使用 EZ-USB™ FX2LP 创建自己的 USB 供应商控制指令

关于本文档

范围及目的

本应用笔记向您介绍了如何使用 EZ-USB™ FX2LP 来创建一个自定义的 USB 器件并使用供应商要求与它通讯。相关代码示例实现了六种不同的供应商命令，并且可以使用 EZ-USB™ FX2LP 开发板 (CY3684) 和赛普拉斯 USB 控制中心应用程序进行测试。

更多代码示例? 我们听到你了.

要访问各种 EZ-USB™ FX2LP 代码示例，请访问我们的 **USB High-Speed Code Examples** 网页.

相关项目

有

相关芯片系列

CY7C68013A/CY7C68014A/CY7C68015A/CY7C68016A

目录

关于本文档 **1**
目录 **1**
1 简介 **3**
2 供应商控制指令 **4**
2.1 0：标准请求 4
2.2 01：类别请求 5
2.3 02：供应商请求 5
3 供应商控制指令示例代码 **6**
4 测试驱动程序 **7**
5 实现方式 **12**
5.1 fw.c 12
5.2 Custom_Request_Descriptors.a51 12
5.3 EEPROM.C 12
5.4 EZUSB.LIB 12
5.5 Custom_Requests.c 13
5.6 USBJmpTb.OBJ 13
6 代码分析 **14**
7 调试代码 **16**
8 自定义 USB 控制中心 **18**
9 最后一步 **19**

目录

10 资源以及其他信息 20
11 总结 21
文档修订记录 22

1 简介

各种标准的 USB 产品 (如键盘、鼠标和磁盘驱动器) 都要遵守由 USB 实施者论坛 (USB-IF) 制订的器件类别。这些器件类别提供了一个标准化的基础，从而能成功实现 USB 产品的应用。

但 USB 设计师们认为，不是所有器件都能符合相关的标准类别。另外，在某些情况下，能够有条件地为某个产品类别添加功能。为了处理好这些情况，设计师们已经创建好了一种被称为“供应商特定”的请求类型。

该 USB 规范中并未对这种请求类型的特殊代码以及使用这些请求的字段中的特殊值进行定义 – 因此每个供应商需要定义这些内容。本应用笔记还提供了 EZ-USB™ FX2LP 固件源代码，用于说明 USB 供应商控制指令的使用情况。

通过使用附件示例中的新型 USB 指令，可以将一个十六进制数字发送给 7 段读取值，从而点亮四个 LED，并读取板上 EEPROM 中的值。您可以轻松地将这个新建的指令集 (和其他的) 添加到正在开发的 EZ-USB™ FX2LP 代码中，用于支持调试。

另一个示例为：一个数据移动器应用可以使用 BULK IN 和 BULK OUT 端点来传输数据，并使用各个自定义指令来实现频带外控制，而不会影响各个 BULK 端点中的数据。自定义指令包含“start sending” (启用发送) 、“stop sending” (停止发送) 和 “loop data” (环路数据) 。将数据传输和控制传输自然分开的方法使设计更加整齐有效。

要想了解它的工作原理，首先要检查由 USB 定义为一个器件请求的 8 个字节。USB 使用该请求启动器件中的所有 USB 操作。

2 供应商控制指令

本章节介绍了一个 USB 器件请求的格式，并描述了基于 bmRequestType 值的不同 USB 请求。

Table 1 USB 器件请求封装格式

字节	字段	含义
0	bmRequestType	请求类型
1	bRequest	实际请求
2	wValue	因请求不同而存在变化
4	wIndex	因请求不同而存在变化
6	wLength	数据字节数

Table 1 显示的是 USB 器件请求包格式。每个请求都包含一个 8 字节的数据包。各个请求始终使用了控制端点 EP0。该数据包是主要的 USB 调度器，它决定了请求的方向、类型和接收器件。各标准请求传输都是在 SETUP 数据包内进行的，并且可选择性地在其后面紧接 DATA 数据包 (若需要) 。

Table 2 bmRequestType 位定义

位	字段	值
7	方向	0：从主机到器件 (OUT) 1：从器件到主机 (IN)
6..5	类型	0：标准 1：类别 **2：供应商** 3：保留
4..0	接收方	0：器件 1：接口 2：端点 3：其他 4-31：保留

Table 2 显示的是第一字节的格式 (即为 bmRequestType) 。位 7 表示的是请求方向。位 6 和为 5：指定请求*类型*。

2.1 0：标准请求

每个 USB 器件必须响应各个标准请求。USB 主机将在插入 USB 器件时发出标准请求，并且在其操作器件内选择不同的配置和界面 (可选) 。插入 USB 器件时的“get acquainted”的过程被称为枚举。更多有关标准请求的信息，请参考 USB 规范中的第九章 (http://www.usb.org/developers/docs/usb_20_070113.zip) 。某个器件要想获得 USB 认证，它需要通过“第九章”中由 USB-IF 制订的兼容性测试。

主机将发出各种标准请求 (又称 Get_Descriptor 请求) ，以初始化器件并查询其功能和需求。最终，主机对器件进行配置 (Set_Configuration 请求) ，并且该器件也准备好执行操作。

2.2 01：类别请求

在进行枚举期间，器件可能会返回标识符信息，用于表示该器件属于某个标准类别。由此可以明确该器件符合 USB 的辅助规范。为了将各请求发送到符合某个 USB 类的器件内，主机需要将这些请求设置为 bmRequestType 字段中介绍的“Class” (0x01) 。这便表示器件请求中其余各字段都有特定的含义，具体情况是由器件类别规范规定的。例如，为了恢复来自 HID 类外设 (鼠标) 的某个报告 (数据结构) ，各个标准请求字段将由 HID 规范预定义为下述情况：

Table 3 HID 类外设的标准请求数据包

字节	字段	含义
0	bmRequestType	1 **01** 00001
1	bRequest	GET_REPORT = 0x01
2	wValue	报告类型和 ID
4	wIndex	接口编号
6	wLength	报告长度

一个标准请求被直接驱动到某个 USB 类别 (bmRequestType 字段中高亮显示的“01”) 后，余下各字段会由其类的规范定义。例如，对于 HID “GET_REPORT”请求，wValueH 字节被定义为一个报告类型 (01 = 输入，02 = 输出，03 = 特性) ，另外 wValueL 字节被定义为可选的报告 ID。不用考虑它们表示什么意思；但要注意，一个类别请求会预定义标准请求数据包中其余字节的含义。

2.3 02：供应商请求

通过使用供应商请求，您可以根据需要定义标准请求数据包中各个字节的含义，从而创建自己的 USB 类别。当主机发出带有一个等于 0/1**10**00000 (8 位，位 0 或位 1 为最关键位) 的标准请求时，它会明白：其余各字节的含义符合您的定义。为了使其发挥作用，主机和器件必须协同实现这些字节的含义。由于 EZ-USB™ FX2LP 器件代码和访问您自定义器件的主机程序都是自己编写的，因此您完全可以确保它们的兼容性。

下面各种情况下需要创建各供应商请求：

- 找不到任何符合您应用的预定义类型。
- 想要向已有的某个类别添加功能。

本应用笔记介绍了如何使用赛普拉斯 **CY3684 EZ-USB™ FX2LP 开发套件**和 Windows 应用来定义使用于 EZ-USB™ FX2LP 开发板的六种供应商请求。此信息有助于您创建自己的供应商控制指令。

3 供应商控制指令示例代码

为掌握本应用笔记，便于练习并修改示例，需要具备下面材料：

- **CY3684 开发套件**：该套件包含 EZ-USB™ FX2LP 评估板，而本应用笔记中将该评估版作为自定义器件使用。
- 可以从 **http://www.cypress.com/?rID=14321** 网址上免费获取 CY3684 EZ-USB™ FX2LP 文档的最新版本。解压该文件，并运行 CY3684Setup.exe 文件。选择“Typical” (典型) 安装，设置程序会在 C:\Cypress\USB 路径中创建安装目录。此目录包含 Windows 驱动程序、免费的 Keil 工具集 (功能齐全，但代码尺寸限为 4 Kbyte) 以及其他 EZ-USB™ FX2LP 的有用工具 (如 EZ-USB™ FX2LP DVK 的硬件设计文件、EZ-USB™ FX2LP DVK 使用指南和固件示例) 。对于本应用笔记，我们只需要使用 Windows 驱动程序和 Keil 工具集。由于应用笔记规定的代码容量小于 4 Kbytes，因此，您可以使用免费的 Keil 版本来学习和修改该代码 (若需要) 。
- 应用程序：下载、解压并将本应用笔记相关文件 (位于同一个应用笔记链接上) 安装在以下路径：C:\Cypress\Custom_Requests。它会安装一个子文件夹“Source”，它包含了所有源文件。

本应用笔记使用了一个 VC#程序来检测自定义器件。您可以从 **http://www.cypress.com/?rID=34870** 网址上下载赛普拉斯免费的 Visual Studio 开发工具。该安装会创建目录“C:\Cypress\Cypress Suite USB x.y.z”，其中 xyz 为版本号。该文件夹包含了一个 Visual Studio 项目 (也称“USB Control Center”)；通过该项目，您可以测试自定义的 EZ-USB™ FX2LP 器件。此外，该文件还包含了已编辑的“.exe”文件及其源代码。因此，您可以运行该文件或使用 Microsoft Visual Studio Express 2008、2010 或 2012 的免费版本来修改源代码。它的执行文件位于 CyUSB.NET\bin 子目录中

4 测试驱动程序

在深入研究代码前，您可通过本章节内容了解如何测试即将定义的自定义器件。请按照下面各个步骤进行操作，了解自定义器件的工作原理：

1. 按照 **Table 4** 准备 EZ-USB™ FX2LP 电路板的跳线器。

Table 4 EZ-USB™ FX2LP 电路板跳线器设置

跳线器	状态	用途
6、7	OUT	配置存储器，用于开发
2	IN	通过 USB 连接器供电给开发板
1、5、10	IN	3.3 V 的局部电源
3	IN	所有 4 个跳线器的状态为 IN — 激活 4 个 LED (D2-D5)
8	OUT 或 IN	未使用 (用于远程唤醒测试)

2. 在板上的左下角，将 EEPROM ENABLE 滑动开关移动到“NO EEPROM” (下面) 的位置。这样能允许 EZ-USB™ FX2LP 芯片作为一个裸器件进行枚举，并准备好将代码下载到板上 RAM 或 EEPROM 中。可将另一个滑动开关 (EEPROM SELECT) 移动到任何位置。
3. 将 EZ-USB™ FX2LP 电路板插入到 PC USB 端口。如果是第一次操作，您会看到弹出消息，提示您安装 USB 驱动程序。请导航到：

   C:\Cypress\USB

   \CY3684_EZ-USB_FX2LP_DVK\1.0

   \Drivers\cyusbfx1_fx2lp

选择您的 Windows 操作系统的相关文件夹。

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

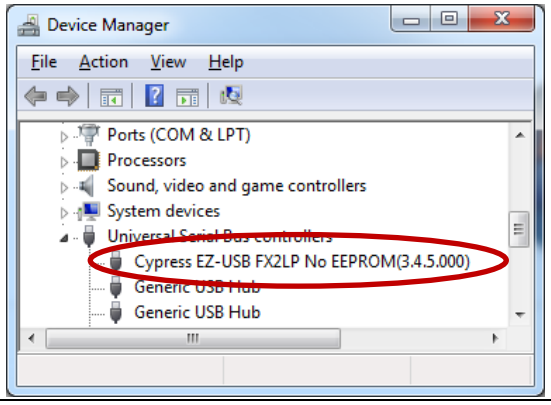


Figure 1 已安装好的 EZ-USB™ FX2LP 电路板驱动程序

4. 在下面路径中启动赛普拉斯 USB Control Panel (控制面板)：

   C:\Cypress

   \Cypress Suite USB 3.4.7

   \CyUSB.NET\bin\CyControl.exe

测试驱动程序

5. 在左侧面板上，您可看到 EZ-USB™ 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 EZ-USB™ FX2LP 电路板，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) 。左侧屏幕上将显示类似于下图的图片：

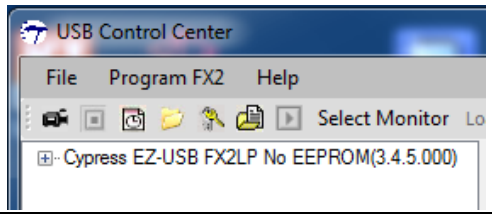


Figure 2　USB 控制中心寻找 EZ-USB™ FX2LP 电路板。

6. 现在，我们可以将 EZ-USB™ FX2LP 自定义器件下载到该电路板上。点击赛普拉斯器件条目，使其高亮显示，然后选择 Program EZ-USB™ FX2/RAM。

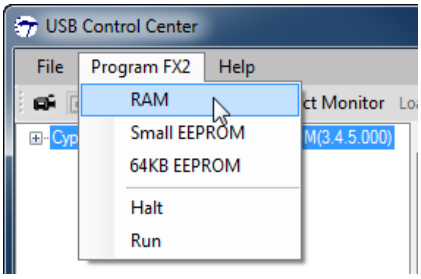


Figure 3　将新的器件代码加载到 RAM

7. 然后，导航到 **C:\Cypress\Custom_Requests\Output**，并选中 Custom_Requests.hex 文件。请注意，**C:\Cypress\Custom_Requests** 路径中存在另一个十六进制文件 (即为“mon-ext-sio1-c0.hex”) ，将在后续部分对其进行说明。然后点击 Open。
8. 查看 USB Control Center 的左侧面板。如果 PC 声音有效，您会听到“USB 连接” 的声音，随后是“USB 断开”的声音，然后将出现新的 USB 器件 (请参考 **Figure 4**) 。作为一个完整性检查，需要点亮七段式小数点，并关闭电路板顶部上的四个 LED。

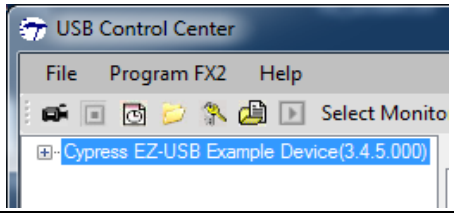


Figure 4　新的 USB 器件

这便是赛普拉斯 ReNumeration (重新枚举) 的工作情况。**Figure 1** 中所显示的初始器件是 USB 装载机，它被固定连接到 EZ-USB™ FX2LP 器件上。设置“NO EEPROM”开关可确保已经使用 EZ-USB™ FX2LP 中的加载器 ID 信息，并不是存储在外部 EEPROM 内的其他信息。在您的代码被加载到 EZ-USB™ FX2LP RAM 内后，该加载器将电气断开，然后作为新的 USB 器件 (由您的自定义代码所定义的) 进行重新连接。与开发代码相同，请重复步骤 6 和步骤 7，实现对每个新建代码进行的测试。

测试驱动程序

Note: 此外，通过将 EEPROM ENABLE 开关移动到它的“EEPROM”(上面) 位置上，然后选择步骤 6 中的“64KB EEPROM”选项，这样便可以使您的代码成为 EZ-USB™ FX2LP 电路板的固定添加件。其下载文件是：Custom_Requests.iic。编程完成后，请先断开，然后重新连接 USB 端口，您的程序将自动下载并运行。

刚刚下载好的代码将实现一个自定义的 USB 器件，从而响应 **Table 5** 中显示的自定义 (供应商) 请求。请注意，左列中的各个字段便是 **Table 3** 中的字段，但我们的代码会根据自定义 USB 器件不同的使用目的对它们进行分配。此外，也存在许多“无需关注”的字段 (表中的 x 条目)，请求时不需要这些内容。您可以在这些字段中轻松地对您器件所需参数进行定义。

此外，您还可以根据需要创建更多的供应商请求，并采用符合您应用程序的任意方式定义 **Table 5** 中所列的字段。这便是供应商请求优势。

Table 5 我们的自定义供应商请求

	上载	重新枚举	I2C 速率	7 段式	LED	EEPROM 写入
bRequest SETUPDAT [1]	**0xa2**	**0xa3**	**0xa4**	**0xa5**	**0xa6**	**0xa7**
wValueL SETUPDAT [2]	addrL	x	0:100 1:400	数字	L[3:0]	addrL
wValueH SETUPDAT [3]	addrH	x	x	x	x	addrH
wIndexL SETUPDAT [4]	0:EEPROM 1:RAM	x	x	x	x	x
wIndexH SETUPDAT [5]	x	x	x	x	x	x
wLengthL SETUPDAT [6]	LenL	x	x	x	x	LenL
wLengthH SETUPDAT [7]	LenH	x	x	x	x	LenH

该代码定义了六列自定义 (供应商) 请求。特殊请求 (上载、重新枚举，等等) 则通过 bRequest 字段定义。例如，具有 bRequest = 0xa5 的字段将一个十六进制数字发送到 EZ-USB™ FX2LP 电路板上的七段式显示屏，用于显示。按照下面步骤进行操作来使用 USB 控制中心检测该请求：

配置 USB 控制中心的“Data Transfers” (数据传输) 项，如 **Figure 5** 所示。请确保选中左侧面板中的“Control endpoint (0x00)”项，并进行下面各项设置：

- 在“Bytes to Transfer”框中选择数值“0”
- 在“Direction”框中选择“Out”
- 在“Req Type”框中选择“Vendor”
- 在“Req Code”框中选择“0xa5”

其中“Target”框的选项不太重要。保留 wValue 和 wIndex 框中的值为 0x0000，然后按下“Transfer Out”按键。数字“0”将出现于 EZ-USB™ FX2LP 电路板的七段式显示屏上。将 wValue 值改为“7”，并再次按下“Transfer Data”按键。读取值将更新为数字 7。现在，请尝试键入 wValue = 0x0F。然后尝试键入 wValue = 0x10. 输入 0x00 - 0x0F 范围外的任意值仅会点亮该小数点。

测试驱动程序

您可能会觉得奇怪：为何“Bytes To Transfer”字段被设置为“0”，但数据的一个字节却被发送到“seven-segment readout”框内。这是因为，读取操作所需的全部数据都适合七个 Standard Request (标准请求) 字节，因此，不需要使用一个分开的数据包。这样可以实现多种供应商请求，从而简化了传输过程 (和代码) 。

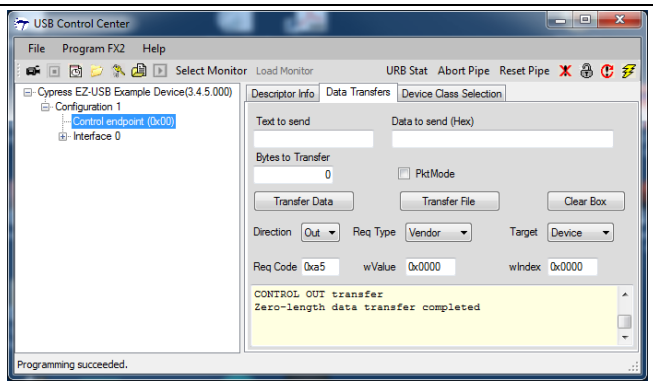


Figure 5　Data Transfers 项设置

首先，请依次完成下面各项：将“Req Code”设置为“0xa6”、将“wValue”设置为 5，并按下“Transfer Data”按键，从而进行测试 LED 请求 (**Table 5** 中的最后列中) 。您会看到 D4 和 D2 LED 被点亮。wValue 的数字是一个 LED 位，它与 bit3 = D5、 bit2 = D4、bit1 = D3 和 bit0 = D2 相映射。因此，5 = 0101 表示：点亮-关闭-点亮-关闭。

要测试“上传”请求，请保持 USB 控制中心栏位的内容，如 Figure 6 所示，然后按“传输数据”。从地址 0x0000 开始的程序 RAM 的前 64 个字节将被上载并显示。

若要从 EEPROM 上传数据，wIndex 值应保持为 0，其他栏位可以与 Figure 6 中的相同。还要确保 EEPROM 使能位于“EEPROM” (向上) 位置，EEPROM SELECT 开关处于“LARGE EEPROM” (向上) 位置。

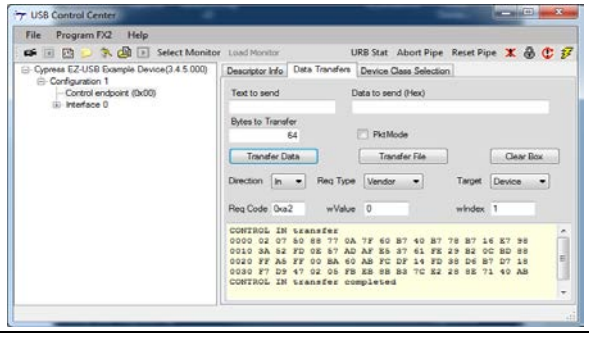


Figure 6　查看板上 RAM 中的内容

除了标准数据包外，该请求还使用了一个 IN 数据包来传输数据。这是因为，标准字段中的各项请求为“write-only” (只写) ，即：它可以将各请求发送到器件内，但不接收数据。“Bytes To Transfer”字段应包含要读取的字节数，并且“Direction”字段应更改为 IN。

最后，通过填写如 Figure 7 所示的栏位来测试“EEPROM Write”请求，并在“Text to Send”字段中提供数据后按“Transfer Data”。这将执行 EEPROM 写操作，“Bytes to transfer”中提供的计数将通过 I2C 写入 large EEPROM。

测试驱动程序

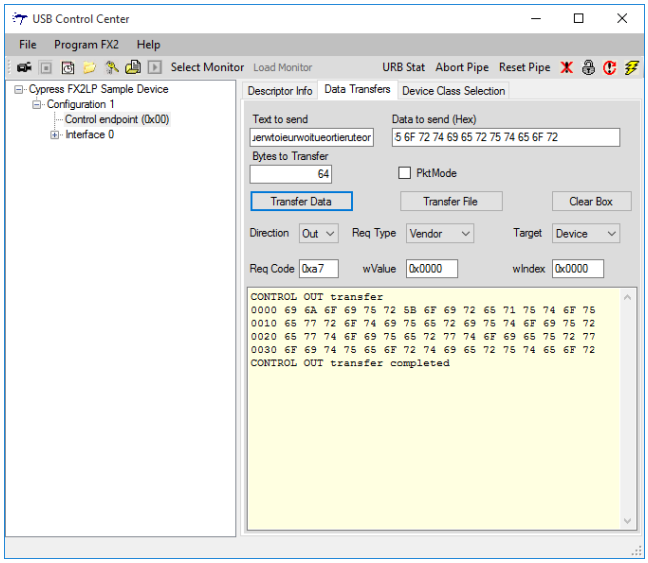


Figure 7　EEPROM 写入

可以通过执行“上传”请求来验证“EEPROM 写入”命令，该请求将读取 EEPROM 数据并在控制中心显示。要执行此读取操作，请按照 Figure 8 中的设置设置字段，然后按“Transfer data”。

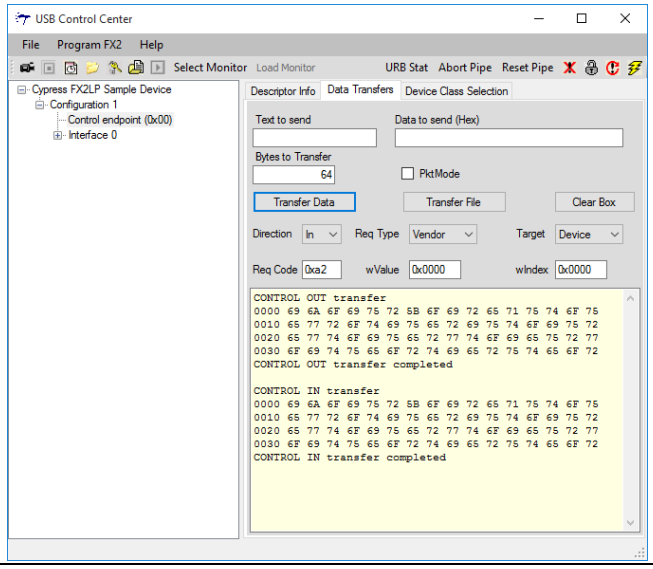


Figure 8　验证 EEPROM 写入

EZ-USB™ FX2LP DVK 中提供的 vend_ax 代码示例还演示了不同供应商命令的实现。安装 EZ-USB™ FX2LP DVK 后，可通过以下路径访问此项。<CY3684 DVK installation path>\ Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Firmware\Vend_ax

5 实现方式

通过导航到 C:\Cypress\Custom_Requests，并双击“Custom_Requests.Uv2”的 Kelis 项目文件，实现为该 EZ-USB™ FX2LP 代码启动 Keil 项目。 这样会构建 Keil 集成开发电路板 (IDE)，并将其数据显示在“Files”项中，如 **Figure 9** 所示。

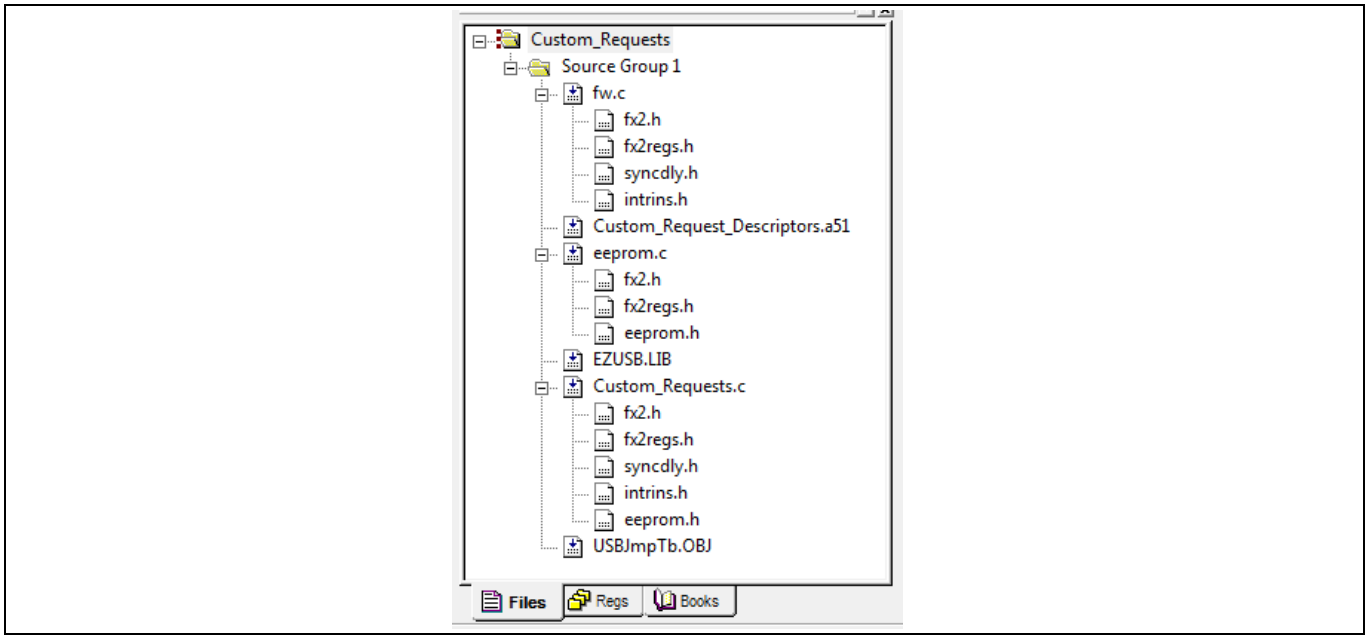


Figure 9 Keil 项目文件

5.1 fw.c

该文件 (由赛普拉斯编写) 被命名为 USB Frameworks (框架)。该文件中已包含用于处理所有 USB 标准指令的代码，因此您不需要再键入代码。此外，该文件还提供了用于处理 USB 标准请求的代码，并且还包括了我们正在关注的内容 — 供应商请求。通常不需要修改该 fw.c 框架文件。但您可以查找此文件中的固件信息，从而了解在进行枚举期间内，器件是如何响应第九章中不同的请求的。

5.2 Custom_Request_Descriptors.a51

该汇编语言文件定义了 USB 标识符，并且它包含了所有与 USB 相关的定义。在该文件内，您可以根据需要进行修改，以配置 USB 资源。典型的修改内容包括：更改 VID/PID (供应商和产品 ID)、端点配置、接口类别和信息字符串。

5.3 EEPROM.C

该文件包含了 EEPROM 的相关功能。它介绍了如何执行一个 EEPROM 读/写操作，以及如何获取页面尺寸信息。

5.4 EZUSB.LIB

EZ-USB 库是一个 8051 .LIB 文件，通过它可以实现多种固件项目的通用功能 (如 I2C 读/写子程序、延迟以及 USB 断开功能)。由于这些功能不需要修改，因此要以库的形式来提供。然而，您需要修改某个功能，或者你只是想知道某个操作的执行方式时，**EZ-USB™ FX2LP 开发套件**还包括库的源代码。

5.5 Custom_Requests.c

你可以学习此文件以了解各供应商控制指令的执行情况。您可以将它作为入门信息，以创建自己的供应商控制指令。该文件中定义了六条供应商控制指令。请注意，赛普拉斯保留了各供应商控制指令 0xa0-0xaf，用于进行代码下载和应用笔记等目的 (本应用笔记便是一个示例) 。

5.6 USBJmpTb.OBJ

这是一个目标文件，它包含了 USB 中断和 GPIF 中断的 ISR 跳转表。

6 代码分析

只需要进行两步简单的操作便可以创建一条供应商控制指令。请参考示例的 Custom_Requests.c 代码。第一步：为供应商控制指令指定符号名：

Code Listing 1

```
#define VR_UPLOAD          0xa2
#define VR_RENUM           0xa3
#define VR_I2C_RATE        0xa4
#define VR_7SEG            0xa5
#define VR_LEDS            0xa6
#define VR_EEPROM_WRITE    0xa7
```

当 EZ-USB™ FX2LP 的 USB Frameworks 代码接收到一个供应商请求时，它将调用下面的函数：

BOOL **DR_VendorCmnd**(void)

该函数的主要内容是一个 switch (切换) 语句，其中每个自定义请求对应一种 case 访问。下面是具体的 7 段式更新情况 (包含缩写注释)：

Code Listing 2

```
switch(SETUPDAT[1])
{
case VR_7SEG:
    while (I2CS & bmSTOP); // wait for STOP
    I2CS = bmSTART;        // set START bit
    I2DAT = 0x42;          // I2C address
    while (!(I2CS & bmDONE));  // wait for done
    val = SETUPDAT[2];     // wValueL=digit 0x00-0x0F
    if(val<=0x0F)
    I2DAT = ~LOOKUP7[SETUPDAT[2]];  // 0-F
    else     I2DAT = 0x7F;     // d.p. only
    while (!(I2CS & bmDONE));  // wait for done
    I2CS = bmSTOP;         // set the STOP bit
    break;
```

该切换语句对 SETUPDAT[1]进行检查，它是标准请求的 bRequest 字段 (**Table 3**) 。请注意，USB Frameworks 代码取代了对 SETUP 数据包中刚收到的 SETUPDAT[0]字段的检查工作。这正是因为：当 bmRequestType 字节 (**Table 3**) 指示了一个供应商请求时，因此，Frameworks (代码库) 将调试 DR_VendorCmnd 函数。简单地为不同 bRequest (SETUPDAT[1]) 值 (我们在 **Table 5** 中为各自定义请求所定义值) 填充代码。

代码分析

Case 语句的前四行实现以下任务： (a) 等待 EZ-USB™ FX2LP I2C 控制器处于休闲状态； (b) 设置 START 位；(c) 发送 I2C 地址 0x42，它是 EZ-USB™ FX2LP 电路板上与七段式读取连接的 I2C 扩展芯片；(d)等待地址字节传输。

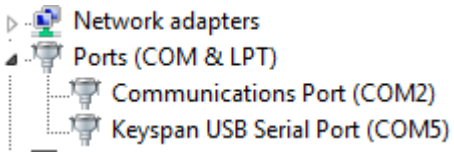
然后，代码将检查 SETUPDAT[2]的值，该值为自定义供应商重新请求的 wValueL 字节 (**Table 5**) 。我们将该字节定义为一个十六进制的数字 (0x00 ~ 0x0F) 。代码将查询这些段的值是否在范围内。若超出范围，它只会显示小数点。最后，代码将发送 I2C STOP 信号，从而完成操作。

该请求会使各字节 SETUPDAT[3-7]为“empty” (空白) ；将这些字节传输到您的自定义器件中时，您应考虑将它们用于其他目的。

7 调试代码

该示例使用了大小约为 3 kbytes 的代码存储器，它符合于 Keil 工具 (试用版) 的容量范围 (4 kbyte) 。您可以修改代码，但需要确保它不会超出容量范围。当然，所有重要的 EZ-USB™ FX2LP 项目都需要一个完整的工具集；要想获取工具集，请联系 Keil。

要想使用 EZ-USB™ FX2LP 电路板的调试功能并获得各 Keil 工具，请将 EZ-USB™ FX2LP 电路板连接到一个 PC 串行端口上。这样便需要一个 USB-串行转换器，如 Keyspan USA-19HS (亚马逊) 。按照供应商说明进行安装该转换器，然后使用 Windows 的 Device Manager (器件管理工具) 寻找它的 COM port 编号。例如：



将串行适配器插入到 SIO-1 (最上面) DB-9 连接器上。请按照下面步骤进行操作，从而使 Keil IDE 使用此端口：

1. 在 Files 选项卡中，右键点击最上面的项目—“Custom_Requests”，然后选择“Options for Target Custom_Requests”
2. 选择 Debug (调试) 选项卡
3. 点击右上角的“Settings”按键
4. 在“Port”下拉列表中选择您的 COM 端口
5. 将波特率设置为 38400
6. 请确保已经选中“Serial Interrupt”项 (如 **Figure 10** 所示)

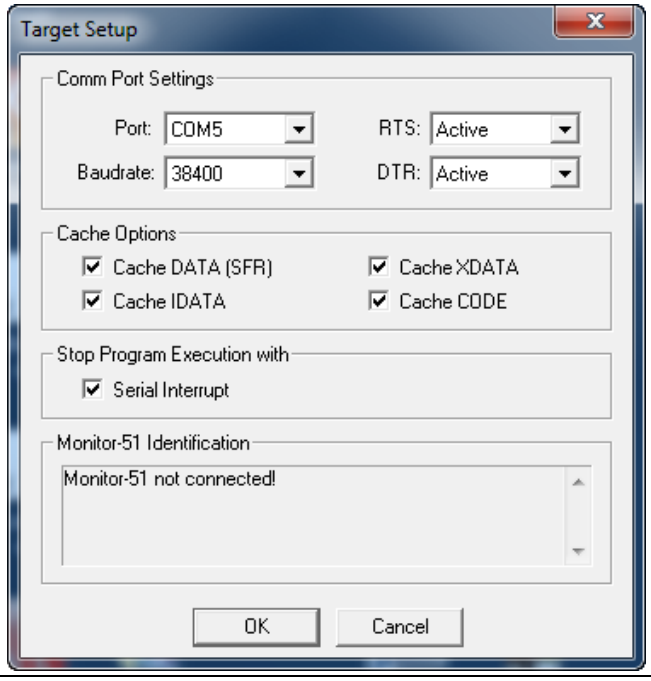


Figure 10 Comm Port Settings (共用端口设置)

最后一步为下载串行调试器，它会与您的开发代码一起运行。返回到您的 USB Control Center 窗口，然后选择“Program EZ-USB™ FX2/RAM”项，并导航到 Cypress\Custom_Request 文件夹。这时您会看到文件“mon-ext-sio1-c0.hex”；这时请点击该文件，将它下载到 EZ-USB™ FX2LP 电路板上。USB 连接器附近的一个绿色 LED 将被点亮，以指示监控器正在运行。

调试代码

*Note: 在文件名称“mon-ext-sio1-c0.hex”中：‘mon’表示监控器监示器；‘ext’表示使用外部 RAM；“sio1”表示使用 SIO-1 串行连接器 (在 USB 连接器附近) ；另外“c0”表示指定下载类型。更多相关信息，请参考 **AN42499 — Keil (TM) 调试器环境的设置、使用和故障排除**。*

选择 Debug/Start-Stop 调试会话，或点击放大镜，以启动该会话。选择 Debug/Go 项，以运行代码。代码被运行后，USB 控制中心会像前面所介绍的内容发送各供应商控制指令。

为了停止调试器，请选择 Debug/Stop Running 项；若想退出调试器，请先将其停止运行，然后选择 Debug/Start-Stop 的调试会话。

如果您想通过将各信息印刷在 HyperTerminal 或 Tera Term 上来调试自己的代码，那么请参考 **AN58009 — EZ-USB™ FX1/FX2LP 固件的串行 (UART) 端口调试**。

8 自定义 USB 控制中心

下面是 Control Center 文件夹的路径：

C:\Cypress\Cypress Suite USB 3.4.7

\CyUSB.NET\examples\Control Center

该文件夹包含多个 Visual Studio 解决方案文档，用于支持进行自定义 C#代码。例如，您可能想更改默认的面板设置 (如：修改 Vendor Req Type、Out direction 或“0”传输位)，这样在每次启动应用程序来测试自定义器件时，便不需重复这些操作。在更深的评级上，源代码还能够齐全地提供自定义 Windows 应用程序，便于与您设计中的自定义 USB 器件协调使用。有关如何设计主机应用程序的更多详细信息，请参阅应用笔记，**AN70983 - Designing a Bulk Transfer Host Application for EZ-USB™ FX2LP/FX3**。

9 最后一步

用于桥接 Windows 代码和您的 EZ-USB™ FX2LP 自定义器件的“glue”包含了一个由赛普拉斯编写的 Windows 驱动程序 (也称为“cyusb.sys”) 以及它的侣伴信息文件 (cyusbfx1_fx2lp.inf) 。赛普拉斯为终端客户 (通过提供赛普拉斯 VID 和自定义 PID 生产基于 EZ-USB™ FX2LP 的产品) 提供了这种可用的驱动器，请完成一个技术支持申请，以获取更多信息。

10 资源以及其他信息

- **AN65209 – Getting Started with EZ-USB™ FX2LP**
- 更多有关使用 KEIL 调试器环境的信息，请参考 **AN42499 — KEIL 调试器环境的使用和故障排除**应用笔记。
- 更多有关串行端口调试的信息，请参考 **AN58009 — EZ-USB™ FX1/FX2LP 固件的串行 (UART) 端口调试**。
- 更多有关 Bootloader 的信息，请参考 **AN50963 — EZ-USB™ FX1/FX2LP 启动选项**。
- 要想获取 USB 供应商控制指令设计和 HID 类别使用指南的高质量参考材料和示例，请访问 **Jan Axelson's Lakeview Research** 网址。
- **USB Complete** (作者：Jan Axelson) 该书对 USB 细节进行了清晰、完整的描述。

11 总结

本应用笔记介绍了一个被称为供应商请求的内置 USB 性能。通过这些请求，您能够创建自己的 USB 指令。本应用笔记所提供的示例固件实现了六个自定义指令，从而配合 EZ-USB™ FX2LP 开发板使用。Windows 的一个应用程序 (使用 C#语言编写)，用于实现 Windows 的通信，并允许对整个自定义器件进行测试。此外，通过源代码和详细的说明内容向您介绍了如何运行和调试 EZ-USB™ FX2LP 的代码。

文档修订记录

版本	提交日期	变更说明
**	2015-07-14	本文档版本号为 Rev**，译自英文版 001-45471 Rev*C。
*A	2017-07-04	更新徽标和版权。
*B	2018-07-20	本文档版本号为 Rev*B，译自英文版 001-45471 Rev*E。
*C	2022-05-09	更新至英飞凌模板 翻译自: 001-45471 Rev*E

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

Edition 2022-05-09
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
001-98025 Rev. *C

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。