



嵌入式Linux 下的USB 驱动开发

www.farsight.com.cn

今天的内容

1. Linux内核对USB规范的支持
2. USB主机驱动程序
3. USB设备驱动程序
4. USB HUB驱动程序
5. OHCI HCD

Linux对USB规范的支持

- ✓ USB-通用串行总线是目前使用最广泛的外部总线
- ✓ USB是采用单一的主从设备通信模式。总线上的唯一的主机负责轮询设备并发动各种传送，因此实现简单，成本相对低廉
- ✓ USB从拓扑上讲类似于主机同外设之间点对点连接，设备连接汇集于集线器上
- ✓ USB最新的规范是USB2.0版本，定义了三种传输速率

∅ Low speed	——	1.5Mbps
∅ Full speed	——	12Mbps
∅ High speed	——	480Mbps

USB规范简介

- ✓ 1994年，一个由四个行业伙伴(Compaq、Intel、Microsoft和 NEC)组成的联盟开始制定 USB 协议。该协议最初的目的是将 PC 与电话相连并提供容易扩展和重新配置的 I/O 接口。1996年 1月，发表了 USB 规范的第一个版本，1998年 9月发表了后续版本(版本 1.1)。这个规范允许 127台设备同时连接到一起，总的通信带宽限制为 12 Mbps。后来，又有三个成员(Hewlett-Packard、Lucent 和 Philips)加入了这个联盟。2000年 4月，发表了 USB 规范的 2.0版本，它支持高达 480 Mbps 的传输率。
- ✓ USB总线特点
 - ⊘ 易用，用来作设备扩展非常方便
 - ⊘ 廉价，并可以支持很高的传输速率
 - ⊘ 支持实时应用，如音视频传输
 - ⊘ 协议灵活，可以支持同步和异步数据的混合传输
 - ⊘ 容易集成到各种主机和常用设备当中
 - ⊘ 为各种不同设备提供统一的标准接口
 - ⊘ 提供了动态增删设备及热插拔能力

USB设备基本概念

✓ USB规范中规定的标准概念由Linux USB core来实现处理

✓ 端点endpoints

∅ 端点是USB总线传输最基本的概念，一个端点可以单方向传输数据。可以把端点看作是一个单方向的管道。端点有4种类型

✓ 接口interfaces

∅ 若干端点可以捆绑起来，成为一个端口。端口可以作为完整的逻辑设备连接，例如鼠标设备、键盘设备。

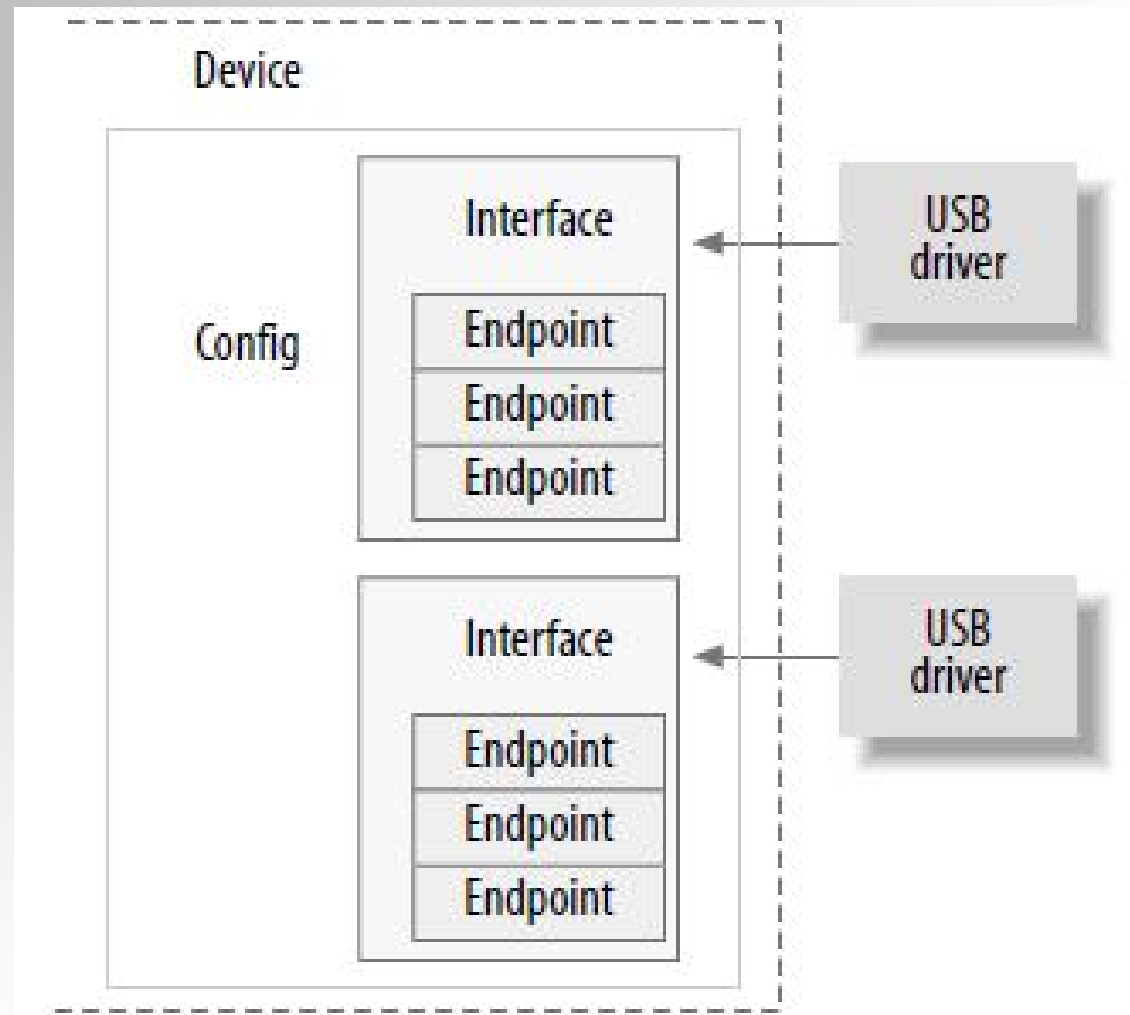
∅ 需要指出的是，一个硬件设备可能包含多个逻辑设备

∅ 接口可以有多个预设值，用来指定不同的参数

✓ 配置configurations

∅ 接口捆绑起来成为配置。一个USB设备可以在不同的配置之间切换，一次只能激活一个配置

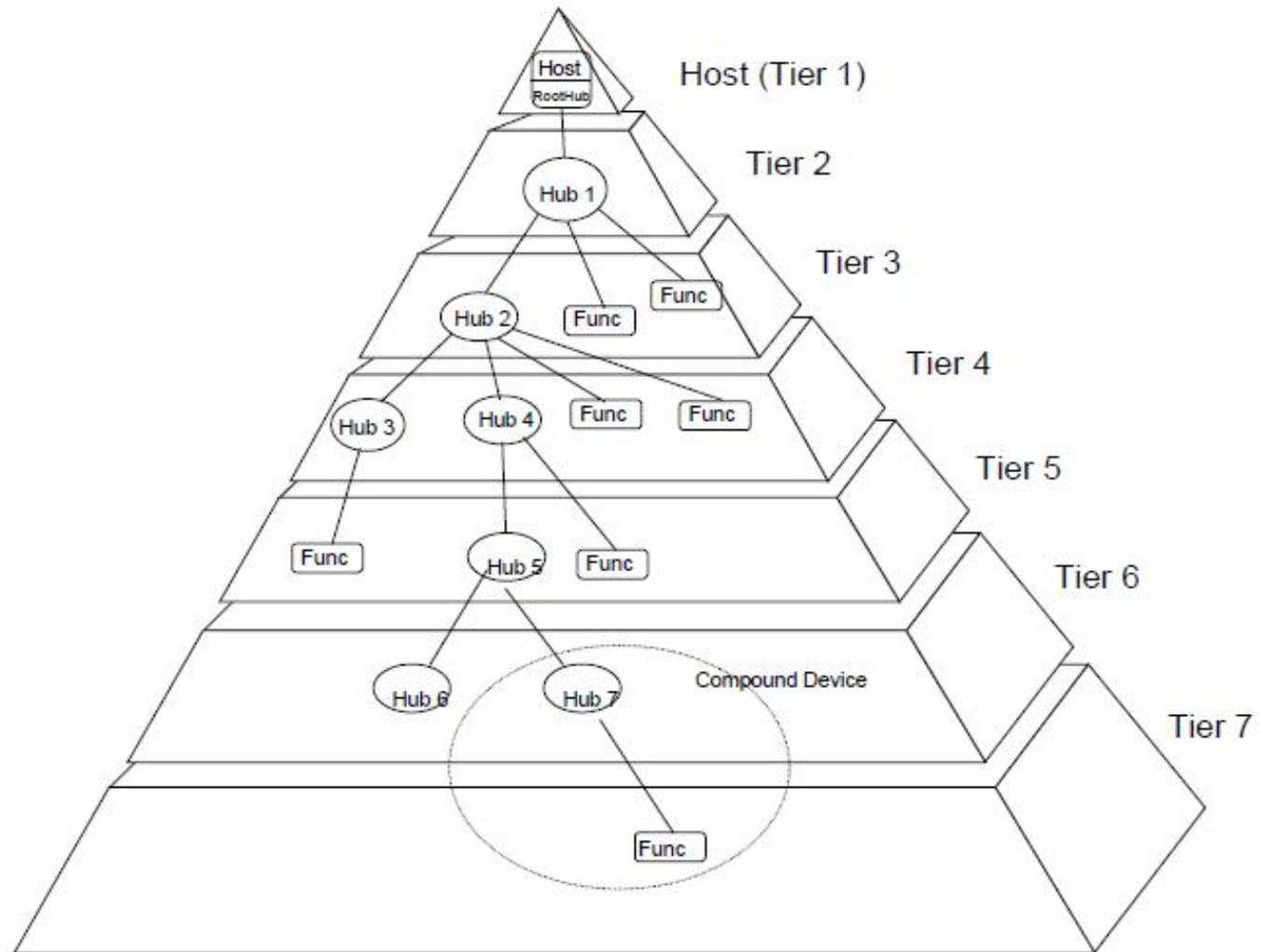
USB设备基本概念示意图



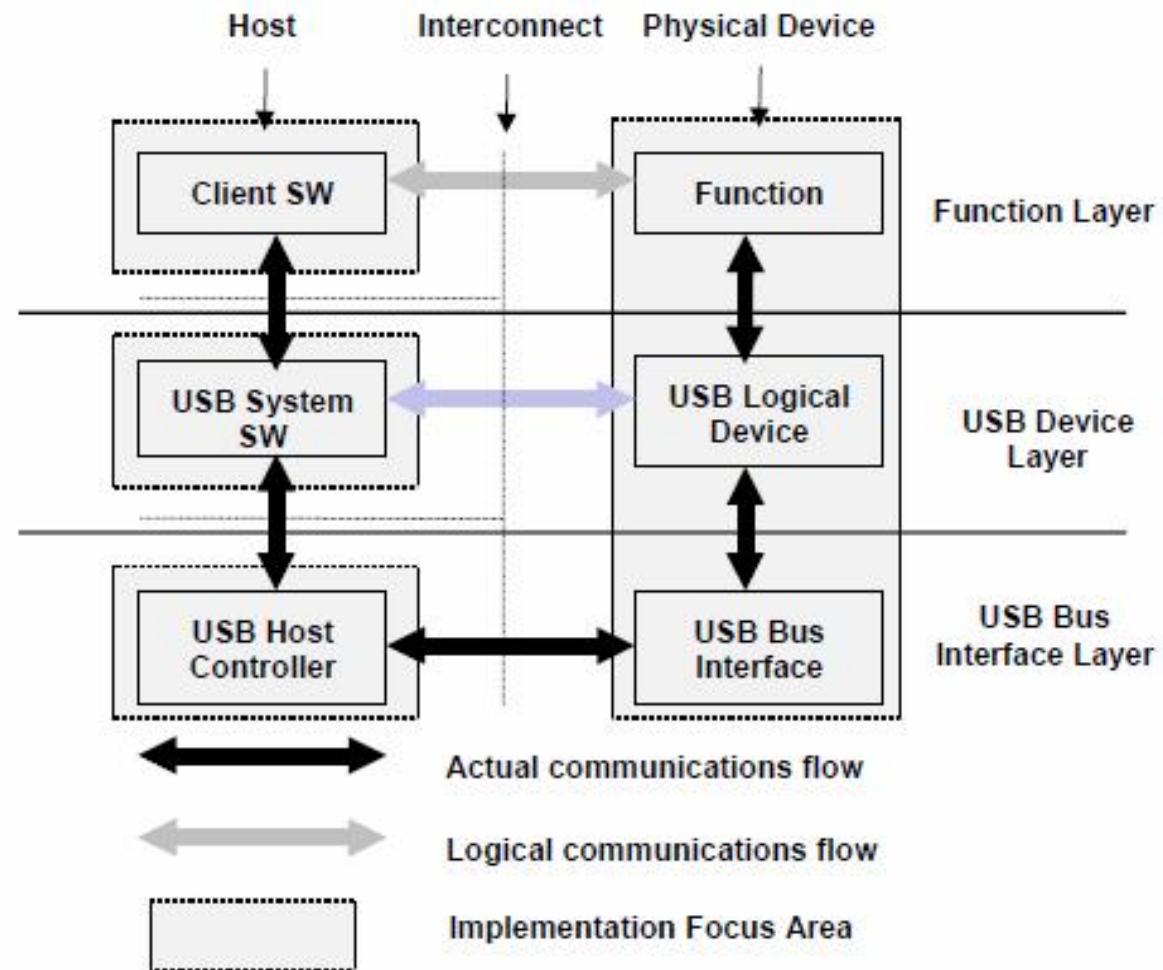
√ USB 总线中的通信可以使用下面四种数据传输类型中的任意一种：

- ∅ **控制传输**：这些是一些短的数据包，用于设备控制和配置，特别是在设备附加到主机上时。
- ∅ **批量传输**：这些是数量相对大的数据包。像扫描仪或者 SCSI 适配器这样的设备使用这种传输类型。
- ∅ **中断传输**：这些是定期轮询的数据包。主控器会以特定的间隔自动发出一个中断。
- ∅ **等时传输**：这些是实时的数据流，它们对带宽的要求高于可靠性要求。音频和视频设备一般使用这种传输类型。

USB拓扑结构



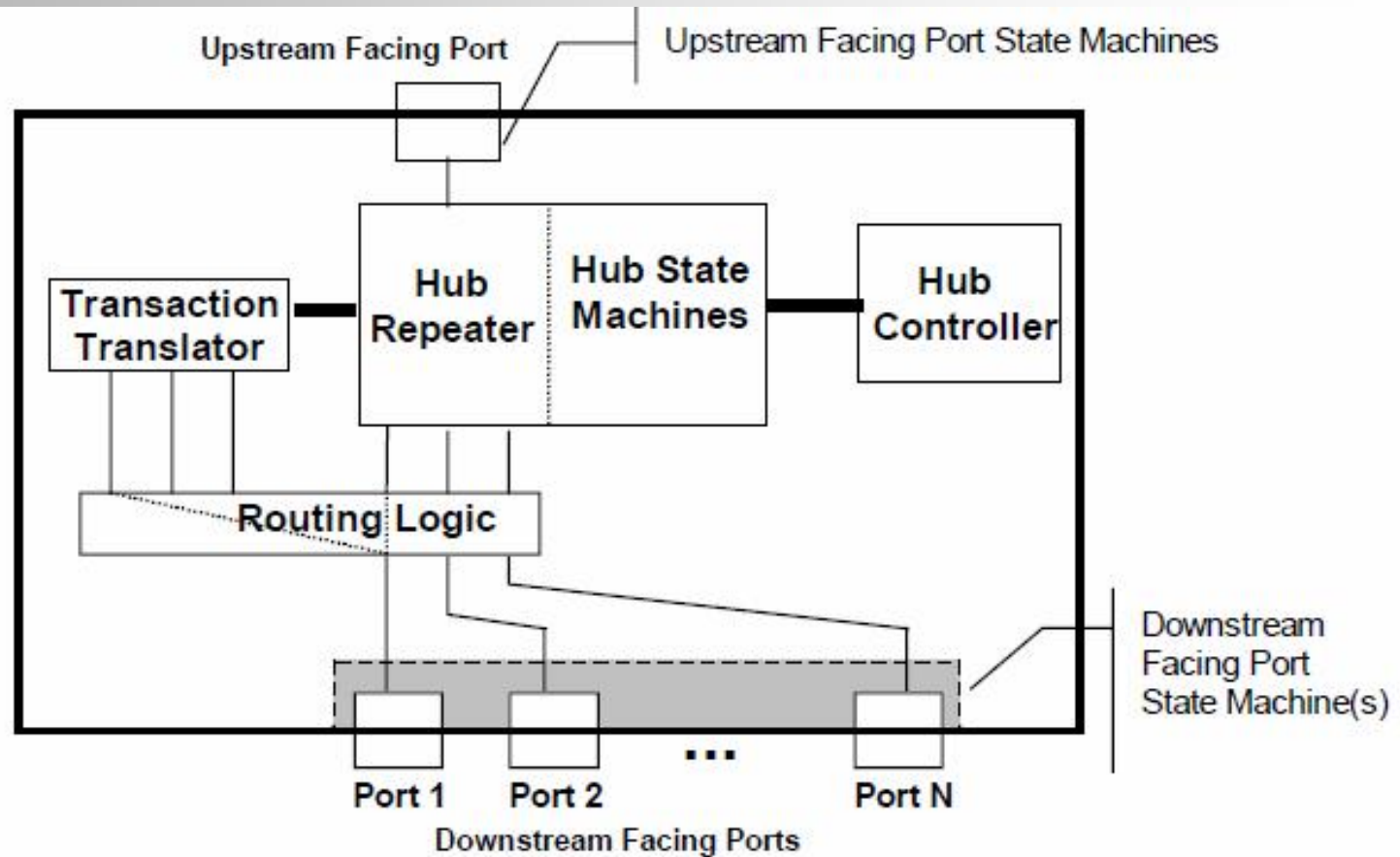
USB层次化连接示意图



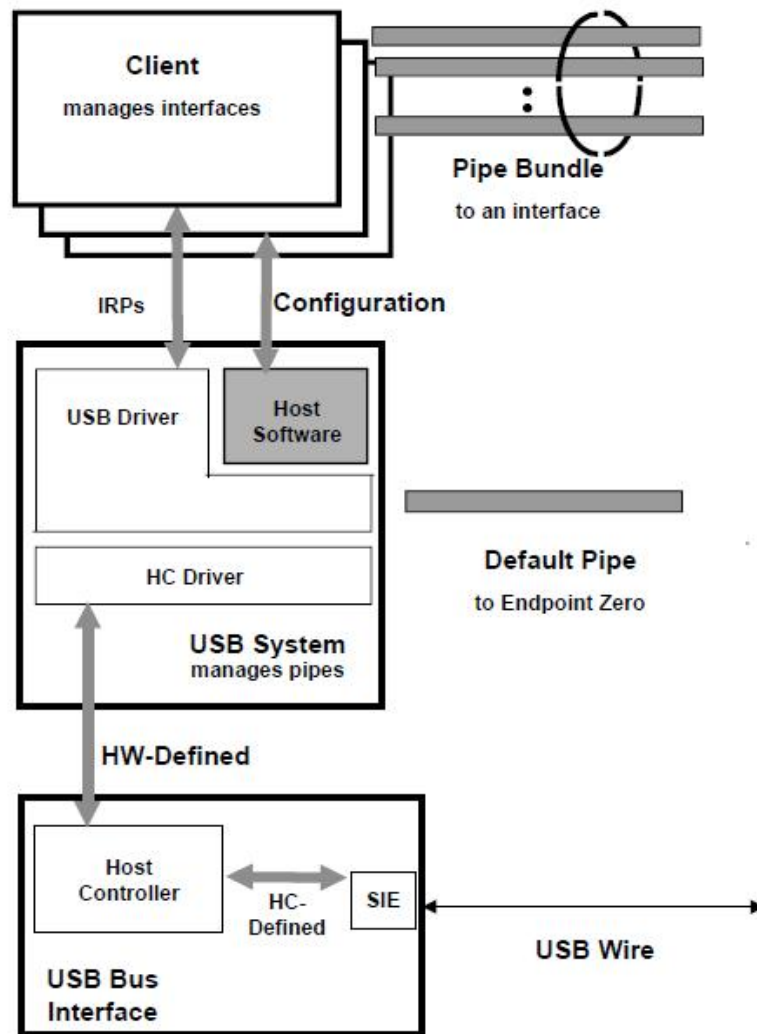
不使用urb进行USB传输

- ✓ 当我们仅仅需要进行简单的USB传输，我们可以绕过urb的复杂机制
- ✓ `int usb_bulk_msg(struct usb_device *usb_dev, unsigned int pipe, void *data, int len, int *actual_length, int timeout);`
- ✓ 该函数用来创建并向一个指定的设备发送bulk包，并且函数一直阻塞直到传输完成。
- ✓ `int usb_control_msg(struct usb_device *dev, unsigned int pipe, __u8 request, __u8 requesttype, __u16 value, __u16 index, void *data, __u16 size, int timeout);`
- ✓ 该函数用来创建并向指定的设备发送control包

USB hub结构



主机驱动程序结构



主机控制器驱动程序

√ 主机控制器驱动程序负责USB总线通信基本的职责

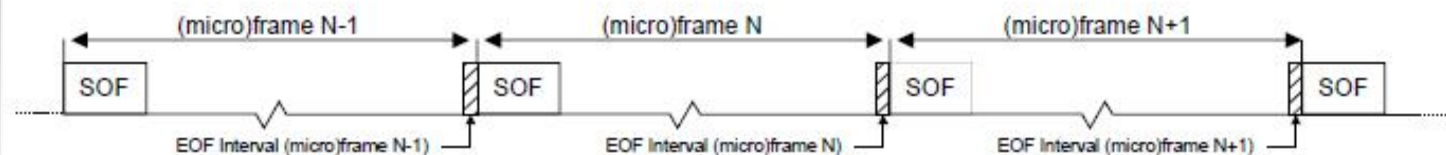
- ∅ 处理USB状态。管理状态并报告状态信息
- ∅ 数据串行/解串行，将设备申请传输的数据转换成比特流
- ∅ 生成frame或microframe
- ∅ 处理数据传输的请求
- ∅ 处理USB总线协议
- ∅ 进行差错检测和控制
- ∅ 处理能源管理请求，把总线置为suspended状态以及响应wakeup事件
- ∅ 提供root hub功能，让设备可以连接到主机控制器

处理USB状态

- ✓ 主机控制器包含一系列状态
- ✓ 当有设备插入一个端口时，主机控制器的状态必须反映这个变化
- ✓ 当USB设备通过resume信号发送一个wakeup申请时，主机控制器必须通报

串行/解串行和数据成帧

- ✓ 由于USB使用串行信号来传输数据，USB总线主机和设备都必须实现SIE
- ✓ 将USB总线上时间划分成时间片是主机控制器的责任
- ✓ SOF标志一个时间片 (micro) frame 的开始

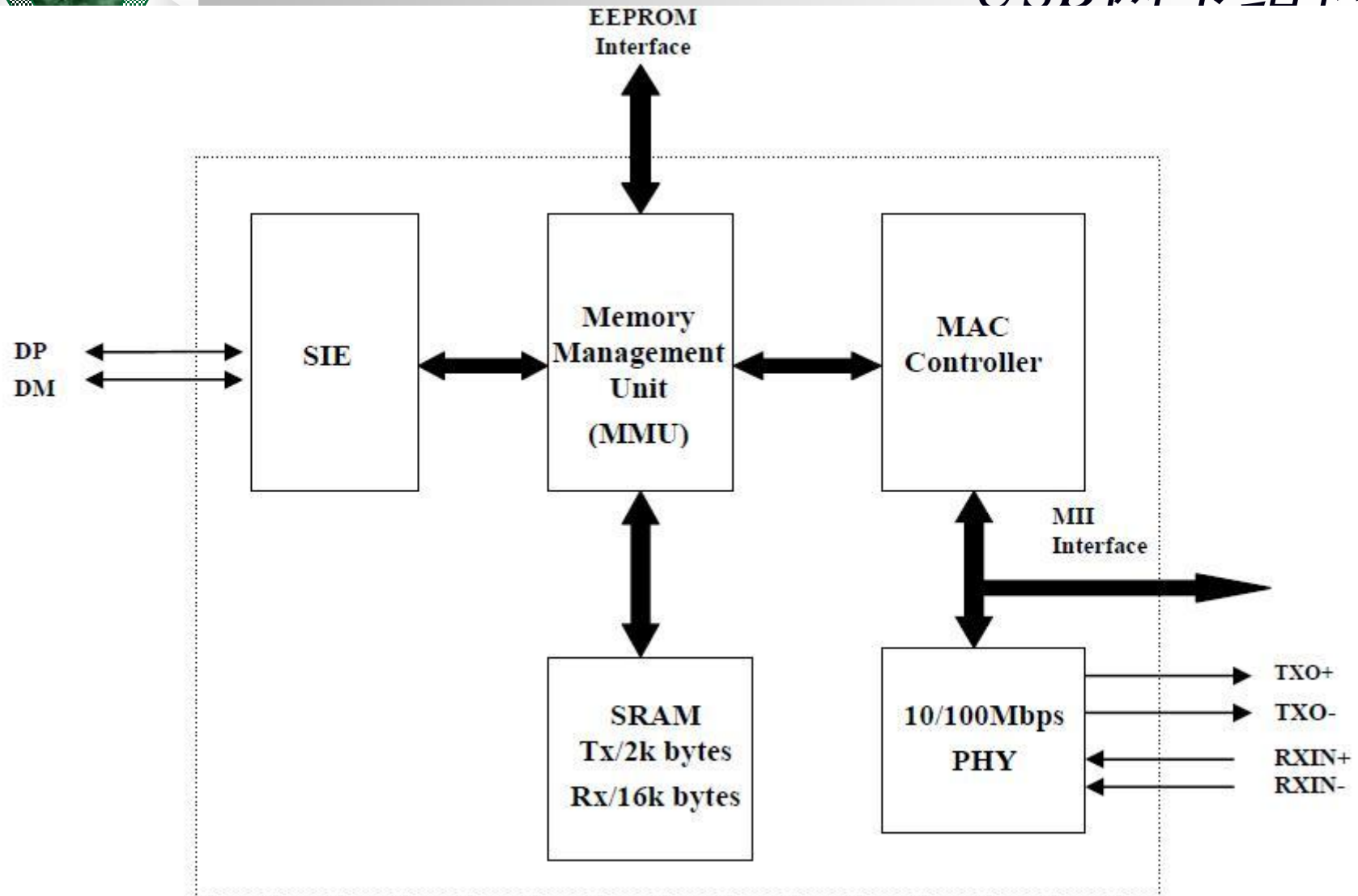


主机控制器协议和数据处理

- ✓ 主机控制器既要向总线发送数据，也要从总线上接收数据并传递给USB子系统
- ✓ 数据传送过程中主机控制器要解析USB模型层次化的协议。包括在传送时插入协议包头信息，接收时剥离并解析协议包头
- ✓ 根据协议将数据包分发到USB core，或做出处理

√ RTL8150简介

- ∅ 一个中断类型输入端点
 - ü 最大数据包长度8字节
- ∅ 一个批量类型的输入端点
 - ü 最大数据包长度64字节
- ∅ 一个批量类型的输出端点
 - ü 最大数据包长度64字节



Linux驱动开发理论课程内容

- ✓ **Linux设备驱动和内核概论**
- ✓ **设备驱动程序工作流程**
- ✓ **字符设备驱动程序**
- ✓ **块设备驱动程序**
- ✓ **Proc文件系统**
- ✓ **文件系统驱动程序**
- ✓ **Framebuffer驱动程序**
- ✓ **Usb设备驱动程序**
- ✓ **网络设备驱动程序**

Linux驱动开发实验课程内容

- ✓ 使用gnu开发编译环境
- ✓ 熟悉驱动程序加载到内核中的方式
- ✓ 数码管的字符驱动程序
- ✓ 键盘驱动程序
- ✓ A/D,D/A驱动
- ✓ Usb Mass Storage驱动
- ✓ Video4Linux驱动
- ✓ 文件系统驱动
- ✓ 以太网卡驱动



www.FarSight.com.cn

谢谢！