

## 7.3 USB (Universal Serial Bus) 1.1 / 2.0

### 7.3.1 Einleitung

Der USB Bus Standard 1.1 wurde 1995 eingeführt. Ziel war es damals ein Plug&Play fähiges Bussystem zu entwickeln. Anfänglich wurde USB vorrangig eingesetzt, wenn es darum ging Peripherie Geräte mit dem Computer zu verbinden. Doch immer komplexere Applikationen, wie z.B. Übertragung von Video und Multimedia Daten ließen den USB Bus, aufgrund seiner geringen Übertragungsrate von 10Mbit/s, schnell an seine Grenzen stoßen.

Im Jahre 2000 wurde mit einer Erweiterung der Spezifikation auf USB 2.0 dieser Nachteil ausgeglichen.

### 7.3.2 Unterschied USB 1.1 und USB 2.0

Die wesentliche Neuerung der zweiten Auflage ist die vierzigmal schnellere Datenübertragung als bei Version 1.1. USB 2.0 ist trotzdem vollständig abwärtskompatibel zum bestehenden Standard. Vorhandene Kabel und Geräte können weiterhin genutzt werden.

Die USB-2.0-Spezifikation sieht vor, die Timeframes im Millisekundenbereich (USB 1.1) in jeweils acht Timeframes zu je 125 Mikrosekunden zu unterteilen. Somit wird die vierzigfache höhere Datenrate von 480 MBit/s erreicht.

Beim Einstecken eines USB-Gerätes schaltet der USB-2.0-Controller automatisch auf die jeweils geforderte Übertragungsgeschwindigkeit um. Dank der Abwärtskompatibilität können sämtliche Datenraten ohne Geschwindigkeitsverlust gleichzeitig genutzt werden.

Ein USB-2.0-Gerät, angeschlossen an einem USB-1.1-Controller kann allerdings nur die maximale Übertragungsrate von 10 MBit/s nutzen.

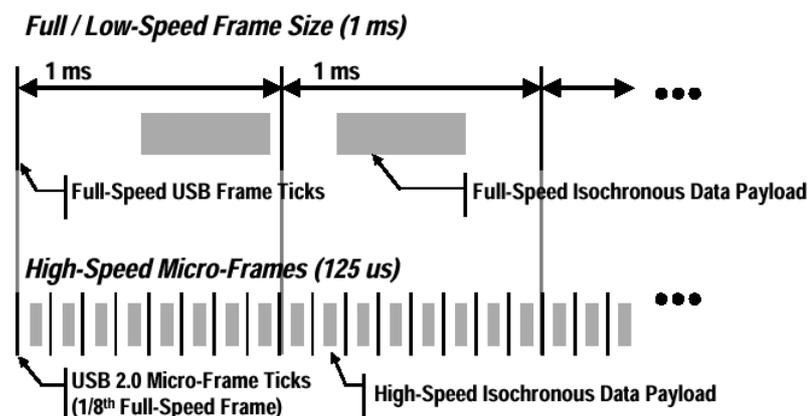


Abb. 7-4 : Frames und Mikroframes

### 7.3.3 Architektur

Ein USB System besteht immer aus einem Host (in der Regel ein PC) und bis zu 127 Peripherie Geräten (Devices). Die Devices sind am Host sternförmig, u.U. über einen Hub angeschlossen. Hubs dienen zur Erweiterung der Anzahl der verfügbaren. Es dürfen jedoch nur maximal fünf Hubs zwischen Host und Device eingesetzt werden.

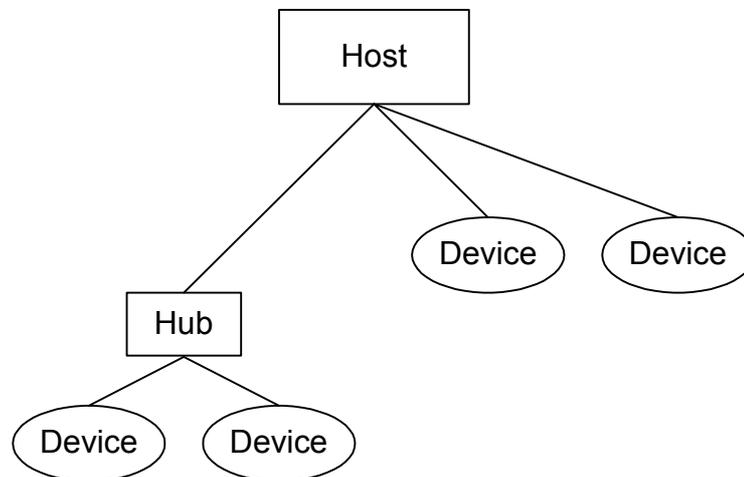


Abb. 7-5 : USB Busarchitektur

### 7.3.4 Protokoll

Host und Devices kommunizieren über Pipes (logische Kanäle) miteinander. Für jede Pipe sind Endpunkte im Host bzw. Device vorhanden. Je nach Übertragungsgeschwindigkeit steht eine unterschiedliche Anzahl von adressierbaren Endpunkten zur Verfügung, wobei alle Devices den Endpunkt 0 unterstützen.

Die Übertragung der Daten erfolgt asynchron oder isynchron (konstante Bandbreite) mit verschiedenen Transfers.

Control Transfers für die Konfiguration von Devices. Paketgröße ist maximal 64 Bytes. Control Transfer sind als einzige Transfervariante Bidirektional.

Isochronous Transfers für die periodische und kontinuierliche Übertragung von zeitkritischen Daten wie z. B. Audio oder Video. Paketgröße ist maximal 1023 Bytes.

Interrupt Transfers für kleine Datenpakete geringer Häufigkeit (z.B. Maus). Paketgröße maximal 8 Bytes für low-speed und 64 Bytes für full-speed Devices.

Bulk Transfers für große Datenmengen und nicht-periodische Übertragung. Dieser Transfer wird für Anwendungen verwendet, die keine bestimmte Bandbreite benötigen und deren Daten auch verzögert gesendet werden können (z.B. Drucker). Paketgröße ist maximal 64 Bytes.

Beim Anschluss neuer Geräte oder beim Reset werden alle Endpunkte auf die default Adresse 0 gesetzt. Erst danach beginnt der Host mit der Konfiguration der

Devices, bei der die Betriebsparameter wie z.B. Anzahl der Endpunkte, Device Adresse, Datenaufkommen, Art der Stromversorgung usw. festgelegt werden. Danach kann ein Device auf Requests vom Host reagieren.

### 7.3.5 Protokoll Overhead

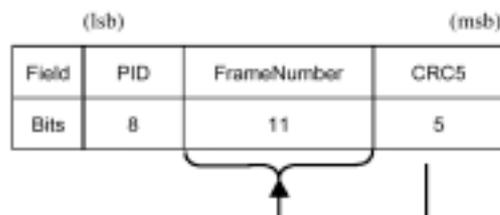
Neben den zu übertragenden Daten muss jedes Datenpaket aufgrund von Synchronisations- und Fehlererkennungszwecken mit einem Overhead versehen werden. Je nach Übertragungsgeschwindigkeit und der Transfervariante ist die Anzahl, der für den Overhead verwendet Bytes, sowie die Struktur der Datenpakete, variabel.

### 7.3.6 Folgende Felder werden verwendet:

- Sync Feld enthält Synchronisationsinformationen für den Empfänger.
- Packet Identifier Feld (PID) beinhaltet Informationen über die verwendete Fehlererkennung und Typ des Datenpakets.
- Adress- und Endpunkt-Feld enthalten Adressen des Empfängers oder Senders.
- CRCs enthält die CRC über die Daten oder Adress und Endpunkt Feld.

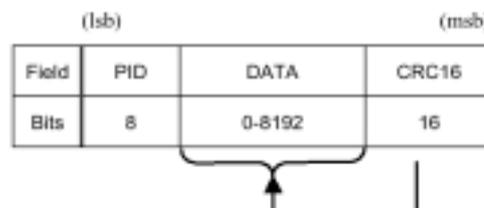
**Beispiele** (Sync Felder weggelassen)

Der Start eines neuen Frames wird durch das Start-of-Frame gekennzeichnet:



**Abb. 7-6 : Start-Of-Frame**

Die zu übertragenden Daten befinden sich in einem Datenframe:



**Abb. 7-7 : Datenframe**

## **Literatur**

- [1] Universal Serial Bus Specification, Revision 2.0 April 27, 2000  
<http://www.usb.org/>
- [2] Universal Serial Bus Specification, Revision 1.1 September 23, 1998  
<http://www.usb.org/>