 OUT 数据操作，可以读取数据

副作用： 无

uint8 USBFS_GetEPAckState(uint8 epNumber)

说明： 通过读取端点控制寄存器中的 ACK 位，确定此端点上是否发生 ACK 数据操作。此函数不清除 ACK 位。

参数： (uint8) epNumber：包含数据端点编号。

返回值： (uint8)：如果发生 ACK 数据操作，则此函数返回非零值。否则返回零。

副作用： 无

uint16 USBFS_GetEPCount(uint8 epNumber)

说明： 返回请求的端点的传输计数。计数寄存器提供的值数据包含 2 个计数，对应于包的两字节校验和。此函数减去这两个计数。

参数： (uint8) epNumber：包含数据端点编号。

返回值： (uint16)：返回指定 USBFS 端点提供的当前字节计数，或者针对无效端点返回 0。

副作用： 无

void USBFS_LoadInEP(uint8 epNumber, uint8 *pData, uint16 length)

说明： 针对 IN 中断或批量传输，加载并使能指定的 USB 数据端点。

参数： (uint8) epNumber：包含数据端点编号。

(uint8) *pData：指向从其加载端点空间数据的数据数组的指针。

(uint16) length：要从数组传输、然后作为 IN 请求结果发送的字节数。有效值介于 0 到 512 之间。

返回值： 无

副作用： 无

uint16 USBFS_ReadOutEP(uint8 epNumber, uint8 *pData, uint16 length)

说明： 将指定数量的字节从端点 RAM 移动到数据 RAM。从端点 RAM 实际传输到数据 RAM 的字节数是主机发送的实际字节数和 wCount 参数请求的字节数二者中的较小者。

参数： (uint8) epNumber：包含数据端点编号。

(uint8) *pData：指向从其加载端点空间数据的数据数组的指针。

(uint16) length：要从 USB OUT 端点传输并加载到数据数组中的字节数。有效值介于 0 到 512 之间。如果主机发送的字节少于请求的字节数，则该函数移动的字节少于请求字节数。

返回值： (uint16)：接收的字节数

副作用： 无

void USBFS_EnableOutEP(uint8 epNumber)

说明： 针对 OUT 批量传输或中断传输，使能指定的端点。不要为 IN 端点调用此函数。

参数： (uint8) epNumber：包含数据端点编号。

返回值： 无

副作用： 无

void USBFS_DisableOutEP(uint8 epNumber)

说明： 禁用指定的 USBFS OUT 端点。不要为 IN 端点调用此函数。

参数： (uint8) epNumber：包含数据端点编号。

返回值： 无

副作用： 无

void USBFS_SetPowerStatus(uint8 powerStatus)

说明： 设置当前电源状态。器件将根据此值答复 USB GET_STATUS 请求。它使得器件能够正确报告其状态是否符合 USB 第 9 章要求。器件可以随时将其电源从自供电更改为总线供电，并将其当前电源报告为器件状态的一部分。您应当在器件从自供电更改为总线供电或从总线供电更改为自供电时调用此函数，并相应设置状态。

参数： (uint8) powerStatus：包含所需的电源状态，1 表示自供电，0 表示总线供电。下面提供了象征名称及其相关值：

电源状态	说明
USBFS_DEVICE_STATUS_BUS_POWERED	将器件设置为总线供电。
USBFS_DEVICE_STATUS_SELF_POWERED	将器件设置为自供电。

返回值： 无

副作用： 无

void USBFS_Force(uint8 state)

说明： 在 D+/D- 线路上强制 USB J、K 或 SE0 状态。此函数为 USB 器件应用程序提供必需的机制来执行 USB 远程唤醒。有关更多信息，请参考 USB 2.0 规范以了解有关挂起和恢复的详细信息。

参数： (uint8) state：一个字节，用于指示要使能四个总线状态中的哪个状态。下面提供了象征名称及其相关值：

状态	说明
USBFS_FORCE_SE0	在 D+/D- 线路上强制单端 0
USBFS_FORCE_J	在 D+/D- 线路上强制 J 状态
USBFS_FORCE_K	在 D+/D- 线路上强制 K 状态
USBFS_FORCE_NONE	将总线返回 SIE 控制

返回值： 无

副作用： 无

void USBFS_SerialNumString(uint8 *snString)

说明： 只有当选择了“序列号字符串”描述符属性中的“用户回调”选项时，此函数才可用。应用程序固件可以在运行时期间提供 USB 器件序列号字符串描述符的源。如果应用程序固件不使用此函数或设置了错误的字符串描述符，将使用默认字符串。

参数： (uint8) *snString：指向用户定义的字符串描述符的指针。字符串描述符应当符合通用串行总线 2.0 版第 9.6.7 章的要求。

返回值： 无

副作用： 无

void USBFS_TerminateEP(uint8 epNumber)

说明： 终止指定的 USBFS 端点。此函数可在端点重新配置之前使用。

参数： (uint8) epNumber：包含数据端点编号。

返回值： 无

副作用： 器件响应 NAK 到终止的端点的 IN 或 OUT 令牌。

人机接口装置 (HID) 类支持

函数	说明
USBFS_UpdateHIDTimer	更新指定接口的 HID 报告定时器，如果定时器到期，则返回 1，如果未到期，则返回 0。如果定时器到期，它将重新加载定时器。
USBFS_GetProtocol	返回指定接口的协议

全局变量

变量	说明
USBFS_hidProtocol	此变量在 USBFS_InitComponent() API 中初始化为 PROTOCOL_REPORT 值。主机使用 HID_SET_PROTOCOL 请求来控制它。USBFS_GetProtocol() API 将该值返回给用户代码。
USBFS_hidIdleRate	此变量控制 HID 报告速率。主机使用 HID_SET_IDLE 请求来控制它，USBFS_UpdateHIDTimer() API 使用它来重新加载定时器。
USBFS_hidIdleTimer	此变量包含定时器计时器，USBFS_UpdateHIDTimer() API 对该定时器进行递减和重新加载。

uint8 USBFS_UpdateHIDTimer(uint8 interface)

说明： 更新 HID 报告空闲定时器并返回状态。如果定时器到期，则重新加载定时器。

参数： (uint8) interface：包含接口编号。

返回值： (uint8)：返回 HID 定时器的状态。下面提供了象征名称及其相关值：

返回值	注释
USBFS_IDLE_TIMER_EXPIRED	定时器已到期。
USBFS_IDLE_TIMER_RUNNING	定时器正在运行。
USBFS_IDLE_TIMER_IDEFINITE	返回当数据或状态更改时是否发送报告。

副作用： 无

uint8 USBFS_GetProtocol(uint8 interface)

说明： 返回所选接口的 HID 协议值。

参数： (uint8) interface：包含接口编号。

返回值： (uint8)：返回协议值。

副作用： 无

引导加载程序支持

USBFS 组件可以用作引导加载程序的通信组件。应当使用下列配置来支持从外部系统到引导加载程序的通信协议：

- 端口号：EP1；方向：OUT；传输类型：INT；最大数据包大小：64
- 端口号：EP2；方向：IN；传输类型：INT；最大数据包大小：64

模板文件 (*bootloader.root.xml*) 中存储了完整的建议配置。在“器件描述符”树中选择“描述符根”，单击“导入”按钮，浏览到以下目录，然后打开 *bootloader.root.xml* 文件。

<INSTALL>\psoc\content\cycomponentlibrary\CyComponentLibrary.cylib\USBFS_v1_60\Custom\template\

有关引导加载程序的更多信息，请参见系统参考指南。

USBFS 组件为引导加载程序的使用提供了一组 API 函数。

函数	说明
USBFS_CyBtldrCommStart	执行针对 USBFS 组件的所有必需初始化，等待仿真，然后使能通信。
USBFS_CyBtldrCommStop	调用 USBFS_Stop 函数。
USBFS_CyBtldrCommReset	复位接收和传输通信缓冲区。
USBFS_CyBtldrCommWrite	允许调用程序将数据写入引导加载程序主机。该函数将处理轮询，以允许数据模块完全发送到主机设备。
USBFS_CyBtldrCommRead	允许调用程序从引导加载程序主机读取数据。该函数将处理轮询，以允许从主机设备完全接收到数据模块。

void USBFS_CyBtldrCommStart(void)

说明： 执行针对 USBFS 组件的所有必需初始化，等待仿真，然后使能通信。

参数： 无

返回值： 无

副作用： 此函数针对 3 V 操作启动 USBFS。

void USBFS_CyBtldrCommStop(void)

说明： 执行 USBFS 组件所需的所有必需关闭任务。
参数： 无
返回值： 无
副作用： 调用 USBFS_Stop() 函数。

void USBFS_CyBtldrCommReset(void)

说明： 复位接收和传输通信缓冲区。
参数： 无
返回值： 无
副作用： 无

cystatus USBFS_CyBtldrCommWrite(uint8 *data, uint16 size, uint16 *count, uint8 timeOut)

说明： 允许调用程序将数据写入引导加载程序主机。该函数将处理轮询，以允许数据模块完全发送到主机设备。
参数： (uint8) *data：指向要发送到器件的数据模块的指针。
(uint16) size：要写入的字节数。
(uint16) *count：指向无符号 short 型变量的指针，该变量用于写入实际写入的字节数。
(uint8) timeout：在由于超时而返回之前等待的单位数。
返回值： (cystatus)：如果未遇到问题，则返回 1；或者返回最能描述问题的值。
副作用： 无

cystatus USBFS_CyBtldrCommRead(uint8 *data, uint16 size, uint16 *count, uint8 timeOut)

说明： 允许调用程序从引导加载程序主机读取数据。该函数将处理轮询，以允许从主机设备完全接收到数据模块。
参数： (uint8) *data：指向用于存储从器件接收的数据模块的区域的指针。
(uint16) size：要读取的字节数。
(uint16) *count：指向无符号 short 型变量的指针，该变量用于写入实际读取的字节数。
(uint8) timeOut：在由于超时而返回之前等待的单位数。
返回值： (cystatus)：如果未遇到问题，则返回 1；或者返回最能描述问题的值。
副作用： 无

USB 挂起、恢复和远程唤醒

USBFS 组件支持 USB 挂起、恢复和远程唤醒。由于这些功能紧密连接到用户应用程序，因此 USBFS 组件提供了一组 API 函数。

函数	说明
USBFS_CheckActivity	检查并清除 USB 总线活动标志。如果自从上次检查后 USB 处于活动状态，则返回 1；否则返回 0。
USBFS_Suspend	此函数禁用 USBFS 模块，并准备断电模式。
USBFS_Resume	此函数在断电模式后使能 USBFS 模块。
USBFS_RWUEnabled	此函数返回当前远程唤醒状态。

uint8 USBFS_CheckActivity(void)

说明： 返回总线的活动状态。清除状态硬件以便在下次调用此子程式时提供最新活动状态。
此函数提供一种方法来检查是否发生任何 USB 总线活动。应用程序使用此函数确定是否满足进入 USB 挂起的条件。

参数： 无

返回值： (uint8) cystatus：标准 API 返回值。

返回值	说明
1	如果自从上次调用此函数后检测到总线活动
0	如果自从上次调用此函数后未检测到总线活动

副作用： 无

void USBFS_Suspend(void)

说明： 此函数禁用 USBFS 模块，并准备断电模式。应当刚好在进入睡眠之前调用。
一旦符合进入 USB 挂起的条件，应用程序将采取适当的步骤减小当前消耗，以满足挂起电流要求。要将 USB SIE 和收发器置于断电模式，应用程序需要调用 USBFS_Suspend() API 函数和 USBFS_CheckActivity() API 来检测 USB 活动。此函数禁用 USBFS 模块，但是维护当前 USB 地址（在 USBCR 寄存器中）。该器件使用睡眠功能减少功耗。

参数： void

返回值： void

副作用：

void USBFS_Resume(void)

说明： 此函数在断电模式后使能 USBFS 模块。应当刚好在从睡眠唤醒后调用。
在器件挂起时，它定期检查以确定是否符合离开挂起状态的条件。检查恢复条件的一种方法是使用睡眠定时器定期唤醒器件。如果符合恢复条件，则应用程序调用 USBFS_Resume() API 函数。此函数使能 USBFS SIE 和收发器，使它们脱离断电模式。它不更改 USBCR 寄存器的 USB 地址字段，维护主机以前分配的 USB 地址。

参数： void

返回值： void

副作用：

uint8 USBFS_RWUEnabled(void)

说明： 此函数返回当前远程唤醒状态。
如果器件支持远程唤醒，则应用程序能够确定主机是否使用 USBFS_RWUEnabled() API 函数使能了远程唤醒。当器件挂起且它确定符合启动远程唤醒的条件时，应用程序使用 USBFS_Force() API 函数在 USB 上强制相应的 J 和 K 状态，标志着远程唤醒。

参数： void

返回值： TRUE - 使能远程唤醒
FALSE - 禁用远程唤醒

副作用：

音频类支持

全局变量

变量	说明
USBFS_currentSampleFrequency	包含当前音频采样频率。主机使用发往端点的 SET_CUR 请求来设置它。

USBFS_ frequencyChanged	此变量用作用户代码的标志，一定要注意，已经向主机发送了更改采样频率的请求。将在下一个 OUT 数据操作时发送采样频率。它包含设置时的端点地址。在主代码中检测新采样频率时建议使用以下代码：
	if((USBFS_frequencyChanged != 0) && (USBFS_transferState == USBFS_TRANS_STATE_IDLE))
	{
	/* Add core here.*/
	USBFS_frequencyChanged = 0;
	}
	检查 USBFS_transferState 变量以确保传输完成。

USBFS_currentMute	USBFS_currentMute：包含主机设置的静音配置。

变量	说明
USBFS_currentVolume	USBFS_currentVolume：包含主机设置的音量级别。

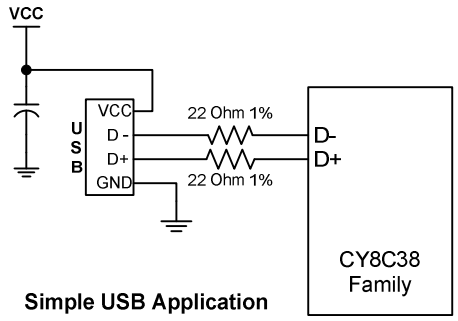
固件源代码示例

PSoC Creator 在“查找示例项目”对话框中提供大量示例项目，其中包括图示和示例代码。若要获取特定于组件的示例，请从“组件目录”或图示组件实例打开该对话框。若要获取一般示例，请从开始页或**文件**菜单打开该对话框。根据需要，在对话框中使用**筛选选项**以缩小可用于选择的项目列表的范围。

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

功能说明

下图显示了一个简单的总线供电 USB 应用，它连接了 PSoC 器件的 D+ 和 D- 引脚。



Simple USB Application

USB 符合性

USB 可以向器件提供各种总线条件，包括总线复位和不同时序要求。提供的示例中并不能正确阐述所有这些条件和要求。您应当确保设计符合 USB 规范的应用程序。

自供电器件的 USB 符合性

如果您要创建的器件采用自供电，则必须通过电阻式网络将 GPIO 引脚连接到 VBUS，并写入固件以监控 GPIO 的状态。可以使用 USBFS_Start() 和 USBFS_Stop() API 子程式来控制 D+ 和 D- 引脚上拉。在您调用 USBFS_Start() 之前，上拉电阻不会向数据线供电。USBFS_Stop() 断开上拉电阻与数据引脚的连接。

根据使用 USBFS_SetPowerStatus() 函数设置的状态，器件对 GET_STATUS 请求做出响应。如果您的器件配置为自供电，若要设置正确的状态，应当至少调用一次 USBFS_SetPowerStatus()。您还应当在器件更改状态时调用 USBFS_SetPowerStatus() 函数。

USB 标准器件请求

本节介绍 USBFS 组件支持的请求。如果不支持请求，则 USBFS 组件使用 STALL 进行响应，表示请求错误。

标准器件请求	USB 组件支持说明	USB 2.0 规范章节
CLEAR_FEATURE	器件：	9.4.1
	接口：	
	端点	
GET_CONFIGURATION	返回当前器件配置值。	9.4.2
GET_DESCRIPTOR	返回指定的描述符。	9.4.3
GET_INTERFACE	返回指定接口的所选备用接口设置。	9.4.4
GET_STATUS	器件：	9.4.5
	接口：	
	端点：	
SET_ADDRESS	设置所有将来器件访问的器件地址。	9.4.6
SET_CONFIGURATION	设置器件配置。	9.4.7
SET_DESCRIPTOR	不支持此可选请求。	9.4.8
SET_FEATURE	器件： DEVICE_REMOTE_WAKEUP 支持由 bRemoteWakeUp 组件参数选择。 不支持 TEST_MODE。	9.4.9
	接口：	
	端点：暂停指定的端点。	
SET_INTERFACE	此请求允许主机为指定的接口选择备用设置。	9.4.10
SYNCH_FRAME	不支持。组件的将来实现会添加对此请求的支持，以使能与重复帧模式的同步传输。	9.4.11

HID 类请求

类请求	USBFS 组件支持说明	HID 的器件类定义 - 章节
GET_REPORT	允许主机通过控制管线接收报告。	7.2.1

类请求	USBFS 组件支持说明	HID 的器件类定义 - 章节
GET_IDLE	读取特定输入报告的当前空闲速率。	7.2.3
GET_PROTOCOL	读取哪个协议当前处于活动状态（引导协议或报告协议）。	7.2.5
SET_REPORT	允许主机向器件发送报告，可能设置输入、输出或功能控制的状态。	7.2.2
SET_IDLE	停止有关“中断输入”管线的特定报告，直到发生新事件或者经过指定的时间量。	7.2.4
SET_PROTOCOL	在引导协议和报告协议之间切换（或反过来切换）。	7.2.6

音频类请求

类请求	USBFS 组件支持说明	音频器件类定义 - 章节
GET_CUR	接口： MUTE_CONTROL VOLUME_CONTROL	5.2.1.1
	端点： SAMPLING_FREQ_CONTROL	
SET_CUR	接口： MUTE_CONTROL VOLUME_CONTROL	5.2.1.2
	端点： SAMPLING_FREQ_CONTROL	

直流电和交流电电气特征

USB 直流电规范

参数	说明	条件	最小值	典型值	最大值	单位
V_{USB_5}	为 USB 操作提供的器件供电	配置了 USB，使能了 USB 调节器	4.35	–	5.25	V
$V_{USB_3.3}$		配置了 USB，不使用 USB 调节器	3.15	–	3.6	V
V_{USB_3}		配置了 USB，不使用 USB 调节器	2.85	–	3.6	V
$I_{USB_Configured}$	器件活动模式下的器件供电电流，总线时钟和 IMO = 24 MHz	V_{DDD} = 5 V	–	7	10	mA
		V_{DDD} = 3.3 V	–	5	8	mA
$I_{USB_Suspended}$	器件睡眠模式下的器件供电电流	V_{DDD} = 5 V，连接到 USB 主机，PICU 配置为在有 USB 恢复信号时唤醒	–	0.5	1.2	mA
		V_{DDD} = 5 V，断开与 USB 主机的连接	–	0.3	1.0	mA
		V_{DDD} = 3.3 V，连接到 USB 主机，PICU 配置为在有 USB 恢复信号时唤醒	–	0.5	1.2	mA
		V_{DDD} = 3.3 V，断开与 USB 主机的连接	–	0.3	1.0	mA

USB 驱动程序交流电规范

参数	说明	条件	最小值	典型值	最大值	单位
Tr	跃变上升时间		–	–	20	ns
Tf	跃变下降时间		–	–	20	ns
TR	上升/下降时间匹配		90%	–	111%	
Vcrs	输出信号交变电压		1.3	–	2	V

组件更改

本节列出组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
1.60	添加了函数 USBFS_TerminateEP(uint8 ep) 以便 NAK 端点。	可以在端点重新配置或器件模式切换之前使用此函数。
	将 USBFS_hidProtocol 变量初始化为 USBFS_InitComponent() 和 USBFS_reInitComonent() 函数中的 HID_PROTOCOL_REPORT 值。	为了符合 HID“7.2.6 Set_Protocol 请求”--- “初始化时，所有器件默认为报告协议”。
	添加了对发往接口的 SET_FEATURE/CLR_FEATURE 请求的支持。	为了通过 WHQL 测试。
	向 SET_IDLE 请求处理中添加了逻辑，以支持正确的时序。	为了符合 HID“7.2.4 Set_Idle 请求”
	添加了对音频类请求的支持：SET_CUR/CLR_CUR 到用于采样频率、静音和音量控制的接口和端点。	为了符合音频类定义“5.2.1.1 设置请求”和“5.2.1.2 获取请求”
	已重命名引导加载程序 API 以便提前具有实例名称。添加了向后兼容定义。	准备将来能够从多个接口引导。
	向数据手册中添加了特性数据	
	对数据手册进行了少量编辑和更新	
1.50.a	进行了数据手册日志累积更改	为了给客户提供方便。
1.50	已添加了 USB 挂起、恢复和远程唤醒功能。	USB 器件应当支持挂起和恢复功能。
	大多数 API 经过重命名以删除外文标记，支持旧名称以满足向后兼容。	为了符合公司代码标准。
	已添加了 GET_INTERFACE/SET_INTERFACE 请求支持。	如果器件具有接口的备用设置，则必须支持 GetInterface/SetInterface 请求。
	集成了特定 API 以支持引导加载程序：CyBtldrCommStart、CyBtldrCommStop、CyBtldrCommReset、CyBtldrCommWrite、CyBtldrCommRead。	USB 可以用作具有此功能的引导加载程序的通信组件。
	向数据手册中添加了泛型 USB 批量包裹传输示例。	为用户描述了泛型 USB 用法。
	在“配置”对话框的“高级”选项卡中添加了 extern_cls 和 extern_vnd 参数。	这些参数在解决方案级别使能其他组件，以提供它们对供货商和类请求本身的处理。
	向“带有手动存储器管理的 DMA”部分中添加了限制。	此限制说明如何正确使用模式 2/3 传输。
	修改了“高级”选项卡布局。	数据网格替换为带有每个参数信息的复选框。这样做是为了提高可用性。
	在“配置”对话框中添加了“音频描述符”选项卡。	这使您可以为您的组件添加和配置音频描述符。

	从开始/停止 API 删除了 SOF ISR 使能/禁用。	每 1ms 发生一次 SOF 中断，但是组件不使用此中断。如果应用程序需要此中断，可以通过调用以下函数使能它： CyIntEnable(USBFS_SOF_VECT_NUM);
1.30.b	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，该工具将报告错误/警告。如果发生此情况，请更新到支持您的目标器件的修订版。
1.30.a	已将本地参数移动到形式参数列表。	为了解决 PSoC Creator v1.0 Beta 4.1 和更早版本中存在的缺陷，更新了组件，以使它可以继续在该工具的较新版本中使用。此组件使用了本地参数（这些参数不向用户公开），以在用户输入时执行后台计算。这些参数已更改为可见的形式参数，但是不可编辑。组件没有功能上的更改，但是现在可在自定义程序对话框的“表达式视图”中看到受影响的参数。
1.30	更新了“配置”对话框和数据手册。	在“配置”对话框的“高级”选项卡中添加了“使能 SOF 输出”参数。 更新了数据手册中的 USBFS_ReadOutEP() 函数以反映正确的返回值。
1.20.b	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，该工具将报告错误/警告。如果发生此情况，请更新到支持您的目标器件的修订版。
1.20.a	已将本地参数移动到形式参数列表。	为了解决 PSoC Creator v1.0 Beta 4.1 和更早版本中存在的缺陷，更新了组件，以使它可以继续在该工具的较新版本中使用。此组件使用了本地参数（这些参数不向用户公开），以在用户输入时执行后台计算。这些参数已更改为可见的形式参数，但是不可编辑。组件没有功能上的更改，但是现在可在自定义程序对话框的“表达式视图”中看到受影响的参数。
1.10.b	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，该工具将报告错误/警告。如果发生此情况，请更新到支持您的目标器件的修订版。
1.10.a	已将本地参数移动到形式参数列表。	为了解决 PSoC Creator v1.0 Beta 4.1 和更早版本中存在的缺陷，更新了组件，以使它可以继续在该工具的较新版本中使用。此组件使用了本地参数（这些参数不向用户公开），以在用户输入时执行后台计算。这些参数已更改为可见的形式参数，但是不可编辑。组件没有功能上的更改，但是现在可在自定义程序对话框的“表达式视图”中看到受影响的参数。

© 赛普拉斯半导体公司，2011。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组件。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组件。若将赛普拉斯产品用于生命支持系统应用中，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。