

9 Universal Serial Bus

Der *Universal Serial Bus* (USB) wurde von der Firma Intel im Jahr 1995 als leistungsfähiger Ersatz für die parallele und die serielle Schnittstelle sowie für den Anschluss von Tastatur und Maus – statt PS/2-Anschluss – eingeführt. Mittlerweile existieren drei USB-Standards und eine Vielzahl von Geräten verfügt über einen USB-Anschluss, so dass der USB eine wichtige Standard-Schnittstelle für Computer, Tablets und Smartphones bildet. Das *USB Implementers Forum* (www.usb.org) ist ein Interessen- und Handelsverband, der sich der Standardisierung und der Weiterführung der USB-Technologie angenommen hat und zu deren Mitgliedern fast alle führenden Hardware-Hersteller gehören.

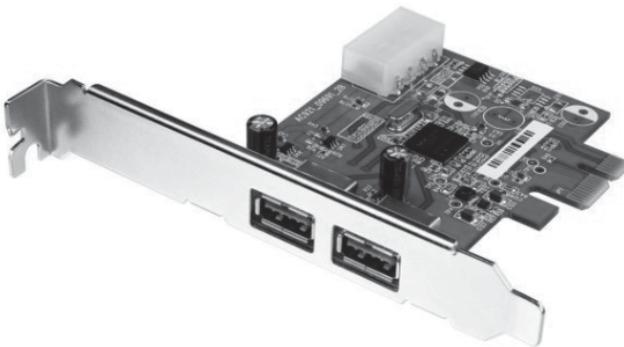


Bild 9.1: Eine PCIe-Erweiterungskarte mit zwei USB-3.0-Anschlüssen

Der traditionelle USB (1.1 und 2.0) verwendet für die Kommunikation lediglich zwei Signalleitungen, was bedeutet, dass in diesem seriellen Datenstrom alle Steuer- und Dateninformationen für eine Vielzahl einzelner Geräte enthalten ist und daher ein recht aufwendiges Protokoll absolviert werden muss, was in den einzelnen Einheiten eine verhältnismäßig hohe Eigenintelligenz in Form spezieller Mikrocontroller erfordert. Man hat es beim USB also weder mit einzelnen Signalleitungen zu tun, mit deren Hilfe die Kommunikation mit der Peripherie wie beim Parallel Port stattfindet, noch mit einem einfachen, seriellen System, wie es bei der RS-232-Schnittstelle der Fall ist, bei der fest definierte Übertragungsparameter wie die Baudrate oder die Anzahl von Daten- und Stopbits die Datenübertragung steuern.

Neben dem ersten USB-Standard (Version 1.1) gibt es seit dem Jahre 2001 den USB-Standard 2.0 und seit 2008 die Version USB 3.0 (Super Speed), wobei grundsätzlich eine Abwärtskompatibilität gegeben ist. Der USB unterstützt damit die vier folgenden Klassen:

- Low Speed: maximal 1,5 MBit/s
- Medium Speed/Full Speed: maximal 12 MBit/s
- High Speed: maximal 480 MBit/s
- Super Speed: maximal 5 GBit/s

Zu den Low Speed-Geräten werden typischerweise Eingabegeräte wie Tastatur und Maus gerechnet. Für den Modus mit 12 MBit/s findet man zwei unterschiedliche Bezeichnungen, und zwar *Medium Speed* und *Full Speed*, was als Maximum des ersten USB-Standards gilt und für alle anderen Einheiten (z. B. Web-Camera, Scanner, Modem) zum Einsatz kommt, die demgegenüber eine höhere Datenübertragungsrate verlangen. Die Full-Speed-Bezeichnung wurde von einigen Herstellern gewählt, um hiermit eben eine hohe Datenübertragungsrate zu signalisieren, was immer wieder zu Verwechslungen mit dem *High Speed Mode* führt, der erst mit dem USB-2.0-Standard eingeführt worden ist. *Full Speed* kennzeichnet somit also nur die »volle Geschwindigkeit«, welche die erste USB-Version zu leisten vermag, und dies sind 12 MBit/s.

In den folgenden Abschnitten werden zunächst die Zusammenhänge für Einheiten nach den Standards USB 1.1 und USB 2.0 behandelt. USB 3.0 wird separat im Abschnitt 9.9 erörtert, welches auf den vorherigen Standards aufbaut und auf den gleichen Prinzipien (Übertragungsmodi, Pipes, Endpoints) beruht. Die wesentliche Neuerung ist dabei die (theoretische) Verzehnfachung der bisher maximalen Datenrate auf 5 GBit/s, was im Wesentlichen durch zusätzliche, getrennte Send- und Empfangsleitungen erreicht wird, so dass USB-3.0-Geräte über neue Steckverbinder verfügen müssen.

9.1 Topologie

Der USB bildet ein serielles Bussystem und stellt sich in seiner Topologie als kombinierte Strang-Sternstruktur dar. Er ist im elektrischen Sinne somit gar kein Bus, sondern lediglich auf logischer Ebene und wird von einem Host (Root Hub) gesteuert, von dem alle Aktivitäten ausgehen. Dabei findet ein Polling-Betrieb statt, und kein USB-Gerät kann von sich aus einen Transfer initiieren.

Der USB-Host bildet üblicherweise auch gleich einen Verteiler (Hub) für den Anschluss mehrerer USB-Geräte, die hier als *Nodes* bezeichnet werden. Ein USB-Hub – Sternverteiler – kann zum einen im Chipset des Mainboards integriert oder auch als eigenständiges Gerät ausgeführt sein. Des Weiteren kann eine geeignete Tastatur oder ein Monitor ebenfalls als USB-Hub fungieren. Diese Hubs erweitern das US-Bussystem durch die Bereitstellung zusätzlicher Ports, wobei jeder Hub einen Upstream- und mehrere Downstream-Ports zur Verfügung stellt und wie jedes andere USB-Gerät zu konfigurieren ist, denn er erhält ebenfalls (automatisch) eine eigene Adresse im System. Diese Hubs erweitern das USB-System durch die Bereitstellung zusätzlicher Ports, wobei jeder Hub einen Upstream- und mehrere Downstream-Ports zur Verfügung stellt und wie jedes andere USB-Gerät zu konfigurieren ist, denn er erhält ebenfalls (automatisch) eine eigene Adresse im System. Ein USB-Hub ist daher nicht einfach nur ein Verteiler, der die Daten lediglich durchreicht, sondern beinhaltet eine gewisse Eigenintelligenz, wofür im Hub üblicherweise ein eigener Mikrocontroller verwendet wird, der den Datenverkehr in Abhängigkeit von den hieran angeschlossenen USB-Geräten entsprechend regelt. Generell spielt ein USB-Hub *die* wesentliche Rolle, um eine entsprechende Kompatibilität

zwischen Einheiten herzustellen, die unterschiedlichen USB-Standards entsprechen, weil der USB-Hub hierfür quasi eine Vermittlerrolle übernimmt.

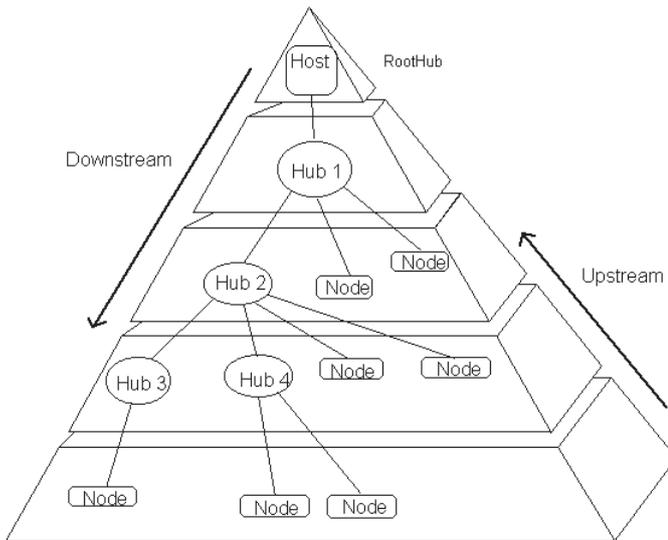


Bild 9.2: Der USB ist als kombinierte Strang-Sternstruktur ausgelegt. Gesteuert wird er von einem Host und die einzelnen Geräte werden als Nodes bezeichnet.

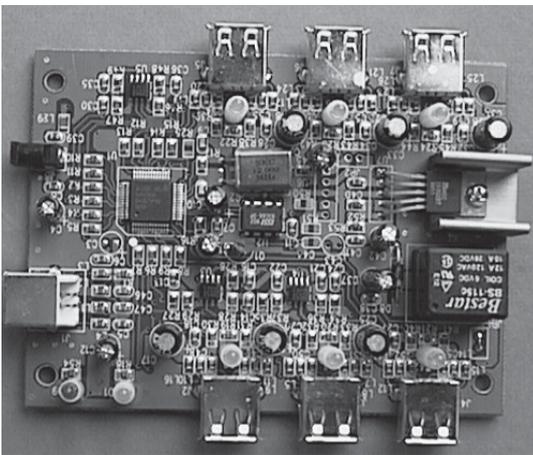


Bild 9.3: Das Innenleben eines USB-Hubs, der sechs Buchsen für die Downstream-Ports und einen (links) für den Upstream-Port besitzt

Zur Geräteidentifizierung dienen eine Herstellerkennung (Vendor ID, 2 Byte), eine Geräteerkennung (Device ID, 2 Byte), eine Versionsnummer (2 Byte) und ein Class Code von 3 Byte. Während der Initialisierung wird vom Host festgestellt, welche Datentypen jeweils unterstützt und welche Bandbreiten von den einzelnen USB-Einheiten benötigt werden. Jede USB-Einheit erhält eine eigene Adresse – wovon maximal 127 möglich sind –, die vom USB-Host nach dem Einschalten des Systems automatisch zugewiesen wird.

9.2 Anschlüsse und Signale

Für die USB-Verbindungen wird ein vierpoliger Anschluss verwendet, der mechanisch unterschiedlich ausgeführt sein kann. Der Connector der A-Serie weist einen rechteckigen Querschnitt und der der B-Serie einen quadratischen auf. Der Grund für die unterschiedlichen Steckverbinder liegt darin, dass mechanisch verhindert werden soll, unzulässige Verbindungen innerhalb der USB-Struktur (vgl. Bild 9.2) herstellen zu können. Die Geräte müssen stets gemäß dieser USB-Pyramidenstruktur miteinander verbunden werden, damit die Upstream- und die Downstream-Richtung eingehalten wird. *Upstream* bedeutet stets die Verbindung nach oben – letztendlich bis hin zum Host (Root Hub), *downstream* nach unten zu den Geräten (Nodes). Durch die unterschiedliche Ausführung der erläuterten USB-Stecker sollte es dabei eigentlich nicht zu unzulässigen Verbindungen kommen.

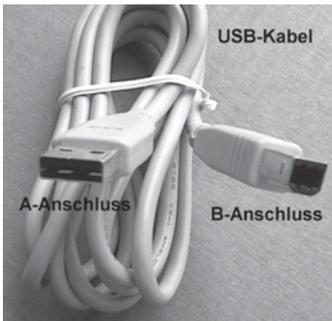


Bild 9.4: Das übliche USB-Kabel verfügt über zwei unterschiedliche Stecker. Der A-Typ wird an den Host (Hub) angeschlossen, der B-Typ an das jeweilige USB-Gerät.

Neben den beiden Standardanschlüssen (A- und B-Typ) gibt es auch noch zwei verschiedene miniaturisierte USB-Anschlussausführungen (Mini- und Micro-USB), die in Form einer kleinen Buchse insbesondere bei Digitalkameras oder Smartphones zu finden sind. Dabei handelt es sich gewissermaßen um eine mechanisch verkleinerte Ausführung des B-Typs, was bedeutet, dass sich auf der anderen Seite des passenden Verbindungskabels ein üblicher A-Anschluss befindet. Bei Low-Speed-Geräten ist das Kabel meist fest angeschlossen, d. h., es gibt hier dann keinen B-Connector.



Bild 9.5: Die miniaturisierten USB-Verbindungen

Soll eine USB-Verbindung länger als 5 m ausfallen, sind so genannte »aktive USB-Kabel« erforderlich, die einen elektrischen Signalverstärker beinhalten. Beim Einsatz eines externen *USB Hub*, etwa zwischen PC und einem USB-Gerät, sind zweimal 5 m, also eine Gesamtlänge von 10 m, zulässig, weil ein *USB Hub* das Signal entsprechend der Spezifikation grundsätzlich verstärken muss.

In Handel sind geradezu eine Unmenge von unterschiedlichen USB-Adaptern zu finden, womit sich prinzipiell die unsinnigsten Verbindungen herstellen lassen, sodass dann auch keine USB-Verbindung zustande kommen kann. Generell sollte die grundsätzliche Topologie entsprechend Upstream- und Downstream-Richtung (Bild 9.2) eingehalten werden und „schnellere“ USB-Geräte (externe Festplatten) sollten mit einer eigenen direkten Root-Hub-Verbindung versehen werden. PCs verfügen meist über mindestens vier integrierte einzelne Root Hubs (Hosts). Außerdem sollten nur maximale Kabellängen von 5 m (3 m bei USB 3.0) zwischen den USB-Einheiten realisiert und keinesfalls (passive) Verlängerungskabel eingesetzt werden.

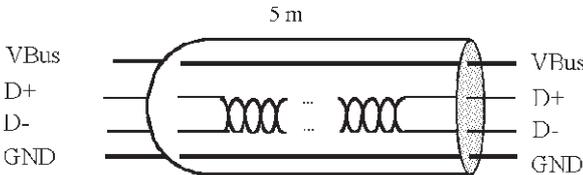


Bild 9.6: Das Kabel für den Universal Serial Bus führt vier Leitungen. Mithilfe der beiden Signalleitungen wird eine differentielle Datenübertragung durchgeführt.

Die USB-Verbindung besteht aus vier Leitungen, und für die Datenübertragung werden lediglich zwei differentielle Signalleitungen (D+, D-) verwendet. Die beiden anderen USB-Leitungen führen das Massesignal und die Spannungsversorgung (V_{cc}) von typisch 5 V, die für die Versorgung von USB-Geräten über den Bus verwendet werden können. Je nach Leistungsaufnahme machen die einzelnen USB-Geräte davon Gebrauch oder nicht, denn für die maximale Stromaufnahme über den Bus sind 500 mA definiert. Einige Geräte, wie etwa Scanner oder auch Modems, benötigen deshalb eine eigene Stromversorgung. Dies gilt jedoch weder für eine USB-Tastatur noch eine USB-Maus, da diese auch bei den konventionellen Anschlüssen von dem PC selbst mit der nötigen Betriebsspannung versorgt werden können. Der USB ermöglicht das *Hot Plugging*, also das Anschließen oder Abziehen eines USB-Geräts während des laufenden Betriebs, wofür die Kontakte der Spannungsversorgung etwas länger ausgeführt sind als die beiden Kontakte für das Datensignal.

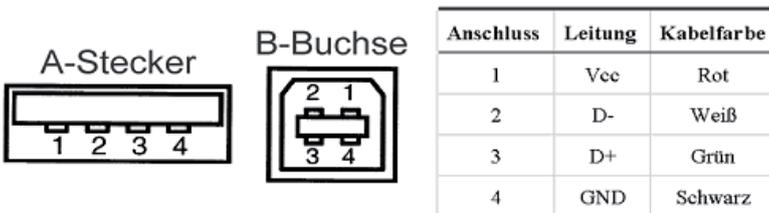


Bild 9.7: Belegung und typische Farbkennzeichnung der USB-Anschlüsse

9.3 Kommunikation

Da für die Datenübertragung beim USB lediglich zwei differentielle Signalleitungen (D+, D-) verwendet werden, werden die einzelnen Bits eines Datenpakets nacheinander übertragen. Kennzeichnend für die zwei differentiellen Datenleitungen D+ und D- ist, dass jeweils die eine Leitung das invertierte Datensignal der anderen Leitung führt. Dadurch reduziert sich die Abstrahlung und erhöht sich die Übertragungssicherheit. Das Differenzsignal beträgt maximal 4 V (typisch 3,3 V). Eine logische Eins liegt dann vor, wenn $D+ \dots D- > 200 \text{ mV}$ ist, und eine Null ergibt sich bei $D- \dots D+ > 200 \text{ mV}$, was praktisch 2,8 V-3,6 V für ein High bedeutet, während ein Low einem Pegel von weniger als 0,3 V entspricht. Ein Taktsignal existiert beim USB nicht, sondern es wird aus dem Datenstrom selbst mittels der NRZI-Codierung (Non Return to Zero Inverted) generiert. Die Codierung der Daten erfolgt derart, dass diese zunächst durch ein Schieberegister in eine serielle Form gebracht werden. Daraufhin durchlaufen sie einen so genannten *Bitstuffer* und werden dann dem NRZI-Encoder zugeführt, der die Daten auf die differentiell arbeitenden Treiber führt.

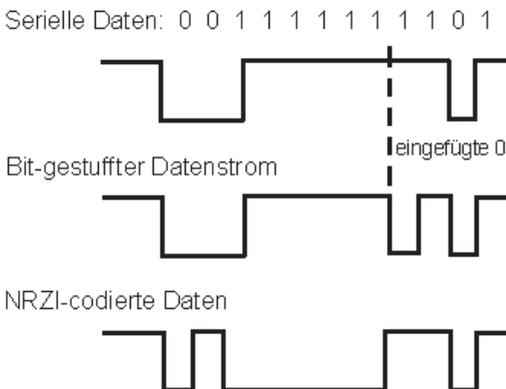


Bild 9.8: Die seriellen Daten durchlaufen vor der NRZI-Codierung einen Bitstuffer, der nach jeder sechsten Eins eine Null einfügt, damit ein Polaritätswechsel stattfindet.

Bei jeder Null im Datenstrom generiert NRZI einen Polaritätswechsel, während die Polarität bei einer Eins unverändert bleibt, was zu einem Synchronisationsverlust zwischen Sender und Empfänger führen kann. Eben aus diesem Grund gibt es den Bitstuffer, der nach sechs aufeinander folgenden Einsen automatisch eine Null einfügt, was zu einem forcierten Polaritätswechsel führt.

Auf der Empfängerseite findet nun der umgekehrte Vorgang statt (Empfang der Daten, NRZI-Decodierung), und ein *Bitdestuffer* entfernt die eingefügte Null wieder, wodurch das Schieberegister wieder die reinen Nutzdaten erhält. Die erwähnten Einheiten wie NRZI-Codierer oder Bitstuffer sind komplett in Hardware realisiert und können daher nicht durch Software ersetzt oder manipuliert werden.

Der Datentransfer findet auf dem Bus nicht kontinuierlich statt. Vielmehr ist die Buszeit in einzelne Zeitabschnitte, Frames beziehungsweise Microframes (USB 2.0) genannt, eingeteilt. Ein Microframe dauert 125 μs . Acht Microframes bilden einen Frame, der 1 ms dauert und bei USB 1.1 den kleinsten Zeitabschnitt darstellt.

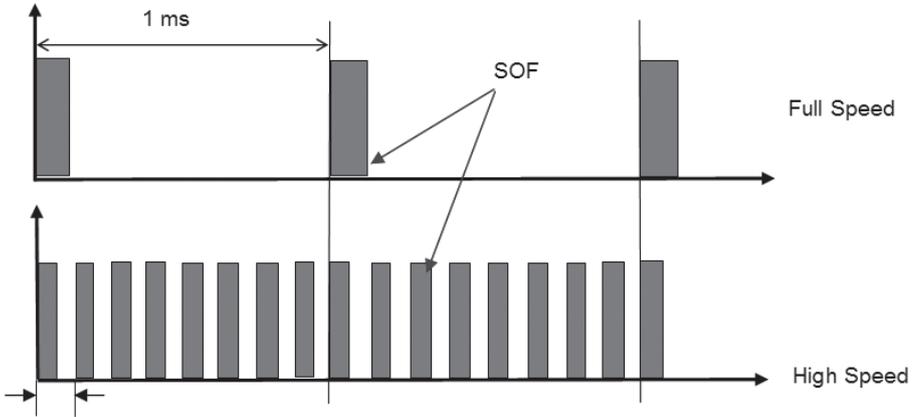


Bild 9.9: Frames und Microframes

Der Host vergibt für die unterschiedlichen Übertragungsarten jeweils eine bestimmte Bandbreite. Für *Control Transfers* sind im High-Speed-Modus ständig 20 % der Bandbreite eines Microframe reserviert. Die restliche Bandbreite wird nach Bedarf auf die anderen Transfers aufgeteilt, wie es im Bild 9.10 angedeutet ist.

Zu Beginn eines jeden Microframe sendet der Host ein *Start of Frame Token* (SOF), um somit den Start eines neuen Frame zu signalisieren. Danach folgen die Datenpakete der jeweiligen Transaktionen. Je nach Busauslastung kann sich eine Daten-Transaktion über mehrere Microframes erstrecken.

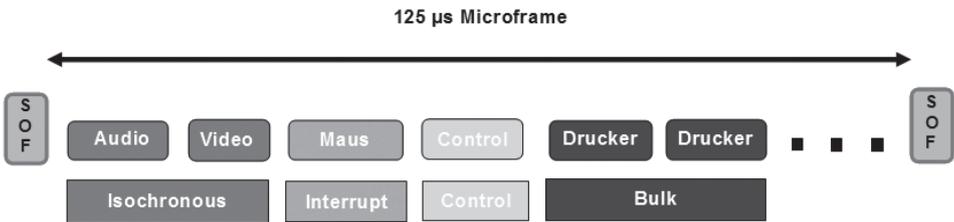


Bild 9.10: Verschiedene Transfers innerhalb eines Microframe

Der Super-Speed-Modus (USB 3.0) ändert nichts an dieser Framestruktur. Bei USB 3.0 reduzieren sich die Reaktions- und Übertragungszeiten durch den Wegfall bestimmter Token und der Präamble (siehe Abschnitt 9.7), was aufgrund zweier zusätzlicher Leitungspaare (SSTX, SSRX) möglich ist.

9.4 Geräteerkennung

Für die USB-Geräteerkennung ist zunächst die Unterscheidung zwischen einem Low-Speed- und einem Medium-Speed-Gerät (Full Speed) von Bedeutung. Jedes USB-Gerät verfügt über einen Widerstand von 1,5 kΩ, der an die Betriebsspannung (3,3 V) einerseits und an die D- oder an die D+-Signalleitung andererseits angeschlossen ist. Ein Low-Speed-Gerät legt diesen Widerstand an die Leitung D-, ein Full-Speed-Gerät an die Leitung D+.