

# Dem Klang auf der Spur Teil 3

Der SID (Sound Interface Device) ist funktionell an das klassische Konzept von Moog angelehnt. Dadurch ist seine Arbeitsweise leicht verständlich. Bild 1 zeigt das Blockschema des SID. Dieses Schema dient dazu, den Fluß der Audio-Signale zu veranschaulichen. Die Funktionsblöcke waren in ähnlicher Form bereits Bestandteile des Synthesizer-Schemas aus Teil 2. Den linken Teil des Schemas bilden drei identische, voneinander unabhängige, Funktionsgruppen.

Eine solche Gruppe setzt sich aus einem DCO (Digital Controlled Oscillator), einem Amplitudenmodulator (oder auch DCA = Digital Controlled Amplifier) und einem Hüllkurvengenerator (EG = Envelope Generator) zusammen. Der DCO kann wahlweise eine von vier Kurvenformen erzeugen: Dreieck, Sägezahn, Rechteck und Rauschen. Dabei ist das Tastverhältnis der Rechteckkurve steuerbar. Unter dem Tastverhältnis versteht man das Verhältnis zwischen der Länge des Kurvenabschnitts mit hoher Spannung zur gesamten Periodenlänge. Eine symmetrische Rechteckkurve hat demnach ein Tastverhältnis von 50 Prozent. Eine Veränderung im Tastverhältnis T bewirkt eine Klangfarbenänderung. So klingt eine Rechteckkurve mit T = 50 Prozent voluminös. Bewegt man sich mit T in Richtung 0 Prozent oder 100 Prozent, so wird der Klang zunehmend obertonreicher aber dünner, da der Anteil des Grundtons abnimmt. Einen besonders lebendigen Klang erhält man durch dynamische Veränderung des Tastverhältnisses, was zwar in der SID-Hardware nicht realisiert, aber softwaremäßig möglich ist.

## Funktioneller Aufbau des SID

Der Hüllkurvengenerator beeinflusst über den Amplitudenmodulator den zeitlichen Lautstärkeverlauf der vom DCO kommenden Kurve. Die Hüllkurve wird nach dem bekannten ADSR-Schema parametrisiert.

### Synchronisation und Ringmodulation

Die senkrechten Verbindungen von DCO1 zu DCO2, von DCO2 zu DCO3 und von DCO3 zurück zu DCO1 können einzeln zu- oder abgeschaltet werden. Sie dienen Spezialeffekten, die

das Spektrum des SID beträchtlich erweitern. Im Normalfall, wenn diese Steuerpfade unwirksam geschaltet sind, schwingen die drei DCOs unabhängig voneinander, jeder in seiner vorprogrammierten Frequenz und Kurvenform. Im Falle der **Synchronisation** zwingt der synchronisierende DCO einen weiteren DCO dazu, gleichphasig zu schwingen. In Bild 2 erzeugt DCO1 eine Sägezahnsschwingung von 100 Hz und synchronisiert DCO2, der ebenfalls auf Sägezahn, aber 350 Hz eingestellt ist. Nach jeweils dreieinhalb Perioden wird DCO 2 gezwungen, eine neue Periode anzufangen. DCO2 erzeugt so eine viel komplexere Kurvenform, als er es ohne Synchronisation tun würde. Variiert man nun noch die Frequenz eines der beiden DCOs, während man die des anderen konstant hält, so erzeugt DCO2 ständig andere Kurvenformen. Man erhält damit schwer zu be-

schreibende, aber meistens »elektronisch« (im Sinne von »unnatürlich«) klingende Muster.

Wenn man zu jedem Zeitpunkt die Werte zweier Kurvenzüge miteinander multipliziert, so spricht man von **Ringmodulation**. Dieser Vorgang hat eine Ähnlichkeit mit der Modulation einer Schwingung mit einer Hüllkurve, wie sie im SID auch vorkommt. Die Hüllkurve ist aber eine Funktion, die sich, verglichen mit dem Signal das sie moduliert, nur langsam verändert. Dadurch bleibt bei dieser Modulation der Klangcharakter des modulierten Signals erhalten, es ändert sich nur seine Lautstärke.

Bei der Ringmodulation dagegen ist das modulierende Signal von ähnlicher Beschaffenheit wie das modulierte Signal. Beide Signale dürfen ähnliche Frequenzen haben und als Kurvenzüge auch positive und negative Werte annehmen, wogegen eine Hüllkurve immer nur nicht-

negative Werte hat. Bei der Ringmodulation geht im Allgemeinen der Klangcharakter beider beteiligten Kurven verloren; es entsteht ein ganz neuer Klang. Man macht sich das am besten anhand zweier Sinusschwingungen klar:

$$\sin(\omega_1 t) \sin(\omega_2 t) = \frac{1}{2} (\cos(\omega_1 - \omega_2)t - \cos(\omega_1 + \omega_2)t)$$

Das bedeutet, daß man durch Multiplikation zweier Sinusschwingungen ein Signalgemisch erhält, das aus Schwingungen mit der Summe und der Differenz der ursprünglichen Frequenzen besteht. Die ursprünglichen Frequenzen verschwinden dabei vollkommen. Die Formel liefert auf der rechten Seite zwar Cosinus-Terme, für den Klang ist das allerdings unerheblich, da sich Sinus und Cosinus nur durch eine Phasenverschiebung unterscheiden. Unterzieht man nun andere Kurvenformen einer Ringmodulation, so muß man alle Obertöne der einen Kurve mit allen Obertönen der anderen Kurve gemäß der obigen Formel verrechnen. Dadurch entsteht ein sehr reichhaltiges neues Obertonspektrum. Die neuen Obertöne stehen dabei nicht mehr in harmonischen, das heißt ganzzahligen Verhältnissen zueinander. Aus diesem Grund eignen sich Ringmodulatorklänge auch kaum zur Wiedergabe von Melodien. Im natürlichen Umfeld findet man unharmonische Obertonverhältnisse zum Beispiel bei Glocken, Gongs und schwingenden Metallplatten. Zu deren Nachahmung eignen sich die Ringmodulatorklänge.

Beim SID steht zur Ringmodulation nur die Dreiecksschwingung zur Verfügung. Wenn DCO1 den DCO2 moduliert, ist das Ringmodulatorprodukt nur dann hörbar, wenn DCO2 auf Dreieckskurve eingestellt ist. Noch komplexere Klänge erhält man beim SID durch Einbeziehen aller drei DCOs in die Modulationskette, wobei man durch den Signalpfad DCO3 nach DCO1 den Kreis auch noch schließen kann. Die Ergebnisse werden dann allerdings schwer vorhersagbar, lassen also noch genug Raum für Experimente und Überraschungen.

### Filterung

Leider hat der SID nur ein gemeinsames Filter für die drei DCO-EG-AM-Gruppen. Möchte man unterschiedliche Klangbilder zur gleichen Zeit erzeugen, so kann man nur durch die Wahl von Kurvenform und Hüllkurve

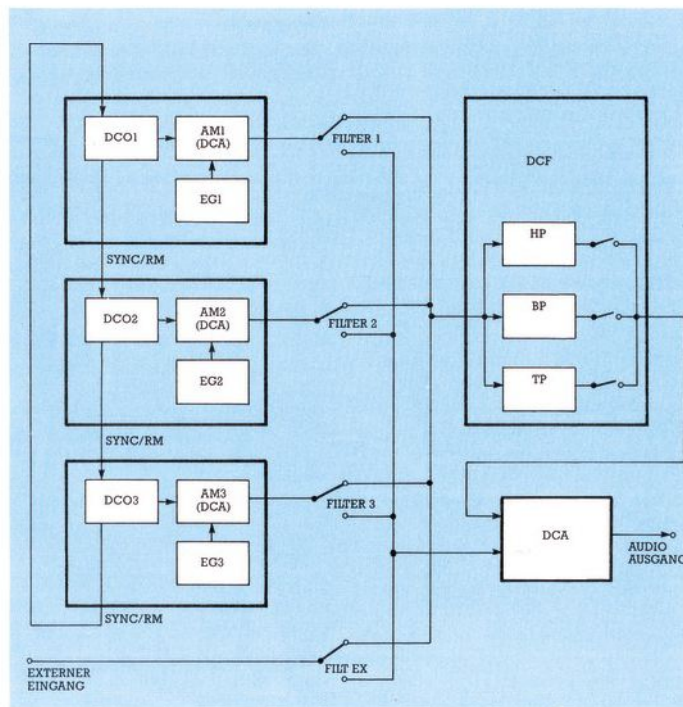


Bild 1. Blockschema des SID

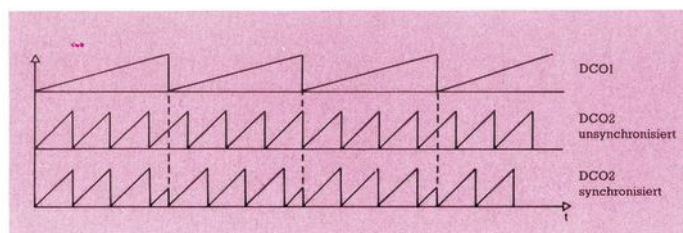


Bild 2. Die Synchronisation

# Nachdem im zweiten Teil dieser Reihe einige allgemeine Grundlagen aus dem Themenkomplex »Musik und Computer« dargestellt worden sind, wird in dieser Folge ausführlich der Synthesizer Chip SID (Sound Interface Device) 6581 vorgestellt.

differenzieren. Man hat aber immerhin die Wahl, ob man die DCO-EG-AM-Produkte überhaupt durch das Filter schickt oder am Filter vorbei direkt zum Ausgang. Diese Funktion erfüllen im Blockschema die Schalter »FILTER 1«, »FILTER 2« und »FILTER 3«. Das Filter kann als Hochpaß, als Bandpaß oder als Tiefpaß wirken, wobei die Funktionen auch parallel schaltbar sind. So erhält man zum Beispiel aus der Kombination Hochpaß und Tiefpaß eine sogenannte Bandsperre (oder »Notch-Filter«), die ein schmales Band aus dem Gesamtspektrum unterdrückt. Es sind die Parameter **Eck- beziehungsweise Mittenfrequenz** (für HP, BP und TP gemeinsam) sowie die **Resonanz** programmierbar. Ein weiteres Qualitätskriterium eines Filters ist seine Steilheit. Diese Größe beschreibt, wie schnell ein Filter in der Umgebung der Eck- beziehungsweise Mittenfrequenz vom durchlassenden in den sperrenden Zustand übergeht.

Als Faustregel gilt, daß ein Filter um so »besser« klingt, je steiler es ist. Beim Hoch- und Tiefpaß des SID beträgt die Steilheit 12 dB/Oktave, beim Bandpaß 6 dB/Oktave. 12 dB/Oktave bedeuten, daß zum Beispiel im Übergangsbereich des Tiefpasses ein Signal um das Vierfache abgeschwächt wird, wenn seine Frequenz verdoppelt wird. 6 dB/Oktave bedeuten eine Abschwächung (oder Anhebung) um das Zweifache bei Frequenzverdopplung. Die Filtersteilheit ist im Allgemeinen und auch beim SID fest vorgegeben. Zur Orientierung sei noch erwähnt, daß die Filter im professionellen Synthesizer meistens eine Steilheit von 24 dB/Oktave haben.

Die gefilterten oder ungefilterten Signale im SID werden auf einen DCA geführt, wo man noch die Gesamtamplitude des Ausgangssignals programmieren kann. Über einen Analogeingang kann man auch noch ein externes Signal gefiltert oder ungefiltert zumischen. Dieses Signal könnte zum Beispiel von einem zweiten SID stammen.

Gegenüber dem im ersten Teil vorgestellten klassischen Synthesizerkonzept findet man im SID keinen LFO, mit dem man Frequenz, Amplitude oder einen Filterparameter modulieren könnte. Diese Funktion kann man aber rein softwaremäßig realisieren. Eine Hilfe dazu können DCO3 und EG3 sein. Man kann zu jedem Zeitpunkt den

Amplitudenwert von DCO3 und von EG3 abfragen. Programmiert man DCO3 als LFO, das heißt auf eine sehr niedrige Frequenz, so kann man die Amplitudenwerte von DCO3 (mit geeigneter Skalierung) zur Frequenz von DCO1 oder DCO2 addieren. Tut man das in regelmäßigen Zeitabständen und mehrfach innerhalb einer Periodendauer von DCO3, so kann man damit ein Vibrato realisieren.

Auf gleiche Weise kann man auch die von EG3 gelieferte Hüllkurve zur Modulation beispielsweise des Filters heranziehen. Beide Möglichkeiten erfordern allerdings zusätzliche schnelle Programme, die sich nur in Maschinensprache realisieren lassen.

Wenn man DCO3 und/oder EG3 zu Modulationszwecken benutzt, möchte man das von AM3 produzierte Signal unter Umständen nicht hören. Dazu kann man es mit dem zusätzlichen Schalter »AUS« unterdrücken.

Im letzten Teil wurde schon angesprochen, daß ein digitaler Synthesizer wie der SID nicht durch Potentiometer und durch Spannungen beeinflusst wird, sondern durch digitalisierte Parameter. Dazu besitzt der SID 29

Register mit einer Länge von 8 Bit. Davon können 25 nur beschrieben werden (ihre Inhalte steuern das Verhalten des SID) und 4 nur gelesen werden. Die 29 Register werden durch die Adreß-Decodierungs-Hardware auf dem CPU-Speicherbereich \$D400 bis \$D41C abgebildet (dezimal: 54272 bis 54300).

## Die Steuerung des SID

Die SID-Register lassen sich so mit allen Hauptspeicherbezogenen Maschinenbefehlen oder mit PEEK und POKE ansprechen. Da in einzelnen Registern oft mehrere Bits mit unterschiedlicher Bedeutung zusammengefaßt sind, erfordert ihre Programmierung ein hohes Maß an maschinennahem Denken, egal ob in Basic oder in Maschinensprache programmiert wird.

Die Tabelle 1 zeigt die Bedeutung der einzelnen Register im Detail. Der Registersatz gliedert sich in drei mal sieben Register zur Steuerung der drei DCO-EG-AM-Gruppen für die drei Stimmen, in vier Register zur Fil-

tersteuerung und in vier Lese-register. Die sieben Register zur Steuerung einer DCO-EG-AM-Gruppe haben für alle drei Stimmen den gleichen Aufbau und die gleiche Bedeutung.

### Frequenz low/high (Register 0 und 1)

Die beiden Register 0 und 1 steuern die Frequenz von DCO1 mit einer Genauigkeit von 16 Bit. Register 0 enthält das niederwertige, Register 1 das höherwertige Byte einer 16-Bit-Größe F. Zwischen der Ausgangsfrequenz f und der Zahl F besteht der Zusammenhang:

$$(1) \quad f = F \times T / 2124 \text{ (Hz)}$$

Dabei ist T die Taktfrequenz der CPU, die auch am SID anliegt. Sie beträgt bei der deutschen Version des C 64 0,985248 MHz

Damit gilt:

$$(2) \quad f = F \times 0,0587254 \text{ (Hz)}$$

oder

$$(3) \quad F = f \times 17,0284$$

Die Gleichungen (2) und (3) zeigen, daß sich die Frequenzen der DCOs sehr fein, in Schritten zu zirka 1/17 Hz, programmieren lassen. Möchte man eine vorgegebene Frequenz f (im Beispiel FAUS) erzeugen, so errechnet man F nach (3),  
10 FAUS = 440

Adresse/Reg.	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	Registername	
54272 0	F7	F6	F5	F4	F3	F2	F1	F0	Frequenz low	Stimme 1
54273 1	F15	F14	F13	F12	F11	F10	F9	F8	Frequenz high	
54274 2	P7	P6	P5	P4	P3	P2	P1	P0	Pulsweite low	Stimme 2
54275 3	unbenutzt	unbenutzt	unbenutzt	unbenutzt	P11	P10	P9	P8	Pulsweite high	
54276 4	Rauschen	Rechteck	Sägezahn	Dreieck	Test	Ringmodulat.	Synchronisat.	GATE	Kontrollregister	Stimme 3
54277 5	Attack 3	Attack 2	Attack 1	Attack 0	Decay 3	Decay 2	Decay 1	Decay 0	Attack/Decay	
54278 6	Sustain 3	Sustain 2	Sustain 1	Sustain 0	Release 3	Release 2	Release 1	Release 0	Sustain/Release	Filter
54279 7	F7	F6	F5	F4	F3	F2	F1	F0	Frequenz low	
54280 8	F15	F14	F13	F12	F11	F10	F9	F8	Frequenz high	Leseregister
54281 9	P7	P6	P5	P4	P3	P2	P1	P0	Pulsweite low	
54282 10	unbenutzt	unbenutzt	unbenutzt	unbenutzt	P11	P10	P9	P8	Pulsweite high	
54283 11	Rauschen	Rechteck	Sägezahn	Dreieck	Test	Ringmodulat.	Synchronisat.	GATE	Kontrollregister	
54284 12	Attack 3	Attack 2	Attack 1	Attack 0	Decay 3	Decay 2	Decay 1	Decay 0	Attack/Decay	
54285 13	Sustain 3	Sustain 2	Sustain 1	Sustain 0	Release 3	Release 2	Release 1	Release 0	Sustain/Release	
54286 14	F7	F6	F5	F4	F3	F2	F1	F0	Frequenz low	
54287 15	F15	F14	F13	F12	F11	F10	F9	F8	Frequenz high	
54288 16	P7	P6	P5	P4	P3	P2	P1	P0	Pulsweite low	
54289 17	unbenutzt	unbenutzt	unbenutzt	unbenutzt	P11	P10	P9	P8	Pulsweite high	
54290 18	Rauschen	Rechteck	Sägezahn	Dreieck	Test	Ringmodulat.	Synchronisat.	GATE	Kontrollregister	
54291 19	Attack 3	Attack 2	Attack 1	Attack 0	Decay 3	Decay 2	Decay 1	Decay 0	Attack/Decay	
54292 20	Sustain 3	Sustain 2	Sustain 1	Sustain 0	Release 3	Release 2	Release 1	Release 0	Sustain/Release	
54293 21	unbenutzt	unbenutzt	unbenutzt	unbenutzt	unbenutzt	GF 2	GF 1	GF 0	Grenzfrequenz low	
54294 22	GF 10	GF 9	GF 8	GF 7	GF 6	GF 5	GF 4	GF 3	Grenzfrequenz high	
54295 23	Resonanz 3	Resonanz 2	Resonanz 1	Resonanz 0	Filter 3	Filter 2	Filter 1	Filter 0	Resonanz/Filter	
54296 24	Aus	Hochpass	Bandpass	Tiefpass	L 3	L 2	L 1	L 0	Mode/Laufstärke	
54297 25	Pot X 7	Pot X 6	Pot X 5	Pot X 4	Pot X 3	Pot X 2	Pot X 1	Pot X 0	Potentiometer X	
54298 26	Pot Y 7	Pot Y 6	Pot Y 5	Pot Y 4	Pot Y 3	Pot Y 2	Pot Y 1	Pot Y 0	Potentiometer Y	
54299 27	07	06	05	04	03	02	01	00	Oszillator 3	
54300 28	H7	H6	H5	H4	H3	H2	H1	H0	Hüllkurve Osz 3	



Tabelle 1. Die Register des SID

Wert Dez.	Hex	Attack	Decay/Release
0	0	2 ms	6 ms
1	1	8 ms	24 ms
2	2	16 ms	48 ms
3	3	24 ms	72 ms
4	4	38 ms	114 ms
5	5	56 ms	168 ms
6	6	68 ms	204 ms
7	7	80 ms	240 ms
8	8	100 ms	300 ms
9	9	250 ms	750 ms
10	A	500 ms	1,5 s
11	B	800 ms	2,4 s
12	C	1 s	3 s
13	D	3 s	9 s
14	E	5 s	15 s
15	F	8 s	24 s

Die Zeiten gelten bei einer System-Taktfrequenz von 1 MHz. Da diese Frequenz beim deutschen C 64 mit 0,985248 MHz nur gering davon abweicht, ist die Tabelle auch für diese Frequenz brauchbar.

Quelle: Commodore 64 Programmers Reference Guide

Tabelle 2. Hüllkurvengeschwindigkeiten

20 F = FAUS \* 17.0284

man zerlegt F in das Low- und das High-Byte,

30 F = INT(F+0.5) :REM RUNDUNG

40 HI = INT(F/256)

50 LO = F - 256\*HI

und besetzt Register 0 und 1 damit

60 SID = 54272 :REM BASIS-ADRESSE

70 POKE SID,LO

80 POKE SID+1,HI

**Pulsweite low/high (Register 2 und 3)**

Der Inhalt dieser Register steuert das Tastverhältnis der Rechteckkurve. Er ist nur dann bedeutend, wenn als Kurvenform das Rechteck gewählt wird. Die Pulsweite kann auf 12 Bit genau festgelegt werden. Register 2 enthält das Low-Byte, Register 3 das High-Byte, von dem nur die unteren 4 Bits (P8 bis P11) berücksichtigt werden. Zwischen der 12-Bit-Größe P und dem Tastverhältnis PAUS besteht der Zusammenhang:

(4) PAUS = P / 40.95 (%)

oder

(5) P = PAUS \* 40.95

Die Programmierung gestaltet sich dann in der Praxis analog zu der der Frequenz:

90 PAUS = 50

100 P = PAUS \* 40.95

110 P = INT(P+0.5) :REM RUNDUNG

120 HI = INT (P/256)

130 LO = P - 256\*HI

140 POKE SID+2,LO

150 POKE SID+3,HI

**Kontrollregister (Register 4)**

Jedes Bit dieses Registers hat eine eigene Bedeutung. GATE (Bit 0)

Dieses Bit steuert den Hüllkurvengenerator EG1. Wird es gesetzt, startet EG1 eine Attack-Decay-Sequenz. Die Hüllkurve bleibt anschließend auf dem programmierten Sustain-Pegel, bis das GATE-Bit zurückgesetzt wird. Durch das Zurücksetzen geht die Hüllkurve in die Release-(Ausklang)-Phase.

**Synchronisation (Bit 1)**

Wird es gesetzt, so wird DCO1 von DCO3, wie schon beschrieben, synchronisiert.

**Ringmodulation (Bit 2)**

Wird es gesetzt, so erzeugt DCO1 das Ringmodulatorprodukt der Dreieckskurven von DCO3 und DCO1. Diese wird allerdings nur dann hörbar, wenn als Kurvenform von DCO1 das Dreieck gewählt wird.

**Test (Bit 3)**

Bei gesetztem Test-Bit wird DCO1 auf Nullpegel gezwungen. Er erzeugt in diesem Zustand keine Schwingung. Man kann DCO1 damit softwaregesteuert synchronisieren, um, ähnlich wie durch die Synchronisation durch DCO3, komplexe Kurvenformen zu erhalten.

**Kurvenform (Bit 4 bis 7)**

Durch Setzen eines dieser 4 Bits wählt man eine der Kurvenformen Dreieck, Sägezahn, Rechteck oder Rauschen. Es muß mindestens eines dieser Bits gesetzt sein, damit überhaupt etwas hörbar wird. Eine Eigenart des SID ist es, daß sich die Kurvenformen **nicht** additiv verhalten. Werden mehrere der Bits 4 bis 7 zugleich gesetzt, so erzeugt der SID eine Kurvenform, deren Amplitudenwerte man durch logische AND-Verknüpfung der Amplitudenwerte der einzelnen Kurven erhält. Diese AND-Verknüpfung muß man sich bitweise auf die Amplitudenwerte darstellenden Bytes angewandt vorstellen.

Das **Rauschen** verdient noch eine besondere Betrachtung. Bei Rauschen kann man nicht von einer Frequenz im üblichen Sinne reden. Dennoch ist der Charakter des SID-Rauschens über die Größe F in Register 0 und 1 beeinflussbar. Rauschen wird im SID durch eine Quasi-Zufallsfolge von numerischen Werten realisiert. Die Rate, mit der der DCO diese Zufallszahlen erzeugt, ist genau die durch F programmierte Frequenz. Ein Rau-

```

100 SI=54272 :REM BASISADRESSE <248>
110 : <168>
120 REM FREQUENZ <119>
130 REM FAUS :F=INT (FAUS*17.0284+0.5) <236>
140 HI=INT (F/256) :LO=F-256*HI <183>
150 POKE SI,LO :POKE SI+1,HI <149>
160 : <218>
170 REM PULSWEITE (TASTVERHAELTNIS) <205>
180 READ PAUS :P=INT (PAUS*40.95+0.5) <216>
190 HI=INT (P/256) :LO=P-256*HI <253>
200 POKE SI+2,LO :POKE SI+3,HI <165>
210 : <012>
220 REM WELLENFORM <102>
230 READ WF:WF=WF AND 254 :REM GATE AUS <077>
240 : <042>
250 REM A D S R <179>
260 READ A:READ D:POKE SI+5,16*A+D <140>
270 READ S:READ R:POKE SI+6,16*S+R <215>
280 : <083>
290 REM FILTER AUS, LAUTSTAERKE MAX <184>
300 POKE SI+23,0:POKE SI+24,15 <218>
310 : <113>
320 REM AUF TASTENDRUCK WARTEN <197>
330 REM UND ADSR SEQUENZ AUSLÖSEN <197>
340 READ T <048>
350 GET A$:IF A$="" THEN 350 <196>
360 POKE SI+4,WF OR 1 <036>
370 FOR I=1 TO T:NEXT <212>
380 POKE SI+4,WF:GOTO 350 <178>
500 REM===== <036>
510 REM KLANGPARAMETER <156>
520 REM===== <057>
530 DATA 440 :REM FREQUENZ (HZ) <091>
540 DATA 50 :REM TASTVERHAELTNIS (%) <198>
550 DATA 64 :REM KURVENFORM <237>
560 DATA 2 :REM ATTACK <104>
570 DATA 8 :REM DECAY <038>
580 DATA 0 :REM SUSTAIN <233>
590 DATA 8 :REM RELEASE <213>
600 DATA 150 :REM ZEIT ZWISCHEN <227>
610 REM ATTACK UND RELEASE <147>

```

Listing 1. Mit Programm 1 kann man sich damit vertraut machen, wie einzelne Töne mit verschiedenen Kurvenformen und Hüllkurven klingen. Die Parameter lassen sich in den DATA-Zeilen am Ende des Programms leicht modifizieren.

```

10 REM===== <118>
20 REM PROGRAMM 2 <058>
30 REM <173>
40 REM EINZEILER FUER SIGNALTON <063>
50 REM===== <158>
60 S=54272:FOR I=0 TO 6:READ X:POKE S+I,X:NEXT
:POKE S+24,15:POKE S+4,17:POKE S+4,16
:DATA 0,99,0,8,0,11,11 <041>

```

Listing 2. Dieser Einzeiler ist gegenüber dem übersichtlichen, aber etwas aufgeblähten Programm 1 so ziemlich die kompakteste Lösung, um dem SID einen Ton zu entlocken. Zur Eingabe muß man die Basic-Schlüsselwörter abkürzen.

```

100 SI=54272 :REM BASISADRESSE <248>
110 : <168>
120 REM TASTVERHAELTNIS = 50% <079>
130 POKE SI+2,0 :POKE SI+3,8 <155>
140 : <198>
150 REM ADSR <079>
160 A=0 :D=0 :POKE SI+5,16*A+D <221>
170 S=15 :R=0 :POKE SI+6,16*S+R <094>
180 : <238>
190 REM FILTER AUS UND LAUTSTAERKE MAX. <060>
200 POKE SI+23,0:POKE SI+24,15 <117>
210 : <012>
220 REM FLO=0, KURVENFORM <244>
230 POKE SI,0 <117>
240 POKE SI+4,65 <152>
250 : <052>
260 REM FHI AUF- UND ABWAERTS STEuern <218>
270 F0=50 :F1=100 <144>
280 FOR I=0.4 TO 50 STEP .2 : <057>
290 : FOR F=F0 TO F1 STEP 1 <089>
300 : POKE SI+1,F <231>
310 : NEXT F <057>
320 : FOR F=F1 TO F0 STEP -1 <034>
330 : POKE SI+1,F <005>
340 : NEXT F <087>
350 NEXT I <042>
360 POKE SI+4,64 <016>

```

Listing 3. Durch Verändern des höherwertigen Frequenz-Bytes in kleinen Schritten entsteht der Eindruck, daß der Ton kontinuierlich in der Höhe schwankt

```

134 REM DAS MASCHINENPROGRAMM BELEGT <155>
136 REM DEN BEREICH ($9000-$90AC) <092>
140 REM <027>
150 REM AUFRUFBEDINGUNGEN <020>
160 REM <047>
170 REM SYS PAR,S,PW,K,A,D,S,R <190>
180 REM <067>
190 REM BELEGT STIMME S (1,2,3) MIT <075>
200 REM <087>
210 REM PW (PULSWEITE 0...4095) <167>
220 REM K (KURVENFORM,SYNCHRONISA- <151>
230 REM TION,RINGMODULATION) <048>
240 REM A,D,S,R (HUELLKURVE 0...15) <119>
250 REM <137>
260 REM SYS EIN,S,F <096>
270 REM <158>
280 REM SCHALTET STIMME S MIT <012>
290 REM FREQUENZ F (0...65535) EIN <041>
300 REM <188>
310 REM SYS AUS,S <045>
320 REM <208>
330 REM SCHALTET STIMME S AUS <061>
340 REM===== <132>
360 : <163>
370 A=36864:B=37036 <162>
380 FOR I=A TO B <032>
390 READ X:POKE I,X:S=S+X:NEXT I <157>
400 IF S<>17579 THEN PRINT"TIFFFEHLER" <087>
410 : <213>
420 PAR=A+66 :EIN=A+122 :AUS=A+161 <049>
430 : <233>
440 REM DATAS FUER MASCHINENPROGRAMM <226>
450 : <253>
460 DATA 000,000,000,000,032,253,174 <031>
470 DATA 032,158,183,224,004,048,003,076 <085>
480 DATA 072,178,224,000,240,249,202,188 <099>
490 DATA 027,144,096,000,007,014,152,072 <096>
500 DATA 032,253,174,032,158,183,224,016 <117>
510 DATA 048,003,076,072,178,104,168,138 <139>
520 DATA 096,032,030,144,010,010,010,010 <101>
530 DATA 141,000,144,032,030,144,013,000 <111>
540 DATA 144,096,032,005,144,138,072,152 <157>
550 DATA 072,032,253,174,032,235,183,142 <165>
560 DATA 000,144,104,168,104,170,173,000 <158>
570 DATA 144,157,002,144,165,021,201,022 <170>
580 DATA 048,003,076,072,178,153,003,212 <195>
590 DATA 165,020,153,002,212,032,049,144 <192>
600 DATA 153,005,212,032,049,144,153,006 <206>
610 DATA 212,096,032,005,144,138,072,152 <223>
620 DATA 072,032,253,174,032,138,173,032 <234>
630 DATA 247,183,104,168,165,020,153,000 <243>
640 DATA 212,165,021,153,001,212,104,170 <233>
650 DATA 189,002,144,009,001,153,004,212 <252>
660 DATA 096,032,005,144,189,002,144,041 <017>
670 DATA 254,153,004,212,096 <207>

```

Listing 4. Die Programmierung des SID über POKE-Befehle ist mühsam. Drei kleine Maschinenprogramme, die man auch als Erweiterung des BASIC-Interpreters auffassen kann, werden von Programm 5 und 6 aufgerufen.

```

100 SI=54272 :REM BASISADRESSE <248>
110 PAR=36930:EIN=36986:AUS=37025 <182>
120 : <178>
130 REM FILTER AUS UND LAUTSTAERKE MAX. <000>
140 POKE SI+23,0:POKE SI+24,15 <057>
150 : <208>
160 REM PARAMETERSAETZE ABARBEITEN <105>
170 READ N <127>
180 FOR I=1 TO N <083>
190 : READ F,A,D,S,R,T1,T2 <002>
200 : SYS PAR,1,0,128,A,D,S,R <221>
210 : SYS EIN,1,F <085>
220 : PRINT"BEISPIEL ";I;" GATE AN"; <046>
230 : FOR J=1 TO T1:NEXT J <253>
240 : SYS AUS,1 <014>
250 : PRINT"(2SPACE)GATE AUS" <027>
260 : FOR J=1 TO T2:NEXT J <029>
270 NEXT I <218>
500 DATA 6 :REM ANZAHL DER P.SAETZE <090>
510 REM===== <046>
520 REM PARAMETERSAETZE <006>
530 REM <163>
540 REM F A D S R T1 T2 <040>
550 REM===== <087>
560 DATA 50000, 0, 0,15, 0, 1500, 200 <000>
570 DATA 25000, 0,12, 0,12, 500, 500 <014>
580 DATA 5000, 8, 5, 2, 5, 1500, 500 <197>
590 DATA 1000, 0,13, 0,12, 500,1500 <030>
600 DATA 750, 14, 0,15,13, 2000,2500 <105>
610 DATA 80, 8, 8, 8,10, 2500, 500 <188>

```

Listing 5. Programm 5 zeigt, wie vielfältig Rauschen klingen kann. Die Parametersätze in den DATA-Zeilen kann man beliebig verändern oder erweitern.

```

100 DIM F(24) :REM ARRAY F. FREQUENZEN <166>
110 SID=54272 :REM BASISADRESSE <070>
120 PAR=36930:EIN=36986:AUS=37025 <192>
130 : <188>
140 REM TONLEITER-FREQUENZEN BERECHNEN <139>
150 FAUS=110:H=2*(1/12) <175>
160 FOR I=0 TO 24 <086>
170 : F(I)=INT(FAUS*17.0284+0.5) <248>
180 : FAUS=FAUS*H <242>
190 NEXT I <137>
200 : <002>
210 REM PARAMETER FESTLEGEN <159>
220 PW=2048 :REM PULSWEITE <142>
230 K=32 :REM KURVENFORM <032>
240 A=0:D=9:S=0:R=9 <098>
250 SYS PAR,1,PW,K,A,D,S,R <252>
260 SYS PAR,2,PW,K,A,D,S,R <008>
270 SYS PAR,3,PW,K,A,D,S,R <019>
280 : <083>
290 REM FILTER AUS UND LAUTSTAERKE MAX. <161>
300 POKE SI+23,0:POKE SI+24,15 <218>
310 : <113>
320 REM ZUFALLSMELODIE <240>
330 I=1 <119>
340 N=INT(RND(1)*25) <171>
350 SYS EIN,I,F(N) <095>
360 FOR J=0 TO 20:NEXT J <034>
370 SYS AUS,I <111>
380 FOR J=0 TO 20:NEXT J <054>
390 I=I+1:IF I=4 THEN I=1 <109>
400 GOTO 340 <177>

```

Listing 6. Die Frequenzen der temperierten Stimmung werden über 2 Oktaven errechnet. Anschließend wird daraus eine Zufallsfolge erzeugt. Durch zyklisches Ansteuern aller drei Stimmen haben die Töne Zeit zum Ausklingen.

schen mit »hoher Frequenz« klingt heller oder »weißer« als Rauschen mit niedriger Frequenz.

Bei der Programmierung des Kontrollregisters muß man sich vorher den Wert jedes einzelnen Bits zurechtlegen und in das Byte packen. Das Anstoßen eines Rechteckklanges beispielsweise wird durch Setzen von Bit 0 und Bit 6 erreicht. Der numerische Wert des Kontrollbytes ist dann:  $2^{10} + 2^{16} = 65$

also:  $240 \text{ POKE SID} + 4,65$

Ein hörbares Resultat wird aber auch erst dann erzielt, wenn vorher die Hüllkurvenparameter vernünftig gesetzt sind.

#### ADSR (Register 5 und 6)

Die Parameter A, D, S und R können in 16 Abstufungen entsprechend 4 Bit Auflösung programmiert werden. Die Stufung für den Sustain-Pegel ist linear. S=0 entspricht dem Ruhepegel, S=15 entspricht dem Maximalpegel nach Attack. Bei S=15 besitzt die Hüllkurve keine Decay-Phase. Die Abstufungen für die Attack-, Decay- und Release-Zeiten sind vernünftigerweise nicht linear, um sehr kurze und sehr lange Zeiten gleichermaßen zu ermöglichen. Tabelle 2 enthält die realisierbaren Attack-, Decay- und Release-Zeiten.

Decay und Release können direkt in die Register 5 und 6 geschrieben werden, während Attack und Sustain vorher mit I6 zu multiplizieren sind, was einer Linksverschiebung um vier binäre Stellen entspricht. Ein Beispiel:

```

160 A = 2 :REM 16 MS
170 D = 12 :REM 3 S

```

```

180 S = 1
190 R = 10 :REM 1.5 S
200 POKE SID+5,A*16+D
210 POKE SID+6,S*16+R

```

Nach vorläufigem Umgehen des Filters und Setzen der maximalen Lautstärke

```

220 POKE SID+23,0:REM FILTER AUS
230 POKE SID+24,15:REM LAUTSTAERKE

```

führt das Setzen des GATE-Bits:  $240 \text{ POKE SID} + 4,65 : \text{REM GATE AN}$

zu einem vernünftigen Resultat. Zum Abschalten des Klanges kann man einfach das Kontrollregister mit 0 besetzen,  $270 \text{ POKE SID} + 4,0$  doch dann hat DCO1 keine Gelegenheit, der Release-Phase entsprechend auszuklingen, da auch das Kurvenform-Bit (Nr. 6) zurückgesetzt ist. Besser ist also, Bit 6 gesetzt zu lassen:  $250 \text{ GET A}: \text{IF A}\$ = "" \text{ THEN } 160$

```

260 REM WARTE AUF EINE TASTE
270 POKE SID+4,64:REM GATE AUS

```

Die Basic-Zeilen zeigen natürlich keinen effizienten Programmierstil; sie sollen nur die Vorgehensweise darstellen.

Auf die Steuerung des Filters und auf die Anwendung der Leseregister wird im nächsten Teil noch ausführlich eingegangen werden. So ist die Abfrage eines analogen Gebers (zum Beispiel Paddle) erst durch den SID möglich. Die sechs kleinen Beispielprogramme demonstrieren einige Effekte, die man mit den bis hierher beschriebenen Registern realisieren kann.

(Thomas Krätzig/aa)