

Soundblaster-Karten

Gerhard Fischer

Die Hardware

1. Allgemeines

Jeder der seine ersten Computererfahrungen an einem Heimcomputer gemacht hat, sei es der C64, einer der Amiga, oder ein anderer Homocomputer, vermißt bei den PC's eine angenehme und für die Simulation verschiedener Animationen wichtige Eigenschaft, nämlich den Ton. Ich spreche hier nicht von dem gnadenlosen PIEP-Ton, der spätestens nach jedem Kaltstart aus dem PC-Speaker tönt, sondern von Tönen, die richtige Musikstücke erklingen lassen können.

Seit dem Beginn des Multimedia-Zeitalters gibt es auch für die PC mehrere Möglichkeiten zu musizieren. Es befinden sich die verschiedensten Arten von Sound-Karten auf dem Markt, die je nach Preisklasse mehr oder weniger zu leisten in stande sind. Der Preis der Sound-Karten sinkt nebenbei immer mehr und mehr ab, so daß eine Sound-Karte mit minimaler Konfiguration, die damals um die 5000 ÖS gekostet hat, heute mit einer wesentlich besseren Ausstattung (Digitaler Kanal, viel mehr FM-Stimmen) schon unter 3000 ÖS zu erwerben ist. Dies ist natürlich auf den Wettbewerb am Sound-Karten-Markt zurückzuführen. Somit ist schon gesagt, daß es beim Kauf einer Sound-Karte aus einer Vielzahl von Sound-Karten zu wählen gilt. Als nächstes folgt eine Auflistung der verschiedensten Sound-Karten.

Roland-LAPC-1

128 verfügbare Sounds (nachträglich veränderbar), 30 Drums, 33 Natur-samples, Anschluß für MIDI-Box, Stereo-Kopfhörer-Anschluß, heute noch ca.10000ÖS.(Fa. Roland)

IBM-Music-Feature-Card (1987)

Bis 8-stimmigen Stereo-Synthesizer, komplette MIDI-Schnittstelle, 24 vorprogrammierte Sounds, Yamaha Soundchip YM-2164, bis zu 4 Sound-Karten in einem System (somit bis 32 Stimmen), Preis ca.9000ÖS. (Fa. IBM)

Game-Blaster (C/MS-Card)

12 kanäligen Stereosound, erzeugt mittels Amplitudenmodulation (schlecht Qualität), für Hobbybereich entwickelt, jedoch war die Zeit für solche Karten noch nicht reif und so verschwand die Karte wieder vom Hardware-Markt. Preis ca 4000 ÖS. (Fa Creative Labs)

AdLib

11 Synthesizer (entweder 9 Melodien-Stimmen oder 6 Melodien-Stimmen und 5 Schlagzeug-instrumente) ,kompartibel zu Roland und IMF-Card, Preis ca.4000 ÖS. (Fa. AdLib=> Kanada)

Sound-Blaster 1.0 (1989)

Vollständig kompartibel zur Game-Blaster und AdLib, 11-stimmiger Mono-Sound, C/MS-Chipsatz, ´digitaler´ Tonkanal mit einer Aufnahme-Samplerrate von 4-15 kHz und Abspiel-Samplerrate von 4-24 kHz, insgesamt 24 Stimmen (11 AdLib, 12 Game-Blaster + 1 DAC), Anschluß für analogen Joystick, MIDI-Schnittstelle, Preis ca. 2000 ÖS. (Fa. Creative Labs)

Sound-Blaster 1.5

Wie Version 1.0 nur ohne C/MS-Chipsatz, daher billiger, aber 12 Game-Blaster kompartible Stimmen weniger, C/MS Chipsatz nachrüstbar, andere Software im Lieferumfang.

Sound-Blaster 2.0

Wie Version 1.5, und C/MS Chipsatz nachrüstbar, 11-stimmiger FM-Sound / Mono und ein digitaler Kanal mit gleicher Aufnahme-Samplerrate von 4-15 kHz aber einer Abspiel-Samplerrate von 4-44,1 kHz (CD-Qualität), jedoch Mono, getrennte Audioeingänge für Mikrofon und Line-In-Signaleingang, Mikrophoneingang passt sich automatisch der Impedanz des vcrwendeten Mikrophons an. (Fa. Creative Labs)

Sound-Blaster-Pro 1.0

benötigt 16-Bit-Steckplatz (statt bisherige Soundkarten mit 8-Bit-Steckplatz), 2x11 Stimmen FM-Sound / Stereo, MIDI-Schnittstelle, Joystickanschluß, Anschluß für CD-Rom-Laufwerk, digitaler Soundkanal mit Stereo-Modus (2x4 kHz bios 2x22.05 kHz, oder 4-44,1 kHz Mono),

per Software kann Lautstärkeregelung der externen Quellen am Eingang, sowie der Gesamtlautstärke erfolgen. (Fa. Creative Labs)

Sound-Blaster-Pro 2.0

Wie Pro 1.0 nur mit neuen OPL3-Chips (leistungsfähiger), verfügt über 20 FM-Sound-Stereo-Stimmen. (Fa. Creative Labs)

Sound Blaster 16

Wichtigste Erneuerung ist die Erweiterung des 8-Bit DAC zu einem 16-Bit DAC, somit ist ein neuer Masstab in Sachen Samplingrate gesetzt, 2x4 kHz bis 2x 44.1 kHz für die Aufnahme- und Abspiel-Samplerrate, ASP-Chip (Advanced Signal Processing) für Datenkompression und Spracherkennung ohne Rechnerbelegung. (Fa. Creative Labs)

Einige Firmen kündigten bereits neue Sound-Karten an, wie z.B. die AdLib Gold 3000 als Nachfolger der AdLib Gold 1000, oder Microsoft mit dem Windows-Sound-System, welches das Sprach-Eingabe-/Ausgabesystem unter Windows verwirklichen soll. Dies ist jedoch zur Zeit noch Zukunftsmusik und wird erst in geraumer Zeit zu erwerben sein.

In den nachfolgenden Kapiteln werden die Angaben immer auf die Sound-Blaster-Pro 2.0 bezogen, außer es wird ein anderer Kartentyp angeführt.

2. Hardware-Einstellungen

Es gibt auf den Sound-Blaster-Karten verschiedene Einstellungen die getätigt und überprüft werden müssen bevor die Karte in Betrieb genommen wird. Diese Einstellungen betreffen die Portadressen, die Interruptnummer, den DMA-Kanal sowie die Joystickaktivierung. Beim Kauf der Karte sind folgende Voreinstellungen vorgenommen worden. Portadresse=220HEX, Interruptnummer=7, DMA-Kanal=1 und Joystick ist aktiviert. Werden noch andere Steckkarten verwendet, wie z.B. einen Netzwerk- oder Scannerkarte, so kann es zu Konflikten zwischen den Karten kommen. Diese Konflikte führen meistens zum Absturz des Rechners und müssen deshalb beseitigt werden. Es kann zu Konflikten kommen wenn z.B. der selbe DMA-Kanal, die gleiche Portadresse, oder ein Interrupt, von zwei oder mehreren Karten benutzt wird.

2.1 Einstellmöglichkeiten der Portadresse

Da die Sound-Blaster-Pro (SBP) und die SB2.0 einen größeren Speicherplatz bei der Portadresse benötigen (zB. 220h bis 233h) als ihre Vorgängermodelle(zB. 220h bis 22Eh) kann nur mehr zwischen 2 mögliche Adressen gewählt werden.

Die Portadressen-Jumper-Möglichkeiten	
SB 1.0/1.5	SB2.0/SBP
220h	220h
230h	---
240h	240h
250h	---
260h	---

2.2 Einstellmöglichkeiten der Interruptnummer

Es gibt sehr viele verschiedene Interrupts, von denen für uns allerdings nur die von 0 bis 10 von Bedeutung sind.

Interrupt 0	System Timer
Interrupt 1	Tastatur
Interrupt 2	frei
Interrupt 3	COM2
Interrupt 4	COM1
Interrupt 5	frei

Interrupt 6	Disketten Controller
Interrupt 7	LPT1(unter OS/2), unter DOS normal frei
Interrupt 8	Uhr / Kalender
Interrupt 9	frei
Interrupt 10	frei

Die Sound-Karten können von den oben gezeigten Interruptnummern folgende benutzen:

Die Interrupt-Jumper-Möglichkeiten	
SB 1.0-2.0	SBP
IRQ 2	IRQ 2
IRQ 3	IRQ 5
IRQ 5	IRQ 7
IRQ 7	IRQ 10

2.3 Einstellmöglichkeiten des DMA-Kanals

Bei den SB1.0 -2.0 konnte der DMA-Zugriff nur aktiviert oder deaktiviert werden. Bei der SBP kann zwischen drei verschiedenen DMA-Kanälen gewählt werden.

SBP
0
1
3

Wo und wie die einzelnen Jumper für die gewünschte Einstellung gesetzt wird entnehmen Sie bitte aus ihrem Sound-Blaster-Handbuch, da dort alle wichtigen Zusatzinformationen enthalten sind.

3. Tonausgabe

Die Ausgabe der gewünschten Töne oder Geräusche kann bei den Sound-Blaster-Karten auf sehr vielseitige Weise geschehen. Die Karten haben alle einen 4-wattigen Verstärker bei 40hm Widerstand, oder 2Watt bei 80hm Widerstand, eingebaut. Somit können bereits kleine Boxen damit betrieben werden. Wünscht man aber eine bessere und vor allem lautere Tonwiedergabe, so sollte der Audioausgang mit einem HI-FI-Verstärker verbunden werden und die Tonausgabe über die Boxen dieser Anlage erfolgen. An dem Audioausgang der Sound-Blaster-Karten kann natürlich auch ein Kopfhörer angeschlossen werden.

4. Setzen der Umgebungsvariablen

Bei der Programmierung der Sound-Blaster-Karten kann man auf zwei Umgebungsvariablen zurückgreifen, welche im Normalfall in der Autoexec.bat mit dem Befehl SET vereinbart wurden. Es ist aber auch möglich, innerhalb von Programmen Umgebungsvariablen zu setzen oder zu löschen. Die erste Umgebungsvariable des Sound-Blasters heißt SOUND und wird z.B. in der AUTOEXEC.BAT vereinbart. In dieser Variablen steht wo die Soundtreiber zu finden sind. Der Befehl könnte z.B. so lauten:

SET SOUND = C:\SBPRO.

Die Umgebungsvariable BLASTER enthält Informationen über die Einstellung der Soundkarte und kann ebenfalls gleich in der AUTOEXEC.BAT vereinbart werden. Diese Vereinbarungen müssen mit den Einstellungen der Hardware, also mit den Jumperstellungen auf der Karte identisch sein. Die Befehlszeile kann z.B. wie folgt aussehen:

SET BLASTER = A220 I7 D1 T4

Mit diesem Befehl werden die Portadresse (A220), die Interruptnummer (I7), der DMA-Kanal (D1) sowie die Kennnummer der Sound-Blaster-Karte (T4) gesetzt.

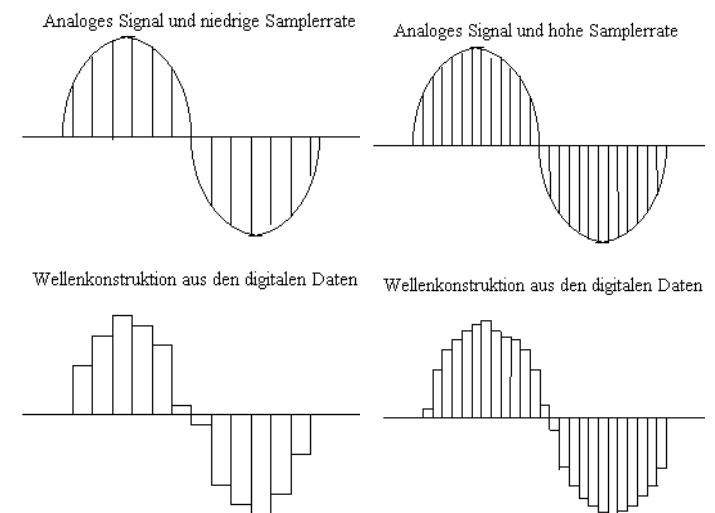
Kennnummer T	Sound-Blaster-Version
1	Sound-Blaster 1.0 / 1.5
2	Sound-Blaster Pro
3	Sound-Blaster 2.0 / 2.5
4	Sound-Blaster Pro2 /Pro3/ Pro4
5	Sound-Blaster Pro (Microchannel)

Nach diesen Einstellungen, sollte das mitgelieferte Testprogramm ausgeführt werden um etwaige Probleme zu erkennen und zu beheben. Wie solche Probleme behoben werden entnehmen Sie am besten aus dem ebenfalls mitgelieferten Benutzerhandbuch.

Die Digitale Tonausgabe der Sound-Blaster-Pro

Wenn der Aufwand der synthetischen Tonerzeugung zu groß wird bleibt immer noch die Möglichkeit des Digitalisierens. Beim Digitalisieren wird ein Ton/ Geräusch/ Musik aufgenommen und kann dann weiterverarbeitet werden. Der Vorgang des Digitalisierens wird auch SAMPLING genannt. Beim Sampling wird ein analoges Signal in ein digitales Signal mittels ADC (Analog-Digital-Converter = Analog-Digital-Umsetzer (ADU)) umgewandelt. Der ADC tastet das analoge Signal ab und wandelt diese in Zahlen um. Die Abtastfrequenz (Sample-Rate) besagt, wie oft das analoge Signal in einer Sekunde abgetastet wird (Sample-Rate deshalb in Hz). Bei einer Sample-Rate von 8000 Hz tastet der ADC das analoge Signal 8000 mal in einer Sekunde ab und speichert jede Abtastung in 8 Bit ab (also 8000 Byte). Wie man sieht ist diese Methode der Musikerzeugung oder -speicherung sehr speicherintensiv. Dazu ein kleines Beispiel. Es soll 1 Minute lang ein Musikstück in CD-Qualität und Stereo aufgenommen werden => $2 \cdot 44,1 \text{ kHz} \cdot 60 \text{ sek} = 5\,292\,000 \text{ Byte} \approx 5 \text{ MB}$ Speicher für 1 min.

Je höher die Sample-Rate desto näher liegt das analoge Signal beim Digitalen, also umso besser ist die Qualität der Aufnahme, aber umso mehr Speicher wird benötigt. Es muß daher je nach Verwendungszweck entschieden werden ob die Qualität oder der Speicherplatz Vorrang hat.



Es ist mit verschiedener Software möglich die Sample-Rate zu verändern, also daß ein Musikstück, welches mit 8 kHz aufgenommen wurde mit z.B. 24 kHz abgespielt wird. Diese Veränderung der Sample-Rate hat außer einer Geschwindigkeitszunahme durch das schnellere Abspielen auch noch einen Tonhöhenveränderung zur Folge. Mit steigender Frequenz, werden die abgespielten Töne höher mit sinkender Frequenz gegenüber der Aufnahmefrequenz werden sie tiefer.

Im zweiten Teil der Referat-Triologie wurde erklärt wie man die FM-Kanäle programmiert. Auf genau dieselbe Art wird der digitale Kanal angesprochen. Es gibt wieder einen Treiber mit dessen Hilfe der digitale Kanal von der DOS-Oberfläche bedient werden kann. Dieser Treiber heißt CT-VOICE.DRV. Creative Labs entwickelte für die Sound-Blaster-Karten ein eigenes Format, welches für die digitalisierten Sounds eingesetzt wird => CT-VOICE-Format. Alle Dateien mit der Endung/Erweiterung .VOC passieren auf diesen Format.

Bei der Programmierung des digitalen Kanals ist es sehr sinnvoll auf dieses Format zurückzugreifen, da es sehr flexibel und komfortabel ist. Eine VOC-Datei läßt sich in 2 Teile teilen. In den Vorspann (Header), der die Datei als CT-Format zu erkennen gibt und in die Daten. Diese Daten enthalten verschiedene Texte, Versionsnummern, Adressen und Längen, sowie 8 Datenblöcke. Diese 8 Datenblöcke sind =>

BLOCK 0 - End Block

Wenn so ein Block erreicht wird, endet die Ausgabe der VOC-Datei.

BLOCK 1 - New Voice-Block

Enthält Samplerrate von der Aufnahme, Informationen gepackte Dateien und Sampledaten.

BLOCK 2 - Subsequent Voice-Block

Dient nur zum aufteilen größerer Sampledaten. Darf nur nach einem Blocktyp 1 kommen, da keine Informationen über Samplerate etc,... kit gegeben werden.

BLOCK 3 - Silence-Block

Erzeugt in einer VOC-Datei Stille (viel Stille mit wenig Speicher).

BLOCK 4 - Marker-Block

Ermöglicht das Setzen von Marker, die auf den Musikablauf keinen Einfluss haben, aber von einem Programm aus abgefragt werden können um z.B. für einen Graphik-Ton-Synchronisation eingesetzt werden zu können.

BLOCK 5 - Message-Block

Für das Einfügen von ASCII-Text mitten in eine VOC-Datei.

BLOCK 6 - Repeat-Block

Wird benötigt um VOC-Sequenzen beliebig oft zu wiederholen.

BLOCK 7 - Repeat-End-Block

Gibt an das von Block 6 bis Block 7 alle dazwischenliegende Datenblöcke wiederholt werden sollen. Verschachtelungen von Schleifen sind allerdings verboten.

Der CT-VOICE-TREIBER

Dieser Treiber wird mit der Sound-Blaster-Karte mitgeliefert und ist ein sehr nützliches Werkzeug, da er sich sehr leicht in eigene Programme einbauen läßt (z.B.: C++, Turbo-Pascal). Alle Treiberfunktionen sind auf das CT-Voice-Format zugeschnitten. Der Treiber muß vor der Benutzung geladen werden. Alle Parameter die an den Treiber übergeben werden sollen, werden mit Hilfe von Prozessor-Register ausgetauscht. Auf die gleich Weise erhält man alle Ergebniswerte zurück. Das wichtigste Register beim Aufruf ist das BX-Register, da mit im die Funktion des Treibers gewählt wird.

Die Funktionen sind oft fast identisch mit den Funktionen des FM-Treibers => SBFMDRV.COM.

Funktion 0

Ermittlung der Treiberversionsnummer des im Speicher installierten SBFMDRV. Die Haupt- und Unternummer steht im AX-Register.

Eingabe = BX=00
Ausgabe = AH=Hauptnummer
AL=Unternummer

Funktion 1

Setze Port I/O-Adresse.

Eingabe = BX=01
AX = Portadresse

Funktion 2

Setze Interrup.

Eingabe = BX=02
AX = Interruptnummer

Funktion 3

Initialisiere Treiber.

Eingabe = BX=03
AX=0 Erfolg
AX=1 SB nicht gefunden
AX=2 Portadressenfehler
AX=3 Interruptfehler

Funktion 4

Lautsprecher ein / aus

Eingabe = BX=04
AL=0 ausschalten
AL=1 einschalten

Funktion 5

Setzt Status-Byte-Adresse. Für die Dauer des Abspielens einer VOC-Datei wird der Wert des Status-Bytes ungleich 0. Am Ende des Stückes nimmt das Statusbyte wieder den Wert 0 an.

Eingabe = BX=05
ES:DI=Statusadresse

Funktion 6

Sample abspielen. Im Registerpaar ES:DI wird die Segment:Offsetadresse des Beginns der Musikdaten übergeben.

Eingabe = BX=06
ES:DI= Adresse der Sampledaten

Funktion 7

Aufnahme eines Samples. Diese Funktion ermöglichtes, digitalen Sound über den DMA-Kanal in den Speicher zu übernehmen.

Eingabe = BX=07
AX=Samplerrate
DX:CX=Länge
ES:DI=Sampleadresse

Funktion 8

Abbrechen der Samplebearbeitung.

Eingabe = BX=08

Funktion 9

Treiber De-Instalation.

Eingabe = BX=09

Funktion 10

Sampleausgabe anhalten.

Eingabe = BX=10
Ausgabe = AX=0, kein Fehler

AX=1, es wurde vorher kein Sample gestartet.

Funktion 10

Sampleausgabe anhalten.

Eingabe = BX=10
Ausgabe = AX=0, kein Fehler

AX=1, es wurde vorher kein Sample gestartet.

Funktion 11

Sampleausgabe fortsetzen.

Eingabe = BX=11
 AX = 0 Erfolg
 AX = 1 kein Erfolg

Funktion 12

Unterbreche Ausgabeschleife.

Eingabe = BX=12
 AX = 0 Am Ende
 AX = 1 Sofort
 Ausgabe = AX=0 Erfolg
 AX=1 kein Erfolg

Funktion 13

Setze Benutzer-Funktion. Wenn das Programm solche Nachrichten verarbeiten soll, so muß im Registerpaar DX:AX die Segment:Offsetadresse der eigenen Routine übergeben werden, die dann jedesmal bei einer System-Exklusive-Message aufgerufen wird. Im Registerpaar ES:DI findet die Routine dann die Adresse des ersten Daten-Bytes der System-Exklusive-Message. Es muß darauf geachtet werden, daß alle Registerinhalte gesichert werden, und daß die Routine durch ein FAR-RET abgeschlossen wird.

Eingabe = BX=13
 DX:AX = Adresse der Benutzer-Routine
 Ausgabe = ES:BX=Adresse auf aktuellen Block

Damit die Ansteuerung des digitalen Kanals auch unter einem C-Programm oder unter einem Turbo-Pascal-Programm möglich ist, muß eine Softwareschnittstelle programmiert werden, oder man nimmt eine fertige Schnittstelle wie sie von verschiedenen Firmen angeboten werden. Diese Schnittstelle bindet man in sein Quellprogramm ein und kann damit bequem den digitalen Kanal der Sound-Blaster-Karte ansprechen. Diese Module enthalten außerdem

meistens gleich eine Fehlererkennungsroutine mit integrierter Textausgabe unter DOS.

Die FM-Stimmen

Die komplette Hardwareerklärung der Sound-Karten wie z.B. die verschiedensten Arten, oder Versionen der Sound-Karten wurde im ersten Teil der Sound-Blaster-Referat-Triologie bereits ausführlich erklärt. Aus diesem Grunde wird hier nicht näher auf dieses Gebiet eingegangen. Im Gegenteil, denn in diesem Teil soll nur über die FM-Stimmen der Sound-Blaster-Pro erzählt werden. (FM = Frequenzmodulation) Es wird vorausgesetzt, daß das Prinzip der FM selbstverständlich ist und deshalb nicht erklärt werden muß.

1. Allgemeines

Damit man seine eigenen Instrumentenstimmen programmieren kann, muß man sich ein wenig mit dem FM-Chip befassen. Dieser FM-Synthesizer-Chip verfügt über 18 sogenannte Operatoren, von denen immer 2 benötigt werden um eine Instrumentenstimme zu erzeugen. Für die Erzeugung von Schlagzeuggeräuschen, Drums etc., wird jedoch immer nur 1 Operator benötigt. Aus diesen Grundbedingungen ergaben sich 3 FM-Betriebsarten.

- 1 FM-Betriebsart 9 verschiedene Melodien-Stimmen (kein Schlagzeug).
- 2 FM-Betriebsart 6 verschiedene Melodien-Stimmen und 5 Schlagzeug-Geräusche.
- 3 FM-Betriebsart "Speech Synthesis" 18 Operatoren für komplexe Schwingungswelle zusammenschalten. (wird normalerweise nicht verwendet, da sich diese Funktion mit dem digitalen Tonkanal wesentlich besser erfüllen läßt.

2. Operator

Wie oben ersichtlich wurde hängt bei der synthetischen Tonerzeugung alles von den Operatoren ab. Ein solcher Operator soll nun genauer betrachtet werden. Er läßt sich in 2 grundlegende Elemente teilen. Dies

ist der Phasen-Modulator und der Hüllkurven-Generator. Ein Ton ist hauptsächlich von 2Größen abhängig, zum Einen von der Frequenz und zum Anderen von der Amplitude. Die Frequenz ist für die Höhe des Tones zuständig und die Amplitude natürlich für die Lautstärke des Tones. Im folgendem Bild sehen wir den Phasen-Modulator, der für die Frequenz zuständig ist und den Hüllkurven-Generator, welcher die Aufgabe des Lautstärkenverlaufes übernimmt.

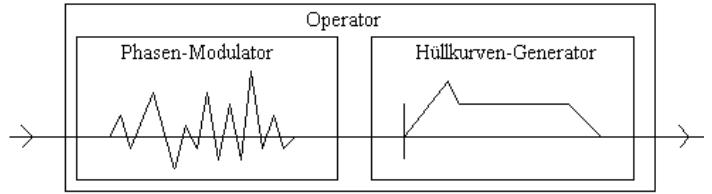
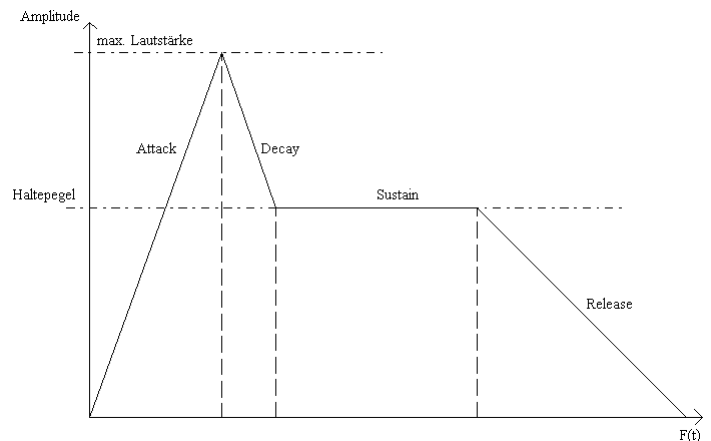


Abb.1 zeigt die vereinfachte Darstellung eines Operators.

3. Hüllkurve

Als nächstes sehen wir uns die Hüllkurve genauer an. Eine Hüllkurve hüllt die Frequenz eines Tones bezüglich ihrer Amplitude ein. Ein Ton eines Klaviers ist ein gutes Beispiel um so eine Hüllkurve zu erklären. Der Ton klingt rasch an, bis zu seiner maximalen Lautstärke (Attack), fällt dann ein bisschen ab (Decay), klingt für sehr kurze Zeit auf einer gleichbleibenden Lautstärke (Sustain) und klingt anschließend langsam aus (Release). Durch das betätigen des Sustain-Pedals kann der Ton noch länger angehalten werden. Das folgende Bild zeigt die Beeinflussung der Amplitude mittels Hüllkurve.



ATTACK (Anstiegszeit)

Gibt die Zeit an, die der Ton benötigt um die volle Lautstärke zu erreichen.

DECAY (Ausklangphase)

Gibt die Zeit an die der ton braucht um von der vollen Lautstärke bis zur Haltelautstärke abzufallen.

SUSTAIN (Haltepegel)

Gibt die Lautstärke an auf der der Ton gehalten wird solange er nicht beginnt endgültig auszuklingen.

RELEASE (2te Ausklangphase)

gibt die Zeit an die der Ton benötigt um vom Haltepegel bis zur völligen Verstummung braucht.

Aus den Anfangsbuchstaben dieser vier Begriffe setzt sich der meistverwendete Namen dieser Hüllkurvengenerators zusammen. Er wird ADSR-Generator genannt.

4. Prinzip der FM-Synthese

Wie schon erwähnt werden für Töne immer 2 Operatoren zusammenschalten. Schaltet man sie so, daß das Ausgangssignal des ersten Operators zum Eingangssignal des Zweiten wird, so übernimmt der Erste die Funktion des Modulators und der zweite Operator die Funktion des Trägers. Wobei der Träger für die Grundfrequenz und der Modulator für die Obertöne zuständig ist. Die Obertöne sind ein sehr wichtiger Teil der Klangfarbe eines Tones, während für die Charakteristik eines Tones der Lautstärkeverlauf und die Frequenz zuständig sind. Führt man das Ausgangssignal der beiden Operatoren an den Eingang des Modulators

zurück, so kann ein sehr komplexer Klang erzeugt werden. Die Abb.3 zeigt eine solche Operatorschaltung für eine FM-Synthese.

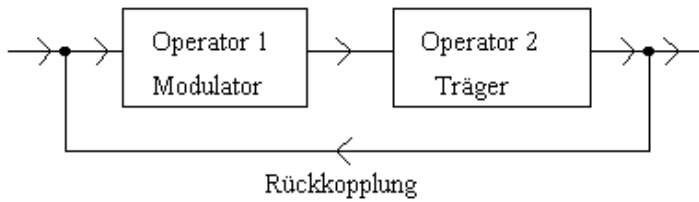


Abb.3: Operatorschaltung bei der FM-Synthese

Eine weitere Art für eine Operatorschaltung wäre die der Additiv-Synthese. Dabei werden die Operatoren parallel geschaltet und das Ausgangssignal an den Eingang des einen Operators zurückgeführt. Mit dieser Schaltungsart lassen sich nicht so komplexe Klänge erzeugen wie mit der FM-Synthese, da es keinen Modulator gibt, der die Obertöne erzeugt. Im folgendem Bild (Abb.4) ist eine Schaltung von der Additiv-Synthese aufgezeichnet.

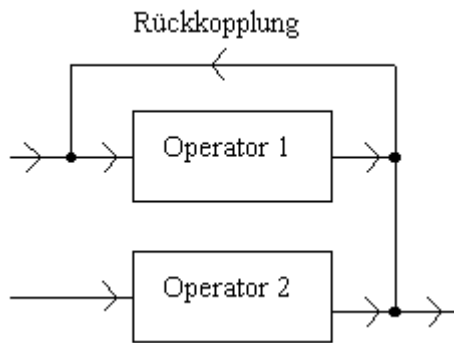


Abb.4.

5. Formate

Es gibt viele verschiedene Formate, die je nach Softwarefirma von einander variieren. Ein Format von der Fa. Creative Labs ist das SBI-Format und steht für Sound Blaster Instrument-Format. Diese Format enthält alle notwendigen Parameter und Informationen für die Register der Operatoren um Instrumentenstimmen zu erzeugen. Jedes SBI ist hex 33 (51) Bytes lang. Diese Bytes haben eine bestimmte Reihenfolge und beinhalten alle Informationen von der Lautstärke des Trägers, des Modulators, Attack, Decay, Sustain und Release von Träger und Modulator, etc.,.....

Ein weiteres Format von Creative Labs ist das CMF-Musikformat. Diese sind spezielle Musikdateien für die SB-Karten und werden mit der Erweiterung .CMF versehen. Eine CMF-Datei teilt sich in drei Blöcke. Diese Blöcke sind der Vorspann auch Header genannt, der Instrumentenblock und der Musikblock.

Im Vorspann der CMF-Datei stehen viele Informationen über den Aufbau der Datei und ist deshalb je nach Inhalt verschieden lang.

Ab dem Byte 28h kann daher schon z.B. der Text oder Autorennamen stehen. Nach diesen Texten folgt der Instrumentenblock. Jedes Instrument ist 16 Byte lang (entsprechen den 16 Bytes ab dem Byte 24h aus dem SBI-Format).

6. Treiber

Der Treiber für die FM-Stimmen von der SB-Karte heißt SBFMDRV.COM. Dieser Treiber muß erst einmal geladen werden um als TSR-Programm resident im Speicher zu stehen. Ist dies erledigt, können geeignete Programme auf diesen Treiber zugreifen, indem der Syteminterrupt aufgerufen wird, indem sich der Treiber installiert hat. Für das Ansteuern des Treibers aus einem z.B. C-Programm (damit eine Musik im Hintergrund spielt etc.) sollten die Funktionen des Treibers bekannt sein. Die Nummer der gewünschten Funktion wird vor dem Aufruf des Interrupts in das BX-Register geladen.

Der FM-Treiber von Creative Labs hat 12 Funktionen (Funktion 0 bis 11). Mit diesen Funktionen kann man nun arbeiten (aus verschiedenen Programmen => C++,etc...), also z.B. für das Ermitteln der Treiberversionsnummer, das Starten der Musik und viele andere wichtige Funktionen. Nachfolgend sind die Funktionen kurz angeführt und erklärt.

Funktion 0

Ermittlung der Treiberversionsnummer des im Speicher installierten SBFMDRV. Die Haupt- und Unternummer steht im AX-Register.

Eingabe = BX=00
Ausgabe = AH=Hauptnummer
AL=Unternummer

Funktion 1

Setze Status-Byte-Adresse auf Für die Dauer des Abspielens einer CMF-Datei wird der Wert des Status-Bytes ungleich 0. Am Ende des Stückes nimmt das Statusbyte wieder den Wert 0 an.

Eingabe = BX=01
DX:AX = Statusadresse

Funktion 2

Setzen der Instrumententabelle. Dient für das setzen der FM-Chip-Parameter zum Abspielen einer CMF-Datei.

Eingabe = BX=02
CX=Anzahl der Instrumente
DX:AX=Adresse des Anfangs der Instrumententabelle

Funktion 3

Setzen der Sytem-Clock-Rate (Timer0). Dazu teilt man die Zahl 1193180 durch den gewünschten Takt und schreibt das Ergebnis in das Register AX. Defaultwert=18,2 Hz.

Eingabe = BX=03
AX=1193189/System-Clock-Rate

Funktion 4

Setzen der Treiber-Clock-Rate. Gibt an auf welche Frequenz der System-Timer0 gesetzt werden soll, wenn ein Musikstück beginnt. Setzen erfolgt wie bei Funktion3.

Eingabe = BX=04
AX=1193189/ Treiber Clock-Rate

Funktion 5

Dient zum Transponieren aller Noten in einem Musikstück. Dies kann in beliebige Halbtonschritte vorgenommen werden.

Eingabe = BX=05
AX=Halbtonschritte

Funktion 6

Starte Musik. Im Registerpaar DX:AX wird die Segment:Offsetadresse des Beginns der Musikdaten übergeben. Im Register AX wird eine Null zurückgeliefert wenn kein Fehler aufgetreten ist, sonst eine 1. Diese Routine setzt das Statusbyte auf FFh, ändert die System-Clock-Rate in die gewünschte Treiber-Clock-Rate um und startet die Musikausgabe mit Hilfe des Timer-Interrupts.

Eingabe = BX=06
DX:AX= Adresse der Musikdaten
Ausgabe = AX=0, kein Fehler
AX=1, es läuft bereits ein Musikstück

Funktion 7

Stoppt die Musik, setzt den Timer0 auf die eingestellte System-Clock-Rate, setzt den Timer-Interrupt zurückauf die ursprüngliche Routine.

Eingabe = BX=07
Ausgabe = AX=0 kein Fehler
AX=1 es befand sich kein Musikstück in

der Ausgabe

Funktion 8

Initialisiert Treiber. Schaltet FM-Chip aus, setzt Instrumenten-Definitionen auf Anfangswerte.

Eingabe = BX=08

Ausgabe = AX=0, kein Fehler

AX=1 es befindet sich noch ein Musik-

stück in der Ausgabe

Funktion 9

Halte Musik an. Das Statusbyte wird nicht verändert.

Eingabe = BX=09

Ausgabe = AX=0, kein Fehler

AX=1, es befindet sich kein Musikstück in

der Ausgabe

Funktion 10

Setze Musik fort. Das angehaltene Musikstück kann wieder fortgesetzt werden.

Eingabe = BX=10

Ausgabe = AX=0, kein Fehler

AX=1, es wurde vorher kein Musikstück

angehalten.

Funktion 11

Setze Benutzer-Funktion. Da das CMF-Format den Standard-MIDI-Format entspricht können auch System-Exklusive-Messages verarbeitet werden. Wenn das Programm solche Nachrichten verarbeiten soll, so muß im Registerpaar DX:AX die Segmenz:Offsetadresse der eigenen Routine übergeben werden, die dann jedesmal bei einer System-Exklusive-Message aufgerufen wird. Im Registerpaar ES:DI findet die Routine dann die Adresse des ersten Daten-Bytes der System-Exklusive-Message. Es muß darauf geachtet werden, daß alle Registerinhalte gesichert werden, und daß die Routine durch ein RETF beendet wird.

Die Routine kann wieder ausgeschaltet werden, wenn in das Register AX und DX eine Null nachgeladen wird, und die Funktion 11 noch einmal aufgerufen wird.

Eingabe = BX=11

DX:AX = Adresse der Benutzer-Routine

7. Programmierung der FM-Stimmen mit BORLAND-C++ 3.0

Die Software-Schnittstelle für die Programmierung des synthetischen Kanals muß man sich selbst programmieren, oder man verwendet eine fertige Schnittstelle wie die Headerdatei CMFTOOL.H. Dieses Modul enthält alle Funktionen (außer die Funktion 11, weil zu speziell) des Treibers und kann vom C-Programm jederzeit benutzt werden. Solche Module gibt es für die verschiedensten Programmiersprachen wie z.B. Turbo-Pascal 6.0.

Anhang**Die Unterschiede der Formate VOC und WAVE**

Es gibt einige Unterschiede zwischen den Format VOC und WAVE, welche in diesem Anhang erläutert werden sollen. Das VOC-Format arbeitet auf der DOS-Oberfläche und kann daher von allen Programmen angesprochen, verwendet werden. Der Name VOC steht für 'Creative Voice File Format' und ist ein Produkt der Fa. Creative Labs. Das WAVE-Format hingegen wurde von der Fa. Microsoft entwickelt und soll für die Tonausgabe unter Windows dienen. Da es sich um verschiedene Formate handelt ist es verständlich, daß sich der Aufbau der Dateien unterschiedlich gestaltet. Aus diesen Unterschieden ergeben sich die Vor- und Nachteile des jeweiligen Formates. Die Erweiterung von Dateien die im VOC-Format abgespeichert wurden ist .VOC bei WAVE-Dateien ist es .WAV.

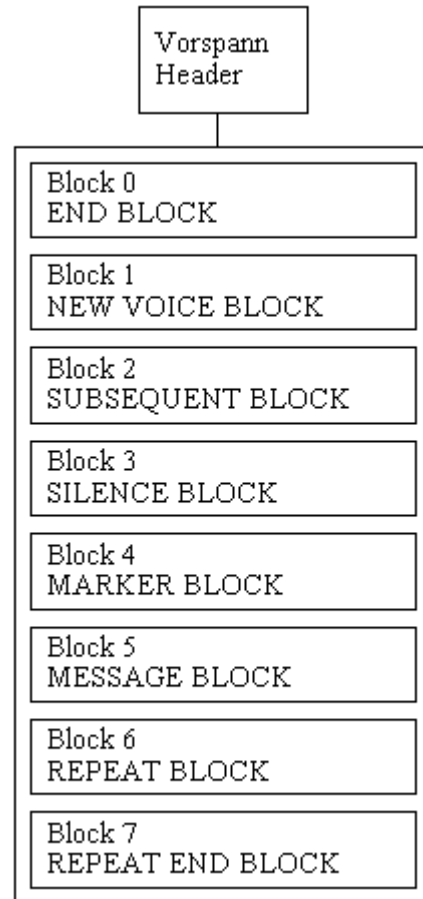
1. Prinzipieller Aufbau einer VOC-Datei

Abb.1

Jede VOC-Datei setzt sich im Prinzip aus dem Vorspann "HEADER" und den anschließenden Datenteil zusammen. Im Vorspann sind Informationen wie z.B. die Versionsnr. des CT-VOICE-Formates, die Adresse der Sampledaten sowie der Text "Creative Voice File", welcher zur Erkennung der VOC-Datei gilt, zusammen. Der Datenteil besteht aus verschiedenen Blöcken wie in der Abb.1 deutlich zu sehen ist. Die Reihenfolge dieser Blöcke ist nicht identisch mit den Blocknummern. Der Block 0 ist ausschließlich nur am Ende des Datenteiles einzusetzen, da nach diesem Block die Tonausgabe gestoppt wird. Im Block 1 stehen alle Informationen die für das Abspielen von Bedeutung sind (Länge, Samplerate, Pack-Information und dann die Sampledaten selbst). Alle weiteren Blöcke können im dritten Teil (Die Digitale Tonausgabe der Sound-Blaster-Pro) nachgelesen werden.

Da der Block 1 beliebig oft (⇒keine Grenze bekannt) in einer VOC-Datei verwendet werden darf, kann eine VOC-Datei beliebig lang sein. Dies ist bereits ein Vorteil des VOC Formates. Ein großer Vorteil des WAVE-Formates ist, daß die Tonausgabe auf jeder beliebigen Soundkarte erfolgen kann, sofern sie von Windows unterstützt wird (Treiberproblem).

2. Prinzipieller Aufbau einer WAVE-Datei

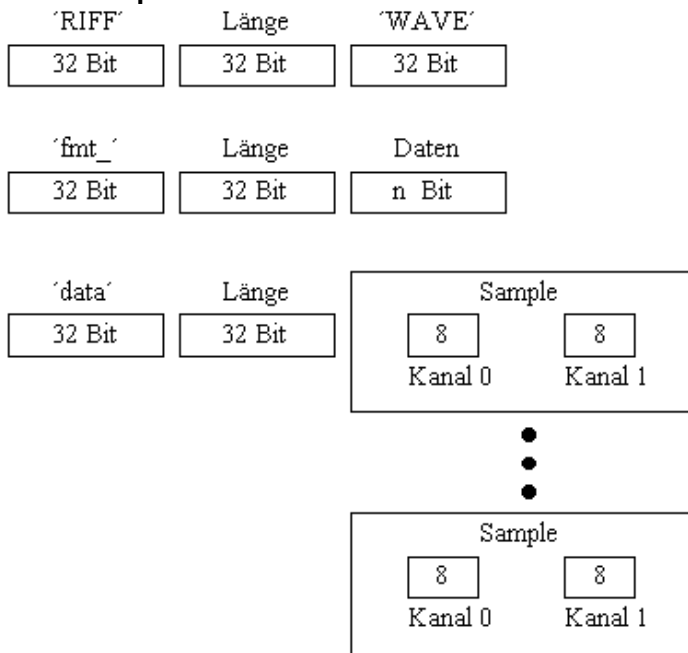


Abb.2

Die Abb.2 zeigt das Schema einer einfachen 8-Bit Stereo-Datei. In den ersten 32 Bit steht das Wort 'RIFF', danach kommt eine 32 Bit Längenangabe. Nach dieser folgt das Wort 'WAVE', usw...

Wie man sieht besteht ein gewaltiger Unterschied zwischen diesen Formaten. Als nächstes sollen die wesentlichsten Unterschiede als Vor- und Nachteile des WAVE-Formates aufgezählt werden.

3. Vorteile des Formates WAVE

Wave Dateien können auf allen Soundkarten ausgegeben werden, sofern diese von Windows unterstützt werden.

...

Viele Editor arbeiten mit dem VOC-Format, da damit bei der Aufnahme von externen Tonquellen (Line-In, Microphon) keinerlei Probleme zu erwarten sind und keine Grenzen innerhalb der Hardwaremöglichkeiten gesetzt sind. Will man nun eine VOC-Datei in eine Wave-Datei umwandeln so benützt man dafür ein Konvertierungsprogramm. Die Nachteile des Wave Formates spürt man am Stärksten bei der Umwandlung einer VOC-Datei in eine WAVE-Datei.

4. Nachteile des Formates WAVE

Das Wave-Format kennt keine Silence-Blöcke, somit kann in eine WAVE-Datei keine Stille eingefügt werden, wenn dies gewünscht sein sollte. Bei der Umwandlung von VOC in WAVE werden die Silence-Blöcke der VOC-Datei daher vernachlässigt.

Das Wave-Format kann Sampleraten nur mit 11.025, 22.05 oder 44.1 kHz ausgeben. Somit können VOC-Dateien die mit einer anderen Samplerate abgespeichert wurden nicht korrekt umgewandelt werden. Auch fällt damit ein toller Effekt in der Tonausgabe aus, der im VOC-Format voll ausgenützt werden kann.

Das Wave-Format kennt obendrein die Schleifenblöcke des VOC-Formates nicht. Diese Eigenschaft macht sich in der Speicherbelegung einer WAVE-Datei bemerkbar, wenn zum Beispiel eine Schleife mit 5 Wiederholungen aus einer VOC-Datei in eine WAVE-Datei umgewandelt wird, da das WAVE-Format diesen Ton-Daten-Block 5 mal in die WAVE-Datei einsetzt.

Auch kann eine WAVE-Datei nur eine Samplerate in einem Musikstück verarbeiten und nicht mehrere wie das bei einer VOC-Datei möglich ist.

Außerdem kann eine Wave Datei die Tondaten nicht packen, was nicht unbedingt zur Verkleinerung der WAVE-Datei beiträgt.

Es gibt sicherlich noch mehrere Vor- bzw. Nachteile des WAVE- oder des VOC-Formates die in diesem Anhang nicht angeführt sind, da sie entweder unerheblich oder mir nicht bekannt sind. □

Support, Santa Cruz Style

or

Where Do These People Come From?

by Jeff Liebermann (jeffl@comix.santa-cruz.ca.us) 07/09/94 (All these really happened to me since 1983.)

1. "My hard disk won't boot". I suggest they take the floppy out of drive A:.. Later when I arrive, they have successfully removed the floppy drive from the machine (with the floppy disk still inside).
2. "My dog goes nuts when I run Windows. No problem with any DOS programs". Her monitor had a cracked flyback transformer. When the multisync monitor switched scan rates upon entering Windows, the high frequency audio produced by the broken flyback was heard by the dog.
3. "Michaelangelo virus ate my hard disk, but I have a tape backup. Can you help me restore the system". No problem. When I arrive, I find the data on the tape was 18 months old and that she had never run a backup. "I thought you just shoved in the tape and it sucked up the data".
4. "How do I get on the national data information super highway?". I ask if he has accounts on any bbs's. He has Netcom, Compuserve, and others. I tell him he's already on the highway. "Is that all there is?" I hangup.
5. "What's the fastest way to move 500MBytes of data daily from Santa Cruz to Los Angeles?". Answer: FedEx.
6. „How many RJ45 connector does it take to build 8ea 10baseT cables?“ Answer: 45. I put the first 16 connectors on with one end backward. I then chopped off the good ends. Chopping off the other 8 connectors and effectively starting over consumed another 16 connectors. The 2nd try resulted in one end being mirror-imaged. Chopping of 8 more connectors I finally got them wired correctly. Then I tested them for continuity and found 5 bad crimps. Total=45.
7. "What kind of hard disk do you have?" Well... It's black with a little red light ... (groan).
8. Most common support call. "I lost my CMOS setup. How many heads, cylinders, and sectors does a _____ drive have?"
9. "I move the mouse in any direction and the cursor only moves an inch or so on the screen and stops". Take the foam shipping ring out from around the mouse ball.
10. "My systems on fire. What do I do?". Ummmmm. Turn it off? "(Click)"
11. Most hated support call: "I'm not sure if we need a computer system. Can you give me the relative advantages of Unix, DOS, Windows, Novell, MacIntosh, Sun, etc...?"
12. Favorite software support call: "I just installed Word 6.0 for Windows. It's really big and slow. How much will it cost to upgrade my machine?"
13. "My floppy drive won't read disks". I suggest they clean out the dust from the drive. "I can't". Huh? "The dust won't move". I find that they were using spray glue near the machine and that all the dust was glued in place.
14. How to impress a new customer: I walk into the computer room and knock the fire extinguisher off the wall which immediately sprays everything with dust.

15. *"My printer stopped working"*. Turn it upside down and shake out the staples and paper clips. Works every time.
16. *"Can you teach me how to use a computer?"*. I answer: No. I just fix the machines, I don't use them.
17. The company motto: *"If this stuff worked, you wouldn't need me"*.
18. From one of my smarter clients: *"Why is something broken every time you're here?"*
19. *"I'm trying to install a 2nd IDE drive. Support told me to take out ALL the jumpers"*. How many did you take out? *"12"*. (What they meant were the two easily accessible jumpers).
20. I call a manufacturer to order a manual on some junk I picked up surplus. The receptionist asks my name and company. She notes that I'm not in their database and could she have my address and phone numbers. No problem. I'm then transferred to the customer service department which notes that I'm not in the database and asks for the same information. The customer service person transfers me to the parts department which notes that I'm not in the database etc... Since the manual will take a few days to arrive, I ask for tech support who notes that I'm not in the database etc... The manual arrived promptly followed by 4 identical envelopes of promotional literature with exactly the same name and address.
21. Question LEAST likely to be answered correctly by support: *"What is the current version of your software/hardware/firmware?"*
22. Pacific Telephone Support Dept (Dial 611 for repair service), now asks you to punch in your phone number, and then warns you that you will be asked to verbally recite the same number when the service operator answers. I wonder what happens if they're different?
23. Email from a friend: *"CanYouFixTheSpaceBarOnMyKeyboard?"*
24. Fax back information service for additional information from one vendor requesting just one item returns a copy of their catalog page plus 10 pages of promotional garbage.
25. Email autoreply from support@_____.com Thank you for your support request. (drivel deleted) Please refer to support request number: Error: cannot create /u/something/filename (4 lines of garbage deleted) in future correspondence. Your request will be processed in the order received. (more garbage with Out of space on hd(1,41) mixed in.)
26. Conversation with support at a certain controller manufacturer. *"I can't answer that, please call your dealer"*. *"I am the dealer."* *"Then call your distributor"* *"He said for me to call you"* *"Then have the customer call us"* *"AAAAAGH!"* <click>
27. Modems and payphones don't mix. I hotwired my laptop into the mouthpiece of a payphone and proceeded to do system maintenance on a customer's machine. The sheriff arrived shortly and proceeded to interrogate me. Someone called complaining that I was using a computer to steal money from the payphone.
28. Having my system page me when it does an unscheduled reboot was a good idea. Having all my customers machines do the same was a mess after a power failure and 100+ pages.
29. *"My hard disk has a virus!"*. How can you tell, I ask? *"When I type DIR, it says VIRUS <DIR> and some date stuff"*. (Hint: Never name the directory for virus scanning software VIRUS).
30. Some monitor manufacturers suggest using alcohol to clean the screen. They forget to mention that the monitor should be off. (Boom).
31. I told a customer to take his machine to a gas station and have them blow the dust out. The gas station hands him a 150psi air nozzle that belches rusty water and oil. I got to clean up the mess for free. He also mangled the floppy heads with the high pressure.
32. Oxymoron candidate: Disk Protector. That's the cardboard disk they shove in the floppy drive for shipping. More drives have been mangled by shoving in the wrong shape, backwards, or bent than have ever been protected by them. Use a floppy disk instead.
33. What's the difference between a Van DeGraf static generator and a belt driven vacuum cleaner? Answer: Not much. Don't use a vacuum to clean your computer.
34. After the cleaning service crashed the computer for the 4th time by plugging the floor sweeper into the UPS, I decided to take action. I suggested they install "child proof" plastic plugs in any outlets deemed worthy of protection. The order went through the chain of confusion, and I was soon blessed with 1000 child proof plugs hot stamped with "Protected". I gave instructions to install about 10 of them on the protected outlets. However, the maintenance person assigned to the task knew nothing and proceeded to plaster every outlet in the building with the plugs. Mutiny was averted by spending all night removing the monsters. Three years later, they are still appearing.
35. Hint: Do not allow long hair black cats to sleep atop laser printers and tape drives. The black hair is almost invisible in black patterns, gears, and rollers.
36. Forensic filth analysis is a new part of computer repair. I now carry a microscope and some chemicals which are used to determine the exact nature of the filth I remove from keyboards, mice, computers, light pens. While nobody pays me to do this, it definitely adds to the entertainment value.
37. Why do customers think that I maintain a document and device driver library for every conceivable board ever made?
38. From a hard disk drive manufacturer: *"The drive stopped working. I popped the little plug and noticed it was awful dry inside. I added some oil but it didn't help"*.
39. Which arrow key? There are 17 arrows on the keyboard.
40. Favorite error message: *"Out of paper on drive D:"* This was produced by a timeout error on a slow WORM drive and a defective AT/IO card.
41. At one time, I was into antique furniture. When I purchased my first computer (IBM 4.77 PC), I decided that it deserved a suitable antique table. I ask the local antique dealer: *"Do you have an antique computer desk?"*. He looks at me with a strange look and says: *"They didn't have computers when this stuff was made"*.
42. When 3.5" floppies first appeared, some users were confused with the operation of the write protect window. One user wanted to be doubly sure that the disk would be safe from his mistakes. He correctly opened the window and just to be sure, covered it with one of the magic write protect tabs from a 5.25" floppy.
43. Favorite Windoze game: *"Guess what this icon does?"*
44. A video store installed the computer on top of the cash drawer. Every time the cash drawer would open, the hard disk would get a good bouncing. I decided that this was technically disgusting, and moved the machine. The next morning, the drive wouldn't spin up (stiction). Solution: Put it back on top of the cash drawer and let it bounce.
45. The curse of the mad labeler. Some of the clone cards I see have stick on METALIZED labels that are quite good at shorting traces. I've fixed a few by just removing the stick-on short. A variation on this effect is the tendency for some distributors to put stick-on labels on TOP of their 486 chips. Then they smear on some silicon grease and bury the mess under a heat sink and fan. The air gap produced between the chip and heat sink severely degrades its cooling value. □