



**COMPAS**  
vydavatelství a služby  
uživatelům mikročítačů

---

# Základy programování

3. díl

ZVUK A HUDBA

PROGRAMOVANI ZVUKU A HUDBY

---

U V O D

---

Počítač Commodore je vybaven jedním z nejdokonalejších elektronických zvukových syntetizátorů používaných ve výpočetní technice. Má 3 hlasy, samostatné adresovatelné řízení; je možné řídit tvar zvuku od jeho vzniku až po zánik, filtrovat ho a modulovat a vyrábět "bílý šum". Všechny tyto možnosti jsou přímo přístupné uživateli pomocí několika lehce použitelných příkazů a funkcí v BASICU nebo assembleru. To znamená, že můžete vytvořit i složité zvuky nebo hudební skladby pomocí lehce vytvořených programů.

Tato část příručky Vám popíše všechny schopnosti obvodu "SID" 6581, který je použit jako zvukový a hudební syntetizátor počítače C 64.

Nemusíte být ani zkušeným programátorem ani hudebním expertem, abyste ze syntetizátoru dostali skutečně použitelné zvuky i hudbu.

V této kapitole jsou uvedeny i vzorové programy s úplným výkladem, které Vám ulehčí start do samostatné práce.

Přístup do zvukového generátoru je možný použitím příkazu POKE a vložení správné hodnoty do specifikovaného registru (místa v paměti). Úplný seznam použitelných paměťových míst je v příloze.

Postupně, krok za krokem, Vás seznámíme se všemi pojmy tak, abyste na konci mohli vytvářet zvuky podle vlastního přání.

Každá část této kapitoly začíná příkladem s podrobným popisem, který Vám ukáže, jak používat popisované příkazy a registry.

Nejdůležitějším pracovním nástrojem ve zvukových programech je příkaz POKE, který naplňuje určené místo v paměti (MEM) požadovanou hodnotou (NUM):

POKE, MEM, NUM

---

Pro zvukový syntetizátor je v paměti vyhrazena oblast od adresy 54272 (\$ D400) do adresy 54296 (\$ D418). Stačí zapamatovat si jen dolní adresu a k ní připočítat číslo od 0 do 24, pro jednotlivé registry obvodu SID.

Hodnota (NUM), kterou můžete do těchto míst vložit příkazem POKE, musí být v rozsahu 0-255.

Po získání zkušeností můžete používat i funkci PEEK, která naopak přečte hodnotu z určeného místa v paměti.

X=PEEK(MEM)

---

Proměnná X dostane po vykonání této funkce (PEEK je funkce, ne příkaz) hodnotu rovnou obsahu určeného paměťového místa. Samozřejmě, že Váš program bude obsahovat i jiné příkazy v BASICU, jejich podrobné vysvětlení najdete v příslušné kapitole.

Po tomto úvodu můžete zkusit první jednoduchý program. Hlásí Váš počítač "READY"? Napište příkaz NEW, opište následující program a uložte si ho na pásku nebo disk. Až potom ho odstartujte příkazem RUN.

Příklad č. 1:

```
5      S=54272
10     FORL=STOS+24:POKE,0:NEXT
20     POKES+5,9:POKE+6,0
30     POKES+24,15
40     READ HF,LF,DR
50     IF HF ( 0 THEN END
60     POKES+1,HF:POKES,LF
70     POKES+4,33
80     FOR T=1 TO DR:NEXT
90     POKES+4,32:FOR T=1 TO 50:NEXT
100    GOTO 40
110    DATA5, 177, 250, 28, 214, 250
120    DATA5, 177, 250, 25, 177, 250
130    DATA5, 177, 125, 28, 214, 125
140    DATA32, 94, 750, 25, 177, 250
150    DATA28, 214, 250, 19, 63, 250
160    DATA19, 63, 250, 19, 63, 250
170    DATA21, 154, 63, 24, 63, 63
183    DATA25, 177, 250, 24, 63, 125
190    DATA19, 63, 250, -1, -1, -1
```

Následující popis jednotlivých řádků programu a jejich činnosti Vám pomůže pochopit nejen jejich význam, ale může Vám být i dobrou pomůckou při další práci, když se budete chtít zorientovat v problémech, které jste dostatečně nepochopili nebo zapomněli.

Popis příkladu č. 1:

```
5      Nastaví do S adresu prvního registru ve zvukovém obvodu
10     Vynulování všech zvukových registrů.
20     Nastaví registry ATTACK/DECAY pro 1. hlas (A=0, D=9).
       Registry pro SUSTAIN a RELEASE jsou vynulovány.
30     Nastaví hlasitost na maximum.
40     Cte hodnoty pro vyšší a nižší část hodnoty frekvence a
       délku noty.
50     Je-li hodnota vyšší části frekvence menší než 0, skladba
       končí.
60     Příkaz POKE naplní příslušné registry pro vyšší a nižší
       část frekvence.
70     Určení pilového tvaru kmitání pro 1. hlas.
80     Casová smyčka určující délku dané noty.
90     Ukončení 1. hlasu.
100    Návrat na další hodnotu.
110    Údaje pro skladbu, vždy trojice hodnot: vyšší část hodnoty
       frekvence, nižší část hodnoty frekvence a délka trvání pro
       každou notu.
190    Poslední nota písně, -1 signalizuje konec.
```

v  
Rízení hlasitosti

Registr 24 v obvodu SID řídí celkovou hlasitost, která může být nastavena v rozsahu 0 až 15 (nižší čtyři bity). Další čtyři bity jsou určeny pro jiné účely, jejich význam bude vysvětlen později. V této chvíli stačí vědět, že hlasitost se nastavuje v registru 24 v rozsahu 0-15. Viz řádek 30 prvního příkladu.

Frekvence a tvar kmitání

Zvuk je vytvářen vlnivým pohybem vzduchu. Charakteristickými veličinami pro každé vlnění jsou: čas mezi dvěma vrcholy - perioda resp. počet cyklů/sec. - frekvence. Výška zvuku je určena frekvencí kmitání. Zvukový generátor v počítači C 64 používá pro určení frekvence dva registry (FHI a FLO = 16 bitů). V příloze jsou uvedené frekvence pro všechny tóny v rozsahu 8 oktáv. Pokud chcete jinou frekvenci, než je uvedena v tabulce, použijte následujícího postupu pro generování požadované výstupní frekvence FOUT.

$FN = FOUT/0.06097$

Do vyššího registru se vloží hodnota

$FHI = INT(FN/256)$

Do nižšího registru se vloží hodnota

$FLO = FN - (256 * FHI)$

Poznámka: Jde o převod hodnoty frekvence do binárního tvaru s přesností na 16 bitů.

Už v této chvíli jste tedy schopni zahrát jednohlasou libovolnou melodii...

Použití více hlasů

Počítač C64 má v obvodu SID vestavěné tři nezávislé oscilátory pro vytvoření tří samostatných zvuků, anebo pro vytvoření vygenerované trojhlasné hudby.

Náš první příklad používal jen jednoho z nich. V následujícím příkladě necháme zahrát všechny tři hlasy spolu.

Program zapíšete a uložíte na pásku nebo disk. Nezapomeňte napsat příkaz NEW před započítím zapisování programu.

Příklad č. 2:

```
10 S=54272:FORL=STOS+24DPOKEL,ODNEXT
20 DIMH(2,200),L(2,200),C(2,200)
30 DIMFQ(11)
40 V(0)=17:V(1)=65:V(2)=33
50 POKES+10,8:POKES+22,128:POKES+23,244
60 FORI=OTO11:READFQ(I):NEXT
100 FORK=OTO2
110 I=0
120 READNM
130 IFNM=/THEN250
140 WA=V(K):WBVWA-1IFNM(OTHER NM--NM:WA=0:WB=0
150 DRprocento=NM/128:OCprocento=(NM-128*DRprocento)/16
160 NT=NM-128*DRprocento-16*OCprocento
```



```
170 FR=FQ(NT)
180 IF OCprocento=7THEN200
190 FORJ=6TOOCprocentoSTEP-1DFR-256*HFprocento
200 HFprocento=FR/256:LFprocento=FR-256*HFprocento
210 IFDRprocento=1THENH(K,I)=HFprocento:L(K,I)=LFprocento:
C(K,I)=WA:I=I+1: GOTO 120
220 FORJ=1TODRprocento-1:H(K,I)=HFprocento:
L(K,I)=LFprocentoDC(K,I)=WA:I=I+1:NEXT
240 I=I+1:GOTO120
250 IFI)IMTHENIM=I
260 NEXT
500 POKES+5,0:POKES+6,240
510 POKES+12,85:POKES+13,133
520 POKES+24,31
540 FORI=0 TO IM
550 POKES,L(0,I):POKES+7,L(1,I):POKES+14,L(2,I)
560 POKES+1,H(0,I):POKES+8,H(1,I):POKES+15,H(2,I)
570 POKES+4,C(0,I):POKES+11,C(1,I):POKES+18,C(2,I)
580 FORT=IT080:NEXT:NEXT
590 FORT=1TO200:NEXT:POKES+24,0
600 DATA34334,36376,38539,40830
610 DATA43258,45830,48556,51443
620 DATA54502,57543,61176,64814
1000 DATA594,594,591,596,596
1010 DATA1618,587,592,587,585,331,336
1020 DATA1097,583,585,585,585,587,587
1030 DATA1609,585,331,337,594,594,593
1040 DATA1618,594,596,594,592,587
1050 DATA1616,
1060 DATA1607
1999 DATA0
2000 DATA583,585,583,583,327,329
2010 DATA1611,583,585,578,578,578
2020 DATA 196,198,583,326,578
2030 DATA326,327,329,327,329,326,578,583
2040 DATA1606,582,322,324,582,587
2050 DATA329,327,1606,583
2060 DATA327,329,587,331,329
2070 DATA329,328,1609,578,834
2080 DATA324,322,327,585,1602
2999 DATA0
3000 DATA567,566,567,304,306,308,310
3010 DATA1591,567,311,310,567
3020 DATA306,304,299,308
3030 DATA304,171,176,306,291,551,306,308
3040 DATA310,308,310,306,295,297,299,304
3050 DATA1586,562,567,310,315,311
3060 DATA1586,562,567,310,315,311
3070 DATA1568,567,560,311,309
3080 DATA308,309,306,308
3090 DATA1577,299,295,306,310,311,304
3100 DATA562,546,1575,0
```

Popis programu č. 2:

Rádek Popis

10 Do S se vloží adresa prvního registru v obvodu SID.

```
170 FR=FQ(NT)
180 IF OCprocento=7THEN200
190 FORJ=6TOOCprocentoSTEP-1DFR-256*HFprocento
200 HFprocento=FR/256:LFprocento=FR-256*HFprocento
210 IFDRprocento=1THENH(K,I)-HFprocento:L(K,I)-LFprocento:
C(K,I)=WA:I=I+1: GOTO 120
220 FORJ=1TODRprocento-1:H(K,I)-HFprocento:
L(K,I)=LFprocentoDC(K,I)=WA:I=I+1:NEXT
240 I=I+1:GOTO120
250 IFI)IMTHENIM=I
260 NEXT
500 POKES+5,0:POKES+6,240
510 POKES+12,85:POKES+13,133
520 POKES+24,31
540 FORI=0 TO IM
550 POKES,L(0,I):POKES+7,L(1,I):POKES+14,L(2,I)
560 POKES+1,H(0,I):POKES+8,H(1,I):POKES+15,H(2,I)
570 POKES+4,C(0,I):POKES+11,C(1,I):POKES+18,C(2,I)
580 FORT=ITO80:NEXT:NEXT
590 FORT=1TO200:NEXT:POKES+24,0
600 DATA34334,36376,38539,40830
610 DATA43258,45830,48556,51443
620 DATA54502,57543,61176,64814
1000 DATA594,594,591,596,596
1010 DATA1618,587,592,587,585,331,336
1020 DATA1097,583,585,585,585,587,587
1030 DATA1609,585,331,337,594,594,593
1040 DATA1618,594,596,594,592,587
1050 DATA1616,
1060 DATA1607
1999 DATA0
2000 DATA583,585,583,583,327,329
2010 DATA1611,583,585,578,578,578
2020 DATA 196,198,583,326,578
2030 DATA326,327,329,327,329,326,578,583
2040 DATA1606,582,322,324,582,587
2050 DATA329,327,1606,583
2060 DATA327,329,587,331,329
2070 DATA329,328,1609,578,834
2080 DATA324,322,327,585,1602
2999 DATA0
3000 DATA567,566,567,304,306,308,310
3010 DATA1591,567,311,310,567
3020 DATA306,304,299,308
3030 DATA304,171,176,306,291,551,306,308
3040 DATA310,308,310,306,295,297,299,304
3050 DATA1586,562,567,310,315,311
3060 DATA1586,562,567,310,315,311
3070 DATA1568,567,560,311,309
3080 DATA308,309,306,308
3090 DATA1577,299,295,306,310,311,304
3100 DATA562,546,1575,0
```

Popis programu č. 2:

Rádek Popis  
10 Do S se vloží adresa prvního registru v obvodu SID.

Udaje o jedné notě: d = délka v 1/16, O = oktáva (1-7), n = nota v oktávě (1-11) jsou zakódovány do jednoho čísla:

$$(((D*8)+O)*16)+N)$$

#### Rízení více hlasů

Rázování jednotlivých hlasů musí být časově koordinováno. V programu podle příkl. 2 je koordinace zabezpečena:

1. Rozdělením každého taktu na 16 částí.
2. Udaje pro každou šestnáctinu taktu jsou uloženy ve třech samostatných polích.

Hodnoty pro vyšší a nižší byty určující frekvenci jsou vypočítány dělením frekvence stejného tónu ze základní (nejvyšší) oktávy dvěma (řádky 180 a 190).

Naplnění bytu řídicího kmitání je startovacím signálem pro započeti hraní tónu anebo pro pokračování už začatého tónu. Tento byte je určen pro celou skladbu v řádku 40. Uvedený způsob je jen jednou z cest pro řízení více hlasů a každý uživatel si může zvolit vlastní metodu.

#### Změna tvaru kmitání

V hudbě se pro označení kvality zvuku používá pojem barva. Primárně je barva zvuku určena tvarem kmitání. Nejběžnějším tvarem kmitání je sinusovka. Obvod SID 6581 dokáže dále vytvořit pilovitý, trojúhelníkový a pravoúhlý tvar kmitů. Prakticky si můžete účinek změny tvaru kmitů ověřit modifikováním programu v příkladu 1 původní program používá pilovitý průběh. Zkuste změnit hodnotu v řádku 70 z 33 na 17 a v řádku 90 z 32 na 16. Po odstartování programu zjistíte změnu kvality zvuku. Pilovitý tvar průběhu se změnil na trojúhelníkový. Třetí generovaný tvar kmitání je pravoúhlý s měnitelnou šířkou pulzů. Šířka pulzů je definována v registrech č. 2 a 3 (pro první hlas). V registru č. 2 je nižší část hodnoty (LPW=0-255). V registru č. 3 jsou pro určení vyšší části hodnoty šířky pulzu vyhrazeny čtyři vyšší bity (HPW=0-15). Spolu je tedy možné uložit 12ti bitové číslo pro specifikaci šířky pulzu, dekadicky v rozsahu 1 - 2048. Šířka pulzu v procentech je vyjádřena:

$$PWOUT = (PWN/40,95)$$

kde PWN je hodnota uložená v registrech č. 2 a 3:

$$PWN = HPW * 256 + LPW$$

pro PWN = 2048 (reg. 2 = 0, reg. 3 = 8) bude průběh pravoúhlý!

Upravte program č. 1 následovně:

15 POKE S+3,8:POKE S+2,0

V řádku 70 změňte hodnotu na 65 a v řádku 90 na 64 a postupně měňte hodnotu v registru 3 od 1 do 8. Po spuštění programu zjistíte, že se zvuk výrazně mění.

Poslední možný průběh kmitání je náhodného tvaru, tkz. "Bílý šum". Tento průběh je vhodný na vytváření různých zvukových efektů. Chcete-li ho slyšet, vložte do řádku 70 hodnotu 129 a do řádku 90 hodnotu 128.

## Trochu teorie pro pochopení kmitání

Každý periodický kmitavý průběh se dá rozložit - skládá se ze základního sinusového kmitání a dalších harmonických složek sinusového tvaru, jejichž frekvence je celým násobkem základní frekvence. Základní frekvence určuje výšku tónu a v hudební teorii se nazývá 1. harmonická frekvence. Velikost dalších harmonických určuje barvu zvuku.

Skutečný hudební nástroj, např. kytara nebo housle, má velmi komplikované spektrum harmonických frekvencí. I při hraní jediného tónu se toto spektrum může dále měnit. V předchozí části jste si odzkoušeli, jak znějí jednotlivé tvary kmitání, které syntetizátor generuje. Jednotlivé harmonické složky se přitom podílejí následovně:

Trojúhelníkové kmitání obsahuje nepatrné harmonické. Velikost každé harmonické složky je nepřímo úměrná mocnině dané harmonické ... trojúhelníkové kmitání je tvarem nejbližší základnímu sinusovému kmitání.

Pilovitý průběh obsahuje všechny harmonické. Velikost každé harmonické složky je nepřímo úměrná číslu dané harmonické. Např. 2. harmonická má 1/2 amplitudy 1. harmonické ... Pravoúhlé kmitání se čtvercovým tvarem obsahuje nepatrné harmonické, jejichž velikost je nepřímo úměrná číslu harmonické. Ostatní pravoúhlé průběhy mají variabilní počet i velikost harmonických. Při změně šířky pulzu se bude barva tónu měnit velmi výrazně. Když budete modelovat určitý zvuk, máte kromě možnosti změny harmonické struktury, která Vás dovede blízko požadovanému cíli, ještě možnost dalšího doladování pomocí určení typu filtrace, o čemž bude podrobnější popis v následujících částech tohoto návodu.

## Obálkový generátor

Hlasitost hudebního tónu se od okamžiku, kdy ho zachytíte poprvé až po jeho doznění, mění. Na začátku, při rozeznívání, hlasitost stoupá. Rychlost stoupání se nazývá ATTACK (úder). Po dosažení vrcholu klesá hlasitost až na určitou střední hodnotu, na které zůstává po určitou dobu, po které začíná doznívání. Rychlost klesání od maxima po úderu na střední hodnotu se nazývá DECAY (uvolnění), samotná střední hodnota SUSTAIN (znění) a rychlost doznívání RELEASE (ukončení). Každý z uvedených parametrů A, D, S, R (resp. společně ADSR) je možné nastavit příslušným řídicím registrem v obvodu SID.

Zadejte znovu program z prvního příkladu. Změňte v něm řádek 20 následovně:

```
20 POKE S+5,88: POKE S+6,195
```

Registry č. 5 a 6 definují parametry ADSR pro 1. hlas. Přitom hodnota A (ATTACK) leží v horní polovině registru č. 5 (horní NYBBLE), D (DECAY) v jeho dolní polovině. Hodnoty A, D se tedy mohou měnit v rozsahu 0-15 a do registru je třeba uložit dekadicky číslo = A 16+D. Úroveň znění tónu S (SUSTAIN) je zakódována číslem ve vyšší polovině registru č. 6 a R (RELEASE) v jeho dolní polovině. Hodnoty rychlosti A, D, R jsou uvedeny v následující tabulce (S je úroveň hlasitosti 0-15):

Hodnota	rychlost A	rychlost D, R
0	2 MS	6 MS
1	8 MS	24 MS
2	16 MS	48 MS
3	24 MS	72 MS
4	38 MS	114 MS
5	65 MS	204 MS
7	80 MS	240 MS
8	100 MS	300 MS
9	250 MS	750 MS
10	500 MS	1,5 MS
11	800 MS	2,4 S
12	1 S	3 S
13	3 S	9 S
14	5 S	15 S
15	8 S	24 S

V programu z příkladu č. 1 můžete vyzkoušet následující změny, případně zkuste vlastní hodnoty:

Zvuk houslí imituje řádek 20 změněný takto:  
20 POKE S+5,80:POKE S+6,89:REM A=5:D=8:S=0;R=9

Změňte průběh na trojúhelníkový a zvuk xylofonu dostanete:

20 POKE S+5,9: POKE S+6,9: REM A=0:D=9:S=0;R=9  
70 POKE S+4,17  
90 POKE S+4,16: FORT=1 TO 50: NEXT

Změnou kmitání na pravouhlý tvar můžeme modelovat piano:

15 POKE S+3,8: POKE S+2,0  
20 POKE S+5,9 D POKE S+6,0: REM A=0:D=9:S=0;R=0  
70 POKE S+4,65  
90 POKE S+4,64: FORT=1 TO 50: NEXT

Nejzajímavější zvuky jsou ty, které nenapodobují žádný skutečný nástroj, ale jsou jedinečným výtvozem syntetizéru. Zkuste např.:

20 POKE S+5,144: POKE S+6,243: REM A=9:D=0:S=15;R=3

### Filtrování

Obsah harmonických v kmitočtovém spektru je možné dále měnit filtrováním. Obvod SID umožňuje použít tři druhy filtrace, které jsou použitelné samostatně nebo ve vzájemné kombinaci. Program z prvního příkladu nám umožní předvést i tyto možnosti. Do programu vložte řádek 15, kterým se určí mezní frekvence filtrace. Číselná hodnota určující tuto frekvenci je uložena ve dvou řídicích registrech - 21 a 22 (horní, dolní byte). Zapnutí filtru pro první hlas řídí registr č. 23.

V řádku 30 změňte data tak, aby byl použit horní filtr (viz tabulka registrů obvodu SID). Po spuštění programu zjistíte, že nižší tóny mají sníženou hlasitost, kvalita tónu se změnila. Je to zapříčiněno použitím uvedeného typu filtru, který tlumí frekvence nižší, než je určená mezní frekvence.

Obvod SID obsahuje tři typy filtrů. V příkladu byl použit horno-propustný filtr, který dovoluje přechod frekvencí vyšších, než je mezní frekvence, a tlumí frekvence nižší.



Dále je v SID dolno-propustný filtr, který propouští frekvence nižší, než je mezní frekvence, a tlumí frekvence vyšší. Posledním typem je pásmová propust, která propouští úzký svazek frekvencí v okolí mezní frekvence a tlumí všechny ostatní. Nižší část tohoto registru řídí hlasitost, vyšší slouží k řízení filtrů. Bit č. 6 ovládá hornopropustný filtr (0=vyp., 1=zap.), bit č. 5 pásmovou propust a bit č. 4 dolnopropustný filtr.

Nižší tři bity hodnoty mezní frekvence jsou umístěny v registru č. 21 (LCF=0-7) a 8 bitů pro vyšší část je v registru č. 22 (HLF=0-255).

Starostlivým výběrem filtrace můžete změnit harmonickou strukturu každého kmitání, a tím dostat požadovaný zvuk. Místo měněním filtrace během jednotlivých fází ASDR určitého tónu je možné vytvořit velmi zajímavé zvukové efekty.

#### Doplňkové možnosti

Parametry určující funkci obvodu SID mohou být měněny dynamicky během trvání tónu nebo zvuku. Pro tento účel jsou v registru č. 27 a 28 uživateli přístupné digitalizované výstupy z oscilátoru č. 3, resp. obálkového generátoru č. 3. Výstup oscilátoru č. 3 (registr 27) je přímo úměrný generovanému průběhu kmitání. Pro pilovitý tvar kmitání oscilátoru č. 3 se bude hodnota v tomto registru postupně zvětšovat rovnoměrnými přírůstky od 0 do 255 rychlostí určenou frekvencí oscilátoru č. 3. Po dosažení hodnoty 255 klesá na nulu a cyklus se opakuje.

Při výběru trojúhelníkového tvaru se bude hodnota měnit rovnoměrně oběma směry; od 0 do 255 a zpět 255 - 0.

Zvolíte-li pravouhlý tvar kmitů, bude se hodnota v registru měnit skokem z 0 na 255 a naopak.

Konečně při volbě "SUMU" dostanete na výstupu řadu náhodných čísel. Je-li oscilátor č. 3 použit na modulaci, nebudete asi chtít jeho výstup slyšet v generovaném zvuku. Nastavením bitu č. 7 v registru 24 na 0 se zvukový výstup 3. hlasu odpojí. Registr 27 sleduje vždy jen výstup z oscilátoru a není ovlivněn generátorem ASDR. Přístup k tomuto generátoru dovoluje reg. č. 25. Číselná hodnota v tomto registru se mění obdobně jako se mění výstup oscilátoru 3. Výstup z tohoto registru je podmíněn zapnutím oscilátoru. Vibrata (rychlá změny frekvence) je možné dosáhnout přidáním výstupu osc. 3 k frekvenci jiného oscilátoru. Program v následujícím příkladu ilustruje tuto možnost.

```
10 S=54272
20 FORL=0T024:POKES+L,0:NEXT
30 POKES+3,8
40 POKES+5,41:POKES+6,89
50 POKES+14,117
60 POKES+18,16
70 POKES+24,143
80 READFR,DR
90 IFFR=0THENEND
100 POKES+4,65
110 FORT=1TODR 2
120 FQ=FR+PEEK(S+27)/2
130 HF=INT(FQ/256):LF=FQAND255
140 POKES+0,LF:POKES+1,HF
```



```
150 NEXT
160 POKES+4,64
170 GOTO80
500 DATA4817,2,5103,2,5407,2
510 DATA8583,4,5407,2,8583,4
520 DATA5407,4,8583,12,9634,2
530 DATA10207,2,10814,2,8583,2
540 DATA9634,4,10814,2,8583,2
550 DATA9634,4,8583,12
560 DATA0
```

Popis programu:

```
10 Nastavení S na adresu 1. registru obvodu SID.
20 Vynulování všech registrů.
30 Nastavení tvaru kmitů (pravouhlej, šířka 58 procent).
40 Nastavení pro 1. hlas: A=2, D=9, S=5, R=9.
50 Frekvence 3. hlasu.
60 Nastavení trojúhel. průběhu pro 3. hlas.
70 Hlasitost=15, vypnutí výstupu 3. hlasu.
80 Ctení dat pro frekvenci a délku tónu.
90 Je-li frekvence=0, STOP.
100 START PULZ pro řízení kmitočtu 1. hlasu.
110 START měření času.
120 Přepnutí na novou frekvenci použitím osc. 3
130 Výpočet vyšší a nižší části hodnoty frekvence.
140 Uložení hodnot pro frekvenci do registrů.
150 Konec měření času.
160 STOP PULZ řízení 1. hlasu.
170 Návrat pro další notu.
500-550 Frekvence a délky not skladby.
560 Nula pro indikaci konce skladby.
```

V dalším příkladě je imitovaný zvuk sirény. Frekvence oscilátoru č. 1 se dynamicky mění podle výstupu oscilátoru č.3, jehož kmitý mají trojúhelníkovou křivku průběhu.

```
10 S=54272
20 FORL=0T024:POKES+L,0:NEXT
30 POKES+14,5
40 POKES+18,16
50 POKES+3,1
60 POKES+24,143
70 POKES+6,240
80 POKES+4,65
90 FR=5389
100 FORT=1T0200
110 FQ=FR+PEEK(S+27)*3,5
120 HF=INT(FQ/256)DLF=FQ-HF*256
130 POKES+0,LF:POKES+1,HF
140 NEXT
150 POKES+24,0
```

Popis programu:

```
10 S=adresa 1. registru SID.
20 Nulování registrů SID.
30 Nastavení nižší části frekvence 3. hlasu.
40 Nastavení trojúhel. tvaru pro 3. hlas.
```

```
50 Nastavení šířky pulzů 1. hlasu.
60 Hlasitost = 15, vypnutí výstupu z osc. 3
70 S=15, R=0
80 START PULZ pro 1.hlas.
90 Nastavení základní frekvence sirény.
100 START časové smyčky.
110 Výpočet nové hodnoty použitím osc. č. 3 (registr 27).
120 Výpočet hodnot vyšší a nižší části frekvence.
130 Uložení hodnot pro frekvenci.
140 Konec časové smyčky.
150 Vypnutí hlasitosti.
```

Kmitání s náhodným tvarem (Bílý šum) je použitelné pro vytvoření množství zvukových efektů. Další příklad napodobuje tleskání použitím filtrování šumového kmitání.

```
10 S=54272
20 FORL=0T024:POKES+6,0:NEXT
30 POKES+0,240:POKES+1,33
40 POKES+5,8
50 POKES+22,104
60 POKES+23,1
70 POKES+24,79
80 FORN=1T015
90 POKES+4,129
100 FORT=1T0250:NEXT:POKES+4,128
110 FORT=1T030:NEXT:NEXT
120 POKES+24,0
```

Popis programu:

```
50 Nastavení mezní frekvence filtru.
60 Zapnutí filtru.
70 Hlasitost=15, hornopropustný filtr.
80 Počítání 15 tlesknutí.
90 START šum. kmitání.
100 Cekací smyčka ukončí kmitání.
110 Cekací smyčka pro začátek dalšího úderu.
120 Vypnutí hlasitosti.
```

Synchronizace a kruhová modulace

Obvod SID 6581 dovoluje vytvářet i složitější harmonické struktury pomocí synchronizace a kruhové modulace dvou hlasů. Synchronizace je založena na logickém sčítání průběhu dvou kmitání. Pokud je jeden nulový, je nulový i výstup. Následující příklad využívá tohoto způsobu na napodobení zvuku letícího komára.

```
10
20 FORL=0T024:POKES+L,0:NEXT
30 POKES+1,100
40 POKES+5,219
50 POKES+15,28
60 POKES+24,15
70 POKES+4,19
```

```
80 FORT=1T05000:NEXT
90 POKES+4,18
100 FORT=1T01000:NEXT:POKES+24,0
```

Synchronizace je zapnuta v řádku 70 nastavením bitů 0, 1 a 4 registru č. 4 na hodnotu 1. Bit č. 1 řídí synchronizaci mezi hlasy 1 a 3. Bity 0 a 4 mají svou normální funkci při řízení 1. hlasu - bit 0 zapíná hlas a bit 4 určuje trojúhelníkový tvar kmitání.

Kruhová modulace (řízení bitem č. 3 v reg. 4) zamění původní výstup osc. 1 za "Kruhově" modulovanou kombinaci osc. 1 a 3. Tato kombinace vytváří např. zvuky imitující zvon nebo gong. Zkuste program, který vytváří úderý hodin:

```
10 S=54272
20 FORL=0T024:POKES+L,0:NEXT
30 POKES+1,130
40 POKES+5,9
50 POKES+15,30
60 POKES+24,15
70 POKES+4,19
80 FORT=1T05000:NEXT
90 POKES+4,18
100 FORT=1T01000:NEXT:POKES+24,0
```

Synchronizace je zapnuta v řádku 70 nastavením bitů 0, 1 a 4 registru č. 4 na hodnotu 1. Bit č. 1 řídí synchronizaci mezi hlasy 1 a 3. Bity 0 a 4 mají svou normální funkci při řízení 1. hlasu - bit 0 zapíná hlas a bit 4 určuje trojúhelníkový tvar kmitání.

Kruhová modulace (řízení bitem č. 3 v reg. 4) zamění původní výstup osc. 1 za "Kruhově" modulovanou kombinaci osc. 1 a 3. Tato kombinace vytváří např. zvuky imitující zvon nebo gong. Zkuste program, který vytváří úderý hodin:

```
10 S=54272
20 FORL=0T024:POKES+L,0:NEXT
30 POKES+1,130
40 POKES+5,9
50 POKES+15,30
60 POKES+24,15
70 FORL=1T012:POKES+4,21
80 FORT=1T01000:NEXT:POKES+4,20
90 FORT=1T01000:NEXT:NEXT
```

Popis programu:

```
40 Hlas 1: A=0, D=9.
70 Počítání úderu, nastavení trojúhel. kmitání s kruhovou
modulací
80 Casová smyčka ukončí kmitání.
90 Další úder.
```