

# 通用串行总线设备分类定义

## 对于视频设备：常见问题

( 常问问题 )

修订版1.1 2005年6

月1日，

## 贡献者

阿卜杜勒·伊斯梅尔R.	英特尔公司
田边昭洋	佳能公司
阿利森·希克斯	德州仪器 ( TI )
阿南德Ganesh神	微软公司
安迪·霍奇森	意法半导体
Anshuman Saxena先生	德州仪器 ( TI )
李贝特朗	微软公司
嫦娥李	凌阳科技有限公司
大卫·瘦瘤	微软公司
埃里克Luttmann	赛普拉斯半导体公司
费尔南多·乌尔维纳	苹果电脑公司
海尔特克纳彭	飞利浦电子
热罗Mudry	罗技公司
小林弘	微软公司
让 - 米歇尔·查尔顿	罗技公司
杰夫·朱	微软公司
肯一郎Ayaki	富士
三夫仁井田	佳能公司
朽木信夫	三洋电机有限公司
奥利维尔Lechenne	罗技公司
保罗·萨克尔	意法半导体
雷米齐默尔曼	罗技公司
真一Hatae	佳能公司
史蒂夫·米勒	意法半导体
Tachio小野	佳能公司
佐藤隆	飞利浦半导体
平田洋一	松下电器产业有限公司

©2001，2002，2003，2004，2005年USB实施者论坛  
版权所有。

#### 知识产权免责声明

本文所提供之“为是”无担保概包括对任何特定目的的适销的担保，健身，或任何其它担保由此产生的任何建议，规范或范例。

许可证是此授予复制和分发本文件仅供内部使用。任何其他许可，明示或暗示，诉讼或其他方式向任何其他知识产权的授权或此旨在。本文档的作者不承担任何责任，包括法律责任侵权的专利权，涉及执行这一文献信息。本文档的作者也不保证或表示这样的实现方式 ( S ) 将不会侵犯这些权利。

所有产品名称均为商标，注册商标，或者其各自所有者的服务标志。

## 修订历史

	日期	描述
1.0		初始版本
1.0A		添加： 2.21隔行扫描视频
1.0B		添加： 2.22净荷报头中的PTS与STC字段的最大值  2.23协议延迟行为 2.24当前和未来的净荷报头格式和扩展
1.0C	2004年6月5日，	添加： 2.25 Motion JPEG格式特性
1.1	6月1日 <sup>ST</sup> ，2005年	新增MPEG2 APT的信息。加入的终端机关联信息 ( RR0052 )。 。新增向前和向后兼容性指南 ( RR0055 )。  增加了对基于流的有效载荷支持 ( MPEG4的SL，VC1，H264 ) 依靠MPEG2系统规范 ( RR0061 )。  新增静止图像的1.30主机行为捕获方法2. ( RR0068 )。应用RR0064。更改VDC以在UVC 术语和缩写。更新部分2.1更新的文件集。

## 目录

1	介绍 .....	1
1.1	目的 .....	1
1.2	范围 .....	1
1.3	相关文档 .....	1
1.4	术语和缩写 .....	1。
2	个常见问题 ( FAQ ) .....	1
2.1	设置的USB视频类文件 .....	1
2.2	主机USB带宽管理支持 .....	4
2.2.1	的Microsoft Windows .....	4
2.2.2	苹果Mac OS X .....	4
2.3	设备备用接口集 .....	5
2.4	USB视频类和HID接口 .....	6
2.5	USB视频类范围 .....	7 ....
2.6	批量与同步传输 .....	8
2.6.1	批量和同步传输的特点 .....	8
2.6.2	散装管道传输 .....	9 ...
	批量传输的2.6.3 DATA PID订货 .....	10
2.7	音频和视频流同步 .....	10
2.8	支持流媒体加密的视频内容通过USB .....	12
2.9	布局一个GUID数据结构的 .....	13
2.10	主机输入和输出数据的处理 .....	13
2.10.1	图表术语 .....	13
2.10.2	输入数据状态图基于帧的视频格式 .....	15
2.10.3	输入数据状态图基于数据流的视频格式 .....	16
2.10.4	输出数据状态图基于帧和基于流的视频Formats .....	17
2.11	帧间隔和帧速率的关系 .....	18
2.12	时钟恢复机制 .....	18
2.13	MaxPayloadTransferSize选择 .....	22
2.14	负载格式向前兼容性 .....	22
2.15	设备/主机处理分区 .....	22
2.16	电源模式控制 .....	23
2.17	探头和提交作业 .....	24
2.18	输入和输出端子协会 .....	27
2.19	多个配色描述符 .....	27
2.20	请求错误代码控制 .....	28
2.21	等时管道和数据可用性 .....	28
2.22	隔行扫描视频 .....	28
2.23	净荷报头中的PTS与STC字段的最大值 .....	29
2.24	协议延迟行为 .....	30
	与设置阶段误差2.24.1 SET_XXX请求 .....	31
	与数据阶段误差2.24.2 SET_XXX请求 .....	32
	2.24.3与设置阶段误差GET_XXX请求 .....	33
2.25	当前和未来的净荷报头格式和扩展 .....	33
2.26	Motion JPEG格式特性 .....	34

2.27	MPEG2-TS APT .....	34
2.28	主机和设备的互操作性.....	35
2.29	基于流的有效载荷支持.....	37
2.29.1	MPEG-4 SL .....	37
2.29.2	VC1 .....	37
2.29.3	H.264 .....	37
2.30	主机行为的静止图像捕获方法2 .....	37

## 表格清单

表2-1的USB视频类文件	1
表2-2设备的替换接口设置	五
一个GUID数据结构的表2-3布局	13
表2-4主机端的时钟恢复算法	20
表2-5视频设备电源	24
色彩标准和方案的表2-6矩阵	27
表2-7互操作性指南	35

## 图一览

图2-1推荐HID实现	7
图2-2输入数据状态图基于帧的视频格式	15
图2-3输入数据状态图基于数据流的视频格式	16
图2-4输出数据状态图和基于流基于帧的视频Formats17图2-5设备/主机处理分区	23



## 1 介绍

### 1.1 目的

此文件提供指导，并经常问到的关于实现USB兼容的视频类设备的问题。还介绍了标准的设备能够实现特定的支持。本文档所提供作为援助的USB视频设备类别规范的实现，并且因此，仅提供信息。如果冲突这个文档和规范之间发生，规范为准。

### 1.2 范围

实现与主机的具体问题没有基础文档的一部分 *对于视频设备USB设备类定义* 本文件中得到解决。

### 1.3 相关文件

USB规范 修订版2.0，2000年4月27日， [www.usb.org](http://www.usb.org)

对于视频设备USB设备类的定义， [www.usb.org](http://www.usb.org)

### 1.4 术语和缩略语

下表定义本文档中使用的术语和缩写。

术语	描述
CSM	内容安全方法
常问问题	经常问的问题
HID	人机接口设备
UVC	USB视频类

## 2 常见问题解答 ( FAQ )

常见问题在下面的章节讨论。

### 2.1 设置的USB视频类文件

题：什么是指定USB视频设备类标准的文件？

回答：下表列出了本文档中包含的USB视频设备类的文档。除了基本规范，其他部分描述的具体领域。

表2-1的USB视频类文件

类别	文件的主题/文件名	描述
基本规范	对于视频设备通用串行总线设备分类定义 ( USB_Video_Class )	定义整体USB视频类的框架。
终端规范	媒体传输终端 ( USB_Video_Transport )	定义媒体传输终端数字视频相机/录像机。

荷载规范MJPEG	( USB_Video_Payload_MJPEG )	定义了有效载荷和报头的使用（如果必要的话），同时基于批量和同步的视频流为MJPEG视频格式。
荷载规范DV	( USB_Video_Payload_DV )	定义了有效载荷和报头的使用（如果必要的话），同时基于批量和同步的视频流为DV视频格式。
荷载规范MPEG2TS	( USB_Video_Payload_MPEG2-TS )	定义了有效载荷和报头的使用（如果必要的话），同时基于批量和同步视频流的MPEG2TS视频格式。
荷载规范MPEG1-SS，MPEG2-PS	( USB_Video_Payload_MPEG1-SS_MPEG2-PS )	修订版1.0。已过时。看到基于有效载荷流规格。定义了有效载荷和报头的使用（如果需要），基于用于MPEG 1-SS和MPEG2-PS视频格式二者批量和同步视频流。
荷载规范未压缩	( USB_Video_Payload_Uncompressed )	定义了有效载荷和报头的使用（如果必要的话），同时基于批量和同步的视频流为未压缩的视频格式。
有效载荷规范卖方	( USB_Video_Payload_Vendor )	修订版1.0。已过时。看到基于有效载荷流规范或有效负载帧规范。定义了有效载荷和报头的使用（如果必要的话），同时基于批量和同步的视频流为一个厂商特定视频格式。
荷载规范基于流		定义了有效载荷和报头的使用（如果需要），

		同时基于批量和同步视频流进行基于数据流的视频格式。
荷载规范的基于帧		定义了有效载荷和报头的使用（如果必要的话），同时基于批量和同步的视频流为基于帧的视频格式。
例	摄像机示例（USB_Video_Example）	提供基于样本视频设备的拓扑结构的描述和要求。
常见问题	常见问题解答（USB_Video_FAQ）	回答了关于USB视频设备规格，执行标准的设备以及主机/设备的交互问题。

## 2.2 主机的USB带宽管理支持

题：如何管理主机的USB带宽的限制量？

回答：下面的段落描述了在各种操作系统可用的带宽管理政策和机制。

### 2.2.1 微软Windows

带宽管理是在视频类驱动程序 (usbvideo.sys) 的自由裁量权。在Microsoft Windows操作系统同步调整带宽使用的唯一机制是通过备用接口选择。微软Windows的所有当前版本需要的带宽被释放，并从一个非零备用接口转换到另一个非零备用接口的过程中重新分配。这意味着，同步流必须停止并重新启动。带宽重新分配是原子，但 *只要* 如果新的带宽需求都可以得到满足。这是可能的，因为所有的备用接口选择请求通过端口驱动程序 (usbport.sys文件) 序列化。然而，如果新的带宽要求不能满足时，旧带宽预留丢失，必须通过一组独立的备选接口请求重新获取 (从而成为非原子)。

当选择备用接口 (也称为配置端点)，类驾驶员能够覆盖 **wMaxPacketSize** 的同步端点那就是接口的一部分。在与视频类驱动程序和设备之间的带宽协商相结合，这将允许设备指定只有一个备用接口 (不是替代其他接口

0)，而类驱动程序将指定商定 **wMaxPacketSize** 在配置端点。

### 2.2.2 苹果的Mac OS X

在Mac OS X同步端点的带宽管理是在视频类驱动程序的自由裁量权。对于在Mac OS X调整等时带宽使用的主要机制是通过可替代的接口选择。的Mac OS X要求带宽从一个非零备选接口改变到另一非零替代接口的过程中被释放，并重新分配。这意味着同步流将被停止和重新启动。带宽重新分配是原子，但 *只要* 如果新的带宽需求都可以得到满足。这是可能的，因为所有的备用接口选择请求通过USB控制器驱动程序序列化。然而，如果新的带宽要求不能满足时，旧带宽预留丢失，必须通过一个独立的备选接口请求重新获取 (从而成为非原子)。

的Mac OS X提供了一种机制来调整同步端点所分配的带宽：如果没有足够的带宽可用于成功地创建同步端点 (使用同步端点描述符所选择的接口指定的wMaxPacketSize)，则

端点将得到的0字节的分配。然后，视频类驱动程序可以协商与视频类设备的带宽需求和使用SetPipePolicy ( ) API来指定端点所需要的实际带宽。这将允许设备指定只有一个备选接口 ( 除备用接口0除外 )，并且让视频类驱动程序和视频类设备协商适当的带宽选择。

如上所述的用于附连到高速集线器全速设备带宽分配以相同的方式完成的。然而，带宽可用的设备的量取决于高速设备的类型。高速集线器可以附加一些或所有端口的一个或多个事务的翻译。每个事务转换器将提供12 MB / s的带宽。所以，只有一个事务转换高速集线器需要分配的12MB / s的带宽 其中它的所有端口。一个4端口USB集线器具有四个交易翻译可以分配12 Mb / s至 每 的端口。

2.3 设备备用接口集

题：如何将一个设备实现确定的备用接口设定的数量的特定 视频流 接口应该支持？

回答：这将取决于该特定支持的各种视频参数组合的带宽使用情况 视频流 接口。例如，假设一个 视频流 接口支持以下参数集：视频格式：NV12 ( 每像素12位 ) 视频帧大小：320×240或640×480帧速率：5，10或15帧

该参数集将具有下表中描述的带宽使用特性 ( 假设与每微一个分组高速传输 )：

表2-2设备的替换接口设置

必需的 ( 以字节为单位 ) 的数据包大小	5个FPS	10个FPS	每秒15帧
320×240	72	144	216
640×480	288	576	864

该装置实施者可以选择实施下面的备用设置：0：MaxPacketSize = 0 1：MaxPacketSize = 290 2：MaxPacketSize = 580 3：MaxPacketSize = 870

随着交替设置这种组合，每帧不超过218个字节被浪费 ( 即，保留的但在流式传输闲置 )，这相当于总的总线带宽的不到3%。当然，更细的粒度和带宽的浪费少可以通过实现多个替代的设置来实现。然而，这可能会增加设备的成本。

的交替设置的最终选择和MaxPacketSize范围是留给设备实现。一般原则是，他们应选择以尽量减少带宽浪费任何支持的视频参数组合。

#### **2.4 USB视频类和HID接口**

**题：**需要USB视频类设备HID？如果没有，什么时候适合使用HID？

**回答：**不需要USB视频类的设备来实现，以设备状态传送给主机HID接口。的HID接口的主要区别是用户意图通过设备上的任意人机接口的通信。例如，用胶带运输设备将可能有按钮，允许用户播放，暂停或停止磁带。如果设备设计者希望的装置，使得主机软件知道这些按钮的状态进行建模，然后HID界面应该使用。然而，磁带走带的状态下，虽然通过在该设备上的按钮来控制，能够独立地通过USB视频类接口表示。也就是说，类驱动程序只关心交通的状态；它并不关心运输作出如何改变其状态。

所述USB视频类接口将是通过该主机控制设备状态的唯一接口。即使该设备实现了HID接口来代表设备上的按钮，这是真实的。随着HID，该设备上的用户控件会从内部装置机构被分离，所有按钮事件会去通过主机。主机会被它们翻译成特定功能的请求，并经由当前活动，视频设备驱动程序实例中继这些请求的设备对这些按钮事件进行响应。下图显示了Microsoft Windows平台上推荐的HID实现。

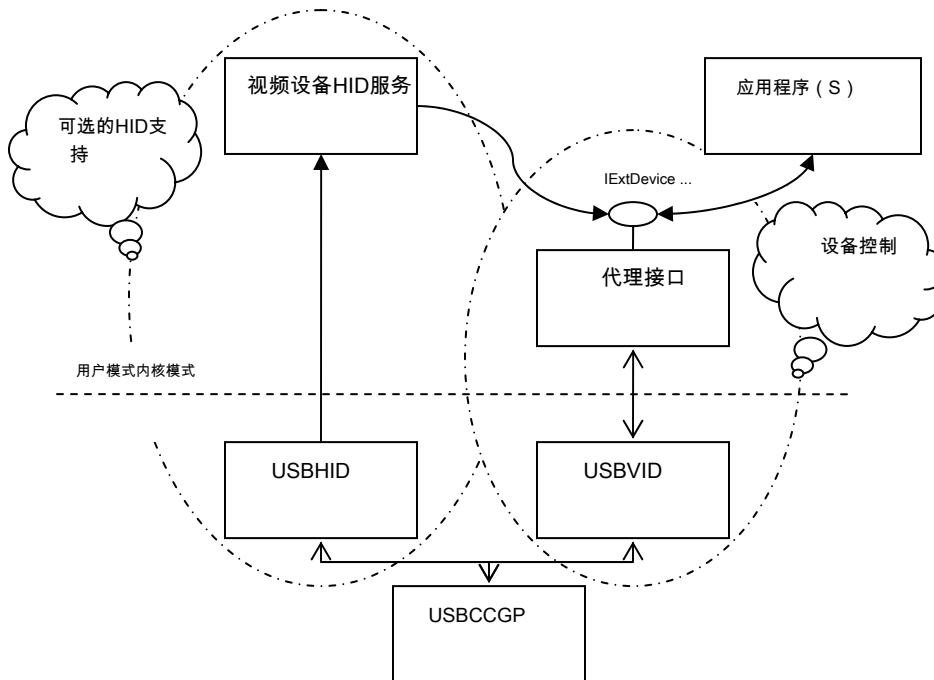


图2-1推荐HID实现

## 2.5 USB视频类范围

**题：**什么是视频设备类范围是什么？

**回答：**视频设备类定义适用于用来操作视频和视频相关的功能复合设备中的所有设备或功能。这包括设备，诸如桌面视频照相机（或“网络摄像头”），模拟视频转换器，模拟和数字电视调谐器，和静止图像照相机和数字摄像机与视频流的功能。

在覆盖这些规格设备类型：

- 桌面视频摄像机（网络摄像机）
- 数码摄像机/录像机（数码摄像机）
- 数字静止图像摄像机和视频流功能
- 模拟视频转换器

设备类型不包括在这些规范：

- 模拟电视调谐
- 数字电视调谐器
- 其他视频设备

注意：每个设备类别可以被添加作为在未来终端或单元。

## **2.6 散装与同步传输**

### **2.6.1 的批量和同步传输问题特点：什么是批量与同步传输的特点是什么？**

回答：批量传输保证了在更高的潜在比特率比同步传输无差错传输。批量传输使用尚未被使用的任何带宽由同步和/或中断端点；然而，本体 - 传输带宽不能被预留或保证。这意味着，在总线上的其它设备的竞争可能减少低于所需视频的实时传输的可用批量传输带宽。

对于视频的实时传输，如视频会议，批量传输不能保证所有的数据将被传递，或者如果交付，则不能保证按时交货。讽刺的是，原因与批量传输的无差错的性质有关。数据可以同时源缓冲时被丢弃，并且该目的地被保持在最低限度（用于低等待时间的要求）。例如，数据可以在源如果前次数据传输仍在进行中，由于重下降。或者，数据可能必须被丢弃，当它终于到达目的地，因为显示时间的样品已经过去了。因此，虽然当然是可能的，批量传输是不用于实时传输的最佳方法（例如，视频会议）由于缺乏批量带宽预留和数据传输的潜在的极大等待时间。然而，如果生产和传送整个视频帧落入帧间隔内不够好，和争用与其它设备的等待时间由用户管理，批量传输可以保证和及时的无差错视频数据的传输。

用于USB视频类设备成批传送最适合用于从顺序存储转移（例如，捕获用于非线性编辑），其中数据的准确性是主要关心的，而不是输送的及时性。

批量传输可以并处在源和目的地的额外缓冲要求。在装置中的缓冲允许数据积累，从而使在传输重试的情况下，额外的数据可以被收集而不是丢弃。这将是转移和从顺序存储，其中的传输机制没有细粒度的节流能力尤其重要。缓冲引入的延迟，而如果过大，不适合用于视频会议。



### 2.6.2 散装管道传输问题：为什么我们需要对视频流的批量管道？

回答：一般情况下，批量管道具有以下三大特点：

- 1) 访问USB带宽可用的基础上。
- 2) 重试传输的，在偶然的传递失败的情况下，由于在总线上的错误。
- 3) 保证数据的传送，但不能保证带宽或等待时间。对于下面的情况下，与前一特征的批量管道（高速，可靠性）比用于传输的视频流数据的等时管道更合适。

#### *视频编辑方案*

在视频编辑，它需要一个小时视频一个小时的数据从一个设备通过使用等时管道传送到主机。然而，它可以使用批量管道代替花费更少的时间。根据USB交通条件，批量管道可以管理比实际时间长度更快的数据。此外，视频数据，其是用于视频编辑很重要的可靠性，可以用批量传送保证由于纠错功能。批量视频传输可能是可能与大容量存储类设备；然而，它并不适用于没有一个文件系统，如FAT，连续介质存储设备，如DV摄像机。

总之，散装视频传输为视频编辑的优选的方法，因为存储在该设备的视频数据可以被更快速和可靠地传送到主机。因此，视频编辑也不会丢失数据，这将是显著价值的视频编辑器来完成。

#### *多摄像机控制方案*

在监控摄像机系统的情况下，许多相机可以连接到一台主机，虽然每一个摄像头就不需要以高帧速率发送视频数据。如果使用等时管道，每个摄像机必须预留带宽；在这种情况下，宿主应用程序将必须控制帧速率，并计算所有的摄像机的带宽，以避免带宽不足。例如，即使有二十个摄像头分别连接到主机，在主机应用程序的带宽控制将会非常复杂。但是，如果使用批量管道，带宽控制，将不会是必要的，这将是宿主应用程序的同时，控制总线上许多相机更容易。

#### *直接录制视频从USB视频设备捕捉到USB大容量存储设备方案*

此方案涉及直接从USB视频设备的USB DVD-RW设备（大容量存储类）记录数据。

如果USB视频设备使用的同步视频传输，同步管会占据大部分的USB带宽。在这种情况下，一个USB DVD-RWMS C装置，其在同一总线上的批量管带宽将被限制的，因为批量管道具有比等时管道中的较低的优先级。其结果是，录制的视频画面可能不是一个可以接受的质量。

但是，如果USB视频设备使用大容量视频传输，它不会保留总线带宽。因此，您可以捕捉到最好的方式将视频，因为两个设备的批量传输管道具有相同的优先级。

### 2.6.3 批量转让问题的数据PID排序：是否需要我的批量端点开始与具有DATA0 PID数据包有效负载的每转？

回答：号中的第2条所述的批量传输的例子 *对于视频设备USB设备类定义* 规格显示在每个传输开始DATA0的PID，但是这只是为了简化图。该 *USB规范2.0修订版* 需要用于批量传送的数据包交替DATA0和DATA1的PID。

## 2.7 音频和视频流同步

题：什么是USB视频设备类提供的同步机制？

回答：要正确地从一个媒体源同步多个音频和视频流，媒体源必须提供一定的信息给媒体宿。此信息包括其本地流的延迟，周期性时钟参考信息，以及用于媒体宿来确定合适的显示时间用于从每个流（相对于其它流）的样品的方法。

### 潜伏

媒体源被要求报告其内部延迟（延迟从数据采集到总线上的数据传送）。这种延迟反映任何缓冲，压缩，解压缩，或者通过流源进行处理引入的延迟。如果没有为每个流的等待时间信息，媒体宿（或呈现设备）不能正确地关联的每个流的呈现时间。

在视频源的情况下，这意味着源必须保证一个样品的部分充分获取作为SOF的  $N_f$  帧的开始  $N$ ）将已经完全发送到总线SOF的  $N+\delta$ 。潜伏  $\delta$  是在帧中表达的源的内部延迟。对于高速端点，分辨率增加至125  $\mu s$ ，并且延迟将在微帧来指定。每个视频流媒体接口必须报告这个延迟值。欲了解更多信息，请参阅的说明 *wDelay* 参数在“视频探头，并承诺控制”的部分

*对于视频设备USB设备类定义* 规范。通过遵循这些规则的等待时间，相位抖动被限制在 $\pm 1$ 毫秒（或 $\pm 125 \mu s$ 表示高速端点）。它是由视频宿通过调度样本的呈现在正确的时刻，考虑到所有的媒体流的内部延迟渲染同步流。

### 时钟参考

时钟基准信息用于由媒体片上进行 *时钟速率匹配*。速率匹配是指与媒体源的采样时钟的媒体宿的渲染时钟的同步。无时钟速率匹配，流会遇到缓冲过度或

运行下错误。这不是由于相对容易执行音频采样率转换的已与音频流的问题。但是，随着视频，采样率转换是显著更加困难，因此需要用于速率匹配的方法。

要了解在稍微不同的速率运行时钟的问题，考虑下面的例子。为简单起见，假定视频缓冲器可以瞬时填充，并且存在可用在视频帧时间间隔内任何给定时间被填充一个缓冲区。还假设两个晶体管辖源和渲染时钟有100ppm（百万分之一）的精度进行操作。精度值是可以被施加为使得对于每一帧，时钟将通过该帧的一小部分是等于比率漂移的比率。换句话说，两个时钟有100ppm的精度可以具有相对于最坏情况漂移到彼此的1/5000日一帧（在它们的有效工作范围内的相反的极端为 $2 \times 100 / 1,000,000$ 累积误差比两个时钟）的。因此，会发生帧毛刺一次，每5000帧。在每秒30帧的帧速率，这将等同于毛刺每166.67秒。在每秒60帧的帧速率，毛刺率较差，与每83.3秒一个毛刺。框架毛刺可以推迟，**但不能避免**，通过增加额外的缓冲它们呈现前举行的视频帧。如果源时钟运行高于渲染时钟速度越慢，缓存欠载只能通过让额外的缓冲区填充到某个阈值渲染之前，导致不可接受的延迟推迟。一旦第一个故障发生时，额外的缓冲是有效没用，因为行为会降低从该点以后的单缓冲情况。

该规范假定在所有情况下，媒体接收器具有对媒体源时钟没有控制，而源和宿不“奴”公共时钟（缺乏足够的分辨率总线时钟）。此外，由于成本限制，附加同步端点进行通信，不会被使用的时钟速率的信息。因此，本说明书中需要的视频流包括可用于调节再现时钟速率的时钟基准信息。时钟参考信息可以在传输流中被封装，或者它可以通过在每个样品报头中的可选字段中提供。该字段成为后者的情况下必需的。

#### *显示时间*

对于固定速率的流中，呈现时间可以从数据流中导出。对于一个固定利率的音频流（例如，PCM），所述媒体信宿可以推导从流偏移（通常因为捕获的开始字节的计数）的呈现时间。对于可变速率流，每个样品必须由呈现时间戳陪同。所述媒体宿负责时间戳转换为本地单元和调节所述时间戳以考虑本地时钟偏移的流开始时，以及占源流延迟。即使视频流可以以固定的帧速率在媒体宿到达，如果他们是受可变速率压缩/编码，它们不被认为固定利率流，需要对样品时间戳。

## 2.8 对于流加密的视频内容通过USB支持

**题：** 什么样的支持将用于通过USB流加密的视频内容可用？

**回答：** 该USB内容安全 ( CS ) 类定义了音频和视频设备的机制来支持加密的数据流。

底座CS类规范标准化所需要一种用于在复合设备中的内容的安全接口的端点，描述符和请求。此CS接口旨在补充现有的音频/视频接口，并允许内容安全方法 ( CSM )，以特定的音频/视频/其他数据传输设备上的端点或接口枚举和应用。

所支持在不同的伴侣规格概述，有两个目前可用于V 1.0的不同的CSM：

- CSM 1：基本认证协议 - 该规范仅定义一个主机请求以检索来自该设备的唯一的信道ID。
- CSM 2：数字传输内容保护 ( DTCP ) ( 又名5C ) - 该规范定义了需要支持DTCP通过USB请求，中断描述符和包格式。的DTCP规范本身限定了AKE ( 认证和密钥交换 ) 协议，密码协商和传输管理的设备和主机之间发送的加密数据需要的。

为了充分利用CS类，以支持加密的视频流，下面将要发生的 ( Microsoft Windows平台 )：

- 由于法律和/或许可问题，DTCP不支持微软Windows平台上。因此，新的CSM就必须定义提供类似的功能 ( 认证/密钥交换/密码协商/传输管理 )。
- 一个内容安全 ( CS ) 类驱动程序将不得不开发控制设备上的CS接口，并提供了一个定义，包括加密/解密功能的新的CSM支持。
- 新的安全视频路径将有适用于Windows平台，将确保形成数据传输通道的驱动程序和应用程序组件进行定义。这不是目前可在微软Windows平台，但正在探索在将来的版本。

2.9 一个GUID数据结构的布局

题：什么是GUID数据结构的布局和设备发送和如何解析这个数据结构？

回答：

一个GUID数据结构的表2-3布局

抵消	领域	大小 ( 字节 )	例
0	数据1	4	47504a4d
4	数据2	2	0000
6	数据3	2	0010
8	Data4a	1	80
9	Data4b	1	00
10	DATA4C	1	00
11	Data4d	1	AA
12	Data4e	1	00
13	Data4f	1	38
14	Data4g	1	9B
15	Data4h	1	71

因此，对于一个GUID定义为{47504A4D-0000-0010- 8000-00AA00389B71}，如在上面的表所描述的，小端字节流表示将是：

送出0x4d，0x4A，为0x50，0X47，0x00时，0x00，为0x10，0x00时，0x80的，0x00时，0x00，和0xAA，0x00时，0x38，0x9B，0x71

2.10 主机输入和输出数据的处理

下面的段落描述主机的数据处理状态机。

2.10.1 图术语

这些都是在下面的状态图中使用的术语的描述：

- **数据** - 这是指一个接收（或传出）的视频数据单元从到达（或将要）通过UVC VideoStreaming管设备。对于同步管道，一个数据单元被定义为USB数据包。对于批量管道，一个数据单元被定义为其中UVC标题附加的最小格式特定的块。假定一个单独的数据单元的尺寸总是小于或等于一个单一的缓冲器的大小。
- **新框架** - 这指示当主机驱动器检测通过在从先前数据包中的SID位的状态的转换接收到新的帧/样品（如果可用）的条件。
- **缓冲** - 这是指当前正在处理的数据缓冲区。数据缓冲器由主机应用程序主机驱动器为顺序读取或写入的IRP（I/O请求的数据包）的一部分提交的。驾驶员将排队这些IRP，并且在队列的头部属于IRP缓冲区是当前缓冲区。对于接收的数据，驱动程序会

填充这个缓冲器与从设备接收的视频数据，和完成读请求（即，返回填充的缓冲器返回到应用程序）时遇到的一个帧边界（由EOF和/或FID比特所指示的）时，或者当缓冲区满。对于传出数据，驱动器将在当前缓冲器向设备发送数据，设定必要的EOF和/或FID比特，并且完成写请求时，所有的缓冲数据已被发送。

- **缓冲区准备就绪** - 这表示是否有当前缓冲区可用。这种情况是真实的，只有当“提取缓冲区”状态下穿过，并挂起的缓冲区队列司机是可利用的最后一个缓冲区读取的尝试，当前缓冲区没有出队并完成。
- **空缓冲区** - 这种情况是真实的，只有如果当前缓冲区可用，它不包含任何数据。
- **缓冲区满** - 这表明当前的缓冲区是满的，或者可用的待填充的剩余缓冲器空间小于单个数据单元的大小。
- **快速完成** - 如果主机检测帧边界（ $EOF = 1$ ）或填充在当前缓冲区之后满的缓冲器，该标志设置在填充缓冲器的阶段。这是重置在获取数据的阶段为零。

### 2.10.2 输入数据状态图基于帧的视频格式

下面的状态图描述了在处理从基于帧的视频格式的设备到达的数据分组中的主机行为。SID和EOS比特的支持是必需的。

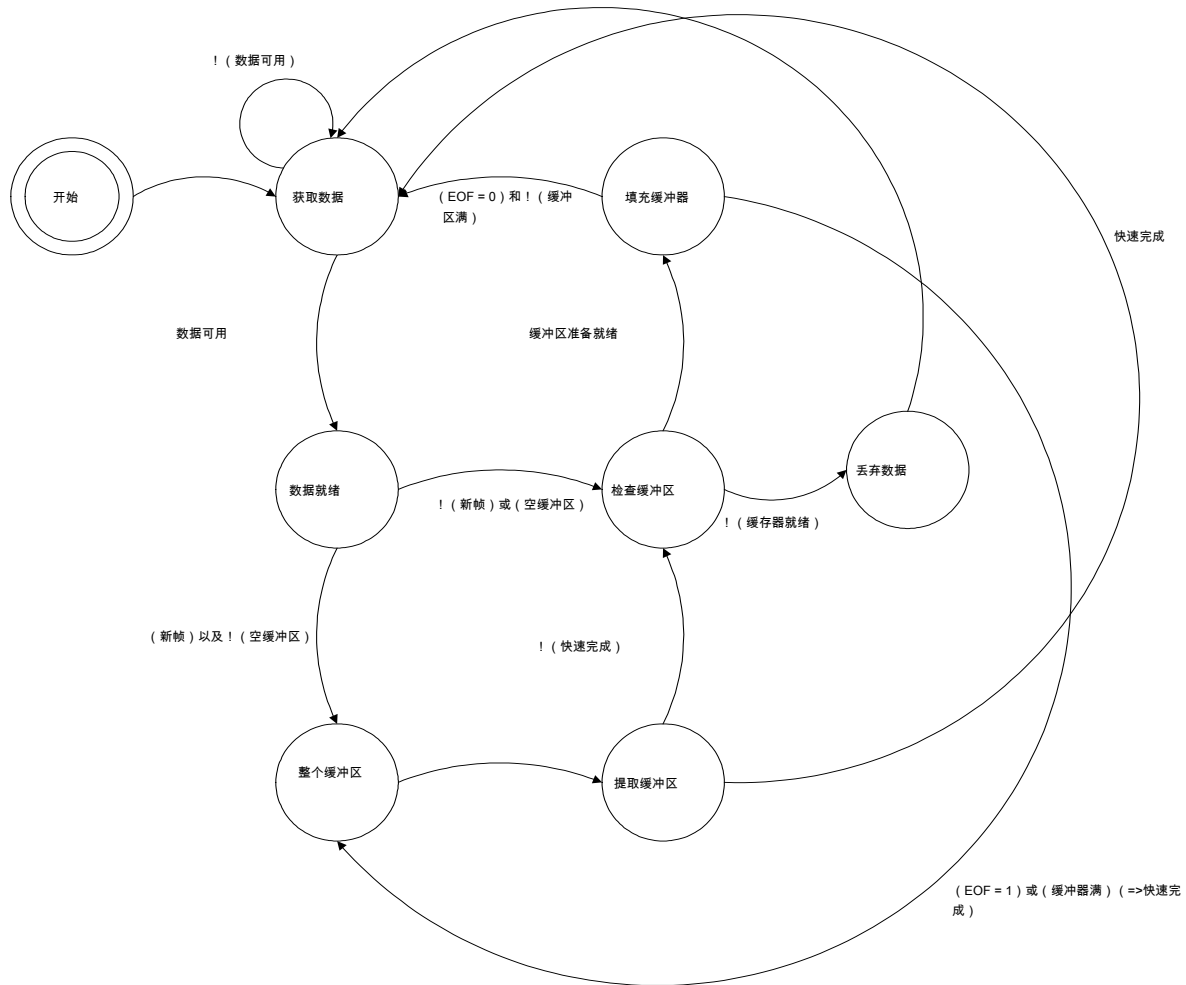


图2-2输入数据状态图基于帧的视频格式

### 2.10.3 输入数据状态图基于数据流的视频格式

下面的状态图描述了在处理来自用于基于数据流的视频格式的设备到达的数据分组中的主机行为。SID和EOS比特的支撑不是必需的。

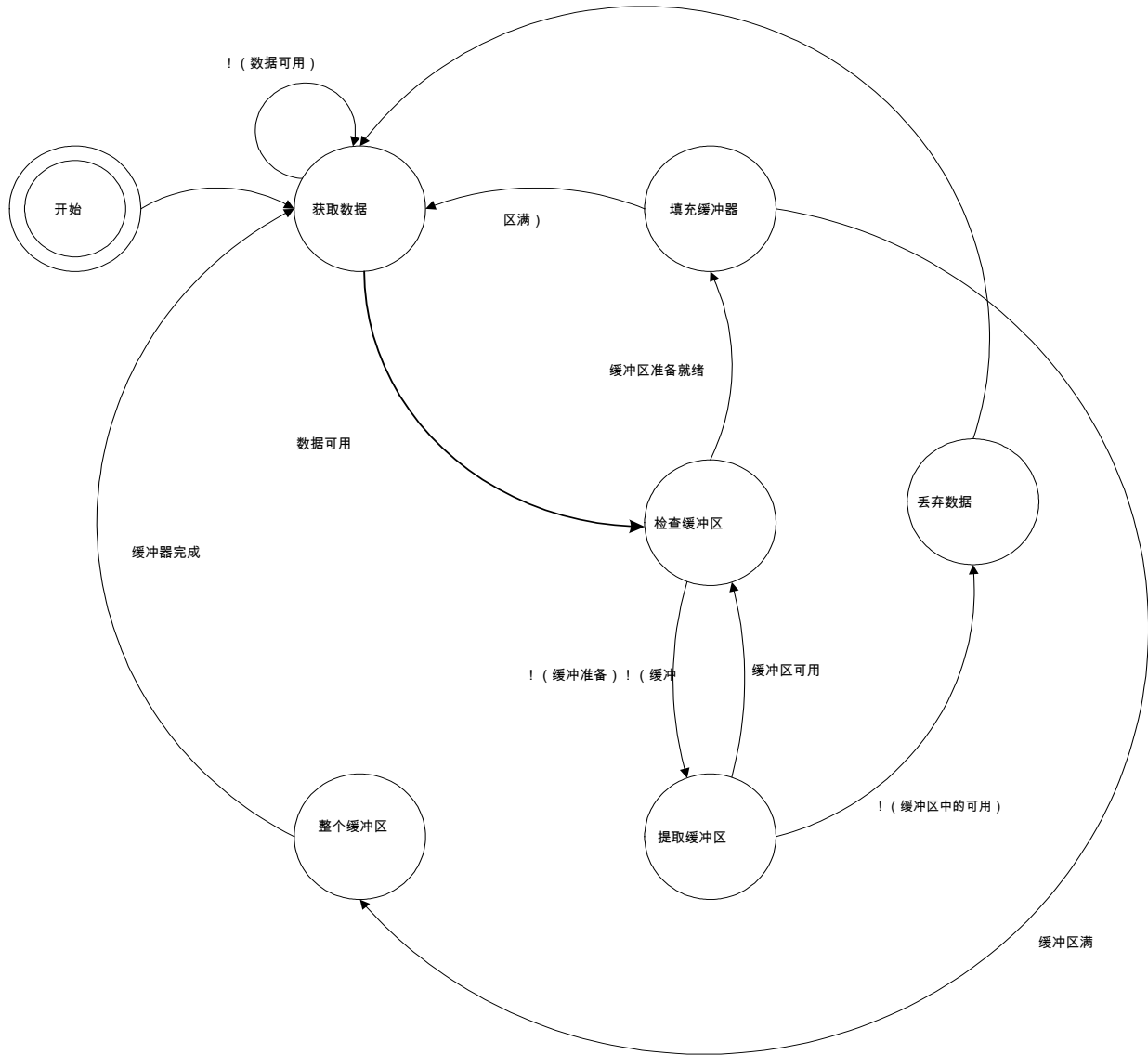


图2-3输入数据状态图基于数据流的视频格式



#### 2.10.4 输出数据状态图基于帧和基于数据流的视频格式

下面的状态图描述了在处理数据的主机行为被发送到用于基于流帧为基础的视频格式的设备。的FID和EOF位由主机基于帧的视频格式的支持。

对于支持嵌入式总线的定时信息的数据格式时，主机可能可能分析该信息，并且选择性地传输油门，以防止在设备的缓冲区溢出的数据。这是通过在以下的图中虚线所示的可选的状态指示。注意，这是从存在于UVC分组头部中的信息的定时不同BFH [0] ( SCR和PTS字段 )，其设备要在其中它们提供所有的情况下解析，以支持速率匹配和流同步。

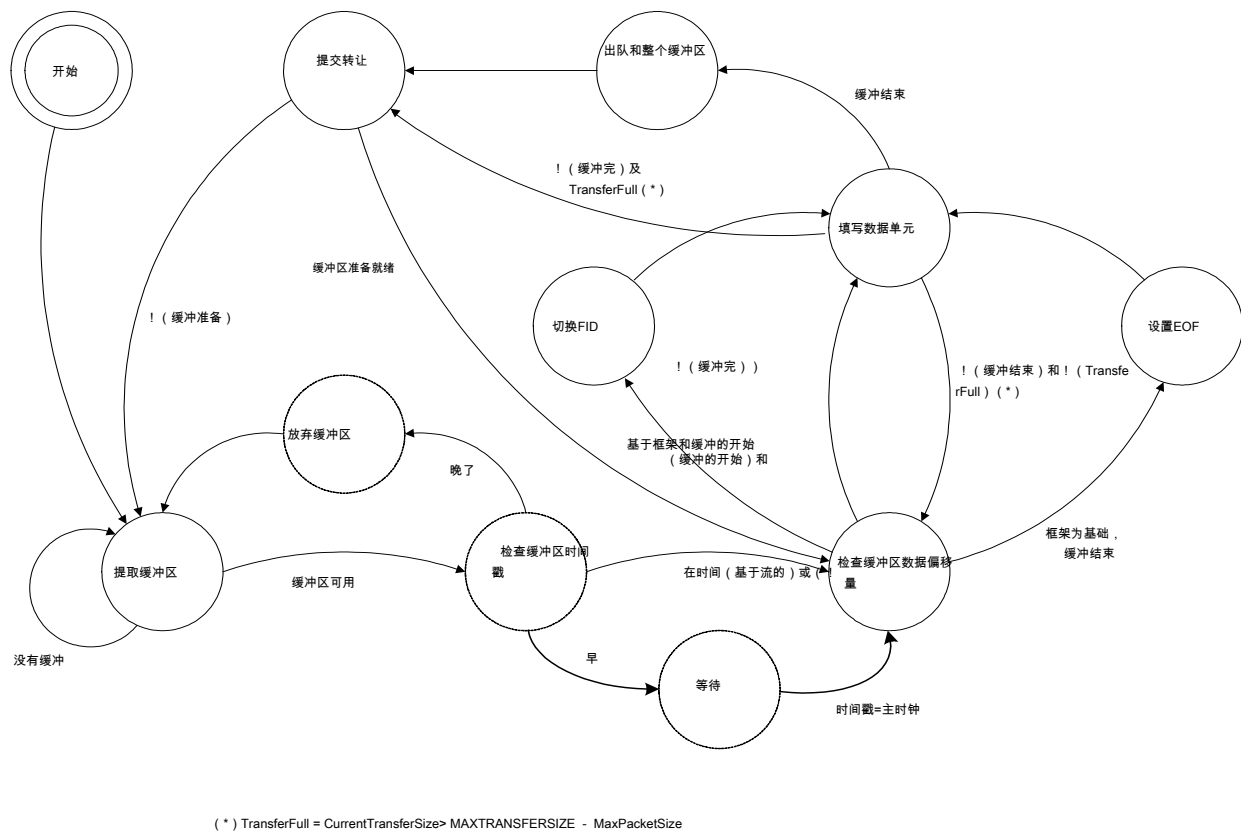


图2-4输出数据状态图基于帧和基于数据流的视频格式

### 2.11 帧间隔和帧速率的关系

题：如何在视频帧间隔从帧速率衍生（在各种有效载荷帧描述符和VS界面控件使用）？

回答：视频帧间隔在100个毫秒单位指定，并且从帧速率推导如下：对于帧速率 $x$ ，视频帧间隔是  $(10,000,000 / X)$  截断为一个整数值。

例如：

每秒15帧：帧间隔 =  $(10,000,000 / 15) = 666666$  30 fps的：帧间隔 =  $(10,000,000 / 30) = 333333$  25帧 (PAL)：帧间隔 =  $(10,000,000 / 25) = 400000$   
29.97 (NTSC)：帧间隔 =  $(10,000,000 / 29.97) = 333667$

### 2.12 时钟恢复机制

题：由USB视频类使用的时钟恢复机制是什么？

回答：所述USB视频类已通过多的附件d使用的参考[1]，其覆盖详细这个问题的相同的原理和术语。虽然采用相同的原理，USB视频类修改某些方面，如下所述。参考文献[1]的SCR（系统时钟基准，作为MPEG2节目流的定义）和PCR中（作为MPEG2传输流中定义节目时钟基准）之间进行区分。该USB视频类看中了一个学期，SCR（源时钟参考）。MPEG2可控硅和PCR中具有不同由于相应类型的流的传输的预定方法间隔的要求。所述USB视频类规范上的100ms的两个批量和同步流的单个最大间隔。

MPEG2的SCR / PCR值被用来在目的地以重建源的系统时间时钟（STC）。在一个典型的实现中，目的地将实施PLL到从一个渲染时钟的STC。参见[1]，附件d这样的PLL的描述。为了支持生产有用 $f$ 通过一个低通滤波器（LPF）的PLL内的值，则必须SCR具有足够的分辨率，以允许时钟到最大压摆率的限制范围内进行调整。

在MPEG2方法是使用用于SCR / PCR，其中最显著33位包含90个千赫时钟值的42位的值，和至少显著9位含有27 MHz时钟值。A 27 MHz时钟提供了超过足够的分辨率来保持一个为0.1Hz /秒最大压摆率的范围内（它需要至少10兆赫的分辨率）。SCR的27MHz时钟部分将一个SCR / PCR间隔期间多次滚动，但90千赫时钟部分可用于重建完整的27MHz时钟。27MHz时钟值被用作输入到LPF。

该USB视频类规范上的SCR 48位值。最不显著32位 ( D31..D0 ) 包含在源从系统时间时钟 ( STC ) 的采样时钟的值。这个时钟可以是任意的分辨率。如果源是USB设备, 分辨率是在装置实施者的判断; 如果源是主机, 该决议是基于对主机可用的最高分辨率时钟。

在该STC被采样的时间必须与USB总线时钟相关。为此目的, 该SCR ( D42..D32 ) 的下一个最显著11位包含一个1千赫SOF计数器, 表示在STC进行取样时的帧编号。该STC在任意SOF边界采样。这是相同的大小和频率, 与USB SOF令牌相关的帧号码; 然而, 它不是必需的, 以匹配当前帧号。这允许使用一个芯片组可以在SOF令牌触发实现 ( 但不是准确地获得帧号 ) , 以保持其自己的帧计数器。

最-显著5个比特 ( D47..D43 ) 被保留, 并且必须被设定为零。STC样本之间的最大时间间隔为100ms, 或视频帧的时间间隔, 取较大值。较短的时间间隔是允许的。

#### 主机端的时钟恢复算法

主机软件暴露了从动于SCR流的时钟。相同的器件时钟用于由所述装置产生的PTS值。主机通过从设备的时钟单元转换到主机时钟单位转换PTS值到天然宿主时间戳。除了重定基时间戳, 使得它们在流的开始为零, 没有其他的缩放或调整而成。时钟速率匹配是通过将比例值由主机软件露出的时钟来实现的。该标定值是通过计算设备和主机时钟之间的斜率并计数超过一个固定长度的滑动时间窗得到。

每一个SCR收到时间主机软件样本基于主机的高分辨率计数器。此高分辨率计数器被称为性能计数器 ( 或QPC的简称 ) 。每个QPC与在QPC进行采样时的USB帧计数器值相关联。其结果是一个时间戳, 在围绕每个USB帧计数器1ms的窗口变化。随着时间的推移, 时间戳平均化, 并有效地无抖动的。斜率 $SCR ( \Delta ) / QPC ( \Delta )$ 为1.0, 如果时钟在相同速率前进。如果它们不是, 则斜率提供了一个直接缩放值到歪斜QPC时钟值到SCR时钟。下面给出的算法是表现在两个主要部分。第一部分示出了在从设备接收到一个SCR的发生的处理。第二部分示出了当主机软件中查询当前主时钟值时发生的处理。正是这个值被调整为保持与器件的时钟速度。所述算法从在捕获场景的主机的角度示出; 然而, 同样的原理可以在渲染场景中应用到设备实现。为了简便起见, 不小心地检测系统错误,

的不连续性，也不会算术溢出。浮点运算使用除非另有说明。该算法可以被修改以使用整数运算。

下面的值和函数定义（临时变量没有在此表中所描述的）：

**表2-4主机端的时钟恢复算法**

Accumulate_QPC ( )	样品主时钟和USB帧编号，纳入新的对与之前的时钟和帧号对。
Accumulate_SCR ( )	结合与以前的时钟和帧号对所述设备的时钟和帧计数器从当前SCR。
SOF_delta	<b>SCR之间观察到的帧（数 <math>N</math>）通过SCR（米），其中，SCR（<math>N</math>）代表最早的SCR可用，和SCR（<math>M</math>）表示在最近。需要注意的是SCR历史可修整以允许时钟若干机制，是更适应在主机和设备时钟的短期波动。</b>
QPC_delta ( $N$ )	数量在主机时钟滴答观察到最近的 $\tilde{n}$ SOF包（返回0，如果历史足够的可用）
SCR_delta ( $N$ )	在设备的时钟在最近的观察帧的数量 $\tilde{n}$ SOF包（返回0，如果历史足够的可用）
QPC_freq	主机时钟的频率（通过宿主依赖机制初始化）
SCR_freq	设备时钟的频率（通过USB视频描述符由所述装置提供的）
ClockSlaveRatio	在每个接收到的SCR计算出的时钟从比（初始值1.0）
MinSlavable	视为有效的时钟随动的最小比值（恒定值0.8）
MaxSlavable	视为有效的时钟随动的最大比率（恒定值1.2）
MinSOF_delta	毫秒时钟携动之前的最小数目可以开始（固定值2000）
QPC ( )	当前主机时钟值
最后未缩放主时钟值CarriedError的BaseQPCForLastReturnedTime记录	
	错误的记录，从投从浮点产生于整数
SlavedClock	该从属时钟值

第1部分：一个SCR的收据

Accumulate\_QPC ( ); 积累SC

R ( );

如果 ( SOF\_delta > MinSOF\_delta ) {

    dHostDelta = QPC\_delta ( SOF\_delta ) / QPC\_Freq; dMasterDelta = SCR\_delta

    ( SOF\_delta ) / SCR\_Freq; 如果 ( dHostDelta != 0 ) {

        dNewRatio = dMasterDelta / dHostDelta; 如果 ( dNewRatio >= Min

        Slavable &&

        dNewRatio <= MaxSlavable ) {ClockSlaveRatio =

        dNewRatio; }}

第2部分：主时钟查询

llNow = QPC ( );

llDelta = llNow - BaseQPCForLastReturnedTime; dDelta = ( 双 ) llDelta; //根据

我们的斜率dDeltaScaled = dDelta \* ClockSlaveRatio规模; //累积误差dDeltaSca

led += CarriedError;

//计算，我们将使用llDelta = ( LONGLONG ) dDeltaScaled非FP值; //携带错误到下一呼叫Ca

rriedError = dDeltaScaled - ( 双 ) llDelta; //保存最后返回时间BaseQPCForLastReturnedTi

me = llNow的基础; SlavedClock += llDelta;

参考文献：

[1] ISO / IEC 13818-1：信息技术 - 运动图像及其伴音信息的通用编码：系统

### **2.13 MaxPayloadTransferSize选择**

**题：**应如何MaxPayloadTransferSize值来选择？

**回答：**如所定义的，MaxPayloadTransferSize参数取决于由一个视频流接口支持的USB传输的类型。

对于一个同步数据流管，该值应是低于或等于在所选择的备用接口的流端点的USB wMaxPacketSize。主机软件负责基于缓冲，延迟和处理开销选择适当的同步传输的大小。

对于批量传输，这个值是特定于实现的，并且受以下权衡：

- 一个较大的值可能增加解码延迟的USB主机控制器完成的批量传输之后将只完成一个I/O请求包。因为该值用于存储器缓冲区分配的主机上，这潜在地增加了主机缓冲要求。
- 一个小的值将增加的I/O请求分组业务的量，并增加要传送流标题的数量。

根据目标应用和折衷如上所述，

MaxPayloadTransferSize应该被设置为10ms的包括价值的数据和dwMaxVideoFrameSize的之间的值。

### **2.14 负载格式向前兼容性**

**题：**如何才能有效载荷未来格式的支持？

**回答：**未来有效载荷格式可以通过定义根据卖方有效载荷规范文档有效载荷被支撑，不需要改变主视频类驱动程序。

### **2.15 设备/主机处理分区**

**题：**与设备/主机处理划分设备支持？

**回答：**设备依靠主机进行一些处理上是由设备类规范的支持。这些装置依靠一个供应商特定有效载荷格式，并依靠提供到设备的单元（一个或多个）和终端（一个或多个）从相关联的编解码器的控制和事件路径主机软件类驱动程序的功能。

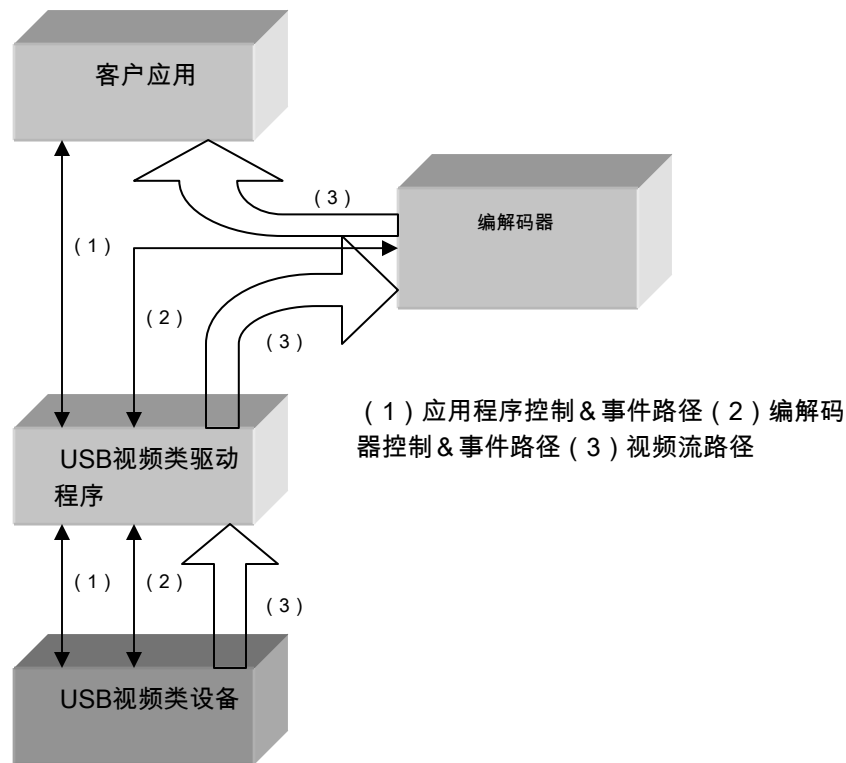


图2-5控制器/主机处理分区

## 2.16 电源模式控制

题：什么是功率模式状态是什么意思？

回答：电力模式控制描述控制装置电源模式。有可用于操作设备三个电源：

- 1) 交流电源
- 2) USB
- 3) 电池

在头两个电源的情况下，用户并不需要与电力消耗有关。但是，如果当其他设备也连接到它的主机无法提供足够的电力，使设备工作时，主机可以使用的电源模式控制，使设备与低功耗工作。

如果电源是电池，用户需要关注与功耗，并应使用的电源模式控制，以节省电池功耗。例如，如果

该器件具有不使用一些设备功能，以降低功耗，使用的电源模式控制这样做的能力。

表2-5视频设备电源

能量源	描述
交流电	通过设备交流电源驱动
USB	通过设备USB电源驱动
电池	设备由电池电源驱动

配置描述符可以指示电源（S）的装置可以使用其中主机（总线供电和/或自供电）。另外的宿主可以使用命令的getStatus确定设备当前是否自供电。然而，主机没有办法知道自供电设备是否使用交流电源或电池电源。指定电源会使主机能够知道在设备侧的供电的状态。

### 2.17 探头和提交作业

题：我的设备只支持一个流参数设置（格式，分辨率，帧速率等），应该怎样处理探测/提交控制来实现？

回答：所有的请求和属性需要支持，但所有的GET\_XXX请求将始终返回相同的数据结构的内容。提交SET\_CUR请求仍必须验证所需的参数。

题：如何通过支持流的属性做我循环？

回答：发出带有GET\_XXX属性的请求。在返回的控制数据结构，通过减去1更改要循环流场（dwFrameInterval，wKeyFrameRate，wPFrameRate，wCompQuality，wCompWindowSize），并发出SET\_CUR要求：这将迫使设备使用下一个更低的支持值这个根据协商环路避免一个GET\_CUR请求被发出时域。主机检测枚举的末尾时，设备状态已经不改变（或所列举的场达到其最小通过GET\_MIN请求返回）。

题：在什么情况下可以探测/提交请求失败？

回答：探针/提交请求将失败：

- bFormatIndex和/或bFrameIndex具有无效值
- 通过bFormatIndex和bFrameIndex指定一个给定的流不能与指定dwMaxPayloadTransferSize支撑。所有其他领域将进行谈判

题：如何主机软件使用的视频探头，并承诺控制协商流参数和视频压缩特性，如关键帧速率，等等？



回答：视频探测器和提交控制是非常灵活，可以根据主机软件的设计和要求许多不同的方式来使用。

以下是如何主机的软件可以使用探头，并承诺控制只是一个具体的例子。

( 注：在此示例中是指所引用的视频压缩场/参数

**KeyFrameRate** , **PFrameRate** , **CompQuality** 和 **CompWindowSize** 字段 ) 的视频压缩领域检索

范围信息 ( 关键帧速率等 )

- 1.主机设置格式指数 ( 和帧索引/间隔，如果支持的话 )，将所有其他字段为零时，问题SET\_CUR探测控制
2. GET\_MIN探测控制以获得用于视频压缩领域最小值
3. GET\_MAX探测控制，以获得视频压缩领域的最大值
4. GET\_DEF探测控制，以获得视频压缩领域的默认值
5. GET\_RES探测控制，以获得视频压缩领域的分辨率值
- 6.主机从步骤1重复与一组不同的格式 ( 和帧索引/间隔 ) 的索引

设置视频压缩参数流开始之前 ( 用户选择的格式/帧索引和帧间隔已知的 )

- 1.设置所有的 **bmHint** 根据用户偏好字段固定/可变
- 2.设置 **MaxPayloadTransferSize** 字段设为零 ( 延迟 和 **MaxVideoFrameSize** 字段始终由装置设定的 )
- 3.准备探针/提交数据结构：设置 **FormatIndex** , **FRAMEINDEX** 和 **FrameInterval** 字段的值由用户/应用选择

设置视频压缩参数 ( **KeyFrameRate** , **PFrameRate** , **CompQuality** , **CompWindowSize** ) 由用户/应用选择值

4. SET\_CUR探测控制
5. GET\_CUR探测控制
- 6.保存 **KeyFrameRate** , **PFrameRate** , **CompQuality** , **CompWindowSize** , **MaxVideoFrameSize** , 延迟 和 **MaxPayloadTransferSize** 从场设备返回。

获取视频压缩参数流开始之前 ( 用户选择的格式/帧索引和帧间隔已知的 )

- 1.如果需要的参数缓存，返回缓存值客户端应用程序
- 2.如果需要的参数没有被缓存，准备ProbeCommit数据结构：设置 **FormatIndex** , **FRAMEINDEX** 和 **FrameInterval** 字段的值由用户/应用选择

设置视频压缩参数 ( **KeyFrameRate** , **PFrameRate** , **CompQuality** , **CompWindowSize** ) 由用户/应用来选择值。

设置基于用户偏好和先前初始化参数bmHint字段来固定/可变。

3. SET\_CUR探测控制
4. GET\_CUR探测控制
5. 返回请求的参数给客户端应用程序

#### 流格式协商 ( 设置为流 )

1. 根据用户的喜好设置所有bmHint字段固定/可变
2. 设置 **MaxPayloadTransferSize** 字段设为零 ( 延迟 和 **MaxVideoFrameSize** 字段始终由装置设定的 )
3. 准备探针/提交数据结构 : 设置 **FormatIndex** , **FRAMEINDEX** 和 **FrameInterval** 字段的值由用户/应用选择

设置视频压缩参数 ( **KeyFrameRate** , **PFrameRate** , **CompQuality** , **CompWindowSize** ) 由用户/应用选择值

4. SET\_CUR探测控制
5. GET\_CUR探测控制
6. 保存 **KeyFrameRate** , **PFrameRate** , **CompQuality** , **CompWindowSize** , **MaxVideoFrameSize** , 延迟 和 **MaxPayloadTransferSize** 从场设备返回。
7. 流当即将开始, SET\_CUR使用来自步骤6的值提交控制 ( 这应该总是成功, 因为它是基于值从探针控制返回 )
8. 试图选择ALT设定基于 **MaxPayloadTransferSize** 保存在步骤6 ( 仅ISOCH管 )
9. 如果失败, 返回故障代码, 或者对于ISOCH管道, 更新 **MaxPayloadTransferSize** 并试图通过去到步骤3以选择较小的备选接口设置。

#### 流中获取视频压缩参数

1. GET\_CUR提交控制

#### 流中设置视频压缩参数

1. GET\_CUR执行 ( 以检索有源器件状态 )
2. 正被设置更新参数
3. 设置 **MaxPayloadTransferSize** 场到当前备选接口最大包大小
4. SET\_CUR探测控制
5. GET\_CUR探测控制
6. 如果返回的参数是可以接受的, 问题SET\_CUR提交控制
7. 如果SET\_CUR发出, 保存 **KeyFrameRate** , **PFrameRate** , **CompQuality** , **CompWindowSize** , 延迟 和 **MaxPayloadTransferSize** 领域
8. 如果SET\_CUR没有发出, 返回失败代码

笔记 :

探头控制是指VS\_PROBE\_CONTROL提交控制是指VS\_COMM  
IT\_CONTROL

### 2.18 输入和输出端子协会

**题：**在输入或输出终端描述符bAssocTerminal字段，什么是两个终端之间的关联的含义是什么？

**回答：**在规范相关联的终端的寓意是，如果我们两个终端相关联，我们不能在同一时间使用它们。例1：

如果我们使用媒体传输终端，如磁带。

由于磁带终端不能同时供读取和写入数据被使用，对应于该磁带中的输入和输出端子应在相应的描述符相关联。

例2：如果我们使用媒体传输终端，如随机存取媒体。由于随机存取介质，可用于阅读和在同一时间写入数据，对应于该随机存取介质的输入和输出端子，不得在相应的描述符关联。

### 2.19 多个配色描述符

**题：**如何处理多个配色的描述？

**回答：**在颜色匹配描述符改变通过用于动态格式改变相同的机制来处理。

**题：**有没有理由在描述三种颜色匹配字段？

**回答：**就在这里。有上色，而且往往这些标准并不反映当前的使用场景相关的众多设备和标准。要解决这些模糊之处需要显式声明的颜色格式。该问题部分地由下面的表格中所示。它显示了不同的场景和在定义颜色所使用的标准。

色彩标准和方案的表2-6矩阵

基色	转换功能	亮度矩阵	示例场景
709	709	709	DVB HDTV标准建议。
709	709	601	典型的网络摄像头
240	709	601	NTSC调谐器/视频捕捉设备
709	709	601	PAL调谐器/视频捕捉设备
709	240	240	HDTV ( 六十分之一千〇三十五 )
170	170	170	DVB 30Hz的SDTV
470	*	*	DVD的
240	470 ( M )	601	NTSC SDTV

709	470 ( B , G )	601	PAL SDTV
709	709	709	ATSC ( 1920至1960年 )

**题：**为什么NV12选择作为推荐4：2：0 YUV平面格式？

**回答：**NV12选择用USB视频类，因为它是显卡的首选格式。这是因为，NV12具有高速缓存一致性和整体显卡的性能方面显著的优势。目前显卡已经开始支持这种格式，并且在将来所有的显卡都有望做到这一点。

## 2.20 请求错误代码控制

**题：**当请求错误代码控制值进行更新？

**回答：**请求错误代码控制是任何控制请求（包括异步控制请求和请求到请求错误代码控制本身）之后更新。如果成功，则请求错误代码控制值被复位为0。失败时（在控制配管协议失速），则请求错误代码控制值被设置为失败原因。异步控制操作（通过状态中断端点）的后续完并不影响请求错误代码控制。

## 2.21 等时管道和数据可用性

**题：**什么是发射器就当没有数据可用于发送同步端点预期的行为？

**回答：**同步端点必须接受（OUT端点）或返回（IN端点）一个零长度分组对于其中数据是不可用的每个总线间隔。请注意，如果用于递送源时钟参考（SCR）的最大时间间隔将时间段期间没有数据否则到期由有效载荷报头的单独（没有有效载荷数据）的有效载荷传递被允许。

使用高带宽同步端点设备，该设备必须准备好接受（OUT端点）或返回（IN）零个长度分组与取决于每微其他事务机会的数量和的量的0,1或2数据的PID缓冲器剩余。请参考第5.9.2和8.5.5的 *USB规范2.0修订版* 对于进一步的细节。此外，EHCI规范的USB修订版1.0的附录d明确指出，USB 2.0主控制器（并从中导出投诉设备什么的）应遵循正确处理高带宽同步端点的规则。

## 2.22 隔行扫描视频

**题：**什么是隔行扫描视频领域的背景下均值：顶部，底部，场1，场2，奇数和偶数？

**回答：**顶部和底部是指在视频帧中的实际图象行的空间术语。第一画面线与顶场相关联，并且所述第二图象扫描线相关联

与底部场。场1和场2是颞条款。第一场到帧被称为场1内被捕获，而另一字段称为字段2单双过载条件，因为它们可能是指奇数/偶数图象行，或奇数/偶数的模拟视频场的行，基于这个原因，我们会从我们的解释使用这些术语避免。

**题：** 什么是USB视频类支持各种隔行视频格式？

**回答：** 该USB视频类需要隔行扫描视频内容中的两种形式发送：

- 1) 隔行扫描：两场是在同一个视频样本中发送，但顶部的线条和底部场是交错的。即这两个领域的线保持交流。
- 2) 每采样1米栏：每个样品仅包含一个字段和FID位被用来指示一个给定的视频样本中是否含有一个顶场或底场（见下文项目）。

**题：** 如何使用FID位表示为隔行扫描视频顶部和底部场？

**回答：** 在一个场的每样本的隔行扫描视频的情况下，为1的FID值表示顶场，而值0表示底场。所述FID位，随着所述D2位使用时

**dwInterlaceFlags** ( Field1First ) 压缩格式描述符的领域，让PC主机相结合的领域以明确的方式。需要注意的是在时间上，场1总是假设为在每个采样视频数据的一个场场2之前发送。

**题：** 什么是隔行扫描标志的“Field1First”有点意思？

**回答：** “Field1First”必须被理解为“Field1First - 空间”。这意味着在空间上第一场或顶场，首先捕获，并且是在每个样品的视频数据一个场将要被发送的第一个字段。

**题：** 应该如何框架描述符字段被解释为一个场，每个样本视频数据？特别是，应该怎样的价值 **wHeight** 是？

**回答：** 帧描述符的值应参考字段仅，而不是对整个视频帧。就这样 **dwMaxVideoFrameBufferSize** 指的字节数一个样品（场）中，帧间隔是指场之间的时间间隔，并 **wHeight** 指的是在一个场的行数。因此，对于一个29.97，720 \* 576的视频，帧描述符会报告

59.94（每秒场）和 **wHeight** = 288线。

**题：** 什么是Microsoft Windows操作系统建议的格式？

**回答：** 在微软操作系统中，每个样品两个字段，行交错格式是最好的支持隔行扫描视频格式。

### **2.23 净荷报头中的PTS和STC领域的最大值**

**题：** 什么是PTS和STC（SCR的一个部分）的净荷报头的字段的最大值？

**回答：** 这些字段的最大值可以是0xFFFFFFFF的（一个32位无符号值的最大值），或者该流减一的时钟分辨率。例如，如果流与13.5MHz处的一个时钟分辨率递送，允许PTS和STC的值达到或者0xFFFFFFFF的或13499999后换到零。它是该视频接收来确定，而在运行时处理这些字段正在使用哪个最大（从主机捕获当从装置，所述装置的固件捕获当主机软件）的责任。字段可以独立于任何的两种方法相同的实现内包装。

## **2.24 协议延迟行为**

**题：** 如何做一个视频功能手柄请求参数的验证和错误报告？

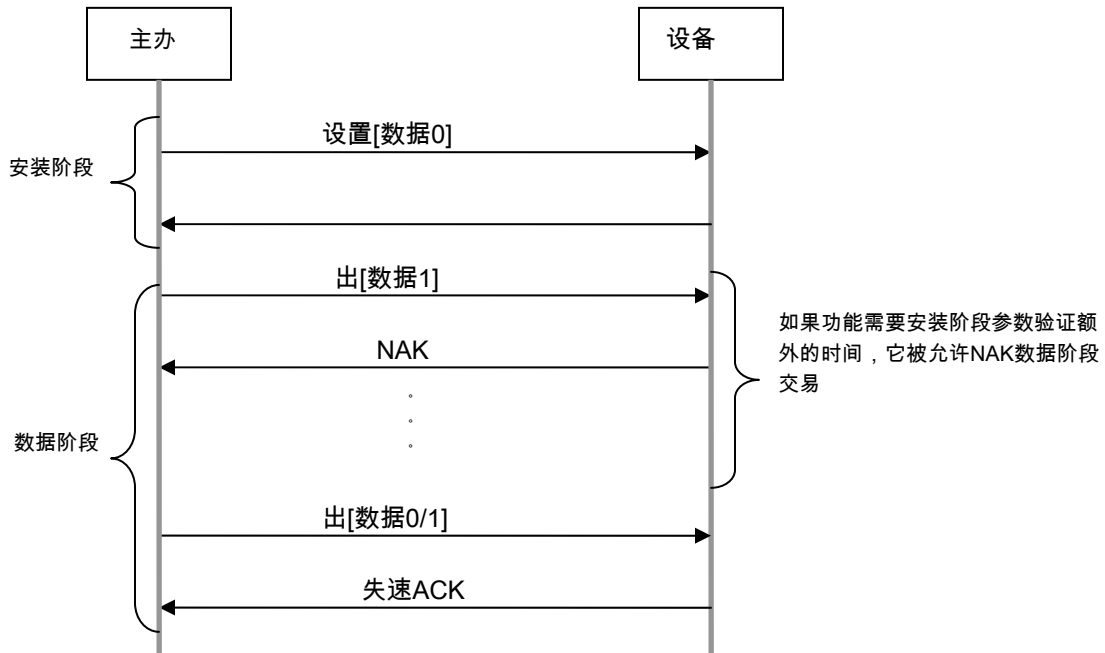
**回答：** 所述USB视频类别规范定义了一个协议STALL以允许视频功能报告的错误; 该功能需要验证控制传输参数和无效时报告错误。

此外，请求错误代码控制定义为允许误差的原因精确的报告。

下面的段落描述控制传输错误的潜在处理。

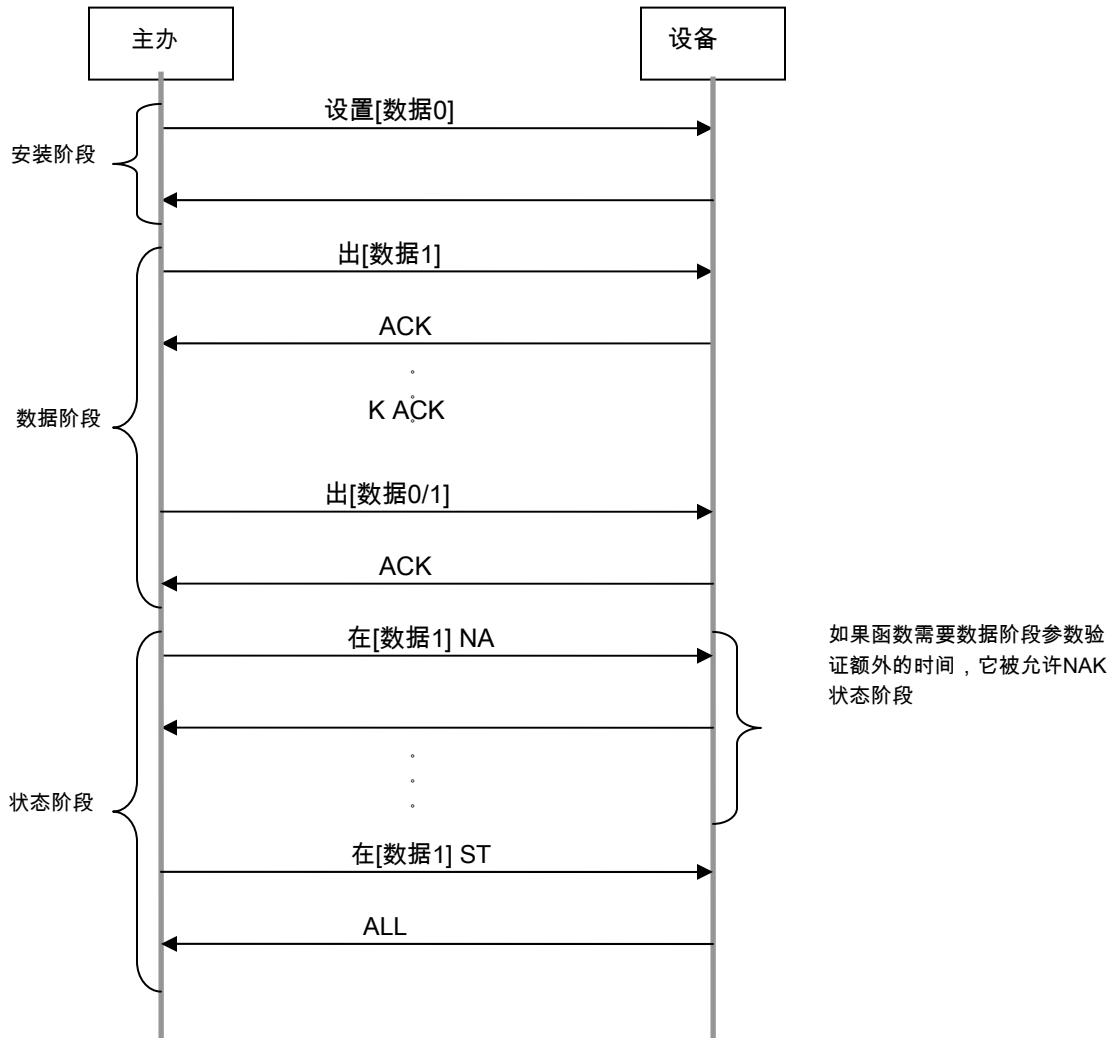
### 2.24.1 与安装阶段错误SET\_XXX请求

这个例子显示了由于无效的设置阶段参数的控制失败写入传输。需要注意的是失速状态也状态阶段过程中报告。



### 2.24.2 与数据阶段错误SET\_XXX请求

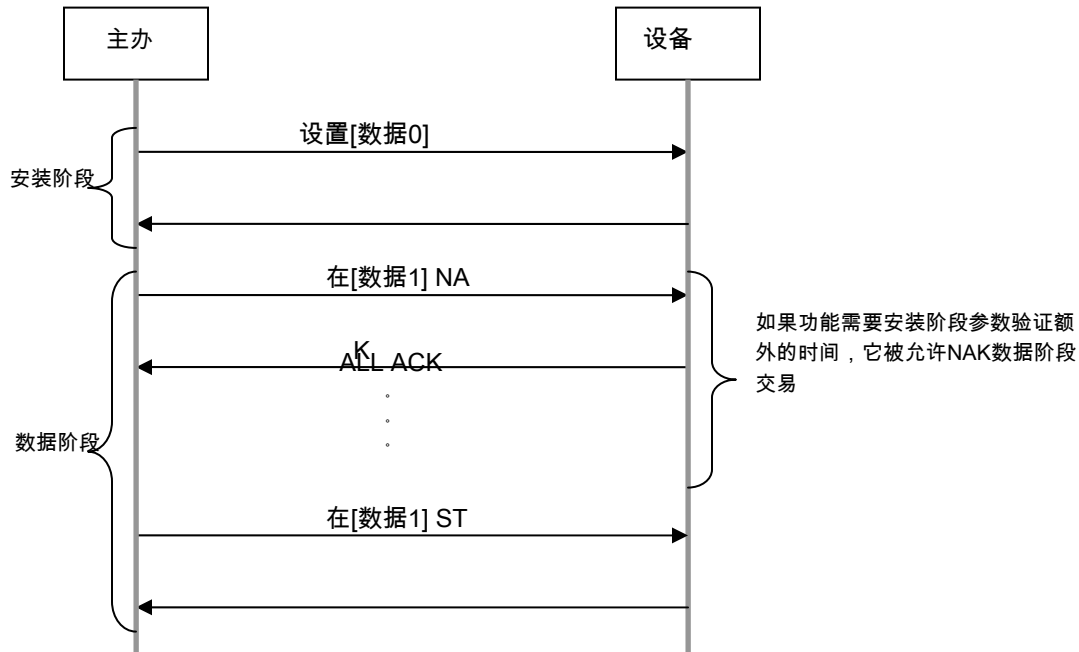
这个例子中示出了由于无效数据阶段参数发生故障的控制写传输。





### 2.24.3 与安装阶段错误GET\_XXX请求

这个例子显示了一个失败的控制读取传输由于无效的设置阶段参数。



### 2.25 当前和未来的净荷报头格式和扩展

题：可以净荷报头是固定的大小？

回答：是。只要所述净荷报头的bHeaderLength值可以指定任何长度，它足以允许净荷报头，以保持由bmHeaderInfo位图字段（一个或多个）所指示的信息。如果bHeaderLength值大于必需的，额外的字节被简单地忽略。

题：能否bmHeaderInfo字节进行扩展，同时保持与增加新的bmHeaderInfo字节之前发布的类驱动程序的版本的向后兼容性？

回答：是。只要前一类驱动程序的版本使用“报头结尾”（D7）的bmHeaderInfo字节（一个或多个）的位，以确定当剩余净荷报头的值（如果有的话）开始，所述净荷报头的bmHeaderInfo部分可以是任何达到由bHeaderLength值和任何附加的净荷报头数据允许的最大字节数。

题：是否有放置在bmHeaderInfo扩展位分配任何限制？

回答：是。“页眉结束”（D7）位必须存在于每个扩展bmHeaderInfo字节。在D7位的值将是0为所有，但最后bmHeaderInfo字节。

## 2.26 Motion JPEG格式特点

题：为什么从我的MJPEG设备输出的颜色似乎是人为提升？

回答：较新的视频卡假设视频是在YCbCr颜色空间（即，使用BT-601视频范围）。如果图像与在0-255范围内的亮度和色度的单位，用于典型的JPEG静态图像编码，然后将饱和度和对比度会人为地看视频时的假设下呈现的水平在提高的YCbCr色彩空间。BT-601指定包括八比特编码，其中Y是在16（黑色）的范围内，以235（白色）。类似地，Cb和Cr是在包容16至240的范围内。

## 2.27 MPEG2-TS APT

题：用于MPEG TS APT方法如何，均实现相关的恒定值和所需的缓冲区大小来确定

回答：下面的两个实施例表明如何确定实现相关的恒定值和所需的缓冲区大小。例1：如果接收器的本地计数器（*microframe\_count* : *microframe\_offset*）保持不变：

在这种情况下，信宿将计算被接收时，它被添加到每个APT值的恒定值。该值是特定流恒定。恒定值将直接决定当每个MPEG分组被释放到应用程序和间接地确定信宿的缓冲区大小。恒定值等于接收器和源的计数器加上一些微帧号之间的差。微帧的额外数量将弥补任何USB抖动和传输延迟。例如，如果在一个流的第一APT值背后的本地正好10微帧（*microframe\_count* : *microframe\_offset*）值被接收时，该恒定值可被设置为13个微帧，使得该特定分组将留在缓冲器3完全相同微帧。由于固有的分组递送抖动通过USB，以及连续包将留在1个5之间微帧缓冲器，条件是相同的恒定值被使用。因此，缓冲器的大小应该足够大，以存储至少5个微帧值得在此实例中的数据。

注意：虽然“3个微帧”的值在本例中为缓冲所提到的，每个系统的值是依赖于实现的。例2：如果接收器的本地计数器（*microframe\_count* : *microframe\_offset*）值在每个流的开始时被初始化：

如果水槽的地方 ( *microframe\_count* : *microframe\_offset* ) 值在一个流的开头适当地初始化，每个APT的值可以针对本地进行比较 ( *microframe\_count* : *microframe\_offset* ) 值来确定递送给应用程序的时间。没有实现相关的常数是必需的。为了应对固有的数据包传输抖动了USB，水槽应该改变其本地定时器设置为源的柜台后面几个微帧所以每个数据包缓冲一段时间保持之前它被释放到应用程序。例如，如果在一个流的第一APT值背后的本地恰好6微帧 ( *microframe\_count* : *microframe\_offset* ) 值，9个微帧的值可以从本地中减去 *microframe\_count* 计数器，以便在这个特定的数据包会在缓存整整3微帧。由于固有的分组递送抖动通过USB，以及连续包将留在1个5之间微帧的缓冲区。因此，缓冲器的大小应该足够大，以存储至少5个微帧值得在此实例中的数据。

2.28 主机和设备的互操作性

题：设备和驱动程序之间的向前和向后兼容性是如何处理的？

回答：该UVC规范定义，通过该设备和驱动程序必须遵守的协议; 作为规范的发展，设备将整合这些变化和驱动程序必须是有弹性的，以他们。设备预计将作出适当的反应基于早期规范版本 ( S ) 到主机驱动程序请求，并承担未指定的字段中有意义的默认值。下面是解决这些互操作性问题的指导方针。

表2-7互操作性指南

设计变更	预计主机行为	如何解决问题/评论
现场的重新定义		保留旧场，并确定新的领域。
附加描述符	跳过它基于类型和子类型 ( 如果不知道，跳过 )	
描述符字段的添加 ( S )	描述符长度字段可以被用于跳过未知字段。	
新终端/单元	主机类驱动程序要么笼统揭露它或忽略它。	如果主机类驱动程序不能笼统公开新终端/单元，新的描述将被老司机被忽略。连同旧的驱动程序一起使用时，设备应具有合理的默认行为。
在新的控制 现有单元/终端 一般暴露或忽略		可以由主机类驱动程序一般暴露。

额外 探头/犯下场	忽略它 ( 设置为默认 )	所述GET_LEN请求允许主机类驱动器尺寸的探针/提交结构正常。主机必须始终提交由设备指定长度的要求，将所有未知领域零。当它提交决胜盘的提交控制，必须保留随后由设备返回的值。
现有的控制变化	主机忽略未知的功能。	<p>以下准则允许维持向前/向后兼容性：</p> <ul style="list-style-type: none"><li>• 新功能是附加字段的形式</li><li>• 原来的领域保持语义和语法上与前面的定义相同。</li><li>• 如果以前没有为控件定义的GET_LEN请求应成为强制性的。</li><li>• 如果GET_LEN请求被支撑，主机必须与由所述设备 ( 即使GET_LEN以前未为该控制所定义的 ) 中指定的长度提交请求。</li><li>• 如果以前没有为控件定义的GET_DEF请求应成为强制性的。主机将只尝试，如果GET_LEN一个GET_DEF请求</li></ul>

		表示有主机不理解的字段。如果GET_DEF请求不支持较早的规范版本，该设备将只返回默认值或在其中GET_DEF要求成为强制性改版后添加的控制范围内的字段。
附加流报头字段	忽略它	可使用当前头定义支持
有效载荷的内容变化	目前不能支持	该UVC规范版本1.1通过延长探针和提交请求提供必要的版本控制。

**2.29 基于流的有效载荷支持**

**题：**基于流的有效载荷，如MPEG-4 SL，VC1，H.264由USB视频设备类规范的支持？

**回答：**这样的有效载荷依赖于MPEG-2系统规格（ISO / IEC 13818：2000 / ITU-T H.222.0建议）。

**2.29.1 MPEG-4 SL**

在MPEG-2系统规格（ISO / IEC 13818：2000 / ITU-T建议H.222.0）节描述了2.11 MPEG-4 SL的封装。

**2.29.2 VC1**

在SMPTE VC1说明书描述在MPEG-2系统VC1有效载荷的封装。

**2.29.3 H.264**

ITU-T建议。H.222.0修正案3描述了MPEG-2系统的H.264封装。

**2.30 主机行为的静止图像捕获方法2**

**题：**（经由共享同步端点即视频流和静止图像捕捉）我的设备仍支持图像捕获方法2。什么是格式协商和静态图像的采集过程中的预期主机的行为呢？

**回答：**主机将使用静止探测器和提交进行控制以通知所希望的静止图像捕获参数（格式，帧大小和压缩指数）的装置中，并获得

捕获缓冲器和有效载荷传输大小即发送的静止图像的匹配的那些参数时，该设备将使用。静止探头和提交谈判将至少执行一次，但可能多次，之前的静态图像捕捉。的次数是取决于应用的。主机需要执行的至少一个

VS\_STILL\_COMMIT\_CONTROL ( SET\_CUR ) 操作之前的第一静止图像传输。以静止图像拍摄时，主机将决定之前还 **最佳** 备用接口通过比较设定为同步端点 **dwMaxPayloadTransferSize** 从静止探头和提交与控制 **wMaxPacketSize** 每个同步端点描述符。虽然这似乎是合乎逻辑的最大可用带宽同步仍然图像传输过程中被使用，如果设备不能按这个速度仍然提供图像数据，这是不恰当的。因此，主机将选择最匹配的备用接口 **dwMaxPayloadTransferSize** 由所述装置提供的值。当静止图像被捕获，主机会：

1. 暂停视频流在端点（如果已激活）
2. 选择最佳的备用接口，用于静止图像拍摄
3. 触发转移
4. 采集静态图像字节
5. 恢复以前的备用接口选择
6. 重新启动视频流（仅在以前的活动）

如果装置正积极流主机不会改变备选接口设置 和 最佳备用接口的用于静止图像捕捉的带宽预留将等于（或小于）当前替代设置的。

如果静止图像捕获所需要的带宽不可用，主机应适度降低，选择代表当时可用的最大带宽的备用接口。在正常情况下，这不应该小于预留给任何活动的视频流传输带宽。该设备将只知道可用带宽的替换接口选择的被动接受者，所以在低带宽条件下的主机行为不在这里讨论。