

## 1 概述

KVM 即键盘(Keyboard)、显示器(Video)、鼠标(Mouse)的缩写组合[1].由于网络延时或被控机分辨率不在 KVM 可控范围内等因素,传统 KVM 在鼠标同步方面存在被控机鼠标与主机鼠标位置偏差过大、鼠标操作延缓甚至无法响应等问题。故提高鼠标同步的精确性和时效性,成为 KVM 技术发展的一项重要内容。目前支持设备通用连接并具有高传输速率的 USB 接口已成为外设连接 PC 主机的主流方式。

USB2.0 接口标准在原有的12 Mb/s 和1.5 Mb/s 传输速率基础上加入480 Mb/s 的高速支持,使单位时间内能够传输和处理更多的事务数据。键盘、鼠标等支持人与计算机交互的设备归为人接口设备(Human Interface Device, HID)类,是最为广泛使用的 USB 设备,并已得到 Windows/Linux 等操作系统内置驱动程序的良好支持。本文依据 HID 类规范协议,通过高速 USB2.0 的硬件支持,向被控机发送信令数据模拟鼠标操作。

## 2 鼠标同步方法

### 2.1 实现原理

KVM 核心技术是通过键盘、鼠标、显示器的适当配置,对 KVM 切换器的多台远程被控机进行管理,实现用一套 I/O 外设去访问和操作多台被控机。图1 为其框架示意图。

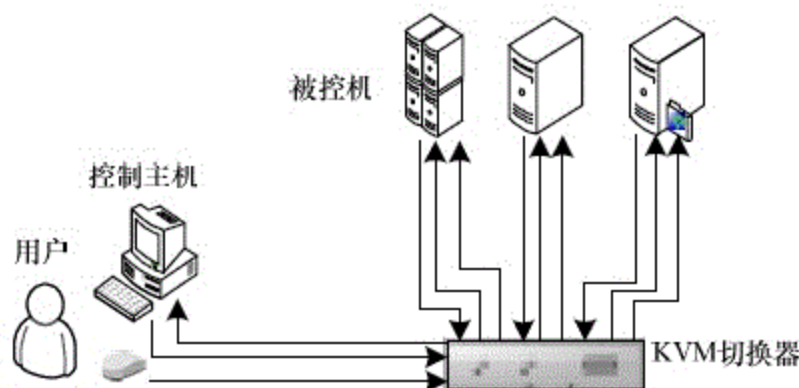


图1 KVM 框架示意图

HID 类设备的信息以描述符形式存储在设备 ROM 中,通过设置、检测相应描述符结构完成设备与主机之间信令、数据的传输和获取。HID 类设备的具体应用数据如键盘键值、鼠标指针值等主要用报告描述符进行描述。报告描述符由多片规则条目的信息由片段组成,可由 HID 类规范自定义数据表达类型。主要条目第3 位数值(Bit2{Absolute(0)|Relative(1)})标识设备操作使用是绝对模式还是相对模式,从而在支持鼠标同步方法中,

有绝对同步和相对同步这2 种方式。鼠标数据内容包括鼠标的按键状态、坐标位置和滚轮滚动值等。

传输方式根据鼠标应用数据量较小、不定时发生、延迟受限等特点，采用对时间有严格限制的中断传输方式。同时，高速 USB2.0 支持单个事务可传送最大数据包的容量达 1 024 Byte,轮询事务的时间间隔仅为125 Fs,支持传输速率可达到24.5 Mb/s.

2.2 相对鼠标同步

相对鼠标同步是根据计算前后2 次鼠标坐标的差值，对当前鼠标位置进行偏移。由于条目以1 Byte 为单位，因此传输单字节坐标偏移的相对同步方式最早应用于 KVM 的鼠标同步。相对鼠标同步的有效数据区定义如表1 所示。

| 同步方式 | 按键状态 | 水平<br>相对偏移 | 竖直<br>相对偏移 | 滚轮滚动格数 |
|------|------|------------|------------|--------|
| 1    | 1    | 1          | 1          | 1      |

表1 相对鼠标同步数据区定义

由字节的低3 位，即0 bit~2 bit 分别表示滚轮、右键和左键按下或弹起的状态，即按键状态。

相对偏移差值数值范围为-127~127,当差值在水平方向(X 轴)或竖直方向(Y 轴)大于 127 时，需要分多次进行移动，即循环多次发送差值数据。滚轮以±1表示向上或向下滚动一格，并可累计滚动格数，以支持更多页面滚动。本文用汇编语言定义相对鼠标同步的报告描述符内容如表2 所示。

| 汇编码           | 对应报告描述表内容                        | 说明       |
|---------------|----------------------------------|----------|
| db 05 h, 01 h | ;Usage Page(Generic Desktop)     | 描述符起始    |
| db 09 h, 30 h | ;Usage(X)                        | X轴方向变量值  |
| db 09 h, 31 h | ;Usage(Y)                        | Y轴方向变量值  |
| db 09 h, 38 h | ;Usage(Mid-Roll)                 | 滚轮变量值    |
| db 15 h, 81 h | ;Logical Minimum(-127)           | 最小值为-127 |
| db 25 h, 7 fh | ;Logical Maximum(127)            | 最大值为 127 |
| db 75 h, 08 h | ;Report Size(8)                  | 描述符大小    |
| db 95 h, 03 h | ;Report Count(3)                 | 描述的变量数目  |
| db 81 h, 06 h | ;Input(Data, Variable, Relative) | 相对同步方式   |

表2 相对鼠标同步的报告描述符

通过界面获得的鼠标坐标值以一个像素为单位，较实际位置有一定的精度损失。且相对鼠标同步每次的偏移值均以上次坐标为基准，数值损失将不断累加。针对此问题，本文在相对同步方式中增加了自适应的误差弥补，对累积误差进行检测和处理，即“残差处理”。该处理方法主要采用坐标值精度转换、累计残差、足1 补齐的方式，具体实现方法如下所述：

(1)坐标值精度转换

采用 short 型双字节数值表示法，以屏幕左上角为原点，且无论何种分辨率均定义右下角坐标为(32 767, 32 767)，将界面获取的坐标根据屏幕分辨率按比例进行转换。同时记录本次的双字节绝对坐标位置作为下次偏移的基准，减少以偏移差值为基准而引入的累积误差。

(2)残差累计及补偿

使用 double 型变量累计每次坐标值转为 short 整型时丢弃的小数值。当累计值大于1 时，在当前坐标差值上补1 再进行发送。同时定时检测误差累计值大于1 的次数频率，当超过一定程度时，采取自动重新同步。即先进行13 次(7127,7127)坐标偏移，将鼠标移动到屏幕左上角，然后再用一次至多次偏移，将鼠标定位到控制主机记录的最后坐标位置。

然后采用13 次坐标偏移，若以1 920×1 080 分辨率屏幕为上界，则在当前主流显示器中，均可以将鼠标移至屏幕左上角。

2.3 绝对鼠标同步

绝对鼠标同步在经过直接传输换算后，其双字节绝对坐标值的每次同步，均需要以原点为基准重新定位鼠标，以去除与前一次鼠标位置的关联，由此避免了相对偏移造成的误差累积。其有效同步数据区定义如表3 所示。

| 同步方式 | 按键状态 | 水平<br>相对偏移 | 竖直<br>相对偏移 | 滚轮滚动格数 |
|------|------|------------|------------|--------|
| 1    | 1    | 2          | 2          | 2      |

表3 绝对鼠标同步的数据区定义

绝对鼠标同步具有更精确的同步效果，并且需要占用报告描述符条目的2 Byte 空间来表示坐标值(滚轮值)，故必须得到操作系统内置 HID 类驱动程序的支持。现代 Windows 系列(XP/Win7/Vista 等)、Mac OS 10.5 以及 Linux 2.6 版本等操作系统均可支持绝对鼠标模式，OS/2 系统也于2010 年5 月发布 xsmouse00.zip 提供支持该模式的鼠标驱动程序。