

Der Serielle Bus des VC 20 und des Commodore 64

Über den seriellen Bus werden alle wichtigen Peripheriegeräte, zum Beispiel der Drucker und das Floppy-Laufwerk, an den VC 20 und den Commodore 64 angeschlossen. Im folgenden soll die Arbeitsweise dieses Busses anhand des Betriebssystems des VC 20 unter Berücksichtigung der Eigenheiten beim Commodore 64 vorgestellt werden.

Commodore hat schon bei seinen früheren Modellen PET, CBM 3032 und CBM 8032 eine etwas modifizierte Form des IEEE-488 oder IEC-Bus verwendet. Dieser Bus ist in der Literatur mittlerweile auch für die CBMs sehr gut dokumentiert (siehe Literaturverzeichnis [1] bis [5]). Für den neuen Bus, der beim VC20 und Commodore 64 eingesetzt wird, trifft das leider nicht zu, da er nicht genormt ist und Commodore detaillierte Informationen über ihn nicht oder noch nicht veröffentlicht hat. Der einzige mir bekannte Artikel [8] ist in den USA erschienen und stammt von »Commodore-Guru« Jim Butterfield.

Worin unterscheidet sich nun dieser neue Bus von seinem Vorgänger?

1. Er hat weniger Leitungen
2. Die Daten werden seriell übertragen
3. Er ist langsamer als der IEEE-488-Bus.

Der serielle Commodore-Bus ist physikalisch ein bidirektionaler Bus, der aus sechs Signalleitungen (siehe Tabelle 1) besteht. Der parallele CBM-Bus ist zirka fünfmal schneller. Die Verwandtschaft mit seinem Vorgänger kann er aber nicht verleugnen, denn die Art, wie angeschlossene Geräte angesprochen werden entspricht dem IEEE-Bus. Es gibt zwei Betriebsarten: Den Kommando- und den Datenmodus.

Das gesamte Geschehen auf dem Bus wird von dem »Controller«, unserem VC 20 oder dem C 64 überwacht und gesteuert (Bild 1). Es darf an dem Bus nur ein Controller angeschlossen sein. Alle Geräte dürfen je nach ihren Möglichkeiten als »Talker« Daten auf den Bus geben und als »Listener« Daten vom Bus lesen.

Ein Drucker wird nur Listenerfunktionen, ein Floppylaufwerk Listener- und Talkerfunktionen haben. Selbstverständlich besitzt unser Computer beide Funktionen.

Damit die Geräte einzeln vom Computer angesprochen werden können, besitzen sie eine Geräteadresse, die Primäradresse. Sie ist für die Commodoregeräte standardmäßig für den Drucker auf 4 und für das Floppylaufwerk auf 8 gesetzt. Um besondere Befehle an ein Gerät übermitteln zu können, zum Beispiel die Anwahl einer bestimmten Druckart bei einem Drucker, besitzt es häufig noch eine Sekundäradresse. Diese Adressen sind für die Geräte unterschiedlich. Ihre Bedeutung ist dem jeweiligen Handbuch zu entnehmen.

Achtung, ich sende

Wie erfährt nun ein angeschlossenes Gerät, daß es gemeint ist? Nun, dafür gibt es die Leitung »ATN«. Hat der Controller diese Leitung auf »Wahr« (Erklärung siehe weiter unten) gesetzt, unterbricht jedes Gerät seine Tätigkeit, denn es weiß: Jetzt kommt ein Befehl. Die Information, die dann vom Controller auf den Bus gelegt wird, wird von den Geräten gelesen und als Primäradresse interpretiert. Alle Geräte bestätigen den Empfang. Die nicht angesprochenen Geräte setzen, sobald die ATN-Leitung wieder auf »Falsch« gesetzt ist, ihre unterbrochene Tätigkeit fort und kümmern sich nicht mehr um das weitere Geschehen auf dem Bus.

Der Computer teilt dem adressierten Gerät mit, ob es als Talker oder als Listener agieren soll. Über

die Sekundäradresse werden vielleicht noch weitere Befehle übermittelt. Dann wird die »ATN«-Leitung vom Computer auf »Falsch« gesetzt. Jetzt kann zwischen Computer und adressiertem Gerät der Datenaustausch stattfinden.

Nachdem das letzte Datum gesendet und empfangen wurde, zieht der Computer die »ATN«-Leitung wieder auf »Wahr«; alle angeschlossenen Geräte reagieren wie oben beschrieben und holen sich wieder die Primäradresse. Der Computer sendet dann zum Beispiel einen »Unlisten« — oder einen »Untalk«-Befehl an das angesprochene Gerät.

Ein Problem gibt es aber noch: Wie weiß der Talker, daß der Listener die Daten auch richtig übernommen hat? Beim IEEE-Bus wird das mit einer Rückmeldung über spezielle Leitungen — den Handshakeleitungen — im Quittungsbetrieb realisiert. Jedes Gerät hat die Möglichkeit, über die jeweilige Leitung den anderen Geräten folgende Informationen zu übermitteln:

DAV die Daten auf dem Datenbus sind gültig (nur Talker)

NDAC die Daten auf dem Datenbus habe ich gelesen (nur Listener)

NRFD ich bin für weitere Daten bereit (nur Listener)

Da aber beim seriellen Bus insgesamt nur sechs Leitungen (siehe Tabelle 1) zur Verfügung stehen, ist der 3-Leitung-Handshake-Betrieb hier nicht durchführbar.

Doch bevor wir in den Ablauf des »Handshake-Betriebes« beim seriellen Bus einsteigen, etwas über die »Logik-Pegel« auf dem Bus.

Der Bus wird in negativer Logik betrieben, das heißt ein Low (zirka 0 Volt) wird als »Wahr«, ein High (zirka 5 Volt) als »Falsch« betrachtet.

Es ergibt sich damit folgende Zuordnung zwischen der Spannung auf der Leitung und ihrem logischen Wert:

High = H-Pegel = zirka +5 V = logisch 0 = »Falsch«
 Low = L-Pegel = zirka 0 V = logisch 1 = »Wahr«

Low und High

Warum das? Sonst ist es doch in der Digital-Technik genau anders herum! Der Grund liegt darin, daß alle Geräte an den Leitungen angeschlossen sind. Die Ausgänge zum Bus hin sind über »Open-Kollektor-Treiber« realisiert. Die Eingänge sind in normaler Transistor-Transistor-Logik (TTL) ausgeführt. Die Beschaltung einer Leitung des seriellen Busses beim VC 20 ist in Bild 2 am Beispiel der Data-Leitung dargestellt.

Signal	Stecker-pin nr.	Bedeutung
IFC	6	Interface Clear Leitung um die angeschlossenen Geräte in einen definierten Anfangszustand zu versetzen (RESET)
DATA	5	Leitung fuer Daten (IN/OUT)
CLK	4	CLOCK Leitung um das Handshakeprotokoll der Dateneübertragung (Taktleitung) zu kontrollieren (IN/OUT)
ATN	3	ATTENTION Leitung um mitzuteilen ob Daten oder Kommandos uebertragen werden
GND	2	GROUND Digitale Masse
SRO	1	SERVICE REQUEST Leitung um dem 'Controller' mitzuteilen, dass Daten bereit stehen (Diese Leitung wird vom Betriebssystem nicht bedient!)

Tabelle 1. Signale und Steckerbelegung des seriellen Busses

Durch diese Schaltungsauslegung ist der aktive Zustand des Busses der L-Pegel, das heißt wenn kein Gerät die Leitung auf 0 Volt zieht, wird die Spannung auf der Busleitung beinahe 5 Volt haben, also High (= Falsch) sein. Ein Bus, an dem keine Peripheriegerate angeschlossen sind, wird also immer im Zustand »Falsch« verharren — der Buscontroller wird dann beim Versuch, ein Gerät anzusprechen, immer die Meldung »Falsch« erhalten und daher einen nicht ordnungsgemäßen Zustand erkennen können.

Noch einen Vorteil hat diese Schaltung. Die Übertragung von Daten darf nicht schneller erfolgen als das langsamste Gerät braucht, um die Daten lesen zu können. Ein kleines Beispiel:

Ein Talker möchte an zwei Listenern Daten senden. Listener 1 ist bereit neue Daten zu übernehmen, er hat also seinen für diese Meldung gedachten Ausgang auf H-Pegel (= »Falsch«) gelegt. Der zweite Listener ist noch nicht soweit, er hält seinen Ausgang noch auf »Wahr«. Damit ist

die Leitung, auf die beide Ausgänge arbeiten, ebenfalls »Wahr« — der Talker weiß, daß er noch warten muß. Er legt seinen Ausgang auf »Falsch«. Die Leitung wird, nachdem kein weiteres Gerät sie auf »Wahr« hält, ebenfalls »Falsch« — der Talker weiß, jetzt kann er eine neue Information auf die Datenleitung legen.

Diese Zusammenhänge wollen wir in zwei Merksätzen zusammenfassen:

Wahr wird eine Leitung, wenn mindestens eines der angeschlossenen Geräte »Wahr« sendet.

Falsch wird eine Leitung nur dann, wenn alle Geräte auf dieser Leitung ein »Falsch« senden.

Damit sind die Logikzustände auf dem seriellen Bus beschrieben. Diese Definition ist überaus wichtig, und man muß sie sich ständig bei der Betrachtung des Geschehens auf dem Bus vor Augen halten. Nach diesem kleinen Exkurs in die Welt der Lows und Highs wollen wir jetzt betrachten, wie Daten auf dem seriellen Bus übertragen werden.

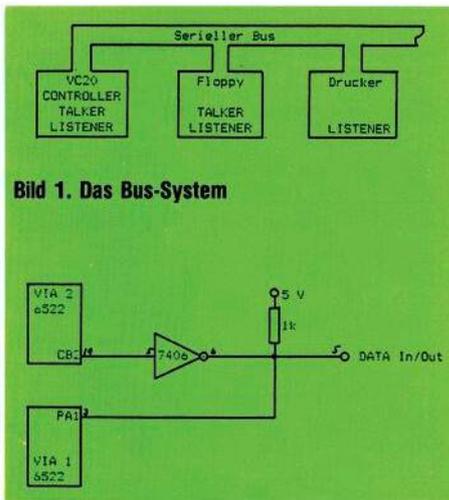
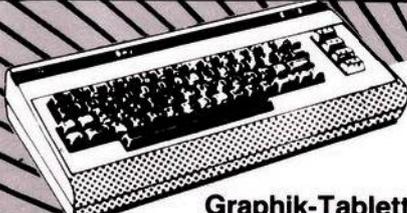


Bild 1. Das Bus-System

Bild 2. Die Beschaltung der Busleitungen (die DATA-Leitung beim VC 20)

Für Commodore VC-20/64



Graphik-Tablett
Zeichnen u. Schreiben Sie in den Rechner! Keine lästige Poke-Programmierung mehr! Mit beiliegender Steuersoftware (Disk) geht das Zeichnen sofort los. Zoom (Ausschnittvergrößerung), Bildspiegelung, automatische Einfärbung uvm. inclusive!

269,-

Grandmaster (20/64) 79,-
Superstarkes Schach!

Speichervollausbau für VC-20

32/27 KByte-Modul
Ersetzt 3 + 8 + 16 KByte oder 8 + 8 + 16 KB kompakt in einem Modul! Voll schaltbar!

179,-

80-Zeichenkarte für C 64 249,-
Gestochen scharfes Profibild!

40/80-Zeichenkarte (20) 229,-

Monitor 12", 15 Mhz 295,-

Eprommer VII (20/64) 179,-
programmiert die EPROMS 2508, 2516, 2716, 2532, 2732. Wird betriebsbereit inklusive Steuersoftware geliefert!

Eprommer VIII (20/64) 249,-
wie oben, jedoch auch für 2764, 27128 geeignet.

Forth-Modul (20/64) 115,-

Centronics Intf. (20/64) 198,-
schließt centr. komp. Drucker an VC's



Sparen Sie den Spezialrecorder

Recorderinterface 49,-
Schließt Ihren Recorder an VC-20 oder C-64. Inclusive Motorsteuerung!

Viele weitere Angebote im **VC-Info 2/84** gegen DM 1,- Porto in Briefmarken.

Klaus Jeschke
Hard-, Software
Im Birkenfeld 3e
6233 Kelkheim
☎ (06198) 7523

Ausgabe 5/Mai 1984

STAR 29

Die Datenübertragung

Die für die Datenübertragung wichtigen Leitungen sind die DATA- und die CLK-Leitung. Die Übertragung eines Bytes (= ein Datenwort aus acht Bit) erfolgt bitweise, beginnend mit dem niederwertigsten Bit, das auf die DATA-Leitung gelegt wird. Die Übertragung eines Bytes erfolgt im Handshake-Betrieb, das heißt mit Rückmeldung vom Listener, daß das Byte empfangen wurde und das nächste gesendet werden kann. Verfolgen wir einmal den Ablauf der Übertragung eines Bytes anhand des Zeitablaufdiagrammes Bild 3 und des Ablaufdiagrammes Bild 4.

Vorbereitung zur Übertragung eines Bytes

CLK- und DATA-Leitung werden beide auf »Wahr« gehalten, wobei der Talker die CLK- und der Listener beziehungsweise alle Listener die DATA-Leitung auf 0 Volt (= »Wahr«) ziehen. Mit diesen Signalen wird jeweils ein »Hier bin ich« signalisiert.

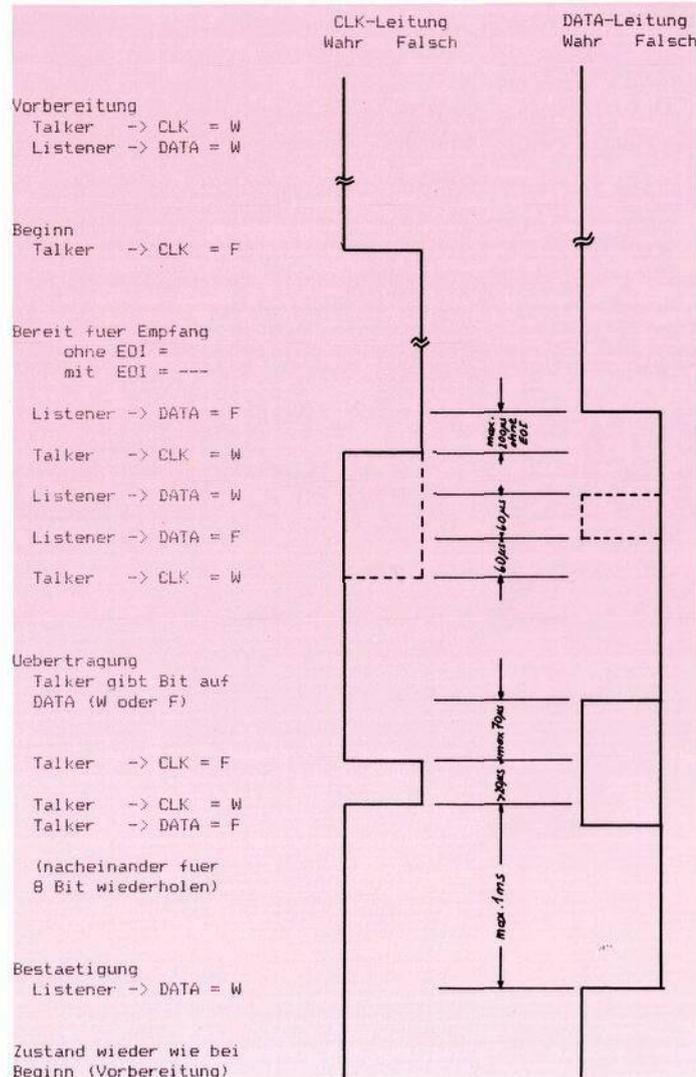


Bild 3. Zeitablaufdiagramm »Übertragung eines Bytes« (nicht maßstabgetreu)

Beginn der Übertragung

Wenn der Talker ein Byte auf dem Bus ausgeben will, läßt er die CLK-Leitung auf »Falsch« gehen. Dieses Signal entspricht einem »Ich bin bereit, ein Zeichen auszugeben«. Der

oder die Listener (in Zukunft werde ich nur mehr von einem Listener sprechen) müssen dieses Signal erkennen und beantworten. Da der Listener zu diesem Zeitpunkt vielleicht noch beschäftigt ist, muß die

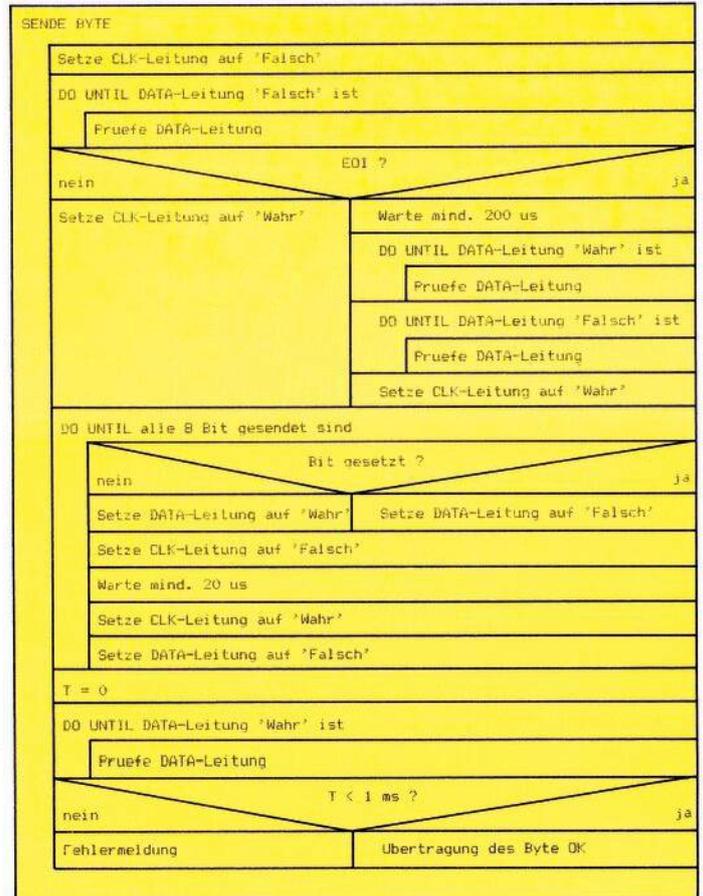


Bild 4. Ablaufschema »Sende ein Byte«

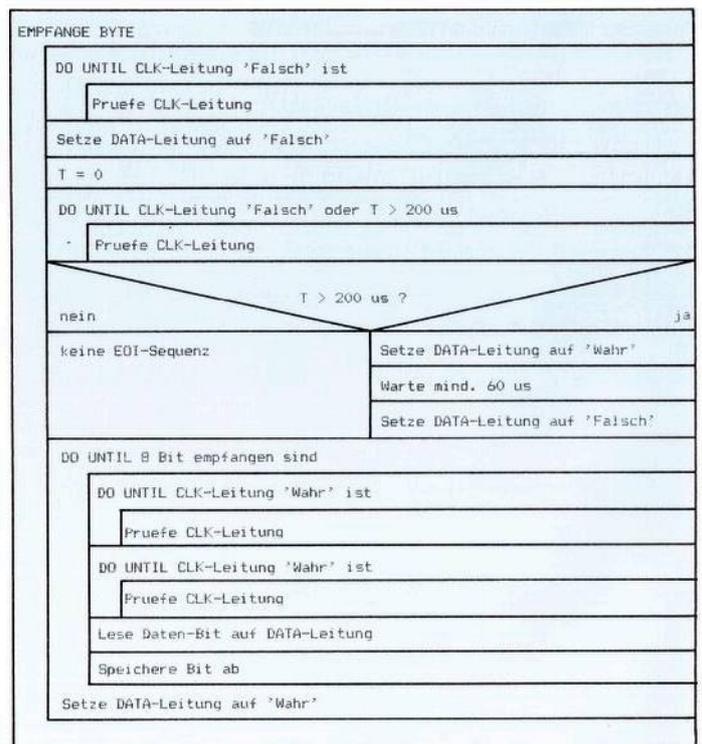


Bild 5. Ablaufschema »Empfange ein Byte«

Quittierung dieses Signals nicht sofort erfolgen, da hier vom Talker eine Zeitüberschreitung nicht überwacht wird.

Bereit für den Datenempfang

Der Listener ist bereit für das nächste Byte; er gibt die DATA-Leitung frei und sagt damit »Ich bin bereit, Daten zu empfangen«. Wenn alle Listener das getan haben, ist die DATA-Leitung jetzt »Falsch«. Der Talker erkennt das und weiß, jetzt sind alle für neue Daten bereit. Nun gibt es zwei Möglichkeiten, was als nächstes passieren kann:

a. Normales Senden.

Der Talker zieht die CLK-Leitung innerhalb von 200 Mikrosekunden — üblicherweise innerhalb von 60 Mikrosekunden — auf »Wahr«, also auf 0 Volt und bestätigt damit die Meldung. Dann beginnt er das Byte bitweise zu senden.

b. EOI-Meldung (= »Dieses Byte ist das letzte«).

Wenn der Talker die Meldung nicht innerhalb von 200 Mikrosekunden bestätigt, weiß der Listener, daß der Talker ihm mitteilen will, daß das Byte, das jetzt gesendet werden soll, das letzte ist. Es kann das letzte in einem sequentiellen File oder in einem Satz eines relativen

Files sein. Wenn der Listener also merkt, daß der Talker das Signal EOI sendet, muß er für mindestens 60 Mikrosekunden die DATA-Leitung auf »Wahr« ziehen und dann wieder auf »Falsch« setzen. Damit hat er dem Talker den Empfang des EOI-Signals bestätigt. Der Talker zieht nun die CLK-Leitung innerhalb von 60 Mikrosekunden auf »Wahr« und beginnt dann das Byte bitweise zu senden.

Die CLK-Leitung ist jetzt, auch wenn die EOI-Sequenz durchgeführt wurde, »Wahr«.

Übertragung des Bytes

Es werden jetzt alle 8 Bits eines Bytes sequentiell, beginnend mit dem niederwertigsten Bit ohne »Handshake« auf die DATA-Leitung gelegt. Beide Leitungen, CLK und DATA werden jetzt allein vom Talker kontrolliert. Für jedes Bit, gesetzt oder nicht, wird die DATA-Leitung auf »Wahr« oder »Falsch« gesetzt. Wenn das Bit auf der DATA-Leitung stabil steht, wird die CLK-Leitung auf »Falsch« gesetzt um mitzuteilen, daß das Bit auf der DATA-Leitung gültig ist. Dieser Vorgang dauert maximal 70 Mikrosekunden. Der Listener hat jetzt wenigstens 20 Mikrosekunden Zeit das Bit zu lesen, da

der Talker während dieser Zeit beide Leitungen in diesem stabilen Zustand hält.

Achtung: Für den Commodore 64 muß diese Zeit auf mindestens 60 Mikrosekunden verlängert werden, da ein Interrupt durch den Videocontroller VIC 6567 42 Mikrosekunden dauert und Daten deshalb in dieser Zeit nicht empfangen werden können.

Wie schon oben gesagt, spielt der Listener hier eine komplett passive Rolle. Er wartet darauf, daß die CLK-Leitung auf »Falsch« geht, liest das Bit, speichert es und bereitet sich auf das nächste Bit vor, wenn die CLK-Leitung wieder »Wahr« wird. Nach Ablauf der »Lesezeit« zieht der Talker die CLK-Leitung wieder auf »Wahr«, setzt die DATA-Leitung auf »Falsch« und bereitet sich für das nächste Bit vor.

Empfangsbestätigung

Nachdem das 8. Bit gesendet worden ist, muß der Listener den Empfang des Bytes bestätigen. Da in diesem Moment die CLK-Leitung »Wahr« und die DATA-Leitung »Falsch« ist, zieht der Listener die Data-Leitung auf »Wahr«. Der Talker muß demnach nach dem letzten Bit die Data-Leitung überwachen.

DATA KASSETTENRECORDER FÜR COMMODORE

Ausstattung:

- Auto Stop
- Bandzählwerk
- Kontroll-LED für Aufnahme
- Datenlade-Überwachung
- Automatische Aussteuerung bei Aufnahme und Wiedergabe
- Start-Stop Steuerung über Computer
- Stromversorgung über Computer
- Anschlußfertig für alle Commodore Computer
- Spezielle Beschichtung der Tonköpfe sichert 100%ige Datenaufnahme und Wiedergabe über mehrere 1000 Betriebsstunden.



Weitere Modelle lieferbar für:

- Atari
- Sinclair
- Apple
- u.v.a.

Postversand ohne Mehrkosten bei Einsendung eines Verrechnungsschecks.

 * demnächst lieferbar: *
 * **Floppy Disk** *
 * für Commodore *

Preis inkl. MwSt. DM 109,—

Händleranfragen erwünscht!

ncs Nettetal
 Computersysteme GmbH

Klemensstraße 7
 4054 Nettetal 2 (Kaldenkirchen),
 Tel. 02157/1616, Teletex/Telex 215732

Alleinvertrieb für:

- Deutschland
- Niederland
- Belgien

Innerhalb von einer Millisekunde muß sie auf »Wahr« liegen, andernfalls gibt der Talker eventuell eine Fehlermeldung aus, da die Übertragung nicht ordnungsgemäß abgelaufen ist.

Damit sind wir am Ende der Übertragung eines Bytes. Der Talker hält die CLK-, der Listener die DATA-Leitung auf »Wahr«. Wir könnten jetzt beim Punkt »Beginn der Übertragung« das nächste Byte senden, wenn kein EOI aufgetreten ist. Ist ein EOI bei der letzten Übertragung gesendet und empfangen worden, ist die Übertragung beendet, beide Leitungen werden freigegeben und sind damit »Falsch«.

Wir wissen mittlerweile wie ein Byte übertragen wird, doch wie werden zum Beispiel die Primäradresse oder Befehle übertragen? Wie schon weiter oben gesagt, gibt es eine besondere Leitung — ATN — mit der der Controller allen angeschlossenen (und natürlich auch eingeschalteten) Geräten mitteilt »Paßt alle auf! Ich will euch was sagen«. Der Controller ist normalerweise der einzige, der diese Leitung auf »Wahr« ziehen darf. Während er also die ATN-Leitung auf »Wahr« hält, sendet er auf der DATA-Leitung Daten. Die Übertragung dieser Daten erfolgt genauso mit Handshake wie oben beschrieben — nur sind es diesmal eben keine Daten sondern Befehle. Darunter fallen die Busbefehle »Talk«, »Listen«, »Untalk« und »Unlisten«. Auch die Geräteadresse, die Sekundäradresse und die Sekundärbefehle »OPEN« und »CLOSE« werden, während die ATN-Leitung »WAHR« ist, gesendet. Der Empfang dieser Befehle wird von allen angeschlossenen Geräten bestätigt, aber nur die tatsächlich angesprochenen führen den übermittelten Befehl aus. Noch einmal: Wird ATN auf »Wahr« gezogen, unterbrechen alle angeschlossenen Geräte ihre derzeitigen Aktionen. Der Controller zieht die CLK-Leitung auf »Wahr«, da er anschließend Daten senden will.

Die Geräte setzen ihre Ausgänge zur CLK-Leitung auf »Falsch« (sie bleibt aber auf »Wahr«, da der Computer sie auf »Wahr« hält). Genau umgekehrt läuft es bei der DATA-Leitung ab — der Computer läßt sie auf »Falsch« gehen, die Geräte setzen sie auf »Wahr«. Die DATA-Leitung wird vom Computer überwacht und wenn die angeschlossenen Geräte die DATA-Leitung nicht innerhalb von einer Millisekunde auf »Wahr« gezogen haben, wird die

Fehlermeldung »DEVICE NOT PRESENT« ausgegeben. Die Übertragung der Daten erfolgt dann auf die schon bekannte Art und Weise.

Ein kleines Beispiel soll zum besseren Verständnis dienen, wie diese Übertragung abläuft. Wir wollen auf dem Gerät 8 — dem Floppylaufwerk — ein Schreibfile eröffnen. Die Basic-Befehlssequenz lautet dann: OPEN1,8,2, "NAME,S,W"

Der Computer wird die ATN-Leitung auf »Wahr« ziehen, dann die Folge »Gerät 8 Listen, Sekundäradresse 2, Sekundärbefehl OPEN« an das Laufwerk senden. Wenn er dann die ATN-Leitung wieder auf »Falsch« setzt, wartet er als Talker mit der CLK-Leitung auf »Wahr«. Das Laufwerk hält als Listener die DATA-Leitung auf wahr. Hiermit ist der Zustand »Vorbereitung zur Übertragung eines Bytes« (siehe oben) hergestellt. So muß es ja auch sein, da der Computer noch weitere Informationen senden will. Er wird als nächstes dann byteweise den Namen, Komma, »S«, Komma und »W« senden. Das »W« wird, da es das letzte Byte ist, mit der »EOI«-Sequenz übermittelt. Danach wird der Computer mit der ATN-Leitung wieder auf »Wahr« ein »Gerät 8 Unlisten«-Befehle senden. Damit ist das File als Schreibfile eröffnet.

Computer als Zuhörer

Irgendwann will dann der Computer Daten auf dieses File schreiben. Er sendet also wieder mit der ATN-Leitung auf »Wahr« ein »Gerät 8 Listen« gefolgt von »Sekundäradresse 2«. Er setzt die ATN-Leitung auf »Falsch« und beginnt dann, wie oben beschrieben, die Daten an das Laufwerk zu senden. Nur das Floppylaufwerk wird diese Daten empfangen und sie in das File mit dem Namen »NAME« schreiben. Das letzte Byte, das der Computer sendet, wird wieder mit »EOI« gesendet. Mit der ATN-Leitung wieder auf »Wahr« gesetzt, sendet er einen »Gerät8Unlisten«-Befehl. Diese Abfolge kann im Laufe einer Programmabarbeitung öfters durchlaufen werden. Zu irgendeinem Zeitpunkt soll dieses Schreibfile wieder mit dem Basic-Befehl CLOSE 1 geschlossen werden. Der Computer wird also, natürlich mit der ATN-Leitung auf »Wahr«, die Folge »Gerät 8 Listen, Sekundäradresse 2, Sekundärbefehl CLOSE« an das Laufwerk senden. Die 1 bei CLOSE dient im Computer nur als Zeiger

auf die tatsächlichen OPEN- und CLOSE-Parameter. Über diese 1 findet der Computer dann die Sekundäradresse 2.

Bis jetzt haben wir den Computer eigentlich immer in der Funktion als Controller und als Talker betrachtet. Wie wir wissen, bleibt er immer der Buscontroller, aber er kann auch als Listener fungieren. Diesen Ablauf haben wir bis jetzt noch nicht beschrieben.

Im Prinzip kennen wir schon alle dazu nötigen Schritte. Während die ATN-Leitung auf »Wahr« ist, sendet der Computer einen Talk-Befehl an das adressierte Gerät. Unmittelbar nachdem die ATN-Leitung wieder auf »Falsch« geht, ist das Gerät immer noch Listener und der Computer Talker. Die DATA-Leitung wird vom Gerät, die CLK-Leitung vom Computer auf »Wahr« gehalten. Das muß sich jetzt aber ändern.

Der Computer zieht seinen DATA-Ausgang auf »Wahr« (die Leitung ist es schon durch das Gerät) und setzt seinen CLK-Ausgang auf »Falsch«. Darauf hat das Peripheriegerät gewartet. Es gibt seinen DATA-Ausgang frei (»Falsch«) und zieht die CLK-Leitung auf »Wahr«. Es hat sich also jetzt wieder der Zustand "Talker hält die CLK-, Listener die DATA-Leitung auf »Wahr«" (siehe den Punkt »Vorbereitung zur Übertragung eines Bytes«) eingestellt. Dieser Ablauf wird vom Computer überwacht und nur wenn alles korrekt abläuft, ist er bereit, Daten zu empfangen. Die Übertragung der Daten erfolgt auf die gleiche Weise wie oben beschrieben.

Damit haben wir alle Übertragungsmöglichkeiten auf dem seriellen Bus erläutert. Wer es verstanden hat, kann die beiden folgenden Fragen beantworten: Warum gibt der Computer keine Fehlermeldung beim Basic-Befehl OPEN4,4 aus, obwohl nur das Gerät 8 abgeschlossen und in Betrieb ist (das Gerät 4 ist nicht abgeschlossen)? Warum gibt er aber eine Fehlermeldung bei PRINT@4, "xxxx" aus (' steht für das Anführungszeichen)?

(Michael Bauer)

Literaturverzeichnis:

- 1) IEC-Bus, Grundlagen — Technik — Anwendungen; Elektronik Sonderheft Nr. 47; Franzis Verlag; München
- 2) Geräte mit dem IEC-Bus einfach gesteuert; Computer Persönlich Nr. 25 vom 15.12.1982 Seite 76; Verlag Markt und Technik; Haar
- 3) Der IEC-Bus-Standard; Computer Persönlich Nr. 24 vom 17.11.1983 Seite 97; Verlag Markt und Technik; Haar
- 4) Piotrowski, Dr. Anton; IEC-Bus; Franzis Verlag; München
- 5) Wunderlich, Franz und Geissen, Bernd; IEE-Bus, Userport + VO-Register; Info-Dienst 1-4; Commodore Benutzer Klub; Frankfurt
- 6) VC 20 Programmierhandbuch; Commodore; Frankfurt
- 7) C 64 Programmierhandbuch; Commodore; Frankfurt
- 8) Butterfield, Jim; How the VIC/64 Serial Bus Works; Computer! Juli 1983 Seite 178; Greensboro
- 9) Englisch, Szczepanowski; Das große Floppy-Buch; Data Becker; Düsseldorf