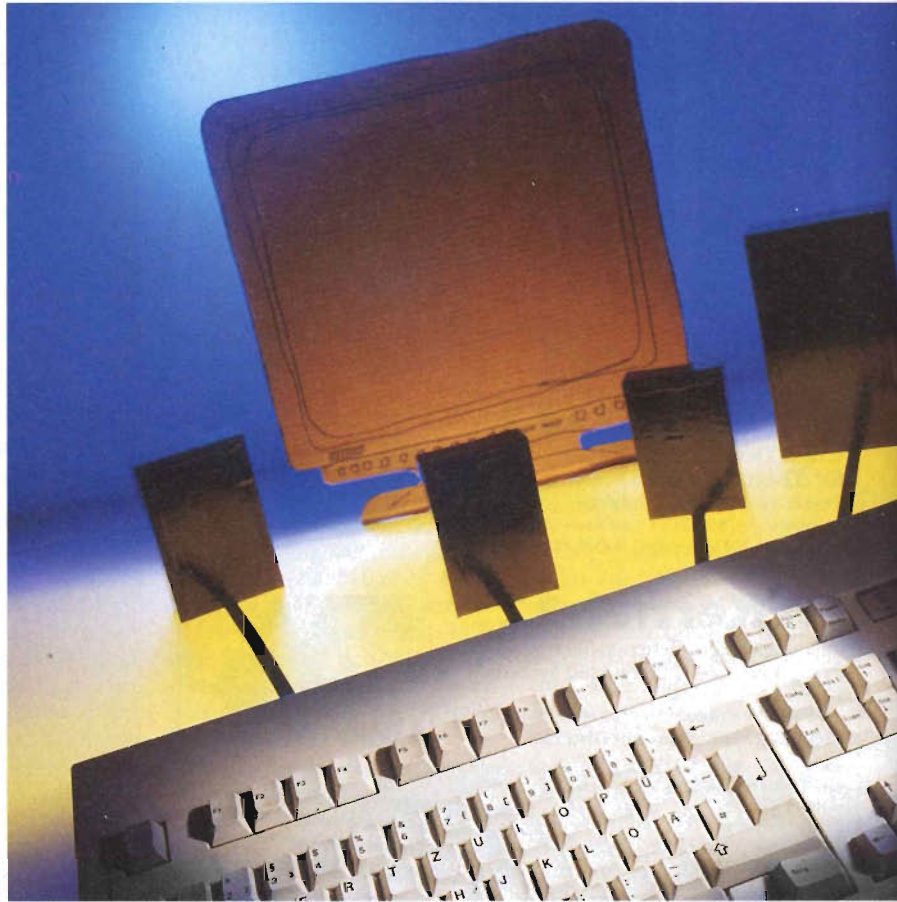


Georg Schnurer

Die Schatten kommen

Erstkontakt mit USB-Devices

Seit gut zwei Jahren geistert ein neuer Peripherie-Bus durch die PC-Welt. Sein Name: Universal Serial Bus. Obwohl sich schon so manches Motherboard mit einem Interface für den Neuling schmückt, blieb das Konzept bislang vage und ließ sich vor allem nicht testen: Es fehlten Adapterkabel, Geräte und nicht zuletzt Treiber. Ist USB also ein 'Useless Serial Bus'?



USB, so versprechen es seine Paten Compaq, DEC, IBM, Intel, Microsoft, NEC und Northern Telecom, erleichtert den Umgang mit dem PC. Eine einheitliche I/O-Schnittstelle, so meinen sie, macht Schluß mit dem Stecker-Wirrwarr und sorgt obendrein dafür, daß auch neue Geräte Anschluß an den PC finden. Besonders angetan hat es den Mannen rund um Intel und Microsoft hier das Telefon, weshalb ja auch Northern Telecom mit im USB-Boot sitzt.

In der Tat ist ein PC heute alles andere als übersichtlich. Das in gut 15 Jahren gewachsene Stecker-Chaos kann einen schon in die Verzweiflung treiben. Neben Tastatur, Maus, Modem, Drucker, Mikrofon, Lautsprecher, Kamera, Scanner will auch neugeschaffene Peripherie mit dem PC verbunden werden. Natürlich nutzt jedes Gerät dazu eine spezielle Schnittstelle mit eigenem Stecker. Zu allem Überfluß sind die vielen Stecker und Buchsen nicht 100prozentig verwechslungssicher: Das beginnt harmlos mit der PS/2-Maus, die ver-

sehtlich im PS/2-Tastaturport steckt, und endet fatal mit dem am Druckerport angeschlossenen Modem oder Scanner.

Mac-Anwender haben es da etwas besser. Apples ADB (Apple Desktop Bus) bietet schon lange eine einheitliche Schnittstelle für Tastatur, Maus und Co., die sich auch neuen Geräten gegenüber aufgeschlossen zeigt. Allerdings ist der ADB extrem langsam, weshalb Drucker, Modem und Scanner doch wieder über separate Anschlüsse verfügen.

Janusköpfig

Der Wunsch, sowohl langsame Geräte wie Tastatur und Maus als auch schnelle Geräte wie Modems oder gar Videokameras über ein und denselben Bus führen zu wollen, sorgt für einige Eigenheiten bei USB. So gibt es einen Low-Speed- (1,5 MBit/s) und einen Medium-Speed-Kanal (12 MBit/s), die über dieselbe Schnittstelle geführt werden. Später soll noch eine High-Speed-Version des USB mit bis zu 500 MBit/s hinzukommen.

Theoretisch wäre diese Unterteilung nicht notwendig gewesen, doch hätten in diesem Fall auch langsame Peripheriegeräte mit demselben Aufwand wie schnelle gebaut werden müssen. Das hätte nach Ansicht der USB-Schöpfer aber die Kosten in die Höhe getrieben und so die Konkurrenzfähigkeit von USB – insbesondere gegenüber den etablierten PC-Schnittstellen – gefährdet. Mit der Aufteilung in verschiedene Geschwindigkeitsbänder hingegen soll es möglich werden, Tastaturen und Mäuse mit USB-Anschluß zu bauen, die nur wenig mehr kosten als konventionelle Geräte.

Unabhängig davon, welcher Geschwindigkeitskategorie ein Gerät angehört, arbeitet es stets mit dem gleichen vierpoligen Stecker. Unterschiede ergeben sich lediglich beim verwendeten Kabel. High-Speed-Geräte nutzen ein geschirmtes und verdrehtes Kabel mit einer Maximallänge von fünf Metern. Low-Speed-Kabel sind ungeschirmt und unverdrillt und dürfen maximal drei Meter lang sein. Damit solch ein USB-Kabel nicht zum Störsender

wird, sind die Signalanstiegszeiten nach unten hin begrenzt (75–300 ns).

Bus-Topologie

Der USB hat eine kombinierte Stern- und Strang-Struktur (siehe Grafik). An seiner Spitze steht der USB-Hostadapter im Rechner. Bei Pentium-PCs mit Intel-Chipsatz befindet sich der USB-Host als sekundäre Funktion innerhalb der PCI-to-ISA-Bridge (PIIX3). Er bietet zwei USB-Anschlüsse und stellt damit bereits einen Hub (Busverteiler) dar.

An die Ausgänge des USB-Hostadapters können wahlweise einzelne Geräte, weitere Hubs oder Kombinationen aus beiden angeschlossen werden. Hubs lassen sich beliebig kaskadieren, wodurch letztlich eine Stern-/Strang-Struktur mit pyramidenförmigem Aufbau entsteht.

Hubs müssen per definitionem stets Low- und Medium-Speed beherrschen. Da Geräteketten (Stränge) nur unter Zuhilfenahme von Hubs in den Geräten selbst entstehen kön-

nen, ist auf diese Art auch immer ein beliebiger Mischbetrieb von schnellen und langsamen USB-Geräten an einem Strang möglich. Es muß nur sichergestellt sein, daß jedes USB-Gerät, das über einen Hub zum Anschluß anderer Geräte verfügt, in Richtung Hostadapter mit einem Kabel für Medium-Speed ausgerüstet ist, auch wenn es selbst nur mit Low-Speed arbeitet (etwa Tastaturen mit integriertem Hub).

Die Verbindung zwischen einem Hub und der darüberliegenden Ebene nennt man Upstream-Port. An den Endpunkten jedes Strangs sitzen die einzelnen USB-Geräte, die die Spezifikation als Node bezeichnet. Insgesamt kann der USB bis zu 127 Geräte ansteuern.

Elektrisch gesehen ist USB, wie der Name schon sagt, ein serieller Bus mit Punkt-zu-Punkt-Verkabelung. Alle Daten werden über ein bidirektionales Leitungspaar geführt. Hinzu kommen noch eine Masse- und eine Versorgungsspannungsleitung (+5 Volt). Um den Bus robuster zu gestalten, hat man sich für differentielle Datenübertragung entschieden (Datenleitung D+ und D-). Externe Störungen wirken so gleichermaßen auf beide Signalleitungen und lassen sich auf der Empfängerseite leicht ausfiltern.

Eine separate Taktleitung gibt es bei USB nicht. Der Bustakt wird vielmehr aus dem Datenstrom selbst generiert. Dazu bedient man sich eines Sync-Signals und der NRZI-Kodierung (Non Return to Zero Invert). Ein einfacher Algorithmus sorgt hier dafür, daß jede logische Null durch einen Signalwechsel symbolisiert wird (siehe Bild). Damit sichergestellt ist, daß genügend '0'-Signale im Datenstrom vorhanden sind, wird vor der NRZI-Kodierung geprüft, ob mehr als sechs Einsen aufeinander folgen. Ist dies der Fall, fügt man eine '0' ein (Bit Stuffing). So hat jedes USB-Device spätestens alle sieben Takte Gelegenheit, sein internes Clock-Signal mit dem Bustakt zu synchronisieren.

Who is who?

Die Unterscheidung von bis zu 127 Geräten geschieht – ähnlich wie bei SCSI – über eine

Geräte-ID. Für deren Vergabe ist der USB-Host zuständig. Beim Einschalten des Systems hat jedes Gerät zunächst die ID 0. Der Host klappert jetzt den USB-Strang Ebene für Ebene ab und teilt jedem Gerät eine ID zu.

Neben dieser ID besitzt jedes USB-Gerät eine fest verdrahtete Hardwarekennung. Diese besteht aus einer Herstellerkennung (Vendor-ID, 2 Byte), einer Device-ID (2 Byte) und einer Versionsnummer (1 Byte). Hinzu kommt noch ein Class-Code (3 Byte: Class, SubClass und Protokoll-Type), anhand dessen ein Gerätetreiber den Gerätetyp erkennen kann.

Bei der Initialisierung informiert das USB-Gerät den Hostadapter darüber, welchen Datentyp es unterstützt und welche Bandbreite es benötigt. Der Host speichert diese Informationen und benutzt sie später, um die zur Verfügung stehende Bandbreite des USB zu verteilen. Dies geschieht über sogenannte Pipes, also Punkt-zu-Punkt-Datenkanäle. Um die Funktionsweise einer Pipe zu erklären, muß man etwas weiter ausholen.

Prinzipiell ist USB ein Polling-Bus. Das heißt, alle Aktivitäten gehen in regelmäßigen Zeitabständen vom Hostadapter aus. Ein USB-Gerät kann von sich aus keinen Transfer initiieren oder etwa einen Interrupt auslösen. Das bedeutet allerdings nicht, daß der Rechner den USB-Controller regelmäßig abfragen muß. Aus Sicht der CPU ist der USB-Hostadapter selbst nämlich ein Interruptfähiges Gerät. Das Polling bezieht sich lediglich auf den USB selbst und wird vom Hostadapter eigenständig erledigt.

Im Prinzip sieht ein Datentransfer auf dem USB etwa so aus. Der Hostadapter fragt das Device zunächst, ob es Daten übertragen will. Das Gerät beantwortet diese Anfrage mit einem Message-Paket, das den Host darüber informiert, ob Daten zu übertragen sind oder nicht. Liegen Daten bereit, so prüft der Host zunächst, ob auf dem Bus genügend Bandbreite zur Übertragung vorhanden ist. Hierzu nutzt es die bei der Geräteinitialisierung ausgelesenen Device-Daten. Wenn noch genügend Bandbreite zur Verfügung steht, wird die Pipe eta-

bliert, und das Device kann seine Daten übertragen. Sobald dies geschehen ist, steht die von der Pipe genutzte Bandbreite wieder dem Bus zur Verfügung.

Reicht die Bandbreite des Busses nicht mehr für eine Datenübertragung aus, so wird der Transferwunsch des Device vom Hostadapter zurückgewiesen. Eine Unterbrechung bereits bestehender Pipes zugunsten einer neuen, wie es sie etwa

beim PCI-Bus gibt, ist bei USB nicht vorgesehen.

Damit keine Daten verlorengehen, muß jedes USB-Gerät einen Puffer haben. Dessen Größe richtet sich nach dem gerätetypischen Datenaufkommen und vor allem nach seiner Datentransferrate. Je höher diese ist, um so größer muß der Puffer sein. Ebenso muß es Zwischenspeicher auf der Host-Seite geben, um die für ein USB-Gerät bestimmten Daten

USB auf einen Blick

Die Idee des Universal Serial Bus bietet viele Vorteile gegenüber den herkömmlichen PC-Schnittstellen. Die wichtigsten sind im folgenden kurz aufgeführt. Ob sich alle genannten Punkte auch in der Praxis bestätigen, wird die Zukunft zeigen.

Einfache Anwendbarkeit

Bei USB gibt es nur einen Steckertyp für alle Geräte, so daß Verwechslungen ausgeschlossen sind. Die Identifikation jedes Geräts geschieht automatisch durch den USB-Hostadapter. Er ist auch für die Grundkonfiguration des USB-Geräts verantwortlich.

Der USB-Basistreiber informiert das Betriebssystem über neu hinzugekommene Geräte und veranlaßt das Laden gerätespezifischer Treiber. In der Regel wird der Anwender damit nicht behelligt.

Auch kennt USB keine besonderen Einstellungen, für die der Anwender zuständig ist. Dieser muß also weder selbst – wie etwa bei SCSI – Geräte-IDs vergeben oder für korrekte Terminierung sorgen, noch muß er sich mit Master-Slave-Jumpers wie bei IDE-Platten herumärgern. Ebenso wenig muß er sich selbst um Bitraten-, Protokoll- oder Handshake-Einstellungen wie etwa bei der seriellen Schnittstelle kümmern.

Viele Anschlüsse

Ein USB-Hostadapter kann bis zu 127 Geräten ansteuern, deren Datenaufkommen zwi-

schen wenigen KBit/s und 12 MBit/s betragen kann. Jedes Gerät kann entweder nur eine oder aber mehrere Funktionen besitzen; eine Tastatur mit integriertem Hub etwa ermöglicht den Anschluß eines weiteren USB-Geräts, dessen Kabel dann mit dem zusätzlichen Anschluß in der Tastatur verbunden wird.

Zeitgenauer Datentransfer

Neben asynchronem ist auch isochroner – also zeitgenauer – Datentransfer möglich, wie ihn beispielsweise Soundwiedergabe benötigt. Der USB kann verschiedene Datenströme quasiparallel übertragen, solange die Gesamtbandbreite des Busses (12 MBit/s beziehungsweise 1,5 MByte/s) nicht überschritten wird.

Flexible Datenpakete

Der USB kann Datenpakete variabler Größe mit (derzeit) zwei unterschiedlichen Datenraten transportieren. Das vereinfacht das Design von USB-Peripherie und senkt deren Kosten.

Robuster Bus

Alle USB-Geräte sind Hotplug-fähig, das heißt, sie dürfen bei laufendem Rechner angeschlossen oder entfernt werden. Die Datenleitungen sind differentiell ausgeführt, was die Störanfälligkeit der Datenübertragung senkt. Der Signalpegel (Spannungshub) ist dadurch aus Sicht des Differenzverstärkers doppelt so hoch wie die Betriebsspannung.

vorzuhalten, wenn der Bus gerade keine Kapazität für deren Übertragung frei hat.

Stream-On

Die meisten Datentransfers bei USB erfolgen über sogenannte 'Stream Pipes'. Sie transportieren in ihrem Datenbereich Informationen ohne festes Format, jedoch in Paketen mit einer maximalen Größe von 1500 Bytes. USB interessiert bei Stream Pipes lediglich der Daten-Header und das Datenende. Die dazwischenliegenden Daten selbst werden unabhängig von ihrem Inhalt übertragen. Über Stream Pipes laufen isochrone Transfers, Bulk-Transfers und Interrupt-Transfers.

Der Begriff 'Interrupt' ist hier allerdings etwas irre-

führend, denn Interrupts im klassischen Sinne gibt es bei USB ja nicht. Die Spezifikation versteht unter einem Interrupt-Transfer vielmehr die unregelmäßige Übertragung – jedoch mit einer garantierten maximalen Service-Periode – kleiner Datenmengen. Als kleine Datenmenge bezeichnet USB maximal 64 Byte bei Medium-Devices und 8 Byte bei Slow-Devices. Wird ein Interrupt-Transfer durch einen Fehler bei der Übertragung gestört, so versucht der USB-Host, die Daten in der nächsten freien Transferperiode erneut zu übertragen. Typischerweise verwenden Input-Devices wie Tastaturen oder Mäuse Interrupt-Transfers.

Bulk-Transfers nutzen dagegen Geräte, die große Datenmengen übertragen wollen. Allerdings dürfen diese Daten

nicht zeitkritisch sein. USB garantiert nämlich nur den Transport der Daten, sobald auf dem Bus Kapazitäten frei sind. Eine bestimmte maximale Service-Zeit oder minimale Bandbreite gibt es nicht. Wie bei Interrupt-Transfers wiederholt der USB-Host die Datenübertragung, wenn ein Busfehler aufgetreten ist.

Zeitgenau

Isochrone, also zeitgenaue Datentransfers laufen ebenfalls über Stream Pipes. USB garantiert für jede isochrone Pipe eine bestimmte Bandbreite und eine maximale Latenzzeit. Darüber hinaus stellt der Bus sicher, daß Daten über die Pipe mit einer konstanten Datenrate übertragen werden, solange die Pipe genutzt wird. Eine automatische Fehlerkorrektur durch Wiederholung wie bei Bulk- und Interrupt-Transfers gibt es allerdings nicht.

Am USB werden mehrere Geräte quasiparallel betrieben. Das erreicht man dadurch, daß die Geräte im Wechsel ihre Datenpakete jeweils mit maximaler Geschwindigkeit übertragen. Während ein Gerät seine Daten überträgt, sammeln die anderen die Daten in ihren Puffern, bis wieder ein versandfertiges Datenpaket zusammengestellt ist.

Isochronen Daten wie Sprache oder Musik muß der USB-Host dabei eine Sonderbehandlung gewähren, indem er für deren Übertragung eine maximale Latenzzeit einhält. Andernfalls gibt es Unterbrechungen, wenn etwa der Puffer eines USB-Lautsprechers leerläuft oder der eines USB-Mikrofons überläuft.

Eine Pipe wird also von ihrer Bandbreite und ihrer minimalen Service-Zeit charakterisiert. Als dritte Kenngröße kommt noch die maximale Datenpaketlänge hinzu, die von der Tiefe des Sende- und Empfangs-Buffers bestimmt wird. Ein Beispiel für zwei parallele isochrone Datentransfers (USB-Stereo-Lautsprecher und USB-Mono-Mikrofon) zeigt die abgebildete Grafik. Sie verdeutlicht auch, wie schnell der paketweise Datentransfer auf den Bus von-statten gehen muß, damit keine Störungen auftreten.

Um zu vermeiden, daß der USB völlig von isochronen

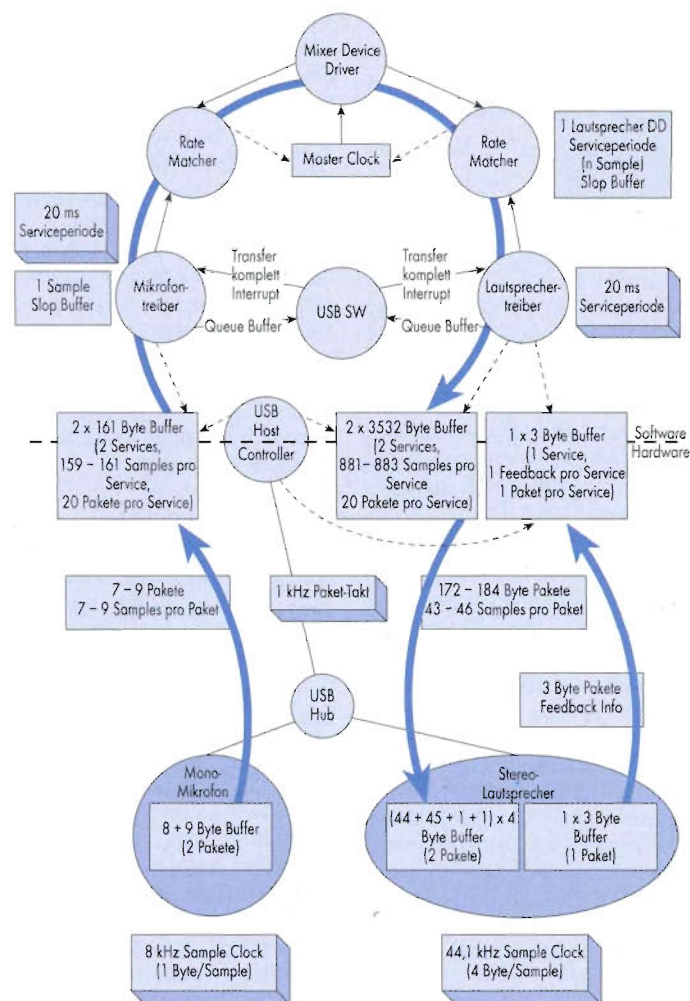
Geräten blockiert wird und so beispielsweise keine Tastatur- oder Mauseingaben mehr möglich sind, reserviert der USB-Host 20 Prozent der zur Verfügung stehenden Bandbreite für asynchrone Geräte und Protokollaufgaben. Damit stehen theoretisch maximal 1,2 MByte/s pro USB-Strang für zeitkritische Geräte zur Verfügung. Je nach Datenart, Paketgröße und Übertragungshäufigkeit verringert sich dieser Idealwert.

So kostet das Bit-Stuffing ebenso Bandbreite wie die zu jedem Datenpaket gehörenden Sync-Impulse und der sonstige Protokoll-Overhead. In der Praxis kann man aber sicher mit einer Bandbreite von etwa einem MByte/s rechnen. Das reicht bequem für Audio- und MPEG-Signale aus, verschleißt dem USB aber den Bereich der Videobearbeitung. Hier dürften deshalb andere Bussysteme wie etwa der FireWire (IEEE 1394) zum Zuge kommen – sofern die USB-Protagonisten nicht schnelligst den High-Speed-USB mit 500 MBit/s nachlegen.

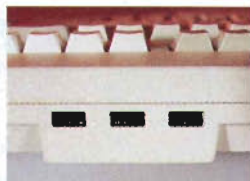
Nachrichtendienst

Neben den Stream Pipes kennt USB noch Message Pipes, die auch jedes USB-Gerät beherrschen muß, allerdings nicht für jede Transferart. Message Pipes haben eine feste Struktur und bestehen aus drei Paketen: dem Request durch den USB-Host, einer Datenübertragung und einem Quittungspaket vom USB-Device. Letzteres ist der entscheidende Unterschied zu Stream Pipes: Zur Pflicht eines USB-Geräts gehört es bei allen Daten, die über Message Pipes verschickt werden, über den Erfolg oder Mißerfolg des Transfers zu berichten. Entsprechende Intelligenz im USB-Gerät vorausgesetzt, könnte es aber auch nach einer Analyse der erhaltenen Daten darin enthaltene logische Fehler selbst erkennen und melden, was der Treibersoftware Gelegenheit gibt, den Transfer zu wiederholen.

Zwischen den drei Paketen einer Message Pipe kann der USB Daten mit anderen Devices austauschen. Verschachtelte Message- oder Stream-Transfers zu einem Device sind allerdings verboten, was den Aufbau von USB-Geräten er-



Zwei parallele isochrone Datenströme bewältigt USB in diesem Beispiel, in dem auch die unterschiedlichen Anforderungen der beiden Geräte deutlich werden. Das Mono-Mikrofon benötigt nur eine geringe Bandbreite und deshalb auch nur recht kleine Buffer. Die USB-Stereo-Lautsprecher sind da deutlich anspruchsvoller.



Für unseren Test standen uns eine USB-Tastatur von Cherry mit integriertem Hub und eine von KeyTronic zur Verfügung.

hebt vereinfacht, da sie so keinen Befehls-Zwischenspeicher nebst entsprechender Logik benötigen.

Message Pipes eignen sich zum Transfer aller Datentypen, können also auch isochrone, Bulk- und Interrupt-Transfers übernehmen. Verbindlich muß jedes Gerät den Umgang mit Message Pipes für Control-Transfers beherrschen, über die die Konfiguration aller USB-Geräte erfolgt.

Treibereien

Wer ein USB-Gerät betreiben will, benötigt dazu neben der Hardware – also USB-Hostadapter und USB-Device – natürlich auch Treiber. Derzeit ist es allerdings auf normalem Weg nicht möglich, entsprechende Software zu ergattern. Wer sich dennoch auf die Jagd begeben will, benötigt zunächst einen betriebssystemspezifischen Basistreiber für den USB-Hostadapter. Unseren Informationen zufolge gibt es einen solchen bislang nur für Windows 95 in der Version 950b. Der Treiber ist im Service-Pack 3 versteckt (\\OSR2\\USB_SUPP) und unterstützt den in den PIIX3 von Intel integrierten USB-Controller und dessen Pendant im VIA-Apollo-VP-Chipsatz.

Mit diesem Treiber allein ist es aber noch nicht getan. Windows 95 kennt dann nämlich nur den USB-Hostadapter und den darauf integrierten USB-Hub. USB-Geräte wie etwa Tastatur oder Maus benötigen noch einen gerätespezifischen Treiber. Diesen fordert Windows 95 an, sobald ein USB-Gerät erstmals erkannt wird. Ob dies beim Hochfahren des Systems geschieht oder später, beim Einstecken des USB-Geräts, ist gleichgültig.

Gerätetreiber sind noch schwieriger zu finden als Basistreiber. Die Zurückhaltung, auf die wir bei der Suche nach USB-Geräten und Treibern bei vielen Firmen stießen, hat ihren Grund sicher auch in dem noch recht wackeligen Gerüst, auf dem USB derzeit softwareseitig steht. Microsoft hatte USB bei der Konzeption von Windows 95 nämlich noch nicht vorgesehen. Entsprechend fehlt dem System eine klare Schnittstelle. Derzeit ist USB also eher an Windows 95 angeflickt (siehe Grafik). Eine saubere Integration ist erst mit Windows 97 respektive Windows NT 5.0 geplant.

Jäger und Sammler

Nach so viel Theorie jetzt zur USB-Praxis. Dazu versuchten wir zunächst, einige USB-Geräte aufzutreiben. Die immer wieder in USB-Ankündigungen erwähnte Digital-Kamera von Kodak konnten wir genauso wenig für einen Test bekommen wie die USB-Maus oder den USB-Scanner von Logitech. Gleiches gilt für den Joystick von Microsoft oder den Philips-Monitor mit integriertem USB-Hub.

Zu letzterem sei angemerkt, daß hier keineswegs das Videosignal per USB transportiert werden soll; die heute übliche Videobandbreite von mindestens 100 MHz (50 MHz Pixeltakt) für jedes der drei Analogsignale ergäbe bei True-Color-Farbtiefe (8 Bit pro Farbe) rund 150 MByte/s. Mit 20 Prozent Aufschlag für Synchronsignale sind das knapp 1,5 GBit/s – weit mehr also, als auch konkurrierende Bussysteme wie FireWire (IEEE 1394) bieten könnten. USB kann hier also nur als Control-Bus dienen und

Dinge wie Farb- oder Geometricabgleich beziehungsweise Power-Management erledigen.

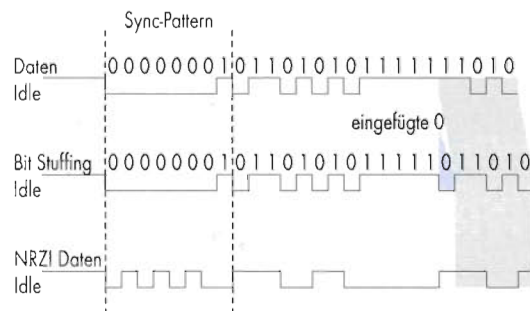
Einzig Cherry erklärte sich sofort bereit, uns eine Vorversion ihrer USB-Tastatur mit integriertem Hub zur Verfügung zu stellen. Später gelang es uns noch, zusätzlich eine USB-Tastatur von KeyTronic in die Hände zu bekommen. Einen geeigneten Tastaturtreiber fanden wir nach langer Suche schließlich in Form einer Alpha-Version von AMI. Von sich aus unterstützte der jedoch zunächst nur die Cherry-Tastatur; erst nach einem kleinen Patch akzeptierte er auch das KeyTronic-Modell.

Was Motherboards mit integriertem USB-Hostadapter betraf, hatten wir ja noch von unserem letzten Board-Test [1] genügend im Labor – dachten wir. Allein, nur eines der 30 Pentium-Boards hatte auch tatsächlich eine USB-Buchse. Bei allen anderen endeten die USB-Ports an einem Pfostenstecker. Wir baten deshalb alle 16 Hersteller, uns zu ihrem Board passende Adapter zu liefern. Letztlich kamen nur acht unserer Aufforderung nach. Der Rest sah sich außerstande, das entsprechende Zubehör zu liefern.

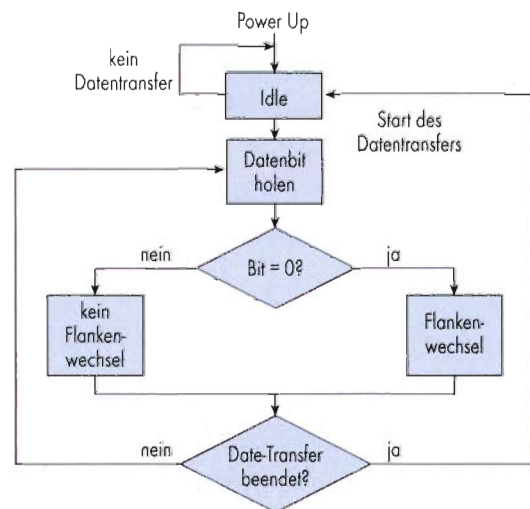
Die gelieferten Adapter lassen sich in zwei Kategorien einteilen: der Typ 1 besteht lediglich aus einem Slotblech mit zwei daran befestigten USB-Buchsen. Typ 2 ist aufwendiger gestaltet. Hier hängt am Slotblech eine kleine Platine, auf der sich neben den beiden USB-Buchsen noch je zwei Stützkondensatoren für die USB-Versorgungsspannung und je ein kleiner Filter (LC-Glied) für die USB-Signale befinden. Am wichtigsten erscheinen mir allerdings die beiden Sicherungen (3 A). Bei einem Kurzschluß auf der USB-Versorgungsleitung ist das Motherboard so wenigstens etwas geschützt. Welcher Hersteller welchen USB-Anschlußtyp lieferte, findet sich in der Tabelle.

Legacy-Support

Die erste – wie wir zunächst dachten – einfachste Hürde, die die Kandidaten zu überwinden hatten, war der sogenannte Legacy-Support für eine USB-Tastatur. Entsprechend der USB-Spezifikation soll eine USB-Tastatur an einem PC bereits beim Booten nutzbar sein. Dazu muß das Rechner-BIOS den USB-Hostadapter aktivie-



Das Takt-signal erhalten USB-Geräte nicht über eine separate Taktleitung. Es wird vielmehr aus dem Datenstrom generiert. Dazu bedient man sich eines Sync-Pakets, Bit-Stuffing und der NRZI-Kodierung.



ren und korrekt initialisieren. Anschließend muß es prüfen, ob eine USB-Tastatur vorhanden ist. Wenn ja, muß es diese ebenfalls initialisieren und eine Emulation des normalen PC-Keyboards (8042) zur Verfügung stellen, damit die USB-Tastatur sich wie eine normale PC-Tastatur verhält.

Diese Grundfunktionalität erwies sich allerdings als schwere Hürde, die letztlich nur eine von 17 Board-/BIOS-Kombinationen sauber überwand. Der Versuch, nur mit einer USB-Tastatur zu booten, scheiterte zunächst mit allen Boards, da der USB-Hostadapter im Setup üblicherweise abgeschaltet war. Wir mußten also erst wieder eine normale PC-Tastatur anschließen und mit dieser den USB-Port im Setup aktivieren. Anschließend waren dennoch nur die beiden Boards von MSI in der Lage, ohne PC-Tastatur zu booten.

Allerdings gab es auch hier einige Einschränkungen. So bietet MSI zwei verschiedene BIOS-Versionen für jedes Board an, eine von AMI und eine von AWARD. Die AWARD-BIOSse weigerten sich aber, mit der KeyTronic-Tastatur zusammenzuarbeiten. Beide Boards (MS-5128 und MS-5136) blieben nach dem Speichertest hängen. Mit der

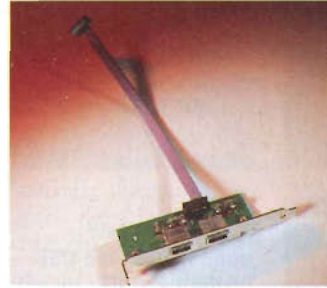
Cherry-Tastatur gab es diese Probleme nicht. Allerdings lieferte die Tastatur beim MS-5136-Board nur sporadisch ein Zeichen. Im anderen Board (MS-5128) arbeitete sie dagegen ohne Störungen.

Beim MS-5136 sah es auch mit AMI-BIOS nicht viel besser aus. Das Board bootete nur, wenn auch eine PC-Tastatur angeschlossen war. Richtig sauber arbeitete lediglich das MS-5128 mit dem AMI-BIOS. Hier liefen beide Tastaturen sauber und ohne Störungen. Es war sogar möglich, die beiden USB- und die PC-Tastatur parallel zu betreiben.

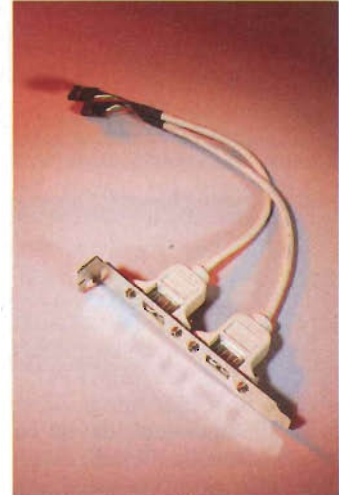
Windows 95

Der Legacy-Support funktionierte auch unter Windows 95 – vorausgesetzt, man verwendet die Version 950b und den Patch aus dem ServicePack 3. Mit den älteren Versionen gibt es Probleme, solange nicht auch zusätzlich eine PC-Tastatur angeschlossen ist. Der Rechner bleibt dann bereits beim Laden von MSDOS 7 hängen. Der Parallelbetrieb von PC- und USB-Tastatur klappt dagegen auch mit den alten Windows-95-Versionen.

Mit den USB-Treibern hatten auch die anderen Boards eine Chance, die Funktion ihrer



Bei den USB-Steckeradaptern gibt es zwei Typen: einfache Slotbleche mit zwei USB-Buchsen und solche mit separater Platine, die auch eine Absicherung der USB-Versorgungsspannung bietet.

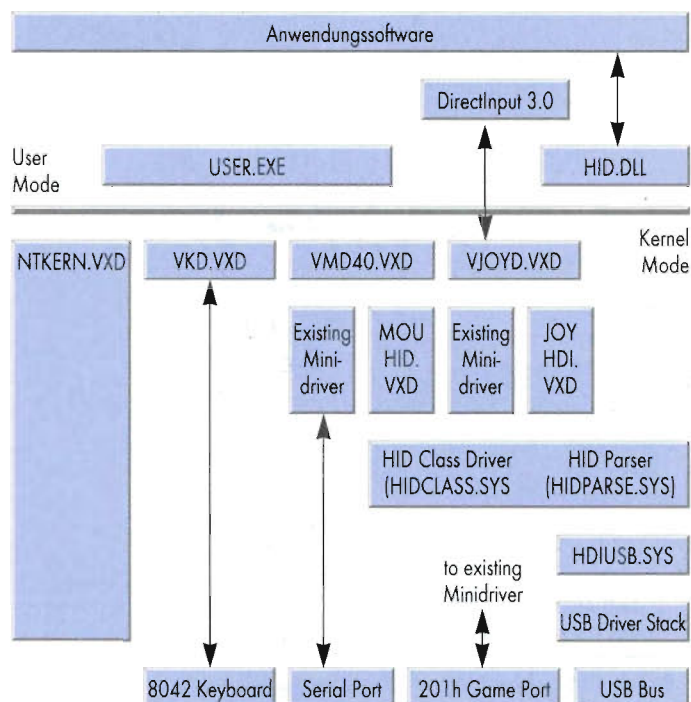


USB-Interfaces unter Beweis zu stellen. Bei den meisten (siehe Tabelle) zeigte sich in etwa das gleiche Verhalten. Beim ersten Start wurde zunächst der per Setup aktivierte USB-Hostadapter erkannt und der Basistreiber geladen. Nach einem Neustart fand Windows 95 auch den in den USB-Controller integrierten Hub und lud den entsprechenden Treiber. Sobald man die KeyTronic-Tastatur einsteckte, erkannte Windows 95 auch diese und fragte nach dem entsprechenden Treiber.

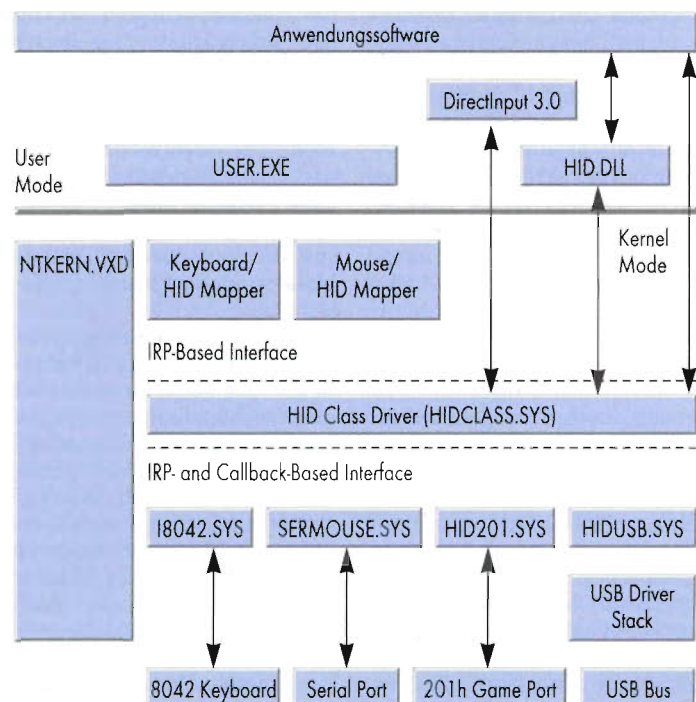
War dieser erst einmal geladen, so konnte man die Tastatur ohne Einschränkungen parallel

zur normalen PC-Tastatur nutzen. Zog man die USB-Tastatur wieder ab, so blieb der Treiber geladen, wurde vom System aber deaktiviert. Beim erneuten Aufstecken der Tastatur war sie dann sofort einsatzbereit.

Ganz so glatt ging es mit der Cherry-Tastatur nicht. Auch hier erkannte Windows 95, wenn die Tastatur angeschlossen wurde und lud den entsprechenden Treiber. Zog man die Tastatur aber ab und steckte sie anschließend wieder ein, blieb das Keyboard tot. Erst wenn man im Gerätemanager manuell ein Aktualisieren der Systemkonfiguration veranlaßt hatte,



Bei den aktuellen Versionen von Windows 95 und NT ist USB nur halbherzig angefflickt.



Sauber integrieren will Microsoft USB erst in Windows 97 respektive Windows NT 5.0.

war die Tastatur wieder da. Hierbei handelt es sich aber wohl um einen Fehler in der AMI-Software, da das Cherry-Modell schließlich im Legacy-Mode, also ohne Windows-95-Treiber, sauber funktionierte.

Wenn unter Windows 95 die USB-Tastatur erst mal lief, konnte man auch die PC-Tastatur abziehen und nur mit der USB-Variante weiterarbeiten, unabhängig davon, ob das BIOS Legacy-Support bot oder nicht. Allerdings sei angemerkt, daß die Standard-PC-Tastatur anders als USB-Geräte kein 'Hot-Plug-Device' ist, also nicht offiziell bei laufendem PC angesteckt oder abgezogen werden sollte. Letztlich hat diese Methode auch wenig Sinn: In allen Varianten, die beim Start eine angesteckte PC-Tastatur erfordern, muß sie früher oder später doch wieder angeschlossen werden.

Der Rest der Welt

Fragt sich natürlich, wie es den anderen Betriebssystemen mit USB-Tastaturen ergeht. Da wir für Windows NT und OS/2 keine passenden Treiber auftreiben konnten, blieb hier nur ein Test des Legacy-Modus. Solange die USB-Tastatur parallel zu einer PC-Tastatur arbeitete, gab es weder unter Windows NT 4.0 noch unter OS/2 Warp 4 Probleme.

War dagegen nur das USB-KeyBoard angeschlossen, weigerten sich, wie bereits erwähnt, die älteren Windows-95-Versionen zu starten. NT 4.0 brauchte

für den Start ohne PC-Tastatur extrem lange, kam aber letztendlich doch auf die Füße. OS/2 Warp 4 schließlich akzeptierte die BIOS-Emulation ohne erkennbare Fehler auf Anhieb.

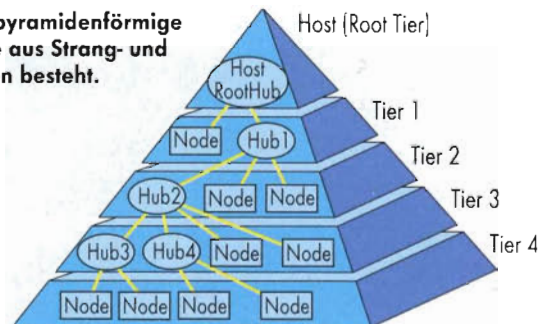
Kurioses

Zum Schluß bleiben noch einige Kuriositäten anzumerken. So lieferte uns WinCo zwar USB-Adapter, doch paßten sie nicht zu den entsprechenden Motherboards. Der Adapter, den uns Flexus/Free für das 586F62 lieferte, paßte dagegen perfekt, allein, im BIOS war keine Möglichkeit vorgesehen, den USB-Port zu aktivieren.

Einen passenden Adapter und ein BIOS, das vermeintlich mit USB umgehen konnte, bescherte uns die Firma Win mit dem MP070. Leider zeigte sich trotz Aktivierung via BIOS kein USB-Hostadapter im Windows-Gerätemanager, und auch CTPCI, das ja direkt die Hardware abfragt, konnte keine entsprechende Funktion entdecken. Das Rätsel löste sich, als wir den verwendeten PIIX3-Chip näher untersuchten. Win hatte die alte A1-Version (erkennbar an der Kennung SU052 auf dem Chip) eingesetzt. In dieser Chip-Version funktioniert der USB-Teil der PCI-to-ISA-Bridge aber noch nicht. Erst die B0-Version (SU093) des PIIX3 enthält einen voll funktionsfähigen USB-Hostadapter.

Warum dagegen das Tyan-Board versagte, bleibt uns ein Rätsel. Das BIOS aktiviert den

USB hat eine pyramidenförmige Topologie, die aus Strang- und Sternelementen besteht.



USB-Port und initialisiert ihn richtig. Windows 95 erkennt den Controller auch und bescheinigt ihm einwandfreie Funktion. Der gelieferte Steckeradapter scheint auch zu funktionieren, dennoch werden beide Tastaturen nicht von Windows 95 erkannt. Beim Einstecken erscheint zwar kurzfristig die Sanduhr, was darauf hindeutet, daß Windows zumindest die Nachricht erhält, daß ein neues USB-Gerät im System steckt. Treiber werden allerdings nicht geladen. Der Versuch, den Tastaturtreiber zwangsweise zu laden, endete schließlich mit einem Blue-Screen.

Fazit

Im Prinzip ist USB eine feine Sache. Der neue Bus scheint robust und vielseitig zu sein und verspricht eine wesentliche Vereinfachung der PC-Verkabelung. Allerdings hat der Neuling derzeit noch einen schweren Stand. Eine USB-Tastatur allein ist meiner Ansicht nach kein Grund dafür, sich auf wackliges Neuland zu begeben. Und

wacklig, das hat unser Test gezeigt, ist derzeit noch alles, was mit USB daherkommt. Heute trifft deshalb wohl noch unsere ketzerische Übersetzung von USB mit 'Useless Serial Bus' zu.

Auf lange Sicht kommt aber wohl niemand an USB vorbei. Das Hardwareangebot dürfte sich Mitte des Jahres erheblich verbreitern und die Kinderkrankheiten sollten bis dahin auch ausgestanden sein. Die Tatsache, daß nahezu jedes moderne PCI-Board bereits mit integriertem USB-Hostadapter angeliefert wird, dürfte ebenfalls zum Markterfolg des neuen PC-Busses beitragen. Ich jedenfalls freue mich bereits auf den ersten USB-Mega-Joystick nebst passendem Adventure-Game. Die Wartezeit überbrücke ich vorerst mit dem Real-Live-Adventure 'PC'. (gs)

Literatur

- [1] Georg Schnurer, Erntezeit, Die aktuellen Pentium-Boards von Asus bis Winco, c't 12/96, S. 296

USB-Support populärer Pentium-Boards

Hersteller	Board-Name	Board-Version	BIOS-Version	USB-Port	USB-Anschluß	Stecker-Typ	Ressourcen IRQ I/O	USB-Boot Sherry Key Tronic	Win95-Legacy Sherry Key Tronic	Win95 Treiber Sherry Key Tronic
SNI	D969	B10-GS2	Phoenix 4.05, 0.99B.969	ja, 2	ja, 2	integriert	IRQ 11 FCE0-FCFF	nein	nein	ja ¹ ja
ASUS	P55T2P4	3.0	AWARD 4.51PG, 1.08	ja, 2	ja, 2	Typ 2	IRQ 11 E400-E41F	nein	nein	ja ¹ ja
	P55TVP4	1.5	AWARD 4.51PG, 1.05	ja, 2	ja, 2	Typ 2	IRQ 11 E400-E41F	nein	nein	ja ¹ ja
	P55T2P4S	1.2	AWARD 4.51PG, 1.09	ja, 2	ja, 2	Typ 2	IRQ 10 E400-E41F	nein	nein	ja ¹ ja
MSI	MS-5128	1.0	AWARD 4.51PG, WH77	ja, 2	ja, 2	Typ 1	IRQ 11 6000-601F	ja	nein	ja ¹ ja
			AMI WinBIOS, AH76	ja, 2	ja, 2	Typ 1	IRQ 11 7F80-7F9F	ja	ja	ja ¹ ja
	MS-5136	1.1	AMI, A175	ja, 2	ja, 2	Typ 1	IRQ 11 7F80-7F9F	nein	nein	nein
Flexus/Free	586F62	1A	AWARD 4.51PG, 09/10/96	ja, 2	ja, 2	Typ 1	-	nein	nein	nein
	586F63	T2	AWARD 4.51PG, 08/28/96	ja, 2	ja, 2	Typ 1	IRQ 10 0120-013F	nein	nein	ja ¹ ja
Tyan	S1563D	PCB-01	AWARD 4.51PG, 10/12/96	ja, 2	ja, 2	Typ 1	IRQ 11 6000-601F	nein	nein	nein
Soyo	Sy-5VA2	Ver. E	AWARD 4.51PG, VA-B1	ja, 2	ja, 2	Typ 1	IRQ 11 6000-601F	nein	nein	ja ¹ ja
	Sy-5VX2	Ver. E	AWARD 4.51PG, VX-B	ja, 2	ja, 2	Typ 1	IRQ 11 6000-601F	nein	nein	ja ¹ ja
Win	MP070	850613	AWARD 4.51PG, 10/01/96	ja, 2	ja, 2	Typ 2	-	nein	nein	nein
	MP071	850815	AWARD 4.51PG, 09/25/96	ja, 2	ja, 2	Typ 2	IRQ 11 6000-601F	nein	nein	ja ¹ ja
WinCo	WP-55TH	0.3	AWARD 4.51PG, 07/18/96	ja, 2	(ja, 2) ¹	-	IRQ 11 6000-601F	nein	nein	nein
	WP-55TV	0.5	AWARD 4.51PG, 09/23/96	ja, 2	(ja, 2) ¹	-	IRQ 11 6000-601F	nein	nein	nein

¹ siehe Text