

Benutzung der SIO2PC Interface Software (Autor: Nick Kennedy)

*** Wichtige Notiz ****

SIO2PC ist ein Hobby von mir und ein Service für Atari-User, kein kommerzielles Projekt. Deshalb übernehme ich keine Verantwortung für die Benutzung der Software oder die Herstellung des Interfaces.

Dies ist die Beschreibung für Release 3.15 Version. Die Software ist Menügesteuert, und ich hoffe, gut selbst erklärend. Hast du ein falsches Kommando gewählt, bringt dich ESCape wieder zurück zum vorhergehenden Menü.

SIO2PC ermöglicht das BOOTen des Atari wobei der PC als Diskettenstation(en) benutzt wird. Die vorher auf dem PC mit SIO2PC angemeldeten Diskettenlaufwerke (Disk-Image-Files) können mit dem Atari benutzt werden, genauso als wenn die Laufwerke am Atari angeschlossen wären.

+++ Anmerkung des Übersetzers:

Im folgenden Text ist des öfteren von einer RamDisk oder einer RAMDISK die Rede. Die unterschiedlichen Schreibweisen sind bewußt gewählt, um hier zwei verschiedene Typen zu kennzeichnen. Im originalen amerikanischen Text war dies nicht der Fall. Dies könnte aber Nicht-Experten total aus dem Konzept bringen.

Die Bedeutungen sind folgende:

- RamDisk: ein auf dem PC im konventionellen Speicher gehaltenes Disk-Image-File (Abbild einer Diskette vom Atari).
- RAMDISK: Die Hardware-RAMDISK des Atari 130XE oder auch Eigenbau-RAMDISK's die vom ATARI als virtuelles Laufwerk D8 benutzt wird.

+++

Kommandos:

C:

CREATE ATARI DISK IMAGE: reserviert RAM für eine RamDisk, Single oder Double Density, giebt die Größe aus, zeigt an wenn kein RAM verfügbar ist und erstellt das File als Swap-File für die RamDisk. Nach erstellen der Disk wird diese formatiert wie mit dem Atari. Der Disk-Status, Größe und Density, ist in der Status-Zeile zu sehen. Ein Einbuchstaben-Feld zeigt 'N' wenn nicht zur RamDisk geschrieben wurde, 'W' wenn geschrieben wurde und 'F' für Swap-File. Diese Felder sind farbig zu sehen (weiß, rot und grün). Hier können alle Disks von #1 bis #8 angemeldet werden.

L:

LOAD DISK IMAGE FROM FILE: Dieses Kommando fragt nach Disk-Nummer und Filenamen. DOS-Fehler werden angezeigt, wenn das angegebene File nicht verfügbar ist. Die File-Information wird in der Statuszeile mit angezeigt. Ist nicht genug Speicher in der RamDisk verfügbar, wird das Swap-File benutzt. (zum erzwingen der Swap-File-Benutzung muß der Parameter "/P" ohne Space angehängt werden.)

S:

SET PORT NUMBER: Dieses Kommando fragt den COM-Port ab, der zum Dialog mit dem Atari benutzt werden soll. beim Start wird die Port-Nummer automatisch abgefragt wenn sie nicht in der Kommandozeile oder in einem Start-Batch definiert war.

H:

DOS SHELL: beendet SIO2PC und startet eine DOS-Shell, Exit kehrt zu SIO2PC zurück. COMMAND.COM muß sich im Root von C: befinden (C:\COMMAND.COM). Es dürfen keine TSR's geladen sein, die eine DOS-Shell benutzen.

K:

LOCK DATA LINE: Standard ist „Auto“, hiermit kann evtl. ein BUG übergangen werden. Mehr dazu in _README.TXT.

D:

DISPLAY DIRECTORY: Ermöglicht das anzeigen der PC-Directory ohne SIO2PC zu verlassen. Es kann ein Pfad angegeben werden und Wildcards dürfen benutzt werden. Du kannst also die Directory des Atari-Disk-Image-Files einsehen. Dateinamen enden mit "S" wenn sie von SPARTADOS zur Directory-extraction benutzt werden.

U:

UNINSTALL DISK: Löscht ein geladenes Disk Image aus dem Programm, bei RamDisk's wird der benutzte Speicher wieder freigegeben. ACHTUNG: Da sich jede RamDisk im Speicher des PC befindet muß diese –wenn sie verändert wurde- vor UNINSTALL gespeichert werden.

A:

ALTER BUS TIMINGS: Ein Untermenü für nicht Standardisierte Systeme, das mehrere Funktionen beinhaltet. Die Kommandozeilen-Logik kann hiermit beeinflusst werden sowie die Hardware konfiguriert. Ein 'i' hinter der Revision-Nummer zeigt an, daß neuere Hardware Benutzt wird, die dem Programm evtl. nicht bekannt ist. Mehr Info's in README.DOC

I:

INSTALL PC FILE AS ATARI DISK: installiert ein (Atari-)PC-File als Atari-Disk- Image. Es erstellt ein "falsches" Atari-Disk-Image das nur dieses eine File enthält. Dieses File kann nur normal gelesen werden, ein Schreibzugriff oder Sektorzugriff ist nicht möglich. Wenn dieses File ein BOOT-File ist, benutze den Schalter /B nach dem Filenamem.

W:

WRITE DISK IMAGE TO FILE: Das ist die grundsätzlich umgekehrte Funktion von "Install PC File as Atari Disk". Es kann von einem Disk-Image ein neues PC-File erstellt werden. Diese Option kann nicht auf Disk-Images angewendet werden, die gerade für den Atari geöffnet sind.

T:

TOGGLE STATUS LINE: Die Status-Zeile gibt Info's über die gerade laufenden Funktionen aus. Sie kann hiermit aus-/eingeschaltet werden.

X:

EXCHANGE DISK I.D. NUMBERS: Ermöglicht, die Nummer des Standard-(BOOT)- Laufwerkes zu ändern. Hast du den Atari von Drive 1 gestartet und in Drive 2 befinden sich Daten, so kannst du jetzt Drive 1 - Drive 2 umkehren.

P:

PRINT_THRU: Weist SIO2PC an, die Atari-Druck-Funktion auf dem PC auszuführen. Du kannst auf dem Atari Daten zum Drucker, zum Bildschirm oder in eine Datei drucken. Dies übernimmt diese Funktion. Dabei wird das Atari-EOL in ein PC-CR oder CR/LF umgewandelt, ATASCII in ASCII und oder die höherwertigen Bytes gelöscht. Ein Feld im Menü zeigt den Status dieser Funktion an.

E:

ENTER PORT ADDRESS: ermöglicht die Änderung der COM-Port-Adresse. Normalerweise ist dies nicht notwendig.

J:

JUMP TO ULTRA SPEED: Diese Kommando schaltet in den 'SpartaDos' High-speed-Mode und zurück. Normalerweise geschieht dies automatisch, falls aber die synchronisation nicht übereinstimmt, kann diese Funktion helfen.

Q:
QUIT PROGRAM: Zurück zum PC-DOS. Warnung: RamDisk's abspeichern.

=====

REDRAW SCREEN: Drücke "R" oder ESCape um den Bildschirm nach einer PRINT-THRU oder anderer Operation wieder herzustellen.

BLANK SCREEN: Drücke "B", um den Bildschirm zu löschen, ESCape zum wiederherstellen.

USING COMMAND LINE: SIO2PC kann auch von der Kommandozeile beim aufrufen konfiguriert werden. Ein Beispiel:

```
SIO2PC 2 L1FILENAME.EXT CD263 CD8510000FILENAME.ATR /F
```

Parameter 1 (2):

2 COM-Port (muß immer als erster Parameter stehen)

Parameter 2 (L1FILENAME.EXT):

L Load Disk-Image-File
1 als Laufwerk Nummer 1
FILENAME.EXT Dateiname des zu ladenden Imagefiles

Parameter 3 (CD263):

C Kommando Create RAMDisk
D Density: D=Double, S=Single
2 Laufwerk 2
6 Kennzahl 6
3 Auswahl 3 = 720 Sektoren, 180 kByte

Parameter 4 (CD8510000FILENAME.ATR /F):

C Kommando Create RAMDisk
D Density: D=Double, S=Single
7 Laufwerk 8
4 Kennzahl 5, frei wählbare Größe
10000 Größe in Sektoren, max. 65535 bei Kennzahl 5 (bis 4 nicht nötig)
RAMDISK.ATR FILENAME.EXT (bis Kennzahl 4 nicht nötig)

/F als File auf Festplatte

SIO2PC kann auch aus einem Batch-File heraus mit Parametern gestartet werden, allerdings sollte die Kommandozeile hinter dem letzten Parameter ein Leerzeichen enthalten.

Noch ein Beispiel:

```
SIO2PC 2 C24 L1A:\DOS.ATR R
```

- startet (SIO2PC)

- benutzt COM(2)
- (C)reate Disk Image als D(2):, Größe: Kennzahl (4) 180kByte
- (L)ädt als D(1) vom Disk Image von (A:\DOS.ATR) als (R)amdisk

Die in Klammern dargestellten Zeichen sind die Parameter der Kommandozeile.

Für Create RamDisk gelten folgende Ordnungen:

Aufruf: C

Dichte: S=Single, D=Double

Laufwerk: 1 bis 8

Größe: 1=64k, 2=90k, 3=130k, 4=180K,
5=frei wählbar bis 65535 Sektoren (die Anzahl muß auf 5 folgen)
6=aus einer Auswahl wählbar:

Dichte	Auswahl	Seiten	Tracks	Secs/Trk	KBytes	Sektoren	
Single	1	1	35	18	79	630	
Single	2	1	35	26	114	910	
Single	3	2	35	18	157	1260	
Single	4	2	35	26	227	1820	
Single	5	1	40	18	90	720	Single Dens.
Single	6	1	40	26	130	1040	Enhanced Dens.
Single	7	1	77	26	250	2002	
Single	8	2	77	26	500	4004	
Single	9	1	80	18	180	1440	
Single	A	1	80	26	260	2080	
Single	B	2	80	18	360	2880	
Single	C	2	80	26	520	8200	
Double	1	1	35	18	157	630	
Double	2	2	35	18	315	1260	
Double	3	1	40	18	180	720	Double Dens.
Double	4	1	77	26	500	2002	
Double	5	2	77	26	1000	4004	
Double	6	1	80	18	360	1440	
Double	7	2	80	18	720	2880	

=====

ACOMINIT.COM: Für ältere Hardware, bei der SIO2PC nicht in der Lage ist, einen COM-Port zu Initialisieren, dient dieses Programm. Normalerweise wird es nicht gebraucht.

TESTIME1.EXE: Testet den Computer, ob er kompatibel mit SIO2PC ist. Starte es, wenn du Schwierigkeiten hast. Testdauer ist 10 Sekunden.

Start-Reihenfolge:

kopiere die Files der Atari-Emulator-Diskette in eine Directory auf dem PC,
Starte SIO2PC so: "SIO2PC # [ENTER]", wobei # die Nummer des COM-Ports ist,
Lade die mitgelieferte Atari-Diskette mit "L" dann "1" dann "ATARI.ATR".

Die Statuszeile müßte jetzt zeigen, daß eine RamDisk geladen ist.
Schalte jetzt den Atari ein, und er BOOTet vom PC.
Die Statuszeile muß nun folgendes zeigen:
Commando: benutzte "R",

DEVID: 31 (für Disk 1),

COM: "H" flashing to "L",

Nun kannst du den Atari benutzen, rufe z.B. "A" für Directory, dann zeigt u.a. diese folgende Files an:

REMOTE .OBJ (SIO2PC für den Atari)

FILE2PC .OBJ (Atari-File in PC-File)

TANK .OBJ (ein Spiel)

KEYER .OBJ (ein HAM RADIO Programm)

VERANTWORTUNG

Ich kann nicht versichern, daß die Software frei von Fehlern oder Viren ist. Die Benutzung der Software geschieht auf EIGENES RISIKO! Gleiches gilt für die Herstellung und Benutzung der Hardware. Ohne praktische Erfahrung im Umgang mit Elektronik sollte niemand den Lötkolben ansetzen.

Fehler in der Schaltung und Konfigurationsunterschiede sind möglich und ich garantiere nicht, daß diese Fehler keine Schäden verursachen. Weiterhin muß -wer mein Interface bauen will- einiges an Kenntnissen in der Elektronik mitbringen. Wer diese nicht hat, kann leicht Schäden verursachen. Generell sollten -wenn der Atari und der PC am seriellen Port verbunden sind- keine direkten Sektor-schreib oder -lese Operationen ausgeführt werden. Der PC ist so komplex (File-Operationen können umgeleitet werden, residente Programme oder TSR's können einwirken etc.), daß ich bei einer solchen Operation nicht für die Datensicherheit garantieren kann.

Ich denke, das SIO2PC-System ist ein Service für die Atari-User-Gemeinschaft. Obwohl ich das Shareware-Konzept gewählt habe, und einen Betrag für den Aufbau des Interfaces verlange, ist diese Arbeit mein Hobby.

WENN DU SOFTWARE, BAUTEILE UND/ODER EIN FERTIGES INTERFACE BEI MIR GEKAUFT HAST, UND DU LEHNST DIE BENUTZUNG AUF EIGENES RISIKO DOCH AB, SCHICKE MIR ALLES ZURÜCK UND DU BEKOMMST DEIN GELD ZURÜCK. WENN DU DAS INTERFACE SELBST AUFBAUST, UND DU ES NICHT SELBST AUSREICHEND PRÜFEN KANNST VOR INBETRIEBNAHME DANN SUCHE DIR RAT UND HILFE BEI EINEM ELEKTRONIK-FACHMANN. ANSONSTEN VERSUCHE NICHT, DAS INTERFACE ZU BAUEN UND ZU BENUTZEN:

BENUTZE SIO2PC NICHT AN COMPUTERN DIE COMMERZIELL GENUTZT WERDEN ODER AUF DENEN HOCHWERTIGE SOFTWARE LÄUFT:

EINFÜHRUNG

Grüße an alle Atari und PC User! SIO2PC ist Shareware. Wenn du nur wenig oder garnichts über das System weißt, gebe ich dir hier Hilfe.

SIO2PC ist ein Hardware- und Softwarepaket zur Benutzung des PC's als Floppy, Drucker, Printfilespeicher etc. für den 8-Bit-Atari.

Möglichkeiten:

- erstellt und verwaltet ein bis vier virtuelle Atari-Floppy's (Disketten) vom PC aus.
- Die virtuellen Disk's können im PC-Speicher (RamDisk) oder auf der Festplatte gehalten werden und können bis zu 65535 Sektoren groß sein.
- Die RamDisk's können auf die Festplatte abgespeichert werden.

- Der Atari kann vom PC aus geBOOTet werden. Es ist keine reale Floppy erforderlich.
- erstellt Single- und Double Density Disketten
- Keine speziellen Atari-Software-Veränderungen. Du kannst dein Lieblings-DOS verwenden
- schneller als jede reale Floppy.
- kann in die Kette der Atari-Zusatzgeräte (Floppy, Drucker etc.) eingefügt werden.
- Die Software ist in Assembler geschrieben, sie ist kompakt und schnell.
- kompatibel zu allen Hardware-Standards des Atari (CIO, SIO und andere Hardware-Calls)
- Print-Thru-Modus überträgt deine Atari-Druckaufträge an den PC und druckt sie dort auf Drucker, Bildschirm oder in ein File.
- File-conversion überträgt und konvertiert PC-Files zum Atari und Atari-Files zum PC.
- Die Status-Zeile zeigt genau an, welche Funktion der Atari gerade von den Diskettenlaufwerken (virtuelle und reale) fordert: Drive-Nummer, Kommando, Sector-Nummer etc.

WENN DU SICHERGEHEN WILLST, DASS DEINE ATARI-DISK-IMAGE-FILES NICHT ZERSTÖRT WERDEN, BENUTZE KEINE DIREKT-SEKTOR-ZUGRIFFE AUF EINE VIRTUELLE FLOPPY:

Hardware-Anforderungen:

SIO2PC verbindet einen PC-kompatiblen Computer mit einem Atari-8-Bit-Computer (400, 800, 800XL, 130XE).

Die Verbindung zwischen dem PC und dem Atari erfordert ein einfaches Interface. Es verwendet einen IC und einige Kondensatoren und kann auf einer 4 x 5 cm großen Platine aufgebaut werden. Wenn du das Interface selbst aufbauen willst, lies BUILT_IT.TXT zur Erläuterung. Darin giblt es detaillierte Info's zum Aufbau.

Technische Informationen zu SIO2PC:

- SIO2PC ist 100% geschrieben in Assembler
- REMOTE.OBJ und FILE2PC.OBJ sind in 6502-Assembler geschrieben
- MAKADISK.EXE ist in 'C' geschrieben
- SIO2PC erkennt die SIO-Bus-Kommandos: READ SECTOR, WRITE SECTOR, PUT SECTOR, STATUS, FORMAT, FORMAT 1050 1.5 DENSITY
- Wenn diese Grundfunktionen benutzt werden, arbeitet SIO2PC mit DOS- und mit Nicht-DOS-Disketten-Bootladern zusammen. Alle diese Disketten können vom PC aus den Atari BOOTen
- SIO2PC benutzt ausschließlich MS-DOS-Aufrufe zu File-Input und File-Output auf dem PC (keine Sector-Zugriffe).
- SIO2PC löscht die Disk-Images, wenn nicht die richtigen Pfad-/Filnamen beim speichern verwendet werden
- SIO2PC benutzt File-Handler, es benötigt also MS-DOS, PC-DOS 2.0 oder höher
- SIO2PC liest den Timer-0-Interrupt aus für seine Arbeit, das ist der BIOS- Timer, 18 mal in der Sekunde, 55 Millisekunden pro "tick". Es wird also im Millisekunden-Bereich gearbeitet, um höchste Genauigkeit zu erreichen.
(wenn dein PC nicht den Standard-Timer-Chip oder einen kompatiblen verwendet, giebt es Probleme. Die meisten PC's sind kompatibel)
- SIO2PC liest die Port-Adresse für den benutzten COM-Port aus dem BIOS-Speicher als Word ab Adresse 0004:0000, 0040:0002, 0040:0004, 0004:0006 für COM1, COM2, COM3, COM4. Ich habe herausgefunden, daß es für COM3 und COM4 nicht unbedingt typische hex Adressen giebt. Es werden aber die typischen Adressen 03F8, 02F8, 03E8, 02E8 für COM 1-4 vom POST ausgelesen. Ist die Adresse auf deinem PC anders, suche in der PC-Dokumentation danach und benutze die Funktion "E" zum ändern der Adresse.

FILE2PC (Programm):

FILE2PC benutzt die SIO2PC-eigenen BUS-ID als Schalter. ID# 03Ahex setzt das SIO2PC-Programm in einen File-Transfer-Modus. Der Atari (geladenenes FILE2PC) fragt den Status des "Gerätes" 03Ahex ab, und erst bei positiver Antwort werden die Sektoren zum PC geschrieben. Ist das erste Status-Byte (von 4) normal (010hex) ist der PC OK für den File-Transfer. Ist das Byte

OFFhex, gibt der Atari eine Fehlermeldung aus. Reihenfolge: Der Atari fragt den Status ab. Ist der OK, wird ein Sector an den PC gesendet, der PC nimmt aus diesem Sector Pfad und Filenamen des zu schreibenden Files, abgeschlossen von einem ATASCII-EOL. Der Atari wartet 4 Sekunden, die der PC zum erstellen des Files verwenden kann.

NOTIZ: Ein mit diesem Namen existierendes File wird überschrieben.

Nach erneuter Status-Abfrage werden die Sektoren des zu kopierenden Files an den PC gesendet. Der Atari fragt nach jedem Sector den PC-Status ab. Die AUX-Bytes, die die Sector-Nummer enthalten, sagen dem PC folgendes:

AUX1 = 0: Full Sector, 128 Bytes

AUX1 = 1: Last Sector, AUX2 enthält den Byte-Zähler (Pointer)

AUX1 = 2: Atari hat einen Fehler festgestellt, Abbruch

STRUKTUR EINES SIO2PC-ATARI-DISK-IMAGE-FILES:

Das ist sehr einfach, die ersten 16 Byte enthalten den Header (Kopf):

1 WORD = spezial-Code*, dies ist ein Atari Disk File

1 WORD = Größe des Disk-Image, in Zahlen (Größe/16)

1 WORD = Sectorgröße, 128 oder 256 Bytes pro Sector

8 leere Bytes (enthalten Nullen)

Nach dem Header folgt das Disk-Image. Das ist eine Folge von Bytes, in der die ersten 128 Byte den Sector 1 enthalten, die zweiten 128 Byte den Sector 2 etc.

* Der "spezial-Code" ist die 16-Bit-Summe der ASCII-Werte des Strings of Bytes NICKATARI". Wenn du ein File ohne dieses erste WORD laden willst, gibt's die Fehlermeldung: "THIS IS NOT AN ATARI DISK FILE". Versuch es.

RamDisk's benutzen:

SIO2PC benutzt -oder kann benutzen- die PC Hard- oder Floppy-Disk zu speichern der Atari-Disketten. Um maximale Geschwindigkeit zu erreichen, kann die Diskette aber auch als RamDisk im Speicher gehalten werden. Wenn der Atari schreibt/liest mit SIO2PC, kommuniziert er also mit dem PC-Speicher. Du kannst dir aussuchen, ob du eine RamDisk oder eine echte Disk benutzen willst. Für Files, die zu groß für eine RamDisk sind, wird automatisch die physische Speicherung gewählt.

TIMING INFORMATION:

Das Atari SIO2PC-Bus Protokoll spezifiziert die minimale und maximale Zeit in der die Atari-Kommandos von den Disketten-Laufwerken beantwortet werden müssen. Das Minimum gilbt der Atari, um die Daten exact an den Bus zu schicken. Das Maximum läßt den Atari, wenn das Peripherie-Gerät nicht in dieser Zeit antwortet, einen Time-Out-Fehler zu melden. Ich habe herausgefunden, das daß System sehr flexibel und versöhnlich ist. Ich habe ausgedehnte Variationen benutzt und hatte immer noch ein arbeitendes System.

Aber für einen Kerl (jetzt bist du gemeint, Joe!) oder vielleicht zwei, habe ich das Menü mit verschiedenen Möglichkeiten getestet. Das System führt einen kleinen Kommando-Tanz durch (Command-Acknowledge-Data-Acknowledge-Complete-etc.) mit timings dazwischen. Durch Testen dieser Zeiten innerhalb der Bus-Rate konnte ich wieder (normal Null-) Wartezeiten einsetzen.

Primär: Am seriellen Bus werden die Daten normal übertragen, ohne Wartezeiten (mit Start- und Stop-Bits) zwischen den Bytes. Jetzt kann man mit Zwischenzeiten experimentieren, mein System verlangsamt und startet normal.

Sekundär: Die Fachliteratur spricht, das einige UART-(serial-) Chips keine schnelle Adressierung

vertragen. Ich habe ein Bit Wartezeit eingebaut, aber du kannst mehr einbauen. Die Wartezeit sieht so aus: Jede Einheit ist 850 Nanosekunden bis eine Millisekunde lang. (Ausnahme ist die Drucker-Zeit mit 1/18tel einer Sekunde) Also müssen die Einheiten vier Hex-Digits groß sein. Keine Panic, das ist einfach. Hier einige Umrechnungen:

Decimal	Hex
0005	0005
0010	000A
0050	0032
0100	0064
0250	00FA
0500	01F4
0750	02EE
1000	03E8
2000	07D0

Das Timing-Menü gibt die Default-Zeiten aus. Wenn du z.B. die Wartezeit für den Drucker verändern willst, benutze #8. Das Problem an der Sache ist, daß diese Zeiten nicht beim Start aus der Kommandozeile verändert werden können. Wenn dein System nicht mit den Standard-Zeiten läuft, mußst du diese nach jedem Programm-Start von Hand verändern.

Hier eine Möglichkeit der permanenten Änderung:

Dazu brauchst du das DOS-Utility "DEBUG", das wohl jeder hat. Man kann aber auch einen Diskeditor (z.B. Norton Utilities) verwenden. Ich habe die Wartezeiten an den Anfang des Programms gelegt und einen ASCII-String an den Anfang gestellt, um es einfacher zu machen.

Wechsele in das Verzeichnis, das SIO2PC.COM enthält. An der DOS-Eingabezeile gibst du ein: DEBUG SIO2PC.COM <ENTER>. Jetzt siehst du einen Bindestrich, er ist das Eingabezeichen von DEBUG.

Drücke D <ENTER>. Eine Tabelle mit Hex-Zahlen erscheint. Links siehst du die zu bearbeitende Adresse als da wäre: SEGMENT:OFFSET des ersten Hex-Bytes jeder Zeile. Ganz rechts ist der ASCII (Text) Code dazu zu sehen. Du siehst den ersten der ASCII-Einträge:

"TIMING DELAY VALUES: T1-T8, 2 BYTES PER ENTRY:.", und schaust in die Zeile mit dem Offset :1030. Diese Zeile enthält die Timing Values:

????:0130 ?? ?? ?? 64 00 B0 04-01 etc.

Erklärung: Jede dieser 2-Digit Hex-Zahlen ist ein Byte. Aber jedes Timing Value ist ein Word (2 Bytes). Und der PC speichert in einem Word das niedrigere (LOW) Byte zuerst. Die Zeiten sind also: T1 = 0064, T2 = 04B0, T3 = 012C etc.

Die gute Seite an der Sache ist, daß es einen ganz einfachen Weg gibt. Mach genau das, was jetzt folgt. drücke im Debugger Q <ENTER> zum verlassen.

Ich empfehle jetzt, SIO2PC.COM auf eine Diskette zu kopieren, damit DEBUG nicht auf der Festplatte arbeitet.

+++ Anmerkung des Übersetzers:

Eine sehr gute Empfehlung, denn DEBUG ist ein sehr mächtiges Werkzeug, mit dem man auf dem PC jede Menge Unsinn verzapfen kann.

+++

OK, hast du dies getan, und die Diskette im Laufwerk A:, gib ein "A:" und <ENTER>. Jetzt "DEBUG SIO2PC.COM" <ENTER>, jetzt das Go-Kommando "G" <ENTER>. SIO2PC startet von der Diskette. Jetzt gib eine Port-Nummer (1-4) ein, wenn das Menü erscheint gehe mit "A" in das Timing-Menü.

Im Timing-Menü kannst du jetzt die Veränderungen vornehmen, die dein PC braucht. Beende das Timing-Menü, beende SIO2PC und du siehst wieder den Bindestrich-Prompt von DEBUG. Zum zurückschreiben von SIO2PC auf Diskette mußst du jetzt nur noch "W" <ENTER> drücken und dann den DEBUGger mit "Q" <ENTER> beenden.

Starte jetzt SIO2PC ohne den DEBUGger und kontrolliere im Timing-Menü die Änderungen. In Der Default-Spalte sind die normalen Zeiten zu sehen und in der Current-Spalte die gültigen Änderungen. NOTIZ: SIO2PC gibt eine Botschaft aus, daß es verändert wurde und dass dies ein möglicher Virus sein

kann. Bei Bestätigung wird ein neuer CRC geschrieben und beim nächsten Programmstart ist wieder alles OK.

DOUBLE DENSITY:

Der Atari akzeptiert die ersten drei Sektoren einer Diskette nur in Single-Density (128 Byte). Dies ist notwendig, um den Atari von der Diskette BOOTen zu können. SIO2PC verwendet ebenfalls dieses Format für den Double-Modus.

FORMAT gegen DISK-GRÖSSEN:

Allgemeines: Benutze 133k für "Enhanced"-DOS 1.5" oder "DOS 2.5" Disk's. Die Größe 183k ist für Double-Density. Es hängt vom DOS ab, ob Double genutzt werden kann. Einige DOS'en können die aktuelle Disk-Größe ausgeben, andere erwarten immer die default-Größe. MyDOS z.B. ist sehr flexibel, SPARTADOS ebenfalls. Ab REV 3.00 antwortet SIO2PC auf "SET"- und "GET"-Konfigurationskommandos. Du kannst also beim erstellen eines Disk-Image-Files die Anzahl der Sektoren spezifizieren. Als Resultat dieser Wahlmöglichkeit kannst du jetzt mit SIO2PC ungeheuer große Disketten erstellen.

File-Zugriff auf "Mega-Disk's":

Für eine Mega-Disk wird meist eine Direct-Zugriff-File erstellt, da der verfügbare Arbeitsspeicher des PC nicht ausreicht. Um eine solche Disk zu erstellen, benutze die Größen-Einstellung #5. Dann kannst du die Größe in Sektoren eingeben. Wenn dich interessiert, wie groß die Disk wird, multipliziere die Sektoren mit 128 bei Single und mit 256 bei Double.

Die Statuszeile der Disk zeigt ein rotes "W", wenn auf die Disk geschrieben wurde und sie noch nicht gespeichert ist, ("N" - File ist gespeichert) bei Direkt-File-Zugriff steht hier "F".

Du kannst das System zwingen, den Direkt-File-Zugriff zu benutzen, indem du beim erstellen hinter dem Filenamen "/P" setzt. Dazwischen darf kein Leerzeichen stehen. Das "P" erscheint in der Disk-Statuszeile.

Setzen / Lesen der Konfiguration:

Selbst MyDOS kann nicht selbstständig erkennen, wenn eine Diskette größer als eine Standard-Disk ist, da das Standard-Atari-SIO-Protokoll eine sehr armselige Routine zur Meldung der Disk-Größe an den Atari enthält.

+++ Anmerkung des Übersetzers:

Diese Meldung übernimmt z.B. die Floppy 810 oder 1050, die beim einlegen der Diskette selbstständig die Größe testet, indem ein Sector auf Track 0 gelesen wird. Bei der letzten Atari-Schöpfung, der XF-551 haben die Entwickler diesen Test wohl vergessen. Hier muß nun wiederum das DOS diese Aufgabe übernehmen.

MyDOS kann auf Übergrößen konfiguriert werden, aber das muß von Hand getan werden.

Siehe hierzu MEGADISK.TXT, es wird ein Beispiel für eine 1440k-Diskette exerziert. MEGADISK.EXE ist ein gepacktes 1440k Disk-Image.

+++

Ich habe die erweiterten SIO-Kommandos zum setzen / lesen der Konfiguration mit in das Programm eingebaut. MyDOS zeigt die aktuelle Disk-Größe nach dem ersten Lese-Zugriff an, aber wie oben beschrieben nur bei Standard-Disk-Größen.

konfigurieren von MyDOS für ein neues (leeres) "Mega"-Disk-Image:

Drücke "O" (Set Configuration), die folgende Frage nach "Configurable" mit "Y"es beantworten, "High Capacity" - "Y"es, "Number of Sectors" -"xxxx" (die Anzahl der Sektoren). Dies muß vor dem formatieren der Diskette getan werden.

Formatieren von "Mega"-Disk-Image:

+++ Anmerkung des Übersetzers:

Der Original amerikanische Text dieses Abschnittes ist so konfus und verwurschtelt (siehe die vorhergehenden Abschnitte), daß ich hier nicht das Original wiedergebe, sondern nur eine Kurzfassung der Anmerkungen. Wie oben schon erwähnt, gibt MEGADISK.TXT eine gut strukturierte Anleitung hierzu.

+++

Die Konfigurationskommandos von SIO2PC und MyDOS sind unterschiedlich und nicht auf einen gemeinsamen Nenner zu bringen. Deshalb macht es sich erforderlich, daß für "Mega"-Disk's nur SINGLE-Density benutzt wird. Double-Density-Disk's werden von MyDOS falsch interpretiert.

Formatieren (+++ ebenfalls Kurzfassung +++):

Beim formatieren von "Mega"-Disk-Images vergeht einige Zeit. Das unbenutzte Disk-Image wird geleert, dann wird das Format aufgebracht. SIO2PC setzt beim erstellen einer solchen "Mega"-Disk ein Flag, das nach dem formatieren der Diskette wieder gelöscht wird. Dies ist notwendig, weil der Atari sonst einen Timeout-Error melden würde.

Laufwerks-Nummern:

Ein weiteres Feature ist, daß du deine Disk-Images als Disk 1 - 8 anmelden kannst, nicht wie bei den Original Floppy's nur als 1 - 4.

Arbeiten mit den Disk-Images:

Das ist sehr einfach. MS-DOS setzt einen Zeiger (Pointer) in geöffnete Files, der vom jeweiligen Programm benutzt wird. SIO2PC benutzt auch einen Zeiger um immer auf den nächsten (Atari-)Sector zu zeigen. Wenn SIO2PC einen Schreib- oder Lesevorgang durchführt, errechnet es den (Disk-Image-)Datei-Offset der nächsten Sector-Nummer. Ist dieser falsch, fragt es (Atari-)DOS nach der neuen Zeigerposition, die sich aus dem internen lesen oder schreiben von 128 bzw. 256 Bytes ergibt und zieht für den internen Zeiger 1 ab. Nun steht der DOS-Zeiger auf dem nächsten 128 oder 256 Byte Block des Files. Der Zeiger wird also automatisch gesetzt, damit der Atari ganz normal mit dem Disk-Image arbeiten kann und keine separaten Software-Eingriffe am Atari notwendig sind. Dies ist auch eine vernünftige Lösung, um die Disk-Images möglichst unfragmentiert zu halten.

Mehr über die Disk-Konfiguration:

Ich habe von der Unfähigkeit bestimmte Disk-Image-Größen zu formatieren gehört. Die "GET"- und "SEND-konfiguration" Funktion in SIO2PC funktionierte scheinbar nicht immer korrekt. Ich habe jetzt eine Routine eingebaut, die die benutzten Atari-DOS'se unterscheidet und sicher bestimmt.

+++ Anmerkung des Übersetzers:

Hier ist von Atari DOS 2.5, MyDOS und SPARTADOS die Rede. Das BIBO-DOS und auch das Turbo-DOS sind deutsche Entwicklungen, die der Programmierer Nick nicht kennt. Es könnte bei Benutzung dieser DOS'se (in nur sehr seltenen Fällen) zu unerklärlichen Fehlern kommen, muß aber nicht, da diese ja "voll DOS 2.5 kompatibel" sind.

+++

Das Problem: Wie kann SIO2PC dem Atari exakt mitteilen, welche Disk-Größe und -Typ gerade verwendet wird. Eine 12 Byte große Konfigurationstabelle tauscht über die GET/SEND-Konfiguration Informationen. Unglücklicherweise ignorieren MyDOS und SPARTADOS die meisten dieser Informationen.

+++ Anmerkung des Übersetzers:

DOS 2.5, BIBO-DOS, TURBO-DOS, HAPPY-DOS usw. verhalten sich ebenso wie das SPARTADOS, der Amerikaner kennt wohl nur das SPARTADOS. Wenn also im folgenden Text von SPARTADOS die Rede ist, trifft dies auch weitgehend auf die anderen DOS'se zu -Nicht auf MyDOS, es bildet hier die große Ausnahme-. Die Ausnahme des SPARTADOS ist, daß es mehr Standardgrößen kennt als die anderen DOS'se, diese Größen sind in der Auswahl #6 des "Create Disk Image"-Menüs zu sehen. Weitere Info's zu SPARTADOS am Schluß der Dokumentation.

+++

Die Info-Variablen sind:

SINGLE/DOUBLE DENSITY: 128 oder 256 Bytes pro Sector

SINGLE/DOUBLE SIDED DISK

ANZAHL DER TRACKS PRO SEITE: TYP 35, 40, 77, 80 excepted

ANZAHL DER SECTORS PRO TRACK: TYP 18 oder 26 excepted

MyDOS nimmt an, daß EIN großer Track alle Sektoren enthält. Das macht die Arbeit mit SIO2PC scheinbar einfach.

Du kannst also eine Diskettengröße außerhalb der Standards erstellen, indem du die Kennzahl #5 für Anzahl Sektoren benutzt. SPARTADOS wiederum erkennt diese Größen nicht bzw. es erkennt sie als eine 1050-Enhanced-Größe. Ist die Disk-Größe eine Standard-Größe, wird die Standard-Information an den Atari gesendet. Ist die Größe außerhalb des Standards, benutzt SIO2PC die MyDOS-Technik und schreibt alle Sektoren in einen Track. Der Standard-Status wird durch eine Tilde (~) im ERR-Feld der Status-Zeile angezeigt.

Beachte, daß die Varianten 1, 2, 3 & 4 der Auswahl #5 als Standard-Größen gelten. Die Äquivalenten Größen der Auswahl #6 sind dafür nicht weiter von Bedeutung. Die meisten DOS'se benutzen die GET/SEND-Konfiguration zur Erkennung der Disk-Größe.

+++ Anmerkung des Übersetzers:

Hierbei ist zu beachten, daß z.B. eine in SINGLE erstellte 180k-Diskette nicht von den DOS'sen erkannt wird, weil eine 180k-Diskette vom Standard her nun mal in DOUBLE-Density formatiert wird. Hier die Standards:

Single	1 Seite,	40 Tracks,	18 Sec/Track,	128 Bytes/Sec,	90 kByte
Enhanced	1 Seite,	40 Tracks,	26 Sec/Track,	128 Bytes/Sec,	133 kByte
Double	1 Seite,	40 Tracks,	18 Sec/Track,	256 Bytes/Sec,	180 kByte
Quad	2 Seiten,	80 Tracks,	18 Sec/Track,	256 Bytes/Sec,	360 kByte

ACHTUNG! Die Disketten mit 128 Bytes/Sector müssen in SIO2PC als SINGLE gewählt werden, die mit 256 Bytes/Sector als DOUBLE. Die Auswahlkriterien "SINGLE" und "DOUBLE" sind hier etwas gewöhnungsbedürftig, da sie ja normalerweise zur Beschreibung des Diskettenformats (Größe) dienen und nicht zur Kennzeichnung der Sektorgröße.

Die QUAD-Density ist mit SIO2PC nicht erstellbar; Ob der Amerikaner keine XF-551 kennt? Es gibt auch eine Deutsche "Floppy 2000", die ebenfalls Quad-Density beherrscht.

+++

Zum Errechnen der Standards gelten folgende Formeln:

Anzahl Sektoren = Tracks * Seiten * Sektoren pro Track

Größe in Byte = Anzahl Sektoren * Bytes pro Sector

Bei Double Density müssen von der errechneten Größe noch 384 Byte abgezogen werden, da die ersten 3 Sektoren nur 128 Bytes enthalten ($3 * 128 = 384$) SIO2PC gibt diese Informationen an den Atari weiter, MyDOS benutzt aber eine eigene Routine zur Erkennung. MyDOS liest einen wahllos ausgewählten Sector der Disk und wertet diesen ohne Rücksicht auf die Standardgrößen aus, scheinbar funktioniert das auch.

Sollten hier Fehler (GET/SEND) auftreten, habe ich ein weiteres Feature in SIO2PC eingebaut. Du kannst den Konfigurations-Transfer für jedes einzelne Disk-Image-(Laufwerk) abschalten. Da das Hauptmenü voll

ist, habe ich diese Funktion im Timing-Menü versteckt. Drücke also "A" für das Timing-Menü, dann "C" für Change Disk konfiguration und es wird der Status der Disk-konfiguration angezeigt. "Y" sagt aus, daß GET/SEND eingeschaltet ist, "N" steht für aus. Zum ändern mußt du nur die Nummer des Laufwerkes drücken.

Simulierte Disketten:

Das folgende Tool nenne ich "simulierte Disk". Die Funktion "I" ermöglicht es, ein PC-File als eine Atari-Diskette zu installieren. Die Funktion ist dieselbe wie bei MAKADISK.EXE. Die Möglichkeit, die Disk in ein Disk-Image zu konvertieren, besteht nicht. Die Funktionsweise ist folgende:

Die simulierte Disk (kein Größenlimit) wird in SINGLE-Density erstellt und ist DOS 2.0/2.5/MyDOS kompatibel. Es Kann NICHT darauf geschrieben werden, da die VTOC nicht korrekt ist. Aus diesem Grund wird die Disk nur sequentiell (ein Sector nach dem anderen) gelesen, ein wahlfreier Zugriff darauf ist ebenfalls nicht möglich.

Du kannst also ein Atari-File mit FILE2PC auf den PC kopieren und es danach direkt durch benutzen dieser Funktion direkt laden oder starten. Bei der Directory-Abfrage wird dieses PC-File angezeigt, die Länge in Sektoren hat keine Bedeutung. Normalerweise wird nach Erreichen des End-Of-File die Disk automatisch ge-unload-ed, wenn du das nicht willst, muß hinter dem Filenamen ein /N (ohne Leerzeichen dazwischen) gesetzt werden. Der File-Pointer wird dann wieder zum Start-Sector gesetzt.

Arbeitsweise: Ein simulierter Directory-Sector wird bei Abfrage des Sectors #361 geliefert. Ebenfalls werden der Sector #1 (zum Density-Test) und der Sector #360 (VTOC) simuliert. Die simulierte Directory liefert einen Dummy-Sector-Pointer zum Sector #401. Dieser simulierte Datensector liefert die Sectornummer für den nächsten Datensector. Dies erklärt auch, warum beim lesen des Files die Sektoren ab #401 in der Statuszeile angezeigt werden.

Beim Versuch, auf die Disk zu schreiben, wird ein Fehler an den Atari geschickt. Wenn du also das File normal benutzen willst, mußt du es wieder in ein Atari-Disk-Image-File kopieren.

Es werden auch BOOT-Files simuliert. Willst du ein solches zum BOOTen des Atari benutzen, dann muß beim laden in SIO2PC ein "/B" angehängt werden. Es kann auch "/B" und "/N" kombiniert werden.

Es gibt eine Menge Möglichkeiten, dieses Feature zu nutzen. Wenn du z.B. nur ein PC-Modem (und keins für den Atari) hast, und du willst aus einer Mailbox Atari-Files downloaden, dann kannst du diese anschließend mit dieser Funktion auf eine Atari-Diskette kopieren.

Status Zeilen Information:

Ich habe eine Statuszeile in das Programm integriert, die an der untersten Bildschirmzeile zu sehen ist. Sie zeigt alle relevanten Informationen beim Diskettenzugriff an und kann beim auffinden von Problemen helfen.

LOC: Ein Code für die gerade ausgeführte Subroutine

CMND: SIO-Kommando des Atari (Read, Write, etc.)

LAST: Vorhergehendes Kommando

DEV: Hex-Nummer des vom Atari angesprochenen Gerätes

31 - 38 = D1 - D8

40 = Drucker

39 = Remote Control Programm

3A = FILE2PC Programm

COM: High- / Low- Status der Atari-COMmand-Leitung

ERR: Ein Error-Code von einer SIO2PC Subroutine, dieses Zeichen kann (mir) beim auffinden von Fehlern helfen (wenn ihr es mir mitteilt)

SEC#: Die aktuelle Sector-Nummer der gerade vom Atari angesprochenen Diskette

RAM: Der freie Arbeitsspeicher des PC. Er steht und fällt mit dem Laden von Disk-Image-Files als RamDisk

SPEED: N zeigt Normal-Speed 19200 Baud, H zeigt den High-Speed-Modus der verschiedenen DOS'se an

Notiz: Die Statuszeile zeigt auch die aktuellen Parameter an wenn das gerade benutzte Gerät kein SIO2PC-Gerät ist. (z.B. eine reale Floppy). Das ermöglicht es, den SIO-Bus quasi zu belauschen.

PC-Speicher-Benutzung (+++ Kurzfassung +++):

Die technische Seite der RAM-Benutzung für die RamDisk ist sehr einfach. SIO2PC nimmt die DOS-Funktion "Allocate Memory" in Anspruch und reserviert den gesamten verfügbaren konventionellen Arbeitsspeicher für sich. Das ist nicht weiter kritisch, da DOS ein Single-Tasking-System ist. Geladene TSR's (im konventionellen Speicher) zwingen SIO2PC dazu, nur den darüberliegenden Speicher zu reservieren. Für die Entscheidung, ob genug Speicher für eine zu erstellende RamDisk vorhanden ist, wird eine DOS-interne Variable benutzt.

Virus-Erkennung:

SIO2PC lädt beim Start nicht sich selbst in den Speicher sondern eine Kopie. Warum? Sollte die gestartete Kopie von einem Virus befallen werden, bleibt das Original davon verschont. Im Fall der Infizierung wird eine Fehlermeldung auf den Bildschirm gebracht und das Programm nicht gestartet. Die Fehlermeldung kommt so zustande: Zuerst wird die Länge des Programms getestet. Ist sie in Ordnung, wird noch eine Checksumme gebildet. Nur wenn beide Zahlen mit dem im Programm gespeicherten Wert übereinstimmen, startet SIO2PC ordnungsgemäß.

Problem-Diagnose:

Wenn Probleme auftauchen, solltest du zuerst die folgenden Zeilen lesen. Ist das Problem nicht zu beheben, interessiert es mich natürlich auch, wende dich an mich.

Welchen Atari kann ich benutzen? 1200? 800? 600? 400? XE? XL?

Welches Atari-DOS, welche Diskettenformate?

Welchen PC? XT? AT? 286? 386? 486?... Welchen CLock Speed?

Ist der COM-Port von anderer Ausrüstung weiter benutzbar?

Welche Version von PC-/MS-DOS?

Was tut TESTIME.EXE in den 10 Sekunden nach dem Start?

Einige Info's über den seriellen (COM) Port des PC.

Welche spezifischen Schritte sind nach auftauchen eines Problems zu tun?

Mögliche Probleme und Lösungen:

Bestimmte Diskettenlaufwerke lassen sich nicht ansprechen: Sind hier Laufwerk 3, 4, 5, 6, 7, 8 nicht ansprechbar (Error 160) so ist wahrscheinlich, das dein DOS nicht für diese Laufwerke eingerichtet ist. Bei Atari-DOS 2 und 2.5 kannst du mit BASIC booten und dann mit dem Kommando "PRINT PEEK (1802)" die dafür zuständige Zahl anzeigen lassen. Folgende Zahlen sind für die Laufwerke vorgegeben: 1 für D1, 2 für D2, 4 für D3 und 8 für D4. Um nun die Laufwerke anzumelden, mußt du die Zahlen für die

geforderten Laufwerke summieren. Für alle vier Laufwerke wäre das also die 15. Schreibe jetzt im BASIC "POKE 1802,15", dann "DOS" und mit der DOS-Funktion "H" wird das modifizierte DOS nun auf die Diskette zurückgeschrieben. Nach dem Neustart des Atari sind nun alle vier Laufwerke verfügbar.

Ein weiteres Problem tut sich hier auf: Wenn du einen 130XE mit einer realen RAMDISK (oder eine Eigenbau RAMDISK) benutzt UND ein DOS daß einen RAMDISK-Handler installiert (also diese auch initialisiert), dann mußt du natürlich daran denken, daß du in SIO2PC kein Disk-Image auf der selben Laufwerksnummer D8 benutzen kannst. Dies würde zu einer Kollision (Error) dieser beiden Laufwerke führen.

+++ Anmerkung des Übersetzers:

Wenn du die oben angegebene Zahlenfolge (1 für D1, 2 für D2, 4 für D3, 8 für D4) fortführst (16 für D5, 32 für D6, 64 für D7) kommt also 128 für D8. Ist eine RAMDISK D8 geladen, muß von "PRINT PEEK (1802)" 128 abgezogen werden, also bei zwei Laufwerken D1 und D2 folgend: $131 - 128 = 3 (1 + 2)$. Ein POKE in diese Speicherstelle mit z.B. der Rechnung $1 + 2 + 4 + 8 + 128 = 143$ (also "POKE 1802,143") funktioniert aber nicht, da die 128 für die RAMDISK D8 erst beim BOOTen von DOS selbst dazugerechnet wird.

Daraus folgt: Um DOS zu verändern, muß es ohne RAMDISK geladen werden.

Entferne dazu RAMDISK.COM von der Diskette (auf eine andere Diskette kopieren und dann löschen oder auch einfach in z.B. in RAMDISK.XXX ändern), boote den Atari neu, verändere DOS nach der obigen Beschreibung und kopiere dann wieder RAMDISK.COM auf diese Diskette. Nun noch einmal BOOTen, alles OK.

+++ Noch eine Anmerkung:

Diese Beschreibung trifft auf DOS 2.0 und 2.5 zu. Andere DOSen nutzen andere Speicherstellen für die Laufwerksnummern-Kennung. Meist ist auch die Praxis weitere Laufwerke anzumelden sehr verschieden.

+++

Ein weiterer Fehler unterläuft mir von Zeit zu Zeit: Ich "Create" eine neue RamDisk (Disk-Image) und versuche darauf zu schreiben, ohne daß ich diese vorher formatiert habe, daß geht natürlich nicht. Auch ein DOS-spezifischer Fehler passiert manchmal. Ich versuche mit der normalen Formatier-Funktion "I" eine 92k-Diskette zu formatieren, aber das DOS 2.5 formatiert hiermit nur 143k-Disketten. Hier kommt nun bei Benutzung von SIO2PC nicht immer die richtige Fehlermeldung, oder der Atari versucht es einfach nochmal, und nochmal, ...

Zur Erinnerung, für das 92k (Single) Format gibt es in DOS 2.5 eine extra Formatier-Funktion. Weiterhin werden beim benutzen einer 64k-RamDisk (Image-File, Größe Nr. 1) die volle Single-Kapazität von 707 freien Sektoren an den Atari übergeben. Beim Versuch, zu viele Daten auf diese zu kopieren, gibt's den Error 138. Die nicht verfügbaren Sektoren werden in der VTOC nicht gekennzeichnet, deshalb diese Fehlermeldung.

Das folgende Problem tritt beim Start von SIO2PC aus einer WINDOWS-Oberfläche auf. Wird also SIO2PC in einer DOS-Shell eines Multitasking-Systems gestartet, so versucht natürlich dieses System, die Kontrolle über diesen virtuellen Computer zu behalten. Dies kostet soviel Rechnerzeit, daß beim High-Speed-Datentransfer Interferenzen (Fehler) entstehen.

Datenlärm, speziell aus dem Drucker:

Ein User erzählte mir, daß sein SIO2PC-Interface-Kabel von Netzkabeln beeinflusst wird. (Ich habe viele ungeschirmte Kabel neben den Netzkabeln zu liegen -auch in der Nähe der Klima-Anlage- ohne daß Probleme auftauchen). Er sagte auch, daß er das Interface in (Grill-) Folie eingewickelt hat zum abschirmen.

Wenn du solche Probleme hast, lege einfach das SIO2PC-Kabel woanders lang, das sollte eigentlich helfen.

+++ Anmerkung des Übersetzers:

Es gibt wohl nur wenige PC's die nicht in einem metallischen Gehäuse stecken, Laptops und dergleichen einmal ausgenommen. Bei Druckern sieht das schon ganz anders aus.

+++

Ein Drucker produziert jede Menge Elektrische Wellen! Ich habe noch keine solchen Probleme gehabt. Kein Wunder, mein PC und mein Atari stehen nicht auf demselben Schreibtisch, jeder hat eine eigene Steckdose und jeder hat seine Abschirmung.

Noch ein Gedanke: stelle sicher, daß dein Druckerkabel und dein SIO2PC-Kabel nicht nebeneinander liegen. Haben dein Druckerkabel und/oder dein SIO2PC-Kabel Befestigungsschrauben am Stecker, sollten diese auch festgezogen werden.

Sollte das alles nicht helfen, gibts es noch eine Variante. Setze den SIO2PC Printmodus auf ein File, ("P" drücken, danach "F" usw., ein Filename muß eingegeben werden) und drucke jetzt den Text vom Atari in das File. Danach beendest du SIO2PC und kopierst das File mit dem DOS-Schalter für binäre Dateien zum Drucker (copy Filename.txt PRN /B). Wenn MS-DOS nicht in den binären Modus gesetzt wird, interpretiert es das erste Hex-byte 1A als File-Ende hört evtl. mitten im Text mit drucken auf.

Lock Data Line:

Der Schalter "K" in SIO2PC erlaubt das schließen des Data-Out-Kanals. Bei einigen Usern tauchte ein Problem auf, wenn der Data-Out-Kanal automatisch in den inaktiven Modus geht. Du solltest diesen Schalter auf AUTO belassen, andernfalls treten Interferenzen auf wenn du versuchst ein Nicht-SIO2PC-Gerät aufzurufen.

Wenn du merkst, daß beim Aufruf eines SIO2PC-Laufwerkes dieser stotternd oder zögernd arbeitet, versuche es mit LOCK. Ich denke aber, das Timing-Menü ist hier der bessere Weg um dieses typische Problem zu beseitigen.

Kopierschutz:

Einige Leute haben mir erzählt das Kopiergeschützte Programme die geknackt waren von einem realen Diskettenlaufwerk starten, von SIO2PC aus nicht. Einer hat mir einen Trick verraten: Er benutzte das Timing-Menü um die I/O-Rate von SIO2PC auf die eines realen Diskettenlaufwerkes herunterzusetzen. Einige Programme testen den "Sector Get" und "Sector Put" um sicherzustellen daß sie nicht von einer RAMDISK gestartet werden (ein Hinweis, daß sie geknackt worden sind).

Interferenzen von einem realen Diskettenlaufwerk:

In meiner Beschreibung empfehle ich das reale Laufwerk #1 abzuschalten und den Atari von der SIO2PC-Disk #1 zu BOOTen. Ich habe einige Beschwerden darüber gehört. Eine kürzlich geborgte Floppy 1050 erzeugte im ausgeschalteten Zustand Interferenzen. Die Lösung: die Floppy eingeschaltet lassen aber den Diskettenriegel nach oben drehen (Diskette auswerfen). In diesem Zustand antwortet die Floppy nicht auf den Aufruf des Gerätes #1.

+++ Anmerkung des Übersetzers:

Hier zeigt sich wiederum der Fleiß der Deutschen, die für die 1050-Floppy mehrere Hardware-Erweiterungen (Turbo-, Happy, Speedy-) entwickelt haben, die der Amerikaner NickAtari wohl nicht kennt. Eine 1050 mit einer solchen Erweiterung (und auch die neu entwickelte Floppy 2000) antwortet sehr wohl auf den Aufruf des Gerätes mit ihrer Kennzahl, auch wenn der Diskettenriegel offen ist. Es bleibt also im Falle des Auftretens dieser Interferenzen nur noch die letzte (und meiner Meinung nach sowieso beste) Lösung, die reale Floppy als Laufwerk D2 oder D3 oder auch D4 zu benutzen.

+++

Notiz zur Status-Zeile:

Unter bestimmten Bedingungen (wenn z.B. der Bildschirm zur Drucker-Ausgabe vom Atari benutzt wird), ist die Statuszeile nicht mehr auf dem aktuellen Stand. Im allgemeinen reicht es ESC zu drücken und der aktuelle Stand ist wieder hergestellt.

User-Fragen:

1) Kann SIO2PC mein Festplatten-Format zerstören?

- Nein, die Atari-Disk-Images werden als ganz normale PC-Files behandelt.

2) Kann ich das SIO2PC-Interface als generelles RS232-Interface benutzen, also kann ich damit ein Modem betreiben?

- Hmm... das ist ja eine gute Frage und die Antwort ist "ja und nein". Elektronisch: Ja, du hast eine echte RS232-Verbindung, aber...

...RS232 enthält mehrere Handshaking-Leitungen in beiden Richtungen. Die meisten Verbindungen nutzen nur eine davon, aber das wird von der Software bestimmt. Ich kenne Computer-zu-Computer Daten-Verbindungen die nur Data-In, Data-Out und Ground (über RS232) benutzen. Mein System hat EINE Handshaking-Leitung vom Atari zu PC, das ist die Atari-Command-Leitung die an PC's RI angeschlossen ist. Du kannst so oft du willst einen Abbruch der Verbindung vom Atari an den PC schicken, der meint dazu immer nur "Ready". Ein Null-Modem-Kabel benutzt die gleiche Praxis wie mein SIO2PC-System. Die Bit-Rate (Baud) und Software auf der Seite des Atari: das SIO-System des OS arbeitet mit einer festen Rate von 19200 Baud, bei Cassette 300. Es wird ein festes Protokoll zur Datenübertragung genutzt das folgt:

Datenblock-Acknowledge-Datenblock-Acknowledge...komplett.

Wenn du eine eigene Software schreiben willst, kannst du mit ihr den Atari POKEY-Chip (UART) direkt programmieren, wie viel Baud es sein sollen. Und du kannst direkt Daten in seinen "Sender" geben, und direkt ankommende Daten aus seinem "Empfänger" lesen.

Wenn du das bringst, kannst du das SIO2PC-Protokoll komplett an deine Bedürfnisse anpassen, auch die Handshaking-Leitungen können damit kontrolliert werden. Dazu braucht man die "Atari Technical User Notes" (Technische Dokumentation). Es macht Spaß dies zu lernen, aber es braucht Zeit. Ich habe mir ein spezielles Terminal-Programm geschrieben welches das SIO (nicht 850) Interface mit 9600 Baud konfiguriert und Verbindung zu einem Packet-Radio-Terminal-Controller mit RS232 hält.

3) Das SIO2PC-Interface ist eine Sackgasse, keine Geräte mehr Anschließbar?

- Das ist wieder so eine Frage von Nichtdenkern. Man kann doch das Interface hinter den anderen Geräten anschließen (außer bei der Cassettenstation XC12) Aber es gibt ja auch noch eine (oder zwei) Variante(n):

Die alte Cassettenstation 1010 (die ja wohl ohnehin keiner mehr benutzt) hat zwei serial-Ports. Reiß den Elektronik-Schrott raus und baue dein SIO2PC-Interface da rein. Alternativ kannst du natürlich auch den Atari selbst aufmachen und dein SIO2PC direkt auf den SIO-Ausgang aufbauen. Dann musst du nur noch einen PC-DBF-9-Ausgang schaffen und ein Kabel dranstecken.

4) Kann SIO2PC auf einem Atari ST mit dem PC kommunizieren?

- NEIN !

5) Kann man den Source-Code von SIO2PC bekommen?

- NEIN !

6) Wie wäre es mit einem default-Pfad?

- Fragt mich einer, ob man nicht einen default-Pfad angeben könne um SIO2PC mitzuteilen in welchem Verzeichnis es die Disk-Image-Files findet. Ich starte SIO2PC nicht in einem anderen Verzeichnis als dem, in dem auch die Disk-Images gespeichert sind. Wer dies unbedingt tun muß, der soll doch gefälligst das DOS dazu benutzen, SIO2PC einen Pfad zuzuweisen. DOS enthält einen Befehl "SUBST", der es erlaubt, ein simuliertes Laufwerk zu schaffen.

Beispiel: Am DOS-Prompt eingeben "SUBST E: C:\ATARI\DISKS". Danach kannst du für jede Disk die du laden willst einfach nur E: eingeben. Ist das leicht genug?

7) Direkt-Zugriff zur Festplatte

- Einige Leute wollen einen Direkt-Zugriff auf die Festplatte (wegen der Geschwindigkeitssteigerung) und Supergroße Atari-Images. Ich bin dagegen direkt in die Sektoren der Festplatte zu schreiben (wie schon mal erwähnt wird dazu eine DOS-Eigene Routine benutzt), da die Möglichkeit der Zerstörung anderer Daten zu groß ist. Ich möchte dieses Risiko nicht eingehen.

**** ERLEDIGT ****

+++ Anmerkung des Übersetzers:

Ich habe in der Übersetzung einige Passagen des Originalen Textes nicht Übersetzt, da ich sie in diesem Zusammenhang für Unwichtig halte. Wer aber Wert darauf legt, die nachfolgen aufgeführten Erläuterungen zu lesen, der muß sich halt ein Original besorgen.

+++

- Technik der DOS-Shell
- Das Shareware-Prinzip
- Bug-Geschichte von SIO2PC

+++++

SIO2PC Instruktionen von Nick Kennedy

Übersetzung von Bernhard Pahl

BIBO-DOS ist ein Produkt der Firma BIBO-Soft.

Floppy 2000 ist ein Produkt der Firma Ulf Peters

Turbo-DOS ist ein Produkt der Firma Reitershahn Computertechnik

etc. etc.