EZ-USB™ FX3 入门

关于本文档

范围及目的

AN75705 文档介绍的是 EZ-USB™ FX3 USB 3.0 器件控制器的入门信息。本文档重点介绍了 EZ-USB™ FX3 的技术参数和应用程序，并且提供了有关各种设计资源的指引，以帮助 FX3 的开发。本文档还引导您使用具有 Eclipse 集成开发环境 (IDE) 特性的英飞凌免费工具，开始开发 EZ-USB™ FX3 固件。最后，它还对 EZ-USB™ FX3 软件以及硬件开发套件进行了概述。

相关器件系列

EZ-USB™ FX3

软件版本

SDK 1.3.4

相关应用笔记

AN70707

要想获取 EZ-USB™ FX3 设计资源的完整列表，请**点击此处**。

更多代码示例？我们明白。

要想获取 USB 超高速代码示例的完整列表，请参考 **http://www.cypress.com/101781**。

目录

关于本文档 **1**
目录 **1**
1 简介 **4**
2 相关资源 **5**
2.1 EZ-USB™ FX3 软件开发套件 6
2.2 GPIF II Designer 6
3 EZ-USB™ FX3 是什么？ **7**
4 EZ-USB™ FX3 特性 **8**
4.1 USB 接口 8
4.2 GPIF II 8
4.3 CPU 8
4.4 JTAG 接口 9
4.5 UART 接口 9
4.6 I^2C 接口 9
4.7 I^2S 接口 9
4.8 SPI 接口 9
4.9 启动选项 9

目录

4.10 时钟 9
4.11 电压域 9
5 EZ-USB™ FX3 的应用开发 10
5.1 FPGA/ASIC 连接到 EZ-USB™ FX3 10
5.2 通过 EZ-USB™ FX3 配置 FPGA 10
5.3 EZ-USB™ FX3 的图像传感器接口 11
5.4 设计 EZ-USB™ FX3 硬件 11
5.5 将 EZ-USB™ FX2LP 设计升级为 EZ-USB™ FX3 11
6 设计资源 12
7 EZ-USB™ FX3 术语 13
8 使用 EZ-USB™ FX3 进行初次 USB 3.0 传输 18
8.1.1 构建并运行 USBBulkLoopAuto 固件 23
8.1.2 将 USBBulkLoopAuto 固件加载到 EZ-USB™ FX3 RAM 内 25
8.1.3 执行 BULK 传输 26
9 开发工具 28
9.1 SuperSpeed Explorer 套件简介 28
9.2 EZ-USB™ FX3 SDK 简介 29
9.2.1 固件库与 API 29
9.2.2 框架 API 30
9.3 EZ-USB™ FX3 固件示例 30
9.3.1 USB BULK 数据回送示例 30
9.3.2 USB ISOCHRONOUS 数据回送示例 31
9.3.3 从 FIFO 应用示例 31
9.3.4 串行接口示例 31
9.3.5 USB BULK/ISOCHRONOUS 数据源库示例 31
9.3.6 存储器编程器示例 31
9.3.7 大容量存储类示例 31
9.3.8 USB 音频类示例 31
9.3.9 双阶段引导器示例 31
9.3.10 USB 主机与 OTG 示例 31
9.4 EZ-USB™ FX3 固件开发工具 31
9.4.1 Eclipse IDE 31
9.4.2 GNU 工具链 32
9.4.3 GPIF II Designer (设计程序) 32
9.4.4 集成调试器 32
9.5 Windows 软件概述 32
9.5.1 Windows USB 设备驱动程序 32
9.5.2 特性 33
9.6 应用接口 33
9.6.1 CyAPI.lib 33
9.6.2 CyUSB.dll 33
9.7 Windows 软件示例 34
9.7.1 BULKLoop 示例 34
9.8 Streamer 示例 34
9.9 适用于 Linux 的 EZ-USB™ FX3 SDK 和软件 35
9.9.1 适用于 Linux 的 EZ-USB™ FX3 SDK 35
9.9.2 用于 Linux 软件的 CyUSB Suite 35
9.10 有用的调试工具 36
9.10.1 USB 3.0 协议分析仪 36
9.10.2 逻辑分析仪 36

目录

10 附录 A. USB 3.0 概述 37
10.1 电气接口 37
10.2 线缆与连接 38
10.3 USB 3.0 与 2.0 39
11 附录 B. 在 Windows 上安装 EZ-USB™ FX3 DVK 驱动程序 41
12 附录 C. EZ-USB™ FX3 DVK 简介 43
12.1 EZ-USB™ FX3 DVK 的 JTAG 调试器 44
文档修订记录 45

1 简介

EZ-USB™ FX3 (后面缩写为 EZ-USB™ FX3) 是一个功能强大的 USB 3.0 外设控制器，可提供集成和灵活的功能。通过 EZ-USB™ FX3，开发者能够将 USB 3.0 功能集成到他们的系统中。

AN75705 文档有助于您了解 EZ-USB™ FX3。它重点介绍了 EZ-USB™ FX3 的关键用途、应用及功能。同时提供了可从英飞凌获取的设计资源的完整列表。本应用笔记介绍了使用 SuperSpeed Explorer 套件 (CYUSB3KIT-003) 进行 USB 传输的流程。

附录 A 提供了有关 USB 3.0 的概述。**附录 B** 介绍了如何安装 EZ-USB™ FX3 Windows 驱动程序 (如果尚未安装)。有关 EZ-USB™ FX3 DVK(CYUSB3KIT-001)的相关信息，请参考 **12**。

2 相关资源

英飞凌网站 (**www.infineon.com**) 上提供了大量资料，有助于为您的设计正确选择所需的 EZ-USB™ FX3 器件，并能够快速有效地将器件集成到设计中。培训视频：**EZ-USB™ FX3 架构概述** (可在本应用笔记的同一个网页上找到)

- 知识库文章：**EZ-USB™ FX3 知识库文章**
- 要了解 EZ-USB™ FX3 的基础知识并加快您的设计，请参阅 John Hyde 撰写的**超高速设备设计示例**文档。
- 概况：**USB 产品系列**、**USB 路线图**
- USB 3.0 产品选择器：**FX3**、**FX3S**、**CX3**、**HX3**
- 应用笔记：英飞凌提供了大量的 USB 应用笔记，包括了从基本到高级的广泛主题。下面列出的是 EZ-USB™ FX3 入门的应用笔记：
  - **AN75705 – EZ-USB™ FX3 入门**
  - **AN76405 – EZ-USB™ FX3 启动选项**
  - **AN70707 – EZ-USB™ FX3/FX3S 硬件设计指南和原理图检查表**
  - **AN65974 – 使用 EZ-USB™ FX3 从设备 FIFO 接口进行设计**
  - **AN75779 – 如何使用 EZ-USB™ FX3 在 USB 视频类别 (UVC) 框架内实现图像传感器连接**
  - **AN86947 – 使用 EZ-USB™ FX3 优化 USB 3.0 的吞吐量**
  - **AN84868 – 使用 EZ-USB™ FX3 通过 USB 配置 FPGA**
  - **AN68829 – 用于 EZ-USB™ FX3 的从设备 FIFO 接口：5 位地址模式**
  - **AN73609 – 在 Linux 上的 EZ-USB™ FX2LP/ FX3 开发 Bulk-Loop 示例**
  - **AN77960 – EZ-USB™ FX3 高速 USB 主机控制器简介**
  - **AN76348 – EZ-USB™ FX2LP 和 EZ-USB™ FX3 应用的区别**
  - **AN89661 – 使用 EZ-USB™ FX3S 设计 USB RAID 1 磁盘**
  - **AN90369 – 如何将 MIPI CSI-2 图像传感器连接至 EZ-USB™ CX3**
- 代码示例：
  - **USB 高速**
  - **USB 全速**
  - **USB 超速**
- 技术参考手册 (TRM)：
  - **EZ-USB™ FX3 技术参考手册**
- 开发套件：
  - **CYUSB3KIT-003 EZ-USB™ FX3 超高速浏览器套件**
- 模型：**IBIS**

相关资源

2.1 EZ-USB™ FX3 软件开发套件

英飞凌为 EZ-USB™ FX3 提供了完整的软件和固件堆栈，这样很容易便能够将超高速 USB 集成到嵌入式应用内。**软件开发套件** (SDK) 中带有各种工具、驱动程序和应用示例，有助于加快应用开发程序。

Table 1 EZ-USB™ FX3 设计资源

设计	资源	资源所在位置
EZ-USB™ FX3 固件	带有 GCC 编译器的 Eclipse IDE 安装包	可从 **EZ-USB™ FX3 SDK** 的安装包中获取
	API 用于执行多种函数	
	固件示例	
	有关使用 SDK 的文档	**EZ-USB™ FX3 编程员手册**
	SDK 提供的所有 API 文档	**EZ-USB™ FX3 SDK API 指南**
主机软件	USB 3.0 驱动程序 — *cyusb3.sys*	可从 **EZ-USB™ FX3 SDK** 安装包获取
	主机应用示例 — Control Center 和 Streamer 的应用	
	USBSuite 应用开发 — 快速入门指南	**USBSuite 应用开发 — 快速入门指南**
GPIF II 接口设计	通过 GPIF II Designer 工具，您可以使用状态机来设计一个自定义的 GPIF II 接口。该工具会生成集成到 EZ-USB™ FX3 固件内的所需代码。	可从 **EZ-USB™ FX3 SDK** 的安装包中获取
	GPIF II 通常用于实现从设备 FIFO、SRAM 从设备以及 ADMux 从设备接口	GPIF II Designer 工具 — 可从 **EZ-USB™ FX3 SDK** 的安装包中获取
	GPIF II 文档包含了工具的使用说明	GPIFII 设计指南 — GPIF II Designer 工具附带 (可从 **EZ-USB™ FX3 SDK** 的安装包中获取)
固件调试	设置并使用 JTAG 调试器	**EZ-USB™ FX3 编程者手册**中的第 12 章

2.2 GPIF II Designer

GPIF II Designer 是一个图形软件，设计师可以通过它来配置 EZ-USB™ FX3 USB 3.0 器件控制器的 GPIF II 接口。

用户使用该工具可以从英飞凌所提供的五个接口选择一个，或从头创建专有的 GPIF II 接口。英飞凌提供了符合工业标准的接口，如异步和同步从设备 FIFO、异步和同步 SRAM 以及异步 SRAM。在具有上面所述接口的系统中，开发者可从一组标准参数 (如总线宽度 (x8、16、x32)、字节顺序、时钟设置) 选择所需要的接口。然后，编译已选的接口。该工具为需要自定义接口的用户提供了一个简洁的三步骤 GPIF 接口开发程序。用户先选择引脚配置和标准参数。然后，他们可以使用可配置操作设计一个虚拟的状态机。最后，用户通过查看输出时序验证是否与所需时序相匹配。一旦完成这三个步骤，便可以使用 EZ-USB™ FX3 对该接口进行编译和集成操作。

3 EZ-USB™ FX3 是什么?

EZ-USB™ FX3 是一个集成了 Arm® Arm9 处理器的 USB 3.0 外设控制器。并行和串行接口提供了与系统中其他设备的高速连接。

系统中 EZ-USB™ FX3 器件的主要功能是传输 USB 主机与外设 (比如摄像头或扫描仪) 之间的高带宽数据。强大的片上 Arm9 处理器使 EZ-USB™ FX3 能够访问数据流，并能高效地处理数据。在不要求 EZ-USB™ FX3 执行数据处理的系统中，Arm9 固件仅初始化和管理 USB 和数据接收/发送设备两个接口间的数据传输。

除了 I2C、SPI、UART 和 I2S 串行接口之外，EZ-USB™ FX3 还有一个高度灵活的可编程接口，即第二代通用可编程接口 (GPIF II)。通过可编程的 GPIF II，EZ-USB™ FX3 可以同各种设备 (包括 FPGA、图像传感器、ADC 以及应用处理器) 相连接。因此，在多种高性能 USB 3.0 应用中，EZ-USB™ FX3 是一个优秀的控制器选择。

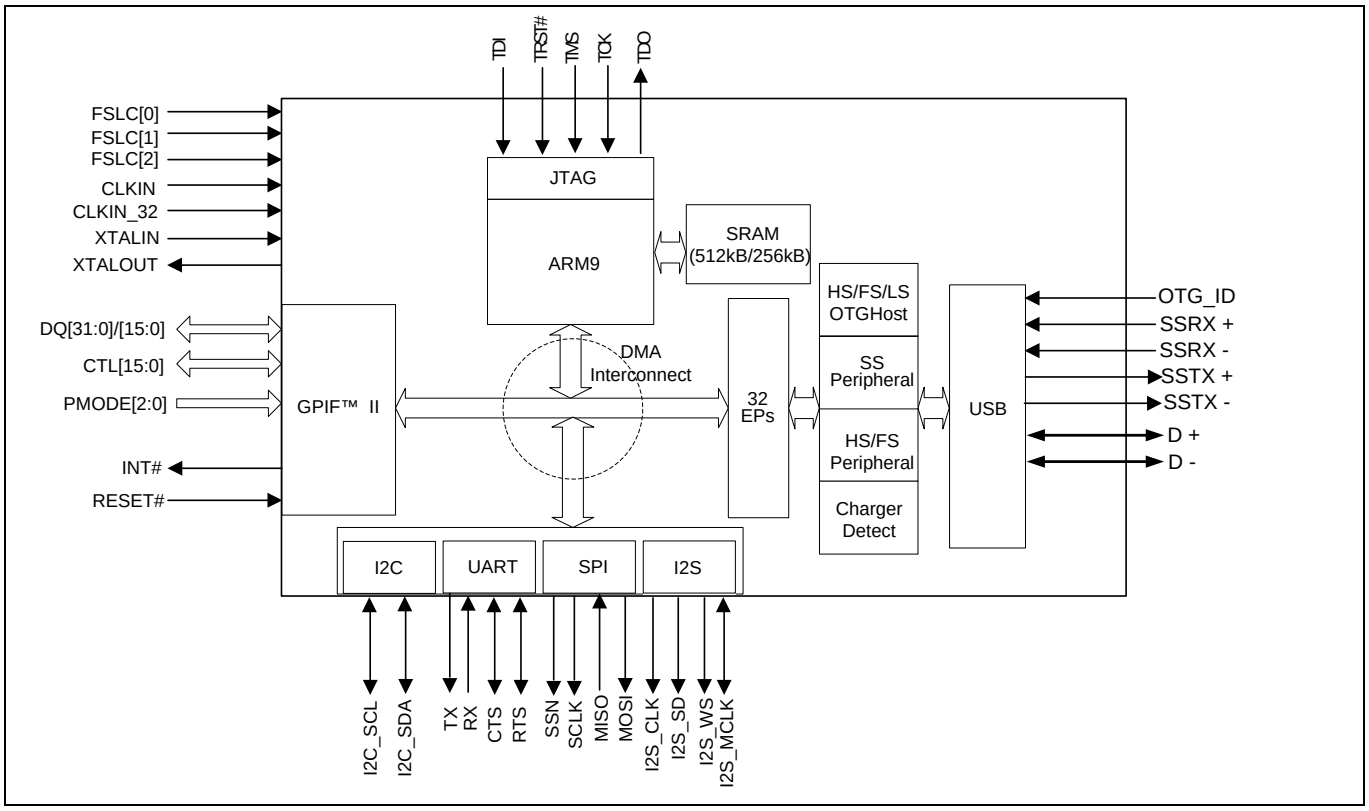


Figure 1 EZ-USB™ FX3 框图

4 EZ-USB™ FX3 特性

该部分对 EZ-USB™ FX3 的主要特性进行简要的介绍。

4.1 USB 接口

EZ-USB™ FX3 的 USB 接口支持下面各特性：

- USB SuperSpeed (超高速) 和 Hi-Speed (高速) 外设功能，符合 USB 3.0 规范 (版本 1.0)。符合本规范的设备 (如 EZ-USB™ FX3) 都与 USB 2.0 规范向后兼容。
- 符合 OTG 补充标准，版本 2.0。EZ-USB™ FX3 支持高速、全速和低速 OTG 双角色设备功能。作为一个外设，EZ-USB™ FX3 能够实现超高速、高速以及全速的传输功能。作为主机，它能够实现高速、全速以及低速传输功能。
- 按照 CEA-936A 规范，在 USB D+/D–线上支持 Carkit Pass-through UART 功能。
- 支持多达 16 个输入端点和 16 个输出端点。

4.2 GPIF II

高性能的 GPIF II 接口 (作为处理器接口模块 (PIB) 的一部分) 与 EZ-USB™ FX2LP 的 GPIF 和从设备 FIFO 接口的功能相似，但其功能更为先进。GPIF II 是一种可编程的状态机，它所启用的灵活接口使用自己的高速时钟，完全独立于 Arm9。在工业标准或专用接口中，GPIF II 能够作为一个主设备或从设备运行。GPIF II 可支持并行和串行接口。

GPIF II 的主要特性包括：

- 可作为主器件或从器件使用。
- 提供 256 种可编程状态。
- 支持 8 位、16 位、24 位和 32 位的并行数据总线。
- 支持频率高达 100 MHz 的接口。
- 使用 32 位数据总线时，支持 14 个可配置的 I/O 引脚 (作为控制信号使用)。控制引脚可作为输入、输出或双向引脚使用。
- 使用 16/8 位数据总线时，支持 16 个控制 I/O 引脚。控制引脚可作为输入、输出或双向引脚使用。

英飞凌的 GPIF II Designer 工具有助于快速开发 GPIF II 状态机，并且包含了常用接口的示例。安装 **EZ-USB™ FX3 SDK** 时，可以使用 GPIF II Designer 工具。

GPIF II 一个常见的应用是作为同步从设备 FIFO 接口，用于很多和 FPGA 的连接中。有关同步从设备 FIFO 接口的详细信息，请参考 **AN65974 – 设计 EZ-USB™ FX3 从设备 FIFO 接口**中的内容。

4.3 CPU

EZ-USB™ FX3 拥有一个片上 32 位 200 MHz ARM926EJ-S 内核 CPU。该内核能直接访问 16 KB 的指令紧耦合存储器 (TCM) 和 8 KB 的数据紧耦合存储器。ARM926EJ-S 内核还提供了用于固件调试的 JTAG 接口。

EZ-USB™ FX3 集成了 512 KB 或 256 KB 的嵌入式 SRAM (取决于所选择的设备型号)，并支持四种启动代码的方法 (USB、GPIF II、I2C 或者 SPI)。

EZ-USB™ FX3 能在多种外设 (如 USB、GPIF II、I2S、SPI 以及 UART) 间实现高效灵活的 DMA 连接。EZ-USB™ FX3 固件配置外设间的数据访问后，DMA 结构将管理传输，而不需要 Arm9 内核参与。安装 **EZ-USB™ FX3 SDK** 后可以使用 EZ-USB™ FX3 示例固件。

4.4 JTAG 接口

EZ-USB™ FX3 的 JTAG 接口包含一个标准的 5 引脚接口，用于连接 JTAG 调试器，以便通过 CPU 内核的片上调试电路来调试固件。ARM926EJ-S 内核的业界标准调试工具可用于 EZ-USB™ FX3 应用开发。

4.5 UART 接口

EZ-USB™ FX3 的 UART 接口支持全双工通信，包括 TX、RX、CTS 以及 RTS 信号。UART 支持各种波特率，从 300 bps 到 4608 Kbps，可通过 EZ-USB™ FX3 固件进行选择。

4.6 I^2C 接口

EZ-USB™ FX3 的 I2C 接口作为 I2C 主设备使用，同 I2C 从设备进行通信。例如，EZ-USB™ FX3 可从连接至 I2C 接口的 EEPROM 引导。EZ-USB™ FX3 的 I2C 主控制器支持多主控制功能和 I2C 时钟延展。I2C 控制器所支持的总线频率有 100 KHz、 400 KHz 和 1 MHz。

4.7 I^2S 接口

EZ-USB™ FX3 拥有一个 I2S 发送器，用于支持外部音频编解码器和其它 I2S 接收器。I2S 接口支持的采样频率有 32kHz、44.1 kHz 和 48 kHz。

4.8 SPI 接口

EZ-USB™ FX3 支持作为 SPI 主设备接口，其最大工作频率为 33 MHz。SPI 控制器支持四种 SPI 通信模式，传输数据大小为 4 位到 32 位。

4.9 启动选项

EZ-USB™ FX3 能够从下面各源中加载启动程序：

USB、I2C、SPI、GPIF II (Booloader 所支持的同步 ADMux、异步 ADMux 或异步 SRAM 接口)。

AN76405 — EZ-USB™ FX3 引导选项包含决定相应引导方法 (用于您的设计) 的详细信息。

4.10 时钟

EZ-USB™ FX3 支持晶振或外部时钟连接。支持的晶振频率为 19.2 MHz，支持的外部时钟频率为 19.2、26、38.4 和 52 MHz。

EZ-USB™ FX3 的时钟输入必须符合 **EZ-USB™ FX3 数据手册**中具体规定的相位噪声和时序抖动要求。

4.11 电压域

EZ-USB™ FX3 各功能模块 (GPIF II、UART/SPI/I2S、I2C、JTAG、USB、时钟和内核) 具有独立的电压域。

EZ-USB™ FX3 的输入电压必须满足 **EZ-USB™ FX3 数据手册**中的具体规定。

进行设计 EZ-USB™ FX3 硬件时，请参考 **AN70707 — EZ-USB™ FX3/FX3S 硬件设计指南和原理图检查表**中介绍的内容。

5 EZ-USB™ FX3 的应用开发

本章节重点介绍了 EZ-USB™ FX3 在系统中的典型应用。

一个完整的 EZ-USB™ FX3 设计包括软件、固件以及硬件的开发。英飞凌为这些内容提供了设计资源。EZ-USB™ FX3 设计资源在 **Table 1** 中列出。

5.1 FPGA/ASIC 连接到 EZ-USB™ FX3

通过高速并行 GPIF II 接口，某些应用可以将 FPGA 或 ASIC 连接到 EZ-USB™ FX3。作为数据的源端或汇端的其它设备则被连接到 FPGA 或 ASIC。这样的应用包括数据采集设备、打印机、扫描仪和成像设备等等。

在上述应用中，EZ-USB™ FX3 作为与 USB 主机进行数据发送和接收的快速数据管道。对于这些应用，EZ-USB™ FX3 到 FPGA/ASIC 的接口通常是同步从设备 FIFO 接口 (**Figure 2**)。

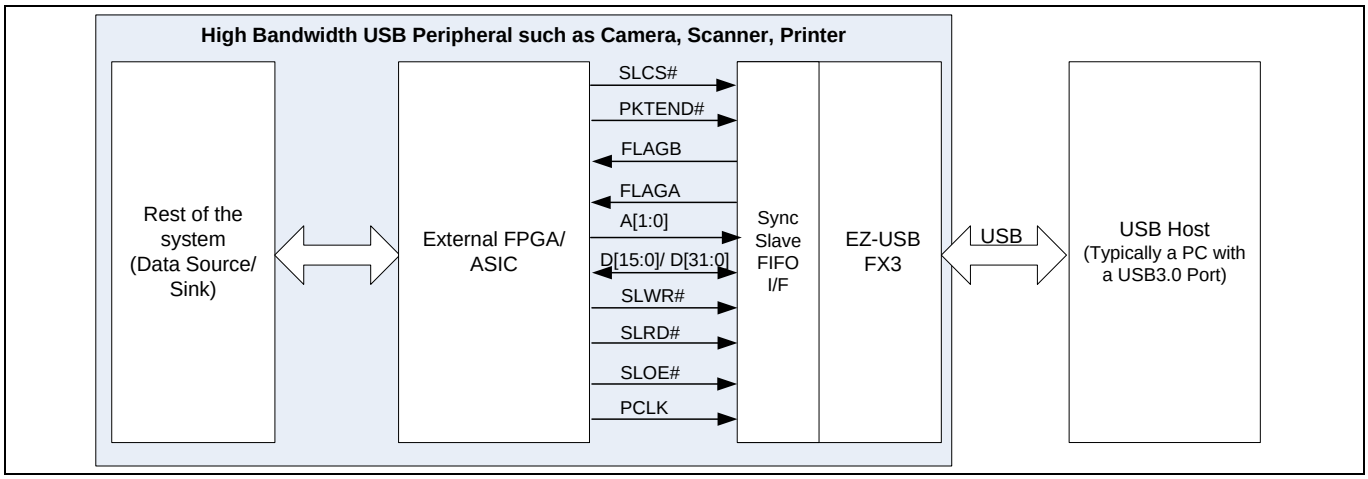


Figure 2 将 FPGA/ASIC 连接到 EZ-USB™ FX3

有关从设备 FIFO 接口以及 FPGA 连接到 EZ-USB™ FX3 的完整设计示例的详细说明，请参阅 **AN65974 — 设计 EZ-USB™ FX3 从设备 FIFO 接口**。

GPIF II 能够实现的另一种接口是同步地址数据复用接口。

5.2 通过 EZ-USB™ FX3 配置 FPGA

在 EZ-USB™ FX3 连接到 FPGA 的应用中，可能会先通过 EZ-USB™ FX3 加载 FPGA bit 文件。此时，该 FPGA 配置文件可以通过 EZ-USB™ FX3 从一个 USB 主机传送到 FPGA。这样可以节省一个 FPGA 启动存储器，并且能够通过 PC 对 FPGA 进行重新配置 (包括更新操作)。有关示例的实现，请参见 **AN84868 — 使用 EZ-USB™ FX3 配置 FPGA**。

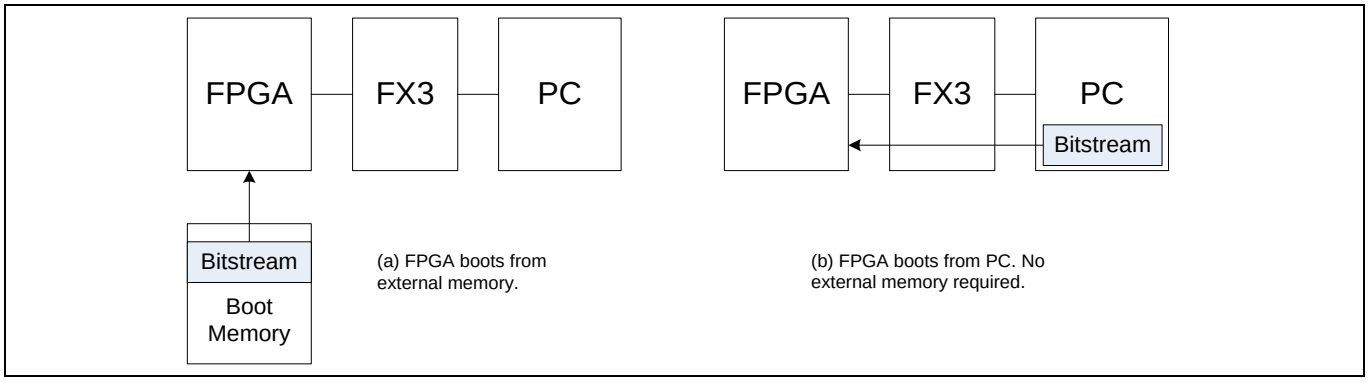


Figure 3 通过 EZ-USB™ FX3 配置 FPGA

5.3 EZ-USB™ FX3 的图像传感器接口

在图像应用中，图像传感器通过并行 GPIF II 接口直接与 EZ-USB™ FX3 相连，视频流通过 EZ-USB™ FX3 从传感器发送给 USB 主机。

通过使用合适的固件对 EZ-USB™ FX3 进行编程，可使该系统同 USB 视频类别相兼容。有关这种应用的示例，请参见 **AN75779 — 如何将图像传感器连接到 USB 视频类别 (UVC) 框架中的 EZ-USB™ FX3**。

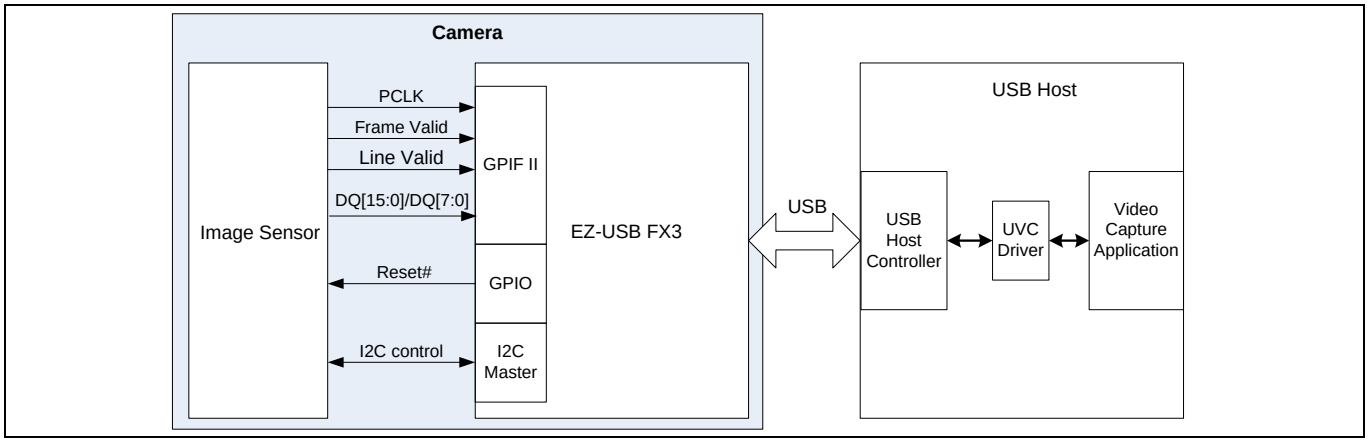


Figure 4 EZ-USB™ FX3 的图像传感器接口

5.4 设计 EZ-USB™ FX3 硬件

由于 USB 3.0 的速度和电源的特点，进行电脑上的设计和布局时请务必小心。设计 EZ-USB™ FX3 硬件时，请参考 **AN70707 — EZ-USB™ FX3/FX3S 硬件设计指南和原理图检查表**中介绍的内容。有关原理图的示例设计，请参见 **EZ-USB™ FX3 DVK 原理图**。

5.5 将 EZ-USB™ FX2LP 设计升级为 EZ-USB™ FX3

如果您想要将现有的 EZ-USB™ FX2LP 设计升级到 EZ-USB™ FX3，请参见 **AN76348 — EZ-USB™ FX2LP 和 EZ-USB™ FX3 应用的区别**中介绍的内容。

6 设计资源

Table 1 列出了英飞凌所提供的各种资源，用于开发 EZ-USB™ FX3。有关 USB SuperSpeed 代码示例的完整列表，请点击**此处**。

7 EZ-USB™ FX3 术语

要想了解 EZ-USB™ FX3 的数据输出和输入，需要明白下面的重要术语：

- 套接字
- DMA 描述符
- DMA 缓冲区
- GPIF 线程

套接字是外设硬件模块和 EZ-USB™ FX3 RAM 之间的连接点。EZ-USB™ FX3 上的每个外设硬件模块 (如 USB、GPIF、UART 和 SPI) 具有各自固定的套接字数量。流经外设的独立数据流数量等于该外设上的套接字数量。套接字的实现包括一组寄存器，用于指向有效的 DMA 描述符，并使能或置位与该套接字相关的中断。

DMA 描述符是一组位于 EZ-USB™ FX3 RAM 中的寄存器。它保存了 DMA 缓冲区的地址和大小，以及指向下一个 DMA 描述符的指针。这些指针构建成了 DMA 描述符链。

DMA 缓冲区是 RAM 的一部分，用于存储通过 EZ-USB™ FX3 设备传输的中间数据。通过 EZ-USB™ FX3 固件，可将部分 RAM 空间分配给 DMA 缓冲区。这些缓冲区的地址被存储在 DMA 描述符中。

GPIF 线程是 GPIF II 模块内的专用数据路径，用于将外部数据引脚同套接字连接起来。

套接字可通过各个事件来互相发出信号，或者它们可通过中断向 EZ-USB™ FX3 CPU 发出信号。该操作由固件配置的。例如，将数据流从 GPIF II 模块传输给 USB 模块。GPIF 套接字可以通知 USB 套接字它已经向 DMA 缓冲区写满了数据，或者 USB 套接字可以通知 GPIF 套接字该 DMA 缓冲区目前为空。该操作被称为自动 DMA 通道。当 EZ-USB™ FX3 CPU 不用修改数据流中的任何数据时，通常会使用自动 DMA 通道。

或者，GPIF 套接字可以向 EZ-USB™ FX3 CPU 发送一个中断，来通知 GPIF 套接字已经写满了 DMA 缓冲区。EZ-USB™ FX3 CPU 可将该信息传递给 USB 套接字。USB 套接字会向 EZ-USB™ FX3 CPU 发送一个中断，来通知 USB 套接字已经读空了 DMA 缓冲区。此时，EZ-USB™ FX3 CPU 可将该信息反馈给 GPIF 套接字。该操作被称为手动 DMA 通道实现。EZ-USB™ FX3 CPU 需要添加、删除或修改数据流中的数据时，通常需要执行该操作。**UVC 应用笔记 (AN75779)** 中的固件示例使用了手动 DMA 通道操作，因为该固件要求添加 UVC 视频数据首部。

将数据写入 DMA 缓冲区内的套接字被称作发送套接字。从 DMA 缓冲区内读取数据的套接字被称作接收套接字。套接字使用存储在 DMA 描述符上的 DMA 缓冲区地址、DMA 缓冲区大小和 DMA 描述符链的值来管理数据。套接字写满或读空 DMA 缓冲区后，需要经过一段时间 (几微秒) 从一个 DMA 描述符转移到另一个描述符。转移过程中，套接字不能传输数据。对于没有流量控制的接口，该延迟是一个问题。图像传感器接口便属于这类情况。

通过使用 GPIF II 模块中的多个 GPIF 线程，可以解决该问题。GPIF II 模块提供四个 GPIF 线程。每次只有一个 GPIF 线程能够传输数据。GPIF II 状态机必须选择一个有效的 GPIF 线程来传输数据。

GPIF 线程选择机制同复用器一样。GPIF II 状态机使用内部控制信号或外部输入 (在**从设备 FIFO 2 位应用笔记**中便为地址线 A1 和 A0) 来选择有效的 GPIF 线程。切换有效的 GPIF 线程时会切换用于数据传输的有效套接字，从而改变用于数据传输的 DMA 缓冲区。GPIF 线程切换没有延迟。

套接字与 GPIF 线程的默认映射如 **Figure 5** 所示 – 套接字 0 对应于 GPIF 线程 0、套接字 1 对应于 GPIF 线程 1、套接字 2 对应于 GPIF 线程 2，以及套接字 3 对应于 GPIF 线程 3。

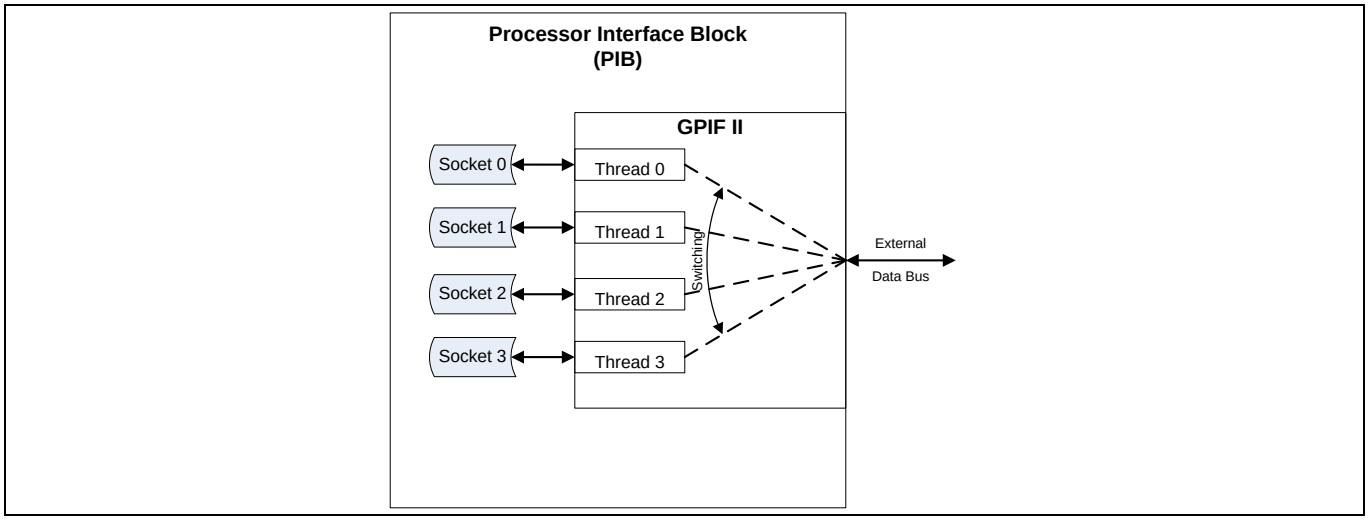


Figure 5　GPIF II 套接字/线程的默认映射图

为了便于用户理解 DMA 传输，下面四个图中说明了套接字的概念。

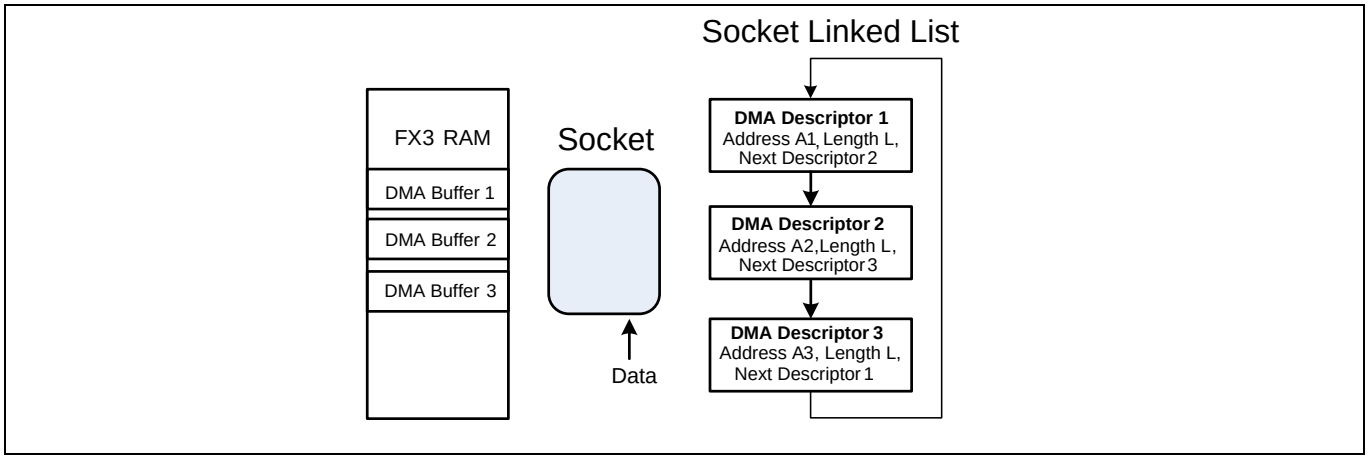


Figure 6　套接字根据 DMA 描述符列表传输数据

套接字链接列表是主存储器中的一组数据结构，这组数据结构又被称为 DMA 描述符。每个描述符指定了 DMA 缓冲区的地址和长度，以及指向下一个 DMA 描述符的指针。套接字运行时，逐个地检索 DMA 描述符，按照描述符指定的地址和长度向 DMA 缓冲区中传输数据。传输 L 个字节后，该套接字会检索下一个描述符，并继续将数据传输到另一个 DMA 缓冲区内。

该结构使套接字变得非常灵活，因为可以在存储器中的任何位置上创建任何数量的 DMA 缓冲区，并且这些缓冲区可以自动被链接在一起。例如，**Figure 7** 中的套接字重复循环检索 DMA 描述符。

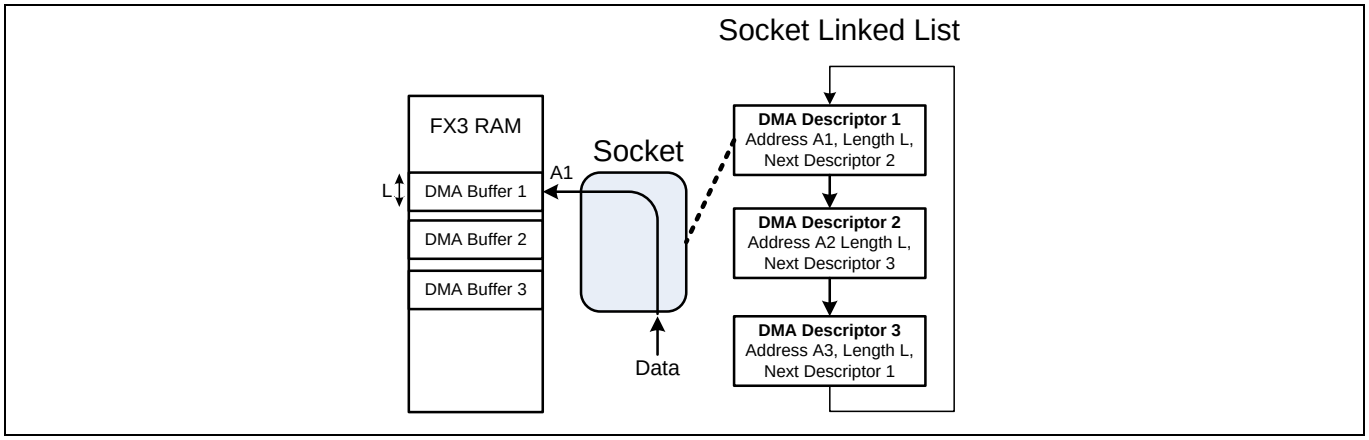


Figure 7　使用 DMA 描述符 1 运行的套接字

在 **Figure 7** 中，该套接字加载了 DMA 描述符 1，并从 A1 地址开始传输操作。在传输完 L 个字节之后，它将检索 DMA 描述符 2，继续从地址 A2 传输 L 个字节数据 (**Figure 8**)。

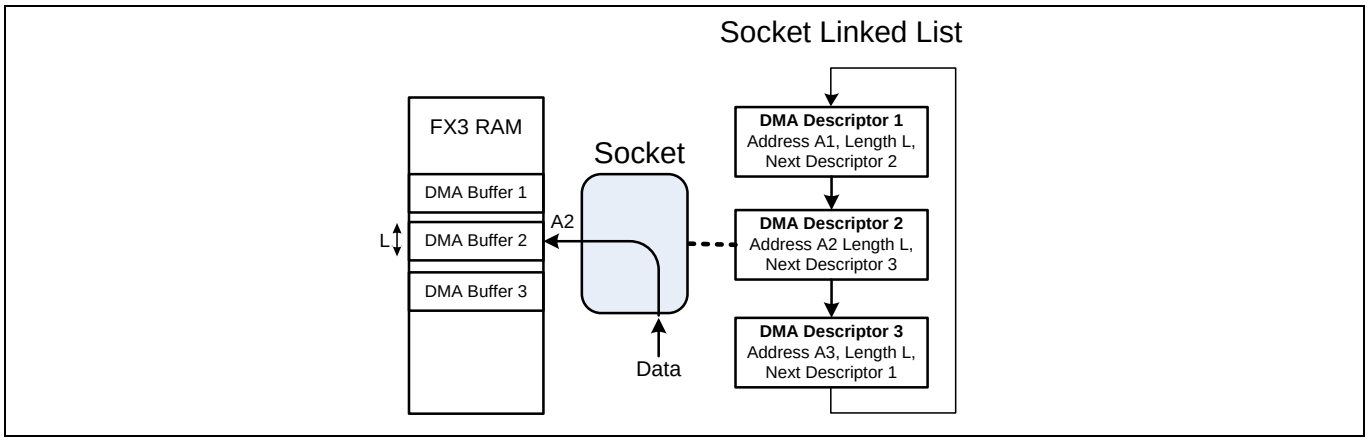


Figure 8　使用 DMA 描述符 2 运行的套接字

在 **Figure 9** 中，该套接字检索第三个 DMA 描述符，并从 A3 地址开始传输数据。L 个字节被传输后，将返回到 DMA 描述符 1 重复该操作。

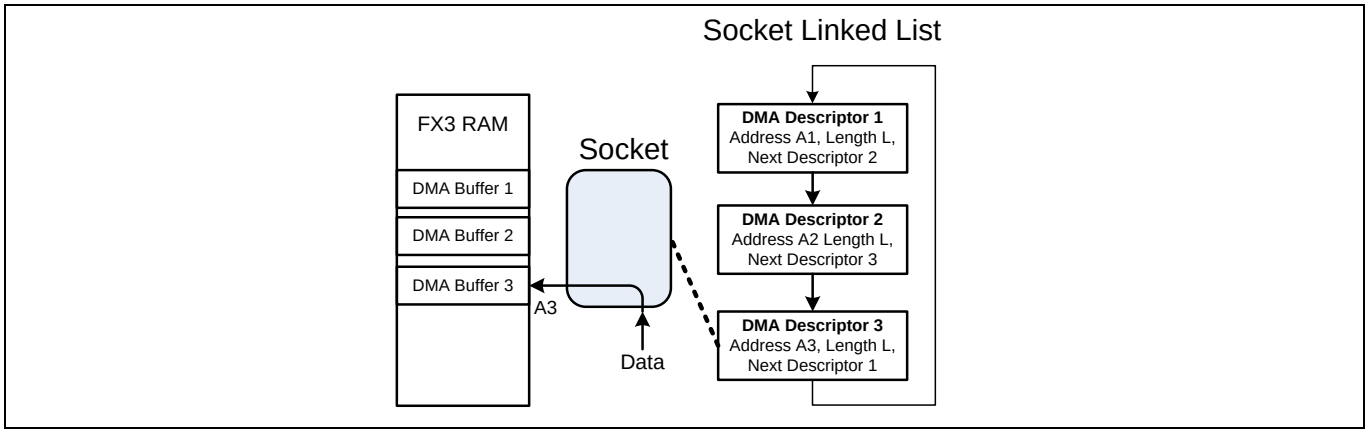


Figure 9　使用 DMA 描述符 3 运行的套接字

Figure 10 显示了 DMA 数据传输的详细过程。例如，示例应用使用了三个长度为 L 个字节的 DMA 缓冲区，这些缓冲区通过链接方式形成一个圆形循环。EZ-USB™ FX3 存储器地址位于左边。蓝色箭头表示该套接字从存储器加载套接字链接列表的描述符。红色箭头表示数据传输路径。下面各步骤描述了数据被发送到内部 DMA 缓冲区时所进行的套接字操作序列。

步骤 1：将存储器中的 DMA 描述符 1 加载到套接字内。获取有关 DMA 缓冲区地址 (A1)、DMA 缓冲区大小 (L) 和下一个描述符 (DMA 描述符 2) 的信息。转到步骤 2。

步骤 2：将数据传输到起始地址为 A1 的 DMA 缓冲区。向 DMA 缓冲区传送了 L 个字节后，将转到步骤 3。

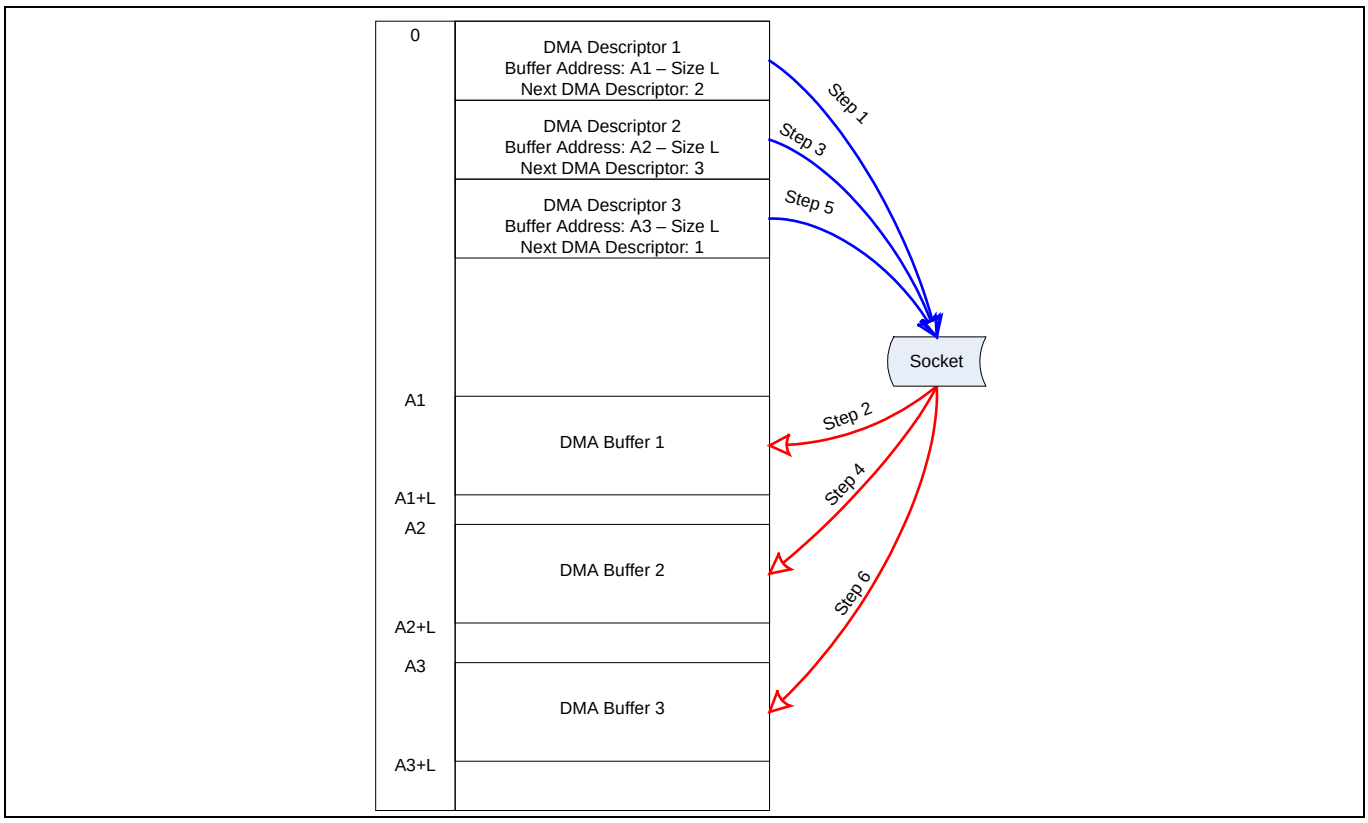


Figure 10　DMA 传输示例

步骤 3： 加载当前 DMA 描述符 1 所指向的 DMA 描述符 2。获取有关 DMA 缓冲区的地址 (A2)、DMA 缓冲区大小 (L) 以及下一个描述符 (DMA 描述符 3) 的信息。

步骤 4：将数据传输到起始地址为 A2 的 DMA 缓冲区。向 DMA 缓冲区传送了 L 个字节后，将转到步骤 5。

步骤 5：加载当前 DMA 描述符 2 所指向的 DMA 描述符 3。获取有关 DMA 缓冲区地址 (A3)、DMA 缓冲区大小 (L) 以及下一个描述符 (DMA 描述符 1) 的信息。

步骤 6：将数据传输到起始地址为 A3 的 DMA 缓冲区。向 DMA 缓冲区传送了 L 个字节后，将返回步骤 1。

如果在套接字从存储器检索下一个 DMA 描述符时 (通常需要 1 微秒)，外部外设进行数据传输，使用该简单的方案会使数据丢失。套接字可以在一个时钟周期内进行切换而不会生成延迟，可利用该特性来解决问题。这样，如果外部外设没有任何流控制机制，便可以使用两个套接字。Figure 11 显示了使用两个套接字进行的数据传输以及执行操作的各步骤。套接字 0 和套接字 1 对 DMA 缓冲区的访问分别使用了红色和绿色箭头 (套接字有各自的独立数据路径) 区分。每一步骤中的‘a’和‘b’操作同时发生。硬件执行的该并行操作可以消除 DMA 描述符的检索时滞，并允许 GPIF II 将数据流连续传输到内部存储器内。

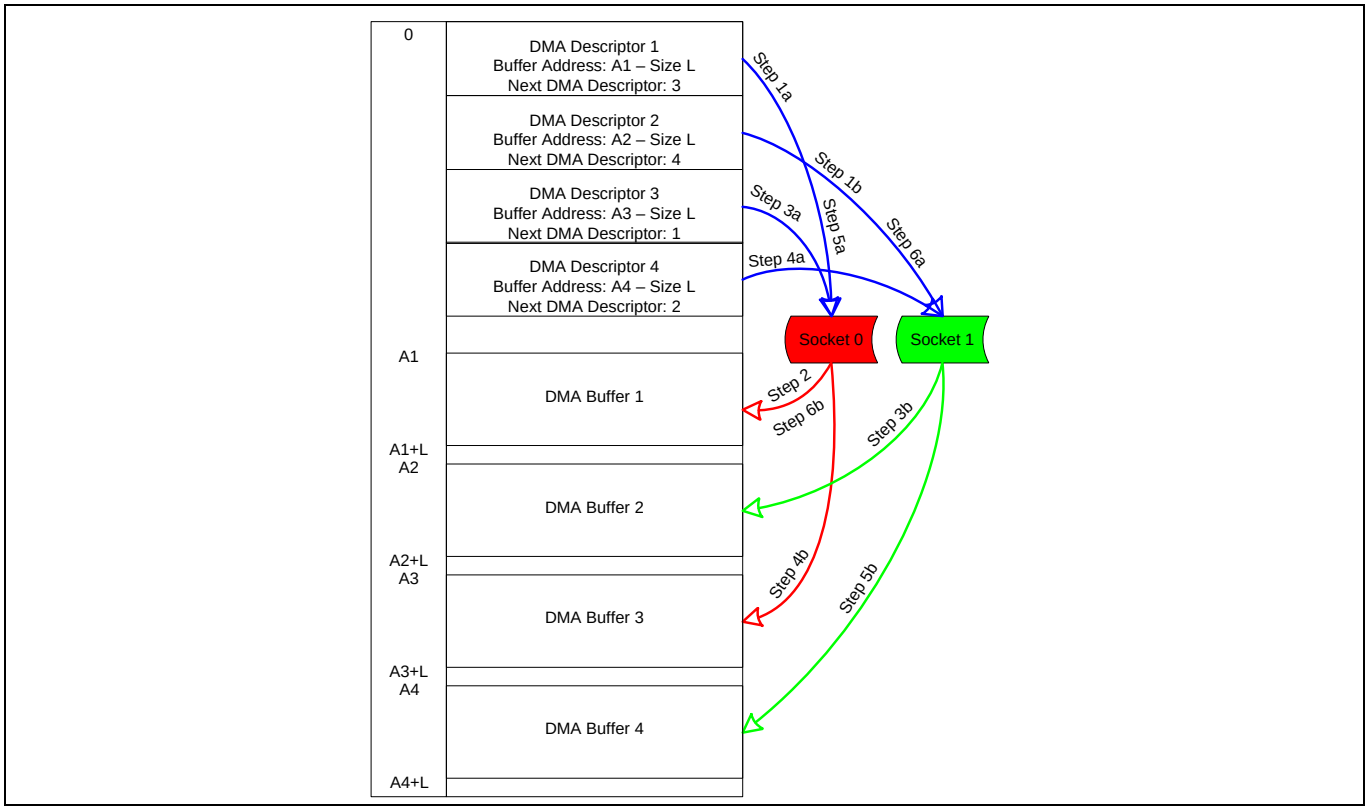


Figure 11　使用双套接字的无缝传输

步骤 1：启动各套接字时，套接字 0 和套接字 1 分别加载 DMA 描述符 1 和 DMA 描述符 2。

步骤 2：当有数据时，套接字 0 将数据传输到 DMA 缓冲区 1 内，该传输的长度为 L 个字节。传输结束后，继续转到步骤 3。

步骤 3：GPIF II 切换 GPIF 线程，从而切换进行数据传输的套接字。套接字 1 开始将数据传输到 DMA 缓冲区 2 内，同时套接字 0 将加载 DMA 描述符 3。在套接字 1 完成传输 L 个字节数据时，套接字 0 已准备好将数据传输到 DMA 缓冲区 3 内。

步骤 4：GPIF II 此时切换回到原始的 GPIF 线程。这时，套接字 0 将长度为 L 的数据传输到 DMA 缓冲区 3 内。同时，套接字 1 将加载 DMA 描述符 4，以便准备好将数据传输到 DMA 缓冲区 4 内。套接字 0 完成传输 L 个字节的数据后，会转到步骤 5。

步骤 5： GPIF II 将套接字 1 的数据传输到 DMA 缓冲区 4 内。同时，套接字 0 加载 DMA 描述符 1 以准备好将数据传输到 DMA 缓冲区 1 内。请注意，步骤 5a 与步骤 1a 相同，但它的套接字 1 不进行初始化，而是同时进行数据传输的。

步骤 6：GPIF II 再次切换套接字后，套接字 0 会将 L 个字节数据传输到 DMA 缓冲区 1 内。假设 UIB 接收套接字此时已经读空了 DMA 缓冲区中的所有数据。套接字 1 同时会加载 DMA 描述符 2 并准备将数据传输到 DMA 缓冲区 2 内。在此，继续转到步骤 3。

只在接收端 (USB) 读空并释放掉 DMA 缓冲区 (以接收 GPIF II 的下一个数据块) 时，GPIF II 套接字才能传输数据。如果接收端的速度不够块，对 DMA 缓冲区的写操作将被忽略，使套接字丢失数据。

8 使用 EZ-USB™ FX3 进行初次 USB 3.0 传输

本节向您介绍了建立并运行一个简单固件示例的步骤，从而您可以执行 USB 主机与 EZ-USB™ FX3 设备间的 BULK OUT 和 BULK IN 传输。

在执行该示例前，请实现以下操作：

1. 准备 **EZ-USB™ FX3 SuperSpeed Explorer 套件 (CYUSB3KIT-003)**。在该过程中，EZ-USB™ FX3 固件是从 USB 主机下载的，因此必须配置 EZ-USB™ FX3 DVK 开发板从 USB 启动。连接所有跳线器 (如 **Figure 12** 所示)，来选择 USB 启动。这些跳线器设置都使用一个红色框或圆圈加亮显示。有关该开发板的详细信息，请参考**套件用户指南**。SuperSpeed Explorer 套件电路板只能由 USB 线缆供电 (总线供电)。

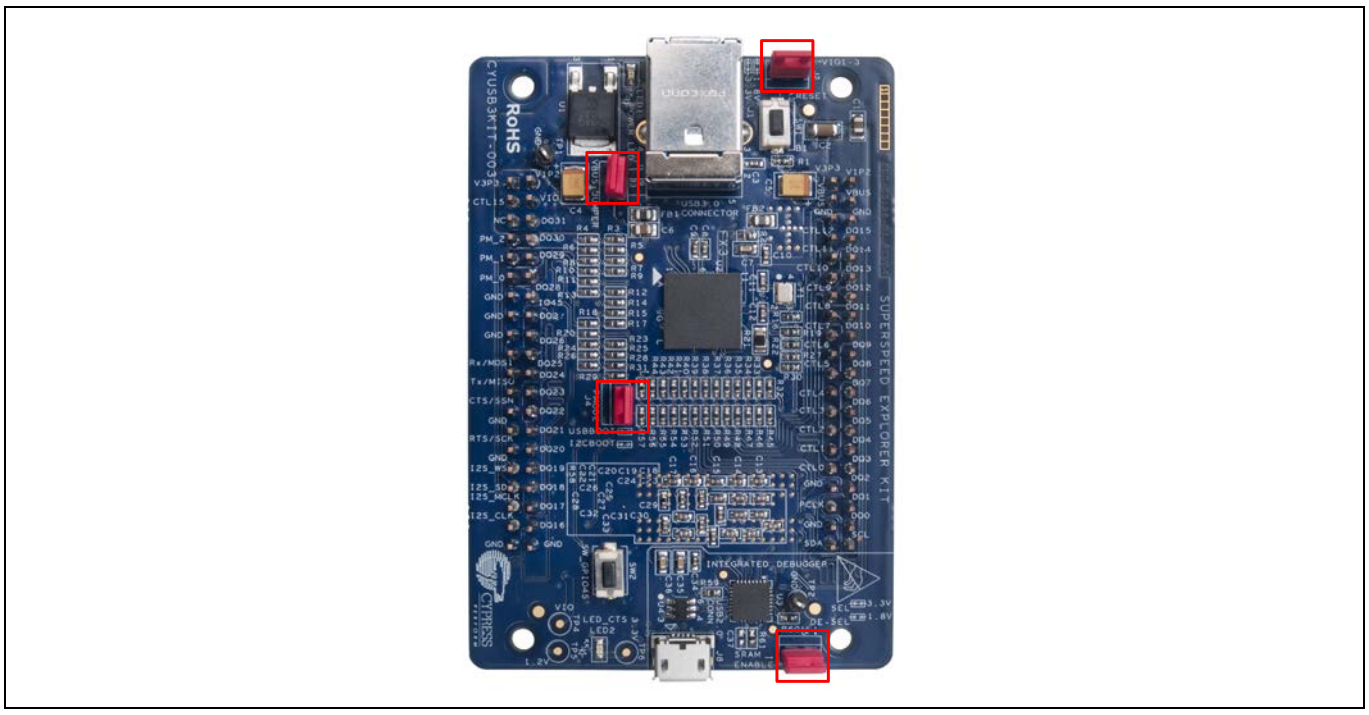


Figure 12 SuperSpeed Explorer 套件电路板的 USB 启动设置

2. 安装 **EZ-USB™ FX3 SDK**。当要求选择 **Installation Type** 项时，请选择默认的 **Typical** 项。安装 EZ-USB™ FX3 SDK 后，将固件示例导入到 Eclipse IDE 工作区内 (Eclipse IDE 安装包作为 SDK 的一部分)。想要将所有固件示例导入到 IDE 内，请按照下列步骤：

   a) 从 Start 菜单内打开 Eclipse IDE。依次选择 **All Programs** > **Cypress** > **Eclipse** > **EZ USB Suite**。接受默认的工作区文件夹，但后面您随时都可以定义自己的路径。

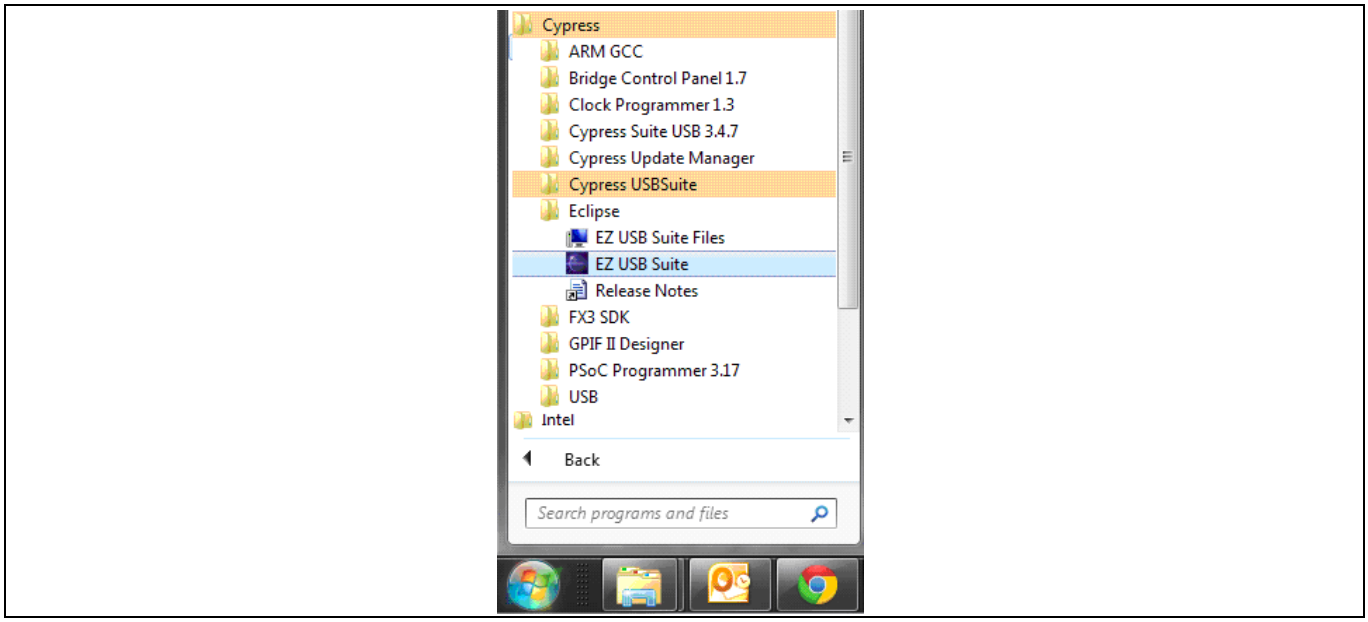


Figure 13　从“Start”菜单打开“Eclipse IDE”

b) 第一次打开 Eclipse IDE 时，您将被提示创建一个存储各 Eclipse 项目的 Workspace 文件夹，如 **Figure 14** 所示。请确保记下创建 Workspace 的位置。您编译的所有固件都位于 Workspace 文件夹中。

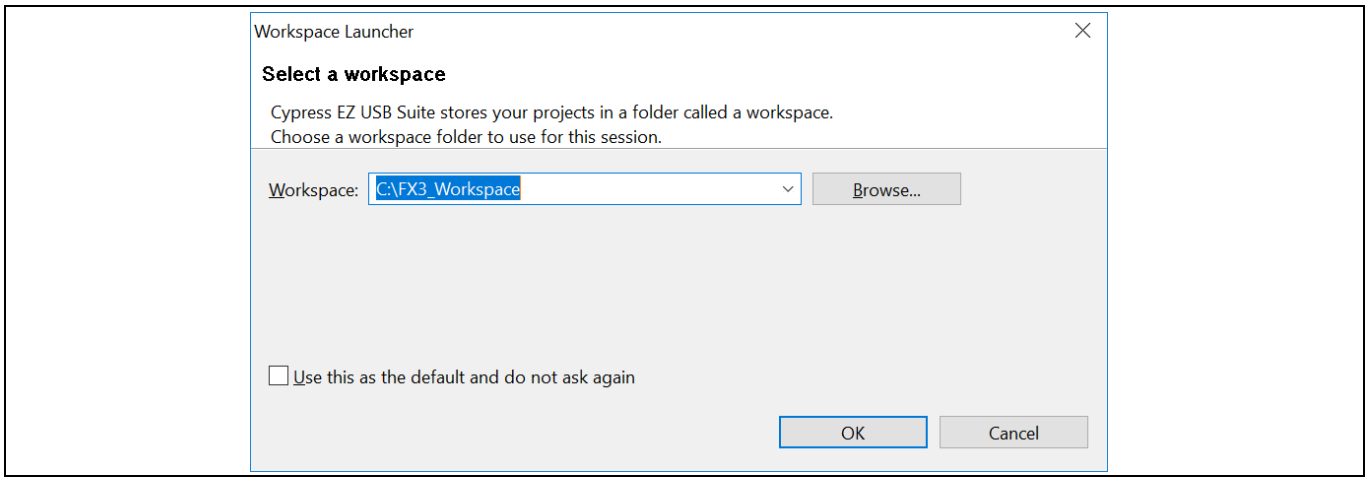


Figure 14　创建新 Workspace

c) 这时将打开一个空的“Eclipse IDE”，在“Project Explorer”选项卡上没有任何项目。下一步需要将英飞凌范例项目导入到“Project Explorer”中。

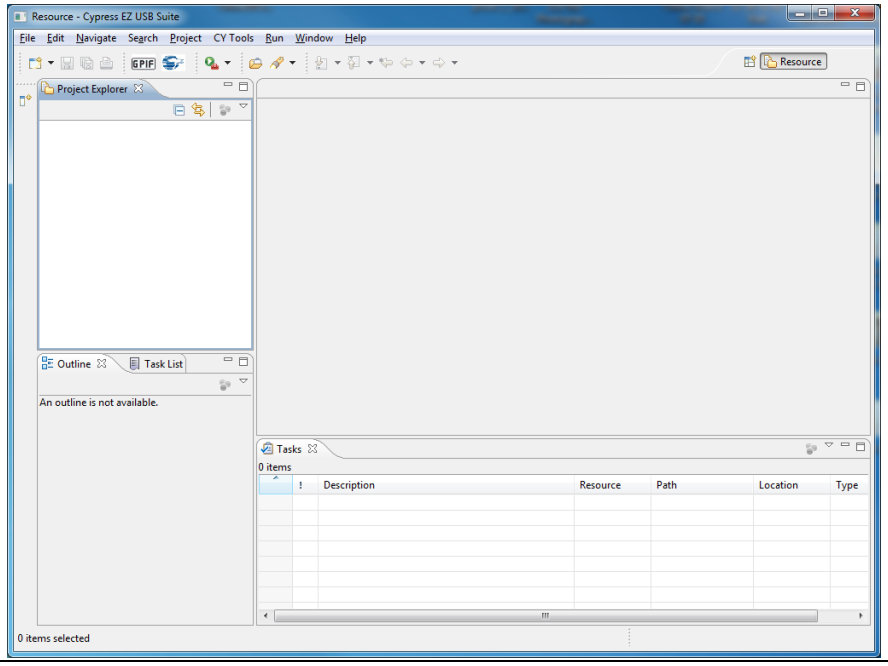


Figure 15　导入项目之前的 Eclipse IDE

d) 依次选择 **File** > **Import**。

e) 依次选择 **General** > **Existing Projects into Workspace**，然后点击 **Next**。

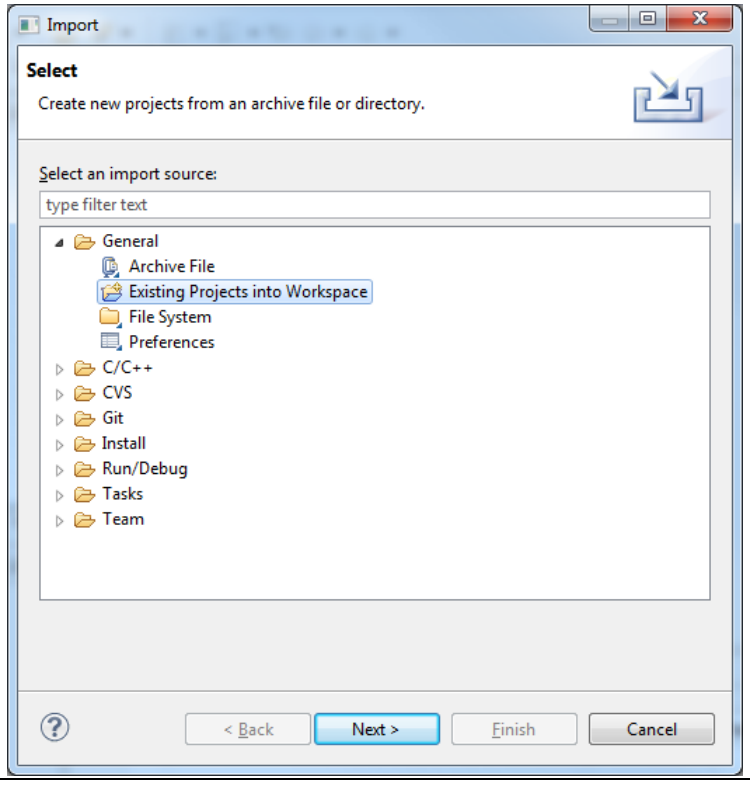


Figure 16　选择“Existing Projects into Workspace”

f) 浏览 SDK 安装包里的固件目录。点击 **Browse...**按钮，然后导航到 Cypress SDK **firmware** 文件夹。对于典型的 Windows 安装，该文件夹位于 `C:\Program Files (Program Files (x86 on 64-bit machines)\Cypress\EZ-USB FX3 SDK\firmware` 路径下。

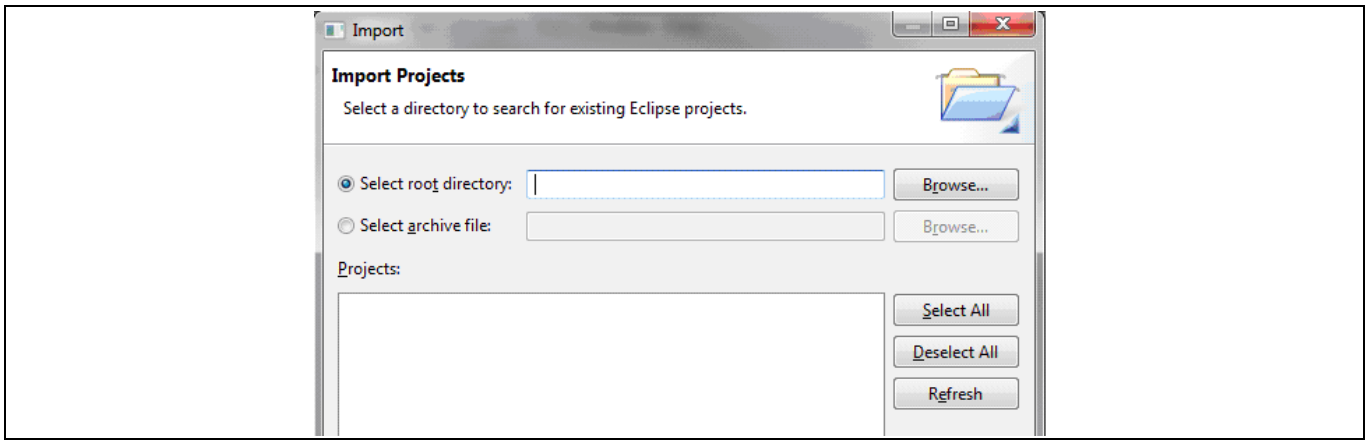


Figure 17　选择根目录

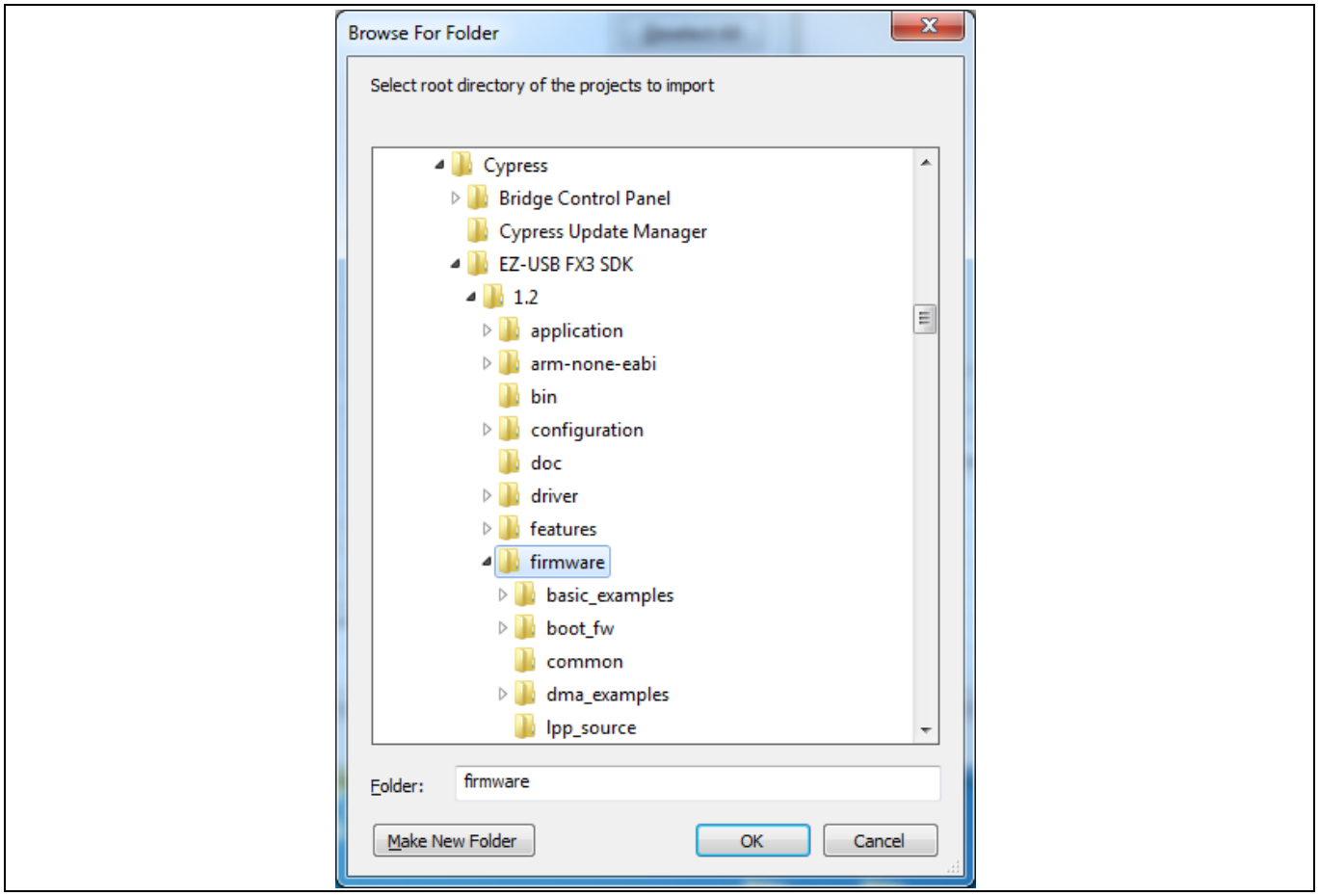


Figure 18　浏览 SDK 安装中的“firmware”目录

g) 勾选所有英飞凌范例项目。或者请点击 Select All 按钮。同样，可以选中 Copy projects into workspace，然后点击 Finish 按键。

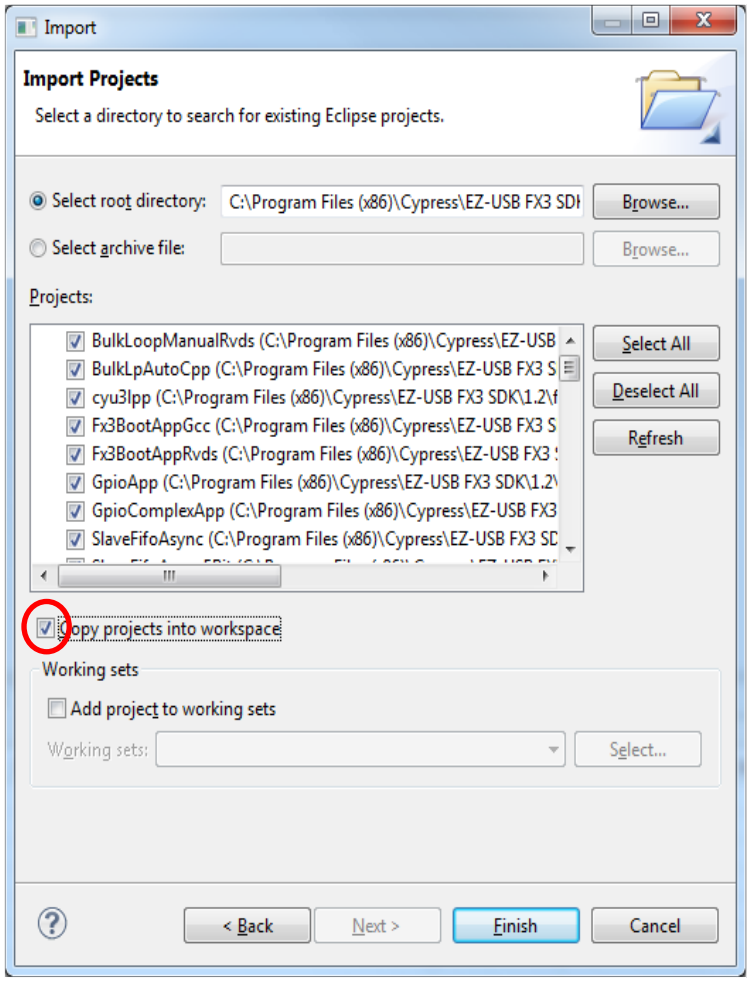


Figure 19　在固件目录下选择所有项目

h) 所有固件示例均被导入到工作区内，并且在“Eclipse Project Explorer”中显示。导入过程中自动编译了各项目，因此需要几分钟的时间来加载并编译所有项目。加载完成后，您可以运行这些项目，也可以修改和重新编译它们。

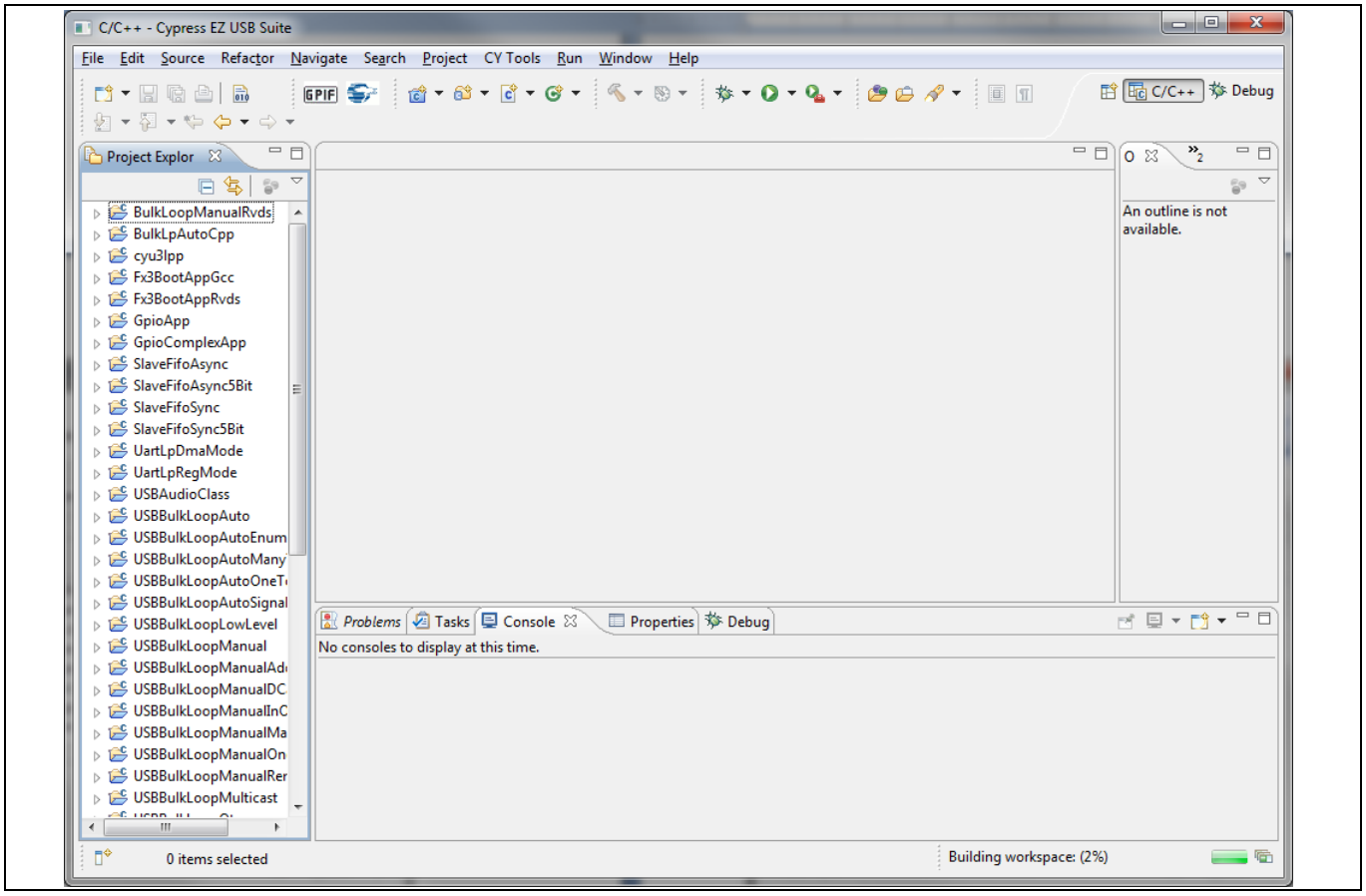


Figure 20　将所有项目导入到工作区内

8.1.1　构建并运行 USBBulkLoopAuto 固件

下一部分内容是构建并运行一个基本的固件示例。

EZ-USB™ FX3 SDK 中所提供的 USBBulkLoopAuto 固件示例默认配置了 BULK 端点 1 IN 以及 BULK 端点 1 OUT。EZ-USB™ FX3 固件使 USB 主机发送给 OUT 端点的数据循环返回到 IN 端点。相同数据可以从 IN 端点传送到 USB 主机。

1. 在“Eclipse IDE”中，通过双击“USBBulkLoopAuto”固件，或者通过点击它名字左边的小扩展箭头，可以打开“USBBulkLoopAuto”固件。您可以按照该固件的默认设置编译它，也可以根据下面介绍的步骤修改端点编号，进而将其更改。要运行该固件而不需要进行任何修改，请跳到 **8.1.2** 节。如想进行轻微修改后再运行，请继续进行步骤 2。建议执行各修改步骤，了解关于 Eclipse 工具链的详细信息，。
2. 在文件列表内，请双击 **cyfxbulkloopauto.h**，使其显示在编辑器窗口内。通过使用*.h* 文件的下方的标识符 CY_FX_EP_PRODUCER、CY_FX_EP_CONSUMER、CY_FX_EP_PRODUCER_SOCKET 以及 CY_FX_EP_CONSUMER_SOCKET，可以在 *cyfxbulkloopauto.h* 文件内定义端点编号及其相应的套接字。

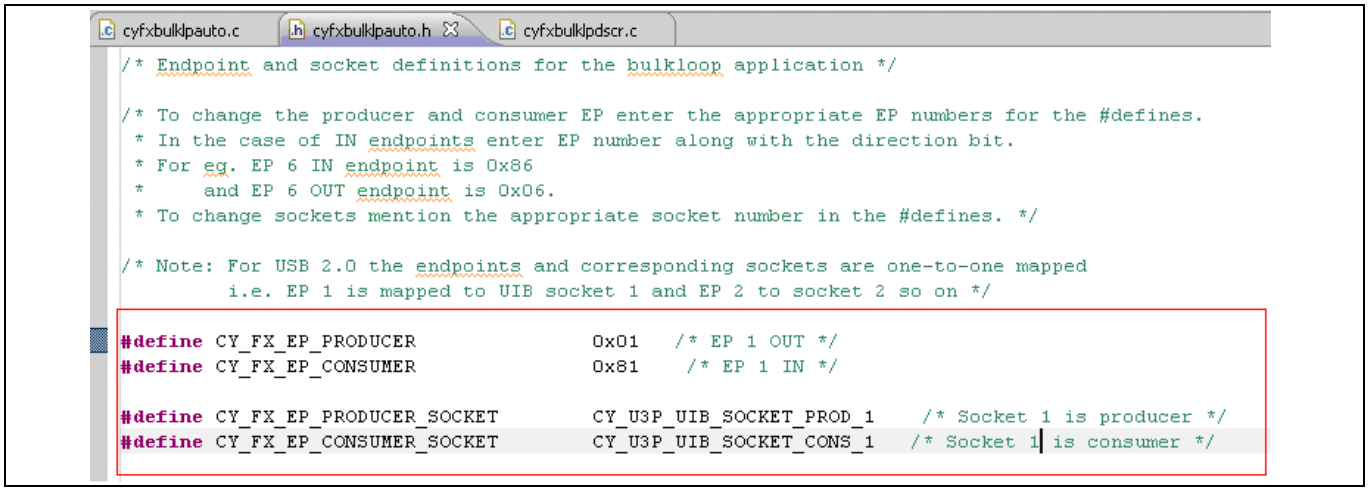


Figure 21 *cyfxbulklpauto.h* 中的端点与套接字定义

3. 使用上面介绍的标识符，端点编号以及相关的套接字从 EP1 被改为 EP2，如 Figure **Figure 22** 所示。因此，在修改后的固件中，EP2 IN 为 BULK IN，则 EP2 OUT 为 BULK OUT。

Note: 在 USB 中，一个端点是通过一个字节(包含：位 7 作为方向(1 表示输入，0 表示输出)；其他 7 个位作为地址(位[6:0])) 来标识的。一个 EZ-USB™ FX3 "套接字"在 USB 传输中相当于一个端点。USB 域中的套接字和其他外设域上(如 GPIF II 或甚至是 CPU) 的套接字相连接，用于数据传输。

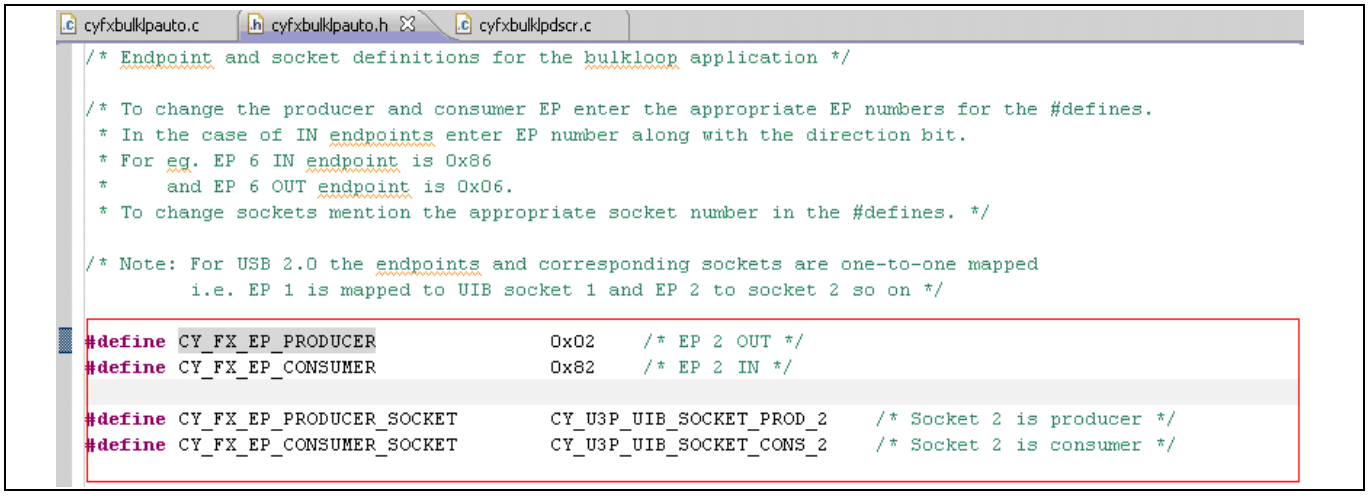


Figure 22 端点编号和相关套接字的修改 (从 EP1 改为 EP2)

4. 请保存修改后的 *cyfxbulklpauto.h* 文件 (**File** >**Save**)。Eclipse 编译过程不会自动保存新编辑的文件。确保在 Project Explorer 屏幕上展开了 USBBulkLoopAuto 项目，然后依次选择 **Project** > **Build Project**，实现项目的编译。

   同样，您可以在编译该项目前选择 **Build Configurations** 项。在 Project Explorer 面板中，右键点击项目名称；然后依次选择 **Build Configurations** > **Set Active** >**Debug/Release** (请参见 **Figure 23**)。

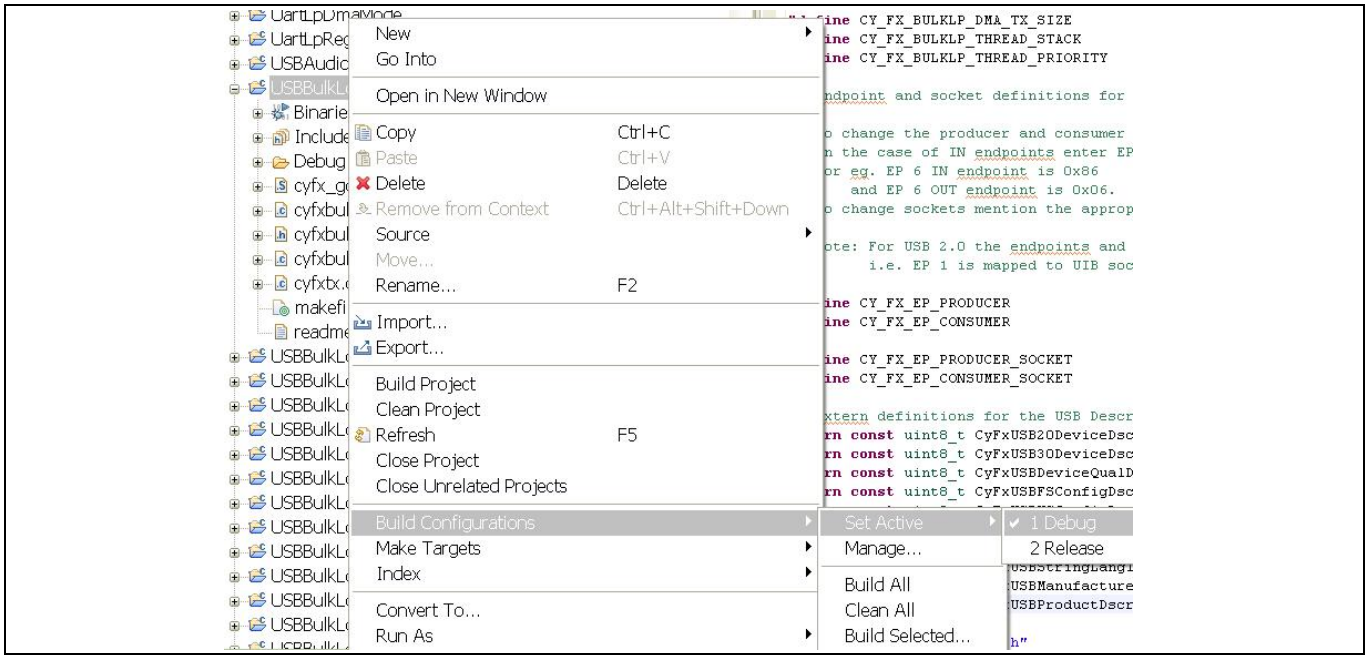


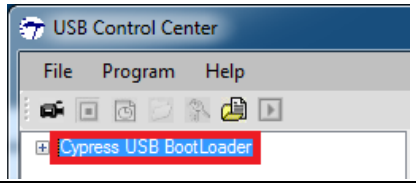
Figure 23 在 Eclipse IDE 中选中 Build Configuration 项

Note: ***Release** (释放) 模式下的固件镜像文件的尺寸小于 **Debug** 模式下的大小。调试模式的固件镜像包含了附加的调试符号，有助于用户使用 JTAG 进行调试。*

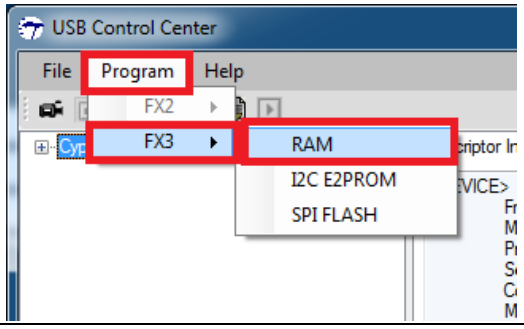
8.1.2 将 USBBulkLoopAuto 固件加载到 EZ-USB™ FX3 RAM 内

通过名为“Cypress Control Center” (英飞凌控制中心) 的应用将代码加载到 EZ-USB™ FX3 中。Control Center 应用作为 EZ-USB™ FX3 SDK 的组成部分安装。

1. 通过 Start 菜单，打开 Cypress Control Center 应用。依次选择 Start > All Programs >Cypress > Cypress USBSuite > Control Center。
2. 使用套件附带的 USB 3.0 线缆，通过 SuperSpeed 或 Hi-Speed USB 端口将开发板与电脑相连。如果您先前已经安装了套件设置和驱动软件，则 USB Control Center 应用将会显示“Cypress USB Bootloader”，表示设备已连接。但是，如果您收到一条“no drive found (未找到驱动程序)”的信息，请参见**附录 B** 来安装 Windows 驱动程序。



3. 要想加载 EZ-USB™ FX3 RAM 的固件，请依次选择 **Program** > **FX3** >**RAM**。



4. 导航到需要加载到 EZ-USB™ FX3 的固件镜像文件。该文件是 Eclipse 在编译过程的最后步骤中生成的 *USBBulkLoopAuto.img* 文件。该文件的默认工作区所在的位置是：C:\Users\antioch\Work\USBBulkLoopAuto\Debug。选择文件后点击 **Open** 按钮，或者双击文件名称。

   USB Control Center 下载新的 EZ-USB™ FX3 代码后，Windows 将检测到名为“Cypress USB BulkLoopExample”的新 USB 设备。这便是 EZ-USB™ FX3 的重新枚举操作。EZ-USB™ FX3 最初作为 Bootloader 出现在 Windows 中，但加载新的代码后，它便同 USB 断开，然后作为由下载代码定义的新设备 (BULK 环路) 重新连接到 USB。注意，EZ-USB™ FX3 “Cypress USB Bootloader”消失，并且在 Control Center 菜单下有了一个新的 USB 外设，显示为“Cypress USB BulkloopExample”。EZ-USB™ FX3 以新的供应商 ID/产品 ID (分别为 0x04B4 和 0x00F0) 重新枚举。

Note: *当您需要下载新的项目，请使用 RESET 开关来复位 EZ-USB™ FX3，使其返回到 Bootloader 模式。*

8.1.3 执行 BULK 传输

1. 展开“Cypress USB Bulkloop Example”条目下所有的子条目。如果您按照第 1 到第 6 步的介绍修改并编译了固件，那么您将会看到 BULK EP2 OUT 和 BULK EP 2 IN 出现在接口 0 中备用设置 0 的下面 (Figure **Figure 24**)。

Note: *如果您没有对固件进行修改，在 Control Center 中会显示默认的“BULK out endpoint (0x01)”和“BULK in endpoint (0x81)”。*

2. 要执行一个 BULK OUT 传输，请点击 Data Transfers 选项卡，然后选择左侧屏幕内的 BULK out endpoint (0x02)项。向 Bytes to Transfer 字段输入需要传输的字节数量；向 Data to send (Hex)字段输入需要传输的数据。Control Center 将自动在十六进制数字对之间插入空格。点击 Transfer Data-OUT。**Figure 24** 显示了将 5 个字节传输到 BULK OUT 端点(0x02)的设置。

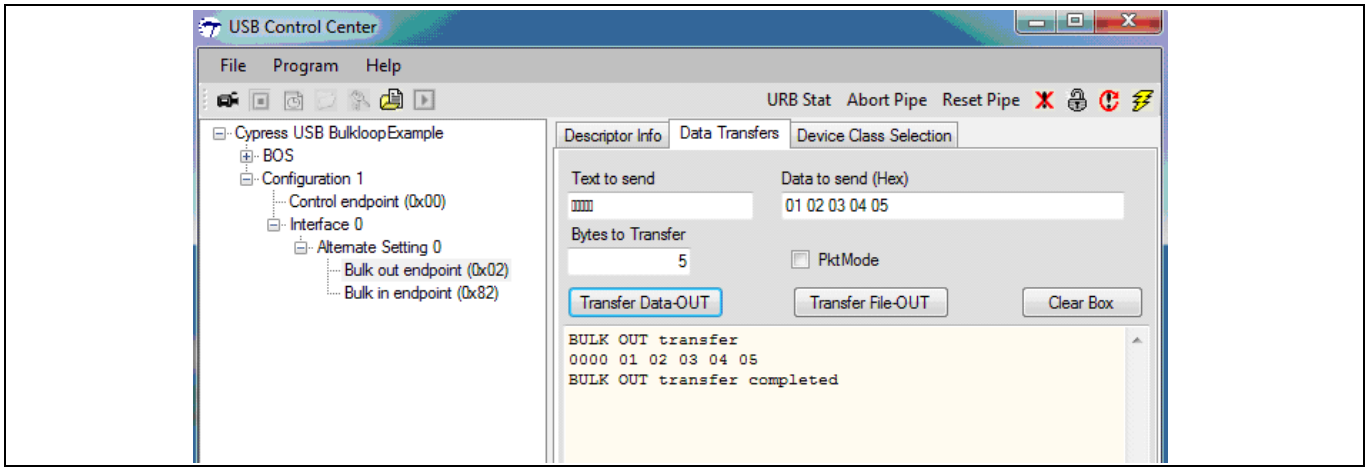


Figure 24 BULK OUT 传输

3. BULK Loop 程序使用相应编号的 IN 端点回送通过 OUT 端点接收的所有数据。要执行 BULK IN 传输，请选择左侧屏幕中的 **BULK in endpoint (0x82)**。向 **Bytes to Transfer** 字段内输入需要传输的字节数量。点击 **Transfer Data-IN**。**Figure 25** 显示了在 IN 端点 (0x82) 上接收到相同的 5 个字节。

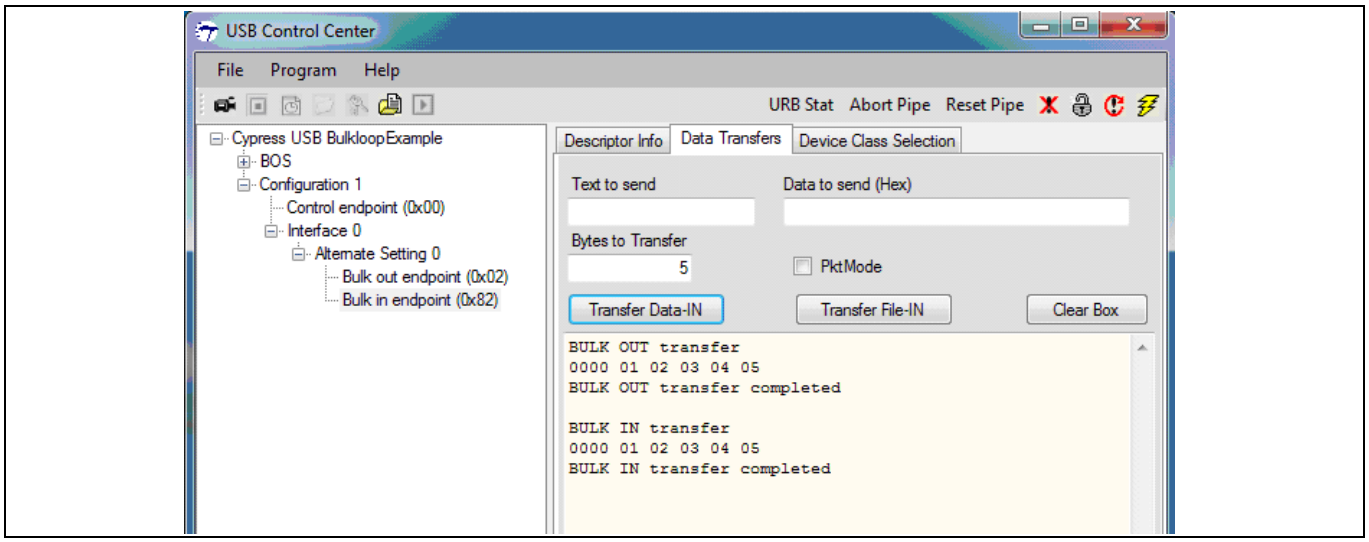


Figure 25 BULK IN 传输

9 开发工具

目前用户已经对 EZ-USB™ FX3 设备进行了基本的操作，本部分将对 SuperSpeed Explorer 套件 (CYUSB3KIT-003) 和 EZ-USB™ FX3 SDK 套件提供简要说明。

9.1 SuperSpeed Explorer 套件简介

英飞凌的 SuperSpeed Explorer 套件提供了您需要使用的硬件。PCB 提供了 EZ-USB™ FX3 所需的时钟和电源，以及可配置 I/O 电压。该套件拥有高速接口，支持与外设连接。SuperSpeed Explorer 套件支持 USB 启动和 I^2C 启动。该开发板还拥有一个 8 Mbit 的 I^2C EEPROM，用于存储固件并测试从 I^2C EEPROM 启动的操作。SDK 提供的英飞凌示例项目包含了固件，用于通过使用 DVK 开发板编程这些设备。如果您未安装 SuperSpeed Explorer 套件，可从**此处**获取它。

两个很重要的硬件部分 (处于 EZ-USB™ FX3 DVK 开发板外) 分别是 USB 3.0 主机 (通常为 PC) 以及与 GPIF II 接口相连的外部设备 (如图像传感器或 FPGA)。有关 SuperSpeed Explorer 套件的详细信息，请参考**套件用户指南**。**Figure 26** 显示了 SuperSpeed Explorer 套件电路板的重点区域。

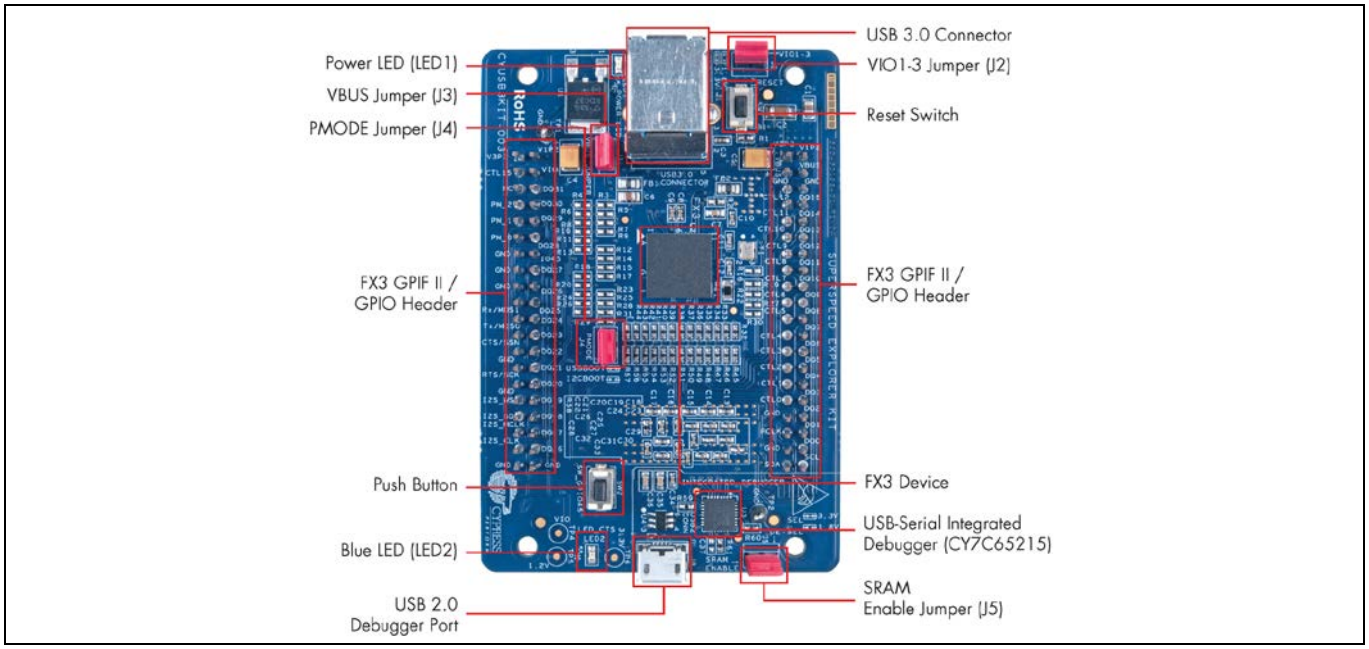


Figure 26 EZ-USB™ FX3 DVK 电路板

Table 2 显示了 DVK 开发板上的重要跳线器和它们的默认设置情况。

Table 2 SuperSpeed Explorer 套件电路板的跳线器

跳线器	跳线器状态	功能
J2	短接	将电压域 VIO1、VIO2 和 VOI3 的电压都设置为 3.3 V，用以支持外部 3.3 V 接口和板上 SRAM。
J3	短接	开发板由 USB 3.0 VBUS 供电。为了保证电路板的正常工作，请插好该跳线器。
J4	开路	如果 EEPROM 上存储一个有效的固件镜像，那么电路板将从 I2C EEPROM 启动。否则，它将返回到 USB 启动。 注意：默认情况下，EEPROM 将存储带有 LED 闪烁功能的 USBBulkSourceSink 固件。
J5	开路	取消选择外部 SRAM。

9.2 EZ-USB™ FX3 SDK 简介

英飞凌提供了一整套用于 EZ-USB™ FX3 的软件和固件库，以便在嵌入式系统环境中集成 USB 应用。软件开发套件 (SDK) 中带有应用示例，可加快应用开发。可从英飞凌网站上下载 SDK。**Figure 27** 显示了安装 SDK 时所创建的目录结构。

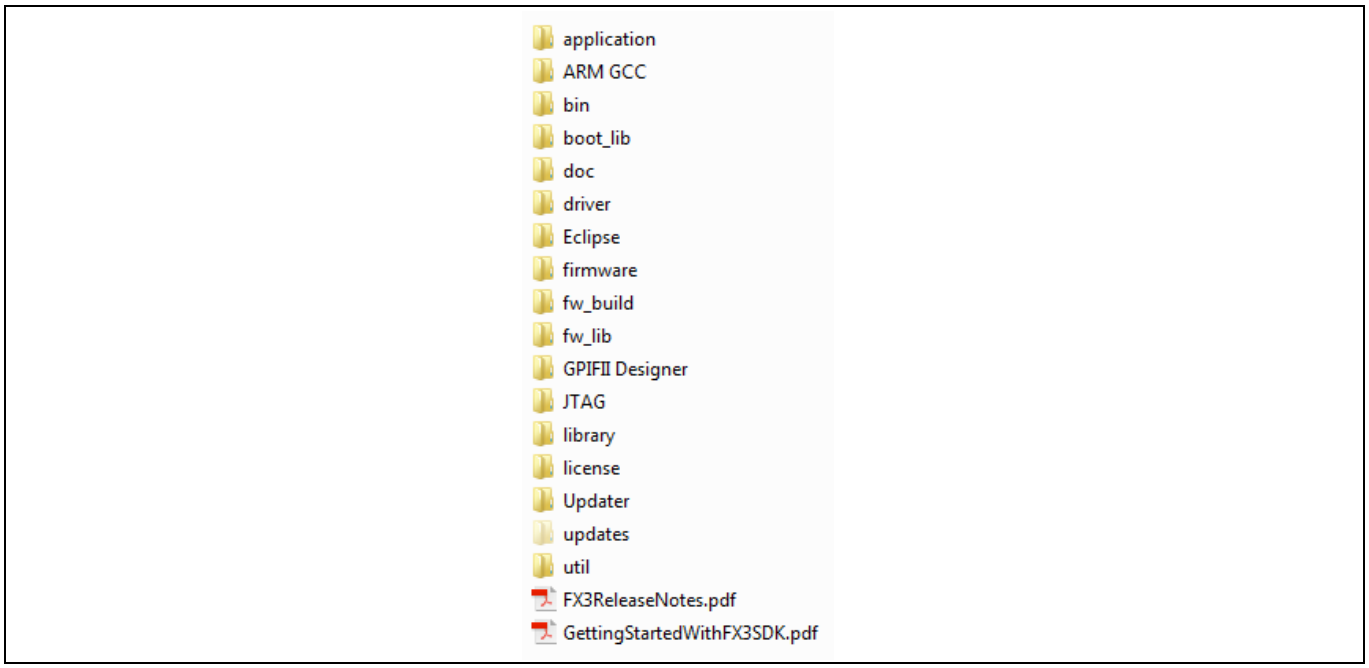


Figure 27 目录结构

9.2.1 固件库与 API

英飞凌提供了一个功能强大的 API 库，能使再复杂的设计也变得更简单。下面介绍了 EZ-USB™ FX3 SDK 的一些优点：

- RTOS
  - 包含在内的 ThreadX Real Time OS (多线程实时操作系统，RTOS) 简化了固件的开发过程。通过 RTOS，可以很容易地创建多个线程以简化固件流程。
- 模块方法
  - 使用基于 API 的方法时，开发人员不需要掌握有关 EZ-USB™ FX3 的寄存器和位的知识，这样可以专注于固件逻辑和流程。该方法有助于使用、调试、修改和支持 EZ-USB™ FX3 SDK。

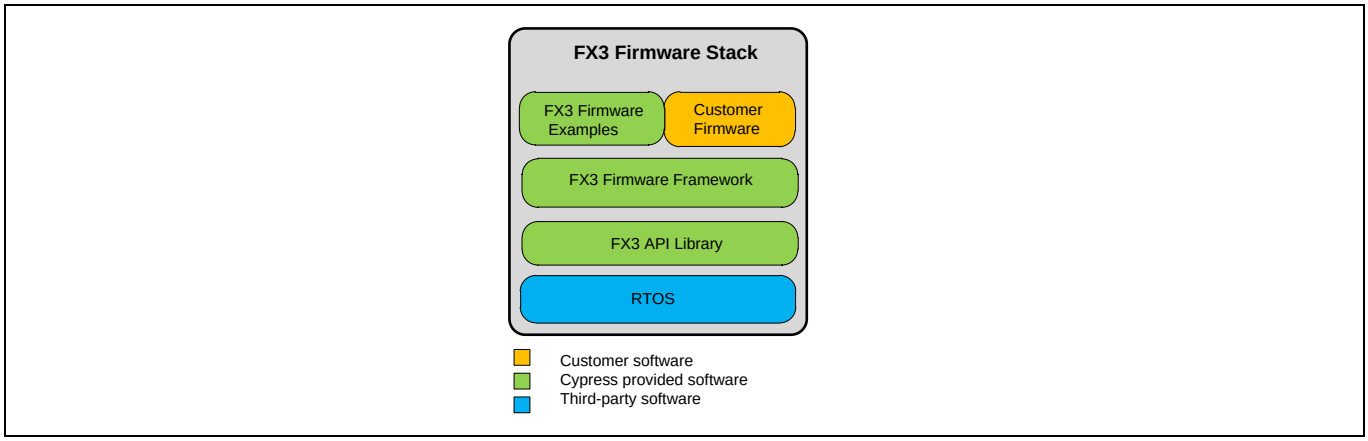


Figure 28　固件 SDK 的结构

API 支持：

- 编程 EZ-USB™ FX3 中每一个独立的模块 (GPIF II、USB 以及串行接口)
- 编程 DMA 引擎，并设置这些模块之间的数据流
- ThreadX OS 调用应用所需的函数
- 调试功能
- USB 主机模式操作
- 电源管理功能

9.2.2　框架 API

该固件 (或应用) 框架包含了启动和初始化代码。同样也包含了 USB、GPIF II 以及串行接口模块使用的独立驱动程序。该框架支持：

- 定义程序的入口
- 执行堆栈设置
- 执行内核初始化
- 提供用于应用线程启动代码的占位符

9.3　EZ-USB™ FX3 固件示例

SDK 提供多个固件示例，如上面的“Eclipse Installation”部分所示。本部分详细介绍了一些更常见的示例固件项目。当前可用的固件示例的完整列表可从 SDK 发布说明中获取。所有示例都能在 USB 2.0 和 USB 3.0 的速度下工作。

Note:　*示例固件以单独的 Eclipse 项目形式提供。这样，通过“Eclipse Project Explorer”，您可对完整示例固件文件列表的情况一目了然。*

9.3.1　USB BULK 数据回送示例

这里介绍了 USB BULK 端点之间的简单回送机制。来自 USB 主机 (PC) 的所有标准设置请求都由 EZ-USB™ FX3 应用示例处理。该固件示例使用 DMA AUTO 或 MANUAL 通道实现回送。

DMA 多通道示例使用了三个端点进行回送。

9.3.2 USB ISOCHRONOUS 数据回送示例

这些示例说明了各 USB ISOCHRONOUS 端点之间的回送机制。它类似于 BULK 回送示例，不同的是这里使用了同步端点，而不是 BULK 端点。

9.3.3 从 FIFO 应用示例

从设备 FIFO 应用示例演示了 USB 主机和外部 FIFO 控制器之间的数据传输。该示例包含了 USB 主机和外部主设备之间的两个单向数据通道。可将 GPIF II 接口配置为使用了一个 16 位或 32 位总线的同步或异步从设备 FIFO 传输。

9.3.4 串行接口示例

这些示例演示的是到 GPIOs、I^2C、SPI 和 UART 的数据访问。

9.3.5 USB BULK/ISOCHRONOUS 数据源库示例

这些示例说明了通过两个 USB BULK/ISOCHRONOUS 端点执行的数据传输/接收机制。

9.3.6 存储器编程器示例

该示例说明了通过 USB 对 I^2C EEPROMS 和 SPI Flash 设备进行的编程操作。读取或写入操作是通过使用预定义的供应商指令完成。使用这些指令可以将启动镜像编程到这些设备中。

9.3.7 大容量存储类示例

该示例说明了一个 USB 海量存储类 (BULK Only Transport) 器件的实施情况，这些器件使用 EZ-USB™ FX3 设备 RAM 中的一小部分作为存储设备。示例显示的是海量存储指令如何在 EZ-USB™ FX3 固件内被解析和处理。

9.3.8 USB 音频类示例

该示例创建了一个 USB 音频类麦克风，它将存储在 SPI Flash 存储器内的 PCM 音频数据传输到 USB 主机内。该示例仅能在 USB 2.0 的速度下运行。

9.3.9 双阶段引导器示例

一套简单的 API 作为一个独立的库使用，以实现二次启动。该示例演示了这些 API 的使用。同时还提供了可用于实时查看工具链的配置文件。

9.3.10 USB 主机与 OTG 示例

这些示例验证了 EZ-USB™ FX3 USB 端口的主机模式和 OTG 模式操作。

9.4 EZ-USB™ FX3 固件开发工具

9.4.1 Eclipse IDE

为 C/C++开发者提供的 Eclipse IDE 是 EZ-USB™ FX3 SDK 的一部分。该 IDE 包括基础 Eclipse 平台和 CPP 功能。IDE 包含开发所需插件。

- 支持 GNU Arm C/C++开发
- Zylin 嵌入式 CDT

开发工具

- 它是 Eclipse IDE 中的一个通用插件，通过该插件可以使用 GNU 调试器调试 EZ-USB™ FX3 固件。
- Java(TM) 平台、标准版实时环境版本 7 (JRE)

9.4.2 GNU 工具链

GNU 工具链作为 EZ-USB™ FX3 SDK 的部分，包括以下内容：

- GCC 编译器 (gcc)
- GNU 链接器 (ld)
- GNU 汇编器 (as)
- GNU 调试器 (gdb)

这些可执行文件都由 Eclipse IDE 调用。

9.4.3 GPIF II Designer (设计程序)

GPIF II 接口设计工具是一个 Windows 应用，并作为 EZ-USB™ FX3 SDK 的一部分。该工具提供了一个图形用户界面，用于为目标设备指定必要的接口。用户使用状态机设计了一个接口，工具将它翻译成一个 C 头文件，并包含到 Eclipse 项目。

9.4.4 集成调试器

SuperSpeed Explorer 套件将 **CY7C65215 USB-Serial IC** 作为集成调试器使用。CY7C65215 是一个全速 USB 串行桥接控制器，提供了两个可配置串行通信模块 (SCB)。SuperSpeed Explorer 套件将 CY7C65215 的第一个串行通道作为 UART 使用，并将第二个串行通道作为 JTAG 接口使用。

SuperSpeed Explorer 套件仅支持由集成调试器提供的 JTAG 接口。集成调试器支持 OpenJTAG 协议，并使用主机上的 OpenOCD 守护进程工具进行调试会话。

有关集成调试器的详细信息，请参考 **SuperSpeed Explorer 套件用户指南**。

9.5 Windows 软件概述

英飞凌提供了设备驱动程序和接口 API 以及各库，用于在 Windows 中开发 EZ-USB™ FX3 USB 应用。

9.5.1 Windows USB 设备驱动程序

“cyusb3.sys”是一个 USB 设备驱动程序，适用对象包括：32 位的 Windows XP、32/64 位的 Windows Vista、32/64 位的 Windows 7 以及 32/64 位的 Windows 8。该驱动程序能够同所有符合 USB 2.0 和 USB 3.0 规范的设备进行通信。它是通用的驱动程序，能够理解基本的 USB 指令。然而它没有支持 USB 设备类协议。例如，驱动程序不能直接将一个 USB 大容量存储设备连接到 Windows 文件系统。应用固件必须实现该逻辑。

英飞凌通用驱动程序的 VID 为 0x04B4，PID 为 0x00F0、0x00F1 和 0x00F3。如果您想用自己的 VID/PID，您必须获得由微软 WHQL 签署的驱动程序。开发过程中，您可以通过禁用电脑上的软件签名强制来使用一个未签名的驱动程序。该驱动程序是自定义的 USB 应用同供货商特定的设备进行通信的理想选择。或者，对于实验或诊断应用，驱动程序可用于将低级 USB 请求发送给所有 USB 设备。为了使用驱动程序同某个设备通信，Windows 必须使该设备同驱动程序相匹配。类库 (*CyAPI.lib* 和 *Cyusb.dll*) 为驱动程序提供了一个高级的编程接口。

9.5.2 特性

- 与 Windows 驱动程序框架 (WDF) 相兼容
- 与所有 USB 2.0 兼容的设备相兼容
- 与 USB 3.0 兼容的设备相兼容
- 支持 USB 3.0 的基本性能
- 支持 Windows 即插即用性能以及电源管理
- 支持 USB 远程唤醒
- 支持 CONTROL (控制)、BULK (批量)、INTERRUPT (中断) 以及 ISOCHRONOUS (同步) 等端点
- 支持多个 USB 设备同时连接
- 支持可自定义的驱动程序 GUID (而无需重建驱动程序)
- 可在每一帧传送多个数据包，因此能够实现高带宽的数据传输

您不需要使用英飞凌所提供的驱动程序。可以编程 EZ-USB™ FX3，以实现标准 USB 类设备。实现这样的设备时，应该使用 USB 类驱动程序来代替。例如：如果 EZ-USB™ FX3 正在执行 USB 视频类别 (UVC)，则不需要使用英飞凌通用驱动程序，取而代之的是操作系统 OS 所提供的 UVC 驱动程序。这样会让 Windows，Linux 和 Mac 的支持变得简单，因为操作系统 OS 已经提供了相应的标准驱动。

9.6 应用接口

9.6.1 CyAPI.lib

CyAPI.lib 为 USB 设备提供了一个既简单又强大的 C++编程接口。C++类库为 *cyusb3.sys* 驱动程序提供了一个高级编程接口。此库只能与该驱动程序所支持的 USB 设备通信。

有关 *CyAPI.lib* 的详细信息，请参考《C++库编程者参考手册》中介绍的内容。它位于 USB 套件帮助文档中，并作为 SDK 的一部分。

9.6.2 CyUSB.dll

CyUSB.dll 是一个托管 Microsoft.NET 类库，为 USB 设备提供了一个高级编程接口。因此应用可通过库方法和属性来访问 USB 设备，而不用通过低级的 Win32 API 调用与 USB 设备驱动程序直接通信。由于 *CyUSB.dll* 是一个托管.NET 库，因此用户可通过 Visual Basic.NET、C#、Visual J#和托管 C++等所有 Microsoft Visual Stuido.NET 语言对它的类和方法进行访问。如要使用该库，请将 *CyUSB.dll* 参考添加到项目的 **References** (参考) 文件夹中。然后，访问 CyUSB 命名空间的任何源文件都必须拥有用于按照准确语法添加命名空间的一行。

有关 *CyUSB.dll* 的详细信息，请参考《C#库编程者参考手册》中的内容。它位于 USB 套件帮助文档中，并作为 SDK 的一部分。

请参考 **USB 套件应用开发 — 快速入门指南，了解有关使用 CyAPI.lib 和 CyUSB.dll 的详细信息。**

9.7 Windows 软件示例

9.7.1 BULKLoop 示例

通过 BULKLoop 应用测试 BULK 端点上数据传输的回送。

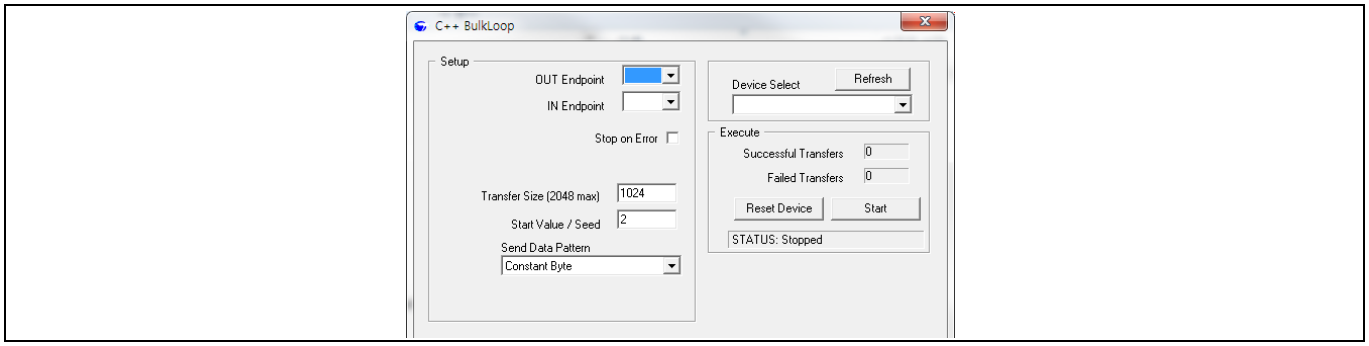


Figure 29 C++ BULKloop 应用

9.8 Streamer 示例

Streamer 应用用于测试 BULK/ISOCHRONOUS 端点上的数据传输。

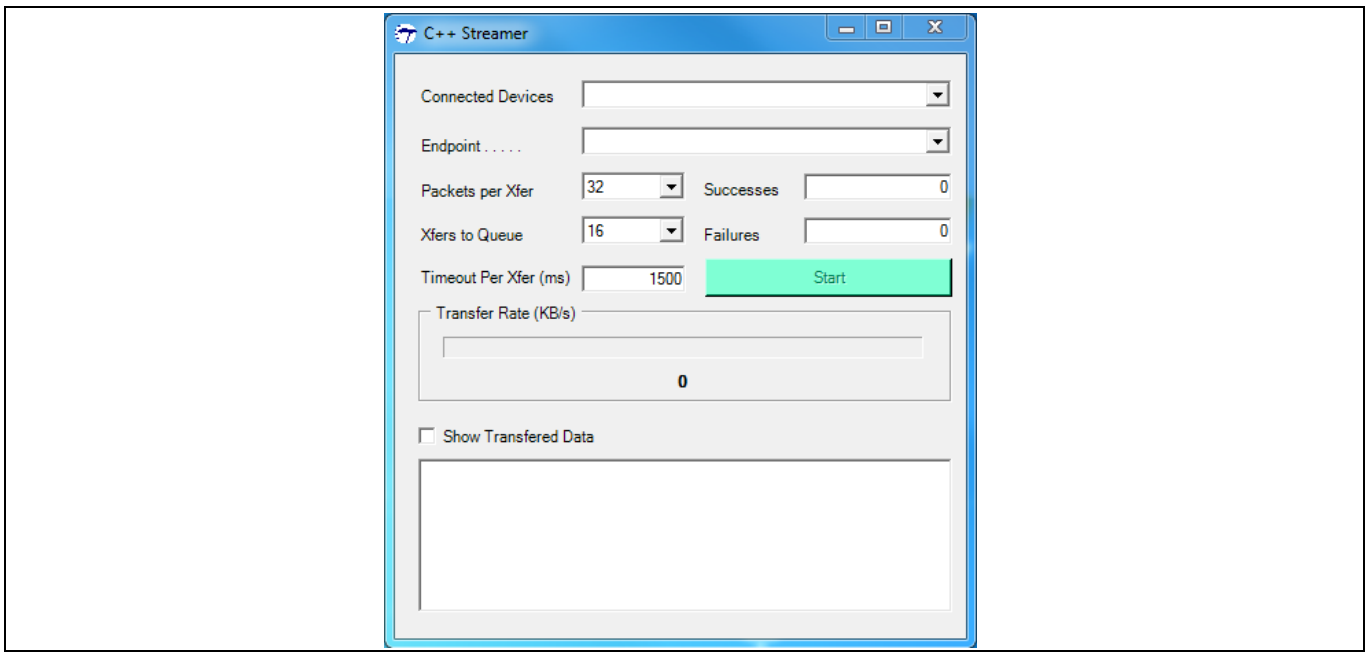


Figure 30 C++ Streamer 应用

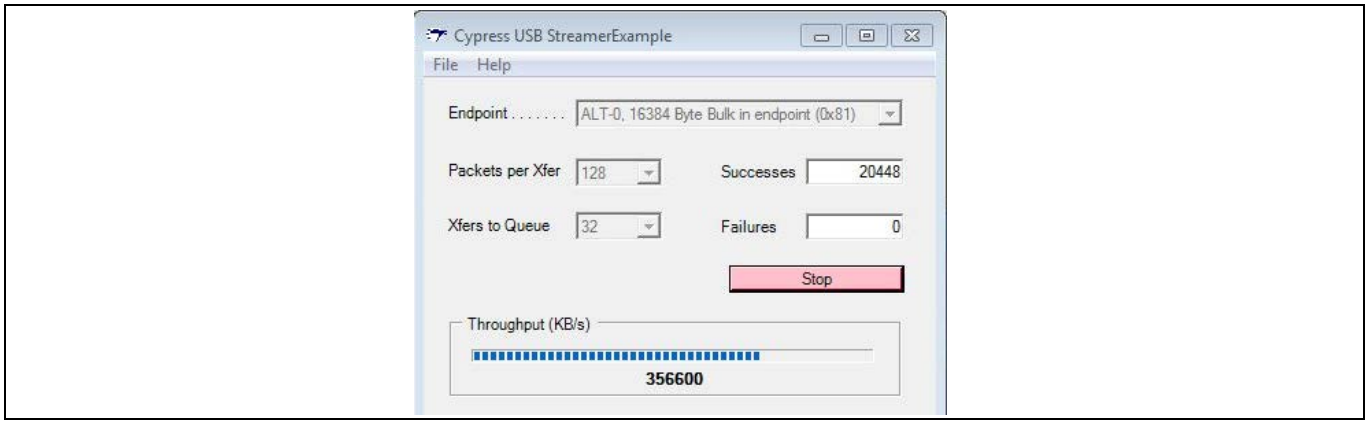


Figure 31 C# Streamer 应用 (提供从设备 FIFO 应用接口传输)

安装 EZ-USB™ FX3 SDK 时，'applications' (应用) 目录中提供了 BULKLoop 以及 Streamer 应用的 C++和 C#语言的实现。

9.9 适用于 Linux 的 EZ-USB™ FX3 SDK 和软件

9.9.1 适用于 Linux 的 EZ-USB™ FX3 SDK

EZ-USB™ FX3 SDK 支持使用 Eclipse IDE 来开发固件以及使用 J-Link JTAG 调试器在 Linux 平台上进行调试。

适用于 Linux 的 EZ-USB™ FX3 SDK 包括以下部分：

- EZ-USB™ FX3 固件库和示例
- Sourcery Arm GNU 工具链
- 用于安装 32 位 Linux 操作系统的 Eclipse IDE
- 用于安装 64 位 Linux 操作系统的 Eclipse IDE
- 用于 Linux 软件的 CyUSB Suite

安装过程包括提取这些档案和设置环境变量。可以在 **Linux 的 EZ-USB™ FX3 SDK** 部分中查看 *FX3_SDK_Linux_Support.pdf* 文档，了解各安装步骤的详细信息。

9.9.2 用于 Linux 软件的 CyUSB Suite

通过用于 Linux 软件的 CyUSB Suite，您可以将固件镜像加载到 EZ-USB™ FX3 设备，并测试设备上的各种接口。请参考位于 **Linux 的 EZ-USB™ FX3 SDK** 安装文件夹中的 *cyusb_linux_user_guide.pdf*：fx3_sdk_<version>_linux\cyusb_linux_1.0.4\docs. 该文档对安装软件、将固件下载到 EZ-USB™ FX3 内、测试供应商扩展、BULK OUT/IN 传输和 ISOCHRONOUS OUT/IN 传输进行了介绍。

Linux 的 CyUSB Suite — 编程者参考手册 (*cyusb_linux_programmers_guide.pdf* 位于相同的文件夹中) 描述了 Linux 的 cyusb 库，同时也介绍了如何使用该库来构建并集成用户编写的应用。通过该软件，您可以进行下列操作：

- 查看所插入的各设备的设备、配置、接口、备用接口以及端点描述符。
- 选择用于通信的特定接口和备用接口。
- 将固件加载到 EZ-USB™ FX3 设备、RAM、基于 I^2C 的 EEPROM 或基于 SPI Flash 内。
- 加载执行指令的特定固件后，可以测试这些指令 (供货商扩展)。

- 通过传送常量/随机/递增数据块，可以测试 BULK OUT 端点。另外，在传送 OUT 数据后，可通过回送数据来测试 BULK IN 端点。
- 测试 ISOCHRONOUS OUT 和 ISOCHRONOUS IN 端点，并测量数据传输速率。

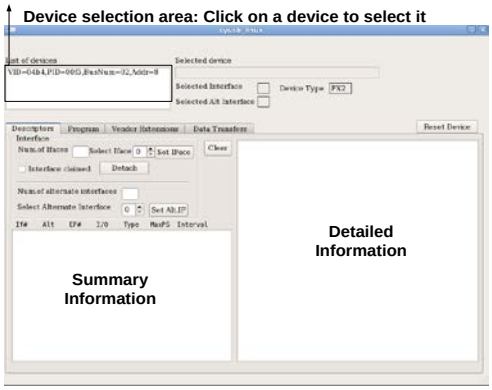


Figure 32　用于 Linux 软件的 CyUSB Suite

9.10　有用的调试工具

9.10.1　USB 3.0 协议分析仪

USB 3.0 协议分析仪是非常有用的调试工具。它可以分析 EZ-USB™ FX3 与主机间的 USB 通信。每个分析仪中的软件工具将对 USB 传输包中的数据进行解码。通过数据分析，可以容易发现问题并最大限度提高性能。一些 USB 3.0 分析仪已经上市。英飞凌没有推荐任何特定分析仪，但可以参考下面各选项：

- 独立式的 USB 3.0 协议分析仪
  - **Ellisys USB 探测器 280**
  - **LeCory USB Voyager M3i**
  - **Beagle USB 5000 SuperSpeed 协议分析仪**
- PC 软件 USB 3.0 协议分析仪
  - **SourceQuest SourceUSB**
  - **SysNucleus USBTrace**

9.10.2　逻辑分析仪

逻辑分析仪可以对数字信号进行简单分析。该分析仪用于观测 EZ-USB™ FX3 和其他外设间的信号。已经上市的逻辑分析仪共有两种：

- 独立式逻辑分析仪
  - **安捷伦 16800 系列便携式逻辑分析仪**
- 基于 PC 的逻辑分析仪
  - **USBee 逻辑分析仪**
  - **ZeroPlus 逻辑分析仪**

请注意，分析仪的信号频率范围应该高于所需分析的信号频率。

10 附录 A. USB 3.0 概述

USB 3.0 支持高达 5 Gbps 的数据传输速率，降低功耗，并与 USB 2.0 向后兼容。由 USB-IF 发布的 USB 规范可在此处找到。**Figure 33** 显示了 USB 3.0 的架构。

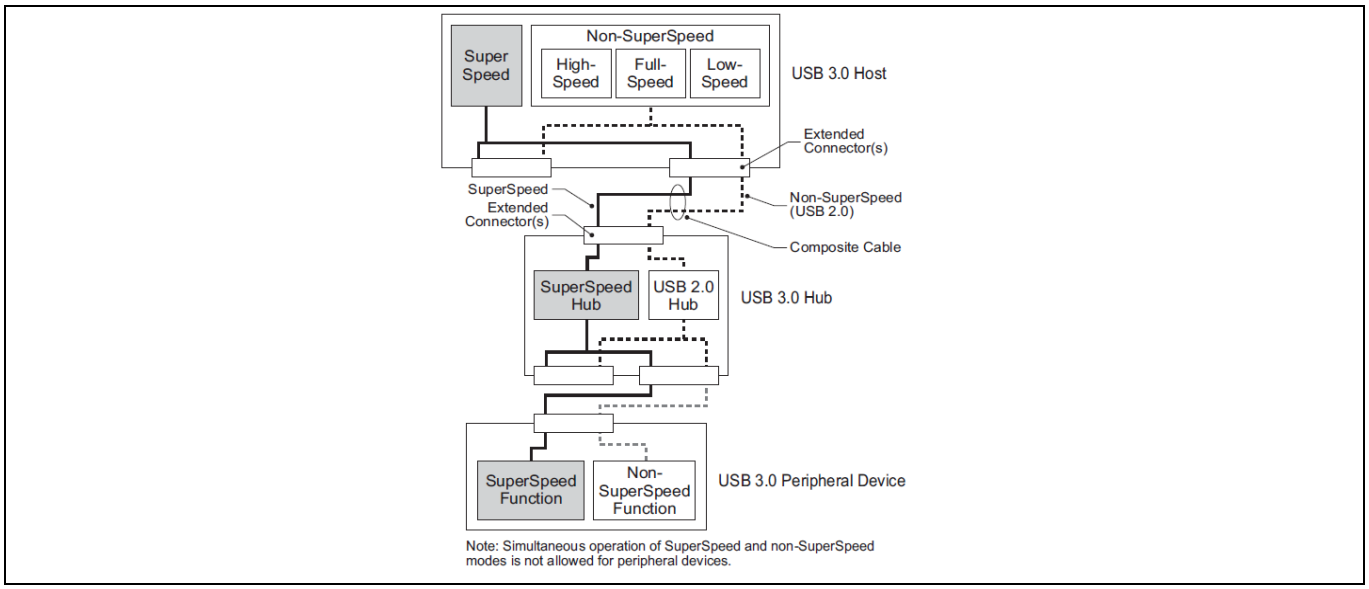


Figure 33 USB 3.0 双总线架构。

参考源：**http://usb.org**

除了 USB 2.0 接口外，USB 3.0 的物理接口增加了两对差分引脚和一个接地引脚，用于超高速传输。这样可以保证 USB 3.0 能够与 USB 2.0 向后兼容。

10.1 电气接口

USB 3.0 的引脚分布不同于 USB 2.0 的引脚分布. 除了具有 USB 2.0 的 VBUS、D–、D+和 GND 引脚外，USB 3.0 还有另外五个引脚，即两对差分引脚和一个接地引脚 (GND_DRAIN)。两对差分引脚用于超高速数据传输，支持双单工超高速的信号传输。GND_DRAIN 引脚用于地线终端、信号完整性的管理以及抗 EMI 干扰性能。**Table 3** 是对九个引脚的说明。

Table 3 USB 3.0 引脚说明

引脚名称	说明
VBUS	电源
D-	USB 2.0 的差分对
D+	
GND	电源的接地端
SSRX-	超高速接收差分对
SSRX+	
SSTX-	超高速传送差分对
SSTX+	
GND_DRAIN	返回信号的接地端

10.2 线缆与连接

与 USB2.0 相比，USB 3.0 添加了四个数据线 (即 SSRX+、SSRX-、SSTX+和 SSTX-) 用于数据传输，以及一个接地线用于地线终端、信号完整性管理以及抗 EMI 的干扰。**Figure 34** 显示了 USB 3.0 线缆的结构。**Table 4** 是这些线的说明。

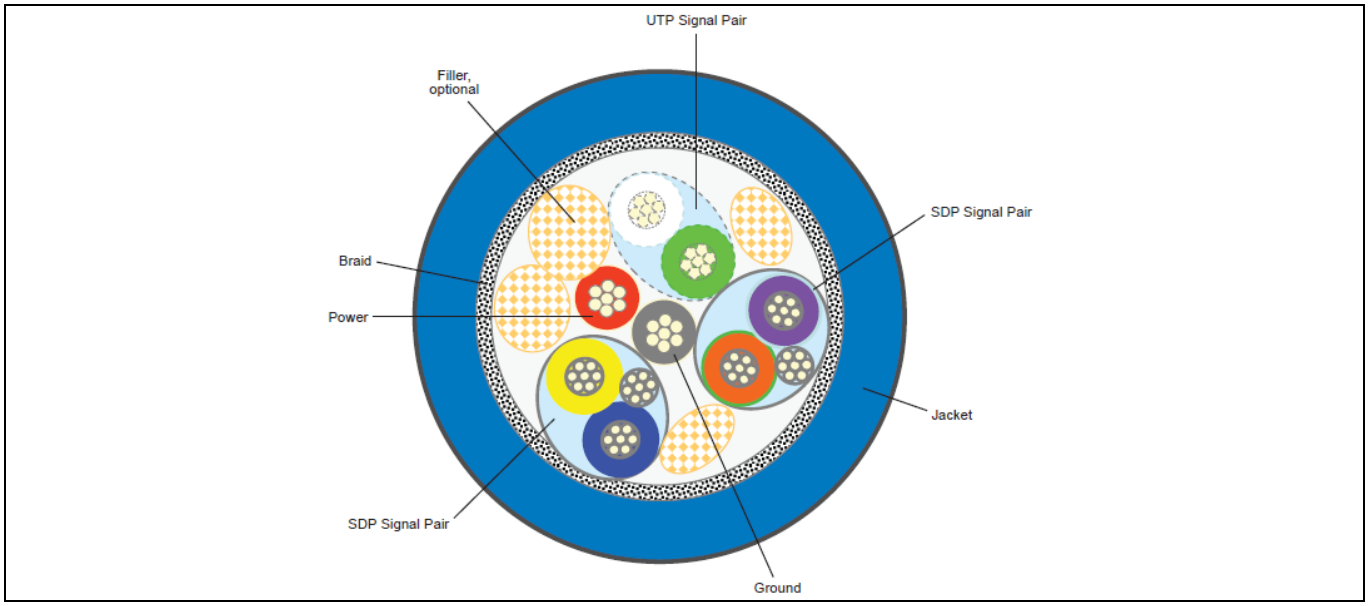


Figure 34 USB 3.0 线缆架构

参考源：**http://usb.org**

Table 4 USB 3.0 线缆说明

名称	说明	颜色
电源	VBUS	红
UTP_D-	USB 2.0 D-	白
UTP_D+	USB 2.0 D+	绿
接地	电源的接地线	黑
SDP1-	屏蔽差分对 1	蓝
SDP1+		黄
SDP1_Drain	SDP1 的接地线	
SDP2-	屏蔽差分对 2	紫
SDP2+		橘
SDP2_Drain	SDP2 的接地线	

USB 3.0 规格中也定义了下面各连接器：

- **USB 3.0 Standard-A 插头和插座**

SuperSpeed 标准中将 USB 3.0 Standard-A 连接器定义为主连接器。该连接器是基于 USB 2.0 Standard-A 连接器的设计，另外还添加了超高速信号。USB 3.0 Standard-A 插座可与 USB 3.0 Standard-A 插头或 USB 2.0 Standard-A 插头连接。USB 3.0 Standard-A 连接器使用唯一的颜色，易于识别。**Figure 34** 显示了所推荐的编码颜色。

- **USB 3.0 Standard-B 插头和插座**

  USB 3.0 Standard-B 连接器用于比较大的固定设备，如外部硬盘驱动器和打印机。USB 3.0 Standard-B 插座可与 USB 3.0 Standard-B 插头或 USB 2.0 Standard-B 插头连接。不允许将 USB 3.0 Standard-B 插头插入 USB 2.0 Standard-B 插座。

- **USB 3.0 Powered-B 插头与插座**

  通过 USB 3.0 Powered-B 连接器，USB 3.0 设备可向 USB 适配器供电，而不需使用外部电源。该连接器的尺寸与 USB 3.0 Standard-B 连接器相同，但添加了两个引脚：一个用于电源供电 (DPWR)，另一个用于接地 (DGND)。

- **USB 3.0 Micro-B 插头与插座**

  USB 3.0 Micro-B 连接器用于体积较小的手提设备。

- **USB 3.0 Micro-AB 和 USB 3.0 Micro-A 连接器**

  USB 3.0 Micro-AB 插座类似于 USB 3.0 Micro-B 插座，只有外形设计是不同的。它可与 USB 3.0 Micro-A 插头、USB 3.0 Micro-B 插头、USB 2.0 Micro-A 插头以及 USB 2.0 Micro-B 插头连接。USB 3.0 Micro-AB 插座只能使用于 OTG 产品。该产品可作为主机或设备使用。USB 3.0 Micro-AB 插座不可用于其他用途。

  USB 3.0 Micro-A 插头类似于 USB 3.0 Micro-B 插头，只有外形和 ID 引脚是不同的。USB 3.0 Micro-A 插头、USB 3.0 Micro-AB 插座、USB 3.0 Micro-B 插座和插头都属于 USB 3.0 Micro 连接器系列。这些接口只是外形不同。与 USB 2.0 Micro-A 插头相同，USB 3.0 Micro-A 插头只用于 OTG 应用。

10.3 USB 3.0 与 2.0

USB 3.0 采用双总线架构，支持 USB 2.0 和 3.0。下表显示的是 USB 3.0 和 USB 2.0 的主要区别。

Table 5 USB 3.0 与 USB 2.0 的区别

特性	USB 2.0	USB 3.0
数据传输速率	480 Mbits/s (高速) 12 Mbits/s (全速) 1.5 Mbits/s (低速)	5.0 Gbits/s (超速) 480 Mbits/s (高速) 12 Mbits/s (全速) 1.5 Mbits/s (低速)
数据传输	半双工 两线差分信号	双单工 四线差分信号
线缆信号数量	共有四个信号： - 两个用于 USB 2.0 (D、D–) - 两个分别用于 VBUS 和 GND	共有九个信号： - 四个用于超高速数据 - 两个用于 USB 2.0 (D、D–) - 三个分别用于 VBUS 和 GND
总线数据传输协议	直接连接主机协议 轮询数据流 数据包传输到所有下行设备 没有数据流复用	直接连接主机协议 异步通知 数据包只能传输到目的设备 可批量传输多个数据流
电源管理	两种模式 - 活动 - 挂起	四种模式 - 活动 (U0) - 闲置、快速 (U1) - 闲置、低速 (U2) - 挂起、低速 (U3)

附录 A. USB 3.0 概述

特性	USB 2.0	USB 3.0
总线供电	低功耗设备：100 mA 高功耗设备：500 mA	低功耗设备：150 mA 高功耗设备：900 mA
端口状态	端口硬件检测连接事件。 系统软件使用端口指令将端口切换到使能状态。	端口硬件检测连接事件，并将端口状态转换为操作状态，准备进行超高速数据通信。
线缆的最大长度	5 米	基于电气规格。 实际上是 3 米长的 26 AWG 类铜线
数据传输类型	四种数据传输类型：控制传输、批量传输、中断传输以及同步传输。	支持 USB 2.0 的四种数据传输类型，并具有超高速传输功能。 批量传输具有数据流传输功能。

11 附录 B. 在 Windows 上安装 EZ-USB™ FX3 DVK 驱动程序

如果在 Windows 计算机上未安装 EZ-USB™ FX3 DVK 套件，那么第一次将 DVK 与计算机相连时，您会看到下面信息。

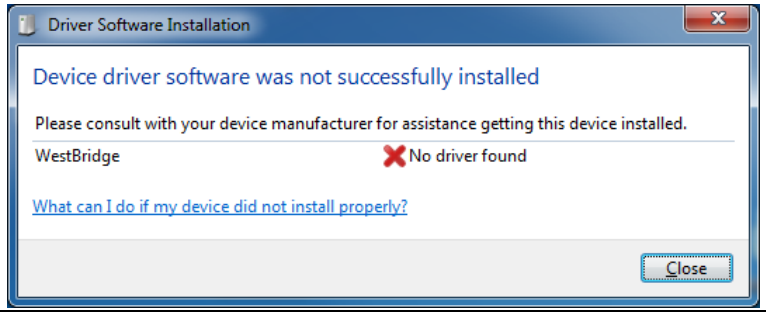


Figure 35 未安装驱动程序时出现的 Windows 信息

关闭信息框并导航到 Windows 设备管理器。要实现该操作，请单击 Windows 窗口中的 **Start** 按键，然后在右栏上右击 **Computer** 项，然后点击 **Properties** 来打开系统信息。然后，在左栏上端点击 **Device Manager** 项。

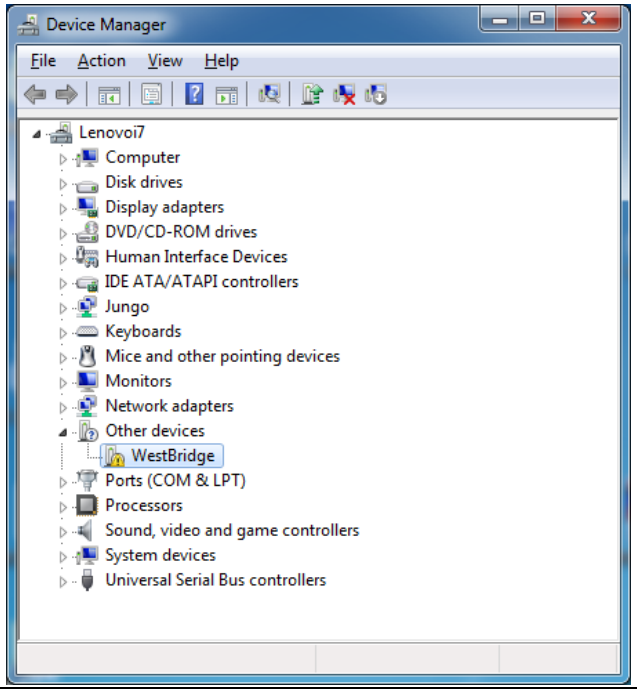


Figure 36 Windows 设备管理器

右击 **WestBridge** 并选择 **Browse my computer for driver software** 项。在 64 位 Windows 7 电脑上，该驱动程序的位置为：C:\Program Files (x86)\Cypress\EZ-USB FX3 SDK\1.3\driver\bin\win7\x64。您的 SDK 版本可能高于 1.2。

设备管理器要删除 WestBridge 入口并将 SDK 板识别为 USB Bootloader (顶端入口)。

附录 B. 在 Windows 上安装 EZ-USB™ FX3 DVK 驱动程序

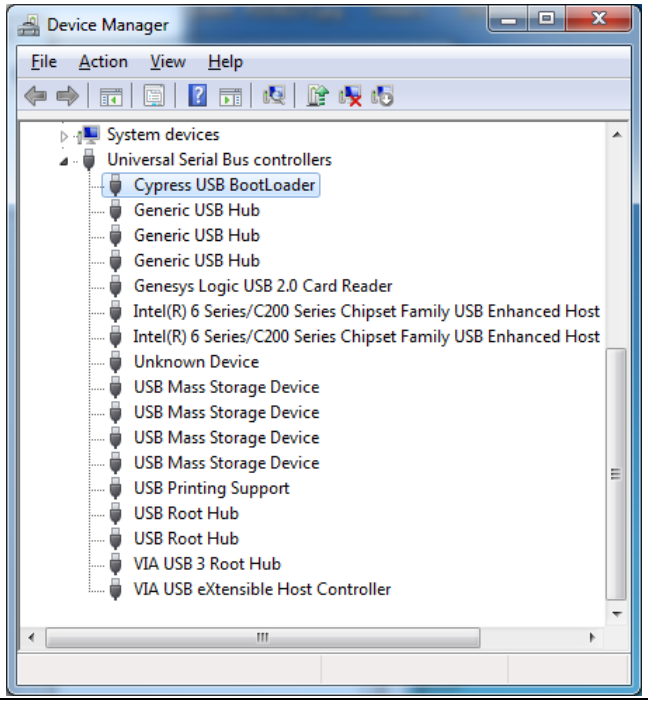


Figure 37　驱动程序安装后 USB BootLoader 在设备管理器上的显示

12 附录 C. EZ-USB™ FX3 DVK 简介

英飞凌的 EZ-USB™ FX3 DVK 提供了您需要使用的硬件。PCB 提供了 EZ-USB™ FX3 所需的时钟和电源，以及可配置 I/O 电压。DVK 带有用于外设接口的高速接口，并且支持各种启动模式。通过 I^2C EEPROM 套接字，可以编程并测试从 I^2C EEPROM 启动的操作。板上 SPI Flash 芯片支持通过 SPI 总线进行的编程和启动操作。SDK 提供的英飞凌示例项目包含了固件，用于通过使用 DVK 开发板编程这些设备。如果您没有 DVK，可以通过点击**此处**获取。

两个很重要的硬件部分 (位于 EZ-USB™ FX3 DVK 开发板外) 分别是 USB 3.0 主机 (通常为 PC) 以及与 GPIF II 接口相连的外部设备 (如图像传感器或 FPGA)。有关使用 EZ-USB™ FX3 DVK 的详细信息，请参见 DVK **用户指南**。**Figure 38** 显示了 EZ-USB™ FX3 DVK 开发板的重点区域。

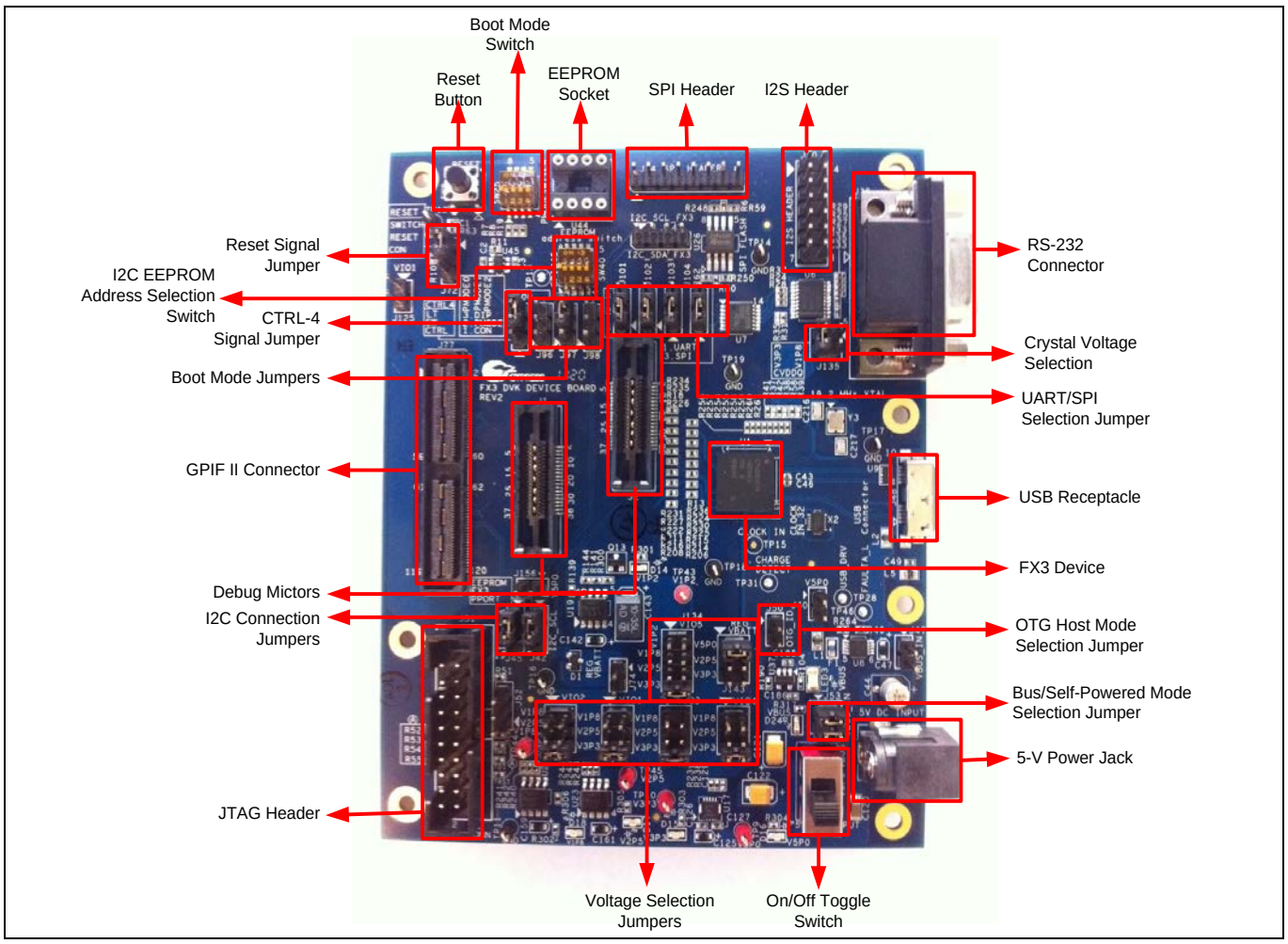


Figure 38 EZ-USB™ FX3 DVK 电路板

附录 C. EZ-USB™ FX3 DVK 简介

Table 6 显示了 DVK 开发板上的重要跳线器和它们的默认设置情况。

Table 6 EZ-USB™ FX3 DVK 开发板跳线器

跳线器/开关	使用跳线器短接的引脚 (默认设置)	函数 (默认设置)
J101	1 和 2	GPIO_46=UART_RTS
J102	1 和 2	GPIO_47=UART_CTS
J103	1 和 2	GPIO_48=UART_TX
J104	1 和 2	GPIO_49=UART_RX
J136	3 和 4	VIO1 (3.3 V)
J144	3 和 4	VIO2 (3.3 V)
J145	3 和 4	VIO3 (3.3 V)
J146	3 和 4	VIO4 (3.3 V)
J134	4 和 5	VIO5 (3.3 V)
J135	2 和 3	CVDDQ (3.3 V)
J143	1 和 6	VBATT (2.5 V)
J96 & SW25	2 和 3	使用 SW25 选择 PMODE0 引脚状态 (ON/OFF)。SW25.1=OFF
J97 & SW25	2 和 3	使用 SW25 选择 PMODE0 引脚状态 (ON/OFF)。SW25.2=OFF
J98	1 和 2	PMODE2 引脚悬空
J72	1 和 2	RESET
J42	未安装	GPIO_58=I2C_SCL
J45	未安装	GPIO_59=I2C_SDA
J100	1 和 2	GPIO_21 = CTL4

12.1 EZ-USB™ FX3 DVK 的 JTAG 调试器

Segger J-Link 探测器是 EZ-USB™ FX3 SDK 中首选的 JTAG 探测器。该探测器 (和 Segger J-Link Arm GDB 服务器) 用于进行调试。Eclipse IDE 同 J-link GDB 服务器相连，用于调试您的固件。为使 Eclipse 能够同 GDB 服务器工作，您需要为 J-link 创建一个调试配置。有关的详细信息，请参考 **EZ-USB™ FX3 编程者手册**中第 12 章的内容。

进行调试时，您可能需要使用其它的 JTAG 探测器 (如 Olimex Arm-USB-OCD 探测器)。在这种情况下，OpenOCD 工具可用于 GDB 连接。**EZ-USB™ FX3 编程者手册**中第 12 章对此进行了详细的介绍。

文档修订记录

版本	提交日期	变更说明
**	2013 -08-12	本文档版本号为 Rev**，译自英文版 001-75705 Rev*B。
*A	2015-08-03	本文档版本号为 Rev*A，译自英文版 001-75705 Rev*E。
*B	2017-04-19	更新徽标和版权。
*C	2017-10-10	没有技术更新。 完成日落评论。
*D	2018-10-11	本文档版本号为 Rev*D，译自英文版 001-75705 Rev*I。
*E	2022-05-19	本文档版本号为 Rev*E，译自英文版 001-75705 Rev. *J。

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-05-19
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2022 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?
Go to www.infineon.com/support

Document reference
001-88777 Rev. *E

重要提示

本文档所提供的任何信息**绝不应当**被视为针对任何条件或者品质而做出的保证（质量保证）。英飞凌对于本文档中所提及的任何事例、提示或者任何特定数值及/或任何关于产品应用方面的信息均在此明确声明其不承担任何保证或者责任，包括但不限于其不侵犯任何第三方知识产权的保证均在此排除。

此外，本文档所提供的任何信息均取决于客户履行本文档所载明的义务和客户遵守适用于客户产品以及与客户对于英飞凌产品的应用所相关的任何法律要求、规范和标准。
本文档所含的数据仅供经过专业技术培训的人员使用。客户自身的技术部门有义务对于产品是否适宜于其预期的应用和针对该等应用而言本文档中所提供的信息是否充分自行予以评估。

如需产品、技术、交付条款和条件以及价格等进一步信息，请向离您最近的英飞凌科技办公室接洽(**www.infineon.com**)。

警告事项

由于技术所需产品可能含有危险物质。如需了解该等物质的类型，请向离您最近的英飞凌科技办公室接洽。

除非由经英飞凌科技授权代表签署的书面文件中做出另行明确批准的情况外，英飞凌科技的产品不应当被用于任何一项一旦产品失效或者产品使用的后果可被合理地预料到可能导致人身伤害的任何应用领域。