UFCOM 虚拟串口驱动程序用户指南

Added by 陈军, last edited by 陈军 on 四月 10, 2018

本文档适用于 UFCOM 1.7.9 版本。

1. UFCOM 简介

UFCOM(USB Flexible COM Port 的缩写) 是由新大陆自动识别公司开发的全新的虚拟串口驱动程序，2017 年起向我们的客户提供，它配合虚拟串口模式的 USB 扫描枪使用，提供和 USB 扫描枪之间的双向通信功能。UFCOM 可运行于 Windows XP ~ Windows 10 x86 & x64 所有版本，也包括同时代的各版本 Windows Server 。

UFCOM 提供的功能和 Windows 系统自带的通用虚拟串口驱动 usbser.sys 类似，可完全替代 usbser.sys ，不但修正了 usbser.sys 的很多糟糕行为，还提供了更强大的功能。UFCOM 的功能亮点如下：

1. 多扫描枪支持。

   支持多只 USB 设备（下文用“扫描枪”指代）同时工作，至少同时支持 8 只，每只扫描枪会在系统中生成一个虚拟串口设备（称 VCOM），各自对应不同的串口号。

2. VCOM Lifemode （虚拟串口生命期模式）。

   通过设定 VCOM Lifemode ，用户可以控制扫描枪被拔除后，VCOM 设备是否继续存在。在很多应用场合下，我们希望扫描枪意外拔除时 VCOM 能够持久存在，扫描枪插回后自动关联到原先的 VCOM，这样，使用 VCOM 的应用程序就不会因为 VCOM 句柄失效而停止工作。

3. Port number Bindmode （串口号绑定模式）。

   通过设定 Bindmode，用户可以指定不同型号、不同序列号的扫描枪是关联到同一个串口号还是不同的串口号。共有 4 种绑定模式可

选。

4. 动态设备显示名。

   在 Windows 设备管理器中，VCOM 设备的名称会依据设备状态动态显示。比如正常工作的扫描枪，设备名称会有 [online] 前缀，拔除的扫描枪，设备名称会有 [offline] 前缀。

5. VCOM Settings 用户界面。

   设备管理器中的 VCOM 设备属性对话框，提供了专门的用户界面来方便地设置 VCOM 的各项属性，可在此查看与修改 Lifemode, Bindmode 以及指定 VCOM 使用的串口号，检查 USB 链路故障，查看数据流量与流速统计等。

针对新大陆扫描枪老用户的一些背景信息

在 UFCOM 之前，我们给用户提供了两套虚拟串口驱动程序。

- USB Datapipe 驱动包: 当扫描枪通信模式被设为 USB Datapipe 或 com0com 虚拟串口时，你需要安装这个。
- CDC Virtual COM Driver: 当扫描枪通信模式被设为 USB-CDC 模式、且 PC 端是 Windows 8.1 或更早版本时，你需要安装这个。此驱动包实质上是引用了 Windows 系统上自带的 usbser.sys，对，仅仅是引用，这意味着我们无法跨大版本升级 usbser.sys，比如，无法将 Windows 10 的 usbser.sys 用到 Windows 7 上。

现在 UFCOM 可以完全替代那两套老的驱动程序，即，不论你将扫描枪设成三种“虚拟串口”模式的哪一种，UFCOM 都将为它们生成 Windows 上的虚拟串口。

UFCOM 同时提供了 Datapipe API 支持，意即，先前使用 udp_op.dll 的老式应用程序也可以配合 UFCOM 工作。注意：部分使用 Datapipe API 的老版本应用程序有 bug，需要更新后方可配合 UFCOM。

2. UFCOM 与虚拟串口设备管理

2.1. UFCOM 安装与卸载

【背景知识】

为了让我们的用户对于 Windows 系统的驱动程序安装有个清晰的认识，这里介绍一些关键知识。所谓的“安装驱动程序”其实包含了三个步骤：

1. 将驱动包(driver package)安装到称作 DriverStore 的系统目录中。可以认为这是个单纯的文件拷贝动作。在 Windows 7~10 上，该目录是 C:\Windows\System32\DriverStore\FileRepository 。Driver package 的内容，通常由 inf, sys, cat 构成，也可能包含一些 dll 和 exe 。其中的 inf 是关键，可以认为 一个 inf 就代表了一个驱动包。

2. Windows 为探测到的硬件设备（也包括虚拟的软件设备）匹配驱动程序。意思是：一个设备被 Windows 探测到后，会向 Windows 通告自己的 hardware-id，Windows 在 DriverStore 中检索各个 .inf 文件的内容（一个 .inf 文件代表一个驱动包），看有没有哪些 inf 宣称自己匹配这个 hardware-id 。有可能多个 inf 都能匹配当前的 hardware-id，但 Windows 同一时刻只会选择其中的一个来驱动当前这个设备。Windows 通常会自动选择最佳的那个，比如有数字签名的，且日期最新的那个。不过如果用户愿意，可以手工切换成非最优的那个驱动包。

3. 装载驱动程序代码(.sys)到内存中执行。硬件设备至此真正地开始和 Windows 交换数据。

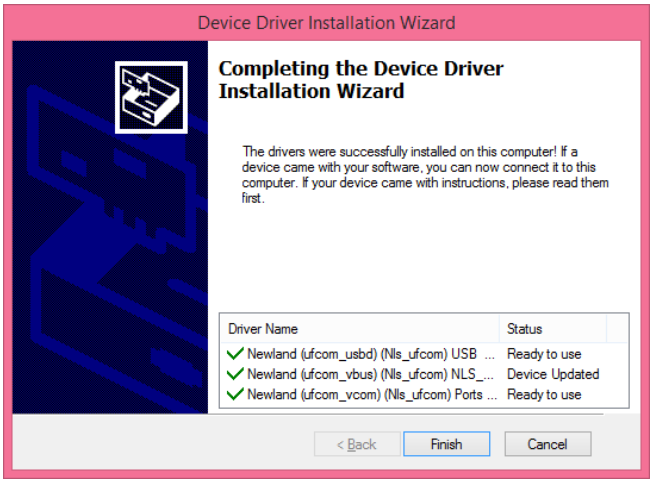
当 Windows 为一个具体设备匹配了一个驱动包后，该匹配关系被 Windows 记录于注册表中（该记录位置被称为 hardware-key），即使设备拔除(Unplug)也不会被删除。当设备第二次插入时，Windows 就无需重新搜索 DriverStore 进行驱动包匹配，而是直接装载之前记录的 .sys 开始执行。串口设备当前使用的串口号（PortName）也被记录于 hardware-key 中，因此扫描枪拔除再插入方能得到和上次相同的串口号。 只有当用户在设备管理器中对一个设备实例进行卸载(Uninstall)操作后，设备实例信息从注册表中删除（hardware-key 被删除），此后再插入设备就会重新触发驱动包搜索与匹配动作。

【 UFCOM 安装过程 】

从 UFCOM 1.4.0 起，UFCOM 安装包以 msi 形式提供，msi 是 Microsoft Windows 平台标准的安装包格式。

用户双击 msi 文件，当有对话框弹出时，回答几次 Next 即可完成安装。安装成功的标志是看到如下的 Completing the Device Driver Installation Wizard 信息，其中应该有三个绿色的勾。

如果 Windows 提示需要重启，请重启 Windows 以保证安装/升级完成。



UFCOM msi 安装包执行了如下动作：

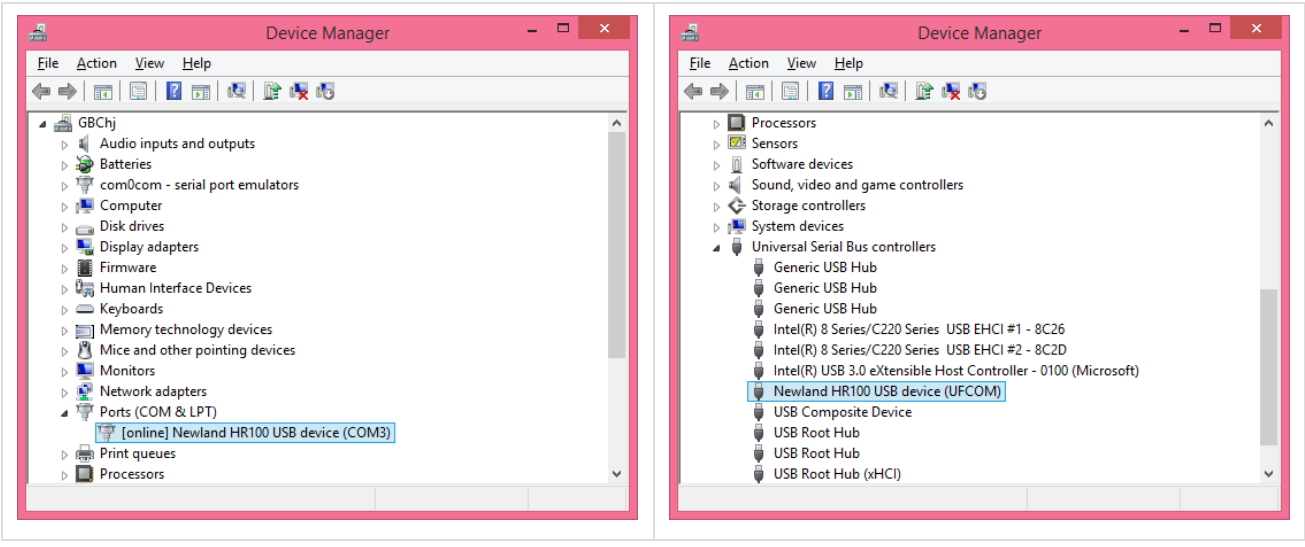
1. 将 inf 表达的驱动文件释放到 `C:\Program Files\UFCOM` (对于 64-bit Windows 是 `C:\Program Files (x86)\UFCOM`)。
2. 执行 DPInst.exe 将 inf 表达的驱动文件 拷贝到 DriverStore 中。
3. 如果当前 Windows 上已经存在有能够匹配 UFCOM 驱动包的设备实例（比如已插入的扫描枪），这些设备实例所匹配的驱动包将会被更新为使用 UFCOM 驱动包。比如，Windows 10 上使用系统自带驱动的 USB-CDC 设备将会被更新为使用 UFCOM 。
4. 如果旧版本的驱动代码(.sys)无法从内存中被卸载（比如相应串口正被打开着），Windows 会提示用户重启整个系统。

对于首次安装 UFCOM 的情况，不用担心插入扫描枪和执行安装包的先后顺序，两者皆可。

对于升级安装，我们建议关闭所有使用操作扫描枪的应用程序后再执行安装包（不必拔除扫描枪），这样可以最大程度减小需要重启 Windows 的概率。

现在请插入 USB 扫描枪（如果还未插入的话），Windows 将为扫描枪匹配并装载驱动程序。第一次插入扫描枪，匹配驱动程序可能会花费几秒至几十秒时间（依 Windows 运行速度而异）。

为了确认扫描枪被成功地驱动，需要打开设备管理器进行验证。在设备管理器中，我们实际上会看到两个新设备生成。其中一个是虚拟串口（左图，称 VCOM），另一个是真实的 USB 设备（右图，暂称其物理设备）。



如果将扫描枪拔除，物理设备必定会从设备管理器中消失(unplugged 状态)，但 VCOM 设备不一定会消失。VCOM 是否跟随物理设备消失，由 VCOM Lifemode 来决定，用户可以自行设定 VCOM Lifemode。

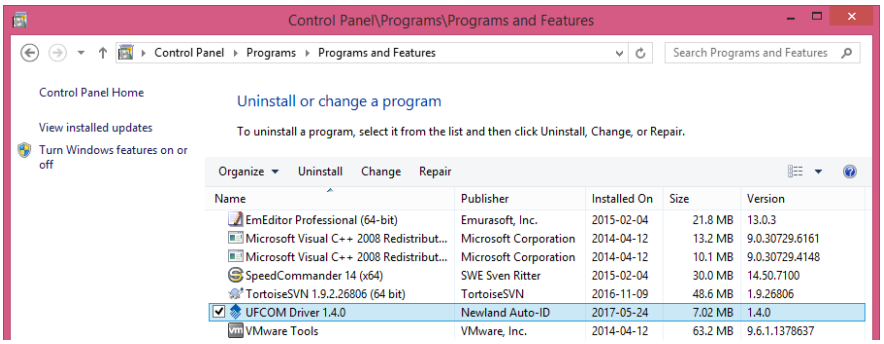
【 UFCOM 升级安装 】

执行新版本的 UFCOM 安装包即可达到升级效果。 新版本的 UFCOM msi 会自动卸载旧版本。

若安装了新版本后回头去安装旧版本，旧版本的 msi 会拒绝安装。如果明确要安装旧版本，用户必须先手工卸载新版本。

【 卸载 UFCOM 】

在控制面板中(AppWiz.cpl)用户可以对 UFCOM 进行卸载。找到名字如 **UFCOM Driver 1.4.0** 的条目，双击之即可卸载。



此处的卸载动作意味着：

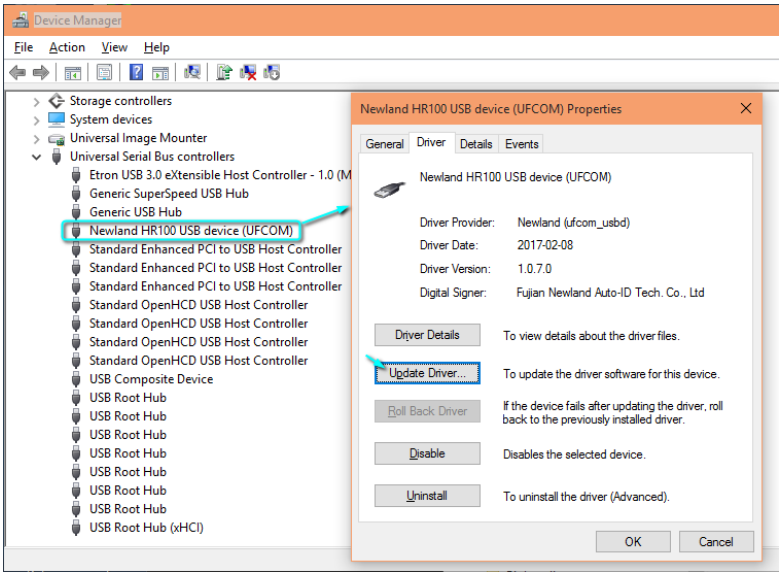
- UFCOM 驱动包从 DriverStore 中删除。
- 当前匹配 UFCOM 驱动包的所有设备节点将重新进行驱动包匹配，也许匹配到另一个驱动程序包、或 Windows 自带的驱动程序。

【 为 USB 设备手动切换驱动程序包 】

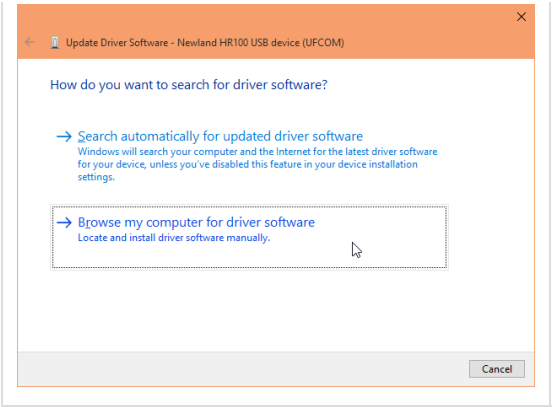
如果因为某种原因你需要针对某个 USB 设备使用老式的 Datapipe 驱动、或更换为 Windows 自带的 usbser.sys ，你不需要卸载 UFCOM 驱动包，只需要针对具体设备更改所装载的驱动程序即可，这是 Windows 自身提供的功能。

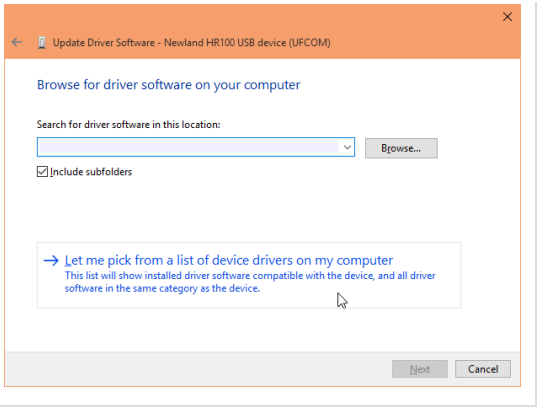
以 Windows 10 为例，操作方法如下：

在设备管理器中找到真实的物理设备节点（特别提示：非 VCOM 设备节点），打开其属性对话框，点击 Update Driver 。

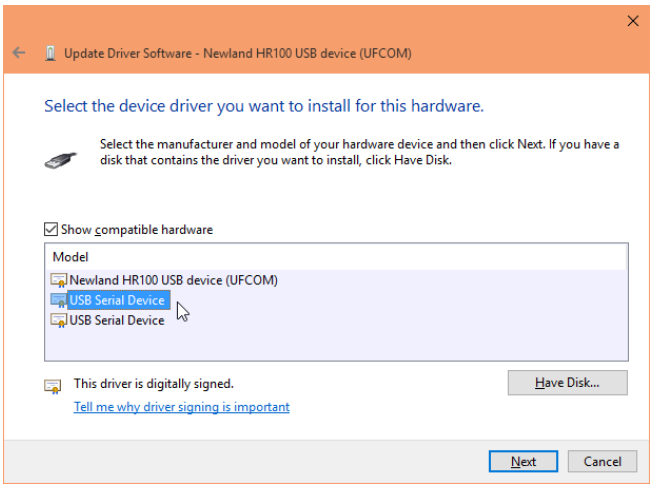


浏览计算机以查找驱动程序软件.	从计算机的设备驱动程序列表中选择.





在 “显示兼容硬件” 列表框中，选择一个驱动包。 此处，驱动包是用 inf 中的设备名称来标识的。本例中， Newland HR100 USB device(UFCOM) 代表 UFCOM 驱动包——标志是设备名称有个 "(UFCOM)" 后缀， USB Serial Device 则是 Windows 自带的 usbser.sys 驱动包。



最后点击一次 Next ，确定使用选中的驱动包。

当然，你可以用同样的方法让 USB 设备换回 UFCOM 驱动包。

> 一个提示：若在同一台电脑上同时安装 UFCOM 和 老式 Datapipe 驱动、且希望 Datapipe 设备由老式驱动接管，你还应该明确告知 UFCOM 放弃对 Datapipe 的接管。
>
> 操作方法是：UFCOM Global Settings → Datapipe device lifemode 设为 “Never”。
>
> 普通用户无需如此折腾，让 Datapipe device lifemode 保持默认值 “On device present” 即可。

-

2.1.1. 静默安装和卸载 UFCOM 驱动包

UFCOM msi 支持标准的静默安装与卸载方法。静默安装使得你可以将 UFCOM 驱动包的安装动作集成到自动化脚本中，以便在无人值守的情况下完成安装动作。

启动静默安装的命令格式如：

```
msiexec /q /i ufcom-1.7.2.msi
```

提示：需要以管理员权限启动以上命令方可成功。成功的标志是， AppWiz.cpl 中出现 UFCOM 条目。

如果在批处理中启动静默安装，且希望等待 msi 安装完毕后再执行批处理的下一个语句，则写法为：

```
start /wait msiexec /q /i ufcom-1.7.2.msi

echo %ERRORLEVEL%
```

一个提示： 你不可以在 msiexec 的命令行上写 `.\ufcom-1.7.2.msi` ，意即，不可以有 `.\` 前缀，那会导致 msiexec 什么也不干就失败退出了。这显然是一个 bug ，但微软一直没有修复该问题。用绝对路径前缀则是可以的。

静默安装新版本，中途将自动卸载旧版本。

msiexec 的退出码（反映为批处理语句中的 %ERRORLEVEL%）可告知静默安装是否成功，退出码为 0 表示成功，其他为失败。有如下典型失败原因：

1. 未使用管理员身份启动 msiexec 。
2. 新版本存在的情况下试图安装旧版本。
3. (1.7.5)用新版本升级旧版本的过程中，某些旧版本文件被占用无法替换、需要重启方可替换。

为了实施静默卸载，我们需要提供原始的 msi 安装包文件。比如，当前系统上安装的版本为 1.7.2，卸载时就要用到 ufcom-1.7.2.msi 。静默卸载命令行如：

```
start /wait msiexec /q /x ufcom-1.7.2.msi
```

2.2. UFCOM 工作行为配置

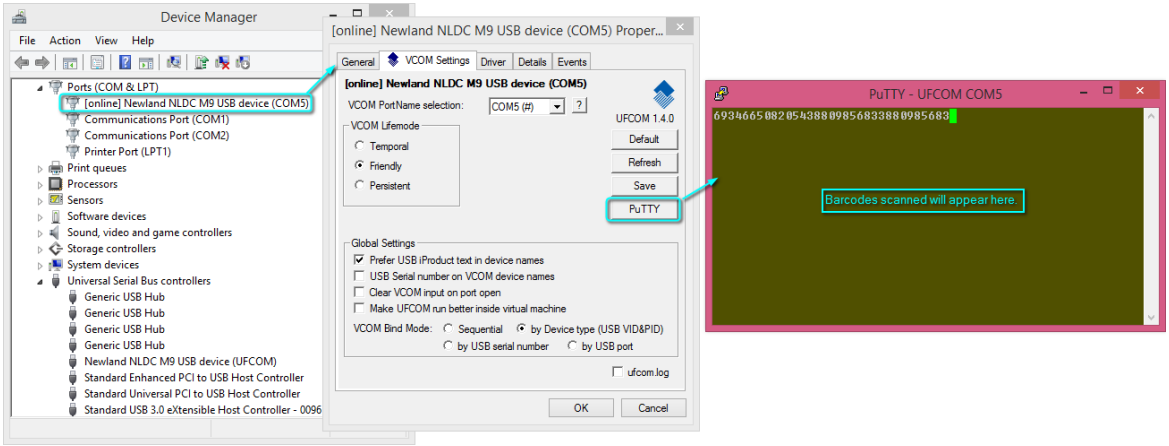
对于 UFCOM 生成的 VCOM ，你可以用任何的串口通信工具打开它，通过标准的 Windows 串口操作 API 来读写串口数据，达到跟 USB 设备通信的目的。常用的串口工具如 Windows XP 自带的超级终端，Windows 7 上没有超级终端了，好在 UFCOM 提供了免费的 PuTTY 可进行简易的串口数据收发测试。如果需要对扫描枪进行高级设置，可以用新大陆提供的 EasySet 。

由于 Windows 的串口操作 API 最初是贴合 RS-232 物理串口设计的，它没有考虑到 USB 设备特有的行为（比如可以被随时拔除插入、节能管理等），这些行为无法用串口 API 来控制，因此，UFCOM 提供了专门的手段来控制这些特有的行为。

2.2.1. 用 PuTTY 测试串口数据收发

在设备管理器中找到 VCOM 节点，双击之（或右键菜单→属性），弹出设备属性对话框，转到 **VCOM Settings** 选项卡（下文称 VCOM Settings UI），点击 PuTTY 按钮，即开启了可以和扫描枪通信的串口终端。

- 此时让扫描枪扫描条码，PuTTY 窗口应能看到条码内容.
- 往 PuTTY 窗口中敲入字符（会有本地回显），这些字符将被送往扫描枪。技巧：若希望发送一个很长的字符串给扫描枪，可以先在文本编辑器中写好该字符串，拷贝之，回到 PuTTY 窗口，鼠标右键点击 PuTTY 窗口空白处（意即粘贴），即可将字符串发送给扫描枪。



2.2.2. VCOM 生命期模式(Lifemode)

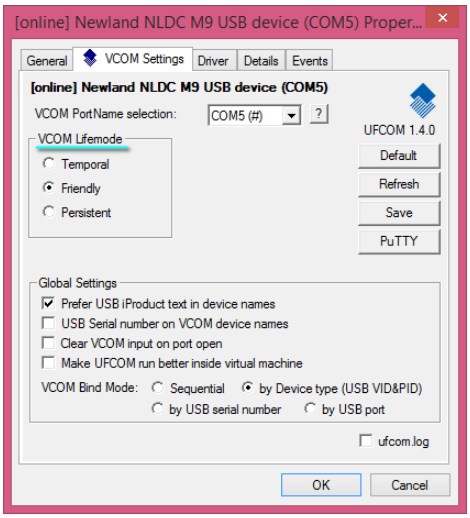
VCOM Lifemode 指出 VCOM 设备什么情况下存在，什么情况下消失。此处，“消失”的意思是，在设备管理器中无法看到 VCOM 设备（除非要求设备管理器显示隐藏的设备）。

Lifemode 名称	VCOM 显示名称前缀	VCOM 存在的条件
短暂模式 (Temporal)	<online>	扫描枪插入时 VCOM 出现，扫描枪拔除后 VCOM 立即消失。
友好模式 (Friendly)，此为默认设定	[online] [offline]	扫描枪插入时 VCOM 出现。 扫描枪拔除时，若已有应用程序正在使用 这个 VCOM（应用程序持有该 VCOM 的句柄），VCOM 不会消失，原先的句柄也仍旧有效。在 VCOM 句柄保持打开的情况下，若扫描枪被插回，则扫描枪恢复和原先 VCOM 的连通。意即，获取了 VCOM 句柄的应用程序，不会受到物理扫描枪短暂拔除的影响，达到一种“自动恢复连接”的效果。 当“扫描枪拔除”和“其对应的 VCOM 句柄被关闭”这两个条件同时成立时，VCOM 消失。
持久模式 (Persistent)	[[online]] [[offline]]	VCOM 永远存在。 持久模式使得 VCOM 像一个物理串口般永远存在于系统中，不论扫描枪是否插入系统。

VCOM 设备显示名称 online, offline 前缀的含义：

- online 表示扫描枪当前插在 Windows 机器上且驱动程序工作正常。
- offline 表示扫描枪已被拔除(physically unplugged)。

设置 Lifemode 的操作方法：在设备管理器找到 VCOM 设备节点（不是找物理设备节点），双击打开设备属性对话框。切换到 "VCOM Settings" 选项卡。此处你可以调整 VCOM 的 Lifemode 。

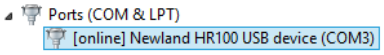


特别提示：插入两把扫描枪将会生成两个 VCOM 设备节点，它们的 VCOM Lifemode 是各自独立的。

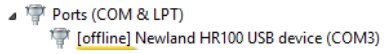
> 你需要 Windows 系统的管理员权限方可修改 VCOM Settings 。

【观察 VCOM 设备 "online/offline" 显示名前缀的变化】

扫描枪插入 Windows ， 对应的 VCOM 设备在设备管理器中将被附上前缀 online 。



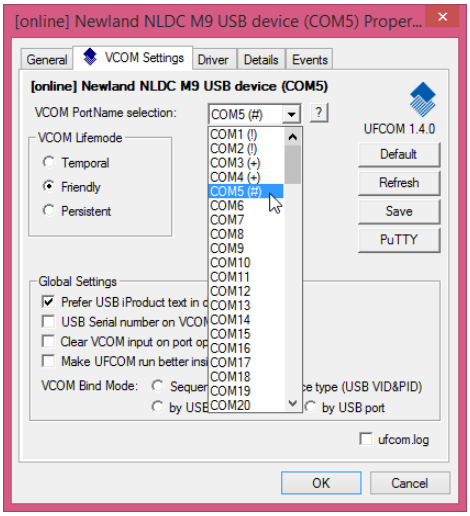
扫描枪拔出，且根据 VCOM Lifemode 规则判定 VCOM 设备应继续存在的情况下，前缀变为 offline 。



2.2.3. 修改 VCOM 串口号

这是一个常用功能。

当 Windows 生成一个新的 VCOM 设备时，系统会为 VCOM 自动选取一个空闲的串口号，通常是分配一个数值最小（从 COM3 开始）且未占用的。当我们对系统自动分配的串口号不满意时，可以通过 VCOM Settings UI 修改它。



在选用新的串口号前，请留意下拉列表中各个备选串口号尾部的 + - ! 后缀，它们有如下含义：

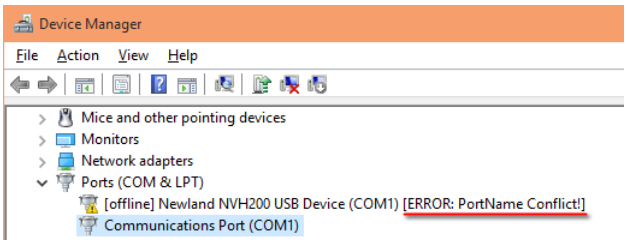
符号	含义
#	此串口号正在被自己使用。
!	此串口号正在被其他串口设备使用，这些串口设备当前真实存在于系统中（且是 plugged-in 状态）。
+	有其他某个 VCOM 设备已使用了此串口号，只不过这些 VCOM 设备处于 unplugged 状态。
-	Windows 的 ComDB 数据库标记该串口号已被某个串口设备使用，这些串口设备暂处于 unplugged 状态。

不必记忆那些符号的含义，点击下拉框右侧的小问号即会得到解释。

我们建议新选用的串口号是不带任何后缀标记的(表示空闲可用)，以防止干扰其他串口设备的工作。

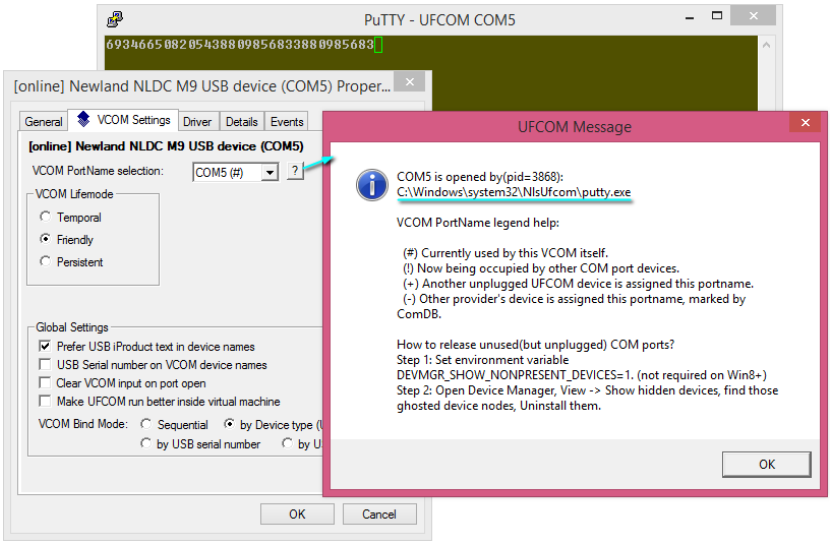
- 如果你选择 (!) 后缀的串口号，必定会遭遇失败。
- 选择 (+) (-) 后缀的串口号是可以的，比如图中的 COM7；只不过，当原先使用 COM7 的那个设备被插回时，原先那个设备可能会工作异常、也可能它会另行选取一个新串口号后正常工作，具体行为由该设备的驱动程序决定。

如果选择了 (!) 标注的串口号，UFCOM 会让该设备节点报错，出错原因会显示在设备名称上。比如，在物理串口 COM1 存在的情况下，强行将 VCOM 串口号设为 COM1，设备管理器将提示 [ERROR: PortName Conflict!] 如图：



2.2.4. 获知 VCOM 当前被哪个程序打开

点击 PortName selection 右侧的小问号，我们能得知当前 VCOM 是否被某个程序打开着。如下图，告知 COM5 正被 putty.exe 打开。



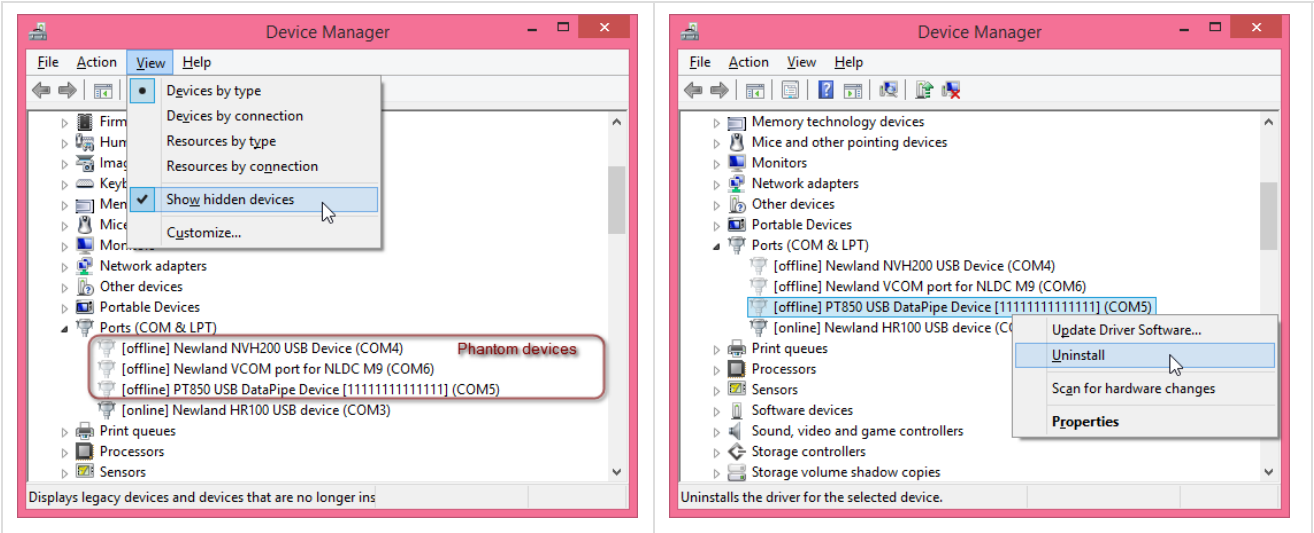
-

2.2.5. 清除不再使用的 VCOM 设备，释放串口号

当我们拔除 USB 扫描枪、且关闭其 VCOM 设备句柄后，我们会看到扫描枪对应的 VCOM 设备节点会从设备管理器中消失（除非 Lifemode=Persistent），但事实上，这种消失更像是隐藏——因为 VCOM 设备节点的信息还存在于注册表中，这些信息包括：此 VCOM 的串口号、Lifemode 等。当“同一个” VCOM 设备实例再次出现时，Windows 就能记得这个 VCOM 之前用的是哪个串口号、哪种 Lifemode，并应用那些设定，这是非常合理的行为。（注：VCOM 设备实例是用所谓的 "设备实例路径(Device instance path)" 字串来标识的，设备属性 Details 选项卡能够查到当前设备的这个字串）

当我们往系统中插入多把不同型号的扫描枪后，可能会有多个串口号被系统标识为“已分配”。若我们不再使用之前插过的扫描枪、想回收那些串口号，即，让系统重新认为它们是空闲串口号，我们得将将占用那些串口号的设备从系统中卸载。

操作方法是：让设备管理器显示隐藏设备 View → Show hidden devices 。此时我们将看到之前插入但现在处于 unplugged 状态的设备（左图）。这种设备被称为影子设备(phantom device)，它的设备图标显示成残影的样子。对影子设备执行卸载动作，可回收它们占用的串口号（右图）。



> 提示：在 Windows XP~Windows 7 系统上，为了看到影子设备，你得事先设置环境变量 DEVMGR_SHOW_NONPRESENT_DEVICES=1 再打开设备管理器。 Windows 8.0 以上的系统无需此额外步骤。

2.2.6. 串口号绑定模式(COM Port Bindmode)

串口号绑定模式是一个全局设定，它会影响系统给 USB 扫描枪分配串口号的规则。

UFCOM 提供了四种绑定模式供选择，你需要发挥一些想象力来理解它。

绑定模式	含义
顺序绑定 (Sequential)	UFCOM 对插入系统的所有扫描枪都不作差别对待，最终效果是，串口号是按顺序分配的。 举例，我们手头有两只扫描枪，插入第一只得 COM3，同时插入第二只得到 COM4。将两只扫描枪都拔除，之后无论先插入哪一只，都会先得到 COM3 。
按型号绑定 (by Device type) , 默认值	简言之，UFCOM 会关心扫描枪的型号，不同型号的扫描枪会分配到不同的串口号。扫描枪的型号是用 USB 设备描述符中的 VID,PID 二元组标识的。 举例，我们手头有一只 HR100 和一只 HR200（它们是不同的型号），第一次插入 HR100 时得到 COM5 ，第一次插入 HR200 得到 COM6。之后，插入 HR200 将总是得到 COM6——不论 HR100 是否已插入；同理，插入 HR100 总是得到 COM5。意即，COM5 被绑定到 HR100 ，COM6 被绑定到 HR200；这种绑定关系一直持续到你将相应的 VCOM 设备卸载为止。
按序列号绑定 (by USB serial number)	不同序列号的扫描枪将得到各自不同的串口号。USB 扫描枪的序列号是由 USB 设备描述符中通告的的序列号 (iSerialNumber 字串)决定的。 当我们需要让同型号的两把扫描枪绑定到不同的 VCOM 时，可以用此设定。但请注意，有些型号的扫描枪自身并未烧录序列号，这些扫描枪就只能回退至按型号绑定的行为。
按 USB 端口绑定 (by USB port)	插入不同 USB 端口的设备将得到不同的串口号。先后插入同一个 USB 端口的不同设备，不论设备型号，都将得到相同的串口号。 此绑定模式在 Windows 7 或更高版本上可用。

额外提示：

- 目前，切换绑定模式会使得当前已经打开的 VCOM 句柄失效（无论何种 Lifemode），应用程序必须重新打开它们。
- 切换绑定模式后，在前一种模式时已经保留的串口号并不会被自动回收。这意味着，对于同一支扫描枪，在不同的绑定模式下它将得到不同的串口号——除非你稍后手动将它们修改成相同的。

2.3. 故障诊断

2.3.1. 驱动包安装过程故障

【 收集安装日志的方法 】

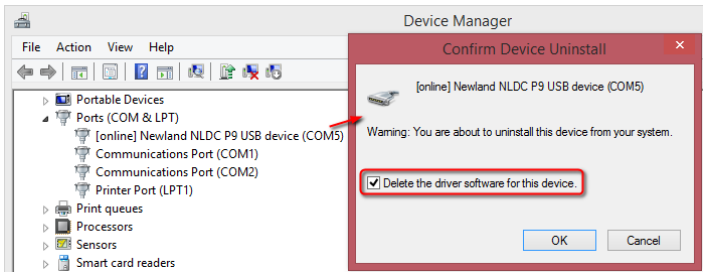
驱动包安装故障需要日志文件来帮助诊断。需要收集如下日志文件：

功能组件	日志文件收集方法
msiexec 日志	需要开启 msiexec 日志生成功能。不要双击运行 msi ，而是在管理员 CMD 窗口中执行如下命令： `msiexec /i ufcom-1.7.2.msi /l*v my.log` 收集 my.log 。
DPInst 日志	安装动作结束后，收集 `C:\Windows\DPINST.LOG` 。 提示：建议预先删除（或改名）原有 DPINST.LOG ，方便观察新内容。
SetupAPI 日志	安装动作结束后，收集 `C:\Windows\Inf\setupapi.dev.log` 。 提示：建议预先删除（或改名）原有 setupapi.dev.log ，方便观察新内容。 提示：Windows XP 上，日志文件是 `C:\Windows\setupapi.log` 。

【 修复驱动包 】

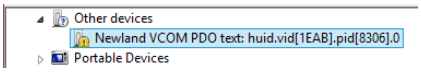
有时候，UFCOM 驱动程序文件会遭遇意外损坏，导致驱动程序无法正确装载，此时可以尝试修复驱动包让其恢复正常。

比如，当我们从设备管理器中删除 VCOM 设备节点时，不小心要求 Windows “删除设备的驱动程序软件”，这将导致之后插入的 USB 扫描枪无法工作。

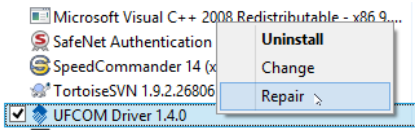


当重新插入 USB 扫描枪时，VCOM 呈现 黄色感叹号状态。

Device status 显示为 The drivers for this device are not installed. (Code 28)



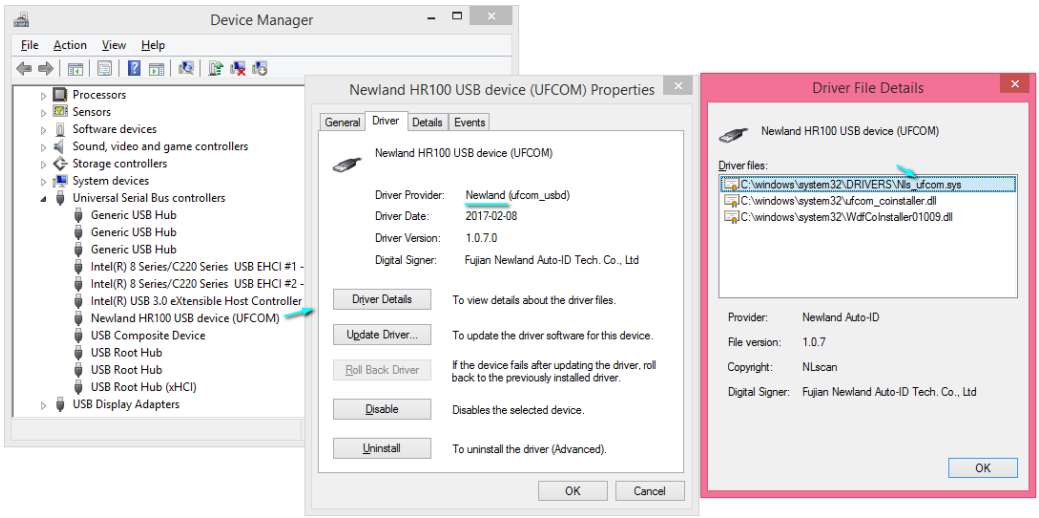
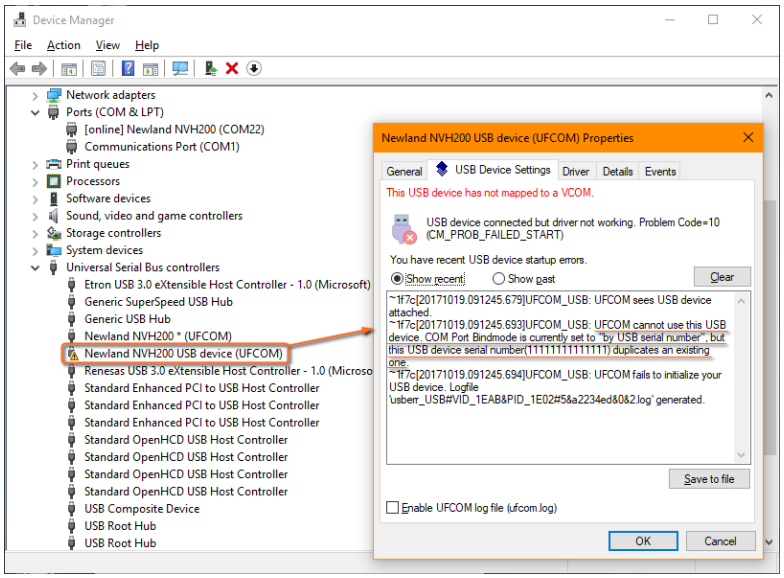
为了修复该故障，我们可以将 UFCOM msi 重新执行一遍，或者，更简单地，在 Appwiz.cpl 中对 UFCOM Driver 进行右键→修复。



2.3.2. VCOM 设备无法生成

有多种情况可能会导致 VCOM 设备无法生成

故障分类	现象、原因与解决方法
USB 物理设备没有被 Windows 检测到	现象：在设备管理器中，"通用串行总线控制器(Universal Serial Bus controllers)" 类别下方，没有看到新设备节点的出现，在设备管理器其他设备类别下也没有看到新设备节点出现。 可能的原因： • 扫描枪没有插入 Windows 主机。 • 扫描枪 USB 功能未开启。 • 扫描枪有硬件故障。 • Windows 上未正确安装 USB 系统驱动（指 USB 控制器驱动）。 现象：插入扫描枪时，Windows 任务栏弹出气泡，说 **USB Device Not Recognized** （无法识别的 **USB** 设备）。 可能的原因： • 扫描枪 USB 硬件存在故障或工作不稳定。 • USB 线缆故障。 检测 USB 物理设备的存在是由 Windows 自身进行的，以上错误都表明 Windows 还未正确探测到 USB 外设的存在，此时设备的控制权还未转移到 UFCOM 手上。
USB 扫描枪当前为键盘模式	现象：扫描枪插入后 Windows 没有生成虚拟串口，但扫描到的条码可以显示于文本编辑器（比如记事本）中。 这说明扫描枪的 USB 通信模式是 USB keyboard（大多数扫描枪的出厂默认）。你需要让扫描枪扫描一个特殊设置码来将其切换为 UFCOM 可识别的模式，比如 USB-CDC 。

故障分类	现象、原因与解决方法
Windows 检测到 USB 物理设备，但使用了其他的驱动程序	现象：USB 设备显示名称不是以 "(UFCOM)" 结尾。 双击 USB 物理设备查看 Driver Provider ， 正常应该显示有 Newland 字样，且 Driver Details 中有 Nls_ufcom.sys 。  如果发现使用了其他的驱动程序（比如使用了 Windows 自带的 usbser.sys 或使用了老式的 Datapipe 驱动），请将其切换为使用 UFCOM。 操作方法前前头 "为 USB 设备手动切换驱动程序包" 小节。 请注意：倘若扫描枪使用了其他的驱动程序，则扫描枪物理设备节点不一定处于 Universal Serial Bus controllers 类别中，驱动程序可以决定它处于任何的类别中。当然，对于 USB 扫描枪，其他驱动程序常用的类别是 “Ports (COM & LPT)”。你可能需要多次拔插扫描枪来找到其所在的类别。
USB 物理设备已检测到，但驱动程序工作不正常	现象：USB 设备图标上有黄色感叹号。 诊断方法：查看 USB 物理设备的属性，转到 UFCOM Device Settings 选项卡，此处会展示 UFCOM 检测到的 USB 启动失败原因。比如，启动全局选项 "When Bindmode is by USB serial number, require unique SN" 的情况下，插入两只同型号且序列号相同的扫描枪，第二只扫描枪无法工作，看到如下错误信息：  你可以将此处的故障信息截屏或保存成文件发送给我们诊断。

故障分类	现象、原因与解决方法
USB 物理设备正常，但 VCOM 设备未出现	此情况比较罕见。为了帮助我们诊断原因，你可以开启 UFCOM 的日志。开启日志后，让故障重现，将日志文件发给我们诊断。 开启日志的方法：打开任意一个 USB 硬件设备节点的属性对话框，勾起底部的 **Enable UFCOM log file (ufcom.log)** 。 日志文件位置： C:\Windows\Temp\ufcom.log 。 日志文件在驱动程序每次重新装载时会清空重建（比如重启 Windows 后），因此不用担心日志文件无限增长。 提示：每次要查看 ufcom.log 时，建议将其拷贝一份（在 Explorer 中 Ctrl+C 再原地 Ctrl+V），查看拷贝副本。如果仅仅用文本编辑器打开 ufcom.log ，当 ufcom.log 在后台被追加内容时文本编辑器很可能无法探测到文件内容更新。
VCOM 设备正常展现，但应用程序和扫描枪通信不正常	该问题比较复杂，见下一节描述。

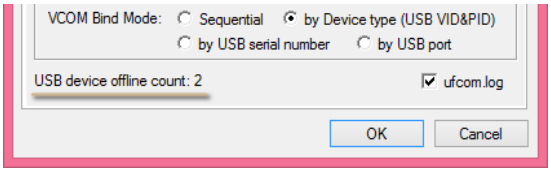
2.3.3. USB 数据传输异常诊断

如果我们遭遇虚拟串口应用程序和 USB 外设数据通信异常，比如，虚拟串口无法读到条码，扫描枪配置程序（EasySet 等）无法探测到扫描枪，EasySet 获取扫描枪图像失败等，在寻求技术支持前，我们可以自己初步诊断故障原因。大致而言，故障可能出现在如下几个环节：

- (A) 扫描枪自身的软硬件。
- (B) USB 数据传输线路。
- (C) PC 端的 USB 硬件和 USB 主控(host controller)驱动程序（由 Windows 提供或 Host Controller 厂商提供）。
- (D) PC 端的 USB client driver （指 UFCOM）。
- (E) 使用 VCOM 的应用程序自身。

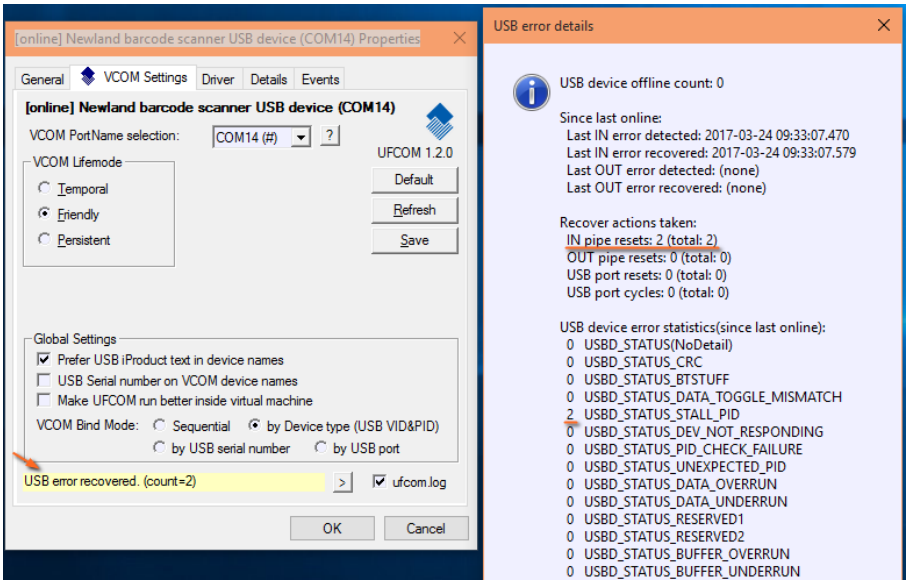
对于普通用户来说，要分辨故障出在哪个环节比较困难。幸运的一点是，UFCOM 可以帮助你直观地判断 B 环节是否有故障。当使用不良的 USB 线缆、有 bug 的 USB hub、USB 端口供电不足、或是 USB 信号受到干扰时，B 环节将暴露出故障。

VCOM Settings UI 的底部状态栏会告知 VCOM 设备对象存在期间发生设备 offline 的次数。当然，我们人为拔除扫描枪将导致 offline 计数增加，但如果我们正常使用过程却发生了 offline 增加的情况，说明 USB 物理信号传输环节很可能有故障。



还有些情况下，USB 传输故障并不表现为 offline ，而是表现为 USB 主控硬件以软件形式向 Windows 报告一个 UsbdStatus 错误，Windows 进而将该错误告知 UFCOM ，UFCOM 会将这种错误通过 VCOM Settings UI 呈现给最终用户。在一个工作良好的系统中，我们不期望看到任何的 UsbdStatus 错误。如果我们发现了这种错误，则暗示 USB 线路存在不稳定的情况。偶发的此类错误可以被软件清除（USB 故障自动恢复），使得最终用户感知不到传输过程被打断，但如果 UsbdStatus 错误发生很频繁，意味着着传输系统非常不可靠，比较大的可能是硬件不稳定导致的，需要进一步排查。

当 UFCOM 检测到 UsbdStatus 错误时，VCOM Settings UI 会将其呈现出来。如下图，UFCOM 报告检测到了 2 个错误，并且已经自动恢复。点击黄色状态条右侧的小箭头，可查详细信息，得知发生了两次 USBD_STATUS_STALL_PID 错误。



注意：在 USB 线路质量很差的情况下，即使 UFCOM 提示错误已经恢复，也不能完全保证 USB 后续通信功能正常。换言之，对于偶发的错误效果才比较好。

提示：VCOM 的 USB 错误统计数字是各个 VCOM 独立的。一个 VCOM 的统计数字在 VCOM 被“拔除”（从设备管理器中消失）后清零。

如果仍旧无法定位故障，请联系我们解决。

3. 高级话题

3.1. 在不同机器间导出与导入 UFCOM 全局设置

UFCOM 的设置有两个层级：

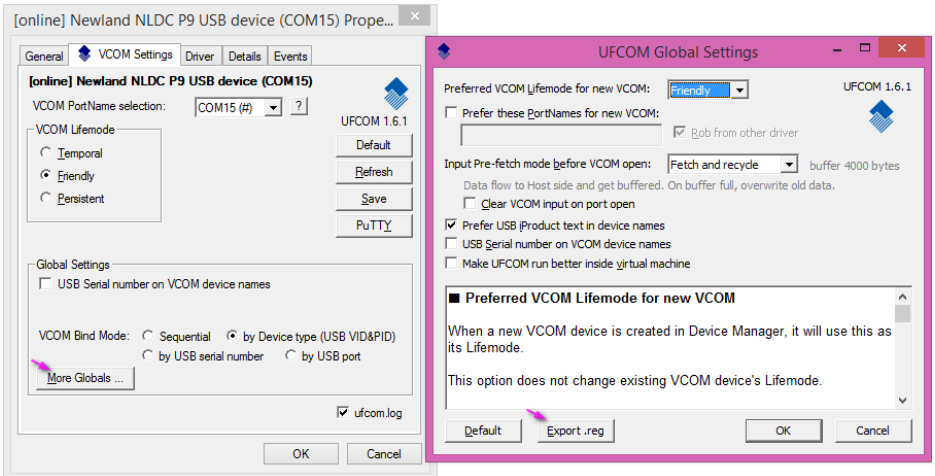
- 全局设置。比如：串口号绑定模式(Bindmode)，Prefetch mode，是否让设备管理器显示 USB 序列号。
- VCOM 特定的设置。比如 VCOM 的串口号(PortName)，生命期模式(Lifemode)。

全局设置的改变不会影响已经生成的 VCOM 设备的特定设置。

UFCOM 目前提供全局设置的导出与导入功能。利用此功能，我们可以将一台样板机器的 UFCOM 全局设置准确地同步到其他机器(Windows PC)。

第一步，我们需要在样板机器上导出当前设置。

在样板机上找到任意一个 VCOM 设备，打开其 VCOM Settings UI，点击 **[More Globals...]**，在 UFCOM Global Settings 对话框，点击 **[Export .reg]** 按钮，生成 ufcom-global.reg 文件，这是 Windows 上标准的注册表文件，该文件包含了样板机上的所有 UFCOM 全局设置。



第二步，在目标机上导入全局设置 。

将 ufcom-global.reg 拷贝到目标机，双击之，Windows 将询问是否要将 .reg 内容导入当前系统。回答是。

第三步，让新设置生效。

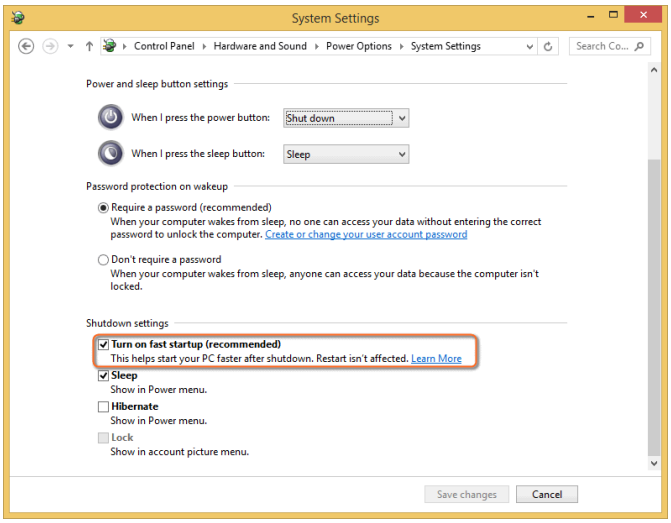
导入设置后，为了确保所有全局设置生效，目标机应该重新装载驱动程序。最简易的做法是重新启动 Windows 。

提示：在 Windows 8/10 上，你不可以用“关机” 再 “开机”来代替“重新启动”。原因是，Windows 8 起微软引入了“快速启动(Fast-startup)”功能（默认开启此功能）。在快速关机启用的情况下，开始菜单执行“关机(shutdown)”，其实质是进行了休眠(hibernate)，休眠的实质是将内存中的信息存入 hiberfil.sys 文件，下次开机时再将 hiberfil.sys 的内容原样恢复到内存中；因此，休眠后再开机无法起到让驱动程序重新初始化的作用。

技巧：在 Fast-startup 已启用的情况下，若希望实现彻底关机，可以使用 shutdown 命令行。

```
shutdown /s /t 1
```

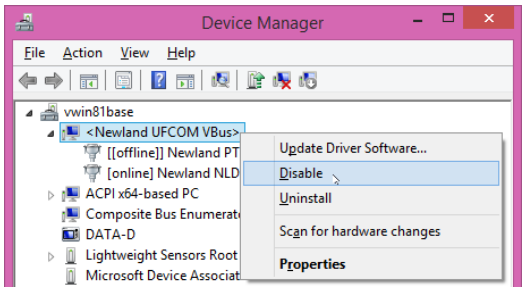
执行前请确保所有的应用程序数据已经保存。

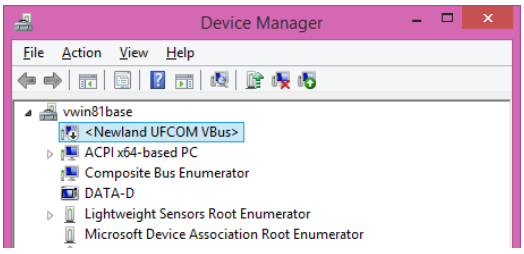


如果你嫌重启 Windows 太费时间，有一个捷径可以不重启 Windows 而达到驱动程序重新装载的效果。操作方法如下：

- 关闭所有使用 UFCOM 虚拟串口的应用程序。
- 拔出所有 UFCOM 掌管的 USB 外设。
- 将 UFCOM Vbus 禁用再启用。

禁用/启用 UFCOM Vbus 的方法如下：打开设备管理器，查看(V)→依连接排序设备(V) ，在顶部附近找到 **<Newland UFCOM VBus>** 节点，禁用之、再启用。





UFCOM Vbus 禁用成功后显示的样子。

注：如果不切换到“依连接排序设备”视图，你也可以在“系统设备”类别下找到 UFCOM VBus 节点。

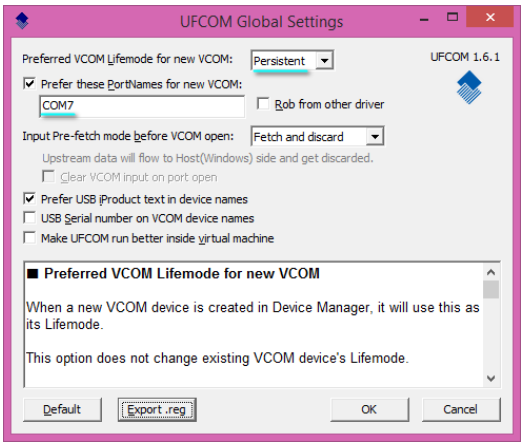
在静默安装的同时导入全局设置

现在举一个例子。假定你希望用静默安装的方式将 UFCOM 部署到多台全新的 Windows 机器上， 同时期望达成以下两个目标：

1. 新机器上插入的第一把扫描枪将自动得到串口号 COM7 。
2. 新生成的 VCOM 采用 Persistent Lifemode 。

由于以上两个要求不是 UFCOM 的默认行为，因此你需要借助全局选项注册表文件。

先在样板机上设定好你期望的全局设置。特别注意：对于要求 1，你需要设置 Prefer these PortNames for new VCOM（简称 GPV Portname list），而非某只具体扫描枪的串口号。



导出 ufcom-global.reg ，将其和 UFCOM msi 安装包一同拷贝到目标机上。在目标机上部署时，应先导入 ufcom-global.reg 再安装 msi ，目标即可达成。

为了达到静默效果，可以使用如下的批处理文件。

```
@REM silent-deploy.bat
@set batdir=%~dp0
@set batdir=%batdir:~0,-1%

reg import "%batdir%\ufcom-global.reg"

start /wait msiexec /q /i "%batdir%\ufcom-1.6.4.msi"
```

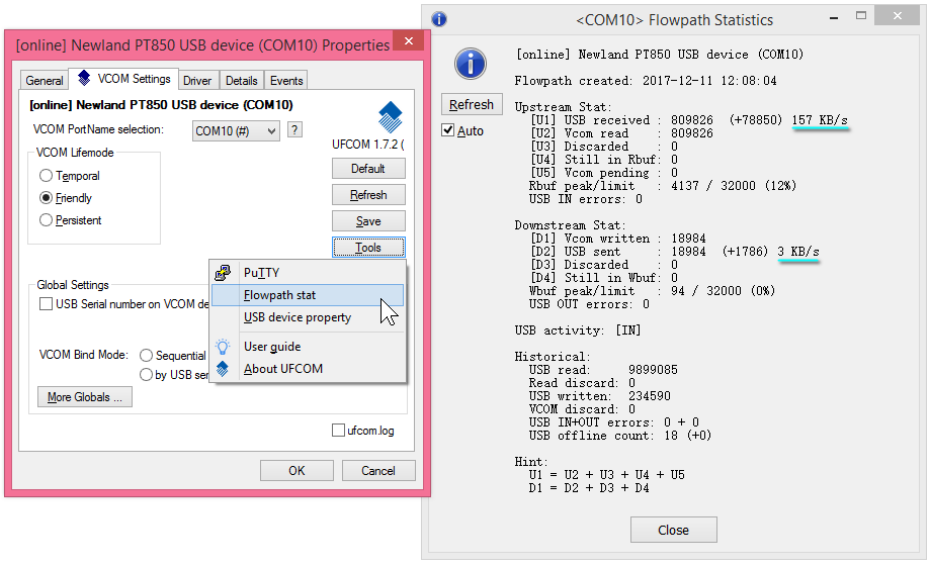
提示：需要以管理员身份来执行以上批处理方可成功。

-

3.2. USB 数据流统计功能 (Flowpath Stat)

Flowpath Stat 能够向我们展示 USB 上行和下行的数据流动统计数据，包括当期会话总字节数、历史总字节数、USB 传输错误数、实时传输速度等。 Flowpath Stat 能够帮助我们诊断应用程序的行为，以及帮助界定 USB 数据传输故障是出在应用程序环节、还是出在驱动程序与硬件环节。

查看入口： VCOM Settings UI → Tools → Flowpath stat 。



各字段解释（Windows PC 视角）：

数据流方向	项目	解释
上行 (Upstream)	U1. USB received	从 USB 线路上收到的字节数（不包括 USB 协议自身的协议字段，下同）。比如，扫描枪扫描一个 13 字节条码，此数值增加 13 。 统计信息每 0.5 秒刷新一次（如果勾起了 Auto）。图中的 (+78850) 表示本次刷新距上次刷新此项目增加了 78850 字节。 末尾的 157KB/s 表示此次刷新计算得出的 USB 上行数据流速为 157 KBytes 每秒。
	U2. Vcom read	被应用程序 ReadFile API 成功读走的字节数。
	U3. Discarded	从 USB 线路收到、但被抛弃的字节数。 比如，Prefetch mode 为 fetch-and-discard 情况下，打开 VCOM 之前 USB 线路发来的字节都将被抛弃，记入此项。 若应用程序关闭 VCOM 时 Rbuf 中仍有数据未被应用层读取，这些数据将被丢弃，记入 U3 。
	U4. Still in Rbuf	从 USB 线路收到、缓存在 UFCOM 上行缓冲区中、尚未被应用层读走的字节数。
	U5. Vcom pending	从 USB 线路收到、且已拷贝到应用层缓冲区中、但 ReadFile 尚未返回给调用者的字节数。 比如，起初 Rbuf 是空的，ReadFile 要求读取 100 字节，读满才算完成，进入等待状态，然后 USB 送来 60 字节，这 60 字节将被记入此项。待 ReadFile 完成后，该 60 字节将被转移到 U2 项目。
	Rbuf peak/limit	Rbuf 的使用峰值用量和 Rbuf 大小。peak 值不会超过 limit 值。 Rbuf 的大小由应用程序 SetupComm 来指定，默认为 32000 。第一次指定后就固定不变了，在 VCOM 关闭前此值将无法被再次设定。
	USB IN errors	USB 上行通道发生过的错误数量。 注：此数值存在并不一定表示应用程序曾经读取了错误的数据。 USB 故障自动机制能够自动修复偶发的 USB 数据错误，应用程序仍旧会读到正确的数据。
下行 (Downstream)	D1. Vcom written	应用层 WriteFile API 报告的“写入成功的字节数”。
	D2. USB sent	USB 线路上成功发送出的字节数。
	D3. Discarded	WriteFile 虽然报告写入成功，但被丢弃的字节数。丢弃意味着这些字节没有被写出 USB 线路。

		有一种情况会导致 discard 字节增加。之前提过， > 当 WriteTotalTimeoutMultiplier=0 , WriteTotalTimeoutConstant=0 , 且未启用输出流控的情况下，UFCOM 保证 WriteFile 不会永久阻塞。 > 若扫描枪被拔除、或下行数据被抑制，WriteFile 会在持续 1 秒无法写出数据后返回，**并告知 WriteFile 成功**，此举是为了模拟向物理串口 WriteFile 不会永久阻塞的效果。 此情况下报告的成功字节数同时将记入 D3 。 若应用程序关闭 VCOM 时 Wbuf 中仍有数据等待发送，这些数据将被丢弃，记入 D3 。
	D4. Still in Wbuf	WriteFile 已经成功、且未被丢弃、也尚未写出 USB 线路的字节数。这些字节等在 UFCOM 的下行缓冲区中随时准备写出 USB 线路。
	Wbuf peak/limit	Wbuf 缓冲区的峰值用量和 Wbuf 大小。peak 值不会超过 limit 值。 Wbuf 的大小由应用程序 SetupComm 来指定，默认为 32000 。第一次指定后就固定不变了，在 VCOM 关闭前此值将无法被再次设定。
	USB OUT errors	USB 下行通道发生过的错误数量。 偶发的下行通道错误也可以被自动恢复。
-	历史总值 (Historical)	以上各项数据的历史总值。即使 Windows 重启，历史总值也会被保留——因为历史统计值记录于注册表中。 USB offline count 处是两个数字，括号外的数字是历史总计掉线次数。括号内的数字是当前 flowpath 对象存续期间检测到的掉线次数，flowpath 被销毁时此数值累积入历史总值。
-	USB activity	此处反应 USB 物理线路上的状态。可能呈现出四种状态： 1. (空) 2. `[IN]` 3. `[IN][OUT]` 4. `[OUT]` 出现 [IN] 表示 USB 线路上此时有 IN 数据包，出现 [OUT] 表示 USB 线路上此时有 OUT 数据包。 VCOM 被程序打开后，持续出现 IN 是正常的，因为 USB Host 随时都在向 USB Slave 查询是否有数据。如果应用程序迟迟不读走上行数据、导致 PC 端 Rbuf 被填满，此时 IN 就会消失，说明 PC 暂时停止了向外设索要数据。 如果应用程序并未持续写出新数据而 OUT 却持续出现，则是不正常的，这意味着下行链路阻塞了，USB 外设没有及时取走 Host 下发的数据。

其他功能：

- 去除 Auto 前的小勾则启用了手动刷新模式，即，点击 Refresh 按钮才会刷新一次。
- 鼠标右键点击左下角或右下角空白处，可得上下文菜单，从中可切换千分号显示，以及拷贝文字到剪贴板。

> **Flowpath 是什么东西？它如何影响 数据流统计值？**
>
> 简要地说，一个插入系统的 USB 物理设备和它对应的 VCOM 构成了一个 flowpath 。
>
> 当 USB 物理设备和 VCOM 设备两者都被拔除时，flowpath 对象被销毁。在一个 flowpath 对象存在的期间，数据流统计值记录于当期会话；flowpath 对象销毁时，当期会话统计值被合并入历史统计值。

-

3.3. msi 安装包生成的注册表项

UFCOM msi 安装后在注册表 HLKM 中创建了如下节点

32-bit Windows	`HKLM\SOFTWARE\Newland Auto-ID\UFCOM`
64-bit Windows	`HKLM\SOFTWARE\Wow6432Node\Newland Auto-ID\UFCOM`

注册表项：

Item name	Item value (sample)	解释
Installed	1	该项目存在表示 UFCOM msi 已经被安装到当前系统上。
Version	1.7.0	本系统上 UFCOM 的版本号。
InstallTargetDir_	C:\Program Files\UFCOM\	告知 UFCOM 被安装在哪个目录中。 注意：目录结尾有个反斜杠。这跟 Appwiz.cpl 报告的 Location 字段一致。

系统集成商可以通过这些注册表项来获知 UFCOM 是否已安装在当前系统中，以此决定是否要主动帮用户升级 UFCOM 。

-

4. VCOM 设备 API 行为说明

UFCOM 较为完整的实现了 Windows 串口 API 的功能。本节告知实现细节。

如果你是 Windows 串口通信程序员，你需要了解本节的信息。

功能类别	实现说明
数据流完整性	对于上行数据流，UFCOM 保证数据的完整性。比如：USB 外设持续送上数据、但应用程序来不及将其读走、使得 UFCOM 自身的上行缓冲区被填满，此时，UFCOM 将暂停向 USB 外设索取数据，而非将 USB 外设送上的数据丢弃。待 UFCOM 上行缓冲区的数据被上层用户取走后，USB 上行数据流恢复流动。 对于下行数据流，分两种情况： • 若应用程序**未启用**下行流控（默认行为），应用程序写出的数据有可能被悄悄丢弃。丢弃行为发生的典型场合是：USB 外设已被拔除，USB 外设未及时读走下行数据（每笔数据默认有 1000 毫秒的时间裕量）。 • 若应用程序明确启用下行流控，则 UFCOM 保证不丢弃下行数据，这同时意味着 WriteFile 有可能会永久阻塞，用户程序得调用 CancelIo 应对此问题。
RS-232 通信参数模拟	不作模拟。用户可设置任意波特率，但设置值被忽略，和扫描枪的通信速度按 USB 线路能够达到的最大速度进行。 数据位、停止位、奇偶校验位的设置被忽略。
Modem 信号线模拟	输出信号：RTS, DTR 的设置被忽略。 输入信号：GetCommModemStatus 总是报告 RING 和 RLSD 为 off 。DSR 和 CTS 信号则进行了模拟。 DSR: 当扫描枪插入时，DSR=On ； 扫描枪拔出后，DSR=Off 。后头“UFCOM 专用 API”章节对于检测扫描枪掉线的方法还有更详细说明。 CTS: 当扫描枪已插入、且下行数据流未受阻时，CTS=On ；扫描枪拔出或下行数据流受阻，CTS=Off 。
硬件流控行为	不支持用模拟的 modem 信号进行上行流控（输入流控），上行流控完全由 USB 线路自身的流控功能决定。 支持下行流控（输出流控）。分两种情况： • VCOM 被应用程序打开后，默认不启用流控，UFCOM 保证 WriteFile 不会永久阻塞。 • 当应用程序以 `DCB.fOutxCtsFlow=1` 调用 SetCommState() 后，下行流控开始起作用，意即， CTS=off 时会使得 WriteFile 阻塞。注意：如果用户设置 `COMMTIMEOUT.WriteTotalTimeoutMultiplier==0` 且 `COMMTIMEOUT.WriteTotalTimeoutConstant==0` ，就可能出现 WriteFile 永久阻塞的情况（若 CTS 一直保持 off），为了让阻塞中的 WriteFile 完成，用户可以用 CancelIo 来取消之。
软件流控行为	不支持 XON/XOFF 方式的软件流控。

<table>
<tr><th>功能类别</th><th>实现说明</th></tr>
<tr><td>WriteFile 行为</td><td>UFCOM 保证 WriteFile 的原子写入行为。即，对于应用层请求写出的数据，WriteFile 或者全部接受，或者一个字节也不接受。</td></tr>
<tr><td>串口超时</td><td>完整实现了 COMMTIMEOUTS 超时参数的语义。针对 USB 设备的特点有一些特殊的设计：
当 WriteTotalTimeoutMultiplier=0 , WriteTotalTimeoutConstant=0 , 且未启用输出流控的情况下，UFCOM 保证 WriteFile 不会永久阻塞。若扫描枪被拔除、或下行数据被抑制，WriteFile 会在持续 1 秒无法写出数据后返回，并告知 WriteFile 成功，此举是为了模拟向物理串口 WriteFile 不会永久阻塞的效果。
当 WriteTotalTimeoutMultiplier=0 , WriteTotalTimeoutConstant=0 , 且启用输出流控的情况下，将所有数据写出前，WriteFile 会永久阻塞。当然，用户可以用 CancelIo 来取消 WriteFile ， CancelIo 后 WriteFile 会立即完成。
当 WriteTotalTimeoutMultiplier 或 WriteTotalTimeoutConstant 任意一者非 0 时，不管是否启用下行流控，用户指定的超时将生效。</td></tr>
<tr><td>串口事件等待</td><td>实现了如下四种事件的通知：<table><tr><th>事件</th><th>触发条件</th></tr><tr><td>EV_RXCHAR</td><td>当输入缓冲区中出现字符时，发生该事件。当应用程序已经得到一次 EV_RXCHAR 事件通知的情况下，若没有新的字符收到，应用程序再次查询将不再得到通知；如果之后收到了新的字符，则必定会再次产生 EV_RXCHAR 通知（不论原先输入缓冲区是否为空）。</td></tr><tr><td>EV_TXEMPTY</td><td>当输出缓冲区变空时，发生该事件，是边缘触发的，意即，当应用程序得到一次 EV_TXEMPTY 通知后，立即再次等待 EV_TXEMPTY 事件，将不会再次得到通知，要经历过一次输出缓冲区填入内容（WriteFile）再排空的过程，才会再次通知 EV_TXEMPTY 。</td></tr><tr><td>EV_DSR</td><td>EV_DSR 是个边缘触发事件，反映 DSR 信号的变化。</td></tr><tr><td>EV_CTS</td><td>EV_CTS 也是个边缘触发事件，反映 CTS 信号的变化。</td></tr></table>另，UFCOM 允许用户使用两个线程进行操作，一个线程调用 WaitCommEvent ， 另一个线程进行 ReadFile , WriteFile 读写数据。

UFCOM 作者认为： 将 EV_RXCHAR 和 EV_TXEMPTY 设计成边缘触发（而非电平触发）不是个好主意，但由于微软针对物理串口 COM1 的 Serial.sys 是如此设计的，UFCOM 只好迁就该行为以便达到尽可能好的应用程序兼容性。</td></tr>
<tr><td>CancelIo</td><td>完全支持。用户可用 CancelIo 取消进行中的 ReadFile ， WriteFile ， WaitCommEvent 。UFCOM 保证 CancelIo 将立即成功，不会阻塞。

特别提示：针对 ReadFile 发起 CancelIo 时，若 VCOM 已经收取了部分的字节，ReadFile 将报告成功，用户能取到已收取的字节。此行为比 Windows 自带的 serial.sys 和 usbser.sys 优秀；后者对于 CancelIo 的情况，只会报告结果为 995(ERROR_OPERATION_ABORTED)。</td></tr>
<tr><td>读写缓冲区大小</td><td>UFCOM 内部针对每个 VCOM 开辟的读写缓冲区默认各为 32000 字节。这意味着，应用层的一次 WriteFile 操作允许的最大字节数是 32000 ，超过此数据量将得到 ERROR_INVALID_PARAMETER(87) 错误 。处理此问题的方法有三种：
修改应用程序代码，将一块大数据切割成小块分次发送。
修改应用程序代码，CreateFile 打开串口后立即调用 SetupComm 指定一个大的写缓冲区。
若无法修改应用程序代码，你可以设置 UFCOM 全局注册表项 TxQueueDefaultSize 来得到一个大的写缓冲区。修改此注册表项后需要重启驱动程序方可生效。
比如，要将 TxQueueDefaultSize 修改为 1024000 字节，可导入如下注册表文件：

<pre>Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Nls_ufcom\Parameters]
"TxQueueDefaultSize"=dword:000fa000</pre>相应的，修改 TxQueueDefaultSize 将影响默认的 RX 缓冲区大小。</td></tr>
</table>

【 针对 Temporal Lifemode 的特别说明 】

对于 Temporal Lifemode 的 VCOM，对应的 USB 物理设备拔除将使得 VCOM 句柄处于失效状态，你用失效句柄再次进行 API 调用，ReadFile、WriteFile 、WaitCommEvent 等，都将立即返回失败，即使重新插回扫描枪，失效句柄仍旧保持失效。此时你能做的只有关闭失效句柄，再调用 CreateFile 重新获得一个 VCOM 句柄。

5. UFCOM 专用 API 指南

借助 UFCOM 提供的专用 API 功能，应用程序能够显著提升用户体验。

Windows 的串口 API 向应用程序提供了一套抽象的数据收发模型，这套模型在很多场合下工作得很好。不过，由于这套 API 接口的定义出现于 USB 规范之前、针对的是物理串口，它并没有考虑到 USB 设备热插拔的特性，没有为热插拔事件定义 API 语义，因此，UFCOM 需要对其作出补充规定。

5.1. 用 vcomtest 测试 UFCOM API 行为

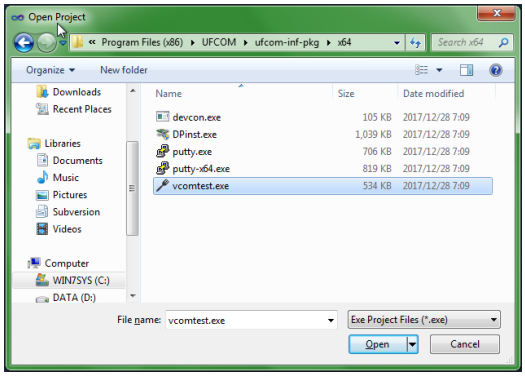
UFCOM 实现了绝大部分的 Windows 串口 API 语义，我们提供 vcomtest 测试程序用于验证这些语义，以及演示 UFCOM 提供的专有功能。我们提供编译好的 vcomtest.exe 以及其主体源代码和 pdb，用户可以在 Visual Studio 调试器中进行源码级调试，以理解它的工作逻辑。不过，你无法自行编译 vcomtest 程序，因为有部分头文件和库我们未提供。

为了对 vcomtest 进行源码调试，你需要 Visual Studio 2005 或更高版本。

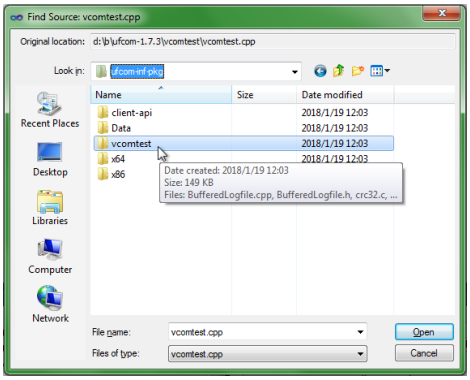
下面以 UFCOM 1.7.3 和 Visual Studio 2010 为例简要讲述操作方法。 用 $UFCOM 代表 UFCOM 的安装目标目录。

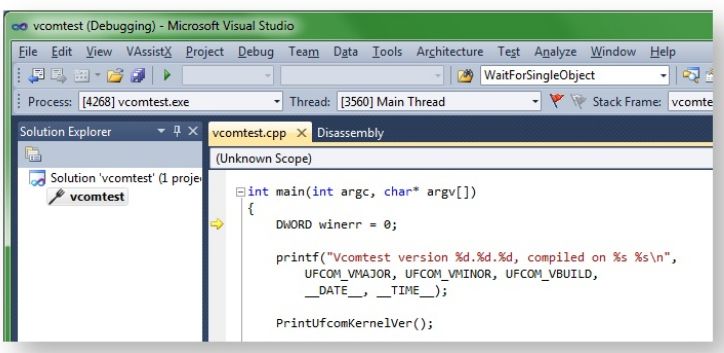
打开 Visual Studio IDE，File → Open → Project/Solution ，要求打开 `$UFCOM\ufcom-inf-pkg\x64\vcomtest.exe` （在 32-bit Windows 上运行，你应该打开 x86 子目录中的 vcomtest）。提示：此处是将 vcomtest.exe 作为工程打开，而非作为普通文件打开。 VS IDE 主菜单执行 Debug → Step Over (F10) 开始调试。此时会有一个提权对话框问是否要提权运行，回答否即可。接着，会有一个 Find Source 对话框问我们 vcomtest.cpp 在何处，将 `$UFCOM\ufcom-inf-pkg\vcomtest` 目录告诉它。如果一切顺利，应该会看到调试器的当前指令（黄色箭头）停在 main() 的第一条语句，说明可以开始源码调试了。

Open vcomtest.exe as a project.



Locate directory of vcomtest.cpp





5.2. 检测 USB 设备的插入、拔除状态

设备的插入状态对应 VCOM 设备在设备管理器中呈现的 online 状态，拔除状态对应设备管理器中呈现的 offline 状态。

Windows API 本身并未定义如何表达设备的 online/offline 状态，因此，很多提供虚拟串口的设备驱动程序（不局限于 UFCOM）通过将设备句柄变为无效来指示设备已被拔除。此处的句柄指 CreateFile 返回的设备句柄，无效的含义是，对句柄进行 ReadFile、WriteFile、WaitCommEvent 将立即返回失败，处于 pending 状态的 overlapped I/O 也会因为句柄突然变成无效而立即完成，这是很有效的“设备拔除”指示方法。回到 UFCOM 上，对于 Temporal lifemode 的 VCOM ，也遵循如此行为。

UFCOM 的特别之处在于， Friendly lifemode 和 Persistent lifemode VCOM 具有连接保持功能，设备拔除并不会导致句柄失效，这给应用程序带来很多便利性，但应用程序如果想在 online/offline 发生变化时采取一些特殊行动（比如在用户界面上提示设备掉线），它该如何得知这个变化呢？

UFCOM 提供的方法是，用 DSR 来指示设备的在线状态， DSR on 对应 online ， DSR off 对应 offline 。DSR 原意是 RS-232 通信协议的一根物理信号线，DSR 的状态由通信的对方来控制。USB 物理上不存在 DSR 信号线，因此 UFCOM 借用 DSR 来表达 online/offline 状态。 用户代码可以通过 WaitCommEvent 来监视 EV_DSR 事件来第一时间获知 DSR 的变化。

请特别注意：EV_DSR 事件仅仅是通知 DSR 状态发生了变化，至于当前变成了 on 还是 off，需要接着调用 GetCommModemStatus 来获取。 举个例子，用户代码先后调用两次 WaitCommEvent 来等待 EV_DSR ，期间顺次发生了如下事件：

1. 第一次 WaitCommEvent 因为 DSR 变为 off 而返回。
2. 在很短的时间内，DSR 变为 on 后立即又变回 off 。
3. 用户第二次调用 WaitCommEvent。

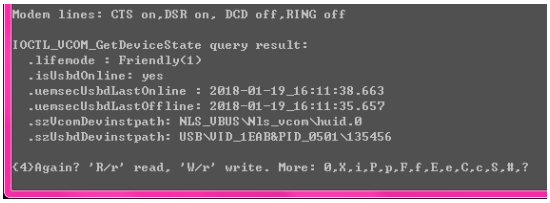
问：第二次的 WaitCommEvent 是否会立即报告发生了 EV_DSR 呢？

答案是会，因为自前一次“发出 EV_DSR 通告”以来，DSR 又发生了变化（不论发生了几次）。“发出 EV_DSR 通告”，指的是 WaitCommEvent 将 EV_DSR 返回给用户的那个时间点。换言之，不管 DSR 变化多少次、变化多频繁，只要用户代码在得到 DSR 变化通告后用 GetCommModemStatus 获取 DSR 的当前状态，一定能正确跟踪 DSR 的最终状态。

> vcomtest 实验指引： vcomtest 打开 Friendly lifemode 或 Persistent lifemode VCOM 后，键盘按 **E** 开启一个后台线程，该线程在循环执行 WaitCommEvent，此时反复拔除插入扫描枪，可看到 EV_DSR 事件报告。按 **e** 可停止 WaitCommEvent 线程。

在前面的例子说明，连续两次 WaitCommEvent 通知之后，GetCommModemStatus 查得的 DSR 有可能是相同的值。如果你担心这是虚假通知的话，有一个更明确的方法可以让你确信 online/offline 状态确实发生过变化。 UFCOM 提供 **`IOCTL_VCOM_GetDeviceState`** 控制码来明确告知 USB 设备当前的 online/offline 状态，以及最近一次 offline 与 online 发生的时间。

```
IOCTL_VCOM_GetDeviceState_ost vcomstate = {};
DWORD retbytes = 0, insize = sizeof(vcomstate);
BOOL b = DeviceIoControl(hCom, IOCTL_VCOM_GetDeviceState, NULL, 0,
    &vcomstate, insize, &retbytes, NULL);
// check vcomstate members for details.
```

> vcomtest 实验指引：vcomtest 打开串口后，按 `i` 查询 VCOM 的当前状态，输出信息中包括了 **`IOCTL_VCOM_GetDeviceState`** 返回的信息。
>
> 比如，扫描枪拔除再插入，按 `i` 查询，可看到 uemsecUsbdLastOnline 时间比 uemsecUsbdLastOffline 要晚。
>
> 
>

-

5.3. 使用 one-time Temporal 特性来及时得到 USB 设备拔除通知

背景信息：UFCOM 的 Friendly lifemode 给用户带来串口句柄保持功能，这在大多数场合下给最终用户带来了很好的使用体验。不过，在一些特殊的场合下，应用程序代码（而非最终用户）希望在 USB 设备被拔除时（VCOM 变为 offline 状态）能够立即得到通知，比如，给 USB 设备更新固件的过程中，USB 设备需要经历一个重启的过程，设备重启间隙就是 VCOM 就呈现为 offline 状态，固件更新程序希望及时捕获该状态并执行特定动作。可以设想以下两种解决方法：

- 方法一，将 VCOM 设为 Temporal lifemode 能够达成该目的，即，设备拔除时 VCOM 句柄瞬间变为失效、此 VCOM 句柄的后续读写都将立即失败。然而，这需要用户在进行固件更新操作前手工切换 lifemode 为 Temporal，固件更新完毕后再切换回 来，很不方便。
- 方法二，借助 WaitCommEvent 来等待 EV_DSR 的变化，以此获知 offline 事件的发生。此举的麻烦之处是：若原先的代码结构是 WriteFile/ReadFile 构成的循环，现在得修改代码加入 WaitCommEvent，这很可能需要较大的代码修改量。

从 UFCOM 1.7.3 起，应用程序还有一个更好的方法。应用程序向 VCOM 句柄发送 **`IOCTL_VCOM_OneTimeTemporalLifemode_Enable`** 控制码，让当前 VCOM 进入 one-time Temporal 状态，此后，一旦 USB 设备变为 offline，VCOM 句柄就会立即变为失效，如同 Temporal lifemode 得到的那般。

启用 one-time Temporal 的代码写法为：

```
HANDLE hcom = CreateFile("\\\\.\\COM3", ...);

DWORD retbytes = 0;
BOOL bSucc = DeviceIoControl(hcom, IOCTL_VCOM_OneTimeTemporalLifemode_Enable,
    NULL, 0, // input
    NULL, 0, // output
    &retbytes, NULL);

if(bSucc)
    printf("Success.\n");
else
{
    DWORD winerr = GetLastError(); // ERROR_INVALID_FUNCTION if UFCOM version too old
    printf("Fail.\n");
}
```

另有： IOCTL_VCOM_OneTimeTemporalLifemode_Disable 禁用 one-time Temporal 模式，
IOCTL_VCOM_OneTimeTemporalLifemode_Query 查询当前 one-time Temporal 模式。具体写法见 vcomtest 源码。

一些提示：

- one-time Temporal 设置值仅针对当前打开句柄生效。换言之，若关闭 VCOM 句柄再重新打开，one-time Temporal 将自动恢复成禁用状态。
- 若发送 IOCTL_VCOM_OneTimeTemporalLifemode_Enable 时，VCOM 对应的 USB 设备已经处于 offline 状态，则 VCOM 句柄立即变为失效。
- USB 设备重新插回，原先 lifemode 是 Friendly 的，当前 lifemode 将仍旧是 Friendly 。这意味着，one-time Temporal 操作不会造成持久的影响，这正是我们期望的。

One-time Temporal 功能同样适用于 Persistent lifemode 。

vcomtest 实验指引：vcomtest 打开串口后，

- 按 < 执行 IOCTL_VCOM_OneTimeTemporalLifemode_Enable
- 按 > 执行 IOCTL_VCOM_OneTimeTemporalLifemode_Disable
- 按 | 执行 IOCTL_VCOM_OneTimeTemporalLifemode_Query

对于一个 Friendly lifemode VCOM 执行 IOCTL_VCOM_OneTimeTemporalLifemode_Enable 后，

- 先拔除设备，再按 **r** 执行一次 ReadFile ，将立即报告 ERROR_BAD_COMMAND。
- 若是先执行 ReadFile 等待数据再拔除设备，则拔除设备的瞬间 ReadFile 将立即报告完成。完成的结果依拔除设备前是否已接收到部分数据而定。若未收到任何数据，将报告错误 ERROR_DEVICE_NOT_CONNECTED；若已收到部分数据，将报告成功，用户可取回那些数据，再次 ReadFile 将立即报告失败——如果缓存的数据已经被前几次 ReadFile 读干净。
- 等待中的 WaitCommEvent 也将因为设备的拔除而立即报告 EV_CTS, EV_DSR 和 EX_RXCHAR 变化，再次调用 WaitCommEvent 将立即报告失败。

提示：设备拔除导致的错误码有可能是 ERROR_DEVICE_NOT_CONNECTED 和 ERROR_BAD_COMMAND 之外的其他值，对于事先无法预料的返回值，用户都应该将它们认为是句柄失效。

-

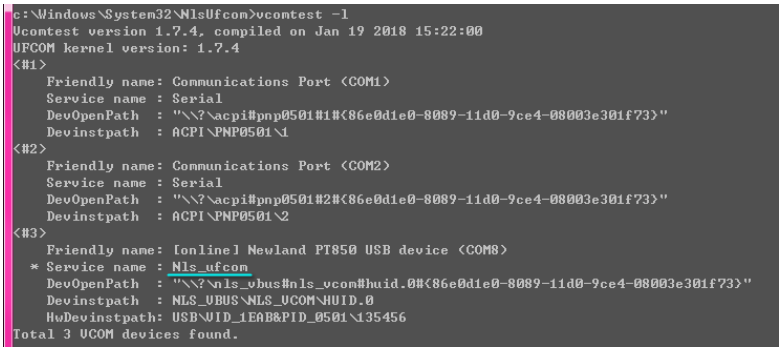
5.4. 如何判断系统中的串口设备是否为 UFCOM 虚拟串口？

有两种方法。

【简易方法】 先打开串口设备获取句柄，发送 IOCTL_VCOM_OneTimeTemporalLifemode_Query（0x16127408），如果 DeviceIoControl 返回成功，则认为该串口设备是 UFCOM 提供的。

【复杂方法】 该方法可以在打开串口设备之前就获知某个串口设备是否由 UFCOM 提供。为了使用该方法，你必须使用 SetupDi... 函数枚举出系统中的所有串口设备，查询它们的 SPDRP_SERVICE 属性，若属性值是 Nls_ufcom ，则是 UFCOM 虚拟串口。

vcomtest 实验指引：用 `vcomtest -l` 可枚举出系统中的串口设备，并得知它们的 SPDRP_SERVICE 属性。



6. UFCOM 版本历史

版本号	发布日期	新功能与修正
1.7.9	2018.04	修正 1.7.6 引入的一个崩溃问题：插着扫描枪进行 msi 安装 / 卸载，有很大概率导致系统崩溃。 建议：从 1.7.6 升级时，请先拔除扫描枪，可防止升级过程发生崩溃。升级过程内部经历过一个卸载旧版本再安装新版本的动作。
1.7.6	2018.02	UFCOM 现通过 Windows 10/8.1/7 三个平台的兼容性认证。认证报告可从微软网站在线获取。
1.7.5	2018.01	针对 VCOM 设备提供 IOCTL_VCOM_GetDeviceState ，用于获取设备状态信息。 修复 Datapipe API 兼容性行为，udp_op.dll 导出函数 get_udp_state_ex() 现可以获知设备拔除状态。全局设置提供三种 Datapipe lifemode 供选择，"On device present" 提供的 udp_op.dll 语义最接近于老式 Datapipe 驱动。
1.7.3	2017.12	应用程序可以针对 VCOM 设备句柄启用 one-time Temporal 特性，以便第一时间得到 USB 设备拔除通知。
1.7.2	2017.11	修正 bug：在 Windows 7+ 上，插着扫描枪进行 UFCOM msi 升级，升级后 VCOM 的串口号会发生变化。从 1.7.2 起往高版本升级，将不再遭遇此问题。 修正 bug：在 Windows XP 上，对 UFCOM msi 进行卸载、重装，必定导致驱动瘫痪，VCOM 无法使用，得手工 Uninstall 所有 UFCOM 设备节点才能保证恢复 。 修正 bug：扫描枪在非常短的时间内拔除再插入，可能导致 VCOM 设备节点丢失（在设备管理器中看不到 VCOM 设备），用户得再次拔插扫描枪来恢复。
1.7.0	2017.10	USB 设备启动失败信息可视化。 提供数据流统计功能(Flowpath Stat)，用户可查看 VCOM 的数据流量和流速。 VCOM Settings UI，PuTTY 按钮改为 Tools 折叠菜单，提供多项工具的入口。 VCOM Settings UI 启用现代视觉风格，和 Windows 系统视觉风格匹配。 UFCOM 安装目标目录与版本号记录于注册表，系统集成商能够以此得知 UFCOM 是否已经被安装。

版本号	发布日期	新功能与修正
1.6.4	2017.08	普通用户(非管理员)现在可以从设备管理器中查看 VCOM settings 各个设置项，但不允许修改它们。 PurgeComm 行为修正、以及多个重要的稳定性修正。 我们提供了 UFCOM 配套测试程序 vcomtest.exe 的源代码和 pdb，用户可以在 Visual Studio 2005+ 中调试该源代码，此举有助于理解 Windows 串口 API 的行为。
1.6.1	2017.07	提供四种 Prefetch mode，决定 VCOM 被应用程序打开前的数据流行为。默认 Prefetch mode 是 Fetch and discard，此行为跟微软自带 usbser.sys 一致。UFCOM 之前版本之提供 No fetch 模式，跟 usbser.sys 行为不一致，会给部分应用程序带来困扰。 对于 VCOM 特定的选项，提供全局偏好值选项(GPV)。借助 GPV，用户可以在驱动包安装时预先指定针对未来 VCOM 的偏好设定值，比如预先执行新生成 VCOM 的串口号。 大多数全局选项放到专门的 UFCOM Global Settings 对话框中，提供 Export .reg 功能快速导出全局设置。 Msi 安装包显示 EULA 信息。
1.5.0	2017.06	UFCOM 现可配合 mscomm32.ocx 控件库工作，该控件库经常由 Visual Basic 6 程序使用。
1.4.0	2017.05	UFCOM 通过 Windows 10 WHQL 认证。UFCOM 安装包以 msi 形式提供，可以被静默安装。 修正 EV_RXCHAR 语义，使其和物理串口(serial.sys)一致。
1.3.5	2017.04	VCOM Settings UI 中集成 PuTTY，方便进行串口收发数据测试。
1.2.0	2017.03	USB 故障自动检测和恢复。
1.1.0	2017.03	USB device 设备属性对话框中可开启 ufcom.log 。 支持 Datapipe API 。
1.0.7	2017.02	首个版本。