

USB Stack - Design der Systemschnittstelle

Überblick

- ❑ Der USB-Standard allgemein
- ❑ Die Implementierung

Überblick

- Der USB-Standard allgemein
 - ◆ Ziele
 - ◆ Topologie
 - ◆ Pipes und Deskriptoren
 - ◆ Transfertypen
 - ◆ Hot Plug'n'Play
- Die Implementierung

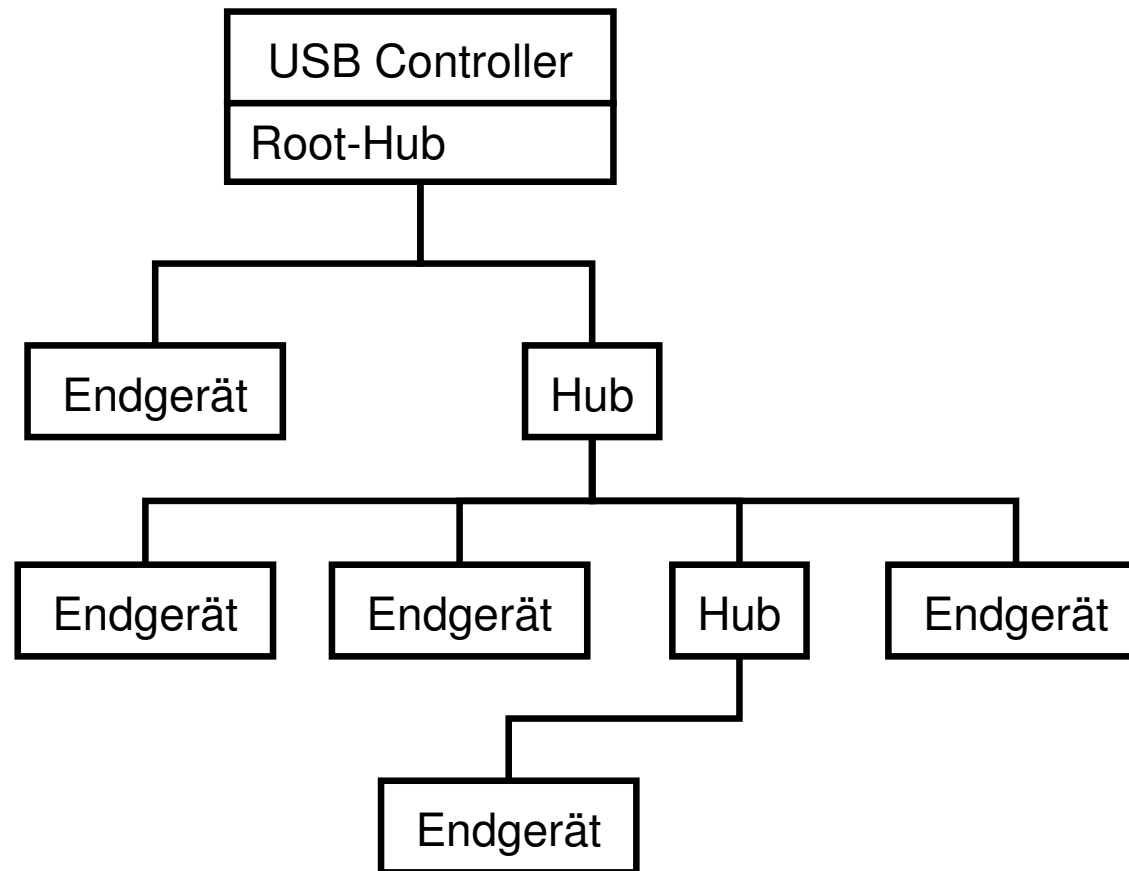
Probleme bei alten Schnittstellen

- 2 Drucker - 1 paralleler Port
- Neustart erforderlich beim Anstecken
- Benutzer muss Übertragungsraten einstellen
- viele Interrupts im Hostsystem werden belegt
- Adapter werden benötigt

Ziele des USB

- ❑ Einfache Bedienung
 - ◆ hot plug'n'play
 - ◆ einheitliche Stecker
- ❑ Stromversorgung von „kleinen“Geräten
- ❑ Unterstützung von allen üblichen Peripheriegeräten

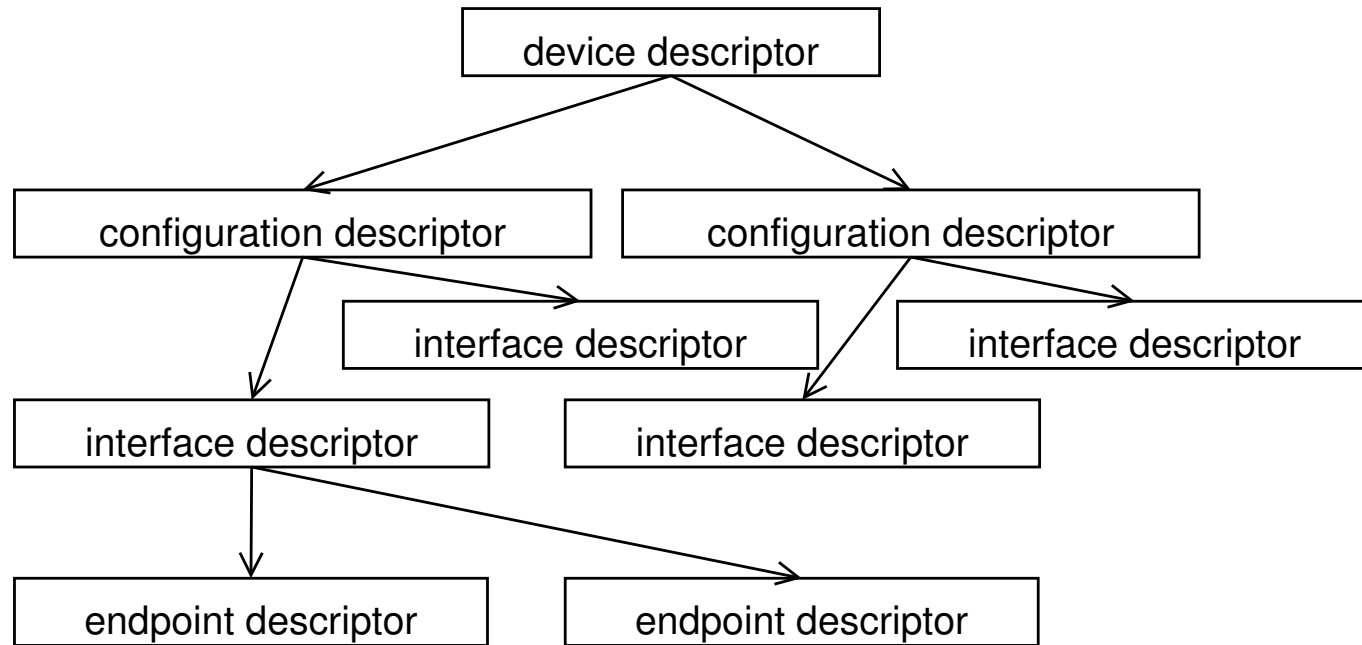
Topologie



Pipes und Endpoints

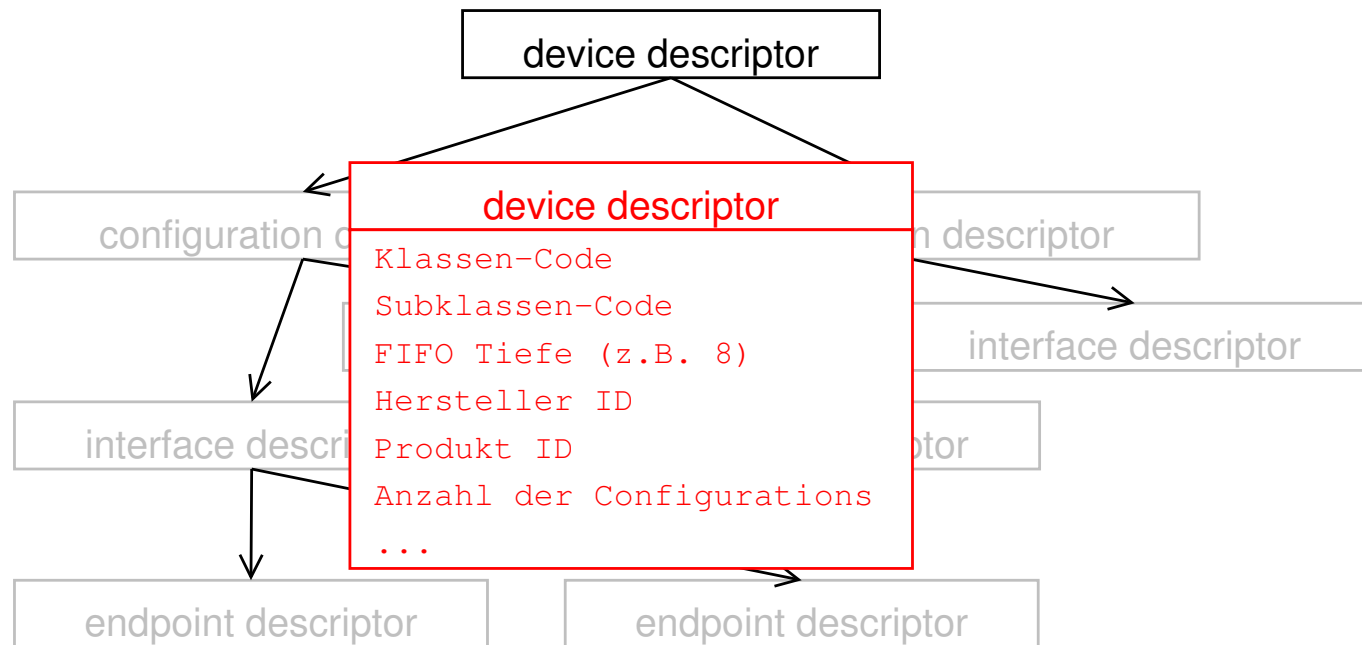
- ❑ Function(USB-Gerät) hat mehrere logische Kanäle
- ❑ Treiber kann Daten mit Pipes an Endpoints der Geräte senden und empfangen
- ❑ Jede Pipe hat einen Transfertyp und eine Richtung

Deskriptoren

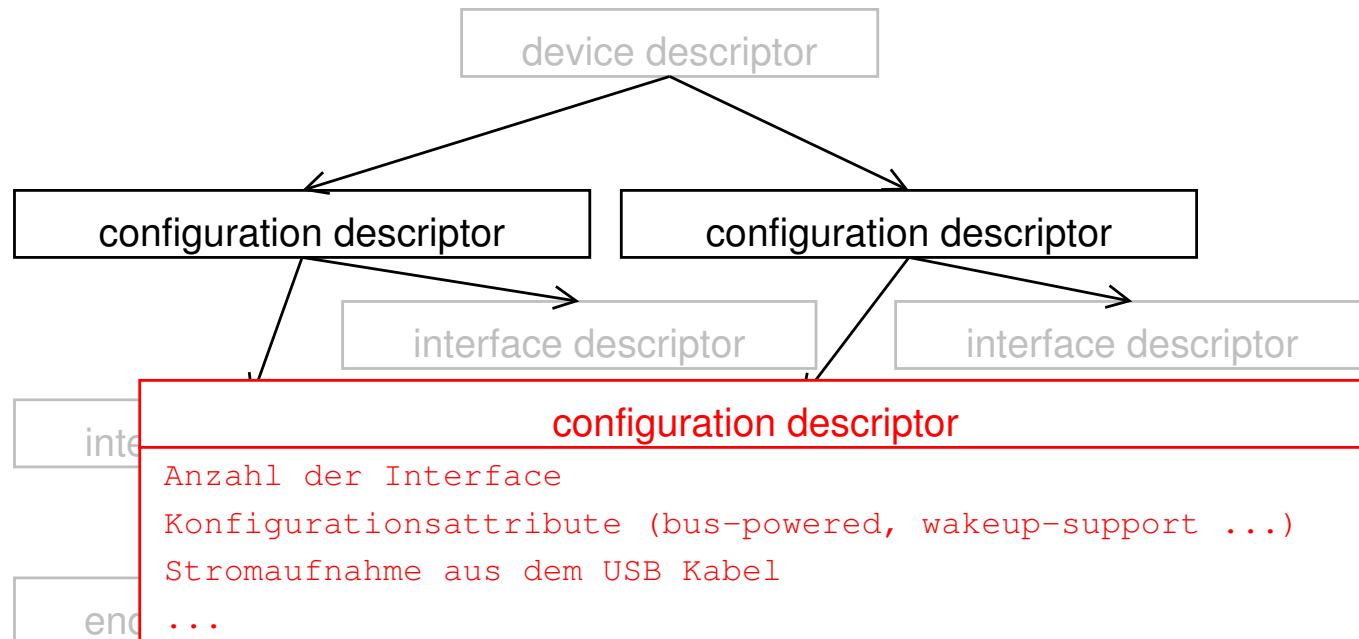


□ Hierarchie der Deskriptoren

Device-Descriptor

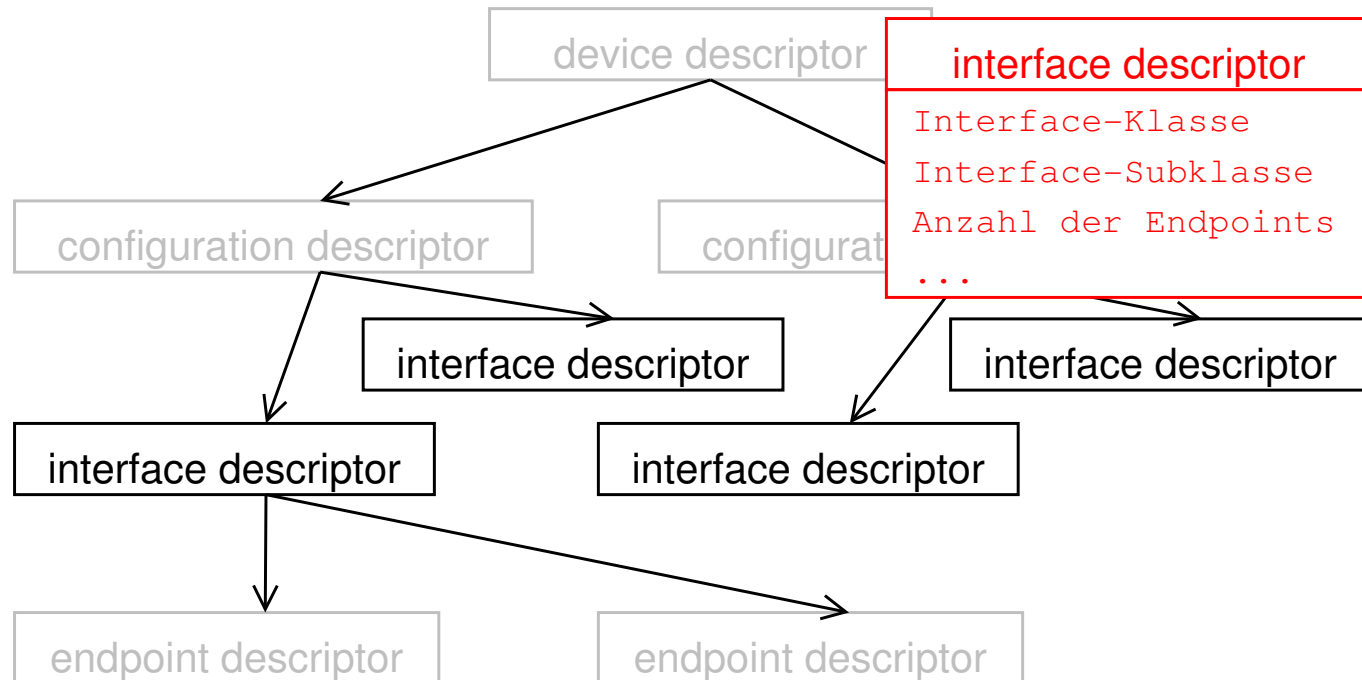


Configuration-Descriptor



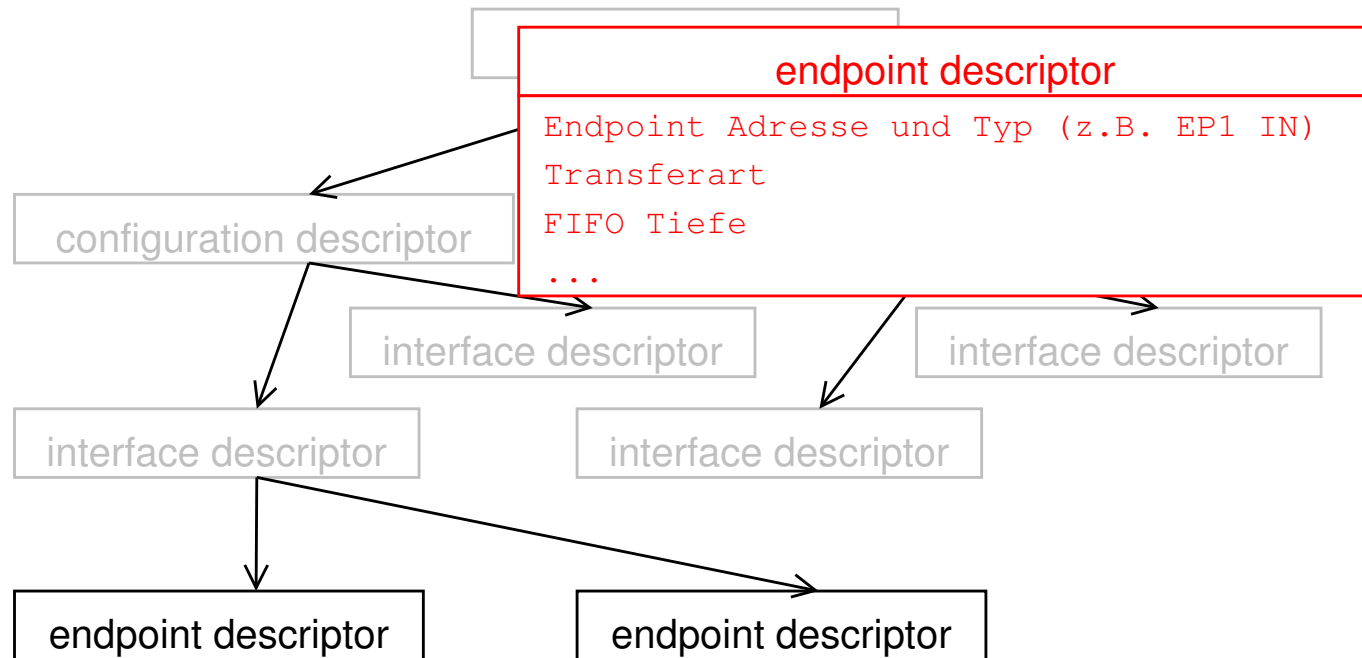
- Treiber kann Configuration auswählen

Interface-Descriptor



□ Treiber kann Interface auswählen

Endpoint-Descriptor



- Treiber kann aus Endpointdescriptoren Kommunikationsparameter auslesen

Transfertypen

- ❑ übertragen wird in 1-ms Frames
- ❑ Single-Master-Bus: Host ist Master, Functions passiv
- ❑ mehrere Transfers unterschiedlichen Typs während eines Frames
- ❑ verschiedene Zwecke benötigen verschiedene Transfer-typen

Lautsprecher, Maus, Scanner, HD

Control-Transfer

- ❑ schickt Request zum Steuern des Geräts
spezifiziert in USB- oder Gerätespezifikation
- ❑ abhängig vom Request werden Daten nachgeschickt
- ❑ auch holen von Daten z.B. Descriptoren möglich
- ❑ spezielle Richtung „Setup“

Interrupt-Transfer

- ❑ in regelmässigem Abstand ein Transferversuch
- ❑ Abfrage von Neuigkeiten
Mausbewegung, Tastatur, Hub
- ❑ einzige Möglichkeit zum Nutzdatenaustausch bei LowSpeed
- ❑ kann natürlich auch senden

Isochronous-Transfer

- ❑ garantierte Übertragungsrate
- ❑ wird beim Einrichten bereits reserviert
- ❑ Fehlerkontrolle nicht möglich
- ❑ für Functions mit konstanter Rate
Lautsprecher, webcam, headsets

Bulk-Transfer

- ❑ zum Auffüllen von Frames benutzt
- ❑ alle Daten die nicht dringend sind *Drucker, Scanner, Speicher-Sticks*
- ❑ effizient

Hot Plug 'n' Play

- ❑ ein Hub meldet Veränderungen an Ports über Interruptpipe
- ❑ der USB-Treiber enumeriert neue Geräte
- ❑ passender Treiber wird geladen
- ❑ bei Abstecken wird der Treiber informiert
Gerät muss kein hot plug'n'play unterstützen

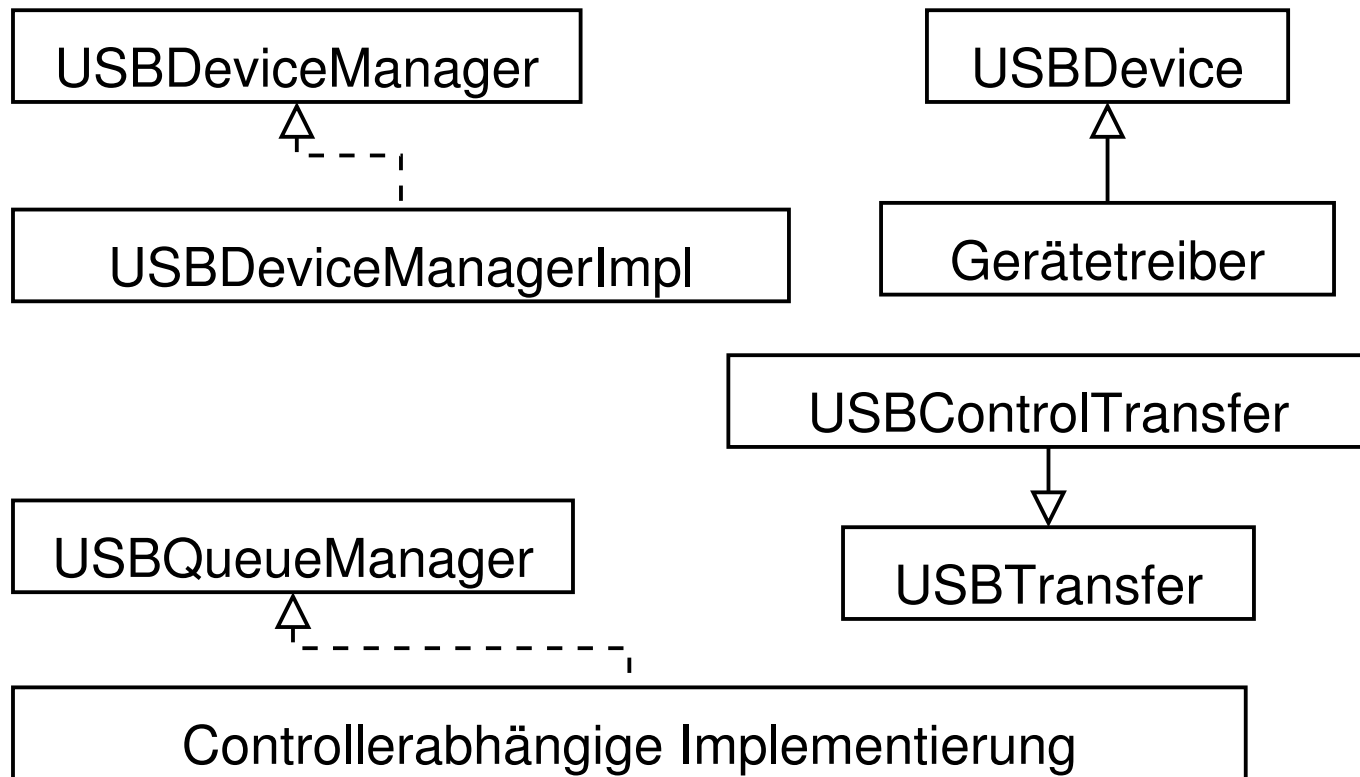
Überblick

- Der USB-Standard allgemein
- Die Implementierung
 - ◆ Aufbau
 - ◆ Die Schnittstelle für den Treiber
 - ◆ Das Interface zum Controllertreiber
 - ◆ Ausstehende Entwicklungen

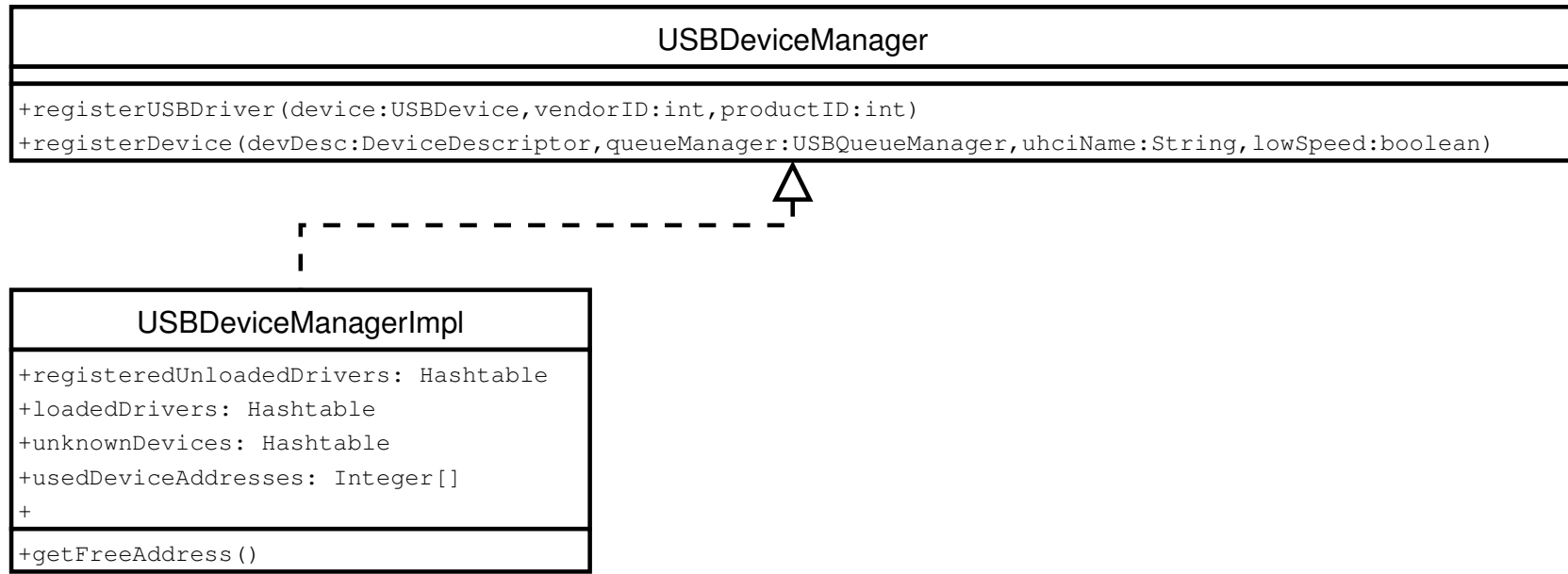
Implementierung

- Zwei wesentliche Aufgabe
 - ◆ Verwaltung der Geräte
 - ◆ Anbieten einer Schnittstelle für Gerätetreiber
- unabhängig von Controllertreiber
UHCI, OHCI

Klassenaufbau



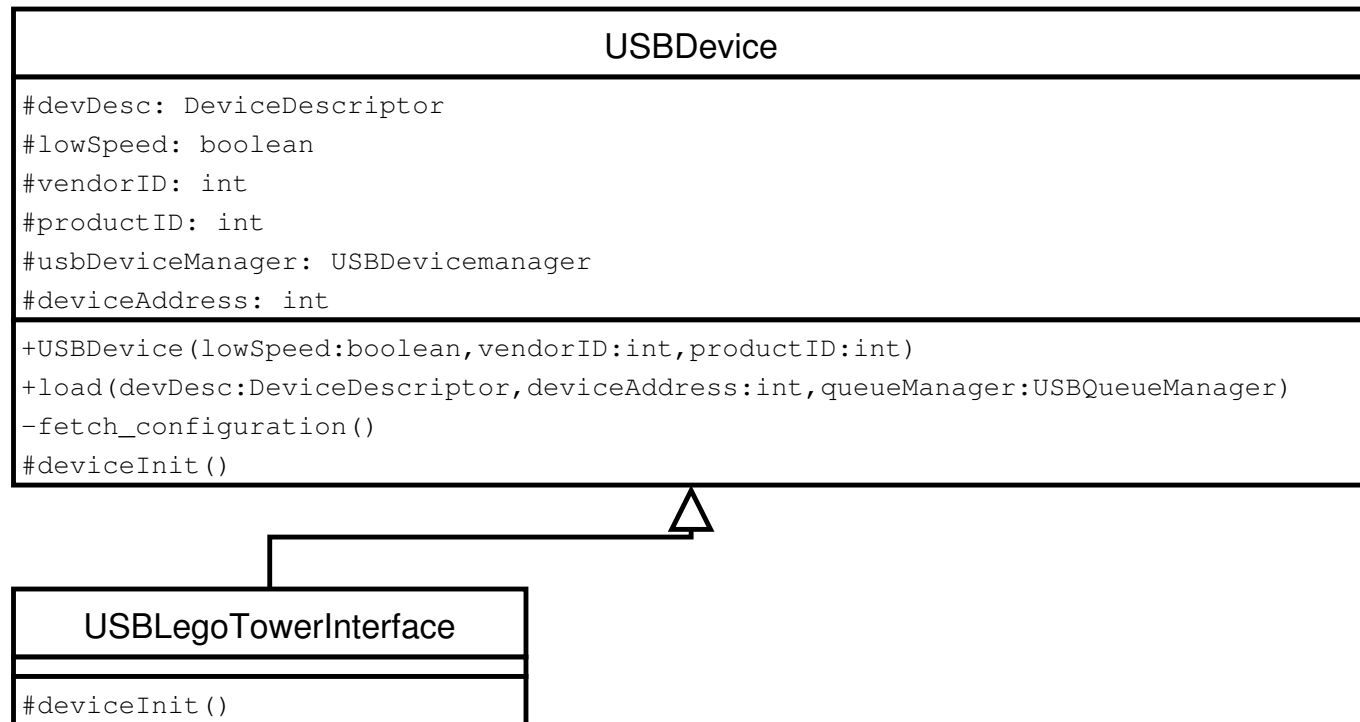
USBDeviceManager



Treiberschnittstelle 1

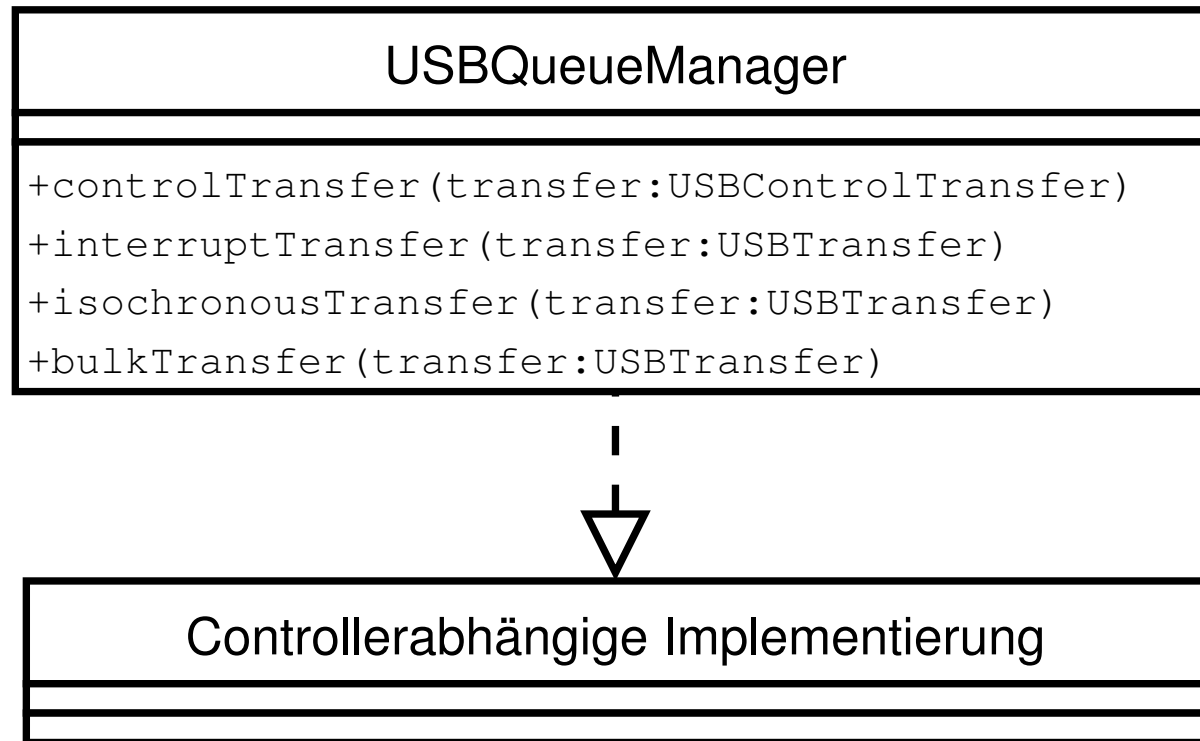
- USBDeviceManager Kernstück des USB-Treiber
 - ◆ enumeriert Geräte
 - ◆ verwaltet Geräte und Treiber
 - ◆ bei bekannten Geräten wird der Treiber geladen

USBDevice



- USBDevice als Basisklasse von USB-Treibern

USBQueueManager



Treiberschnittstelle 2

- ❑ Beim Laden des Treibers bekommt dieser Adresse, Devicedescriptor und Queuemanager übergeben
- ❑ USBQueueManager wird von Treiber benutzt um Transfers vorzunehmen
 1. erstellt Memory, bei OUT Transfers auch “befüllt“
 2. erstellt USBTransfer Objekt
deviceAddress, endPoint, fifoSize, mem
 3. ruft den QueueManager auf um den Transfer vorzunehmen

Controllerinterface

- ❑ USBQueueManager deklariert die Übertragungsaufforderungen
- ❑ diese muss der Controllertreiber definieren
- ❑ Controllertreiber instanziiert den USB-Stack, sowie alle Endgerätetreiber

Ausstehende Entwicklungen

- mehrmaliges Laden desselben Treibers
- Treiberzugriff auf zugeordnete Function beschränken
- USB - Requests in USBDevice definieren
- blockieren und erfolgsmeldung beim Queuemanage-
raufruf
- geeignete Konzepte für Isochronous- und Interrupt-
Transfer

Quellen und weiterführende Informationen

Literatur

- [1] Dipl.-Ing. Hans Joachim Kelm. „USB 1.1, Universal Serial Bus“. Franzis Verlag, 2000, Poing.
- [2] Die USB 1.1 Spezifikation:
<http://www.usb.org/developers/docs/usbspec.zip>
veröffentlicht von Compaq Computer Corporation,
Intel Corporation, Microsoft Corporation, NEW Corporation am 23.9.1998

[3] Die USB-Stack Implementierung für jx im Rahmen des AKBP II