

AN10986

LPC1300 通过 USB 在线系统编程

Rev.1-24 2010 9 月

使用手册

文件信息

信息	内容
关键字	LPC1300, USB, 在线系统编程 (ISP) , 大容量存储类型 (MSC), Cortex-M3, Windows , Apple OS-X , Linux
摘要	这份使用手册介绍了如何在 Windows, Apple OS-X 以及 Linux 等操作系统上使用微控制器 LPC1300 (基于 Cortex-M3 内核, 具有片上 USB 在线编程功能), 在片上 flash 存储器上更新固件。



1、简介

基于 ARM Cortex-M3 CPU 架构的 LPC1300 系列微控制器，在嵌入式应用中具有高级别块集成支持和低功耗的特点。LPC1300 辅助外设，包括高达 32KB 的 flash 存储器，8KB 的数据存储器，USB 接口，1 个 UART，1 个 SSP 控制器，SPI 接口，I2C 接口，8 通道 10 位 ADC，4 个通用定时器/PWMs，以及高达 40 个通用 I/O 引脚。

另外，可在线系统编程的片上 ROM（a bootloader）支持 UART 和 USB flash 编程，类似于 APIs 的用户代码编程方式。flash API 通过一个简单的接口实现板上 flash 编程功能，并可随时进入 ISP 模式。USB API 能够兼容人机接口设备（HID）和大容量存储设备（MSC）的新产品，既不需要用户编写驱动程序，也不需要存储在 flash 上。

包含在该使用手册上的条目如下

- 1、USB 在线系统编程概述
- 2、USB ISP 详述
- 3、自动化的 USB ISP
- 4、USB ISP 的自动化入口
- 5、软件实例
- 6、结论

2、USB In-System Programming（ISP） 概述

LPC1300 片上的 USB ISP 固件，终端用户可以通过标准个人电脑操作系统对其进行编程和更新。本文档引用了一部分 LPC1343 的内容，其他同样带有片上 USB 的 LPC1300 系列产品也作了类似处理。

在上电期间保持 PIO0_1 低电平将引发片上 ISP 固件进入 ISP 模式（除非被 NO_ISP 代码读保护 [CRP] 模式所禁止）。一旦进入 ISP 模式，USB VBUS 上的 PIO0_3 口将会进行电平检测。如果是高电平，将会进入 USB ISP；若是低电平，则进入 UART ISP。在用户指南手册标题为“Boot Process Flowchart”的章节中，有图表详细的介绍了这一过程。

进入 USB ISP 模式后，LPC1300 可以把片上全速 USB 接口作为大容量存储设备使用。这个磁盘设备含有 FAT12 文件系统，是大多数操作系统的标准磁盘设备。盘符将指出 CRP 状态而且盘中将包含单个文件 firmware.bin。在代码保护的设置允许的情况下，删除和重写这个文件将写到 flash 存储器。读 flash 存储器上的内容，就像复制 firmware.bin 文件一样简单。

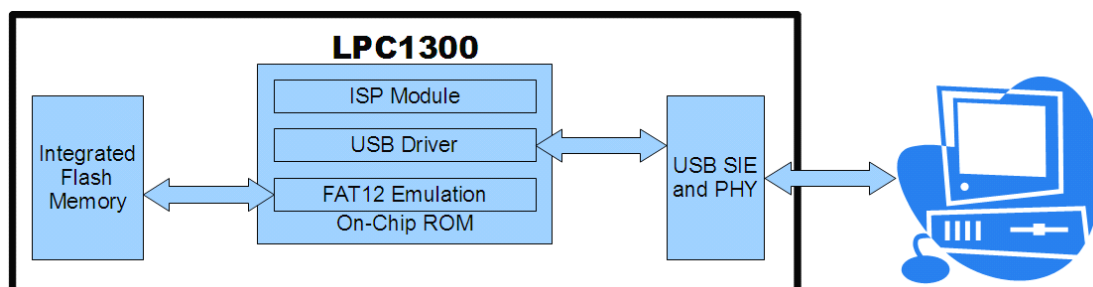


图 1 LPC 1300 USB ISP 系统

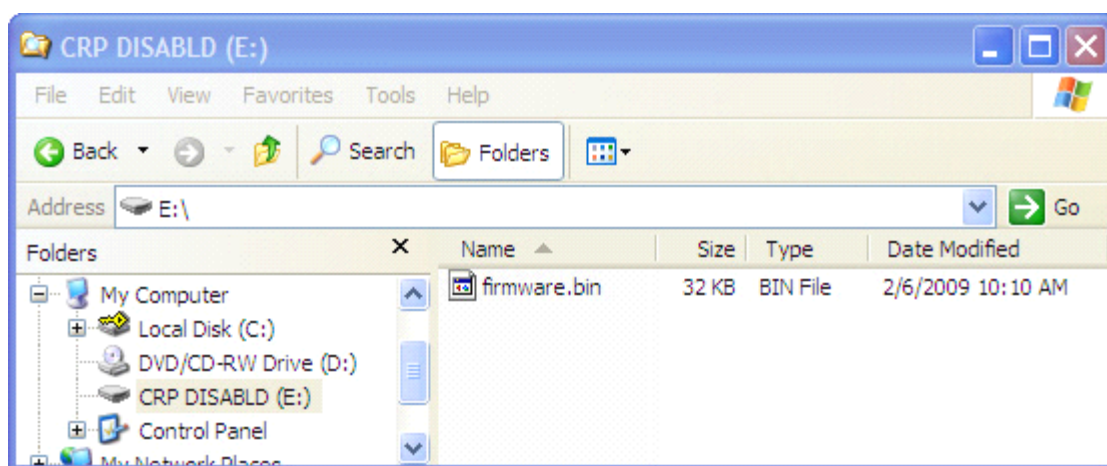


图 2 Windows XP 上的 USB ISP firmware.bin 文件

3. USB ISP 详述

LPC1300 片上 USB ISP 固件，虚拟 FAT 文件系统，这样就可以通过 PC 软件读写固件。我们详细描述了在几种标准的主流操作系统上更新固件的模拟过程和技巧。

LPC1300连接USB主机，并初始化USB ISP模式之后，枚举了USB供应商ID 0x04CC

和设备ID 0x0003。这些信息通常对终端用户屏蔽，但通过固件更新软件，这些信息可以用来寻找已经连接到计算机上的，含有LPC1300的USB设备。一旦找到这些设备，将会发送存储设备查询指令，之后会返回一串描述LPC微控制器产品的附件。LPC134X产品的查询字符串是“NXP LPC134X IFLASH 1.0”。通过设备产商的名称“NXP”以及设备型号“LPC134X IFLASH”也可以识别这些设备。这样，设备型号ID的全称为“NXP_LPC13XX_IFLASH”。

LPC1300模拟的磁盘有不用的卷标号，卷标号取决于代码读保护（CRP）的设置和重新编程过程中的细微改变。这些内容在下面会给出简要介绍，完整的文档请参考LPC1300用户手册中的UM10375部分。

如果 CRP1 或者 CRP2 可用，当文件被删除或复制新文件而导致原文件改变，用户 flash 将被擦除。

如果 CRP1 可用，或者两个 CRP 都不用，用户 flash 将被擦除或重新编程（当复制新文件时）。尽管这样，也只有被新文件覆盖的区域才会被擦除或重新编程。因此，理想情况下，新的编程文件将包含整个 flash 容量（LPC1300 为 32KB），这样，整个 flash 容量将处于已知状态。使用填补（32KB 或 flash 大小）编程文件，也可以通过编程工具，对已连接的 LPC 部件进行错误检查（比较新的固件文件与设备上 firmware.bin 文件的大小），而不需要额外的配置信息。

注意：只有Windows命令支持LPC1300 flash 映像文件夹的复制和删除。在覆盖的过程中虚拟盘的空间不足以存储Windows创建的临时文件时，覆盖操作使用Windows的资源管理器将不成功。

Table 286. CRP levels for USB boot images

CRP status	Volume label	Description
No CRP	CRP DISABLD	The user flash can be read or written.
CRP1	CRP1 ENABLD	The user flash content cannot be read but can be updated. The flash memory sectors are updated depending on the new firmware image.
CRP2	CRP2 ENABLD	The user flash content cannot be read but can be updated. The entire user flash memory is erased before writing the new firmware image.
CRP3	CRP3 ENABLD	The user flash content cannot be read or updated. The bootloader always executes the user application if valid.

图 3 LPC1300 用户手册-CRP 磁盘卷标号

引导盘虚拟的 FAT 文件系统，组成了一个单一的文件 firmware.bin，文件中包含了整个 flash 容量的信息。通常，一个盘的容量必须稍大于所需的存储容量，取决于目录和顶层的文件系统分配表。在 LPC1300 虚拟的 FAT 文件系统下，需要 4 个额外的块提供给引导块、根目录、

文件分配表。因为 MCU 需要使用 ISP 来编程整个 flash 存储器，这些额外的块使用 RAM 和 ROM 上的数据模拟，而不是映射到 flash 上作为代码存储。因此，掉电时文件系统的元数据将丢失，只有通过文件写的方式编入 flash 的数据才会被保存。

在 flash 上，写进文件系统的数据按照磁盘块的顺序存储，以块 4 作为起点。如果 `firmware.bin` 文件被删除，PC 在运行 Windows 时将以块 4 作为起点分配任一新的文件，并随着写入数据的增多而使用更多的块。这意味着，在 Windows 中，可以使用任一标准的程序或工具，把固件写进 LPC1300。在 Windows 浏览器窗口中，用户可以删除 `firmware.bin` 文件，并拖动一个新文件来编程 flash。不幸的是，Mac 和 Linux 机器上的 FAT 文件系统以不同的规则分配块，数据写进 ISP 磁盘，固件写进 flash，重新编排。这样，将导致固件更新不成功。为此，有两种变通方法。最常用的方法是适当的覆盖 `firmware.bin` 文件，另一个更暴力的方法是获得管理权限直接进行磁盘设备写/dev 操作。

4、自动化 USB ISP

有时候，要求系统执行固件的更新不需要用户介入。要求用户确定哪个盘驱动器连接到 USB 设备，或要求他们手动删除和重写一个文件，常常是很复杂的事情。在电脑程序的控制下，固件经常需要被自动更新。此部分将描述这是如何实现。为了在电脑上全自动化运行，程序需要找到正确的 USB 设备进行更新，改变 USB 设备“handle”到一个文件路径，检查 CRP 模式，存盘的文件烧写到 flash，在确保 flash 内容保存的情况下最终卸载盘设备器。这些步骤将被单独的讨论。这能有效的验证被烧写的固件文件。通过 LPC134X 用户手册里的“Criterion for Valid User Code”，可以对其进行验证，这已用 Windows ISP 工具执行过。它也合理的验证在被烧写的设备空间里是相同长度。这是在 Mac 和 Linux 系统中应用的 ISP 例子。注意：在这些步骤中产生的一些偏差，在不同操作系统的简易执行中影响不大。

4.1 找到正确的 USB 设备

枚举 USB 设备的方法取决于不同的操作系统。在 Linux 上，大多数分配提供一个叫 `lsusb` 的程序，这程序能搜索指定供应商 ID 和产品 ID 的 USB 设备。然后，`udevadm`（目前系统上使用 `udev` 设备文件系统）可以分列出 USB 型号 ID 和系统设备路径。LPC1300 系列的型号 ID 为“`NXP_LPC13XX_IFLASH`”。一旦供应商 ID、产品 ID 和型号 ID 被确认，你肯定会发现一个处于 ISP 模式的 NXP LPC1300 微控制器。在 Windows 和 Mac OS-X 中，这个步骤有点不同。在 Mac 上，可以从磁盘数据库取得的 USB 设备型号信息，也就没有去找实际 USB 设备。相反，直到正确的生产商和设备型号被找到，磁盘设备才开始检测。在 Windows 下，有相似的过程，通过 `iTuner UsbManager` 来检索磁盘信息。

4.2 改变 USB 设备控制权到文件路径

在Linux中，一个USB设备的控制权能被转换到文件路径，通过循环访问在udev文件系统的所有盘设备和为它们的系统设备路径查询udevadm，直到一个带有ISP的USB设备被找到。

- 1、lsusb以生产商ID和NXP ISP设备产品ID来命名。lsusb输出总线和设备ID到standard out（控制台）。输出结果用来为USB设备创建一个udev路径，例如，总线2设备3的路径为/dev/bus/usb/002/003。
- 2、一个盘设备名从/dev 截取，在我们的样品脚本中，我们使用一个与/dev/sd[a-z]匹配的通配符，并在设备间循环，这些设备与像/dev/sda, /dev/sdb, /dev/sdc,etc 这样的组合相匹配。这就可以在系统上找到磁盘驱动器，包括硬盘和USB大容量存储器。
- 3、完整的低级盘设备文件路径 用udevadm info -q path -n /dev/sdX 查找，如果盘设备与我们找到的USB设备有联系，那么返回的文件路径以低级USB设备文件路径开头。
- 4、最后，mount命令的输出用于决定在步骤2测试过的盘设备名安装到linux文件系统的哪个地方。我们现在有了一个文件系统路径用于我们的的USB ISP设备并能够访问firmware.bin

在 Mac 和 Windows 系统下，我们采用 盘操作（替代USB设备操作）和查询操作系统的盘符和文件路径

4.3 检查代码的read保护（CRP）

LPC1300 USB ISP的特色是设置USB盘的盘符指出CRP模式。USB ISP flash编程器的稳定设计将检查盘符来确认CRP没有使能。在Linux系统下，FAT盘文件系统盘符能用mttools包读出。遗憾的是，这不被推荐，因为它需要administrative来访问。另一种方法在Linux系统下读文件系统盘符是使用 mount -l 命令，如果你的系统mount版本能够支持的话。这将列出所有安装的文件系统和它们的盘符。在 Mac 和 Windows 操作系统中，盘符信息显示在DiskDescription目录或在Volume中来自 WMI 各自的目标检索。如果CRP1或CRP2被使能，在USB ISP 盘中的 firmware.bin 文件在新文件能写入之前肯定要删除来禁止代码保护。删除 firmware.bin 后，设备需要掉电和重连接，以使改变的CRP设置生效，这样固件就能更新。如果CRP3被使能，设备不能被擦除和固件不能被更新。

4.4 写文件到程序flash

在Windows操作系统下，几乎任一标准文件系统写时序都这样工作，Windows多样的FAT文件系统以ISP磁盘上的第一个空闲块作为起始，来分配块。在Linux操作系统中，设备自己能直接被写（用/dev），但需要同时使用dd来确认块序列。这方法不被推荐，因为它要求用户有 administrative 的权限。第二个选择是打开USB ISP 盘中已存在的 firmware.bin 并重写内容。如果文件打开后没有截位，新的数据将以与已有文件（ISP固件建立的文件）相同的规则写入。（记住“file metadata”是通过ROM代码创建的，且不存储到flash）通过conv=nocreat,notrunc.选项，使用dd来写入firmware.bin文件，可以实现重写。如果代码用C语言，尽可能的使用 open(path, O_RDWR) 或

`fopen(path, "r+")`。此外，在Windows操作系统中没有必要覆盖已有的文件，这样，就有存储空间可用，只要覆盖或删除当前文件以释放存储空间。在我们提供的Windows程序例子中，我们使了C# 函数 `System.IO.File.WriteAllBytes(filePath, firmwareData)`，在写之前做了简化处理。

4.5 卸载盘设备

大多数操作系统在一个延时之后将最终完整写到盘，但强制卸载是最后的有效步骤，以确保操作系统完成数据写入磁盘的操作，这样的话，当设备成功移除时就会告知用户。为了卸载操作成功，所有打开的文件包括 `firmware.bin` 文件必须先关闭。在Linux操作系统中，可以使用 `posix`（可移植接口）标准 `umount` 命令。通常，卸载自动加载的USB大容量存储设备不需要管理权限。在Mac操作系统中，因为磁盘仲裁器能够自动管理加载操作，使用 `DADiskUnmount` 系统呼叫是最好的路径。在Windows操作系统中，大多数用户使用自我卸载或使用 `SetupAPI` 里的 `CM_Request_Device_Eject` 函数。我们的例子中 Windows ISP 更新器不卸载设备当它，反而，它会显示一个对话框提醒用户。

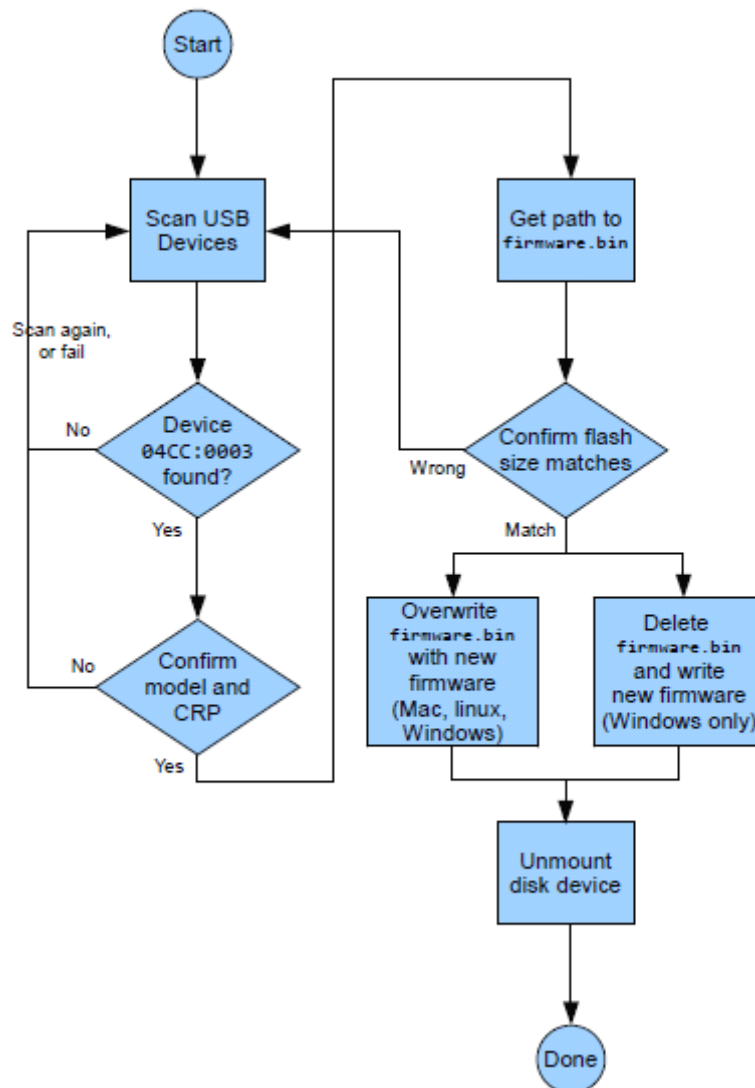


图4

5、自动进入USB ISP

在前面的章节，我们解释了在USB ISP模式下，PC更新设备固件的过程。当固件未编程，或者拉低一个引脚（在LPC134x系列是PIO0.1脚），设备复位/上电，NXP LPC1300微控制器默认为ISP模式。有时，ISP模式下更新固件、连接设备，用户不想去按一下按钮。这样的话，我们可以设计出允许ISP模式下编程的固件。

在LPC1300片上ROM中，通过In-Application Programming API请求，可以实现ISP模式下编程。固件更新之后，用户需要重启设备以使新的固件生效。下载新固件之后，新固件可以预设置看门狗定时器来重新设置LPC1300。下面的流程图显示了设备自动化ISP过程。名为“autoisp”的软件实例演示了这一技术。

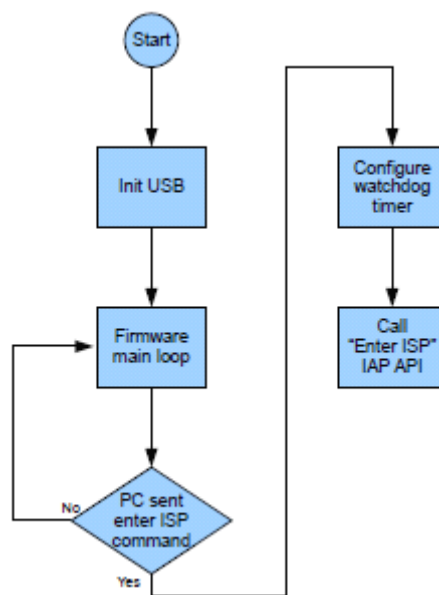


图5

6、软件例子

为了尽早建立用户界面友好的固件更新工具，NXP发布了三个实例（一个ISP下载程序并提供了源代码）。提供一个基于C#的应用程序窗体，可运行于Windows XP、Windows Vista和Windows 7。使用Microsoft Visual Studio C# Express 2010，可以让Windows的应用得到扩展。一个用C语言编写的命令行工具可以在Mac上使用。这个工具在Apple Xcode中得到加强，可以运行在OS-X 10.5或更高的版本上。最后，Ubuntu Linux也有一个程序化的脚本，已在Ubuntu 10.04通过测试，并可以运行在具有相同udev文件系统和工具的不同版本的Linux上。

所有工具都是开放的、没有限制，可重复使用，除了Windows应用[取决于iTuner(<http://ituner.codeplex.com/>) 计划的USB磁盘设备等级]。这是一个在开源许可下,要求公开源代码的衍生著作。

应用手册中还包括一个实现代码读保护的blinky的程序。这个程序用在the LPCXpresso IDE，在the LPCXpresso板上让LED闪烁，在嵌入式板上让七段数码管显示屏计数。有九种程序编译版本，所以，设备可以反复的编程（使用不同的CRP模式和LED闪烁/计数速率）。还有一个叫“autoisp”的程序。这个例子演示了如何在程序控制下进入ISP模式。在LED闪烁十秒钟之后，进入ISP模式。如果设备连接到PC上，它将顺序输出，固件开始闪动。这期间，LPC1343进入ISP模式，看门狗定时器继续工作。时间到了之后，LPC1343复位并再次运行固件。

最后，还有一个叫“padto”的小命令程序。这个程序用来从开发工具获得二进制输出，并填补成0xFF字节，直到与LPC上的flash大小相匹配。这种匹配有助于PC主机上的ISP下载程序，确保固件适用于连接到USB上的特定的LPC134X。

Windows NXPISP 实用工具

这程序是基于Microsoft .NET，当setup.exe 程序运行时启动，附带的Visual Studio Installer 将检查电脑和下载 .NET 的最新版本（如果还未安装）。因此，当你第一次运行setup.exe程序时，请确认你已经通过宽带连接到互联网上。在Windows .NET子系统更新完之后，NXPISP程序将自动安装并运行。NXPISP运行时，点击“Select Firmware.”一个文件选择对话框将会弹出，让你选择一个.bin文件。这个.bin文件将用“Criterion for Valid User Code”核对，即向量表的前8个32位字总和为0。如果 bin 文件符合要求，“Select Firmware”按钮的那个绿色的检查标志将显示正确，否则。显示一个红色的X。

实际烧写一个设备，单击“Update Firmware.”，“Update Firmware.”按钮工作起来像一个触发器，而且当单击时能停留在压低或再一次单击时释放，当按键被按下，电脑将扫描NXP ISP设备，如果有一个被发现，它的固件将被更新，而且“Update Firmware”按钮的绿色检查标志显示正确并且按钮将释放，如果没有设备被找到，“Update Firmware”按钮将停留在压低而且工具将等待设备被连接。设备被找到和更新后，绿色检查标志将显示而且“Update Firmware”按钮将弹起，如果实用工具等待设备连接（“Update Firmware”按钮仍然压低）它将通过再一次单击“Update Firmware”按钮取消而且导致弹起。

安全弹出USB ISP设备以确保新固件已经写入（Windows NXPISP工具不会卸载设备）。通过Windows硬件弹出图标和Windows浏览器弹出选项来安全弹出设备。这个步骤失败将导致固件恶化。