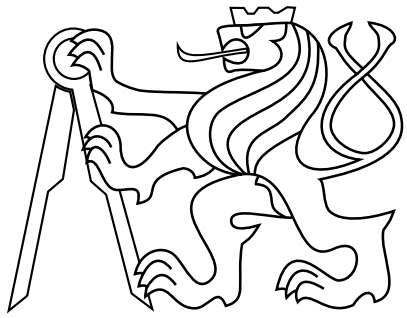
České vysoké učení technické v Praze
Fakulta elektrotechnická



BAKALÁŘSKÁ PRÁCE

Externí mikroprocesorová jednotka pro syntézu zvukových signálů

studijní program: Komunikace, Multimédia a Elektronika

studijní obor: Aplikovaná Elektronika

vedoucí práce: Doc. Dr. Ing. Jiří Hospodka

Praha 2015 **Pavel Vančura**

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra mikroelektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **VANČURA Pavel**

Studijní program: Komunikace, multimédia a elektronika
Obor: Aplikovaná elektronika

Název tématu: **Externí mikroprocesorová jednotka pro syntézu zvukových signalů.**

Pokyny pro vypracování:

1. Navrhněte externí procesorovou jednotku pro generování periodického signálu s definovaným průběhem a to buď v časové nebo kmitočtové oblasti.
2. Uvažujte kmitočet základní harmonické v rozsahu 8 oktáv, tj. přibližně od 33 Hz do 4,2 kHz.
3. Vzorkovací kmitočet volte 48 kHz, dynamické rozlišení 16 bitů.
4. Pro syntézu využijte možnosti kmitočtové modulace.
5. Práce by měla obsahovat: Výběr vhodného mikrokontroleru a vývojové desky, obvodovou realizaci periferních částí a softwarovou realizaci digitálních oscilátorů řízených pomocí MIDI protokolu.

Seznam odborné literatury:

[1] Eduardo Reck Miranda: Computer Sound Design. Focal Press 2002. ISBN 0-240-51693-1, NTK 6B080.
[2] Eduardo Reck Miranda, John Al Biles: Evolutionary Computer Music. Springer-Verlag London Limited 2007. ISBN 10: 1-84628-599-2, NTK 6B080
[3] Mark Vail: Vintage synthesizers. Miller Freeman Books 2000. ISBN 0-87930-303-3

Vedoucí: **doc.Dr.Ing. Jiří Hospodka**

Platnost zadání: 31. .8. 2016

L.S.

prof. Ing. Miroslav Husák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 1. 2015

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

V Praze dne

_____

podpis

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Doc. Dr. Ing. Jiřímu Hospodkovi za odbornou a pedagogickou činnost. Mé poděkování také patří Prof. Ing. Pavlovi Sovkovi CSc. za cenné rady a připomínky. Dále děkuji i svým kolegům Bc. Jiřímu Závorkovi a Filipu Řežábkovi za pomoc a kritiku při zpracování této práce.

Abstrakt

Náplní této bakalářské práce je syntéza zvuku metodou frekvenční modulace s využitím mikrokontroleru. Teoretická část se zabývá problematikou generování digitálních periodických signálů a využití frekvenční modulace pro syntézu zvuku. Konec teoretické části je věnován výběru vhodného mikrokontroleru a vývojového prostředí. Druhá část práce je věnována vlastnímu návrhu, jeho řešení a realizací. Vlastní návrh zahrnuje nastavení potřebných periferií mikrokontroleru, porovnání několika softwarových metod pro generování digitálních periodických signálů a praktické využití frekvenční modulace pro syntézu zvuku. Ovládání a nastavování parametrů softwaru a návrhu plošného spoje bude věnován závěr této práce.

Klíčová slova

syntezátor, STM32F4 Discovery, zvuková syntéza, frekvenční modulace, digitální generování periodických signálů

Abstract

The major objective of this Bachelor thesis is frequency modulation synthesis with using of a microcontroller. The theoretical part deals with issue of generating digital periodic waveforms and theory of frequency modulation synthesis. At the end of the theoretical part is covered the selection of the most suitable microcontroller and development tools. Second part of this work is focused on the own practical solution and realization. The practical solution comprises configuration of microcontroller peripheral devices, comparison of generating digital periodic waveforms methods and practical utilization of frequency modulation synthesis. Methods for adjusting software parameters and PCB design are described in the conclusion.

Keywords:

synthesizer, STM32F4 Discovery, sound synthesis, frequency modulation, digital waveform generation

Obsah

Seznam použitých zkratek a symbolů

Zkratka/symbol	Popis
ADC	analog digital converter
ADSR	attack, decay, sustain, release (obálka)
COARSE	posun v ladění o oktávy
DAC	digital analog converter
DMA	direct memory access
DPS	deska plošného spoje
NCO	numerically controlled oscillator
ENV	envelope generator
FINE TUNE	jemné ladění
FM	frekvenční modulace
I2C	sběrnice pro digitální komunikaci
I2S	sběrnice po přenos digitálního zvuku
LED	light emitting diode
LEVEL	úroveň signálu
LFO	low frequency oscillator
MCU	mikrokontroler
MSB	bit nejvyššího řádu
NC	nezapojeno
SEQ	sekvencer
SNR	odstup signál šum
SPI	sběrnice digitální komunikace
USART	universal synchonous reciever transmiter

1 Úvod

S rozvojem elektrotechniky vznikla myšlenka uměle elektronicky napodobit zvuk klasických hudebních nástrojů. Pro hudební producenty byla zajímavá představa vlastnit v nahrávacím studiu syntezátor, který bude schopen imitovat klasické hudební nástroje především z ekonomických a časových důvodů. Správně napodobit zvuk klasického hudebního nástroje, například akustické kytary je obtížné, protože spektrum takového zvuku se mění velmi složitě v čase. Na druhou stranu i syntezátor, zejména analogový má svůj specifický zvukový charakter. Z tohoto důvodu se syntezátory staly spíše osobitými nástroji samy o sobě, tak jako akustická kytara.

Rozvoj vědy a techniky přinesl rozvoj i zvukové syntézy. Nejvíce rozšířenou a první metodou zvukové syntézy je rozdílová syntéza, která je založena na generování přesných hudebních signálů s velkým množstvím harmonických složek a s pomocí filtrů je tvarována určitá zvuková barva. Opakem této metody je aditivní syntéza, která je založena na postupném přidávání harmonických složek s určitou amplitudou. S rozvojem digitální techniky kolem roku 1983 přišla japonská značka Yamaha se svým legendárním a velmi úspěšným syntezátorem DX7 [1]. Tento syntezátor využívá jako metodu zvukové syntézy frekvenční modulaci. Krátce poté uvedla firma Casio u své série CZ modulaci fáze. Modulace fáze poskytuje podobné výsledky jako modulace frekvence, avšak ne totožné.

Začátkem 21. století se začíná opět objevovat poptávka po klasických analogových syntezátorech. Analogové systémy mají svůj nezaměnitelný zvukový charakter, který se velmi těžko napodobuje digitálně. Velké firmy s nastavenými výrobními linkami na digitální systémy jen s velmi malou pružností reagují na současné trendy. Letošní veletrh NAMM 2015 potvrdil tento analogový trend. Velké firmy jako je Roland, nebo Korg, opět představily jako novinky své reedice úspěšných analogových modelů. Dokonce i americká firma Moog, jejíž zakladatel R.A. Moog je považován za zakladatele rozdílových syntezátorů tak, jak je známe dnes, zavedla do výroby znovu svůj úspěšný modulární systém prodávaný v sedmdesátých letech minulého století. Doba vakua, kdy byly na trhu k dostání jen bazarové analogové syntezátory, dala možnost vzniknout malým projektům, nadšencům jako je například dnes už etablovaná firma Mutable-Instruments z Francie. Tato firma byla velmi úspěšná se svým malým syntezátorem Shruthi [2], který byl prodáván jako stavebnice. Je to rozdílový syntezátor s digitálně generovanými vlnami a analogovými filtry. Dalším úspěšným projektem je PreenFM [3], dnes už v reedici PreenFM2. Tento digitální syntezátor využívá frekvenční modulaci a je koncipován na základě úspěšného modelu Yamaha DX7.

Hlavními cíly této práce je vybrat dostatečně výkonný mikrokontroler a navrhnout externí jednotku pro generování periodického průběhu v rozsahu osmi oktáv, která pro zvukovou syntézu využívá frekvenční modulace. Vzorkovací kmitočet bude 48 kHz a dynamický rozsah 16 bitů. Práce je rozdělena do dvou částí. Teoretickou část, která se zabývá generováním digitálních periodických signálů a frekvenční modulací. Konec teoretické části zahrnuje výběr a porovnání vhodného vývojového prostředku. Druhá část práce je věnována vlastnímu řešení, které lze rozdělit na softwarovou část a hardwarovou část. Softwarová část se zabývá realizací a porovnáním metod generování digitálních periodických signálů a frekvenční modulací vycházející z teoretické části. Hardwarová část se zabývá připojením potřebných kontrolních prvků k mikrokontroleru (včetně nastavení periferií) pro nastavování parametrů, zobrazování hodnot nastavených parametrů a také obvodům pro připojení MIDI vstupu.

Výstupem výše definované struktury práce bude samostatná jednotka schopná generovat statická zvuková spektra s dostatečnou zvukovou kvalitou. Statická zvuková spektra nejsou pro syntézu hudebních nástrojů dostatečná. Pro dynamické změny spekter v čase jsou třeba modulátory, např. ADSR. Nicméně, cílem této práce je vytvořit základ, který bude možno v pozdějších projektech dále rozvíjet. Bude možné přidat softwarové modulátory nebo díky realizaci pomocí mikrokontroleru uvažovat o přidání analogových filtrů.

2 Teoretická část

Teoretická část práce bude věnována popisu třech oblastí. Jako první bude zmíněn protokol MIDI, který je důležitý pro řízení syntezátoru z externí klaviatury. Protokol MIDI bude omezen pouze na využití MIDI pro indikaci stisku klávesy a nastavení příslušné frekvence digitálního oscilátoru podle konkrétní noty. MIDI dále nabízí možnosti nastavování parametrů a další funkce. Tyto složitější funkce MIDI jsou ale mimo rozsah této práce. Další oblastí bude problematika, vlastnosti a techniky generování digitálních periodických signálů. Tato část práce je důležitá, protože na kvalitě digitálního oscilátoru závisí kvalita zvuku celého syntezátoru. Konec teoretické části bude věnován frekvenční modulaci a její aplikaci ve zvukové syntéze.

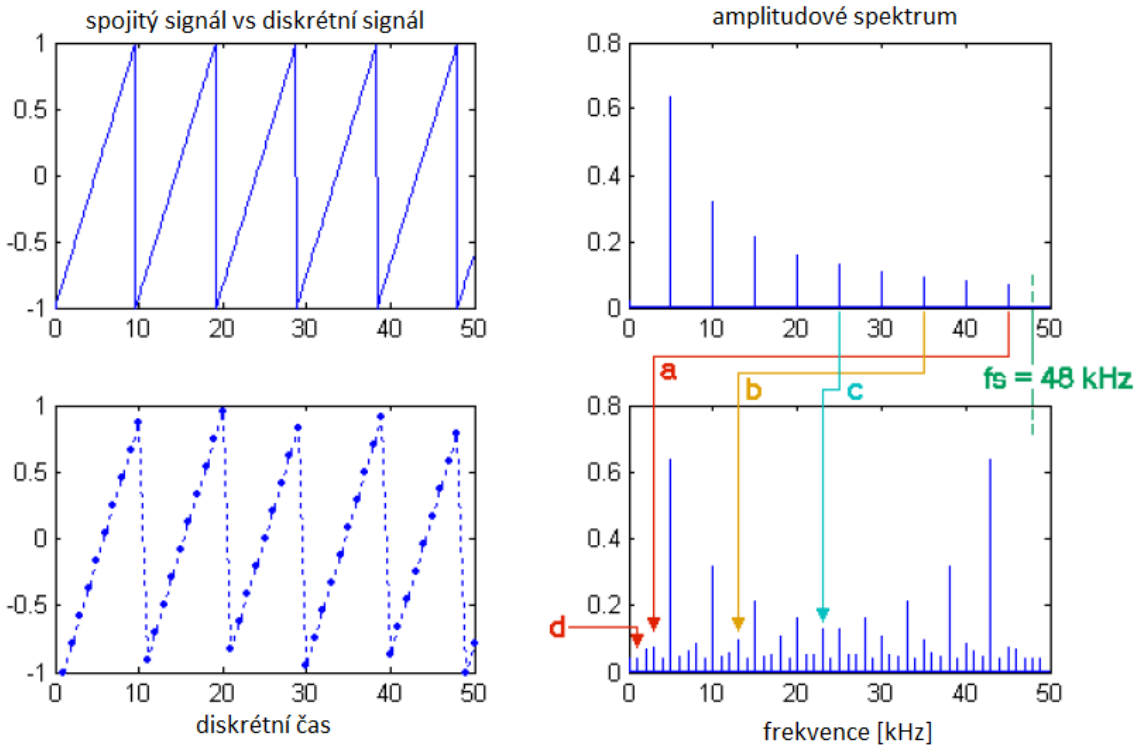
2.1 MIDI

MIDI (Musical Instruments Digital Interface) je digitální komunikační protokol, který slouží k propojení a komunikaci hudebních nástrojů, syntezátorů a dalšího počítačového vybavení. První verze MIDI se poprvé objevila v létě roku 1983. Významný propagátorem se stala americká společnost ATARI. Pomocí MIDI lze ovládat či zaznamenávat noty a dynamiku, řídit tempo a nastavovat parametry. Více zařízení lze propojovat v sérii za sebou. Každé zařízení v řetězu má svůj kanál. Kanálů je maximálně 16. To znamená, že lze propojit maximálně 16 zařízení do série. MIDI pracuje na frekvenci 31.25 KHz.

Základní komunikaci lze jednoduše vysvětlit například při stisknutí klávesy. Při stisku klávesy MIDI klaviatury se na výstupní MIDI port (konektor DIN) vyšle kontrolní byte (1001000=144), to říká druhému zařízení pracující na stejném kanálu, že se jedná o MIDI zprávu nota zapnuta. Kontrolní byte je následován bytem s číslem noty. Pokud byla stisknuta klávesa C4, hodnota tohoto bytu bude (111100). Dále za bytem s číslem noty je vyslán byte s hodnotou razance úhozu (dynamiky). Na stejném principu funguje veškerá MIDI komunikace. Kontrolní byte říká, o jakou zprávu se jedná a potom následuje byte, nebo dva byty s hodnotou parametru. Zařízení, které přijímá MIDI zprávu, rozlišuje, zdali se jedná o kontrolní byte, nebo byte nesoucí hodnotu podle MSB. Pokud je MSB = 1, jde o MIDI zprávu nesoucí hodnotu kontrolního bytu. Pokud MSB = 0 jedná se o byte nesoucí hodnotu parametru. Když vezmeme v úvahu, že byte má osm bitů a MSB informuje o typu zprávy, tak pro hodnotu parametru zbývá sedm bitů, tedy rozsah 0-127 decimálně.

2.2 Analogový versus digitální signál

Dle [4], „*na rozdíl od analogových obvodů, je digitální signál v čase diskrétní a proto má periodické spektrum s periodou vzorkovacího kmitočtu fs. Toto tvrzení se dá dokázat prozkoumáním navzorkované verze spojitého signálu, jehož dvoustranné spektrum je superponováno celými násobky vzorkovacího kmitočtu fs vzorkovacím procesem. Komponenty signálu na fundamentální frekvenci fo vedou k dalším komponentům na výstupu oscilátoru daným vztahem* $N * fs \pm fo$*, kde N jsou celá čísla. Tyto nechtěné artefakty se vracejí do slyšitelného audio pásma a není možné je odstranit. Následující obrázek 3.2.1 ukazuje pilovitý signá o frekvenci* $f_o = 5kHz$ *a jeho representaci v diskrétním čase včetně aliasingu, jehož psycho-akustická relevance bude diskutována na bázi vybraných komponent a-d.*“



Obr. 3.2.1 Analogový versus digitální signál [4]

(a) *Nechtěný komponent 9-té harmonické originálního signálu:* $f_s - 9f_o = 3kHz$. *Slyšitelný.*

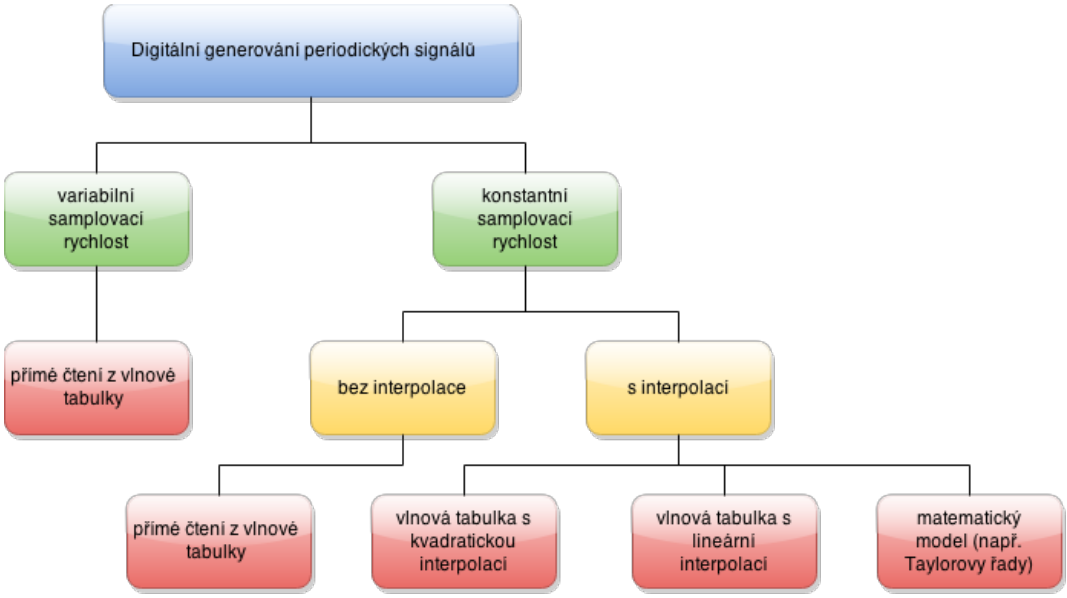
(b) *Nechtěný komponent 7-mé harmonické originálního signálu:* $f_s - 7f_o = 13kHz$. *Částečně maskovaný druhou harmonickou originálního signálu. Navíc ve vysokých kmitočtech je lidský sluch nedokonalý a tento komponent je vnímám spíše jako jas.*

(c) *Nechtěný komponent 5-té harmonické originálního signálu* $f_s - 5f_o = 23kHz$. *Neslyšitelný.*

(d) *Nechtěný komponent 19-té harmonické originálního signálu:* $f_s - 19f_o = 1kHz$. *Slyšitelný, ale pokud je amplituda o 80 dB níže než originální signál, je tento komponent zanedbatelný.*

2.3 Digitální generování periodických signálů

Tato kapitola se zabývá možnostmi, jak lze generovat periodické signály s využitím mikrontroleru v rozsahu přibližně 20 Hz – 20 kHz. Obr. 3.2.1 ukazuje základní rozdělení digitálních oscilátorů dle [5].



Obr. 3.2.1 Základní rozdělení digitálních oscilátorů [5]

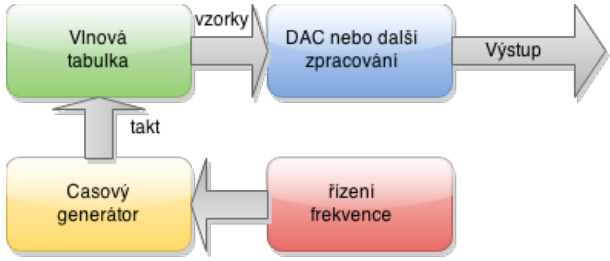
Pro digitální oscilátor potřebujeme zdroj časových „impulsů“. Tento zdroj je použit pro volání určité funkce, kde dochází ke kalkulaci vzorku, který je následně předán DAC. Ke změně frekvence může docházet dvojím způsobem. A to změnou frekvence zdroje taktů (variabilní vzorkovací kmitočet), nebo fázovým akumulátorem (konstantní konstantní vzorkovací kmitočet). Obě tyto metody včetně interpolace budou vysvětleny v následujících kapitolách.

Vlnová tabulka, neboli v překladu „wavetable“, někdy se říká i look-up tabulka, obsahuje vzorky jednoho cyklu několika různých vlnových forem. Například PPG Wave 2.3, jeden z prvních „wavetable“ syntezátorů obsahoval vlnovou tabulku šedesáti čtyř různých vlnových cyklů. Vlnová tabulka lze realizovat v jazyce C například pomocí jednorozměrného pole, kde jsou jednotlivé cykly vln řazeny za sebou. Velikosti vlnových tabulek se volí s mocninou dvou, například 128, 256, 512 vzorků. Každý vzorek má v poli svůj unikátní index. Výhody použití vlnových tabulek jsou:

- Lze generovat libovolný periodický signál
- Lze přepínat výstupní vlny a tím tvořit spektrální modulaci. Například lze generovat střídavě jeden cyklus sinu a jeden cyklus pilovitého signálu. Nebo provádět přepínaní vln na výstup například pomocí ADSR.

2.3.1 Digitální oscilátor s variabilním vzorkovacím kmitočtem

U tohoto principu se dosahuje změny frekvence pomocí změny vzrokovacího kmitočtu. Blokový diagram je znázorněn na obr. 2.3.1.1.



Obr. 2.3.1.1 Blokové schéma oscilátoru s variabilním vzorkovacím kmitočtem

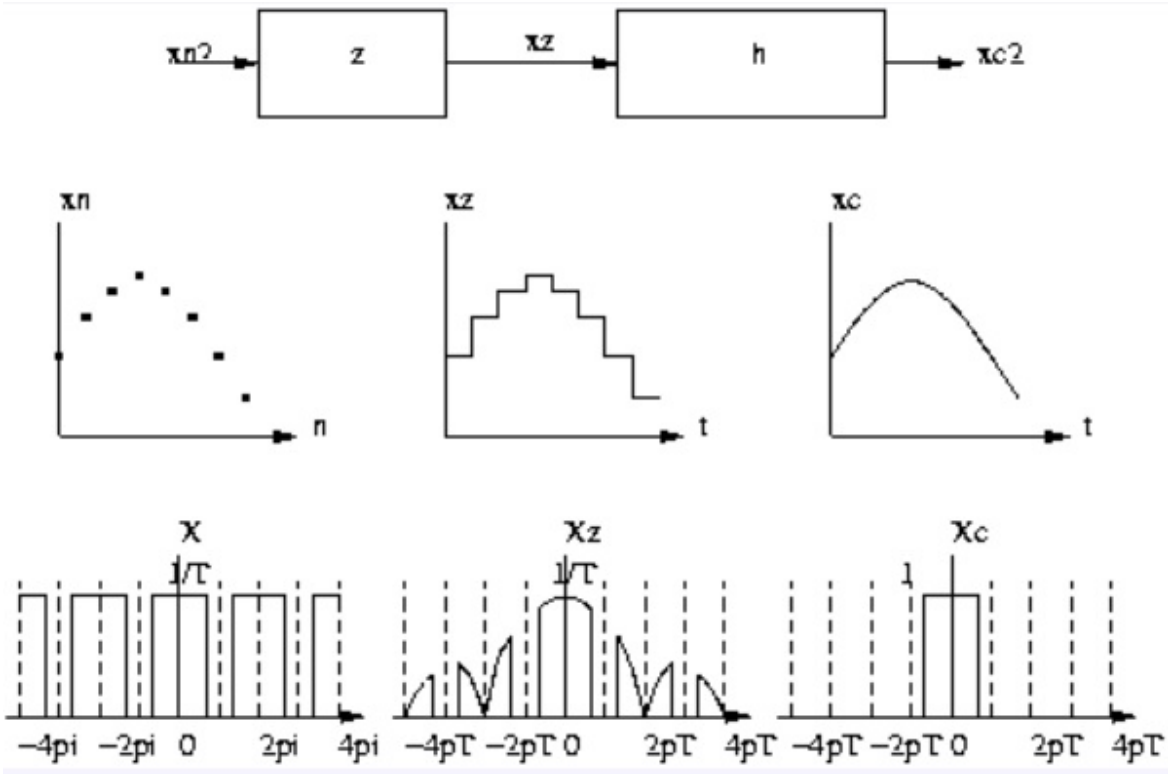
Představme si, že máme ve vlnové tabulce 2048 vzorků vlny například sinus. Těchto 2048 vzorků tvoří jeden vlnový cyklus. Dále si představme, že máme časový generátor t_p s pravidelným taktem, který na každý časový puls předá jeden vzorek z vlnové tabulky na DAC, nebo k dalšímu zpracování. Tento časový zdroj může být u mikrokontroléru přerušení vyvolané například přetečením časovače. Pokud chceme generovat vlnu tohoto sinu o frekvenci A4 = 440 Hz, musí mít časový zdroj takt:

$$t_p = 440 * 2048 = 901.12kHz \tag{1}$$

V případě, že chceme generovat vlnu o frekvenci kolem 5 kHz, byl by potřeba časový zdroj o taktu přes 10 MHz. Jelikož 5 kHz je stále ještě relevantní hudební frekvence, kterou by každé NCO mělo být schopné generovat a dále pokud vezmeme v úvahu to, že potřebujeme více nezávislých NCO, je jasné, že tento princip je výpočetně velmi náročný. Narážíme tedy na první problém a to je výpočetní náročnost. Dle vztahu (1) je patrné, že frekvence časového zdroje závisí na počtu vzorků vlnové tabulky. Kdyby velikost vlnové tabulky byla poloviční, byla by i potřebná frekvence časového zdroje poloviční. Zde se nabízí jedna metoda eliminace výpočetní náročnosti, používaná u prvních digitálních syntezátorů, která se nazývá upsampling. Tato metoda používá pro každou oktávu jinou vlnovou tabulku. Pro každou vyšší oktávu se vlnová tabulka dělí dvěma. Oktáva 1 bude používat vlnovou tabulku o velikosti 2048 vzorků, oktáva 2 - 1024 vzorků, oktáva 3 - 512 vzorků... V každé oktávě je nota s nejvyšší frekvencí nota B. Pro B_1 je frekvence 61,735 Hz, pro$B_2 = 123,471Hz$, pro $B_3 = 246,942Hz$... Všimněme si nyní frekvence časového zdroje, pro $B_1 = 61,735 * 2048 = 126,433kHz$, pro $B_2 = 123,471*1024 = 126,434kHz$... Vidíme, že maximální frekvence pro nejvyšší noty v každé oktávě je stejná. Touto metodou je tedy možné generovat například sinus s frekvencí kolem 5 kHz s maximální frekvencí časového zdroje přibližně 125 kHz. Nicméně i přes použití metody upsampling je výpočetní náročnost značná, pokud vezmeme v úvahu, že mikrokontroler potřebuje ještě vykonávat další úlohy. NCO s variabilním vzorkovacím kmitočtem bylo použito u prvních digitální syntezátorů, samplerů, například E-mu Emulator II, Korg DSS-1, Akai S900/S950.

Podle [6] *„společně s neadekvátní rekonstrukcí signálu byly ve zvukovém spektru slyšitelné kopie vyšších frekvencí. Technicky jsou tyto kopie frekvencí nechtěné, nicméně vytváří krásný „rozjasněný“ zvukový charakter. Velmi se liší od aliasingu, protože tyto*

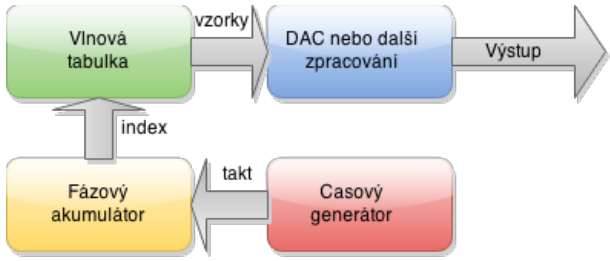
kopie jsou spíše vyšší harmonické, než neharmonické tóny. Diskrétní signál lze interpretovat jako amplitudově modulovaný impuls (obr. 2.3.1.2 vlevo nahoře) s periodickým spektrem (obr. 2.3.1.2 vlevo dole). DAC udržuje konstantní hodnotu do dalšího vzorku (obr. 2.3.1.2 vprostřed nahoře) a filtruje periodické spektrum sample funkcí (obr. 2.3.1.2 vprostřed dole). Vyšší frekvenční komponenty bývají většinou filtrovány rekonstrukčním filtrem což má za následek hladký signál (obr. 2.3.1.2) vpravo nahoře) a náležité spektrum (obr. 2.3.1.2 vpravo dole). První syntezátory ovšem nepoužívaly rekonstrukční filtr pro „správné“ zpracování signálu. Následkem toho měl jejich zvuk „ostrý“ a zajímavý charakter. Moderní techniky, které používají fázové akumulátory a interpolaci, produkují „správný“ signál. Postrádají ovšem zajímavý charakter prvních digitálních syntezátorů. Zvuk moderních nástrojů má „tupější“ charakter, protože ve spektru chybí obrazy vyšších frekvencí. “



Obr. 2.3.1.2 Vlastnosti digitálních signálů [6]

2.3.2 Princip činnosti a vlastnosti digitálního oscilátoru s fázovým akumulátorem

Tento princip se liší od předchozího principu popsaného v předchozí kapitole 2.3.1 tím, že ke změně frekvence NCO dochází změnou vzdálenosti při čtení z vlnové tabulky. Blokové schéma je znázorněno na obr. 2.3.2.1.



Obr. 2.3.2.1 Blokové schéma oscilátoru s fázovým akumulátorem

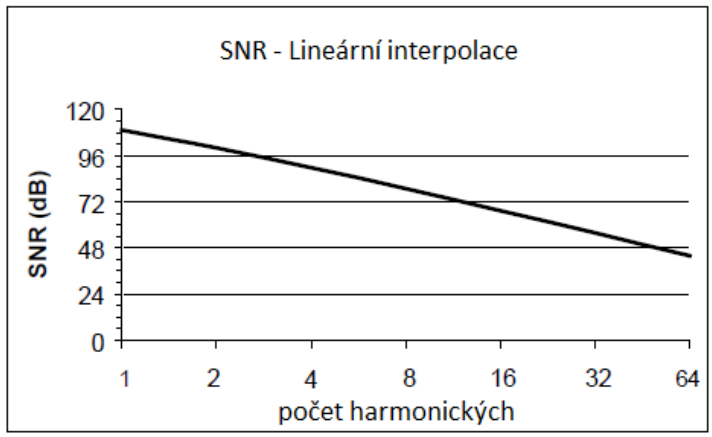
Máme-li vlnovou tabulku o velikosti $N = 2048$ vzorků a časový zdroj t_s o frekvenci 44100 Hz, bude základní frekvence $f_z = 44100/2048 = 21,533Hz$. Pokud budeme číst z tabulky vždy ob jeden vzorek, tedy vždy jednu hodnotu vzorku vynecháme, bude frekvence dvakrát vyšší $f_z = (2 * 44100)/2048 = 43,066Hz$. To je přesně to co provádí blok fázový akumulátor, vypočítá vzdálenost mezi vzorky pro konkrétní frekvenci. Tato hodnota se poté po každém taktu časového generátoru přičítá, akumuluje. Odtud název fázový akumulátor. Poté co hodnota akumulátoru dosáhne velikosti vlnové tabulku (posledního vzorku), tak se odečte právě velikost vlnové tabulky a proces se opakuje. Výpočet hodnoty je dán vztahem (2). Kde f je požadovaná frekvence, N je velikost vlnové tabulky a f_s je vzorkovací kmitočet.

$$inkrement = f * N/f_s \tag{2}$$

Podle [7] *„algoritmus fázového akumulátoru je velmi jednoduchý“*:

```
phase = 0;
increment = hz / sr;
while (true) {
output_sample = phase*wavetable_length;
phase = phase + increment;
 if (phase >= 1){ phase = phase - 1; }};
```

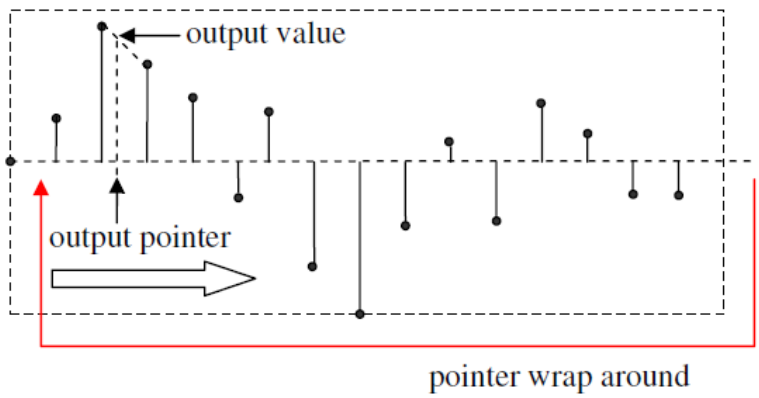
Podle [7] *„ hz je požadovaná frekvence (tón), sr je vzorkovací kmitočet, wavetable_length je velikost vlnové tabulky. Proměnné phase a increment jsou desetinná čísla, tedy musí být datového typu floating point. Proměnná phase je v intervalu [0,1), pokud je hodnota větší nebo rovna jedné, je jednička odečtena. To zajišťuje periodicitu. Tento kód je volán konstantním vzorkovacím kmitočtem, například standardní audio vzorkovací kmitočet je 44100 Hz, při velkosti vlnové tabulky 2048 vzorků. Proměnná output_sample určuje pozici vzorku uloženého v poli o velikosti 2048 hodnot. Jelikož pozice vzorku musí být celé číslo, je třeba proměnnou output_sample zaokrouhlovat na celé číslo. Tím vzniká chyba, která neovlivňuje frekvenci, ale ovlivňuje SNR a tím spektrum signálu. SNR lze zvýšit interpolací. Na obr. 2.3.2.1 je ukázáno SNR wavetable NCO s lineární interpolací v závislosti na 1-64 harmonických stejné amplitudy ve vlnové tabulce s 1024 vzorky. Můžeme zde vidět, že se SNR s vyššími harmonickými snižuje. To je z toho důvodu, že harmonickou s číslem 64 je efektivně pouze* $1024/64 = 16$ *vzorků.“*



Obr. 2.3.2.1 SNR v závislosti na počtu harmonických [7]

2.4 „Wavetable“ oscilátor

„Wavetable“ oscilátor je základním prostředkem pro generování statických spekter. Může být využit nejen pro generování zvukových spekter, ale také například jako generátor funkcí v měřících aplikacích. Tradičně „wavetable“ – vlnová tabulka, neobsahuje pouze jeden vlnový cyklus, nýbrž několik například 64 vlnových cyklů řazených za sebou. Tento oscilátor má výhodu nízké výpočetní náročnosti a má dobré spektrální výsledky s lineární interpolací.



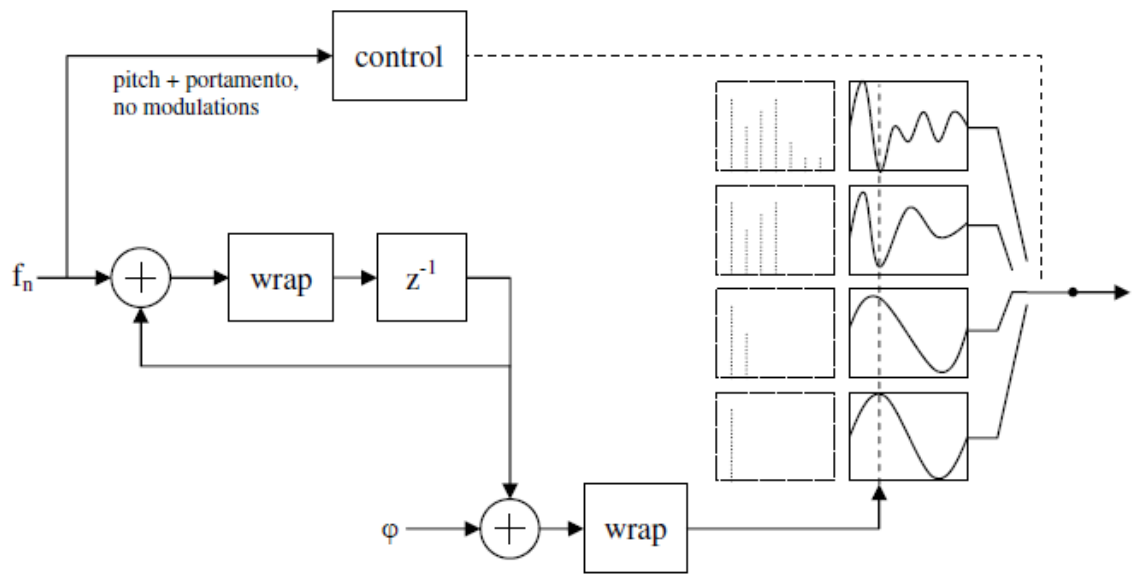
Obr. 2.4.1 Princip wavetable oscilátoru (N = 16 vzorků) [4]

Dle [4] *„Hodnoty z tabulky x[k] a výstupní spektrum X[n] je přímo vázáno k diskrétní Fourierově transformaci.“*

$$X[n] = \sum_{n=0}^{\infty} x[k] e^{\frac{-2\pi jkn}{N}}$$

$$x[k] = \frac{1}{N} \sum_{n=0}^{\infty} X[n] e^{\frac{2\pi jkn}{N}}$$

Dle [4] *„Spektrum vlnové formy v tabulce se záměrně pohybuje jen v rozmezí pro* $k = 0$ *a* $N/2$*, čímž* $X[0]$ *a* $X[N/2]$ *budou vždy nulové. Základní frekvence* f_0 *výstupního signálu je dána vzorkovacím kmitočtem* f_s *a velikostí tabulky N.* α *je jeden inkrement ukazatele ve vlnové tabulce. Pokud* $\alpha = 1$ *je generována přirozená základní frekvence vlnové tabulky* $f_o(nat) = f_s/N$ *a přirozené harmonické složky* $kf_o(nat)$ *. Proto* α *může být interpretován jako přenosový faktor relativní k* $f_o(nat)$ *. Pokud* f_0 *je příliš nízko pod* $f_o(nat)$ *, výsledný zvuk začíná ztrácet výšky, proto by neměla být velikost vlnové tabulky příliš malá. Nejčastější volba je N=512 vzorků. V praktických aplikacích počet harmonických* k_{max} *nebývá vyšší než 250. To umožňuje snížit aliasing pod požadovanou úroveň zvýšením velikosti tabulky na* $N >> k_{max}$*. Přenosový faktor* α *by měl být limitován v rozsahu* $[0,75; 2]$ *pro systém s* $f_s = 48kHz$ *a spektrem do 20 kHz proto, abychom se vyhnuli tupému zvuku a aliasingu. Zde vzniká problém. Pokud chceme udržet přenosový faktor v tomto rozsahu, bude možné měnit frekvenci* f_0 *pouze v rozsahu přibližně jedné oktávy. Jako první se nabízí řešení vytvořit novou, dvakrát menší tabulku. Tato tabulka by měla dvakrát vyšší základní frekvenci* $f_o(nat)$ *a tím by se frekvenčně pokryla další oktáva. Následkem toho se bohužel aliasing zvýší, protože typická spektra mají energii koncentrovanou na nízkých k. Navíc vyšší* $f_o(nat)$ *potlačuje sluchové maskování aliasingových komponent pod fundamentální frekvencí. Sluch je více citlivý se zvyšující se frekvencí. Z výše uvedených důvodů se udržuje velikost tabulky konstantní a tabulka, která má pokrývat vyšší oktávu má nulovou horní polovinu spektra předcházející tabulky. Právě toto umožňuje zvýšit přenosový faktor* α*.“*



Obr. 2.4.2 Wavetable oscilátor [4]

Příklad návrhu wavetable oscilátoru dle [4]:

$$N = 512, f_s = 48kHz, f_{audio} = 20kHz, \alpha_{min} = 0.9, \alpha_{max} = 1.3$$

Ze zadání lze spočítat následující parametry pro návrh.

$$f_{o(nat)} = \frac{48000}{512}, p = \frac{\alpha_{max}}{\alpha_{min}} = 1.444$$

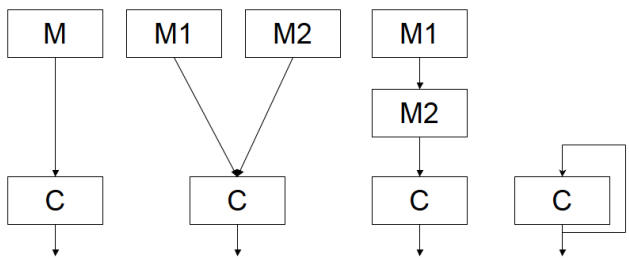
$$X[n] = 0 \quad pro \quad N - k_{max} > k > k_{min}$$

Postup návrhu wavetable oscilátoru

index tabulky	počet harmonických	frekvenční rozsah
0	$213(= k_{max(0)} = \frac{Nf_{audio}}{f_s})$	$20\text{-}122(= r_o = \alpha_{max} f_{o(nat)})$
1	$147(= \frac{k_{max(0)}}{p})$	$122\text{-}176(= r_o p)$
2	$102(= \frac{k_{max(0)}}{p^2})$	$176\text{-}254(= r_o p^2)$
3	71	254-367
4	49	367-531
5	34	531-766
6	23	766-1107
7	16	1107-1599
8	11	1599-2309
9	8	2309-3336
10	5	3336-4818
11	4	4818-6960
12	2	6960-10053
13	1	10053-20000

2.5 Frekvenční modulace a její využití pro syntézu zvuku

Dle [8] *„FM syntéza pracuje na stejných principech jako frekvenční modulace u radiových přenosů. Základní FM syntézu tvoří dva oscilátory generující sinus na audio frekvencích. První oscilátor nazývaný modulátor moduluje frekvenci druhého oscilátoru, „nosnou“. Tato jednoduchá architektura je schopná generovat širokou škálu odlišných bohatých timbrů (zvukových barev). Složitější FM syntezátory vyžadují více oscilátorů zapojených různými způsoby. Některé způsoby zapojení ukazuje obr. 2.5.1. M = modulátor, C = carrier (nosná) a šipky znázorňují tok audio signálu. Zapojení lze kombinovat.“*



Obr. 2.5.1 Nejčastější zapojení modulátoru a „nosné“ u FM syntezátorů [9]

V digitálních syntezátorech se používá také modulace fáze, která má obdobné výsledky jako frekvenční modulace. Princip se ale liší. Máme-li signál

$$f(t) = A\cos(\omega_c t + \phi(t)) \tag{3}$$

kde $\phi(t)$ je fázový posuv. Pokud $\phi(t) = \beta\cos(\omega_i t)$ dostaneme

$$f(t) = A\cos(\omega_c t + \beta\cos(\omega_i t)) \tag{4}$$

Vztah (4) je vztahem pro modulaci fáze, která je modulována signálem $\phi(t)$. U frekvenční modulace modulujeme frekvenci a ne fázi. Máme-li signál

$$f(t) = A\cos(\phi(t)) \tag{5}$$

$$\phi(t) = \int_0^t (\omega_c + \omega_m(t))dt = \omega_c t + \int_0^t \omega_m(t)dt = \omega_c t + 2\pi \int_0^t K_f m(t)dt \tag{6}$$

Pokud $m(t) = B\cos(\omega_i t)$ dostaneme

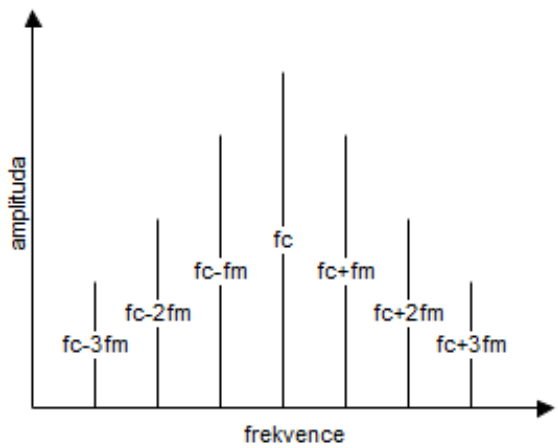
$$f(t) = A\cos(\omega_c t + 2\pi \int_0^t K_f m(t)dt) = A\cos(\omega_c t + \frac{2\pi B k_f}{\omega_i}\sin(\omega_i t))$$

Jestliže označíme $\beta = \frac{2\pi B k_f}{\omega_i}$ dostaneme stejný vztah jako u modulace fáze.

$$f(t) = A\cos(\omega_c t + \beta\sin(\omega_i t)) \tag{7}$$

Je důležité zdůraznit, že vztah (7) platí pouze pro $m(t) = B\cos(\omega_i t)$. Pro jiný signál, bude výsledek integrálu jiný a právě zde je patrný rozdíl mezi modulací frekvence a modulací fáze.

Podle [8] *„Spektrum FM signálu složeno z carrier frekvence a určitého počtu postranních pásem. Pro postranní páry platí vztahy: $f_c + nf_m$ a $f_c - nf_m$, kde n je celé číslo větší než nula.“*



Obr. 2.5.2 Příklad spektra FM signálu [8]

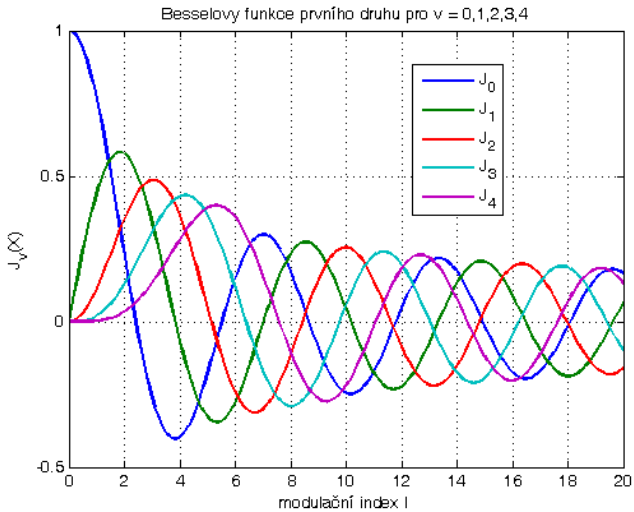
„Amplitudy postranních pásem určeny frekvenční odchylkou d. Čím větší je odchylka, tím více postranních pásem je generováno. Ovšem čím více je generováno postranních pásem, tím menší je amplituda carrier f_c. Je to tím, že množství energie obsažené v amplitudě f_c, je rovnoměrně rozloženo do postranních pásem. Modulační index I je poměr mezi frekvenční odchylkou a frekvencí modulátoru. Právě modulační index určuje počet postranních pásem.“

$$I = \frac{d}{f_m} \tag{8}$$

„Spektrum FM signálu je určováno souborem funkcí známých jako Besselovy funkce $J_n(i)$. Obr. 2.5.3 ukazuje grafickou interpretaci čtyř Besselových funkcí $J_0(i)$ pro carrier,

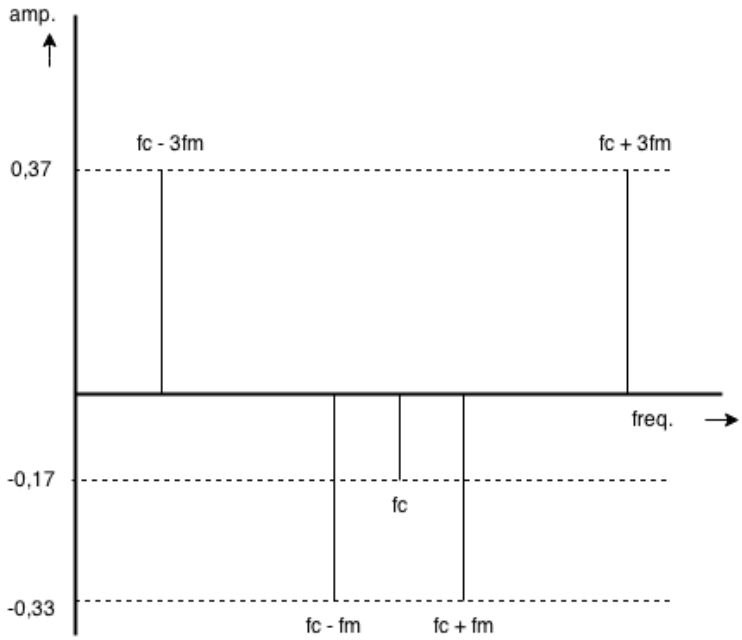
$J_1(i)$ pro k=1, $J_2(i)$ pro k=2, $J_3(i)$ pro k=3. Osa x znázorňuje modulační index a osa y amplitudu. Tyto funkce určují velikost amplitudy postranních pásem v závislosti k posunu vůči carrier frekvenci. Besselovy funkce neurčují přímo amplitudu postranních pásem, nýbrž pouze koeficient. Amplitudy příslušného postranního pásma lze získat ze vztahu (9), kde A_i jsou amplitudy postranního pásma s příslušným modulačním indexem, A je amplituda carrier frekvence a c_i je koeficient odečtený z odpovídající Besselovy funkce.“

$$A_i = C_i A \tag{9}$$



Obr. 2.5.3 Besselovy funkce

„Například pokud I=0 (tzn. bez modulace), potom posun vůči $J_0(i)$ bude nulový a amplituda carrier bude na maximální frekvenci (tzn. $c_0 = 1$), postranní pásma budou nulová. Pokud I=1, potom $c_0 = 0,76$, $c_1 = 0,44$, $c_2 = 0,11$ atd. Důležité pravidlo pro výpočet FM spektra je, že liché části spektra nalevo se navíc násobí faktorem -1. Při pohledu na Besselovy funkce je jasné, že amplitudy mohou být teoreticky i „záporné“, v závislosti na hodnotě modulačního indexu. Například pokud I=5, potom koeficient pro první pár postranního pásma bude $c_1 = -0,33$. Ve skutečnosti ale „záporné“ amplitudy neexistují. „Záporný signál“ zde indikuje, že postranní pásmo je mimo fázi (obr. 2.5.4) a ve výsledném zvuku se neprojeví. Projeví se pouze kladné části.“



Obr. 2.5.4 Příklad „záporných“ amplitud spektra FM signálu [8]

„Důležité u FM syntézy jsou změny spektra v čase. Spektrum klasickým hudebních nástrojů se v čase mění, není konstantní. Je tedy nutné dynamicky měnit hodnotu modulačního indexu I, nebo dynamicky frekvenci modulátoru. Modulační index u FM syntézy se nejčastěji (např. Yamaha DX7) mění ADSR generátorem. Průběh ADSR obálky nastavuje hodnotu modulačního indexu I a tím se dynamicky mění spektrum signálu. Další faktor, který ovlivňuje spektrum frekvenčně modulovaného signálu, je poměr frekvencí mezi nosnou a modulátorem. “

$$r = \frac{f_c}{f_m} \tag{10}$$

Dle [4] lze shrnout základní vlastnosti FM takto:

- *Frekvenční komponenty se objevují na n-násobcích $|f_c \pm nf_m|$. Pro dosažení harmonického spektra musí být f_m celým násobkem, nebo dílčím zlomkem f_c.*
- *Hlasitost jednotlivých spektrálních komponent je určena Besselovými funkcemi n-tého řádu pomocí modulačního indexu I. Vyšší I znamená více spektrálních komponent, bohatší spektrum. Pro $n > I$ harmonické velmi rychle ztrácí velikost amplitudy což efektivně omezuje výstupní spektrum. Lze dokázat, že tyto harmonické ubývají v amplitudě rychleji než exponenciálně.*
- *Frekvenční komponenty se mohou objevovat dvakrát s různou amplitudou a opačnou fází. To způsobuje charakteristické díry ve spektru. Signály s vysokým modulačním indexem I mají sklon znít nepříjemně kvůli charakteristické špičce v okolí If_m. Vyšší hodnoty I než 10 jsou málokdy používány. Navíc, harmonické za účelem dosažení dynamických spekter se nevyvíjí přirozeným způsobem, pokud je I modulován na vysokých hodnotách. Je rozumné udržet I kolem 1,5 a složitější spektrum realizovat složitějším modulátorem, nebo zpětnou vazbou.*

2.6 Prostředky pro vývoj aplikace

Tato část práce se zabývá porovnáním mikrokontrolerů, vývojových prostředků a vývojových prostředí vhodných pro aplikaci zvukové syntézy. První část této kapitoly krátce popisuje nejvhodnější vybrané prostředky a druhá část je věnována opět krátkému popisu vybraného mikrokontroléru a zvolenému vývojovému prostředí.

Při výběru vývojového prostředku bylo posuzovnáno několik základních kritérií. Zejména dostatečný výpočetní výkon, kvalitní zvukový výstup, dostatek periférií pro připojení ovládacích prvků, MIDI, zobrazovače a také cenu. Na trhu je dostatek produktů navržených přímo pro digitální zpracování signálu. Výběr byl soustředěn spíše na levnější produkty. V následujících kapitolách budou porovnány tři vybrané vývojové prostředky. Všechny tři vývojové desky lze připojit k PC přes USB a na vývojové desce se nachází, mimo jiné, konektor pro zvukový výstup.

2.6.1 MPLAB dsPIC DSC Starter kit

DSC Starter kit je vývojový prostředek od společnosti Microchip. Obsahuje ladící (debug) a programovací (programmer) obvody a mikrokontrolér dsPIC33FJ256GP506. DSC lze připojit přímo k PC pomocí USB a programovat přes kvalitní prostředí MPLAB v jazyce C. DSC obsahuje zvukový vstup, který je přiveden na interní AD převodník dsPIC33FJ256GP506. Výstupní audio signál z dsPIC33FJ256GP506, je digitální PWM signál přivedený na dolní propust a výstupní jack 3.5 mm konektor.

Vybrané parametry dsPIC33FJ256GP506:

- sběrnice 16bit
- takt 80MHz
- programová paměť 256KB
- RAM 16 384 KB
- I/O piny 56
- Digitální komunikace 2x UART, 2x SPI, 2x I2C
- 1x AD převodník 18x12bit
- Rozlišení PWM 16bit
- 13x časovač
- DMA 8 kanálů

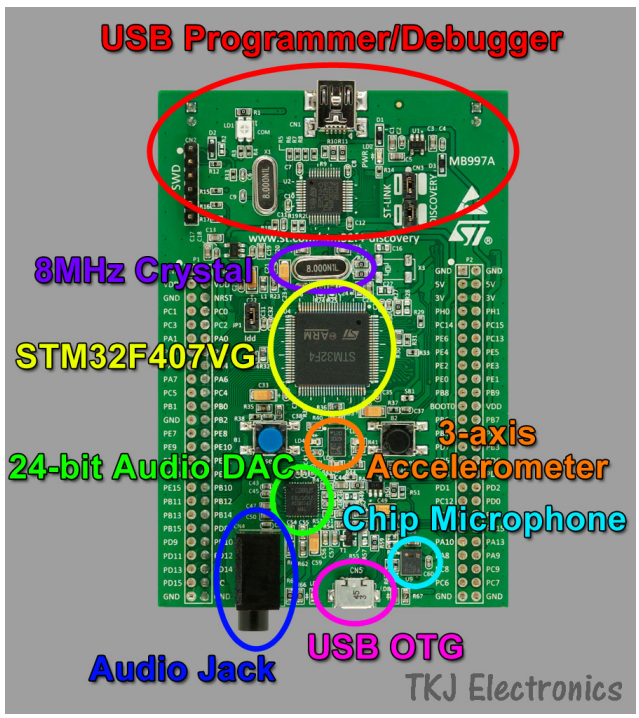
2.6.2 STM32F4 DISCOVERY

STM32F4 DISCOVERY je vývojový prostředek od firmy STMicroelectronics, který obsahuje obvody pro programování (programmer) a ladění (debugger) s připojením přes USB a mikrokontroler ARM Cortex-M4. Přímo na plošném spoji se mimo procesoru nachází také DAC převodník CS43L22 využívající delta-sigma převodník s rozlišením až 24bit/96kHz. Tento vývojový prostředek obsahuje také jedno uživatelské

tlačítko, integrovaný mikrofon, akcelerometr, 3.5 mm jack výstupní konektor a USB konektor pro externí paměťové médium.

Vybrané parametry mikrokontroleru STM32F4VTG6:

- MCU STM32F4VTG6 ARM Cortex M4
- sběrnice 32bit
- takt až 168 MHz
- Programová paměť 1024KB
- RAM 192 KB
- I/O pinů 80
- Digitální komunikace 6x USART, 3x SPI, 3x I2C, I2S, 3xADC
- 14x časovač
- 2x DMA po 8 kanálech



Obr. 2.6.2.1 STM32F4 Discovery [10]

DISCOVERY má na stranách vyvedeno všech 100 pinů MCU. Porty jsou označeny PA až PE po šestnácti pinech. Detailní popis zapojení pinů i funkce pinů lze najít ve schématu zapojení DISCOVERY a v dokumentaci STM32F407VG na stránkách STMicroeletronics [11]. Vývojová prostředí pro ARM Cortex-M4 existuje několik. Např. Atollic True-Studio, MDK-Keil, MicroC for ARM, nebo Coocox CoIDE. Všechna tato vývojová prostředí umožňují programování v C, C++, nebo assembleru.

2.6.3 Raspberry Pi model B+

Raspberry Pi je velmi výkonný malý mikropočítač s procesorem ARM11. Po zapnutí je možné nahrávat operační systém LINUX z mikro SD karty. Raspberry Pi má grafický výstup a snadno připojit monitor a pomocí USB konektorů připojit klávesnici, případně myš. Poté lze Raspberry používat jako PC s Linuxovým jádrem. Pomocí aplikace pro psaní kódu v C, C++, například QT Creator, lze vyvíjet software a přímo ovládat GPIO vstupy/výstupy, které jsou součástí Raspberry Pi. Zvukový výstup je k dispozici přes 3.5mm JACK konektor, nebo HDMI.

Vybrané parametry Raspberry Pi B+:

- ARM 11
- Takt 700 MHz
- 512MB SDRAM
- Grafický procesor Dual Core Video Core IV
- 40 GPIO I/O
- Ethernet, HDMI

2.6.4 Porovnání a výběr vývojového prostředku

Následující tabulka shrnuje vybrané parametry zvažované při výběru.

Tab. 2.6.1 Porovnání vývojových prostředků

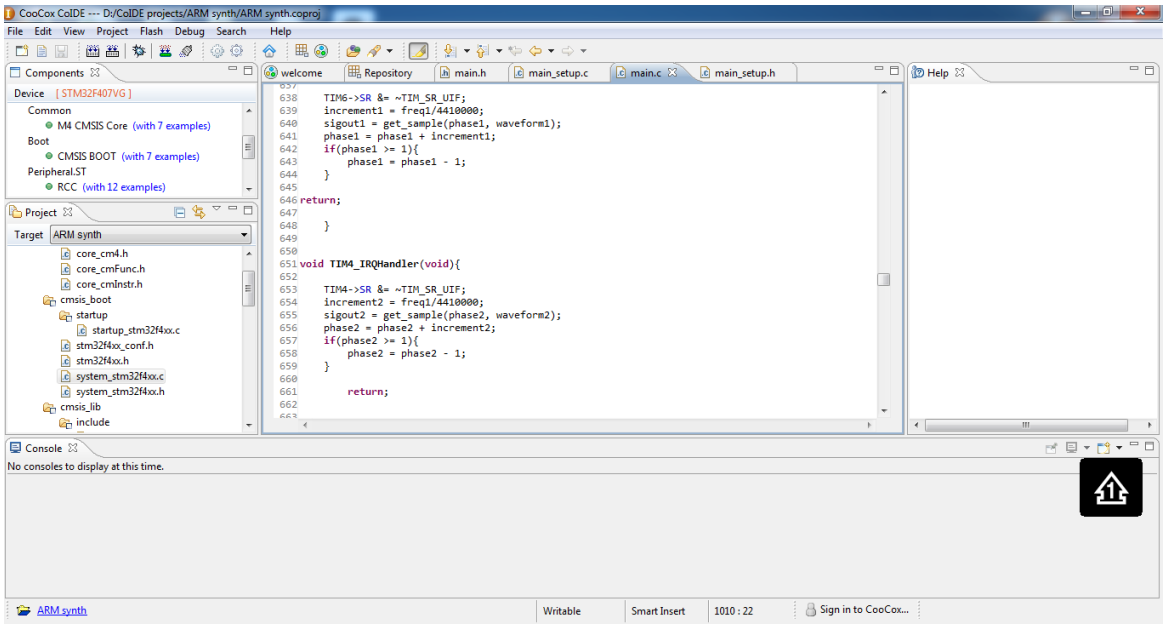
	MPLAB DSC Starter kit	**STM32F4 Discovery**	**Raspberry Pi B+**
Mikrokontroler	dsPIJ33FC256GP506	ARM Cortex-M4	ARM11
Takt [MHz]	80	168	700
RAM [MB]	16	192	512
GPIO	56	80	40
DAC	PWM	Delta-Sigma	PWM
Grafický výstup	Ne	Ne	Ano
Přibližná cena (k 1.5.2015) (KČ)	1400	450	900

V mém výběru jsem zvolil STM32F4 discovery, jelikož nabízí dostatečný výkon pro mé řešení, je levný a má ze třech porovnávaných prostředků nejkvalitnější zvukový výstup. Dále uvažuji o použití Raspberry PI B+ při vývoji dalších výpočetně náročnějších zvukových aplikací, kde ovšem budu pro kvalitnější zvukový výstup nutné použít externí DAC. Například Cirrus CS43L22, který je použit přímo na STM32F4 Discovery.

2.6.5 Vývojové prostředí CooCox CoIDE

CoIDE je vývojové prostředí pro ARM-Cortex M4, M3 a M0. Výhodou CoIDE oproti jiným vývojovým prostředím je neomezené množství dat u verze zdarma ke

stažení [12], které může být nahráno do paměti MCU. CoIDE nemá integrovaný překladač. Je tedy nutné nainstalovat a v CoIDE nastavit ARM GNUCompiler.



Obr. 2.6.5.1 Vývojové prostředí Coocox CoIDE

CoIDE se skládá z pěti hlavních oken. Vlevo nahoře je okno „Components“ kde je vidět typ nastaveného MCU. Dále se v tomto okně nachází seznam připojených knihoven, které lze přidávat, nebo odebírat v okně „Repository“. Každá knihovna obsahuje i několik příkladů. V okně „Project“ je seznam souborů, které jsou k projektu připojeny a budou kompilovány. Dvojklikem je možné obsah příslušného souboru zobrazit a případně upravit v hlavním okně a uložit. V prostředním hlavním okně lze psát kód jednotlivých souborů a provádět nastavení. Po kliknutí na knihovnu v okně „Components“ se v okně „Help“ zobrazí nápověda funkcí příslušné knihovny. Nakonec okno „Console“ informuje o stavu kompilace, nahrání do paměti procesoru a případných chybách.

Přehled nejdůležitějších funkcí hlavní lišty:

 vytvoření nového projektu

 vytvoření nového C nebo H souboru

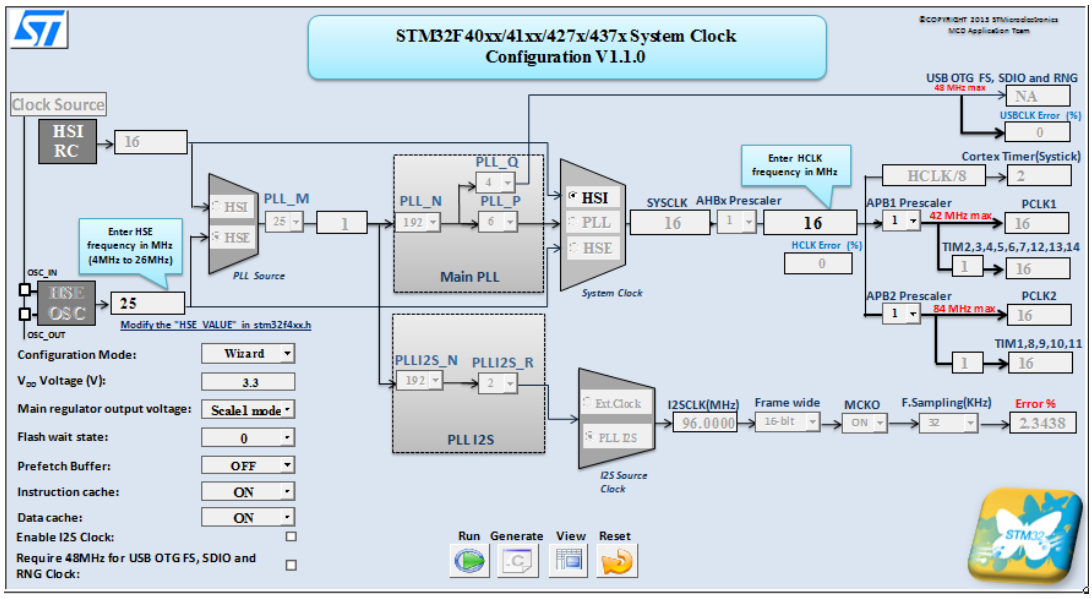
 zkompilování kódu

 ladění, debugger

 nahrání zkompilovaného kódu do paměti MCU

 nastavení

K tomu aby byl CoIDE schopen zkompilovat kód je nutné mít nainstalovaný ARM GCC kompiler (nejlépe poslední verze). Dále po instalaci GCC je třeba nastavit v CoIDE poject − > Select Toolchain Path složku “bin” v adresáři, kde je nainstalován kompiler GCC. Při novém vytvoření projektu se v adresáři projektu vytvoří systémový soubor system_stm32f4xx.c. Tento soubor obsahuje mimo jiné nastavení systémového času a větví nazvaných AHB1 a AHB2. Z větví APB1 a APB2 je potom stanoven takt například pro MIDI, nebo časovače. Firma STMicroelectronics vyvinula jednoduchý program pro excel (obr. 2.6.5.2), kde lze snadno nastavit systémový čas, APB1 a APB2 prescaler, takt externího oscilátoru a další. Tento program po nastavení vygeneruje soubor system_stm32f4xx.c. Tímto souborem je nutno nahradit soubor stejného názvu u nově vytvořeného projektu.



Obr. 2.6.5.2 Konfigurační nástroj STMicroelectronics [13]

3 Vlastní řešení

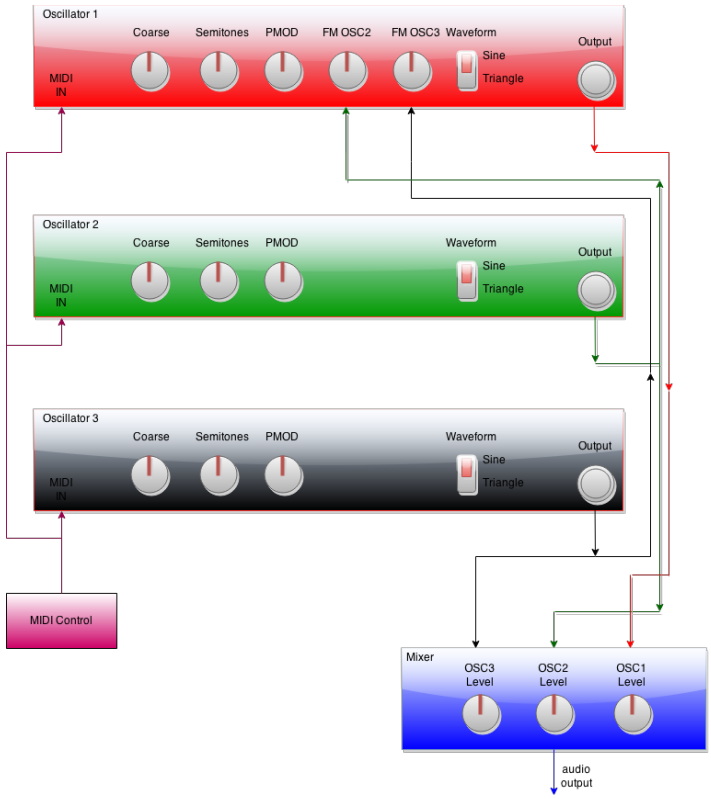
Tato část práce je věnována mému vlastnímu návrhu a popisu transformace návrhu ve funkční celek. Postupně bude představen návrh řešení a jeho hardwarová a softwarová realizace. Prostor bude věnován také měření spektrálních charakteristik základních zvukových signálů různých typů digitálních oscilátorů realizovaných na STM32F4 Discovery a jejich porovnání s oscilátorem komerčně vyráběného digitálního syntezátoru a také porovnání s vlastnostmi analogového signálu z VCO.

Obr. 3.1 blokové schéma mého vlastního návrhu softwarové části syntezátoru. Syntezátor se skládá ze třech identických MIDI řízených digitálních oscilátorů. Oscilátory jsou identické, až na to že oscilátory 2 a 3 nemohou být frekvenčně modulovány, z důvodu nestability systému. Ovladače mají následující funkce:

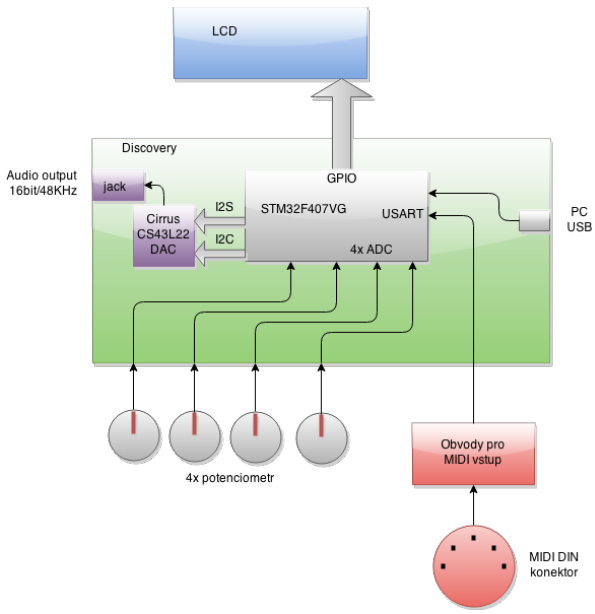
- COARSE (oktávy) - přepíná frekvenci +/- dvě oktávy
- SEMITONES (půltóny) – rozsah 0 až +12 půltónů (oktáva)
- OSC1-3 LEVEL – nastavení amplitudy výstupního signálu jednotlivých oscilátorů
- FM OSCx – nastavení hloubky frekvenční modulace oscilátorů 2 a 3, nastavuje modulační index
- WAVEFORM – volba výstupního signálu, sinus, nebo trojúhelník
- PMOD – ovladač pro nastavení hloubky modulace výšky tónu

Výstupní signál z oscilátorů 2 a 3 je přiveden na vstup pro frekvenční modulaci oscilátoru 1. Tyto signály jsou také přivedeny na výstupní směšovač, takže je možné kombinovat i s frekvenčně modulovaným signálem. Důležitý je i ovladač PMOD, který umožňuje nastavit hloubku modulace frekvence, ovšem ne na audio frekvenci, jako je tomu u FM vstupů, ale tento ovladač slouží pro pomalé změny frekvence. Pomocí ovladače PMOD je možné docílit dynamických změn spektra. Dynamické změny lze napodobit některým z modulačních generátorů, jako například nízkofrekvenční oscilátor, generátor obálky, nebo například sekvencer pro krokové změny. Bez PMOD modulace bude možné generovat pouze statická zvuková spektra, případě spektra dynamicky měnit manuálně, nikoli podle definovaných průběhů.

Aby bylo možné nastavovat parametry syntezátoru, zobrazovat hodnoty parametrů či přijímat MIDI zprávy, je nutné k STM32F4 Discovery připojit příslušný hardware. Následující obr. 3.2 ukazuje návrh připojení zobrazovače, potenciometrů, MIDI a audio výstupu na piny mikrokontroléru. V následujících kapitolách bude popsáno připojení tohoto hardwaru k STM32F406VTG6 procesoru a nastavení příslušných periferií. Další část vlastního řešení bude věnována softwarové části, digitálním oscilátorům, výsledkům z měření a frekvenční modulaci. Nakonec bude popsáno ovládání a návrh plošného spoje.



Obr. 3.1 Blokové schéma návrhu softwarové části

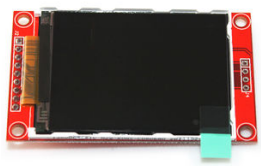


Obr. 3.2 Blokové schéma návrhu hardwarové části

3.1 Připojení zobrazovače k STM32F4 Discovery

Zobrazování hodnot parametrů je důležitou součástí syntezátoru. Jsou kladeny požadavky na dobrou čitelnost, jak co se týče velikosti znaků, tak i čitelnost z úhlu. Další důležitý požadavek je kladen na rychlost komunikace. Navíc při výběru zobrazovače byla požadována dostupnost knihovny a minimální počet nutných pinů MCU. Výběr byl posuzován mezi třemi typy zobrazovačů. První byl QVGA TFT SPI 240x320 bodů, druhý LCD zobrazovač 16 znaků, 2 řádky s řadičem HD44780 a třetí LCD zobrazovač ale 20 znaků, 4 řádky se stejným řadičem HD44780. Porovnání dle tab. 3.1.1.

Tab. 3.1.1 Porovnání zobrazovačů

Zobrazovač	výhody	Nevýhody
 **Obr. 3.1.1 QVGA TFT bisplej [14]**	SPI komunikace, barevný, obsahuje slot pro SD kartu, dostupná knihovna, počet použitých pinů MCU - 6	Pomalá komunikace, špatná čitelnost
 **Obr. 3.1.2 1602 LCD [15]**	Dobře čitelný, 16 znaků, 2 řádky, levný, dostupná knihovna, rychlá komunikace, použitých pinů MCU – 7(4bit mód)	Malá velikost zobrazovací plochy
 **Obr. 3.1.3 2004 LCD [15]**	Dobře čitelný, 20 znaků, 4 řádky, dostupná knihovna, rychlá komunikace, použitých pinů MCU – 7(4bit mód)	Vyšší cena oproti 1602 LCD

Dle tab. 3.3.1 byl zvolen displej QC2004A (obr. 3.1.3) z důvodu rychlé komunikace, dobré čitelnosti a velikosti zobrazovací plochy. QC2004A má pro připojení vyvedené piny 1-16 v horní části plošného spoje. Následující tab. 3.1.2 popisuje funkci pinů LCD a připojení na piny STM32F4 Discovery.

Tab. 3.1.2 Připojení QC2004A k STM32F4 Discovery

QC2004A pin	Funkce pinu	DISCOVERY pin
1	GND	GND
2	5V	5V
3	kontrast	trimr
4	RS	PB4
5	RW (read, write)	GND
6	Enable	PB7
7	D0	NC
8	D1	NC
9	D2	NC
10	D3	NC
11	D4	PC9
12	D5	PC13
13	D6	PC14
14	D7	PC15
15	Podsvícení anoda	3V
16	Podsvícení katoda	GND

Pro softwarovou komunikaci s QC2004A je k dispozici knihovna HD44780 [16] pro jazyk C. Tato knihovna se skládá ze dvou souborů, tm_stm32f4_hd44780.h a tm_stm32f4_hd44780.c. První soubor je nutno zahrnout do projektu pomocí příkazu include.
V tm_stm32f4_hd44780.h je také nutno nastavit správné piny dle tab. 3.1.2.

Před použitím knihovny je nutné inicializovat LCD na 20 znaků, 4 řádky příkazem:

```
TM_HD44780_Init(20, 4);
```

Po inicializaci lze zapisovat znaky z tabulky ASCII. V mém případě po zapnutí DISCOVERY, než program přejde do nekonečné smyčky, se pomocí následujících příkazů na QC2004A zobrazí text v uvozovkách.

```
TM_HD44780_Puts(0, 0, "STM32F4 Discovery");
TM_HD44780_Puts(0, 1, "FM Synthesizer");
TM_HD44780_Puts(0, 2, "Pavel Vancura");
TM_HD44780_Puts(0, 3, "Bakalarska prace");
```

První parametr ve funkci TM_HD44780_Puts() určuje posun zleva, rozsah je 0-19. Druhý parametr definuje řádek, 0-3. Třetí je přímo text, který se zobrazí, musí být v uvozovkách.

Poslední nutný příkaz z této knihovny je příkaz pro vymazání textu displeje.

```
TM_HD44780_Clear();
```

Další užitečné příkazy této knihovny lze najít na stránkách autora knihovny [16].

3.2 MIDI vstup pro STM32F4 Discovery

Protokol MIDI je standardem pro digitální komunikaci hudebních nástrojů. Dle kapitoly 2.1 může MIDI zpráva obsahovat různé parametry a jejich nastavení. Využití protokolu MIDI bude omezeno pouze na příkaz nota zapnuta. První byte tohoto příkazu má hodnotu 144 a je následován bytem, jehož hodnota representuje číslo stisknuté noty. Tyto dva příkazy umožní nastavovat digitální oscilátory na přesný hudební kmitočet. Jiné příkazy v této práci nejsou třeba. Aby bylo možné připojit MIDI zařízení (např. klaviaturu) k DISCOVERY, je nutné připojit několik nezbytných komponent. Schéma zapojení je v příloze A (hlavní schéma zapojení). Jednoduchý obvod se skládá z DIN konektoru, 6N138 optočlenu U4, třech odporů R5, R6, R7 a diody D1. MIDI signál je připojen na piny 2 a 4 DIN konektoru. Příslušným kabelem se obě zařízení propojí. Optočlen U4 je použit ze dvou důvodů. Elektricky izolovat obě zařízení, a proudově nezatěžovat zařízení, které vysílá data. Odizolovaný MIDI signál o úrovních 0 a +5V je připojen na pin PA3 pin DISCOVERY, který je přímo vstupem USART mikrokontroleru.

Zpracovaní MIDI dat provádí USART2 MCU. USART (universální synchronní asynchronní vysílač přijímač) je popsán v referenčním manuálu RM0090 [11]. Nejprve je třeba nakonfigurovat pin PA3 na který je MIDI vstup připojen. GPIO piny mohou pracovat v různých módech dle [11]. GPIO piny lze nastavovat přímo zápisem do příslušných registrů, nebo pomocí CMSIS [17]. Následující kód se nachází v souboru main_setup.c. První řádek definuje strukturu. Dále zápisem do struktury se definuje číslo pinu, alternující mód vstupního pinu, GPIO frekvence, push pull rezistor a pull up rezistor. Poté se struktura inicializuje. Nakonec se pin PA3 připojí na periferii USART2.

```
//USART2 PA3 RX pin (MIDI in)
GPIO_InitTypeDef GPIO_InitStruct;
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_3; // Pin PA3
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AF; //pin v AF módu
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStruct.GPIO_OType = GPIO_OType_PP;//push pull mód
GPIO_InitStruct.GPIO_PuPd = GPIO_PuPd_UP; //pull up rezistor aktivován
GPIO_Init(GPIOA, &GPIO_InitStruct); //inicializace struktury
GPIO_PinAFConfig(GPIOA, GPIO_PinSource3, GPIO_AF_USART2);//připojení PA3 RX
    pin na USART2 GPIOA, GPIO_PinSource3, GPIO_AF_USART2
```

Při inicializaci MCU jsou všechna zařízení (periferie) vypnutá, včetně GPIO, kvůli úspoře energie, je třeba zapnout napájení příslušných periferií dle [11]. Následující příkaz zapne port GPIOA a USART2, tentokrát přímým zápisem do RCC registru, bez použití CMSIS.

```
RCC->AHB1ENR |= (RCC_AHB1ENR_GPIOAEN | RCC_AHB1ENR_USART2EN);
```

Nastavení samotného USART2, které popisuje následující kód je zapsán na řádcích 84-104 souboru main_setup.c. Po definování struktury se nastaví frekvence na 31,25 kHz,tj. pracovní kmitočet MIDI jak je popsáno v kapitole 3.1. Dále podle 3.1 se nastaví délka jednoho přijatého digitálního slova na byte a USART2 se nastaví do módu přijímače. Poté se struktura inicializuje.
Příkaz USART_ITConfig povoluje vyvolání přerušení, pokud se v USART2 registru

nachází nový byte. Posledním příkazem se USART2 uvede do činnosti.

```
USART_InitTypeDef USART_InitStruct;
USART_InitStruct.USART_BaudRate = 31250;//pracovní kmitočet USART 31.25KHz
USART_InitStruct.USART_WordLength = USART_WordLength_8b;//délka slova byte
USART_InitStruct.USART_StopBits = USART_StopBits_1; //1 stop bit
USART_InitStruct.USART_Parity = USART_Parity_No;//bez paritního součtu
USART_InitStruct.USART_Mode = USART_Mode_Rx; //USART v režimu přijímače
USART_Init(USART2, &USART_InitStruct);//inicializace struktury
USART_ITConfig(USART2, USART_IT_RXNE, ENABLE); //povolení přerušení
    USART_Cmd(USART2, ENABLE); //zapnutí USART2
```

Pro vyvolání přerušení na příchozí byte je třeba ještě nastavit NVIC (Nested Vectored Interrupt Controller) dle [11]. V příslušném kanále je uveden název funkce přerušení USART2_IRQn, nastaví se priorita a nakonec se kanál uvede v činnost.

```
NVIC_InitTypeDef NVIC_InitStructure;
NVIC_InitStructure.NVIC_IRQChannel = USART2_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
```

Výše uvedeným nastavením USART2, GPIOA, NVIC a zapojením dle schématu 1 je možné přijímat a zpracovávat MIDI data přicházející na pin PA3 STM32F407VG. Po stisku klávesy vyšle připojení MIDI zařízení (klaviatura) byte s číslem 144. Tento byte je uložen v DR registru [11] USART2. Ihned poté se volá přerušení a provede se následující kód.

```
void USART2_IRQHandler(void){
char val = USART2->DR;
if(val == 144){
note_on = true;}
if(note_on == 1 && val >= 24 && val <= 119){
freq1 = set_frequency(val);
note_on = false;
}}
```

Při prvním volání funkce přerušení se vyzvedne hodnota z DR registru USART2 a uloží se do proměnné val. Následně se testuje, jestli se v proměnné val nachází hodnota 144. Právě tato hodnota znamená dle kapitoly 2.1 příkaz nota zapnuta. Pokud ano, nastaví se proměnná note_on na hodnotu true. Dle kapitoly 2.1 je kontrolní byte bezprostředně následován bytem s hodnotou čísla noty. Volá se tedy znovu tato funkce, protože registr DR obsahuje nový byte. Tentokrát je splněna podmínka note_on = true a pokud se číslo noty nachází v rozmezí 24-119, zavolá se funkce set_frequency(val), jejíž argumentem je číslo noty. Tato funkce spočítá frekvenci a výsledek se uloží do globální proměnné freq1, která je potom použita k nastavení frekvence NCO. Rozmezí not 24-119 je C1 až C9, frekvenčně 32.7 Hz až 7,901 KHz.

3.3 Připojení potenciometrů k STM32F4 Discovery

Hlavní funkcí potenciometru je nastavovat parametry syntezátoru. Připojení potenciometru k MCU znázorňuje hlavní schéma zapojení, příloha A. Hodnota odporové dráhy potenciometru je 10 kΩ. Hodnotu lze volit libovolně v řádech kilo ohmů, nejlépe s lineární charakteristikou. Potenciometr je zapojen mezi napětí 0 a 3 V. Na jezdci potenciometru, který je zapojen na pin MCU se mění napětí v rozsahu 0-3 V. Toto napětí na pinech je vzorkováno pomocí ADC a ukládáno do proměnné. Ta je potom použita k nastavení určitého parametru. Velmi zajímavou funkcí mikrokontroleru STM32F407VG je DMA (Direct Memory Access) popsána v [11]. Pomocí DMA lze ukládat hodnoty potenciometrů do proměnných bez jediného taktu procesoru. Ukládání se děje na pozadí, zatímco procesor vykonává jinou činnost. DMA lze využít také u např. USART, DAC atp. Dle schématu v příloze A jsou zapojeny čtyři potenciometry na piny PC1, PC2, PC4 a PC5. Tyto piny jsou nastaveny jako vstup v analogovém módu. Více o analogovém módu pinu v [11]. Následující nastavení pomocí CMSIS [17] je pro pin PC1. Pro zbylé tři piny je nastavení totožné. Liší se pouze číslo pinu.

```
GPIO_InitTypeDef GPIO_ADC4_InitStruct;
GPIO_ADC4_InitStruct.GPIO_Pin = GPIO_Pin_2;//pin
GPIO_ADC4_InitStruct.GPIO_Mode = GPIO_Mode_AIN;//vstup analogový mód
GPIO_ADC4_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_ADC4_InitStruct.GPIO_PuPd = GPIO_PuPd_NOPULL; //bez pushpull
GPIO_Init(GPIOC, &GPIO_ADC4_InitStruct); //inicializace
```

Dále je třeba zapnout napájení portu GPIOC, ADC1 a DMA2 zápisem do RCC registru [11].

```
RCC->APB2ENR |= RCC_APB2ENR_ADC1EN;
RCC->AHB1ENR |= (RCC_AHB1ENR_DMA2EN | RCC_AHB1ENR_GPIOCEN);
```

Následující kód ukazuje nastavení ADC1. Detailní informace o nastavení jsou popsány v dokumentaci [11]. První řádek po definování názvu struktury je zarovnání dat. Protože ADC1 DR registr je 32-bit a rozlišení převodníku je nastaveno na 8-bit, data budou zarovnána vpravo. Dále je povolen nepřetržitý režim, to znamená, že po dokončení jedné konverze následuje hned další konverze. Protože jsou zapojeny čtyři potenciometry, jen nastaven počet konverzí ADC_NbrOfConversion na čtyři. Poté je povolen scan režim. Scan režim umožňuje konverzi jednoho kanálu, když je dokončena, konvertuje se druhý kanál, dále třetí, čtvrtý a pak zase první. Po inicializaci struktury se uvede ADC1 v činnost. Nakonec pomocí příkazu ADC_RegularChannelConfig se přiřadí příslušnému kanálu číslo převodu a vzorkovací kmitočet. Kanál 11, 12, 14, 15 odpovídají pinům PC1, PC2, PC4, PC5.

```
ADC_DeInit();
ADC_InitTypeDef ADC1_InitStructure;
ADC1_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;//zarovnání
ADC1_InitStructure.ADC_Resolution = ADC_Resolution_8b;//rozlišení 8b
ADC1_InitStructure.ADC_ContinuousConvMode = ENABLE;//nepřetržitý rež.
ADC1_InitStructure.ADC_ExternalTrigConvEdge = ADC_ExternalTrigConvEdge_None;
ADC1_InitStructure.ADC_NbrOfConversion = 4;
ADC1_InitStructure. ADC_ScanConvMode = ENABLE;//scan režim
ADC_Init(ADC1, &ADC1_InitStructure);
```

```
ADC_Cmd(ADC1, ENABLE);//spuštění ADC1
ADC_RegularChannelConfig(ADC1, ADC_Channel_11, 1, ADC_SampleTime_84Cycles);
ADC_RegularChannelConfig(ADC1, ADC_Channel_14, 2, ADC_SampleTime_84Cycles);
ADC_RegularChannelConfig(ADC1, ADC_Channel_15, 3, ADC_SampleTime_84Cycles);
ADC_RegularChannelConfig(ADC1, ADC_Channel_12, 4, ADC_SampleTime_84Cycles);
```

Poslední nastavení je provedeno u výše zmíněného DMA. Postup při nastavení a použití DMA je popsán v dokumentaci [11]. První zápis do struktury po jejím definování je nastavení ukazatele na adresu periferie. V tomto případě je to DR registr ADC1. Další příkaz je nastavení ukazatele na adresu tentokrát v paměti. Na tuto adresu se bude ukládat hodnota z DR registru ADC1. Příkaz DMA_DIR je nastavení typu převodu, v tomto případě z periferie do paměti. Velikost bufferu je 4, protože jsou zapojené 4 potenciometry. Dále je vypnuto inkrementování adresy periferie, ale důležitý je následující řádek, kde je povoleno inkrementování adresy paměti. To zajistí, že hodnota z potenciometru 1 se bude ukládat do prvku pole s indexem 0, hodnota potenciometru 2 do prvku pole s indexem 1, hodnota potenciometru 3 do prvku pole s indexem 2 a hodnota potenciometru 4 do prvku pole s indexem 3. Dále je nastavena velikost přenášených dat. Jelikož je definováno pole typu 16bit, přenáší se hodnota 16bit. DMA_Mode_Circular znamená, že se po dokončení transferu 4 do pole s indexem 3, se opět začne provádět transfer 1 do pole s indexem 0. Poté se proces opakuje.

Dále už je jen nastavení priority a nastavení DMA kanálu. Každá periferie má svůj DMA kanál, na kterém je schopna přenášet data. Tento DMA kanál ukazuje tabulka 42 a 43 v dokumentaci [11]. Využitelný je i double buffer mód popsaný v dokumentaci [11]. V tomto módu probíhá přenos dat na dvě různá místa v paměti. To má výhodu ve zpracování ukládaných dat, protože nelze zpracovávat hodnotu, když probíhá DMA transfer. Pokud se začne manipulovat s místem v paměti při probíhajícím transferu, tak DMA tok vyhlásí chybu a je nutné celý DMA reinicializovat. V double buffer módu probíhá transfer střídavě do dvou různých adres v paměti. Jedna adresa se může zpracovávat, zatímco transfer probíhá do druhé adresy a naopak. Zbylé příkazy v kódu níže jsou už jen inicializace a uvedení DMA v činnost.

```
DMA_InitTypeDef DMA_CH0_InitStruct;
DMA_CH0_InitStruct.DMA_PeripheralBaseAddr = (uint32_t)&ADC1->DR;
DMA_CH0_InitStruct.DMA_Memory0BaseAddr = (uint32_t)&ADC_Buffer;
DMA_CH0_InitStruct.DMA_DIR = DMA_DIR_PeripheralToMemory;
DMA_CH0_InitStruct.DMA_BufferSize = 4;
DMA_CH0_InitStruct.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_CH0_InitStruct.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_CH0_InitStruct.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
DMA_CH0_InitStruct.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
DMA_CH0_InitStruct.DMA_Mode = DMA_Mode_Circular;
DMA_CH0_InitStruct.DMA_Priority = DMA_Priority_High;
DMA_CH0_InitStruct.DMA_Channel = DMA_Channel_0;
DMA_DoubleBufferModeConfig(DMA2_Stream0, (uint32_t)&ADC_Buffer1,
    DMA_Memory_0);
DMA_DoubleBufferModeCmd(DMA2_Stream0, ENABLE);
DMA_Init(DMA2_Stream0, &DMA_CH0_InitStruct);
DMA_Cmd(DMA2_Stream0, ENABLE);
ADC_DMACmd(ADC1, ENABLE);
```

3.4 Audio výstup

STM32F4 Discovery, jak bylo zmíněno v kapitolách 2.2 a 2.4, obsahuje digitálně analogový převodník Cirrus CS43L22 s rozlišením až 24 bit/96 kHz. CS43L22 není součástí mikrokontroleru STM32F407VG, ale nachází se mimo na plošném spoji Discovery, obr. 2.6.2.1. Zapojení s piny mikrokontroleru a ostatními obvody ukazuje schéma v příloze A. K CS43L22 je přímo zapojen audio výstup (obr. 2.6.2.1) buďto pro sluchátka, nebo pro aktivní reproduktory. CS43L22 se nastavuje přes sériovou sběrnici I2C [11] a digitální zvukový signál je přenášen sériovou zvukovou sběrnicí I2S [11]. Je možné přenášet kvalitní digitální audio signál až o hloubce 24bit a vzorkovacím kmitočtem až 96 KHz. Detailní informace o nastavení CS43L22 jsou k nalezení v dokumentaci [18].

Přenášet data přes I2S je jedna z možností, jak získat z DISCOVERY kvalitní audio signál. Existuje však ještě další možnost. STM32F407VG má integrované dva digitálně analogové převodníky. Převodník DAC1 je vyveden na výstup PA4 a převodník DAC2 je vyveden na pin PA5. Tyto převodníky mají 12bit rozlišení. Při pohledu na schéma zapojení CS43L22 v příloze A je vidět, že pin PA4 je zapojen na CS43L22 pin 29 a 30. To je proto, že CS43L22 může pracovat v tzv. „analog thru“ módu. V tomto módu prochází analogový signál z pinů 29 a 30 CS43L22 (tedy i z pinu PA4, kde je zapojena DAC1) přímo na sluchátkový výstup. Lze tedy převádět digitální signál přes integrované digitálně analogové převodníky a lze tento signál přivést i na sluchátkový výstup Discovery.

Pro úplnost budou uvedeny obě varianty nastavení pro zvukový výstup. Nejprve bude uvedeno nastavení pro 12-bit interní DAC a nastavení CS43L22 do analog thru módu. Pin DAC1 PA4 je nastaven jako výstupní v analogovém módu dle následujícího kódu.

```
GPIO_InitTypeDef GPIO_DAC1_InitStruct;
GPIO_ADC4_InitStruct.GPIO_Pin = GPIO_Pin_4;
GPIO_ADC4_InitStruct.GPIO_Mode = GPIO_Mode_AN;
GPIO_ADC4_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_ADC4_InitStruct.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_Init(GPIOA, &GPIO_DAC1_InitStruct);
```

Nastavení DAC v STM32F407VG je jednoduché. Podrobný popis a funkce DAC jsou popsány v dokumentaci [11]. Následující kód je přímý zápis do registrů bez použití CMSIS [17]. Nejprve je třeba zapnout napájení DAC, to je provedeno na prvním řádku zápisem do RCC registru. Poté se zapne DAC1 nastavením bitu EN1 v DAC1 CR registru. Na následujícím řádku je zapnutí výstupního bufferu vynulováním bitu BOFF1 a vypnutí triggeru. V případě, že není použit žádný trigger a hodnota, která má být konvertována se nachází v DHR12R1 registru DAC1, bude tato hodnota konvertována při následujícím taktu procesoru. Pokud by byl použit trigger z nějakého zdroje, tak právě tento trigger by určoval, kdy bude hodnota konvertována. Na posledním řádku už je jen pro jistotu nulován bit WAVE1 pro generování vlnové formy.

```
RCC->APB1ENR |= RCC_APB1ENR_DACEN;
DAC->CR |= DAC_CR_EN1;
DAC->CR &= ~(DAC_CR_BOFF1 | DAC_CR_TEN1);
DAC->CR &= ~(DAC_CR_WAVE1);
```

Přesto, že byly uvedeny dvě možnosti, jak získat z DISCOVERY analogový audio signál, kvůli vyššímu rozlišení a vyššímu SNR byl použit integrovaný převodník CIRRUS CS43L22. Pro nastavení přes I2C a posílání dat přes I2S byla použita knihovna [19]. CIRRUS DAC je nakonfigurována na 16 bit/48 kHz mód odpovídající zadání.

```
codec_init();
codec_ctrl_init();
I2S_Cmd(CODEC_I2S, ENABLE);
```

První funkce codec_init() nastaví potřebné piny a periferie I2C a I2S. Druhá funkce codec_ctrl_init() nakonfiguruje CS43L22 pomocí I2C zpráv. Poté se uvede v činnost I2S.

Nyní jsou všechna potřebná zařízení nakonfigurována a už stačí jen posílat data DAC přes I2S pomocí následující funkce.

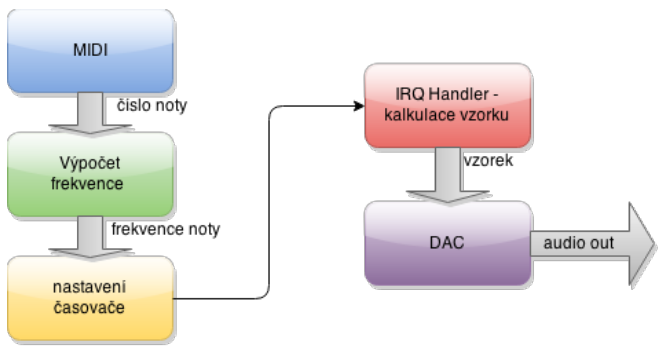
```
SPI_I2S_SendData(CODEC_I2S, (uint_16t)output);
```

3.5 Realizace digitálních oscilátorů na STM32F4 Discovery

Tato kapitola bude věnována možnostem, jak realizovat digitální NCO na STM32F4 Discovery. Budou představeny tři odlišné principy. První bude „wavetable“ oscilátor s variabilním vzorkovacím kmitočtem, dále bude následovat princip „wavetable“ oscilátoru s fázovým akumulátorem a nakonec oscilátor využívající pro generování vlny matematické funkce. Cílem práce v této části není dosáhnout digitálního oscilátoru se špičkovými parametry, ani změřit přesné hodnoty. Cílem je porovnat a vyzkoušet různé metody a přibližně změřit, případně poslechem ohodnotit kvality jednotlivých metod. Zejména šum a také výpočetní náročnost. Vybraný algoritmus bude poté použit pro další práci. Výsledky všech metod včetně jejich spekter budou porovnány na konci této kapitoly s VCO i komerčně vyráběným digitálním syntezátorem Clavia Nord Modular G2. Měření budou provedena softwarovým analyzátorem ARTA [20] přes AD převodníky M-Audio Delta 1010. Technické parametry M-Audio Delta 1010 - příloha B.

3.5.1 „Wavetable“ oscilátor s variabilním vzorkovacím kmitočtem

Princip oscilátoru s variabilním vzorkovacím kmitočtem byl popsán v kapitole 2.3.1. Princip softwarové realizace ukazuje následující blokový diagram na obr. 3.5.1.1.



Obr. 3.5.1.1 Blokové schéma oscilátoru s variabilním vzorkovacím kmitočtem

Poté co dojde k přijetí MIDI zprávy s číslem noty a volání funkce (modrý blok) popsané v kapitole 5.2, je z čísla noty vypočtena příslušná frekvence (zelený blok). Z frekvence je poté vypočten koeficient hodnoty do Auto-reload registru časovače (žlutý blok). Tento registr se nazývá ARR. Hodnota k, která musí být pro generování konkrétní frekvence f zapsána do ARR registru je spočítána dle následujícího vztahu.

$$k = \frac{T_{clk}}{W_{length} f} \tag{11}$$

Kde T_{clk} je frekvence taktu časovače. T_{clk} je dána nastavením busu AHB1. Nastavení AHB1 je patrné na obr. 2.6.5.2. T_{clk} je nastavena na 42 MHz. W_{length} je velikost vlnové tabulky, 2048 vzorků. Například pokud je požadovaná frekvence f=32,7 Hz, bude hodnota koeficientu k=627. Pokud časovač počítá o nuly, tak každý jeho takt bude hodnota v pracovním registru časovače zvýšena o jednotku. Až časovač dosáhne hodnoty 627, dojde k jeho přetečení a hodnota pracovního registru se vynuluje. Bude generováno přerušení (IRQ Hadler). Ve funkci přerušení dojde k předání jednoho vzorku pro DAC a inkrementování pozice o jednotku. Funkce přerušení je stále volána při hodnotě pracovního registru časovače 627 (pokud nedošlo ke změně frekvence), dochází k inkrementování ukazatele na pozici vzorku. Až dosáhne pozice posledního vzorku, hodnoty 2047, ukazatel se vrátí zpět na nulu a proces se opakuje. Protože 2048 vzorků tvoří jeden vlnový cyklus, je zajištěn periodický signál na výstupu.

Tímto způsobem lze generovat i složité periodické signály, záleží pouze na hodnotách vzorků na jednotlivých pozicích. Při frekvenci (nota C1) f=32,7 Hz, tedy hodnotě koeficientu koeficientu k=627, se volá funkce přerušení s frekvencí přibližně 69 KHz. Pro notu C2 je koeficient dvakrát menší a hodnota frekvence volání funkce přerušení dvakrát vyšší. Pro notu C6 je frekvence volání přerušení přibližně 414 kHz. Dle návrhu v kapitole 3 jsou třeba tři oscilátory. Je tedy nutné mít tři časovače a tři funkce přerušení, aby oscilátory mohly pracovat nezávisle. To je velmi výpočetně náročné a navíc je třeba kromě oscilátorů vykonávat další činnosti, jak například obsluha displeje. Tento problém částečně řeší metoda upsampling zmíněná krátce v kapitole 2.3.1. Upsampling jednoduše dělí vlnovou tabulku pro každou vyšší oktávu dvěma. To znamená pro první oktávu 2048 vzorků, pro druhou oktávu 1024 vzorků atd. Tímto způsobem se dá omezit maximální frekvence volání funkce přerušení na přibližně 125 kHz. Discovery má systémový čas 168 MHz, nicméně i frekvence volání přerušení 3x125 kHz je velmi výpočetně náročná. Metoda oscilátoru s variabilním vzorkovacím kmitočtem umožnila implementovat dva oscilátory na STM32F4 Discovery, obsluhu displeje, MIDI a další nezbytné úlohy. Tři oscilátory se vzhledem k výpočetní náročnosti nepodařilo realizovat. Navzdory výpočetní náročnosti má tato metoda výhodu v rovnoměrně rozestoupených vzorcích podél celého jednoho vlnového cyklu. Není tedy potřeba metoda interpolace vzorků.

Nastavení časovače (časovač 6) ukazuje následující kód. Na prvním řádku dojde k zapnutí napájení časovače zápisem do RCC registru. Poté zápisem do CR1 registru je nastaven bit ARPE, který nutí časovač obnovit novou hodnotu v ARR registru. Dále je vynulován bit OPM, tím čítač čítá stále dokola a nezastaví se po dosažení hodnoty v ARR registru. Také je vynulován bit UDIS který povoluje obnovení hodnoty v ARR registru buďto při přetečení čítače, nebo při nastavení UG bitu. Nastavení bitu UIE v DIER registru umožňuje generování přerušení při přetečení čítače, nebo nastavení UG bitu. PSC je registr pro nastavení děličky systémového času. Registr ARR jak již bylo zmíněno obsahuje hodnotu do jaké čítač čítá. Bit CEN v CR1 registru zapíná čítač.

Bit UG v EGR registru na posledním řádku naplní ARR registr hodnotou 627.

```
RCC->APB1ENR |= RCC_APB1ENR_TIM6EN;
TIM6->CR1 |= TIM_CR1_ARPE;
TIM6->CR1 &= ~TIM_CR1_OPM;
TIM6->CR1 &= ~TIM_CR1_UDIS;
TIM6->DIER |= TIM_DIER_UIE;
TIM6->PSC = 0;
TIM6->ARR = 627;
TIM6->CR1 |= TIM_CR1_CEN;
TIM6->EGR |= TIM_EGR_UG;
```

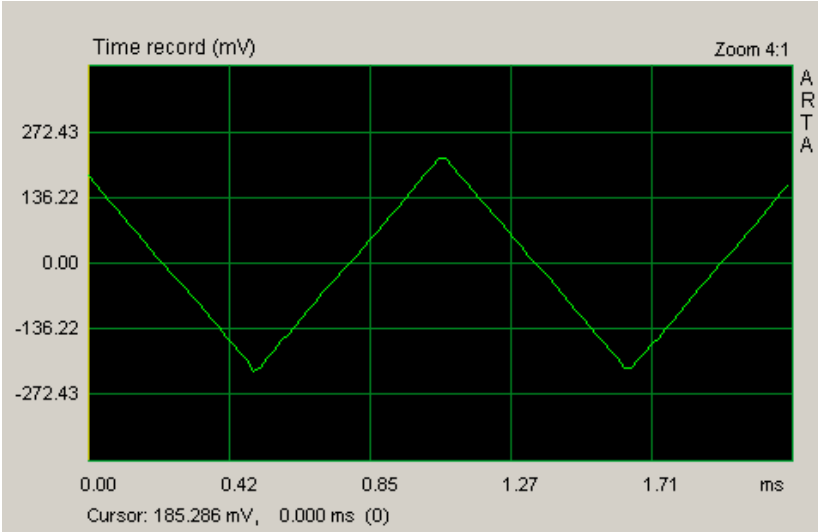
Nastavení časovačů pro jednotlivé oscilátory se liší pouze v čísle časovače. V tomto případě je použit časovač 6. Pro ostatní oscilátory jsou použity časovače 4 a 7. Jak již bylo zmíněno výše, STM32F407VG má 14 časovačů, ovšem všechny nejsou identické, viz [11]. Aby bylo možné používat funkci přerušení, je nutné nastavit také NVIC (Nested Vectored Interrupt Controller) dle [11]. V příslušném kanále je uveden název funkce přerušení TIM6_DAC_IRQn, nastaví se priorita a nakonec se kanál přerušení zapne.

```
NVIC_InitTypeDef NVIC_InitTIM6;
NVIC_InitTIM6.NVIC_IRQChannel = TIM6_DAC_IRQn;
NVIC_InitTIM6.NVIC_IRQChannelPreemptionPriority = 1;
NVIC_InitTIM6.NVIC_IRQChannelSubPriority = 1;
NVIC_InitTIM6.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitTIM6);
```

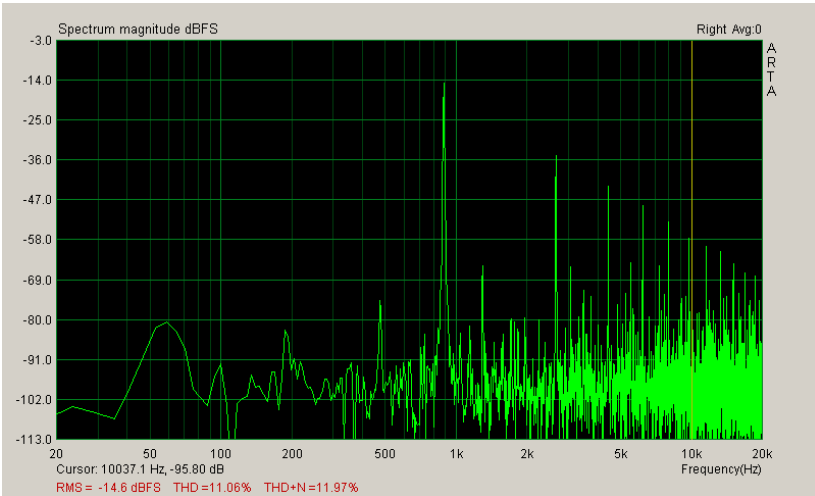
Zjednodušená funkce přerušení je uvedena v následujícím kódu. Na začátku funkce je nutné vynulovat bit UIF, jinak by se funkce přerušení volala stále znovu. Při prvním volání funkce se proměnná accu1 = 0. Do proměnné sigout1 bude tedy zapsán prvek na pozici nula pole wavetable1[0]. Velikost pole je pro první oktávu 2048 vzorků. Proměnná accu1 se při každém volání inkrementuje o jednotku a určuje pozici vzorku, který bude předán na výstup. Pokud je hodnota v accu1 vyšší nebo rovna 2047, zapíše se do accu1 zpět nula. To zajišťuje periodický signál na výstupu.

```
void TIM6_DAC_IRQHandler(void){
TIM6->SR &= ~TIM_SR_UIF;
sigout1 = wavetable1[accu1];
accu1++;
if(accu1 >= 2047){
accu1 = 0;
}}
```

Na obrázcích 3.5.1.2 a 3.5.1.3 je signál v časové i spektrální oblasti generovaný tímto typem oscilátoru. Jedná se trojúhelníkový signál o frekvenci 880 Hz.



Obr. 3.5.1.2 Signál generovaný oscilátorem s var. vzorkovacím kmitočtem

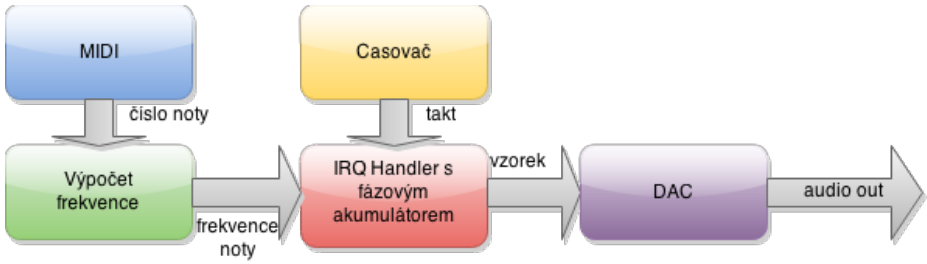


Obr. 3.5.1.3 Spektrum signálu na obrázku 3.5.1.2

Na obr. 3.5.1.3 je vidět základní harmonická složka (880 Hz) a vyšší harmonické ubývající v amplitudě s rostoucím číslem harmonické. Dále je patrný šum, který se zvyšuje s rostoucí frekvencí. Další zkoumání tohoto typu digitálního oscilátoru je kvůli výpočetní náročnosti bezvýznamné. Zvukovou kvalitu tohoto typu oscilátoru je možné posoudit v audio ukázce (Audio 9) na přiloženém CD.

3.5.2 „Wavetable“ oscilátor s fázovým akumulátorem

Blokové schéma zapojení „wavetable“ oscilátoru s fázovým akumulátorem je patrné z blokového diagramu na obr. 3.5.2.1. Hlavní rozdíl oproti metodě popsané v kapitole 3.5.1 je, že funkce přerušení IRQ Handler se volá s konstantním vzorkovacím kmitočtem. Vzorkovací kmitočet bývá obvykle u audio signálů 44.1 kHz – 96 kHz. Dosažení změny kmitočtu se provádí právě fázovým akumulátorem. Po přijetí MIDI zprávy s číslem noty (modrý blok) se zavolá funkce pro výpočet frekvence noty (zelený blok). Kmitočet noty je dále použit pro výpočet hodnoty pro fázový akumulátor. Hodnota fázového akumulátoru závisí na taktu časovače, frekvenci a velikosti vlnové tabulky dle vztahu (2) v kapitole 2.3.2.



Obr. 3.5.2.1 Blokové zapojení digitálního oscilátoru s fázovým akumulátorem

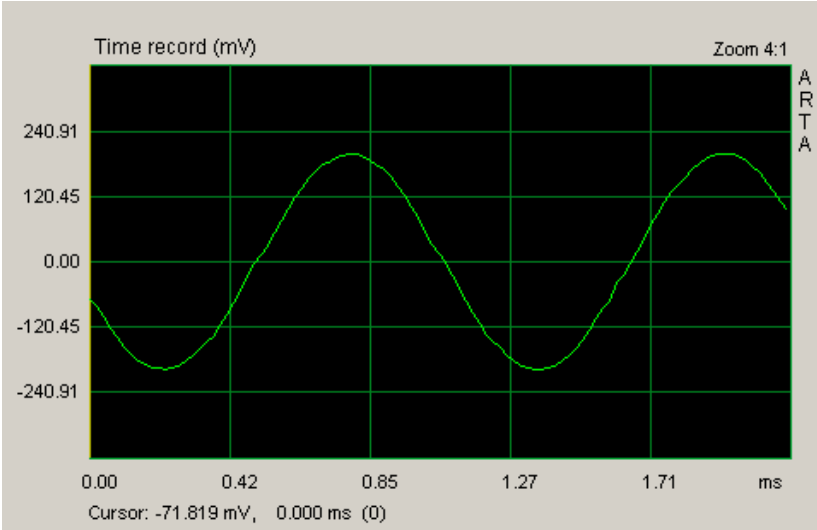
Nastavení časovače je totožné s nastavením u oscilátoru s variabilním vzorkovacím kmitočtem pouze s rozdílem, že do ARR registru je zapsána jedna neměnná hodnota.

```
RCC->APB1ENR |= RCC_APB1ENR_TIM6EN;
TIM6->CR1 |= TIM_CR1_ARPE;
TIM6->CR1 &= ~TIM_CR1_OPM;
TIM6->CR1 &= ~TIM_CR1_UDIS;
TIM6->DIER |= TIM_DIER_UIE;
TIM6->PSC = 0;
TIM6->ARR = 952;
TIM6->CR1 |= TIM_CR1_CEN;
TIM6->EGR |= TIM_EGR_UG;
```

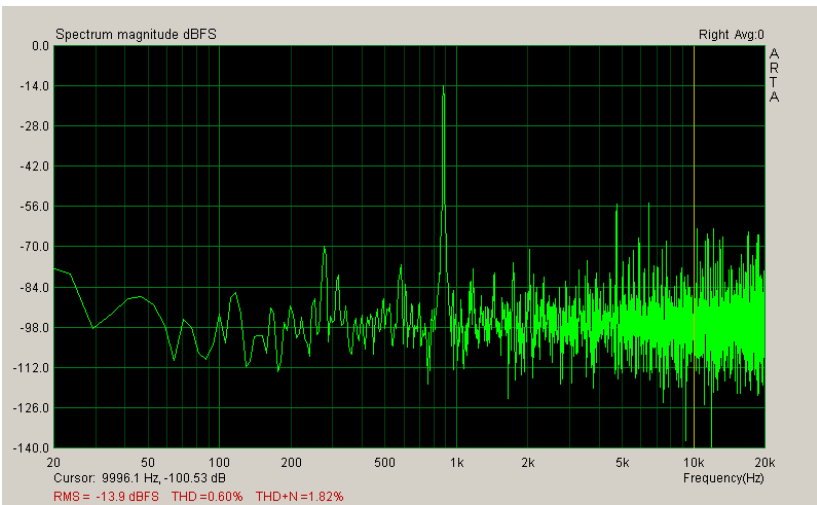
Hodnota 952 v ARR registru nastavuje volání funkce přerušení (IRQ Handler) s frekvencí $f = \frac{42000000}{952} = 44118Hz$. Pro oscilátory 2 a 3 jsou použity časovače 4 a 7 s identickým nastavením jako v případě časovače 6. Také je třeba nadefinovat NVIC pro všechny časovače dle postupu popsaného v kapitole 3.5.1. Ke změně frekvence oscilátoru dochází ve funkci přerušení pomocí fázového akumulátoru podrobně popsaného v kapitole 2.3.2. Následující kód ukazuje funkci přerušení a fázový akumulátor.

```
void TIM6_DAC_IRQHandler(void){
TIM6->SR &= ~TIM_SR_UIF;
increment1 = freq1/44100*2048;
sigout1 = wavetable1((int)phase);
phase1 = phase1 + increment1;
if(phase1 >= 2048){
phase1 = phase1 - 2048;
}}
```

Na prvním řádku funkce přerušení časovače 6, je vymazání příznakového bitu, jinak by se funkce přerušení volala po opuštění znovu. Následuje konstanta, kterou je nutno přičítat každé volání přerušení (hodnota fázového akumulátoru), aby při daném vzorkovacím kmitočtu (44.1 kHz) a velikosti tabulky 2048 vzorků, bylo dosaženo hudebního tónu, který je dán proměnnou freq1. Frekvenci lze změnit pomocí MIDI zprávy, nebo hodnotou z potenciometru, či jiného modulačního zdroje. Do proměnné sigout1 se ukládá výstupní vzorek, který se nachází na pozici vlnové tabulky určené hodnotou phase1 zaokrouhlené na celé číslo. Pokud hodnota phase1 přesahuje velikost vlnové tabulky, je odečtena hodnota 2048 pro generování periodického signálu. Následující obrázky 3.5.2.1 a 3.5.2.2 ukazují sinusový signál v časové a spektrální oblasti o frekvenci 880 Hz, který je generován oscilátorem s fázovým akumulátorem bez interpolace.



Obr. 3.5.2.1 Periodicky generovaný sinus digitálním oscilátorem s fázovým akumulátorem bez interpolace



Obr. 3.5.2.2 Spektrum signálu na obrázku 3.5.2.1

Na obrázku 3.5.2.2 je hlavní složka sinusového signálu o amplitudě -13,9 dBFS a frekvenci 880 Hz. Při této frekvenci bylo naměřeno softwarem ARTA THD+N=1,82%. To je vysoká hodnota v porovnání s komerčně vyráběnými digitálními syntezátory. Tato hodnota se s rostoucím kmitočtem zvyšuje. S klesajícím kmitočtem naopak klesá. Porovnání bude ukázáno v kapitole 3.5.4.

U oscilátoru s fázovým akumulátorem dochází k chybě při čtení z tabulky. Tuto chybu lze nejlépe vysvětlit následujícím příkladem. Máme tabulku o velikosti 2048 vzorků například sinu. Chceme generovat hudební tón o frekvenci 440 Hz. Vzorkovací kmitočet činní 44100 Hz. Koeficient, který je třeba přičítat fázovým akumulátorem je dán vztahem $\frac{440*2048}{44100} = 20,43$. Ovšem pro čtení z tabulky je třeba celé číslo. Proto je nutné zaokrouhlit tuto hodnotu buďto nahoru nebo dolů. Zde právě vzniká zmiňovaná chyba, která dle teoretické části znamená vyšší šum a nižší dynamický rozsah. Graficky je takto chybně vznikající hodnota vzorku vidět na obrázku 2.4.1. Správnou hodnotu

lze dopočíst lineární, případně kvadratickou interpolací. Kvadratická interpolace je výpočetně náročná. Proto bude zmíněna jen metoda lineární interpolace.

$$\frac{y_2 - y_1}{y - y_1} = \frac{x_2 - x_1}{x - x_1} \tag{12}$$

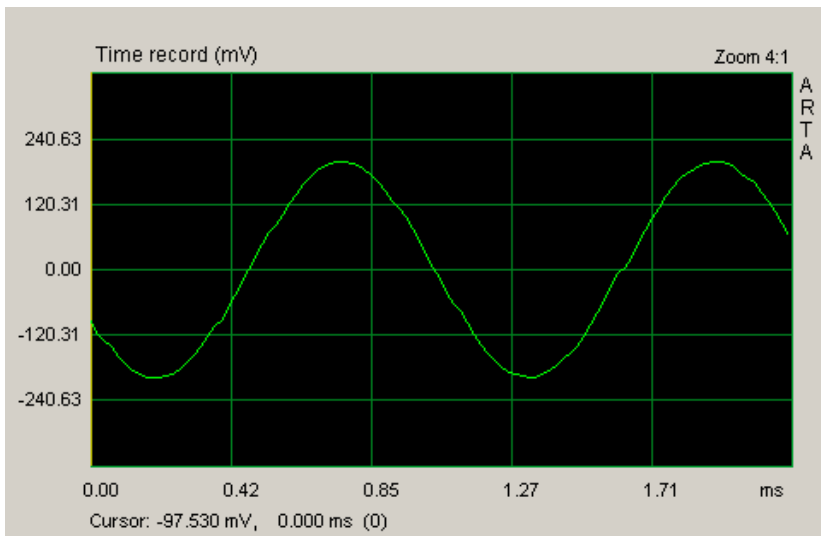
Kde x=20,43 je koeficient fázového akumulátoru. $x_2 = 21$ a $x_1 = 20$ jsou hodnoty zaokrouhlené nahoru a dolů. Potom y_1 a y_2 jsou vzorky vlnové tabulky odpovídající pozicím $x_2 = 20$ a $x_1 = 21$. Například $y_2 = 1144$ a $y_1 = 1100$. Potom hodnota y je lineární interpolací mezi vzorky y_1 a y_2.

$$y = \frac{(x - x_1)(y_2 - y_1)}{x_2 - x_1} + y_1 = \frac{(20,43 - 20)(1144 - 1100)}{21 - 20} = 1118,92 \tag{13}$$

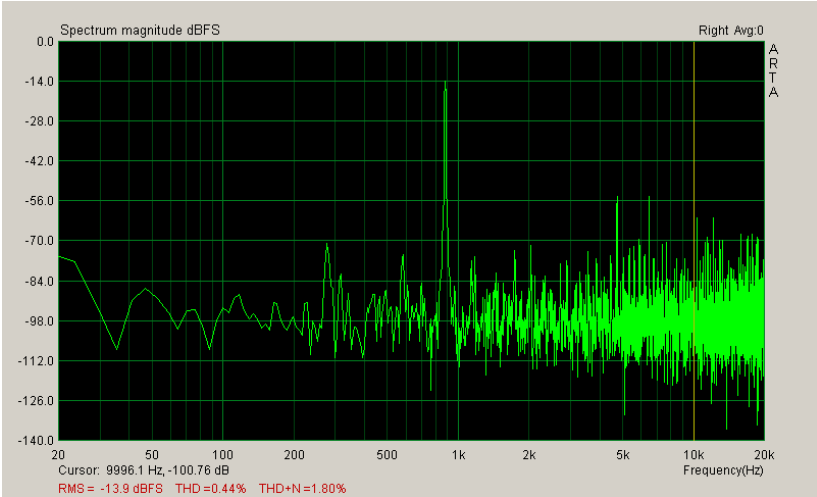
Provedení interpolace v programovacím jazyce C ukazuje následující kód. Funkce floorf zaokrouhluje desetinné číslo směrem dolů.

```
float x0 = frekvence/samplerate*2048;
int x1 = floorf(x0);
int x2 = x1 + 1;
int y2 = wavetable1[x2];
int y1 = wavetable1[x1];
y = (((y2 - y1)*(x0 - x1))/(x2 - x1)) + y1;
```

Interpolace mezi vzorky je nutná po menší velikosti tabulek. V mém případě pro velikost 2048 vzorků, není rozdíl v šumu rozsahu velký. To je vidět na následujících obrázcích 3.5.2.3 a 3.5.2.4.



Obr. 3.5.2.3 Periodicky generovaný sinus digitálním oscilátorem s fázovým akumulátorem s interpolací



Obr. 3.5.2.4 Spektrum signálu na obrázku 3.5.2.3

Hodnota THD+N na frekvenci 880 Hz při použití lineární interpolace je 1.8 %, jak je vidět na obrázku 3.5.2.4, což je téměř totožná hodnota jako bez použití interpolace (obr. 3.5.2.2). To je způsobeno tím, že chyba při čtení z tabulky při velikosti 2048 vzorků, není veliká. Odtud plyne zjištění, že pro vyšší velikost tabulky, není nutné interpolaci realizovat a tím lze ušetřit dost výpočetního výkonu pro jiné operace.

3.5.3 Oscilátor s matematickým modelem

V případě, že oscilátor bude generovat pouze základní periodické signály jako čtverec, trojúhelník, sinus, pilu, tak lze využít pro generování signálu jednoduchých matematických funkcí. Samotný fázový akumulátor uvedený v předchozí kapitole tvoří funkci, která generuje pilovitý signál. Trojúhelníkový signál je skoro stejný jako pilovitý s tím rozdílem, že náběžná hrana stoupá 2x rychleji a v polovině cyklu se hodnota začne odečítat a hrana klesá. Generování čtvercového signálu je jednoduché, stačí přepínat úrovně v přesně daných intervalech. Generování sinu je na rozdíl od předchozích obtížnější. Počítat hodnotu sinu v reálném čase je výpočetně náročné. Ke snížení výpočetní náročnosti lze využít Taylorovy řady s tím, že se použijí pouze první dva členy a zbylé se zanedbají.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} .... = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \tag{14}$$

Výpočet lze zjednodušit tím, že se místo Taylorova polynomu pro sinus, použije Taylorův polynom pro kosinus.

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} .... = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \tag{15}$$

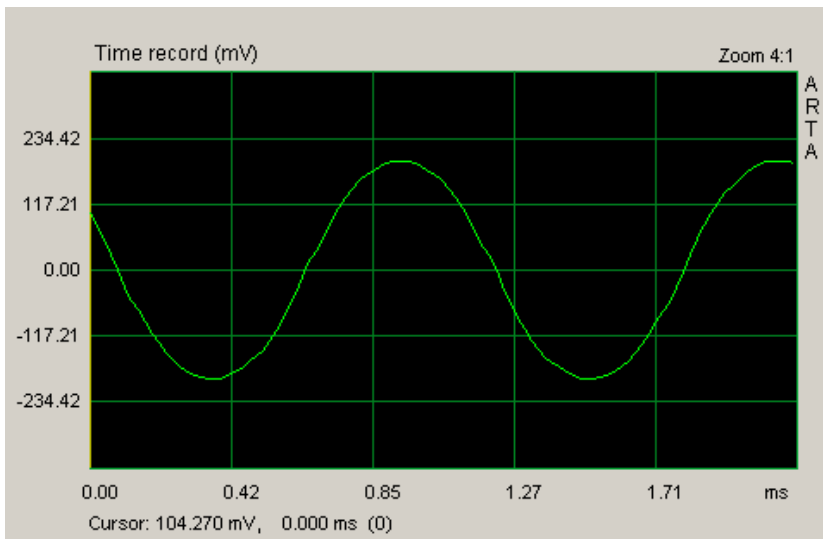
S tím, že se použijí pouze první dva členy rozvoje.

$$\cos(x) \doteq 1 - \frac{x^2}{2!} \tag{16}$$

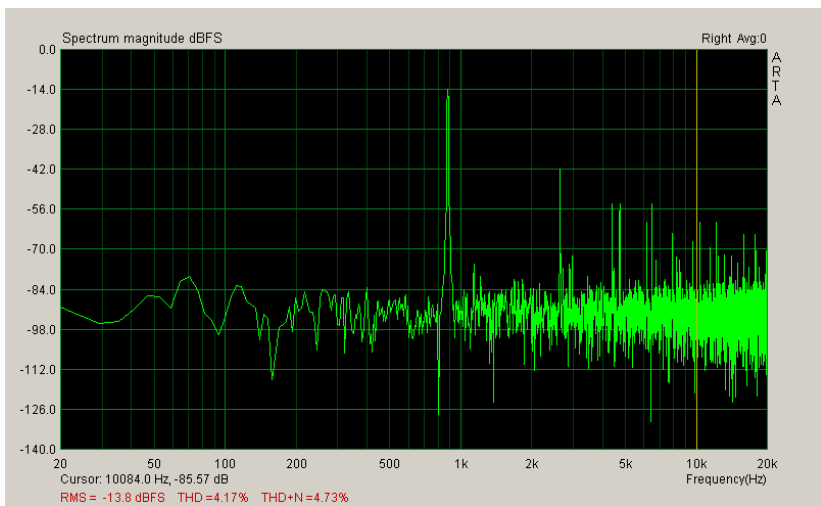
Následující kód je realizace sinu v prostředí CoIDE. Princip nastavení časovačů zůstává stejný jako u oscilátoru s fázovým akumulátorem. Následující část kódu je vykonávána při přerušení od přetečení časovače se vzorkovacím kmitočtem 44100Hz.

```
float increment3 = oscilparams3.freq/44100;
phase3 += increment3;
if(phase3 >= PI){
phase3 -= 6.283185;
sinea3 = ~sinea3;
}
if(sinea3==true){
oscilparams3.output = 2 + (phase3*phase3)/2;
}
else{
oscilparams3.output = 12 - (phase3*phase3)/2;
}
```

Na prvním řádku kódu se spočítá konstanta, která odpovídá požadované frekvenci. Tato konstanta se přičítá, dokud nedosáhne hodnoty π. Poté se přepne výstup z konvexní paraboly na konkávní. Změřený signál v časové a spektrální oblasti ukazuje obrázek 3.5.3.1 a 3.5.3.2.



Obr. 3.5.3.1 Sinus generovaný digitálním oscilátorem s matematickým modelem



Obr. 3.5.3.2 Spektrum signálu na obr. 3.5.3.1

Na obrázku 3.5.3.2 je vidět, že amplituda šumu je pro oscilátor s matematicky modelovaným sinem je hladší než v předchozích případech. Rovněž je vidět, že vlivem nedokonalosti sinu (kvůli aproximaci), signál obsahuje více harmonických složek, tedy větší celkové harmonické zkreslení (THD). Také hodnota THD+N, které je v tomto případě vysoká, kvůli špatné aproximaci. Pokud ale odečteme hodnotu THD, dostaneme pro šum lepší výsledek než v předchozích případech. Přibližně 0,5 %. Zvukovou ukázku tohoto typu oscilátoru je možné posoudit v audio ukázce (Audio 8) na přiloženém CD.

3.5.4 Porovnání výsledků digitálních oscilátorů

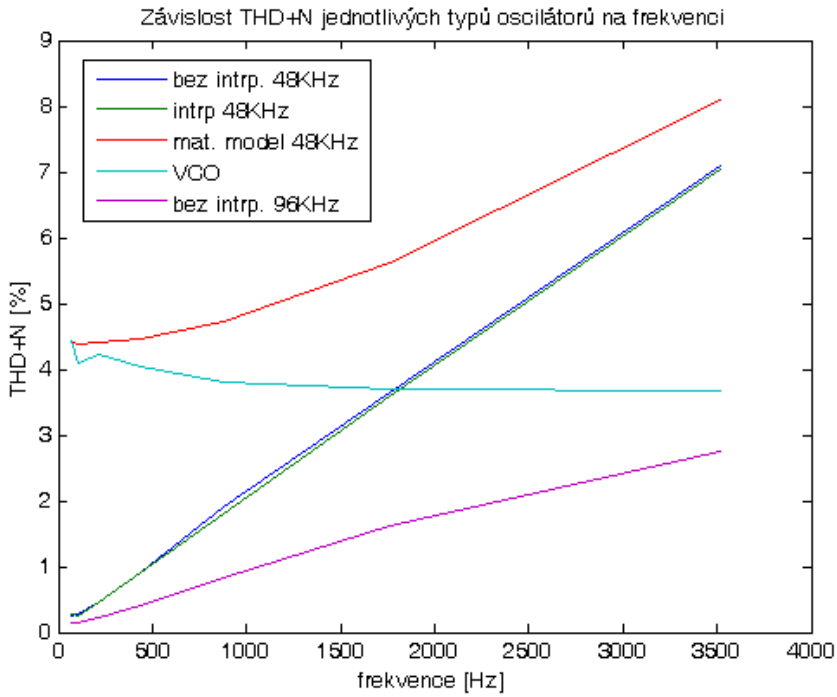
V této kapitole budou porovnány výsledky z měření THD+N všech výše zmíněných metod digitálních oscilátorů. Výsledky budou porovnány s VCO z modulárního hudebního syntezátoru postavené dle plánů dostupných z [21] a parametry v příloze B. K porovnání bude také použit profesionální digitální syntezátor Clavia Nord Modular. K měření byla dále použita zvuková karta M-Audio Delta 1010 s rozlišením 24bit/96kHz. Další parametry k M-Audio Delta 1010 v příloze B. Jako měřící software byl použit software ARTA [20]. Jako ukazatel kvality oscilátoru bylo použito měření celkového harmonického zkreslení plus šum (THD+N). Tento parametr software ARTA přímo dokáže měřit. Následující tabulka (tab. 3.5.4.1) ukazuje naměřené výsledky.

Tab. 3.5.4.1 Porovnání výsledků měření THD+N digitálních oscilátorů

	THD+N [%]						
Frekvence [Hz]	Fázový akumulátor s intrp. *SR = 44.1 kHz	Fázový akumulátor bez intrp. *SR = 44.1 kHz	matemat. model *SR = 44.1 kHz	Variab. sampl. rychlost	VCO	Fázový akumulátor bez intrp. *SR = 96KHz	Nord Modular
65	0,28	0,25	4,4	11,98	4,43	0,15	0,19
110	0,26	0,27	4,39	11,96	4,09	0,15	0,072
220	0,45	0,46	4,41	11,96	4,22	0,22	0,025
440	0,9	0,91	4,47	11,96	4,04	0,42	0,018
880	1,8	1,92	4,73	11,97	3,79	0,82	0,018
1760	3,6	3,64	5,62	12,08	3,69	1,63	0,018
3520	7,04	7,1	8,1		3,66	2,74	0,018

**SR – vzorkovací kmitočet*

Následující graf na obr. 3.5.4.1 ukazuje graficky hodnoty z tab. 3.5.4.1 kromě metody s variabilním vzorkovacím kmitočtem. U této metody, jako jediné, byl použit trojúhelníkový signál, a proto výsledky nejsou porovnatelné. U ostatních oscilátorů byl použit sinusový výstupní signál.

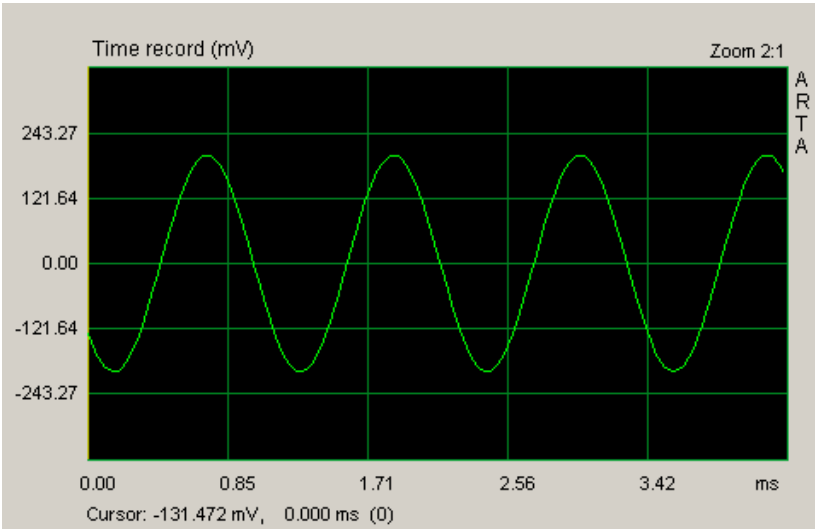


Obr. 3.5.4.1 Grafické zobrazení výsledky dle tab. 3.5.4.1

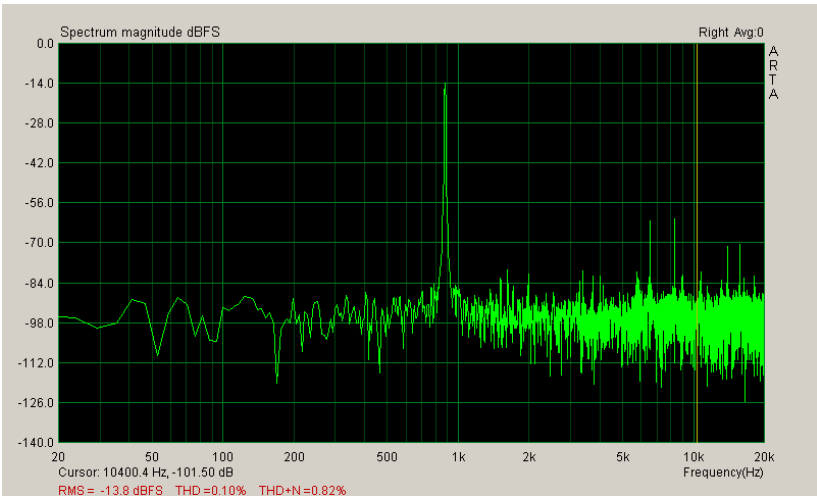
Toto měření odhaluje několik zajímavých zjištění. Jako první si lze všimnout, že mezi oscilátorem s interpolací a bez interpolace, při vzorkovacím kmitočtu 44,1 kHz, není velký rozdíl. Metoda interpolace je výpočetně náročná, jak bylo zmíněno v kapitole 3.5.2. Proto tuto metodu není nutné implementovat a tím lze ušetřit výpočetní výkon. To potvrzuje i typ digitálního oscilátoru s matematickým modelováním, který je popsán v kapitole 5.5.3. U matematického modelu je hodnota THD vyšší kvůli nepřesné aproximaci sinu. Pokud by ale šum závisel výhradně na přesnosti hodnoty vzorku, měla by hodnota THD+N zůstat po celý měřený rozsah konstantní, tak jak je tomu u VCO. Jak je vidět tak u matematického modelu stoupá hodnota šumu strmě. Významného zlepšení spektrální čistoty bylo dosaženo až zvýšením vzorkovací kmitočet na 96 kHz. Maximální hodnota THD+N klesla u oscilátoru s fázovým akumulátorem bez interpolace po zvýšení vzorkovací kmitočet 2x o 4.36 %. U VCO je značné harmonické zkreslení. To je způsobeno nepřesným vlnovým tvarováním analogovými obvody. Hodnota THD+N u VCO však s rostoucí frekvencí neroste, naopak klesá. Naměřené hodnoty z profesionálního digitálního syntezátoru Nord Modular G2X jsou po celý měřený frekvenční rozsah téměř na nulové hodnotě. Vzhledem k tomu, že Nord Modular G2X používá bitový tok 24bit a vzorkovací kmitočet 96 kHz a jistě bude obsahovat kvalitnější součástky a dokonalejší softwarové metody, není možné dosáhnout stejných výsledků. Jako poslední zbývá posoudit digitální oscilátor s variabilním vzorkovacím kmitočtem. Tento oscilátor lze porovnat pouze subjektivně na základě poslechového testu. Dle mého názoru je zvuk z tohoto oscilátoru kvalitnější, než u oscilátorů s fázovým akumulátorem. Problém je ovšem s jeho výpočetní náročností a proto je pro tuto práci nerealizovatelný. Zvyšující se hodnota šumu oscilátorů s fázovým akumulátorem je dána zvyšujícím se přenosovým činitelem α, jak je popsáno v kapitole 2.4. Je to dána hlavně použitím tabulky o velikosti N = 2048, kdy přenosový činitel je nízký pro nižší frekvence a narůstá se zvyšující se frekvencí. Při použití tabulky s menším počtem vzorků, např. N = 512 by byl šum frekvenčně vyrovnanější. Na druhou stranu by bylo

pravděpodobně nutné použít lineární interpolaci. Po zvýšení vzorkovacího kmitočtu na 96 kHz bylo dosaženo dobrých výsledků, a proto byla ponechána velikost tabulky na N = 2048.

Závěrem této kapitoly je třeba zdůraznit, že cílem porovnávání digitálních oscilátorů nebylo detailní zkoumání digitálních a analogových oscilátorů. Cílem bylo vybrat nejvhodnější metodu s přijatelnými zvukovými výsledky pro další práci. Nakonec byl zvolen typ oscilátoru s fázovým akumulátorem bez interpolace s vzorkovacím kmitočtem 96 kHz (obr. 3.5.4.2 a obr. 3.5.4.3). Tento oscilátor je snadno realizovatelný, zvukové výsledky jsou přijatelné a výpočetní výkon na STM32F4 Discovery je dostatečný. Zvukové ukázky je možno posoudit a audio souborů přiložených na CD. Popis audio souborů je k nalezení v příloze D.



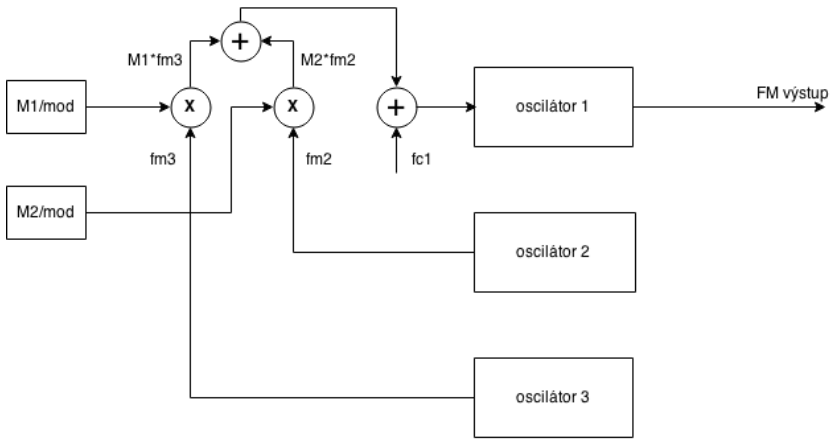
Obr. 3.5.4.2 Signál z „wavetable“ oscilátoru s fázovým akumulátorem bez interpolace s vzorkovacím kmitočtem 96 kHz



Obr. 3.5.4.3 Spektrum signálu z obrázku 3.5.4.2

3.6 Frekvenční modulace

Dle návrhu na obrázku 3.1 je patrné zapojení vstupů frekvenční modulace. Oscilátor 1 může být modulován současně, nebo samostatně oscilátory 2 a 3. Následující obrázek 3.6.1 ukazuje podrobněji směrování signálů frekvenční modulace.

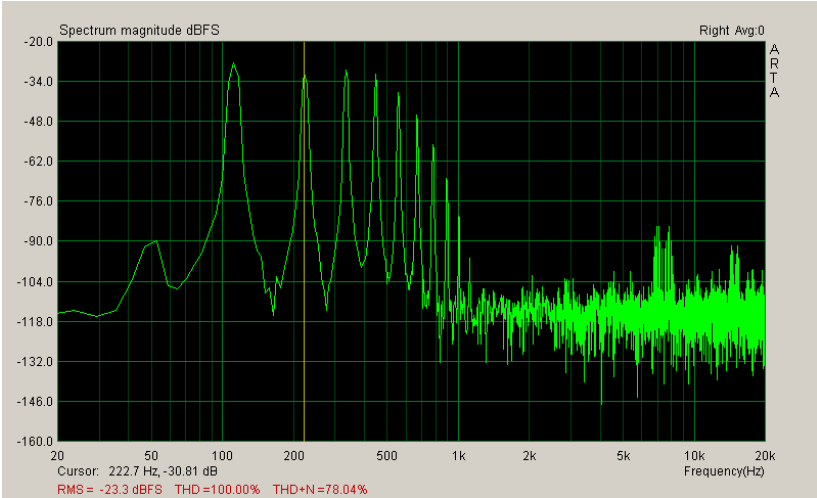


Obr. 3.6.1 Zapojení FM vstupu oscilátoru 1

Dle obrázku 3.6.1 jsou výstupy oscilátorů 2 a 3 zapojené do násobiček. Násobičky umožňují nastavení modulačního indexu potenciometry FM OSC2 a FM OSC3 dle obrázku 3.1. Pokud je nastaven modulační index pomocí potenciometru na pevně danou hodnotu, je výstupem oscilátorů příslušné spektrum, které je statické. Proto u násobiček je možnost ovládat modulační index externím signálem. Také dle obrázku 3.1 je možné modulovat přímo frekvenci všechno oscilátorů ovladačem PMOD. Tento typ modulace může být například signál z ADSR obálky, nebo nízko-frekvenčního oscilátoru (LFO) atp. ADSR ani LFO zatím nejsou softwarově implementovány a jde pouze o přípravu pro pozdější využití. V budoucnu je implementace LFO i ADSR nutná, protože bez těchto kontrolních signálů lze generovat pouze statická spektra. Spektrum každého zvuku se mění dynamicky v čase. Audio signály z oscilátorů 1 a 2 vynásobené příslušným modulačním indexem se sčítají navzájem a také se základní frekvencí fc oscilátoru. Pokud všechny tři oscilátory generují harmonický signál, bude signál z oscilátoru 1 dán následujícím vztahem.

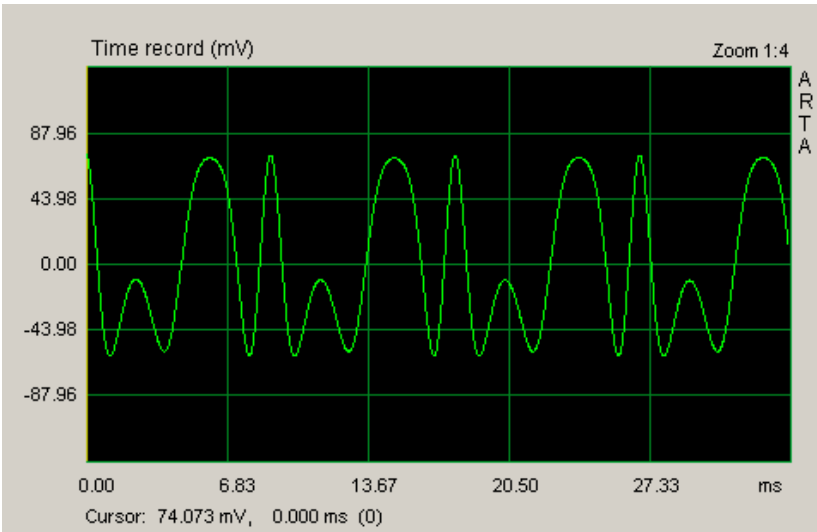
$$f_t = A\sin(\omega_1 t + \varphi_1 + M1\sin(\omega_2 t + \varphi_2) + M2\sin(\omega_3 t + \varphi_3)) \tag{17}$$

Pro jednoduchost a ověření správné funkce frekvenční modulace byl v následujícím testu zapnutý pouze oscilátor 2 dle obrázku 3.6.1. Oscilátor 3 byl ponechán vypnutý, respektive modulační index byl nastaven na nulu. Nejprve bylo testováno chování FM při identických frekvencích oscilátorů 1 a 2. Byla zvyšována hodnota pouze modulačního indexu. Při nulové hodnotě modulačního indexu k frekvenční modulaci nedochází, zatímco při zvyšování modulačního indexu roste počet harmonických složek. Následující obrázek 3.6.2 ukazuje spektrum signálu oscilátoru 1 s fundamentální frekvencí 110 Hz, který je modulován oscilátorem 2 se stejnou frekvencí. Modulační index byl nastaven na hodnotu přibližně 0,15.



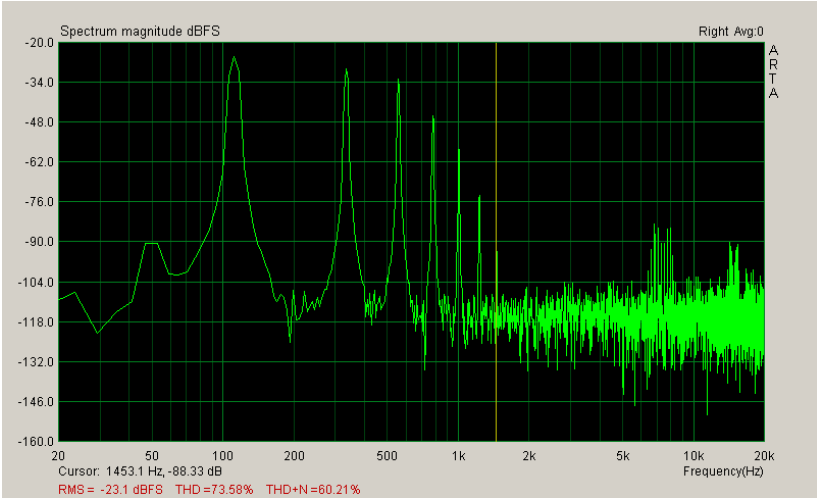
Obr. 3.6.2 FM spektrum signálu, oscilátor 1 – 110 Hz, oscilátor 2 – 110 Hz, modulační index 0,15

Při výše zmíněném nastavení byly generovány frekvence 110 Hz (základní), 220 Hz (1. harmonická), 329,6 Hz, 440 Hz, 554,3 Hz, 659,2 Hz, 783,9 Hz, 880 Hz a 1046 Hz. Stejný signál v časové oblasti ukazuje obrázek 3.6.3.



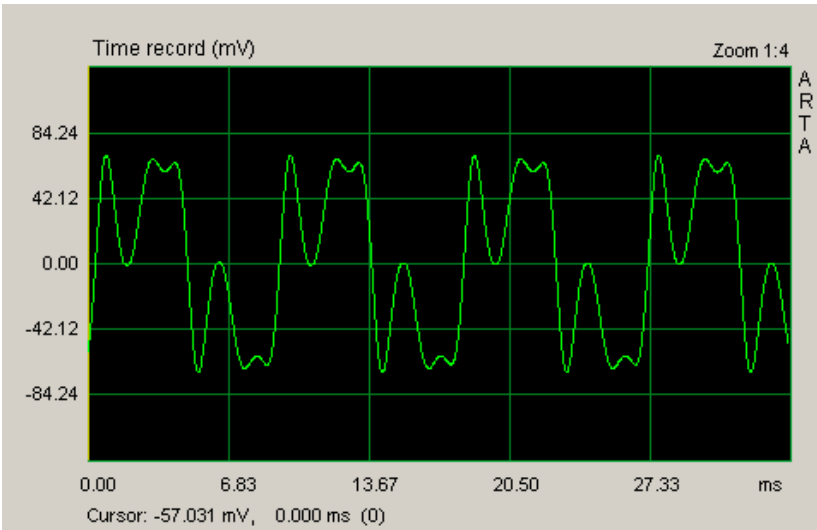
Obr. 3.6.3 Spektrum signálu na obr. 4.6.2 obrázku v časové oblastii

V druhém testu byla změněna frekvence oscilátoru 2 (modulátoru) o jednu oktávu výše na 220 Hz. Základní frekvence oscilátoru 1 byla ponechána 110 Hz. Oscilátor 3 byl vypnutý. Modulační index zůstal stejný jako v předchozím případě. Přibližně 0.15. Spektrum tohoto nastavení ukazuje obrázek 3.6.4.



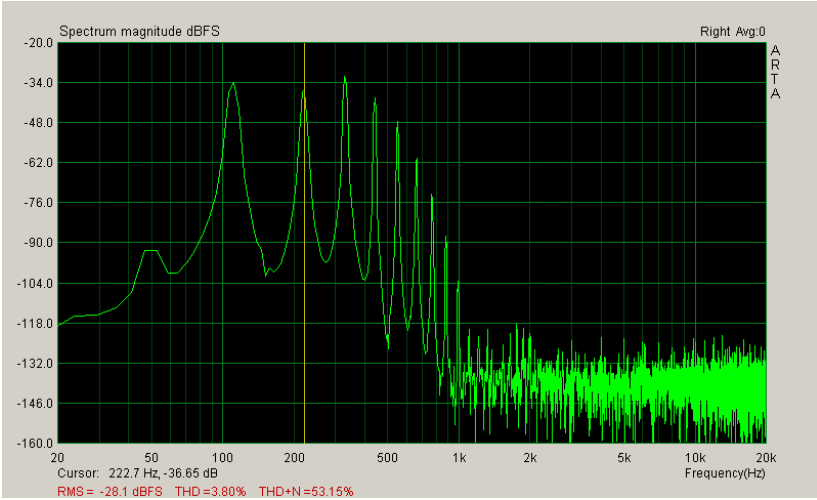
Obr. 3.6.4 Spektrum FM signálu, oscilátor 1 - 110 Hz, oscilátor 2 – 220 Hz, modulační index - 0,15

Na obrázku 4.6.4 je zřejmá změna po zvýšení frekvence modulátoru 2x. Je generováno méně harmonický složek s jiným pořadím. Základní harmonická zůstala stejná 110 Hz, dále 329,6 Hz, 554,3 Hz, 783,9 Hz, 1046 Hz, 1244 Hz, 1480 Hz. Stejný signál v časové oblasti ukazuje obrázek 4.6.5.



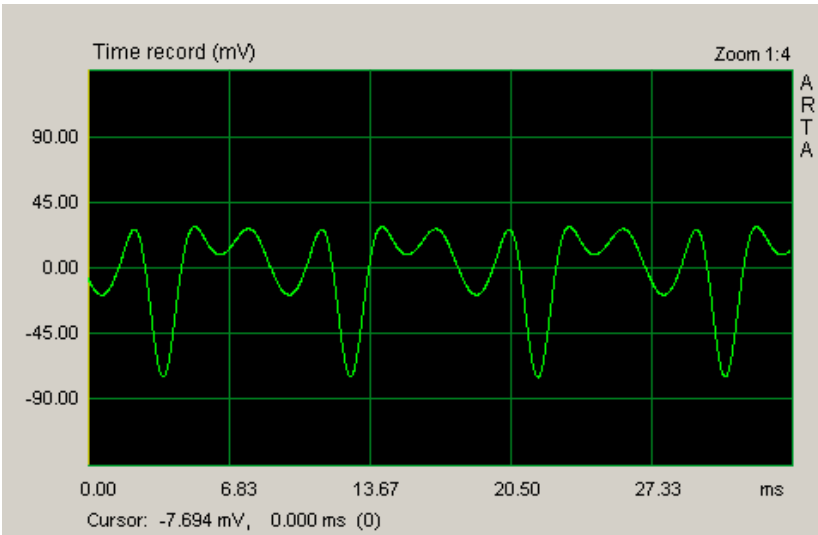
Obr. 4.6.5 Signál v časové oblasti spektra na obr. 3.6.4

Porovnání a správná funkčnost frekvenční modulace byla ověřena na profesionálním hudebním syntezátoru Clavia Nord Modular G2. Je to digitální hudební nástroj s asociovaným podpůrným softwarem, kde lze libovolně kombinovat oscilátory, filtry apod. Byla provedena tedy identická nastavení. Oscilátor 1 se základní frekvencí 110 Hz, oscilátor 2 (modulátor) rovněž 110 Hz, modulační index přibližně stejný, 0,15. Spektrum z Nord Modularu ukazuje obrázek 3.6.6. Strukturu zapojení (patch) v Nord Modularu ukazuje obrázek 3.6.8.



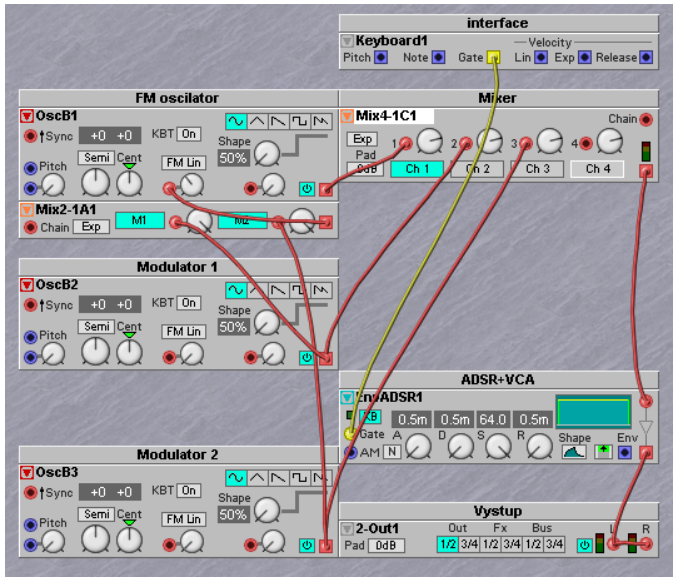
Obr. 3.6.6 FM spektrum Nord Modularu G2, oscilátor 1 – 110 Hz, oscilátor 2 – 110 Hz, modulační index 0,15

Porovnáním obrázků 3.6.2 a 3.6.6 je vidět, že spektra mají identické harmonické složky pouze s tím rozdílem, že Nord Modular má významně nižší šum. To je dáno důvody zmíněnými v kapitole 3.5.4. Tímto porovnáním se podařilo ověřit, že frekvenční modulace pracuje dle návrhu správně. Na obrázku 3.6.7 je pro úplnost uveden signál z obrázku 3.6.6 v časové oblasti.



Obr. 3.6.7 Signál z obrázku 4.6.6 v časové oblasti

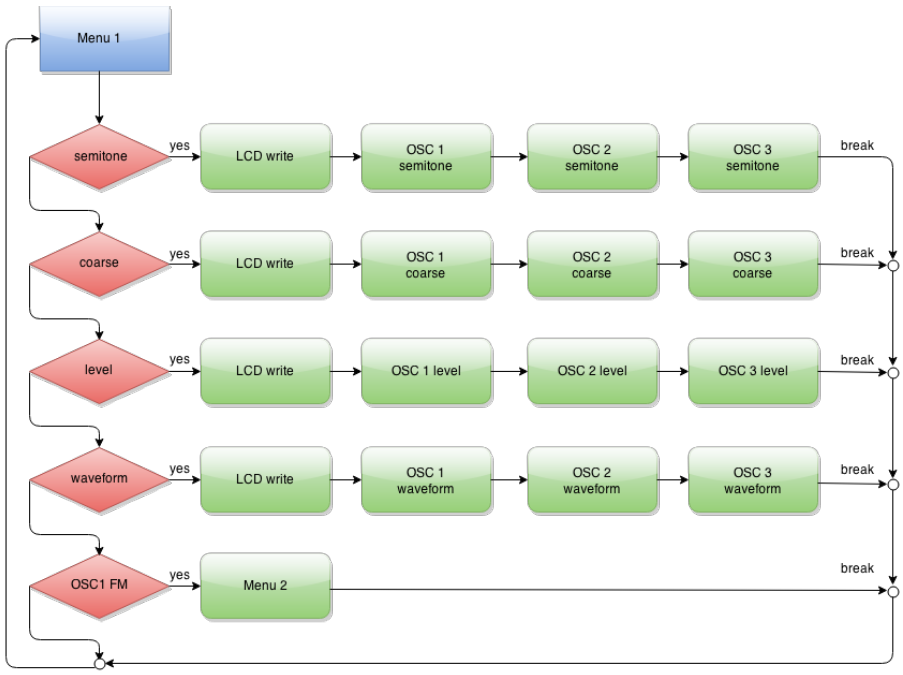
Frekvenční modulace je velmi perspektivní metoda tvorby zvukových spekter. Existuje mnoho prostoru, kde lze s frekvenční modulací experimentovat. Například zavést různá zapojení modulátorů, nebo zavést zpětnou vazbu, případně komplexní modulátory. Zvukové ukázky frekvenční modulace jsou uvedeny příloze D (Audio 2-7). Jednotlivé audio soubory jsou v příloze na CD.



Obr. 3.6.8 Nord Modular patch

3.7 Ovládání a nastavování parametrů syntezátoru

Aby bylo možné vytvářet různá zvuková spektra, je třeba měnit parametry syntezátoru v reálném čase. Tyto změny lze provádět pomocí čtyř připojených potenciometrů dle obr. 3.2. Hodnoty z potenciometrů jsou použity buďto pro pohyb v menu, nebo pro nastavení příslušného parametru. Ovládání je rozděleno do dvou menu. V menu 1, potenciometr 1 (první zleva na obr. 3.2) ovládá pohyb v menu pomocí softwarového přepínače. Hodnoty potenciometrů 2,3 a 4 jsou v menu 1 po příslušné matematické úpravě použity přímo pro nastavení konkrétního parametru. Následující obr. 3.7.1 ukazuje blokové schéma menu 1.



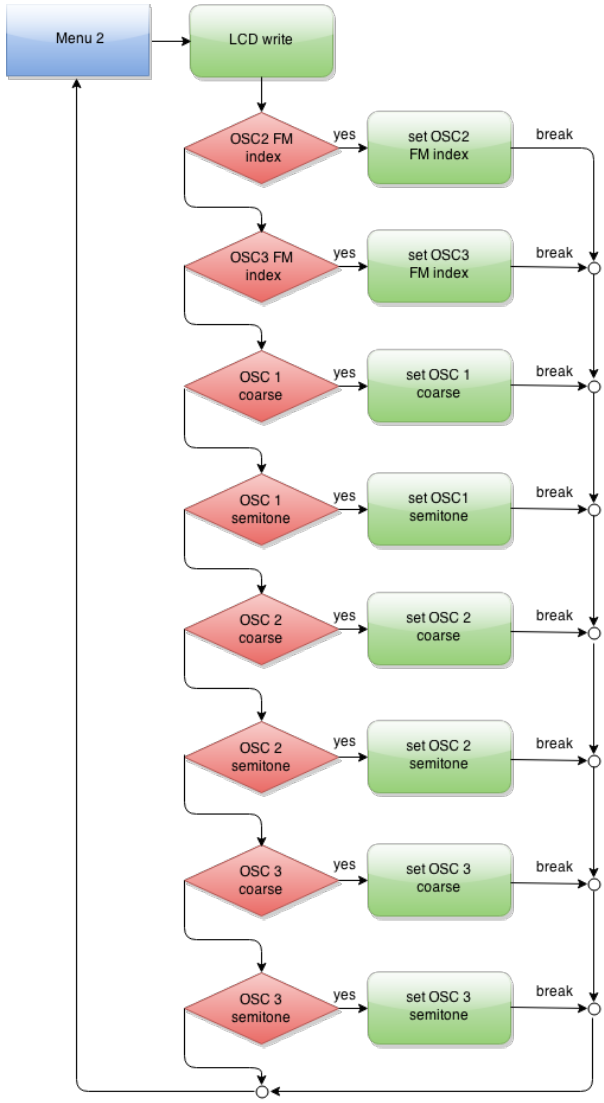
Obr. 3.7.1 Blokový diagram hlavní části ovládání (menu 1)

Menu 1 má pět položek. Po vstupu do konkrétní položky vždy dochází k zápisu na displej, kde se program nachází. Na displej je také zapisována hodnota konkrétního parametru. Položka 5 mají vlastní menu 2, které bude vysvětleno níže. Následující tab. 3.7.1 vysvětluje význam jednotlivých parametrů položek 1-4 včetně jejich rozsahů.

Tab. 3.7.1 Přehled nastavovaných parametrů a jejich rozsahů

Potenciometr				**Poznámka**
1	**2**	**3**	**4**	
semitone	OSC 1 semitone	OSC 2 semitone	OSC 3 semitone	Nastavuje ladění oscilátoru v rozsahu 0 až +12 půltónů
coarse	OSC 1 coarse	OSC 2 coarse	OSC 3 coarse	Nastavuje ladění oscilátorů v rozsahu -2 a +2 oktávy
level	OSC 1 level	OSC 2 level	OSC 3 level	Nastavuje výstupní hlasitost oscilátorů v rozsahu 0-10 kroků
waveform	OSC 1 waveform	OSC 2 waveform	OSC 3 waveform	Volba výstupní vlnové formy, sinus, nebo trojúhelník

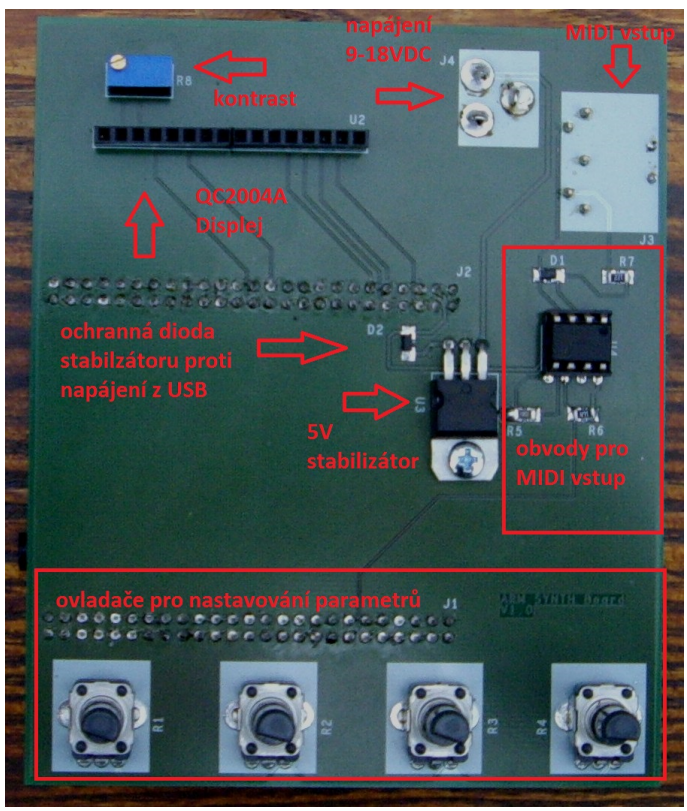
Z položky 5 menu 1 (OSC1 FM) lze vstoupit do menu 2, obr. 3.7.2. V menu 2 lze nastavit modulační index. Dále zde lze měnit stejné parametry jako v menu 1, tj. změny frekvence o půltóny či oktávy. Spojení nastavení modulačního indexu a nastavení frekvence je právě z toho důvodu, že spektrum signálu závisí u frekvenční modulace právě na těchto dvou parametrech. Proto je výhodné mít tyto parametry přístupné v jedné nabídce.



Obr. 3.7.2 Blokové schéma ovládání menu 2

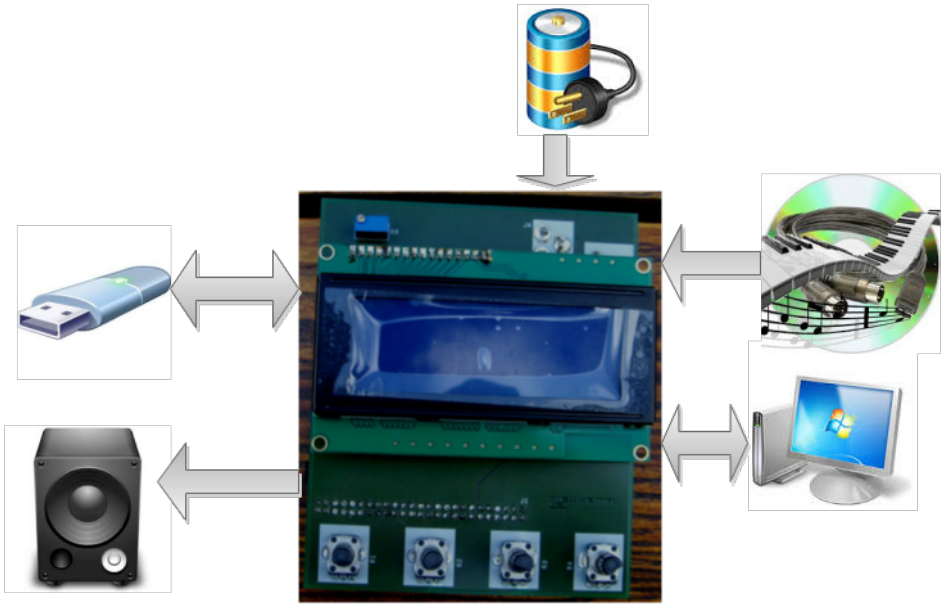
3.8 Návrh plošného spoje

Návrh plošného spoje odpovídá zapojení na obrázku 3.2 graficky a schématu zapojení v příloze A (hlavní schéma). Na obrázku 3.2 není zakreslen napájecí konektor pro připojení adaptéru. Plošný spoj byl koncipován tak, aby byl odnímatelný displej a odnímatelný STM32F4 Discovery kit. Dále byl požadavek mít dostatek místa mezi jednotlivými potenciometry, tak aby s nimi bylo možné pohodlně otáčet. Plošný spoj má rozměry 117,4x97,8mm. Vstupy a výstupy byly navrženy tak, aby byly rovnoměrně rozloženy podél obdélníkového tvaru, avšak ne zpředu. Konektor pro napájení z adaptéru se nachází samotný v zadní části. Je to hlavně z důvodu, aby při zapojování ostatních zařízení nedošlo k nechtěnému odpojení napájení. Z levé strany plošného spoje se nachází 3.5mm jack zvukový výstup STM32F4 Discovery, která je nasunuta ze spodní části na dva 50-ti pinové konektory. Na téže straně se rovněž nachází mikro USB konektor, který je také součástí STM32F4 Discovery. Tento konektor slouží například k připojení paměťového média. Momentálně není využíván. Na pravé straně plošného spoje se nachází mini USB konektor pro připojení PC a DIN konektor pro MIDI vstup. Z vrchu plošného spoje je 16-pin konektor pro připojení alfanumerického displeje QC2004A a trimr pro nastavení kontrastu displeje. Kromě již zmíněných čtyř potenciometrů se shora DPS nachází 5V stabilizátor L7805CV a také obvody pro MIDI vstup dle schématu v příloze A. Napájení je možné dvěma způsoby. Buďto externím adaptérem s rozsahem napětí přibližně 9-18VDC, nebo z PC pomocí USB. Schottkyho dioda na výstupu stabilizátoru zabraňuje toku proudu do stabilizátoru, když je voleno napájení z USB. V případě napájení z adaptéru je na této diodě úbytek napětí přibližně 0,3V což je přijatelné. Právě z důvodu nízkého úbytku napětí byla zvolena Shottkyho dioda.

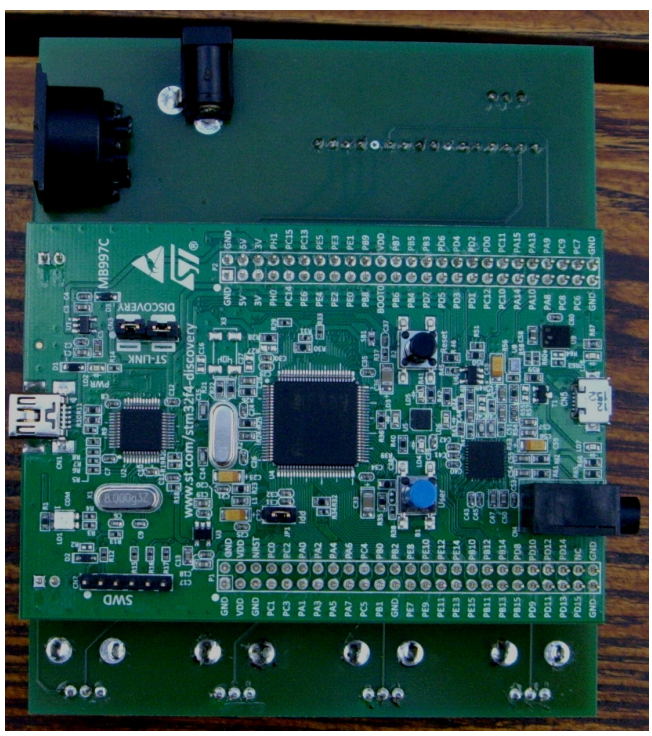


Obr. 3.8.1 Navržený plošný spoj

Plošný spoj je vyroben z materiálu FR4 o tloušťce 1,5mm s dvěma vrstvami 18μm Cu. Povrchová úprava bezolovnatý HAL. Návrh byl proveden programovým balíkem Cadence. Seznam použitých součástek je uveden v příloze C. Následující obrázky ukazují výsledek práce z pohledu shora i zdola.



Obr. 3.8.2 Externí mikroprocesorová jednotka pro syntézu zvukových signálu z pohledu shora



Obr. 3.8.3 Externí mikroprocesorová jednotka pro syntézu zvukových signálů z pohledu zdola

4 Závěr

Hlavním cílem této práce bylo vybrat mikrokontroler s dostatečným výkonem a navrhnout externí jednotku pro syntézu zvukových signálů v rozsahu osmi oktáv, která využívá frekvenční modulace jako metodu zvukové syntézy. Vzorkovací kmitočet 48 kHz a dynamický rozsah 16 bitů. Takto definovaný koncept měl tvořit základ, který bude možno v další práci rozvíjet. Všechny kladené cíle se podařilo splnit. Podařilo se vybrat levný a dostatečně výkonný mikrokontroler ARM Cortex M4 společně s digitálně analogovým převodníkem na jedné vývojové desce STM32F4 Discovery. Zvolený mikrokontroler poskytuje dostatečnou výkonovou rezervu i pro další rozvoj práce.

K mikrokontroleru byl připojen alfanumerický displej s velikostí 20 znaků, 4 řádky a byla realizována jeho softwarová obsluha. Pro nastavování parametrů byly připojeny čtyři potenciometry a byla nastavena periferie ADC převodníku včetně využití přenosu hodnoty na určené místo v paměti bez nutnosti využití procesoru (DMA). Dále byly navrženy obvody pro MIDI vstup a byla nastavena periferie USART pro příjem MIDI zpráv. Podařilo se také využít kvalitní DA převodník CS43L22 na vývojové desce STM32F4 Discovery a posílat zvuková data přes I2S rozhranní v požadovaném rozlišení 48 kHz a dynamickém rozsahu 16 bitů. Pro hardware byl navržen plošný spoj, který je možné propojit s PC, MIDI klaviaturou a zvukovým výstupem. Zařízení je také možno připojit na externí napájecí zdroj při absenci napájení přes USB.

Značná část práce byla věnována porovnávání způsobů realizace digitálních oscilátorů. Bylo vyzkoušeno několik metod a cílem bylo dosáhnout přijatelného odstupu signál šum. Metoda oscilátoru s variabilním vzorkovacím kmitočtem je metodou s dobrými zvukovými vlastnostmi, ale má nevýhodu v tom, že se obtížně softwarově realizuje. Metoda s matematickým modelováním sinu má rovněž značné výpočetní nároky a navíc naměřené výsledky nebyly dostatečné. Přijatelného odstupu signál šum se podařilo dosáhnout až při použití „wavetable“ oscilátoru s fázovým akumulátorem přičemž byla zvýšena vzorkovací kmitočet na 96 kHz. Zvyšující se šum s rostoucí frekvencí generovaného signálu je dán použitím vlnové tabulky s velikostí 2048 vzorků a přenosovým činitelem α, který se zvyšuje s rostoucí frekvencí. Při použití vlnové tabulky s například velikostí 512 vzorků by byl šum na vyšších frekvencích nižší, ale na druhou stranu by byl šum vyšší na nižších frekvencích a bylo by nutné použít lineární interpolaci. Po zvýšení vzorkovacího kmitočtu „wavetable“ oscilátoru s fázovým akumulátorem a vlnovou tabulkou s velikostí 2048 vzorků, se podařilo dosáhnout dobrých výsledků za cenu nízkých výpočetních nákladů. Touto metodou se podařilo realizovat tři digitální oscilátory s volitelnými výstupními vlnami sinus a trojúhelník. Mimo přepínání výstupních vlnových forem se podařilo implementovat ovladače nastavení přepínání oktáv, půltónů a nastavení hlasitosti u všech oscilátorů.

Dále se podařilo realizovat metodu frekvenční modulace, která přináší perspektivní výsledky pro syntézu zvuku. Spektra frekvenčně modulovaného signálu byla porovnána s komerčně vyráběným digitálním syntezátorem a bylo ověřeno, že metoda frekvenční modulace pracuje správně. Původní návrh počítal s možností frekvenční modulace u všech tří oscilátorů. Nicméně během zkoušek bylo zjištěno, že pokud jeden oscilátor je frekvenčně modulován a zároveň moduluje kmitočet modulátoru, stane se tato smyčka zpětné vazby nestabilní. To se u digitálního signálu projeví šumem na výstupů místo periodického průběhu. Proto se návrh omezil pouze na možnost frekvenčně modulovat jeden oscilátor zbylými dvěma oscilátory. V ovládání byl implementován ovladač pro nastavení modulačního indexu každého modulátoru.

Frekvenční modulace nabízí další možnosti rozvoje projektu. Například by bylo možné pracovat na zavedení stabilní zpětné vazby. Další vývoj by se mohl ubírat k realizaci modulátorů pro pomalé změny frekvence oscilátorů. Například ADSR obálka, nízko frekvenční oscilátor nebo sekvencer. Výše definovaná struktura je schopna generovat pouze statická spektra, která nejsou z hlediska zvukové syntézy tak zajímavá. Pomocí modulátoru je možné modulovat modulační index nebo frekvenci oscilátorů a tím, dynamicky „rozhýbat“ stávající statické spektrum. Dynamickými spektry lze napodobovat spektra klasických hudebních nástrojů. Napodobování klasických hudebních nástrojů je hlavní myšlenka, pro kterou byl syntezátor původně navržen. Podstatou této práce bylo však navrhnout základ, který bude možno dále rozvíjet a to se podařilo.

Reference

[1] M. Vail, Vintage Synthesizers. Miller Freeman Books, 2000, iSBN 0-87930-303-3.

[2] M. Instruments. Shruthi. Accessed: 2014-12-21. [Online]. Available: http://mutable-instruments.net/shruthi1

[3] PreenFM2. Accessed: 2015-1-19. [Online]. Available: http://www1.coocox.org/CoIDE/Compiler_Settings.html

[4] B. Frei. (2002) Digital sound generation. Accessed: 2015-3-16. [Online]. Available: https://www.zhdk.ch/index.php?id=71591

[5] P. Symons, Digital Waveform Generation. Cambridge University Press, 2013, iSBN 978-1-107-02097-9.

[6] Gearslutz. Taxonomy early digital synthesizers. Accessed: 2014-9-23. [Online]. Available: https://www.gearslutz.com/board/electronic-music-instruments-electronic-music-production/826280-taxonomy-early-digital-synthesizers.html

[7] C. M. S. of computer science. Interpolation error in waveform table lookup. Accessed: 2014-1-10. [Online]. Available: https://www.cs.cmu.edu/~rbd/papers/tlu98/tlu98.pdf

[8] E. R. Miranda, Computer sound design: synthesis techniques and programming. Taylor & Francis, 2002, vol. 1, iSBN 0-240-51693-1.

[9] E. R. Miranda and A. Biles, Evolutionary computer music. Springer, 2007, iSBN 1-84628-599-2.

[10] TKJelectronics. Accessed: 2014-9-26. [Online]. Available: http://blog.tkjelectronics.dk

[11] STMicroelectronics. Reference manual pro stm32f405xx/07xx, stm32f415xx/17xx, stm32f42xxx a stm32f43xxx. Accessed: 2014-9-26. [Online]. Available: http://www.st.com/web/en/resource/technical/document/.../DM00031020.pdf

[12] Coocox. Coocox coide. Accessed: 2014-9-26. [Online]. Available: http://www.coocox.org/

[13] STMicroelectronics. Accessed: 2014-9-26. [Online]. Available: http://www.st.com

[14] Elecfreaks. Accessed: 2014-9-26. [Online]. Available: http://www.elecfreaks.com

[15] Santy. Accessed: 2014-9-26. [Online]. Available: http://www.santy.cz

[16] Stm32f4-discovery. Accessed: 2014-9-26. [Online]. Available: http://www.stm32f4-discovery.com

[17] ARM. Cmsis – cortex microcontroller software interface standard. Accessed: 2014-9-26. [Online]. Available: http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php

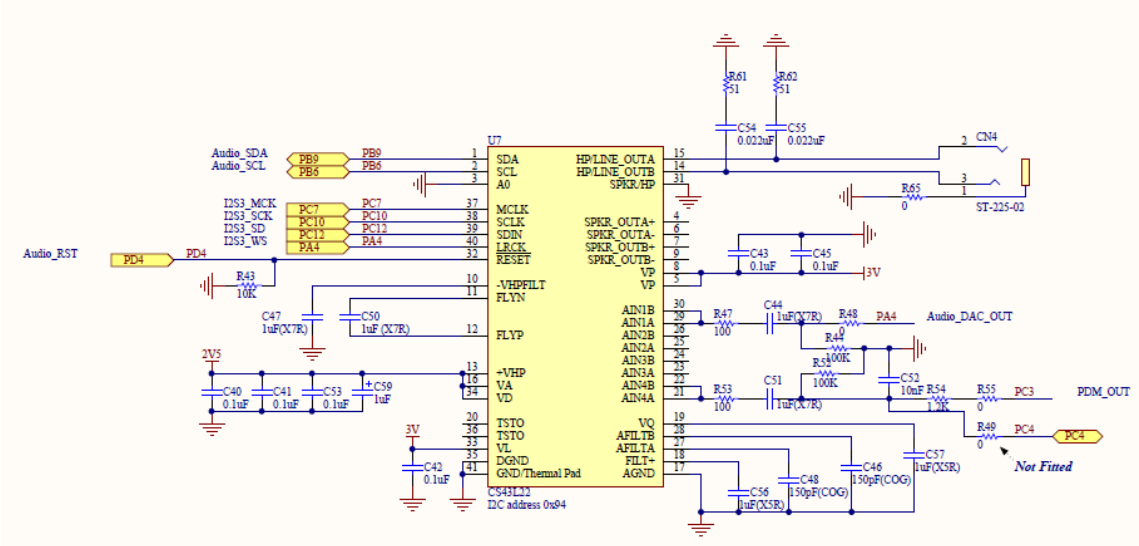
[18] C. Logic. Accessed: 2014-10-10. [Online]. Available: http://www.cirrus.com/en/products/cs43l22.html

[19] Mind-Dump. Configuring the stm32f4-discovery for audio. [Online]. Available: http://www.mind-dump.net/configuring-the-stm32f4-discovery-for-audio

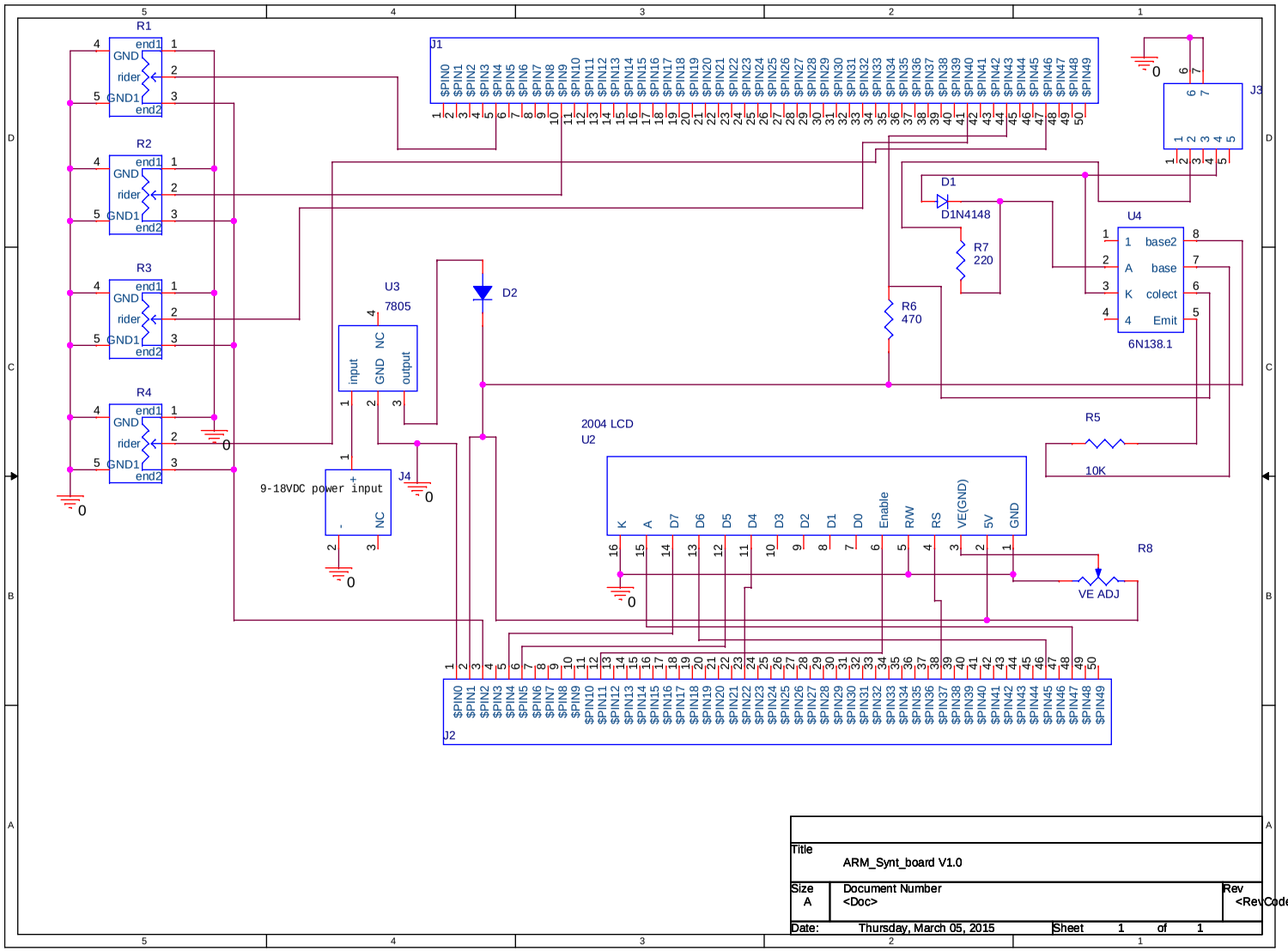
[20] Arta. Audio measurement and analysis software. Accessed: 2014-12-16. [Online]. Available: http://www.artalabs.hr

[21] Yusynth. Vco. Accessed: 2014-12-15. [Online]. Available: http://yusynth.net/Modular/index_en.html

A Schéma zapojení



Zapojení DAC CS43L22 na vývojové desce STM32F4 Discovery


R1
R2
R3
R4
end1
GND
rider
GND1
end2
J1
$PIN0 $PIN1 $PIN2 $PIN3 $PIN4 $PIN5 $PIN6 $PIN7 $PIN8 $PIN9 $PIN10 $PIN11 $PIN12 $PIN13 $PIN14 $PIN15 $PIN16 $PIN17 $PIN18 $PIN19 $PIN20 $PIN21 $PIN22 $PIN23 $PIN24 $PIN25 $PIN26 $PIN27 $PIN28 $PIN29 $PIN30 $PIN31 $PIN32 $PIN33 $PIN34 $PIN35 $PIN36 $PIN37 $PIN38 $PIN39 $PIN40 $PIN41 $PIN42 $PIN43 $PIN44 $PIN45 $PIN46 $PIN47 $PIN48 $PIN49
J3
D1
D1N4148
R7
220
R6
470
U4
base2
base
colect
Emit
6N138.1
U3
7805
input
GND
output
NC
D2
9-18VDC power input
J4
2004 LCD
U2
K A D7 D6 D5 D4 D3 D2 D1 D0 Enable R/W RS VE(GND) 5V GND
R5
10K
R8
VE ADJ
J2
0
Title
ARM_Synt_board V1.0
Size
A
Document Number
<Doc>
Rev
<RevCode>
Date:
Thursday, March 05, 2015
Sheet
1
of
1

B Technické parametry použitých zařízení

Technické parametry VCO:

- Charakteristika 1V/oktávu
- Výstupní signály sinus, pila, čtverec, trojúhelník
- Frekvenční rozsah 16Hz – 20kHz
- Napájení +/-15VDC
- Amplituda výstupních signálů 4Vpp
- Ovladače: frequency, fine tune, lin./exp. FM level, PWM, pulse width
- Kontrolní vstupy: 2x 1V/oct., PWM, lin./exp FM

Technické parametry M-Audio Delta 1010:

- Frekvenční rozsah: 20Hz-22kHz, +/-0.3dB
- Dynamický rozsah (A-weighted): 109dB (A/D), 117dB (D/A)
- SNR (A-weighted): -109 dB (A/D), -117dB (D/A)
- THD +N: 0.00072% (A/D), 0.00200% (D/A)

C Seznam použitých součástek

Typ	Označení	Hodnota
RK09D1130C1B – potenciometr (ALPS)	R1, R2, R3, R4	10K lin
rezistor SMD 1206	R5	10K, 1%
rezistor SMD 1206	R6	470R, 1%
rezistor SMD 1206	R7	220R, 1%
trimr 10-otáček	R8	2K
MBR0520LT1G – Dioda Schottky	D1, D2	—
6N138 – optočlen DIP8	U4	—
L7805CV – 5V stabilizátor	U3	1A
QC2004 – alfanumerický displej 20x4	U2	—
BL834DG – dutinková lišta	J1, J2	á 50 pin
DIN5ZPS180 – DIN zásuvka do DPS	J3	—
PC-GK2.1 napájecí konektor	J4	—
STM32F4 Discovery kit	—	—

D Přehled audio souborů na CD

- **Audio 1** - příklad generování sinusového periodického hudebního signálu metodou „wavetable“ oscilátoru s fázovým akumulátorem bez interpolace se vzorkovacím kmitočtem 96 kHz. Dochází ke skokové změně frekvence zvyšující se od 110 Hz (A2) až po 3520 Hz (A7). V ukázkách Audio 2 - 7 je použit tento typ oscilátoru.

- **Audio 2** - příklad spektra frekvenčně modulovaného audio signálu. Modulovaný oscilátor (oscilátor 1) i modulátor (oscilátor2) byly nastaveny na shodné frekvence 220 Hz (sinus). V nahrávce je slyšet postupné zvyšování modulačního indexu z nuly na hodnotu 1,5 a zpět. Při zvyšování je patrný nárůst harmonických složek a při snižování jejich úbytek.

- **Audio 3** - v této nahrávce je nastaven modulovaný oscilátor na 220 Hz (sinus) a u modulátoru dochází ke změně frekvence z 55 Hz do 880 Hz (sinus). V důsledku změny poměrů obou oscilátoru jsou patrné změny ve spektru signálu. Modulační index je nastaven na pevnou hodnotu.

- **Audio 4** - tento příklad je podobný jako v nahrávce Audio 3 s tím rozdílem, že modulátor je nastaven na pevnou frekvenci 220 Hz (sinus) a u modulovaného oscilátoru se mění frekvece od 55 Hz do 880 Hz (sinus). Modulační index je nastave na pevnou hodnotu.

- **Audio 5** - v této ukázce je modulovaný oscilátor nastaven na 220 Hz (sinus), modulátor také na 220 Hz (sinus) a dochází nejprve ke změně frekvence modulovaného oscilátor o +11 půltónů, potom je změně frekvence modulátoru o +11 půltonů, následně je frekvence obou oscilátorů postupně snížena zpět na 220 Hz.

- **Audio 6** - příklad MIDI sekvence frekvenčně modulovaného signálu. Modulovaný oscilátor i modulátor jsou nastaveny na sinus a dochází k nahodilým změnám o půltóny.

- **Audio 7** - v tomto příkladu je nastavena frekvence modulovaného oscilátoru na 220 Hz (sinus) a signál je modulován dvěmi oscilátory 2 a 3, které jsou nastaveny také na sinus. Nejprve dochází ke zvyšování modulačního indexu oscilátoru 2 a následně ke zvyšování modulačního indexu oscilátoru 3.

- **Audio 8** - sinusový signál z oscilátoru s matematickým modelem popsaného v kapitole 3.5.3. Dochází ke skokové změně frekvence od 110 Hz (A2) až po 3520 Hz (A7).

- **Audio 9** - trojúhelníkový periodický signál z „wavetable“ oscilátoru s variabilním samplovacím kmitočtem popsaného v kapitole 3.5.1. Dochází ke skokové změně frekvence od 110 Hz (A2) až po 1760 Hz (A6).