

FX2LP™ 入门手册

作者: Rama Sai Krishna.V

关联项目: 有

关联芯片系列: CY7C68013A/14A/15A/16A/53

软件版本: 无

相关文档: [请点击此处](#)。[更多代码例程?](#)获取更对关于 FX2LP 的例程请访问 [USB High-Speed Code Examples webpage](#)。

你在寻找 USB3.0 外设控制器吗?

USB3.0 系列请访问 [USB 3.0 Product Family webpage](#)。

AN65209 介绍了 EZ-USB® FX2LP™ USB 2.0 器件控制器。本应用笔记可帮助您为 FX2LP 编译项目，并探索它的各种开发工具；然后向您介绍了相应文档，这样能够深入了解 FX2LP。

目录

简介	1
Bulkloop_FX2LP 项目	1
USB 2.0	2
FX2LP 简介	3
FX2LP 特性	3
串行接口	5
CPU 及存储器	5
FX2LP 的示例应用	6
赛普拉斯设计资源	6
FX2LP 开发工具	8
FX2LP 开发板	9
运行 Bulkloop 演示	9
FX2LP 固件开发	12
GPIF Designer (GPIF 设计程序)	14
Windows 端	15
赛普拉斯 USB 驱动程序	15
赛普拉斯库	16
汇总	17
附录 A: FX2LP 开发套件 (DVK)	18
固件示例项目	19
附录 B: 赛普拉斯的额外 USB 高速器件	20
附录 C: 第三方开发套件和软件开发套件和 SDK	21
第三方软件开发套件 SDK	21
附录 D: 应用笔记和参考设计	22
应用笔记	22
附录 E: 向 .inf 文件添加自定义 VID 和 PID	24

简介

赛普拉斯 EZ-USB FX2LP (以下简称为 FX2LP) 是一个灵活的 USB 2.0 外设控制器，设计它来处理 USB 2.0 的最大带宽。要充分利用 USB 2.0 480 Mbps 的信号速率，需要 FX2LP 包含专门的硬件，以缓冲 USB 数据并无缝连接到各种高带宽的外部设备，如 MCU、ASIC 和 FPGA。简要介绍了 USB 2.0 后，本应用笔记还会对 FX2LP 特性进行说明，这些特性有助于提高其吞吐量。

只有支持 IC 的工具性能优良时，它才能达到最佳性能，因此本应用笔记的剩余部分会介绍 [FX2LP 开发套件 \(DVK\)](#) 和 FX2LP 固件与软件开发工具。一个同伴 zip 文件包含了具有代码示例的文件夹以及本应用笔记中详细介绍的一个应用。简单地说，它们分别为：

Bulkloop_FX2LP 项目

本应用笔记附带的 Bulkloop 项目显示了如何通过构造 FX2LP 固件来创建 USB 器件。大部分 USB 请求处理是由赛普拉斯提供的 USB 固件框架执行的，该框架具有仅提供给特定应用需求的用户代码。*bulkloop.c* 文件包含了完整的 USB 器件模板，该模板可作为自定义应用的基础使用。

赛普拉斯工具包括 FX2LP CPU (8051) 的 Keil uVision2 免费评估版。Keil uVision2 集成开发环境 (IDE) 编译用户代码和 USB 固件框架来产生 hex 文件。该 hex 文件需要通过已提供的 **USB 控制中心** 应用下载，并通过所提供的 **Bulkloop_VCS** 应用来测试。

作为固件工具讨论的一部分，介绍了 GPIF Designer 来显示如何使用图形化波形输入来创建自定义接口（请参见 [GPIF Designer \(GPIF 设计程序\)](#) 部分的内容）。

VS_Control Center

它是一个使用 Visual C#编写的 Microsoft Visual Studio 解决方案，创建一个称为 **USB 控制中心**的应用。通过该应用将 FX2LP 代码（hex 文件）下载到 FX2LP 开发套件内。可执行的文件用于即时使用，而源代码用于参考。它采用了赛普拉斯.NET 库，显示了多个高级函数的使用方法。

Bulkloop_VCS

它是一个使用 Visual C#编写的简单 Windows Visual Studio 应用。它具有连续数据循环和字节计数的功能，用于测量各种 USB 控制卡和计算机的 USB 传输带宽。与 USB 控制中心相同，它们都提供了源代码和可执行的文件。

Windows 驱动程序以及库

赛普拉斯所提供的 Windows 驱动程序用于支持基于 FX2LP 的设计。它支持所有的 USB 传输类型，并能够以二进制的形式与客户产品一起使用。

USB 控制中心和 Bulkloop_VCS 应用使用 **CyUSB.dll** — 赛普拉斯所提供的一种托管 Microsoft .NET 类库。这使得它能够与 Visual Basic.NET、Visual C#、Visual C++（Win Forms）和 Visual J#语言相兼容。该库提供了高级函数的调用方式，这些调用方式可降低驱动程序的复杂性，并提出了一个非常简单的 USB 编程模型。

USB 2.0

USB 用于标准化计算机外设（如键盘、鼠标、打印机、笔式驱动器、硬盘和便携式媒体播放器）的连接。USB 为外设器件提供通信和电源。USB 已经成为 PC 和消费类器件中的占主导地位的连接解决方案。它的流行主要是由于其本身具有的易用性，通过其标准化和强大的基础结构来实现。

1996 年发布的 USB 1.0 规范定义了两种传输速度，用以在那段时间内对通用器件进行分类。低速器件的运行速度为 1.5 Mbps，以支持如键盘和操纵杆等器件。全速器件的运行速度为 12 Mbps，以支持较高带宽的器件（如打印机和磁盘驱动器）。次要的规范修订版 1.1（发布于 1998 年）主要用于解决集线器问题，并已成为第一代广为采用的规范。

USB 2.0 规范发布于 2000 年。它将信号传输速率提高到 480 Mbps，并将该速度命名为高速（Hi-Speed）。USB 2.0 规范与先前的传输速率相兼容。

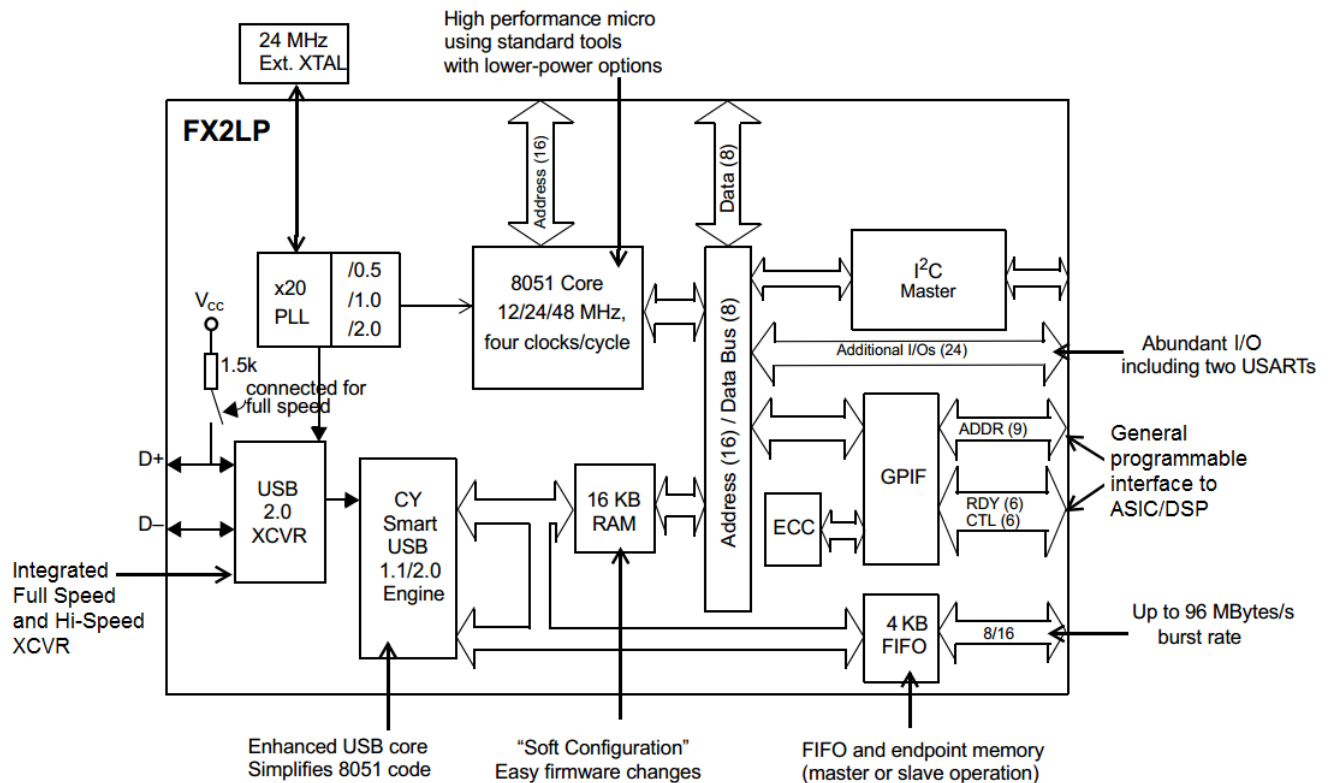
2008 年发布的 USB 3.0 规范将信号速率提高到 5 Gbps（该速度被称为超高速（SuperSpeed））。赛普拉斯 FX3 系列支持该速度。

本应用笔记主要讲述的是在高速和全速下运行的赛普拉斯 FX2LP。[附录 B](#) 列出了赛普拉斯中其他可用的高速器件。赛普拉斯的所有 USB 控制器都列在[这里](#)。有关 USB 2.0 的详细信息，请参考 [USB 101: 通用串行总线 2.0 简介的内容](#)。

FX2LP 简介

FX2LP 在单个芯片上集成了 USB 2.0 收发器、智能串行接口引擎 (SIE)、大的数据缓冲区、增强型 8051 微控制器以及可编程外设接口。图 1 显示的是简化的 FX2LP 框图。

图 1. FX2LP 框图



FX2LP 主要用于在 **USB** 主机与外设器件之间进行数据传输。**FX2LP** 通过使用并行和串行接口提供该连接。在将数据发送到 **USB** 主机前不需要 **FX2LP** **CPU** 修改数据的系统中，**FX2LP** 固件仅需要初始化它的硬件传输单元，这样不需要 **CPU** 的干预仍能够执行高带宽的 **USB** 传输。

可对 FX2LP 进行编程，以枚举为任何 USB 器件类型，从而符合标准 USB 类（如 HID 或大容量存储或自定义器件）。

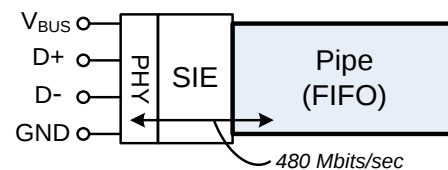
FX2LP 特性

该部分对 FX2LP 的主要特性进行了简要介绍。

USB 接口

使用高速 USB 传输来支持管道的两端。在其中一端上，管道通过 USB 接口以 480 Mbps 的速度接收并发送数据，如图 2 所示。并行接口部分会对管道的另外一端进行相关说明。

图 2。USB 连接到数据管道



要处理 USB 主机端，FX2LP 包括：

- 包含 USB 收发器的物理总线接口 (PHY) 支持高速和全速 USB 传输。
- 智能串行接口引擎 (SIE)。SIE 将 PHY 信号转换为字节。赛普拉斯的所有 USB SIE 添加逻辑用于处理低级

USB 的细节，如：纠错和数据包 ID (PID) 同步，这样可解除编程器执行这些任务。这个智能 SIE 将字节传送到端点 FIFO 存储器内。智能 SIE 还包含逻辑，以便将 FX2LP 枚举为完整的 USB 器件，另外它能把代码加载到其内部 RAM 中。

- 七个端点支持以下全部四个 USB 传输类型：CONTROL（控制传输）、BULK（批量传输）、INTERRUPT（中断传输）以及 ISOCHRONOUS（同步传输）。其中一个端点是 USB 数据的发送器或接收器。
- 大端点缓冲区 (FIFO) 具有多个缓存操作。两个、三个或四个缓存操作允许 USB 传输与外设连接，这样能增加吞吐量。
- 低功耗工作 (FX2LP 中的“LP”)。USB VBUS 线提供 5 V 电压，并为外设器件提供有限的电源。FX2LP 的低功耗允许您创建总线供电的应用。例如，FX2LP 开发套件是总线供电的，无需外部电源。另外，FX2LP 还可使用在自供电的设计中，其中外设器件自己提供电源。

并行接口

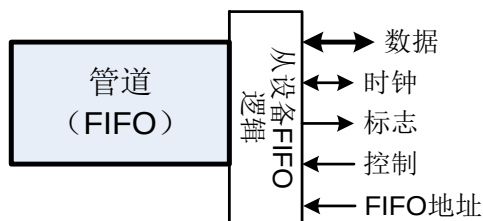
USB 2.0 规范和 FX2LP 缓存通过 USB 接口以高速度将字节传入和传出各端点 FIFO，但这些只是它们工作的一半。管道其他端点将 FIFO 数据传入和传出芯片的速度必须与 USB 传输速率相匹配。FX2LP 包含两个专门设计用于该目的的硬件接口：从设备 FIFO 和通用可编程接口 (GPIF)，如图 1 所示。

注意：除了高速传输逻辑外，FX2LP 8051 还为要求修改 USB 和外设接口间传输数据的应用提供端点 FIFO 的随机访问。

从设备 FIFO

FX2LP 提供的一个从设备 FIFO 接口用于包含 FIFO 控制器（如：MCU、FPGA 或 ASIC）的外部器件（请参见图 3 所示）。

图 3. FX2LP 从设备 FIFO 接口



从设备 FIFO 接口：

- 提供带宽为 8 位或 16 位的数据总线。
- 异步操作（无时钟）或同步操作（有时钟）。

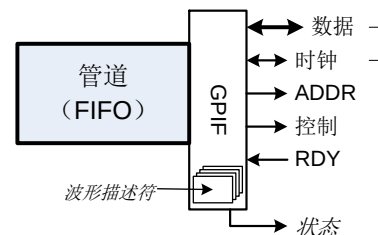
- 接收外部时钟或使用内部 FX2LP 30/48 MHz 时钟（在同步模式中），在 CLKOUT 引脚上提供该时钟。如果外部外设器件根据 CLKOUT 引脚上所提供的时钟工作，那么它不再需要晶体。
- 具有输出标志来表示 FIFO 状态，比如：满和空状态。
- 具有控制输入 OE#、RD# 和 WR#。
- 具有两个地址线，用于选择四个 FX2LP FIFO 中的某一个。
- 当 FIFO 已满或被清空时，会自动启动 USB 传输。
- 具有一个控制信号 (PKETND)，接口器件可以使用它来启动“短数据包”（FIFO 有未满足的数据）。短数据包经常作为长 USB 传输的最后数据包出现。

在不需要 FX2LP 的数据处理的系统中，FX2LP 从设备 FIFO 接口需要较小的固件，用以选择从设备 FIFO 接口和配置标志来显示 FIFO 的状态。FX2LP 数据手册提供了从设备 FIFO 接口的波形和时序值。

通用可编程接口

并非所有外部控制器都为连接到 FIFO 的目的而设计。因此，FX2LP 提供了一个高速接口 GPIF（如图 4 所示），该接口无需额外的胶合逻辑仍能直接连接到通用接口（如：磁盘驱动器、FPGA 和 ASIC）。GPIF 的核心是一个状态机——您要使用 GPIF Designer 开发波形来控制状态机。GPIF 由四个波形描述符中的一个来驱动，其中波形描述符是包含所有波形信息的数据结构。GPIF Designer 有助于设计员不要了解描述符格式。GPIF Designer 使用图形化波形输入来创建 C 语言的源文件，能将该文件包含在 FX2LP 项目中。

图 4. FX2LP GPIF 接口



GPIF（请参见 GPIF Designer（GPIF 设计程序）部分）：

- 提供了带宽为 8 位或 16 位的数据总线。
- 同步操作（有时钟）。
- 接收外部时钟或使用内部 FX2LP 30-/48-MHz 时钟（在同步模式中），在器件引脚上提供该时钟。
- 提供了九个地址输出。可初始化地址，并按时钟进行递增。
- 提供了六个控制 (CTL) 和就绪 (RDY) 信号。并且能够按时钟对六个控制输出进行编程。举一个简单的例子，可以使用这些控制信号来创建 RD# 和 WR# 选

通。其中，可对其持续时间和极性进行编程。在 GPIF 状态机中，可以测试六个就绪输入，以便读取状态和执行接口同步。三个状态输出（GSTATE [2:0]）用于显示 GPIF 状态，这样对使用逻辑分析仪调试非常有用。

串行接口

- 除 USB 和外设接口外，FX2LP 还包含：
- I²C 主设备，仅在频率为 100 或 400 kHz 时工作。
- 两个标准的 8051 USART。它们是具有同步选项的标准 UART。用于 USART 接口的引脚是单独的 I/O 引脚，但不能与 GPIO 引脚复用。
- 多达 40 个 GPIO。这些引脚用途广泛，可作为 GPIO 引脚或支持从设备 FIFO 或 GPIF 接口的引脚。[封装选择部分](#)对这些接口选项进行了说明。

CPU 及存储器

8051

- FX2LP 具有一个 8051 内核，该内核包含了两个 USART、三个计数器/定时器和一个增强型中断系统。该内核可以使用频率为 48 MHz、24 MHz 或 12 MHz 的时钟。16 KB 的片上代码/数据 RAM 支持 CPU。
- 增强型中断系统使用了一个“自动向量”机制来根据需要服务的 USB 活动自动调用 27 个 USB 中断服务子程序（ISR）中的一个。自动递增指针的硬件（“自动指针”）会加快模块传输。

启动选项

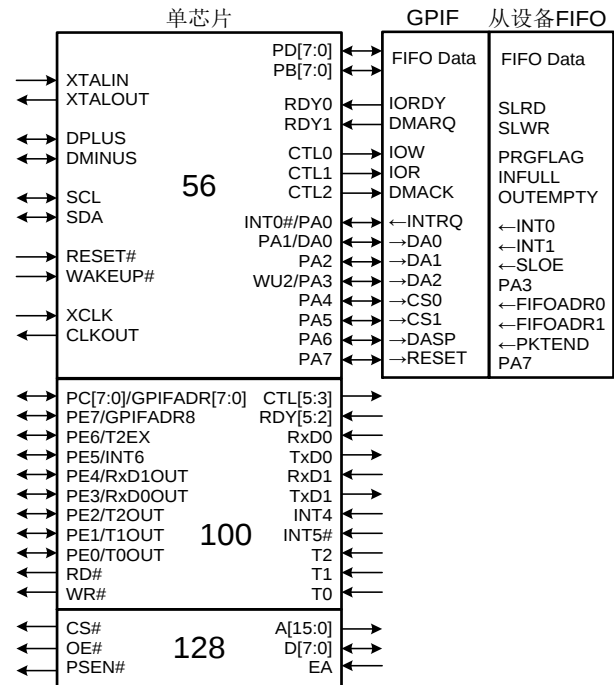
- FX2LP 使用 RAM 来存储程序。FX2LP 具有以下启动选项：
 - USB 启动
 - I²C 启动
 - 从外部并行存储器启动
- 当连接到 USB 接口时，其智能 SIE 枚举为可将程序代码加载到其内部 RAM 中的 USB Bootloader。完成加载代码后，FX2LP 自己会断开与 USB 的连接，并使用下载的代码定义的器件名称，立即重新连接至 USB。该过程被称为 Re-Numeration™（重新枚举）。
- 上电时，FX2LP 编程 RAM 可以通过外部串行 EEPROM 加载。欲了解更多有关启动选项的信息，请参考 [AN50963 — EZ-USB® FX1™/FX2LP™启动选项](#) 中介绍的内容。

封装选择

FX2LP 拥有三种不同封装，如图 5 所示：

- 56 引脚 SSOP、QFN 和 VFBGA
- 100 引脚 TQFP 封装
- 128 引脚 TQFP 封装

图 5. 三种 FX2LP 封装选择



56 引脚封装是 FX2LP 的最低成本版本。56 引脚模块的左边沿上的信号（如图 5 所示）是所有系列中封装版本的通用信号。所有封装版本均可以使用以下三种模式：单芯片、GPIF 和从设备 FIFO。这些模式定义了图 5 所示的 56 引脚模块右边沿上的信号。8051 使用内部寄存器选择接口模式。单芯片模式是加电时的默认配置。

100 引脚的封装通过添加 44 个引脚将功能添加到 56 引脚的封装中：

- 两个额外的 8 位 I/O 端口：即 PORTC 和 PORTE。
- 七个额外的 GPIF 控制（CTL）和就绪（RDY）信号。
- 九个非复用的控制信号（两个 UART，三个定时器输入，INT4 和 INT5#）。
- 在 PORTE 端口上复用的八个额外控制信号。
- 在 PORTC(8)和 PORTE(1)上复用的九个 GPIF 地址线
- 可作为 PORTC 端口的读写选通的 RD#和 WR#信号。

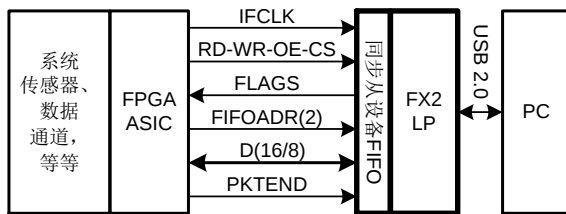
128 引脚的封装负责添加 8051 地址和数据总线以及它们的控制信号。添加的这些引脚允许 FX2LP 访问外部 8051 存储器。在 FX2LP 开发板上使用该封装。

FX2LP 的示例应用

使用从设备 FIFO 连接 FPGA/ASIC

在图 6 中，FPGA 或 ASIC 包含了直接连接至 FX2LP 从设备 FIFO 引脚的 FIFO 控制器。FPGA/ASIC 还连接到特定于某些应用的硬件，如：数据记录器或图像传感器。虽然使用了 IFCLK 信号来显示同步 FIFO，但 FX2LP 仍支持异步（无时钟）FIFO 接口。

图 6. FPGA/ASIC 将 FX2LP 视为 FIFO



更多参考资料

有关更多 FX2LP 从设备 FIFO 接口的详细信息，请参考 [AN61345 — 使用 EZ-USB FX2LP™从设备 FIFO 接口进行设计](#)，用以将 FX2LP™与 FPGA 连接，在该资料内提供了详细的设计实例。

某些接口芯片无需修改便能够提供所需的 FIFO 接口信号。例如，MPEG 解码器可以将其信号映射到 FX2LP FIFO，如下表所示：

表 1. MPEG 解码器与 FX2LP 连接

MPEG 解码器信号	FX2LP 信号
MPEG_CLK	IFCLK
MPEG_SYNC	PKTEND#
MPEG_VALID	SLWR#
D[7:0]	FD[7:0]
外部调谐器控制	I²C 总线

更多参考资料

赛普拉斯为“TV Dongle”提供了一个参考设计（记录在 [FX2LP DMB-T / H TV Dongle 参考设计](#)中）。

如果不能将外部芯片正确映射到 FX2LP 从设备 FIFO 信号（未根据表 1 进行），则无需其他外部逻辑并能够对 FX2LP GPIF 进行编程，使之同所需信号相匹配。

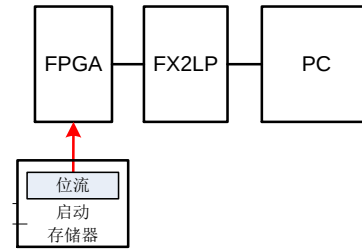
从 FX2LP 中启动 FPGA

FX2LP 使用于有效的集成系统。图 7 和图 8 说明的是 FX2LP 可以保存外部逻辑的另一种方式。

赛普拉斯设计资源

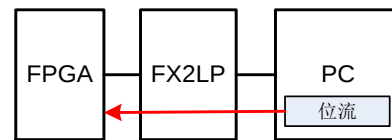
赛普拉斯 FX2LP 设计资源包括数据手册、应用笔记、评估套件、参考设计、固件示例和软件工具。表 2 中总结了各种资源。

图 7.从外部存储器中启动 FPGA



上电时，FPGA 使用外部存储器来启动它的配置位流（图 7）。

图 8. 通过 FX2LP 启动 FPGA



可以配置 FX2LP，以从 PC 中加载 FPGA 位流文件，如图 8 所示。这样便不需要启动存储器，并使 FPGA 可以从 PC 端更新。

更多参考资料

有关使用 Xilinx Spartan-3E FPGA 执行示例，请参考 [AN63620 — 通过 USB（使用 EZ-USB FX2LP™）配置 Xilinx Spartan-3E FPGA](#)。

表 2. FX2LP 资源汇总

设计	可用资源	资源所在的位置
硬件	开发电路板 – 原理图、电路板文件和文档	开发套件（DVK）原理图 电路板文件位于已安装的 FX2LP DVK 的文件夹中 DVK 用户指南 DVK 快速入门指南
	硬件设计指南包括：有关晶振的建议、电源的去耦电容以及 PCB 布局	应用笔记 — AN15456
	IBIS 模型	http://www.cypress.com/?id=193&rtID=114
FX2LP 固件	Keil IDE 的免费版（代码大小最大为 4 KB）	已安装的 FX2LPDVK 的文件夹中
	固件示例	
	在使用 FPGA 的完整设计中的同步从设备 FIFO 固件示例	应用笔记 — AN61345
固件调试	Keil 调试器环境的设置、使用和故障排除	应用笔记 — AN42499
	FX1/FX2LP 固件的串行（UART）端口调试	应用笔记 — AN58009
主机 PC 软件	USB 2.0 驱动程序 – cyusb.sys	位于已安装的 Suite USB 的文件夹中。该 Suite USB 安装文件（.exe）同样位于已安装的 FX2LP DVK 的文件夹中。 同样， Mac OS 的 Suite USB 位于该文件夹中。使用 Linux 的 FX3 SDK 来获得类似于 Linux 平台控制中心的主机应用。
	主机应用示例 — 控制中心和 Streamer 应用	
GPIF 接口设计	“GPIF Designer” 工具使你能够设计某个 GPIF 波形信号，并生成集成到固件中的代码。	位于已安装的 GPIF Designer 的文件夹中。 GPIF Designer 安装文件（.exe）也位于已安装的 FX2LP DVK 的文件夹中。
	流行的 GPIF 实现示例	应用笔记 — AN57322 ：通过使用 GPIF，SRAM 与 FX2LP 相连接 AN66806 – EZ-USB® FX2LP™ GPIF 设计指南 AN63787 — 使用 8 位异步接口配置 EZ-USB®FX2LP™ GPIF 和从属 FIFO 的示例
	有关 GPIF 和使用工具指导的文档	“GPIF Designer” 的用户指南，“GPIF Designer” 工具附带的
其他资料		
FX2LP 数据手册	http://www.cypress.com/?rtID=38801	
FX2LP 技术参考手册	http://www.cypress.com/?rtID=38232	
应用笔记	http://www.cypress.com/?id=193&rtID=76	
参考设计	http://www.cypress.com/?id=193&rtID=201	
知识库文章	http://www.cypress.com/?id=193&rtID=118	

有关 USB 2.0 的材料	http://www.beyondlogic.org/usbnutshell/usb1.shtml AN57294 — USB 101: 通用串行总线 2.0 简介
第三方开发套件	http://www.ztex.de/usb-fpga-1 http://www.opalkelly.com/products/xem6010/ “附录 C: 第三方开发套件和软件开发套件和 SDK”介绍这些开发套件的特性。

FX2LP 开发工具

图 9 显示的是与开发和测试 FX2LP 固件相关的步骤。

图 9. 开发和测试固件的步骤

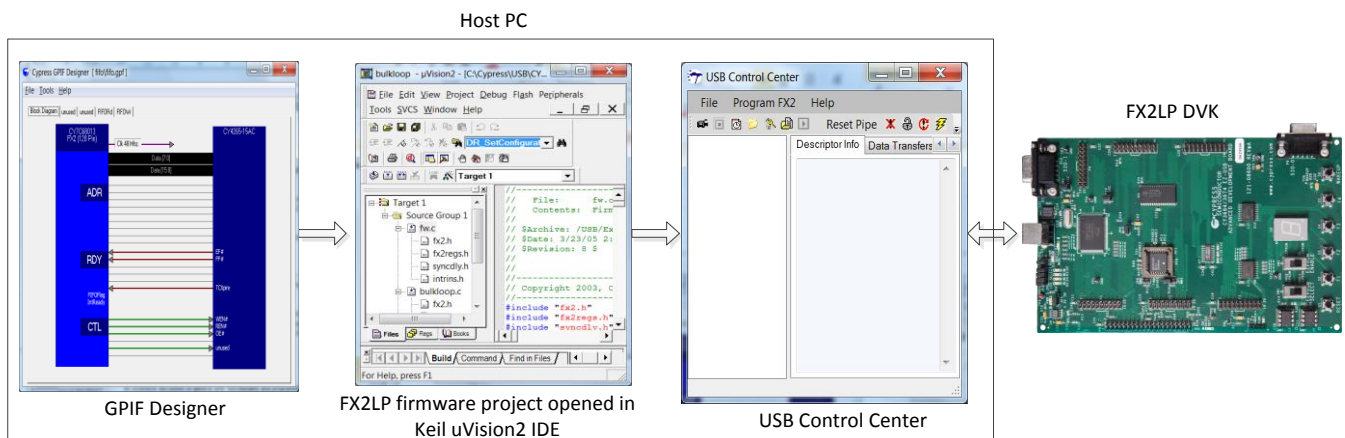
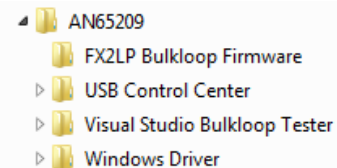


图 10. 应用笔记文件夹



1. 使用 GPIF Designer 开发接口波形，从而实现同连接到 FX2LP GPIF 的外设器件进行通信。导出.c 文件，并将.c 文件集成到 FX2LP 固件项目中。如果未使用 GPIF 接口，便不再需要执行该步骤。在 GPIF Designer 一节中详细介绍了使用 GPIF Designer 的信息。
2. 使用赛普拉斯提供的固件框架来开发 FX2LP 固件。集成第一步所导出的 '.c' 文件。使用 Keil uVision2 IDE 编译固件项目。该步骤将生成 .hex 和 .iic 文件。 .hex 文件用于编程 FX2LP 的 RAM，而 .iic 文件用于编程连接至 FX2LP 的 EEPROM。FX2LP 固件开发一节介绍了 FX2LP 固件框架和编译 FX2LP 固件的各个步骤。
3. 使用控制中心应用来编程 FX2LP 的 RAM 或连接至 FX2LP 的 EEPROM。使用 USB 控制中心一节介绍了控制中心应用的用途。
4. 通过 FX2LP DVK 提供的便利验证固件的功能。FX2LP 开发板部分和附录 A: FX2LP 开发套件 (DVK) 一节中详细介绍 FX2LP DVK 的信息。

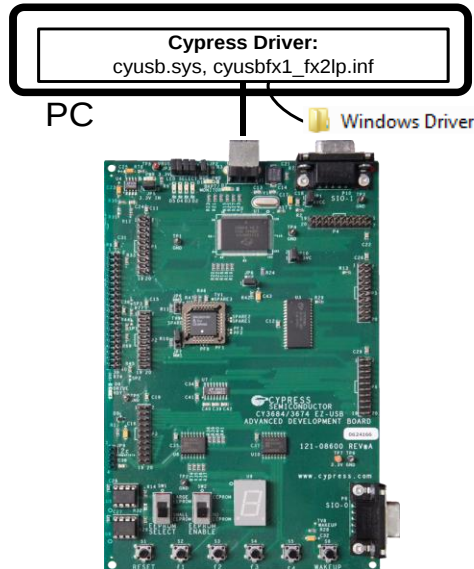
本应用笔记附带的 zip 文件包含以下目录结构：

注意：解压后的应用笔记文件夹包含了赛普拉斯各种代码示例和可供下载的开发工具的副本。为方便使用，将它们放在一个文件夹中。更多有关开发的资料，请访问赛普拉斯网站，以获取最新版的文件版本。

本节提供了创建名称为 **bulkloop** 的 FX2LP 固件示例并将其下载到 FX2LP 开发套件中的用法。同时，它还提供各种工具，这些工具通过使用 FX2LP 使任何 USB 外设器件容易设计。如果您未有 FX2LP DVK，指导仍提供可用 FX2LP 芯片和工具的概述。本节向您介绍了如何创建和运行 Windows 应用，以测试 **bulkloop** 设计。

FX2LP 开发板

图 11. FX2LP 开发套件和 Windows 驱动程序



有关 FX2LP 开发套件的信息，请访问 <http://www.cypress.com/?rID=14321> 网站。

该套件附带（提供免费下载）的是开发软件封装，该封装包含了 Keil uVision2 IDE（限制于大小为 4 千字节的目标）免费版本和 C 编译器。附录 A 提供了关于 DVK 的更详细信息。

对于初次将电路板连接到 PC，请按照下面步骤进行：

1. 准备 FX2LP 电路板跳线器，如表 3 所示。

表 3. EZ-USB FX2LP 板跳线器设置

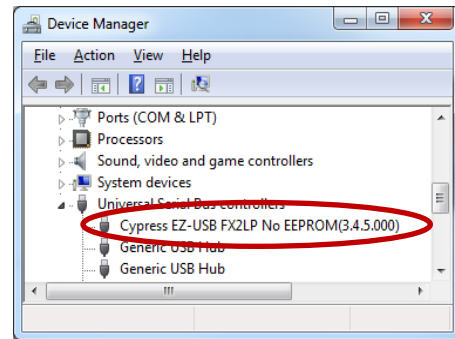
跳线器	状态	用途
6、7	拔出	配置存储器，用于开发
2	插入	通过 USB 连接器供电给电路板
1、5、10	插入	局部 3.3 V 电源
3	插入	插入所有 4 个跳线器 — 激活 4 个 LED（D2-D5）
8	插入或拔出	未使用（用于远程唤醒测试）

2. 在电路板上的左下角，将 EEPROM ENABLE 滑动开关移动到“NO EEPROM”（下面）的位置。这样会使 FX2LP 芯片自己枚举为代码加载器。可将另一个滑动开关（EEPROM SELECT）移动到任何位置。
3. 将 FX2LP 开发板插入到 PC USB 端口。如果这是第一次执行该操作，您会看到弹出消息，提示您安装 USB

驱动程序。浏览到应用笔记驱动程序文件夹，然后选择与您的 Windows 操作系统相对应的子文件夹。

您可以通过查看 Windows Device Manager（Windows 设备管理程序）来确认是否成功安装好了驱动程序。

图 12. FX2LP 开发板驱动程序（Bootloader）已被安装



安装该驱动程序后，与 FX2LP 开发板进行通信的任何 Windows 应用会确认它。

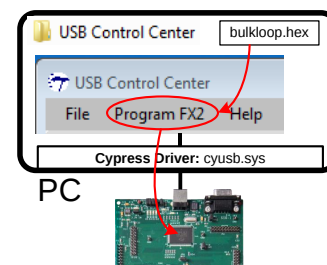
介绍如何创建 **bulkloop.hex** 文件前，先向您介绍如何使用两个 Windows 应用下载和测试 FX2LP 固件的下面两个部分。

运行 Bulkloop 演示

使用 USB 控制中心

图 13 的文件夹包含 Microsoft Visual C#解决方案，用于创建名字为 **USB 控制中心** 的 Windows 应用。该应用用于将 FX2LP 固件（hex 文件）下载到 FX2LP 芯片内。完成下载固件后，FX2LP 开发板会自动断开 USB 的连接，然后作为通过下载的 hex 文件定义的器件名称重新连接到 USB（ReNumeration™）。如果加载的 hex 文件具有自定义供应商 ID（VID）和产品 ID（PID），那么你需要将这些值添加到赛普拉斯的.inf 文件内。

图 13. USB 控制中心将 Hex 文件加载到 FX2LP 开发板上。



更多参考资料

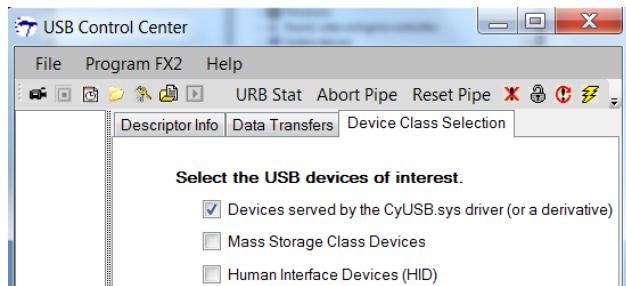
更多有关添加自定义 VID 和 PID 到.inf 文件内的信息，请参考附录 E：向.inf 文件添加自定义 VID 和 PID。

USB 控制中心还有一个“Data Transfer”（数据传输）选项卡，通过它您可以启动 USB 传输流入和流出 FX2LP DVK，以测试 **bulkloop** 应用或你编写的任何其他固件。

附带的 zip 文件包含了 Visual Studio 解决方案（**AN65209\USB 控制中心**）以及它自己的编译后的二进制。要尝试进行上述操作，请按照下面各步骤：

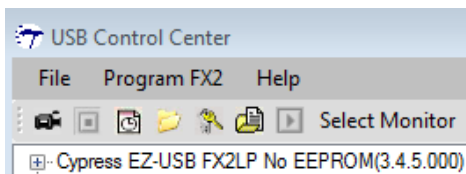
1. 双击在 Debug（调试）或 Release（释放）文件夹中的 **CyControl.exe** 文件（**AN65209\USB Control Center\bin**）。

在左侧屏幕上，您可看到 FX2LP 电路板以及与其相连的其他 USB 器件。如果需要简化该显示屏，使之仅显示 FX2LP 开发板，则在右侧屏幕上点击 **Device Class Selection** 选项卡，并取消勾选所有选项（除 **Devices served by the CyUSB.sys driver (or a derivative)** 选项外）。



然后，左侧屏幕上将显示类似于图 14 的图片。

图 14. USB 控制中心查找 FX2LP 开发板。

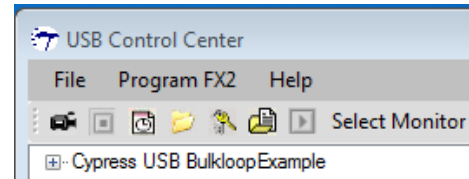


2. 至此，您已准备好下载 FX2LP 固件 **bulkloop.hex**(AN65209\FX2LP Bulkloop Firmware) 到 FX2LP DVK 内。点击赛普拉斯器件条目使其高亮显示，然后依次选择 **Program FX2 > RAM**。打开 **FX2LP Bulkloop Firmware** 文件夹，并选择 **bulkloop.hex** 文件。完成下载代码后，FX2LP 开发套件自己断开与 USB 的连接，然后以下载的固件 **bulkloop.hex** 创建的器件名称重新连接到 USB（图 15）。上面描述的是赛普拉斯的 ReNumeration™ — 表示两次对 FX2LP 进行枚举，第一次列举为代码加载器，第二次列举为加载的器件。

连接高速 USB 时，LED D5 将每秒闪烁 8 次，如果连接全速 USB 时，那么 LED 会每秒闪烁一次。通过查看 FX2LP DVK 上存在的 LED D5 切换频率，您可以观察到这一点。连接高速 USB 时，你会看到 LED 缓慢闪烁一秒左右，然后开始快速闪烁。这是因为 USB 器件最初全速连接，然后协商为以高速连接至主机。这是由赛普拉斯 USB 框架（显示在 **FX2LP 固件开发**部分中）处理的多个固件的详细信息中的一个。

FX2LP DVK 上的 7 段显示屏应发亮为数字 0，表示准备好传回主机的数据包数量 — 不收到任何数据包并没有回送任何数据包。

图 15.FX2LP 开发板再次出现



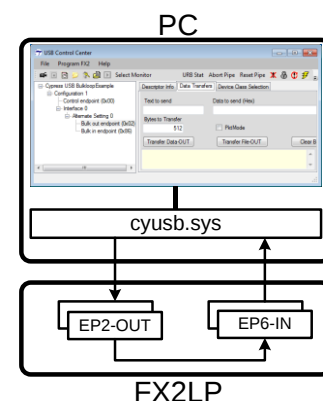
注意：将任何 hex 文件下载到 FX2LP DVK 前，首先要通过保持“EEPROM ENABLE”开关在“NO EEPROM”位置上来按下“Reset”按钮（图 11，左下角），以对 FX2LP 进行复位，从而使能 FX2LP Bootloader。

图 16 显示的是 USB 控制中心和加载 **bulkloop.hex** 文件的 FX2LP 开发板的交互过程。根据 PC 的连接，**bulkloop** 固件枚举为全速器件或高速器件，并将从 OUT 端点 2 缓冲区接收的 BULK 数据复制（返回）到 IN 端点 6 缓冲区内，以便传输回 PC。如框图显示，端点是双缓存的 — 是本节介绍的特性。

注意：因为 USB 面向主机为中心，所以“OUT”（输出）是指数据从主机传输到器件；“IN”（输入）是指数据从器件传输到主机。

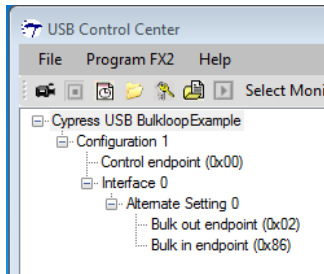
bulkloop 固件同样包含了代码，用于有用的操作，如：控制 FX2LP 开发板上的 LED 和 7 段显示屏，并使用端点中断。**FX2LP 固件开发**部分详细介绍与该代码所相关的内容。

图 16. USB 控制中心测试 Bulkloop 固件



3. 在 USB Control Center 窗口中，展开“Bulkloop Example”树，以便显示所实现的 BULK 端点（图 17）。

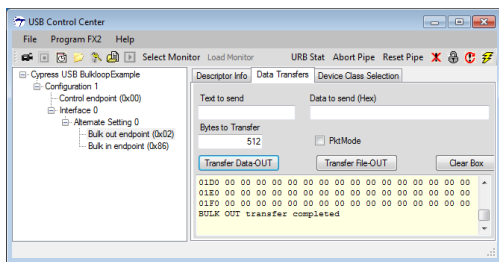
图 17.Bulkloop 示例器件内部



- 选择 **Data Transfer** “数据传输”选项卡。在屏幕的左侧，点击 **Bulk out endpoint (0x02)** 项，并记录“Transfer Data”按钮为 **Transfer Data-OUT**（传送数据输出）。点击该按钮会导致：

- 将 512 个字节（具有默认值为零）大小的数据从 PC 传输到 FX2LP DVK。
- LED D3 发生闪烁则表示进行的是 OUT 传输。
- 7 段显示屏的值增为 1，表示通过 EP2-OUT 接收了一个数据包，并将该数据包加载到 EP6-IN 端点 FIFO 内，准备传输到主机。

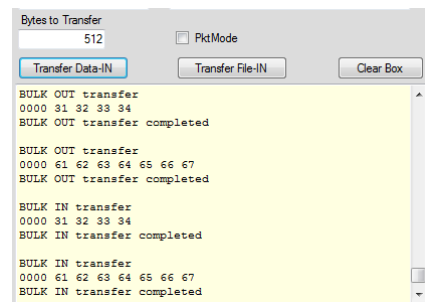
图 18. 成功传输 BULK-OUT



- 再次点击 **Transfer Data-OUT** 按钮。PC 将第二个数据包发送到 FX2LP 内，并且 7 段显示屏值会增为 2。
- 突出显示 **Bulk in endpoint (0x86)** 项。目前，将“传输”按钮标签显示为 **-IN**。点击该按钮会导致：
 - 将 FX2LP 中的 512 个字节数据传输到主机，USB 控制中心将该数据显示为十六进制值。
 - LED D2 闪烁用于表示 IN 传输。
 - 7 段显示屏值减少为 1。
- 重新点击 **Transfer Data-IN** 按钮。第二个行的数据包被传输给主机，并且 7 段显示屏值指示没有数据包正在等待进行传输。该序列会确认两个端点的双缓存操作。
- 再次选择 **Bulk out** 端点。然后，将鼠标光标放在 **Text to send** 框中，并键入 1234。每次键入数字时，在 **Data to send**（用于发送的数据）框和 **Bytes to Transfer**（用于传输的字节）框中显示的十六进制值会增加。点击 **Transfer Data-OUT** 按钮。

- 在 **Text to send** 框中，键入 **abcdefg**，然后点击 **Transfer Data-OUT** 按钮。
- 选中“Bulk in endpoint”，并保持 **Bytes to Transfer** 的默认值为 512 个字节。点击 **Transfer Data-IN** 按钮两次，以读取两个数据包已经传输的数据。

图 19. 两个数据包 OUT，两个数据包 IN

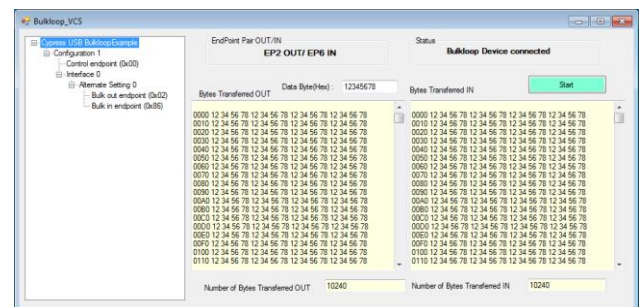


如图 19 所示，USB 的特性表现为：它是 USB 外设，该 USB 外设会始终发送比请求字节低的字节数量（在这里是 512 个字节）和有效的字节（4 或 7）。

使用 BULKLOOP_VCS

被称为 **Bulkloop_VCS** 的第二个 Visual Studio 项目从 FX2LP 开发板流线性地传入和传出。它连续发送和接收 BULK（批量）数据，同时保持运行传输字节的总量。与 USB 控制中心相同，在屏幕的左侧，Bulkloop_VCS 显示了器件的各种特性。与 USB 控制中心不同的是，当按下 **Start** 按钮时，它会自动进行设置，并连续进行传输。

图 20.Bulkloop_VCS 窗口



使用该应用前，请通过 USB 控制中心下载 **bulkloop.hex** 文件，如前一节所示。通过在“NO EEPROM”位置保持“EEPROM ENABLE”开关，确保先按下“FX2LP Development Kit RESET”按钮以使能 FX2LP Bootloader。

FX2LP 固件开发

本节介绍了生成 **bulkloop.hex** 文件有关的步骤。

Keil uVision2

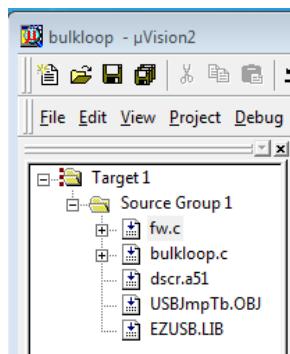
赛普拉斯 FX2LP DVK 开发板为 8051 提供了免费演示版的 Keil uVision2 工具。该 IDE 具有全面的功能，但限制代码大小最大为 4096 个字节。赛普拉斯的演示项目（如 **bulkloop.uv2**）满足该限制，所以您可以研究和修改代码。较大的项目需要购买 Keil uVision2 的完整工具箱。

赛普拉斯 USB 固件框架

与 Keil 工具结合使用，赛普拉斯提供了被称为 USB 固件框架的一组文件，以处理低级的 USB 细节。通过这种方法，您可以集中精力开发具体的应用代码。

当打开 **bulkloop.uv2** 项目时，可以查看下面各项目文件：

图 21.bulkloop.uv2 项目文件



- **fw.c** 是赛普拉斯的 USB 固件框架。

更多有关固件框架的详细信息，请参考 [FX2LP 开发套件用户指南.pdf](#) 中第五章的内容。

- **bulkloop.c** 是用户的应用。赛普拉斯的其他示例会将该代码模块命名为 **peripheral.c**；对用户的具体项目名称重新命名该模块是一个好主意。
- **dscr.a51** 是一个汇编语言文件，它包含了所需的数据来枚举 USB 器件。该文件包括以 **.db**（定义字节）语句格式的表数据。您编辑该文件，从而为您的设计自定义各个项目，如 VID/PID 和描述性字符串。
- **USBmpTb.OBJ** 是 FX2LP 架构所需要的跳转表，无需对其进行修改。
- **EZUSB.LIB** 包括 FX2LP 有用函数库，主要处理 I²C 通信。FX2LP DVK 包含了 EZUSB.LIB 库文件的源代码，并它们位于 **Target\Lib\LP** 子文件夹中。

图 22. USB 固件框架结构

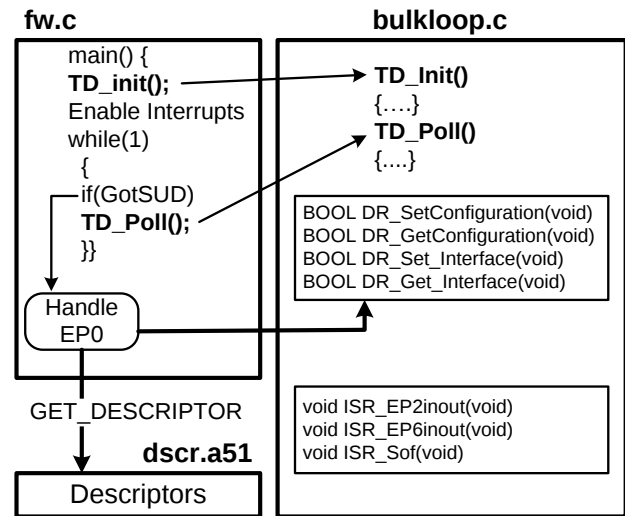


图 22 显示了各代码模块的组合方式。

Fw.c 文件包含 **main** 函数。它执行 USB 维持的大部分操作（如进行枚举），并且每当需要自定义时，它都会调用应用代码（**bulkloop.c**）中特定名称的外部函数。一般情况下，不需要修改 **Fw.c** 文件。执行各普通操作后，它将调用 **bulkloop.c** 中所提供的外部函数（称为 **TD_init**）。（前缀 **TD** 表示“任务调度”）。然后，它会进入一个无限循环，以通过 **CONTROL** 端点 0 检查 **SETUP** 数据包是否到来。该循环还会检查 USB 暂停事件，但 **bulkloop** 应用不会使用该循环。每次进入该循环，它都会调用在 **bulkloop.c** 文件中提供的外部函数 **TD_Poll**。在该应用中，**TD_Poll** 函数执行将循环端点数据从 PC 传输到 FX2LP 的工作。

每个 USB 外设都通过它们的 **CONTROL** 端点接收两个请求类型：枚举和操作。

枚举

当连接 USB 器件时，主机 PC 将发送多个 **GET_DESCRIPTOR** 请求来确定器件类型及其要求。这些操作属于枚举过程的一部分。**fw.c** 代码会截取这些请求，并通过使用 **dscr.a51** 文件中存储的数值处理请求。

使用 USB 框架的优势在于已经测试和验证过代码，并通过了 USB “第 9 章”中的要求。第 9 章是指 USB 规范中的一章，介绍器件请求（通过 EP0）及其正确响应。

操作

需要用户代码时，**fw.c** 都会调用一个带有 **DR** 前缀（器件请求）的特定名称的外部函数（存储在 **bulkloop.c** 文件中）。对于 **bulkloop** 这种简单的应用，仅使用一个配置和一个接口。因此，图 22 中所显示的两对 **DR_Set-Get** 函数只存储由主机发送的“Set”值，并在主机发出“Get”请求时对该值进行回送。对于更加复杂的配置，您可以通过使

用这些 DR 调用（“hooks”）更改摄像头的分辨率，或将请求路由到两个不同的接口。

因为，**bulkloop.c** 为所有 USB 器件包含完整模板，所以您可以将它们作为基础进行自定义。本节余下内容描述了该文件的三个部分，这三个部分要求用户代码实现 **bulkloop** 应用。

TD_Init

该函数执行以下操作：

- 将 8051 时钟频率设置为 48 MHz。
- 关闭开发板上的四个 LED。通过读取特定的存储器位置打开和关闭 LED。这种方法无需占用任何 I/O 引脚仍能控制 LED。
- 将 EP2 配置为 BULK-OUT 端点，并将 EP6 配置为 BULK-IN 端点。这两种设置都是双缓存的，并使用了 512 字节的 FIFO（图 16）。
- 使能（“激活”）EP2-OUT 以接受两个数据包。通过将任意值写入到设置 MSB 的字节计数寄存器内，可以激活（arm）OUT 端点。设置 MSB（称为“SKIP 位”）会跳过该数据包。
- 使能 FX2LP 双自动指针。这些硬件指针将自动递增，以便存储器到存储器字节从 EP2-OUT 缓冲区有效传输到 EP6-IN 缓冲区。
- 使能三种中断：SOF、EP2 和 EP6 端点中断。

TD_Poll

在 **fw.c** 文件的无限循环中调用 TD_Poll（图 22）。**bulkloop** 应用仅需要两个任务：

1. 使用 7 段显示屏的值来更新等待传输到主机的数据包数量。FX2LP 寄存器 EP6CS（端点 6 控制和状态）提供了位 6-4 中的数据包数量。

更多参考资料

更多有关寄存器的详细信息，请参考 [FX2LP 技术参考手册](#) 第 15 章中的内容。

2. 检查端点 FIFO 标志，以便确定将 OUT 数据包传输到 IN 缓冲区的时间。如果到时间，则使用自动指针将数据包从 EP2-OUT 缓冲区传输到 EP6-IN 缓冲区内。

为了解第二项如何执行传输，需要了解有关 FX2LP 端点 FIFO 标志的下面两个点：

- a) 当使用多个缓冲区时，（EP2468STAT 寄存器的）FULL 标志和 EMPTY 标志反映的是所有缓冲区，而不是一个。因此，对于使用双缓冲区技术的情况，如果接收到一个 OUT 数据包，则 FULL 标志仍处于未激活状态，这是因为第二个数据包仍可用于 OUT 传输。接收第二个数据包后，才会激活 FULL 标志。同样，两个缓

冲区都为空时，才能激活 IN 端点 EMPTY 标志，以就绪使 8051 将新数据写入到两个缓冲区内。

- b) 成功接收或传输 USB 数据包后，FX2LP 才会更新（EP2468STAT 寄存器的）FIFO 标志。

因此，满足下面的两个条件后将发生循环传输：

- EP2-OUT 非为空，以及
- EP6-IN 未滿。

换句话说，EP2 具有一个数据包，EP6 具有存储一个数据包的空间。通过这种方法执行测试会处理任何数据包大小，并可考虑到使用双缓冲区。

中断服务子程序

bulkloop.c 文件为每个 USB 中断源提供了 ISR 函数。几个 ISR 为 **fw.c** 文件中的代码设置标志，并且其他 ISR 会执行 USB 枚举功能。您只需要为您应用所使用的 ISR 填写代码。

下面的 ISR 会向您介绍如何清除一个 FX2LP USB 中断请求：

```
// Setup Token Interrupt Handler
void ISR_Sutok(void) interrupt 0
{
    EZUSB_IRQ_CLEAR();
    USBIRQ = bmSUTOK; // Clear SUTOK IRQ
}
```

interrupt（中断）关键字后面的数字“0”表示所有 USB 中断请求的 ID。清除两个中断请求标志的顺序是：首先清除通用 USB 中断标志，然后清除单独 USB 源标志（在该示例中则为“Setup Token Arrived”标志）。该 ISR 是代码“hook”的示例；当接收 SETUP 数据包时，通过将代码插入该 ISR，您可以执行任何操作。

bulkloop 应用需要四个自定义的 ISR：

Set_Configuration ISR

主机设置 FX2LP 配置是枚举过程中的最后步骤。这时很适合初始化应用的硬件。在此期间，对驱动 7 段显示屏的 I²C 单元进行初始化。

EP2INOUT/EP6INOUT ISR

EP6-IN 发送数据包或 EP2-OUT 接受数据包时，则这些 IRQ 会运行。ISR 代码打开 FX2LP 开发板上的一个 LED，然后设置 **inblink**（EP6-IN）或 **outblink**（EP2-OUT）变量（在 **bulkloop.c** 中），以便控制 LED 的发亮时长。

SOF ISR

作为一个方便的定时器，SOF ISR 在全速下每一毫秒会运行一次，并在高速下每 125 微秒会运行一次。ISR 运行 500 次后，ISR 代码会对 FX2LP 开发板上的 LED 切换一次，相当于全速每秒切换一次，并且在高速中每秒切换 8 次。ISR

代码也会递减 **inblink** 和 **outblink** 变量，当接收 IN 和 OUT 数据包时设置这些变量，当计数器到达零值时这些变量会关闭指示灯 LED。

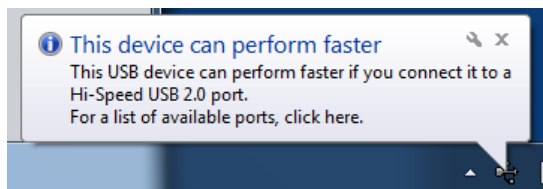
处理 USB 双速

USB 2.0 器件复位后，它会全速工作。然后，主机通过使用低级总线信号与器件一起协商，以确定总线的运行速度。FX2LP 提供了一个中断，表示它刚切换到高速操作状态，这是主机和器件速度协商的结果。

USB 2.0 器件必须在全速和高速下运行。高速 USB 器件提供了两套描述符，一套描述符用于一个速度。两个 ISR 根据速度指定适当的描述符：

- **ISR_Ures** (USB Reset) 代码指定全速描述符为 *当前速度* 描述符，并指定高速描述符为 *其他速度* 描述符。如果将高速 USB 器件插入全速端口内，无需任何操作。
- **ISR_Highspeed** 中断服务代码互换 *当前/其他* 描述符——现在，高速换为 *当前速度*，并全速换为 *其他速度*。通过这种双重定义，如果 Windows 检测到插入全速端口内的高速器件，它会显示如下信息：

图 23. Windows 识别高速器件可更好执行的时间

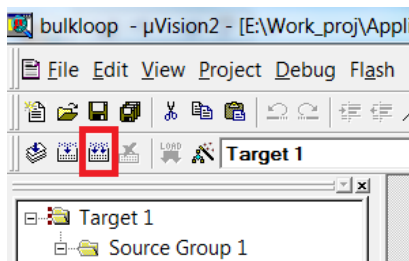


得到注意的一点是 USB 管理列出固件框架为您所提供的功能。双描述符和描述符交换代码专为您而编写。您只需要为应用填写描述符字段。

编译 Bulkloop 项目

在“FX2LP Bulkloop Firmware”文件夹中，双击 **bulkloop.uv2** 文件。这样会打开 uVision2 IDE 和下载 bulkloop 项目。编译和链接项目，单击 **Rebuild All Target Files**（重建所有目标文件）按钮。

图 24. uVision2 IDE

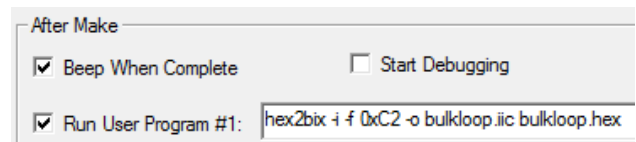


这样会创建前一节中下载的 **bulkloop.hex** 文件。

安装 Keil uVision2 时，请注意下面内容：

- 有时，从网站下载安装的解压缩文件为只读文件。这时请右键单击文件夹，选择“Properties”，并取消选中“Read-only”（只读）复选框（如果已选中它）。
- 最终 Keil 编译步骤是自动运行被称为 **hex2bix.exe** 的程序，该程序将 Keil 输出转换为“iic”文件。该文件被编码到项目中，因此移动或安装 Keil 工具会打破路径。通过将 **hex2bix.exe** 文件复制到 Keil 项目文件夹中并运行工具（而不要指定路径）可解决该问题，（可在本应用笔记的同伴代码中执行上述操作）。通过右键单击“Keil project”窗口中的顶层文件（默认名称为“Target 1”），选择“Options...”，然后选择“Output”选项卡，您可以查看路径。路径会如下显示：

图 25. “hex2bix”工具的路径

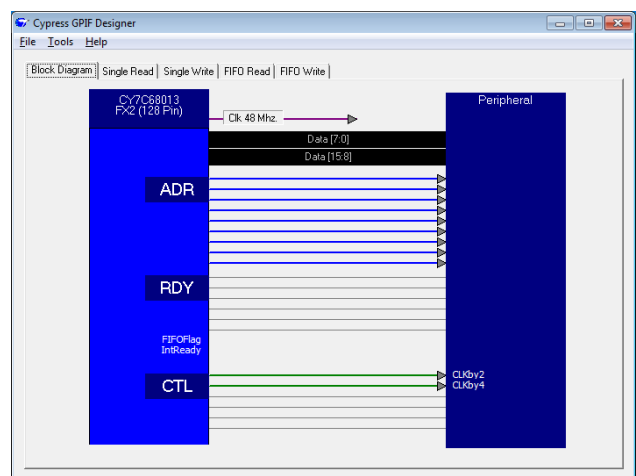


GPiF Designer (GPiF 设计程序)

bulkloop 示例将处理片上所有操作，其中 8051 将检查标志，传输各端点 FIFO 之间的字节，并更新开发套件状态指示灯（LED 和 7 段显示屏）。大部分的 FX2LP 设计都执行了更多的操作，传输 FX2LP 和外面外设（如磁盘驱动器或图像传感器）之间的数据。为实现该目的，GPiF 为外部系统提供快速和灵活的接口。本节重点介绍了 **AN66806 — EZ-USB FX2LP GPiF 入门**，提供使用赛普拉斯的 GPiF Designer 工具的指南。

GPiF 设计的第一步是在“block diagram”（框图）选项卡中填写接口信号，如图 26 所示。

图 26. GPiF Designer 的“Block Diagram”选项卡



GPiF 有六种控制（CTL）输出和六种就绪（RDY）输入，通过右键单击“RDY”或“CTL”文本框可重命名 Block

diagram 选项卡中的这些输入和输出信号。您设定的信号名称用于整个 GPIF Designer，并作为状态机的决定点和波形的名称选项。

图 27. GPIF 波形条目创建状态

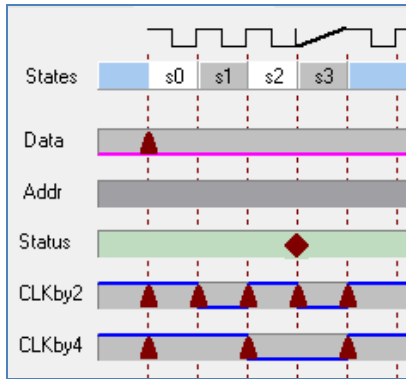


图 27 显示了 GPIF Designer 波形条目的屏幕。在这个简单的示例中，对内部 48 MHz 时钟进行 2 和 4 分频，并在重新命名的 CTL0 (CLKby2) 和 CTL1 (CLKby4) 引脚上输出这两个时钟。垂直虚线表示时钟边沿上可以发生切换操作；可以将时钟边沿设置为上升沿或下降沿。要创建一个切换，请在一个时钟边沿上点击所需要的波形。这样会触发波形，并插入红色三角形，以标记切换点。您可以删除这些三角形或将它们拖到其他时钟边沿上，并且只要点击鼠标便能够直接将逻辑电平设置为 1 或 0。

当您进行定义波形转换时，GPIF Designer 会在最顶行自动创建各个状态；在该示例中，具有四个状态：从状态 S0 到状态 S3。因为该波形专门为生成时钟输出而设计，所以设计它来用于无限重复的操作。通过从状态 3 返回到状态 0 实现无限重复的操作。在 Status 线上插入决定点 (decision point) 可实现分支的状态 — 点击在状态 3 起始点的虚线可以插入决定点。GPIF Designer 插入一个菱形来表示决定状态。右键单击该菱形会显示出一个对话框，通过该对话框您可以使用两个变量和三个逻辑运算符 (AND、OR、XOR) 创造逻辑公式，用于分支条件。通过为 if 条件和 else 条件选择相同的状态目的地可以创建无条件分支。

更多参考资料

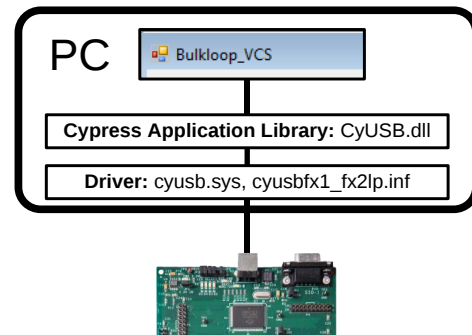
更多有关 GPIF 的详细信息，请参考 [FX2LP 技术参考手册](#) 中第 10 章的内容。

更多有关连接 RAM 到 FX2LP 的 GPIF 的信息，请参考[通过 GPIF 将 SRAM 与 FX2LP 相连接 — AN57322](#)。

Windows 端

运行 [Bulkloop](#) 演示向您介绍了如何使用 **USB 控制中心** 和 **BULKLOOP_VCS** Windows 应用来测试 Keil 工具生成的 bulkloop hex 文件。本节说明了赛普拉斯的工具和其他供应商工具的概述，这些工具可用于创建同基于 FX2LP 的器件进行通信的 Windows 应用。

图 28. 使用赛普拉斯驱动程序和库，Windows .NET 应用与 FX2LP 相连接



赛普拉斯 USB 驱动程序

赛普拉斯 USB 驱动程序 (**CyUSB.sys**) 是一个功能强大的 Windows 驱动程序，可用于主机和 USB 器件间的连接。在主机 PC 上安装该驱动程序，将其作为 EZ-USB FX2LP DVK 安装的一部分，并随附 **Windows Driver** 文件夹中的 zip 文件。该驱动程序以二进制的形式与基于 FX2LP 的器件一起分销。

- **CyUSB.sys** 是一个内核模式 USB 功能的驱动程序，能够与所有符合 USB 2.0 标准器件进行通信。该驱动程序是通用的，与基本 USB 指令相兼容，但不能执行 USB 器件类更高级的指令。
- 驱动程序非常适合在自定义 USB 应用中与供应商指定的器件相连接，或者将低级 USB 请求发送到任何 USB 器件，用于实验或诊断应用。
- 驱动程序支持的操作系统/平台是 Windows XP、Windows Vista 和 Windows 7，均包括 32 位和 64 位版本。

如果您修改了 .inf 文件中 (例如) 包括贵公司的 VID 和/或 PID，或者包括自定义的描述性文本字符串，赛普拉斯建议您获取 Microsoft 的“WHQL”认证 (Windows 硬件质量实验室的 WHQL 标准) 和 Windows 徽标认证，这样在您安装驱动程序时，便不会再收到警告消息。[附录 E: 向.inf 文件添加自定义 VID 和 PID](#) 介绍添加自定义 VID/PID 到 .inf 文件中的方法。

更多参考资料

更多有关 **CyUSB.sys** 的信息，请参考 **Cypress Suite USB\Driver** 文件夹中保存的 **CyUSB.pdf** (安装赛普拉斯 Suite USB 后才出现该文件夹)。

驱动程序/库的替代方案

其他 Windows 驱动程序和库均可用。Microsoft 的 WinUSB (请参考 [msdn](#) 网站，了解详细信息) 包括一个内核模式驱动程序 (**Winusb.sys**) 和一个显示 WinUSB 函数的用户模式动态链接库 (**Winusb.dll**)。通过使用这些高级函数，您可以使用用户模式软件管理 USB 器件。Windows XP、

Windows Vista、Windows 7 和 Windows 8 都提供了 **Winusb.sys**。与赛普拉斯的驱动程序相同，它要求通过一个 *.inf* 文件附带，进行安装。

注意：

- **WinUSB.sys** 不支持 ISOCHRONOUS 传输类型，但是赛普拉斯的驱动程序支持所有 USB 传输类型。
- **Linux 和 Mac** 用户可以调查 LIBUSB，它是一套开放源代码的用户模式的 USB 子程序。可以查看的内容，了解更加详细的信息。

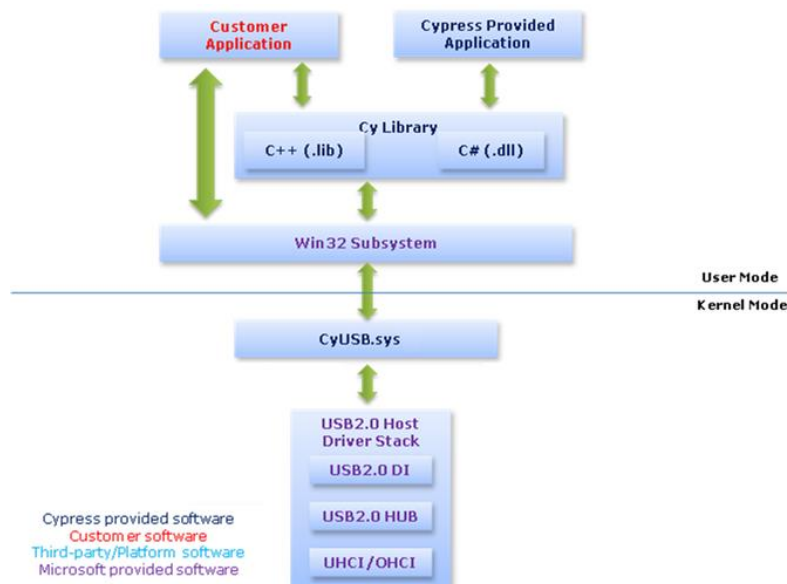
在 Linux 机器中，通过使用用于 Linux 软件的 CyUSB Suite（作为 Linux FX3 SDK 的一部分，并可用于 FX2），您可以将固件镜像加载到 FX2/FX3 器件，并在器件上测试各种接口。请参考位于 [Linux 的 FX3 SDK 安装文件夹](#)

（*fx3_sdk_v1.3_linux\cyusb_linux_1.0.4\cyusb_linux_1.0.4\docs*）中的 *cyusb_linux_user_guide.pdf*。该文档介绍的是安装软件、下载固件到 FX2/FX3 内、测试供应商扩展、BULK OUT/IN 传输和 ISOCHRONOUS OUT/IN 传输的方法。Linux 操作系统中的 CyUSB Suite — Programmer 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，同时介绍如何编译并将用户写入应用集成到该库内。

同样可以编程 FX2LP，使之与 Windows 标准类（例如：人机接口设备（HID）、通信（COMM）设备或大容量存储（MSC）设备）相兼容。赛普拉斯网站拥有 FX2LP 示例程序，用于这些通用类型。符合 Windows 类型的优势是当器件初始插入 Windows 时不需要安装驱动程序 — 由于驱动程序是 Windows 的一部分。

赛普拉斯库

图 29. 赛普拉斯库与应用进行通信的详细概况



Bulkloop_VCS 应用是软件结构的示例，该软件结构用于使用赛普拉斯.NET 库的任何 Windows 与 FX2LP 进行的通信。使用 C# 语言的程序创建库调用，用于访问赛普拉斯驱动程序，从而与 FX2LP 进行通信（请参考图 28）。**CyUSB.dll** 库支持 Microsoft .NET 语言。赛普拉斯还提供了被称为 **CyAPI.lib** 的 C++ 类库。这两种库会随附 **Cypress Suite USB** 安装。

正如赛普拉斯 USB 固件框架可以在 FX2LP 中简化了代码开发，这些库简化了 Windows 中的代码。软件模型（基于 **CyUSB.dll**）采用了下面三个元素：

- **USBDeviceList** 类创建所有连接的 USB 器件的列表。构造函数可以过滤掉单一的器件类（例如：HID 或

MSC）列表，或由赛普拉斯驱动程序支持的器件列表。

- 器件列表成员是 **CyUSBDevice** 类的示例。
- **CyUSBDevice** 类会揭示一个或多个 **CyUSBEndpoints**，从而执行数据传输。

与多种 Win32 API 调用（如：**DeviceIOControl**）相比，使用这种高级模式与 USB 器件进行通信是一个很大的改进。

使用 Microsoft Visual C# Express 版本来打开 **Bulkloop_VCS** 解决方案并检查代码是熟悉赛普拉斯库的一个好方法。

更多参考资料

更多有关开发主机应用的详细信息，请参考应用笔记，[AN70983 — 为 EZ-USB® FX2LP™/FX3™设计批量传输主机应用](#)的内容。

更多有关.NET dll 库的详细信息，请参考 **Cypress Suite USB\CyUSB.NET** 文件夹中存在的 *CyUSB.NET.pdf*（成功安装赛普拉斯 Suite USB 后会出现该文件夹）。有关 **CyAPI.lib** 的详细信息，请参考 **Cypress Suite USB\CyAPI** 文件夹中的 *CyAPI.pdf*。

汇总

设计赛普拉斯 FX2LP 来满足 USB 2.0 高速设计要求。赛普拉斯提供了大量的支持资料，以帮助设计周期中的每一步。本应用笔记先介绍了 FX2LP，然后介绍赛普拉斯硬件、固件和软件工具，BULK 传输示例的设计、创建和测试。该示例可以作为您自定义应用的基础。

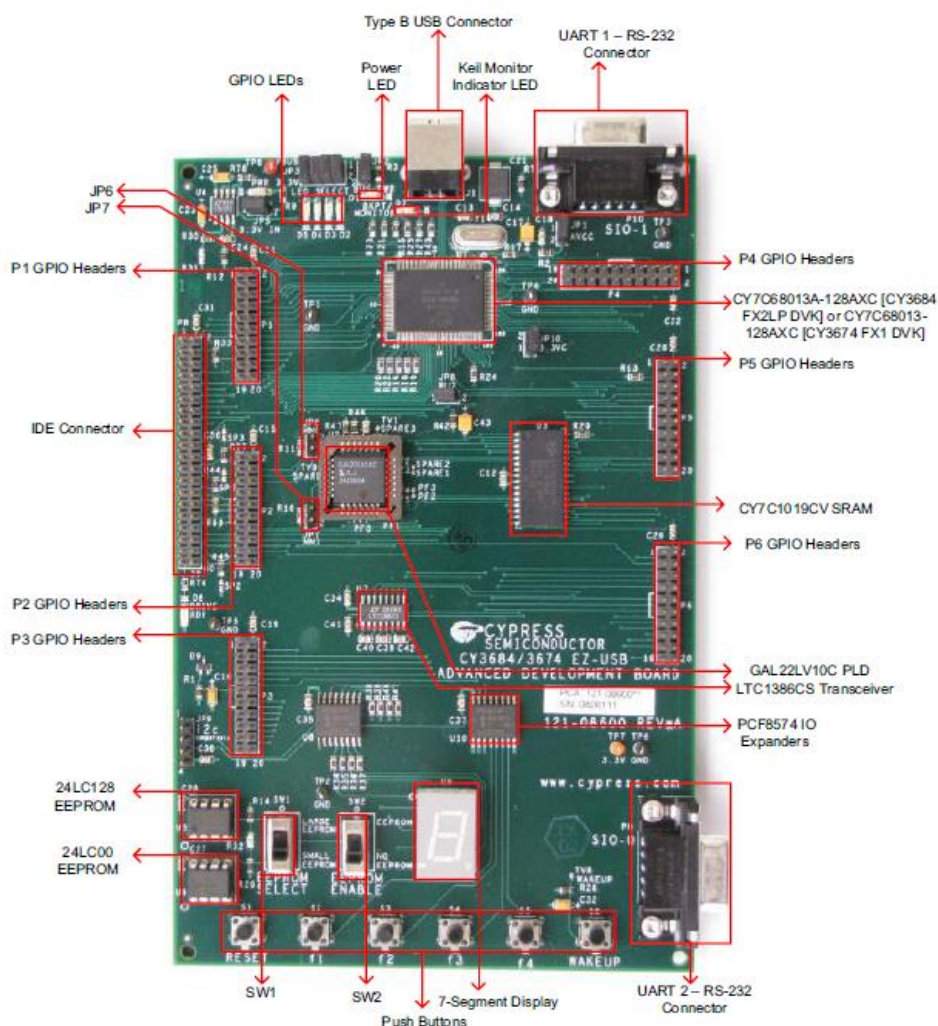
关于作者

姓名：	Rama Sai Krishna
职务：	高级应用工程师
联系邮箱：	rskv@cypress.com

附录 A: FX2LP 开发套件 (DVK)

CY3684EZ-USB FX2LP 开发套件是完整的开发资源。它提供了一个用以开发及测试自定义项目的平台。开发套件包含用于固件、硬件和软件等方面的设计的辅助材料。图 30 显示的是 FX2LP DVK 开发板组件。

图 30.FX2LP DVK 开发板组件



FX2LP DVK 是 FX2LP 的一个理想的调试平台。该开发板可以从 USB 加载代码镜像（如本应用笔记中介绍）或将代码编程到板上 EEPROM 中。

当为 USB 2.0 高速器件开发硬件时，开发板布局和设计会决定项目的成败。为了帮助开发人员避免常见的布局错误，赛普拉斯提供了某些资源，这些资源对设计新板非常有用。[AN1168](#) 和 [AN15456](#) 应用笔记给出了 FX2LP 的设计指南。此外，FX2LP DVK 安装的 *hardware*（硬件）子目录中还包含了 FX2LP 开发套件原理图、BOM（用于开发板）以及开发板布局和设计文件。

固件示例项目

如 [FX2LP 固件开发](#) 部分介绍，FX2LP DVK 提供了一个 USB 固件框架库，该框架库满足 USB 2.0 规范中第 9 章对高速器件的要求。通过使用常见的操作（如：FX2LP 芯片的初始化、USB 标准器件请求处理、USB 挂起 - 恢复电源管理）的赛普拉斯代码，USB 框架简化并加速了自定义固件的开发。USB 固件框架还提供了 hook 函数和固件示例，以简化固件的开发过程。您可以对 USB 描述符表和代码进行写操作，以实现所需要的功能，而不用担心低级别的 USB 细节。

安装 [CY3684EZ-USB FX2LP 开发套件](#) 后，*固件* 目录包含了表 4 中所列出的示例。可以将这些示例作为参考使用，也可以将它们作为基于 FX2LP 自定义产品的基础使用。

表 4.FX2LP 固件示例的说明

序号	固件示例	说明
1	hid_kb	通过使用 DVK 开发板上各按键和 7 段显示屏来仿真 HID 类键盘。
2	Bulkloop	它是一个批量回送测试操作，该操作执行 EZ-USB 批量端点。它将 EP2OUT 和 EP4OUT 分别回送到 EP6IN 和 EP8IN 内，并更新开发板上指示灯。
3	Bulkext	是一个使用外部 RAM 的批量回送测试。将数据从 OUT 端点缓冲区复制到 DVK 开发板上的外部 RAM 内，然后再返回到 IN 端点缓冲区。它将 EP2OUT 和 EP4OUT 分别回送到 EP6IN 和 EP8IN 内。
4	Bulksrc	它包含了 BULK 数据的无限提供者和消费者，仅用于测试。使用 CyConsole 或 CyBulk 来驱动它。EP2OUT 和 EP4OUT 会接收所有 BULK 的数据包。在高速运行时，EP6IN 会始终返回一个大小为 512 个字节的数据包，但在全速运行时返回的数据包大小为 64 个字节。根据 EP8IN 中缓冲区的可用性，将 EP4OUT 的最新数据包写入到 EP8IN 内。
5	dev_io	它包含了各源文件，用于创建简单的开发板 I/O 的示例。该软件说明了如何使用 EZ-USB 开发套件中的按键和 LED。
6	EP_Interrupts	使用端点中断批量回送固件。
7	extr_intr	使用 INT0、INT1、INT4、INT5 和 INT6 处理外部中断。
8	ibn	使用 IBN (In Bulk NAK) 中断将 EP2OUT 和 EP4OUT 分别批量回送到 EP6IN 和 EP8IN 内，以启动传输操作。
9	LEDCycle	它用于表示 DVK 上通用指示器 LED (D2、D3、D4、D5) 的使用方法。
10	Pingnak	使用 PING NAK 中断将 EP2OUT 和 EP4OUT 分别批量回送到 EP6IN 和 EP8IN 内，以开始进行传输操作。
11	iMemtest	测试片上 RAM。
12	vend_ax	显示如何实现供应商特定的指令。

USB 固件框架使用 EZ-USB 库 (EZUSB.LIB)，该文件实现了多个固件项目的共同函数。不需要修改这些函数，因此要以库的形式来提供。然而，您需要修改某个函数，或者您只是想知道某个操作的执行方式时，该套件会包括库的源代码。更多有关 EZ-USB 库 (5.4 节) 和固件框图 (第 5 章) 的信息，请查看 [CY3684 DVK 套件指南](#)。在 DVK 中，该库位于 ‘TargetLib\LP’ 文件夹中。为了帮助客户使用广泛的赛普拉斯高速产品来快速设计他们的应用，赛普拉斯公司提供了所有 [USB 高速代码示例](#) 的完整列表。有关其他 USB 高速器件、套件、SDK、应用笔记和参考设计的信息，请参考下面内容。

注意： 这些软件资源仍在不断升级。因此，建议从 [赛普拉斯网站](#) 下载最新软件。

附录 B：赛普拉斯的额外 USB 高速器件

赛普拉斯 USB 高速外设系列包括下列的其他器件：

- **AT2LP**：赛普拉斯的 EZ-USB® AT2LP™（CY7C68300C/301C/320C）作为从一个 USB 端口到一个或两个基于 ATA/ATAPI 的大容量存储器件端口之间的固定功能桥接器使用。通过 AT2LP 上的 PATA 接口可以使用设计中的硬盘驱动器（HDD）、CF 卡（compact flash）和固态硬盘（SSD）。AT2LP 非常适用于大容量存储类应用，并且能够加快上市时间，而不需要使用自定义固件。AT2LP 支持与 ATA/ATAPI-6 相兼容的所有大容量存储器件。
- **NX2LP-Flex**：赛普拉斯 EZ-USB NX2LP-Flex™（CY7C68033/34）是 SLC NAND 控制器可编程的低功耗 USB。通过其可编程性，可使设计者根据 NAND 器件支持进行编程来包含控制器内的各种特殊性能；这是与固定控制器相比所具有的优势。安装在 NX2LP-Flex 中的硬件 ECC 引擎支持 1 位错误纠正和 2 位错误检测。
- **SX2**：EZ-USB SX2（CY7C68001）是可编程器件，能够与所有外部主设备（如标准微处理器、DSP、ASIC 和 FPGA）一起工作，以便支持所有外设设计中的 USB 2.0。SX2 具有一个内置的 USB 收发器、串行接口引擎（SIE）以及一个指令译码器，用于发送和接收 USB 数据。该控制器有四个端点，共享大小为 4 KB 的 FIFO 空间，从而获得最大的灵活性和吞吐量。SX2 有三个地址引脚和一个可选的 8 位或 16 位数据总线，用于指令和数据输入或输出。
- **FX2LP18**：MoBL-USB FX2LP18（CY7C68053）的工作电压为 1.8 V，适用于低功耗的手持设备。[AN6076](#) 列出了 FX2LP 和 FX2LP18 之间的区别。

附录 C：第三方开发套件和软件开发套件和 SDK

1. Opal Kelly 的 FPGA + FX2LP 开发板：

请访问下面网站查看有关该板的详细信息：

<http://www.opalkelly.com/products/xem6010/>

特性：

- 用于下载和控制的高速度 USB 2.0 接口（赛普拉斯 FX2LP — CY68013A）
- Xilinx Spartan-6（XC6SLX45-2FGG 或 XC6SLX150-2FGG）
- 32 Mib 的串行闪存（Numonyx M32P25）
- 128 MiByte DDR2（Micron MT47H64M16HR）
- 小外形：比尺寸为 75 毫米 x 50 毫米 x 15.9 毫米（2.95" x 1.97" x 0.63"）的信用卡更小
- 通过外部直流电源自供电
- 多 PLL、多输出的时钟发生器（赛普拉斯 CY22393）

2. ZTEX 的 FPGA + FX2LP 开发板：

请访问下面网站查看有关该开发板的详细信息：

<http://www.ztex.de/usb-fpga-1/>

某个相同 ZTEX 电路板的特性：

- 拥有赛普拉斯 CY7C68013A EZ-USB FX2LP 微控制器
- 高速度（480 Mbps）USB 接口
- Xilinx Spartan 3 XC3S400 FPGA
- 60 个通用 I/O（GPIO）
- 20 个特殊 I/O（SIO）
- 128 kb EEPROM（用于存储固件）
- 闪存存储器（可选）

第三方软件开发套件 SDK

ZTEX 提供了与基于 FX2LP 开发板一起使用的 SDK；另外它还提供基于 JAVA 的 API，这有助于开发主机软件。更多详细信息，请访问 <http://www.ztex.de/firmware-kit/> 网站。

附录 D：应用笔记和参考设计

应用笔记

- [AN15456 — EZ-USB® FX2LP™和 EZ-USB FX1™成功硬件设计和调试指南](#)

本应用笔记指出可能发生（特别是在高速工作的情况下）的 USB 硬件设计问题。它也有助于构建电路板前捕获潜在问题，同时协助启动并运行电路板时的调试操作。

- [AN50963 — EZ-USB® FX1™/FX2LP™启动选项](#)

本应用笔记介绍了将固件下载到 FX1/FX2LP 内的各种方式。

- [AN66806 — EZ-USB® FX2LP™ GPIF 设计指南](#)

本应用笔记说明了使用 GPIF Designer 来开发 GPIF 波形信号所需的各个步骤。

- [AN61345 — 执行 FX2LP™-FPGA 接口](#)

本应用笔记提供示例项目，以使 FPGA 与 FX2LP 相连接。该接口为基于 FPGA 的应用（如数据采集、工业控制和监控以及图像处理）执行高速度的 USB 连接。FX2LP 在从设备 FIFO 模式下运行，FPGA 作为主设备。本应用笔记还提供了示例 FX2LP 固件来实现从设备 FIFO，并提供了示例 VHDL 和 Verilog 项目来实现 FPGA。

- [AN57322 — 通过 GPIF 使 SRAM 与 FX2LP 相连接](#)

本应用笔记说明了如何使用通用可编程接口（GPIF）将赛普拉斯的 CY7C1399B SRAM 连接到 FX2LP。本笔记描述了使用 GPIF Designer 来制定读和写信号的方式。它还提供了连接 FX2LP 到其它 SRAM 的参考。

- [AN58009 — FX1/FX2LP 固件的串行（UART）端口调试](#)

本应用笔记描述了 FX2LP 固件中所需进行串行调试的代码。此代码会使开发者记录 PC 终端程序中的调试信息和变量的实时数值，或通过使用 FX2LP 中的 UART 引擎将数据捕获到一个文件内。

- [AN42499 — Keil 调试器环境的设置、使用和故障排除](#)

本应用笔记为刚入门用户提供了使用 Keil 调试器的逐步指导。该指南的内容包括：从 PC 到 SIO-1/0 的串行线缆连接、监控器代码下载以及所需的项目设置。另外，它还提供如何启动和停止某个调试会话、设置断点、单步运行代码以及解决潜在问题。

- [AN4053 — EZ-USB® FX2 和 EZ-USB FX2LP 上经过 Isochronous/Bulk 端点的数据流](#)

本应用笔记提供了有关使用 EZ-USB FX2 或 EZ-USB FX2LP 器件的流应用的背景信息。本笔记介绍了有关经过 BULK 端点、ISOCHRONOUS 端点和高带宽

ISOCHRONOUS 端点的数据流的信息，同时还说明了高带宽应用中使用 FX2/FX2LP 时需要注意的有关设计的问题。

- [AN58069 — 在 FX2LP 中使用从设备 FIFO 模式执行 8 位的并行 MPEG2-TS 接口](#)

本应用笔记说明了如何使用从设备 FIFO 模式执行 8 位的并行 MPEG2-TS 接口。示例代码将 EZ-USB FX2LP 作为接收器端使用，并将数据发生器作为数据流的源端使用。本应用笔记提供了有关硬件连接和示例代码的内容。

- [AN58170 — 使用 EZ-USB 的代码/存储器划分](#)

EZ-USB FX2 系列芯片包含一个 8051 内核。8051 内核具有 16 位地址线，并且可访问 64 KB 大小的存储器。但是，有些应用需要访问大于 64 KB 的空间。本应用笔记说明了解决 64 KB 界限的方式。

- [AN1193 — 在基于赛普拉斯 EZ-USB FX2LP 的应用中使用定时器中断](#)

本应用笔记协助 EZ-USB® FX2LP 固件开发者在他们的应用中使用定时器中断。

- [AN63787 — 使用 8 位异步接口配置 EZ-USB® FX2LP™ GPIF 和从设备 FIFO 的示例](#)

本应用笔记描述了在手动模式和自动模式下如何配置 EZ-USB FX2LP 中的通用可编程接口（GPIF）和从设备 FIFO，以便执行 8 位异步并行接口。使用两个背对背连接的 FX2LP 开发套件来测试本应用笔记：第一个套件以主设备模式工作，第二个套件以从设备模式工作。

- [AN61244 — EZ-USB 中的固件优化](#)

本应用笔记描述了 EZ-USB 中的固件优化方式。有些方式适用于所有处理器，则有些方式只适用于 EZ-USB FX2LP 的 8051 内核。

- [AN74505 — EZ USB FX2LP：使用 LIBUSB 开发 MAC OS X 上的 USB 应用](#)

本应用笔记介绍了建立在 MAC OS 平台上使用 libusb 的主机应用。主机应用（Cocoa 应用）使用由 libusb API 提供的接口同 FX2LP 的 BULK IN 和 BULK OUT 端点进行通信。该主机应用针对传输特定 VID/PID（=0x04B4/0x1004）识别的器件执行传输。

- [AN58764 — 在 FX2LP 中执行虚拟的 COM 端口](#)

本应用笔记说明了如何使用标准的 Windows 驱动器在 FX2LP 中执行虚拟的 COM 端口。该信息有助于从 UART 移植到 USB。

- [AN45471 — 面向 FX2LP 的供应商指令的设计指南](#)

本应用笔记说明了如何编写 USB 供应商指令来执行特定产品。另外，本手册还说明如何使用赛普拉斯的 CyConsole 工具来发出供应商指令。

参考设计

某些 FX2LP 的参考设计适用于通用的应用。参考设计包括了适当时提供给设计的示范源代码、参考原理图和 BOM。

赛普拉斯网页上的可用参考设计为：

- **CY4661 — 具有指纹认证安全功能的外部 USB 硬盘驱动（HDD）**

赛普拉斯和 UPEK 的 CY4661 参考设计套件为用户提供具有指纹认证安全的外部 USB 硬盘驱动（HDD）的全套解决方案，从而能够保护与验证数据。参考设计使用了 UPEK 的 Touch Strip 指纹认证解决方案（TCS3 刷卡指纹传感器和 TCD42 安全 ASIC）。

- **FX2LP DMB-T/H TV 软件狗参考设计**

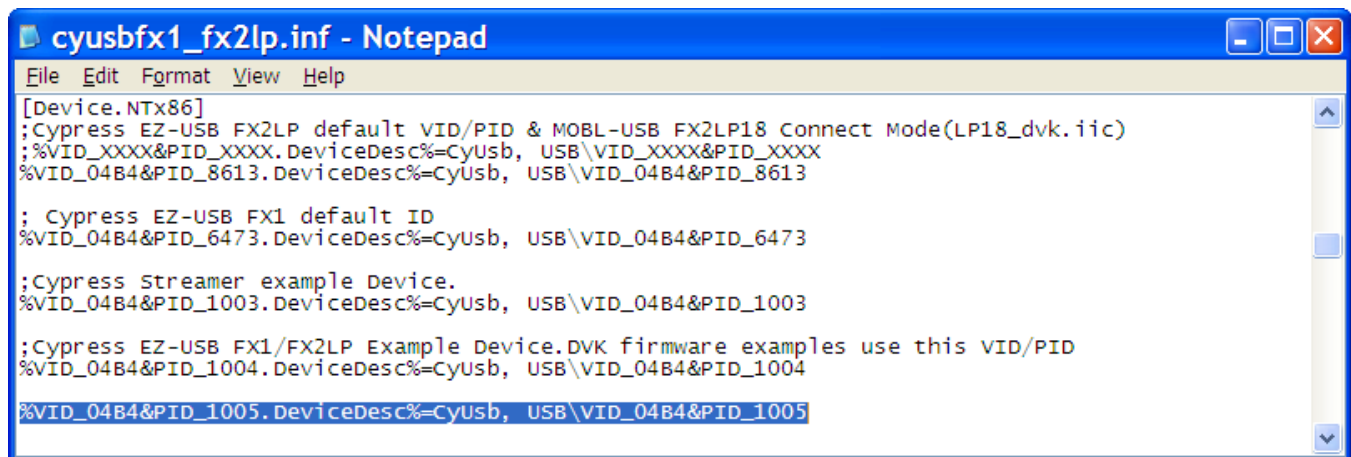
该参考设计套件是基于赛普拉斯 FX2LP 和 Legend Silicon 芯片集的。被捕捉和解调的 RF 信号由 Legend Silicon 芯片集转换为 MPEG2 TS 流，然后通过 FX2LP 将该信号发送到 PC。PC 使用媒体播放器来播放这些数据流。这是一个完整的设计，包括了所有需要的文件。

附录 E：向.inf 文件添加自定义 VID 和 PID

Bulkloop 固件示例使用 VID 0x4B4 和 PID 0x1004。这些值包含在本应用笔记附带的 **cyusbf1_fx2lp.inf** 文件内。如果您修改了 FX2LP 固件项目中的 VID 和 PID 值，并且没有将这些值列在 .inf 文件中，那么这时您需要添加这些值，这样赛普拉斯的开发工具（如 USB 控制中心）才可识别您所使用的设备。下面各步说明了如何将自定义 VID 和 PID 值添加到 **cyusbf1_fx2lp.inf** 文件内。

打开 **cyusbf1_fx2lp.inf** 文件。因为这是文本文件，所以您可以使用任何文本编辑器，如 WordPad（写字板）。添加您的自定义 VID 和 PID，如图 31 和图 32 所示。这些图中所示的步骤假设自定义 VID 为 0x4B4 和 PID 为 0x1005。请注意，这些步骤是针对 Windows XP 的 32 位平台进行的。

图 31. 向 Cyusb.inf 添加了自定义 VID 和 PID



```

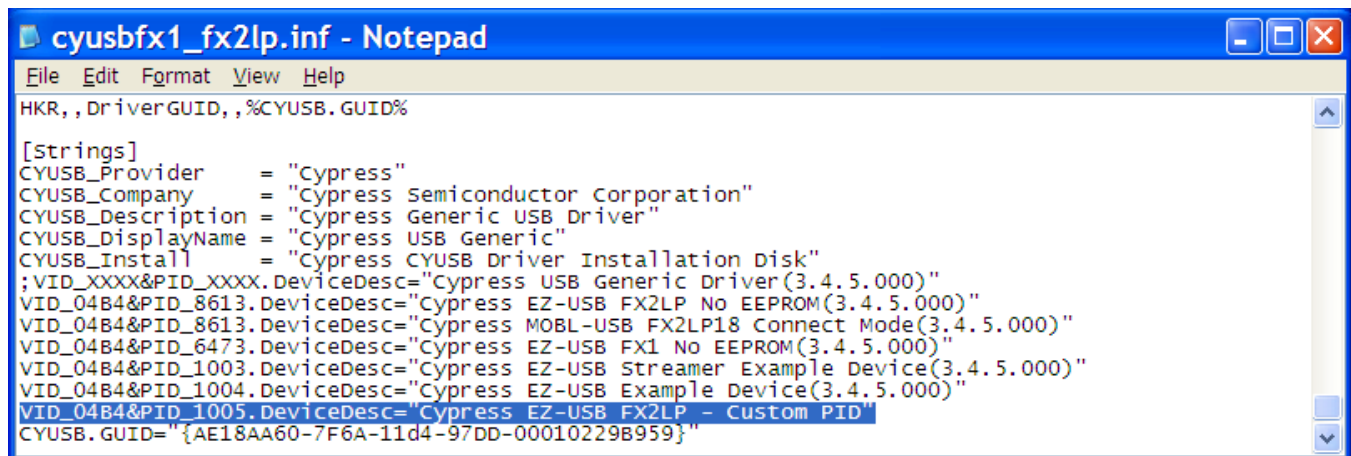
[Device.NTx86]
;Cypress EZ-USB FX2LP default VID/PID & MOBL-USB FX2LP18 Connect Mode(LP18_dvk.1ic)
;%VID_XXXX&PID_XXXX.DeviceDesc%=CyUsb, USB\VID_XXXX&PID_XXXX
;%VID_04B4&PID_8613.DeviceDesc%=CyUsb, USB\VID_04B4&PID_8613

; Cypress EZ-USB FX1 default ID
;%VID_04B4&PID_6473.DeviceDesc%=CyUsb, USB\VID_04B4&PID_6473

;Cypress Streamer example Device.
;%VID_04B4&PID_1003.DeviceDesc%=CyUsb, USB\VID_04B4&PID_1003

;Cypress EZ-USB FX1/FX2LP Example Device.DVK firmware examples use this VID/PID
;%VID_04B4&PID_1004.DeviceDesc%=CyUsb, USB\VID_04B4&PID_1004
;%VID_04B4&PID_1005.DeviceDesc%=CyUsb, USB\VID_04B4&PID_1005
  
```

图 32. 向 Cyusb.inf 添加了自定义 VID 和 PID



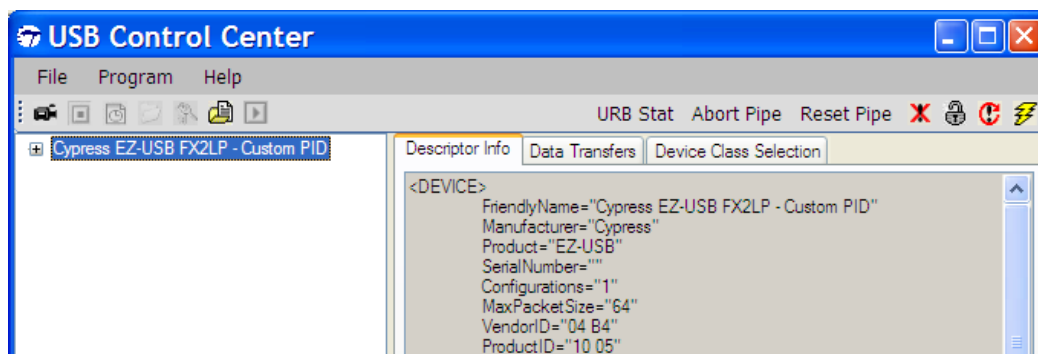
```

HKR,,DriverGUID,,%CYUSB.GUID%

[Strings]
CYUSB_Provider      = "Cypress"
CYUSB_Company       = "Cypress Semiconductor Corporation"
CYUSB_Description   = "Cypress Generic USB Driver"
CYUSB_DisplayName   = "Cypress USB Generic"
CYUSB_Install       = "Cypress CYUSB Driver Installation Disk"
;VID_XXXX&PID_XXXX.DeviceDesc="Cypress USB Generic Driver(3.4.5.000)"
VID_04B4&PID_8613.DeviceDesc="Cypress EZ-USB FX2LP No EEPROM(3.4.5.000)"
VID_04B4&PID_8613.DeviceDesc="Cypress MOBL-USB FX2LP18 Connect Mode(3.4.5.000)"
VID_04B4&PID_6473.DeviceDesc="Cypress EZ-USB FX1 No EEPROM(3.4.5.000)"
VID_04B4&PID_1003.DeviceDesc="Cypress EZ-USB Streamer Example Device(3.4.5.000)"
VID_04B4&PID_1004.DeviceDesc="Cypress EZ-USB Example Device(3.4.5.000)"
VID_04B4&PID_1005.DeviceDesc="Cypress EZ-USB FX2LP - Custom PID"
CYUSB.GUID="{AE18AA60-7F6A-11d4-97DD-00010229B959}"
  
```

进行更改后，将 FX2LP 开发板连接到 PC，然后下载具有 VID 0x4B4 和 PID 0x1005 的固件镜像（.hex 文件）。驱动程序安装器要求驱动程序的位置时，您将它指向到已修改的 **cyusbf1_fx2lp.inf** 文件，然后绑定它到 **CyUSB.sys** 内。FX2LP 被绑定到 **CyUSB.sys** 后，它将出现在 Control Center 的左侧面板，如图 33 所示。目前，您可以通过“控制中心”使用自定义 VID 和 PID 来执行数据传输。

图 33. USB 控制中心中所列出的自定义 FX2LP 器件



文档修订记录

文档标题: AN65209 — FX2LP™入门手册

文档编号: 001-88669

修订版	ECN	变更者	提交日期	变更说明
**	4085304	KKCN	08/02/2013	本文档版本号为 Rev**, 译自英文版 001-65209 Rev*C。
*A	4887194	LIP	08/17/2015	无技术更新。
*B	4904059	WEIZ	09/01/2015	本文档版本号为 Rev*B, 译自英文版 001-65209 Rev*E。
*C	5716139	AESATMP9	04/27/2017	更新徽标和版权。
*D	5883386	ZHUB	09/14/2017	本文档版本号为 Rev*D, 译自英文版 001-65209 Rev*G。

全球销售和设计支持

赛普拉斯公司拥有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。要想查找离您最近的办事处，请访问赛普拉斯所在地。

产品

ARM® Cortex® 微控制器	cypress.com/arm
汽车级产品	cypress.com/automotive
时钟与缓冲器	cypress.com/clocks
接口	cypress.com/interface
物联网	cypress.com/iot
存储器	cypress.com/memory
微控制器	cypress.com/mcu
PSoC	cypress.com/psoc
电源管理 IC	cypress.com/pmic
触摸感应	cypress.com/touch
USB 控制器	cypress.com/usb
无线连接	cypress.com/wireless

PSoC® 解决方案

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

赛普拉斯开发者社区

[论坛](#) | [WICED IoT 论坛](#) | [项目](#) | [视频](#) | [博客](#) | [培训](#) | [组件](#)

技术支持

cypress.com/support

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。



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

赛普拉斯半导体公司，2010-2017 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。没有任何电子设备是绝对安全的。因此，尽管赛普拉斯在其硬件和软件产品中采取了必要的安全措施，但是赛普拉斯并不承担任何由于使用赛普拉斯产品而引起的安全问题及漏洞的责任，例如未经授权的访问或使用赛普拉斯产品。此外，本材料中所介绍的赛普拉斯产品有可能存在设计缺陷或设计错误，从而导致产品的性能与公布的规格不一致。（如果发现此类问题，赛普拉斯会提供勘误表）赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，WICED，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。