

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Václav Vlček

### **Tvorba zvuku pomocí aditivní a FM syntézy**

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Tomáš Holan, Ph.D.

Studijní program: Informatika, Obecná informatika

2007

Chtěl bych poděkovat vedoucímu práce za krásné téma, pomoc při jeho realizaci a také nalezené gramatické chyby a překlepy v této práci, na které mne upozornil.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 15.5.2007

Václav Vlček

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Základy</b>	<b>6</b>
2.1	Instalace a spuštění . . . . .	6
2.2	Základy ovládání . . . . .	6
<b>3</b>	<b>Zvuk a FM syntéza</b>	<b>9</b>
3.1	Co je to zvuk? . . . . .	9
3.2	Podrobněji o sčítání signálů . . . . .	10
3.3	Základy FM syntézy . . . . .	11
3.4	Trocha historie FM syntézy . . . . .	12
<b>4</b>	<b>MIDI</b>	<b>13</b>
4.1	Co je MIDI? . . . . .	13
4.2	Zprávy MIDI . . . . .	14
4.3	Virtuální MIDI rozhraní . . . . .	15
<b>5</b>	<b>Popis výpočtu zvuku a jednotlivých parametrů</b>	<b>17</b>
5.1	Nastavení k přehrávání MIDI souborů . . . . .	17
5.2	Mapa nástroje . . . . .	18
5.3	Mapa operátoru . . . . .	19
5.4	Parametry operátoru (karty operátorů označené čísly) . . . . .	20
5.5	Parametry nízkofrekvenčního oscilátoru (karta LFO) . . . . .	22
5.6	Obecné parametry (karta General) . . . . .	22
5.7	Příklad nastavení zvuku (flétna) . . . . .	24
<b>6</b>	<b>Programová realizace</b>	<b>26</b>
6.1	Vlákno hlavního okna . . . . .	26
6.2	Výpočetní vlákno . . . . .	27
<b>7</b>	<b>Problémy při realizaci</b>	<b>29</b>
<b>8</b>	<b>Odkazy na další zdroje informací</b>	<b>31</b>
	<b>Literatura</b>	<b>32</b>

Název práce: Tvorba zvuku pomocí aditivní a FM syntézy

Autor: Václav Vlček

Katedra (ústav): Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Tomáš Holan, Ph.D.

e-mail vedoucího: tomas.holan@mff.cuni.cz

Abstrakt: Tato práce popisuje instalaci a ovládání vytvořeného programu, principy použité k výpočtu zvuku (aditivní a FM syntézu) a jednotlivá nastavení, kterými je možno vytvářený zvuk ovlivnit. Dále se věnuje té části standardu MIDI, která byla v programu použita, a z malé části i jeho historii. V závěru shrnuje problémy, které přinesla realizace programu.

Klíčová slova: FM syntéza, MIDI, Yamaha DX-7

Title: Generating of sound using additive and FM synthesis

Author: Václav Vlček

Department: Department of Software and Computer Science Education

Supervisor: RNDr. Tomáš Holan, Ph.D.

Supervisor's e-mail address: tomas.holan@mff.cuni.cz

Abstract: This paper describes the installation and use of created program as well as technics for sound generating (additive and FM synthesis) and options affecting the result sound. It contains that part of MIDI standard which is used in the program and a small part of its history, too. Finally, it summarizes problems which appeared during creating of the program.

Keywords: FM synthesis, MIDI, Yamaha DX-7

# Kapitola 1

## Úvod

Program je určen k tvorbě zvuku pomocí aditivní a FM syntézy. Tento text začíná popisem toho, jak můžete program zprovoznit na svém počítači a základů jeho ovládání. Po přečtení této kapitoly byste měli být schopni jej použít k vytváření zvuku (hraní). Následuje kapitola o zvuku a postupu použitém k jeho generování a kapitola o některých částech standardu MIDI. Věci zde popsané by měly přispět ke snažšímu pochopení významu jednotlivých parametrů ovlivňujících výsledný zvuk.

Další kapitola obsahuje detailní popis jednotlivých parametrů, je v ní obsažena i přehledná mapa nástroje a operátorů (pokud máte tu možnost, vřele doporučuji si ji vytisknout a mít při ruce při čtení textu, který za ní následuje, případně i při vytváření nového zvuku). V jejím závěru je uveden příklad vytvoření zvuku flétny.

Nakonci textu je popsáno, jak program funguje uvnitř, a postřehy učiněné při jeho realizaci.

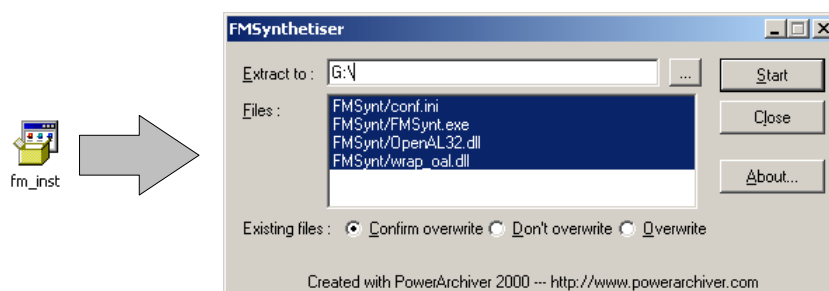
# Kapitola 2

## Základy

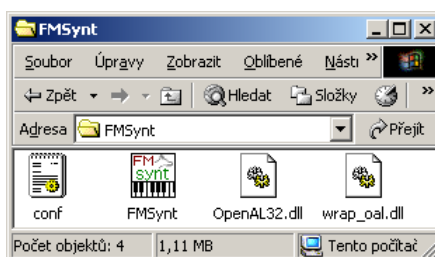
V této kapitole se dočtete, jak program zprovozníte na svém počítači, a také něco o jeho ovládání.

### 2.1 Instalace a spuštění

Instalační program je umístěn v souboru `fm_inst.exe`, po jeho spuštění máte možnost vybrat, kam se program uloží.

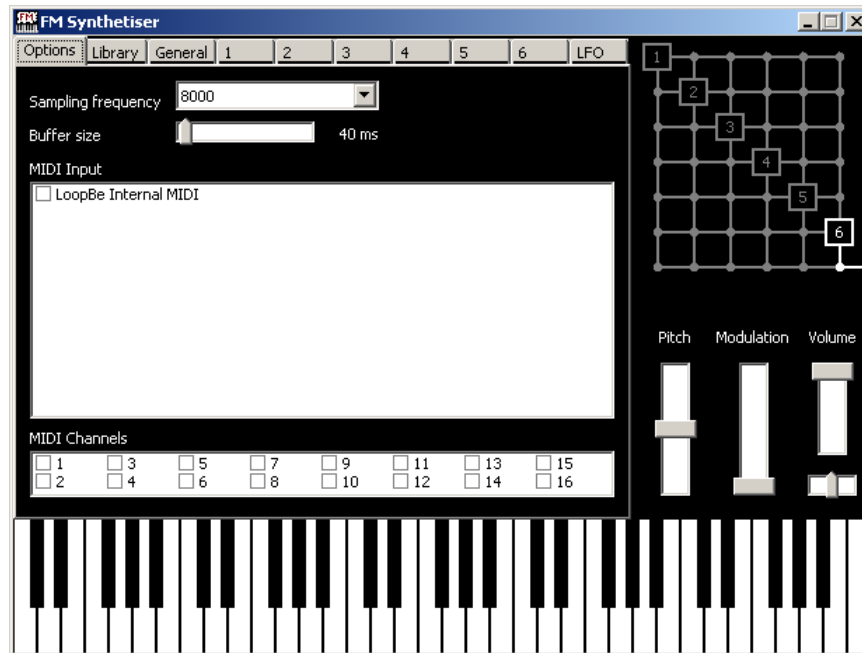


Program pak spustíte z pomocí `FMSynt.exe` z místa, kam byl instalován.



### 2.2 Základy ovládání

Po spuštění se před vámi otevře okno programu. Lze je rozdělit do několika částí. Největší část (vlevo nahoře) je část pro nastavení zvuku, která se skládá z několika karet.



Na kartě *Options* je možnost zvolit vzorkovací frekvenci (*Sampling rate*) použitou při generování zvuku. Vyšší vzorkovací frekvence přináší kvalitnější zvuk, ale nese s sebou vyšší zatížení procesoru. Volba *Buffer size* udává velikost bufferu použitého při generování zvuku. Větší buffer přináší větší zpoždění mezi akcí provedenou uživatelem a její realizací (např. dobu mezi stiskem klávesy a okamžikem, kdy začne znít), ale příliš malý buffer se také může nepříznivě projevit na kvalitě zvuku. Dále je zde možné zvolit vstupní MIDI zařízení (například připojené klávesy) a kanály, z nichž mají být přijímány zpávy, těmi je pak možno ovládat vybrané funkce programu.

Karta *Library* slouží k ukládání a k práci s uloženými zvuky. Do každého řádku lze zapsat jméno (nezávisle na tom, zda je na příslušné pozici uložený zvuk), které se uchová i pro příští spuštění. Tlačítkem *Save* se uloží aktuální nastavení parametrů zvuku (z dalších karet) na právě zvolenou pozici. *Load* naopak slouží k vyvolání nastavení zvuku z označené pozice.

Část pro nastavování zvuku ještě obsahuje karty s parametry ovlivňujícími generovaný zvuk, ale vysvětlení jejich významu zatím odložíme (až po vysvětlení principů, na kterých je generování zvuku vystavěno).

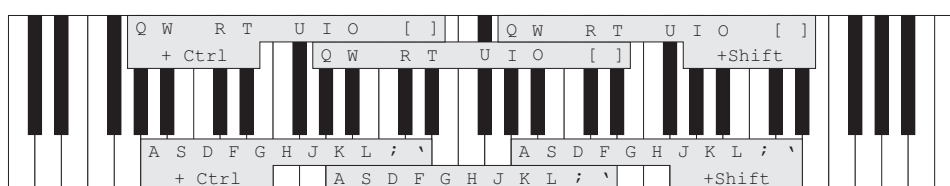
Vpravo nahoře je tzv. modulační matice, pomocí které lze určit, jak a z jakých složek se zvuk poskládá (vše bude jasné až po popisu tvorby zvuku). Kliknutím na operátor (znázorněn čtverečkem s číslem uprostřed) se otevře příslušná karta a operátor se orámuje žlutě. Aktivní operátory jsou bílé, neaktivní šedé. Puntíky znázorňují propojení příslušných operátorů. Také mění svou barvu podle stavu (aktivní nebo neaktivní), který lze měnit kliknutím.

Pod modulační maticí jsou tři posuvníky. Posuvník *Pitch* umožňuje měnit výšku tónu (rozsah a krok změny lze určit volbou na kartě *General*). Posuvníkem *Modulation* lze měnit výstupní zvuk opět v závislosti na dalším nastavení. Pomocí *Volume* se určuje hlasitost (snížení hlasitosti může zamezit deformaci zvuku vlivem ořezávání příliš vysokých amplitud), pod ním se nachází posuvník vyvážení mezi jednotlivé ka-

nály stereofonního signálu.

Pokud byste potřebovali přenést nastavení (všechny uložené zvuky včetně nastavené vzorkovací frekvence a velikosti buffru – tedy vše z oblasti karet pro nastavování zvuku), pak stačí zkopírovat soubor `conf.ini`, ve kterém jsou uloženy všechny uvedené vlastnosti.

Nepřehlédnutelnou součástí celého okna je klaviatura. Lze ji ovládat myší. Pokud chcete nechat některý tón hrát déle, můžete stisknout klávesu `Ctrl` při uvolňování tlačítka myši a tón bude znít stále. K ovládání klaviatury lze též použít klávesnici počítače. Rozložení kláves je uvedeno na následujícím obrázku. Klávesy jsou přiřazeny tak, aby sousední bílé klapky klaviatury odpovídaly sousedním klávesám klávesnice v anglickém i českém rozložení (a černé klapky byly umístěny na příslušných místech). Navíc je možno použít klávesu `Shift` k dočasnému posunutí kláves klávesnice o oktávu výše a klávesu `Ctrl` na posun o oktávu níže.





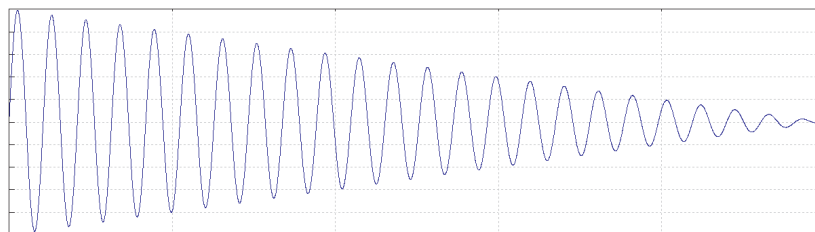
# Kapitola 3

## Zvuk a FM syntéza

### 3.1 Co je to zvuk?

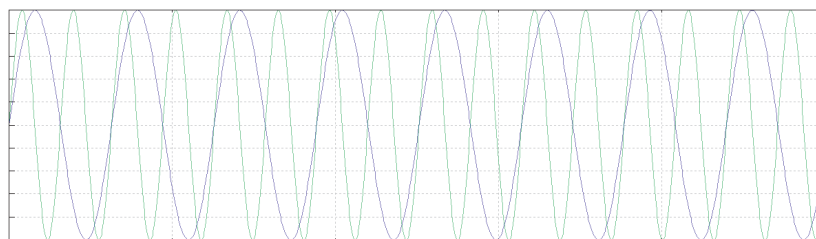
Lidské ucho je orgán schopný vnímat mechanické vlnění okolního prostředí jako zvuk (ve spolupráci s mozkiem, samozřejmě). Toto vlnění může vznikat rozličnými způsoby. Například tak, že kmitá nějaký pevný předmět (struna hudebního nástroje, ladička, membrána reproduktoru, ...), který rozkmitává okolní vzduch (stejně jako se rozvlní vodní hladina po vhození kamene); pomocí vlnění vzduchu se zvuk šíří až k uchu posluchače. Jinou možností je rozkmitání vzduchu třeba na hraně píšťalky, kde se vzduch dělí na dva proudy (jeden prochází tělem píšťalky a druhý odchází ven).

Pokud chceme zvuk popsat, budou nás zajímat dva hlavní pojmy. Prvním bude tzv. *amplituda*, která udává hlasitost zvuku (v analogii s vodní hladinou by odpovídala výšce vlny). Tento pojem lze chápat ve dvou významech, buď jako *okamžitou amplitudu* (výšku vodní hladiny v daném místě v konkrétní okamžik), nebo jako *maximální amplitudu* (maximální výšku, do které hladina v daném místě vystoupí). Pokud se mluví o amplitudě, myslí se většinou maximální amplituda.



*Zvuk s konstantní frekvencí a klesající amplitudou*

Druhým pojmem používaným pro charakterizaci zvuku je *frekvence*, ta vypovídá o výšce tónu (rozestup mezi jednotlivými vlnami). Frekvence se udává v hertzech (Hz), je to číslo určující počet kmitů za sekundu.



Dva tóny (zelený má dvojnásobnou frekvenci - je o oktávu vyšší)

V hudbě se používá pojem *výška tónu*, to je jen jiné vyjádření frekvence zvuku tvořeného hudebním nástrojem. Existuje mezinárodní dohoda, která určuje výšku *komorního A* na 440 Hz. Z této hodnoty lze dopočítat frekvence všech ostatních tónů podle následujícího vzorce

$$f = 440 \cdot 2^{\frac{\text{počet půltónů nad komorním A}}{12}}$$

to znamená, že tón o oktávu (dvanáct půltónů) vyšší má dvojnásobnou frekvenci.

Možná vás napadne otázka, jak potom poznáte zvuk houslí od zvuku klavíru, když je na oba nástroje zahrán stejný tón. On totiž ani klavír, ani housle nevydávají jen ten samotný tón, který je zahrán, ale zároveň i další složky s jinými frekvencemi, které se sčítají do výsledného zvuku. Tón, na který je nástroj naladěný, mívá největší amplitudu (je to trochu složitější otázka anatomie lidského ucha a psychoakustiky), ostatní jsou tišší a určují barvu zvuku. Vjem ze zahráného tónu přináší i plynulá změna amplitudy.

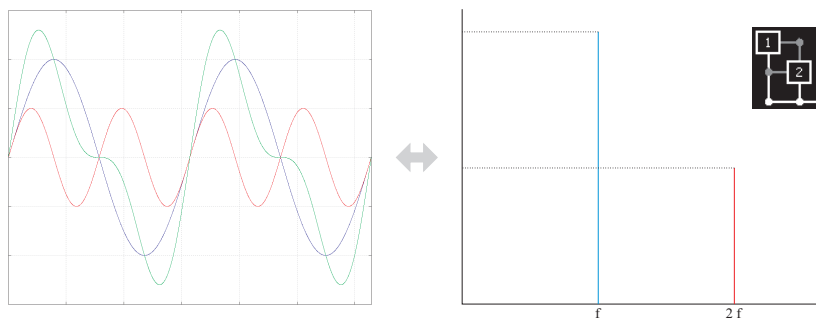
Tento program má schopnost poskládat zvuk dvěma způsoby. Buďto přímo sčítat jednotlivé tóny, nebo použít tzv. FM syntézu, o které ještě bude řeč. Lze v něm nastavit mnoho parametrů, ale ty vždy nějakým způsobem ovlivňují buď frekvenci, nebo amplitudu složky používané k tvorbě zvuku.

## 3.2 Podrobněji o sčítání signálů

Řekli jsme, že zvuk může vzniknout součtem několika tónů. Existuje i matematická teorie, umožňující zpětně rozložit součtový signál do těchto složek (do tzv. *spektra*; bližší informace můžete najít pod pojmy jako je Fourierova transformace, spektrální analýza a Fourierův integrál). Pro nás je důležité, že tyto popisy jsou (v případě spojitých jevů, jakým je i vlnění) ve vzájemně jednoznačném vztahu.

Můžeme tedy popsat zvuk pomocí jeho složek (výsledný zvuk získáme pomocí součtu jejich okamžitých amplitud v příslušném okamžiku). V našem programu jsou součty znázorněny puntíky na modulační matici a generátory harmonického signálu (tj. sinusovek) jako čtverečky.

Představme si následující příklad: zdroje 1 a 2 generují tón (zdroj 2 s dvojnásobnou frekvencí a poloviční amplitudou), na levém obrázku jsou vidět oba tóny i vzniklý *součet*, vpravo pak spektrum.

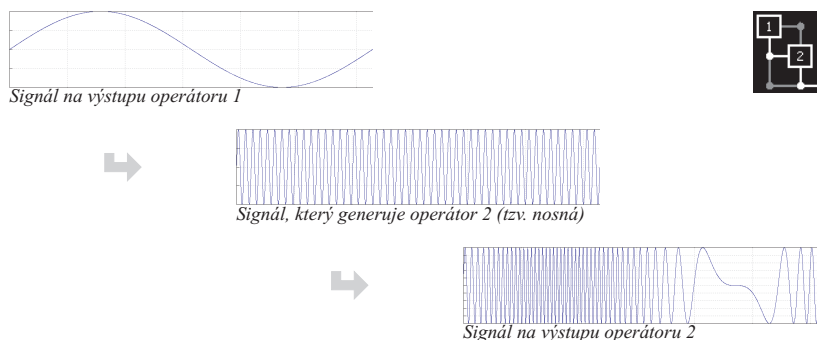


Porovnání zvuku zakresleného jako průběh okamžité amplitudy a ve složkách.

Tóny se ale mohou lišit i o jiný násobek než celý. Základní tón (to co chápeme pod pojmem výšky tónu) mívá zpravidla největší amplitudu a nejnižší frekvenci. Složitější zvuky se mohou skládat i z mnoha tónů (spektrum může být i spojitě – rozklad do spojitého spektra viz Fourierův integrál). Pokud bychom tento program používali jen k takovému sčítání, byly by možnosti omezené (bylo by možno skládat jen z 6ti složek).

### 3.3 Základy FM syntézy

Pro tvorbu složitějších zvuků je použit ještě jeden přístup, takzvaná FM modulace. Můžeme si ji představovat tak, že do oné krabičky (v terminologii obdobných hudebních nástrojů se označuje slovem *operátor*) vstupuje nějaký signál a ona sama podle jeho okamžité amplitudy zrychluje nebo zpomaluje signál, který generuje a posílá na výstup (mění jeho okamžitou frekvenci). Na vstup lze přivést signál získaný libovolným způsobem.



FM modulace

Pro matematictější zaměřené duše zde uvádím vztah, který lze použít k výpočtu FM modulace:

$$a = \sin \left( 2\pi f_{\text{nosná}} \cdot \left( x + \int_0^x \text{modulovaný signál}(t) dt \right) \right)$$

kde  $a$  označuje okamžitou amplitudu počítanou v čase  $x$ ,  $f_{\text{nosná}}$  je frekvence, na kterou se moduluje (v příkladu je to vnitřní frekvence operátoru 2). Integrál zajišťuje jakousi kumulaci změny amplitudy v čase.

Tak, jak je vzorec napsán, závisí *hloubka modulace* (tj. míra, jakou ovlivňuje jednotková amplituda modulovaného signálu změnu výstupní frekvence) na vnitřní frekvenci operátoru. V jiných oblastech by tomu tak nemuselo být, ale v případě tvorby zvuku je to přirozené, protože nechceme, aby hloubka modulace byla absolutní, ale závisela na výšce tónu (a výstupní signál závisel na frekvenci generovaného tónu jen "svou rychlostí" a neměnil tvar v závislosti na výšce).

### 3.4 Trocha historie FM syntézy

Frekvenční modulace byla vymyšlena již v polovině třicátých let a používala se (a ještě stále používá) k přenosu zvukového signálu v pásmu velmi krátkých vln (vysílání rádia). Zde se ovšem používají frekvence nosného signálu okolo 100 MHz.

Použití FM modulace jakožto postupu k tvorbě zvuku zkoumal v letech 1967 – 1968 John Chowning ze Stanfordské univerzity. V roce 1975 si ji nechal patentovat (tento patent vypršel v roce 1995) a později udělil licenci firmě Yamaha.

Tato firma uvedla v roce 1983 na trh svůj hudební nástroj Yamaha DX-7, který prováděl FM syntézu na digitální bázi. Yamaha DX-7 patřila k nejprodávanějším nástrojům (přes 170 000 kusů za 5 let), dokázala vytvářet čisté zvuky a přitom vážila jen 14,2 kg, měla klaviaturu s 61 klávesami, dynamikou úhozu a aftertouchem (zpracování změn přítlaku již stisknuté klávesy). K syntéze zvuku používala 6 operátorů (buď mohly sloužit jako oscilátor – generovat sinusový průběh, nebo k FM modulaci), které bylo možno propojit do 32 předem daných *algoritmů* (tak se v oblasti FM syntézy označuje konkrétní pospojování). O výstup se staral 10bitový D/A-převodník. Měla i rozhraní MIDI, ale vzhledem k tomu, že byla uvedena na trh dříve, než se ustálil MIDI standard, nebyla s ním plně kompatibilní (používá rozsahy 0 – 99 namísto 0 – 127 a při číslech mimo rozsah dochází k přebuzení). Tento nástroj se stal i vzorem pro program, o němž právě čtete (obsahuje stejně pojmenované parametry zvuku, ale autor si není jist tím, zda všechny odpovídají svými rozsahy – nebyl dostupný nástroj, jen uživatelská příručka, kde nejsou popsány výsledky odpovídající danému nastavení, ba dokonce z ní není možné jednoznačně určit pořadí, v jakém jsou operace aplikovány – bylo zvoleno nejlogičtější možné).

FM syntéza se používala i ve zvukových kartách (hlavně na počátku 90. let) jakožto hardwarový MIDI syntetizátor. Nejvíce se hodí k tvorbě zvuků kovů, skleněných předmětů a futuristicky znějících zvuků, ale lze ji použít i k napodobení jiných reálných nástrojů. Dnes se (hlavně v těchto oblastech) stále používá, ale k tvorbě zvuku reálných nástrojů se prosadila spíše syntéza pomocí samplů (v paměti uložených nahrávek nástrojů, které jsou dynamicky upravovány podle potřeby – výška tónu, ...).



Yamaha DX-7

# Kapitola 4

## MIDI

Většinu informací uvedených v této kapitole jsem čerpal z knihy [1], ve které můžete najít další podrobnosti.

### 4.1 Co je MIDI?

MIDI je standard popisující rozhraní, které slouží ke vzdálenému ovládní hudebních nástrojů, sekvencerů (zařízení, které posílá příkazy jiným zařízením), apod. Vznikl v době, kdy takovéto standardy existovaly jen v rámci každé z firem působících v dané oblasti. V roce 1982 se společnosti Roland Corp. a Sequential Circuits dohodly, že umožní propojení mezi svými nástroji. O rok později se k této iniciativě přidaly i další společnosti jako Yamaha a Oberheim, a tak vznikl standard MIDI. Výše zmiňovaná Yamaha DX-7 (1983) byla jedním z prvních plně digitálních syntetizátorů s vestavěným MIDI rozhraním.

V té době se rozvíjely také osobní počítače a společnost Roland se chopila příležitosti ovládní syntetizátorů z počítače a uvedla na trh zařízení MPU-401, které umožnilo připojit MIDI zařízení k počítači (byla to karta do rozhraní ISA a externí box s konektory).

Později vznikla organizace MIDI Manufacturer's Association, která se o standard stará. MIDI soubory, jakožto možnost uložení sekvence příkazů (neboli *zpráv*) vynikl až později, stejně jako standard General MIDI, který určoval mimo jiné přiřazení čísel zvukům hudebních nástrojů.

Data jsou po rozhraní MIDI přenášena sériově, jednosměrně. Na zařízení se mohou vyskytnout MIDI porty tří druhů: MIDI Out (slouží pro data, která jsou zařízením generována), MIDI In (rozhraní, na kterém zařízení „poslouchá“) a MIDI Thru (sem jsou kopírována data z MIDI In a slouží pro spojení několika zařízení do série).

Jedním rozhraním lze ovládat až 16 různých nástrojů (resp. i více, pokud mají některé hrát tentýž zvuk). To je způsobeno tím, že hlavička každé zprávy obsahuje 4 bity nesoucí číslo kanálu.

## 4.2 Zprávy MIDI

Pomocí zpráv se přenášejí informace typu: „stisknuta klávesa  $A_4$ , lehce“, „ovladač výšky tónu nastaven na 50 %“. Zprávy na většinu obvyklých akcí jsou obsaženy přímo ve standardu. Navíc existují zprávy SysEx (System Exclusive), jimiž je možno ovládat specifické vlastnosti hudebních nástrojů (např. přenášet vzorky zvuků do paměti hudebního nástroje, či měnit rozličná nestandardní nastavení).

Zprávy se po MIDI rozhraní přenáší po bytech. Byty, které mají nejvyšší bit 1 určují typ zprávy (označují se jako *Status byty*), zpravidla za nimi následuje jeden nebo dva byty, které nesou parametry (ty mají nejvyšší bit nulový - odtud rozsahy 0-127).

Status byte	1. byte dat	2. byte dat	Význam
1000cccc	0nnnnnnn	0vvvvvvv	Note Off (uvolnění klávesy) na kanálu <i>c</i> , tón číslo <i>n</i> , rychlostí <i>v</i>
1001cccc	0nnnnnnn	0vvvvvvv	Note On (stisknutí klávesy) na kanálu <i>c</i> , tón číslo <i>n</i> , rychlostí <i>v</i> (je-li 0, pak se jedná o uvolnění klávesy)
1010cccc	0nnnnnnn	0vvvvvvv	Polyphonic Key Pressure (Aftertouch; změna síly stisku již stisknuté klávesy)
1011cccc	0#####	0VVVVVVV	Control Change (změna hodnoty nějakého ovladače), # označuje číslo ovladače, V jeho novou hodnotu
1100cccc	0ppppppp	-	Změna vybraného zvuku (volba „hudebního nástroje“); zpráva je jen dvoubytová
1110cccc	01111111	0hhhhhhh	Pitch wheel change (změna hodnoty ovladače výšky tónu), 1 označuje méně významných sedm bitů, h nejvýznamnějších sedm bitů 14-ti bitového čísla (hodnota $2000_{16}$ označuje střed - tón beze změny)

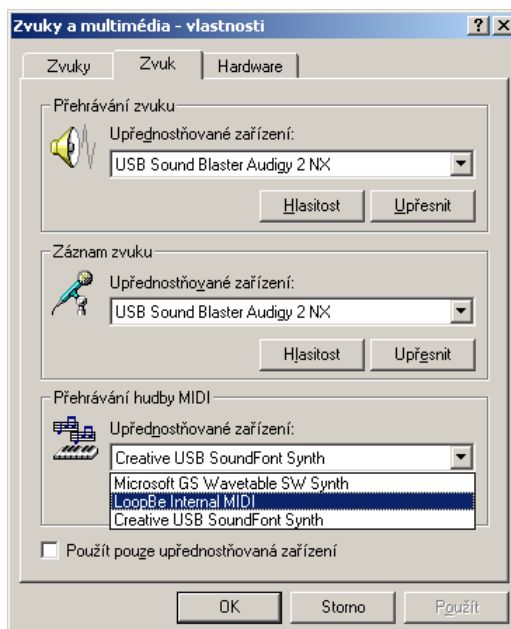
Zvláštní postavení má zpráva Control Change. Je to jedna zpráva pro několik různých ovladačů. Její 1. datový byte určuje, kterého ovladače se hodnota týká. Následující tabulka ukazuje jeho hodnoty pro vybrané ovladače (ty, které byly použity v popisovaném programu). Některé ovladače mají dvě zprávy. Jednu pro základní přesnost (rozsah 0 - 127), druhou pro zvýšení přesnosti (ta však nemusí být podporována všemi zařízeními).

1. datový byte	Přiřazený ovladač
01 <sub>16</sub>	Modulation wheel
07 <sub>16</sub>	Hlasitost
21 <sub>16</sub>	Modulation wheel (zvýšení přesnosti)
37 <sub>16</sub>	Hlasitost (zvýšení přesnosti)
40 <sub>16</sub>	Pedál sustain (hodnota 2. byte dat $\geq 64$ značí stisknutý pedál)
41 <sub>16</sub>	Pedál portamento (hodnota 2. byte dat $\geq 64$ značí stisknutý pedál)

### 4.3 Virtuální MIDI rozhraní

Pokud chceme v operačním systému umožnit používat program ostatním programům jako výstupní MIDI zařízení, musí být napsán jako ovladač (na úrovni jádra systému). Pokud to učiníme, bude jej možné vybrat jako standardní výstupní MIDI zařízení a používat z libovolného přehrávače hudby. Jestliže ale chceme, aby bylo možno program ovládat vstupním MIDI zařízením, musíme využít systémových služeb ke čtení zpráv z fronty, což je mnohem jednodušší.

Kdybychom požadovali oboje, musíme napsat dva programy. Nicméně existuje ještě jiná možnost - virtuální MIDI propojení. To je program, který se chová jako vstupní i výstupní MIDI zařízení (je tedy napsán na úrovni jádra systému). Chová se tak, že data, která mu nějaký program (např. přehrávač hudby) pošle jako výstupnímu MIDI zařízení ihned přepokopíruje na rozhraní, které se chová jako vstupní, odkud si je může vyzvednout jiný program (např. zde popisovaný) pomocí systémových služeb a zpracovat.



*Nastavení virtuálního MIDI rozhraní jako předvoleného*

Takovým programem je např. *LoopBe* (pro Win 2000/XP/Vista) nebo *Hubi's MIDI Loopback* (pro Win 95/98/ME). Odkazy na jejich domovské stránky jsou uvedeny v závěru. Pomocí jednoho z těchto dvou programů tedy můžete přehrávat hudbu z MIDI souborů pomocí našeho programu a libovolného přehrávače MIDI souborů (je nutné vybrat virtuální rozhraní MIDI v ovládacím panelu Zvuky a multimédia). Jako zdroj MIDI událostí je možné, kromě přehrávače, použít i virtuální MIDI klaviaturu (odkaz je také uveden v seznamu v závěru).

Pokud budete chtít, aby náš program zpracovával zprávy z více kanálů najednou, musíte počítat s tím, že nástroj může mít vybrán jen jeden zvuk (a vše tedy bude hrát jedním). Dále je také třeba počítat s tím, že MIDI soubory předpokládají standardní číslování zvuků (hudebních nástrojů), a tedy výsledek nemusí znít tak, jak byste předpokládali. Nicméně i když je program ovládán MIDI zprávami, máte stále možnost jej nastavovat z jeho rozhraní (a tedy i načíst jiný nástroj, než byl vybrán pomocí MIDI zprávy).



# Kapitola 5

## Popis výpočtu zvuku a jednotlivých parametrů

Všechny parametry, které jsou vyobrazeny na schématu mapy nástroje, jsou dostupné z karet *General* a *LFO*, parametry ze schématu operátoru jsou dostupné na kartách operátorů (s čísly).

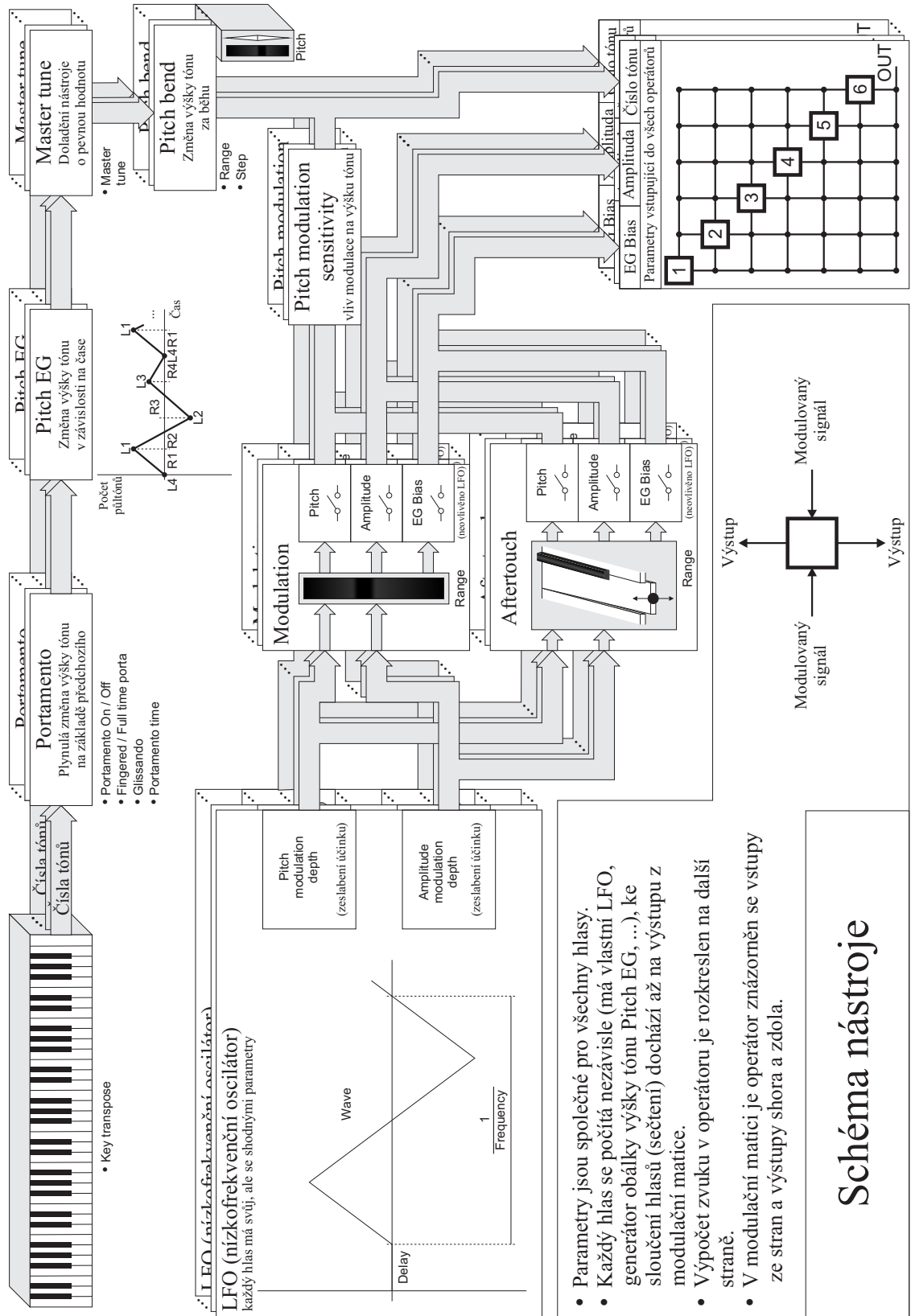
### 5.1 Nastavení k přehrávání MIDI souborů

V předchozí kapitole jsme se zmiňovali o virtuálních MIDI rozhraních. Zde se pokusíme o praktičtější popis, jak vše nastavit, aby bylo možno přehrávat MIDI soubory. Nejdříve si nainstalujte jeden ze zmiňovaných programů (podle vašeho operačního systému). Pokud váš program pro přehrávání hudby nemá možnost volby výstupního MIDI zařízení, nastavte v ovládacím panelu *Zvuky a multimedia*, na kartě *Zvuk* v kolonce *Upřednostňované MIDI zařízení* to, které odpovídá nainstalovanému programu.

V našem programu pak po spuštění zaškrtněte instalované virtuální MIDI zařízení jako MIDI vstup a vybrané kanály, jejichž příkazy mají být zpracovávány. Nyní bude náš program použit k reprodukci MIDI souboru přehrávaného ve vašem oblíbeném přehrávači. Současně s přehráváním souboru můžete měnit různá nastavení (např. vybraný hudební nástroj).

Pokud budete chtít přehrávat více kanálů různými nástroji, nic (kromě nároků na výpočetní výkon) nebrání tomu spustit program vícekrát a nechat každou instanci přehrávat zprávy z jiného kanálu.

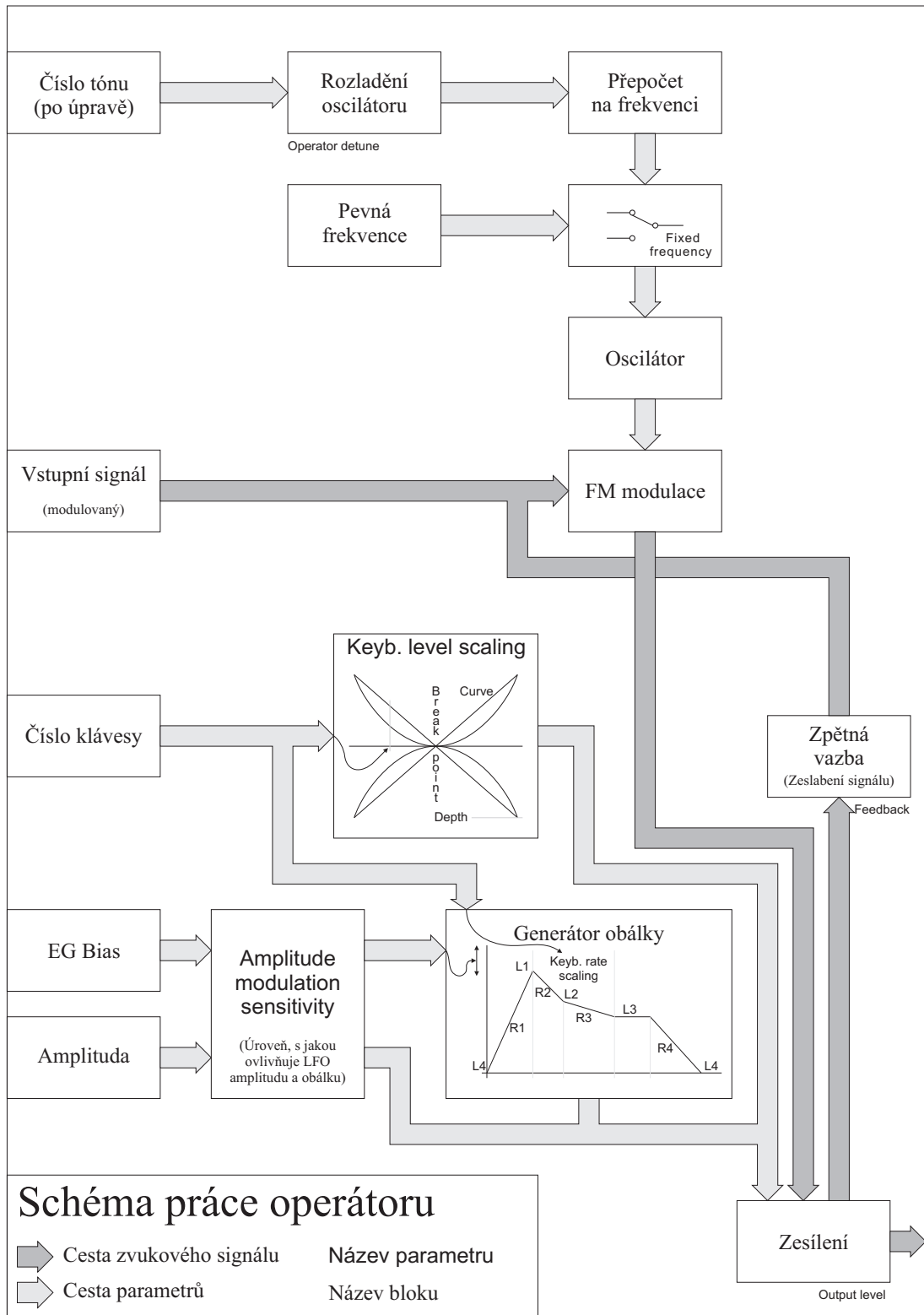
## 5.2 Mapa nástroje



- Parametry jsou společné pro všechny hlasy.
- Každý hlas se počítá nezávisle (má vlastní LFO, generátor obálky výšky tónu Pitch EG, ...), ke sloučení hlasů (sečtení) dochází až na výstupu z modulační matice.
- Výpočet zvuku v operátoru je rozkreslen na další straně.
- V modulační matici je operátor znázorněn se vstupy ze stran a výstupy shora a zdola.

**Schéma nástroje**

## 5.3 Mapa operátoru



## 5.4 Parametry operátoru (karty operátorů označené čísla)

Začneme u popisu funkce operátorů. Operátor může sloužit ke dvěma účelům. Nemá-li na vstup přiváděn žádný signál, slouží jako oscilátor (generuje harmonický průběh – sinusoidu), chová se vlastně tak, jako by moduloval nulový vstupní signál (tj. nemění frekvenci).

Druhé použití je jako FM modulátor. V tomto případě si vnitřně generuje sinusoidu a na ní moduluje vstupní signál.

### Enable

Rozhoduje o tom, zda je operátor v provozu. Není-li tento parametr zaškrtnut, má operátor na výstupu stále nulovou hodnotu.

### Frequency

Zde jsou dvě možnosti. Pokud je vybrána volba *Hz*, je vnitřní frekvence operátoru pevně zvolená, nezávislá na stisknuté klávese. Je-li vybrána volba *x*, provede se nejdříve přepočítání čísla tónu (upraveného parametry jako je portamento, Master tune, ... dle mapy nástroje tak, že obsahuje i desetinnou část) na frekvenci odpovídající tomuto tónu a číselná hodnota parametru *Frequency* se použije jako násobek frekvence.

Volba *x / Hz* je ve schématu operátoru označována jako parametr *Fixed Frequency*.

### Detune

Tento parametr se bere v potaz, pouze pokud je zvolen násobek frekvence tónu a slouží k jemnějšímu nastavení frekvence.

### Feedback

Určuje úroveň zpětné vazby – množství signálu z výstupu tohoto operátoru, které je přiváděno zpět na jeho vstup.

### Synchronize

Pokud je toto políčko zaškrtnuto, uvede se fáze operátoru vždy se stiskem klávesy do fáze 0 – sinusový průběh generovaný vnitřním oscilátorem tak není posunut.

### Output level

Jedná se o výstupní zesílení signálu. Pokud je operátor připojen přímo na výstup, určuje tento parametr hlasitost. Je-li výstup připojen na vstup jiného operátoru, rozhoduje o hloubce modulace – jakou měrou je ovlivňován signál operátoru, na jehož vstup je připojen.

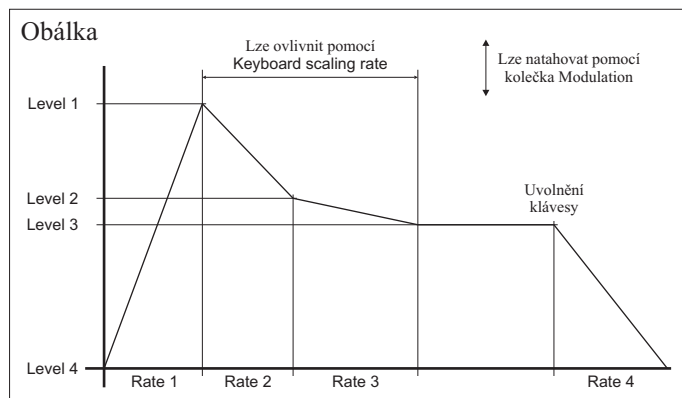
### Envelope generator

Generátor obálky umožňuje změnu výstupního zesílení závislou na čase od okamžiku stisku klapky na klaviatuře. V případě přímého připojení operátoru na vstup umožňuje měnit hlasitost (např. rychlý náběh po úderu, postupný pokles do oblasti výdrže a při puštění klávesy pozvolný sestup do úplného ticha). Stejně tak lze v čase měnit i úroveň modulace na další operátor (je-li výstup operátoru připojen na vstup

jiného). Význam jednotlivých parametrů přehledně znázorňuje obrázek. Parametr určující počáteční a koncovou hodnotu je shodný.

Úroveň zesílení (natahování obálky nahoru a dolů) lze ovlivnit také pomocí posuvníku Modulation (viz *General, Modulation, EG Bias*)

I když je možné nastavit koncovou úroveň na nenulovou hodnotu, zvuk se počítá pouze do doby, než uplyne nejdelší čas obálky ze všech operátorů připojených přímo na výstup (z důvodu nižšího zatížení procesoru).



#### Keyboard rate scaling

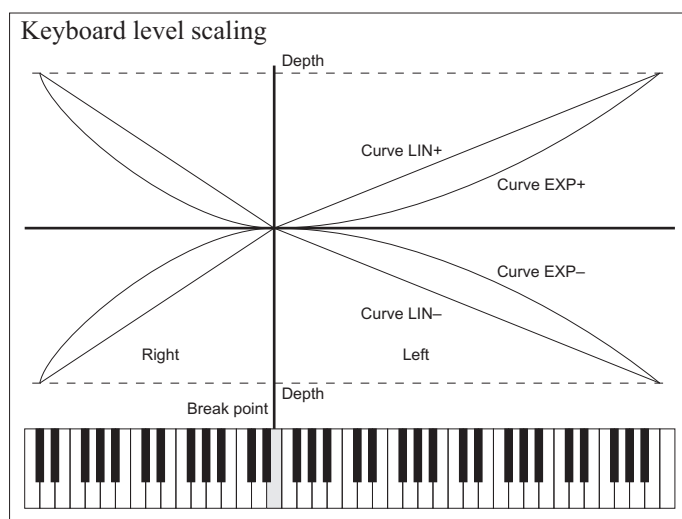
Tento parametr souvisí s generátorem obálky. Umožňuje prodloužit část 2 a 3 u hlubších tónů a zkrátit u vyšších.

#### Aplitude modulation sensitivity

Slouží k nastavení úrovně, jakou ovlivňuje nízkofrekvenční oscilátor (LFO) amplitudu (jak moc dochází ke ztišování a zesilování řízenému LFO).

#### Keyboard scaling

Umožňuje měnit výstupní úroveň v závislosti na výšce tónu, který byl zahrán. Význam dalších parametrů je popsán v obrázku.

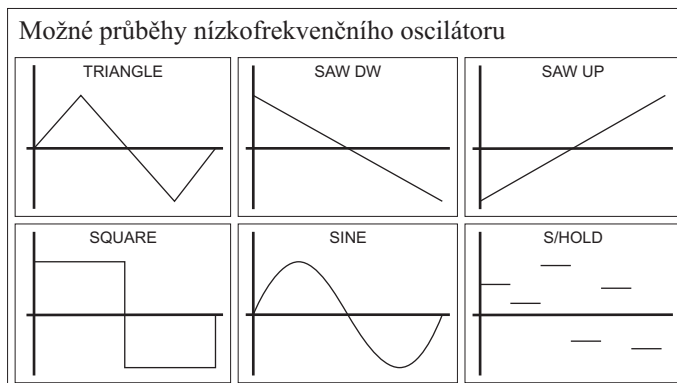


## 5.5 Parametry nízkofrekvenčního oscilátoru (karta LFO)

Nízkofrekvenční oscilátor slouží k přidávání vibrata (pravidelná změna výšky tónu v čase) nebo tremola (pravidelná změna hlasitosti znějícího tónu). Míru účinku lze ovlivnit pomocí posuvníku *Modulation* a funkce *Aftertouch* připojených MIDI kláves (a následujících parametrů).

### Wave

Průběh generovaný nízkofrekvenčním oscilátorem.



### Frequency

Frekvence nízkofrekvenčního oscilátoru.

### Delay

Čas od stisku klavky, po kterém začne nízkofrekvenční oscilátor pracovat (do té doby generuje nulový signál).

### Pitch modulation depth

Úroveň ovlivnění výšky tónu (společně s *Pitch modulation sensitivity*, ale narozdíl od něj je *Pitch Modulation Depth* ihned na výstupu nízkofrekvenčního oscilátoru).

### Amplitude modulation depth

Nastavuje míru ovlivnění amplitudy (dále je míra ovlivněna posuvníkem *Modulation* a parametrem *Amplitude modulation sensitivity*, které se nachází u každého operátoru).

### Synchronize

Rozhodnutí, zda začíná operátor generovat ve fázi 0, nebo v náhodné fázi.

## 5.6 Obecné parametry (karta General)

Zde je vše, co se nehodilo jinam. Hlavně nastavení vlastností *Aftertouch*, posuvníku *Modulation* a několik parametrů, které se týkají změny výšky tónu (ty vlast-

nosti, které nějakým způsobem přetvoří číslo tónu dříve, než se tato hodnota dostane do operátoru).

### Modulation

Umožňuje nastavit, které vlastnosti budou ovlivňovány posuvníkem *Modulation* (výšku tónu *Pitch*, výstupní amplitudu *Amplitude* operátoru a natahování obálky na výšku *EG Bias* – viz také *Amplitude modulation depth*). A rozsah *Range* společný všem vlastnostem.

### Polyphonic

Rozhoduje o tom, zda bude nástroj schopen hrát více tónů najednou (až 16 hlasů), nebo pouze jeden. Počet hlasů také ovlivňuje režimy portamenta.

### Key transpose

Umožňuje „posunout“ zobrazovanou klávesnici o oktávu (nebo dvě) nahoru nebo dolů.

### Master tune

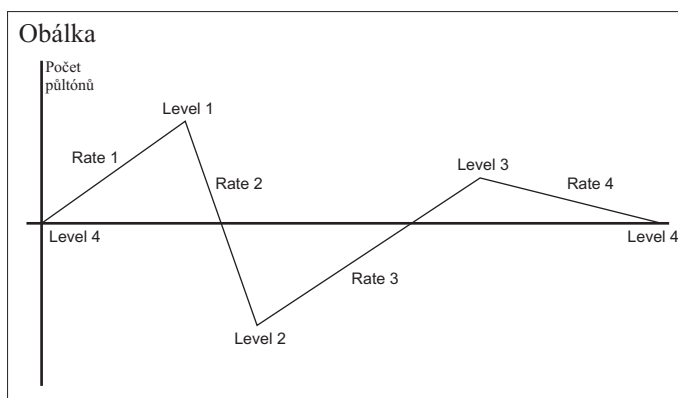
Tato hodnota umožňuje doladit výšku tónu s jinými hudebními nástroji. Umožňuje ladění o 75 centů (setin půltónu) výše i níže.

### Pitch modulation sensitivity

Slouží k nastavení míry, do jaké lze ovlivnit výšku tónu pomocí posuvníku *Modulation*.

### Pitch EG

Umožňuje periodicky měnit výšku tónu v závislosti na čase. Parametry jsou shodné s parametry obálky.



### Portamento

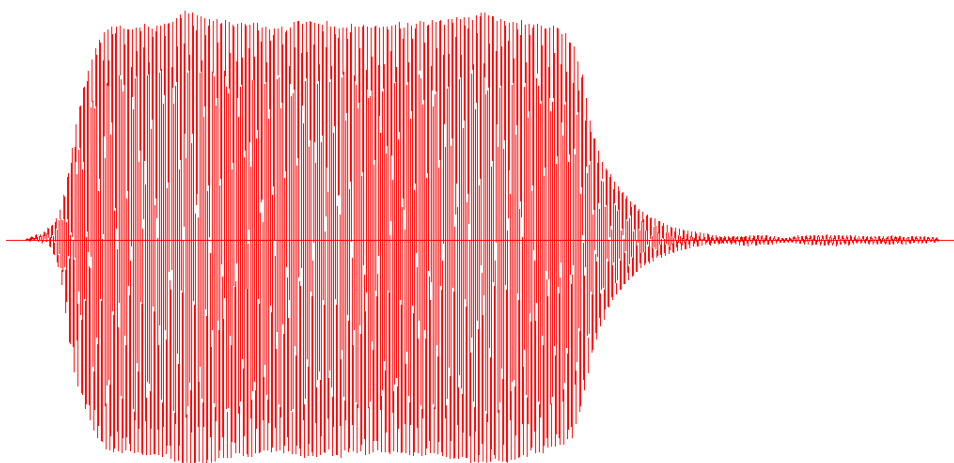
Portamento umožňuje samovolnou změnu výšky tónu. Je-li zapnuta volba *Glissando*, pak je změna naprosto plynulá, pokud aktivní není, pak změna probíhá po půltónech. Parametrem *Time* lze určit čas, který změna zabere. Jeho funkce je rozdílná pro vícehlasý a jednohlasý režim.

Ve vícehlasém režimu (viz parametr Polyphonic) se aplikuje, pouze pokud má začít znít sedmnáctý hlas, v tom případě je umlčen první hlas, který začal znít a dojde k plynulé změně výšky na výšku nového tónu.

V monofonním režimu je na výběr mezi dvěma druhy. *Fingered porta* způsobí to, že se efekt aplikuje, pouze pokud předchází tón ještě nedozněl, *Full time porta* způsobí to, že se projeví vždy (i když mezi tóny byla delší časová mezera).

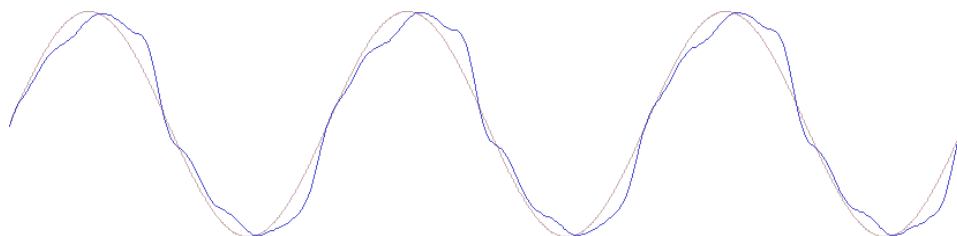
## 5.7 Příklad nastavení zvuku (flétna)

V závěru této kapitoly zkusíme nastavit zvuk, který by připomínal flétnu. Na následujících dvou obrázcích je vidět záznam zvuku reálného nástroje (altová zobcová flétna Yamaha YRA-312B III).



*Celý zahrnaný tón (pro určení tvaru obálky; znázorňuje 332 ms dlouhý úryvek)*

První obrázek nám ukazuje tvar obálky (to bude amplituda operátoru přivedeného na výstup). Lze ji rozdělit do tří částí: nástupu (ten je v programu modelován úseky *Rate 1* až *Rate 3*), výdrže („rovná“ oblast, kdy do flétny foukáme; chceme, aby odpovídala okamžiku, kdy je klávesa stisknuta) a sestupu (*Rate 4*). V našem programu lze obálku vytvarovat jenom z předem daného počtu úseček. Již teď je vidět, že nastane problém se sestupem, protože na ten máme jen jednu úsečku.



*Zvětšený úryvek průběhu reálného signálu s proloženou sinusoidou*

Druhý obrázek ukazuje průběh signálu. Při troše fantazie na něm lze napočítat deset „zlomů“ v každé periodě. K vytvoření podobného průběhu použijeme součet



dvou signálů. První bude vyjadřovat základní frekvenci (nechť se o něj stará operátor 6), druhý zajistí ony zlomy, bude mít pětinasobnou frekvenci a mnohem nižší amplitudu (o to se bude starat operátor 4). Operátor 5 použijeme k mírné úpravě základního signálu FM modulací.

Začneme tedy s nastavováním. Pro získání prázdného nastavení načteme tlačítkem *Load* (na kartě *Library*) zvuk z prázdné pozice. Klepnutím na příslušný uzel v modulační matici připojíme operátory 4 a 6 na výstup. Oba operátory zapneme zaškrtnutím *Enable* na jejich kartách. Zapneme také synchronizaci se stiskem klávesy volbou *Synchronize*. Operátoru 6 nastavíme frekvenci na 1 x (chceme, aby odpovídala stisknuté klávese) a operátoru 4 na 5 x. Operátoru s nosnou frekvencí (tj. 6) nastavíme výstupní zesílení *Output level* na plných 100, operátoru 4 jen na 10. Obálku nastavíme podle následující tabulky (hodnoty získané z nahrávek nástroje) u obou operátorů shodně.

Segment	Rate	Level
1	8	0,06
2	85	0,24
3	95	0,08
4	0	0,5

Nyní už by se měl zvuk podobat dechovému nástroji. Výsledný zvuk ještě trochu přikrášlíme použitím FM modulace. Připojíme výstup operátoru 5 na vstup operátoru 6. Zapneme jej a také zapneme jeho synchronizaci se stiskem klávesy. Jeho frekvenci nastavíme na 2 x, to by při vhodně velké amplitudě mělo způsobit přiblížení se obdélníkovému signálu (jak se píše v tipech k programování DX7 - viz odkazy v závěru) a trochu plnější zvuk. Výstupní zesílení tedy nastavíme na 7,5 a všechny položky *Level* na 100, časy můžeme nechat na 0.

Ještě jedno důležité upozornění na závěr. Pokud programujete zvuk, ve kterém mohou na výstup přicházet složky o vysoké frekvenci a nezanedbatelné amplitudě (FM s vyšší hloubkou modulace je v tomto směru celkem zrádná), měli byste mít na paměti Shannonův vzorkovací teorém - k přenosu (a reprodukci) signálu je třeba použít minimálně dvojnásobnou vzorkovací frekvenci, jinak může docházet k výraznému zkreslení („naprosto jiný“ signál). Proto je důležité uvážlivě volit nastavení *Sampling frequency* na kartě *Options*.

# Kapitola 6

## Programová realizace

Program je dvouvláknový (nepočítáme-li knihovnu, která se stará o přehrávání zvuku). Hlavní vlákno se stará o vykreslování okna aplikace a interakci s uživatelem. Vždy, když uživatel změní hodnotu nějakého parametru, oznámí to zprávou druhému vláknu (které se stará o vlastní výpočet zvuku). Kromě toho ještě pravidelně přistupuje k výpočetnímu vláknu (i když nepředává žádný parametr), aby bylo jisté, kam až může být zvuk dopočítán (označí okamžik, před kterým nenastane změna, která by ovlivnila generovaný zvuk).

Výpočetní vlákno poskytuje hlavnímu vláknu dvě procedury, které umožňují výše zmíněné (pravidelný přístup a provádění změn). Po svém spuštění inicializuje některé pomocné hodnoty a knihovnu umožňující přehrávání zvuku (OpenAL). Pak již v cyklu počítá (vždy nejdříve zkontroluje, zda se v daném okamžiku nemá aplikovat nějaká změna z fronty a poté spočítá hodnotu jednoho vzorku – okamžitou amplitudu výstupního signálu v čase) a přehrává zvuk (předá knihovně k přehrání celý připravený úsek – buffer, o jehož přehrání se knihovna postará sama).

### 6.1 Vlákno hlavního okna

Po svém spuštění vytvoří výpočetní vlákno a přiřadí mu vysokou prioritu (aby docházelo co nejméně k přerušování toku vypočteného zvuku). Poté načte jména uložených tónů z konfiguračního souboru (`conf.ini` – je organizován jako klasický ini soubor – v Delphi je použit objekt *TMemIniFile*) a tón uložený na první pozici. Hodnoty tohoto tónu také předá výpočetnímu vláknu.

Při každé změně nějakého parametru pošle příslušnou změnu výpočetnímu vláknu. Změny shodných parametrů z karet operátoru mají společnou obsluhu a to, o který operátor se jedná, je rozlišováno vlastností *Tag* příslušného objektu.

Za zmínku stojí ještě způsob odchyťávání kláves, které slouží ke stisku kláves z klaviatury. Hlavnímu formuláři je nastavena vlastnost *KeyPreview*, dostává tedy kódy stisknutých kláves dříve, než jsou odeslány objektu, kterému jsou určeny. Zde jsou zpracovány jako klapka klaviatury (ovšem jen za podmínky, že není prováděn zápis do seznamu názvů tónů). Aby nebylo písmeno vpisováno do Editů sloužících k nastavení tónu, je vždy při stisku aktivní Edit zamčen pro zápis a při uvolnění

klávesy opět odemknut. Jistě, bylo by možné odchyťávat klávesy již na úrovni smyčky zpráv, ale zvolený způsob umožňuje případný snazší převod na jinou platformu.

Při ukončení programu hlavní vlákno nejdříve zapíše konfiguraci do konfiguračního souboru a uzavře ho. Nakonec zastaví vlákno výpočetní a zruší je.

## 6.2 Výpočetní vlákno

Poskytuje hlavnímu vláknu proceduru *Access*, to je ta, pomocí které hlavní okno označuje okamžik, před kterým již nepřijde žádná zpráva o změně.

K předávání zpráv od hlavního vlákna slouží procedura *AddChange*, která má za parametr typ *TChange*. Tento parametr tvoří záznam, který v sobě obsahuje položku označující typ změny (položka výčtového typu), operátor, jehož se změna týká, a novou hodnotu parametru. Změny jsou přidávány do jednosměrného seznamu (který představuje frontu zpráv) včetně čísla vzorku, při kterém má být změna provedena (počítá se z času uloženého při spuštění vlákna, času, kdy je změna přidávána, a ze vzorkovací frekvence). Stav parametrů odpovídající stavu v čase počítaného vzorku je uložen ve struktuře *StateAndSettings*. Díky tomu, že je ukládán čas, kdy má být změna provedena (zapsána do *StateAndSettings*), se změny projevují se stejnými časovými rozestupy, s jakými byly uživatelem vykonány (i když se díky bufferování výstupního zvuku projeví se zpožděním). Změny fronty jsou uzavřeny do kritické sekce, aby nedošlo k poškození struktury při současném zápisu i čtení.

Při vytvoření výpočetního vlákna je inicializována knihovna OpenAL pro výstup, fronta a některé hodnoty výchozí hodnoty operátoru (jako je fáze, ve které se nachází vnitřní oscilátor).

V rámci běhu vlákna se vytvoří objekt pozorovatele a zdroje knihovny OpenAL (která sama o sobě umožňuje práci i s prostorovým zvukem). Poté se již ve smyčce počítají hodnoty vzorků odpovídající danému číslu vzorku (to je uchováváno v proměnné *ActualSampleNumber* viditelné z celého kódu výpočetního vlákna) – vždy se nejdříve aplikují změny z fronty (s číslem menším nebo rovným číslu počítaného vzorku) a poté se počítá aktuální hodnota vzorku (pomocí funkce *ComputeSample* – ta dělá přesně to, co je zobrazeno na obrázku mapy nástroje). Pokud se stane, že vlákno dopočítá všechny vzorky až k času posledního přístupu (tj. nejsme si jisti, že nepřijde žádná změna od uživatele), výpočetní vlákno si na chvíli (1 ms) odpočine.

V *ComputeSample* se počítá s tónem podle číslování MIDI, ale rozšířeným do reálných čísel (to umožňuje počítat i s hodnotami mezi dvěma tóny – hodí se to pro výpočet glissanda, obálky výšky tónu, ...). Spočítá se hodnota výšky tónu, která se pak předává operátorům (funkci *ComputeOutputOfOperator* – funkce, které je možno dotázat se na aktuální hodnotu výstupu operátoru). Zeptá se na výstupní hodnoty operátorů připojených na výstup nástroje (a ty se pak ptají na hodnoty výstupů operátorů připojených na jeho vstup – počítá se tedy jen to, co je opravdu potřeba). Pro urychlení výpočtu (a zamezení jeho zacyklení) se ukládá aktuální i předešlá hodnota výstupu operátoru (počítá se jen jednou).

Spočítají se hodnoty všech znějících hlasů a ty se sečtou (složí do jednoho signálu).

Obě funkce (*ComputeSample* i *ComputeOutputOfOperator*) jsou jen přepisem příslušných obrázků do programovacího jazyka.

Hodnotu nízkofrekvenčního oscilátoru počítá funkce *ComputeLFOLevel* (hodnota je společná pro všechny operátory, ale rozdílná pro každý hlas).

# Kapitola 7

## Problémy při realizaci

První drobný problém vyvstal při specifikaci vytvářeného programu. Dokumentace dostupná k nástroji Yamaha DX-7 je vcelku stručná a význam některých parametrů zvuku nebylo možno vyčíst jednoznačně, zvolil jsem vždy ten, který se mi jevil jako nejpravděpodobnější (často jsem také porovnával s chováním podobných nastavení v programu Native Instruments FM7, který je softwarovou napodobeninou zmiňovaného nástroje, ale mnohé parametry se v něm nastavují naprosto jinak).

Jeden problém vyvstal také s hudební terminologií (sám jsem se totiž nikdy neučil hrát na žádný hudební nástroj). Šlo o správnou interpretaci pojmů portamento a glissando. Z vysvětlení v různých (převážně anglických) encyklopediích vyplývaly dva odlišné a protichůdné významy. Nebylo z nich totiž jasné, který pojem znamená naprosto plynulou změnu, a který změnu po pultónech. Vše se nakonec vyřešilo s pomocí slovníčku [2]. Z něj vyplynulo, že termín portamento má dva různé významy jeden obecnější (obecně sklouznutí z jednoho tónu na druhý) a pak ten označující způsob hry (hraní lehce s co nejkratšími pauzami), glissando pak znamená plynulé sklouznutí.

Pokud jde o programování, tak jsem na žádný závažnější problém nenarazil (vše jsem vyřešil do druhého dne). Jen se vše velmi špatně ladilo, protože se jedná o vícevláknovou aplikaci. Naštěstí to téměř nebylo potřeba. Celé počítání zvuku jsem vpodstatě odladil v hlavě, a k mému překvapení v něm bylo jen pár nepatrných chybek. S laděním mi také významně pomohla moje zvuková karta, která umožňuje nahrávat to, co je na jejím výstupu.

Přesto jsem narazil na jeden problém. Tím byla výpočetní náročnost. Téměř vše se totiž počítá v plovoucí řádové čárce. Po několika optimalizacích jsem se dostal do stavu, kdy už bylo možné program rozumně používat (na mém počítači - Intel Pentium III 1133 MHz) a přesto neměly vliv na kvalitu výsledného zvuku. Z pohledu výpočetní náročnosti jsem zkoumal i program NI FM7, ale u něj jsem zjistil, že když delší dobu generuje jeden tón pomocí FM syntézy, tak se „oscilátor rozladí“ a barva tónu se změní. Zůstal jsem tedy u výpočtů s desetinnými čísly.

Největší problém se však ukázal až v samotném závěru. Bylo to používání samotného programu k vytváření nových zvuků. Zde jsem se marně snažil vytvořit nějaký předem požadovaný zvuk. Dokonce jsem hloubku FM modulace nastavil tak, aby přibližně odpovídala programu FM7, ale ani to nepomohlo. Výsledný vjem ze zvuku

je totiž ovlivňován mnoha parametry (např. časy náběžné a sestupné hrany tónu). Zkoušel jsem i návod, na nějž je odkaz v závěru, i nastavovat výstupní průběh tak, aby se podobal požadovanému nástroji, ale nebylo to nic platné. Používání jen ke sčítání není příliš složité, problémy nastávají, když chceme použít k FM syntéze signál, který již není harmonický (sinusovka), ale složený např. FM modulací nebo i součtem. Velmi špatně si lze představit, co bude výsledkem. Proto je také u programu jen pár mých neúspěšných pokusů.

# Kapitola 8

## Odkazy na další zdroje informací

Pokud Vás zajímá více o FM syntéze, nástroji Yamaha DX-7, nebo návody na jeho programování, pak Vám doporučuji následující literaturu. Osobně jsem čerpal z manuálu k nástroji, poznámek o jeho vlastnostech a znalostí ze střední školy (vzorec pro FM modulaci jsem snadno odvodil ze znalosti jejího principu).

1. <http://en.wikipedia.org/wiki/Synthesizer> – o syntéze zvuku obecně (anglicky)
2. [http://en.wikipedia.org/wiki/Frequency\\_modulation\\_synthesis](http://en.wikipedia.org/wiki/Frequency_modulation_synthesis) – o FM syntéze (anglicky)
3. [http://hem.passagen.se/tkolb/art/synth/dx7\\_e.htm](http://hem.passagen.se/tkolb/art/synth/dx7_e.htm) – o nástroji DX-7 (anglicky)
4. <http://www.maths.abdn.ac.uk/~bensondj/html/dx7.html> – odkazy na manuály k DX-7
5. <http://www.maths.abdn.ac.uk/~bensondj/dx7/manuals/prgrmdx7.pdf> – několik tipů, jak programovat tento hudební nástroj
6. <http://openal.com/> – Knihovna OpenAL použitá pro výstup zvuku
7. <http://www.noeska.com/doal/> – OpenAL pro Delphi
8. <http://www.midi.org/> – Stránky MIDI Manufacturers Association
9. [http://www.midimountain.com/delphi\\_midi.html](http://www.midimountain.com/delphi_midi.html) – Knihovna použitá pro práci s MIDI
10. <http://members.magnet.at/hubwin/midi.html> – Virtuální propojení pro Win95, Win98, WinME
11. <http://www.nerds.de/en/loopbe1.html> – a pro Win2000, WinXP, Vista
12. <http://www.granucon.com/vmk.html> – Virtuální MIDI klávesy

# Literatura

- [1] Guérin R.: *Velká kniha MIDI: standardy, hardware, software*, Computer Press, Brno, 2004.
- [2] Válek J.: *Italské hudební názvosloví*, Panton, Praha, 1991.