

DIN EN 62680-1

ICS 35.200

Einsprüche bis 2013-05-11

Entwurf**Schnittstellen des Universellen Seriellen Busses für Daten und Energie –****Teil 1: Festlegung des Universellen Seriellen Busses****(IEC 100/1981/CDV:2012);****Englische Fassung FprEN 62680-1:2012**

Universal Serial Bus interfaces for data and power –

Part 1: Universal Serial Bus Specification

(IEC 100/1981/CDV:2012);

English version FprEN 62680-1:2012

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2013-03-11 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 775 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die Englische Fassung des europäischen Dokuments FprEN 62680-1:2012 „Schnittstellen des Universellen Seriellen Busses für Daten und Energie – Teil 1: Festlegung des Universellen Seriellen Busses“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 100/1981/CDV:2012 „Universal Serial Bus interfaces for data and power – Part 1: Universal Serial Bus Specification“ unverändert in den Entwurf FprEN 62680-1:2012 übernommen worden.

Das internationale Dokument wurde vom TC 100 „Audio, video and multimedia systems and equipment“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründen, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Das K 742 hat mit RS_742_2012-0064 beschlossen, dass die Bedingungen des DIN-Präsidialbeschlusses 1/2004 für das vorliegende Normprojekt erfüllt sind.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der IEC 62680-1 veröffentlicht. Zu deren Abschnitt 2, der die Begriffe und Abkürzungen festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 742 „Audio-, Video- und Multimediasysteme, -geräte und -komponenten“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Nationaler Anhang NA (informativ)

2 Begriffe und Abkürzungen

Dieses Kapitel listet und definiert die in dieser Festlegung verwendeten Begriffe und Abkürzungen.

Die deutschen Übersetzungen haben Empfehlungscharakter und können nur die ungefähre Begrifflichkeit in Deutsch wiedergeben. Unter Experten wird empfohlen, ausschließlich die englischen Fachbegriffe zu verwenden, um die Gefahr von Missverständnissen zu reduzieren.

Abschluss (en: termination)	passive Komponente, die am Ende eines Kabels angebracht wird, um zu verhindern, dass Signale reflektiert oder wiederholt werden
Abschnitt (en: stage)	Teil einer Folge aus einer Steuerungsübertragung; Abschnitte enthalten den Setup-Abschnitt, den Daten-Abschnitt und den Status-Abschnitt
Abtastrate (Fs) (en: sample rate (Fs))	Anzahl der Abtastungen je Sekunde, ausgedrückt in Hertz (Hz)
Abtastraten-Wandlung (en: sample rate conversion (SRC))	bestimmte Umsetzung des RA-Verfahrens zur Nutzung abgetasteter analoger Datenströme
Abtastung (en: sample)	kleinste Dateneinheit, mit der ein Endpunkt arbeitet; eine Eigenschaft des Endpunktes
abwärts (en: downstream)	Richtung des Datenstroms vom Host oder weg vom Host. Ein Abwärtsanschluss ist der Anschluss an einer Hub, der elektrisch am weitesten von dem Host entfernt ist, und der Datenverkehr von der Hub erzeugt. Abwärtsanschlüsse empfangen Aufwärts-Datenverkehr.
ACK	Handshake-Paket, das eine positive Bestätigung gibt
aktives Gerät (en: active device)	ein mit Strom versorgtes Gerät, das nicht im Status „Suspend“ ist
Anforderung (en: request)	eine an ein USB-Gerät gerichtete Anforderung, die in einem Datenteil eines SETUP-Pakets enthalten ist
Anschluss (en: port)	Zugangspunkt zu oder von einem System oder Schaltkreis. Für USB ist es der Punkt, an dem ein USB-Gerät angefügt wird.
asynchrone Daten (en: asynchronous data)	Daten, die zu unregelmäßigen Zeitpunkten mit vereinfachten Anforderungen an Wartezeiten gesendet werden
asynchrone RA (en: asynchronous RA)	ankommende Datenrate F_{s_i} und ausgehende Datenrate F_{s_o} des RA-Verfahrens sind unabhängig (d. h. es gibt keinen gemeinsamen Takt). Siehe auch „Anpassung der Datenrate“.
asynchrone SRC (en: asynchronous SRC)	ankommende Datenrate F_{s_i} und ausgehende Datenrate F_{s_o} des SRC-Verfahrens sind unabhängig (d. h. es gibt keinen gemeinsamen Takt). Siehe auch „Anpassung der Datenrate“.
Audiogerät (en: audio device)	Gerät, das eine Quelle für oder ein Empfänger von analogen Daten ist

aufwärts (en: upstream)	Richtung des Datenflusses zum Host. Ein Aufwärts-Anschluss ist der Anschluss des elektrisch am nächsten zum Host gelegenen Gerätes, da einen Aufwärts-Datenstrom von der Hub erzeugt. Aufwärts-Anschlüsse empfangen Abwärts-Datenströme.
Augenmuster (en: eye pattern)	Darstellungsform von USB-Signalen, die minimale und maximale Spannungspegel sowie Jitter erkennt
AWG#	Maßeinheit für den Kabelquerschnitt, festgelegt in der Norm „American Wire Gauge“
Bandbreite (en: bandwidth)	Anzahl der je Zeiteinheit übertragenen Daten, typischerweise Bit je Sekunde (b/s) oder Bytes je Sekunde (B/s)
Beginn des Frames (SOF) (en: Start-of-Frame (SOF))	die erste Transaktion in jedem (Mikro-)Frame. Der SOF ermöglicht es Endpunkten, den Beginn des (Mikro-)Frames zu erkennen und interne Endpunkte-Takte mit dem Host zu synchronisieren.
big endian	Verfahren, Daten zu speichern, bei dem das höchste Byte (MSB) von Werten mit mehreren Bytes an der unteren Speicheradresse speichert. Zum Beispiel legt ein 16 Bit ganzzahliger Wert im „big endian“-Format das niedrigste Byte (LSB) in die obere Adresse und das höchste Byte (MSB) in die niedrigere Adresse. Siehe auch „little endian“.
Bit	eine von digitalen Computern verwendete Informationseinheit. Sie stellt das kleinste Stück adressierbarer Information in einem Computer dar. Ein Bit ist die Auswahl zwischen zwei Möglichkeiten und wird typischerweise durch eine logische Eins (1) oder Null (0) dargestellt.
Bit-Auffüllung (en: bit stuffing)	Einsetzen eines „0“-Bits in einen Datenstrom, um einen elektrischen Übergang auf den Datenleitungen zu erzeugen, während eine PLL verriegelt bleibt
b/s	Übertragungsrate ausgedrückt in Bits je Sekunde
B/s	Übertragungsrate ausgedrückt in Bytes je Sekunde
Bus-Aufzählung (en: bus enumeration)	Erkennung und Identifizierung von USB-Geräten
Byte	Datenelement mit der Größe acht Bit
Charakteristika (en: characteristics)	Qualitäten eines USB-Gerätes, die unveränderbar sind, z. B. ist die Geräteklasse ein Geräte-Charakteristikum
Client	auf dem Host ansässige Software, die mit der USB-Systemsoftware zusammenarbeitet, um die Datenübertragung zwischen Host und einer Funktion zu ermöglichen. Der Client ist oft der Anbieter von Daten und der Nutzer von übertragenen Daten.
CRC	siehe „zyklische Redundanz-Prüfung“
CTI	Computer-Telefon-Integration

Datenraten-Anpassung (en: rate adaptation)	Verfahren, bei dem ein eingehender Datenstrom der Datenrate F_{S_j} mit gewissen, von einem Datenraten-Algorithmus festgelegten Qualitätsverlusten in einen ausgehenden Datenstrom der Datenrate F_{S_o} umgewandelt wird. Für dieses Verfahren sind Fehlerabwehrmechanismen erforderlich. F_{S_j} und F_{S_o} können unterschiedlich und asynchron sein. F_{S_j} ist die Eingangs-Datenrate des RA und F_{S_o} ist die Ausgangs-Datenrate des RA.
Datenstromleitung (en: stream pipe)	Leitung, die Daten als Strom von Abtastungen ohne festgelegte USB-Struktur überträgt.
Dienst (en: service)	von der Systemprogrammierungs-Schnittstelle (SPI) bereitgestelltes Verfahren
Dienste-Intervall (en: service interval)	Periode zwischen aufeinanderfolgenden Anforderungen zum Senden oder Empfangen von Daten an einen USB-Endpunkt
Dienste-Jitter (en: service jitter)	Abweichung der Lieferung eines Dienstes von der geplanten Lieferzeit
Dienste-Rate (en: service rate)	Anzahl der an einem bestimmten Endpunkt durchgeführten Dienste je Zeiteinheit
DWORD	Datenelement mit der Größe von zwei Worten (d. h. vier Bytes oder 32 Bit)
dynamische Einfügung und Entfernung (en: dynamic insertion and removal)	Fähigkeit, während der Host arbeitet, Geräte hinzuzufügen und zu entfernen
E ² PROM	siehe „Electrical Erasable Programmable Read Only Memory“
EEPROM	siehe „Electrical Erasable Programmable Read Only Memory“
Electrical Erasable Programmable Read Only Memory (EEPROM)	nicht-flüchtige, wiederbeschreibbare Speichertechnologie
Endbenutzer (en: end user)	Benutzer von einem Host
Endpunkt (en: endpoint)	siehe „Geräteendpunkt“
Endpunkt mit hoher Bandbreite (en: high-bandwidth endpoint)	Endpunkt eines Hochgeschwindigkeits-Gerätes, der mehr als 1 024 Bytes und weniger als 3 073 Bytes je Mikroframe überträgt
Endpunktadresse (en: endpoint address)	Kombination einer Endpunktnummer und einer Endpunkttrichtung in einem USB-Gerät. Jede Endpunktadresse unterstützt Datenübertragung in eine Richtung.
Endpunktnummer (en: endpoint number)	ein 4-Bit-Wert zwischen 0H und einschließlich FH, der mit einem Endpunkt auf einem USB-Gerät zusammenhängt

Endpunkttrichtung (en: endpoint direction)	Richtung der Datenübertragung auf dem USB. Die Richtung kann entweder IN oder OUT sein. IN bezieht sich auf Datenübertragung zum Host; OUT bezieht sich auf Datenübertragung vom Host.
EOF	Ende des (Mikro-)Frames
EOP	Ende des Pakets
Externer Anschluss (en: external port)	siehe Anschluss
Fähigkeiten (en: capabilities)	Eigenschaften eines USB-Gerätes, die von einem Host verwaltet werden
falsches EOP (en: false EOP)	störendes, üblicherweise rausch-induziertes Ereignis, das von einem Paketempfänger als ein EOP ausgelegt wird
Frame	Zeiteinheit von 1 Millisekunde auf schnellen bzw. langsamen Bussen
Frame-Muster (en: frame pattern)	Folge von Frames, die ein wiederkehrendes Muster in der Anzahl von übertragenen Abtastungen je Frame aufweisen. Für eine 44,1 kHz Tonübertragung könnte das Frame-Muster neun Frames mit 44 Mustern betragen, gefolgt von einem Frame mit 45 Mustern.
F _s	siehe „Abtastrate“
Funktion (en: function)	USB-Gerät, das die Fähigkeit hat, einen Host zu versorgen, wie z. B. eine ISDN-Verbindung, ein digitales Mikrofon oder Lautsprecher
Gerät (en: device)	logische oder physikalische Einheit, die eine Funktion ausführt. Die aktuell beschriebene Einheit hängt vom Kontext der Referenz ab. Auf der untersten Ebene kann ein Gerät eine einzelne Hardware-Komponente sein, wie z. B. ein Speicher. Auf einer höheren Ebene kann es eine Ansammlung von Hardware-Komponenten sein, die eine bestimmte Funktion erfüllen, wie z. B. ein USB-Schnittstellengerät. Noch eine Ebene höher kann ein Gerät eine mit USB verbundene Funktion sein, z. B. ein Daten-/Fax-Modem. Geräte können physikalisch, elektrisch, adressierbar und logisch sein.
Geräteadresse (en: device address)	7-Bit-Wert, der die Adresse eines USB-Gerätes darstellt. Die Geräteadresse ist die Standardadresse (00H), wenn das USB-Gerät zum ersten Mal eingeschaltet oder zurückgesetzt wird. Die USB-Systemsoftware ordnet den Geräten dann eindeutige Adressen zu.
Geräteendpunkt (en: device endpoint)	eindeutig adressierbarer Teil eines USB-Gerätes, das die Quelle oder der Empfänger von Informationen bei einer Kommunikation zwischen dem Host und dem Gerät ist. Siehe auch „Endpunktadresse“.
Geräte-Ressourcen (en: device resources)	vom USB-Gerät bereitgestellte Ressourcen, wie Pufferspeicher und Endpunkte. Siehe auch „Host-Ressourcen“ und „Universelle serielle Bus-Ressourcen“.
Gerätesoftware (en: device software)	Software, die für die Verwendung eines USB-Gerätes zuständig ist. Diese Software kann, muss aber nicht, zuständig sein für die Konfiguration des Gerätes zur Verwendung.

geteilte Transaktion (en: split transaction)	von Host-Controllern und Hubs unterstützte Transaktionsart. Diese Transaktionsart ermöglicht es Geräten mit voller Geschwindigkeit und niedriger Geschwindigkeit, an mit hoher Geschwindigkeit betriebenen Hubs angeschlossen zu werden.
Handshake-Paket (en: handshake packet)	Paket, das bestimmte Bedingungen akzeptiert oder zurückweist. Siehe z. B. „ACK“ und „NAK“.
Hochgeschwindigkeit (en: high-speed)	USB-Betrieb mit 480 Mb/s. Siehe auch „niedrige Geschwindigkeit“ und „volle Geschwindigkeit“
Host	Host-Computer-System, auf dem der USB-Host-Controller installiert ist. Dies enthält die Host-Hardware-Plattform (CPU, Bus, usw.) und das verwendete Betriebssystem.
Host-Controller	USB-Schnittstelle des Hosts
Host-Controller-Treiber (en: host controller driver (HCD))	USB-Softwareebene, die die Hardware des Host-Controllers abstrahiert. Der Host-Controller-Treiber stellt einen SPI für die Zusammenarbeit mit dem Host-Controller bereit. Der Host-Controller-Treiber versteckt die spezifischen Hardware-Ausführungen des Host-Controllers.
Host-Ressourcen	Vom Host bereitgestellte Ressourcen, wie z. B. Pufferspeicher und Unterbrechungen. Siehe auch „Geräte-Ressourcen“ und „Universelle serielle Bus-Ressourcen“.
Hub	USB-Gerät, das zusätzliche Verbindungen zum USB bereitstellt.
Hub Tier	Eins plus die Anzahl der USB-Verbindungen in einem Verbindungspfad zwischen dem Host und einer Funktion. Siehe auch Bild 4.1.
Hüllkurvendetektor (en: envelope detector)	elektronischer Schaltkreis in einem USB-Gerät, der die USB-Datenleitungen überwacht und bestimmte spannungsbezogene Signalcharakteristika erkennt
I/O Anforderungspaket (en: I/O request packet)	nachweisbare Anforderung von einem Software Client, um Daten zwischen ihm selbst (auf dem Host) und dem Endpunkt eines Gerätes in der richtigen Richtung zu bewegen
IRP	siehe „I/O Anforderungspaket“
IRQ	siehe „Unterbrechungsanforderungen“
isochrone Daten (en: isochronous data)	Datenstrom, dessen Zeitverhalten von der Lieferrate vorgegeben ist
isochrone Übertragung (en: isochronous transfer)	eine der vier USB-Übertragungstypen. Die isochrone Übertragung wird verwendet, wenn mit isochronen Daten gearbeitet wird. Isochrone Übertragung ermöglicht periodische, durchgängige Kommunikation zwischen Host und Gerät. Siehe auch „Übertragungstypen“.
isochroner Empfänger-Endpunkt (en: isochronous sink endpoint)	Endpunkt, der einen vom Host gesendeten isochronen Datenstrom verarbeiten kann

isochroner Quellen-Endpunkt (en: isochronous source endpoint)	Endpunkt, der einen zum Host gesendeten isochronen Datenstrom erzeugen kann
isochrones Gerät (en: isochronous device)	Einheit mit isochronen Endpunkten, wie in den USB-Festlegungen erklärt, die abgetastete analoge Datenströme oder synchrone Datenströme sendet oder empfängt.
Jitter	Tendenz zum Verlust der Synchronisation, verursacht durch mechanische oder elektrische Wechsel. Genauer, die Phasenverschiebung digitaler Impulse über ein Übertragungsmedium.
kb/s	Übertragungsrate, ausgedrückt in Kilobit je Sekunde
kB/s	Übertragungsrate, ausgedrückt in Kilobyte je Sekunde
Konfigurations-Software (en: configuring software)	auf dem Host ansässige Software, die für die Konfigurierung eines USB-Gerätes zuständig ist. Dies kann ein System-Konfigurator sein oder eine gerätespezifische Software.
Leitung (en: pipe)	logisches Abstraktum, das die Verbindung zwischen einem Endpunkt auf einem Gerät und der Software auf dem Host darstellt. Eine Leitung hat mehrere Attribute, z. B. kann eine Leitung Daten als Datenströme übertragen (Datenstromleitung; en: stream pipe) oder als Mitteilungen (Mitteilungsleitung; en: message pipe). Siehe auch „Datenstromleitung“ und „Mitteilungsleitung“.
little endian	Verfahren, Daten zu speichern, bei dem das niedrigste Byte (LSB) von Werten mit mehreren Bytes an der unteren Speicheradresse speichert. Zum Beispiel legt ein 16-Bit ganzzahliger Wert im „little endian“-Format das niedrigste Byte (LSB) in die untere Adresse und das höchste Byte (MSB) in die nächste Adresse. Siehe auch „big endian“.
LOA	Verlust der Bus-Aktivität, dargestellt durch ein SOP ohne ein entsprechendes EOP
LSb	niedrigstes Bit in einem Datenwert
LSB	niedrigstes Byte in einem Datenwert
Mb/s	Übertragungsrate, ausgedrückt in Megabit je Sekunde
MB/s	Übertragungsrate, ausgedrückt in Megabyte je Sekunde
Mengenübertragung (en: bulk transfer)	eine der vier USB-Übertragungsverfahren. Mengenübertragung besteht aus einer nicht-periodischen, großen, pulsähnlichen Kommunikation, die üblicherweise zur Übertragung die gesamte verfügbare Bandbreite nutzt und die auch verzögert werden kann, bis eine entsprechende Übertragungsbandbreite zur Verfügung steht. Siehe auch „Übertragungstyp“.
Mikroframe (en: micro frame)	Zeiteinheit von 125 Mikrosekunden, die auf Hochgeschwindigkeits-Bussen bereitgestellt wird
Mitteilungsleitung (en: message pipe)	bidirektionale Leitung, die Daten mittels eines Anforderungs-/Daten-/Status-Modells überträgt. Die Daten haben eine eingeprägte Struktur, die es ermöglicht, Anfragen zuverlässig zu erkennen und zu kommunizieren.

MSb	höchstes Bit in einem Datenwert
MSB	höchstes Byte in einem Datenwert
NAK	Handshake-Paket, das eine negative Bestätigung anzeigt
Niedrige Geschwindigkeit (en: low-speed)	USB-Betrieb bei 1,5 MB/s. Siehe auch „volle Geschwindigkeit“ und „hohe Geschwindigkeit“.
non return to zero invert (NRZI)	Verfahren zur Einkodierung serieller Daten, bei dem Einsen und Nullen durch gegenteilige und abwechselnde Werte dargestellt werden, wobei es keine Rückkehr auf Null-Spannung (Referenz) zwischen enkodierten Bits gibt. Dies beseitigt die Notwendigkeit von Takt-Impulsen.
NRZI	siehe „non return to zero invert“
Objekt (en: object)	Host-Software oder Datenstruktur, die eine USB-Funktionseinheit darstellt
Paket (en: packet)	Datenbündel, das zur Übertragung in Gruppen organisiert wird. Pakete enthalten üblicherweise drei Elemente: Steuerinformation (z. B. Quelle, Ziel und Länge), die zu übertragenden Daten und Bits zur Fehlererkennung und -korrektur.
Paket ID (PID)	Feld in einem USB-Paket, das den Pakettyp anzeigt und als Folgerung das Paketformat und die dem Paket zugeordnete Art der Fehlererkennung.
Paket-Puffer (en: packet buffer)	der von einem USB-Gerät verwendete logische Puffer zum Senden und Empfangen einzelner Pakete. Dieser legt die maximale Paketgröße fest, die ein Gerät senden und empfangen kann.
Paket ID (PID)	Feld in einem USB-Paket, das den Pakettyp anzeigt und als Folgerung das Paketformat und die dem Paket zugeordnete Art der Fehlererkennung
Phase	Token-, Daten- oder Handshake-Paket. Eine Transaktion hat drei Phasen.
phasenstarre Schleife (en: phase locked loop (PLL))	Schaltkreis, der als Phasendetektor arbeitet, um einen Oszillator in Phase mit einer Eingangsfrequenz zu halten
physikalisches Gerät (en: physical device)	Gerät, das eine physikalische Ausführung hat, z. B. ein Lautsprecher, ein Mikrofon oder ein CD-Player
PID	siehe „Paket ID“
PLL	siehe „phasenstarre Schleife“
POR	siehe „Power on reset“
power on reset (POR)	Wiederherstellung eines Speichergerätes, Registers oder Speichers in einen festgelegten Status nach dem Einschalten
programmierbare Datenrate (en: programmable data rate)	entweder eine feste Datenrate (Einzelfrequenz-Endpunkte), eine begrenzte Anzahl von Datenraten (32 kHz, 44,1 kHz, 48 kHz) oder eine kontinuierlich programmierbare Datenrate. Die genauen Programmiermöglichkeiten eines Endpunktes müssen in der entsprechenden klassenspezifischen Endpunktbeschreibung beschrieben werden.

Protokoll (en: protocol)	spezieller Satz von Regeln, Verfahren oder Übereinkünften bezogen auf Format und Zeitverhalten von Datenübertragungen zwischen zwei Geräten
Puffer (en: buffer)	Speicher, der unterschiedliche Datenraten oder zeitversetzte Ereignisse bei der Übertragung von einem Gerät zu einem anderen ausgleicht
RA	siehe „Datenraten-Anpassung“
Root Hub	USB-Hub, die direkt an den Host-Controller angebunden ist. Diese Hub (Tier 1) ist an den Host angebunden.
Root-Anschluss (en: root port)	Anschluss für abgehende Daten an einer Root-Hub
Sendeabruf (en: polling)	eine mindestens einmalige Anfrage an mehrere Geräte, ob sie irgendwelche Daten zum Übertragen haben
SOF	siehe „Beginn des Frames“
SOP	Beginn eines Pakets
SPI	siehe „Systemprogrammierungs-Schnittstelle“
SRC	siehe „Abtastraten-Wandlung“
Standardadresse (en: default address)	eine von den USB-Festlegungen bestimmte und vom USB-Gerät beim Einschalten oder beim Reset verwendete Adresse. Die Standardadresse ist 00H.
Standardleitung (en: default pipe)	von der USB-Systemsoftware erzeugte Mitteilungsleitung zur Übergabe von Steuerungs- und Statusinformation zwischen Host und dem Endpunkt Null eines USB-Gerätes
Steuerungs-Endpunkt (en: control endpoint)	Paar von Geräte-Endpunkten mit derselben Endpunkt-Nummer, die von einer Steuerleitung verwendet werden. Steuer-Endpunkte übertragen Daten in beide Richtungen und nutzen daher beide Endpunkt-Richtungen einer Geräte-Adresse und die Kombination von deren Endpunkt-Nummern. Daher benötigt jeder Steuer-Endpunkt zwei Endpunkt-Adressen.
Steuerungsleitung (en: control pipe)	dasselbe wie eine Mitteilungsleitung
Steuerungsübertragung (en: control transfer)	eines der vier USB-Übertragungsverfahren. Steuerungsübertragung unterstützt Kommunikationen wie Konfiguration/Befehl/Statustyp zwischen Client und Funktion. Siehe auch „Übertragungstyp“.
Störgeräusch (en: babble)	unerwartete Bus-Aktivitäten, die außerhalb eines festgelegten Punktes in einem (Mikro-)Frame auftreten
synchroner RA (en: synchronous RA)	eingehende Datenrate F_{s_i} und ausgehende Datenrate F_{s_o} des RA-Verfahrens werden aus demselben Haupttakt abgeleitet. Es gibt eine feste Beziehung zwischen F_{s_i} und F_{s_o} .
Synchronisierungsart (en: synchronization type)	Klassifizierung, die die Fähigkeit eines isochronen Endpunktes charakterisiert, sich mit anderen isochronen Endpunkten zu verbinden

Systemprogrammierungs-Schnittstelle (SPI) (en: system programming interface (SPI))	festgelegte Schnittstelle für von der Systemsoftware bereitgestellte Dienste
TDM	siehe „Zeitbereichs-Multiplexing“
TDR	siehe „Zeitbereichs-Reflektometer“
timeout	Erkennung fehlender Bus-Aktivitäten über bestimmte Zeiträume
Token-Paket (en: token packet)	Paketart, die erkennt, welche Transaktion auf dem Bus durchgeführt wird
Transaktion (en: transaction)	Lieferung eines Dienstes an einen Endpunkt; bestehend aus einem Token-Paket, einem optionalen Datenpaket und einem optionalen Handshake-Paket. Je nach Transaktionsart sind bestimmte Pakete zugelassen oder gefordert.
Transaktionsübersetzer (en: transaction translator)	funktionale Komponente einer USB-Hub. Der Transaktionsübersetzer antwortet auf bestimmte Hochgeschwindigkeits-Transaktionen und übersetzt diese dann in Transaktionen mit voller oder niedriger Geschwindigkeit, wenn Geräte mit voller oder niedriger Geschwindigkeit am Anschluss für abgehende Daten vorhanden sind.
Treiber (en: driver)	auf Hardware bezogen ist dies ein I/O-Pfad, der eine externe Last ansteuert. Auf Software bezogen ist es ein für die Schnittstelle zu einem Hardware-Gerät verantwortliches Programm, d. h. ein Geräte-Treiber.
Turnaround-Zeit (en: turn-around time)	Zeit, die ein Gerät benötigt, um die Übertragung eines Paketes zu beginnen, nachdem ein Paket empfangen wurde, um Kollisionen auf dem USB-Bus zu vermeiden. Diese Zeit hängt ab von der Länge des Kabels und dessen Charakteristika für die Laufzeitverzögerung sowie dem Ort des übertragenden Gerätes in Bezug auf andere Geräte am USB-Bus.
Übertragung (en: transfer)	eine oder mehrere Bus-Transaktionen, um Informationen zwischen einem Software-Client und seiner Funktion hin und her zu bewegen
Übertragungsart (en: transfer type)	bestimmt die Charakteristika des Datenflusses zwischen dem Software-Client und seiner Funktion. Vier Standard-Übertragungsarten sind festgelegt: Steuerung, Unterbrechung, Menge und Isochron.
universelle serielle Bus-Ressourcen (en: universal serial bus resources)	von USB bereitgestellte Ressourcen, wie z. B. Bandbreite und Leistung. Siehe auch „Geräte-Ressourcen“ und „Host-Ressourcen“.
universeller serieller Bus-Treiber (USB-D) (en: universal serial bus driver (USB-D))	Software-Einheit auf dem Host, die zuständig ist für die Bereitstellung gemeinsamer Dienste für Clients, die eine oder mehrere Funktionen auf einem oder mehreren Host-Controllern beeinflussen
Unterbrechungsanforderung (en: interrupt request (IRQ))	Hardware-Signal, das es dem Gerät erlaubt, Aufmerksamkeit von einem Host anzufordern. Der Host ruft üblicherweise ein Dienste-Unterbrechungsverfahren auf, um die Anforderung verursachende Bedingung abzuarbeiten.

Unterbrechungsübertragung (en: interrupt transfer)	eine der vier USB-Übertragungstypen. Die Charakteristika der Unterbrechungs-Übertragung sind kleine Datenmengen, nicht-periodisch, niederfrequent und mit begrenzter Verzögerung. Unterbrechungs-Übertragung wird üblicherweise zur Bedienung von Dienste-Anforderungen verwendet. Siehe auch „Übertragungstyp“.
USB-D	siehe „universeller serieller Bus-Treiber“
USB-IF	USB Implementers Forum, Inc., Non-Profit-Unternehmen, dass gegründet wurde, um die Entwicklung von USB-konformen Produkten voranzutreiben und die Technologie zu unterstützen.
virtuelles Gerät (en: virtual device)	Gerät, das von einer Software-Schnittstellenebene dargestellt wird. Ein Beispiel eines virtuellen Gerätes ist eine Festplatte mit zugehörigem Gerätetreiber und Client-Software, die die Wiedergabe einer Audio-Datei vom Typ .WAV ermöglicht.
voll-duplex (en: full-duplex)	Computer-Datenübertragung, die gleichzeitig in beide Richtungen erfolgt
Volle Geschwindigkeit (en: full-speed)	USB-Betrieb bei 12 MB/s. Siehe auch „niedrige Geschwindigkeit“ und „hohe Geschwindigkeit“.
Wort (en: word)	Datenelement in der Größe von zwei Bytes (16 Bit)
Zeitbereichs-Multiplexing (TDM) (en: time division multiplexing (TDM))	Verfahren zur Übertragung mehrfacher Signale (Daten, Sprache und/oder Video) gleichzeitig über ein Kommunikationsmedium durch Verschachtelung eines jeden Signals mit einem anderen
Zeitbereichs-Reflektrometer (TDR) (en: time domain reflectrometer (TDR))	Messinstrument, das Impedanz-Charakteristika auf den USB-Leitungen messen kann
zurückziehen (en: retire)	Aktion, die den Dienst für eine Übertragung beendet und den entsprechenden Software-Clients über die Beendigung informiert
zyklische Redundanz- Prüfung (CRC) (en: cyclic redundancy check)	Datenprüfung zur Feststellung, ob ein Fehler beim Übertragen, Lesen oder Schreiben von Daten aufgetreten ist. Das Ergebnis eines CRC wird üblicherweise gespeichert oder mit den geprüften Daten übertragen. Das gespeicherte oder übertragene Ergebnis wird mit einer für die Daten berechneten CRC verglichen, um festzustellen, ob ein Fehler aufgetreten ist.

INTRODUCTION

The IEC 62680 series is based on a series of specifications that were originally developed by the USB Implementers Forum (USB-IF). These specifications were submitted to the IEC under the auspices of a special agreement between the IEC and the USB IF.

The USB Implementers Forum, Inc.(USB-IF) is a non-profit corporation founded by the group of companies that developed the Universal Serial Bus specification. The USB-IF was formed to provide a support organization and forum for the advancement and adoption of Universal Serial Bus technology. The Forum facilitates the development of high-quality compatible USB peripherals (devices), and promotes the benefits of USB and the quality of products that have passed compliance testing.

ANY USB SPECIFICATIONS ARE PROVIDED TO YOU "AS IS, "WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE USB IMPLEMENTERS FORUM AND THE AUTHORS OF ANY USB SPECIFICATIONS DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OR INFORMATION IN THIS SPECIFICATION.

THE PROVISION OF ANY USB SPECIFICATIONS TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

Entering into USB Adopters Agreements may, however, allow a signing company to participate in a reciprocal, royalty-free licensing arrangement for compliant products. For more information, please see:

<http://www.usb.org/developers/docs/>

http://www.usb.org/developers/devclass_docs#approved

IEC DOES NOT TAKE ANY POSITION AS TO WHETHER IT IS ADVISABLE FOR YOU TO ENTER INTO ANY USB ADOPTERS AGREEMENTS OR TO PARTICIPATE IN THE USB IMPLEMENTERS FORUM.”

This series covers the Universal Serial Bus interfaces for data and power and consists of the following parts:

IEC 62680-1 Universal Serial Bus interfaces for data and power - Part 1: Universal Serial Bus Specification, Revision 2.0

IEC 62680-2 Universal Serial Bus interfaces for data and power - Part 2: USB Micro-USB Cables and Connectors Specification, Revision 1.01

IEC 62680-3 Universal Serial Bus interfaces for data and power - Part 3: USB Battery Charging Specification, Revision 1.2

IEC 62680-4 Universal Serial Bus interfaces for data and power - Part 4: Universal Serial Bus Cables and Connectors Class Document Rev. 2.0

This part of the IEC 62680 series consists of several distinct parts:

- the main body of the text, which consists of the original specification developed by the USB IF;
- Annexes A to H, which consist of the amendments and corrigenda brought to the original specification by the USB IF.

1	CONTENTS	
2	1 Chapter 1 Introduction	17
3	1.1 Motivation	17
4	1.2 Objective of the Specification	17
5	1.3 Scope of the Document	18
6	1.4 USB Product Compliance	18
7	1.5 Document Organization	18
8	2 Chapter 2 Terms and Abbreviations	19
9	3 Chapter 3 Background	25
10	3.1 Goals for the Universal Serial Bus	25
11	3.2 Taxonomy of Application Space	25
12	3.3 Feature List	26
13	4 Chapter 4 Architectural Overview	28
14	4.1 USB System Description	28
15	4.1.1 Bus Topology	28
16	4.2 Physical Interface	29
17	4.2.1 Electrical	29
18	4.2.2 Mechanical	30
19	4.3 Power	30
20	4.3.1 Power Distribution	30
21	4.3.2 Power Management	31
22	4.4 Bus Protocol	31
23	4.5 Robustness	31
24	4.5.1 Error Detection	32
25	4.5.2 Error Handling	32
26	4.6 System Configuration	32
27	4.6.1 Attachment of USB Devices	32
28	4.6.2 Removal of USB Devices	32
29	4.6.3 Bus Enumeration	32
30	4.7 Data Flow Types	32
31	4.7.1 Control Transfers	33
32	4.7.2 Bulk Transfers	33
33	4.7.3 Interrupt Transfers	33
34	4.7.4 Isochronous Transfers	33
35	4.7.5 Allocating USB Bandwidth	34
36	4.8 USB Devices	34
37	4.8.1 Device Characterizations	34
38	4.8.2 Device Descriptions	35
39	4.9 USB Host: Hardware and Software	37
40	4.10 Architectural Extensions	37
41	5 Chapter 5 USB Data Flow Model	38
42	5.1 Implementer Viewpoints	38
43	5.2 Bus Topology	39
44	5.2.1 USB Host	40
45	5.2.2 USB Devices	40

46	5.2.3	Physical Bus Topology.....	41
47	5.2.4	Logical Bus Topology	42
48	5.2.5	Client Software-to-function Relationship	43
49	5.3	USB Communication Flow.....	43
50	5.3.1	Device Endpoints.....	45
51	5.3.2	Pipes	46
52	5.3.3	Frames and Microframes	48
53	5.4	Transfer Types	48
54	5.4.1	Table Calculation Examples.....	49
55	5.5	Control Transfers	50
56	5.5.1	Control Transfer Data Format	50
57	5.5.2	Control Transfer Direction.....	51
58	5.5.3	Control Transfer Packet Size Constraints.....	51
59	5.5.4	Control Transfer Bus Access Constraints	52
60	5.5.5	Control Transfer Data Sequences	55
61	5.6	Isochronous Transfers	56
62	5.6.1	Isochronous Transfer Data Format.....	56
63	5.6.2	Isochronous Transfer Direction	56
64	5.6.3	Isochronous Transfer Packet Size Constraints	56
65	5.6.4	Isochronous Transfer Bus Access Constraints	58
66	5.6.5	Isochronous Transfer Data Sequences	59
67	5.7	Interrupt Transfers	59
68	5.7.1	Interrupt Transfer Data Format	59
69	5.7.2	Interrupt Transfer Direction	60
70	5.7.3	Interrupt Transfer Packet Size Constraints.....	60
71	5.7.4	Interrupt Transfer Bus Access Constraints	61
72	5.7.5	Interrupt Transfer Data Sequences	63
73	5.8	Bulk Transfers	63
74	5.8.1	Bulk Transfer Data Format.....	63
75	5.8.2	Bulk Transfer Direction	64
76	5.8.3	Bulk Transfer Packet Size Constraints	64
77	5.8.4	Bulk Transfer Bus Access Constraints	64
78	5.8.5	Bulk Transfer Data Sequences	66
79	5.9	High-Speed, High Bandwidth Endpoints	66
80	5.9.1	High Bandwidth Interrupt Endpoints	67
81	5.9.2	High Bandwidth Isochronous Endpoints	67
82	5.10	Split Transactions.....	69
83	5.11	Bus Access for Transfers	69
84	5.11.1	Transfer Management.....	69
85	5.11.2	Transaction Tracking	72
86	5.11.3	Calculating Bus Transaction Times.....	73
87	5.11.4	Calculating Buffer Sizes in Functions and Software	75
88	5.11.5	Bus Bandwidth Reclamation	75
89	5.12	Special Considerations for Isochronous Transfers	75
90	5.12.1	Example Non-USB Isochronous Application.....	77
91	5.12.2	USB Clock Model	79

— Entwurf —

– 7 –

92	5.12.3 Clock Synchronization	81
93	5.12.4 Isochronous Devices	81
94	5.12.5 Data Prebuffering	89
95	5.12.6 SOF Tracking	89
96	5.12.7 Error Handling	90
97	5.12.8 Buffering for Rate Matching	91
98	6 Chapter 6 Mechanical	92
99	6.1 Architectural Overview	92
100	6.2 Keyed Connector Protocol	92
101	6.3 Cable	93
102	6.4 Cable Assembly	93
103	6.4.1 Standard Detachable Cable Assemblies	93
104	6.4.2 High-/full-speed Captive Cable Assemblies	95
105	6.4.3 Low-speed Captive Cable Assemblies	97
106	6.4.4 Prohibited Cable Assemblies	99
107	6.5 Connector Mechanical Configuration and Material Requirements	99
108	6.5.1 USB Icon Location	100
109	6.5.2 USB Connector Termination Data	100
110	6.5.3 Series “A” and Series “B” Receptacles	101
111	6.5.4 Series “A” and Series “B” Plugs	104
112	6.6 Cable Mechanical Configuration and Material Requirements	107
113	6.6.1 Description	108
114	6.6.2 Construction	108
115	6.6.3 Electrical Characteristics	111
116	6.6.4 Cable Environmental Characteristics	111
117	6.6.5 Listing	111
118	6.7 Electrical, Mechanical, and Environmental Compliance Standards	111
119	6.7.1 Applicable Documents	117
120	6.8 USB Grounding	117
121	6.9 PCB Reference Drawings	117
122	7 Chapter 7 Electrical	121
123	7.1 Signaling	121
124	7.1.1 USB Driver Characteristics	124
125	7.1.2 Data Signal Rise and Fall, Eye Patterns	130
126	7.1.3 Cable Skew	139
127	7.1.4 Receiver Characteristics	139
128	7.1.5 Device Speed Identification	141
129	7.1.6 Input Characteristics	142
130	7.1.7 Signaling Levels	145
131	7.1.8 Data Encoding/Decoding	157
132	7.1.9 Bit Stuffing	157
133	7.1.10 Sync Pattern	159
134	7.1.11 Data Signaling Rate	159
135	7.1.12 Frame Interval	159
136	7.1.13 Data Source Signaling	160
137	7.1.14 Hub Signaling Timings	162

138	7.1.15 Receiver Data Jitter	163
139	7.1.16 Cable Delay	165
140	7.1.17 Cable Attenuation	166
141	7.1.18 Bus Turn-around Time and Inter-packet Delay	167
142	7.1.19 Maximum End-to-end Signal Delay	168
143	7.1.20 Test Mode Support	169
144	7.2 Power Distribution	170
145	7.2.1 Classes of Devices	170
146	7.2.2 Voltage Drop Budget	174
147	7.2.3 Power Control During Suspend/Resume	175
148	7.2.4 Dynamic Attach and Detach	176
149	7.3 Physical Layer	177
150	7.3.1 Regulatory Requirements	177
151	7.3.2 Bus Timing/Electrical Characteristics	177
152	7.3.3 Timing Waveforms	187
153	8 Chapter 8 Protocol Layer	190
154	8.1 Byte/Bit Ordering	190
155	8.2 SYNC Field	190
156	8.3 Packet Field Formats	190
157	8.3.1 Packet Identifier Field	190
158	8.3.2 Address Fields	191
159	8.3.3 Frame Number Field	192
160	8.3.4 Data Field	192
161	8.3.5 Cyclic Redundancy Checks	192
162	8.4 Packet Formats	193
163	8.4.1 Token Packets	193
164	8.4.2 Split Transaction Special Token Packets	194
165	8.4.3 Start-of-Frame Packets	198
166	8.4.4 Data Packets	199
167	8.4.5 Handshake Packets	200
168	8.4.6 Handshake Responses	201
169	8.5 Transaction Packet Sequences	202
170	8.5.1 NAK Limiting via Ping Flow Control	211
171	8.5.2 Bulk Transactions	215
172	8.5.3 Control Transfers	220
173	8.5.4 Interrupt Transactions	223
174	8.5.5 Isochronous Transactions	223
175	8.6 Data Toggle Synchronization and Retry	228
176	8.6.1 Initialization via SETUP Token	229
177	8.6.2 Successful Data Transactions	229
178	8.6.3 Data Corrupted or Not Accepted	229
179	8.6.4 Corrupted ACK Handshake	230
180	8.6.5 Low-speed Transactions	231
181	8.7 Error Detection and Recovery	232
182	8.7.1 Packet Error Categories	232
183	8.7.2 Bus Turn-around Timing	232

184	8.7.3	False EOPs	233
185	8.7.4	Babble and Loss of Activity Recovery	234
186	9	Chapter 9 USB Device Framework	235
187	9.1	USB Device States	235
188	9.1.1	Visible Device States	235
189	9.1.2	Bus Enumeration	239
190	9.2	Generic USB Device Operations	239
191	9.2.1	Dynamic Attachment and Removal.....	240
192	9.2.2	Address Assignment	240
193	9.2.3	Configuration	240
194	9.2.4	Data Transfer	241
195	9.2.5	Power Management	241
196	9.2.6	Request Processing	241
197	9.2.7	Request Error	243
198	9.3	USB Device Requests.....	244
199	9.3.1	bmRequestType.....	244
200	9.3.2	bRequest	244
201	9.3.3	wValue	244
202	9.3.4	wIndex	245
203	9.3.5	wLength	245
204	9.4	Standard Device Requests.....	245
205	9.4.1	Clear Feature	247
206	9.4.2	Get Configuration	248
207	9.4.3	Get Descriptor	248
208	9.4.4	Get Interface.....	249
209	9.4.5	Get Status	249
210	9.4.6	Set Address.....	251
211	9.4.7	Set Configuration.....	251
212	9.4.8	Set Descriptor.....	252
213	9.4.9	Set Feature.....	252
214	9.4.10	Set Interface.....	254
215	9.4.11	Synch Frame	254
216	9.5	Descriptors	254
217	9.6	Standard USB Descriptor Definitions	255
218	9.6.1	Device	255
219	9.6.2	Device_Qualifier	257
220	9.6.3	Configuration	258
221	9.6.4	Other_Speed_Configuration.....	260
222	9.6.5	Interface	260
223	9.6.6	Endpoint	262
224	9.6.7	String.....	265
225	9.7	Device Class Definitions	266
226	9.7.1	Descriptors	266
227	9.7.2	Interface(s) and Endpoint Usage.....	266
228	9.7.3	Requests	266
229	10	Chapter 10 USB Host: Hardware and Software.....	267

230	10.1 Overview of the USB Host	267
231	10.1.1 Overview	267
232	10.1.2 Control Mechanisms	270
233	10.1.3 Data Flow	270
234	10.1.4 Collecting Status and Activity Statistics	271
235	10.1.5 Electrical Interface Considerations	271
236	10.2 Host Controller Requirements	271
237	10.2.1 State Handling	271
238	10.2.2 Serializer/Deserializer	272
239	10.2.3 Frame and Microframe Generation	272
240	10.2.4 Data Processing	273
241	10.2.5 Protocol Engine	273
242	10.2.6 Transmission Error Handling	273
243	10.2.7 Remote Wakeup	274
244	10.2.8 Root Hub	274
245	10.2.9 Host System Interface	274
246	10.3 Overview of Software Mechanisms	274
247	10.3.1 Device Configuration	274
248	10.3.2 Resource Management	276
249	10.3.3 Data Transfers	277
250	10.3.4 Common Data Definitions	278
251	10.4 Host Controller Driver	278
252	10.5 Universal Serial Bus Driver	279
253	10.5.1 USBBD Overview	279
254	10.5.2 USBBD Command Mechanism Requirements	280
255	10.5.3 USBBD Pipe Mechanisms	282
256	10.5.4 Managing the USB via the USBBD Mechanisms	284
257	10.5.5 Passing USB Preboot Control to the Operating System	286
258	10.6 Operating System Environment Guides	287
259	11 Chapter 11 Hub Specification	288
260	11.1 Overview	288
261	11.1.1 Hub Architecture	288
262	11.1.2 Hub Connectivity	289
263	11.2 Hub Frame/Microframe Timer	291
264	11.2.1 High-speed Microframe Timer Range	291
265	11.2.2 Full-speed Frame Timer Range	291
266	11.2.3 Frame/Microframe Timer Synchronization	292
267	11.2.4 Microframe Jitter Related to Frame Jitter	294
268	11.2.5 EOF1 and EOF2 Timing Points	294
269	11.3 Host Behavior at End-of-Frame	297
270	11.3.1 Full-/low-speed Latest Host Packet	297
271	11.3.2 Full-/low-speed Packet Nullification	297
272	11.3.3 Full-/low-speed Transaction Completion Prediction	297
273	11.4 Internal Port	298
274	11.4.1 Inactive	299
275	11.4.2 Suspend Delay	299

— Entwurf —

– 11 –

276	11.4.3 Full Suspend (Fsus)	299
277	11.4.4 Generate Resume (GResume).....	299
278	11.5 Downstream Facing Ports.....	299
279	11.5.1 Downstream Facing Port State Descriptions	301
280	11.5.2 Disconnect Detect Timer	305
281	11.5.3 Port Indicator.....	306
282	11.6 Upstream Facing Port.....	307
283	11.6.1 Full-speed	307
284	11.6.2 High-speed	308
285	11.6.3 Receiver	308
286	11.6.4 Transmitter	311
287	11.7 Hub Repeater.....	312
288	11.7.1 High-speed Packet Connectivity	312
289	11.7.2 Hub Repeater State Machine.....	315
290	11.7.3 Wait for Start of Packet from Upstream Port (WFSOPFU).....	317
291	11.7.4 Wait for End of Packet from Upstream Port (WFEOPFU)	317
292	11.7.5 Wait for Start of Packet (WFSOP).....	317
293	11.7.6 Wait for End of Packet (WFEOP)	317
294	11.8 Bus State Evaluation	317
295	11.8.1 Port Error	318
296	11.8.2 Speed Detection	318
297	11.8.3 Collision	318
298	11.8.4 Low-speed Port Behavior.....	319
299	11.9 Suspend and Resume	319
300	11.10 Hub Reset Behavior	321
301	11.11 Hub Port Power Control.....	322
302	11.11.1 Multiple Gangs	322
303	11.12 Hub Controller	323
304	11.12.1 Endpoint Organization	323
305	11.12.2 Hub Information Architecture and Operation	323
306	11.12.3 Port Change Information Processing	324
307	11.12.4 Hub and Port Status Change Bitmap	325
308	11.12.5 Over-current Reporting and Recovery.....	326
309	11.12.6 Enumeration Handling	327
310	11.13 Hub Configuration	327
311	11.14 Transaction Translator	328
312	11.14.1 Overview	328
313	11.14.2 Transaction Translator Scheduling	330
314	11.15 Split Transaction Notation Information	333
315	11.16 Common Split Transaction State Machines.....	335
316	11.16.1 Host Controller State Machine	336
317	11.16.2 Transaction Translator State Machine	340
318	11.17 Bulk/Control Transaction Translation Overview	345
319	11.17.1 Bulk/Control Split Transaction Sequences	346
320	11.17.2 Bulk/Control Split Transaction State Machines.....	352
321	11.17.3 Bulk/Control Sequencing	357

322	11.17.4 Bulk/Control Buffering Requirements	358
323	11.17.5 Other Bulk/Control Details	358
324	11.18 Periodic Split Transaction Pipelining and Buffer Management	358
325	11.18.1 Best Case Full-Speed Budget	359
326	11.18.2 TT Microframe Pipeline	359
327	11.18.3 Generation of Full-speed Frames	360
328	11.18.4 Host Split Transaction Scheduling Requirements	360
329	11.18.5 TT Response Generation	363
330	11.18.6 TT Periodic Transaction Handling Requirements	364
331	11.18.7 TT Transaction Tracking	366
332	11.18.8 TT Complete-split Transaction State Searching	367
333	11.19 Approximate TT Buffer Space Required	368
334	11.20 Interrupt Transaction Translation Overview	368
335	11.20.1 Interrupt Split Transaction Sequences	368
336	11.20.2 Interrupt Split Transaction State Machines	372
337	11.20.3 Interrupt OUT Sequencing	378
338	11.20.4 Interrupt IN Sequencing	379
339	11.21 Isochronous Transaction Translation Overview	380
340	11.21.1 Isochronous Split Transaction Sequences	381
341	11.21.2 Isochronous Split Transaction State Machines	384
342	11.21.3 Isochronous OUT Sequencing	388
343	11.21.4 Isochronous IN Sequencing	389
344	11.22 TT Error Handling	389
345	11.22.1 Loss of TT Synchronization With HS SOFs	389
346	11.22.2 TT Frame and Microframe Timer Synchronization Requirements	390
347	11.23 Descriptors	392
348	11.23.1 Standard Descriptors for Hub Class	392
349	11.23.2 Class-specific Descriptors	399
350	11.24 Requests	401
351	11.24.1 Standard Requests	401
352	11.24.2 Class-specific Requests	402
353	Appendix A Transaction Examples	418
354	A.1 Bulk/Control OUT and SETUP Transaction Examples	418
355	A.2 Bulk/Control IN Transaction Examples	444
356	A.3 Interrupt OUT Transaction Examples	469
357	A.4 Interrupt IN Transaction Examples	489
358	A.5 Isochronous OUT Split-transaction Examples	512
359	A.6 Isochronous IN Split-transaction Examples	521
360	Appendix B Example Declarations for State Machines	535
361	B.1 Global Declarations	535
362	B.2 Host Controller Declarations	538
363	B.3 Transaction Translator Declarations	540
364	Appendix C Reset Protocol State Diagrams	544
365	C.1 Downstream Facing Port State Diagram	544
366	C.2 Upstream Facing Port State Diagram	546
367	Index	551

— *Entwurf* —

– 13 –

368

369

Universal Serial Bus Specification

Compaq
Hewlett-Packard
Intel
Lucent
Microsoft
NEC
Philips

Revision 2.0
April 27, 2000

Scope of this Revision

The 2.0 revision of the specification is intended for product design. Every attempt has been made to ensure a consistent and implementable specification. Implementations should ensure compliance with this revision.

Revision History

Revision	Issue Date	Comments
0.7	November 11, 1994	Supersedes 0.6e.
0.8	December 30, 1994	Revisions to Chapters 3-8, 10, and 11. Added appendixes.
0.9	April 13, 1995	Revisions to all the chapters.
0.99	August 25, 1995	Revisions to all the chapters.
1.0 FDR	November 13, 1995	Revisions to Chapters 1, 2, 5-11.
1.0	January 15, 1996	Edits to Chapters 5, 6, 7, 8, 9, 10, and 11 for consistency..
1.1	September 23, 1998	Updates to all chapters to fix problems identified.
2.0 (draft 0.79)	October 5, 1999	Revisions to chapters 5, 7, 8, 9, 11 to add high speed.
2.0 (draft 0.9)	December 21, 1999	Revisions to all chapters to add high speed.
2.0	April 27, 2000	Revisions for high-speed mode.

Universal Serial Bus Specification
Copyright © 2000, Compaq Computer Corporation,
Hewlett-Packard Company,
Intel Corporation,
Lucent Technologies Inc,
Microsoft Corporation,
NEC Corporation,
Koninklijke Philips Electronics N.V.
All rights reserved.

399

INTELLECTUAL PROPERTY DISCLAIMER

400

401

402

403

404

405

406

THIS SPECIFICATION IS PROVIDED TO YOU "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

407

All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

408

Please send comments via electronic mail to techsup@usb.org

409

410

For industry information, refer to the USB Implementers Forum web page at <http://www.usb.org>

411

Acknowledgement of USB 2.0 Technical Contribution

412

413

414

415

The authors of this specification would like to recognize the following people who participated in the USB 2.0 Promoter Group technical working groups. We would also like to thank others in the USB 2.0 Promoter companies and throughout the industry who contributed to the development of this specification.

Hub Working Group

John Garney	Intel Corporation (Chair/Editor)
Ken Stufflebeam	Compaq Computer Corporation
David Wooten	Compaq Computer Corporation
Matt Nieberger	Hewlett-Packard Company
John Howard	Intel Corporation
Venkat Iyer	Intel Corporation
Steve McGowan	Intel Corporation
Geert Knapen	Royal Philips Electronics
Zong Liang Wu	Royal Philips Electronics
Jim Clee	Lucent Technologies Inc
Jim Guziak	Lucent Technologies Inc
Dave Thompson	Lucent Technologies Inc
John Fuller	Microsoft Corporation
Nathan Sherman	Microsoft Corporation
Mark Williams	Microsoft Corporation
Nobuo Furuya	NEC Corporation
Toshimi Sakurai	NEC Corporation
Moto Sato	NEC Corporation
Katsuya Suzuki	NEC Corporation

416

417

418

Electrical Working Group

Jon Lueker	Intel Corporation (Chair/Editor)
------------	----------------------------------

David Wooten	Compaq Computer Corporation
Matt Nieberger	Hewlett-Packard Company
Larry Taugher	Hewlett-Packard Company
Venkat Iyer	Intel Corporation
Steve McGowan	Intel Corporation
Mike Pennell	Intel Corporation
Todd West	Intel Corporation
Gerrit den Besten	Royal Philips Electronics
Marq Kole	Royal Philips Electronics
Zong Liang Wu	Royal Philips Electronics
Jim Clee	Lucent Technologies Inc
Jim Guziak	Lucent Technologies Inc
Par Parikh	Lucent Technologies Inc
Dave Thompson	Lucent Technologies Inc
Ed Giaimo	Microsoft Corporation
Mark Williams	Microsoft Corporation
Toshihiko Ohtani	NEC Corporation
Kugao Ouchi	NEC Corporation
Katsuya Suzuki	NEC Corporation
Toshio Tasaki	NEC Corporation

1 Chapter 1 Introduction

1.1 Motivation

The original motivation for the Universal Serial Bus (USB) came from three interrelated considerations:

- Connection of the PC to the telephone

It is well understood that the merge of computing and communication will be the basis for the next generation of productivity applications. The movement of machine-oriented and human-oriented data types from one location or environment to another depends on ubiquitous and cheap connectivity. Unfortunately, the computing and communication industries have evolved independently. The USB provides a ubiquitous link that can be used across a wide range of PC-to-telephone interconnects.

- Ease-of-use

The lack of flexibility in reconfiguring the PC has been acknowledged as the Achilles' heel to its further deployment. The combination of user-friendly graphical interfaces and the hardware and software mechanisms associated with new-generation bus architectures have made computers less confrontational and easier to reconfigure. However, from the end user's point of view, the PC's I/O interfaces, such as serial/parallel ports, keyboard/mouse/joystick interfaces, etc., do not have the attributes of plug-and-play.

- Port expansion

The addition of external peripherals continues to be constrained by port availability. The lack of a bi-directional, low-cost, low-to-mid speed peripheral bus has held back the creative proliferation of peripherals such as telephone/fax/modem adapters, answering machines, scanners, PDA's, keyboards, mice, etc. Existing interconnects are optimized for one or two point products. As each new function or capability is added to the PC, a new interface has been defined to address this need.

The more recent motivation for USB 2.0 stems from the fact that PCs have increasingly higher performance and are capable of processing vast amounts of data. At the same time, PC peripherals have added more performance and functionality. User applications such as digital imaging demand a high performance connection between the PC and these increasingly sophisticated peripherals. USB 2.0 addresses this need by adding a third transfer rate of 480 Mb/s to the 12 Mb/s and 1.5 Mb/s originally defined for USB. USB 2.0 is a natural evolution of USB, delivering the desired bandwidth increase while preserving the original motivations for USB and maintaining full compatibility with existing peripherals.

Thus, USB continues to be the answer to connectivity for the PC architecture. It is a fast, bi-directional, isochronous, low-cost, dynamically attachable serial interface that is consistent with the requirements of the PC platform of today and tomorrow.

1.2 Objective of the Specification

This document defines an industry-standard USB. The specification describes the bus attributes, the protocol definition, types of transactions, bus management, and the programming interface required to design and build systems and peripherals that are compliant with this standard.

The goal is to enable such devices from different vendors to interoperate in an open architecture. The specification is intended as an enhancement to the PC architecture, spanning portable, business desktop, and home environments. It is intended that the specification allow system OEMs and peripheral developers adequate room for product versatility and market differentiation without the burden of carrying obsolete interfaces or losing compatibility.

1.3 Scope of the Document

The specification is primarily targeted to peripheral developers and system OEMs, but provides valuable information for platform operating system/ BIOS/ device driver, adapter IHVs/ISVs, and platform/adaptor controller vendors. This specification can be used for developing new products and associated software.

1.4 USB Product Compliance

Adopters of the USB 2.0 specification have signed the USB 2.0 Adopters Agreement, which provides them access to a reciprocal royalty-free license from the Promoters and other Adopters to certain intellectual property contained in products that are compliant with the USB 2.0 specification. Adopters can demonstrate compliance with the specification through the testing program as defined by the USB Implementers Forum. Products that demonstrate compliance with the specification will be granted certain rights to use the USB Implementers Forum logo as defined in the logo license.

1.5 Document Organization

Chapters 1 through 5 provide an overview for all readers, while Chapters 6 through 11 contain detailed technical information defining the USB.

- Peripheral implementers should particularly read Chapters 5 through 11.
- USB Host Controller implementers should particularly read Chapters 5 through 8, 10, and 11.
- USB device driver implementers should particularly read Chapters 5, 9, and 10.

This document is complemented and referenced by the Universal Serial Bus Device Class Specifications. Device class specifications exist for a wide variety of devices. Please contact the USB Implementers Forum for further details.

Readers are also requested to contact operating system vendors for operating system bindings specific to the USB.

494 **2 Chapter 2**
495 **Terms and Abbreviations**

496 This chapter lists and defines terms and abbreviations used throughout this specification.

ACK	Handshake packet indicating a positive acknowledgment.
Active Device	A device that is powered and is not in the Suspend state.
Asynchronous Data	Data transferred at irregular intervals with relaxed latency requirements.
Asynchronous RA	The incoming data rate, F_{s_i} , and the outgoing data rate, F_{s_o} , of the RA process are independent (i.e., there is no shared master clock). See also rate adaptation.
Asynchronous SRC	The incoming sample rate, F_{s_i} , and outgoing sample rate, F_{s_o} , of the SRC process are independent (i.e., there is no shared master clock). See also sample rate conversion.
Audio Device	A device that sources or sinks sampled analog data.
AWG#	The measurement of a wire's cross section, as defined by the American Wire Gauge standard.
Babble	Unexpected bus activity that persists beyond a specified point in a (micro)frame.
Bandwidth	The amount of data transmitted per unit of time, typically bits per second (b/s) or bytes per second (B/s).
Big Endian	A method of storing data that places the most significant byte of multiple-byte values at a lower storage address. For example, a 16-bit integer stored in big endian format places the least significant byte at the higher address and the most significant byte at the lower address. See also little endian.
Bit	A unit of information used by digital computers. Represents the smallest piece of addressable memory within a computer. A bit expresses the choice between two possibilities and is typically represented by a logical one (1) or zero (0).
Bit Stuffing	Insertion of a "0" bit into a data stream to cause an electrical transition on the data wires, allowing a PLL to remain locked.
b/s	Transmission rate expressed in bits per second.
B/s	Transmission rate expressed in bytes per second.
Buffer	Storage used to compensate for a difference in data rates or time of occurrence of events, when transmitting data from one device to another.
Bulk Transfer	One of the four USB transfer types. Bulk transfers are non-periodic, large bursty communication typically used for a transfer that can use any available bandwidth and can also be delayed until bandwidth is available. See also transfer type.
Bus Enumeration	Detecting and identifying USB devices.
Byte	A data element that is eight bits in size.
Capabilities	Those attributes of a USB device that are administrated by the host.
Characteristics	Those qualities of a USB device that are unchangeable; for example, the device class is a device characteristic.
Client	Software resident on the host that interacts with the USB System Software to arrange data transfer between a function and the host. The client is often the data provider and consumer for transferred data.
Configuring Software	Software resident on the host software that is responsible for configuring a USB device. This may be a system configurator or software specific to the device.
Control Endpoint	A pair of device endpoints with the same endpoint number that are used by a control pipe. Control endpoints transfer data in both directions and, therefore, use both endpoint directions of a device address and endpoint number combination. Thus, each control endpoint consumes two endpoint addresses.
Control Pipe	Same as a message pipe.
Control Transfer	One of the four USB transfer types. Control transfers support configuration/command/status type communications between client and function. See also transfer type.
CRC	See Cyclic Redundancy Check.

CTI	Computer Telephony Integration.
Cyclic Redundancy Check (CRC)	A check performed on data to see if an error has occurred in transmitting, reading, or writing the data. The result of a CRC is typically stored or transmitted with the checked data. The stored or transmitted result is compared to a CRC calculated for the data to determine if an error has occurred.
Default Address	An address defined by the USB Specification and used by a USB device when it is first powered or reset. The default address is 00H.
Default Pipe	The message pipe created by the USB System Software to pass control and status information between the host and a USB device's endpoint zero.
Device	A logical or physical entity that performs a function. The actual entity described depends on the context of the reference. At the lowest level, device may refer to a single hardware component, as in a memory device. At a higher level, it may refer to a collection of hardware components that perform a particular function, such as a USB interface device. At an even higher level, device may refer to the function performed by an entity attached to the USB; for example, a data/FAX modem device. Devices may be physical, electrical, addressable, and logical. When used as a non-specific reference, a USB device is either a hub or a function.
Device Address	A seven-bit value representing the address of a device on the USB. The device address is the default address (00H) when the USB device is first powered or the device is reset. Devices are assigned a unique device address by the USB System Software.
Device Endpoint	A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device. See also endpoint address.
Device Resources	Resources provided by USB devices, such as buffer space and endpoints. See also Host Resources and Universal Serial Bus Resources.
Device Software	Software that is responsible for using a USB device. This software may or may not also be responsible for configuring the device for use.
Downstream	The direction of data flow from the host or away from the host. A downstream port is the port on a hub electrically farthest from the host that generates downstream data traffic from the hub. Downstream ports receive upstream data traffic.
Driver	When referring to hardware, an I/O pad that drives an external load. When referring to software, a program responsible for interfacing to a hardware device, that is, a device driver.
DWORD	Double word. A data element that is two words (i.e., four bytes or 32 bits) in size.
Dynamic Insertion and Removal	The ability to attach and remove devices while the host is in operation.
E ² PROM	See Electrically Erasable Programmable Read Only Memory.
EEPROM	See Electrically Erasable Programmable Read Only Memory.
Electrically Erasable Programmable Read Only Memory (EEPROM)	Non-volatile rewritable memory storage technology.
End User	The user of a host.
Endpoint	See device endpoint.
Endpoint Address	The combination of an endpoint number and an endpoint direction on a USB device. Each endpoint address supports data transfer in one direction.
Endpoint Direction	The direction of data transfer on the USB. The direction can be either IN or OUT. IN refers to transfers to the host; OUT refers to transfers from the host.
Endpoint Number	A four-bit value between 0H and FH, inclusive, associated with an endpoint on a USB device.
Envelope detector	An electronic circuit inside a USB device that monitors the USB data lines and detects certain voltage related signal characteristics.
EOF	End-of-(micro)Frame.
EOP	End-of-Packet.
External Port	See port.

Eye pattern	A representation of USB signaling that provides minimum and maximum voltage levels as well as signal jitter.
False EOP	A spurious, usually noise-induced event that is interpreted by a packet receiver as an EOP.
Frame	A 1 millisecond time base established on full-/low-speed buses.
Frame Pattern	A sequence of frames that exhibit a repeating pattern in the number of samples transmitted per frame. For a 44.1 kHz audio transfer, the frame pattern could be nine frames containing 44 samples followed by one frame containing 45 samples.
F _s	See sample rate.
Full-duplex	Computer data transmission occurring in both directions simultaneously.
Full-speed	USB operation at 12 Mb/s. See also low-speed and high-speed.
Function	A USB device that provides a capability to the host, such as an ISDN connection, a digital microphone, or speakers.
Handshake Packet	A packet that acknowledges or rejects a specific condition. For examples, see ACK and NAK.
High-bandwidth endpoint	A high-speed device endpoint that transfers more than 1024 bytes and less than 3073 bytes per microframe.
High-speed	USB operation at 480 Mb/s. See also low-speed and full-speed.
Host	The host computer system where the USB Host Controller is installed. This includes the host hardware platform (CPU, bus, etc.) and the operating system in use.
Host Controller	The host's USB interface.
Host Controller Driver (HCD)	The USB software layer that abstracts the Host Controller hardware. The Host Controller Driver provides an SPI for interaction with a Host Controller. The Host Controller Driver hides the specifics of the Host Controller hardware implementation.
Host Resources	Resources provided by the host, such as buffer space and interrupts. See also Device Resources and Universal Serial Bus Resources.
Hub	A USB device that provides additional connections to the USB.
Hub Tier	One plus the number of USB links in a communication path between the host and a function. See Figure 4-1.
Interrupt Request (IRQ)	A hardware signal that allows a device to request attention from a host. The host typically invokes an interrupt service routine to handle the condition that caused the request.
Interrupt Transfer	One of the four USB transfer types. Interrupt transfer characteristics are small data, non-periodic, low-frequency, and bounded-latency. Interrupt transfers are typically used to handle service needs. See also transfer type.
I/O Request Packet	An identifiable request by a software client to move data between itself (on the host) and an endpoint of a device in an appropriate direction.
IRP	See I/O Request Packet.
IRQ	See Interrupt Request.
Isochronous Data	A stream of data whose timing is implied by its delivery rate.
Isochronous Device	An entity with isochronous endpoints, as defined in the USB Specification, that sources or sinks sampled analog streams or synchronous data streams.
Isochronous Sink Endpoint	An endpoint that is capable of consuming an isochronous data stream that is sent by the host.
Isochronous Source Endpoint	An endpoint that is capable of producing an isochronous data stream and sending it to the host.
Isochronous Transfer	One of the four USB transfer types. Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device. See also transfer type.
Jitter	A tendency toward lack of synchronization caused by mechanical or electrical changes. More specifically, the phase shift of digital pulses over a transmission medium.
kb/s	Transmission rate expressed in kilobits per second.

kB/s	Transmission rate expressed in kilobytes per second.
Little Endian	Method of storing data that places the least significant byte of multiple-byte values at lower storage addresses. For example, a 16-bit integer stored in little endian format places the least significant byte at the lower address and the most significant byte at the next address. See also big endian.
LOA	Loss of bus activity characterized by an SOP without a corresponding EOP.
Low-speed	USB operation at 1.5 Mb/s. See also full-speed and high-speed.
LSb	Least significant bit.
LSB	Least significant byte.
Mb/s	Transmission rate expressed in megabits per second.
MB/s	Transmission rate expressed in megabytes per second.
Message Pipe	A bi-directional pipe that transfers data using a request/data/status paradigm. The data has an imposed structure that allows requests to be reliably identified and communicated.
Microframe	A 125 microsecond time base established on high-speed buses.
MSb	Most significant bit.
MSB	Most significant byte.
NAK	Handshake packet indicating a negative acknowledgment.
Non Return to Zero Invert (NRZI)	A method of encoding serial data in which ones and zeroes are represented by opposite and alternating high and low voltages where there is no return to zero (reference) voltage between encoded bits. Eliminates the need for clock pulses.
NRZI	See Non Return to Zero Invert.
Object	Host software or data structure representing a USB entity.
Packet	A bundle of data organized in a group for transmission. Packets typically contain three elements: control information (e.g., source, destination, and length), the data to be transferred, and error detection and correction bits.
Packet Buffer	The logical buffer used by a USB device for sending or receiving a single packet. This determines the maximum packet size the device can send or receive.
Packet ID (PID)	A field in a USB packet that indicates the type of packet, and by inference, the format of the packet and the type of error detection applied to the packet.
Phase	A token, data, or handshake packet. A transaction has three phases.
Phase Locked Loop (PLL)	A circuit that acts as a phase detector to keep an oscillator in phase with an incoming frequency.
Physical Device	A device that has a physical implementation; e.g., speakers, microphones, and CD players.
PID	See Packet ID.
Pipe	A logical abstraction representing the association between an endpoint on a device and software on the host. A pipe has several attributes; for example, a pipe may transfer data as streams (stream pipe) or messages (message pipe). See also stream pipe and message pipe.
PLL	See Phase Locked Loop.
Polling	Asking multiple devices, one at a time, if they have any data to transmit.
POR	See Power On Reset.
Port	Point of access to or from a system or circuit. For the USB, the point where a USB device is attached.
Power On Reset (POR)	Restoring a storage device, register, or memory to a predetermined state when power is applied.
Programmable Data Rate	Either a fixed data rate (single-frequency endpoints), a limited number of data rates (32 kHz, 44.1 kHz, 48 kHz, ...), or a continuously programmable data rate. The exact programming capabilities of an endpoint must be reported in the appropriate class-specific endpoint descriptors.

Protocol	A specific set of rules, procedures, or conventions relating to format and timing of data transmission between two devices.
RA	See rate adaptation.
Rate Adaptation	The process by which an incoming data stream, sampled at F_{s_i} , is converted to an outgoing data stream, sampled at F_{s_o} , with a certain loss of quality, determined by the rate adaptation algorithm. Error control mechanisms are required for the process. F_{s_i} and F_{s_o} can be different and asynchronous. F_{s_i} is the input data rate of the RA; F_{s_o} is the output data rate of the RA.
Request	A request made to a USB device contained within the data portion of a SETUP packet.
Retire	The action of completing service for a transfer and notifying the appropriate software client of the completion.
Root Hub	A USB hub directly attached to the Host Controller. This hub (tier 1) is attached to the host.
Root Port	The downstream port on a Root Hub.
Sample	The smallest unit of data on which an endpoint operates; a property of an endpoint.
Sample Rate (Fs)	The number of samples per second, expressed in Hertz (Hz).
Sample Rate Conversion (SRC)	A dedicated implementation of the RA process for use on sampled analog data streams. The error control mechanism is replaced by interpolating techniques.
Service	A procedure provided by a System Programming Interface (SPI).
Service Interval	The period between consecutive requests to a USB endpoint to send or receive data.
Service Jitter	The deviation of service delivery from its scheduled delivery time.
Service Rate	The number of services to a given endpoint per unit time.
SOF	See Start-of-Frame.
SOP	Start-of-Packet.
SPI	See System Programming Interface.
Split transaction	A transaction type supported by host controllers and hubs. This transaction type allows full- and low-speed devices to be attached to hubs operating at high-speed.
SRC	See Sample Rate Conversion.
Stage	One part of the sequence composing a control transfer; stages include the Setup stage, the Data stage, and the Status stage.
Start-of-Frame (SOF)	The first transaction in each (micro)frame. An SOF allows endpoints to identify the start of the (micro)frame and synchronize internal endpoint clocks to the host.
Stream Pipe	A pipe that transfers data as a stream of samples with no defined USB structure.
Synchronization Type	A classification that characterizes an isochronous endpoint's capability to connect to other isochronous endpoints.
Synchronous RA	The incoming data rate, F_{s_i} , and the outgoing data rate, F_{s_o} , of the RA process are derived from the same master clock. There is a fixed relation between F_{s_i} and F_{s_o} .
Synchronous SRC	The incoming sample rate, F_{s_i} , and outgoing sample rate, F_{s_o} , of the SRC process are derived from the same master clock. There is a fixed relation between F_{s_i} and F_{s_o} .
System Programming Interface (SPI)	A defined interface to services provided by system software.
TDM	See Time Division Multiplexing.
TDR	See Time Domain Reflectometer.
Termination	Passive components attached at the end of cables to prevent signals from being reflected or echoed.
Time Division Multiplexing (TDM)	A method of transmitting multiple signals (data, voice, and/or video) simultaneously over one communications medium by interleaving a piece of each signal one after another.

Time Domain Reflectometer (TDR)	An instrument capable of measuring impedance characteristics of the USB signal lines.
Timeout	The detection of a lack of bus activity for some predetermined interval.
Token Packet	A type of packet that identifies what transaction is to be performed on the bus.
Transaction	The delivery of service to an endpoint; consists of a token packet, optional data packet, and optional handshake packet. Specific packets are allowed/required based on the transaction type.
Transaction translator	A functional component of a USB hub. The Transaction Translator responds to special high-speed transactions and translates them to full/low-speed transactions with full/low-speed devices attached on downstream facing ports.
Transfer	One or more bus transactions to move information between a software client and its function.
Transfer Type	Determines the characteristics of the data flow between a software client and its function. Four standard transfer types are defined: control, interrupt, bulk, and isochronous.
Turn-around Time	The time a device needs to wait to begin transmitting a packet after a packet has been received to prevent collisions on the USB. This time is based on the length and propagation delay characteristics of the cable and the location of the transmitting device in relation to other devices on the USB.
Universal Serial Bus Driver (USBD)	The host resident software entity responsible for providing common services to clients that are manipulating one or more functions on one or more Host Controllers.
Universal Serial Bus Resources	Resources provided by the USB, such as bandwidth and power. See also Device Resources and Host Resources.
Upstream	The direction of data flow towards the host. An upstream port is the port on a device electrically closest to the host that generates upstream data traffic from the hub. Upstream ports receive downstream data traffic.
USBD	See Universal Serial Bus Driver.
USB-IF	USB Implementers Forum, Inc. is a nonprofit corporation formed to facilitate the development of USB compliant products and promote the technology.
Virtual Device	A device that is represented by a software interface layer. An example of a virtual device is a hard disk with its associated device driver and client software that makes it able to reproduce an audio .WAV file.
Word	A data element that is two bytes (16 bits) in size.

Chapter 3 Background

This chapter presents a brief description of the background of the Universal Serial Bus (USB), including design goals, features of the bus, and existing technologies.

3.1 Goals for the Universal Serial Bus

The USB is specified to be an industry-standard extension to the PC architecture with a focus on PC peripherals that enable consumer and business applications. The following criteria were applied in defining the architecture for the USB:

- Ease-of-use for PC peripheral expansion
- Low-cost solution that supports transfer rates up to 480 Mb/s
- Full support for real-time data for voice, audio, and video
- Protocol flexibility for mixed-mode isochronous data transfers and asynchronous messaging
- Integration in commodity device technology
- Comprehension of various PC configurations and form factors
- Provision of a standard interface capable of quick diffusion into product
- Enabling new classes of devices that augment the PC's capability
- Full backward compatibility of USB 2.0 for devices built to previous versions of the specification

3.2 Taxonomy of Application Space

Figure 3-1 describes a taxonomy for the range of data traffic workloads that can be serviced over a USB. As can be seen, a 480 Mb/s bus comprehends the high-speed, full-speed, and low-speed data ranges. Typically, high-speed and full-speed data types may be isochronous, while low-speed data comes from interactive devices. The USB is primarily a PC bus but can be readily applied to other host-centric computing devices. The software architecture allows for future extension of the USB by providing support for multiple USB Host Controllers.

PERFORMANCE	APPLICATIONS	ATTRIBUTES
LOW-SPEED • Interactive Devices • 10 – 100 kb/s	Keyboard, Mouse Stylus Game Peripherals Virtual Reality Peripherals	Lowest Cost Ease-of-Use Dynamic Attach-Detach Multiple Peripherals
FULL-SPEED • Phone, Audio, Compressed Video • 500 kb/s – 10 Mb/s	POTS Broadband Audio Microphone	Lower Cost Ease-of-Use Dynamic Attach-Detach Multiple Peripherals Guaranteed Bandwidth Guaranteed Latency
HIGH-SPEED • Video, Storage • 25 – 400 Mb/s	Video Storage Imaging Broadband	Low Cost Ease-of-Use Dynamic Attach-Detach Multiple Peripherals Guaranteed Bandwidth Guaranteed Latency High Bandwidth

Figure 3-1. Application Space Taxonomy

526 3.3 Feature List

527 The USB Specification provides a selection of attributes that can achieve multiple
528 price/performance integration points and can enable functions that allow differentiation at the
529 system and component level. Features are categorized by the following benefits:

530 Easy to use for end user

- 531 • Single model for cabling and connectors
- 532 • Electrical details isolated from end user (e.g., bus terminations)
- 533 • Self-identifying peripherals, automatic mapping of function to driver and configuration
- 534 • Dynamically attachable and reconfigurable peripherals

535 Wide range of workloads and applications

- 536 • Suitable for device bandwidths ranging from a few kb/s to several hundred Mb/s
- 537 • Supports isochronous as well as asynchronous transfer types over the same set of wires
- 538 • Supports concurrent operation of many devices (multiple connections)
- 539 • Supports up to 127 physical devices
- 540 • Supports transfer of multiple data and message streams between the host and devices
- 541 • Allows compound devices (i.e., peripherals composed of many functions)
- 542 • Lower protocol overhead, resulting in high bus utilization

543 Isochronous bandwidth

- 544 • Guaranteed bandwidth and low latencies appropriate for telephony, audio, video, etc.

545 Flexibility

- 546 • Supports a wide range of packet sizes, which allows a range of device buffering options
- 547 • Allows a wide range of device data rates by accommodating packet buffer size and
548 latencies
- 549 • Flow control for buffer handling is built into the protocol

550 Robustness

- 551 • Error handling/fault recovery mechanism is built into the protocol
- 552 • Dynamic insertion and removal of devices is identified in user-perceived real-time
- 553 • Supports identification of faulty devices

554 Synergy with PC industry

- 555 • Protocol is simple to implement and integrate
- 556 • Consistent with the PC plug-and-play architecture
- 557 • Leverages existing operating system interfaces

558 Low-cost implementation

- 559 • Low-cost subchannel at 1.5 Mb/s
- 560 • Optimized for integration in peripheral and host hardware
- 561 • Suitable for development of low-cost peripherals
- 562 • Low-cost cables and connectors
- 563 • Uses commodity technologies

564 **Upgrade path**

- 565 • Architecture upgradeable to support multiple USB Host Controllers in a system

566

4 Chapter 4 Architectural Overview

This chapter presents an overview of the Universal Serial Bus (USB) architecture and key concepts. The USB is a cable bus that supports data exchange between a host computer and a wide range of simultaneously accessible peripherals. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. The bus allows peripherals to be attached, configured, used, and detached while the host and other peripherals are in operation.

Later chapters describe the various components of the USB in greater detail.

4.1 USB System Description

A USB system is described by three definitional areas:

- USB interconnect
- USB devices
- USB host

The USB interconnect is the manner in which USB devices are connected to and communicate with the host. This includes the following:

- Bus Topology: Connection model between USB devices and the host.
- Inter-layer Relationships: In terms of a capability stack, the USB tasks that are performed at each layer in the system.
- Data Flow Models: The manner in which data moves in the system over the USB between producers and consumers.
- USB Schedule: The USB provides a shared interconnect. Access to the interconnect is scheduled in order to support isochronous data transfers and to eliminate arbitration overhead.

USB devices and the USB host are described in detail in subsequent sections.

4.1.1 Bus Topology

The USB connects USB devices with the USB host. The USB physical interconnect is a tiered star topology. A hub is at the center of each star. Each wire segment is a point-to-point connection between the host and a hub or function, or a hub connected to another hub or function. Figure 4-1 illustrates the topology of the USB.

Due to timing constraints allowed for hub and cable propagation times, the maximum number of tiers allowed is seven (including the root tier). Note that in seven tiers, five non-root hubs maximum can be supported in a communication path between the host and any device. A compound device (see Figure 4-1) occupies two tiers; therefore, it cannot be enabled if attached at tier level seven. Only functions can be enabled in tier seven.

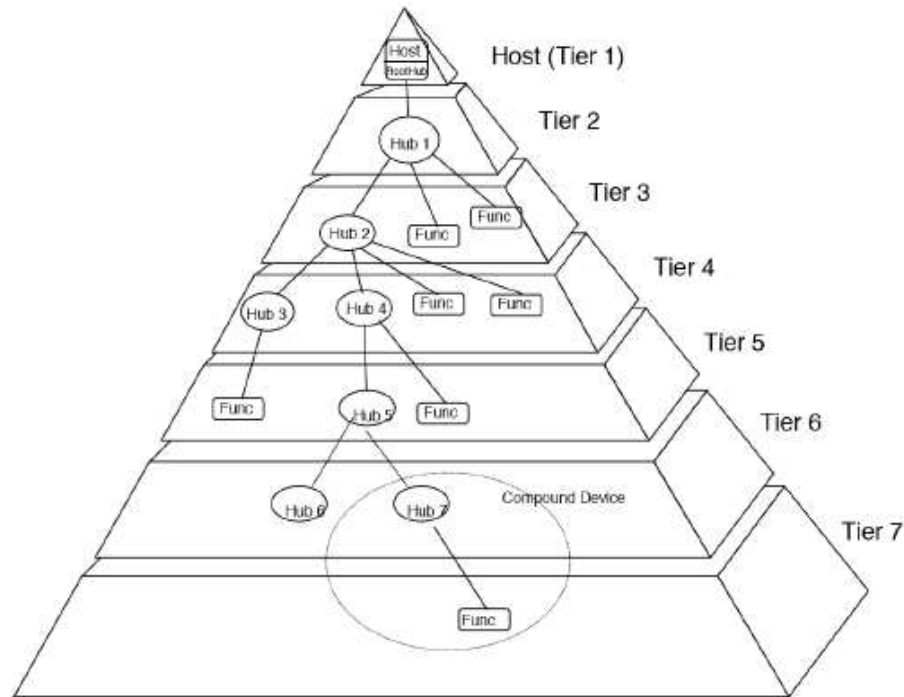


Figure 4-1. Bus Topology

4.1.1.1 USB Host

There is only one host in any USB system. The USB interface to the host computer system is referred to as the Host Controller. The Host Controller may be implemented in a combination of hardware, firmware, or software. A root hub is integrated within the host system to provide one or more attachment points.

Additional information concerning the host may be found in Section 4.9 and in Chapter 10.

4.1.1.2 USB Devices

USB devices are one of the following:

- Hubs, which provide additional attachment points to the USB
- Functions, which provide capabilities to the system, such as an ISDN connection, a digital joystick, or speakers

USB devices present a standard USB interface in terms of the following:

- Their comprehension of the USB protocol
- Their response to standard USB operations, such as configuration and reset
- Their standard capability descriptive information

Additional information concerning USB devices may be found in Section 4.8 and in Chapter 9.

4.2 Physical Interface

The physical interface of the USB is described in the electrical (Chapter 7) and mechanical (Chapter 6) specifications for the bus.

4.2.1 Electrical

The USB transfers signal and power over a four-wire cable, shown in Figure 4-2. The signaling occurs over two wires on each point-to-point segment.

There are three data rates:

- 627 • The USB high-speed signaling bit rate is 480 Mb/s.
 - 628 • The USB full-speed signaling bit rate is 12 Mb/s.
 - 629 • A limited capability low-speed signaling mode is also defined at 1.5 Mb/s.
- 630 USB 2.0 host controllers and hubs provide capabilities so that full-speed and low-speed data
 631 can be transmitted at high-speed between the host controller and the hub, but transmitted
 632 between the hub and the device at full-speed or low-speed. This capability minimizes the
 633 impact that full-speed and low-speed devices have upon the bandwidth available for high-
 634 speed devices.
- 635 The low-speed mode is defined to support a limited number of low-bandwidth devices, such
 636 as mice, because more general use would degrade bus utilization.
- 637 The clock is transmitted, encoded along with the differential data. The clock encoding
 638 scheme is NRZI with bit stuffing to ensure adequate transitions. A SYNC field precedes each
 639 packet to allow the receiver(s) to synchronize their bit recovery clocks.

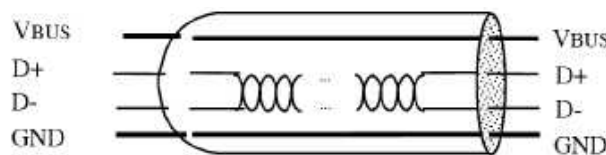


Figure 4-2. USB Cable

642 The cable also carries VBUS and GND wires on each segment to deliver power to devices.
 643 VBUS is nominally +5 V at the source. The USB allows cable segments of variable lengths,
 644 up to several meters, by choosing the appropriate conductor gauge to match the specified IR
 645 drop and other attributes such as device power budget and cable flexibility. In order to
 646 provide guaranteed input voltage levels and proper termination impedance, biased
 647 terminations are used at each end of the cable. The terminations also permit the detection of
 648 attach and detach at each port and differentiate between high/full-speed and low-speed
 649 devices.

650 4.2.2 Mechanical

651 The mechanical specifications for cables and connectors are provided in Chapter 6. All
 652 devices have an upstream connection. Upstream and downstream connectors are not
 653 mechanically interchangeable, thus eliminating illegal loopback connections at hubs. The
 654 cable has four conductors: a twisted signal pair of standard gauge and a power pair in a
 655 range of permitted gauges. The connector is four-position, with shielded housing, specified
 656 robustness, and ease of attach-detach characteristics.

657 4.3 Power

658 The specification covers two aspects of power:

- 659 • Power distribution over the USB deals with the issues of how USB devices consume
 660 power provided by the host over the USB.
- 661 • Power management deals with how the USB System Software and devices fit into the
 662 host-based power management system.

663 4.3.1 Power Distribution

664 Each USB segment provides a limited amount of power over the cable. The host supplies
 665 power for use by USB devices that are directly connected. In addition, any USB device may
 666 have its own power supply. USB devices that rely totally on power from the cable are called
 667 bus-powered devices. In contrast, those that have an alternate source of power are called
 668 self-powered devices. A hub also supplies power for its connected USB devices. The
 669 architecture permits bus-powered hubs within certain constraints of topology that are
 670 discussed later in Chapter 11.

671 4.3.2 Power Management

672 A USB host may have a power management system that is independent of the USB. The USB
673 System Software interacts with the host's power management system to handle system power
674 events such as suspend or resume. Additionally, USB devices typically implement additional
675 power management features that allow them to be power managed by system software.

676 The power distribution and power management features of the USB allow it to be designed
677 into power-sensitive systems such as battery-based notebook computers.

678 4.4 Bus Protocol

679 The USB is a polled bus. The Host Controller initiates all data transfers.

680 Most bus transactions involve the transmission of up to three packets. Each transaction
681 begins when the Host Controller, on a scheduled basis, sends a USB packet describing the
682 type and direction of transaction, the USB device address, and endpoint number. This packet
683 is referred to as the "token packet." The USB device that is addressed selects itself by
684 decoding the appropriate address fields. In a given transaction, data is transferred either
685 from the host to a device or from a device to the host. The direction of data transfer is
686 specified in the token packet. The source of the transaction then sends a data packet or
687 indicates it has no data to transfer. The destination, in general, responds with a handshake
688 packet indicating whether the transfer was successful.

689 Some bus transactions between host controllers and hubs involve the transmission of four
690 packets. These types of transactions are used to manage the data transfers between the host
691 and full-/low- speed devices.

692 The USB data transfer model between a source or destination on the host and an endpoint on
693 a device is referred to as a pipe. There are two types of pipes: stream and message.
694 Stream data has no USB-defined structure, while message data does. Additionally, pipes
695 have associations of data bandwidth, transfer service type, and endpoint characteristics like
696 directionality and buffer sizes. Most pipes come into existence when a USB device is
697 configured. One message pipe, the Default Control Pipe, always exists once a device is
698 powered, in order to provide access to the device's configuration, status, and control
699 information.

700 The transaction schedule allows flow control for some stream pipes. At the hardware level,
701 this prevents buffers from underrun or overrun situations by using a NAK handshake to
702 throttle the data rate. When NAKed, a transaction is retried when bus time is available. The
703 flow control mechanism permits the construction of flexible schedules that accommodate
704 concurrent servicing of a heterogeneous mix of stream pipes. Thus, multiple stream pipes
705 can be serviced at different intervals and with packets of different sizes.

706 4.5 Robustness

707 There are several attributes of the USB that contribute to its robustness:

- 708 • Signal integrity using differential drivers, receivers, and shielding
- 709 • CRC protection over control and data fields
- 710 • Detection of attach and detach and system-level configuration of resources
- 711 • Self-recovery in protocol, using timeouts for lost or corrupted packets
- 712 • Flow control for streaming data to ensure isochrony and hardware buffer management
- 713 • Data and control pipe constructs for ensuring independence from adverse interactions
- 714 between functions

715 4.5.1 Error Detection

716 The core bit error rate of the USB medium is expected to be close to that of a backplane and
717 any glitches will very likely be transient in nature. To provide protection against such
718 transients, each packet includes error protection fields. When data integrity is required, such
719 as with lossless data devices, an error recovery procedure may be invoked in hardware or
720 software.

721 The protocol includes separate CRCs for control and data fields of each packet. A failed CRC
722 is considered to indicate a corrupted packet. The CRC gives 100% coverage on single- and
723 double-bit errors.

724 4.5.2 Error Handling

725 The protocol allows for error handling in hardware or software. Hardware error handling
726 includes reporting and retry of failed transfers. A USB Host Controller will try a transmission
727 that encounters errors up to three times before informing the client software of the failure.
728 The client software can recover in an implementation-specific way.

729 4.6 System Configuration

730 The USB supports USB devices attaching to and detaching from the USB at any time.
731 Consequently, system software must accommodate dynamic changes in the physical bus
732 topology.

733 4.6.1 Attachment of USB Devices

734 All USB devices attach to the USB through ports on specialized USB devices known as hubs.
735 Hubs have status bits that are used to report the attachment or removal of a USB device on
736 one of its ports. The host queries the hub to retrieve these bits. In the case of an attachment,
737 the host enables the port and addresses the USB device through the device's control pipe at
738 the default address.

739 The host assigns a unique USB address to the device and then determines if the newly
740 attached USB device is a hub or a function. The host establishes its end of the control pipe
741 for the USB device using the assigned USB address and endpoint number zero.

742 If the attached USB device is a hub and USB devices are attached to its ports, then the above
743 procedure is followed for each of the attached USB devices.

744 If the attached USB device is a function, then attachment notifications will be handled by host
745 software that is appropriate for the function.

746 4.6.2 Removal of USB Devices

747 When a USB device has been removed from one of a hub's ports, the hub disables the port
748 and provides an indication of device removal to the host. The removal indication is then
749 handled by appropriate USB System Software. If the removed USB device is a hub, the USB
750 System Software must handle the removal of both the hub and of all of the USB devices that
751 were previously attached to the system through the hub.

752 4.6.3 Bus Enumeration

753 Bus enumeration is the activity that identifies and assigns unique addresses to devices
754 attached to a bus. Because the USB allows USB devices to attach to or detach from the USB
755 at any time, bus enumeration is an on-going activity for the USB System Software.
756 Additionally, bus enumeration for the USB also includes the detection and processing of
757 removals.

758 4.7 Data Flow Types

759 The USB supports functional data and control exchange between the USB host and a USB
760 device as a set of either uni-directional or bi-directional pipes. USB data transfers take place

between host software and a particular endpoint on a USB device. Such associations between the host software and a USB device endpoint are called pipes. In general, data movement through one pipe is independent from the data flow in any other pipe. A given USB device may have many pipes. As an example, a given USB device could have an endpoint that supports a pipe for transporting data to the USB device and another endpoint that supports a pipe for transporting data from the USB device.

The USB architecture comprehends four basic types of data transfers:

- Control Transfers: Used to configure a device at attach time and can be used for other device-specific purposes, including control of other pipes on the device.
- Bulk Data Transfers: Generated or consumed in relatively large and bursty quantities and have wide dynamic latitude in transmission constraints.
- Interrupt Data Transfers: Used for timely but reliable delivery of data, for example, characters or coordinates with human-perceptible echo or feedback response characteristics.
- Isochronous Data Transfers: Occupy a prenegotiated amount of USB bandwidth with a prenegotiated delivery latency. (Also called streaming real time transfers).

A pipe supports only one of the types of transfers described above for any given device configuration. The USB data flow model is described in more detail in Chapter 5.

4.7.1 Control Transfers

Control data is used by the USB System Software to configure devices when they are first attached. Other driver software can choose to use control transfers in implementation-specific ways. Data delivery is lossless.

4.7.2 Bulk Transfers

Bulk data typically consists of larger amounts of data, such as that used for printers or scanners. Bulk data is sequential. Reliable exchange of data is ensured at the hardware level by using error detection in hardware and invoking a limited number of retries in hardware. Also, the bandwidth taken up by bulk data can vary, depending on other bus activities.

4.7.3 Interrupt Transfers

A limited-latency transfer to or from a device is referred to as interrupt data. Such data may be presented for transfer by a device at any time and is delivered by the USB at a rate no slower than is specified by the device.

Interrupt data typically consists of event notification, characters, or coordinates that are organized as one or more bytes. An example of interrupt data is the coordinates from a pointing device. Although an explicit timing rate is not required, interactive data may have response time bounds that the USB must support.

4.7.4 Isochronous Transfers

Isochronous data is continuous and real-time in creation, delivery, and consumption. Timing-related information is implied by the steady rate at which isochronous data is received and transferred. Isochronous data must be delivered at the rate received to maintain its timing. In addition to delivery rate, isochronous data may also be sensitive to delivery delays. For isochronous pipes, the bandwidth required is typically based upon the sampling characteristics of the associated function. The latency required is related to the buffering available at each endpoint.

A typical example of isochronous data is voice. If the delivery rate of these data streams is not maintained, drop-outs in the data stream will occur due to buffer or frame underruns or overruns. Even if data is delivered at the appropriate rate by USB hardware, delivery delays

807 introduced by software may degrade applications requiring real-time turn-around, such as
808 telephony-based audio conferencing.

809 The timely delivery of isochronous data is ensured at the expense of potential transient losses
810 in the data stream. In other words, any error in electrical transmission is not corrected by
811 hardware mechanisms such as retries. In practice, the core bit error rate of the USB is
812 expected to be small enough not to be an issue. USB isochronous data streams are allocated
813 a dedicated portion of USB bandwidth to ensure that data can be delivered at the desired rate.
814 The USB is also designed for minimal delay of isochronous data transfers.

815 **4.7.5 Allocating USB Bandwidth**

816 USB bandwidth is allocated among pipes. The USB allocates bandwidth for some pipes when
817 a pipe is established. USB devices are required to provide some buffering of data. It is
818 assumed that USB devices requiring more bandwidth are capable of providing larger buffers.
819 The goal for the USB architecture is to ensure that buffering-induced hardware delay is
820 bounded to within a few milliseconds.

821 The USB's bandwidth capacity can be allocated among many different data streams. This
822 allows a wide range of devices to be attached to the USB. Further, different device bit rates,
823 with a wide dynamic range, can be concurrently supported.

824 The USB Specification defines the rules for how each transfer type is allowed access to the
825 bus.

826 **4.8 USB Devices**

827 USB devices are divided into device classes such as hub, human interface, printer, imaging,
828 or mass storage device. The hub device class indicates a specially designated USB device
829 that provides additional USB attachment points (refer to Chapter 11). USB devices are
830 required to carry information for self-identification and generic configuration. They are also
831 required at all times to display behavior consistent with defined USB device states.

832 **4.8.1 Device Characterizations**

833 All USB devices are accessed by a USB address that is assigned when the device is attached
834 and enumerated. Each USB device additionally supports one or more pipes through which
835 the host may communicate with the device. All USB devices must support a specially
836 designated pipe at endpoint zero to which the USB device's USB control pipe will be attached.
837 All USB devices support a common access mechanism for accessing information through this
838 control pipe.

839 Associated with the control pipe at endpoint zero is the information required to completely
840 describe the USB device. This information falls into the following categories:

- 841 • **Standard:** This is information whose definition is common to all USB devices and includes
842 items such as vendor identification, device class, and power management capability.
843 Device, configuration, interface, and endpoint descriptions carry configuration-related
844 information about the device. Detailed information about these descriptors can be found
845 in Chapter 9.
- 846 • **Class:** The definition of this information varies, depending on the device class of the USB
847 device.
- 848 • **USB Vendor:** The vendor of the USB device is free to put any information desired here.
849 The format, however, is not determined by this specification.

850 Additionally, each USB device carries USB control and status information.

851 4.8.2 Device Descriptions

852 Two major divisions of device classes exist: hubs and functions. Only hubs have the ability
853 to provide additional USB attachment points. Functions provide additional capabilities to the
854 host.

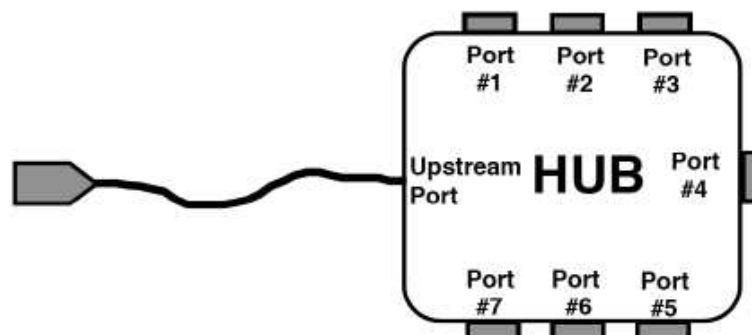
855 4.8.2.1 Hubs

856 Hubs are a key element in the plug-and-play architecture of the USB. Figure 4-3 shows a
857 typical hub. Hubs serve to simplify USB connectivity from the user's perspective and provide
858 robustness at relatively low cost and complexity.

859 Hubs are wiring concentrators and enable the multiple attachment characteristics of the USB.
860 Attachment points are referred to as ports. Each hub converts a single attachment point into
861 multiple attachment points. The architecture supports concatenation of multiple hubs.

862 The upstream port of a hub connects the hub towards the host. Each of the downstream ports
863 of a hub allows connection to another hub or function. Hubs can detect attach and detach at
864 each downstream port and enable the distribution of power to downstream devices. Each
865 downstream port can be individually enabled and attached to either high-, full- or low-speed
866 devices.

867 A USB 2.0 hub consists of three portions: the Hub Controller, the Hub Repeater, and the
868 Transaction Translator. The Hub Repeater is a protocol-controlled switch between the
869 upstream port and downstream ports. It also has hardware support for reset and
870 suspend/resume signaling. The Host Controller provides the communication to/from the host.
871 Hub-specific status and control commands permit the host to configure a hub and to monitor
872 and control its ports. The Transaction Translator provides the mechanisms that support full-
873 /low-speed devices behind the hub, while transmitting all device data between the host and
874 the hub at high-speed.



875
876 **Figure 4-3. A Typical Hub**

877 Figure 4-4 illustrates how hubs provide connectivity in a typical desktop computer
878 environment.

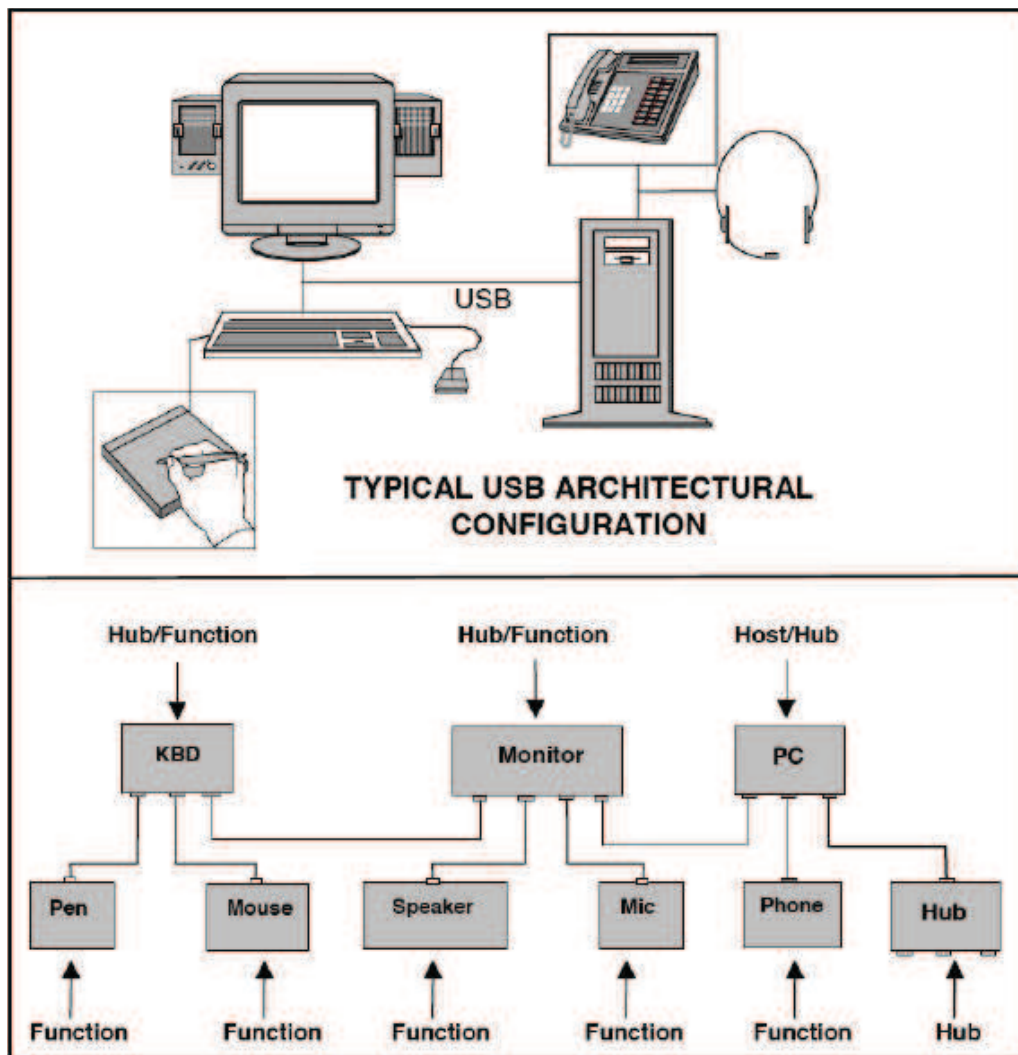


Figure 4-4. Hubs in a Desktop Computer Environment

4.8.2.2 Functions

A function is a USB device that is able to transmit or receive data or control information over the bus. A function is typically implemented as a separate peripheral device with a cable that plugs into a port on a hub. However, a physical package may implement multiple functions and an embedded hub with a single USB cable. This is known as a compound device. A compound device appears to the host as a hub with one or more non-removable USB devices.

Each function contains configuration information that describes its capabilities and resource requirements. Before a function can be used, it must be configured by the host. This configuration includes allocating USB bandwidth and selecting function-specific configuration options.

Examples of functions include the following:

- A human interface device such as a mouse, keyboard, tablet, or game controller
- An imaging device such as a scanner, printer, or camera
- A mass storage device such as a CD-ROM drive, floppy drive, or DVD drive

895 **4.9 USB Host: Hardware and Software**

896 The USB host interacts with USB devices through the Host Controller. The host is
897 responsible for the following:

- 898 • Detecting the attachment and removal of USB devices
- 899 • Managing control flow between the host and USB devices
- 900 • Managing data flow between the host and USB devices
- 901 • Collecting status and activity statistics
- 902 • Providing power to attached USB devices

903 The USB System Software on the host manages interactions between USB devices and host-
904 based device software. There are five areas of interactions between the USB System
905 Software and device software:

- 906 • Device enumeration and configuration
- 907 • Isochronous data transfers
- 908 • Asynchronous data transfers
- 909 • Power management
- 910 • Device and bus management information

911 **4.10 Architectural Extensions**

912 The USB architecture comprehends extensibility at the interface between the Host Controller
913 Driver and USB Driver. Implementations with multiple Host Controllers, and associated Host
914 Controller Drivers, are possible.

915

5 Chapter 5

USB Data Flow Model

This chapter presents information about how data is moved across the USB. The information in this chapter affects all implementers. The information presented is at a level above the signaling and protocol definitions of the system. Consult Chapter 7 and Chapter 8 for more details about their respective parts of the USB system. This chapter provides framework information that is further expanded in Chapters 9 through 11. All implementers should read this chapter so they understand the key concepts of the USB.

5.1 Implementer Viewpoints

The USB provides communication services between a host and attached USB devices. However, the simple view an end user sees of attaching one or more USB devices to a host, as in Figure 5-1, is in fact a little more complicated to implement than is indicated by the figure. Different views of the system are required to explain specific USB requirements from the perspective of different implementers. Several important concepts and features must be supported to provide the end user with the reliable operation demanded from today's personal computers. The USB is presented in a layered fashion to ease explanation and allow implementers of particular USB products to focus on the details related to their product.

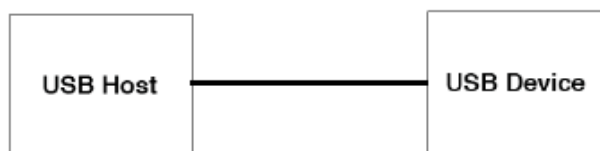


Figure 5-1. Simple USB Host/Device View

Figure 5-2 shows a deeper overview of the USB, identifying the different layers of the system that will be described in more detail in the remainder of the specification. In particular, there are four focus implementation areas:

- USB Physical Device: A piece of hardware on the end of a USB cable that performs some useful end user function.
- Client Software: Software that executes on the host, corresponding to a USB device. This client software is typically supplied with the operating system or provided along with the USB device.
- USB System Software: Software that supports the USB in a particular operating system. The USB System Software is typically supplied with the operating system, independently of particular USB devices or client software.
- USB Host Controller (Host Side Bus Interface): The hardware and software that allows USB devices to be attached to a host.

There are shared rights and responsibilities between the four USB system components. The remainder of this specification describes the details required to support robust, reliable communication flows between a function and its client.

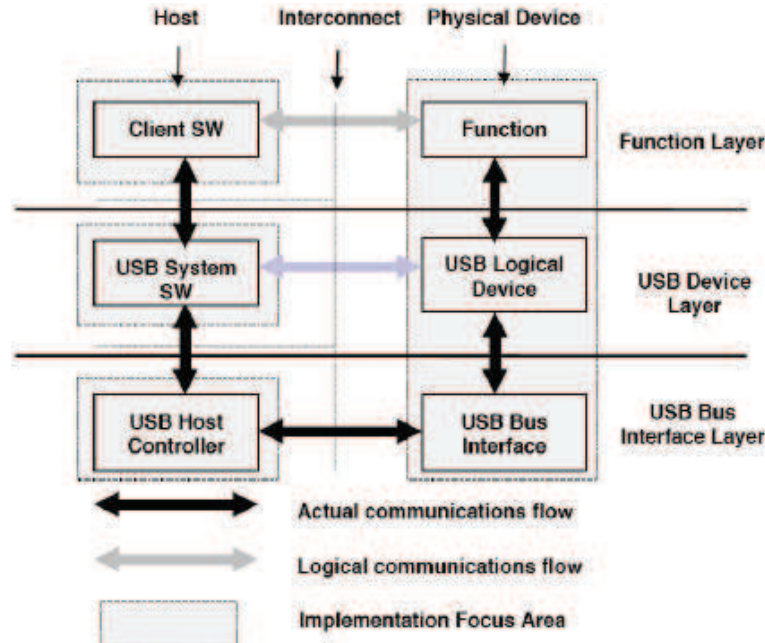


Figure 5-2. USB Implementation Areas

As shown in Figure 5-2, the simple connection of a host to a device requires interaction between a number of layers and entities. The USB Bus Interface layer provides physical/signaling/packet connectivity between the host and a device. The USB Device layer is the view the USB System Software has for performing generic USB operations with a device. The Function layer provides additional capabilities to the host via an appropriate matched client software layer. The USB Device and Function layers each have a view of logical communication within their layer that actually uses the USB Bus Interface layer to accomplish data transfer.

The physical view of USB communication as described in Chapters 6, 7, and 8 is related to the logical communication view presented in Chapters 9 and 10. This chapter describes those key concepts that affect USB implementers and should be read by all before proceeding to the remainder of the specification to find those details most relevant to their product.

To describe and manage USB communication, the following concepts are important:

- **Bus Topology:** Section 5.2 presents the primary physical and logical components of the USB and how they interrelate.
- **Communication Flow Models:** Sections 5.3 through 5.8 describe how communication flows between the host and devices through the USB and defines the four USB transfer types.
- **Bus Access Management:** Section 5.11 describes how bus access is managed within the host to support a broad range of communication flows by USB devices.
- **Special Consideration for Isochronous Transfers:** Section 5.12 presents features of the USB specific to devices requiring isochronous data transfers. Device implementers for non-isochronous devices do not need to read Section 5.12.

5.2 Bus Topology

There are four main parts to USB topology:

- **Host and Devices:** The primary components of a USB system
- **Physical Topology:** How USB elements are connected

- 979 • Logical Topology: The roles and responsibilities of the various USB elements and how the
- 980 USB appears from the perspective of the host and a device
- 981 • Client Software-to-function Relationships: How client software and its related function
- 982 interfaces on a USB device view each other

983 5.2.1 USB Host

984 The host's logical composition is shown in Figure 5-3 and includes the following:

- 985 • USB Host Controller
- 986 • Aggregate USB System Software (USB Driver, Host Controller Driver, and host software)
- 987 • Client

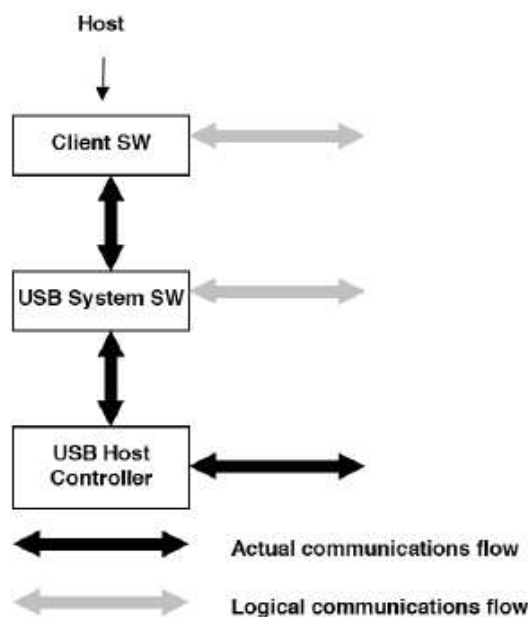


Figure 5-3. Host Composition

990 The USB host occupies a unique position as the coordinating entity for the USB. In addition

991 to its special physical position, the host has specific responsibilities with regard to the USB

992 and its attached devices. The host controls all access to the USB. A USB device gains

993 access to the bus only by being granted access by the host. The host is also responsible for

994 monitoring the topology of the USB.

995 For a complete discussion of the host and its duties, refer to Chapter 10.

996 5.2.2 USB Devices

997 A USB physical device's logical composition is shown in Figure 5-4 and includes the following:

- 998 • USB bus interface
- 999 • USB logical device
- 1000 • Function

1001 USB physical devices provide additional functionality to the host. The types of functionality

1002 provided by USB devices vary widely. However, all USB logical devices present the same

1003 basic interface to the host. This allows the host to manage the USB-relevant aspects of

1004 different USB devices in the same manner.

1005 To assist the host in identifying and configuring USB devices, each device carries and reports

1006 configuration-related information. Some of the information reported is common among all