

UŽIVATELSKÁ PŘÍRUČKA

Hudba a zvuk C64

URČENO PRO POČÍTAČE COMMODORE 64/128

**commotronic**

(HARDWARE & SOFTWARE)



## O B S A H

1.0	Elektronická syntéza zvuku	1
2.0	Teorie programování zvuku na C-64	1
2.1	Adresa, registr & programování	1
2.2	Ušemocný příkaz POKE	2
2.3	Ještě jednou registr	2
2.4	Registry pro frekvenci	3
2.5	Registr pro hlasitost	3
2.6	Registry pro barvu zvuku	4
2.6.1	Registr pro tvar signálu	4
2.6.2	Zvuková brána	4
2.6.3	S obdélníkem jsou komplikace	5
2.7	Obálka tónu - parametry ADSR	5
2.8	Filtry	6
3.0	Praktické programování	8
3.1	Program TON	9
3.2	Program TON 2	9
3.3	Program TUAR SIGNALU	10
3.4	Program OBALKA	11
4.0	Volba hlasů a výšky tónu (not)	12
4.1	Program AKORD	12
4.2	Program STUPNICE	12
4.3	Program PISNICKA	13
5.0	Zvukové efekty	15
5.1	Program PANENKA	15
5.2	Program UYSTREL	15
5.3	Program MOTORY	16
5.4	Program RIZENI	16
5.5	Program ENTERPRISE	17
6.0	Přílohy	18
6.1	Programování registrů	18
6.2	Označení registrů	19
6.3	Tabulka tónů	20



## Úvod

Když jsme se poprvé seznamovali s počítačem Commodore C-64 pomocí her, byli jsme všichni překvapeni škálou a kvalitou zvuků, které dokáže počítač vyluzovat. Po prvním opojení ohromnou nabídkou her pro naši šedesátčtyřku jsme si všichni také zkusili samostatné programování.

Basic V 2, kterým je C-64 vybaven, není příliš "user friendly", neboli uživatelsky přátelský, zvláště při programování barev, zvuků a sprajtů. Také manuál počítače, pro většinu psaný neznámým jazykem a nedostatek technických informací o SIDu pokusy o programování neulehčil. Z těchto důvodů a pro nutnost použití příkazů POKE a řady čísel vzdala pak většina z nás - uživatelů programování zvukového obvodu SID dříve, než se s jeho možnostmi seznámila.

Cílem této příručky je odstranit tento nedostatek a dát všem možnost krok po kroku odhalit vše, co se v SIDu skrývá. Dokonalým zvukem doplněné vlastní programy pak přinesou uspokojení autorům a radost uživatelům.

Commotronic



## 1.0 Elektronická syntéza zvuku

Elektronická výroba zvuků je v dnešní počítačové době doménou takzvaných syntezátorů. To jsou počítače, které řídí podle hudebníkem vytvořeného programu elektronické generátory zvukových frekvencí a elektronické filtry v syntezátoru zabudované. Ty spolu s dalšími elektronickými obvody přístroje dokáží napodobit prakticky jakýkoliv zvuk. Velký počet současně řízených generátorů dává velký počet imitovaných nástrojů, takže dostatečně výkonný syntezátor dokáže napodobit celý orchestr.

Náš počítač Commodore C-64 není tak výkonný jako hudební syntezátor, nicméně dokáže vyrábět tři různé zvuky zároveň. S jejich pomocí pak napodobí také prakticky jakýkoliv zvuk.

Aby počítač vyráběl námi požadované zvuky, musíme mu stanovit všechny parametry, které charakter zvuku ovlivňují.

Je jich celá řada a proto se jimi budeme dále podrobněji zabývat. Nejdříve těmi, které jsou pro výrobu tónů nezbytně nutné, potom těmi, které se využívají k řízení barvy tónů a ke speciálním efektům.

Nezbytně nutné k zaznění jakéhokoliv tónu jsou tyto parametry:

- \* výška tónu,
- \* hlasitost,
- \* tvar signálu,
- \* zvuková obálka.

První dva parametry jsou běžně známy a není je třeba rozebírat. Jen si uvedeme, že výška tónu je dána frekvencí, t.j. počtem kmitů - oscilací za sekundu. Výšku tónu i hlasitost tedy budeme muset při programování zvuků stanovovat.

Druhé dva parametry způsobují, že dokážeme sluchem rozlišit různé hudební nástroje. Tvarů signálu je nekonečně mnoho, náš počítač umí vyrábět (generovat) 4 jejich základní tvary, s jejichž pomocí můžeme dostatečně věrně napodobit téměř vše.

## 2.0 Teorie programování zvuků na C-64

Námi tolik obdivované zvukové schopnosti počítače Commodore C-64 má na svědomí zabudovaný zvukový syntezátor - integrovaný obvod SID - Sound Interface Device 6581.

Ten umí vyrábět zvuky, hluky různých kmitočtů a s jejich pomocí napodobit téměř jakýkoliv zvuk.

S ním a jeho programováním se budeme dále seznamovat.

### 2.1 Adresa, registr & programování

Paměť našeho C-64 si můžeme přirovnat ke skříni plné zásuvek -



paměťových míst, do kterých my počítači ukládáme instrukce a data a počítač pak do nich nahlíží a provádí podle nich to, co po něm požadujeme. Každá zásuvka má svoje číslo - adresu. U počítači musí být vše zcela přesně a bezchybně stanoveno, jinak počítač nefunguje. Také při programování zvuku jsou přesně stanoveny adresy, které nám umožňují domlouvat se se SIDem, to je dávat mu pokyny k výrobě tónů. Pod těmito adresami jsou přístupná paměťová místa SIDu, kterým říkáme registry. Podle čísel uložených v jednotlivých registrech, vyrábí SID jednotlivé tóny.

Po uložení všech čísel - parametrů tónu do paměťových buněk SIDu (registrů), přikážeme SIDu, aby námi definovaný tón nechal zaznít. Po odeznění tónu dodáme SIDu parametry dalšího tónu a opět jej necháme zaznít. Tak pokračujeme stále dokola u jednoho, dvou nebo maximálně tří hlasů, které může SID nechat najednou znít.

## 2.2 Ušemocný příkaz POKE

U programovém jazyce BASIC U 2.0, kterým je C-64 vybaven, používáme pro ukládání čísel do registrů příkaz **POKE** *adresa, číslo*.

Příkaz vyjme ze zásuvky-registru, jejíž číslo jsme v příkazu určili lístek se starým číslem a vloží tam nový s námi požadovaným číslem. Nejlépe si to ukážeme příkladem.

Barva rámečku obrazovky je v C-64 uložena na adrese 53280. Normálně je zde uloženo číslo 14 - standardní barva, kterou máme na obrazovce při zapnutí počítače. Pokud chceme změnit barvu rámečku musíme ze zásuvky s číslem 53280 vyjmout lístek s číslem 14 a uložit tam nový, s číslem námi požadované barvy.

To se dělá právě s použitím příkazu POKE.

Takže zapište na obrazovku příkaz **POKE 53280,2**, stiskněte <RETURN>.

Barva rámečku se změní na červenou.

Zkuste si i ostatní barvy, označené čísly 0 - 15!

Zcela stejně funguje i ukládání čísel do registrů SIDu.

## 2.3 Ještě jednou registr

Základem programování SIDu je ukládání parametrů jednotlivých tónů do jeho registrů a jejich následné využití k výrobě tónu.

Registr pro ukládání barvy rámečku obrazovky měl svoje přesné číslo - 53280.

Základní adresou registrů SIDu je adresa 54272. Pro snazší programování přidělíme jednotlivým adresám SIDu označení ze dvou písmen, které se nám bude lépe pamatovat než pětimístná adresa.

Takže základní adresa - první adresa SIDu bude mít označení S1. Casem si zapamatujeme adresu tohoto registru:

**S1 = 54272**



Známe tedy základní adresu SIDu. Bude dobré, když si řekneme, jaké mají registry jejichž čísla od této adresy při práci se SIDem odvozujeme, praktický význam.

## 2.4 Registry pro frekvenci

Úroveň tónu je fyzikálně určena kmitočtem - frekvencí. Frekvence - jeden z nejdůležitějších parametrů tónu, je ukládána do prvních dvou registrů SIDu.

Pohled to tabulky tónů nám však odhalí skutečnost, že některým tónům odpovídají velká čísla, jako 40000 i více. Tak velká čísla do registrů SIDu ani do paměti počítače příkazem POKE uložit nejdu. Do jednotlivých zásuvek - registrů - se dají ukládat jen čísla která nejsou větší, než 255!

Proto si velká čísla musíme rozdělit na 2 menší čísla a uložit je do dvou registrů.

Tato dvě čísla nazýváme LOW-Byte a HIGH-Byte a profesionálové si tyto názvy zkrátily na Lo-byte a Hi-byte (foneticky lou-bajt a haj-bajt).

Počítač tato čísla dokáže z velkého čísla spočítat sám podle následujících vzorců:

$HI = INT(\text{číslo}/256),$

$LO = \text{číslo} - (HI\text{-bajt} * 256)$

Můžete si to prakticky odzkoušet a porovnat si výsledky s tabulkou zvuků, kde již máte tyto dvě hodnoty vypočteny.

Pro pochopení postupu programování je dobré prostudovat si také na str. 57 tabulku zvukových registrů SID a všimnout si, které parametry se musí pro SID takto rozdělovat!

Takže se po tom složitém vysvětlování dostáváme konečně k tomu, abychom si označili zkratkou první dva registry.

Protože slouží k ukládání frekvence v podobě dvou čísel - Lo a Hi bajtu, označíme si je FL a FH.

FL bude nižší bajt frekvence, FH bude vyšší bajt frekvence.

$FL = S1 = 54272$  (nižší bajt frekvence)

$FH = S1 + 1 = 54273$  (vyšší bajt frekvence)

## 2.5 Registr pro hlasitost

Hlasitost se dá SIDem nastavit na 16 úrovní. 0 značí vypnutý zvuk (regulátor hlasitosti na minimum), 15 značí maximum (regulátor hlasitosti na maximum). Parametr hlasitosti budeme označovat HL a budeme jej ukládat do registru 24.

$HL = S1 + 24 = 54296$

Hlasitost nastavujeme pro všechny 3 kanály - hlasy najednou. Maximální hlasitost dostaneme, když do registru HL uložíme číslo 15:

**POKE HL,15**



Pokud chceme SID vypnout, neboli "ztlumit na minimum", provedeme to poukem:

### **POKE HL,0**

Obvykle nastavujeme hlasitost pevně na začátku hudebního programu. Nicméně změnou hlasitosti v průběhu znění melodie je možno získat zajímavé efekty.

## **2.6 Registry pro barvu zvuku**

### **2.6.1 Registr pro tvar signálu**

C-64 může díky SIDu vyrábět tyto základní tvary signálu:

- trojúhelník,
- pilu,
- obdélník,
- šum (hluk) - nepravidelný signál.

Tvar signálu ovlivňuje podstatně barvu vydávaného tónu.

Pokud chceme nastavit jako tvar signálu například trojúhelník, musíme zadat následující příkaz:

### **POKE TS, 16 + 1**

Číslo 16 určuje tvar signálu, 1 znamená zapnutí zvukové brány.

Bývá zvykem stanovovat v programu tvar signálu jako poslední a zároveň spouštět zvukovou bránu, což se provede například výše uvedeným poukem. Jak vyplývá z popisu registrů SIDu na str. 57, určujeme tvar signálu těmito hodnotami:

trojúhelník - 16  
 pila - 32  
 obdélník - 64  
 hluk (šum) -128.

Z uvedených čísel vyplývá, že každý tvar je v SIDu určen pomocí jednoho bitu v registru 4. Tvar signálu si označíme TS a budeme jej ukládat do registru 4.

$$TS = S1 + 4 = 54276$$

### **2.6.2 Zvuková brána**

Jak funguje již několikrát zmíněná zvuková brána?

Při práci se SIDem uložíme do registrů požadované parametry tónů, pak je necháme zaznít, uložíme parametry dalšího tónu a tak dále stále dokola.

Prakticky však musíme zajistit ještě jednu věc. Před ukládáním parametrů tónu do registrů musíme nastavit bránu (bit 0 uvedeného registru) na úroveň LOW. Jinak by uložení prvního parametru nového tónu do registru považoval SID za příkaz ke generování dalšího tónu a nepočkal by, až mu zadáme všechny parametry. Jaký zmatek by to způsobilo, si můžeme vyzkoušet úpravou některého z dodaných demo-programů.



Teprve po naplnění všech registrů, nutných pro vytvoření námi požadovaného zvuku parametry naplníme také registr 4 a nastavíme zvukovou bránu na HIGH, abychom odstartovali generování námi požadovaného tónu. Tato zásada je respektována ve všech popsáných DEMO-programech.

Zvolíme proto hodnotu odpovídající zvolenému tvaru signálu, přičteme k ní 1 pro zapnutí zvukové brány a celé číslo zapíšeme do registru dříve uvedeným poukem. U tom okamžiku se spustí proces generování naprogramovaného tónu.

### 2.6.3 S obdélníkem jsou komplikace

Pokud si jako tvar signálu vybereme obdélník (64), musíme ještě zapsat do SIDu pomocný parametr mezi 0 a 4095 pro tzv. obsah mezery (poměr horní a dolní úrovně signálu, nebo poměr signál/mezera).

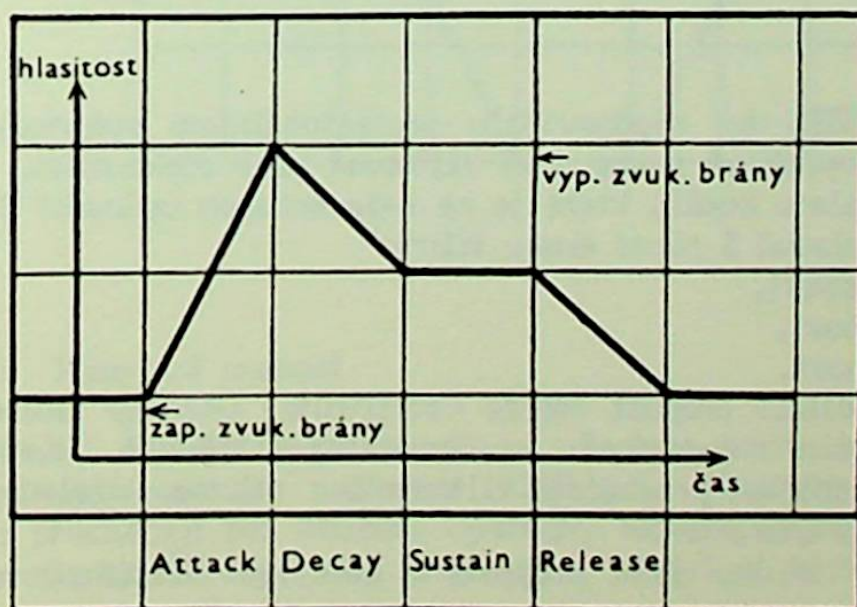
Číslo 4095 je opět větší, než se vejde do jednoho registru ( tam vleze nejvýše číslo 255). Proto je musíme rozdělit do dvou registrů 2 a 3. Zvolíme si také označení pro tento parametr. Myslím, že **P**, jako poměr by vyhovovalo. Co říkáte? A protože zde budeme mít zase LOW a HIGH bajt, doplníme označení na **PL** a **PH**.

**PL** = **S1** + 2 - 54274 (LO-byte poměru pulz/mezera)

**PH** = **S1** + 3 - 54275 (HI-byte poměru pulz/mezera)

### 2.7. Obálka tónu - parametry ADSR

Náběh a průběh tónu, ovlivňují parametry dále značené jako ATTACK, DECAY, SUSTAIN a RELEASE. Jsou ukládány do registrů 5 a 6. Pro pořádek si budeme tyto 4 parametry označovat jejich prvními písmeny, **A, D, S, R**. Jejich skutečný vliv na obálku tónu nám přibližuje obr. 1.



Obr.1 Obálka tónu ADSR



Uvidíme, že tyto parametry mají vliv na hlasitost v průběhu znění tónu. Nasazení (ATTACK) tónu určuje čas, který tón potřebuje pro dosažení maximální (předem nastavené) hlasitosti. Tento parametr je různý pro dechové nástroje a například kytaru.

Odeznění (DECAY) je hodnota pro rychlost, s jakou se hlasitost po dosažení maximální hlasitosti snižuje na setrvalou hodnotu. Má význam opět zvláště u dechových nástrojů, zatím co u kytary se nevyskytuje.

Setrvání (SUSTAIN) představuje hlasitost, na které se tón delší dobu drží beze změny. Je tomu tak například u dechových nástrojů trubky nebo flétny.

Doznění (RELEASE) určuje dobu, po kterou tón po vypnutí zvukové brány doznívá k nule. Tato doba je odlišná například pro trubku a pro kytaru.

Hodnoty pro nasazení (ATTACK, A) tónu a jeho odeznění (DECAY, D), které musí být nastaveny pro každý hlas tak jako tvar signálu, jsou ukládány do jednoho registru jedním číslem.

Abychom si udělali ve věcech pořádek, označíme si tento registr AD.

$$AD = S1 + 5 = 54277$$

Budeme si pamatovat, že pro společné zadávání parametrů využíváme vzorec:

$$P = A * 16 + D$$

Další registr slouží k zápisu parametrů SUSTAIN, S a RELEASE, R.

Registr pro parametry S a R si označíme zkratkou SR.

$$SR = S1 + 6 = 54278$$

Ke sloučení obou parametrů do jednoho čísla a jeho zápisu do registru opět použijeme vzorec:

$$P = S * 16 + R$$

Pokud by vás zajímalo, jak to, že se vlezou dva parametry do jednoho registru, tak je to tím, že je programujeme hodnotami 0 - 15, na což stačí vždy 4 bity osmibitového registru SIDu a 2 x 4 bity dvou parametrů je proto možno uložit do jednoho registru s velikostí 8 bitů.

## 2.8 Filtry

Možnosti SIDu by neodpovídaly profesionálnímu syntezátoru, kdyby nebyl schopen vyrábět zvuky také filtrovat přes elektronické filtry. Ty dále rozšiřují paletu zvuků, které je ze sebe schopen vyloudit C-64.

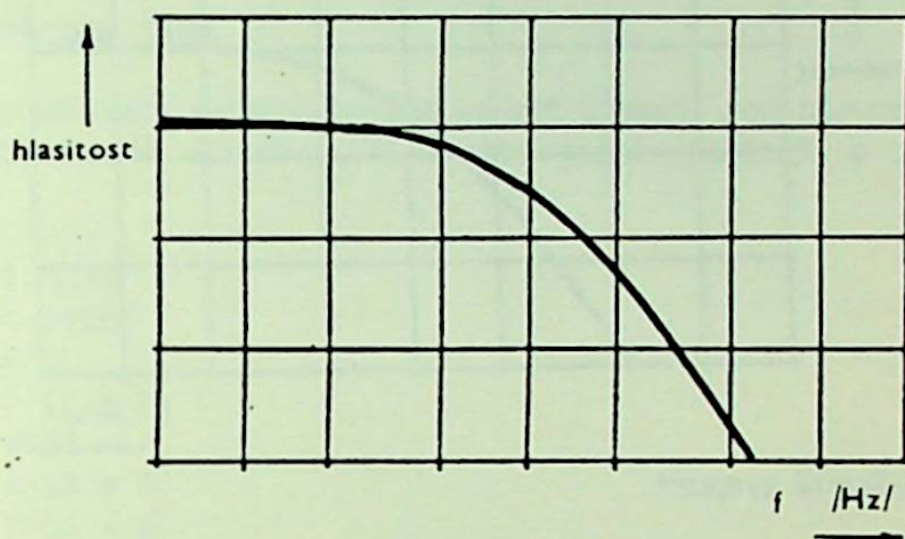
SID dokáže simulovat 3 různé druhy filtrů:

- hloubkovou propust,
- pásmovou propust,
- výškovou propust.

Hloubková (dolní) propust odřeže (odfiltruje) všechny kmitočty, které leží nad zadanou frekvencí se směrnici 12dB/okt. Kmitočty pod stanoveným kmitočtem procházejí filtrem bez útlumu. Výsledný zvuk má objemný a temný charakter.

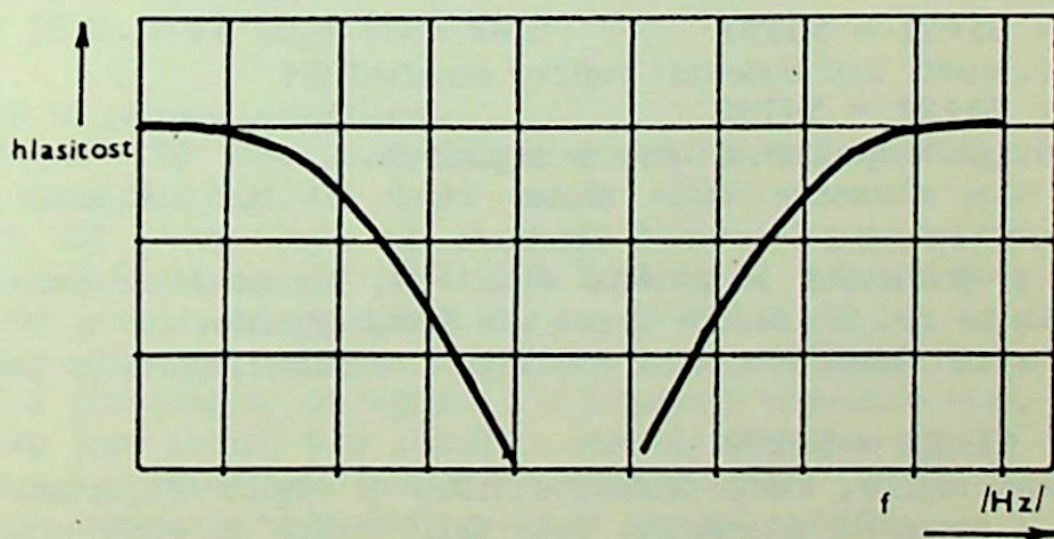
Průběh propustné křivky dolní propusti - závislost hlasitosti na kmitočtu je na obr. 2.





**Obr.2** Hloubková propust .

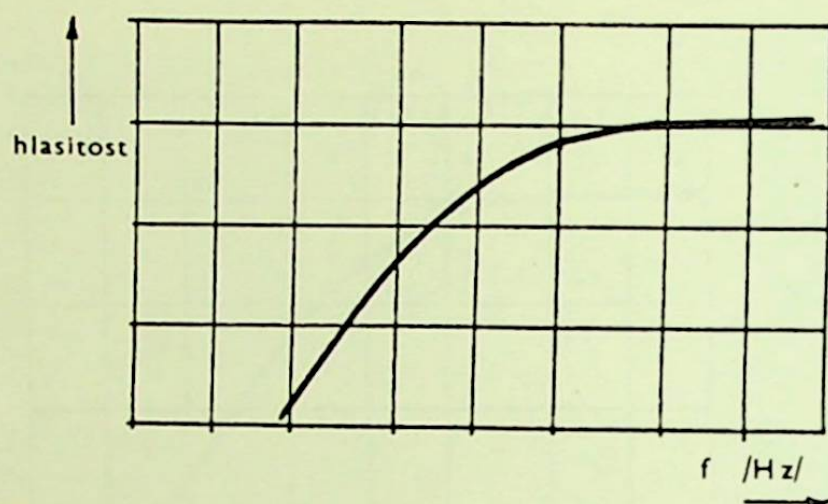
Pásmová propust odřeže kmitočty, které leží pod a nad stanoveným frekvenčním rozsahem se směrnicí 6 dB/okt. Výsledný tón zní dutě. Průběh propustné křivky pásmové propusti - závislost hlasitosti na kmitočtu je na obr. 3.



**Obr.3** Pásmová propust

Ušková (horní) propust odřeže všechny kmitočty, které leží pod zadanou frekvencí se směrnicí 12dB/okt. Kmitočty nad stanovenou frekvencí procházejí bez útlumu. Výsledný tón zní tenče až ostře. Průběh propustné křivky horní propusti - závislost hlasitosti na kmitočtu je na obr. 4.





**Obr.4** Úyřková propust

Vřechny typy filtrů je možno vzájemně kombinovat a získat tak zajímavé zabarvení tónu.

Společně s filtry je obvykle využívána i rezonance, umožňující zdůraznění propouřtěného pásma a obohacení rejstříku zvuků.

Pro využití filtrů jsou důležité registry 21-24.

První tři bity registru 21 slouží k zápisu nižšího bajtu frekvence filtru. Pro označení by bylo vhodné použít písmeno F, to jsem však již použili pro označení frekvence. Proto jsme si pro tento registr vybrali označení RL.

$$RL = S1+21 = 54297$$

Druhý, vyšší bajt bude mít logicky označení RH

$$RH = S1+22 = 54298$$

Další registr spojený s filtry je registr 23.

Ten nám stanovuje číslo hlasu, který má být filtrován a velikost rezonance.

Jeho programování je poněkud složitější, ale pokud si vezmete k ruce tabulku na str. S7, nebude to pro Vás žádný problém.

Registr 23 slouží pro práci s filtry a rezonancí, proto je jeho označení FR.

$$FR = S1+23 = 54299$$

Poslední registr, který souvisí s filtry je registr 24, který používáme také k nastavení hlasitosti. Jeho další funkce je výběr typu filtru. K tomu slouží 4 horní bity registru. Jak je programujeme, známe již z popisu registrů AD a SR.

### 3.0 Praktické programování

A nyní dost teorie. Je nejvyšší čas na to, abychom si na prvním programu prakticky ukázali jak se zvuk programuje.



### 3.1 Program TON

Program máte na programové kazetě/disketě pod názvem TON. Nemusíte jej proto pracně opisovat. Natáhněte jej do počítače a spusťte příkazem RUN.

```

5  REM TON
10 S1 = 54272
20 FL = S1
30 FH = S1 + 1
40 TS = S1 + 4
50 AD = S1 + 5
60 SR = S1 + 6
70 HL = S1 + 24
80 A=1: D=9
90 S=4: R=4
100 POKE HL,15
110 POKE AD,A*16+D
120 POKE SR,S*16+R
130 POKE FH,29: POKE FL,69
140 POKE TS,16+1
150 FOR T=1 TO 500: NEXT
160 POKE TS,16: POKE AD,0: POKE SR,0

```

Pro jistotu si program vysvětlíme.

U řádků 10 - 70 jsme si definovali adresy jednotlivých registrů tak, jak jsme si dříve určili.

U řádků 80 a 90 jsme si stanovili hodnoty parametrů, tvořících "obálku" výsledného tónu.

U řádku 100 je nastavena hlasitost.

U řádku 110 a 120 jsou nastaveny parametry tvořící "obálku" tónu.

U řádku 130 jsou uloženy do registrů 2 a 3 hodnoty frekvence tónu.

U řádku 140 jsme zvolili tvar signálu a otevřeli zvukovou bránu.

Řádek 150 určuje dobu trvání tónu s využitím smyčky.

U řádku 160 jsou vypnuty registry pro tvar signálu a obálku.

Po spuštění tohoto programu vám z reproduktoru zazní tón A.

### 3.2 Program TON 2

Tento program podobný programu TON jsme zařadili pro ty, kteří se chtějí do tónů více zaposlouchat. Proto má program smyčku, která se opakuje při každém stisku klávesy.

```

5  REM TON 2
10 S1 = 54272

```



```

20 FL = S1
30 FH = S1 + 1
40 TS = S1 + 4
50 AD = S1 + 5
60 SR = S1 + 6
70 HL = S1 + 24
80 A=5: D=1
90 S=0: R=0
100 PRINT " STISKNI KLAVESU"
110 GET A$: IF A$="" THEN 110
120 POKE HL,15
130 POKE AD,A+16*D
140 POKE SR,S+16*R
150 POKE FH,8: POKE PL,0
160 POKE FH,14: POKE FL,162
170 POKE TS,64+1
180 FOR T=1 TO 200: NEXT
190 POKE TS,0
200 GOTO 110

```

Aby nám bylo vše jasné, program si vysvětlíme.

Rádky 10 - 70 definují adresy jednotlivých registrů. Jde o programový blok, který se bude opakovat u většiny uvedených programů. Proto je u dalších výpisů programů nebudeme uvádět.

U řádku 120 je nastavena hlasitost na maximum, řádek 130 a 140 nastavuje registry pro obálku tónu ADSR. U řádku 150 je nastaven poměr signál/mezera pro obdélníkový tvar signálu, u řádku 160 je nastavena frekvence tónu.

Rádek 180 určuje délku tónu a řádek 190 vypíná zvukovou bránu.

U řádku 200 program skáče na řádek 110 a čeká na stisk klávesy.

U programu využíváme hlas 1 k výrobě tónu s krátkým náběhem a krátkým odezněním po dosažení maxima. Pokud chceme vyzkoušet vliv obálky ADSR na znění tónu, můžeme měnit parametry u řádků 80 a 90.

Pokud změním parametry na A=14 a D=11, dostaneme tón připomínající zvuk hoboje nebo obdobného dřevěného nástroje.

Bude vhodné když budeme delší dobu experimentovat se změnami tvaru signálu a obálkou tónu, abychom získali cit pro jejich nastavování podle vlastních představ.

### 3.3 Program TVAR SIGNALU

Další program nám ukáže, jak rozdílně zní tóny při různých tvarech signálu. Dále je zde využita smyčka FOR - TO - NEXT k tvorbě stoupajícího tónu. Tímto způsobem, zvláště když zrychlíme změnu tónu změnou kroků ve smyčkách, získáme efektní zvuky. Tento program je nahraný pod názvem **TVAR SIGNALU**.



```

80 A=0: D=0
90 S=15: R=0
100 POKE TS,0: POKE TS+7,0: POKE TS+14,0
110 POKE PL,5: POKE PH,8
120 POKE AD,A*16+D: POKE SR,S*16+D
130 POKEFR,0: POKE HL, 15
140 PRINT" TROJUHELNIK"
150 T=16: GOSUB 300
160 PRINT" PILA"
170 T=32: GOSUB 300
180 PRINT" OBDELNIK"
190 T=64: GOSUB 300
200 PRINT" HLUK"
210 T=128: GOSUB 300
220 END
300 POKE FL,0: POKE FH,0
310 POKE TS, T+1
320 FOR I=0 TO 255: FOR J=0 TO 255 STEP 50
330 POKE FL,J: POKE FH,I
340 NEXT I,J
350 POKE TS,T
360 RETURN

```

### 3.5 Program OBALKA

Další program nám objasní funkci generátoru obálky (ADSR). Řádky 10 - 70 opišeme z prvního programu a připojíme k nim dále uvedené.

```

80 A=9: D=9:
90 S=8: R=9
100 H=400
110 POKE AD,16*A+D: POKE SR,16*S+R
120 POKE FR,0: POKE HL,15
130 POKE FL,37: POKE FH,17
140 POKE TS,32+1
150 FOR I=0 TO H: NEXT
160 POKE TS,32

```

Zkusíme zase měnit hodnoty uvedené v řádcích 80 - 90. Tím získáme představu a cit, jaký vliv mají jednotlivé parametry na výsledný tón. Hodnoty A,D,S,R se mohou pohybovat v rozsahu 0 - 15. Pokud je hodnota R alespoň 1, musíme dát pozor, kdy zvukový kanál vypneme, abychom tón "neustřihli", než dozní.



#### 4. Uolba hlasů a výšky tónů (not)

Dále si uvedeme programy, které nám objasní, jak programujeme více hlasů najednou a sled různě vysokých a dlouhých tónů, vytvářející melodii.

##### 4.1. Program AKORD

Pro snazší začátek vlastního programování vícehlasých melodií a zvuků je dále uveden a popsán program, který využívá všech tří hlasů SIDu k imitaci akordu. Zaznamenaný je pod názvem AKORD. Natáhněte jej do počítače a spusťte příkazem RUN.

```

80 A=1: D=0
90 S=13: R=0
100 C=1000
110 POKE AD,A*16+D: POKE SR,S*16+R
120 POKE AD+7,A*16+D: POKE SR+7,S*16+R
130 POKE AD+14,A*16+D: POKE SR+14,S*16+R
140 POKE FR,0: POKE HL,15
150 POKE FL,37: POKE FH,17
160 POKE FL+7,154: POKE FH+7,21
170 POKE FL+14,177: POKE FH+14,25
180 POKE TS,16+1: POKE TS+7,32+1: POKE TS+14,64+1
190 FOR I=1 TO C: NEXT
200 POKE TS,16: POKE TS+7,32: POKE TS+14,64
210 END

```

Protože programujeme 3 hlasy najednou, označení shodných registrů pro různé hlasy by již bylo obtížné. To proto, že počítač při práci s označením proměnných respektuje pouze první dva znaky označení a FH1 a FH2 jsou pro něj shodná označení. Proto jsme zvolili v programu uvedený postup a definovali pouze základní adresy hlasů a registry, které jsou pro všechny hlasy společné.

Ověřte si, které tóny jsme do akordu naprogramovali. Můžete k tomu použít tabulku tónů v příloze. Postup s využitím tabulky je následující:

Chceme naprogramovat notu C-6. V tabulce tónů si najdeme notu C-6 a jí odpovídající parametr - 17821. Ten rozdělíme do dvou částí (ty jsou v tabulce opět uvedeny), které jdou do registrů SIDu zapsat a které použijeme pro vlastní zápis dat do programu.

FH = 69, FL = 157.

##### 4.2. Program STUPNICE

K tomu, abychom mohli programovat melodie na C-64, nemusíme být žádní muzikanti. Dokážeme si to pomocí



příkladu, který následuje. Příklad ukazuje, jak se posloupnost tónů pro počítač programuje.

U příkladu využijeme pro jednoduchost jen jeden hlas ze tří, které máme k dispozici.

Tento program máte nahraný pod názvem STUPNICE.

```

80 A=9: D=0
90 S=4: R=4
100 POKE HL,15
110 POKE AD,A+16*D
120 READ X: READ Y
130 IF Y=-1 THEN POKE TS,0: END
140 POKE FH,X: POKE FL,Y
145 POKE TS,16+1
150 FOR T=1 TO 100: NEXT
160 POKE TS,0
170 FOR T=1 TO 50: NEXT
180 GOTO 40
190 DATA 17,103,19,137,21,237,23,59,26,20,29,69,32,219,34,207
200 DATA -1,-1

```

Pokud chceme změnit zvuk například tak, aby se podobal cembalu, změníme řádek 140 na:

**145 POKE TS,33**

Pomocí tohoto "pouku" se změní tvar signálu z trojhelníku na pilu. Tím získáme signál bohatý na sudé harmonické krmitočky, s charakteristickým zvukem.

Ale volba tvaru signálu není jedinou možností, jak měnit barvu tónu. Změnou obálky tónu můžeme také dosáhnout změny například cembala na banjo. Stačí změnit v řádku 80 A z 9 na 3:

**80 A=3: D=0**

Stejně jako na pravém syntezátoru můžeme i u C-64 průběžně měnit barvu tónu různých imitovaných hudebních nástrojů. Jak se to dělá, a jak k tomuto účelu využíváme jednotlivé registry SIDu, si vysvětlíme dále.

## 4.2 Program PISNICKA

Konečně se dostáváme k finále!

Podle následujícího příkladu můžeme komponovat písničky a pak je přehrávat. Počítač v tomto příkladu využívá jen hlas 1. Ušimněte si, že při využití našeho programového modulu s řádky 10 - 90 je programové řešení zapisování dat do jednotlivých registrů, v našem příkladě FL, FH jednodušší a kratší.

U řádcích 110, 120 a 130 jsou naprogramována data udávající měnící se parametry tónů, t.j. výšku tónu a jeho délku.



```

80 A=15: D=3
90 S=9: R=0
100 POKE HL,15
110 POKE TH,13: POKE TL,15: POKE AD,A+D*16: POKE SR,S+16*R
120 READ H: READ L: READ DT
130 IF X=-1 THEN END
140 POKE FH,H
150 POKE FL,L
160 POKE TS,64+1
170 FOR T= 1 TO DT: NEXT
180 POKE TS,0
190 GOTO 120
200 DATA 17,103,250,21,237,250,26,20,400,21,237,100,26,20,250,
    29,69,250
210 DATA 26,20,250,0,0,250,21,237,250,26,20,250,29,69,1000,
    26,20,250,0,0,250
220 DATA -1,-1,0

```

#### Popis programu

U řádků 10 - 70 jsou definovány adresy jednotlivých registrů.

U řádku 80 a 90 jsou stanoveny hodnoty parametrů obálky tónu ADSR.

U řádku 100 a 110 jsou uloženy do registrů hlasitosti, poměr signál/mezera a registrů pro ADSR parametry námi stanovené hodnoty.

U řádku 120 začíná smyčka čtení dat a přehrávání melodie. Jsou čteny vždy 3 hodnoty najednou. První dvě pro výšku tónu, třetí pro délku tónu.

U řádku 130 kontroluje program, zda již není na konci dat a nemá se ukončit.

U řádků 140 - 150 jsou do registrů ukládány právě načtené hodnoty výšky tónu. U řádku 160 je definován tvar signálu a otevřena zvuková brána.

Řádek 170 je časová smyčka, která určuje délku tónu.

Řádek 180 vypíná zvukovou bránu a u řádku 190 program se program vrací na řádek 120 k načtení nových hodnot výšky tónu a jeho délky.

U řádků 200 - 210 jsou data, v nichž je postupně za sebou stanovena dvěma hodnotami výška tónu a třetí hodnotou jeho délka.

U řádku 220 uvedená data slouží k ukončení programu podle testu u řádku 130. Musí obsahovat 3 hodnoty, neboť načítání dat probíhá po trojicích a program by nám při nedodržení počtu oznámil chybu "OUT OF DATA ERROR".

Pokud do datových řádků zadáme vlastní data, získáme vlastní melodii. Pociťit se však musíme u zadávání délky tónů parametrem DT, aby odpovídala rytmu melodie. Pokud budeme opisovat melodii z not, bude nám dobrým vodítkem délka tónů z notového zápisu.



## 5 Zvukové efekty

Na rozdíl od hudby slouží zvukové efekty k podmalování děje na obrazovce (exploze kosmické lodi), nebo mají uživatele programů informovat nebo varovat (třeba když hrozí nebezpečí vymazání diskety). Dále jsou uvedena některá doporučení, která mohou programování ulehčit.

- \* Měňte hlasitost tónu v době, kdy zní. Docílíte tak efektu, který se podobá ozvěně.
- \* Skákejte rychle mezi dvěma tóny tam a zpět. Získáte tak tremolo.
- \* Vyzkoušejte nejružnější kombinace tvarů signálu.
- \* Pořádně si vyzkoušejte různé průběhy obálky tónu ADSR.
- \* Pomocí rozdílného programování tří hlasů C-64 (například jeden hlas necháme znít dlouho stejně, zatím co druhé dva se mění) docílíme také překvapující efekty.
- \* Užívejte obdélníkový tvar signálu a měňte poměr signál/mezera.
- \* Zaexperimentujte si s hlukovým generátorem při výrobě zvuků exploze, kroků, výstřelů atd.
- \* Měňte rychle výšku tónu (frekvenci) přes více oktáv.
- \* Užívejte při experimentování základní programové moduly dle našich příkladů. Ustaníte si tak práci a získáte lepší přehled o běhu programů.

Dále uvedené příklady můžete dle vlastní potřeby zabudovat do vlastních programů jako podprogramy.

### 5.1 Program PAHENKA

```
80 A=10: D=0
90 S=15: R=0
100 POKE HL,15: POKE TH,13: POKE TL,15: POKE AD,A+D*16: POKE
SR,S+16*R
130 POKE TS,64+1
140 FOR X=250 TO 0 STEP -2: POKE FH,40: POKE FL,X: NEXT
150 FOR X=150 TO 0 STEP -4: POKE FH,40: POKE FL,X: NEXT
160 POKE TS,0
```

### 5.2 Program UYSTREL

Dále uvedený program se hodí jako zvukový doprovod vlastních her.

```
80 A=0: D=15
90 S=0: R=0
```



```

100 FOR X=15 TO 0 STEP -1
110 POKE FH,40: POKE FL,200
120 POKE TS,128+1
130 NEXT
140 POKE TS,0

```

### 5.3 Program MOTORY

Příklad ukazuje, jak je možno při využití všech tří hlasů zapisovat data do registrů pomocí smyčky.

```

100 S1 = 54272
110 FOR K = 0 TO 24: READ X: POKE S1 + K, X: NEXT
120 DATA 9,2,0,3,0,0,240
130 DATA 12,2,0,4,0,0,192
140 DATA 16,2,0,6,0,0,64
150 DATA 0,30,243,31: REM FILTR
160 POKE S1 + 4,64+1: POKE S1+11,64+1: POKE S1+18,64+1
170 GET A$: IF A$ = "" THEN 170
180 POKE S1+4,64: POKE S1+11,64: POKE S1+18,64

```

Tento program využívá všech tří hlasů SIDu. Data do registrů SIDu jsou uložena v řádku 110.

V řádku 160 jsou stanoveny tvary signálu pro všechny 3 hlasy a otevřeny všechny 3 zvukové brány. V řádku 170 čeká program na stisk klávesy a v řádku 180 jsou uzavřeny zvukové brány.

### 5.4. Program RIZENI

V dalším příkladě se řídí frekvence tónu v závislosti na průběhu obalové křivky tónu. K tomu je využit hlas 3, neboť pouze u tohoto hlasu lze číst stav obálky tónu. Doporučujeme také experimentování v řádku 100.

```

80 A=9: D=9
90 S=9: R=9
100 H=30
110 POKE FR,0: POKE HL,15
120 POKE AD,A*16+D: POKE SR,S*16+R
130 POKE TS,32+1
140 FOR I=1 TO H: POKE FH, PEEK(54300): NEXT
150 FOR I=0 TO R*4: POKE FH, PEEK(54300): NEXT
150 POKE TS,32

```



## 5.5. Program ENTERPRISE

Nakonec si necháme zaznít zvukový signál kosmické lodi ENTERPRISE: Zvídavým doporučujeme, aby si program prostudovali stejně jako předchozí.

```

80 A=15: D=0:
90 S=8: R=13:
100 H=8000
110 POKE FR,0: POKE HL,15
120 POKE FL,0: POKE FH,30
130 POKE FL+7,0: POKE FH+7,1
140 POKE FL+14,0: POKE FH+14,100
150 POKE AD,16*A+D:POKE SR,S*16+R
160 POKE TS,128+1: POKE TS+14,23
170 FOR I=0 TO H: NEXT
180 POKE TS,128: POKE TS+14,16

```

Závěr našeho minikurzu programování zvuku na C-64 nabízí na kazetě/disketě program MINISYNTZATOR.

Tento program umožňuje pohodlnější experimentování s parametry zvuků.

Jeho menu nabízí:

- \*možnost nastavení parametrů,
- \*hraní na simulovaném klávesovém nástroji,
- \*zápis melodie o délce až 1000 tónů na disketu/kazetu,
- \*natažení a přehrávání melodie ze záznamu,
- \*opětné přehrávání právě zahrané melodie.

Uzhledem k délce není zde uveden výpis programu. Ale protože je program celý v Basicu, můžete si jej prostudovat na obrazovce, nebo si jej vytisknout na tiskárně.

Uvedené příklady vám určitě daly dostatečné znalosti o programování SIDu. Nyní se již můžete pustit sami do programování hudby a zvuků na C-64. Přejeme vám hodně zábavy!

Commotronic květen 91



## 6 Přílohy

### 6.1 Programování registrů

Nastavení výšky tónu:

```
P =(0 až 65535)
HI = INT(P/256)
POKE FH,HI
POKE FL,P-(256*HI)
```

Nastavení tvaru signálu a zvukové brány:

```
P=16(nebo 32,64,128)
POKE TS,P+1
```

Uypnutí brány:

```
P =(16,32,64,128)
POKE TS,P
```

Nastavení šířky pulsu (jen u obdélníku, také P=64):

```
P =(0 až 65535)
HI = INT(P/256)
POKE PH,HI
POKE PL,P-(256*HI)
```

Attack a Decay:

```
P =(0 až 15)
POKE AD,A*16+D
```

Sustain a Release:

```
P =(0 až 15)
POKE SR,S*16+R
```

Hlasitost a mód filtru:

```
P =(0 až 15)
P1=(0 až 7)
POKE HL,P+16*P1
```

Zapnutí filtru a rezonance:

```
P =(0 až 15)
P1=(0 až 15)
POKE FR,P+16*P1
```

Uypnutí filtru:

```
POKE FR,0
```



Frekvence filtru:

```
P = (0 až 2047)
HI = INT(P/8)
POKE RH, HI
POKE RL, P-8*HI
```

## 6.2 Označení registrů

Základní adresa SIDu:	<b>S1 = 54272 (\$D400)</b>
Frekvence (kmitočet) tónu - LO-bajt:	<b>FL</b> FL=S1(+7,+14)
Frekvence (kmitočet) tónu - HI-bajt:	<b>FH</b> FH=S1+1(+8,+15)
Šířka pulsu - LO-bajt:	<b>PL</b> PL=S1+2(+9,+16)
Šířka pulsu - HI-bajt:	<b>PH</b> PH=S1+3(+10,+17)
Tvar signálu+zvuk. brána:	<b>TS</b> TS=S1+4(+11,+18)
Attack a Decay:	<b>AD</b> AD=S1+5(+12,+19)
Sustain a Release:	<b>SR</b> SR=S1+6(+13,+20)
Frekvence filtru - LO-bajt:	<b>RL</b> RL=S1+22
Frekvence filtru - HI-bajt:	<b>RH</b> RH=S1+21
Zapnutí filtrů+rezonance:	<b>FR</b> FR=S1+23
Hlasitost a druh filtru:	<b>HL</b> HL=S1+24



## 6.3 Tabulka tónů

U této příloze je úplný seznam not, k nim náležejících frekvencí tónů a jim odpovídajících parametrů pro SID. Parametry jsou dále rozděleny do dvou čísel LO-bajt a HI-bajt, která musíme pro zaznění zvoleného tónu uložit do registrů FL a FH zvukového obvodu SID.

Číslo	Nota-oktáva	Frekvence (Hz)	Parametr	HI-bajt	LO-bajt
0	C-0	16.4	278	1	22
1	C#-0	17.3	295	1	39
2	D-0	18.4	313	1	57
3	D#-0	19.4	331	1	75
4	E-0	20.6	351	1	95
5	F-0	21.8	372	1	116
6	F#-0	23.1	394	1	138
7	G-0	24.5	417	1	161
8	G#-0	26.0	442	1	186
9	A-0	27.5	468	1	212
10	A#-0	29.1	496	1	240
11	H-0	30.9	526	2	14
12	C-1	32.7	557	2	45
13	C#-1	34.6	590	2	78
14	D-1	36.7	625	2	113
15	D#-1	38.9	662	2	150
16	E-1	41.2	702	2	190
17	F-1	43.7	743	2	231
18	F#-1	46.2	788	3	20
19	G-1	49.0	834	3	66
20	G#-1	51.9	884	3	116
21	A-1	55.0	937	3	169
22	A#-1	58.3	992	3	224
23	H-1	61.7	1051	4	27
24	C-2	65.4	1114	4	90
25	C#-2	69.3	1180	4	156
26	D-2	73.4	1250	4	226
27	D#-2	77.8	1325	5	45
28	E-2	82.4	1403	5	123
29	F-2	87.3	1487	5	207
30	F#-2	92.5	1575	6	39
31	G-2	98.0	1669	6	133
32	G#-2	103.8	1768	6	232
33	A-2	110.0	1873	7	81
34	A#-2	116.5	1985	7	193
35	H-2	123.5	2103	8	55
36	C-3	130.8	2228	8	180
37	C#-3	138.6	2360	9	56



Číslo	Nota-oktáva	Frekvence (Hz)	Parametr	HI-bajt	LÜ-bajt
38	D-3	146.8	2500	9	196
39	D#-3	155.6	2649	10	89
40	E-3	164.8	2807	10	247
41	F-3	174.6	2974	11	158
42	F#-3	185.0	3150	12	78
43	G-3	196.0	3338	13	10
44	G#-3	207.7	3536	13	208
45	A-3	220.0	3746	14	162
46	A#-3	233.1	3969	15	129
47	H-3	246.9	4205	16	109
48	C-4	261.6	4455	17	103
49	C#-4	277.2	4720	18	112
50	D-4	293.7	5001	19	137
51	D#-4	311.1	5298	20	178
52	E-4	329.6	5613	21	237
53	F-4	349.2	5947	23	59
54	F#-4	370.0	6301	24	157
55	G-4	392.0	6676	26	20
56	G#-4	415.3	7072	27	160
57	A-4	440.0	7493	29	69
58	A#-4	466.2	7939	31	3
59	H-4	493.9	8411	32	219
60	C-5	523.3	8911	34	207
61	C#-5	554.4	9441	36	225
62	D-5	587.3	10002	39	18
63	D#-5	622.3	10597	41	101
64	E-5	659.3	11227	43	219
65	F-5	698.5	11894	46	118
66	F#-5	740.0	12602	49	58
67	G-5	784.0	13351	52	39
68	G#-5	830.6	14145	55	65
69	A-5	880.0	14986	58	138
70	A#-5	932.3	15877	62	5
71	H-5	987.8	16821	65	181
72	C-6	1046.5	17821	69	157
73	C#-6	1108.7	18881	73	193
74	D-6	1174.7	20004	78	36
75	D#-6	1244.5	21193	82	201
76	E-6	1318.5	22454	87	182
77	F-6	1396.9	23789	92	237
78	F#-6	1480.0	25203	98	115
79	G-6	1568.0	26702	104	78
80	G#-6	1661.2	28290	110	130
81	A-6	1760.0	29972	117	20
82	A#-6	1864.7	31754	124	10



Číslo	Nota-oktáva	Frekvence (Hz)	Parametr	HI-bajt	LO-bajt
83	H-6	1975.5	33642	131	106
84	C-7	2093.0	35643	139	59
85	C#-7	2217.5	37762	147	130
86	D-7	2349.3	40008	156	72
87	D#-7	2489.0	42387	165	147
88	E-7	2637.0	44907	175	107
89	F-7	2793.2	47578	185	218
90	F#-7	2960.0	50407	196	231
91	G-7	3136.0	53404	208	156
92	G#-7	3322.4	56580	221	4
93	A-7	3520.0	59944	234	40
94	H-7	3729.3	63508	248	20

Tato tabulka není ve všech případech závazná. U případě, že využíváte více hlasů hrajících stejný tón, doporučujeme, abyste druhý a třetí hlas, mírně posunuli změnou LO-bajtu. Dostanete tak plnější tón.



**COMMOTRONIC, Jesenická 67, 787 01 Šumperk**