



TECHNISCHE  
UNIVERSITÄT  
DRESDEN

Fakultät Informatik Institut für Technische Informatik, Professur für VLSI-Entwurfssysteme, Diagnostik  
und Architektur

# ZWISCHENPRÄSENTATION DIPLOMARBEIT

Entwurf und Implementierung eines modularen USB-Stacks  
für eingebettete Controller ohne Betriebssystem

Uwe Pfeiffer

Dresden, 19.1.2015



DRESDEN  
concept  
Entwickler aus  
Wissenschaft  
und Kultur

# Gliederung

- USB
- Motivation
- Ziele
- konkrete Umsetzung
- Ausblick

## 01 USB

- serielles Bussystem
- Verbindung zwischen Peripheriegerät und Host
- Host-gesteuert
- Plug-and-play
- Stromversorgung für Peripheriegeräte möglich
- verschiedene Geschwindigkeiten

# Datenfluss

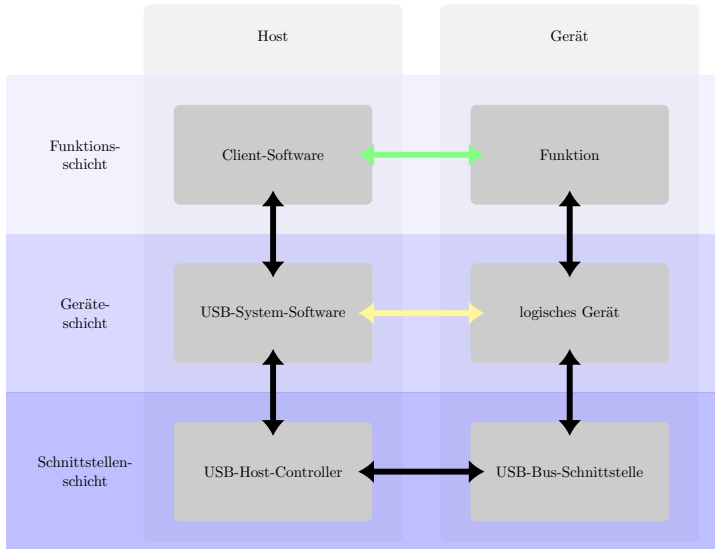


Abbildung: Überblick über Datenfluss

# Deskriptoren

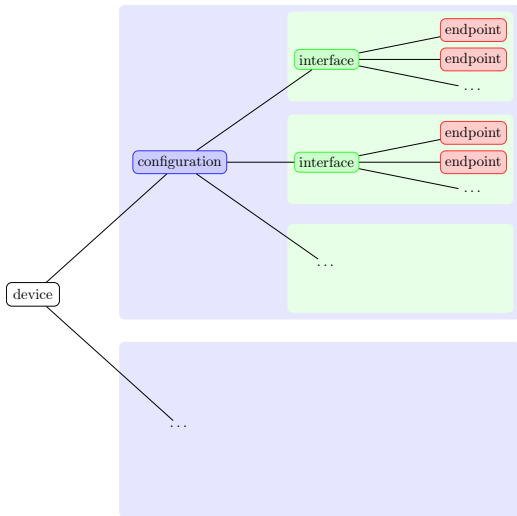


Abbildung: Hierarchie der Standard-Deskriptoren

## 02 Motivation

- USB kompliziert/komplex
- jeder Hersteller mit eigenem Beispielcode
- viel selbst zu implementieren

## 03 Ziele

- einfache und schnelle Erstellung von Firmware
- zur Verwendung wenig Wissen über USB notwendig
- mit geringem Aufwand auf andere Hardware portierbar
- einfache Verwendung von USB-Klassen (auch mehrere)
- möglichst voller Umfang von USB-2.0 verwendbar
- objektorientiert mit C++

## 04 konkrete Umsetzung

- Standard-Befehle und Erweiterbarkeit für andere Befehle
- Verwendung beliebiger Deskriptoren
- Daten senden/empfangen
- Vorgaben beim Stromverbrauch einhalten
- getrennte Schicht für hardwarespezifischen Teil



## USB-Kern

- Template-Method-Pattern
- abstrakte Basisklasse beinhaltet quasi kompletten USB-Stack
- konkrete Unterklasse implementiert nur Zugriff auf spezifische Hardware
- Zugriff auf Deskriptoren über Schnittstelle
- Datenfluss erfolgt über Endpunktobjekte

# Daten senden/empfangen

- Zugriff über Endpunkte
- Datenpuffer versus Callback-Funktionen
- Schwierigkeit Kapselung/DMA

# Deskriptoren

- Arrays versus Baum
- dynamisch versus statisch

# Deskriptoren

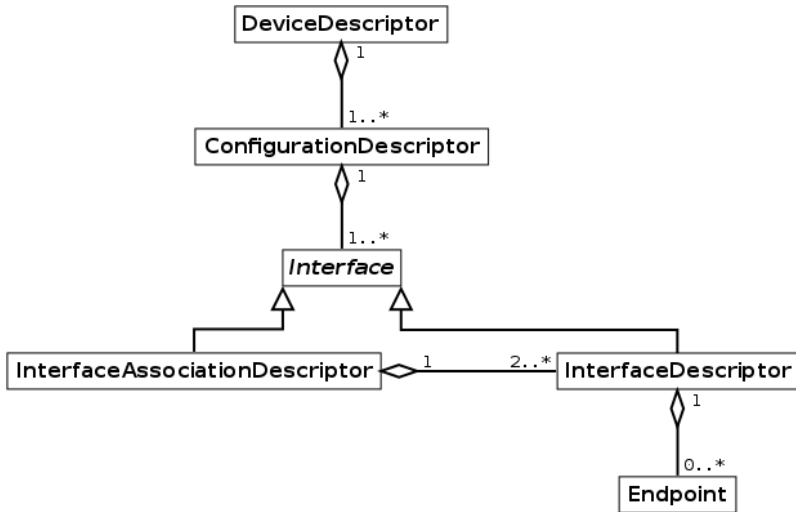


Abbildung: schematischer Aufbau der Deskriptoren

## 05 Ausblick

- Klassen
  - Erweiterung des vorhandenen Deskriptorbaums
  - Registrierung eigener Funktionen für spezifische Funktionen
  - Verarbeiten der Daten an den Endpunkten
  - Bereitstellen eigener Schnittstelle für Daten?
- weitere Unterstützung beim Erstellen der Deskriptoren?
- Portierung auf weitere Mikrocontroller
- so weit wie möglich USB-2.0-kompatibel