



第二讲：连接和枚举

USB技术 应用与开发

演讲人：蔡亮

CONTENTS

01 | 主设备与从设备

02 | 连接与检测

03 | 枚举过程

04 | 控制传输

01

主设备与从设备

主设备与从设备

USB通讯的两端分别称为：HOST（主设备/USB主机）和Device（从设备/USB设备）

常见的主设备：计算机



USB主设备一般有以下功能：

- 检测USB设备的插拔动作
- 管理主从通讯之间的控制流
- 管理主从通讯之间的数据流
- 记录主机状态和设备动作信息
- 控制主控制器和USB设备间的电气接口

主设备与从设备



USB设备按功能分类：
集线器和功能设备

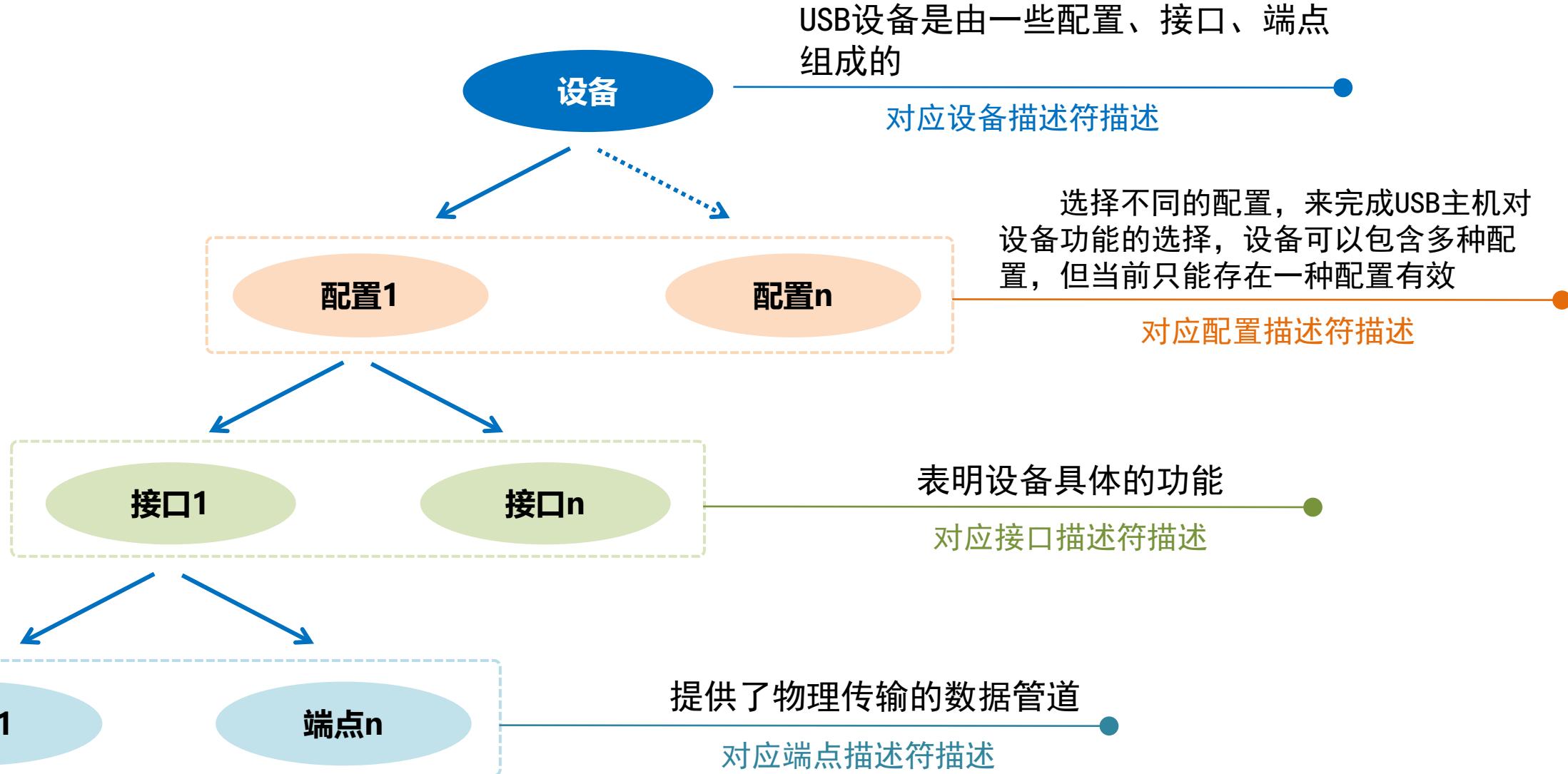
集线器：

- 用于扩展USB主机的USB端口
- 结构上有一个上行端口，多个下行端口
- 支持级联，系统中最多5个集线器（不包括主机的根集线器）
- 支持速度转换

功能设备：

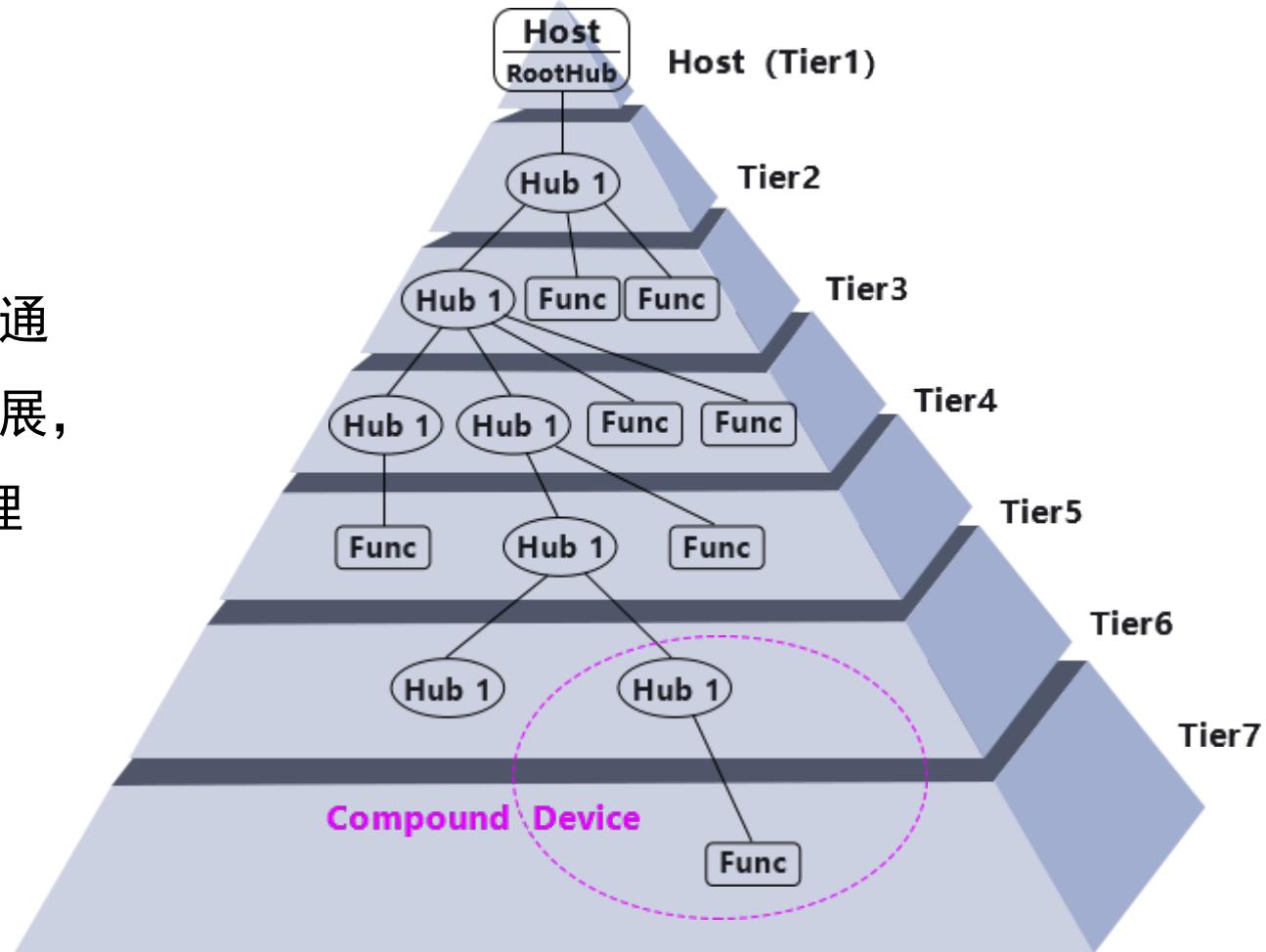
- 一个独立的外围设备，可以是单一功能也可以是多功能的合成设备
- 内部包含有描述自身功能和资源需求的配置信息

主设备与从设备



主设备与从设备

在USB系统中，一般 USB主机通过集线器（HUB）进行 USB设备扩展，以层次性的星型拓扑结构进行物理连接。



主设备与从设备



主机和USB设备之间通讯物理通道： 主设备分配的地址/默认地址0 + 从设备固有端点号。



主机和USB设备之间时间长度单位： 帧 / 微帧。



主机和USB设备之间协议处理基本单位： 事务处理。

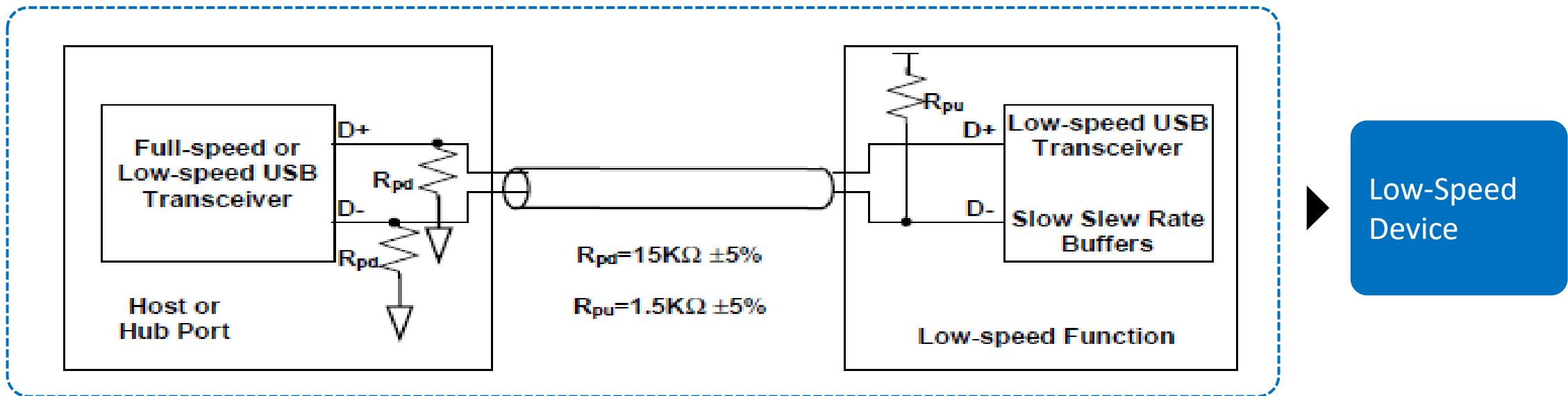


主机和USB设备之间通讯： 在基本单元 “事务” 中， 主机总是发起者，并且和设备交互应答方式进行通讯。

02

连接与检测

连接与检测



1

USB主机端口在 D+ 和 D- 上都有 $15\text{k}\Omega$ 电阻接地

2

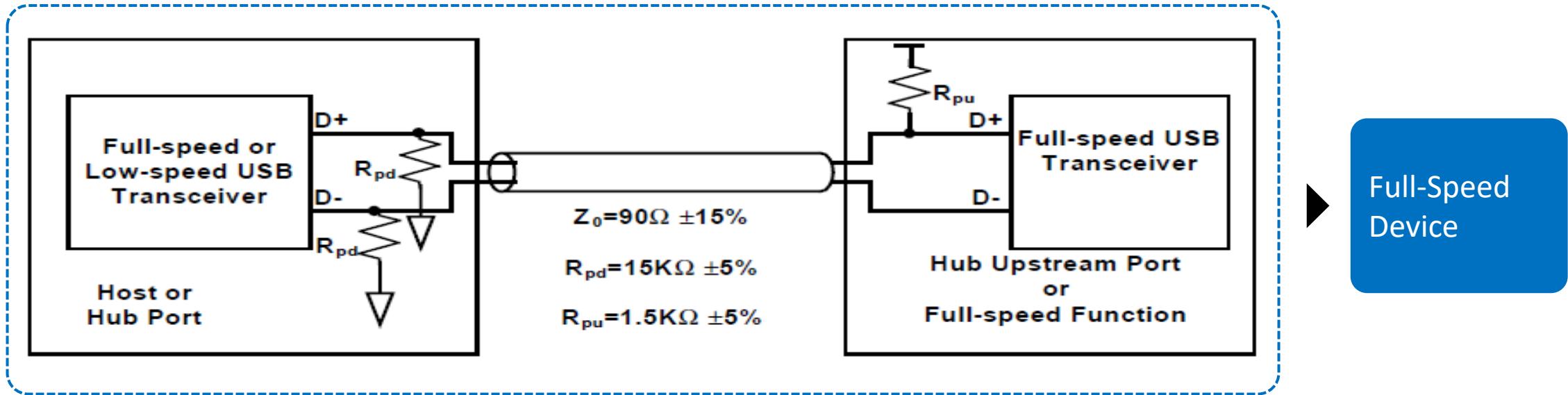
低速USB从设备在 D- 上连有 $1.5\text{k}\Omega$ 的电阻到 3.0V-3.6V电压

3

没有连接：主机端口检测到 D+ 和 D- 电压都近地 0V

低速设备连接：主机端口检测到 D- 电压约 3V, D+ 电压近地 0V

连接与检测



1

USB主机端口在 D+ 和 D- 上都有 $15k\Omega$ 电阻接地

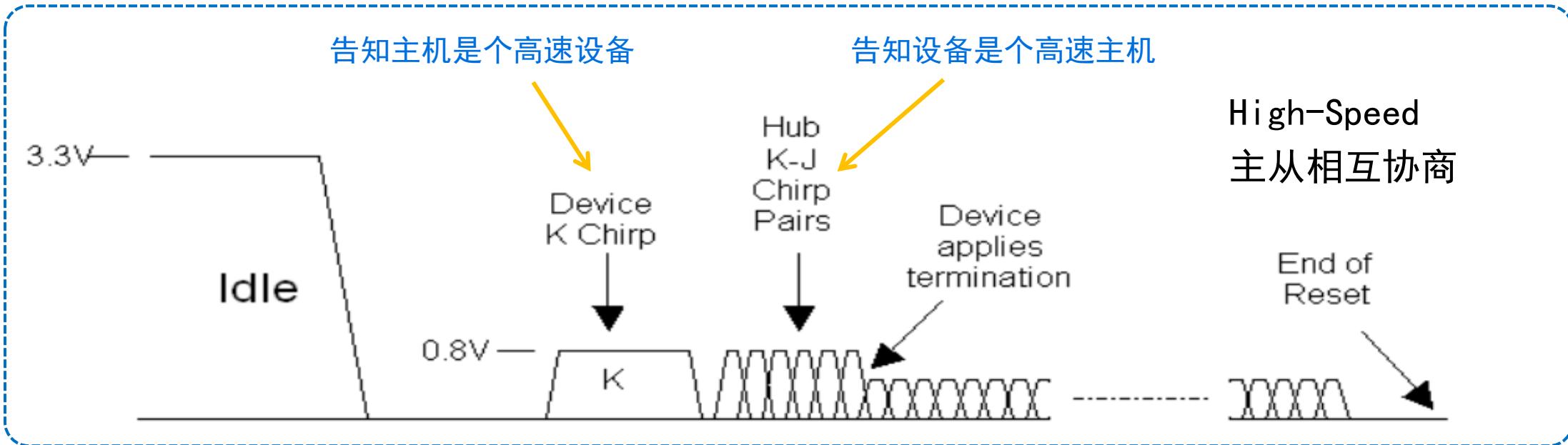
2

全速USB从设备在 D+ 上连有 $1.5k\Omega$ 的电阻到 3.0V-3.6V电压

3

没有连接：主机端口检测到 D+ 和 D- 电压都近地 0V

全速设备连接：主机端口检测到 D+ 电压约 3V, D- 电压近地 0V



- 高速设备先以全速设备结构和主机连接，它们之间做双向检测
- 主机输出总线复位信号期间，USB设备以是否可以产生 Chip K信号来表明高速或全速身份
- 在Chip K信号后，主机是否发生KJ序列来表明高速主机身份或者是全速主机身份
- 匹配到高速主机和高速设备后，USB设备断开D+上的 $1.5\text{k}\Omega$ 的上拉电阻，连接D+/D-上的高速终端电阻，进入默认的高速状态，否则以全速状态通讯



- 设备断开
- 设备连接
- 低速设备
- 全速设备
- 高速设备



- 主机检测到 D+ 和 D- 上近地状态 (0V) , 并持续2. 5us以上
- 主机检测到 D+ 或 D- 上有高电平 (3V) , 并持续2ms以上
- 主机检测到 D- 上高电平
- 主机检测到 D+ 上高电平 (有可能是高速设备)
- 主机检测到 D+ 上高电平, 然后通过一系列协商握手信号确认双方身份
(双向检查)

- 总线几种状态

常见的几种状态	描述
正常工作	总线上存在周期性SOF包
总线复位	总线维持SE0状态 > 10ms
总线挂起	总线无活动 > 3ms

常见几种变化	触发点
无连接 -> 连接	D+/D- 上出现高电平>2ms
正常 -> 挂起	J状态保持 > 3ms
挂起 -> 正常 (唤醒)	出现 K状态信号并持续一段时间

03

枚举过程

枚举过程



枚举： USB主设备向USB从设备通过获取各种描述符，从而了解设备属性，知道是什么样的设备，并加载对应的USB类、功能驱动程序，然后进行后续一系列的数据通信。

- 特点：**
- 主设备连接识别从设备必须的过程
 - 由多个控制传输构成
 - 经过地址0（缺省地址）到其他地址（主设备分配地址）的通讯
 - 对于挂载多个USB从设备的系统，主设备是逐一进行枚举操作

枚举过程

枚举过程

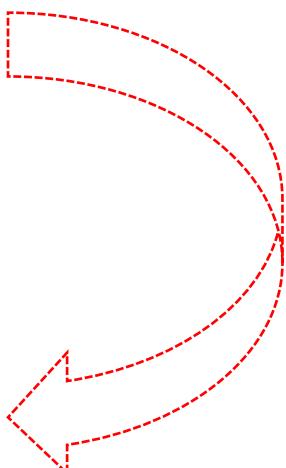
- USB设备上电（一般从USB口取电）并连接到USB总线
- 主机检测到总线上有设备连接
- 主机会等待至少100ms用于连接的机械、电气特性稳定
- 主机执行总线复位至少10ms，并得到USB设备通讯速度
- 主机驱动总线空闲至少10ms用于做恢复时间
- 主机发出获取设备描述符请求（缺省地址）
- 主机为从设备分配唯一设备地址，后续通讯用此地址
- 主机以新地址发出获取设备描述符请求
- 主机以新地址发出获取配置描述符请求，获取设备全部配置
- 主机分析获取的描述符信息，并做相应记录和处理
- 主机发送设置配置请求，为从设备选择一个合适的配置

枚举过程

```
ConfigDescr [ ] =  
{  
    配置描述符  
    +  
    接口描述符  
    +  
    类描述符  
    +  
    端点描述符  
    ...  
};
```

- 设备描述符：第一个需要获取的描述符，长度固定18字节。

- 配置描述符：描述了设备特定的配置，提供了当前配置下设备的功能接口，供电方式，耗电等信息。是一个配置的集合，集合长度不固定，包含了配置描述符、接口描述符、类定义描述符、端点描述。

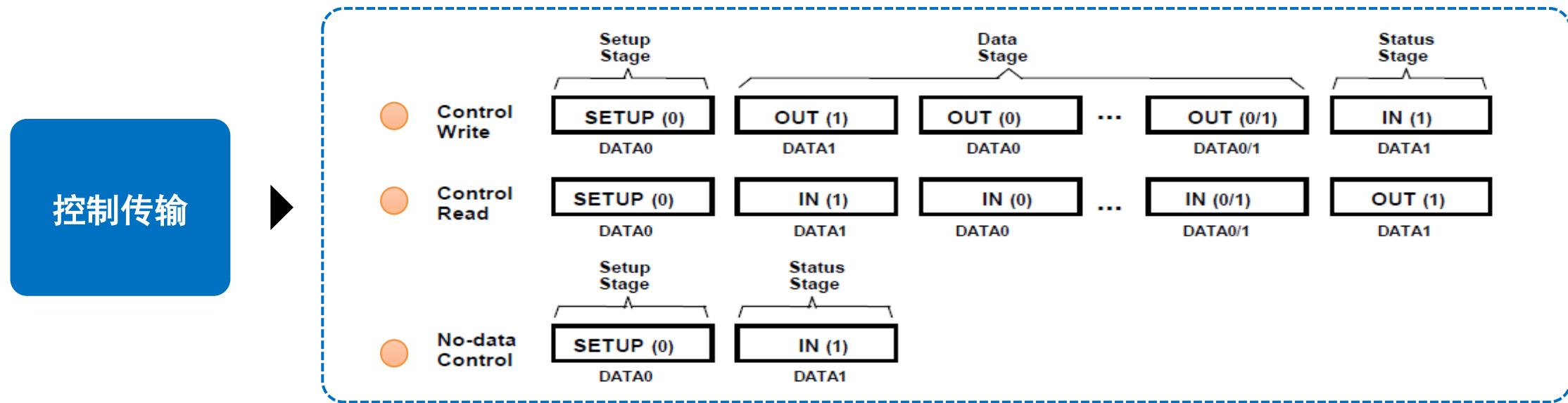


多个接口时，继续添加
接口+类+端点描述

04

控制传输

控制传输



- 1 是所有USB从设备必须支持的传输方式，固定使用端点0通讯
- 2 控制传输的方向是双向的，既可以主机下传数据给设备，又可以从设备上传数据
- 3 多用于主设备和从设备进行信息、功能、状态等方面的获取和修改

建立阶段 — Setup 事务

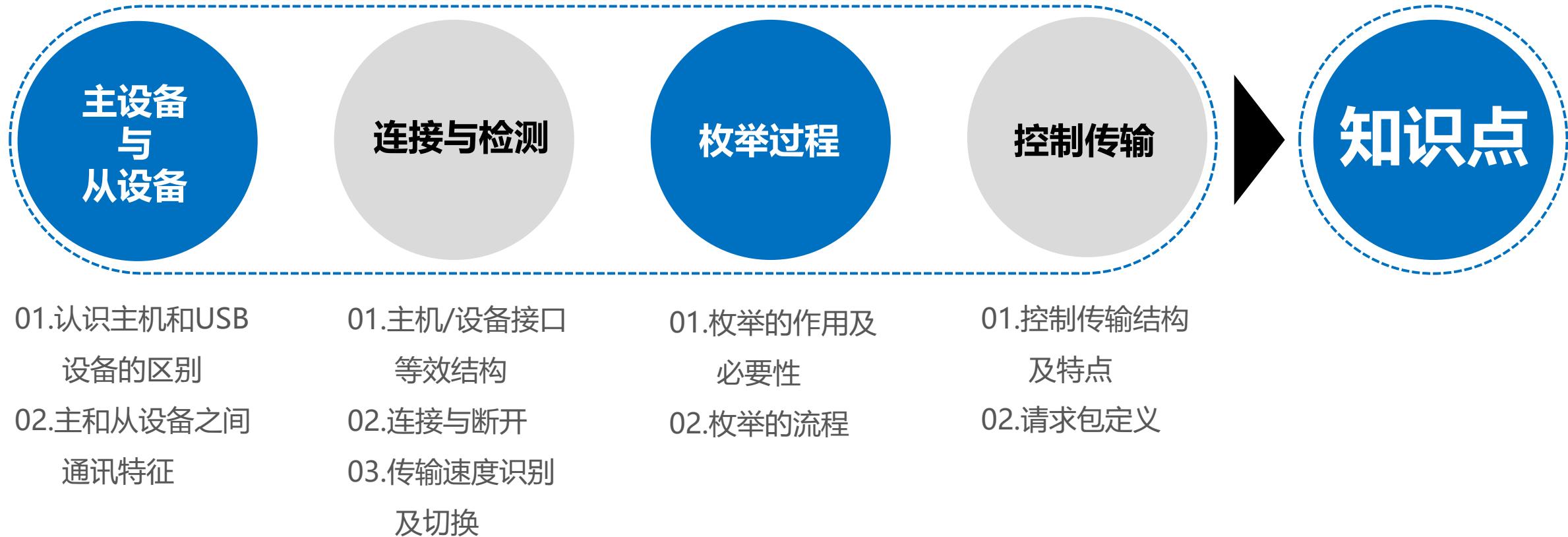
- 固定的8字节结构数据 
- 中间的数据包的 PID - DATA0
- 应答包必须为 ACK

状态阶段 — 一个IN/OUT 事务

- 数据包长度为 0
- 数据包的 PID - DATA1
- 执行的事务和数据阶段事务相反
- 没有数据阶段，执行IN事务

数据阶段 — 一个或多个连续的IN/OUT事务

- 不是必要阶段
- 控制读 — 连续IN事务， 控制写 — 连续OUT事务
- 事务数据包从 PID_DATA1开始， 然后进行 1-0-1-0-… 交替翻转
- 应答包支持 ACK / NAK / STALL





Thank you

感谢观赏



微信公众号

<http://wch.cn>
tech@wch.cn
025-84730668

控制传输——Setup Stage



控制传输中的 Setup Stage 部分（Setup事务），主机发出8字节命令请求，格式如下：



控制传输结构——Control Write



→ 126.0	SETUP	03	00	(8 byte) 21 09 00 02 00 00 01 00	Class_Cmd:0200
→ 126.1	DATA0				
← 126.2	ACK				
→ 128.0	OUT	03	00	(1 byte) 03	
→ 128.1	DATA1				
← 128.2	ACK				
→ 130.0	IN	03	00	(0 byte)	
→ 130.1	DATA1				
→ 130.2	ACK				
→ 130.3	VCK				
→ 130.4	DATA1				
→ 130.5	TR	02	00	(0 byte)	

← Setup stage

← Data stage

← Status stage

控制传输结构——Control Read



→ 15.0	SETUP	03	00	(8 byte) 80 06 00 01 00 00 12 00	Get_DevDesc:00
→ 15.1	DATA0				
← 15.2	ACK				
→ 17.0	IN	03	00		
← 17.1	DATA1			(8 byte) 12 01 10 01 00 00 00 08	
→ 17.2	ACK				
→ 19.0	IN	03	00		
← 19.1	DATA0			(8 byte) 3C 41 05 21 50 03 01 02	
→ 19.2	ACK				
→ 21.0	IN	03	00		
← 21.1	DATA1			(2 byte) 00 01	
→ 21.2	ACK				
→ 22.0	OUT	03	00		
→ 22.1	DATA1			(0 byte)	
← 22.2	ACK				
← SS' S	VCLK				
→ SS' I	DATA1			(0 byte)	
→ SS' O	ONI	03	00		
→ SS' S	VCLK				

← Setup stage

← Data stage

← Status stage

控制传输结构——No-data Control



→ 12.0	SETUP	00	00	(8 byte) 00 05 03 00 00 00 00 00	Set_Address:03
→ 12.1	DATA0				
← 12.2	ACK				
→ 14.0	IN	00	00		
← 14.1	DATA1			(0 byte)	
→ 14.2	ACK				
→ 14.3	VACK				
→ 14.4	DATA2			(0 byte)	
→ 14.5	ACK				

← Setup stage

← Status stage