EZ-USB™ FX1/FX2LP 启动选项

About this document

Scope and purpose

AN50963 详细介绍了赛普拉斯 EZ-USB™ FX1/FX2LP USB 2.0 外设控制器所支持的各种启动选项。

相关项目

有

关联芯片系列

CY7C68013A/14A/15A/16A、**CY7C64713**

软件版本

CyConsole 版本 1.7.0.2

相关应用笔记

AN65209

获取最新版应用笔记，或相关代码示例请访问赛普拉斯网站

更多代码示例？我们听到了

要访问各种 FX2LP 代码示例，请访问我们的 **USB 高速代码示例网站**.

你在寻找 USB 3.0 外设控制器?

要访问 USB 3.0 产品系列，请访问我们的 **USB 3.0 产品系列网站** .

Table of contents

About this document **1**
Table of contents **1**
1 简介 **3**
1.1 Bootloader 序列 3
2 I²C 启动 **4**
2.1 C0 加载：从 I²C 器件加载 USB ID 4
2.2 C2 加载：从 I²C 器件加载固件 4
2.3 EEPROM 硬件连接 6
2.4 编程 I²C EEPROM 6
3 USB 启动 **10**
3.1 主机应用 GUI 示例 10
3.1.1 实现 EZ-USB™ 固件下载工具 11
3.1.2 算法 12

Table of contents

3.2 使用脚本文件自动下载固件 13
3.2.1 创建脚本文件 13
3.2.2 创建 INF 文件 15
3.2.3 使用脚本文件下载固件 15
4 从外部并行存储器启动 16
4.1 启动要求 16
4.2 固件更改 17
4.3 硬件更改 17
4.3.1 开发板的更改 17
5 总结 18
文档修订记录 19

1 简介

赛普拉斯的 EZ-USB™ FX1/FX2LP 为 USB 2.0 外设器件提供了一个高度集成的解决方案。FX1/FX2LP 使用了一个基于工业标准 8051 微控制器的集成式 CPU。通过（基于 RAM 的）软架构，可以灵活地进行配置和升级。

FX1/FX2LP 提供了以下启动选项：

- I^2C 启动：通过使用 I^2C EEPROM，将含有供应商 ID（VID）/产品 ID（PID）或仅包含 VID/PID/器件 ID（DID）信息的完整固件下载到 FX2LP 内部 RAM 中。

*Note: 您必须使用 **USB 实施者论坛**所分配给您的 VID，只能将赛普拉斯 VID（尤其是赛普拉斯的默认 VID/PID）用于测试目的。该规则使用于您的固件 VID/PID 和 EEPROM VID/PID。*

- USB 启动：通过使用 **FX2LP 开发套件**（DVK）附带的 Control Center（控制中心）工具将固件镜像下载到 FX2LP 内部 RAM 中。枚举 FX2LP 器件后，您可以将固件从 USB 主机中加载到 FX2LP 内。FX2LP 可使用赛普拉斯的默认 VID/PID 或自定义的 VID/PID 进行枚举。如果将仅带有 VID/PID 信息的 I^2C EEPROM 连接至 FX2LP，则进行枚举时会使用该 VID/PID。此时，您可以使用主机应用将固件下载到 FX2LP 中。
- 从外部并行存储器启动：EZ-USB™ FX2LP/FX1 芯片可执行外部并行存储器中的固件（该存储器已连接至地址/数据总线）。只有通过 128 引脚的 EZ-USB™芯片才能添加片外数据和编程存储器（100 引脚和 56 引脚 EZ-USB™芯片不支持外部存储器）。

EZ-USB™ FX2LP 器件（CY7C68013A/14A/15A/16A）支持全速模式和高速模式。EZ-USB™ FX1 器件（CY7C64713）仅支持全速模式。FX1/FX2LP CPU 是一个增强型的 8051，具有快速运行和新增性能。

Note: 在本应用笔记中，除非另有说明，否则介绍的 FX2LP 特性同样适用于 FX1。

1.1 Bootloader 序列

复位后，FX2LP Bootloader 会执行下述流程，以选择启动选项。

1. 如果（连接到 I^2C 总线或并行地址/数据总线的）片外存储器未连接 EZ-USB™，那么使用由硬连线的内部逻辑提供的描述符和 VID/PID/DID 将该存储器枚举为默认的 USB 器件。
2. 如果一个包含自定义 VID/PID/DID 值的 EEPROM 与 EZ-USB™ I^2C 总线相连接，那么同样会将 EZ-USB™枚举为默认的 USB 器件，但它使用自身的内部值来替换 EEPROM 的 VID/PID/DID 值。
3. 如果包含了 EZ-USB™固件的 EEPROM 与 I^2C 总线相连，那么该固件将自动从 EEPROM 加载到 EZ-USB™的片上 RAM 内。然后 CPU 退出复位状态，以执行该引导加载代码。
4. 如果出现下面 3 种条件，EZ-USB™将开始执行片外存储器中的固件：
   - 闪存、EPROM 或其他存储器与地址/数据总线相连（仅限于 128 引脚的数据包）。
   - 不存在符合上述要求的正确格式 EEPROM。
   - EA（外部访问）引脚被置于高电平状态（表示 EZ-USB™从片外存储器的 0x0000 地址开始执行代码）。

   重枚举（ReNumeration）是一个重要的概念，它与 FX2LP 启动相关。当默认 USB 器件进行枚举，并且 USB 主机将固件和描述符表格下载到 EZ-USB™内后，它会开始执行所下载的代码。执行该代码会以电气方式模拟 USB 器件的物理断开或连接，并将 EZ-USB™作为第二个器件再次枚举。在下载代码和描述符所定义的 USB 上执行该过程。该专利的第二个枚举过程被称为‘ReNumeration’。

   EZ-USB™技术参考手册（TRM）中的第 3 章“枚举和重枚举”详细介绍了 FX2LP 的启动序列。

2 I²C 启动

使用 I²C 接口将固件或 VID/PID/DID 信息下载到 FX2LP 中。如果包含了 FX2LP 固件或 VID/PID/DID 信息的 EEPROM 连接到 I²C 总线上，会自动将 EEPROM 中的固件或 VID/PID/DID 下载到 FX2LP 内。如果 EEPROM 包含的是固件，它将被加载到片上 RAM 内。然后 CPU 会退出复位状态，以执行该引导加载代码。下面是通过 I²C 总线执行的两种加载类型。欲了解更多信息，请参考**技术参考手册（TRM）**中的 3.2 节“EZ-USB™启动模式”。

2.1 C0 加载：从 I²C 器件加载 USB ID

加电复位时，如果 FX2LP 通过读取地址 0 的 0xC0 值来检测到 EEPROM 与它的 I²C 总线相连，那么 EZ-USB™ 会自动将 EEPROM 中的 VID、PID 和 DID 复制到内部存储器内。**Table 1** 显示的是“C0 加载”的数据格式。

Table 1 “C0 加载”格式

EEPROM 地址	内容
0	0xC0
1	供应商 ID （VID） L
2	供应商 ID （VID） H
3	产品 ID （PID） L
4	产品 ID （PID） H
5	器件 ID（DID）L
6	器件 ID（DID）H
7	配置字节

第 8 个 EEPROM 字节包含的配置位控制着下列各项：

- I²C 总线速度：默认值为 100 kHz。
- 断开连接状态：该断开连接状态的默认值用于使 EZ-USB™退出复位状态后设置 EZ-USB™与 USB 的连接状态。

技术参考手册中第 3.4.2 节“存在串行 EEPROM，第一个字节为 0xC0”介绍了各个配置位的完整说明。

FX2LP 将这些 EEPROM 字节提供给主机，以响应主机的获取描述符器件请求。这 6 个字节只会代替默认 USB 器件描述符中的 VID/PID/DID 字节。这样会使主机 OS 加载与 EEPROM 中的 VID/PID/DID 值相匹配的主机驱动程序。

2.2 C2 加载：从 I²C 器件加载固件

加电复位时，如果 FX2LP 通过读取地址 0 上的 0xC2 值检测到 EEPROM 与 I²C 相连，那么它会下载 EEPROM 中的固件。**Table 2** 显示的是“C2 加载”的 EEPROM 镜像格式。与 C0 加载不同，C2 加载不需要从 USB 主机上下载固件。

Note: 技术参考手册中第 3.4.3 节“存在串行 EEPROM，第一个字节为 0xC2”介绍了各个配置位的完整说明。

赛普拉斯提供了名称为 CyConsole 和 Control Center（控制中心）的主机应用工具。在开发过程中，可以使用其中任意一个工具来编程 EEPROM。对于 Linux 开发者，可以从 **Linux 平台的 FX3 软件开发套件**中获取主机应用。

I2C 启动

FX2LP 将说明 **Table 2** 所示的格式；它对该格式进行解码，然后将其下载到相应的存储器位置中。通过 Hex2bix 工具，可以为 C2 加载生成 EEPROM 镜像。欲了解有关 Hex2bix 工具的详细信息，安装 **FX2LP DVK** 后请参考位于 *C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Utilities\Hex2Bix* 中的 *readme.txt* 文件。

Table 2 “C2 加载”格式

EEPROM 地址	内容
0	0xC2
1	供应商 ID （VID） L
2	供应商 ID （VID） H
3	产品 ID （PID） L
4	产品 ID （PID） H
5	器件 ID（DID）L
6	器件 ID（DID）H
7	配置字节
8	长度 H
9	长度 L
10	起始地址 H
11	起始地址 L
–	数据模块
–	
–	长度 H
–	长度 L
–	起始地址 H
–	起始地址 L
–	数据模块
–	
–	0x80
–	0x01
–	0XE6
–	0x00
最后一个地址	00000000

EZ-USB™ 使用 EEPROM 地址引脚 A2、A1 和 A0 来判断是连接到小型 EEPROM 还是大型 EEPROM。 单字节地址 EEPROM（小型 EEPROM）必须绑定到地址 000，而双字节地址 EEPROM（大型 EEPROM）必须绑定到地址 001。通常，双字节地址 EEPROM 用于 C2 加载。 有关 EZ-USB™ 设备如何确定它是连接到小型 EEPROM 还是大型 EEPROM 的更多详细信息，请参见技术参考手册 (TRM) 的第 13.6 节“EEPROM 引导加载程序”

Note: 虽然 FX2LP 可以从 EEPROM （64 KB）中执行 C2 加载，但硬件 Bootloader 只能将代码下载到片上 RAM （16 KB）内。使用 Bootloader 时，固件允许通过一个两阶段的引导加载将代码下载到外部 RAM 内。Hex2bix 工具具有嵌入式 Bootloader，通过使用‘-e’选项可以执行该工具。‘-e’选项用于为外部 RAM 创建一个 IIC 格式的文件。

DVK 的原理图显示了 EEPROM 与 FX2LP 间的连接示例。应用于大型和小型 EEPROM 的芯片分别为 24LC128-I/P 和 24LC00/P。

2.3 EEPROM 硬件连接

Figure 1 和 **Figure 2** 分别显示的是 C0 加载和 C2 加载的连接框图。

在 SCL 和 SDA 线路上都需要使用一个外部上拉电阻。该电阻的推荐值为 2.2 kΩ。请执行下面的流程来建立该连接：

- 确保上拉电阻连接在连接的 FX2LP 一侧。
- 通过添加一个开关或跳线器，使 SDA 引脚与 FX2LP 断开连接。这样会禁用 EEPROM 启动，从而对 EEPROM 进行系统内编程。

对于 C0 加载，请使用 EEPROM 地址引脚 A2、A1 和 A0 来表示 EEPROM 是单字节地址（000）。

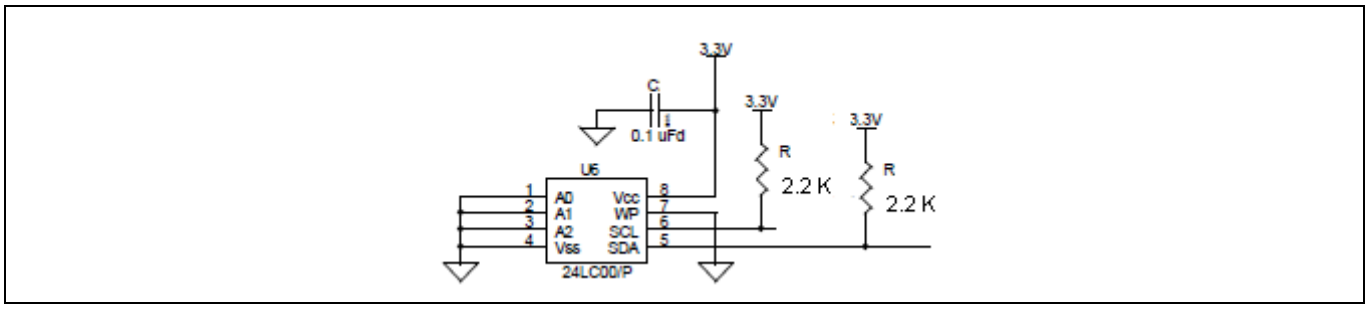

Figure 1 用于 C0 加载的 EEPROM 连接

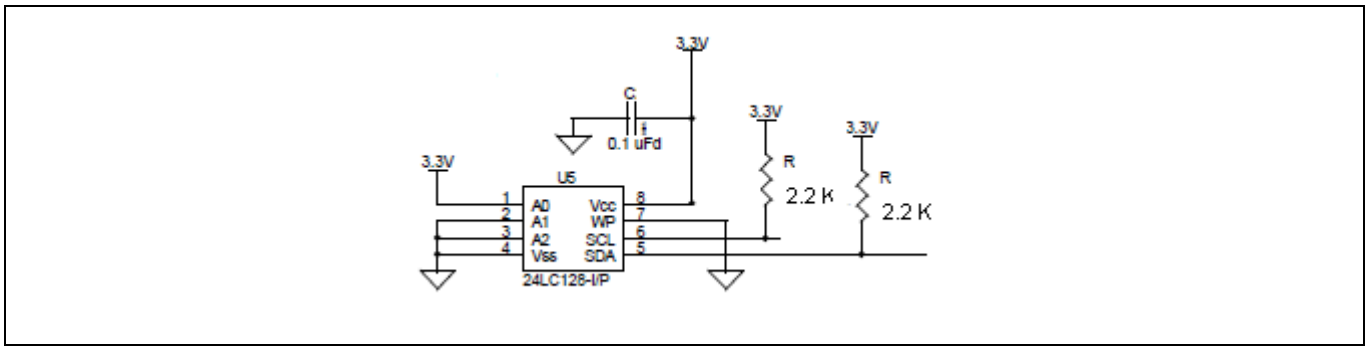

Figure 2 用于 C2 加载的 EEPROM 连接

2.4 编程 I²C EEPROM

通过下面两种方法，可以编程 EEPROM：预编程 EEPROM，并将其拖放到系统内；或者使用控制中心工具对它进行系统内编程。下面的流程介绍的是使用控制中心工具来执行 FX2LP 开发套件中的 C0 加载：

1. 为 FX2LP 开发套件板供电，不需要将其连接至 EEPROM，这样器件可以使用赛普拉斯的默认 VID/PID 进行枚举。需要将器件绑定到 *CyUSB3.sys* 上，从而能够通过控制中心来访问它。有关详细信息，请参考 **CyUSB3.sys Programmer 的参考**中介绍的“使器件与驱动程序相互匹配”一节。
2. 开发板进行枚举后，将开发套件板上的 SW2 和 SW1 开关分别设置为“EEPROM”和“小型 EEPROM”。
3. 打开控制中心工具，并导航到 *C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Firmware\Vend_ax*（根据安装的不同，该位置会存在差异）中的 DVK 固件示例 Vend_ax，然后点击“Open”。这样可将 *Vend_Ax.hex* 文件下载到 FX2LP 的内部 RAM 内。有关使用控制中心工具来编程 FX2LP 的详细信息，请参考*<install_path>\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Windows Applications\Application Source files\c_sharp*（安装 **SuiteUSB** 后）中的 *CyControlCenter.pdf* 文件。

I2C 启动

4. 请选择“Control endpoint (0x00),”，然后在 **Req Code** 字段内键入“0xA2”，在 **wValue** 和 **wIndex** 字段内键入“0x0000”，分别在 **Direction**、 **Req type** 和 **Target** 字段内选择“Out”、“Vendor”和“Device”。最后，将 VID/PID 组合键入到 **Data to send (Hex)**字段内，如 **Figure 3** 所示。**Table 3** 显示的是控制中心的“**Data to Send**”字段中所键入的字节。
5. 点击 **Transfer Data**。**Figure 3** 显示 EEPROM 已经成功上载。
6. 要想检查所编程的 VID/PID，请拔下器件，然后再插上去。此时，器件会使用所编程的 VID/PID 进行枚举。请注意，仅当 DVK 绑定到 CyUSB3.sys 驱动程序时，它才会在控制中心实用程序中重新枚举。 例如，传递“C0 B4 04 04 10”使其枚举为“Cypress FX2LP Sample Device”。

另外，还能通过使用小型 EEPROM 选项来执行 C0 加载。请依次执行下述操作：打开控制中心工具，按照上述第一步和第二步执行，点击 **Program FX2** 选项卡，然后选择“small EEPROM”。浏览到包含 VID/PID 信息的 IIC 文件，并**打开**它。

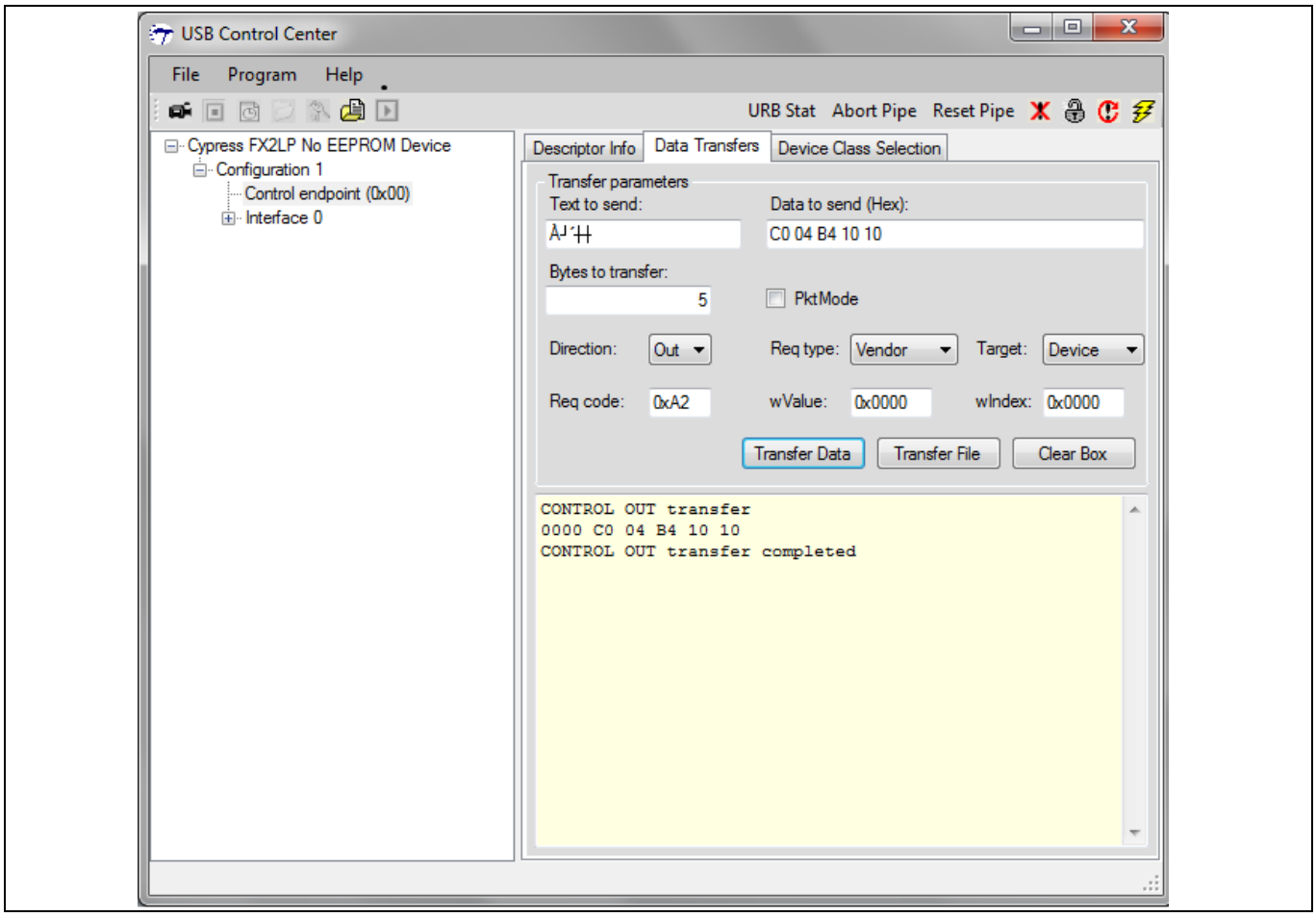

Figure 3　　EEPROM 编程

注意：传输的数据格式应遵循 **Table 3** 中的 C0 加载格式。

Table 3　　C0 加载字节

字节编号	参数	备注
字节 0	C0	表示有效的小型 EEPROM
字节 1	VID L	VID 的低位字节
字节 2	VID H	VID 的高位字节
字节 3	PID L	PID 的低位字节

I2C 启动

字节编号	参数	备注
字节 4	PID H	PID 的高位字节

下面流程介绍的是使用控制中心工具来执行 FX2LP 开发套件中的 C2 加载：

1. 在 FX2LP DVK 开发板上，选择“SW2-NO EEPROM”，并使用 USB A-B 线缆将板上的 J1 连接器连接到 Windows PC USB 主机控制器端口上。EZ-USB™器件使用默认的 VID/PID 进行枚举。
2. 下载 EEPROM 镜像文件（*.iic*）前，要将开关设置选为“SW2-EEPROM”和“SW1-LARGE EEPROM”，以选择开发板上的大型 EEPROM U5 。
3. 从 Windows 中依次选择 **Start** > **All Programs** > **Cypress** > **Cypress USBSuite** > **Control Center**，实现打开控制中心 PC 应用（请参考 **Figure 4**）。这时我们会看到 EZ-USB™ FX2LP 被罗列为“Cypress FX2LP No EEPROM Device”。
4. 在 EZ-USB™接口窗口中，请依次选择 **Program** > **FX2** > **64KB EEPROM**，然后浏览到项目文件夹，如 **Figure 4** 所示。从 *C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Firmware\Bulkloop* 或相应的安装路径中选择 *bulkloop.iic* 镜像文件。
5. 按下 RESET 按键 S1。器件将使用新的 VID/PID 进行枚举，然后提示驱动程序。要想通过控制中心工具来访问器件，从而测试批量循环操作，需要将器件同 *CyUSB3.sys* 绑定在一起。

Note: *对于 FX1/FX2，赛普拉斯将 CyUSB.sys 替换为 CyUSB3.sys。赛普拉斯已经不再支持 CyUSB.sys。*

I2C 启动

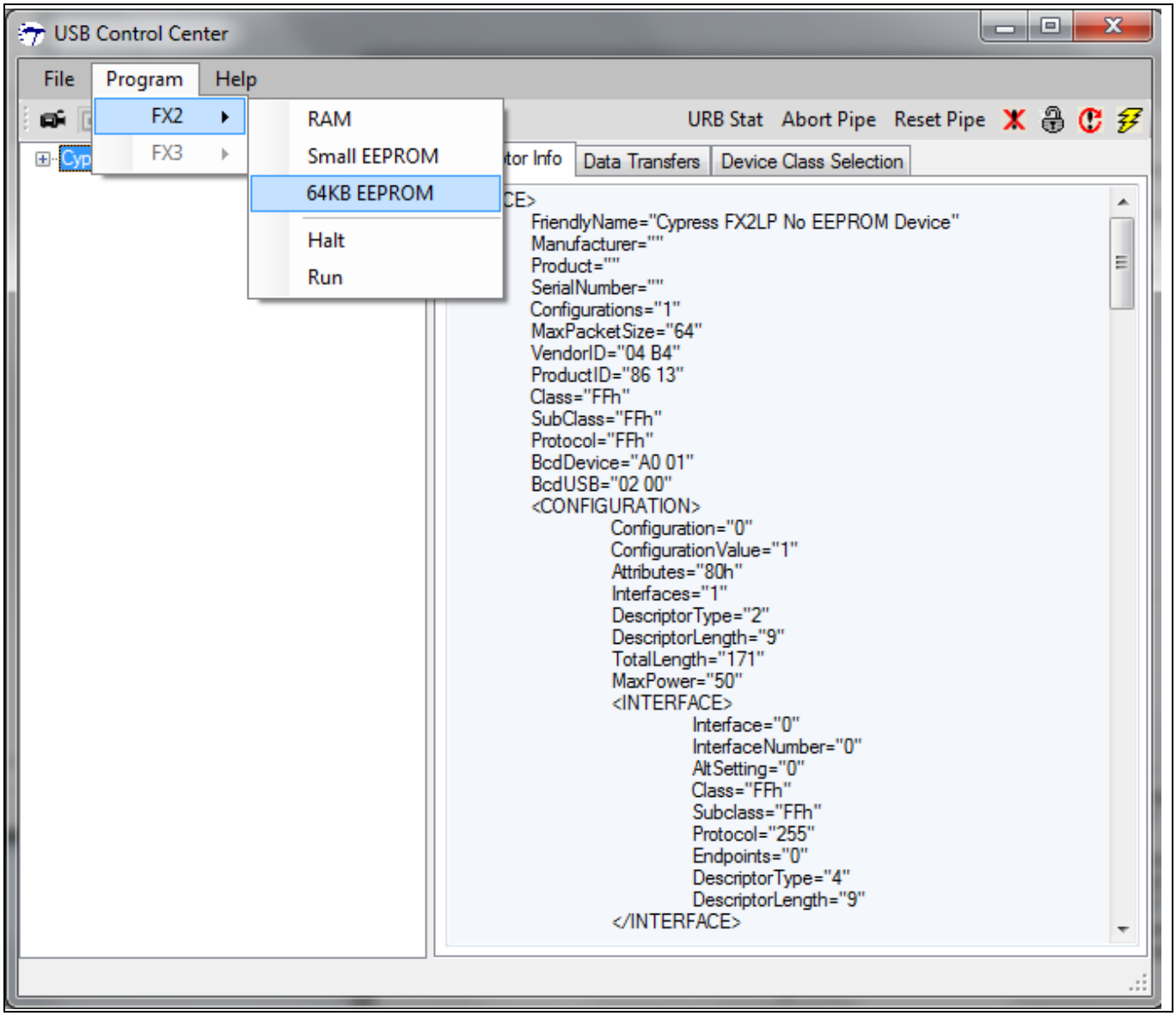

Figure 4　C2 加载

3 USB 启动

主机应用可以使用供应商指令将固件下载到 FX2LP 中。USB 端点 0 协议为供应商特定的请求提供了一个机制。对于供应商请求，将 bmRequestType 字段的位 6:5 设置为“10”。

FX2LP 为 0xA0 供应商请求提供局部支持，以便能够对 FX2LP 内部存储器进行写/读操作。主机应用可以使用供应商请求 0xA0 来将固件下载到 FX2LP 中。在 0xA0 供应商请求中，wValue 字段用于指定将被访问的存储器地址。0xA0 供应商请求的数据阶段用于将数据写入到存储器内。**Table 4** 显示的是 A0 供应商请求格式。欲了解更多信息，请参考**技术参考手册（TRM）**中第 3.8 节“用于固件加载的 EZ-USB™供应商请求”部分的内容。

Table 4 A0 供应商请求格式

Byte	Field	Value	Meaning	EZ-USB Response
0	bmRequest	**0x40**	Vendor Request, OUT	None required
1	bRequest	**0xA0**	'Firmware Load'	
2	wValueL	**AddrL**	Starting Address	
3	wValueH	**AddrH**		
4	wIndexL	0x00		
5	wIndexH	0x00		
6	wLenghtL	**LenL**	Number of Bytes	
7	wLengthH	**LenH**		

3.1 主机应用 GUI 示例

本应用笔记附带了基于 *CyAPI.lib*（VC++）的“EZ-USB™固件下载工具”主机应用，它是通过使用 Visual Studio 2008 来开发的。它介绍了如何使用供应商请求 0xA0 来下载固件。CyAPI.lib 是 **SuiteUSB 3.4 – Visual Studio** 的 USB 开发工具的一部分。 CyUSB.dll 是 CyAPI.lib 的 C# 副本，包含一个 LoadRAM() API，用于将固件下载到 FX2LP。

该主机应用有两个下拉列表；设备列表和目标设备列表。设备列表标识连接到 CyUSB3.sys 的 USB 设备并用于选择目标设备。 目标设备列表用于选择目标设备是 FX2LP 还是 FX/ANXX。点击“**Download Firmware**”按键，可将固件下载到 FX2LP 器件中，另外点击“**Clear**”按键可清除文本框。下载到器件中的值会在该主机应用的文本框中被更新。**Figure 5** 显示的是下载固件的主机应用 GUI。

Note: 该工具已经在 Windows XP、Windows Vista、Windows 7、Windows 8.1 以及 Windows 10 等操作系统中成功进行了测试。

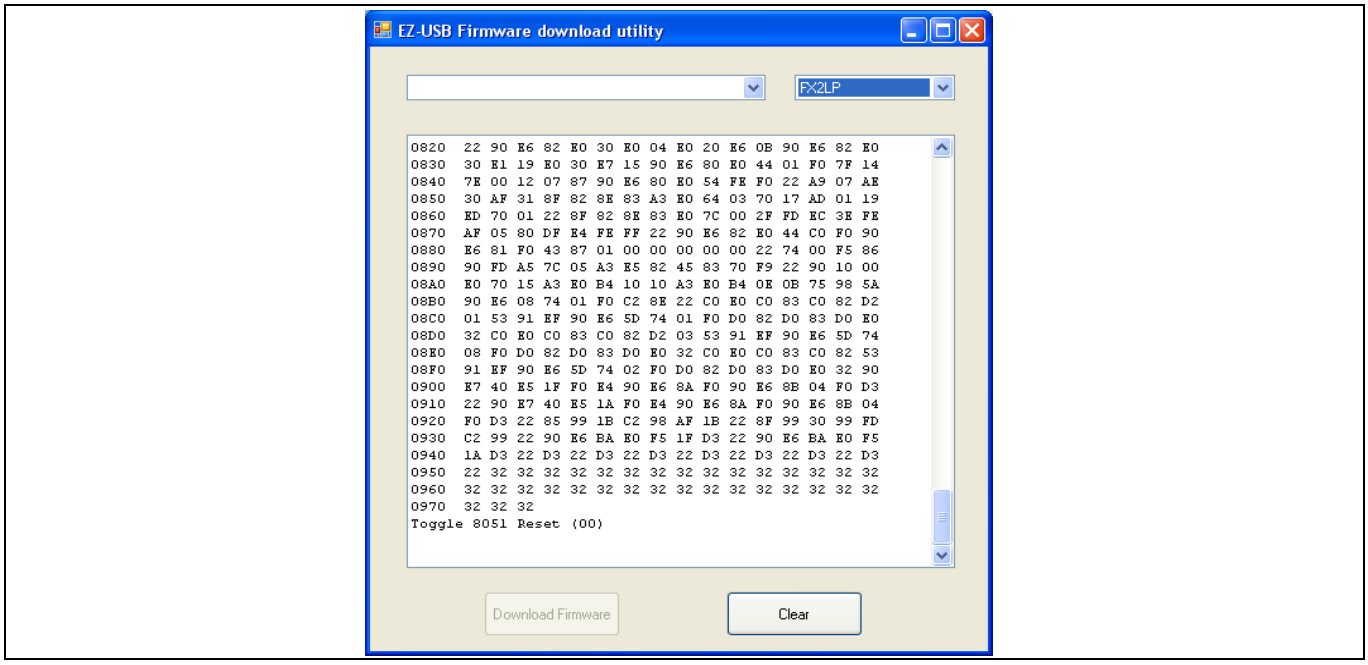

Figure 5　固件下载主机应用

Note:　该示例主机应用只能与 CyUSB3.sys 驱动程序配合使用。

3.1.1　实现 EZ-USB™ 固件下载工具

本应用笔记附带的主机应用演示了如何将固件从 USB 主机下载到 FX2LP 中。通过这种方法不再需要将固件存储到一个大型 EEPROM 内，这样您可以使用一个既小又便宜的 EEPROM。

在这种情况下，主机先保持 CPU 的复位状态，并使用 EZ-USB™默认 USB 器件下载固件。然后，主机使 CPU 退出复位状态，以进行下载代码。通过使用供应商指令 A0，可以将固件下载到 FX2LP 中。为了使 CPU 进入或退出复位状态，主机必须对 CPUCS 寄存器进行写操作。使用 FX2LP 的 CPUCS 寄存器中 8051RES 位（位 0）来复位或运行 8051。USB 主机可以通过供应商请求 0xA0 写入该位。主机写入‘1’来复位 8051；写入‘0’来运行 8051。

即使在重枚举后仍可能使用固件加载请求 0xA0，但只有将 8051 保持为复位状态时，该请求才有效。

如果应用实现供应商特定的 USB 请求，但又不想使用固件加载功能，那么请不要将 bRequest 的值 0xA0 使用于您的自定义请求。为避免未来可能发生的不兼容情况，赛普拉斯保留了 0xA0 到 0xAF 的供应商请求。

通过 Keil µVision2 可以为 FX2LP 开发固件。它输出 intel 十六进制格式的固件（**http://microsym.com/editor/assets/intelhex.pdf**）。解析该固件文件以获得下载到 FX2LP 中的固件镜像。EZ-USB™固件下载工具将解析固件文件，并将它下载到 FX2LP 中。

*Note:　因为各 CPUCS 间的地址不同（AN21xx、FX、FX1 系列的地址为 0x1F92；FX2LP 的地址为 0xE600），并且 AN21xx、FX、FX1 芯片与 FX2LP 芯片间的寄存器地址也不一样，所以 AN21xx、FX、FX1 芯片使用的 Vend_ax 固件与 FX2LP 芯片使用的固件也不相同。主机应用演示的是 AN21xx 和 FX 的固件下载，同时还介绍了易于使用的 ArrayList 格式的 Vend_ax 固件。**AN45471 — 使用 FX2LP 创建自己的 USB 供应商指令**介绍供应商指令的详细信息，同时介绍了如何测试 Vend_ax 中所包含的指令（Vend_ax 中定义某些供应商指令）。*

3.1.2 算法

供应商请求 0xA0 只能访问 FX2LP 的内部存储器。因此，当固件的某一部分被存储在外部存储器内时，便需要使用某种方法来访问外部存储器。作为 DVK 的一部分的 Vend_ax 示例可实现该操作。安装 DVK 后，可以在 *C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Firmware\Vend_ax*（根据安装的不同而位置不同）中找到它。该固件通过实现供应商请求 0xA3 来对 FX2LP 的外部存储器进行读/写操作。该供应商请求也能访问内部存储器。

Intel Hex 标准格式

Hex 文件中的每一行都有相同的基本结构，被称为‘记录’。每条‘记录’共包含以下六个部分：

- **起始代码**：为一个字符，即一个 ASCII 冒号‘:’。
- **字节计数**：为两个十六进制数字，即数据字段中的字节数量（十六进制数字对）。
- **地址**：为 4 个十六进制数字，即存储器开始部分的 16 位数据地址。
- **记录类型**：为两个十六进制数字，即 00 到 05，用于定义数据字段的类型。
- **数据**：它是 *n* 个数据字节的序列，由 2*n* 个十六进制数字表示。
- **校验和**：它是两个十六进制数字，即所有字段（字段 1 和 6（起始代码‘:’ 字节和校验和的两个十六进制数字）除外）的数值和的 2 的补码的最低有效字节。

Note: 有关 hex 文件格式的详细信息，请参考规范中介绍的内容。

Figure 6 显示的是通过主机应用将固件下载到 FX2LP 中所使用的算法。

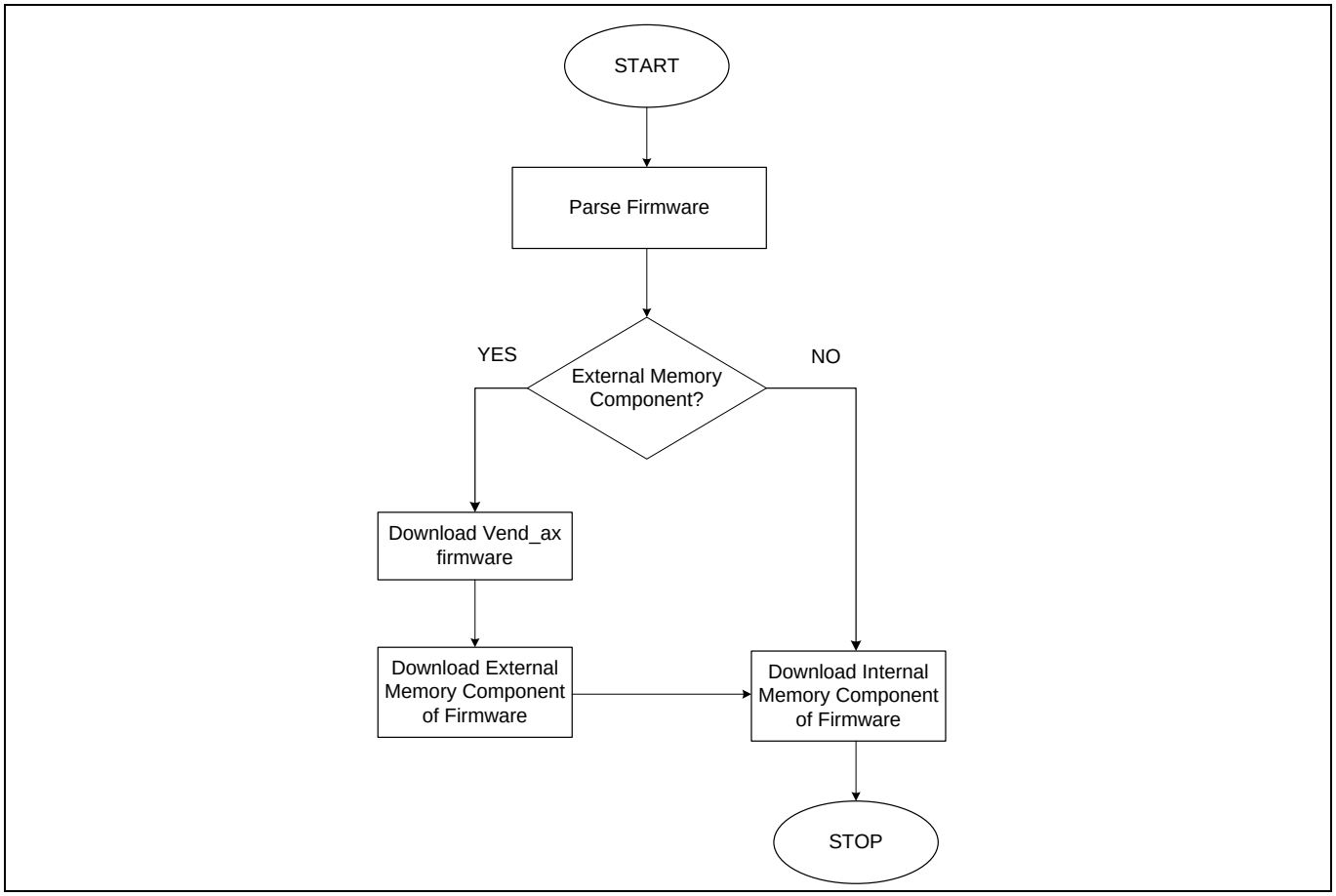

Figure 6 固件下载的算法

EZ-USB™固件下载工具中的函数

下面是主机应用中的各个函数：

- **WndProc(message %m)：**该用户定义的函数用于处理发送到某个窗口的信息。
- **CPU_Reset(bool ON)：** 该函数会根据 Boolean 输入的情况复位或运行 FX2LP CPU。如果该输入为 true（真），那么将复位 CPU；如果该输入为 false（假），则会运行 CPU。
- **GetDevice()：**该函数用于将被绑定到 *CyUSB.sys* 的器件填充到 Device_List 下拉列表内，并处理这些器件。
- **Error(String^ err)：**该函数用于打印错误信息。
- **InitVendAX(bool FX2LP)：**该函数用于初始化一个包含了 Vend_ax 固件的阵列（VendAX）。该函数的 Boolean 输入指定固件是否包含一个外部存储器组件。VendAX 阵列用于将 Vend_ax 固件下载到 FX2LP 中。
- **Firmware_Download(String^ file)：**该函数将固件文件的路径作为输入使用，以便解析固件并将其下载到 FX2LP 中。当“VendAX”作为输入端进行传输时，该函数用于将 Vend_ax 固件下载到 FX2LP 中。当固件具有一个外部存储器组件时，将使用递归方式调用该函数，以下载 Vend_ax 固件。FX2LP 内部 RAM 大小为 16 KB。上一代 FX2LP（如：AN21xx、FX 等）的 RAM 只为 8 KB。由于 0xA3 可以访问内部和外部存储器，因此主机应用会将 0x2000 作为外部存储器的起始位置，以简化算法。

3.2 使用脚本文件自动下载固件

控制中心工具可以创建脚本文件，从而能够自动将固件下载到 FX2LP/FX1 中。要使用此技术，FX2LP/FX1 通过“C0 加载”使用 VID/PID 启动，如 **C0 加载：从 I2C 器件加载 USB ID** 章节所述。INF 文件将 Bootloader VID/PID 绑定到包含记录序列的脚本文件内，从而下载固件。进行枚举前，通过使用脚本文件将固件下载到器件内。然后，器件会使用下载的固件中所包含的 VID/PID 进行枚举。该方法的优点是不需要将固件存储在大型 EEPROM 内。因此，可以使用一个又小又便宜的 EEPROM。它还可以自动从 USB 下载固件。只有赛普拉斯提供的 *CyUSB3.sys* 驱动程序才支持脚本功能。

Note: 使用于所下载的固件的 VID/PID 值应与 Bootloader VID/PID 值互异。否则，将递归下载固件。

3.2.1 创建脚本文件

拥有工作的固件后，请使用 FX2LP 开发套件执行下述操作，从而创建脚本以下载固件：

1. 断开 SDA 线路与 FX2LP 间的连接（将 SW2 连接到“No EEPROM”的位置）。
2. 使用 USB 线缆将开发板连接到电脑上。
3. 允许该开发板使用其默认设置进行枚举。在设备管理器中，FX2LP 必须枚举为“赛普拉斯 FX2LP 无 EEPROM 器件”（VID/PID 0x04B4/0x8613），并且 FX1 必须枚举为“赛普拉斯 EZ-USB™ FX1 无 EEPROM 器件”（VID/PID 0x04B4/0x6473）。在 **EZ-USB™开发套件用户指南**（该指南是开发套件文档的一部分）中第 7.2.4 节“绑定赛普拉斯 USB 驱动程序，用于下载的固件镜像”中介绍的是将驱动程序绑定到器件的流程。
4. 打开 Control Center（控制中心）窗口。
5. 在 Control Center 窗口中左上角点击“Record”（记录）按键，如 **Figure 7** 所示。

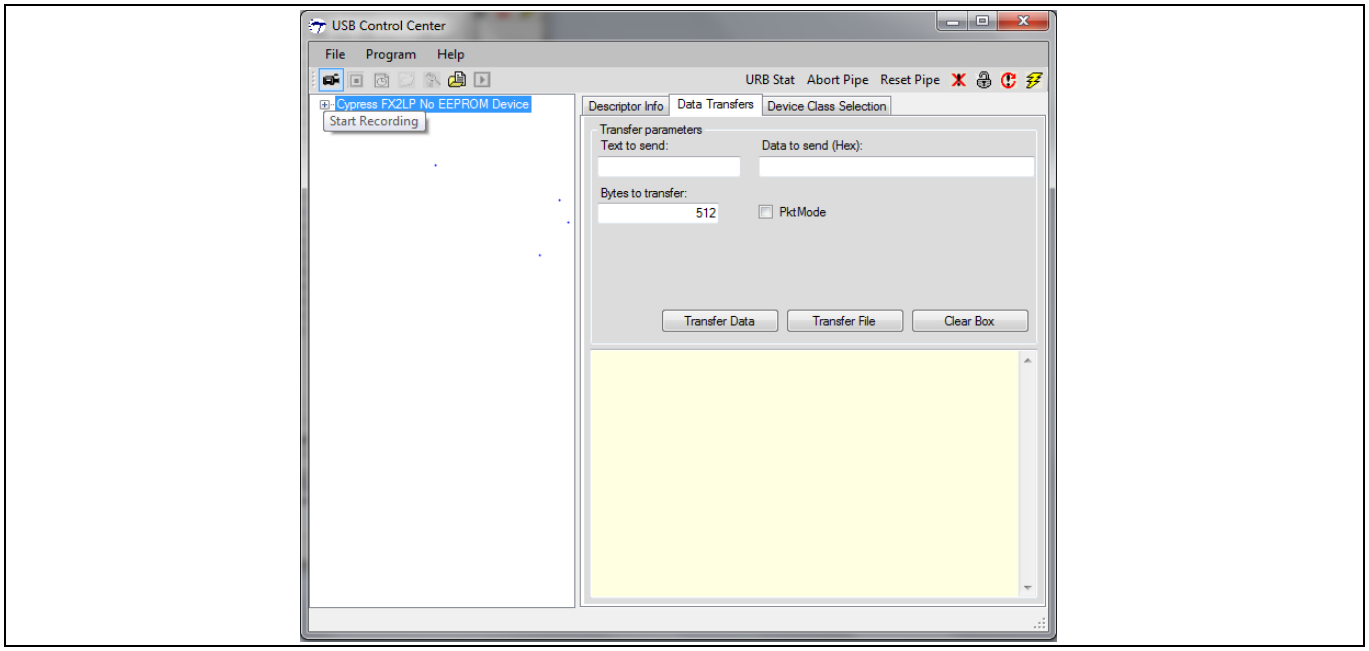

Figure 7　开始记录脚本

6. 依次选择 **Program** > **FX2** > **RAM**，然后为所生成的脚本选择固件（hex 文件）。
7. 当成功下载程序到 RAM 内时，请点击“Start recording”按键旁边的“Stop recording”（停止记录）按键，如
   Figure 8 所示。将脚本保存为扩展名为*.spt* 的文件格式。
8. 给您的文件命名为“*MyDevice.spt*”，然后保存它。现在已经创建好了您的脚本。

Note:　有些情况下，不能创建脚本，是因为控制中心生成了错误。如果发生这种情况，请关闭控制中心工具，然后在特权用户配置模式中再打开它，并重新执行上述各步骤，以创建脚本文件。

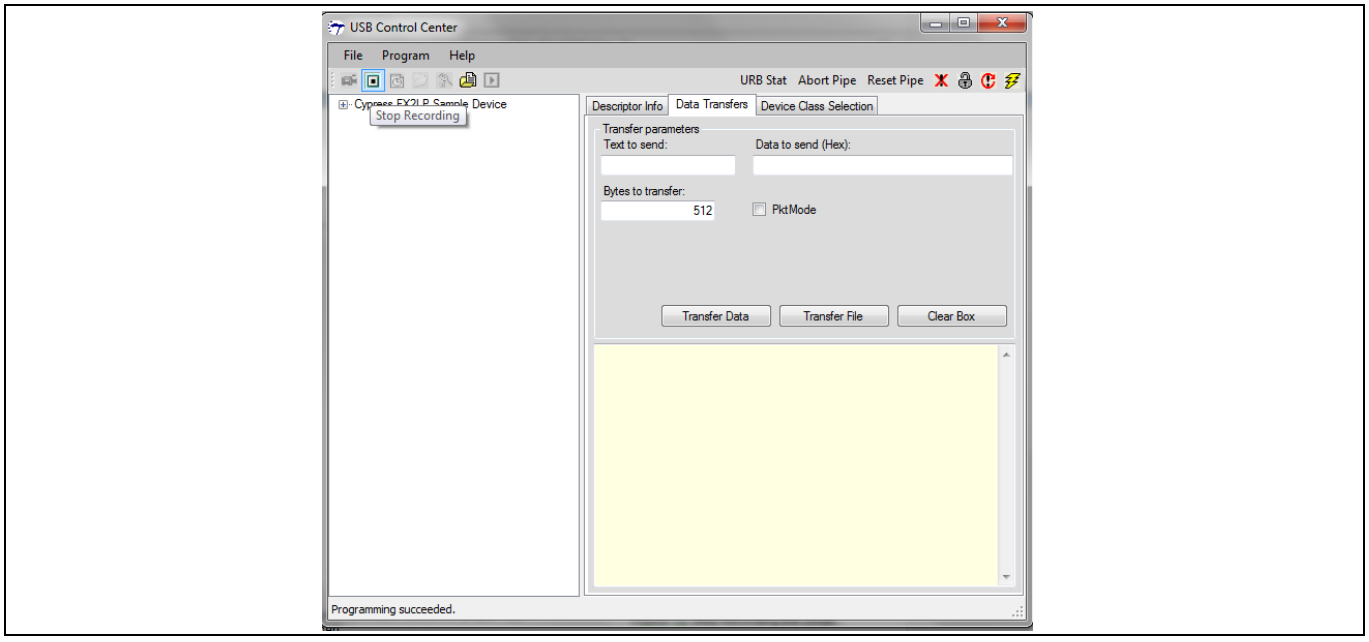

Figure 8　停止记录脚本

3.2.2 创建 INF 文件

要使用前面所创建的脚本下载固件，您必须创建一个 INF 文件。该文件可识别编程到器件小型 EEPROM 中的 VID/PID。然后，它会使用脚本文件将固件下载到器件中。用于参考的一个 INF 示例文件作为应用笔记项目附带的一部分。有关创建 INF 文件来下载基于脚本的固件的步骤，请参考 **CyUSB3.sys Programmer 的参考**中第二部分的内容。

1. 先从本应用笔记附带的 INF 示例文件 *cyusb3.inf* 开始。设置该文件的目的是为了方便运行脚本文件。
2. 使用文本编辑器（如记事本）打开 *cyusb3.inf*。
3. 在 *cyusb3.inf* 中，将 04B4/8613 VID/PID（是 FX2LP 的标准 Bootloader VID/PID）替换为下载到小型 EEPROM 的 VID/PID。
4. 将 04B4/1004 VID/PID（是批量循环固件 VID/PID）替换为您固件的 VID/PID。
5. 在 INF 文件的底部（“字符串”部分）中，请确保所显示的 VID/PID 与用于脚本下载前的 VID/PID 并与下载固件后器件使用来进行枚举的 VID/PID 相匹配。在引号内添加一些有意义的文本内容。
6. 保存新的 INF 文件。本应用笔记附带的 INF 文件包含一个数字签名。请注意，对该 INF 文件进行任何更改都会使目录文件中的数字签名失效。您需要获得被认证的 INF 文件。或者为了使用它，取消签名，这样需要禁用 Windows 8、Windows 8.1 和 Windows 10 中的驱动程序签名强制签名功能。Windows 7、Windows Vista 和 Windows XP 将给您提供一个警告，但允许将器件绑定到驱动程序内，而无需禁用驱动程序的签名功能。

3.2.3 使用脚本文件下载固件

本应用笔记附带的文件使用脚本将批量环路固件（带有 VID 04B4 和 PID 1004）下载到 FX2LP。 通过 C0 加载下载到小型 EEPROM 的 VID/PID 值为 04B4/8613。 因此，包含的 INF 文件将指定这些 ID。 这些文件与本应用笔记一起位于基于操作系统和 CPU 架构的不同文件夹中。 您可以尝试使用这些 VID/PID 值，在这种情况下，您可以直接使用附加文件而无需任何更改。 如果您想尝试不同的 VID/PID 值，请更改以下内容：用于下载到小型 EEPROM 的 IIC 文件、要下载到 RAM 的固件、脚本文件和 INF 文件。这些文件使用的默认 VID/PID 为 0x04B4/0x8613。固件的 VID/PID 为 0x04B4/0x1004。

1. 拥有脚本、INF 文件和开发板后，请使用脚本将固件下载到 FX2LP 中。
2. 请确保脚本文件、驱动文件和 INF 文件都位于同一个文件夹中。
3. 将开发板连接到主机，然后将它同应用笔记附带且位于“Drivers”文件夹中的驱动程序绑定起来。
4. Windows XP、Windows Vista、Windows 7、Windows 8 和 Windows 8.1（32 位和 64 位）使用了不同的 INF 文件；本应用笔记附带了这些文件，并它们位于“Drivers”文件夹内。选择与您的操作系统相对应的 INF 文件。

4 从外部并行存储器启动

您可以将外部并行存储器连接到 EZ-USB™地址/数据总线上。固件不会被加载到 EZ-USB™的内部 RAM 内；却从外部并行存储器直接运行它。该存储器可以是闪存、EEPROM 或任意其他并行存储器。

Note: 只有 128 引脚封装才拥有外部地址/数据总线。因此，本应用仅限于下面各个芯片：CY7C68013A-128AXC、CY7C68014A-128AXC、CY7C64713-128AXC 和 CY7C64714-128AXC。

为了满足该模式要求，需要更改固件和开发板。

4.1 启动要求

1. 该模式的代码存储器必须位于外部并行存储器中。如果加电复位时 EA 输入引脚被设置为逻辑高电平，将更改 EZ-USB™ FX2LP 的存储器映射情况，以便整个 64 KB 的外部存储器能映射为代码空间，并且整个内部存储器（主 RAM）可作为数据空间使用。

Note: 与 EZ-USB™的其他 Von Neumann 架构配置不同，将 EA 设置为 1 会使 EZ-USB™处于 Harvard 架构配置。

2. 首先，EZ-USB™尝试通过 I^2C 总线从串行 EEPROM 启动。它会检查第一个字节是否为 0xC0 或 0xC2，以决定所使用的启动方法。为了避免通过串行 EEPROM 启动加载，不能在 I^2C 总线上存在任何串行 EEPROM。如果已经连接某个外部串行 EEPROM，那么第一个字节的数值不能是 0xC0 或 0xC2。
3. 满足这两个要求便能确保 EZ-USB™能从连接到地址/数据总线的外部存储器中查找固件，并且如果 FX2LP DVK 的跳线器被设置为 MM0 = IN 和 MM1 = OUT，则会在加电复位时开始执行地址 0x0000 的代码。有关在 FX2LP DVK 上映射存储器的详细信息，请参考 **EZ-USB™开发套件用户指南**中第 4.8 节“存储器映射情况”的内容。
4. 用于进行测试的 WinBond W27C512-45Z 并行 EEPROM 具有访问时间为 45 ns。更多有关 W27C512-45Z 的信息，请参见 **W27C512 数据手册**。可以使用 Atmel AT27C512R 代替它。有关更多信息，请参考 **AT27C512R 数据手册**。**Figure 9** 显示的是外部并行存储器的连接详情。请确保该芯片具有足够的电源，某些芯片需要的电源比 DVK 支持的电源高。[1]

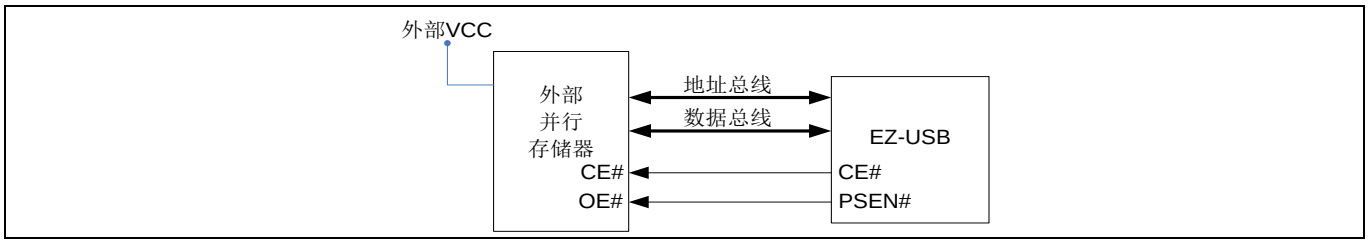

Figure 9 外部并行存储器的连接详情

外部并行存储器需要具有下列特性：

- 访问速度要足够快，使 EZ-USB™能够读取代码。请参考 **FX2LP** 数据手册，了解有关访问时间 tACC1 的详细信息。
- 要将固件的原始二进制镜像从地址位置 0x0000 加载到外部并行存储器中，却不需要使用任意起始字节（与 IIC 文件镜像不同的是 IIC 文件镜像会下载到串行 EEPROM 中）。Keil 软件所生成的 hex 文件

1 赛普拉斯未对 Atmel AT27C512R 进行测试。根据数据手册的审查，AT27C512R 是一个可替代的芯片。它是一个 OTP（一次性可编程）芯片。可以重新对该芯片进行编程；重新编程后，它具有较慢的访问速度，并能够同运行频率为 24 MHz 的 FX2LP 一起使用。

的格式为 Intel hex。不能直接下载该文件。如果 Programmer 工具没有自动将该文件转换为原始二进制文件格式，那么下载该文件到外部并行存储器前，需要执行转换操作。已经成功对 ALL-100 Programmer 进行过测试，从而编程 FX2LP DVK 上的 WinBond W27C512-45Z 并行 EEPROM。它的软件封装包带有一个工具，用于将 Intel hex 文件转换成原始的 hex 文件。Hex2bix 工具生成的 IIC 文件将起始字节插入到固件镜像内；不应使用它来进行外部并行存储器中的下载操作。

5. 设置数据的指针只能访问 0x0000 到 0x3FFF 和 0xE000 到 0xE1FF 的内部存储器位置。如果固件使用 SUDPTRH:L 来指向到用于枚举的描述符，那么必须将这些描述符存储在内部数据空间内。使用启动方法时，描述符和固件均被存储在 EZ-USB™外部的代码存储器中。因此，启动时应该无条件将这些描述符从外部存储器复制到内部存储器内。

4.2 固件更改

根据 EZ-USB™框架，您可以更改任意固件，使之成为 **CY3684 开发套件**内容的一部分，以便运行外部并行存储器中的固件。如**启动要求**一节中所述，请确保无条件将描述符复制到 EZ-USB™内部存储器（主 RAM）中。EZ-USB™框架中的更改如下。

1. 在 *fw.c* 中，main 函数的 TD_Init()后面有一个描述符设置部分。在本部分，移除‘if’条件“if ((WORD)&DeviceDscr & 0xC000)”，并且无条件地执行该‘if’条件中的所有代码（大部分代码都用于复制描述符）。这样可确保标准 USB 描述符（器件/配置描述符）的值总是被复制到内部存储器（主 RAM）内。
2. 对用于将这些标准 USB 描述符从外部存储器复制到内部存储器内的行进行更改，以便给编译器通知存储在外部存储器中的描述符实际上是位于代码存储器中：将“*((BYTE xdata *)IntDescrAddr+i) = *((BYTE xdata *)ExtDescrAddr+i);”更改为“*((BYTE xdata *)IntDescrAddr+i) = *((BYTE code *)ExtDescrAddr+i);”。
3.
4. 在将描述符从外部存储器复制到内部存储器的步骤中，请为定义类别特定描述符的应用添加类别特定的描述符。对于 HID KB 示例，目前在 main 函数中更新了三个描述符指针（即 pHIDDscr、pReportDscr 和 pReportDscrEnd）。
   即在 main 函数中更新了这些指针的偏移量。它们在 DR_GetDescriptor 函数（*periph.c* 中的函数）中的重新评估功能将被清除。

4.3 硬件更改

对硬件进行最重要的更改为：将 EA 引脚置为逻辑 1。但是当使用 DVK 开发板时，必须更改某些硬件连接，以便能从外部并行存储器运行 EZ-USB™。**CY3686 DVK** 和 **CY4611B DVK** 都要进行这些更改。

4.3.1 开发板的更改

1. DVK 开发板拥有一个连接到地址/数据总线的 SRAM（在 CY3686 上，该 SRAM 被标记为 U3，而在 CY4611B 上该 SRAM 被标记为 U4）。为了避免总线冲突，请取出或禁用该芯片。通过将 SRAM 芯片使能引脚（引脚 5 — RAMCE#）设置为逻辑 1，可以禁用该芯片。
2. 通过 GAL22LV10C 芯片（U2）路由 SRAM 的控制信号，在将外部并行存储器访问的信号发送给 SRAM 前，GAL22LV10C 芯片将转换这些信号。从套接字移除 AL22LV10C 或重新编程它，以阻止这些信号通过该芯片。
3. 请确保 I^2C 总线上的串行 EEPROM 已被断开或被禁用。要想禁用该 EEPROM，要将 CY3686 上的 SW2 设置为‘No EEPROM’，并移除 CY4611B 上的跳线器 J22。

5 总结

本文档介绍了 FX1/FX2LP 中的启动选项、它们的意义、优点和缺点。通过本文档，可以为您的终端应用选择最佳的启动选项。

文档修订记录

Document version	Date of release	Description of changes
**	2012-12-11	本文档版本号为 Rev**，译自英文版 001-50963 Rev*E。
*A	2015-08-12	本文档版本号为 Rev*A，译自英文版 001-50963 Rev*J。
*B	2016-12-09	已更新为新模板。 完成日落评论。
*C	2017-06-30	更新徽标和版权。
*D	2020-01-06	更新为新模板。 正在完成日落复审。
*E	2022-02-24	译自英文版 001-50963 Rev. *L 更新至英飞凌模板

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

Edition 2022-02-24
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.cypress.com/support

Document reference
001-85240 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。