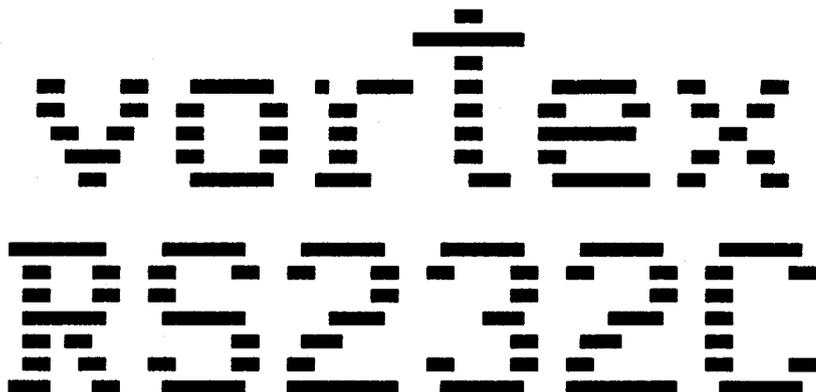


Vortex RS 232 C Schnittstelle

Benutzerhandbuch





BENUTZERHANDBUCH

Rev.1/27.6.86

1. Auflage

(c) 1986 vortex Computersysteme GmbH
Falterstraße 51-53
7101 Flein
Tel.: 07131/52061

Inhaltsverzeichnis

Seite 1	Zu allererst...
Seite 2	Die RS-Schnittstelle
Seite 2	Zur Historie des RS-Standards
Seite 2	Die Verdrahtung der RS-Schnittstelle
Seite 7	Allgemeine Begriffserklärung
Seite 9	Allgemeine Bedienungshinweise
Seite 11	Die serielle Schnittstelle unter CP/M 2.2
Seite 11	INSTALL *** vortex-Laufwerke ***
Seite 13	SETUP *** Amstrad/Schneider-Laufwerke ***
Seite 15	Die serielle Schnittstelle unter CP/M 3.0
Seite 16	Die Basic-Befehle der RS232C-Schnittstelle
Seite 18	BASE - Ram-Basisadresse eines Roms ermitteln
Seite 19	BLOW - eine ASCII-Datei intelligent senden
Seite 20	BREAKSEND - Break an die Schnittstelle senden
Seite 21	CHANNEL - SIO-Kanal wählen
Seite 22	CLOSESIO - auf All sent warten
Seite 22	COUNTER - Zähler im 8253 direkt einstellen
Seite 23	CTRLACTION - Steuerzeichen interpretieren
Seite 24	CTRLDISPLAY - Steuerzeichen darstellen
Seite 24	FULLDUPLEX - kein lokale Verbindung
Seite 25	HALFDUPLEX - lokale Verbindung
Seite 26	INBLOCK - eine Zeichenkette einlesen
Seite 27	INCHAR - ein Zeichen einlesen
Seite 28	INFILE - eingelesene Daten in Datei sichern
Seite 29	INITSIO - Neuinitialisierung der Schnittstelle
Seite 30	NOXON - kein Protokoll bei SERIAL
Seite 30	OUTBLOCK - eine Zeichenkette senden
Seite 31	OUTCHAR - ein Zeichen senden
Seite 32	OUTFILE - den Inhalt einer Datei senden
Seite 32	PARALLEL - Druckerausgabe über Centronics
Seite 33	RINGWAIT - auf Ring Detect warten
Seite 34	ROMCAT -- Katalog aller Roms ausgeben
Seite 35	ROMOFF - Ausschalten von Background-Roms
Seite 36	SETBLOCKEND - Blockende-Kennung definieren
Seite 37	SETENDFILE - Dateiende-Kennung definieren
Seite 37	SETSIO - Einstellen der RS232C-Parameter
Seite 39	SETTIMEOUT - definieren der Timeout-Zeit

Seite 40	SERIAL - Druckerausgabe über RS232C
Seite 41	SIO - Status der Schnittstelle ermitteln
Seite 42	STATUS - Basic-Programmstatus ausgeben
Seite 43	SUCK - eine ASCII-Datei intelligent empfangen
Seite 43	TERMINAL - RS232C als Ein-/Ausgabeeinheit
Seite 44	TRANSFER.COM - CP/M-Transferprogramm generieren
Seite 45	VERSION - Versions-Nummer und -Datum ausgeben
Seite 46	XON - Protokoll bei SERIAL einschalten
Seite 47	ANHANG 1 - Die RAM-Belegung
Seite 48	ANHANG 2 - Die Hardware
Seite 51	ANHANG 3 - Der Schaltplan
Seite 52	ANHANG 4 - Maschinensprache
Seite 54	ANHANG 5 - das Protokoll von BLOW und SUCK
Seite 56	ANHANG 6 - ein Beispiel unter Basic
Seite 59	Die vortex Business Reply Card
Seite 60	Stichwortverzeichnis

ANMERKUNG :

Vervielfältigung und Weitergabe dieses Handbuches - auch auszugsweise - bedürfen der vorherigen schriftlichen Zustimmung der Fa. vortex GmbH, Flein.

Vervielfältigung und Weitergabe der Hard- und Software der vortex RS232C-Schnittstelle ist untersagt.

Alle technischen Daten, Informationen sowie Eigenschaften der in diesem Benutzerhandbuch beschriebenen Schnittstelle wurden nach bestem Wissen zusammengestellt und entsprechen dem Stand bei Drucklegung. Änderungen und Verbesserungen des Produktes sind möglich.

Die Fa. vortex übernimmt keinerlei Gewähr, was die Eignung dieses Produktes für bestimmte Anwendungen betrifft.

Hardware-Entwicklung: Wolfgang Michel
Software-Entwicklung: Roman Majer
Benutzerhandbuch: Roman Majer

Dieses Handbuch wurde auf einem CPC 464 + vortex Speichererweiterung + vortex WD20 + Okidata Microline 292 mit dem Textverarbeitungsprogramm Wordstar 3.0 verfasst und ausgedruckt.

CP/M 2.2 und CP/M 3.0 sind eingetragene Warenzeichen der Fa. Digital Research.
Wordstar ist ein eingetragenes Warenzeichen der Fa. Micropro.

Zu allererst . . .

Seien Sie begrüßt unter den Anwendern der seriellen Datenübertragung! Ein neues professionelles Peripheriegerät von vortex nennen Sie Ihr Eigen und die Möglichkeiten, um die Sie Ihren CPC-Rechner damit erweitert haben, sind gar nicht so schnell zu erfassen.

Mit der RS232C-Schnittstelle von vortex sind nun auch vor Ihnen die Mailboxen nicht mehr sicher, Sie können sich endlich auch mit seriellen Druckern sowie anderer RS-Peripherie herumärgern, oder Sie können von Ihrem Tauschpartner in Timbuktu die neueste Software aus Übersee in minutenschnelle "abstauben" (kostet ja nur hundert Mark Telefongebühren pro Kilobyte), ... HALT HALT HALT HALT ... so könnte es Ihnen gehen, wenn Sie diese Anleitung sofort beiseite legen oder diagonal lesen! Machen Sie es deshalb dieses Mal anders als die meisten Käufer, und führen Sie sich dieses Handbuch **vor** dem Gebrauch der Schnittstelle zu Gemüte - es könnte von Vorteil sein.

Aber was soll das ganze Geunke, Sie lesen ja immer noch, und deshalb fangen wir mal lieber mit einer ganz normalen Einleitung an.

Die Serielle Datenübertragung bringt eine Vielzahl von Fachwörtern mit sich, die erst einmal erklärt werden müssen, bevor man den Gebrauch der vortex-Schnittstelle im Detail darlegt. Deshalb fängt unser Handbuch auch mit einer allgemeinen Erklärung der RS232C-Schnittstelle an und wird mit Hinweisen zur Benutzung der Schnittstelle unter Basic und CP/M weitergeführt. Danach erfolgt eine alphabetische Auflistung aller neuen Basic-Befehle und deren Funktion. Im Anhang finden Sie zum Beispiel den Schaltplan, die RAM-Aufteilung des von der Schnittstelle reservierten RAM-Speichers, Maschinensprachroutinen, etc.

Wir denken, daß Sie in diesem Handbuch den richtigen Einstieg in das weite Feld der RS-Schnittstelle und deren Möglichkeiten haben, und wir können Ihnen eigentlich nur noch viel Spaß beim Gebrauch Ihrer neuen vortex-Schnittstelle wünschen. Es muß ja nicht immer so enden, wie es weiter oben geschildert wurde ...

Ihr vortex Team

Die RS-Schnittstelle

Zur Historie des RS-Standards

In den "Pionierzeiten" der Computerei kam naturgemäß einmal der Ruf nach einem Schnittstellenstandard für die Übertragung von Daten mittels Telefonleitungen auf. 1969 wurde von der EIA (Electronics Industries Association) sowie den Bell-Laboratorien ein entsprechender Standard veröffentlicht. Dies war der EIA RS232-Standard der kurze Zeit später nochmals in kleinen Details geändert und zum RS232C-Standard wurde. Dies ist die Abkürzung für: Recommended Standard Number 232 Revision C. Der Standard hatte bis dahin nur einen Sinn: Die Schnittstelle zwischen Datenendeinrichtungen und Datenübertragungseinrichtungen zur Übertragung von seriellen Daten zu spezifizieren (so der offizielle Titel des Standards). Dabei bezeichnet man als Datenendeinrichtungen (Data Terminal Equipment oder DTE) ein Computer-Terminal und als Datenübertragungseinrichtung (Data Communications Equipment oder DCE) ein Modem. Der Standard besteht im wesentlichen aus drei Teilen:

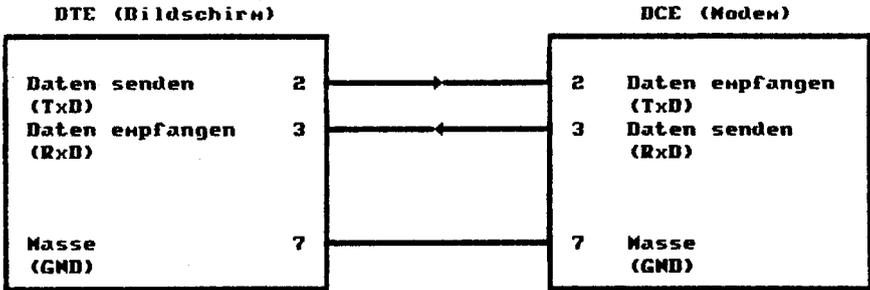
1. **Elektrische Eigenschaft der Signale:** Dieser Teil beschreibt die elektrischen Aus- und Eingänge der Schnittstelle und die Spannungsniveaus der logischen Nullen und Einsen an den Leitungen.
2. **Mechanische Eigenschaften:** Dieser Punkt legt die Eigenschaft der Verbindungselemente fest und besagt, daß die Schnittstelle aus einem Stecker und einer Steckdose besteht. Dabei hat die Steckdose auf der Seite des DCE zu sein. Die eigentliche Form der Verbindungselemente wurde dabei nicht spezifiziert. Der heute allgemein übliche 25-polige D-Sub-Steckverbinder kommt von einem ganz anderen Standard (ISO-Standard) und hat sich im Laufe der Zeit quasi in den RS-Standard "hineingemogelt". Er wird ebenfalls von uns verwendet. Seine genaue Pin-Belegung können Sie dem Anhang entnehmen.
3. **Beschreibung der Leitungen:** Mit diesem Teil wird die Funktion der einzelnen Schnittstellen-Leitungen festgelegt. Es wurden insgesamt 21 Leitungen definiert, von denen in unserem Falle allerdings nur ein Teil relevant ist.

Die Verdrahtung der RS-Schnittstelle

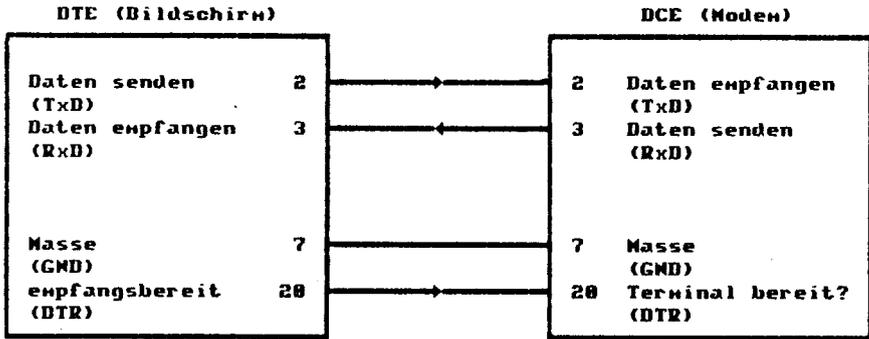
In ihrer primitivsten Form besteht eine serielle Schnittstelle aus nur drei Leitungen. Dies sind Daten senden, Daten empfangen und Masse. Sie gewähren bei gleicher Übertragungsgeschwindigkeit einen bidirektionalen Datenaustausch zwischen DTE und DCE. Für die Ausgänge am DTE gilt immer: Masse ist gemäß Standard immer Pin 7, Daten senden ist Pin 2 und Daten empfangen ist Pin 3. Die englischen Bezeichnungen und daraus resultierenden Abkürzungen für diese drei Leitungen lauten wie folgt:

Daten empfangen - Received Data (RxD)
Daten senden - Transmitted Data (TxD)
Masse - Ground (GND)

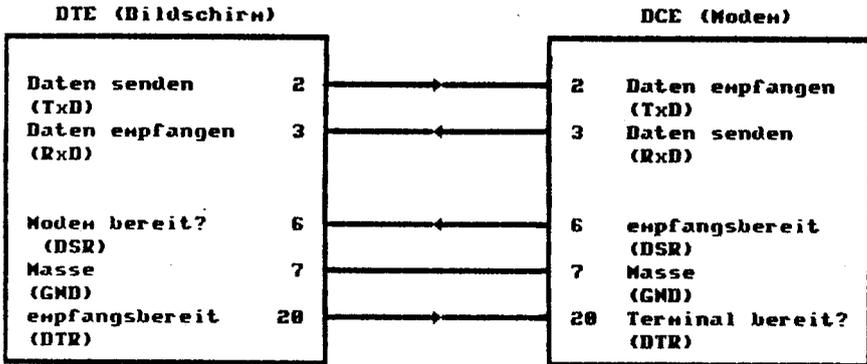
Mit einem Schaubild kann man nun diese Einfach-Schnittstelle verdeutlichen:



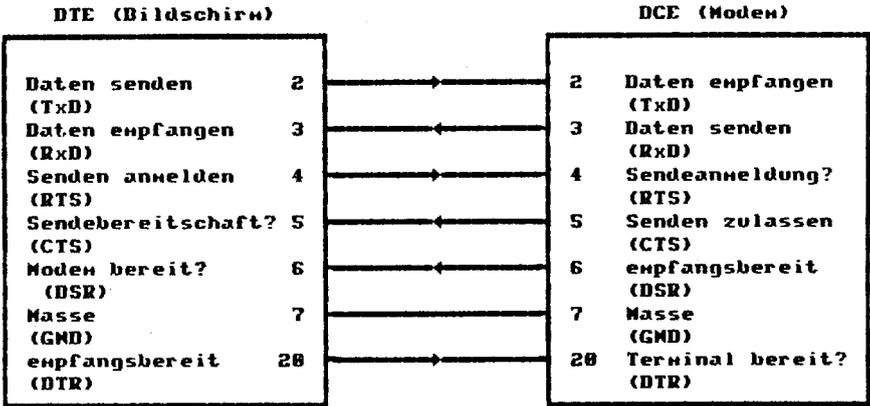
Um eine richtige Übertragung der Daten zu gewährleisten, muß diese mit einem bestimmten Protokoll überprüft werden. Da wir bislang nur drei Leitungen haben, geschieht dies in diesem Fall programmgesteuert. Das heißt, das Terminal sendet ein spezielles Zeichen um dem Modem zu sagen, daß es derzeit nicht empfangsbereit ist, und das Modem wartet solange, bis das Terminal ein anderes spezielles Zeichen zur Bestätigung der Empfangsbereitschaft sendet. Solch ein Protokoll nennt man Software-Handshake. Im Gegensatz dazu gibt es noch den Hardware-Handshake: das Protokoll über spezielle zusätzliche Leitungen. Eine solche Leitung ist bei der RS232C-Schnittstelle der Pin 20: Data Terminal Ready (DTR). DTR heißt "DTE ist empfangsbereit" und wird vom Terminal gesetzt, wenn es empfangsbereit ist und gelöscht, wenn es gerade nicht empfangen kann. Das Modem schaut entsprechend am Pin 20 nach, ob es etwas an das Terminal senden kann oder nicht. Unser Schaubild wurde also um die DTR-Leitung erweitert und sieht so aus:



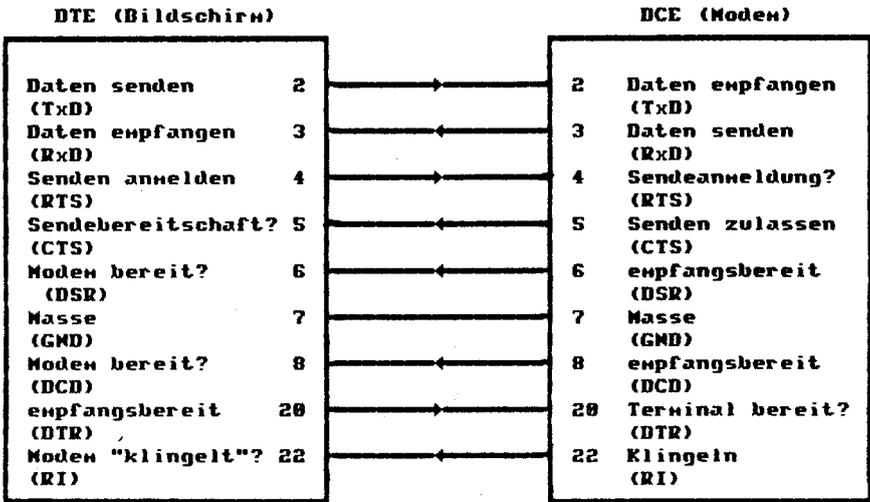
DTR zeigt also Empfangsbereitschaft des Terminals an. Entsprechend gibt es im RS232C-Standard auch das Gegenstück für das Modem, um dem Terminal zu sagen, ob es empfangsbereit ist oder nicht. Die entsprechende Leitung liegt auf Pin 6 und heißt Data Set Ready (DSR). Wenn Pin 6 vom Modem gesetzt wurde kann das Terminal Daten senden, wurde Pin 6 gelöscht, so muß das Terminal mit dem Senden von Daten warten. Unser um eine Leitung erweitertes Schaubild sieht dann so aus:



Zwei weitere Leitungen der RS-Schnittstelle ermöglichen ein erweitertes Übertragungsprotokoll. Es sind die Leitungen Request To Send RTS und Clear To Send CTS. RTS zeigt dabei dem Modem an, daß das Terminal ein Zeichen senden möchte, und das Modem erlaubt das Senden des Zeichens durch Setzen von CTS. Erst dann wird das Zeichen gesendet. Unser wiederum erweitertes Schaubild der RS-Schnittstelle sieht nun so aus:

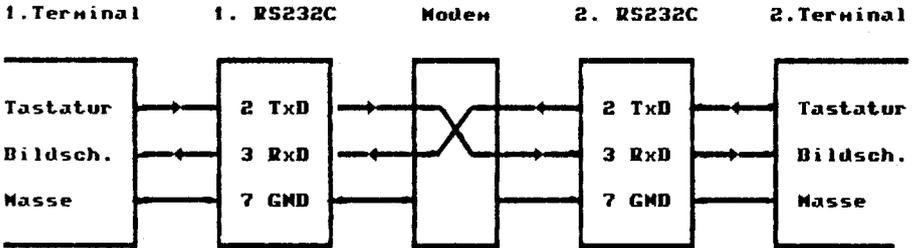


Weitere von uns verwendete Leitungen der RS-Schnittstelle sind Data Carrier Detect (DCD) und Ring Indicator (RI). DCD ist eigentlich von der Funktion her mit DSR gleichzusetzen. Mit RI meldet sich ein Modem am Terminal an, es "klingelt" also und löst damit normalerweise eine Routine am Terminal aus, die die Verbindung herstellt. RI ist allerdings als optionale Funktion der RS-Schnittstelle zu betrachten - Verbindungen können auch ohne RI hergestellt werden. Daraus ergibt sich dann das endgültige Schaubild:



Alle anderen Leitungen der Schnittstelle sind bei uns nicht angeschlossen.

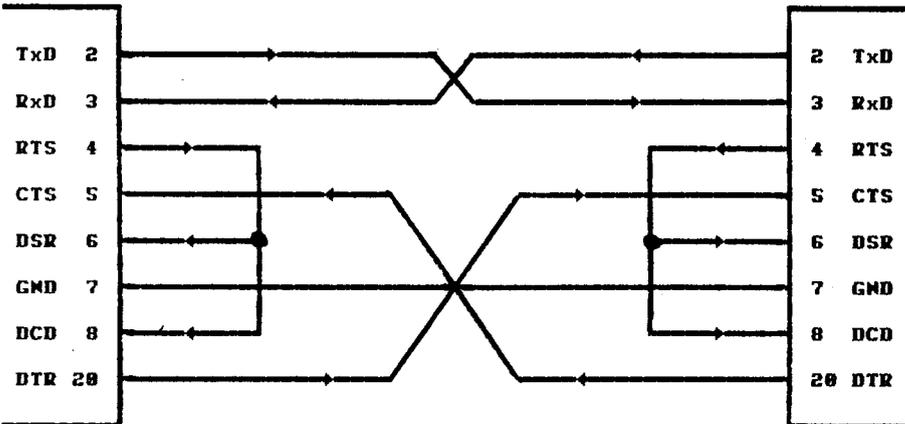
Sollte man die RS-Schnittstelle also im Modem-Betrieb nutzen, dann genügt es, ein Eins-zu-Eins-Verbindungskabel an die Schnittstelle anzuschließen. Man könnte auch sagen, daß der Anschluß eines Modems nur einer Verlängerung der Leitungen entspricht. Dies wäre die einfachste Benutzung der Schnittstelle, so wie sie der RS-Standard auch vorsieht. Eine idealisierte Verbindung zwischen zwei Terminals mittels eines Modems würde dann so aussehen:



Wie Sie sehen, verkreuzt das Modem RxD und TxD automatisch, so daß die vom Terminal 1 gesendeten Daten auf der Empfangsleitung von Terminal 2 transferiert werden und umgekehrt. Schließt man allerdings ein Terminal oder einen Drucker direkt an die RS-Schnittstelle an, so muß man mittels des Kabels dafür sorgen, daß RxD und TxD verkreuzt werden. Normalerweise nimmt man für diesen Fall ein sogenanntes Null-Modem-Kabel (abgekürzt NMC), das von der Verdrahtung so aussieht:

vortex RS232C Schnittstelle

Stecker des Peripheriegerätes



Allgemeine Begriffserklärung

Baud: Die Geschwindigkeit, mit der die RS-Schnittstelle Daten sendet oder empfängt. Dabei wird die Geschwindigkeit in übertragenen Bits pro Sekunde angegeben, wobei hier Datenbits, Start-Parität- und Stopbits gezählt werden. So entsprechen 9600 Baud bei einem Stopbit etwa 960 übertragenen Zeichen.

Datenbits: Die Anzahl der Datenbits gibt den Wertebereich der Daten an, die über die serielle Schnittstelle gesendet und empfangen werden. Bei 8 Datenbits können z.B. Werte zwischen 0 und 255 übertragen werden.

DFÜ: Dies ist das Kürzel für Datenfernübertragung. Damit bezeichnet man die Übertragung von Computer zu Computer über Telefonleitungen.

Eine einfache Möglichkeit DFÜ mit der vortex-RS232C-Schnittstelle zu betreiben ist im Rom der Schnittstelle schon eingebaut: Ein kleines Terminal-Programm (siehe RSX-Befehl TERMINAL) ermöglicht es Ihnen z.B. eine Mailbox anzurufen. Was Sie dazu noch benötigen ist ein Akustik-Koppler, die Telefonnummer einer Mailbox und eine Portion Ausdauer beim Wählen dieser Nummer - Mailboxen sind nämlich sehr stark frequentiert. Ein freie Leitung bei manchen Mailboxen ist schon fast "Sechs Richtigen im Lotto" gleichzusetzen ...

Beachten Sie beim Herstellen einer Verbindung mit dem Befehl TERMINAL bitte, daß Sie natürlich zuvor die Schnittstellen-Parameter entsprechend den Ansprüchen der Mailbox einstellen. Normalerweise arbeiten Mailboxen mit 300 Baud Übertragungsrate, acht Datenbits, einem Stopbit und keiner Paritätsprüfung.

Wenn Sie die DFÜ allerdings zu Ihrem Hobby machen (was mit der hundertprozentigen Auslastung Ihrer Schnittstelle gleichzusetzen ist), so empfiehlt sich ein komfortables DFÜ-Programm. Solch eines wäre beispielsweise das CP/M-Programm KERMIT (Public Domain).

Hardware-Flowcontrol: Dies ist die Möglichkeit von Sender und Empfänger, den Datenfluß über Leitungen der RS-Schnittstelle zu kontrollieren. Diese Leitungen sind CTS und DCD. Eine Erklärung der Leitungen finden Sie im Kapitel "Verdrahtung der Schnittstelle".

Wenn Hardware-Flowcontrol nicht zugelassen ist, sendet und empfängt die Schnittstelle unabhängig vom Zustand der Leitungen CTS und DCD. Die Zustandskontrolle, ob nun Hardware-Flowcontrol ein- oder ausgeschaltet ist, wird mit Pin 20 der Schnittstelle angegeben. Bei eingeschaltetem Zustand wird DTR permanent auf aktiv gesetzt.

Parität: Wenn bei der seriellen Datenübertragung mittels einer RS-Schnittstelle eine Paritätsprüfung zugelassen ist, wird mit einem Bit dem Empfänger angezeigt, ob innerhalb der Datenbits eine gerade oder ungerade Anzahl von gesetzten Bits gesendet wurde. Die für gerade und ungerade stehenden Fachwörter sind EVEN und ODD. Durch eine solche Überprüfung lassen sich Übertragungsfehler schon auf sehr niedriger Ebene relativ sicher feststellen.

Stopbit(s): Um den serielle Datenverkehr zu synchronisieren, werden nach den Datenbits und dem Paritätsbit noch zusätzliche Informationen übergeben. Dies ist das Stopbit oder die Stopbits. Es können vom Anwender ein, eineinhalb oder zwei Stopbits eingestellt werden.

Timeout: Wenn die RS-Schnittstelle unter Basic ein Zeichen absenden soll, und dies nach einer gewissen Zeit immer noch nicht möglich war, erfolgt ein sogenannter "Timeout". Dies bedeutet, daß der Versuch das Zeichen zu übertragen abgebrochen, und eine entsprechende Fehlermeldung rückgemeldet wird. Die Zeit, nach der ein Timeout erfolgt, kann durch den Benutzer definiert werden (mit dem RSX-Befehl SETTIMEOUT).

Timer-Baustein: Zur Generierung der verschiedenen Baudraten ist auf der RS-Schnittstelle ein 8253-Timer-Baustein enthalten, der richtig programmiert dem SIO-Baustein Teilerverhältnisse liefert. Aus dem Grundtakt und diesen Teilerverhältnissen stellt der SIO-Baustein dann die Baudraten her.

Z80-SIO oder SIO: Abkürzung für Serial Input Output. Damit ist der serielle Peripheriebaustein der Z80-Bauteile-Familie gemeint, welcher normalerweise das Herz einer seriellen Schnittstelle an einem Z80-Computer ist. In der vortex-RS232C-Schnittstelle haben wir allerdings den Z80-DART verwendet, welcher eine Untermenge der Z80-SIO darstellt. Wenn also in diesem Handbuch von SIO oder DART gesprochen wird, so ist immer der DART-Baustein gemeint.

Allgemeine Bedienungshinweise

Herstellen der Hardware-Verbindung

Vor Gebrauch der Schnittstelle sollte diese zuerst mit dem Peripheriegerät verbunden werden. Das geschieht mit einem Kabel. Welche Leitungen Sie dabei nehmen müssen, entnehmen Sie bitte dem Kapitel "Verdrahtung der Schnittstelle". Sollten Sie nicht mit einem Modem arbeiten, so beachten Sie bitte, daß RxD und TxD (Pin 2 und 3) von Ihnen gekreuzt werden.

Einstellen der Geschwindigkeit am Peripheriegerät

Bei der RS-Schnittstelle haben Sie die Möglichkeit die Übertragungsgeschwindigkeit (=Baudrate) zu ändern. Deshalb müssen Sie sicherstellen, daß sowohl Ihre RS-Schnittstelle, wie auch das angeschlossene Peripheriegerät (z.B. Drucker oder Terminal) mit der gleichen Geschwindigkeit arbeiten. Drucker haben im allgemeinen die Möglichkeit, Baudraten mittels kleiner Schalter (=DIP-Switches) einzustellen. Wie dies an Ihrem Gerät geht, entnehmen Sie bitte dem entsprechenden Handbuch. Eine andere Möglichkeit bieten zum Beispiel Terminals, bei denen sich solche Parameter in einem Installationsmenü einstellen lassen. Die Übertragungsgeschwindigkeit, die Sie empfehlenermaßen einstellen sollten, ist 9600 Baud. Diese Baudrate gewährleistet sowohl eine akzeptable Geschwindigkeit, wie auch eine gute Übertragungssicherheit. Zudem ist dies die Standardeinstellung der vortex RS-Schnittstelle.

Die Geschwindigkeit der RS-Schnittstelle einstellen

Die RS-Schnittstelle arbeitet nach dem Einschalten mit einer Übertragungsgeschwindigkeit von 9600 Baud. Um diese an ein angeschlossenes Peripheriegerät anpassen zu können, haben Sie verschiedene Möglichkeiten:

Unter Basic: Den RSX-Befehl */SETSIO,baudrate*

CP/M 2.2: Die Baudrate der RS-Schnittstelle wird beim Booten von CP/M eingestellt. Um diese Einstellung zu ändern gibt es die Programme *INSTALL* (vortex-Laufwerke) und *SETUP* (Schneider-/Amstrad-Laufwerke). Eine nähere Beschreibung beider Programme finden Sie im CP/M-Teil des Handbuches.

CP/M 3.0: Die nach dem Booten von CP/M eingestellte Baudrate läßt sich hier mit den Befehlen *SETSIO,baudrate* und *DEVICE SIO(baudrate)* einstellen.

Beispiel: Um die Baudrate auf 2400 Baud herunterzusetzen lautet der Basicbefehl `/SETSIO,2400` und unter CP/M 3.0 geben Sie `SETSIO 2400` oder `DEVICE SIO[2400]` ein. Bitte beachten Sie dabei, daß eine Änderung mittels SETSIO nicht mit DEVICE rückgefragt werden kann. Sollten Sie unter CP/M 3.0 beim Laden schon eine andere Geschwindigkeit haben wollen, so übernehmen Sie den obigen DEVICE-Befehl in die Datei PROFILE.SUB.

Datenbits, Stopbits und Parität einstellen

Zusätzlich zu der Geschwindigkeit sind noch einige andere Randbedingungen richtig einzustellen, um regulären Datenfluß zu gewährleisten. Zum einen ist dies die Anzahl der Datenbits, die zwischen 5 und 8 frei wählbar ist, die Anzahl der Stopbits (eins, eineinhalb oder zwei), und ob eine Paritätsprüfung stattfinden soll (wenn ja: gerade oder ungerade Parität?). Wie Sie diese Parameter an Ihrem angeschlossenen Peripheriegerät einstellen, finden Sie in dem zugehörigen Handbuch. Die Einstellung dieser Parameter an der vortex RS-Schnittstelle finden Sie im Basic- und CP/M-Teil dieses Handbuchs.

Datenbits, Stopbits und Parität müssen genauso wie die Übertragungsgeschwindigkeit am Peripheriegerät und an der RS-Schnittstelle gleich eingestellt sein, damit der Datenfluß regulär abgewickelt wird.

Die serielle Schnitt- stelle unter CP/M 2.2

Der auf der RS-Schnittstelle enthaltene Rom wird nicht unter CP/M benutzt, da hier die entsprechenden RS-Treiberrouinen schon im Diskettenbetriebssystem (VDOS 2.1 oder AMSDOS) enthalten sind und voll in das CP/M 2.2-Betriebssystem integriert wurden. Das heißt also im Klartext, daß die gesamte Verwaltung der RS-Schnittstelle schon vom CP/M 2.2 selbst "gemanaged" wird, und die Arbeit mit der Schnittstelle absolutem Standard entspricht. Nun einige Worte zu diesem "Standardbetrieb" einer RS-Schnittstelle unter CP/M 2.2:

Schon beim Start (neudeutsch: Booten) des CP/M wird die SIO initialisiert. Dies geschieht mittels der im Parametersektor auf Diskette abgelegten Einstellungen, welche Sie mit den Dienstprogrammen INSTALL (vortex) oder SETUP (Amstrad/Schneider) neu einstellen können.

1. INSTALL *** nur vortex-Laufwerke!!! ***

A>install <ENTER>

INSTALL 1.0 - Power up Installationen (C)1985 vortex GmbH

Im Parametersektor sind noch 439 Bytes frei

Änderung der

- 1 - Floppy Disk Zeitkonstanten
- 2 - Tastaturbelegung
- 3 - CP/M Parameter (I/O Byte, Power-up Strings)
- 4 - Communication Interface Parameter
- 5 - Installation/Ende

Ihre Wahl:4

Im Parametersektor sind noch 439 Bytes frei

Communication Interface Parameter

- 1 Baudrate Kanal A:Senden 9600
- 2 Baudrate Kanal A:Empfangen 9600
- 3 Baudrate Kanal B:Senden/Empfangen 9600
- 4 Kanal A:8 Datenbits, 1 Stopbit, keine Paritaet
- 5 Kanal B:8 Datenbits, 1 Stopbit, keine Paritaet
- 6 Hauptmenue

Ihre Wahl: 1

Mögliche Werte: 50, 75, 110, 150, 200, 300, 600, 1200, 1800,
2000, 2400, 3600, 4800, 9600, 19200

Neuer Wert: 3600 <ENTER>

Im Parametersektor sind noch 439 Bytes frei

Communication Interface Parameter

- 1 Baudrate Kanal A:Senden 3600
- 2 Baudrate Kanal A:Empfangen 9600
- 3 Baudrate Kanal B:Senden/Empfangen 9600
- 4 Kanal A:8 Datenbits, 1 Stopbit, keine Paritaet
- 5 Kanal B:8 Datenbits, 1 Stopbit, keine Paritaet
- 6 Hauptmenue

Ihre Wahl: 6

Im Parametersektor sind noch 439 Bytes frei

Anderung der

- 1 - Floppy Disk Zeitkonstanten
- 2 - Tastaturbelegung
- 3 - CP/M Parameter (I/O Byte, Power-up Strings)
- 4 - Communication Interface Parameter
- 5 - Installation/Ende

Ihre Wahl:5

Neue Parameter installieren J/N ? J

CP/M neu starten J/N ? J

44k CP/M vers. 2.2 - 08/85 vortex GmbH

A>

Wenn Sie diese Sequenz mit dem Programm INSTALL und einem vortex-Diskettenlaufwerk durchlaufen haben, wurde im Parametersektor der eingelegten CP/M-Diskette als Default-Wert für <Baudrate Kanal A senden> den Wert 3600 abgelegt. Nach Neustart des CP/M von dieser Diskette ist also 3600 Baud für Kanal A senden eingestellt.

2. SETUP * nur Amstrad/Schneider-Diskettenlaufwerke!!! *****

CP/M 2.2 - Amstrad Consumer Electronics plc

A>setup <ENTER>

** Initial command buffer empty

Is this correct (Y/N): Y

Drücken Sie nun solange die Taste Y, bis Sie zur dreizehnten Frage kommen, die so aussieht:

Z80 SIO Channel A: 9600 tx baudrate, 9600 rx baudrate, 8 data bits
NO parity, 1 stop bit

Is this correct (Y/N): N

Enter channel A parameters: 3600 <ENTER>

Z80 SIO Channel A: 3600 tx baudrate, 9600 rx baudrate, 8 data bits
NO parity, 1 stop bit

Is this correct (Y/N): Y

Z80 SIO Channel B: 9600 baudrate, 8 data bits
NO parity, 1 stop bit

Is this correct (Y/N): Y

Do you want to update your system disc (Y/N): Y

Do you want to restart CP/M (Y/N): Y

CP/M 2.2 - Amstrad Consumer Electronics plc

A>

Nach durchlaufen dieser Fragen im Programm SETUP haben Sie als Default-Parameter für Kanal A senden 3600 Baud im Parametersektor Ihrer CP/M-Diskette abgelegt und durch den CP/M-Neustart gleich installiert.

Zulässige Werte im Programm SETUP für Baudraten sind: 50, 75, 110, 150, 200, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 9600, 19200. Die mögliche Anzahl der Datenbits ist 5, 6, 7 oder 8. Um die Parität zu ändern geben Sie ODD, EVEN oder NONE ein. Stopbits muß eine der Zahlen 1, 1.5 oder 2 sein. Im Beispiel oben wurde nur der erste Parameter der SIO-Einstellung geändert. Wollen Sie allerdings mehrere Parameter ändern, so trennen Sie die Parameter durch eine Leerstelle. Das würde dann beispielsweise so aussehen: 19200 19200 7 EVEN 2 <ENTER>.

Wollen Sie nun, nach erfolgter Einstellung der SIO-Parameter, über die RS-Schnittstelle etwas ausgeben oder empfangen, so geschieht dies mittels der CP/M 2.2-Utility STAT.

STAT wird dazu verwendet den logischen Einheiten des CP/M 2.2 physikalische Einheiten zuzuordnen. Logische Einheiten können hierbei sein: CON, RDR, PUN und LST, die physikalischen Einheiten sind TTY, CRT und LPT. Normalerweise sind die Einheiten so zugeordnet:

Konsole (CON:)	---	Tastatur/Bildschirm (CRT:)
Drucker (LST:)	---	Centronics-Schnittstelle (LPT:)
Eingabegerät (RDR:)	---	RS232C-Schnittstelle (TTY:)
Ausgabegerät (PUN:)	---	RS232C-Schnittstelle (TTY:)

Will man diese Zuordnung nun mittels STAT ändern, so geschieht dies nach folgender Allgemeinregel:

STAT logische Einheit=physikalische Einheit

Möchte man zum Beispiel die Druckerausgabe auf die serielle Schnittstelle umleiten, da dort ein serieller Drucker angeschlossen ist, so lautet die Befehlssequenz STAT LST:=TTY: <ENTER>.

Danach erfolgen alle Druckerausgaben (z.B. nach Eingabe von CTRL P) nicht mehr auf die eingebaute Centronics-Schnittstelle, sondern auf die RS-Schnittstelle.

Zur Übertragung von Dateien unter CP/M 2.2 können Sie das Programm TRANSFER.COM verwenden, welches Ihnen mit dem RSX-Befehl TRANSFER.COM auf Diskette generiert wird. Es sendet oder empfängt Dateien entsprechend dem im Anhang 5 beschriebenen Software-Protokoll. Der einzige Unterschied: Sie können auch andere als nur ASCII-Dateien senden.

Zur Generierung und Handhabung des Programmes schauen Sie bitte im Kapitel "Die Basic-Befehle" unter der Beschreibung des Befehls TRANSFER.COM nach.

Die serielle Schnittstelle unter CP/M 3.0

Genau wie unter CP/M 2.2 wird auch beim Booten von CP/M 3.0 die serielle Schnittstelle entsprechend den im EMS-File abgelegten Standardwerten sofort initialisiert. Eine Änderung dieser Standardwerte ist allerdings nicht direkt im EMS-File möglich (etwa so, wie unter CP/M 2.2 mittels INSTALL oder SETUP der Parametersektor geändert wird). Unter CP/M 3.0 muß man das Programm SETSIO mit entsprechenden Parametern in die Datei PROFILE.SUB einbinden, die ja nach jedem Bootvorgang einmal abgearbeitet wird (sofern sie auf Diskette vorhanden ist!).

Das BIOS des CP/M 3.0 beinhaltet als physikalische Einheit die serielle Schnittstelle und ähnlich dem im vorhergehenden Kapitel beschriebenen Vorgang mittels STAT, besteht unter CP/M 3.0 die Möglichkeit, logischen Einheiten andere physikalische Einheiten zuzuordnen. Hierzu verwendet man unter CP/M 3.0 allerdings das Programm DEVICE, welches ebenfalls mit der Syntax `DEVICE logische Einheit=physikalische Einheit` aufgerufen wird.

Als Default-Einstellung nach dem Booten von CP/M 3.0 sind die physikalischen Einheiten den logischen Einheiten wie folgt zugeordnet:

```
Konsole (CON:)      ---> Tastatur/Bildschirm (CRT)
Drucker (LST:)     ---> Centronics-Schnittstelle (LPT)
Zusatzeinheit (AUX:)-> RS232C-Schnittstelle (SIO)
```

Geben Sie beispielsweise als Befehl `DEVICE LST:*SIO <ENTER>` ein, so erfolgt die Ausgabe auf den Druckerkanal nicht mehr über die eingebaute Centronics-Schnittstelle (LPT), sondern über die angeschlossene RS-Schnittstelle (SIO).

Wollen Sie nun die Parameter der seriellen Schnittstelle ändern, so müssen Sie dies, wie weiter oben schon angesprochen, mit dem Programm SETSIO machen. Der Aufruf erfolgt so:

```
setsio tx 110 rx 3600 bits 8 parity none stop 2 xon off handshake on <ENTER>
```

wobei die Reihenfolge und Anzahl der übergebenen Parameter unbedeutend ist. Wichtig ist nur, daß die Werte der übergebenen Parameter regulär sind. Diese Werte sind:

Baudraten (RX und TX): 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200.

Databits (BITS): 5, 6, 7, 8.

Parität (PARITY): EVEN, ODD, NONE.

Stopbits (STOP): 1, 1.5, 2.

Software-Protokoll (XON): ON, OFF.

Hardware-Flowcontrol (HANDSHAKE): ON, OFF.

Die Basic-Befehle der RS232C-Schnittstelle

Die vortex-RS232C-Schnittstelle wird serienmäßig mit einem ROM ausgeliefert, der die Schnittstelle für den Benutzer durch neue Befehle unter Basic sehr einfach nutzbar macht. Diese zusätzlichen Basic-Befehle sind als sogenannte RSX-Befehle in das Betriebssystem des CPC-Rechners eingebunden und werden durch Voranstellen des "RSX-Balkens" vor den Befehlsnamen aufgerufen. Diesen Balken erreichen Sie durch gleichzeitiges Drücken der Tasten SHIFT und @. Ein komplett eingegebener RSX-Befehl sieht also beispielsweise so aus: */fullduplex <ENTER>*.

Einige der zusätzlichen Befehle benötigen beim Aufruf noch einen oder mehrere Parameter. Sind in der Beschreibung Parameter in Klammern gesetzt, so sind sie optional und müssen nicht unbedingt übergeben werden. Ansonsten sind sie allerdings zwingend zu übergeben.

Parameter, die nur Werte definieren (beispielsweise Baudraten bei dem RSX-Befehl SETSIO) müssen Integerwerte sein, d.h. Zahlenwerte zwischen 0 und 65535. Diese können direkt eingegeben (z.B.: */setsio,4800*) oder in Variablen übergeben werden (z.B.: *defint b:b=4800:/setsio,b*).

Werden als Parameter allerdings Variablen erwartet, um Inhalte aus ihnen zu lesen oder deren Inhalte zu verändern (z.B. bei den Statusrückmeldungen), so ist die Adresse der Variablen als Parameter zu übergeben. Dies geschieht ganz einfach durch Voranstellen des @ vor den Variablennamen. Als Beispiel: *defint s:s=@:/sio,@s*. Bitte beachten Sie dabei, daß die übergebenen Variablen wirklich definiert sind, d.h. das ihnen zuvor schon einmal ein Wert zugewiesen wurde, und daß numerische Variablen immer vom Typ INTEGER sein müssen.

Die RSX-Befehle des RS232C-Roms lassen sich prinzipiell in acht Gruppen einordnen:

1. Befehle zur grundlegenden Handhabung der seriellen Schnittstelle. Dazu zählen die Befehle BREAKSEND, CHANNEL, CLOSESIO, INITSIO, RINGWAIT, SETSIO, SETTIMEOUT und SIO.

2. Kommandos zur Übertragung von einzelnen Zeichen: INCHAR und OUTCHAR.

3. Kommandos zur Übertragung von Zeichenketten: INBLOCK, OUTBLOCK und SETBLOCKEND.

4. Dateiübertragung ohne Softwareprotokoll: INFILE, OUTFILE und SETENDFILE.

5. Dateiübertragung mit Softwareprotokoll: BLOW und SUCK.

6. Datenübertragung im Terminal-Modus: CTRLACTION, CTRLDISPLAY, FULLDUPLEX, HALFDUPLEX und TERMINAL.

7. Einbinden eines seriellen Druckers. Als Befehle dienen hierzu NOXON, PARALLEL, SERIAL und XON.

8. sonstige Befehle: BASE, STATUS, ROMCAT und ROMOFF.

Bei der Programmierung unserer Schnittstelle lehnten wir uns an die Syntax der von der Fa. Amstrad in England vertriebenen RS-Schnittstelle an und können damit an der Benutzeroberfläche die größtmögliche Kompatibilität bieten. In klaren Worten heißt das, daß sämtliche Befehle der Amstrad-Schnittstelle (bis auf die in Deutschland sowieso recht sinnlose PRESTEL-Emulation und dieser Befehlsgruppe) eine Untermenge unserer Befehle sind und die gleiche oder eine erweiterte Funktion haben. Allerdings unterscheiden sich die Befehle auf Maschinensprach-Ebene komplett und die Aufteilung des RS-Rambereiches (siehe Anhang 1) differiert ebenfalls. Also alle Maschinenprogrammierer bei direkten Romeinsparungen aufgepasst!

Desweiteren folgt nun eine alphabetische Erklärung der Basic-Befehle.

BASE - Ram-Basisadresse eines Roms ermitteln

Aufruf: *lBASE, Romnummer, @Statusvariable <ENTER>*

Rückgabe: die Ram-Basisadresse in der Statusvariablen.

Jeder in das Basic-Betriebssystem eines CPC-Rechners eingebundene Erweiterungs-Rom benötigt einen Arbeitsbereich im Ram des Computers. Dieser wird von dem jeweiligen Rom beim Einschalten des Systems automatisch reserviert und vom frei verfügbaren Basic-Speicherplatz abgezogen. Wenn Sie eine Diskettenstation besitzen, ist Ihnen dies sicherlich schon aufgefallen.

Oft genug kann es vorkommen, daß man zum direkten Zugriff auf den Ram-Bereich eines Roms dessen Startadresse benötigt. Dies könnte zum Beispiel der Fall sein, wenn Sie im Diskcontroller-Arbeitsbereich den Disk-Parameterblock manipulieren oder im SIO-Arbeitsbereich als Beispiel die Baudraten-Teiler direkt ändern wollen.

Diese Start- oder Basisadresse des von einem Background-Rom reservierten Rambereiches wird mit dem BASE-Befehl in die zuvor definierte Integer-Statusvariable eingetragen. Bitte beachten Sie, daß die gültigen Rom-Nummern beim CPC 464 zwischen 0 und 7, beim CPC 664 und 6128 zwischen 0 und 15 liegen dürfen. Wird ein unzulässiger Wert als Rom-Nummer übergeben, kehrt der Befehl ohne Auswirkung zurück.

Beispiel:

*** Baudrate von Kanal B auf 19200 setzen ***

Ready

10 defint s <ENTER>

20 s=0 <ENTER>

30 lbase,2,@s <ENTER>

40 s=s+10 <ENTER>

50 poke s,7:poke s+1,0

60 print "in Adresse &";hex\$(s);" steht der 8253-Teiler."

Ready

■

Erklärung: in den Zeilen 10 und 20 wird als erstes die Integer-Variable S definiert, die als Rückmelde-Variable fungieren wird. Danach erfolgt die Ermittlung der Ram-Basisadresse des SIO-Roms. Bitte beachten Sie dabei, daß dieser von uns standardmäßig mit der Rom-Nummer 2 ausgeliefert wird und deshalb an dieser Stelle bei Aufruf des BASE-Befehls auch im Beispiel die Nummer 2 übergeben wird. Sollten Sie auf der RS-Platine die Rom-Nummer geändert haben, müssen Sie in Zeile 30 entsprechend eine andere Rom-Nummer übergeben (die richtige Rom-Nummer können Sie mit dem RSX-Befehl

ROMCAT ermitteln!).

Gemäß Anhang 1 (Rambellegung des SIO-Roms) steht in der elften und zwölften Speicherzelle des von der SIO reservierten Rams der für die Baudrate des Kanals B zuständige Teiler für den Timer-Baustein 8253 (der Teiler hat den Wert 125000/Baudrate). Durch Einschreiben des Wertes &0007 (Teiler für Baudrate 19200) in diese Adressen (dies geschieht in Zeile 40 und 50) und nachfolgende Initialisierung der SIO mit dem RSX-Befehl INITSIO haben Sie dann 19200 als Baudrate für Kanal B eingestellt.

verwandte Befehle:

ROMCAT, ROMOFF

BLOW - eine ASCII-Datei intelligent senden

Aufruf: `/BLOW, @Dateiname <ENTER>`

Rückgabe: nichts.

Oft will man von einem Computersystem zum anderen Daten übertragen, was Sie sicherlich bislang per Diskette oder Cassette getan haben. Mit der RS-Schnittstelle haben Sie jetzt eine Möglichkeit, das Übertragen direkt und schnell zu vollziehen: per Telefonleitung und Akkustikkoppler oder per direkter Kabelverbindung eine ASCII-Datei an einen anderen Rechner einfach abzusenden. Dieser andere Rechner muß allerdings ebenfalls mit einer seriellen Schnittstelle von vortex oder Amstrad ausgerüstet sein, damit Sie eine ASCII-Datei mit dem BLOW-Befehl übertragen können. Der Grund hierfür ist das von BLOW und SUCK verwendete Übertragungsprotokoll, mit dem gewährleistet wird, daß die gesendeten Daten am anderen Ende der Leitung auch richtig empfangen werden. Ganz präzise gesagt: Wenn Sie eine Datei mit BLOW senden wollen, geht dies nur, wenn die Daten mit dem SUCK-Befehl an einem anderen CPC empfangen werden.

Hierbei ist noch zu beachten, daß der Empfänger den Transfer-Vorgang zuerst mit dem SUCK-Befehl starten muß, bevor der Sender mit dem BLOW-Befehl die Übertragung beginnt. Ebenfalls wichtig: Es können mit diesem RSX-Befehl nur ASCII-Dateien gesendet werden, d.h. mittels dem Basic-Befehl OPENOUT oder unter CP/M erstellte Dateien, sowie CP/M-Programme. Wollen Sie Basic-Programme übertragen, so müssen Sie diese zuerst im ASCII-Format abspeichern (Bsp.: `SAVE "TEST",A`). Eine direkte Übertragung von "normal" abgespeicherten Basic-Programmen oder Binär-Dateien ist nur unter CP/M mit dem Programm TRANSFER.COM möglich, welches mit dem gleichen Protokoll wie BLOW und SUCK arbeitet.

Soweit die prinzipielle Erklärung des intelligenten Datei-

Transfers ("intelligent" da Fehler erkannt und berichtigt werden). Wenn Sie interessieren sollte, wie das Protokoll von BLOW und SUCK intern aussieht, so können Sie dies im Anhang 5 nachlesen.

Beispiel:

*** Senden der Datei WS.COM ***

Ready

as="ws.com" <ENTER>

!blow,@a\$ <ENTER>

Datei uebertragen

■

Erklärung: Wenn ein Empfänger den Befehl SUCK gestartet hatte und die Parameter der beiden Schnittstellen genau gleich eingestellt sind, wird die Datei WS.COM gesendet. Es wird in der Folge eine von vier möglichen Meldungen ausgegeben: 1. Datei nicht vorhanden, 2. keine ASCII-Datei, 3. Fehler aufgetreten oder 4. Datei gesendet.

verwandte Befehle:

SUCK

BREAKSEND - warten bis alle Zeichen gesendet sind und dann ein Break-Kommando an die RS232C-Schnittstelle senden

Aufruf: **!BREAKSEND, @Statusvariable, Break-Zeit <ENTER>**

Rückgabe: in der Statusvariablen steht:

- 0 - Break wurde ordnungsgemäß gesendet.
- 512 - Timeout erfolgte, ohne das der Sende-Puffer leer war, Break wurde nicht gesendet.

Es wird gewartet, bis der Sende-Puffer (Transmitter Buffer) der RS232C-Schnittstelle leer ist. Erfolgte allerdings zuvor ein Timeout (siehe SETTIMEOUT), so kehrt die Routine mit einem Fehlercode zurück. Wenn der Buffer leer war, wird noch eine gewisse Zeit mit einem gesetzten Break-Bit im Write-Register 5 der SIO gewartet. Diesen Zeitraum bestimmt man mit dem zweiten Parameter, der sogenannten Break-Zeit. Sie darf zwischen 0 und 65535 liegen und gibt die zu wartende Zeit in Millisekunden an.

Beispiel:

*** auf <all sent> prüfen und 2 Sekunden <break> senden ***

```
defint s <ENTER>
s=0 <ENTER>
!breaksend,@s,2000 <ENTER>
Ready
■
```

verwandte Befehle:

CLOSESIO, RINGWAIT, SIO

CHANNEL - selektieren des aktiven RS232C-Kanales (in XRS- und RS-Modul nicht vorhandenn!!!)

Aufruf: *!CHANNEL,Kanalnummer <ENTER>*

Rückgabe: nichts.

Die vortex-RS232C-Schnittstelle basiert auf einem DART-Baustein mit zwei Kanäle, wobei die Schnittstelle im vortex-XRS-Modul nur den Kanal A zur Verfügung stellt. Besitzen Sie eine vortex-SPplus-Speichererweiterung oder einen vortex-"Super-Controller", so können Sie auch den Kanal B der RS232C-Schnittstelle benutzen. Um in diesem Fall zwischen den Kanälen umschalten zu können, steht Ihnen der Befehl CHANNEL zur Verfügung. Der Parameter gibt an, auf welchen Kanal der Schnittstelle in der Folge ausgegeben und empfangen wird, wobei nur das untere Bit des Wertes relevant ist, d.h. eine gerade Zahl selektiert Kanal A und eine ungerade Zahl selektiert Kanal B.

Bitte beachten Sie nochmals, daß dieser Befehl beim XRS-Modul nicht implementiert ist, da dort nur der Kanal A zur Verfügung steht.

Beispiel:

*** Kanal B der RS232C anwählen (sofern verfügbar) ***

```
Ready
!channel,1 <ENTER>
Ready
■
```

verwandte Befehle:

CLOSESIO - warten bis der Sende-Puffer leer ist

Aufruf: *!CLOSESIO, @Statusvariable <ENTER>*

Rückgabe: in der Statusvariablen steht 0, wenn der Befehl ordnungsgemäß beendet wurde oder 256, wenn zuvor die Timeout-Zeit überschritten wurde.

Es wird gewartet, bis der Sende-Puffer leer ist ('all sent' gesetzt), und danach wird der für Hardware-Flowcontrol zuständige Pin 20 (DTR) abgeschaltet. Erfolgt allerdings zuvor ein Timeout, so wird die Routine mit dem Fehlercode 256 in der Statusvariablen verlassen.

Beispiel:

*** die Schnittstelle "schließen" ***

```
Ready
defint s <ENTER>
s=0 <ENTER>
!closesio,@s <ENTER>
Ready
■
```

verwandte Befehle:

BREAKSEND, CLOSESIO, RINGWAIT, SIO

COUNTER - Zähler im 8253 direkt einstellen

Aufruf: *!COUNTER, cnt0 (,cnt1 (,cnt2)) <ENTER>*

Rückgabe: nichts.

Mittels einem bis drei Parameter lassen sich die Counter im 8253-Timerbaustein direkt verändern, was allerdings nur absolute Experten tun sollten. Sie haben damit die Möglichkeit, andere als die bei SETSIO erlaubten Baudraten zu generieren. Zur näheren Erläuterung des 8253 betrachten Sie bitte das Beispiel bei dem Befehl BASE oder Anhang 2 (Die Hardware).

Beispiel:

*** Kanal A Sendebaudrate auf 19200 setzen ***

Ready
/counter,60007 <ENTER>
Ready
■

verwandte Befehle:

SETSIO

CTRLACTION - Steuerzeichen im Terminal-Modus interpretieren

Aufruf: /CTRLACTION <ENTER>

Rückgabe: nichts.

Nach Eingabe von CTRLACTION werden alle von der RS232C-Schnittstelle empfangenen Steuerzeichen im Terminal-Modus (siehe RSX-Befehl TERMINAL) wieder als solche auf dem Bildschirm ausgegeben, was auch der Default-Zustand nach Einschalten der Schnittstelle ist.

Beispiel:

*** Steuerzeichen im Terminal-Mode als solche interpretieren ***

Ready
/ctrlaction <ENTER>
Ready
■

Erklärung: Wird nach diesem Befehl mit TERMINAL der Terminal-Modus aufgerufen, und ein an die RS232C-Schnittstelle angeschlossenes Gerät liefert den ASCII-Wert 12, so wird am CPC-Rechner der Bildschirm gelöscht.

verwandte Befehle:

CTRLDISPLAY, FULLDUPLEX, HALFDUPLEX, TERMINAL

CTRLDISPLAY - Steuerzeichen im Terminal-Modus graphisch darstellen

Aufruf: */CTRLDISPLAY <ENTER>*

Rückgabe: nichts.

Im Terminal-Modus (nach Gebrauch des RSX-Befehls *TERMINAL*) werden die Steuerzeichen nicht mehr als solche interpretiert, sondern es werden die entsprechenden Graphikzeichen mit den ASCII-Werten 0 bis 31 dargestellt. Dies entspricht beispielsweise dem Basic-Befehl *PRINT CHR\$(1)CHR\$(12)*, welcher nicht mehr den Bildschirm löscht, sondern das Zeichen mit dem ASCII-Wert 12 ausgibt.

Beispiel:

*** Steuerzeichen im Terminal-Mode als Zeichen darstellen ***

```
Ready
/ctrldisplay <ENTER>
Ready
■
```

verwandte Befehle:

CTRLACTION, FULLDUPLEX, HALFDUPLEX, TERMINAL

FULLDUPLEX - unterbricht die lokale Verbindung zwischen Tastatur und Bildschirm im Terminal-Modus

Aufruf: */FULLDUPLEX <ENTER>*

Rückgabe: nichts.

Unterbricht die durch *HALFDUPLEX* hergestellte Verbindung zwischen Tastatur und Bildschirm, was auch der Default-Zustand des Terminal-Modus ist.

Beispiel:

*** keine lokale Verbindung im Terminal-Mode ***

```
Ready
/fullduplex <ENTER>
Ready
■
```

Erklärung: nach Aufruf des Terminal-Modus (mit |TERMINAL <ENTER>) werden nur noch die von der Schnittstelle empfangenen Zeichen auf dem Bildschirm dargestellt.

verwandte Befehle:

CTRLACTION, CTRLDISPLAY, HALFDUPLEX, TERMINAL

HALFDUPLEX - stellt eine lokale Verbindung zwischen Tastatur und Bildschirm in Terminal-Modus her

Aufruf: |HALFDUPLEX <ENTER>

Rückgabe: nichts.

Im Default-Zustand werden im Terminal-Modus nur die von der RS232C-Schnittstelle empfangenen Zeichen auf dem Bildschirm dargestellt. Gibt man aber vor dem Aufruf von TERMINAL den Befehl HALFDUPLEX ein, so wird eine lokale Verbindung zwischen Tastatur und Bildschirm hergestellt. Das heißt, daß zu den von der RS232C-Schnittstelle empfangenen Zeichen auch noch die auf der Tastatur eingetippten Zeichen auf dem Bildschirm dargestellt werden.

Beispiel:

*** lokale Verbindung im Terminal-Mode herstellen ***

```
Ready
/halfduplex <ENTER>
Ready
■
```

Erklärung: Starten Sie nun mit !TERMINAL den Terminal-Modus, so werden die an der RS232C-Schnittstelle empfangenen Zeichen und zusätzlich die Zeichen, die Sie über Tastatur eingeben am Bildschirm dargestellt.

verwandte Befehle:

CTRLACTION, CTRLDISPLAY, FULLDUPLEX, TERMINAL

INBLOCK - eine Zeichenkette von der RS232C-Schnittstelle einlesen

Aufruf: *!INBLOCK, @Statusvariable, @Eingabestring <ENTER>*

Rückgabe: in der Statusvariablen steht:

- 0 - Die Zeichenkette wurde komplett empfangen
- 0*256+n - n Zeichen inklusiv Blockende-Kennung wurden empfangen.
- 1*256 - Fehler: mehr als zwei Parameter übergeben
- 2*256+n - Timeout erfolgte beim (n+1)ten Zeichen und n Zeichen wurden erfolgreich eingelesen.
- 3*256+n - als n-tes Zeichen wurde eine Break-Sequenz empfangen.
- 4*256+n - das n-te Zeichen wurde mit Framing Error gelesen (=Fehler bei Stopbit(s)).
- 5*256+n - das n-te Zeichen wurde mit Overrun Error gelesen (Überlauf).
- 6*256+n - das n-te Zeichen wurde mit Parity Error gelesen (Fehler bei Paritätsprüfung).

Trat mehr als ein Hardware-Fehler auf, so wird der Fehler mit der niedrigsten Fehlernummer zurückgemeldet, d.h. Break vor Framing-, Overrun- und Parity-Error.

im Eingabestring steht:

- bei Status 0*256: die gelesene Zeichenkette.
- bei Status 1*256: der alte Inhalt.
- ansonsten: die unvollständig gelesene Zeichenkette.

Mit diesem Befehl kann man ganze Zeichenketten von der RS232C-Schnittstelle in eine zuvor definierte Stringvariable einlesen. Das Einlesen von der Schnittstelle wird beendet, wenn ein Timeout erfolgte, wenn die Blockende-Kennung (siehe SETBLOCKEND) eingelesen wurde oder wenn der <Eingabestring> gefüllt ist.

Beispiel:

*** eine 10-stellige Zeichenkette einlesen ***

```
Ready
defint s <ENTER>
s=0 <ENTER>
a$="xxxxxxxxxx" <ENTER>
/inblock,@s,@a$ <ENTER>
Ready
■
```

Erklärung: Im Normalfall (d.h. wenn kein Fehler auftrat und Fehlercode 0 übergeben wurde) stehen nun in der Stringvariablen A\$ die letzten zehn von der RS232C-Schnittstelle eingelesenen Zeichen.

verwandte Befehle:

OUTBLOCK, SETBLOCKEND

INCHAR - ein Zeichen von der RS232C-Schnittstelle einlesen

Aufruf: */INCHAR, @Statusvariable, @Eingabevariable <ENTER>*

Rückgabe: in der Statusvariablen steht:

- 0*256 - Ein Zeichen wurde ohne Fehler empfangen
- 1*256 - Fehler: mehr als zwei Parameter übergeben
- 2*256 - Timeout erfolgte, es wurde kein Zeichen gelesen
- 3*256 - eine Break-Sequenz wurde empfangen
- 4*256 - Zeichen wurde mit Framing Error gelesen
- 5*256 - Zeichen wurde mit Overrun Error gelesen
- 6*256 - Zeichen wurde mit Parity Error gelesen

Trat mehr als ein Hardware-Fehler auf, so wird der Fehler mit der niedrigsten Fehlernummer zurückgemeldet, d.h. Break vor Framing-, Overrun- und Parity-Error.

in der Eingabevariablen steht:

- bei Status 0*256: das gelesene Zeichen
- bei Status 1*256 und 2*256: nichts.
- ansonsten: das fehlerhaft gelesene Zeichen.

Mit dem Befehl INCHAR wird versucht, ein Zeichen von der RS232C-Schnittstelle in die <Eingabevariable> zu lesen. Erfolgte vor dem möglichen Lesen eines Zeichen der Timeout, so wird der Versuch mit einer entsprechenden Statusmeldung abgebrochen.

Beispiel:

*** ein Zeichen in die Variable Z einlesen ***

```
Ready
defint s,z <ENTER>
s=0 <ENTER>
z=0 <ENTER>
/inchar,@s,@z <ENTER>
Ready
■
```

Erklärung: Wenn als Fehlercode der Wert 0 rückgemeldet wurde (d.h. kein Fehler auftrat), steht in der Integervariablen Z der ASCII-Wert des von der RS232C-Schnittstelle empfangenen Zeichens.

verwandte Befehle:

OUTCHAR

INFILE - empfangene Daten in eine Datei wegschreiben

Aufruf: /INFILE, @Dateiname <ENTER>

Rückgabe: nichts.

Alle an der RS232C-Schnittstelle anstehenden Daten werden in die Datei weggeschrieben, deren Name in der Stringvariablen <Dateiname> übergeben wird. Dies wird erst beendet, wenn von der RS232C-Schnittstelle die Dateiende-Kennung (siehe SETENDFILE) eingelesen wird.

Beispiel:

*** eine Datei mit dem Namen TEST.ASC einlesen ***

```
Ready
a$="test.asc"
/infile,@a$ <ENTER>
Ready
■
```

Erklärung: gesetzt den Fall, mittels SETENDFILE wurde als Dateiende-Kennung der Wert &1A eingestellt (was auch der Defaultwert ist), so werden nach Eingabe der beiden im Beispiel abgedruckten Befehle alle von der Schnittstelle empfangenen Zeichen in die Datei TEST.ASC weggeschrieben und der Vorgang wird erst beendet, wenn der Wert &1A empfangen wurde.

verwandte Befehle:

OUTFILE, SETENDFILE

INITIO - Neuinitialisierung der RS232C-Schnittstelle

Aufruf: *INITIO* <ENTER>

Rückgabe: nichts.

Sämtliche zum Betrieb der vortex-RS-Schnittstelle intern relevanten Daten sind im vom Rom reservierten Rambereich abgelegt. Ändern Sie beispielsweise eine Baudrate, so wird diese zuerst in entsprechenden Speicherzellen abgelegt und danach initialisiert eine Maschinensprach-Routine die Schnittstelle entsprechend dem Inhalt dieses Rambereiches. Welche Daten in diesem Bereich abgelegt werden, können Sie dem Anhang 1 entnehmen.

Versierte Profis haben jetzt schon die Möglichkeit erkannt, durch direktes Einschreiben in diesen Rambereich selber etwas an der Einstellung der Schnittstelle "manipulieren" zu können.

An solche Profis unter den Schnittstellen-Programmierern dachten wir bei der Implementation des RSX-Befehl INITIO. Er entspricht der oben erwähnten Routine, welche die Schnittstelle aus dem Inhalt des Rambereiches initialisiert.

Beispiel:

*** die SIO neu initialisieren ***

```
Ready
/initio <ENTER>
Ready
```

■

verwandte Befehle:

SETSIO

NOXON - Ausschalten des XON/XOFF-Software-Protokolles

Aufruf: *INOXON* <ENTER>

Rückgabe: nichts.

NOXON schaltet das XON/XOFF-Software-Protokoll (siehe RSX-Befehl SERIAL) ab.

Beispiel:

*** Softwareprotokoll für Kanal 8 abschalten ***

```
Ready  
/inoxon <ENTER>  
Ready
```

■

verwandte Befehle:

PARALLEL, SERIAL, XON

OUTBLOCK - eine Zeichenkette an die RS232C-Schnittstelle absenden

Aufruf: *OUTBLOCK, @Statusvariable, @Ausgabestring* <ENTER>

Rückgabe: in der Statusvariablen steht:

- 0*256 - Die Zeichenkette wurde ohne Fehler gesendet
- 1*256 - Fehler: mehr als zwei Parameter übergeben
oder der übergebene String hat die Länge 0.
- 2*256+n - Timeout erfolgte und n Zeichen hätten noch
abgesandt werden müssen.

Mit diesem Befehl kann man ganze Zeichenketten an die RS232C-Schnittstelle absenden, die zuvor in eine Stringvariable geschrieben wurden. Die Übertragung des Stringinhaltes wird beendet, wenn dieser komplett gesendet wurde oder wenn die Timeout-Zeit überschritten wurde. Eine weitere Möglichkeit des Abbruchs ist der Versuch, die Blockende-Kennung als Teil des Strings zu senden.

Beispiel:

*** den Inhalt von A\$ absenden ***

```
Ready
defint s <ENTER>
s=0 <ENTER>
a$="Dies ist ein Test"
!outblock,@s,@a$ <ENTER>
Ready
```

verwandte Befehle:

INBLOCK, SETBLOCKEND

OUTCHAR - ein Zeichen an die RS232C-Schnittstelle senden

Aufruf: !OUTCHAR, @Statusvariable, Ausgabewert <ENTER>

Rückgabe: in der Statusvariablen steht:

- 0*256 - das Zeichen wurde ohne Fehler gesendet
- 1*256 - Fehler: mehr als zwei Parameter übergeben
- 2*256 - Timeout erfolgte und nichts wurde gesendet.

Es wird versucht, das dem <Ausgabewert> entsprechende Zeichen an die RS232C-Schnittstelle abzusenden. Konnte das Zeichen bis zum Timeout nicht gesendet werden, so wird dies entsprechend in der Statusvariablen rückgemeldet.

Beispiel:

*** das Zeichen "A" absenden ***

```
Ready
defint s <ENTER>
s=0 <ENTER>
!outchar,@s,65 <ENTER>
Ready
```

verwandte Befehle:

INCHAR

OUTFILE - den Inhalt einer ASCII-Datei an die RS232C senden

Aufruf: *!OUTFILE, @Dateiname <ENTER>*

Rückgabe: nichts.

Der Inhalt der Datei, deren Name in der Stringvariablen <Dateiname> übergeben wurde, wird an die RS232C-Schnittstelle abgesendet. Der Befehl OUTFILE unterscheidet sich hierbei wesentlich von dem Befehl BLOW, da keinerlei Protokoll verlangt wird. Das heißt, daß der Inhalt einfach abgesendet wird, egal was für ein Peripheriegerät an die RS-Schnittstelle angeschlossen ist. So kann man beispielsweise eine Datei sehr einfach drucken, wenn ein serieller Drucker an der Schnittstelle angeschlossen ist.

Beispiel:

*** die ASCII-Datei TEST.ASC absenden ***

```
Ready
a$="test.asc"<ENTER>
!outfile,@a$ <ENTER>
Datei uebertragen
Ready
■
```

verwandte Befehle:

OUTFILE, SETENDFILE

PARALLEL - Umschalten der Druckerausgabe auf die parallele Schnittstelle

Aufruf: *!PARALLEL <ENTER>*

Rückgabe: nichts.

Wenn der List-Kanal #8 des Basic mit dem Befehl SERIAL auf die RS232C-Schnittstelle umgeleitet wurde, macht der Befehl PARALLEL dies wieder rückgängig. Alle LIST#8- und PRINT#8-Ausgaben unter Basic erfolgen von nun an wieder über die eingebaute Centronics-

Schnittstelle.

Beispiel:

*** Kanal 8 - Ausgabe auf die Centronics-Schnittstelle ***

```
Ready  
/parallel <ENTER>  
Ready  
■
```

verwandte Befehle:

NOXON, SERIAL, XON

RINGWAIT - warten auf -Ring detect-

Aufruf: */RINGWAIT, @Statusvariable, Timeout-Zeit <ENTER>*

Rückgabe: in der Statusvariablen steht 0, wenn <Ring detect> erkannt wurde und 2*256, wenn ein Timeout erfolgte.

Nach Aufruf des Befehls RINGWAIT wird solange gewartet, bis an der RS232C-Schnittstelle der Pin 22 (Ring Indicator) von einem Peripheriegerät gesetzt wird oder wenn ein Timeout erfolgte. Die übergebene Timeout-Zeit unterscheidet sich von der Timeout-Zeit des Befehls BREAKSEND oder der durch den Befehl SETTIMEOUT definierten Timeout-Zeit. Die Zeiteinheit dieses Zählers ist eine Sekunde.

Beispiel:

*** 10 Sekunden auf <ring indicator> warten ***

```
Ready  
defint s <ENTER>  
s=0 <ENTER>  
/ringwait,@s,10 <ENTER>  
Ready  
■
```

verwandte Befehle:

BREAKSEND, CLOSESIO, SIO

ROMCAT - Ausgabe der ROMs und deren RSX-Befehle

Aufruf: *!ROMCAT* (*, Romnummer*) <ENTER>

Rückgabe: entsprechende Bildschirmausgabe.

Der Befehl ROMCAT hat zwei verschiedene Funktionen:

1. Ruft man ihn ohne Übergabe eines Parameters auf, so erhält man auf dem Bildschirm eine Liste aller Background-Roms angezeigt. Diese kann z.B so aussehen:

Ready

!romcat <ENTER>

Hinen: 9BDD

<i>Nr.</i>	<i>Ty</i>	<i>Ma</i>	<i>Ve</i>	<i>Mo</i>	<i>Base</i>	<i>Name</i>
<i>00</i>	<i>80</i>	<i>01</i>	<i>00</i>	<i>00</i>	<i>----</i>	<i>BASIC</i>
<i>01</i>	<i>80</i>	<i>01</i>	<i>00</i>	<i>00</i>	<i>----</i>	<i>BASIC</i>
<i>02</i>	<i>80</i>	<i>01</i>	<i>00</i>	<i>00</i>	<i>----</i>	<i>BASIC</i>
<i>03</i>	<i>80</i>	<i>01</i>	<i>00</i>	<i>00</i>	<i>----</i>	<i>BASIC</i>
<i>04</i>	<i>80</i>	<i>01</i>	<i>00</i>	<i>00</i>	<i>----</i>	<i>BASIC</i>
<i>05</i>	<i>01</i>	<i>00</i>	<i>01</i>	<i>00</i>	<i>9C62</i>	<i>M WLK</i>
<i>06</i>	<i>01</i>	<i>00</i>	<i>00</i>	<i>02</i>	<i>9C66</i>	<i>S I O</i>
<i>07</i>	<i>01</i>	<i>00</i>	<i>04</i>	<i>00</i>	<i>A50C</i>	<i>V ROM</i>

Ready

■

In der ersten Spalte wird die Nummer des Roms angezeigt, wobei beim CPC 464 die maximale Nummer 7 und beim 664 sowie 6128 die größte Nummer 15 ist. Die zweite bis fünfte Spalte gibt verschiedene vom Programmierer des jeweiligen Roms am Anfang des Roms eingetragene Statusinformation zur Versionsnummer des Roms. Sofern es sich bei dem jeweiligen Rom um einen Background-Rom handelt, wird in der sechsten Spalte dessen Ram-Basisadresse (siehe BASE-Befehl) ausgegeben. In der siebten und letzten Spalte wird dann der Rom-Name ausgegeben (gleichbedeutend mit dem Namen der Romwalk-Routine des Roms). Ist ein Rom nicht vorhanden, wird automatisch die gleiche Angabe wie beim Basic-Rom gemacht.

2. Wenn Sie beim Aufruf des ROMCAT-Befehls mittels eines Parameters eine Rom-Nummer übergeben, erhalten Sie dessen RSX-Befehle und die jeweilige Startadresse der Routine. Bitte beachten Sie dabei, daß von dem Befehl am CPC 464 als maximale Rom-Nummer die Sieben und bei den beiden anderen CPCs die Fünfzehn akzeptiert wird. Eine Liste sähe dann beispielsweise so aus:

Ready
/romcat,6 <ENTER>

Adr. Name

C006 S I O
C221 CTRLDISPLAY
DS10 SIO
BD19 FRAME
etc.

Ready

■

verwandte Befehle:

BASE, ROMCAT

ROMOFF - Ausschalten von Background-Roms

Aufruf: /ROMOFF (,Parameterliste) <ENTER>

Rückgabe: nichts.

Manche Programme benötigen eine bestimmte Rom-Konfiguration. Viele auf Cassetten vertriebene Spielprogramme wollen am CPC 464 sogar den Originalzustand, damit man sie überhaupt laufen lassen kann. Deshalb implementierten wir den Befehl ROMOFF. Mit ihm können Sie alle oder selektiv einen, bzw. mehrere Roms abschalten.

Geben Sie bei Aufruf des Befehls keinen Parameter an, so werden alle Roms abgeschaltet, geben Sie allerdings Parameter an, so werden die den Parametern entsprechenden Roms ab- und alle anderen vorhandene eingeschaltet. Bitte beachten Sie dabei, daß die Background-Roms am CPC 464 die Nummern eins bis sieben und an den beiden anderen CPC-Typen die Nummern 1 bis fünfzehn belegen können. Wenn Sie als Parameter Zahlen außerhalb des entsprechenden Wertebereiches übergeben, werden diese ignoriert. Beispiel:

Beispiel:

*** die Background-Roms mit den Nummern 6 und 7 ausschalten ***

Ready
/romoff,6,7 <ENTER>
Ready

■

Erklärung: Jetzt sind die Roms mit den Nummern sechs und sieben nicht mehr verfügbar, und normalerweise haben Sie damit die Speichererweiterung und das Diskettenbetriebssystem abgeschaltet (sofern Sie die beiden Peripheriegeräte besitzen!).

verwandte Befehle:

BASE, ROMCAT

SETBLOCKEND - Definition der Blockende-Kennung

Aufruf: *!SETBLOCKEND, Blockende-Kennung <ENTER>*

Rückgabe: nichts.

Normalerweise wird bei Blocktransfer-Kommandos (INBLOCK, OUTBLOCK) die ganze Zeichenkette an die Schnittstelle gesendet oder empfangen. Will man allerdings beim Erkennen eines bestimmten Wertes den Blocktransfer beenden, so kann man mit diesem Befehl ein ASCII-Zeichen zwischen 0 und 255 als Blockende-Kennung definieren. Hat man z.B. 65 als Blockende-Kennung definiert, so bricht der Blocktransfer jedesmal beim Versuch ein "A" zu senden ab. Den Default-Zustand (keine Blockende-Kennung) erreicht man durch Übergabe eines Wertes der größer als 255 ist.

Beispiel:

*** 27 als Blockende-Kennung anmelden ***

Ready
!setblockend,27 <ENTER>
Ready

■

verwandte Befehle:

INBLOCK, OUTBLOCK

SETENDFILE - Definition der Dateiende-Kennung

Aufruf: *!SETENDFILE, Dateiende-Kennung <ENTER>*

Rückgabe: nichts.

Durch SETENDFILE wird ein ASCII-Wert als Dateiende-Kennung für die Befehle INFILE und OUTFILE definiert. Der Empfang von Daten mittels INFILE wird beendet, wenn diese Dateiende-Kennung auftritt. Ist mit OUTFILE eine Datei komplett gesendet worden, so wird automatisch noch die Dateiende-Kennung gesendet. Der übergebene Wert wird MOD 256 zur Definition der Dateiende-Kennung genommen (d.h. nur die unteren 8 Bit sind relevant). Der Defaultwert der Dateiende-Kennung ist dezimal 27 (CTRL Z).

Beispiel:

*** 65 (das Zeichen A) als Dateiende-Kennung anmelden ***

Ready

!setendfile,65 <ENTER>

Ready

■

Erklärung: Wird mit INFILE jetzt eine Datei mit den von der RS-Schnittstelle empfangenen Daten gefüllt, so wird die Operation abgebrochen und die Datei geschlossen, sobald ein "A" von der RS-Schnittstelle empfangen wurde.

verwandte Befehle:

INFILE, OUTFILE

SETSIO - Einstellen der RS232C-Parameter

Aufruf: *!SETSIO, Baudrate (, Empfangs-Baudrate (, Hardware-Flowcontrol (, Data-Bits (, Parität (, Stop-Bits)))) <ENTER>*

Rückgabe: nichts.

Mit dem SETSIO-Befehl können die fundamentalen Parameter der RS232C-Schnittstelle verändert werden. Die Default-Einstellung

dieser Parameter nach Einschalten der Schnittstelle sieht wie folgt aus:

9600 Baud Sende-Baudrate
9600 Baud Empfangs-Baudrate
Hardware-Flowcontrol eingeschaltet
8 Data-Bits
keine Paritätsprüfung
1 Stop-Bit.

Alle Parameter des Befehls bis auf den ersten sind optional. Hat ein Parameter einen unzulässigen Wert, so wird er ignoriert. Die zulässigen Werte mit denen die Baudraten der Schnittstelle eingestellt werden können sind: 19200, 9600, 4800, 3600, 2400, 2000, 1800, 1200, 600, 300, 200, 150, 110, 75 und 50. Wird als Baudrate ein Wert gewählt, der nicht in dieser Tabelle enthalten ist, so meldet das Betriebssystem *Unzulässige Baudrate!* und ändert nichts an der Einstellung der Schnittstelle.

Der erste Parameter, die Baudrate ist ein Wert aus der Tabelle der zulässigen Baudraten. Gibt es keinen weiteren Parameter, so spezifiziert dieser Wert die Sende- und Empfangs-Baudrate, ansonsten nur die Sende-Baudrate. Dies gilt nur für den Kanal A, da beim Kanal B der Schnittstelle die Sende- und Empfangs-Baudrate immer gleich ist. Deshalb spezifiziert der möglicherweise übergebene zweite Parameter nur bei Kanal A die Empfangsbaudrate und wird einfach ignoriert, wenn der Kanal B aktiv ist. Der Wert für diese zweite Baudrate darf ebenfalls eine der zulässigen Baudraten aus obiger Tabelle sein.

Hardware-Flowcontrol wird mit dem dritten Parameter bestimmt. Der Integer-Wert gibt an, ob Hardware-Flowcontrol ein- oder ausgeschaltet ist, wobei 0 für ausgeschaltet und ein Wert ungleich 0 für eingeschaltet steht.

Die Anzahl der Data-Bits in der Parameterliste ist ein Integer-Wert zwischen 5 und 8 und gilt für Empfangen und Senden.

Der fünfte Parameter ist ein Wert zwischen 0 und 2. Dabei steht 0 für keine Paritätsprüfung (NO Parity), 1 für ungerade Parität (ODD Parity) und 2 für gerade Parität (EVEN Parity).

Wie die Parität ist auch der letzte Parameter eine Integer-Zahl zwischen 0 und zwei und gibt die Anzahl der Stop-Bits an. Dabei steht 0 für ein Stop-Bit, 1 für eineinhalb Stop-Bits und 2 für zwei Stop-Bits.

Beispiel:

*** Baudraten auf 4800 (senden) und 1200 (empfangen) setzen ***

```
Ready  
/setsio,4800,1200 <ENTER>  
Ready
```

■

Erklärung: Wenn derzeit der Kanal A der Schnittstelle aktiv ist, werden die Sende- und Empfangsbaudrate entsprechend geändert. Sollte Kanal B aktiv sein, so werden die Sende- und Empfangsbaudrate beide auf den Wert 4800 gesetzt.

verwandte Befehle:

INITTSIO, SIO

SETTIMEOUT - definieren der Timeout-Zeit

Aufruf: *!SETTIMEOUT, Timeout-Zeit <ENTER>*

Rückgabe: nichts.

Übertragen von Daten an eine Schnittstelle kann auch zu einem "Aufhängen" des Systems führen, wenn die entsprechende Routine dauernd versucht einen Wert an die Schnittstelle zu senden, das angeschlossene Peripheriegerät allerdings nicht empfangsbereit ist. Daher gibt es bei allen Break-, Zeichen-, und Block-Kommandos eine Timeout-Zeit zu definieren. Dies ist die Zeit, nach der die Sende- oder Empfangs-Routine mit einem Fehlercode zurückkehrt, wenn sie den Wert nicht absenden konnte. Die Timeout-Zeit darf im Wertebereich zwischen 1 und 65535 liegen, wobei die Zeit in Millisekunden angegeben wird.

Beispiel:

*** 10 Sekunden Timeout-Zeit anmelden ***

Ready
!settimeout,10000 <ENTER>
Ready
■

Erklärung: Wird jetzt zum Beispiel mit OUTCHAR versucht, ein Zeichen an die Schnittstelle abzusenden und dies gelang dem Betriebssystem nicht innerhalb von zehn Sekunden, so kehrt es mit ins Basic zurück. Es wird dann in der Statusvariablen ein entsprechender Fehlercode zurückgemeldet.

verwandte Befehle:

SERIAL - Umschalten der Druckerausgabe auf die serielle Schnittstelle

Aufruf: */SERIAL* <ENTER>

Rückgabe: nichts.

Nach Eingabe des Befehls SERIAL wird die komplette Druckerausgabe unter Basic auf die serielle Schnittstelle umgeleitet, d.h. wenn ein Drucker mit serieller Schnittstelle an die vortex-RS232C-Schnittstelle angeschlossen ist, bewirken die Basic-Befehle PRINT#8 und LIST#8 einen Ausdruck auf diesem Drucker.

Im Default-Zustand arbeitet die RS232C-Schnittstelle dann mit einem sogenannten XON/XOFF-Software-Protokoll. Das bedeutet, daß ein an die RS232C-Schnittstelle angeschlossenes Peripheriegerät ein XOFF-Zeichen rückmelden kann, wenn es nicht mehr empfangsbereit ist. Die RS232C-Schnittstelle wartet dann solange, bis das Peripheriegerät ein XON-Zeichen sendet und damit wieder Empfangsbereitschaft signalisiert. Sollte das Peripheriegerät aus irgendwelchen Gründen kein XON mehr rückmelden, so können Sie die Warteschleife der RS232C-Schnittstelle durch Drücken der ESC-Taste beenden.

Die Werte für XON und XOFF sind:

XON: CTRL Q oder 17 dezimal

XOFF: CTRL S oder 19 dezimal.

Beispiel:

*** Kanal 8 - Ausgabe auf die RS232C-Schnittstelle legen ***

Ready

/serial <ENTER>

Ready

■

verwandte Befehle:

NOXON, PARALLEL, XON

SIO - Status der RS232C-Schnittstelle ermitteln

Aufruf: *!SIO, @Statusvariable <ENTER>*

Rückgabe: in der Statusvariablen steht bitsignifikant:

- Bit 15 - 0
- Bit 14 - Framing error occurred
Stop-Bit(s) irregulär empfangen.
- Bit 13 - Receive overrun error occurred
mehr als 3 Zeichen wurden empfangen, ohne das
ein Zeichen von der SIO abgeholt wurde.
- Bit 12 - Parity error occurred
bei eingeschalteter Paritätsprüfung wird mit
diesem Bit ein aufgetretener Fehler angezeigt.
- Bit 11 - 0
- Bit 10 - 0
- Bit 9 - 0
- Bit 8 - All sent
gesetzt, wenn alles komplett gesendet ist.
- Bit 7 - Break received
eine Unterbrechungssequenz (Null-Charakter
plus Framing Error) wurde empfangen.
- Bit 6 - 0
- Bit 5 - Clear To Send
zeigt den invertierten Zustand des CTS-Pins.
- Bit 4 - Ring detect
ein Peripheriegerät meldete sich.
- Bit 3 - Data Carrier Detect
zeigt den invertierten Zustand des DCD-Pins.
- Bit 2 - Transmitter buffer empty
Sende-Buffer ist leer.
- Bit 1 - 0
- Bit 0 - Received Data available
gesetzt, wenn ein empfangenes Zeichen von der
SIO abgeholt werden kann.

Nach Ausführung dieses Befehls werden in der Statusvariablen die Leseregister 0 und 1 der SIO zurückgeliefert, wobei das Leseregister 0 in den Bits 0 bis 7 und das Leseregister 1 in den Bits 8 bis 15 des Werts der Statusvariablen liegt.

Beispiel:

*** den SIO-Status in die Variable S einlesen ***

```
Ready
defint s <ENTER>
s=0 <ENTER>
!sio,@s <ENTER>
Ready
■
```

verwandte Befehle:

BREAKSEND, CLOSESIO, RINGWAIT

STATUS - Basic-Programmstatus ausgeben

Aufruf: *!STATUS* <ENTER>

Rückgabe: entsprechende Bildschirmausgabe.

Für den Basic-Programmierer ein sicherlich interessanter Befehl: mit STATUS erhalten Sie auf dem Bildschirm Himem, Programmstart, -ende und -länge, den freien Speicherplatz und das erste benutzerdefinierte Zeichen rückgemeldet. Dies sieht dann so aus:

```
Ready
!status <ENTER>

HIMEM:           &9BDD
Programmstart:  &016F
Programmende:   &0172
Programmlaenge  3 Bytes
freier Speicher 39531 Bytes
Symbol After    240
```

Ready

■

SUCK - eine ASCII-Datei intelligent empfangen

Aufruf: *!SUCK, @Dateiname <ENTER>*

Rückgabe: nichts.

Mit Start des Befehls SUCK wird eine Datei empfangen, die von einem CPC mit dem Befehl BLOW gesendet wird. Das Übertragungsprotokoll von SUCK können Sie dem Anhang 5 entnehmen und die nähere Beschreibung der BLOW/SUCK-Übertragung finden Sie bei der Erklärung des Befehls BLOW.

Bitte beachten:

Da SUCK zuerst auf eine Meldung des Senders wartet (STX), muß bei einer Dateiübertragung mit BLOW/SUCK der Empfänger SUCK starten, bevor der Sender die Übertragung mit BLOW beginnt.

Beispiel:

*** die ASCII-Datei "TEST.ASC" mit Protokoll empfangen ***

Ready

a\$="test.asc" <ENTER>

!suck,@a\$ <ENTER>

Datei uebertragen

verwandte Befehle:

BLOW

TERMINAL - Ein-/Ausgabeeinheit über RS232C-Schnittstelle herstellen

Aufruf: *!TERMINAL (,baudrate) <ENTER>*

Rückgabe: nichts.

Mit TERMINAL leitet man die Tastatureingabe auf die RS232C-Schnittstelle, um und die von der RS232C-Schnittstelle empfangenen Zeichen werden auf dem Bildschirm dargestellt. Dieser Modus wird durch Drücken der ESC-Taste verlassen. Dabei ist zu

beachten, daß die ESC-Taste nicht durch den Basic-Befehl KEY DEF undefiniert wurde. Sollte dies der Fall sein, so müssen Sie eine Taste mit dem zugewiesenen ASCII-Wert &FC drücken. Übergeben Sie gleichzeitig mit dem Aufruf von TERMINAL eine der zulässigen Baudraten, so wird die Sende- und Empfangsbaudrate entsprechend geändert.

Prinzipiell haben Sie mit diesem RSX-Befehl die Möglichkeit, mit einer Mailbox in Kontakt zu treten. Die Schnittstellenparameter müssen allerdings zuvor alle richtig eingestellt sein. Wollen Sie in größerem Ausmaß die Datenfernübertragung betreiben, so empfiehlt sich ein DFÜ-Programm. Als Beispiel sei hierfür das CP/M-Programm KERMIT genannt.

Beispiel:

*** den Terminal-Modus aufrufen ***

```
Ready  
/terminal <ENTER>
```

■

verwandte Befehle:

CTRLACTION, CTRLDISPLAY, FULLDUPLEX, HALFDUPLEX

TRANSFER.COM - Generieren des CP/M-Uebertragungsprogrammes

Aufruf: /TRANSFER.COM <ENTER>

Rückgabe: nichts.

Um unter CP/M Programme mit einem Software-Protokoll zu übertragen, müßten wir eigentlich ein Programm auf Diskette mit dieser RS-Schnittstelle ausliefern oder Ihnen zumuten, ca. 600 Bytes manuell einzugeben. Wir dachten uns allerdings, daß es auch etwas einfacher geht: Legen Sie einfach eine CP/M-Diskette in das derzeit aktive Laufwerk, ein und starten Sie den Befehl TRANSFER.COM . Wenn das Ready des Basic wieder erscheint, befindet sich auf dieser Diskette ein Programm namens TRANSFER.COM . Unter CP/M können Sie nun dieses Programm mit der Syntax *TRANSFER filename.typ* starten und die als *filename.typ* spezifizierte Datei gemäß dem unter BLOW und SUCK erklärten Software-Protokoll übertragen.

Beispiel:

*** TRANSFER.COM generieren und WS.COM übertragen ***

```
Ready
/a <ENTER>
Ready
/transfer.com <ENTER>
Ready
/cpm <ENTER>
A>transfer ws.com <ENTER>
(B)low oder (S)uck ? b
```

Erklärung: Zuerst wird das Laufwerk A angewählt, in dem sich eine nicht schreibgeschützte CP/M-Diskette mit genügend freiem Speicherplatz befinden sollte. Nach Eingabe des RSX-Befehls TRANSFER.COM befindet sich das CP/M-Programm TRANSFER.COM auf der Diskette, das Sie nach Booten des CP/M-Betriebssystems starten können. Das Programm fragt Sie dann, ob Sie die als Parameter angegebene Datei senden wollen (B), oder ob Sie eine Datei diesen Namens empfangen wollen (S).

verwandte Befehle:

BLOW, SUCK

VERSION - Versions-Nummer und -Datum ausgeben

Aufruf: /VERSION <ENTER>

Rückgabe: entsprechende Bildschirmausgabe.

VERSION liefert Ihnen auf dem Bildschirm die Versions-Nummer des RS-Roms und das Datum der letzten Änderung dieses Roms. Das sieht dann so aus:

```
Ready
/version <ENTER>
```

```
vortex RS232C Version 1.0
letzte Aenderung: 18.08.86
```

```
Ready
```

■

XON - Einschalten des XON/XOFF-Software-Protokolles

Aufruf: */XON* <ENTER>

Rückgabe: nichts.

Mit diesem Befehl schalten Sie das XON/XOFF-Softwareprotokoll (siehe RSX-Befehl SERIAL) ein, d.h. er macht nur dann einen Sinn, wenn mit NOXON dieses Softwareprotokoll zuvor ausgeschaltet wurde.

Beispiel:

*** Kanal 8 - Ausgabe mit Software-Protokoll ***

Ready
/xon <ENTER>
Ready
■

verwandte Befehle:

NOXON, PARALLEL, SERIAL

ANHANG 1 - Die RAM-Belegung

Wie jeder andere ins Betriebssystem eines CPC-Rechners integrierte ROM, benötigt der ROM der RS232C-Schnittstelle von vortex ebenfalls eine gewisse Anzahl von Bytes im RAM des CPCs. Dieser Rambereich wird automatisch beim Einschalten des Rechners reserviert und die Startadresse dieses Bereiches ist dynamisch, da ja eine unterschiedliche Anzahl von ROMs in das System mit eingebunden sein kann. Damit Sie die Startadresse trotz allem erfahren können, haben wir den Befehl BASE (siehe oben) implementiert.

Die folgende Belegungstabelle des Rambereiches des RS232C-ROMs versteht sich immer als Offset zu der Startadresse des Rambereiches, die Sie wie erwähnt mit BASE ermitteln können.

Offset		Defaultwert	Bedeutung
hexa	dezi		
00	00	&44	Write-Register 4 Kanal A
01	01	&E1	Write-Register 3 Kanal A
02	02	&6A	Write-Register 5 Kanal A
03	03	&44	Write-Register 4 Kanal B
04	04	&E1	Write-Register 3 Kanal B
05	05	&6A	Write-Register 5 Kanal B
06	06	&000D	Counter 8253 Kanal A transmit
08	08	&000D	Counter 8253 Kanal A receive
0A	10	&000D	Counter 8253 Kanal B trans/receive
0C	12	0	Software-Flagbyte mit folgender Bedeutung der einzelnen Bits:
		Bit 0:	derz. aktiver SIO-Kanal (A/B)
		Bit 1:	Timeout aktiv/nicht aktiv
		Bit 2:	Terminal-Modus full-/halfduplex
		Bit 3:	Blockende-Kennung aktiv/nicht aktiv
		Bit 4:	XON/XOFF aktiv/nicht aktiv
0D	13	0	Timeout-Wert
0F	15	&C3xxxx	Sprung auf Terminal-Textausgabe
12	18	0	Blockende-Kennung
13	19	27	Dateiende-Kennung
14	20	&xxxxx	Adresse der seriellen Druckerausgabe im ROM (Far-Call-Adresse)
16	22	0	ROM-Nummer des RS232C-ROMs
17	23	&xxxxxxx	Platz für die Sicherung der Drucker-Indirection bei Aufruf von SERIAL
1A	26	&DF	RST 3 (Far Call) für die
1B	27	&xxxxx	serielle Druckeroutine
1D	29	&C9	und der Rücksprung des Far Calls.
1E	30	&xxxxx	Adresse des Dateinamen bei BLOW
20	32	&xxxxx	letzte Blocknummer bei BLOW/SUCK
2048	Bytes		Puffer für die Dateioptionen
149	Bytes		Puffer für BLOW-/SUCK-Blöcke

ANHANG 2 - Die Hardware

Die Hardware der vortex RS232C-Schnittstelle besteht aus folgenden Komponenten:

- 1 Z80-DART
- 1 8253 Timer
- 1 Eprom (128 oder 256)
- 1 ICL 7660
- diverse Logikbauteile
- diverse passive Bauteile

Das Herz der Schnittstelle ist der Z80-DART, der von der Funktion eine Untermenge der normalen Z80-SIO ist. Er organisiert nach entsprechender Einstellung der internen Register den kompletten Datenverkehr, er wandelt parallele Daten in serielle Daten (Senden) und umgekehrt (Empfangen), fügt Stop- und Paritätsbits an etc. Das heißt also, daß nach entsprechender Grundeinstellung nur noch ein Datum übergeben werden muß und das Senden besorgt der DART, bzw. man kann sich ein empfangenes Datum einfach vom DART holen, wenn dieser den Empfang eines Datums gemeldet hat.

Die Register unterteilen sich in zwei Gruppen: den Write-Registern WRO bis WR7, die nur beschrieben werden können, und den Read-Registern RRO bis RR2, welche nur gelesen werden können. Die Funktion der einzelnen Register ist wie folgt zu erklären:

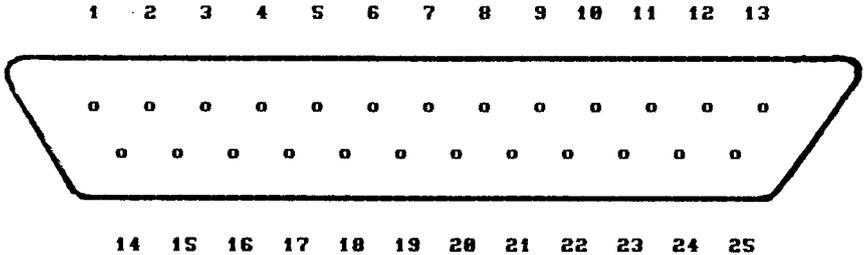
- WRO - Zeiger auf die anderen Register, die nur indirekt beschrieben werden können und diverse Sonderfunktionen (Error-Reset, etc.).
- WR1 - Interrupt- und Übertragungsmodus-Definitionen.
- WR2 - Interrupt-Vektor (nur Kanal B).
- WR3 - Empfangsparameter und andere Controlbits.
- WR4 - diverse Empfangs- und Sendeparameter.
- WR5 - Sendeparameter und andere Controlbits.
- WR6 - für SDLC-Mode.
- WR7 - für SDLC-Mode.
- RRO - Sende-/Empfangsstatus, Interrupt- und externer Status.
- RR1 - spezieller Empfangsstatus.
- RR2 - veränderter Interrupt-Vektor (nur Kanal B).

Für den Betrieb unserer Schnittstelle sind nur die Write-Register 0, 3, 4 und 5, sowie die Read-Register 0 und 1 von Bedeutung. Eine genauere Beschreibung, die den Rahmen dieses Handbuches sprengen würde, erhalten Sie aus entsprechender Fachliteratur. Zu empfehlen sind hierzu beispielsweise die beiden Bücher Z80 DESIGNERS GUIDE von Mostek und Z80 ANWENDUNGEN aus dem Sybex-Verlag (ISBN 3-88745-037-X).

Der Timerbaustein 8253 hat nur eine Aufgabe, die allerdings zum Betrieb der Schnittstelle sehr wesentlich ist:

Er teilt den anstehenden Grundtakt von 2 MHz (=Systemtakt/2) durch vorgegebene Teiler und liefert den neuen Takt an den DART, der dann daraus wiederum eine entsprechende Baudrate generiert.

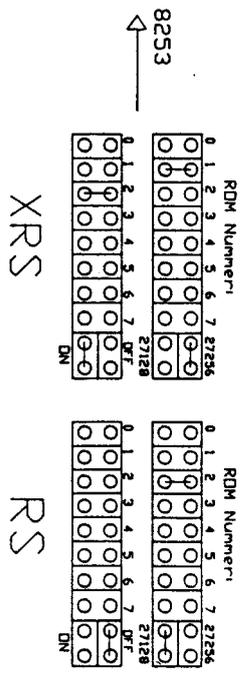
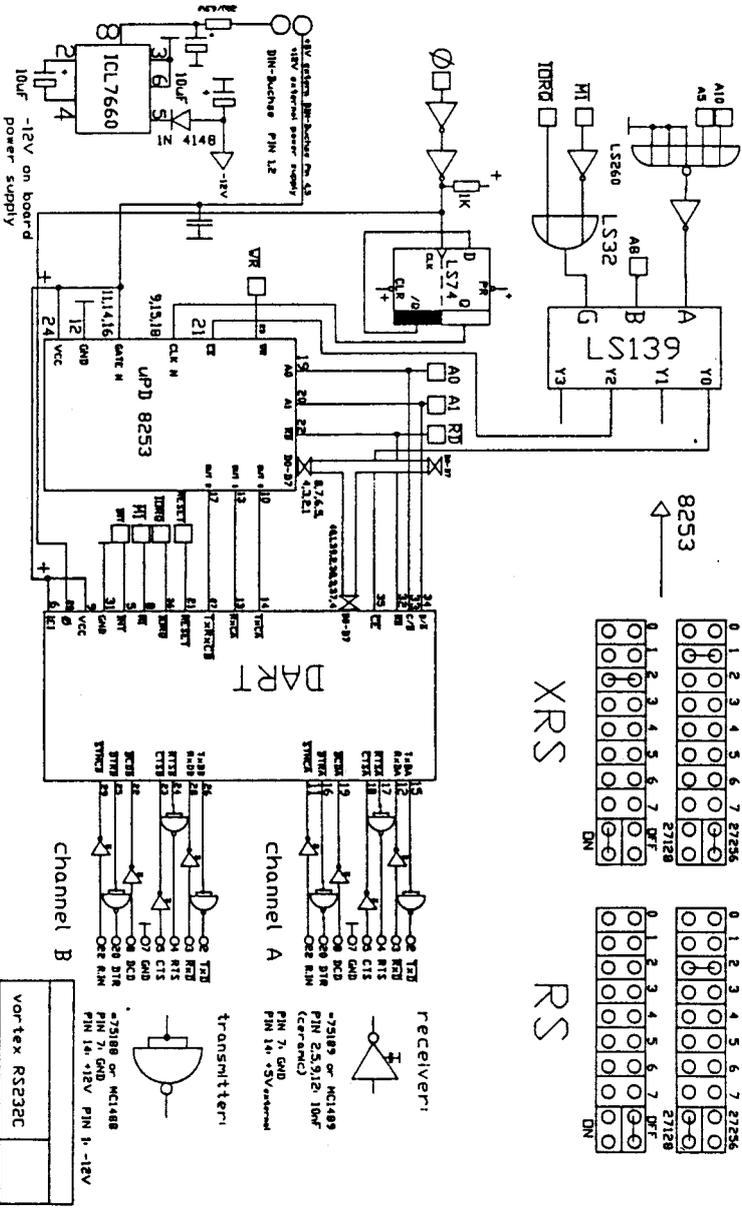
Steckeransicht an der RS232C-Schnittstelle



Pinbelegung der vortex RS232C-Schnittstelle

Pin 1 nicht benutzt	Pin 14 nicht benutzt
Pin 2 TxD (transmit)	Pin 15 nicht benutzt
Pin 3 RxD (receive)	Pin 16 nicht benutzt
Pin 4 RTS	Pin 17 nicht benutzt
Pin 5 CTS	Pin 18 nicht benutzt
Pin 6 nicht benutzt	Pin 19 nicht benutzt
Pin 7 GMD	Pin 20 DTR
Pin 8 DCD	Pin 21 nicht benutzt
Pin 9 nicht benutzt	Pin 22 Ring Indicator
Pin 10 nicht benutzt	Pin 23 nicht benutzt
Pin 11 nicht benutzt	Pin 24 nicht benutzt
Pin 12 nicht benutzt	Pin 25 nicht benutzt
Pin 13 nicht benutzt	

ANHANG 3 - Der Schaltplan



vortex RS232C

ANHANG 4 - Maschinensprache

Vielleicht interessiert Sie es, wie Sie unter Maschinensprache Daten von der Schnittstelle holen oder auf diese wegschreiben können. Deshalb folgen hier zur Verdeutlichung zwei kleine Routinen in Maschinensprache.

```

;-----
;
; Routine wartet solange, bis die SIO ausgabebereit ist
; und übergibt dann den Akkuinhalt an den Kanal A.
;
; Einsprung: zu sendender Wert im Akku.
;
; Aussprung: nichts.
;
putsio: push bc                ; wird benutzt, deshalb sichern.
        push af                ; Datum sichern.
        ld bc,OFADDH           ; DART Control Port Kanal A.
pwait:  in a,(c)                ; RRO in Akku holen.
        bit 2,a                 ; "TX-Buffer-Empty" testen.
        jr z,pwait             ; Buffer nicht leer --> warten.
        pop af                 ; gesichertes Datum holen.
        dec bc                 ; DART Datenport Kanal A in BC.
        out (c),a              ; Datum absenden.
        pop bc                 ; BC wieder herstellen
        ret                    ; und die Routine verlassen.
;-----
;
; Routine wartet bis ein Datum von der Schnittstelle zum
; Empfang angeboten wird (Kanal A), und liest dieses dann
; in den Akku.
;
; Einsprung: in (HL) muß der Wert stehen, mit dem WR5 des DART
; initialisiert wurde (man legt es bei der Ini-
; tialisierung einfach im RAM ab).
;
; Aussprung: eingelesenes Datum im Akku.
;
getsio: push bc                ; BC wird benutzt, deshalb sichern.
        ld bc,OFADDH           ; Dart Control Port Kanal A.
gwait:  in a,(c)                ; RRO des DART in den Akku holen.
        rrca                   ; RX-Flag in Carry schieben.
        jr c,get1              ; Bit war gesetzt --> senden.
        ld a,5                  ; Write-Register 5 des DART
        out (c),a              ; anwählen.
        ld a,(hl)               ; WR5-Wert aus dem Ram holen.
        set 7,a                 ; DTR-Bit des WR5 setzen
        out (c),a              ; geändertes WR5 in DART schreiben.

```

```
in    a,(c)           ; RRO nochmals lesen.
rrca                      ; RX-Flag in Akku schieben.
ld    a,5             ; WRS des DART
out   (c),a          ; anwählen.
ld    a,(hl)         ; alten WRS-Wert aus dem Ram holen
out   (c),a          ; und wieder in den DART schreiben.
jr    nc,gwait       ; RX-Flag nicht gesetzt --> warten.
get1: dec bc         ; Datenport Kanal A in BC.
in    a,(c)         ; Datum einlesen.
pop   bc             ; alten BC-Inhalt wieder herstellen
ret                      ; und die Routine verlassen.
```

ANHANG 5 - das Protokoll von BLOW und SUCK

Für Profis, die es interessiert mit welchen Übertragungsprotokoll die RSX-Befehle BLOW und SUCK, sowie das CP/M 2.2-Programm TRANSFER.COM arbeiten, folgt hier die Beschreibung dieses Protokolls.

*** DAS PROTOKOLL DES SENDERS (BLOW) ***

1. Senden von STX und warten auf ACK, ETX oder NAK. Bei Empfang von NAK wird Punkt 1 wiederholt, bei ETX wird die Übertragung abgebrochen und bei ACK wird mit Punkt zwei fortgefahren.
2. Den Filenamen (16 Bytes), eine 2 Byte Blocknummer, eine 1 Byte Blocklänge (0 bis 128 als Wert), den Datenblock (Anzahl: 0 bis 128 Bytes) und eine 2 Byte Prüfsumme über den Datenblock senden. Wenn als Blocklänge null übergeben wird, bedeutet dies, daß das Ende der zu übertragenden Datei erreicht ist.
3. Warten auf ACK, ETX oder NAK.
4. Wenn ETX empfangen wurde, wird die Übertragung abgebrochen, bei NAK wird mit Punkt 2 die Übertragung desselben Blocks wiederholt, bei ACK wird mit der Übertragung des nächsten Blocks und Punkt 2 weitergemacht, bzw. wenn der letzte Block gerade gesendet wurde, wird die Übertragung beendet.

*** DAS PROTOKOLL DES EMPFANGERS (SUCK) ***

1. Auf STX warten und ACK senden, wenn STX empfangen wurde.
2. Dateiname, Blocknummer, Blocklänge, Datenblock und Prüfsumme empfangen (nähere Erläuterung beim Befehl BLOW!).
3. Es wird der Dateiname überprüft (derselbe Name wie beim ersten Block ?) und es wird überprüft, ob die empfangene Blocknummer folgerichtig nummeriert ist. Stimmt eine der Bedingungen nicht, wird ETX gesendet und der Empfang beendet.
4. Die empfangenen Daten werden auf die empfangene Prüfsumme hin überprüft und es wird getestet, ob ein Hardwarefehler auftrat. Wenn etwas nicht stimmig ist, wird NAK gesendet und mit Punkt 2 weitergemacht (den gleichen Block nochmals empfangen), ansonsten wird ACK gesendet.
5. Wenn als Blocklänge null übergeben wurde, wird der Empfang beendet, ansonsten wird mit Punkt 2 weitergemacht (den nächsten Block empfangen).

Da die Übertragung einer Datei mit dem Senden von STX und warten auf ACK beginnt, muß der Empfänger mit dem Start von SUCK beginnen, bevor der Sender den Befehl BLOW aufruft.

Die Werte für ACK, ETX, NAK und STX sind:

ACK: CTRL F oder 6 dezimal

ETX: CTRL C oder 3 dezimal

NAK: CTRL U oder 21 dezimal

STX: CTRL B oder 2 dezimal.

ANHANG 6 - ein Beispiel unter BASIC

Um den Gebrauch von einigen der neuen RSX-Befehle zu demonstrieren, folgt hier ein BASIC-Programm. Damit können Sie dann etwas einfacher DFU betreiben. Schnittstellenparameter können menügesteuert eingegeben werden, man hat einen Puffer, in den empfangene Zeichen abgelegt werden, man kann diesen Puffer auf Diskette sichern, etc. Am besten, Sie tippen das Programm ab und probieren es ganz einfach aus...

Aber beachten Sie bitte zwei Dinge: Sichern Sie das Programm vor dem ersten Start auf Diskette - ansonsten könnte alles verloren gehen, was Sie bis dahin abgetippt haben. Der zweite Punkt: der Puffer ist ca. 24 KByte lang (exakt: HIMEM-&2000) und da beim Laden einer ASCII-Datei in den Puffer keinerlei Längenüberprüfung gemacht wird, obliegt es Ihrer Verantwortung keine längeren Dateien zu laden! Das heißt im Klartext: Niemals zu lange Dateien in den Puffer laden - das ergibt den garantierten Absturz!

```

10 DEFINT s,z
20 DIM b(15)
30 s=0
40 oben=HIMEM
50 unten=&2000
60 MEMORY &16FF
70 point=unten
80 DATA 50,75,110,150,200,300,600,1200,1800,2000,2400,3600,4800,9
600,19200
90 FOR i=1 TO 15:READ b(i):NEXT i
100 DATA &dd,&66,&01,&dd,&6e,&00,&dd,&e5,&46,&23,&7e,&23,&66,&6f,
&11,&00,&18,&cd,&77,&bc,&01,&ff,&ff,&30,&12,&fe,&16,&20,&0b,&21,&
00,&20,&cd,&80,&bc,&77,&23,&03,&38,&f8,&cd,&7a,&bc,&dd,&e1,&dd,&6
6,&03,&dd,&6e,&02,&71,&23,&70,&c9
110 FOR i=0 TO 54:READ wert:POKE &1700+i,wert:NEXT i
120 DATA &dd,&66,&01,&dd,&6e,&00,&46,&23,&7e,&23,&66,&6f,&11,&00,
&18,&dd,&e5,&cd,&8c,&bc,&dd,&e1,&d0,&dd,&46,&03,&dd,&4e,&02,&21,&
00,&20,&7e,&cd,&95,&bc,&23,&0b,&78,&b1,&20,&f6,&c3,&8f,&bc
130 FOR i=0 TO 44:READ wert:POKE &1780+i,wert:NEXT i
140 baud1=300:baud2=300
150 DAT=8:STP=0:parity=0
160 ÖSETTIMEOUT,10
170 ÖSETSIO,baud1,baud2,0,dat,parity,stp
180 ON BREAK GOSUB 1130:MODE 2
190 PRINT"1 - TERMINAL"
200 PRINT"2 - Parameter einstellen"
210 PRINT"3 - Puffer laden"
220 PRINT"4 - Puffer sichern"
230 PRINT"5 - Puffer senden"
240 PRINT"6 - Puffer fuellen"
250 PRINT:PRINT"0 - Programmende"

```

```

260 k$=INKEY$
270 IF k$<>"1" THEN 440
280 REM *** TERMINAL ***
290 ON BREAK GOSUB 1140
300 MODE 2:PRINT"TERMINAL - Menue mit zweimal ESC":PRINT
310 s=0
320 z=0
330 CALL &BB8A
340 e=0
350 IF e=1 THEN 180 ELSE ÖINCHAR,$s,$z
360 IF s<>0 THEN 400
370 CALL &BB8A:PRINT CHR$(z);:CALL &BB8A:POKE point,z
380 point=point+1:cnt=cnt+1
390 IF point=oben THEN point=unten
400 k$=INKEY$
410 IF LEN(k$)=0 THEN 350
420 ÖOUTCHAR,$s,ASC(k$)
430 GOTO 350
440 IF k$<>"2" THEN 860
450 REM *** Parameter einstellen ***
460 MODE 2
470 PRINT"1 - Sendebaudrate (derzeit: ";baud1;")"
480 PRINT"2 - Empfangsbaudrate (derzeit: ";baud2;")"
490 PRINT"3 - Datenbits (derzeit: ";dat;")"
500 PRINT"4 - Paritaet (derzeit: ";
510 IF parity=0 THEN PRINT"keine Paritaetspruefung )"
520 IF parity=1 THEN PRINT"gerade Paritaet )"
530 IF parity=2 THEN PRINT"ungerade Paritaet )"
540 PRINT"5 - Stopbits (derzeit: ";
550 IF stp=0 THEN PRINT"ein )" ELSE IF stp=1 THEN PRINT" eineinh
alb )" ELSE PRINT" zwei )"
560 PRINT:PRINT"0 - HAUPTMENU"
570 k$=INKEY$:IF k$="" THEN 570
580 IF k$<>"1" THEN 620
590 MODE 2:PRINT"SENDEBAUDRATE":PRINT
600 GOSUB 660
610 baud1=baud:GOTO 460
620 IF k$<>"2" THEN 700
630 MODE 2:PRINT"EMPFANGSBAUDRATE":PRINT
640 GOSUB 660
650 baud2=baud:GOTO 460
660 REM ... Baudrate waehlen ...
670 FOR i=1 TO 15:PRINT CHR$(64+i);" - ";b(i):NEXT i
680 k$=UPPER$(INKEY$):IF k$<"A" OR k$>"O" THEN 680
690 baud=b(ASC(k$)-64):RETURN
700 IF k$<>"3" THEN 750
710 MODE 2:PRINT"DATENBITS":PRINT
720 PRINT"1 - fuef Datenbits":PRINT"2 - sechs Datenbits":PRINT"3
- sieben Datenbits":PRINT"4 - acht Datenbits"
730 k$=INKEY$:IF k$<"1" OR k$>"4" THEN 730
740 dat=ASC(k$)-&2C:GOTO 460
750 IF k$<>"4" THEN 800
760 MODE 2:PRINT"PARITAET":PRINT
770 PRINT"1 - keine Paritaetspruefung":PRINT"2 - ungerade Paritae
t":PRINT"3 - gerade Paritaet"

```

```

780 k$=INKEY$:IF k$<"1" OR k$>"3" THEN 780
790 parity=ASC(k$)-&31:GOTO 460
800 IF k$<>"5" THEN 850
810 MODE 2:PRINT"STOPBITS":PRINT
820 PRINT"1 - ein Stopbit":PRINT"2 - eineinhalb Stopbits":PRINT"3
- zwei Stopbits"
830 k$=INKEY$:IF k$<"1" OR k$>"3" THEN 830
840 stp=ASC(k$)-&31:GOTO 460
850 IF k$="0" THEN 170 ELSE 570
860 IF k$<>"3" THEN 920
870 REM *** Puffer laden ***
880 MODE 2:PRINT"PUFFER LADEN":PRINT:INPUT"Dateiname";d$
890 CALL &1700,$s,$d$
900 point=&2001+s
910 GOTO 180
920 IF k$<>"4"THEN 980
930 REM *** Puffer speichern ***
940 MODE 2:PRINT"PUFFER SPEICHERN":PRINT:INPUT"Dateiname";d$
950 s=point-untent+1
960 CALL &1780,$s,$d$
970 GOTO 180
980 IF k$<>"5" THEN 1040
990 REM *** Puffer senden ***
1000 MODE 2:FOR i=untent TO point:PRINT CHR$(PEEK(i));
1010 $OUTCHAR,$s,PEEK(i)
1020 IF s=512 THEN 1010
1030 NEXT i:GOTO 180
1040 IF k$<>"6" THEN 1100
1050 MODE 2:PRINT"Puffer fuellen - Menue mit zweimal ESC":PRINT
1060 e=0:point=untent:ON BREAK GOSUB 1140:CALL &BB8A
1070 IF e=1 THEN 180 ELSE k$=INKEY$:IF k$="" THEN 1070
1080 CALL &BB8A:PRINT k$;:CALL &BB8A:POKE point,ASC(k$):point=poi
nt+1:IF point=obent THEN point=untent
1090 GOTO 1070
1100 IF k$<>"0" THEN 260
1110 MODE 2
1120 END
1130 RETURN
1140 e=1:RETURN

```