

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## SOFTWAREVÝ SAMPLER SE ZVUKY CIMBÁLU

THE SOFTWARE SAMPLER WITH SOUNDS OF OF THE DULCIMER

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Jan Bičan

### VEDOUCÍ PRÁCE

SUPERVISOR

MgA. Mgr. Ondřej Jirásek, Ph.D.

BRNO 2017



# Bakalářská práce

bakalářský studijní obor **Audio inženýrství**  
Ústav telekomunikací

**Student:** Jan Bičan

**ID:** 145970

**Ročník:** 3

**Akademický rok:** 2016/17

## NÁZEV TÉMATU:

### Softwarový sampler se zvuky cimbálu

#### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte jednoduchý softwarový sampler obsahující tóny celého rozsahu nástroje různé dynamiky a druhů excitace – obalené a neobalené paličky, nahrajte vzorky zvuků cimbálu a vytvořte rejstříkové sady cimbálových zvuků.

#### DOPORUČENÁ LITERATURA:

[1] GEIST, B. Akustika - jevy a souvislosti v hudební teorii a praxi. Praha: MUZIKUS s.r.o., 2005. ISBN 978-8086253312.

[2] SYROVÝ, V. Hudební akustika. Praha: AMU, 2003. ISBN 978-80-7331-127-8.

[3] RUSS, M. Sound Synthesis and Sampling. Focal Press, 1996. ISBN 0-240-51429-7

**Termín zadání:** 1.2.2017

**Termín odevzdání:** 8.6.2017

**Vedoucí práce:** MgA. Mgr. Ondřej Jirásek, Ph.D.

**Konzultant:**

**doc. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

#### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

# ABSTRAKT

Hlavní cíl této práce je vytvoření VST zásuvného modulu (pluginu), který bude sloužit jako výstupní část elektronického cimbálu. Práce nejprve vysvětluje základní pojmy sampler, cimbál, VST, MIDI, mikrofony a mikrofonní techniky snímání. Poté popisuje jednotlivé části tvorby softwarového sampleru se zvuky cimbálu. To zahrnuje nahrání jednotlivých tónů (vzorků) cimbálu, jejich třídění, editaci a implementaci do technologie VST. Popisuje problematiku konkrétního nástroje a odlišnosti od jiných, příbuzných zásuvných modulů a navrhuje řešení. Na konci práce je zhodnocen její průběh, testování s hardwarovými částmi elektronického cimbálu a její nedokonalosti s návrhy na budoucí zlepšení.

# KLÍČOVÁ SLOVA

Sampler, cimbál, VST, VSTi, MIDI

# ABSTRACT

Major aim of this thesis is to create VST plugin module, which would be an output part of electronic cimbalom (dulcimer). Thesis explains basic terms at first, such as sampler, cimbalom, VST, MIDI, microphones and microphone techniques for recording. Then it describes particular segments of creating a software sampler with cimbalom samples. That includes recording each of individual cimbalom tones, sorting and editing these samples and implementing to VST technology. Thesis also describes instrument's difficulties and differences from other related plugin modules and offers solution. There is a thesis's assessment at the end of the document, where is mentioned its progress, testing with hardware components of electronic cimbalom and its imperfections with suggestions of future improvement.

# KEYWORDS

Sampler, dulcimer (cimbalom), VST, VSTi, MIDI

BIČAN, J. *Softwarový sampler se zvuky cimbálu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 50 s. Vedoucí bakalářské práce MgA. Mgr. Ondřej Jirásek, Ph.D..

# PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Softwarový sampler se zvuky cimbálu jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto semestrálního projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

# PODĚKOVÁNÍ

Děkuji vedoucímu práce panu MgA. Mgr. Ondřeji Jiráskovi, Ph.D. za odborné vedení, konzultace a podnětné návrhy k práci. Dále děkuji panu doc. Ing. Jiřímu Schimmelovi, Ph.D. za konzultace a pomoc při realizaci nahrávání v bezodrazové komoře.

V Brně dne .....

.....

(podpis autora)



Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Purkynova 118, CZ-61200 Brno  
Czech Republic

<http://www.six.feec.vutbr.cz>

## PODĚKOVÁNÍ

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....

(podpis autora)



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# OBSAH

Úvod	11
<b>1 ZÁKLADNÍ POJMY</b>	<b>12</b>
1.1 Sampler	12
1.1.1 Historie samplerů	12
1.1.2 Princip funkce	13
1.1.3 Analogově – digitální převod	13
1.2 MIDI – obecné informace	14
1.2.1 Historie, význam a popis	15
1.2.2 Přenos, hardware a konektory	15
1.3 MIDI zprávy	16
1.3.1 MIDI zprávy NOTE On a NOTE Off	17
1.3.2 MIDI zpráva Control Change	19
1.3.3 MIDI zpráva Program Change	21
1.3.4 General MIDI	21
1.4 VST, ASIO	22
1.4.1 VST	22
1.4.2 ASIO	22
1.5 Cimbál	22
1.5.1 Historie	23
1.5.2 Konstrukce nástroje	23
1.5.3 Rozsah a ladění strun	24
<b>2 TVORBA VZORKŮ</b>	<b>26</b>
2.1 Typy mikronů	26
2.1.1 Kapacitní mikrofony	26
2.1.2 Páskové mikrofony	26
2.1.3 Směrové charakteristiky	27
2.2 Stereofonní snímání	28
2.3 Způsob snímání	29
2.4 Nahrávání, třídění a editace vzorků	32
2.4.1 Mix, střih a úprava vzorků	33

<b>3</b>	<b>IMPLEMENTACE VST</b>	<b>35</b>
3.1	JUCE.....	35
3.2	Problematika, odlišnosti.....	35
3.3	Řešení kódu.....	36
3.3.1	Základní prvky .....	36
3.3.2	Načtení vzorku .....	36
3.3.3	Spouštění vzorků.....	37
3.3.4	Načtení jednotlivých typů paliček – zvukových bank .....	37
3.3.5	Vývojový diagram .....	39
3.3.6	Uživatelské rozhraní .....	40
3.3.7	Verze „HW“ .....	42
<b>4</b>	<b>SHRUTÍ VÝVOJE PRÁCE</b>	<b>43</b>
4.1	Komplikace při nahrávání.....	43
4.2	Testování.....	43
4.2.1	Samostatné testování pluginu v host aplikacích .....	43
4.2.2	Testování s hardwarovou částí.....	44
<b>5</b>	<b>ZÁVĚR</b>	<b>45</b>
	<b>Literatura</b>	<b>46</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>47</b>
	<b>Seznam příloh</b>	<b>48</b>
<b>A</b>	<b>Obsah přiloženého disku DVD</b>	<b>49</b>
<b>B</b>	<b>Návod k obsluze</b>	<b>50</b>



## SEZNAM OBRÁZKŮ

Obr. 1.3.1: Struktura stavového a datového bytu [7].....	16
Obr. 1.3.2: Struktura zprávy Note On [7] .....	18
Obr. 1.3.3: Struktura zprávy Note Off [7] .....	18
Obr. 1.3.4: C-dur akord v MIDI kódu.....	19
Obr. 1.3.5: Struktura zprávy Control Change [7] .....	20
Obr. 1.3.6: Struktura zprávy Program Change [7].....	21
Obr. 1.5.1: Cimbál značky ARTEX Bohák .....	24
Obr. 1.5.2: Rozložení strun a tónů cimbálu .....	25
Obr. 1.5.3: Křížení strun cimbálu .....	25
Obr. 2.1.1: Princip konstrukce kapacitního mikrofonu [10].....	26
Obr. 2.1.2: Princip konstrukce páskového mikrofonu [10] .....	27
Obr. 2.1.3: Nejčastější směrové charakteristiky mikrofonů .....	27
Obr. 2.2.1: Hřebenový filtr .....	28
Obr. 2.2.2: Bílý šum.....	28
Obr. 2.3.1: Rozdíly mezi polohou mikrofonu při tónu velké C.....	29
Obr. 2.3.2: Pohled shora na rozestavění mikrofonů .....	30
Obr. 2.3.3: Pohled z boku na rozestavění mikrofonů .....	31
Obr. 2.3.4: Fotografie z nahrávání 1 .....	31
Obr. 2.3.5: Fotografie z nahrávání 2 .....	32
Obr. 2.4.1: Tři použité typy paliček.....	33
Obr. 3.3.1: Zjednodušený popis posloupnosti událostí programu a MIDI převodníku ..	40
Obr. 3.3.2: Grafické uživatelské rozhraní pluginu.....	41
Obr. 4.2.1: Testování pluginu v DAW systému Cubase 5.....	44

## SEZNAM TABULEK

Tab. 1.3.1: Kanálová MIDI data [7] .....	17
Tab. 1.3.2: Režimy přijímače MIDI .....	17
Tab. 1.3.3: MIDI čísla not [6] .....	18
Tab. 1.3.4: Hodnoty velocity a korespondující hudební označení [7], [3] .....	19
Tab. 1.3.5: MIDI Spínače [7] .....	20
Tab. 1.3.6: MIDI povely [7] .....	20
Tab. 1.3.7: Kategorie hudebních nástrojů [3] .....	21

# ÚVOD

Tato práce má za úkol vytvořit zásuvný VST plugin modul, který bude sloužit jako třetí, výstupní část elektronického cimbálu. Prakticky se jedná o zaznamenání jednotlivých tónů cimbálu, jejich různých artikulací, hlasitostí a o následnou editaci těchto vzorků. Poté se práce zabývá implementací vzorků do technologie VST tak, aby byl zásuvný modul schopný správně reagovat na MIDI zprávy a generovat zvuk podle příslušných not.

Nejprve jsou objasněny pojmy sampler, MIDI, VST a sám hudební nástroj cimbál. Poté se práce věnuje problematice ideálního snímání zvuku cimbálu, aby se docílilo co nejdokonalšího zvuku, což v sobě zahrnuje mimo typů mikrofonů samých i místnost a správnou úpravu zaznamenaných vzorků. Zahnutí těchto vzorků do funkčního zásuvného modulu VST je další částí této práce. Zde je mimo jiné nastíněn rozdílný postup při algoritmizaci např. od virtuálního klavíru řízeného protokolem MIDI.

V závěrečných částech práce hodnotí svůj průběh, popisuje testování s hardwarovými částmi elektronického cimbálu a v závěru shrnuje cíle, kterých bylo a nebylo dosaženo. V závěru je také možné nalézt návrhy na případné budoucí vylepšení.

# 1 ZÁKLADNÍ POJMY

První kapitola obsahuje vysvětlení pojmů sampler, MIDI a VST, které se v této práci vyskytují. Také je zde popsán akustický hudební nástroj cimbál.

## 1.1 Sampler

Sampler je multifonní digitální nástroj, který může být jak softwarový ve formě programu, nebo hardwarový ve fyzické podobě konkrétního přístroje. Obecně se jedná o zařízení, které do své banky umožňuje vložit určitý počet vzorků (samplů) v podobě kratších nebo delších zvukových souborů, které se na základě příkazu spouští. Takovýmto příkazem je např. změna hodnoty fyzického ovládacího prvku u hardwarového sampleru a u softwarového to může být třeba zpráva z protokolu MIDI, který je popsán v následující podkapitole.

Další způsob jak lze samplery dělit je podle toho, jestli mají klaviaturu nebo tlakové „pady“, resp. pouze tlačítka.

První možnost tvorby rozsahu sampleru s klaviaturou je metodou zónování. Spočívá v tom, že se jeden střední vzorek použije pro celý rozsah v rámci např. jedné oktávy tak, že ostatní tóny z tohoto rozsahu jsou frekvenčně modulovány. Tato modulace však posunuje nejen fundamentální složky tónů ale i jejich formanty.

Druhá možnost, dnes častěji využívaná, dovoluje mít pro každou klávesu příslušný zvukový vzorek, protože dnešní technologie na to má již z hlediska paměti RAM a velikosti médií dostatečnou kapacitu. Klaviaturové samplery disponují často tzv. dynamikou úhozu, která na základě síly stisknutí klávesy vyhodnotí hlasitost přehrávaného zvuku. K zástupcům tlakových padů lze zařadit např. elektrické bicí se samplerem. Ty mají vedle dynamiky úhozu také možnost měnit zvuk pro konkrétní pad. Z tohoto již plyne velká variabilita samplerů obecně. Máme-li k dispozici rozsáhlé zvukové banky, jsme schopni vytvářet komplexní skladby bez nutnosti fyzické přítomnosti daných nástrojů. K tvorbě elektronické hudby se hojně využívají maticové pad samplery, které do svých tlačítek, umístěných do libovolně velké matice, umožňují vložit libovolné vzorky. Uživatel je poté schopen tyto vzorky rytmicky dle libosti a každý ještě např. dále modulovat. Většina dnešních samplerů má nastavitelný přechod mezi samplery – délku dozvuku a rychlost náběhu pro plynulost nebo naopak pro zřetelnost. [1], [6], [11], [12]

V případě elektronického cimbálu a tedy i této práce bude využíván softwarový sampler s již pevně vloženými a upravenými vzorky pro každý tón zvlášť.

### 1.1.1 Historie samplerů

Samplery, nejprve jako hardwarové přístroje, vznikly kolem roku 1980 a v porovnání s dnešními modely neměly velké možnosti. „Prvními zástupci jsou klaviaturový Fairlight CMI z roku 1979 a o 20000 amerických dolarů levnější Emulator od EMU SYSTEMS z roku 1981. Tento osmibitový sampler stál asi 8000 amerických dolarů

a dokázal hrát pouze dva zvuky současně, ale pořád nebyl masově dostupný.“ První levný sampler byl na trhu dostupný od roku 1985 – ENSONIQ Mirage. Dal se pořídít pod 1700 amerických dolarů, byl opět osmibitový, operoval se 128 kB RAM a 3,5 palcovou floppy disk mechanikou, uživatelský interface byl primitivní a komunikace probíhala pomocí hexadecimálních kódů. Měl ovšem na svou dobu zdařilou firemní databázi vzorků, přičemž jejich kvalitu nebylo možné dosáhnout v amatérských domácích podmínkách. [1]

„Na začátku 90. let se začaly elektronické nástroje výrazně rozšiřovat a to způsobilo i rozšíření samplerů z profesionálních studií do řad hudebníků. Některé firmy ale přestaly existovat, samplery nevyrobí, nebo se spojily s jinými.“ Samplery jsou dnes pro původní využití zastaralé, vytlačují je technologie založené na matematickém modelování. V oblibě jsou dnes pro integraci do různých klávesových a kytarových syntetizátorů, looperů jako efektové kontroléry nejen pro elektronickou hudbu. V dnešní době se hardwarové samplery široce protínají se syntetizátory a nabízí tudíž nekonečné možnosti úprav jak syntetických zvuků, tak nahraných, živých zvuků. [1]

Softwarové samplery přicházely na scénu v 90. letech a po roce 2000, kdy se zvýšil výkon počítačů, a ty tak začínaly zvládat i operace se zvukem v reálném čase. „Lze je rozdělit na *virtuální rytmy*, *frázové samplery* a *standartní samplery*. První typ využívá hlavně vzorky rytmických bicích nástrojů a simuluje tím tradiční *rytmer*.“ To je matrice úderů, kdy na časové horizontální ose se nacházejí jednotlivé pravidelné části taktace skladby a na vertikální ose leží zvuková banka použitelných vzorků. „Frázové samplery pracují podobně jako rytmy s tím rozdílem, že nevyužívají krátké vzorky, ale několikataktové *patterns* – vzory, které jsou tvořené i z jiných než perkusivních nástrojů, a vrství se např. za sebe a tvoří smyčky.“ Standartním samplerem se označuje ten, který má možnost uložit delší vzorky a nabízí např. jejich prolínání, různé možnosti ovládaní atd. V dnešní době je jejich využití velmi široké, pokročilejší třeba zvládnou rozdělení dlouhého zvukového souboru např. stupnice do jednotlivých tónů a ty poté uloží a reagují např. na údaje z klaviatury. [1]

### 1.1.2 Princip funkce

Pokud má sampler pracovat se vzorky nahraného nebo přejatého zvuku, je nutné, aby byl analogový signál převeden na digitální. Vzorek poté může být uložen do paměti RAM nebo na úložné zařízení. Po zpracování je při přehrávání převeden zpět na napěťové úrovně, čili zpět do analogové formy. [7]

### 1.1.3 Analogově – digitální převod

A/D převod je možné rozdělit do tří částí: vzorkování, kvantování, kódování.

**Vzorkování** lze popsat jako děj, při kterém se spojitý analogový signál v čase rozdělí do diskrétních vzorků, jejichž počet za jednu sekundu je určen vzorkovacím kmitočtem  $f_{VZ}$ . Z toho vyplývá, že čím bude vzorkovací kmitočet vyšší, tím přesnější bude převod analogového signálu na digitální. Při vzorkování je žádoucí dodržet vzorkovací teorém

$$f_{VZ \min} \geq 2 \cdot f_{\max}, \quad (1.1)$$

který říká, že minimální vzorkovací frekvence  $f_{VZ\ min}$  musí být alespoň dvakrát větší než maximální frekvence  $f_{max}$ , která se v signálu vyskytuje. Pokud podmínka splněna není, dochází k tzv. *aliasingu*, kdy dochází k překrytí frekvenčních spekter vzorkovaného signálu a tudíž ke ztrátě informace. Frekvenční rozsah lidského sluchového orgánu je u naprosté většiny jedinců maximálně od 20 Hz do 20 kHz. Proto by minimální vzorkovací frekvence měla být alespoň 40 kHz. V praxi se nejčastěji používají hodnoty 44,1 kHz pro obecné komerční využití a 48 kHz, 96 kHz pro studiové účely. [7]

**Kvantování** je proces, kdy se diskrétním vzorkům přiřazuje nejbližší možná kvantovací hladinu tak, aby se číselná hodnota od původní napěťové hodnoty signálu lišila, co nejméně. Údajem, který reprezentuje počet kvantovacích hladin je bitová hloubka. Je to počet bitů největšího čísla v binární soustavě snížený o jeden bit vyhrazený znaménku, které je možné pomocí daného počtu bitů zapsat. Tento počet převedený do desítkové soustavy určuje počet kvantovacích hladin. Lze jej vyjádřit jako číslo 2 umocněné na bitovou hloubku:

$$K_{10} = 2^{n-1}, \quad (1.2)$$

kde  $n$  značí bitovou hloubku. Máme-li například bitovou hloubku 3 bity, budou kvantovací hladiny 4. Ve zvukové technice se používají hodnoty 16, 20 a 24 bitů; to odpovídá počtu hladin 32 768, 524 288 a 8 388 608. Kvantovací krok převodníku  $Q$  lze určit z rovnice:

$$Q = \frac{1}{2^{(n-1)-1}}, \quad (1.3)$$

kde  $n$  značí opět bitovou hloubku. Při kvantování nastává jev způsobený přiřazením vzorků do nejbližších, avšak ne dokonale odpovídajících hladin. Označuje se kvantovací šum a zmenšuje se při použití vyšší bitové hloubky. Kvantovací šum zdola omezuje dynamický rozsah navzorkovaného signálu. Odstup signálu od šumu, anglickou zkratkou označován SNR – Signal to Noise Ratio, který využívá celý rozsah převodníku, od kvantovacího šumu určíme pomocí odhadu:

$$SNR = 20 \log \left( \frac{1}{Q} \right), \quad (1.4)$$

kde  $Q$  značí kvantovací krok převodníku. Bitovým hloubkám 16, 20, 24 bitů odpovídají hodnoty  $SNR$  90,31 dB, 114,39 dB a 138,47 dB. Signály, které nebudou využívat celý rozsah A/D převodníku budou mít odstup od kvantovacího šumu menší. Lze tvrdit, že kvantování je proces, jehož přesnost ovlivňuje výslednou kvalitu A/D převodu. [7]

„**Kódování** je převod vzorků diskrétních v čase i v úrovni na binární číslo. V profesionální zvukové technice se pro další zpracování digitálních zvukových signálů používá téměř výhradně PCM (Pulzně kódová modulace), při které nedochází ke ztrátě informace nebo omezení šířky pásma rychlých přechodů a při editaci umožňuje nástřih s přesností na jeden vzorek.“ [7]

## 1.2 MIDI – obecné informace

Zkratka MIDI reprezentuje následující slova v angličtině: Musical Instrument Digital Interface, což lze přeložit jako digitální rozhraní pro hudební nástroje.

Jedná se o všeobecný standard, který umožňuje komunikaci mezi hudebním zařízením a počítačem a mezi hudebními zařízeními navzájem. Protokol MIDI specifikuje společný dorozumivací jazyk. „Přirovnat jej lze k USB (Universal Serial Bus – univerzální sériová sběrnice). Je to protokol, který, stejně jako MIDI, zároveň stanovuje typy předepsaných konektorů, jenž umožňuje normovaný typ propojení a vzájemné komunikace.“ [7]

### 1.2.1 Historie, význam a popis

S představením prvních syntetizátorů v pozdních 70. letech, které byly schopné hrát pouze monofonně (1 tón současně), postupně nastávala potřeba vzájemného propojení a synchronizace dvou a více zařízení. V roce 1982 se setkali zástupci firem Roland Corp. a Sequential Circuits a řešili způsob propojení syntetizérů různé značky a typu. Společně se dohodli na protokolu UMI (Universal Musical Instrument). Do projektu se poté zapojily další společnosti, jako Oberheim a Yamaha a roku 1983 vznikl protokol MIDI 1.0. Byl prvním standardizovaným rozhraním přijatým všemi výrobci elektronických hudebních nástrojů. Kvůli nejasnostem byly v roce 1984 založeny normovací asociace a roku 1985 už vyšla podrobná norma specifikující MIDI. Norma tohoto komunikačního rozhraní se však stále vyvíjela a upravovala přibližně do roku 2000. [3], [7]

### 1.2.2 Přenos, hardware a konektory

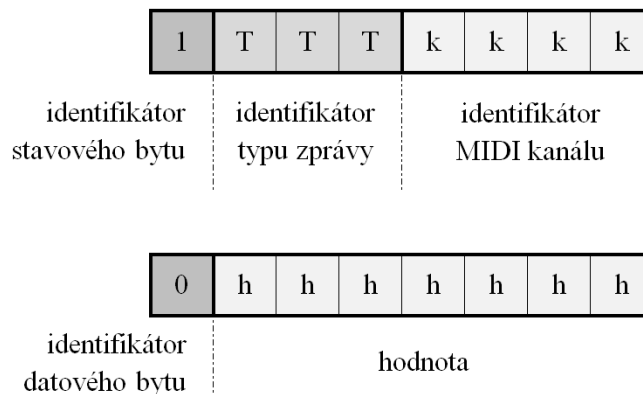
MIDI informace jsou přenášeny za sebou, pomocí sériového asynchronního protokolu. Sběrnici je 5 mA proudová smyčka. Rychlost přenosu je 31 250 b/s, se kterou je schopný přenést např. až 500 not za sekundu. Rámec se skládá z jednoho startbitu, osmi datových bitů a jednoho stopbitu. Doba trvání jednoho rámce je tedy 320  $\mu$ s. MIDI data mohou být vysílána až v 16 virtuálních kanálech. [3], [7]

Jak už bylo zmíněno dříve, MIDI standardizuje také fyzické připojení pomocí normovaných konektorů, kabelů a zásuvek. Bez použití opakovače se udává maximální délka kabelu 10 až 15 metrů (pro systémové zprávy 5 metrů), jinak bude docházet ke ztrátě dat a ke zvyšování zpoždění. Používá se DIN konektor s pěti piny, z nichž jsou využity pouze 3. Typy připojení MIDI kabelu a jednotlivých zásuvek jsou tři: In, Out a Thru. MIDI Out je výstupem posílaných MIDI dat (zpráv). Přes tento konektor proudí data jedním směrem ze zařízení, jako jsou např. klávesy, zvukové moduly, sekvencery atd. Konektorem MIDI In zařízení přijímá MIDI data z jiného zařízení. Zásuvka MIDI Thru slouží k propojení více než 2 zařízení. Funguje tak, že jakákoliv informace, která dorazí do zásuvky MIDI In, je automaticky přesměrována na MIDI Thru, nikoliv na MIDI Out. Při dvou zařízeních se propojuje MIDI Out prvního zařízení s MIDI In druhého a MIDI In prvního s MIDI Out druhého. Při zapojení více zařízení, např. tří řazených za sebe A, B, C se použije Thru konektor následujícím způsobem. Ze zařízení A vedeme z konektoru Out kabel na konektor In zařízení B. Ze zařízení B vedeme kabel konektorem Thru do konektoru In zařízení C. Nyní zařízení B pouze přijímá a přeposílá MIDI data dále do zařízení C. Kvůli reakční době součástek zahrnutých v řetězci je možné propojit maximálně 4 zařízení. Dnešní osobní počítače většinou neobsahují MIDI DIN rozhraní, tudíž kvůli kompatibilitě valná většina zařízení komunikujících skrze MIDI protokol využívá univerzální sériové rozhraní USB. Typ B na výstupu zařízení a typ A na vstupu osobního počítače. [3], [7]

### 1.3 MIDI zprávy

Označení MIDI zpráva (MIDI message) se používá pro základní datový bloku přenášející určitou MIDI informaci. MIDI zpráva se skládá z jednoho stavového bytu (Status Byte) a několika datových bytů (Data Byte). Pro přenášení informace používá MIDI protokol 8 bitová slova a pro rozlišení stavového a datového bytu používá MSB (Most Significant Bit – číslo nejvíce vlevo) bytu. Při MSB = 1 jde o stavový, při MSB = 0 jde o datový byte. Na obr. 1.3.1 je znázorněna struktura stavového a datového bytu, z které vyplývá, že stavový byte rozlišuje 8 různých typů zprávy a 16 MIDI virtuálních kanálů, datový byte dokáže rozlišit 128 hodnot. [7]

MIDI zprávy se dělí na kanálové zprávy (Channel Message) a systémové zprávy (System Message). Kanálové přenášejí ve stavovém bytu informaci o virtuálním datovém kanále a dělí se na hlasová data (Channel Voice Message) a na data režimu (Channel Mode Message). Systémové zprávy informaci o MIDI kanále nepřenášejí (jsou společné pro všechny virtuální kanály) a dělí se na zvláštní systémová data (System Exclusive Message), která umožňují přenos větších specifických datových bloků, dále na společná systémová data (System Common Message) a na data reálného času (System Real Time Message), která slouží k vzájemné časové synchronizaci několika zařízení. [7]



Obr. 1.3.1: Struktura stavového a datového bytu [7]

Pro zvětšení průchodnosti na MIDI sběrnici se případě kanálových dat nemusí posílat úplné MIDI zprávy. Pokud se nemění typ informace stanovený stavovým bytem, stačí vysílat pouze datové byty. Tento stav se nazývá průběžný (Running Status) a šetří asi třetinu kapacity sběrnice. Platí až do obdržení nového stavového bytu. [7]

Data v protokolu MIDI jsou přijímána a odesílána s touto prioritou:

- a) resetování systému
- b) data reálného času
- c) zvláštní systémová data
- d) společná systémová data
- e) kanálová data



V tabulce 1.3.1 jsou uvedena kanálová MIDI data, jejich význam a počet datových bytů a v tabulce 1.3.2 režimy přijímače MIDI, popsané níže.

Tab. 1.3.1: Kanálová MIDI data [7]

MIDI zpráva	Význam	ID	Počet datových bytů
Note Off	nota vypnuta	0	2
Note On	nota zapnuta	1	2
Polyphonic Key Pressure	individuální tlaková citlivost	2	2
Control Change	změna kontroleru	3	2
Program Change	volba programu	4	1
Channel Pressure	společná tlaková citlivost	5	1
Pitch Bend Change	ohýbání tónu	6	2

Přijímač MIDI může být nastaven pomocí parametrů OMNI (On/Off) a POLY/MONO do čtyř režimů. Parametry MONO a POLY určují, jak jsou přiřazeny jednotlivé hlasy přijímače, pokud je simultánně přijato více not. [7]

Tab. 1.3.2: Režimy přijímače MIDI

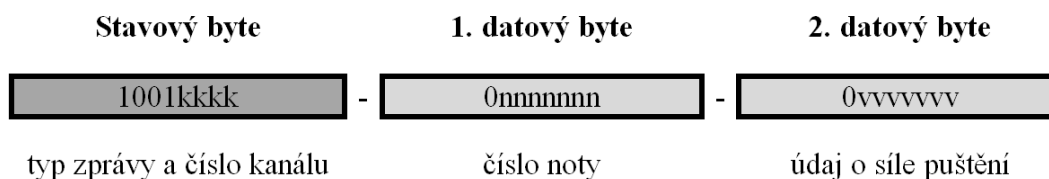
číslo a název režimu	OMNI (On/Off)	POLY/MONO	Výsledek
1 – OMNI On, POLY	reaguje na data ze všech kanálů	při přijetí více not, hrají hlasy polyfonně	nástroje ze všech kanálů hrají vícehlasně
2 – OMNI On, MONO	reaguje na data ze všech kanálů	při přijetí více not, hraje každý hlas monofonně noty přijaté na odpovídajícím kanálu	nástroje ze všech kanálů hrají jednohlasně
3 – OMNI Off, POLY	reaguje na data pouze ze zvoleného kanálu	při přijetí více not, hrají hlasy polyfonně	nástroj ze zvoleného kanálu hraje vícehlasně
4 – OMNI Off, MONO	reaguje na data pouze ze zvoleného kanálu	při přijetí více not, hraje každý hlas monofonně noty přijaté na odpovídajícím kanálu	nástroj ze zvoleného kanálu hraje jednohlasně

### 1.3.1 MIDI zprávy NOTE On a NOTE Off

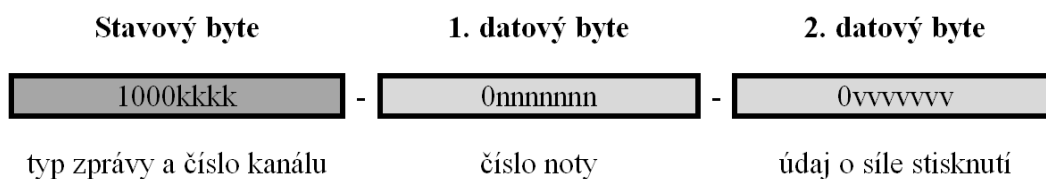
Zprávy typu Channel Voice jsou nejčastěji přenášená data na MIDI zařízení obecně. Mezi ně patří nejčastěji přenášené zprávy z klaviatury např. MIDI kláves – zprávy Note On a Note Off, které řídí spouštění a vypínání jednotlivých not. Zůstaneme-li u klaviatury, tak se NOTE On zpráva posílá při stisku klávesy. Nese v sobě dvě informace: číslo zahráné noty (viz tab. 1.3.3) a úroveň hlasitosti (velocity), se kterou byla nota zahrána. Obě hodnoty mohou být v rozmezí 0 až 127. Note Off zpráva je odesílána při puštění klávesy (odmáčknutí) a její hodnota velocity popisuje úroveň

hlasitosti při puštění klávesy. Tyto dvě zprávy se od sebe liší pouze čtvrtým bitem zleva ve stavovém bytu (Note On = 1, Note Off = 0). Přiřazení hudební dynamiky k hodnotám velocity popisuje tab.1.3.4. [3]

Zpráva Note On a Note Off se skládá ze 3 bytů. Status byte, který je první v pořadí obsahuje informaci o kanále a typu zprávy. První datový byte obsahuje informace o čísle noty a druhý o hodnotě velocity. Tyto tři bloky MIDI zpráv Note On a Note Off jsou znázorněny na obrázku 1.3.2 a 1.3.3.



Obr. 1.3.2: Struktura zprávy Note On [7]



Obr. 1.3.3: Struktura zprávy Note Off [7]

„To, že je přenášena informace o hlasitosti uvolnění klávesy, umožňuje napodobovat některé techniky hraní např. na strunné nástroje, ve skutečnosti se ale tato informace nevyužívá často a má většinou hodnotu 0, resp. 127.“ [7]

Tab. 1.3.3: MIDI čísla not [6]

oktáva	hud. označení	C	C#	D	D#	E	F	F#	G	G#	A	A#	H
-2	sub-subkontra	0	1	2	3	4	5	6	7	8	9	10	11
-1	subkontra	12	13	14	15	16	17	18	19	20	21	22	23
0	kontra	24	25	26	27	28	29	30	31	32	33	34	35
1	velká	36	37	38	39	40	41	42	43	44	45	46	47
2	malá	48	49	50	51	52	53	54	55	56	57	58	59
3	jednočárkovaná	60	61	62	63	64	65	66	67	68	69	70	71
4	dvoučárkovaná	72	73	74	75	76	77	78	79	80	81	82	83
5	tříčárkovaná	84	85	86	87	88	89	90	91	92	93	94	95
6	čtyřčárkovaná	96	97	98	99	100	101	102	103	104	105	106	107
7	pětičárkovaná	108	109	110	111	112	113	114	115	116	117	118	119
8	šestičárkovaná	120	121	122	123	124	125	126	127				

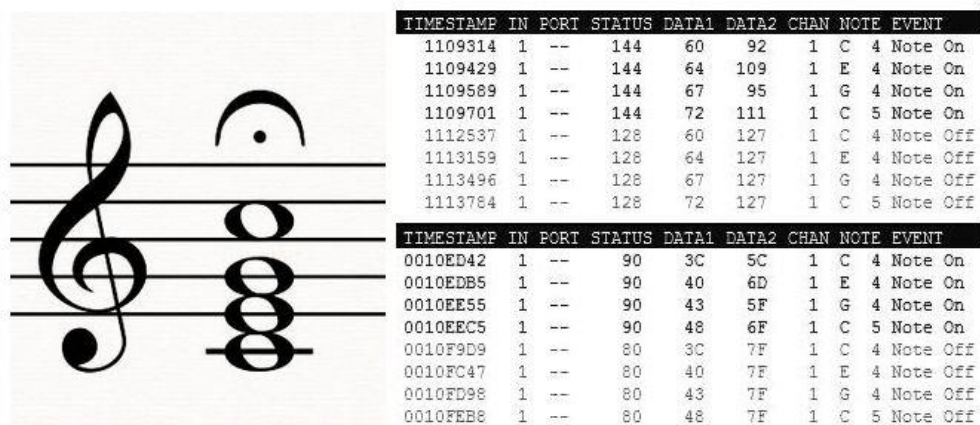
Tab. 1.3.4: Hodnoty velocity a korespondující hudební označení [7], [3]

Dynamické názvosloví	Hudební označení	Hudební notace	MIDI velocity
vypnuto	-	-	0
nejslaběji	<i>piano pianissimo</i>	<i>ppp</i>	1 až 15
velmi lehce	<i>pianissimo</i>	<i>pp</i>	16 až 31
lehce	<i>piano</i>	<i>p</i>	32 až 47
lehce až středně lehce	<i>mezzopiano</i>	<i>mp</i>	48 až 63
středně silně	<i>mezzoforte</i>	<i>mf</i>	64 až 79
silně	<i>forte</i>	<i>f</i>	80 až 95
velmi silně	<i>fortissimo</i>	<i>ff</i>	96 až 111
nejsilněji	<i>forte fortissimo</i>	<i>fff</i>	112 až 127

Klaviatury bez dynamiky úhozu by měly vysílat hodnotu velocity 64. Speciálním případem velocity je hodnota 0, která znamená „nota vypnuta“. Díky tomu není nutno při uvolnění klávesy posílat zprávu „nota vypnuta“ (Note Off, viz dále), ale zprávu nota zapnuta s dynamikou 0. Po přijetí této zprávy přestane MIDI přijímač odpovídající tón hrát. Umožňuje to využít průběžného stavu (Running Status) pro řízení stavu kláves. [7]

Na obr. 1.3.4 je pro ilustraci zobrazen kód MIDI zpráv Note On a Note Off při akordu C-dur nejprve v desítkové soustavě, níže pak v binární. Sloupec TIMESTAMP značí časovou adresaci všech MIDI zpráv, jednotkou jsou milisekundy.

Pozn.: v kolonce NOTE je nesprávné číslo oktávy, program MIDI-OX se patrně řídí dle označení standardu hudebního a ne MIDI.

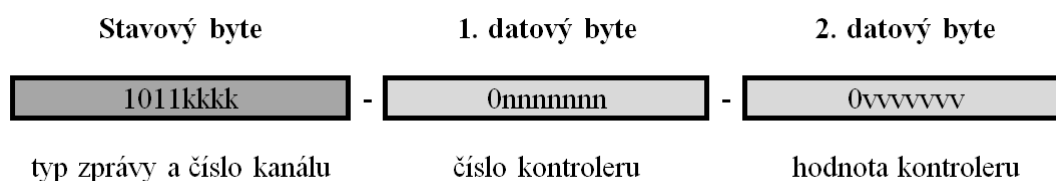


Obr. 1.3.4: C-dur akord v MIDI kódu

### 1.3.2 MIDI zpráva Control Change

Tato zpráva se českým ekvivalentem označuje jako změna kontroleru a přenáší informace o změně jeho hodnoty. Takovým kontrolerem je ovládací parametr MIDI nástroje, např. hlasitost, modulace, banka zvuků atd. Struktura zprávy obsahuje opět 3 byty. Informaci o tom, že se jedná o zprávu Control Change přenáší stavový byte 2. až 4. bitem (011). V prvním datovém bytu je obsažena informace o čísle kontroleru

a v druhém je nová hodnota tohoto kontroleru. Čísla i hodnoty kontrolerů mohou nabývat hodnot od 0 do 127. (viz obr. 1.3.5). [7]



Obr. 1.3.5: Struktura zprávy Control Change [7]

Pro většinu ovládacích zařízení MIDI je 128 hodnot, kterých může kontroler nabývat, nedostačující. Proto jsou pro některé MIDI kontrolery vyhrazeny dvě MIDI čísla s celkovým rozlišením 14 bitů, tj. 16384 hodnot, kterých může kontroler nabývat. Zprávy Control Change se z hlediska funkce dělí na průběžné, spínače, inkrementační/dekrementační a na povely. V rámci elektronického cimbálu postačí typy spínače a povely. Tabulky 1.3.5 a 1.3.6 ukazují přiřazení těchto typů jednotlivých změn kontroleru číslu kontroleru a popisují jejich funkci. Spínače nabývají pouze 2 stavů. Pro hodnoty 0 až 63 jsou vypnuté a pro hodnoty 64 až 127 jsou zapnuté. U povelů, až na povel Local Control, nezáleží na přenášené hodnotě. Funkce je totiž jasně dána přijetím čísla kontroleru, hodnota druhého datového bytu je tedy nulová. [7]

Tab. 1.3.5: MIDI Spínače [7]

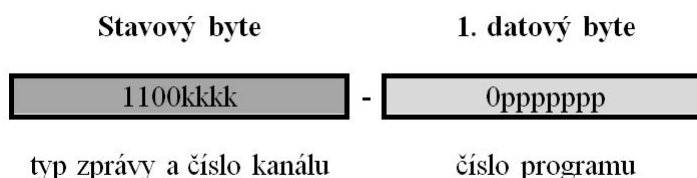
číslo	název spínače	význam
64	<i>Sustain</i>	podržení znějících tónů
65	<i>Portamento</i>	zapne efekt glissando mezi tóny
66	<i>Sostenuto</i>	prodloužení doznění všech znějících tónů
67	<i>Softpedal</i>	třetí pedál klavíru – tlumič
68	<i>Legato</i>	zapíná svazování not
69	<i>Hold 2</i>	podržení znějících tónů v jiném segmentu obálky než Sustain

Tab. 1.3.6: MIDI povely [7]

číslo	název povelu	význam
120	<i>All Sound Off</i>	všechny zvuky vypnuty
121	<i>Reset All Controllers</i>	reset všech kontrolerů
122	<i>Local Control</i>	lokální řízení zapnuto (127) / vypnuto (00)
123	<i>All Notes Off</i>	všechny noty vypnuty
124	<i>Omni Off</i>	režim Omni zapnut
125	<i>Omni On</i>	režim Omni vypnut
126	<i>Mono On</i>	režim Mono zapnut (Poly vypnut)
127	<i>Poly On</i>	režim Poly zapnut (Mono vypnut)

### 1.3.3 MIDI zpráva Program Change

Změna programu přenáší informaci o zvoleném zvukovém programu – rejstříku. Má pouze jeden datový byte přenášející číslo programu. Pro většinu nástrojů je 128 hodnot pro volbu programu málo. Programy jsou proto organizovány v bankách, které se přepínají pomocí kontroleru Bank Select, které mají číslo 0/32. Na změnu programu se nejprve očekává Bank Select MSB (0), pak Bank Select LSB (32) a až poté zprávu Program Change. Na obrázku 1.3.6 je znázorněna struktura zprávy Program Change. [7]



Obr. 1.3.6: Struktura zprávy Program Change [7]

### 1.3.4 General MIDI

GM představuje přesnější specifikaci a rozšíření MIDI standardu, omezuje se však jen na definici určitých dohod. GM tedy nezavádí žádné nové příkazy, jen stávající normu MIDI specifikuje přesněji a restriktivněji.

Zařízení, které má licenci GM a nese GM logo, musí splňovat řadu podmínek. Musí mít 24 hlasů rozdělených dynamicky pro každý kanál, musí umět přijímat na všech 16 kanálech současně, musí rozumět specifikovaným příkazům atd.

Jednou ze základních vlastností GM je uspořádaný seznam zvuků, programů, samplů. GM nijak nespecifikuje, jak má který nástroj znít – to se ponechává na výrobci. Pevně dáno je však číslo programu jednotlivého zvuku nástroje. Všechna zařízení označená GM obsahují 128 tříděných a číselně seřazených samplů. Rozdělení do kategorií ilustruje tabulka 1.3.7. [3], [9]

Tab. 1.3.7: Kategorie hudebních nástrojů [3]

číslo programu	kategorie	číslo programu	kategorie
1 až 8	<i>Piano</i>	65 až 72	<i>Reed</i>
9 až 16	<i>Chromatic Percussion</i>	73 až 80	<i>Pipe</i>
17 až 24	<i>Organ</i>	81 až 88	<i>Synth Lead</i>
25 až 32	<i>Guitar</i>	89 až 96	<i>Synth Pad</i>
33 až 40	<i>Bass</i>	97 až 104	<i>Synth Effects</i>
41 až 48	<i>Strings</i>	105 až 112	<i>Ethnic</i>
49 až 56	<i>Ensemble</i>	113 až 120	<i>Percussive</i>
57 až 64	<i>Brass</i>	121 až 128	<i>Sound Effects</i>

## 1.4 VST, ASIO

Systemy VST (Virtual Studio Technology) a ASIO (Audio Streaming Input Output) jsou technologie vyvinuté pro zpracování signálu v reálném čase na osobních počítačích. Důraz se klade na minimální dopravní zpoždění, které vzniká při převodu signálu ve zvukové kartě do digitální podoby a zpět do analogové, na spolehlivost a na úplnou kontrolu nad datovými toky v systému. I přesto, že oba systémy VST, ASIO vyvinula firma Steinberg, jsou na sobě nezávislé a lze je kombinovat s jinými. [7]

### 1.4.1 VST

Tato technologie umožňuje komunikaci softwarových hudebních nástrojů a efektů s počítačovými programy. Rozdělit se podle funkce dají na VST, což jsou moduly, které pouze upravují do nich přichozí signál a na VSTi, které generují vlastní signál a popřípadě jej umožňují dále upravovat. To ostatně znázorňuje malé písmeno i, které označuje instrument – nástroj. Softwarové samplery a syntezátory spadají do této skupiny, a tudíž i softwarový sampler se zvuky cimbálu může nést označení VSTi. Další rozdělení může být třeba na dané verze VST. První verze se objevila v roce 1996, verze 2.0 vznikla 1999, roku 2006 přišlo rozhraní 2.4 a verze 3.0, 3.5, 3.6 v letech 2008, 2011 a 2013. Vzájemná spolupráce počítačových programů a těchto zásuvných plugin modulů ve výsledku dokáže nahrazovat hardwarové studiové nástroje a efekty. Systém VST implementuje správu plugin modulů, nejčastěji uvnitř DAW (Digital Audio Workstation), což je počítačová aplikace, která slouží k nahrávání, úpravě, míchání zvukových signálů (ProTools, Cubase, Studio One, Logic Pro atd.) V oblasti VST se jedná o tzv. hostitelské aplikace, které zajišťují komunikaci mezi uživatelským softwarovým rozhraním a zvukovou kartou a jejím ovladačem ASIO, který je výrobcí dodáván ke každé zvukové kartě. [7]

### 1.4.2 ASIO

Systém ASIO poskytuje hostitelské aplikaci vstupní a výstupní toky zvukových dat a stará se o správnou komunikaci mezi zvukovou kartou a DAW. „Rozhraní není vázáno na pevný počet vstupních a výstupních kanálů, neomezuje vzorkovací kmitočet. Pro snížení dopravního zpoždění se používá dvojitá vyrovnávací paměť pro každý kanál, která je volitelná uživatelem.“ Při jejím zvětšování dochází ke zvyšování dopravního zpoždění a při její nedostatečné velikosti dochází ke slyšitelnému zhoršení kvality zvuku, kdy za nastavený časový interval karta nezvládá zpracovat data. [7]

## 1.5 Cimbál

Cimbál je strunný lidový nástroj, na který se hraje paličkami a je základní prvek v obsazení cimbálové muziky na Moravě, v Maďarsku a na Slovensku. V cimbálové muzice plní úlohu vyplňování harmonických mezer mezi ostatními nástroji, jako jsou typicky housle, viola, kontrabas, klarinet a flétna. Interpreti těchto nástrojů zpravidla stojí rozestoupeni okolo cimbálu. V dnešní době je kromě folklorních muzik a souborů využívány i na sólovou produkci a tvorbu ve vážné hudbě i v ostatních a experimentálních směrech.

### 1.5.1 Historie

Cimbálu a jemu podobných nástrojů je celosvětově velmi mnoho různých typů a tím i různých barev zvuků. Řadí se do skupiny chordofonů a k nejstarším hudebním nástrojům. S jeho vyobrazením se lze setkat na starých iránských a asyrských reliéfech. Jeho raná podoba v podobě malého a lehkého lichoběžníkového nástroje se vyskytuje v perské kulturní oblasti, na Kavkaze a na západě Řecka. [5], [4]

Nástroj se do Evropy dostal prostřednictvím arabských obchodníků a posléze ji celou pokryl. Rozeznával se dřevěnými tyčinkami nebo nehty. Typická hra paličkami je doložena k roku 1300. O 20 let později je doloženo první vyobrazení *psaltéria*, označení pro obecný drnkací nástroj. [5], [4]

K prvnímu přiblížení do dnešní podoby došlo v roce 1690 v Merseburgu u Halle, kdy ho převratně zdokonalil Pantaleon Hebenstreit. Nástroj celkově zvětšil a opatřil ho systémem kobylek, čímž značně zvětšil jeho rozsah. Další vylepšení provedl v roce 1792 jistý kapelník Zimmermann, který zvýšil jeho hlasitost a zpevnil nástroj rozmístěním duší pod dělicí kobylky. Z venkovského prostředí Čech, Moravy a Slezska se objevují první zprávy o cimbálu od poloviny 15. Století. [5], [4]

S názvem cimbál se lze v českých literárních pramenech setkat od roku 1680. Od roku 1850 začala dechová hudba vytlačovat muziky s malým cimbálem, kterých poté ubývalo. Největší průlom ve zdokonalení a vytvoření velkého pedálového nástroje, který je již velmi podobný s dnešní verzí zhotovil Vensel Josef Schunda roku, který rozšířil rezonanční skříň a postavil cimbál na čtyři nohy. V roce 1874 ho vybavil pedálovým systémem a dbal na jeho výtvarné řešení. Tento zdokonalený cimbál vytlačil původní nástroje a rozšířil se po celé Evropě. O zdokonalení cimbálu se ve 20. století zasloužil také rod Boháků, počínaje Lajosem Bohákem, zakladatelem firmy. Vstup velkého cimbálu do české lidové hudby nastal na přelomu 19. a 20. století. Od té doby se nástroj pomalu integroval do převážně moravské lidové kultury, vznikaly první cimbálové muziky a tradiční způsoby lidové hry a tance sbírané od 90. let 19. století jsou zachovávány dodnes. [5], [4]

Na obrázku 1.5.1 je zepředu zobrazen cimbál značky ARTEX Bohák, budapešťské firmy, která ho vyrobila v roce 1983.

### 1.5.2 Konstrukce nástroje

Při výrobě se nejprve zhotoví rezonanční skříň z různých druhů dřeva (smrk, buk, javor, překližka), které jsou k sobě sešroubovány. Do této rezonanční skříně se umísťuje tzv. pancíř, což je litina – kovová vzpěra vážící okolo 25 kg. Vzpěra brání zhroucení skříně pod velkým tahem strun. Do rezonanční skříně se dále vloží duše, které tak jako u houslí přenáší přes kobylku vibrace na spodní desku. Konstrukce horní rezonanční desky a její žebrování nejvíce ovlivňuje kvalitu zvuku nástroje a je předmětem stálého výzkumu a vývoje. Kobylky jsou vyrobené z javorového dřeva a mají tvar malých hříbků, na kterých jsou pražce z mosazného drátu. Závěsné a ladící kolíky jsou z oceli špičkové kvality, mají niklovou povrchovou úpravu a levotočivý závit. [5]

Struny se dělí na opřádané – pro nižší tóny a neopřádané (hladké) pro vyšší tóny. Počet strun pro jednotlivé tóny může být 1 až 4, jejich povrch se pocínuje, aby byly odolnější. Na opřádku se používá měděný drát a dává basovým tónům vyrovnaný zvuk. Tlumiče jsou plstěná dusítka, která tlumí vibrace strun. Jsou ovládány pomocí drátu







Obr. 1.5.2: Rozložení strun a tónů cimbálu



Obr. 1.5.3: Křížení strun cimbálu

## 2 TVORBA VZORKŮ

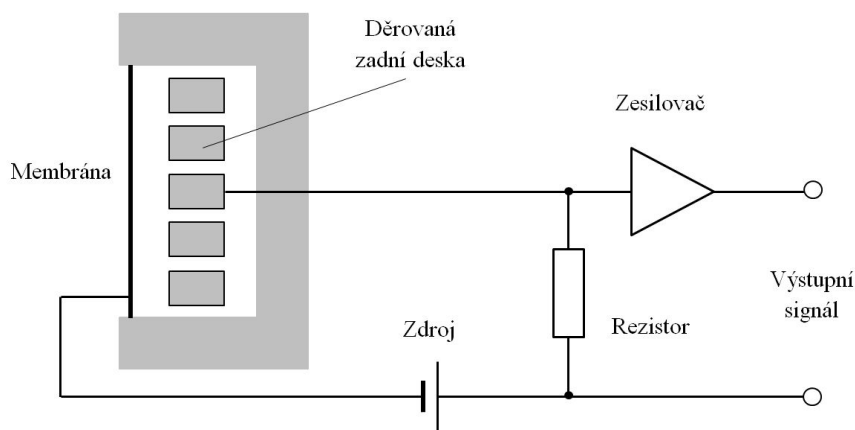
Tato kapitola pojednává o vytvoření zvukové banky pro cimbálový sampler. Jedná se tedy o zaznamenání jednotlivých tónů, hlasitostí a artikulací. Nejprve však popisuje typy mikrofonů a vhodné techniky pro snímání nástroje.

### 2.1 Typy mikronů

Volba mikrofonů je pro výsledný zvuk důležitá, chceme-li zachovat co možná nejuvěrnější zvuk cimbálu. Tato podkapitola popisuje typy, které byly pro záznam použity.

#### 2.1.1 Kapacitní mikrofony

Těmto mikrofonům se také často říká kondenzátorové. Je-li do nich přivedeno tzv. fantomové napájení, nejčastěji 48 V, tak na svých destičkách uchovávají elektrický náboj podobně jako kondenzátory. Jejich princip funkce spočívá v měnící se vzdálenosti izolačních destiček a tím se měnícím napětím. Vzdálenost destiček, z nichž jedna je pevná a druhá má podobu pružné plastické membrány, se mění působením změny akustického tlaku právě na tuto membránu. Připojí-li se mezi napájecí napětí a kapacitu určitý odpor, lze v závislosti na protékajícím proudu odvodit napěťové změny, které jsou přiváděny na vstup vysokoimpedančního zesilovače, viz obrázek 2.1.1 (tranzistory FET nebo elektronkové zesilovače). Tyto mikrofony jsou velmi citlivé v celém slyšitelném pásmu (20 Hz až 20 kHz) a při kvalitních zesilovačích i nízkošumové. K danému snímání jsou tedy velmi vhodné. [10]

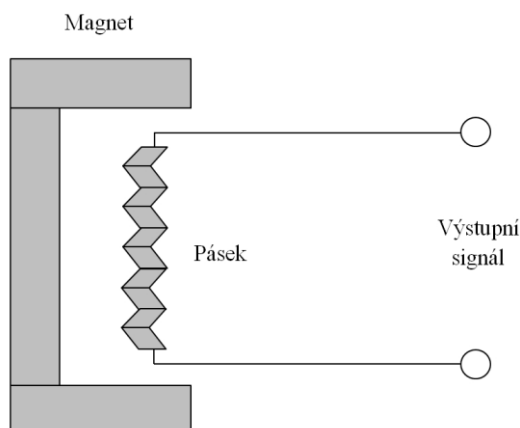


Obr. 2.1.1: Princip konstrukce kapacitního mikrofonu [10]

#### 2.1.2 Páskové mikrofony

Princip funkce páskového mikrofonu vychází z principu dynamického. U dynamického vzniká signál tak, že změna akustického tlaku rozkmitává cívku připevněnou

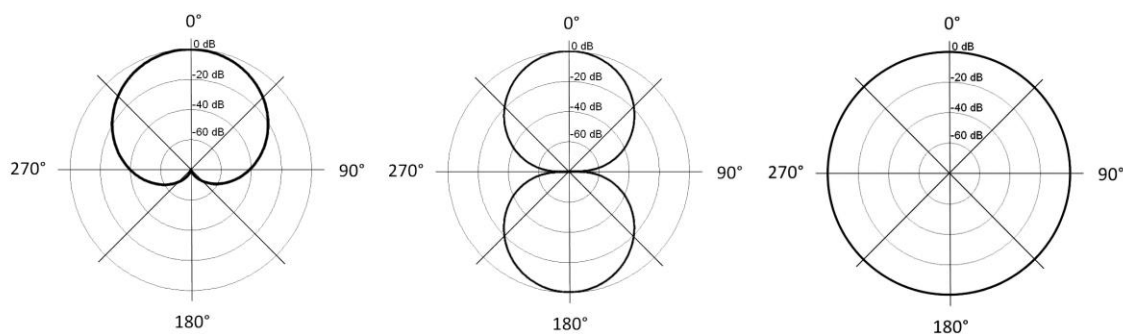
k membráně. Cívka s membránou kmitá v magnetickém poli permanentního magnetu, tudíž se v závitech cívky indukuje proud. Je však velice slabý a musí se zesilovat. Páskové mikrofony se liší pouze nahrazením cívky s membránou tenkým vodivým páskem. Je to vlastně dynamický mikrofon s jedním závitem cívky, vzniklé napětí je proto velmi slabé a většinou se přímo v mikrofonu se zesiluje. K tomu je zapotřebí fantomové napájení, jako u kapacitních mikrofonů. Páskový mikrofon má většinou osmičkovou směrovou charakteristiku, viz kapitola 2.1.3. Princip konstrukce je na obrázku 2.1.2. [10]



Obr. 2.1.2: Princip konstrukce páskového mikrofonu [10]

### 2.1.3 Směrové charakteristiky

Směrová charakteristika je podobně jako frekvenční charakteristika, která udává míru citlivosti ve frekvenčním spektru, důležitou vlastností mikrofonu. Ze směrové charakteristiky lze určit, s jakou citlivostí je mikrofon schopen snímat mimo jeho osu. Znárodnuje se do kruhového diagramu tak, že spojnice  $0^\circ$  a  $180^\circ$  je osou mikrofonu s tím, že  $0^\circ$  značí mikrofon zepředu. V každém dalším úhlu s touto osou je citlivost dána zobrazením příslušného útlumu. Čím je charakteristika blíže ke středu, tím hůře je schopen snímat z daného úhlu. Obrázek 2.1.3 ukazuje tři nejčastější mikrofonní směrové charakteristiky: kardioidní, osmičkovou a všesměrovou – kulovou. Jelikož je většina mikrofonů symetrických, někdy se každá půlka diagramu používá pro jiné kmitočty. Pro jiné kmitočty lze také využít jiný typ čar. [10]



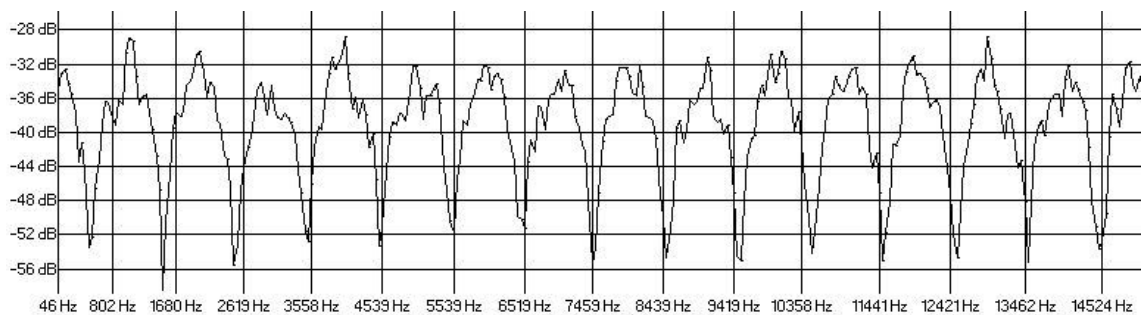
Obr. 2.1.3: Nejčastější směrové charakteristiky mikrofonů

## 2.2 Stereofonní snímání

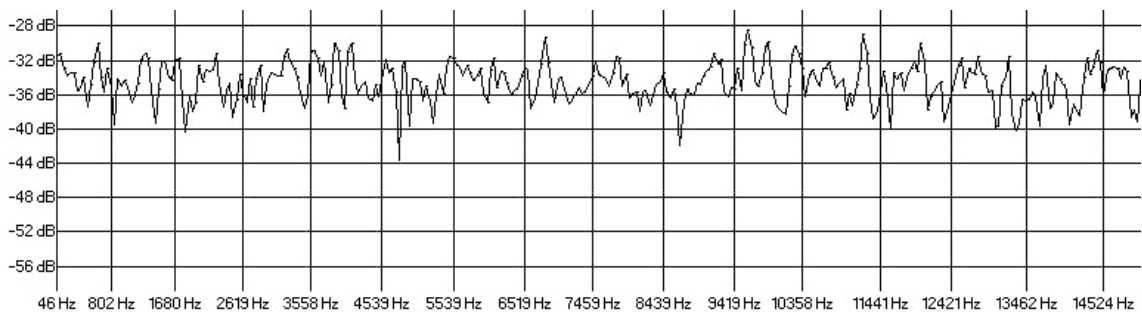
Cimbál lze v mnoha ohledech přirovnat klavíru, v některých se ale odlišuje. To, v čem jsou si tyto dva nástroje podobné, kromě vzniku tónu, je poměrně velký rozsah, takže zasahují do nízkých i vysokých kmitočtů. Dále se u nich uplatňuje dílčí význam stereofonního snímání (kromě lepšího pokrytí rozsahu), ale u obou jiným způsobem. Klavír se zpravidla snímá stereofonně, aby se docílilo rozdělení pravé a levé ruky, čili melodie ve vyšším rejstříku a harmonie v nižším rejstříku. Díky konstrukci a rozložení strun na cimbálu dostaneme však frekvenčně podobný zvuk z obou snímaných stran. Liší se ovšem v poměru jednotlivých tónů – některé tóny zní s větší intenzitou z levé strany, jiné zase z pravé. Proto se na vytváření vzorků pro cimbálový sampler využijí hlavně stereofonní techniky snímání.

U stereofonních typů snímání je třeba dávat pozor na jev zvaný hřebenový filtr, jehož vznik je většinou nežádoucí. Nastává, pokud se zvuk vlivem rozdílných vzdáleností zdroje zvuku od mikrofonů na některých frekvencích kombinuje ve fázi a dochází k součtu obou úrovní, zatímco jiné frekvence se kombinují v protifázi a při mixu signálů z mikrofonů dochází k částečnému, či úplnému odečítání signálu. [8], [10]

Hřebenový filtr lze pozorovat na obrázku 2.2.1. V zobrazeném případě se jedná o součet dvou stejných signálů, konkrétně bílého šumu, které jsou od sebe zpožděny o 1 ms, což by při teplotě vzduchu 20° C odpovídalo rozdílu vzdáleností mikrofonů od bodového zdroje asi 34cm. Na obr. 2.2.2 je pro srovnání původní signál bílého šumu. Obrázky jsou výsekem spektra výstupu FFT spektrální analýzy programu WaveLab.



Obr. 2.2.1: Hřebenový filtr



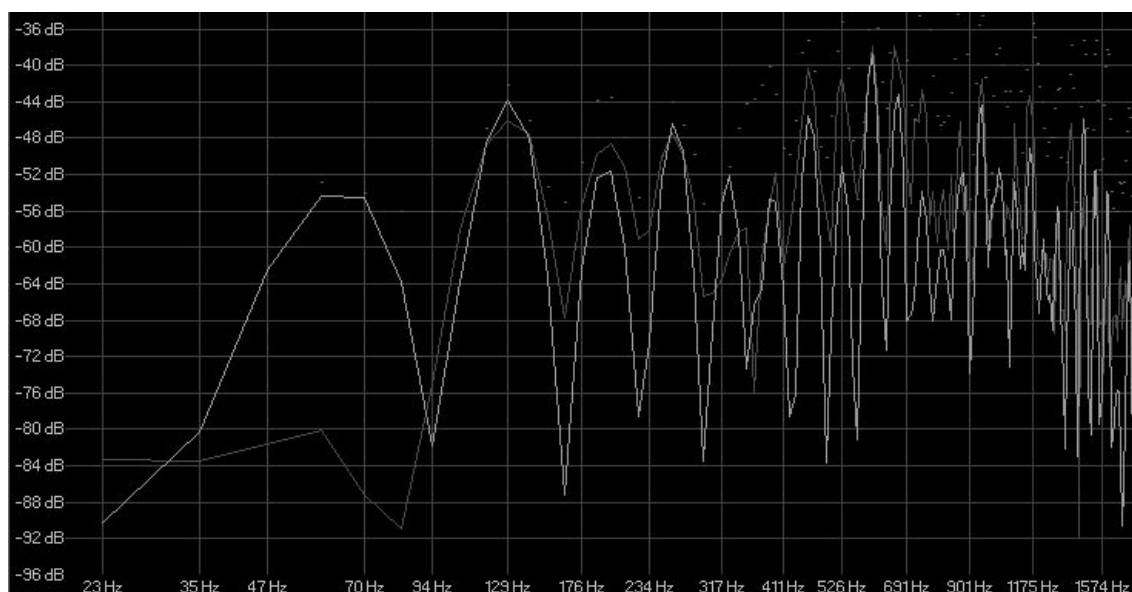
Obr. 2.2.2: Bílý šum

Problémy s fází lze značně omezit vhodným, souměrným umístěním mikrofonů od zdroje zvuku, nebo jejich vhodnou konfigurací, např. XY, která využívá bezprostřední blízkosti kapslí mikrofonního páru a je zmíněna v následující podkapitole.

## 2.3 Způsob snímání

Nahrávání vzorků pro elektronický cimbál proběhlo 28.4.2017 v bezodrazové komoře v budově T12, VUT FEKT v Brně. Nahrávaným nástrojem byl cimbál značky Jiří Galuška, Ostrožská Nová Ves, vyrobený roku 2001. Při živé produkci se cimbály nejčastěji snímají téměř kontaktně dvěma kapacitními mikrofony s kardioidní charakteristikou namířenými do rezonančních otvorů na stranách nástroje.

Při dalším zkoumání vyzařování nástroje v rámci semestrální práce byl umístěn velkomembránový kapacitní mikrofon pod spodní desku cimbálu a bylo zjištěno, že téměř všechny tóny velké oktávy rezonují přes tuto desku i směrem dolů pod nástroj. Tato část spektra spadá do nižších kmitočtů, které s dobrými výsledky snímá páskový mikrofon. Proto bylo při návrhu rozhodnuto, že bude při nahrávání pod cimbálem umístěn. Na obrázku 2.3.1 je pro ilustraci zdůvodnění tohoto umístění zobrazeno FFT frekvenční spektrum nejnižšího tónu rozsahu cimbálu, velkého C v programu WaveLab. Tmavší čára přísluší signálu z mikrofonu namířeného shora do rezonančního otvoru (v obrázcích 2.3.2 a 2.3.3 označen A, B) a světlejší čára ukazuje podobu spektra získaného kapacitním velkomembránovým mikrofonom AKG p420 umístěným pod rezonanční deskou nástroje (v obrázcích 2.3.2 a 2.3.3 označen P). Fundament tónu velké C, který má v přirozeném ladění při  $a^1 = 442$  Hz (na tuto frekvenci byl nástroj naladěn) frekvenci 65,7 Hz. Z těchto informací je na obrázku 2.3.1 jasně patrný rozdíl ve velikosti signálu mezi polohou A, B a P mikrofону na fundamentu tónu velké C, což je asi 26 dB. Páskový mikrofon, který byl pro tento účel vybrán, nese označení Rode NTR.



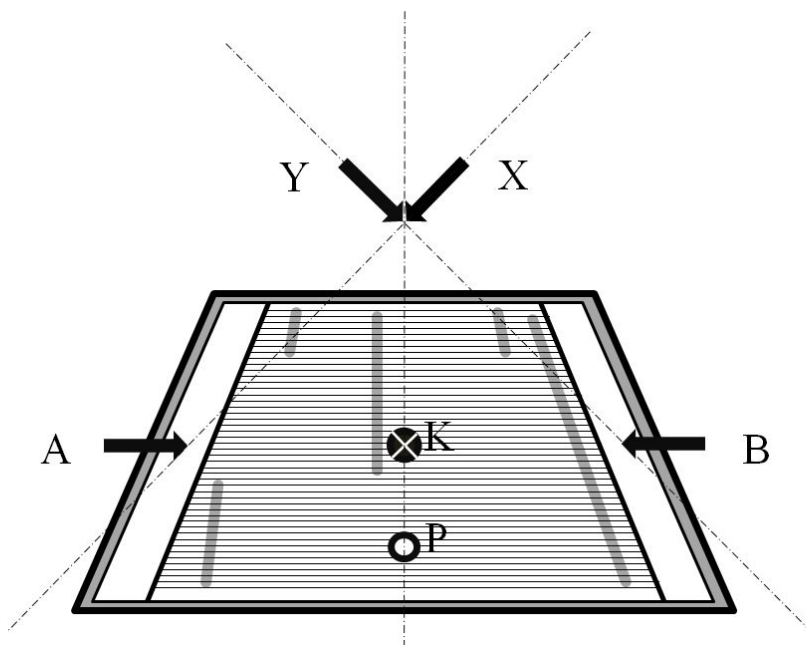
Obr. 2.3.1: Rozdíly mezi polohou mikrofónu při tónu velké C

Hlavními mikrofony, mířícími do rezonančních otvorů, byl pár kondenzátorových kardioidních mikrofonů AKG CK 91. V obrázcích 2.3.2 a 2.3.3 jsou mikrofony označeny A, B.

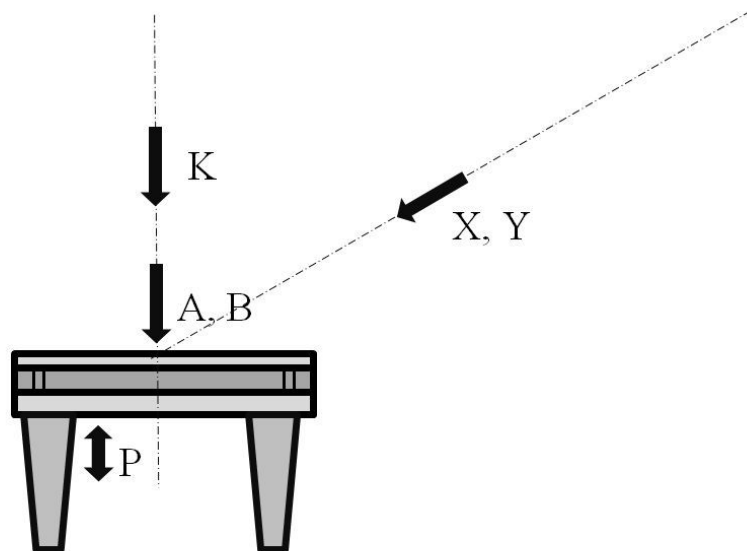
Pro další snímání byl vybrán velkomembránový kondenzátorový kardioidní mikrofon Rode NT2, umístěný cca 70 cm nad cimbálem. Důvodem pro použití této konfigurace je posílení artikulačních schopností nástroje.

I přesto že byly snímány jednotlivé tóny zvlášť, tak se při sešlápnutém pedálu rozeznáním jedné struny rozeznávají i další vyšší harmonické odpovídajících strun. Proto je důležité snímat nejen kontaktně ale i ambientně, z větší dálky, kde se jednotlivé kmitočty spojují. Tato konfigurace se při živých vystoupeních a nahráváních příliš nepoužívá, kvůli zvýšeným přeslechům a nepohodlí interpreta, pro účely vytváření zvukové banky v bezodrazové komoře je vhodná. Je přirozené zvuk vyzařovaný z nástroje snímat i z větší dálky. Cimbál byl proto snímán také stereo párem mikrofonů Rode NT5, umístěným před cimbálem v konfiguraci XY, kdy jsou mikrofonní kapsle velmi blízko u sebe a je mezi nimi úhel 90°. Umístění všech mikrofonů lze vidět na obrázcích návrhu 2.3.2 a 2.3.3 a také na fotografiích z nahrávání. (obrázky 2.3.4 a 2.3.5)

Při nahrávání za vzorkovací frekvence 48 kHz a bitové hloubky 24 bitů byl využit mikrofonní předzesilovač RME OctaMic II, kterým se digitálně směřoval signál do režijní místnosti. K dispozici byly tudíž ještě 2 další vstupy pro mikrofony, které sice byly využity pro snímání cimbálu z dálky cca 3 metry, avšak z výsledné nahrávky musel být signál z těchto mikrofonů vyřazen z důvodů slabého signálu a tím zvýšeného rušení, problematika je dále řešena v podkapitole 2.4.1.



Obr. 2.3.2: Pohled shora na rozestavění mikrofonů

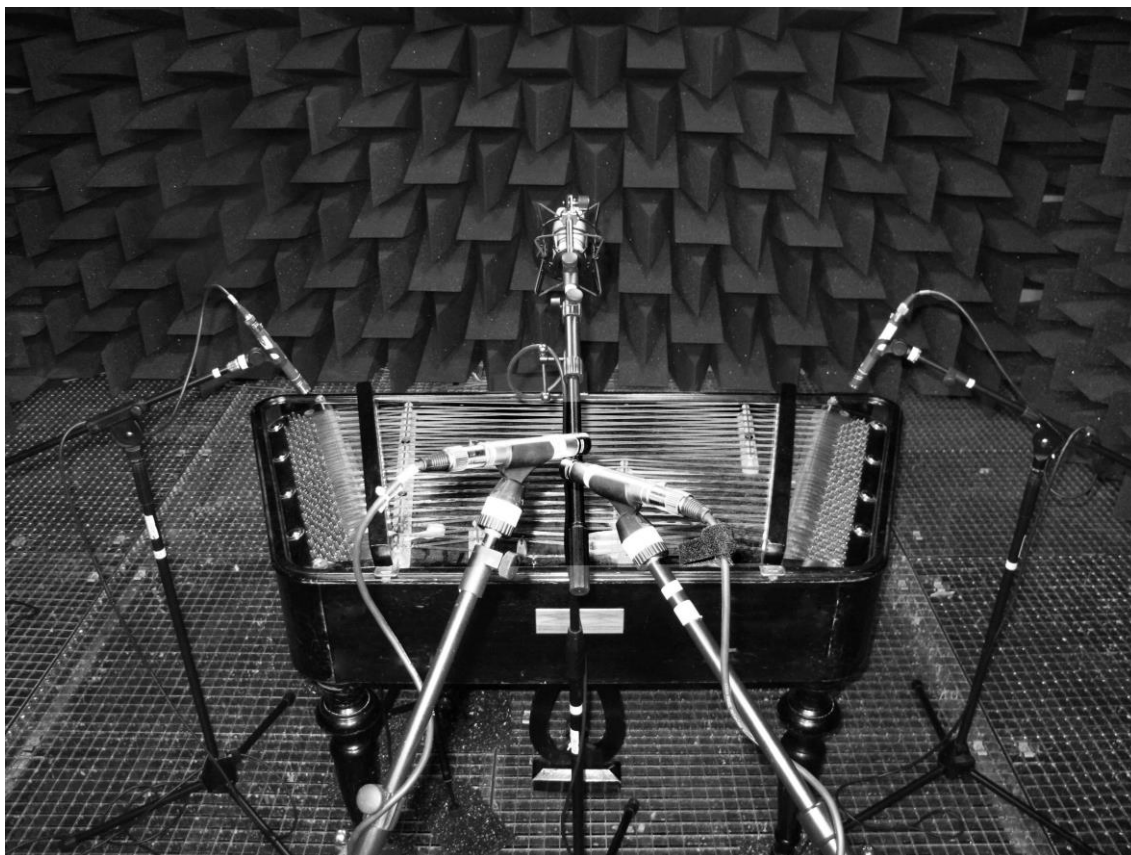


Obr. 2.3.3: Pohled z boku na rozestavení mikrofonů



Obr. 2.3.4: Fotografie z nahrávání 1





Obr. 2.3.5: Fotografie z nahrávání 2

## 2.4 Nahrávání, třídění a editace vzorků

Při návrhu organizace a systému nahrávání bylo přihlédnuto k reálnému zvuku výsledného sampleru, ale i k časovému omezení nahrávacího procesu a výsledné míře komplexnosti. Tvorba vzorků je tedy kompromisem mezi co nejvěrnějším výsledným zvukem a jednoduchým provedením.

Kvůli pokrytí artikulačních a dynamických možností nástroje byly nahrávány tři druhy paliček, viz obrázek 2.4.1 (hustě obalené, tence obalené a neobalené), tři druhy dynamiky a tóny pro oba dva stavy pedálu (*staccato*, *tenuto*). To dohromady na 58 tónech cimbálu tvoří 1044 vzorků. Tence obalené (tvrdé) a neobalené paličky jsou vyrobeny z ořechového dřeva, měkké pak z třešňového. Všechny byly obaleny vatou. Tři hlasitostní úrovně jsou rovnoměrně zeslabovány tak, aby vytvořily počet dynamických hladin – aby mezi třemi zaznamenanými dynamikami byly menší rozdíly.





Obr. 2.4.1: Tři použité typy paliček

### 2.4.1 Mix, střih a úprava vzorků

Tato část práce probíhala s lehce pozměněným pořadím, než bývá u míchání nahrávek zvykem. Jednotlivé tóny se nahrávaly za sebou, téměř bez střihů, a proto bylo velmi vhodné smíchat poměry ze všech mikrofonů ještě před střihem. Docílilo se tím toho, že celý rozsah nástroje ve všech dynamikách a artikulacích bude znít přirozeně. Neupravený zvukový materiál měl celkovou délku přes pět hodin.

Signál ze stereofonních mikrofonů je do stera rozložen z pozice hráče a to plně vlevo a plně vpravo. Signály z mikrofonů Rode NT5 byly zeslabeny o 3 dB, signál z páskového mikrofonu Rode NTR zeslaben nebyl, jelikož některé vyšší tóny prostupovaly rezonanční deskou a skříní směrem dolů, což výsledný zvuk obohacovalo. Signál z mikrofonu Rode NT2 byl zeslaben o 2 dB. Hlavní pár mikrofonů AKG CK 91 také zeslaben nebyl. Celkový master byl poté zeslaben o 3,8 dB tak, aby byl nejhlasitější tón z celého nahrávání byl těsně pod hranicí ořezu. Jelikož v tomto výpisu nejsou zohledněny konkrétní citlivosti mikrofonů, tak není příliš informativní. Hodnoty gain (zisk) na mikrofonním předzesilovači však byly nastaveny téměř stejně, lze tedy alespoň poměrově určit míru výsledného zeslabení.

Po exportování stereo master stopy následoval střih jednotlivých tónů. Jelikož je cimbál lidový nástroj, nemá mechanickou dokonalost na takové úrovni jako např. klavír. Proto je docela náročné udržet sešlápnutý pedál po dobu znění jednoho tónu (někdy až 40 sekund). Přirozeně se proto při nahrávání stávalo, že tlumící mechanika, nebo židle vrzaly. Ne ve všech případech to však bylo znatelné na první poslech z místnosti režie. Proto v této části střihu bylo zapotřebí některé vzorky tónů těchto nechtěných ruchů za pomoci vhodného střihu, např. uprostřed znění tónu, zbavit. Při delším ruchu nebo při krátkých opakovaných za sebou, bylo nutné při střihu část vzorku „klonovat“. Kdyby se totiž vystříhla delší část, byl by mezi částmi slyšitelný

dynamický skok a vzorek by se zkrátil. Při klonování byla do vzorku vložena vystřihnutá část, která se v něm již vyskytuje, ale se změnou hlasitosti tak, aby byl její celkový pokles přirozený. Pokud se ruch nenacházel v bezprostřední blízkosti za úderem, tak tato metoda fungovala poměrně účinně a ve valné většině případů se tyto nechtěné ruchy podařilo odstranit.

Při nahrávání každého tónu zvlášť a následném skládání těchto tónů do souzvuků nastává součet nežádoucích frekvencí, které jsou při přehrávání jednoho vzorku neslyšitelné. Tento součet by znehodnocoval zvukový výstup sampleru, a proto po získání jednotlivých tónů následovala filtrace nežádoucích vysokých frekvencí. Ta byla prováděna filtrem typu dolní propust, který měl strmost  $-48$  dB/okt. Tento filtr musel být u každého tónu, nebo alespoň skupiny tónů, specifický a automatizovaný, aby se neomezilo frekvenční spektrum při úderu, kdy je nejbohatší na vyšší harmonické složky. Pokles těchto vyšších harmonických (5. a vyšší) nastává prakticky ihned po úderu, tudíž od této chvíle se snižoval mezní kmitočet filtru. Při dozívání dlouhých tónů cimbálu se uplatňuje hlavně 3 – 5 vyšších harmonických složek, které mají nejvyšší kmitočet okolo 1 – 4 kHz, dle výšky tónu a jeho dynamiky. Pásmo nad tyto kmitočty je tedy ve všech dozívajících tónech odfiltrováno a zároveň nedošlo ke změně barvy nástroje při excitaci struny, kdy filtrace není použita.

## 3 IMPLEMENTACE VST

Pro snazší vytvoření zásuvného modulu je potřeba využít tzv. VST SDK<sup>1</sup> – Software Development Kit, volně dostupný vývojářský balíček od firmy Steinberg, v němž je již vyřešena komunikace mezi MIDI zařízením, DAW host aplikací a zásuvným modulem. Plugin je verze VST 2 SDK, jejímž výstupem je soubor typu dynamická knihovna.

### 3.1 JUCE

JUCE<sup>2</sup> je multiplatformní opensource framework. Obsahuje set vývojářských knihoven a ve spolupráci s VST SDK je pomocí tohoto nástroje možno vytvářet VST pluginy, audio i další aplikace unifikovaně pro platformy Windows, Mac OS X, Linux a Android. Pro použití si stačí stáhnout si balíček, zaregistrovat se, otevřít program Projucer, vytvořit nový projekt a otevřít si jej poté v jednom z podporovaných kompilátorů. (Visual Studio, Xcode, Linux Makefile a Code::Blocks). Na internetových stránkách JUCE jsou k dispozici také tutoriály, popis jednotlivých tříd, jejich funkcí a argumentů a také uživatelské fórum. Pro splnění licenčních podmínek při nekomerčním využití stačí, pokud se v pluginu při startu zobrazí logo JUCE, což si Framework vytvoří automaticky a není nutné se tím zabývat.

### 3.2 Problematika, odlišnosti

Ačkoliv je cimbál v mnoha ohledech klavíru podobný, v určitých věcech se hodně liší. Například při vysílání MIDI zpráv se u MIDI klaviatury při stisknutí klávesy odešle zpráva Note On a při puštění Note Off. Toto chování dostatečně reprezentuje klavír jako akustický nástroj při převodu na MIDI zprávy, u cimbálu to ovšem funguje odlišně. Při úderu stejnou paličkou na strunu cimbál generuje v zásadě dva druhy zvuku, které se ovlivní polohou pedálu. Při hře se ale palička odděluje ze strun, aby struna mohla při sešlápnutém pedálu správně znít. Žádný fyzický impuls tedy z paličky nepříjde, aby mohl znění zastavit. To se ovlivňuje až puštěním pedálu. Problém nastává při opakovaných úderech do jedné struny za střídavého použití obou paliček (*tremolo*). Ta spočívá v opakovaných úderech oběma paličkami střídavě na jednu strunu. Na nepředvídatelné zprávy o zapnutí noty musí systém reagovat správně i bez přijaté zprávy o vypnutí noty tak, aby nedošlo např. k přebuzení na výstupu, nebo špatnému spojení přerušovaných vzorků. Návrh na řešení tohoto problému je popsán v podkapitole 3.3.5 a je částečně závislý na způsobu vysílání zpráv vysílaných z hardwarové části.

---

<sup>1</sup> Dostupné z: <https://www.steinberg.net/en/company/developers.html>

<sup>2</sup> Dostupné z: <https://www.juce.com/>

## 3.3 Řešení kódu

Při návrhu implementace bylo zvažováno více druhů řešení. V části semestrální práce nebylo definitivně stanoveno použití konkrétních knihoven a nástrojů. Jednou ze zvažovaných možností bylo použít opensource plugin *mdaPiano*<sup>3</sup>, který by byl následně upravován. To se však kvůli nutnosti zahrnout knihovnu pro spouštění zvukových souborů ukázalo jako nevhodné a zdlouhavé. Finální rozhodnutí použít framework JUCE má následující důvody. Při dodržení licenčních podmínek je zcela volně dostupný, předvytvořené třídy, jejich funkce a spolupráce s VST SDK umožňují poměrně snadné a přehledné sestavení funkčního a multi-platformně kompatibilního zásuvného modulu. Při vývoji bylo využíváno tutoriálů a fóra na webu JUCE<sup>4</sup>.

### 3.3.1 Základní prvky

Pokud se pro vytvoření nového projektu použije program Projucer, který je součástí balíčku JUCE, tak se po zvolení typu nového projektu Audio Plugin vytvoří čtyři elementární soubory *PluginProcessor* a *PluginEditor* s příslušnými hlavičkovými soubory. Objekt *PluginProcessor* se stará o zpracování dat a o kontrolu audio, MIDI vstupů a výstupů. Objekt *PluginEditor* má za úkol zprostředkovat komunikaci mezi GUI – uživatelským rozhraním a objektem *PluginProcessor*, přičemž komunikace směřuje vždy směrem *PluginEditor* → *PluginProcessor* a ne obráceně. Zdrojový kód je v případě tohoto pluginu rozdělen mezi soubory: *PluginEditor.h*, *PluginEditor.cpp*, *PluginProcessor.h*, *PluginProcessor.cpp*, *Sample.h*, *Sample.cpp*, *SamplesBank.h*, *SamplesBank.cpp* a *Constants.h*.

### 3.3.2 Načtení vzorku

Výhodou JUCE, která usnadní práci při načítání souborů, je výchozí třída *AudioFormatManager*. Ta obsahuje seznam dostupných audio formátů a rozhoduje o jejich způsobech otevření. Souvisí s ní podtřída *AudioFormatReader*, která čte konkrétní typ souboru a načítá jednotlivé vzorky zvukového souboru. Ve vytvořené třídě *Sample* se v části *public* nachází metody *getNextSample()*, *setNoteOn()*, *setNoteOff()*. V části *private* je to hlavně atribut *AudioSampleBuffer*. V konstruktoru třídy *Sample* jsou vypočteny koeficienty vyhlazovacího filtru, který je využit pro postupný útlum předčasně vypnutého vzorku na základě vztahů<sup>5</sup>

```
rampCoefA = exp(-(MATH_2PI) / (RAMP_TIME_S * SAMPLE_RATE));  
rampCoefB = 1.f - rampCoefA;
```

Pomocí třídy *AudioFormatReader* je nastavena velikost bufferu a využitím metody *getWritePointer* jsou získány ukazatele do pole vzorků.

```
sampleBuffer.setSize(reader->numChannels, reader->lengthInSamples);  
reader->read(&sampleBuffer, 0, reader->lengthInSamples, 0, true, true);  
chL = sampleBuffer.getWritePointer(0);  
chR = sampleBuffer.getWritePointer(1);
```

<sup>3</sup> Dostupné z: <http://mda.smartelectronix.com/>

<sup>4</sup> <https://www.juce.com/tutorials>, <https://forum.juce.com/>

<sup>5</sup> <http://musicdsp.org/archive.php?classid=3#257>

### 3.3.3 Spouštění vzorků

Spouštění a vypínání vzorků probíhá ve třídě *Sample* již zmíněnými metodami *setNoteOn* a *setNoteOff*.

```
void Sample::setNoteOn(float gainInDB) {
    playNote = true;
    readIndex = 0;
}
void Sample::setNoteOff() {
    gainTarget = 0.f;
}
```

Pro získání vzorků konkrétního samplu je volána metoda *getNextSample*. Ta zkontroluje, zda je nastavena proměnná *PlayNote* na *true*. V případě, že ano, zapíše požadované vzorky do referencí získaných prostřednictvím parametru metody. Pokud byla v minulosti volána metoda *setNoteOff* (proměnné *gainCurrent* a *gainTarget* se nerovnají), je s každým dalším vzorkem proveden pozvolný přechod mezi tímto a předchozím pomocí vyhlazovacího filtru. V případě, že zesílení je rovno nule, je proměnná *playNote* nastavena na *false* a *readIndex* na 0.

```
if (gainCurrent != gainTarget) {
    gainCurrent = (gainTarget * rampCoefB) + (gainCurrent * rampCoefA);
    if (fabs(gainTarget - gainCurrent) <= RAMP_ERR) gainCurrent = gainTarget;
        if (gainCurrent == 0.f) {
            readIndex = 0;
            playNote = false;
            return;
        }
}
```

Po získání vzorku je *readIndex* inkrementován a pokud je jeho hodnota vyšší než délka samplu, je proměnná *playNote* nastavena na *false* a *readIndex* na 0.

```
readIndex++;
if (readIndex >= sampleBuffer.getNumSamples()) {
    readIndex = 0;
    playNote = false;
}
```

### 3.3.4 Načtení jednotlivých typů paliček – zvukových bank

Načítání vzorků a jejich třídění podle hlasitosti a délky probíhá v třídě *SamplesBank*, která disponuje výčetnými typy *ESticks* (pro určení typu paliček), *ELength* (pro určení délky tónu) a *EVelocity* (pro určení hlasitosti). V závislosti na konkrétní kombinaci parametrů (typ paliček, délka tónu, hlasitost) metody *importSamples* se postupně sestaví celý název souboru .wav z daného adresáře, který se naimportuje do jednoho z šesti polí *OwnedArray*. Níže je úryvek kódu. Z názvu souboru, který je pojmenovaný např. "1\_short\_mf\_12.wav" je potom dáno, že se jedná o měkké paličky, krátký tón a hlasitost mezzoforte. Číslo 12 udává pořadí tónu od začátku rozsahu číslovaného od nuly. Rozsah tónů nástroje je tedy 0 až 57.

```

void SamplesBank::importSamples(ELength length, EVelocity velocity,
OwnedArray<Sample> &samplesArray) { String fileName;
    // sticks type
    if (sticks == eStickSoft) fileName = "1_";
    else if (sticks == eStickHard) fileName = "2_";
    else if (sticks == eStickWood) fileName = "3_";
    else {
        return;
    }
    // sample length
    if (length == eLengthLong) fileName += "long_";
    else if (length == eLengthShort) fileName += "short_";
    else {
        return;
    }
}

```

Po načtení všech 6 typů vzorků, které mohou hrát zároveň (*forteLong*, *mezzoforteLong*, *pianoLong*, *forteShort*, *mezzoforteShort*, *pianoShort*) následuje funkce *getNextSample*, která se volá z *PluginProcessoru* a probíhá v ní součet signálu z aktivních vzorků na výstup. Kvůli tomu, aby signál na výstupu nemusel být omezován limiterem, který by omezil dynamiku nástroje, dochází k celkovému dvojnásobnému zeslabení.

```

void SamplesBank::getNextSample(float &l, float &r) {
    float noteL;
    float noteR;
    for (int i = 0; i < 58; ++i) {
        noteL = 0;
        noteR = 0;
        forteLong.getUnchecked(i)->getNextSample(noteL, noteR);
        l += noteL;
        r += noteR;
        ...
        ...
    }
    // gain reduction
    l *= 0.5f;
    r *= 0.5f;
}

```

Další metodou v třídě *SamplesBank* je *setNoteOn*, která dle MIDI zprávy zapíná danou notu a výpočtem dle hodnoty velocity přepočítá její zeslabení. Pravidlo, dle kterého se zeslabuje, je součástí obrázku 3.3.1. Váhování zesílení dle příslušné hodnoty MIDI velocity obstarává metoda *calcGainInDecibels*, výpočet je zobrazen níže.

```

float SamplesBank::calcGainInDecibels(float in, float inStart, float
inEnd, float outStart, float outEnd)
{
    // check
    if ((inEnd - inStart) == 0.f) return 0.f;

    // regular
    return outStart + ((outEnd - outStart) / (inEnd - inStart)) *
(in - inStart);
}

```

Následuje úryvek již zmíněné metody *setNoteOn*.

```
void SamplesBank::setNoteOn(int note, int8 velocity, bool sustain)
{
    // instrument range in MIDI notes, index Cimbél range (0-57)
    if (note <= 35) return;
    if (note >= 94) return;
    int index = note - 36;

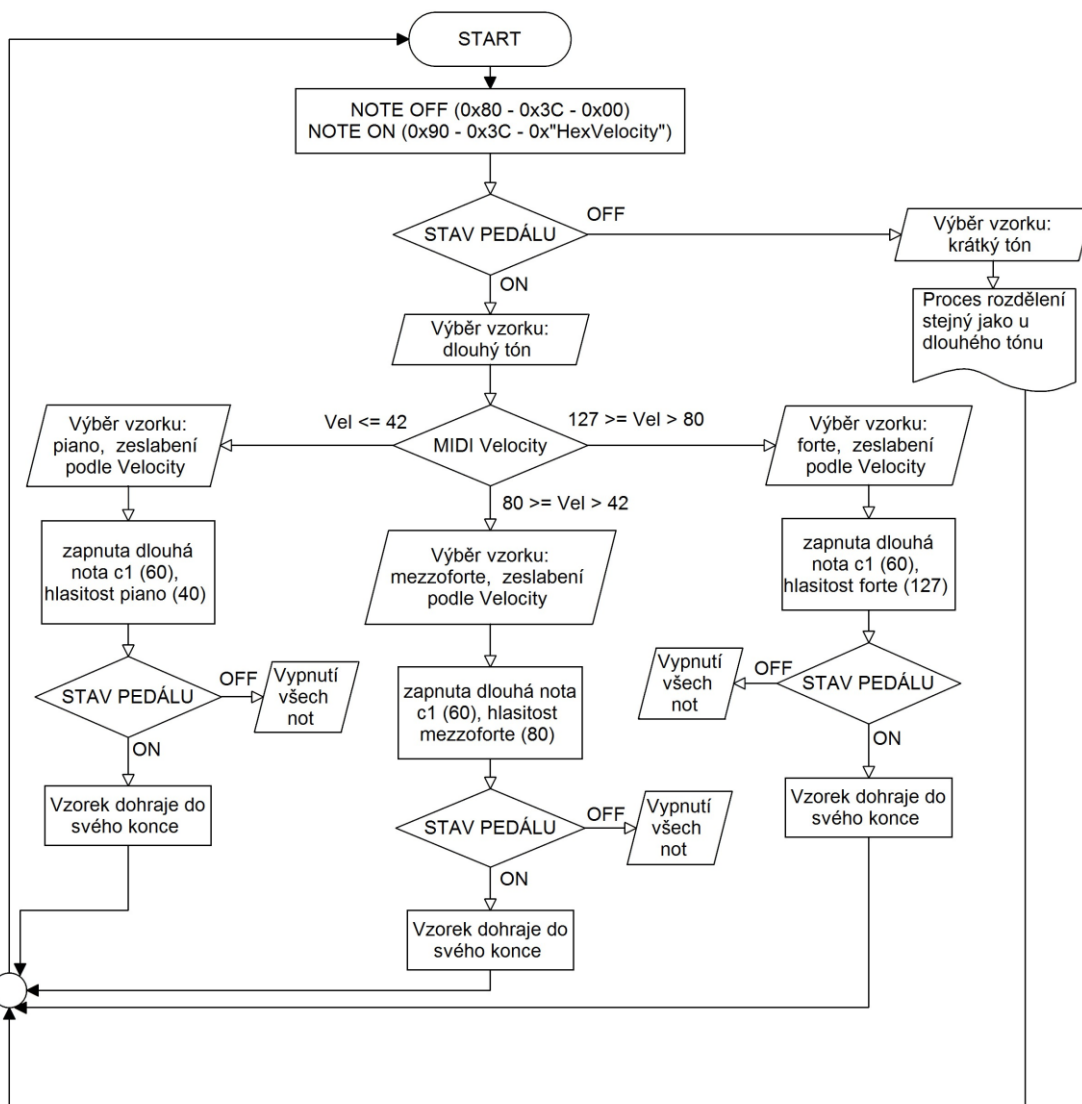
    // pedal on
    if (sustain) {
        if (velocity <= VEL_THRESH_PIANO) {
            if (index >= 0 && index < pianoLong.size()) {
                pianoLong.getUnchecked(index)->setNoteOn(calcGainInDecibels(velocity,
                VEL_PIANO_MIN, VEL_PIANO_MAX, VEL_PIANO_MIN_GAIN,VEL_PIANO_MAX_GAIN));
            }
        }
    }
    ...
}
```

Hodnoty *VEL\_(velocity thresholds)* jsou definovány v hlavičkovém souboru *Constants.h*. Obdobně kód vypadá pro vypnutý pedál a pro vypnutou notu bez nutnosti váhovaného zeslabování.

### 3.3.5 Vývojový diagram

Na obrázku 3.3.1 je zjednodušený a částečný popis posloupnosti událostí MIDI převodníku a funkcí programu při jedné zvolené bance. Vysílání MIDI zpráv Note On a Note Off zajišťuje MIDI převodník. Nota c' (c1) s hlasitostmi nevyžadujícími změnu zesílení je zde uvedena pouze jako příklad.

Program postupně provádí výběr vzorku na základě aktuálního stavu pedálu (Control Change 64 – Sustain) a na základě přijaté hodnotě Velocity. Po vyhodnocení stavu pedálu a hlasitosti se vybere konkrétní vzorek. Při opakovaném úderu musí nejprve přijít příslušná zpráva Note Off (s hlasitostí 0) a ihned poté zpráva Note On na stejnou notu s již příslušnou odpovídající hlasitostí (velocity) zahrané noty. Tento způsob byl navrhnout kvůli způsobu hry na cimbál a speciálně kvůli již zmíněné herní technice *tremolo*. Aby se tedy nezahltil výstup MIDI zprávami Note On, je zapotřebí tohoto řešení, které zajistí vypnutí příslušné noty (tónu) před jejím opakovaným spuštěním.



Obr. 3.3.1: Zjednodušený popis posloupnosti událostí programu a MIDI převodníku

### 3.3.6 Uživatelské rozhraní

Jak již bylo zmíněno výše, grafické uživatelské rozhraní je řešeno v části *PluginProcessorEditor*. V případě tohoto pluginu se v editoru nezjišťují žádné parametry, ale jsou zde pouze tři obrázková tlačítka (*ImageButton*), která umožňují výběr paliček (zvukové banky) kliknutím na ně. Dále je zde výpisové pole (*Label*), které zobrazuje zvolený typ paliček.

Při stisku obrázkového tlačítka je volána metoda *buttonClicked* třídy *CimbelJuceAudioProcessor*. Poté je na základě hodnoty jejího parametru s názvem *buttonThatWasClicked* možné zjistit, které ze tří tlačítek bylo stisknuto. Níže je pro ilustraci úryvek kódu, který popisuje, co se má provést při stisku jednoho z tlačítek.



```

void CimbelJuceAudioProcessorEditor::buttonClicked (Button*
buttonThatWasClicked)
{
    if (buttonThatWasClicked == imageButton)
    {
        processor.BankSelect(1);
        label->setText(CharPointer_UTF16(L"Vybráno: MĚKKÉ
        PALIČKY"), (dontSendNotification));
        label->setFont(Font("Book Antiqua", 20.00f, Font::bold));
        label->setColour(Label::textColourId, Colours::greenyellow);
    }
}

```

Po rozpoznání, o které tlačítko se jedná, je volána metoda procesoru *BankSelect*, ve které je nejdříve pozastaveno přehrávání (metoda *suspendProcessing*) a uvolněna stávající banka samplů a následně načtena nově zvolená banka a opět spuštěno přehrávání.

```

void CimbelJuceAudioProcessor::BankSelect(int TypPalicek)
{
    if (TypPalicek == 1)
    {
        suspendProcessing(true);
        samplesBank = NULL;
        samplesBank = new SamplesBank
        (SamplesBank::ESticks::eStickSoft, samplesDirectory);
        suspendProcessing(false);
    }
}

```

Na obrázku 3.3.2 je grafické rozhraní ve výchozím stavu při zapnutí pluginu (načtená banka měkkých paliček) a při zvoleném typu „neobalené paličky“.



Obr. 3.3.2: Grafické uživatelské rozhraní pluginu

### 3.3.7 Verze „HW“

Kvůli kompatibilitě a reprezentativnějšímu chování dle akustické předlohy nástroje byla vytvořena druhá verze pluginu (první nese označení „Keys“). Důvody tohoto vytvoření jsou uvedeny v podkapitole 4.2.2.

Tato verze sebou nese nevýhodu používání na klasických MIDI klávesách. Kvůli tomu, že se v této verzi při hraní dlouhých not (s pedálem) zapnutá nota nevypne při příchozí zprávě Note Off, ale až při vypnutí pedálu, dojde ke skokové změně signálu na výstupu a tudíž k lupnutí. Při velké dynamice je toto lupnutí těžko slyšitelné, ale při mírné dynamice registrovatelné je. Tato verze by však měla sloužit pro cimbálovou hardware část, kde se před každou vyslanou Note On zprávou vyšle Note Off zpráva, což by mělo tento problém odstranit.

Jednou z nejpodstatnějších metod *AudioProcessoru* je metoda *processBlock*, která dostává ukazatel na klíčové objekty tříd *AudioBuffer* a *MidiBuffer* obsahující audio data a MIDI zprávy.

```
void CimbelJuceAudioProcessor::processBlock
(AudioSampleBuffer& buffer, MidiBuffer& midiMessages)
{
    MidiBuffer::Iterator midiIterator(midiMessages);
    MidiMessage msg(0xf0);
    int samplePosition;
    while (midiIterator.getNextEvent(msg, samplePosition)) {
```

Uvnitř této metody se nachází i jedna z diferencí verzí „Keys“ a „HW“. Verze „HW“ poté nereaguje na přijaté zprávy NoteOff.

```
        if (msg.isController() == true)
        {
            if (msg.isSustainPedalOn() == 1) sustain = true;
            else
            {
                sustain = false;
                samplesBank->setAllNotesOff();
            }
        }
    else {
        if (msg.isNoteOn())
            samplesBank->setNoteOn(msg.getNoteNumber(),
            msg.getVelocity(), sustain);
        /*
        if (msg.isNoteOff())
            samplesBank->setNoteOff(msg.getNoteNumber());
        */
    }
}
```

V třídě *samplesBank* ve verzi „HW“ přibývá metoda *setAllNotesOff*, která vypíná všechny aktivované noty a je volaná z *Processoru* při vypnutí pedálu, tj. při přijaté MIDI zprávě typu Control Change (64), hodnota 0.

```
void SamplesBank::setAllNotesOff()
{
    for (int i = 0; i < 58; ++i)
        setNoteOff(i+36); } }
```

## 4 SHRNUTÍ VÝVOJE PRÁCE

V této kapitole jsou shrnuty komplikace při nahrávání a jejich řešení. Dále tato část obsahuje poznatky z testování.

### 4.1 Komplikace při nahrávání

Při nahrávání se objevily určité obtíže a pro úplnost práce je třeba zmínit jejich operativní řešení.

Jedním z problémů, které se týkaly nahrávaného nástroje, je absence tlumícího mechanismu u tónů dis2, g2, a2, b2, h2 a tónů c3 až a3. V dnešní době se sice již vyrábí nástroje, které mají tlumení na všech těchto tónech, ale jelikož jsou to tóny patřící do horní části rozsahu, struny jsou již poměrně krátké a hodně napnuté, výsledkem je pak velmi krátce znějící tón. Podobně to mají např. určité druhy pianin. Kvůli dosažení homogenity krátkých tónů se však i tyto krátké tóny krátkých strun musely utlumit. Efektivním řešením tohoto problému bylo tlumení položením druhé paličky na strunu. Některé tóny mají tak krátké struny, že je při jejich tlumení značně potlačený fundament a tón lze špatně rozeznat. Při úderech na vyšší struny se ostatní struny rozeznávaly až příliš a interpret musel držet několik okolních problémových strun, aby byl tón dobře rozpoznatelný. Jelikož jsou struny jednoho tónu v těsné blízkosti, při velké dynamice se stávalo, že se při úderu vychylují tak, až mezi nimi dojde ke kontaktu, což vedlo k nežádoucímu zvuku, podobnému přebuzení. Tento problém se týká velmi málo tónů, a proto se tím práce nezabývá. Nástroje se ještě týká omezená dynamika u neobalených paliček. Ty mají sice bohatý a průrazný zvuk, ale jejich dynamické možnosti jsou velmi omezené, aby nedošlo k jejich poškození.

Další problém se týkal bezodrazové komory. Její odhlučnění není dokonalé na všech frekvencích a ze všech stran. Často se stávalo, že pronikal okolní hluk z chodby a podlaží nad komorou. Většinou se ho však podařilo zaregistrovat a pozastavit nahrávání, v ostatních případech se dal omezit filtrem typu horní propust s mezním kmitočtem v rozmezí 70 - 200 Hz, dle výšky tónu.

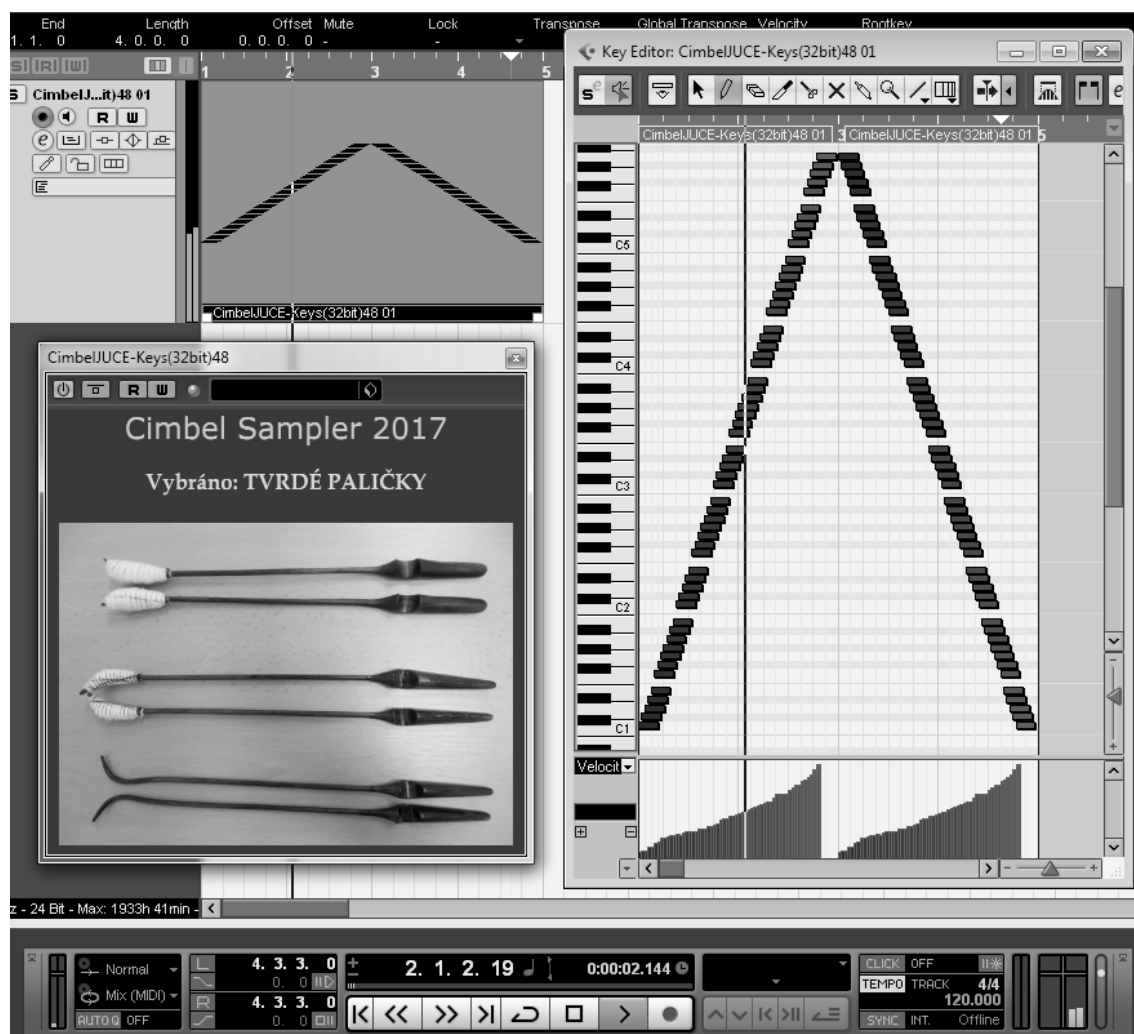
### 4.2 Testování

Nezbytným krokem při vývoji pluginu je testování chování v různých host aplikacích včetně DAW systémů.

#### 4.2.1 Samostatné testování pluginu v host aplikacích

Obě verze (32bit, 64bit) dvou verzí pluginů (Keys, HW) byly testovány v příslušných verzích DAW systémů Cubase 5, Cubase 7.5, StudioOne 3, AbletonLive 9. K testování bylo převážně používáno MIDI zařízení AKAI LPK 25, které má i přes svou velikost dostatečné možnosti pro testování. Na obrázku 4.2.1 je zachyceno testování z DAW Cubase 5 (32bit). Zde a v ostatních DAW systémech probíhalo testování

také pomocí přednastavených MIDI zpráv s postupnou změnou dynamiky a postupnou změnou rozsahu.



Obr. 4.2.1: Testování pluginu v DAW systému Cubase 5

Ve všech aplikacích Plugin pracuje správně, reaguje na všechny zprávy, včetně pedálu tak, jak má, nepadá a nezasekává se. Nevýhodou je pomalé první načítání banky po startu počítače. Pokud ale každá banka byla alespoň již jednou načtena stejnou verzí pluginu, je rychlost načítání bank závislá na frekvenci RAM a procesoru počítače.

#### 4.2.2 Testování s hardwarovou částí

Při testování ještě nedokončené a nedoladěné verze hardwarové části bylo zjištěno, že kvůli nestandardnosti MIDI hardware zařízení bude patrně dost problému se vzájemnou kompatibilitou. Elektronickému cimbálu nefungoval správně pedál a celý koncept se jevil jako omezený a nestabilní. Nicméně i přesto, že je projekt rozdělen do tří individuálních částí, které nejsou hodnoceny za společnou funkčnost, byla vytvořena druhá verze programu, která lépe reprezentuje vlastnosti akustického nástroje a měla by být kompatibilnější s MIDI převodníkem.

## 5 ZÁVĚR

Výsledkem této bakalářské práce je funkční VSTi plugin sampler se zvuky cimbálu. Vzorky byly nahrány v bezodrazové komoře na FEKT T12 na nástroj značky Jiří Galuška, Ostrožská Nová Ves, vyrobený roku 2001 a naladěný na  $a^{\flat} = 442$  Hz. Plugin, který je označen CimbelJUCE, má sloužit jako výstupní část elektronického cimbálu. Tento zásuvný modul je poskytnut ve dvou hlavních verzích: pro hardwarovou část elektronického cimbálu a pro MIDI nástroj klávesového typu („HW“ a „Keys“).

Za cíl, kterého nebylo dosaženo, lze částečně považovat neodladění chyb výsledného konceptu elektronický cimbál. U toho však nelze jednoznačně říci, v které části se chyba nachází a nelze tudíž koncept hodnotit jako celek. Problémem je hlavně to, že hardwarová část není standardizované MIDI zařízení, tudíž zatím nelze zaručit bezchybnost provozu a minimalizaci chyb. Důvodem, proč je k dispozici celkem osm verzí výsledného programu, je nepřizpůsobivost programu vzorkovací frekvenci host aplikace. Plugin má tedy vzorky a verze pro vzorkovací frekvence 44,1 kHz a 48 kHz, pro 32bitové a 64bitové host aplikace a dva již zmíněné hlavní typy „Keys“ a „HW“. Ty jsou blíže popsány v příloze A. Důvodem pro rozdělení na verze 44,1 kHz a 48 kHz je výkonové přizpůsobení, které však díky tomu, že se do paměti RAM načítá necelý dvojnásobek velikosti vzorků, nemá příliš velký význam. Výsledné rozdíly využití paměti RAM obou verzí (44,1 kHz a 48 kHz) jsou pak pouze cca 150 MB, někdy i méně.

Mezi návrhy pro zlepšení lze například zařadit již zmíněnou adaptabilitnost pluginu na libovolný vzorkovací kmitočet hostitelské aplikace. Profesionální samplery obsahují také například více než 3 hlasitostní hladiny, aby se lépe pokryly dynamické možnosti nástrojů, zmenšila rozdílnost dynamických hladin mezi sebou a tím se zjemnily přechody těchto hladin. Do vylepšení by bylo dále vhodné umožnit uživateli volbu některých parametrů, například možnost smíchání vlastního poměru z nahrávaných mikrofonů. To by ovšem zvětšilo nároky na RAM a vyžadovalo i sofistikovanější způsob ukládání samplů do speciálních formátů. O několik úrovní výše, co se obtížnosti týče, by byla dynamická modulace signálu, poměrů harmonických složek atd. na základě způsobu a místa úderu do struny.

# LITERATURA

- [1] FORRÓ, Daniel. *Musitronika: elektroakustické hudební nástroje, část IV. Samplery*. Brno: Janáčkova akademie múzických umění, 2001. ISBN 80-85429-57-8.
- [2] GEIST, Bohumil. *Akustika: jevy a souvislosti v hudební teorii a praxi*. Praha: Muzikus, 2005. ISBN 80-86253-31-7.
- [3] GUÉRIN, Robert. *Velká kniha MIDI: standardy, hardware, software*. Brno: Computer Press, 2004. ISBN 80-7226-985-2.
- [4] HOLCOVÁ, K. *Konstrukční vývoj cimbálu na Moravě*. Olomouc: Univerzita Palackého v Olomouci, Filozofická fakulta. Katedra muzikologie, 2014. 71 s., 8 s. příloh. Bakalářská práce. Vedoucí práce: doc. PhDr. Jiří Kopecký, Ph.D.
- [5] KÁČER, J. *Cimbál a cimbalisté na Uherskohradištsku*. Brno: Masarykova univerzita, Filozofická fakulta. Ústav evropské etnologie, 2011. 153 s., 4 s. příloh. Diplomová práce. Vedoucí práce: doc. PhDr. Martina Pavlicová, CSc.
- [6] Sampler (musical instrument). *Wikipedia*. [online]. 7.12.2016 [cit. 2016-12-9]. Dostupné z: [https://en.wikipedia.org/wiki/Sampler\\_\(musical\\_instrument\)](https://en.wikipedia.org/wiki/Sampler_(musical_instrument))
- [7] SCHIMMEL, J. *Studiová a hudební elektronika*. Vyd. 2. V Brně: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 195 s. ISBN 978-80-214-4452-2.
- [8] SYROVÝ, Václav. *Hudební akustika*. 2., dopl. vyd. V Praze: Akademie múzických umění, 2008. Akustická knihovna Zvukového studia Hudební fakulty AMU. ISBN 978-80-7331-127-8.
- [9] TEOCHARISOVÁ, Vanda. *Sound design: zvuková syntéza a tvůrčí programování zvuků v praxi*. Praha: Muzikus, c2009. ISBN 978-80-86253-53-4.
- [10] VLACHÝ, Václav. *Praxe zvukové techniky*. 3., aktualiz. a dopl. vyd. Praha: Muzikus, c2008. ISBN 978-80-86253-46-6.
- [11] What are Software Samplers?. *Tweakheadz*. [online]. 2010 [cit. 2016-11-09]. Dostupné z: <http://tweakheadz.com/software-samplers/>
- [12] What is the difference between Synthesizers, Samplers, and Sequencers? . *Music and Light unlimited*. [online]. 2002 [cit. 2016-11-15]. Dostupné z: [http://musicandlight-unlimited.com/audio/what\\_is\\_synth\\_samp\\_seq](http://musicandlight-unlimited.com/audio/what_is_synth_samp_seq).

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

$f_{vz}$	Vzorkovací kmitočet
$K_{10}$	Počet kvantovacích hladin v desítkové soustavě
$Q$	Kvantovací krok převodníku
$SNR$	Odstup signálu od (kvantovacího) šumu
ASIO	Audio Streaming Input Output, ovladač toku zvuku na vstupu a výstupu zařízení
DAW	Digital Audio Workstation, digitální hudební pracovní stanice, tj. software pro nahrávání, tvorbu a úpravu zvuku
DIN	Deutsches Institut für Normung, již neplatná norma a označení konektorů
FFT	Fast Fourier Transform, rychlá Fourierova transformace
GM	General MIDI specifikace
GUI	Graphic User Interface, grafické uživatelské rozhraní (např. zásuvného modulu)
JUCE	Jules' Utility Class Extensions, multiplatformní framework pro zjednodušenou tvorbu např. audio pluginů a aplikací
LSB	Least Significant Bit, nejméně významný bit
MIDI	Musical Instrument Digital Interface, digitální rozhraní pro hudební nástroje
MSB	Most Significant Bit, nejvýznamnější bit
PCM	Pulse Code Modulation, pulzně kódová modulace
RAM	Random Access Memory, volatilní operační paměť
SDK	Software Development Kit, balíček ke zjednodušení vývoje softwaru
USB	Universal Serial Bus, univerzální sériová sběrnice
VST	Virtual Studio Technology, technologie zásuvných modulů pro využití hudebními programy virtuálního hudebního studia
VSTi	zásuvný modul, který generuje zvuk

# SEZNAM PŘÍLOH

**A** Obsah přiloženého disku DVD

**B** Návod k obsluze



# A OBSAH PŘILOŽENÉHO DISKU DVD

## PLUGIN.rar

### Složka 44@16

**Složka CimbelPluginSamples44k** – složka obsahující všechny vzorky rozsahu nástroje v kvalitě 44,1 kHz a 16 bitů, sloužící jako zdrojová data pro verze 44

**CimbelJUICE-HW(32bit)44.dll** – výsledný plugin ve verzi pro hardware, 32bitovou host aplikaci a vzorky v kvalitě 44,1 kHz a 16 bitů

**CimbelJUICE-HW(64bit)44.dll** – výsledný plugin ve verzi pro hardware, 64bitovou host aplikaci a vzorky v kvalitě 44,1 kHz a 16 bitů

**CimbelJUICE-Keys(32bit)44.dll** – výsledný plugin ve verzi pro klávesy, 32bitovou host aplikaci a vzorky v kvalitě 44,1 kHz a 16 bitů

**CimbelJUICE-Keys(64bit)44.dll** – výsledný plugin ve verzi pro klávesy, 64bitovou host aplikaci a vzorky v kvalitě 44,1 kHz a 16 bitů

### Složka 48@24

**Složka CimbelPluginSamples48k** – složka obsahující všechny vzorky rozsahu nástroje v kvalitě 48 kHz a 24 bitů, sloužící jako zdrojová data pro verze 48

**CimbelJUICE-HW(32bit)48.dll** – výsledný plugin ve verzi pro hardware, 32bitovou host aplikaci a vzorky v kvalitě 48 kHz a 24 bitů

**CimbelJUICE-HW(64bit)48.dll** – výsledný plugin ve verzi pro hardware, 64bitovou host aplikaci a vzorky v kvalitě 48 kHz a 24 bitů

**CimbelJUICE-Keys(32bit)48.dll** – výsledný plugin ve verzi pro klávesy, 32bitovou host aplikaci a vzorky v kvalitě 48 kHz a 24 bitů

**CimbelJUICE-Keys(64bit)48.dll** – výsledný plugin ve verzi pro klávesy, 64bitovou host aplikaci a vzorky v kvalitě 48 kHz a 24 bitů

## CODE.rar

**Složka JUICE** – balíček knihoven, také stažitelný z webu JUICE<sup>6</sup>

**Složka CimbelJUICE** – složka s projektem v prostředí Microsoft Visual Studio Community 2015 (Pro spuštění projektu je nutné otevřít *Solution* projektu CimbelJUICE\Builds\VisualStudio2015\CimbelJUICE.sln)

**Složka VST\_SDK** – složka s vývojářským balíčkem<sup>7</sup> od firmy Steinberg

**Jan\_Bican\_Bakalarska\_prace\_2017.pdf** – elektronická verze textové části práce

---

<sup>6</sup> Dostupné z: <https://www.juce.com/>

<sup>7</sup> Dostupné z: <https://www.steinberg.net/en/company/developers.html>

## B NÁVOD K OBSLUZE

### 1. Zkopírovat složku se vzorky do adresáře Dokumenty

Je zapotřebí si zvolit alespoň jednu složku se vzorky (CimbelPluginSamples44k, CimbelPluginSamples48k) a nakopírovat do adresáře Dokumenty.

### 2. Zkopírovat příslušnou dynamickou knihovnu do adresáře s VST pluginy

Podle typu hostitelské aplikace (32bit, 64bit) a zvolené složky (44,48) si vyberte příslušnou dynamickou knihovnu (soubor .dll) a zkopírujte ho do adresáře, z kterého Vaše hostitelská aplikace načítá VST pluginy.

*(Příklad: Cubase 5(32bit) se vzorky v kvalitě 48 kHz/24 bitů, verze pro hardware: zvolit složku se vzorky: CimbelPluginSamples48k, zvolit dynamickou knihovnu: CimbelJUCE-HW(32bit)48.dll)*

### 3. Otevřít hostitelskou aplikaci a nastavit příslušnou vzorkovací frekvenci

Před otevřením je zapotřebí mít zapojené a zapnuté MIDI zařízení, pokud je v plánu nějaké používat. Po otevření hostitelské aplikace je zapotřebí nastavit vzorkovací frekvenci (Sample Rate) na hodnotu zvolenou v předchozím bodě. Při použití verzí 44 (44k) je nutné ji nastavit na 44100 Hz a při použití verzí 48 (48k) na hodnotu 48000 Hz. Případně nastavit také MIDI zařízení.

### 4. Vyčkat při prvním načítání

Při prvním načítání pluginu se může stát, že se budou vzorky načítat déle a nezávisle na výkonu počítače. Platí to i pro první načítání každé ze 3 bank. Pokud byla už každá banka alespoň jednou načtena stejnou verzí pluginu, je rychlost načítání bank závislá na frekvenci paměti RAM a procesoru počítače. Je také vhodné sledovat vytížení RAM např. ve Správci úloh.