

## A5. Digitale Schnittstellen

A5.1 Schnittstellen für Computerperipherie 

A5.2 Universal Serial Bus

A5.3 FireWire

Literatur:

Messmer/Dembowski: PC-Hardwarebuch Kapitel 32

## Hardware-Schnittstellen

- Standard zur Verbindung von Geräten
  - z.B. Computer und Peripherie
- Prinzip 1: Parallele Übertragung
  - so viele Datenleitungen wie Wortbreite in Bits
- Prinzip 2: Serielle Übertragung
  - Bits werden in Folge übertragen
  - Synchroner Übertragung:
    - » Takt- oder Handshake-Information separat auf eigener Leitung übertragen
  - Asynchrone Übertragung:
    - » Synchronisationsinformation in der Nachricht eingebettet (Start- und Stopbits)

## Klassische PC-Peripherieschnittstellen

- Parallele Schnittstelle (Centronics)

- Druckeranschluss beim IBM-PC
- 36-adriges Kabel, davon 18 genutzt
- modernisierte Variante: IEEE-1284



parallel (DB25S) + Game port

- Serielle Schnittstellen (RS-232 bzw. V.24)

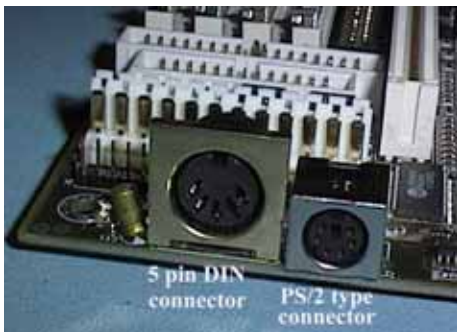
- 25-polige und 9-polige Stecker
- Standard unterstützt asynchronen und synchronen Austausch
  - » Üblich: asynchroner Austausch, 9-polige Stecker
- Betriebsmodi:
  - » Simplex: Leitung nur in einer Richtung genutzt
  - » Halbduplex: Leitung abwechselnd in verschiedenen Richtungen genutzt
  - » Voll duplex: gleichzeitige Nutzung in verschiedenen Richtungen (Zwei Kabel oder logische Kanäle)



seriell (DB25P und DB9P)

## Verschiedenste Schnittstellen

- Z.B. Anschluss von Tastatur und Maus:
  - DIN- und Mini-DIN-Stecker (PS/2)
- Z.B. zum Anschluss schneller Peripherie
  - SCSI-Interface
- Insgesamt: unübersichtlich, unflexibel



<http://members.iweb.net.au/~pstorr/pcbook>



<http://not.hellalame.com/nes>

## A5. Digitale Schnittstellen

A5.1 Schnittstellen für Computerperipherie

A5.2 Universal Serial Bus



A5.3 FireWire

Literatur:

Don Anderson (MindShare), Universal Serial Bus System  
Architecture, 2nd ed., Addison-Wesley 2001

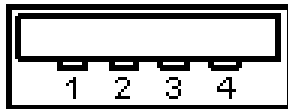
<http://www.beyondlogic.org/usbnutshell>

## Universal Serial Bus (USB)

- Entwickelt durch Industriekonsortium (u.a. Intel)
  - verbreitet seit ca. 1995
  - Derzeit zwei Generationen im Markt: Version 1.1 und 2.0
- Vereinheitlichung von Peripherie-Schnittstellen
- Entwurfsziele
  - Einheitliche Steckverbinder für alle Endgeräte (von der Tastatur bis zum CD-Brenner)
  - Unterstützung für Vervielfachung von Anschlüssen (Baum-Topologie, bis zu 127 Geräte an einem Port)
  - "hot plugging": Ein- und Ausstecken im laufenden Betrieb
  - Stromversorgung integriert (100 bis 500 mA)
  - verschiedene Leistungsklassen
  - preisgünstig
  - niedriger Energieverbrauch

## USB: Stecker, Versionen, Geschwindigkeit

- USB-Geschwindigkeitsklassen:
  - 1,5 Mb/s (*low speed*)
  - 12 Mb/s (*full speed*)
  - 480 Mb/s (*high speed*)
- Versionen 1.0 und 1.1 unterstützen nur 1,5 und 12 Mb/s
- Kabellänge 3m (abgeschirmte Kabel 5m)



Type A USB connector  
Upstream (Host, Hub)

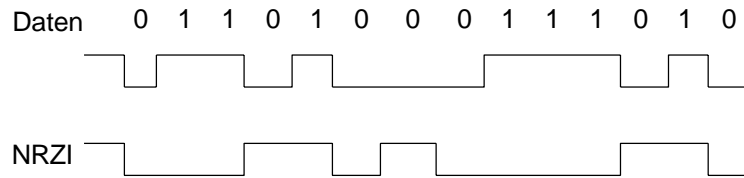


Type B USB connector  
Downstream (Gerät)

## USB: Leitungen

- Sehr einfache Belegung:
  - Pin 1: Busspannung (5 Volt)
  - Pin 2: D+ (Daten)
  - Pin 3: D– (Daten)
  - Pin 4: Erde
- Differenzielle Signalleitungen:
  - Spannung zwischen D+ und D– (max. 4V)
  - Logische Eins:  $U(D+ \text{ zu } D-) > 200 \text{ mV}$
  - Logische Null:  $U(D- \text{ zu } D+) > 200 \text{ mV}$
- Takt im Signal durch NRZI-Codierung (*Non-return-to-zero-inverted*)

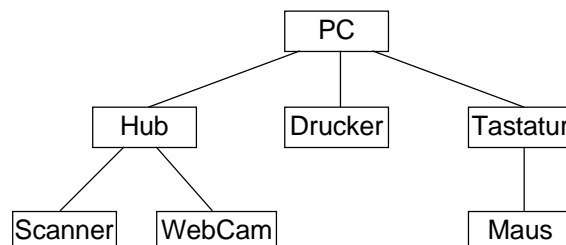
## NRZI-Codierung



- Bei jeder Null Paritätswechsel generiert
- Bei Eins bleibt Polarität unverändert
- Nach sechs aufeinanderfolgenden Einsen zusätzliche Null eingefügt (*bit stuffing*)
- Bei USB komplett in Hardware realisiert

## USB: Topologien

- *Hubs* werden an einen USB-Port angeschlossen und bieten selbst mehrere USB-Schnittstellen an: Baum-Topologie
  - Auch Geräte können als Hubs dienen (z.B. Tastatur)



## USB: Endpoint/Transfer-Typen

- Control Transfers
  - z.B. zur Aushandlung von Geschwindigkeit und Geräteart (*enumeration*)
  - bidirektional
- Interrupt Transfers
  - Unidirektional
  - Eigentlich Polling und kein Hardware-Interrupt, aber garantierte Latenzzeit
  - Wiederholversuch bei Fehlern
- Isochronous Transfers
  - Stetiger und regelmässiger Datenstrom, z.B. Audio und Video Streaming
  - Unidirektional
  - Begrenzte Latenzzeit, garantierte Bandbreite
  - Keine Wiederholversuche bei Fehlern
- Bulk Transfers
  - Grosse, unregelmässig auftretende Datenmengen (*burst*)
  - Unidirektional
  - Keine Garantien bezüglich Bandbreite oder Latenz
  - Nur bei Full- und High-Speed

## USB: Deskriptoren

- Jedes USB-Gerät enthält eine Reihe von Deskriptoren, die durch die Treiber-Software ausgelesen und interpretiert werden:
  - Device Descriptor
    - » Geräteklasse, Hersteller, Anzahl möglicher Konfigurationen
  - Configuration Descriptors
    - » Verschiedene Alternativkonfigurationen (Interfaces)
  - Interface Descriptors
    - » Je Interface: Anzahl der Endpunkte, spezifische Attribute
  - Endpoint Descriptors
    - » Je Endpunkt: Transfertyp, Richtung, Bandbreite, Abfragefrequenz, ...
  - String Descriptors
    - » Textanzeigen
  - Class-Specific Descriptors
    - » je nach Geräteklasse

## USB: Geräteklassen

- USB definiert Standardschnittstellen für die wichtigsten Geräteklassen
  - Erlaubt damit die Verwendung standardisierter Treiber
- Audio Class
- Communications Device Class
- Content Security
- Human Interface Device Class
- Image Device Class
- IrDA Class (Infrarot-Schnittstelle)
- Mass Storage Device Class
- Display Device Class (Monitor-Konfiguration)
- Physical Interface Class (z.B. Force Feedback)
- Power Device Class
- Printing Device Class

## USB: Geräteklasse "Mass Storage Device"

- Zur Steuerung der Zugriffe wird der bewährte "SCSI-2"-Befehlssatz verwendet
  - SCSI = Small Computer Systems Interface
  - ANSI-Standard X3.131
  - ursprünglich ein eigener physikalischer Schnittstellen-Standard
- Beispiel für eine Geräteklasse mit vielen Unterklassen:
- General Mass Storage Subclass (Wechselmedien)
  - Floppy Disk, Magneto-Optical, Zip, ...
- CD-ROM Subclass
- Tape Subclass
- Solid State Subclass

## A5. Digitale Schnittstellen

A5.1 Schnittstellen für Computerperipherie

A5.2 Universal Serial Bus

A5.3 FireWire



Literatur:

Don Anderson (MindShare), FireWire System Architecture,  
Addison-Wesley 1999

## FireWire

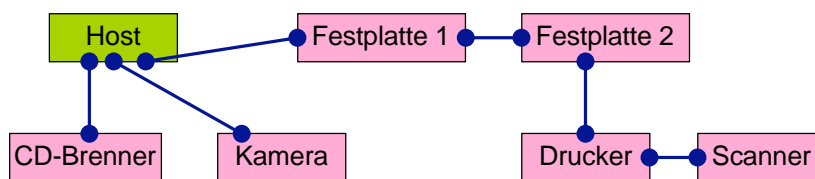


- Serielle Hochgeschwindigkeits-Schnittstelle
  - Implementierung von IEEE-Standard 1394 (1987), aktuell 1394b (1999)
  - Realisiert als Apple NuBus90, nicht weiterverfolgt
  - 1994: *1394 Trade Organisation* (Computer- und Audio-/Video-Technikfirmen)
  - Heutzutage in allen Apple-Geräten, vielen PCs und in fast allen digitalen Videokameras
    - » Sony-Name für FireWire: *i.Link*
- Datenübertragungsraten:
  - 100, 200 MBit/s (unüblich)
  - 400 MBit/s (weit verbreitet)
    - » Vergleich: USB 2.0 High-Speed 480 MBit/s
    - » Vergleich: Fast Ethernet 100 MBit/s
  - 800 MBit/s (aktueller Stand der Technik 2003, "FireWire-800")
  - Standardisiert aber noch nicht in Produktreife:  
1,6 GBit/s und 3,2 GBit/s
  - Siehe auch: <http://www.1394ta.org>



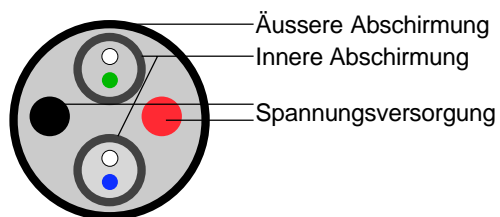
## FireWire: Topologie

- Pro Port bis zu 16 Geräte in Hintereinanderschaltung (*daisy chain*)
  - Nicht zulässig aber physikalisch möglich: Kreis!
  - Abstand zwischen Geräten max. 4,5 m
  - Stranglänge insgesamt max. 72 m
- Automatische Adresszuweisung
- Anschliessen/Entfernen im laufenden Betrieb (*hot plugging*)
- Beispiel für mögliche Topologie:

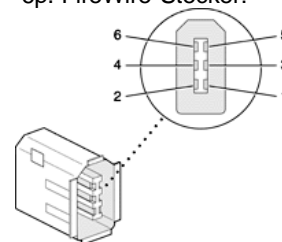


## FireWire: Kabel und Stecker

- 6-polig:
  - Spannungsversorgung (8-40 V, bis zu 1,5 A)
  - Erde
  - 2 Paare von verdrehten Datenleitungen (*twisted pair*): Signal und Takt
    - » beide Signalwege differenziell
- 4-polig:
  - ohne Spannungsversorgung, z.B. in manchen Notebooks
- Kabel relativ dünn und flexibel

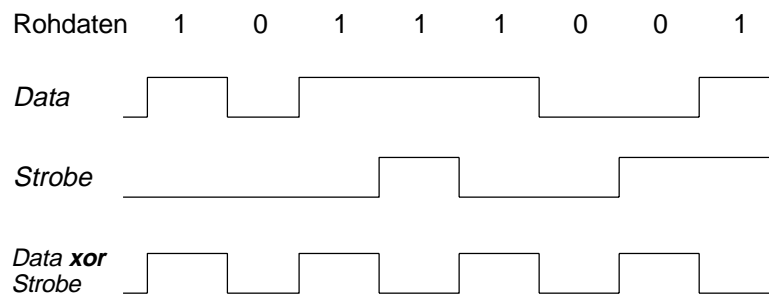


6p. FireWire-Stecker:



## FireWire: Datenübertragung

- Datenübertragung immer unidirektional (Halb-Duplex)
- Daten (*data*) werden NRZ- (*non-return-to-zero*-)codiert
  - Signalwechsel nur bei Bitwechsel
- Takt (*strobe*) ermöglicht Ableitung eines Rechteck-Taktsignals (durch XOR mit dem Datensignal)



## FireWire: Arbitration

- Zwei Verkehrsarten: *asynchron* und *isochron*
- Elektrotechnisch auf dem Bus erkennbar: frei, A->B, B->A
- Knoten müssen die Benutzung des Busses *aushandeln*
- *Asynchrone Arbitration*:
  - Stellt sicher, dass jeder Knoten in fairer Weise Buszugang erhält
- *Isochrone Arbitration*:
  - Stellt sicher, dass jeder Knoten den gewünschten Anteil an der verfügbaren Bandbreite erhält, falls vorhanden
  - Typischerweise 80% für isochronen Verkehr reserviert
- Kombination Asynchron/Isochron:
  - Asynchroner Verkehr benachteiligt, deshalb Möglichkeit von *Prioritäten*