



## **Středoškolská technika 2012**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **Využití MIDI komunikace při výuce hudby**

**Lukáš Lokajíček**

Vyšší odborná škola a Střední průmyslová škola, Šumperk,  
Gen. Krátkého 1, Šumperk

## ANOTACE

Tato práce se zabývá využitím MIDI komunikace při výuce hudební teorie a zdokonalování hudebního relativního sluchu. Výsledek tohoto projektu je zařízení, které díky MIDI signálům posílaných z MIDI zařízení (např. Digitální klavír) vyhodnocuje komunikační kanál, stisknutou klávesu, zahrany interval, akord a stupnice. Další funkcí je posílání MIDI zpráv, díky kterým klavír s MIDI vstupem přehrává intervaly, akordy a stupnice. Srdcem tohoto výrobku je mikropočítač Atmega8 a jako zobrazovací jednotka je použitý LCD displej.

Klíčová slova: MIDI; Atmega8; Klavír; Hudba; Hudební sluch; Hudební teorie

## ANNOTATION

This project deals with MIDI communication and its use in teaching music theory and improvement of relative musical ear. The result of this project is a device which receives MIDI signals from a MIDI device (e.g. digital piano) and analyzes the communication channel, pressed key, played interval, chord and scale. Another function is transmitting MIDI messages to piano with a MIDI input, which plays intervals, chords and scales. The heart of this product is a microcomputer Atmega8. LCD display is used as a display unit.

Key words: MIDI; Atmega8, Piano, Music, Musical ear, Musical theory

## Obsah

<b>1 ÚVOD</b> .....	<b>4</b>
<b>2 MIDI KOMUNIKACE A HUDEBNÍ ZAŘÍZENÍ</b> .....	<b>5</b>
<b>2.1 PRINCIP MIDI</b> .....	<b>5</b>
<b>2.2 MIDI PROTOKOL</b> .....	<b>5</b>
2.2.1 Struktura protokolu .....	5
2.2.1.1 Running Status .....	6
2.2.1.2 MIDI zprávy Nota zapnuta/vypnuta .....	6
2.2.1.3 MIDI zpráva All Note Off .....	7
2.2.2 Způsob posílání dat .....	7
<b>3 VYUŽITÍ MIDI KOMUNIKACE PŘI VÝUCE HUDBY</b> .....	<b>9</b>
<b>3.1 JAK VYPADÁ HUDEBNÍ ANALYZÁTOR</b> .....	<b>9</b>
<b>3.2 PŘIPOJENÍ HUDEBNÍHO ANALYZÁTORU</b> .....	<b>10</b>
<b>3.3 PRVNÍ SPUŠTĚNÍ</b> .....	<b>11</b>
<b>3.4 OVLÁDÁNÍ HUDEBNÍHO ANALYZÁTORU</b> .....	<b>12</b>
<b>3.5 KONFIGURACE</b> .....	<b>12</b>
<b>3.6 JEDNOTLIVÉ FUNKCE HUDEBNÍHO ANALYZÁTORU</b> .....	<b>13</b>
3.6.1 První blok „MIDI Analýza“ .....	13
3.6.1.1 Zjištění kanálu.....	13
3.6.1.2 Stisknutá klávesa.....	14
3.6.1.3 Intervaly .....	15
3.6.1.4 Akordy .....	16
3.6.1.5 Stupnice.....	18
3.6.2 Druhý blok „Trénink sluchu“ .....	19
3.6.2.1 Intervaly .....	19
3.6.2.2 Akordy .....	20
3.6.2.3 Stupnice.....	21
<b>4 KONSTRUKČNÍ DOKUMENTACE</b> .....	<b>22</b>
<b>4.1 PRINCIP ČINNOSTI</b> .....	<b>22</b>
<b>4.2 SCHÉMA HUDEBNÍHO ANALYZÁTORU</b> .....	<b>23</b>
<b>4.3 KLIŠÉ HUDEBNÍHO ANALYZÁTORU</b> .....	<b>24</b>
<b>4.4 OSAZOVACÍ SCHÉMA</b> .....	<b>25</b>
<b>4.5 REALIZACE HUDEBNÍHO ANALYZÁTORU</b> .....	<b>26</b>

<b>4.6</b>	<b>ROZPISKA SOUČÁSTEK.....</b>	<b>27</b>
<b>5</b>	<b>OBSLUŽNÝ SOFTWARE .....</b>	<b>28</b>
<b>5.1</b>	<b>UART KOMUNIKACE.....</b>	<b>28</b>
<b>5.2</b>	<b>ZPRACOVÁNÍ PŘIJATÝCH DAT.....</b>	<b>29</b>
5.2.1	Zjištění čísla MIDI kanálu.....	30
5.2.2	Stisknutá klávesa .....	31
5.2.3	Analýza intervalů .....	32
5.2.4	Analýza Akordů .....	34
5.2.5	Analýza stupnic .....	37
5.2.6	Zjištění konfigurace.....	38
5.2.7	Přehrávání intervalů .....	39
5.2.8	Přehrávání akordů .....	40
5.2.9	Přehrávání stupnic .....	41
<b>5.3</b>	<b>VÝVOJOVÝ DIAGRAM PROGRAMU.....</b>	<b>42</b>
<b>6</b>	<b>RESUMÉ .....</b>	<b>43</b>
<b>6.1</b>	<b>V ČEKÉM JAZYCE.....</b>	<b>43</b>
<b>6.2</b>	<b>V ANGLICKÉM JAZYCE.....</b>	<b>43</b>

## 1 ÚVOD

Myšlenkou tohoto projektu je výuka hudební teorie zábavnou a praktickou metodou. Postupně co se budete učit, si budete moct vyzkoušet na klavíru a hudební analyzátor vás bude přitom provázet a trénovat. Nejlepší na tom je, že co se naučíte, si hned můžete vyzkoušet, aniž byste se báli toho, že děláte něco špatně, nebo se něco špatně učíte.

Výuka s hudebním analyzátozem už nebude jen sezení u učebnice a vyplňovat cvičení, které vám nic neříká, nebo nechápete jeho smysl. U výuky rovnou uslyšíte znění jednotlivých tónů, intervalů, akordů a stupnic.

Poslední významnou schopností hudebního analyzátoru je možnost si prohození rolí s ním. Už mu nebudete hrát intervaly (akordy, stupnice) aby je vám analyzoval, teď už to bude tak, že hudební analyzátor vám něco zahraje, a analyzovat to budete vy, na něj už jen zbude vás utvrdit ve vašem tvrzení, nebo ho vyvrátit. Někteří už jistě pochopili význam tohoto tréninku, a kdo ne, tak pro ujasnění se jedná o trénink relativního hudebního sluchu.

Všechny funkce jsem popsal v dokumentaci a uvedl jsem k nim jeden příklad. V příloze je pracovní sešit, který jsem vytvořil k hudebnímu analyzátoru, kde můžeme najít více způsobů použití.

Tato dokumentace počítá se znalostí hudební teorie. V první příloze je hudební teorie (nauka), kterou jsem zpracoval pro ty, kteří chtějí lépe pochopit funkci a užitečnost toho hudební analyzátoru. Hudební nauka je popisována od základu, až k složitějším záležitostem jako jsou například septakordů

## 2 MIDI KOMUNIKACE A HUDEBNÍ ZAŘÍZENÍ

### 2.1 PRINCIP MIDI

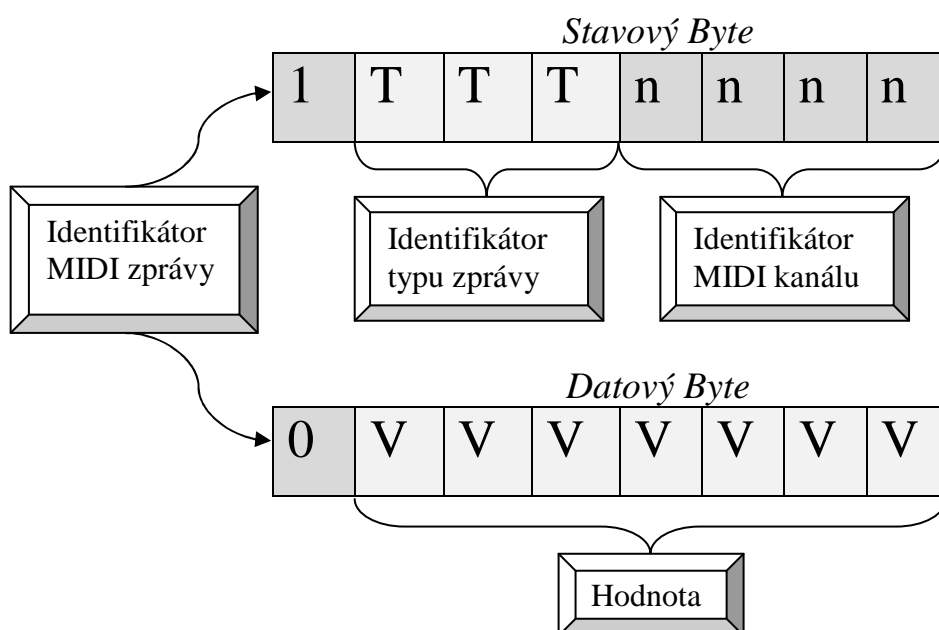
MIDI (**M**usical **I**nstruments **D**igital **I**nterface) je standardizovaná komunikace mezi elektronickými zařízeními. První myšlenkou bylo využití toho MIDI protokolu pouze na elektronické hudební nástroje, v dnešní době se využívá k řízení kompletní techniky využívané hudebníky (např. studiová a jevištní technika). Jelikož se bavíme o hudbě, tak by se mohlo zdát, že MIDI pracuje s audio signálem, ale není tomu tak. V MIDI zprávách jsou uloženy jednotlivé úkony v digitální podobě, které se provedly při hraní na elektronický hudební nástroj.

### 2.2 MIDI PROTOKOL

#### 2.2.1 Struktura protokolu

Základní datový blok přenášející určitou informaci se nazývá "MIDI zpráva" (MIDI message). Ta se skládá z jednoho stavového a několika datových MIDI bytů. Těmto bytům se také říká "MIDI událost" (MIDI event). MIDI Byte je osmibitový datový typ, kde nejvýznamnější bit určuje, zda jde o stavový byte (MSB=1) nebo datový byte (MSB=0). Na Obr. 1 je struktura stavového a datového bytu.

MIDI zprávy se dělí na tzv. kanálová data a systémová data. Kanálová data přenášejí ve stavovém eventu informaci o virtuálním datovém kanále. Ve stavovém eventu jsou pro identifikaci MIDI kanálu vyhrazeny čtyři bity (Obr. 1), proto mohou být po jedné fyzické MIDI sběrnici přenášena kanálová data až v 16 virtuálních kanálech. Systémová data informaci o MIDI kanále nepřenášejí => jsou společná pro všechny kanály. Dolní čtyři bity stavového eventu slouží k identifikaci typu systémových dat. Systémová data se dělí na zvláštní systémová data, která umožňují přenos větších datových bloků, a na data reálného času, která slouží k vzájemné časové synchronizaci několika zařízení.



Obr. 1 Struktura stavového a datového bytu protokolu MIDI

Počet datových bytů závisí na typu MIDI zprávy, tj. na stavovém bytu. Například zpráva "Nota zapnuta" přenáší číslo noty (tón) a sílu stlačení klávesy. Nejvýznamnější bit v datovém bytu je vyhrazen pro identifikátor, proto mohou být hodnoty přenášené pomocí datových bytů v rozsahu 0 až 127.

### 2.2.1.1 Running Status

Aby se zvětšila průchodnost MIDI sběrnice, není nutné v případě kanálových dat vždy vysílat úplné MIDI zprávy. Pokud se nemění typ informace určený stavovým bytem, stačí vysílat jenom datové byty. Za kompletní zprávu je považován pouze odpovídající počet datových bytů. Tím se ušetří asi třetina kapacity MIDI sběrnice. Tento stav se nazývá *Running Status* (průběžný stav). Tento stav platí až do té doby, než je přijat odlišný stavový byte.

### 2.2.1.2 MIDI zprávy Nota zapnuta/vypnuta

#### 2.2.1.2.1 Note On

MIDI zpráva *Note On* (nota zapnuta) má identifikátor 1 (MSB = 1) a přenáší informace o stisknuté klávese, tj. o zahrané notě. První datový byte přenáší informaci o MIDI čísle noty a druhý datový byte informaci o dynamice, s jakou byla nota zahrána (viz Obr. 2 Obr. 2).

Stavový Byte	1. Datový Byte	2. Datový Byte
1001nnnn	0kkkkkkk	0vvvvvvv
\$9n (n = číslo kanálu)	Číslo noty	Rychlostní data

Obr. 2 Struktura MIDI zprávy *Note On*

Pro čísla MIDI not je možno použít sedm bitů, tj. hodnoty od 0 do 127. V MIDI jsou noty očíslovány od 2C do G8 (viz Tab. 2). Rozsah klaviatury klavíru je od A0 do C7, čili podle Tab. 2 MIDI čísla not 21 až 108.

Druhý datový byte zprávy *Note On* určuje dynamiku, s jakou byla nota zahrána. Tuto informaci lze využít pro nastavení intenzity zahraneho tónu nebo pro nastavení jiného parametru měnícího barvu zvuku. Vytvořit klaviaturu s dynamickými snímači není jednoduché. Naštěstí je dynamika přímo úměrná rychlosti stlačení klávesy, takže stačí snímat dobu mezi sepnutím dvou spínačů postupně aktivovaných při stlačení klávesy. Klaviaturou jsou tedy snímána a ve zprávě *Note On* přenášena rychlostní data, tzv. *Velocity*. Někdy může být prezentování zprávy *Note Off*, jako *Note On* s *velocity* = 0. Vztah rychlostních dat a dynamiky tónu ukazuje Tab. 1.

Velocity	Dynamika	Hudební označení (Italské)
0	OFF	<i>Note Off</i>
1	ppp	Pianissimo piano
.	pp	Pianissimo
.	pp	piano
64	mp	Mezzo-piano
	mf	Mezzo-forte
.	f	Forte
.	ff	Fortissimo
127	fff	Fortissimo forte

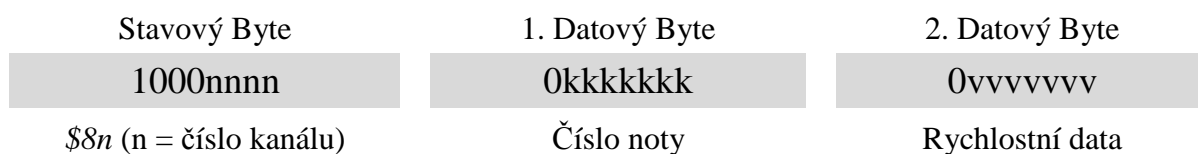
Tab. 1 MIDI rychlostní data

Oktáva	Hudební označení	C	Cis	D	Dis	E	F	Fis	G	Gis	A	Ais	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. 2: MIDI označení not

#### 2.2.1.2.2 Note Off

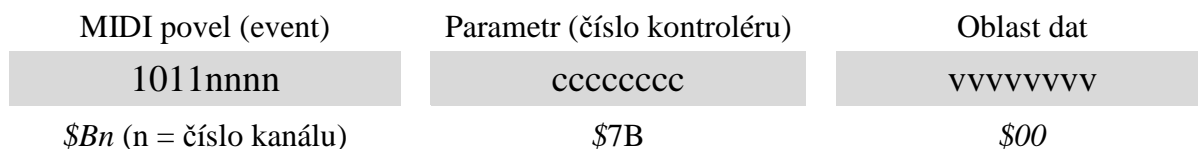
MIDI zpráva *Note Off* (nota vypnuta) má identifikátor 0 a přenáší informace o uvolněné klávese. První datový byte přenáší informaci o MIDI čísle noty a druhý datový byte informaci o rychlosti, s jakou byla klávesa uvolněna (Obr. 3). To, že je přenášena informace o rychlosti uvolnění klávesy, umožňuje napodobovat některé techniky hraní např. na strunné nástroje, ve skutečnosti se ale tato informace většinou nepoužívá.



Obr. 3 Struktura MIDI zprávy *Note Off*

#### 2.2.1.3 MIDI zpráva All Note Off

Samozřejmě, že tímto nekončí výčet všech MIDI zpráv (těch je kolem 35), ale uvádím zde jen ty MIDI zprávy, které jsem využíval pro své zařízení. Čísla (na Obr. 4), před nimiž je znak „\$“ jsou prezentována v hexadecimální soustavě. Tyto čísla jsem vyhledal v brožuře od mého digitálního klavíru.



Obr. 4 Struktura MIDI zprávy All Note Off

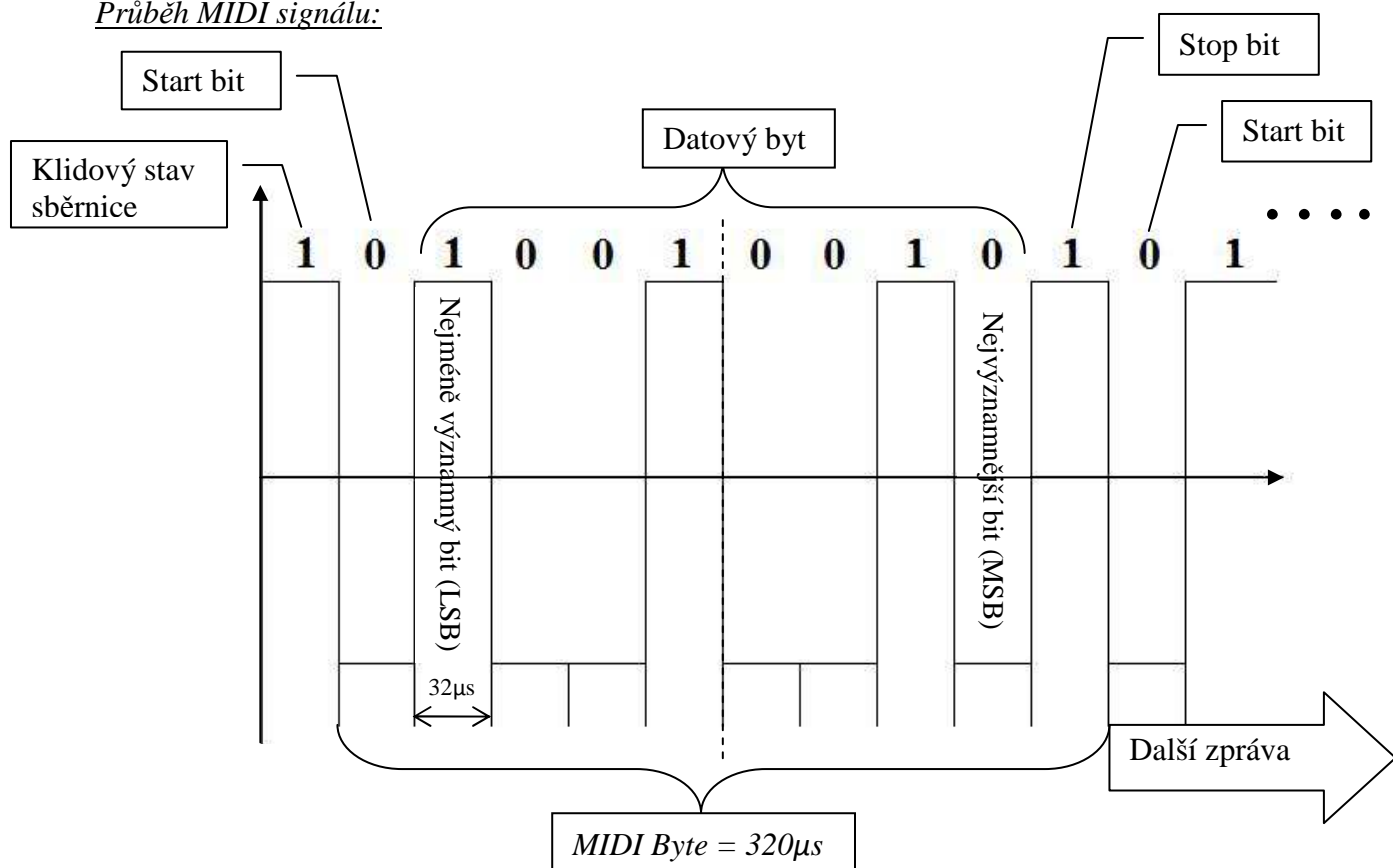
### 2.2.2 Způsob posílání dat

MIDI zařízení posílá data podobným způsobem jako komunikační rozhraní RS-232C, to znamená, že je posílá data asynchronně a jednosměrně, proto jsou potřeba 2 linky (IN/OUT). Jediným rozdílem mezi způsobem komunikace RS-232C a MIDI je, že MIDI sběrnice reprezentuje logické stavy pomocí 5mA proudové smyčky. Logické jedničky odpovídá otevřená proudová smyčka, neboli proud neprotéká, při logické nule je proudová smyčka uzavřena a proud protéká. MIDI signály se posílají invertované.



Když MIDI zařízení neposílá žádné znaky, tak je na sběrnici neustále logický stav jedna. Přenos zprávy začíná start-bitem s úrovní logické nuly, dále se odešle 8 datových bitů a to v pořadí od nejméně významného bitu po nejvýznamnější bit. Přenos se jedné MIDI zprávy se ukončuje jedním stop-bitem s úrovní logické jedničky. Každá zpráva obsahuje 10 bitů, které mají svůj význam. Jak už bylo zmíněné, tak 2 bity jsou pouze pro synchronizaci a zbylých 8 bitů jsou data nesoucí informace, jak jde vidět na Obr. 5 **Průběh MIDI signálu**. Přenosová rychlost (baud rate) MIDI zařízení je 31 250Baudů.

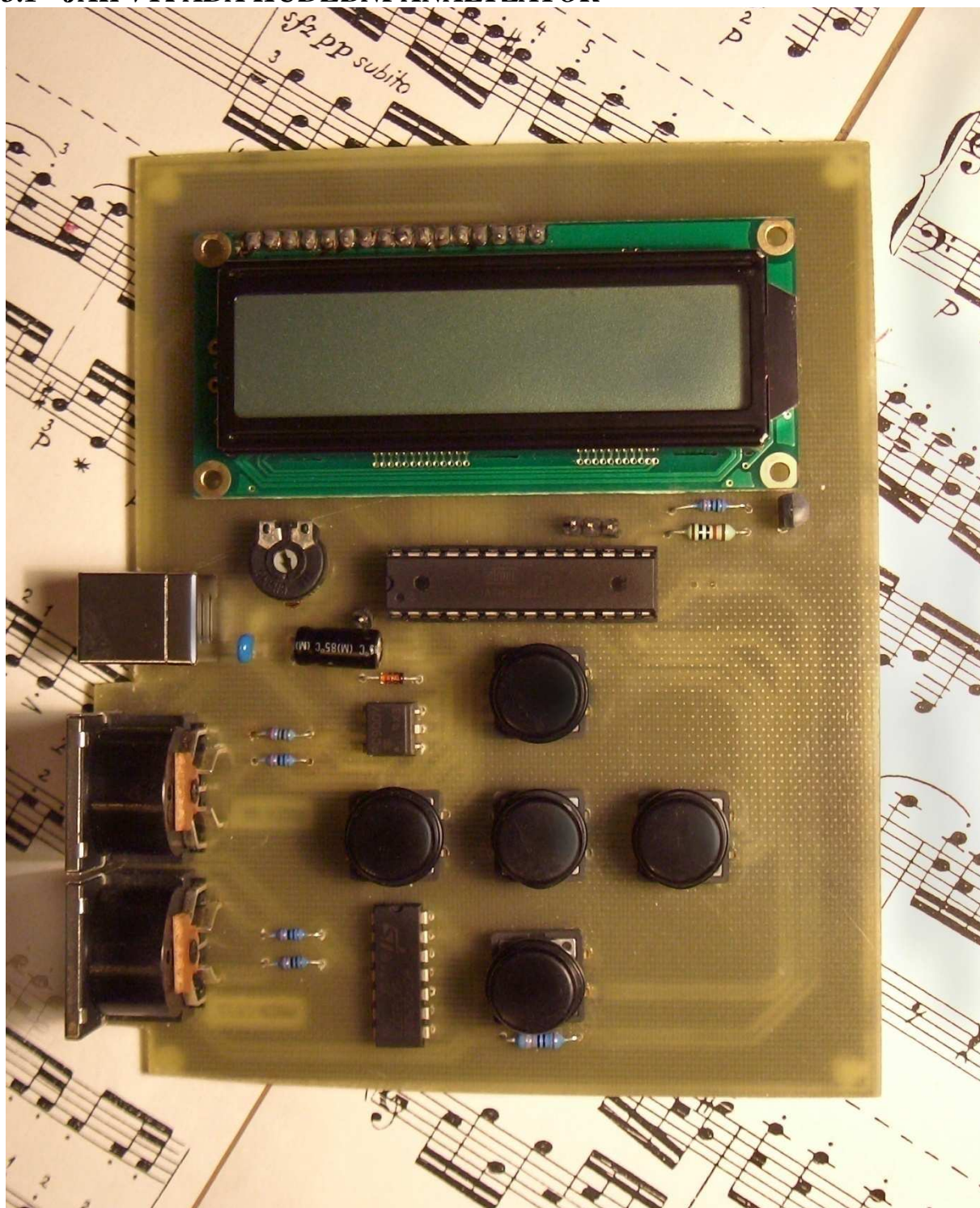
*Průběh MIDI signálu:*



Obr. 5 Průběh MIDI signálu

### 3 VYUŽITÍ MIDI KOMUNIKACE PŘI VÝUCE HUDBY

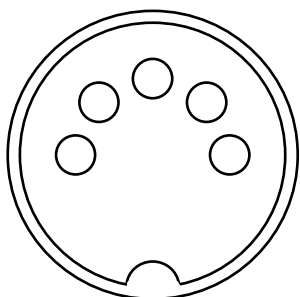
#### 3.1 JAK VYPADÁ HUDEBNÍ ANALYZÁTOR



Obr. 6 Popis hudebního analyzátoru

### 3.2 PŘIPOJENÍ HUDEBNIHO ANALYZÁTORU

Hudební analyzátor připojujeme pomocí dvou MIDI kabelů s konektorem DIN 5.

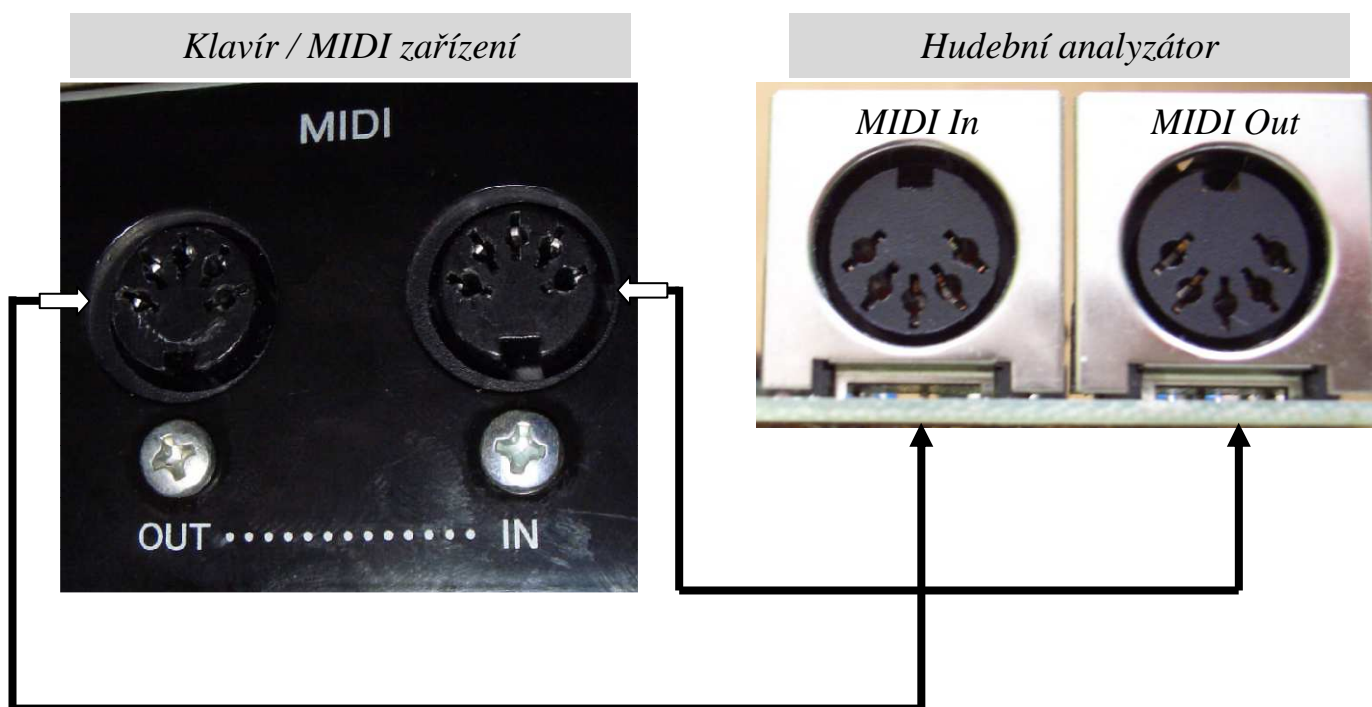


Obr. 7 Schéma DIN5 konektoru



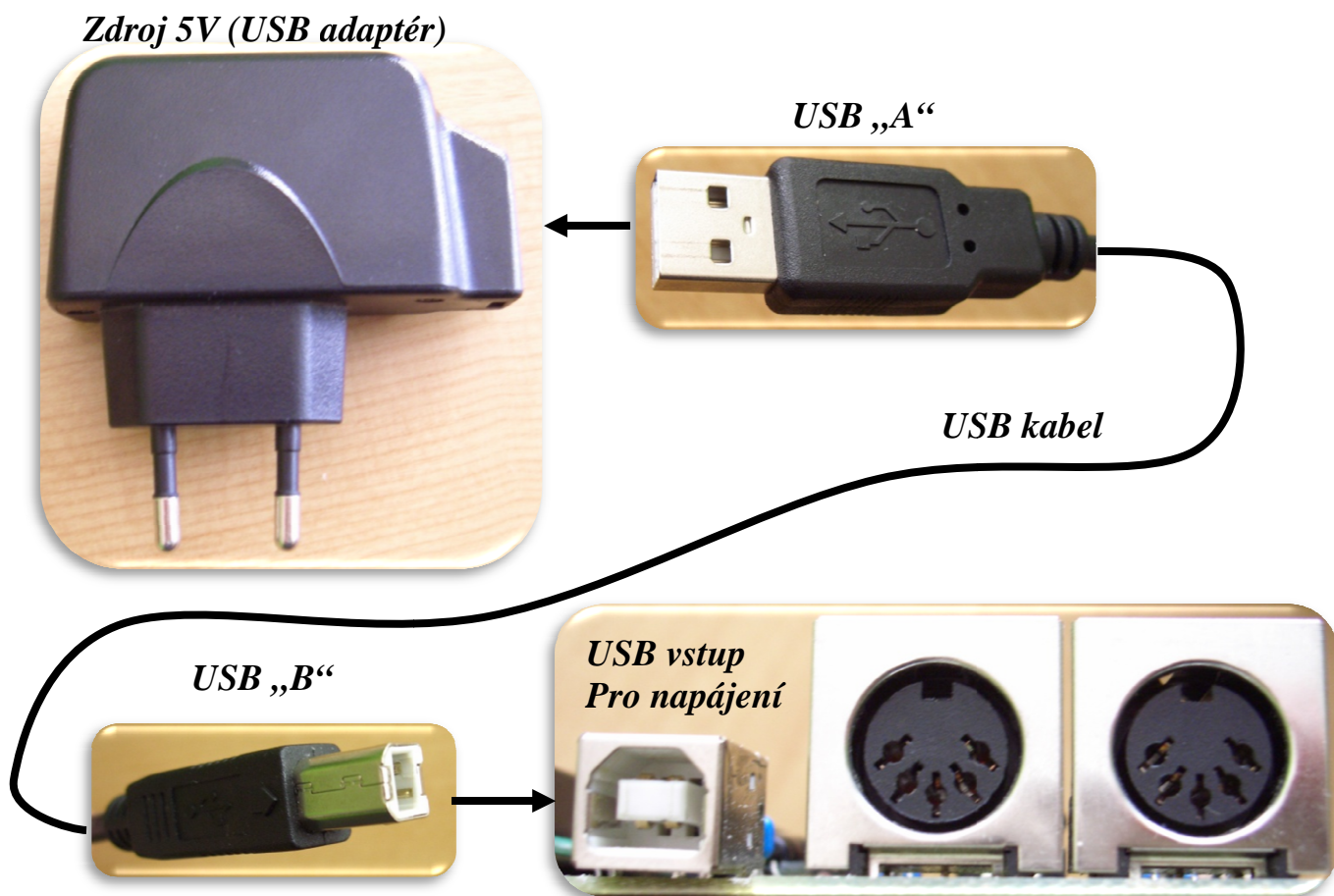
Obr. 8 DIN5 konektor

Označený vstup a výstup hudebního analyzátoru propojíme s MIDI zařízením následovně:



Obr. 9 Propojení MIDI zařízení s hudebním analyzátozem

Teď už jen zbývá připojení napájecího napětí. Jako napájecí konektor jsem použil USB vstup typu „B“. Použití USB konektoru neznamená, že potřebujeme k provozu nutně počítač s USB výstupy, stačí použít transformátor, který má výstup 5V (4,5 – 5,5 V). V dnešní době se často používají nabíječky na mobily, které mají výstup 5V a USB konektor. K propojení nám postačí USB kabel (z typu „A“ na typ „B“).



Obr. 10 Připojení napájení

### 3.3 PRVNÍ SPUŠTĚNÍ

Po připojení všech výše uvedených kabelů se nám rozsvítí displej, na kterém by mělo být zobrazeno:



Obr. 11 Úvodní obrazovka

Pokud ne, a zobrazí se hláška „Připojte klavír“, zkontrolujte připojení všech kabelů. Pokud je vše správně připojeno a hláška přetrvává, zkuste prohodit vstup s výstupem na MIDI zařízení.

### 3.4 OVLÁDÁNÍ HUDEBNÍHO ANALYZÁTORU

Na kompletní ovládání nám stačí pět tlačítek, která jsou umístěna tak, aby bylo ovládání co nejjednodušší a intuitivní. Na obrázku (Obr. 6) je všech pět tlačítek popsáno. Všechna tlačítka, kromě enteru, slouží na pohyb kurzoru, popřípadě mají i vedlejší vlastnost. Tlačítka nereagují okamžitě, aby se zamezilo nechtěnému opakování příkazu.

- Tlačítko *Vlevo* – kromě pohybu kurzoru (například při výběru ANO/NE) slouží pro odchod z podprogramu do hlavního programu. Když jsme v hlavním programu MIDI analýzy a chceme přejít přes rozcestník do „*Tréninku sluchu*“, stačí tlačítko podržet trochu déle, abychom se dostali do rozcestníku.
- Tlačítko *Vpravo* – jeho speciální funkce se používá pouze v podprogramu „*Stisklá klávesa*“, kde vymaže displej, kde se vypisují zahrané tóny.
- Tlačítko *Nahoru* – jeho speciální funkci využijeme v případě „*Tréninku sluchu*“ pro zvyšování úrovně obtížnosti.
- Tlačítko *Dolů* – stejně jako tlačítko *Nahoru* je jeho speciální funkcí pro snižování úrovně obtížnosti při „*Tréninku sluchu*“.
- Tlačítko *Enter* – díky poslednímu tlačítku můžeme vstupovat do podprogramů a potvrzovat různé volby. Při „*Tréninku sluchu*“ také funguje jako spouštění přehrávání intervalů, akordů a stupnic.

### 3.5 KONFIGURACE

Jelikož bylo snahou vytvořit hudební analyzátor, který by byl univerzální, a fungoval by na více MIDI zařízení, nejen určité značky, tak jsem program upravil tak, aby rozeznával dva druhy protokolu. Uživatel nemusí složitě zjišťovat, který protokol MIDI zařízení využívá ke komunikaci, tak jsem vytvořil program, který se o zjištění druhu protokolu postará.

Po spuštění hudebního analyzátoru se dostaneme přes rozcestník do podprogramu „*MIDI Analýza*“ a to tak, že ukazatelem (popsán na Obr. 11) najedeme na „*MIDI Analýza*“, klikneme na tlačítko *Enter*. Teď jsme se dostali do podprogramu. Pomocí šipky dolů prolistujeme seznam funkcí, jakmile najdeme funkci „*Konfigurace*“, najedeme na ni ukazatelem a stiskneme tlačítko *Enter*. Na obrazovce se vypíše hláška „*Stiskni klávesu*“, tak tedy příkaz provedeme a stiskneme klávesu na klavíru a budeme ji držet, dokud nás nevyzve hudební analyzátor, abychom ji pustili. Po zobrazení hlášky „*Pust klávesu*“ provedeme požadovaný úkon a tím konfiguraci ukončíme, automaticky se vrátíme do podprogramu „*MIDI Analýza*“.

Provést konfiguraci je důležité pokaždé, když změním nástroj, ke kterému připojujeme hudební analyzátor. Pokud neměníme připojené MIDI zařízení, tak nemusíme provádět konfiguraci, jelikož si hudební analyzátor zapamatuje naše nastavení.

## 3.6 JEDNOTLIVÉ FUNKCE HUDEBNÍHO ANALYZÁTORU

Funkce programu dělím na dva velké bloky, a to na Analýzu přijatých dat a posílání dat do klavíru.

- První blok „MIDI Analýza“ umí zjistit číslo kanálu, na kterém komunikuje klavír. Jakou klávesu jsme právě stiskli (s možností zobrazení oktávy), interval který jsme zahráli (harmonicky/melodicky), akord (vybrané trojzvuky a čtyřzvuky), a na závěr prvního bloku analýza zahrané stupnice.
- Druhý blok „Trénink sluchu“ posílá data do klavíru, který přehrává intervaly, akordy a stupnice s různou obtížností pro posluchače. Tento blok má sloužit k zlepšení relativního hudebního sluchu.

### 3.6.1 První blok „MIDI Analýza“

#### 3.6.1.1 Zjištění kanálu

##### Co to je MIDI kanál?

Je to virtuální sběrnice kudy prochází data k MIDI zařízením a zpět. Jedno MIDI zařízení (digitální hudební nástroj) může posílat či přijímat data maximálně přes jeden z 16 MIDI kanálů (jsou i výjimky, kdy se posílá jeden signál do více či všech MIDI kanálů). Toto omezení je dáno MIDI normou.

Komunikace přes kanály se většinou využívá v přítomnosti více digitálních hudebních nástrojů. Výhodou kanálů je možnost aplikování různých zvukových úprav na každý kanál zvlášť. Při zpracování MIDI signálů na počítači je výhoda, že díky kanálům můžeme rozlišovat, co bylo přijaté od kterého hudebního nástroje.

##### Zjištění, na kterém kanálu hudební nástroj komunikuje.

MIDI kanál zjistíme pomocí hudebního analyzátoru tak, že zvolíme v rozcestníku „MIDI Analýza“ a jako první funkcí je „Zjištění kanálu“. Stiskneme tlačítko *Enter*, na displeji se objeví hláška „Stiskni klávesu“. Stiskneme klávesu a displeji se objeví číslo, na kterém daný hudební nástroj komunikuje. Někdy se může stát, že po stisku klávesy se číslo MIDI kanálu neodešle (to záleží na protokolu, na kterém hudební nástroj komunikuje), tudíž hudební analyzátor nezjistí číslo MIDI kanálu. Když se to stane, tak stačí stisknout klávesu znovu. Pro opuštění této funkce stiskneme tlačítko *Vlevo*, a dostaneme se do menu „MIDI Analýza“.

Pokud se jedná o kytaru s MIDI výstupem, tak můžeme zjistit, na kterém kanále se odesílají data jednotlivých strun. Stačí brnkat do jednotlivých strun a rovnou se bude měnit číslo kanálu, v závislosti na strunách, na které brnkáme. Pokud se čísla nemění, tak kytara posílá informace o zahraných tónech pouze na jednom kanále, tudíž nerozlišuje jednotlivé struny do kanálů.

### 3.6.1.2 Stisknutá klávesa

Díky této funkci se můžeme naučit správně pojmenovat jednotlivé tóny a určit, v jaké oktávě se nacházejí. Tato funkce poslouží hlavně začínajícím hudebníkům.

Funkci „Stisklá klávesa“ najdeme v rozcestníku „MIDI Analýza“, hned jako druhá funkce. Když na ni najedeme kurzorem a stiskneme *Enter*, tak se náš hudební analyzátor zeptá, zda chceme rozlišovat i v jaké oktávě se hraný tón nachází.

Když zvolíme možnost bez vypisování čísel oktáv, tak se na displeji budou vypisovat pouze názvy hraných tónů za sebou. Druhou možností je vypisování názvy hraných tónů i s číslem oktávy.

Pokud máme již celý displej zaplněný názvy zahráných tónů, stačí stisknout tlačítko *Vpravo*, celý displej se smaže a můžeme pokračovat v zjišťování dalších názvů hraných tónů.

V kapitole 1.3 (Tónová soustava, jména tónů, notové písmo) jsme si uvedli, jak z názvu poznáme, zda se jedná o tón, který je v subkontraoktávě, nebo ve dvoučárkované oktávě.

Ze začátku budeme potřebovat ke správnému čtení noty obrázek Obr. 28 Klaviatura - Celý tón a půltón, tóny enharmonické) a pro správné pojmenování tabulku číslo Tab. 6: Tónová soustava – evropské rozdělení tónů do oktáv). Když se budeme postupně učit pojmenovávat jednotlivé noty v určitých blocích, tak se dostaneme i na úroveň kdy nebudeme potřebovat žádné tabulky ani obrázky na to, abychom pojmenovali všechny noty i se správným označením oktávy, které vidíme v naší ukázce.



Obr. 12 Ukázka aplikování funkce „Stisklá klávesa“

Funkci „Stisklá klávesa“ se dá využít i obráceně, nemusíme jen číst noty, hrát je, a poté si ověřit správnost přečtených not. Můžeme zkusit zapsat tóny, které jsme zahráli, do notové osnovy. Ze začátku není jednoduché rovnou zapisovat noty do notové osnovy, které jsme zahráli, tak můžeme zkusit přepsat názvy not do notové osnovy.

Mějme například zapsané názvy, které jsme si zahráli:

- d², h¹, c², f², e², a¹, e², d², h¹, e¹, g¹, g, h, e¹, f¹, d²

Přepis by mohl vypadat následovně:



Obr. 13 Přepis názvů not do notové osnovy

### 3.6.1.3 Intervaly

Funkce intervaly nám pomůže s tvořením intervalů, s jejich pojmenování a identifikování. Znat intervaly je důležité zejména pro lepší pochopení harmonií a tvorbu akordů.

Funkci „Intervaly“ najdeme v rozcestníku „MIDI Analýza“, jako třetí funkce. Po zvolené této funkce si musíme vybrat, zda chceme analyzovat intervaly melodické či harmonické. Naši volbu potvrdíme *Enterem* a můžeme se pustit do analyzování.

Když jsme si zvolili možnost hraní intervalů melodicky tak pro lepší orientaci v tom, kdy hrajeme první a druhý tón intervalu se vypisuje vpravo dole na displej hvězdička. Tato hvězdička značí, že hudební analyzátor čeká na první tón intervalu. Jakmile zahrajeme první tón, hvězdička zmizí, a můžeme zahrát druhý tón. Po zahrání druhého tónu se vypíše interval, který jsme zahráli spolu s hvězdičkou.

Zvolili jsme li možnost hraní intervalů harmonicky, nemusíme na nic čekat a rovnou hrát harmonicky intervaly jeden za druhým. Po zahrání intervalu harmonicky se na displeji vypíše jeho název.

Hudební analyzátor rozlišuje intervaly v rámci jedné oktávy, přesáhneme-li rámec jedné oktávy a zahrajeme interval větší než oktáva, tak nám hudební analyzátor vypíše jaký je to interval a vpravo dole na displej vypíše o kolik oktáv je druhý tón v intervalu posunutý.

Jako ukázkou si uvedeme noty zapsané v notové osnově.



Obr. 14 Intervaly v notové osnově

Obecný postup určování intervalů je následující:

1. Uvědomíme si durovou stupnici od prvního tónu
2. Nalezneme příslušný stupeň ve stupnici
3. Pokud tón horní tón není v utvořené stupnici, pak najdeme nejbližší vyšší tón ve stupnici a alterujeme ho.

Určení 1. Intervalu:

1. tón je C, vytvoříme stupnici C-dur (c,d,e,f,g,a,h). Druhý tón je E, ten se nachází na třetím stupni stupnice, tudíž se tento interval jmenuje *Velká tercie*.

Určení 2. Intervalu

1. tón je E, vytvoříme stupnici E-dur (e,fis,gis,a,h,cis,dis). Druhý tón je G, jelikož stupnice E-Dur nemá tón G, tak najdeme nejbližší vyšší tón. Tím tónem gis, který je na 3. stupni = *Velká tercie*. Jelikož jsme zvýšili (alterovali) hledaný tón, tak musíme snížit interval *Velká tercie*, stane se z ní *Malá tercie* a ta je naším hledaným intervalem.

3. Interval  $c^1 - c^2$  = Čistá oktáva
4. Interval  $c^1 - a^1$  = Velká sexta
5. Interval  $f^1 - g^1$  = Velká sekunda



6. Interval  $a^1 - d^2 =$  Čistá kvarta

7. Interval  $c^1 - a^1 =$  Velká tercie

Určení 8. Intervalu:

Nejprve určíme interval  $F - D =$  Velká sexta

Postupně budeme přidávat posuvky  $F - Des =$  Malá sexta

Další přidání béčka  $F - Deses =$  zmenšená sexta (enharmonická záměna = Čistá kvinta)

Přidání jednoho křížku  $Fis - Deses =$  dvojnemenšená sexta (= zmenšená kvarta)

Přidání posledního křížku  $Fisis - Deses =$  trojnemenšená sexta (= čistá kvarta)

Jelikož používání trojnemenšených a dvojnemenšených intervalů je nepraktické, tak se používají intervaly enharmonické. Takže 8. Interval v našem příkladě je *Čistá kvarta*.

### 3.6.1.4 Akordy

Bez akordů si nedokážeme představit žádnou písničku, pouhá melodie nedokáže vytvořit takovou náladu, jako akordy, které melodii podbarví a tím písnička dostane nový rozměr. Proto potřebujeme dobře znát akordy, abychom mohli hrát doprovody k písničkám. Akordy se nejčastěji zapisují pomocí akordových značek, jsou však i skladby, kde harmonický doprovod (rozložený akord, nebo jeho část) je přímo psán do not.

Funkci „Akordy“ najdeme v rozcestníku „MIDI Analýza“. Když na ni najedeme kurzorem a stiskneme *Enter*, tak nás hudební analyzátor upozorní na to, že rozeznává pouze akordy zahrané harmonicky = podmínkou je, že akord musí být zahrán harmonicky a všechny tóny musí zaznít v jednu chvíli, jinak analyzátor neanalyzuje zahráný akord, nebo ho analyzuje špatně. Tato funkce dokáže rozeznávat všechny akordy, o kterých jsem se zmínil v hudební teorii, samozřejmě že rozeznává i obraty kvintakordů a sextakordů. Pomocí této funkce můžeme ověřovat, zda jsme zadaný akord správně vytvořili, nebo jsme ho ze zadaných tónů správně analyzovali.

#### Vytvoření akordů:

Vytvořte následující akordy: C, Dmi, Hmi<sup>7</sup>, Fdim, Gmi<sup>5-</sup>, C<sup>5+</sup>, A<sup>7</sup>, B<sup>7maj</sup>, Es, Emi<sup>7maj</sup>

Po vytvoření akordů ověřte hudebním analyzátozem.

C – jedná se o Cdur tvořený velkou tercií a čistou kvintou od tónu C. {C – E – G}

Dmi – Dmoll mollový kvintakord (malá tercie + čistá kvinta). {D – F – A}

Hmi<sup>7</sup> – měkce malý septakord (malá tercie + čistá kvinta + malá septima) {H – D – F# – A}

Fdim – zmenšeně zmenšený septakord {F – G# – H – D}

Gmi<sup>5-</sup> – zmenšený kvintakord {G – B – D<sup>b</sup>}

C<sup>5+</sup> – zvětšený kvintakord {C – E – G#}

A<sup>7</sup> – tvrdě malý septakord {A – C# – E – G}

B<sup>7maj</sup> – tvrdě velký septakord {B – D – F – A}

Es – durový kvintakord {E<sup>b</sup> – G – B}

Emi<sup>7maj</sup> – měkce velký septakord {E – G – H – D#}

#### Analyzujte akord

Po přečtení a zahrání se na hudebním analyzátoru objeví název akordu. Když akord určíme, můžeme vyzkoušet vytvoření prvního a obratu.



Obr. 15 Analýza akordu

1. Akord – G
2. Akord – E<sub>m</sub>
3. Akord – F<sup>5+</sup>
4. Akord – D<sub>m</sub><sup>7maj</sup>
5. Akord – A<sup>5-</sup>
6. Akord – C<sup>7</sup>

### Obraty kvintakordů

Zjištění obratu kvintakordu je jednoduché, jelikož nám hudební analyzátor přímo napíše název obratu.

Zahrajeme-li 1. obrat C-dur, tak hudební analyzátor nám napíše na displej:

**C – dur**  
**Sextakord**

Při druhém obratu se vypíše na displej místo sextakordu „kvartsextakord“

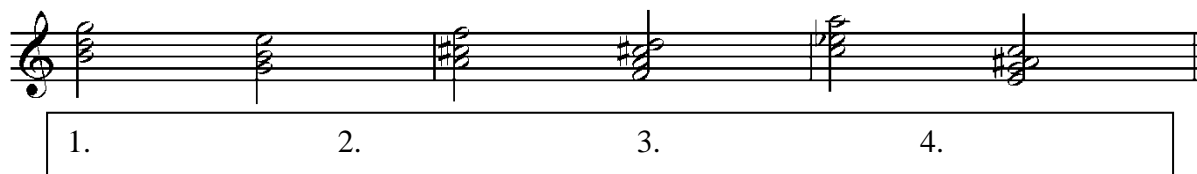
### Obraty septakordů

Když zahrajeme 1. obrat dominantního septakordu C<sup>7</sup>, tak na displeji již nevidíme přímo název obratu, ale pouze číslo, o který obrat jde. Číslo najdeme v pravém dolním rohu displeje.



Obr. 16 Displej - zobrazení obratu

Vytvoření prvních obratů je velice jednoduché. Stačí, když nejnižší tón v akordu přeložíme o oktávu výš, tím dostaneme 1. Obrat. Vytvoříme-li k předešlému zadání (Obr. 15) první obraty, budou vypadat následovně:



Obr. 17 první obraty akordů

Vytvoření druhého obratu spočívá v přeložení nejnižšího tónu v prvním obratu akordu o oktávu výš. Druhý obrat je zároveň posledním obratem pro kvintakord, dalším přeložením tónu vniká základní akord celý posunutý o oktávu výš.

Třetí obrat je pouze u septakordů (nónových akordů a dalších), vytvoříme ho přeložením nejnižšího tónu u druhého obratu akordu.

### 3.6.1.5 Stupnice

Jak už jsme zjistili, tak stupnice jsou důležité pro určování intervalů a akordů. Dále jejich znalost využijeme při improvizaci a při hraní skladeb. Ve skladbách jsou většinou použity stupnice a akordy, hlavně jejich části, které jsou různě poskládané.

Poslední funkcí v rozcestníku „MIDI Analýza“ jsou „Stupnice“. Hudební analyzátor dokáže rozeznat nejpoužívanější stupnice a to durové a mollové, hrané od nejnižšího po nejvyšší stupeň stupnice. Z hudební teorie, mollové stupnice se dělí na tři druhy – Aiolská, Melodická a Harmonická. Všechny stupnice mají 8 stupňů, tudíž hudební analyzátor čeká, až zahráme všechny stupně stupnice. Abychom se mohli orientovat v tom, kdy hudební analyzátor začíná analyzovat novou stupnici, tak v pravém dolním rohu displeje se vypíše hvězdička. Hvězdička nám tedy značí, že následující nota, která zazní, bude brána jako první stupeň stupnice. Jakmile zahráme první stupeň stupnice, hvězdička zmizí. Jakmile zahráme stupnici, na displeji se objeví název stupnice, který jsme zahráli, nebo informace o tom, že jednotlivé tóny netvoří žádnou z uvedených stupnic. Hned po vypsání této zprávy se na rohu displeje vypíše hvězdička, která nám signalizuje začátek nové stupnice. Hudební analyzátor při hraní hlídá jednotlivé velikosti intervalů mezi jednotlivými stupni stupnice, pokud bude interval větší než malá tercie, tak hudební analyzátor automaticky vypíše na displej informaci o nesmyslné stupnici a začne analyzovat stupnici novou.

Pomocí této funkce můžeme ověřovat, zda jsme jednotlivé stupnice správně vytvořili a zahráli. Když už dobře známe stupnice podle sluchu, můžeme si vyzkoušet vytvořit jakoukoliv stupnici od jakéhokoliv tónu, a postupovat podle sluchu. Při hraní stupnice podle sluchu nám dá hudební analyzátor tolik času, kolik potřebujeme na to, abychom si rozmysleli, jaký bude druhý, třetí a další stupeň stupnice. Pokud máme stupnici správně naučenou a neudělali jsme přehmat, tak hudební analyzátor bude s námi souhlasit v zahrané stupnici, a pokud ne, tak nám na displej vypíše: „To co jsi zahrál, není stupnice“

### Stupnice prakticky



Obr. 18 Analýza stupnice

1. Stupnice je bez jakéhokoli křížku

Určíme půltóny mezi jednotlivými stupni: C - D = 2; D - E = 2; E - F = 1; F - G = 2; G - A = 2; A - H = 2; H - C = 1. Schéma půltónů je: 2, 2, 1, 2, 2, 2, 1 tohle schéma je shodné s durovou stupnicí, jelikož se začínalo od tónu C, tak se stupnice jmenuje C-dur.

2. Stupnice = A-moll aiolská

3. Stupnice = A-dur

4. Stupnice = D-moll harmonická

Stupnicemi končí první rozcestník „MIDI Analýza“, ještě tam najdeme funkci „Konfigurace“, kterou jsme si již dříve popisovali v kapitole 3.5 (Konfigurace).

### 3.6.2 Druhý blok „Trénink sluchu“

Jak už název napovídá, tento blok se bude zabývat tréninkem sluchu. V této kapitole si rozebereme jednotlivá cvičení a napíšeme si, co můžeme očekávat v jednotlivých stupních obtížnosti.

#### 3.6.2.1 Intervaly

V této funkci se naučíme rozpoznávat intervaly pomocí sluchu od nejjednodušších po složitější, postupně budou přibývat další a další. Vždy se budeme učit určovat intervaly, které jsou stoupající, nebo harmonicky znějící. Funkce intervaly má 12 úrovní obtížnosti.

„Intervaly“ najdeme v rozcestníku „Trénink sluchu“ hned jako první funkce, po potvrzení tlačítkem *Enter* se dostaneme do funkce „Intervaly“, kde nás hudební analyzátor vyzve ke zvolení si úrovně obtížnosti přehrávaných intervalů. Úroveň obtížnosti se indikuje číslem od 1 do 12, obtížnost zvyšujeme tlačítkem *Nahoru*, a snižujeme tlačítkem *Dolů*, po potvrzení úrovně stiskneme tlačítko *Enter*. Hudební analyzátor bude přehrávat intervaly, vždy když stiskneme tlačítko *Enter*. Po přehrávání intervalu hudebním analyzátozem nás vyzve k tomu, abychom zahráli stejný interval (melodicky) od kteréhokoliv tónu. Před zahráním intervalu se na displeji vpravo dole vytiskne hvězdička, která značí, že hudební analyzátor čeká na první tón intervalu. Jakmile zahráme první tón, hvězdička zmizí, a můžeme zahrát druhý tón. Pokud jsme zahráli interval správně, tak nám analyzátor vypíše na displej „Správně“ v opačném případě to bude zpráva „Špatně“. Teď bude hudební analyzátor čekat na to, až stiskneme znovu tlačítko *Enter*, aby mohl přehrát další interval a mi jej analyzovali. Jako u všech funkcí, i u této funkce funguje tlačítko *Vlevo*, díky kterému se dostaneme zpět do rozcestníku „Trénink sluchu“

Intervaly jsou ve třech blocích, v každém dalším bloku přibudou nové intervaly. Doporučuji jednotlivé bloky opouštět, až máte jistotu ve všech intervalech, které jsou přehrávané v daném bloku. Intervaly, se přehrávají v náhodném pořadí, takže se nemůže stát, že byste našli nějakou periodu intervalů, která by se neustále opakovala.

#### Úrovně obtížnosti:

1. Melodicky – Čistá prima, velká sekunda a Velká tercie - od tónu  $c^1$
2. Harmonicky – Čistá prima, velká sekunda a Velká tercie - od tónu  $c^1$
3. Melodicky – Čistá prima, velká sekunda a Velká tercie - od tónu  $c^1$  až po tón  $c^2$

4. Harmonicky – Čistá prima, velká sekunda a Velká tercie - od tónu  $c^1$  až po tón  $c^2$
5. Melodicky – Čistá prima až po čistou kvintu - od tónu  $c^1$
6. Harmonicky – Čistá prima až po čistou kvintu - od tónu  $c^1$
7. Melodicky – Čistá prima až po čistou kvintu - od tónu  $c^1$  až po tón  $c^2$
8. Harmonicky – Čistá prima až po čistou kvintu - od tónu  $c^1$  až po tón  $c^2$
9. Melodicky – Čistá prima až po čistou oktávu - od tónu  $c^1$
10. Harmonicky – Čistá prima až po čistou oktávu - od tónu  $c^1$
11. Melodicky – Čistá prima až po čistou oktávu - od tónu  $c^1$  až po tón  $c^2$
12. Harmonicky – Čistá prima až po čistou oktávu - od tónu  $c^1$  až po tón  $c^2$

### 3.6.2.2 Akordy

Poznat Dur a moll, to snad není žádný problém, ale co takový tvrdě velký septakord od tvrdě malého septakordu? Pokud máte podobný problém rozeznat tyto akordy podle sluchu, tak tenhle problém vám vyřeší funkce „Akordy“. Tato funkce je obsáhlejší z důvodů velké škály akordů (mluvíme pouze o akordech, které jsme se naučili v kapitole 1.8 (Akordy)). Funkce „Akordy“ je podobná funkci „Intervaly“, až na to, že má 32 úrovní obtížností a místo intervalů se přehrávají akordy.

„Akordy“ najdeme v rozcestníku „Trénink sluchu“. Hudební analyzátor si opět vynutí zvolení úrovně obtížnosti přehrávaných akordů. Úroveň obtížnosti se indikuje číslem od 1 do 32, obtížnost zvyšujeme tlačítkem *Nahoru*, a snižujeme tlačítkem *Dolů*, po potvrzení úrovně stiskneme tlačítko *Enter*. Hudební analyzátor bude přehrávat akordy, vždy když stiskneme tlačítko *Enter*. Po přehrávání akordu hudebním analyzátozem nás vyzve k tomu, abychom zahráli stejný akord (vždy harmonicky) od kteréhokoliv tónu. Pokud jsme analyzovali akord správně, tak nám analyzátor vypíše na displej „Správně“ v opačném případě to bude zpráva „Špatně“. Hudební analyzátor vždy čeká na to, až stiskneme znovu tlačítko *Enter*, aby mohl přehrát další akord a mi jej analyzovali. Jako u všech funkcí, i u této funkce funguje tlačítko *Vlevo*, díky kterému se dostaneme zpět do rozcestníku „Trénink sluchu“

Akordy jsou ve třech blocích, v každém dalším bloku přibudou nové akordy. Doporučuji jednotlivé bloky opouštět, až máte jistotu ve všech akordech, které jsou přehrávané v daném bloku. Stejně jako u intervalů je výběr akordů náhodný.

#### Úrovně obtížnosti:

1. Melodicky – Dur/moll kvintakord - od tónu  $c^1$
2. Harmonicky – Dur/moll kvintakord - od tónu  $c^1$
3. Melodicky – Dur/moll kvintakord - od tónu  $c^1$  až po tón  $c^2$
4. Harmonicky – Dur/moll kvintakord - od tónu  $c^1$  až po tón  $c^2$
5. Melodicky – zvětšený/zmenšený kvintakord - od tónu  $c^1$
6. Harmonicky – zvětšený/zmenšený kvintakord - od tónu  $c^1$
7. Melodicky – zvětšený/zmenšený kvintakord - od tónu  $c^1$  až po tón  $c^2$
8. Harmonicky – zvětšený/zmenšený kvintakord - od tónu  $c^1$  až po tón  $c^2$

9. Melodicky – Dur, moll, zvětšený, zmenšený kvintakord - od tónu  $c^1$
10. Harmonicky – Dur, moll, zvětšený, zmenšený kvintakord - od tónu  $c^1$
11. Melodicky – Dur, moll, zvětšený, zmenšený kvintakord - od tónu  $c^1$  až po tón  $c^2$
12. Harmonicky – Dur, moll, zvětšený, zmenšený kvintakord - od tónu  $c^1$  až po tón  $c^2$
  
13. Melodicky – Tvrdě velký/Tvrdě malý septakord - od tónu  $c^1$
14. Harmonicky – Tvrdě velký/Tvrdě malý septakord - od tónu  $c^1$
15. Melodicky – Tvrdě velký/Tvrdě malý septakord - od tónu  $c^1$  až po tón  $c^2$
16. Harmonicky – Tvrdě velký/Tvrdě malý septakord - od tónu  $c^1$  až po tón  $c^2$
  
17. Melodicky – Měkce malý/Zmenšeně zmenšený septakord - od tónu  $c^1$
18. Harmonicky – Měkce malý/Zmenšeně zmenšený septakord - od tónu  $c^1$
19. Melodicky – Měkce malý/Zmenšeně zmenšený septakord - od tónu  $c^1$  až po tón  $c^2$
20. Harmonicky – Měkce malý/Zmenšeně zmenšený septakord - od tónu  $c^1$  až po tón  $c^2$
  
21. Melodicky – Všechny zmíněné septakordy - od tónu  $c^1$
22. Harmonicky – Všechny zmíněné septakordy - od tónu  $c^1$
23. Melodicky – Všechny zmíněné septakordy - od tónu  $c^1$  až po tón  $c^2$
24. Harmonicky – Všechny zmíněné septakordy - od tónu  $c^1$  až po tón  $c^2$
  
25. Melodicky – Všechny zmíněné septakordy + Dur/moll - od tónu  $c^1$
26. Harmonicky – Všechny zmíněné septakordy + Dur/moll - od tónu  $c^1$
27. Melodicky – Všechny zmíněné septakordy + Dur/moll - od tónu  $c^1$  až po tón  $c^2$
28. Harmonicky – Všechny zmíněné septakordy + Dur/moll - od tónu  $c^1$  až po tón  $c^2$
  
29. Melodicky – Všechny zmíněné septakordy a kvintakordy - od tónu  $c^1$
30. Harmonicky – Všechny zmíněné septakordy a kvintakordy - od tónu  $c^1$
31. Melodicky – Všechny zmíněné septakordy a kvintakordy - od tónu  $c^1$  až po tón  $c^2$
32. Harmonicky – Všechny zmíněné septakordy a kvintakordy - od tónu  $c^1$  až po tón  $c^2$

### 3.6.2.3 *Stupnice*

Funkce „*Stupnice*“ nám pomůže se zdokonalit v rozlišování čtyř stupnic a to durových a mollových (Aiolská, Harmonická a Melodická).

„*Stupnice*“ najdeme v rozcestníku „*Trénink sluchu*“. Tentokrát si vybíráme pouze mezi dvěma úrovněmi. Obtížnost zvyšujeme tlačítkem *Nahoru*, a snižujeme tlačítkem *Dolů*, tlačítkem *Enter* potvrdím naši volbu. Hudební analyzátor přehraje stupnici, vždy když stiskneme tlačítko *Enter*. Po přehrání stupnice nás hudebním analyzátozem vyzve k tomu, abychom zahráli stejnou stupnici od kteréhokoliv tónu. Všechny stupnice mají 8 stupňů, tudíž hudební analyzátor čeká, až zahrajeme všechny stupně stupnice. Abychom se mohli orientovat v tom, kdy hudební analyzátor začíná analyzovat novou stupnici, tak v pravém dolním rohu displeje se vypíše hvězdička. Hvězdička nám tedy značí, že následující nota, která zazní, bude brána jako první stupeň stupnice. Jakmile zahrajeme první stupeň stupnice, hvězdička zmizí. Po zahrání stupnice nám hudební analyzátor vypíše, zda jsme zahráli stupnici správně, či ne. Vždy vypíše správnou odpověď na druhý řádek displeje. Teď bude hudební analyzátor čekat na to, až stiskneme znovu tlačítko *Enter*, aby mohl přehrát další stupnici a mi ji mohli analyzovat.

Jako u všech funkcí, i u této funkce funguje tlačítko *Vlevo*, díky kterému se dostaneme zpět do rozcestníku „*Trénink sluchu*“.

#### Úrovně obtížnosti:

1. Úroveň – hrání všech stupnic od tónu  $c^1$
2. Úroveň – hrání všech stupnic od tónu  $c^1$  až po tón  $c^2$

Funkce „*Stupnice*“ která se nachází v rozcestníku „*Trénink sluchu*“ ukončuje celý výčet funkcí hudebního analyzátoru. Když zvládnete všechny úrovně tréninku sluchu, tak nebudete mít problém například se splněním požadavků ohledně hudebního sluchu na přijetí na konzervatoř. Samozřejmě, že tyto cvičení nejsou jen kvůli přijímačkám na konzervatoř, ale i pro zlepšení hudební představitivosti. Nebudete mít problém si zahrát melodii, kterou jste slyšeli, a při lepším poslechu můžete i odvodit akordy a už nebudete potřebovat noty, které třeba nejdou ani sehnat, nebo se vám nelíbí úprava autora not.

## **4 KONSTRUKČNÍ DOKUMENTACE**

Při návrhu hudebního analyzátoru jsem část zapojení čerpal z uvedené literatury[1], zbytek jsem zapojil podle datasheetu Atmegy8 s menšími úpravami.

### **4.1 PRINCIP ČINNOSTI**

Hlavním prvkem v zapojení hudebního analyzátoru je mikropočítač Atmega8, který se stará o celý chod hudebního analyzátoru. Další důležitou částí hudebního analyzátoru je displej, který zobrazuje analyzovaná data. Pět tlačítek slouží k ovládní hudebního analyzátoru a komunikaci s ním.

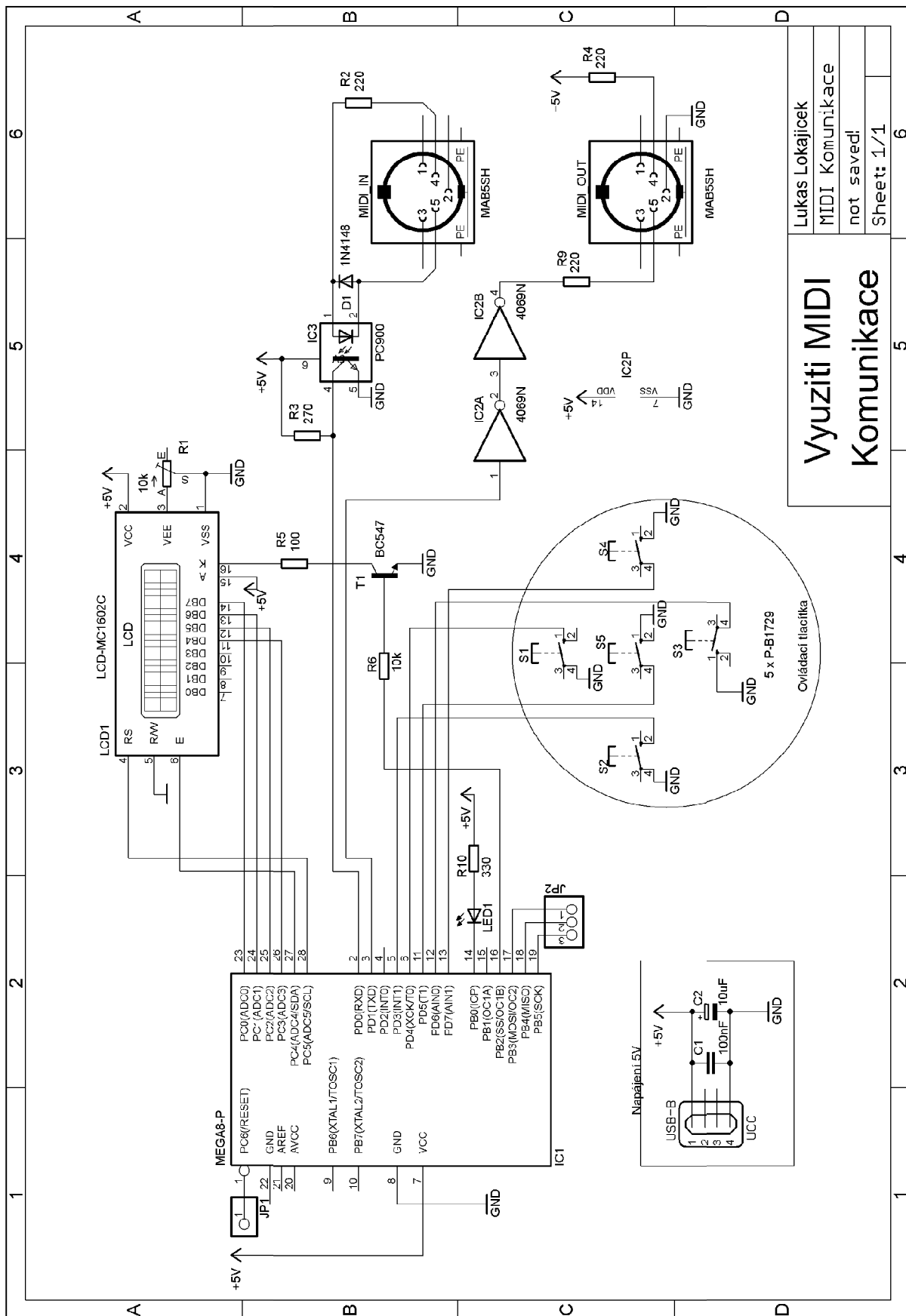
Mikropočítač komunikuje s hudebním nástrojem pomocí zařízení UART (Universal Asynchronous Receiver and Transmitter). Jelikož mikropočítač zpracovává napěťové úrovně 5V a hudební nástroj s MIDI výstupem reprezentuje logické stavy pomocí 5mA proudové smyčky, tak musíme použít jednoduchý převodník. V zapojení jsem použit optoelektronický oddělovací člen PC900 (dále optočlen), který převádí signál z 5mA smyčky na napěťové úrovně. Když hudební nástroj pošle pulz, rozsvítí se IR dioda uvnitř optočlenu, fototranzistor se otevře a napájecí napětí přes odpor svede k zemi, tudíž na výstupu optočlenu bude 0V. Nepošle-li hudební nástroj žádný signál, dioda nesvítí, fototranzistor je uzavřený a na výstupu optočlenu je 5V. Když ještě jednou podíváme, na funkci optočlenu, tak zjistíme, nejen že signál převádí na jiné úrovně, ale zároveň ho invertuje. V kapitole 2.2.2 jsme se dozvěděli, že MIDI signály jsou invertované, tudíž jsem nemusel používat invertor.

LCD displej je připojen pomocí čtyř datových vodičů, ostatní jsou použity jako adresové. Trimr u displeje slouží pro nastavení kontrastu displeje. Podsvícení je spínáno přes NPN tranzistor ve spínacím režimu, jelikož nemůžu tolik zatěžovat Atmegu.

Díky tlačítkům se můžeme pohybovat v prostředí hudebního analyzátoru. Atmega má na vstupech, kde jsou připojená tlačítka, pull-up rezistory neboli upínací rezistory. Když tlačítko není stisklé, tak je na vstupu logická jednička. Při stisku tlačítka připojíme vstup Atmegy na zem, tím svedeme napájecí napětí přes vnitřní odpor Atmegy a na vstupu bude logický stav 0.

Napájecí napětí je přivedeno ze stabilizovaného zdroje do obvodu přes USB-B konektor.

## 4.2 SCHÉMA HUDEBNÍHO ANALYZÁTORU



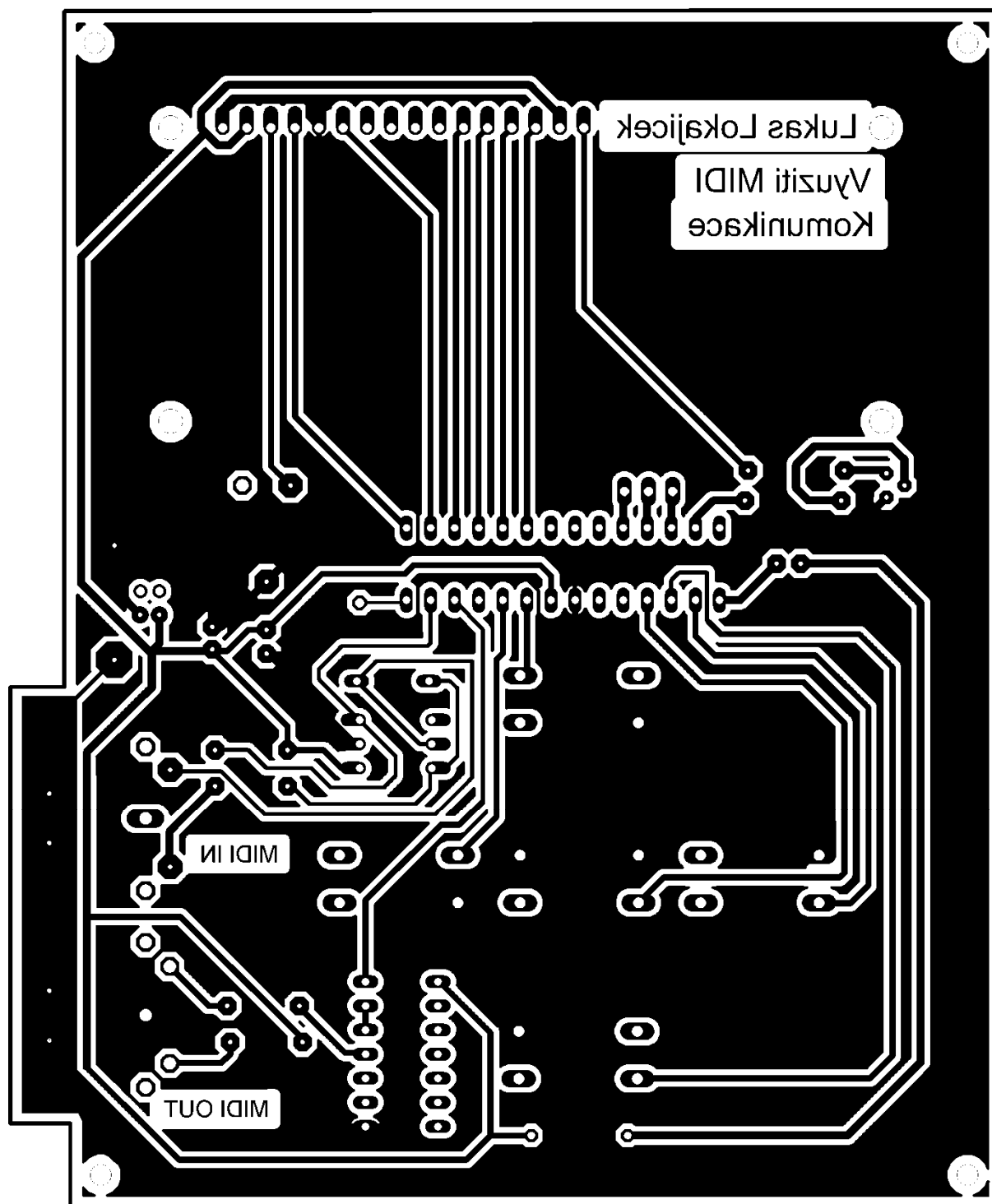
**Využití MIDI  
Komunikace**

Lukas Lokajicek  
MIDI Komunikace  
not saved!  
Sheet: 1/1

Obr. 19 Schéma hudebního analyzátoru



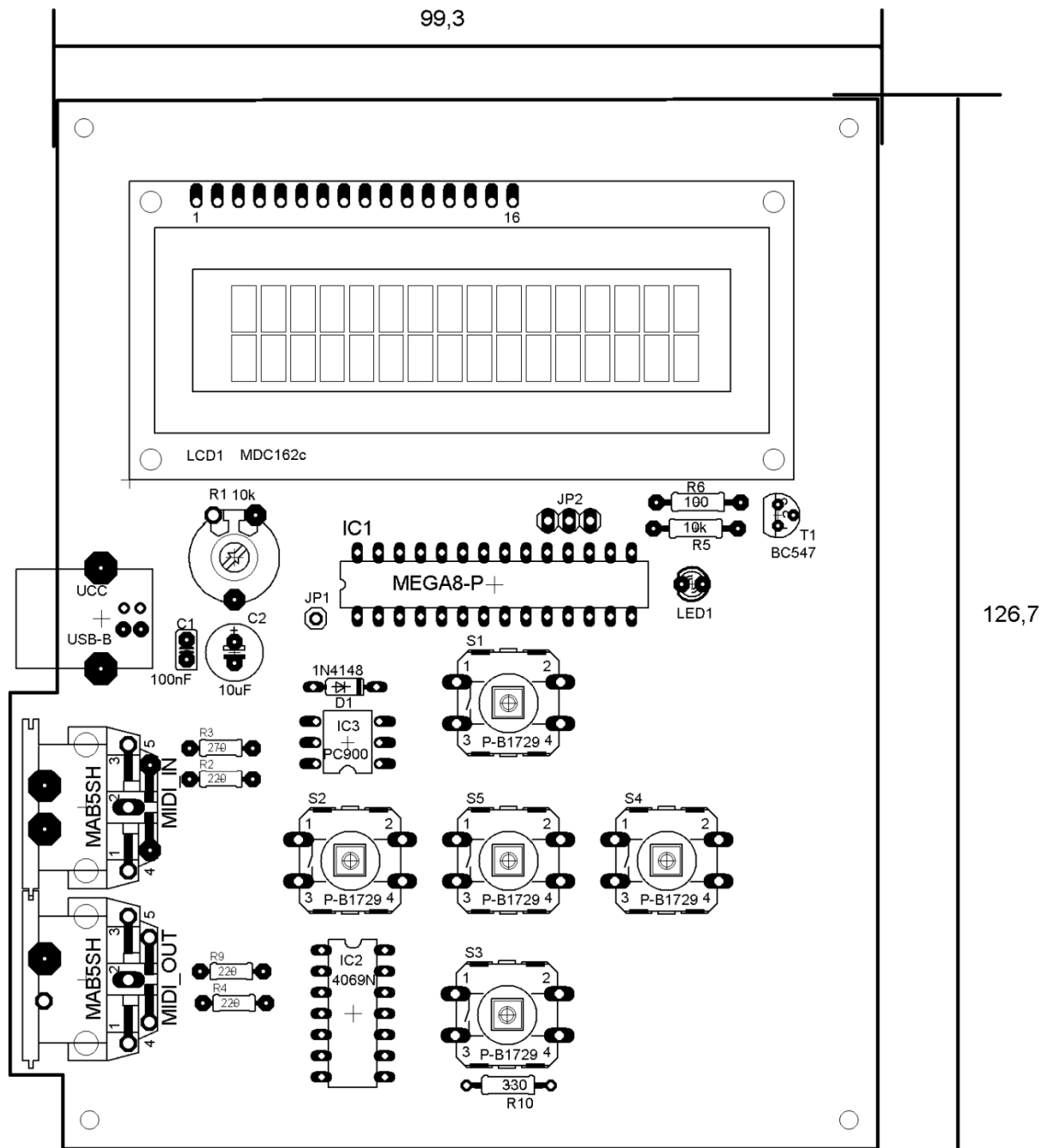
### 4.3 KLIŠÉ HUDEBNÍHO ANALYZÁTORU



Obr. 20 Klišé hudebního analyzátoru

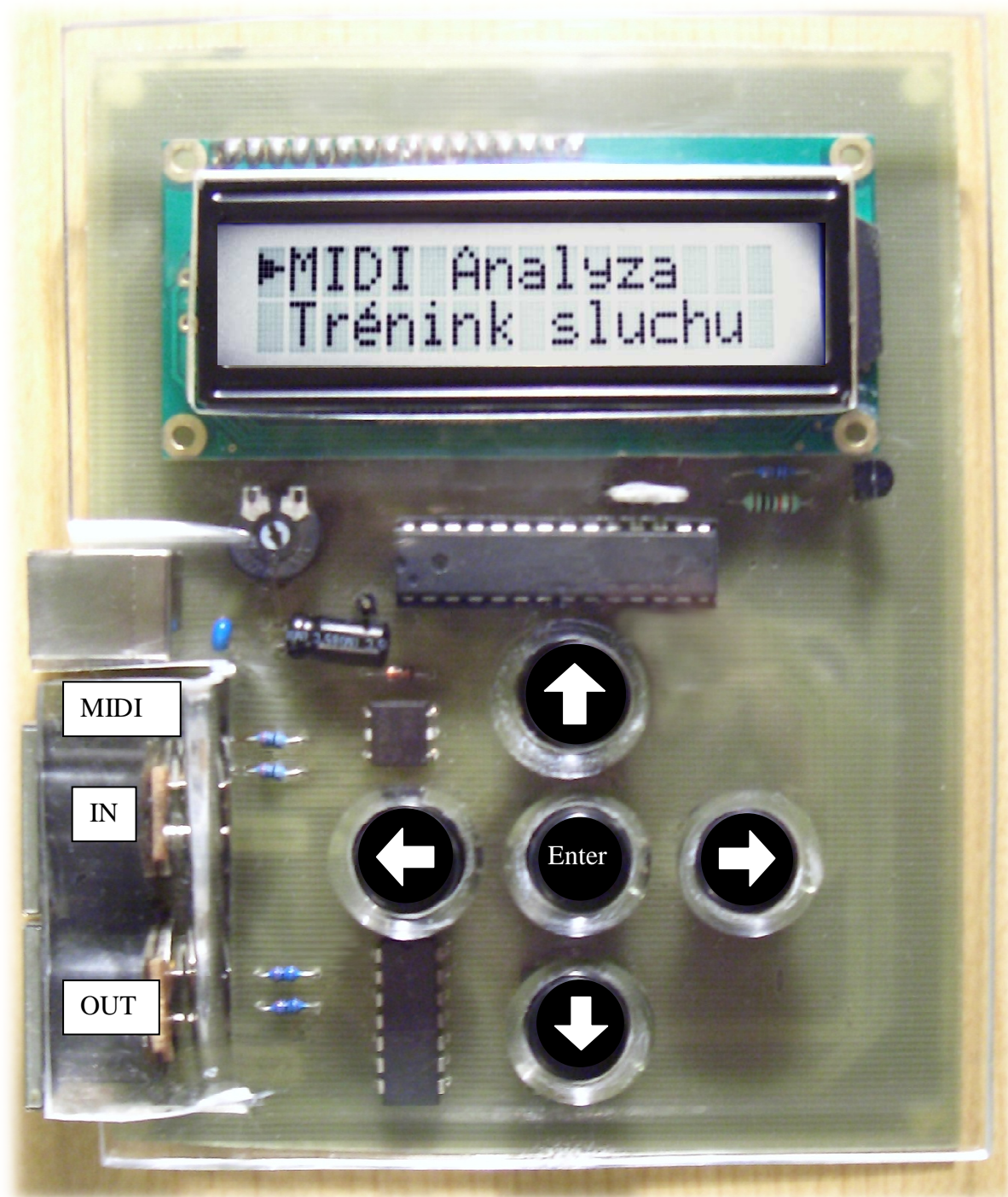
Rozměry klišé jsou shodné s rozměry osazovacím schématem.(127,6 x 99,3)

## 4.4 OSAZOVACÍ SCHÉMA



Obr. 21 Osazovací schéma

#### 4.5 REALIZACE HUDEBNÍHO ANALYZÁTORU



Obr. 22 Zrealizovaný hudební analyzátor

## 4.6 ROZPISKA SOUČÁSTEK

### Rezistory:

Označení ve schématu	Hodnota	Poznámka
R <sub>1</sub>	10kΩ	Trimr lineární
R <sub>2</sub>	220Ω	¼W
R <sub>3</sub>	270Ω	¼W
R <sub>4</sub>	220Ω	¼W
R <sub>5</sub>	100Ω	¼W
R <sub>6</sub>	10kΩ	¼W
R <sub>9</sub>	220Ω	¼W
R <sub>10</sub>	330Ω	¼W

### Kondenzátory:

Označení ve schématu	Hodnota	Poznámka
C <sub>1</sub>	100nF	Keramický
C <sub>2</sub>	10uF	Elektrolitycký, 16V

### Diody:

Označení ve schématu	Typové označení	Poznámka
D <sub>1</sub>	1N4148	Rychlá dioda
LED <sub>1</sub>	L-LTL4231N-T	Zelená LED 10mA

### Tranzistor:

Označení ve schématu	Typové označení	Poznámka
T <sub>1</sub>	BC547-B	NPN tranzistor

### Integrované obvody:

Označení ve schématu	Typové označení	Poznámka
IC <sub>1</sub>	Atmega8A-16PU	Mikro počítač s patičí
IC <sub>2</sub>	HCF4069UBE	CMOS invertor
IC <sub>3</sub>	PC900V	Optočlen

### Konektory:

Označení ve schématu	Typové označení	Poznámka
MIDI IN (MAB5SH)	DIN 5 P ZP90-ST	DIN5 do DPS, zásuvka
MIDI OUT (MAB5SH)	DIN 5 P ZP90-ST	DIN5 do DPS, zásuvka
U <sub>cc</sub> USB-B	USB1X90B PCB	USB B do DPS, zásuvka 90°
JP <sub>1</sub>	S1G40	Oboustranný kolík 1x
JP <sub>2</sub>	S1G40	Oboustranný kolík 3x

Ostatní:

Označení ve schématu	Typové označení	Poznámka
S <sub>1</sub> – S <sub>5</sub>	P-B1729	5 x mikrospínač
LCD <sub>1</sub>	MC1602	LCD displej

## 5 OBSLUŽNÝ SOFTWARE

Obslužný software jsem psal v jazyku C ve vývojovém prostředí Code Vision AVR. Neuvádím zde svůj celý program, jelikož chci zabránit kopírování mé dlouhodobé práce. Pouze vás provedu hlavními částmi programu a mojí myšlenkou.

### 5.1 UART KOMUNIKACE

Velmi důležitou částí programu je inicializace UART komunikace. Code Vision AVR nabízí při spuštění průvodce, díky kterému můžeme nastavit různé způsoby posílání dat a hlavně rychlost posílání a přijímání dat (bitů). Inicializací se rozumí správné nastavení typu protokolu, pomocí kterého budeme komunikovat. Z úvodu této dokumentace již víme, jak MIDI komunikace probíhá, a jaké má vlastnosti.

Pomocí průvodce jsem si nastavil:

- 1 Start bit
- 8 datových bitů
- Žádný paritní bit
- 1 stop bit

Průvodce samozřejmě nabídl i několik rychlostí komunikace v Baudech (jednotka rychlosti polání informací/bitů za sekundu). Mezi svými nabízenými hodnotami neměl požadovanou rychlost MIDI sběrnice a to 31 250bit/s (Baud). V závislosti na rychlosti komunikace se ukládá do registru **UBRR** číslo, které nadefinuje rychlost komunikace. Uložené číslo v registru **UBRR** se nerovná rychlosti v baudech, je to pouze vztah mezi taktovací frekvence Atmegy a rychlosti komunikace. V datasheetu je uvedený vzorec, podle kterého se dá vypočítat číslo, určené pro uvedený registr.

$$UBRR = \frac{f_{oscilátoru\ atmegy}}{16 \cdot BaudRate} - 1 = \frac{1\ 000\ 000Hz}{16 \cdot 31\ 250} - 1 = 2 - 1 = 1$$

Kdyby číslo UBRR nevyšlo celé, nastávala by chyba při přenosu. I přesto zde uvádím (pro úplnost) i způsob výpočtu, které nám odhalí chybovost přenosu dat.

$$BaudRate = \frac{f_{oscilátoru\ atmegy}}{16 \cdot (UBRR + 1)} = \frac{1\ 000\ 000Hz}{16 \cdot (1 + 1)} = 31\ 250\ Baud$$

Zde můžeme vidět, že při uložení čísla **1** do registru UBRR při frekvenci oscilátoru Atmegy 1Mhz bude rychlost komunikace nastavena na požadovaných 31 250Baudů.

Chybovost přenosu dat vypočítáme pomocí vztahu:

$$Errorr = \left( \frac{BaudRate (který\ vyšel\ při\ kontrole)}{BaudRate(požadovaný)} - 1 \right) \cdot 100 = 0\%$$

Jak jsem již dříve řekl, když číslo, které zadáváme do registru **UBRR**, vyjde bezzbytku, chybovost 0%. Tuto informaci jsem pouze potvrdil i výpočtem. Chybovost, které by byla menší než 0,5% se dá ještě tolerovat, při vyšší chybovosti bych musel změnit frekvenci oscilátoru Atmegy. Nutnost změny taktovací frekvence Atmegy by nebyla nutná, kdybych mohl použít i paritní bit (ten však MIDI sběrnice nepodporuje).

### Inicializace UART:

```
void USART_Init(void) //nastavení baud rate
{
    UBRRH = (unsigned char)(1 >> 8);           //horní bit
    UBRRL = (unsigned char)1;                 //dolní bit
    UCSRB = (1<<RXEN)|(1<<TXEN);              // povolení přijímání a
    odesílání dat
    UCSRC = (1<<URSEL)|(0<<USBS)|(3<<UCSZ0);   //formát: 8data, 1stop bit
}
```

Pro příjem dat a vysílání se používá dvou příkazů. Příjem dat se zajišťuje pomocí funkce `getchar()`. Funkce `putchar(proměnná)` odesílá data (proměnné) na UART.

### Příklad programu Příjem a Odesílání dat:

```
#include<stdlib.h>
int receive;

receive = getchar();           //uloží se přijatá z UART data do proměnné „a“
putchar(receive);             //odeslání přijatého znaku uloženého do proměnné „a“ na UART
```

## **5.2 ZPRACOVÁNÍ PŘIJATÝCH DAT**

Přijatá data jsou ve formátu čísla, typu *integer* (reálné číslo – 16 bitů). Číslo přijaté z MIDI je v rozsahu 0 – 127. Nejdříve musím zjistit, zda se jedná o stavový či datový byte. Z teorie MIDI komunikace víme, že rozdílnost těchto dvou zpráv rozlišujeme pomocí prvního bitu.

Po rozlišení stavového či datového bytu rozhodnu, zda s ním chci pracovat či ne, pokud ne, tak znovu odešlu požadavek na příjem dalšího bytu. Jedná se li o případ chtěné informace, tak ji buď uložím do proměnné, nebo provedu výpočet a následně vyhodnocení, které se projeví zobrazením zprávy na displeji.

### 5.2.1 Zjištění čísla MIDI kanálu

Jak je již výše zmíněno vždy musím rozlišovat datový a stavový bit. Víme, že o tom rozhoduje první bit MIDI bytu. Jak jsem to tedy řešil? Díky tomu, že prostředí umí pracovat s binárními čísly (i když se jedná o jazyk C), tak jsem použil maskování proměnných.

Ve stavovém bytu je číslo kanálu ve čtyřech bitech s nejnižší hodnotou

#### Maskování:

Maskování je vybrání pouze užitečných bitů, pomocí logické funkce AND (bitový součin). Nepotřebné/ostatní bity buď vynulují, nebo nastavím do jedničky. Následně zjišťuji užitečné informace.

*Legenda (pro příklad 1 i 2):*

*pom* – pomocná proměnná

*receive* – proměnná, v které je uložena hodnota z UART

*Příklad programu 1:*

```
receive = getchar();
pom = receive & 0b1000 0000;
if (pom == 0b1000 000)
    //jedná se o stavový byt
else
    //jedná se o datový byt
```

*Příklad výpočtu 1:*

$$\begin{array}{r} 1000\ 1101 \\ \underline{AND\ 1000\ 0000} \\ 1000\ 0000 \end{array}$$

Ať je číslo v *receive* jakékoliv, vždy se vybere pouze horní bit, který určuje, zda se jedná o stavový, či datový typ MIDI bytu. Pomocnou *pom* jsem použil z důvodu zachování přijatého čísla v *receive*, kdybych si ho přepsal, tak ho už nikdy nezískám zpět, jelikož se nedá zažádat o znovu zaslání čísla.

Dalším maskováním zjistím, poslední čtyři bity ve stavovém bytu, ve kterých je číslo kanálu.

*Příklad programu 2:*

```
receive = getchar();
pom = receive & 0b00001111; //maska, která vybere dolní čtyři byty
lcd_putchar(pom); /*vytisknutí čísla MIDI kanálu na displej (pouze
princiálně)*/
```

*Příklad výpočtu 2:*

$$\begin{array}{r} 1000\ 1101 \\ \underline{AND\ 0000\ 1111} \\ 0000\ 1101 \end{array}$$

Tisk čísla MIDI kanálu na displej je trochu složitější, proto uvádím jen podstatu zjištění MIDI kanálu.

## 5.2.2 Stisknutá klávesa

Funkce přijme číslo z klavíru, které následně zpracuje. Číslo prochází přes filtr, který vybírá pouze čísla not. Jak u této funkce, tak i u všech ostatních funkcí jsou napsány dva druhy filtrů, mezi nimiž se volí v závislosti na konfiguraci hudebního analyzátoru, kvůli univerzálnosti. Po uložení čísla noty do proměnné se zmenšuje tak dlouho, dokud je větší než jedenáct. Funkce, která se stará o vyhodnocení názvu umí rozlišit jen číslo od 0 do 11, tak proto se číslo zmenšuje.

*Dva filtry:*

```
receive = getchar();           //přijme znak
if(Config)
{
    pom = receive & 0b10000000;    //maska, která vybere nejvyšší bit
    if(!pom)
        d = 1;
    else
        d = 0;
}
else
{
    pom = receive & 0xF0;
    if(0x90)
        d = 1;
    else
        d = 0;
}

if(d)    //rozohdne zda se jedná o stavový (=1) nebo datový (=0) byte.... vybírá datový
{
    if(Config)
        note = receive;           //vožení čísla noty do proměnné "note"
    else
        note = getchar();
        velocity = getchar();    //vložení čísla (síla stlačení) do proměnné "velocity",
                                //toto číslo následuje okamžitě za číslem noty
    if (velocity)    //když velocity = 0, znamená to, že klávesa byla puštěna, mě
//zajímá jen stisknutí klávesy, takže analyzují jen stisky kláves (tzn. velocity ≠ 0)
    {
        while (note > 11) //zmenšuje tak dlouho, dokud je proměnná note větší než 11,
        {
            note = note - 12;    //pokaždé se zmenší proměnná note
            octave = octave + 1; //proměnná octave zvýší o jedničku, takže
//v proměnné octave je číslo, které udává, v jaké oktávě se nachází daný tón, pak
//už jen záleží na nás, zda zvolíme možnost tisknutí čísla oktávy na displej.
        }
    }
    Nazev(note,d); //funkce vyhodnocující název noty
    ...atd.
```



### 5.2.3 Analýza intervalů

Z hudební teorie víme, že se intervaly určují pomocí znalostí durových stupnic. Nejedná se o jedinou metodu, dá se i využít vypočítání počtů půltónů mezi jednotlivými tóny (i bez znalosti stupnic).

Nejdříve vyčkám na přijetí čísla noty, které si následovně uložím. Čísla dvou not jsem od sebe odečetl, podle znaménka jde poznat, zda se zahrál první nižší či vyšší tón (tuto informaci jsem však nevyhodnocoval). Rozdíl těchto dvou not je rovný počtu půltónů mezi jednotlivými tóny. Pak už jen stačilo vyhodnotit, pomocí následující tabulky, jaký byl zahráný interval.

Počet půltónů	Název	Počet půltónů	Název
0	Prima	7	Čistá kvinta
1	Malá sekunda	8	Malá sexta
2	Velká sekunda	9	Velká sexta
3	Malá tercie	7	Čistá kvinta
4	Velká tercie	10	Malá septima
5	Čistá kvarta	11	Velká septima
6	Zvětšená kvarta	12	Čistá oktáva

Tato tabulka platí pouze, zda byl interval zahrán v rámci jedné oktávy. Intervaly, které jsou přes oktávu, je nutné zmenšit o celou oktávu (nebo její násobek), aby tabulka odpovídala intervalům. Zmenšování se provádí stejně jako u funkce „*Stisknutá klávesa*“

Funkce „*Intervaly*“ mají dvě možnosti zahrání intervalů, a to melodicky a harmonicky.

- Harmonicky – jelikož čísla z klavíru přijdou bezprostředně za sebou, není problém příjem dat, jelikož hudební analyzátor předpokládá, že přijme 4 byty a hned vyhodnocuje. Odečte první notu od druhé a tím zjistí interval, pokud je číslo záporné, pracuje s absolutní hodnotou intervalu.
- Melodicky – Tentokrát hudební analyzátor nemůže čekat na 4byty a hned vyhodnocovat, nejdříve přijme číslo jedné noty a pak čeká, na to, až zahrajeme druhou notu. Po zahrání druhé noty teprve hudební analyzátor vyhodnocuje interval (klesající i vzrůstající)

#### Příjem intervalu (dvou not) harmonicky zahráného:

```
receive = getchar(); //přijme znak – zjištění první noty
if(Config) //případ pro 1. konfiguraci
{
    pom = receive & 0b10000000; //maska, která vybere nejvyšší bit
    if (pom == 0 && receive != 0) //rozohdne zda se jedná o stavový/datový...
        vybírá datový
        {
            velocity = getchar(); //vlození čísla (síla stlačení) do proměnné "velocity",
            toto číslo následuje okamžitě za číslem noty
            if(velocity)
                note1 = receive; //vození čísla první noty do proměnné "note1"
```

```

receive = getchar(); //přijme znak – zjištění druhé noty
pom = receive & 0b10000000; // maska, která vybere nejvyšší bit
if (pom == 0) //rozohdne zda se jedná o stavový/datový... vybírá datový
{
    velocity = getchar();
    if(velocity)
    {
        note2 = receive; // vožení čísla první noty do proměnné "note2"
        IAnalyze(note1, note2); //zavolání funkce, která analyzuje
        interval
    }
}

}

else //případ pro 2. konfiguraci
{
    pom = receive & 0xF0; //aplikace masky
    if(pom == 0x90) //vybírá pouze datový typ
    {
        note1 = getchar(); //uložení 1. noty
        getchar(); //přijme znak
        receive = getchar();
        pom = receive & 0xF0; //aplikace masky
        if(pom == 0x90) //vybírá pouze datový typ
        {
            note2 = getchar(); //uložení 2. noty
            getchar(); //přijme znak (jen vyprázdnění zásobníku)
            IAnalyze(note1,note2); //zavolání funkce, která analyzuje interval
        }
    }
}
}

```

Funkce „*IAnalyze(note1,note2)*“ už jen zjistí absolutní hodnotu rozdílu dvou not a vytiskne název na displej.

Zjištění melodicky zahraného intervalu je o něco málo složitější v tom, že se příjem noty musí dát do smyčky, která neustále ověřuje, zda už nepřišli informace o zahraně notě. Jakmile informace přijdou, smyčka se ukončí a pošle zjištěné data na analýzu, přesněji pošle data funkci „*IAnalyze(note1, note2)*“.

#### Vypočítání absolutní hodnoty intervalu:

```

if(note1 > note2)
    degree = note1 - note2;
else
    degree = note2 - note1;

```

## 5.2.4 Analýza Akordů

Funkce „Akordy“ zjišťuje čísla not obdobným způsobem, stejně tak, jako intervaly mezi jednotlivými tóny. Jelikož hudební analyzátor rozlišuje čtyřzvuky a trojzvuky, tak je program napsán tak, že vždy očekává 4 tóny, jakmile dostane informaci o prvním tónu, okamžitě vyžaduje druhou, třetí a čtvrtou notu. Po vyžádání informací zjistí, kolik tónů přijal, když je jich méně než 3, tak nic neanalyzuje, pokud jsou to tři tak analyzuje trojzvuk a pokud dostal všechny 4 tóny (pokud byly zahrány) tak vyhodnocuje čtyřzvuk. Podmínkou je, že akord musí být zahrán harmonicky a všechny tóny musí zaznít v jednu chvíli, jinak analyzátor neanalyzuje zahráný akord, nebo ho analyzuje špatně. Jelikož tóny byly zahrány všechny v jednu chvíli a MIDI posílá informace jednu po druhé, tak se stává, že klavír může poslat interval v náhodném pořadí tónů, proto je před analýzou musíme správně porovnat, dle výšky.

```
Funkce bubble sort: //srovnání proměnných dle velikosti
for (i = 3; i > 0; i--)
{
    for (j = 1; j <= i; j++)
    {
        if (note[j-1] > note[j])
        {
            pom = note[j-1];
            note[j-1] = note[j];
            note[j] = pom;
        }
    }
}
```

Atmega má omezenou velikost paměti, tak bylo prioritou psaní co nejúspornějšího programu, co se týče velikosti. Následkem úspor místa je složitější program, který není tolik čitelný pro programátora. Zpracování čtyř proměnnými je zbytečně složité a zabere hodně místa, proto jsem pro akordy vymyslel zakódování tří či čtyř tónů.

Způsob zakódování spočívá v tom, že se nejdříve zjistí, které číslo tónu udává název akordu. Číslo je samozřejmě uloženo a ponecháno do doby, než se použije pro analýzu. Následuje vypočítání intervalů, zde se začíná lišit zakódování druhu akordu.

### Trojzvuky:

1. Výpočet prvního intervalu, který se nachází mezi prvním a druhým tónem akordu
2. Výpočet druhého intervalu, který se nachází mezi druhým a třetím tónem akordu
3. Jednoduchá zakódování podle vzorce:  $x = \text{první\_interval} + 7 \times \text{druhý\_interval}$   
Násobící konstanta je vybrána záměrně 7, jakožto prvočíslo. Jelikož interval není větší než 5 tak proměnná „x“ bude vždy jiná, ať zahrajeme jakékoliv pořadí intervalů (neboli akordů)
4. Následuje blok podmínek, který porovná proměnnou „x“ s konstantami (v kolonce výsledek), které jsou v následující tabulce (Tab. 3 Zakódování trojzvuků), a výsledek se vytiskne na displej.
5. Pokud se proměnná „x“ neshoduje s žádnou konstantou určující trojzvuk (pořadí a velikost intervalů), tak displej zůstane prázdný.

Trojzvuky									
7	násobící konstanta								
Dur					moll				
Kvintakord		Kvart-sextakord			Kvintakord		Kvart-sextakord		
Interval	výsledek	Interval	výsledek	Interval	výsledek	Interval	výsledek	Interval	výsledek
4	25	5	33	3	31	5	26	3	26
3		4		4		3			
Sextakord		Sextakord			Sextakord		Sextakord		
Interval	výsledek	Interval	výsledek	Interval	výsledek	Interval	výsledek	Interval	výsledek
3	38	4	39	4	39	5	39	5	39
5		5							
Zmenšený					Zvětšený				
Základní		2. Obrat			Základní		Obratem zvětšeného kvintakordu vznikají enharmonicky zaměnitelné akordy		
Interval	výsledek	Interval	výsledek	Interval	výsledek				
3	24	6	27	4	32				
3		3		4					
1. Obrat		1. Obrat			1. Obrat				
Interval	výsledek	Interval	výsledek	Interval	výsledek				
3	45	6	45	6	45				
6		6							

Tab. 3 Zakódování trojzvuků

### Čtyřzvuky:

1. Výpočet prvního intervalu, který se nachází mezi prvním a druhým tónem akordu
2. Výpočet druhého intervalu, který se nachází mezi druhým a třetím tónem akordu
3. Výpočet třetího intervalu, který se nachází mezi třetím a čtvrtým tónem akordu
4. Zakódování podle vzorce:  $x = 17 \times Interval_1 + 7 \times Interval_2 + 11 \times Interval_3$   
Čísla jsem zase volil tak, aby se nestalo, že dvou akordům připadala jedna konstanta.
5. Následuje blok podmínek, který porovná proměnnou „x“ s konstantami, které jsou v následující tabulce (Tab. 4 Kódování čtyřzvuk), a výsledek se vytiskne na displej.
6. Pokud se proměnná „x“ neshoduje s žádnou konstantou určující čtyřzvuk (pořadí a velikost intervalů), tak displej zůstane prázdný.

Čtyřzvuky - septakordy										
Základní poloha										
Tvrdě velký			Tvrdě malý			Měkce malý			Změněně malý	
4	<b>133</b>		4	<b>122</b>		3	<b>112</b>		3	<b>116</b>
3			3			4			3	
4			3			3			4	
Změněně zmenšený			Měkce velký			Zvětšeně velký				
3	<b>105</b>		3	<b>123</b>		4	<b>129</b>			
3			4			4				
3			4			3				
1. Obrat										
Tvrdě velký			Tvrdě malý			Měkce malý			Změněně malý	
3	<b>90</b>		3	<b>94</b>		4	<b>111</b>		3	<b>101</b>
4			3			3			4	
1			2			2			2	
Enharmonická záměna zmenšeně zmenšeného septakordu			Měkce velký			Zvětšeně velký				
			4	<b>107</b>		4	<b>100</b>			
			4			3				
			1			1				
3. Obrat										
Tvrdě velký			Tvrdě malý			Měkce malý			Změněně malý	
4	<b>119</b>		3	<b>109</b>		3	<b>98</b>		4	<b>115</b>
1			2			2			2	
4			4			3			3	
Enharmonická záměna zmenšeně zmenšeného septakordu			Měkce velký			Zvětšeně velký				
			6	<b>142</b>		3	<b>102</b>			
			1			1				
			3			4				
2. Obrat										
Tvrdě velký			Tvrdě malý			Měkce malý			Změněně malý	
1	<b>78</b>		2	<b>95</b>		2	<b>99</b>		2	<b>88</b>
4			4			3			3	
3			3			4			3	
Enharmonická záměna zmenšeně zmenšeného septakordu			Měkce velký			Zvětšeně velký				
			1	<b>75</b>		1	<b>89</b>			
			2			4				
			4			4				

Tab. 4 Kódování čtyřzvuků

### 5.2.5 Analýza stupnic

Příjem čísel od MIDI zařízení je neustále stejný, jediné co se mění, je zpracování a ukládání různých počtů dat. U stupnic funkce přijímá osm not, které ukládá nanejvýš dva tóny, díky kterým zjišťuje jednotlivé intervaly, které se rovnou kódují. Výpočet probíhá následovně:

1. Uložení čísla prvního tónu, který zároveň určuje název stupnice
2. Uložení čísla druhého tónu
3. Odečtení prvního tónu od druhého tónu a výsledek se uloží do proměnné „x“
4. Uložení čísla třetího tónu
5. Odečtení druhého tónu od třetího tónu, výsledek výpočtu se přičte k proměnné „x“
6. ... sčítání probíhá obdobně až do zahrnutí posledního stupně celé stupnice

Výsledky se porovnávají stejně jako u akordů. Podle následující tabulky (Tab. 5 Kódování stupnic)

Stupnice										
Dur					Hramonická					
č	Intervaly	Váha	Mezi výpočet		č	Intervaly	Váha	Mezi výpočet		
1	2	3	6		1	2	3	6		
2	2	5	10		2	1	5	5		
3	1	7	7		3	2	7	14		
4	2	9	18		4	2	9	18		
5	2	11	22		5	1	11	11		
6	2	1	2		6	3	1	3		
7	1	15	15		7	1	15	15		
Výsledek			<b>80</b>		Výsledek			<b>72</b>		
Aioslká					Melodická					
č	Intervaly	Váha	Mezi výpočet		č	Intervaly	Váha	Mezi výpočet		
1	2	3	6		1	2	3	6		
2	1	5	5		2	1	5	5		
3	2	7	14		3	2	7	14		
4	2	9	18		4	2	9	18		
5	1	11	11		5	2	11	22		
6	2	1	2		6	2	1	2		
7	2	15	30		7	1	15	15		
Výsledek			<b>86</b>		Výsledek			<b>82</b>		

Tab. 5 Kódování stupnic

## 5.2.6 Zjištění konfigurace

Konfigurace probíhá pouze ve dvou bodech, a to stisknutí a puštění klávesy. Některé digitální hudební nástroje odesílají trochu jinak oznámení o puštěné klávese.

### První případ:

Klavír při stisknutí klávesy pošle stavový byt (= informace o stisknuté klávese):

1	0	0	1	N	N	N	N
---	---	---	---	---	---	---	---

Za stavovým bytem následují dva datové byty (s číslem noty a se silou stlačení).

Při puštění se pošle jeden stavový byt (= informace o puštění klávesy), který je odlišný od stavového bytu stisknuté klávesy pouze v horním nibblu:

1	0	0	0	N	N	N	N
---	---	---	---	---	---	---	---

Za stavovým bytem následují dva datové byty (s číslem noty a se silou stlačení). Tento protokol je shodný s obvyklým protokolem MIDI

### Druhý případ:

Tato úprava obvyklého MIDI protokolu vznikla se záměrem redukovat množství posílaných dat. Klavír při stisknutí klávesy pošle stavový byt (= informace o stisknuté klávese):

1	0	0	1	N	N	N	N
---	---	---	---	---	---	---	---

Za stavovým bytem následují dva datové byty (s číslem noty a se silou stlačení). Doposud protokol je stejný, právě v následující zprávě je změna. Místo poslání stavového bytu se pouze pošle datový byte = číslo noty, které se pustila s dynamikou nula (2. Stavový byte). Touhle redukcí jsme dosáhli úspory jednoho bytu. Při dalším stisku klávesy a puštění už to budou dohromady dva byty a tak dál.

Funkce „*Konfigurace*“ přijme informaci o stisku klávesy, které není tolik důležitá. Po přijetí MIDI zprávy o puštění klávesy hudební analyzátor rozezná, zda dostal zprávu datového bytu, který začíná kombinací 1 0 0 0 (= datový byte puštění klávesy). Pokud žádný takový byte nepřijme, tak jen zjistí, zda se opravdu pustila klávesa, nastaví hodnotu proměnné na „1“ a uloží do EEPROM, aby i po restartování zařízení si stále Atmega pamatovala předešlou konfiguraci. V opačném případě hudební analyzátor uloží do EEPROM hodnotu „0“ a hudební analyzátor bude vždy očekávat datový byte oznamující puštění klávesy. Tímto je konfigurace úplná. Funkce automaticky odejde do rozcestníku „*MIDI Analýza*“.

Dva již popsané případy MIDI protokolu počítají s tím, že MIDI zařízení neustále posílá na sběrnici průběžný stav (logický stav „1“). V případě použití virtuálního pianu na počítači a převodníku z USB na MIDI se tento průběžný stav neodesílá. Kvůli tomu hudební analyzátor nefunguje tak jak by měl. Možná byste očekávali, že je to chyba, ale právě naopak, zásadně nesouhlasím s použitím hudebního analyzátoru a virtuálního pianu. Důvody jsou jednoduché, hudební analyzátor by ztrácel svoji účinnost, pro kterou byl vytvořen. Chci, aby se s hudebním analyzátozem hrálo na opravdový hudební nástroj, a to kvůli zlepšení nejen teoretických schopností, ale hlavně praktických.

## 5.2.7 Přehrávání intervalů

První funkcí, které využívá i odesílání dat do digitálního hudebního nástroje je „Intervaly“. Funkci najdeme v rozcestníku „*Trénink sluchu*“, kde i ostatní funkce využívají jak příjem, tak odesílání dat. Jak už víme, tak funkce má 12 úrovní. V závislosti na úrovni obtížnosti funkce „Intervaly“ zavolá funkci, která losuje mezi čísli, které představují intervaly. Losování se zahájí stiskem tlačítka *Enter* a ukončí se puštěním tlačítka.

### Program losování:

```
void Interval_Counter(void)
{
    while(Enter)           //dokud není stisknuté tlačítko Enter
    {
        getchar();        //vyprázdnění zásobníku pro příjem dat z USART
        if(!Left)         //když se stiskne tlačítko Vlevo
            break;        //zajištění vyskočení ze smyčky
    }
    while(!Enter)         //po dobu držení tlačítka Enter
    {
        for(degree = 0; degree <= velocity; degree++) //zvětšování čísla v závislosti na
                                                         úrovní obtížnosti
        {
            ++note[2];    //využití pro přehrání intervalu od náhodného tónu
            if(!Left)     //když se stiskne tlačítko Vlevo
                break;    //zajištění vyskočení ze smyčky
        }
    }
}
```

Běh programu následuje vrácení zpět na funkci „Intervaly“, která vyhodnotí vylosované číslo a přidá k vylosovaným číslům konstantu, která určí v závislosti na úrovni obtížnosti, zda se bude hrát interval harmonicky nebo melodicky a od kterého tónu.

Funkce „Intervaly“ zavolá další funkce, která se postará o odeslání dat do digitálního hudebního nástroje. V průběhu odesílání dat hudební nástroj hned přehrává získané data.

Na displeji se vytiskne zpráva, abychom přehráli interval. Hned na to se volá již popsaná stejnojmenná funkce, která je v rozcestníku „*MIDI Analýza*“. Tudiž hudební analyzátor čeká na příjem dvou not. Jakmile zahrajeme interval, funkce v rozcestníku „*MIDI Analýza*“ pošle data funkci, která je v rozcestníku „*Trénink sluchu*“, tato funkce porovná vylosovaný interval s intervalem, který byl zahrán na hudební nástroj. V závislosti na shodnosti se vytiskne na displej zpráva, která oznámí správnost zahráného intervalu. Posledním krokem této funkce je vytisknutí na displeji názvu intervalu, který byl zahrán hudebním analyzátozem.



## 5.2.8 Přehrávání akordů

Funkce přehrávání akordů je velmi podobná funkci *Intervaly*, rozdíl je pouze v množství zpracovávaných tónů (intervalů). Tato funkce zabírá relativně hodně paměti, jelikož je na výběr mezi 32 úrovněmi obtížnosti. Losování náhodných akordů zase určuje stisk tlačítka *Enter*. Tentokrát je losovacích funkcí víc, přesněji 8 (u funkce „*Intervaly*“ jsou pouze dvě). Když máme vylosovaný akord, funkce zavolá jinou funkci, které přehraje akord, a program se vrátí k předešlé funkci.

Na displej se vypíše, ať zahrajeme akord (který jsme slyšeli), tudíž hudební analyzátor čeká na příjem dat. Příjem dat má v režii zase jiná funkce, která již byla popsána v rozcestníku „*MIDI Analýza*“. Zahrajeme-li akord, běh programu se vrátí k funkci „*Akordy*“ v rozcestníku „*Trénink sluchu*“ a porovná námi zahráný akord a akord, který byl zahrán hudebním analyzátozem. Na displej se vypíše, zda jsme správně sluchově rozeznali akord a pro naši informaci vypíše název akordu, který byl přehrán hudebním analyzátozem.

Příklad generování akordů v závislosti na úrovni obtížnosti:

```
switch(level)
{
    case 1: //případ pro první úroveň
        Akord_Counter_Dur_moll(); // vygeneruje intervaly pro Dur či moll v
závislosti na době držení tlačítka Enter
        receive = 500; //doba mezi 1. a 2. zahrnou notou v milisekundách
        note[2] = 0x3C; // číslo noty C = hraní jakéhokoliv akordu od C
        Trojzvuk_Play(); //Funkce přehrávající trojzvuky
        break;

    case 2: //případ pro druhou úroveň
        Akord_Counter_Dur_moll(); // vygeneruje intervaly pro Dur či moll v
závislosti na době držení tlačítka Enter
        receive = 1; //konstanta pro harmonicky znějící akord
        note[2] = 0x3C; // číslo noty C
        Trojzvuk_Play(); //Funkce přehrávající trojzvuky
        break;

    case 3: //případ pro třetí úroveň
        Akord_Counter_Dur_moll(); // vygeneruje intervaly pro Dur či moll v
závislosti na době držení tlačítka Enter
        receive = 500; //konstanta pro melodicky znějící akord
        Interval_Norm(); //Změní rozsha proměnné note[2] od 60 do 72, to
má za příčinu výběr mezi tóny  $c_1 - c_2$ .

        Trojzvuk_Play();
        break; //Funkce přehrávající trojzvuky

//obdobně pokračují případy a po úroveň 32
```

## 5.2.9 Přehrávání stupnic

V této funkci najdeme pouze dvě úrovně a to z důvodů, že se jedná o tři různé stupnice. Mohlo by se zdát, že funkce nebude tak náročná na paměť ale opak je pravdou. Každá stupnice má osm stupňů, a sedm mezistupňů. To znamená, že se musí losovat mezi třemi různě uspořádanými sedmi mezistupni. Jinak zbytek funkce je podobné jako akordy, či intervaly. Také se znovu využívá funkce „Stupnice“, která je v rozcestníku „MIDI Analýza“.

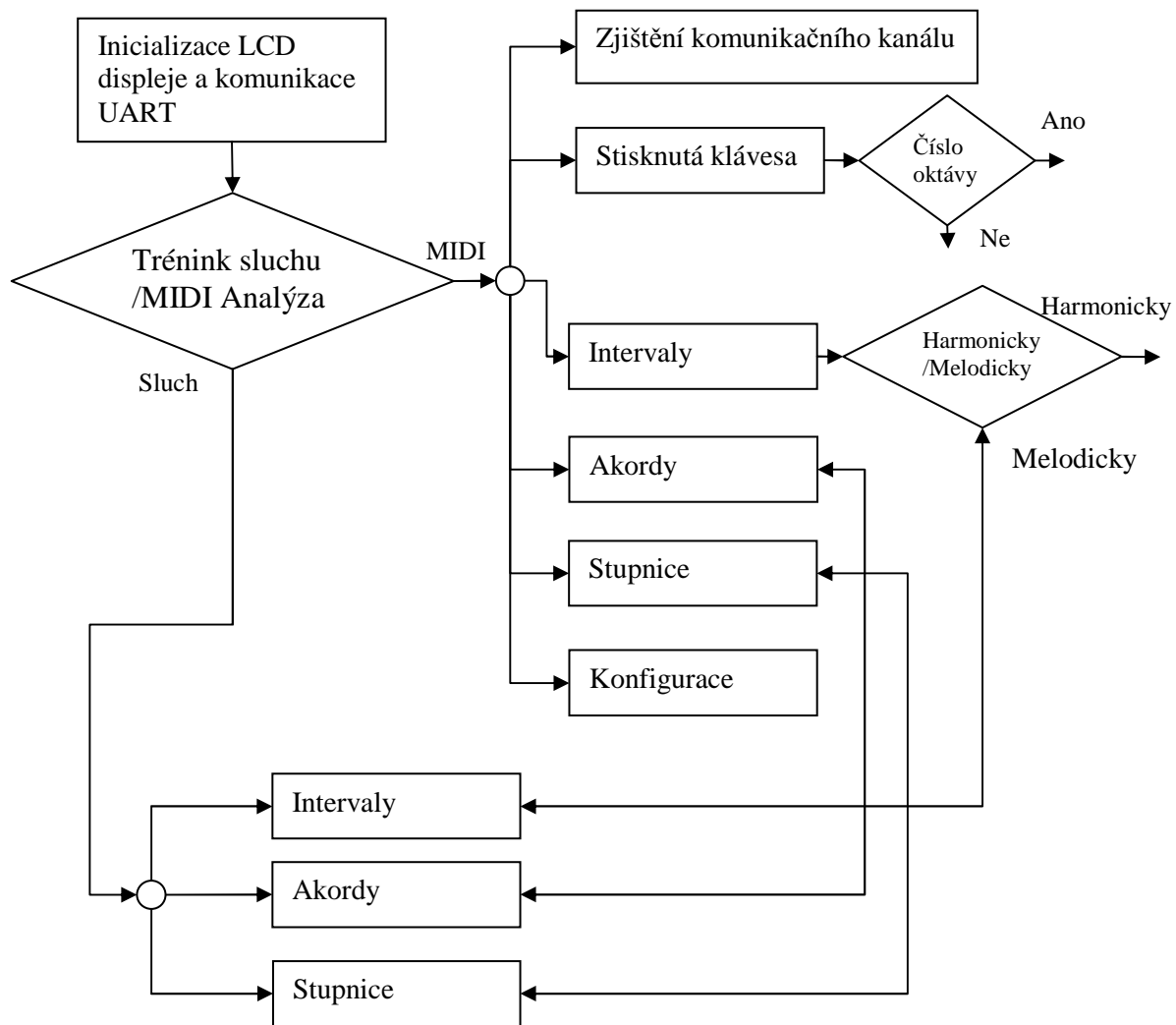
Část programu přehrávající stupnice:

```
void Stupnice_Play()
{
    putchar(0x90); //příkaz posílající číslo 90(v hex) do hudebního zařízení
    for(degree = 0; degree <= 7; degree++) //smyčka která přehrává jednotlivé
    stupně = program tuto smyčku vykoná osmkrát
    {
        putchar(0x90); //stavový byte – nota stisknuta
        putchar(note[2]); //datový byte – číslo noty
        putchar(Volume); // síla stlačení klávesy
        delay_ms(500); //doba ticha v ms, mezi jednotlivými stupni
        putchar(0x80); //stavový byte – nota puštěna
        putchar(note[2]); //datový byte – číslo noty
        putchar(0); // síla stlačení klávesy
        note[2] = note[2] + deg[degree]; //zvýšení další noty o určitý interval
    }
}
```

Část programu, který rozhoduje o názvu stupnice a jejího druhu:

```
if(deg[7] == 0) //číslo, které pro mě znamená Durovou stupnici
{
    if(Main_note == 3 || Main_note == 8 || Main_note == 10) //čísla tónů, které nemůžou
    být alterované pomocí křížku, jelikož by vznikla například stupnice Dis-Dur, taková stupnice,
    ani akord neexistuje. Proto názvu se musí docílit snížením vyššího tónu béčkem, tudíž se jedná
    stupnici Es- Dur
    Nazev(Main_note,2); //když se jedná o jednu z těchto jmen akordů, které se musí
    zmenšovat, tak se funkci „Nazev“ předá parametr2, který zajistí ve funkci snížení tónu.
    else
    Nazev(Main_note,0); // jinak vypíše název noty popřípadě vyjádřený křížkem.
    lcd_putse(Dur); //nakonec se vypíše z EEPROM druh stupnici – „Dur“
}
```

### 5.3 VÝVOJOVÝ DIAGRAM PROGRAMU



Obr. 23 Zjednodušený vývojový diagram

## **6 RESUMÉ**

### ***6.1 V ČEKÉM JAZYCE***

Původně jsem si nekladl, tak velké cíle, aby hudební analyzátor měl tolik funkcí, ale postupně se program začal rozrůstat, až jsem se dostal do fáze, kdy jsem začal být omezován velikostí paměti Atmega8. Když mě poprvé Atmega8 ohlásila, že má nedostatek paměti, tak jsem začal pracovat na redukci programu, aby nezabíral tolik místa, ale měl pořád stejné funkce. Nyní jsem skončil s programem na řádku 2870. Program je plnohodnotný a splňuje více než moje očekávání. Doufám, že pomůže zkvalitnit a zpříjemnit výuku hudební teorie.

### ***6.2 V ANGLICKÉM JAZYCE***

Initially I didn't make so large aim, to musical analyzer has so many function, but gradual program began expand to the point, when I began to be limited by memory size Atmega8. When Atmega8 reported me that she has low memory, so I began to work on reducing the program to didn't occupy so much memory space, but had always same function. I finished program on row 2870. The program is full-value and more than meets my expectations. I hope, it will help improve the quality and enjoyable education musical theory.

Soupis použité literatury:

[1] <http://www.forummidit.wz.cz/dokumenty/MIDI1.pdf>

Seznam obrázků:

Obr. 1 Struktura stavového a datového bytu protokolu MIDI .....	5
Obr. 2 Struktura MIDI zprávy <i>Note On</i> .....	6
Obr. 3 Struktura MIDI zprávy <i>Note Off</i> .....	7
Obr. 4 Struktura MIDI zprávy <i>All Note Off</i> .....	7
Obr. 5 Průběh MIDI signálu .....	8
Obr. 6 Popis hudebního analyzátoru .....	9
Obr. 7 Schéma DIN5 konektoru Obr. 8 DIN5 konektor .....	10
Obr. 9 Propojení MIDI zařízení s hudebním analyzátozem .....	10
Obr. 10 Připojení napájení .....	11
Obr. 11 Úvodní obrazovka .....	11
Obr. 12 Ukázka aplikování funkce „Stisklá klávesa“ .....	14
Obr. 13 Přepis názvů not do notové osnovy .....	14
Obr. 14 Intervaly v notové osnově .....	15
Obr. 15 Analýza akordu .....	17
Obr. 16 Displej - zobrazení obratu .....	17
Obr. 17 první obraty akordů .....	18
Obr. 18 Analýza stupnice .....	18
Obr. 19 Schéma hudebního analyzátoru .....	23
Obr. 20 Kliše hudebního analyzátoru .....	24
Obr. 21 Osazovací schéma .....	25
Obr. 22 Zrealizovaný hudební analyzátor .....	26
Obr. 23 Zjednodušený vývojový diagram .....	42

Seznam tabulek:

Tab. 1 MIDI rychlostní data .....	6
Tab. 2: MIDI označení not .....	7
Tab. 3 Zakódování trojzvuků .....	35
Tab. 4 Kódování čtyřzvuků .....	36
Tab. 5 Kódování stupnic .....	37

## Příloha č. 1

# 1 NEZBYTNÁ HUDEBNÍ TEORIE

## 1.1 ZVUK A TÓN

Zvuky jsou vše, co slyšíme. Vznikají chvěním hmoty. Chvějící se těleso rozechvívá okolní vzduch a ten rozechvívá sluchové ústrojí v našem uchu – toto chvění v uchu vnímáme jako zvuk.

Zvuk dělíme na tóny a hluky. Některá tělesa kmitají stále stejně rychle, chvějí se pravidelně, jiná kmitají nestejně rychle, nepravidelně. Tóny vznikají pravidelným chvěním / hluky nepravidelným chvěním hmoty.

Tóny mají určitou přesnou výšku, kterou můžeme napodobit zpěvem nebo na hudebním nástroji. Slyšíme-li například nějaký tón hraný na housle, můžeme jej též zazpívat nebo zahrát na klavíru nebo na kytáře. Tyto tony se budou vzájemně lišit barvou, ale budou mít stejnou výšku.

Hluky jsou všechny ostatní zvuky kromě tonů, například rány, šumy, praskání, skřípaní, syčení, šramot. Mají výšku neurčitou, můžeme sice někdy přibližně určit, zda znějí vysoko, hluboko nebo ve střední poloze, ale nemůžeme jejich výšku stanovit tak přesně jako u tonů, a proto ji také nemůžeme zazpívat ani zahrát na hudební nástroj.

V hudbě užíváme především tonů. Jsou to tony hudebních nástrojů a lidského hlasu (zpěvu). Kromě tonů se však v hudbě uplatňují i některé hluky, například zvuk malého bubínku, činelů nebo trianglu.

## 1.2 VLASTNOSTI TÓNŮ

Základní vlastnosti tónů jsou čtyři: **délka**, **síla** (hlasitost tonu), **barva** (témbr tónu) a **výška**.

Slovo tón se v hudbě užívá v několikerém významu, např. jako znějící tón (akustický jev), nebo prvek tónové soustavy (Základní tón, zvýšený tón), interval (celý tón, půltón).

## 1.3 TÓNOVÁ SOUSTAVA, JMÉNA TÓNŮ, NOTOVÉ PÍSMO

Tónová soustava je přehledné uspořádání všech tónů, užívaných v hudbě, podle jejich výšek. Tónová soustava si nevšímá ani délky, ani síly nebo barvy tónu, ale pouze jejich výšky.

Základem naší tónové soustavy je sedm tónů, které se jmenují „c, d, e, f, g, a, h“. Těchto sedm tónů se v tónové soustavě několikrát opakuje v různých výškových polohách. Souhrnně se nazývají Základní Tónová řada.

Od výchozího tónu c k nejbližšímu opakovanému c nachází osm stupňů: 1. c, 2.d, 3.e, 4.f, 5.g, 6.a, 7.h, 8.c. Vzdálenost mezi těmito dvěma c se proto nazývá oktáva.

Tónová soustava obsahuje celkem devět oktáv, z nichž každá má své jméno: subkonta oktáva C2, kontra C1, velká C, mala c, jednočárkovaná c1 nebo c´ až pětičárkovaná oktáva c5 nebo c''''''.

Tónová soustava evropské hudby – rozdělení tonů do oktáv:

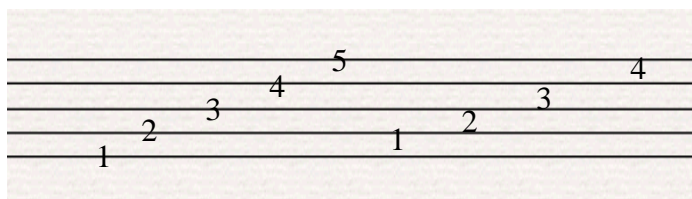
1. Subkontra oktáva						A <sub>2</sub>	H <sub>2</sub>
2. Kontra oktáva	C <sub>1</sub>	D <sub>1</sub>	E <sub>1</sub>	F <sub>1</sub>	G <sub>1</sub>	A <sub>1</sub>	H <sub>1</sub>
3. Velká oktáva	C	D	E	F	G	A	H
4. Malá oktáva	c	d	e	f	g	a	h
5. Jednočárkovaná oktáva	c <sup>1</sup>	d <sup>1</sup>	e <sup>1</sup>	f <sup>1</sup>	g <sup>1</sup>	a <sup>1</sup>	h <sup>1</sup>
6. Dvoučárkovaná oktáva	c <sup>2</sup>	d <sup>2</sup>	e <sup>2</sup>	f <sup>2</sup>	g <sup>2</sup>	a <sup>2</sup>	h <sup>2</sup>
7. Tříčárkovaná oktáva	c <sup>3</sup>	d <sup>3</sup>	e <sup>3</sup>	f <sup>3</sup>	g <sup>3</sup>	a <sup>3</sup>	h <sup>3</sup>
8. Čtyřčárkovaná oktáva	c <sup>4</sup>	d <sup>4</sup>	e <sup>4</sup>	f <sup>4</sup>	g <sup>4</sup>	a <sup>4</sup>	h <sup>4</sup>
9. Pětáčárkovaná oktáva	c <sup>5</sup>						

Tab. 6: Tónová soustava – evropské rozdělení tonů do oktáv

Notové písmo

Značka pro tón je *nota*, která je jakýmsi „hudebním písmenkem“. Nota může mít různou podobu. Má vždy hlavičku (buď vyplněnou černě, nebo bílou, nevyplněnou). Někdy je k ní připojena *nožka* (svislá čárka, směřující buď po levé straně hlavičky dolů, nebo po pravé straně nahoru).

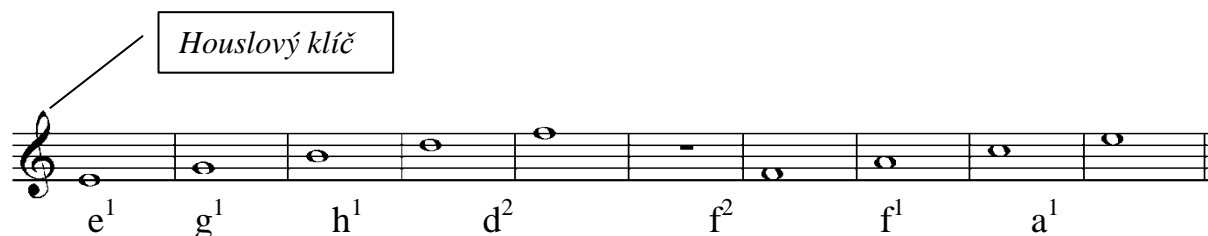
Noty píšeme na *notovou osnovu*, která se skládá z pěti linek a čtyř mezer. Počítáme je zdola nahoru.



Obr. 24 Notová osnova

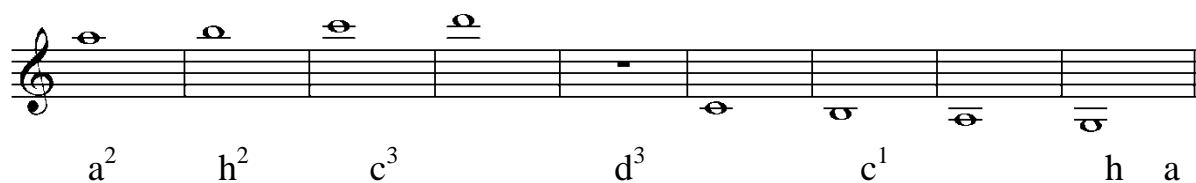
Výška tónu udává klíč a poloha noty na notové osnově. *G klíč* (čili soulový) určitě, že n druhé lince se píše nota jednočárkované g.

Noty na linkách a v mezerách



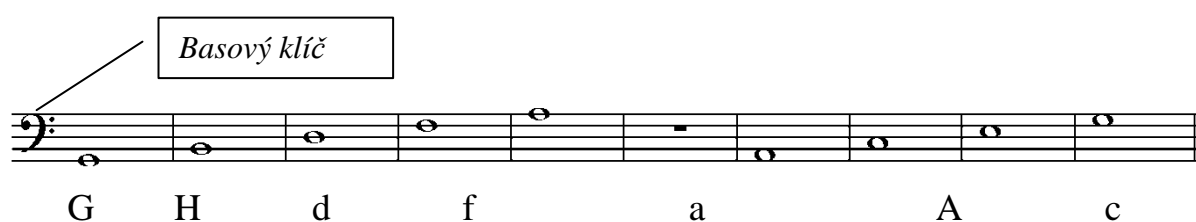
Obr. 25 G klíč

Další noty píšeme na *pomocných linkách* nad i pod osnovou



Obr. 26 Noty na pomocných linkách

*Klíč basový* dává notám nová jména. Píše se na 4. Linku a určuje, že nota na 4. Lince je malé *f*. Proto mu říkáme také *f klíč*. Návčik jako u klíče houslového.



Obr. 27 *f* klíč

#### 1.4 TÓNY ZVÝŠENÉ A SNÍŽENÉ

Každý tón naší tónové soustavy můžeme jednou nebo dvakrát zvýšit nebo snížit. Jednoduché zvýšení označujeme v názvu tónu příponou – is, dvojitě zvýšení příponou – isis (snížení jednoduché –es, dvojitě –eses ).

Jen v některých případech používáme názvů odlišných: es, as, asas. Místo názvu hes se v praxi častěji podle dávné tradice užívá názvu b.

Tóny c, d, e, f, g, a, h se považují za základní, tóny zvýšené a snížené se souhrnně nazývají tóny odvozené nebo alterované. *Snižování a zvyšování tónů se pak též řeka alterace.*

*Tóny základní a odvozené můžeme shrnout takto:*

Tóny dvakrát zvýšené	cisis	disis	eisis	fisis	gisis	aisis	hisis
Tóny jedno zvýšené	cis	dis	eis	fis	gis	ais	his
<b>Tóny základní</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>a</b>	<b>h</b>
Tóny jednou snížené	ces	des	es	fes	ges	as	b (hes)
Tóny dvakrát snížené	ceses	deses	eses	feses	geses	asas	heses

Tab. 7: Základní a odvozené tóny

#### 1.5 CELÝ TÓN A PŮLTÓN, TÓNY ENHARMONICKÉ

Základních a odvozených tónů je v rámci oktávy 35. V oktávě však rozlišujeme pouze 12 různých tónových výšek.



<b>K L A V I A T U R A</b>			h i s i s c i s d e s	d i s e s f e s e s		e i s i s f i s g e s	g i s a s	a i s b c e s e s		h i s i s c i s d e s		
		aisis	his	cisis	disis	eis	fisis	gisis	aisis	his	cisis	
		<b>h</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>a</b>	<b>h</b>	<b>c</b>	<b>d</b>	
	ces	deses	eses	fes	geses	asas	heses	ces	deses	eses		

Obr. 28 Klaviatura - Celý tón a půltón, tóny enharmonické

Nejmenší vzdálenost ve výšce dvou tónů užívaná v naší hudbě se nazývá půltón. Oktáva sestává z 12 půltónů. Dva půltóny tvoří celý tón. Jednoduché zvýšení nebo snížení výšky představuje vždy posun o půltón. Tóny, které mají stejnou výšku, ale různý název, se jmenují tóny enharmonické. Enharmonická záměna znamená záměnu daného tónu za tón stejné výšky, ale jiného jména.

## 1.6 STUPNICE

### 1.6.1 Stupnice a tónina

Stupnice je stoupající nebo klesající řada tónů v rozmezí jedné oktávy, uspořádaná podle určitých pravidel. Tato pravidla se týkají především počtu tónů v oktávě a vzdálenosti mezi jednotlivými stupni stupnice.

Rozlišujeme stupnici a tóninu. Mají společný základní tón a podle něho se také tóniny stejně jako stupnice jmenují. Tento základní tón se v tónině nazývá tónika. Tónina je volné pořadí tónů stupnice v hudbě, v určité melodii se ani nemusí vyskytovat všechny tóny příslušně stupnice, ale tóny musí být uspořádaný tak, aby tónika vynikala jako tón hlavní. Tónikou velmi často melodie začíná nebo se k ní vrací, nejnápadněji se však tónika uplatňuje v závěru melodie. Melodie končící tónikou působí jako přesvědčivě uzavřena, takže nepocítujeme nutnost dalšího pokračování. Můžeme též říci, že stupnice sestává z tónů tóniny, seřazených za sebou podle výšky, a že začíná a končí tónikou. Stupně tóniny se označují pořadovými čísly podle svého umístění ve stupnici. Tónika je první stupeň a jí jsou blízké třetí a pátý stupeň. Ostatní stupně v tónině se od tóniky odlišují výrazněji, zvláště 7. stupeň, citlivý tón. Tento tón v melodii velmi často tíhne k dalšímu pokračování k tónice. Říkáme, že citlivý tón se rozvádí do tóniky.

## 1.6.2 Durové stupnice

Na začátku má velkou tercii – rozpětí 2 celé tóny názvy durových stupnic píšeme vždy s velkým písmenem. Durová stupnice má osm stupňů, ale pouze sedm stupňů různých, proto se řadí mezi sedmistupňové stupnice. Vzdálenosti mezi stupni durové stupnice jsou celotónově a půltónově, v tomto pořadí:

1            1            ½            1            1            1            ½

Rozlišujeme stupnice s křížky a s béčky. Durová stupnice, v níž by se současně vyskytovaly křížky i béčka, neexistuje. Stupnice řadíme za sebou tak, aby postupně přibývalo po jednom # nebo po jednom b. Souhrn křížků nebo béček, které každá stupnice vlastní, se nazývá předznamenání stupnice.

... s křížky

O stupnicích s křížky platí toto pravidlo: další durové stupnice s křížky stavíme na pátém stupni stupnice předešlé a zvyšujeme sedmy stupeň.

Pořadí stupnic podle přibývajících křížků: C, G, D, A, E, H, Fis, Cis

Pořadí přibývajících křížků: fis, cis, gis, dis, asi, eis, his

... s béčky

Durové stupnice s by stavíme na čtvrtém stupni stupnice předešlé a snižujeme vždy čtvrtý stupeň.

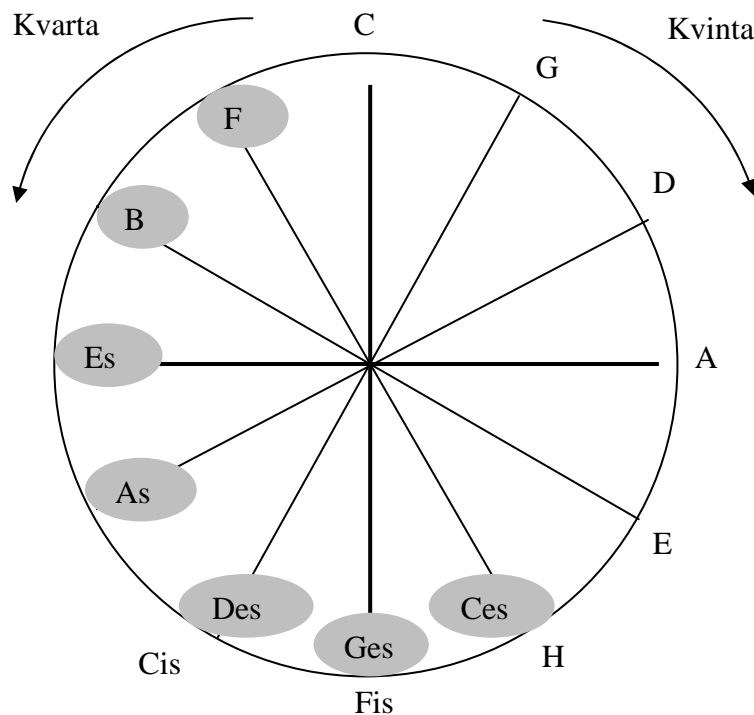
Pořadí stupnic podle přibývajících béček: F, B, Es, As, Des, Ges, Ces

Pořadí přibývajících béček: b, es, as, des, ges, ces, fes

## 1.6.3 Kvintový a kvartový kruh

Vzdálenost (interval) mezi prvním a pátým stupněm v durové stupnici je kvinta, mezi prvním a čtvrtým je kvarta. Můžeme říci, že durové stupnice s křížky postupují po kvintách a durové stupnice s béčky po kvartách. Durových stupnic s předznamenáním je i se stupnicí C dur celkem 15, v oktávě se však nalezu 12 různých výšek. Z toho plyne, že některé stupnice musí být enharmonické (stejně znějí, ale jinak se jmenují).

Kvarty a kvinty jsou vzájemnými převraty, a proto řada kvint čtena pozpátku dává řadu kvart.



Obr. 29 Kvart-Kvintový kruh

#### 1.6.4 Mollové stupnice přirozené

Na začátku má malou tercii – rozpětí 1 a  $\frac{1}{2}$  celého tónu  
 Rozlišujeme tři druhy mollových stupnic:

- Přirozené (Aiolské)
- Harmonické
- Melodické

Základní řada přirozené mollové stupnice: a, h, c, d, e, f, g, a

Vzdálenosti mezi stupni mollové stupnice jsou celotónové a půltónové, v tomto pořadí:

1             $\frac{1}{2}$             1            1             $\frac{1}{2}$             1            1

Názvy mollových stupnic píšeme malými písmeny. Východiskem pro mollové stupnice je stupnice a moll. Přirozeně mollové stupnice začínají na šestém stupni paralelní stupnice durové.

- #

mollové stupnice s křížky stavíme na pátém stupni stupnice předešle a v nové stupnici zvyšujeme vždy druhý stupeň

- b

mollové stupnice s béčky stavíme na čtvrtém stupni stupnice předešlé a snižujeme vždy šestý stupeň.

### 1.6.5 Stejnojmenné durové a mollové stupnice

	1.	2.	3.	4.	5.	6.	7.	8.
A dur	a	h	<i>cis</i>	d	e	<i>fis</i>	<i>gis</i>	a
a moll	a	h	<i>c</i>	d	e	<i>f</i>	<i>g</i>	a

Tab. 8: Rozdílné stupně stejnojmenných stupnic

Stejnojmenné durová a mollová přirozená stupnice mají společné stupně 1., 2., 4., 5., a 8. Liší se stupněm třetím, tím se také především odlišuje tónorod durový a mollový. Předznamenání stejnojmenných stupnic durových a mollových se liší vždy o tři posuvky, mollové stupnice mají vždy o tři béčka více nebo o tři křížky méně, než jejich stejnojmenné stupnice durové.

Durová stupnice		Mollové stupnice	
		7#	asi
		6#	dis
		5#	gis
7#	Cis	4#	cis
6#	Fis	3#	fis
5#	H	2#	h
4#	E	1#	e
3#	A	0	a
2#	D	1b	d
1#	G	2b	g
0	C	3b	c
1b	F	4b	f
2b	B	5b	b
3b	Es	6b	es
4b	As	7b	as
5b	Des		
6b	Ges		
7b	Ces		

Tab. 9: Stejnojmenné stupnice

### 1.6.6 Mollové stupnice Harmonické a Melodické

V harmonické stupnici mollové se zvyšuje 7. stupeň. Původní celotónová vzdálenost 7. a 8. stupněm v přirozené mollové stupnici se v harmonické mění na půltónovou. Sedmý stupeň v mollové stupnici se tak shoduje se sedmým stupněm v durové stupnici a to je důležité zvláště pro harmonii.

Mollová stupnice harmonická vznikne ze stupnice mollové zvýšením sedmého stupně.

<b>Harmonická</b>	1.	2.	3.	4.	5.	6.	7.	8.
A dur	a	h	<i>cis</i>	d	e	<i>fis</i>	<i>gis</i>	a
A moll	a	h	<i>c</i>	d	e	<i>f</i>	<i>gis</i>	a
Intervaly		1	½	1	1	½	1½	½

Tab. 10: Porovnání durové a harmonické mollové stupnice

Původní celotónová vzdálenost mezi 6. a 7. stupněm (f-g) se rozšířila na vzdálenost jednoho a půl tonu (f-gis). Tento interval se nazývá zvětšená sekunda a pokládá se obvykle za nezpěvný krok a melodie se mu kromě zvláštních případů vyhýbají.

Nezpěvný krok mezi 6. a zvýšeným 7. stupněm v harmonické moll můžeme odstranit zvýšením šestého stupně v moll. Vzdálenost se tím zkrátí na celotónovou. Protože se tak děje z melodických důvodů, nazývá se takto upravená mollová stupnice melodická.

Mollová stupnice melodická vznikne ze stupnice mollové harmonické zvýšením šestého stupně.

<b>Melodická</b>	1.	2.	3.	4.	5.	6.	7.	8.
A dur	a	h	<i>cis</i>	d	e	<i>fis</i>	<i>gis</i>	a
A moll	a	h	<i>c</i>	d	e	<i>fis</i>	<i>gis</i>	a
Intervaly		1	½	1	1	2	1	½

Tab. 11: Porovnání durové a melodické mollové stupnice

Protože v klesající stupnici není zvýšení 7. stupně potřebné, pouští se od něj, a tím se stává zbytečné i zvyšování 6. stupně. Klesající melodická mollová stupnice se pak zcela shoduje s přirozenou.

## 1.7 INTERVALY

Interval znamená v hudbě výškovou vzdálenost mezi dvěma libovolnými tóny. V názvech intervalů označujeme postupně každý tón diatonické řady latinskou číslovkou:

Prima	Sekunda	Tercie	Kvarta	Kvinta	Sexta	Septima	Oktáva	Nóna	Decima	Undecima	Duodecima	Tercdecima	Kvartdecima	Kvintdecima
C	D	E	F	G	A	H	c	d	e	f	g	a	h	c'
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

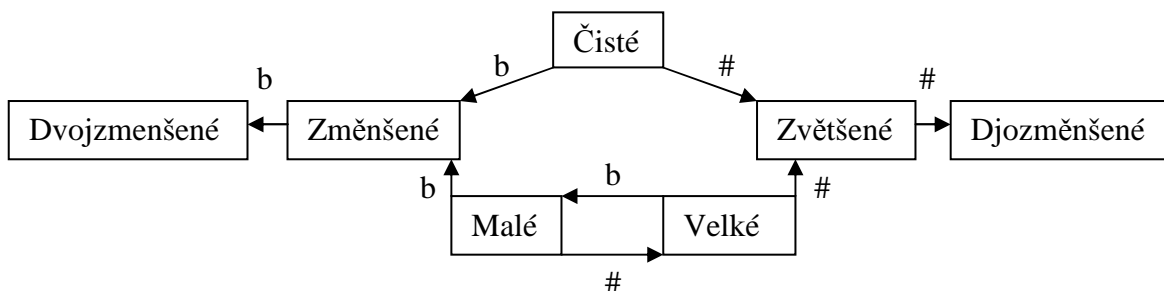
Tab. 12: Názvy základních intervalů

Latinských číslovek užíváme nejen k označení intervalů, ale též jako názvů jednotlivých stupňů ve stupnici.

Při číslování může být východiskem kterýkoliv tón. Máme-li stanovit název intervalu, musíme vymezeny prostor mezi dvěma tóny vyplnit všemi vynechanými diatonickými tóny. Jejich počet, včetně obou okrajových, pak určuje příslušný název. Název intervalu sestává

vždy ze dvou slov. Jedno slovo je číslovka, která určuje velikost intervalu přibližně, druhé slovo je přídatně jméno, které určení intervalu upřesňuje.

Intervaly třídíme též na základní a odvozené. Základní intervaly členíme rovněž do dvou skupin, na intervaly velké a malé. Odvozené intervaly vytváříme z intervalů základních rozšiřováním nebo zužováním po půltónech. Čisté intervaly mohou pak být zvětšené nebo zmenšené, velké mohou být zvětšené, malé nebo zmenšené a všechny tyto intervaly mohou být dvojnásobně zvětšené nebo dvojnásobně zmenšené.



Obr. 30 Odvozené intervaly

Intervalová nomenklatura vznikla na základě diatonické soustavy, v níž vzdálenost sousedních tónů je jednou půltónem, jindy celým tónem. Tento dvojnásobný rozměr zdělila i základní jednotka intervalů. Chceme-li míru výškového odstupů vyjádřit přesně, nestačí jednoduchý název nebo jednoduché číselné označení, je nezbytné k němu připojit další údaj.

Pro přesné určení intervalu máme několik termínů:

čistý, velký, malý, zvětšený, zmenšený, dvojnásobně zvětšený, dvojnásobně zmenšený

1. Čistým intervalem může být jedině:

- 1, prima - nulová vzdálenost mezi tóny stejné výšky
- 4, kvarta - rozpětí 2 a ½ celého tónu
- 5, kvinta - rozpětí 3 a ½ celého tónu
- 8, oktáva - rozpětí 6 celých tónů

Tyto intervaly nemohou být velkými ani malými!

2. Velkým nebo malým intervalem může být jedině:

- 2, sekunda
  - malá - rozpětí ½ celého tónu
  - velká - rozpětí 1 celý tón
- 3, tercie
  - malá - rozpětí 1 a ½ celého tónu
  - velká - rozpětí 2 celé tóny
- 6, sexta
  - malá - rozpětí 4 celé tóny
  - velká - rozpětí 4 a ½ celého tónu

7, septima	- malá	- rozpětí 5 celých tónů
	- velká	- rozpětí 5 a ½ celého tónu

Tyto intervaly nemohou být čistými!

Intervaly dále rozlišujeme podle různých hledisek na různé skupiny a druhy. Zní-li oba tóny intervalu současně, jde o interval harmonický, zní-li za sebou, vzniká interval melodický. V notách píšeme tóny harmonických intervalů pod sebe, tóny melodických intervalů za sebou. Harmonické intervaly třídíme též na konsonance a disonance.

Při určování intervalů, u nichž jeden nebo oba tóny jsou sníženy či zvýšeny (nebo dvakrát snižené/zvýšené), určíme tak, že nejdříve odstraníme všechny posuvky a z nově vzniklého intervalu odvodíme postupný zvyšování/snižování interval skutečný. Stejným způsobem i tvoříme intervaly.

## 1.8 AKORDY

### 1.8.1 Hlavní druhy akordů

Akord je souzvuk nejméně tři tónů různé výšky. Hrajeme-li tón akordu po sobě jako melodii, vznik akord rozložený, říkáme též, že akord hrajeme v melodickém rozkladu.

Akordy třídíme podle různých hledisek.

- Podle počtu tónů: trojzvuky, čtyřzvuky, pětizvuky a více zvuky
- Podle stavby: akordy složené z tercií nebo z kvart
- Podle zvukového působení: konsonance a disonance
- dále na: základní a jejich obraty

Nejběžnější jsou akordy sestavené z tercií (terciový systém). Nazývají se podle intervalu mezi nejnižším a nejvyšším tónem. Ze dvou tercií vznikají kvintakordy (ze dvou tercií postavených na sebe vzniká kvinta), ze tří tercií vznikají septakordy (krajní tóny tvoří septimu).

Jako u intervalů, tak i u akordů vznikají obraty přeložením spodního tónu akordu o oktávu výš.

Ve vícehlasých skladbách rozlišujeme akordy a průběžné harmonie. Akordy jsou samostatné jednotky, kdežto průběžné harmonie jsou náhodné souzvuky, vznikající uplatňováním ozdobných a melodických tónů.

Podobně jako harmonické intervaly rozřazujeme i akordy na konsonance a disonance. Dnes se za konsonance považují pouze durový a mollový kvintakord a jejich obraty, ostatní akordy patří k disonancím.

### 1.8.2 Kvintakord

Kvintakord je trojzvuk, který vznikl kombinací dvou tercií. K jeho stavbě používáme velké a malé tercií a v důsledku toho máme čtyři kvintakordy:

### 1. Durový (velký, tvrdý)

Rozpětí - čistá kvinta  
tercie

Jeho přesné určení je odvozeno z velké tercie, která mu dodala zvláštní zvukový charakter.

Označení: C, C-dur

C - E - G

Sestava – velká tercie, malá

### 2. Molový (málý, měkký)

Rozpětí – čistá kvinta

Jeho přesné určení je odvozeno z malé tercie, která mu dodala zvláštní zvukový charakter.

Označení: c, Cmin, Cmi, Cm

C - E<sup>b</sup> - G

Sestava – malá tercie, velká tercie

### 3. Zmenšený

Rozpětí – zmenšena kvinta

Označení: Cmi<sup>5-</sup>, C<sup>o</sup>, C-, C<sup>5-</sup>

C - E<sup>b</sup> - G<sup>b</sup>

Sestava – dvě malé tercie

### 4. Zvětšený

Rozpětí – zvětšená kvinta

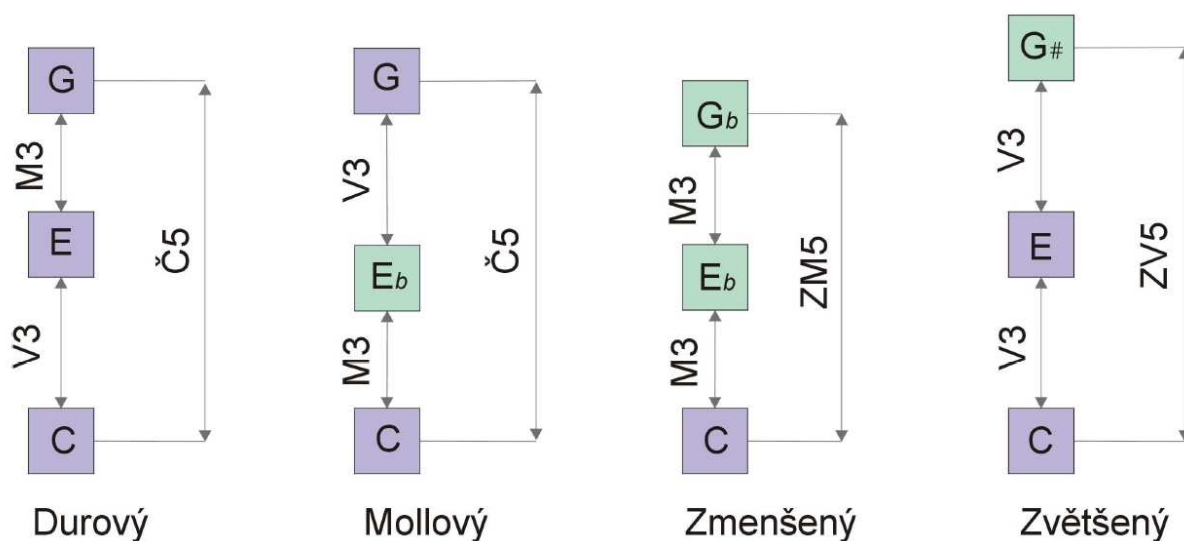
Označení: C<sup>5+</sup>, C<sup>5#</sup>

C - E - G<sup>#</sup>

Sestava – dvě velké tercie

Durové a mollové kvintakordy se jmenují podle své primy. Jména durových kvintakordů píšeme velkým písmenem, u mollových akordů malým písmenem.

## Kvintakordy



Obr. 31 Kvintakordy

U akordů určujeme intervaly vzestupně od nejnižšího tónu.

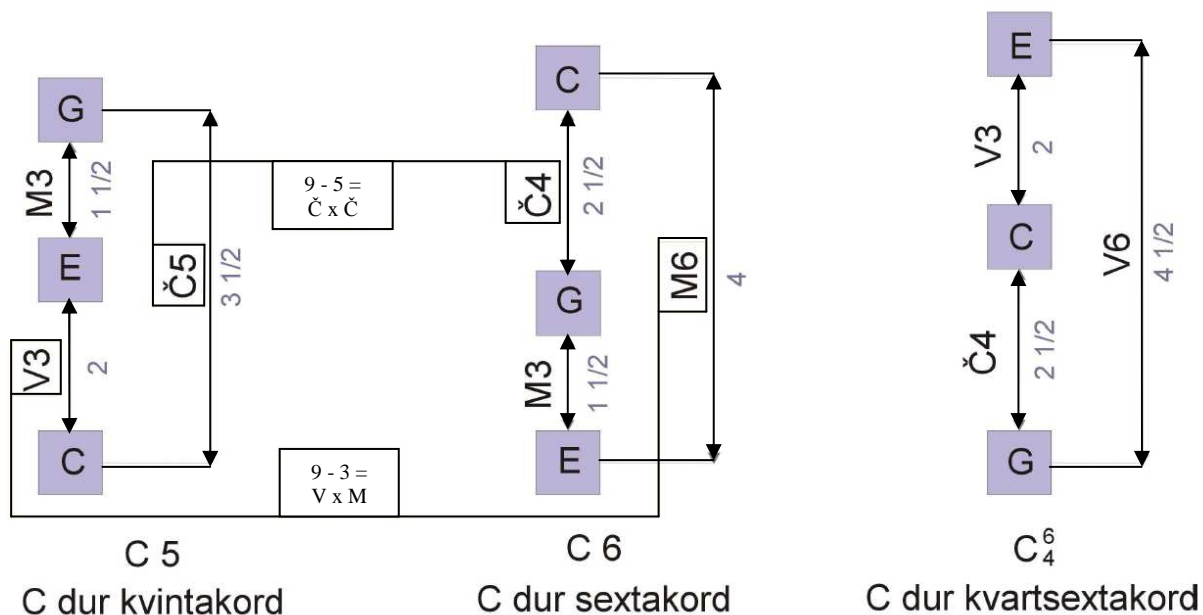
### 1.8.2.1 Obraty kvintakordů

Kvint akord má dva obraty, sextakord a kvartsextakord. První obrat vznikne přeložením primy kvintakordu o oktávu výš. Protože jeho krajní tony tvoří sextu, jmenuje se *sextakord*.



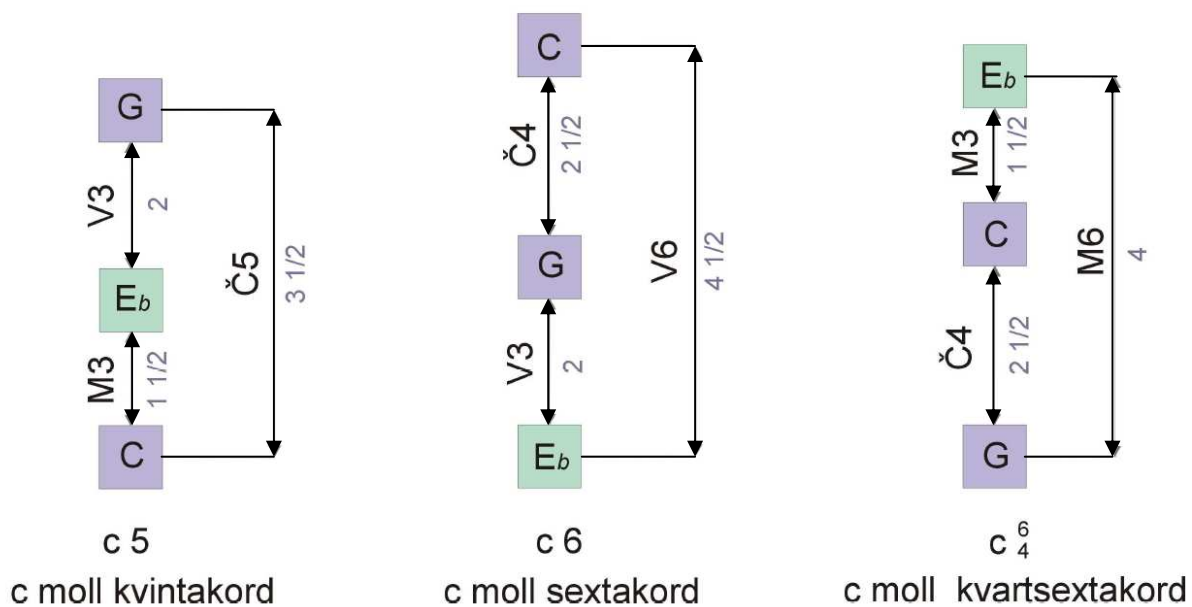
Druhý obrat vytvoříme přeložením spodního tónu sextakordu o oktávu výš. Jeho primou je původní kvinta, akord sestává z kvarty a sexty a odtud pochází i jeho název: *kvartsextakord*.

### Obraty durového kvintakordu od tónu C



Obr. 32 Obraty durových kvintakordů

### Obraty mollového kvintakordu od tónu C



Obr. 33 Obraty mollových kvintakordů

Obdobně se tvoří obraty u zmenšených a zvětšených kvintakordů. Protože je velká tercie enharmonická zmenšené kvartě, jsou zvětšený kvintakord a jeho obraty vzájemně enharmonické.

Například máme-li zvětšený akord složený z těchto tónů C-E-Gis, tak prvním obrat je E-Gis-C, ale zároveň můžeme tvrdit, že se jedná o zvětšený kvintakord E-dur.

### 1.8.3 Septakord

Septakordy vznikají připojením další tercie ke kvintakordům. Septakord sestává ze tří tercií, jeho krajní tóny tvoří septimu. Z malých a velkých tercií můžeme utvořit celkem sedm septakordů.

Názvy septakordů jsou závislé na tom, jaký kvintakord tvoří první tři tóny septakordu (jeho prima, tercie a kvinta) a jaká je jeho septima, zda je velká, malá nebo zmenšená. Názvy septakordů se skládají ze dvou slov, první slovo označuje kvintakord, druhé septimu, přičemž pro označení kvintakordů se používá zjednodušených názvů tvrdě, měkce, zmenšeně a zvětšeně ... (malý nebo velký septakord apod.).

**1. Tvrdě velký C<sup>7maj</sup>** **C – E –**  
**G - H**

Rozpětí – velká septima

Sestava – durový kvintakord + velká tercie

Obvykle je označován pouze jako velký nebo durový.

**2. Tvrdě malý C<sup>7</sup>** **C – E – G**  
**- H<sup>b</sup>**

Rozpětí – malá septima

Sestava – durový kvintakord + malá tercie

Nejběžnějším názvem je septakord dominantní.

**3. Měkce velký Cmi<sup>7maj</sup>** **C – E<sup>b</sup> –**  
**G - H**

Rozpětí - velká septima

Sestava – mollový kvintakord + velká tercie

**4. Měkce malý Cmi<sup>7</sup>** **C – E<sup>b</sup> – G**  
**- H<sup>b</sup>**

Rozpětí - malá septima

Sestava – mollový kvintakord + malá tercie

Běžným je opět zkrácené označení malý nebo mollový.

**5. Zmenšeně malý Cmi<sup>7/5-</sup>** **C – E<sup>b</sup> – G<sup>b</sup> -**  
**H<sup>b</sup>**

Rozpětí - malá septima

Sestava – zmenšený kvintakord + velká tercie

Bývá označován jako polozmenšený.

## 6. Zmenšeně zmenšený Cdim H<sup>bb</sup>

C – E<sup>b</sup> – G<sup>b</sup> –

Rozpětí - zmenšená septima  
Sestava – zmenšený kvintakord + malá tercie  
V praxi se vžil název zmenšený.

## 7. Zvětšeně velký C<sup>7/5+ maj</sup> G<sup>#</sup> – H

C – E –

Rozpětí - velká septima  
Sestava – zvětšený kvintakord + malá tercie

### 1.8.3.1 Obraty septakordů

Každý septakord má tři obraty, které se tvoří podobně jako obraty kvintakordů. **Názvy obratů septakordu jsou voleny podle intervalů, které tvoří tóny septimy (původní prima a septima septakordu) s nejhlubším tónem obratu.**

U prvního obratu tvoří původní septima se spodním tónem obratu interval kvinty a původní prima interval sexty. Obrat se podle toho jmenuje **kvintsextakord** a značí se <sup>6</sup>5.

U druhého obratu tvoří původní septima se spodním tónem obratu interval tercie a původní prima interval kvarty. Obrat se podle toho jmenuje **terckvartakord** a značí se <sup>4</sup>3.

Nejhlubším tónem třetího obratu septakordu je původní septima a s původní primou tvoří interval sekundy. Obrat se podle toho jmenuje **sekundakord** a značí se 2.

U všech septakordů je tomu stejně, rozdíly jsou jen ve velikosti tercií, sekund a dalších v nich uplatněných intervalů.

V celkovém souhrnu představují septakordy a jejich obraty 28 souzvuků.

Podrobnější rozebírání obratů septakordů by bylo zbytečně obsáhlé a pro znalosti potřebné pro tento projekt zbytečné.

## 1.9 RELATIVNÍ A ABSOLUTNÍ HUDEBNÍ SLUCH

*Relativní hudební sluch* je schopnost analyzovat pohyb melodie a odlišit vzdálenosti mezi jednotlivými tóny. Psychologové a hudební pedagogové ho v učebnicích většinou řadí mezi (z větší části vrozené) vlohy. Těm, kdo laicky řečeno "hudební sluch nemají", se říká amutici. Opět se tradičně popisuje, že *expresivní amúzie* neznamená ztrátu schopnosti analyzovat tón, ale neschopnost intonovat (zpívat přesně) melodii, a to přesto, že takový člověk "uvnitř" melodii slyší dobře a je schopen vnímat, že ji nezpívá správně. *Expresivní amúzie* se tedy týká spíše dovednosti čistě zpívat než samotného sluchu. *Impresivní amúzie* znamená neschopnost nejen správně melodii intonovat, ale i neschopnost si ji správně představit. Impresivní amutik neslyší, že zpívá falešně a neslyší, když falešně zpívají ti okolo. Expresivních amutiků je spousta, impresivních je hodně málo.

Naopak *absolutní sluch* znamená schopnost nejen vnímat správně nejen relativní složku melodie (interval), ale i její absolutní tonální posazení. Absolutní sluchař si poslechne

nahrávku a řekne vám, že ta skladba je třeba od D. Také umí vzít do rukou noty a zazpívat dané tóny v jejich správné absolutní výšce. [1]

Seznam použité literatury:

[1] <http://mojeskola.net/hudebni-a-absolutni-sluch>

[2] <http://www.farnost-zatec.cz/soubory-ke-stazeni/Varhany/vseobecna-hudebni-nauka.pdf>

Seznam tabulek:

Tab. 6: Tónová soustava – evropské rozdělení tonů do oktáv .....	2
Tab. 7: Základní a odvozené tóny.....	3
Tab. 8: Rozdílné stupně stejnojmenných stupnic.....	7
Tab. 9: Stejnojmenné stupnice .....	7
Tab. 10: Porovnání durové a harmonické mollové stupnice.....	8
Tab. 11: Porovnání durové a melodické mollové stupnice .....	8
Tab. 12: Názvy základních intervalů.....	8

Seznam obrázků:

Obr. 23 Notová osnova.....	2
Obr. 24 G klíč.....	2
Obr. 25 Noty na pomocných linkách .....	3
Obr. 26 f klíč.....	3
Obr. 27 Klaviatura - Celý tón a půltón, tóny enharmonické.....	4
Obr. 28 Kvart-Kvintový kruh.....	6
Obr. 29 Odvozené intervaly .....	9
Obr. 30 Kvintakordy .....	11
Obr. 31 Obraty durových kvintakordů .....	12
Obr. 32 Obraty mollových kvintakordů .....	12