

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

EXPERIMENTÁLNÍHO HARDWAROVÝ HUDEBNÍ NÁSTROJ, PŘEVÁDĚJÍCÍ ÚDAJE ZE SENZORŮ NA ZVOLENÉ HUDEBNÍ PARAMETRY

EXPERIMENTAL HARDWARE MUSICAL INSTRUMENT, CONVERTING DATA FROM SENSORS TO
SELECTED MUSICAL PARAMETERS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Stanislav Šindler

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. MgA. Mgr. Dan Dlouhý, Ph.D.

BRNO 2021

Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Stanislav Šindler

ID: 195766

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Experimentálního hardwarový hudební nástroj, převádějící údaje ze senzorů na zvolené hudební parametry

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření funkčního prototypu experimentálního hardwarového hudebního nástroje - syntetizéru, který je řízen údaji ze senzorů snímajících teplotu vzduchu, intenzitu světla, vlhkost vzduchu apod. Nástroj bude využívat mikrokontrolerovou platformu Teensy a rozšiřující patici Teensy Audio Shield. Syntetizér bude naprogramován v jazyce C++, program pro komunikaci s počítačem bude využívat programovacího prostředí Max/MSP. Nastavování parametrů syntetizéru bude probíhat v rámci grafického uživatelského rozhraní se standardními prvky (otočnými ovladači, tlačítky, displejem apod.).

DOPORUČENÁ LITERATURA:

[1] GUŠTAR, M. Elektrofony II. Uvnitř, Praha, 2008. ISBN 978-80-239-8447-7.

[2] FORRÓ D. Svět MIDI. Grada, Praha, 1997. 375s. ISBN 80-7169-412-6.

Termín zadání: 1.2.2021

Termín odevzdání: 31.5.2021

Vedoucí práce: doc. Ing. MgA. Mgr. Dan Dlouhý, Ph.D.

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

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

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Experimentální hardwarový hudební nástroj nebo zvukový zdroj“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

podpis autora

Obsah

Úvod	9
1 Teoretická část studentské práce	11
1.1 Syntéza umělého zvuku	11
1.1.1 Součtová (additivní) syntéza	11
1.1.2 Rozdílová (Subtraktivní) syntéza	12
1.1.3 Modulační syntéza frekvenční <i>FM</i> (Frequency modulation) syntéza	12
1.1.4 Modulační syntéza amplitudová <i>AM</i> (Amplitude modulation)	13
1.1.5 Modulační syntéza kruhová <i>RM</i> (Ring Modulation) syntéza	14
1.2 Efektové procesory	15
1.2.1 Efektové procesory měnící kmitočtové spektrum	15
1.3 Efektové procesory uplatňující zpožďovací linku	17
1.4 Platforma Teensy	18
1.4.1 O vývojové desce Teensy 3.2	18
1.4.2 Programování mikrokontroléru Teensy	19
1.4.3 Rozšiřující modul Teensy Audio Shield	20
1.4.4 Návrhový nástroj Audio Design Tool	20
1.4.5 Princip programování v AUT	21
1.4.6 Sběrnice I ² C	22
1.5 Použitý hardware	22
1.5.1 Rotační enkodér	22
1.5.2 Displej	22
1.5.3 Modulární konektor 8P8C	23
1.5.4 Senzory	23
1.6 Programovací jazyk Cycling Max [®]	24
1.6.1 Prostředí Max [®]	25
1.6.2 Základní stavební prvky v Max [®]	26
2 Navržené softwarové řešení	29
2.1 Navržený zvukový algoritmus v prostředí AUT	29
2.1.1 Návrh syntetizéru a použité objekty v AUT	29
2.1.2 Návrhu syntetizéru v AUT	30
2.2 Propojení Teensy pomocí Max [®]	31
2.2.1 Program pro obousměrnou komunikaci s mikrokontrolérem Teensy	31
2.3 Kód pro uživatelské rozhraní a připojené senzory	32

2.3.1	Prvky uživatelského rozhraní	33
2.3.2	Připojný senzorový modul	35
3	Realizace	37
3.1	Architektura zařízení	37
3.2	Zvolené součástky	38
3.2.1	Displej	38
3.2.2	Enkodér	38
3.2.3	Potenciometr	38
3.2.4	Deska se senzorem teploty a tlaku	38
3.2.5	Deska se senzorem osvětlení a přiblížením	39
3.3	Vytvoření obvodu	39
3.3.1	Diagram zapojení	39
3.3.2	Vytvoření plošného spoje	39
3.3.3	Zapojení konektorů pro připojení externích senzorů	41
3.4	Menu a uživatelské prostředí	42
3.5	Zapouzdření zařízení	44
4	Praktické využití nástroje	49
	Závěr	51
	Literatura	53
	Seznam symbolů, veličin a zkratk	55

Seznam výpisů

1.1	Příklad programu pro Teensy, který čte stav připojených potenciometrů a vypisuje je do konzole	20
2.1	Výňatek z programu pro demonstraci mapování potenciometru na ovládání dvou různých parametrů	34
2.2	Výňatek z programu pro použití hodnot ze senzoru k ovládání zvukových parametrů	35

Úvod

Při vytvoření prvních digitálních syntetizátorů bylo patrné, že tento způsob zpracování a modulování signálu přinesl obrovský potenciál rozšíření kreativních úprav zvuku. Otevřely se dveře ke komplikovanějším průběhům, rychle se rozšiřujícímu množství efektových procesorů a převážně, automatizaci jednotlivých parametrů zvuku, za účelem dosažení nových barev. Základní parametry zvuků lze na syntetizátorech zpravidla nastavit pomocí otočných a posuvných ovladačů, které způsobí v obvodu příslušnou změnu. Z příchodem výkonnějších a dostupnějších počítačů v začátcích devadesátých let dvacátého století můžeme pozorovat rozvíjející se trend popularity programů určených k syntéze zvuku pro domácí počítače. Popularita těchto programů v dnešní době pravděpodobně několikrát převyšuje samostatné digitální syntetizátory s vlastním fyzickým uživatelským prostředím. To je sice možno suplovat libovolným MIDI ¹ zařízením, avšak tím mnohdy uživatel trpí na specifickosti rozhraní pro daný nástroj ve prospěch univerzálnosti.

V této bakalářské práci chci vyrobit hardwarový syntetizátor, který bude mít vlastní fyzické uživatelské rozhraní, díky použitému mikrokontroléru bude schopen modulovat parametry zvuku s připojenými senzory měřícími fyzikální veličiny teploty, atmosférického tlaku a světla. Tyto funkcionality jsou zamýšlené jako ukázka možností, jak lze využít mikrokontrolérovou platformu k nestandardnímu ovládní parametrů zvuku. Pro syntetizátor bude vyvinuto komunikační prostředí a doplňkové ovládní v programu Cycling Max[®], kvůli možnosti rozšíření funkcí syntetizátoru - propojení s běžným domácím počítačem, vizualizací dat, potenciálně ovládním přes prohlížeč. Bakalářskou práci chci ukázat, jak lze pomocí programovacích prostředí a knihoven docílit finančně dostupného hardwarového syntetizátoru s velkým potenciálem rozšiřitelnosti.

¹MIDI(Musical Instrument Digital Interface) - průmyslový standard pro digitální hudební nástroje

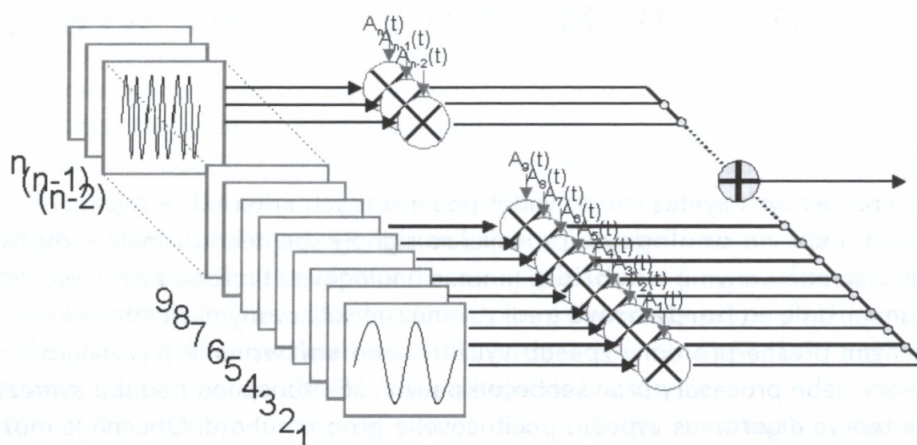
1 Teoretická část studentské práce

1.1 Syntéza umělého zvuku

Sic lze vysledovat elektronické nástroje až k Prokopu Divišovi a jeho nástroji Denis D'or zkonstruovaném kolem roku 1750, první aplikace v této kapitole uvedených zvukových syntéz je o století mladší. Byly k nim potřeba elektromagnetické nástroje, jejichž prototypy vznikaly od 19. století. K jejich další evoluci došlo vznikem elektronických nástrojů, jež je plně rozvinuly až k dnešním digitálním variantám [4]. V této kapitole jsou popsány základní druhy syntéz, běžně využívaných v existujících syntetizátorech, které budou využity v této práci.

1.1.1 Součtová (additivní) syntéza

Princip součtové, neboli additivní, syntézy je založen na pozorování, že všechny přirozeně vyskytující se akustické signály lze rozložit na fundament - základní harmonickou složku a vyšší harmonické složky, f_2, f_3 [1]. Tento poznatek, který zformuloval Joseph Fourier v 19. století, je znám dnes jako Fourierův zákon. Z toho vyplývá, že lze libovolný zvuk složit z konečného počtu harmonických složek. Schéma součtové syntézy je vyobrazeno na obr. 1.1. V případě součtové syntézy je užívanou praktikou skládání komplexního zvuku nejen z harmonických složek, ale i z inharmonických - to je tzv. nefourierovský přístup k součtové syntéze. Součtová syntéza harmonického typu ku příkladu využívají elektronické varhany firmy Hammond, s typicky devíti harmonickými složkami. [4]



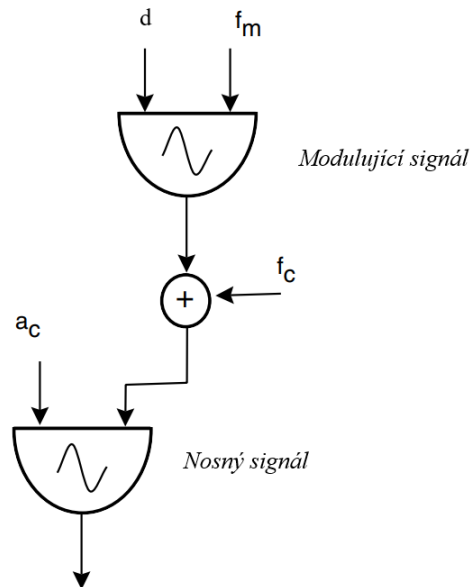
Obr. 1.1: Schéma součtové syntézy s N harmonickými [4]

1.1.2 Rozdílová (Subtraktivní) syntéza

Rozdílová syntéza, také někdy nazývaná subtraktivní, používá jako zdroj vstupního zvuku komplexní signál s mnoha vyššími harmonickými složkami. Ten následně pomocí skupiny filtrovacích prvků, měnících podobu kmitočtového spektra – podíl intenzit jednotlivých harmonických složek na výsledném zvuku, vytváří podobu výsledného signálu[3]. Vzhledem ke skutečnosti, že filtry můžeme pouze zvýraznit či potlačit určitou část kmitočtového spektra, využívají se při tomto druhu syntézy signály se širokým kmitočtovým spektrem. Prakticky to jsou např. na harmonické složky bohaté kmity (obdélníkové, pilové, trojúhelníkové), generátory šumu, impulzy, či krátké nahrané úseky, tzv. *sample* [5].

1.1.3 Modulační syntéza frekvenční *FM* (Frequency modulation) syntéza

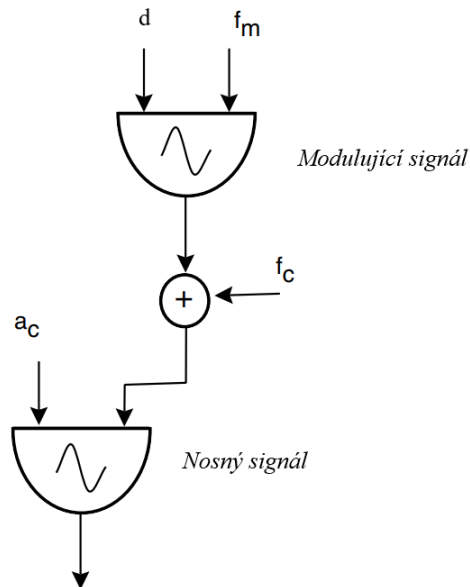
Princip modulační syntézy je založen na modulování kmitočtu prvního oscilátoru kmitočtem oscilátoru druhého. První oscilátor označujeme názvem nosný, druhý jako modulující, či modulátor[3]. Těmto propojeným signálům se předávají primární parametry – kmitočet nosného signálu (f_c), kmitočet modulátoru (f_m) a modulační index (d , také někdy jako „hloubka modulace“). Principiální schéma *FM* modulace je vyobrazeno na obr. 1.2. Pro elektronickou *FM* syntézu se propojují jednotlivé oscilátory, jejichž kmitočty leží ve slyšitelném spektru. Zpravidla se používá od 2 do 6 oscilátorů. Praktickou výhodou *FM* syntézy jsou obrovské zvukové možnosti získatelné relativně jednoduchými prostředky, avšak, na rozdíl od výše uvedených typů syntéz z podkapitol 1.1.1 a 1.1.2 nemá *FM* syntéza ekvivalent u akustických nástrojů, a proto nelze předem dobře odhadnout výsledek zvukového charakteru bez důkladné znalosti *FM* syntézy [4].



Obr. 1.2: Signálový diagram *FM* syntézy s pěti parametry

1.1.4 Modulační syntéza amplitudová *AM* (Amplitude modulation)

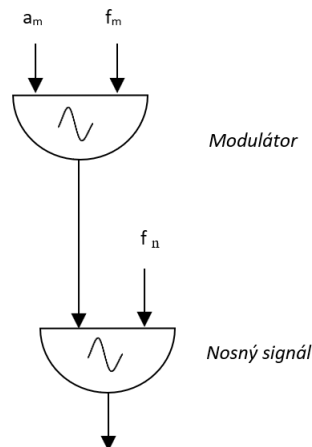
Amplitudová modulace vzniká v situaci, kdy modulátor řídí amplitudu nosného signálu. Typickým praktickým jevem, který je ve své podstatě amplitudovou modulací, je efekt *tremolo*, kdy je nosný signál modulovaný signálem, který má kmitočet pod hranicí slyšitelného spektra. Nižší hranice slyšitelného spektra se uvádí 16 Hz [1]. Klasická amplitudová modulace sčítá výstup modulátoru s modulačním offsetem a_c . Výsledné kmitočtové spektrum obsahuje v případě jednoduché *AM* tři kmitočty – kmitočet nosné vlny, kmitočet získaný odečtením kmitočtu a_m od kmitočtu a_n a kmitočet vzniklý součtem kmitočtů a_m a a_n . Na obr. 1.3 je ukázán signálový diagram *AM* syntézy.



Obr. 1.3: Signálový diagram AM syntézy s

1.1.5 Modulační syntéza kruhová RM (Ring Modulation) syntéza

Kruhová modulace je variantou amplitudové modulace. Při tomto způsobu modulace je amplituda nosného signálu ovlivněna pouze modulačním signálem. Výsledné spektrum signálu neobsahuje původní kmitočet nosné vlny, nýbrž rozdíl a součet kmitočtů f_c a f_m . Tím vznikne signál s nejasným tónem [6]. Diagram RM (Ring Modulation) lze vidět na obr. 1.4.



Obr. 1.4: Signálový diagram *RM* syntézy

1.2 Efektové procesory

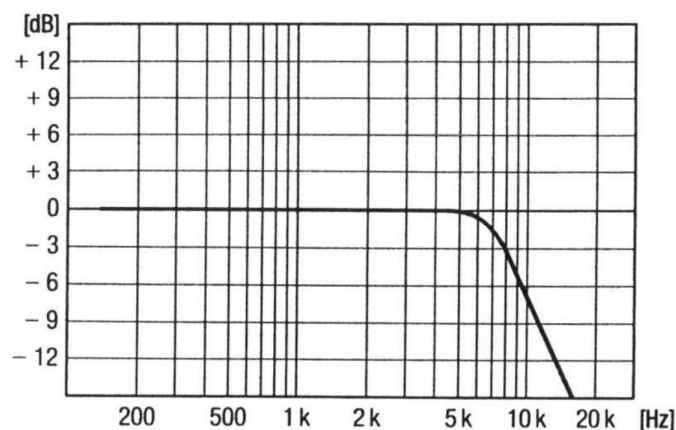
Mimo úpravy zvuku změnou nezávislých proměnných, které nám poskytují od základu jednotlivé druhy zvukové syntézy je u moderních syntetizátorů je možnost aplikovat na zvukový signál několik různých efektových procesorů. Ty mohou změnit nejenom podobu kmitočtového spektra, ale i jejich časový průběh. Efektové procesory uvedené v této kapitole jsou obsaženy v knihovně objektů, jež budou použity v návrhu programu syntetizátoru, a budou využity ve výsledné realizaci této práce.

1.2.1 Efektové procesory měnící kmitočtové spektrum

Pro změnu kmitočtové spektra zvukového signálu se používá různých návrhů filtrů. Ty jsou pro svůj účel uzpůsobeny tak, aby měly dostatečnou šířku pásma pro použití se zvukovým signálem. Všechny uvedené filtry mají společný parametr – mezní kmitočet. Je to kmitočet, při kterém poklesne modul kmitočtové charakteristiky o 3dB od maxima. Za tímto mezním kmitočtem je již signál utlučen podle strmosti filtru [2] – směrnice, udávající rozdíl krajních hodnot amplitud na jednu oktávu (tj. množina kmitočtů mezi mezní frekvencí a její harmonickou), včetně jich samých. Etablované strmosti filtrů jsou např. 6 dB/okt., 12 dB/okt., či 24 dB/okt..

Filtr typu Dolní propust - *DP*

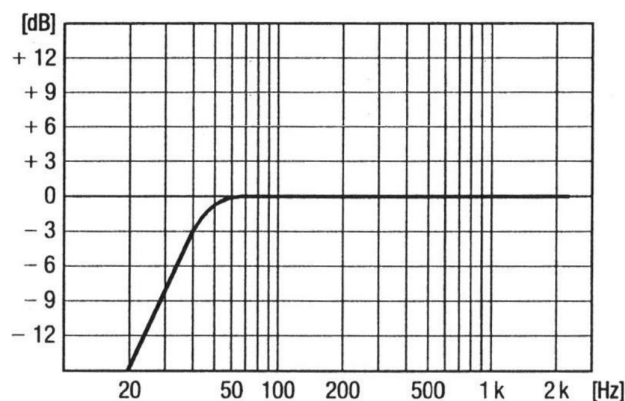
Filtr typu dolní propust nechává spektrum nezměněné do dosáhnutí mezního kmitočtu. Všechny kmitočty nad tento mezní kmitočet jsou potlačeny podle strmosti filtru. [1]. Na přiloženém obr. 1.5 je vidět kmitočtové spektrum filtru typu dolní propust s mezním kmitočtem o hodnotě 8 KHz, a strmosti filtru 12 db/okt..



Obr. 1.5: Kmitočtová charakteristika filtru typu dolní propust [2]

Filtr typu Horní propust - *HP*

Filtr tohoto typu má obdobnou funkci jako filtr typu dolní propust. Narozdíl od něj, filtr typu horní propust do dosažení mezního kmitočtu signál potlačuje podle strmosti filtru, nad mezní kmitočet nechává spektrum nezměněné. Na přiloženém obr. 1.6 je vidět kmitočtové spektrum filtru typu horní propust s mezním kmitočtem hodnoty 40 Hz, a strmosti filtru 12 db/okt..



Obr. 1.6: Kmitočtová charakteristika filtru typu horní propust [2]

Filtr typu Pásmová propust - *PP*

Pásmová propust je kombinací filtrů *HP* a *DP* v kaskádním zapojení [4]. Protože je kombinací obou, má dva mezní kmitočty, odpovídající mezním kmitočtům jednotlivých filtrů. U filtru typu pásmová propust se někdy uvádí jeho třetí parametr - střední kmitočet filtru, f_0 . Ten je geometrickým součinem mezních kmitočtů jednotlivých filtrů, f_{HP} a f_{DP} , viz rovnice 1.1 [5].

$$f_0 = \sqrt{f_{DP} \cdot f_{HP}} \quad (1.1)$$

1.3 Efektové procesory uplatňující zpoždovací linku

Tato kategorie efektů využívá umělé zpoždění signálu, kdy je zvuková informace uložena do mezipaměti a je následně přehrána buď samostatně či v kombinaci s původním signálem. Nezpžděný signál označujeme jako „přímou větev“, zpžděný signál jako „zpoždovací linku“. Tím může vzniknout efekt ozvěny(echo). Pro ten je důležitá vlastnost lidského vnímání odraženého zvuku. Pokud je mezi přímým a odraženým signálem příchozím do sluchového aparátu prodleva delší než 100 ms, začínáme jej vnímat odděleně [4].

Na principu zpožděné linky fungují níže uvedené efekty - *Delay*, *Reverb* a *Flanger*.

Delay

Delay, v českém překladu „Zpoždění“, je sám o sobě zpoždovací linkou, je tedy stavebním prvkem všech efektových procesorů, které fungují na tomto principu.

Samostatný Delay většinou dává uživateli přístup ke konečnému množství zpoždovacích linek, u kterých si sám může určit parametry zpoždění a úrovně hlasitosti. Objekt Delay efektu, který je použit v této práci, umožňuje uživateli konfigurovat osm zpoždovacích linek.

Reverb

Reverb je efektním procesorem simulující ozvěnu signálu. Dnešní digitální efektní procesory typu reverb již pracují se simulací zadaného prostoru – toho je dosaženo několika linkami typu delay pro simulaci množiny zpožděných signálů s různou hodnotou zpoždění, vypočítaných na základě rozměrů napodobovaného prostoru. V pokročilejších variantách obsahuje efektní procesor i sérii filtrů, upravujících signál tak, aby co nejvíce odpovídal kmitočtovému spektru zvuku znějícím v napodobovaném prostoru.

Flanger

U efektního procesoru typu flanger je využita zpoždovací linka zpravidla s nižšími hodnotami v řádu desítek milisekund. Rozdílným je od efektů Reverb a Delay tím, že je časové zpoždění modulováno oscilátorem – modulátorem. Modulátoru lze nastavit kmitočet a hloubku modulace. Pro kombinaci takto modulovaného signálu a signálu původního vznikl samostatný efekt s názvem Chorus [2].

1.4 Platforma Teensy

Mikrokontrolér Teensy je mikroprocesorový obvod, také často referovaný výrobcí pro své uplatnění jako vývojová deska. Byla vyvinuta společností PJRC.COM, LLC, jako výkonnější alternativa desek vyvíjených společností Arduino[10]. S platformou Arduino sdílí programovací jazyk a většinu knihoven, tím pádem se těší obrovského množství dokumentovaných ukázek primárně vyvinutých pro zařízení od společnosti Arduino. Vybrán pro tuto práci byl kvůli existenci rozšíření vhodného pro návrh zvukového syntetizéru. Taktéž má dostatečný výkon, relativně vysoký počet digitálních a analogových vstupů pro různá sensorová zařízení, dobrou zpětnou i dopřednou kompatibilitu, a několik dalších vlastností, které jsou podrobněji rozebrány v podkapitolách níže.

1.4.1 O vývojové desce Teensy 3.2

Vybranou deskou pro tuto práci se stala deska Teensy, model 3.2. Vydána v roce 2014 si svou kvalitou a přístupností dokázala vytvořit úctyhodnou uživatelskou základnu.

Níže přiložená tabulka 1.1 udává některé z parametrů desky Teensy 3.2.. Deska je osazena 32-bitovým procesorem typu architektury ARM, flash pamětí o kapacitě 256kb, RAM kapacity 64K. Poskytuje uživateli 33 vstupních pinů, což je pro případné připojení externích senzorických zařízení, či modulů, příhodné. Důležitým atributem umožňujícím zpracování dvoukanálového signálu je ten, že porovnáními s jinými populárními vývojovými deskami, jako je např. Arduino Mega či SparkFun RedBoard Artemis dva AD převodníky. [9].

Parametr	Hodnota	Jednotka
Kmitočet procesoru	72	MHz
Kmitočet procesoru dosažitelný přetaktováním	96	MHz
Velikost Flash paměti	256	kiloBajt
Šířka pásma	192	MB/s
Velikost Cache paměti	256	Bajt
Velikost RAM paměti	64	KiloBajt
Velikost EEPROM paměti	2	KiloBajt
Počet kombinovaných digitálních vstupů/výstupů	34	-
Výstupní napětí	3.3	Volt
Vstupní napětí	5	Volt
Analogové vstupy	21	-
Počet AD převodníků	2	-
Jmenovité rozlišení převodníků	16	Bit
Použitelné rozlišení převodníků	12	Bit
Bitové rozlišení převodníků	12	Bit
USB výstup	1	-
Výstupů pro sériovou linku	3	-

Tab. 1.1: vybrané parametry mikrokontroléru Teensy 3.2.

1.4.2 Programování mikrokontroléru Teensy

Podobně jako mikrokontroléry společnosti Arduino, program pro zařízení Teensy lze napsat v libovolném IDE (Integrated development enviroment – Vývojové prostředí pro psaní počítačového kódu). Nativní možností je populární *Arduino IDE*, alternativně lze program zkompileovat ve stále populárnějším IDE *Visual Studio Code* s přídatným prostředím pro programování mikrokontroléru *PlatformIO*. V této práci byl použit zejména druhý uvedený editor .Kód, který je pro mikrokontroléry Teensy psán v jazyce C++, využívá knihovny určené pro komunikaci s používaným

mikrokontrolérem. Výpis 1.1 ukazuje příklad programu pro použitý mikrokontrolér. Program globálně inicializuje digitalizaci vstupních napěťových hodnot potenciometrů a následně je v sekci loop vypíše do konzole sériového výstupu.

Výpis 1.1: Příklad programu pro Teensy, který čte stav připojených potenciometrů a vypisuje je do konzole

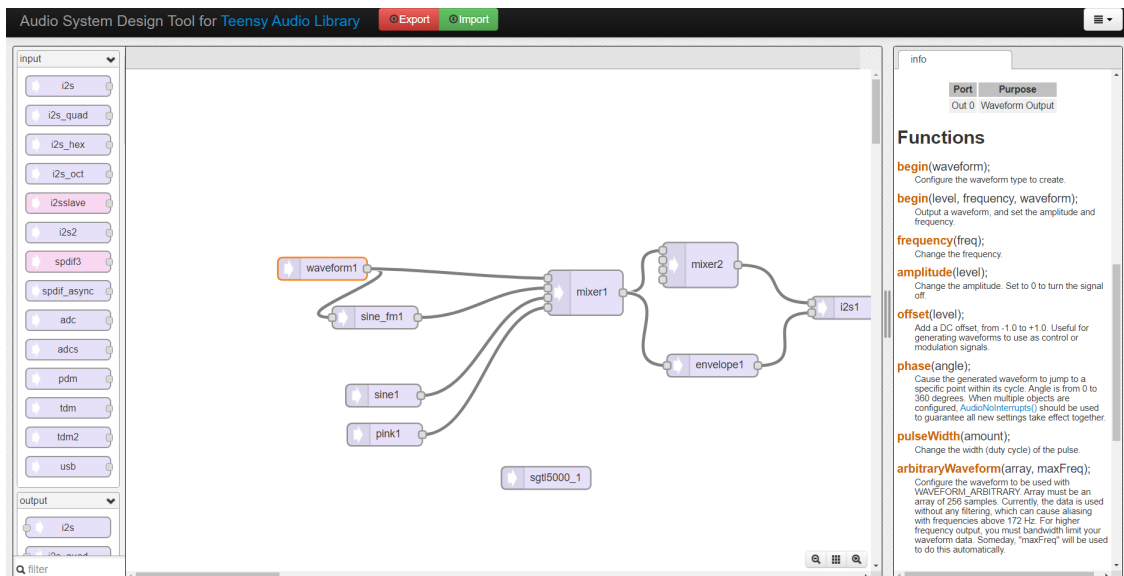
```
#include <Arduino.h>
1
2
3
const int potPin1 = A2;
4
const int potPin2 = A3;
5
6
void setup(){
7
  Serial.begin(9600);
8
}
9
void loop() {
10
  int pot1 = analogRead(A2);
11
  Serial.println(pot1);
12
  int pot2 = analogRead(A2);
13
  Serial.println(pot2);
14
}
15
```

1.4.3 Rozšiřující modul Teensy Audio Shield

Rozšiřující modul Teensy Audio Shield (dále jen „zvukový modul“), je externí elektronický obvod, který lze připojit k mikrokontroléru přímým spojením odpovídajících pinů zvukového modulu a mikrokontroléru. V případě této práce bylo provedeno pomocí připájení kolíkových lišt na mikrokontrolér a dutinkových lišt na desku zvukového modulu. Základními komponentami desky je stereo kodek SGTL5000, zajišťující komunikaci s mikrokontrolérem skrze sběrnici I²C. Tento kodek umožňuje výstup skrze napájenou 3.5 mm zdířku typu TRS, vývody typu line out nezesíleného signálu, a line in vývodu pro vstup nezesíleného signálu.[9].

1.4.4 Návrhový nástroj Audio Design Tool

Pro snadné a intuitivní programování zvukových úprav bylo pro mikrokontrolér se zvukovým modulem vytvořeno společností PJRC.com návrhové prostředí Audio Design Tool (dále jen AUT). AUT je generátorem C++ kódu, jež funguje na principu vizuálních programovacích prostředí. Uživatel tohoto druhu programovacího prostředí, na rozdíl od řádkových editorů klasických programovacích jazyků, jako je například C či TI BASIC, nepíše kód na jednotlivé řádky, nýbrž rozmisťuje funkce



Obr. 1.7: Uživatelské rozhraní nástroje Audio Design Tool

a proměnné, většinou v podobě bloků, na plochu programovacího prostředí. Na obr. 1.7 je příklad návrhu syntetizátoru s možností změny tvaru vln. V případě, že má funkce nějaký vstup či výstup, lze ji napojit k jiné funkci či proměnné pomocí spojovací čar, takzvaných patch propojení. Tento způsob programování se ukázal, kvůli dobře pozorovatelnému signálovému toku, výhodným pro programování zvukových a obrazových programů. O AUT lze říci, že není přímo vizuálním programovacím prostředím, jelikož návrh v něm vytvořený nefunguje jako samostatný program. Podrobnější pohled na jeho funkci je popsán v následující podkapitole.

1.4.5 Princip programování v AUT

Jak bylo popsáno výše, návrh vytvořený v AUT není samostatně funkčním kódem. Zpravidla, funkční bloky v sobě ukrývají C++ objekt, pro který lze vyvolat některou z jeho funkcí. Funkce, dostupné pro libovolný objekt, jsou popsány v okně „info“ (na obr. 1.7 je umístěno vpravo okna). Minimum objektů obsažených v AUT funguje bez vyvolání funkcí, které ve většině případů vyžadují vstupní proměnné. Ty už uživatel funkcím předává v IDE, ve kterém tvoří kód pro mikrokontrolér, viz kapitola 1.4.2. Převod mezi AUT a IDE je realizován tlačítkem „export“, jež vyvolá výpis kódu v textové podobě. Tu uživatel zkopíruje do IDE, kde je následně naprogramována konkrétní funkcionalita jednotlivých objektů vyvoláváním funkcí a předáváním proměnných.

Z napsaného vyplývá, že s dostupnou dokumentací lze objekty tvořit a propojovat přímo v IDE. Tato varianta však, vyžaduje nějaký jiný pomocný druh grafické

representace, je pomalejší a méně uživatelsky přívětivá.

1.4.6 Sběrnice I²C

I²C je dvou vodičová sběrnice od firmy Philips. Díky jejím výhodám – malému počtu linek a nízkému zatížení řídicího procesoru – je oblíbeným řešením komunikace periférií s počítačem. K přenosu dat se využívají dva vodiče označené SDA, použitý pro datový přenos, a SCL, pro přenos hodinového signálu. Pro nejnovější revizi I²C protokolu je maximální kmitočet SCL signálu 5 MHz [15]. Při užití I²C se stanoví řídicí obvod, nazývaný Master, a řízený obvod - Slave. Řídicí obvod generuje hodinový signál a posílá jej obvodům řízeným. Řízený obvod posílá datovou sběrnici logické hodnoty na základě příkazů z řídicího obvodu. [14].

1.5 Použitý hardware

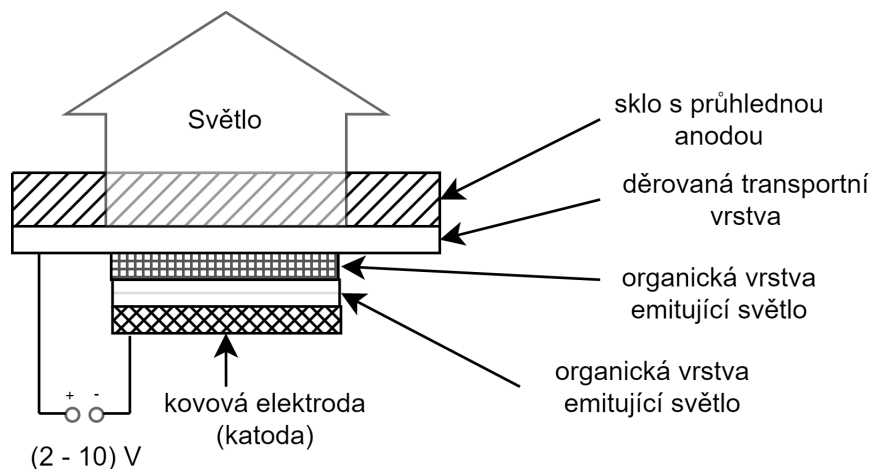
1.5.1 Rotační enkodér

Rotační enkodér je elektromechanický převodník převádějící rotační pohyb hřídele a sekvence digitální impulsů. Dělí se podle principu spuštění impulzního signálu na mechanické, optické a magnetické. V této práci je využit mechanický enkodér. Ten obsahuje kruhovou sérii vodivých ploch. Ty jsou spínány dopadem dvou od sebe vzdálených vodivých kontaktů, jež se pohybují společně s ovládanou hřídelí. Při dopadu kontaktu na vodivou plochu je generována pulzní vlna. Podle rozdílu fáze mezi jednotlivými generovanými vlnami je určen směr rotace otáčení.

1.5.2 Displej

Displej je zobrazovací jednotkou, umožňující komunikaci s uživatelem skrze textové a grafické prvky, jejichž souhrn označujeme zkratkou GUI (Graphical user interface – grafické uživatelské rozhraní). To je v době této práce nejčastějším komunikačním prostředkem mezi uživatelem a zařízením, umožňujícím pozorovat grafické reakce ovládaní v reálném čase. Na trhu lze sehnat nespočet displejů různých parametrů. Z technologického hlediska je zde zmíněno několik nejrozšířenějších typů zobrazovacích jednotek: LED (Light Emitting Diode) — elektroluminiscenční dioda), LCD ((Liquid crystal display – Displej z tekutých krystalů), OLED (Organic Light Emitting Diode – grafické uživatelské rozhraní) [15]. V této práci je využit displej technologie OLED. Ta funguje na principu přeměny elektrické energie na světlo a využívá vlastnosti elektroluminiscence, při které materiál, kterým prochází elektrický proud, emituje světlo. V případě technologie OLED je světlo emitujícím prvek to vysoce svítivý

pevný organický materiál, umístěný mezi průhlednou a kovovou elektrodou. Na obr. 1.8 lze vidět jednotlivé popsané části na řezu OLED buňkou.



Obr. 1.8: Řez buňkou OLED

Oproti technologii LCD jsou displeje OLED tenčí, jednotlivé body jsou velmi dobře viditelné ze stran, čímž je zajištěn výborný zorný úhel, a displeje mají malý proudový odběr.

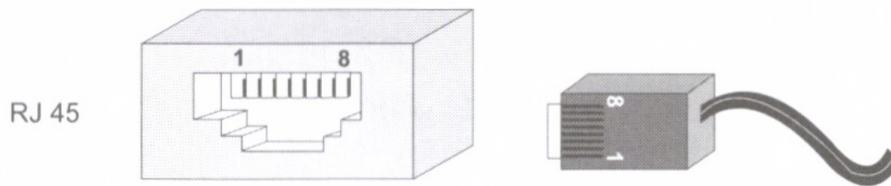
1.5.3 Modulární konektor 8P8C

Modulární konektor 8P8C (8 position 8 contact – 8 pozic, 8 kontaktů) je široce používaným pro přenos ethernetového připojení. Na obr. 1.9 je vedle sebe znázorněna podoba zásuvky a vidlice 8P8C s očíslovanými kontakty. V této práci byl zvolen pro připojení externích senzorů k samotnému zařízení, a to kvůli dostatečnému množství kontaktů – navrhované zařízení využije 2 kontakty pro napájecí větev, a 6 kontaktů pro přenos dat – a popularitě, díky které jsou kabely běžně dostupné v různých délkách a cenových relacích.

Je dobré zmínit, že se pro tento typ konektoru vžilo chybné označení „RJ-45“, a to natolik, že je často používán samotnými výrobci součástek, či v odborné literatuře. „RJ-45“ však označuje pouze zapojení pro konkrétní rozhraní přenosu dat [16].

1.5.4 Senzory

Senzor (a také „snímač“) je prvek převádějící neelektrickou veličinu na elektrickou. Často vzniká ambivalence mezi použitím tohoto slova jak pro základní funkční část měřicího obvodu, tak pro celý obvod. Proto často v odborné literatuře rozlišujeme



Obr. 1.9: Číslování kontaktů konektoru 8P8C [14].

pojem „čidlo“ pro funkční část měřicího obvodu, a pojem „senzor“ pro obvod obsahující čidlo [11].

Senzor teploty

Čidel teploty je na základě principu několik druhů, vyjmenujme zde např. rezistivní, piezoelektrický či piezoresistivní. V této práci bude použit senzor s piezoresistivním čidlem. Piezoresistivní čidlo funguje díky složení několika vrstev křemíkových a hliníkových materiálů s různými koeficienty teplotní roztažnosti. Při změně teploty na sebe jednotlivé vrstvy působí silou, jež ohýbá kombinaci vrstev, a vzhledem k piezoresistivnímu jevu tímto působením tlaků v zařízení vzniká časově rozdílný odpor [13].

Senzor osvětlení

V cizojazyčné literatuře jsou označovány zkratkou ALS (Ambient light sensors – senzor okolního osvětlení). Lze na trhu najít senzory těchto typů technologií čidel: fotoelektrické, fotodiodové a integrované obvody měřící osvětlení. Poslední zmíněná varianta má řešit nedostatky fotodiodových a fotoelektrických obvodů. Integrovaný obvod měřící osvětlení obsahuje polovodičovou fotodiotu konvertující světlo na elektrický proud. V této práci bude použit digitální integrovaný obvod pro měření osvětlení. Ten obsahuje obvod pro zesílení výstupu fotodiody, a analogově-digitální převodník, který posílá data do připojeného I² „master“ zařízení [12].

1.6 Programovací jazyk Cycling Max[®]

V této práci navržený syntetizátor je navržen jako tzv. stand-alone zařízení, tj. zařízení nevyžadující ke své funkci připojení k osobnímu počítači. Jedním z vedlejších cílů této práce je rozšířit jeho funkčnost ovládáním a úpravou parametrů skrze osobní počítač s nainstalovaným systémem Windows. Toho je v této práci dosaženo pomocí

programovacího vizuálního programovacího jazyka Max[®], taktéž znám v kombinaci se zkratkou jako Max/MSP. Pro tuto práci je použitý Max[®] ve verzi 8.

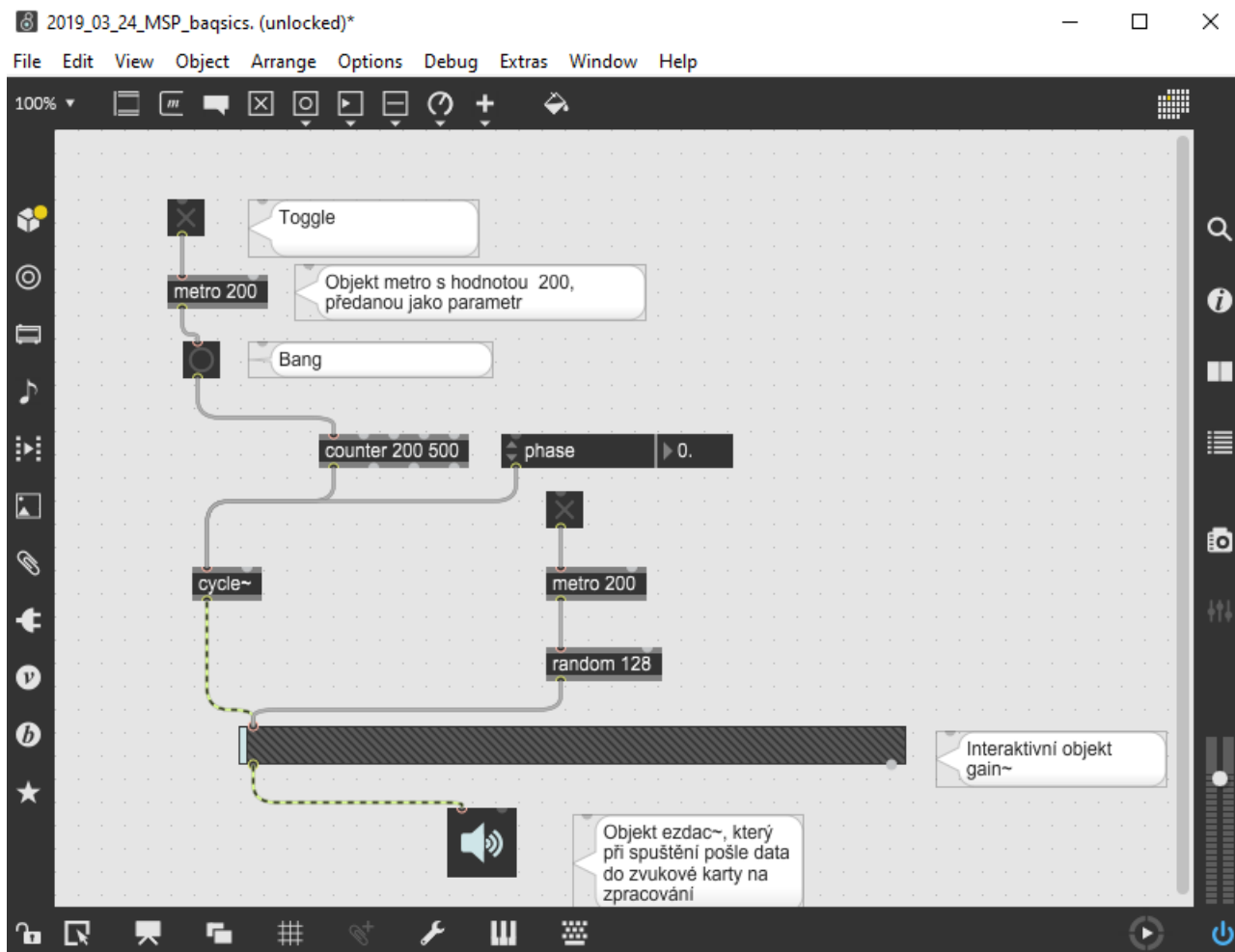
Vznik Max[®] se datuje do roku 1987, kdy v pařížském institutu IRCAM(Instut de Recherche et Coordination Acoustique/Musique). Vývojář Miller Puckette a Philippe Manoury zde pracoval na novém softwaru, jež by podle jeho slov „navázal komunikaci mezi hudebníkem a virtuálním orchestrem, jež by si však stále uchovávala charakter hudebníka“. Jeho práce vyústila v první verze Max[®] vydanou v roce 1990 [7]. Zmínku si zde zaslouží v roce 1995 zveřejněná Pucketteova open-source¹ obdoba Max[®] nazvaná Pure Data, jež se dnes těší velké popularitě.

Max[®] je na rozdíl od prostředí AUT, popisovaném v kapitole 1.4.4, plnohodnotným vizuálním programovacím jazykem. Pro uživatele to znamená, že Max[®] má vlastní kompilátor, konzoli, debugovací nástroje a další nástroje pro tvoření programu dostupných v populárních IDE. Max[®] v nejnovější verzi č. 8 podporuje npř. zpracování obrazového signálu, až osmikanálového zvukového signálu. Vzhledem k faktu, že k syntéze zvuku slouží v této práci mikrokontrolér Teensy, tyto funkcionality nebudou využity.

1.6.1 Prostředí Max[®]

Obdobně jako v AUT se v jazyku Max[®] programuje pokládáním objektových bloků na plochu, kde jim následně uživatel předává proměnné a propojuje objekty mezi sebou. Tím je zajištěno předávání dat mezi objekty. Objekt může přijímat určité množství vstupů, to je realizováno propojením výstupního uzlu (tzv. *outlet*) datového bloku, a vstupního uzlu (tzv. *inlet*) požadovaného objektu. Popsaným způsobem se tvoří celkový program libovolné komplexnosti, v terminologii programu Max[®] je takový program označován jako *patch* [7]. Na obr. 1.10 je ukázka vytvořeného patche, automaticky zvyšující frekvenci sinusového signálu, jež posílá do zvukového výstupu.

¹Open-source - Software se zveřejněným zdrojovým kódem



Obr. 1.10: Patch vytvořený v programu Max[®], verze 8

1.6.2 Základní stavební prvky v Max[®]

Prostředí Max[®], jak už bylo napsáno, poskytuje uživateli několik objektů, pomocí kterých uživatel staví výsledný program. Obsahuje na jednu stranu funkce, jež uživatel se základní znalostí klasických programovacích jazyků pozná, protože jsou názvem a funkčností analogické svým protějškům v jazycích, jako je C++, Javascript atd.. Z těch lze zde uvést např. funkce *for*, *if*, *else*, logické operátory či funkci *print*. Na stranu druhou Max[®] má několik, jež jsou obdobou základních datových typů v klasických programovacích jazycích, ale podléhají nomenklatuře programu Max[®]. Níže je v odrážkách krátce popsáno několik základních typů objektů, bez kterých se při tvorbě programů v Max[®] lze těžko obejít. Jejich grafickou podobu a názorné použití ukazuje na obr. 1.11, kde jsou označeny komentáři uvnitř samotného *patche*.

- *Button* - Button vyvolává na svém výstupu zprávu *Bang*. Ta je základním datovým typem v Max[®]. Lze jej připodobnit generátoru impulzu, který vyvolá

funkci připojeného objektu.

- *Toggle* - Toggle je funkčně obdobou datového typu *Boolean*, proměnné, která může nabýt logické hodnoty 1, či 0.
- *Number* - Number blok funguje jako proměnná, která do sebe ukládá číslo formátu *int* - celočíselnou hodnotu.
- *Message* - Message je primárně určen k předávání zprávy uvnitř message bloku jako parametru dalším funkcím.
- *Comment* - Komentářový blok nemá žádný inlet ani outlet, slouží k popisu tvořeného programu. Podobně jako používané datové prvky a objekty je lze umisťovat kamkoli do plochy *Patche*.



Obr. 1.11: Patch demonstrující použití základních prvků Max[®]

2 Navržené softwarové řešení

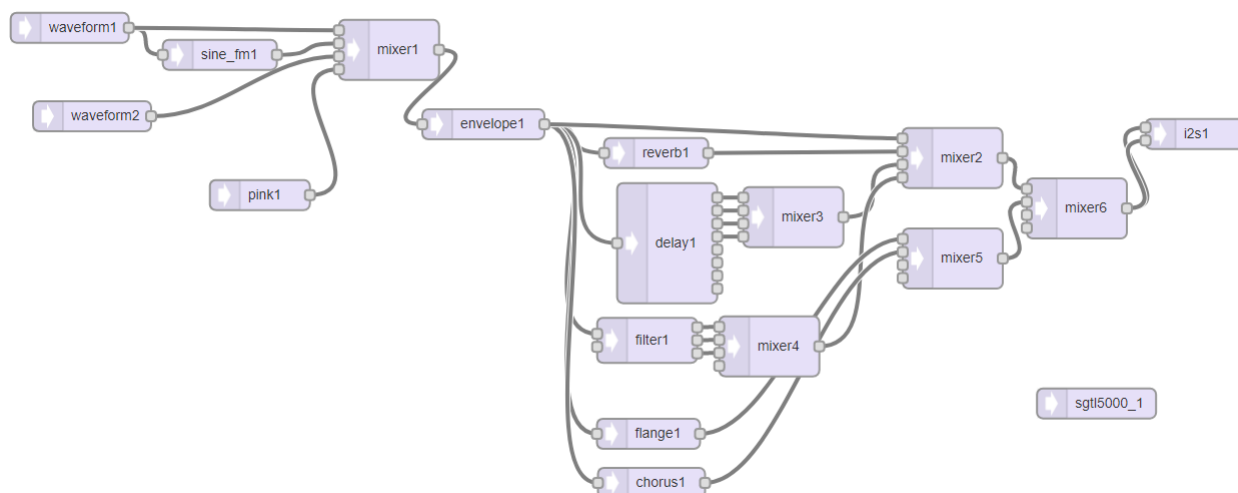
Vzhledem k povaze hardwarového syntetizéru je tato kapitola rozdělena tématické okruhy obdobné okruhům řešícím jednotlivá návrhová prostředí v kapitole 1. Těmi jsou návrh zvukového programu v AUT, programové řešení v kódu nahraném na mikrokontroléru Teensy, a komunikační řešení v programu Max[®]. V těchto jednotlivých podkapitolách je popsán navržený princip funkce, který je využit v napsaném programu pro navrhované zařízení

2.1 Navržený zvukový algoritmus v prostředí AUT

Zvukový program navržený v AUT udává, jaké kombinace syntetizovaných signálů a efektů se mohou dostat na zvukový výstup zařízení. Připomeňme skutečnost, že výsledný kód generovaný v AUT je následně importován do C++ kódu nahraného do mikrokontroléru Teensy. Po jeho spuštění již nelze při zapnutém mikrokontroléru přepojovat vstupy a výstupy objektů mezi sebou, lze pouze pomocí funkcí přepínat varianty, jež nám použité objekty poskytují. Z předešlého vyplývá, že je žádoucí navrhnout signálové cesty v AUT tak, aby poskytovali co největší počet kombinací. Uživatel následně objekty spouští.

2.1.1 Návrh syntetizéru a použité objekty v AUT

Na příloženém obr. 2.1 je propojení objektů mezi sebou. Snažil jsem se jím vyhovět své představě, co by zařízení mělo zvládnout. Použité objekty odpovídají většině metodám syntézy a filtrace zvuku, popsaných v teoretickém úvodu této práce.



Obr. 2.1: Návrh zvukového algoritmu v programu AUT

2.1.2 Návrhu syntetizéru v AUT

V této kapitole jsou krátce popsány jednotlivé objekty, které jsou použité v návrhu zvukového programu, viz 2.1.

Waveform

Generátor kmitů, který posílá ze svého výstupu signál o tvaru vlny, jež si uživatel vybere z 8 poskytnutých možností. Podle typu vlny může objektu předat parametry kmitočtu, amplitudy, fáze, či šířky pulzu.

ModulatorFM

Objekt umožňující FM syntézu. Nosným signálem je v tomto případě harmonický signál, modulovaný vstupní vlnou, kterou tomuto objektu předává objekt *waveform1*.

Mixer

Mixer je v této práci používán pro dvojí účel. Lze jím měnit poměry mezi jednotlivými vstupy, jejichž sečtený signál přivede na výstup. Také však může fungovat jako přepínač, který rozhodne, který ze čtyř vstupních signálů pošle na výstup.

Reverb

Objekt umožňující použít jednoduchý reverb. Požaduje vstupní parametr zpoždění.

Delay

Objekt poskytující uživateli možnost využít až 8 zpoždovacích linek. Každé z nich lze předat parametr zpoždění mezi přímým signálem a zpoždovací linkou. *Delay* objekt je omezen maximální

Filter

Objekt *filter* má tři výstupy, každý z nich odpovídá jednomu typu filtru - HP, DP a PP. Můžeme mu pomocí jednotlivých funkcí předat parametr mezního kmitočtu a rezonance Q.

Envelope

Objekt, skrze který lze na příchozí signál aplikovat ADSR(Attack, delay, sustain, release – tzv. obálka zvuku) dynamickou obálku. Ta příchozímu signálu určí amplitudový průběh závislý na parametrech časů jednotlivých složek A,D,S a R.

2.2 Propojení Teensy pomocí Max[®]

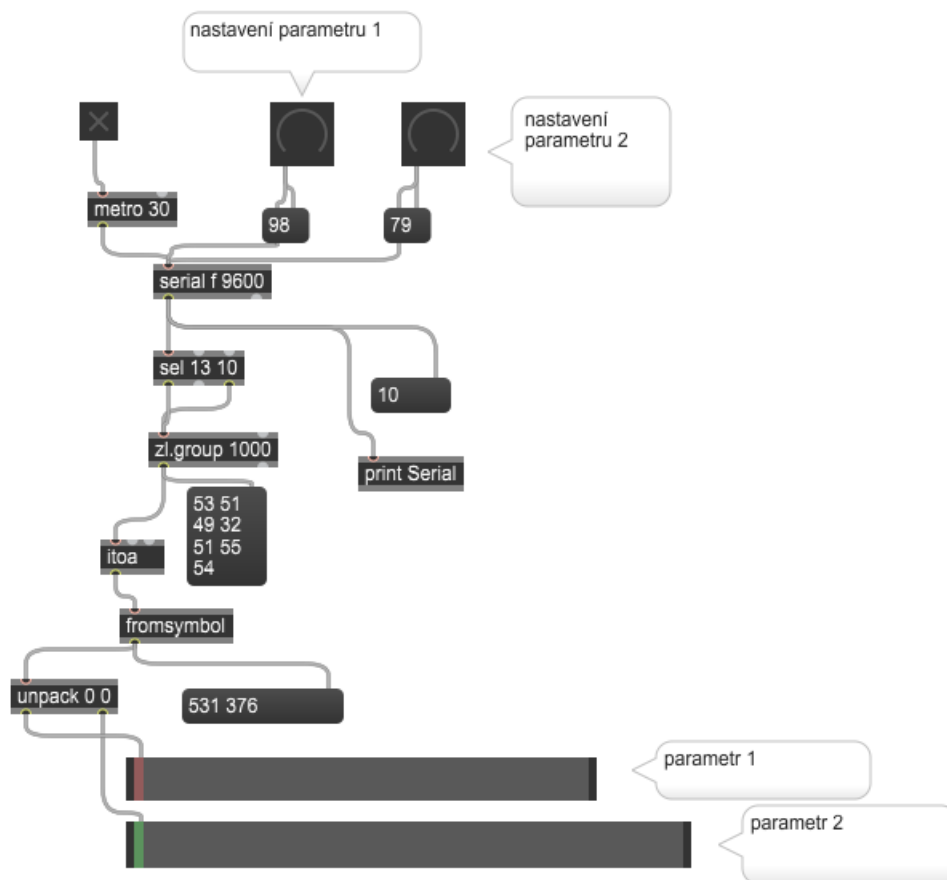
V této kapitole popíšeme navržený princip pro obousměrnou komunikaci s Teensy skrze osobní počítač s programem Max[®].

2.2.1 Program pro obousměrnou komunikaci s mikrokontrolérem Teensy

Teensy komunikuje s osobním počítačem skrze patch - program vytvořený v Max[®]. Je přiložen na obr. 2.2. Zajišťuje odesílání a přijímání dat skrze sériovou komunikaci. V navržené verzi umožňuje uživateli poslat skrze sériový port do mikrokontroléru dvě hodnoty - Parametr 1 a Parametr 2. Pro to je použit objekt *serial*, sloužící pro sériovou komunikaci mezi počítačem s běžícím patchem, a připojeném zařízení vyhovující standartu UART (Universal Asynchronous Receiver/Transmitter – standart asynchronní komunikace). Tento objekt pro své fungování potřebuje poskytnout dva parametry. Prvním z nich je znak, jehož pořadí v abecedě odpovídá číslu, které operační systém přiřadil portu s připojením sériovým zařízením, v ukázce návrhu je to port č. 6., tedy *serial* a *unpack* zajišťuje správnou interpretaci číselných hodnot příchozích z Teensy. Zpráva příchozí sériovou linkou je kódovaná v ASCII ¹. Každá zpráva poslaná mikrokontrolérem počíná ASCII znakem 10, a končí ASCII znakem

¹(American Standard Code for Information Interchange – standard pro kódování znaků v informatice číselnými hodnotami)

13. Tyto znaky jsou v uvedeném patchi vyfiltrovány a výstupem probíraného stromu jsou dvě hodnoty předávané mikrokontrolérem - v tomto případě jsou to hodnoty 531 a 376 vypsané v nejnižším *message* bloku.



Obr. 2.2: Navržený program v Max[®] pro komunikaci uživatele s mikrokontrolérem

2.3 Kód pro uživatelské rozhraní a připojené senzory

Komunikace mezi uživatelem a syntetizérem, za předpokladu, že jej neovládá skrze Max[®], bude realizováno pomocí elektronických součástek zapojených do analogových a digitálních vstupů mikrokontroléru. Těmi jsou spínače, potenciometry, enkodéry a externí senzor, jemiž v této práci jsou senzor osvětlení obsahující funkci měření vzdálenosti, a senzor teploty. V této kapitole uvedu část počítačového kódu pro použité prvky, jež zajistí komunikaci mezi součástkou a mikrokontrolérem. Výpisky z programů uvedené v této podkapitole jsou oproti použitému programu zjednodušené kvůli ilustraci principu funkcí. Celý program využitý v zařízení je přiložen k této práci v podobě digitální přílohy.

2.3.1 Prvky uživatelského rozhraní

Uživatelské rozhraní bude realizováno pomocí primárně rotačních prvků. Výhoda návrhu syntetizéru na digitální platformě je možnost přiřazení jednomu kontrolujícímu prvku různé parametry tzv. mapováním. Realizace tohoto však musí počítat s tím, že každý zvukový objekt může vyžadovat vstupní parametr ve znatelně odlišných číselných rádech. Například, pro kmitočet oscilátoru je vhodné uvažovat hodnoty od 20 do 10000 Hz, ale např. pro zesílení lze uvažovat pouze hodnoty od 0 do 1. Tento problém je řešen pomocí vytvoření mezí pro každý parametr, které vstupní hodnota přiřazeného kontrolního prvku, násobena koeficientem hloubky modulace nastaveného uživatelem, nemůže přesáhnout. Demonstrace základního mapování je uvedena ve výpise 2.1. Ten při zvolení příslušné položky v menu a stisknutí tlačítka upraví hodnotu prvku do použitelných mezí a přiřadí ji proměnné předávané jako parametr zvukovému objektu.

Výpis 2.1: Výňatek z programu pro demonstraci mapování potenciometru na ovládání dvou různých parametrů

```
1 // Mapovani promenne counterEncoder1 na kmitočet oscilátoru
2
3 void loop() {
4     //inicializace proměnné, jež čte stav enkodéru
5     long newEncoder1;
6     newEncoder1 = encoder1.read();
7     //funkce stanovující meze vystupních hodnot enkodér
8     //č.1 na spodní mez 0, a horní mez 1000, s krokem 1
9     counterEncoder1 = formatEncoder(newEncoder1);
10    ...
11    // Při zvolení položky Encoder1frequencyOSC1 v menu
12    //a stiskem tlačítka se provede příkaz, přidružující
13    //OSC1inputNumber číslu, vyhrazenému pro zvolený kontroler
14    if(button.fallingEdge() &&
15    menuItem = Encoder1frequencyOsc1){
16    OSC1inputNumber = 1;
17    }
18    ...
19    // Funkce switch, jež přiřadí hodnotu hodnotu
20    //counterEncoder1, upravenou o meze a koeficient
21    switch (OSC1inputNumber){
22        case 1:
23            min1 = 0;
24            max1 = 10000;
25            frequency1 = setBounds(counterEncoder1,
26            min1, max1)*multiplier1;
27            break;
28        ...
29    }
30    }
31    // Předání proměnné frequency1 jako parametr kmitočtu
32    //prvnímu oscilátoru waveform1
33    waveform1.frequency(frequency1);
34
35 }
```

Mapování prvků sebou přináší

2.3.2 Připojný senzorový modul

Princip ovládání zvukových parametrů senzory fyzikálních veličin je demonstrován na senzoru BMP280, který je piezorezistivním senzorem teploty a tlaku. Tento senzor bude použit i ve výsledné práci. Díky své popularitě v Arduino komunitě pro něj existuje knihovna `<BMP280.h>`, která je použita v níže přiložené ukázce, viz výpis 2.2. Samozřejmě lze podobně jako ve výpisu 2.1 mapovat hodnoty ze senzoru na různé parametry.

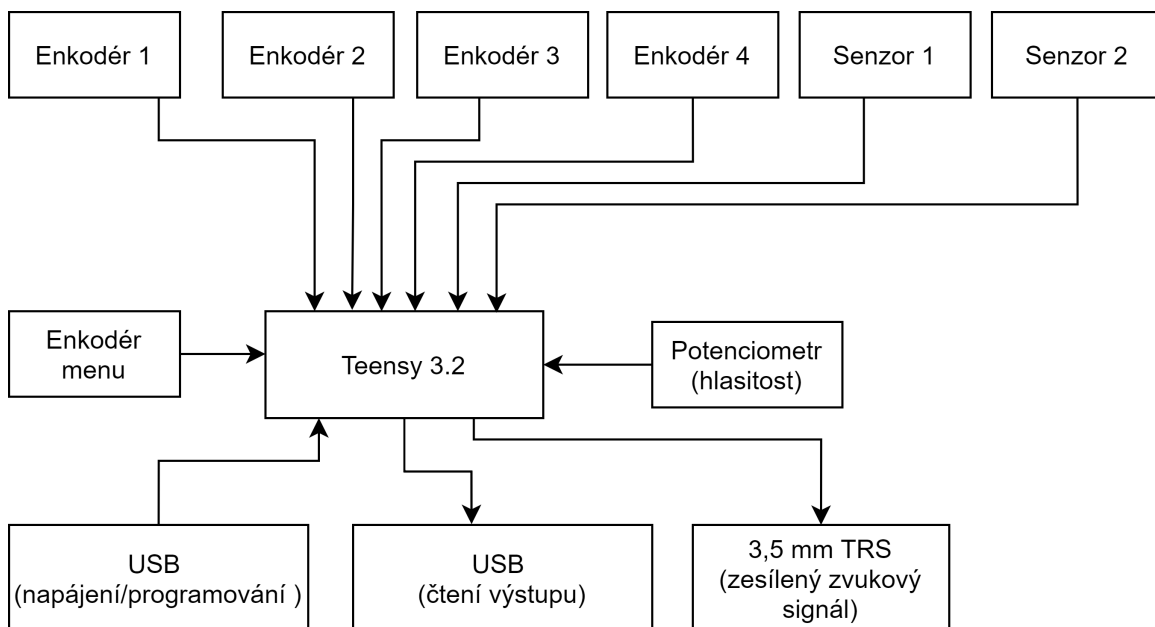
Výpis 2.2: Výňatek z programu pro použití hodnot ze senzoru k ovládání zvukových parametrů

```
float vlhkost; 1
float teplota; 2
/*Promene ukladajici udaje relativni vlhkosti 3
v procentech a naměřené teploty ve stupních Celsia 4
*/ 5
void setup() 6
{ 7
  Serial.begin(9600); 8
  BMP280.begin(); 9
} 10
 11
void loop() 12
{ 13
  tlak = BMP280.read(); 14
  teplota = BMP280.readTemperature(); 15
  /*konverze hodnot na desetiné číslo 16
s maximální hodnotou 1.0 17
*/ 18
  mixer1.gain(1, teplota/40.0); 19
  mixer1.gain(3, vlhkost/70.); 20
} 21
 22
```


3 Realizace

3.1 Architektura zařízení

Zařízení bylo navrženo tak, aby obsahovalo 6 vstupních ovládacích prvků, jež lze přiřadit libovolným zvukovým parametrům, poskytnutých v nabídce GUI. Těmito zařízeními jsou 4 rotační enkodéry a 2 externí sensory. Na vstup zařízení je připojen pátý enkodér, pevně přiřazený pro orientaci uživatele v GUI, a potenciometr, jež je pevně přiřazený k ovládání koncové hlasitosti. Posledním vstupem, jež je uživatelsky nekontrolovatelný, je vstup samotného mikrokontroléru, se zásuvkou typu „USB Micro-B“. Skrze něj je k zařízení připojený napájecí zdroj. Ten by měl vyhovovat specifikacím výrobce mikrokontroléru, kterými jsou napětí v rozmezí 3,3V až 5V, a proud dosahujícího maximálně 1A [9]. Lze také pomocí tohoto vstupu zařízení dodatečně programovat. Tento vstup slouží i jako výstup v případě posílání dat do připojeného počítače pomocí programů využívajících sériové připojení. Primárním výstupem zařízení je zásuvka typu 3,5mm TRS, do které je posílaný zesílený zvukový signál. Blokový diagram signálových cest popsanych v této podkapitole je uveden na obr. 3.1.



Obr. 3.1: Blokový diagram vstupních a výstupních zařízení

3.2 Zvolené součástky

3.2.1 Displej

Použitý displej je technologie OLED, jež byla popsána v podkapitole 1.5.2. Konkrétní model používá sběrnici I²C, díky které sdílí datové kontakty s připojenými senzory.

- Rozlišení: 128 x 64 px
- Rozměry zobrazovací plochy: 29.42 x 14.7 mm
- Vstupní napětí: 3.3V - 5.0V

3.2.2 Enkodér

Zvolený rotační enkodér je typu „Keyes KY-040“. Je dodáván již s plošným spojem, což usnadňuje připájení k obvodu a upevnění k zapouzdření. Níže jsou vypsány parametry enkodéru ze stránky prodejce [18]. Zde může být problematický parametr životnosti čítající 30 000 cyklů, kterého, vzhledem k povaze ovládání nástroje, nebude pravděpodobně těžké dosáhnout v rozmezí životnosti zařízení.

- Napájení: 5 V
- Pulzů na otáčku: 20
- Životnost: 30 000 cyklů

3.2.3 Potenciometr

Potenciometr pro udávání úrovně výstupní hlasitosti je zmiňován v produktové stránce přídatného zvukového modulu, která udává doporučený odpor 25 k Ω . Byl proto pořízen potenciometr stejného odporu.

- Odpor: 25 k Ω
- Úhel elektrické rotace: 280°
- Úhel mechanické rotace: 300°

3.2.4 Deska se senzorem teploty a tlaku

Zvolenou deskou pro měření teploty a tlaku se stala komerčně dostupná varianta založená na piezorezistivním senzoru typu BMP280 od společnosti BOSCH. Níže jsou uvedeny rozsahy pro měřené veličiny.

- Rozsah měření tlaku: 300 – 1100 hPA
- Rozsah měření teploty: -26°C – 80°C

3.2.5 Deska se senzorem osvětlení a přiblížením

Vybraná deska je obsahuje senzor integrovaného obvodu pro měření osvětlení APDS-9930 od společnosti BROADCOM. Níže jsou uvedeny meze pro měření obou vstupních veličin.

- Maximální měřená hodnota přiblížení: 10 mm
- Minimální hodnota osvětlení: 0.01 lux

3.3 Vytvoření obvodu

Obvod byl během vývoje zapojen v kontaktním nepájivém poli a propojen vodiči. Po potvrzení funkčnosti a zhodnocení konečnosti stavu zařízení byl obvod zapájen na plošný spoj. V této podkapitole je podrobněji rozepsán postup zhotoveného obvodu.

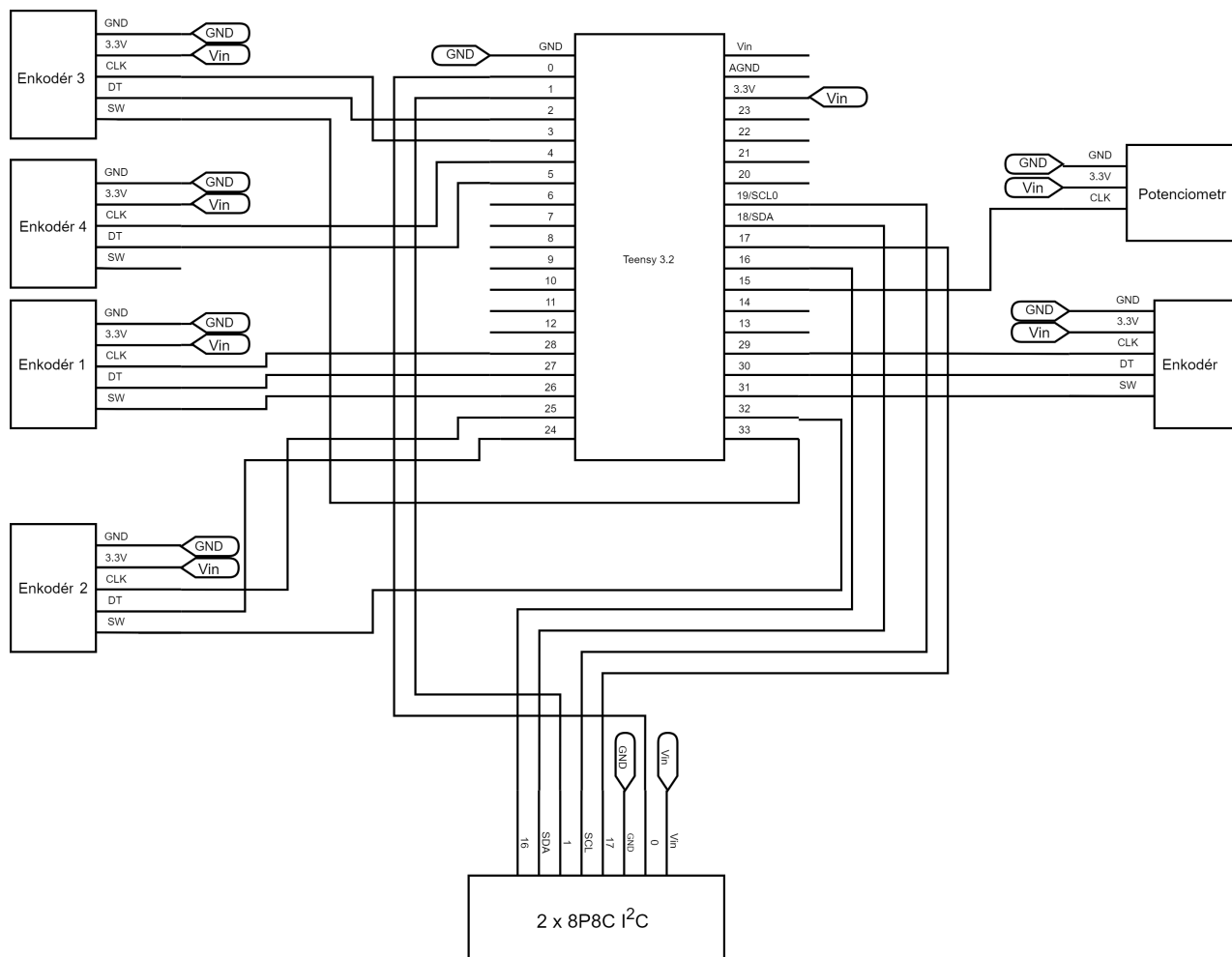
3.3.1 Diagram zapojení

Podle zapojení prototypu zařízení v kontaktním nepájivém poli byl zhotoven diagram zapojení, viz obr. 3.2. Ukazuje připojení jednotlivých kontaktů ovládacích prvků a vstupů k desce mikrokontroléru. Obvod vypracovaný v této práci využívá všechny digitální vstupy, jež mikrokontrolér nabízí. Nezapojené vývody, viditelné na diagramu zapojení, plně využívá připojený Audio Shield, není je proto možné využít k dalším periferiím. Z tohoto důvodu zůstal nezapojen kontakt „SW“ Enkodéru č. 5, sloužící k přenosu signálu při stisknutí tlačítka enkodéru. Vzhledem k faktu, že tlačítka enkodérů čísel 2 - 5 nejsou při ovládání zařízení nijak využita, nebyl tento nedostatek více řešen. V případě potřeby připojení více kontaktů ovládacích prvků by bylo možné využít npř. 8-bitový střádač.

3.3.2 Vytvoření plošného spoje

Po záznamu diagramu v předchozí podkapitole byl obvod rozpojen a vyjmut z kontaktního nepájivého pole, aby by přenesen na plošný spoj. Vzhledem k relativní jednoduchosti obvodu byl použit univerzální vrtaný plošný spoj s kontakty sdruženými po třeticích, viz obr. 3.3. Na něj byly podle uvedeného diagramu ručně připájeny součástky a přípojná místa. Součástky byly propojeny s deskou přímým napájením na vývody v desce součástky, a na vývody ve vytvořeném plošném spoji. Pro spojení mikrokontroléru a obvodu Audio Shield byly do plošného spoje napájeny dutinkové lišty, do kterých se mikrokontrolér zasune, viz obr. 3.4.

Tento způsob připojení byl zvolen z důvodu zajištění jednoduché opravitelnosti zařízení v případě selhání mikrokontroléru, a dopředné kompatibilitě s aktuálními

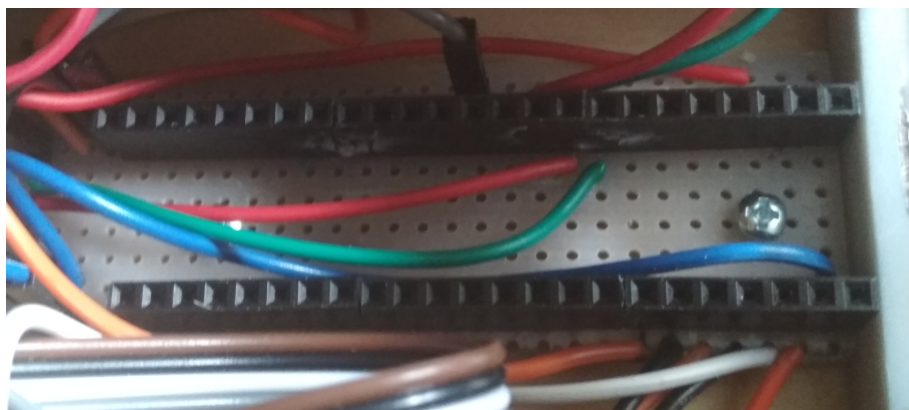


Obr. 3.2: Diagram propojení jednotlivých periferií s mikrokontrolérem

modely řady Teensy. Těmi jsou v době této práce mikrokontroléry Teensy 4.0 a Teensy 4.1.



Obr. 3.3: Hlavní plošný spoj s připájenými vývody součástek



Obr. 3.4: Dutinkové lišty v hlavním plošném spoji pro připojení mikrokontroléru

3.3.3 Zapojení konektorů pro připojení externích senzorů

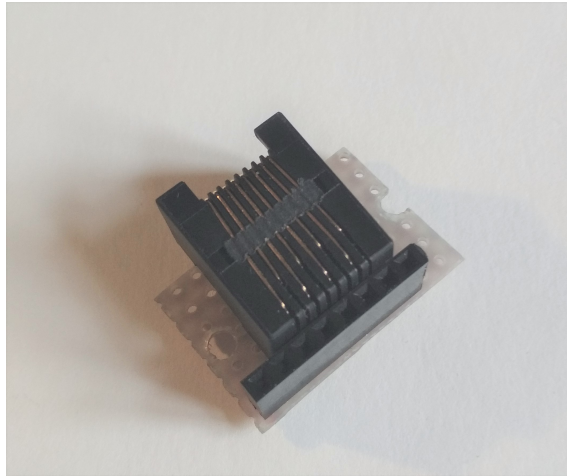
Zvolené konektory pro připojení typu 8P8C jsou zmíněny v podkapitole 1.5.3. Jelikož většina senzorů využívá sběrnici I²C (viz. 1.4.6), byla prozkoumána možnost existujícího standardu rozhraní pro přenos I²C signálu pomocí zkoumaného konektoru. Žádný takový standard nebyl autorem nalezen, a proto bylo pro zařízení zhotoveno rozhraní, snažící se ctít posloupnosti rozmístění kontaktů na většině senzorů. Výsledné rozmístění kontaktů, číslované podle obr. 1.5.3 je vypsáno v tab. 3.1. Toto řešení je proprietární pouze pro konstruované zařízení. Ovšem k povaze této práce je použitý způsob vyhovující.

Pořadí 8P8C kontaktu	Kontaktu na mikrokontroléru
1	16
2	SDA
3	1
4	SCL
5	17
6	GND
7	0
8	Vin

Tab. 3.1: Tabulka propojení kontaktů 8P8C a kontaktů mikrokontroléru

Podle navrženého systému zapojení konektorů byla zkonstruována deska s dvěma zásuvkami typu 8P8C, jejíž blokové znázornění je uvedeno na obr. 3.2. Ta je připojena k hlavnímu plošnému spoji a následně zapouzdřena společně s mikrokont-

rolérem. Pro senzory byly vytvořeny dva plošné spoje s připájenými dutinkovými lištami pro možnou výměnu senzorů, viz obr. 3.5.



Obr. 3.5: Zkonstruovaná deska pro připojení senzoru se zásuvkou typu 8P8C a připájenými dutinkovými lištami

3.4 Menu a uživatelské prostředí

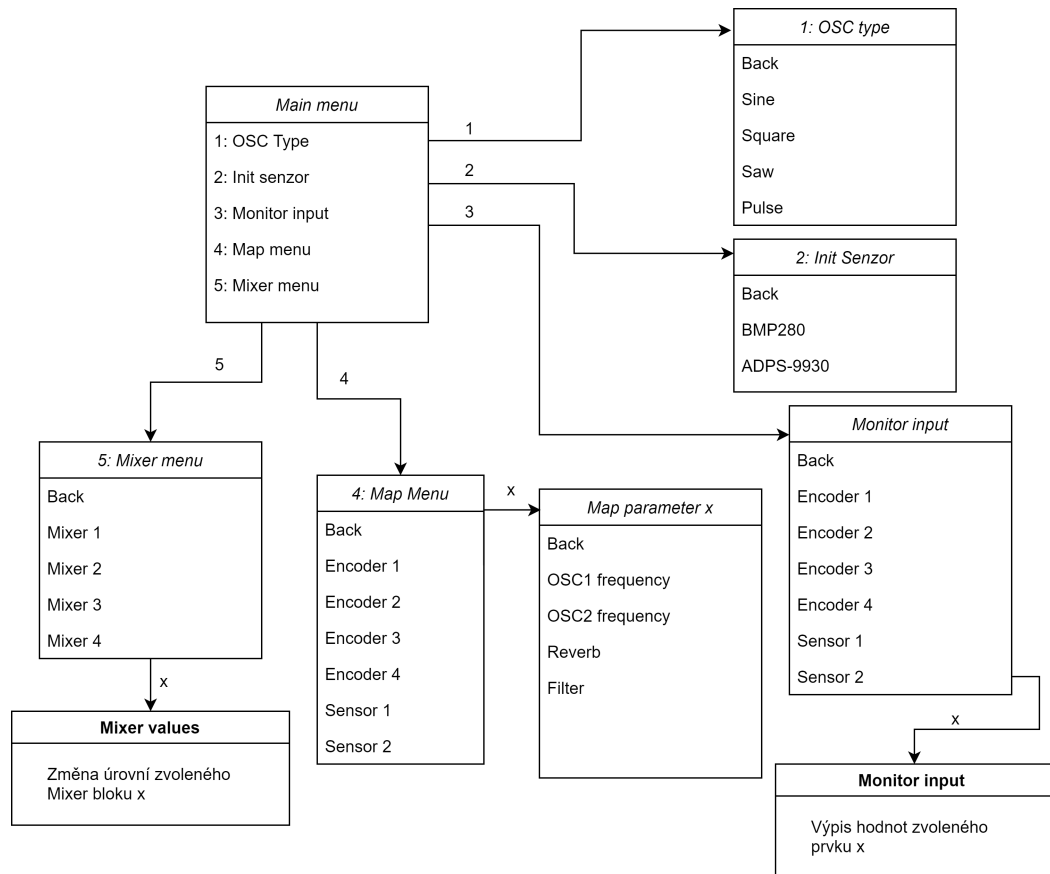
Návrh menu podléhal především počtu manipulovatelných položek a rozměru displeje. Princip všech menu a podmenu je založen na vertikální nabídce - jednotlivé položky jsou listovány pod sebou, v případě zvolení se buď provede akce, nebo se otevře podmenu či stavová obrazovka. Diagram propojení jednotlivých obrazovek mezi sebou je znázorněn na obr. 3.6. Mezi jednotlivými položkami si uživatel vybírá rotací „menu enkodéru“, viz obr. 3.1. Stisknutím tlačítka enkodéru se provede akce podle zvýrazněné položky. V podkapitolách níže je uveden krátký popis jednotlivých položek.

Hlavní menu

V nabídce „Main menu“ si uživatel zvolí výběr položky, které jsou zpravidla podnabídkami sdružující logické celky parametrů.

OSC type

V nabídce „OSC type“ uživatel zvolí typ vlny zvoleného oscilátoru. Kliknutím na jednu z položek může volit mezi sinusovým, trojúhelníkovým, pilovým, čtvercovým, či pulzním signálem.



Obr. 3.6: Blokový diagram jednotlivých menu, podmenu, a výpisových obrazovek

Init sensor

Skrze nabídku „Init sensor“ uživatel inicializuje typ připojeného senzoru. Existence tohoto menu je opodstatněna připojováním a odpojováním externích sensorů za běhu zařízení ¹. To probíhá připojením senzoru k zařízení a následnému zvolení odpovídající položky s názvem připojeného senzoru.

Map menu

V tomto podmenu si uživatel zvolí vstupní prvek, jež chce přiřadit k ovládání zvukového parametru. Po zvolení zamýšleného prvku se objeví navazující podnabídka, kde zvolí zvukový parametr, npř. kmitočet oscilátoru č. 1.

Mixer menu

V „Mixer menu“ uživatel volí mezi jednotlivými „Mixer“ bloky. Ty odpovídají názvům navrženého programu v AUT. Nevýhoda tohoto řešení je však vhodná zna-

¹tzv. Hot Plugging

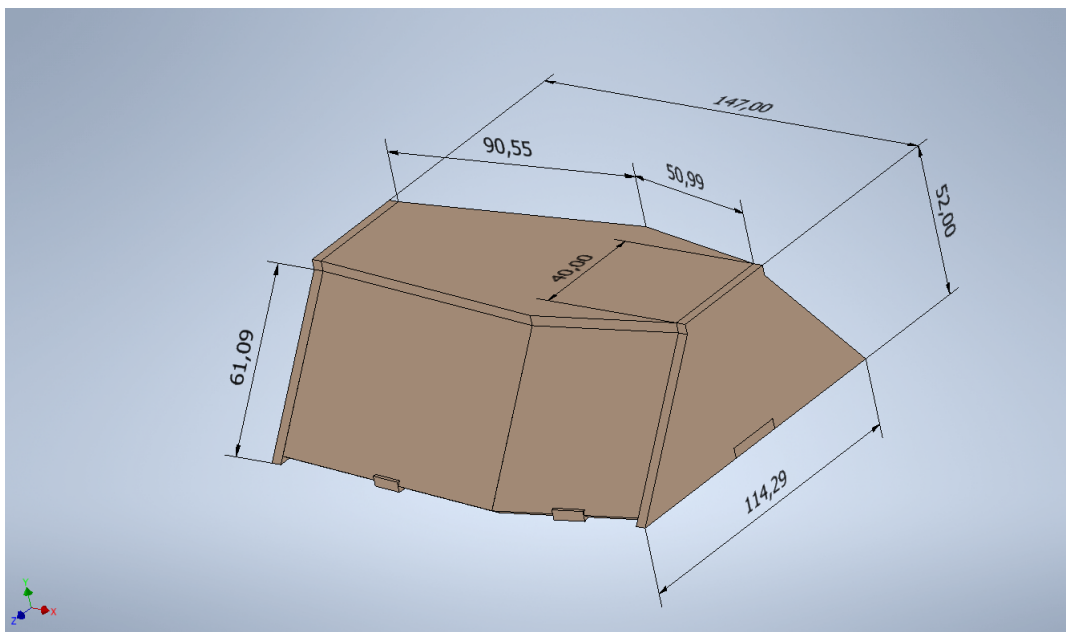
lost uživatele souvislostí mezi jednotlivými „Mixer“ bloky a signálovou cestou. Po zvolení bloku je však uživateli ukázána obrazovka s možností manipulace úrovní jednotlivých vstupů do „Mixer“ bloku, které jsou pojmenovány podle vstupů - npř. „OSC1“, či „reverb“, proto se nestane, že by si zařízení neznalý uživatel vybíral vstup bez vědomí, jakou úroveň hlasitosti ovládá.

3.5 Zapouzdření zařízení

Vzhledem k relativní kompaktnosti použitého obvodu byl navržen tvar zapouzdření odpovídajících rozměrům obvodu. V této části práce bylo kladeno za cíl zhotovit přenosné a ergonomické zapouzdření.

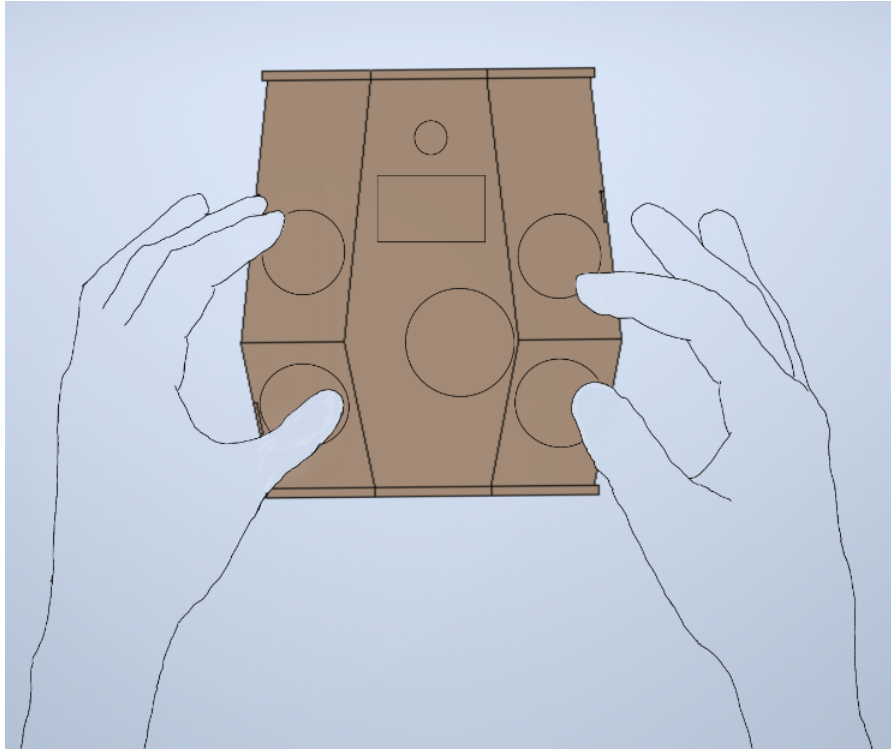
Návrh

Vzhledem k absenci klaviatury byla vnímána potřeba vytvořit fyzickou podobu zařízení, jež umožňuje ovládání více zvukových parametrů současně, je dostatečně kompaktní, a v nejlepším případě motivuje osvojení si nástroje hlouběji. Tyto nároky se promítly do návrhu digitálního trojrozměrného modelu, viditelného na obr. 3.7, který byl vymodelován v počítačovém programu AutoCad Inventor 2020.



Obr. 3.7: Digitální model pouzdra zařízení s okótovanými rozměry v mm

Digitální podoba byla zhotovena na základě finálního tvaru ze série kartonových prototypových modelů, na nichž byl posuzován subjektivní vjem manipulace

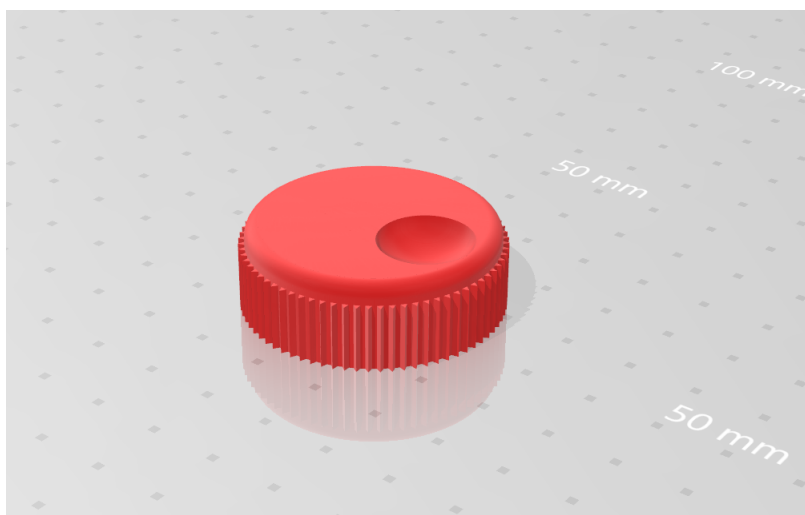


Obr. 3.8: Náčrt umístění rukou na modelu pouzdra se zakreslenou polohou ovládacích prvků

rotačních enkodérů pomocí několika prstů zároveň. Výška zapouzdření je dána výškou obvodu - ta je poměrně vysoká kvůli použitým dutinkovým a kolíkovým lištám propojujícím mikrokontrolér Teensy se zvukovým modulovem. Výsledné pouzdro ve tvaru osmistěnu umožňuje rotaci enkodérů s opřenými dlaněmi o plochu, na níž je zařízení umístěno. Sklon stěn byl kompromisem mezi pohodlím a intuitivností rotace prstů a malou plochou podstavy zařízení. V této části procesu byla také zvolena pohodlná vzdálenost mezi enkodéry na jednotlivých stěnách tak, aby maximální vzdálenost konečků prstů nebyla nepříjemná a zároveň aby v poloze nejmenší vzdálenosti konečků prstů nebylo třeba ohýbat prsty více v kloubech. Náčrt výsledného rozpoložení ovládacích prvků v měřítku s dlaněmi je ukázán na obr. 3.8.

Návrh a zhotovení krytek ovládacích zařízení

Kvůli dosažení jemnosti otáčení enkodéru byla snaha zhotovení krytky enkodéru s dostatečně velkým průměrem, z praktických experimentů byl tento průměr stanovený větší 30 mm. Taktéž byla potřeba na ploše krytky prvku mít nerovnost povrchu umožňující ovládat prvek jedním prstem. Ideální řešením byla krytka ve tvaru rotačního válce s vroubkováním pláště, a půlkulovou prohlubní v podstavě. S tímto tvarem

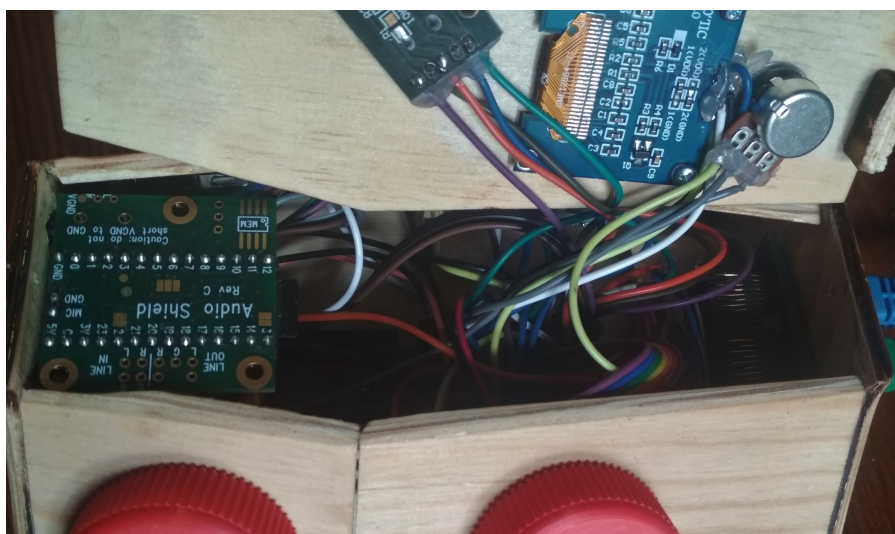


Obr. 3.9: Upravený model krytek na rotační enkodéry

krytky enkodéru se lze setkat npř. u ovládání některých laboratorních zdrojů. Pro účely práce byl využit veřejně dostupný model „Spinner Knob for Rotary Encoder“ od uživatele „MiseryBot“, dostupný na online databázi modelů Thingiverse. Rozměry modelu byly pro tuto práci upraveny, to především kvůli kompenzaci tloušťky použitého dřeva a velikosti stěn. Výsledný průměr krytky činí 37mm, její výška je 13mm. Upravený model krytky je ukázán na obr. 3.9.

Zhotovení pouzdra a kompletace zařízení

Z uvedeného digitálního modelu byl zhotoven výkres, do nějž byly přidány otvory pro ovládací enkodéry, vstupní napájecí konektor, a vstupní konektory pro připojení senzorů. Tento výkres byl následně využit k zhotovení jednotlivých součástí zapouzdření pomocí laserové řezačky, konkrétně modelem Speedy 400 od společnosti Trotec. Tou byly části vyřezány z dřevěné překližky o tloušťce 4 mm. Vyřezané stěny byly vyztuženy a spojeny tmelem, kterým byly vyplněny i vzniklé mezery v konstrukci. Do pouzdra byl instalován plošný spoj se součástkami, jež byly upevněny pomocí vrutů, díky kterým by mělo být proveditelné zařízení nedestruktivně rozebrat v případě úprav či oprav. Horní víko je z tohoto důvodu upevněno také na dvou vrutech. Systém odklopení horního víka, viz obr.3.10, tím přispívá k jednoduchosti potenciální výměny za aktuálnější model Teensy 4.0, díky dopředné kompatibilitě zmíněné v podkapitole 3.3.2. Výsledná podoba zařízení je vyobrazena na obr. 3.11.



Obr. 3.10: Odklopené víko zařízení po odstranění zajišťujících vrutů



Obr. 3.11: Zapouzdřené funkční zařízení s připojenými senzory BMP280 a ADPS-9930

4 Praktické využití nástroje

Sestrojený nástroj může být využitý, vzhledem k použitým sensorům, pro doslova ambientní hru, kdy pomocí pečlivého nastavení parametrů může uživatel využít nástroj bez jeho ovládání v reálném čase. To může hráči odemknout způsob vnímání prostoru jako hudebního nástroje, v jehož měnicích se vlastnostech lze nacházet inspiraci. Vzhledem k použitým konektorům lze použít 8P8C kabely i do délky 10 metrů, což bylo ověřeno autorem. Tento technický parametr umožňuje hráč být relativně svobodný v umístění senzoru, což je vnímána jako jedna z největších výhod navržené konektivity, kdy může hráč umístit např. senzor osvětlení na parapet v opačném rohu místnosti od umístění nástroje. Lze tímto zhudebňovat rušné ulice, východy slunce, průjezdy tunelem a nespočet jiných scénářů, jež uživatele napadne. Tento způsob „hry“ nabírá dále na zajímavosti při uvážení mobilního použití nástroje, např. při běhu, jízdě na kole či jiném dopravním prostředku.

Dalším samozřejmým použitím je hra při aktivním použití ovládacím prvků. Vzhledem k navrženým ovládacím prvkům nelze předpokládat tónovou hru, spíše způsob hraní založený na dojmu expresivity pomocí změny rychlosti rotace otočných prvků. Díky nastavitelnosti rozmezí a jemnosti hodnoty předávané ovládacím prvkem zvukovému parametru může hráč nastavit distinktivnost kroků a může tím tvořit jak chaoticky znějící hudbu s rychle měnícími se parametry, tak kompozice s velmi pomalou změnou parametrů k navození iluzorní statičnosti hudby. Tento způsob hry je samozřejmě umocněný způsobem využití externích sensorů, kde lze například připojený senzor přiblížení použít jako rytmický prvek pravidelným zakrytím rukou hráče, či jak bylo popsáno v odstavci výše, nalézt v prostředí fenomén, který poskytuje podobný průběh - např. rotující listy ventilátoru. V této části bych rád zmínil praktickou zkušenost, jež jsem měl s ovládacím otočných prvků, jejichž způsob nejen pro autora, ale i osoby, jimž byl syntetizér zapůjčen, považovali za velmi uspokojivý, ale při ovládacím více prsty najednou neintuitivní. Nabízí se zde otázka, jaké virtuozity lze s konstruovaným nástrojem v tomto ohledu dosáhnout, což je podmíněno tím, jak dobře je možné si osvojit kruhový pohyb čtyřmi prsty najednou při použitém tvaru zařízení.

Věřím, že s postupným užíváním zařízení bude nacházeno vícero způsobů hry, jež jistě budou narůstat s rozšiřováním softwarové výbavy nástroje.

Závěr

V rámci této bakalářské práce byly popsány základní způsoby syntézy a úpravy zvukového signálu, jež jsou standardy hudebních syntetizérů. Na základě těchto poznatků byl navrhnout a zkonstruován digitální syntetizér s připojitelnými externími senzory ovlivňující uživatelem zvolené parametry zvuku. Tento proces dal vzniknout několika problematikám, např. řešení samotného rozhraní pro připojení senzorů, škálování hodnot pro praktické použití při ovládání jednotlivých parametrů, přiřaditelnost vstupních prvků jednotlivým parametrům a navržení ergonomické a zajímavé podoby zapouzdření, jež by umožňovala nestandardní hru na zkonstruovaný nástroj. Častým tématem této práce byla opravitelnost a rozšiřitelnost zařízení. Ta byla zajištěna do té míry, kdy není možno upravit fyzickou podobu zařízení a jeho napevno umístěných ovládacích prvků, avšak lze vyjmout hlavní obvod zařízení a jednotlivé prvky vyměnit, či opravit. Zařízení je možné vyměnit řídicí mikrokontrolér za jiné modely použité řady popsané v práci. Vzhledem k jednoduché přítomnosti programovacímu vstupu lze upravovat nahraný software, měnit zvukový algoritmus, či upravovat funkce a průběhy ovládacích prvků. Hotovým výrobkem mimo nástroj vznikla tedy prakticky samostatná platforma, jež je možnou rozvíjet způsoby, jež použité součástky umožňují. Jedním z nejočividnějších by byla realizace připojení klaviatury pomocí protokolu MIDI, naprogramování různých průběhů pro běh enkodérů, či zkonstruování rozbočovače pro zapojení více senzorů, a implementace více typů senzorů. Další kategorií vylepšení by mohlo být zahrnutí sequenceru a nahrávacích/přehrávacích funkcí. Mimo samotnou realizaci záměru práce demonstrovala, využití několika komerčně a veřejně dostupných prostředků k realizaci vlastního návrhu. Finální podoba zařízení splnila většinu cílů předevzatých si v semestrální práci, jež této práci předcházela.

Literatura

- [1] SYROVÝ, V. *Hudební akustika*. Praha: Akademie múzických umění, 2003. Akustická knihovna (Akademie múzických umění v Praze. Hudební fakulta. Zvukové studio). 392 s. ISBN 8073319012
- [2] SCHIMMEL, J. *Studiová a hudební elektronika*. 1. Brno: Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací, 2015. ISBN 978-80-214-4452-2
- [3] KOPECKÝ, P. *Základy elektronického zvuku a jeho kreativní zpracování*. Praha: Akademie múzických umění, 2017. 141 s. ISBN 978-80-7331-431-6
- [4] URBAN, O. *Instrumentář elektroakustického zvuku 1*. V Praze: Akademie múzických umění, 2007. p. [5]. ISBN 978-80-7331-115-5
- [5] CIPRIANI, A., GIRI, M. *Electronic Music and Sound Desing: theory and practice with Max/MSP. vol 1.*, Řím: Contemponet s.a.s.,2010 539 s. ISBN 978-88-905484-0-6
- [6] MIRANDA, Eduardo Reck *Computer Sound Synthesis for the Electronic Musician*, 2. vyd. Velká Británie: Focal Press, 2002 563 s. ISBN 0-240-51693-1
- [7] MAZZOLA, G., PANG, Y., HEINZE W. *Basic Music Technology*, Cham: Springer International Publishing, 2018 ISBN 978-3-030-00981-6
- [8] WINKLER, T. *Composing Interactive Music*, Cham: Springer International Publishing, 2018 ISBN 978-3-030-00981-6
- [9] Produktová stránka výrobce PJRC.com *Products*. [online]. 2014. Dostupné z: <https://www.pjrc.com/store/>
- [10] *GETTING STARTED WITH TEENSY* [online]. 2019 [cit. 2020-13-11]. Dostupné z: <https://www.circuito.io/blog/teensy-guide/>
- [11] SPIŠAK, J., IMRAMOVSKÝ M., PENHAKER M. *Snímače a senzory v biomedicíně* [online]. 6. Ostrava: VŠB - Technická univerzita Ostrava, 2007 [cit. 2021-5-29]. ISBN 978-80-248-1607-4. Dostupné z: <https://dnnt.mzk.cz/uuid/uuid:831d1232-c483-442c-a6cd-4466e3d03740>
- [12] ROHM Semiconductor USA, LCC *Ambient Light Sensor (ALS) Applications in Portable Electronics*. [online]. 2009 [cit. 2021-30-05]. Dostupné z: https://www.rohm.com/documents/11303/41217/CNA09016_wp.pdf/65cafa78-92ee-4d06-b2b6-289ddcd70eda?t=1459534857327

- [13] Cai, Chunhua & Tan, Junyan & Hua, Di & Qin, Ming & Zhu, *Piezoresistive temperature sensors fabricated by a surface micromachining CMOS MEMS process*. [online]. Nianfang. (2018). [cit. 2021-30-05]. Dostupné z: <https://www.nature.com/articles/s41598-018-35113-z.pdf?proof=t>
- [14] JAHELKA, M. *Programování jednočipových mikropočítačů PIC*. [online]. Ostrava: VŠB - Technická univerzita Ostrava, 2008 [cit. 2021-23-05]. ISBN 978-80-248-1695-1. Dostupné z: <https://dnnt.mzk.cz/uuid/uuid:d101e8ce-615d-47a3-b71d-6fee2f896160>
- [15] SELECKÝ, Matúš. *Arduino: uživatelská příručka*. 1. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.
- [16] TRULOVE, J. *Sítě LAN: hardware, instalace a zapojení*. [online]. Praha: Grada, 2009 [cit. 2021-23-05]. Profesionál. ISBN 978-80-247-2098-2. Dostupné také z: <https://dnnt.mzk.cz/uuid/uuid:5fe187c0-b3e7-11e3-b833-005056827e52>
- [17] KABELOVÁ, A., DOSTÁLEK L. *Velký průvodce protokoly TCP/IP a systémem DNS* [online] 5., aktualiz. vyd. Brno: Computer Press, 2008 [cit. 2021-23-05]. ISBN 978-80-251-2236-5. Dostupné z: <https://dnnt.mzk.cz/uuid/uuid:7d6d1760-d907-11e8-a5a4-005056827e52>
- [18] *Keyes KY-040 Rotační encoder s tlačítkem* [online]. Produktová stránka prodejce laskaarduino.cz 2021 [cit. 2021-27-05]. Dostupné z: <https://www.laskaarduino.cz/keyes-ky-040-rotacni-encoder-s-tlacitkem/>

Seznam symbolů, veličin a zkratek

DSP	číslicové zpracování signálů – Digital Signal Processing
ASDT	Audio System Design Tool – prostředí pro návrh zvukových funkcí pro zařízení Teensy
IDE	Integrated development enviroment – Vývojové prostředí pro psaní počítačového kódu
LFO	Low frequency oscillator – Nízkofrekvenční oscilátor
ADSR	Attack, delay, sustain, release – tzv. obálka zvuku
UART	Universal Asynchronous Receiver/Transmitter – standart asynchroní komunikace
ASCII	American Standard Code for Information Interchange – standard pro kódování znaků v informatice číselnými hodnotami
ALS	Ambient light sensors – senzor okolního osvětlení
8P8C	8 position 8 contact – 8 pozic, 8 kontaktů
GUI	Graphical user interface – grafické uživatelské rozhraní
LED	Light Emitting Diode) — elektroluminiscenční dioda
LCD	(Liquid crystal display – Displej z tekutých krystalů
OLED	Organic Light Emitting Diode – grafické uživatelské rozhraní
AM	amplitudová modulace
FM	frekvenční modulace
RM	kruhová modulace
a_m	amplituda modulátoru
a_n	amplituda nosného signálu
f_c	kmitočet nosného signálu
f_m	kmitočet modulátoru
f_{HP}	mezní kmitočet filtru typu horní propust

f_{DP}	mezní kmitočet filtru typu horní propust
f_0	střední kmitočet filtru pásmová propust
d	modulační index

DSP (číslicové zpracování signálů – Digital Signal Processing)