



请注意赛普拉斯已正式并入英飞凌科技公司。

此封面页之后的文件标注有“赛普拉斯”的文件即该产品为此公司最初开发的。请注意作为英飞凌产品组合的部分,英飞凌将继续为新的及现有客户提供该产品。

文件内容的连续性

事实是英飞凌提供如下产品作为英飞凌产品组合的部分不会带来对于此文件的任何变更。未来的变更将在恰当的时候发生,且任何变更将在历史页面记录。

订购零件编号的连续性

英飞凌继续支持现有零件编号的使用。下单时请继续使用数据表中的订购零件编号。

为 EZ-USB® FX2LP™/FX3™ 设计 Bulk 传输主机应用

作者: Gayathri Vasudevan

相关器件系列: CY7C6801XA、CYUSB3014

相关文档: 要获取相关文档的完整列表, 请[单击此处](#)。

要想获取本应用笔记的最新版本或相关项目文件, 请访问 <http://www.cypress.com/AN70983>。

赛普拉斯通过 Visual Studio .NET 库为主机 PC 提供了软件开发工具。该库允许通过 Windows 操作系统层进行编码, 这样有助于简化过程。本应用笔记介绍了 .NET 类库并解释了如何创建 Windows 示例, 从而通过 FX2LP 或 FX3 开发套件的 Bulkloop 固件示例来发送和接收数据。

目录

1 简介	1	6 固件	11
2 功能概述	2	6.1 FX2LP Bulkloop 固件	11
3 应用笔记文件夹布局	3	6.2 FX3 Bulkloop 固件	13
3.1 开发板	3	7 故障排除	14
3.2 使用 USB 控制中心将固件下载到开发板	4	8 系统要求	14
4 应用概述	6	8.1 硬件	14
4.1 特性	6	8.2 开发软件	14
4.2 操作说明	7	9 相关文档	14
5 使用赛普拉斯 .NET 库	8	10 总结	15
5.1 添加 CyUSB.dll 的参考	8	11 术语表	16
5.2 源代码概述	9	A 附录: FX2LP/FX3 DVK 驱动程序的 Windows 安装	17
5.3 异步通信	10		

1 简介

本应用笔记演示了如何使用基于 Microsoft .NET 语言的赛普拉斯的库进行主机 PC 端应用开发, 从而与赛普拉斯的 FX2LP 和 FX3 进行通信。通过该库, Visual C#、Visual Basic 或 Visual C++ 程序可以在抽象程度较高的层面上与基于 FX2LP 或 FX3 的器件进行通信。例如, USBDeviceList 和 CyUSBDevice 类 (在 [源代码概述](#) 一节进行了解释) 将器件和端点作为简单类的成员罗列出来。与多种底层的 Win32 API 调用相比, 使用这种高层模块与 USB 器件进行通信是一个很大的改进。例如, 只通过 15 行 C# 代码, 您便可以编程整个 USB 器件树的显示。

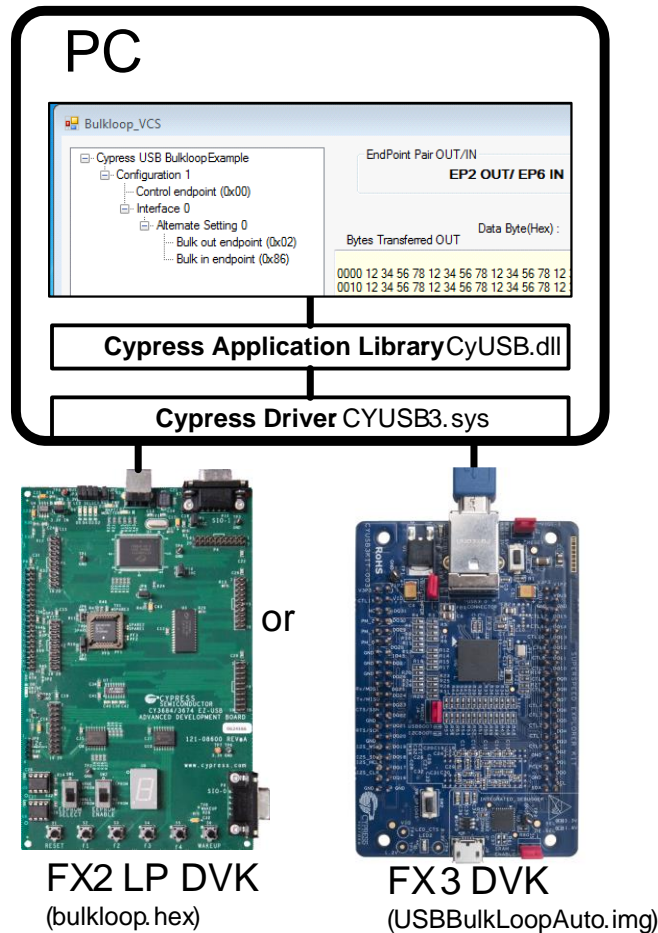
本应用笔记附带的 Visual Studio 解决方案提供了以 Visual C#、Visual Basic 和 Visual C++ 语言创建的示例项目。主机工程 (Project) 由 Microsoft Visual Studio Express 2012 创建并进行了测试。主机应用写入 FX2LP 或 FX3 的 BULK IN 和 BULK OUT 端点。库类与赛普拉斯 USB 驱动程序 CyUsb3.sys 进行通信, 以便访问这些器件和端点。

为方便练习 Windows Visual Studio 代码, 我们随本篇应用笔记提供了适用于 FX2LP 和 FX3 评估板的 Bulkloop 示例固件。该固件接收主机 PC 数据, 并按照指令将数据回送给 PC。

本应用笔记对 USB 连接中的 Windows 和器件进行了概述。它解释了用于与本应用笔记相关的示例代码的软件布局, 并且对 FX2LP 和 FX3 评估板作为检测目标时需要进行的准备工作提供了详细信息。应用笔记还演示了检测编程预编译 Windows 代码的方法。这样, 您可以在不使用 Visual Studio 的条件下运行 Windows 应用。

2 功能概述

图 1. 系统框图



注意：图 1 中所示的 FX3 DVK 只是典型的。可以使用 [CYUSB3KIT-001/CYUSB3KIT-003](#)。

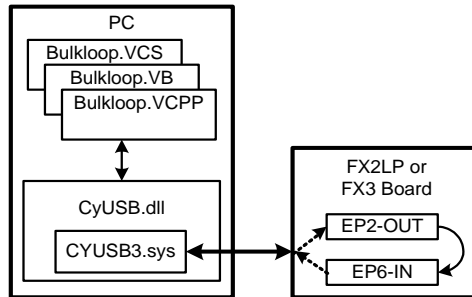
在图 1 中，主机 PC 通过 USB 线缆连接到两个赛普拉斯目标板其中的一个。FX2LP 或 FX3 评估板下载的固件可在端点 2-OUT 上接收 BULK 数据，并将该数据复制到端点 6-IN 缓冲区内，以供 PC 接收使用。在图 1 中，固件文件名称显示在每个评估板的底部。

注意：USB 的方向是以主机为中心。因此，OUT 是指从 PC 到器件，IN 则是指从器件到 PC。

主机应用先调用 CyUSB.dll 库，该库再与赛普拉斯 USB 驱动程序 CYUSB3.sys 进行通信，将数据传输给 EP2-OUT。当主机向 EP6 发送一个 IN 请求时，BULK 数据将通过 CyUsb3.sys 和 CyUsb.dll 库调用返回到主机应用内，并在此时被显示。BULK 数据的传输和接收情况以及已传输的字节数量均显示在一个 PC 文本框中。该数据流程如图 2 所示。

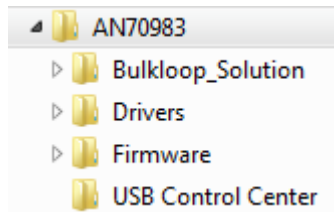
Microsoft Visual Studio 解决方案包括 Visual C#（图 1 中所示的 Bulkloop_VCS）、Visual Basic 和 Visual C++ 的项目。它们生成相同的结果。

图 2. BULK 数据流



3 应用笔记文件夹布局

图 3. 应用笔记文件夹布局



赛普拉斯提供了用以开发固件的软件工具，并且提供了基于.NET 的应用，用以与 FX2LP 和 FX3 进行通信。为便于用户查找，本应用笔记所使用的文件可从附带 zip 文件中获取。要下载完整的工具套件，请访问[赛普拉斯网站](#)。

图 3 显示了该应用笔记文件夹布局。

- **Bulkloop_Solution:** 这是一个 Microsoft .NET 解决方案，使用三种编程语言（Visual Basic、Visual C#或 Visual C++）均可实现 Bulkloop 测试用例。
- **驱动程序:** 第一次将 FX2LP 或 FX3 开发板连接到 PC 时，Windows 要求提供相应驱动程序的位置。该文件夹包含了 Windows XP (“wxp”)、Windows Vista (“wlh”) 和 Windows 7 (“win7”) 的驱动程序。Vista 和 Windows 7 驱动程序适用于 32 位系统 (“x86”) 和 64 位系统 (“x64”)。驱动程序名称为 cyusb3.sys，适用于 FX3 和 FX2LP。
注意: ezusb.sys 和 cyusb.sys 是赛普拉斯所发布的 FX2LP 器件系列的旧版驱动程序。我们不建议将 ezusb.sys 或 cyusb.sys 使用于新设计。更多有关迁移到 cyusb3.sys 的详细信息，请参考从 [CyUSB.sys 迁移到 CyUSB3.sys 的 EZ-USB® FX2LP™ 驱动程序](#)。
- **固件:** 该文件夹包含 FX2LP 和 FX3 评估板的 BULK 回送固件。其中也包含了源代码。因此，您可下载好完整的开发工具套后再进行学习、调整和编译代码。
- **USB 控制中心 (Control Center):** 该 Visual C#应用来自 [EZ-USB FX3 软件开发套件 \(SDK\)](#)，用于将 FX2LP 和 FX3 固件下载到相应的开发板上。准备 USB 目标板

3.1 开发板

FX2LP 开发板包括在 [FX2LP 开发套件](#) 内。FX3 电路板包括在 [FX3 开发套件 \(CYUSB3KIT-001 或 CYUSB3KIT-003\)](#) 内。执行 Windows 代码前，先要准备好开发板，以便实现图 3 右侧显示的回送功能。本应用笔记所附带的.zip 文件包含两个固件目录，一个用于赛普拉斯 FX2LP ([Bulkloop_FX2LP](#))，另一个用于赛普拉斯 FX3 ([Bulkloop_FX3](#))。

下载回送固件前，要确保已经按照表 1 (对于 FX2LP) 和表 2/表 3 (对于 FX3) 准确设置了跳线器和开关。

表 1. FX2LP 评估板设置

跳线器/开关	用途	设置
JP1、JP10	3.3 V 电压提供给 EZ-USB 芯片	IN、IN
JP2	VBUS 供电	IN
JP3	激活 4 个 LED	IN (x4)
JP5	3.3 V 电流监控器	IN
JP6、JP7	FX2LP 寄存器映射 (内部 + 外部 RAM)	OUT、OUT
JP8	辅助唤醒	x
SW1	小容量/大容量 EEPROM	x
SW2	EEPROM 使能	OFF (无 EEPROM)

表 2. FX3 评估板 (CYUSB3KIT-001) 设置

跳线器/开关	用途	设置
PMODE 开关	选择 USB 下载	1、2: OFF (向下) ; 3、4: x
J96、J97	选择 USB 下载	上面两引脚 (2-3)
J98	选择 USB 下载	无连接
J53	VBUS 供电	IN

表 3. FX3 评估板 (CYUSB3KIT-003) 设置

跳线器/开关	用途	设置
J3	开发板由 USB 3.0 VBUS 供电	短接
J4	选择 USB 下载	短接
J5	取消选择外部 SRAM	开路

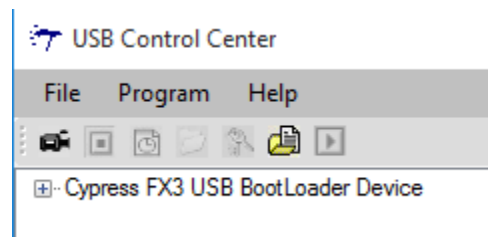
开发板按照设置表配置后，FX2LP 和 FX3 将使用内部逻辑，作为一个 USB 下载器件进行枚举，准备通过 USB 下载用户固件。

3.2 使用 USB 控制中心将固件下载到开发板

双击 USB 控制中心文件夹中的 *CyControl.exe* 文件，将出现 USB 控制中心控制台。图 4 是 FX2LP 或者 FX3 电路板显示在左侧面板中的器件列表中。

当使用了附件中的 *CyUSB3.inf* 文件时，FX3 开发板显示为“Cypress USB Bootloader”（FX2LP 开发板显示为“FX2LP Default Device: NO EEPROM”）。

图 4. USB 控制中心器件列表



如果左侧面板上不显示任何信息，那么可能开发板的配置不准，或 Windows 驱动程序未被安装。更多详细信息，请参考附录。

注意：USB 控制中心的某些版本可能将所有连接的 USB 器件罗列在左侧面板。为了简化显示，请点击 **Device Class Selection**（器件类型选择）选项卡，然后取消选择所有复选框，仅留下 **Devices served by the CyUSB.sys drive...**（与 CyUSB.sys drive... 结合使用的器件）。

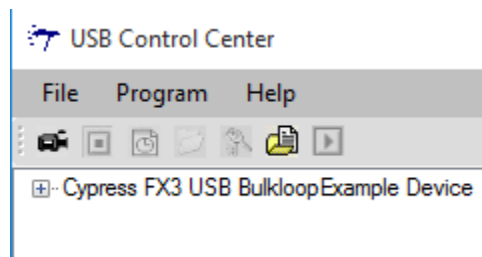
想要将 Bulkloop 固件下载到某个开发板，请选择您在使用的电路板（图 4）。点击 **Program**，选择 FX2 或 FX3，并且选择 **RAM**，以将固件下载到片上编程寄存器中。然后导航到相应文件，位置如下：

FX2: Firmware\Bulkloop_Fx2LP\bulkloop.hex

FX3: Firmware\Bulkloop_FX3\USBBulkLoopAuto\Debug\ USBBulkLoopAuto.img

双击文件来下载固件。USB 下载器件被 USB 回送器件所代替，如图 5 所示。

图 5. Bulkloop 器件



这是 EZ-USB 的重新枚举过程。Bulkloop 固件下载后，开发板自动与 USB 断开，然后当做由下载固件控制的器件重新连接到 USB。此时，FX2LP 和 FX3 开发板就可以作为执行本应用笔记中 PC 示例代码的 USB 回送外设使用。

4 应用概述

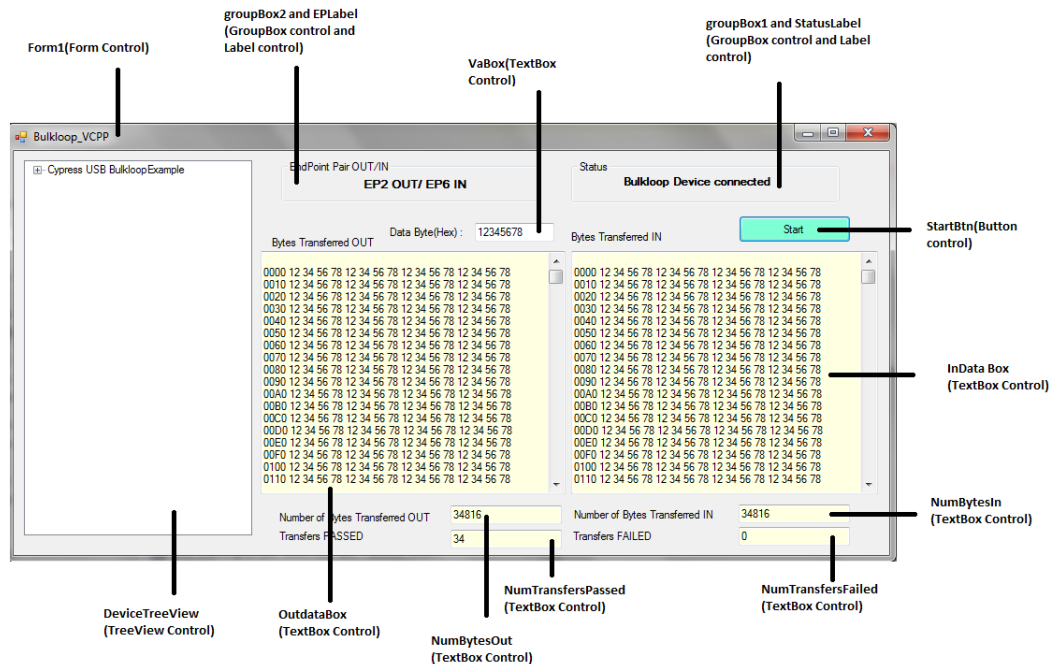
想要检测应用，请在与您.NET 语言相关的目录中双击可执行文件：

- VB_Application\Application\Bulkloop_VB.exe
- VC++_Application\Application\Bulkloop_VCPP.exe
- VCS_Application\Application\Bulkloop_VCS.exe

4.1 特性

图 6 显示了 Bulkloop 应用的用户界面。

图 6. 用户界面



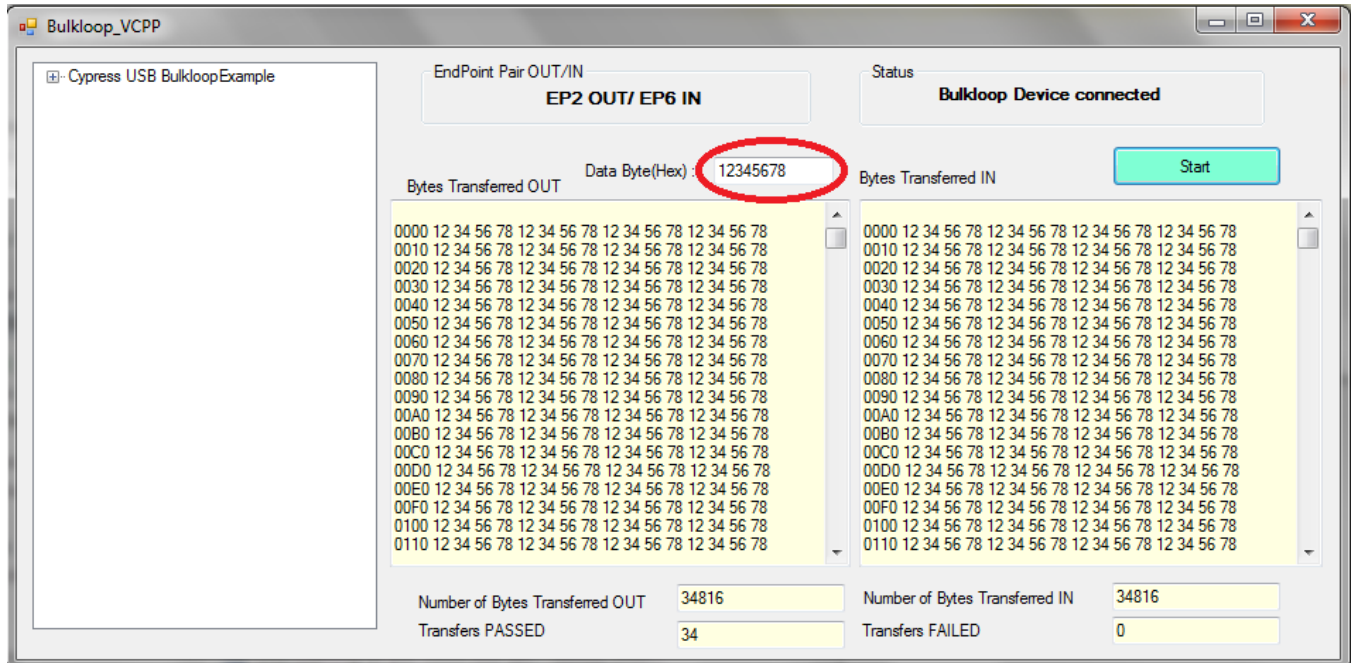
该应用用户界面显示下面各项：

- **状态：** 显示目标器件当前是连接还是断开。
- **端点对 OUT/IN：** 目标器件上的端点对，作为数据传输到目标器件或从目标器件传输的 OUT 和 IN 端点。该内容仅用于提供信息（并非用户可编辑），反应了源代码中的设置。
- **数据字节（十六进制）：** 用户可选的数据模式（最大长度为八个半字节或四个字节），在 Bulkloop 过程中，将以回送模式将这些数据字节传输。
- **字节传输 OUT：** 从主机 PC 传输到目标器件的字节记录。每一行以字节在行中的初始地址开始。
- **字节传输 IN：** 与 OUT 相同，但传输的字节是从目标器件到主机 PC 的 IN 字节。
- **字节传输 OUT 的数量：** 从主机 PC 传输到目标器件的字节数量。
- **字节传输 IN 的数量：** 从目标器件传输到主机 PC 的字节数量。
- **启用/停止：** Bulkloop 过程启用/停止按键。
- **DeviceTreeView：** 显示连接到 Cyusb3.sys 的所有器件。

4.2 操作说明

当主机检测到回送器件时，状态框会指示“连接”，并且左面板显示器件树，用以说明 USB 器件的内部结构。您可以输入某个字节模式用以传输，或使用默认的 0x12345678 字节模式。点击 **Start** 按键；回送器件将连续接收和发送数据包。

图 7. Bulkloop 应用的数据传输



点击 **Stop** 以停止传输。对于不同主控制器和 PC，字节计数器均可用于测量该应用中的 USB 传输带宽。

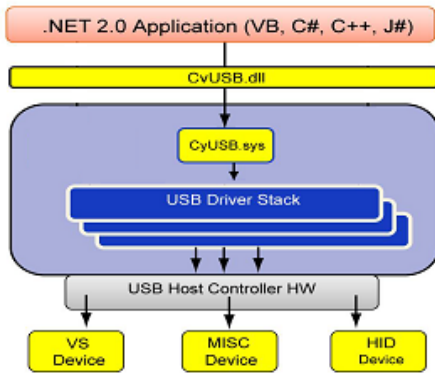
文档的剩下部分对 .NET 语言应用提供了详细说明。

5 使用赛普拉斯.NET 库

5.1 添加 CyUSB.dll 的参考

CyUSB.dll 是一个托管 Microsoft .NET 的类库，它为 USB 设备提供了一个高级编程接口。用户可通过 Visual Basic.NET、C#、Visual J#和托管 C++ 等任何 Microsoft Visual Studio.NET 语言对它的类和函数进行访问。

图 8. 使用 CyUSB.dll 的应用开发



想要使用赛普拉斯库，请执行下面两个步骤：

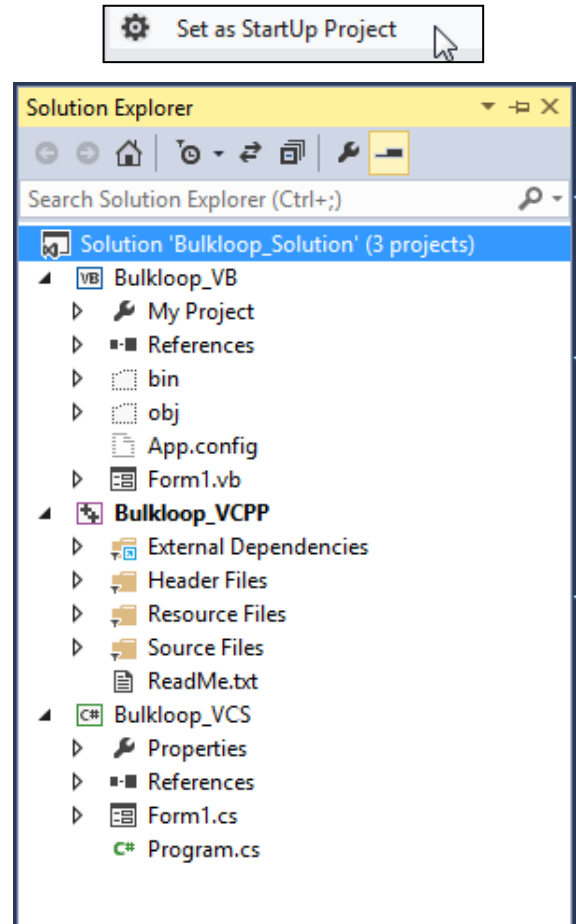
1. 在您的项目中，添加 *CyUSB.dll* 的参考。
2. 在您的源文件中添加一行，以将这一库包括在程序内。

图 9 显示了库参考在 Bulkloop_Exerciser 解决方案的 Visual Basic、VC++和 VC#项目中的位置。在库参考扩展时，如果不显示 CyUSB，请右键点击项目名称，选择“参考”，并浏览包含 *CyUSB.dll* 的文件夹。本应用笔记所提供的项目使用的路径为 C:\AN70983\CyUSB.dll。如果您将应用笔记文件安装在其他地方，那么需要更新参考，以确保它们与您的文件位置匹配。

当您点击 **Build** 按键时，解决方案中的所有项目将编译；任何错误可能来自于三个工程。您可以在 **build** 时取消包含其他两个工程，来限制您所使用的工程的错误信息。在 **Solution Explorer**（解决方案资源管理器）屏幕中，右键点击解决方案名称，选择 **Properties**，然后取消选择需要从“**Build**”操作中移除的项目。

如果在‘**Build**’操作中有多个项目，请通过右击项目名称选择语言，并将其设为启动项目。这是一种在多个语言中快速切换的方法。

图 9. 将 CyUSB.dll 参考添加到三个项目



根据语言类型，根据表 4 在文件顶部的代码中添加一个参考。

表 4. NET 语言的参考句法

语言	文件	添加行
Visual Basic	form1.vb	Imports CyUSB
Visual C#	form1.cs	Using CyUSB
Visual C++	form1.h	#using "..\..\Application\CyUSB.dll" (or the respective path in your computer)

5.2 源代码概述

下表对主机应用代码的功能和重要步骤进行了解释。

表 5. 代码概述

操作/功能	说明	CyUSB.dll API
对象初始化	<ul style="list-style-type: none"> 创建并初始化 USBDeviceList、CyUSBDevice、CyBulkEndPoint 类别以及其他所需变量的实例。 “usbDevices”代表可通过类库访问的 USB 器件的动态列表。它自动生成的 USBDevice 对象代表了所有连接到 Cyusb3.sys 的器件。这些器件由“DEVICES_CYUSB”器件选择器掩码服务。 	<ul style="list-style-type: none"> USBDeviceList 代表可通过类库访问的 USB 器件的动态列表。创建 USBDeviceList 实例时，它自动生成的 USBDevice 对象代表了所有由指定器件选择器掩码提供服务的 USB 器件。这些 USBDevice 对象均已正确初始化以供随时使用。 CyUSBDevice 类代表一个连接到 CyUSB.sys 器件驱动程序的 USB 器件。 CyBulkEndPoint 是 CyUSBEndPoint 抽象类的子类。
器件连接和断开检测	<ul style="list-style-type: none"> 当主机与目标硬件连接时，event_handler 函数 usbDevices_DeviceAttached() 被调用。 当主机与目标硬件断开连接时，event_handler 函数 usbDevices_DeviceRemoved() 被调用。 然后，两个 event_handler 函数将调用 RefreshDevice()。 	—
RefreshDevice()	<ul style="list-style-type: none"> 根据器件的 VID 和 PID，将“myDevice”实例化为所需器件。本示例中的 Bulkloop 固件使用的 VID/PID：0x04B4/0x00F0。 如果多个设备使用了相同的 VID/PID，那么“myDevice”将实例化为最先连接的器件。 使能或禁用状态标签和 Start/Stop 按键。 检测到所使用的器件时，需要分别将“BulkOutEndpt”和“BulkInEndpt”实例化为 EP2 和 EP6 端点，因为本示例中的 Bulkloop 固件使用了相同的端点。 清除器件树，并使用连接到 Cyusb3.sys 的最新器件列表对其进行更新。 	<ul style="list-style-type: none"> USBDeviceList[int VID, int PID]: 该索引运算符根据列表中 USBDevice 对象的 VendorID 和 ProductID 提供对 USBDeviceList 中各器件的访问。 EndpointOf(byte addr): 返回地址为 addr 的 CyUSBEndpoint 对象。所有端点的地址与 addr 不同时，将回送空字符 (null)。
数据传输操作	<ul style="list-style-type: none"> 通过点击 Start 按键，可启用 StartBtn_Click() 事件处理函数。 该函数先从应用用户界面上的‘Data Bytes(Hex)’中采用用户输入的数值，将按键状态改为“Stop”，并启用“tXfers”流程（数据通信在此实际上进行）。 其他父进程只关注应用用户界面事件和其他 OS 事件。 如果进程正在执行，那么该事件处理函数将终止进程并将按键文字返回到“Start”。 	—
TransfersThread()	<ul style="list-style-type: none"> 调用 SetOutputData() 来设置 OUT 传输的数据。 启动进程后，通过 XferData() API 开始按照 OUT 和 IN 方向进行数据传输。 	XferData(ref byte[] buf, ref int len): XferData 用于接收 buf 中 len 个 byte 的数据，或者向 buf 发送 len 个 byte 的数据。这是库中用于传输数据的主要 I/O 方法。它实现了同步（阻塞）I/O 操作，并且在数据传输完成或者超时时间结束之前不会执行返回。如果超时时间完成前，数据操作成功完成，那么会返回正确 (true)。实际上所传输的字节数量被输入到 len 内。
SetOutputData()	<ul style="list-style-type: none"> 该函数从应用用户界面的‘Data Bytes(Hex)’中采取用户输入的数据，然后使用 OUT 传输中所发送的字节来填充‘outData[]’缓冲区。 	—
StatusUpdate()	<ul style="list-style-type: none"> 这是用于更新 UI 的 callback 函数，由 TransfersThread 调用。它使用传输字节和传输字节的数量来更新 UI 文本框。 	—

5.3 异步通信

该应用通过异步通信在主机和目标器件间传输数据。使用该方法，数据传输函数 ‘XferData’ 将不执行返回操作，直到完整的数据被传输完为止。也就是说，应用一直被阻塞，直到数据传输完成为止。然后，开始进行下一个数据传输操作。该方法可用于大多数应用。但是对于需要使用全可用数据传输带宽的应用，应该使用更好的方法，即异步通信。异步传输比较复杂，适用于高级用户。赛普拉斯 USBSuite 的 ‘Streamer’ 示例揭示了异步通信方法的使用。

6 固件

固件对端点和接口进行配置，从而进行 Bulk 数据回送操作。进行 Bulk 传输时需要配置两个端点：OUT 和 IN 端点。从主机传输的数据被存储在 OUT 端点缓冲区内。该数据被传输到 IN 端点上，然后再根据要求送回主机。

6.1 FX2LP Bulkloop 固件

下面的代码段说明了 FX2LP Bulkloop 固件的重要代码函数。

操作	说明	代码段	函数名称	文件名称
描述符信息	<ul style="list-style-type: none"> ■ VID: 0x04B4 ■ PID: 0x00F0 ■ 器件类型: 供应商 ■ 配置数量: 1 ■ 接口数量: 1 ■ 端点数量: 2 ■ 提供的端点: <ul style="list-style-type: none"> □ EP2 BULK OUT, 最大数据包为 512 字节 □ EP6 BULK IN, 最大数据包为 512 字节 	—	—	dscr.a51
初始化	<ul style="list-style-type: none"> ■ 基于 FX2LP 的目标器件的组件初始化是由 TD_Init() 函数执行的。该函数通过 main()调用, 用于进行初始化。 	—	TD_Init()	bulkloop.c
	将 CPU 时钟设置为 48 MHz。	<pre>CPUCS = ((CPUCS & ~bmCLKSPD) bmCLKSPD1);</pre>		
	<ul style="list-style-type: none"> ■ 配置 FX2LP 的接口。 ■ FX2LP 被配置为从设备 FIFO 模式。 ■ 接口时钟选择为内部 48 MHz 时钟。 	<pre>IFCONFIG = 0x40;</pre>		
	端点的配置如下: <ul style="list-style-type: none"> ■ 端点 2 — OUT、Bulk、双缓冲、512 字节 ■ 端点 6 — IN、Bulk、双缓冲、512 字节 	<pre>EP2CFG = 0xA2; SYNCDELAY; EP6CFG = 0xE2; SYNCDELAY;</pre>		
	<ul style="list-style-type: none"> ■ 配置后, OUT 端点需做好准备接收主机发来的数据包。因为端点经过了双缓冲, 因此为了使端点做好准备, 需要跳过两个数据包。实际上, 准备工作指的是释放缓冲区, 从而能够接收主机发来的数据包。 ■ 通过向字节计数寄存器位 7 写入 '1', 可实现跳过数据包。 	<pre>EP2BCL = 0x80; SYNCDELAY; EP2BCL = 0x80; SYNCDELAY;</pre>		
<ul style="list-style-type: none"> ■ 在 TD_Poll 中使能用于数据传输的 AUTO 指针。 	<pre>AUTOPTTRSETUP = 0x01;</pre>			

	说明	代码段	函数名称	文件名称
数据回送	<ul style="list-style-type: none"> ■ Bulkloop 操作由 TD_Poll() 函数实现。器件处于闲置状态时，该函数被重复调用。 	—	TD_Poll()	bulkloop.c
	<ul style="list-style-type: none"> ■ 检测端点 2 上是否存在数据。该操作是通过读取端点状态寄存器 (EP2468STAT) 中的端点 2 空位实现的。 ■ 如果端点 2 存在数据 (从主机传输来)，检查端点 6 是否能够接收该数据。该操作是通过读取端点状态寄存器中满位所指示的端点 6 ‘Full’ 标志实现的。 ■ 如果端点 6 未滿，将传输数据。 	<pre>if (!(EP2468STAT & bmEP2EMPTY)) { if (!(EP2468STAT & bmEP6FULL)) { } } </pre>		
	<ul style="list-style-type: none"> ■ 为了传输数据，数据指针被初始化到相应缓冲区内。第一个自动指针被初始化到端点 2 FIFO 缓冲区的第一个字节。 	<pre>APTR1H = MSB(&EP2FIFOBUF); APTR1L = LSB(&EP2FIFOBUF); </pre>		
	<ul style="list-style-type: none"> ■ 第二个自动指针被初始化到端点 6 FIFO 缓冲区的第一个字节。 	<pre>AUTOPTRH2 = MSB(&EP6FIFOBUF); AUTOPTRL2 = LSB(&EP6FIFOBUF); </pre>		
	<ul style="list-style-type: none"> ■ 所传输的字节数量是从端点 2 的字节计数寄存器中读取得到的。EP2BCL 和 EP2BCH 寄存器包含了主机写入 FIFO 缓冲区的字节数量。这两个寄存器提供了传输到 FIFO IN 和 OUT 缓冲区的数据字节的数量，直到数据停止传输到外设为止。 	<pre>count = (EP2BCH << 8) + EP2BCL; </pre>		
	<ul style="list-style-type: none"> ■ 因为自动指针已被使能，因此指针将自动递增并且 EXTAUTODAT2 = EXTAUTODAT1 语句会将数据从端点 2 传输到端点 6。每当实现该语句时，自动指针都会递增。该语句重复执行，从而将各字节从端点 2 传输到端点 6。 	<pre>for(i = 0x0000; i < count; i++) { EXTAUTODAT2 = EXTAUTODAT1; } </pre>		
	<ul style="list-style-type: none"> ■ 数据传输后，端点 2 需要重新准备，以接收主机的新数据包。端点 6 已经就绪，即：FIFO 缓冲区可用，主机可以读取端点 6 上的数据。 	<pre>EP6BCH = EP2BCH; SYNCDELAY; EP6BCL = EP2BCL; SYNCDELAY; EP2BCL = 0x80; </pre>		—

6.2 FX3 Bulkloop 固件

下面的代码段对 FX3 Bulkloop 固件的代码函数进行了说明。

	说明	FX3 API	函数名称	文件名称
描述符信息	<ul style="list-style-type: none"> ■ VID: 0x04B4 ■ PID: 0x00F0 ■ 器件类型: 供应商 ■ 配置数量: 1 ■ 接口数量: 1 ■ 端点数量: 2 ■ 提供的端点: <ul style="list-style-type: none"> □ EP2 BULK OUT, 最大数据包为 1024 字节 (对于 SS) 或 512 字节 (对于 HS) □ EP6 BULK IN, 最大数据包为 1024 字节 (对于 SS) 或 512 字节 (对于 HS) 	—	—	cyfxbulkpdscri.c
初始化	<ul style="list-style-type: none"> ■ FX3 器件被初始化 ■ 对 FX3 的 61 个可用引脚设置功能模式。 ■ 然后, RTOS 内核由 main()调用。这是无返回调用, 并会设置 Threadx 内核。该函数应该是 main()函数最后调用的函数。 	CyU3PDeviceInit() CyU3PDeviceConfigureIOMatrix() CyU3PKernelEntry()	main()	cyfxbulkpauto.c
	<ul style="list-style-type: none"> ■ 器件和所有模块被初始化后, FX3 应用定义函数将由系统模块调用。可在这里创建应用进程和其他所需 OS 基元。至少要创建一个进程。但不应该明确调用该函数。 ■ BulkLpAppThread_Entry 即为所创建的应用。 	CyU3PThreadCreate()	CyFxBulkLpAppDefine()	
	<ul style="list-style-type: none"> ■ CyFxBulkLpAppInDebugInit()可初始化调试模块 ■ CyFxBulkLpAppInInit()可初始化 Bulkloop 应用 	—	BulkLpAppThread_Entry()	
	<ul style="list-style-type: none"> ■ 启用 USB 3.0 平台的 USB 器件驱动程序并创建控制端点所需的 DMA 通道 ■ 向 USB 驱动程序注册一个 USB 设置请求处理程序。 ■ 向 USB 驱动程序注册一个 USB 事件 callback 函数。CyFxBulkLpAppInUSBEventCB 在该应用中被注册。 ■ 注册一个 USB 3.0 LPM 请求处理程序 callback。 ■ 向 USB 驱动程序注册一个 USB 描述符。驱动程序能够记住每种受支持类型中的一个描述符, 即: 它能够记住多达八个不同的字符串描述符。驱动程序仅保留了传输到该函数的描述符指针; 它不会复制描述符。因此, USB 驱动程序运行时, 调用函数不应释放这些描述符缓冲区。 ■ 使能/禁用 USB 连接。该函数用于使能或禁用 USB 3.0 平台上的 USB PHY, 并控制着同 USB 主机的连接。 	<ul style="list-style-type: none"> ■ CyU3PUsbStart() ■ CyU3PUsbRegisterSetupCallback() ■ CyU3PUsbRegisterEventCallback() ■ CyU3PUsbRegisterLPMRequestCallback() ■ CyU3PUsbSetDesc() ■ CyU3PConnectState() 	CyFxBulkLpAppInInit()	
	<ul style="list-style-type: none"> ■ CyFxBulkLpAppInStart ()在设置配置事件发生时被调用, 这时候 USB 端点也得到配置。 	CyU3PSetEpConfig()	CyFxBulkLpAppInStart()	

	说明	FX3 API	函数名称	文件名称
数据回送	<ul style="list-style-type: none"> 创建一个 DMA 通道。DMA 通道需要使用一个发送端套接字、一个接收端套接字、一套缓冲区和一个 callback 函数。在该示例中，在 EP2（作为发送端套接字）和 EP6（作为接收端套接字）之间创建了一个自动通道。每当 EP2 接收数据时，数据将自动被传输给 EP6。 在 DMA 通道上设置传输。函数会在所选的 DMA 通道上启用数据操作。 	<ul style="list-style-type: none"> CCyU3PDmaChannelCreate() CyU3PDmaChannelSetXfer() 	CyFxBulkLpApplnStart()	

7 故障排除

1. 如果不正确添加和引用 CyUSB.dll，主机应用程序将无法编译，并且 CyUSB.dll API 将无法访问。请按照[添加 CyUSB.dll 的参考](#)一节中所提供的指导进行操作。
2. 如果任何 CyUSB API 无法实现，请检查相关 API 的返回代码。安装 SuiteUSB 后，返回代码值和其含义被记录在位于“<InstallDirectory>\Cypress Suite USB 3.4.x\CyUSB.NET\CyUSB.NET”的 CyUSB.dll 文档中。
3. 如果 XferData/BeginDataXfer API 失败，请使用“UsbdStatus”或“UsbdStatusString()”来获取最后一次调用 XferData 或 BeginDataXfer API 的错误代码。
4. 如果 API 无法访问器件，请确保正确分配器件句柄。
5. 如果在 Visual Studio 2010 (VS 2010) 中未打开项目，请在 VS 2010 中创建新项目，移动源文件，然后将 CyUSB.dll 库添加到项目内。现有功能将适用于 VS 2010 及更高版本。

8 系统要求

8.1 硬件

8.1.1 FX2LP

FX2LP DVK (CY3684) 开发板作为该 FX2LP 示例中的开发和检测平台使用。安装好 FX2LP DVK 后，您可以通过硬件文件夹获取 DVK 的详细原理图。更多有关开发板的信息，请参考 FX2LP DVK 所提供的 EZ-USB_入门文档中‘EZ-USB 高级开发板’部分的内容。

8.1.2 FX3

FX3 DVK (CYUSB3KIT-001 或 CYUSB3KIT-003) 开发板作为该 FX3 示例中的开发和检测平台使用。安装 FX3 DVK 后，您可以通过硬件文件夹获取 DVK 的详细原理图。

8.2 开发软件

- Microsoft .NET 框架 4.5
- ControlCenter — 通过安装赛普拉斯 USB 套件获取。
- Keil µVision 2: 4 KB 评估版本适用于 CY3684 DVK。如果需要完整的版本，请联系 Keil 公司。
- Eclipse IDE — 通过安装 FX3 SDK 获取。
- Microsoft Visual Studio 2010 或 2012。

9 相关文档

- [UsbSuiteQuickStartGuide](#) 对开始学习使用 CyUSB.dll 开发 Visual C# 的主机应用和使用 CyAPI.lib 开发 Visual C++ 的主机应用非常有用。
- 可通过赛普拉斯 USB 套件获取编程者参考 C# 库 (CyUSB.NET.pdf)。
- 赛普拉斯 USB 套件和 FX3 SDK 还提供了其他应用示例，如：

- **Streamer:** 可用于检测数据传输。
- **Control Center:** 用户接口，用于同 CyUSB3.sys 适用的 USB 器件进行通信。
- **EZ-USB FX2LP/FX3** — 在 **Linux** 上开发 **Bulk-Loop** 示例介绍了如何使用 **libusb** 进行开发基于 **Linux** 操作系统的 USB 主机应用，从而能够使用于赛普拉斯 **EZ-USB FX2LP/FX3** 产品。本应用笔记仅供参考。赛普拉斯还提供可在 **FX3 SDK** 网站上得到的 **Linux SDK**。
- **EZ-USB FX2LP** — 使用 **LIBUSB** 开发 **MAC OS X** 的 **USB** 应用说明了如何使用 **libusb-1.0** 来开发 **MAC OS X 10.6/10.7** 的 USB 主机应用（**Cocoa** 应用），从而能够使用于赛普拉斯 **EZ-USB FX2LP** 产品。本应用笔记仅供参考。赛普拉斯还提供可在 **FX3 SDK** 网站上得到的 **MAC SDK**。
- 有关 **USB SuperSpeed** 和 **USB Hi-Speed** 代码示例的完整列表的信息，请依次访问 <http://www.cypress.com/101781> 和 <http://www.cypress.com/101782>。

10 总结

本应用笔记介绍了使用 **Microsoft .NET** 语言和赛普拉斯提供的库来构建主机 **PC** 应用的各种方法，以便与赛普拉斯的 **FX2LP** 和 **FX3** 器件进行通信。

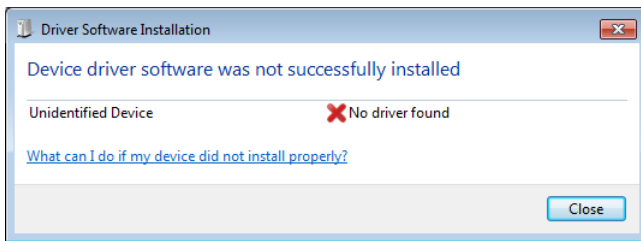
11 术语表

术语	含义
Microsoft Visual Studio	Microsoft Visual Studio 是 Microsoft 的集成开发环境 (IDE)。它用于开发使用本机代码和托管代码的控制平台和图形用户界面, 以及 Windows Forms 或 WPF 应用, 适用于 Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight 支持的平台。
Microsoft Visual C#	Microsoft 的 C# 语言版本, 专门设计用于托管环境。
Microsoft Visual C++	Microsoft 的商业 IDE 产品, 适用于 C、C++ 和 C++/CLI 编程语言。
Microsoft Visual Basic	Visual Basic .NET (VB.NET) 是一个面向对象的电脑编程语言, 可认为它是典型的 Visual Basic (VB) 见过大型改进得到, 适用于 .NET 框架。Visual Basic 是 Microsoft 的第三代事件驱动编程语言和 IDE, 适用于 Microsoft 的 COM 编程模式。
CYy3684 DVK	适用于 EZ-USB FX2LP 系列的赛普拉斯开发套件。更多详细信息, 请参考赛普拉斯网站。
CYUSB3KIT – 001/ CYUSB3KIT – 003 (EZ-USB® FX3™ SuperSpeed Explorer 套件)	适用于 EZ-USB FX3 的赛普拉斯开发套件。更多有关信息, 请参考 CYUSB3KIT-001 或 CYUSB3KIT-003 。
EZ-USB	赛普拉斯所有高速 USB 产品 (如 FX2LP) 的统称。
FX2LP	赛普拉斯高速 USB 控制器。更多有关信息, 请参考 CYUSB3KIT-001 或 CYUSB3KIT-003 。
FX3	赛普拉斯的 EZ-USB FX3 是新一代 USB 3.0 外设控制器, 可提供集成且灵活的功能。更多详细信息, 请参考赛普拉斯网站。
赛普拉斯 USB 套件	适用于 Visual Studio 的赛普拉斯 USB 开发工具。更多详细信息, 请参考赛普拉斯网站。
CyUSB.dll	赛普拉斯库, 适用于采用了 .NET 技术的软件开发。更多详细信息, 请参考赛普拉斯 USB 套件中的编程者参考 C# 库 (CyUSB.NET.pdf)。
Cyusb3.sys	适用于 USB 通信的赛普拉斯 Windows 驱动程序。更多详细信息, 请参考赛普拉斯 CyUsb3.sys 编程者参考。
USB 控制中心	赛普拉斯应用实现通用的 USB 功能, 如适用于器件识别、代码镜像下载等。该应用可通过赛普拉斯 USB 套件获取。
Keil uVision	4 KB 评估版本适用于 CY3684 DVK。如果需要完整的版本, 请联系 Keil。
Eclipse IDE	FX3 SDK 为 C/C++ 开发者提供的 Eclipse IDE 作为开发工具的一部分。该 IDE 包括 Eclipse 基础平台和 CPP 功能。通过该固件开发环境, 您可以为 FX3 进行固件应用开发、构建和调试。

A 附录：FX2LP/FX3 DVK 驱动程序的 Windows 安装

如果您的 Windows 计算机上尚未安装 FX2LP/FX3 DVK，那么第一次将 DVK 连接到计算机时，您会看到下面信息。

图 10. 驱动程序软件安装



关闭信息框并导航到 Windows 设备管理器。要实现该操作，请点击 **Start** 按钮，在右边屏幕右击 **Computer** 项，然后选择 **Properties** 来打开系统信息。点击左边屏幕上的 **Device Manager**。在 Device Manager 屏幕中，FX2LP 项显示为“Unknown Device”，FX3 为“WestBridge”。

图 11. FX2LP 为 Unknown Device

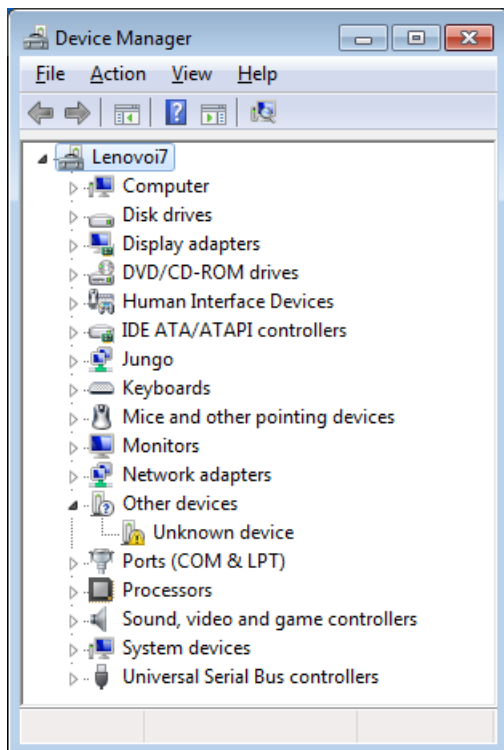


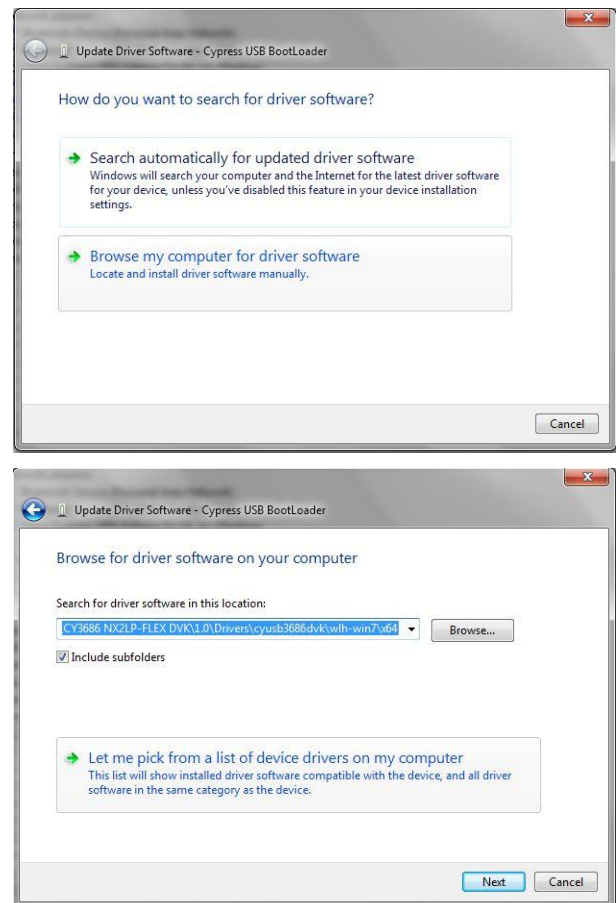
图 12. FX3 为 WestBridge



右击器件（带有惊叹号），然后选择 **Update Driver**。

选中 **Browse my computer for driver software** 选项。通过浏览，选择所需的驱动程序软件文件夹，然后点击 **Next**。

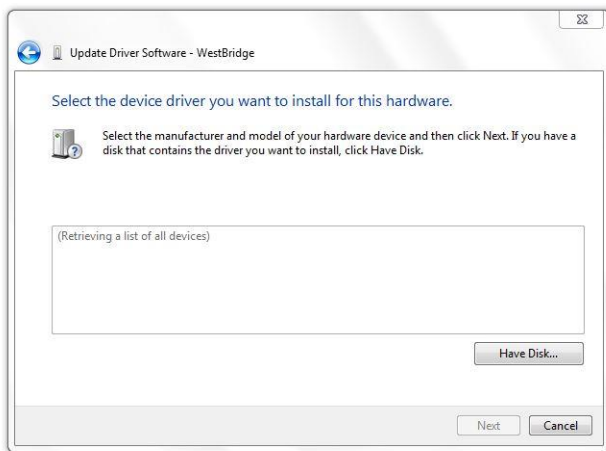
图 13. 更新驱动程序



点击 **Have Disk** 按钮，然后浏览到 *AN70983_Project\Drivers* 以选择 *CyUSB3.inf*，然后再选择 **OK**。根据操作系统，使用文件夹 (*AN70983_Project\Drivers*) 中的 *inf* 文件，具体情况如下表所示。

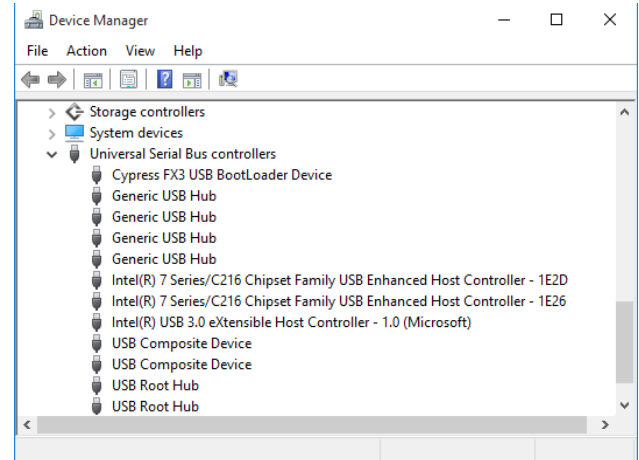
操作系统	文件夹路径
Windows XP 32 位	wxp\x86
Windows 7 32 位	win7\x86
Windows 7 64 位	win7\x64
Windows Vista 32 位	wlh\x86
Windows Vista 64 位	wlh\x64

图 14. 点击 ‘Have Disk’，以选择 CyUSB3.inf



点击 **Next**，然后继续安装驱动程序。驱动程序安装后，**Device Manager** 窗口将移除带黄色惊叹号的器件，并显示成功连接到驱动程序的器件。

图 15. 连接到驱动程序后的赛普拉斯器件



文档修订记录

文档标题：AN70983 — 为 EZ-USB® FX2LP™/FX3™ 设计 Bulk 传输主机应用

文档编号：001-98019

版本	ECN	变更者	提交日期	更改说明
**	4802493	LYAO	07/14/2015	本文档版本号为 Rev**, 译自英文版 001-70983 Rev*C。
*A	5796042	AESATMP8	07/04/2017	更新徽标和版权。
*B	5967002	YYCA	11/21/2017	本文档版本号为 Rev*B, 译自英文版 001-70983 Rev*F。
*C	6264220	SSAS	07/27/2018	无变更。Sunset Review.

销售、解决方案以及法律信息

全球销售和 design 支持

赛普拉斯公司拥有一个由办事处、解决方案中心、原厂代表和经销商组成的全球性网络。如欲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/pmhc
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 MCU](#)

赛普拉斯开发者社区

[社区](#) | [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。



赛普拉斯半导体公司
198 Champion Court
San Jose, CA 95134-1709

© 赛普拉斯半导体公司，2011-2018 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安全漏洞的责任，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。文件使用者应负责正确设计、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、设定为或授权使用作武器操作、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。